Automated patching : Pivot from Manual to Scalable with Oracle Database Lifecycle Management Pack

Ashwin Vaidya

Ashwin Vaidya PMP , ITIL

MTS-2 Database Engineering , Database Platform Security

About me

- Oracle Technologist since 1995
- Specialist in RAC, VLDB, Golden Gate & 24x7 installations
- Managed Large Oracle ERP programs as program manager
- Varied responsibilities at Paypal including DB engineering/architecture/operations
- Mongodb/Cassandra/Couchbase

About Paypal

- Oracle powers our transactional data
- One of the largest private clouds in the world
- 16PB stored on Oracle
- 10+ TB streamed daily via Goldengate
- 192 million active accounts

Two decades ago, our founders invented payment technology to make buying and selling faster, secure, and easier—and put economic power where it belongs: **In the hands of people.**

Global scale

Our customers can accept payments in > 100 currencies,
withdraw funds to their bank accounts in 56 currencies,
and hold balances in their PayPal accounts in 25 currencies.

Almost 8,000 PayPal team members provide support to our customers in over 20 languages.

We are a trusted part of people's financial lives and a partner to merchants in 200+ markets around the world.

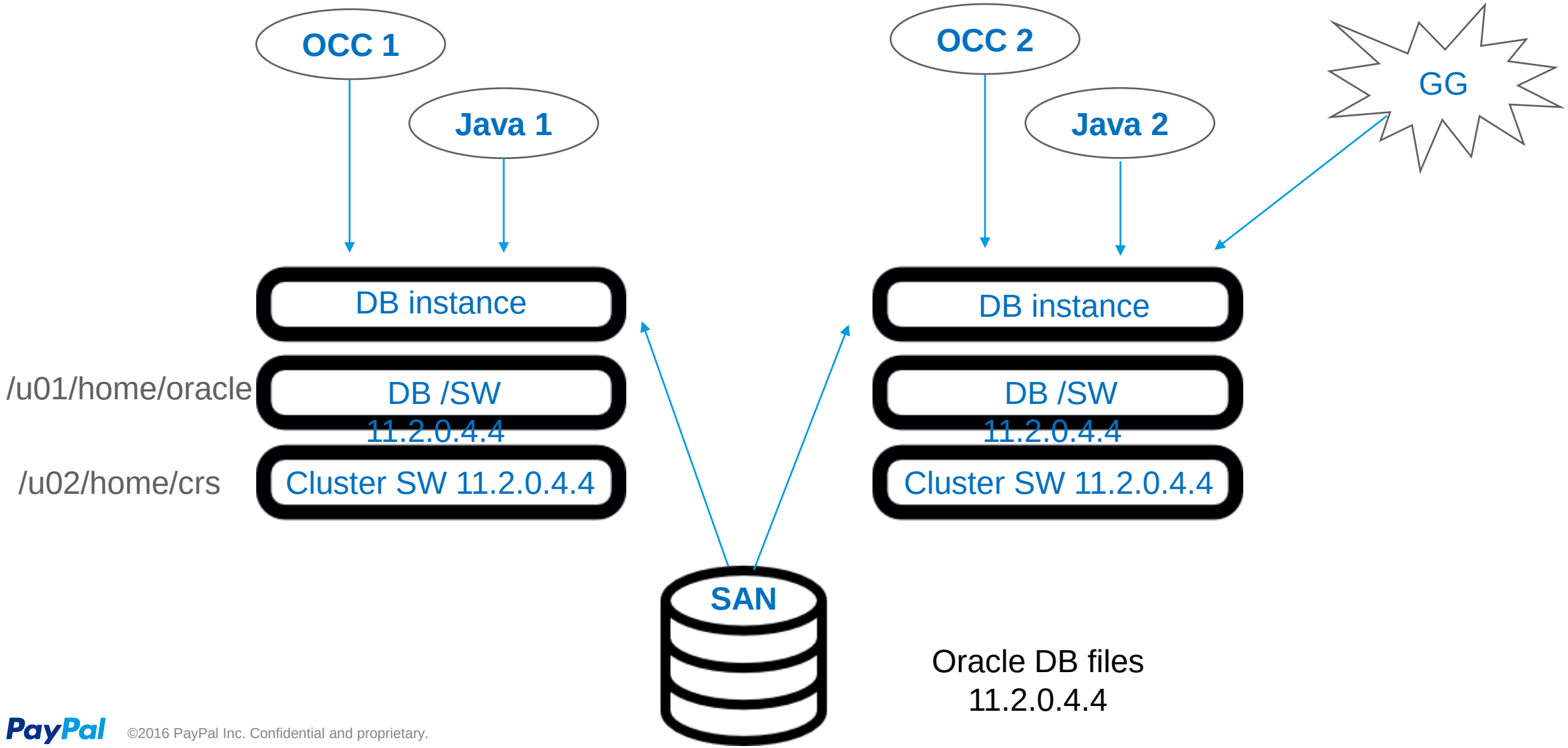
Business Challenge of Compliance & Configuration

Increasing burden of responding to internal and external audits and enforcing standards

- Initial deployment via gold images radically reduced build times
- Patching remained an enormous business challenge
- Aside from availability impacts, required 4-8 hours of dedicated DBA time per cluster

