
SHAREPLEX

SharePlex Skills 101

The Future of Always-On Data Infrastructure

Bill Brunt
SW Sales Engineer

Quest
Where Next Meets Now.

Topics

- 3rd in series for Season 2
- The Challenge to Always-On Data Infrastructure
- Unsupported Datatypes
- Basic architecture for Database Migration
- High level Steps
- Activation
- Instantiation
- Catching Up: Replication Volume Planning & Strategies
- Validation
- Failing back
- Zero Downtime
- Beyond the data

Quest
Where Next Meets Now.

SharePlex Skills 101

SharePlex Skills 101 Webcast Series: On Demand | Quest

Season 2

- The Future of Always-On Data Infrastructure
  Sep 3, 2025
- Build Smarter Data Pipelines with SharePlex & AI
  Aug 6, 2025
- SharePlex Unveiled
  Jul 16, 2025

Season 1

- What's New in SharePlex
  Sep 18, 2024
- Maximizing Your SharePlex: The Why and How to Upgrade
  Nov 13, 2024
- My First 80 hours as a SharePlex Administrator
  Nov 13, 2024
- SharePlex: Out of Sync Explained
  Dec 11, 2024
- Best Practices for SharePlex
  Mar 12, 2025

Challenge: Always-On Data Infrastructure

What impacts service level agreements (SLA)?

Planned Events

- Hardware
- Operating System
- Database
  - Archive
  - Maintenance
  - Migration
- Application

Unplanned Events

- Recoverable
- Non-recoverable

Challenge: Always-On Data Infrastructure

SLA: Key Process Indicators (KPIs)

Planned Events

- Scheduled maintenance
- Exception maintenance
- Change control – failback plan
- Change success rate

Unplanned Events

- Recovery Time Objective (RTO)
- Recovery Point Objective (RPO)
- Non-recoverable event

Both, Individual or other

- Availability Compliance
- Users Impacted
- Cost per minute
- Incident Response Time
- Mean Time To Repair
- Transactions Per Second

Unsupported Data Types

There are ways…driven by Cost per Minute, Availability Compliance

1. **The tables are small – export/import during the users switching over.**

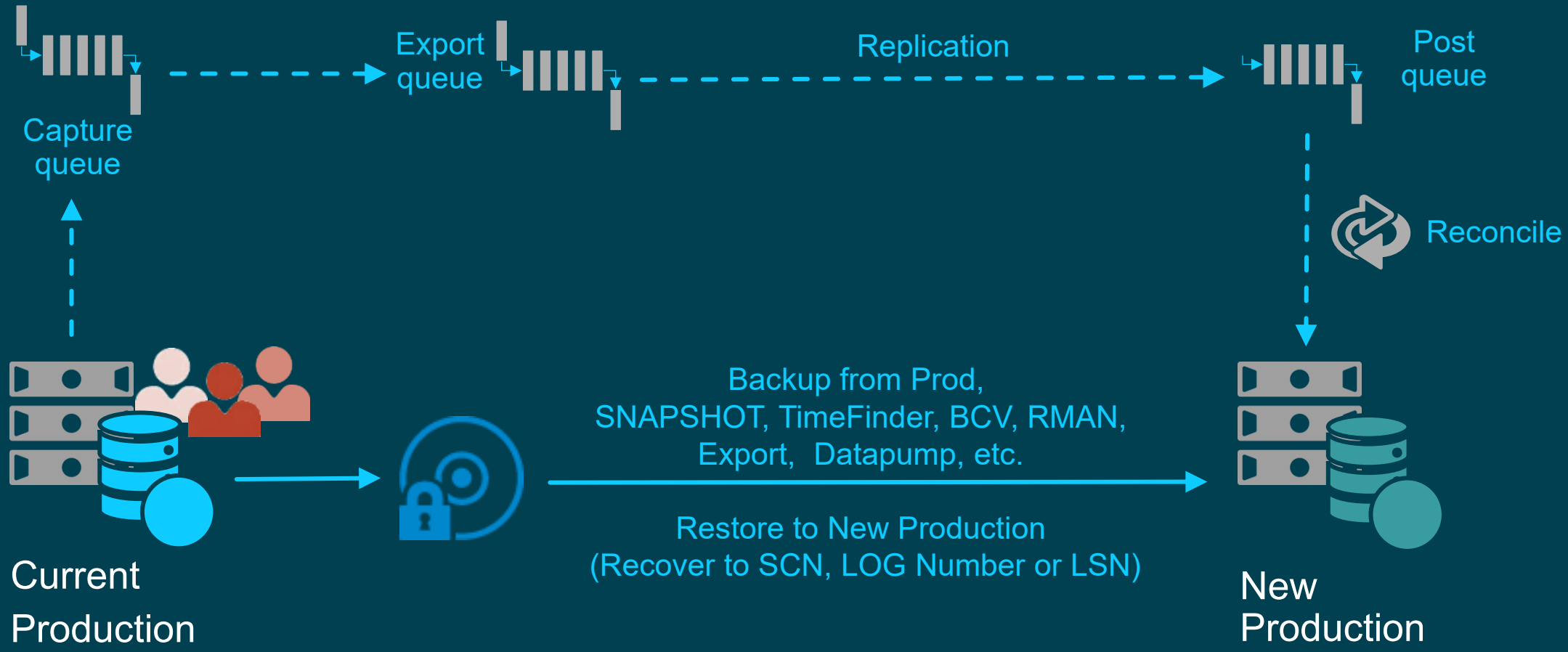
2. **Method 1: Larger unsupported tables (UST) – triggers to record row primary key.**
    1. Using vertical partitioning, plan on replicating only those columns with supported data types
    2. Time how long to export and import the table with unsupported datatypes, call it Time to Export and Import (TEI).
    3. Create and implement trigger inserts record of non-supported datatype change with table, rowid, primary key, column changed and timestamp – change table (CT).
    4. At least TEI minutes before the planned cut over, export and import the table.
    5. During switch over, use CT, for each table with unsupported datatype, create new table we'll call (Net Change Table) with primary key and unsupported datatype for rows that have changed since export started.
    6. Export NCT on source
    7. Import NCT on target
    8. Update UST with changed columns from NT.

