ORACLE

Oracle Database Reliability

In the context of distributed databases

Francisco Munoz Alvarez

Distinguished Product Manager

Oracle Database High Availability (HA), Scalability and Maximum Availability Architecture (MAA) Team

 @fcomunoz

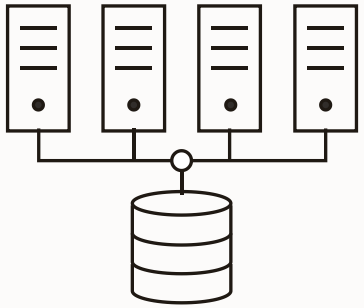
 http://www.linkedin.com/in/franciscomunozalvarez

 www.oraclemaa.com

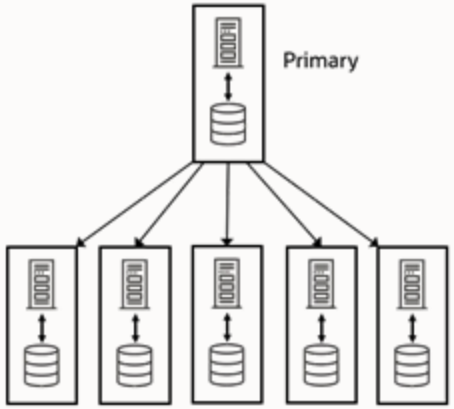
Some key concepts

High Availability (HA) and Scalability Concepts

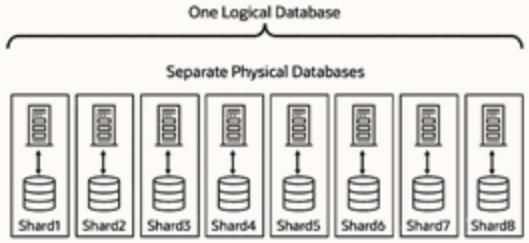
Shared Disk / Shared Cache



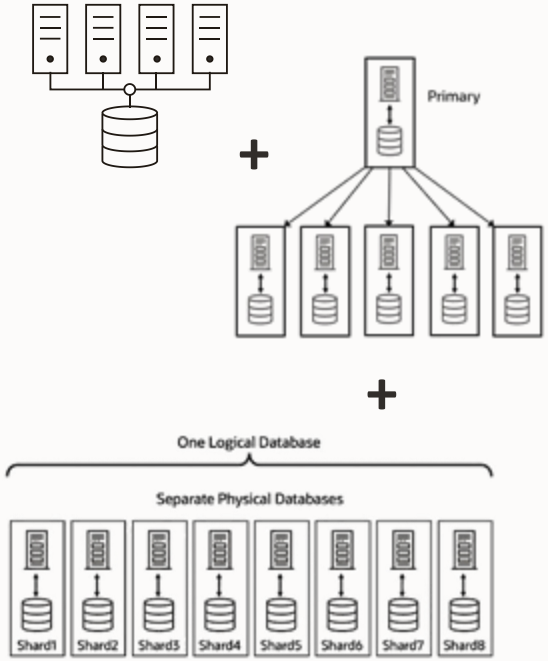
Replication / Read Replicas



Sharding

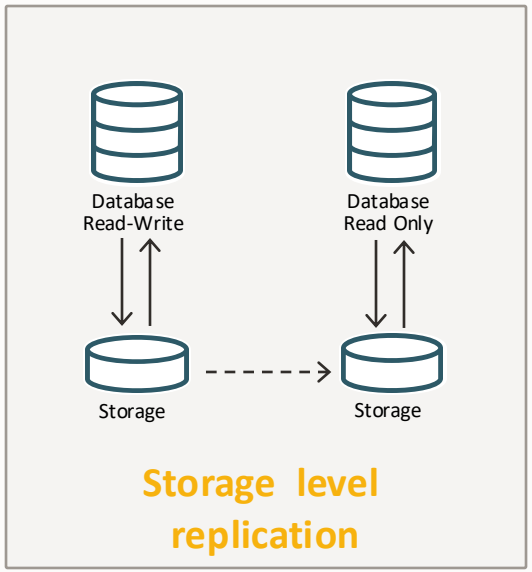


Combination thereof



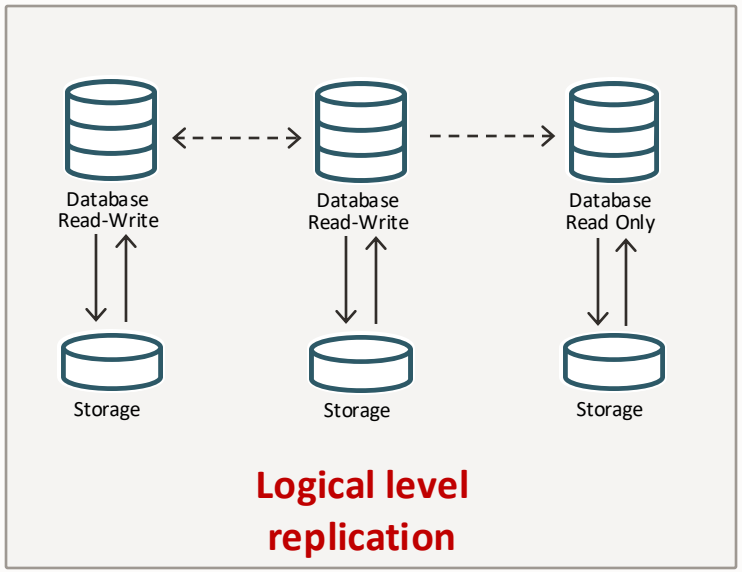
https://blogs.oracle.com/exadata/horizontal-scaling-with-oracle-database

Replication Types



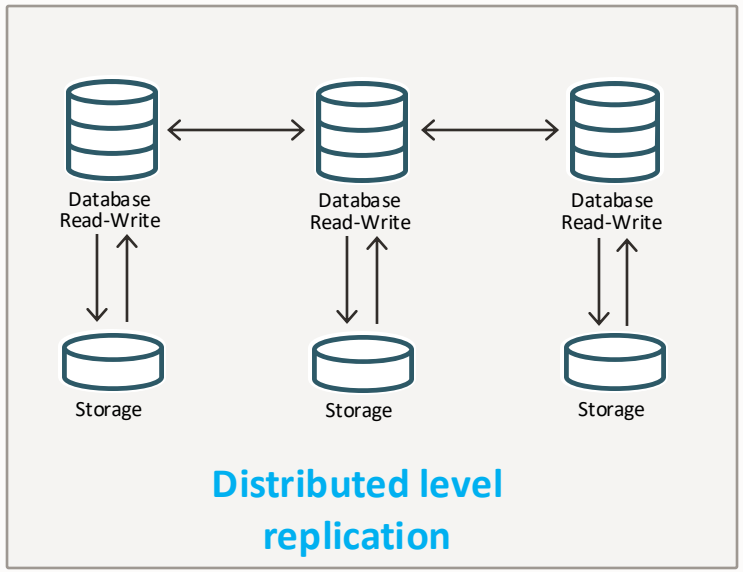
Storage level replication

AWS Aurora
AlloyDB



Logical level replication

Oracle
PostgreSQL
SQL Server



Distributed level replication

Spanner
CockroachDB
YugabyteDB
TiDB

https://blogs.oracle.com/maa/post/about-the-usefulness-of-redo-apply-in-the-storage-tier

Same Concepts – Different Names

New In 23ai

	Leader Database	Voting Database	Read-only Follower Database	Active-Active Replicated Database (***)
Oracle DB	Primary	Node (RAC), Standby (Sync Affirm)	Read Standby (Active Data Guard)	Multi-Active Replica (GoldenGate)
Oracle Sharding (with RAFT replication)	Leader (Raft*)	Replication Unit (RU), Replication Unit Leader	RO-Standby N/A	Active-Active DB N/A
CockroachDB	Leader (Raft*)	Replica, Lease-Holder	Non-voting Replica	N/A
YugabyteDB	Master (Raft*)	Follower, Lease-Holder	Read Replica /Observer Node (Not for DR) xCluster Replication (For DR)	N/A
Google Spanner	Leader (Paxos**)	Read-Write Replica, Witness Replica, Lease-Holder	Read-only Replica	N/A

(***) Asynchronous active-active replication between independent Clusters

* https://en.wikipedia.org/wiki/Raft_(algorithm) ** https://en.wikipedia.org/wiki/Paxos_(computer_science)

RAFT Protocol in SQL Distributed Databases

RAFT is a consensus algorithm that ensures a group of nodes agree on a series of log entries (like SQL transactions). Each node in a cluster has one of three roles:

- **Leader**: Handles all client writes.
- **Follower**: Replicates entries from the leader.
- **Candidate**: Can be elected leader in a new term if the current leader fails.

Key Phases:

- **Log replication**: Leader appends entries and replicates to followers.
- **Leader election**: Triggered if the leader is unresponsive. A majority is required for an election.

RAFT Protocol in SQL Distributed Databases

RAFT Leader Election Process:

1. A follower detects the leader’s unresponsiveness and becomes a **candidate**.
2. It increments the term and sends vote requests.
3. Other nodes respond with votes.
4. If a majority is reached, the candidate becomes the new **leader**.
5. The leader resumes processing log entries.

Impact on Availability:

- During the election, writers are unavailable.
- Short timeouts cause instability; long timeouts delay recovery.

Some SQL Distributed Databases Comparison

Feature	CockroachDB	YugabyteDB	TiDB
RAFT Implementation	Custom per-range RAFT	DocDB RAFT with strong sync	Per-region RAFT groups
Leader Election Time	~300ms (tunable) – whole process up to 9s	~150- 300ms (adaptive) – whole process up to 3s (TCP Timeouts can take up to 15s)	~200ms – whole process up to 20s
Multi-leader Write Support	No (single per range)	No	No

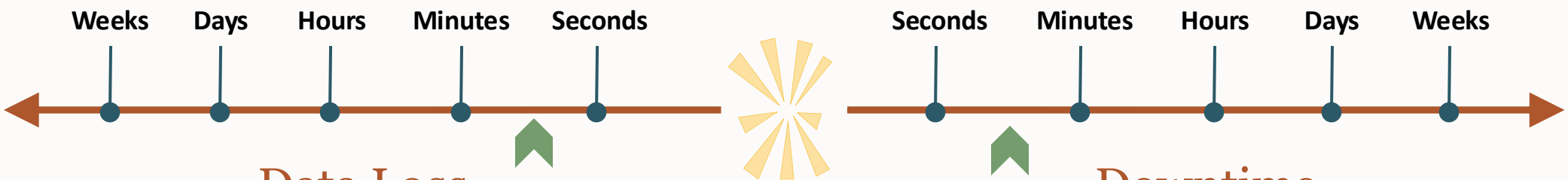
Factors Affecting Election Time:

- **Network Latency:** Higher network latency between data centers can increase the time required for leader election and data replication.
- **Cluster Size:** As the number of nodes increases, the overhead of managing elections and replicating logs grows, potentially impacting scalability and election time.

