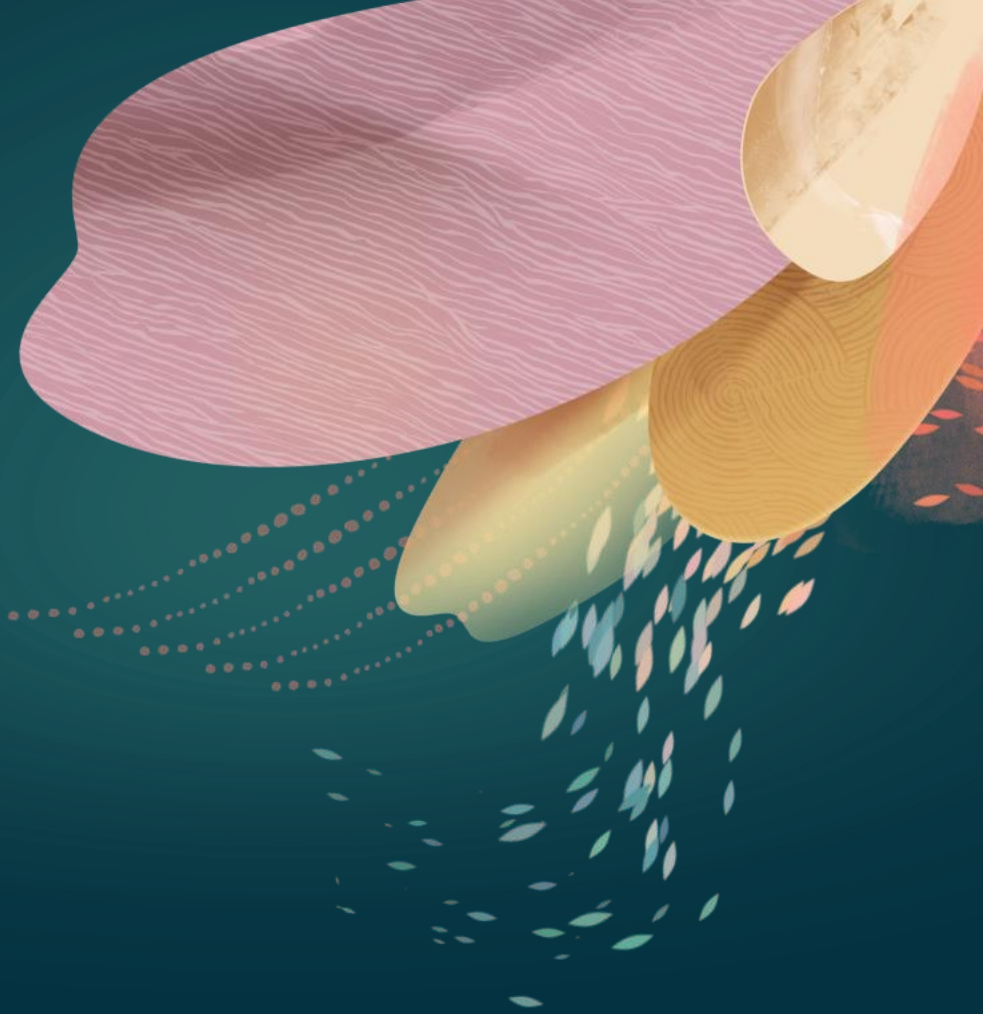


Oracle Globally Distributed Autonomous Database

Transparently Distribute Data Across the Globe for Scale, Survivability, and Sovereignty

Habib Ur Rahman

Principal Product Manager

Oracle Globally Distributed Database and True Cache

Dec 2024

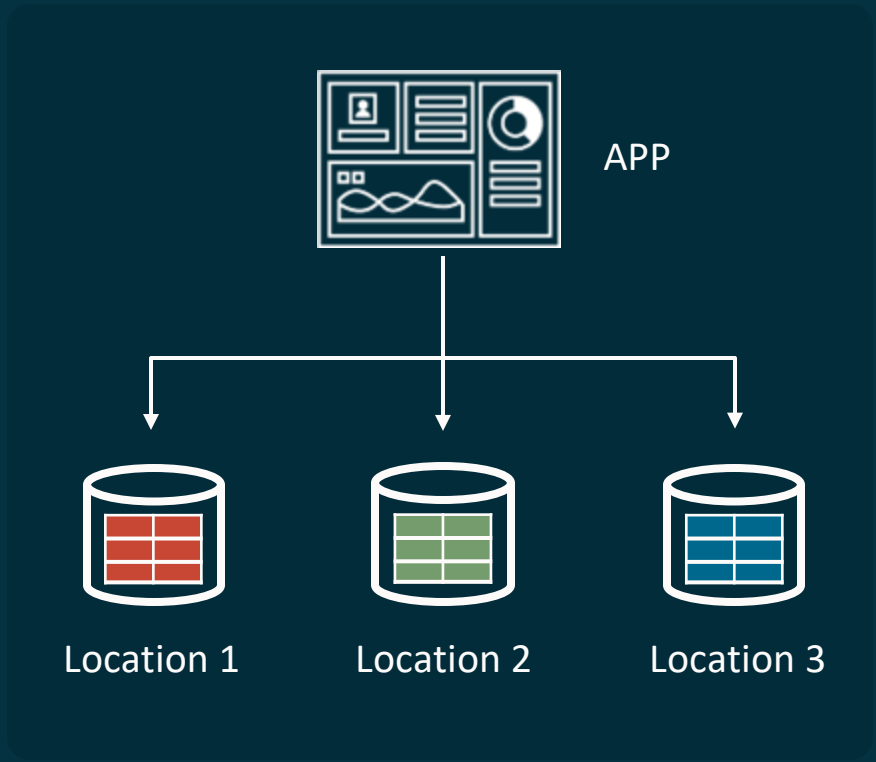
Confidential - Oracle Restricted

What is a Distributed Database?

It is a database that stores data across multiple physical locations instead of one location

- Each location stores a subset of the data

The physical distribution of data is hidden from applications



The Oracle Database became a distributed database in 2017 when native database sharding was released

Today, it powers many critical distributed applications around the world

Mobile Messaging

Credit Card Fraud Detection

Payment Processing

Personalized marketing

Internet Infrastructure

Smart Power Meters (IoT)

Confidential - Oracle Restricted

There are two main use cases for Distributed Databases

Ultimate Scalability and Survivability

Data is distributed and replicated across databases for hyper-scale and fault tolerance

Transparent Data Sovereignty

Each country's data is stored in-country to help implement regulatory requirements

Confidential - Oracle Restricted

An Oracle Globally Distributed Database is a single logical database

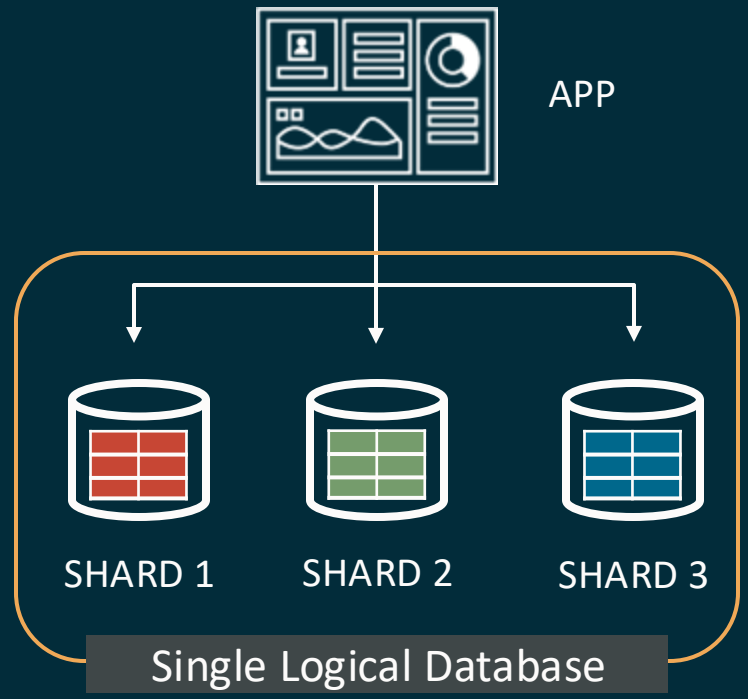
Data is physically distributed across multiple databases, which are called shards

Data in each shard is replicated for survivability

All shards can process application requests

- Active-active architecture

Data can be redistributed across shards, data centers, and regions while the database is running



Benefits of Sharding

Hyperscale

Add shards online to increase database size and throughput. Online elasticity

Fault Isolation

Break the monolith

Fault of one shard has no impact on others

Data Sovereignty

User-defined data placement for complying with regulatory requirements

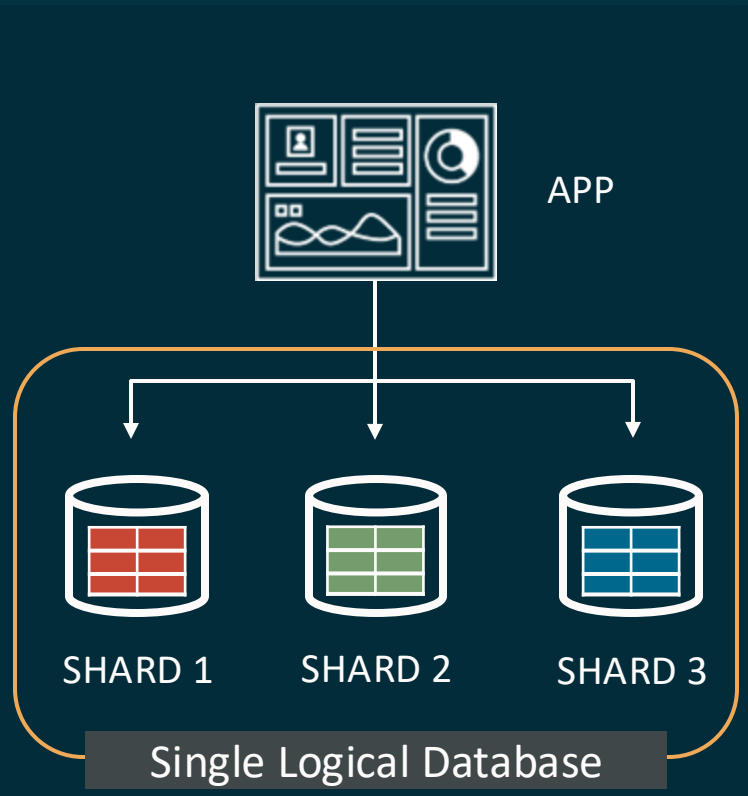
The distribution of data is hidden from applications

