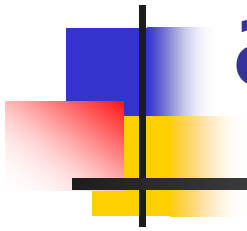
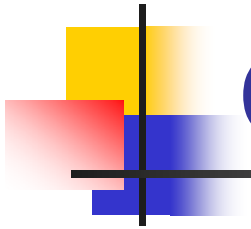


Oracle Replication Experience at FNAL

By Anil Kumar/Nelly Stanfield

CD/CSS/DSG Oct 21, 2004

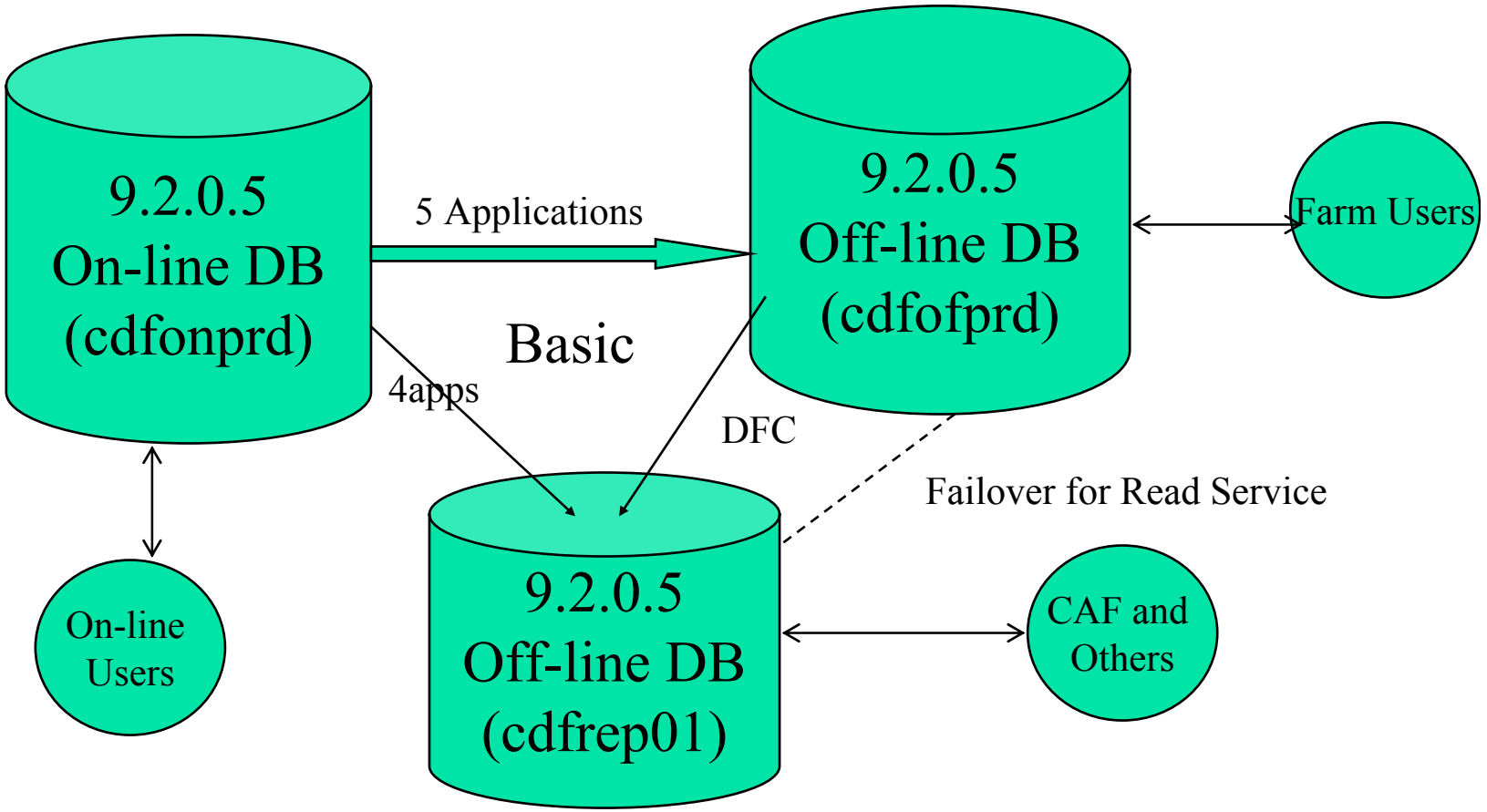


Outlines

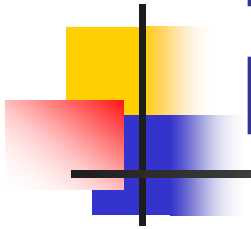
- Current Replication Scenario at CDF
- Oracle Streams Implementation at CDF
- Facts about Mix of Read-only snapshot and streams replication.
- Oracle Streams Features/Constructs used
- Development/Test Experience
- Issues/Concerns with Streams Deployment
- Replication in 10g
- Summary

CDF Db Current Scenario

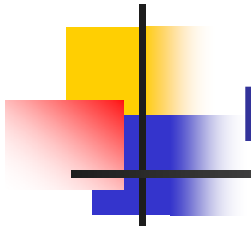
CDF Basic/Read-only Replication



Facts about BASIC/READ ONLY Replication

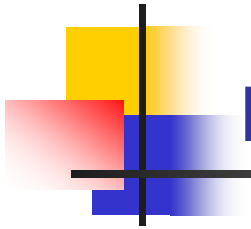


- Each replica is refreshed every 15 minutes.
- There is lag of 8 mins for refresh between replicas to get best usage of caching on the source database.
- Latency for all apps refresh is < 2 min for 15 minutes of data.
- During major maintenance/clean-up of tables (though this activity is very rare), servers parameters are tuned for replication to catch. This activity is co-ordinated with Database group.



