



The long-requested Data Guard Feature is *planned to be* there in 21c

Data Guard per Pluggable Database (DGPDB)

Francisco Munoz Alvarez

Distinguished Product Manager

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



@fcomunoz



<http://www.linkedin.com/in/franciscomunozalvarez>



www.oraclemaa.com



ORACLE

Agenda

- Discovering Data Guard
- Non-CDB to CDB to PDB
- DGPDB Architecture
- Future Direction?
- Conclusion

Impact of database downtime



\$350K

average cost of downtime
per hour



\$10M

average cost of unplanned data
center outage or disaster



87 hours

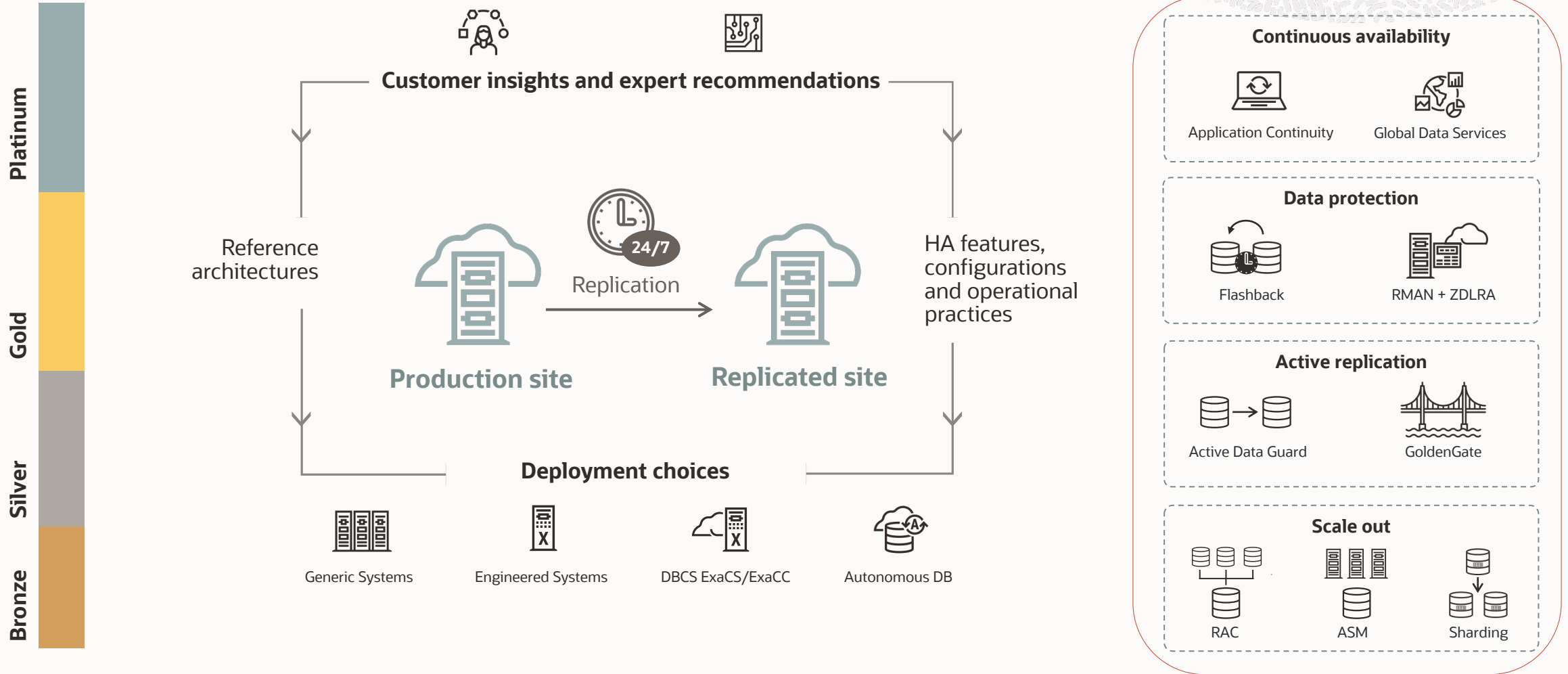
average amount of downtime
per year



91%

percentage of companies that have
experienced an unplanned data
center outage in the last 24
months

Oracle Maximum Availability Architecture (MAA)



Challenges of deploying highly available systems



Cost and complexity



Lack of skills



Risk of failure



Data Guard

Capabilities Included with Oracle Database Enterprise Edition (EE)



Zero or sub-second data loss protection

Strong isolation using continuous Oracle validation

Lost-write detection

Universal support – all data types and applications

Comprehensive monitoring with Enterprise Manager

Automatic database failover

Automatic client failover

Standby-first patch apply

Database rolling maintenance

Select platform migrations

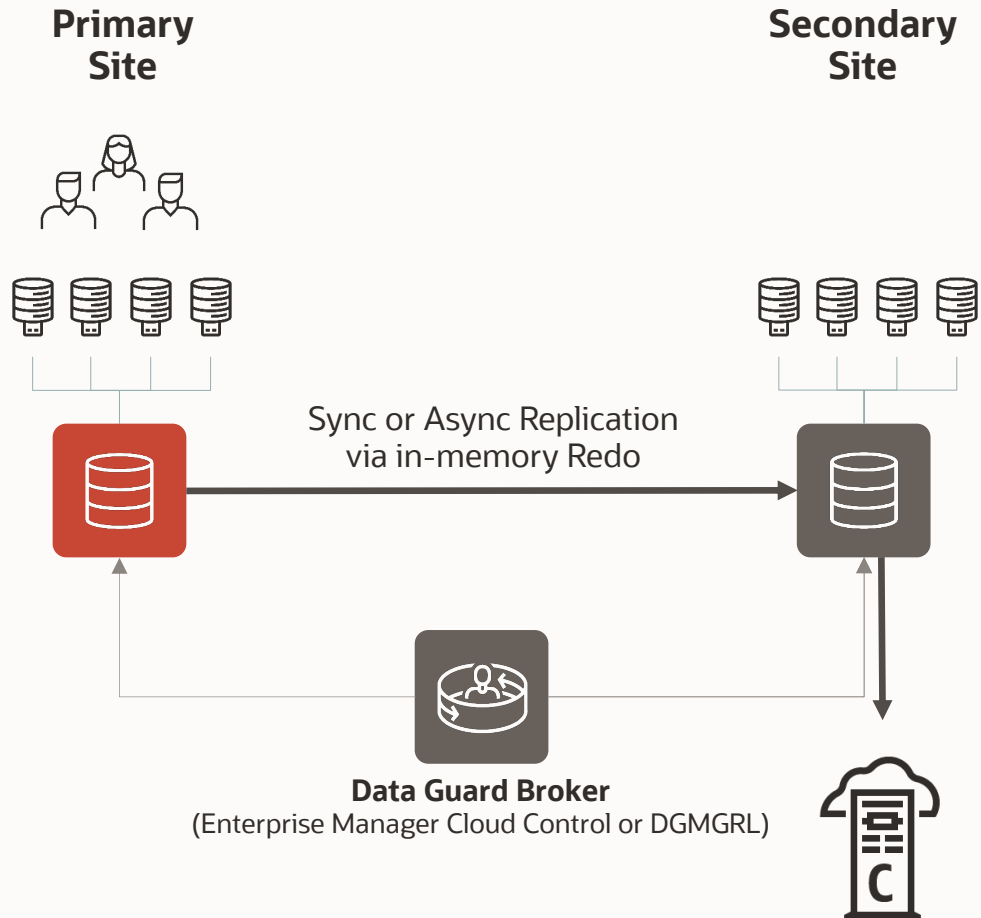
Extreme throughput - supports all workloads

Dual-purpose standby for development and test

Integrated management



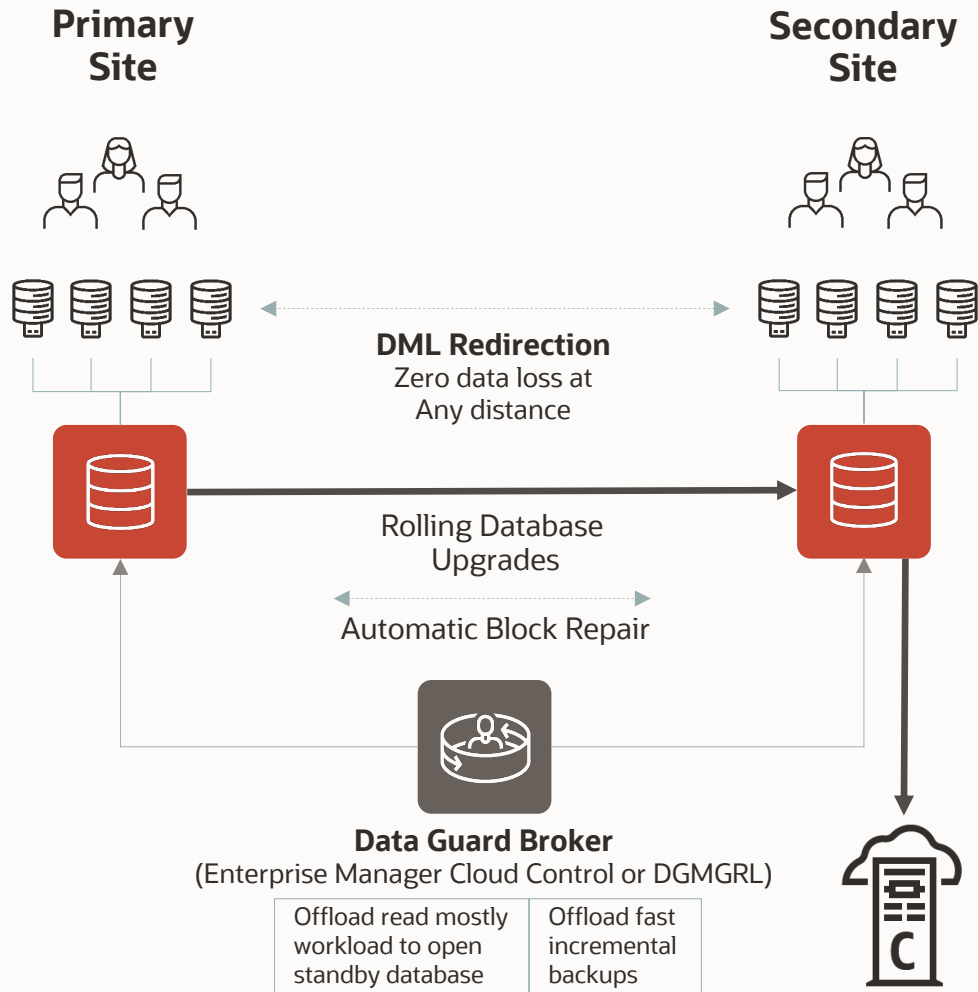
Oracle Data Guard (DG)



- **Basic DR (included with DB EE)**
 - License primary and secondary sites
- **Active-passive**
 - Standby is used only for failovers
- **Automatic failover to Standby site**
- **Zero / near-zero data loss**
- **Continuous data validation**
- **Simple migrations and upgrades**

<https://www.oracle.com/database/technologies/high-availability/dataguard-activedataguard-demos.html>

Oracle Active Data Guard (ADG)

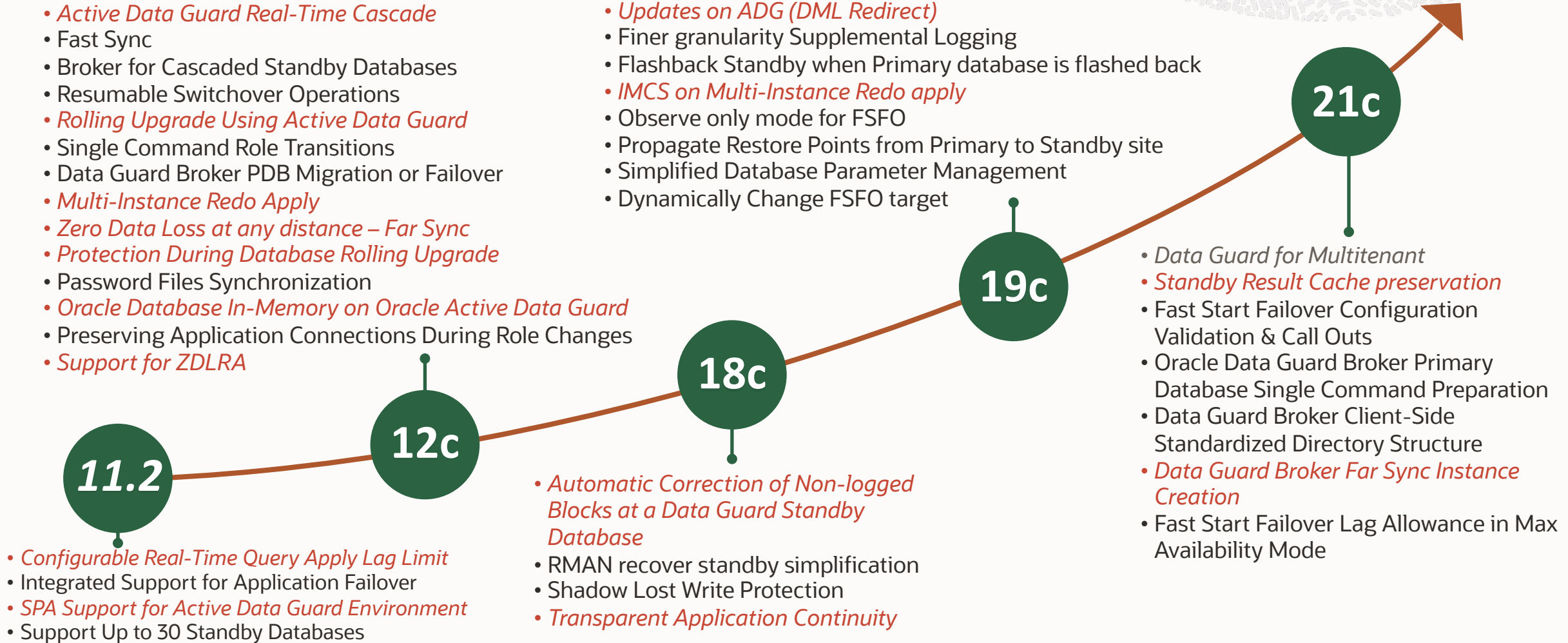


- **Advanced Disaster Recovery**
- **Active-active***
 - Queries, reports, backups
 - Occasional updates (19c)
 - Assurance of knowing system is operational
- **Automatic block repair**
- **Application Continuity**
- **Zero data loss across any distance**
- **Many other features**

<https://www.oracle.com/database/technologies/high-availability/dataguard-activedataguard-demos.html>

