# Oracle18c/19c & Centrally Managed Users (CMU)

DB User Management Made Easy

**Simon Pane**

**December 4, 2019**

Pythian

Pythian
love your data®

# Simon Pane

## Pythian Principal Consultant

- ~25 years Oracle experience
- Community Volunteer
- Oracle ACE
- Oracle Certified

Google Cloud Certified – Associate Cloud Engineer
Google Cloud Certified – Professional Cloud Architect
Oracle Certified Professional – Database 12c Administrator
Oracle Certified Expert
Oracle Certified Associate
Oracle Certified Specialist
Oracle ACE

Canada

# Conference and/or Webcast Speaker For



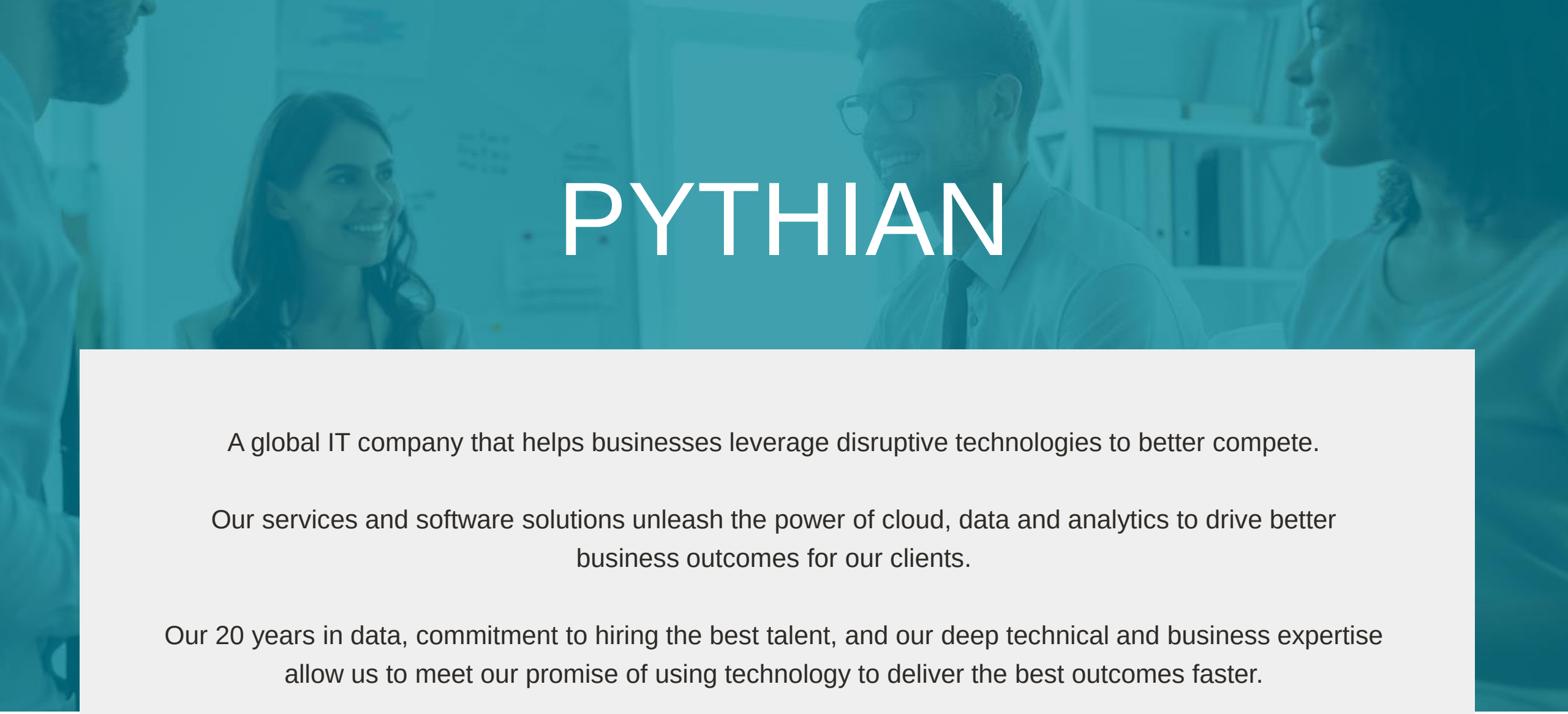












# PYTHIAN

A global IT company that helps businesses leverage disruptive technologies to better compete.

Our services and software solutions unleash the power of cloud, data and analytics to drive better business outcomes for our clients.

Our 20 years in data, commitment to hiring the best talent, and our deep technical and business expertise allow us to meet our promise of using technology to deliver the best outcomes faster.

Pythian

LOVE YOUR DATA

**22**

**Years in Business**

**400+**

**Experts in 35 Countries**

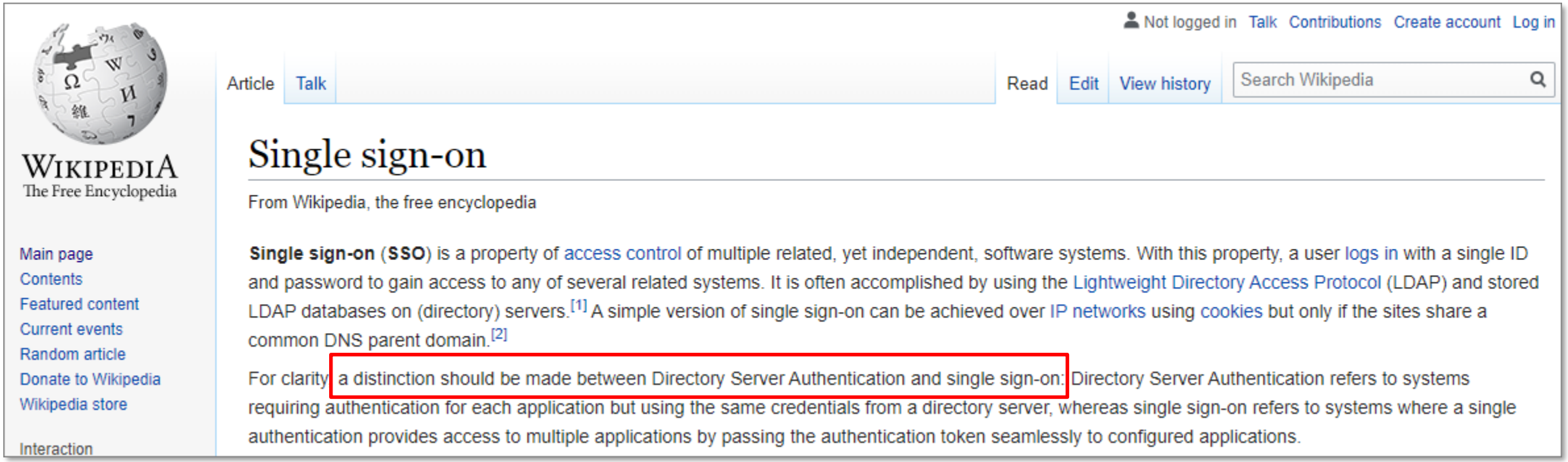
**350+**

**Clients Globally**

# PYTHIAN TIMELINE

| 1997-2012 | 2013-2014 | 2015 | 2016 | 2017 |
|---|---|---|---|---|
| Remote Database Management Services–Oracle, Microsoft SQL Server, MySQL | Cloud emerges, DevOps practice established<br><br>Hadoop practice established<br><br>First Cloud Managed Service | Expanded Open Source–databases Cassandra, MongoDB<br><br>Cloud partnerships with Google, AWS, Microsoft<br><br>Analytics practice established | Competencies grow with Cloud partners–Data, Machine Learning, Migrations, DevOps | 11,000 database systems under Pythian management<br><br>Analytics as a Service launches<br><br>Completed one of the world's most complex Cloud Migrations |

# Quick Definitions



Not logged in Talk Contributions Create account Log in

Article Talk | Read Edit View history | Search Wikipedia

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction

## Single sign-on

From Wikipedia, the free encyclopedia

**Single sign-on** (**SSO**) is a property of access control of multiple related, yet independent, software systems. With this property, a user logs in with a single ID and password to gain access to any of several related systems. It is often accomplished by using the Lightweight Directory Access Protocol (LDAP) and stored LDAP databases on (directory) servers.[1] A simple version of single sign-on can be achieved over IP networks using cookies but only if the sites share a common DNS parent domain.[2]

For clarity a distinction should be made between Directory Server Authentication and single sign-on: Directory Server Authentication refers to systems requiring authentication for each application but using the same credentials from a directory server, whereas single sign-on refers to systems where a single authentication provides access to multiple applications by passing the authentication token seamlessly to configured applications.

- CMU support both "**Directory Server Authentication**" and "**Single Sign-on**"

https://en.wikipedia.org/wiki/Single_sign-on

# What is Achievable

```
PS > echo "
>> set heading off
>> select 'DB_NAME (from v$database) : '||name,
>>        'SESSION_USER                : '||sys_context('USERENV','SESSION_USER'),
>>        'AUTHENTICATED_IDENTITY      : '||sys_context('USERENV','AUTHENTICATED_IDENTITY'),
>>        'AUTHENTICATION_METHOD       : '||sys_context('USERENV','AUTHENTICATION_METHOD'),
>>        'AUTHENTICATION_TYPE         : '||sys_context('USERENV','AUTHENTICATION_TYPE'),
>>        'LDAP_SERVER_TYPE            : '||sys_context('USERENV','LDAP_SERVER_TYPE'),
>>        'ENTERPRISE_IDENTITY         : '||sys_context('USERENV','ENTERPRISE_IDENTITY')
>>   from v$database;
>> " | sqlplus -s -L /@ORCL

DB_NAME (from v$database) : XE
SESSION_USER              : SIMON@STAGECOACH.NET
AUTHENTICATED_IDENTITY    : simon@STAGECOACH.NET
AUTHENTICATION_METHOD     : KERBEROS
AUTHENTICATION_TYPE       : NETWORK
LDAP_SERVER_TYPE          :
ENTERPRISE_IDENTITY       : simon@STAGECOACH.NET


PS >
```

# AGENDA

- Overview of CMU
- One-time Active Directory Config
- One-time RDBMS Home Config
- User and Role Mapping and Testing
- Troubleshooting & Common Issues Reference

# Background & How it Works

# Oracle Possibilities with Directory Services

1. Federate OCI with an IdP
2. Centralise Net Naming Services in AD, OID, or any LDAP compliant directory
3. User management through Enterprise User Security (EUS) and OUD HARD !!!
4. **NEW:** Oracle Database 18c authentication and authorisation for multiple 18c+ databases within Microsoft Active Directory