Unsupported Data Types

There are ways…driven by Cost per Minute, Availability Compliance

1. **The tables are small – export/import during the users switching over.**
2. **Method 2: Larger unsupported tables (UST) – triggers to record row changes.**
    1. Using vertical partitioning, plan on replicating only those columns with supported data types
    2. Create trigger to convert unsupported data type to encoded supported type i.e., Nested table stored in CLOB via JSON representation.
3. **Other options…restrict transactions, etc., etc.**

Basic Architecture: Database Migration



Steps for using SharePlex for Availability

Deep Technical Content

- **Unsupported** data types
- **Activation** – starting replication
- **Instantiation** – means of creating the target database
- **Catching up** – working through the backlog created during instantiation
- **Validation** – Ensure the source and target contain the same data
- **Failback** – when the new environment does not work as expected
  i.e., in change control approval questioning “What will you do if the new system does not work?

Activation – start replication

KPI(s): Scheduled Maintenance, Exception Maintenance, Availability Compliance & Cost per minute

Activity during Activation:

- Assign Activation ID,
- Process by Object ID,
- Create Object cache,
- Insert Change Marker in replication – recreates queues.
- Brief lock on table,
- NOWAIT in applications can fail

Activation Is Required when:

- Tables added or removed
- Routes changed
- Start replicating upon activation
- Fail with DDL running,
- Failed tables not replicated

Verification:

- Syntax check
- Report error for non-existent object,
- Report error for unreachable host (not SharePlex)
- Report duplication entries for single object*
- Skipped objects and reasons

Enhancement request for datatype validation.

Activation – start replication (Cont'd)

Activation Tuning Options:

- If multiple data sources, multiple instances of sp_ctrl (beyond today's scope)
- Increasing activation threads (up to 32) controlled by:
  - Each activate command issue:
    ```
    activate config sales threads=3
    ```
  - Global across all sp_ctrl for that SharePlex instance:
    ```
    SET PARAM SP_OCF_THREAD_COUNT 10
    ```
- Multiple instances of SharePlex for same data source.

Documentation:

"The range of valid values for n is 1 to 32, but it is recommended that you use no more than 5 threads because the benefits of using threads generally diminish beyond that point."

"As an example, for a 32-CPU machine with a large disk array, 10 or more threads could show improved performance".

Best Practices:

- Test, use support if needed and cite all parts of documentation.

Instantiation – creating the target

Flexible except SCN or LSN required

Documentation is strong

- The key steps are to stop post, activate, create a backup, restore, use restored database to create target and use reconcile to the either SCN or LSN (pglsn) to start posting only the new data not in the database.

Strategies

- Getting it right once
- Break it into pieces, progressive activation, progressive instantiation, etc. by using newly introduced routes with new queues and then reconcile to the queue – a new configuration has everything currently activate plus additional.

