MySQL Fabric

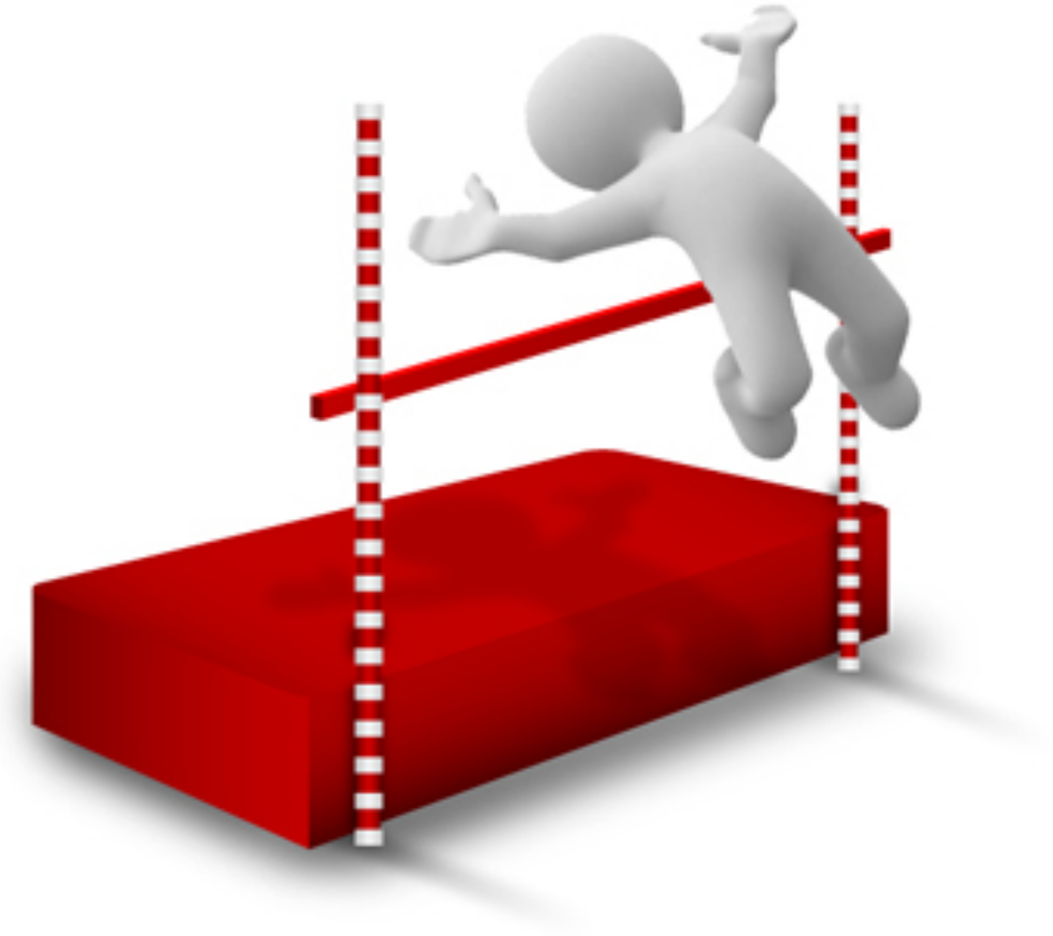
Sharding & High Availability

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.

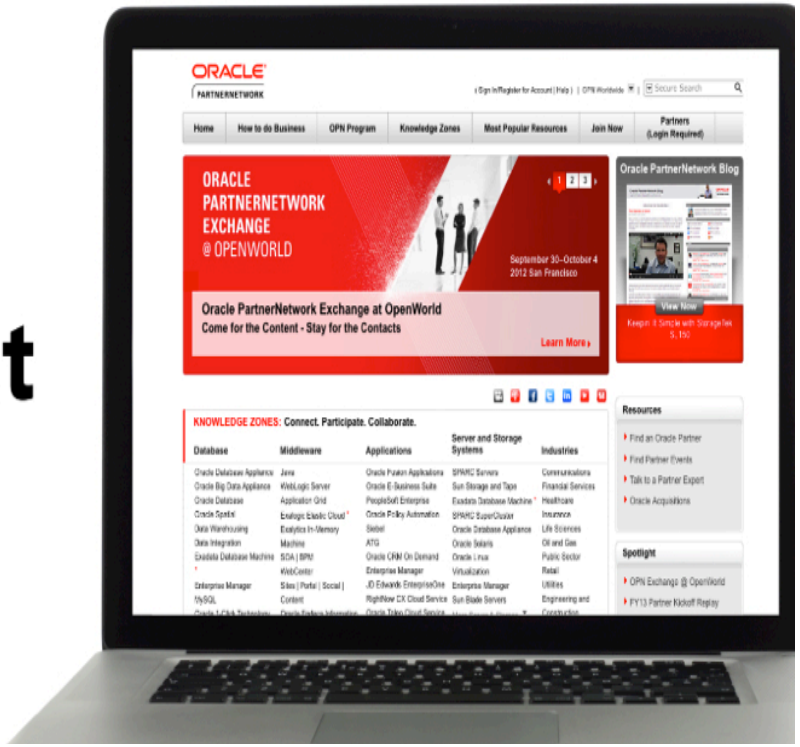
Raising The Bar

Again and Again, Evolving MySQL for You



2.3B
Internet Users

Source: IDC



ORACLE PARTNERNETWORK

| Sign In/Register for Account | Help | | OPN Worldwide | Secure Search

Home | How to do Business | OPN Program | Knowledge Zones | Most Popular Resources | Join Now | Partners (Login Required)

ORACLE PARTNERNETWORK EXCHANGE @ OPENWORLD

September 30–October 4 2012 San Francisco

1 2 3

Oracle PartnerNetwork Exchange at OpenWorld
Come for the Content - Stay for the Contacts

Learn More ›

Oracle PartnerNetwork Blog

View Now

Keepin' it Simple with StorageTek 5,150

KNOWLEDGE ZONES: Connect. Participate. Collaborate.

Database	Middleware	Applications	Server and Storage Systems	Industries
Oracle Database Appliance	Java	Oracle Fusion Applications	SPARC Servers	Communications
Oracle Big Data Appliance	WebLogic Server	Oracle E-Business Suite	Sun Storage and Tape	Financial Services
Oracle Database	Application Grid	PeopleSoft Enterprise	Exadata Database Machine *	Healthcare
Oracle Spatial	Exalogic Elastic Cloud *	Oracle Policy Automation	SPARC SuperCluster	Insurance
Data Warehousing	Exalytics In-Memory Machine	Siebel	Oracle Database Appliance	Life Sciences
Data Integration	SOA | BPM	ATG	Oracle Solaris	Oil and Gas
Exadata Database Machine *	WebCenter	Oracle CRM On Demand	Oracle Linux	Public Sector
Enterprise Manager	Sites | Portal | Social | Content	Enterprise Manager	Virtualization	Retail
MySQL		JD Edwards EnterpriseOne	Enterprise Manager	Utilities
		RightNow CX Cloud Service	Sun Blade Servers	Engineering and

Resources

- Find an Oracle Partner
- Find Partner Events
- Talk to a Partner Expert
- Oracle Acquisitions

Spotlight

- OPN Exchange @ OpenWorld
- FY13 Partner Kickoff Replay

1.1B Global
3G mobile subscribers

Source: Mobithink

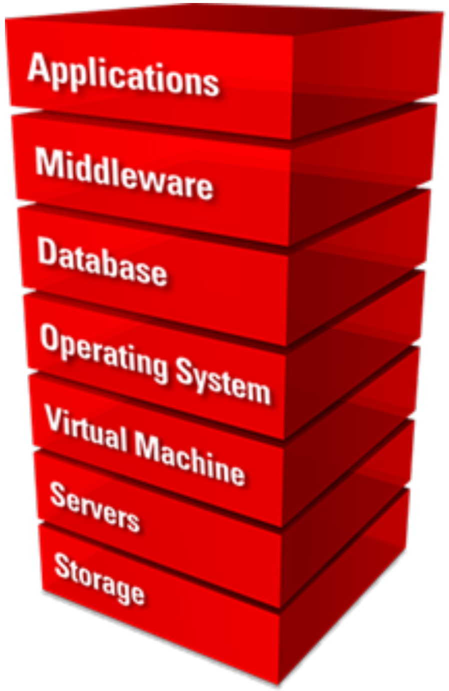
3B to 50B
Devices

Source: Ericsson

Complete Solutions

On Premises and in the Cloud

- Best of breed components at every level of the stack
- Complete: Meets most customer requirements



That’s why MySQL Matters to Oracle and its customers

Program Agenda

1. Requirements for Next Gen Services
2. Simple, transparent High Availability
3. Delivering SQL & ACID at scale
4. Where MySQL Fabric fits
5. Getting started

