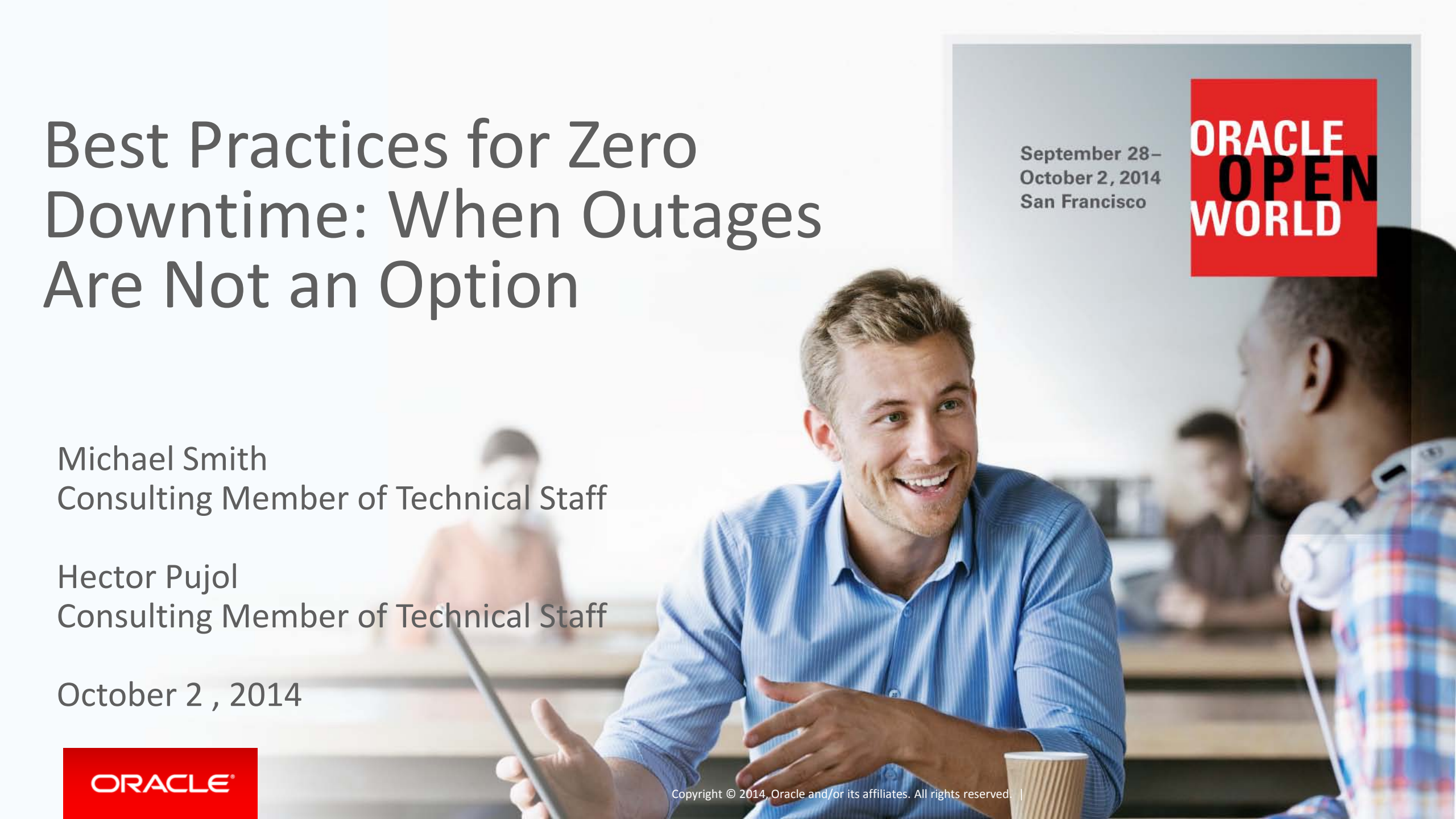

ORACLE®

Best Practices for Zero Downtime: When Outages Are Not an Option



Michael Smith
Consulting Member of Technical Staff

Hector Pujol
Consulting Member of Technical Staff

October 2 , 2014

ORACLE

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.

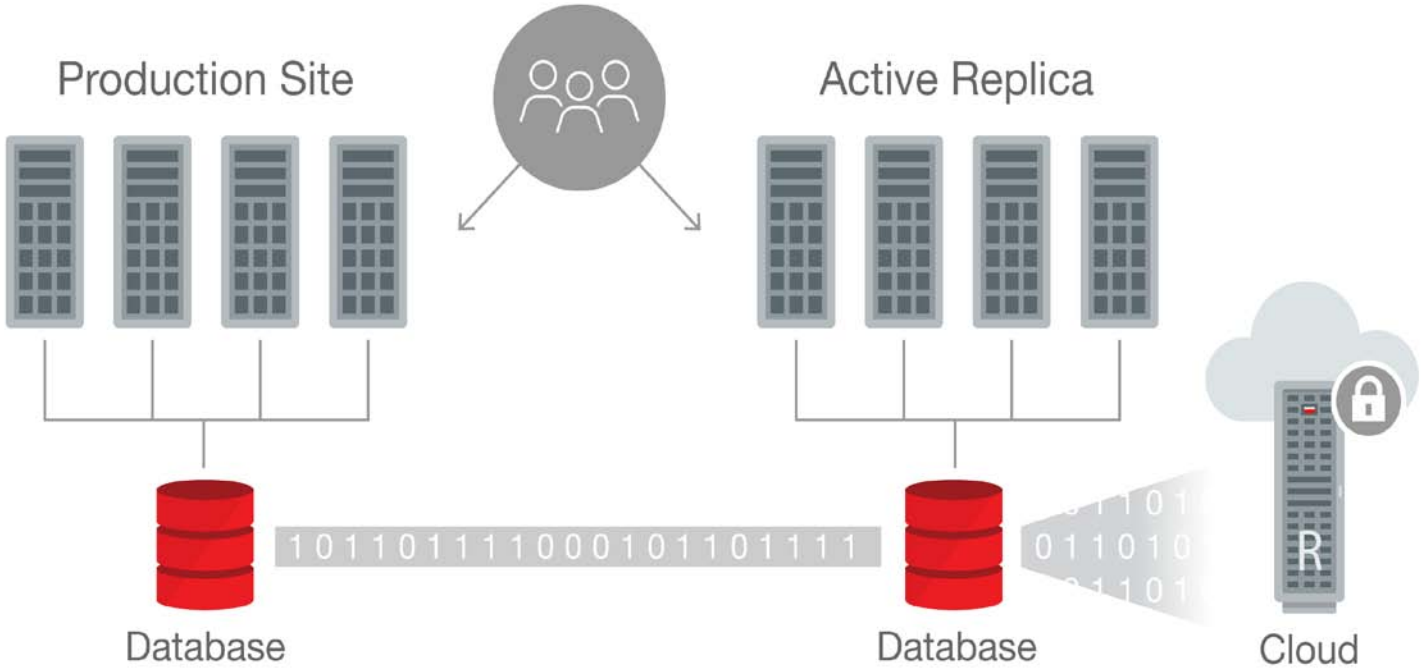
Program Agenda

1. Defining Platinum Level Service
2. Application Continuity
3. Application Performance During Outages

Defining Platinum Level Service

Maximum Availability Architecture (MAA)

