Always On, Without Fail: MySQL Group Replication

Luís Soares (luis.soares@oracle.com)
Principal Software Engineer, MySQL Replication Lead

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.

Program Agenda

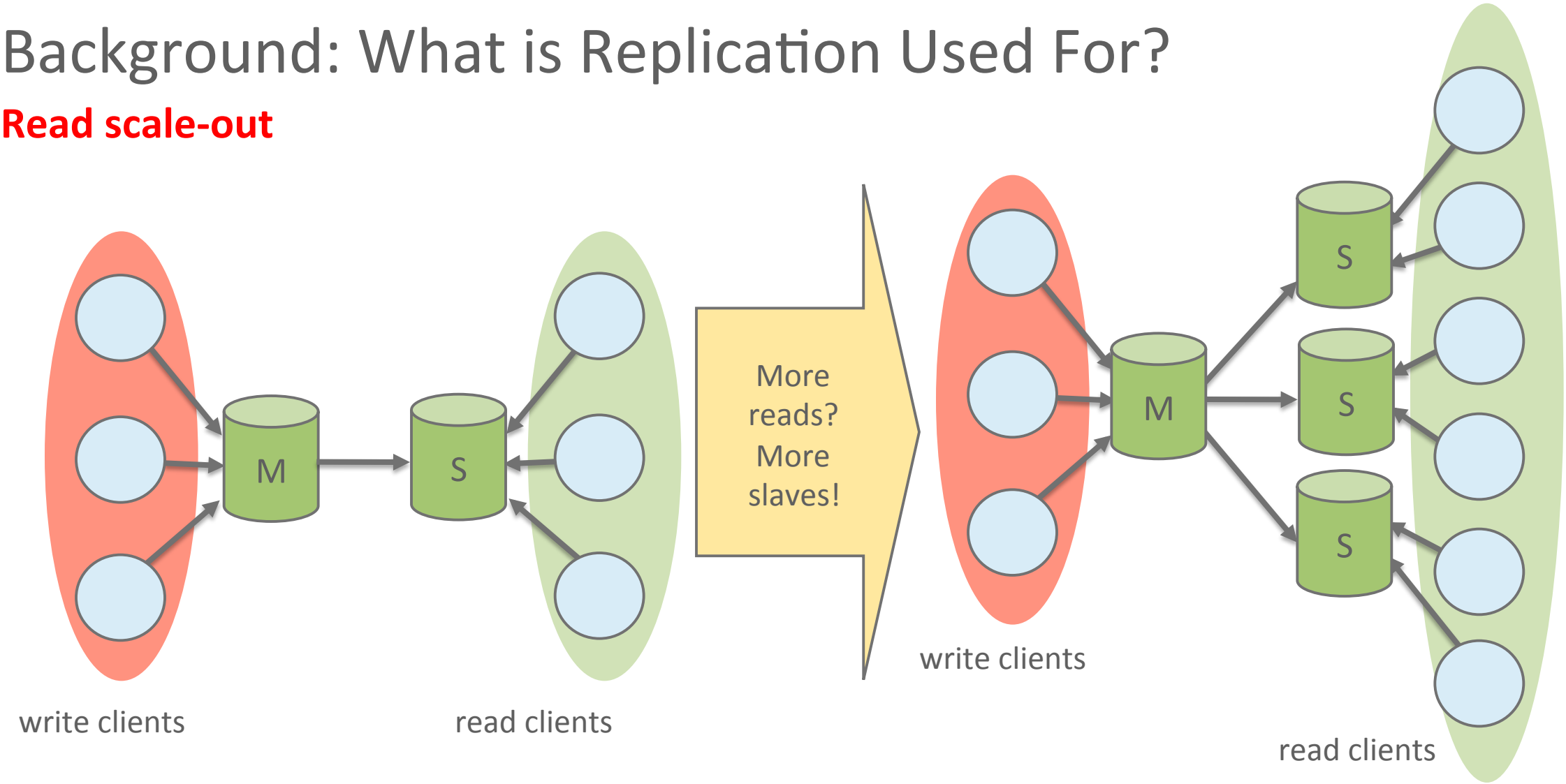
Program Agenda

1. Background
2. MySQL Group Replication
3. Architecture
4. Use cases
5. Performance
6. Conclusion

1 Background

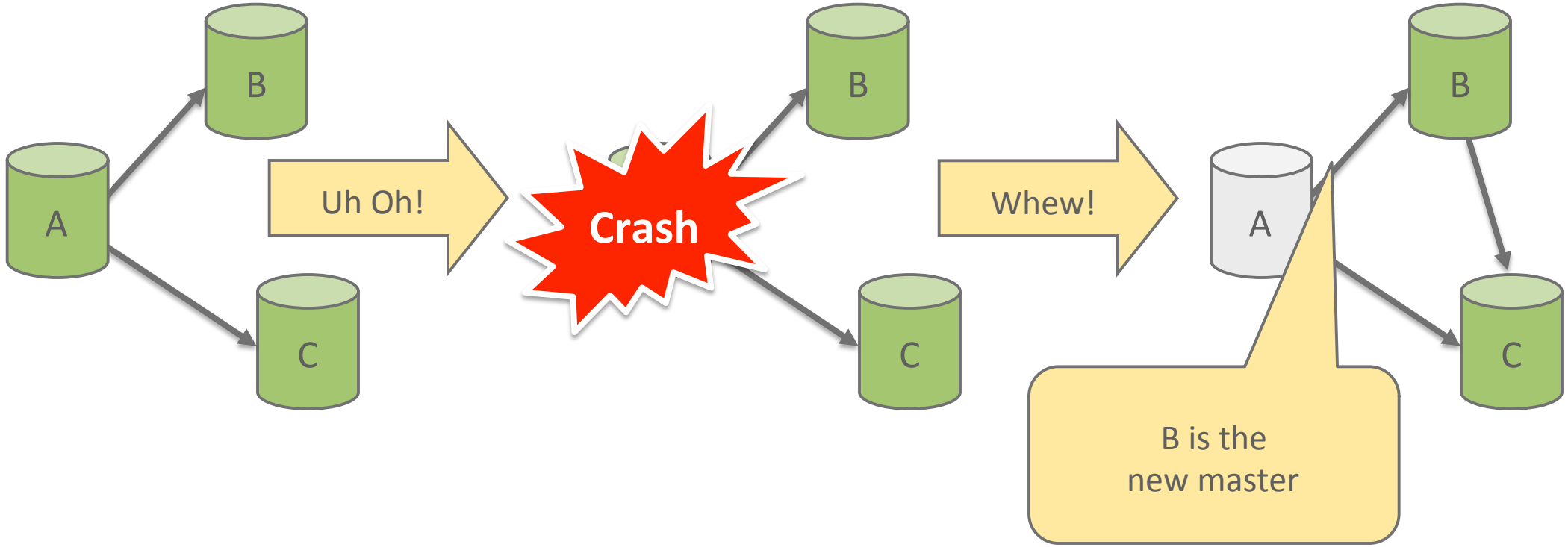
Background: What is Replication Used For?