Program Agenda

1 Requirements for Next Gen Services

2 Simple, transparent High Availability

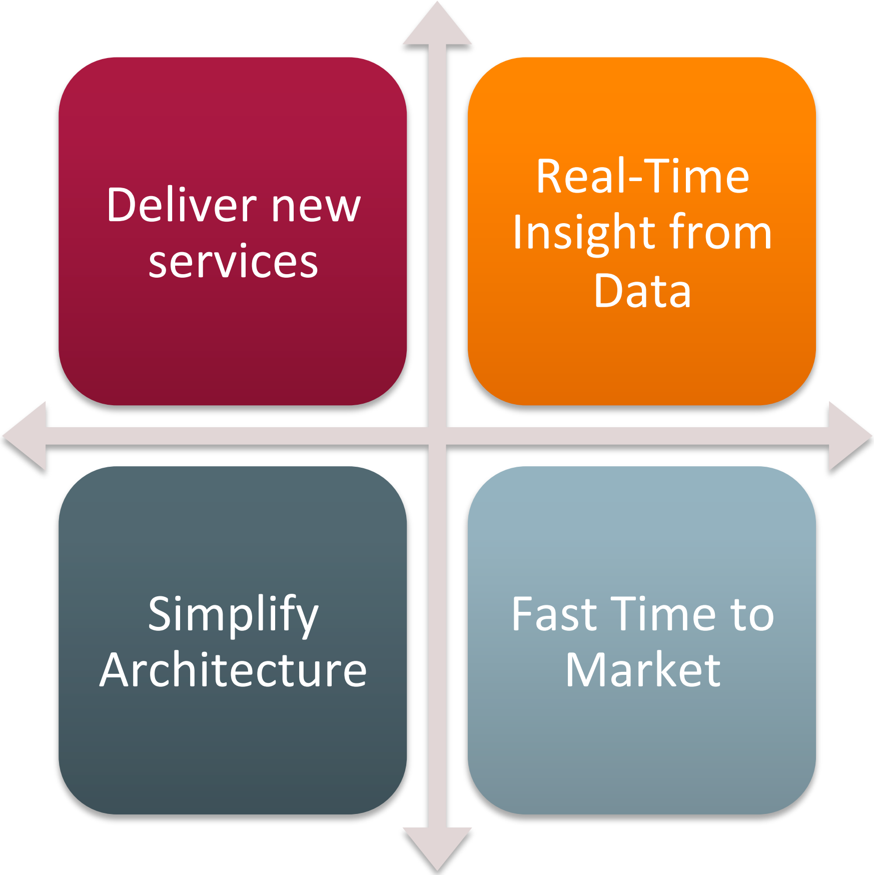
3 Delivering SQL & ACID at scale

4 Where MySQL Fabric fits

5 Getting started

Business Priorities for IT

Focus on driving the business rather than on infrastructure



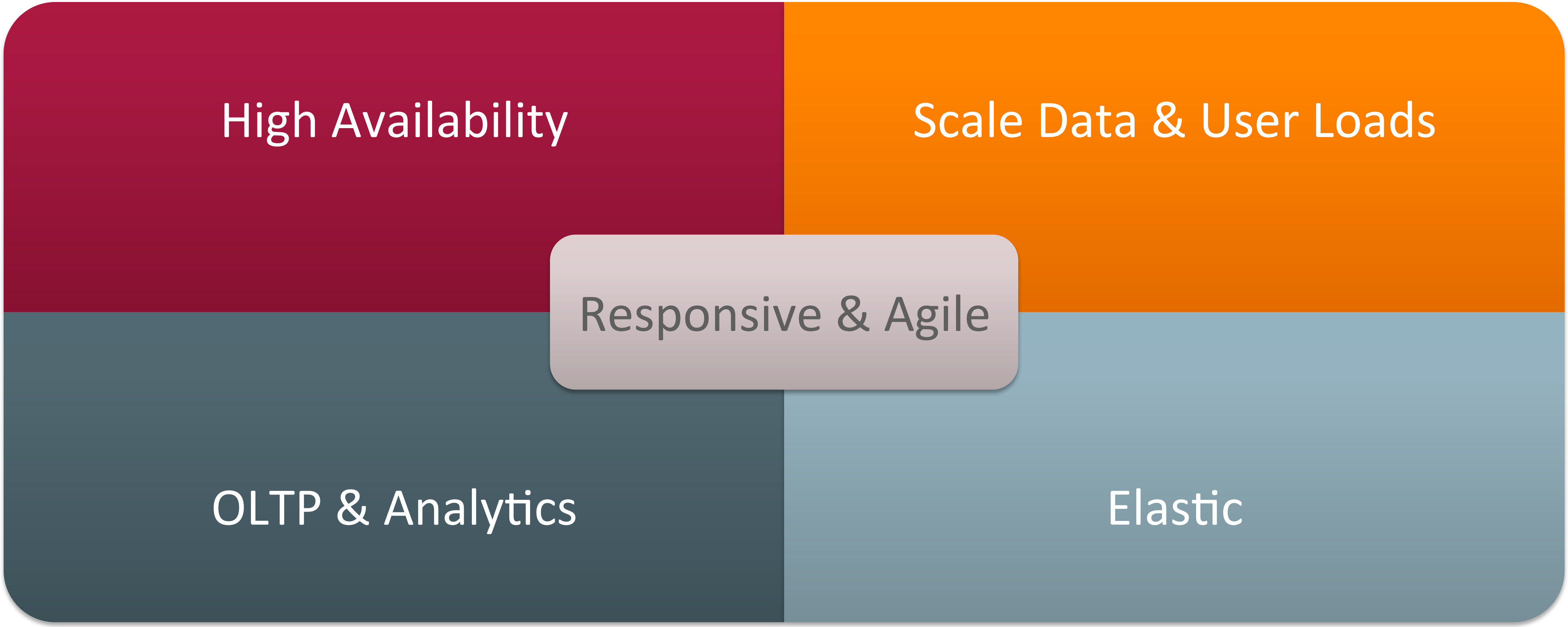
Deliver new services

Real-Time Insight from Data

Simplify Architecture

Fast Time to Market

Application Requirements



High Availability

Scale Data & User Loads

Responsive & Agile

OLTP & Analytics

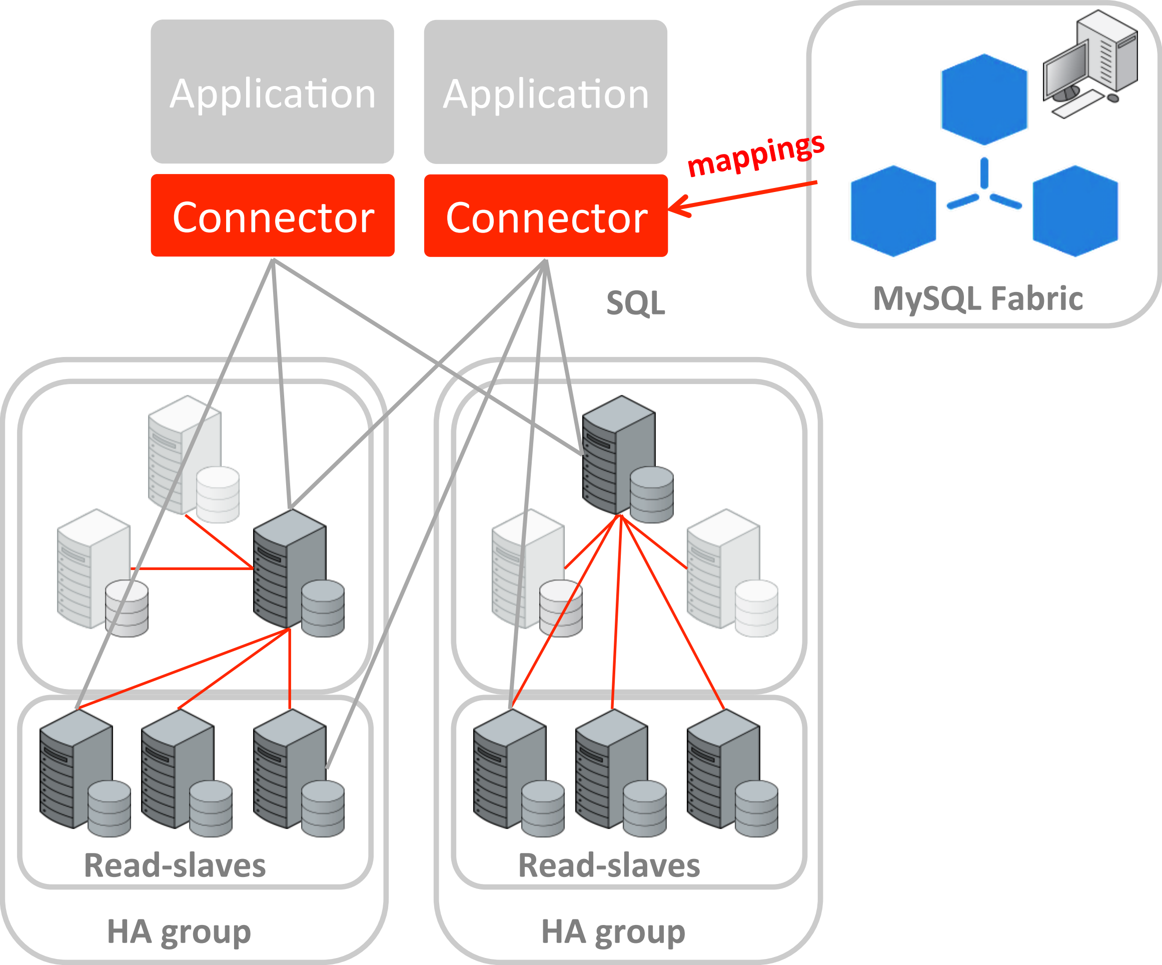
Elastic

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 (pre-production), .NET, C (labs)
  - Lower latency, bottleneck-free
- Server provisioning using OpenStack etc.