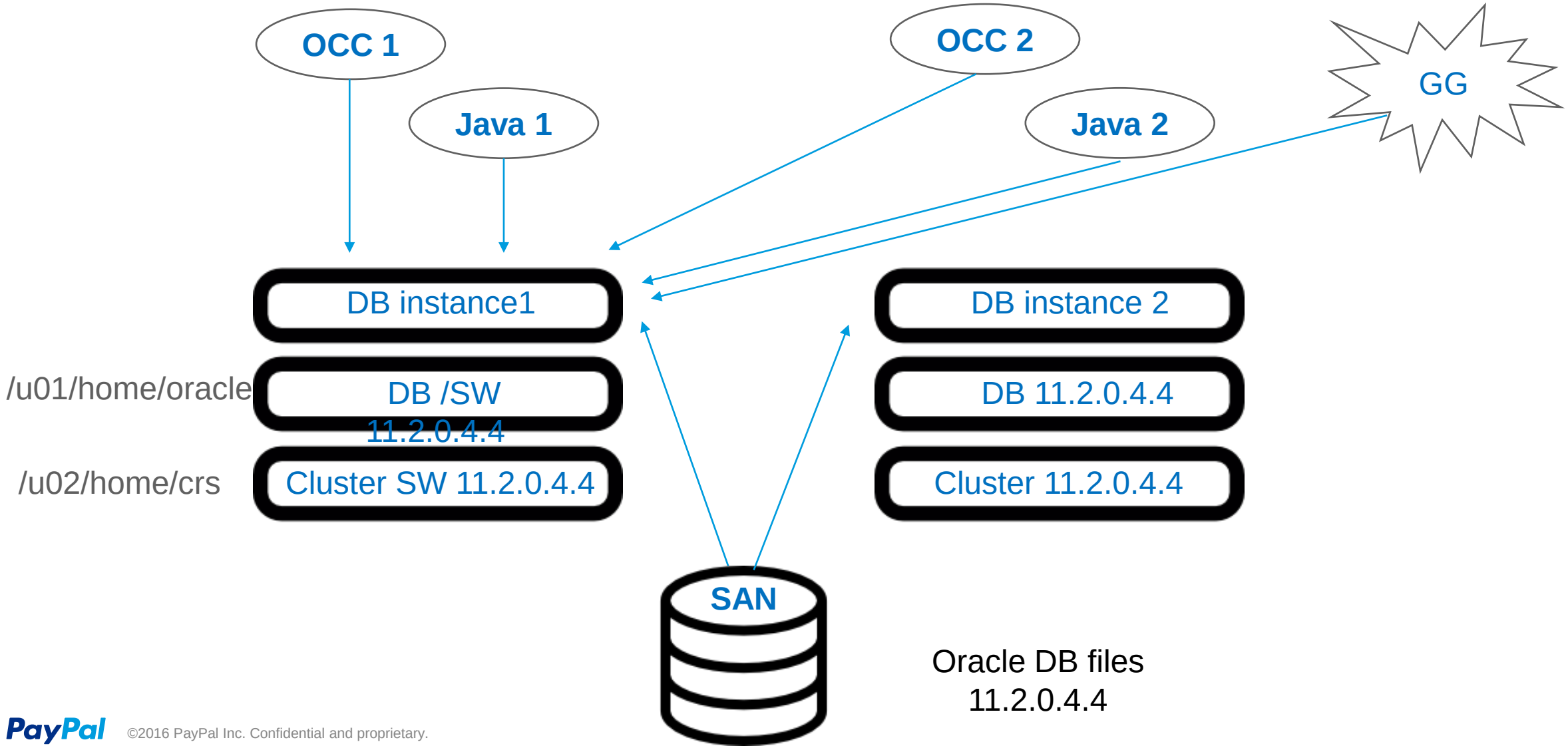
Table of contents

RAC RAC Everywhere : A basic Oracle DB @ Paypal



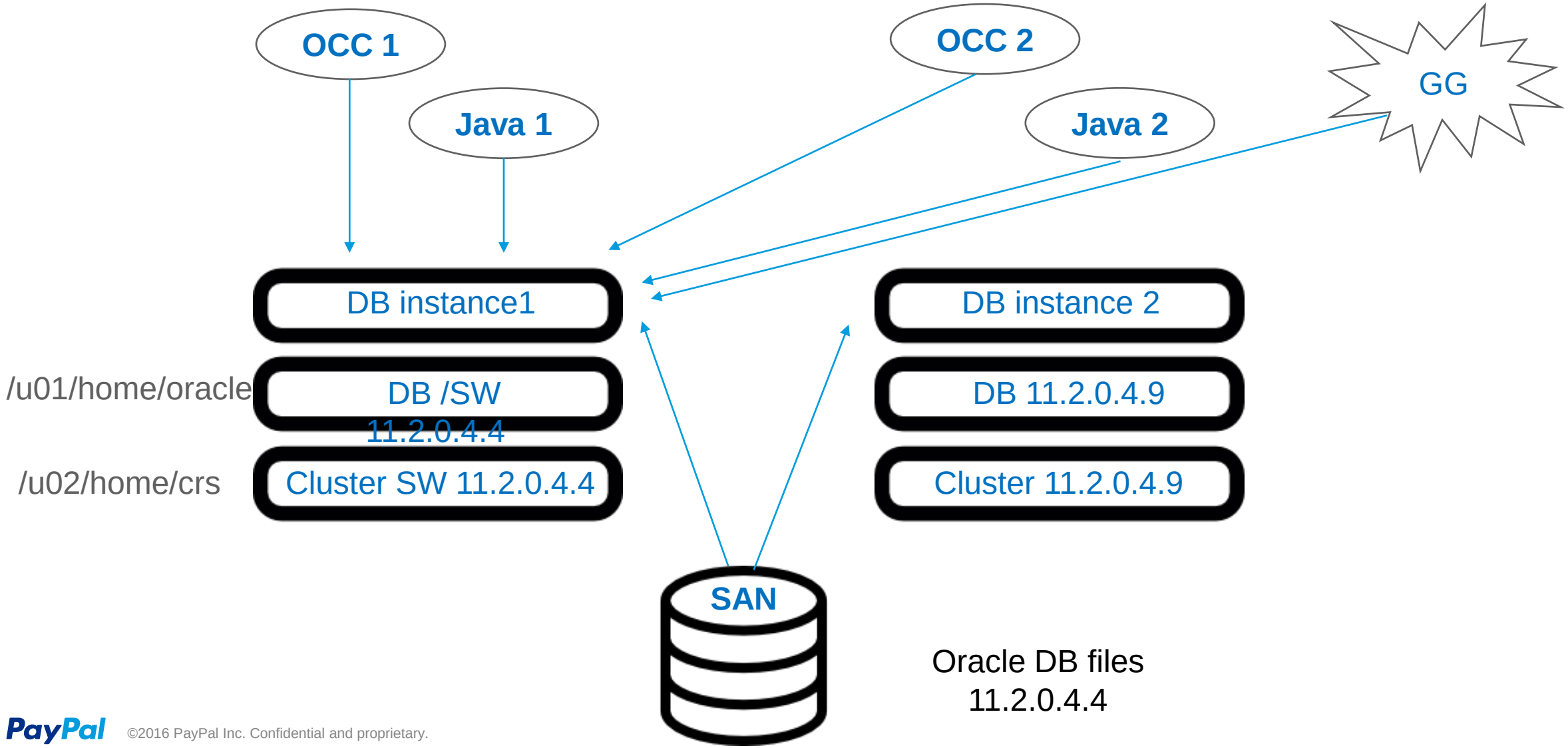
Patching a DB : Rolling patching is a must to ensure 100% availability

