

## Integrating Oracle Databases with NoSQL Databases for Linux on IBM LinuxONE and z System Servers

Sam Amsavelu  
Oracle on z Architect  
IBM Washington Systems Center  
[samvelu@us.ibm.com](mailto:samvelu@us.ibm.com)

## Copyright and Trademark Information

- For IBM – can be found at

<http://www.ibm.com/legal/us/en/copytrade.shtml>

- For Oracle – can be found at

<http://www.oracle.com/us/legal/index.html>

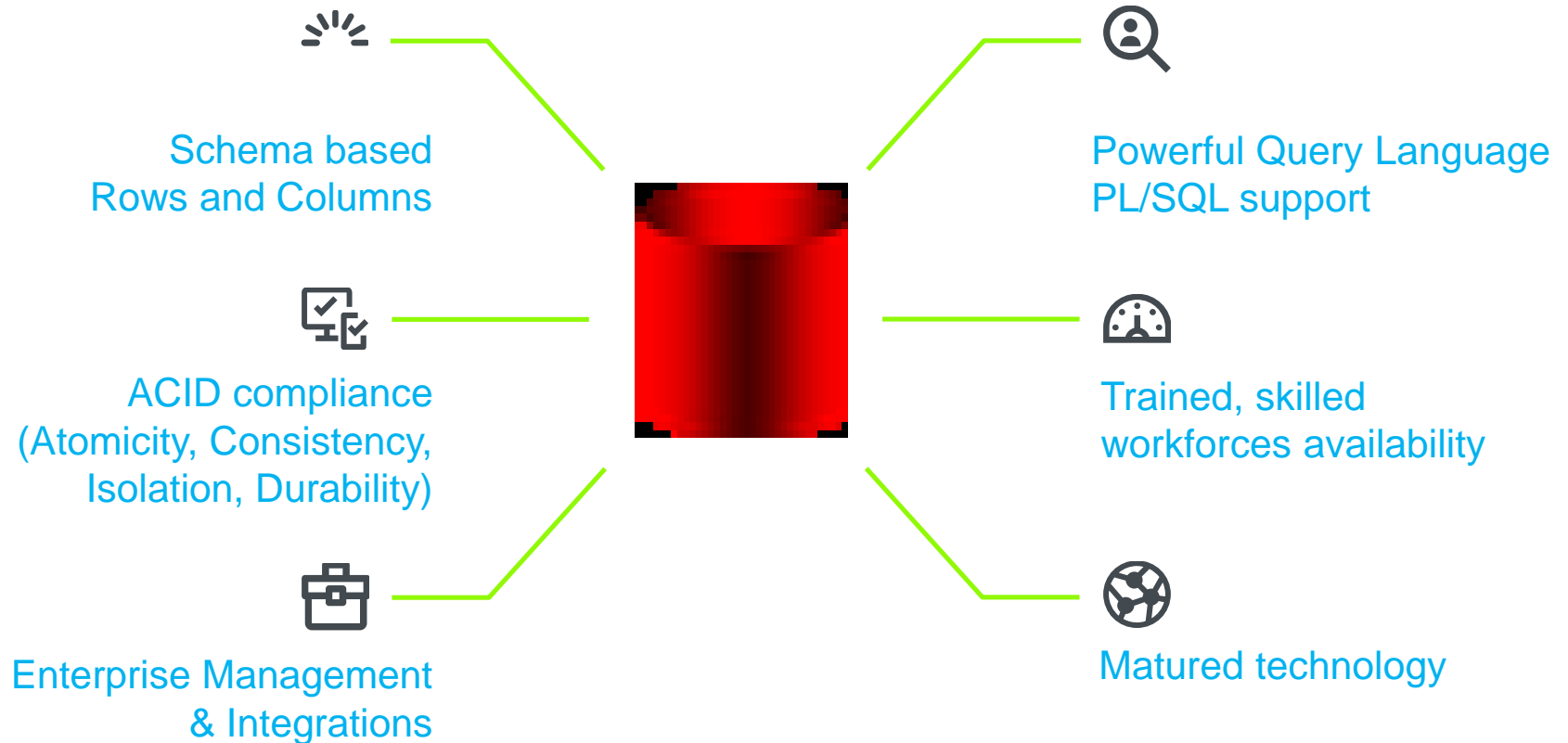
## Agenda

- Databases Primer
- Integration between RDBMS and NoSQL Databases
- Oracle GoldenGate
- Demo

