Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

ORACLE OPEN WORLD

October 1–5, 2017
SAN FRANCISCO, CA

Oracle Active Data Guard

Build out your Oracle Maximum Availability Architecture Platinum Configuration

Larry M. Carpenter
Master Product Manager
Oracle High Availability Systems

ORACLE

Agenda

MAA Reference Architectures Review

Building Oracle Data Guard Platinum

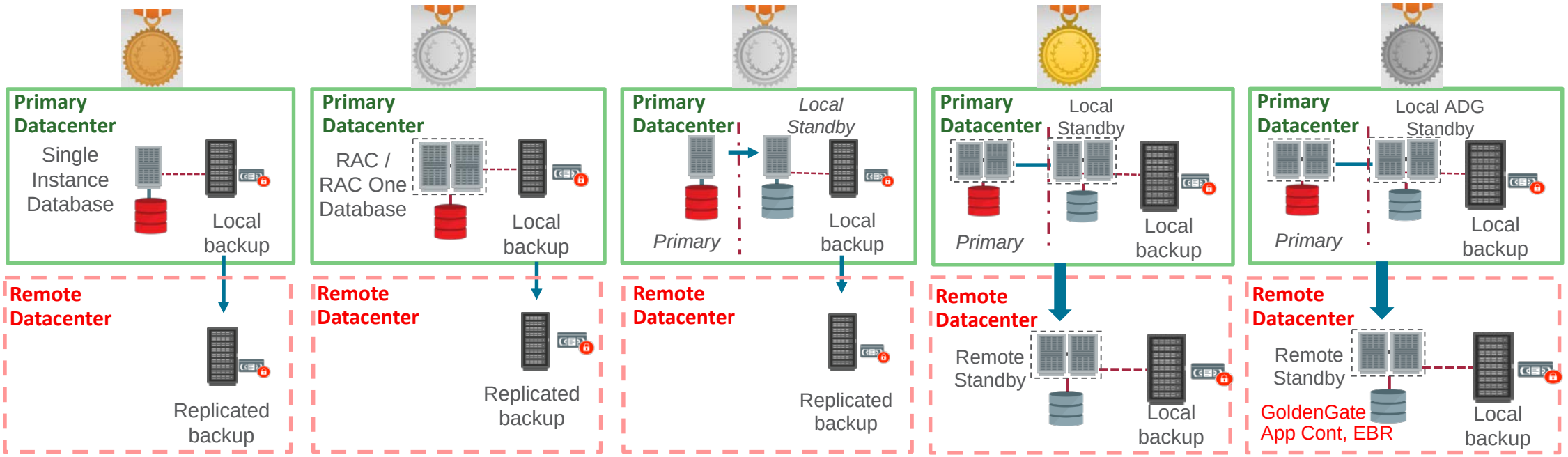
A look forward to the next release

In Closing

MAA Reference Architectures Review

You Know the MAA Reference Architectures

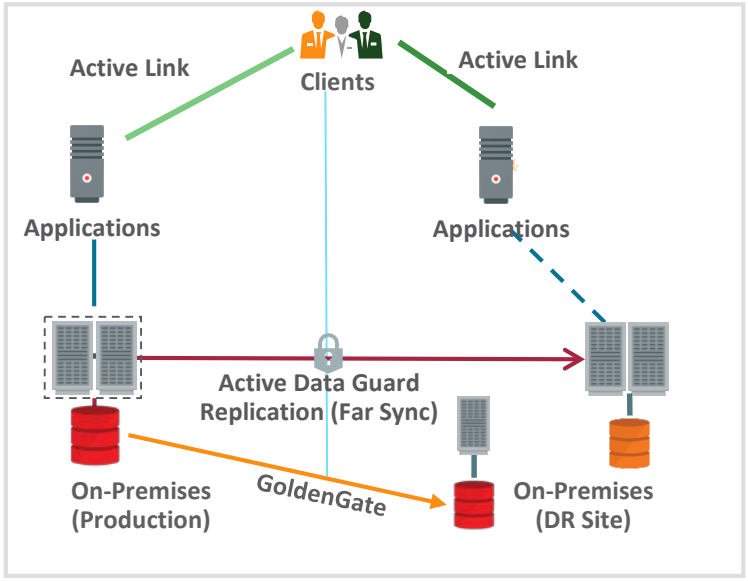
You've Seen All The Pictures



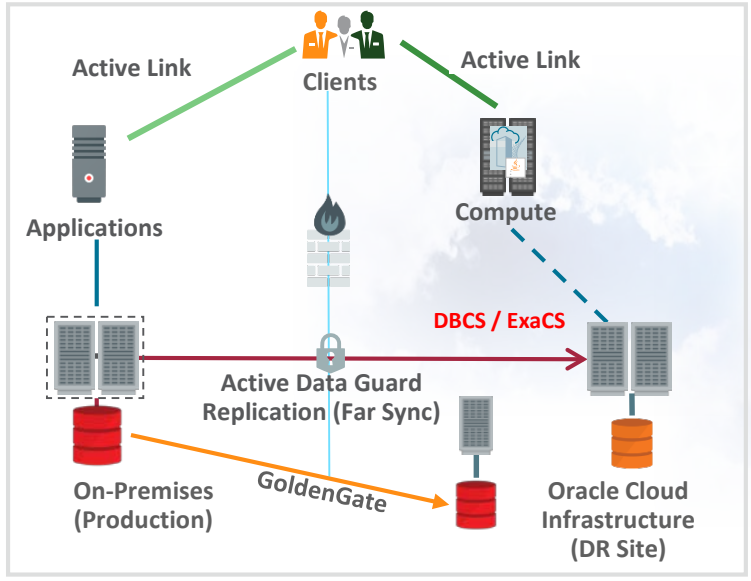
BRONZE **SILVER** **SILVER+** **GOLD** **PLATINUM**

And Where You Can Set It All Up

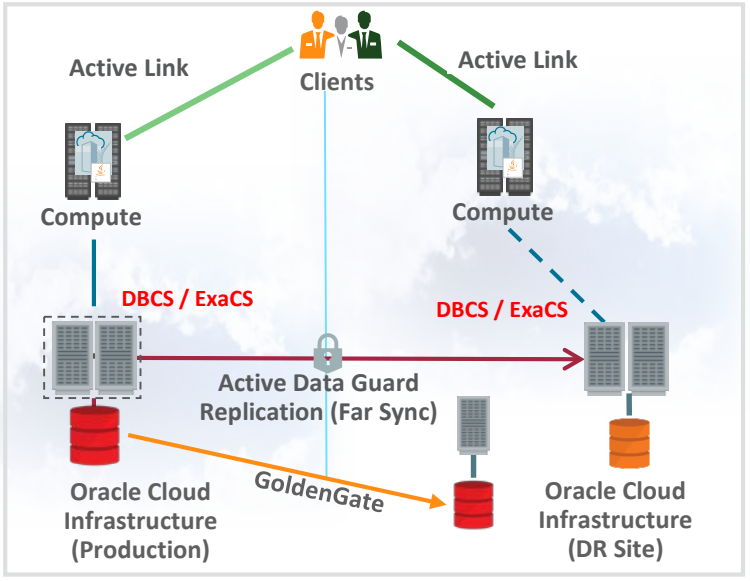
On Premises



Hybrid

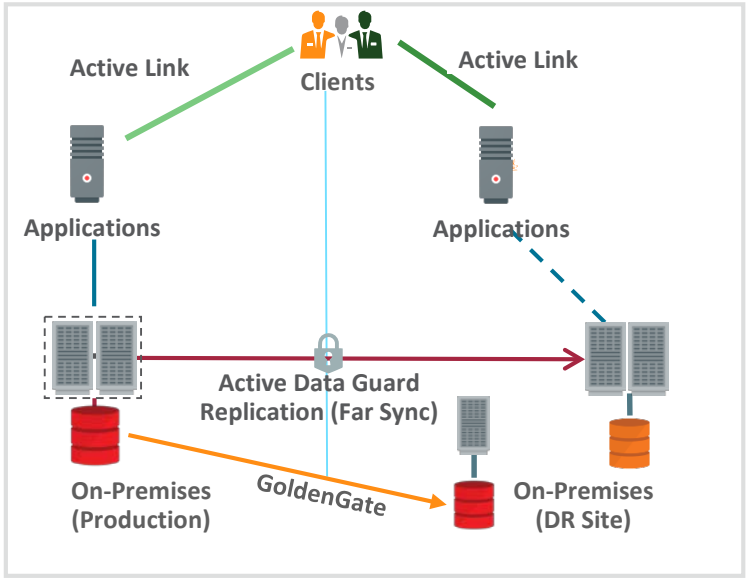


In The Cloud

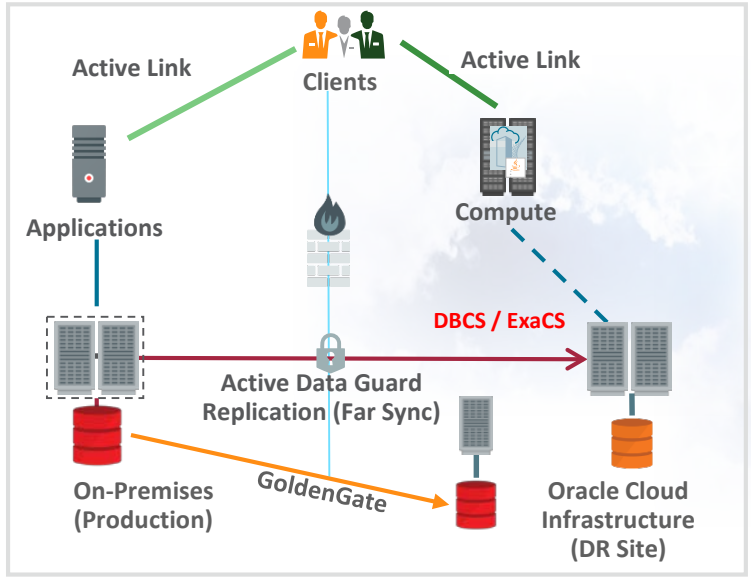


And Where You Can Set It All Up

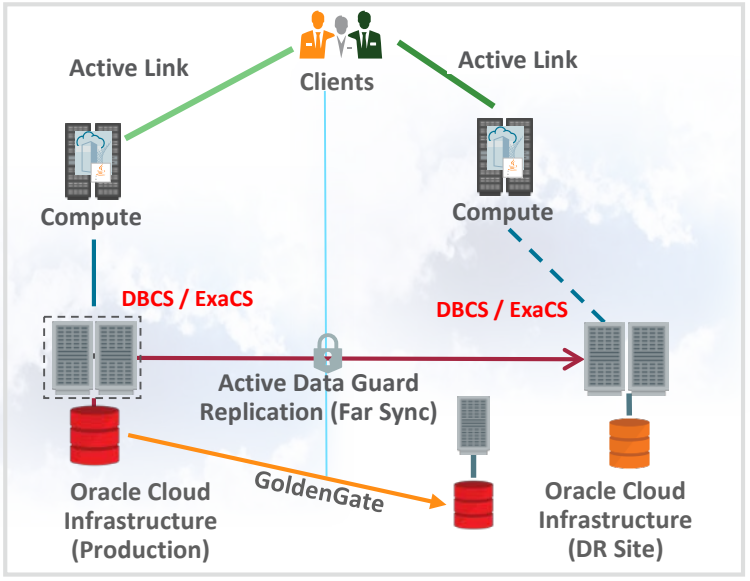
On Premises



Hybrid



In The Cloud



- The good news is – In the end it's all the same basic process to set up.
- And the requirements and objectives are also the same.