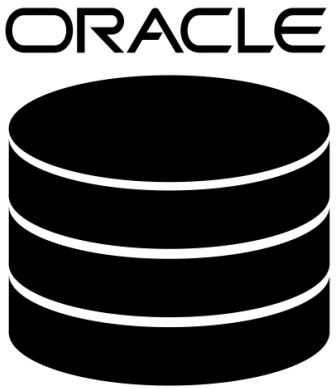
- **No additional licenses required**
- **No additional software tiers to add**
- **Compatible with 11g and 12c clients**

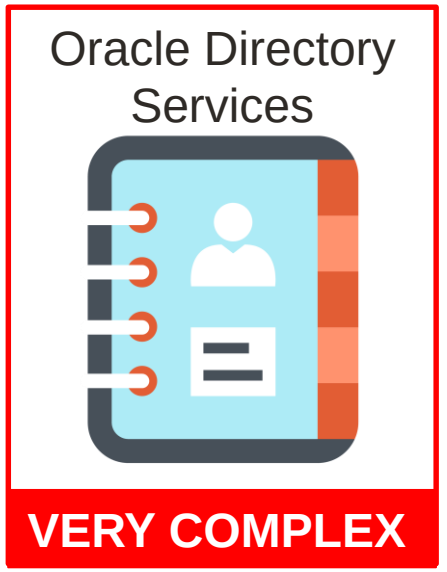
# Some Foundational Basics …

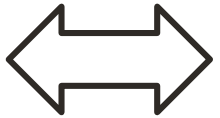
- “Active Directory” (AD) is Microsoft’s customised LDAP Directory Service
  - Supports many common LDAP features and tools
  - Is based on the concept of an AD “schema” which holds properties of objects
- Runs on one or more “Domain Controllers” (DCs)
  - Other services such as DNS often run on the same DCs
- Minimum version for CMU is Microsoft Windows Server 2008 R2

# The Difference is Profound

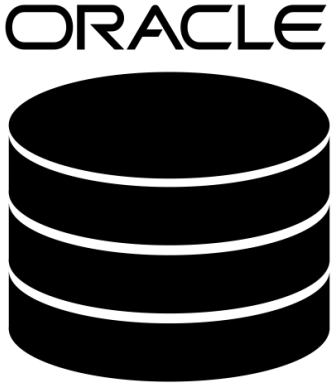
**EUS / OUD and syncing of AD users/groups**



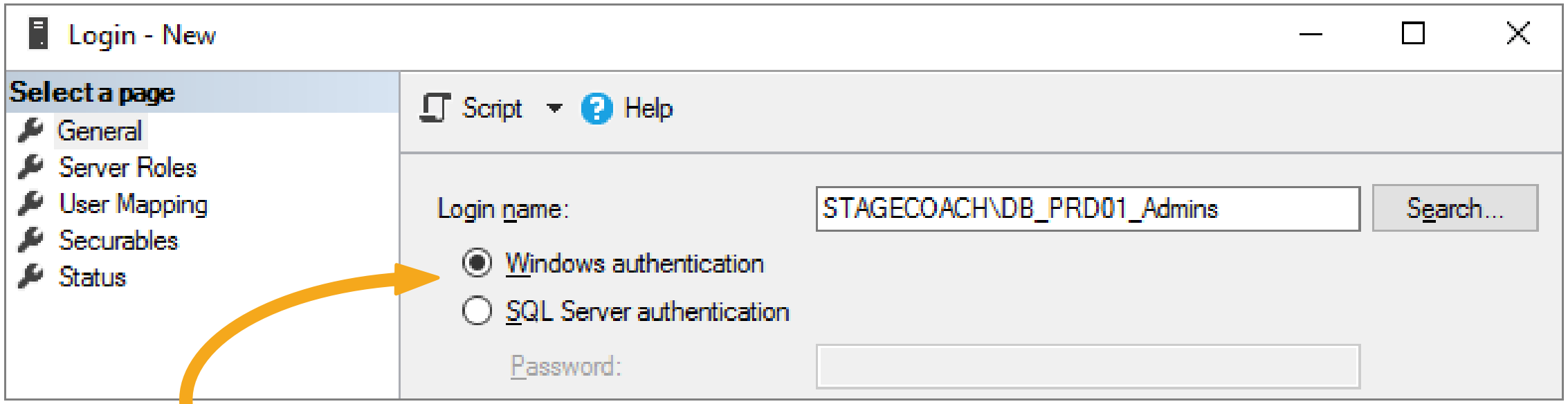

**Database 18c talks directly to AD**




# Why Do We Want To Do This?

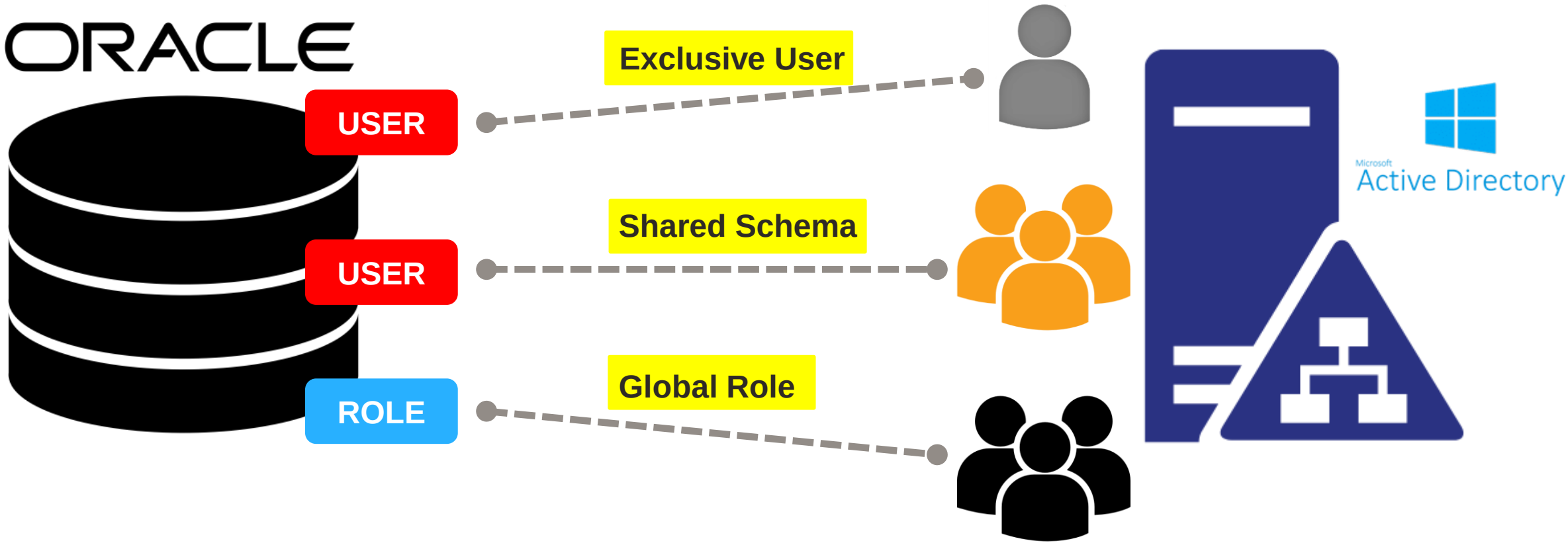
- Centralise (some) DB user management
  - If organisationally using Active Directory, then users are almost certainly added/maintained there anyway
  - Removes user account and user password layer from the database
  - Can leverage Active Directory security groups – map to database roles/privileges
  - Reduced DBA administration workload
  - With shared DB schemas, no onboarding or offboarding at the DB level
- What's not included
  - Integration with any other LDAP directory service – only Active Directory currently

# Similar to SQL Server Integrated Logons



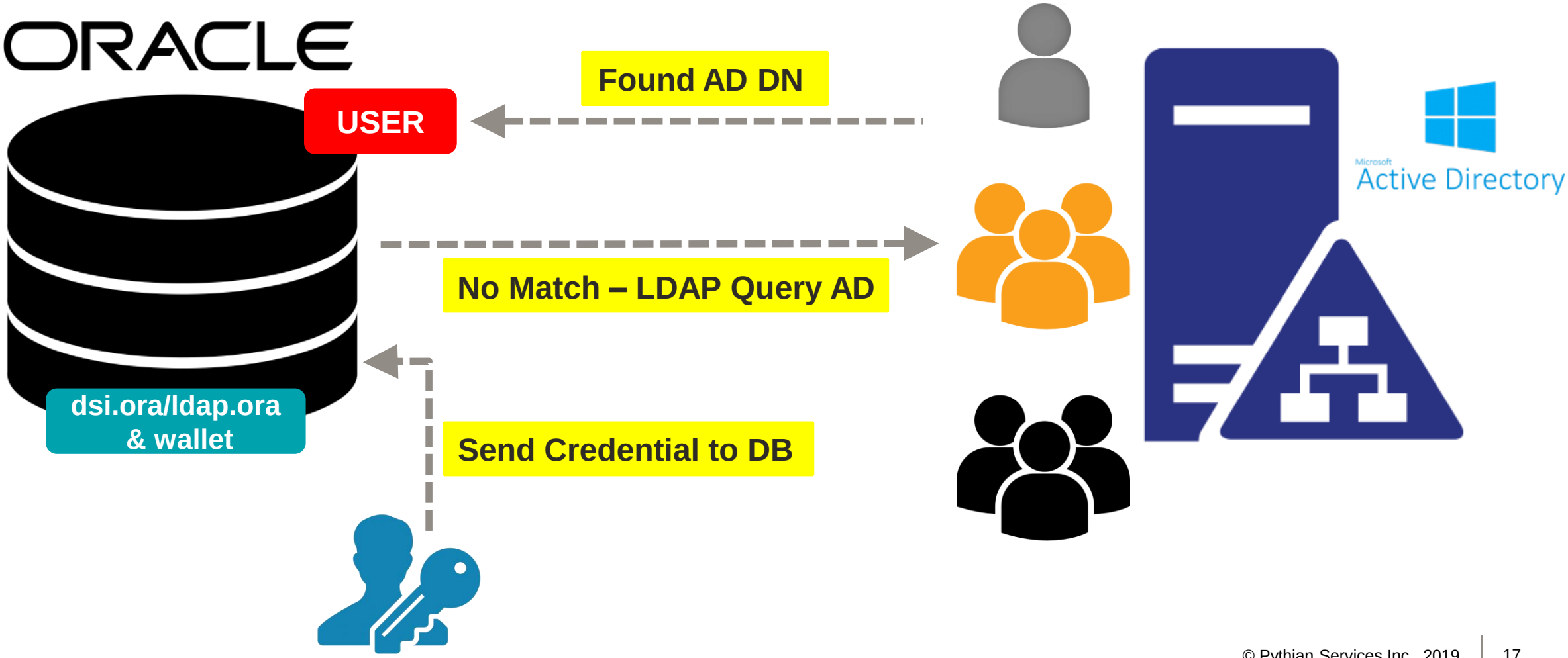

Now we really have the same options with Oracle Database 18c+

# "… IDENTIFIED GLOBALLY AS …"




*NOTE: Oracle User/Role names don't need to match AD User/Group names*

# Logical Connection Flow



# Authentication and Authorisation Options

- Oracle Database 18c provides several AD authentication options:
  1. Password ← Compatibility
  2. Kerberos ← Recommended
  3. SSL Certificate (PKI)