- Oracle-optimized architecture
- Extensive HA best practices
  - Oracle Database
  - Oracle Fusion Middleware
  - Oracle Applications
  - Cloud Control
  - Oracle Engineered Systems
  - Partner solutions



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

Oracle MAA Availability Tiers

Availability Service Levels for Unplanned and Planned Outages

PLATINUM
- Zero Outage for Platinum Ready Applications
- Zero data loss

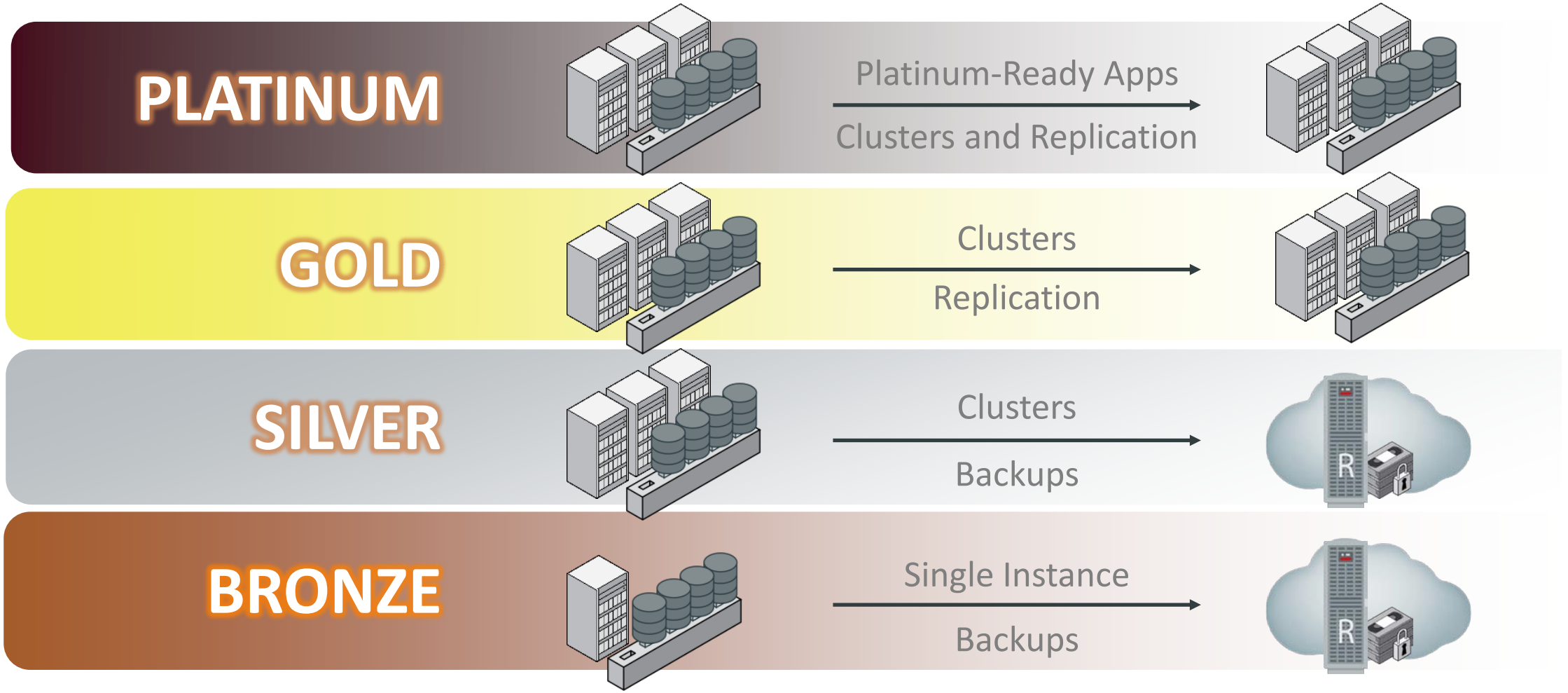
GOLD
- Comprehensive HA and Disaster Protection
- Recovery in seconds with zero or near-zero data loss

SILVER
- High Availability (HA) for Recoverable Local Outages
- Backups plus redo for Oracle data protection