112044 → 112049



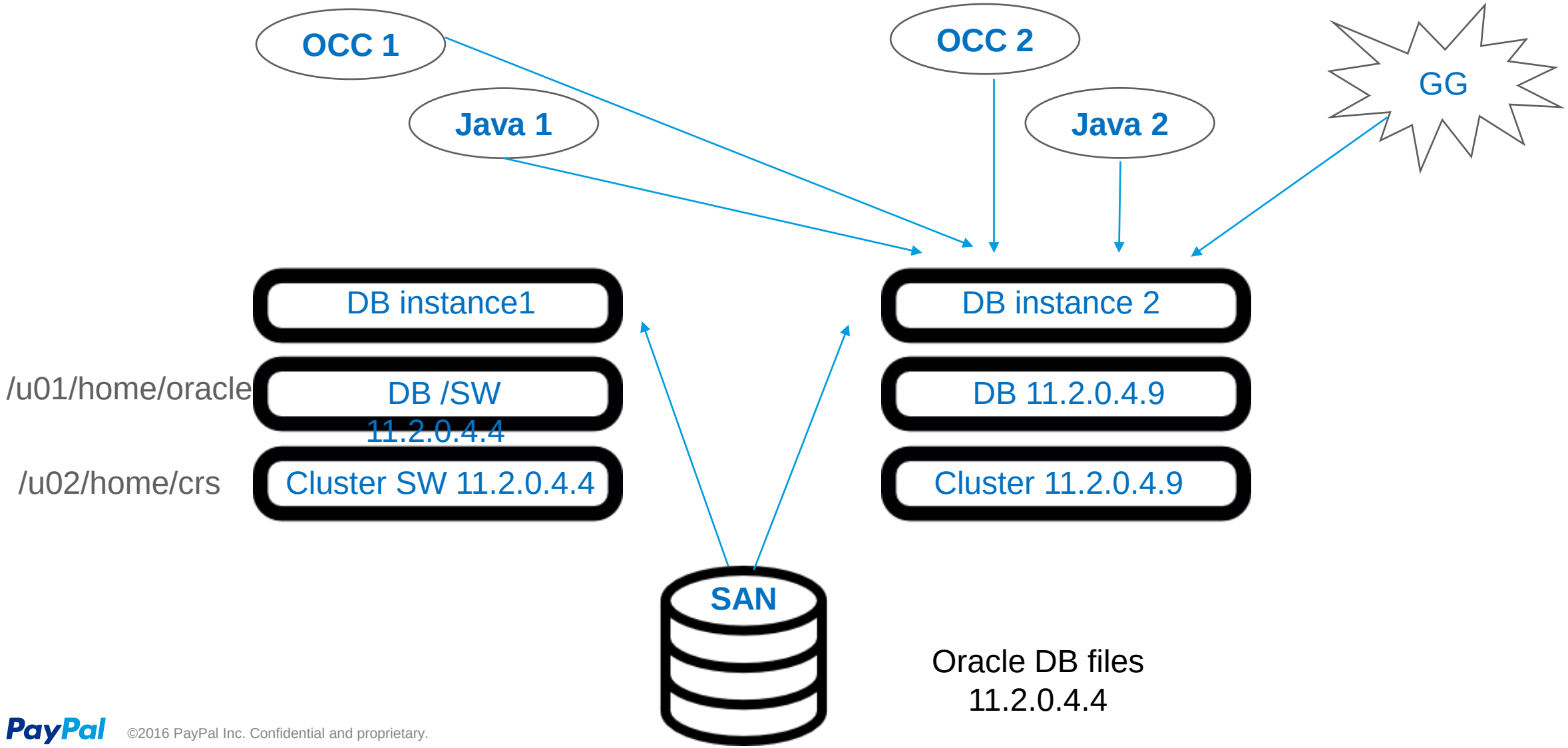
Patching a DB : Rolling patching is a must to ensure 100% availability

112044 → 112049



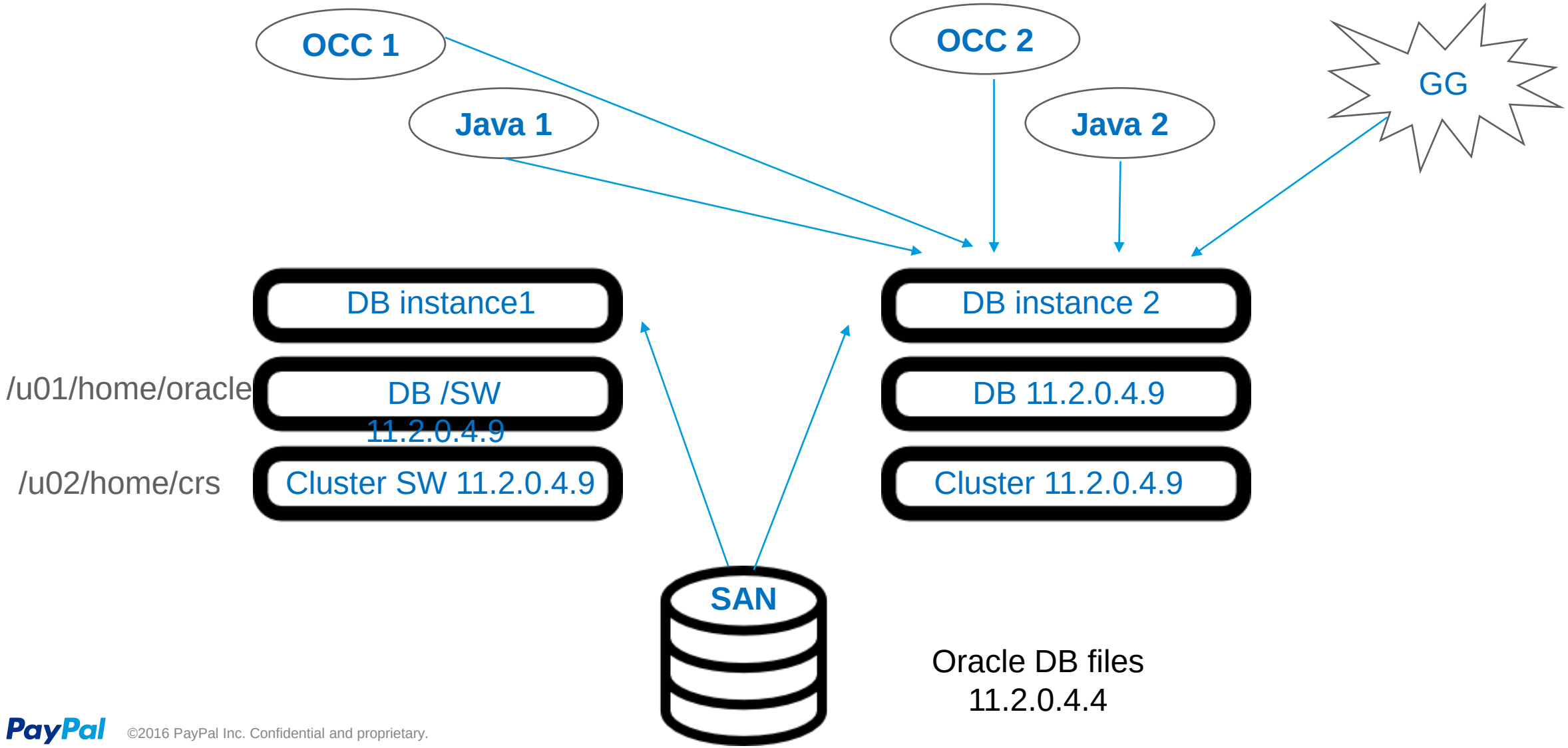
Patching a DB : Rolling patching is a must to ensure 100% availability