Program Agenda

1 Requirements for Next Gen Services

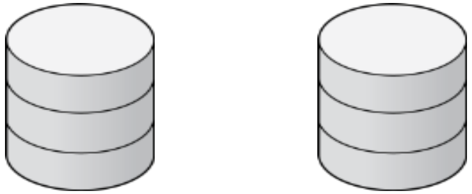
2 Simple, transparent High Availability

3 Delivering SQL & ACID at scale

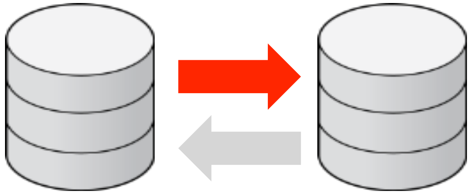
4 Where MySQL Fabric fits

5 Getting started

Layers of HA

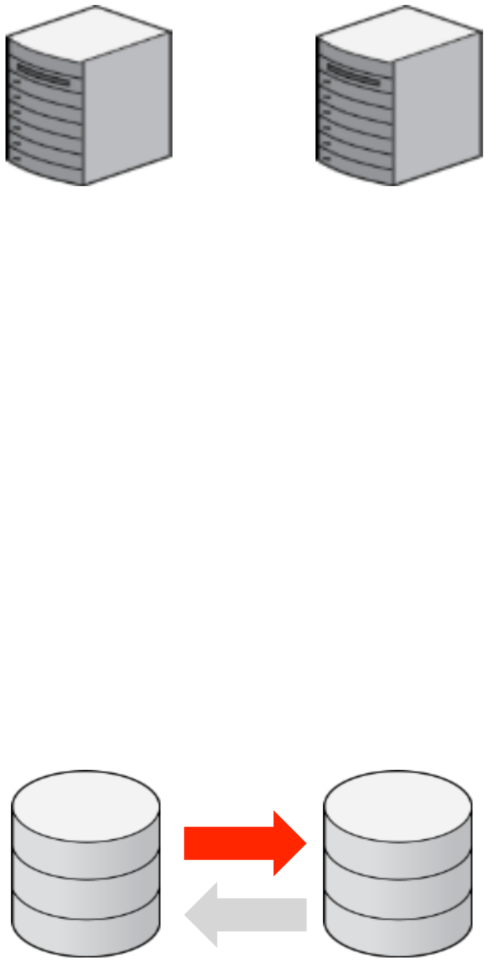


Layers of HA



Data Redundancy

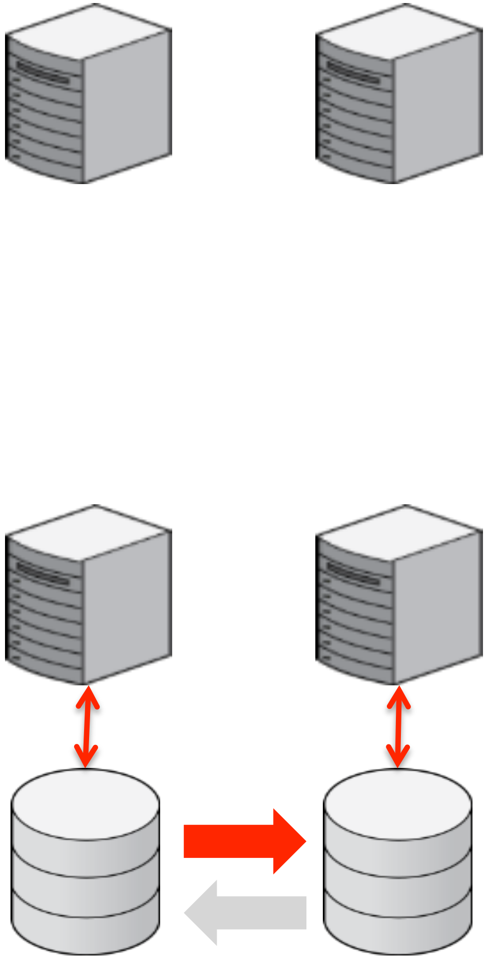
Layers of HA



Redundant App Servers

Data Redundancy

Layers of HA

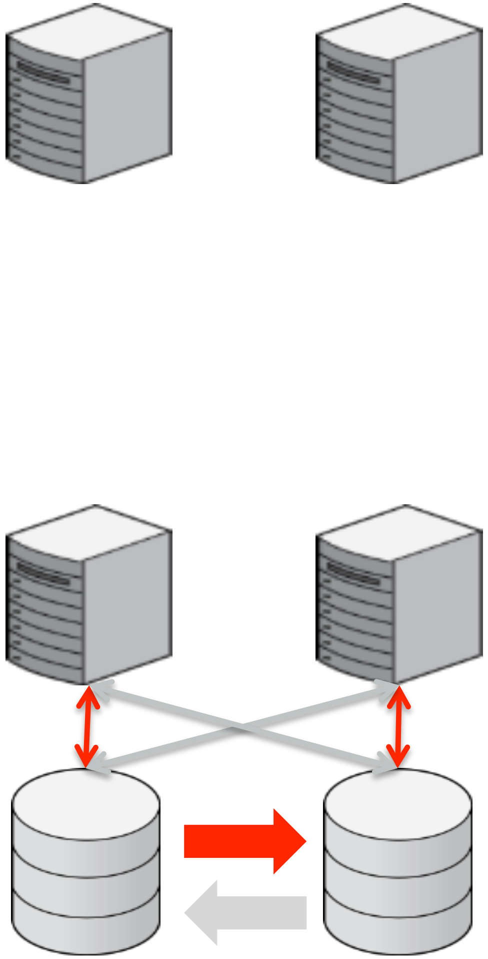


Redundant App Servers

Redundant Access to Data

Data Redundancy

Layers of HA

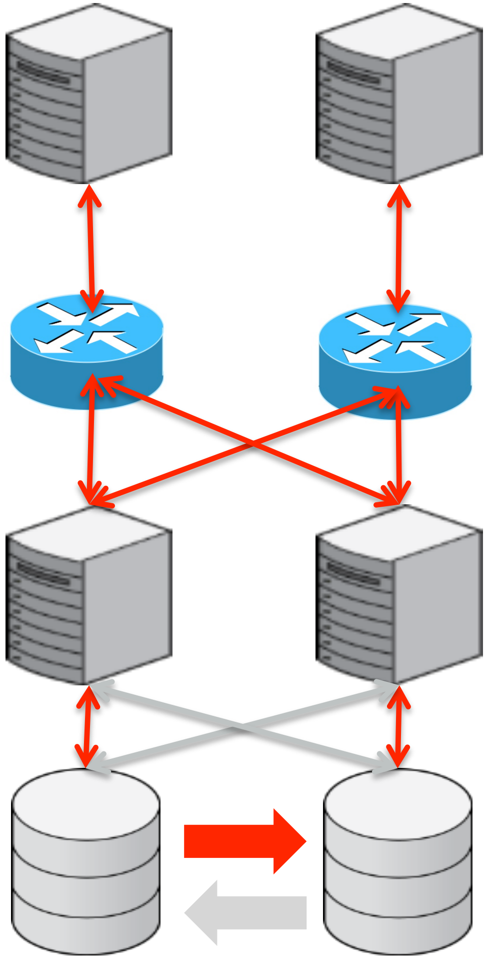


Redundant App Servers

Redundant Access to Data

Data Redundancy

Layers of HA



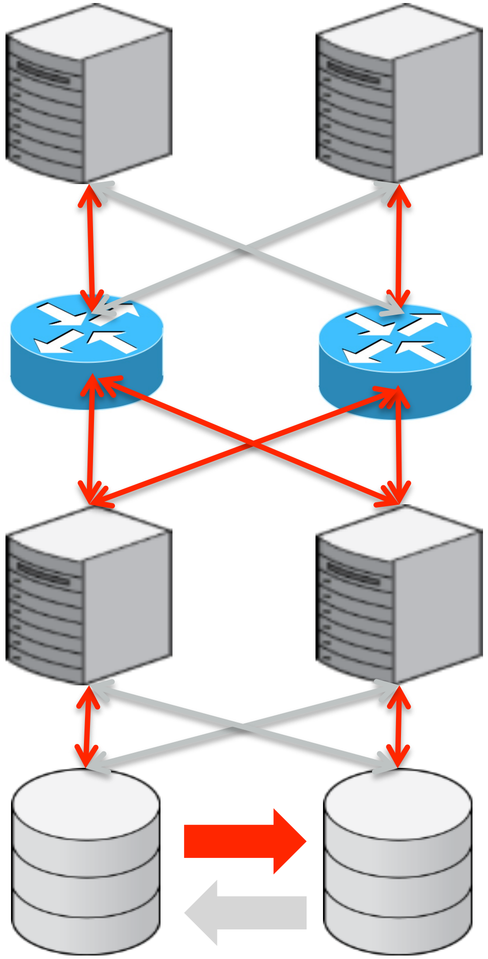
Redundant App Servers

Routing to the Data

Redundant Access to Data

Data Redundancy

Layers of HA



Redundant App Servers

Routing to the Data

Redundant Access to Data

Data Redundancy

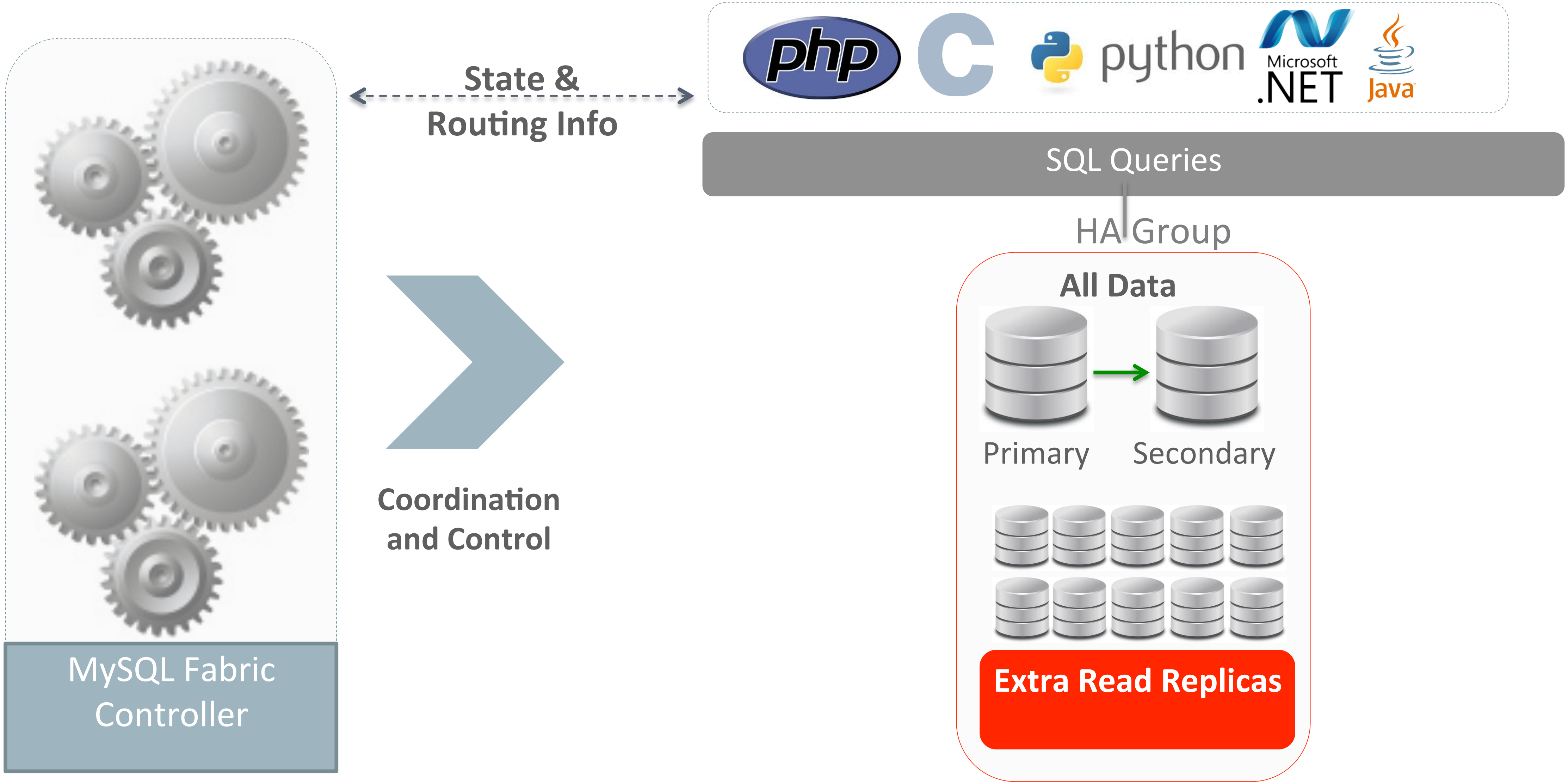
Asynchronous vs. Synchronous Replication

- Asynchronous
  - MySQL Default
  - In **parallel**: Master acks to app and sends transaction to slave
    - Fast
    - Risk of lost changes if master dies

- 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

- 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)

