NoCOUG

Summer Conference 2007

Date: Thursday, August 16, 2007

Time: 1:00 PM – 2:00 PM

Venue: Chevron, Room 1220

San Ramon, CA

ORACLE®

Oracle Database 10g: Implement Streams

Daniel T. Liu
Principal Solution Architect

Agenda

- **Oracle Streams Overview**
- **Oracle Streams Process Architecture**
- **Oracle Streams Rules**

Agenda

- **Administration Tools for Oracle Streams**
- **Streams Database Configuration**
- **Streams Implementation**
- **11g Enhancements**
- **Summary**
- **Q & A**

Oracle Streams Overview

- **Oracle9i's new replication product.**
- **Similar to logical standby database**
- **Heterogeneous information sharing.**
- **Changes are captured at source or target database.**
- **Propagate information within a database or from one database to another.**

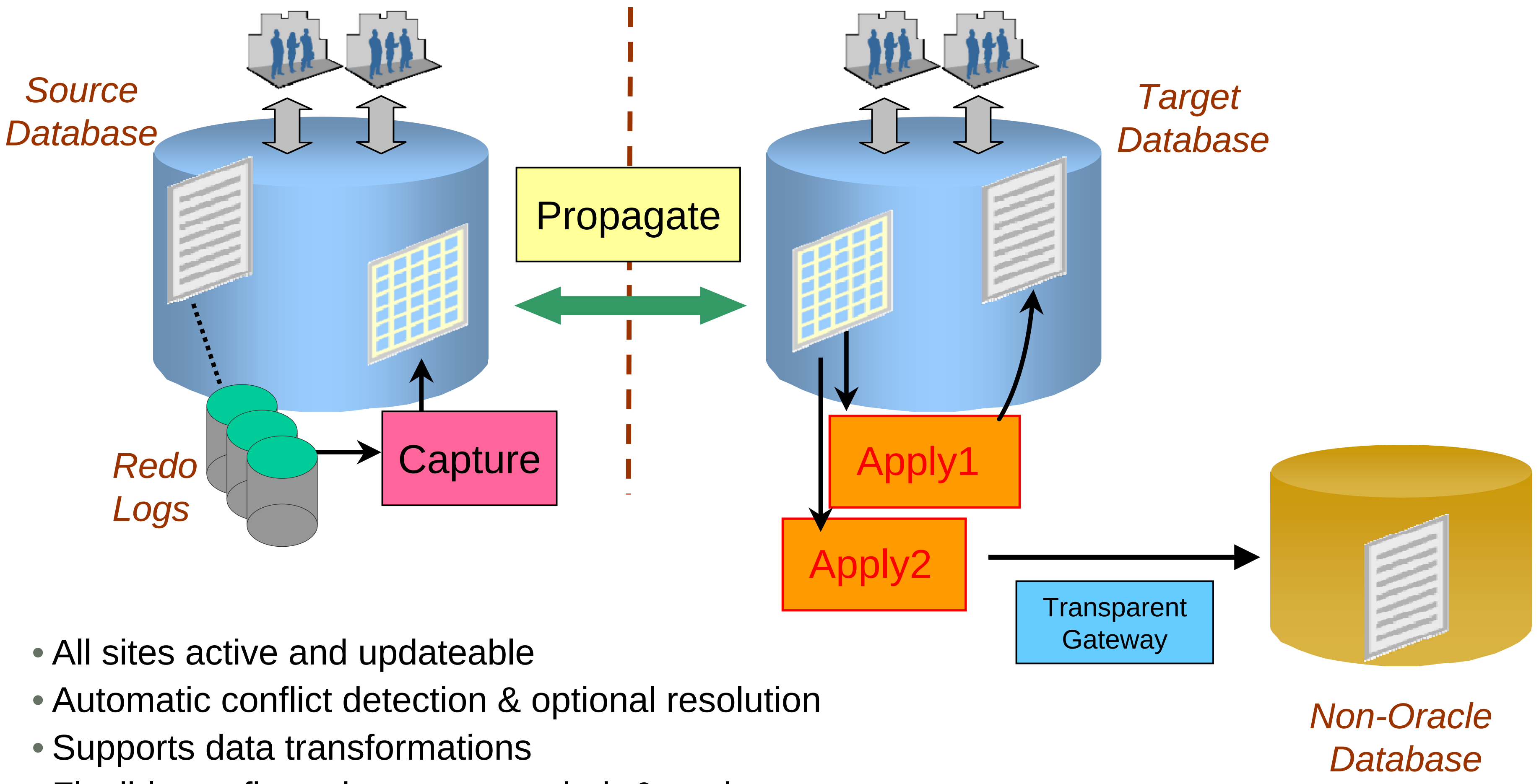
Oracle Streams Overview

- **The unit of information that is put into a stream is called an event:**
  - **DDL or DML changes, formatted as an LCR**
  - **User-created events**
- **Events are staged in and propagated between queues.**

Oracle Streams Overview

- **There are three basic tasks of a stream**
  - **Capture**
  - **Staging (place events in a queue)**
  - **Apply (consumption)**

Streams Capture & Apply



- All sites active and updateable
- Automatic conflict detection & optional resolution
- Supports data transformations
- Flexible configurations – n-way, hub & spoke, …
- Database platform / release / schema structure can differ
- Provides HA for applications where update conflicts can be avoided or managed

Oracle Streams Terminology

- **Message**
- **Queue**
- **Enqueue**
- **Dequeue**
- **Agent**

Oracle Streams Terminology

- **Subscriber**
- **Consumer**
- **Propagate**
- **Logical Change Records (LCRs)**
- **User-Created Events**

Oracle Streams Terminology

- **Message**
  - **A message is the smallest unit of information that is inserted into and retrieved from a queue**
- **Queue**
  - **A queue is a repository for messages. Queues are stored in queue tables. Each queue table is a database table and contains one or more queues. Each queue table contains a default exception queue.**
- **Enqueue**
  - **To place a message in a queue**

ORACLE

Oracle Streams Terminology

- **Dequeue**
  - **To consume a message. A single message can be processed and consumed by more than one consumer.**
- **Agent**
  - **An agent is an end user or an application that uses a queue. An agent is associated with a database user, which is assigned privileges that are required by the end user or application.**

Oracle Streams Terminology

- **Subscriber**
  - **This is a name or address that is used to identify a user or an application interested in messages in a multi-consumer queue.**
- **Consumer**
  - **A consumer is a name or address that is used to identify a user or application that dequeues or processes messages in a single consumer queue.**

Oracle Streams Terminology

- **Propagate**
  - **To transmit messages or events from one queue to another. Propagation is handled by job queue processes. Propagation to the remote queue uses database links over Oracle Net Services.**

Oracle Streams Terminology

- **Logical Change Records (LCRs)**
  - **A capture process reformats change from redo log into logical change records (LCRs). An LCR is an object with a specific format that describes a database change.**
  - **A capture process captures two types of LCRS**
    - **DML (row) LCRs**
    - **DDL LCRs**