112044 → 112049



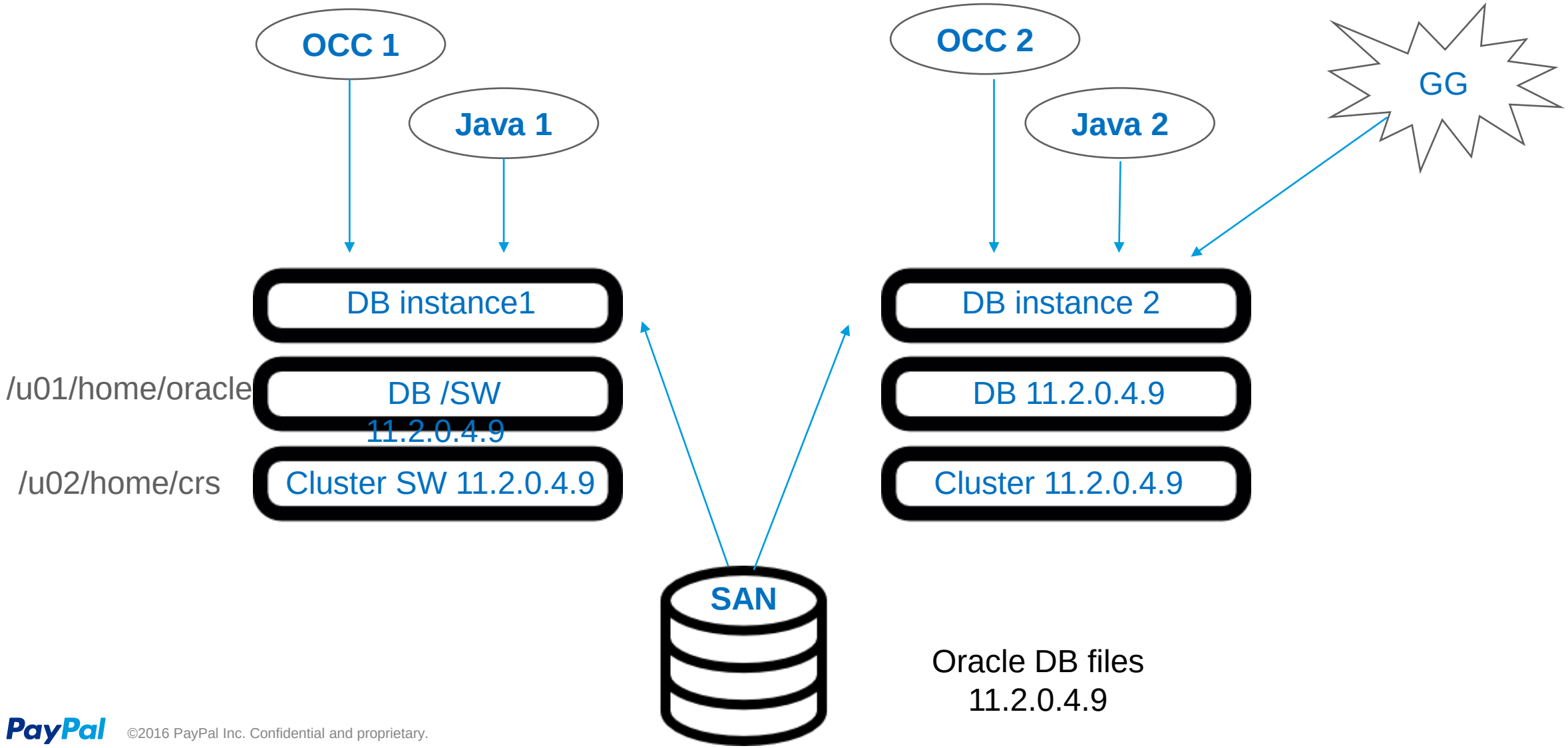
Patching a DB : Rolling patching is a must to ensure 100% availability

112044 → 112049



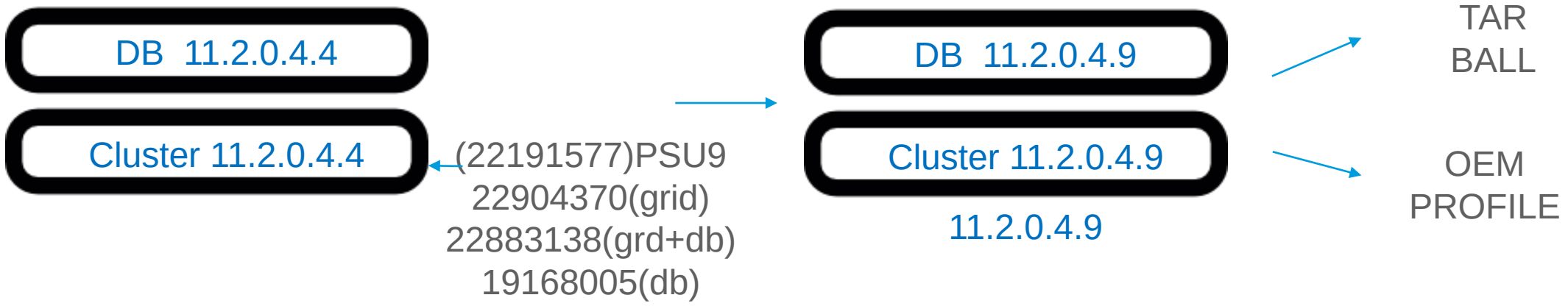
Patching a DB : Rolling patching is a must to ensure 100% availability

112044 → 112049



Manual Patching – Creating a Gold Image

- SOLARIS * PRIMARY
- LINUX * ADGRO



Manual-Patching

Using the TAR BALL/PROFILE deployment

- Shutdown
- Move 11204 11204_OLD
- Move 112049 11204
- Copy 11204_OLD/Config files 11204/Config
- Start CRS
- Start DB

- Using OEM profile to deploy new dir (but keeping HOME dir same) necessitates running clone.pl

Technical Challenges to delivering succesful automation

1. Application connections rolls are a significant challenge (OCC/Java pools/ GG)
2. Brown-outs due to reconfiguration at instance startup/stop, necessitate minimum no of stops/starts
3. Exact moment of startup/stop db instance need to be controlled
4. Patching GRID+ DB Home required db owner /super user privileges
5. PSU patching is actually PSU++ Patching

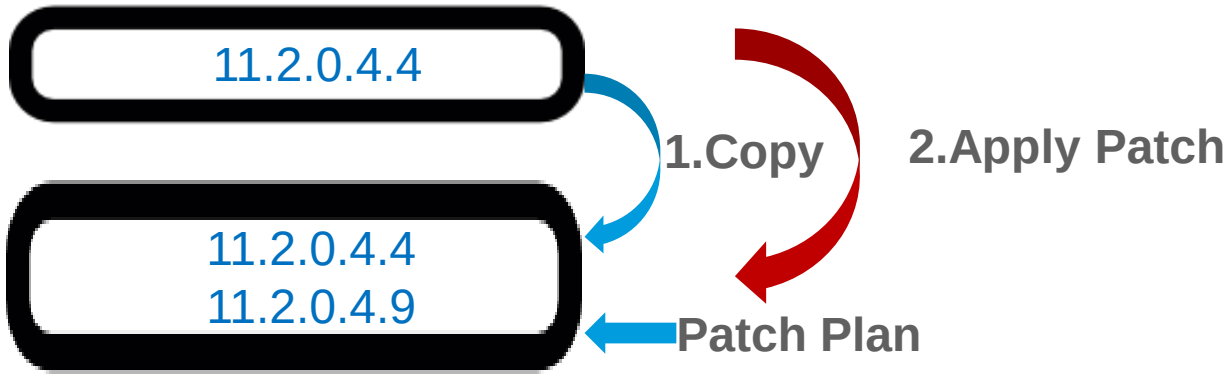
PayPal version will have for example:

PSU patch &

- 5 additional “cluster + DB” patches
- 2 cluster only patches
- 4 DB only patches

OEM – Patching Plan

- Advantages:
  - OEM has inbuilt procedures for fine-grained control of each step including DB start/stop
  - OEM has all the metadata/admin framework readily available, including methods for allowing limited root access
- Patching Plan – Out of place patching



- Stop database instance
- Point to 112049 home
- Start database instance

Issues with Patching Plan

1. Cannot combine multiple patches with one patching plan -a problem for PSUs with MLRs added
2. Does not use a standard gold image, so end state is neither guaranteed nor stable.
3. Out of place patching changes home name
4. One-off , piecemeal patching.