Application requests that only need data from a single shard are transparently sent directly to that shard

Requests that need data from multiple shards are automatically split into multiple requests that are sent to the appropriate shards and committed atomically

Queries can be parallelized both within and across shards to implement massively parallel analytics

- Next generation Big Data, done right



Let’s look at two customer examples

Data Sovereignty is becoming mandatory in India for payment data

The Reserve Bank of India Localization Regulations state

Payment data must reside in India
if both the payer and payee are Indian entities

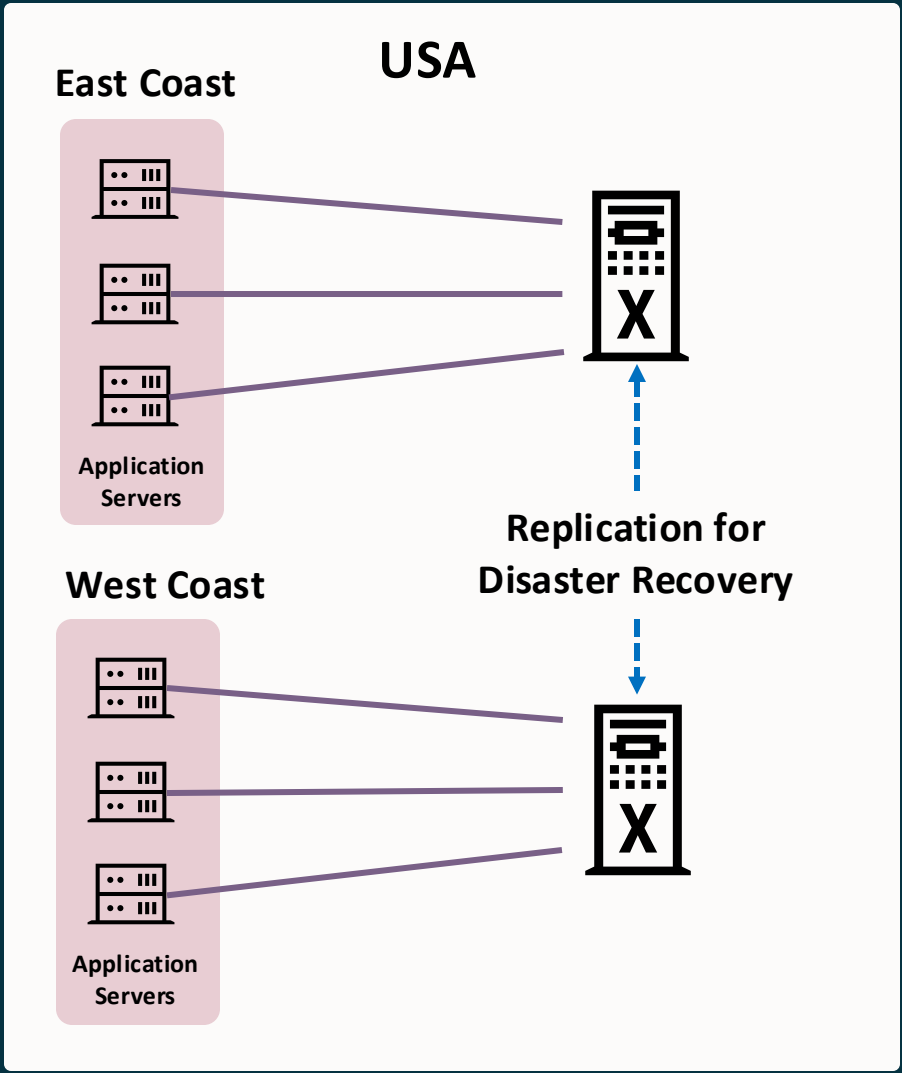
Show-stopper for global financial services companies that store data for all countries in a single database

Key observation

- Data must be stored in India
- Can be accessed from anywhere

Confidential - Oracle Restricted

One of the largest US banks had to rearchitect their payment database to satisfy this regulation

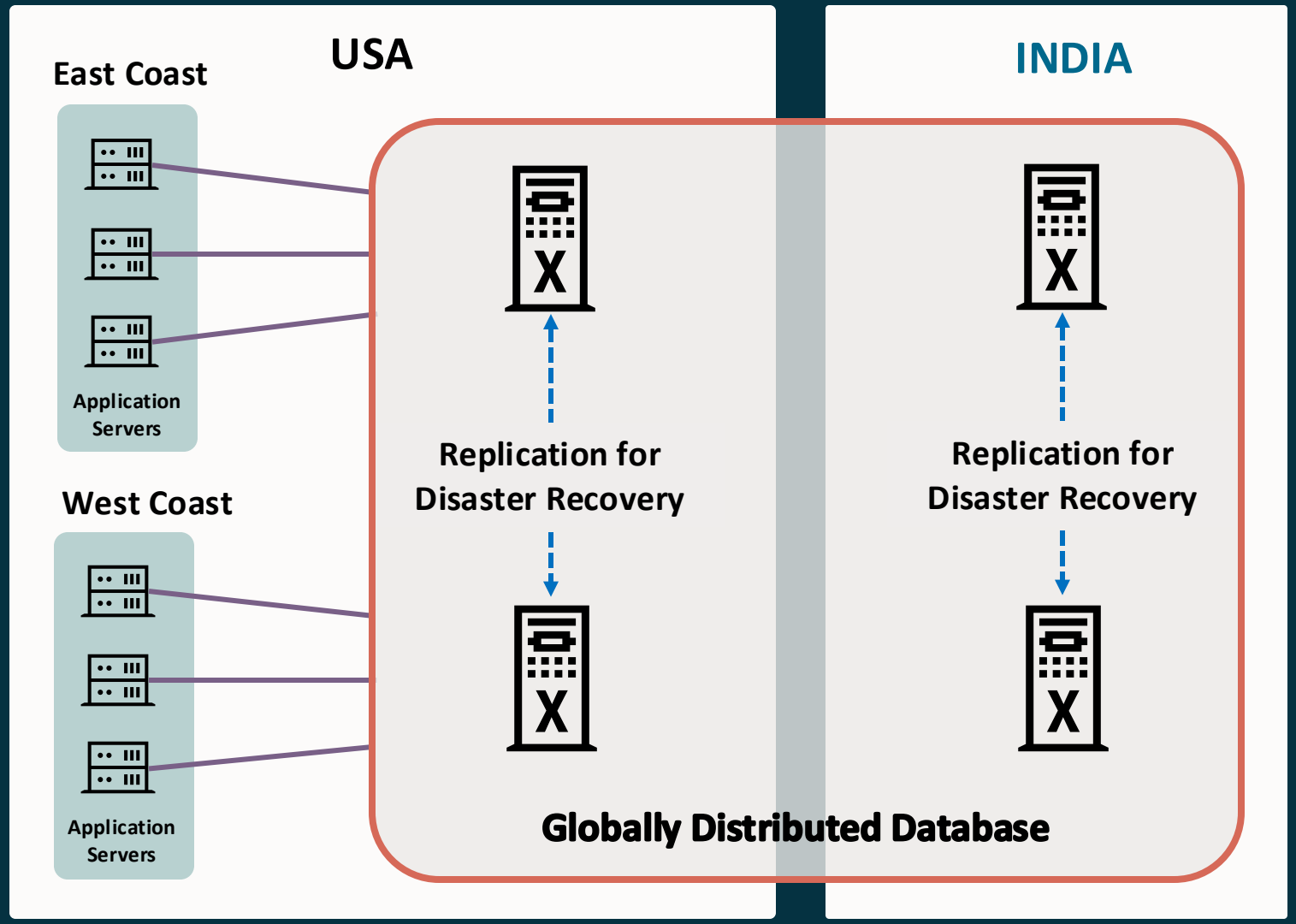


Initially all worldwide payment data was stored in the US on Exadata systems

The application and database tiers were replicated across regions inside the US for disaster recovery

Confidential - Oracle Restricted

Oracle Globally Distributed Database enabled the bank to easily comply with India's regulations



New shards were created in India to hold Indian data

The new database architecture required minimal application changes

The highly complex application-tier architecture did not need to be redundantly deployed in India

Confidential - Oracle Restricted

BlueKai is a leading data platform for digital marketing campaigns

Accessed in real-time by hundreds of millions of consumers as they surf the internet

BlueKai is a Hyper-Scale Workload on a Multi-Petabyte Database
that requires Near-Instant Response Times