MySQL Fabric Framework (HA)



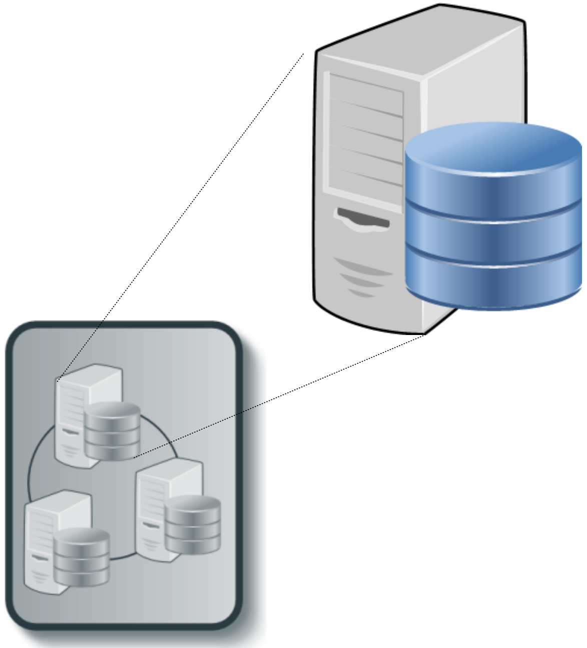
MySQL Replication & MySQL Fabric HA

How this effects failover

- MySQL Replication is the initial implementation used in HA Groups
  - **PRIMARY** = Replication Master & receives all writes
  - **SECONDARY** = Replication Slave & receives share of reads
- Failover
  - MySQL Fabric detects failure of PRIMARY/Master
  - Selects a SECONDARY/Slave and promotes it
  - Updates State Store
  - Updated state fetched by Fabric-aware connectors

High-Availability Group Concept

- Abstract Concept
  - Set of servers
  - Server attributes
- Connector Attributes
  - Connection information
  - **Mode**: read-only, read-write, ...
  - **Weight**: distribute load
- Management Attributes
  - **State**: state/role of the server



State:	Primary
Mode:	Read-Write
Host:	server-1.example.com

Create HA Groups and add Servers

- Define a group

```
mysqlfabric group create my_group
```

- Add servers to group

```
mysqlfabric group add my_group server1.example.com

mysqlfabric group add my_group server2.example.com
```

Create HA Groups and add Servers

- Promote one server to be primary

```
mysqlfabric group promote my_group
```

- Tell failure detector to monitor group

```
mysqlfabric group activate my_group
```