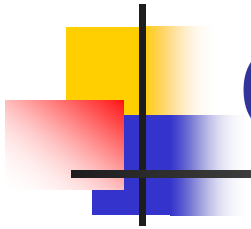
Basic Vs Streams Replication

- Basic Replication has drawbacks of scalability. All Replicas have to be originated from on-line.
- Security issue – All remote replicas have to talk to On-line with Basic Replication.
- Manual DDL sync-up. Object access is unavailable for this sync-up



Basic vs Streams Contd

- Streams Replicas can produce further replicas.
- Remote replicas need not to talk to on-line in streams replication.
- DDL sync-up is automatic in streams replication.

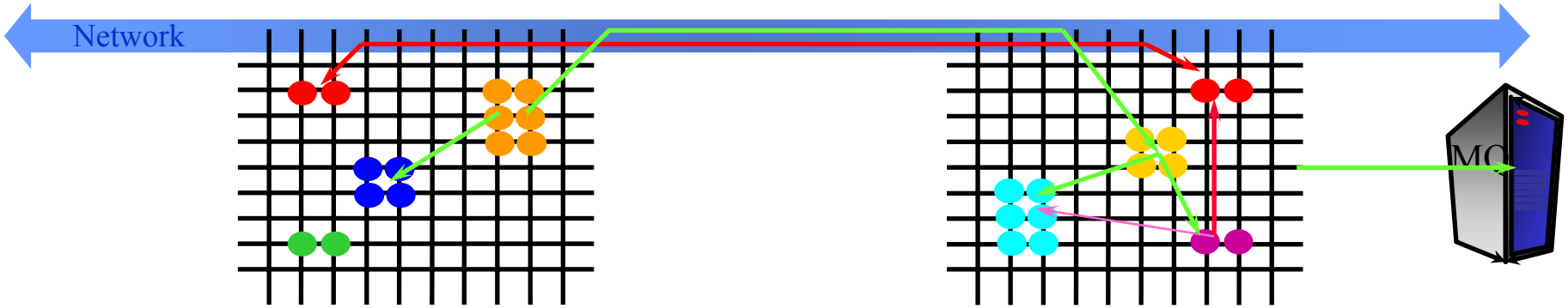


Goals of streams replication

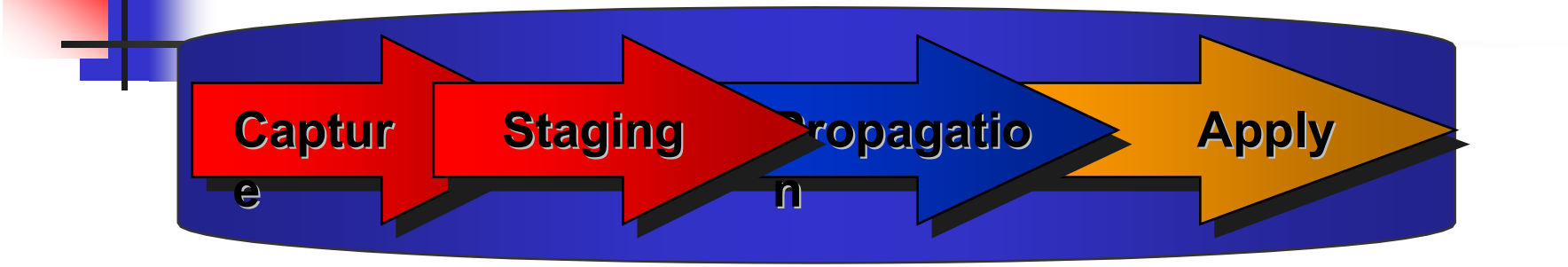
- DDL Sync Up
- Fire Walls
- Scalability (Replicas of Replica)
- Robustness/Reliability
- Easy to USE and Troubleshoot

Oracle Streams

- Oracle9*i*/*10g* can share information flexibly

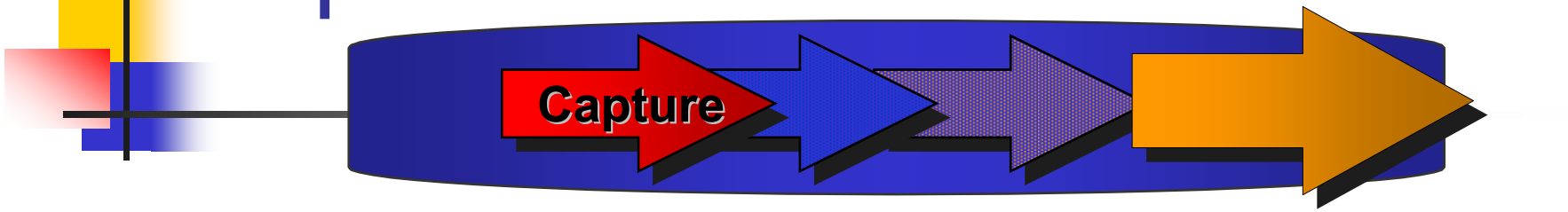


Streams Basic Elements



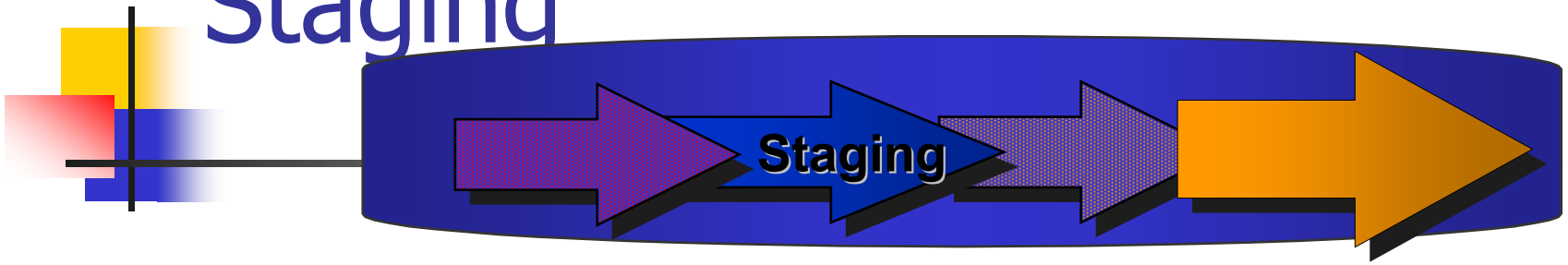
- Three basic elements in each database
  - Capture (Capture/Evaluation/Creation LCR)
  - Staging
  - Propagation
  - Apply

