

ORACLE®

# Oracle OpenWorld '14

## Oracle Active Data Guard and Oracle GoldenGate High-Availability Best Practices

BG Garin, Stephan Haisley  
Enterprise Replication Server Technologies  
Oracle Corporation



## Safe Harbor Statement

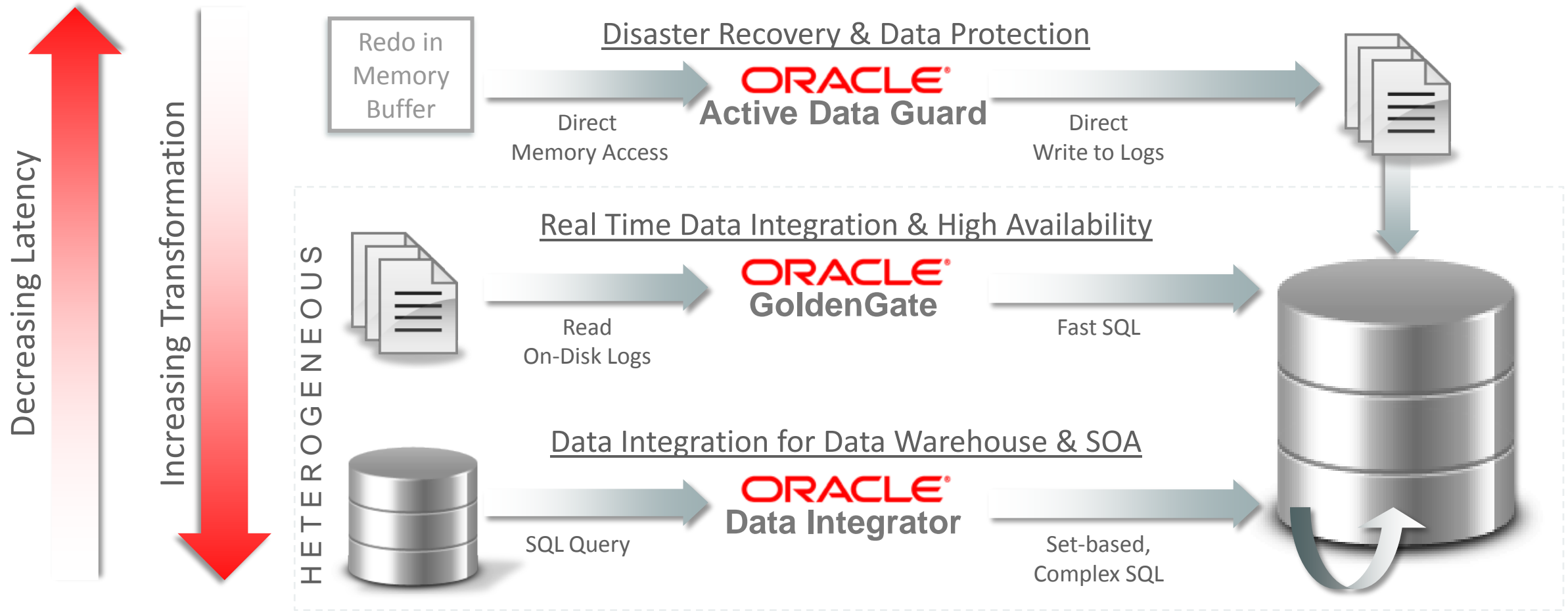
The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development,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

# Program Agenda

- 1 Overview of Oracle GoldenGate
- 2 Overview of Integrated Extract / Integrated Replicat
- 3 Overview of Oracle Active Data Guard /FSFO
- 4 Technical Challenges with Integration
- 5 Oracle's Bundled Agent(XAG) extends HA for GoldenGate
- 6 Case Study with GoldenGate and Data Guard
- 7 Summary

# Oracle Has You Covered

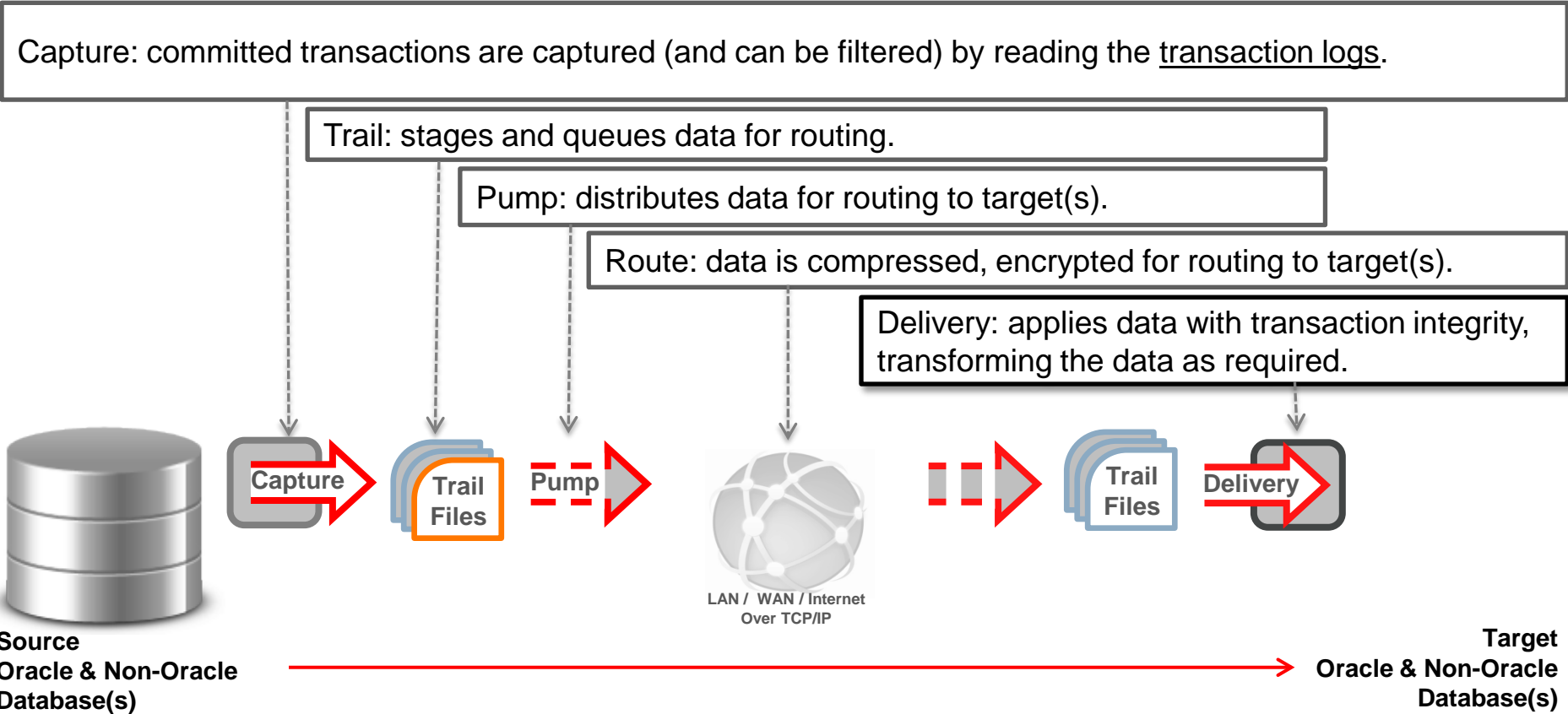
## Oracle Data Integration Solutions & Maximum Available Architecture (MAA)



