ORACLE

Oracle Global Data Services (GDS)

Automated Workload Management for Replicated Databases

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

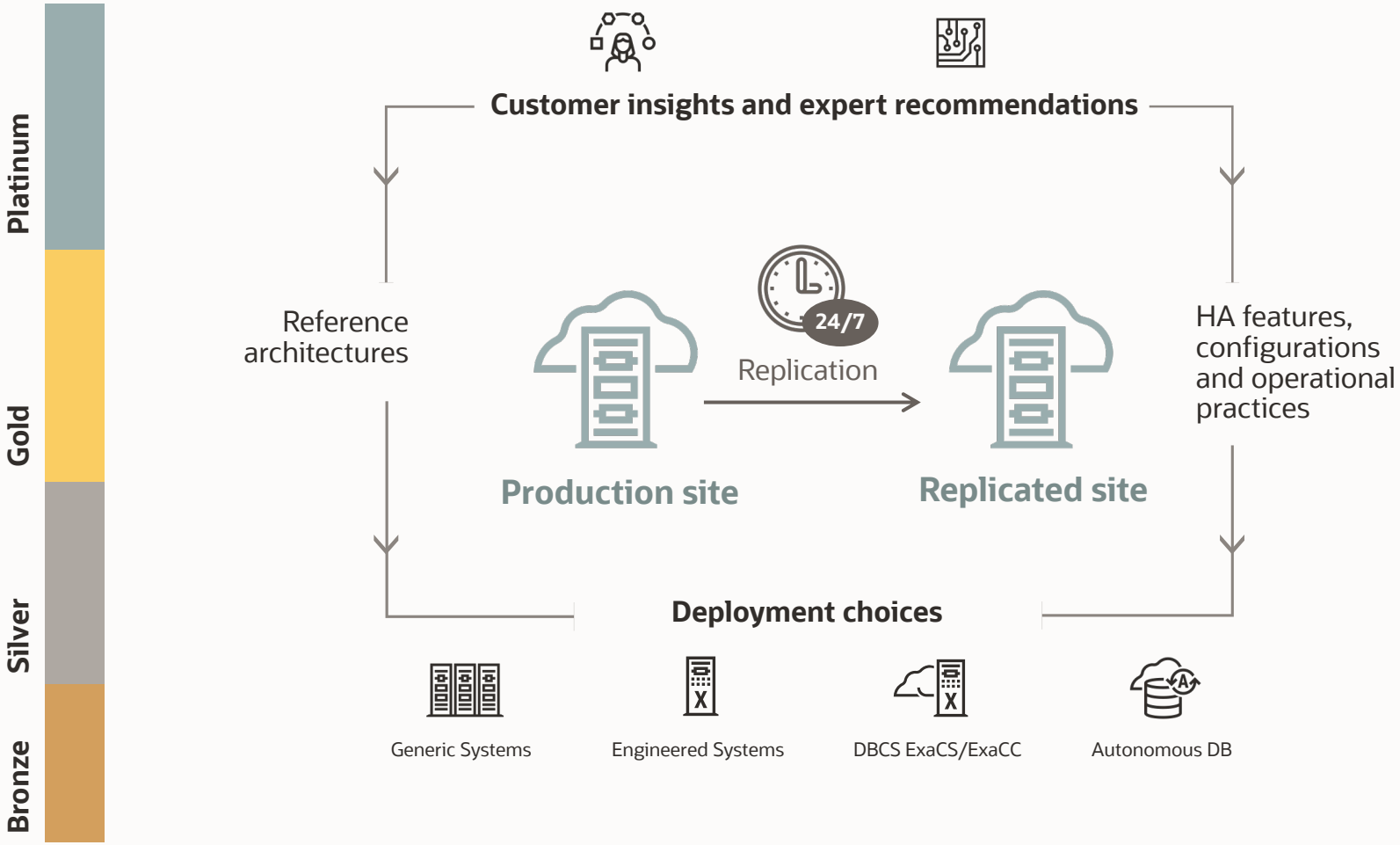
Challenges of deploying highly available systems

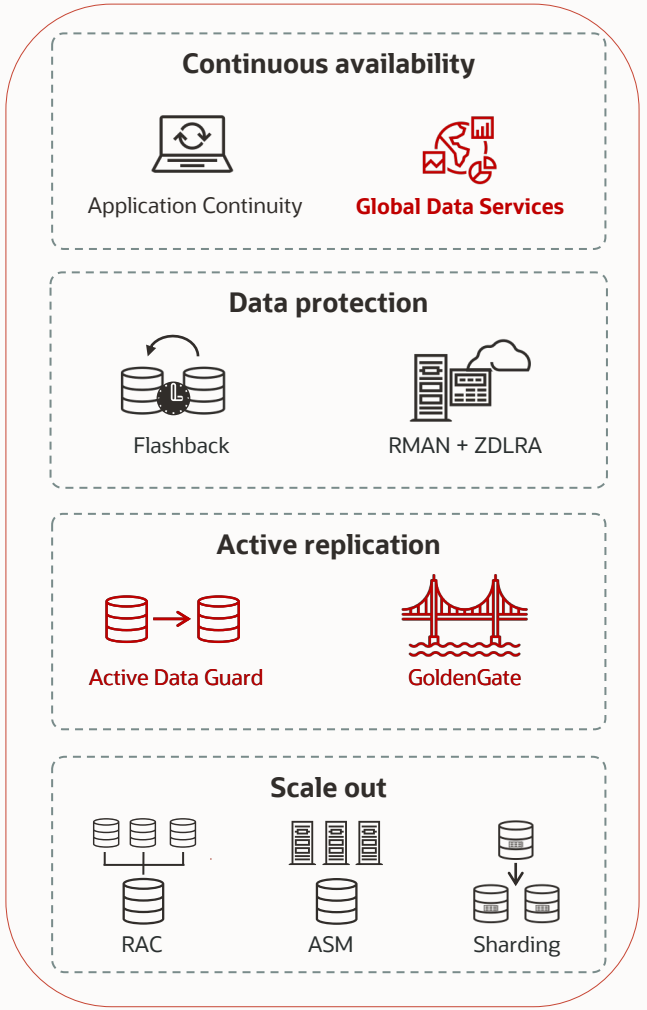
Cost and complexity

Lack of skills

Risk of failure

Oracle Maximum Availability Architecture (MAA)





MAA reference architectures

Availability service levels

Bronze	Silver	Gold	Platinum
Dev, test, prod	**Prod/departmental**	**Business critical**	**Mission critical**
Single instance DB Restartable Backup/restore	**Bronze +** Database HA with RAC Application continuity	**Silver +** DB replication with Active Data Guard	**Gold +** GoldenGate Edition based redefinition

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

Agenda

1 Workload management challenges of replicas

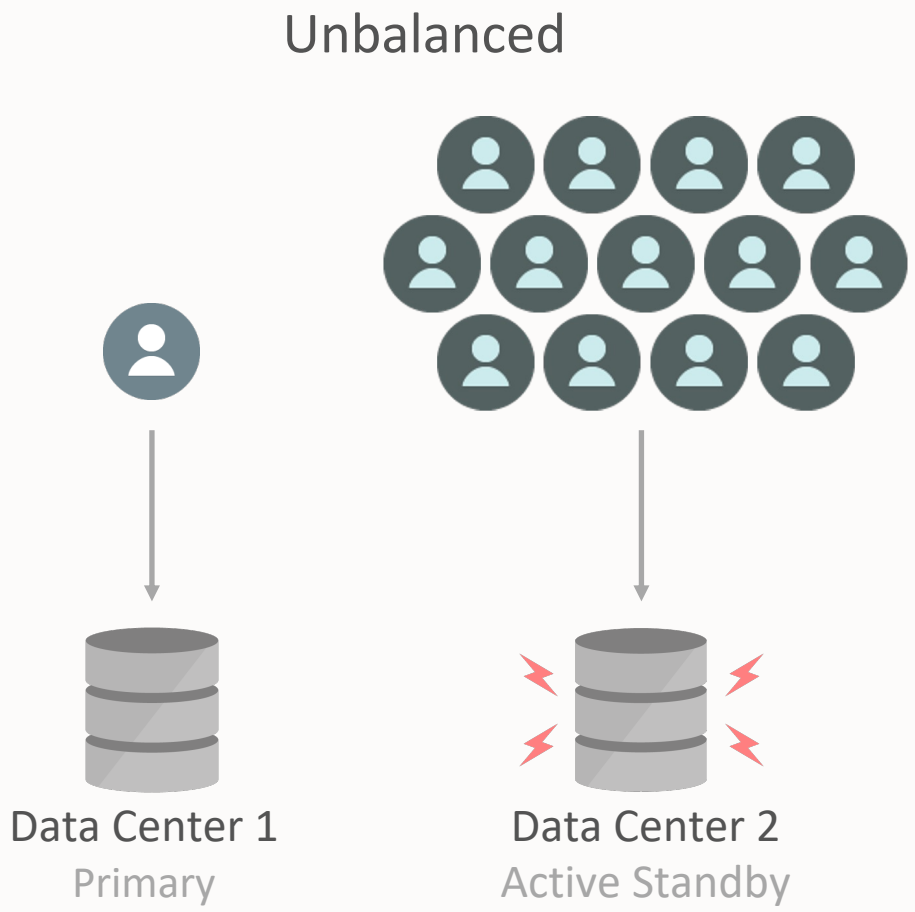
2 Introduction to Global Data Services (GDS)