Catchup

Current load plus capacity to work through backlog.

The effort to ensure you can catch up and process the backlog ranges from trivial to deep expertise.

If there is significant available additional capacity for higher volumes beyond the regular workload with existing configuration, there is no work.

If there is little available capacity, this can be a huge task.

Start by figuring out what your headroom is and how. Easier said than done.

Replication Volume Planning or Catchup

Variables to determine bound by time or throughput.

Variable	Expanded name	Units	Description
tth	Total Time Hours	Hour(s)	Time to replicate total volume (backlog + on going activity)
apr	Application Rate	GB / Hour	Redo rate of application
spr	SharePlex Rate	GB / Hour	Rate at which SharePlex is processing transactions
psh	Poster Stopped Hours	Hour(s)	Hour(s) Poster stopped time - how long post is stopped or estimate of maintenance activity

Total Volume

apr * tth or spr * (tth - psh)

Planning Calculations Algebra

Therefore:

```
pr*tth =  spr*(tth-psh)
apr*tth =  spr*tth - spr*psh
apr*tth - spr*tth = -spr*psh
-apr*tth + spr*tth =  spr*psh
spr*tth - apr*tth =  spr*psh
tth * (spr - apr) =  spr*psh
tth  = (spr * psh) / (spr - apr)
```

Planning for existing replication and wanting to know Total Time Hours

```
tth = ?
apr = 15 GB/hour
spr = 50 GB/hour
psh = 3 hours
```

```
tth = (50 GB/hr * 3 hr) / (50 GB/hr - 15 GB/hr)
= (150 GB) / (35 GB /hr)
= 4.285714 Hours
```

Planning for Total Time Hours and wanting to know rate.

```
spr = ?
tth = 4 hours
apr = 15 GB/hour
psh = 3 hours
```

```
apr*tth = spr*(tth-psh)
spr*(tth-psh) = apr*tth
spr = apr*tth / (tth-psh)
= (15 * 4) / (4 - 3)
= 60 / 1
```

Planning Calculations Summary

Total Time Hours `tth = (spr * psh) / (spr - apr)`

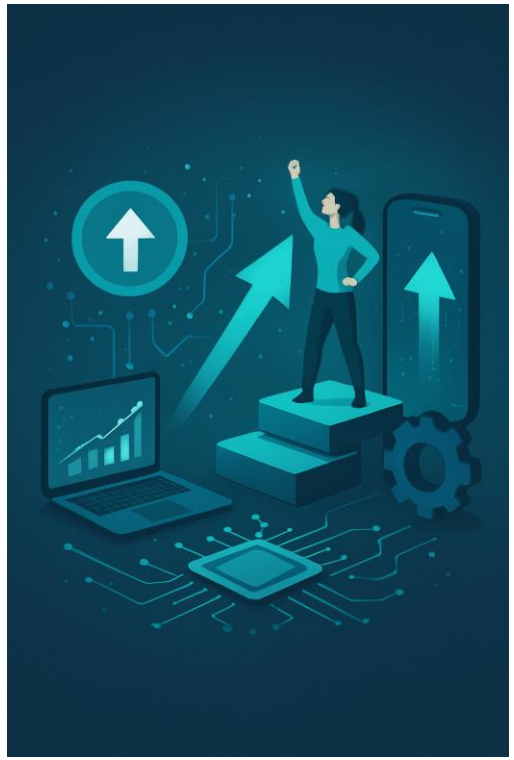
SharePlex Rate `spr = apr*tth / (tth-psh)`

Variable	Expanded name	Units	Description
tth	Total Time Hours	Hours	Time to replicate total volume (backlog + on going activity)
apr	Application Rate	GB / Hour	Redo rate of application
spr	SharePlex Rate	GB / Hour	Rate at which SharePlex is processing transactions
psh	Poster Stopped Hours	Hours	Hours Poster stopped time - how long post is stopped or estimate of maintenance activity