What Are Your Main Objectives?

- Recovery Point Object (RPO)
  - How much data can you afford to lose, ever?
- Recovery Time Objective (RTO)
  - How fast do you have to be back up?
- Performance Level Objective (PLO)
  - How well do you expect to perform post-event?
- Perceived Application Outage (PAO)
  - What does an application outage actually mean?
- Yes, I made the last two up. But they are just as important.

Objectives For Local Failures And Complete Disasters

- RTO=Hours/Days, RPO=Hours/Days, PLO=None?
  - **Bronze**
- RTO=Seconds/Hours/Days, RPO=Days/Hours, PLO=Reduced?
  - **Silver**
- RTO=Seconds/Days, RPO=Seconds/Minutes, PLO=Similar/Reduced?
  - **Silver+**
- RTO=Seconds/Minutes, RPO=Zero/Seconds, PLO=Similar/Reduced?
  - **Gold**
- RTO=Zero, RPO=Zero, PLO=Exactly the same all the time?
  - **Platinum**

What about Perceived Application Outage (PAO)?

- PAO=Total outage?
  - **Bronze**
- PAO=Somewhat Transparent/Total outage?
  - **Silver or Silver+**
- PAO=Mostly Transparent?
  - **Gold**
- PAO=Always Transparent?
  - **Platinum**

So, In Summary

- Platinum requires
  - RTO=Zero
  - RPO=Zero
  - PLO=Exactly the same all the time
  - PAO=Always Transparent
- Now, let's go build this beast.

Building Oracle Data Guard Platinum

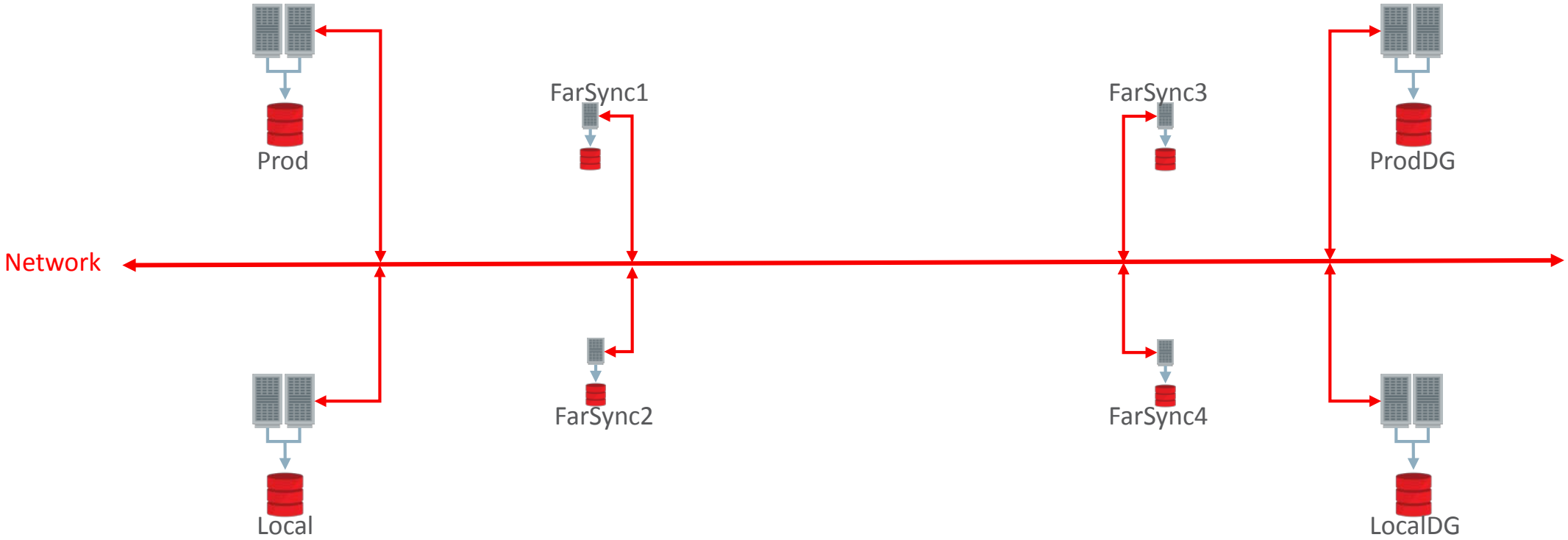
Building Platinum in Oracle Database 12.2.0.1

- Allocate at least 4 Database Servers x Number of RAC nodes desired
  - Oracle Database 12.2.0.1 and RAC, RAC One Node or Oracle Restart
  - With the required memory, CPUs and Storage
    - If these are in the Oracle Cloud Infrastructure then, depending on their use requirements, the Standbys can be fewer CPUs and Memory than the Primary as they can be increased at role change events. Storage would be the same size and same performance
- Allocate 4 Smaller Far Sync servers
  - Oracle Database 12.2.0.1 and Oracle Restart
  - Less Memory, Fewer CPUs and a lot less storage but with the same I/O capabilities
- Allocate network connectivity greater than your redo generation rates
- Configure TNSNAMES connectivity between all 8+ systems

Sizing the Far Sync Server

- Disk required for Far Sync instance is negligible.
  - Requires disk space for the Standby Redo Log files
  - Space for archive log files sufficient to handle the largest gap you expect to happen.
- Memory parameters controlled per SPFILE
- I/O speed of the SRL disks at the Far Sync is most important
  - They need to be fast if the Primary generates a lot of redo per second.
- Redo to Terminal Standby can be compressed
  - Will require more CPU and the Advanced Compression Option license

The Starting Configuration



Ensure That You Have Isolation

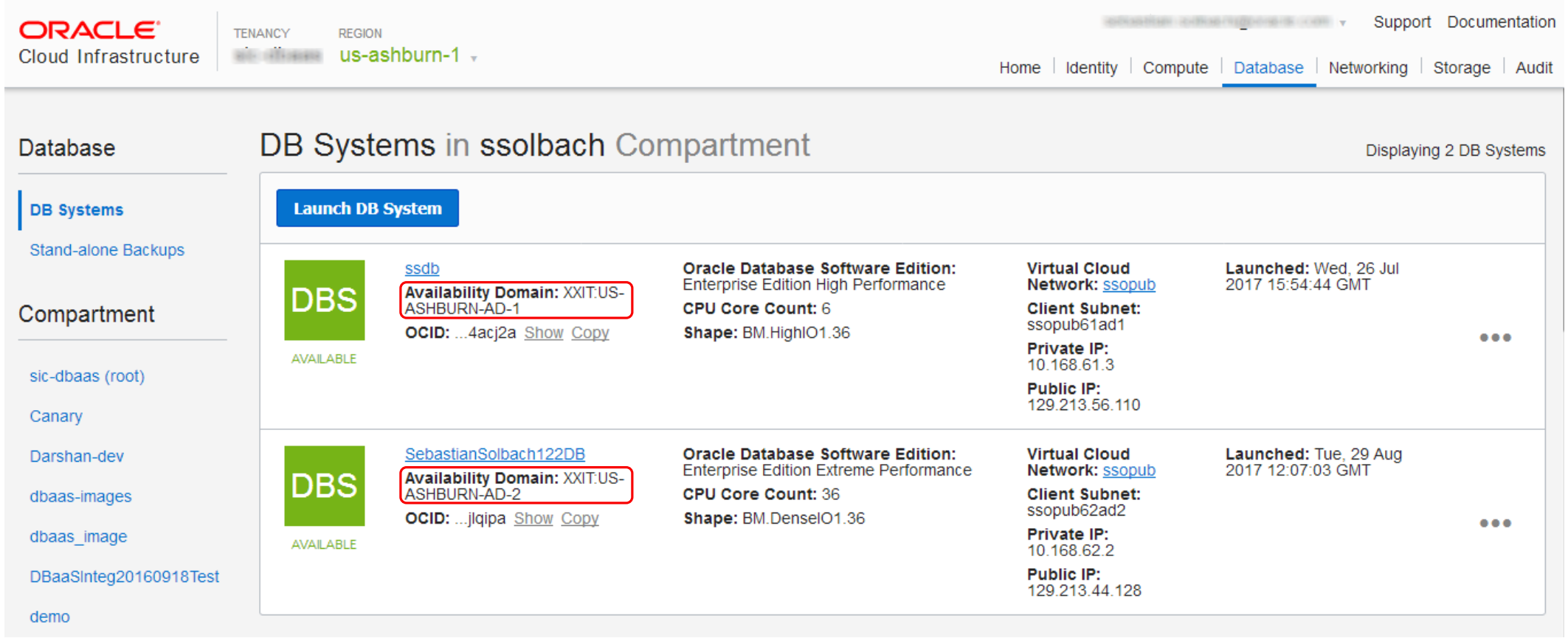
- Prod and Local systems need to be close but isolated from each other
- ProdDG and LocalDG systems need to be close and isolated as well
- But both pairs need to be remote from each other!
- The Far Sync systems also need to follow the same rules and be isolated from their respective databases.
  - You don't want a local failure to take out more than one object
    - If possible
- This means that you need at least 2 sites with 3 data centers each
  - Four or more data centers at each site would be even better

Can I do this in the Oracle Cloud Infrastructure?

- Yes.
  - The examples in this talk were executed on several systems in the Oracle Cloud Infrastructure Classic environment.
    - While it was possible to get the isolation between the main and disaster recovery sites it is not possible to ensure local isolation between the databases and the Far Sync instances in Classic.
- Fully controlled isolation is possible in the Oracle Cloud Infrastructure.
- The next two slides will show examples of how to do that.