1 Million

Transactions per second

30 Billion

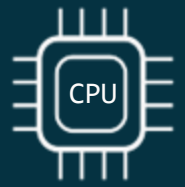
API calls per day

1.6 Millisecond

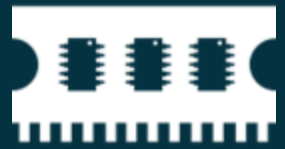
API response time

BlueKai now runs on Oracle Globally Distributed Database

104
Commodity Servers



5,408
CPU cores



77.4
Terabytes of memory

It was migrated from a combination of Aerospike, Cassandra, and ScyllaDB

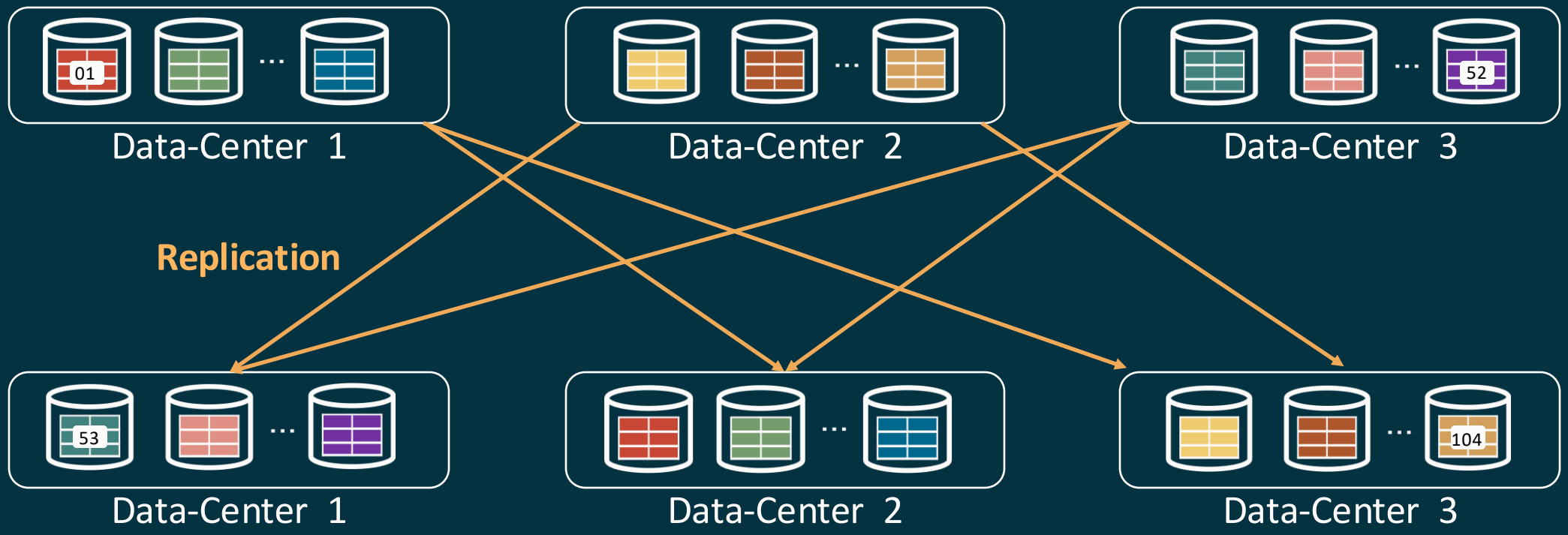
- More details on BlueKai's deployment can be found here

The simplicity of the new architecture plus the power of SQL enables BlueKai to innovate many times faster than before

Confidential - Oracle Restricted

BlueKai uses a 2.5 Petabyte distributed database

Distributed across 3 Data Centers for Scalability and Survivability



Oracle has more data distribution methods than any other distributed database

Matching the distribution method to the needs of the application is critical for a distributed database

Oracle supports Value-Based and System Managed data distribution

Value based distributes data by list or range of value

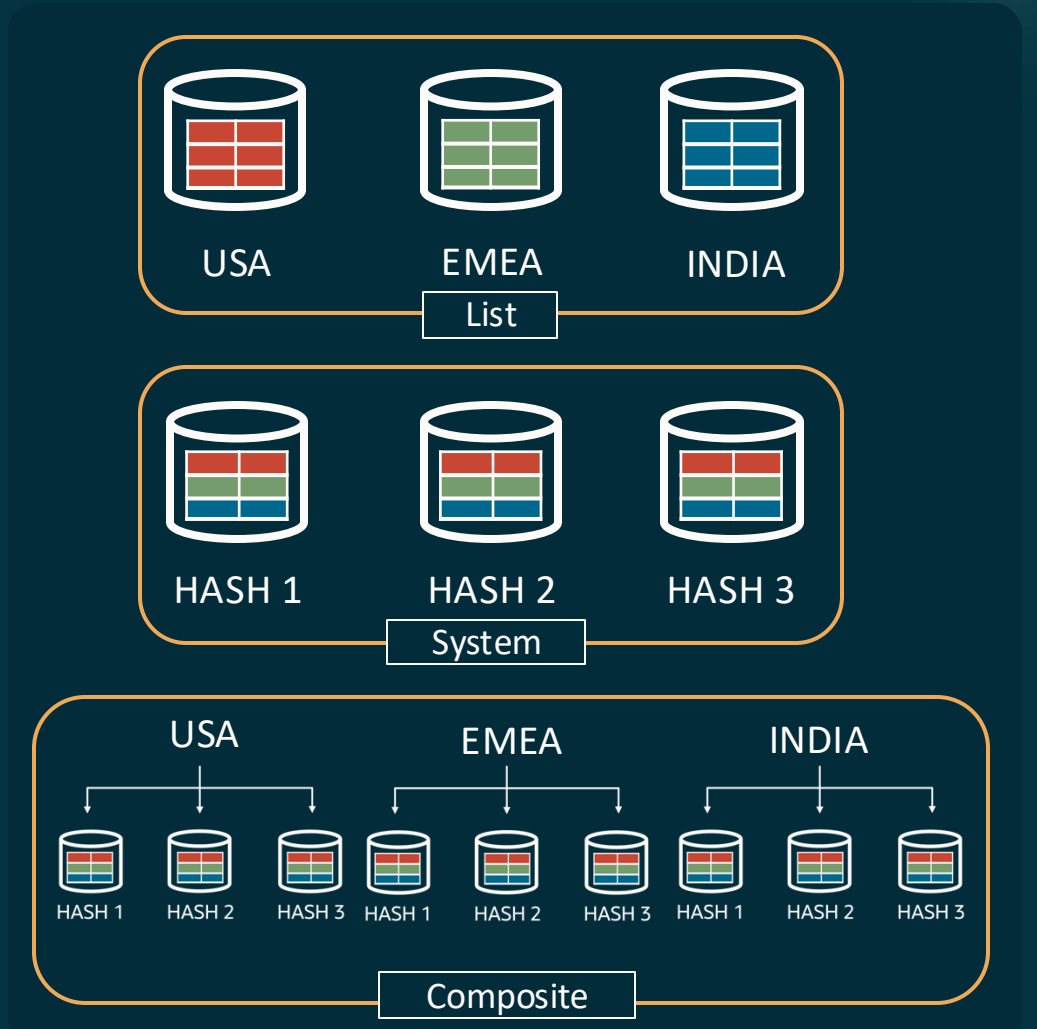
- For example, country or phone numbers

Or system managed distributes data by a consistent hash

- For example, customer id, device id

Or composite distributed data at two levels

- First by range or list of value (country) second by hash (customer id)

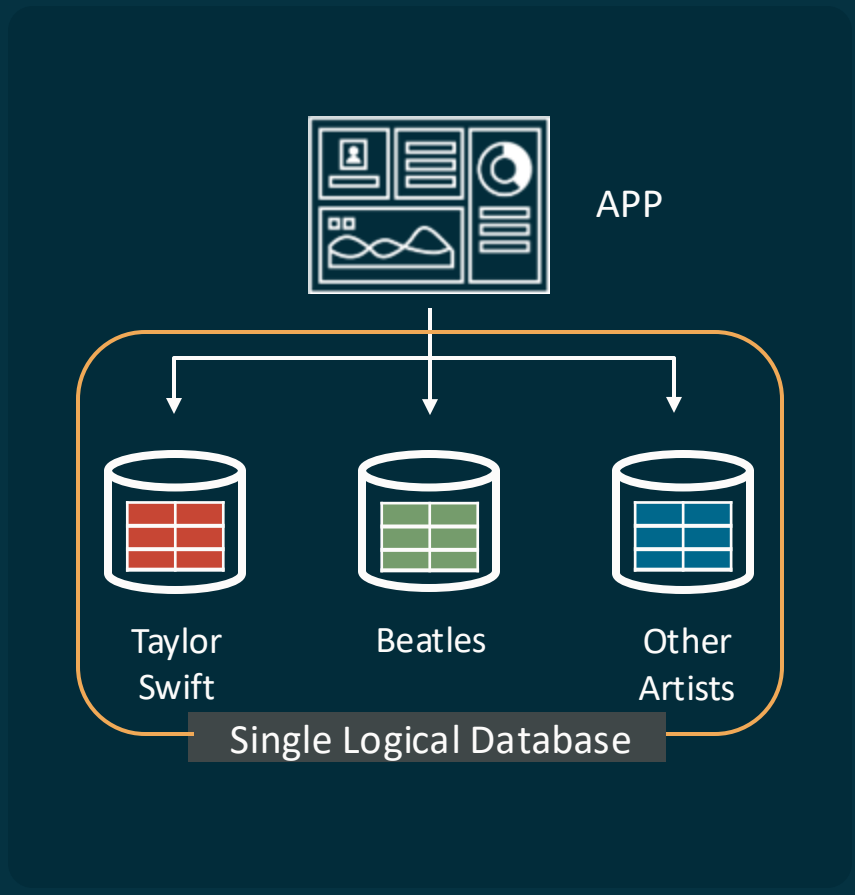


Confidential - Oracle Restricted

Oracle supports User-Defined data distribution

Used when data requires special handling such as skewed data

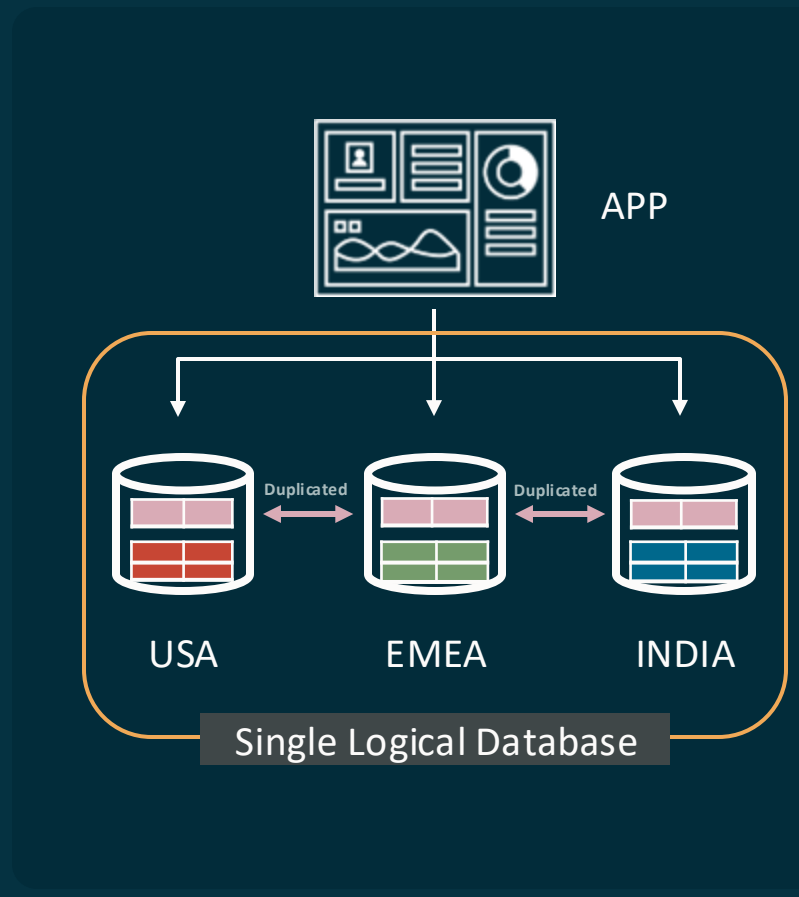
- For example you can store Taylor Swift and Beatles data in their own shards
- Lump smaller artists' data together in a separate shard



