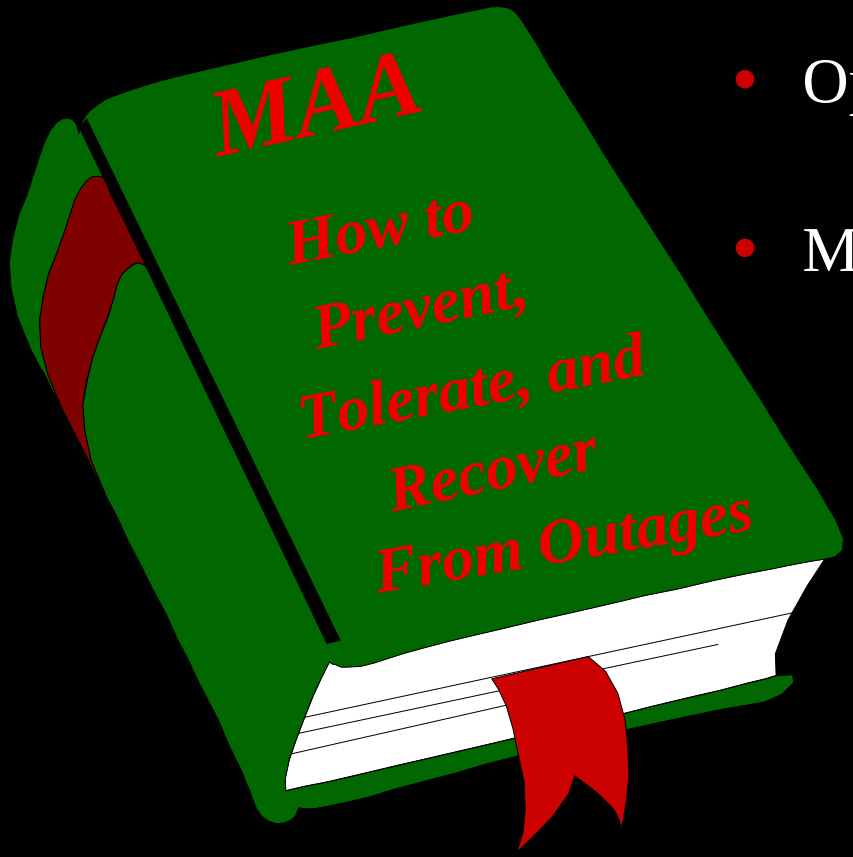
DataGuard Tuning & Best Practices

Mike Smith
Oracle

Maximum Availability Architecture (MAA)

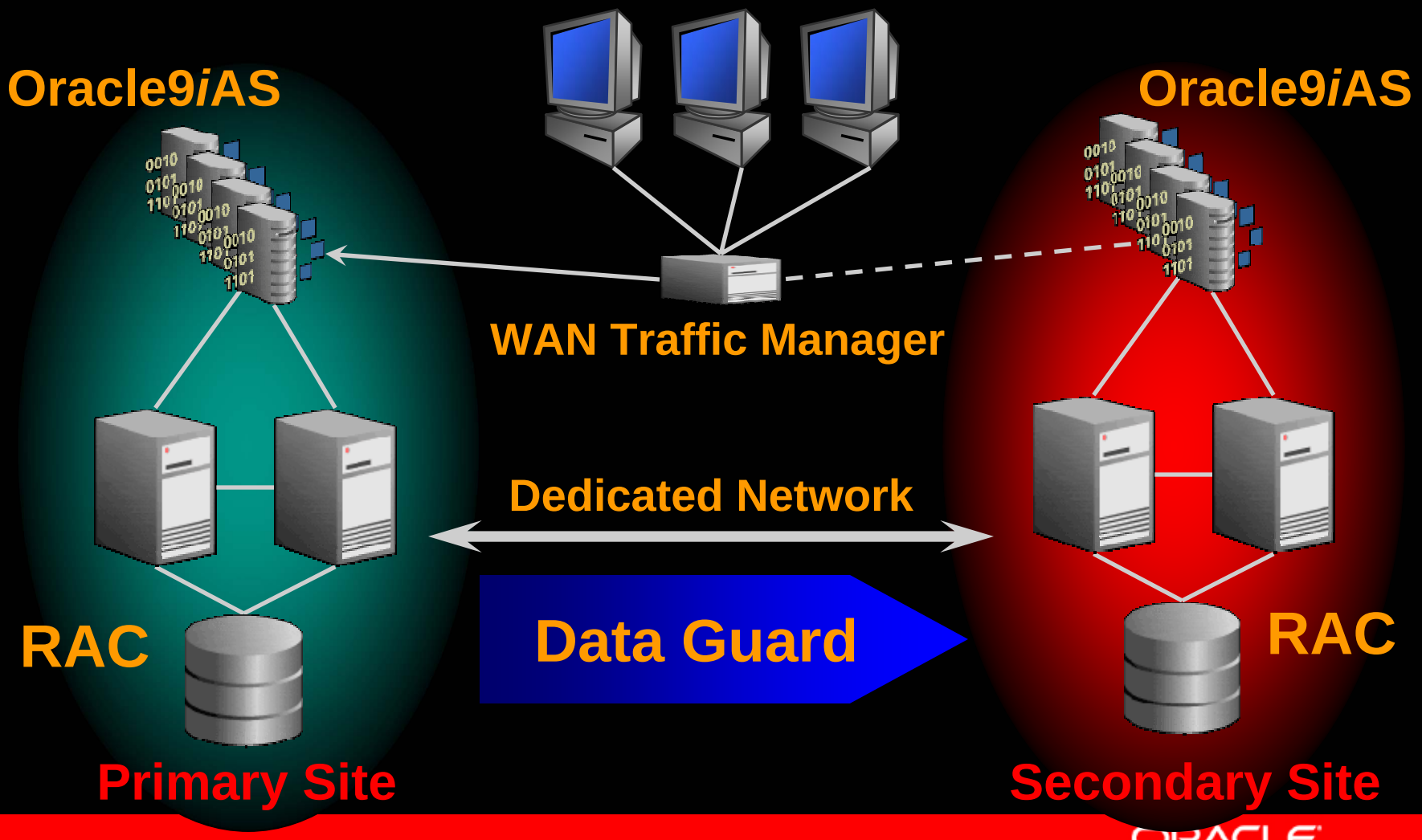


- Operational Practices are key
  - Technology alone is not enough
- MAA is a blueprint for HA & DR
  - Tested, validated, and documented best practices
    - Database, Storage, Cluster, Network, & Middle Tier
    - 20 person year effort
  - otn.oracle.com/deploy/availability

Maximum Availability = HA Architecture + Best Practices

ORACLE

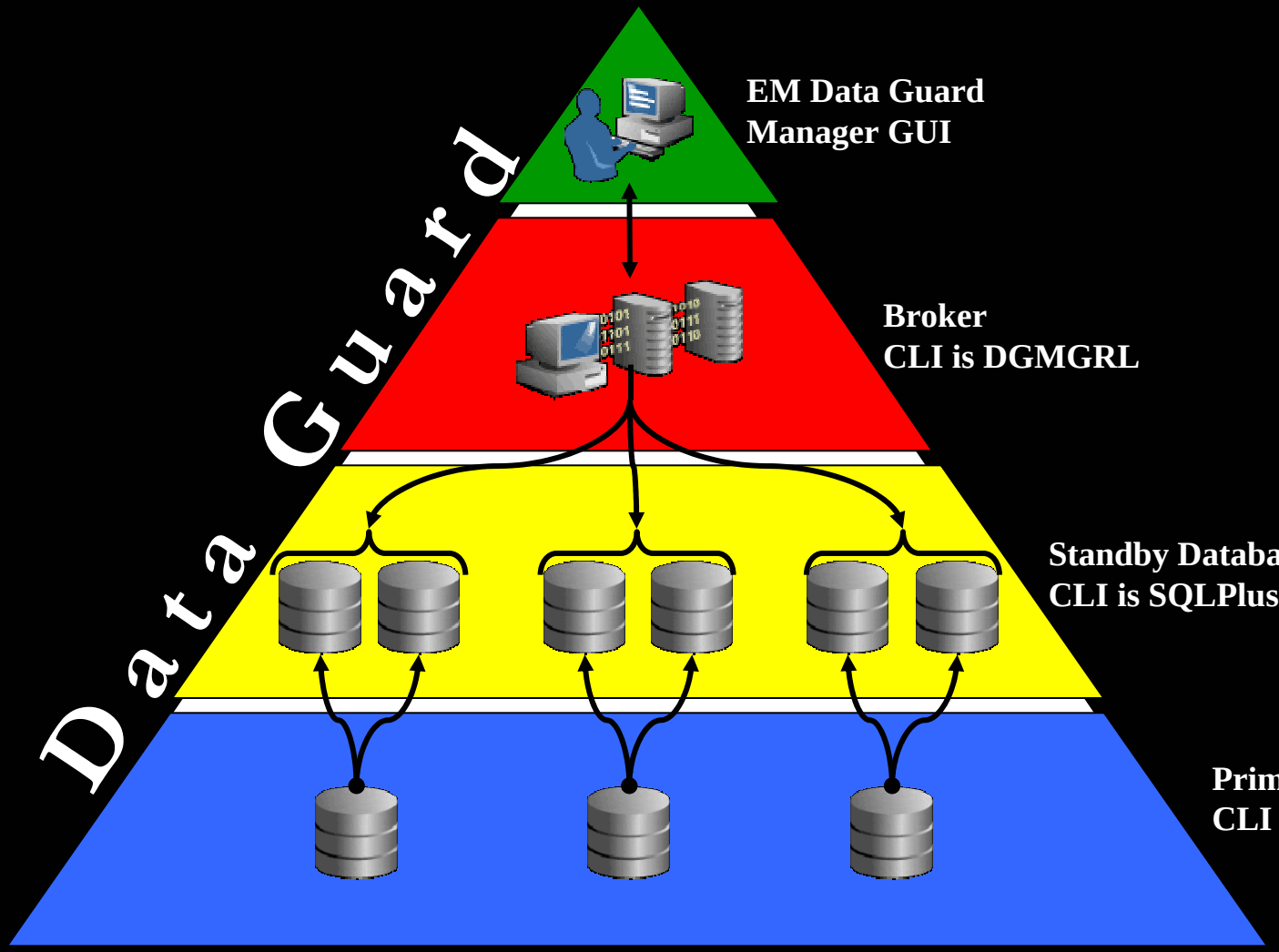
Maximum Availability Architecture



Just what is Data Guard?