- Oracle Database 18c provides several AD authorisation options:
  - Normal Oracle Database built-in technologies (roles, privileges, etc.)
  - Active Directory Security Groups

# To Put it Simply

- Using the "**Password**" configuration option:

**PASSWORD**

  - Database connections still require credentials (username & password)
  - Password is validated against Active Directory instead of the database
  - Essentially "re-prompting" – compromised desktop != DB access

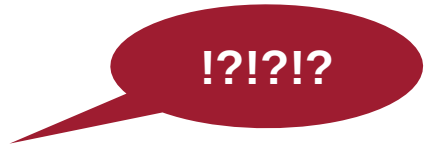
- Using the "**Kerberos**" configuration option:

**KERBEROS**

  - Active Directory issues Kerberos "tickets" (TGT)
  - Tickets are used for authentication – no credential (no username or password) required for DB connections

PASSWORD

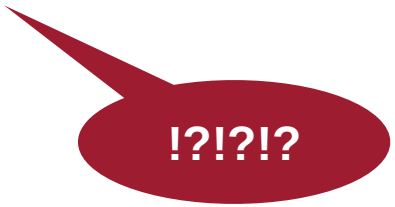
# Summary of Implementation Steps




1. Extend the AD Schema and install the Oracle “Password Verifiers”
2. Create an “Oracle Service Directory User” (for DB <-> AD communication)
3. Configure the RDBMS home to integrate with AD via the Service Directory User and the AD’s “Public Certificate”
4. Create “`... IDENTIFIED GLOBALLY ...`” database users and/or roles

# Summary of Implementation Steps

1. Create a “service principal” for the DB server in Active Directory
2. Extract the “key table” for the “service principal” and copy to the DB server
3. Configure Kerberos settings and SQLNET.ORA on DB server
4. Create “`... IDENTIFIED EXTERNALLY ...`” database users and/or groups
5. Configure client Kerberos settings and SQLNET.ORA



***Explaining "Password" authentication going forward as it seems to usually be the most applicable***

***But paper explaining "Kerberos" setup is available upon request***

# Active Directory Implementation Steps

***This might seem a little complicated at first but really is not.  And is only a one-time setup!***

# Test Environment Summary – OCI Based

- Oracle Linux 7.7 database server with Oracle 18c **XE** RDBMS home:
  - Using default locations for certain files such as `dsi.ora` and Oracle Wallet
- XE means a CDB database with one pluggable database **XEPDB1**:
  - CMU works fine with PDB or non-CDB database
- One Windows 2016 Standard Edition Domain Controller (DC)

# Prerequisites

- An Active Directory (AD) forest and domain controller (DC)
  - Administrative access to the DC – AD schema will be extended
- Easy to setup your own PoC / test lab using a cloud environment (OCI):
  - Provision new Windows 2016 Server (Standard edition on VM will suffice)
  - Install and configure Active Directory Domain Services
  - Install and configure Active Directory Certificate Services

**Easy to follow step-by-step instructions**

- **Step-by-step blog series (for DBAs to implement) coming out soon.**

**Easy to setup a complete test environment in OCI or on-prem VM**

# Overview of the Active Directory Setup

1. Create an Oracle "*Service Directory User*" (in AD)
    - Is the credential that the database software will use to interact with (query) AD
2. Install the Oracle "*Password Filter*" into each Domain Controller (DC)
    - Will allow AD to capture a password hash compatible with Oracle queries
3. Extract the DC Public Certificate for the Oracle DB to connect with
4. Create AD groups (and optionally new users)
5. Configure the Database users based on the AD "*Distinguished Name*" (DN)

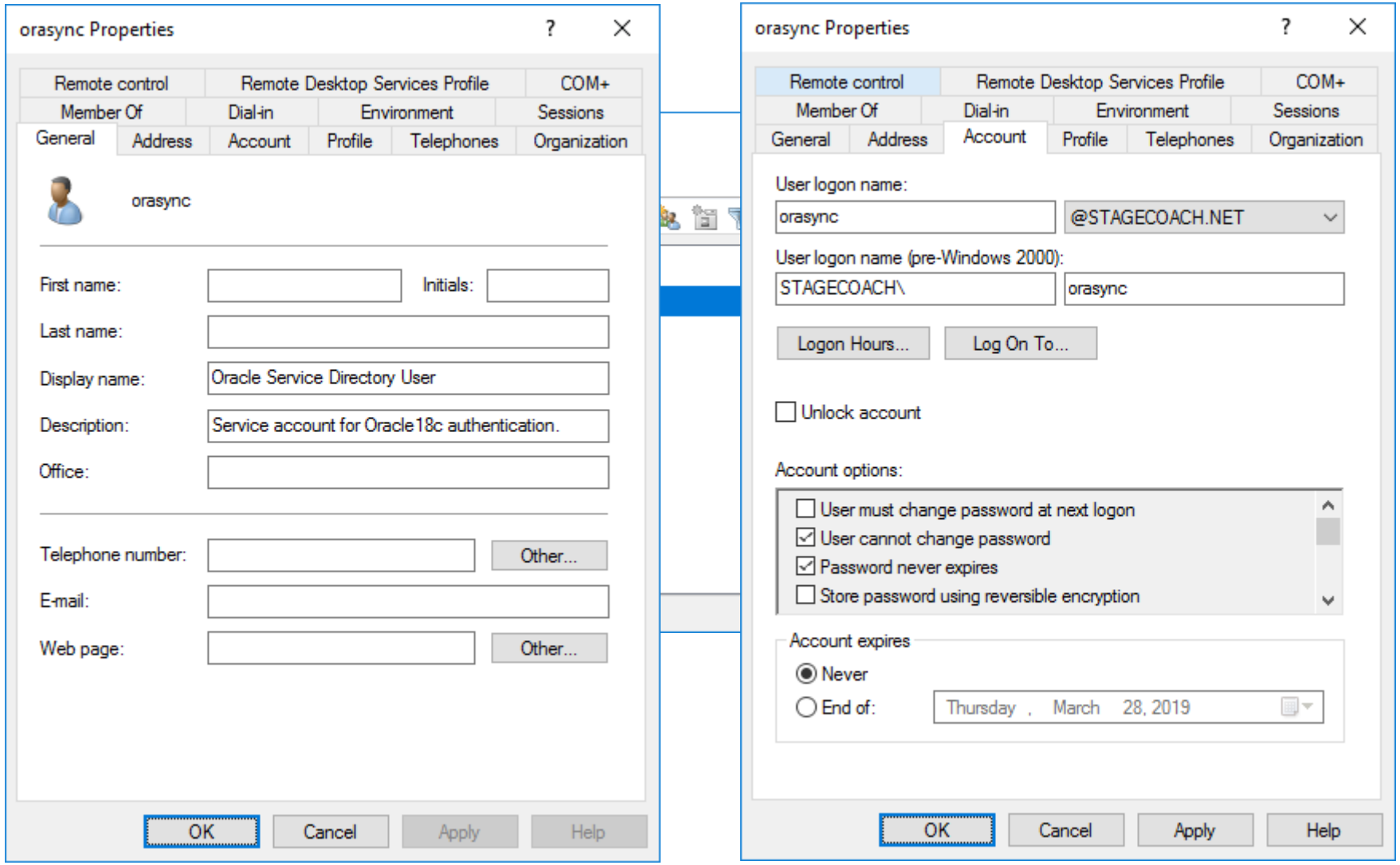
# Creating the Oracle Service Directory User

- An AD user that the Oracle Database software will use for AD interaction
- Sample Windows PowerShell script:

```
New-ADUser `
    -Name = "orasync" `
    -UserPrincipalName = "orasync@stagecoach.net" `
    -DisplayName = "Oracle Service Directory User" `
    -Description = "Service account for Oracle Database authentication." `
    -Path = "CN=Managed Service Accounts,DC=stagecoach,DC=net" `
    -ChangePasswordAtLogon = $false `
    -PasswordNeverExpires = $true `
    -CannotChangePassword = $true `
    -Enabled = $true `
    -AccountPassword(Read-Host -AsSecureString "Initial Password:")
```

**A Personal Preference:**
- **Alternatively use “Users”**
- **This path is referenced in later commands**

# The Oracle Service Directory User



# Permissions for the Oracle Service Directory User

- Not very clear in the official documentation
- Actual AD implementation steps:

  All tasks -> Delegate Control

  Select the Oracle Services Directory User

  Choose the "**Create a custom task to delegate**" radio box

  Select the "**Only the following objects in the folder**" radio box, then the "**User objects**" check-box

  Choose both the "**General**" and "**Property specific**" check-boxes

  Select the "**Read**" and "**Write lockout Time**" permissions.