BRONZE
- Basic Service Restart
- Backups plus redo for Oracle data protection

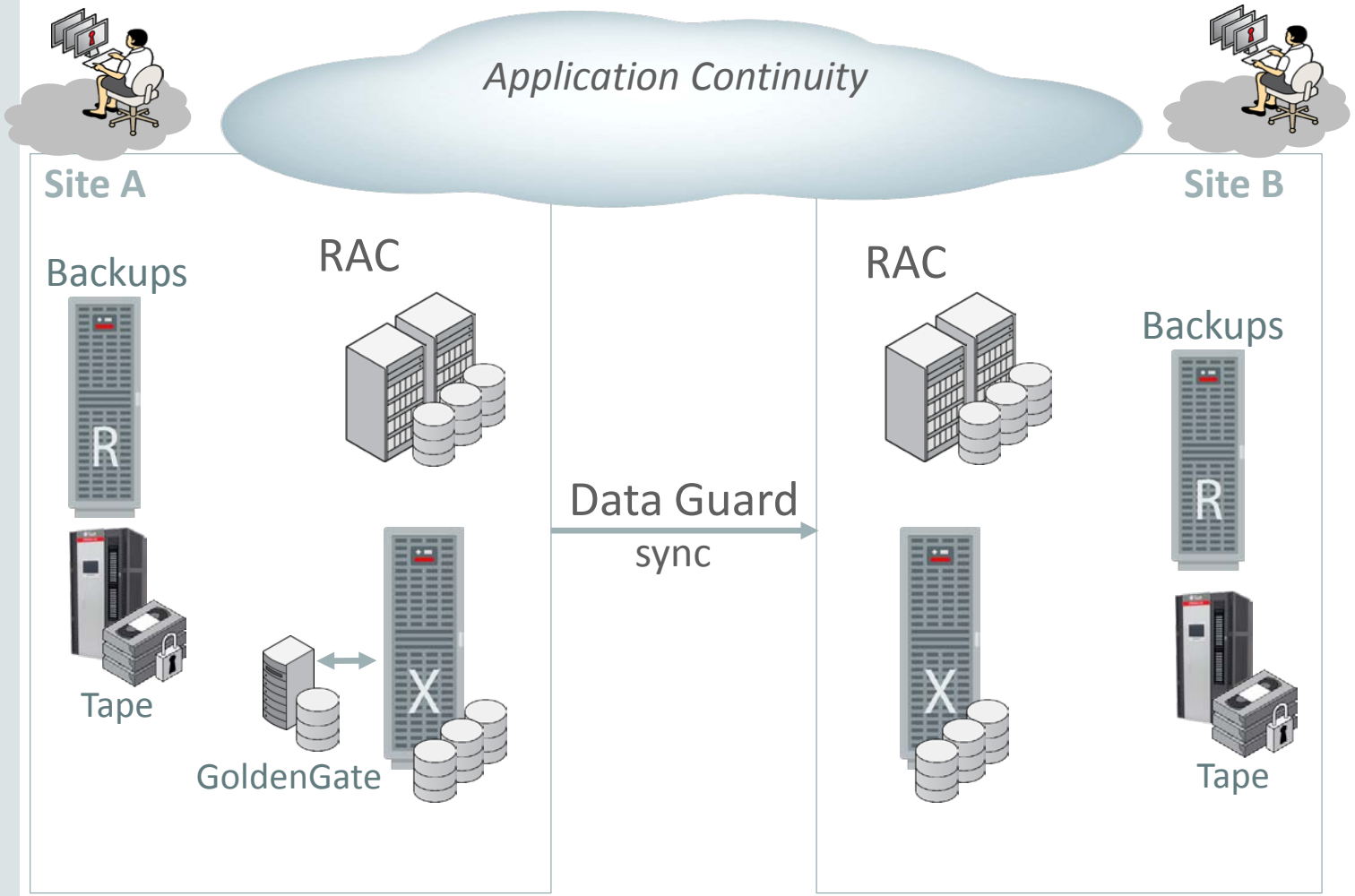
Oracle MAA Availability Tiers

Reference Architectures



Platinum Tier

Zero Application Outage for Platinum Ready Applications



- Outages masked from applications, in-flight transactions preserved
  - Application Continuity
- Zero data loss failover, LAN or WAN
  - Active Data Guard / Far Sync
- Bi-directional replication and zero downtime maintenance
  - Oracle GoldenGate
- Online patching for applications
  - Edition-based Redefinition
- Automated workload management
  - Oracle Global Data Services

Zero Outage for Platinum Ready Applications

Platinum-Level High Availability and Disaster Recovery

	Events	Downtime	Data Loss Potential
Unplanned Outages	Database instance failure	Zero application outage	Zero
	Recoverable server failure	Zero application outage	Zero
	Data corruption, database unable to restart, site failure	Zero application outage	Zero
Planned Maintenance	Online file move, reorganization/redefinition, patching	Zero application outage	Zero
	Hardware or operating system maintenance and database patches that cannot be done online but are qualified for RAC rolling install	Zero application outage	Zero
	Database upgrades: patch sets, full database releases	Zero application outage	Zero
	Platform migrations	Zero application outage	Zero
	Application upgrades that modify database objects	Zero application outage	Zero

Application Continuity

Application Continuity

Masking Outages from Your Application

- FAN enabled fast outage notification and recovery
- Application continuity now *masks* outages from applications
- Outage scenarios are tricky for applications to handle
- Available in 12c with Standalone JDBC and packaged with:
  - Oracle Universal Connection Pool (UCP)
  - Oracle WebLogic Server 12c
  - Oracle JDBC-based third party application servers

Application Continuity

Masking Outages from Your Application

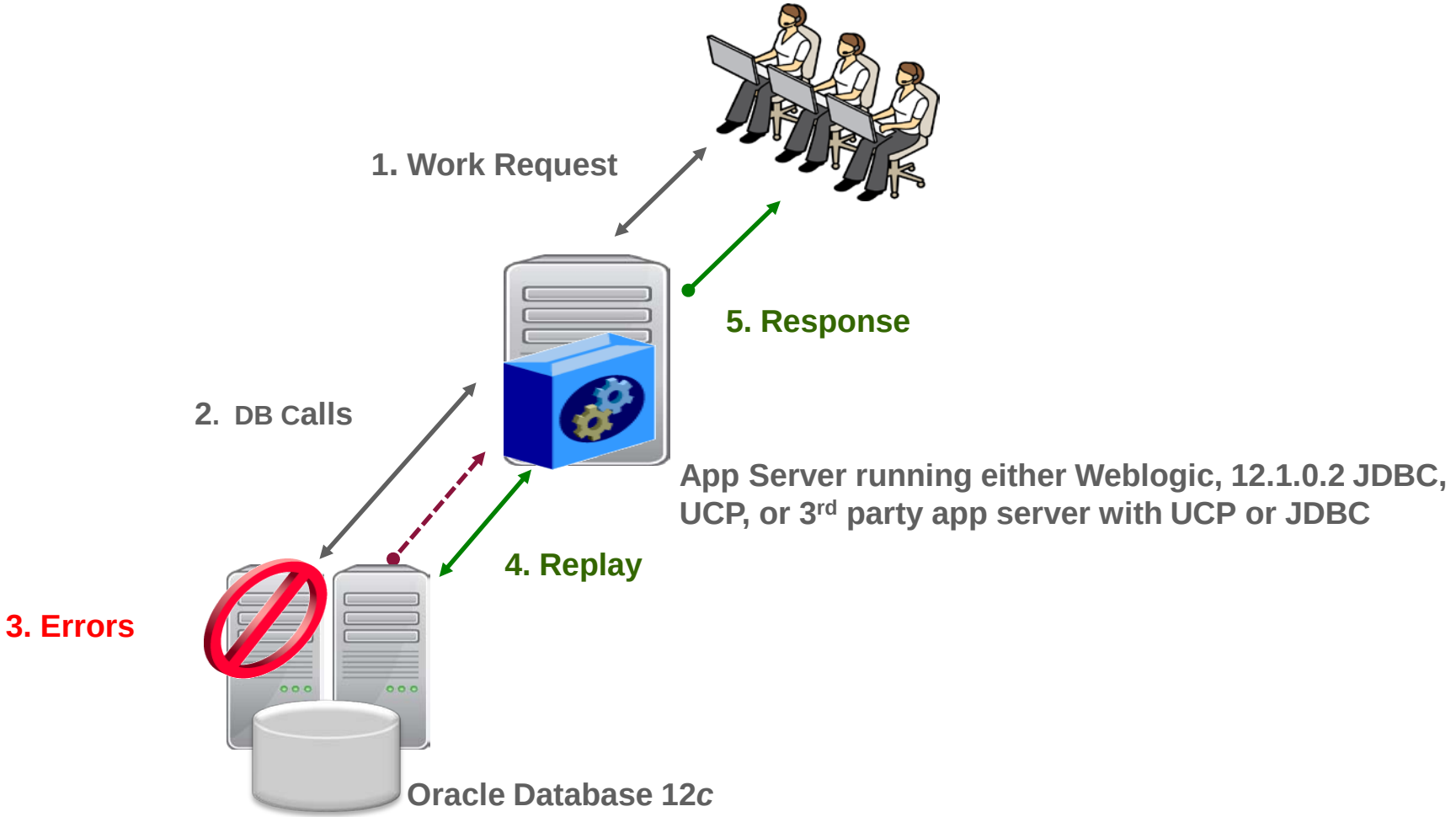
- Application Continuity masks these outages:
  - Process or service failure
  - Instance failure
  - Node failure
  - Database failure
- The application will not see errors
  - It may see a response time delay depending on the type of failure