Understanding RPO and RTO

Disaster Recovery protects your RPO

High Availability protects your RTO



Data Loss

(Recovery Point Objective)

Downtime

(Recovery Time Objective)

Recovery Point Objective (RPO)

Tolerance for data loss (sec's, hours, days); determines frequency of backups and replication approaches

Recovery Time Objective (RTO)

The shorter the Recovery Time Objective (RTO) the quicker you get back to business

Possible data loss impact

Business availability / continuity impact

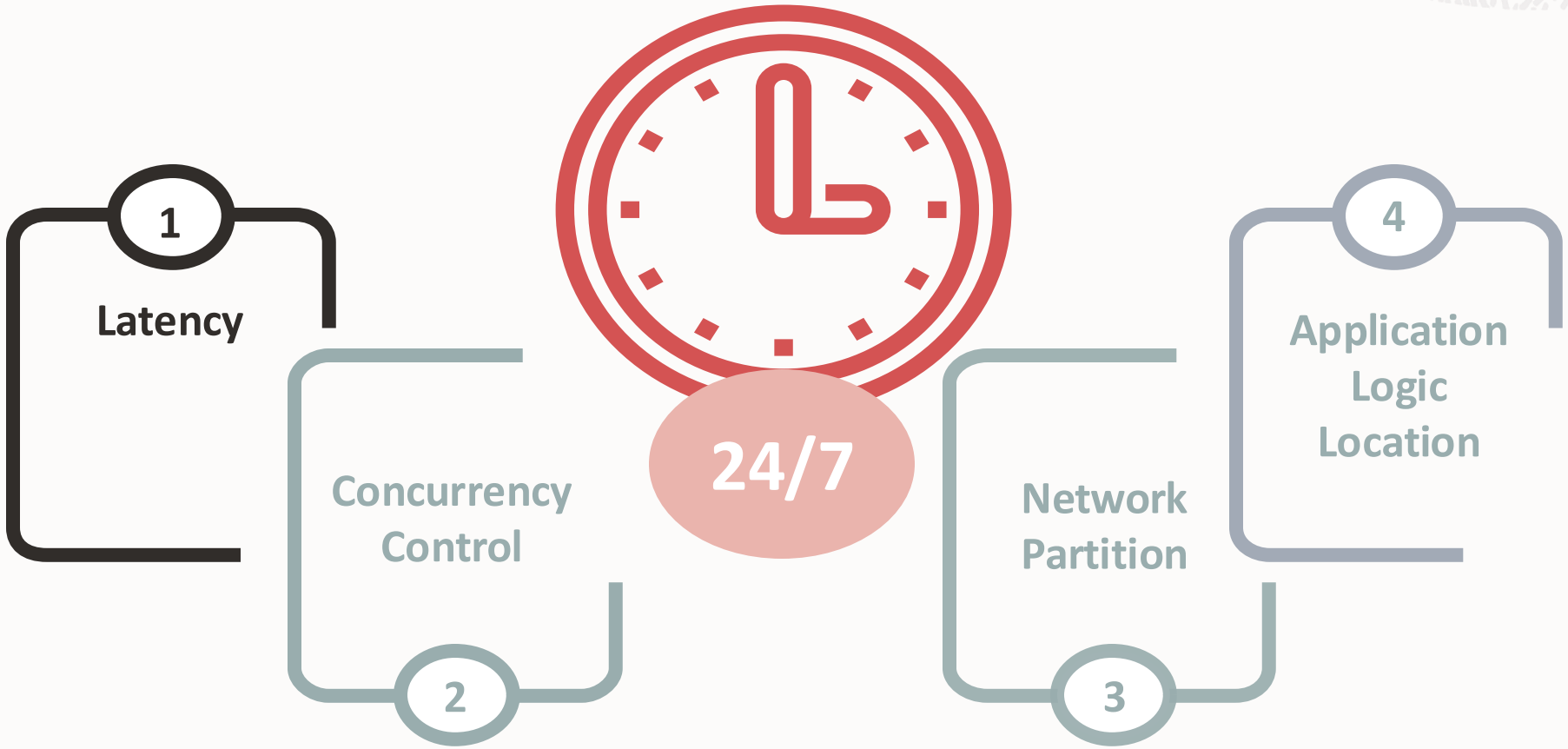
Some More SQL Distributed Databases Comparison

Capability	CockroachDB	YugabyteDB	TiDB
Read Replica Support	Yes (geo-partitioned)	Yes (follower reads)	Yes (TiFlash replicas)
Replica Consistency	Eventually consistent	Timeline Consistency	Strong Consistency Stale Read (Timestamp Consistency)
DR Read Usage	Recommended	Recommended	Recommended

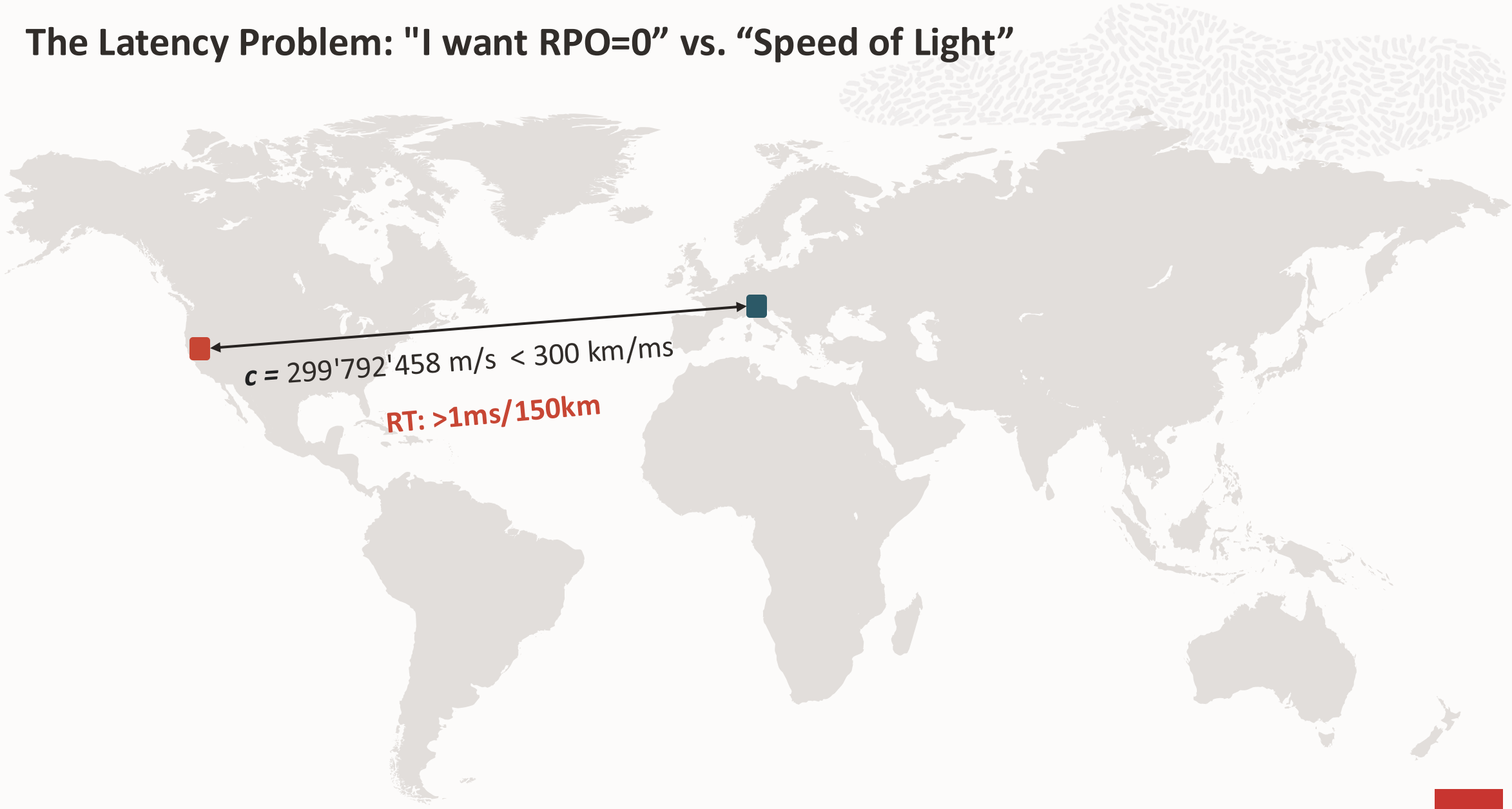
Factors Influencing Data Reliability(*)

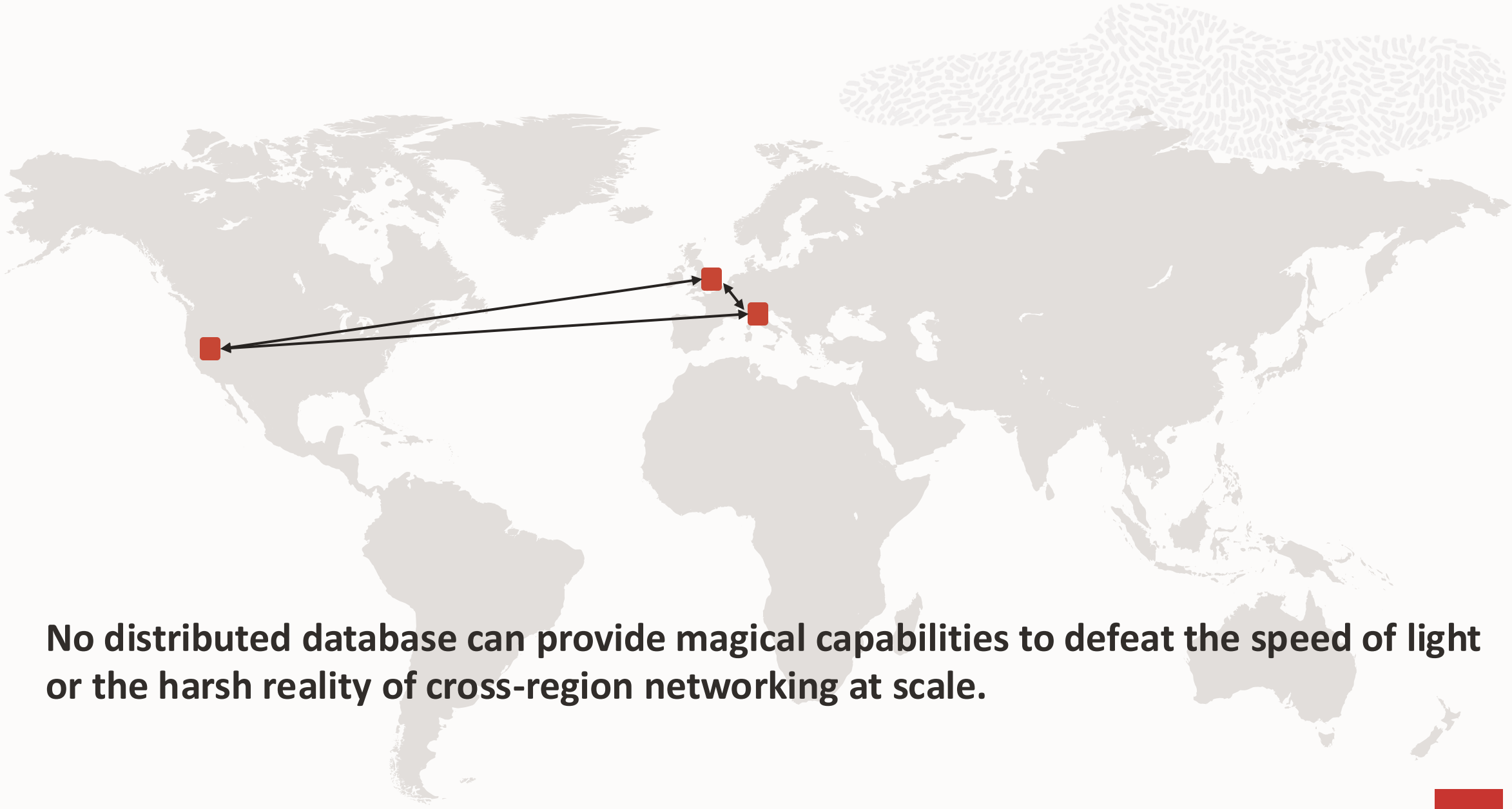
(*) https://en.wikipedia.org/wiki/Data_reliability

