Developer Runbook for Application Continuity

Carol Colrain

Ian Cookson

Troy Anthony

Oracle Database Development

Safe Harbor

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, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Statements in this presentation relating to Oracle's future plans, expectations, beliefs, intentions and prospects are "forward-looking statements" and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle's Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading "Risk Factors." These filings are available on the SEC's website or on Oracle's website at http://www.oracle.com/investor. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.

Program Agenda

What is Continuous Availability?

Maintenance Without Impact

Transparent Application Continuity

Customer Examples

What is Continuous Availability?

Continuous Availability focuses on the User Experience

- Planned maintenance, timeouts, unplanned outages hidden
- Inflight work continues
- Performance is predictable

It is easy to achieve Continuous Availability today

Is continuous availability achievable today?

Absolutely, with best practices.

Stars align when our tips are followed

What are the pre-requisites to achieve this?

1. **A well behaved application** – we will explain how TAC solves this
2. **Enabled Client Stack** (driver, pool, or app server)
   Oracle or enabled 3rd party stack
   Properly configured Net connection
3. **Properly configured database and Grid Infrastructure**
   Configuration necessary to cover likely outages
4. **Business SLAs can be met during outages**
   Recovery time performance targets
   Maintenance window

Many deployments do not currently meet these requirements!

Baseline – Understand where you are

Is the workload divided into services?

Is the application using a proper connection string?

If using a connection pool are connections returned regularly?

Does your application have built-in failover?

Do you have head room to failover to other nodes?

Are application level timeouts set?

Are special features used: XA, external files, side-effects such as fie transfers, OJVM

Download questionnaire from OTN

Not essential, but a head start if you do

New Concept: Request Boundaries

Delineate the Unit of Work

Oracle Pools 12c+, Java Standard (JDK9+), Transparent Application Continuity (TAC)

```
PoolDataSource pds = GetPoolDataSource();
Connection conn = pds.getConnection();
PreparedStatement pstmt =

      …
      SQL, PL/SQL
      …

conn.commit();
conn.close();
```

Begin Request

Request Body often ends with **COMMIT**

End Request

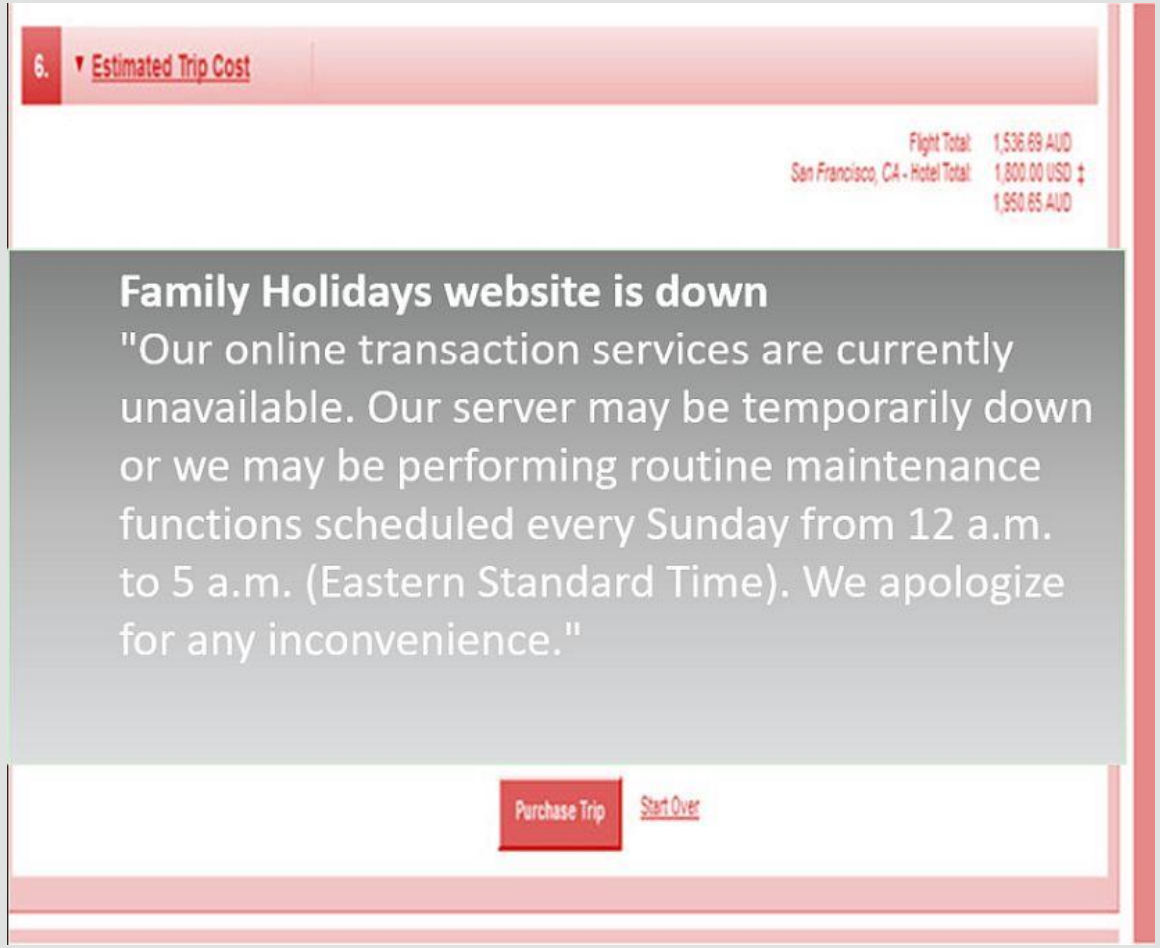
How do we use Requests?

- Draining
- Load Balancing
- Recovery of in-flight work
- Planned Failover
- Reset session state
- Performance metrics
- Protection level

Maintenance with No Impact

Preventable situations:

- Service is unavailable
- Application owners unable to agree maintenance windows
- Long running jobs see errors
- DBA's and engineers work off hours
- Application and middleware components need to be restarted



Drain Work before Maintenance

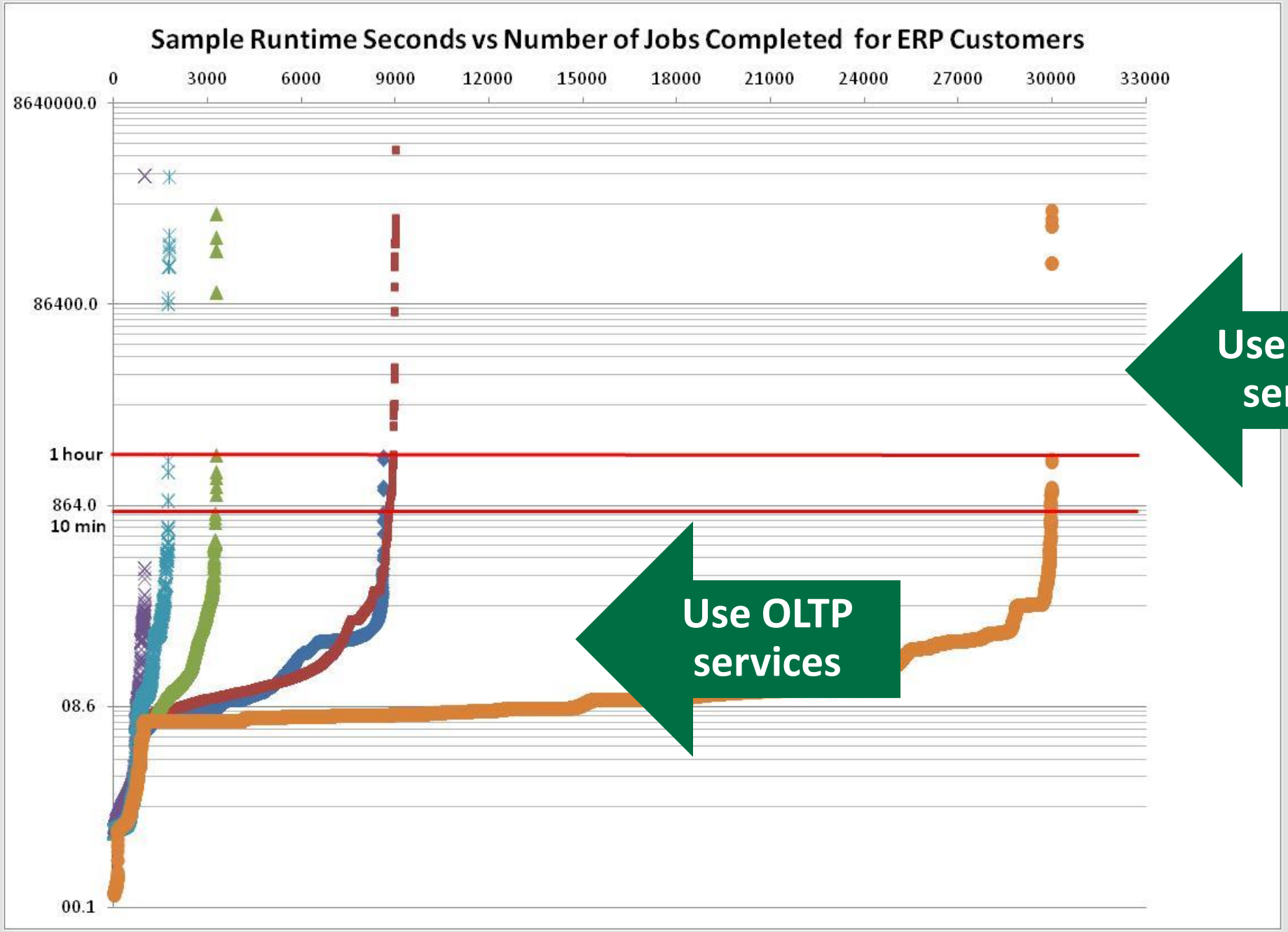
Services are relocated

Automatically, user work drains

Many places to drain:

- Return to connection pool
- Connection tests
- Log-Off
- Planned Failover

Keep it simple – Drain Batch then Online

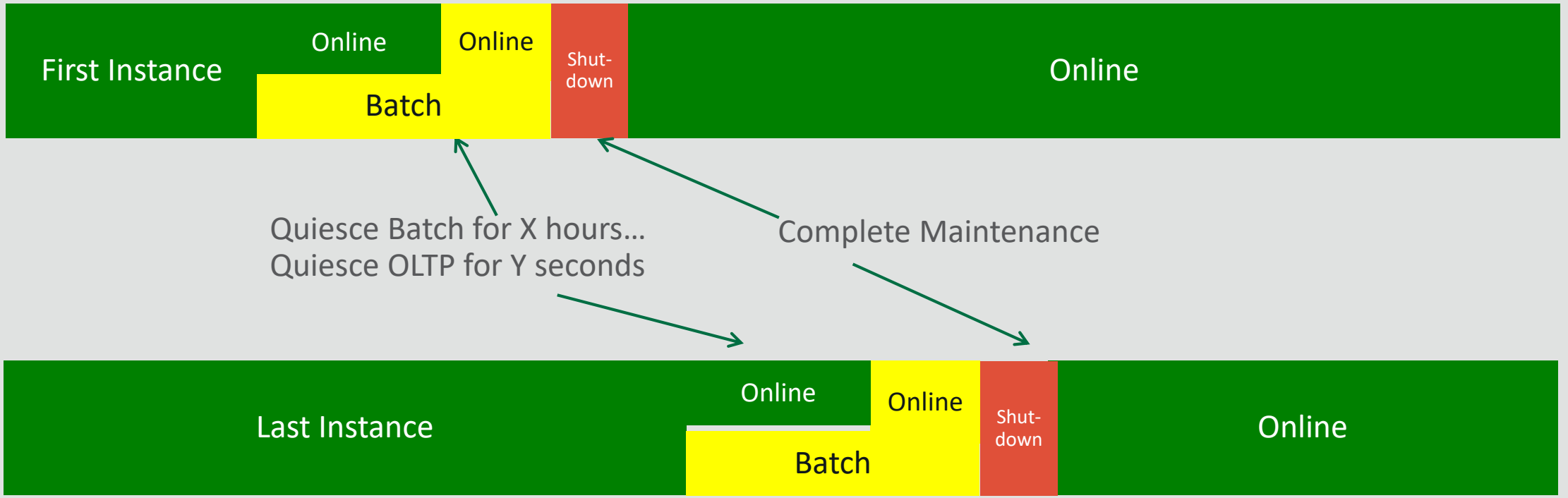


Stagger Draining – Batch then Online Services

Use at least two application services – online and batch/backend

Start draining batch early

Online drains in seconds



Tip: Use Oracle Pool, Return to Pool

BEST SOLUTION – FAN Drains and Rebalances

Applications use	**Oracle Pools: WebLogic Active GridLink, UCP, ODP.NET managed and unmanaged, OCI Session Pool, Tuxedo** 3rd party App Servers using UCP: IBM WebSphere and Liberty, Apache Tomcat, NEC WebOTX, Red Hat WildFly (JBoss), Spring
DBA uses	`srvctl [relocate|stop] service -drain_timeout`
Sessions Drain Automatically	Immediately new work is redirected Idle sessions drain gradually **Active sessions are released when returned to pools**

FAN

Drains & rebalances

FAN Auto-configures for INSTANT Interrupt

The dead thing cannot tell you it is dead

All Oracle uses ONS

- JDBC Universal Connection Pool
- OCI/OCCI driver
- ODP.NET Unmanaged Provider (OCI)
- ODP.NET Managed Provider (C#)
- OCI Session Pool
- WebLogic Active GridLink
- Tuxedo
- JDBC Thin Driver **(new 12.2)**
- **CMAN in Traffic Manager mode (new 18c)**

Auto-Configured

```
DESCRIPTION =
    (CONNECT_TIMEOUT=90)
    (RETRY_COUNT=20)(RETRY_DELAY=3)
    (TRANSPORT_CONNECT_TIMEOUT=3)
    (ADDRESS_LIST =
        (LOAD_BAL
        ( ADDRESS = (PROTOCOL = TCP)
        (HOST=primary-scan) (PORT=1521)))
    (ADDRESS_LIST =
        (LOAD_BA
        ( ADDRESS = (PROTOCOL = TCP)
        (HOST=second-
        scan)(PORT=1521)))
(CONNECT_DATA=(SERVICE_NAME=gold)))
```

ONS Node Set 1

ONS Node Set 2

FAN for JDBC

12c or later JDBC clients and database

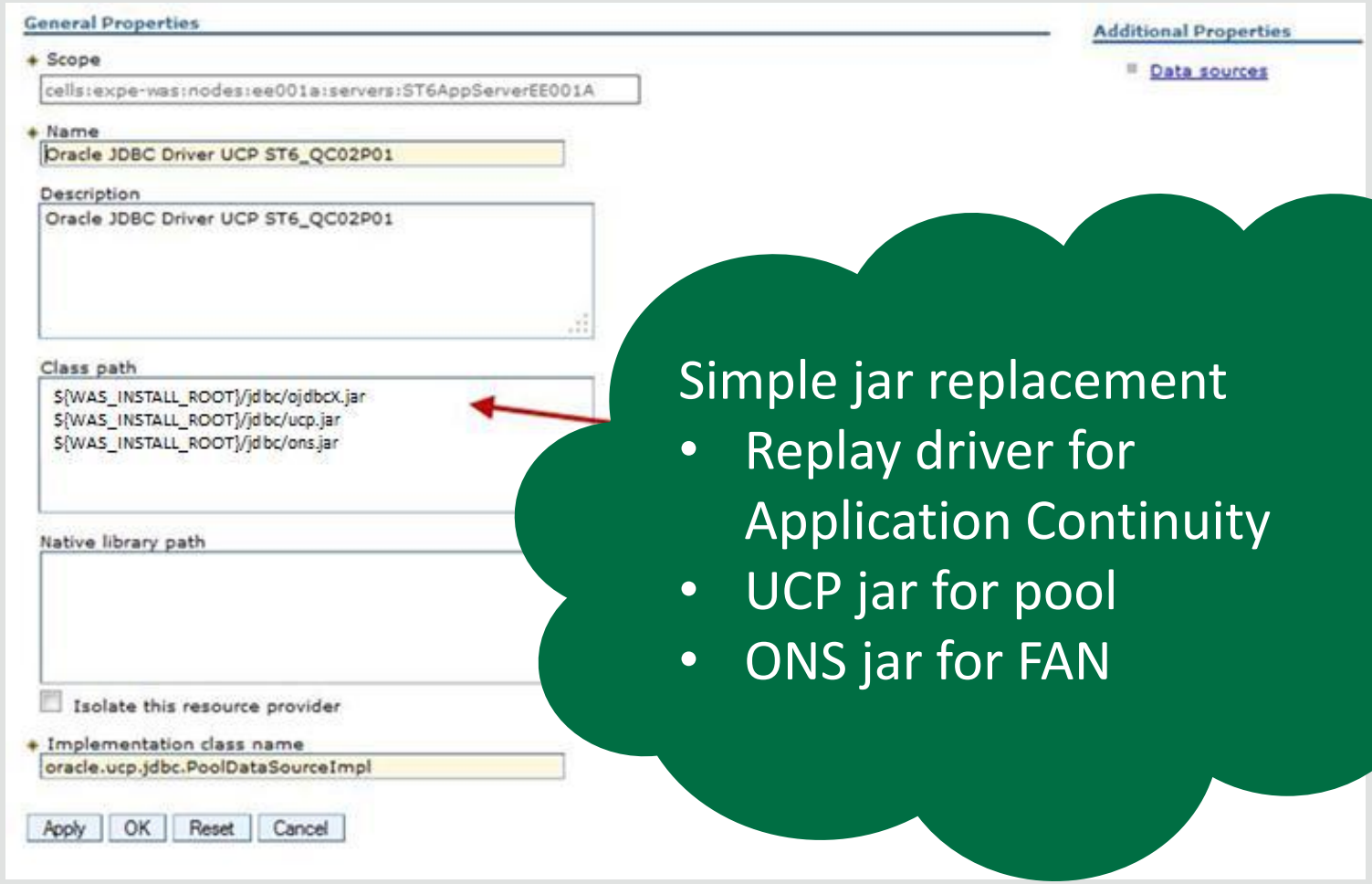
1. Use recommended URL for auto-ons
2. ons.jar (plus Wallet jars, osdt_cert.jar, osdt_core.jar, oraclepki.jar) on the CLASSPATH
3. UCP pool property fastConnectionFailoverEnabled=true
4. Third party pools, UCP is recommended
5. Open port 6200 for ONS

Before 12c clients

Set oracle.ons.nodes =mysun05:6200,mysun06:6200, mysun07:6200,mysun08:6200

Tip: 3rd Party Java Applications

A very simple jar replacement



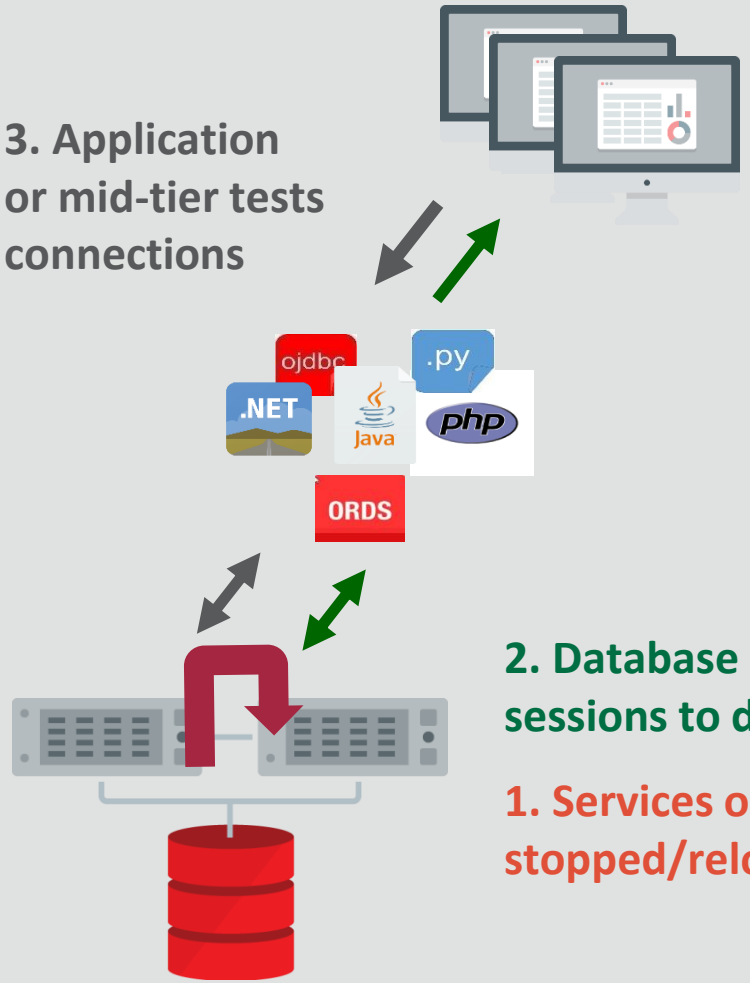
Simple jar replacement

- Replay driver for Application Continuity
- UCP jar for pool
- ONS jar for FAN

- IBM WebSphere
- IBM Liberty
- Apache Tomcat
- NEC WebOTX
- Red Hat WildFly (JBoss)
- Hibernate
- Spring
- custom

Tip: Enable Connection Tests

18c Database and 18c Drivers



Application “tests” the connection

Database/driver responds connection is bad

New work continues on another connection

Tip: View in DBA_CONNECTION_TESTS
Add more tests with DBMS_APP_CONT_ADMIN

Tip: Enable Connection Tests

Client-Side

UCP	`setValidateConnectionOnBorrow()`, isValid is local
OCI	`OCI_ATTR_SERVER_STATUS` in server handle
ODP.NET	`CheckConStatus` is on by default.