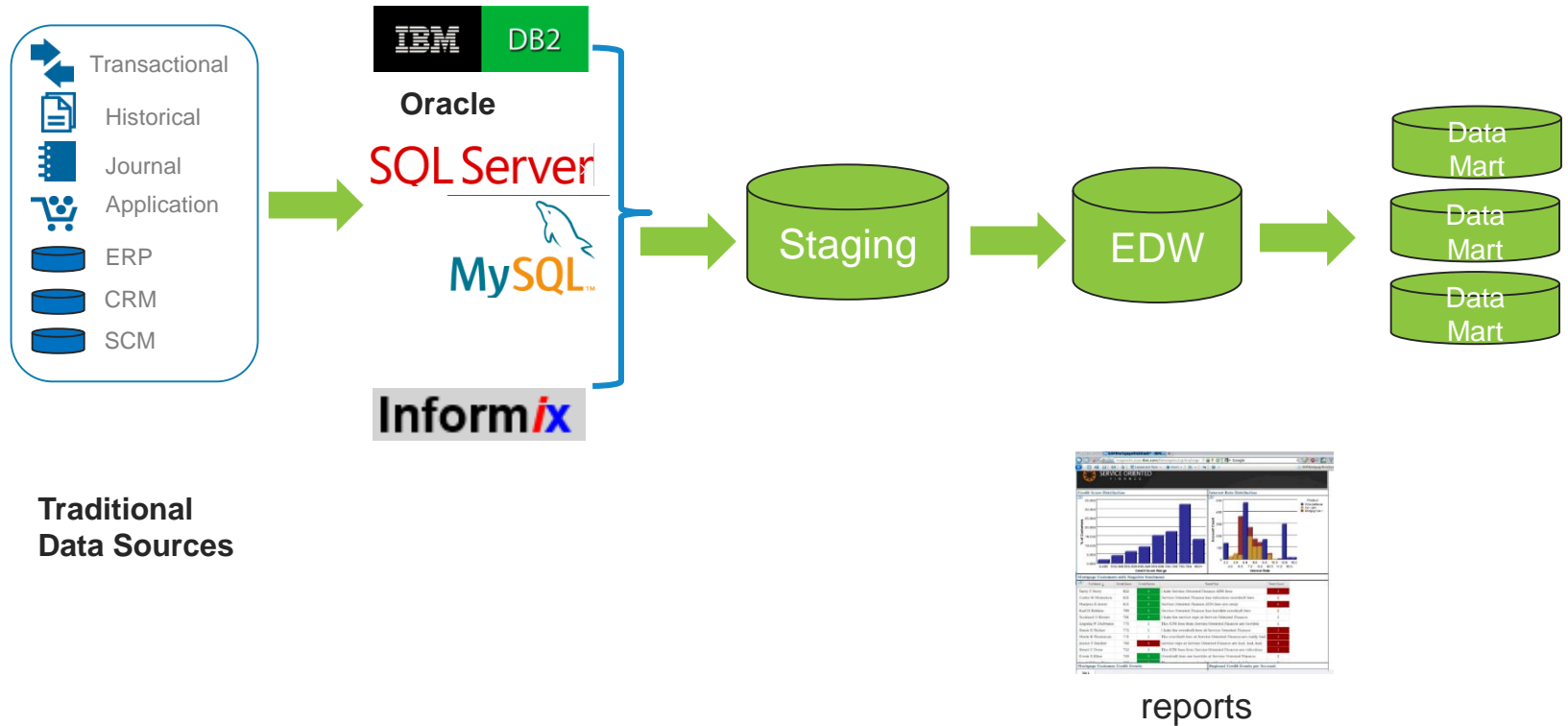
## Types of Databases

- Relational Databases
- Non Relational Databases

## Relational Database Management Systems



# Typical Landscape of Data and Analytics



## Big Data

- With the advent of Web 2.0, static web pages have become dynamic and social media is all around us.
- Everyone is tweeting, posting, blogging, sharing photos, chatting and commenting.
- The Internet of Things (IoT) is emerging — a rapidly growing network of connected devices that collect and exchange data, such as sensors and smart devices.
- Huge amounts of data is getting generated and are handled in real time
- Altogether, this generates huge amounts of new data that businesses want to absorb and use to stay ahead, to provide features such as product recommendations and a better customer experience. The data can be analyzed in search of patterns for applications such as fraud detection and behavior analytics.

## Big Data Characteristics Four 'V's

### Volume

Data volume  
is exploding

### Variety

Various formats,  
types, and  
structures  
Static data vs.  
streaming data

### Velocity

Rate of Data  
generation is  
very high and  
requirement  
for quick  
Online Data  
Analytics

### Veracity

Tolerance for  
inaccuracy of  
data expected.  
Uncertainty due  
to inconsistency,  
incompleteness,  
latency,  
ambiguities, or  
approximations



## Problems in current Data Model

- Much of the new data is *unstructured*, which means that it can't be neatly stored in a tabular database.
- Imagine trying to design a database to hold data on your grocery shopping — what you like, how often you buy it, whether you prefer milk(regular, 2%,1%, 0%) or cream with your coffee, sugar, which type organic, brand or generic.
- So new types of databases are needed to store the big data, and they need to be non-relational.