Oracle supports Duplicated data distribution

Smallish tables that are read-mostly can be duplicated across all shards

Used to avoid cross-shard queries and cross-shard referential integrity checking

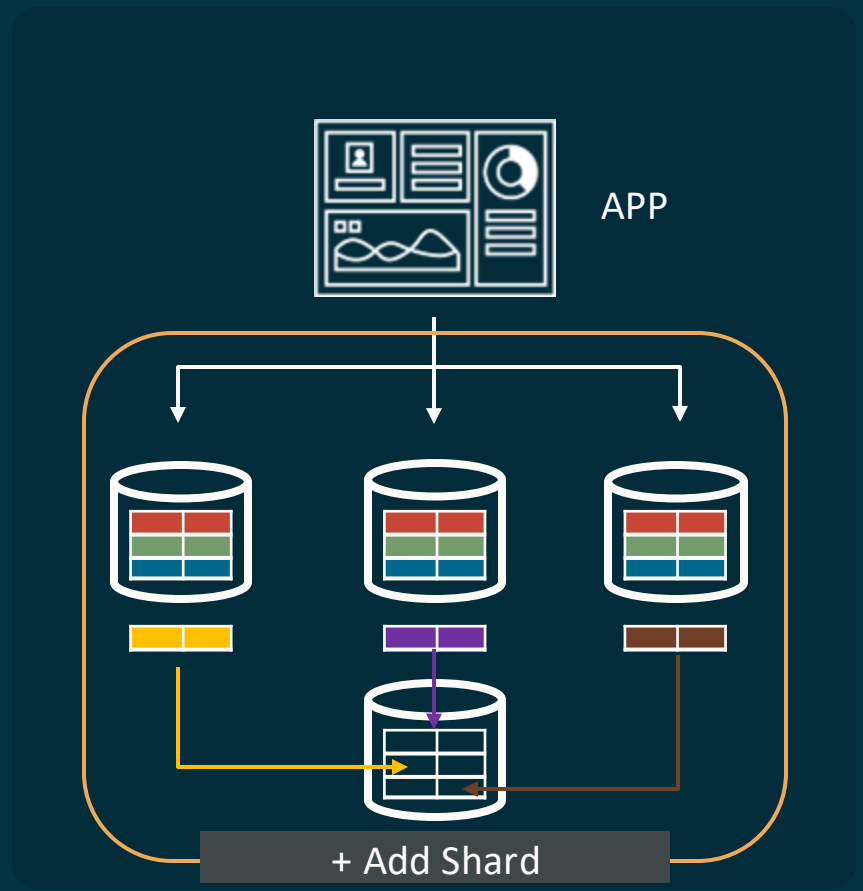


Online Addition and Automated Data Redistribution across Shards

To scale-out or scale-in, shards can be added or removed

Data across shards is automatically rebalanced with minimum data movement

This is an online operation and doesn't incur any downtime



Oracle has more replication methods than any other distributed database

Matching the replication method to the needs of the application is critical for a distributed database

Confidential - Oracle Restricted

Raft-Based Replication for Survivability Extreme

New replication method that uses the popular Raft quorum-based replication protocol

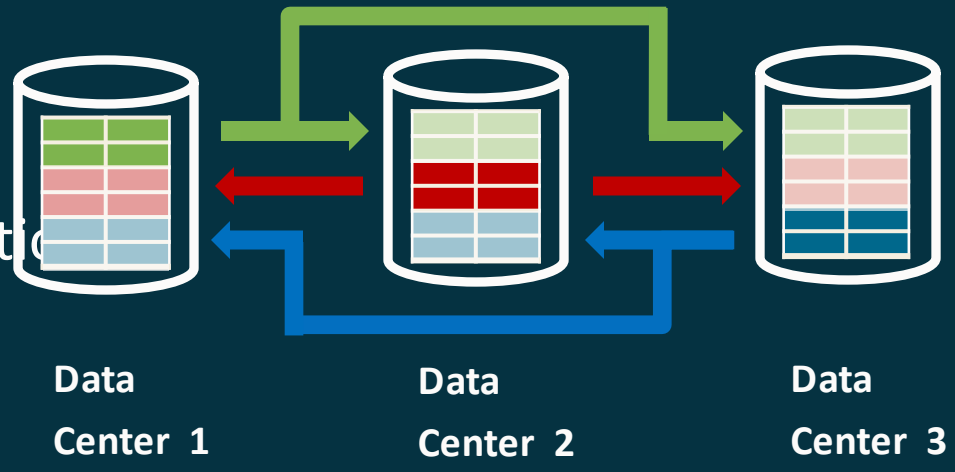
- Provides automatic failover to a replica in under 3 seconds

Implements an active-active symmetric configurati

- Each shard accepts writes and reads for a subset of data

Delivers zero data loss

- Using high-performance synchronous replication across shards



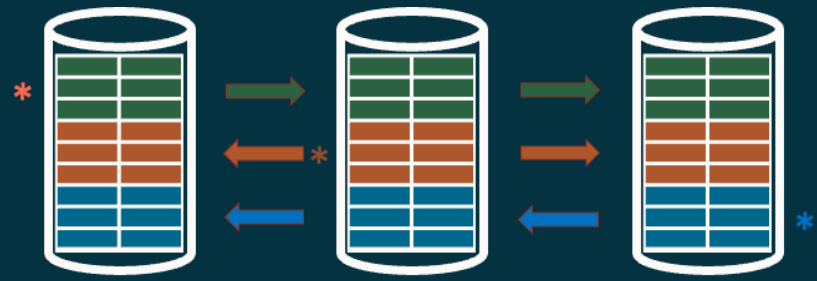
Quorum Based Consensus

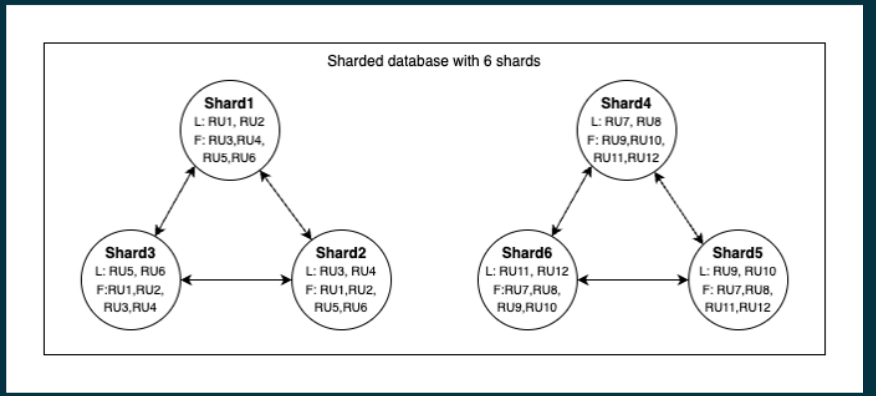
Active-Active Global Scale Database

Database Sharding with Raft Replication

New In 23c

- Built-in replication integrated with transaction execution
- Fast and automatic sub-3-second failover with zero data loss
- Active-active, symmetric configuration
  - Each shard accepts writes and reads for a subset of data
- Easy: no need to configure Data Guard or GoldenGate for shards





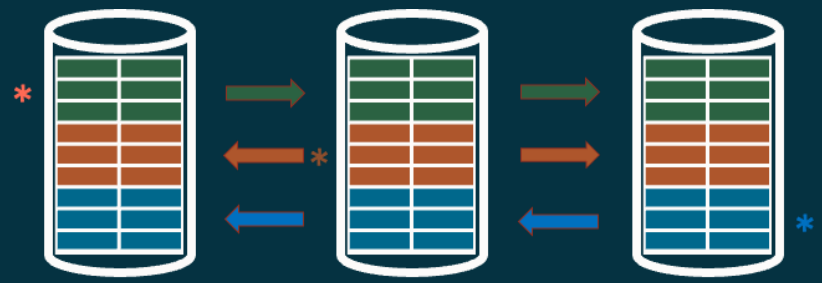
Confidential - Oracle Restricted

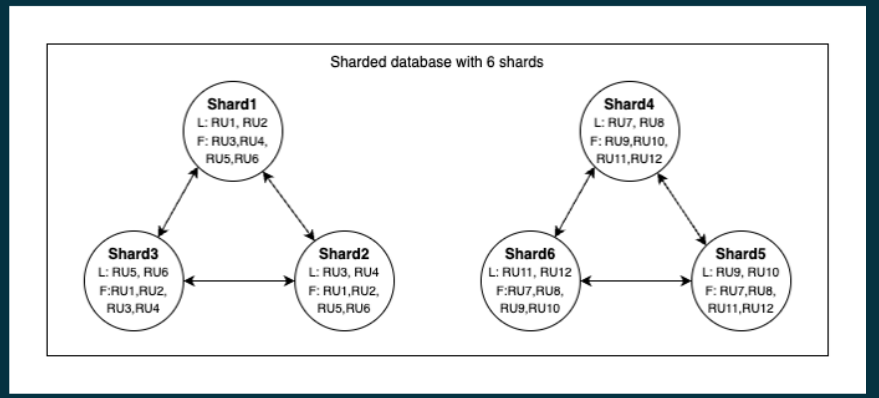
Active-Active Global Scale Database cont…