Database-side

CONNECTION_TEST	SQL_CONNECTION_TEST	ENABLED
SQL_TEST	SELECT 1 FROM DUAL	Y
SQL_TEST	SELECT COUNT(*) FROM DUAL	Y
SQL_TEST	SELECT 1	Y
SQL_TEST	BEGIN NULL;END	Y
PING_TEST		NA
ENDREQUEST_TEST		NA

Tip: Enable Connection Tests for Application Servers

Application Server	Test Name	Connection Test to DB
Oracle WebLogic – Generic and Multi data sources	TestConnectionsOnReserve TestConnectionsOnCreate	isUsable() SQL – SELECT 1 FROM DUAL
Oracle WebLogic Active GridLink	Embedded	isUsable()
IBM WebSphere	PreTest Connections	SQL – SELECT 1 FROM DUAL
RedHat WildFly (JBoss)	Check-valid-connection-sql	SQL – SELECT COUNT(*) FROM DUAL
Apache Tomcat	TestOnBorrow TestOnRelease	SQL – SELECT 1 FROM DUAL
ODP.Net Unmanaged	Connection.status()	OCI_ATTR_SERVER_STATUS

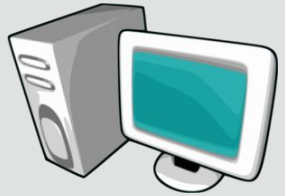
Tip: Enable Connection Tests for Applications

Application	Condition	Connection Test to DB
eBusiness Suite	Connection borrowed from WebLogic	TestConnectionsOnReserve with “BEGIN NULL;END;”
Fusion Applications	Connection returned to WebLogic and C++ pools and checked	TestConnectionsOnReserve with isValid() OCI_ATTR_SERVER_STATUS
Siebel	Connection requested	OCI_ATTR_SERVER_STATUS
Peoplesoft	Connection requested	OCI_ATTR_SERVER_STATUS
Customer example	Custom pool with Metadata table Checks status every 60 seconds	OCI_ATTR_SERVER_STATUS

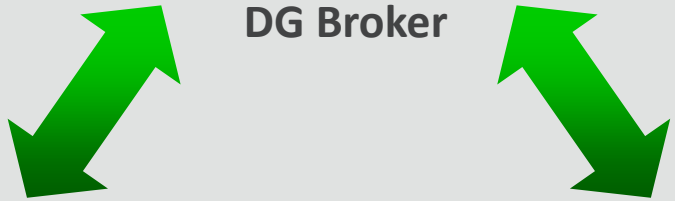
Align Application Timeouts

TIP: Nothing to do when RAC or RAC One

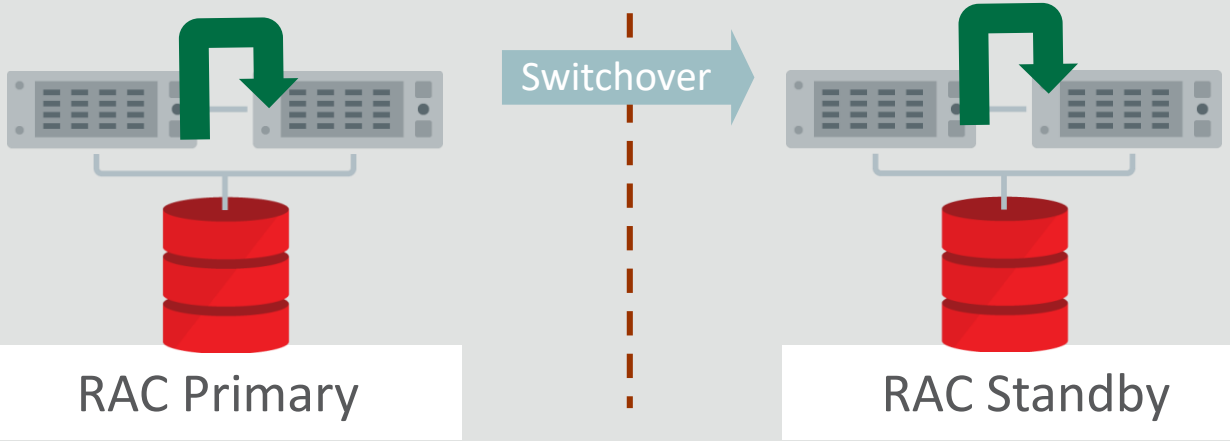
TIP: Set Application Timeout > Drain + Switch to DG



DG Broker



Stop or Relocate Service to Drain Work

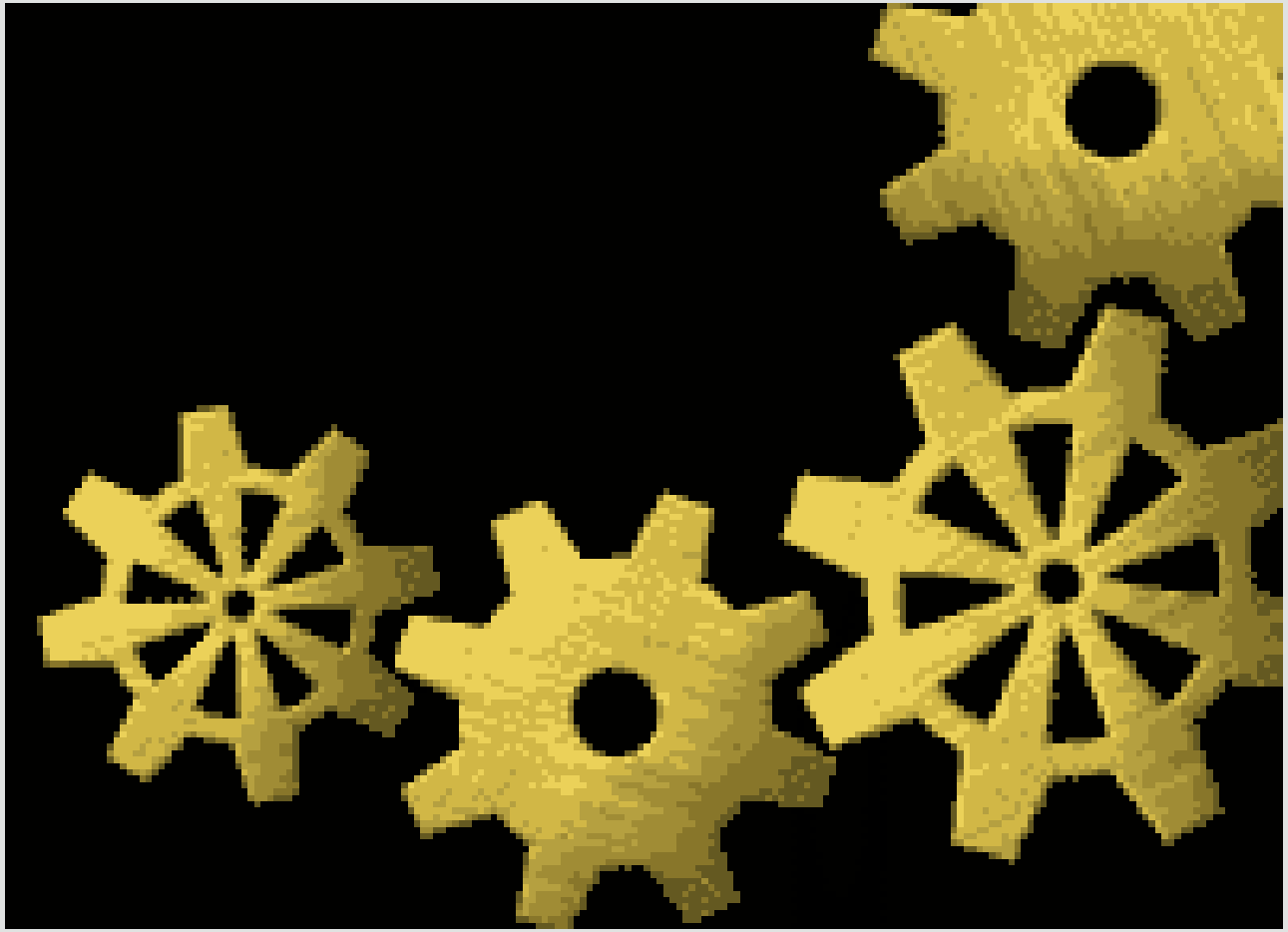


Drain Work

Switchover

For DBA Operations

Available 12.2



Preset services to drain

drain_timeout 600 sec.

stopoption immediate

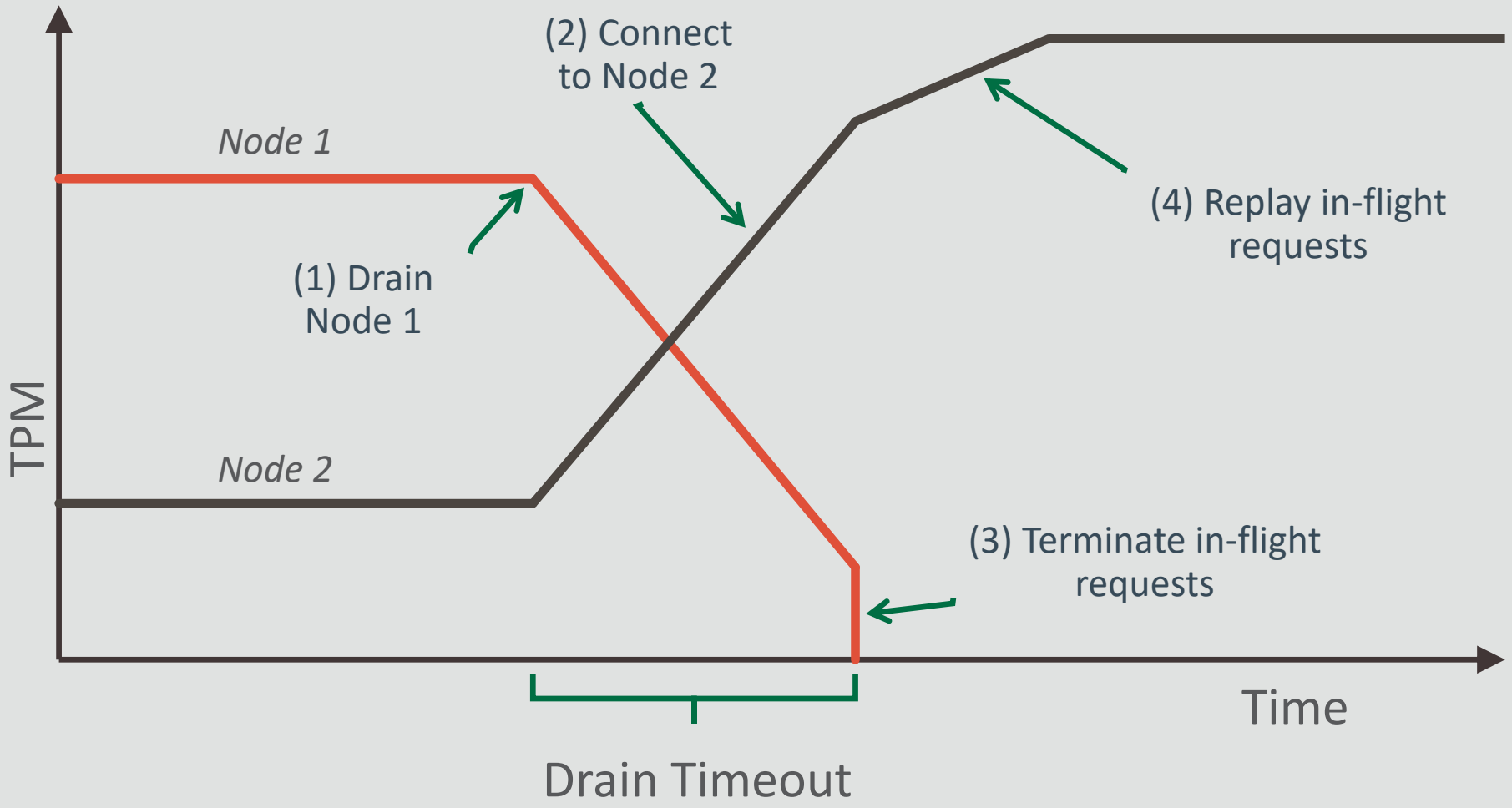
Group operations pdb, instance, node, database