Top 4 Factors Influencing Reliability



The Latency Problem: "I want RPO=0" vs. "Speed of Light"





No distributed database can provide magical capabilities to defeat the speed of light or the harsh reality of cross-region networking at scale.

SQL Distributed Databases Cross-Region Integrated DR/HA

Designing RAFT groups across multiple regions is essential for resilience.

Challenge:

If quorum spans East + West, East outage = no quorum = no writes.

Recommendation:

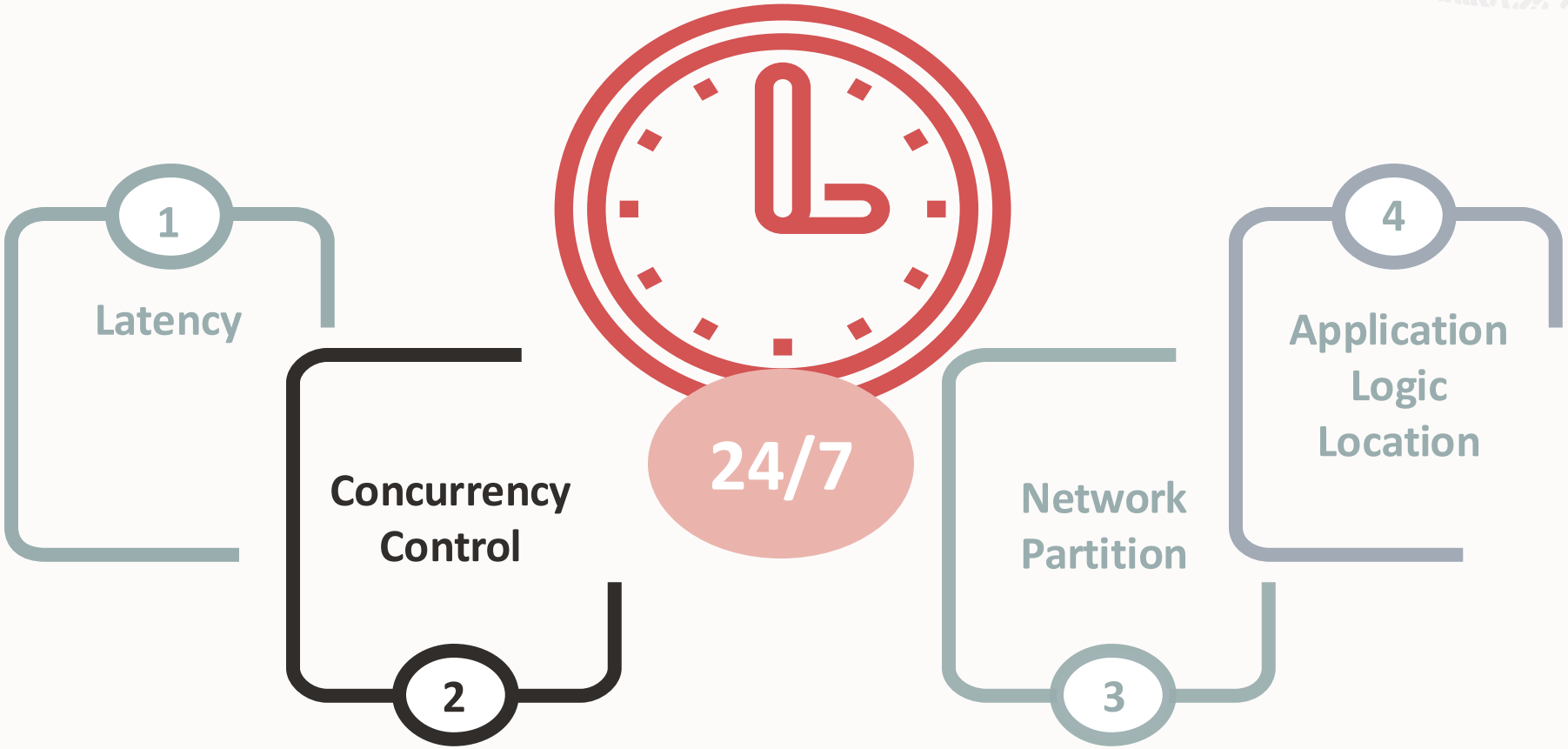
Split voting nodes across 3 zones and use observer/read-only replicas in DR sites.

Different Approaches – Different Results

CockroachDB ensures Disaster Recovery (DR) capabilities only within the same region. No multi-region implementations are possible without incurring severe penalties to the applications using it (as per example, latency).

Oracle Database can easily fulfil any cross-region Disaster Recovery (DR) requirements, including multi-active (since 2009) and zero data loss at any distance.

Top 4 Factors Influencing Reliability



Concurrency Control

- **Optimistic locking**, where a record is locked only when changes are committed to the database
- **Pessimistic locking**, where a record is locked while it is edited

https://docs.oracle.com/en/database/oracle/oracle-database/19/cncpt/data-concurrency-and-consistency.html#GUID-E8CBA9C5-58E3-460F-A82A-850E0152E95C

SQL Distributed Databases Data Consistency & Failover

During failover, transactions may roll back if not committed by a majority.

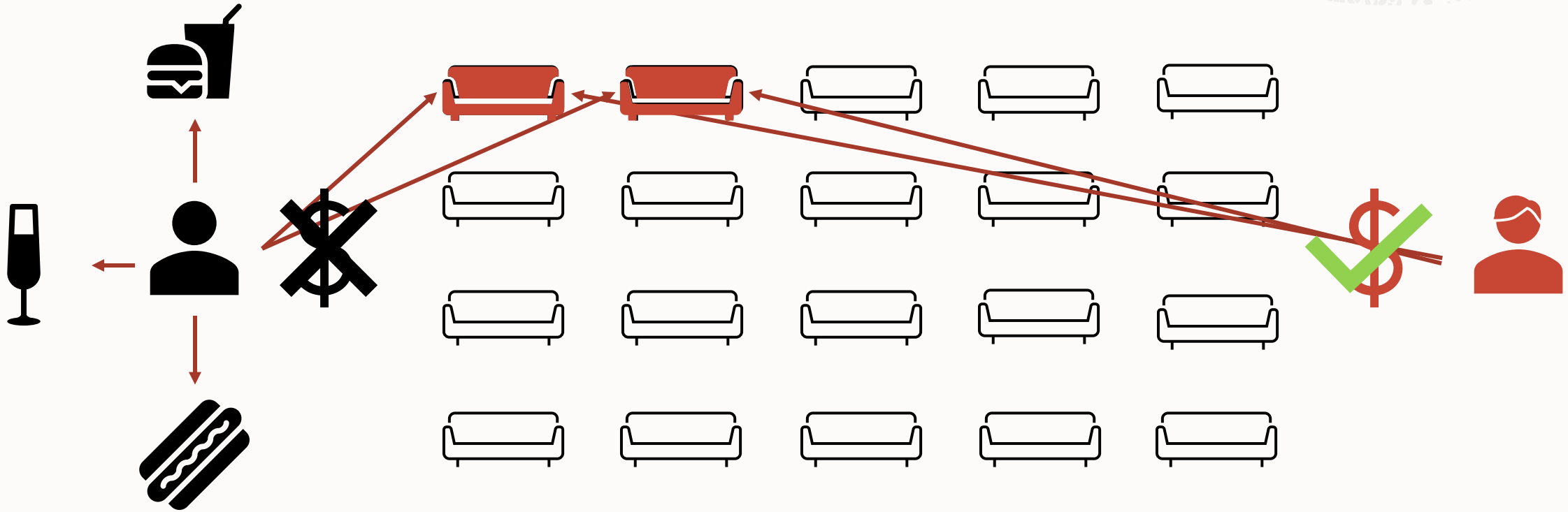
Feature	CockroachDB	YugabyteDB	TiDB
Write Durability Model	Majority write	Majority write	Raft commit based
Transaction Model (Default)	Serializable	Snapshot Isolation	Snapshot Isolation

SQL Distributed Databases Concurrency Control and Lost Writes

Most use MVCC and transactional layers on top of RAFT.

