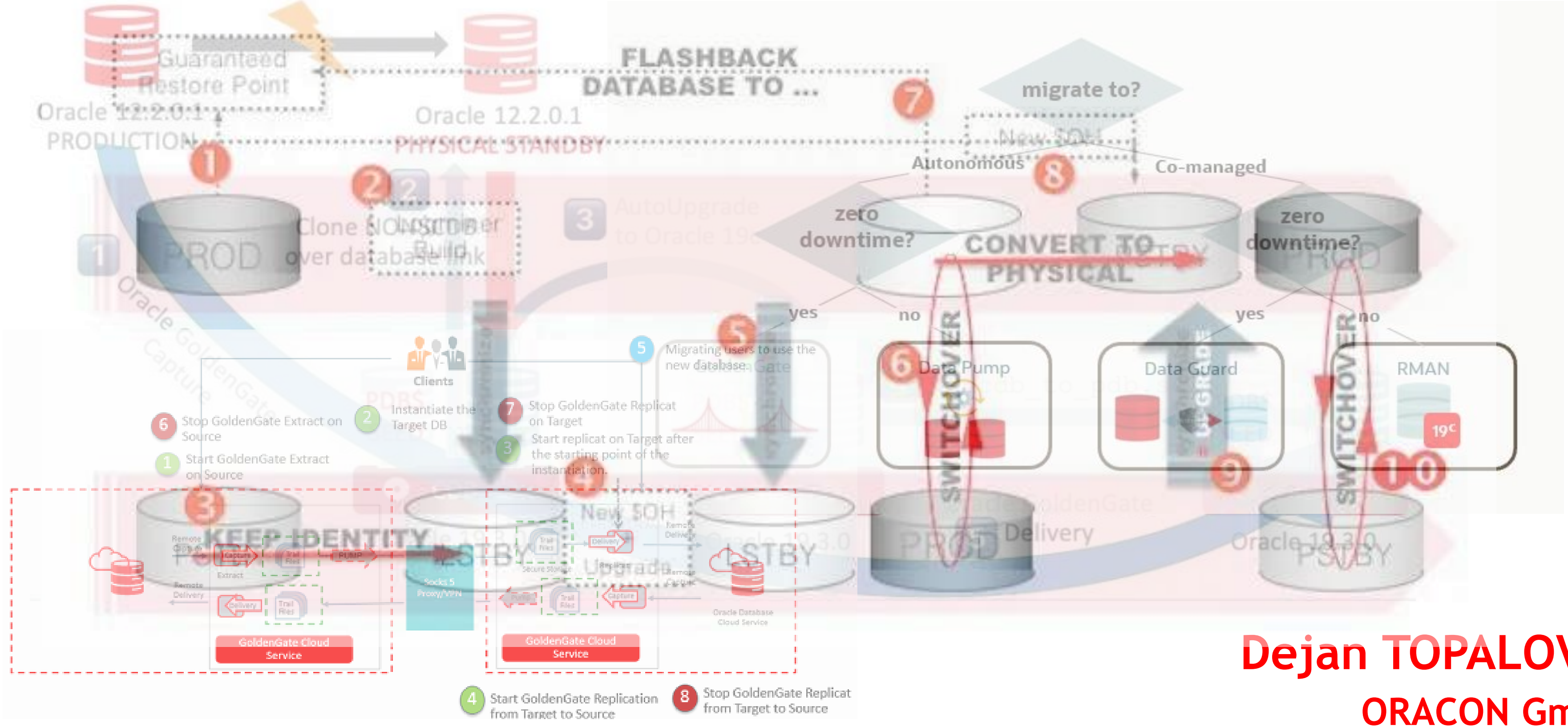
Rolling Upgrade

Keep it simple!


Guaranteed Restore Point
Oracle 12.2.0.1 PRODUCTION
Oracle 12.2.0.1 PHYSICAL STANDBY
FLASHBACK DATABASE TO ...
migrate to?
New SOH
Autonomous
Co-managed
zero downtime?
CONVERT TO PHYSICAL
yes
no
Data Pump
Data Guard
RMAN
19c
SWITCHOVER
UPGRADE
PROD
PSTBY
Oracle 18.3.0
Clone NONCDB over database link
Instantiate the Target DB
AutoUpgrade to Oracle 19c
Clients
Migrating users to use the new database
Stop GoldenGate Extract on Source
Start GoldenGate Extract on Source
Stop GoldenGate Replicat on Target
Start replicat on Target after the starting point of the instantiation.
KEEP IDENTITY
LSTBY
Upgrade
Delivery
GoldenGate Cloud Service
Socks 5 Proxy/VPN
Oracle Database Cloud Service
Start GoldenGate Replication from Target to Source
Stop GoldenGate Replicat from Target to Source
1
2
3
4
5
6
7
8
9
10

Dejan TOPALOVIĆ

ORACON GmbH

About me

- Working with Oracle since 2003.
- **OCM** (Oracle Certified Master)
- **OCE** (Oracle Certified SQL Expert)

@oradeto | E-Mail: dejan@oracon.at

Oracle ACE Associate

- Running two companies in Vienna, Austria:
  - **ORACON** GmbH
  - **Finee** GmbH (Co-founder)

- **Customers**:
  - Raiffeisen Bank
  - Erste Bank
  - Agrar Markt Austria
  - DB Schenker
  - Europ Assistance
  - Frequentis
  - etc.