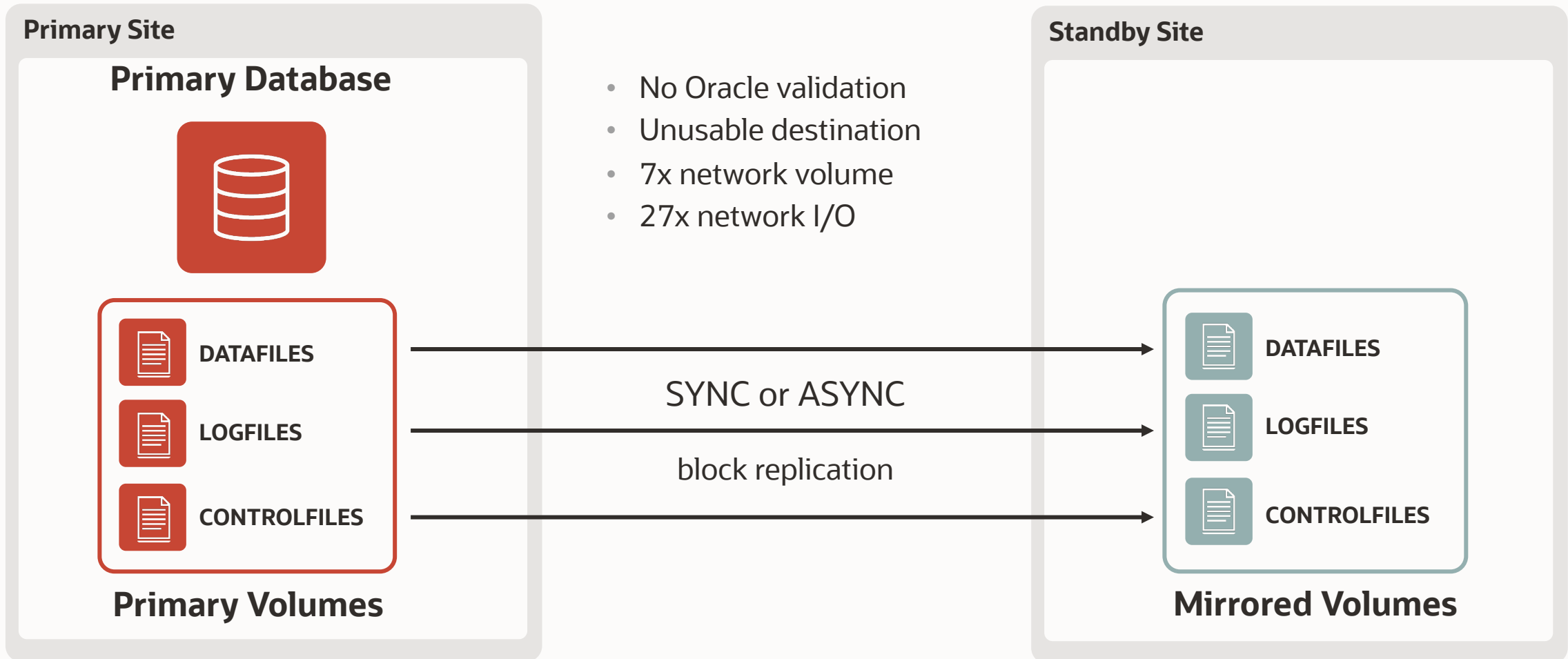
Oracle *Active* Data Guard

Actively protecting data towards the future *both* on-premises and in the cloud



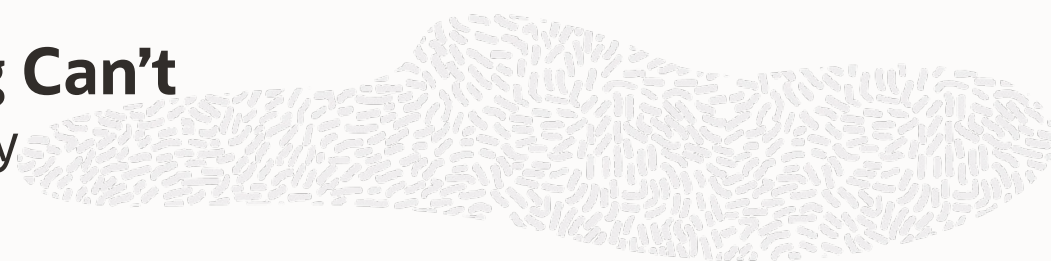
Storage Remote Mirroring Architecture

Mirrors every write to every file including those that are **corrupted or encrypted by ransomware**



Data Guard Does What Storage Mirroring Can't

Isolate Corruption, Protect Data, Maintain Availability

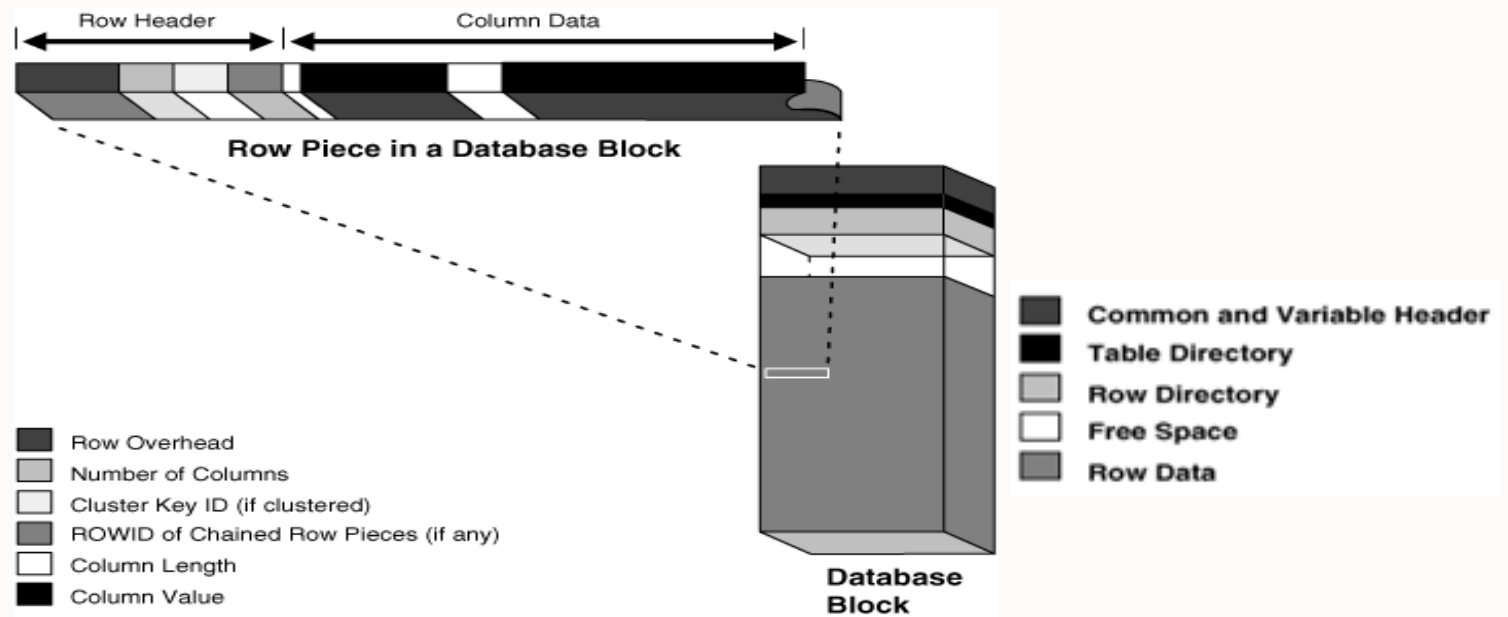


Storage Remote Mirroring...
blocks are just bits on a disk

Data Guard uses physical and logical
data consistency checks for end to end data integrity

Block 3941 (0x0f65)

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	
000	20	20	20	54	77	61	73	20	74	68	65	20	6e	69	67	68	
010	74	20	62	65	66	6f	72	65	20	73	74	61	72	74	2d	75	
020	70	20	61	6e	64	20	61	6c	6c	20	74	68	72	6f	75	67	
030	68	20	74	68	65	20	6e	65	74	2c	0a	20	20	20	20	20	
040	6e	6f	74	20	61	20	70	61	63	6b	65	74	20	77	61	73	
050	20	6d	6f	76	69	6e	67	3b	20	6e	6f	20	62	69	74	20	
060	6e	6f	72	20	6f	63	74	65	74	2e	0a	20	20	20	54	68	
070	65	20	65	6e	67	69	6e	65	65	72	73	20	72	61	74	74	
080	6c	65	64	20	74	68	65	69	72	20	63	61	72	64	73	20	
090	69	6e	20	64	65	73	70	61	69	72	2c	0a	20	20	20	20	
0a0	20	68	6f	70	69	6e	67	20	61	20	62	61	64	20	63	68	
0b0	69	70	20	77	6f	75	6c	64	20	62	6c	6f	77	20	77	69	
0c0	74	68	20	61	20	66	6c	61	72	65	2e	0a	20	20	20	54	
0d0	68	65	20	73	61	6c	65	73	6d	65	6e	20	77	65	72	65	
0e0	20	6e	65	73	74	6c	65	64	20	61	6c	6c	20	73	6e	75	
0f0	67	20	69	6e	20	74	68	65	69	72	20	62	65	64	73	2c	
100	0a	20	20	20	20	20	20	77	68	69	6c	65	20	76	69	73	69
110	6f	6e	73	20	6f	66	20	64	61	74	61	20	6e	65	74	73	
120	20	64	61	6e	63	65	64	20	69	6e	20	74	68	65	69	72	
130	20	68	65	61	64	73	2e	0a	20	20	20	20	41	6e	64	20	49
140	20	77	69	74	68	20	6d	79	20	64	61	74	61	73	63	6f	
150	70	65	20	74	72	61	63	69	6e	67	73	20	61	6e	64	20	
160	64	75	6d	70	73	0a	20	20	20	20	20	20	70	72	65	70	61
170	72	65	64	20	66	6f	72	20	73	6f	6d	65	20	70	72	65	
180	74	74	79	20	62	61	64	20	62	72	75	69	73	65	73	20	
190	61	6e	64	20	6c	75	6d	70	73	2e	0a	20	20	20	57	68	
1a0	65	6e	20	6f	75	74	20	69	6e	20	74	68	65	20	68	61	
1b0	6c	6c	20	74	68	65	72	65	20	61	72	6f	73	65	20	73	
1c0	75	63	68	20	61	20	63	6c	61	74	74	65	72	2c	0a	20	
1d0	20	20	20	20	49	20	73	70	72	61	6e	67	20	66	72	6f	
1e0	6d	20	6d	79	20	64	65	73	6b	20	74	6f	20	73	65	65	
1f0	20	77	68	61	74	20	77	61	73	20	74	68	65	20	6d	61	

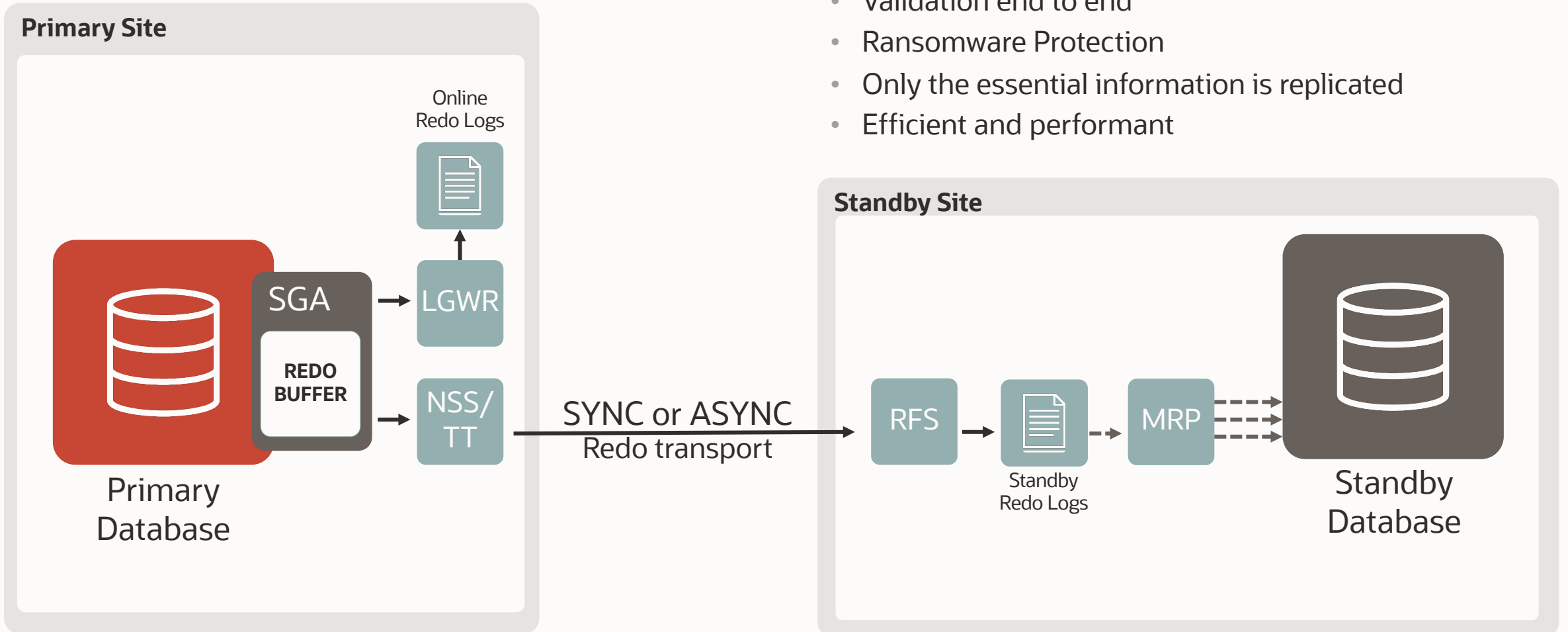


See My Oracle Support Note 1302539.1 for details



Data Guard is optimized for the database

It efficiently maintains a **physical copy** of production and **guarantees its integrity**



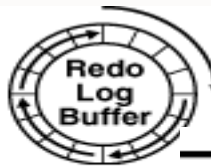
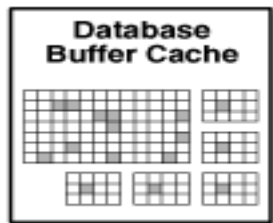
- Validation end to end
- Ransomware Protection
- Only the essential information is replicated
- Efficient and performant



Data Guard Provides Strongest Fault Isolation and Best Performance

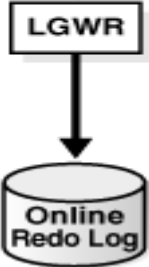


System Memory (SGA)

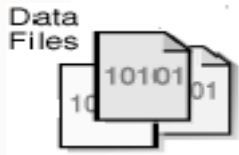


TCP/IP

To Standby Databases



DBWn



Data Guard transmits redo blocks directly from SGA: like a *memcpy* over the network

Redo received / applied by running Oracle instance: continuous Oracle-integrated data validation

- Best isolation from lower layer faults
- Best performance since no disk I/O
- Best network utilization: only redo sent
- Transactional consistency: always
- Corrupted blocks auto-repaired *
- Database-integrated application failover

* Requires Active Data Guard License

Oracle Active Data Guard Compared to Storage Remote Mirroring

<https://www.oracle.com/a/tech/docs/adg-vs-storage-mirroring.pdf>

Oracle Replication done right

<https://blogs.oracle.com/maa/replication-done-right>





**Everything you will see
is a “Technology Preview”
and will *eventually*
become GA in 21c**

*“Today Oracle announced the general availability (GA) of Oracle Database 12c, the first database designed for the Cloud. **Oracle Multitenant**, new with Oracle Database 12c, is a key component of this – a new architecture for consolidating databases and simplifying operations in the Cloud.”*