Feature	CockroachDB	YugabyteDB	TiDB
Concurrency Control	MVCC	MVCC + 2PC (two-phase commit)	MVCC + Per-region
Default Locking Mechanism	Optimistic (Default) – Pessimistic manually only (code)	Optimistic (Default with Snapshot Isolation level) – Pessimistic (Read Committed Isolation Level)	Pessimistic (Default) and Optimistic
Lost Write Handling	Strong prevention	Strong prevention	Depends on network

Example of a hypothetical system using Optimistic Concurrency Control



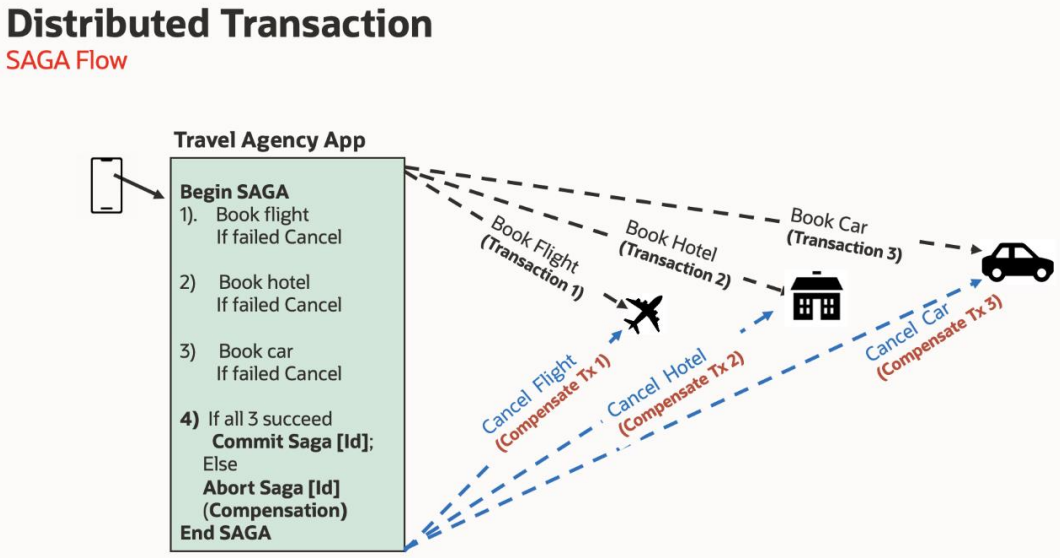
Different Approaches – Different Results

Using optimistic concurrency control by default (like CockroachDB does) means developers may be required to fully redesign it data model and application

The best of both worlds: Oracle automatically assures read consistency per your requirements (statement-level or transaction-level read consistency).

New with 23ai

- **Oracle Database SAGA Pattern** is a robust approach to maintaining data consistency across multiple microservices involved in long-running transactions. A saga is a sequence of local transactions that, collectively, achieve a global transaction. It uses either orchestration or choreography to manage the sequence and potential rollbacks

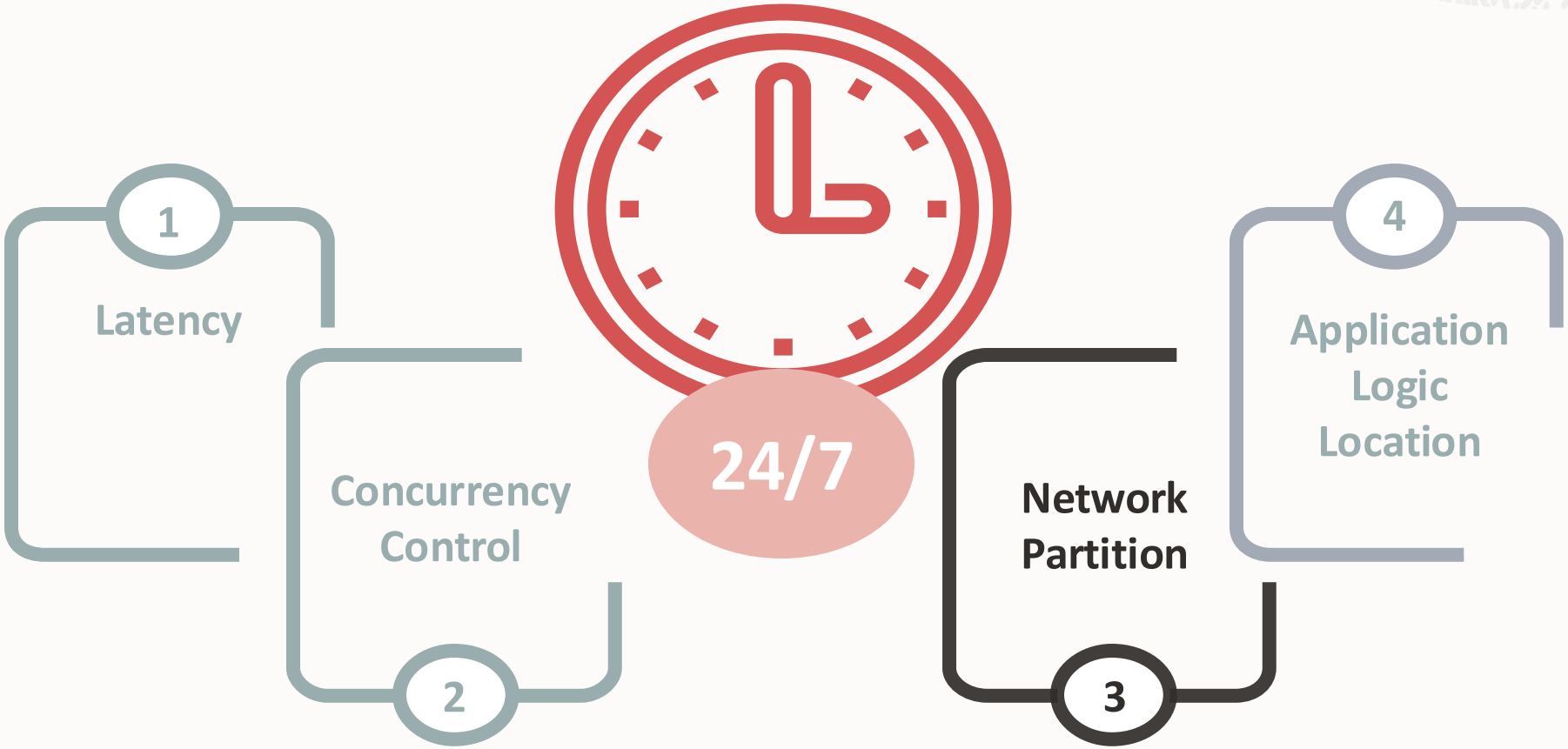


New with 23ai

- **Lock-Free Reservation** is a feature that significantly improves concurrency on numerical column updates, particularly for "hot" data such as account balances, inventory quantities, and similar numeric aggregate data. It allows multiple concurrent transactions to modify the same column value without being blocked by each other, which is especially useful for scenarios with high concurrency and updates to low cardinality tables.

- **Transaction Priority** is a feature that allows applications to assign priority levels or levels of importance (HIGH, MEDIUM, or LOW) to transactions. This enables the database to automatically roll back lower-priority transactions that are blocking higher-priority ones, potentially improving overall system concurrency and responsiveness.

Top 4 Factors Influencing Reliability



Network Partitioning

DID IT CRASH?

DID IT STALL?

DID IT KEEP COMMITTING?

WHAT IS THE LEADER DOING?

- If the leader is still committing
  - DATA LOSS and possible split-brain
- If the leader crashed
  - *Maybe* no data loss

Different Approaches – Different Results

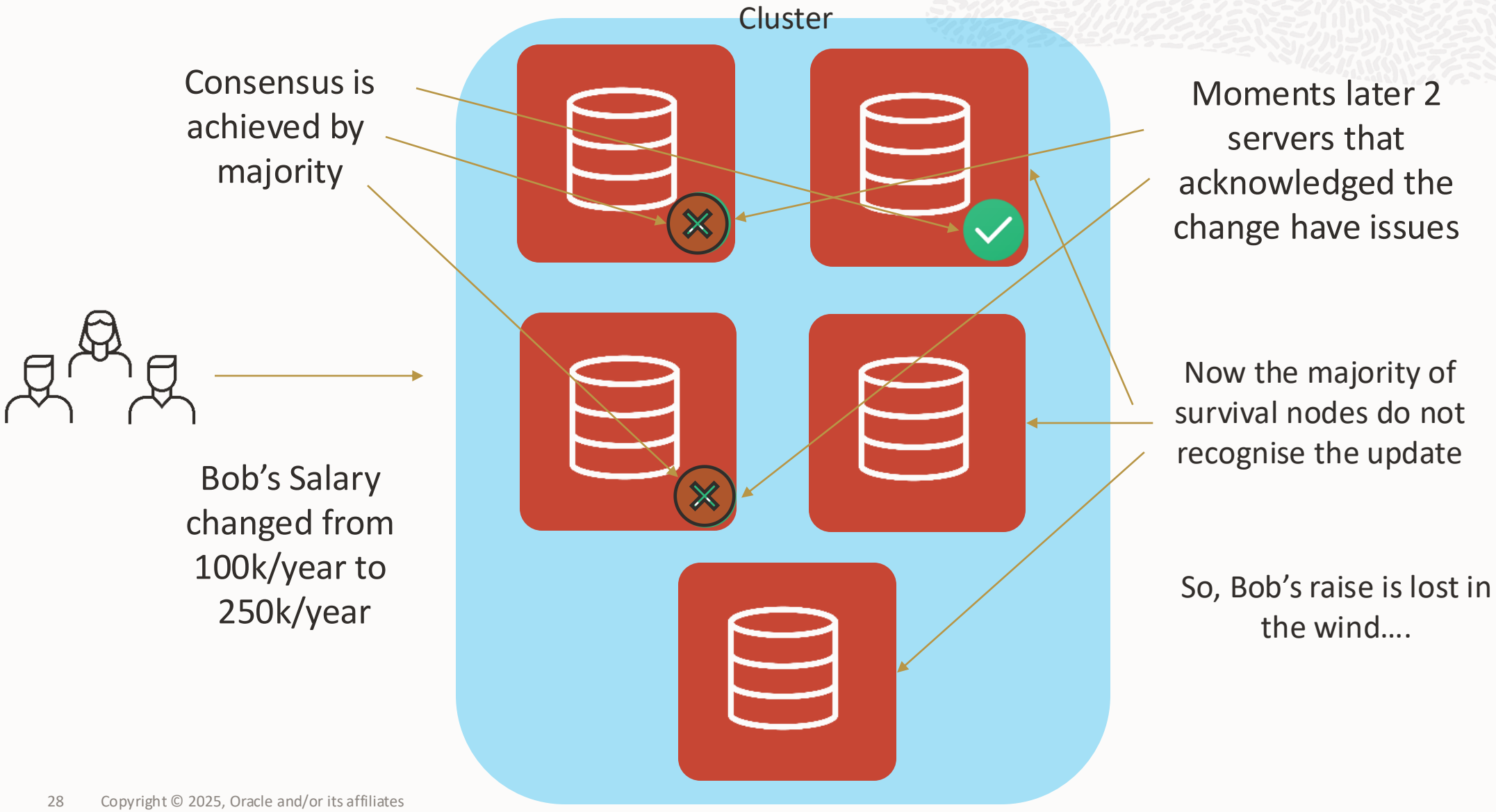
With CockroachDB a network issue and a change of leader could trigger a lost update scenario.

With Oracle Database, lost updates would not occur due to network partitioning.