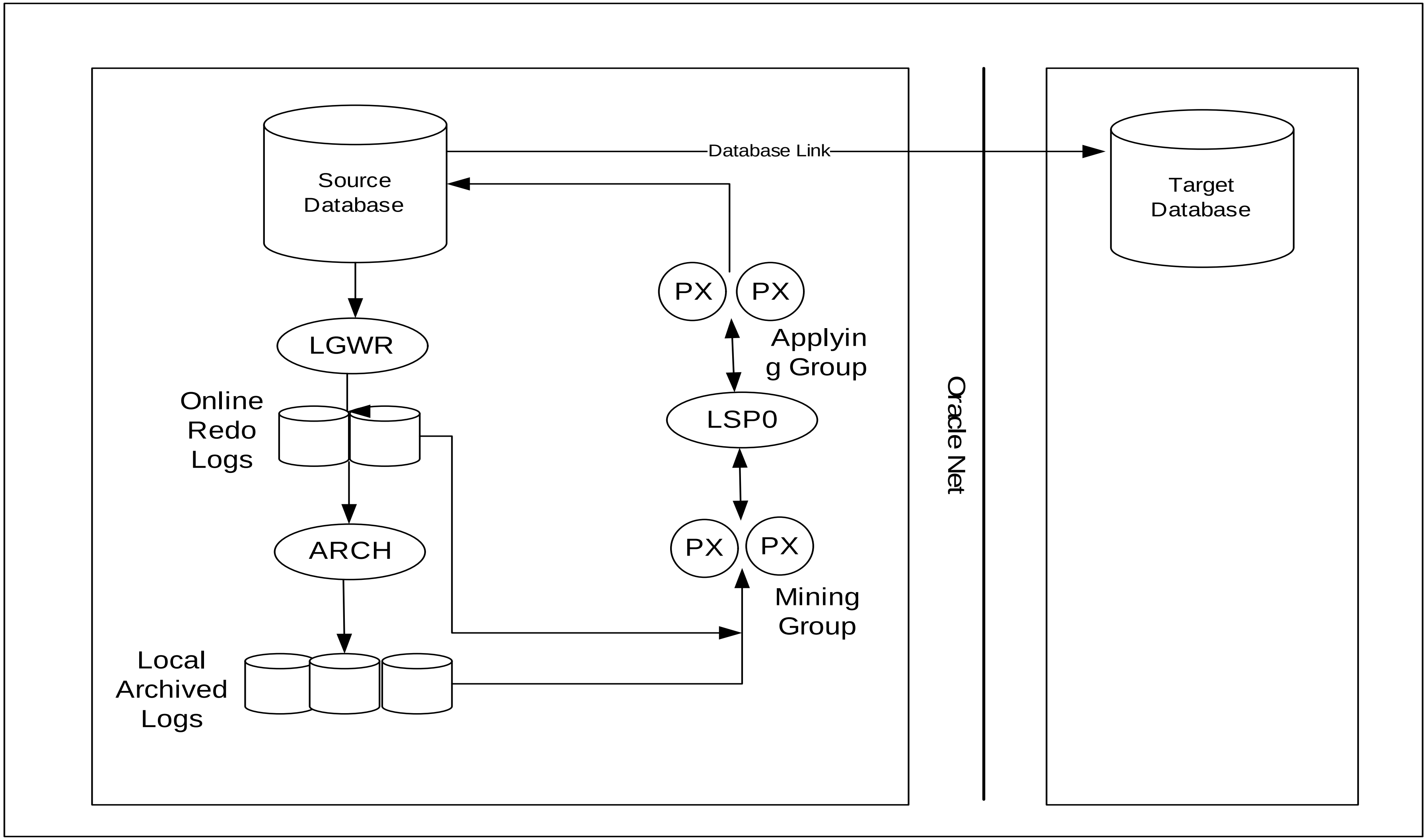
Oracle Streams Terminology

- **User-Created Events**
  - **Which are not LCRs, consist of data converted into a SYS.AnyData type event.**
  - **To enqueue these events, you can uses:**
    - **PL/SQL DBMS_* packages**
    - **Oracle Call Interface (OCI)**
    - **Oracle C++ Call Interface**
    - **Java message Service (JMS)**

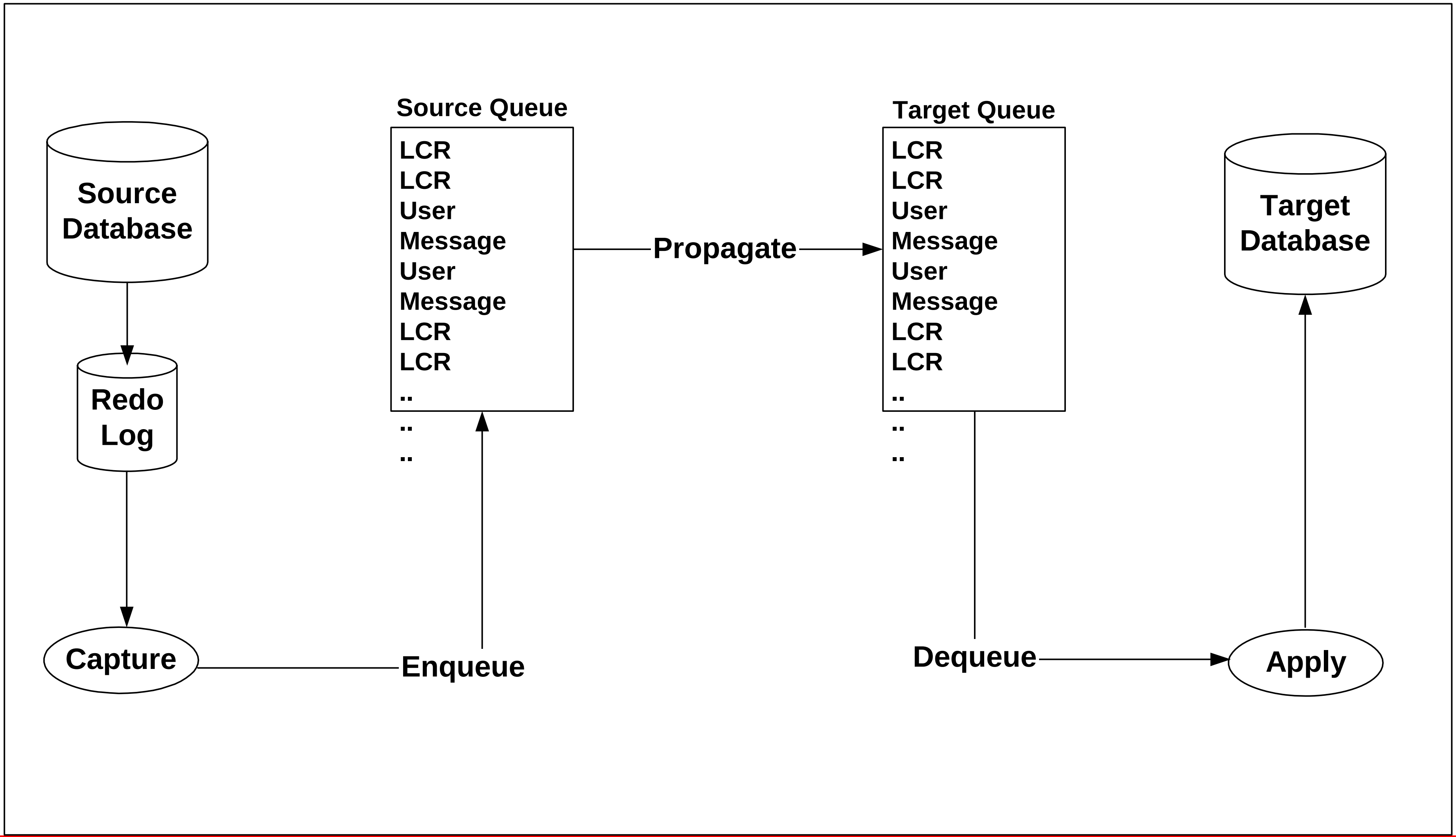
Oracle Streams Process Architecture

- Capture changes at a database.
- Enqueue events into a queue.
- Propagate events from one queue to another.
- Dequeue events.
- Apply events at a database.