- Data Guard helps you protect your Data.
  - Takes your data and automatically puts it elsewhere
  - Makes it available for Failover in case of failure.
- The other capabilities are pure bonus.
  - Switchover for Maintenance
  - Reporting
  - Off-loading Queries
  - Backups

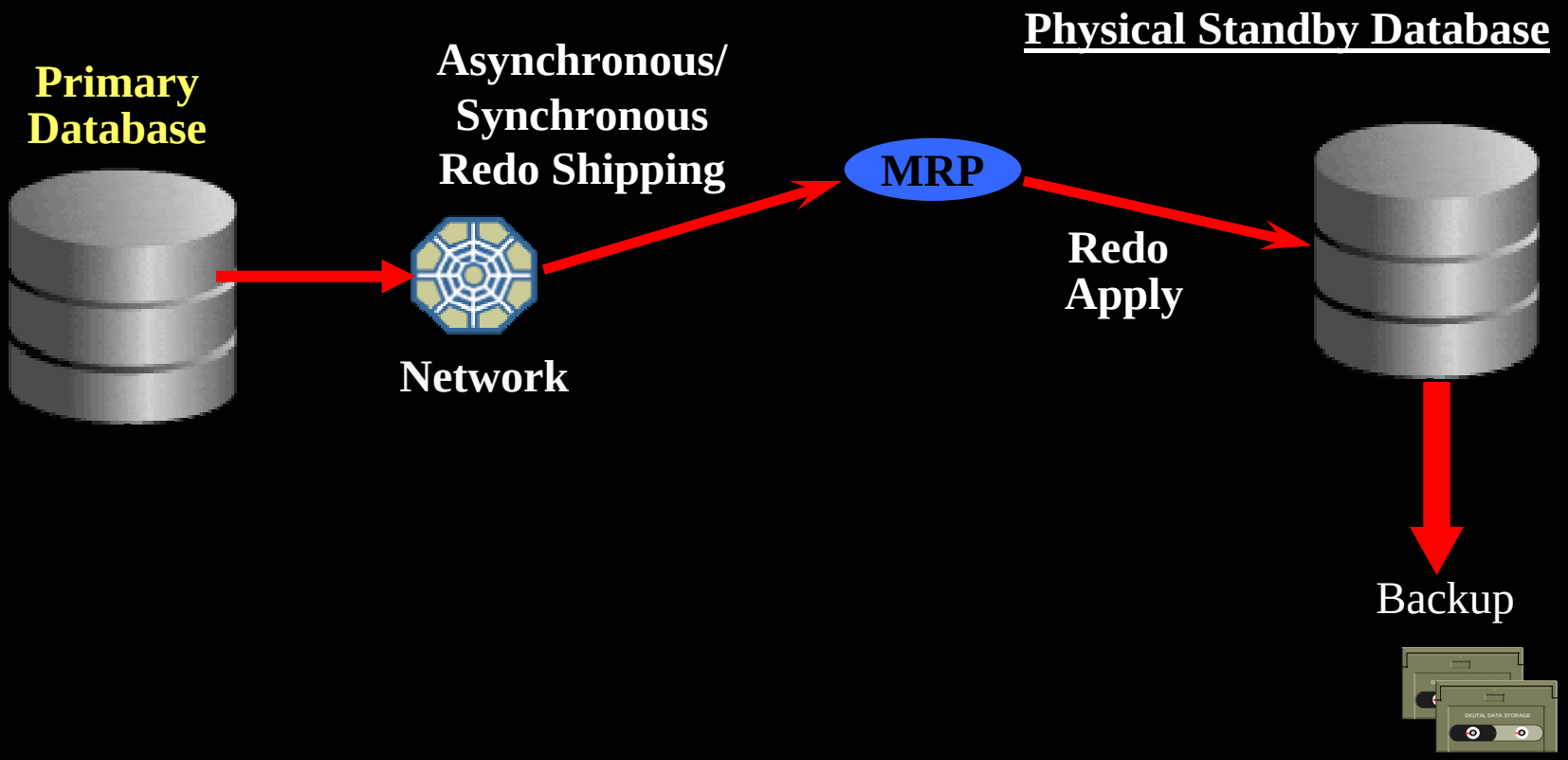
The Data Guard 'Pyramid'



At the Highest Level

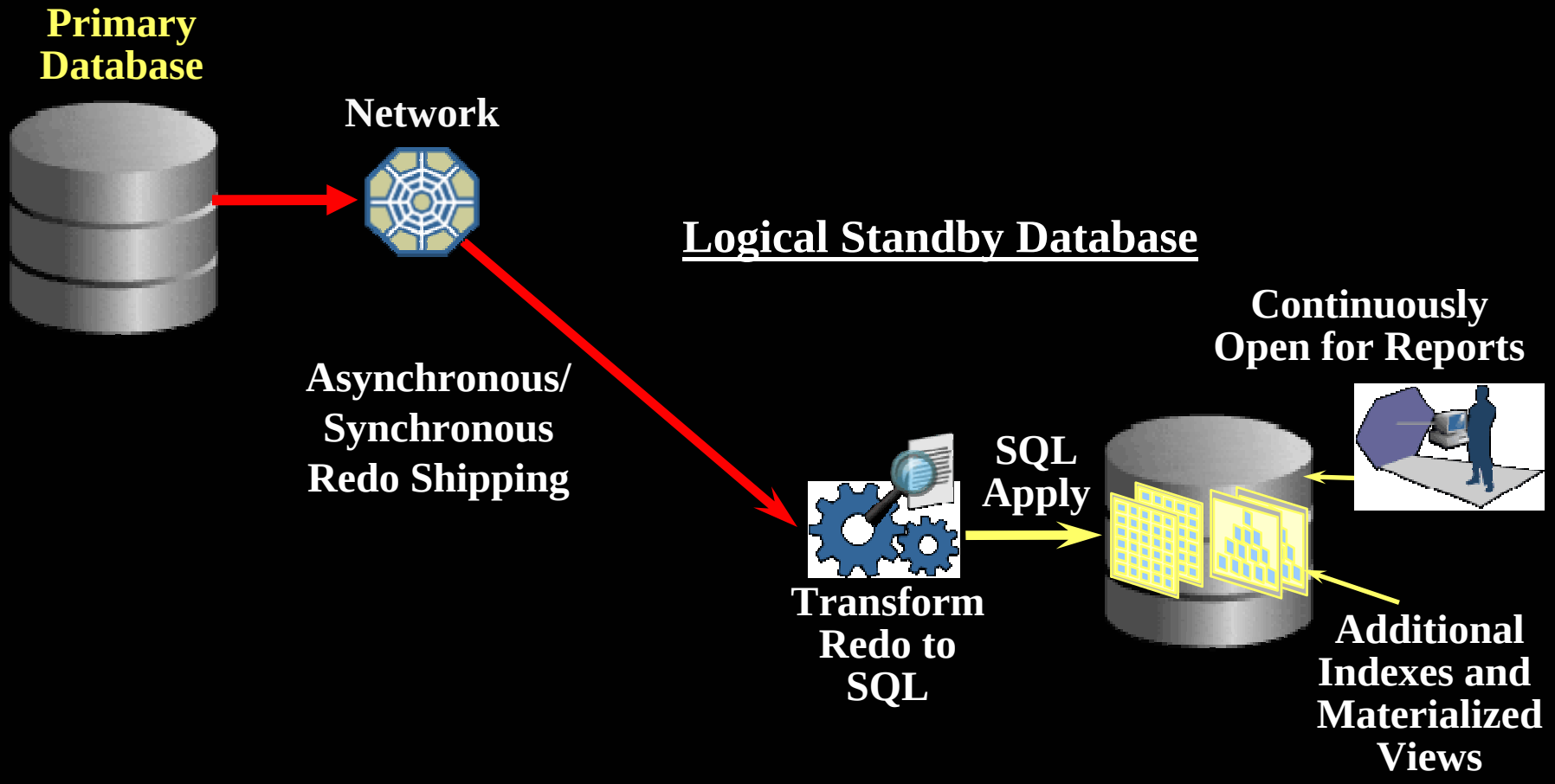
- Data Guard comprises of two parts
  - REDO APPLY
    - Maintains a physical, block for block copy of the Production (also called Primary) database.
  - SQL APPLY
    - Maintains a logical, transaction for transaction copy of the Production database.