Capture



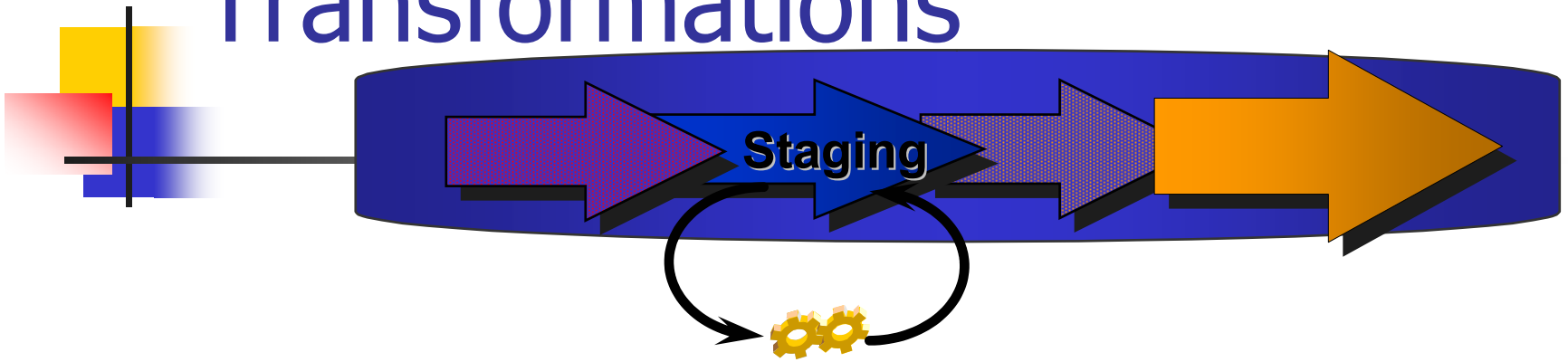
- Streams captures events
  - Implicitly: log-based capture of DML and DDL - Schema Level Rules are defined with complex rules
  - Explicitly: Direct enqueue of user messages

Staging



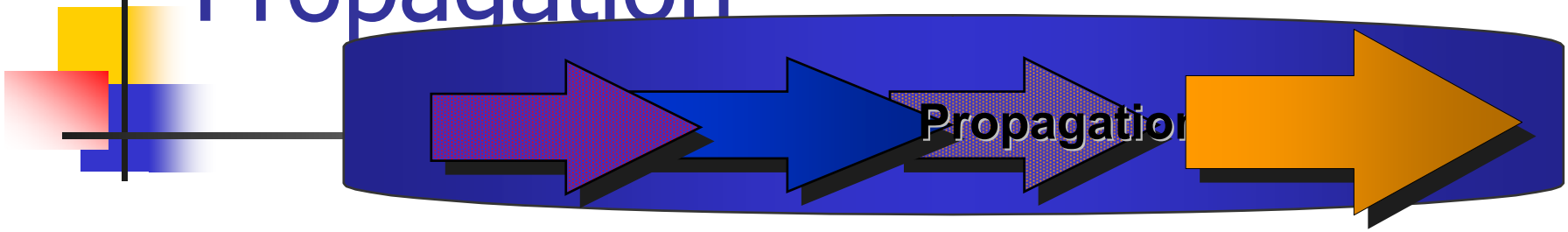
- Streams publishes captured events into a staging area
  - Implemented as a queue
  - Supports for new self-describing type “any” data type allows a single staging area to hold any type of data
  - All events, LCRs and user-messages, can be staged in the same queue
  - Messages remain in staging area until consumed by all subscribers (receiving ends)

Transformations



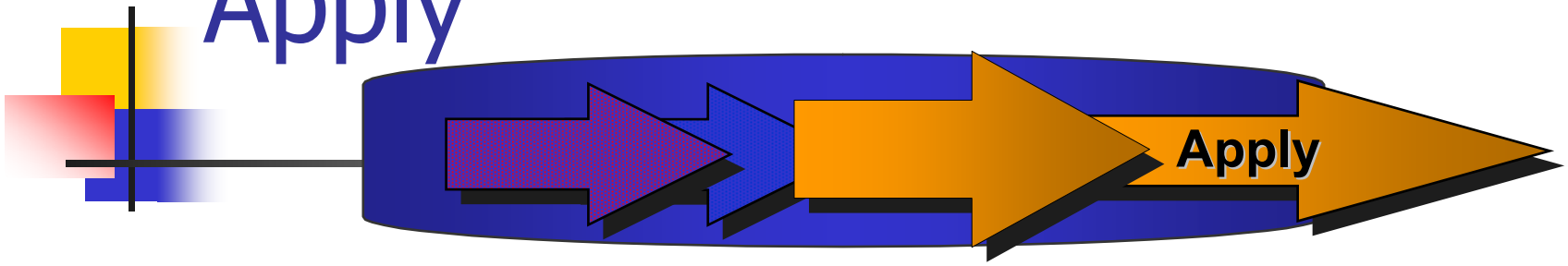
- Transformations can be performed
  - as events enter the staging area
  - as events leave the staging area
  - as events propagate between staging areas
- Transformation examples
  - change format, data type, column name, table name

Propagation



- Streams propagate the data in terms of LCRs (Logical Change Records) to the destination queue with default latency of 5 secs from Enqueue time of LCR.
- One can use a schedule too i.e. propagation wake up every 3 minutes and work for 3 mins and sleep for 15 secs. Works fine.