- Or from Windows PowerShell:

```
dsacls "CN=orasync,CN=Managed Service Accounts,DC=STAGECOACH,DC=NET" /I:P /G "STAGECOACH\orasync:WP;lockoutTime"
dsacls "CN=orasync,CN=Managed Service Accounts,DC=STAGECOACH,DC=NET" /I:P /G "STAGECOACH\orasync:RP"
```

# Copy the Password Filter Installer to the DC

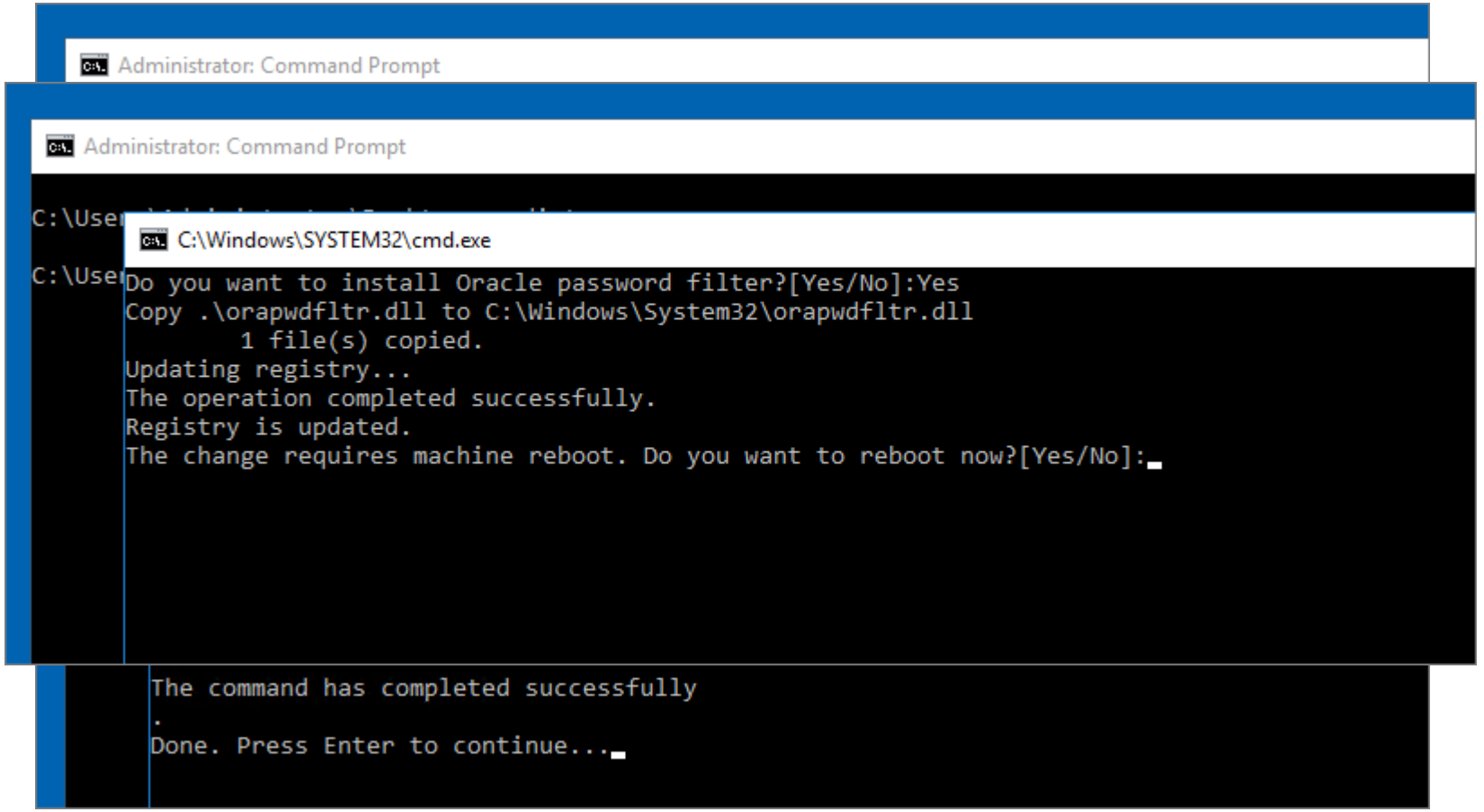
- Extends the Active Directory Schema:
  - Adds the "`orclCommonAttribute`" for user accounts
- Creates three new AD groups that will use the password filter
- Must install on every DC (reboot required)
- Copy the `${ORACLE_HOME}/bin/opwdintg.exe` file from an RDBMS home
  - Must be an Oracle18c+ home
  - Can be copied from a Linux home (same endian)

Remember: Not required with Kerberos!

```
$ ls -lh ${ORACLE_HOME}/bin/*.exe
-rw-r--r--. 1 oracle oinstall 183K Feb  7  2018 /u01/app/oracle/product/18.0.0/dbhome_1/bin/opwdintg.exe
$
```

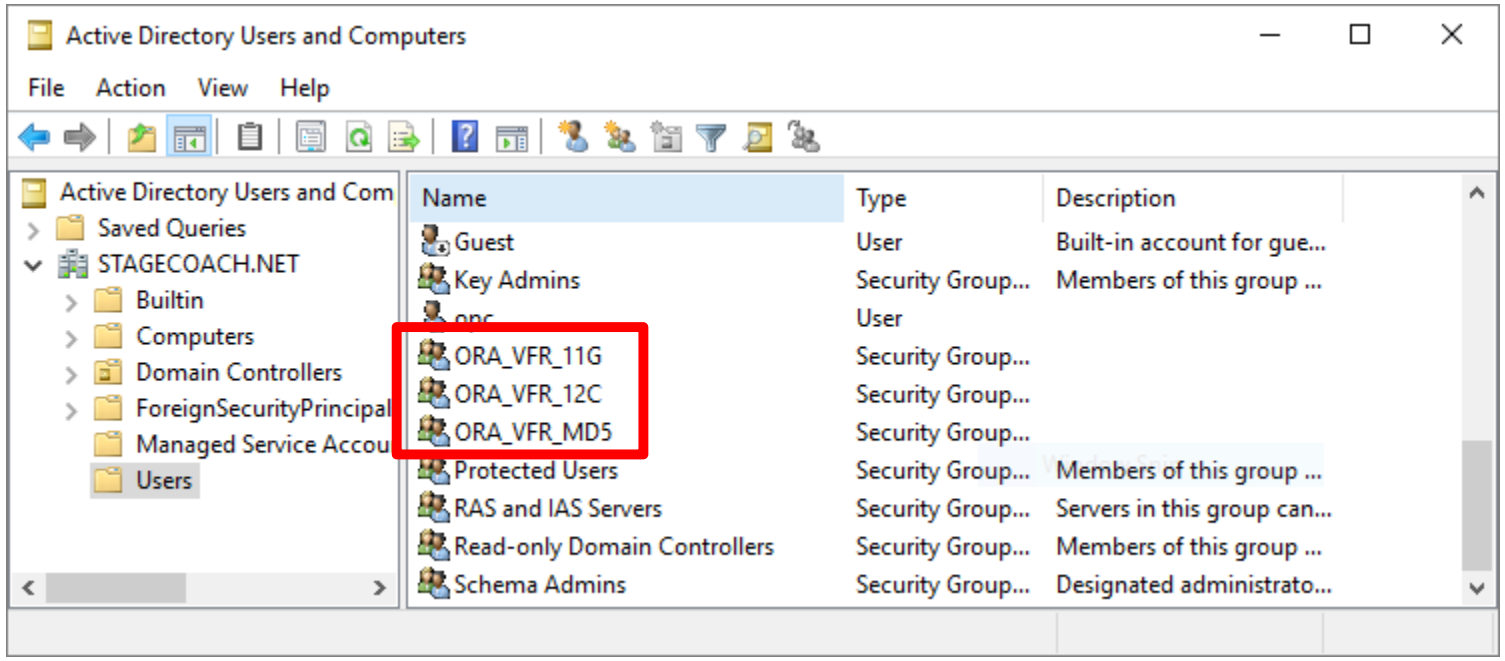
# Install the Password Filter into AD

- IMPORTANT: a Domain Controller reboot is required!



# The Result: New AD Groups

- Three new AD groups for the Oracle Database 11g password verifier, 12c password verifier, and WebDAV client:



# A Quick Test/Verification

- LDAP utilities have been in RDMS homes for many releases

```
${ORACLE_HOME}/bin/ldapsearch \
   -b "DC=stagecoach,DC=net" \
   -D "CN=orasync,CN=Managed Service Accounts,DC=stagecoach,DC=net" \
   -h 10.0.0.12 -p 389 -q \
   "cn=orasync" description

${ORACLE_HOME}/bin/ldapbind \
   -D "CN=orasync,CN=Managed Service Accounts,DC=stagecoach,DC=net"
   -h 10.0.0.12 -p 389 -q

${ORACLE_HOME}/bin/ldapbind \
   -b "DC=stagecoach,DC=net" \
   -D "CN=orasync,CN=Managed Service Accounts,DC=stagecoach,DC=net" \
   -h 10.0.0.12 -p 636 -q -U3 \
   -W "file:/u01/app/oracle/admin/PRD01/wallet" -Q \
   "cn=orasync" description
```

**To test two-way authentication using certificate**

**Location of Oracle Wallet containing cert**

# Export the Server's Public Certificate

- From the GUI or a PowerShell cmdlet:

```
# Extract details of the Server's self-issued certificate:
$Cert = Get-ChildItem Cert:\LocalMachine\My | `
   Where-Object {$_.subject -match [Environment]::GetEnvironmentVariable("computername")+"."
   +[Environment]::GetEnvironmentVariable("userdnsdomain")}

# Export the certificate to a .cer file
Export-Certificate -Cert $Cert -FilePath .\$Env:computername.cer -Type CERT -Force
```

- Recommend the above command over the `certsrv.msc` (GUI) for reliability
- Manually copy the exported public certificate to the database server

# Database Home Configuration

# Specifying the Active Directory Servers

- List AD servers in a `dsi.ora` file (use of an `ldap.ora` is not recommended)

```
cat <<EOT > ${ORACLE_HOME}/ldap/admin/dsi.ora
DSI_DIRECTORY_SERVERS = (10.0.0.12:389:636)
DSI_DEFAULT_ADMIN_CONTEXT = "DC=stagecoach,DC=net"
DSI_DIRECTORY_SERVER_TYPE = AD
EOT

cat ${ORACLE_HOME}/ldap/admin/dsi.ora
```

**Optional**

**Can use hostname or FQDN and list multiple DCs**

# Create a Wallet File

- To hold the "*Service Directory User's*" credential and the certificate

```
mkdir -p ${ORACLE_BASE}/admin/${ORACLE_SID}/wallet
cd ${ORACLE_BASE}/admin/${ORACLE_SID}/wallet

orapki wallet create -wallet . -auto_login
mkstore -wrl . -createEntry ORACLE.SECURITY.USERNAME orasync
mkstore -wrl . -createEntry ORACLE.SECURITY.DN "CN=orasync,CN=Managed Service Accounts,DC=STAGECOACH,DC=NET"
mkstore -wrl . -createEntry ORACLE.SECURITY.PASSWORD Welcome1
orapki wallet add -wallet . -cert Active_Directory.cer -trusted_cert

orapki wallet display -wallet .
mkstore -wrl . -viewEntry ORACLE.SECURITY.DN -viewEntry ORACLE.SECURITY.PASSWORD -viewEntry ORACLE.SECURITY.USERNAME
```