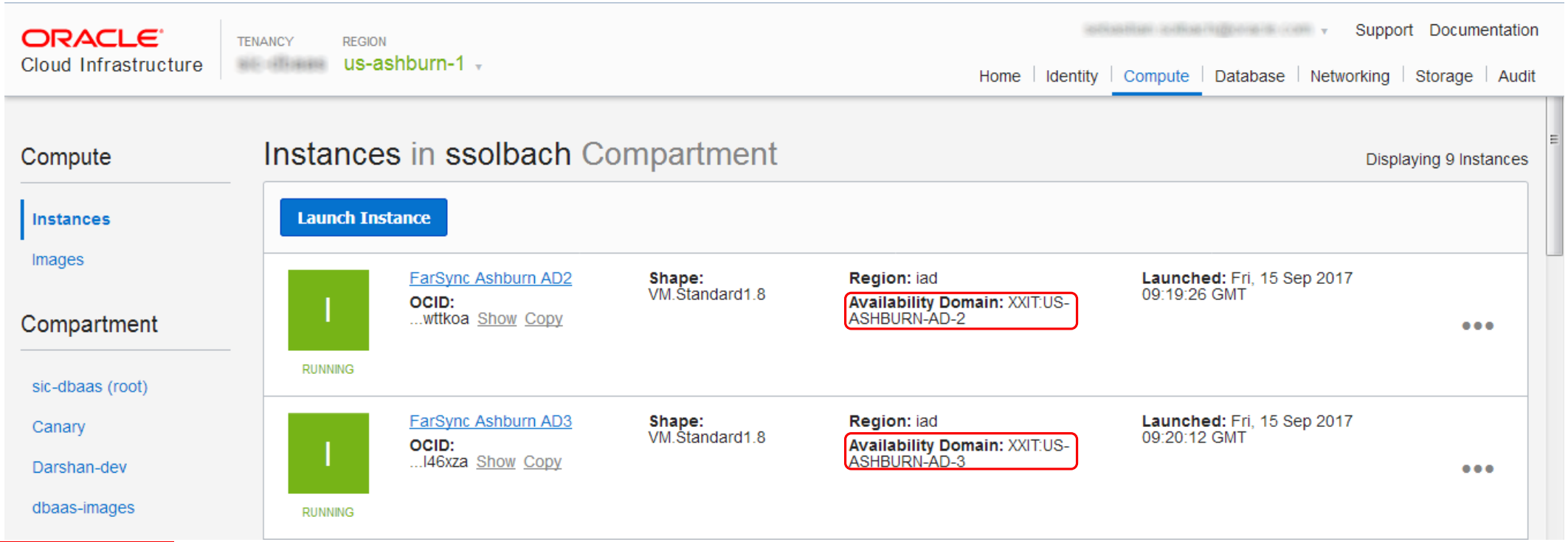
Creating Isolated Database Systems

- Create two DB Systems in one Region on two Availability Domains

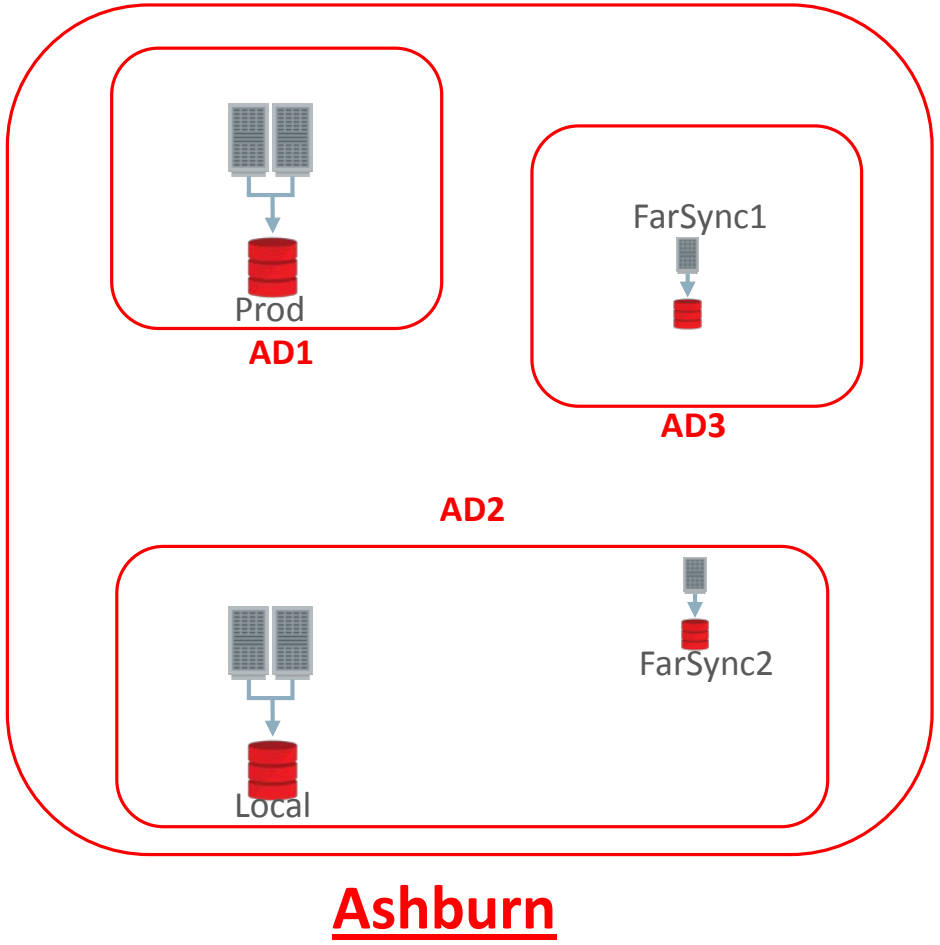


Creating Isolated Far Sync Systems

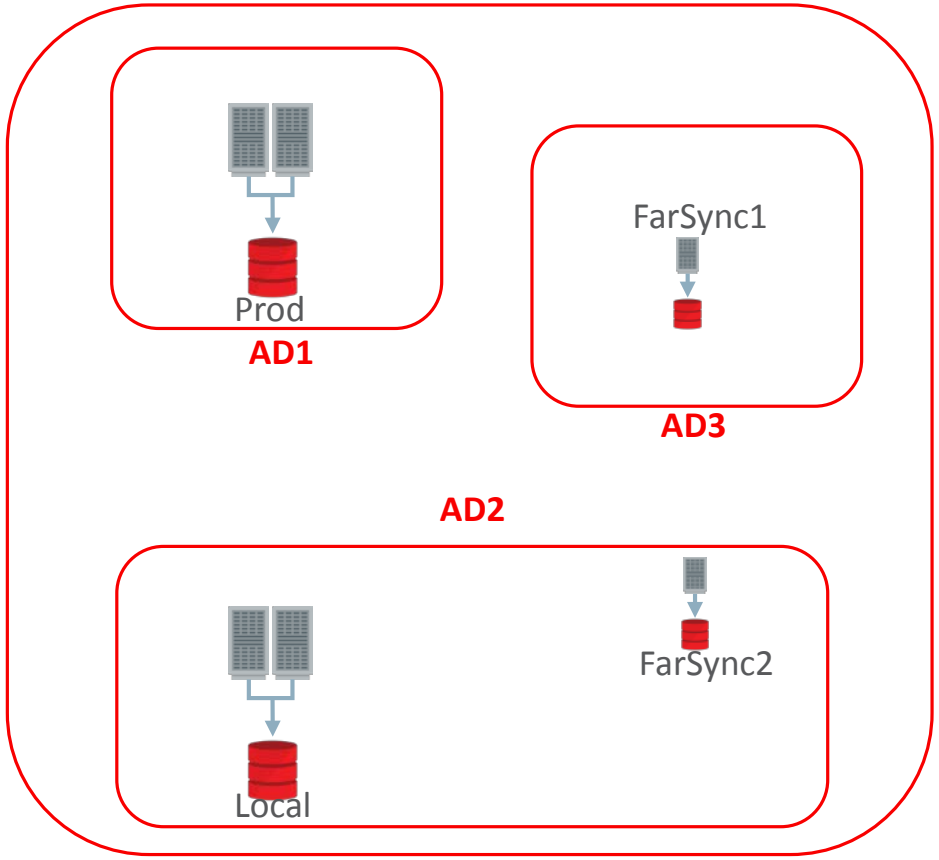
- Create two Instances in the same Region as the Database Systems
  - One on the 3rd Availability Domain (This will be the main Far Sync)
  - The other on the Local standby Availability Domain (This will be the Backup Far Sync)

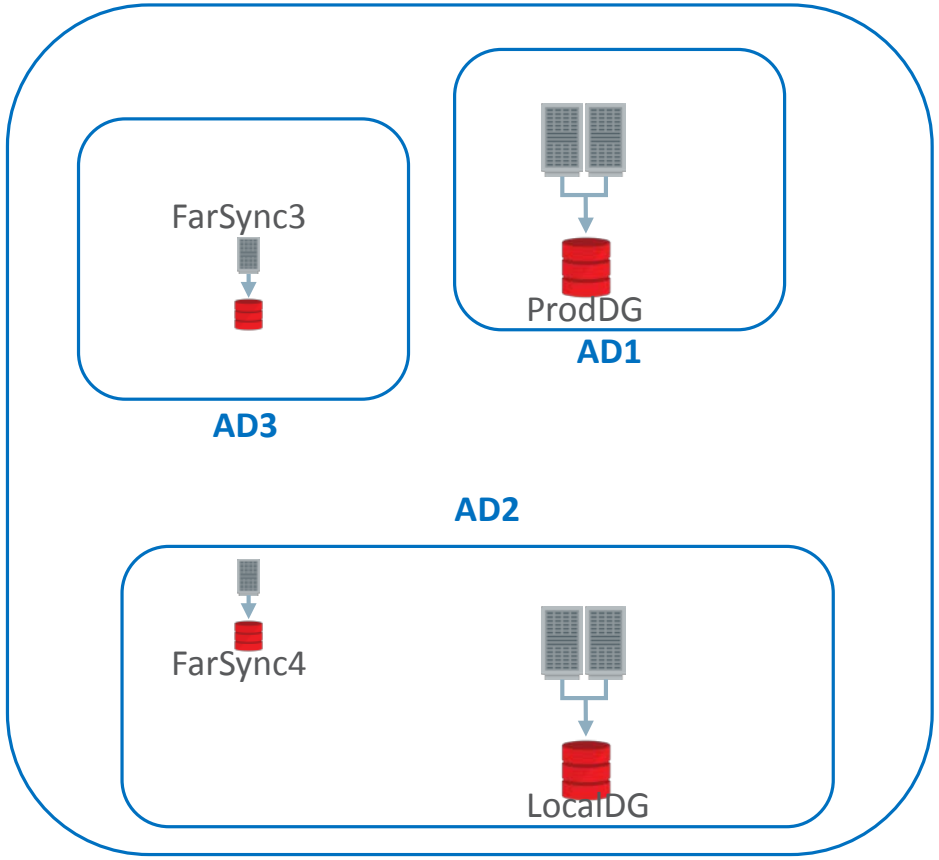


So Far You Have This



Do The Same Setup in a Different Region





Some Ground Rules

- During the creation of the Standby Databases and the Far Sync Instances
  - Remove any existing Data Guard parameters from the Primary
  - Do not set any Data Guard parameters other than the Broker ones on the next slide.
    - The Broker will set all the necessary parameters for you after which you can modify them.
    - As of Oracle Database 12c Release 1 the Broker requires that no Redo Transport parameters (where the SERVICE attribute is used) be defined prior to creating the Broker configuration.
      - 'Foreign' redo destinations for downstream GoldenGate capture are OK. (NOREGISTER attribute)
  - If using the RMAN duplication method and you create a DB_UNIQUE_NAME static entry remember to remove it when you are done.
  - It not using Oracle Restart, RAC One Node or RAC create the "DB_UNIQUE_NAME_DGMGRL" static entries on each database system.
  - Do not create **any** other static entries.