REDO Apply Architecture



- Maintains a ‘Physical’ block for block copy of the Primary Database

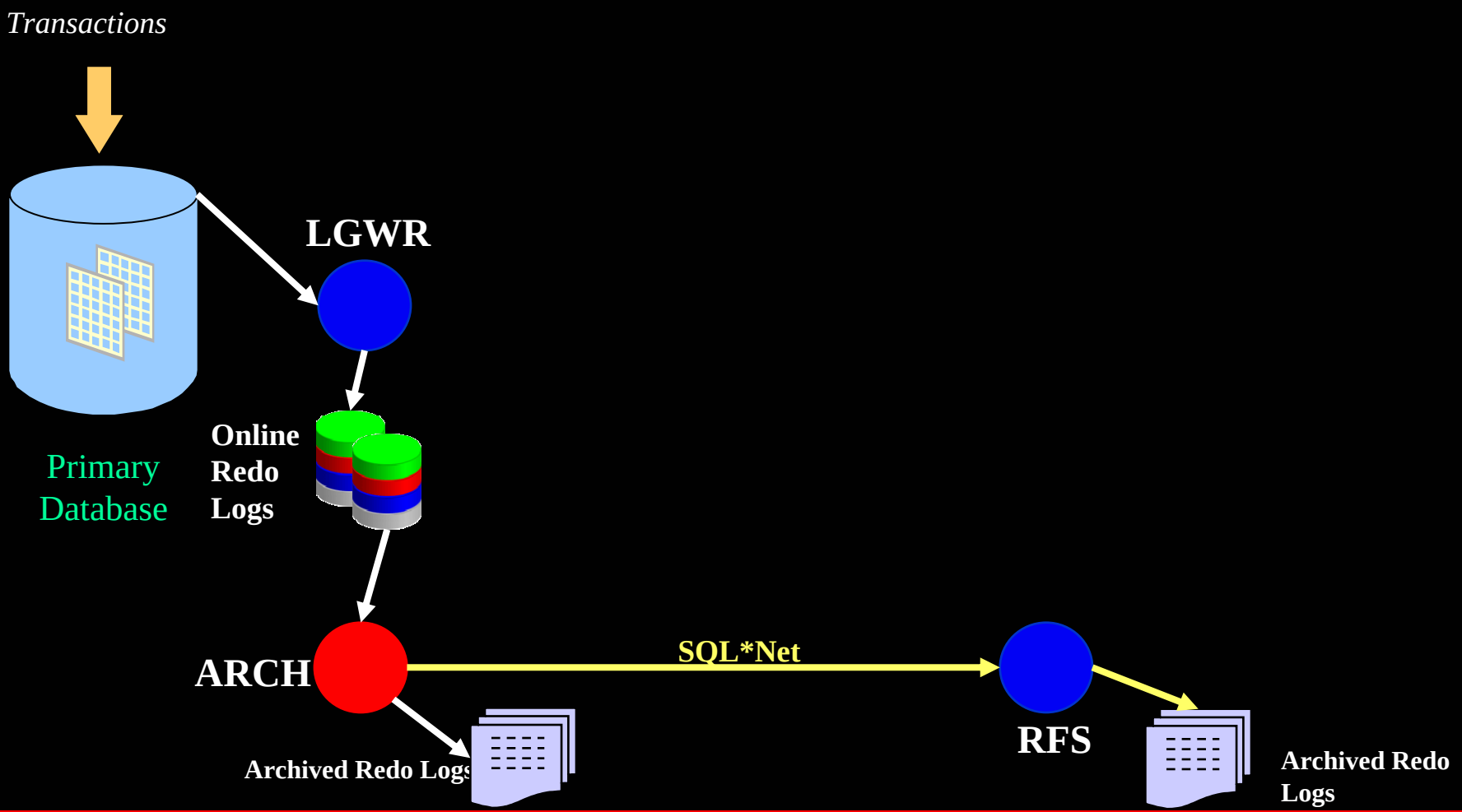
SQL Apply Architecture



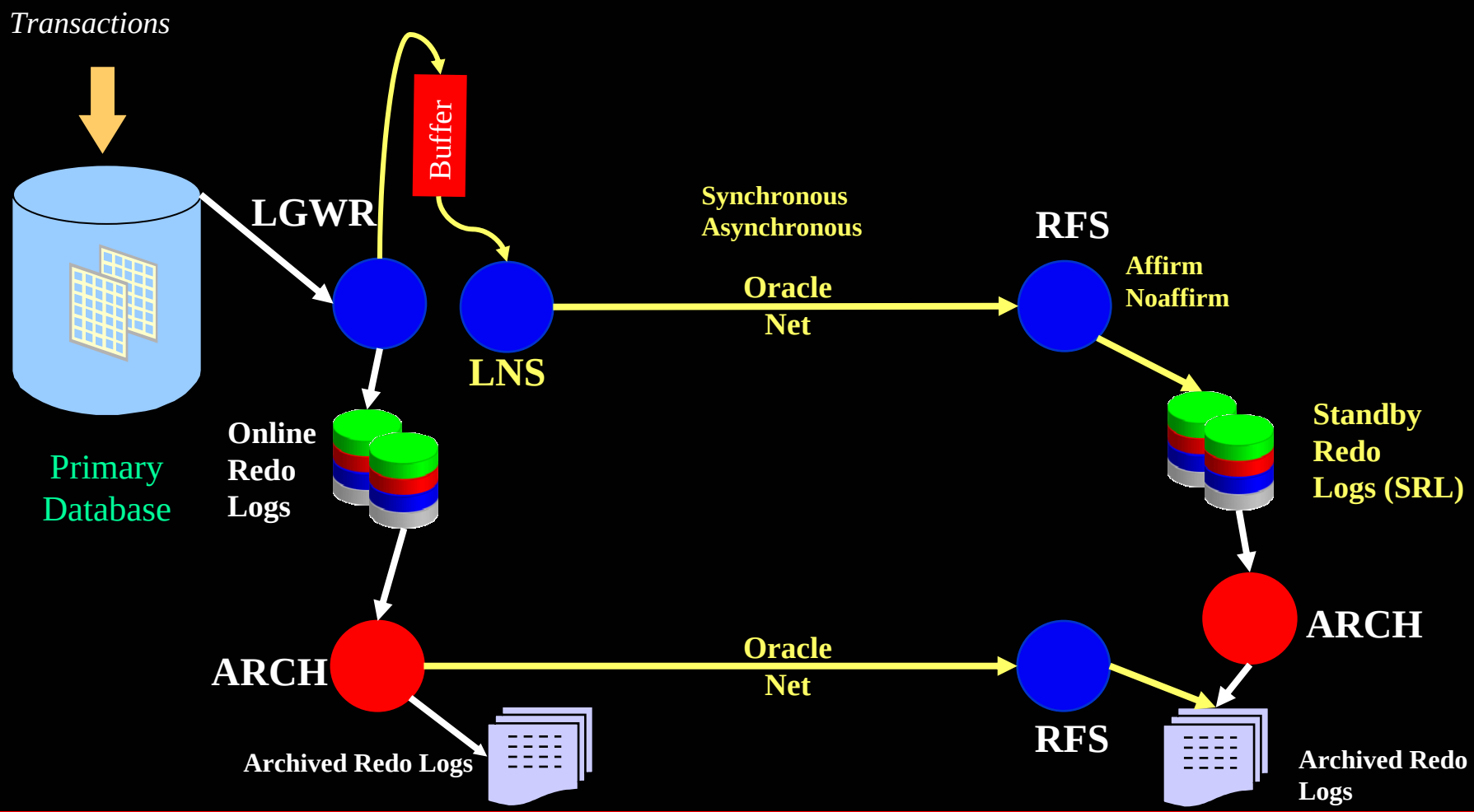
- Maintains a 'Logical' transactional copy of the Primary Database

Redo Transport Services
Getting the data there

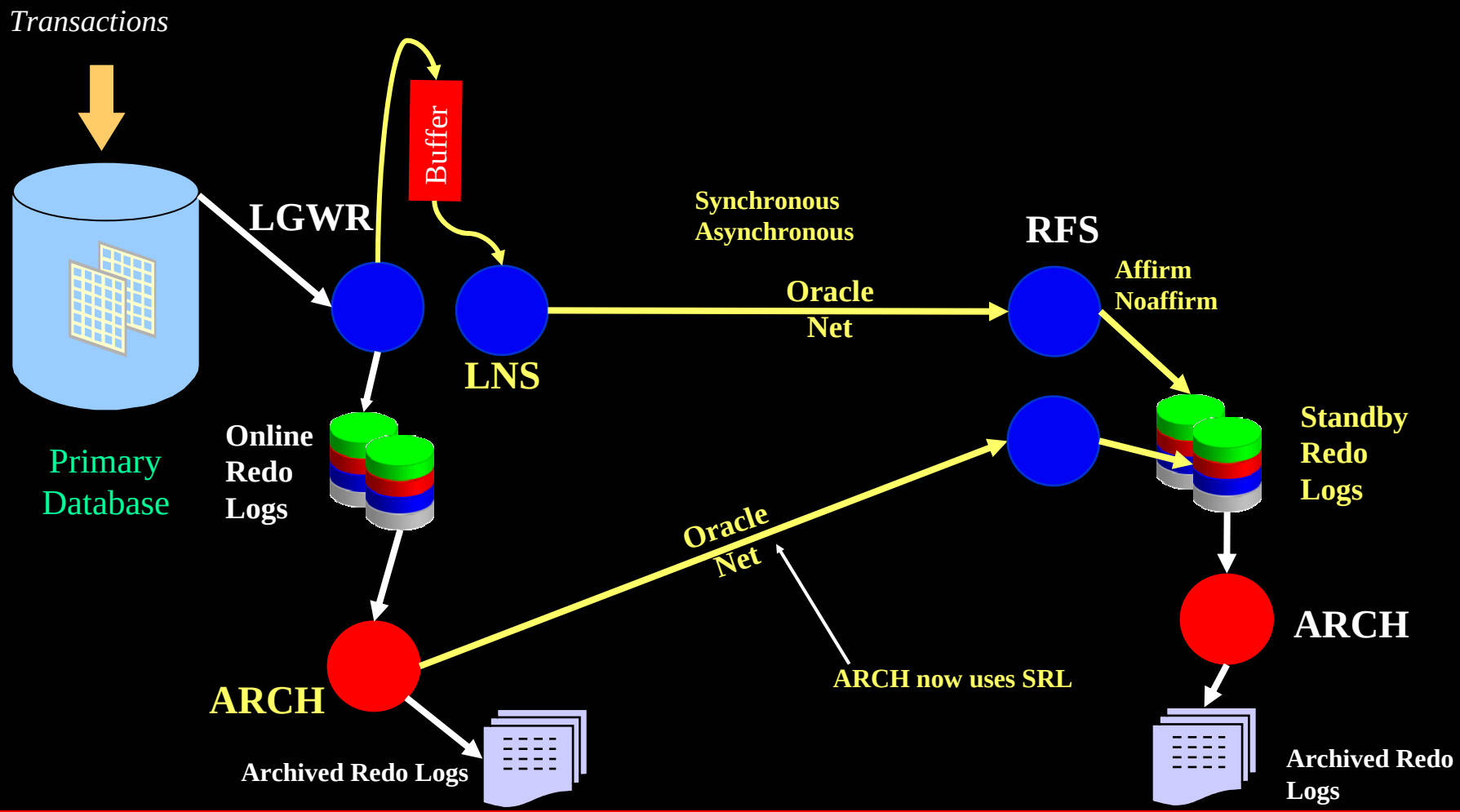
Oracle8*i*


Transactions
LGWR
Online
Redo
Logs
Primary
Database
ARCH
SQL*Net
RFS
Archived Redo Logs
Archived Redo
Logs

