

ORACLE

# Oracle Fleet Patching and Provisioning

Provision, Patch and Upgrade Oracle Grid and Oracle RAC Databases with Ease

---

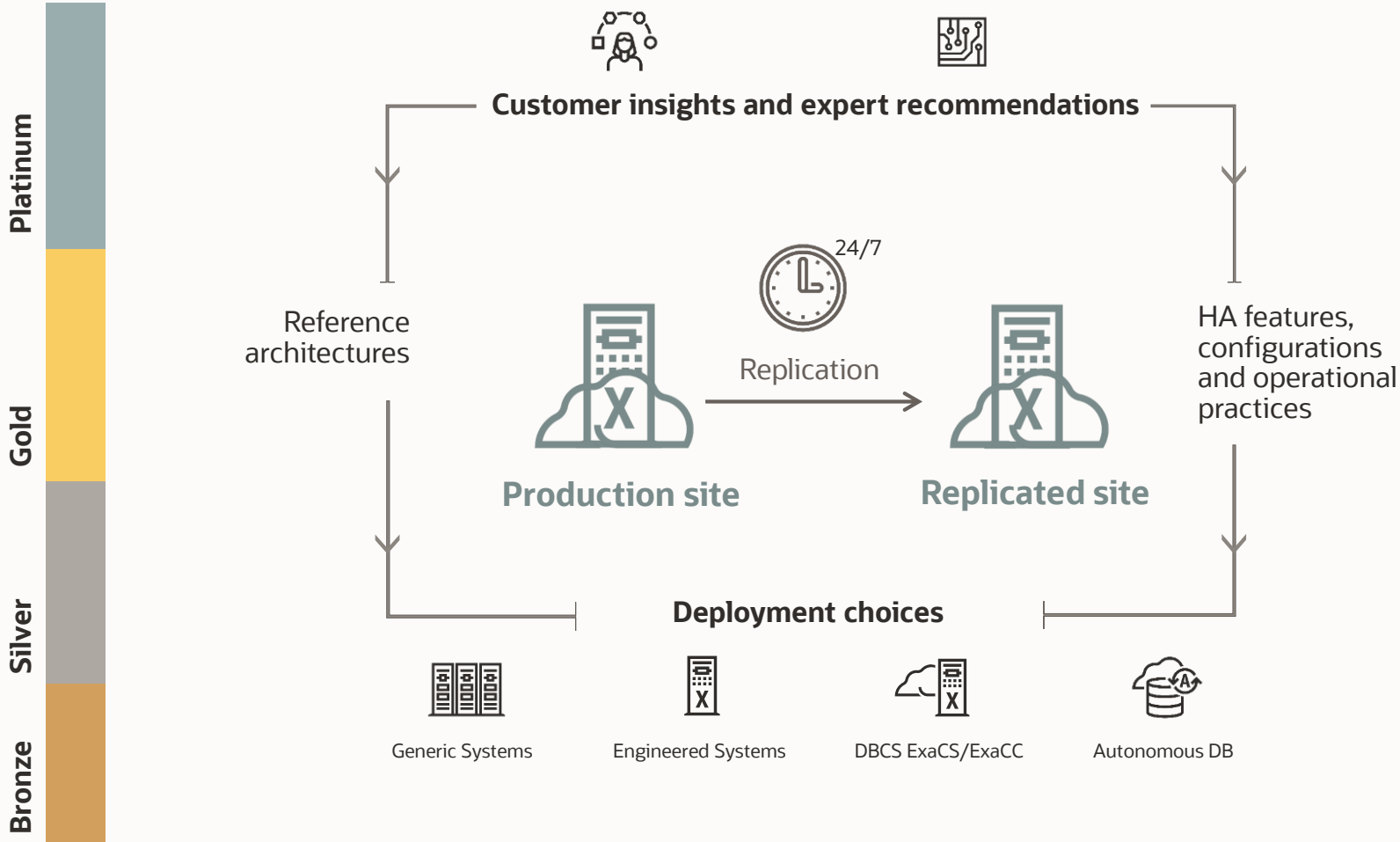
**Ludovico Caldara**

Principal Product Manager

Oracle Database High Availability (HA),  
Scalability and Maximum Availability Architecture (MAA)



# Oracle Maximum Availability Architecture (MAA)



**Continuous availability**

Application Continuity      Global Data Services

---

**Data protection**

Flashback      RMAN + ZDLRA

---

**Active replication**

Active Data Guard      GoldenGate

---

**Scale out**





RAC      ASM      Sharding



# MAA reference architectures

## Availability service levels

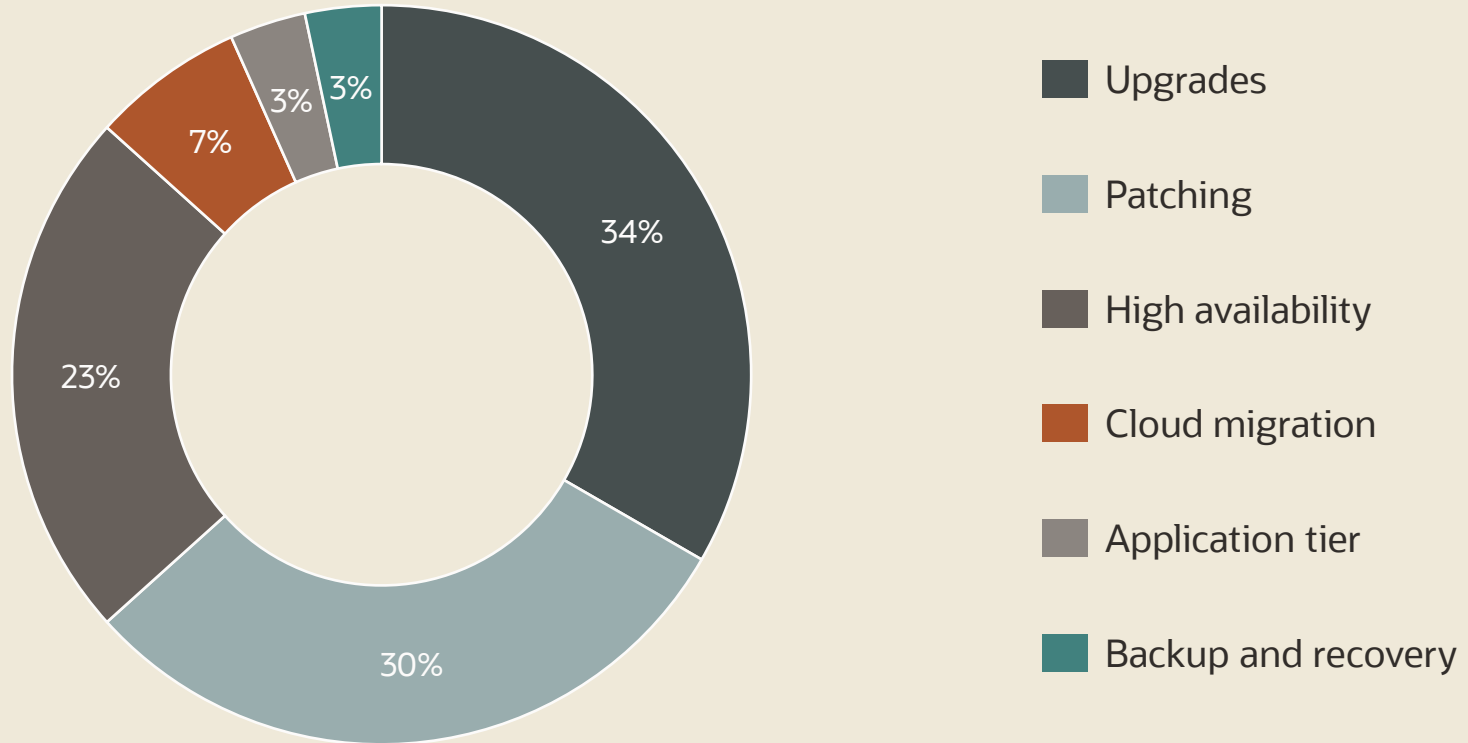


Bronze	Silver	Gold	Platinum
<b>Dev, test, prod</b>	<b>Prod/departmental</b>	<b>Business critical</b>	<b>Mission critical</b>
	<b>Bronze +</b>	<b>Silver +</b>	<b>Gold +</b>
Single instance DB	Database HA with RAC	DB replication with Active Data Guard	GoldenGate
Restartable	Application continuity		Edition-Based Redefinition
Backup/restore			
			

All tiers exist with on-premises and cloud. However, platinum currently must be configured manually while bronze to gold are covered with cloud tool automation for the most part depending on the desired RTO (i.e. FSFO and multiple standby databases still must be manually configured for example)



# Top pain points\*



\*MAA Customer Summit 2019 survey results

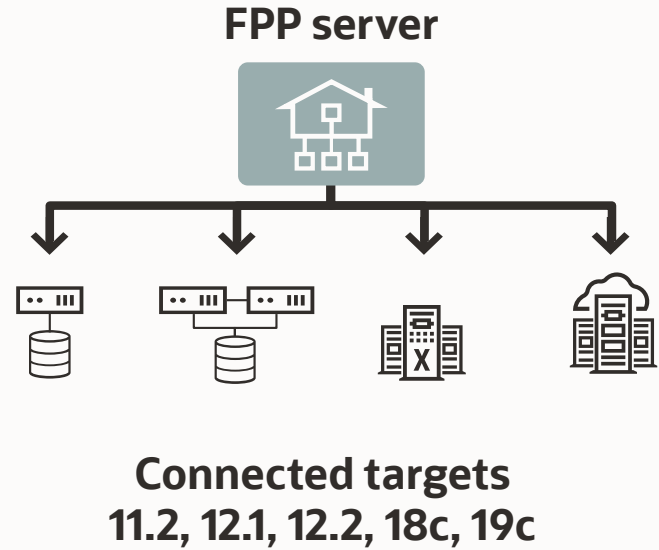


# Overview



**Gold image repository**

- 11.2.0.4
- 12.1.0.2
- 12.2.0.1
- 19.3.0
- 19.9.0



- Simple
- Enables mass operations on 1000s of nodes
- Prevents errors, enable easy corrections
- Ship only once/subscriber base
- Target operations aware