3 GDS concepts and architecture

4 GDS use cases

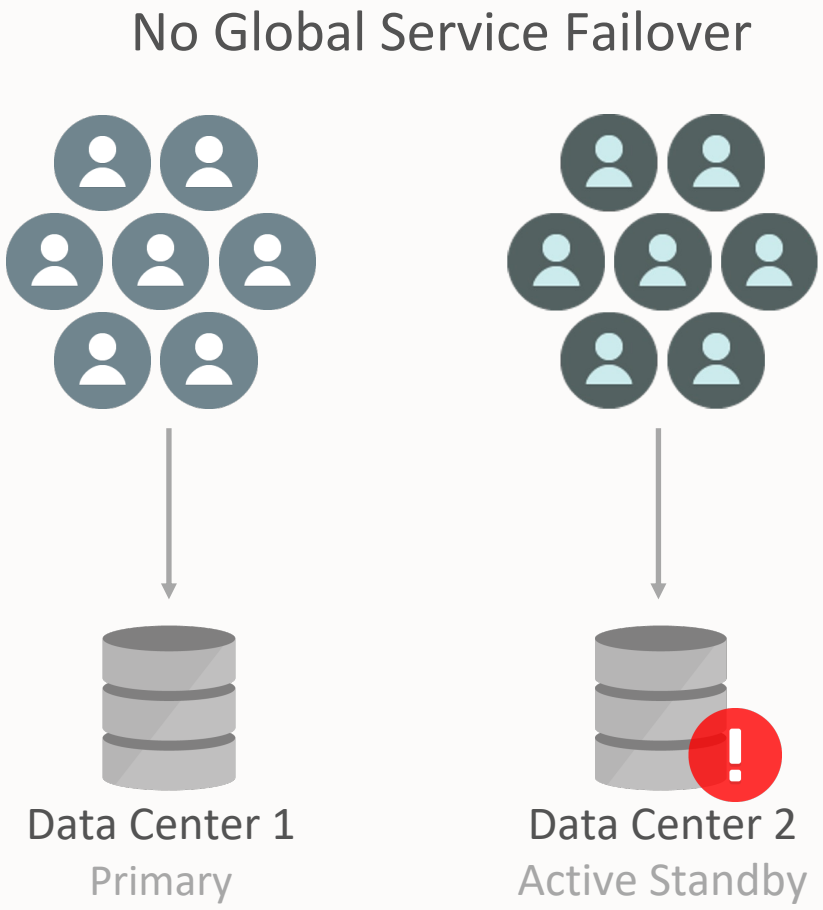
5 Summary

Challenges of Replicas – Workload Balance



- No automated load balancing
- Sub-optimal resource utilization

Challenges of Replicas – Service Failover



- App outages when replicas fail
- No Service HA

Agenda

1 Workload management challenges of replicas

2 Introduction to Global Data Services (GDS)

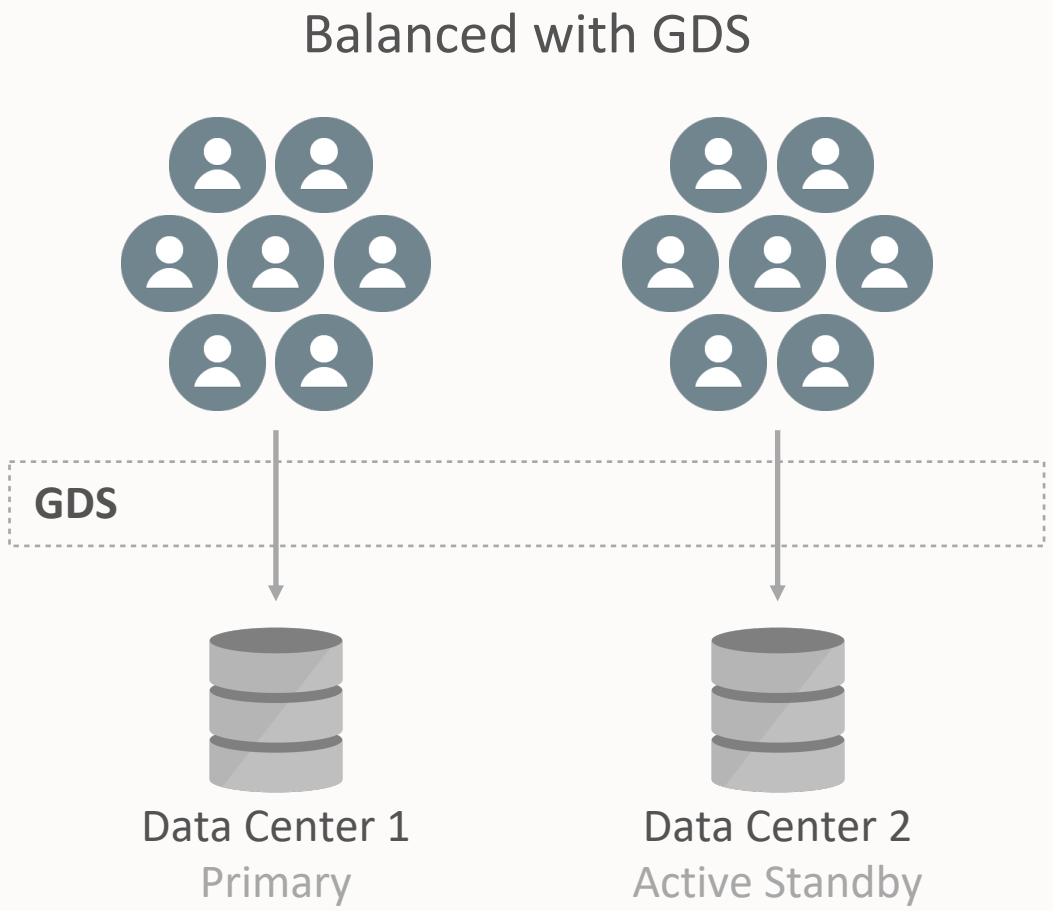
3 GDS concepts and architecture

4 GDS use cases

5 Summary

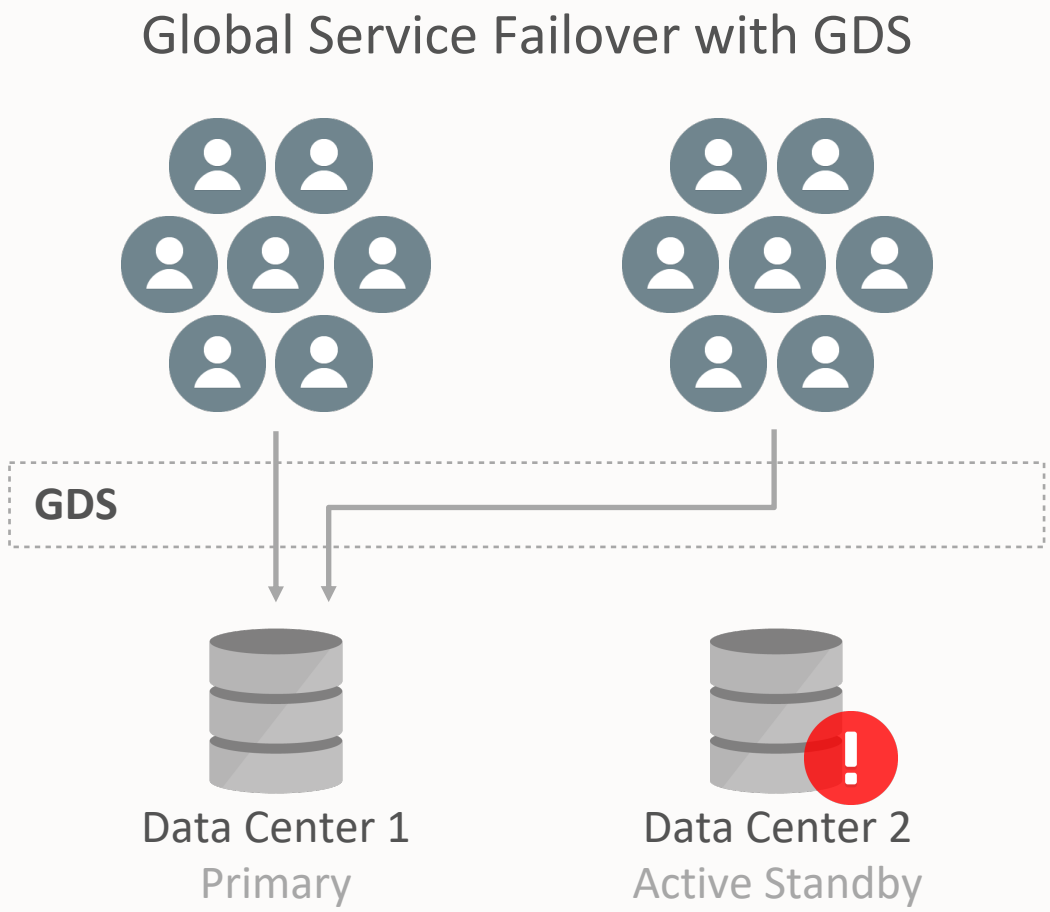
Oracle Global Data Services (GDS)

- Automatic and transparent client workload management across replicas
- Extends the concept of services to replicas
- Capabilities
  - Workload routing based on load, locality or lag
  - Service failover across replicas
- Benefits
  - Maximize application performance
  - Mitigate downtime during planned and unplanned outages
  - Manage resources of replicas with one interface



Oracle Global Data Services (GDS)

- Automatic and transparent client workload management across replicas
- Extends the concept of services to replicas
- Capabilities
  - Workload routing based on load, locality or lag
  - Service failover across replicas
- Benefits
  - Maximize application performance
  - Mitigate downtime during planned and unplanned outages
  - Manage resources of replicas with one interface



Workload Management for Database Replicas with GDS

Centralized service management

Workload routing (region-based & lag-based)

Inter-database service failover

Role based global services

Load balancing (connect-time & run-time)

Agenda

1 Workload management challenges of replicas

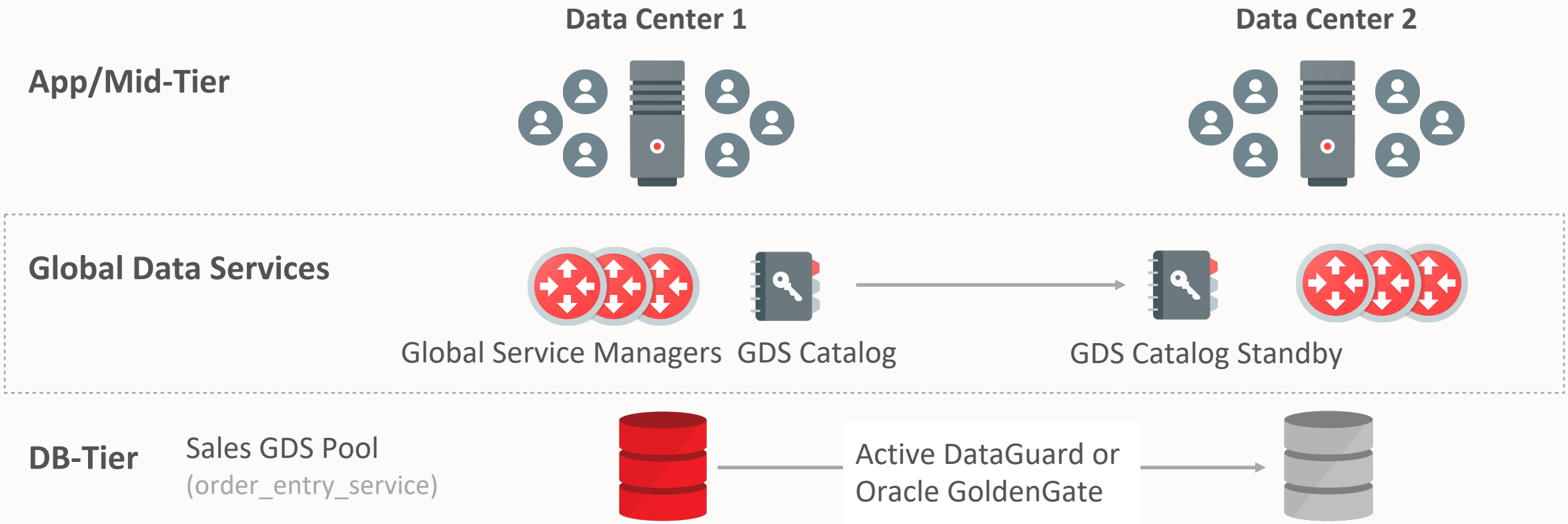
2 Introduction to Global Data Services (GDS)

3 GDS concepts and architecture

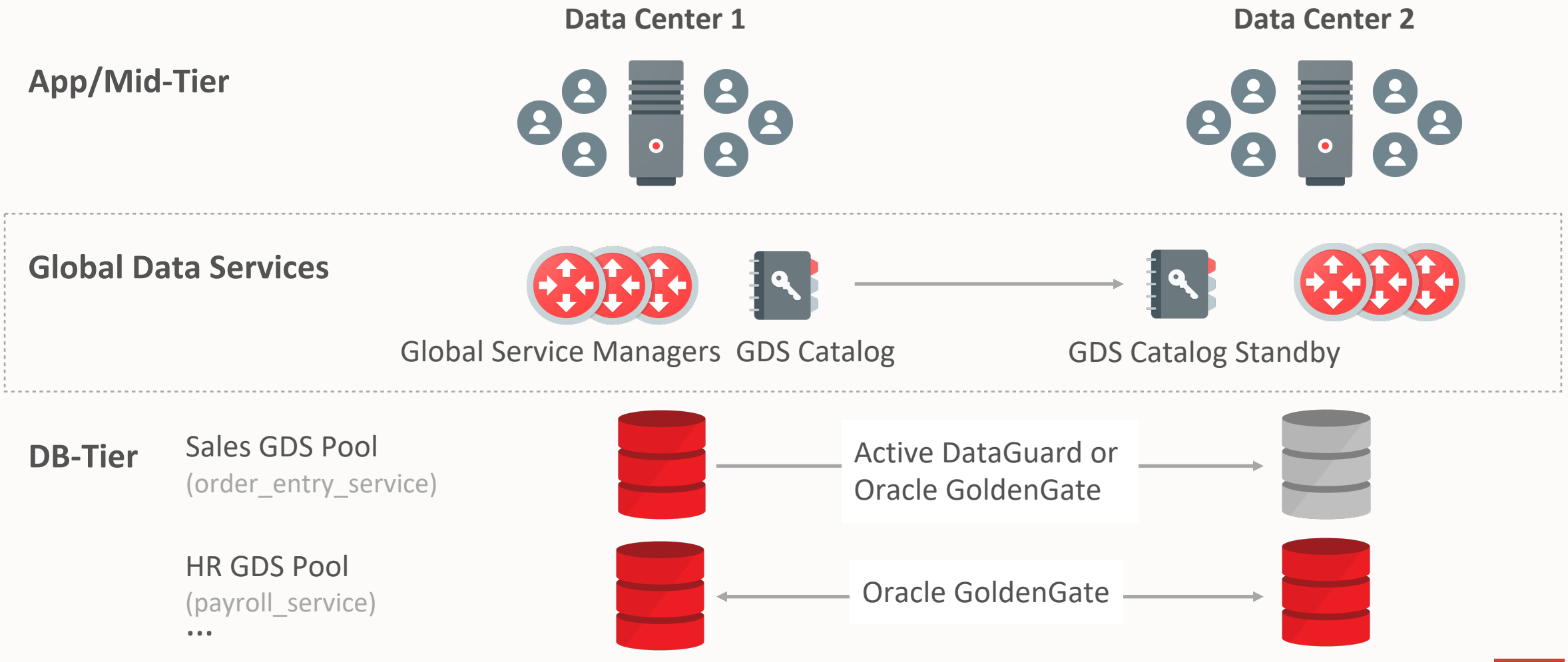
4 GDS use cases

5 Summary

GDS Architecture



One GDS Infrastructure For Many Replicated Configurations



GDS Components

- **Global Service Manager (GSM)**
  - Regional listener to the incoming database connections
  - Performs Connect-time load balancing
  - Publishes FAN events (via ONS) for service failovers and run-time load balancing advisory
  - Inter-database Service failover & management
- **GDS Catalog** - Stores GDS configuration metadata
- **GDS Region** - Group of databases and clients in close network proximity, e.g., East, West
- **GDS Pool** - Databases that offer a common set of global services, e.g., HR, Sales
- **Global Service** - Database Service provided by multiple databases with replicated data
  - Local service + {Locality, replication lag, role, database cardinality, load balancing goals}
  - Establish workload management policies via Service attributes