FAN full and in-band NEW IN 19c

Drain Sessions Before Maintenance

Demonstration

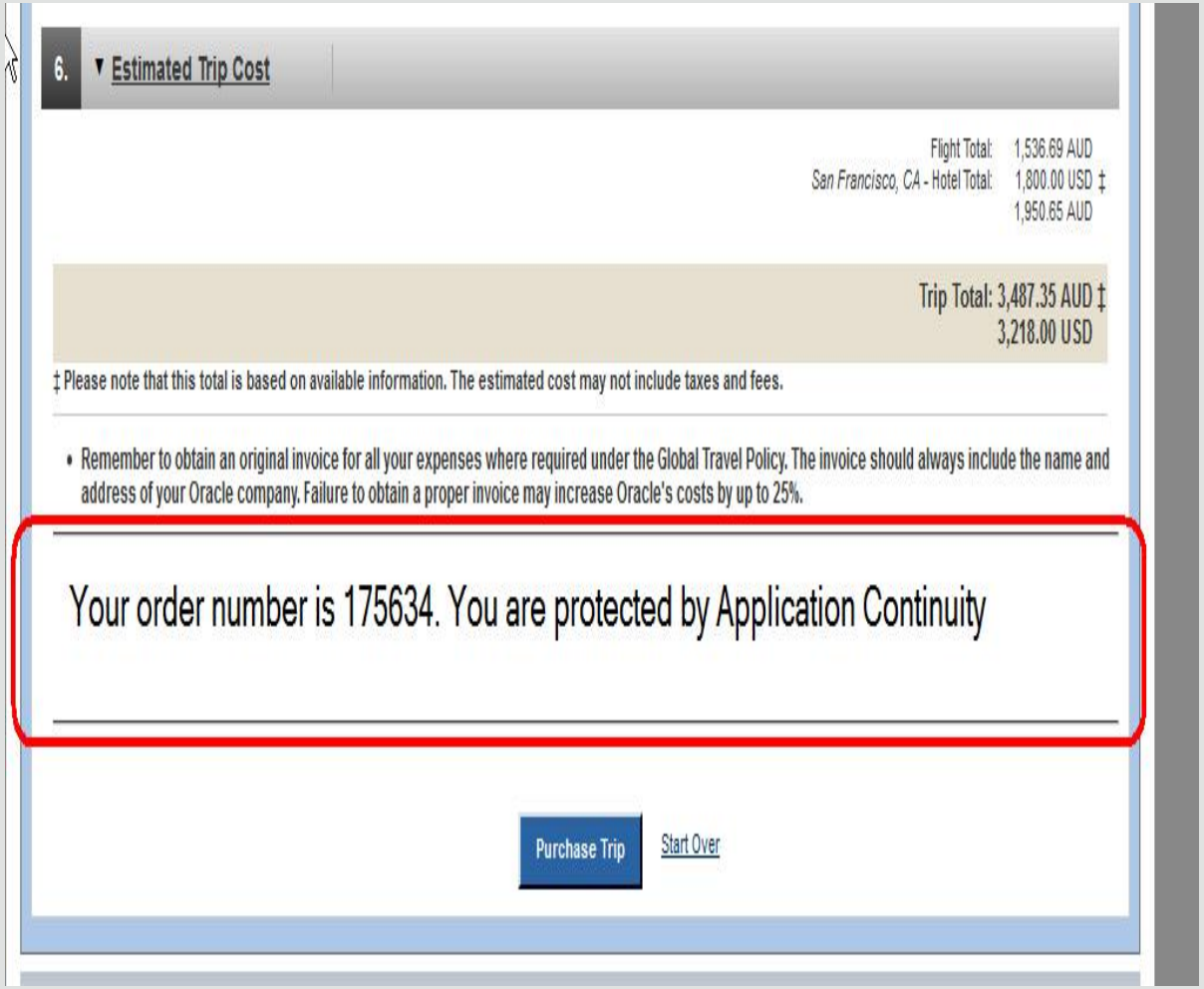
Drain… Connect… Failover



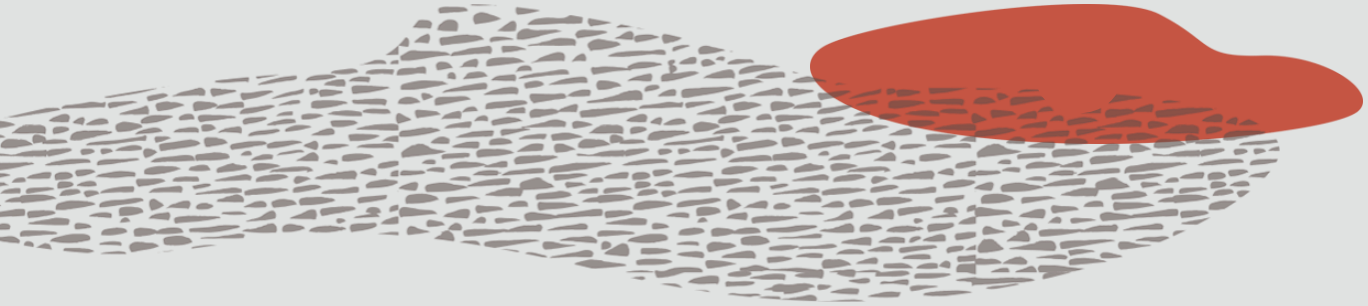
Transparent Application Continuity

In-flight work continues:

- Replays in-flight work on recoverable errors
- Masks hardware, software, network, storage, session errors and timeouts
- 12.1 JDBC-Thin, UCP, WebLogic Server, 3rd Party Java application servers
- 12.2 OCI, ODP.NET unmanaged, JDBC Thin on XA data source
- Transparent Application Continuity NEW IN 19c
- TAC on by default on ADB Dedicated



Standardize on TAC to hide impact of outages



Application Continuity Explained

Normal Operation

Client marks requests: explicit and **implicit**.

Server **tracks session state**, decides which calls to replay, disables side effects.

Directed, client holds original calls, their inputs, and validation data.

Failover Phase 1: Reconnect

Checks replay is enabled

Verifies timeliness

Creates a new connection

Checks target database is legal for replay

Uses Transaction Guard to guarantee commit outcome

Failover Phase 2: Replay

Restores and verifies the session state

Replays held calls, restores mutables automatically

Ensures results, states, messages match original.

On success, returns control to the application

Exclusions

When replay is not enabled

Application Level

Default database or default PDB service

Deprecated, JDBC classes before 18c

Remainder of Request

Alter system, database, session (subset)

Best effort streaming lobs

XA after promote

OCI – old OCIStmtPrepare, misc apis

DB links ADG to primary

Target Database

Different Database

DBMS_ROLLING*

Golden Gate

3rd Party Replication

Using Transparent/Application Continuity

1. Request Boundaries	Use Oracle pools and Return to Pool
2. Session State	Use FAILOVER_RESTORE
3. Side Effects	TAC - not replayed, AC - customizable
4. SYSDATE, Sequences,...	Do nothing for SQL, Grant for PL/SQL
Reset State	Reset session state between requests NEW IN 20c
Coverage	Always know your protection level

Request Boundaries Are Automatic

Transparent Application Continuity

NEW IN 19c

- Request boundaries advanced automatically
- Capture re-enables, if disabled
- Smaller capture set means faster recovery
- Pooled, non-pooled and long running

For highest protection

- **Return connections to pool**
- **Return cursors to driver statement cache**
- **Reset states (option)**

Explicit Request Boundaries, Standard in JDK9

Required for AC, Recommended with TAC

Oracle

Return connections

UCP

WebLogic GridLink

ODP.NET unmanaged

OCI Session Pool

SQL*Plus

Use UCP or Java Standard

Return connections

IBM WebSphere

IBM Liberty

Apache Tomcat

NEC WebOTX

RedHat Wildfly (JBoss)

Spring

custom

Custom Java

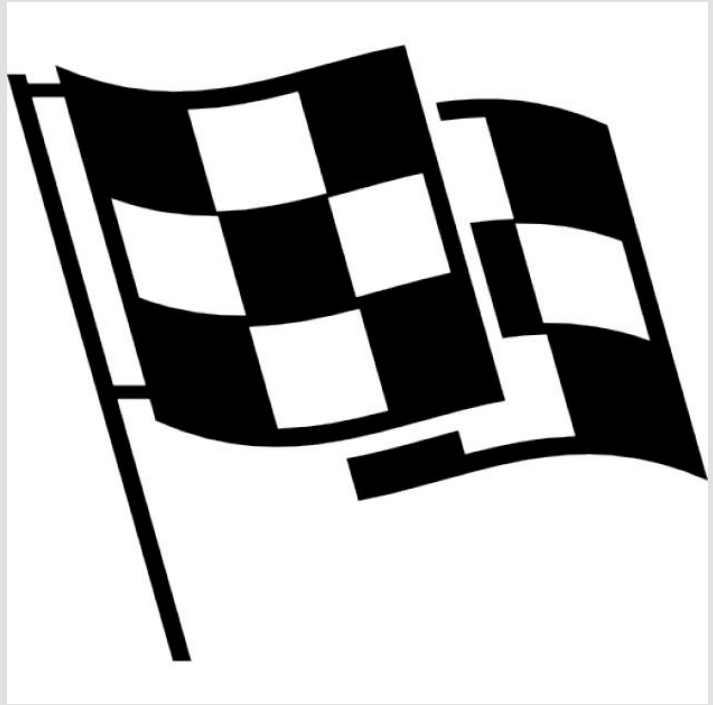
Return connections

Add Request boundaries

- beginRequest
- endRequest

Session State Restored for Replay

- Restore session states before replaying
  - `FAILOVER_RESTORE` on your service
- Full parameter restore NEW IN 19c
  - Configure wallets
- Customization
  - Labels, TAF callback, Logon triggers