Application Continuity
How it works

- Key Components (Challenges) of Application Continuity
  - Services, FAN, load balancing, and FCF for the connection pool
  - Recoverable Errors
  - Commit Outcome using Transaction Guard
  - Request boundaries
  - Verification of expected database call results
  - Session consistency and integrity checks
  - Safe restoration of mutable values at replay

Application Continuity

How it works



Application Continuity
Assess your Application

- Assess your application for:
  - Request boundaries
    - Automatic with WebLogic, UCP, 3rd Party app servers using UCP, or Oracle JDBC 12.1.0.2 with PooledConnection interface
  - Eliminate deprecated JDBC concrete classes
  - Decide to disable calls that should NOT be replayed
    - UTL_HTTP, use of external wall clock time, or other external dependencies
  - Decide to allow restore of mutable functions
    - SYSDATE, SYSTIMESTAMP, SYS_GUID, sequence.NEXTVAL
  - Session state changes requiring callbacks

Application Continuity
Configuration Steps

- Configure UCP to use the replay driver and set pool settings
- Configure Java Net Connections
- Configure Service Attributes
- Grant Mutables
- Check resources for client and server

Application Continuity

Configure JDBC Replay Driver and Pool Settings

- Replay Driver
  - UCP Example of using the replay driver:

```
try {
      pds = PoolDataSourceFactory.getPoolDataSource();
      pds.setConnectionFactoryClassName("oracle.jdbc.replay.OracleDataSourceImpl");
…
```

- Pool Settings
  - Ensure the pool doesn't close down an inactive connection too soon

```
pds.setONSConfiguration("nodes=maa05:6200,maa06:6200,maa07:6200,maa08:6200")
pds.setFastConnectionFailoverEnabled(true);
pds.setInactiveConnectionTimeout(86400);
pds.setConnectionWaitTimeout(86400);
pds.setTimeoutCheckInterval(3600);
pds.setAbandonedConnectionTimeout(86400);
```

Application Continuity

Configure Client Connections

New

Retry while service is unavailable

Safe for logon storms

```
url = "jdbc:oracle:thin:@(DESCRIPTION =
    (CONNECT_TIMEOUT=90)  (RETRY_COUNT=30)(RETRY_DELAY=10)
    (ADDRESS_LIST =
    (LOAD_BALANCE=on)
    (ADDRESS = (PROTOCOL = TCP)(HOST=austin-scan)(PORT=1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST=houston-scan)(PORT=1521)))
    (CONNECT_DATA=(SERVICE_NAME = oltpworkload)))"
```

Balance IP's in Scan

Application Continuity

Configure Service Attributes

- Service attributes
  - **FAILOVERTYPE** : Set this to TRANSACTION to enable Application Continuity
  - **COMMIT_OUTCOME** : Set this to TRUE to enable Transaction Guard (mandatory)
  - **REPLAY_INIT_TIME** : Duration after which replay is not started
  - **FAILOVERRETRY** : Number of connection retries for each replay attempt.
  - **FAILOVERDELAY** : Delay in seconds between connection retries

Configuring for Outages - Services

- Create role based services on primary and standby clusters
  - Ensures services start on the correct database
  - Efficient routing of work of different workloads
- Services should be created with Application Continuity and standby attributes previously mentioned

```
 srvctl add service -db <db_unique_name> -service <service_name>
[-role [PRIMARY][,PHYSICAL_STANDBY][,LOGICAL_STANDBY][,SNAPSHOT_STANDBY]]
- failovertype TRANSACTION –commit_outcome TRUE

srvctl add service -db EMEA -service PLATINUM –role PRIMARY -failovertype
TRANSACTION -commit_outcome TRUE -replay_init_time 600 -failoverretry 30
-failoverdelay 10
```

Application Continuity
Configure Approved Mutables

- Decide to allow mutables for delayed execution
  - You must understand how mutables are used before allowed to replay them
  - The ALTER SEQUENCE and GRANTs tell Application Continuity what to do
- Mutable Syntax:

```
ALTER SEQUENCE.. [sequence object] [KEEP|NOKEEP];
GRANT KEEP SEQUENCE <SEQUENCE NAME> on sequence object [to USER] ;
```

- Needed for users that don't own the sequence

```
GRANT [KEEP DATE TIME | KEEP SYSGUID].. [to USER]
```

Application Continuity
Check Resources for Client and Server

- Middle Tier
  - Replay driver may use more memory than base driver
  - Amount of additional usage is negligibly more if number of calls/request are small
  - Allocate sufficient memory to the JVM (e.g. Xms4096m for 4 GB) to minimize garbage collection impacts
  - Slight additional CPU usage for building proxy objects, etc
- Server Tier
  - Some additional CPU usage for managing the validation