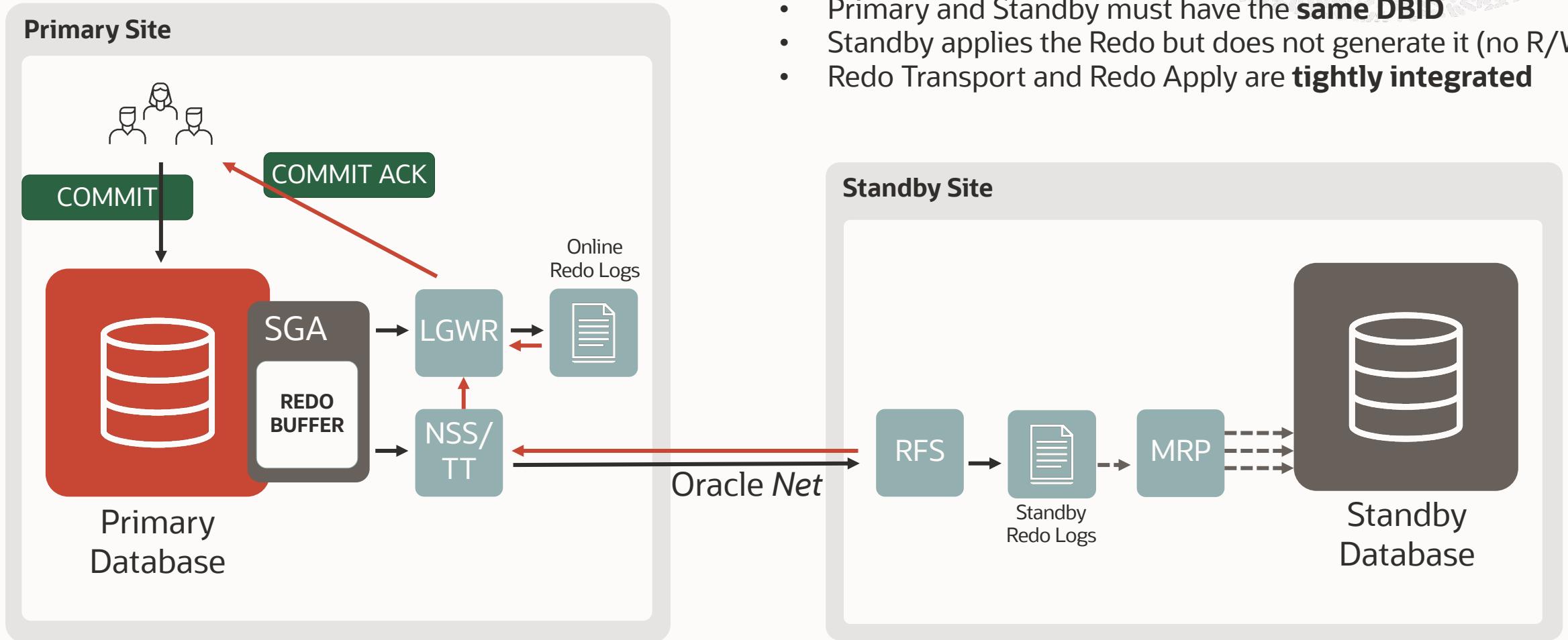
➤ Patrick Wheeler, June 25th 2013



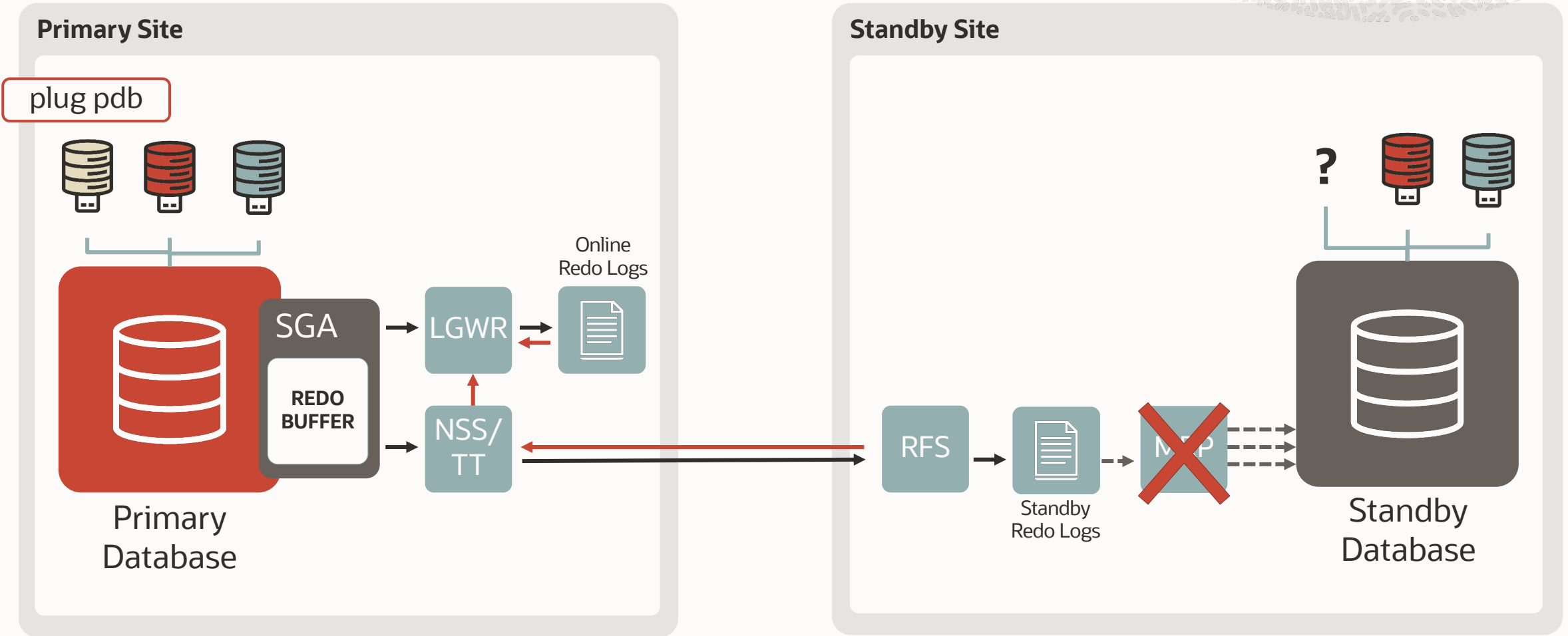
Non-CDB Data Guard Process Architecture

Built and perfected over 20+ years for non-CDB

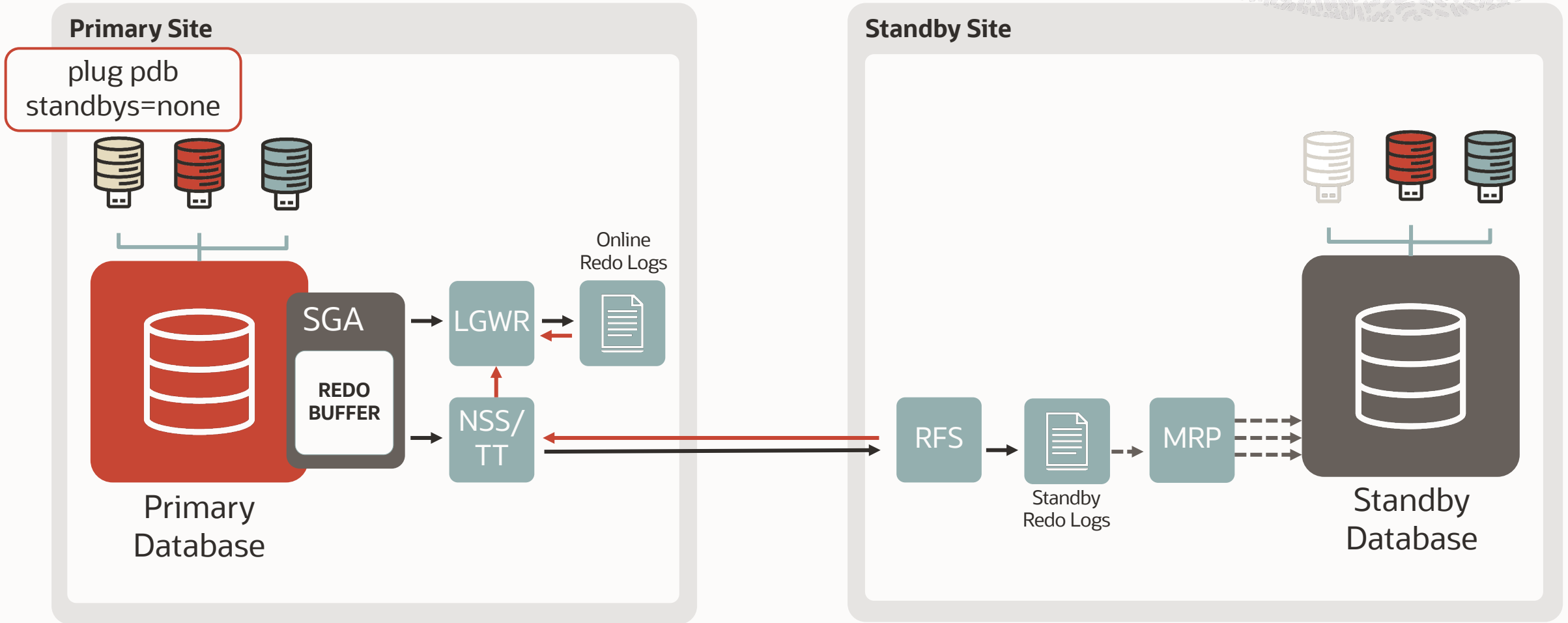
- Primary and Standby must have the **same DBID**
- Standby applies the Redo but does not generate it (no R/W)
- Redo Transport and Redo Apply are **tightly integrated**



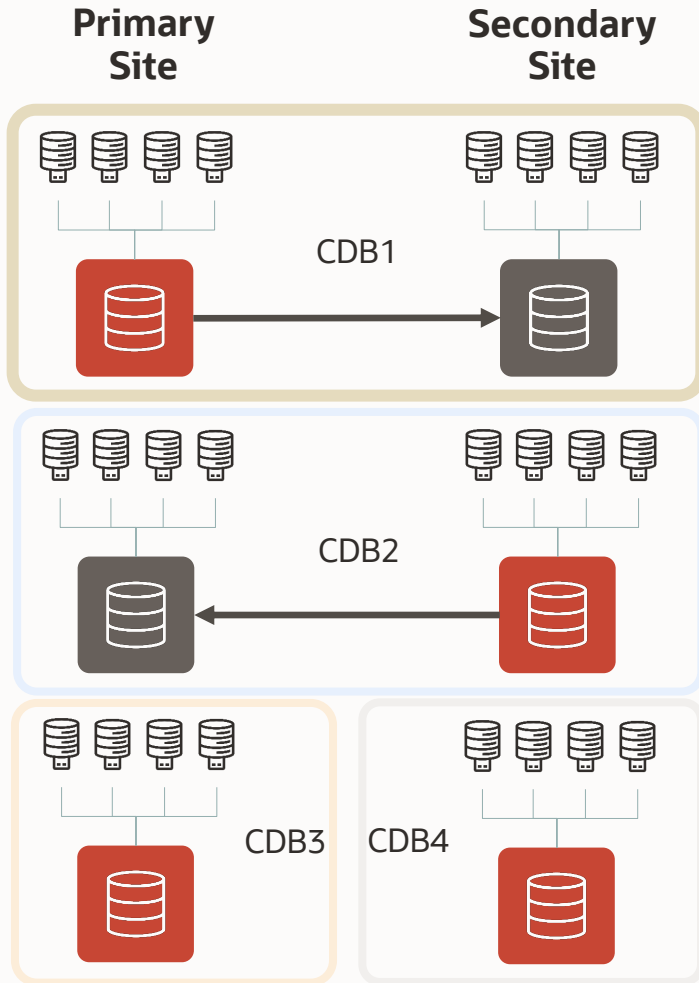
Oracle Data Guard for Container Databases – 12.1.0.1



Oracle Data Guard for Container Databases – 12.1.0.2



Oracle Data Guard for Container Databases – Consolidation



All PDBs in a CDB are either Primary or Standby

- Role transition at CDB level

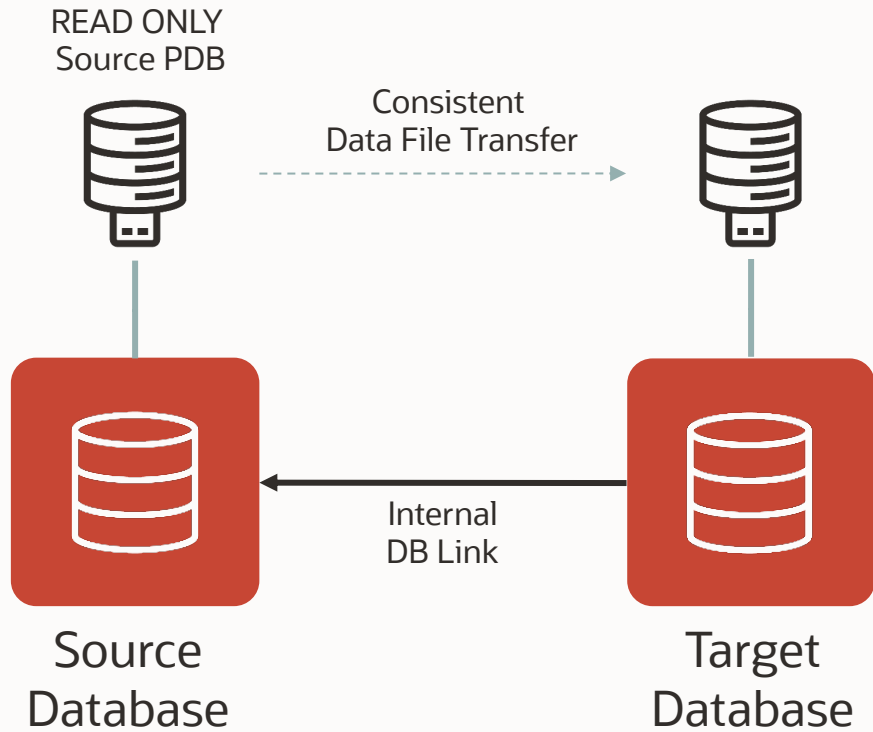
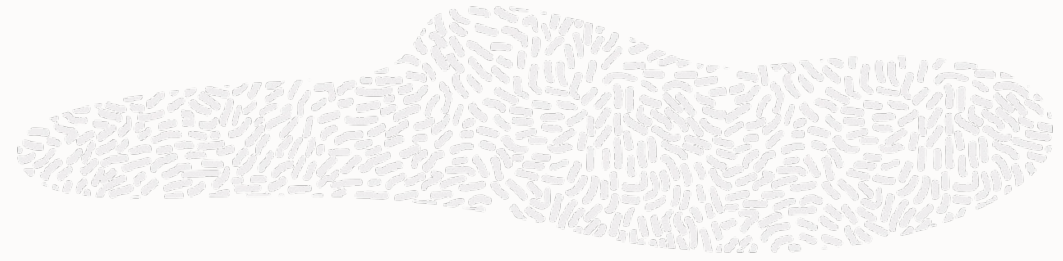
Multiple CDBs required to accommodate role requirements

- For protected and unprotected PDBs
- For applications active on different sites