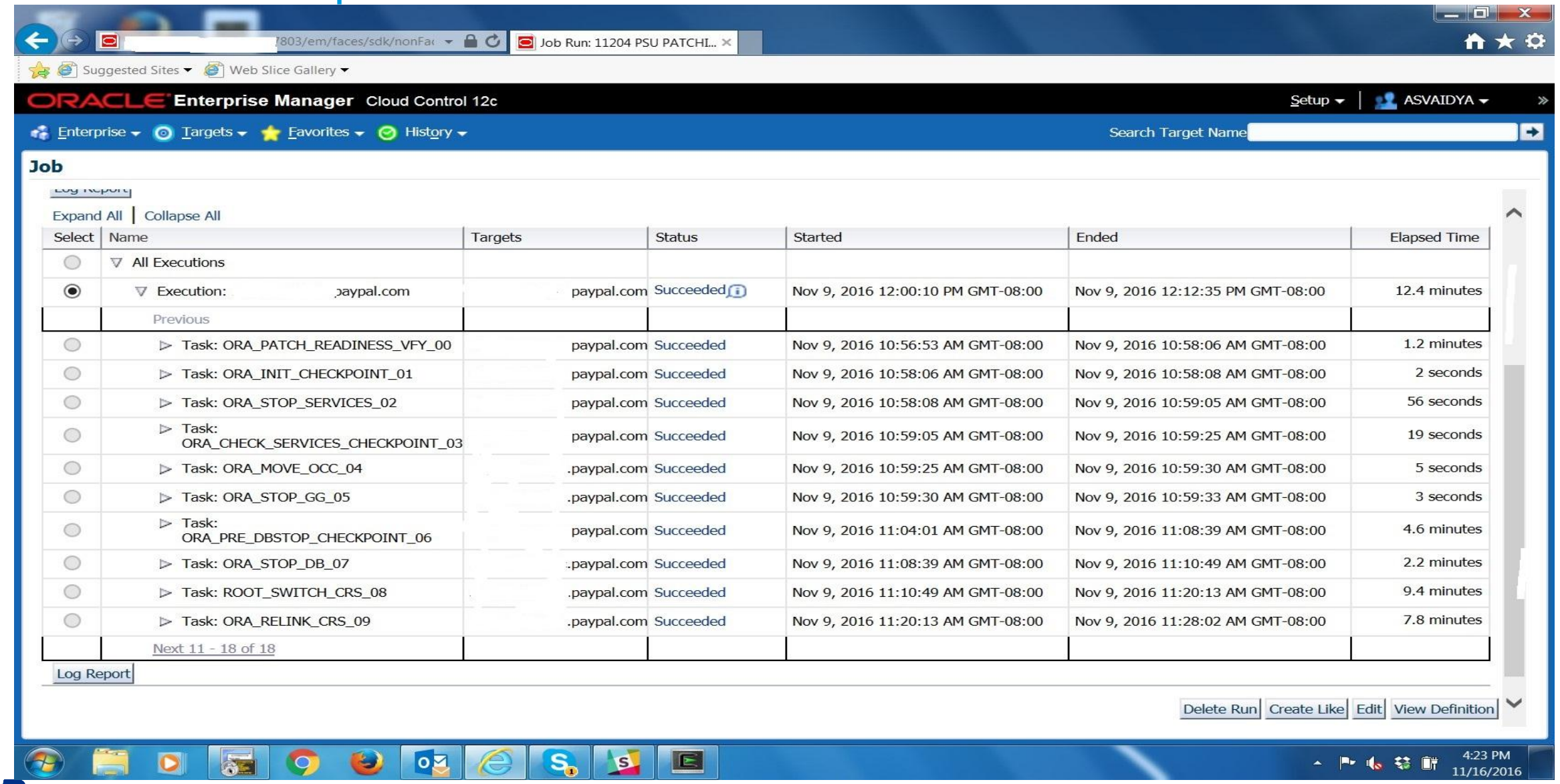
Automation with Scripts

1. Total Control
2. Uses OEM framework. Gold Image deployment is the core strategy
3. Can be submitted with a wrapper OEM job
4. Lets us keep the same HOME name