Agenda

1. Preparations for the migration & upgrade (ca. 5 minutes)
2. Overview of some methods & options for the **migration** (ca. 5 minutes)
3. Overview of some methods & options for the **upgrade** (ca. 5 minutes)
4. **Migration** & **Upgrade**: Keep it **simple!** (ca. 20 minutes)
5. Gotchas & Restrictions (ca. 5 minutes)
6. Questions & Answers (no time! ;-))

Preparation: Setting up the goals & targets

Kick-off meeting with the customer:

- Taking an overview of the current environment
- Defining RPO (Recovery Point Objective) and RTO (Recovery Time Objective)
    - Defining max. downtime & fallback scenario
- Defining team members & their roles/duties
- Defining a timescope & Go-Live dates
- Choosing a right method for the migration and upgrade

Preparation: Setting up the goals & targets

- **Downtime (RPO/RTO)**: max. 1 hour
- **Budget**: max. EUR 100.000
- **Team members**: 1 Project manager, 2 System Administrators (incl. Network, storage etc.), 1 internal Oracle DBA & 1 ext. Oracle DBA (me!), 2 Application managers, 2 Testers
- **Source DB**: EE, single instance, 11.2.0.4 on Windows, on-premises, no Standby DB
- **Target DB**: EE, single instance, 19c on Windows, on Azure Cloud, no Standby DB
- **Timescope**: end of June (2.5 months; before „high season“)

Some methods for the database migration

- DataPump Export/Import
- Transportable Tablespaces (xTTS)
- Full Transportable Export/Import (FTEX)
- ZDM (Zero Downtime Migration)
- RMAN Incremental Backups
- GoldenGate
- physru.sh
- DBMS_ROLLING
- Rolling Upgrade using Transient Logical Standby Database

→ Some methods for the database upgrade

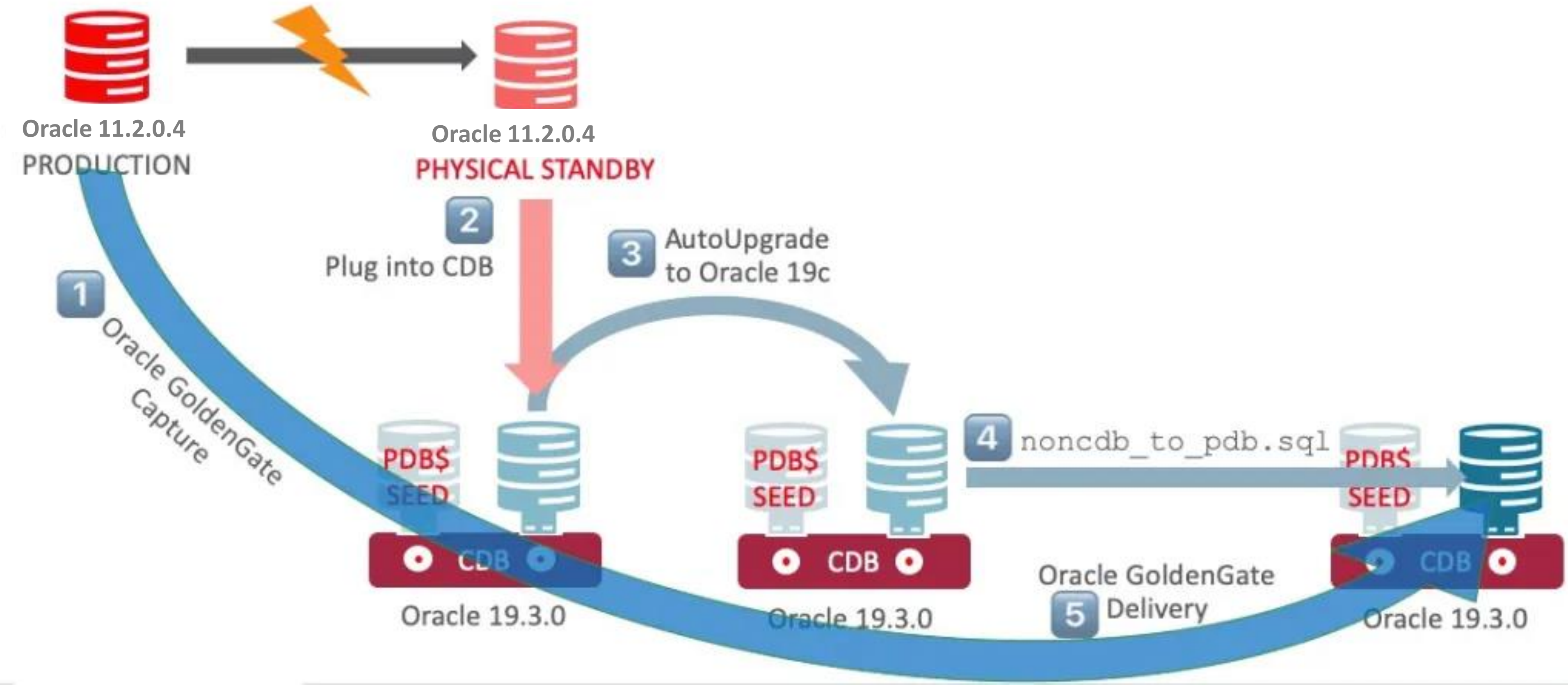
Some methods for the database upgrade

- DataPump Export/Import
  - ACLs, Java Classes incl. grants
- **DBUA**
  - Guaranteed Restore Point & Flashback: Check out for FLASHBACK_ON=NO in v$tablespace!
- Autoupgrade.jar
  - Tried to use it for a DB on Windows on 3 different environments – **failed**! (Mike Dietrich we have to talk!)

Migration: Why Rolling Upgrade?