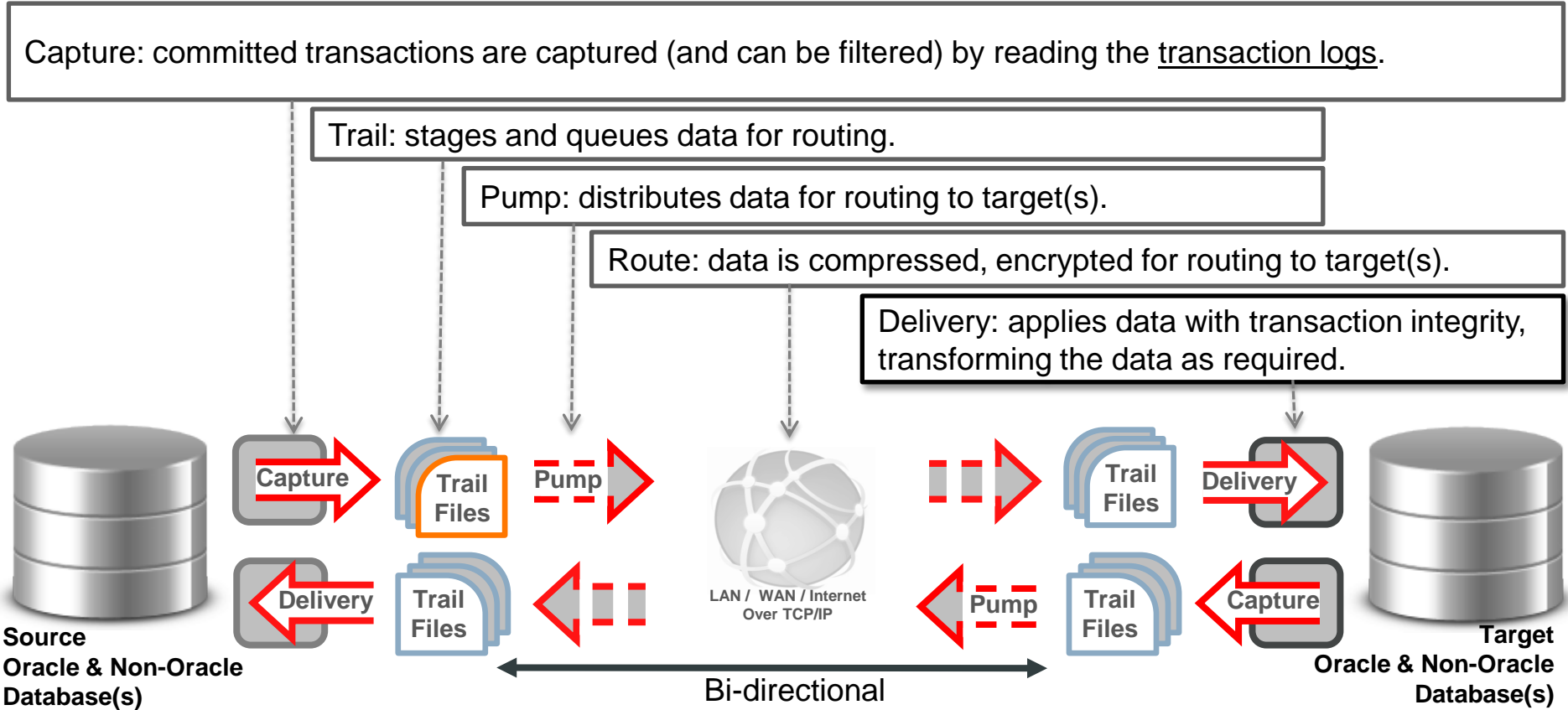
# Program Agenda

- 1 Overview of Oracle GoldenGate
- 2 Overview of Integrated Extract / Integrated Replicat
- 3 Overview of Oracle Data Guard
- 4 Technical Challenges with Integration
- 5 Oracle's Bundled Agent(XAG) Extends HA for GoldenGate
- 6 Case Study with GoldenGate and Data Guard
- 7 Summary

# How Oracle GoldenGate Works



# How Oracle GoldenGate Works





# Program Agenda

- 1 Overview of Oracle GoldenGate
- 2 Overview of Integrated Extract/ Integrated Replicat
- 3 Overview of Oracle Data Guard
- 4 Technical Challenges with Integration
- 5 Oracle's Bundled Agent(XAG) Extends HA for GoldenGate
- 6 Case Study with GoldenGate and Data Guard
- 7 Summary

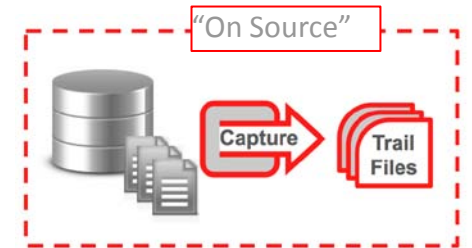
# Integrated Extract: Overview

- Introduced in GoldenGate 11.2.1
- Integrated Extract for Oracle Source databases only
  - Database releases: 11.2.0.3 and beyond
- As of GoldenGate 12.1.2.0.0, it supports use of the Integrated Dictionary for DDL support (no more trigger)
  - Database release: 11.2.0.4 with 11.2.0.4 at the source system



# Integrated Extract: Overview

- Supports multiple deployment configuration
  - On-Source : Source database and Integrated Extract are on the same machine
  - Downstream : Integrated Extract runs on different database – typically on different machines
- Easy transitions for existing GoldenGate customers
  - Customers may choose which option they prefer based on their requirements



# Integrated Replicat

- Introduced in GoldenGate 12.1.2.0.0
- Integrated Replicat for Oracle target databases only
  - Database releases: 12c and 11.2.0.4
- Leverages database parallel apply servers via inbound server for automatic dependency aware parallel apply
- Minimal changes to Replicat configuration
- Single replicat, no need to use @RANGE or THREAD or other manual partitioning



# Integrated Replicat

## Key Features

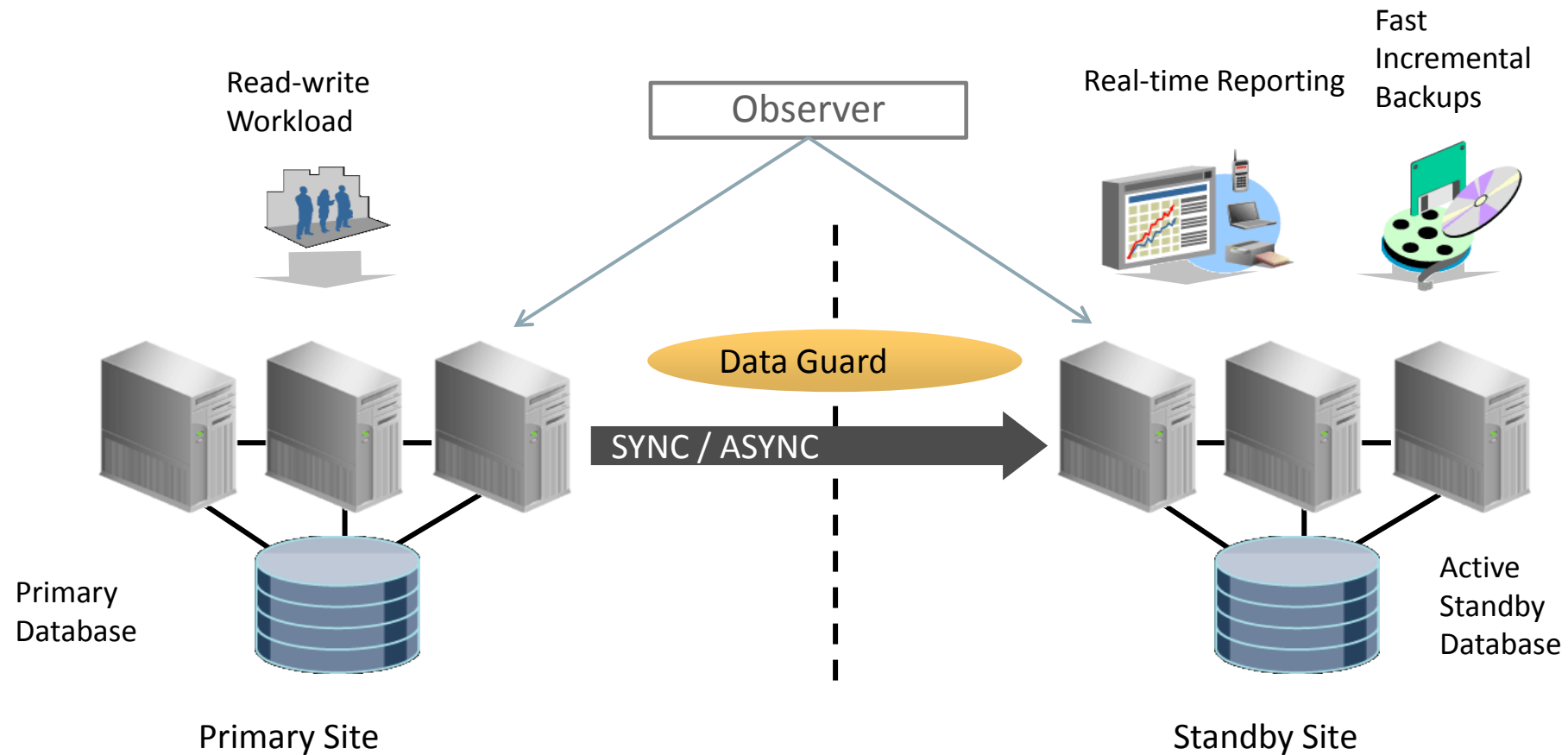
- Streaming Protocol with asynchronous processing
- Applies source transactions in parallel based on dependencies between transactions
- Automatic tuning of parallelism based on workload
- Conflict detection and resolution
  - GoldenGate CDR, REPERROR, and HANDLECOLLISIONS
  - DML, DDL and error handlers
  - Unhandled errors redirected to Replicat for retry and failure processing
- Session redo tags provide fine-grained filtering, such as cycle prevention in bidirectional replication