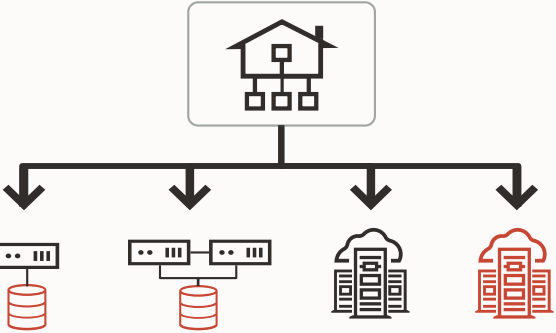


# Fleet patching and provisioning support

## Database, GoldenGate and grid infrastructure



- Single instance
- Oracle restart
- Oracle RAC one
- Oracle RAC



- Generic Software
- Customizable

## Multi-OS



# A rich set of features for your database fleet

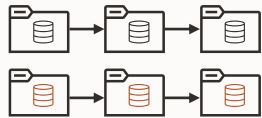
## Database and grid infrastructure home management



Install



Remove

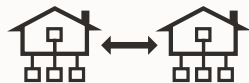


Series

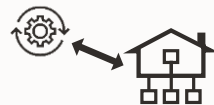


Drift detection

## Integration



FPP peering



REST APIs

## Oracle Database (SI, RAC, RACON)



Provision



Patch



Upgrade

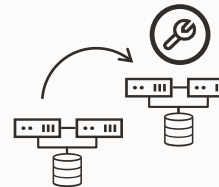


Scale

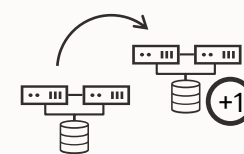
## Oracle Grid Infrastructure



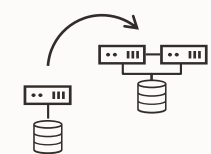
Provision



Patch



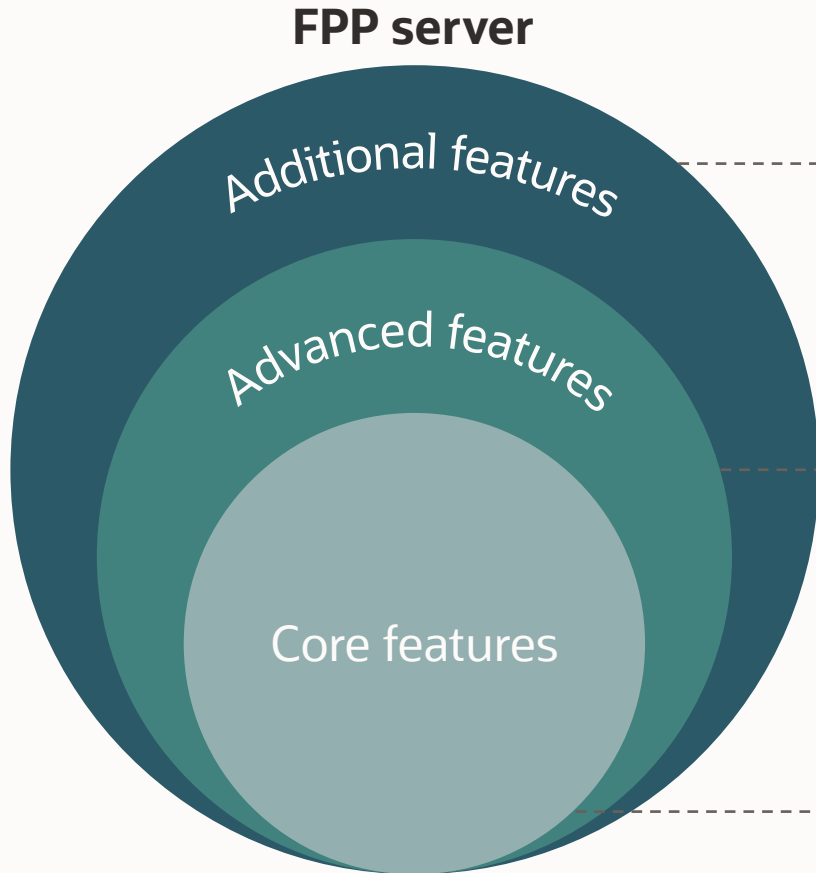
Upgrade



Scale



# Feature overview



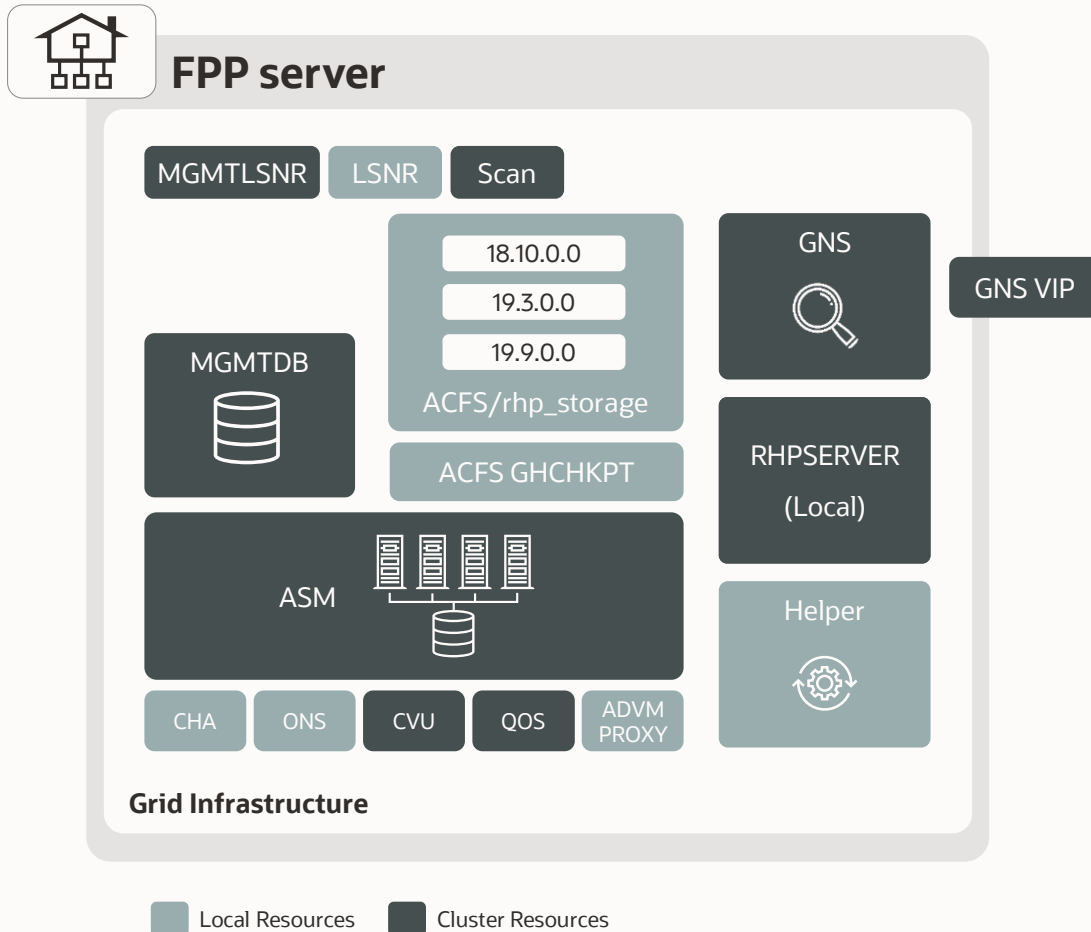
- Zero downtime upgrade
- Server peering
- REST APIs
- Zero downtime GI patching

- Database provisioning
- Grid infrastructure provisioning
- Database upgrade
- Grid infrastructure upgrade
- Custom user actions

- Oracle home management
- Gold image management
- Gold image series
- Database patching
- Grid infrastructure patching
- Drift detection