Best protection today for production workloads

- All Active Data Guard features are available for DG per CDB

12cR1 – PDB “Cold” Clone



Two Primary databases: different DBID

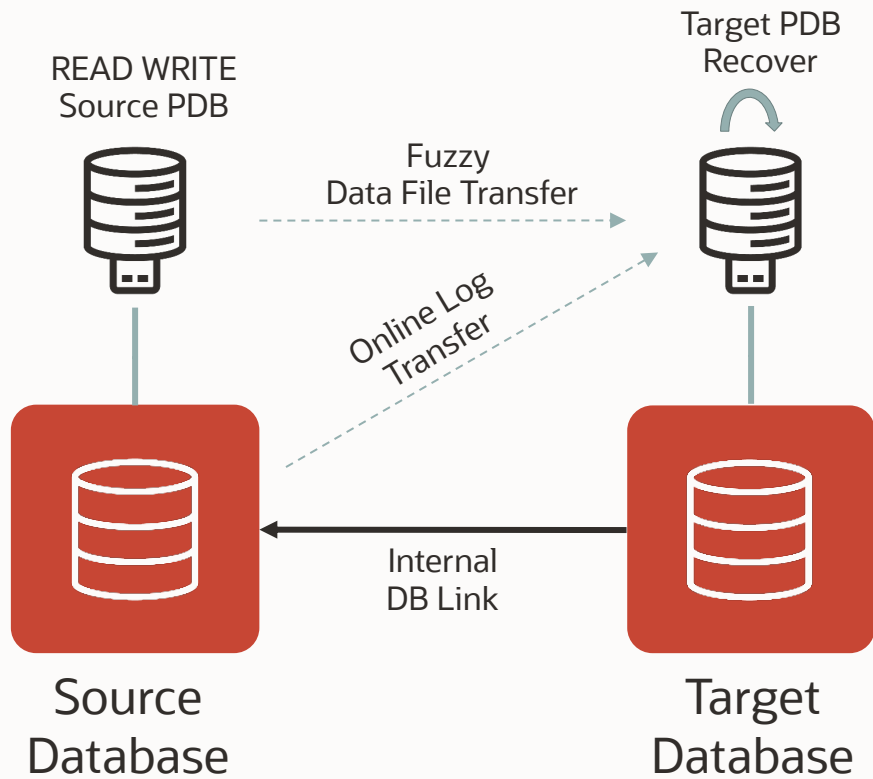
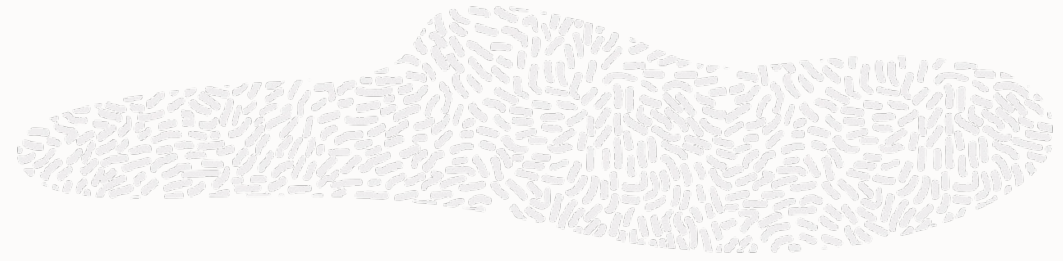
- Recovery of **Foreign Redo** not possible in 12cR1
- Shared undo

PDB Cold Clone

- Source must be open Read Only for consistent data file transfer (No Redo Generated)
- Equivalent to a *plug database*



12cR2 – PDB “Hot” Clone



PDB Recovery with Foreign Redo

- Recovery with Foreign Redo introduced in 12cR2

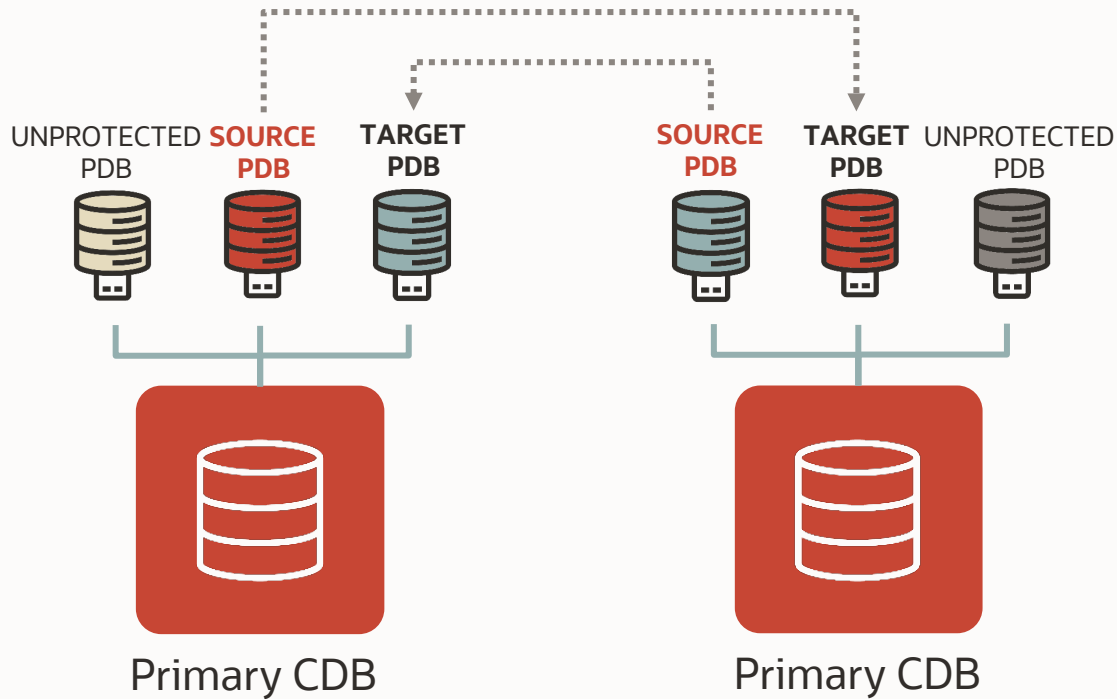
Pull of Online Log

- NSS/TT processes not available for non-DG configurations
- Redo transfer is still a one-off operation, network outages are problematic

Local Undo



Current status of Data Guard per PDB (DGPDB)



Two primary CDBs

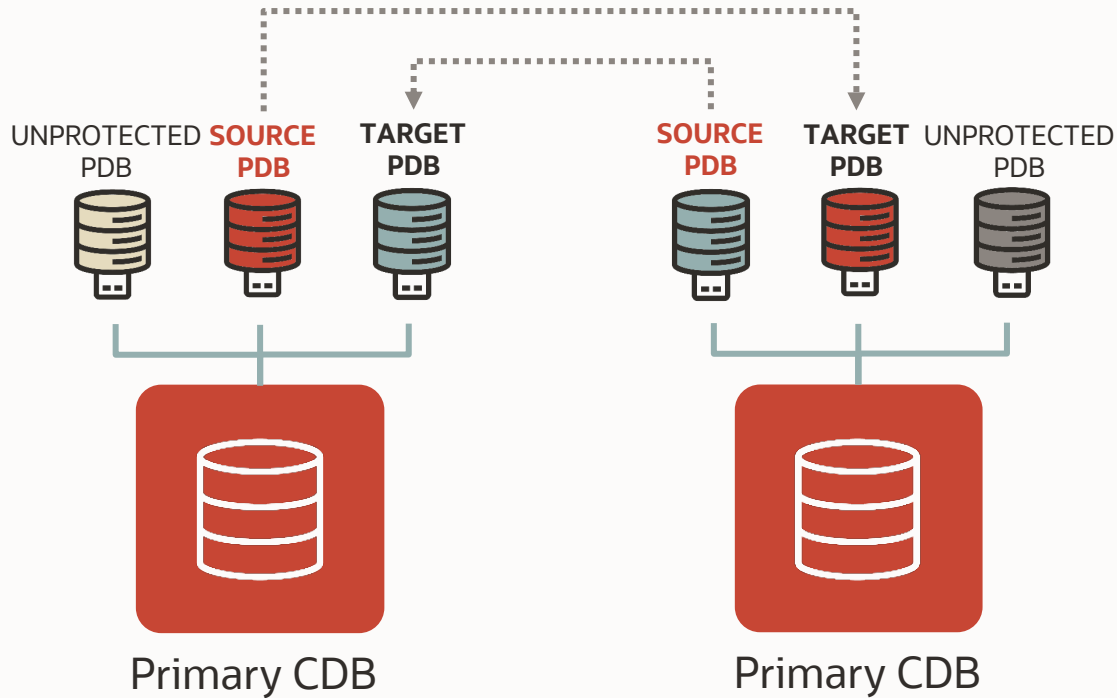
- Both open Read Write, different DBID

Disaster Protection at PDB level

- Protection on PDB or CDB level using real-time Apply
- No need to fail over a full Container Database!
- Role transition on a single PDB with the broker
- Automatic gap fetching from the source



Current limitations in DGPDB



- **Only two CDBs in the configuration**
- Only one Target PDB per Source PDB
- **Missing RMAN integration**
 - Backups on one site cannot be restored on the other site
- **No Real-time Query or DML redirect**
 - The target PDB must be mounted to apply the redo stream
- **No Fast-Start Failover**
 - Beware of split-brains when failing over manually
- Only Maximum Performance protection (**ASYNC**)
- Reinstatement might require a full copy after a failover
- **Archivelog deletion must be done manually**
 - CONFIGURE ARCHIVELOG DELETION POLICY TO NONE;
- Tempfiles added on the Target PDB only after switchover
- No Parallel Recovery, **Flashback**, or Automatic Block Repair
- DG per PDB and DG per CDB are mutually exclusive

Data Guard for Multitenant PDB versus CDB Configuration

Current limitations as “DGPDB” evolves to encompass more ADG features

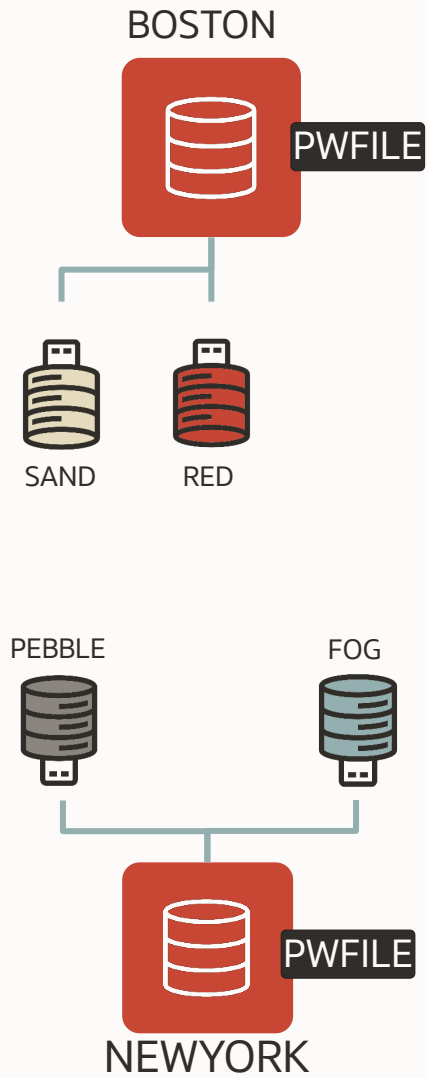
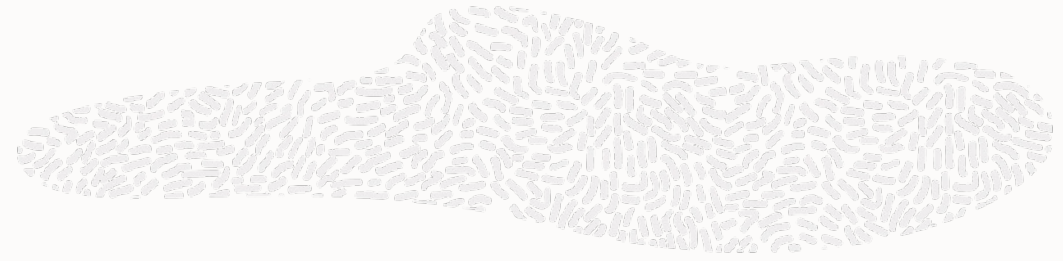
	Data Guard per CDB	Data Guard per PDB in 21c
Configuration	Once per CDB	For each PDB
Switchover and failover	Per CDB	Per PDB
Protection Modes	ALL	MAX PERFORMANCE only
Fast-Start Failover	✓	X
Real-Time Query	✓	X
Automatic Block Repair	✓	X
DML redirect	✓	X
Rolling Maintenance	✓	X
Application Continuity	✓	X
Multiple Standbys	✓	X
Reinstate using Flashback	✓	X
Max members	31	2



DGPDB architecture

—
Current architecture planned for the GA release

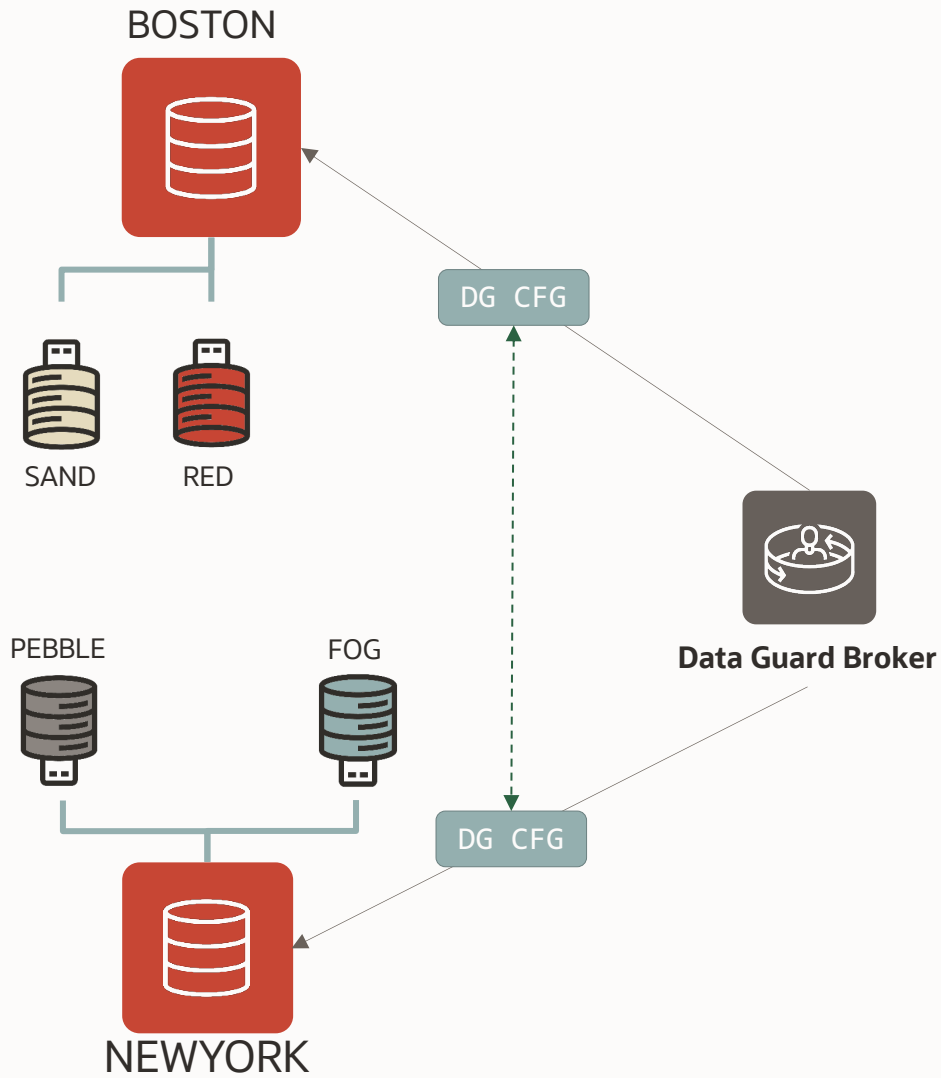
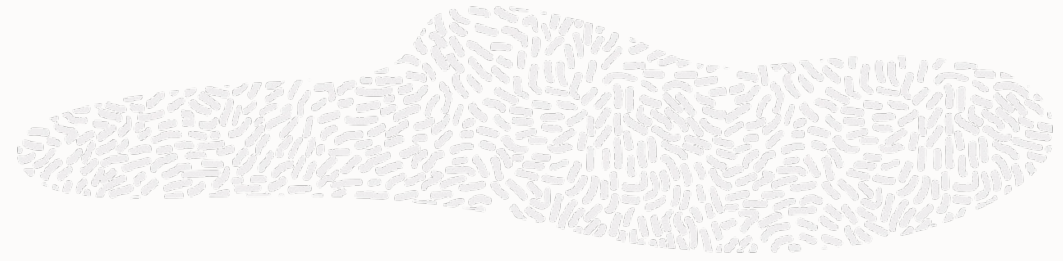
Two Primary CDBs



- Same preparation as *traditional* Primary CDBs
 - Initialization parameters
 - Force Logging (and optionally Flashback)
 - SQL*Net Configuration (TNS and Listener)
 - Same Password File or Setup Wallet