Create The Standby Databases From Prod

- On the Primary database, do the following
  - Add Standby Redo Log files
    - So they get created on each Standby Database and Far Sync.
  - Set the Broker file parameters correctly as you do not want the defaults
    - dg_broker_config_file1 = '/u01/app/oracle/product/12.2.0/dbhome_1/dbs/dr1Prod_01.dat'
    - dg_broker_config_file2 = '/u01/app/oracle/product/12.2.0/dbhome_1/dbs/dr2Prod_01.dat'
    - Change the defaults and remember they must both be visible to all RAC nodes of a database
  - Set the Broker Start parameter to TRUE
    - dg_broker_start = 'TRUE'

Create the Standby Databases

- On each Standby system create a Standby Database
  - See MOS Note “Creating a Physical Standby Database in an 11.2, 12.1, 12.2 or later environment” (Doc ID 2275154.1) for your options.
- In our example these are the standbys from the Primary Prod_01 (on Prod)
  - Prod_02 (on ProdDG)
  - Local_01 (on Local)
  - Local_02 (on LocalDG)

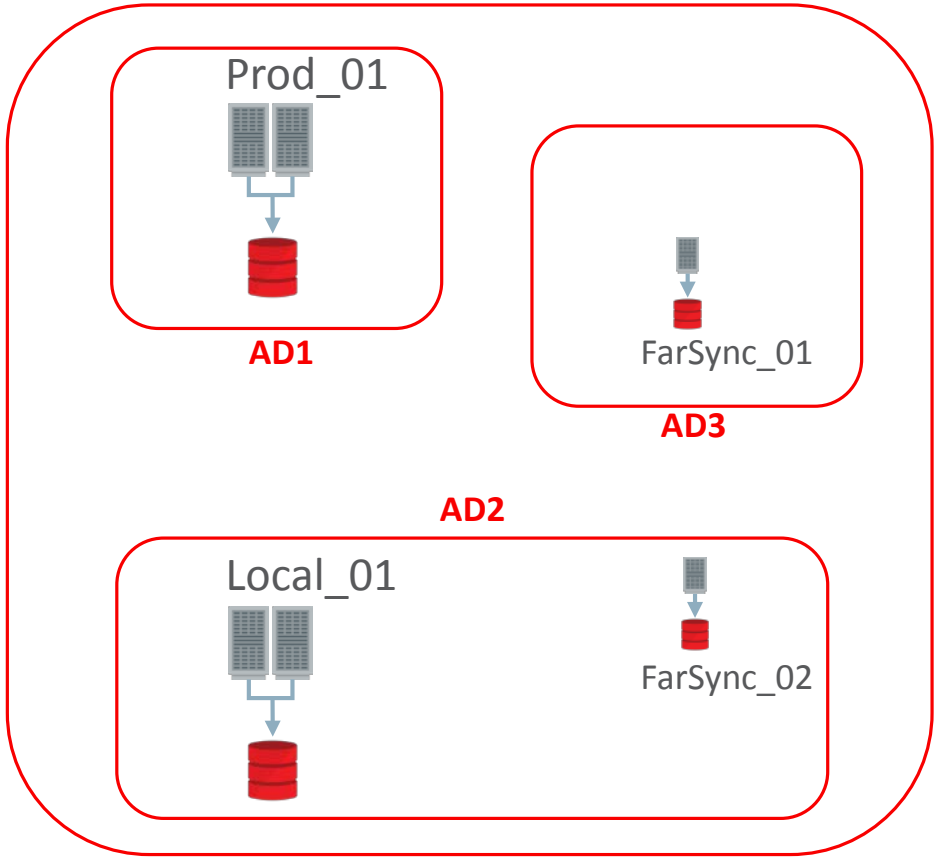
Create the Far Sync Instances

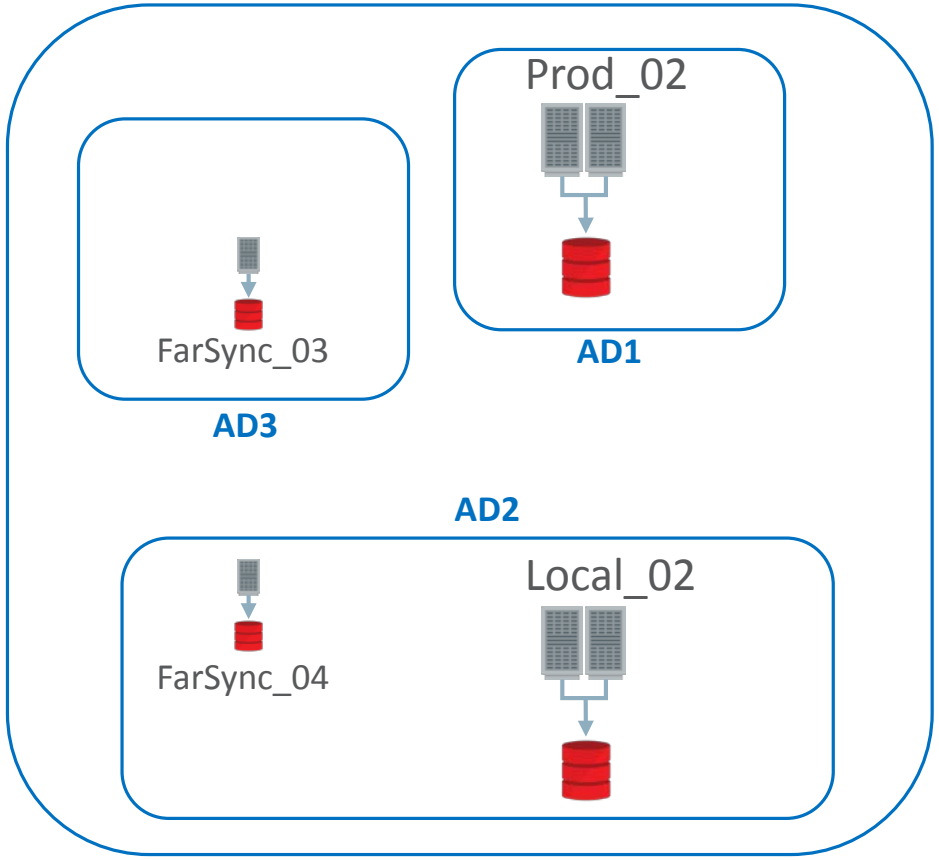
NEW IN 12.2

- Same basic procedure as the Standby Databases
  - We will create FarSync_01, FarSync_02, FarSync_03, FarSync_04
  - Create various directories, the Static entry, password file and the temp init.ora file
  - Start the Far Sync Instance NOMOUNT

```
rman
connect target sys/password@Prod_01;        # Primary
connect auxiliary sys/password@FarSync_01; # FarSync
run {
  allocate channel prmy1 type disk;
  allocate auxiliary channel stby1 type disk;

   duplicate target database for Farsync from active database
     spfile
        parameter_value_convert 'Prod','FarSync1'
        set 'db_name'='Prod'
        set 'db_unique_name'='FarSync_01'
    nofilenamecheck;}
```

Everything is Ready but not Talking to Each Other





Enter The Oracle Data Guard Broker!

- This is where you will find errors if you did not follow the rules.
  - Make sure you set the Broker parameters correctly.
    - The file location (directories) you specified must exist on each database and Far Sync
    - Both Broker files on each Clustered database must be the same 2 files for all RAC Nodes!
- Basic flow is as follows
  - Create the configuration
  - Add the 3 Standby Databases
  - Add the 4 Far Sync Instances
  - Configure Redo Transport
  - Enable the configuration
  - Ensure Zero Data Loss

Create the configuration

Must connect to the Primary database

```
[oracle@Prod-dg01 ~]$ dgmgrl sys/Password@prod_01
DGMGRL for Linux: Release 12.2.0.1.0 - Production on Wed Aug 16 18:20:55 2017

Copyright (c) 1982, 2017, Oracle and/or its affiliates.  All rights reserved.

Welcome to DGMGRL, type "help" for information.
Connected to "Prod_01"
Connected as SYSDBA.
DGMGRL> create configuration “fsc”
> as primary database is “Prod_01”
> connect identifier is “Prod_01”;

Configuration "fsc" created with primary database "Prod_01"
```

Add the 3 Standby databases

```
DGMGRL> add database “Prod_02” as connect identifier is “Prod_02”;

Database "Prod_02" added

DGMGRL> add database “Local_01” as connect identifier is “Local_01”;

Database "Local_01" added

DGMGRL> add database “Local_02” as connect identifier is “Local_02”;

Database "Local_02" added
```