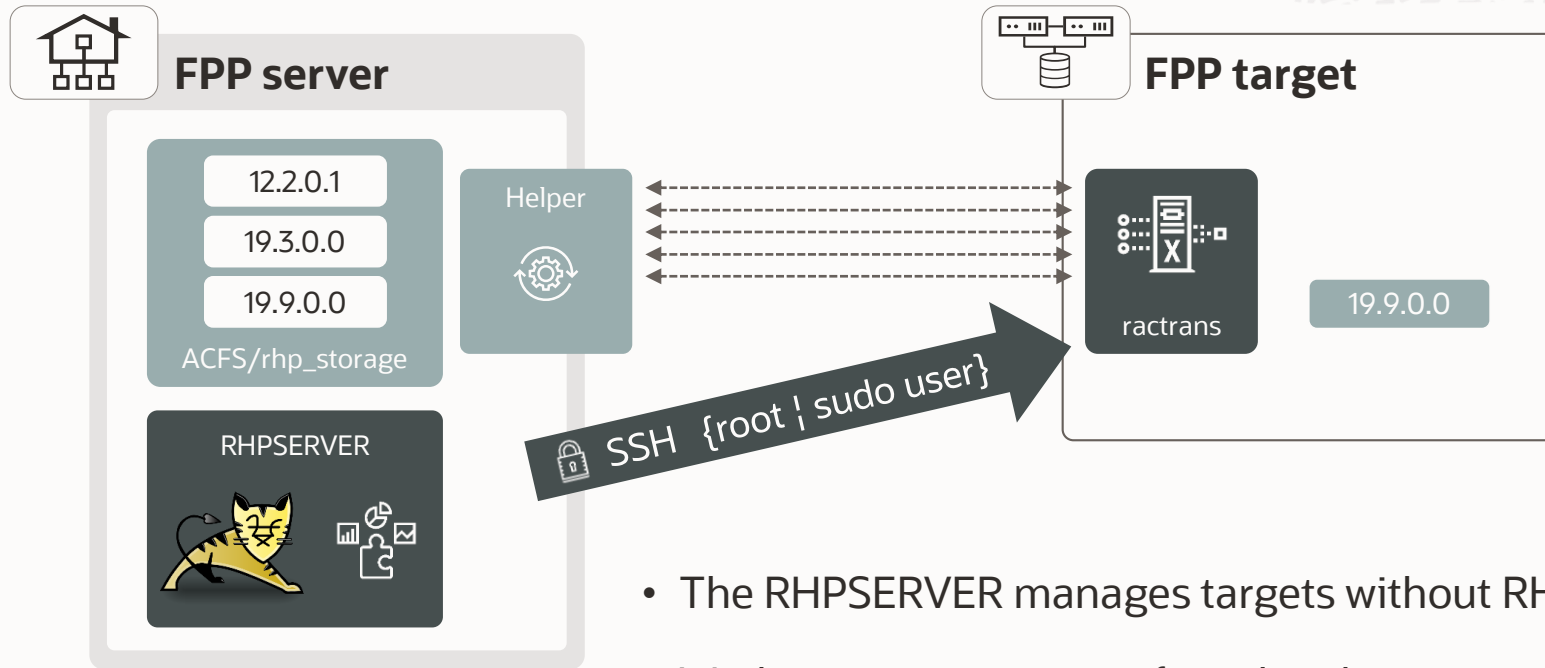
# FPP server architecture



- Server installed, ready to host Grid Infrastructure  
Required firewall ports are open between FPP server and targets
- Install Grid Infrastructure 19c **with the MGMTDB**  
Note: If MGMTDB configuration is skipped (default in 19c), additional steps are required
- Remove the local automaton  
`# srvctl remove rhpserver -f`
- Configure the GNS  
`# srvctl add gns -vip x.x.x.x`
- Configure and start the RHPSERVER  
`# srvctl add rhpserver -storage /rhp -diskgroup data`  
`$ srvctl start rhpserver`
- Start working with RHPCTL!  
`$ rhpctl import image -image DB193_Base \  
-zip /tmp/LINUX.X64_193000_db_home.zip \  
-imagetype ORACLEDBSOFTWARE`

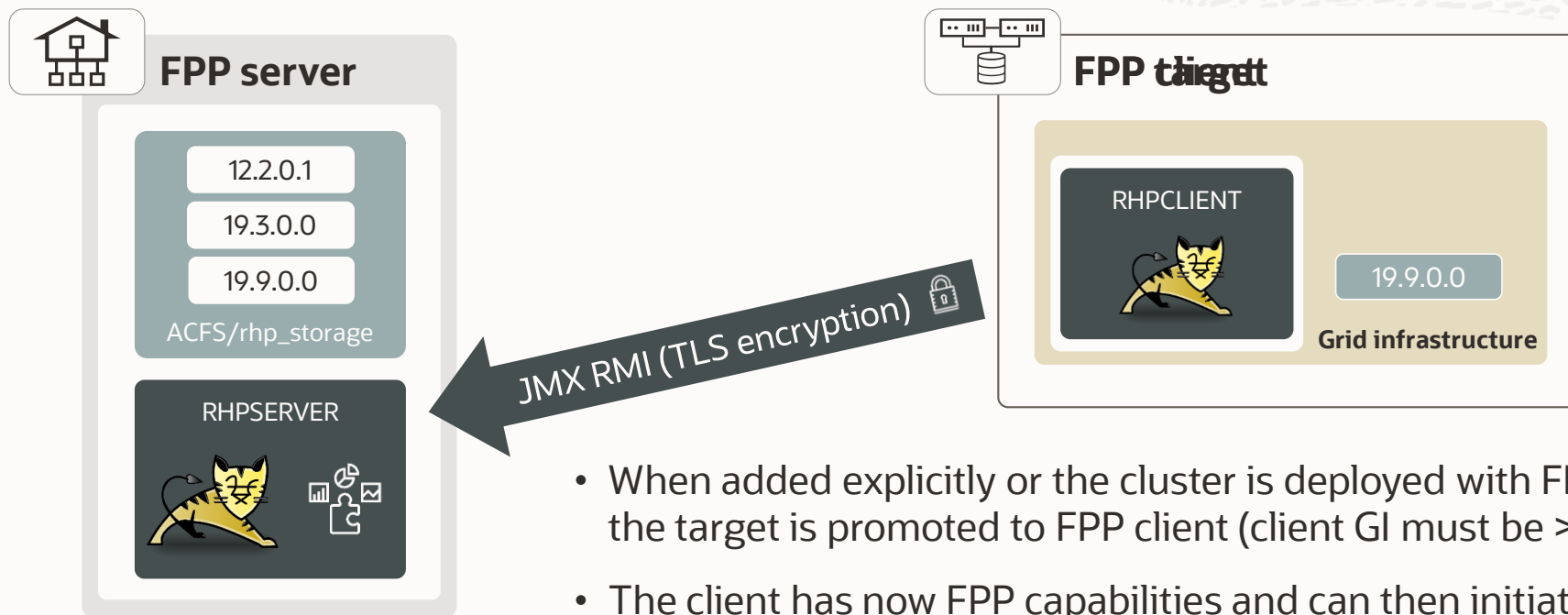


# FPP targets without RHPCLIENT



- The RHPSERVER manages targets without RHPCLIENT through SSH
- Working copies are transferred to the target using «ractrans»  
`$ rhpctl add workingcopy -image ... -path ... -workingcopy ... -targetnode ... -root`
- The progress is tracked thanks to a listener on the FPP Server.

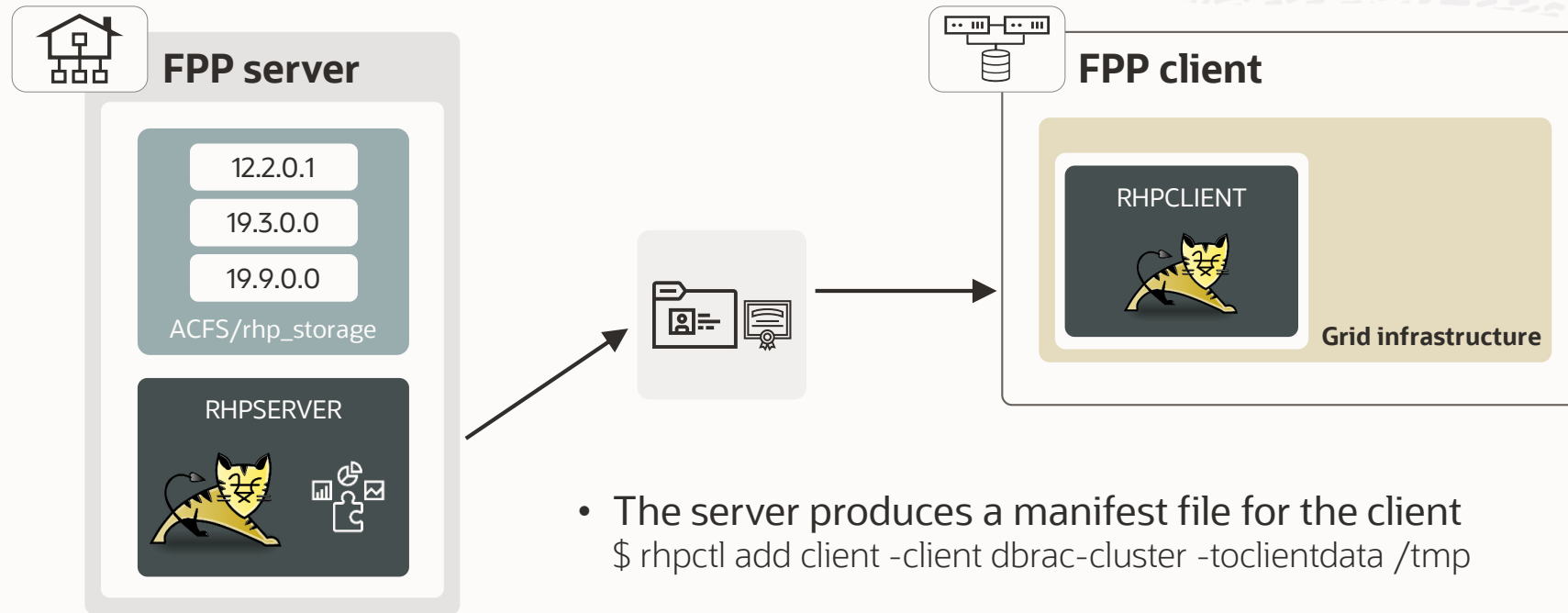
# FPP clients



- When added explicitly or the cluster is deployed with FPP, the target is promoted to FPP client (client GI must be  $\geq 12.2$ )
- The client has now FPP capabilities and can then initiate its own operations
- FPP client and server communicate through TLS encrypted JMX:RMI. SSH is not needed anymore. File transfer via “ractrans”.



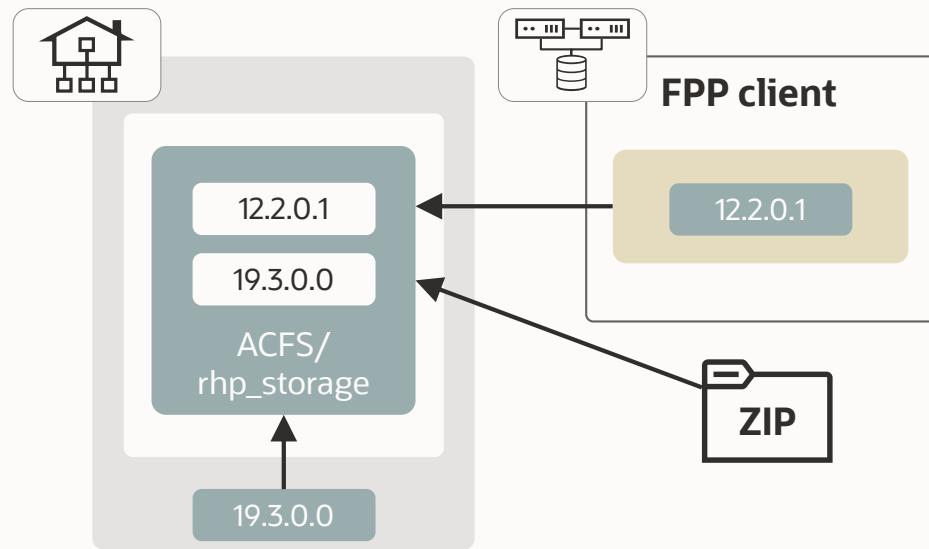
# Adding FPP clients



- The server produces a manifest file for the client  
`$ rhpctl add client -client dbrac-cluster -toclientdata /tmp`
- It must be copied on the client, which uses it to connect to the server  
`$ srvctl add rhpclient -clientdata /tmp/dbrac-cluster-cluster.xml  
[ -diskgroup ... -storage ... ]`

