



Data and Application Modeling in the Brave New World of Oracle Sharding

NoCOUG @ PayPal – Spring 2019

John Kanagaraj, Sr. MTS, Architect @ PayPal • 09-May-2019

(Previously presented at Oracle OpenWorld 2018 @ SFO)

Agenda

Introduction

Sharding – What/Why/How

Data Model and Other Considerations

Introduction to Oracle Sharding

Wrap up: Q & A

Setting the Stage

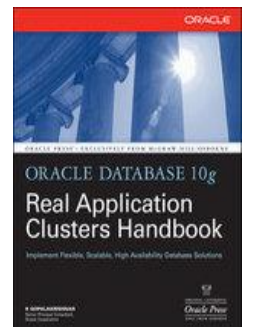
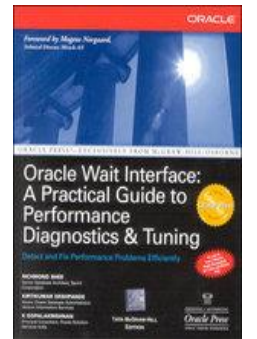
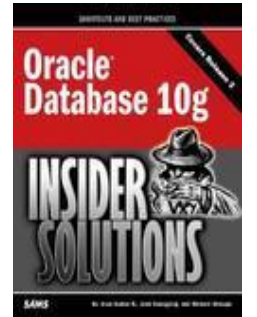
- Audience Survey
 - Facing Data Scaling challenges?
 - Using Oracle (or other) Sharding methods?
 - Considering Sharding as a scaling method?
- Scope
 - What exactly is Sharding?
 - What are the cons?
 - What is the process to evolve to a Sharded system?
- What we will not get into (in detail, that is!)
 - Oracle Sharding features (Tons of other presentations for this!)

Speaker Qualifications

- Currently Sr. Database/Data Architect @ PayPal
- Has been working with Oracle Databases and UNIX for 3+ decades
- Working on various NoSQL technologies for the past 4 years
- Has worked on many Sharded applications – Both Oracle and NoSQL
- Author, Technical editor, Oracle ACE *Alumni*, Frequent speaker
- Loves to mentor new speakers and authors!
- <http://www.linkedin.com/in/johnkanagaraj>



ORACLE
ACE Alumni



Two decades ago, our founders invented payment technology to make buying and selling faster, secure, and easier; and put economic power where it belongs: **In the hands of people**