Read scale-out



Background: What is Replication Used For?

Redundancy: If master crashes, promote slave to master



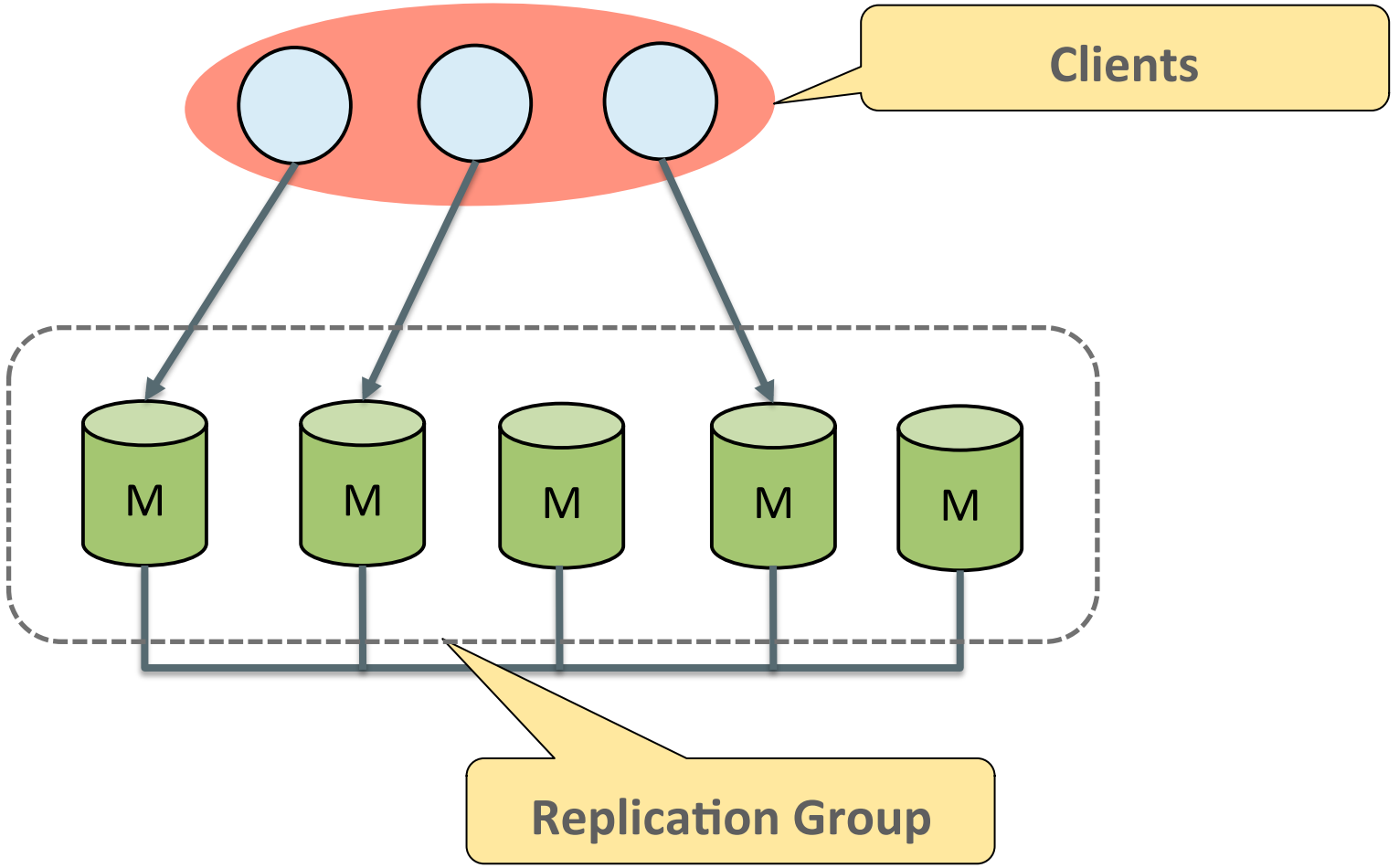
MySQL Group Replication

- **What is MySQL Group Replication?**

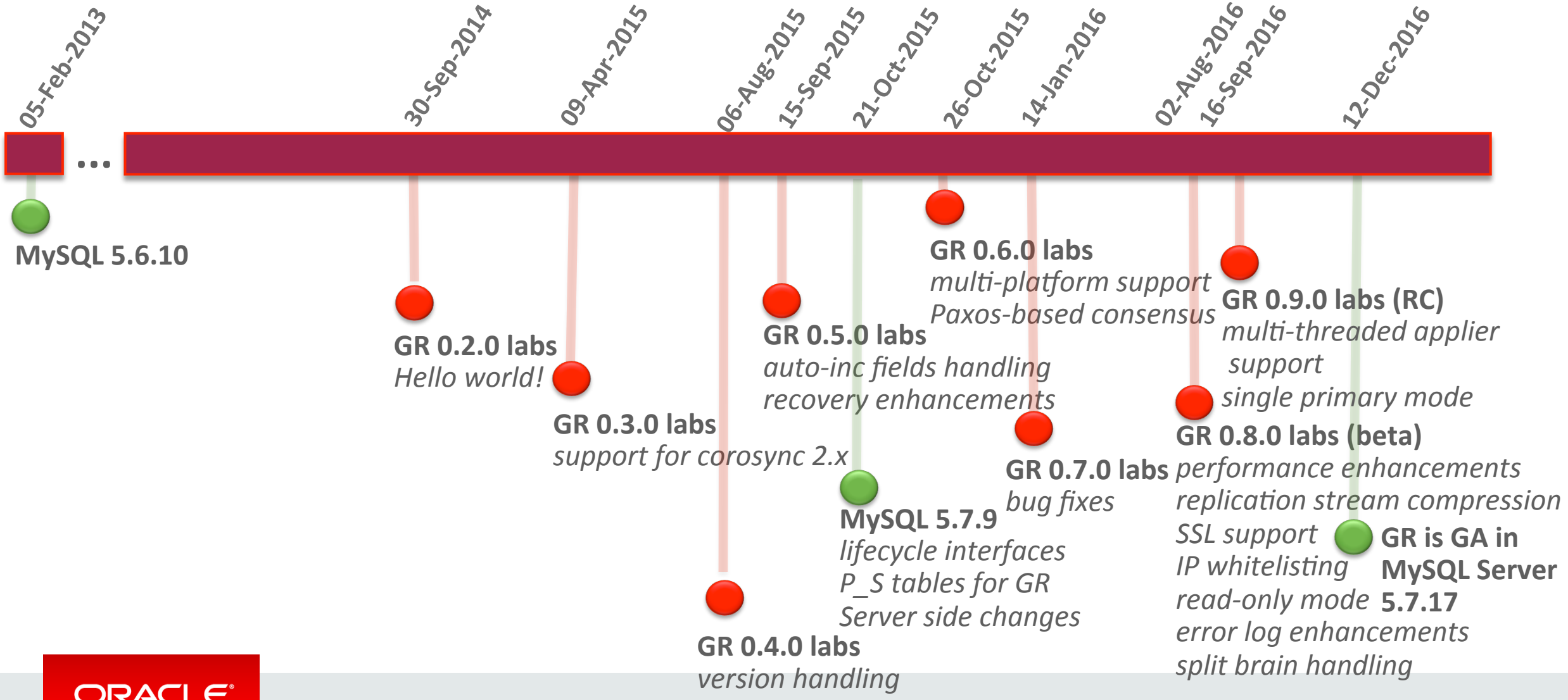
  "Multi-master **update everywhere** replication plugin for MySQL with built-in **automatic distributed recovery**, **conflict detection** and **group membership**."

- **What does the MySQL Group Replication plugin do for the user?**
  - Removes the need for handling server fail-over.
  - Provides fault tolerance.
  - Enables update everywhere setups.
  - Automates group reconfiguration (handling of crashes, failures, re-connects).
  - Provides a highly available replicated database.