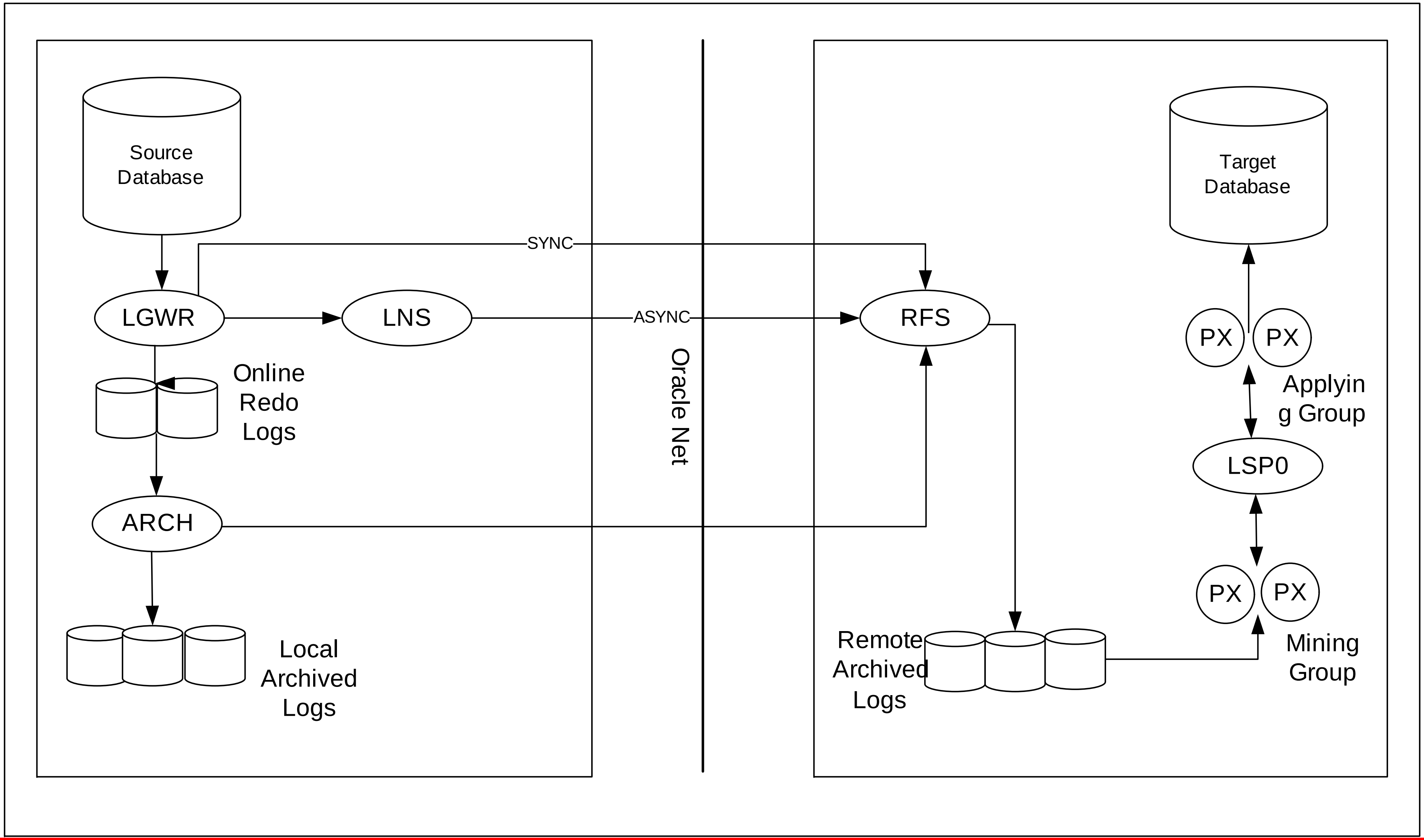
Up Streams Capture Architecture



Oracle Streams Processes Architecture



Down Streams Capture Architecture



Oracle Streams Rules

- Rules are used to control which information to share and where to share it.
- Rules can be used during capture, propagate, and apply processes.
- Rules can define in three level:
  - Table
  - Schema
  - Global

ORACLE

Oracle Streams Administration Tools

- **Oracle-Supplied PL/SQL packages**
  - **DBMS_STREAMS_ADM**
  - **DBMS_CAPTURE_ADM**
  - **DBMS_PROPAGATION_ADM**
  - **DBMS_APPLY_ADM**
- **Streams Data Dictionary views**
  - **DBA_APPLY**
  - **V$STREAMS_CAPTURE**
- **Oracle Enterprise Manager**

Database Configuration

- DB Parameters must be set at each site:
  - COMPATIBLE
  - PROCESS
  - SHARED_POOL_SIZE
  - STREAMS_POOL_SIZE
  - TIMED_STATISTICS

Database Configuration

- COMPATIBLE :

9.2.0	Oracle 9i Release 2
10.1.0	Oracle 10g Release 1
10.2.0	Oracle 10g Release 2

Database Configuration

- PROCESSES: Make sure the value of this parameter allows for all background processes, such as locks, job queue processes, and parallel execution processes. In Streams, capture processes and apply processes use background processes and parallel execution processes, and propagation jobs use job queue processes.

Database Configuration

- SHARED_POOL_SIZE : Each capture process needs 10MB of shared pool space, but Streams is limited to using a maximum of 10% of the shared pool. So shared_pool_size has to be set to at least 100MB. If you wish to run multiple capture processes, then this parameter needs to be set to an even higher value.

Database Configuration

- STREAMS_POOL_SIZE: For a database that has a capture process, you should set this parameter to 100 MB initially and then tune upward as needed. In addition, the Streams pool is used for internal communications during parallel capture and apply.

Database Configuration

- STREAMS_POOL_SIZE: You should increase the size of the Streams pool for each of the following factors:
  - 10 MB for each capture process parallelism
  - 1 MB for each apply process parallelism
  - 10 MB or more for each queue staging captured events
  - For example, if parallelism is set to 3 for a capture process, then increase the Streams pool by 30 MB. If parallelism is set to 5 for an apply process, then increase the Streams pool by 5 MB.