Restore Before Replaying

nls_nchar_conv_excp
nls_calendar
nls_comp
nls_currency
nls_date_format
nls_date_language
nls_dual_currency
nls_iso_currency
nls_language
nls_length_semantics
nls_numeric_characters
nls_sort
nls_territory
nls_time_format
nls_time_tz_format
nls_timestamp_format
nls_timestamp_tz_format
allow_rowid_column_type
approx_for_aggregation

approx_for_count_distinct
approx_for_percentile
cursor_sharing
default
plscope_settings
plsql_ccflags
plsql_debug
plsql_optimize_level
plsql_warnings
recyclebin
result_cache_mode

aq_tm_processes
cell_offload_compaction
cell_offload_parameters
cell_offload_plan_display
cell_offload_processing
cell_offloadgroup_name
commit_logging
commit_wait
commit_write
containers_parallel_degree
_default
java_jit_enabled
max_dump_file_size
multishard_query_data_consistency
multishard_query_partial_results
object_cache_max_size_percent
object_cache_optimal_size
olap_page_pool_size
optimizer_adaptive_plans

optimizer_adaptive_reporting_only
optimizer_adaptive_statistics
optimizer_dynamic_sampling
optimizer_features_enable
optimizer_index_caching
optimizer_index_cost_adj
optimizer_inmemory_aware
optimizer_mode
optimizer_use_invisible_indexes
optimizer_use_pending_statistics
parallel_force_local
_retained_size
sort_area_size
spatial_vector_acceleration
sql_trace
sqltune_category
star_transformation_enabled
statistics_level
temp_undo_enabled
timed_os_statistics
timed_statistics
tracefile_identifier
workarea_size_policy

Set **FAILOVER_RESTORE** on your service and use wallets

NEW IN 19c

Side Effects Not Replayed

TAC – stops capture automatically until next enable point

TAC decides if any requests should not be replayed, e.g.

- `UTL_HTTP`
- `UTL_URL`
- `DBMS_FILE`
- `DBMS_FILE_TRANSFER`
- `UTL_SMTP`
- `UTL_TCP`
- `UTL_MAIL`
- `EXTPROC`

Customized? - use AC

Restore SYSDATE, SYSTIMESTAMP, Sequences …

NEW IN 19c

During replay the same values are restored for SYSDATE, SYSTIMESTAMP, and SEQUENCES

- Automatically for SQL
- Grant keep for PL/SQL

For owned sequences:

```
ALTER SEQUENCE.. [sequence]  [KEEP|NOKEEP]
CREATE SEQUENCE.. [sequence]  [KEEP|NOKEEP]
```

Grant and Revoke for other users:

```
GRANT [KEEP DATE TIME | KEEP SYSGUID][to USER]
REVOKE [KEEP DATE TIME | KEEP SYSGUID] [from USER]
GRANT KEEP SEQUENCE on [sequence][to USER]
REVOKE KEEP SEQUENCE on [sequence][from USER]
```

Reset Session State Between Requests

NEW IN 20c

Applications use state in requests

- Temporary tables
- PL/SQL globals
- Cursors in fetch ....

Database resets state at end of request

Next request starts with clean state

Service attribute (RESET_STATE)

Always Know Your Protection Level

- AWR, system, session, service statistics
- Your application is fully protected when

  cumulative user calls in request = cumulative user calls protected

Statistic	Total	per Second	per Trans
cumulative begin requests	1,500,000	14,192.9	2.4
cumulative end requests	1,500,000	14,192.9	2.4
cumulative user calls in request	6,672,566	63,135.2	10.8
cumulative user calls protected	6,672,566	63,135.2	10.8

acchk – Protection Report when needed

Outage Type	Status	Message
Coverage checks		**TotalRequest = 1088** **PASS = 1084** **WARNING = 1** **FAIL = 3**
	FAIL	Trace file name = db1_ora_30467.trc Line number of Request start = 1409 Request number = 6 SERVICE NAME = (srv_auto_pdb1) MODULE NAME = (SQL*Plus) ACTION NAME = () CLIENT ID = () Coverage(%) = 12 Protected Calls = 1 Unprotected Calls = 7 Row number of the last call before DISABLED : 1422, Disable reason : ORA-41429
	WARNING	Trace file name = CDB12_ora_321597.trc Line number of Request start = 653 Request number = 1 SERVICE NAME = (PDB1_tp.cloud.com) MODULE NAME = (JDBC Thin Client) ACTION NAME = () CLIENT ID = () Coverage(%) = 25 Protected Calls = 1 Unprotected Calls = 3 Row number of the last call before DISABLED : 668, Disable reason : ORA-41409
	FAIL	Trace file name = CDB12_ora_292714.trc Line number of Request start = 1598 Request number = 7 SERVICE NAME = (PDB1_tp.cloud.com) MODULE NAME = (SQL*Plus) ACTION NAME = () CLIENT ID = () Coverage(%) = 16 Protected Calls = 1 Unprotected Calls = 5 Row number of the last call before DISABLED : 1622, Disable reason : ORA-41429
	FAIL	Trace file name = CDB12_ora_112022.trc Line number of Request start = 1167 Request number = 3 SERVICE NAME = (PDB1_tp.cloud.com) MODULE NAME = (JDBC Thin Client) ACTION NAME = () CLIENT ID = () Coverage(%) = 0 Protected Calls = 0 Unprotected Calls = 1 Row number of the last call before DISABLED : 1024, Disable reason : ORA-41406
	PASS	Report containing checks that passed: /scratch/nfs/acchk/orachk_dbj14_091119_10491/reports/acchk_scorecard_pass.html

Disable Reason
ORA-41429: side effect detected

Configuration for Clients

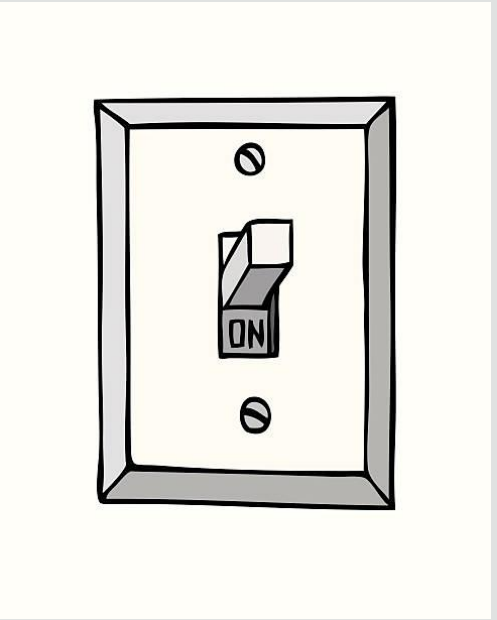
JDBC Thin

- Use replay data source
- `replay datasource=oracle.jdbc.replay.OracleDataSourceImpl`
- Use JDBC driver statement Cache
- Use acchk to check for concrete classes pre-18c driver

OCI, ODP.Net unmanaged, SQL*Plus (19c), open source

- On when enabled on the service
- Use OCI driver statement Cache

Configuration for Services



DRAIN_TIMEOUT = < long time, in seconds >

STOP_OPTION = IMMEDIATE

FAILOVER_TYPE = AUTO or TRANSACTION

FAILOVER_RESTORE = AUTO or LEVEL1

COMMIT_OUTCOME = TRUE

AQ_HA_NOTIFICATIONS=True for FAN OCI

REPLAY_INITIATION_TIMEOUT = 300 (seconds before canceled)

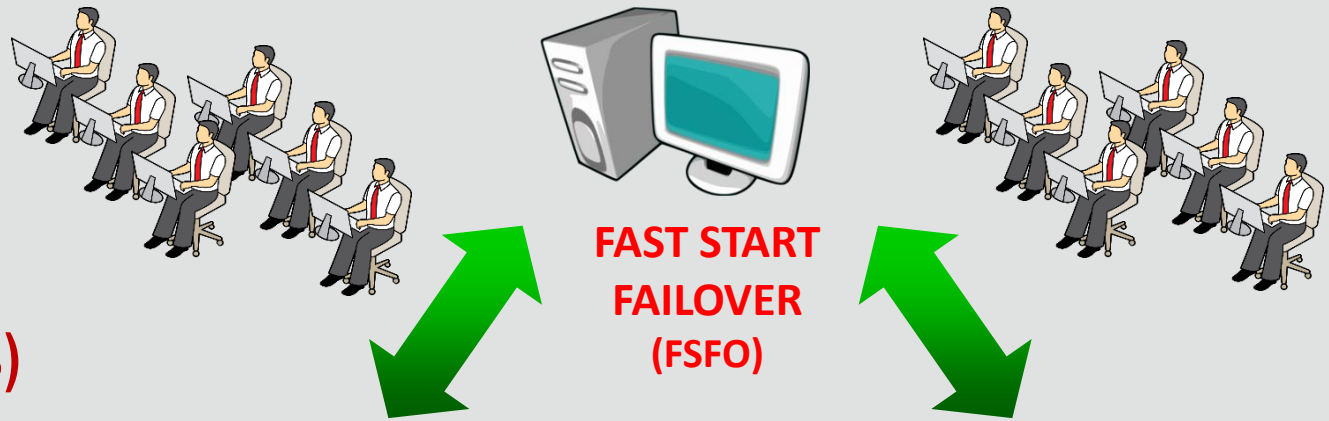
RESET_STATE = NONE or LEVEL1 (for stateless apps)

NEW IN 20c

Align Application Timeout

Application Timeout

- **FSFO + Crash Recovery**
- **2 x MISSCOUNT + FSMT (60s)**

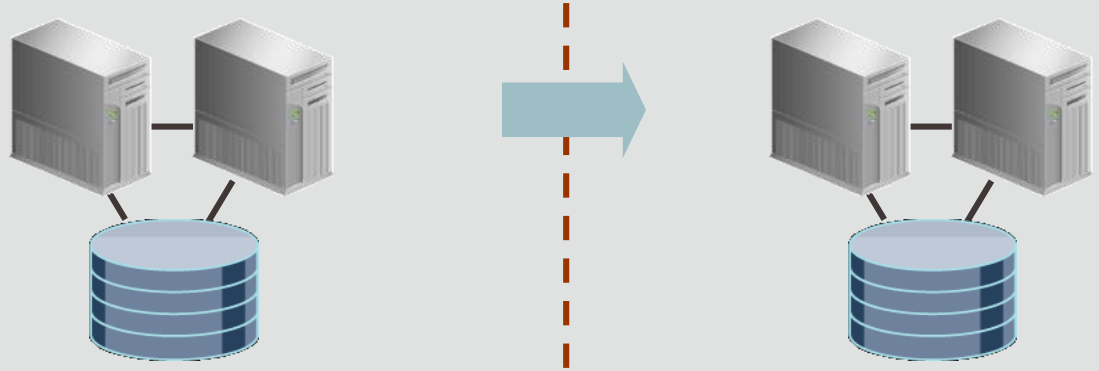


e.g. - MISSCOUNT (15)

EXA Only (2)

FAST START MTTR TARGET (30)

AC REPLAY TIMEOUT (600)



Crash Recovery (FSMT +Open)

AC REPLAY TIMEOUT

RAC Primary

RAC Standby

What do I use?