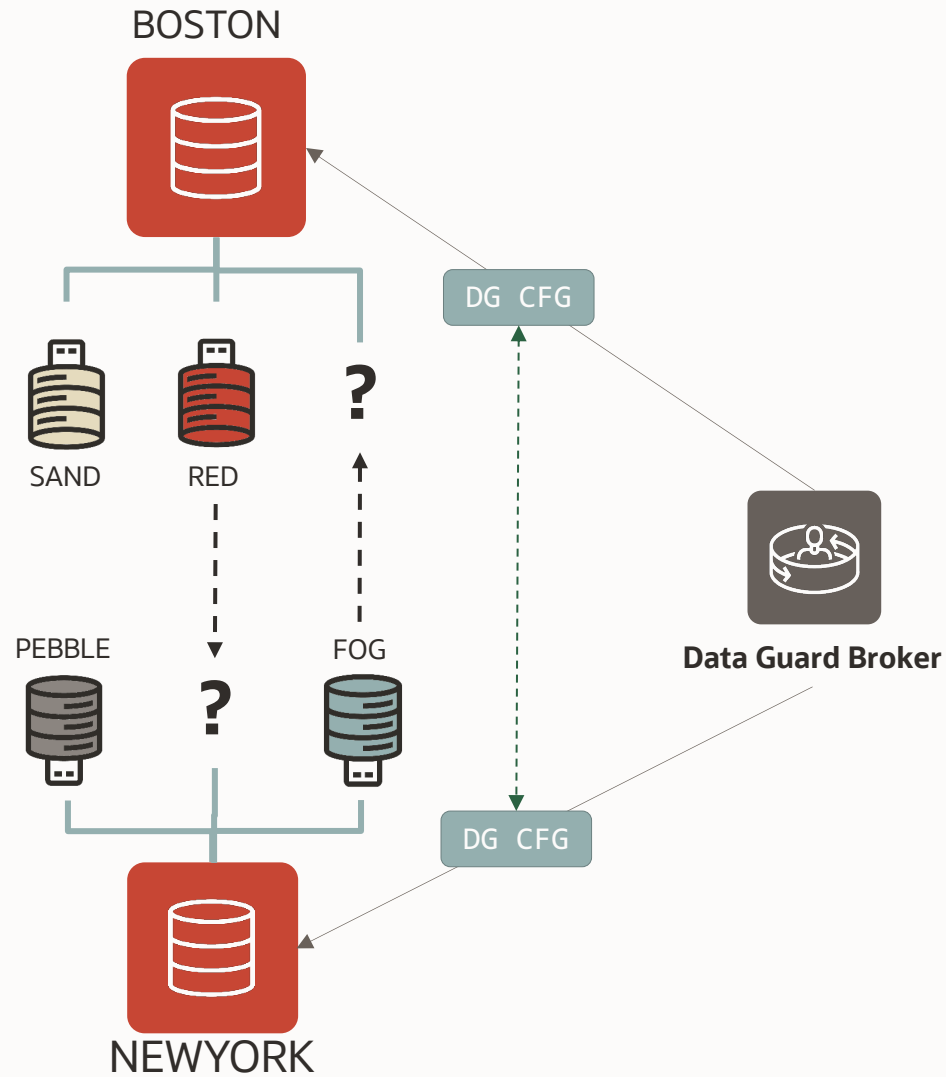
Two Data Guard Configurations



- Each configuration knows about the other
- Log shipping does not start without Target PDBs



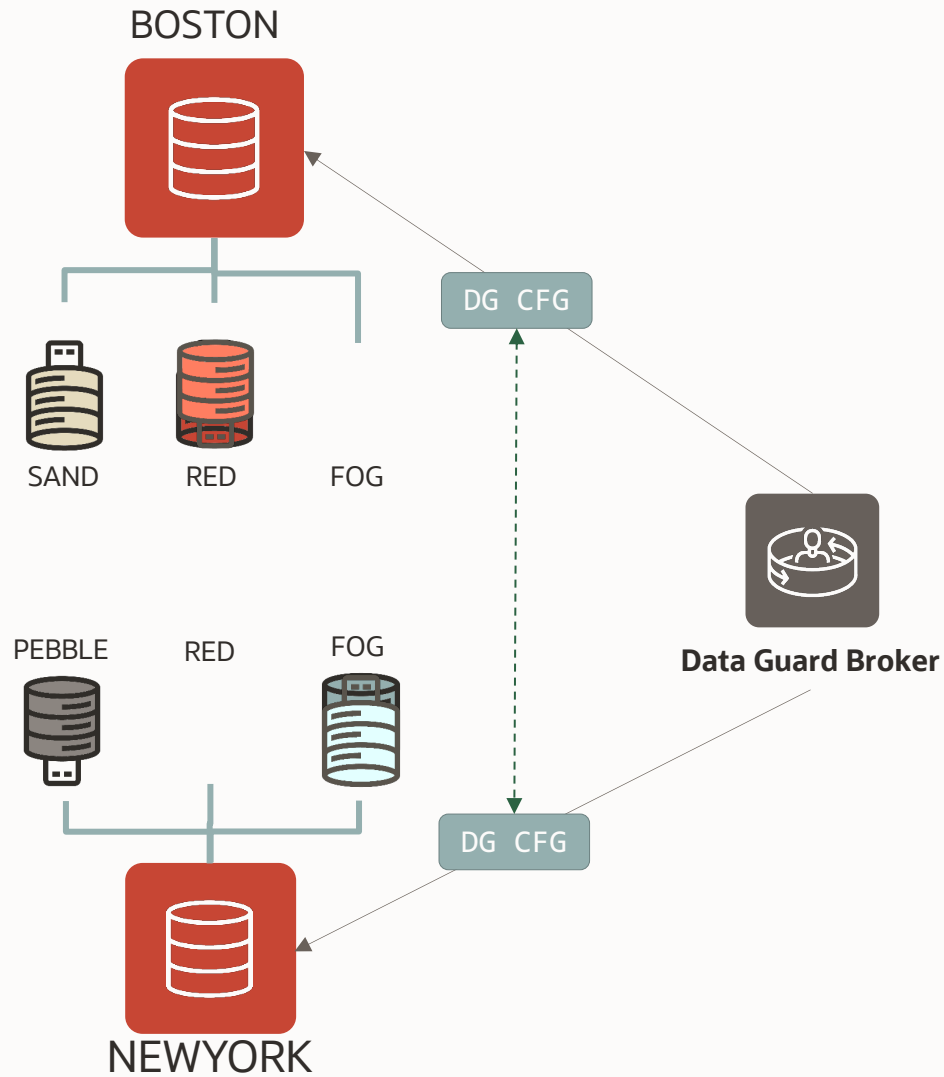
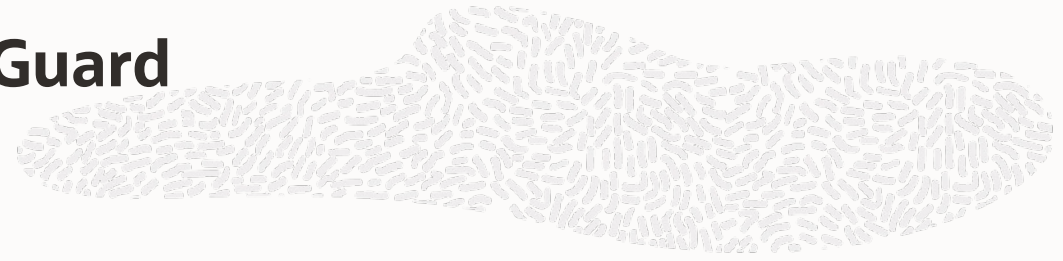
Add target Pluggable Databases to Data Guard



- The PDBs are added in a new “*STANDBY*” mode
 - Datafiles are *not* copied by as part of this command
- Each primary detects the remote Target PDB
 - The log shipping starts
 - The apply not yet: no datafiles to recover



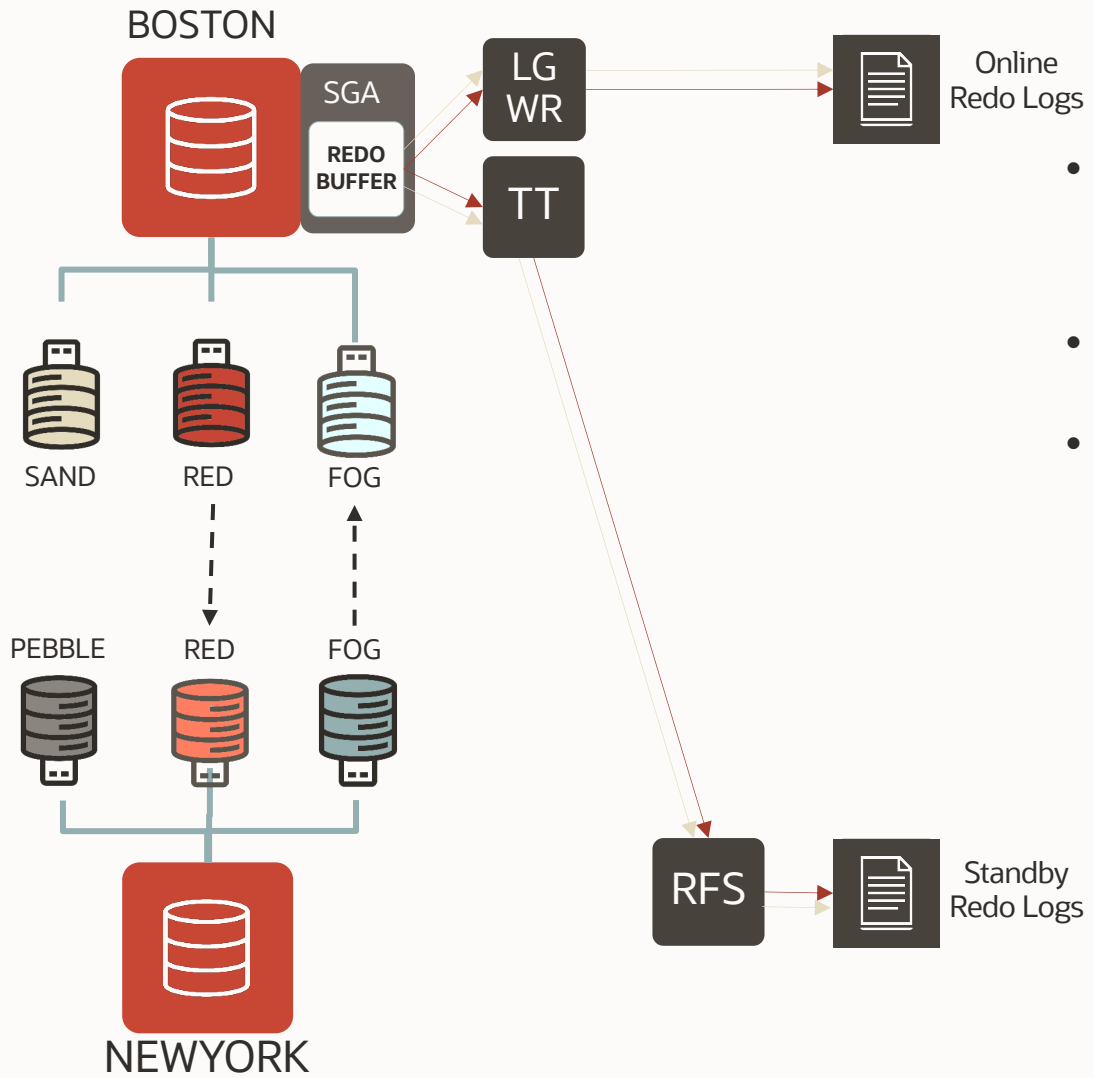
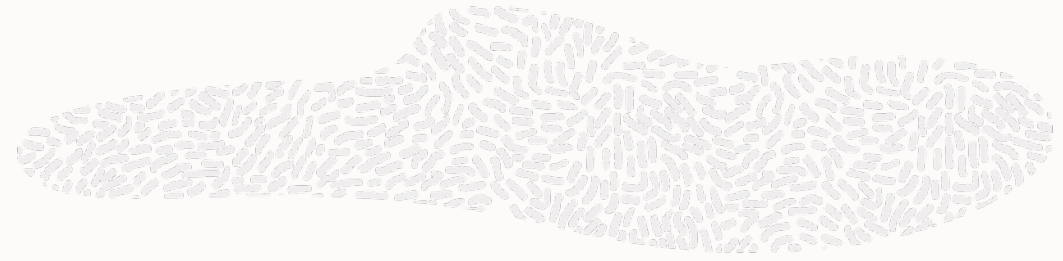
Add target Pluggable Databases to Data Guard



- PDB datafiles must be copied manually to the target database



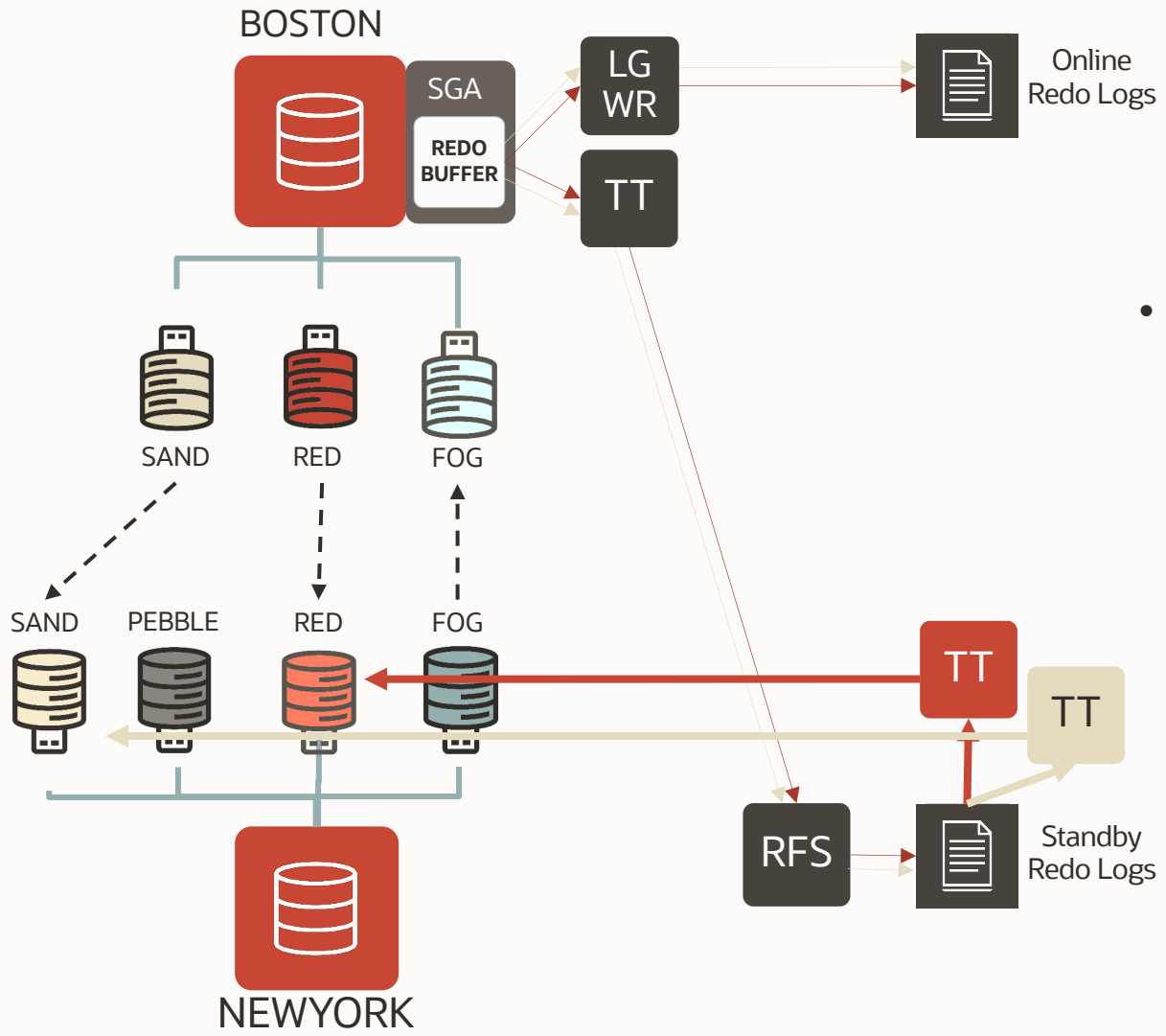
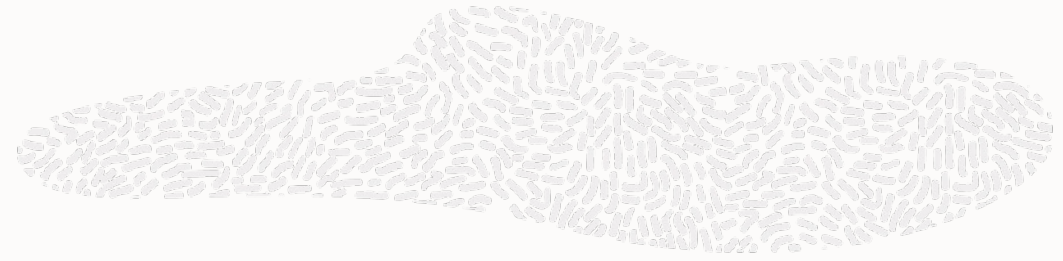
Transport and Apply Architecture