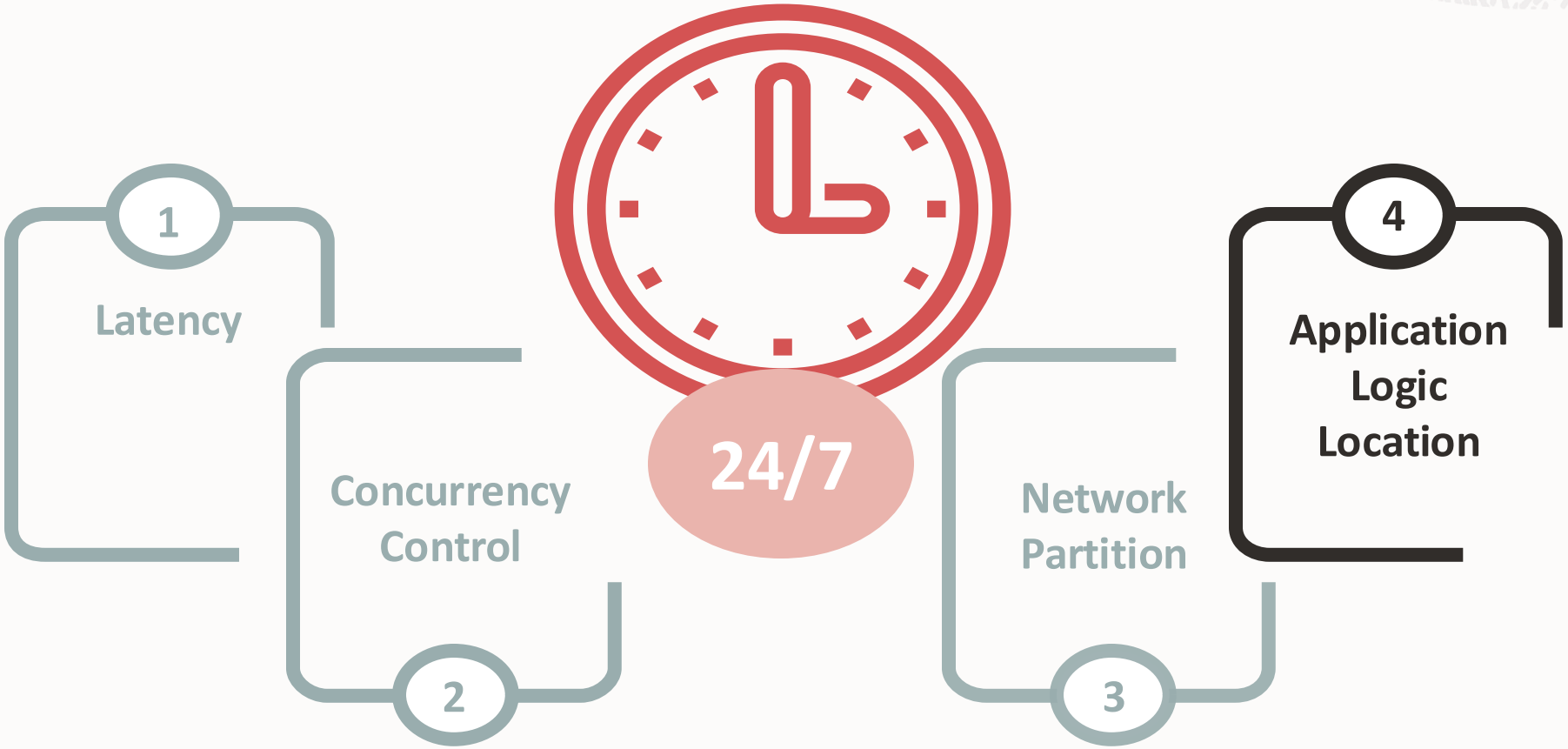
With Oracle Real Application Clusters (RAC), when used for high availability and scalability

Oracle Data Guard, when used in maximum availability mode or higher

The Hypothetical Journey of a lost update that could happen with Raft replication and optimistic concurrency control

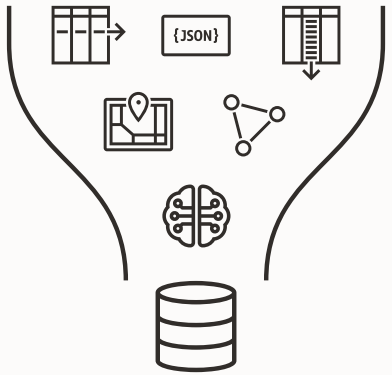


Top 4 Factors Influencing Reliability



Application Logic is best stored in Oracle Converged Database

A complete database that makes it dramatically easier to develop and run modern apps

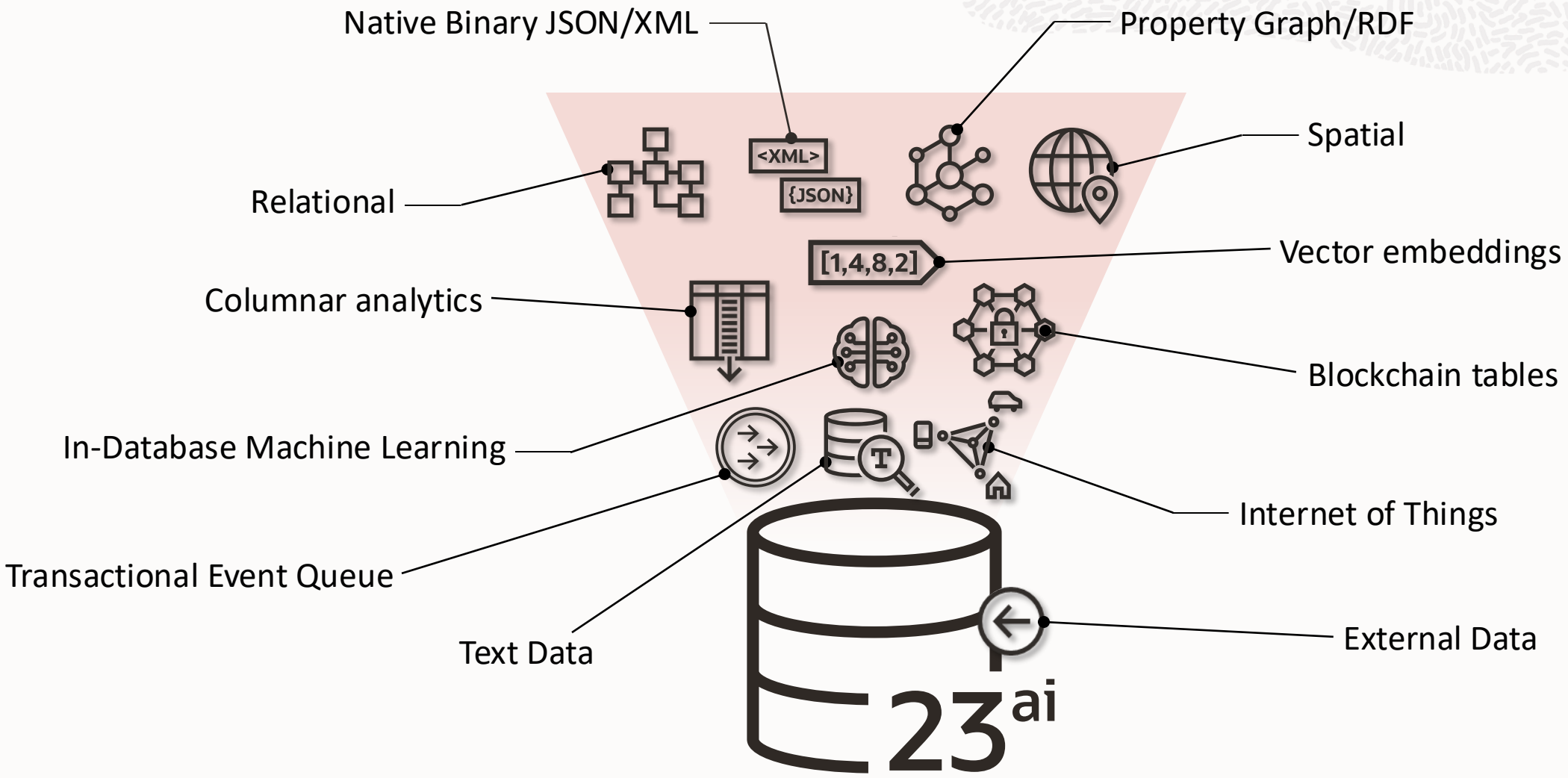


Converged Database

- **No need to fragment** data across databases to support new app requirements
- **Scaling and availability are transparent**, without sacrificing data consistency
- **No need to compromise** on functionality or performance
  - Oracle's data technologies are rated industry-leading in each area

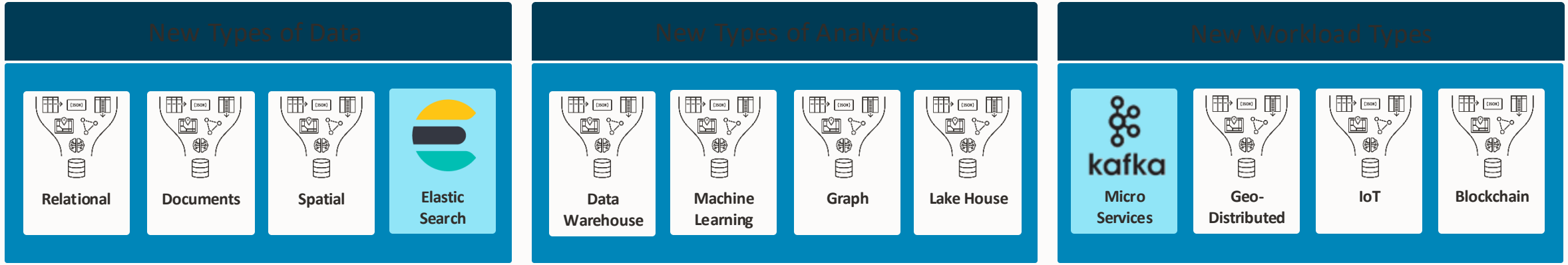
Creating a Fully Complete Database has Taken Decades of Effort by Thousands of Engineers

Oracle Converged Database

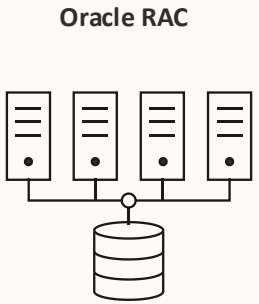


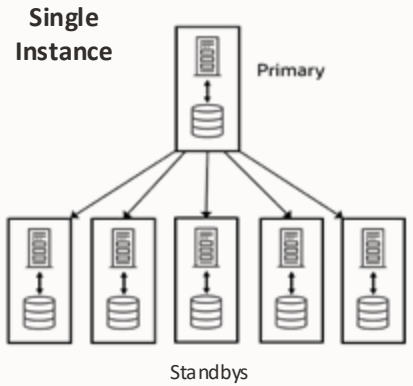
Oracle Converged Database provides choices

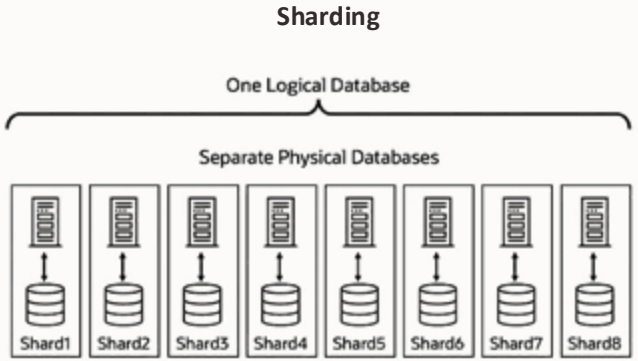
Converged does not mean data must be in one monolith database – **YOU choose**:



(Support for all modern data types and analytics are included at no additional charge.)