Database Configuration

- TIMED_STATISTICS If you want to collect elapsed time statistics in the data dictionary views related to Streams, then set this parameter to true. The views that include elapsed time statistics include:
  - V$STREAMS_CAPTURE
  - V$STREAMS_APPLY_COORDINATOR
  - V$STREAMS_APPLY_READER
  - V$STREAMS_APPLY_SERVER.

ORACLE

Database Configuration

- DB Parameters must be set at source site:
  - LOG_ARCHIVE_DEST_n
  - LOG_ARCHIVE_DEST_STATE_n
  - REMOTE_ARCHIVE_ENABLE
  - UNDO_RETENTION

Database Configuration

- LOG_ARCHIVE_DEST_n
  - To use downstream capture and copy the redo log files to the downstream database using log transport services, at least one log archive destination must be at the site running the downstream capture process. Defines up to ten log archive destinations, where n is 1, 2, 3, ... 10

Database Configuration

- LOG_ARCHIVE_DEST_STATE_n
  - To use downstream capture and copy the redo log files to the downstream database using log transport services, make sure the destination that corresponds to the LOG_ARCHIVE_DEST_n destination for the downstream database is set to enable. The parameter suffix (1 through 10) specifies one of the ten corresponding LOG_ARCHIVE_DEST_n destination parameters.

Database Configuration

- REMOTE_ARCHIVE_ENABLE
  - Enables or disables the sending of redo archival to remote destinations and the receipt of remotely archived redo. To use downstream capture and copy the redo log files to the downstream database using log transport services, this parameter must be set to true at both the source database and the downstream database.

Database Configuration

- UNDO_RETENTION >900 Specifies (in seconds) the amount of committed undo information to retain in the database. For a database running one or more capture processes, make sure this parameter is set to specify an adequate undo retention period. If you are running one or more capture processes and you are unsure about the proper setting, then try setting this parameter to at least 3600. If you encounter "snapshot too old" errors, then increase the setting for this parameter until these errors cease. Make sure the undo tablespace has enough space to accommodate the UNDO_RETENTION setting.

Database Configuration

- DB Parameters that are specific to propagation:
  - GLOBAL_NAMES
  - JOB_QUEUE_PROCESSES

Database Configuration

- GLOBAL_NAMES
  - This parameter has to be set to TRUE at each database if you want to use Streams to share information between databases.
  - This parameter requires a database link name to match the name of the host to which it connects.

Database Configuration

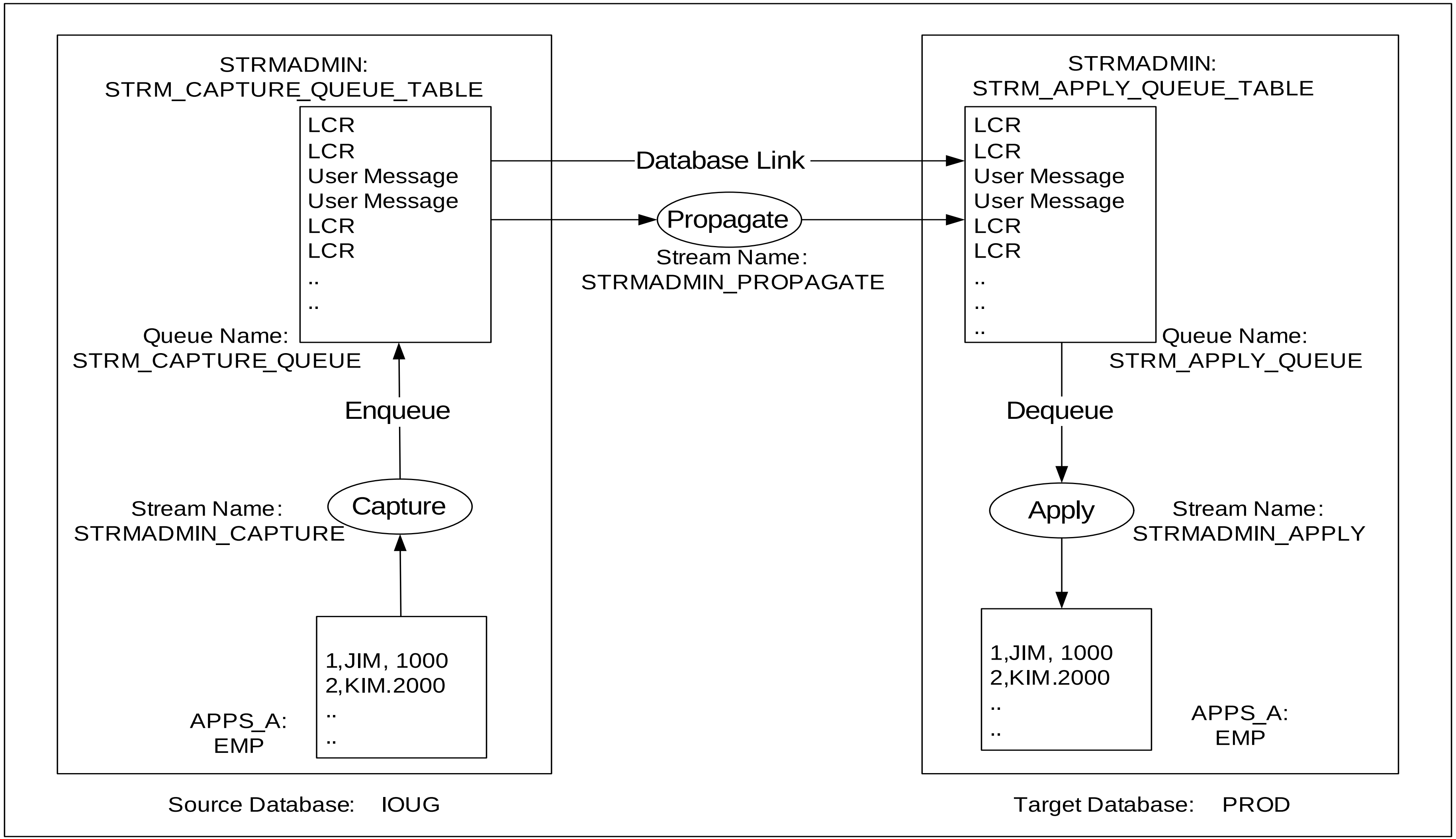
- JOB_QUEUE_PROCESSES : This parameter specifies the number of processes that can handle requests created by DBMS_JOB or DBMS_SCHEDULER packages. Ensure that it is set to 2 or higher. The default value of 10 is acceptable for most systems.