Application Continuity
Additional Tips

- UCP Coding Best Practices
  - Get connections from the pool and CLOSE them after the request
    - Request boundaries automatically set between *getConnection()* and *close()*
    - Keep request boundaries sensible for efficiency and timely planned maintenance operations
  - Close connections as part of a *finally{}* block
    - Test connections to ensure they aren't already closed nor null before closing them
    - Test connections with isValid() before rolling them back

Application Continuity

Additional Tips on Killing Sessions

DBA Command	Replays
srvctl stop service -db orcl -instance orcl2 -force	**YES**
srvctl stop service -db orcl -node rws3 -force	**YES**
srvctl stop service -db orcl -instance orcl2 **–noreplay** -force	
srvctl stop service -db orcl -node rws3 **–noreplay** -force	
alter system kill session … immediate	**YES**
alter system kill session … **noreplay**	
dbms_service.disconnect_session([service], dbms_service. **noreplay**)	

Application Performance During Outages

Defining Outages

- Planned
  - Software updates
  - Hardware changes
- Unplanned
  - Application outages
  - Partial site / Full site

Planned Maintenance

- RAC rolling patch apply:
  - Drain resources from instance to be patched
  - Bring down instance
  - Apply patch
  - Restart instance and repeat on remaining instances
- Which patches are eligible for RAC rolling?
  - 96% of one off patches
  - All PSU/CPU
  - Nearly all bundle patches

Planned Maintenance

- Data Guard Rolling upgrades for patchsets and major versions
  - Upgrade standby site
  - Drain connections from primary and switchover to standby
  - Reconnect connections to new primary at the higher version
  - Upgrade new standby via redo stream
- Same procedures as used for RAC rolling

Planned Maintenance – Moving Connections

- Operational Steps
  1. Check current status of the services
  2. Stop services normally (not using FORCE option) on the node to be maintained
  3. Disable the service to prevent unexpected restart
  4. Disconnect long-running sessions after the current transaction completes

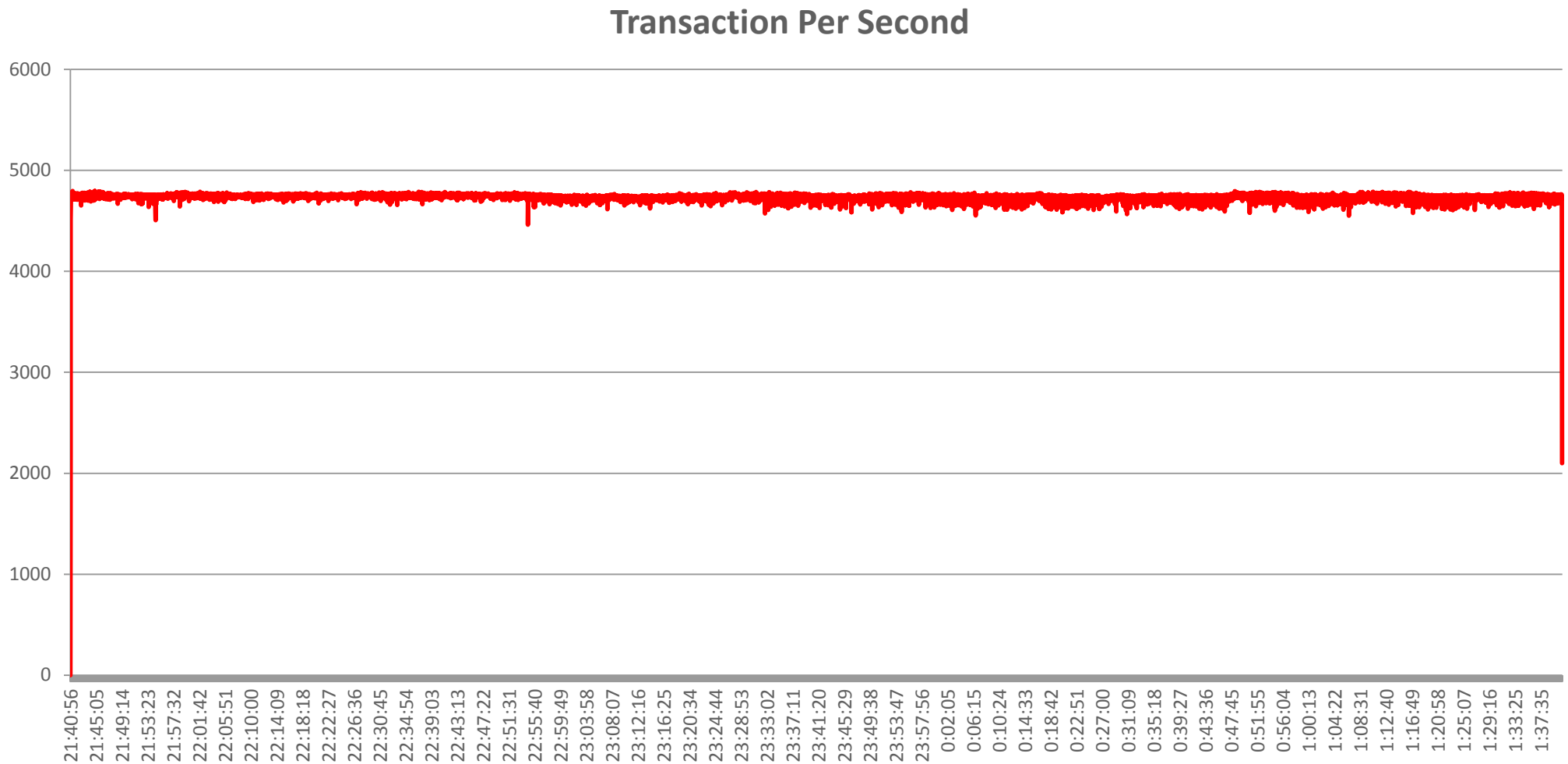
     Run the `DBMS_SERVICE.DISCONNECT_SESSION` package
  5. Repeat Steps 2 - 4 for all services affected by the maintenance
  6. Shutdown the database instance IMMEDIATE

Planned Maintenance – Moving Connections

- Operational Steps (Cont'd)

7. Perform desired maintenance
8. Start up instance(s) on the node
9. Enable services that were previously disabled
10. Start services that were stopped.
11. Observe that connections are appearing on the service again
12. Repeat this procedure for each node or instances that need to be shutdown for maintenance