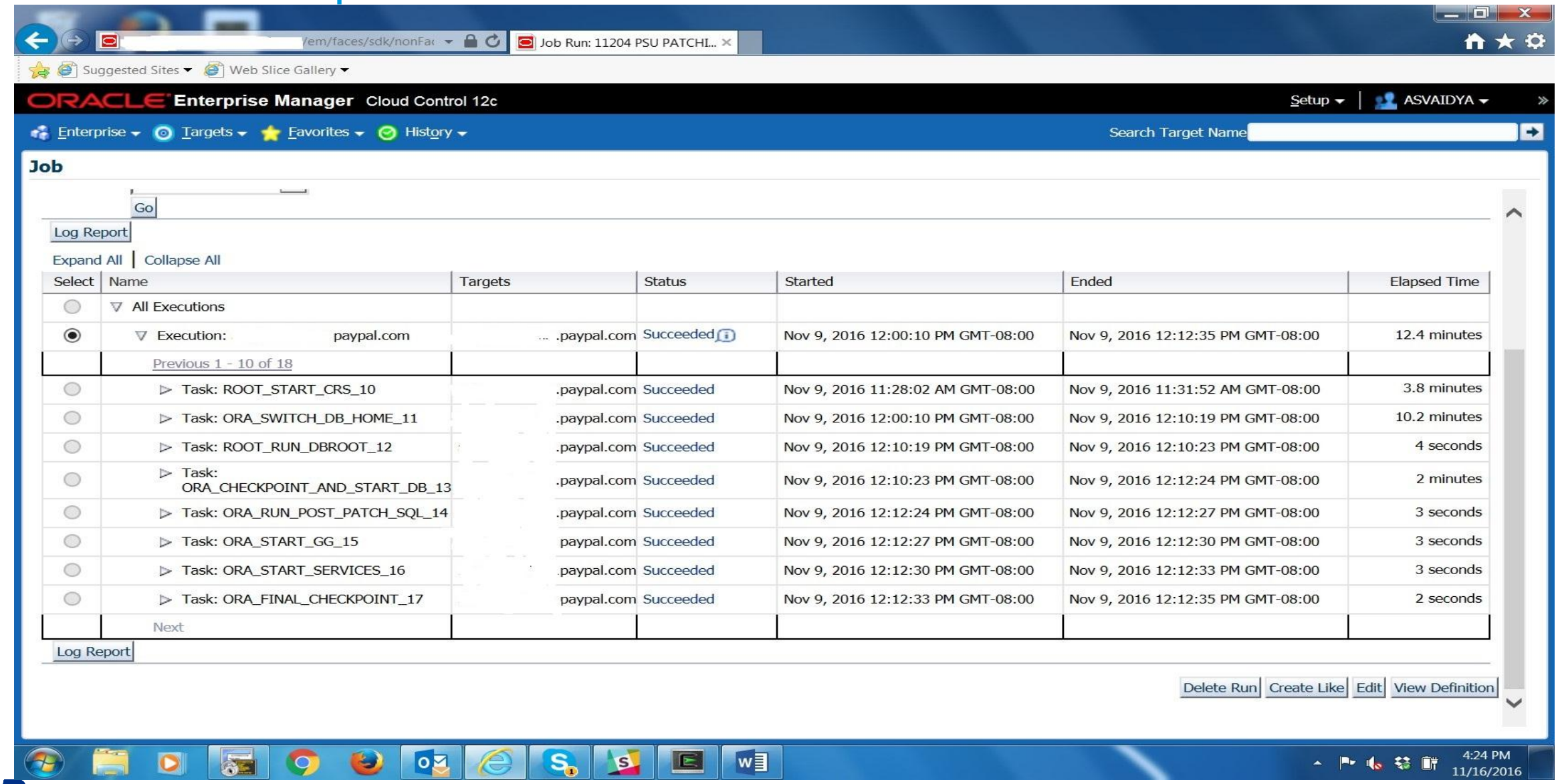
Cons

- Need to push the scripts to every node being patched and manage those deployments