Database Configuration

- Only databases that are capturing changes via Streams capture process must be in archived mode
- The databases that are applying changes via Streams apply process will not necessary be in archived mode

ORACLE

Streams Implementation Overview



Destination Database Setup

- Step 1: Create Streams Administrator
- Step 2: Grant the necessary privileges to the Streams Administrator
- Step 3: Create streams apply queue
- Step 4: Add apply rules for the table at the destination database

Destination Database Setup

- Step 5: Specify an 'APPLY USER' at the destination database
- Step 6: Set Apply Process options
- Step 7: Start the Apply Process

Step 1: Create Streams Administrator

```
connect SYS/password as SYSDBA

create user STRMADMIN identified by STRMADMIN;
```

Step 2: Grant Necessary Privileges to Streams Administrator

- In order to create capture and apply processes, the Streams Administrator must have DBA privilege. This privilege must be explicitly granted to the Streams Administrator.

```
grant CONNECT, DBA, IMP_FULL_DATABASE,
  EXP_FULL_DATABASE to  "STRMADMIN";


DBMS_STREAMS_AUTH.GRANT_ADMIN_PRIVILEGE('STRMA
  DMIN');
```

Step 3: Create Streams Apply Queue

```
connect STRMADMIN/STRMADMIN


BEGIN
 DBMS_STREAMS_ADM.SET_UP_QUEUE(
  queue_table => ‘STRM_APPLY_QUEUE_TABLE’,
  queue_name => ‘STRM_APPLY_QUEUE’,
  queue_user => ‘STRMADMIN’);
END;
/
```

Step 4: Add Apply Rules for the Table at the Destination Database

```
BEGIN
DBMS_STREAMS_ADM.ADD_TABLE_RULES(
  table_name => ‘APPS_A.EMP',
  streams_type => 'APPLY',
  streams_name => 'STRMADMIN_APPLY',
  queue_name => 'STRMADMIN.STRM_APPLY_QUEUE‘,
  include_dml => true,
  include_ddl => true,
  source_database => ‘ioug.world');
END;
/
```

Step 5: Specify an 'APPLY USER' at the Destination Database

```
BEGIN
DBMS_APPLY_ADM.ALTER_APPLY
(
  apply_name => 'STRMADMIN_APPLY',
  apply_user => ‘APPS_A‘
);
END;
/
```

Step 6: Set Apply Process Options

- If you do not wish the apply process to abort for every error that it encounters, you can set the below parameter.
  - Y: (Default Value) apply process would abort due to any error.
  - N: apply process will continue, but the error details would be logged in DBA_APPLY_ERROR.

```
BEGIN
DBMS_APPLY_ADM.SET_PARAMETER
( apply_name => 'STRMADMIN_APPLY',
  parameter => 'DISABLE_ON_ERROR',
  value => 'N' );
END;
/
```

Step 7: Start the Apply Process

```
BEGIN
DBMS_APPLY_ADM.START_APPLY
(
  apply_name => 'STRMADMIN_APPLY‘
);
END;
/
```

Source Database Setup

- Step 1: Move LogMiner tables from SYSTEM/SYSAUX tablespace
- Step 2: Turn on supplemental logging
- Step 3: Create Streams Administrator
- Step 4: Grant the necessary privileges
- Step 5: Create a database link to the destination database
- Step 6: Create streams queue

ORACLE

Source Database Setup

- Step 7: Add capture rules for the table at the source database
- Step 8: Add propagation rules for the table at the source database
- Step 9: Instantiation of tables from Source to Destination Database
- Step 10: Start the Capture Process

Step 1: Move LogMiner tables from SYSTEM/SYSAUX TABLESPACE

- By default, all LogMiner tables are created in the SYSTEM or SYSAUX tablespace. It is a good practice to create an alternate tablespace for the LogMiner tables.

```
CREATE TABLESPACE LOGMNRTS DATAFILE 'logmnrtbs.dbf'
  SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED;
BEGIN DBMS_LOGMNR_D.SET_TABLESPACE('LOGMNRTS');
END;
/
```

Step 2: Turn on Supplemental Logging

- Turn on supplemental logging for source database or table

```
connect SYS/password as SYSDBA

ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (primary key,
  unique, foreign key) COLUMNS;

ALTER TABLE apps_a.emp ADD SUPPLEMENTAL LOG GROUP
  pk_emp (id) ALWAYS;
```

Step 3: Create Streams Administrator

```
connect SYS/password as SYSDBA

create user STRMADMIN identified by STRMADMIN;
```

Step 4: Grant necessary privileges to Streams Administrator

- In order to create capture and apply processes, the Streams Administrator must have DBA privilege. This privilege must be explicitly granted to the Streams Administrator.

```
grant CONNECT, DBA, IMP_FULL_DATABASE,
  EXP_FULL_DATABASE to  "STRMADMIN";


DBMS_STREAMS_AUTH.GRANT_ADMIN_PRIVILEGE('STRMA
  DMIN');
```

Step 5: Create a Database Link to the Destination Database

```
connect STRMADMIN/STRMADMIN

CREATE DATABASE LINK PROD
    CONNECT TO STRMADMIN
    IDENTIFIED BY STRMADMIN
    USING 'prod'
/
```

Step 6: Create Streams Capture Queue

```
connect STRMADMIN/STRMADMIN
BEGIN
 DBMS_STREAMS_ADM.SET_UP_QUEUE
(
  queue_table => 'STRM_CAPTURE_QUEUE_TABLE',
  queue_name => ‘STRM_CAPTURE_QUEUE',
  queue_user => 'STRMADMIN‘
 );
END;
/
```

Step 7: Add Capture Rules for the Table at the Source Database

```
BEGIN
DBMS_STREAMS_ADM.ADD_TABLE_RULES(
  table_name => ‘APPS_A.EMP',
  streams_type => ‘CAPTURE',
  streams_name => 'STRMADMIN_CAPTURE',
  queue_name => 'STRMADMIN.STRM_CAPTURE_QUEUE',
  include_dml => true,
  include_ddl => true,
  source_database => ‘ioug.world’);
END;
/
```

Step 8: Add Propagation Rules for the Table at the Source Database

```
BEGIN
DBMS_STREAMS_ADM.ADD_TABLE_PROPAGATION_RULES(
 table_name => ‘APPS_A.EMP',
 streams_name => 'STRMADMIN_PROPAGATE',
 source_queue_name => 'STRMADMIN.STRM_CAPTURE_QUEUE',
 destination_queue_name => 'STRMADMIN.STRM_APPLY_QUEUE@PROD',
 include_dml => true,
 include_ddl => true,
 source_database => ‘IOUG.WORLD');
END;
/
```

Step 9: Instantiation of Tables from Source to Destination Database

- When data replication is setup for an object, the source object needs to be instantiated.
- The instantiation process is the method used for creating a copy of the source object at the destination database and setting up the appropriate SCN (System Change Number) for replication.