About PayPal



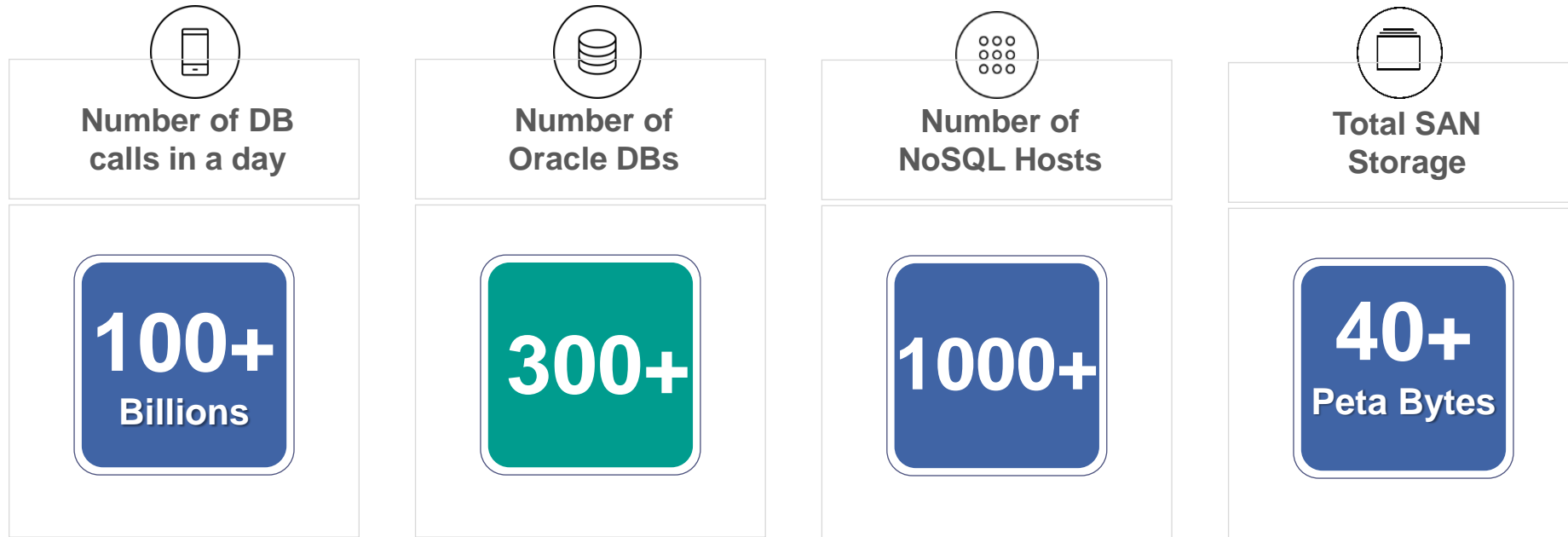
Our **250+** Million consumers can accept payments in **> 100** currencies, withdraw funds to their bank accounts in **56** currencies, hold balances in their PayPal accounts in **25** currencies and interact with **20M+** Merchants across **19K+** corridors



Almost **8000** PayPal team members provide support to our customers in over **20** languages