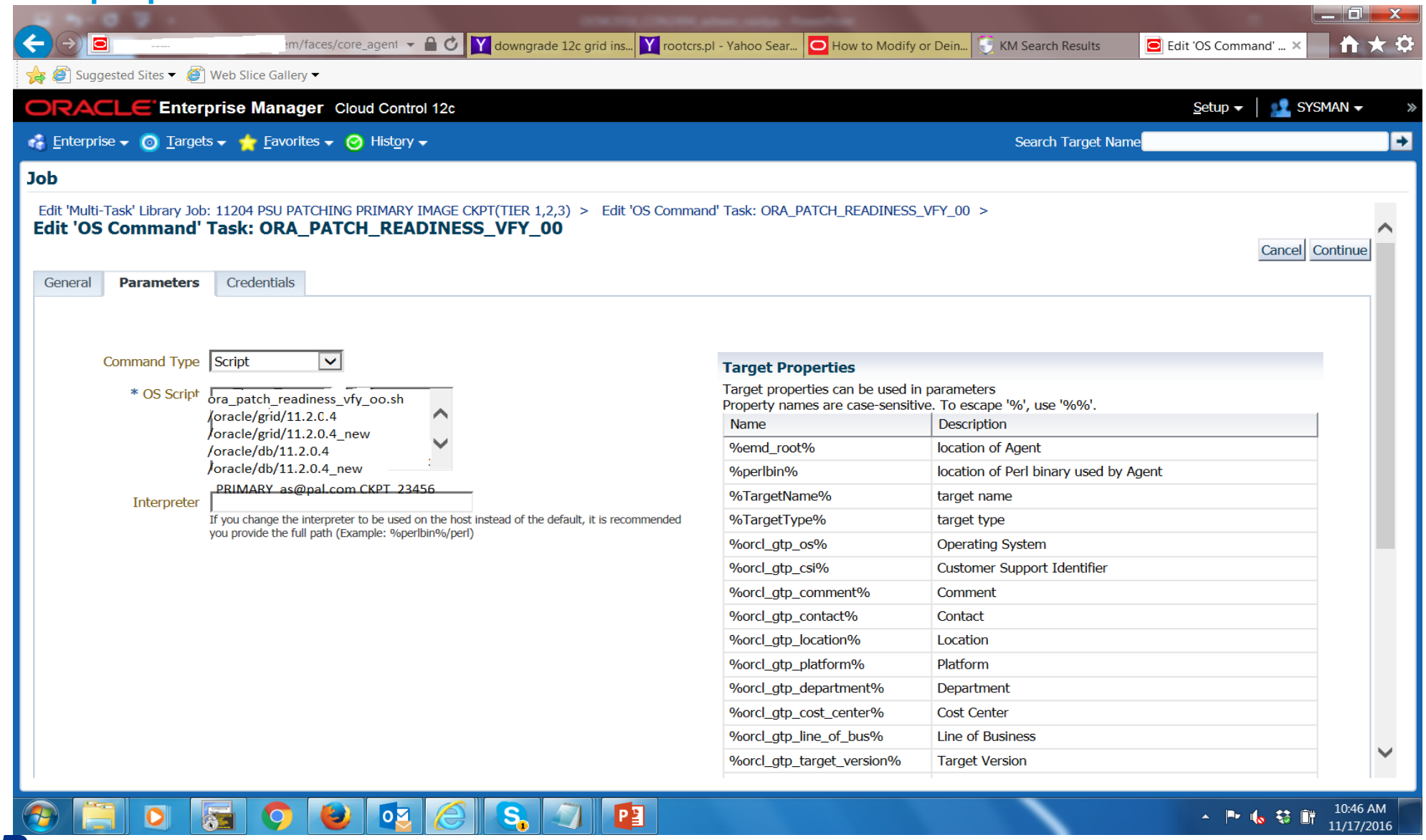
OEM Job & scripts



OEM Job & scripts



Script parameters



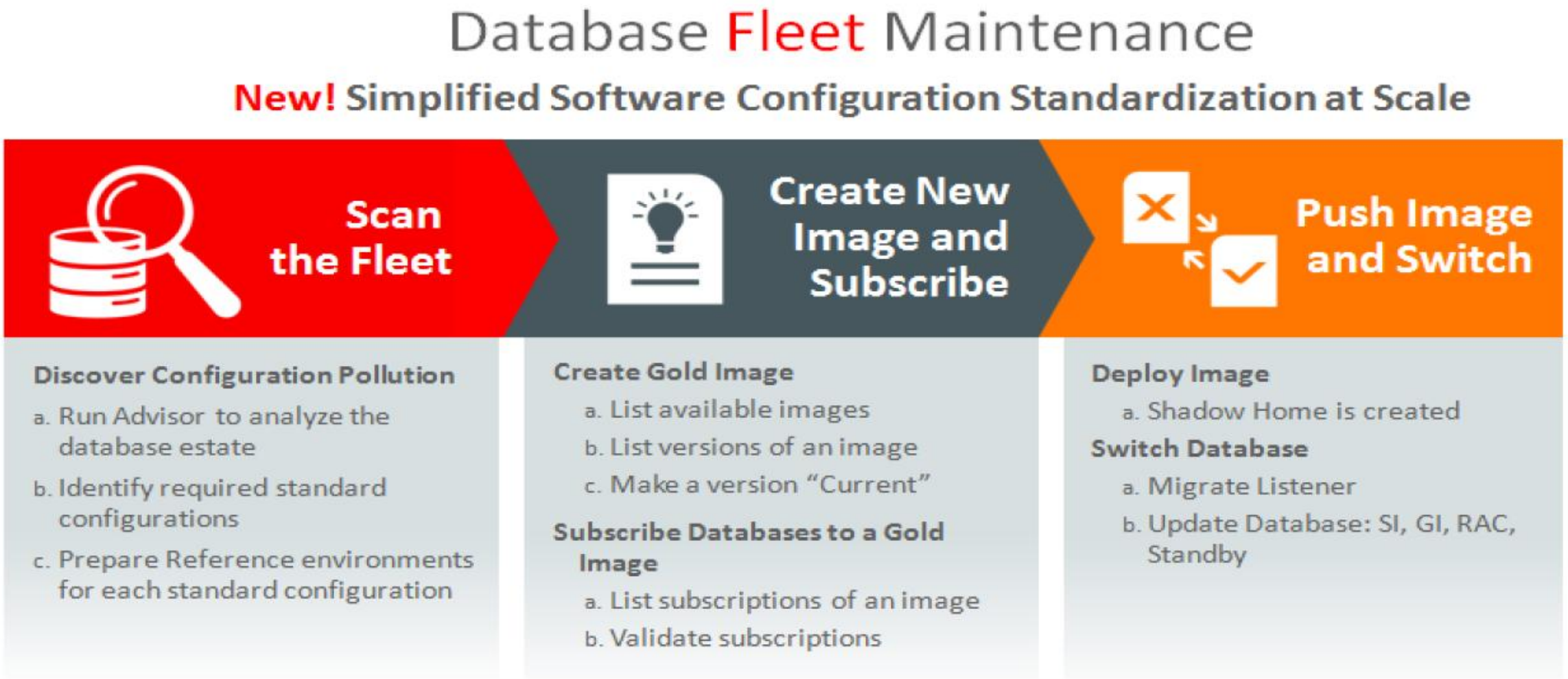
CONFIG FILE

```
IMAGE_TYPE=PRIMARY
GRIDHOMECURRENT=/oracle/grid/11.2.0.4
GRIDHOMESTAGE=/oracle/grid/11.2.0.4_new
DBHOMECURRENT=/oracle/db/11.2.0.4
DBHOMESTAGE=/oracle/db/11.2.0.4_new
PRIMARY_DB_NAME=
PATCH_NUM=23054359
PATCH_DESC=JULY2016PSU
CLUSTER_DB=dblabstb_clusterdb
INSTANCE_NAME=DBLABSTB_1
DB_ROLE=PHYSICAL STANDBY
WALLET_DIR=/oracle/admin/dblabstb/wallet
STOP_SESSION_COUNT=10
INSTANCE_NUMBER=        1
NODE_INSTANCE_COUNT=1
NODE_COUNT=2
DB_INSTANCE_COUNT=2
CRS_STORAGE_OPTION=1
CRS_NODEVIPS='nodex-vip/55.55.55.0/ixgbe2,nodey-
vip/55.55.55.0/ixgbe0'
VOTING_DISKS=NO_VAL
OCR_LOCATIONS=NO_VAL
RAC_OFF=30
```

```
EMAILID=asvaidya@paypal.com
CKPT_YES=NOCKPT
LOCAL_NODE=nodex.paypal.com
CLUSTER_NODES=nodex.paypal.com,nodey.paypal.com
LOCAL_NODE_SHORT=nodex
CLUSTER_NODES_SHORT=nodex,nodey
CENTRAL_INV_LOC=/oracle/db/oraInventory
DB_HOME_NAME="Db_11204"
GRID_HOME_NAME="Grid_11204"
CLUSTER_NAME=DBLAB_cluster
NETWORKS="ixgbe2"/16.100.8.0:public,"igb1"/199.999.0.0:cl
uster_interconnect
SCAN_NAME=DBLAB-scan.paypal.com
SCAN_PORT=2115
GNS_CONF=false
ASM_DISK_GROUP=DBLAB_DATA
ASM_DISCOVERY_STRING=/dev/rdsk/c0t*d0s4
ASM_DISKS=/dev/rdsk/c0xxxx00005084d0s4
ASM_REDUNDANCY=EXTERNAL
MRP_ORIGINAL_COUNT=
GG_PROCESS_FOUND=NO
GG_HOME_EXISTS=NO
CRS_OFFLINE_COUNT=      1
```

The future is here!