Database Sharding with Raft Replication

New In 23c

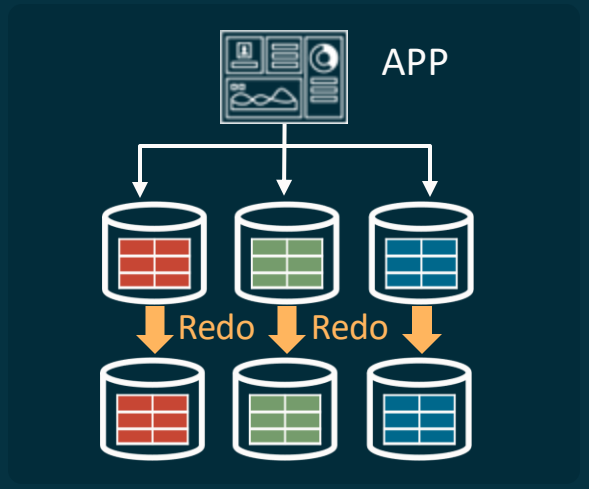
- Sharded database is divided into multiple replication units (RUs)
  - Replicas of RUs are spread evenly across 3 (or more) shards
  - Each shard is (leader) primary for some RUs and follower (replica) for other RUs
- Builds on popular Raft distributed consensus protocol
  - Guarantees consistency among replicas in case of failures, network partition, message loss, or delay
  - Automatic reconfiguration after failure, or when number of shards changes





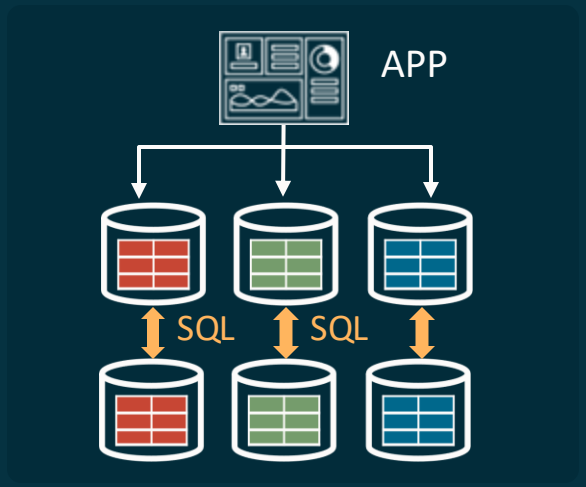
Confidential - Oracle Restricted

Oracle replication can meet any scaling or survivability need



Redo level replication using Active Data Guard provides

- Fastest performance
- Most complete SQL functionality
- Readable replicas
- Simplest operation



SQL level replication using GoldenGate provides

- Fastest failover
- Fully writeable replicas
  - Including conflict avoidance and resolution

Confidential - Oracle Restricted

Oracle has more shard deployment architectures than any other distributed database

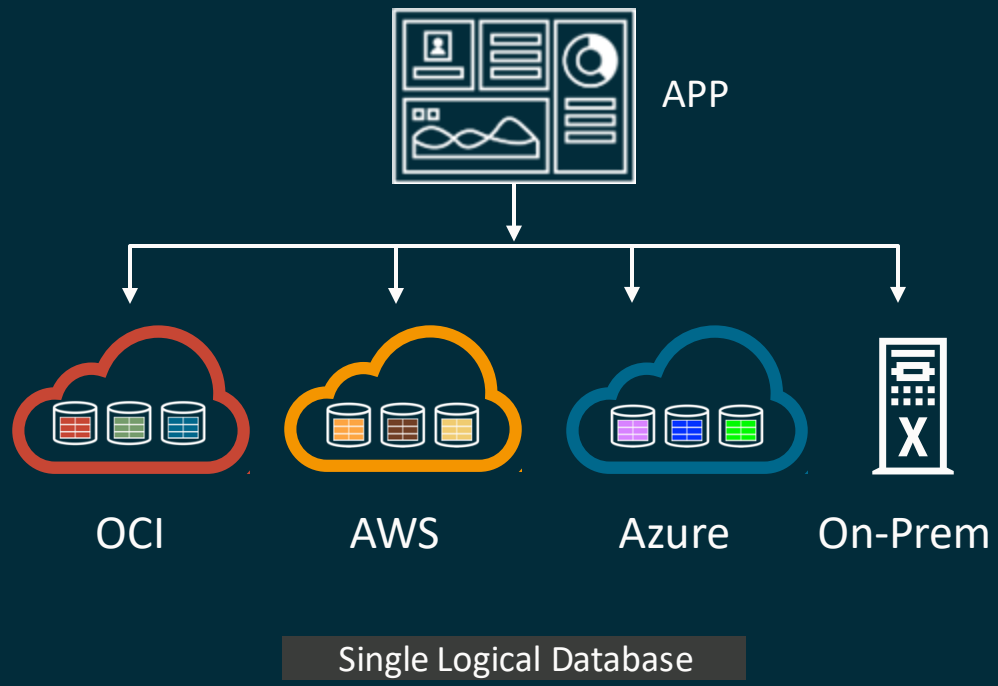
Matching the shard deployment architecture to the needs of the application is critical for a distributed database

Confidential - Oracle Restricted

Oracle can deploy shards across on-prem, cloud, and multiple clouds

You can choose the deployment option independently for each shard or country

- Can optimize for the scaling needs and cloud availability of each country



Oracle's Globally Distributed Database is designed for AI applications

Brings natural language query to Globally Distributed Databases

Autonomous Database Select AI can translate natural language questions into SQL using an AI Large Language Model (LLM)

The SQL query is automatically routed to the appropriate country or shard by the Globally Distributed Database

"How many total streams for each Tom Cruise movie were viewed in India this month?"

LLM Generates

```
SELECT
  m.title AS movie_title,
  COUNT(s.views) AS total_streams
FROM movie m
  JOIN sales s  ON m.movie_id = s.movie_id
  JOIN actors a ON m.movie_id = a.movie_id
WHERE  a.actor = 'Tom Cruise'
  AND  s.country = 'India'
  AND  EXTRACT(MONTH FROM s.day_id) =
       EXTRACT(MONTH FROM SYSDATE)
GROUP BY m.title
```

AI Vector Search and RAG on Globally Distributed Database 23ai

Oracle Distributed database will add Hyper-Scale and Data Sovereignty to Oracle Database 23ai

AI Vector Search

- and to Retrieval Augmented Generation (RAG)

Customers will be able to combine similarity search using AI vectors with search on business data about customers and products in a single distributed query

Oracle Globally Distributed Autonomous Database

Available Now

Confidential - Oracle Restricted

Adds autonomous management to eliminate the operational complexity of Distributed Databases and reduce costs

Globally Distributed Database

+

Autonomous Database

=

Globally Distributed Autonomous Database

The simplest, most functional, most mission-critical cloud native distributed database service in the world

Serverless, elastic, auto-scale architecture dramatically lowers costs

Key Takeaways

Oracle Globally Distributed Database

Ultimate
Scalability and Survivability

Transparent
Data Sovereignty

- Most fully-featured distributed database
- More data distribution, replication, and deployment methods than any other database
- Converged Architecture makes data sovereignty easy for modern apps that use multiple data types and workloads
- Raft-based replication provides fast quorum-based failover
- Supports leading-edge AI such as Vector Search
- Autonomous capabilities remove complexity and reduce cost

Confidential - Oracle Restricted

Resources

Live Labs

- Learn how to achieve Data Sovereignty with Oracle Globally Distributed Database
- Oracle Globally Distributed Database Quick Start
- Oracle Globally Distributed Database for DBA
- Learn how to use Oracle Sharding for mission-critical, internet-scale apps
- Use Raft Replication with Distributed Database for Resilient Never-Down Apps

Documentation

- Oracle Globally Distributed Database
- Globally Distributed Autonomous Database

Website

- Oracle Globally Distributed Database
- GloballyDistributed Autonomous Database

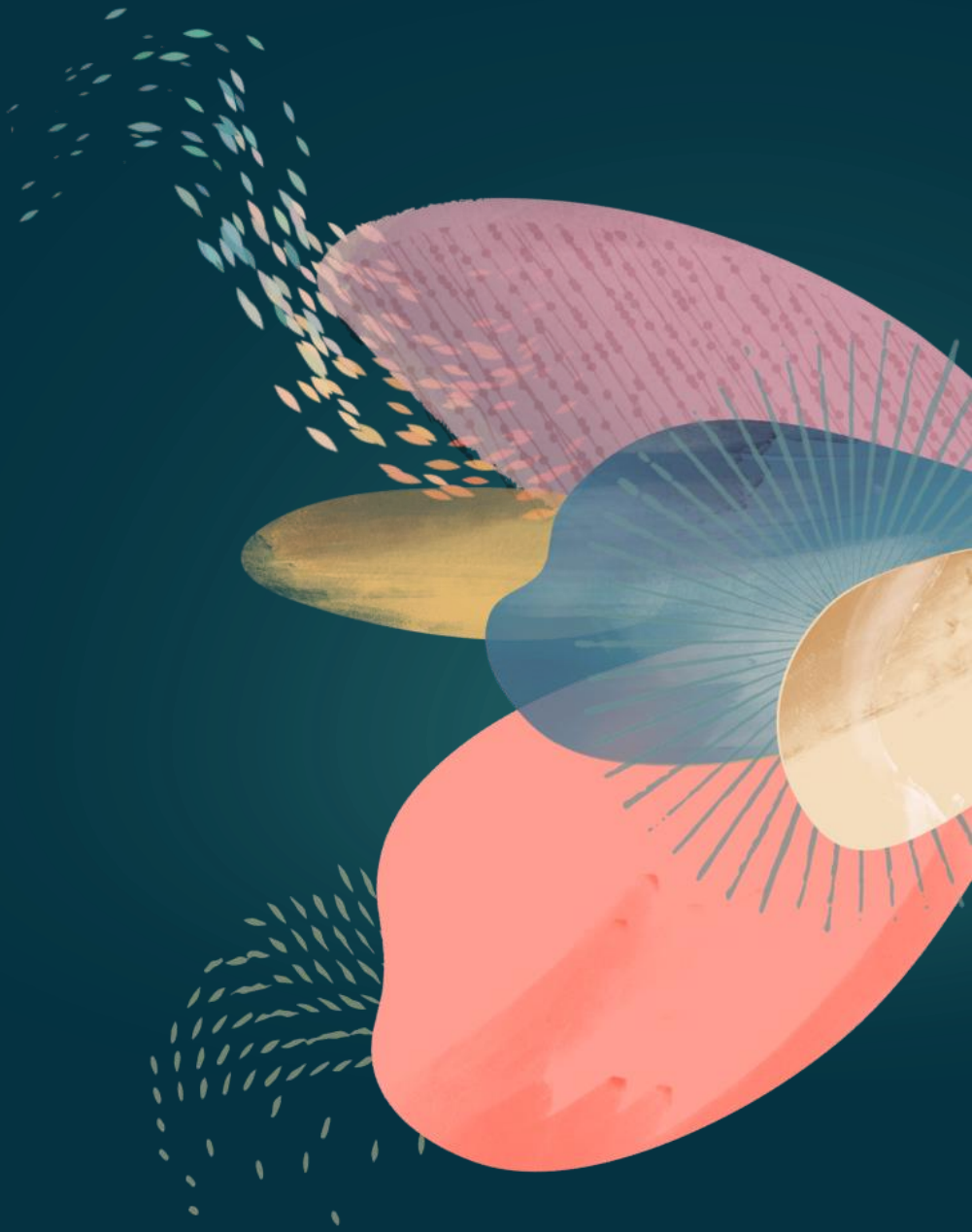
Confidential - Oracle Restricted

ORACLE

Where To Get More Information

Try our Live Labs

- Oracle Globally Distributed Database
- Raft Replication Live Lab
- Habib.ur.Rahman@oracle.com