GDS – A shared infrastructure

A Single GDS manages

- 5000 GDS Pools
- 10 GDS Regions
- 5 GSMs per Region
- 10,000 Database instances
- 10,000 Global Services
- 1000 Mid-tier connection pools

GDS Databases

- Must be Oracle Database EE 12.1+
- Can be Single Instance or RAC
- Can be CDB or Non-CDB
- Can run on commodity or Engineered systems (Oracle Exadata, ODA)
- Managed with GDSCTL CLI or Enterprise Manager DB Plug-in
- Must be licensed for Active Data Guard or Oracle GoldenGate

GDS Deployment
High Level Steps

- Install GSM software on GSM servers
  - Min of 1 GSM per region
  - Recommended 3 GSMs/region
- Pre-create GDS catalog database
- Setup GDS Administrator accounts & privileges
- Configure GDS
  - Create GDS Catalog
  - Add GSMs, Regions, Pools, Databases, Global Services
- Setup client connectivity

GDS Deployment

Setup GDS Accounts & Privileges

- On the GDS Catalog database:

```
SQL> create user mygdsadmin identified by passwd_mygdsadmin;
SQL> grant gsmadmin_role to mygdsadmin;

SQL> alter user gsmcatuser account unlock;
SQL> alter user gsmcatuser identified by passwd_gsmcatuser;
```

- On each of the GDS Pool databases:

```
SQL> alter user gsmuser account unlock;
SQL> alter user gsmuser identified by passwd_gsmuser;
```

GDS Deployment
Configure GDS

From a GSM node, use GDSCTL to configure GDS

- **create catalog** -database <host_name>:1521:catdb.acme.com –user mygdsadmin/passwd_mygdsadmin –region siteA, siteB
- **add gsm** -gsm gsm1 -listener 1571 -catalog <host_name>:1521:catdb –region siteA
- **start gsm** –gsm gsm1

  …
- **add gdspool** -gdspool sales
- **add database** -connect <host_name>:1521:**db01** -gdspool sales –**region SiteA**
- **add database** –connect <host_name>:1521:**db02** -gdspool sales –**region SiteB**
- **add service** -service sales_qry_srvc -gdspool sales **-preferred** db01 **–available** db02
- **start service** -service sales_qry_srvc –gdspool sales
- For Data Guard, use “add brokerconfig” instead of “add database”

Client Connectivity in GDS – TNS Entry

```
sales_reporting_srvc =
  (DESCRIPTION =(CONNECT_TIMEOUT=90)(RETRY_COUNT=30)(RETRY_DELAY=3)(TRANSPORT_CONNECT_TIMEOUT=3)
   (FAILOVER=ON)
    (ADDRESS_LIST =                                          ← DatacenterA's GSMs
     (LOAD_BALANCE=ON)
     (ADDRESS = (PROTOCOL = TCP)(HOST = gsm-host1a)(PORT = 1571))
     (ADDRESS = (PROTOCOL = TCP)(HOST = gsm-host2a)(PORT = 1571))
     (ADDRESS = (PROTOCOL = TCP)(HOST = gsm-host3a)(PORT = 1571))

    )
    (ADDRESS_LIST =                                          ← DatacenterB's GSMs
     (LOAD_BALANCE=ON)
     (ADDRESS = (PROTOCOL = TCP)(HOST = gsm-host1b)(PORT = 1572))
     (ADDRESS = (PROTOCOL = TCP)(HOST = gsm-host2b)(PORT = 1572))
     (ADDRESS = (PROTOCOL = TCP)(HOST = gsm-host3b)(PORT = 1572))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = sales_reporting_srvc.sales.oradbcloud) (REGION=WEST)
    )
  )
```

GDS-Ready Application - Requirements

- Define the Global Services as per the application requirements
- Use Oracle Integrated Connection Pools/Drivers (OCI, JDBC, ODP.NET, WebLogic)
  - IBM WebSphere, Apache Tomcat, Red Hat JBoss are supported when using Oracle UCP
  - For UCP, include the ojdbc8.jar, ucp.jar and ons.jar in the CLASSPATH
- Connection URL (or TNS entry) must include:
  - GSM Listener end points
  - CONNECT_TIMEOUT, RETRY_COUNT, RETRY_DELAY, TRANSPORT_CONNECT_TIMEOUT parameters
  - SERVICE_NAME
  - For locality based routing, specify client's REGION
- Use 12.2 clients - Fast Connection Failover (FCF) is auto-enabled
  - For pre 12.2 clients, enable (FCF) via `setFastConnectionFailoverEnabled = true`
- Set planned draining period system property for graceful draining
  - For UCP `-Doracle.ucp.PlannedDrainingPeriod=30`

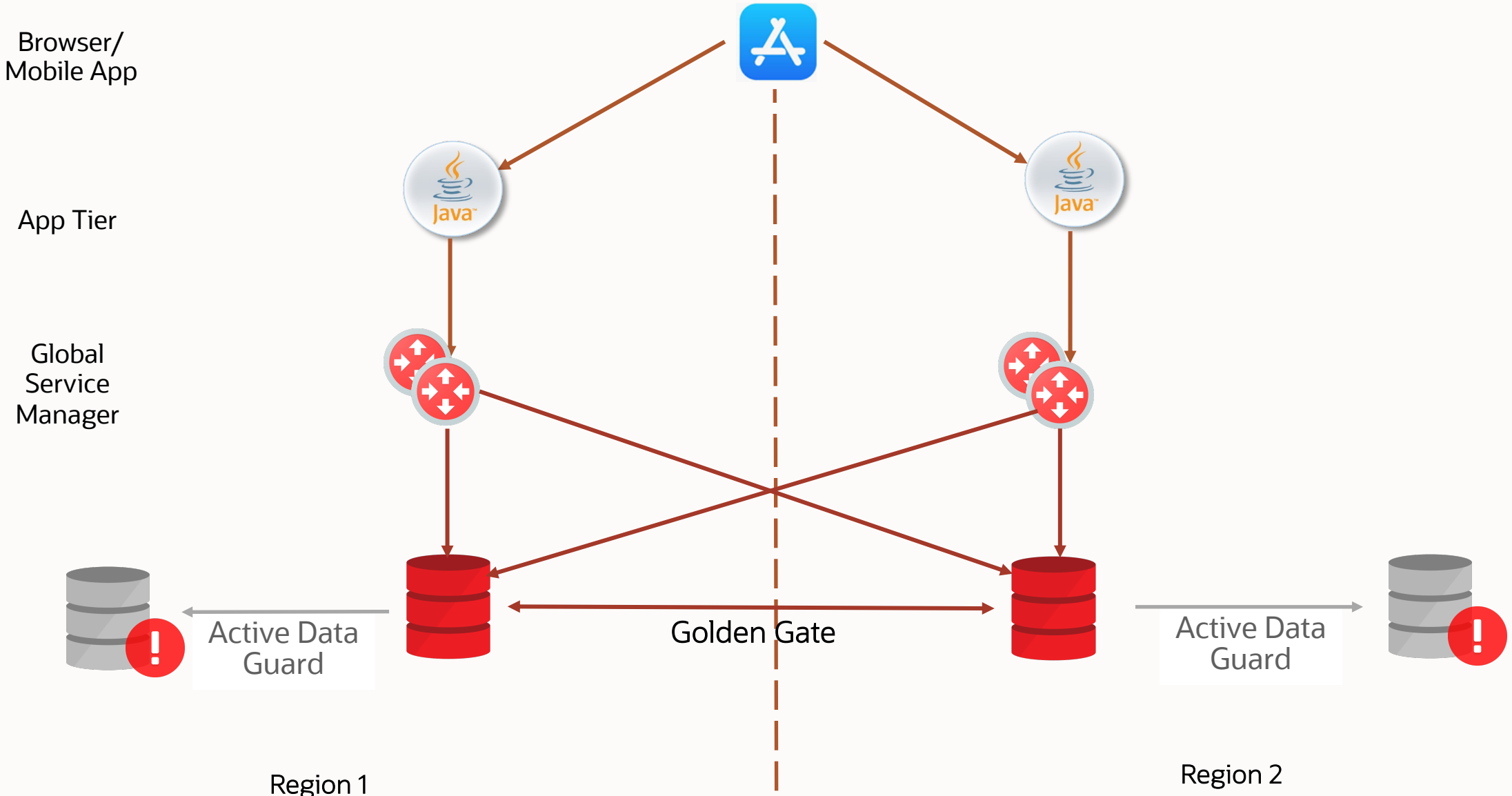
Supported GDS Clients for Load Balancing & Failover

- All clients
  - Connect-time Load Balancing (CLB) across databases
    - Choose the best DB instance to connect
- Oracle integrated connection pool based clients
  - Run-time load balancing (RLB)
    - Selecting a cached connection (belonging to least loaded database instance) for a work request
- Gravitation
  - Redistribution of connections between instances
    - Instances that are able to process more database requests have more connections established to them
- Proactive handling of instance UP and DOWN events
  - Remove dead connections from pool preventing application from using them
  - Proactively establish connections to a instance which was restarted
- Oracle clients use GDS's ONS to receive FAN events (CLB, RLB & Fast Connection Failover (FCF))

Agenda

1 Workload management challenges of replicas
2 Introduction to Global Data Services (GDS)
3 GDS concepts and architecture
4 GDS use cases
5 Summary

Active-Active Multi-Region Deployment with OGG and ADG



Discover Financial Services (DFS)

Digital banking and payment services company

Overview

- Millions of merchant and cash access locations
- Global payments network with acceptance around the world.
- DFS has architected for high availability in a consolidated Oracle database environment hosting thousands of OLTP, batch, and warehouse application services

Oracle Solutions

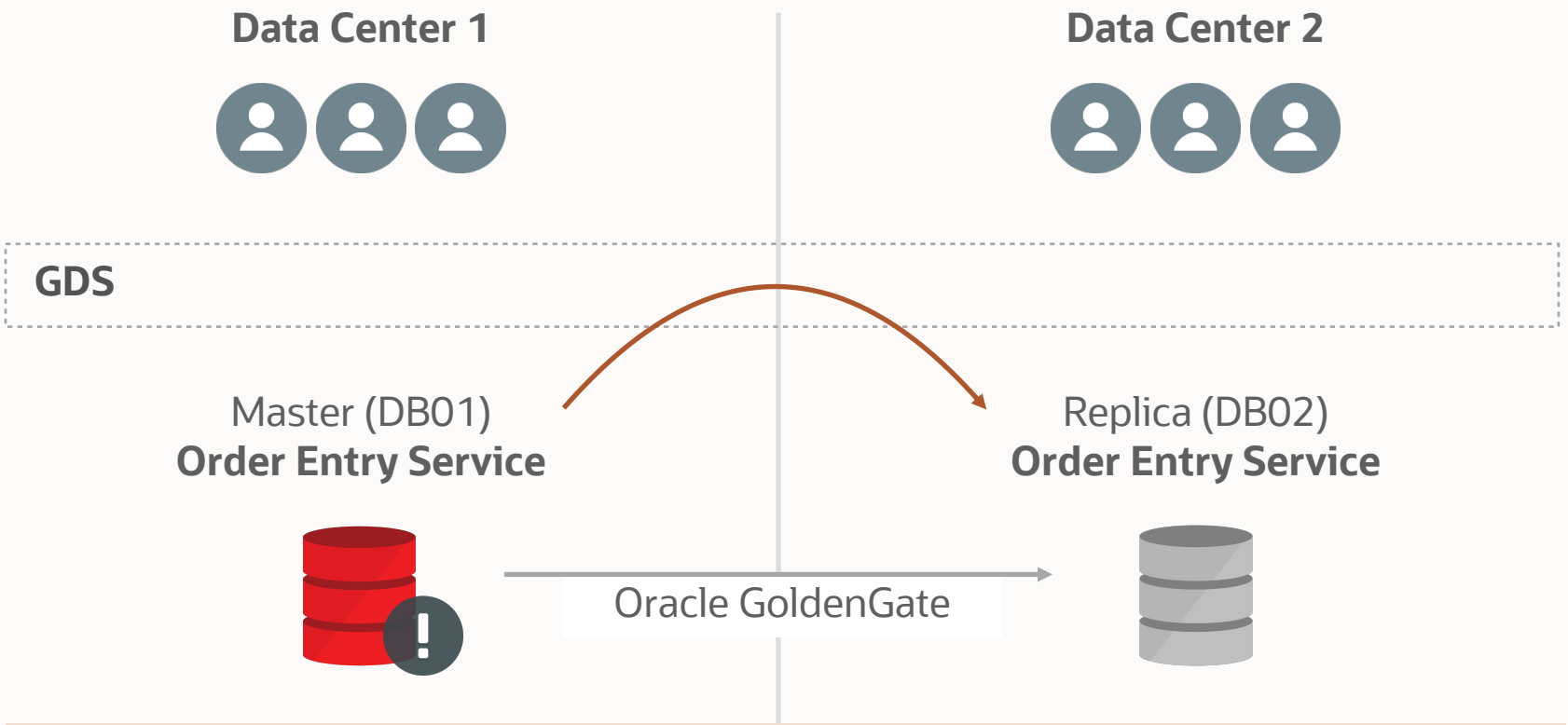
- Oracle Exadata
- Oracle Global Data Services
- Oracle GoldenGate
- Oracle Active DataGuard
- Oracle Recovery Manager
- Oracle Application Continuity
- Oracle I/O Resource Management