Distributed Databases - Comparison Matrix

	On-Premises Availability	Distribution Layer	Data Modeling	Converged Database	Triggers, Cursors and Stored Procedures within DB	Distributed Capabilities Since	CC (*) Mode (Default)	Popularity Ranking**
CockroachDB	Yes	Raft with Range Sharding (Hash-Sharded Indexes)	Limited Data and Partition Types	No	**No**	2015	Optimistic	**64**
YugabyteDB	Yes	Raft with Hash and Range Sharding	Limited Data and Partition Types	No	Yes	2016	Optimistic	**111**
Google Spanner	No	Raft with Range Sharding	Limited Data Types	No	**No**	2017	Pessimistic	**97**
Oracle Database	Yes	In-Memory Redo (Physical and Logical) Replication, Shard (*), **and Raft** **NEW IN 23ai**	Huge amount of Data and Partition Types	Yes	Yes	1986 with version 5	Pessimistic & Optimistic	**1**

Many more years of innovation than all other competitors together!

(*) Concurrence Control ** Reference: https://db-engines.com/en/ranking

(*) **User-Defined**, **Range-Based**, **List-Based**, and **Composite Sharding**.

Oracle Database with MAA

Impact of downtime

$350K
average cost of downtime per hour

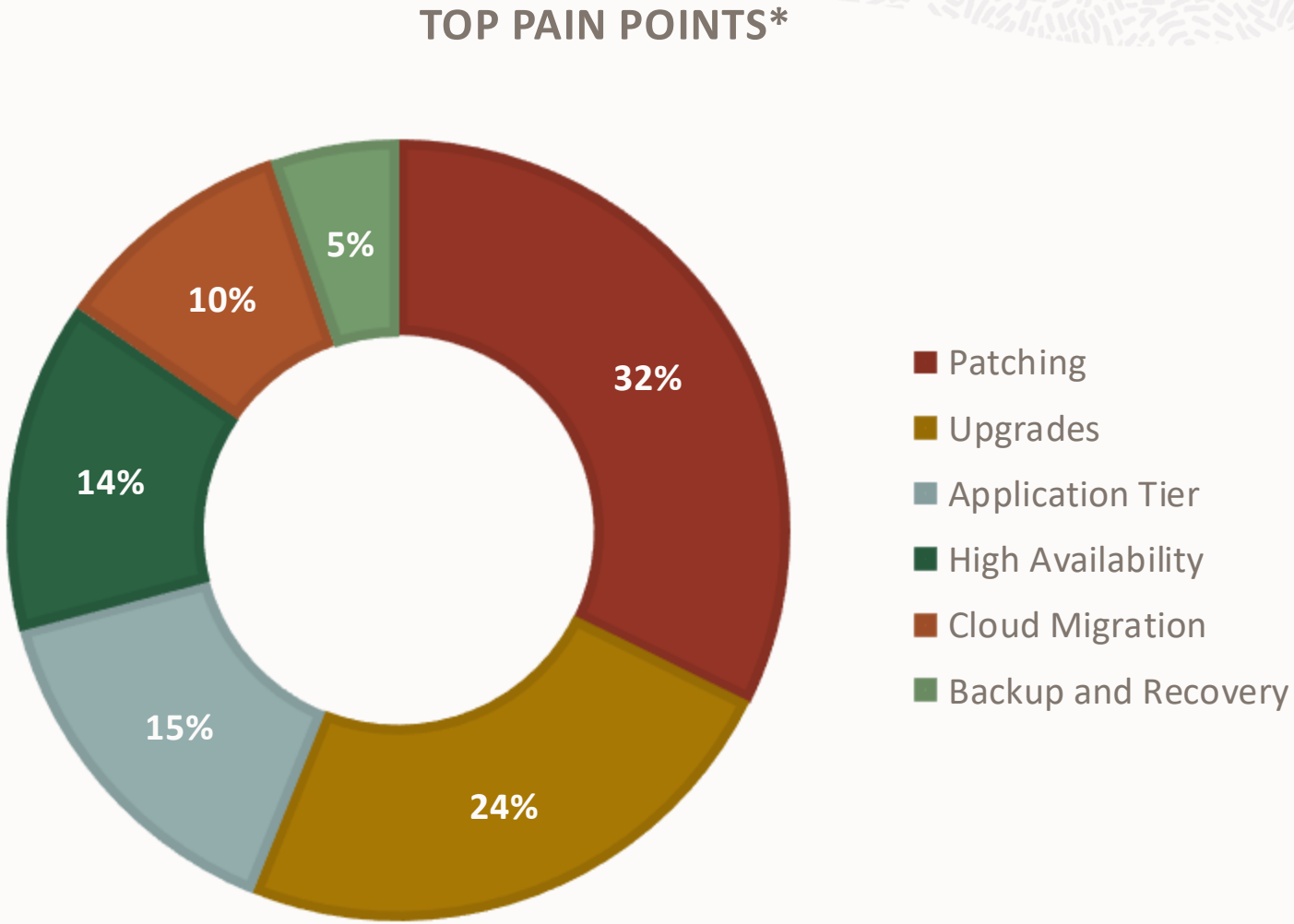
87 hours
average amount of downtime per year

$10M
average cost of unplanned data center outage or disaster

91%
percentage of companies that have experienced an unplanned data center outage in the last 24 months

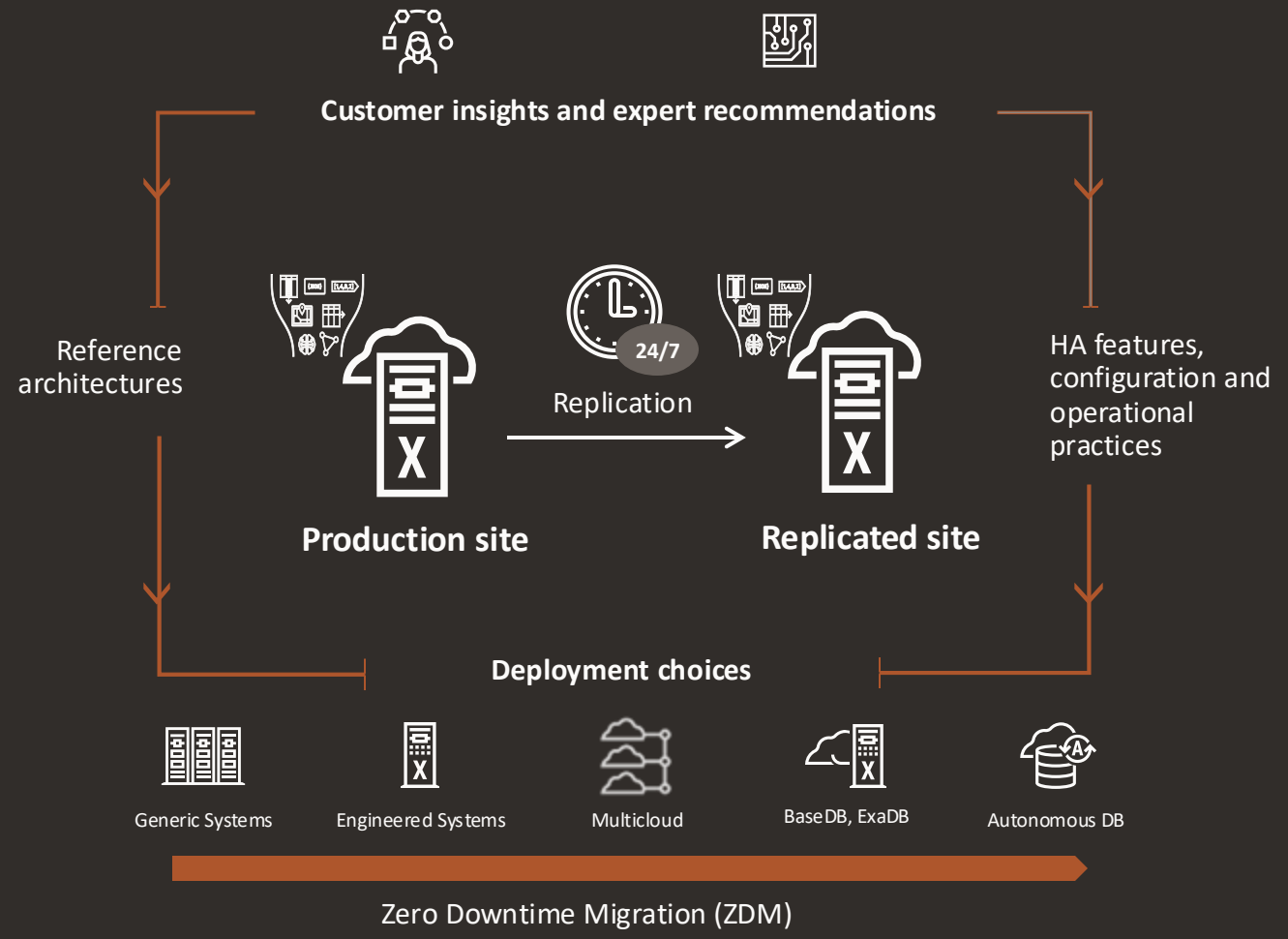
Addressing Day to Day Challenges…

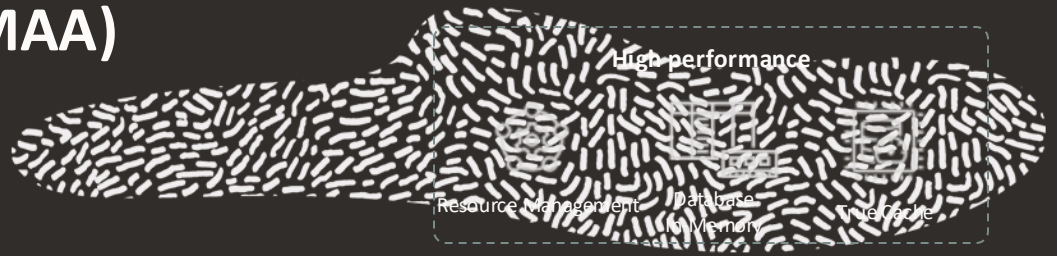
- Minimize downtime and complexity during patching, migrations & upgrades
- Eliminate data loss and downtime during unexpected outages
- Application tier connectivity and role transitions
- Be prepared while avoiding complexity