- NoSQL databases ---
  - Not “No SQL”
  - May be “Not only SQL”
  - But definitely Not Relational

## NoSQL characteristics

- No schema required before inserting data
- No schema change required to change data format
- Auto-Sharding without application participation
- Distributed queries
- Managed Cache
- Data Synchronization (multi-datacenter)

# NoSQL Databases Come in more than one flavor

## All in the NoSQL Family

NoSQL databases are geared toward managing large sets of varied and frequently updated data, often in distributed systems or the cloud. They avoid the rigid schemas associated with relational databases. But the architectures themselves vary and are separated into four primary classifications.



### Document databases

Store data elements in document-like structures that encode information in formats such as JSON. Common uses include content management and monitoring Web and mobile applications.

EXAMPLES: Couchbase Server, CouchDB, MarkLogic, MongoDB



### Graph databases

Emphasize connections between data elements, storing related "nodes" in graphs to accelerate querying. Common uses include recommendation engines and geospatial applications.

EXAMPLES: InfiniteGraph, Neo4j



### Key-value databases

Use a simple data model that pairs a unique key and its associated value in storing data elements. Common uses include storing clickstream data and application logs.

EXAMPLES: Aerospike, DynamoDB, Redis, Riak



### Wide column stores

Also called table-style databases—store data across tables that can have very large numbers of columns. Common uses include Internet search and other large-scale Web applications.

EXAMPLES: Accumulo, Cassandra, HBase, Hypertable, SimpleDB

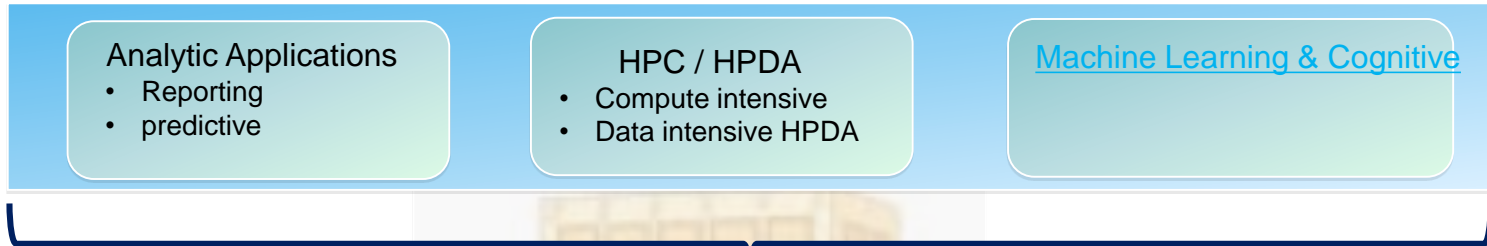
Source: [TechTarget](#)