We are a trusted part of people's financial lives and a partner to merchants in **200+** markets around the world

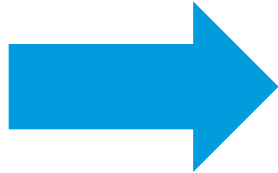
Database Footprint



Scaling is Hard!

Challenges at Scale

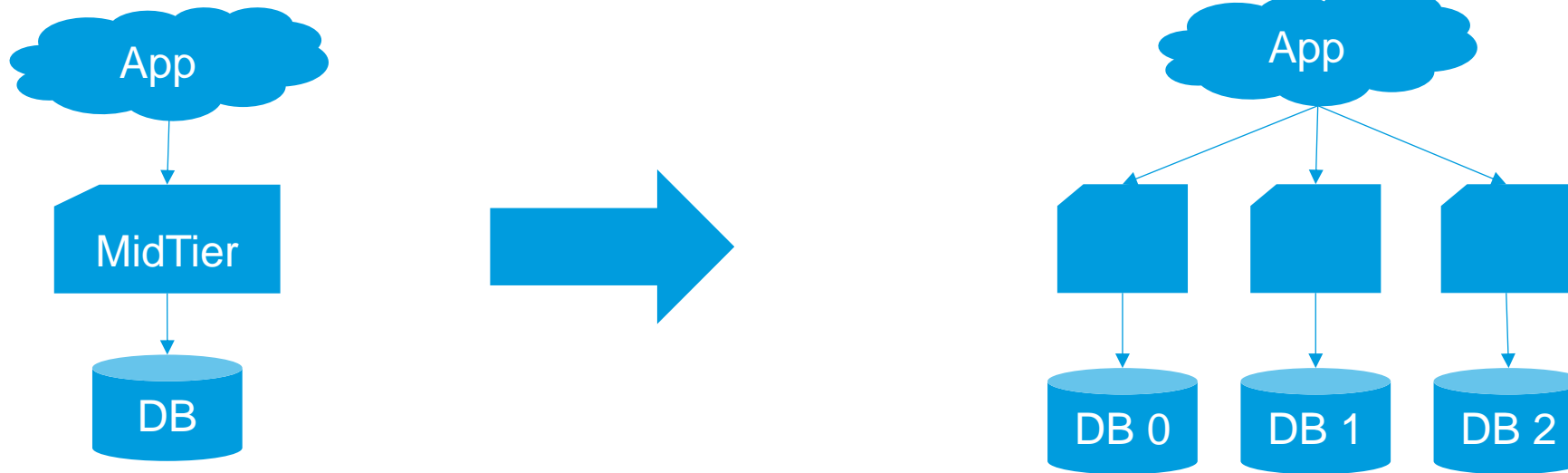
- Pushing the limits
 - Connections
 - Memory
 - Interconnect
 - CPU
 - DDL on busy tables
 - RAC reconfiguration
 - Redo rate
 - I/O latencies
 - SAN Storage limits
 - Replication latencies