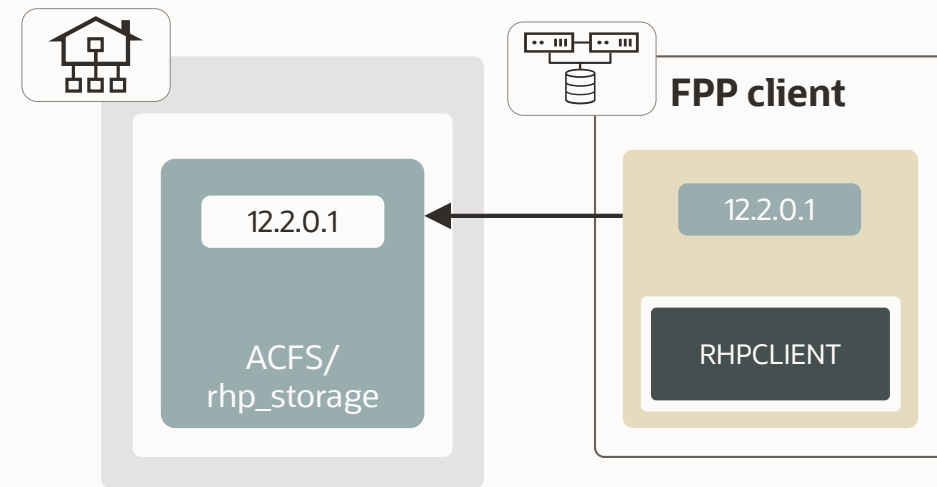
# Importing images (import vs. add)

## rhpcctl import image



- From zip file
- From existing unmanaged home (local or remote)

## rhpcctl add image



- From existing provisioned working copies



# Gold image store on the FPP server

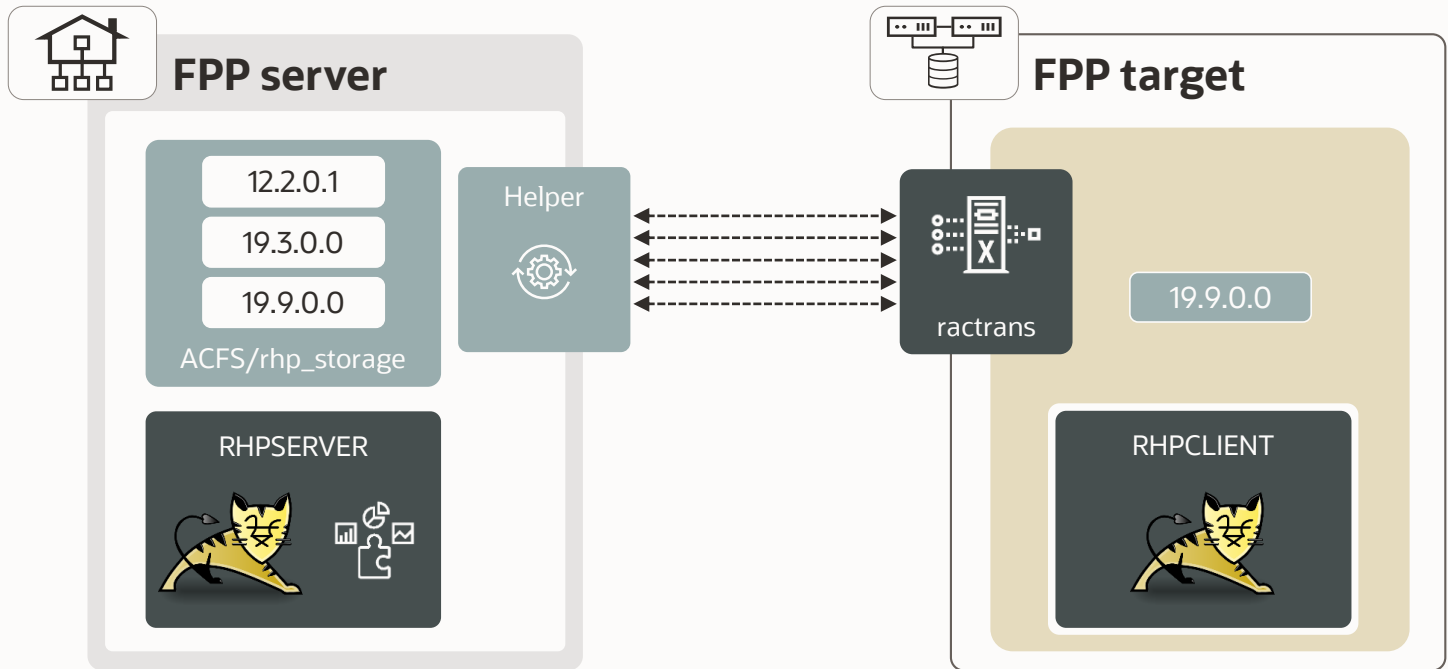


```
$ rhpctl query image
fpps.vcnfpppubsubnet.vcnfpppub.oraclevcn.com: Audit ID: 28
Image name: db_19_9_0_oci

$ df -k /rhp_storage/images/idb_19_9_0_oci844354
Filesystem          1K-blocks    Used Available Use% Mounted on
/dev/asm/ghvol193693-42 25165824 13125480 12040344 53% /rhp_storage/images/idb_19_9_0_oci844354

$ acfsutil snap info /rhp_storage/images/idb_19_9_0_oci844354
snapshot name:          idb_19_9_0_oci
snapshot location:     /rhp_storage/images/idb_19_9_0_oci844354/.ACFS/snaps/idb_19_9_0_oci
RO snapshot or RW snapshot: RO
parent name:          /rhp_storage/images/idb_19_9_0_oci844354
snapshot creation time: Mon Dec 7 13:59:53 2020
file entry table allocation: 201588736 ( 192.25 MB )
storage added to snapshot: 201588736 ( 192.25 MB )
```

# Adding workingcopies with LOCAL storagetype



- Filesystem existence and size are not managed by FPP
- The Oracle Home will be on a local filesystem (must provision on all cluster nodes)
- Whether client (JMX) or not (SSH), the transfer is done via ractrans.
- 6 ports needed, configurable with: `srvctl modify rhpserver - port_range <range>`



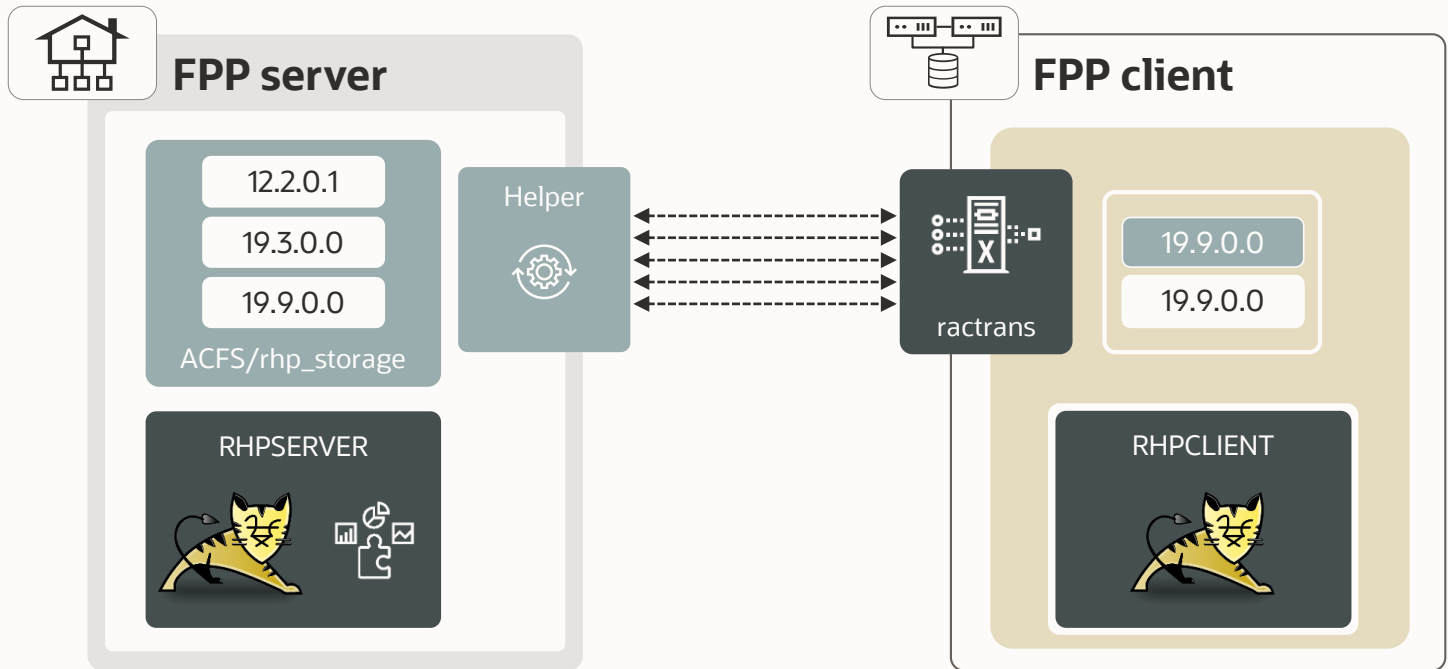
# Adding workingcopies with LOCAL storagetype

```
rhpcctl add workingcopy -image db_19_9_0_oci -storagetype LOCAL \  
-user oracle -oraclebase /u01/app/oracle -client dbSys67uwrlqq \  
-workingcopy WC_db_19_9_0_oci_FPFC1 -path /u01/app/oracle/product/19.9.0.0/db_oci
```

```
<INVENTORY>  
<VERSION_INFO>  
  <SAVED_WITH>12.2.0.7.0</SAVED_WITH>  
  <MINIMUM_VER>2.1.0.6.0</MINIMUM_VER>  
</VERSION_INFO>  
<HOME_LIST>  
<HOME NAME="OraGrid190" LOC="/u01/app/19.0.0.0/grid" TYPE="0" IDX="1" CRS="true"/>  
<HOME NAME="WC_db_19_9_0_oci_FPFC1" LOC="/u01/app/oracle/product/19.9.0.0/db_oci" TYPE="0"  
IDX="4"/>  
</HOME_LIST>  
<COMPOSITEHOME_LIST>  
</COMPOSITEHOME_LIST>  
</INVENTORY>
```