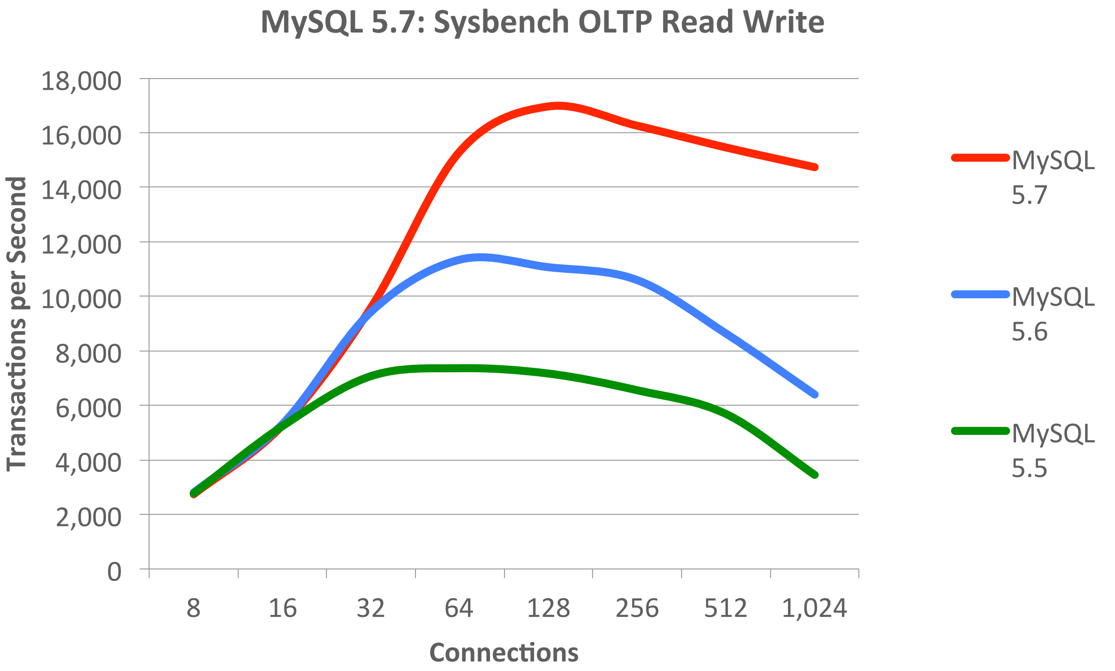
Program Agenda

1. Requirements for Next Gen Services
2. Simple, transparent High Availability
3. **Delivering SQL & ACID at scale**
4. Where MySQL Fabric fits
5. Getting started

The Path to Scalability

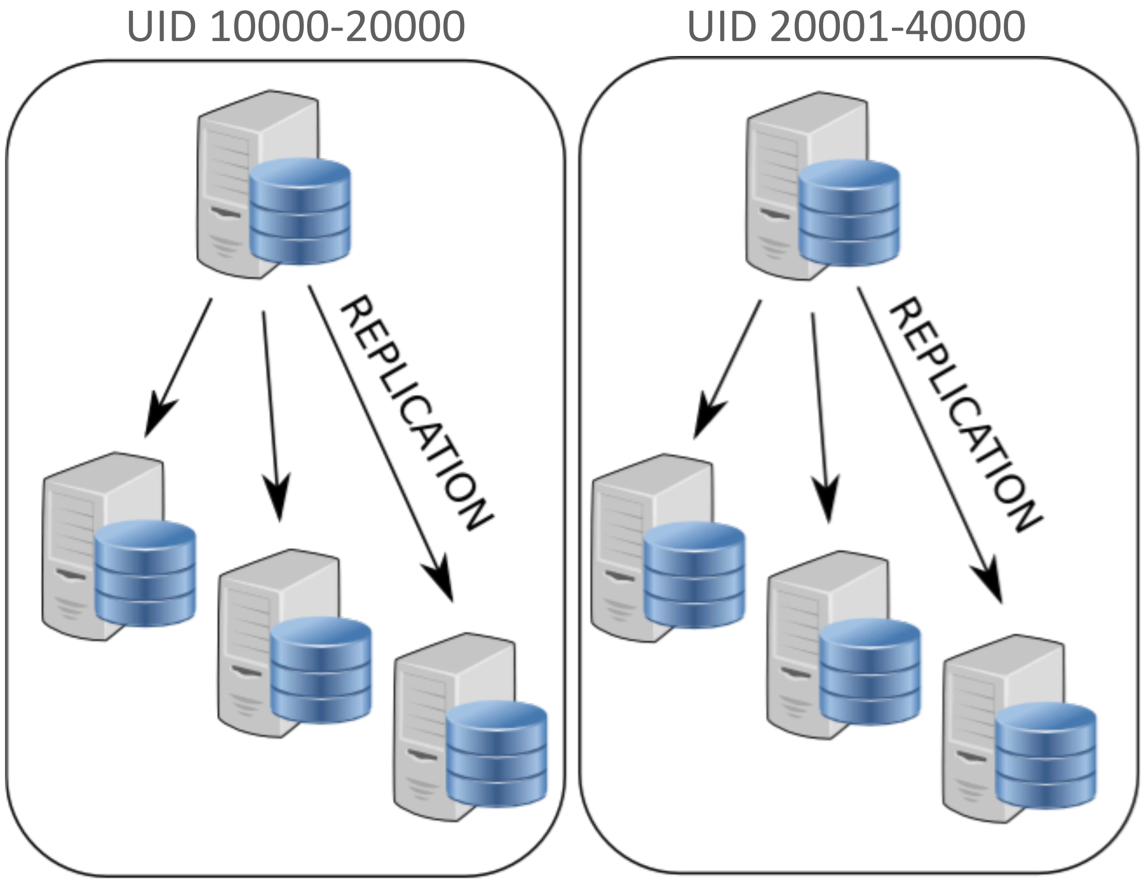
Scaling-Up can take you a long way

- Scaling on dense, multi-core, multi-thread servers
  - 10s - 100GBs RAM
  - SSDs
- Scale across cores within a single instance
- You can get a long way with MySQL 5.6 & 5.7!



Benefits of Sharding

- Write scalability
  - Can handle more writes
- Large data set
  - Database too large
  - Does not fit on single server
- Improved performance
  - Smaller index size
  - Smaller working set
  - Improve performance (reads and writes)