- Solutions
 - Separate reads for RO scaleout (ADG/GG)
 - Split by Microservices/Data domains
 - Connection multiplexing
 - Write isolation
 - Custom caching
 - And finally
 - Sharding!

Scaling is Shard!

What exactly is Sharding?



- App code addresses a single database
- Requires connection management layer
- Data model can be complex/multi-level
- Object relationships fully supported
- All Joins and Transactions fully available
- Scaling up is only option

- Objects chopped into pieces (Shard)
- Shards placed on multiple database
- All objects spread evenly* on shards
- New challenges at multiple levels:
 - Data model
 - Joins and Transactions
 - Connection and SQL routing
- Introduces “rigidity”

Sharding is Hard!

Sharding Challenges

Data Model and Transactions

Type of Sharding: System, List, Range

Choice of Key to Shard

Access Pattern and Transaction limitations

Relaxing Normalization

Rigidity in future data model changes

Joins and SQL

No cross shard joins allowed

No cross shard SQL

“Sharded” SQLs need Shard key

“Sharded” vs. “Non Sharded” SQL

Rigidity in SQL evolution for supporting future requirements

Connections and routing

Methodology used to distribute data equally among all buckets

Connection pool exhaustion per shard

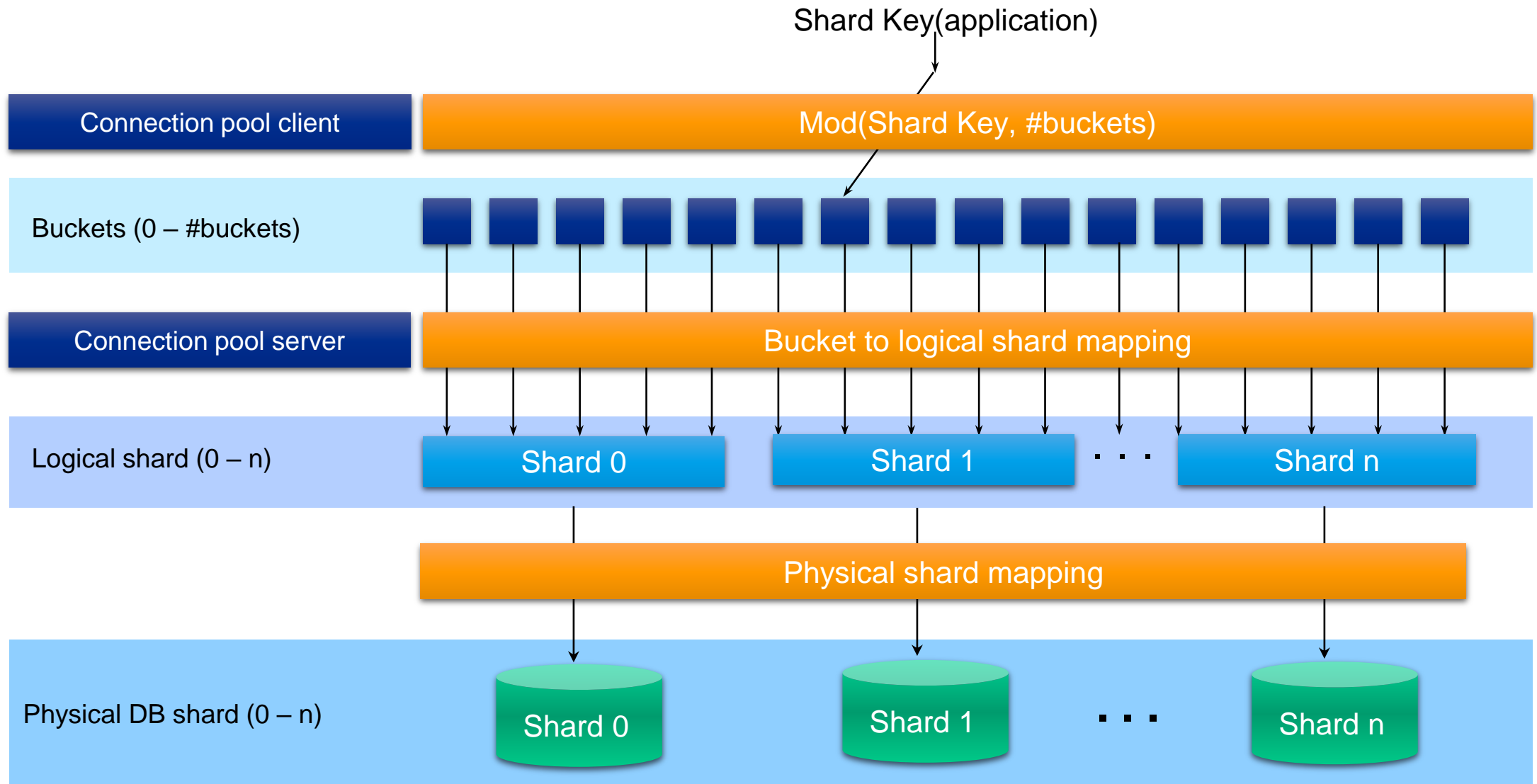
Scatter-gather resiliency model

Separation of access by secondary keys

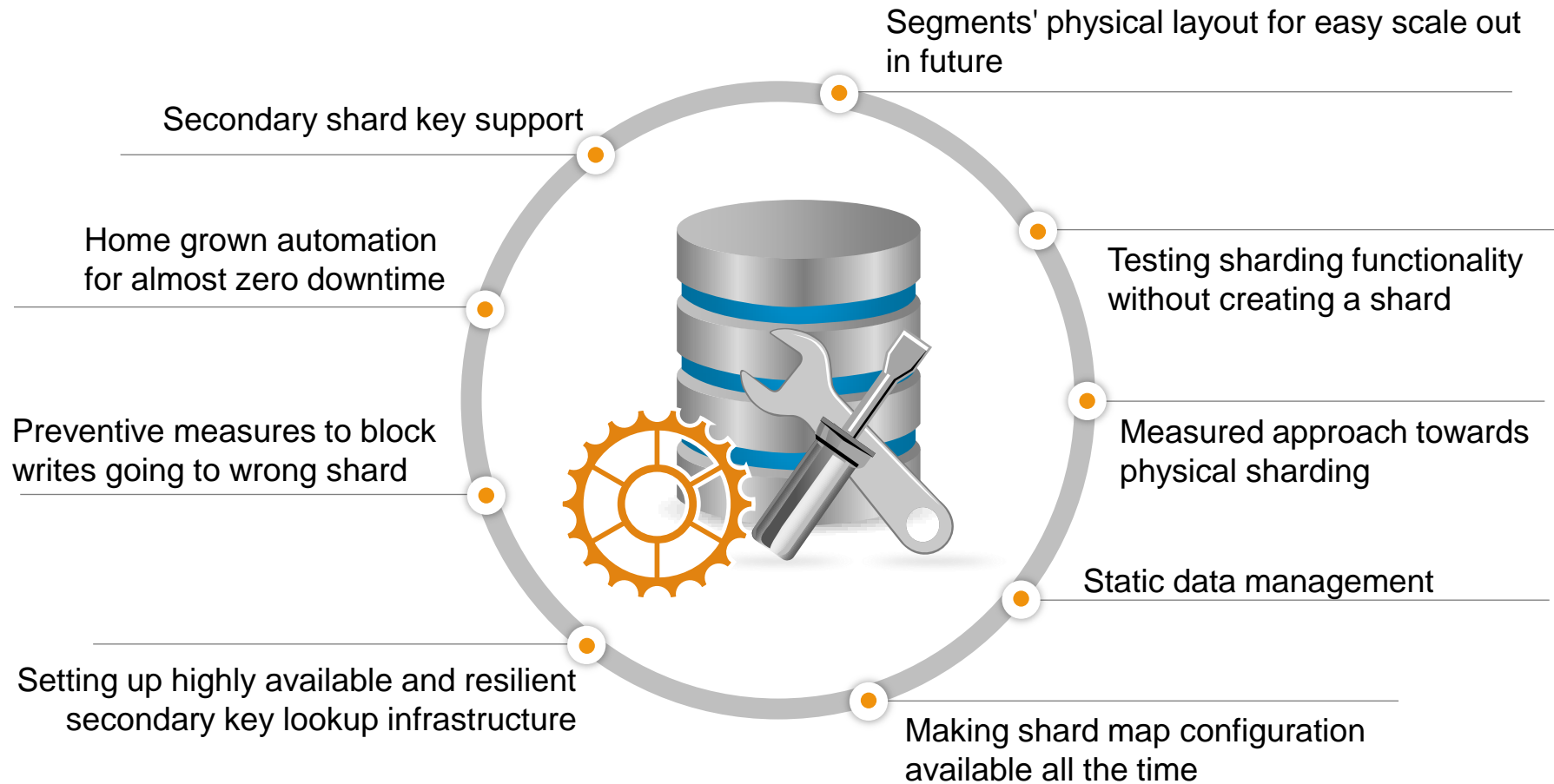
Monitoring enhancements for tracking each bucket and shard traffic

Key is Key!