# Program Agenda

- 1 Overview of Oracle GoldenGate
- 2 Overview of Integrated Extract / Integrated Replicat
- 3 Overview of Oracle Active Data Guard
- 4 Technical Challenges with Integration
- 5 Oracle's Bundled Agent(XAG) Extends HA for GoldenGate
- 6 Case Study with GoldenGate and Data Guard
- 7 Summary

# Oracle Active Data Guard

## Disaster Recovery and Read-Only Offload to an Active Standby



# Oracle Data Guard Concepts

- **Switchover:** Planned role transition from a primary database to one of its standby database.
  - DGMGRL > SWITCHOVER TO CHICAGO
    - Requires connectivity to both primary and standby database
    - Ensures all redo has shipped
    - Orderly transition (both databases agree on every thing)
- **Failover:**
  - DGMGRL > FAILOVER TO CHICAGO
    - No connectivity with primary
    - New primary becomes the source of truth



# Data Guard FSFO

- Observer Process communicates with both Primary and Standby
- Will initiate failover to standby if certain triggering events happen
  - Connectivity loss between the Primary and Standby or Primary and Observer AND user specified threshold timeout has expired
  - Database health check detects any of the failures at the Primary Database
    - Datafile has gone offline because of an I/O error
    - Control file is deemed to be corrupt
    - Log Writer (LGWR) process gets an I/O error and cannot write to any log file
    - ARCHIVER cannot write because of I/O error
    - Dictionary corruption is detected

# Data Guard FSFO

- Zero Data Loss Mode
  - Redo transport set to SYNC with MaxAvailability protection mode
- User-specified Data Loss Mode
  - Redo transport set to ASYNC with MaxPerformance protection mode
  - User can specify maximum amount of data loss (FastStartFailoverLagLimit)
- Reinstatement of the Failed Primary Database
  - Following the failover, Data Guard Broker will automatically try to reinstate the failed primary as a new standby database

# Program Agenda

- 1 Overview of Oracle GoldenGate
- 2 Overview of Integrated Extract / Integrated Replicat
- 3 Overview of Oracle Data Guard
- 4 Technical Challenges with Integration
- 5 Oracle's Bundled Agent(XAG) Extends HA for GoldenGate
- 6 Case Study with GoldenGate and Data Guard
- 7 Summary

# Challenge 1: Thread count Mismatch

- Active Data Guard widely used for offloading read-intensive applications
  - A large percentage of deployments are RAC
  - Number of threads *often* do not match between Primary and Standby database
- Post Role Transition
  - Thread counts likely different
  - *Need capability to handle such mismatches transparently*

## Challenge 2: Resetlogs Change on Failover

- Fast Start Failover (FSFO) widely used
  - Failover *ALWAYS* results in creation of a new database incarnation
  - Depending on situation, multiple fast start failovers can happen in a short period of time
  - *Need to handle resetlogs operation transparently*

# Challenge 3: Fuzziness in Redo Data

- Zero Data Loss Guarantee
  - SYNC Transport
  - Redo is written in parallel to standby and online redo logs
  - Commits are not acknowledged to the user until an ACK is received from Standby
  - Redo state is fuzzy until ACK is received
    - Commit in both ORL and SRL (Good case)
    - Commit in ORL but not in SRL
    - Commit in SRL but not in ORL
    - Commit in neither ORL nor SRL (Good Case)
  - *Need a way to avoid redo fuzziness during Extract*

# Challenge 4: GoldenGate must stay in sync with a data loss failover

- In an ASYNC FSFO configuration, the user can define an acceptable window of dataloss. Transitions of this nature expect some amount of data loss so the replication must account for this.
  - GoldenGate must remain consistent with the failover target database

# Challenge 5: Migrating the GoldenGate Stack on Failures

- After a RAC node fails GoldenGate must resume
  - GoldenGate processes should not require manual intervention to restart on node failures if other nodes are available
- Similarly after role transitions, the GoldenGate processes must be cleaned up at the old primary and restored on the new primary
  - In earlier releases this was a cumbersome process that involved user scripting and role transition triggers
  - *GoldenGate must seamlessly follow the primary database*



# Challenge 6: Consistency for GoldenGate data/metadata

- Primary and Standby databases likely do not share storage
- GoldenGate files must be available post role transition
  - Checkpoint file
  - Bounded Recovery file
  - Trail file
  - Parameter file
  - *GoldenGate must be kept consistent post role transition*
    - *True for both no data loss and data loss transitions*

# Challenge 7: Routing of trail data post role transition

- After a Data Guard role transition of a TARGET database;
  - Pump may need to send trail data to new primary
  - Replicat processes wait until new trail file data is received
  - **Need mechanism to ensure proper routing of trail after transition events**

# Why tell you this?

- Understand the technical and procedural complexity around the role transition events
- Understand what Zero Data Loss means
  - Database is the source of truth
  - Only the RDBMS can give you zero data loss
  - **Must** be integrated with the RDBMS software

# Program Agenda

- 1 Overview of Oracle GoldenGate
- 2 Overview of Integrated Extract / Integrated Replicat
- 3 Overview of Oracle Data Guard
- 4 Technical Challenges with Integration
- 5 Oracle's Bundled Agent(XAG) extends HA for GoldenGate
- 6 Case Study with GoldenGate and Data Guard
- 7 Summary

# Oracle Bundled Agent (XAG)

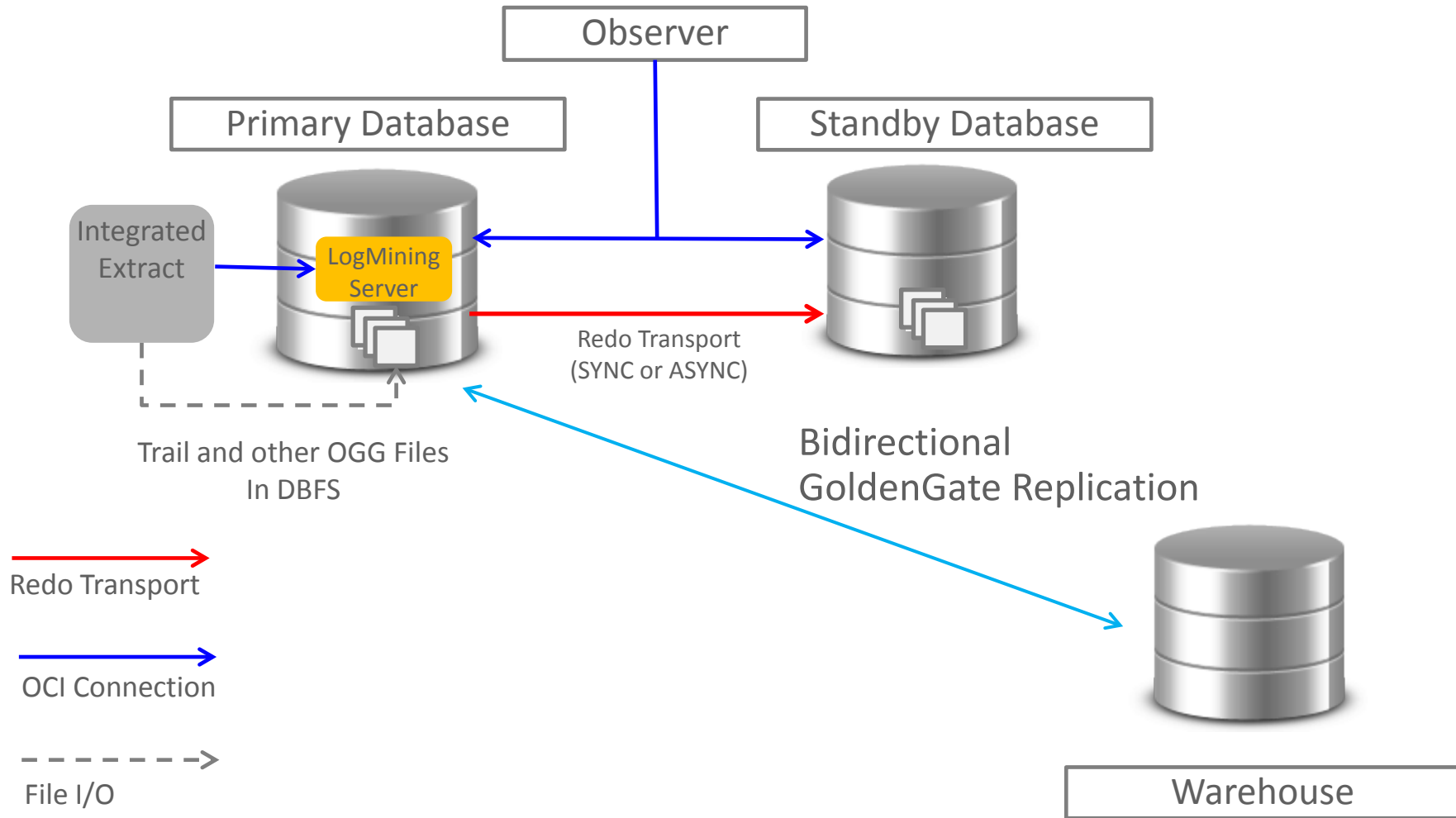
- Clusterware specific to managing GoldenGate resources
  - XAG allows you to register a GoldenGate instance with CRS to improve HA
- Uses AGCTL for registering/starting and stopping resources
- It solves the key process related issue of detecting and migrating a GoldenGate instance, i.e. full failover support for GoldenGate
  - XAG provides increased availability in the face of failures:
    - Loss of instance (RAC node failure)
    - Loss of primary database (Data Guard Failover integration)