- The actual log shipping is done at CDB level
 - The stream has all the PDBS of the Source CDB
- Retries, gap fetch, monitoring have DG robustness
- Standby Redo Logs are required for Real-Time Apply
 - One set of SRLs for each source CDB (one today)
 - SRLs creation “at PDB level”



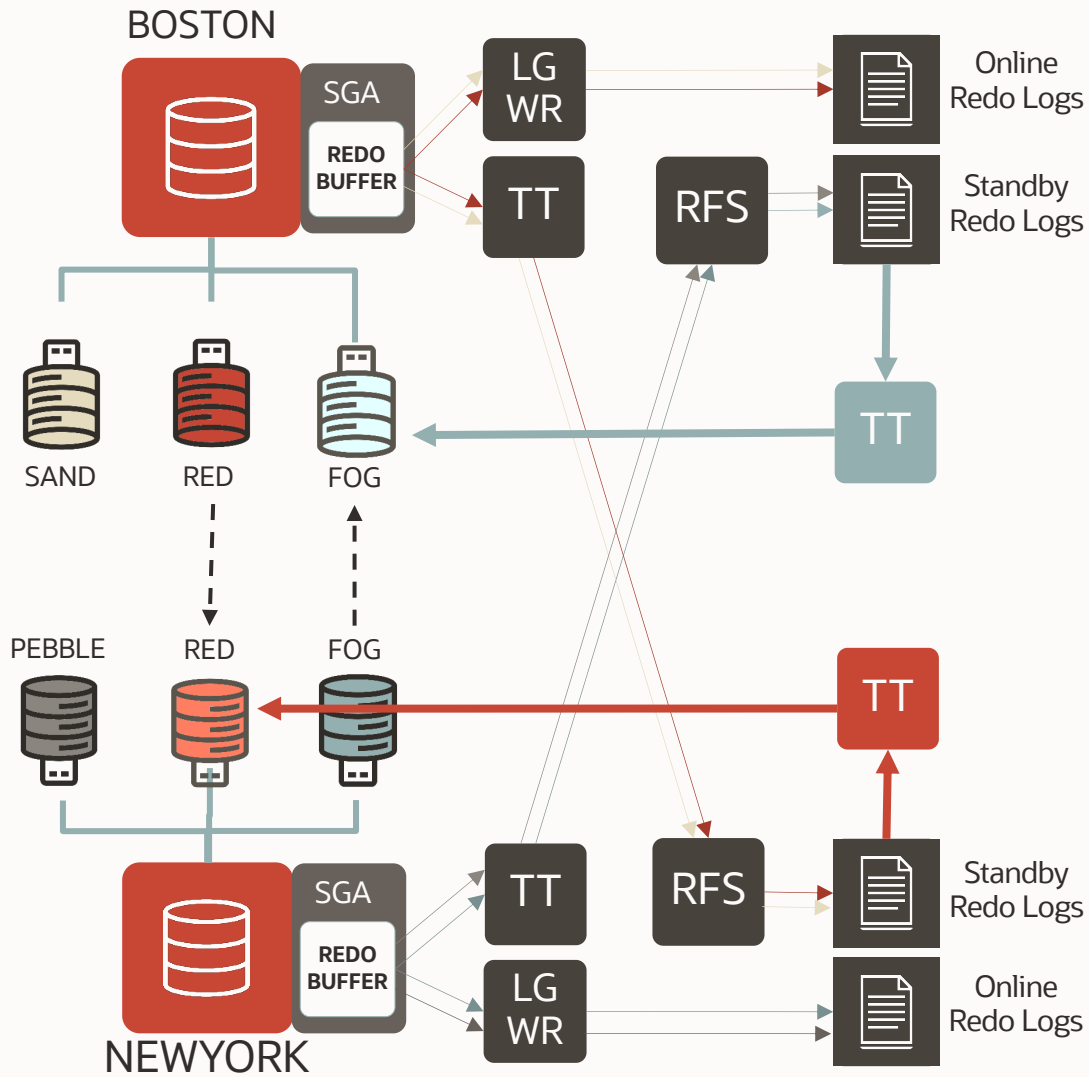
Transport and Apply Architecture



- One Apply Process per PDB
 - TTnn for PDB recovery instead of MRPn



Transport and Apply Architecture



```
DGMGRL> show all pluggable database at boston;
```

Pluggable database - SAND at boston

Data Guard Role: **Not Protected**

Pluggable database - RED at boston

Data Guard Role: **Primary**

Con_ID: 4

Active Target: con_id 5 at newyork

Pluggable Database Status:

SUCCESS

Pluggable database - FOG at boston

Data Guard Role: **Physical Standby**

Con_ID: 5

Source: con_id 3 at newyork

Transport Lag: 0 seconds (computed 39 seconds ago)

Intended State: APPLY-ON

Apply State: Running

Apply Instance: boston

Average Apply Rate: (unknown)

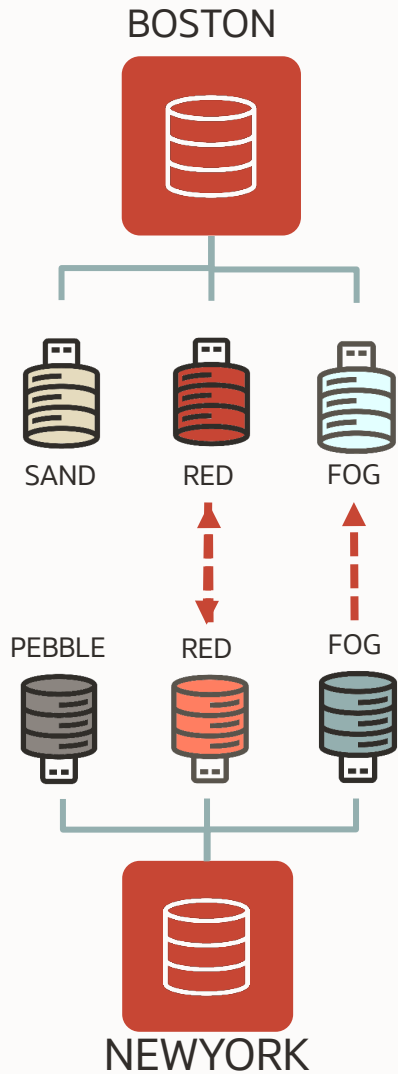
Real Time Query: OFF

Pluggable Database Status:

SUCCESS



PDB Switchover



```
DGMGRL> ! date
Executing operating system command(s):" date"
Sat Nov 13 22:19:25 GMT 2021

[...]

Performing switchover NOW, please wait...

Closing pluggable database 'RED'...
Switching 'RED' to standby role...
Waiting for 'RED' to recover all redo data...
Stopping recovery at 'RED'...
Converting 'RED' to primary role...
Opening new primary 'RED'...
Waiting for redo data from new primary 'RED'...
Starting recovery at new standby 'RED'...

Switchover succeeded, new primary is "RED"
```

```
DGMGRL> ! date
Executing operating system command(s):" date"
Sat Nov 13 22:19:38 GMT 2021
```

13 seconds!

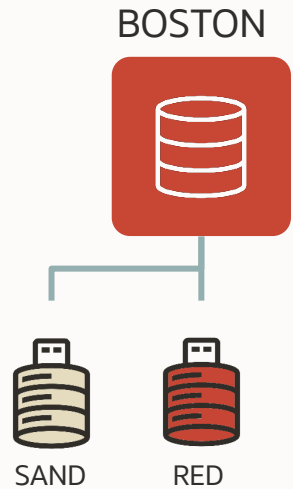
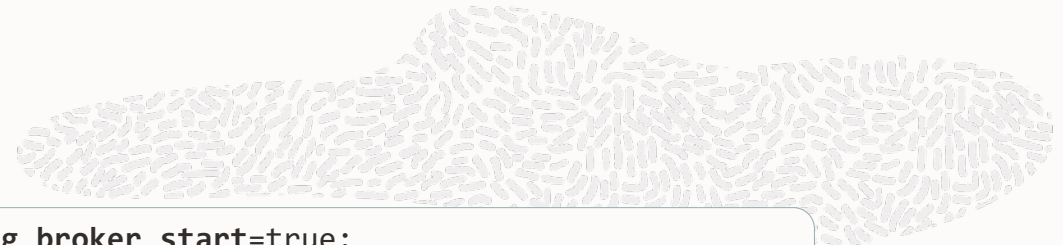


DGPDB setup

—
Current steps for the Technology Preview in 21c



Prepare both CDBs for Data Guard

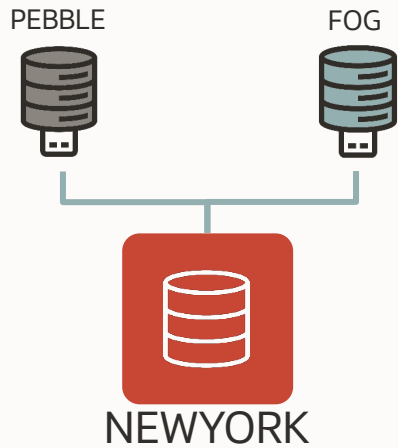


- Change the parameters

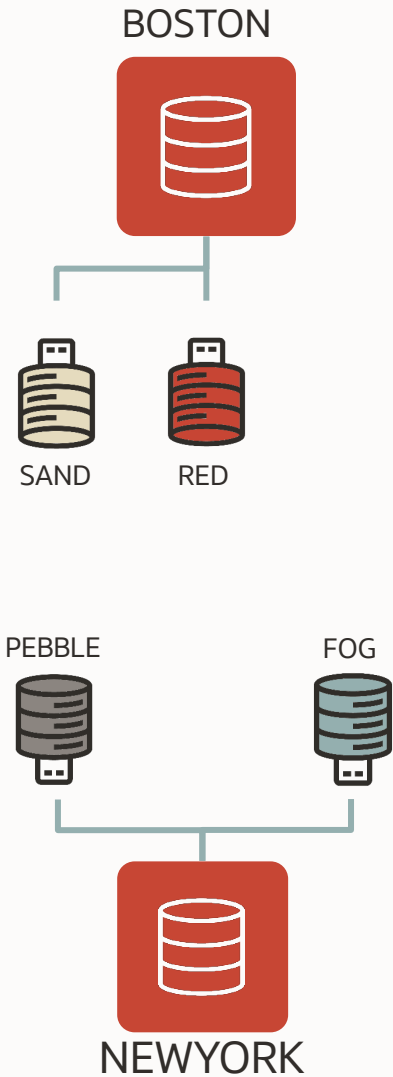
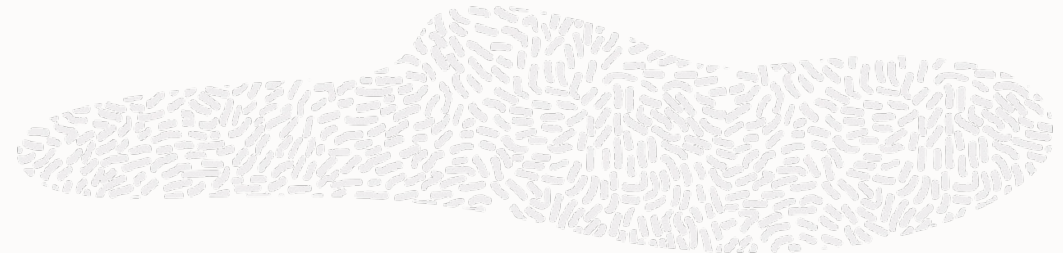
```
SQL> alter system set dg_broker_start=true;  
SQL> alter system set standby_file_management=auto;
```

- Unlock SYSDBG and change its password (some steps will require SYS though)

```
SQL> alter user SYSDBG account unlock;  
SQL> grant SYSDBG to SYSDBG;  
SQL> alter user SYSDBG identified by secret;
```



Prepare both CDBs for Data Guard (cont.)



- Add the TNS aliases in `$(orabasehome)/network/admin/tnsnames.ora` so that they point to CDB\$ROOT

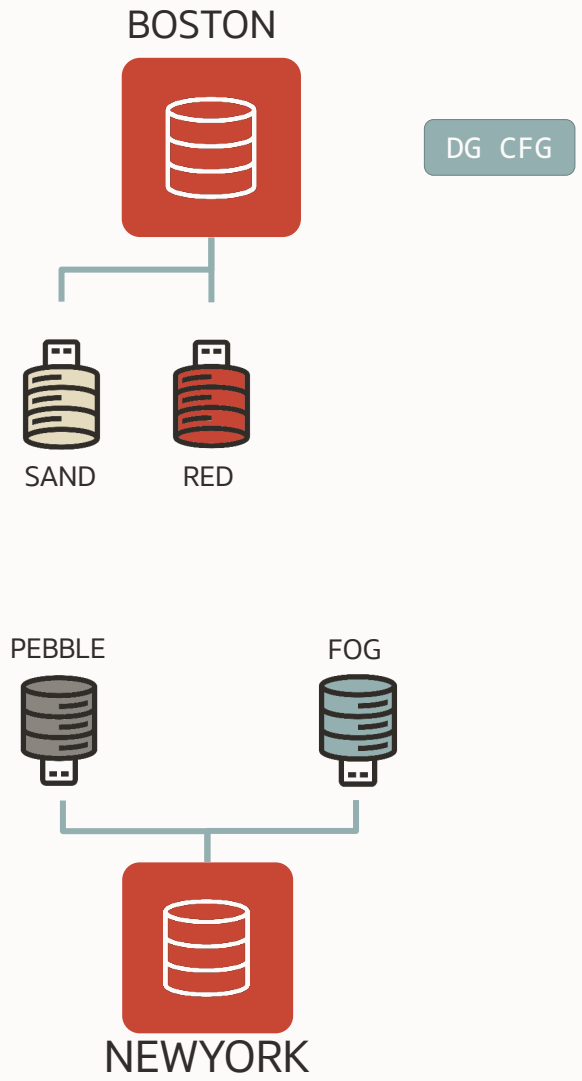
```
BOSTON = (DESCRIPTION =  
  (ADDRESS = (PROTOCOL = TCP)(HOST = host1.us.oracle.com)(PORT = 1521))  
  (CONNECT_DATA = (SERVER = DEDICATED)(SERVICE_NAME = boston.us.oracle.com))  
)  
NEWYORK = (DESCRIPTION =  
  (ADDRESS = (PROTOCOL = TCP)(HOST = host2.us.oracle.com)(PORT = 1521))  
  (CONNECT_DATA = (SERVER = DEDICATED)(SERVICE_NAME = newyork.us.oracle.com))  
)
```