A glimpse into tuning

Work in progress: Many GB per hour – another SharePlex Skills 101

- Multiple cops
- Specific parameters
- Very workload dependent
- Upcoming session



```
#source tables      target tables           routing map
#splex.demo_src      splex.demo_dest         qol8spo171x02:expQ_Regular*qol8qspo172x03@o.ORCLPDB1
expand bmf1.%0      bmf1.%                  ip-172-31-82-234.ec2.internal:bmf1_0*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf1.%1      bmf1.%                  ip-172-31-82-234.ec2.internal:bmf1_1*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf1.%2      bmf1.%                  ip-172-31-82-234.ec2.internal:bmf1_2*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf1.%3      bmf1.%                  ip-172-31-82-234.ec2.internal:bmf1_3*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf1.%4      bmf1.%                  ip-172-31-82-234.ec2.internal:bmf1_4*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf1.%5      bmf1.%                  ip-172-31-82-234.ec2.internal:bmf1_5*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf1.%6      bmf1.%                  ip-172-31-82-234.ec2.internal:bmf1_6*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf1.%7      bmf1.%                  ip-172-31-82-234.ec2.internal:bmf1_7*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf1.%8      bmf1.%                  ip-172-31-82-234.ec2.internal:bmf1_8*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf1.%9      bmf1.%                  ip-172-31-82-234.ec2.internal:bmf1_9*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
shareplex@ip-172-31-82-234.ec2.internal:config$ cat ~/vardir2200/config/test_bmf2
datasource:o.ORCL

#source tables      target tables           routing map
#splex.demo_src      splex.demo_dest         qol8spo171x02:expQ_Regular*qol8qspo172x03@o.ORCLPDB1
expand bmf2.%0      bmf2.%                  ip-172-31-82-234.ec2.internal:bmf2_0*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf2.%1      bmf2.%                  ip-172-31-82-234.ec2.internal:bmf2_1*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf2.%2      bmf2.%                  ip-172-31-82-234.ec2.internal:bmf2_2*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf2.%3      bmf2.%                  ip-172-31-82-234.ec2.internal:bmf2_3*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf2.%4      bmf2.%                  ip-172-31-82-234.ec2.internal:bmf2_4*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf2.%5      bmf2.%                  ip-172-31-82-234.ec2.internal:bmf2_5*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf2.%6      bmf2.%                  ip-172-31-82-234.ec2.internal:bmf2_6*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf2.%7      bmf2.%                  ip-172-31-82-234.ec2.internal:bmf2_7*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf2.%8      bmf2.%                  ip-172-31-82-234.ec2.internal:bmf2_8*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf2.%9      bmf2.%                  ip-172-31-82-234.ec2.internal:bmf2_9*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
shareplex@ip-172-31-82-234.ec2.internal:config$ cat ~/vardir2300/config/test_bmf3
datasource:o.ORCL

#source tables      target tables           routing map
#splex.demo_src      splex.demo_dest         qol8spo171x02:expQ_Regular*qol8qspo172x03@o.ORCLPDB1
expand bmf3.%0      bmf3.%                  ip-172-31-82-234.ec2.internal:bmf3_0*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf3.%1      bmf3.%                  ip-172-31-82-234.ec2.internal:bmf3_1*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf3.%2      bmf3.%                  ip-172-31-82-234.ec2.internal:bmf3_2*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf3.%3      bmf3.%                  ip-172-31-82-234.ec2.internal:bmf3_3*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf3.%4      bmf3.%                  ip-172-31-82-234.ec2.internal:bmf3_4*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf3.%5      bmf3.%                  ip-172-31-82-234.ec2.internal:bmf3_5*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf3.%6      bmf3.%                  ip-172-31-82-234.ec2.internal:bmf3_6*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf3.%7      bmf3.%                  ip-172-31-82-234.ec2.internal:bmf3_7*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf3.%8      bmf3.%                  ip-172-31-82-234.ec2.internal:bmf3_8*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
expand bmf3.%9      bmf3.%                  ip-172-31-82-234.ec2.internal:bmf3_9*ip-10-11-72-138.us-west-2.compute.internal@o.ORCL
shareplex@ip-172-31-82-234.ec2.internal:config$
```