*Based on MAA Summit survey results

Next Gen Maximum Availability Architecture (MAA)





Continuous availability

Application Continuity

Online Redefinition

Edition-based Redefinition

Data protection

Flashback

RMAN

ZDLRA+ ZRCV

Active replication

Active Data Guard

Full Stack DR

GoldenGate

Scale out & Lifecycle

RAC

Globally Distributed Database

FPP

Real Application Testing

Next-Gen MAA Reference Architectures

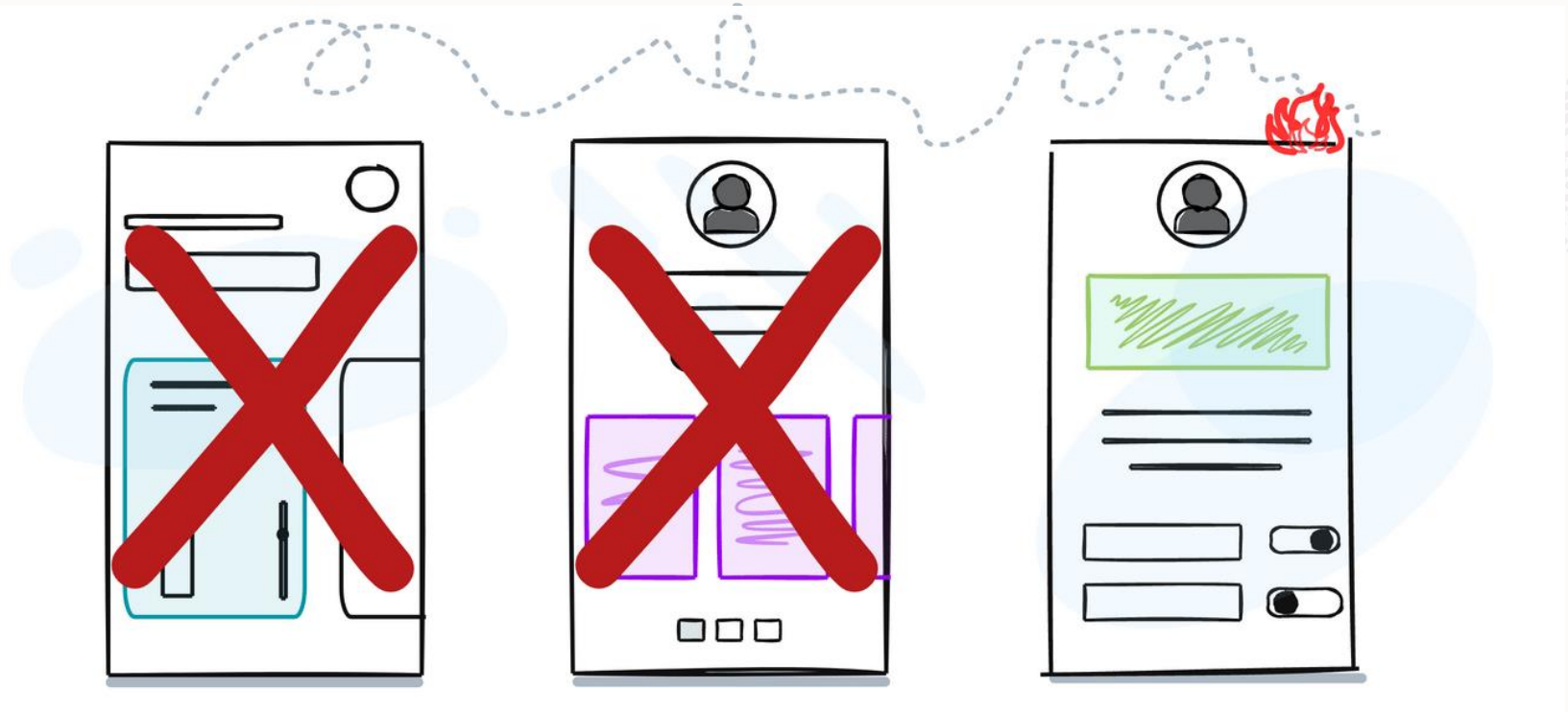
Availability service levels for the next generation of Oracle Database

Bronze	Silver	Gold	Platinum	Diamond (NEW)
Dev, test, prod	**Prod/departmental**	**Business critical**	**Mission critical**	**Extreme availability**
Single instance DB Restartable Backup/restore	**Bronze +** Database HA with RAC or Local Data Guard Client failover HA best practices Application Continuity (optional)	**Silver with RAC +** DB replication with (Active) Data Guard with automatic failover Client failover DR best practices	**Gold with** Exadata *and either option:* Option 1: GoldenGate with Oracle Database 19c ***OR*** Option 2: (Active) Data Guard with Oracle AI Database 26ai	**Configuration** GoldenGate 26ai replicas, *each running*: Oracle AI Database 26ai + RAC on Exadata + (Active) Data Guard
Recoverable local failure: Minutes to hour Disasters: Hours to days RPO < 15 min	Recoverable local failure: seconds to minutes Disasters: Hours to days RPO < 15 min	Recoverable local failure: Less than 60 seconds Disasters: < 5 min RPO = zero or near zero	Recoverable local failure: Less than 20 seconds Disasters: < 30 secs RPO = zero or near zero	Recoverable local failures: Less than 10 seconds Disasters zero to 10 secs RPO = zero or near zero

Oracle Database with MAA vs CockroachDB at a Glance

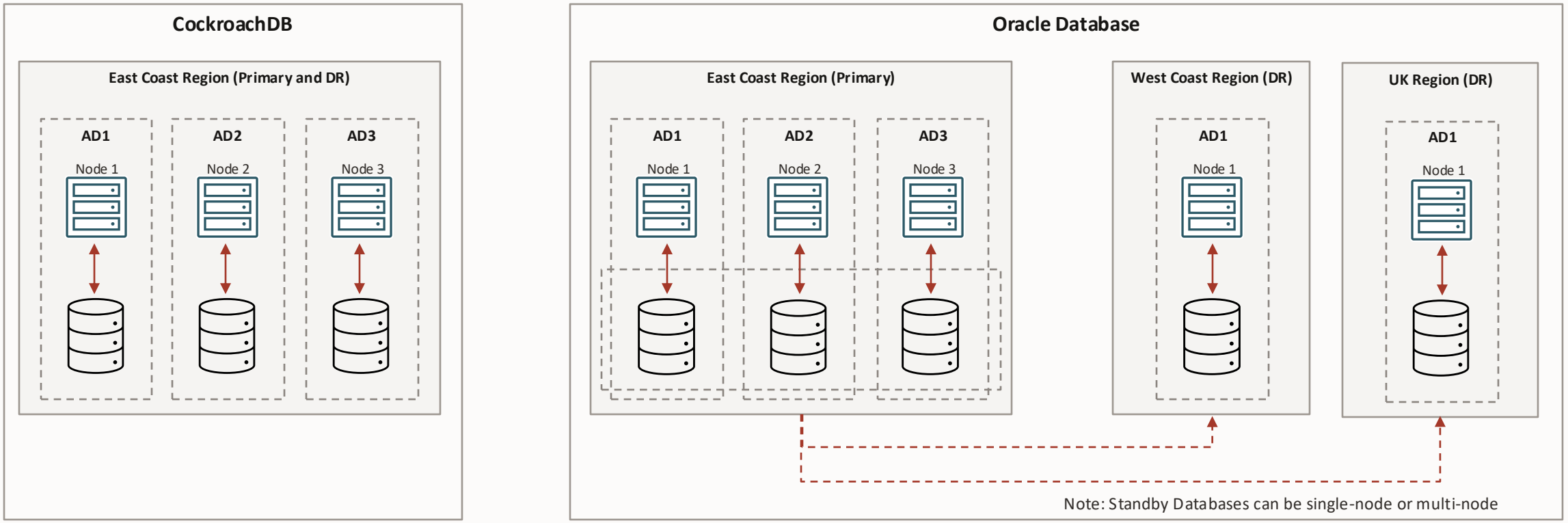
Capabilities	Oracle Gold & Platinum MAA	CockroachDB
Full Disaster Recovery (regional protection)	✓	⚠
High Consistency (within a region)	✓	✓
One Node Survival	✓	✗
Time Machine Recovery	✓	✗
Full Backup and Recovery	✓	✓
Full Backup and Recovery (with Zero Data Loss)	✓	✗
Near Zero Downtime Upgrade/Patching	✓	✓
Automatic Conflict Detection and Resolution	✓	✗
Ransomware Protection at Database Level	✓	✗
Automated Failover and Transaction Replay	✓	✗
End-to-End Validation – Data Corruption Prevention	✓	✗
Zero Data Loss at any Distance	✓	✗
Application Versioning	✓	✗
Converged Database	✓	✗
Data sovereignty	✓	✓
Easy to Install/Configure	✗	✓

✓ Fully Supported ✗ Not Supported ⚠ Warning - Performance/Scalability Implications