Sharding Framework



Sharding Implementation Challenges



Sharding Principles

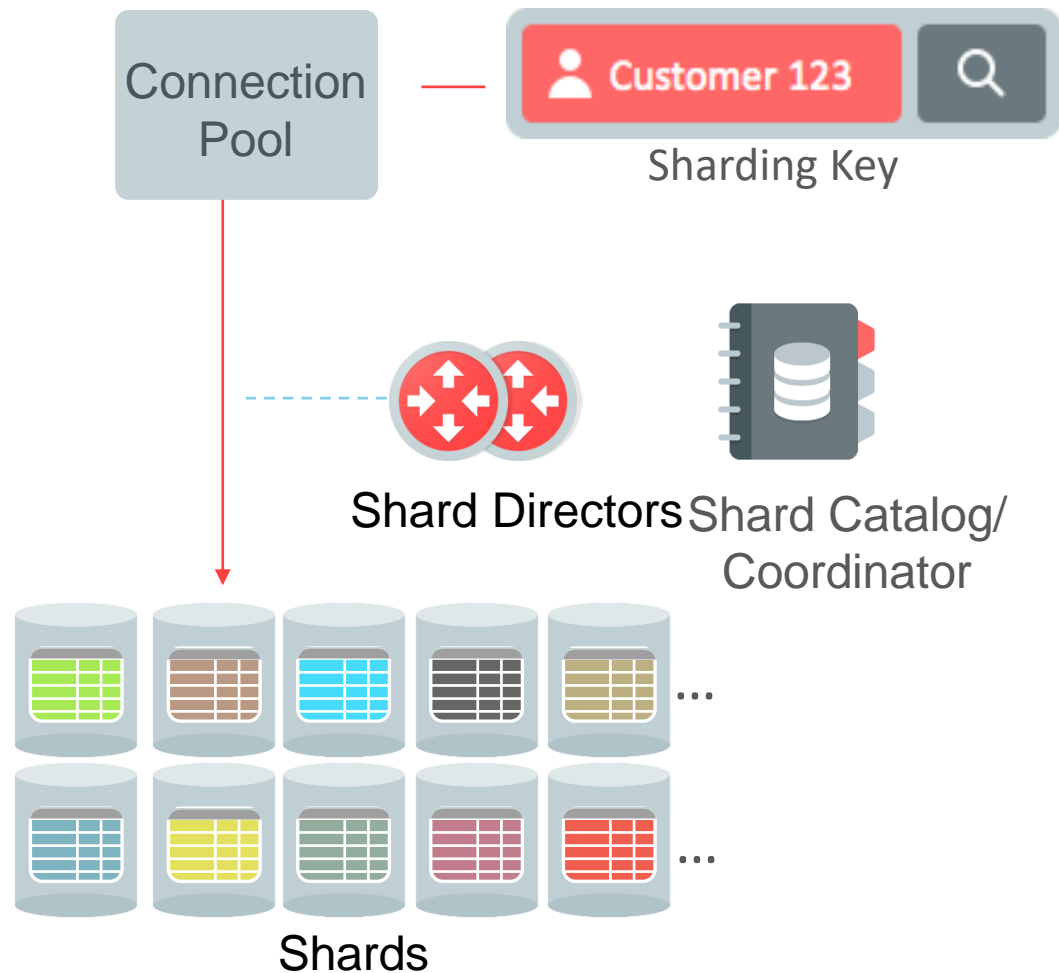
- Sharding is mostly a Scalability play
 - Also enables Data Sovereignty and Data Proximity
- "Key is Key" – Choose one STRONG key and align. E.g. Account/User ID
 - Majority of access patterns should be via Shard Key
 - Non-shard key access can be expensive; forces "scatter-gather" pattern
- Data model challenges: CAP Theorem applies (2 or 3!)
 - Normalization is relaxed
 - No cross-shard transaction/query
 - Joins and ACID principles usually not available in Sharded systems (e.g. NoSQL)
- "Lookup" requirement for Common data elements – Replicate to every shard
- Scheme for mapping logical to physical is critical for future scale-out

Sharding is Hard!

But Oracle Sharding makes it easier!