Multi-Model = more than one type of data store, e.g.  
Couchbase is a Key Value + Document database

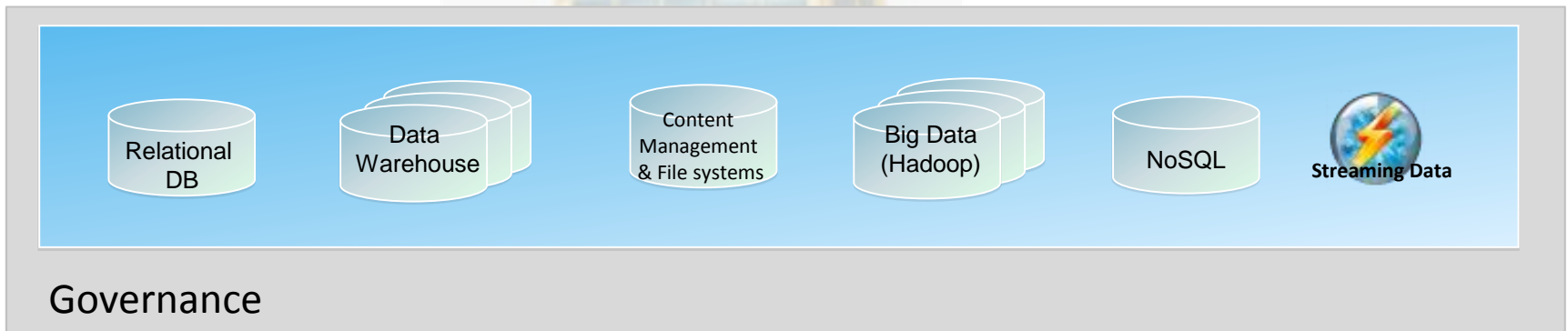
# Data in the Enterprise

## (Polyglot Persistence)

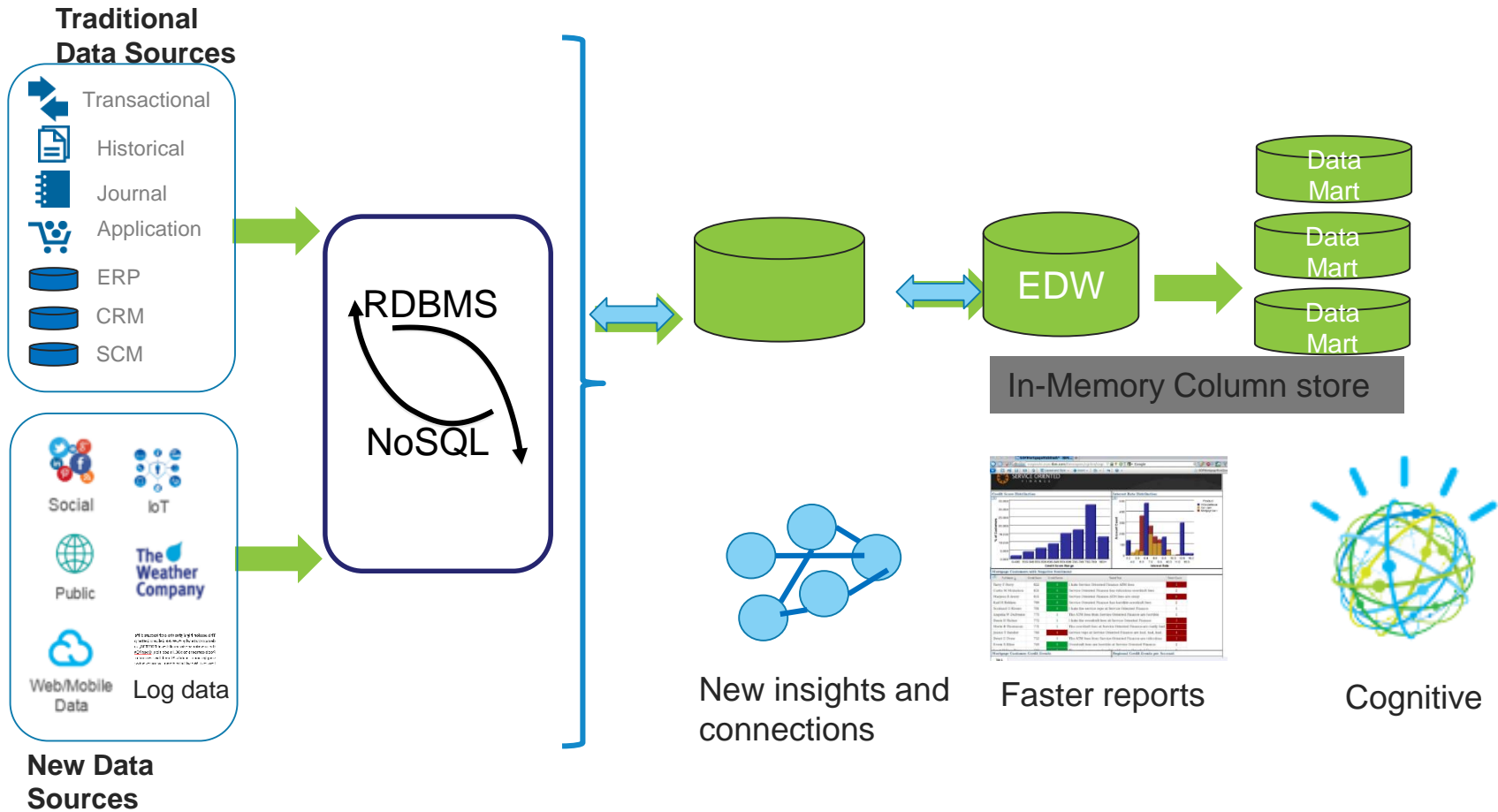
### Analytics & Cognitive applications



### Enterprise Data layer



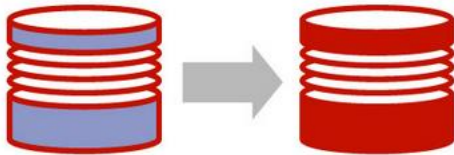
# Changing Landscape of Data and Analytics



# GoldenGate – Oracle Data Integration Product

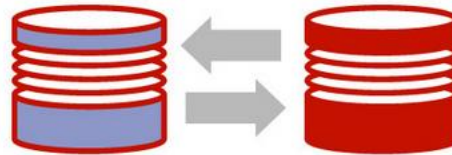
## UNIDIRECTIONAL

Reporting Instance



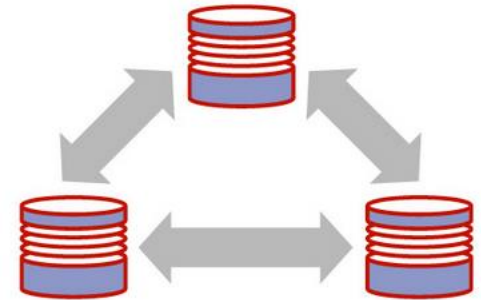
## BI-DIRECTIONAL

Instant Failover "Active-Active"