# Program Agenda

- 1 Overview of Oracle GoldenGate
- 2 Overview of Integrated Extract / Integrated Replicat
- 3 Overview of Oracle Data Guard
- 4 Technical Challenges with Integration
- 5 Oracle's Bundled Agent(XAG) Extends HA for GoldenGate
- 6 Case Study with GoldenGate and Data Guard
- 7 Summary

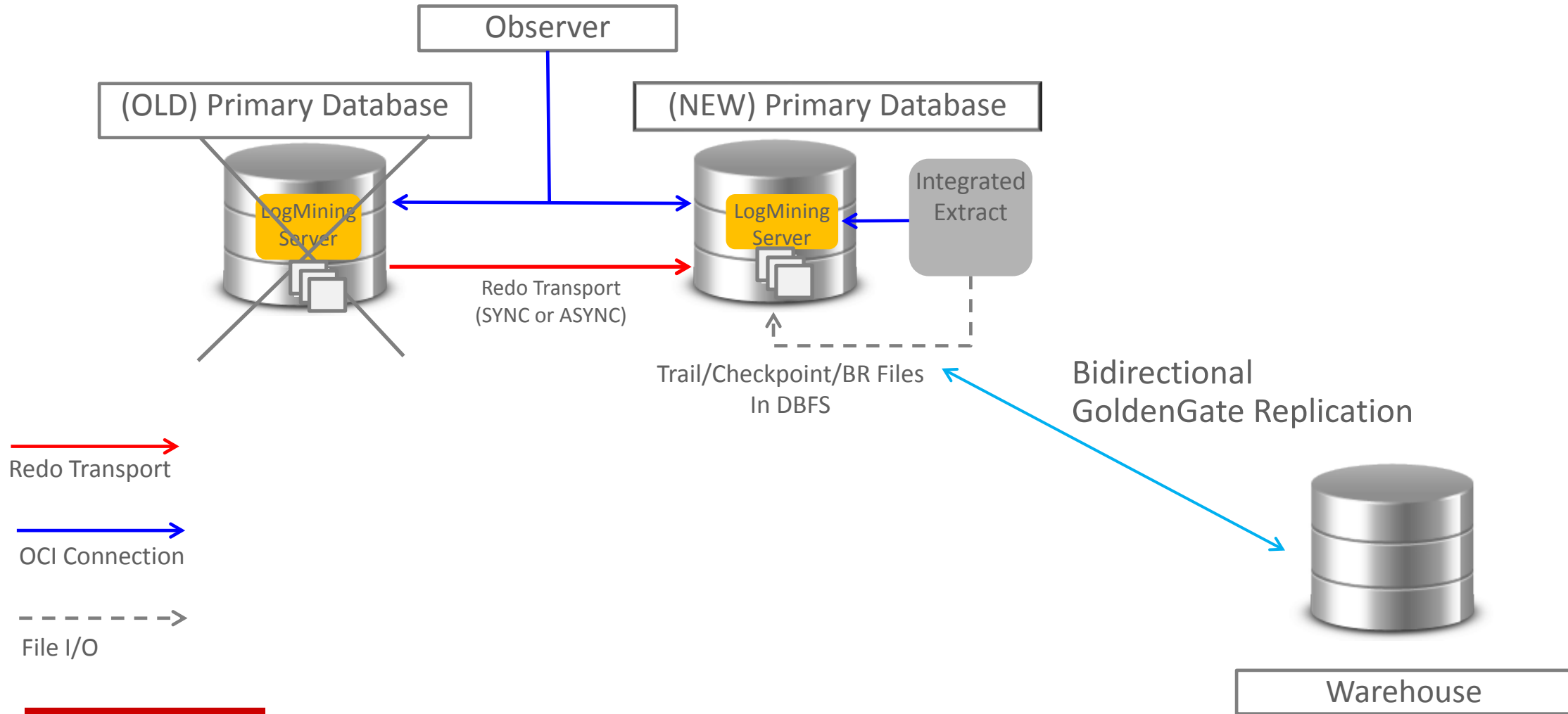
# Case Study with GoldenGate and Data Guard

# Sample Deployment





# Sample Deployment – Post Role Transition



# Database Connectivity

- Make all OGG components connect to database using Role-Based Services
  - Declarative way to specify a service should be published only when the database has a specific role
  - **Publish a service only when database has the PRIMARY role**
- Create role-based service:

```
srvctl add service -db GG2PRMY -service oggserv -role PRIMARY  
-notification TRUE -tafpolicy BASIC -failovermethod BASIC  
-failoverdelay 3 -failoverretry 60 -failovertype SELECT  
-preferred GGS21 -available GGS22
```

```
srvctl add service -db GG2STBY -service oggserv -role PRIMARY  
-notification TRUE -tafpolicy BASIC -failovermethod BASIC  
-failoverdelay 3 -failoverretry 60 -failovertype SELECT  
-available GGS21
```

# Net Aliases

- Net Alias in tnsnames.ora at Primary

```
– ggconn = (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)
(HOST=BOSTON-SCAN) (PORT=2140)) (FAILOVER=on)(LOAD_BALANCE=off)
(CONNECT_DATA= (SERVICE_NAME=oggserv)
(FAILOVER_MODE=(TYPE=SESSION)(METHOD=BASIC)(RETRIES=20)(DELAY=60))))
```

- Net Alias in tnsnames.ora at Standby

```
– ggconn = (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)
(HOST=CHICAGO-SCAN) (PORT=2140)) (FAILOVER=on)(LOAD_BALANCE=off)
(CONNECT_DATA= (SERVICE_NAME=oggserv)
(FAILOVER_MODE = (TYPE=SESSION) (METHOD=BASIC) (RETRIES=20) (DELAY=60))))
```

# Create DBFS CRS Resource

- DBFS is mounted on RAC node that will run GoldenGate
- Create CRS action script (MOS note 1054431.1)
  - Must contain START, STOP, CHECK functions
- Create CRS resource:

```
crsctl add resource dbfs_mount -type cluster_resource \  
-attr "ACTION_SCRIPT=$ACTION_SCRIPT, CHECK_INTERVAL=30,RESTART_ATTEMPTS=10, \  
START_DEPENDENCIES='hard(ora.$DBNAME.db)pullup(ora.$DBNAME.db)',\  
STOP_DEPENDENCIES='hard(ora.$DBNAME.db)', SCRIPT_TIMEOUT=300"
```

- DBFS service will be managed from Bundled Agent

# Register GoldenGate with Bundled Agent (XAG)

- Register with XAG at the primary

```
agctl add goldengate ggprmy --gghome /u01/oracle/goldengate --oracle_home $ORACLE_HOME  
--db_services ora.oggserv.svc --filesystems dbfs_mount --monitor_extracts ext1,dpump1  
--monitor_replicats rep1--environment_vars 'TNS_ADMIN=...' --dataguard_autostart yes  
--nodes scac07adm05,scac07adm06 --vip_name gg_vip_source
```

- Register with XAG at the standby

```
agctl add goldengate ggstby --gghome /u01/oracle/goldengate --oracle_home $ORACLE_HOME  
--db_services ora.oggserv.svc --filesystems dbfs_mount --monitor_extracts ext1,dpump1  
--monitor_replicats rep1 --environment_vars 'TNS_ADMIN=...' --dataguard_autostart yes  
--nodes scab02adm06
```