Oracle Sharding | Architecture & Key Features

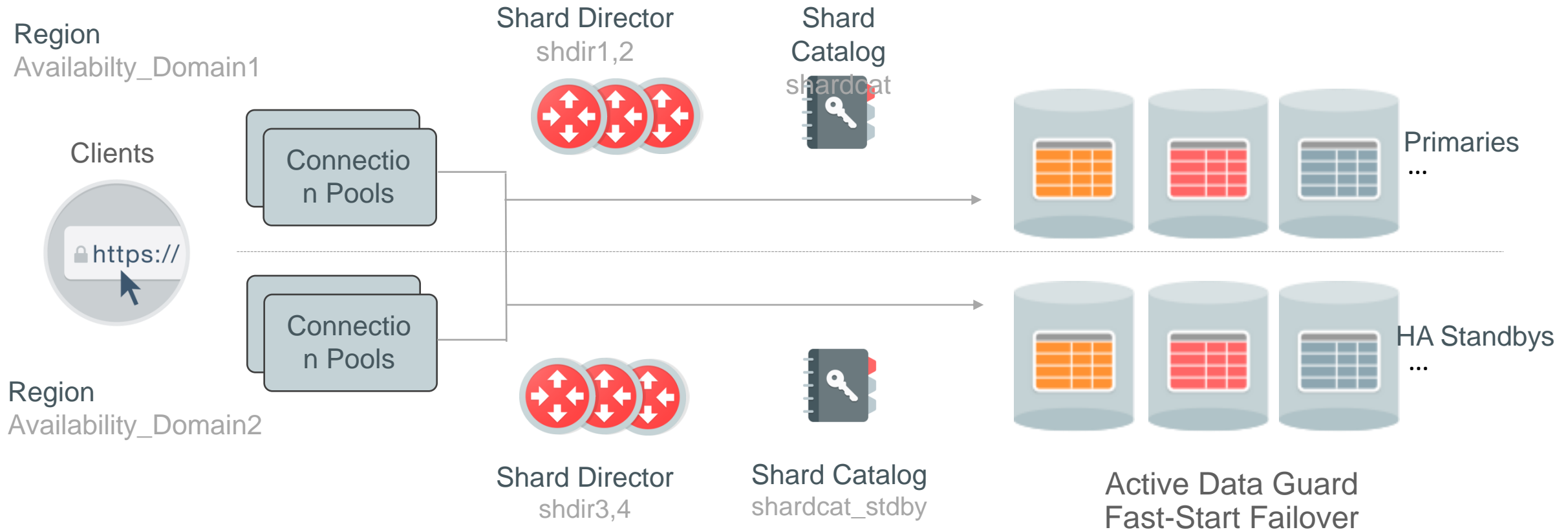
Slide Courtesy Oracle Corp



- Automated deployment of up to 1000 shards with replication (Active Data Guard and Oracle GoldenGate)
- Sharding methods
 - System-managed, User-defined Composite
- Centralized schema maintenance
 - Native SQL for sharded and duplicated tables
 - Relational, JSON, LOBs and Spatial support
- Direct routing and Proxy routing
- Online scale-out w/auto resharding or scale-in
- Midtier Sharding
 - Scale midtiers along with shards
- RAC Sharding
 - Gives a RAC DB, the performance and scalability of Oracle Sharding with minimal application changes

Oracle Sharding | Maximum Availability Architecture

Slide Courtesy Oracle Corp



Sharding: Typical Use Cases and Anti-patterns

Use Cases

- Large user-base facing application
 - Rapid and viral growth anticipated
 - Ever-increasing growth Y-o-Y
 - High data quality, effective SQL access
 - Scale out required for read and write
 - High Availability requirements
- “Fast Data” use cases
 - High / Fast Write rate
 - IoT : “Internet of Things” workload

Anti-patterns

- Complex data model with many relationships
- If Sharded design, presence of
 - Multiple “strong” keys or lack of strong key
 - Evolving data model not in line with original key
 - Complex/large transaction requirements: consistency
 - Large percentage of “scatter-gather”, secondary key access
 - Active common workload that needs replication



Provide feedback via the Mobile App!

Link up with me on LinkedIn
John Kanagaraj, PayPal

Oracle Sharding Sessions and Demos in OOW 2018

Slide Courtesy Oracle Corp



Monday, Oct 22nd

9 AM | High Availability and Sharding Deep Dive with Next-Gen Oracle Database [TRN4032] | **Moscone West-3007**

3.45 PM | Industrial-strength Microservice Architectures with Next-Gen Oracle Database [TRN5515] | **Moscone West-3003**

Tuesday, Oct 23rd

12:30 PM | Oracle Sharding: Geo-Distributed, Scalable, Multimodel Cloud-Native DBMS [PRO4037] | **Moscone West-3007**

3.45 PM | Data and Application Modeling in the Brave New World of Oracle Sharding [BUS1845] | **Moscone West-3007**

Thursday, Oct 24th

11 AM | Oracle Maximum Availability Architecture: Best Practices for Oracle Database 18c [TIP4028] | **Moscone West-3006**

1 PM | Using Oracle Sharding on Oracle Cloud Infrastructure [CAS5896] | **Moscone West-160**



Sharding Demo Booth - 1615

High Availability, Moscone South Exhibition Hall

Contact:

Nagesh Battula
Sharding Product Management
srinagesh.battula@oracle.com



@OracleSharding & @NageshBattula
<https://www.oracle.com/goto/sharding>