Environment Details

- ~3000 application services
- Across ~650 RAC databases
- Running on Exadata
- With Global Data Services with ~260 GDS Pools.
- Implemented Oracle's Maximum Availability Architecture
- Automation to reduce Mean Time to Recovery (MTTR) in their Oracle Database environment

Service Failover for Oracle GoldenGate Master-Replica

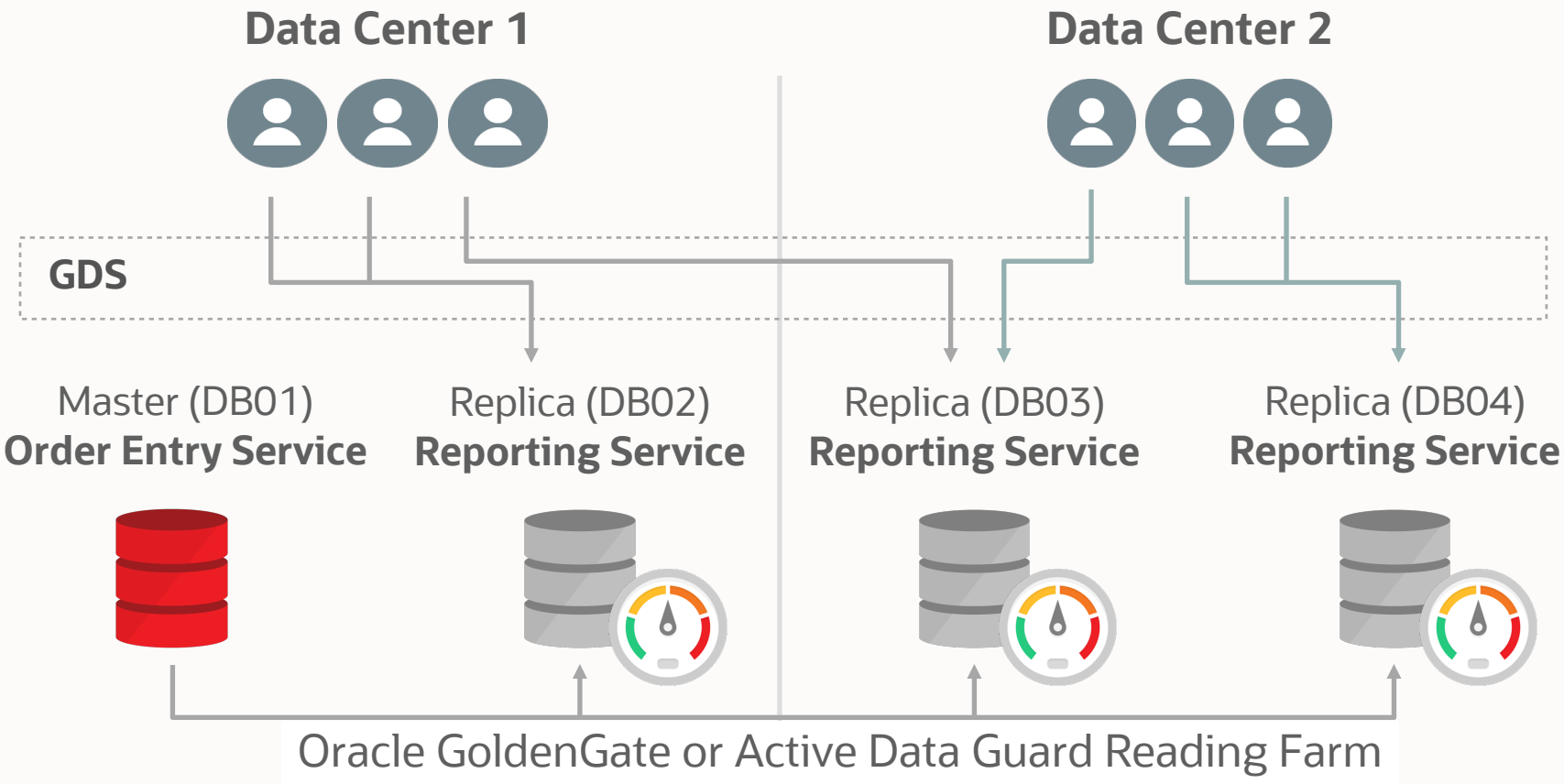
- Inter-database Service failover within and across regions
- Higher availability and improved manageability




```
GDSCTL>add service -service order_entry_service -gdspool sales
        -preferred DB01  -available DB02
```

Load Balancing for Reader Farms

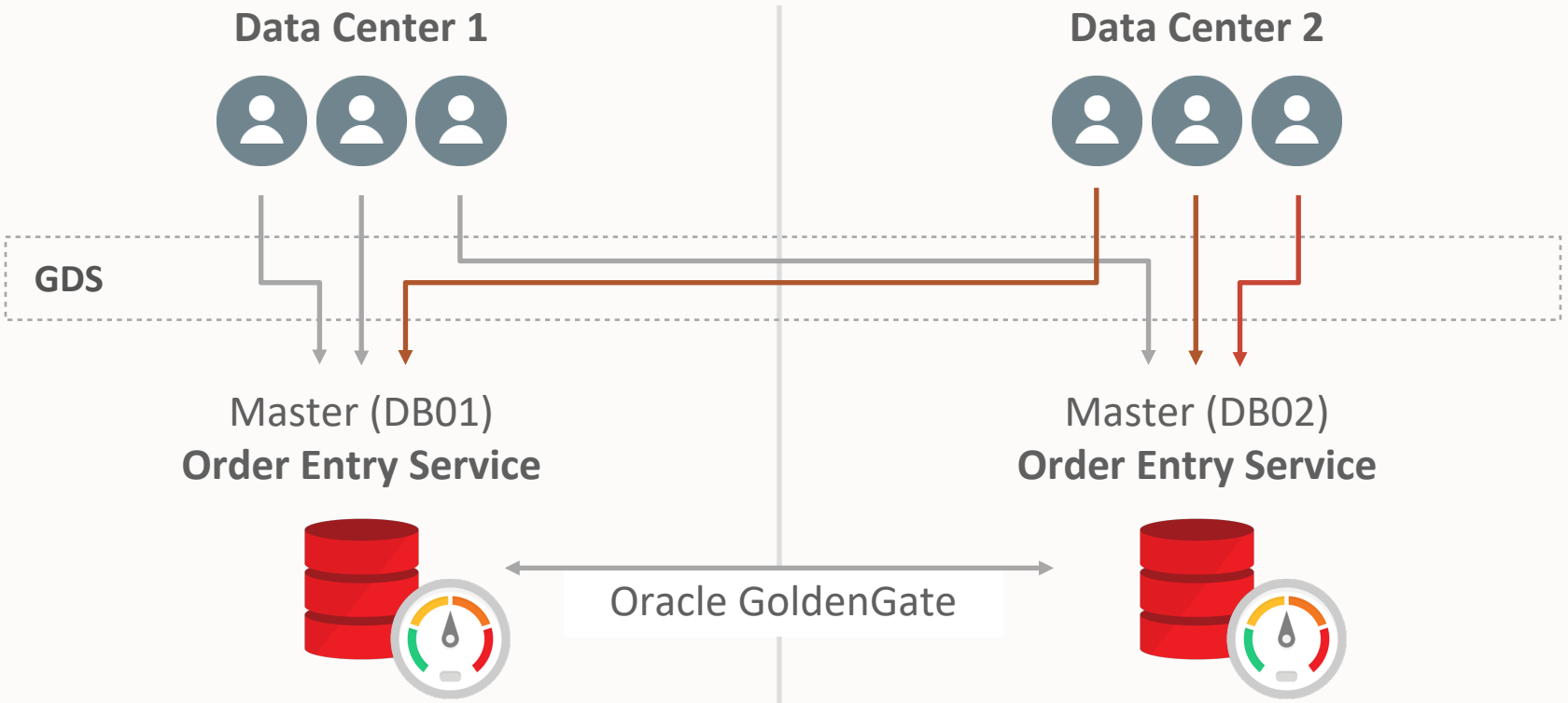
- With GDS, route Read Write workload to primary/master
- Balance Read Only workload on the reader farm
- Improved resource utilization and higher scalability for Read workloads




```
GDSCTL>add service -service reporting_srvc -gdspool sales
   -preferred_all -role PHYSICAL_STANDBY -clbgoal LONG -rlbgoal SERVICE_TIME
```

Load Balancing for Active/Active Oracle GoldenGate

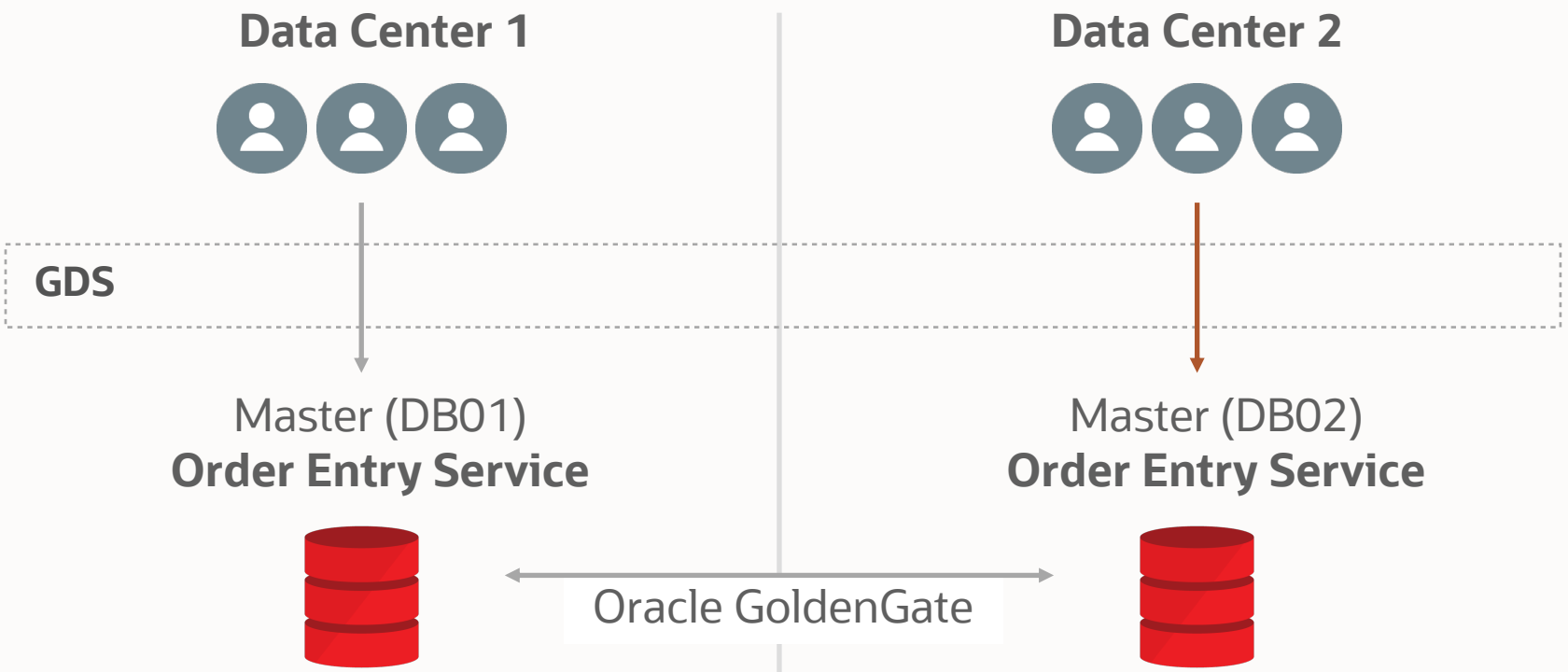
- Application handles multi-master conflict resolution
- GDS provides connect-time and run-time load balancing (within and across data centers) for all work requests




```
GDSCTL>add service -service order_entry_srvc -gdspool sales
       -preferred_all -clbgoal LONG
```

