

MySQL HA Solutions

Selecting the best approach to protect access to your data

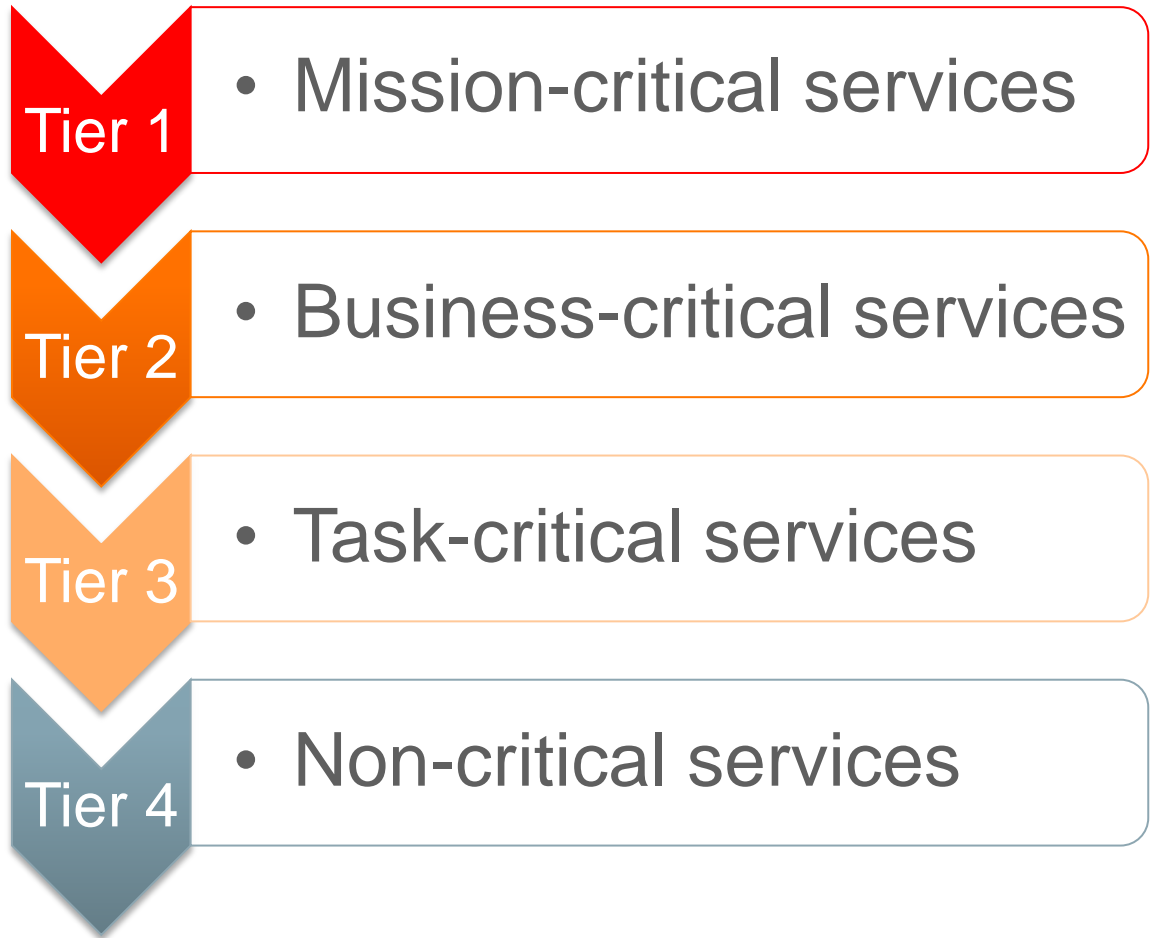
Sastry Vedantam
sastry.vedantam@oracle.com

Safe Harbor Statement

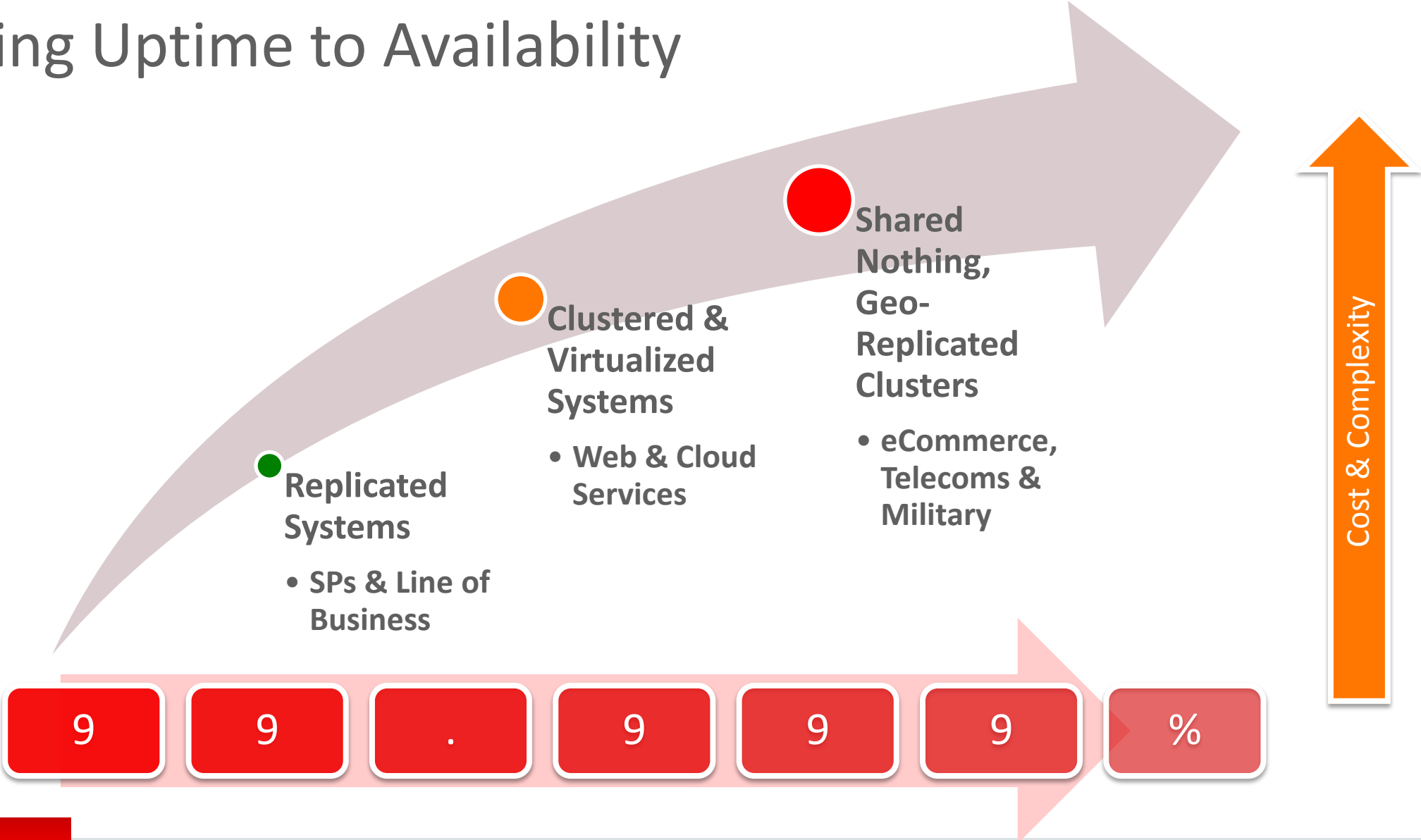
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Don't assume 99.999% HA needed for all apps