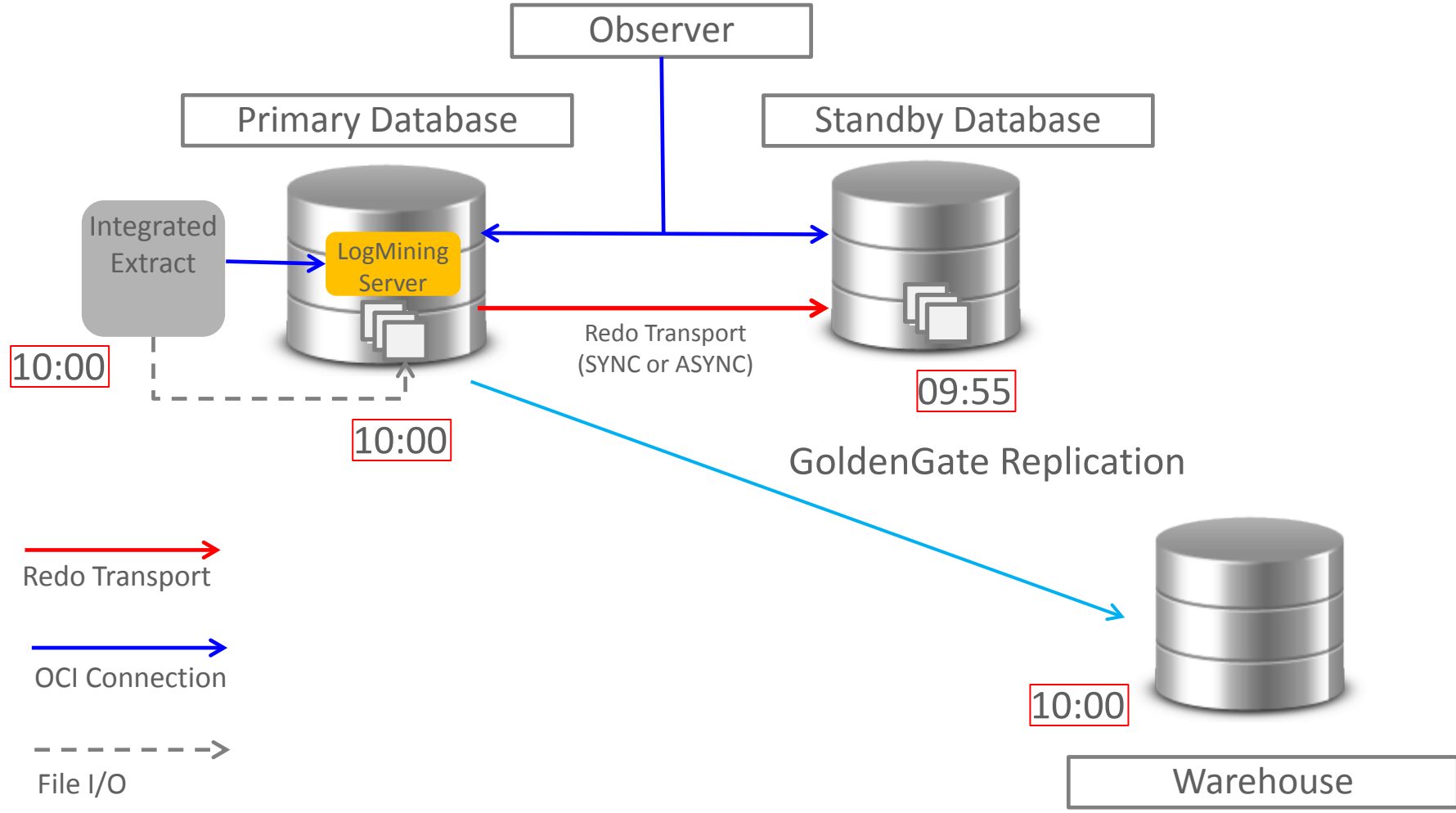
- Start Extract using Agent Control

– agctl start goldengate ggprmy

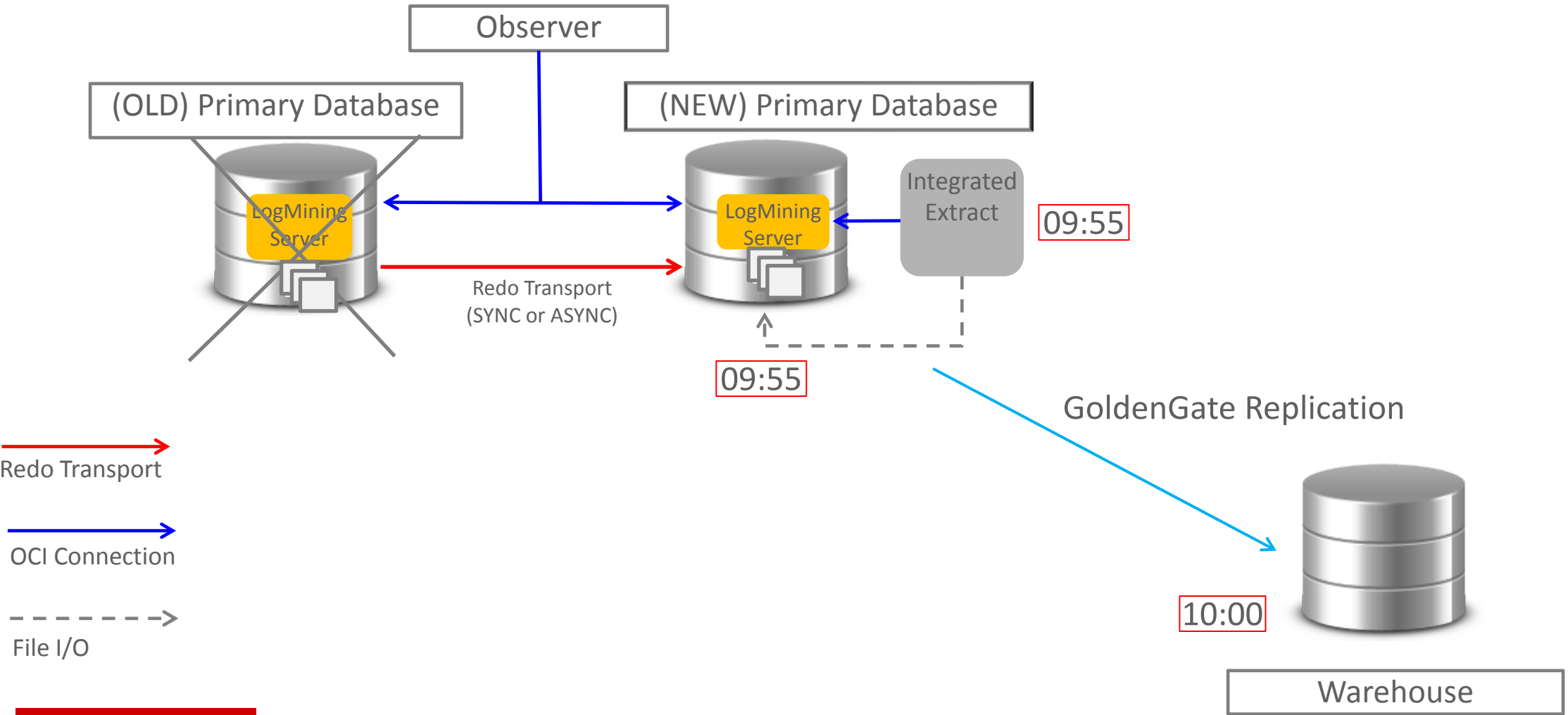
# Data Guard – Bounded Data Loss (ASYNC)

- Data Guard in MaxPerformance(ASYNC) mode permits data loss on the standby
- Amount of data loss with FSFO controlled by FastStartFailoverLagLimit (default is 30seconds)
- Extract must only mine from redo already applied to Standby
  - Prevents data being replicated to a target database and then missing from the source after a failover
- Add Integrated Extract parameter `TRANLOGOPTIONS HANDLEDLFAILOVER`