ORACLE

Step 9: Instantiation of Tables from Source to Destination Database

- If the objects are not present in the destination database:
  - Export from the Source Database

```
exp USERID=APPS_A TABLES=EMP FILE=emp.dmp GRANTS=Y
  ROWS=Y LOG=exp.log OBJECT_CONSISTENT=Y INDEXES=Y
  STATISTICS = NONE
```

  - Import into the Destination Database

```
imp USERID=APPS_A FULL=Y CONSTRAINTS=Y FILE=emp.dmp
  IGNORE=Y GRANTS=Y ROWS=Y COMMIT=Y LOG=imp.log
  STREAMS_INSTANTIATION=Y
```

Step 9: Instantiation of tables from Source to Destination Database

- If the objects are already present in the destination database:
  Method 1: By means of Metadata-only export/import

  - **Export from the Source Database**
    `exp USERID=APPS_A TABLES=emp FILE=emp.dmp GRANTS=Y` **`ROWS=N`**
    `LOG=exp.log` **`OBJECT_CONSISTENT=Y`** `INDEXES=Y STATISTICS = NONE`
  - **Import into the Destination Database**
    `imp USERID=APPS_A FULL=Y CONSTRAINTS=Y FILE=emp.dmp IGNORE=Y`
    `GRANTS=Y ROWS=Y COMMIT=Y LOG=imp.log` **`STREAMS_INSTANTIATION=Y`**

Step 9: Instantiation of tables from Source to Destination Database

- If the objects are already present in the destination database:
  Method 2: By manually instantiating the objects

  - Get the Instantiation SCN at the source database

```
connect STRMADMIN/STRMADMIN@ioug
set serveroutput on
DECLARE iscn NUMBER; -- Variable to hold instantiation SCN

BEGIN
  iscn := DBMS_FLASHBACK.GET_SYSTEM_CHANGE_NUMBER();
  DBMS_OUTPUT.PUT_LINE ('Instantiation SCN is: ' || iscn);
END;
```

Step 9: Instantiation of tables from Source to Destination Database

- Instantiate the objects at the destination database with this SCN value

```
connect STRMADMIN/STRMADMIN@prod

BEGIN DBMS_APPLY_ADM.SET_TABLE_INSTANTIATION_SCN(
  source_object_name => ‘APPS_A.EMP',
  source_database_name => ‘IOUG.WORLD',
  instantiation_scn => &iscn
  );
END;
```

ORACLE

Step 10: Start the Capture Process

```
connect STRMADMIN/STRMADMIN@IOUG

BEGIN
 DBMS_CAPTURE_ADM.START_CAPTURE(
 capture_name => 'STRMADMIN_CAPTURE');
END;
/
```

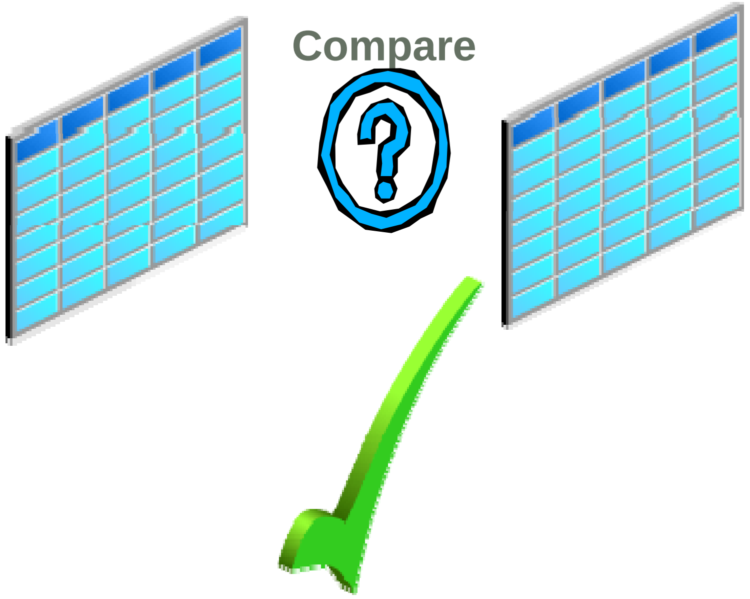
Streams 11g Enhancements

- Additional Data Type Support
- Table data comparison
- Synchronous capture
- Manageability & Diagnosibility improvements
- Performance improvements
- Streams AQ Enhancements
- More …

Newly Supported Datatypes

- **XMLType**
  - Storage CLOB
- **Transparent Data Encryption (TDE)**
  - Default: Capture TDE=> Apply TDE
  - PRESERVE_ENCRYPTION apply parameter controls behaviour when destination columns are not encrypted

Table Data Comparison



- Compare data between live sources
  - Compare 11.1 with 10.1, 10.2 or 11.1
- Recheck
  - In-flight data
  - Rows that are different
- Converge feature
  - Identify “truth” database (local or remote) for row diffs

```
DBMS_COMPARISON
```

Split and Merge of Streams

- Maintains high performance for all replicas
- Automated, fast “catch-up” for unavailable replica

- **Split_streams** (DBMS_STREAMS_ADM)
  - Use for slow/unavailable subscriber
  - Clones queue, capture, propagation
  - Threshold for automatic merge
- **Merge_streams**
  - Merge cloned stream back to original configuration

ORACLE

Split and Merge of Streams

Challenge

- With hub&spoke configurations, when one destination is unavailable, queue spill impacts performance for other destinations