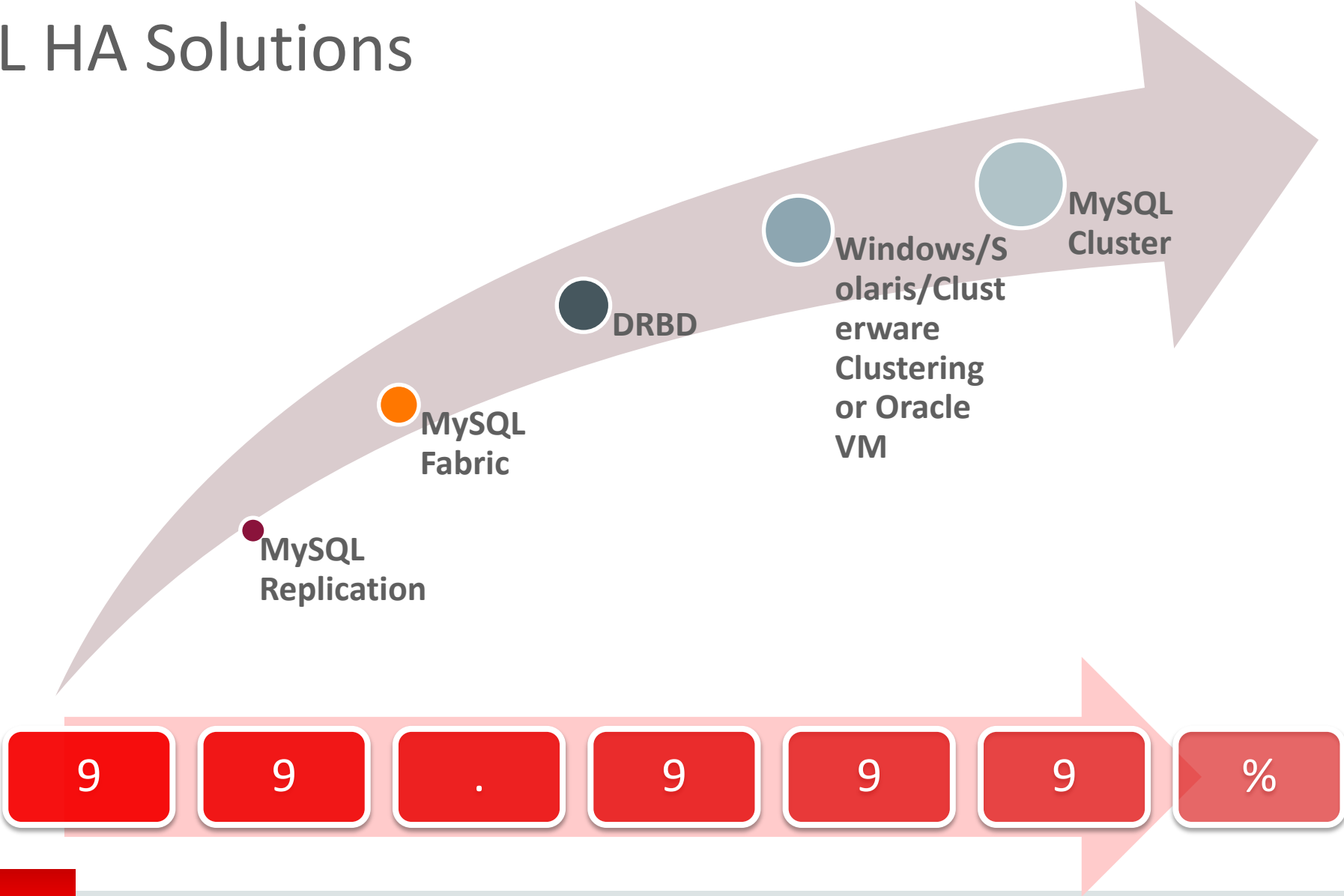
- Recovery Time Objective
 - Maximum length of downtime before there is break in “business continuity”
- Recovery Point Objective
 - Point in time to which data must be recovered when service is re-established