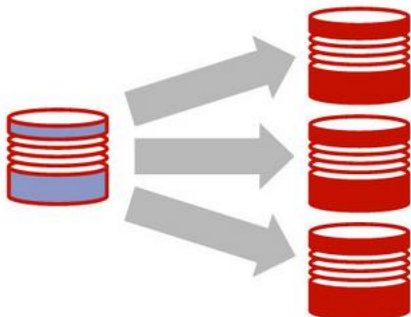
## PEER-TO-PEER

Load Balancing, High Availability



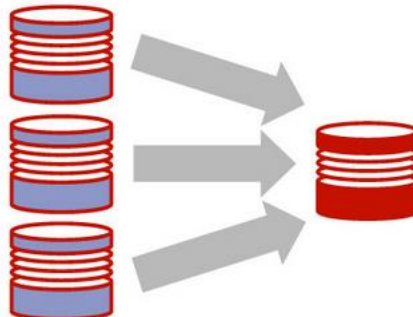
## BROADCAST

Data Distribution



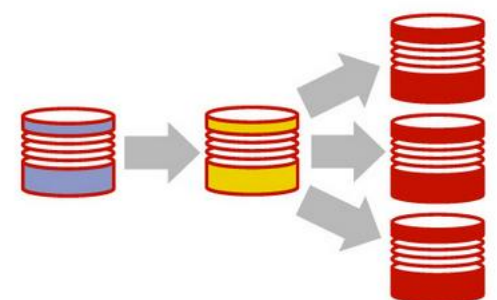
## CONSOLIDATION

Data Warehouse/Mart/Store



## CASCADING

Scalability, Database Tiering



## GoldenGate's fundamental building blocks

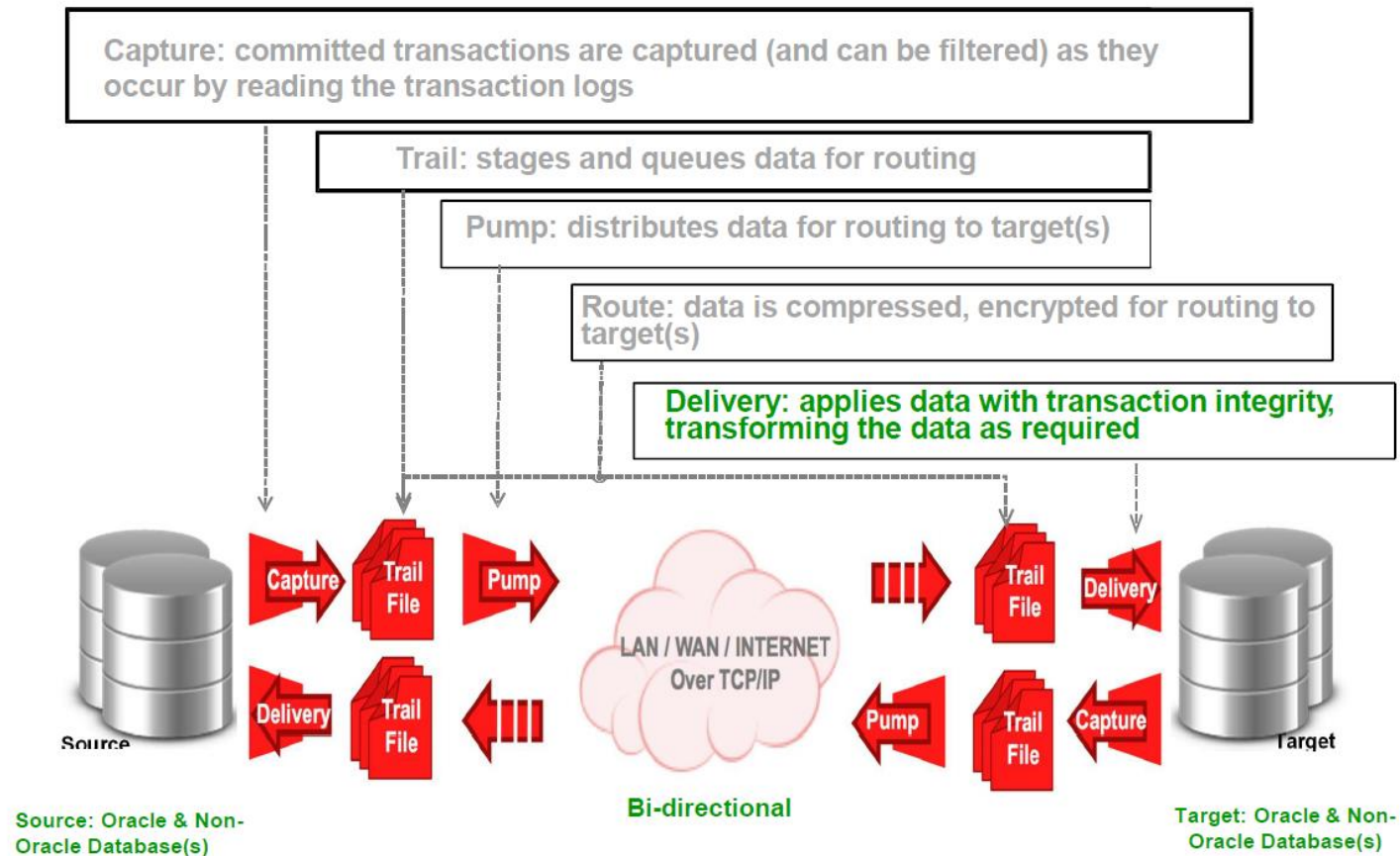
- Capture process
  - known as **Extract**, obtains the necessary data from database transaction logs
  - Extract process regularly **checkpoint** its read and write position to insure it can recover its processes without data loss in the case of failure
  - Classic and Integrated (Using Oracle database log miner)
- Trail files
  - GoldenGate converts the captured data into a Canonical Format and writes into 'local trail files' (source)
  - GoldenGate creates and uses trail files on Source side as well as Target side.
- Data pump
  - an additional Extract process called Data Pump reads data from 'Local Trail Files' and sends data in large blocks across a TCP/IP network to the target system and the transferred data is assembled and written in 'Remote trail files' (to avoid SPoF)