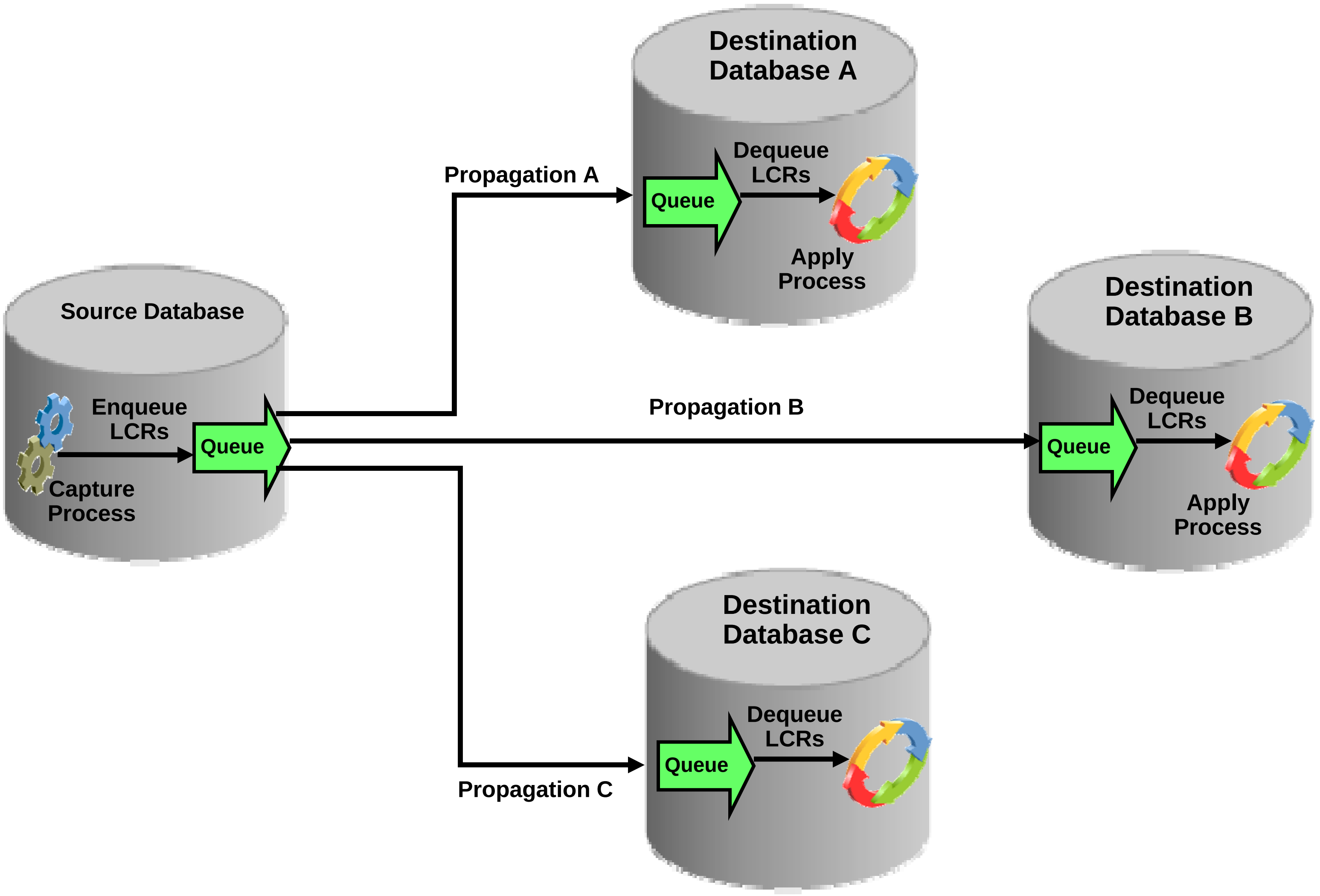
Solution

- Split the queue between live and down destinations
- Merge queues after recovery

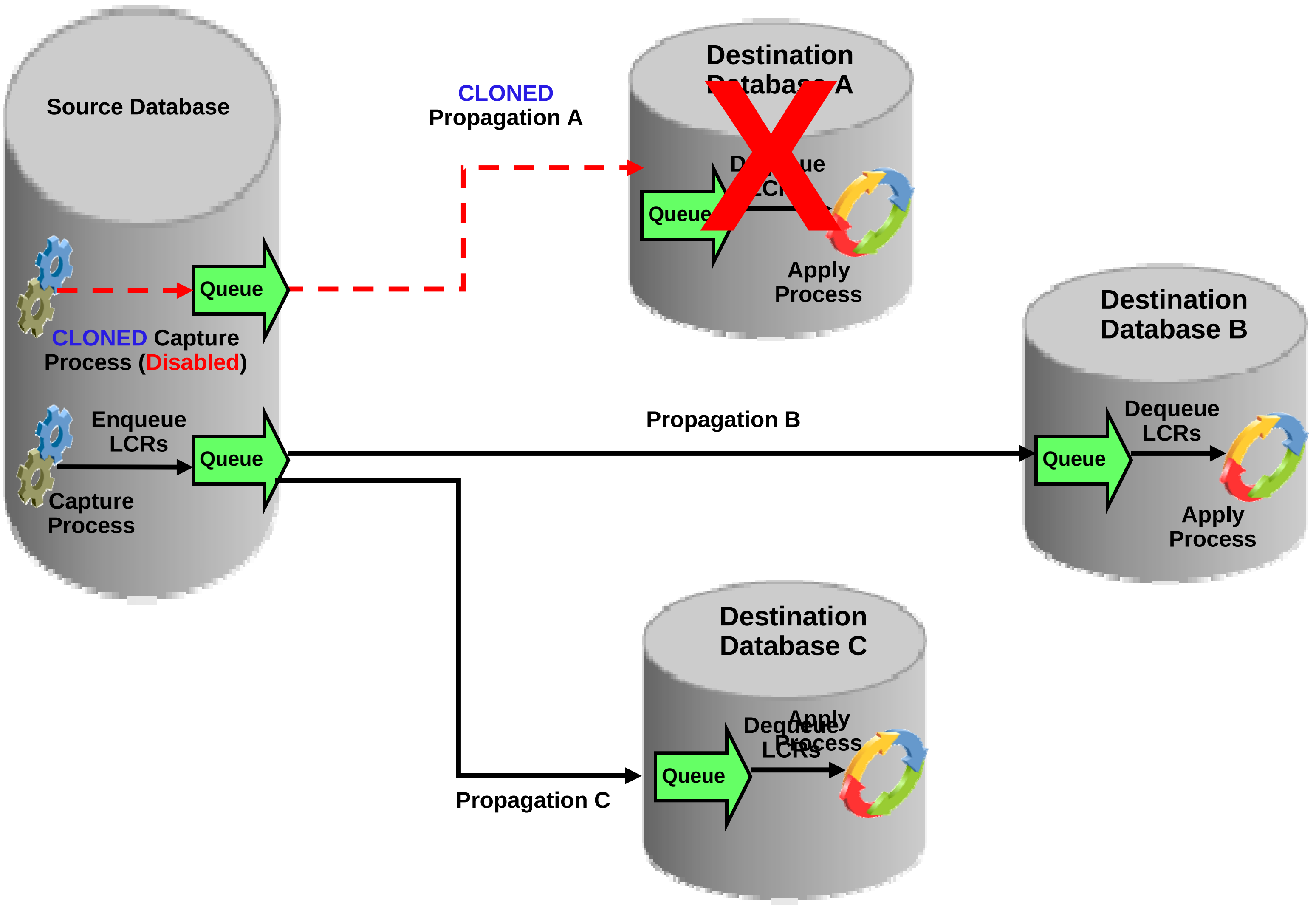
➢ Maintains high performance for all replicas