# Data Guard – Bounded Data Loss (ASYNC)

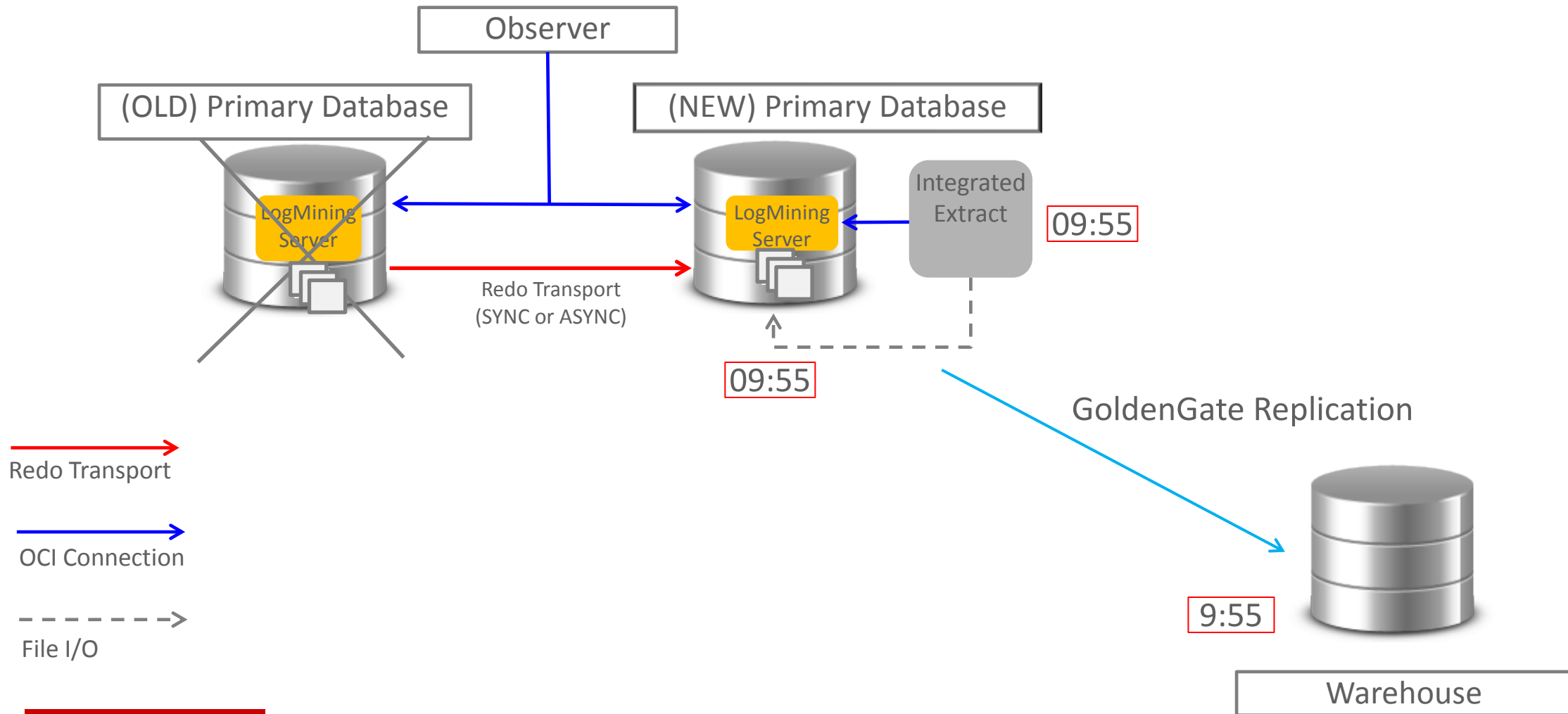


# Data Guard – Bounded Data Loss (ASYNC)





# Data Guard – Bounded Data Loss (ASync) tranlogoptions HANDLEDLFAILOVER



# Program Agenda

- 1 Overview of Oracle GoldenGate
- 2 Overview of Integrated Extract / Integrated Replicat
- 3 Overview of Oracle Data Guard
- 4 Technical Challenges with Integration
- 5 Oracle's Bundled Agent(XAG) Extends HA for GoldenGate
- 6 Case Study with GoldenGate and Data Guard
- 7 Summary

# GoldenGate's Integrated Stack is Engineered for Data Guard

- RAC instance addition/removal
  - Thread count change based on DG role transition handled without user intervention
  - Transparent support for RAC-One
- Resetlogs
  - Will automatically detect resetlogs operation in redo logs and take the correct branch of redo
  - Transparent handling of repositioning in presence of resetlogs operation
- Redo fuzziness around failover
  - In local mode, knows to avoid fuzziness (stays behind unacknowledged commit)
  - In downstream mode, can be configured to avoid redo fuzziness
- It handles dataloss role transitions
  - TRANLOGOPTIONS HANDLEDLFAILOVER

# GoldenGate extends HA with XAG and DBFS

- XAG seamlessly relocates GoldenGate on instance or primary failure
- Consistency of GoldenGate data and metadata ensured with DBFS
  - Store trail and checkpoint data on DBFS to guarantee consistency on role transitions
- Provides flexible routing of trail data post role transition
  - Use a Virtual IP to identify the TARGET host machine
    - Transferred between primary and standby on role transition
  - Enable AUTOSTART of the Data Pump process on the SOURCE
    - After a role transition completes Data Pump will restart sending trails to new primary

# Summary

- Important MOS Notes/White Papers
  - Client Failover Best Practices for Highly Available Oracle Databases: Oracle Database 12c (MAA White Paper, September 2014)
    - <http://www.oracle.com/technetwork/database/availability/client-failover-2280805.pdf>
  - Oracle GoldenGate with Real Application Clusters Configuration (MAA White Paper, August (2013))
    - <http://www.oracle.com/technetwork/database/features/availability/maa-goldengate-rac-2007111.pdf>