Accelerate your Applications with True Cache

True SQL and Object Cache

Habib ur Rahman

Principal Product Manager

Oracle Globally Distributed Database and True Cache

Dec 2024

Confidential - Oracle Restricted

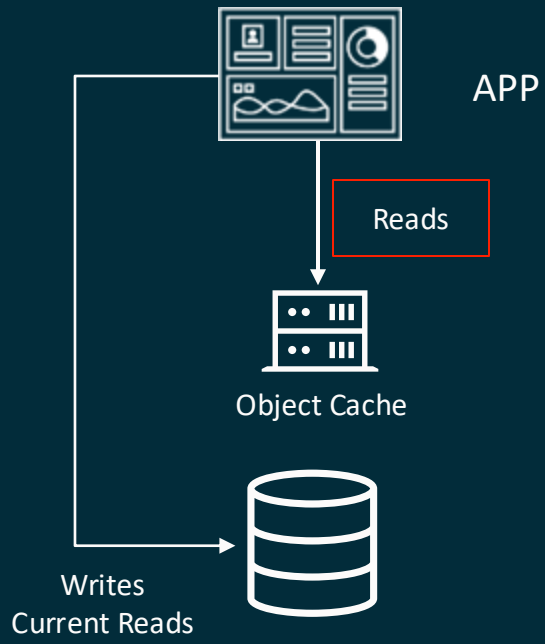
Cache use in Applications

To accelerate performance, applications use a separate cache cluster

- Examples include Redis & Memcached

Reads that do not need the most current data are directed to the cache

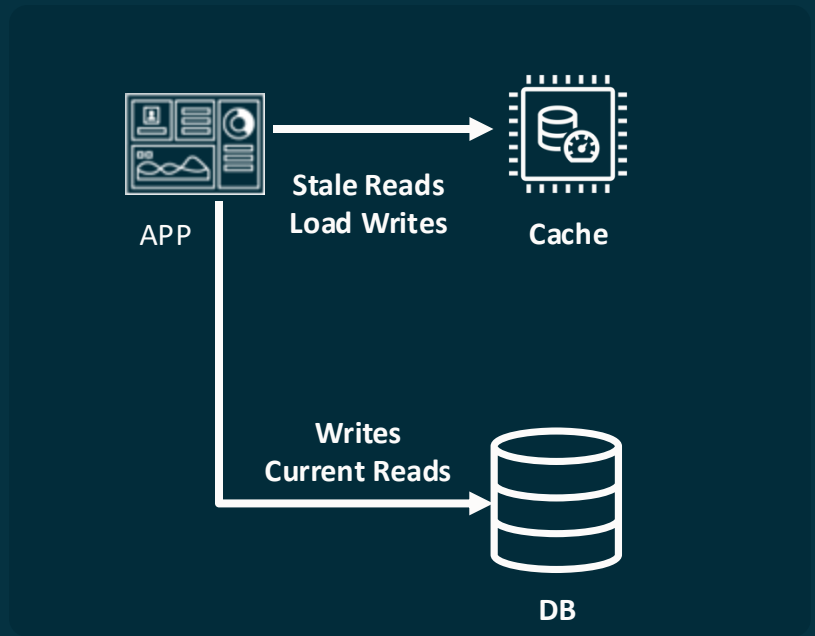
Writes and current reads go to the backend database since that is the source of truth



Developers are responsible for loading the cache

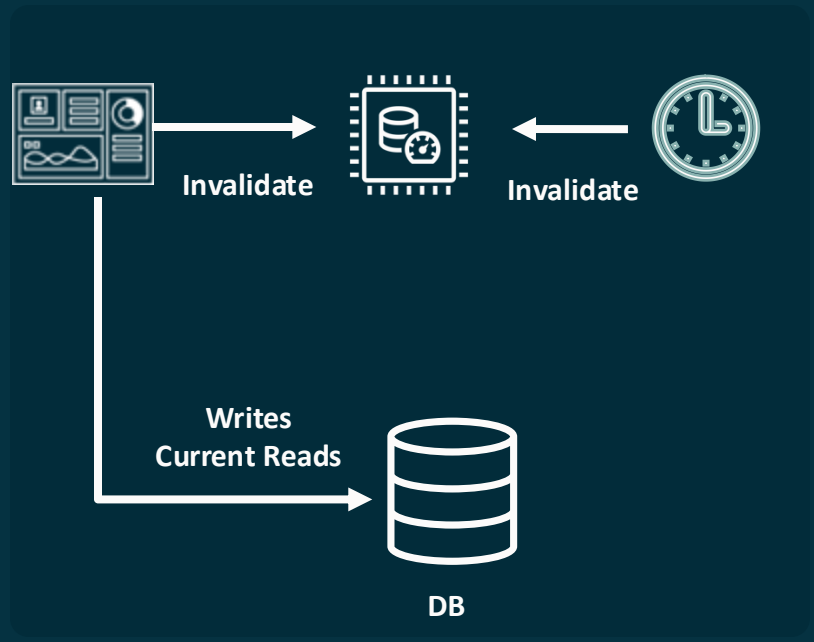
Developers are responsible for loading the cache, either

1. Load the cache at application startup
2. On each cache miss, read the value from the RDBMS, load it in the cache, and then return the data to the app



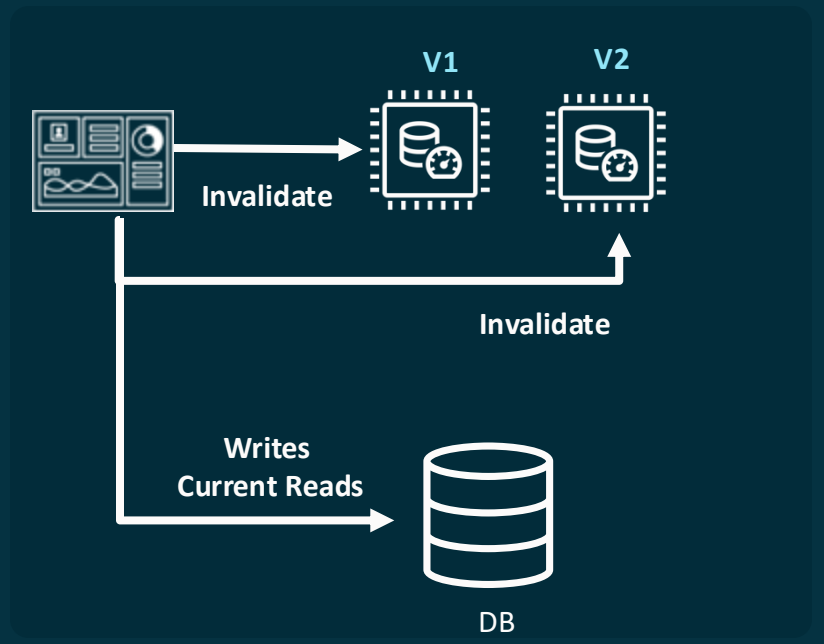
Developers are responsible for cache consistency with DB

- Cached content will become stale as the RDBMS is updated
  - Updates done directly against the database don't update the cache
  - Most apps are not written to manually maintain the cache for consistency and must therefore work directly against the database
- Some caches workaround this by allowing apps to configure Time to Live for different objects
  - Doesn't work well: frequent invalidations reduce cache effectiveness, less frequent invalidations result in a higher chance of getting stale data
  - Difficult to get this right, and keep it right as the application workload changes
- DDLs on RDBMS could invalidate entire cache contents (like purging a partition or dropping a column)



Developers are responsible for cache consistency with other caches

- When you read data from Oracle, we guarantee consistency among multiple values (Consistent Read)
  - Even if the data is updated by some other process
- Even if the application invalidates the cache on updates, it cannot update all the other caches [Cache inconsistent with other caches]
- For data stored in the cache, developers are responsible for enforcing data integrity and consistency [Cache inconsistent with itself]
  - Each object is independently maintained. This means if you read two objects, they are very likely to be from different points in time & hence inconsistent (your accounts won't balance)
- Limits cacheable data: Cannot cache complex nested objects with relationships



Limited data type and data format support

- Only scaler types like string, integer, and JSON supported
- Data cannot be accessed in Columnar formats for Analytics/Reporting
- Oracle database supports a variety of data types like Relational, JSON, Text, Spatial, and Graph and allows access in a row as well as columnar format
- Developers are responsible for handling additional data type conversion