Region Affinity in Active/Active Oracle GoldenGate

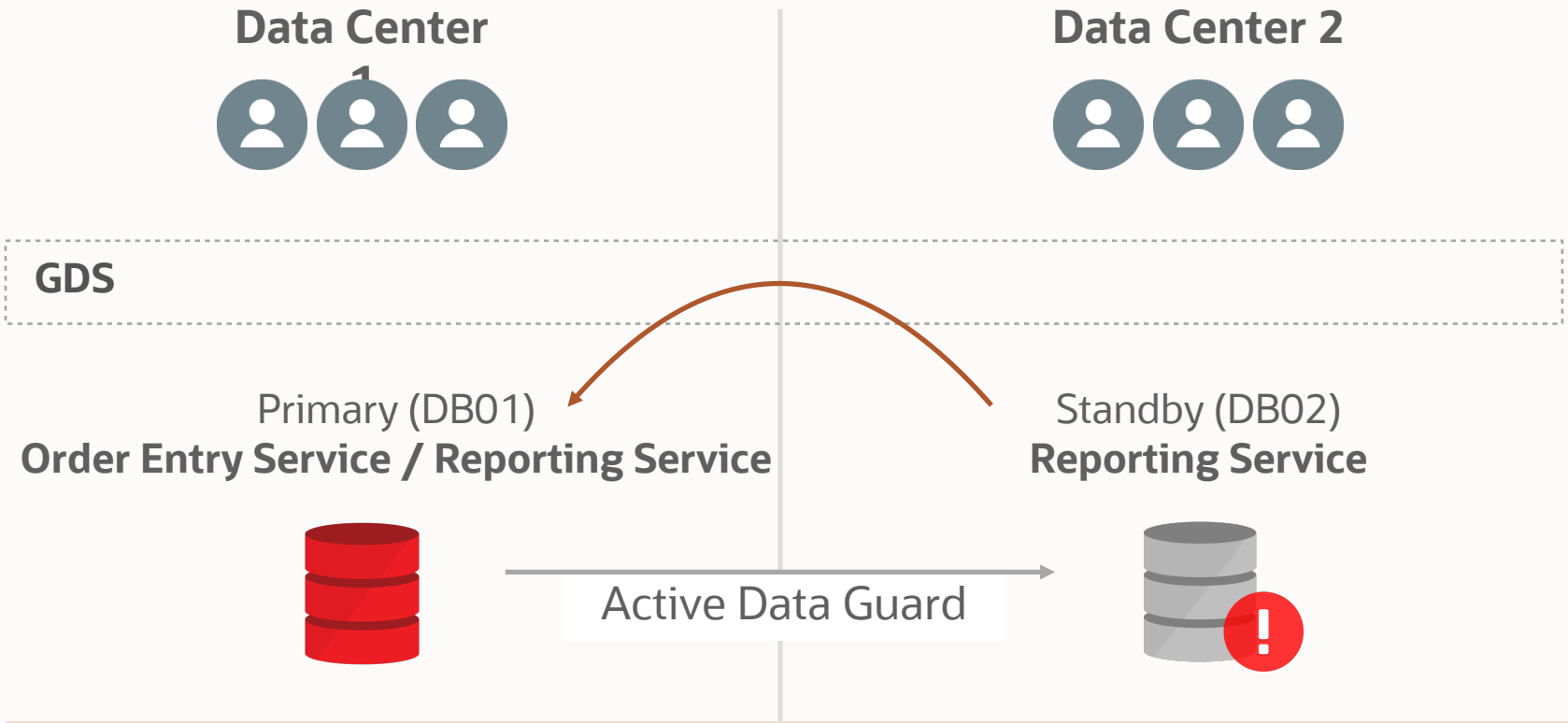
- Application handles multi-master conflict resolution
- GDS can route all workloads to nearest and best database in the client's region




```
GDSCTL>add service -service order_entry_service -gdspool sales
      -preferred_all -locality LOCAL_ONLY -region_failover
```

Service Failover for Active Data Guard

- Inter-database Service failover within and across regions
- Higher availability and improved manageability

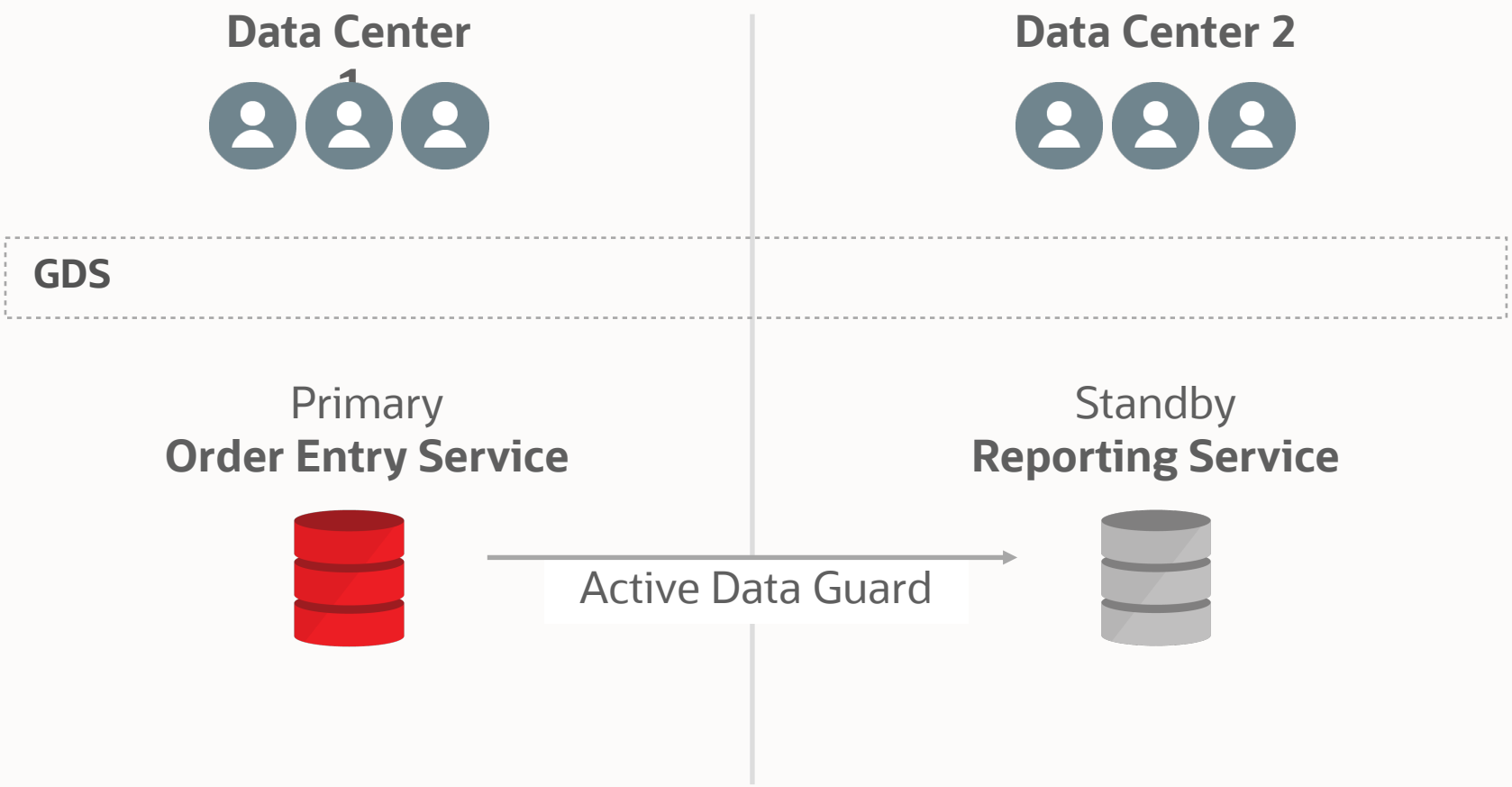



```
GDSCTL>add service -service reporting_service -gdspool sales
       -preferred_all -role PHYSICAL_STANDBY -failover_primary
```

Role based Global Services

For Active Data Guard

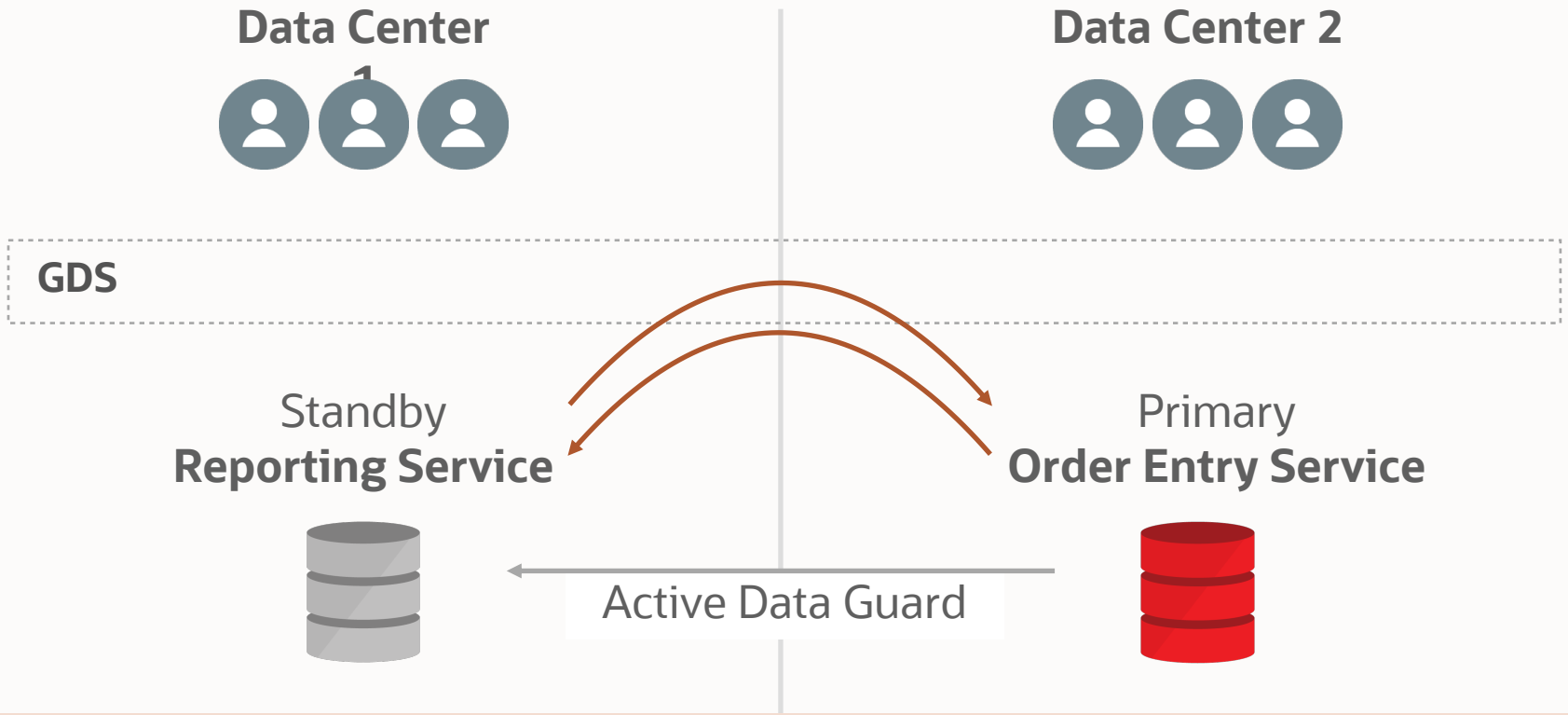
- Order Entry Service runs on Primary
- Reporting Service runs on Standby



Role based Global Services

For Active Data Guard

- Order Entry Service runs on Primary
- Reporting Service runs on Standby
- Upon Data Guard role change, GDS fails over services based on Role