# Adding workingcopies with RHP\_MANAGED storagetype



- Target filesystem is on ACFS, managed by FPP
- A new filesystem is created, it will host a read-only copy of the image
- The image is transferred with ractrans
- A RW snapshot is created, the home is attached to the inventory

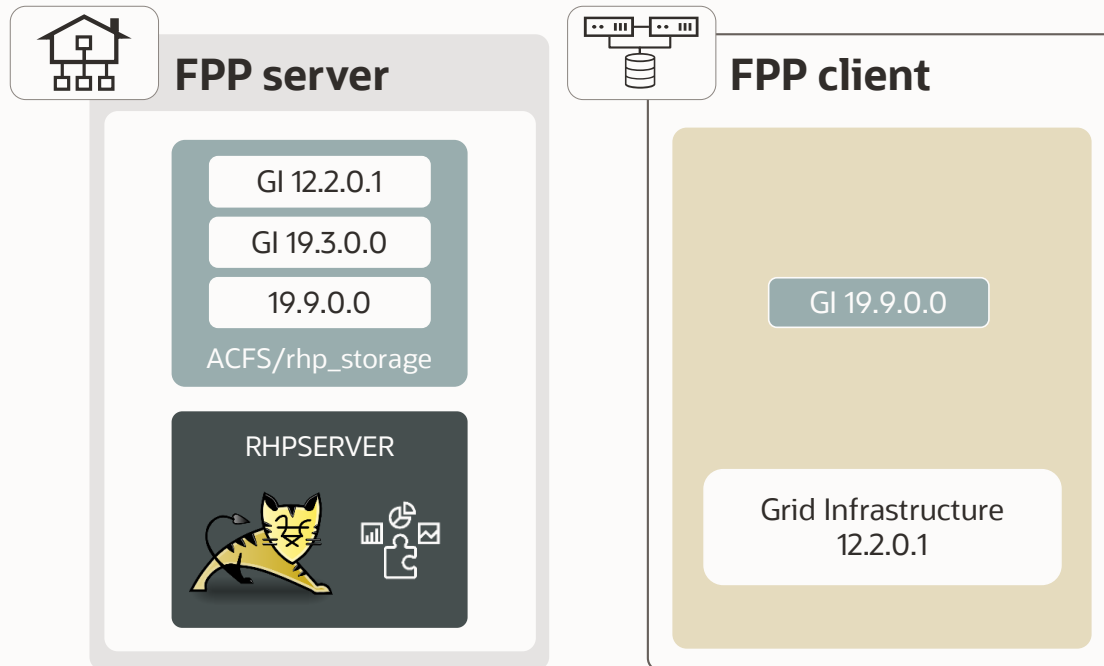


# Adding workingcopies with RHP\_MANAGED storagetype

```
rhpcctl add workingcopy -image db_19_9_0_oci -storagetype RHP_MANAGED \  
  -user oracle -oraclebase /u01/app/oracle -client dbSys67uwrlqq \  
  -workingcopy WC_db_19_9_0_oci_FPFC1_RHP
```

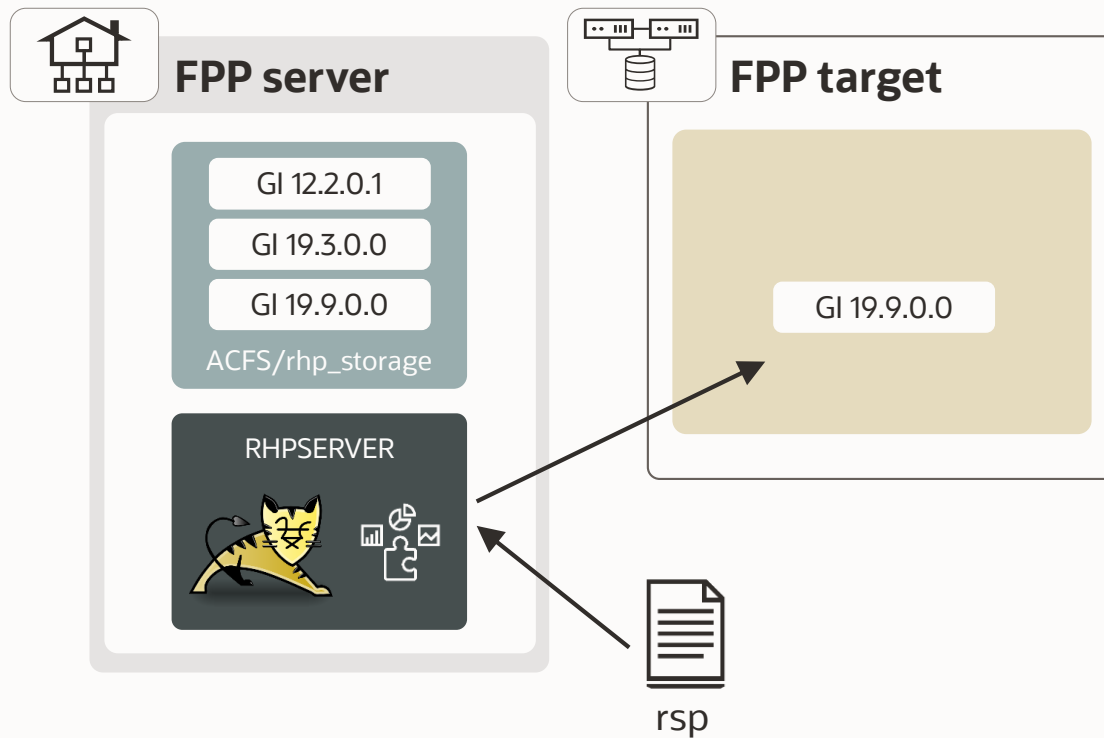
```
<INVENTORY>  
<VERSION_INFO>  
  <SAVED_WITH>12.2.0.7.0</SAVED_WITH>  
  <MINIMUM_VER>2.1.0.6.0</MINIMUM_VER>  
</VERSION_INFO>  
<HOME_LIST>  
<HOME NAME="OraGrid190" LOC="/u01/app/19.0.0.0/grid" TYPE="0" IDX="1" CRS="true"/>  
<HOME NAME="WC_db_19_9_0_oci_FPFC1_RHP"  
  LOC="/rhp_storage/images/idb_19_9_0_oci/.ACFS/snaps/wWC_db_19_9_0_oci_FPFC1_RHP/swhome"  
  TYPE="0" IDX="3"/>  
</HOME_LIST>  
<COMPOSITEHOME_LIST>  
</COMPOSITEHOME_LIST>  
</INVENTORY>
```

# Adding grid infrastructure workingcopy to an existing server/cluster



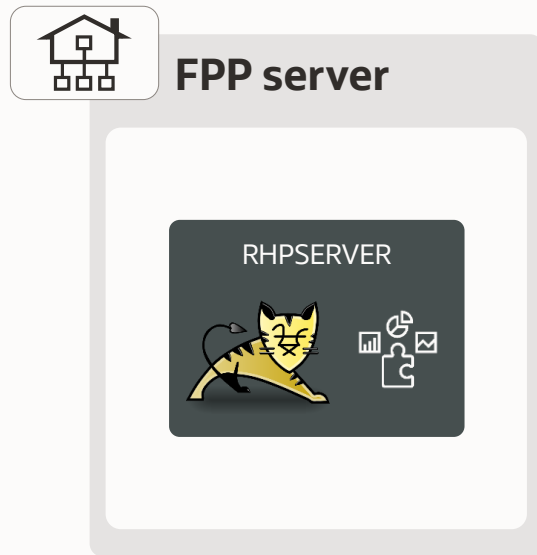
- GI working copies can only be LOCAL
- GI Software copy works like database software copies
- FPP detects users and groups and assign correct ownership
- A GI stack already exists, the install is «software\_only»

# Adding grid infrastructure workingcopy to a new server/cluster



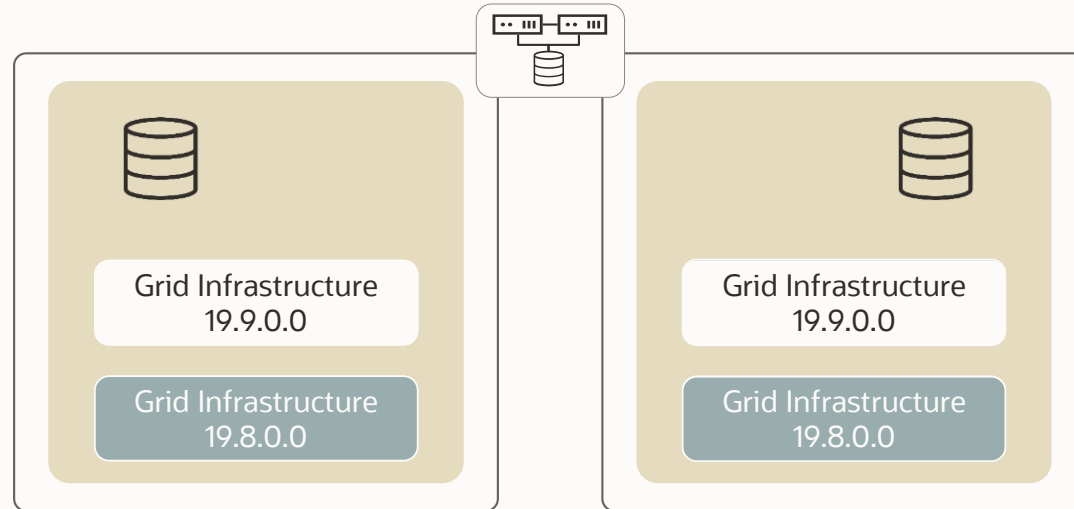
- GI Software is copied
- A responsfile can be provided to configure the cluster
- FPP takes care of installing and configuring the cluster

# Grid infrastructure patching



- Always rolling
- Serial/parallel
- User-defined batch/smart
- Only GI/GI and DB