# Database Configuration

- Initialisation Parameter adjustments:

```
SQL> ALTER SYSTEM SET ldap_directory_access='PASSWORD' SCOPE=both;

System altered.

SQL> ALTER SYSTEM SET ldap_directory_sysauth=YES SCOPE=spfile   /* Optional */ ;

System altered.

SQL> !orapwd file=${ORACLE_HOME}/bin/orapwd${ORACLE_SID} format=12.2   # Optional #

Enter password for SYS:

SQL>
```

- Instance restart required for the optional `ldap_directory_sysauth` change

# Database User / Role Configuration

# Database Catalog Differences

- Normal database authenticated users – DB stored credentials:

```
USERNAME PASSWORD_HASH                                  AUTH_TYP EXTERNAL_NAME
-------- ---------------------------------------------- -------- ----------------------------------------
SCOTT    S:ABB999A0C4672B5A5E5DF1628DC8D1BC0AB398 PASSWORD
         AF4A8272E69947F97BE5B4;T:D08D189CDB2B553
         FF85185625B14ECCAD362ACA5F9BA807D343E13D
         DAE583A7FB5EE5777228206AB20F0A8E0450A465
         1B82225DCF92BF12ACAA54B275E42FFB008E4EB5
         196572B53C0221B76B86FA258
```

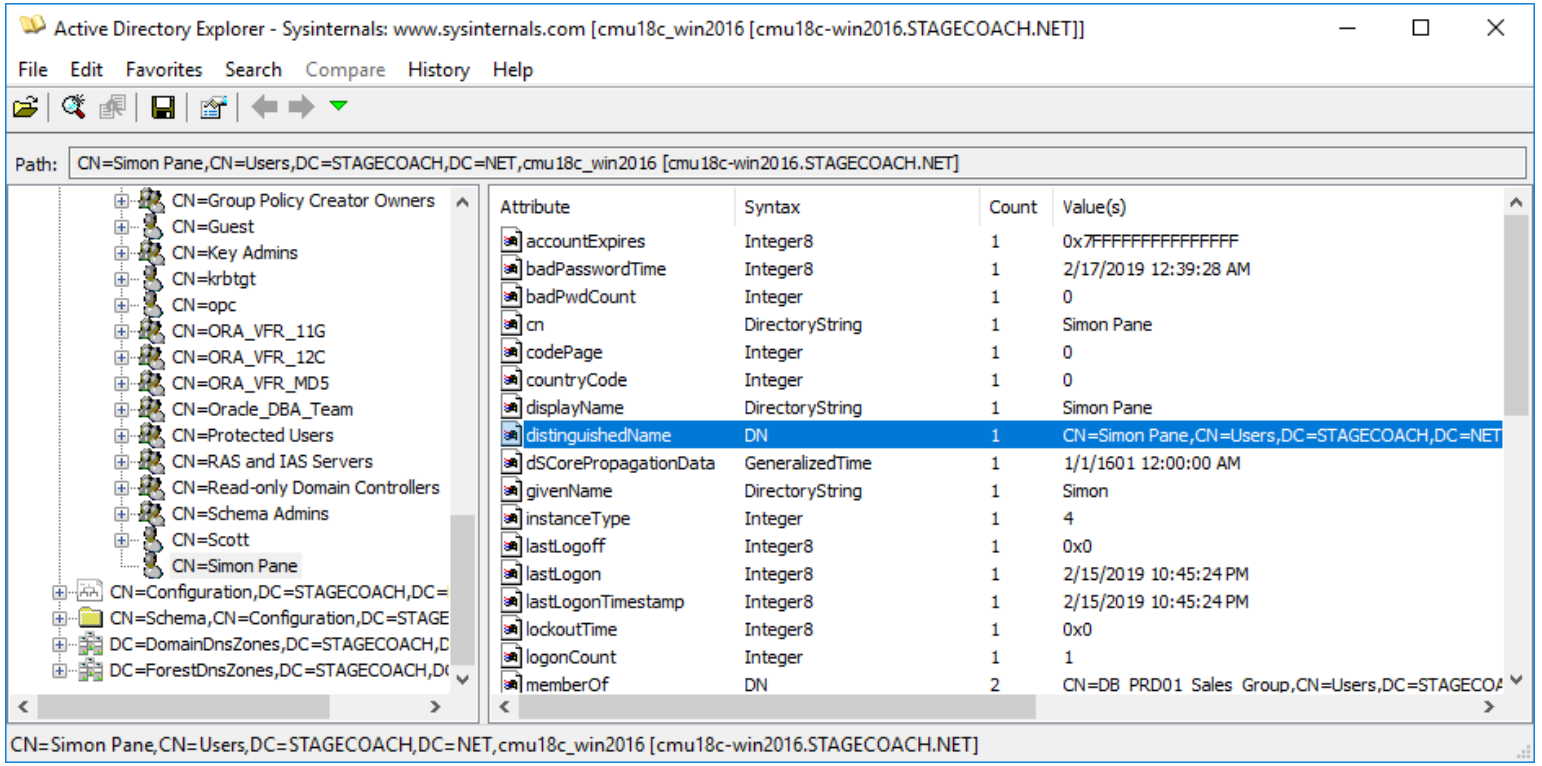
- New Active Directory authenticated users – AD stored credentials:

```
USERNAME PASSWORD_HASH                                  AUTH_TYP EXTERNAL_NAME
-------- ---------------------------------------------- -------- ----------------------------------------
SCOTT                                                   GLOBAL   cn=Scott,cn=Users,dc=stagecoach,dc=net
```

**Active Directory "Distinguished Name"**

# Recommended AD Query Tool

- AD Explorer - Windows Sysinternals
  - Single executable utility
  - Useful for obtaining the user's "Distinguished Name" and checking the "`orclCommonAttribute`"



# Command Line Alternatives

- Command shell example:

```
dsquery user -name simon -o dn
```

- PowerShell example:

```
Get-ADUser -Identity "simon" -properties DistinguishedName,orclCommonAttribute
```

```
PS C:\Users\Administrator> Get-ADUser -Identity "simon" -properties DistinguishedName,orclCommonAttribute

DistinguishedName    : CN=Simon Pane,CN=Users,DC=STAGECOACH,DC=NET
Enabled              : True
GivenName            : Simon
Name                 : Simon Pane
ObjectClass          : user
ObjectGUID           : 76438e16-125c-4dff-a153-d8628905e6d5
orclCommonAttribute  : {MR-SHA512}v8oLSV8JuPW8jhjTv2cTcubULP6+yvbeZqZyqwrPpKAqkyx+wXnE8hEKU5kovMDXEudfN8+XZy1A4aTDrdAxek
                       S9yCjOMwq7B1P4yI2S7k4=
SamAccountName       : simon
SID                  : S-1-5-21-2551431580-742512773-3804340073-1112
Surname              : Pane
UserPrincipalName    : simon@STAGECOACH.NET
```

# Create Users and Roles

- Use "`... INDENTIFIED GLOBALLY AS ...`"
- Obtain the "Distinguished Names" from Active Directory

```
CREATE USER ad_simon_pane IDENTIFIED GLOBALLY AS 'cn=Simon Pane,cn=Users,dc=STAGECOACH,dc=NET';
CREATE USER ad_dba_team   IDENTIFIED GLOBALLY AS 'cn=Oracle_DBA_Team,cn=Users,dc=STAGECOACH,dc=NET';
CREATE ROLE ad_sales_role IDENTIFIED GLOBALLY AS 'cn=DB_PRD01_Sales_Group,cn=Users,dc=STAGECOACH,dc=NET';
```

- Existing database users can also be migrated via "`ALTER USER ... IDENTIFIED GLOBALLY AS ...`"
- Administrative users and connections are also supported

# Connection Options

- Can perform the database connection using:
  1. Using the "**down-level logon name**" (or "SAMAccountName", "pre-Windows 2000 logon name") : `DOMAIN\User`
  2. Using the "**User Principal Name**" (or "UPN"): `User@Domain`
  3. Just using the "**User Login Name**" : `User`
- Local BEQ and TNS connections supported
- Examples:

```
SQL> connect "STAGECOACH\simon"@ORCL

SQL> connect "simon@stagecoach.net"@ORCL

SQL> connect simon
```

# Other Suggestions

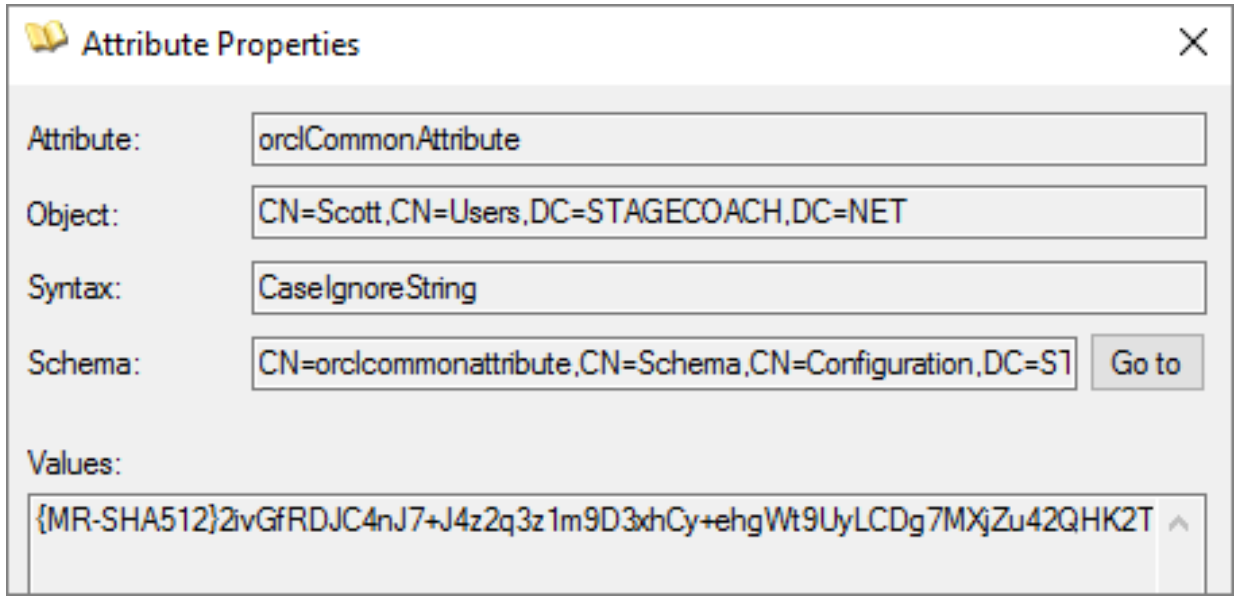
- Use a good nomenclature to make AD users/groups easily identifiable:

```
USERNAME                       AUTHENTI EXTERNAL_NAME                                          PASSWORD
------------------------------ -------- ------------------------------------------------------ ---------
AD_ORACLE_DBA_TEAM             GLOBAL   cn=Oracle_DBA_Team,cn=Users,dc=STAGECOACH,dc=NET       GLOBAL
AD_PRD01_ADMINS                GLOBAL   cn=DB_PRD01_Admins,cn=Users,dc=STAGECOACH,dc=NET       GLOBAL
AD_PRD01_USERS                 GLOBAL   cn=DB_PRD01_Users,cn=Users,dc=STAGECOACH,dc=NET        GLOBAL
AD_SCOTT                       GLOBAL   cn=Scott,cn=Users,dc=stagecoach,dc=net                 GLOBAL
AD_SIMON                       GLOBAL   cn=Simon Pane,cn=Users,dc=stagecoach,dc=net            GLOBAL
```