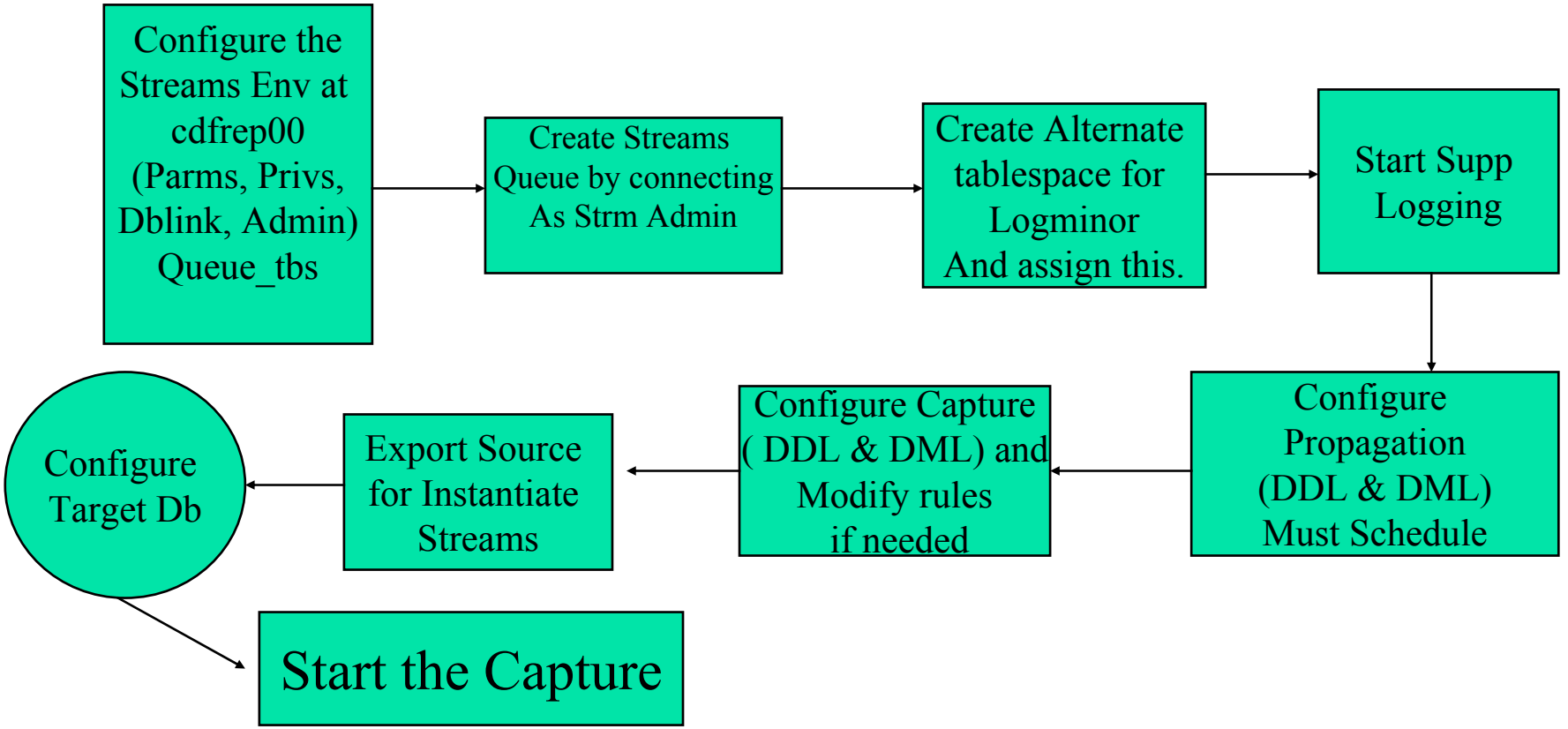
Apply



- Staged events are consumed by subscribers
  - Implicitly: Apply Process
    - Default Apply
    - User-Defined Apply