My Application uses..	TAC	AC	Draining
I don't know	yes	no	yes
Transactions	yes	yes	yes
Oracle state (temp lobs, PL/SQL, temp tables, aq..)	yes	yes	yes
No connection pool	yes	no	yes
Initial state on the session	Yes and custom	Yes and custom	yes
Side effects (utl_mail, dbms_file_transfer..)	Yes, not replayed	Yes, custom	yes

Customer Stories – Unplanned Outages










Chicago Mercantile Exchange

CME Group Overview

CME Group is the world's leading and most diverse derivatives marketplace bringing together those who need to manage risk or those that want to profit by accepting it.

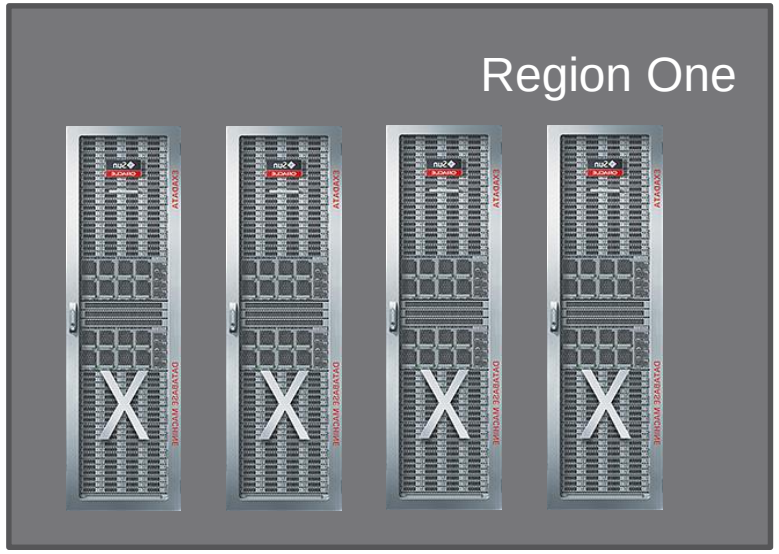
- Operating Multiple Exchanges – CME, CBOT, Nymex and COMEX
- Trade hundreds of products across the globe on a single platform
- Average daily volume of 15.6 million contracts

- CME Clearing – matches and settles all trades and guarantees the creditworthiness of every transaction
- Cleared more than 3.9 billion contracts with a value exceeding $1 quadrillion
- Highest Volume Day – 44.5 million contracts after the election

CME HIGH AVAILABILITY OVERVIEW

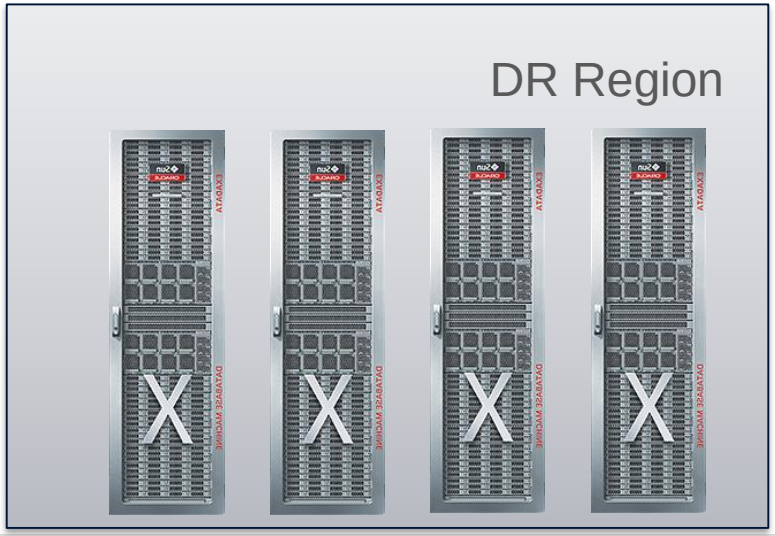
Requirements

- **Critical DB's – 10 second to SLA**
- **Component Failure Cannot cause DR Event**
- 24X7 Application up time
  - Including Planned Maintenance
- RPO – 30 seconds (Disaster Only)
- RTO – 2 hours (Disaster Only)



Solution

- Exadata
  - Addresses Performance
  - Allows Consolidation
  - Reduces recovery time (component failure)
- Active Data Guard
- **Application Continuity** – Planned/unplanned



WHY CME IS ADOPTING APPLICATION CONTINUITY

- Database Outages cause in-flight work to be lost
- A Database Outage can effect many applications concurrently due to schema consolidation
- Critical Applications are becoming 24x7 – These are referential applications
- Database planned downtime on behalf of patching is exceedingly harder to schedule due to shrinking maintenance windows.
- Avoid dedicating maintenance windows to the database group
- Applications work together as a system. It can take serval hours to start and normalize

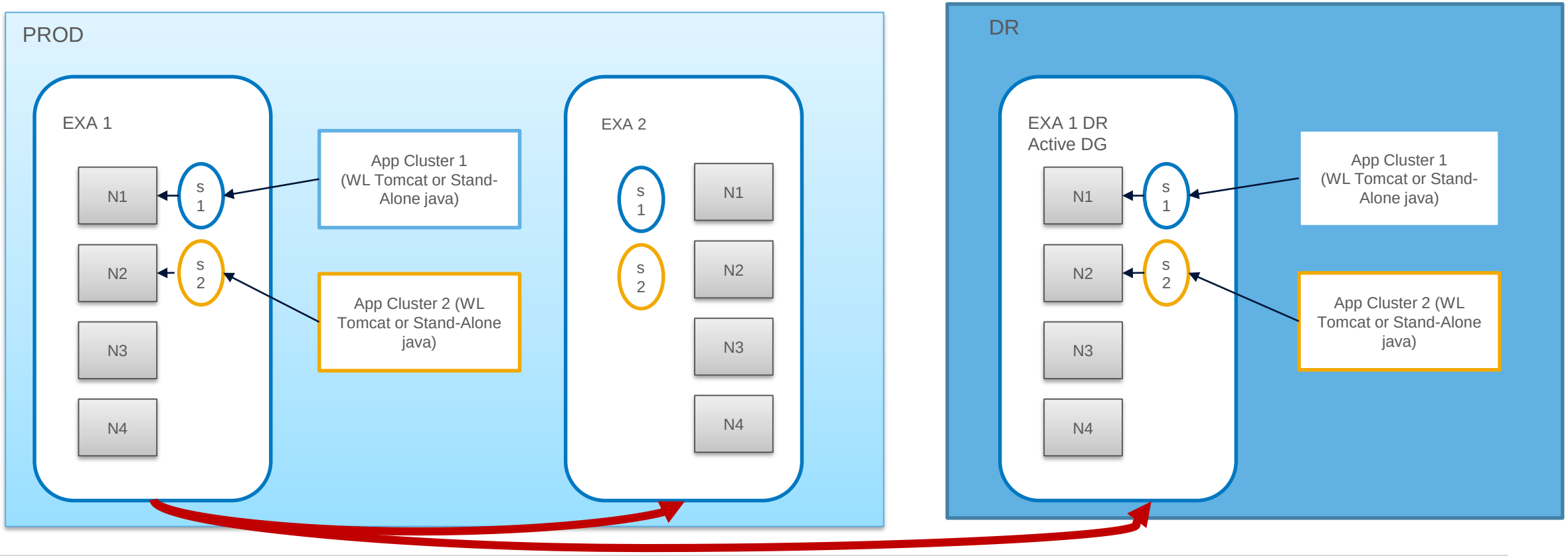
Failure Proofing Applications Is Hard

What's Hard	What's Todays Solution
Hanging on TCP/IP Timeouts – Application is not aware of an issue because there has been no ack for the last operation	• FAN – Fast Application Notification • FCF – Fast Connection Failover • ONS – Notification Services These features work together to overcome TCP hangs
Reconnecting to surviving nodes or standby database after failure	Application Continuity automatically performs connection retries all configurable in the connection string
Assuring any in-flight transactions were committed to the database.	Application Continuity features handle this transparently. Transactions are crosschecked and replayed safely
Confidence leaving applications live during planned Database Maintenance	AC has proven to be resilient at CME.

Normal Operation

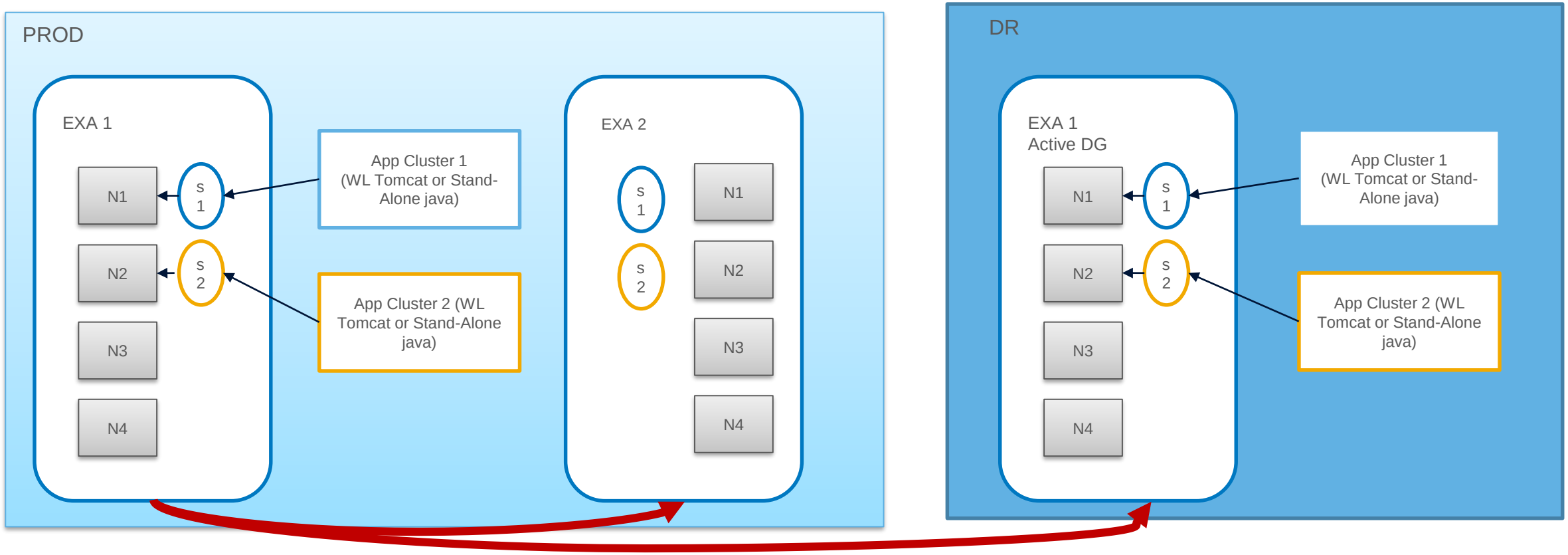
- All OLTP services configured as 1 active, rest available
- Over 400 services across environment
- Over 100 applications
- Node capacity actively managed