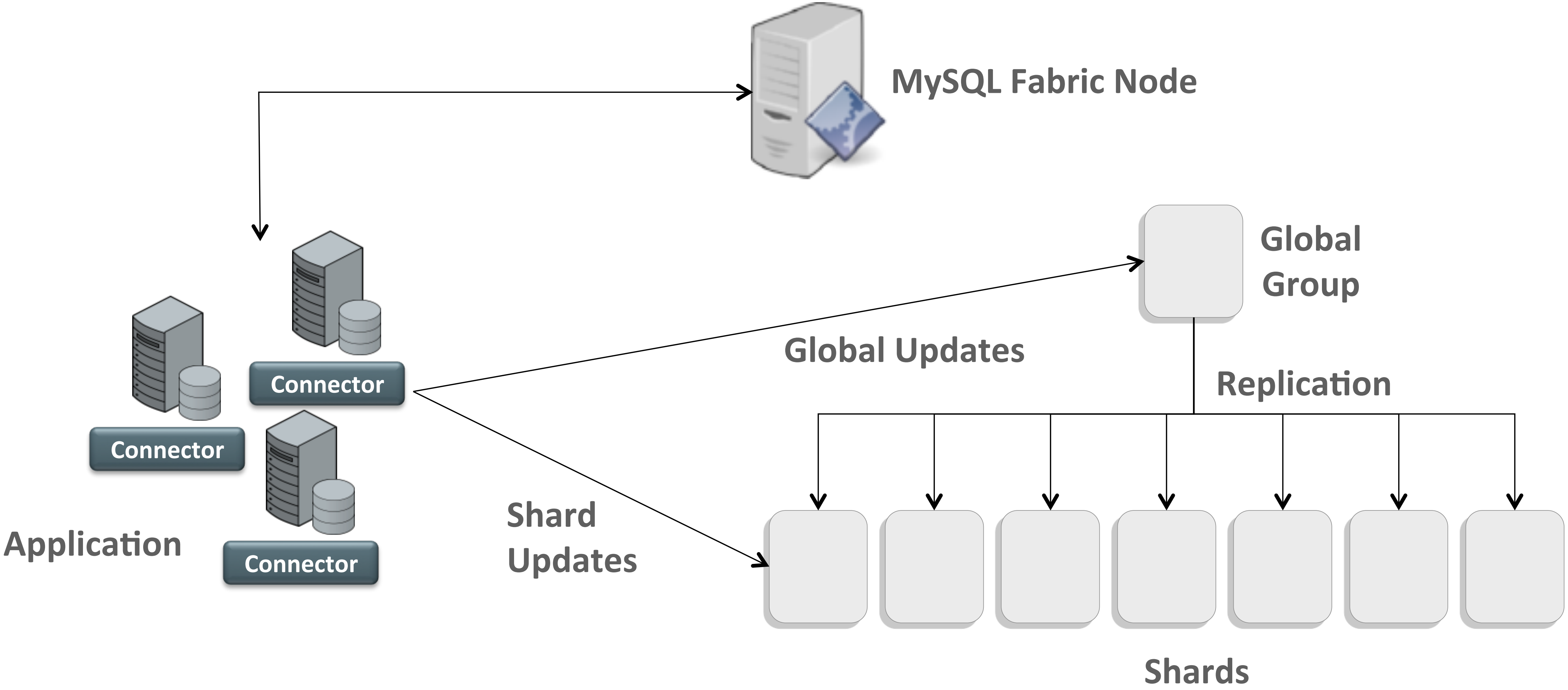


MySQL Fabric Sharding & Provisioning Features

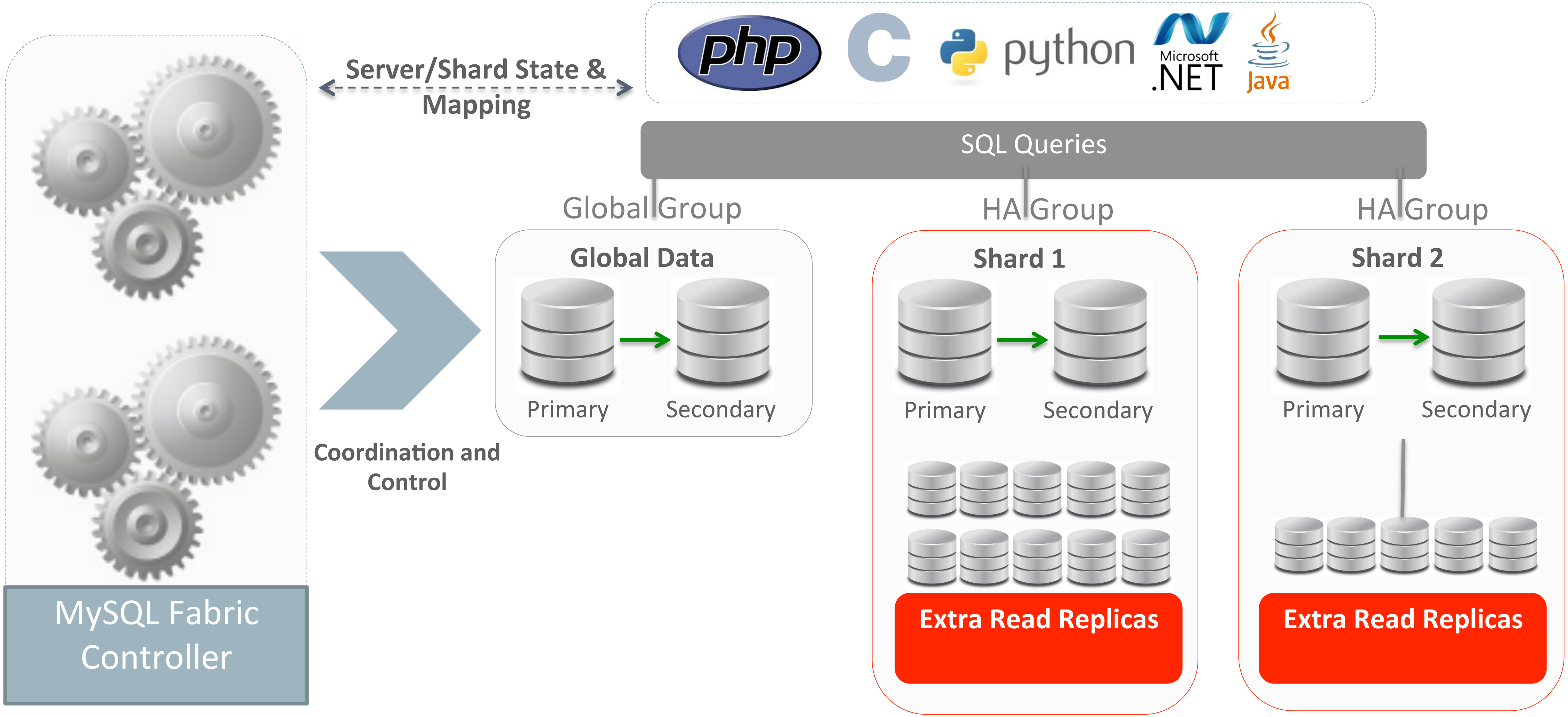
- Connector API Extensions
  - Support Transactions
  - Support full SQL
- Decision logic in connector
  - Reducing network load
- Shard Multiple Tables
  - Using same key
- Global Updates
  - Global tables
  - Schema updates

- Sharding Functions
  - Range
  - (Consistent) Hash
- Shard Operations
  - Shard move
  - Shard split
- Server Provisioning
  - Integrated with OpenStack & other frameworks