Database Fleet Maintenance

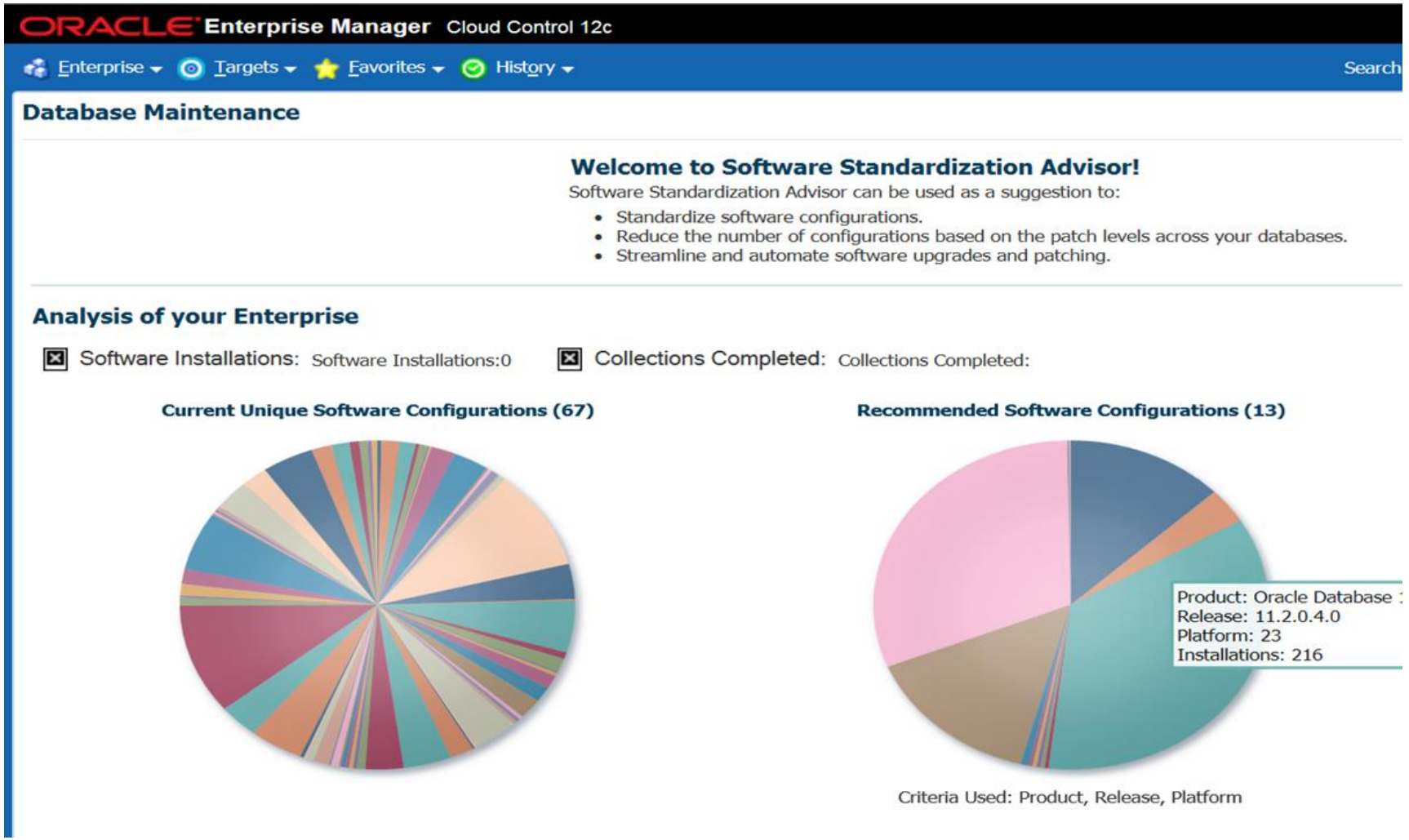


OEM Fleet Management Features

1. End State driven management
2. Can be run from a central location (Uses EMCLI, the OEM command line interface)
3. Uses gold image
4. Granular control
5. Drift management (spotting outliers) and compliance tracking is easy.

OEM Fleet Management - Start

Targets/Databases/Maintenance (Software Standardization Advisor)



OEM Fleet Management Flow -- Prepare

FOR GRID HOME*

1. Create Gold Image for GRID_HOME (off a patched & baked CRS env). (image is created with image_id)
2. If need be, update the Gold Image (each update gets a version id)
3. Set the latest/desired version as CURRENT

FOR DB HOME*

1. Create Gold Image for DB_HOME (off a patched & baked DB env). (image is created with image_id)
2. If need be, update the Gold Image (each update gets a version id)
3. Set the latest/desired version as CURRENT

* Needs to be done only once for a given OS platform / IC protocol (UDP/RDS)

OEM Fleet Management Flow -- Deploy

FOR GRID HOME*

1. For the given target, subscribe to an Image(_id).
2. Deploy the image on the target cluster in a NEW_ORACLE_HOME dir.
   - CURRENT version of the subscribed image will be selected

FOR DB HOME*

1. For the given target, subscribe to an Image(_id).
2. Deploy the image on the target cluster in a NEW_ORACLE_HOME dir
   - CURRENT version of the subscribed image will be selected

* Repeat for every target cluster

OEM Fleet Management Flow -- switch / update

FOR GRID HOME*

1. Switch GRID_HOME from current to new gold (patched) previously deployed.

FOR DB HOME*

1. Switch DB_HOME from current to new gold (patched) previously deployed.
2. Start-up DB

OEM Fleet Management Flow -- Prepare

FOR GRID HOME*

1. Create Gold Image for GRID_HOME (off a patched & baked CRS env). (image is created with image_id)

```
emcli db_software_maintenance -createSoftwareImage -input_file="data:/xyz/input_cluster”
    Input_cluster:
      IMAGE_NAME="Grid Home 11.2.0.4.9 PSU9  Gold Image"
      REF_TARGET_NAME=Grid_home_11204_2_somenode.paypal.com
      IMAGE_SWLIB_LOC=Database Provisioning Profiles/11.2.0.4.0/solaris_sparc64
      REF_GI_CREDENTIALS=ORACLE:SYSMAN
      STORAGE_NAME_FOR_SWLIB=default_loc
  emcli db_software_maintenance -getImages
```

2. If need be, update the Gold Image (each update gets a version id)
3. Set the latest/desired version as CURRENT

```
emcli db_software_maintenance -updateVersionStatus -version_id=39E663C148FXXXXXX -status=CURRENT
```

FOR DB HOME*

1. Create Gold Image for DB_HOME (off a patched & baked DB env). (image is created with image_id)

```
REF_TARGET_NAME=Db_home_11204_1_somenode.paypal.com
```

1. If need be, update the Gold Image (each update gets a version id)
2. Set the latest/desired version as CURRENT

OEM Fleet Management Flow -- Deploy

FOR GRID HOME*

1. For the given target, subscribe to an Image(_id).

```
emcli db_software_maintenance -subscribeTarget -target_name="cluster123" -target_type=cluster  -image_id=39E788794B9XXXXXX
```

1. Deploy the image on the target cluster in a NEW_ORACLE_HOME dir.

- CURRENT version of the subscribed image will be selected

```
emcli db_software_maintenance -performOperation -name="Deploy 112049 GI Home" \
-purpose=DEPLOY_GI_SOFTWARE \
-target_type=cluster  -target_list=cluster123 \
-normal_credential=ORACLE:SYSMAN  -privilege_credential=ROOT:SYSMAN \
-new_oracle_home="/xyz/crs/112049"
```

FOR DB HOME*

1. For the given target, subscribe to an Image(_id).

```
emcli db_software_maintenance -subscribeTarget -target_name="somedb" -target_type=rac_database  -image_id=39E788794B91XXXX
```

1. Deploy the image on the target cluster in a NEW_ORACLE_HOME dir

```
emcli db_software_maintenance -performOperation -name="Deploy 112049 DB Home" \
-purpose=DEPLOY_DB_SOFTWARE  -target_type=rac_database  -target_list=somedb \
-normal_credential=ORACLE:SYSMAN  -privilege_credential=ROOT:SYSMAN \
-new_oracle_home="/xyz/db/112049"
```

OEM Fleet Management Flow -- switch / update

FOR GRID HOME*

1. Switch GRID_HOME from current to new gold (patched) previously deployed.

```
emcli db_software_maintenance -performOperation -name="Cluster123: Update GI Cluster"
-purpose=UPDATE_GI -target_type=cluster -target_list=cluster123
-normal_credential=ORACLE:SYSMAN   -privilege_credential=ROOT:SYSMAN
-node_list=nodex.paypal.com -startupDBAfterSwitch=false
```

FOR DB HOME*

1. Switch DB_HOME from current to new gold (patched) previously deployed.
2. Start-up DB
3. Run the required post patch SQL scripts **!!! Winner**

```
emcli db_software_maintenance -performOperation -name="nodex:Update DB Cluster"
 -purpose=UPDATE_RACDB -target_type=rac_database -target_list=somedb
 -normal_credential=oracle:SYSMAN -privilege_credential=ROOT:SYSMAN
 -node_list=nodex.paypal.com -dataguard_role=primary -ignoreStandbyPrereq=true
```

State of the Nation: benefits of Fleet Management

1. Patching time 20 min/ node
2. Different DB Clusters patched in Parallel
3. Target of <15 distinct configuration looks achievable
4. Much standardized and therefore stable site
5. Compliance Reporting/ Drift Management is easier

QUESTIONS?