MySQL Group Replication



Timeline - Server GA and GR Releases

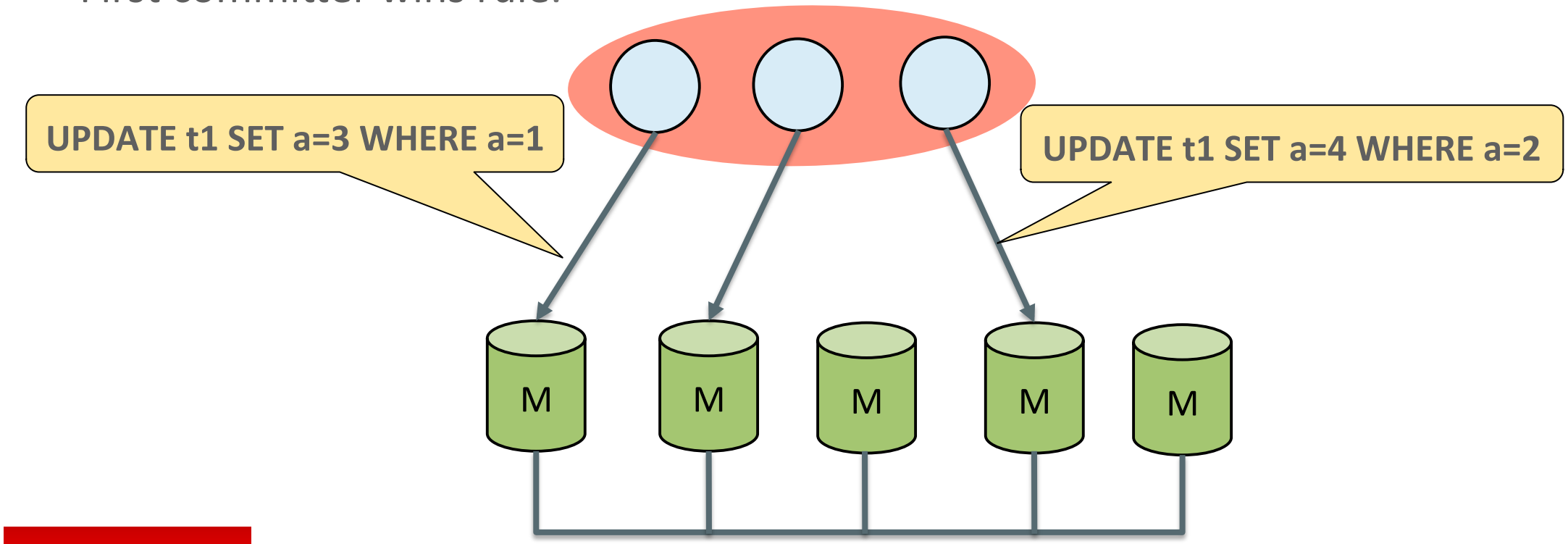


2 MySQL Group Replication

2.1 Multi-Master

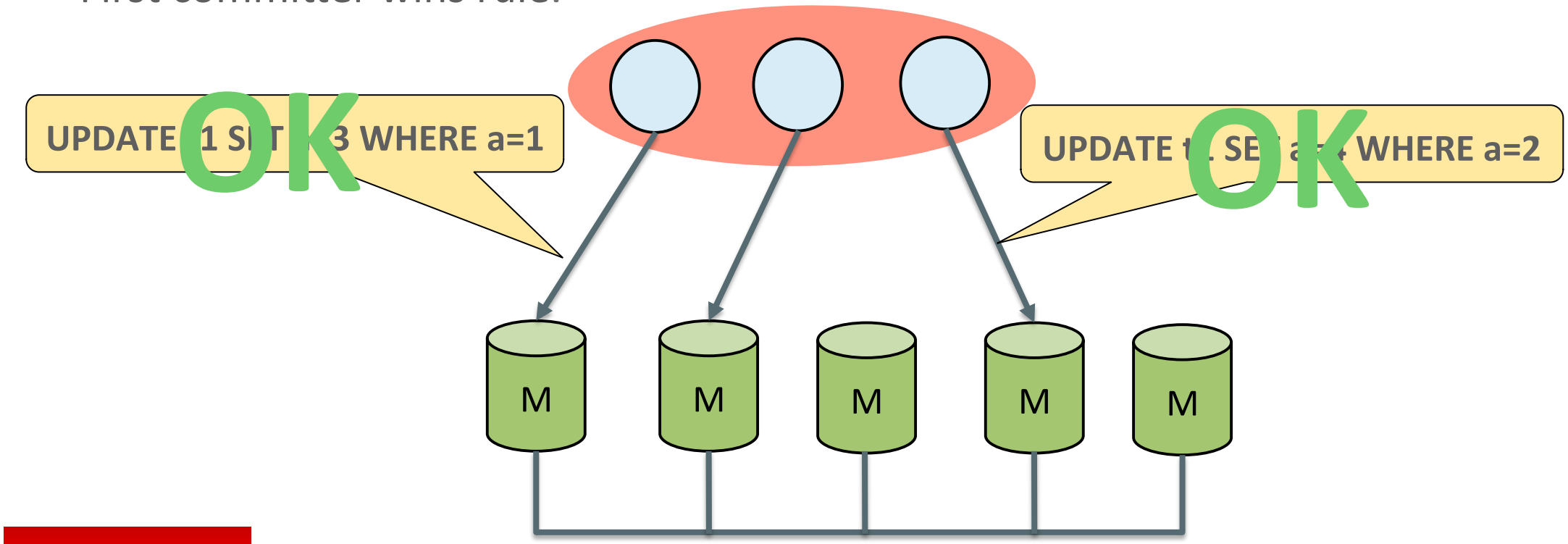
Multi-Master update everywhere!