## GoldenGate's fundamental building blocks

- Apply process
  - known as Replicat, is the final step in the data delivery
  - It reads the trail file and applies it to the target database in the form of DML (deletes, updates and inserts) or DDL\*. (database structural changes)
  - Can be concurrent with the data capture or performed later
  - Can be configured to apply desired functions
  - Replicat process regularly **checkpoint** its read and write position to ensure it e can recover its processes without data loss in the case of failure
  
- Manager process
  - The Manager Process runs on both source and target systems
  - It controls activities such as starting, monitoring, and restarting processes; allocating data storage and reporting errors and events
  - The Manager Process must exist in any GoldenGate implementation



# How Oracle GoldenGate Works



## GoldenGate Big Data

- Handlers available:
  - HDFS Handler
  - HBase Handler
  - Flume Handler
  - Kafka Handler
  - Cassandra Handler
  - **MongoDB Handler**
  - JBDC Handler
  
- MongoDB Handler can be used to replicate the transactional data from Oracle GoldenGate trail to a target MongoDB database
- A record in MongoDB is a Binary JSON (BSON) document, which is a data structure composed of field and value pairs.
- The values of fields may include other documents, arrays, and arrays of documents.

## Monitoring Oracle GoldenGate

- **Oracle GoldenGate Monitor:**

- Provides secure viewing, management, and alerting capabilities for GoldenGate with end-to-end topology solution displays and customizable topology views.