OLTP Workload During RAC Planned Maintenance



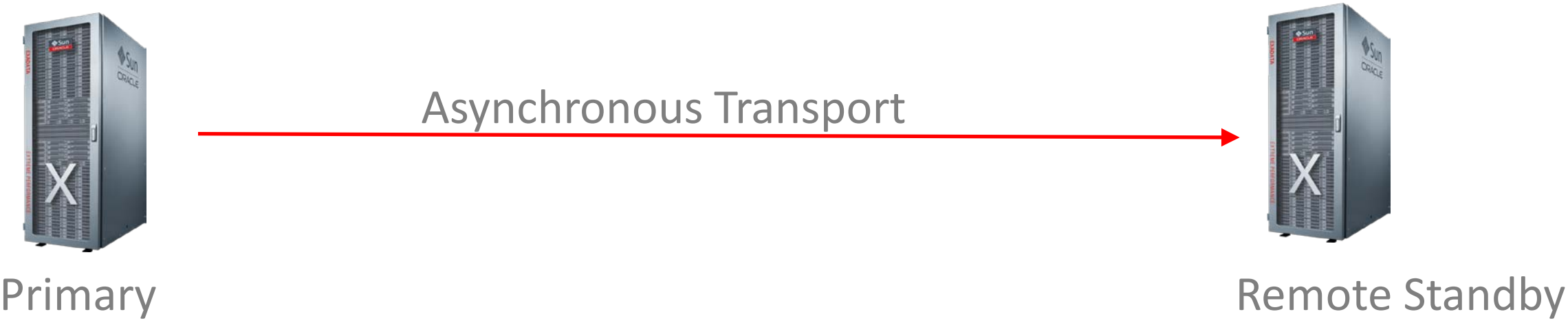
No impact to application during service stop/starts

Unplanned Outages

- Application Continuity provides zero application outage for:
  - RAC node / instance failure
  - Complete clusterware failure
  - Database failure
  - Site failure
- Full site failover vs partial site failover
  - Full site failover typically has redundant mid tiers at remote site
    - Must start mid tiers at the remote site
  - Partial site fails over application connections to the new primary
    - Mid tiers do not have to be restarted

MAA Configuration Options

Remote Disaster Recovery with Maximum Performance



- Second system deployed for remote DR
  - Asynchronous redo transport, Data Guard Maximum Performance
  - Active Data Guard: offload read-only reporting

MAA Configuration Options

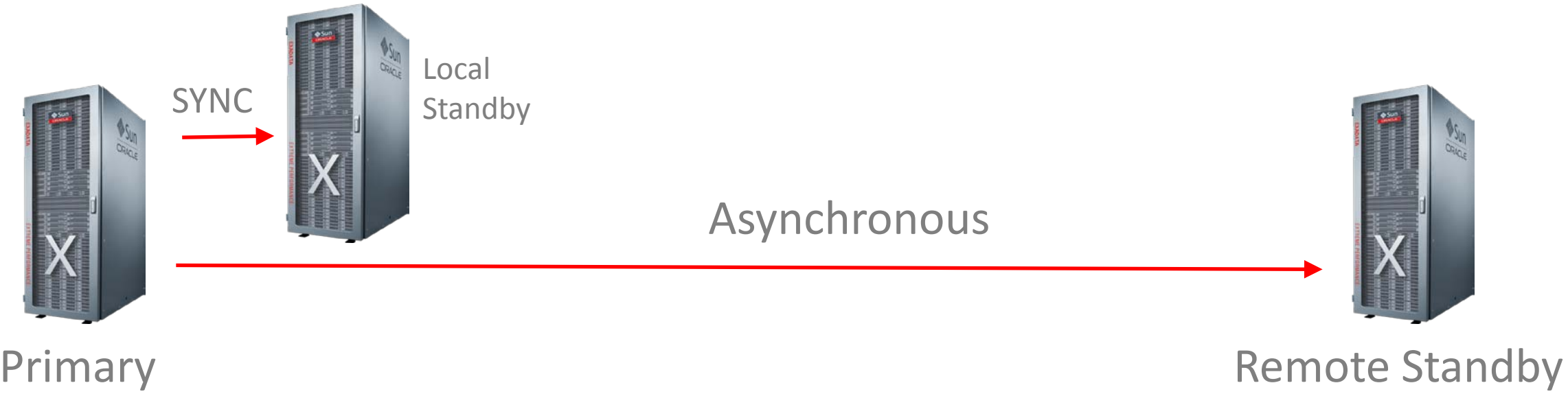
Local Disaster Recovery with Zero Data Loss

Primary Local Standby

- Second system deployed for local DR (within 200 miles)
  - Synchronous redo transport, Data Guard Maximum Availability
  - Active Data Guard: offload read-only reporting