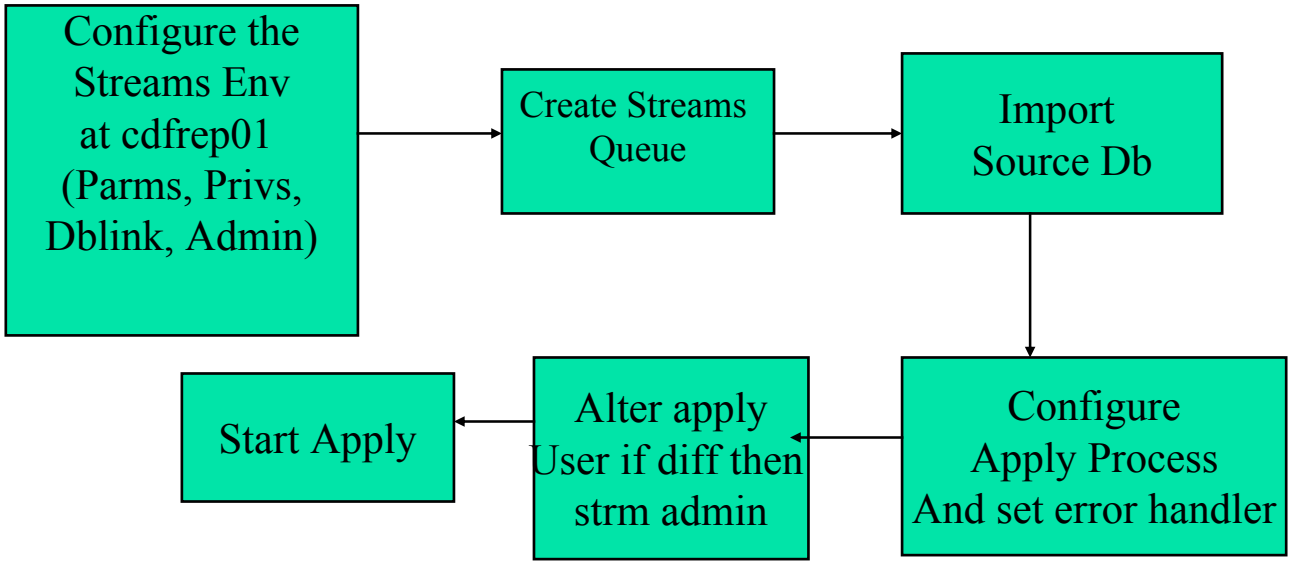
Process Flow at the Source Database

Configuring the Processes at Source Database

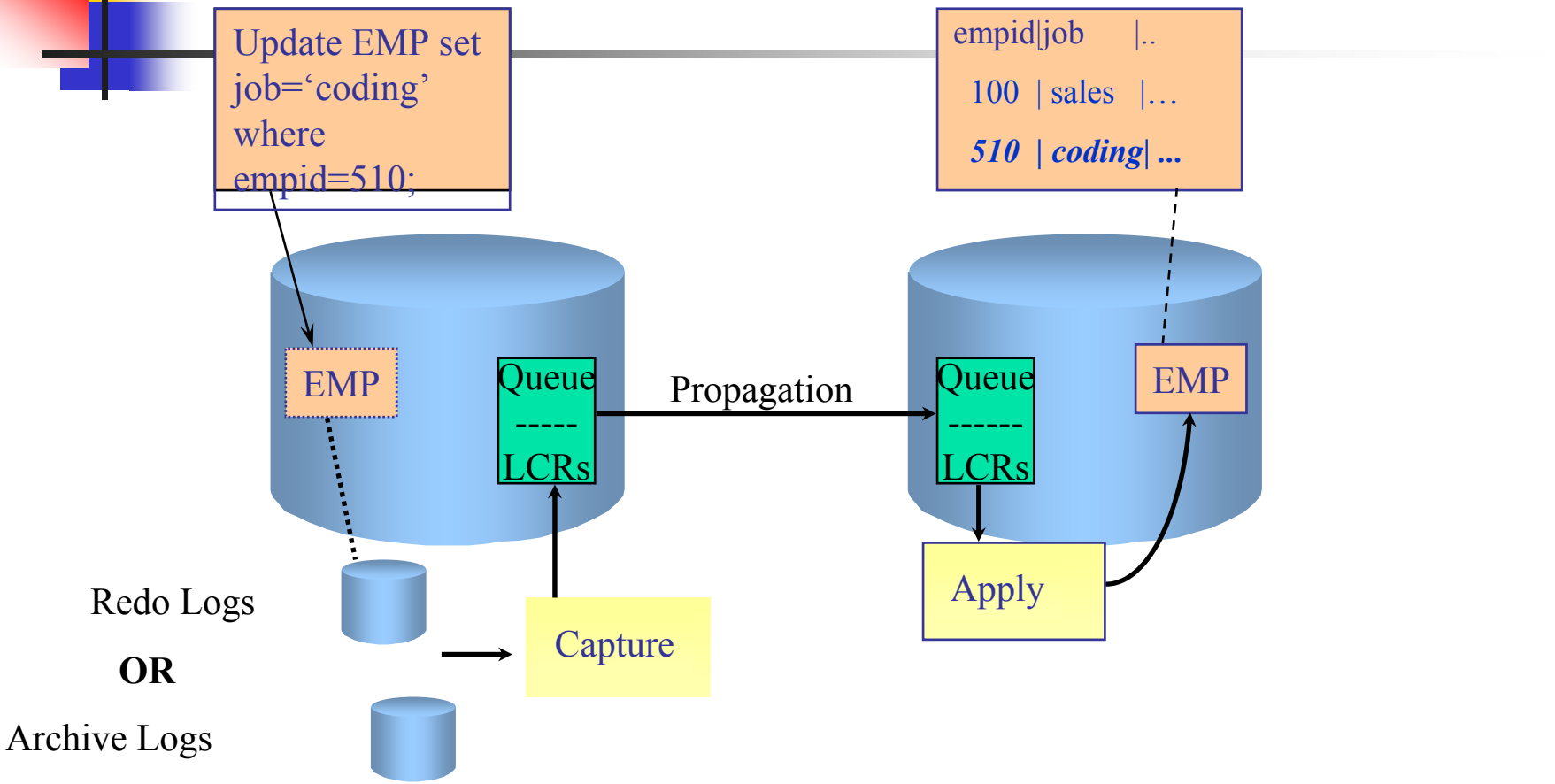


Process Flow at Target DB

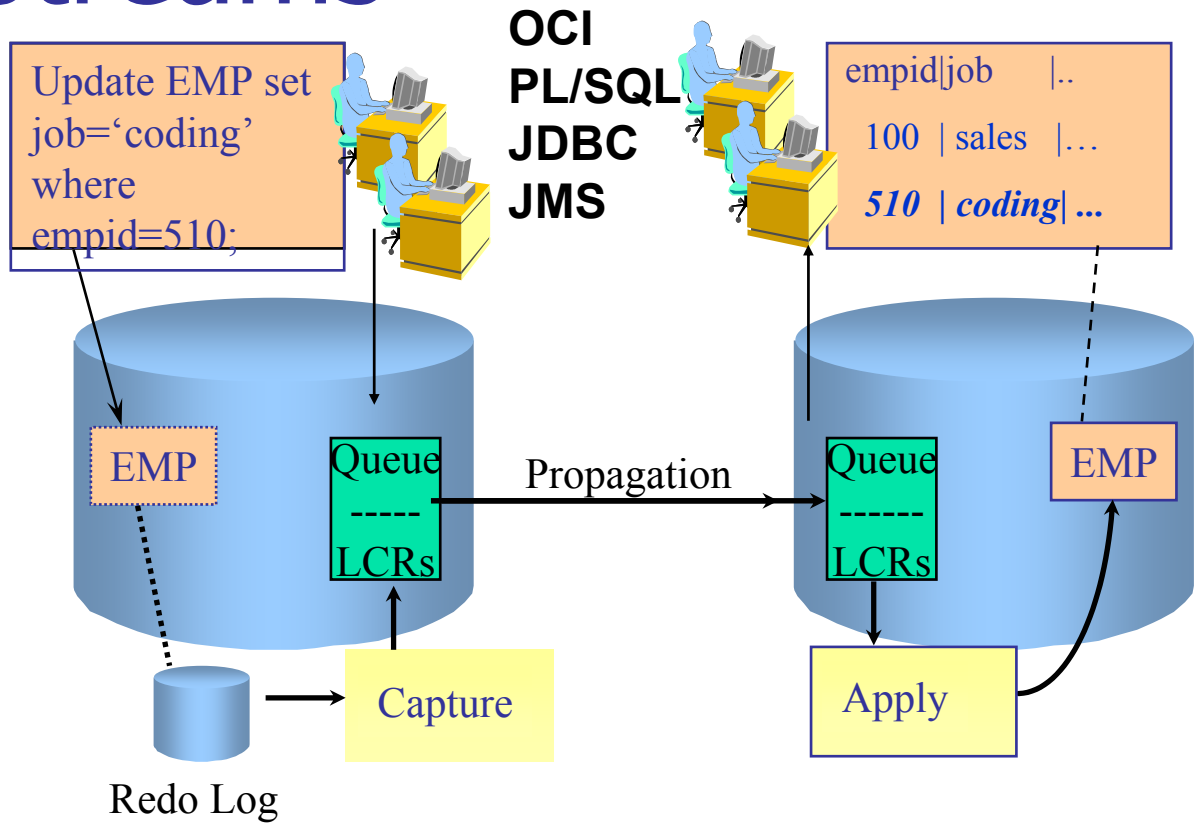
Configuring the processes at Target DB



Replication with Streams

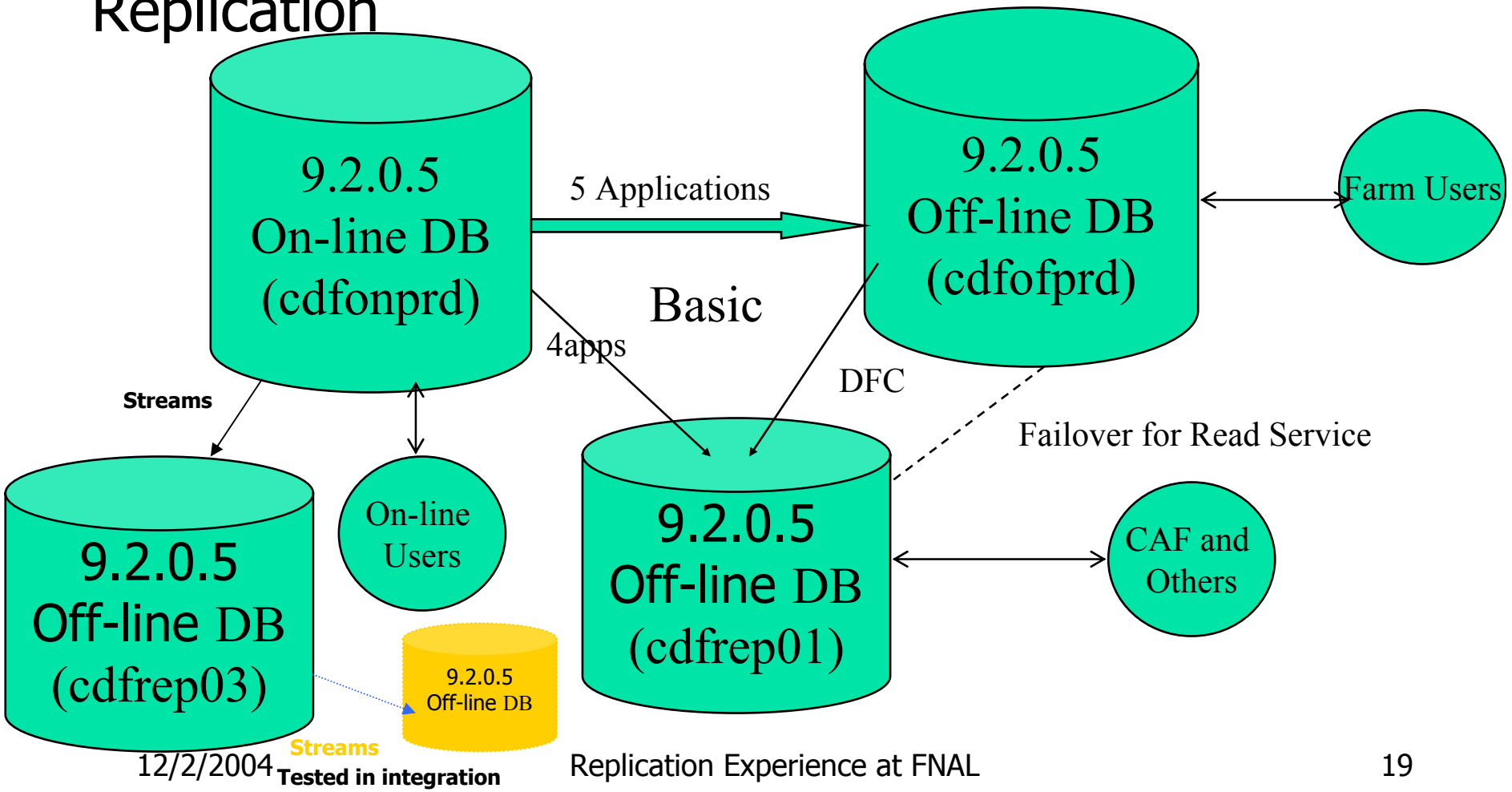


Information Sharing with Streams

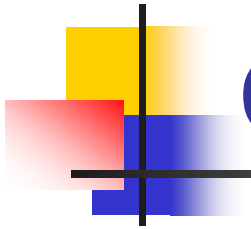


CDF Db Current Streams Scenario

CDF Mix of Basic/Read-only and Streams Replication

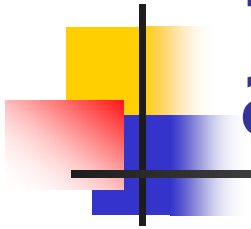


Facts about Mix of BASIC/READ ONLY and Streams Replication at CDF



- Special rules to bypass the activity related to basic replication. This makes the rules complex.
- BASIC replication's MLOG$ and RUPD$ streams objects catalog purged at source and target. Also MLOG$ and RUPD$ tables dropped on stream's target.