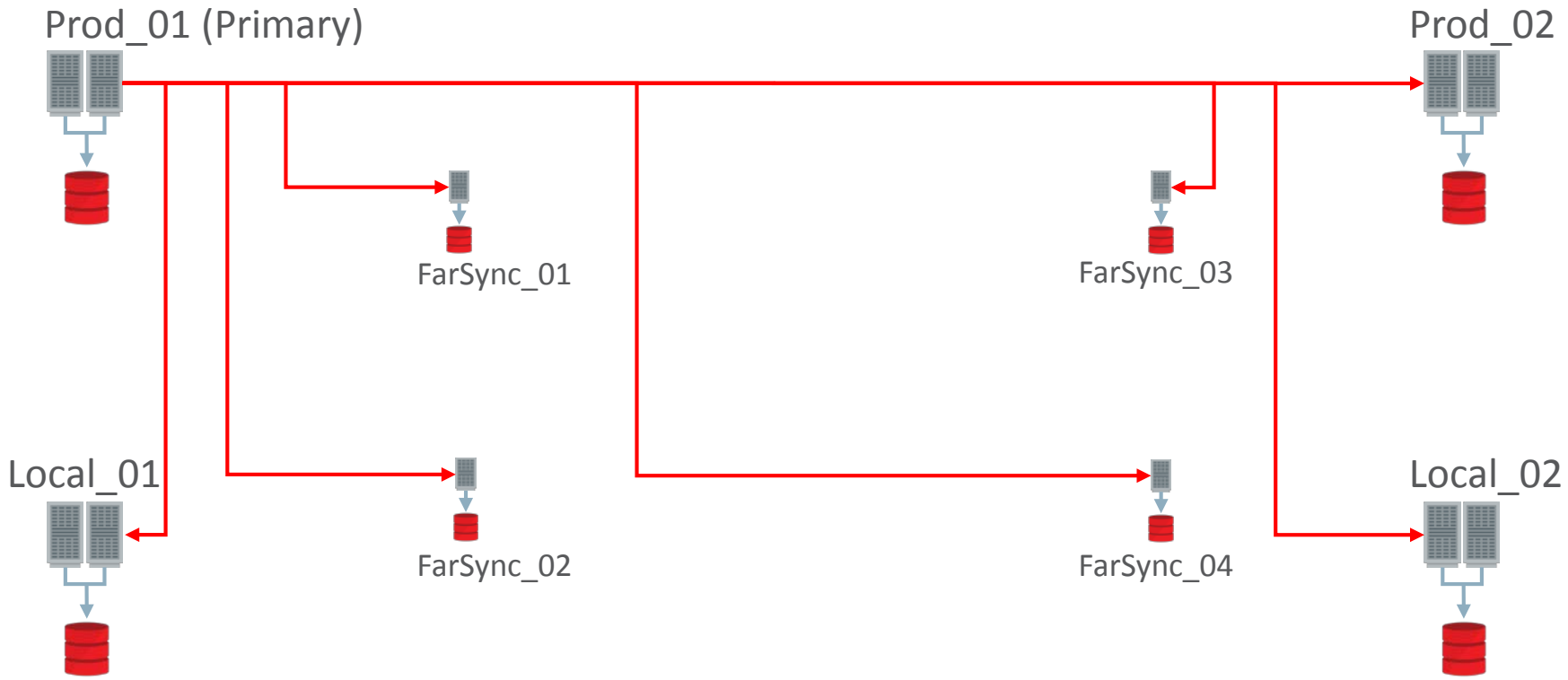
Add the 4 Far Sync Instances

```
DGMGRL> add far_sync "FarSync_01" as connect identifier is FarSync_01;

far sync instance "FarSync_01" added

DGMGRL> add far_sync "FarSync_02" as connect identifier is FarSync_02;

far sync instance "FarSync_02" added

DGMGRL> add far_sync "FarSync_03" as connect identifier is FarSync_03;

far sync instance "FarSync_03" added

DGMGRL> add far_sync "FarSync_04" as connect identifier is FarSync_04;

far sync instance "FarSync_04" added
```

The Current Configuration

```
DGMGRL> show configuration;

Configuration - fsc

  Protection Mode: MaxPerformance
  Members:
  Prod_01    - Primary database
    Prod_02    - Physical standby database
    Local_02   - Physical standby database
    Local_01   - Physical standby database
    FarSync_01 - Far sync instance
    FarSync_02 - Far sync instance
    FarSync_03 - Far sync instance
    FarSync_04 - Far sync instance

Fast-Start Failover: DISABLED

Configuration Status:
DISABLED
```

This is Too Much Talking Going On (If Enabled)



Fix Redo Transport – Ashburn to Phoenix

```
DGMGRL> edit far_sync "FarSync_01" set property redoroutes=‘
(Prod_01 : Local_01, (FarSync_03, FarSync_04, Prod_02, Local_02))
(Local_01 : Prod_01, (FarSync_03, FarSync_04, Prod_02, Local_02))
(Prod_02 : Prod_01, Local_01) (Local_02 : Prod_01, Local_01)';
Property "redoroutes" updated

DGMGRL> edit far_sync "FarSync_02" set property redoroutes=‘
(Prod_01 : Local_01, (FarSync_03, FarSync_04, Prod_02, Local_02))
(Local_01 : Prod_01, (FarSync_03, FarSync_04, Prod_02, Local_02))
(Prod_02 : Prod_01, Local_01) (Local_02 : Prod_01, Local_01)';
Property "redoroutes" updated

DGMGRL> edit database "Prod_01" set property redoroutes=‘
(Local: (FarSync_01 SYNC, FarSync_02 SYNC, Prod_02, Local_01 SYNC, Local_02))';
Property "redoroutes" updated

DGMGRL> edit database "Local_01" set property redoroutes=‘
(Local: (FarSync_01 SYNC, FarSync_02 SYNC, Prod_02, Prod_01 SYNC, Local_02))';
Property "redoroutes" updated
```

Fix Redo Transport – Phoenix to Ashburn

```
DGMGRL> edit far_sync "FarSync_03" set property redoroutes=‘
(Prod_02 : Local_02, (FarSync_01, FarSync_02, Prod_01, Local_01))
(Local_02 : Prod_02, (FarSync_01, FarSync_02, Prod_01, Local_01))
(Prod_01 : Prod_02, Local_02) (Local_01 : Prod_02, Local_02)';
Property "redoroutes" updated

DGMGRL> edit far_sync "FarSync_04" set property redoroutes=‘
(Prod_02 : Local_02, (FarSync_01, FarSync_02, Prod_01, Local_01))
(Local_02 : Prod_02, (FarSync_01, FarSync_02, Prod_01, Local_01))
(Prod_01 : Prod_02, Local_02) (Local_01 : Prod_02, Local_02)';
Property "redoroutes" updated

DGMGRL> edit database "Prod_02" set property redoroutes=‘
(Local: (FarSync_03 SYNC, FarSync_04 SYNC, Prod_01, Local_01, Local_02 SYNC))';
Property "redoroutes" updated

DGMGRL> edit database "Local_02" set property redoroutes=‘
(Local: (FarSync_03 SYNC, FarSync_04 SYNC, Prod_02 SYNC, Prod_01, Local_01))';
Property "redoroutes" updated
```

Enable the Configuration and Raise the Protection Mode

- Enable the Configuration

```
DGMGRL> enable configuration;
Enabled.
```

- Since we have set SYNC to the Far Sync we can raise the Protection Mode

```
DGMGRL> edit configuration set protection mode as MaxAvailability;
Succeeded.
```

The ‘Enabled’ Configuration

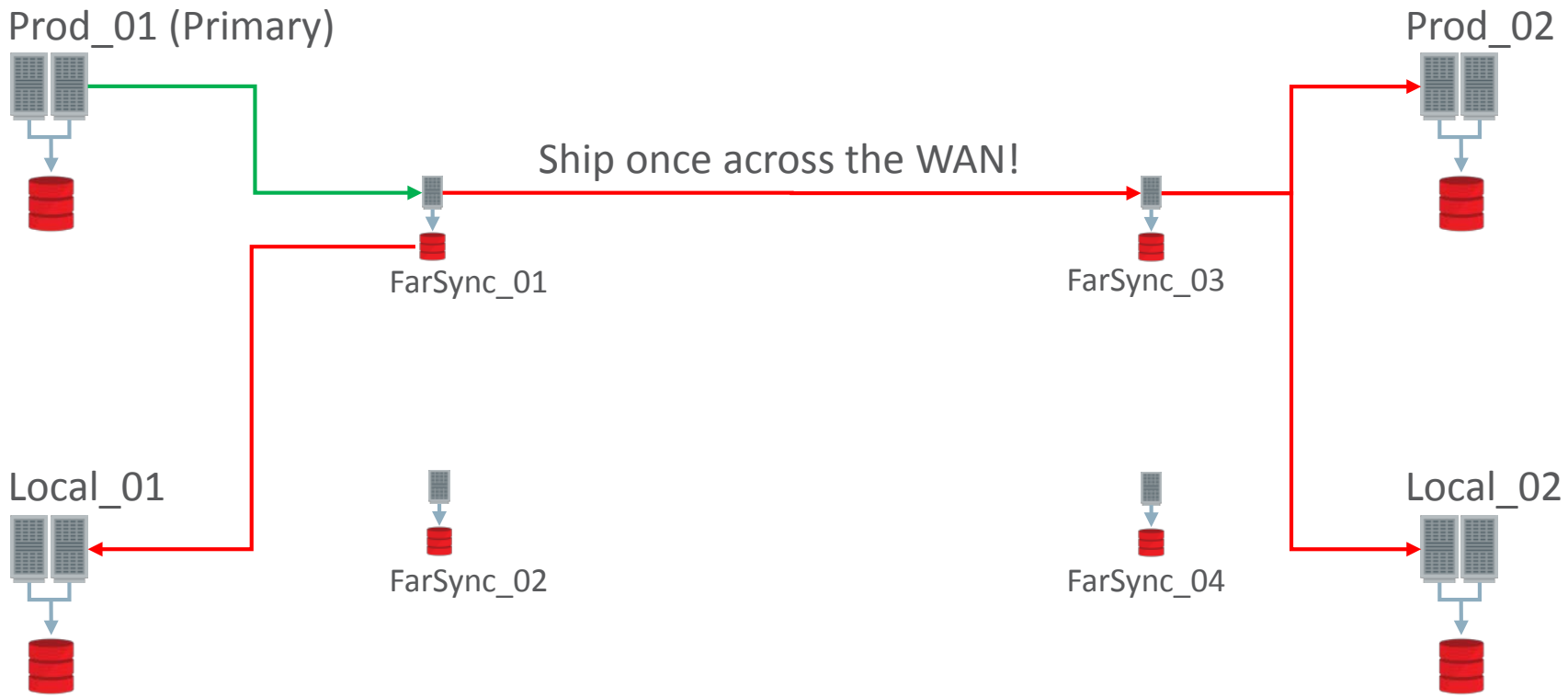
```
DGMGRL> show configuration;
Configuration - fsc

  Protection Mode: MaxAvailability
  Members:
  Prod_01    - Primary database
    FarSync_01 - Far sync instance
      Local_01   - Physical standby database
      FarSync_03 - Far sync instance
        Prod_02    - Physical standby database
        Local_02   - Physical standby database

  Members Not Receiving Redo:
  FarSync_02 - Far sync instance (alternate of FarSync_01)
  FarSync_04 - Far sync instance (alternate of FarSync_03)

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS   (status updated 11 seconds ago)
```

The ‘Enabled’ Configuration

Current Priority 1

```
DGMGRL> show configuration;
Configuration - fsc

  Protection Mode: MaxAvailability
  Members:
  Prod_01    - Primary database
    FarSync_01 - Far sync instance
      Local_01   - Physical standby database
      FarSync_03 - Far sync instance
        Prod_02    - Physical standby database
        Local_02   - Physical standby database

  Members Not Receiving Redo:
  FarSync_02 - Far sync instance (alternate of FarSync_01)
  FarSync_04 - Far sync instance (alternate of FarSync_03)

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS   (status updated 11 seconds ago)
```

Now We're Talking!