Mapping Uptime to Availability



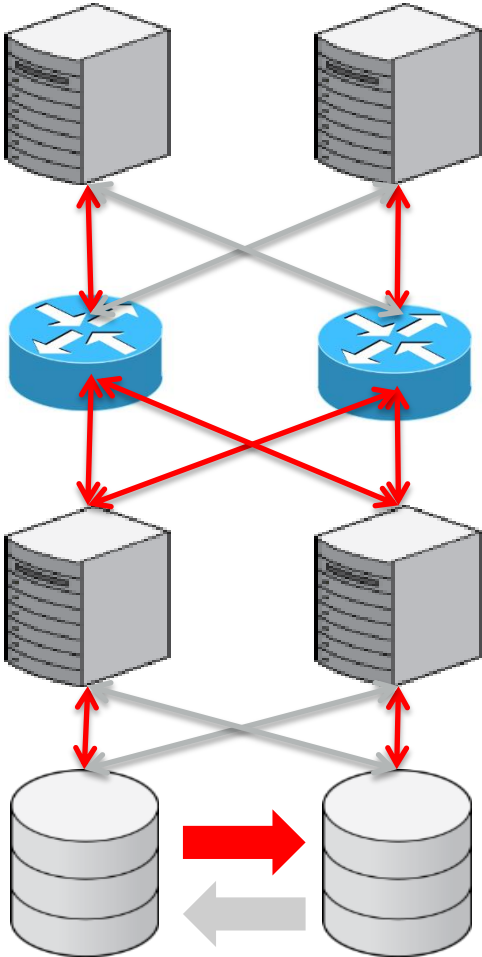
MySQL HA Solutions



Layers of HA

It's not just reliably storing the data

Redundant Access to Data



Redundant App Servers

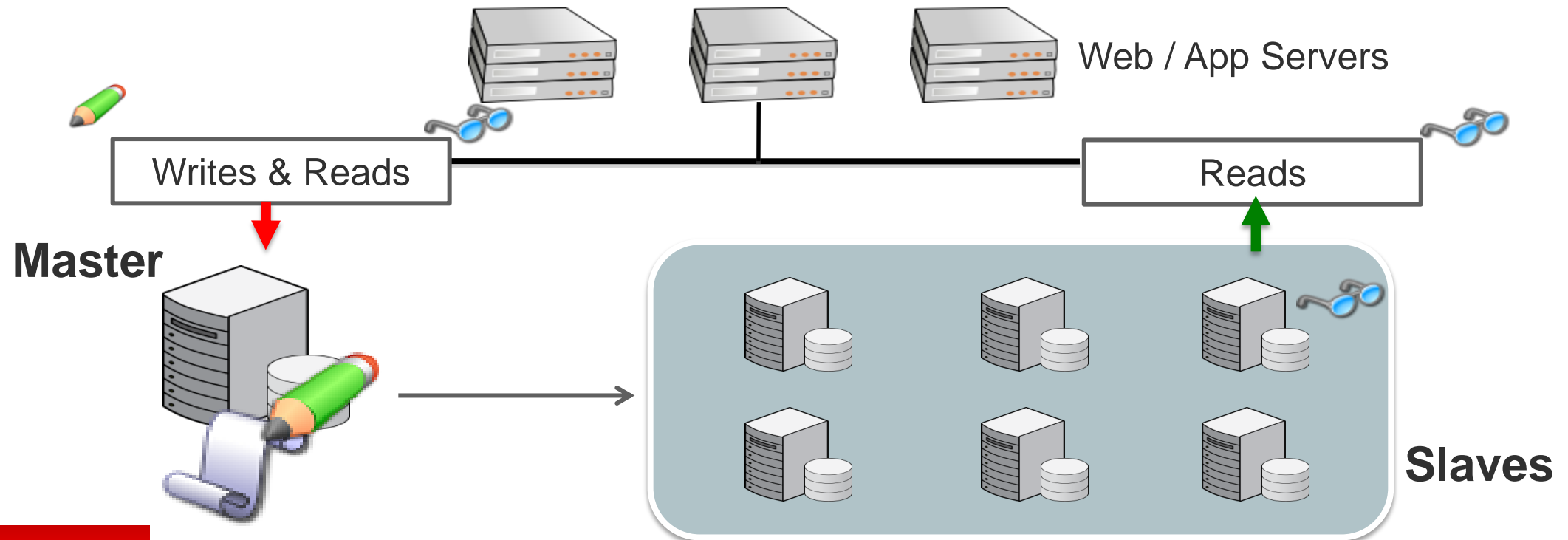
Routing to the Data

Data Redundancy

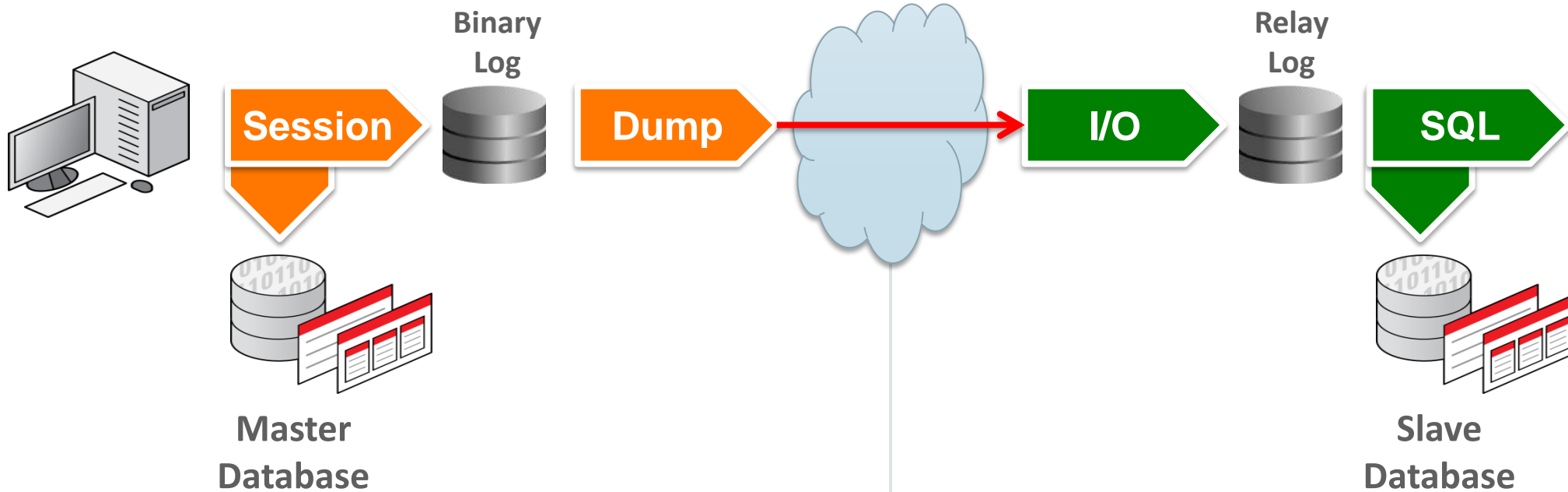


Why Replicate?

- Duplicates database from a “master” to a “slave”
 - Redundant copies of the data provide foundation for High Availability
 - Scale out by distributing queries across the replication farm



MySQL Replication Workflow



- Session thread: processes queries from the application – writes data to master database & associated events to binary log
- Dump thread: reads events from binary log and sends them to a slave

- I/O thread: receives replication events and stores them in slave's relay log
- SQL thread: reads replication events from slave's relay log and applies them to slave database

Asynchronous vs. Synchronous Replication

- Asynchronous
 - MySQL Default
 - In **parallel**: Master acks to app and sends transaction to slave
 - Fast
 - Risk of lost changes if master dies
- Semi-Synchronous
 - MySQL 5.5+ - Enhanced in MySQL 5.7
 - **Serially**: Master waits for change to be received by slave then In **parallel** ack to app and apply changes on slave
 - Intermediate latency
 - Lossless (MySQL 5.7)
- Synchronous
 - Only available with MySQL Cluster
 - **Serially**: Master waits for change to be applied on all slaves before ack to app
 - Higher latency
 - If Active/Active, best suited to small transactions
 - Lossless

MySQL 5.6 Replication

Performance and Scalability

- 5x Higher slave throughput with MTS (multiple schemas)
- Reduced master impact – Binlog Group Commit

Simplicity

- Global Transaction IDs

Automation

- MySQL Replication Utilities (including auto-failover)

Robustness

- Crash safety
- Replication checksums

Insurance

- Time Delayed Replication

MySQL 5.7 Replication – **Pre-GA**

Performance and Scalability

- Intra-schema Multi-Threaded Slave
- Faster Master

Monitoring

- Performance Schema

Operational Simplicity

- Dynamic slave filters
- On-line CHANGE MASTER

Resilience

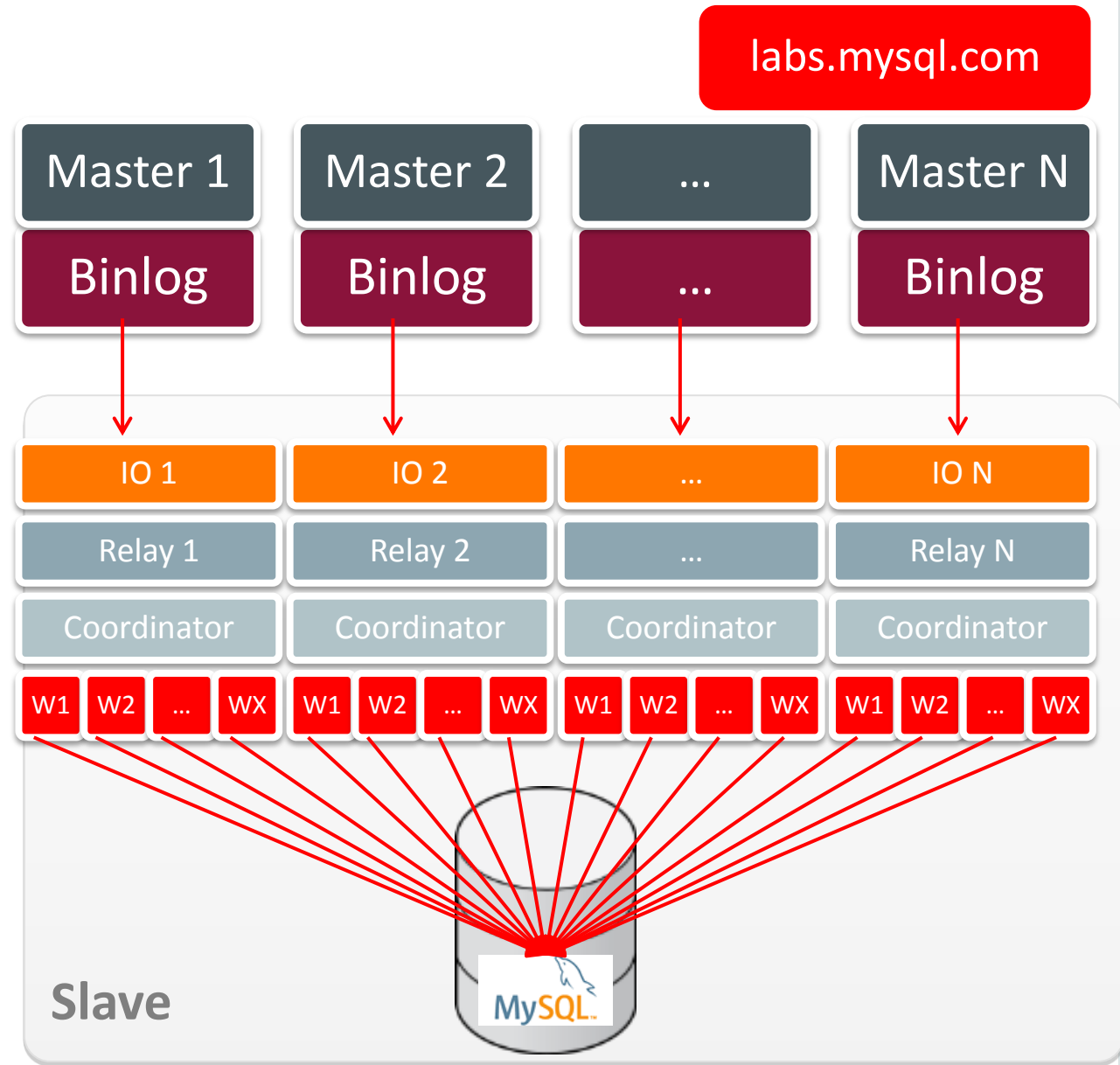
- Lossless Semi-Synchronous Replication
- Multiple Semi-Synchronous Acks