- **Oracle GoldenGate Director**

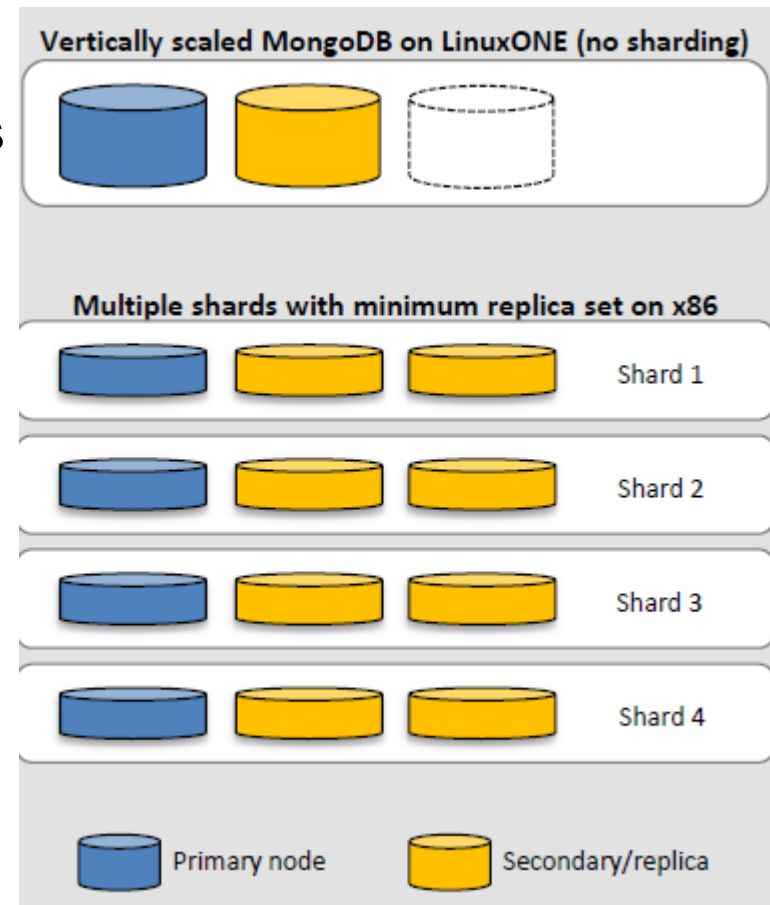
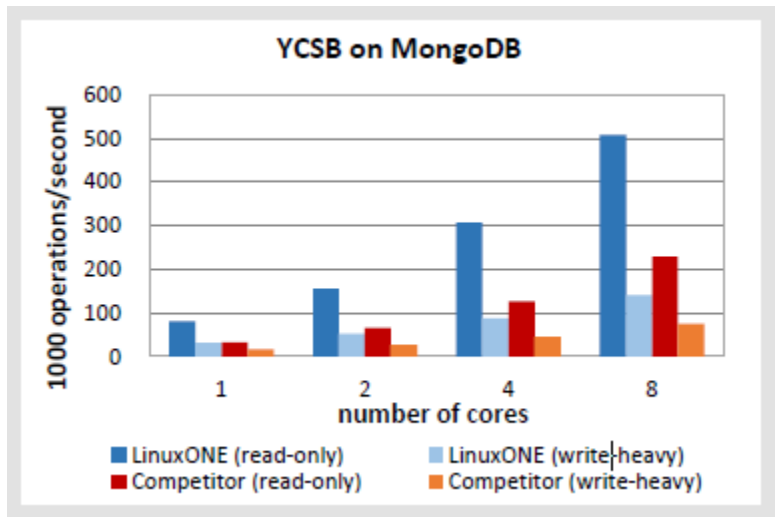
- Works on legacy GoldenGate instances
- Has basic configuration, management, monitoring, and alerting

- **Oracle GoldenGate Plug-in for Enterprise Manager:**

- Provides secure viewing, management, and alerting capabilities for GoldenGate while leveraging Oracle Enterprise Manager framework.

## Why MongoDB on LinuxONE and z System Servers

- High-performance data serving
- Better data consistency and reduced overhead
- Lower Total Cost of Ownership
- Security and resilience = Trusted operations
- Enterprise grade High Availability



## Demo Use Case

- The customer has catalog application for multiple products
- Each product has different attributes like product specs, pictures etc.,
- All these were suitable for a document database where there is no schema but each product was represented by product id and description
- Customer implemented that in MongoDB
- The resulting cart transaction was stored as system of records in Oracle Database.
- Some of the user information customer wanted to store in MongoDB

## Demo walkthrough

- Check the GoldenGate monitoring through OEM
- Configure the GoldenGate for Big Data adapter
- Check the Customer table in Oracle
- Check the MongoDB collections for Customer
- Check the Extract job of GoldenGate is running
- Check the Replicate job of GoldenGate for Big Data is running
- Create records in Customer table at Oracle
- Check the Extract Statistics
- Check the Replicate Statistics
- Check the MongoDB collections for Customer



Thank You!