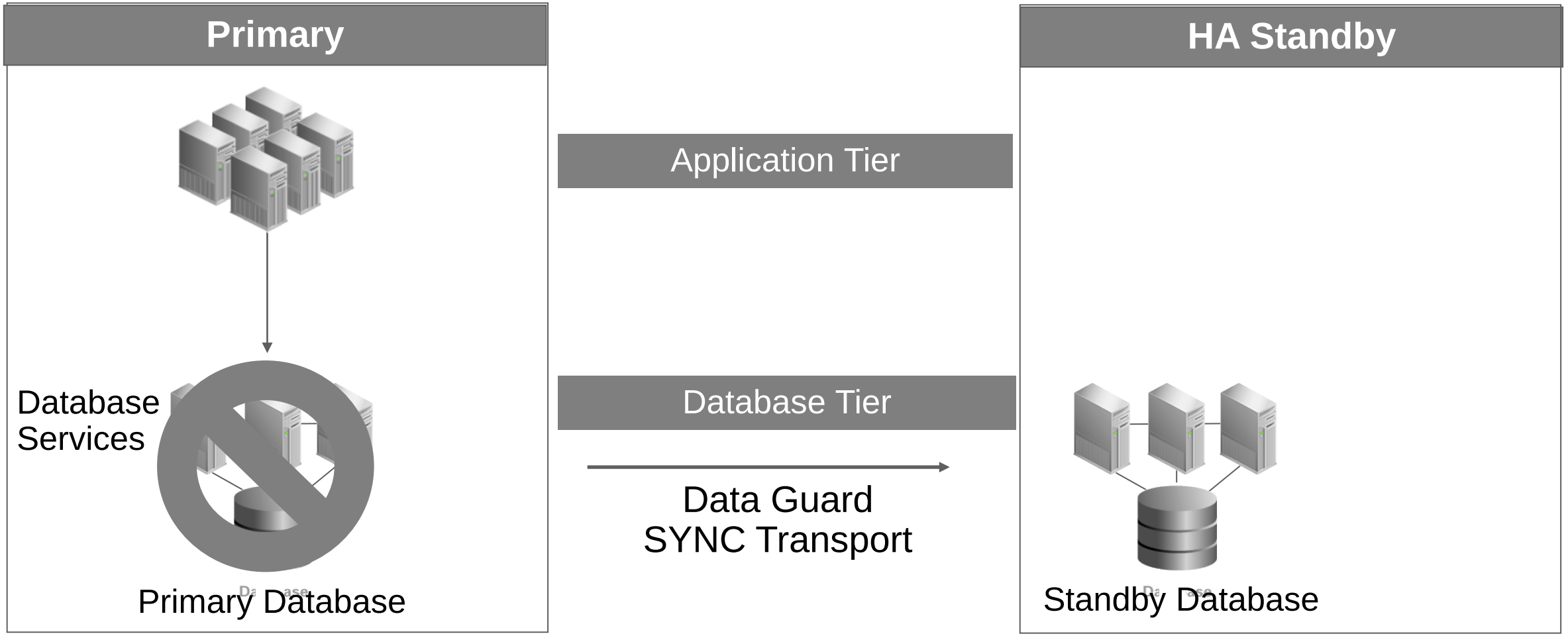
MAA Configuration Options

Multi-Standby: Local HA Failover plus Geographic Protection

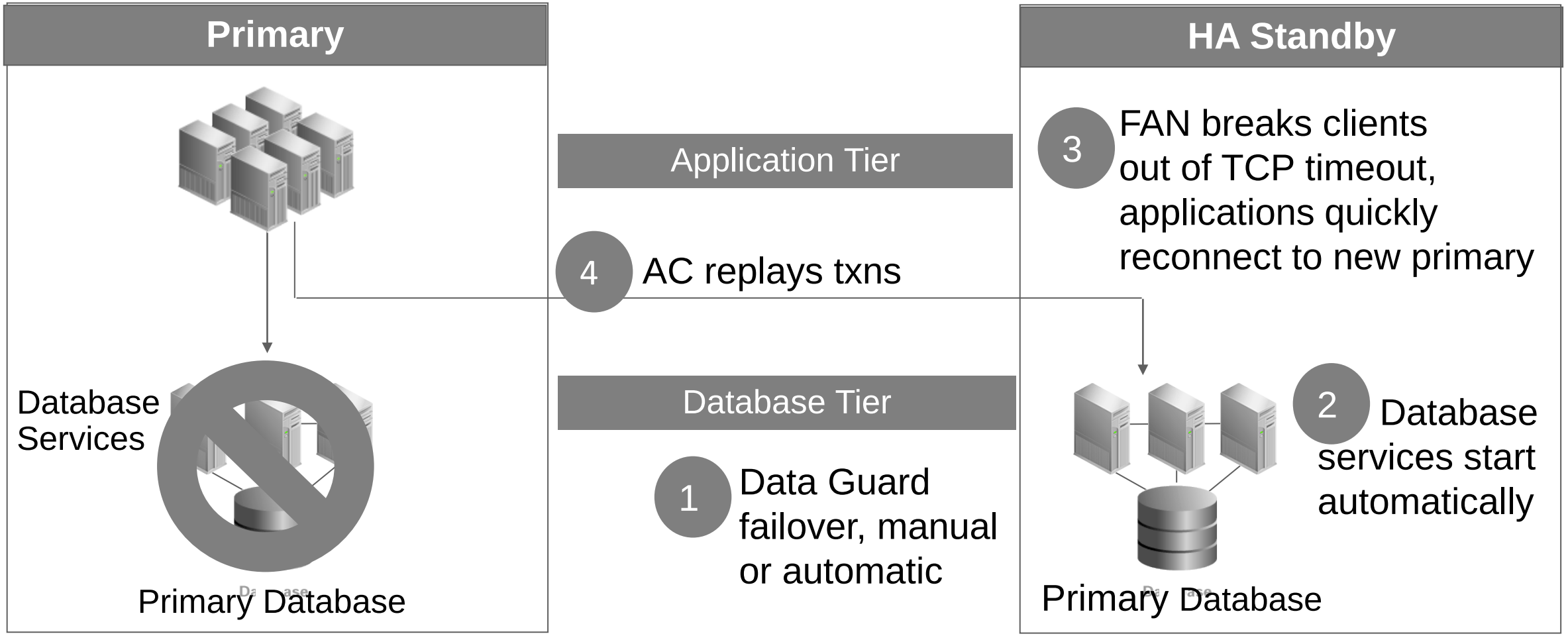


- Dual standby configuration
  - Local standby is primary failover target with zero data loss
  - Remote standby is failover of last resort
  - Either is used to offload read-only workload, backups, rolling upgrades, test