Be sure your system has adequate resources – can halt the under resourced

```
shareplex@ip-172-31-82-234.ec2.internal:bin$ cat spsp1.ksh
#!/usr/bin/ksh
. ~/.kshrc
sp1
echo 'SET PARAM SP_QUE_MAX_QUEUES 31' | $SP_SYS_PRODDIR/bin/sp_ctrl
echo 'SET PARAM SP_QUE_POST_SHMSIZE 2047' | $SP_SYS_PRODDIR/bin/sp_ctrl
echo 'SET PARAM SP_QUE_Q_SHMSIZE 2047' | $SP_SYS_PRODDIR/bin/sp_ctrl
echo 'SET PARAM SP_QUE_SHMDBUF 65536' | $SP_SYS_PRODDIR/bin/sp_ctrl
echo 'SET PARAM SP_SHS_SHMSIZE 2047' | $SP_SYS_PRODDIR/bin/sp_ctrl
echo 'set param SP_OCT_OLOG_RDS_MINER 1' | $SP_SYS_PRODDIR/bin/sp_ctrl
shareplex@ip-172-31-82-234.ec2.internal:bin$ cat spsp2.ksh
#!/usr/bin/ksh
. ~/.kshrc
sp2
echo 'SET PARAM SP_QUE_MAX_QUEUES 31' | $SP_SYS_PRODDIR/bin/sp_ctrl
echo 'SET PARAM SP_QUE_POST_SHMSIZE 2047' | $SP_SYS_PRODDIR/bin/sp_ctrl
echo 'SET PARAM SP_QUE_Q_SHMSIZE 2047' | $SP_SYS_PRODDIR/bin/sp_ctrl
echo 'SET PARAM SP_QUE_SHMDBUF 65536' | $SP_SYS_PRODDIR/bin/sp_ctrl
echo 'SET PARAM SP_SHS_SHMSIZE 2047' | $SP_SYS_PRODDIR/bin/sp_ctrl
echo 'NOTICE: Be sure to restart cop to put parameters in effect.'

shareplex@ip-172-31-82-234.ec2.internal:bin$ cat spsp3.ksh
#!/usr/bin/ksh
. ~/.kshrc
sp3
echo 'SET PARAM SP_QUE_MAX_QUEUES 31' | $SP_SYS_PRODDIR/bin/sp_ctrl
echo 'SET PARAM SP_QUE_POST_SHMSIZE 2047' | $SP_SYS_PRODDIR/bin/sp_ctrl
echo 'SET PARAM SP_QUE_Q_SHMSIZE 2047' | $SP_SYS_PRODDIR/bin/sp_ctrl
echo 'SET PARAM SP_QUE_SHMDBUF 65536' | $SP_SYS_PRODDIR/bin/sp_ctrl
echo 'SET PARAM SP_SHS_SHMSIZE 2047' | $SP_SYS_PRODDIR/bin/sp_ctrl
echo 'NOTICE: Be sure to restart cop to put parameters in effect.'

shareplex@ip-172-31-82-234.ec2.internal:bin$
```