- Most Application Servers “Lie in Wait”
- Critical Applications are connected in a RO mode



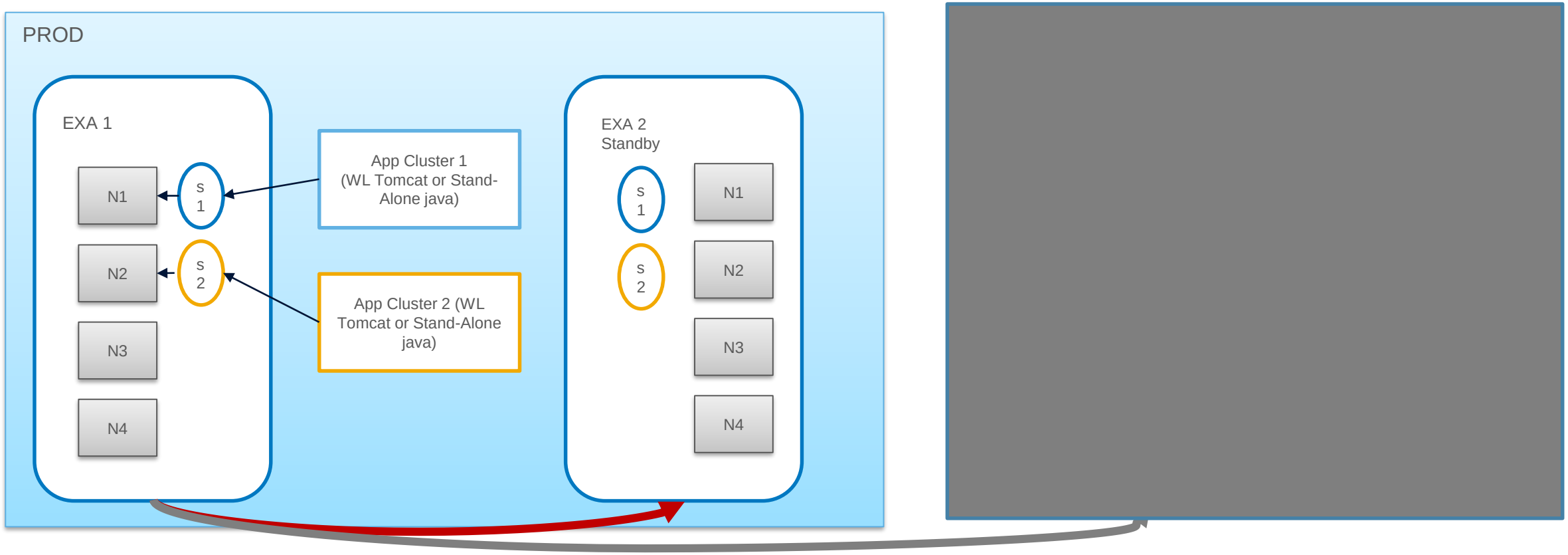
Planned Maintenance

- Exadata Full Stack Patching takes 4 hours at best
- CME does not do rolling patches (duration too long)
- AC allows apps to stay up and undergo updates while patching happens.



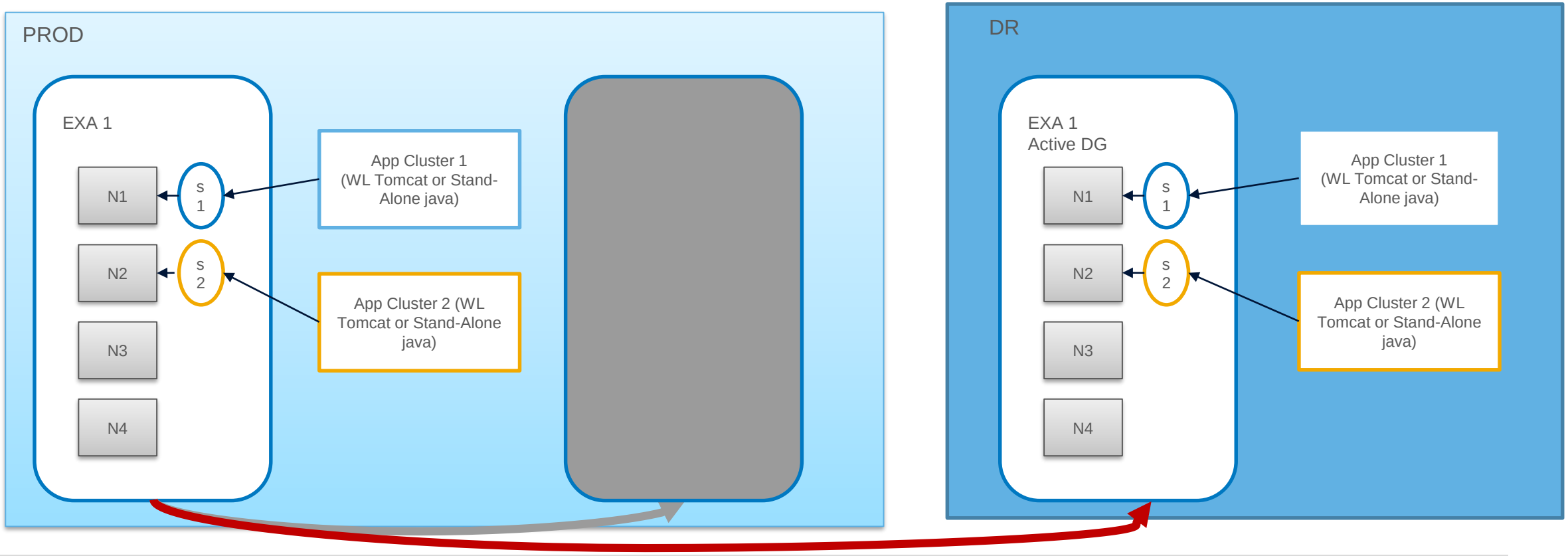
Planned Maintenance

- DR is always patched first
- Applications in DR are taken offline
- Normal change window applies
- Application changes in PROD coincide with DR patching



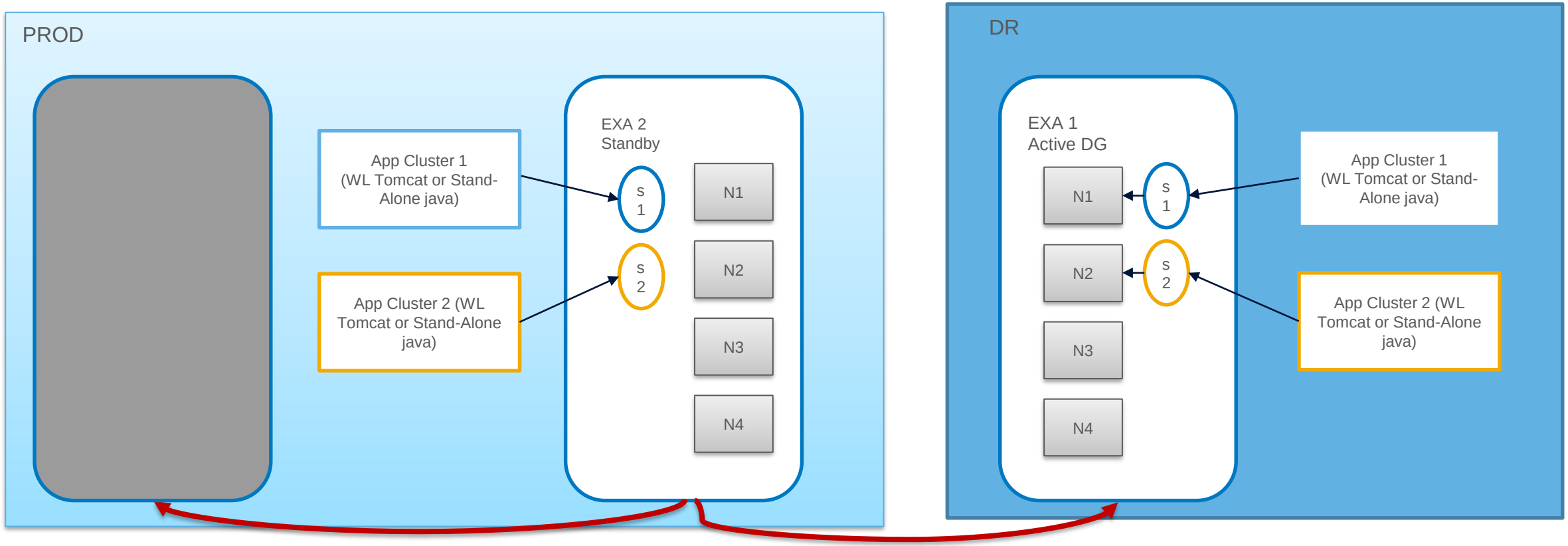
Planned Maintenance

- Local Standby databases are patched after DR
- Patching the local standby database does not impact running application
- Patched during normal maintenance window
- Application changes and testing can continue



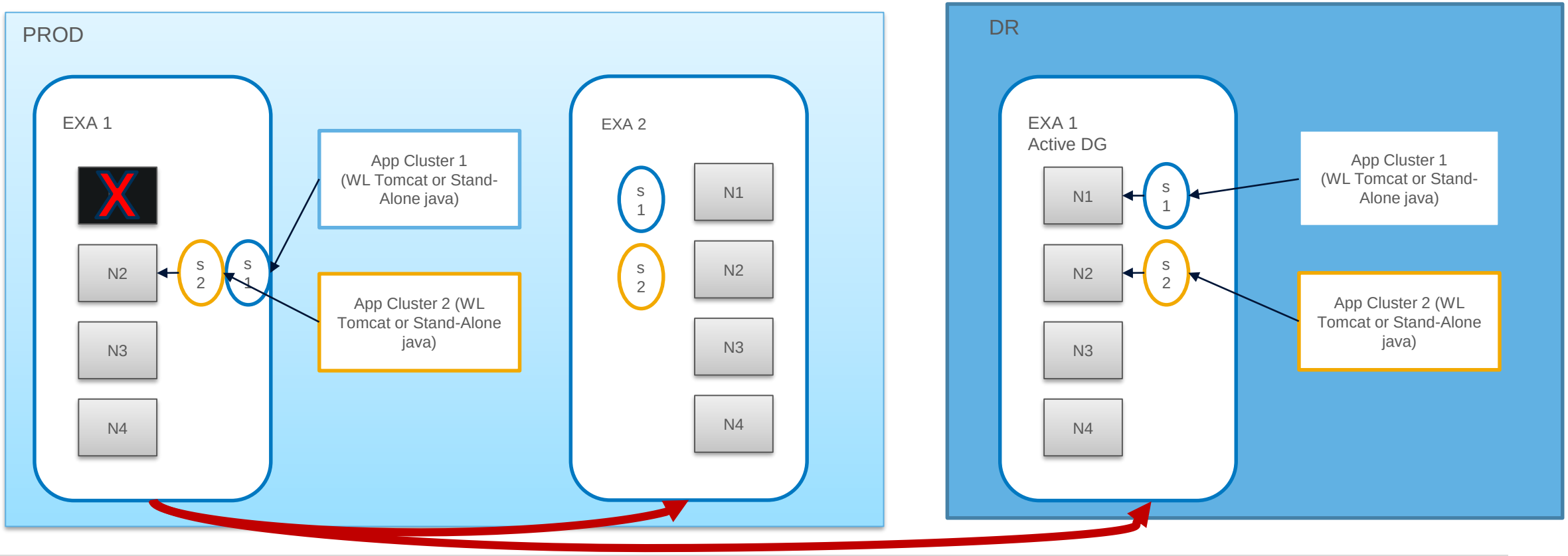
Planned Maintenance – Database As A Service