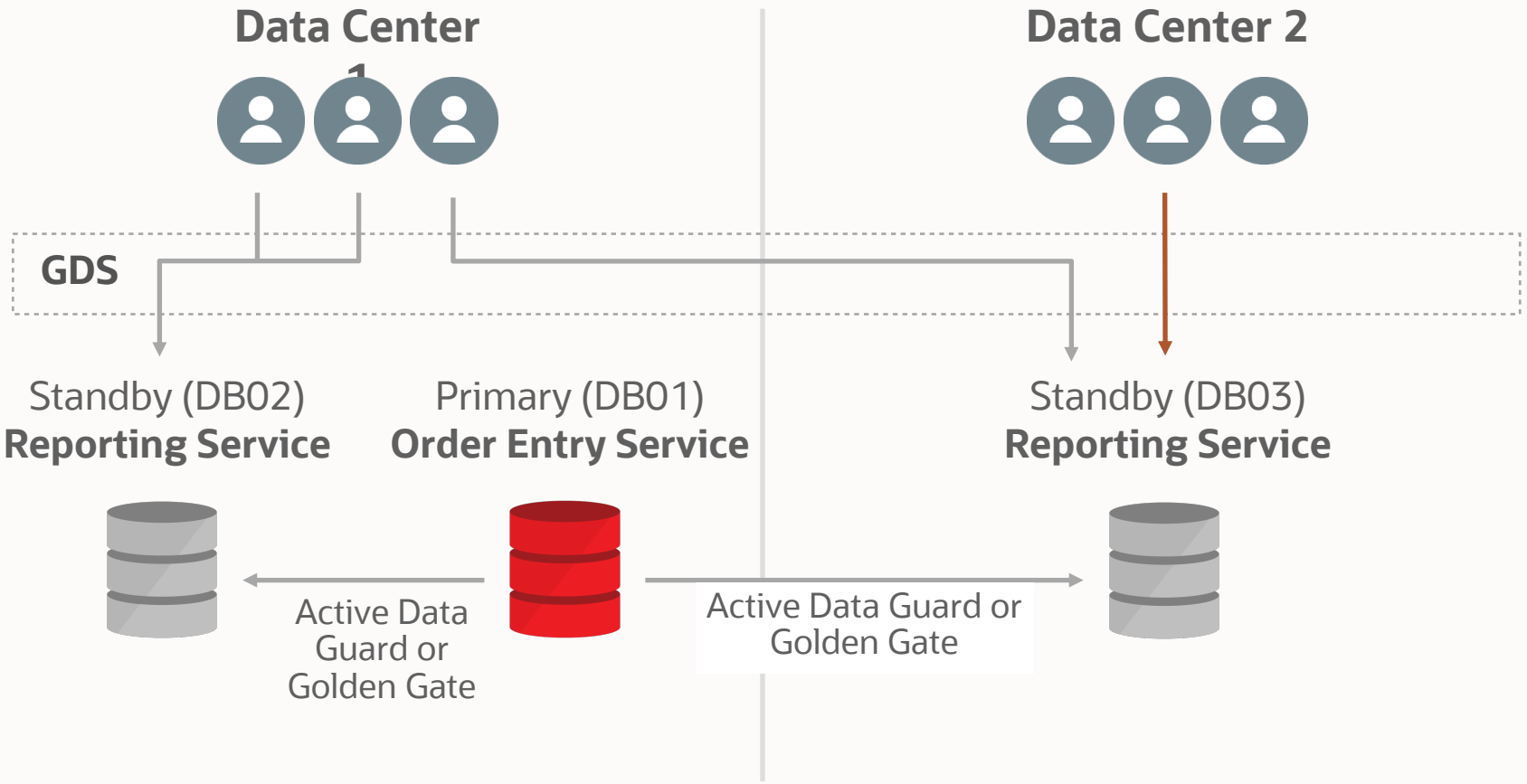



```
GDSCTL>add service -service order_entry_service -gdspool sales -preferred_all -role PRIMARY
GDSCTL>add service -service reporting_service -gdspool sales -preferred_all -role PHYSICAL_STANDBY
```

Routing based on Replication Lag Tolerance

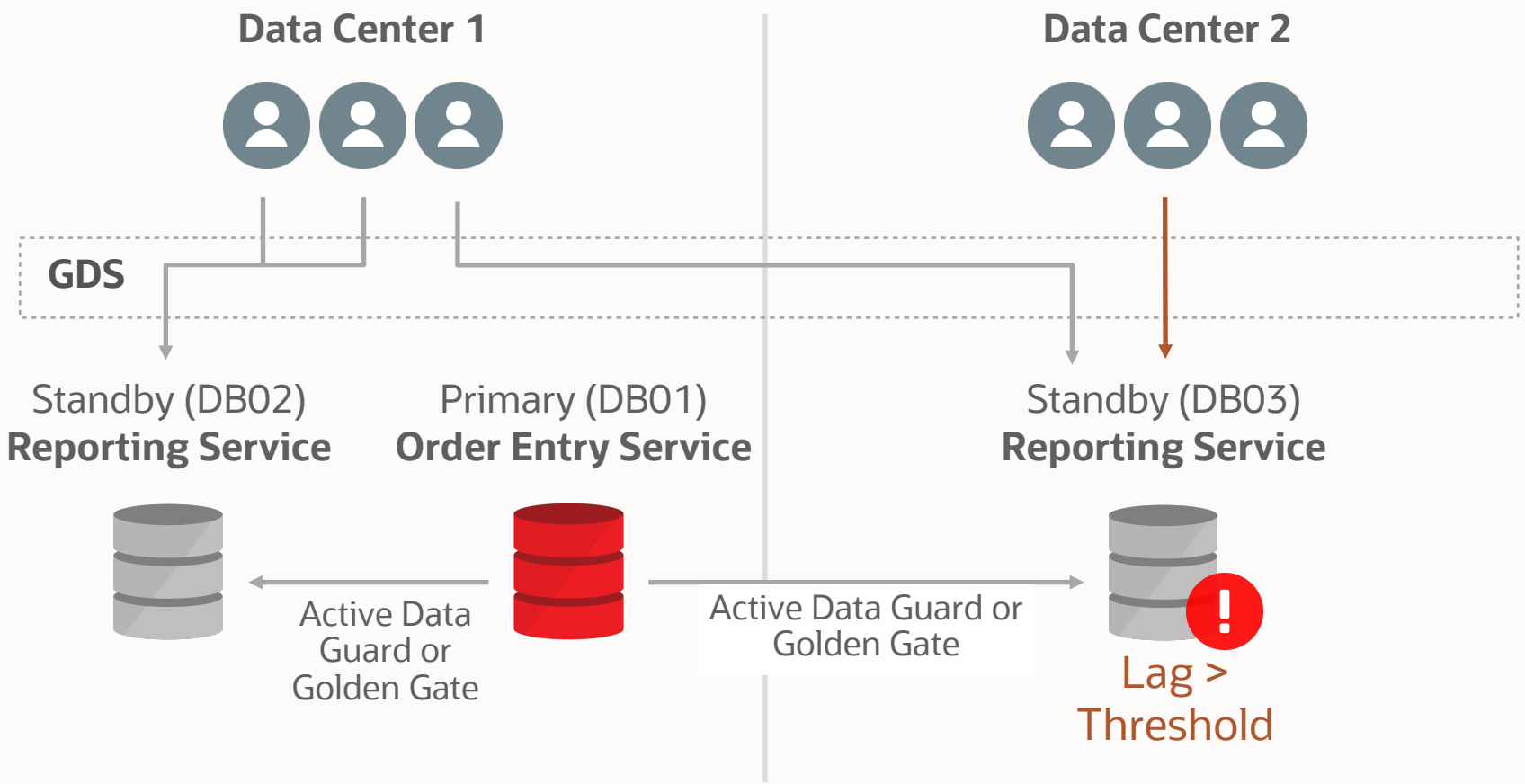
For Active Data Guard

- Specify replication lag limit for a service.
- GDS ensures that service runs on Active Data Guard standby(s) with lag less than this limit
- Improved data quality



Routing based on Replication Lag Tolerance
For Active Data Guard

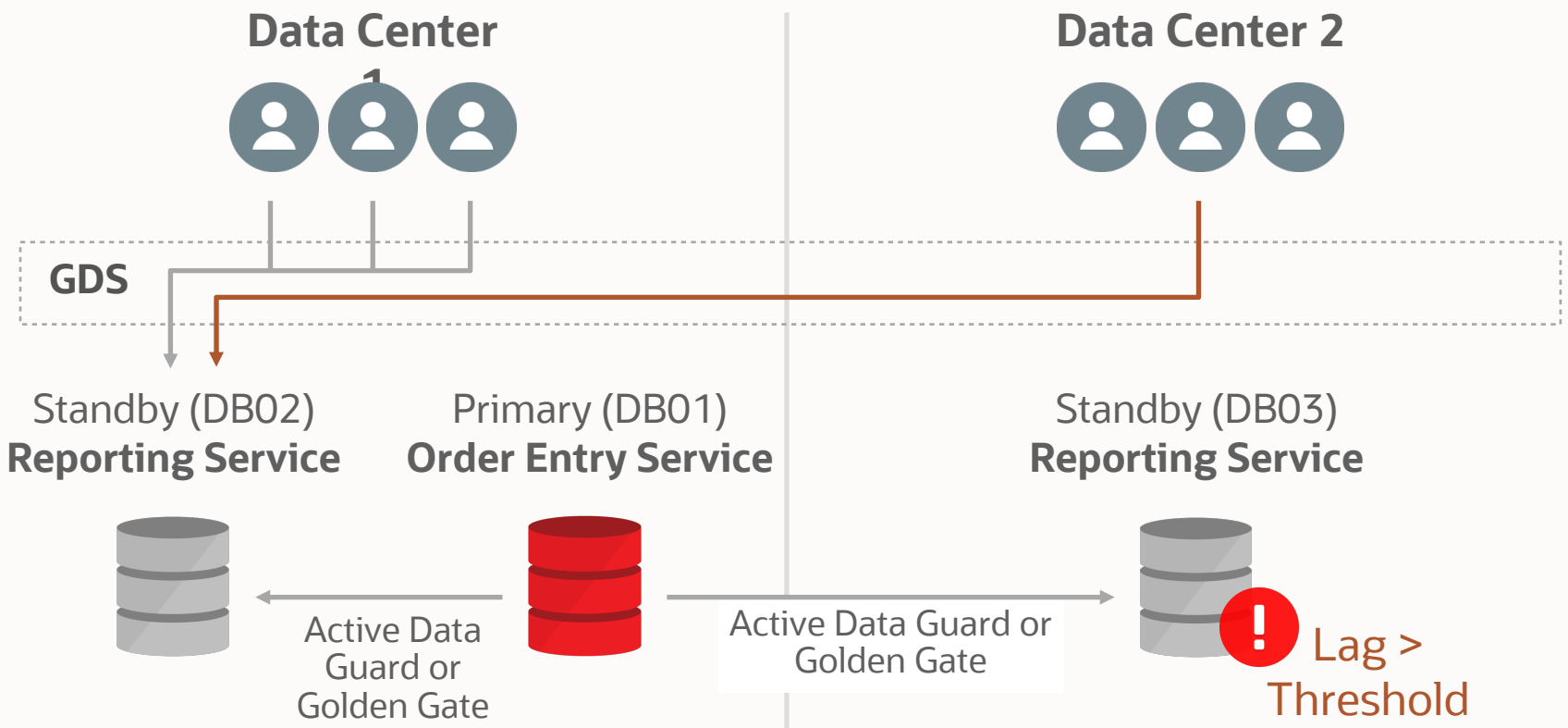
- Specify replication lag limit for a service.
- GDS ensures that service runs on Active Data Guard standby(s) with lag less than this limit
- Improved data quality



Routing based on Replication Lag Tolerance

For Active Data Guard

- Specify replication lag limit for a service.
- GDS ensures that service runs on Active Data Guard standby(s) with lag less than this limit
- Improved data quality




```
GDSCTL>add service -service reporting_srvc -gdspool sales
       -preferred_all -role PHYSICAL_STANDBY -lag 180
```

Mitigate Unplanned Outage with Oracle GDS

Automatic Failover of Client Workload to another Datacenter

Applications using …	**Oracle pools or drivers : UCP, ODP.NET, OCI, WebLogic Active GridLink** **3rd party App Servers using UCP: IBM WebSphere, Apache Tomcat**
Application config	• Subscribe to FAN events (published by GDS via ONS) by enabling Fast Connection Failover (FCF) • TNS entry to include RETRY_COUNT, CONNECT_TIMEOUT and TRANSPORT_CONNECT_TIMEOUT
Unplanned events	For Oracle GoldenGate and Active Data Guard: Global Service failovers For Active Data Guard: Data Guard role change
Sessions Drain	FAN posts unplanned downtime event and FCF ensues. • New work is redirected by GSM listeners immediately • Idle sessions are released immediately • Checked out connections receive invalid connection; Application closes the connection and gets new one from another database in the pool

Zero-downtime Planned Maintenance with Oracle GDS

Transparently move Client Workload to another Datacenter

Applications using …	**Oracle pools or drivers : UCP, ODP.NET, OCI, WebLogic Active GridLink** **3rd party App Servers using UCP: IBM WebSphere, Apache Tomcat**
Application config	• Subscribe to FAN events (published by GDS via ONS) by enabling Fast Connection Failover (FCF) • TNS entry to include RETRY_COUNT, CONNECT_TIMEOUT and TRANSPORT_CONNECT_TIMEOUT
DBA Step during planned event	GDSCTL> relocate service -gdspool sales -service sales_global_srvc -old_db ogg1 -new_db ogg2
Sessions Drain	FAN posts planned downtime event (REASON: USER) and FCF drains sessions as work completes • New work is redirected by GSM listeners immediately • Idle sessions are released immediately • Active sessions are released when returned to pools

Customer Case Studies

Large SaaS Provider for Clinical Trials

Cloud-based platform for clients to build their own clinical trials and perform medical research