What appears in **`v$session`** and other views

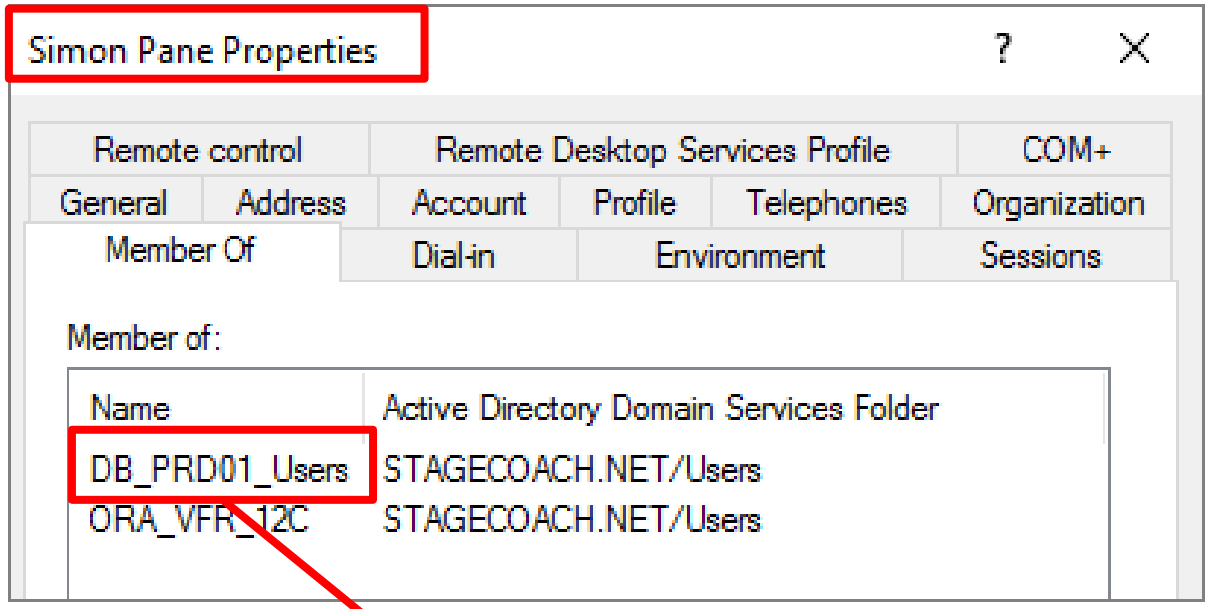
# Must Change AD Password After Creation

- Because the AD “Password Verifier” groups are assigned ***after*** creation
- Need to be part of the verifier groups to store hash in `orclCommonAttribute`
- Users usually have to change their password on first (Windows) login anyway




Reminder: Not required with Kerberos!

# Example Shared Schema Configuration




```
USERNAME                       AUTHENTI EXTERNAL_NAME                                    PASSWORD
------------------------------ -------- -------------------------------------------------- ---------
AD_PRD01_USERS                 GLOBAL   cn=DB_PRD01_Users,cn=Users,dc=STAGECOACH,dc=NET   GLOBAL
```

# The Typical Session Properties

- Only shows the Shared Schema details:

```
SQL> connect "simon@stagecoach.net"@PRD01
Enter password:
Connected.
SQL>
SQL> SELECT SYS_CONTEXT('USERENV','SESSION_USER')   AS session_user,
  2         SYS_CONTEXT('USERENV','SESSION_SCHEMA') AS session_schema,
  3         SYS_CONTEXT('USERENV','CURRENT_USER')   AS current_user,
  4         SYS_CONTEXT('USERENV','CURRENT_SCHEMA') AS current_schema,
  5         user
  6    FROM dual;

SESSION_USER         SESSION_SCHEMA       CURRENT_USER         CURRENT_SCHEMA       USER
-------------------- -------------------- -------------------- -------------------- --------------
AD_PRD01_USERS       AD_PRD01_USERS       AD_PRD01_USERS       AD_PRD01_USERS       AD_PRD01_USERS

SQL>
```

# Authentication and Identity Properties

- Does show all of the pertinent information:

```
SQL> connect "simon@stagecoach.net"
Enter password:
Connected.
SQL>
SQL> SELECT SYS_CONTEXT('USERENV','AUTHENTICATED_IDENTITY') AS authenticated_identity,
  2         SYS_CONTEXT('USERENV','AUTHENTICATION_METHOD')  AS authentication_method,
  3         SYS_CONTEXT('USERENV','IDENTIFICATION_TYPE')    AS identification_type,
  4         SYS_CONTEXT('USERENV','LDAP_SERVER_TYPE')       AS ldap_server_type,
  5         SYS_CONTEXT('USERENV','ENTERPRISE_IDENTITY')    AS enterprise_identity
  6    FROM dual;

AUTHENTICATED_IDENTITY AUTHENTICATION_METHOD IDENTIFICATION_TYPE LDAP_SERVER_TYPE ENTERPRISE_IDENTITY
---------------------- --------------------- -------------------- ---------------- ------------------------
simon@stagecoach.net   PASSWORD_GLOBAL       GLOBAL SHARED        AD               cn=Simon Pane,cn=Users,
                                                                                   dc=STAGECOACH,dc=NET
```

# A Simple Auditing Test

- Audit `create session` and connect using a Shared Schema:

```
SQL> audit create session;

Audit succeeded.

SQL> connect "STAGECOACH\simon"@PRD01
Enter password:
Connected.
SQL>
SQL> connect "simon@stagecoach.net"@PRD01
Enter password:
Connected.
SQL>
```

# Audit Records

```
SQL> SELECT username, extended_timestamp, comment_text FROM dba_audit_trail ORDER BY 1,2;

USERNAME       EXTENDED_TIMESTAMP                  COMMENT_TEXT
-------------- ----------------------------------- ------------------------------------------------------------
AD_PRD01_USERS 16-FEB-19 04.14.38.796356 PM -07:00 Authenticated by: DIRECTORY PASSWORD;EXTERNAL NAME: cn=Simon
                                                   Pane,cn=Users,dc=STAGECOACH,dc=NET;AUTHENTICATED IDENTITY:
                                                   STAGECOACH\simon; Client address:
                                                   (ADDRESS=(PROTOCOL=tcp)(HOST=10.0.0.13)(PORT=36076))

AD_PRD01_USERS 16-FEB-19 04.14.38.804294 PM -07:00
AD_PRD01_USERS 16-FEB-19 04.14.38.921909 PM -07:00 Authenticated by: DIRECTORY PASSWORD;EXTERNAL NAME: cn=Simon
                                                   Pane,cn=Users,dc=STAGECOACH,dc=NET;AUTHENTICATED IDENTITY:
                                                   simon@stagecoach.net; Client address:
                                                   (ADDRESS=(PROTOCOL=tcp)(HOST=10.0.0.13)(PORT=36080))

AD_PRD01_USERS 16-FEB-19 04.14.42.656294 PM -07:00

SQL>
```

# Both Exclusive and Shared Schema Matches?

- Connects as the Exclusive User over the Shared Schema

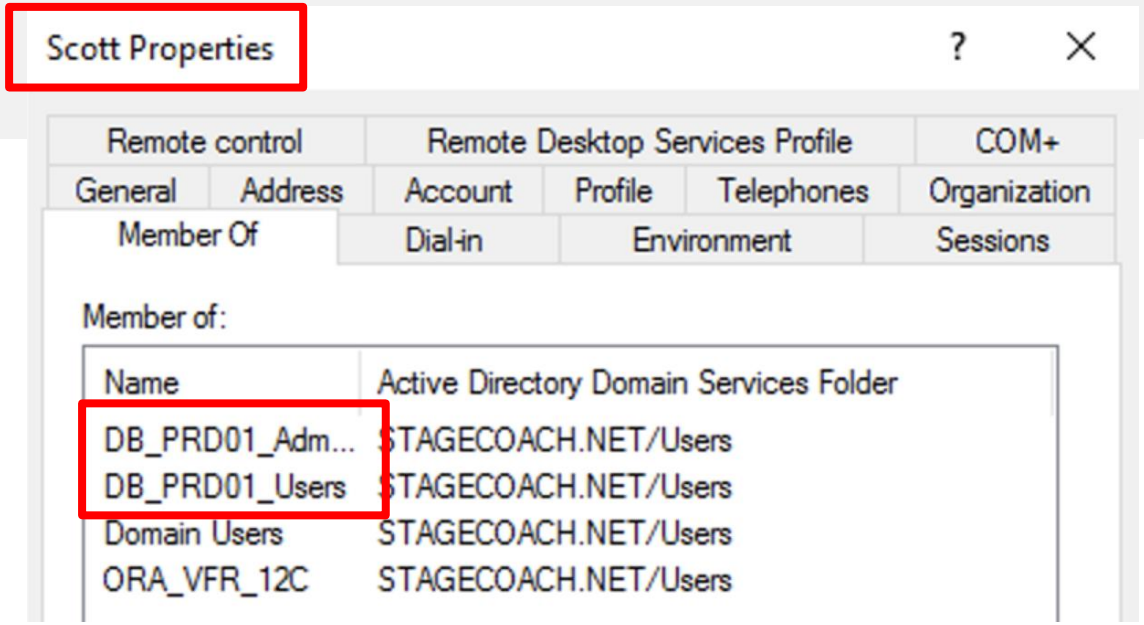
```
USERNAME                       AUTHENTI EXTERNAL_NAME                                          PASSWORD
------------------------------ -------- ------------------------------------------------------ ----------
AD_ORACLE_DBA_TEAM             GLOBAL   cn=Oracle_DBA_Team,cn=Users,dc=STAGECOACH,dc=NET GLOBAL
AD_SIMON                       GLOBAL   cn=Simon Pane,cn=Users,dc=stagecoach,dc=net      GLOBAL
```

```
SQL> connect "simon@stagecoach.net"@PRD01
Enter password:
Connected.
SQL>
SQL> show user
```