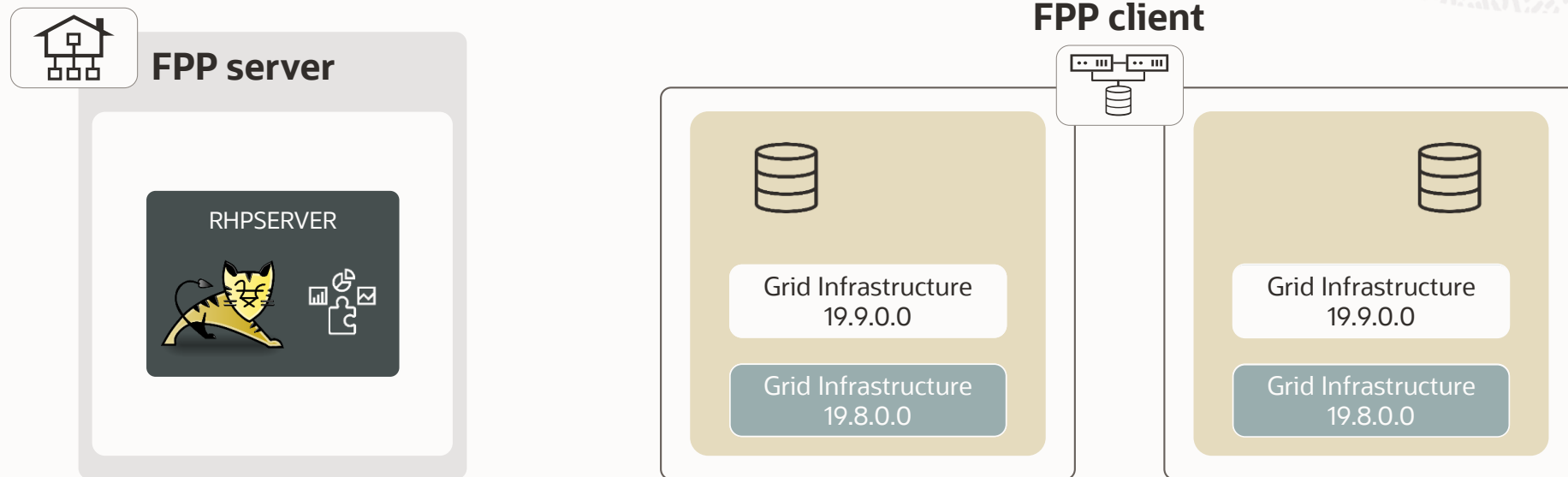
## FPP client



```
rhpcctl move gihome \  
-sourcewc WC_gi19800_c11 \  
-destwc WC_gi19900_c11 \  
-keepplacement
```



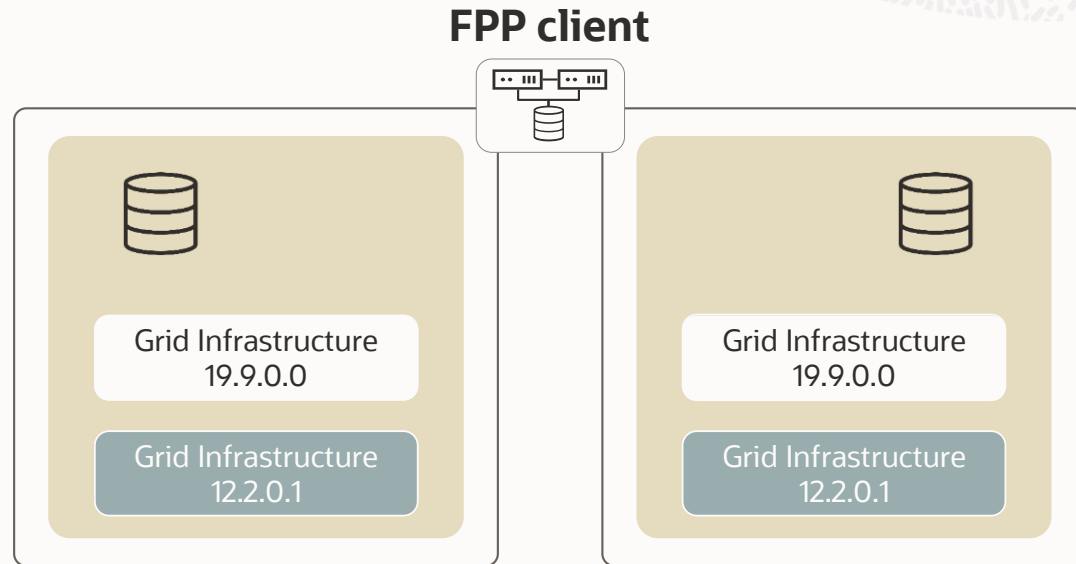
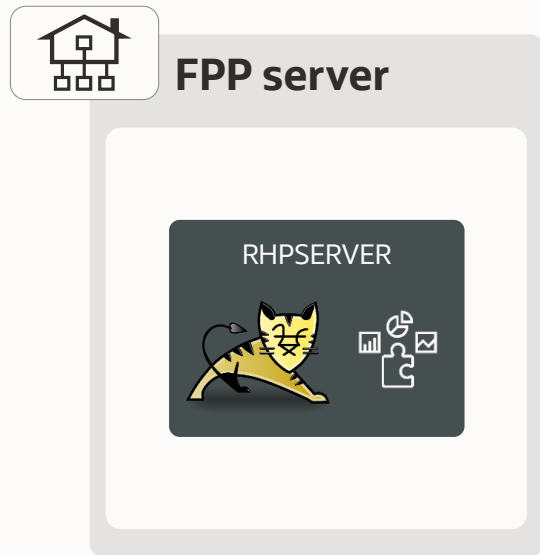
# Zero downtime grid infrastructure patching (TGIP)



- >=19.5 feature
- Some bugs in early versions
- Practically requires GI and DB >=19.8
- AFD and ACFS drivers: deferred restart

```
rhctl move gihome \  
-sourcewc WC_gi19800_c11 \  
-destwc WC_gi19900_c11 \  
-tgip
```

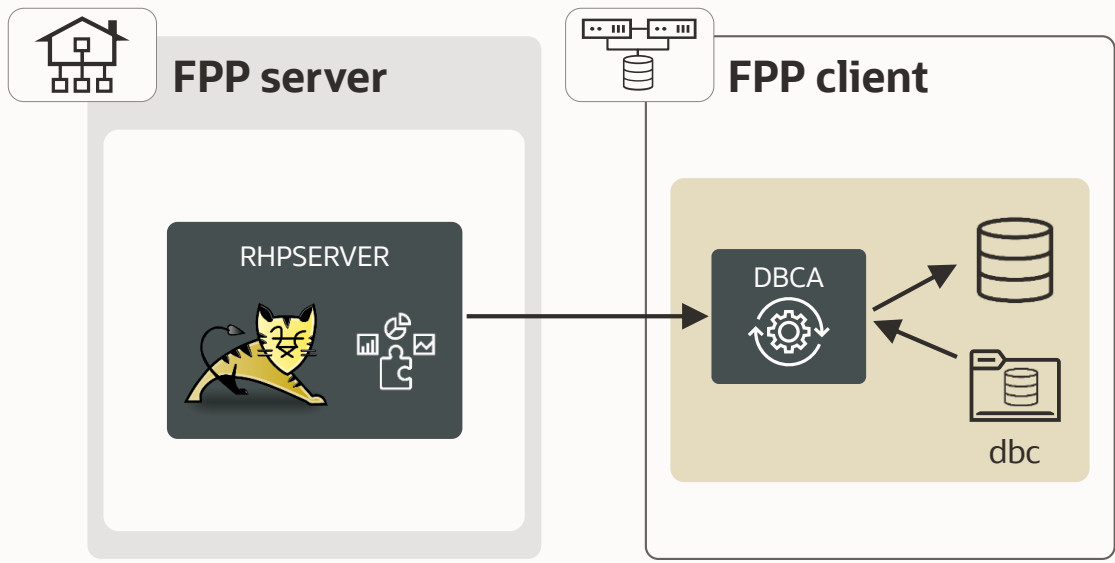
# Grid infrastructure upgrade



```
rhpcctl upgrade gihome \  
-sourcewc WC_gi12201_cl1 \  
-destwc WC_gi19900_cl1
```



# Provisioning databases



- FPP can provision SINGLE, RAC, RACONENODE databases to FPP Clients
- It executes database creation assistant (DBCA)
- Template files must exist either in the Gold Image or locally on the FPP Client

```
rhptl add database -workingcopy <workingcopy> \  
-dbname <dbname> ... \  
-dbtemplate <template_file>
```