Be sure your system has adequate resources – can halt the under resourced

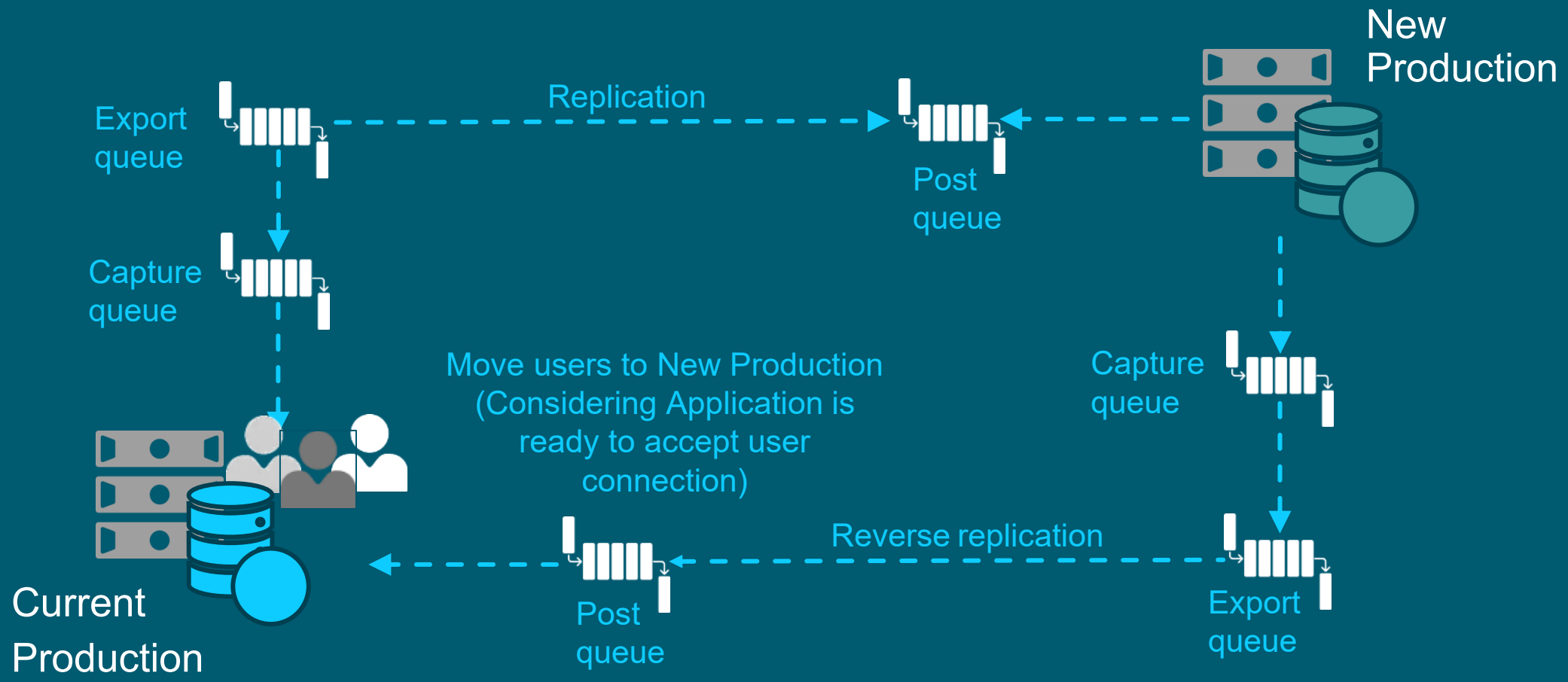
Beyond the data

Quest does comprehensive migrations

Quest

Where Next Meets Now.

Failback


New Production
Export queue
Replication
Post queue
Capture queue
Capture queue
Move users to New Production (Considering Application is ready to accept user connection)
Reverse replication
Post queue
Export queue
Current Production

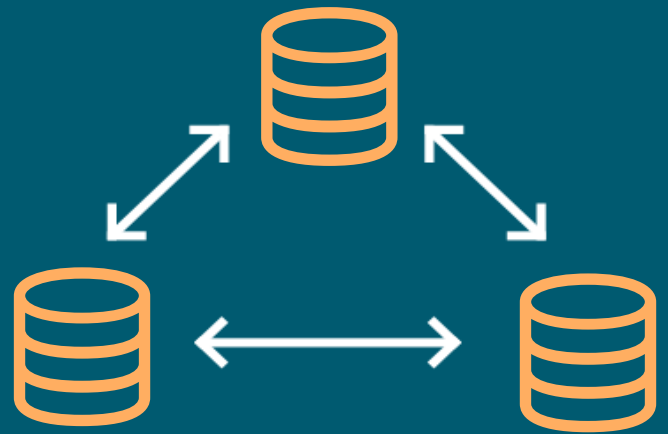
Zero Downtime

With the right tool

Active-Active Replication & **Scalability** Through **Load Balancing**

- Eliminate the database as a single point of failure
- Bi-Directional replication
- Built-in conflict resolution
- Support heterogeneous OS, Oracle/PostgreSQL databases and versions
- A node goes offline through attrition, no new connections allowed, work done, comes back online.

Data Availability



Firsthand presentation by customers

Hotwire Uses SharePlex® for AWS Cloud Migration - Database Management - Blogs - Quest Community

If you're Hotwire, you turn to SharePlex® for a complete AWS cloud migration tool and upgrade solution that really delivers zero downtime. Representatives from Hotwire recently shared their story at Oracle OpenWorld, and we have the video to prove it.

They told quite a story that involved an active-active datacenter, bidirectional replication, creating a completely new application stack, conflict resolution, upgrading the Oracle database and not losing a single transaction.

Beyond the data

Quest does comprehensive migrations

SharePlex

- Reduction in effort to migrate data, ~40%
- Dramatic reduction in downtime, usually >99%
- Failback from Postgres to Oracle
- User testing with real and recent transactions

Toad

Manages workflow

a) Code object conversion:
Toad Data Studio - Oracle to PostgreSQL Code Migration using AI

b) Identity object conversion

Erwin

Automated schema conversion

Questions?

Support

Award winning 24x7 support

Better operations

Zero downtime and data lost

Trusted partner

to global organizations with 95% satisfaction

Quest®
Where Next Meets Now.

Looking Forward to Seeing You There!

Skills 101 – SharePlex

Season 2 Episode 4

Benchmark Skills for Maximizing SharePlex Scalability

October 29th , 2025

Clay Jackson
SW Sales Engineer

www.quest.com/data-management-skills-training

Quest
Where Next Meets Now.