- Any two transactions on different servers can write to the same tuple.
- Conflicts will be detected and dealt with.
  - First committer wins rule.



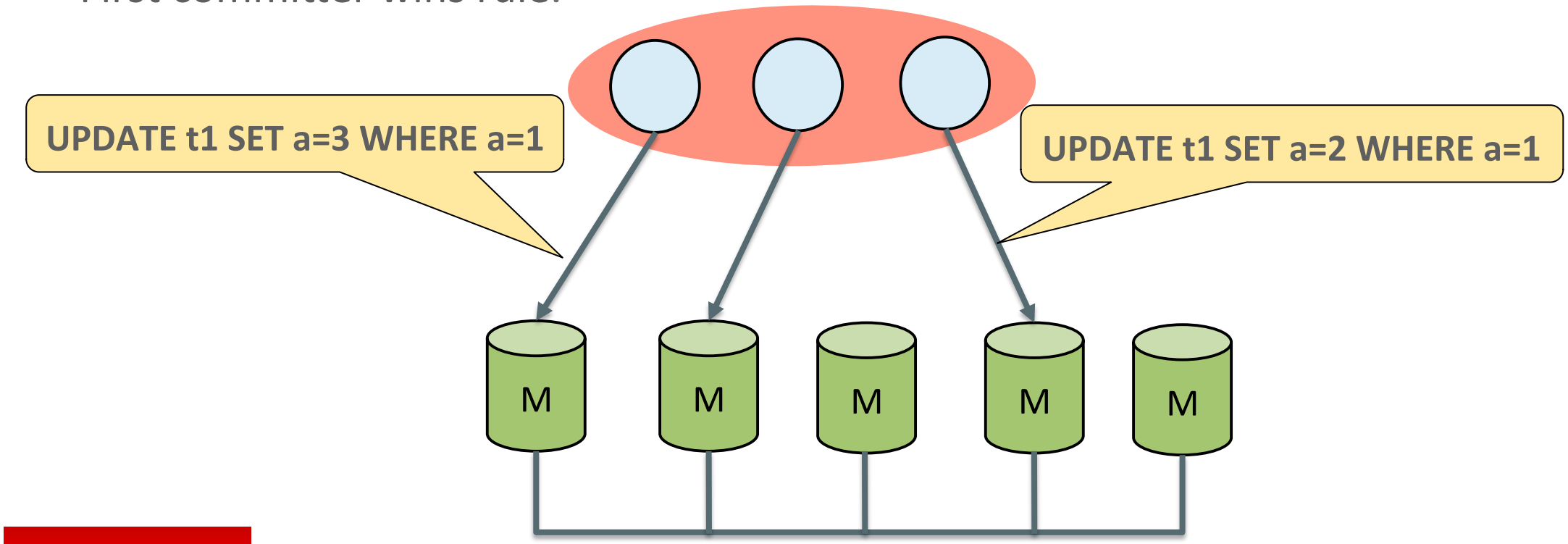
Multi-Master update everywhere!