# Member of Multiple AD Groups?

**TIP: Don't Do!**

```
SQL> connect "scott@stagecoach.net"@PRD01
Enter password:
ERROR:
ORA-28306: The directory user has 2 groups mapped to different database global users.


Connected.
SQL>
```




- Only connects as the one Shared Schema based on lowest `USER_ID`

# Global Roles

- Create a Database Role that maps to an AD group:

```
CREATE ROLE ad_sales_role IDENTIFIED GLOBALLY AS
'CN=DB_PRD01_Sales_Group,CN=Users,DC=STAGECOACH,DC=NET';
```

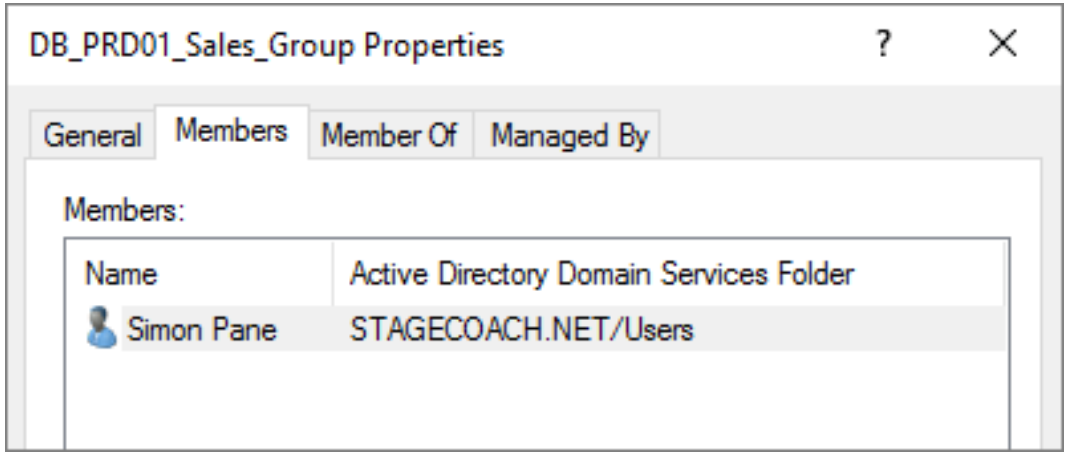
```
ROLE           AUTHENTICAT PASSWORD EXTERNAL_NAME
-------------- ----------- -------- ------------------------------------------
AD_SALES_ROLE  GLOBAL      GLOBAL   cn=DB_PRD01_Sales_Group,cn=Users,dc=STAGECOACH,dc=NET
```

# Global Role Membership

- Can’t grant in the DB – membership assigned through AD group:

```
SQL> GRANT ad_sales_role TO clark;
GRANT ad_sales_role TO clark
*
ERROR at line 1:
ORA-28021: cannot grant global roles
```

- Effectively grant through AD Group membership



# Global Roles – Activated When Connected

- After connecting:

```
SQL> connect "simon@stagecoach.net"
Enter password:
Connected.
SQL>
SQL> SELECT role FROM session_roles ORDER BY 1;

ROLE
--------------
AD_SALES_ROLE
CONNECT

SQL>
```

# Issues and Troubleshooting

# What does ORA-01017 Actually Mean?

- Error **`ORA-01017`** is commonly returned due to a wide variety of causes

```
ERROR:
ORA-01017: invalid username/password; logon denied


SQL> !oerr ora 01017
01017, 00000, "invalid username/password; logon denied"
// *Cause:
// *Action:
```

- Really means: **could not validate that the credential is valid**:
  - Bad Password
  - DC unreachable (due to setup, networking, routing, permissions, or server down)

# First Check the Obvious: Verify the Password

- Test AD user password:

```
C:\>runas /u:simon@stagecoach.net notepad.exe
Enter the password for simon@stagecoach.net:
Attempting to start notepad.exe as user "simon@stagecoach.net" ...
RUNAS ERROR: Unable to run - notepad.exe
1385: Logon failure: the user has not been granted the requested logon type at this computer.

C:\>
```

- AD user may have:
  - Expired password
  - Locked account due to failed login attempts

**The Oracle Database honors these**

# Verify the Connection: Test Using an LDAP Query

```
$ ${ORACLE_HOME}/bin/ldapbind \
>    -D "CN=orasync,CN=Managed Service Accounts,DC=stagecoach,DC=net" \
>    -h 10.0.0.12 -p 389 -q
Please enter bind password:
bind successful
$
```

```
$ ${ORACLE_HOME}/bin/ldapsearch \
>    -b "DC=stagecoach,DC=net" \
>    -D "CN=orasync,CN=Managed Service Accounts,DC=stagecoach,DC=net" \
>    -h 10.0.0.12 -p 389 -q "cn=orasync" description
Please enter bind password:
CN=orasync,CN=Managed Service Accounts,DC=STAGECOACH,DC=NET
description=Service account for Oracle18c authentication.
```

- Check firewalls
  - At the network level, the DB server level, and Domain Controller level
  - ICMP (ping) tests

# Firewall Rules – Common LDAP Ports Required

| Stateful Rules | | | | |
|---|---|---|---|---|
| **Source:**<br>10.0.0.0/24 | **IP Protocol:**<br>TCP | **Source Port Range:**<br>All | **Destination Port Range:**<br>389 | **Allows:** TCP traffic for ports: 389 Lightweight Directory Access Protocol (LDAP) |
| **Source:**<br>10.0.0.0/24 | **IP Protocol:**<br>TCP | **Source Port Range:**<br>All | **Destination Port Range:**<br>636 | **Allows:** TCP traffic for ports: 636 |

# Active Directory Policies and Passwords

- Oracle DB prevents connections when the AD status is:
  - “password expired”
  - “password must change”
  - “account locked out”
  - “account disabled”
- Remember to change the AD password **after** adding the user to the Oracle password verifier group(s) in AD

# Connection Tracing

- Additional details can be obtained using tracing:

```
alter system set events='trace[gdsi] disk low';
                                     off
```

- Then review the resulting trace file in the ADR:

```
[28994890]kzlg discovered server type: AD
[28994890]kzlg AD user name: STAGECOACH\simon
[28994890]kzlg found dn in wallet
[28994890]kzlg found pwd in wallet
[28994890]kzlg found usr in wallet
[28994890]kzlg discovered ldaptype: AD
[28994890]kzlg ldap_open 10.0.0.12:636
[28994890]kzlg DB-LDAP init SSL succeeded.
...
```

# Lacking Critical Detail in Oracle Return Codes

```
SQL> connect "simon@stagecoach.net"@PRD01
Enter password:
ERROR:
ORA-01017: invalid username/password; logon denied


Warning: You are no longer connected to ORACLE.
SQL>
```

- Within Active Directory (and associated trace file messages)
  - "User must change password at next logon": `kzlg polerr=28223`
  - "Account disabled": `kzlg polerr=28052`
  - "Password incorrect": `kzlg polerr=0 ; KZLG_ERR: LDAPERR=49, OER=28043`
  - Cannot contact AD DC: `KZLG_ERR: 28030 from kzlgOpenBind`

# User “locked” in Active Directory

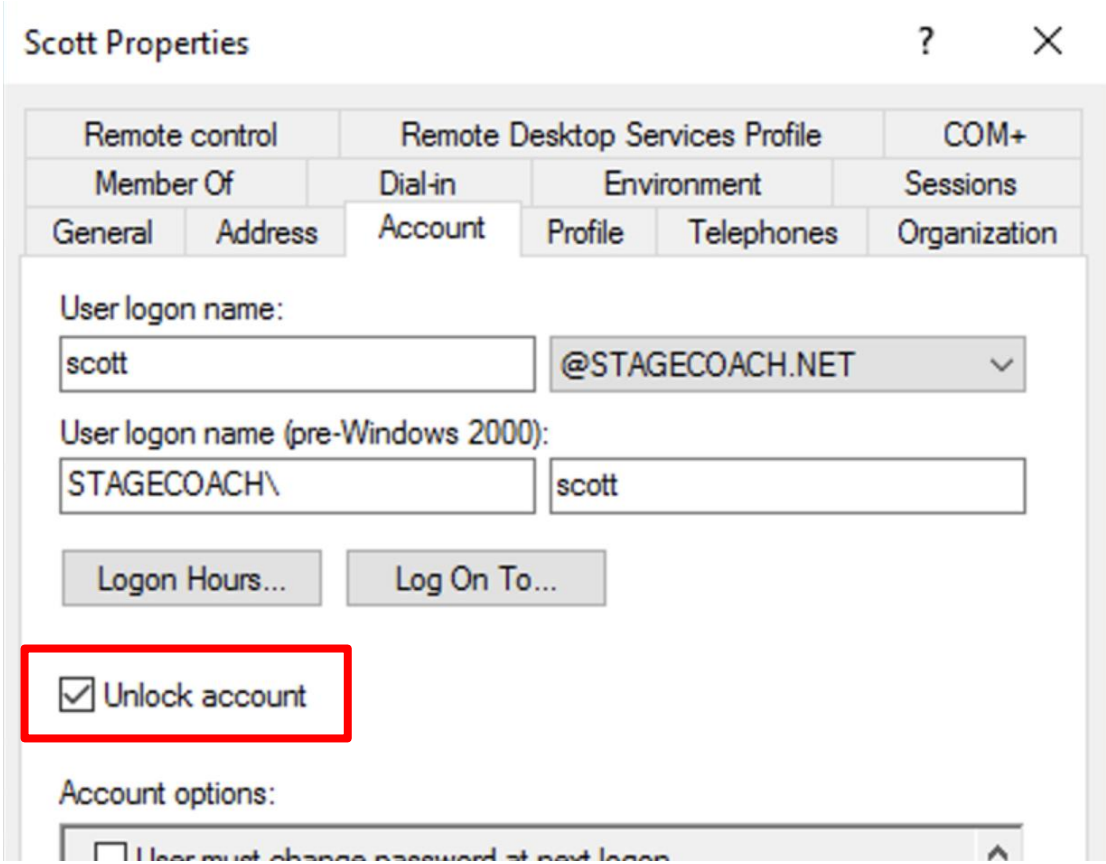
```
SQL> connect "simon@stagecoach.net"@PRD01
Enter password:
ERROR:
ORA-28300: No permission to read user entry in LDAP directory service.


Warning: You are no longer connected to ORACLE.
SQL>
```

```
[28994890]KZLG_ERR: failed to modify user status Insufficient access
[28994890]KZLG_ERR: LDAPERR=50, OER=28300
```

- Usually associated with new AD accounts that have never logged into Windows
- Unlock within AD

# User “locked” in Active Directory

- Is a group policy causing them to lock too easily?