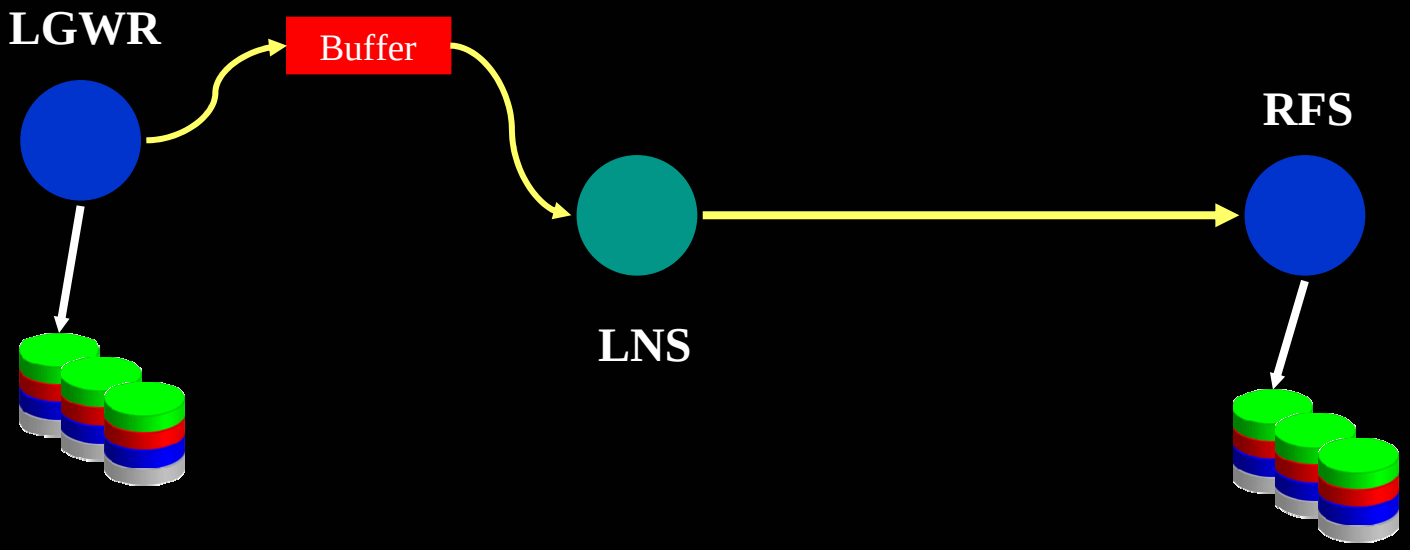
Oracle9*i* Physical Standby



Oracle10*g* - No Difference!

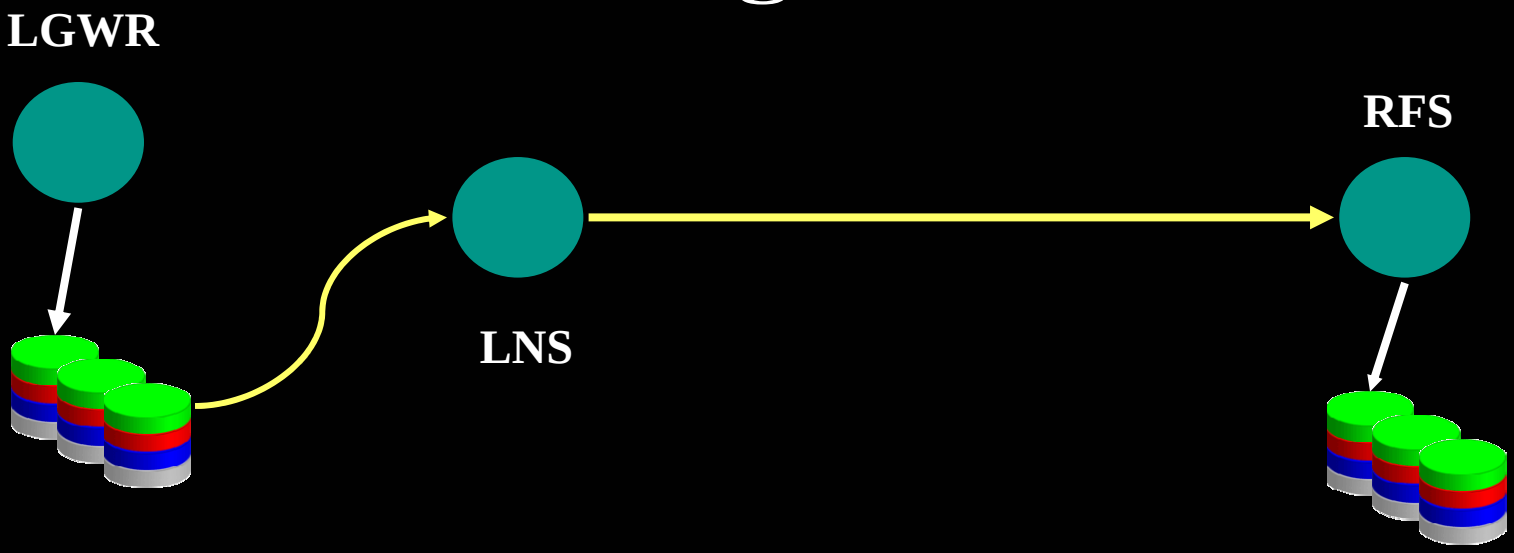


ASYNC Transport
Oracle Database 10*g* Release 1



- In 9*i* the 'ASYNC' buffer (in red above) could be sized from 1mb (2048, the default) up to 10mb (20480)
- In 10*g*R1 the maximum has been raised to 50mb (102400)

ASYNC Transport
Oracle Database 10*g* Release 2



- ASYNC buffer eliminated completely!
- LNS Process chases the Log Writer (LGWR) through the online redo log.
  - 'Buffer' can never fill.
- LGWR process not impeded by LNS issues.

Data Guard Best Practices:
Faster Redo Transport

- Set SDU=32K
- Tune network parameters that affect network buffer sizes and queue lengths
- Ensure sufficient network bandwidth for maximum database redo rate + other activities

Note: Please refer to MAA paper, "Oracle9*i* Data Guard: Primary Site and Network Configuration Best Practices"

http://www.oracle.com/technology/deploy/availability/pdf/MAA_DG_NetBestPrac.pdf

Oracle 10g Release 2 paper coming soon

Data Guard Best Practices:
Tune Network Parameters

- Send and receive buffer size = 3 x bandwidth delay product (BDP)

  BDP = 1,000 Mbps * 25ms (.025 secs)