MORE Resilience

- GTIDs stored in transactional tables
- Automatic slave retries

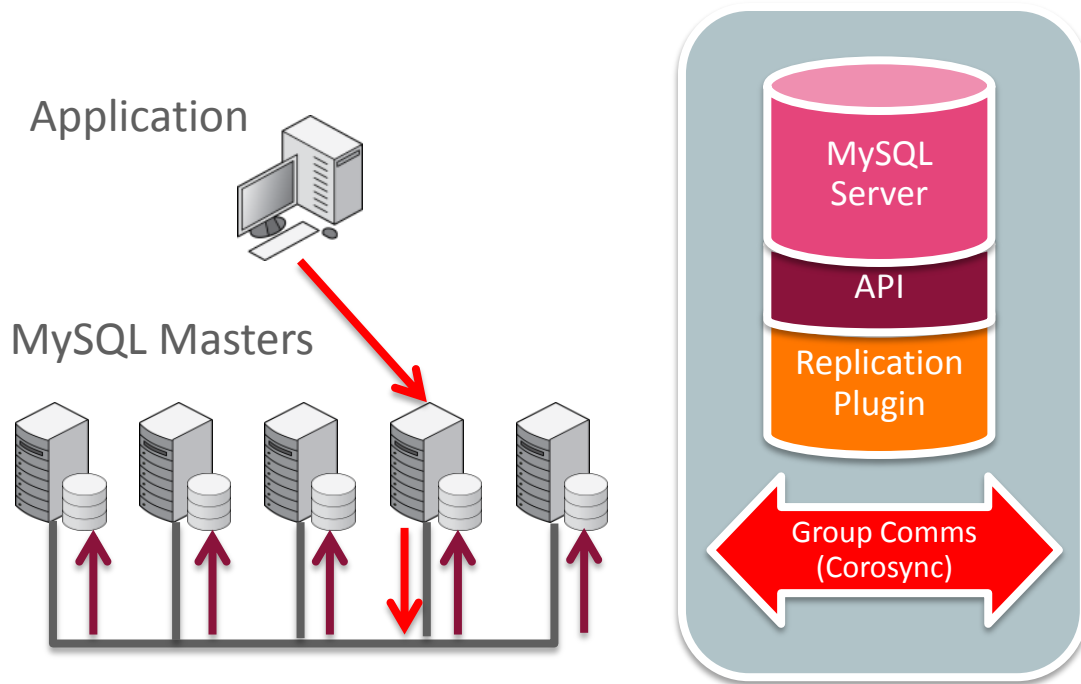
Multi-Source Replication

- Consolidate updates from multiple Masters into one Slave
 - Consolidated view of all shards
 - More flexible topologies
 - Centralized point for backups
- Compatible with Semi-Sync Replication & enhanced MTS
- Master-specific slave filters



MySQL Group Replication

labs.mysql.com



- Shared-nothing virtually synchronous database system
- Multi-master update anywhere
 - Conflict detection and resolution (transaction rollback)
 - Optimistic State Machine Replication
- Automatic group membership management and failure detection
 - No need for server fail-over
 - Elastic scale out/in
 - No single point of failure
 - Automatic reconfiguration
- Well integrated
 - InnoDB
 - GTID-based replication
 - PERFORMANCE_SCHEMA

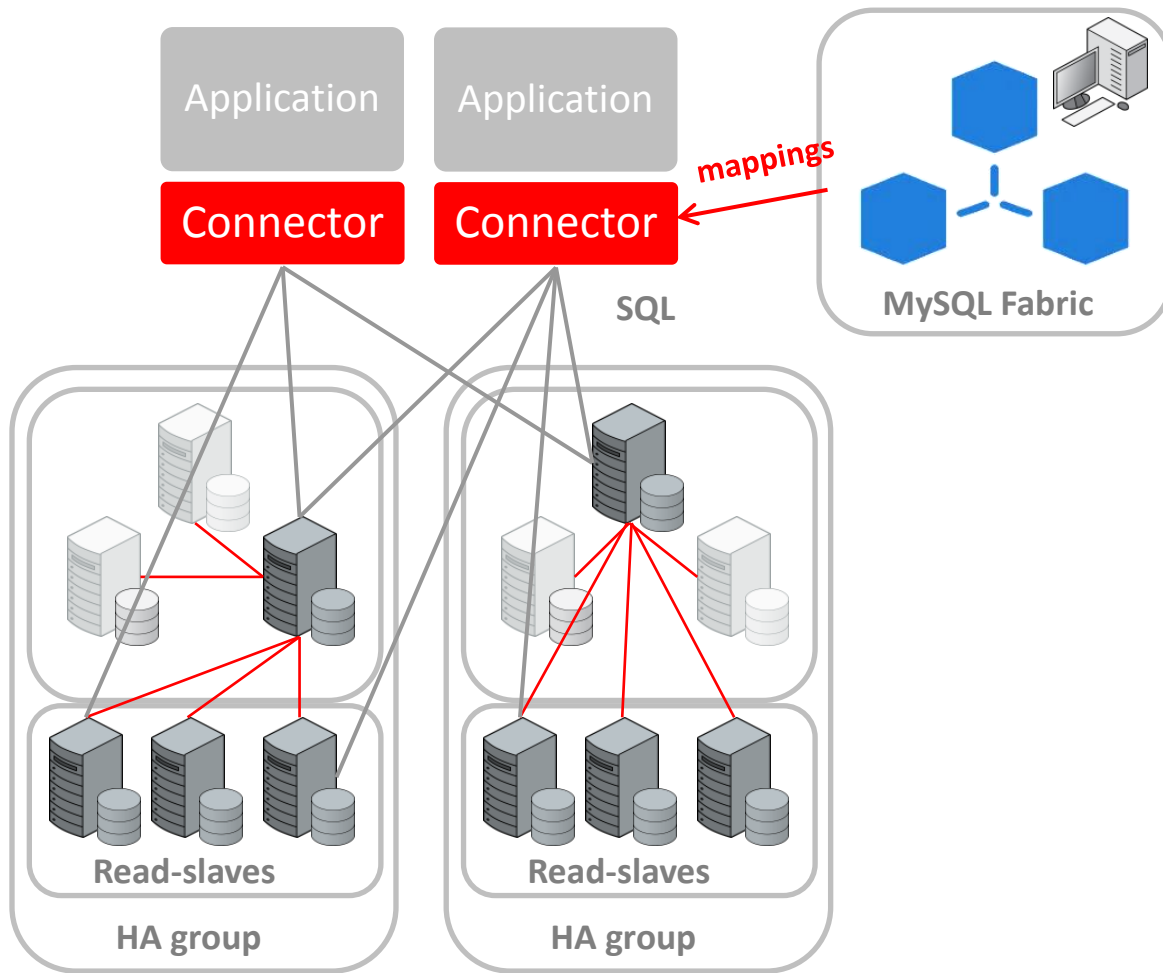
MySQL Fabric

An extensible and easy-to-use framework for managing a farm of MySQL server supporting high-availability and sharding