- Any two transactions on different servers can write to the same tuple.
- Conflicts will be detected and dealt with.
  - First committer wins rule.



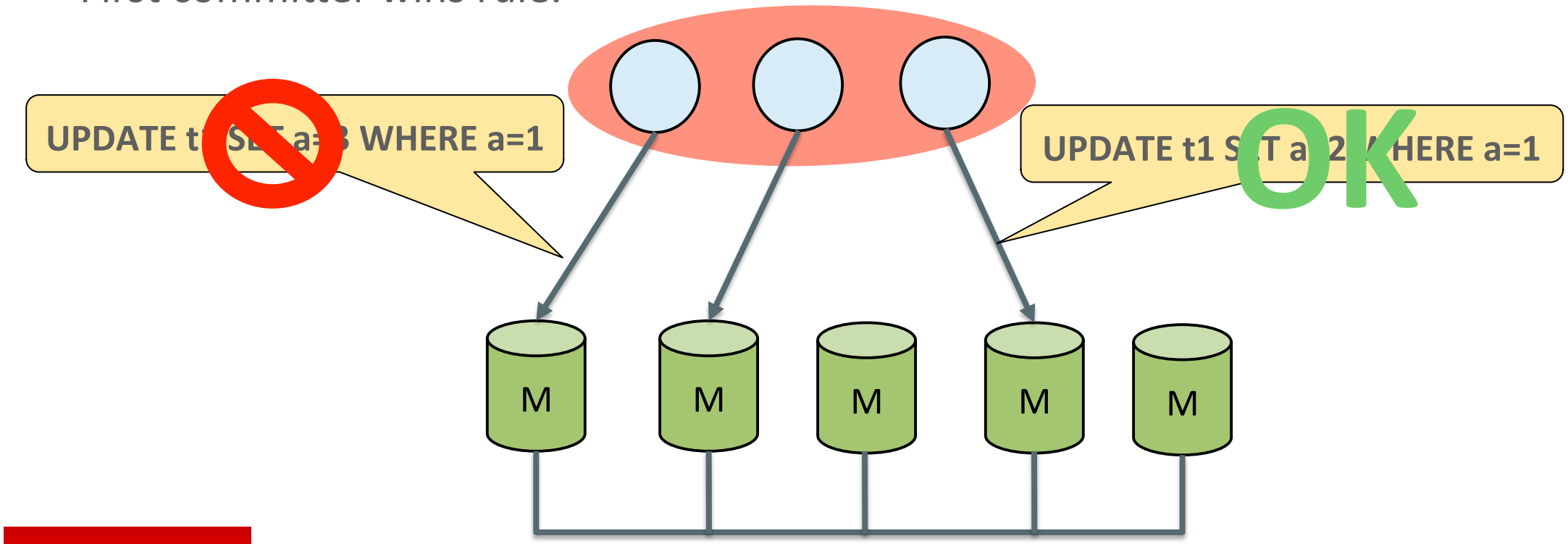
Multi-Master update everywhere!

- Any two transactions on different servers can write to the same tuple.
- Conflicts will be detected and dealt with.
  - First committer wins rule.



Multi-Master update everywhere!

- Any two transactions on different servers can write to the same tuple.
- Conflicts will be detected and dealt with.
  - First committer wins rule.