# **Hardware and Software Engineered to Work Together**

ORACLE®

# Questions and Answers





# Resources

Oracle Data Integrator



Oracle Data Integration

Oracle GoldenGate



OracleDataintegration

Oracle Enterprise Data Quality



ORCLGoldenGate

Oracle Enterprise Metadata Management



blogs.oracle.com/dataintegration

Oracle Data Services Integrator



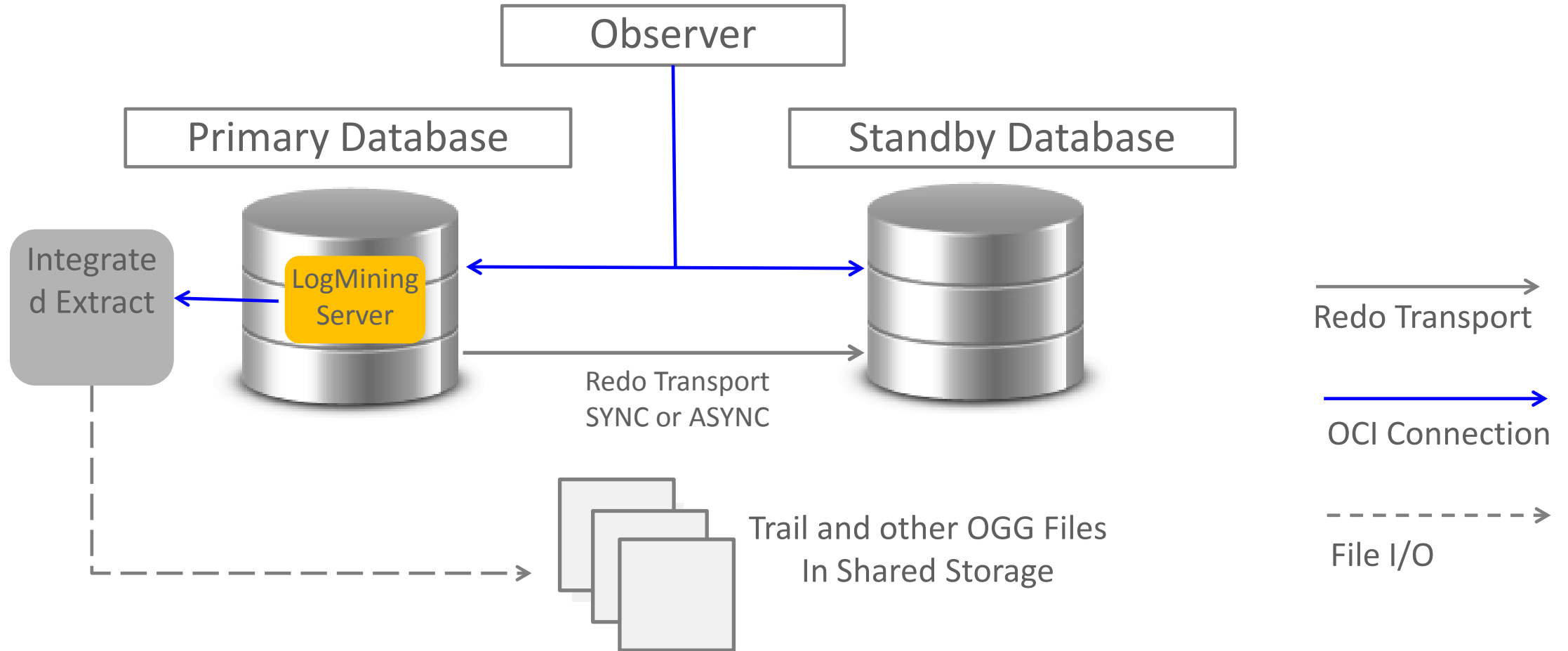
OracleGoldenGate

**ORACLE**<sup>®</sup>  
Data Integration

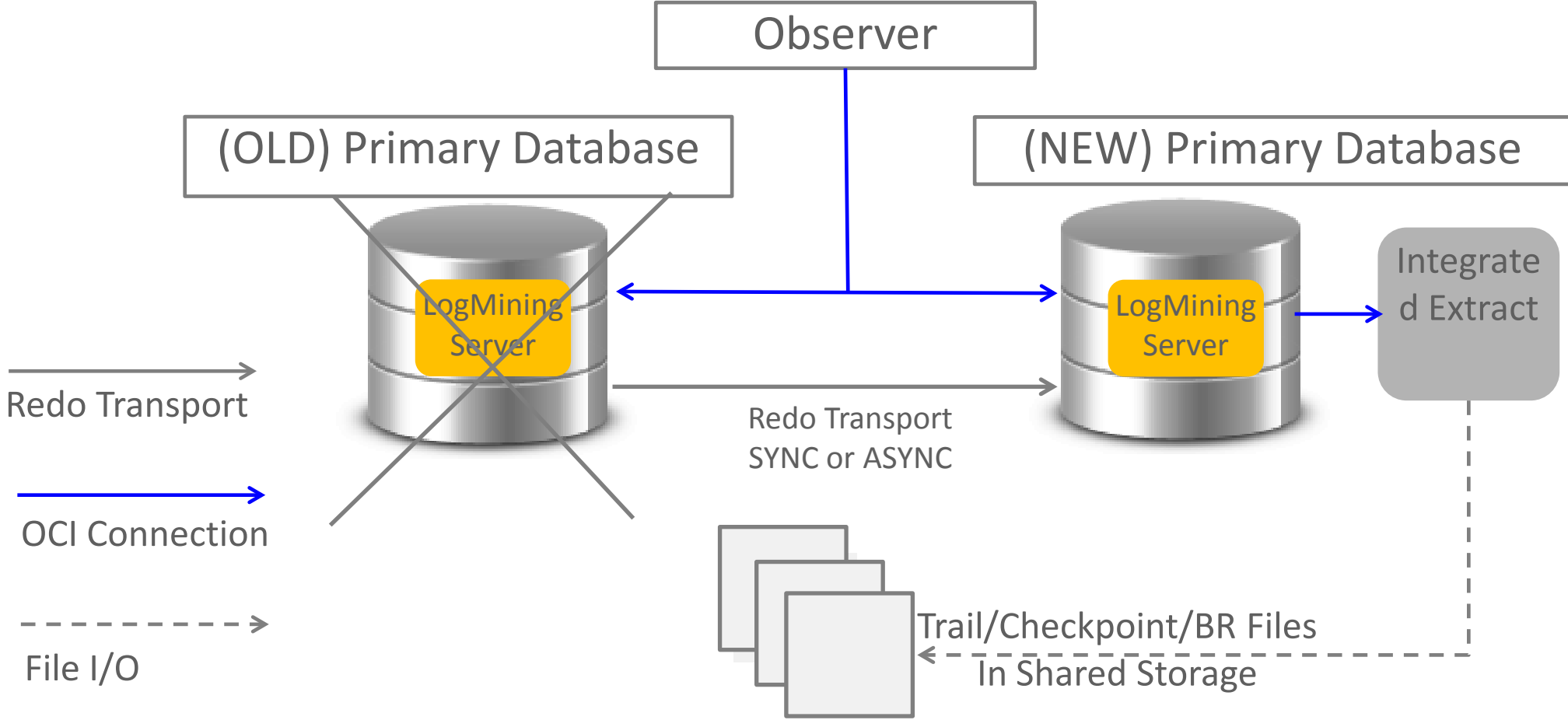
<http://www.oracle.com/us/products/middleware/data-integration/overview/index.html>

# **Hardware and Software Engineered to Work Together**

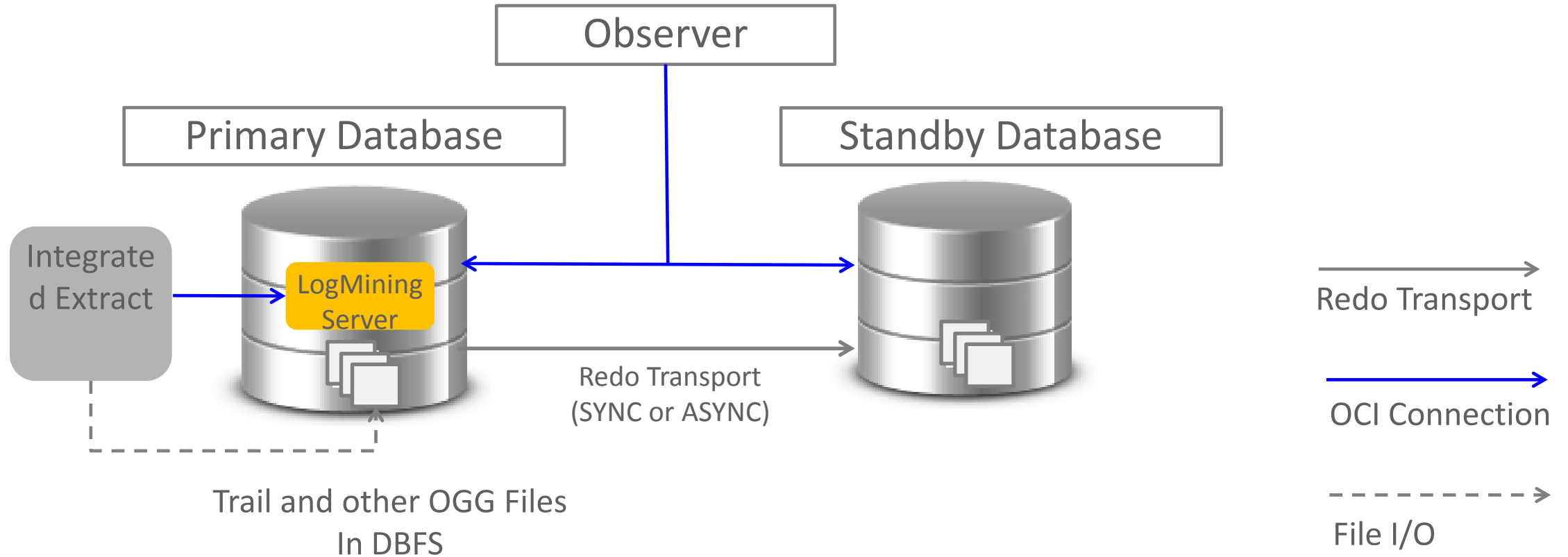
# Integrated Extract (Shared Storage): On-Source



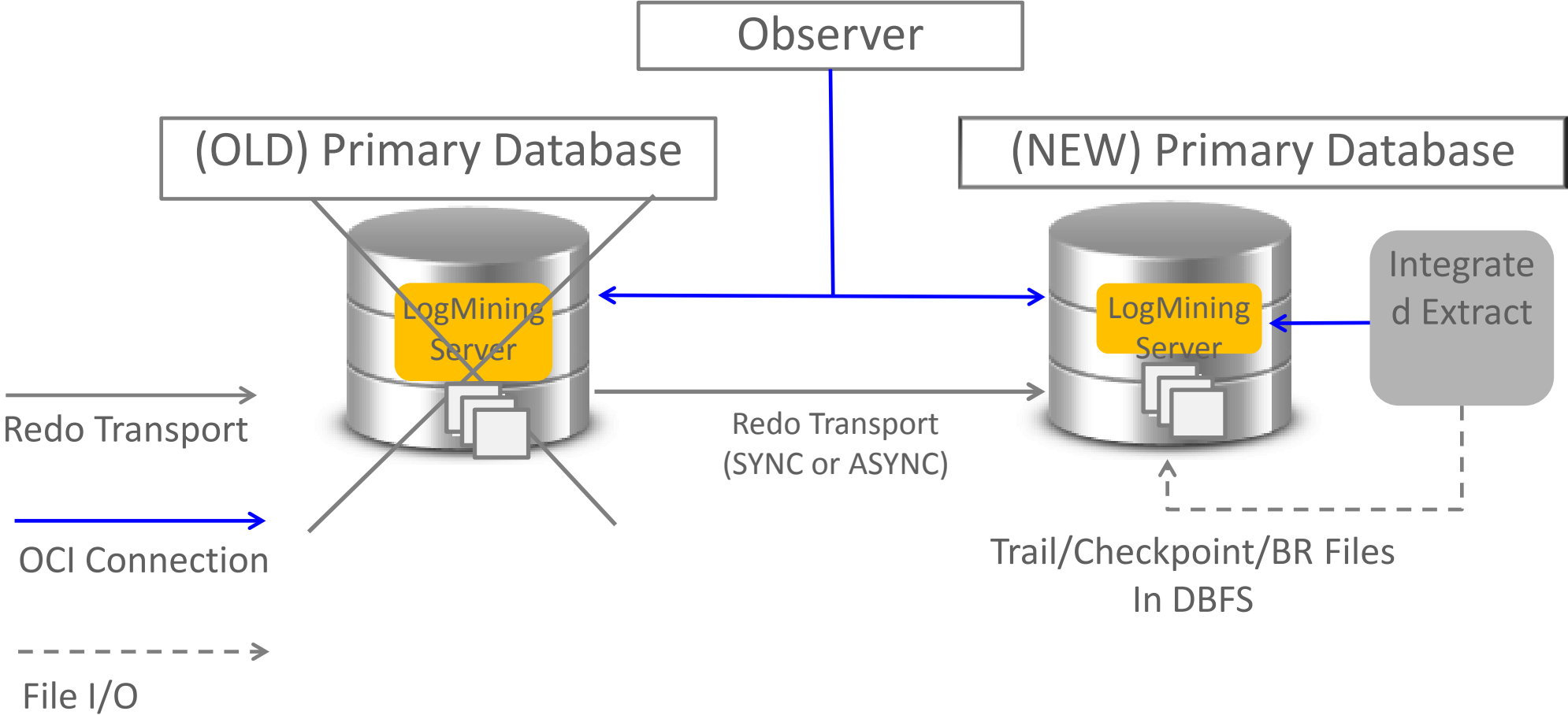
# Integrated Extract (Shared Storage): On-Source (Post- Failover)