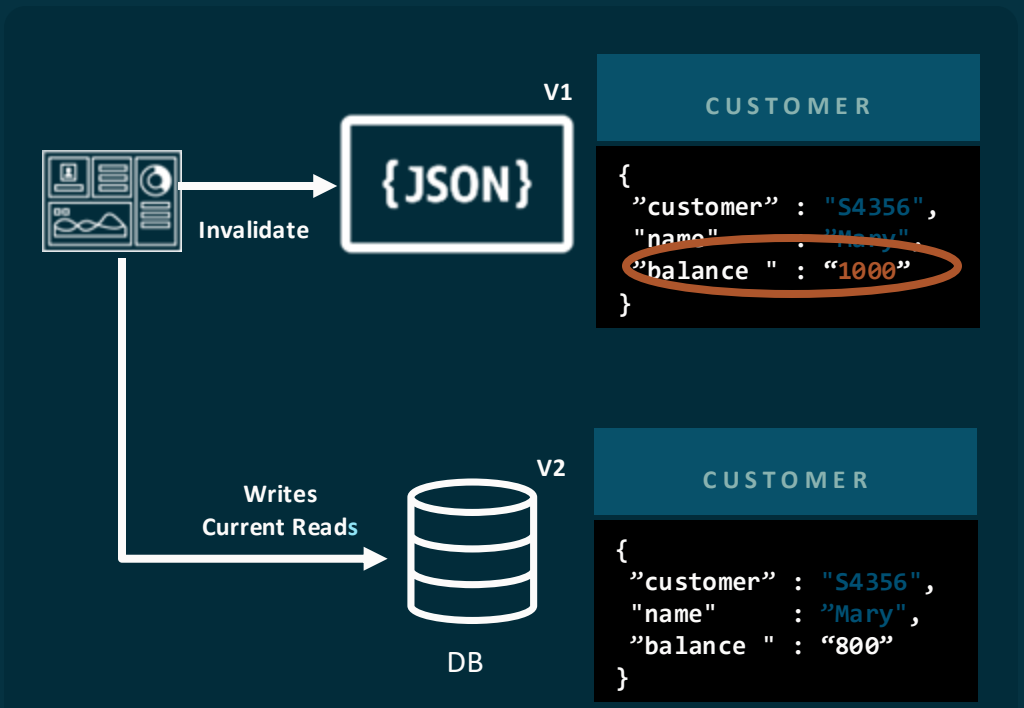
JSON Document exacerbates Caching Problem

When data is stored as JSON documents in an Object Cache, the caching problem gets exacerbated

- When a JSON object is stored in normalized tables, updates to the rows in the underlying tables should result in cache invalidation of the associated JSON attributes. This is hard to do
- Furthermore, need to handle DDL that changes the structure of the underlying tables (e.g., add column, or redefine type of existing column,…).

- No Document/Attribute level security in the cache
  - Cannot store sensitive data in cache

- Developers are responsible for updating and securing JSON documents in the cache



Confidential - Oracle Restricted

Missing Enterprise-Grade Features

- Suboptimal query performance for complex workloads
  - Missing features like sub-partitioning, secondary indices, parallel queries, multi-threading
- Less secure
  - Missing security features like user management, strong encryption, object-level security, row-level security, separation of admin role access
- Lower SLAs
  - Missing high availability features for multi-AZ/Region deployment
- Limited observability and missing enterprise-grade management

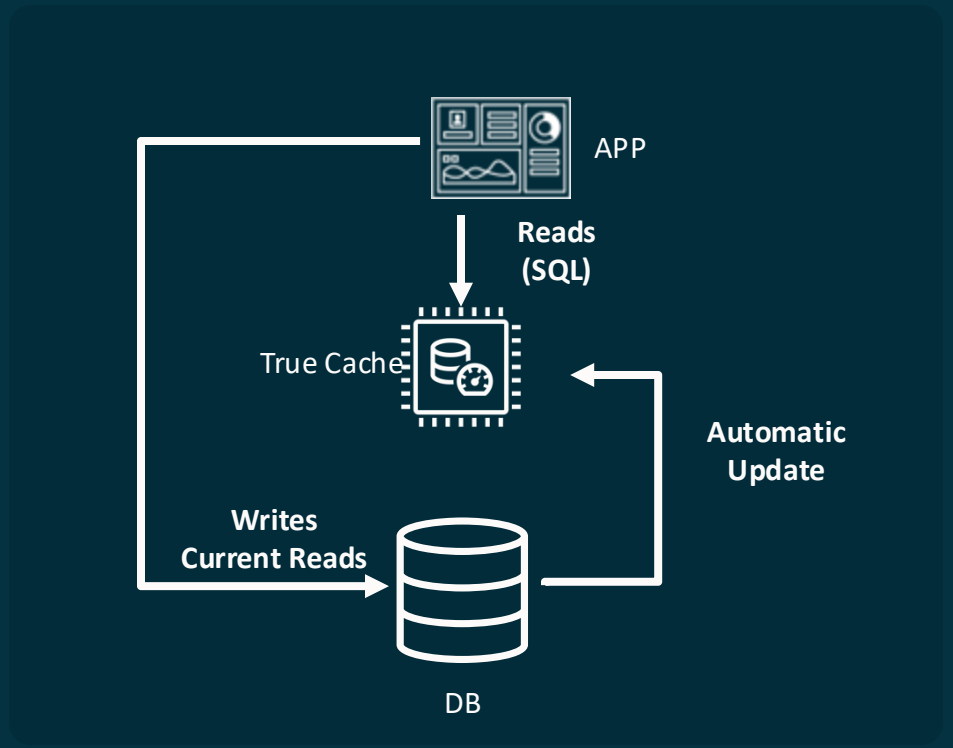
- Developers must write additional logic to compensate for the missing features

Confidential - Oracle Restricted

Oracle True Cache

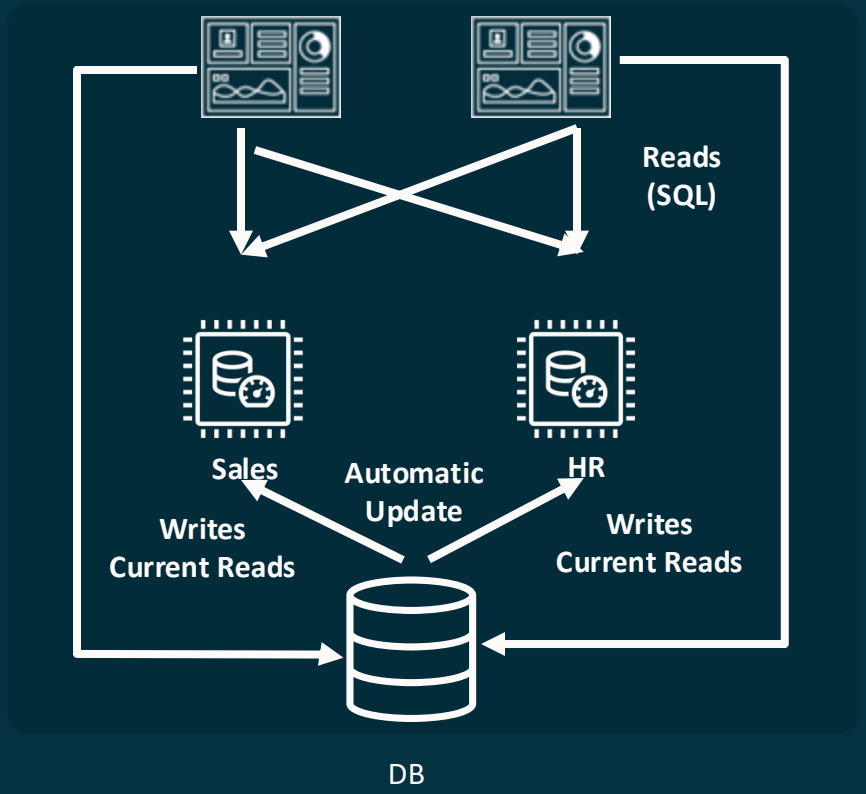
Oracle True Cache

- In-memory, consistent, and automatically managed SQL and key-value (object/JSON) cache
- Deployed as a read-only cache in front of an Oracle database
- Conceptually, a diskless Active Data Guard (ADG)
- True Cache instance is automatically updated
  - It uses Physical redo to apply updates, which is much more efficient than fetching data via SQL
- Queries that can use cached data can be issued to True Cache
  - True Cache can handle all the queries like the Oracle database
  - Data is always consistent across multiple tables and objects
  - DDLs automatically propagated to the cache



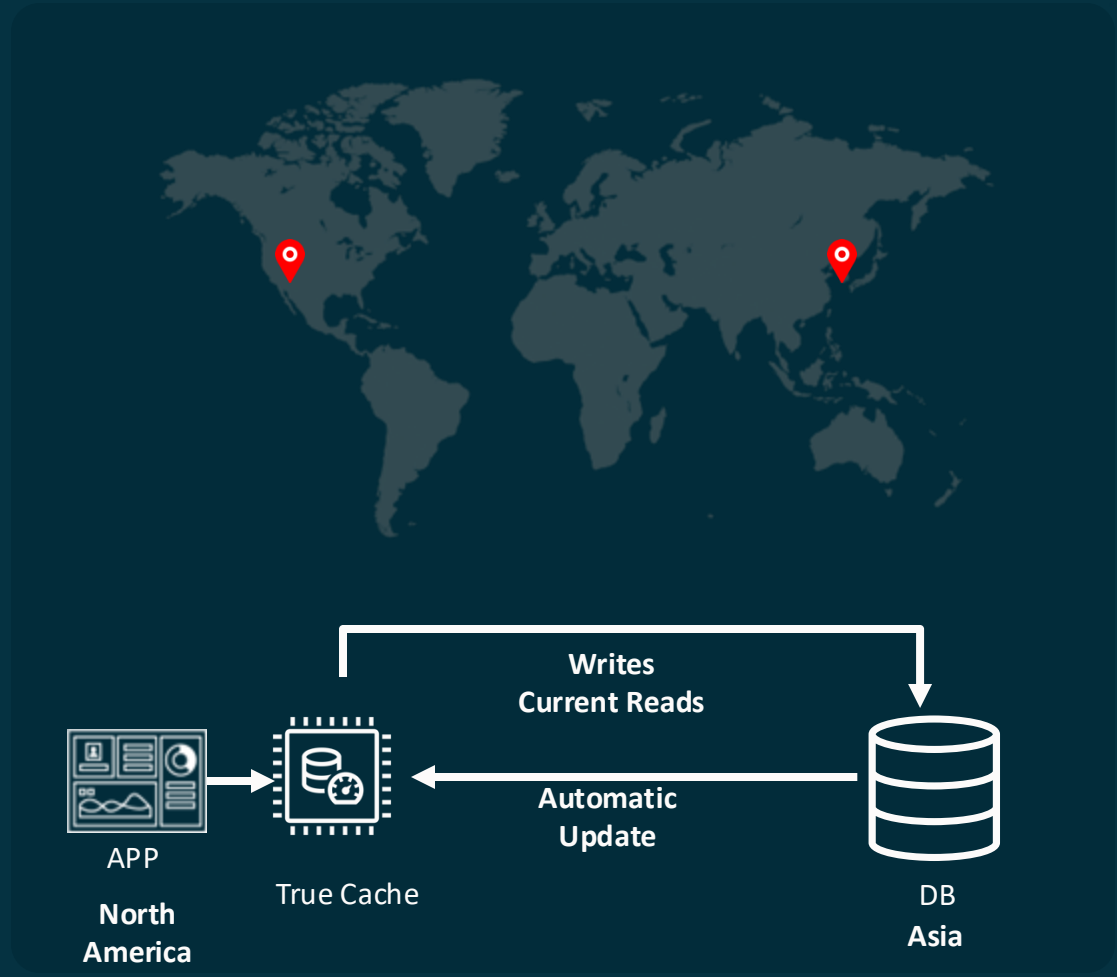
Scaling True Cache with partitioned configuration