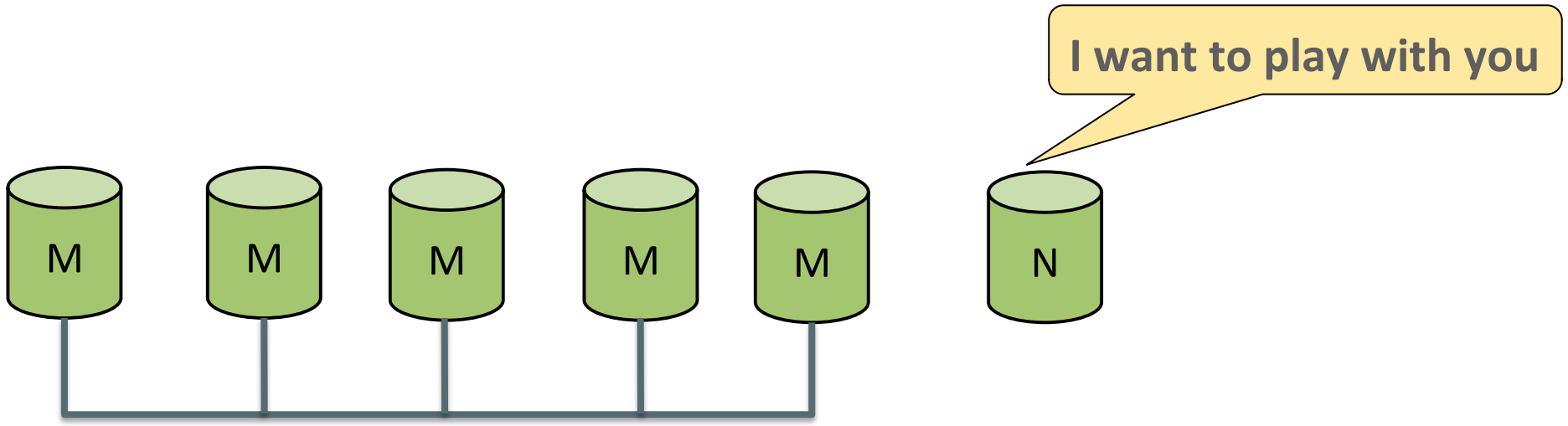
2 MySQL Group Replication

2.1 Multi-Master

2.2 Automatic distributed server recovery

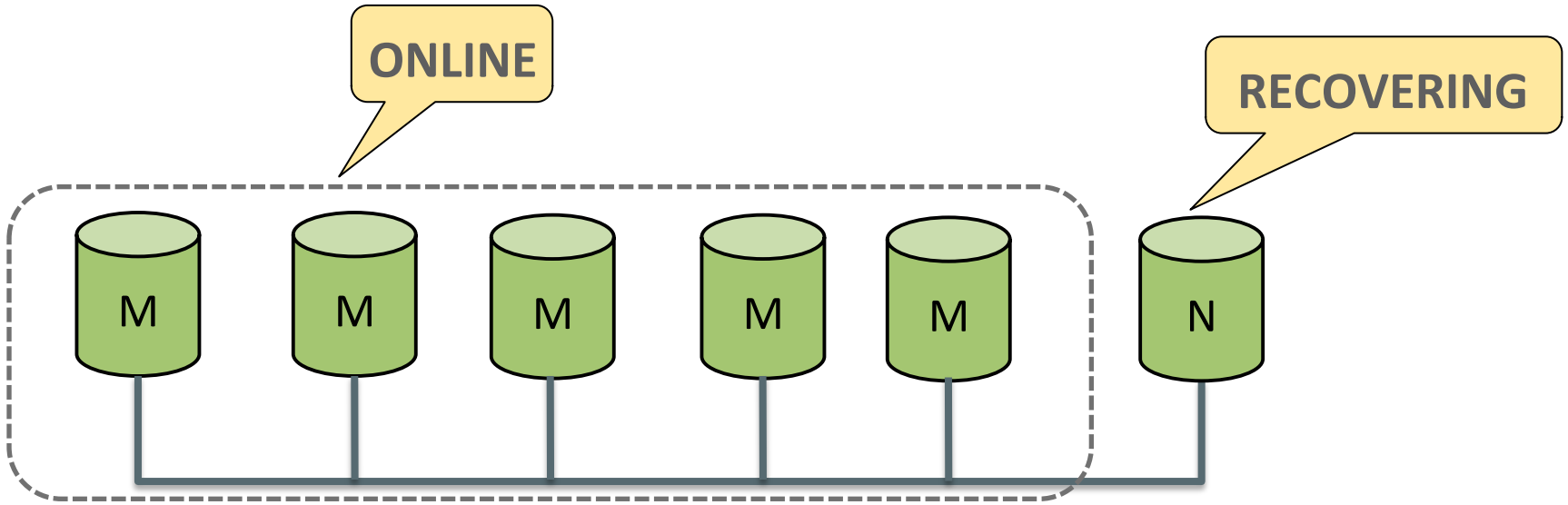
Automatic distributed server recovery!

- Server that joins the group will automatically synchronize with the others.



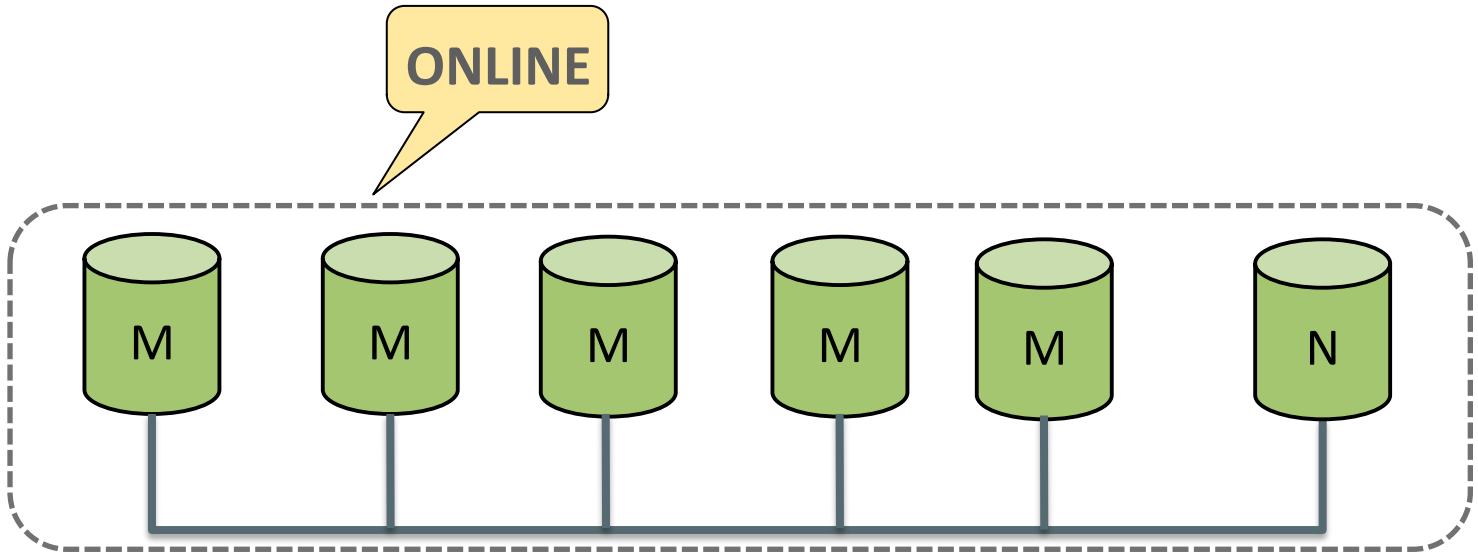
Automatic distributed server recovery!