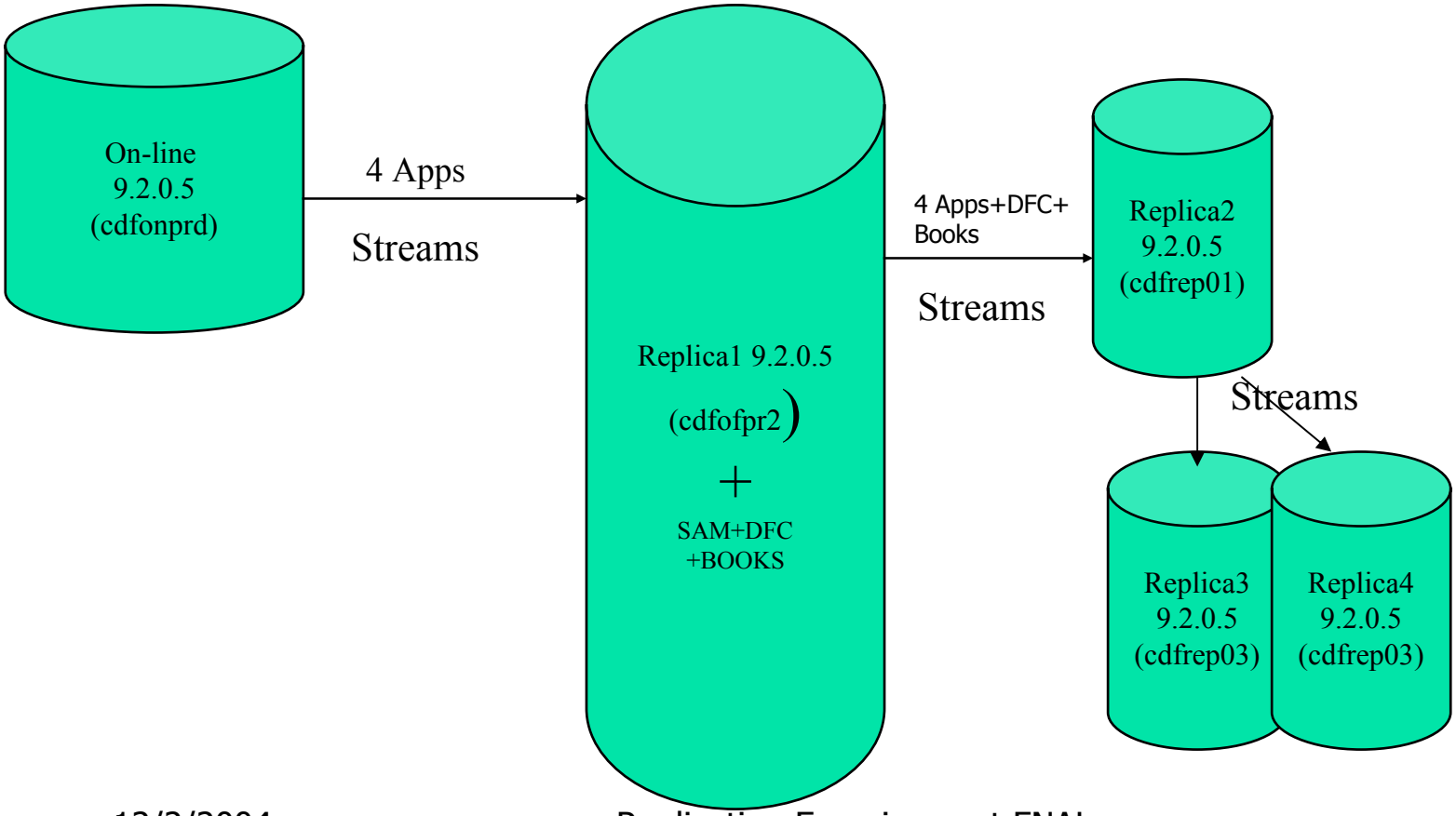
Features considered for streams at CDF

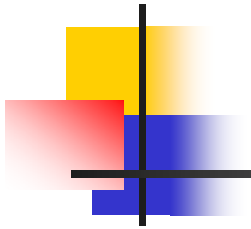


- Reserved 30% of shared pool for streams. In 10g streams has dedicated memory parameter.
- Schema Level Rules are considered Vs Global or Table Level.
- Uni-directional streams (Not muti-master due to used case at CDF . Also muti-master is complex).
- One capture and Propagation at each source db.
- One Apply for each target.
- Manual Implementation of Flow Control Management. In 10g , it is automatic.
- Degree of 3 is used for parallelism of streams capture process.
- Automatic Monitoring of streams processes and latency for capture and apply.
- Automatic start of streams process if aborted.
- Log the monitor data for capture and apply processes for later analysis.

CDF Future Streams Implementation

- CDF Replication using Oracle Streams





Development/Test Experience

- **Testing is the Key ….**

•Initially Oracle streams had lot of bugs.

http://www-css.fnal.gov/dsg/internal/ora_repl/status_streams_replication031803.htm

http://www-css.fnal.gov/dsg/internal/ora_repl/status_streams_replication_12_03_03.htm

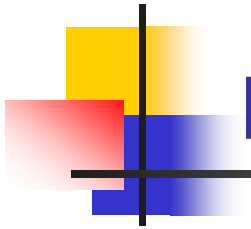
•April 2003 Integration Test as defined succeeded.

• June 2003 Backed the production implementation due to bug# 3001947 (bug was encountered whenever a LOB INDICATOR LCR is passed to the transformation.)

•Aug-Sep 2003 – More Integration Test. Patched as patches were available. Parallelism of streams process was introduced in 9.2.0.3 . Performance of streams was a big issue. Capture latency was very high. Jan 2004 Another Integration Test – used parallelism feature.

•June 2004 – Streams halted with Ora-600 error. Long running transaction cause streams to abort. Upgraded to 9.2.0.5

September, 2004 Streams in beta production at CDF.



Development/Test Experience – Contd

- Jul-Aug 2003 9.2.0.3- Streams replication did hit a performance bottleneck. Capture process very slow, The latency for capture event was 32,947 Secs. Capture process was processing logs at a rate of 150MB/hour.

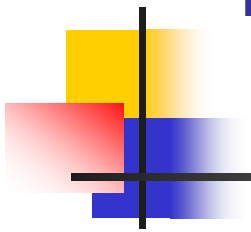
- 9.2.0.4 has fix for parallelism.

- Jan 2004 Re-establish the streams target on new machine using parallelism of degree 3.

- 220% improvement. Capture process processed the log at 330MB/hr Vs 150MB/hr without parallelism.