- **ZDM**: free, but according to Oracle, it works for Oracle Cloud and AWS; not specified for Azure Cloud
- **physru.sh** is unsupported for >11g versions, and for Linux
- **DBMS_ROLLING** package starting with 12.1.0.2, requires license for Active Data Guard & needs time to implement it
- **GoldenGate** too expensive & too complex to set up, and needed only once for the migration; **Free of charge for 183 days**, but only for Oracle Cloud
- **xTTS**, **FTEX** & **DataPump**: downtime too big
- **Rolling Upgrade**: free, min. downtime

Migration: Why Rolling Upgrade?

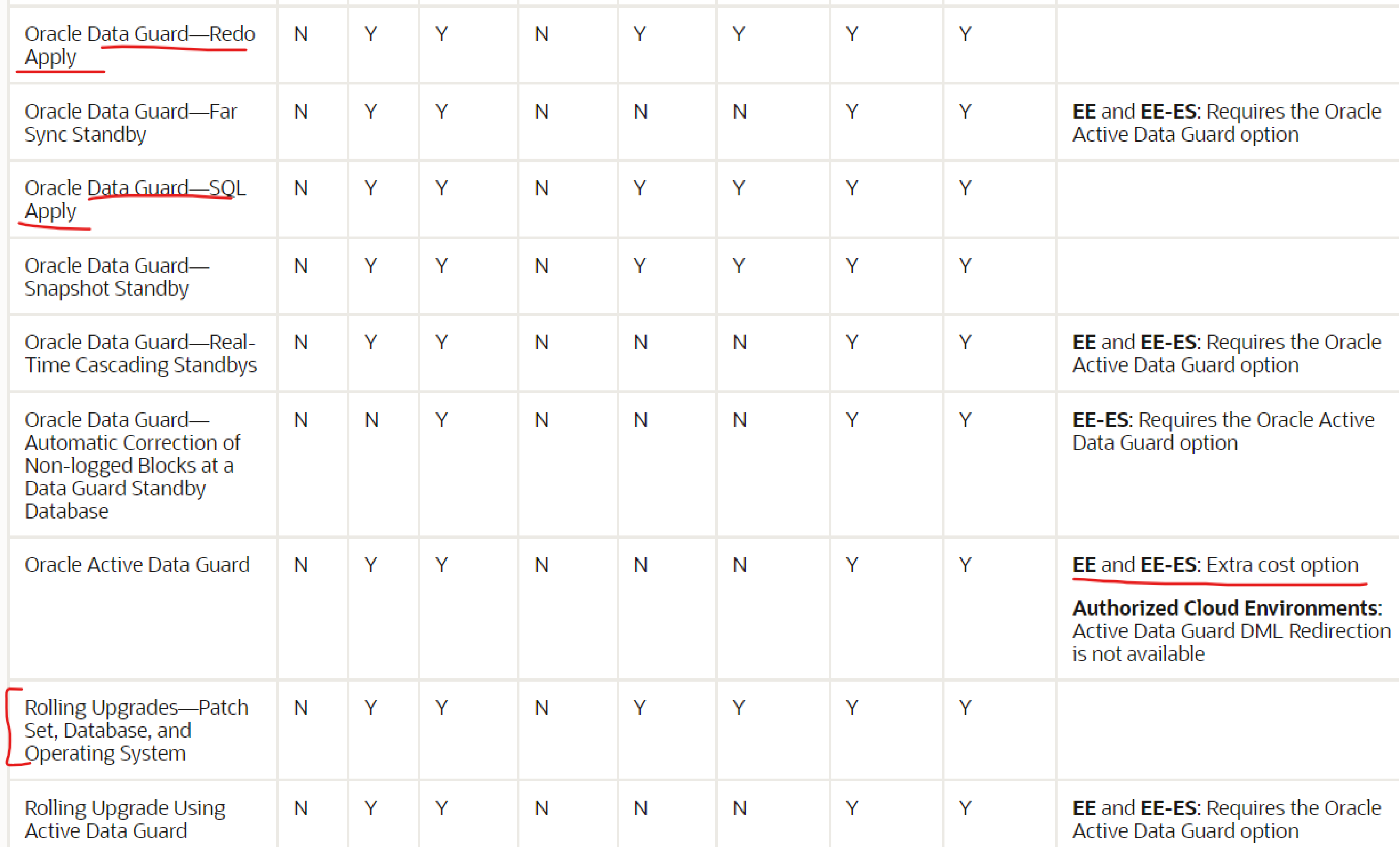


Rolling Upgrade

Picture: Oracle Corp., Mike Dietrich

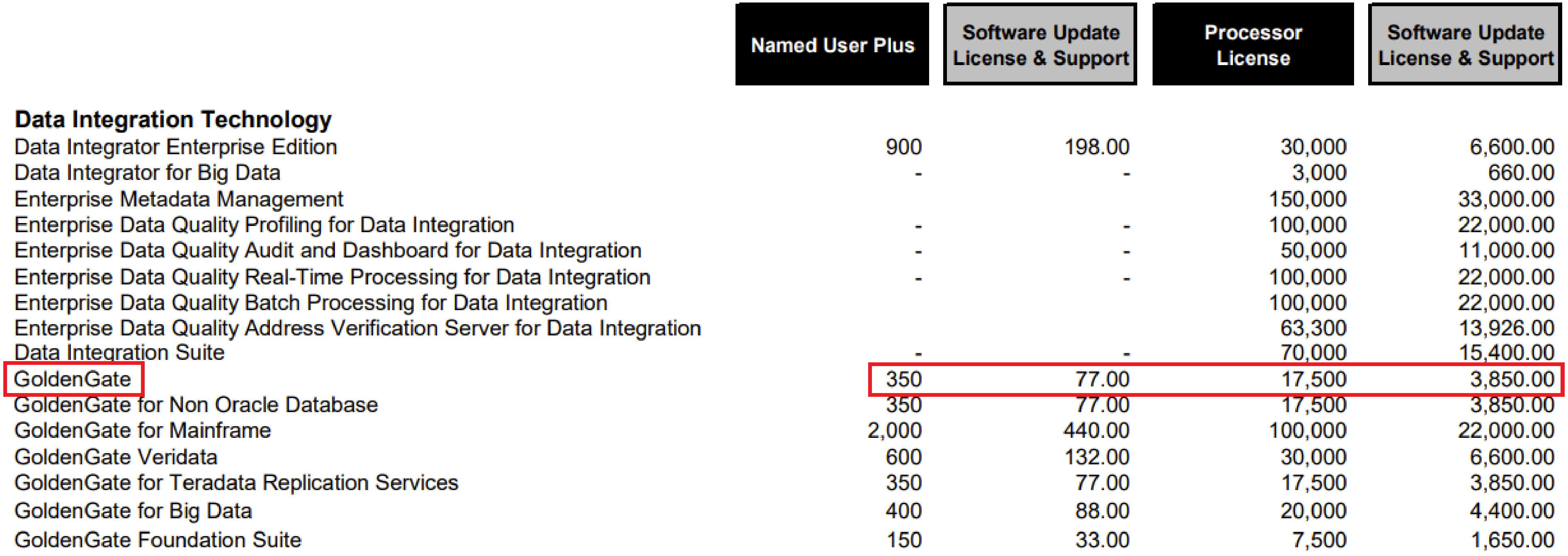
→ Migration: Licensing

Migration: Licensing



Oracle Data Guard—Redo Apply	N	Y	Y	N	Y	Y	Y	Y	
Oracle Data Guard—Far Sync Standby	N	Y	Y	N	N	N	Y	Y	**EE** and **EE-ES**: Requires the Oracle Active Data Guard option
Oracle Data Guard—SQL Apply	N	Y	Y	N	Y	Y	Y	Y	
Oracle Data Guard—Snapshot Standby	N	Y	Y	N	Y	Y	Y	Y	
Oracle Data Guard—Real-Time Cascading Standbys	N	Y	Y	N	N	N	Y	Y	**EE** and **EE-ES**: Requires the Oracle Active Data Guard option
Oracle Data Guard—Automatic Correction of Non-logged Blocks at a Data Guard Standby Database	N	N	Y	N	N	N	Y	Y	**EE-ES**: Requires the Oracle Active Data Guard option
Oracle Active Data Guard	N	Y	Y	N	N	N	Y	Y	**EE** and **EE-ES**: Extra cost option **Authorized Cloud Environments**: Active Data Guard DML Redirection is not available