- Different True Cache instances can support different database services
- Different True Cache instances cache different sets of data
- The aggregate cache size of all True Cache instances can be much bigger than that of the primary



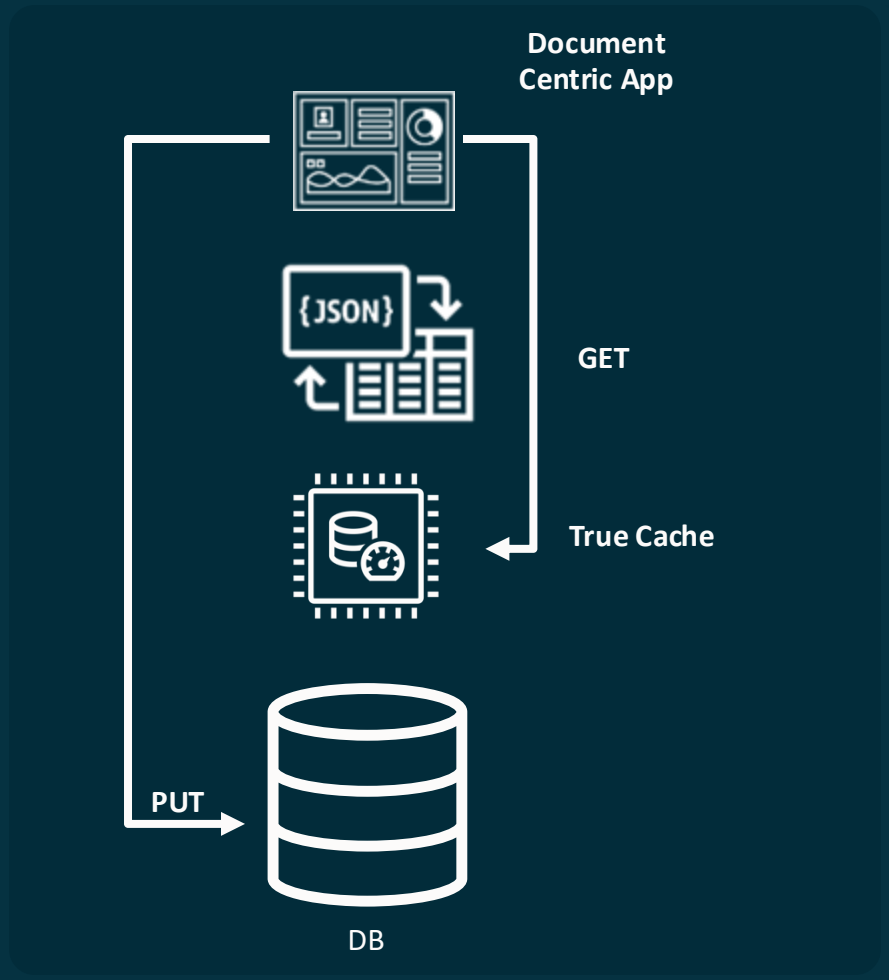
Global Proximity to Application

- Cache can be deployed closer to the Application if DB is in a remote location
- This is useful for Data Residency use cases
  - True Cache does not store any data on disk; hence is a data-processing tier
- This helps boost read performance significantly



True “Document and Relational” Cache

- With the new Document and Relational duality capabilities of Oracle database, developers can easily build document-centric apps
  - That stores new documents as relational data
  - Operate on existing relational data as documents
  - Create JSON microservice on top of a relational database
- JSON documents can be read directly from True Cache
  - GETs can be offloaded to the cache
  - PUTs go to the primary



True Cache Mitigates Deficiencies of conventional caches

Operation	Conventional Caches
Loading the cache	Developer responsibility
Cache consistency with DB	Developer responsibility
Cache consistency with values in the same cache	Developer responsibility
Cache consistency with other caches	Developer responsibility
Complex data type support	Developer responsibility
Full JSON support	Developer responsibility
Comprehensive security	Developer responsibility
Parallel processing	Developer responsibility
High Availability	Developer responsibility

Confidential - Oracle Restricted

True Cache mitigates deficiencies of conventional caches

Operation	Conventional Caches	True Cache
Loading the cache	Developer responsibility	Automated
Cache consistency with DB	Developer responsibility	Automated
Cache consistency with values in the same cache	Developer responsibility	Automated
Cache consistency with other caches	Developer responsibility	Automated
Complex data type support	Developer responsibility	Automated
Full JSON support	Developer responsibility	Automated
Comprehensive security	Developer responsibility	In-Built
Parallel processing	Developer responsibility	In-Built
High Availability	Developer responsibility	In-Built

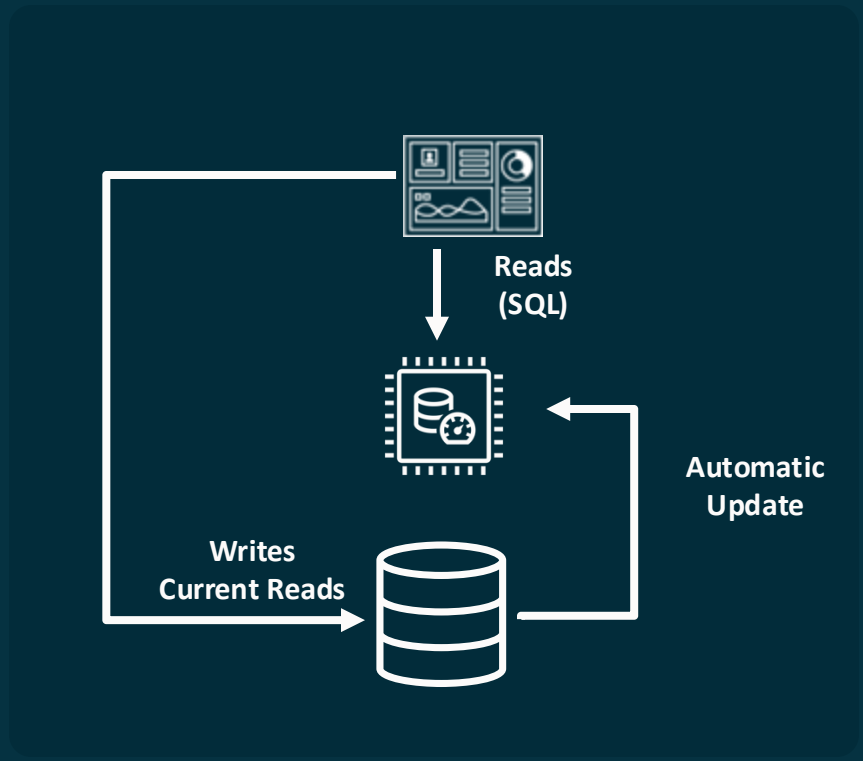
Confidential - Oracle Restricted

How To Use True Cache

The application can query data from True Cache in the following two modes

1. The application maintains two connections:
   - A read-only connection to True Cache and a read-write connection to the primary, and uses the read-only connection for offloading queries to True Cache
2. The application maintains one connection.

- The JDBC driver maintains two connections and does the read-write split under application control
- The application uses setReadOnly() calls to mark some sections of code as “read-only”
- This is an existing JDBC API which MySQL already uses for similar purposes
- JDBC driver will send read-only queries to True Cache and DMLs and DDLs to the primary



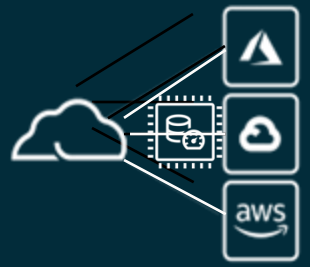
True Cache Use Cases

True Cache can be deployed as a:

Mid-Tier Cache

Edge Cache

Cross-Region Cache



Cross-Cloud Cache

Benefit To The Business

Cost Savings

- Improve application performance without rewriting the applications
- No investment in additional caching products and skills
- Single cache for different data types and formats (document, relational, row, columnar)
- Offload caching to commodity hardware from the database (which might be on more expensive hardware)

Stronger security, which comes with Oracle database

Leverage the full power of Oracle Database

Confidential - Oracle Restricted

Customer Feedback

Stock Exchange – Offload read queries for stock tickers to True Cache with 10 TB of cached data for a 200 TB database.

6 True Cache instances (for load balancing) are being deployed, which would have required a 200-node Redis cluster

Mobile Phone Manufacture – Offload read-only queries to True Cache as the primary is maxed out with vertical scaling

Financial Institution – Offload fraud detection application's AI Model Inferencing to True Cache as models are trained once a day, and most fraud detection queries are read-only.

Marketing – Would like to use True Cache for real-time marketing campaigns. Data in conventional caches is stale and does not allow for real-time tweaking of marketing campaigns.

Oracle Fusion Applications – Cache for Business Object layer for objects that do not change as often

Oracle Banking Apps – Testing to offload Reads to Cache for the core banking application, which is a read-intensive application

Confidential - Oracle Restricted

Key Takeaways

Oracle True Cache

- Most fully-featured SQL and Object Cache
- Automated Data Refresh
- Full Oracle Database Capabilities
- Supports leading-edge AI such as Vector Search to enable Semantic Cache

A True Cache with less effort

Why settle for less?

For More Information

oracle.com/database/truecache