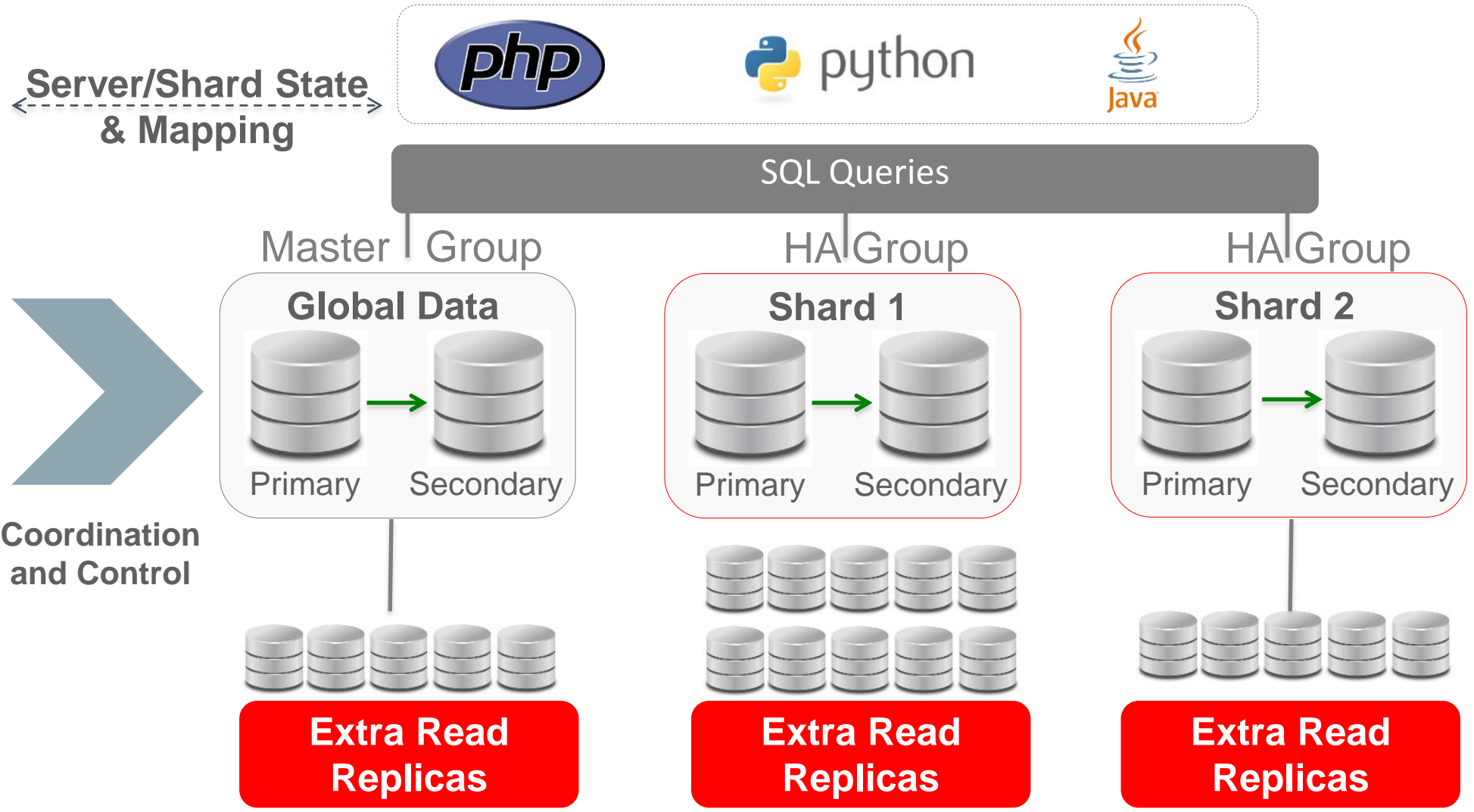
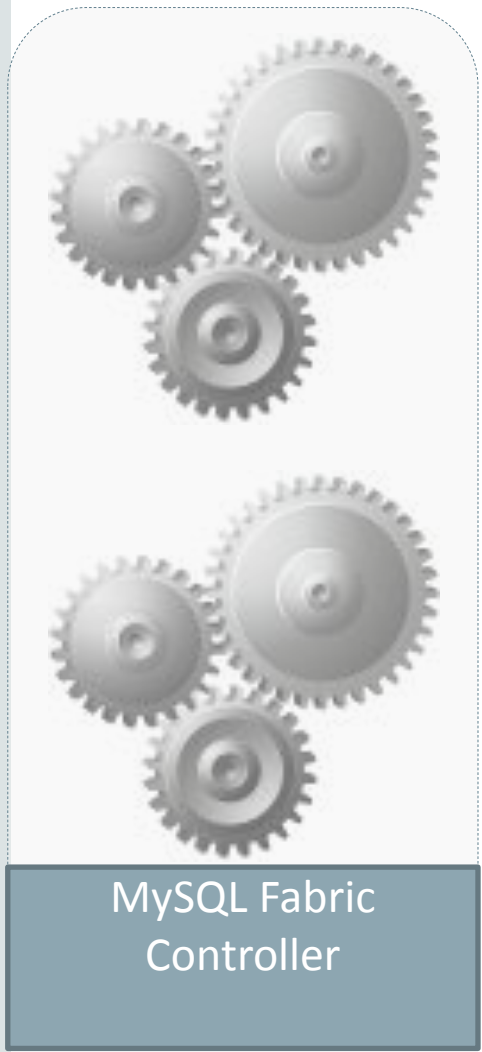
MySQL Fabric 1.5