- Reserving 30% memory for streams didn't help since no spilling of events to disk while processing.

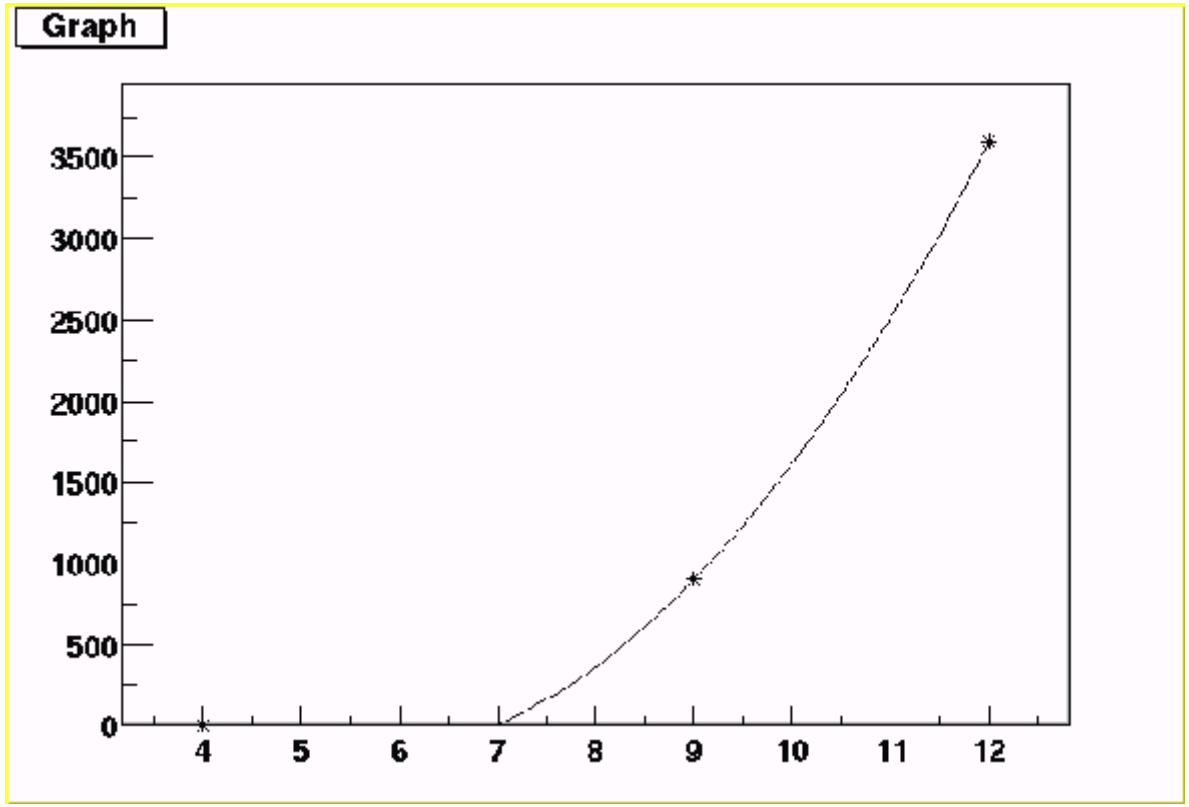
Development/Test Experience – Contd



- We had 430MB of archives being processed in 1hr 15 Mins at speed of 340MB/hr with 30% shared pool Vs 330MB/hr with 10% of shared_pool for streams.
- Since we have mix of BASIC and streams , streams replication rule are complex. That may be the performance bottleneck
- "When rules are complex, it is not possible to perform the fast rule evaluation optimization" as per product manager.

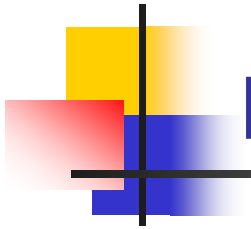
Development/Test Experience – Contd

Capture Latency Graph



Latency in secs

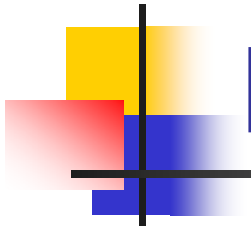
Activity in Gb/day



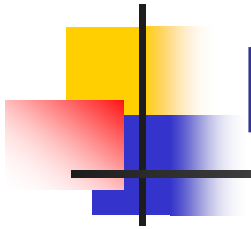
Development/Test Experience – Contd

- For archive load around 4GB/day then latency is around 8 Secs.
  And with archive load 9GB/day then latency is around 15 Minutes and with archive load 12GB/day then latency is around 60 Minutes.
  The simulation test results : http://www-css.fnal.gov/dsg/internal/ora_repl/simulation_load_test020504.htm

Issues/Concern with Streams Deployment



- Capture process takes 25% of CPU
- 2 times more space needed for archive area.
- Streams Performance will be very slow if receiving site is down for extended period of time due to heavy backlog of processing.
- During major maintenance on db in terms of db clean-up, streams process will be very slow due to high archive activity.

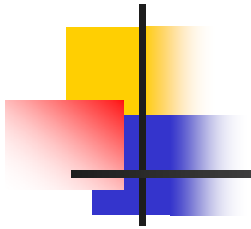


Replication in 10g

- 10g

Interesting features for streams in 10g:

- Auto Spill Management.
- No more complex rules (automatically bypass data for basic replication)
- Dedicated shared pool for streams processes.



Summary

- 10g should be considered for streams
- Auto spill management , Dedicated streams memory. Support for long running transactions in 10g.
- Very large Archive area needed (to support streams and also to support extended period of downtime at target sites)
- More CPU. Streams processing is CPU intensive.
- Multi-master streams replication is complex in implementation and conflict resolution. Also target sites should be robust enough to recover in case of failure.
- One Capture, one Propagation and apply is recommend.
- Capture with degree of parallelism 3 is recommended.
- OEM Monitoring and automatic start of streams process.
- Implement Manual Flow control if it is 9.2.0.5
- Schema level rules should be considered Vs Global database rules in terms of streams implementation. Otherwise global db operations e.g. create tablespace, drop tablespace etc. will be propagated.