- Server that joins the group will automatically synchronize with the others.



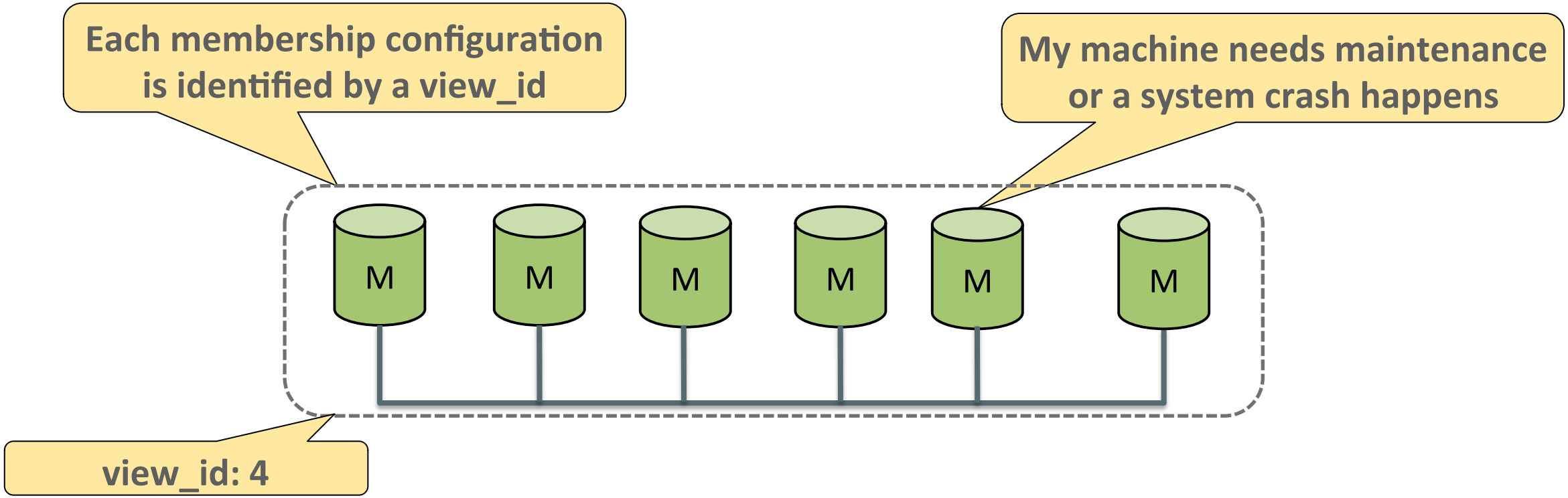
Automatic distributed server recovery!

- Server that joins the group will automatically synchronize with the others.



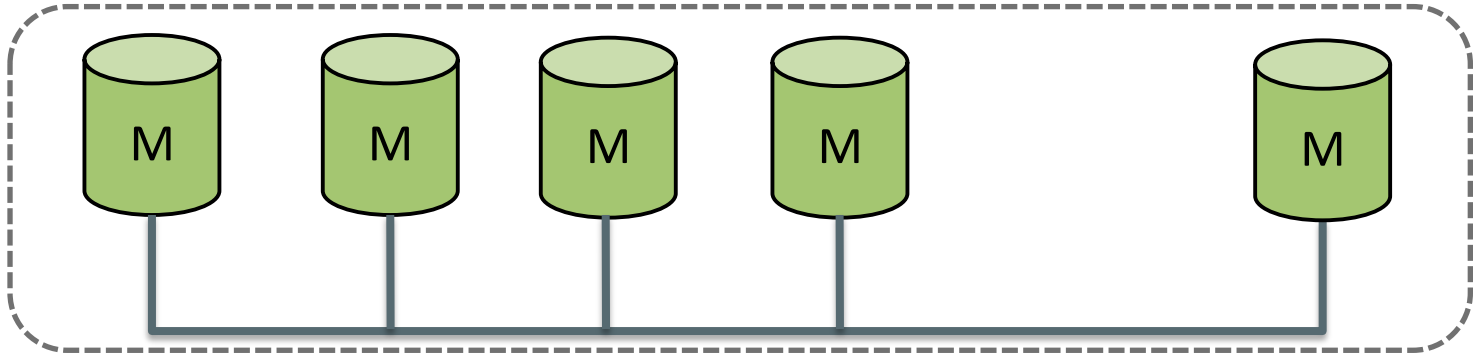
Automatic distributed server recovery!

- If a server leaves the group, the others will automatically be informed.