High Availability + Sharding-Based Scale-out



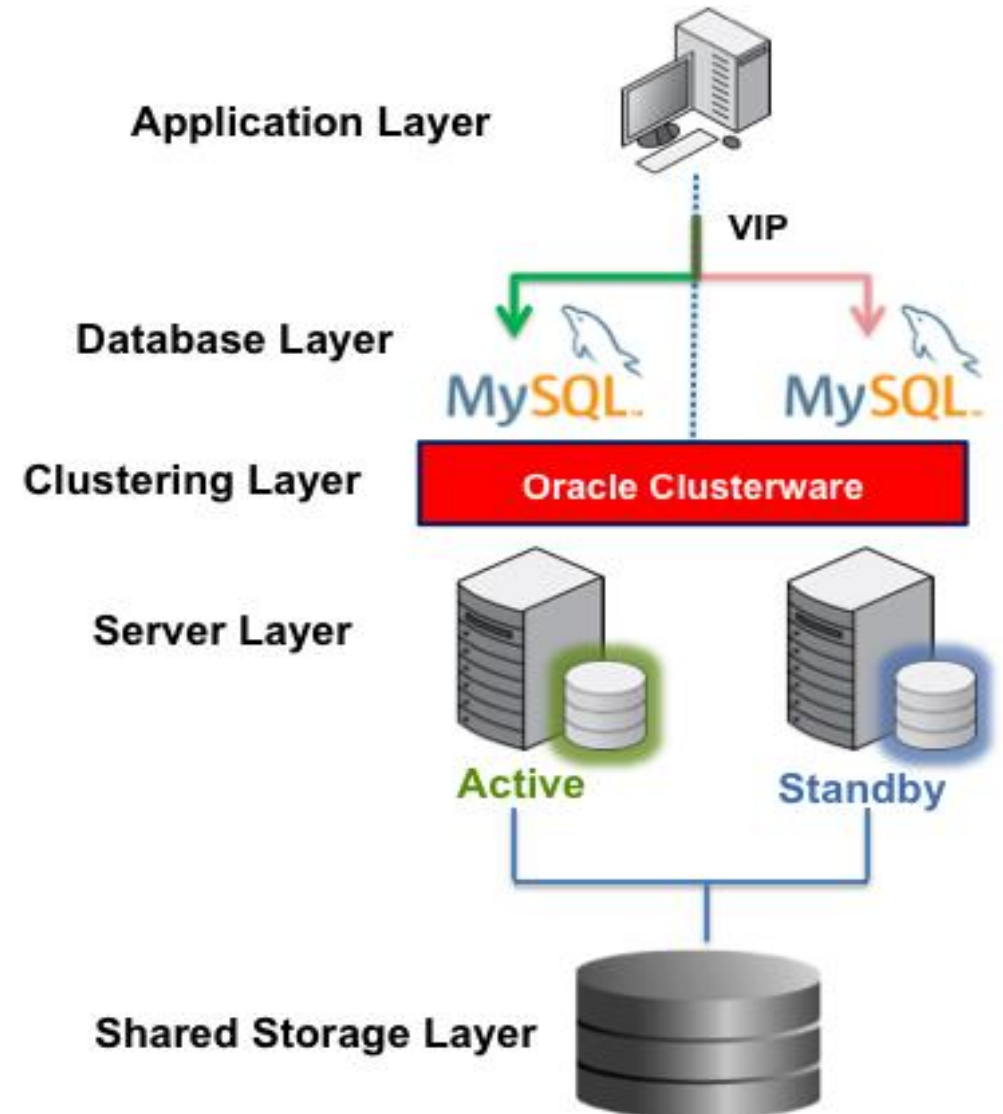
- High Availability
 - Server monitoring with auto-promotion and transparent application failover
- Optionally scale-out through sharding
 - Application provides shard key
 - Range or Hash
 - Tools for resharding
 - Global updates & tables
- Fabric-aware connectors rather than proxy: Python, Java, PHP, .NET, C (labs)
 - Lower latency, bottleneck-free
- Server provisioning using OpenStack etc.

MySQL Fabric Framework



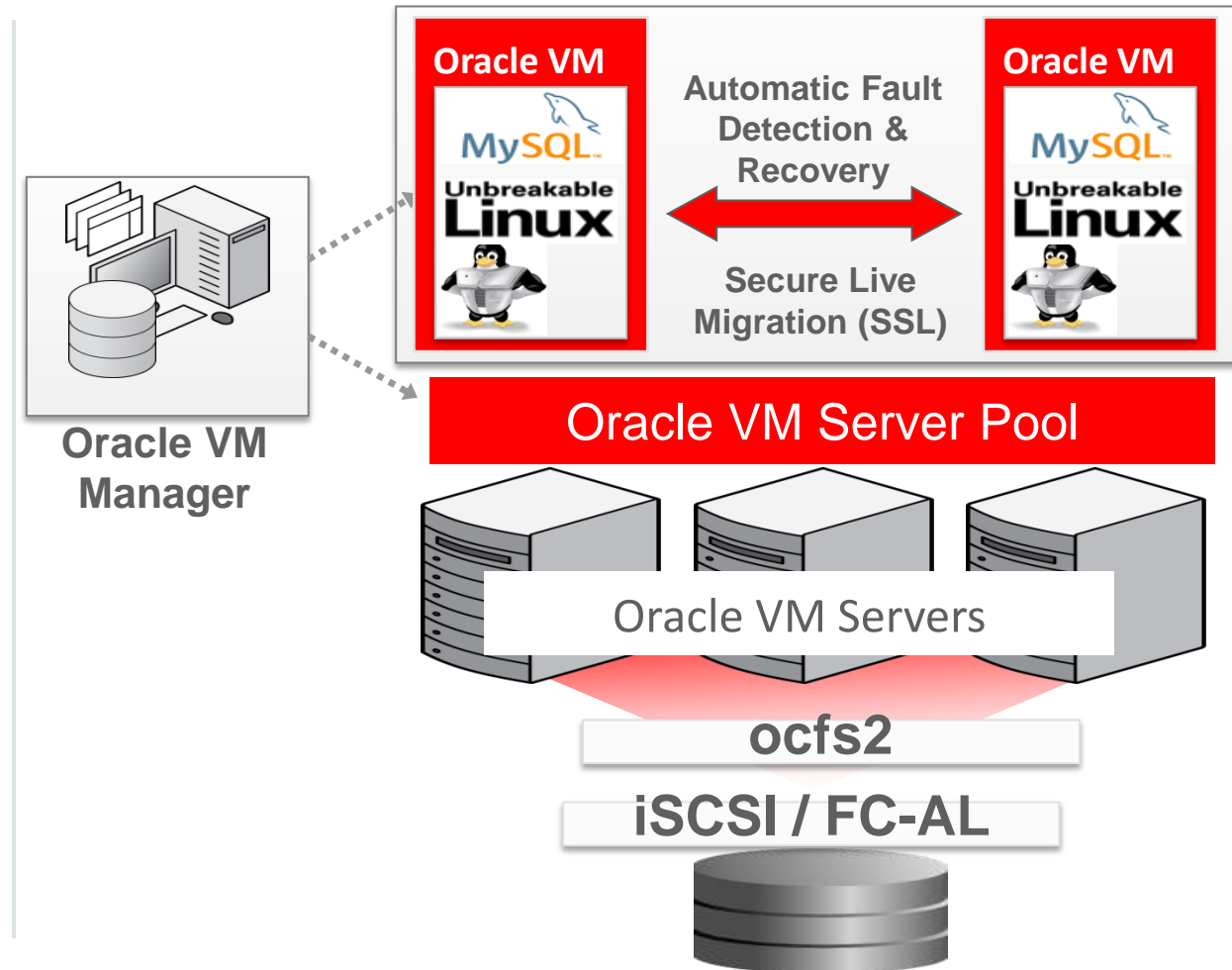
MySQL on Oracle Clusterware

- Oracle Clusterware unifies servers in a server farm to form a cluster
 - At the core of Oracle RAC
- Oracle Cluster 12c includes MySQL Server 5.6 agent
- Planned migration and failover of MySQL database
 - Hidden from the application