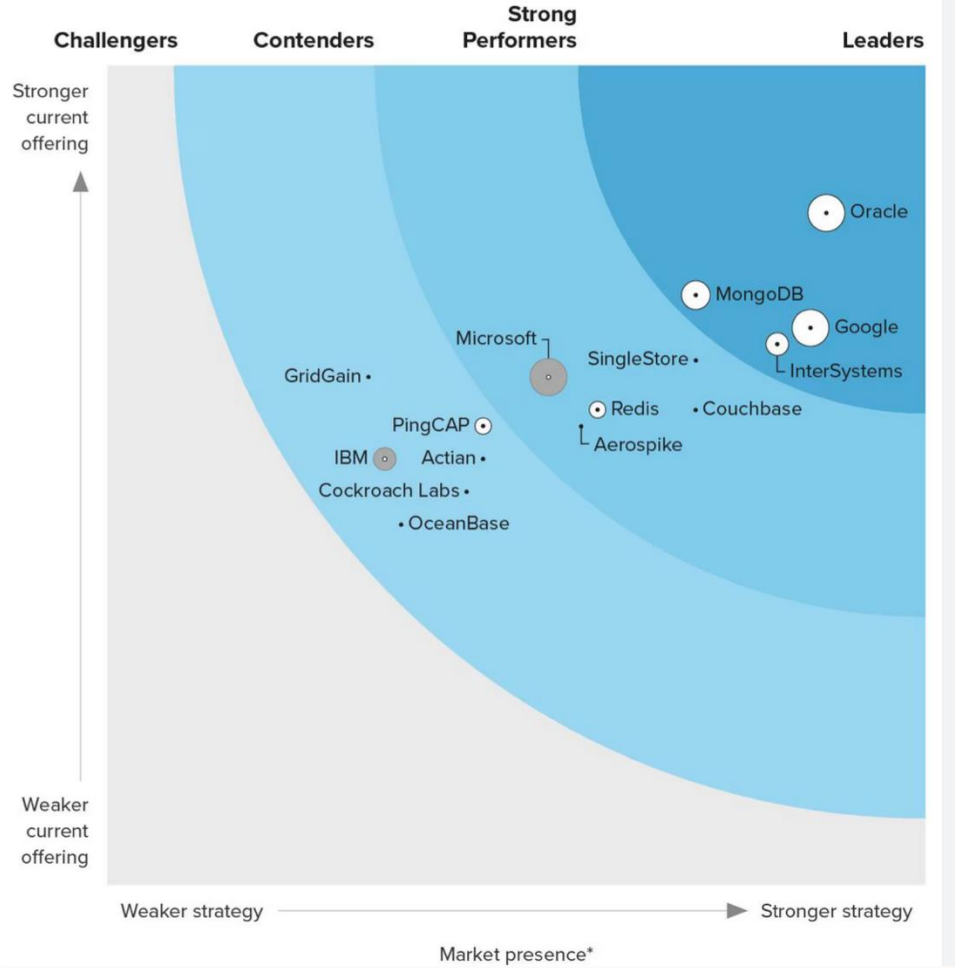


A distributed database is a useful technology for achieving an acceptable level of high availability but is *not* built with disaster recovery in mind.

Disaster Recovery (DR) Scenarios



Industry Leading



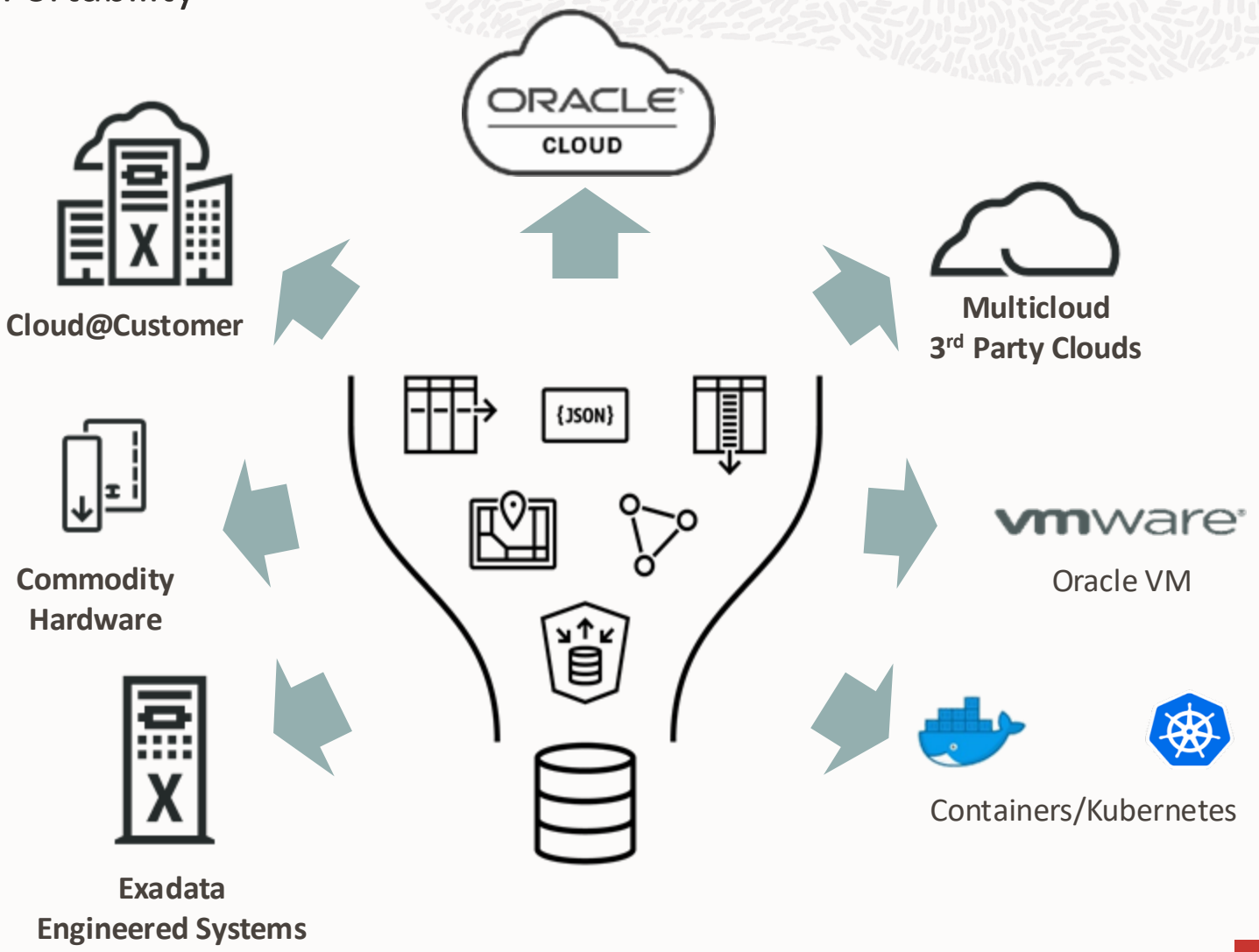
Leader in Translytical Database.

Forrester rates Oracle the strongest leader in the Forrester Wave: Translytical Data Platforms Q4 2024. The rating in this Forrester evaluation validates Oracle Database for its ability to support converged OLTP and Analytical (Translytical) workloads using Database In-Memory, Exadata, extensive multi-model capabilities, and support for relational and non-relational, structured and unstructured.

Oracle Database – Deployment Choices

Deploy Oracle Anywhere – Extreme Portability

Same database, same skills
Fast and efficient deployments

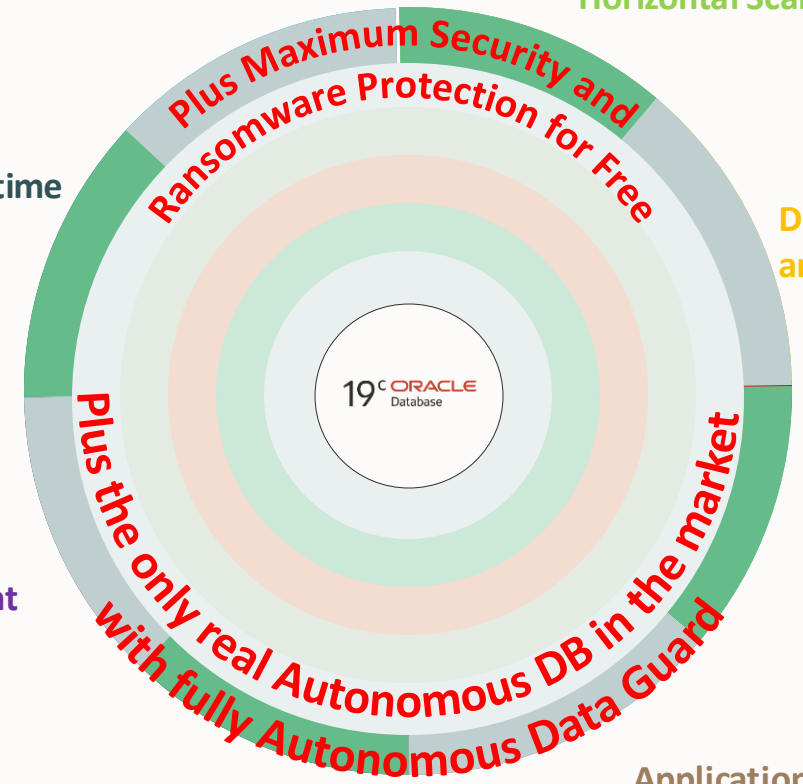


Conclusion

When choosing a technology for distributed database deployments it is important to balance the architecture requirements with the business needs and service level objectives

Summary and Conclusion

Oracle Database is all-inclusive, converged by design, and protected by MAA



Automatic Block Media Recovery
RMAN – Block Media Recovery
Zero Data Loss Recovery Appliance (ZDLRA)

Data Corruption Protection

High Availability & Horizontal Scalability

Oracle Real Application Clusters
Oracle Sharding
Oracle GoldenGate

Oracle Real Application Clusters
Oracle GoldenGate
Edition Base Redefinition
Rolling Upgrade
Oracle Fleet Patching and Provisioning (FPP)
Zero Downtime Migration (ZDM)

Zero Downtime Patches/ Upgrades/ Migrations

Disaster and Recovery

Oracle Data Guard
Oracle Active Data Guard
Oracle GoldenGate
Oracle Flashback
Zero Data Loss Recovery Appliance (ZDLRA)
Oracle Database Zero Data Loss Autonomous Recovery Service

Oracle Automatic Storage Management

Storage Management

Multi-Master Capabilities

Oracle Real Application Clusters
Oracle GoldenGate
Oracle Active Data Guard (DML Redirection)
Oracle Sharding with Raft

NEW IN 23c

Full Stack DR
Fast-Start Failover (FSFO)
Oracle Fleet Patching and Provisioning (FPP)
Global Data Services

Automation

Application Continuity

Oracle Real Application Clusters
Global Data Services
Oracle Application Continuity
Transparent Application Continuity
Edition Base Redefinition
Oracle GoldenGate

Top 5 reasons you should always take into consideration when thinking to replace Oracle with a distributed database

1. Changes to the application and data model layer are required to handle requirements that were naturally managed by the database.
2. You may need multiple technologies to fulfill all business/application requirements (adding complexity, risk, and cost).
3. A “low-cost solution” comes with a cost, including a lack of crucial functionalities and being charged for Enterprise Support and backup, not free.
4. The “try again later, scenario” - A lost transaction equals lost revenue and, maybe, brand impact!
5. Finally, possible data loss due to inefficient Disaster Recovery capabilities. Ensure all your requirements can be fulfilled, especially when discussing RTO and RPO requirements.

Want more details?

- The Fundamentals of Reliability in the Context of Distributed Databases - Part 1
- From Chaos to Order: The Importance of Concurrency Control within the Database - Part 2
- The Need for Speed: Physics, Latency, Emergent Technologies, and the Future of Enterprise Database Transactions - Part 3
- Demystifying the Safety Net: Is Your Database Prepared for Anything? – Part 4

Any Questions?

Thank you!


@fcomunoz


http://www.linkedin.com/in/franciscomunozalvarez


www.oraclemaa.com