Integrated Application Failover



Integrated Application Failover



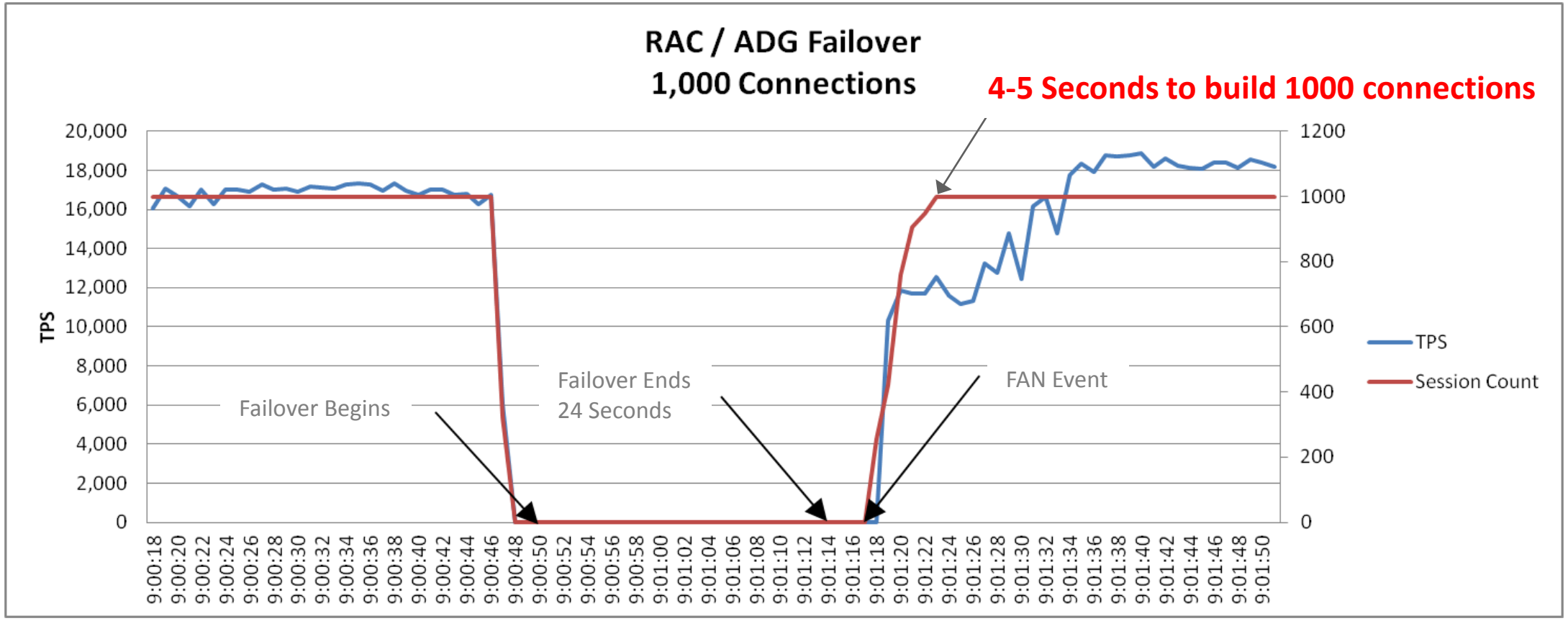
Demo

Eliminating Logon Storms

- Before: 5000 simultaneous connections would experience some failures and take 46 seconds
- Now: 5000 simultaneous connections, zero client-side errors and all clients get connected in 6-8 seconds.
  - Increase listen backlog at OS level: net.core.somaxconn=6000
  - Increase listener queuesize: TCP.QUEUESIZE=6000
- Opposite concern of throttling logon storms still leverage connection_rate from the listener

Application Failover Timings

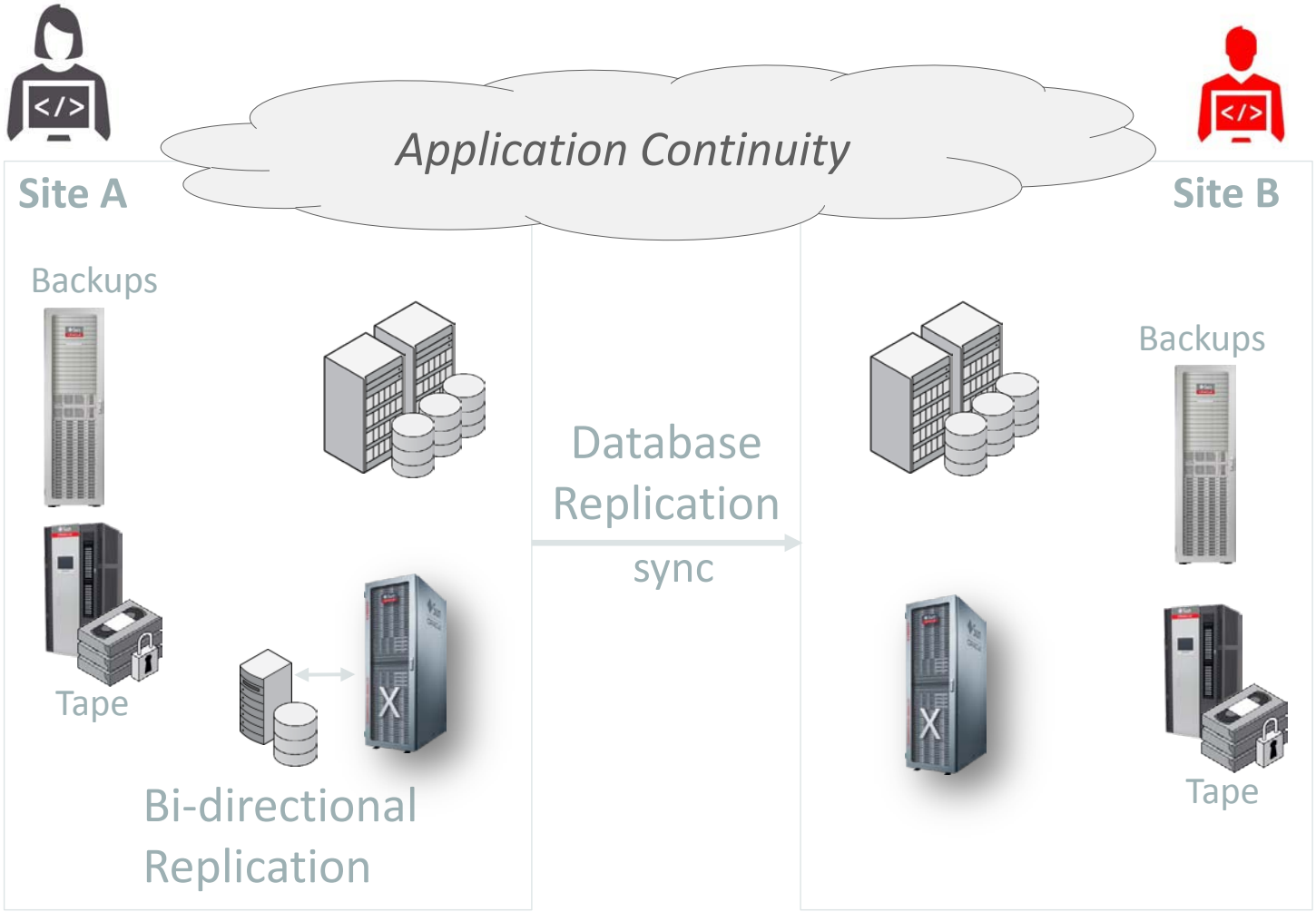
UCP with AC/TG



Wrap up

Platinum Tier

Zero Application Outage for Platinum Ready Applications



- Outages masked from applications, in-flight transactions preserved
  - Application Continuity
  - Zero data loss failover, LAN or WAN
    - Active Data Guard / Far Sync
- Bi-directional replication and zero downtime maintenance
  - Oracle GoldenGate
- Online patching for applications
  - Edition-based Redefinition
- Automated workload management
  - Oracle Global Data Services

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.

Hardware and Software
Engineered to Work Together


ORACLE®