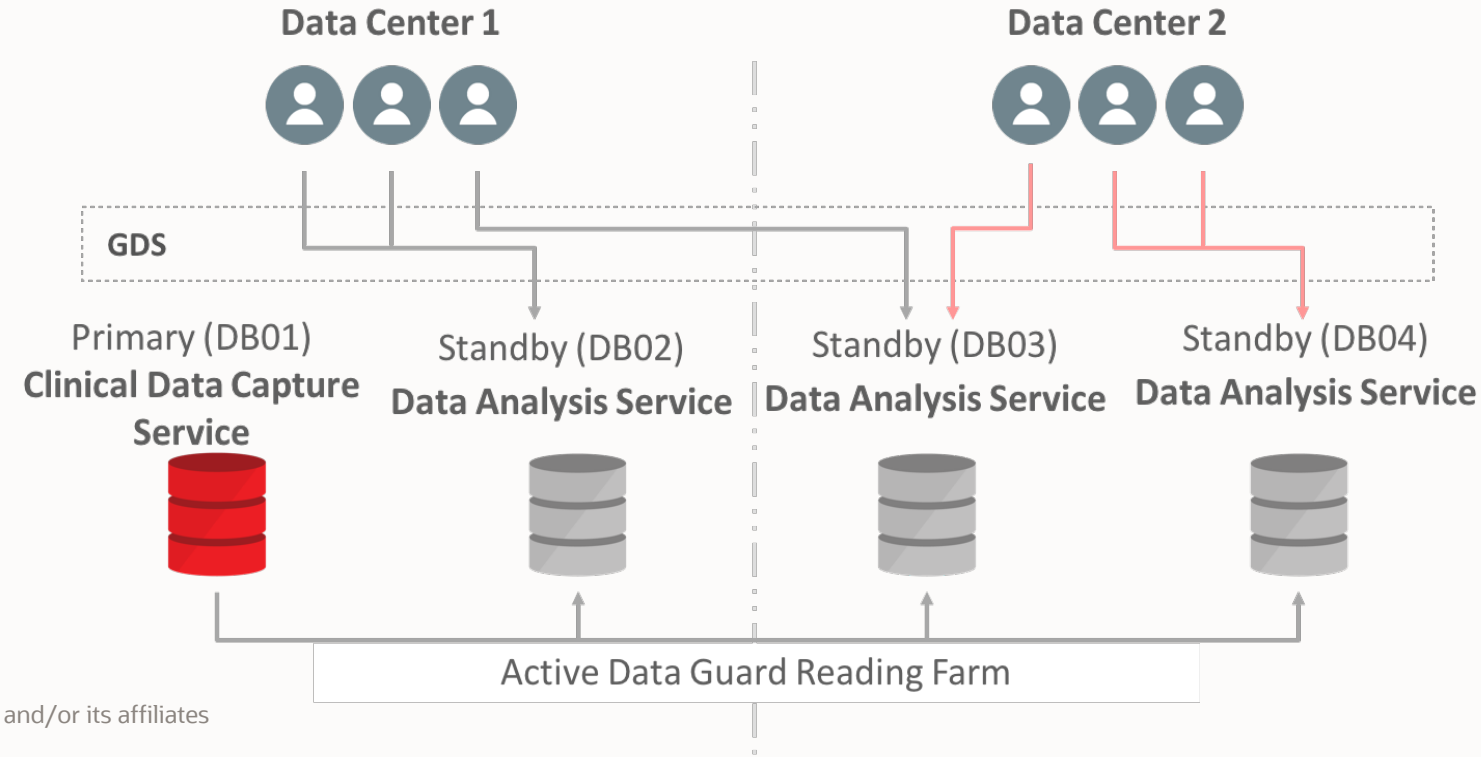
Goal: Scalable architecture for analysis of patient-related data for clinical operations

Global Load Balancing

Centralized Service Management

Challenges

- High scalability for data analysis
- Balance read-only workload on the reader farm



Results/ Benefits

- GDS allowed a scalable SaaS platform for clinical data analysis
- Zero licensing cost – as Oracle GDS is included with Active Data Guard

Global Hotel/Resorts Company Operating in 35 Countries

Mission Critical Application for Check-in and Points-exchange

Goal: Distributed infrastructure that provides continuous availability for unplanned and planned outages

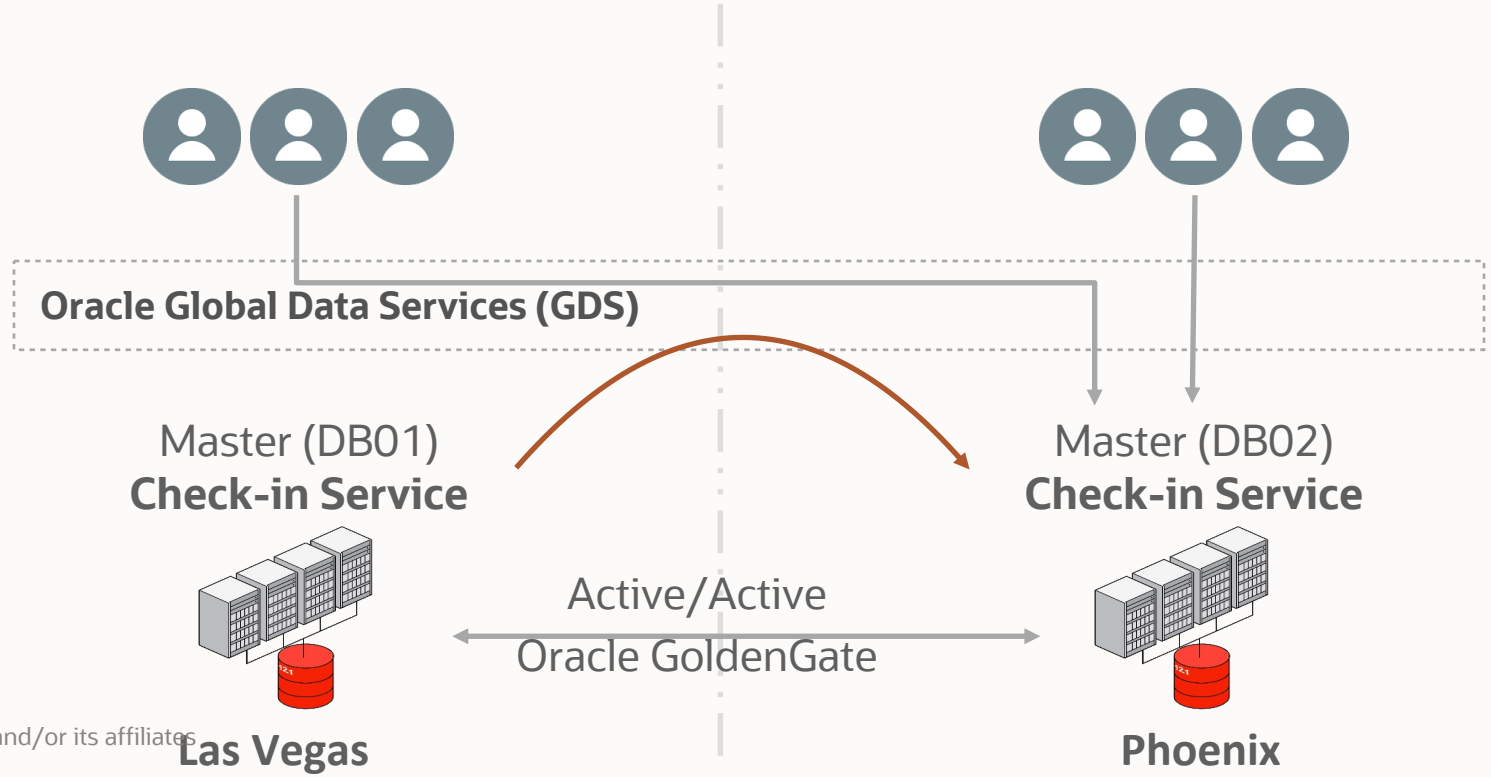
Global Service Failover

Centralized Service Management

Continuous Availability

Challenges

- Application downtime impacts customer check-in.
- Application needs to be patched numerous times a week (sometimes in a day).
- Takes couple of hours of downtime .



Results/ Benefits

- DBA applies the patch and fails over the global service to the patched environment .
- With Oracle GDS, application downtime has been eliminated.

Unspecified US Government Agency

Mission Sensitive Application

Goal: Achieve database high availability for all workloads running in a replicated environment

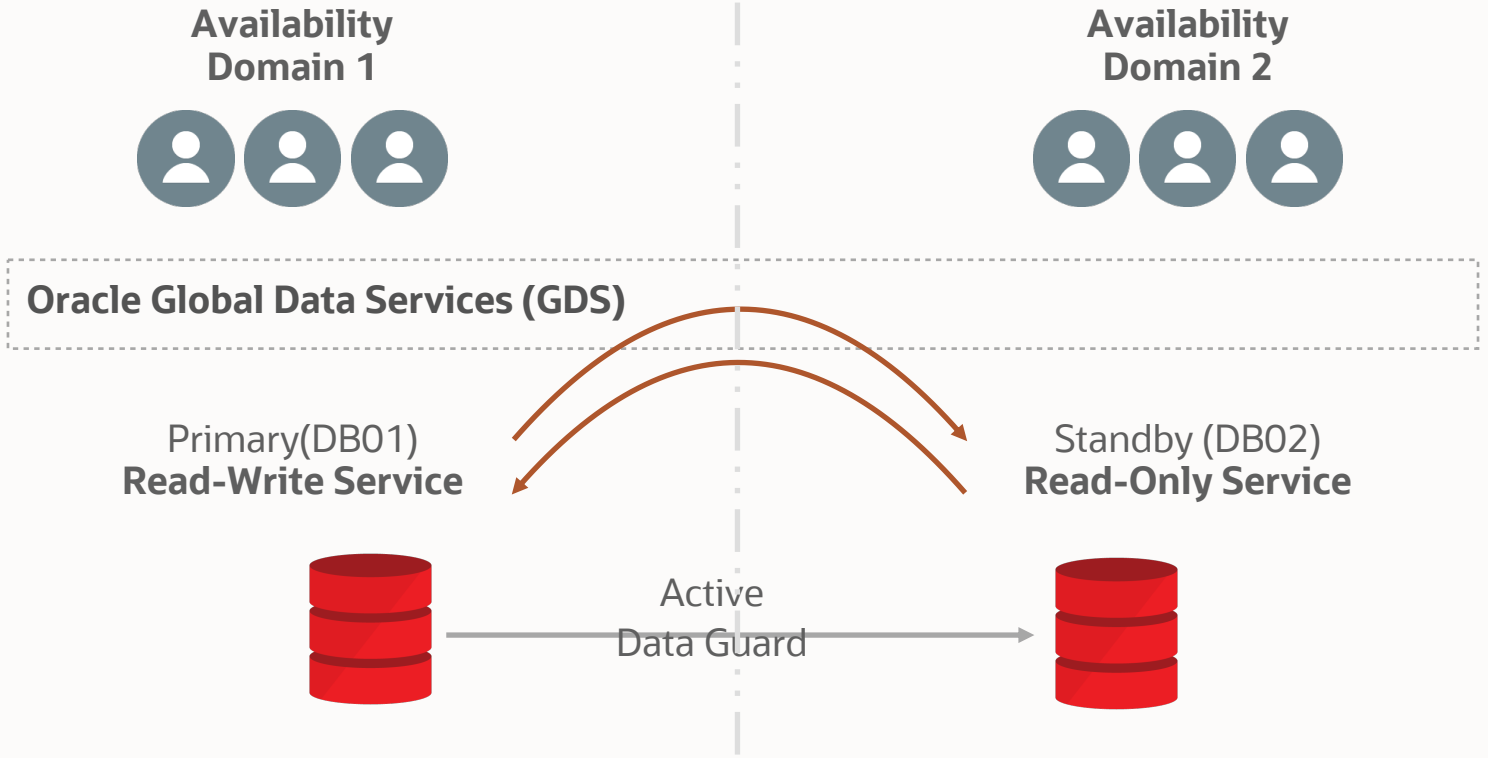
Automatic Global Service Failover

Downtime Reduction

Locality-based Routing

Challenges

- Database downtime impacts the application's availability
- Application requires high uptime and performance



Results/ Benefits

- Leveraged Role-based global services with Oracle GDS.
- Global services are automatically relocated upon role transition.
- Improved application performance via GDS locality based routing.

"Global Data Services will help MorphoTrak improve systems utilization by dynamically load balancing application queries between replicated databases across distributed data centers. We had already eliminated the cost of idle capacity by deploying Oracle RAC and Active Data Guard, and Oracle Database 12c takes us to another level. It replaces static load balancing between data centers with intelligent, real-time automation that efficiently utilizes all available capacity yielding greater ROI."