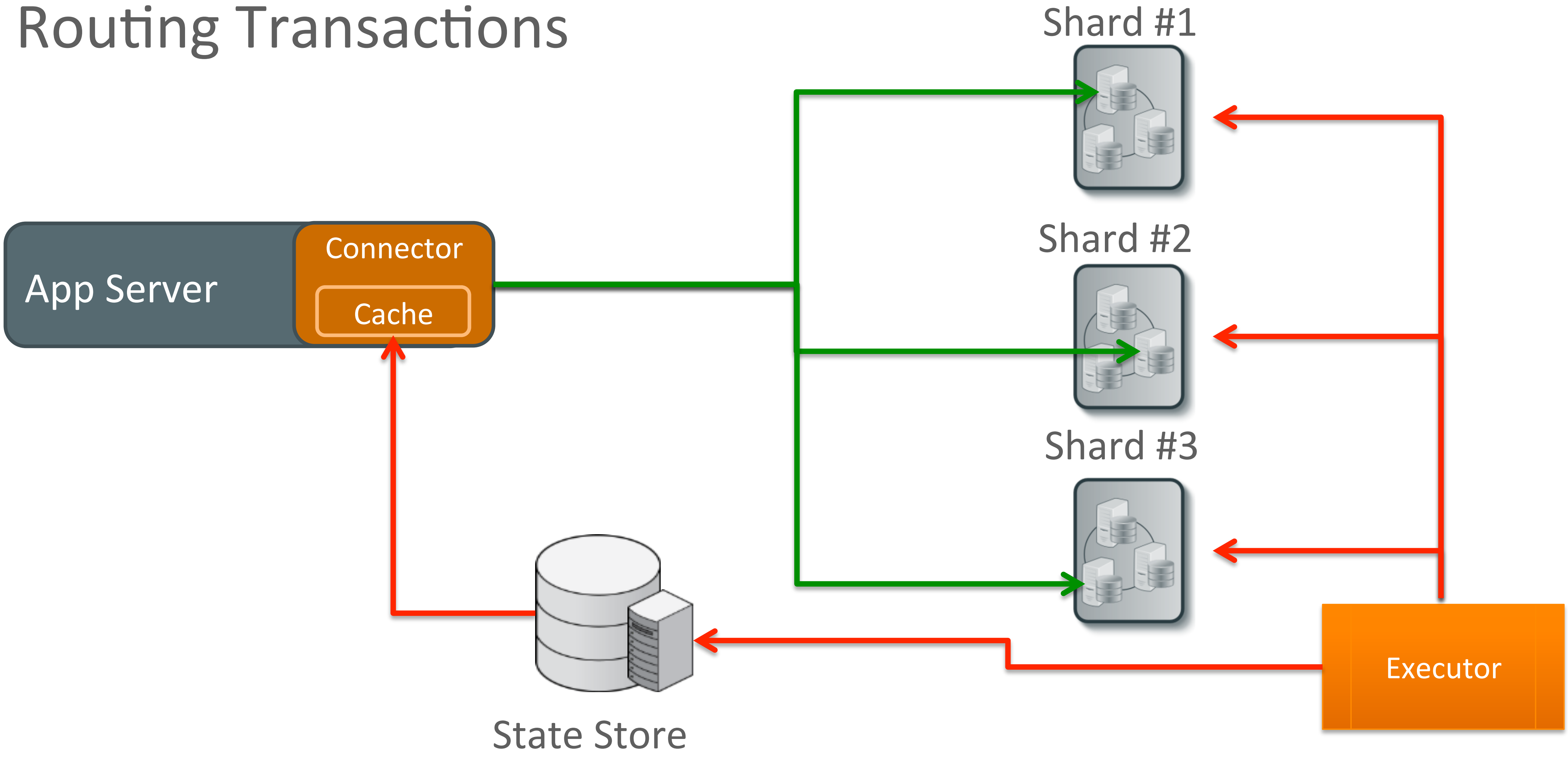
Sharding Architecture



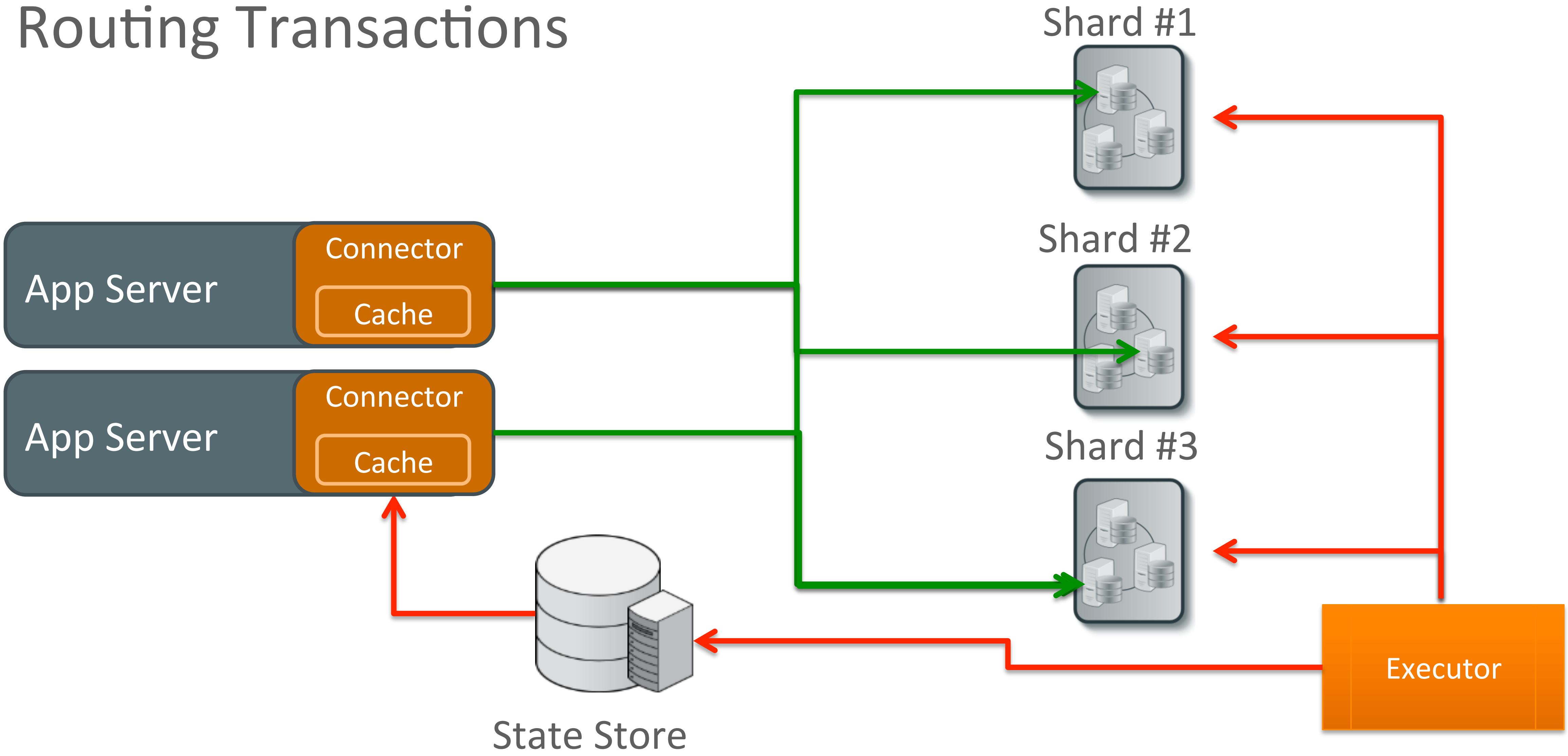
MySQL Fabric (HA + Sharding)



Routing Transactions



Routing Transactions



MySQL Fabric: Sharding Setup

- Set up some groups
  - **`my_global`** – for global updates
  - **`my_group.N`** – for the shards
  - Add servers to the groups
- Create a shard mapping
  - A “distributed database”
  - Mapping keys to shards
  - Give information on what tables are sharded
- Add shards

MySQL Fabric:

Moving and Splitting Shards

- Moving a shard (id=5) from existing group to another (my_group.8)

```
mysqlfabric sharding move 5 my_group.8
```

- Splitting a shard (id=5) into two parts with new half stored in group my_group.6

```
mysqlfabric sharding split 5 my_group.6
```

Connector API: Shard Specific Query

- Indicate tables to be used in query
  - **Property**: tables
  - Fabric will compute map
- Indicate read-only queries
  - **Property:** mode

- Provide sharding key
  - **Property**: key
  - Fabric will compute shard
- Joins within the shard (or with global tables) supported

```
conn.set_property(tables=["test.subscribers"], key=sub_no, mode=fabric.MODE_READONLY)
cur = conn.cursor()
cur.execute(
    "SELECT first_name, last_name FROM subscribers WHERE sub_no = %s", (sub_no)
    )
for row in cur:
    print row
```

Connector API: Global Update

- Set global scope
  - **Property**: scope
  - Query goes to global group

```
conn.set_property(tables=[], scope='GLOBAL')
cur = conn.cursor()
cur.execute("ALTER TABLE test.subscribers ADD nickname VARCHAR(64)")
```

Server Provisioning – OpenStack Nova Integration

```
> mysqlfabric provider register
my_stack \
 my_user my_password \
 http://8.21.28.222:5000/v2.0/ \
 --tenant=my_user_role \
 --provider_type=OPENSTACK
```

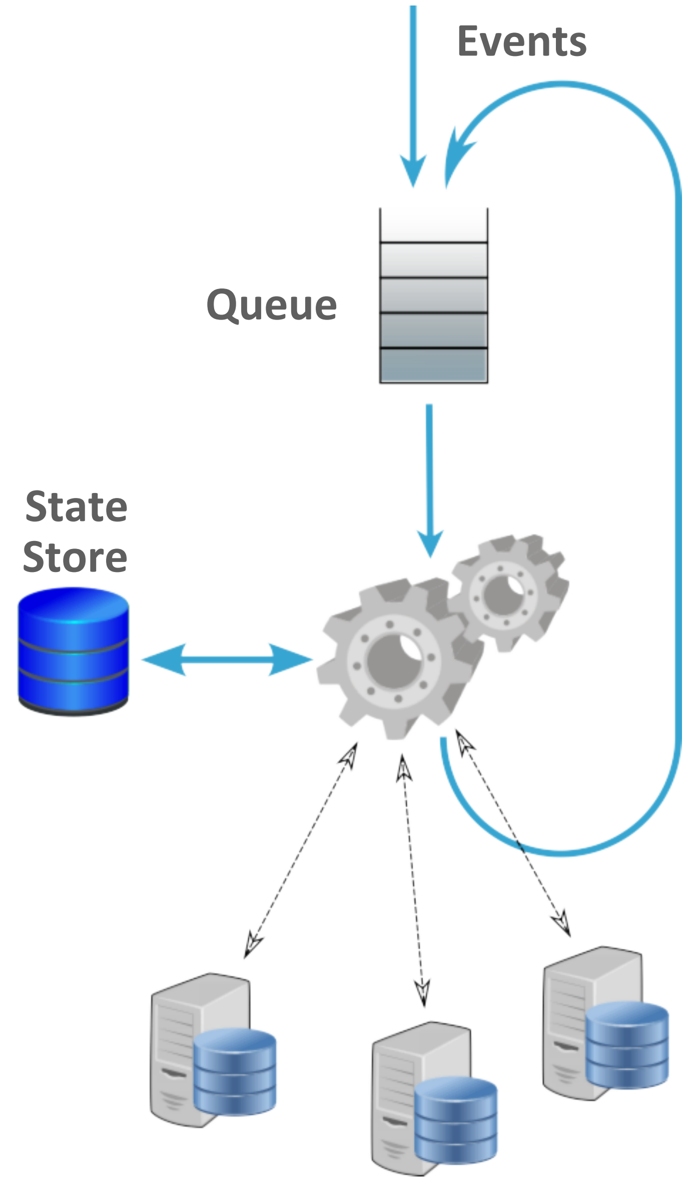
```
> mysqlfabric machine create
my_stack \
 --image id=8c92f0d9-79f1-4d95-
b398-86bda7342a2d \
 --flavor name=m1.small
```

```
> mysqlfabric machine list my_stack
```

- Fabric creates new machines, & MySQL Servers
  - Initially using OpenStack Nova
  - Other frameworks on the way (OpenStack Trove, AWS,…)
- Server setup
  - Clones slave
  - Sets up replication
  - Performs custom operations