- AC compliant applications stay running and available
- Non compliant applications are stopped and restarted (Transition period)
- A database switchover is performed
- An LDAP job modifies connection strings for non compliant apps
- Non compliant apps are restarted
- Changes and testing continues during maintenance window
- Process repeated for fail back



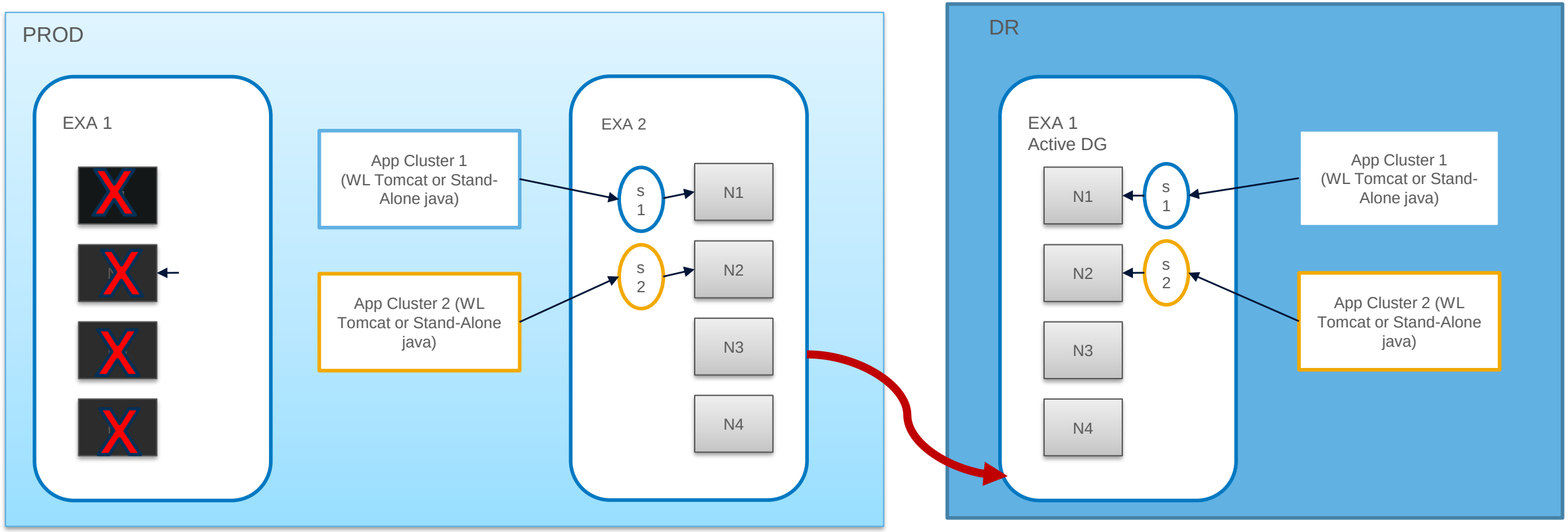
UNPLANNED OUTAGES

- Node 1 fails
- All services fail to available instance (2 illustrated)
- Application connections follow service location using Application Continuity



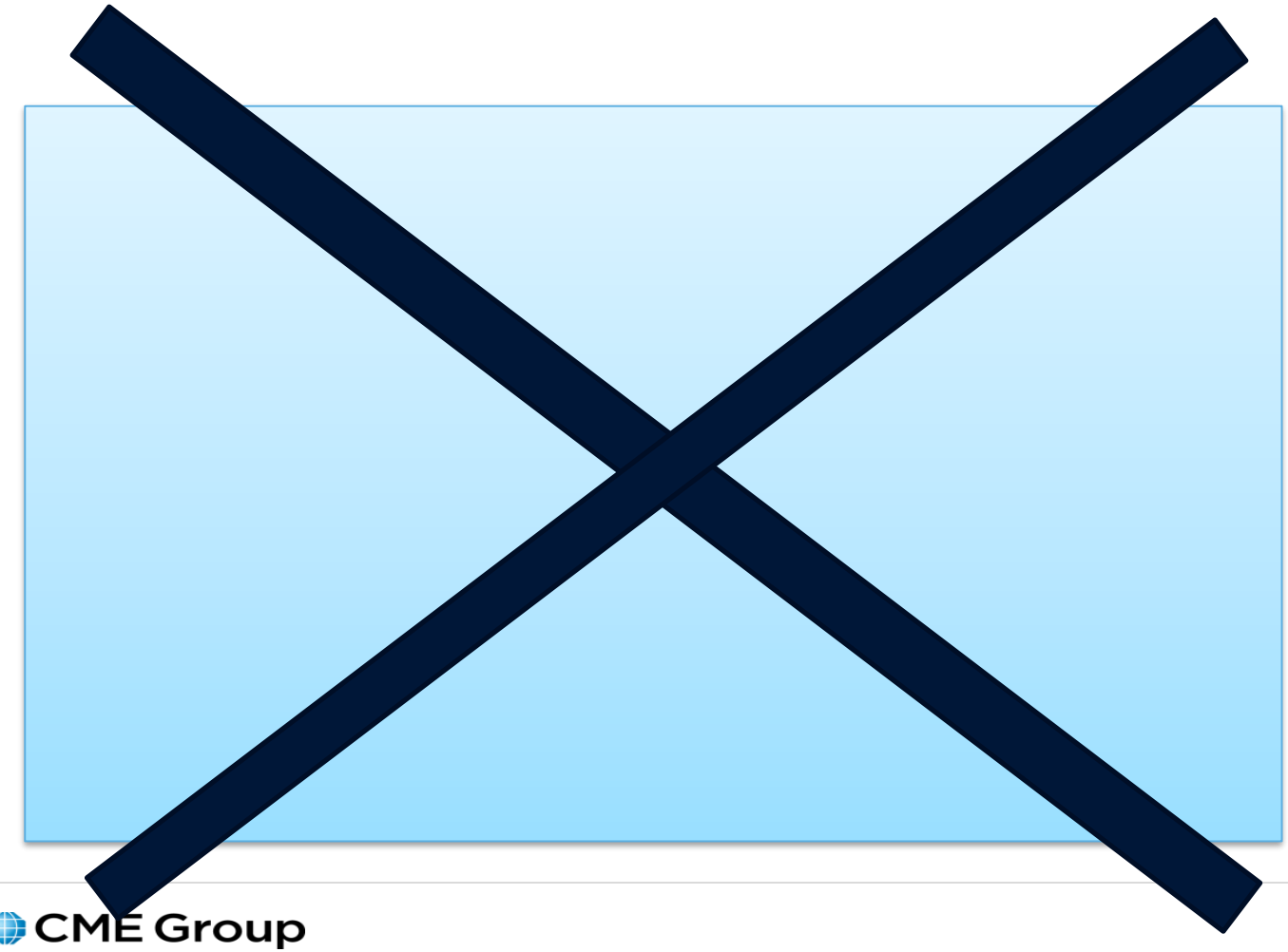
UNPLANNED OUTAGES

- What if the whole Exadata Fails?
- **At CME – this is not allowed to cause a DR event**

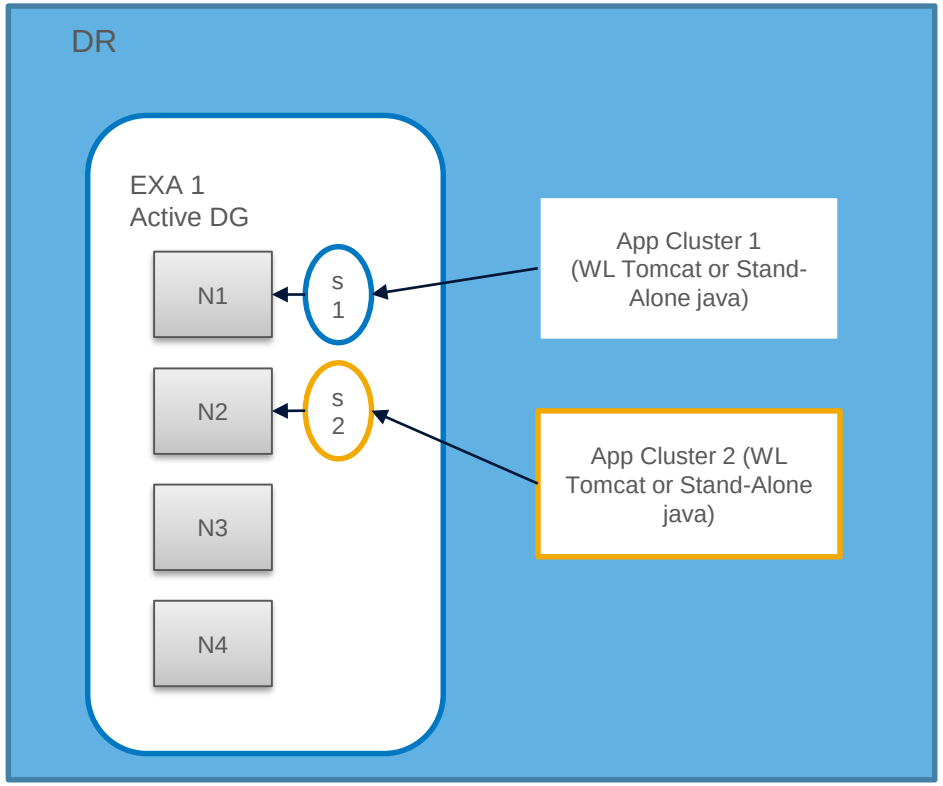


UNPLANNED OUTAGES

- Catastrophic Data Center Failure
  - Uncontrolled network outage (All HA FAILS)
  - Physical Damage to building
  - EXA 1 and EXA 2 fail in same week



- Critical Apps Up for customer RO access
- Databases are converted – Apps convert to RW
- All apps started - < 2 hours
- All automated



CME Best Practices

- Good test environment that mirrors production
- Credible HA and DR testing methodology on a mandatory interval commensurate with your change rate
- Application simulation for testing that is realistic
- Capacity Planning – keep utilization of servers <50%
- Client Interrupted using FAN and FCF
- Time Based Failover, supported by Application Continuity
- No Single Points of Failure
- Strong Change Control

Safe Harbor

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, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Statements in this presentation relating to Oracle's future plans, expectations, beliefs, intentions and prospects are "forward-looking statements" and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle's Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading "Risk Factors." These filings are available on the SEC's website or on Oracle's website at http://www.oracle.com/investor. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.