  = 1,000,000,000 * .025

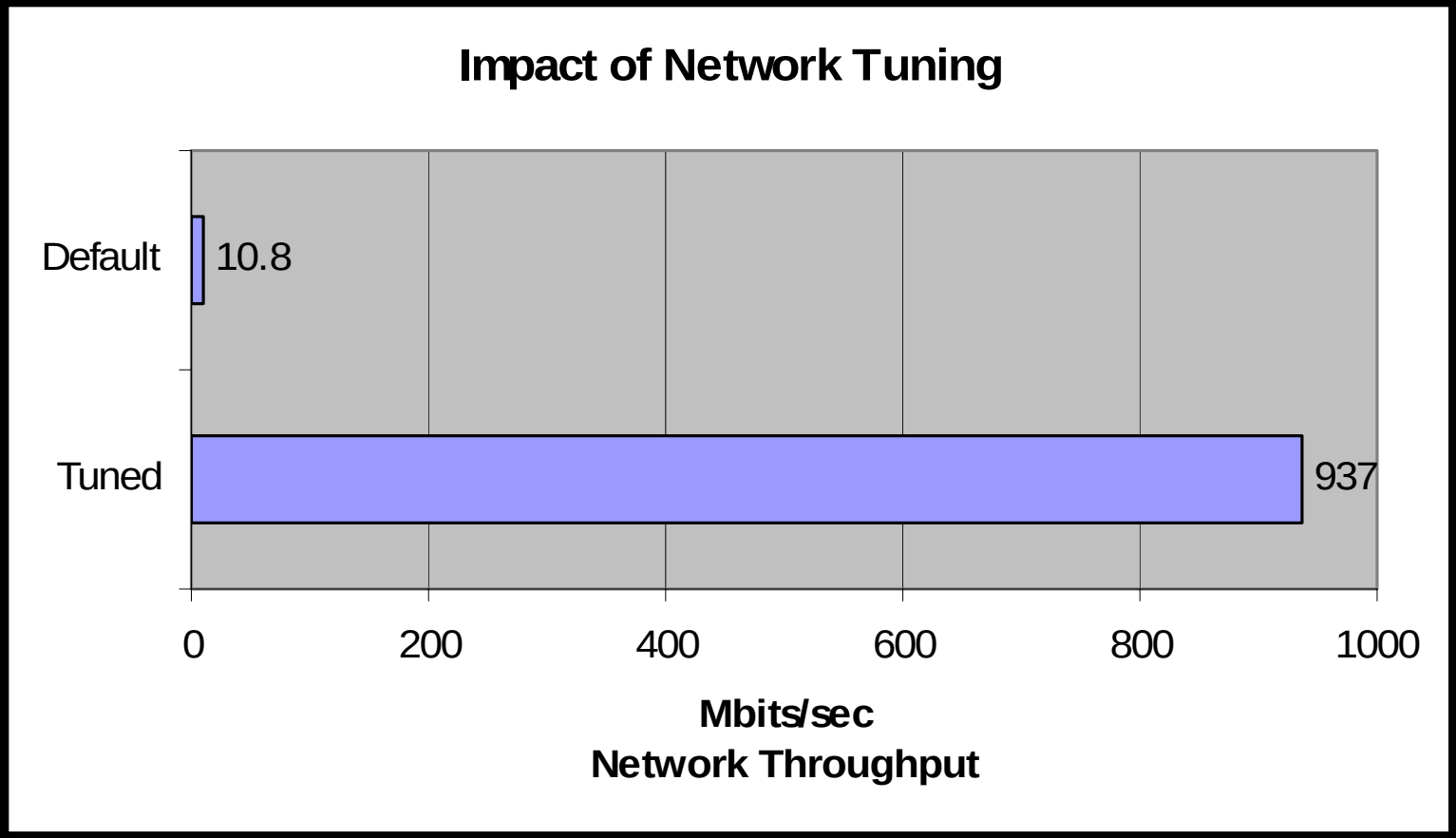
  = 25,000,000 Megabits / 8 = 3,125,000 bytes

- Tune network device queues to eliminate packet losses and waits. Set device queues to a minimum of 10,000 (default 100)

* BDP = the product of the estimated minimum bandwidth and the round trip time between the primary and standby server

ORACLE

Impact of Network Tuning



Oracle MAA Test Result

Setting up a Standby
Enabling Redo Transport

Prepare the Standby System

1. Install Oracle Database Enterprise edition.
2. Setup and start a listener.
3. Add a tnsnames entry to point to the Primary database.
4. Create the necessary directories for the standby.
   - Oradata, Archive, Flash recovery, Admin
5. Best Practice
   - Identical systems and disk setup. (But not required)
   - This example assumes this is true.

ORACLE

Prepare the Primary System

1. Enable archiving on the Primary database.
   -
```
SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP MOUNT;
SQL> ALTER DATABASE ARCHIVELOG;
SQL> ALTER DATABASE OPEN;
```
2. Enable logging on the Primary database.
   -
```
SQL> ALTER DATABASE FORCE LOGGING;
```
3. Create a password file for the Primary database.
4. Add a tnsnames entry to point to the Standby database.

Get the necessary files

1. Make a hot backup of the Primary database
2. Obtain the init parameters
   - `SQL> CREATE PFILE FROM SPFILE;`
3. Create the standby control file.
   - `SQL> ALTER DATABASE CREATE PHYSICAL STANDBY CONTROLFILE AS '<path><filename>';`
4. Copy these files to the standby system.

Prepare the Standby database

1. Add the ‘standby mode’ parameters
   - FAL_SERVER=<Primary Database TNSNAME>
   - FAL_CLIENT=<Standby Database TNSNAME>
   - STANDBY_ARCHIVE_DEST=‘<path>’
     - Always add the trailing slash
   - DB_UNIQUE_NAME=<Standby unique name> 10*g*
   - LOG_ARCHIVE_CONFIG=DG_CONFIG=
     (<primary unique name>,<standby unique name>) 10*g*
   - STANDBY_FILE_MANAGEMENT=AUTO
2. Create a password file for the standby database.
3. Add the standby database to the ‘oratab’ file

Startup the Standby Database

1. Create the spfile
   - `SQL> CREATE SPFILE FROM PFILE;`
2. In Oracle9*i*
   - `SQL> STARTUP NOMOUNT`
   - `SQL> ALTER DATABASE MOUNT STANDBY DATABASE;`
3. In Oracle Database 10*g*
   - `SQL> STARTUP MOUNT`
4. Redo cannot be received until this step

Start Sending Redo!

- On the Primary database setup the parameters.
  - SQL> ALTER SYSTEM SET
    LOG_ARCHIVE_DEST_2='SERVICE=<Standby TNSNAME>
    LGWR ASYNC=20480 NET_TIMEOUT=30 REOPEN=30
    DB_UNIQUE_NAME=<standby unique name>
    VALID_FOR=(ONLINE_LOGFILE,PRIMARY_ROLE)'
    SCOPE=BOTH;
  - Make sure you use NET_TIMEOUT and REOPEN.
- Switch logs!
- If you choose higher than Maximum Performance
  - SQL> ALTER DATABASE SET STANDYBY DATABASE
    TO MAXIMIZE <AVAILABILITY | PROTECTION>;

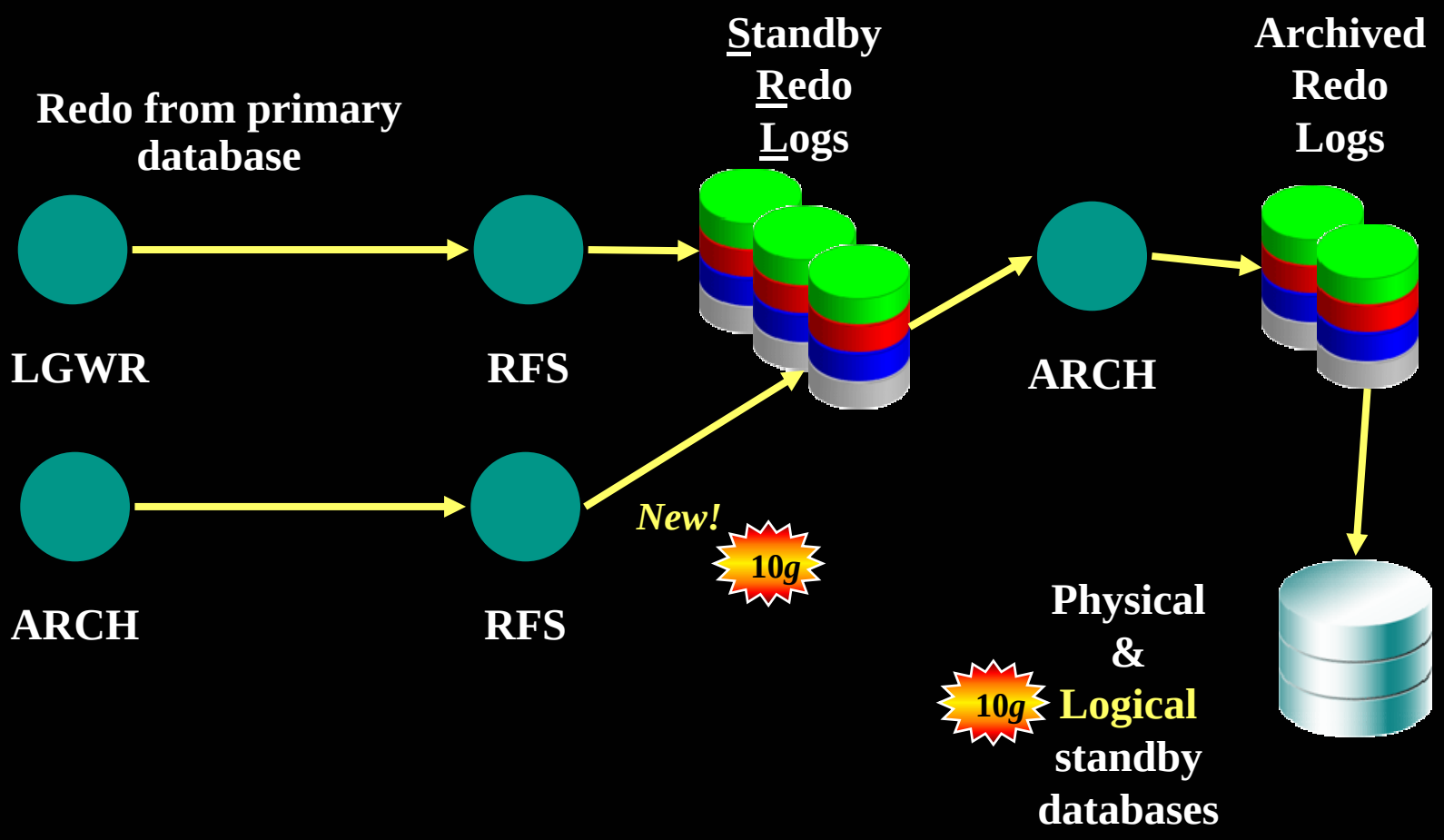
Verify that all is well.

1. Check that the primary is sending redo
   - `SQL> SELECT STATUS,ARCHIVED_SEQ# FROM V$ARCHIVE_DEST_STATUS WHERE DEST_ID=2;`
   - `SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;`
   - `SQL> SELECT STATUS,ARCHIVED_SEQ# FROM V$ARCHIVE_DEST_STATUS WHERE DEST_ID=2;`
2. Status should be ‘VALID’ and the sequence number should increase by 1

Add Standby Redo Logs

- A pool of log file groups on a standby database
  - Used just like the online redo logs on a primary
  - Requires local archiving on the standby database
  - Requires at least same size and number of Primary database online redo logs but more is better
  - Cannot be assigned to a thread in 9*i*
- In Oracle9*i* only standby destinations defined to use the log writer (LGWR) would use the SRL.
  - And only Physical standby databases supported them

SRL Architecture



Benefits

- Better Performance
  - Standby redo logs are pre-allocated files
  - Can reside on raw devices
- Better Protection
  - Can have multiple members
  - If primary database failure occurs, redo data written to standby redo logs can be fully recovered.