Prod_01 Primary

```
DGMGRL> show configuration when primary is "Prod_01";

Configuration when Prod_01 is primary - fsc

  Members:
  Prod_01    - Primary database
    FarSync_01 - Far sync instance
      Local_01   - Physical standby database
      FarSync_03 - Far sync instance
        Prod_02    - Physical standby database
        Local_02   - Physical standby database
      FarSync_04 - Far sync instance (alternate of FarSync_03)
        Prod_02    - Physical standby database
        Local_02   - Physical standby database
      Prod_02    - Physical standby database (alternate of FarSync_03)
      Local_02   - Physical standby database (alternate of FarSync_03)
    FarSync_02 - Far sync instance (alternate of FarSync_01)
      Local_01   - Physical standby database
      FarSync_03 - Far sync instance
        Prod_02    - Physical standby database (alternate of FarSync_03)
        Local_02   - Physical standby database (alternate of FarSync_03)
      FarSync_04 - Far sync instance (alternate of FarSync_03)
        Prod_02    - Physical standby database (alternate of FarSync_03)
        Local_02   - Physical standby database (alternate of FarSync_03)
      Prod_02    - Physical standby database (alternate of FarSync_03)
      Local_02   - Physical standby database (alternate of FarSync_03)
    Prod_02    - Physical standby database (alternate of FarSync_03)
    Local_01   - Physical standby database (alternate of FarSync_01)
    Local_02   - Physical standby database (alternate of FarSync_03)
```

Prod_01 Primary

```
DGMGRL> show configuration when primary is "Prod_01";

Configuration when Prod_01 is primary - fsc

  Members:
  Prod_01    - Primary database
    FarSync_01 - Far sync instance
      Local_01   - Physical standby database
      FarSync_03 - Far sync instance
        Prod_02    - Physical standby database
        Local_02   - Physical standby database
      FarSync_04 - Far sync instance (alternate of FarSync_03)
        Prod_02    - Physical standby database
        Local_02   - Physical standby database
    Prod_02    - Physical standby database (alternate of FarSync_03)
    Local_02   - Physical standby database (alternate of FarSync_03)
    FarSync_02 - Far sync instance (alternate of FarSync_01)
      Local_01   - Physical standby database
      FarSync_03 - Far sync instance
        Prod_02    - Physical standby database (alternate of FarSync_03)
        Local_02   - Physical standby database (alternate of FarSync_03)
      FarSync_04 - Far sync instance (alternate of FarSync_03)
        Prod_02    - Physical standby database (alternate of FarSync_03)
        Local_02   - Physical standby database (alternate of FarSync_03)
    Prod_02    - Physical standby database (alternate of FarSync_03)
    Local_02   - Physical standby database (alternate of FarSync_03)
  Prod_02    - Physical standby database (alternate of FarSync_03)
  Local_01   - Physical standby database (alternate of FarSync_01)
  Local_02   - Physical standby database (alternate of FarSync_03)
```

Priority 1

Prod_01 Primary

```
DGMGRL> show configuration when primary is "Prod_01";

Configuration when Prod_01 is primary - fsc

  Members:
  Prod_01    - Primary database
    FarSync_01 - Far sync instance
      Local_01   - Physical standby database
      FarSync_03 - Far sync instance
        Prod_02    - Physical standby database
        Local_02   - Physical standby database
      FarSync_04 - Far sync instance (alternate of FarSync_03)
        Prod_02    - Physical standby database
        Local_02   - Physical standby database
      Prod_02    - Physical standby database (alternate of FarSync_03)
      Local_02   - Physical standby database (alternate of FarSync_03)
    FarSync_02 - Far sync instance (alternate of FarSync_01)
      Local_01   - Physical standby database
      FarSync_03 - Far sync instance
        Prod_02    - Physical standby database (alternate of FarSync_03)
        Local_02   - Physical standby database (alternate of FarSync_03)
      FarSync_04 - Far sync instance (alternate of FarSync_03)
        Prod_02    - Physical standby database (alternate of FarSync_03)
        Local_02   - Physical standby database (alternate of FarSync_03)
      Prod_02    - Physical standby database (alternate of FarSync_03)
      Local_02   - Physical standby database (alternate of FarSync_03)
    Prod_02    - Physical standby database (alternate of FarSync_03)
    Local_01   - Physical standby database (alternate of FarSync_01)
    Local_02   - Physical standby database (alternate of FarSync_03)
```

Priority 8

Prod_01 Primary

```
DGMGRL> show configuration when primary is "Prod_01";

Configuration when Prod_01 is primary - fsc

  Members:
  Prod_01     - Primary database
    FarSync_01 - Far sync instance
      Local_01   - Physical standby database
      FarSync_03 - Far sync instance
        Prod_02    - Physical standby database
        Local_02   - Physical standby database
      FarSync_04 - Far sync instance (alternate of FarSync_03)
        Prod_02    - Physical standby database
        Local_02   - Physical standby database
      Prod_02    - Physical standby database (alternate of FarSync_03)
      Local_02   - Physical standby database (alternate of FarSync_03)
    FarSync_02 - Far sync instance (alternate of FarSync_01)
      Local_01   - Physical standby database
      FarSync_03 - Far sync instance
        Prod_02    - Physical standby database (alternate of FarSync_03)
        Local_02   - Physical standby database (alternate of FarSync_03)
      FarSync_04 - Far sync instance (alternate of FarSync_03)
        Prod_02    - Physical standby database (alternate of FarSync_03)
        Local_02   - Physical standby database (alternate of FarSync_03)
      Prod_02    - Physical standby database (alternate of FarSync_03)
      Local_02   - Physical standby database (alternate of FarSync_03)
    Prod_02     - Physical standby database (alternate of FarSync_03)
    Local_01    - Physical standby database (alternate of FarSync_01)
    Local_02    - Physical standby database (alternate of FarSync_03)
```

Final Priority 8

Local_01 Primary

```
DGMGRL> show configuration when primary is "Local_01";

Configuration when Local_01 is primary - fsc

  Members:
  Local_01   - Primary database
    FarSync_01 - Far sync instance
      Prod_01    - Physical standby database
      FarSync_03 - Far sync instance
        Prod_02    - Physical standby database
        Local_02   - Physical standby database
      FarSync_04 - Far sync instance (alternate of FarSync_03)
        Prod_02    - Physical standby database
        Local_02   - Physical standby database
      Prod_02    - Physical standby database (alternate of FarSync_03)
      Local_02   - Physical standby database (alternate of FarSync_03)
    FarSync_02 - Far sync instance (alternate of FarSync_01)
      Prod_01    - Physical standby database
      FarSync_03 - Far sync instance
        Prod_02    - Physical standby database (alternate of FarSync_03)
        Local_02   - Physical standby database (alternate of FarSync_03)
      FarSync_04 - Far sync instance (alternate of FarSync_03)
        Prod_02    - Physical standby database (alternate of FarSync_03)
        Local_02   - Physical standby database (alternate of FarSync_03)
      Prod_02    - Physical standby database (alternate of FarSync_03)
      Local_02   - Physical standby database (alternate of FarSync_03)
    Prod_02    - Physical standby database (alternate of FarSync_03)
    Prod_01    - Physical standby database (alternate of FarSync_01)
    Local_02   - Physical standby database (alternate of FarSync_03)
```

Prod_02 Primary

```
DGMGRL> show configuration when primary is "Prod_02";

Configuration when Prod_02 is primary - fsc

  Members:
  Prod_02     - Primary database
    FarSync_03 - Far sync instance
      Local_02   - Physical standby database
      FarSync_01 - Far sync instance
        Prod_01    - Physical standby database
        Local_01   - Physical standby database
      FarSync_02 - Far sync instance (alternate of FarSync_01)
        Prod_01    - Physical standby database
        Local_01   - Physical standby database
      Prod_01    - Physical standby database (alternate of FarSync_01)
      Local_01   - Physical standby database (alternate of FarSync_01)
    FarSync_04 - Far sync instance (alternate of FarSync_03)
      Local_02   - Physical standby database
      FarSync_01 - Far sync instance
        Prod_01    - Physical standby database (alternate of FarSync_01)
        Local_01   - Physical standby database (alternate of FarSync_01)
      FarSync_02 - Far sync instance (alternate of FarSync_01)
        Prod_01    - Physical standby database (alternate of FarSync_01)
        Local_01   - Physical standby database (alternate of FarSync_01)
      Prod_01    - Physical standby database (alternate of FarSync_01)
      Local_01   - Physical standby database (alternate of FarSync_01)
    Prod_01     - Physical standby database (alternate of FarSync_01)
    Local_01    - Physical standby database (alternate of FarSync_01)
    Local_02    - Physical standby database (alternate of FarSync_03)
```

Local_02 Primary

```
DGMGRL> show configuration when primary is "Local_02";

Configuration when Local_02 is primary - fsc

  Members:
  Local_02   - Primary database
    FarSync_03 - Far sync instance
      Prod_02    - Physical standby database
      FarSync_01 - Far sync instance
        Prod_01    - Physical standby database
        Local_01   - Physical standby database
      FarSync_02 - Far sync instance (alternate of FarSync_01)
        Prod_01    - Physical standby database
        Local_01   - Physical standby database
      Prod_01    - Physical standby database (alternate of FarSync_01)
      Local_01   - Physical standby database (alternate of FarSync_01)
    FarSync_04 - Far sync instance (alternate of FarSync_03)
      Prod_02    - Physical standby database
      FarSync_01 - Far sync instance
        Prod_01    - Physical standby database (alternate of FarSync_01)
        Local_01   - Physical standby database (alternate of FarSync_01)
      FarSync_02 - Far sync instance (alternate of FarSync_01)
        Prod_01    - Physical standby database (alternate of FarSync_01)
        Local_01   - Physical standby database (alternate of FarSync_01)
      Prod_01    - Physical standby database (alternate of FarSync_01)
      Local_01   - Physical standby database (alternate of FarSync_01)
    Prod_02    - Physical standby database (alternate of FarSync_03)
    Prod_01    - Physical standby database (alternate of FarSync_01)
    Local_01   - Physical standby database (alternate of FarSync_01)
```

Can You Do This Without The Broker?

- Yes, but don't.
  - The Broker dynamically adds the following LOG_ARCHIVE_DEST_*n* parameters
    - Five parameters Prod_01
    - Five parameters on FarSync_01 and FarSync_02
    - Two parameters on FarSync_03 and FarSync_04
  - Without the Broker you would need to configure all the parameters yourself
    - Five parameters on Prod_01
    - Five parameters on Local_01
    - Five parameters on Prod_02
    - Five parameters on Local_02
    - Seven parameters on FarSync_01 and FarSync_02
    - Seven parameters on FarSync_03 and FarSync_04

Will Oracle Cloud Infrastructure Do This For Me?