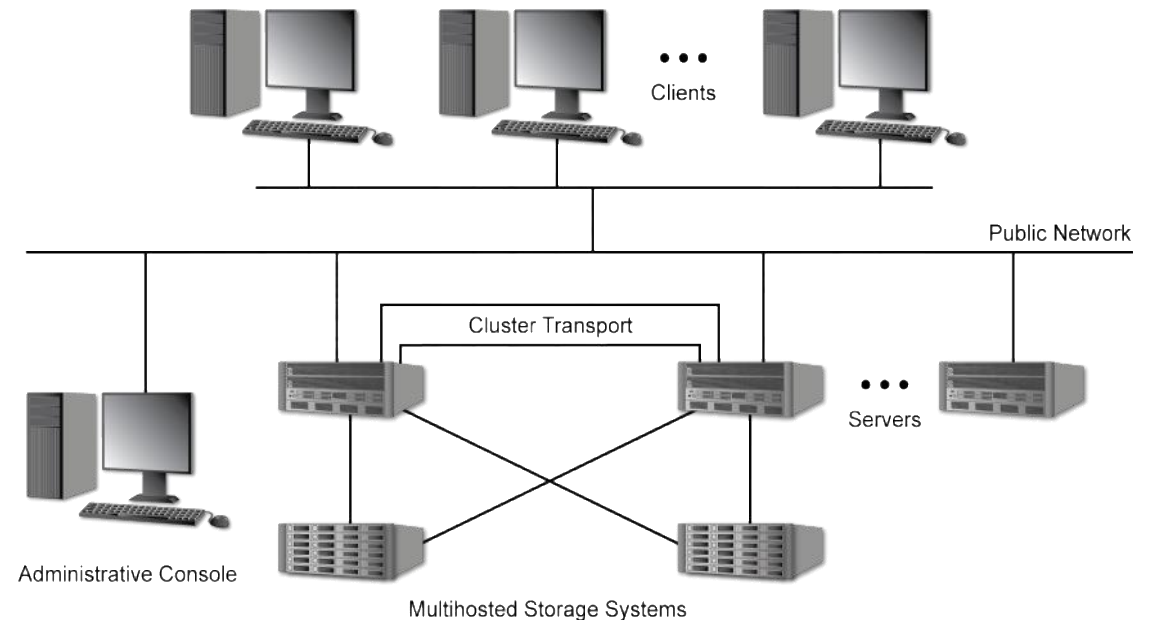
Oracle VM Template for MySQL

- Pre-Installed & Pre-Configured
- Full Integration & QA Testing
- Single Point of Support



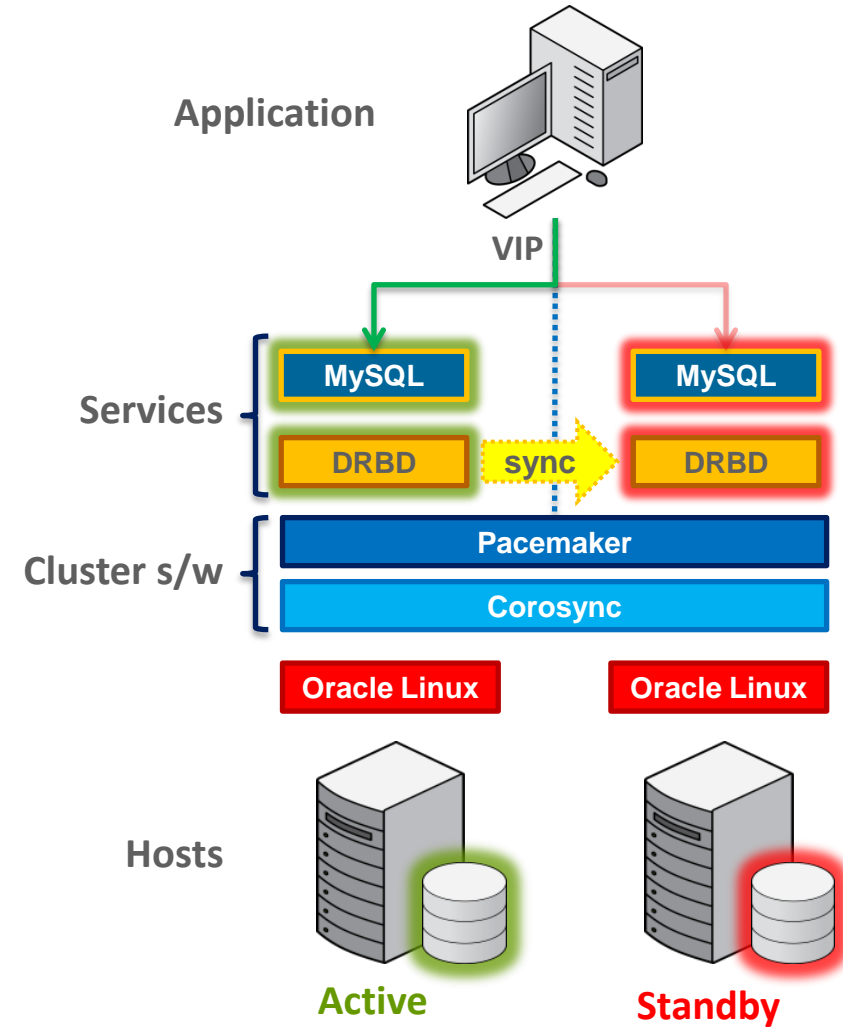
Oracle Solaris Clustering

- Kernel based heartbeating and monitoring
- SPARC and x86. Solaris Virtualization-aware
- MySQL agent included with Oracle Solaris Cluster
- Learn more:
<http://www.oracle.com/technetwork/server-storage/solaris-cluster/overview/index.html>



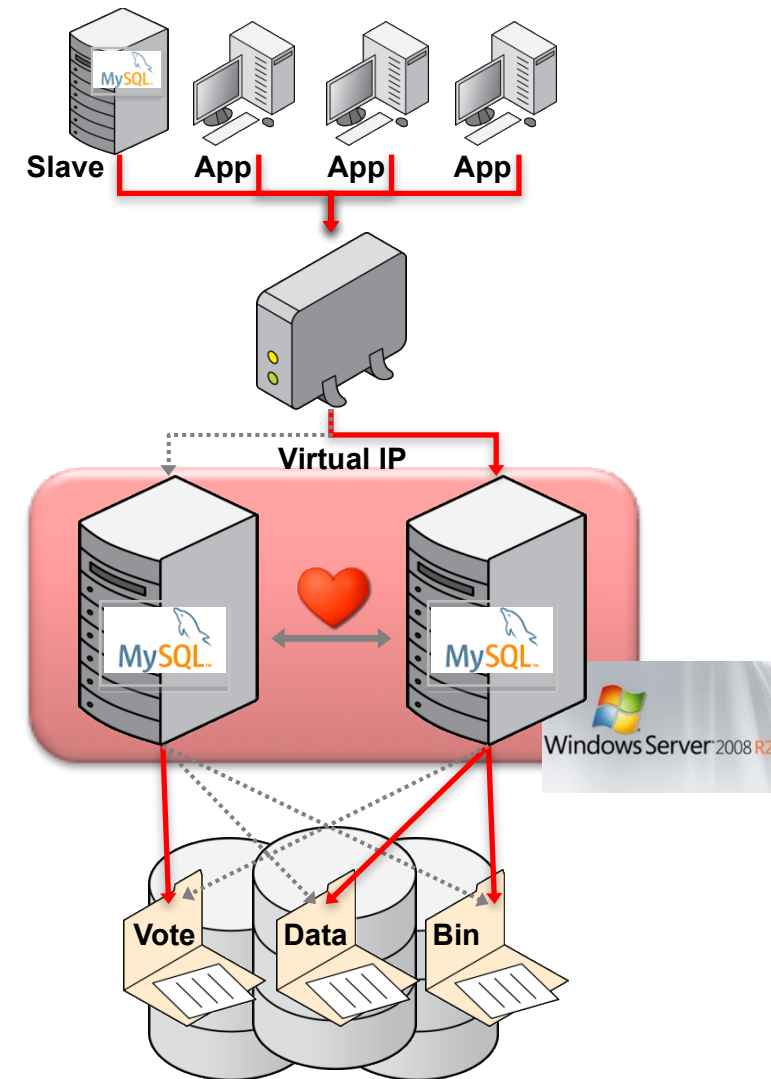
Oracle Linux and DRBD Stack

- Based on distributed storage
 - NOT physical shared storage
- Synchronous replication at block device level eliminates risk of data loss
- Open source, mature & proven
- Certified and supported by Oracle