Creating Standby Redo Log Files

1. Use the keyword STANDBY on the log file SQL
   - `SQL> ALTER DATABASE ADD STANDBY LOGFILE '<SRL Name>' SIZE 100M;`
2. Log file sizes must match exactly with the Primary online redo log files.
3. Create at least as many SRL as you have Primary Online Redo Logs
4. Starting with 10*g* you can (and should) assign them to threads in a RAC.

Choose Your Protection Mode

Protection Mode	Failure Protection	Redo Shipping
Maximum Protection Zero Data Loss	Protects Against Primary and Network Failure	LGWR SYNC Must have SRL
Maximum Availability Zero Data Loss	Protects Against Primary Failure	LGWR SYNC Must have SRL
Maximum Performance	Best Effort Against Primary Failure	LGWR ASYNC Should have SRL

- Maximum Performance can also use ARCH

What about SQL Apply?

- Creation of Logical Standby databases has evolved during versions Oracle9*i*, Oracle Database 10*g* Release 1 and Oracle Database 10*g* Release 2.
- In Oracle9*i* the only documented way was using a cold backup of the Primary database.
- Starting with Oracle Database 10*g* Release 1 the process starts with a Physical standby database.
- A method was devised and published on Metalink to reduce the amount of downtime for Oracle9*i*.

Preparing for SQL Apply

- Several restrictions with Data types, Table types and functionality.
- Restrictions greatly reduced with Oracle Database 10*g* Release 1 and further reduced with Release 2
- Verify, in all releases, that your Primary database can support a Logical standby database.
  - `DBA_LOGSTDBY_UNSUPPORTED`
  - `DBA_LOGSTDBY_NOT_UNIQUE`
- Turn on supplemental logging
  - Automatic in Oracle Database 10g Release 2

ORACLE

Oracle9*i* SQL Apply

- Use the cold backup method from Chapter 4 of the documentation.
- Or refer to Metalink note 278371.1
  - Creating a Logical Standby with Minimal Production Downtime

Oracle Database 10*g* Release 1 SQL Apply

- Create a Physical Standby database.
- Replace the Physical standby control file with a Logical standby control file.
  - `SQL> ALTER DATABASE CREATE LOGICAL STANDBY CONTROLFILE AS '<path><filename>';`
- Restart physical standby apply.
  - When it stops, shut down the Physical standby
  - Run NID, fix the parameters and create a new password file.
- Open the standby resetlogs and start SQL Apply

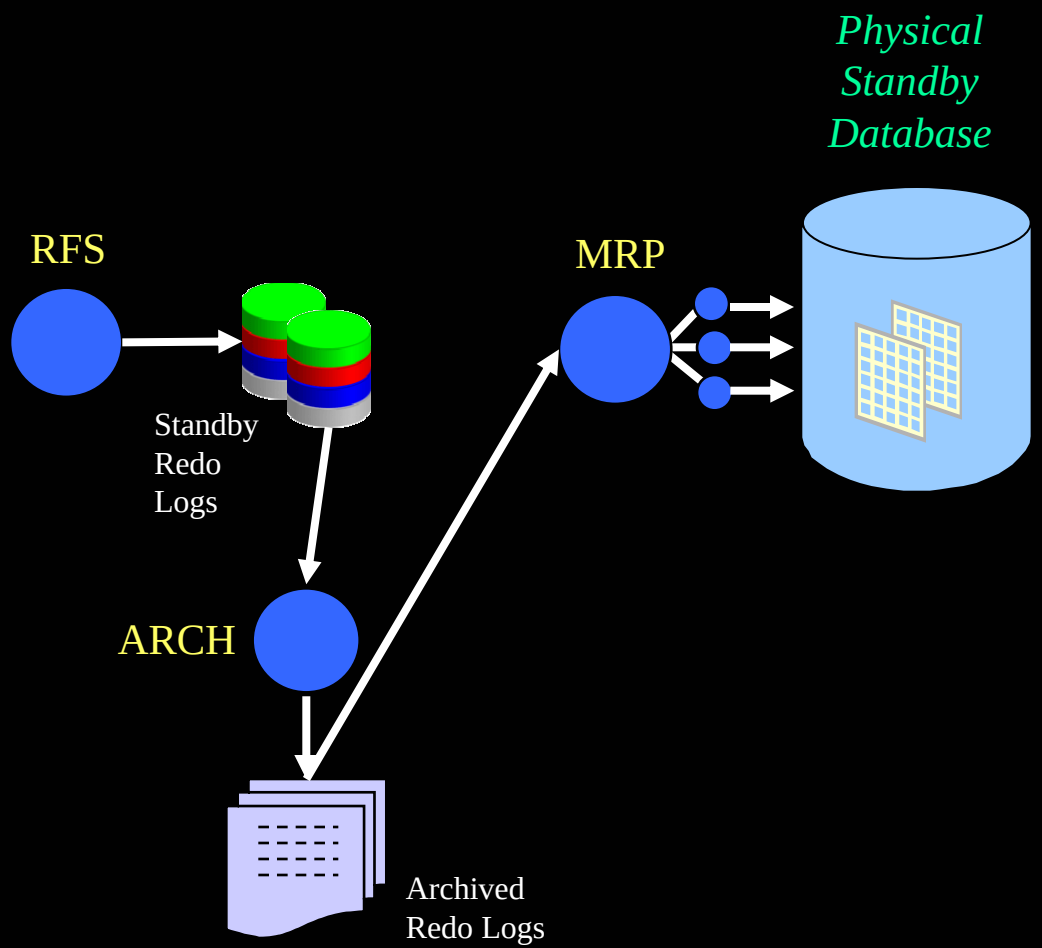
ORACLE

Oracle Database 10*g* Release 2 SQL Apply

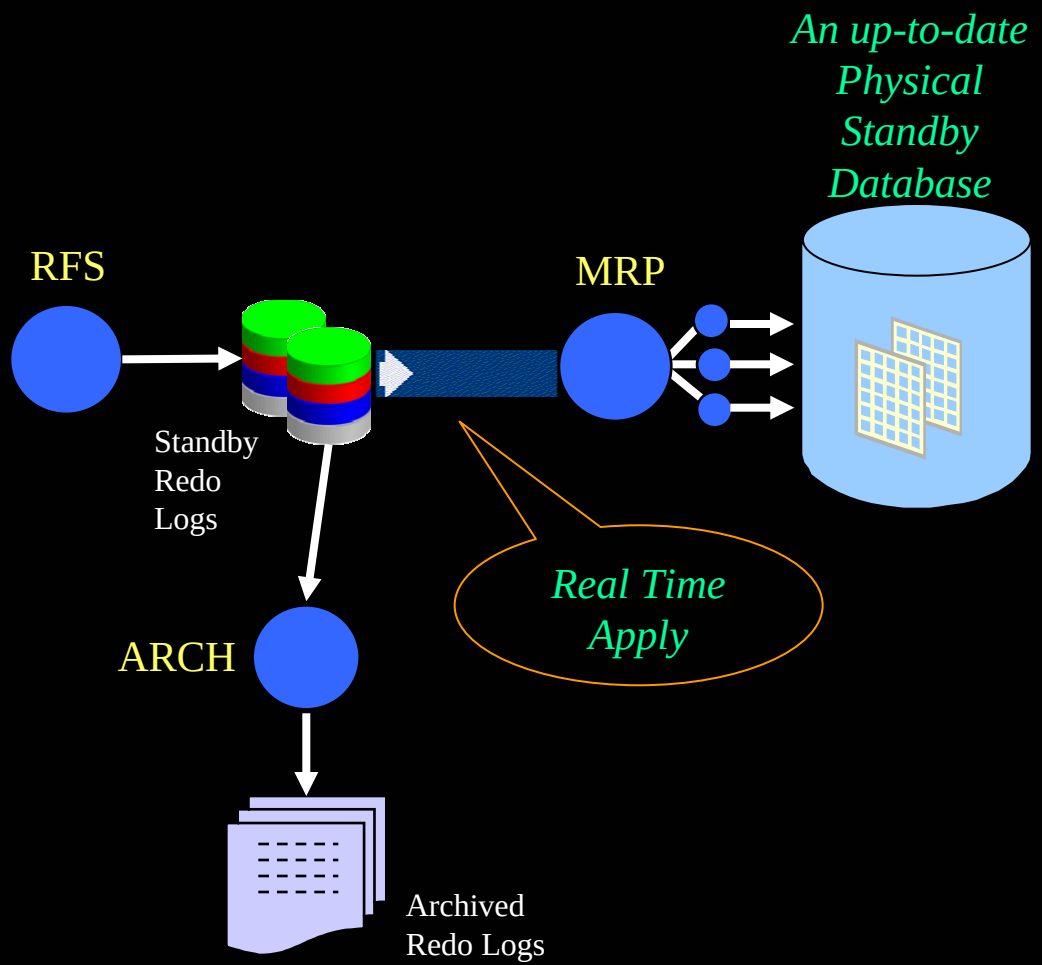
- Create a Physical Standby database.
  - When synchronized, stop Redo Apply
- Execute the dictionary build on the Primary
  - `SQL> EXECUTE DBMS_LOGSTDBY.BUILD;`
- Restart the Apply on the Physical Standby.
  - `SQL> ALTER DATABASE RECOVER TO LOGICAL STANDBY <new dbname>;`
  - Create a new password file.
- Open the standby resetlogs and start SQL Apply