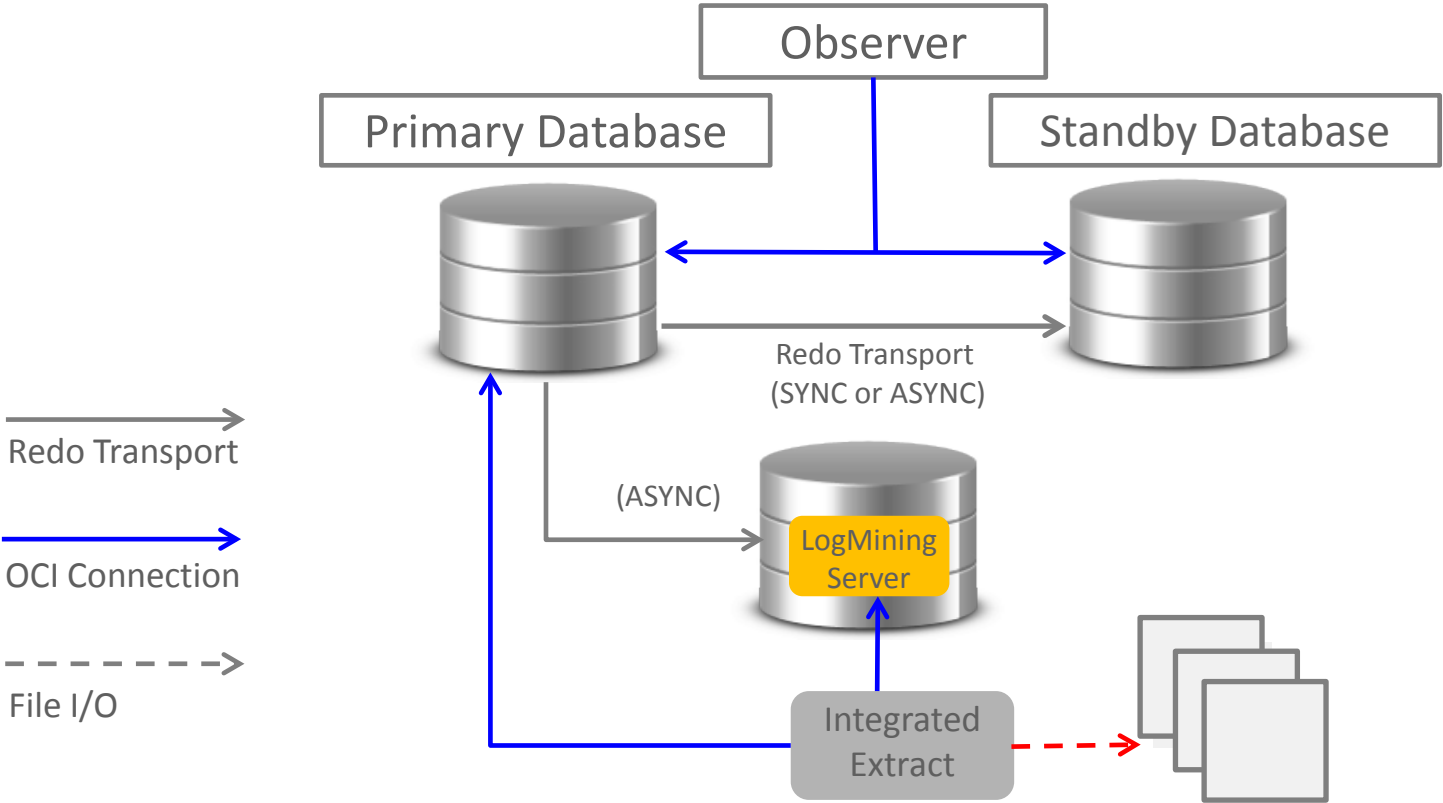
# Integrated Extract (DBFS): On-Source Deployment



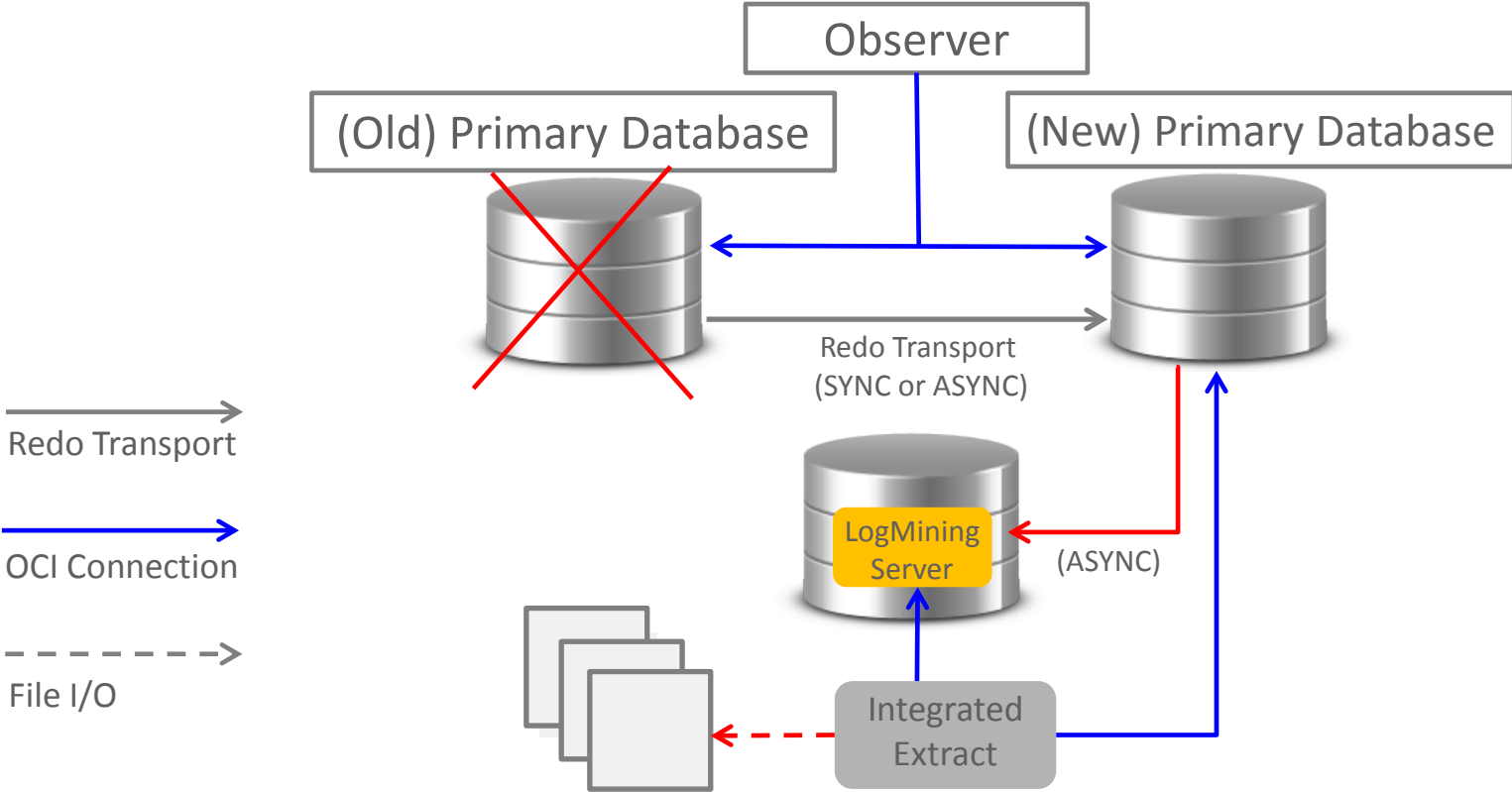
# Integrated Extract (DBFS): On-Source Post-Failover



# Integrated Extract: Downstream Deployment for Data Guard



# Integrated Extract: Downstream Deployment Post-Failover





# Integrated Extract: Downstream Deployment Cascaded Transport