Rolling Upgrades—Patch Set, Database, and Operating System	N	Y	Y	N	Y	Y	Y	Y	
Rolling Upgrade Using Active Data Guard	N	Y	Y	N	N	N	Y	Y	**EE** and **EE-ES**: Requires the Oracle Active Data Guard option

Migration: Licensing



	Named User Plus	Software Update License & Support	Processor License	Software Update License & Support
Data Integration Technology				
Data Integrator Enterprise Edition	900	198.00	30,000	6,600.00
Data Integrator for Big Data	-	-	3,000	660.00
Enterprise Metadata Management			150,000	33,000.00
Enterprise Data Quality Profiling for Data Integration	-	-	100,000	22,000.00
Enterprise Data Quality Audit and Dashboard for Data Integration	-	-	50,000	11,000.00
Enterprise Data Quality Real-Time Processing for Data Integration	-	-	100,000	22,000.00
Enterprise Data Quality Batch Processing for Data Integration			100,000	22,000.00
Enterprise Data Quality Address Verification Server for Data Integration			63,300	13,926.00
Data Integration Suite	-	-	70,000	15,400.00
GoldenGate	350	77.00	17,500	3,850.00
GoldenGate for Non Oracle Database	350	77.00	17,500	3,850.00
GoldenGate for Mainframe	2,000	440.00	100,000	22,000.00
GoldenGate Veridata	600	132.00	30,000	6,600.00
GoldenGate for Teradata Replication Services	350	77.00	17,500	3,850.00
GoldenGate for Big Data	400	88.00	20,000	4,400.00
GoldenGate Foundation Suite	150	33.00	7,500	1,650.00

2 x 4 CPUs x $ 17.500 = $ 140.000

Migration: Rolling Upgrade - Steps

Steps to Perform a Rolling Upgrade With an Existing Physical Standby:

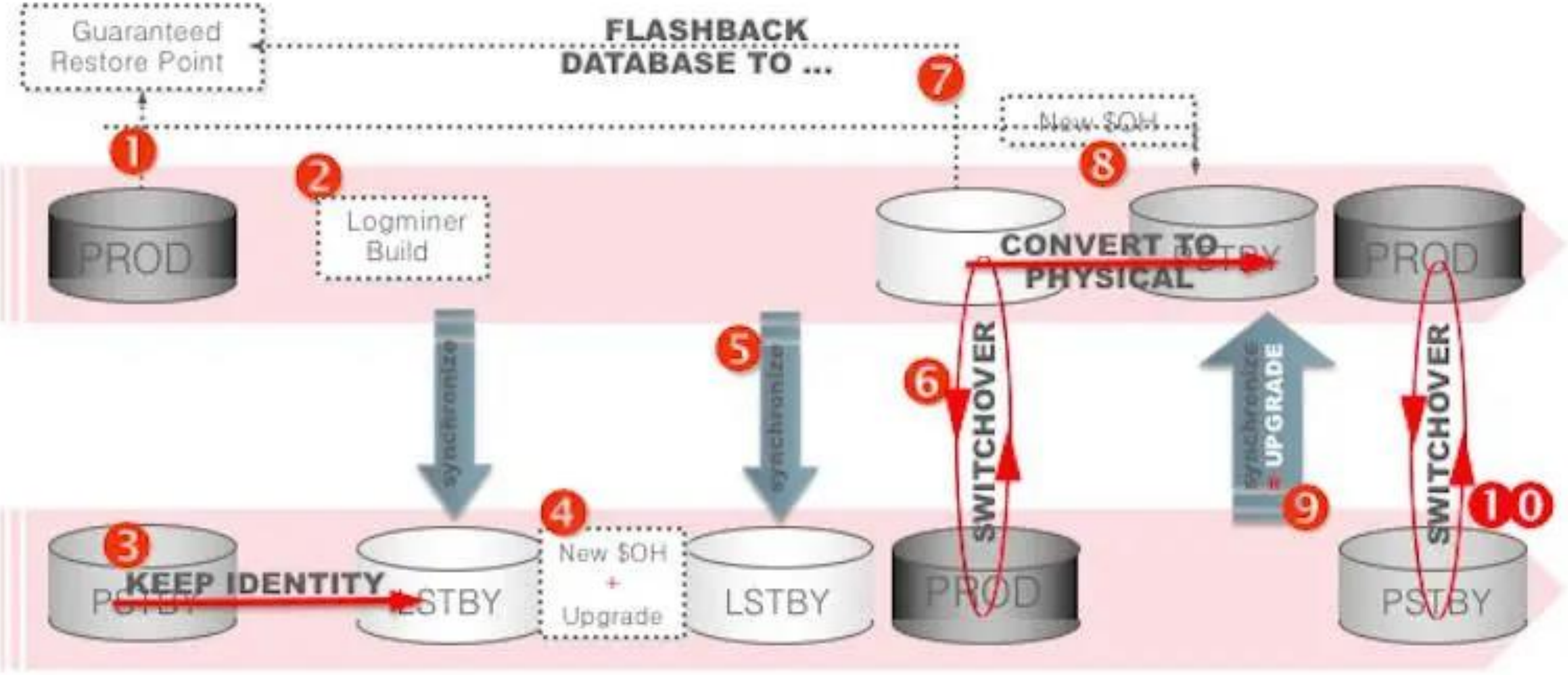
Step	**Description**
Step 1	Prepare the primary database for a rolling upgrade (perform these steps on Database A)
Step 2	Convert the physical standby database into a logical standby database (perform these steps on Database B)
Step 3	**Upgrade the logical standby database and catch up with the primary database** (perform these steps on Database B)
Step 4	Flashback Database A to the guaranteed restore point (perform these steps on Database A)
Step 5	Mount Database A using the new version of Oracle software
Step 6	Convert Database A to a physical standby
Step 7	Start managed recovery on Database A
Step 8	Perform a switchover to make Database A the primary database
Step 9	Clean up the guaranteed restore point created in Database A

Migration: Rolling Upgrade

***Steps to Perform a Rolling Upgrade With an Existing Logical Standby*:**

Step	Description
Step 1	Prepare for rolling upgrade
Step 2	Upgrade the logical standby database
Step 3	Restart SQL Apply on the upgraded logical standby database
Step 4	Monitor events on the upgraded standby database
Step 5	Begin a switchover
Step 6	Import any tables that were modified during the upgrade
Step 7	Complete the switchover and activate user applications
Step 8	Upgrade the old primary database
Step 9	Start SQL Apply on the old primary database
Step 10	Optionally, raise the compatibility level on both databases
Step 11	Monitor events on the new logical standby database
Step 12	Optionally, perform another switchover

Migration: Rolling Upgrade



Rolling Upgrade

Picture: Oracle Corp., Mike Dietrich

→ Migration: Rolling Upgrade – Setup

Migration: Rolling Upgrade - Setup

Install and configure the Software on the new server:

- Set up a new DB server (storage, network, users)
- Zip and transfer the original 11g Oracle Home
- Unzip the 11g Oracle Home and clone it:

```
set PERL5LIB=C:\oracle\11.2.0.4\perl\lib
set PATH=C:\oracle\11.2.0.4\perl\bin;%PATH%
cd C:\oracle\11.2.0.4\clone\bin
C:\oracle\11.2.0.4\perl\bin\perl clone.pl
ORACLE_HOME="C:\oracle\11.2.0.4"
ORACLE_HOME_NAME="OraDB11gR1_home"
ORACLE_BASE="C:\oracle" ORACLE_HOME_USER=oracle
```

- Set up tnsnames.ora and listener.ora
- Create a dummy DB instance
- Drop datafiles, but leave the Service/Instance (or use **oradim**)

Migration: Rolling Upgrade - Technical details

Set up a physical Standby database:

- Either use the init file from deleted instance or create a new one (as a copy from the source DB), and set up all parameters accordingly
- Check the connectivity between both servers
- Use RMAN to duplicate the source database to standby:

```
rman target sys@databaseA auxiliary sys@databaseB
```

```
RMAN> RUN {
ALLOCATE CHANNEL c1 TYPE DISK;
ALLOCATE CHANNEL c2 TYPE DISK;
ALLOCATE AUXILIARY CHANNEL caux1 TYPE DISK;
DUPLICATE TARGET DATABASE
    FOR STANDBY
    FROM ACTIVE DATABASE
    DORECOVER
    NOFILENAMECHECK;
}
```

Migration: Rolling Upgrade - Technical details

Set up a physical Standby database:

- Create standby redo logs on both databases:

```
 select add_standby
    from (
select 'alter database add standby logfile thread '|| thread# ||'
group '|| to_char((select max(group#) from v$log ) + rownum)||
                '(''&REDOLOGLOCATION'|| to_char((select max(group#)
from v$log ) + rownum) ||'.log'') size '|| (select
max(bytes/1024/1024) from v$log) ||'M REUSE;' as add_standby
        from v$log
        order by thread#, group#
     );
```

- Enable flashback and start the synchronization:

```
SQL> alter database flashback on;
SQL> alter database recover managed standby database using current
logfile disconnect;
```

Migration: Rolling Upgrade - Technical details

Set up a physical Standby database:

- Synchronization before converting to the Logical Standby database:
  - Either set up a Data Guard

    or
  - Recover the standby database using RMAN (standby needs to be in MOUNT mode):

```
RMAN> recover database from service ‘PRODA’
       section size  512m using compressed backupset;
```

Migration: Rolling Upgrade - Technical details

Convert physical Standby database to the Logical:

- On primary check, which tables columns have unsupported data types:

```
SELECT DISTINCT OWNER, TABLE_NAME FROM
DBA_LOGSTDBY_UNSUPPORTED;
```

- On primary database execute this command:

```
begin
dbms_logstdby.build;
end;
/
```

If you skip this step, then the next one on standby (KEEP IDENTITY) will hang forever!

Migration: Rolling Upgrade - Technical details

Convert physical Standby database to the Logical:

- On standby:

```
ALTER DATABASE RECOVER TO LOGICAL STANDBY KEEP IDENTITY;
ALTER DATABASE OPEN;
```

- Verify the database role:

```
select name,open_mode,db_unique_name,database_role
from v$database;

EXECUTE DBMS_LOGSTDBY.APPLY_SET('LOG_AUTO_DELETE', 'FALSE');
EXECUTE DBMS_LOGSTDBY.APPLY_SET('MAX_EVENTS_RECORDED',
DBMS_LOGSTDBY.MAX_EVENTS);

EXECUTE DBMS_LOGSTDBY.APPLY_SET('RECORD_UNSUPPORTED_OPERATIONS', 'TRUE');
ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE;
```

- When the database is synchronized, then stop redo apply:

```
ALTER DATABASE STOP LOGICAL STANDBY APPLY;
```

Migration: Rolling Upgrade - Technical details

Database Upgrade - Prechecks:

- On standby server, execute these commands:

```
C:\ORACLE\11.2.0.4\jdk\bin\java -jar
C:\ORACLE\19\rdbms\admin\preupgrade.jar FILE DIR C:\oracle
```

- And execute the commands provided in the log directory, i.e.:

```
@C:\ORACLE\preupgrade_fixups.sql
alter system set DB_RECOVERY_FILE_DEST_SIZE=32768m;
alter system set java_pool_size=117440512;
alter system set processes=300 scope=spfile;
alter system set shared_pool_size=692060160;
purge dba_recyclebin;
exec dbms_stats.gather_dictionary_stats;
@?/rdbms/admin/emremove.sql
@?/olap/admin/catnoamd.sql
@?/rdbms/admin/catnoexf.sql
```

Migration: Rolling Upgrade - Technical details

Database Upgrade - Prechecks:

- On standby server, recompile invalid objects:

```
@?\rdbms\admin\utlrp
SET SERVEROUTPUT ON;
EXECUTE DBMS_PREUP.INVALID_OBJECTS;
```