- We're working on it.
  - HA or DR standbys are possible today
    - Soon you will be able to do a combination of both automatically
  - Automatic Failover (FSFO) is coming (Local or Remote)
  - Remote Zero Data Loss is coming (Far Sync)
- Building out Data Guard Platinum is on the books

So, What Have We Achieved?

- RTO=Zero?
  - Planned - “Zero” with Service/Session Draining, reasonably transparent – See PAO
  - Unplanned - Disconnect/Loss of Inflight Transaction
    - Even with Fast_Start Failover (FSFO) - Local or Remote
- RPO=Zero?
  - Planned – Zero
  - Unplanned – Zero, unless there is a total failure of the current production site
- PLO=Exactly the same all the time? - Check!
- PAO=Always Transparent?
  - Planned – Check, but with a ‘pause’ : Unplanned – Visible and not Zero

Current RTO Capabilities

- Planned events require a switchover (The ‘pause’ but no disconnect)
  - O/S Upgrades
  - Grid Infrastructure
  - RDBMS (Standby First Patching and Rolling Upgrades)
  - Application Upgrades (No switchover required here)
    - When done directly in the Primary database using Edition Based Redefinition (EBR)
- Unplanned events
  - Current Read/Write users experience a transaction loss and disconnect
  - Readers on the Active Data Guard Failover Standby continue with a pause.
  - Readers on bystander Active Data Guard Standbys see no interruption

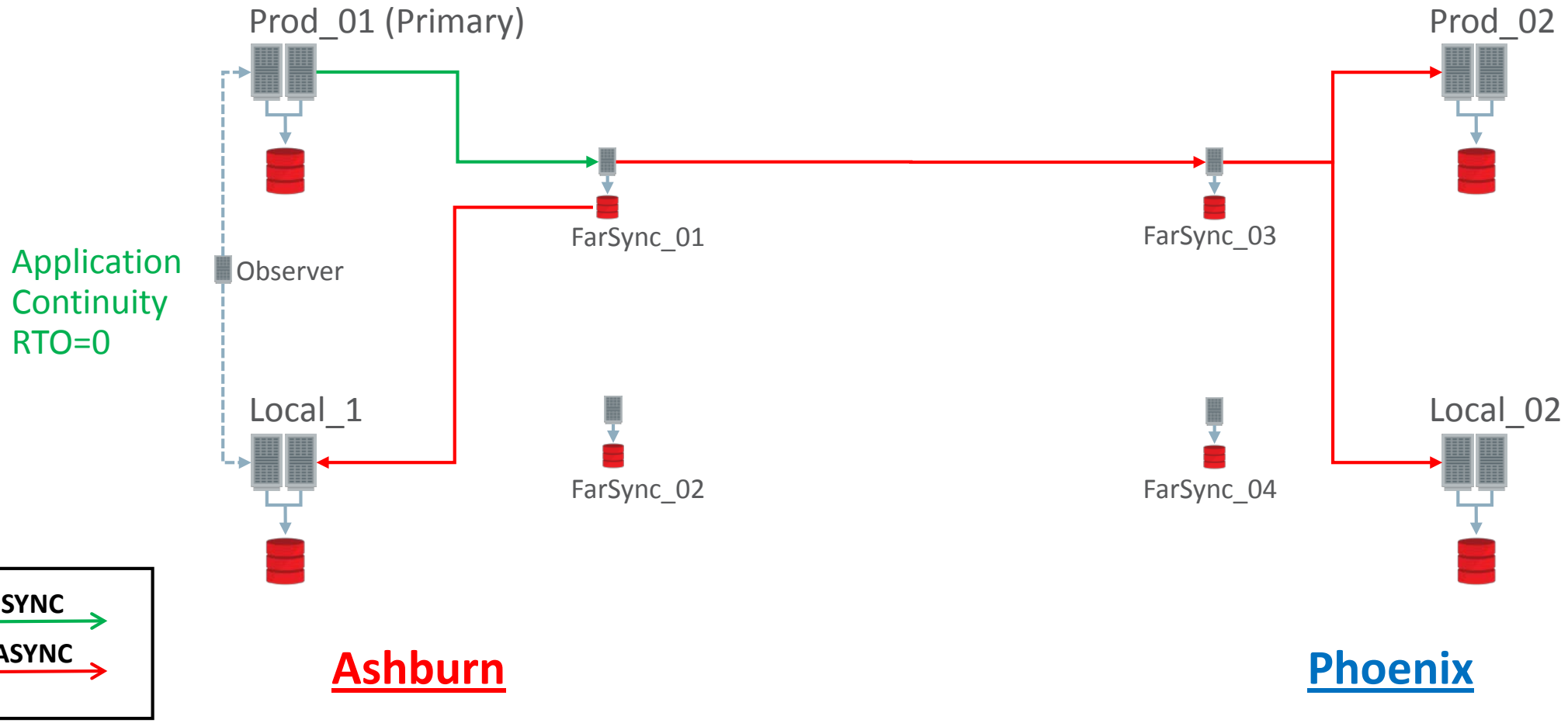
Current RPO Capabilities

- Planned Events
  - RPO is always Zero since a Data Guard Switchover is always Zero Data Loss
- Unplanned Events
  - Production Failure
    - RPO is Zero
  - Production Total Site Failure
    - Zero, as long as you protect the ‘Production’ Far Syncs from any failure
      - Otherwise a failover to the remote standby will result in some data loss.

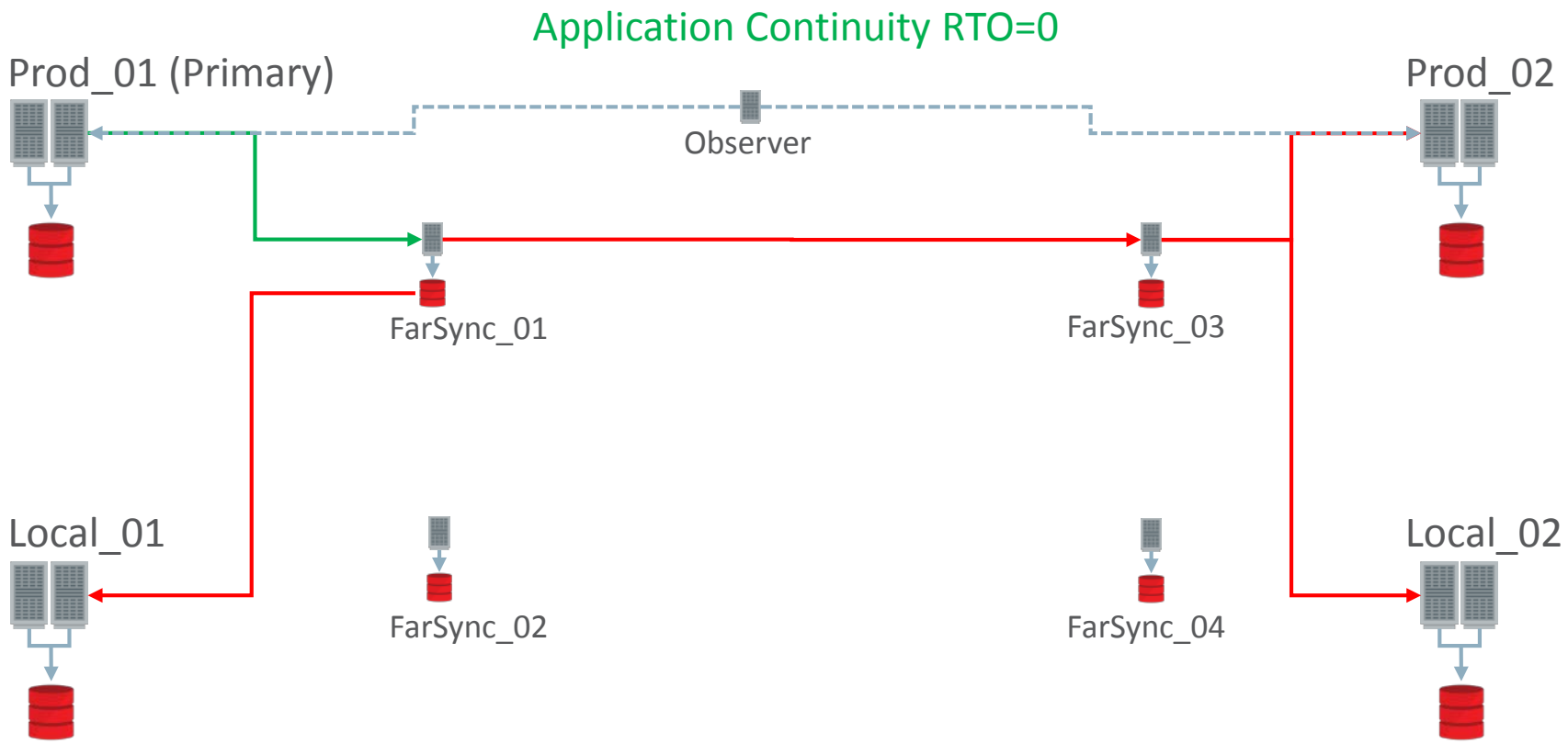
Improve RTO & PAO with Application Continuity

- Combined with Sync Fast Start Failover (FSFO), Application Continuity can render an unplanned failure transparent to the application
  - RTO = Zero
  - RPO = Zero
- However FSFO can only provide this locally or remotely with our configuration, not both.
  - This means Prod_01 to Local_01 (Locally) or Prod_01 to Prod_02 (Remotely).
    - Prod_01 to Prod_02 requires that the FarSync_01 or FarSync_02 were synchronized at failure time.
    - Prod_02 to Prod_01 requires the same for FarSync_03 or FarSync_04.