➢ Automated, fast “catch-up” for unavailable replica

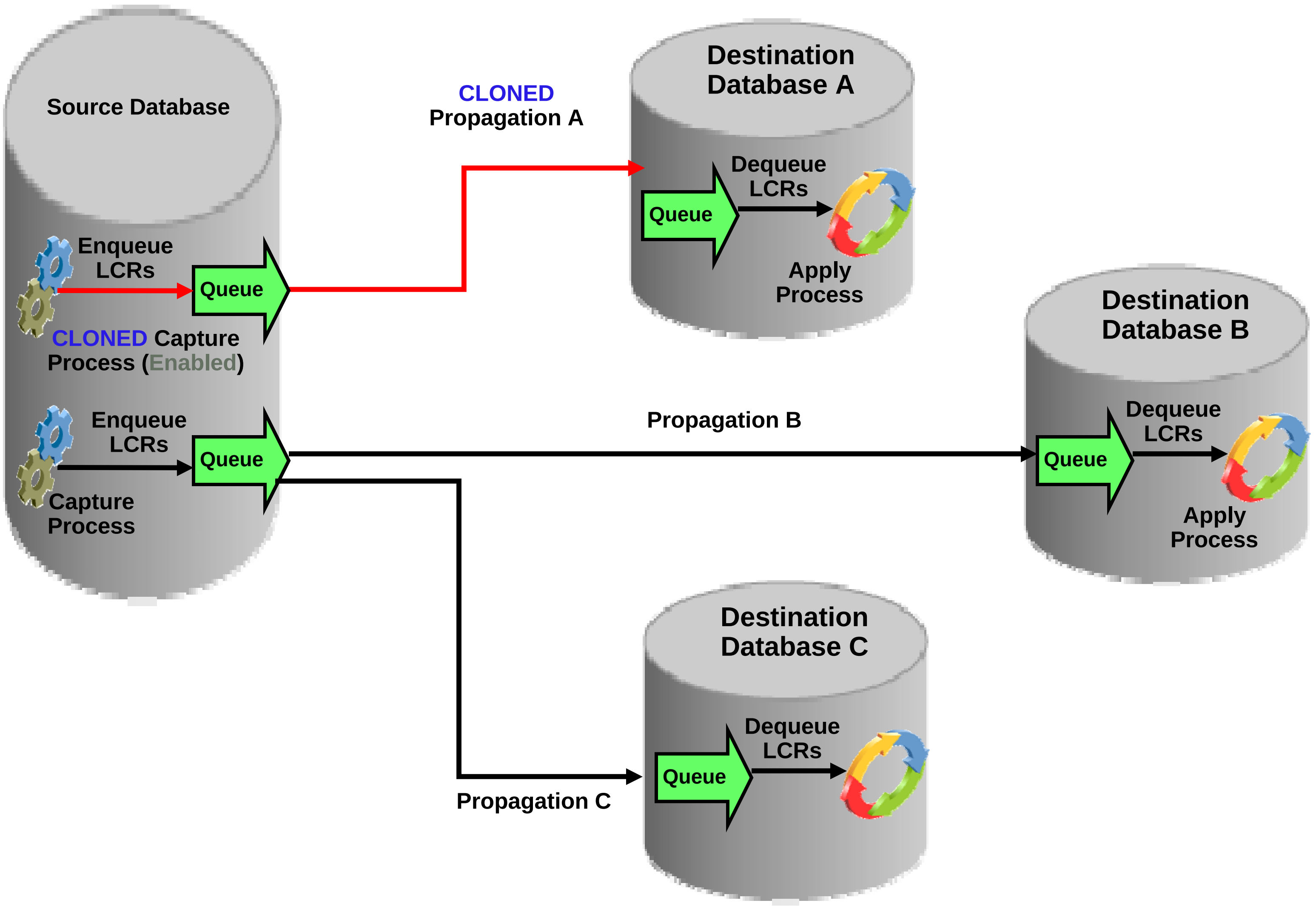
Streams: Hub with 3 Spokes



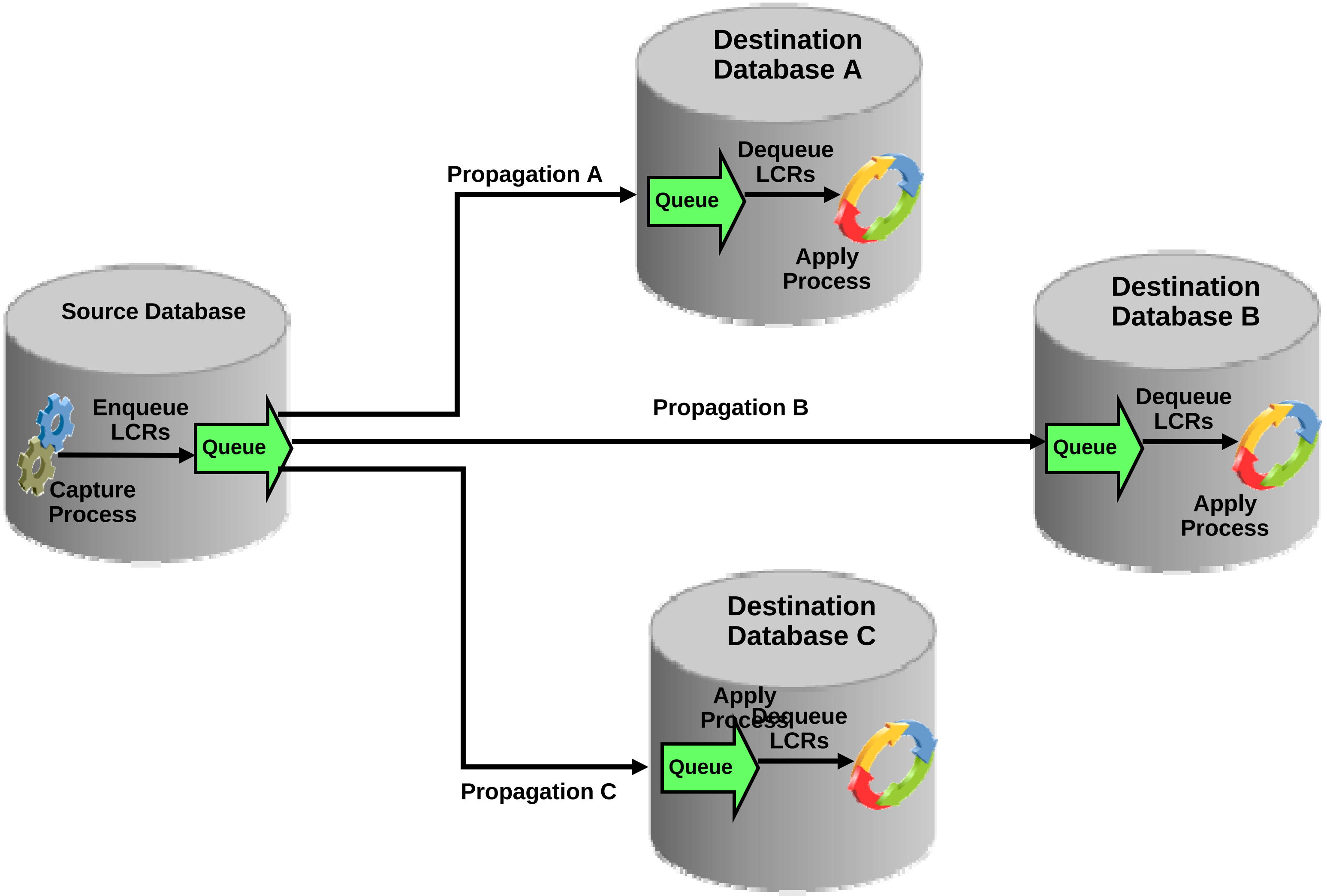
Split Streams: Site A Unavailable



Split Streams: Site A Available



Merge Streams: Original Configuration



Oracle Streams Summary

- Oracle Streams Terminology
- Oracle Streams Process Architecture
- Oracle Streams Rules
- Administration Tools for Oracle Streams
- Streams Database Configuration
- Streams Implementation
- 11g Enhancements

References

Oracle Database 10g New Features, Ault, Liu and Tumma; Rampant Techpress;
Oracle Database 10g Documentation Library;
Oracle Metalink Support;
Implementing Oracle9i Data Guard for Higher Availability, Daniel T. Liu; DBAZine;
Top DBA Shell Scripts for Monitoring Database, Daniel T. Liu; DBAZine;

I would also like to acknowledge the assistance of Larry Bailey, Husam Tomeh of FARES, Tammy Bednar, Larry Carpenter, Sushil Kumar, Patricia McElroy and Schwinn Ulrike of Oracle Corporation.

Thanks For Coming !!

Daniel Liu Contact Information

Phone: (714) 376-8416

Email: daniel.liu@oracle.com

Email: daniel_t_liu@yahoo.com

Company Web Site:

http://www.oracle.com