MySQL Fabric executor

- Event driven
  - Events will trigger execution of procedures
  - Procedures can trigger events themselves
  - Each step of a procedure is called a *job*
- Procedures
  - Written in Python
  - Interacts with servers
  - Write state changes into state store
  - Lock manager for conflict resolution
    - Conservative two-phase locking strategy
    - Avoid deadlocks



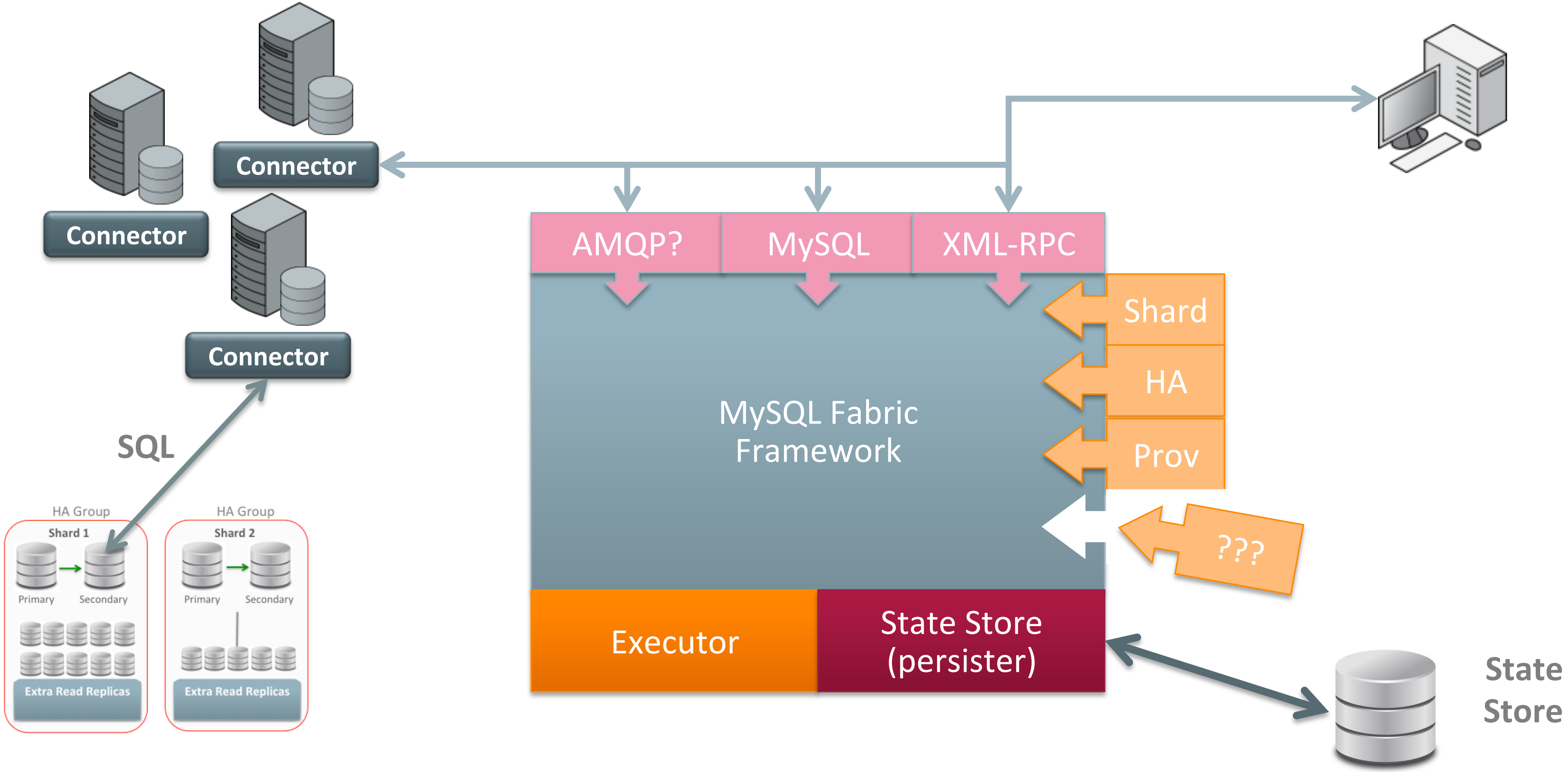
Example of User-Defined Executor Script:
Automatically replacing a server in a group on failure

- Register procedure for event
  - `@on_event` decorator
  - Accept event to register for
- Fetch the group the server belonged to
- Fetch a new server from the provider
- Add the server to the group

```
@on_event(SERVER_LOST)
def _add_server(group_id, server_uuid):
  group = Group.fetch(group_id)
  machines = PROVIDER.create_machines(
          parameters
    )
  server = MySQLServer( server_uuid,
    address
    )
  MySQLServer.add(server)
  group.add(server)
  _configure_as_slave(server)
```

MySQL Fabric Node

Extensible Architecture



MySQL Fabric: Goals & Features

- Connector API Extensions
  - Support Transactions
  - Support full SQL
- Fabric-Aware Connectors at GA:
  - PHP (pre-GA) + Doctrine, Python, Java + Hibernate, .NET, C (pre-GA)
- Decision logic in connector
  - Reducing latency &network load
- Load Balancing
  - Read-Write Split
  - Distribute transactions

- Global Updates
  - Global data
  - Schema updates
- Sharding Functions
  - Range
  - (Consistent) Hash
- Shard Operations
  - Shard move
  - Shard split
- Server Provisioning
  - OpenStack Integration (& other frameworks)

MySQL Fabric – Current Limitations

- Routing is dependent on Fabric-aware connectors
  - Currently Java (+ Hibernate), PHP (pre-GA), Python, .NET & C (labs)
- MySQL Fabric node is a single (non-redundant process)
  - HA Maintained as connectors continue to route using local caches
- Establishes asynchronous replication
  - Manual steps to switch to semisynchronous

- Sharding not completely transparent to application (must provide shard key – column from application schema)
- No cross-shard joins or other queries
- Management is through CLI, MySQL protocol or XML/RPC API
  - No GUI

Program Agenda

1 Requirements for Next Gen Services

2 Simple, transparent High Availability

3 Delivering SQL & ACID at scale

4 Where MySQL Fabric fits

5 Getting started

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	✖	✔	✔	✔	✔	✔	✔	✔
Zero Data Loss	MySQL 5.7	MySQL 5.7	✔	✔	✔	✔	✔	✔
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	✔	✔	✔	✔	✔	✖	✔	✔

Program Agenda

1 Requirements for Next Gen Services

2 Simple, transparent High Availability

3 Delivering SQL & ACID at scale

4 Where MySQL Fabric fits

5 Getting started

Oracle University MySQL Training Services

Prepare Your Organization to Enable Reliable and High-Performance Web-Based Database Applications

RECENTLY RELEASED

!!ALL NEW!! MySQL Cluster Training

To Register your interest to influence the schedule on this newly released course – go to education.oracle.com/mysql and click on the **MySQL Cluster Course**

"Training and team skill *have the most significant impact on overall performance of technology and success of technology projects." - IDC, 2013*

Premier Support customers eligible to save 20% on learning credits.

Benefits

- Expert-led training to support your MySQL learning needs
- Flexibility to train in the classroom or online
- Hands-on experience to gain real world experience
- Key skills needed for database administrators and developers

Top Courses for Administrators and Developers

- MySQL for Beginners
- MySQL for Database Administrators
- MySQL Performance Tuning
- MySQL Cluster – **NEW - Register Your Interest!**
- MySQL and PHP - Developing Dynamic Web Applications
- MySQL for Developers
- MySQL Developer Techniques

Top Certifications

- MySQL 5.6 Database Administrator
- MySQL 5.6 Developer

To find out more about available MySQL Training & Certification offerings, go to: education.oracle.com/mysql

MySQL Fabric Resources

- Download and try
  http://dev.mysql.com/downloads/fabric/
- Documentation
  http://dev.mysql.com/doc/mysql-utilities/en/fabric.html
- MySQL Fabric on the web
  http://www.mysql.com/products/enterprise/fabric.html
- Forum (MySQL Fabric, Sharding, HA, Utilities)
  http://forums.mysql.com/list.php?144
- Tutorial: MySQL Fabric - adding High Availability and Scaling to MySQL
  http://www.mysqlhighavailability.com/mysql-fabric/mysql-fabric-adding-high-availability-and-scaling-to-mysql
- White Paper: MySQL Fabric - A Guide to Managing MySQL High Availability and Scaling Out
  http://www.mysql.com/why-mysql/white-papers/mysql-fabric-product-guide
- Webinar Replays
  http://www.mysql.com/news-and-events/on-demand-webinars/#en-20-41

ORACLE®