With Application Continuity - Local



With Application Continuity – Or Remote



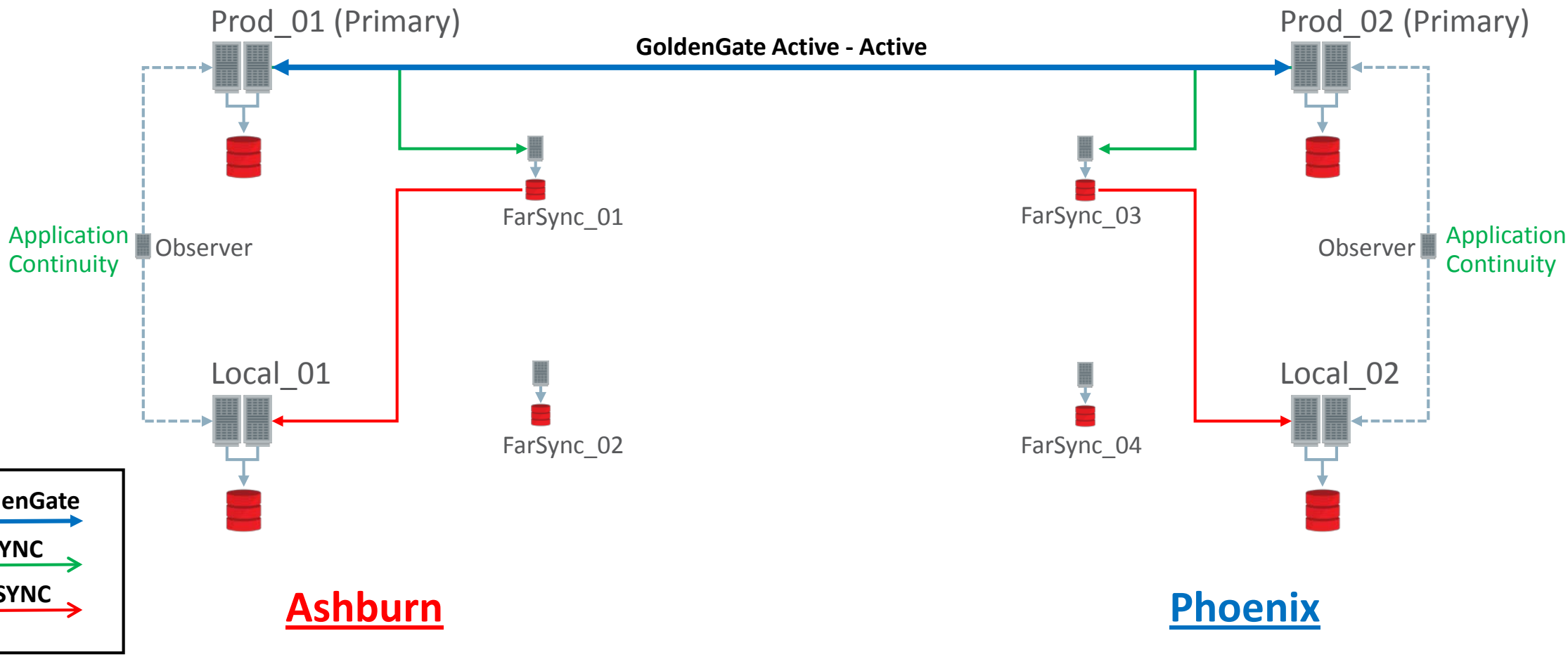
Extended FSFO Features in Oracle Database 12.2

- Multiple Observers can be started
  - Maximum of 3
  - Only one is the master at anytime
  - Changing the master requires communication with the Primary
- Multiple Standby Targets can also be defined
  - If the current target fails, another standby in the list is chosen
  - Changing the target requires communication with the Primary
  - So Local_01, Prod_02 and Local_02 can be targets, in that order
    - If Local_01 fails FSFO will be set up between Prod_01 and Prod_02
  - Requires that the application tier can see both for Application Continuity to work

How Do We “Kick It Up a Notch”?

- Currently the Standby databases are available for Read Transactions
  - Ancillary DML is also possible but not full application transactions
- To improve RTO you need to introduce GoldenGate
  - No [Active – Passive] as it buys you nothing.
  - Full [Active – Active] is required, complete with conflict detection and resolution
- Planned Events
  - Makes PAO fully transparent for Planned Events
    - Users move as they become idle
- Unplanned Events
  - Total geographic failures, 50% of users lose their transaction and disconnect

Create Two Distinct Broker Configurations



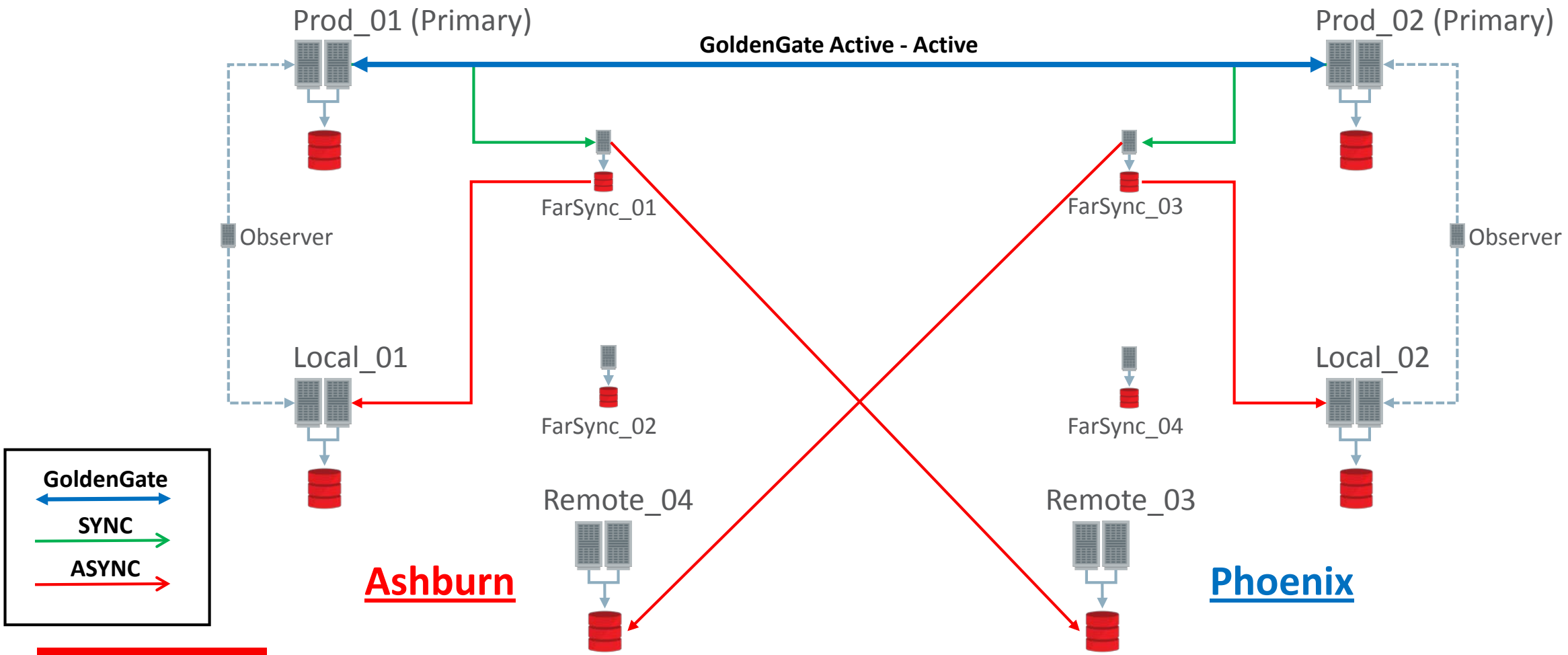
Now Where Are We?

- RTO=Zero?
  - Planned – Zero, Users move when ready, no disconnect, no transaction loss
  - Prod_01 or Prod_02 Unplanned – Zero, With FSFO and Application Continuity
  - Geographic Unplanned – 50% disconnect and lose transactions
- RPO=Zero?
  - Planned – Zero
  - Unplanned – Zero for any single failure. But not zero for a geographical failure.
- PLO=Exactly the same all the time? - Check!
- PAO=Always Transparent?
  - Planned? Check - Unplanned? Not Zero through all failures

What is the Next Step?

- To get the RTO to 100% zero
  - We need Application Continuity between the two GoldenGate databases.
    - We're not there yet.
- To get the RPO to 100% zero
  - Both sides of the configuration need to have zero data loss standbys outside their geographical locations.
    - This means two more standbys.
    - Remember - Zero, as long as you protect the 'Production' Far Syncs from any failure
      - Otherwise a failover to the remote standby will result in some data loss.

Add the Two Standby Databases



Building Platinum in Oracle Database 12.2.0.1

- From Bronze to Platinum is quite an evolution
- When your Boss says “I want RTO=0 and RPO=0”
  - You need to understand what that actually means.
  - Ask detailed questions
    - What exactly does RTO of Zero mean?
    - Is there any flexibility in the RPO of Zero?
    - What do they consider the PLO to be?
    - And is there any variation in what we consider the PAO?
      - You’ll need to introduce that one!
  - And respond accordingly
    - What they are asking for is not cheap

A Look Forward to the Next Release

Review - Data Guard 12.1 New Features Summary

- Data Guard
  - SYSDG Role for Data Guard specific operations
  - Multitenant Database Standbys
  - Default Real Time Apply
  - Transparent Online Data file movement
  - Single SQL*Plus Switchover command
  - No Primary instance shutdown for switchover
  - FastSync (SYNC NOAFFIRM)
  - Enhanced Extended Datatype Support for upgrades
  - Logical Standby support for additional data types
    - XMLType data for all storage models, Oracle Spatial, Oracle Multimedia, Oracle Text, Objects and Collections (including VARRAYs and nested collections), Database File System (DBFS), XDB, Oracle SecureFiles (deduplication), and User-defined types.
  - DBMS_SCHEDULER support for Rolling Upgrades
  - Broker VALIDATE DATABASE capability
  - Enhanced Broker configuring with RedoRoutes
  - Broker Resumable switchover
  - New Broker Observer properties and capabilities
  - Broker support of Cascading Redo destinations

- Active Data Guard
  - Real Time Cascading Redo destinations
  - WAN Distance Zero Data Loss with Far Sync
  - Global Temporary Table DML on a standby
  - Sequences (global and session) on a standby
  - DBMS_ROLLING automated rolling upgrades
  - Support for Oracle Application Continuity
  - Support for Oracle Global Data Services

Review - Data Guard 12.2 New Features Summary

- Data Guard
  - Use DBCA and EMCLI to create Standbys
  - Use RMAN and Enterprise Manager to create Far Sync Instances in addition to Standbys
  - Rest Interface to the Data Guard Broker
  - Chef tools to create and manage Data Guard standbys
  - Scripting with the Broker DGMGRL command line interface
  - Easily creating Subset Standby databases in Multitenant
  - Migrate and Failover PDBs with the Broker
  - Automatic Password File management
  - Easy repair of No Logging operations
  - Data File Block Comparison between Primary and Standby
  - Multiple FSFO Targets and Observers
  - FSFO in Maximum Protection mode
  - Zero Data Loss Failovers in any mode with Storage Failures
  - Enhanced control of Alternate destinations
  - Convert to TDE easily with little or no downtime

- Active Data Guard
  - Using all your Standby Instances to apply redo
  - Data Guard Broker Support and involvement in DBMS_ROLLING controlled rolling upgrades
  - Use the In Memory Column Store on your Standby
  - Use AWR and SQL Tuning Advisor to diagnose and tune Redo Apply and Query issues on your Active Data Guard Standby
  - Move users transparently during role change with Session Draining on the Primary and Preserving existing connections at the Active Data Guard Standby
  - More protection with Auto Block repair enhancements

Q & A

Safe Harbor Statement

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Integrated Cloud

Applications & Platform Services


ORACLE®