# Difficult Problem: LDAP Bind Errors

- SYMPTOM / ERROR from SQLPlus:
  - `ORA-01017: invalid username/password; logon denied`
- SYMPTOM / ERROR from alert log:
  - `ORA-28043: invalid bind credentials for DB-OID connection`
- SYMPTOM / ERROR from trace file:
  - `KZLG_ERR: failed to sasl bind to LDAP server. err=49`

# Trace File from Network Issue

```
ERROR:
ORA-01017: invalid username/password; logon denied


Warning: You are no longer connected to ORACLE.
$ grep -ih kzlg *.trc
[28994890]kzlg AD user name: STAGECOACH\simon
[28994890]kzlg found dn in wallet
[28994890]kzlg found pwd in wallet
[28994890]kzlg found usr in wallet
[28994890]kzlg found domain STAGECOACH; dc=STAGECOACH,dc=NET; 1 dirsrv
[28994890]kzlg ldap_open 10.0.0.12:636
[28994890]kzlg DB-LDAP init SSL succeeded.
[28994890]KZLG_ERR: failed to sasl bind to LDAP server. err=49
[28994890]KZLG_ERR: ldap_bind_s on SSL failed. err=49
[28994890]KZLG_ERR: LDAPERR=49, OER=28043
[28994890]KZLG_ERR: ldap_bind err=28043
[28994890]kzlg doing LDAP unbind
[28994890]KZLG_ERR: 28043 from kzlgOpenBind.
[28994890]KZLG_ERR: failed to connect to ldap
```

# Two Possible Causes

1. Check that the Oracle Directory User's credentials in the wallet are valid:

```
orapki wallet display -wallet .

mkstore -wrl . -viewEntry ORACLE.SECURITY.DN

mkstore -wrl . -viewEntry ORACLE.SECURITY.USERNAME

mkstore -wrl . -viewEntry ORACLE.SECURITY.PASSWORD
```

2. Networking resolution / firewall / routing
   - Check resolution in DNS server or in local `/etc/hosts` file as a workaround if needed

# Networking Solution

- On Domain controller determine the internal (private) IP, hostname, and FQDN. From Windows Command Prompt:

```
hostname

hostname | nslookup
```

- On DB Server ensure that LDAP port 636 can be reached for the IP, hostname, and FQDN (output from all three above):

```
(echo > /dev/tcp/10.0.0.12/636) >/dev/null 2>&1 && echo "OPEN" || echo "CLOSED"

(echo > /dev/tcp/DC2/636) >/dev/null 2>&1 && echo "OPEN" || echo "CLOSED"

(echo > /dev/tcp/DC2.STAGECOACH.net/636) >/dev/null 2>&1 && echo "OPEN" || echo "CLOSED"
```

**Testing:**
**1) Private network IP**
**2) Hostname**
**3) FQDN**

# Authorisation is Still Database Based

```
SQL> connect "scott@stagecoach.net"@PRD01
Enter password:
ERROR:
ORA-01045: user AD_SCOTT lacks CREATE SESSION privilege; logon denied


SQL>
```

- Still need to setup grants, roles, etc within the database via a normal role, global role or direct grant
- Granting to either the Exclusive User or Shared Schema

# A Few More Experienced Errors

```
ORA-12638: Credential retrieval failed

ORA-12641: Authentication service failed to initialize
```

- Usually related to **SQLNET.ORA**, specifically **SQLNET.AUTHENTICATION_SERVICES**
- **ORA-12638** is really a “catch-all” error – SQLNET tracing might be required

# One Final Error / Solution

```
ORA-12638: ORA-28276: Invalid ORACLE password attribute
```

- No shadow password in **`orclCommonAttribute`** in Active Directory
  - Change AD password to create shadow hash

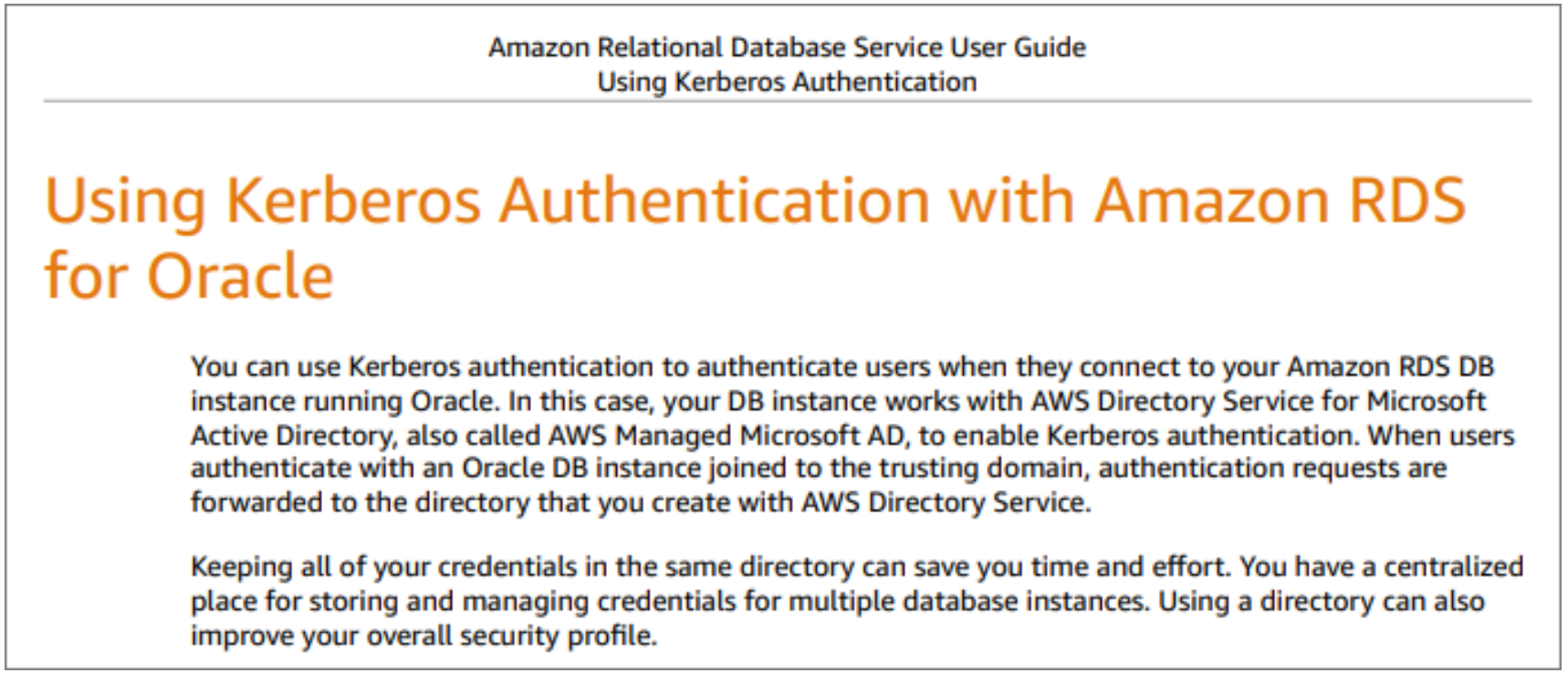
- Ensure user is part of **`ORA_VFR_...`** AD Security Group

# WRAP UP!

# Summary…

- CMU finally means authorisation and authentication can finally be ***easily*** offloaded to Microsoft Active Directory:
  - If using AD organisationally, new users will need to be in AD anyway
- Some initial one-time setup is required:
  - AD schema needs to be extended & password filter installed (for “password” option)
  - RDBMS home requires `dsi.ora`, Oracle Wallet, and initialisation parameters
  - Less AD setup required for “Kerberos” based authentication
- Actual Database user and role management is easy

# Interesting: Even Included with Amazon RDS



Amazon Relational Database Service User Guide
Using Kerberos Authentication

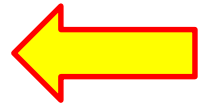
## Using Kerberos Authentication with Amazon RDS for Oracle

You can use Kerberos authentication to authenticate users when they connect to your Amazon RDS DB instance running Oracle. In this case, your DB instance works with AWS Directory Service for Microsoft Active Directory, also called AWS Managed Microsoft AD, to enable Kerberos authentication. When users authenticate with an Oracle DB instance joined to the trusting domain, authentication requests are forwarded to the directory that you create with AWS Directory Service.

Keeping all of your credentials in the same directory can save you time and effort. You have a centralized place for storing and managing credentials for multiple database instances. Using a directory can also improve your overall security profile.

- Not personally tested
- Reference: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/oracle-kerberos.html

# MOS Notes

- How To Configure Authentication For The Centrally Managed Users In An 18c Database (Doc ID 2462012.1)
- Tracing CMU connection issues (Doc ID 2470608.1)
- 18c Active Directory Password Authentication Fails With ORA-28276 for Client Connections Below 12c (Doc ID 2472256.1)
- How To Configure Kerberos Authentication In A 12c Database (Doc ID 1996329.1)
- Configuring ASO Kerberos Authentication with a Microsoft Windows 2008 R2 Active Directory KDC (Doc ID 1304004.1)
- Kerberos Troubleshooting Guide (Doc ID 185897.1)
- Bug 28994890 : CMU-AD: CUMULATIVE FIXES FOR DATABASE 18C 

# COLLABORATE 20

TECHNOLOGY & APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY

## APRIL 19-23, 2020
## LAS VEGAS

The COLLABORATE Quest Forum is the official Oracle user event for Oracle PeopleSoft, JD Edwards, Cloud, and Database & Technology products.

**DRIVE** innovation. **IMPROVE** customer experience. **CONNECT** with experienced professionals.

# REGISTRATION IS NOW OPEN!

**Make your plans early and save up to 40%**

**questoraclecommunity.org/collaborate**

# THANK YOU

**http://bit.ly/OraCMU-UKOUG19**

**pane@pythian.com**