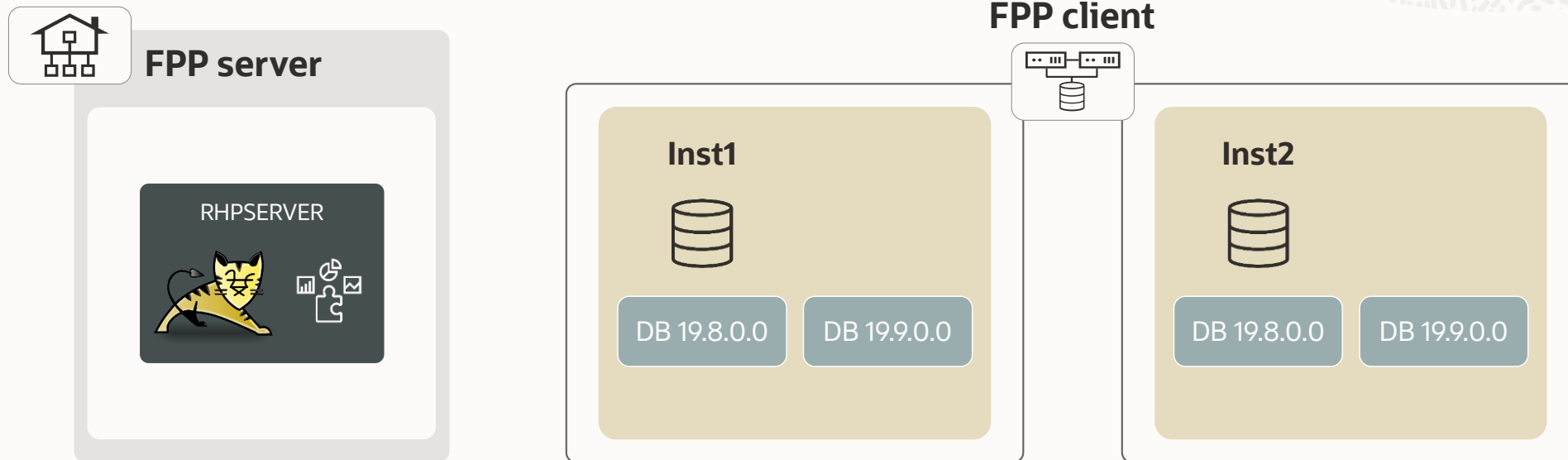


# Provisioning databases

```
rhpcctl add database \  
-workingcopy WC_db_19_9_0_oci_FPPC1_RHP \  
-dbname racldb2_fra1nn \  
-datafileDestination DATA \  
-node fppc1 \  
-dbtype RAC \  
-cdb \  
-dbtemplate db_19_9_0_oci:assistants/dbca/templates/seed_db.dbc
```

```
$ ls -tr /u01/app/oracle/cfgtoollogs/dbca/racldb2_fra1nn  
initracldb2frTempOMF.ora.1115202092759 cloneDBCreation.log catclust_catcon_77650.lst  
racldb2_fra1nn.log rmanUtil CreateClustDBViews.log  
trace.log_2020-12-14_05-55-05PM plugDatabase.log lockAccount.log  
initracldb2frTempOMF.ora.1115202094834 ordlib0.log utlrp0.log  
rmanDeleteFiles.sql ordlib_catcon_75303.lst utlrp_catcon_85815.lst  
racldb2_fra1nn0.log execemx0.log postDBCcreation.log  
trace.log_2020-12-15_09-25-26AM execemx_catcon_76689.lst racldb2_fra1nn1.log  
tempControl.ctl postScripts.log trace.log_2020-12-15_09-45-48AM  
CloneRmanRestore.log catclust0.log
```

# Database patching

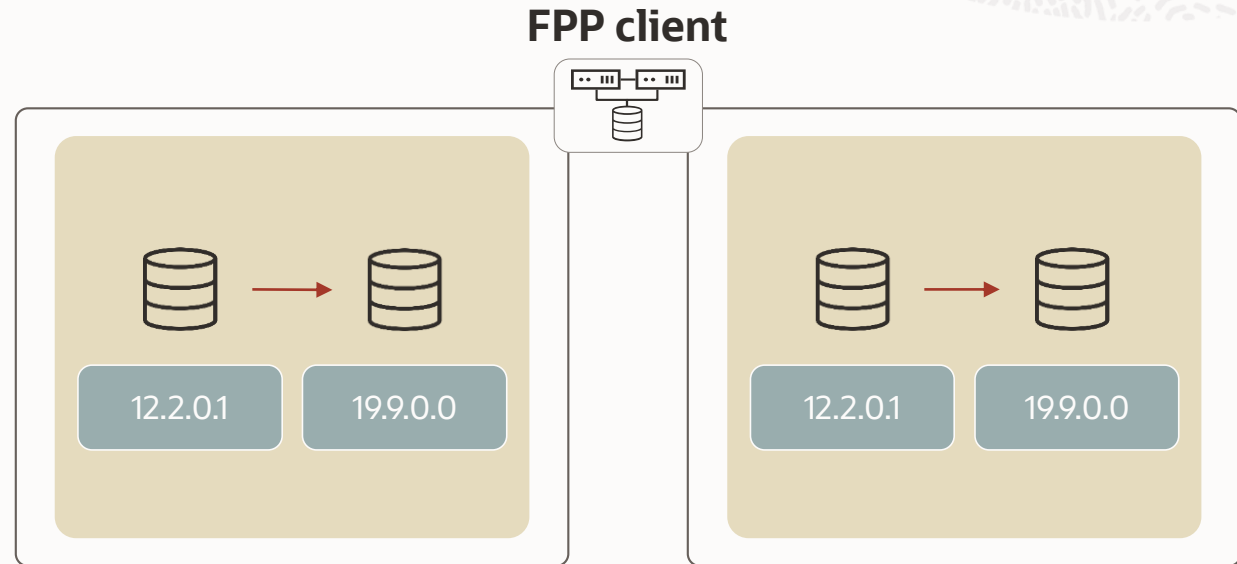
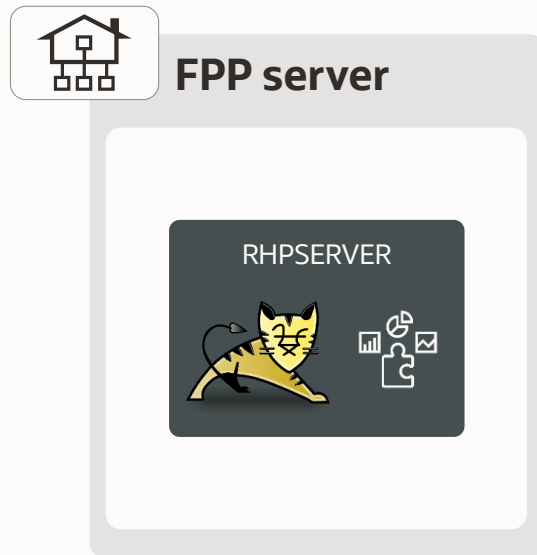


```
rhpctl move database \  
-sourcewc WC_db19800_c11 \  
-patchedwc WC_db19900_c11
```

- Rolling by default
- Automated datapatch execution
- Service Drain Timeout honored



# Database upgrades



```
rhpctl upgrade database \  
-dbname racldb2_fra1nn \  
-sourcewc WC_db12201_c11 \  
-destwc WC_db19900_c11
```

- It executes database upgrade assistant (DBUA)
- Needs careful planning (pre-checks, pre-fixups)
- ⚠️ As bullet-proof as DBUA can be
- Autoupgrade can be used starting with FPP 21c

# Image series



- Ordered list of images
- Users can subscribe and get notified via e-mail about new images
- Order is useful to get the latest image to patch the DBs



## FPP server

series\_db\_19c

19.3.0.0

19.8.0.0

19.9.0.0

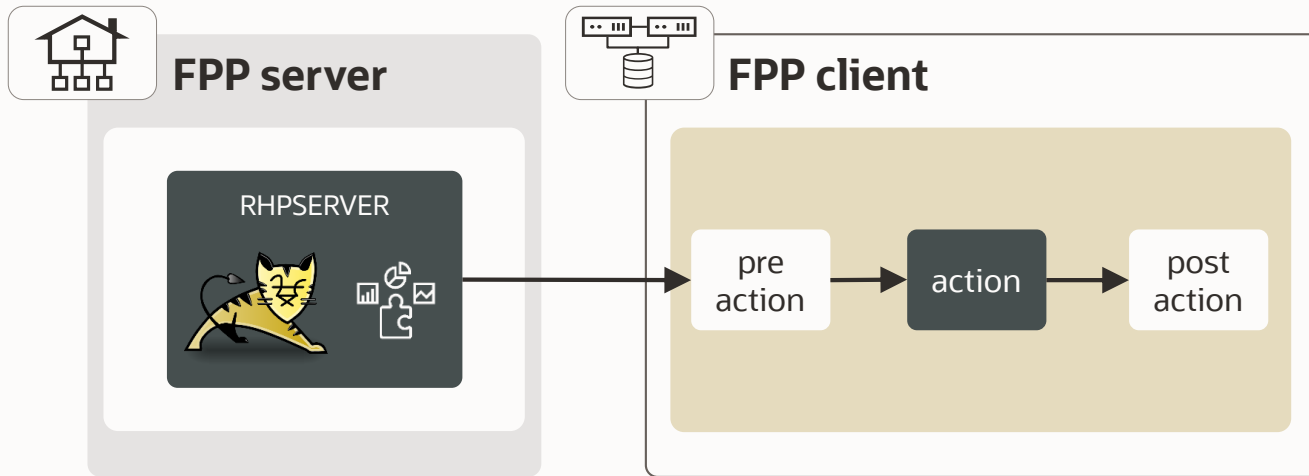
```
rhpcctl add series -series series_db_19c
```

```
rhpcctl insertimage series -series series_db_19c -image db_19_3_0
```

```
rhpcctl insertimage series -series series_db_19c -image db_19_8_0
```

```
rhpcctl insertimage series -series series_db_19c -image db_19_9_0
```

# User actions



- Pre or Post actions
- On one or all nodes
- On server or client

```
rhptcl add useraction -post -optype ADD_DATABASE \  
-onerror CONTINUE \  
-useraction action_post_add_database \  
-actionsript /var/opt/dbascripts/action_post_add_database.sh \  
-runscope ONENODE
```



# User actions: example script

```
#!/bin/sh
# convert parameters to variables
for i in "$@" ; do
    export $i
done

L_CRG_HOME=$(cat /etc/oracle/olr.loc | grep crs_home | awk -F= '{print $2}')
L_OH=$(($L_CRG_HOME/bin/rhpctl query workingcopy -workingcopy $RHP_WORKINGCOPY -metadataonly | grep "Software
home path:" | awk '{print $NF}')
L_HOSTNAME=$(($L_CRG_HOME/bin/olsnodes -l)
L_SID=$(echo $RHP_DBNAME | cut -c 1-8)

# add EMCC target
/var/opt/dbascripts/emcli/emcli add_target -name="$RHP_DBNAME" \
    -type="oracle_database" -host="$L_HOSTNAME" \
    -credentials="UserName:dbsnmp;password:secret;Role:Normal" \
    -properties="SID:$L_SID;Port:1521;OracleHome:$L_OH"

# register in RMAN catalog
export ORACLE_HOME=$L_OH
$L_OH/bin/rman target / catalog rman/secret@rcvcat <<EOF
    register database;
    exit
EOF
```