Apply Services
Getting the data into the standby

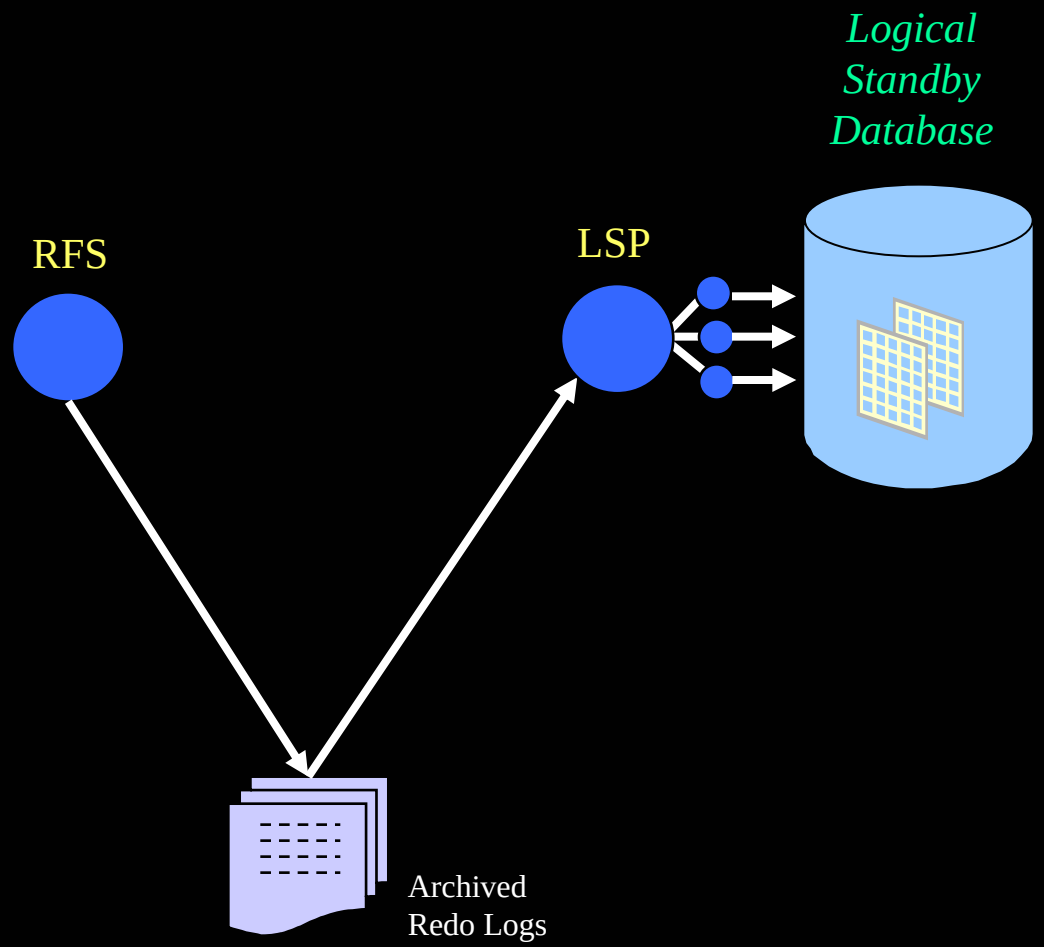
Redo Apply Architecture



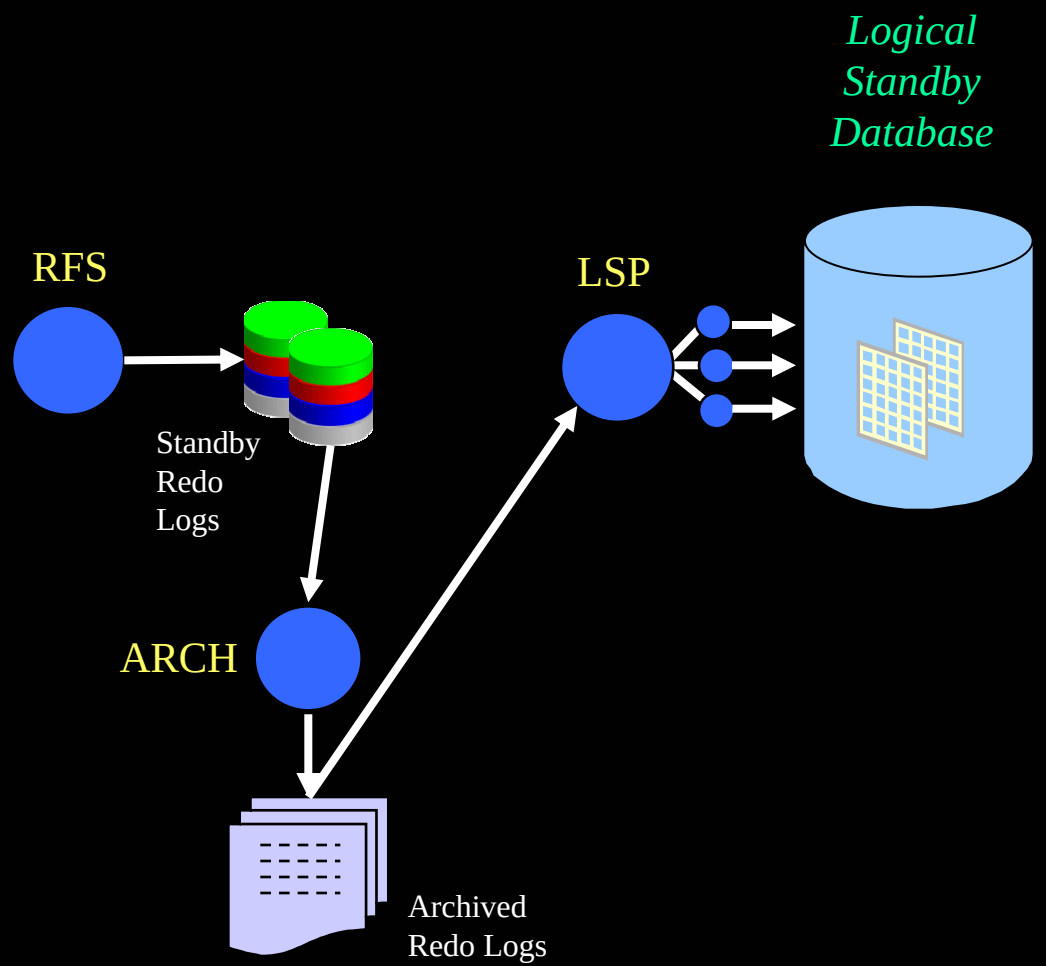
Redo Apply Architecture (RTA)