- Enable force logging

```
SQL> alter database force logging;
```



Add the Broker Configurations



- Create the configuration for the first CDB. Which one is irrelevant!

```
DGMGRL> connect sys/<my_SYS_password>@boston
Connected to "boston"
DGMGRL> create configuration boston primary database is boston
connect identifier is boston;
Configuration "boston" created with primary database "boston"
DGMGRL> show configuration

Configuration - boston

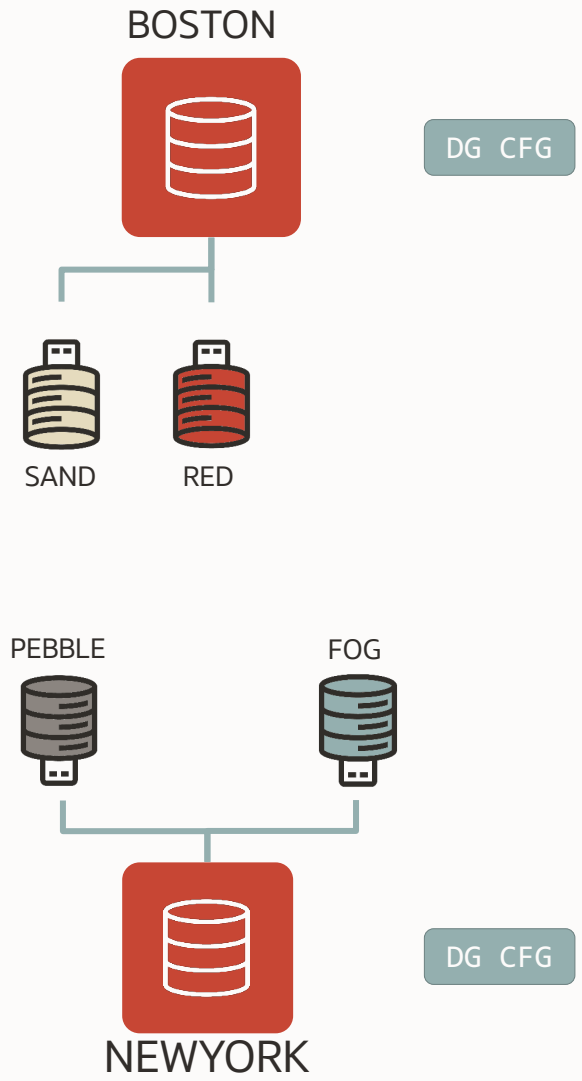
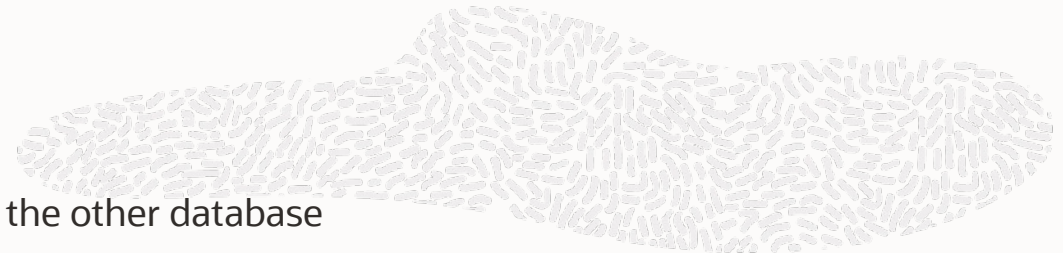
Protection Mode: MaxPerformance
Members:
  boston - Primary database

Fast-Start Failover: Disabled

Configuration Status:
DISABLED
```



Add the Broker ConfigurationS (cont.)



- Add the **configuration** for the other database

```
DGMGRL> add configuration newyork connect identifier is newyork;  
Configuration newyork added.  
DGMGRL> enable configuration all;  
Succeeded.  
DGMGRL> show configuration
```

Configuration - boston

Protection Mode: MaxPerformance
Members:
boston - Primary database
newyork - Primary database in newyork configuration

Fast-Start Failover: Disabled

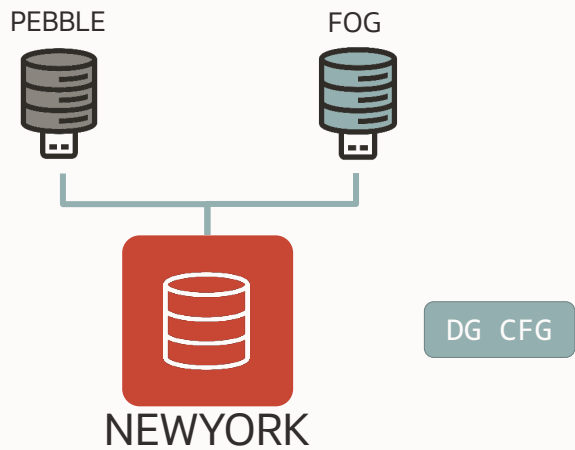
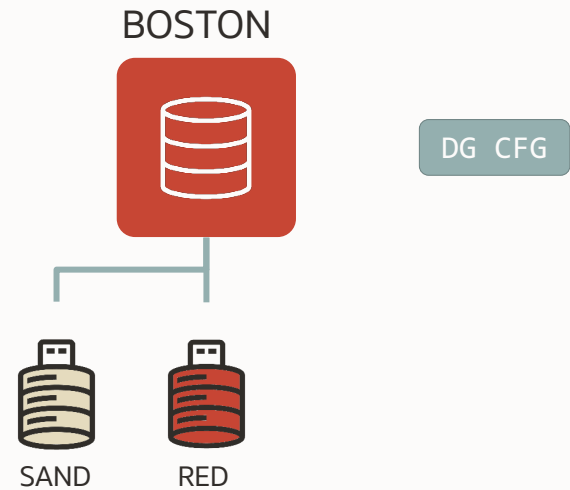
Configuration Status:
SUCCESS (status updated 4 seconds ago)

New Commands

- Log shipping does not happen at this stage



Check the PDBs before adding them



- Check the PDBs

```
DGMGRL> show pluggable database red at boston;  
  
Pluggable database 'red' at database 'boston'  
  
Data Guard Role: Not Protected  
DGMGRL> show pluggable database fog at newyork;  
  
Pluggable database fog at database 'newyork'  
  
Data Guard Role: Not Protected
```

- Make sure the PDBs are open READ WRITE

```
SQL> connect sys/<my_SYS_password>@boston as sysdba  
Connected.  
SQL> select open_mode from v$pdb where name='RED';  
  
OPEN_MODE  
-----  
READ WRITE  
  
SQL> connect sys/<my_SYS_password>@newyork as sysdba  
Connected.  
SQL> select open_mode from v$pdb where name='FOG';  
  
OPEN_MODE  
-----  
READ WRITE
```

SYSDG won't work here

Add the PDBs to DGPDB

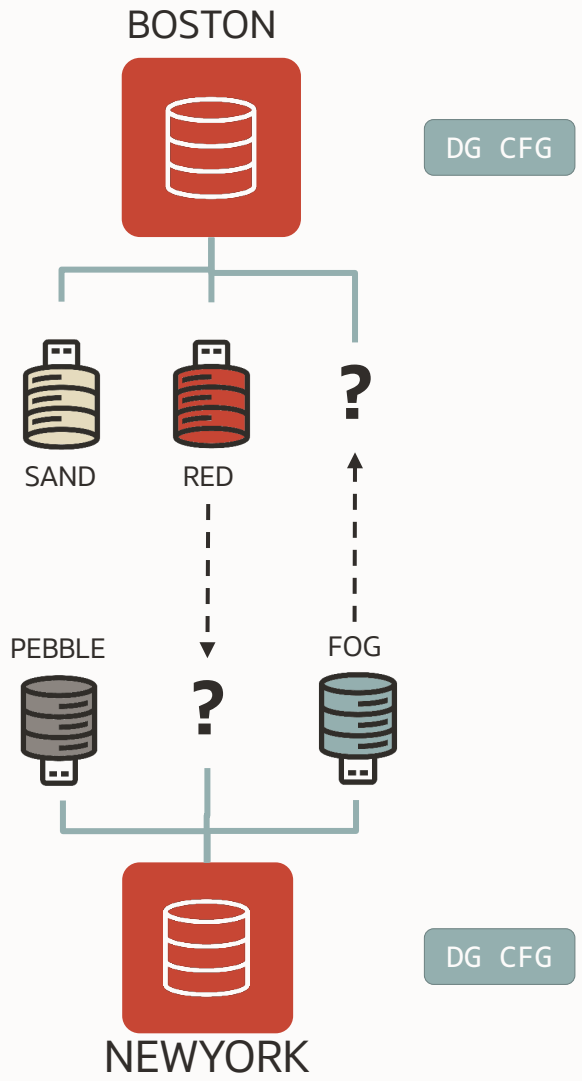
- Add the PDBs that you want to protect

```
DGMGRL> add pluggable database red at newyork source is red at boston
pdbfilenameconvert is ''/BOSTON/RED/','/NEWYORK/RED/'';
Enter password for DGPDB@boston:
Enter password for DGPDB@newyork:

Pluggable Database "RED" added

DGMGRL> add pluggable database fog at boston source is fog at newyork
pdbfilenameconvert is ''/NEWYORK/FOG/','/BOSTON/FOG/'';

Pluggable Database "FOG" added
```



- The pdbfilenameconvert is mandatory
- The password is for internal activity between the CDBs. It's required when setting up DGPDB the first time. It can differ between source and target CDB.
- The two databases start the log shipping, but the PDBs are NOT there yet
- The next status shows the redo transport setup between the two CDBs.

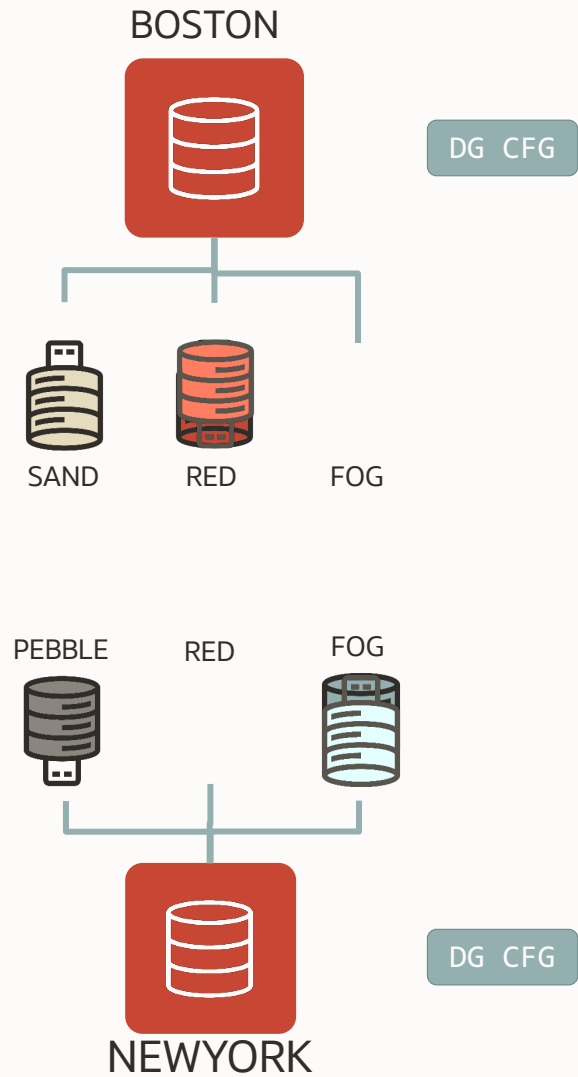
```
DGMGRL> show configuration

Configuration - boston

Protection Mode: MaxPerformance
Members:
boston - Primary database
newyork - Primary database in newyork configuration
```



Copy the PDBs *manually*



- This requires BEGIN BACKUP and END BACKUP as RMAN integration is not there yet

```
SQL> connect sys/<my_SYS_password>@boston  
SQL> alter session set container=red;
```

Session altered.

```
SQL> alter database begin backup;
```

Database altered.

```
SQL> host scp -r /u01/data/BOSTON/RED oracle@host2:/u01/data/NEWYORK  
SQL> alter database end backup;
```

Database altered.

```
SQL> connect sys/<my_SYS_password>@newyork  
SQL> alter session set container=fog;
```

Session altered.

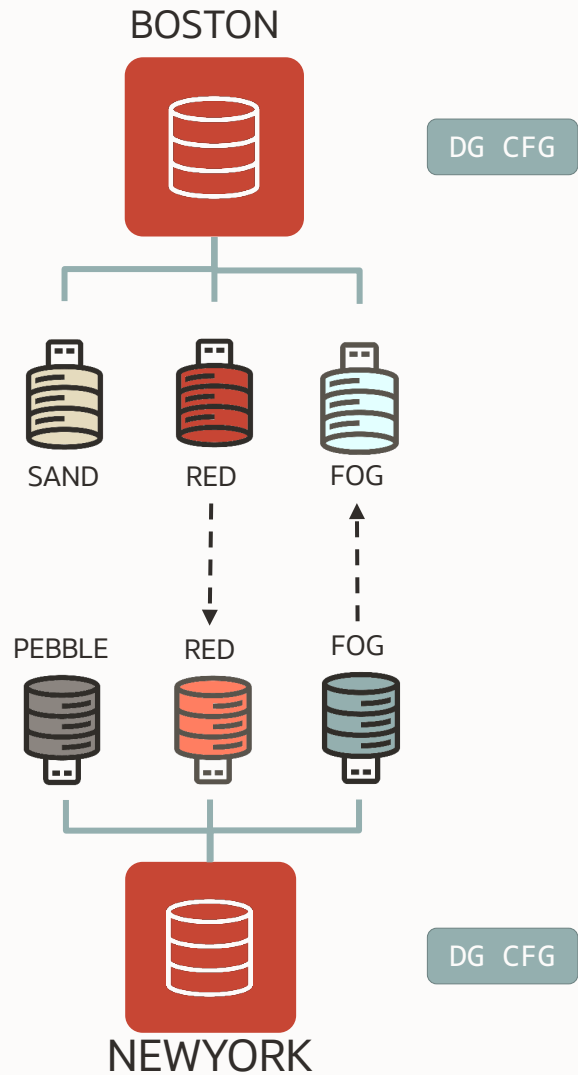
```
SQL> alter database begin backup;
```

Database altered.

```
SQL> host scp -r oracle@host2:/u01/data/NEWYORK/FOG /u01/data/BOSTON  
SQL> alter database end backup;
```

Database altered.