# Drift reporting

```
$ rhpctl query image -image db_19_9_0_giaas -drift
fpps.vcnfpppubsubnet.vcnfpppub.oraclevcn.com: Audit ID: 89
Image name: db_19_9_0_giaas
Owner: grid@dbSysxcfxydga
Site: dbSysxcfxydga
Access control: USER:grid@dbSysxcfxydga
Access control: ROLE:OTHER
Access control: ROLE:GH_IMG_PUBLISH
Access control: ROLE:GH_IMG_ADMIN
Access control: ROLE:GH_IMG_VISIBILITY
Parent Image:
Software home path: /rhp_storage/images/idb_19_9_0_giaas14118/.ACFS/snaps/idb_19_9_0_giaas/swhome
Image state: PUBLISHED
Image size: 8286 Megabytes
Image Type: ORACLEDBSOFTWARE
Image Version: 19.0.0.0.0:19.9.0.0.0
Groups configured in the image: OSDBA=dba,OSOPER=oper,OSBACKUP=backupdba,OSDG=dgdba,OSKM=kmdba,OSRAC=racdba
Image platform: Linux_AMD64
Interim patches installed: 31772784,31771877
Contains a non-rolling patch: FALSE
Complete: TRUE
Additional bug fixes in working copies based on this image: 16662822
```

# Audit

```
$ rhpctl query audit -operation add -entity workingcopy
```

```
Audit ID: 37
```

```
Start time: 2020-12-11T15:35:25.852
```

```
Command executed: rhpctl add workingcopy -image db_19_9_0_oci -storagetype LOCAL -workingcopy  
WC_db_19_9_0_oci_FPPC1 -user oracle -oraclebase /u01/app/oracle -client dbSys67uwrlqq -path  
/u01/app/oracle/product/db_19_9_0_oci
```

```
End time: 2020-12-11T15:42:35.000
```

```
Command result: SUCCESS
```

```
User name: grid
```

```
Node name: fpps@dbSysxcfydga
```

```
Target cluster: dbSys67uwrlqq
```

```
Audit ID: 47
```

```
Start time: 2020-12-14T15:49:40.166
```

```
Command executed: rhpctl add workingcopy -image db_19_9_0_giaas -storagetype RHP_MANAGED -workingcopy  
WC_db_19_9_0_giaas_FPPC1_RHP -user oracle -oraclebase /u01/app/oracle -client dbSys67uwrlqq
```

```
End time: 2020-12-14T16:01:03.000
```

```
Command result: SUCCESS
```

```
User name: grid
```

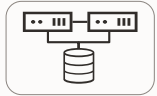
```
Node name: fpps@dbSysxcfydga
```

```
Target cluster: dbSys67uwrlqq
```

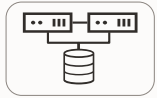


# Server peering

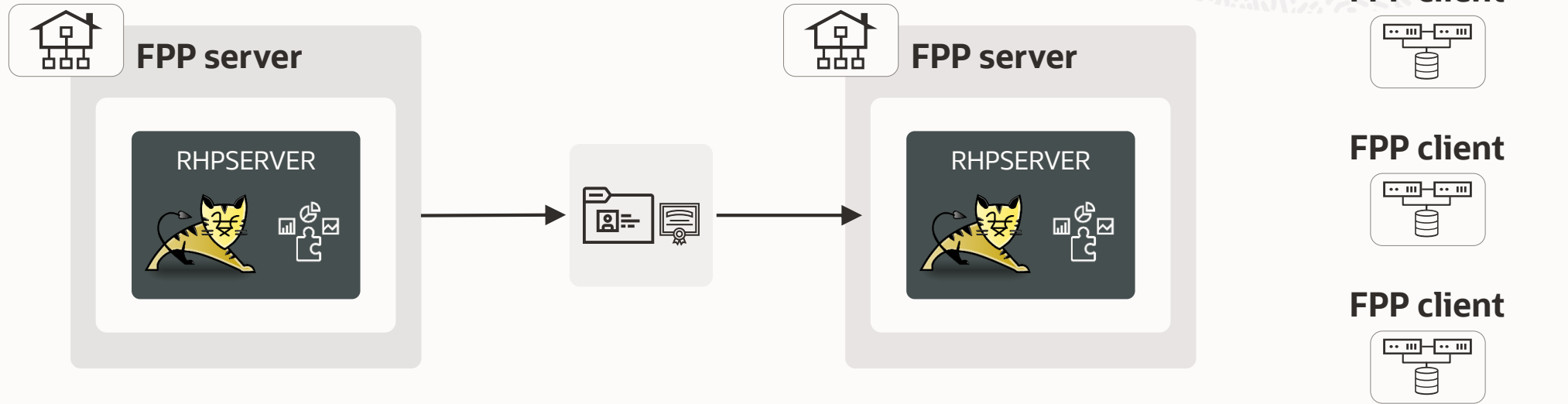
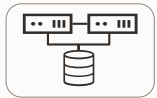
FPP client



FPP client

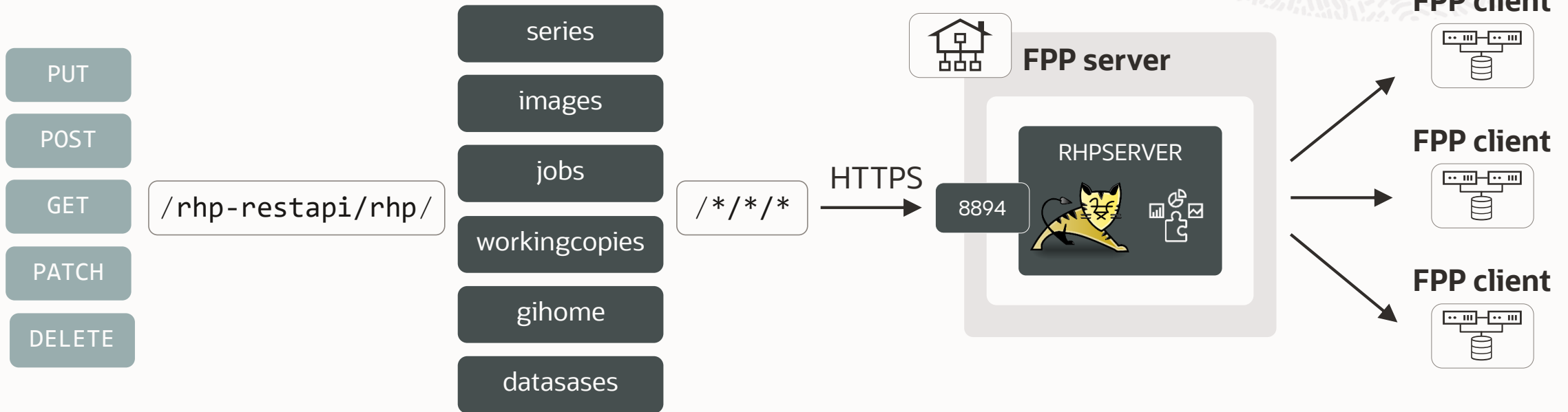


FPP client



- The first server produces its manifest file  
(1)\$ `rhptl export server -serverdata /tmp/server1.xml`
- The second server uses it to peer with the first one  
(2)\$ `rhptl register server -server server1 -serverdata /tmp/server1.xml -root`
- Images are transferred between peer servers to have a uniform image repository across regions

# REST APIs

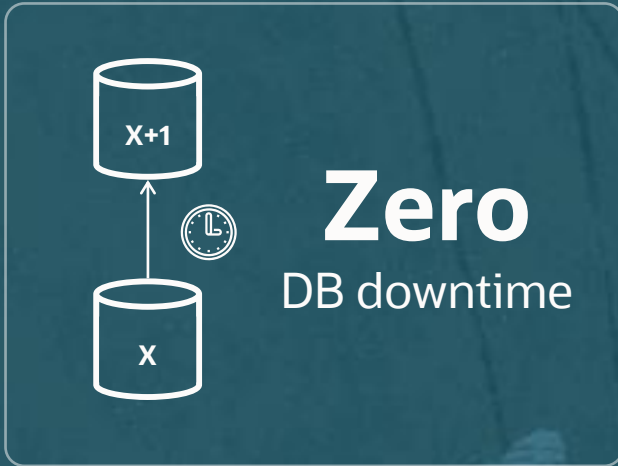


- Integrate FPP with Ansible, APEX, Rundeck, custom applications
- REST API documentation **REST APIs for Oracle Database** book  
<https://docs.oracle.com/en/database/oracle/oracle-database/19/dbrst/>



# Zero downtime upgrade (ZDU)

## Zero DB downtime upgrade



## Fully automated workflow

Clone

Switch over

Upgrade

Sync up

Switch back

## Robust, fast and space efficient

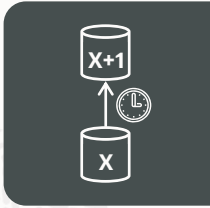
### Robust

- Resume-able after failures
- Revertible for fast rollback/fall back.

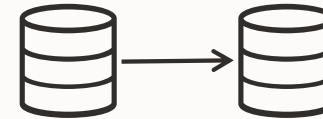
### Fast and space efficient

- Fast cloning with snapshots for databases hosted on ACFS
- Full database copy also supported

# Zero downtime upgrade (ZDU)



GoldenGate based synchronization and switchover



Data Guard transient logical standby (TLS) based synchronization and switchover

End-to-end automation

No extra hardware needed

Fleet level capability

Automatic setup/teardown and configuration of GG and DG. No prior knowledge required.



# Additional information



Oracle fleet patching and provisioning landing page

<https://www.oracle.com/goto/fpp>

Fleet patching and provisioning and maintenance 19c documentation

<https://docs.oracle.com/en/database/oracle/oracle-database/19/cwadd/rapid-home-provisioning.html>

Technical brief: Oracle fleet patching and provisioning (FPP) introduction and technical overview

<https://www.oracle.com/a/tech/docs/fpp-technical-brief-19c.pdf>

Fleet patching and provisioning and maintenance 18c cookbook

<https://www.oracle.com/technetwork/database/database-cloud/private/rapid-home-provisioning-2405191.pdf>

FPP vagrant provisioning for VirtualBox and KVM

[https://blogs.oracle.com/db\\_maintenance/oracle-fleet-patching-and-provisioning-meets-vagrantvirtualbox](https://blogs.oracle.com/db_maintenance/oracle-fleet-patching-and-provisioning-meets-vagrantvirtualbox)



# Thank you

---



ORACLE

Our mission is to help people see  
data in new ways, discover insights,  
unlock endless possibilities.