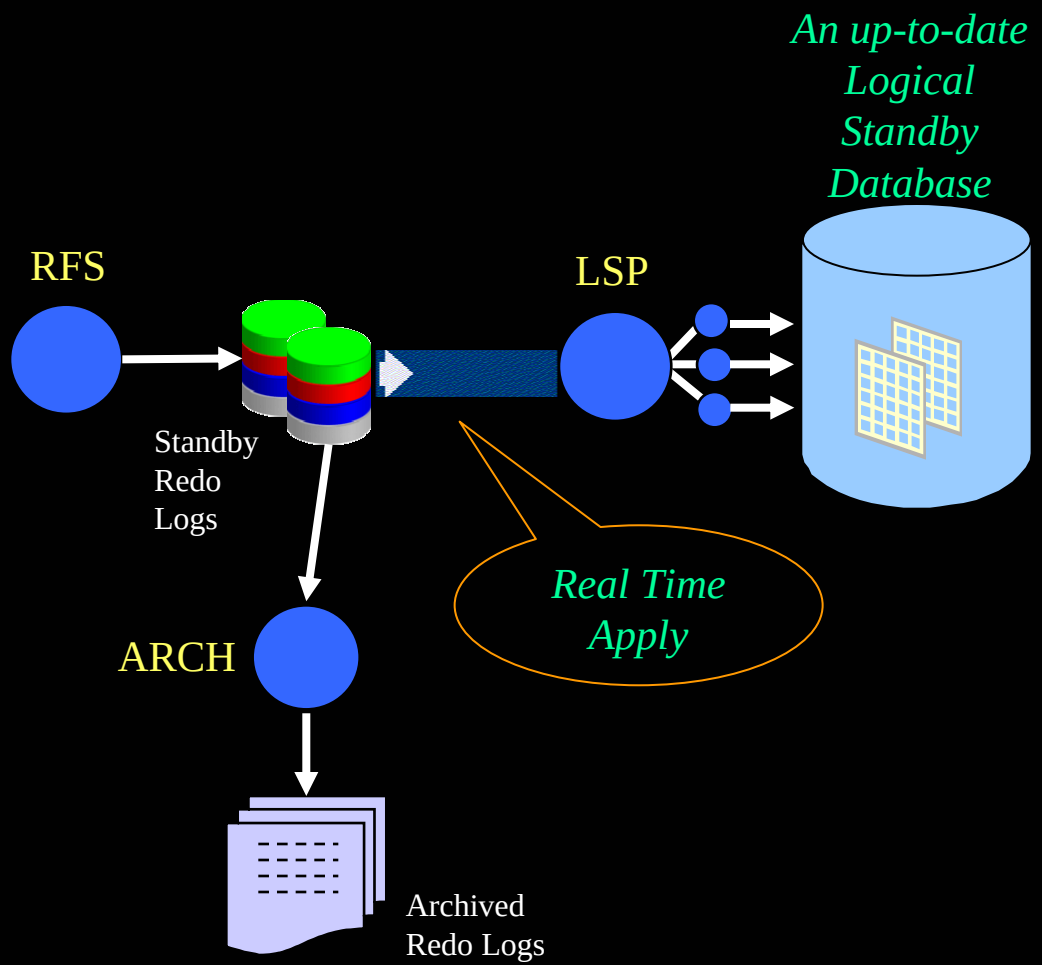
Oracle9*i* SQL Apply Architecture



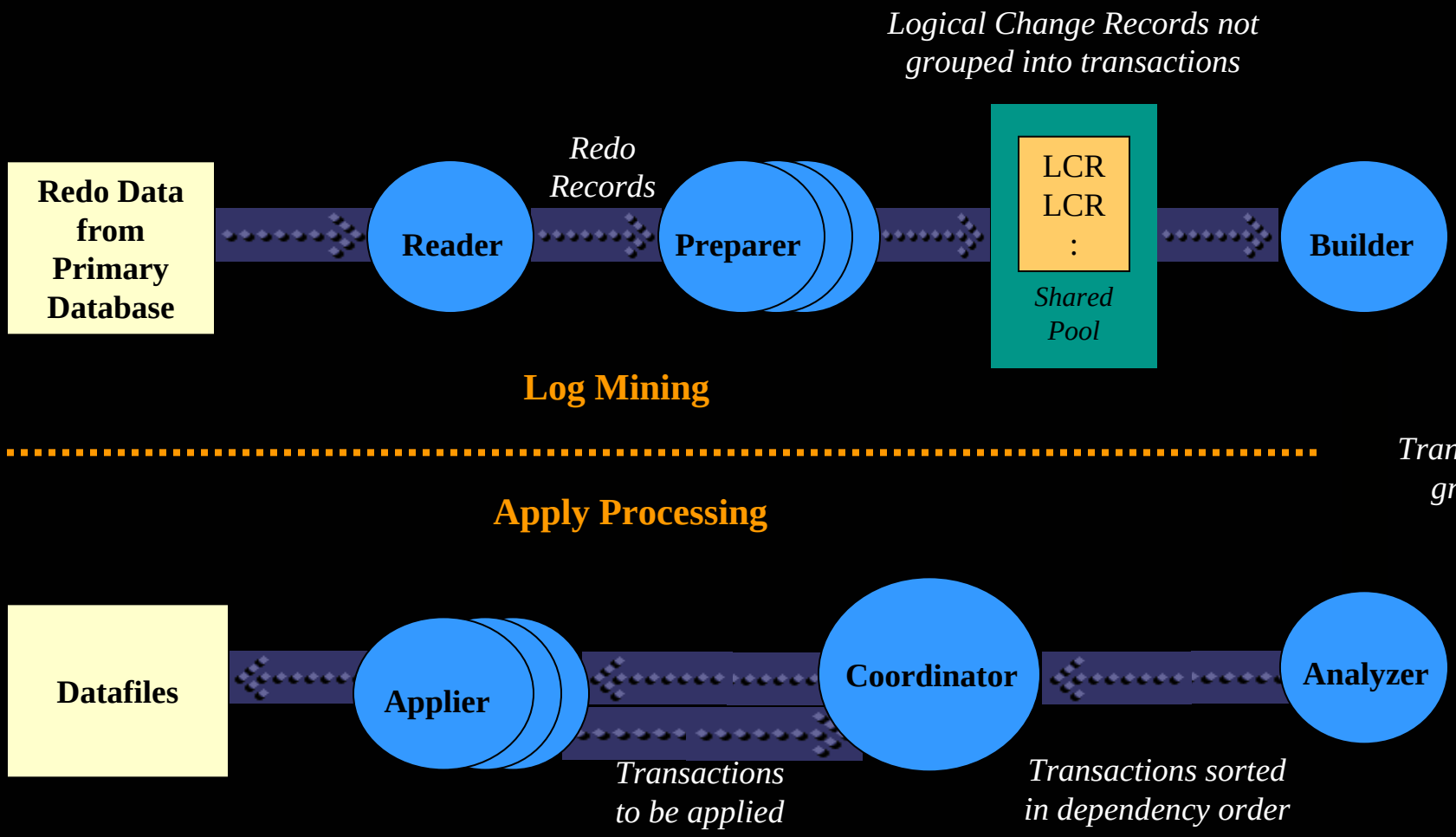
SQL Apply Architecture (10*g*)



SQL Apply Architecture (RTA)



SQL Apply Process Architecture



Starting up Apply Services

Redo Apply

- Starting apply
  - `SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE DISCONNECT;`
- Stopping apply
  - `SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;`

Replace the Temporary file

- You need to replace the temporary datafile.
  - `SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;`
  - `SQL> ALTER DATABASE OPEN READ ONLY;`
  - `SQL> ALTER TABLESPACE temp ADD TEMPFILE '/oradata/temp01.dbf' SIZE 40M REUSE;`
  - `SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT;`
  - I waited until this point to mention this because you must start redo apply before you can do this.
- Not necessary in Oracle Database 10*g* Release 2

SQL Apply

- Starting apply
  - `SQL> ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE;`
  - `SQL> ALTER DATABASE START LOGICAL STANDBY APPLY INITIAL;`
- Stopping apply
  - `SQL> ALTER DATABASE STOP LOGICAL STANDBY APPLY;`

Role Transition
Trading Places

Overview

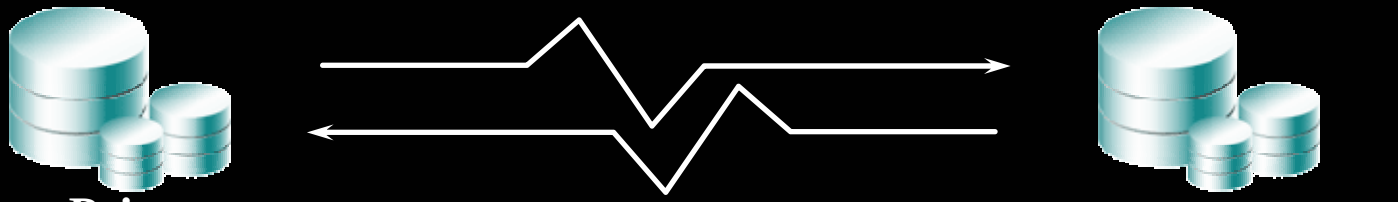
- There are two ways to change roles.
  - Switchover
    - Changing roles with someone else and letting them take over while you become a standby
    - Switchover should be done regularly to ensure everything works.
  - Failover
    - Assigning someone else to take over when the original boss goes away
    - You hope you never have to do a Failover but believe me, you will..

Prepare for Switchover

- On the Standby database setup Redo parameter.
  - SQL> ALTER SYSTEM SET
    LOG_ARCHIVE_DEST_2='SERVICE=<Primary TNSNAME>
    LGWR ASYNC=20480 NET_TIMEOUT=30 REOPEN=30
    DB_UNIQUE_NAME=<Primary unique name>
    VALID_FOR=(ONLINE_LOGFILE,PRIMARY_ROLE)
    SCOPE=BOTH;

    10*g* 10*g*
  - If you are using Oracle9*i* you will need to 'DEFER' this destination manually when a database is in Standby mode.
- Add the 'standby role' parameters to the Primary.
- Add Standby Redo Logs to the Primary database
  - If you haven't already!

Switchover to a Physical Standby



1
```
ALTER DATABASE COMMIT TO
SWITCHOVER TO STANDBY
WITH SESSION SHUTDOWN;
```

2
```
ALTER DATABASE COMMIT
TO SWITCHOVER TO
PRIMARY;
```

3
```
RESTART DATABASE
```

4
```
RESTART DATABASE
```
(Only ALTER DATABASE OPEN necessary in Oracle Database 10*g* Release 2 if standby was never opened Read Only)

10*g*

5
```
ALTER DATABASE RECOVER
MANAGED STANDBY DATABASE
USING CURRENT LOGFILE
DISCONNECT;
```

Failover to a Physical Standby

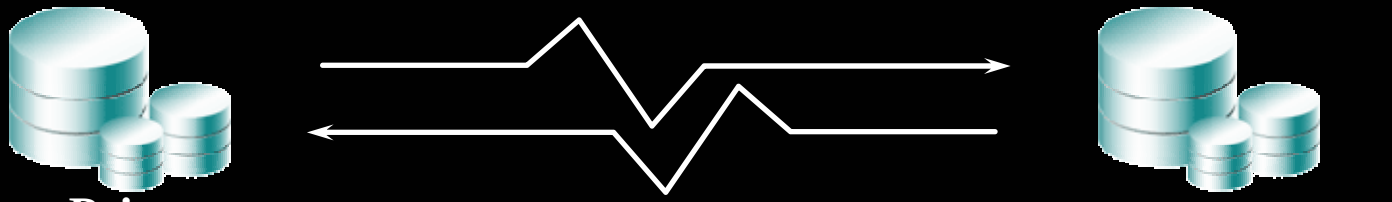
Physical Standby Database

1
```
ALTER DATABASE RECOVER
MANAGED STANDBY
DATABASE FINISH;
```

2
```
ALTER DATABASE COMMIT
TO SWITCHOVER TO
PRIMARY;
```

3
```
RESTART DATABASE
```

Switchover to a Logical Standby



10*g*

1 `ALTER DATABASE PREPARE TO SWITCHOVER TO STANDBY;`

2 `ALTER DATABASE PREPARE TO SWITCHOVER TO PRIMARY;`

10*g*

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

4 `ALTER DATABASE COMMIT TO SWITCHOVER;`

5 `ALTER DATABASE START LOGICAL STANDBY APPLY;`

Failover to a Logical Standby

Logical Standby Database

1
```
ALTER DATABASE STOP
LOGICAL STANDBY APPLY;
```

2
```
ALTER DATABASE ACTIVATE
LOGICAL STANDBY
DATABASE;
```

For more information on Oracle High Availability, Disaster Protection, Backup & Recovery, and Storage Management technology go to:
http://otn.oracle.com/deploy/availability/

Monthly
4th Thursday
6pm – 8pm

IBM Center
Rocky Point

WWW.SOUG.NET