Windows Server Failover Clustering

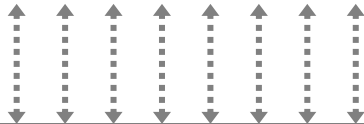
- Native Windows HA clustering with MySQL
- Quorum (3rd vote), data (InnoDB + schema) & binaries (optional) stored in shared storage (iSCSI & FCAL)
- Loss of service = couple of seconds + InnoDB recovery time
- Cluster managed through MS Failover Cluster Management snap-in GUI



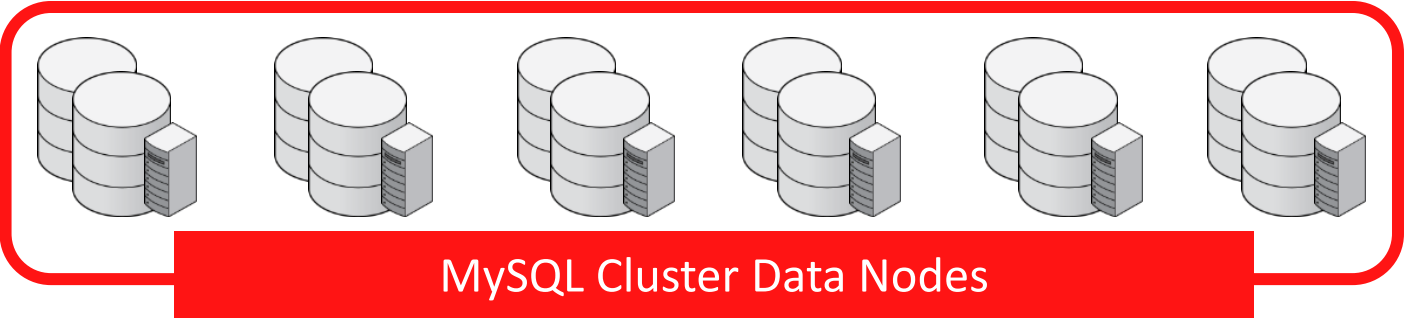
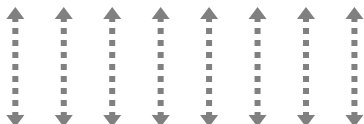
MySQL Cluster Scaling



Clients

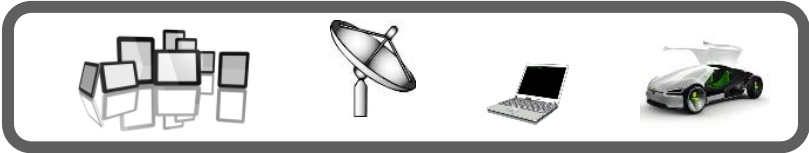


Application Layer

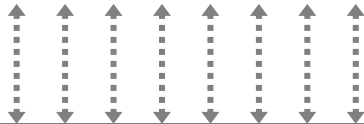


Data Layer

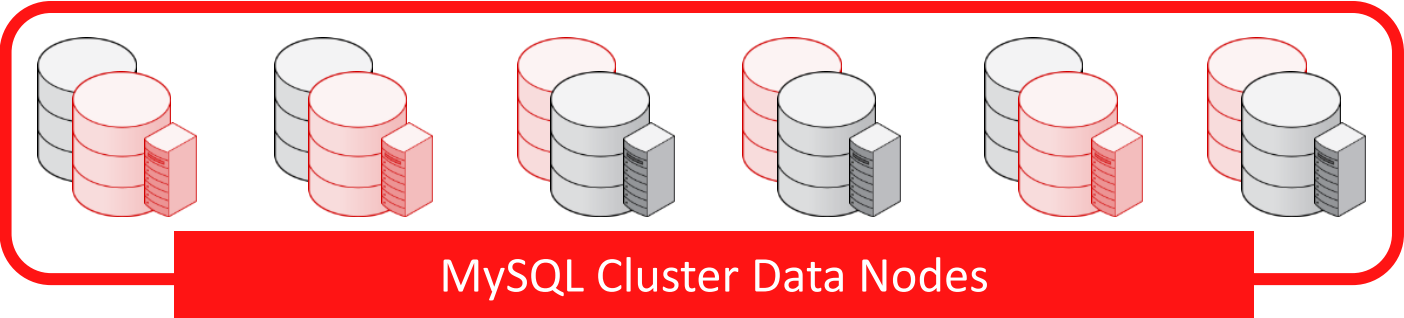
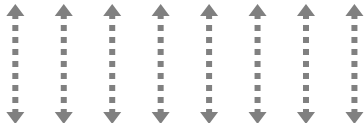
MySQL Cluster HA



Clients



Application Layer



Data Layer

Active-Active Geo-Replication



- Asynchronous replication between MySQL Clusters
- Active-Active
 - Update anywhere
 - Conflict detection
 - Application notified through exception tables
 - Can opt to have conflicts resolved automatically
 - Auto-conflict-resolution
 - Conflicting transaction and dependent ones are rolled-back (MySQL Cluster 7.4 – **pre GA**)
- No changes to application schema

Oracle MySQL HA & Scaling Solutions

	MySQL Replication	MySQL Fabric	Oracle VM Template	Oracle Clusterware	Solaris Cluster	Windows Cluster	DRBD	MySQL Cluster
App Auto-Failover	✘	✓	✓	✓	✓	✓	✓	✓
Data Layer Auto-Failover	✘	✓	✓	✓	✓	✓	✓	✓
Platform Support	All	All	Linux	Linux	Solaris	Windows	Linux	All
Clustering Mode	Master + Slaves	Master + Slaves	Active/Passive	Active/Passive	Active/Passive	Active/Passive	Active/Passive	Multi-Master
Failover Time	N/A	Secs	Secs +	Secs +	Secs +	Secs +	Secs +	< 1 Sec
Scale-out	Reads	✓	✘	✘	✘	✘	✘	✓
Cross-shard operations	N/A	✘	N/A	N/A	N/A	N/A	N/A	✓
Transparent routing	✘	For HA	✓	✓	✓	✓	✓	✓
Shared Nothing	✓	✓	✘	✘	✘	✘	✓	✓
Storage Engine	InnoDB+	InnoDB+	InnoDB+	InnoDB+	InnoDB+	InnoDB+	InnoDB+	NDB
Single Vendor Support	✓	✓	✓	✓	✓	✘	✓	✓

White Papers & Webinar Replays

High Availability

- <http://www.mysql.com/why-mysql/white-papers/#en-22-16>
- <http://www.mysql.com/news-and-events/on-demand-webinars/#en-20-16>

MySQL Replication

- <http://www.mysql.com/why-mysql/white-papers/#en-22-38>
- <http://www.mysql.com/news-and-events/on-demand-webinars/#en-20-38>

MySQL Cluster

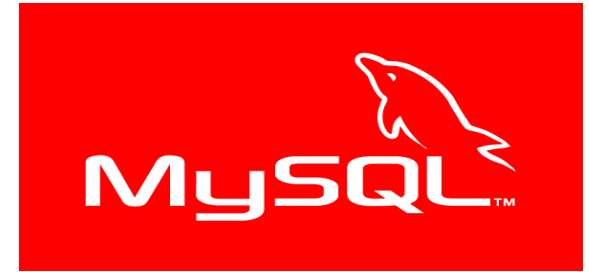
- <http://www.mysql.com/why-mysql/white-papers/#en-22-28>
- <http://www.mysql.com/news-and-events/on-demand-webinars/#en-20-28>

MySQL Central @ OpenWorld

October 25 – 29, San Francisco

- Keynote
- Conferences Sessions
- Birds-of-a-feather sessions
- Tutorials
- Hands-on Labs
- Demos
- Receptions
- OpenWorld Extensive Content





Thanks for attending!