– Aris Prassinos, Chief Engineer, MorphoTrak, SAFRAN Group

Agenda

1 Workload management challenges of replicas
2 Introduction to Global Data Services (GDS)
3 GDS concepts and architecture
4 GDS use cases
5 Summary

Database Workload Management for Oracle Replicas

	Network Load Balancers	Oracle GDS
Locality based routing	✓	✓
Connect-time database load balancing	✓	✓
Publish routing and failover intelligence to clients	○	✓
Replication lag based database workload routing	○	✓
Inter-database global Service failover	○	✓
Automatic role based global Services	○	✓
Centralized management of database Services across replicas	○	✓
Native integration for Active Data Guard	○	✓
Cost Effectiveness	Additional $$$	Included with Active Data Guard or Oracle GoldenGate

Oracle Maximum Availability Architecture (MAA)

Production Site

RAC
- *Scalability*
- *Server HA*

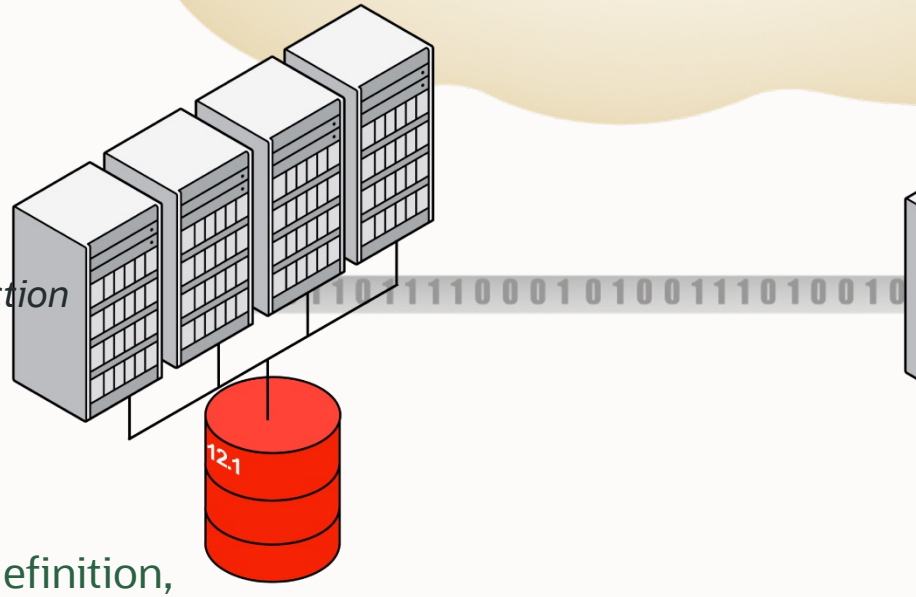
ASM
- *Local storage protection*

Flashback
- *Human error correction*

Global Data Services
- *Service Failover / Load Balancing*

Application Continuity
- *Application HA*

Enterprise Manager Cloud Control
- *Site Guard, Coordinated Site Failover*



Active Replica

Active Data Guard
- *Data Protection, DR*
- *Query Offload*

GoldenGate
- *Active-active replication*
- *Heterogeneous*

Edition-based Redefinition,
Online Redefinition, Data Guard, GoldenGate
- *Minimal downtime maintenance, upgrades, migrations*

Recovery Appliance,
RMAN, Oracle Secure Backup,
- *Backup to disk, tape or cloud*

Summary

- GDS provides workload routing, load balancing, service failover & management for replicated databases
- Key benefits
  - Applications use GDS to maximize performance and availability
  - Mitigate downtime during planned and unplanned outages
  - Better resource utilization of replicas

Additional Resources

https://www.oracle.com/goto/gds

https://www.oracle.com/goto/maa


ORACLE

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

Backup Slides

Global Service Attributes – Service Placement

Preferred

- Databases designated to provide the Global Service

Available

- Databases that provide Global Service if not enough *Preferred* databases are running
- If one of the preferred databases fails, then GSM maintains the cardinality of the Global service by starting the service on an *Available* database

Preferred_All

- All databases in a GDS Pool are preferred for the Global Service

Options for **Add Service :**

- {-preferred_all | -preferred *dbname_list [-available dbname_list] }*
- GDSCTL>add service -service *sales_qry_srvc* -gdspool sales -preferred sfo -available bos
- GDSCTL>add service -service *sales_reporting_srvc* -gdspool sales -preferred_all

Global Service Attributes – Service Placement (cont'd)

Role based Global Service

- Service should be active only when the database is either a primary or standby
- Can be started on a database if its role matches global service's role attribute
- Options for **Add Service :**
  - [-role {PRIMARY | PHYSICAL_STANDBY [-failover_primary] | LOGICAL_STANDBY | SNAPSHOT_STANDBY}]
- GDSCTL>add service -gdspool sales -service *sales_reporting_srvc* -preferred_all –role physical_standby

Lag Tolerance

- Establish application's tolerance for non-current data
- Specify the lag limit for the Global Service in seconds
- Options for **Add or Modify Service :**
  - -lag {lag_value | ANY}
- GDSCTL>add service -service *sales_reader_lag180_srvc* -gdspool sales -preferred_all -role physical_standby -lag 180

Global Service Attributes - Locality Based Routing

Achieve geographical affinity between clients and databases

Options for **Add or Modify Service**

- [-locality {ANYWHERE | LOCAL_ONLY [-region_failover]}]

Locality ANYWHERE

- Client connections and work requests are routed to any region for load balancing or failover
- GDSCTL>add service -service *sales_reader_srvc* -gdspool sales –preferred_all –locality ANYWHERE

Locality LOCAL_ONLY

- Regardless of load, GDS will not route to databases in other regions
- GDSCTL>add service -service *sales_reader_srvc* -gdspool sales –preferred_all –locality LOCAL_ONLY

Locality LOCAL_ONLY –region_failover

- Client connections and work requests are routed to another region when all databases in a region have failed
- GDSCTL>add service -service *sales_reader_srvc* -gdspool sales –preferred_all –locality LOCAL_ONLY –region_failover

Global Service Attributes – Load Balancing

Connect-time Load Balancing (CLB)

GDS support CLB for all clients

- Directs connection requests to the best database instance in GDS pool
- Takes into account
  - Load statistics from all GDS pool databases
  - Inter-region network latency, locality and CLB goal
- Options for Add Service :
  - [-clbgoal {SHORT | LONG}]
  - GDSCTL>add service -service sales_clb_srvc -gdspool sales –preferred_all -clbgoal LONG

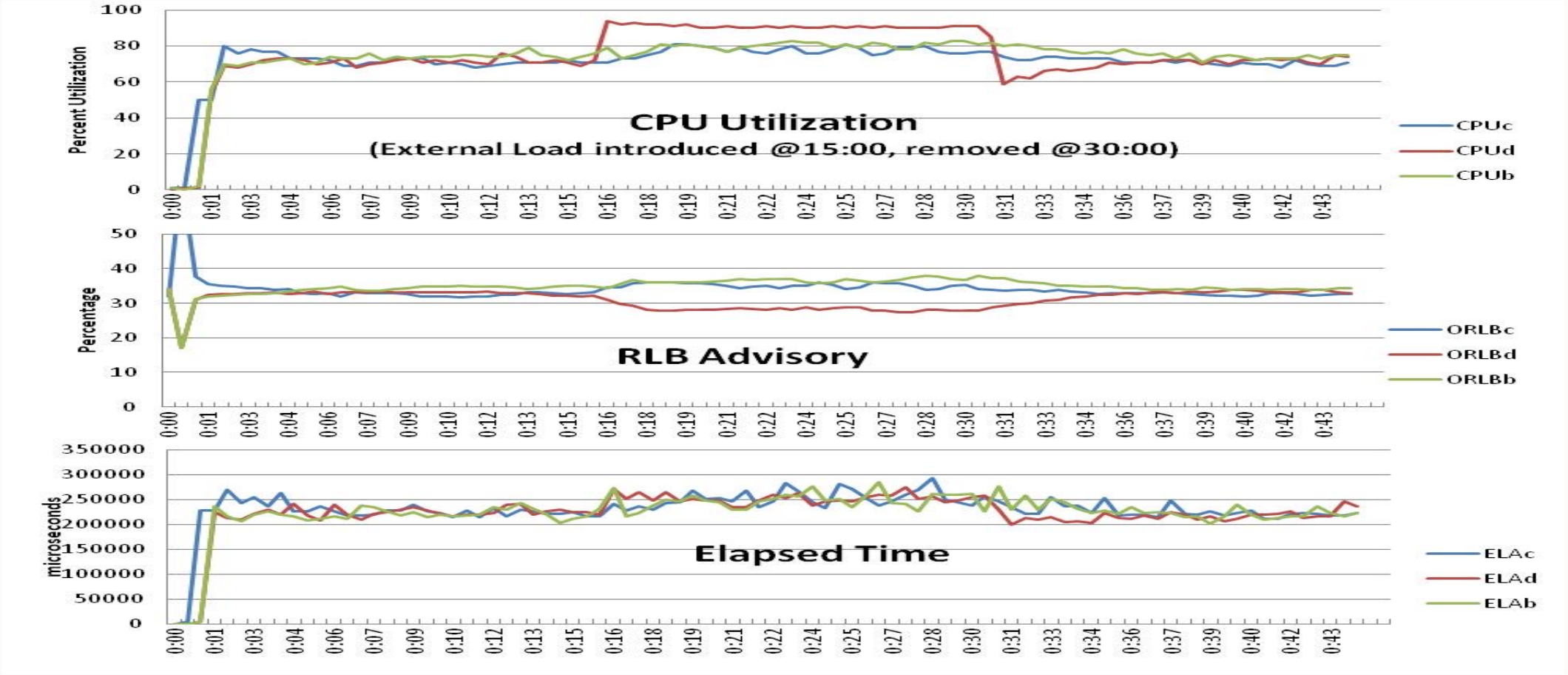
Run-time Load Balancing (RLB)

GDS supports RLB feature of connection pools for OCI, JDBC/UCP, ODP.NET, WLS

- Publishes RLB Advisory to clients
- Based on advisory, clients distribute workload requests across persistent connections spanning GDS Pool database instances
- Takes into account
  - Per-service performance data from pool databases
  - Inter-region network latency, locality and RLB goal
- Options for Add Service :
  - [-rlbgoal {SERVICE_TIME | THROUGHPUT}]
  - GDSCTL>add service -service sales_rlb_srvc -gdspool sales –preferred_all –rlbgoal SERVICE_TIME

Run-time Load Balancing with GDS

Standalone Identical Database Servers with External Load

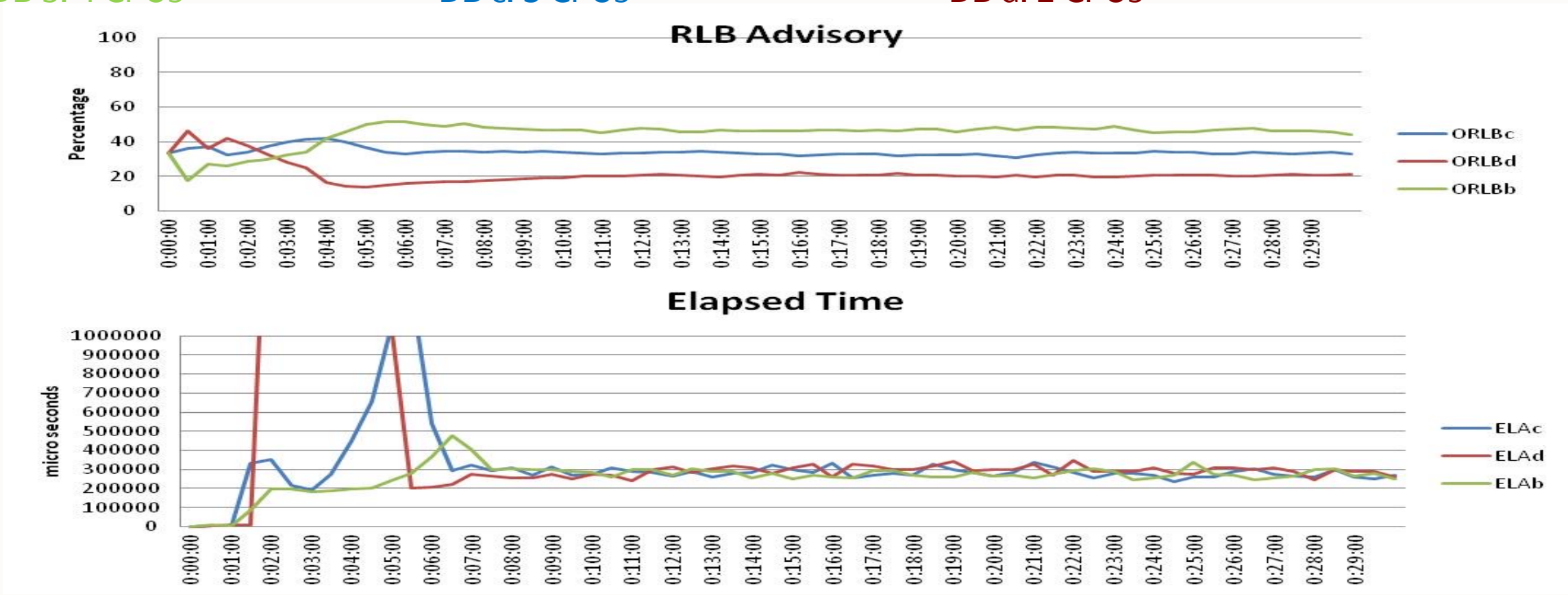


Routing responds gracefully to changing system conditions

Run-time Load Balancing with GDS

Standalone Asymmetrical Database Servers

DB b: 4 CPUs DB c: 3 CPUs DB d: 2 CPUs



GDS does intelligent load balancing even across asymmetrical database servers