The Transport succeeds but the Apply is stopped



```
DGMGRL> show configuration
```

```
Configuration - boston
```

```
Protection Mode: MaxPerformance
```

```
Members:
```

```
boston - Primary database
```

```
Warning: ORA-16910: Inconsistency detected for one or more pluggable databases
```

```
newyork - Primary database in newyork configuration
```

```
Data Guard for PDB: Enabled in SOURCE and TARGET role
```

```
Configuration Status:
```

```
WARNING (status updated 46 seconds ago)
```

```
DGMGRL> show pluggable database red at newyork;
```

```
Pluggable database - red at newyork
```

```
Data Guard Role: Physical Standby
```

```
Con_ID: 4
```

```
Source: con_id 3 at boston
```

```
Transport Lag: 0 seconds (computed 22 seconds ago)
```

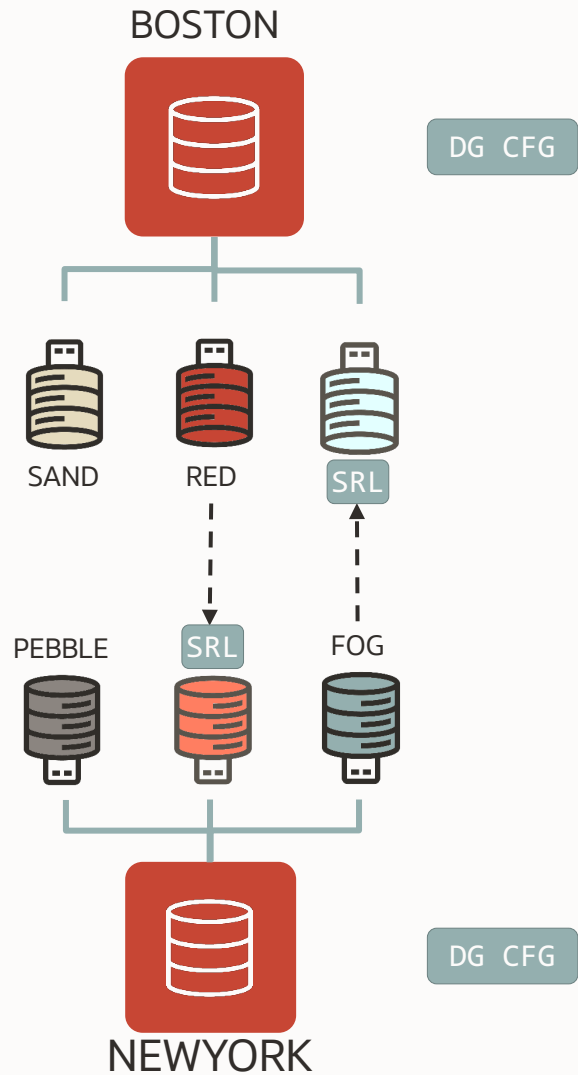
```
Intended State: APPLY-ON
```

```
Apply State: Not Running
```

```
Pluggable Database Status:
```

```
ORA-16766: Redo Apply is stopped
```

Add the Standby Redo Logs to the Target PDBs



- Once per CDB, but at PDB level, on the **TARGET PDB**

```
SQL> connect sys/<my_SYS_password>@boston as sysdba
Connected.
SQL> alter session set container=fog;
Session altered.
SQL> alter database add standby logfile thread 1
2 group 4 ('/u01/fra/BOSTON/onlinelog/standby_redo04.log') size 200M,
3 group 5 ('/u01/fra/BOSTON/onlinelog/standby_redo05.log') size 200M,
4 group 6 ('/u01/fra/BOSTON/onlinelog/standby_redo06.log') size 200M,
5 group 7 ('/u01/fra/BOSTON/onlinelog/standby_redo07.log') size 200M;

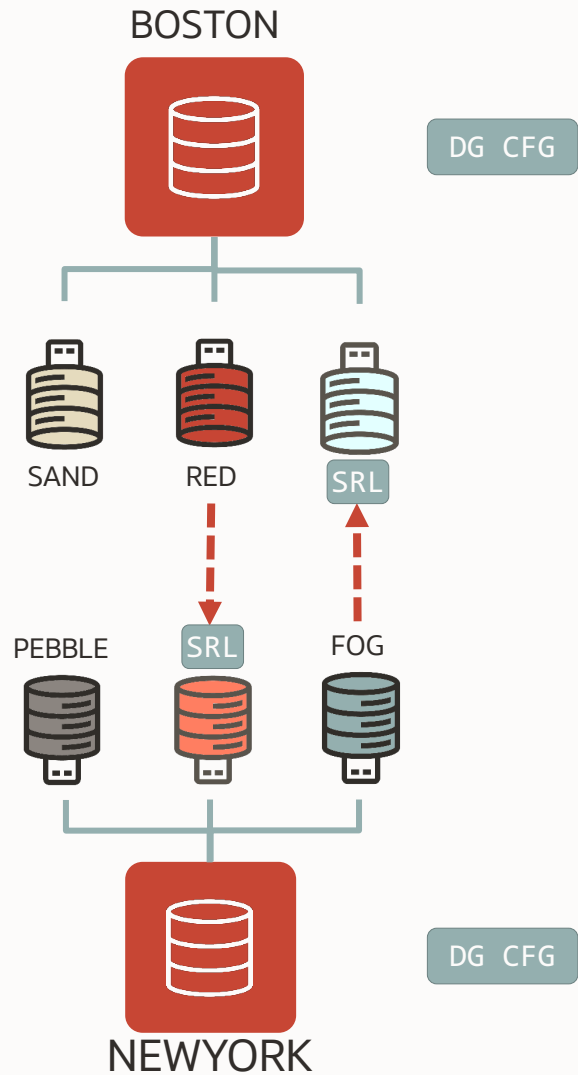
Database altered.
SQL> connect sys/<my_SYS_password>@newyork as sysdba
Connected.
SQL> alter session set container=red;
Session altered.
SQL> alter database add standby logfile thread 1
2 group 4 ('/u01/fra/NEWYORK/onlinelog/standby_redo04.log') size 200M,
3 group 5 ('/u01/fra/NEWYORK/onlinelog/standby_redo05.log') size 200M,
4 group 6 ('/u01/fra/NEWYORK/onlinelog/standby_redo06.log') size 200M,
5 group 7 ('/u01/fra/NEWYORK/onlinelog/standby_redo07.log') size 200M;

Database altered.
```

- The Standby Redo Logs are visible at PDB level, but with CON_ID=0



Start the Redo Apply



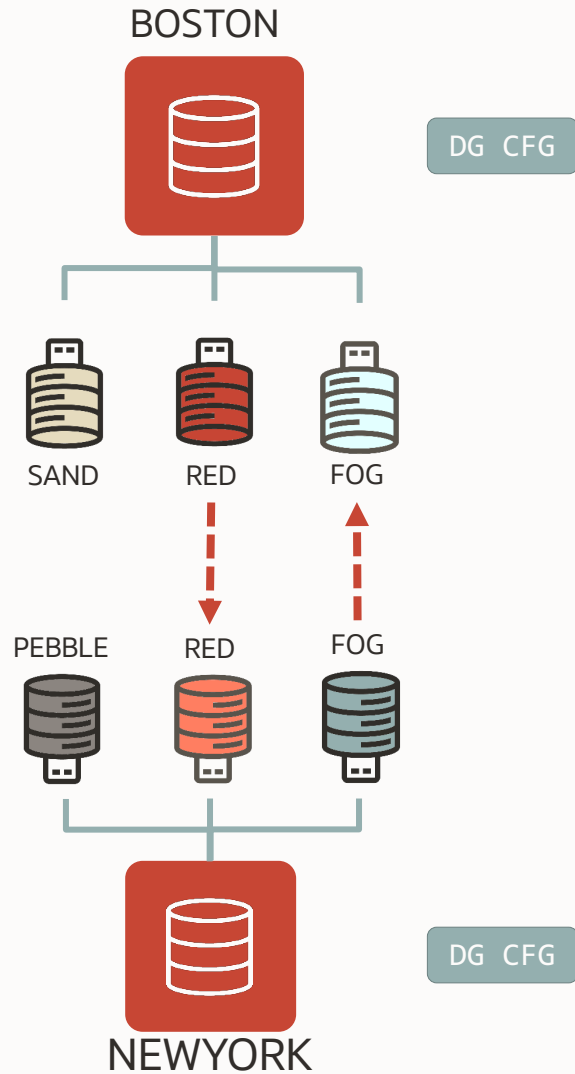
- Notice the new syntax per PDB:

```
DGMGRL> connect sys/<my_SYS_password>@boston
Connected to "boston"
Connected as SYSDBA.
DGMGRL> edit pluggable database fog at boston set state='APPLY-ON';
Succeeded.
DGMGRL> edit pluggable database red at newyork set state='APPLY-ON';
Succeeded.
```

- There is one set of SRLs to receive all the foreign redo coming from the source CDB
- The whole redo stream is sent from source to target CDB
- Only the relevant redo data is applied to the target PDBs
- Target PDBs added after the first one will use the existing standby redo logs



Verify the PDBs with DGMGRL



```
DGMGRL> show all pluggable database at boston;
```

```
Pluggable database - SAND at boston
```

```
Data Guard Role:      Not Protected
```

```
Pluggable database - RED at boston
```

```
Data Guard Role:      Primary
```

```
Con_ID:                4
```

```
Active Target:         con_id 5 at newyork
```

```
Pluggable Database Status:
```

```
SUCCESS
```

```
Pluggable database - FOG at boston
```

```
Data Guard Role:      Physical Standby
```

```
Con_ID:                5
```

```
Source:                con_id 3 at newyork
```

```
Transport Lag:         0 seconds (computed 39 seconds ago)
```

```
Intended State:        APPLY-ON
```

```
Apply State:           Running
```

```
Apply Instance:        boston
```

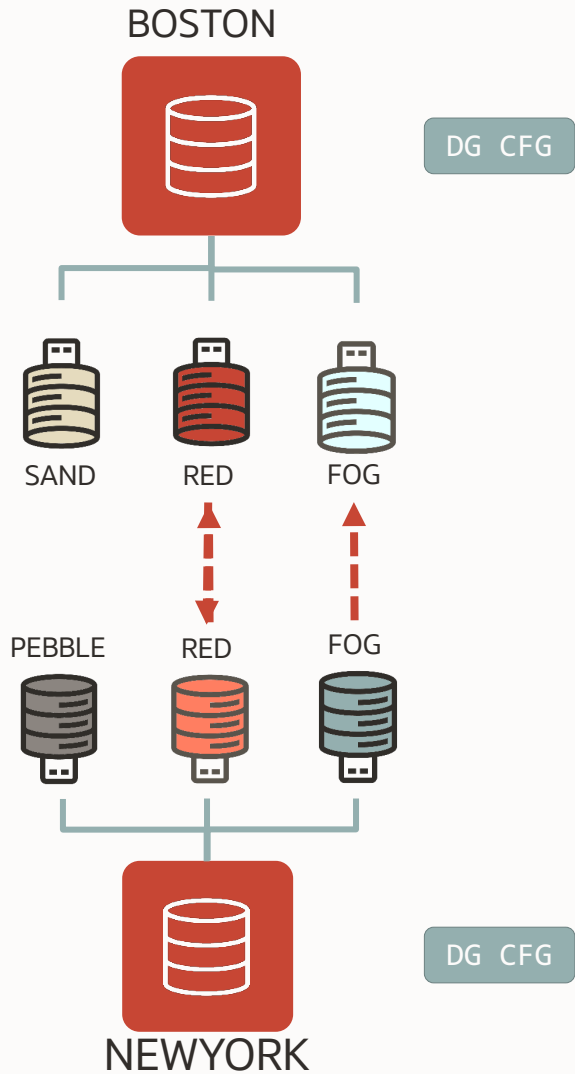
```
Average Apply Rate:    (unknown)
```

```
Real Time Query:       OFF
```

```
Pluggable Database Status:
```

```
SUCCESS
```

PDB Switchover



```

DGMGRL> ! date
Executing operating system command(s):" date"
Sat Nov 13 22:19:25 GMT 2021
DGMGRL> switchover to pluggable database red at newyork;
Verifying conditions for Switchover...

Source pluggable database is 'RED' at database 'boston'

Performing switchover NOW, please wait...

Closing pluggable database 'RED'...
Switching 'RED' to standby role...
Waiting for 'RED' to recover all redo data...
Stopping recovery at 'RED'...
Converting 'RED' to primary role...
Opening new primary 'RED'...
Waiting for redo data from new primary 'RED'...
Starting recovery at new standby 'RED'...

Switchover succeeded, new primary is "RED"
DGMGRL> ! date
Executing operating system command(s):" date"
Sat Nov 13 22:19:38 GMT 2021
    
```

13 seconds!



Hints on the future direction

Hints on the future direction

Removing the current limitations in next releases

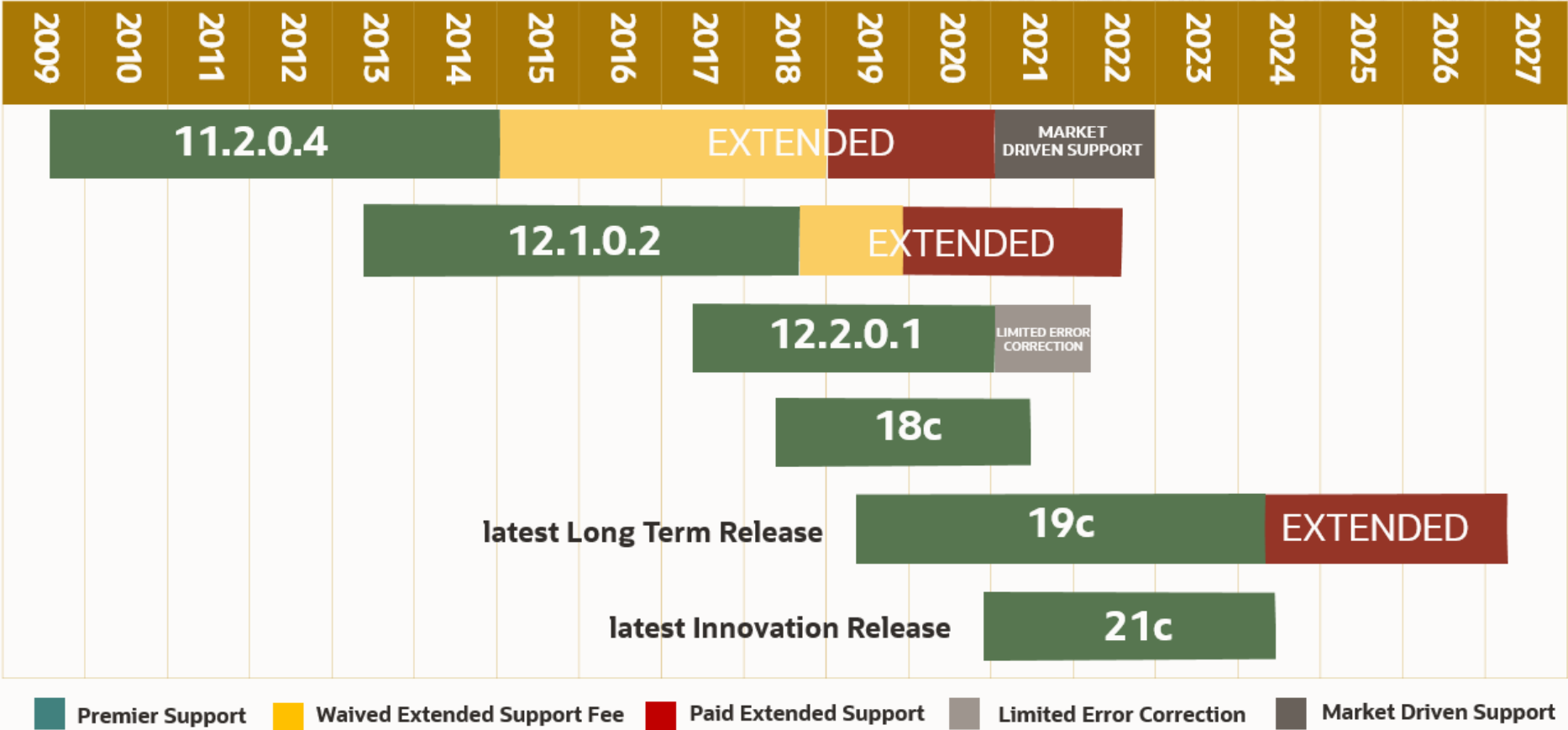


Top Priorities

- **RMAN integration for backup/restore**
- **Deletion of foreign archive logs from the FRA**
- **Temporary files on the Target PDB**
- **Easier SRL management**
- **Real-Time Query**



Database Releases and Support Timelines



Release Schedule of Current Database Releases (Doc ID 742060.1)



Conclusion



ORACLE

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



ORACLE

Francisco Munoz Alvarez

Distinguished Product Manager

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



@fcomunoz



<http://www.linkedin.com/in/francisco Munoz Alvarez>



oraclemaa.com



francisco.munoz.alvarez@oracle.com