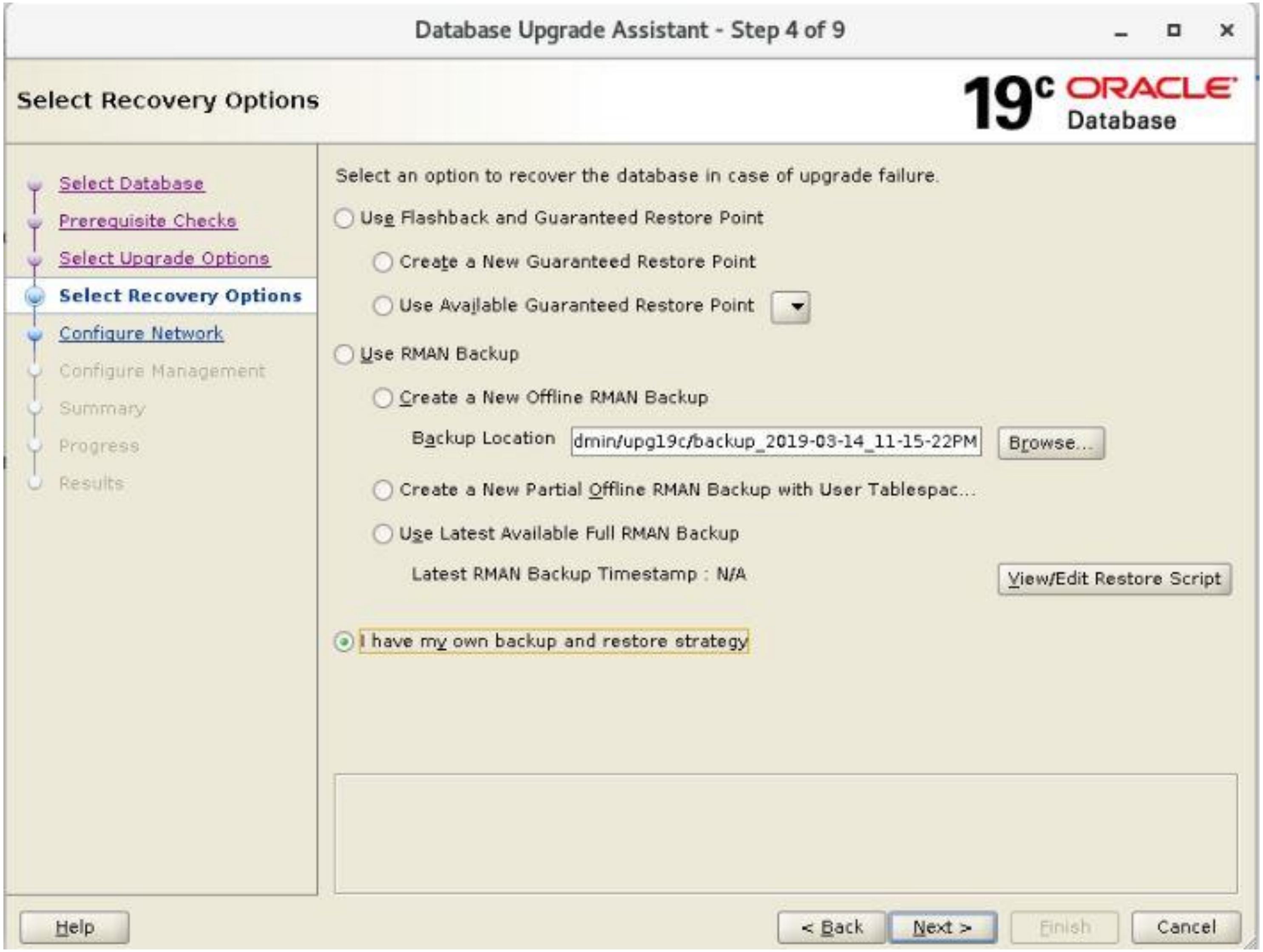
- Check, for which users the password shall be changed:

```
select username, password_versions from dba_users where
password_versions = '10G';
```

- Before upgrade, disable all „on database“ triggers:

```
 SELECT OWNER, TRIGGER_NAME FROM DBA_TRIGGERS WHERE
      TRIM(BASE_OBJECT_TYPE)='DATABASE' AND OWNER NOT IN (SELECT
GRANTEE FROM
      DBA_SYS_PRIVS WHERE PRIVILEGE='ADMINISTER DATABASE TRIGGER');
```

Migration: Rolling Upgrade - DBUA



Select Recovery Options

Select an option to recover the database in case of upgrade failure.

- ○ Use Flashback and Guaranteed Restore Point
  - ○ Create a New Guaranteed Restore Point
  - ○ Use Available Guaranteed Restore Point
- ○ Use RMAN Backup
  - ○ Create a New Offline RMAN Backup
    - Backup Location: dmin/upg19c/backup_2019-03-14_11-15-22PM
  - ○ Create a New Partial Offline RMAN Backup with User Tablespac...
  - ○ Use Latest Available Full RMAN Backup
    - Latest RMAN Backup Timestamp : N/A
- ● I have my own backup and restore strategy

Picture: Oracle Corp.

→ Migration: Rolling Upgrade – Technical details

Migration: Rolling Upgrade - Technical details

Database Upgrade – Postfixups and checks:

- On standby server, execute these commands:

```
@C:\ORACLE\postupgrade_fixups.sql
select name,open_mode,db_unique_name,database_role,version from
v$database,v$instance;
SELECT version FROM v$timezone_file;
select count(*) from dba_objects where status='INVALID';

select COMP_ID,COMP_NAME,VERSION,STATUS from dba_registry;
```

- Change the passwords, if needed
- If applications are using old Oracle clients, then adapt sqlnet.ora on standby server:

```
  SQLNET.ALLOWED_LOGON_VERSION=11
  SQLNET.ALLOWED_LOGON_VERSION_CLIENT=11
  SQLNET.ALLOWED_LOGON_VERSION_SERVER=11
```

Migration: Rolling Upgrade - Technical details

Restart Logical redo apply:

- On standby server, execute these commands:

```
ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE;

ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MON-YY HH24:MI:SS';
SELECT SYSDATE, APPLIED_TIME FROM V$LOGSTDBY_PROGRESS;

SET LONG 1000
SET PAGESIZE 180
SET LINESIZE 79

SELECT EVENT_TIMESTAMP, EVENT, STATUS FROM DBA_LOGSTDBY_EVENTS ORDER
BY EVENT_TIMESTAMP;
```

Migration: Rolling Upgrade - Technical details

Switchover:

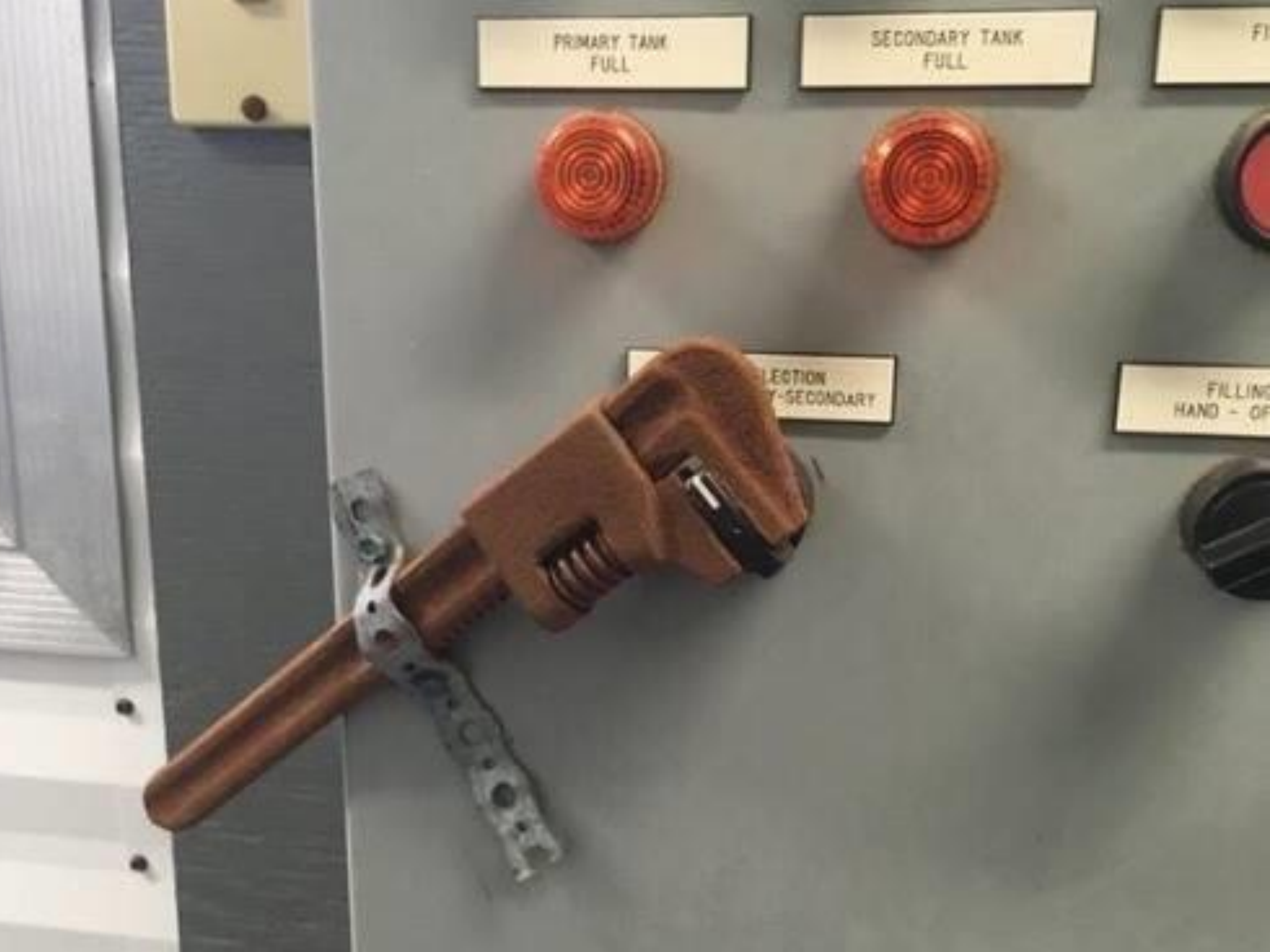
- On primary server, execute this command:

```
alter system set job_queue_processes=0;

ALTER DATABASE COMMIT TO SWITCHOVER TO LOGICAL STANDBY;
```

- On standby server, execute these commands:

```
SELECT SWITCHOVER_STATUS FROM V$DATABASE;

ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY;
```

→ Migration: Rolling Upgrade – Gotchas


PRIMARY TANK
FULL
SECONDARY TANK
FULL
FI
LECTION
Y-SECONDARY
FILLIN
HAND - OF

Migration: Rolling Upgrade - Gotchas

- COVID19: 2 team members off for 10 days
- Alternative Go-Live date! First attempt failed.
  - Both databases had PRIMARY role!
  - `ALTER DATABASE PREPARE TO SWITCHOVER CANCEL;`
- Hardcoded parameters!
- Network ACLs
- No prepared MAA connection strings
- Unsupported data types
- **The Incarnation# of the database needs to be brought in sync**. Otherwise, the **next** attempt to do a Transient Logical Standby Rolling Upgrade may fail!

Migration: Rolling Upgrade - Restrictions

- Logical standby databases do not support Oracle Label Security.
- Logical standby databases do not fully support an Oracle E-Business Suite implementation because there are tables that contain unsupported data types.
- Data type restrictions (11.2): *
    - » BFILE
    - » Collections (including VARRAYS and nested tables)
    - » Multimedia data types (including Spatial, Image, and Oracle Text)
    - » ROWID, UROWID
    - » User-defined types

Migration: Rolling Upgrade - Restrictions

- Data type restrictions (12.1) *:
  - BFILE
  - ROWID, UROWID
  - Collections (including VARRAYs and nested tables)
  - Objects with nested tables and REFs
  - The following Spatial types are not supported:
    - MDSYS.SDO_GEORASTER
    - MDSYS.SDO_TOPO_GEOMETRY
  - Identity columns

Dejan TOPALOVIĆ
Oracle Consultant
ORACON GmbH

+43 650 305 45 45
dejan@oracon.at
https://oracon.at

→ END

Illustration: La Linea, Osvaldo Cavandoli ; Pictures: Oracle Corp., Mike Dietrich, Sinan Petrus Toma