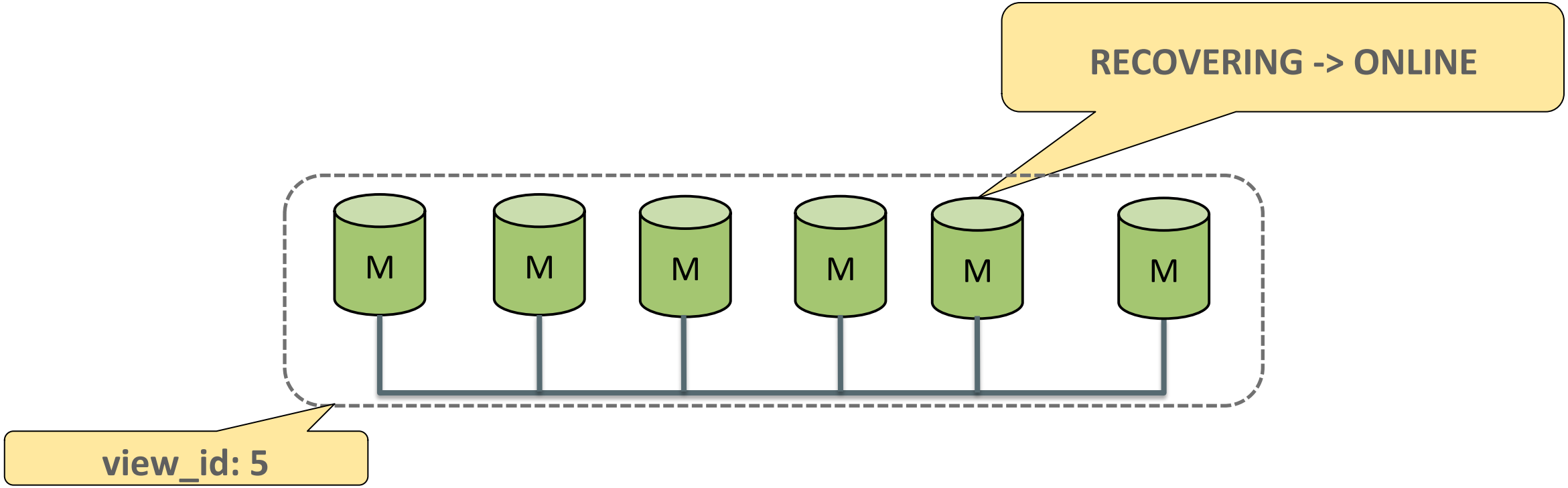
Automatic distributed server recovery!

- If a server leaves the group, the others will automatically be informed.



Automatic distributed server recovery!

- Server that (re)joins the group will automatically synchronize with the others.



2 MySQL Group Replication

2.1 Multi-Master

2.2 Automatic distributed server recovery

2.3 MySQL Look & Feel

MySQL Look & Feel!

- MySQL Plugin
  - Regular MySQL Plugin. Nothing new.
- MySQL InnoDB
  - Use InnoDB as normally you would. Nothing new.
  - Transparent optimizations in InnoDB to better support Group Replication.
- MySQL Performance Schema
  - Monitor Group Replication using regular Performance Schema tables. Nothing new.

MySQL Look & Feel!

- Outcome
  - Group Replication is no alien component.
  - Existing MySQL users feel right at home.
  - New MySQL users only have to learn MySQL tech, nothing else.

2 MySQL Group Replication

2.1 Multi-Master

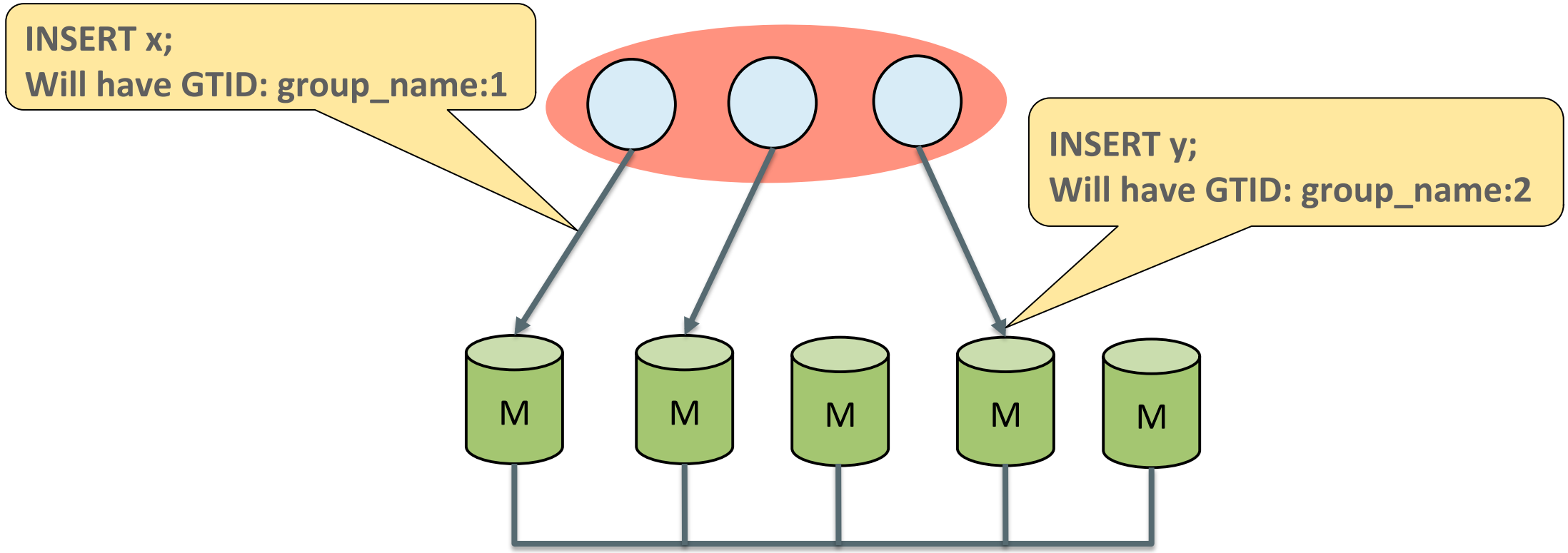
2.2 Automatic distributed server recovery

2.3 MySQL Look & Feel

2.4 Full GTID support

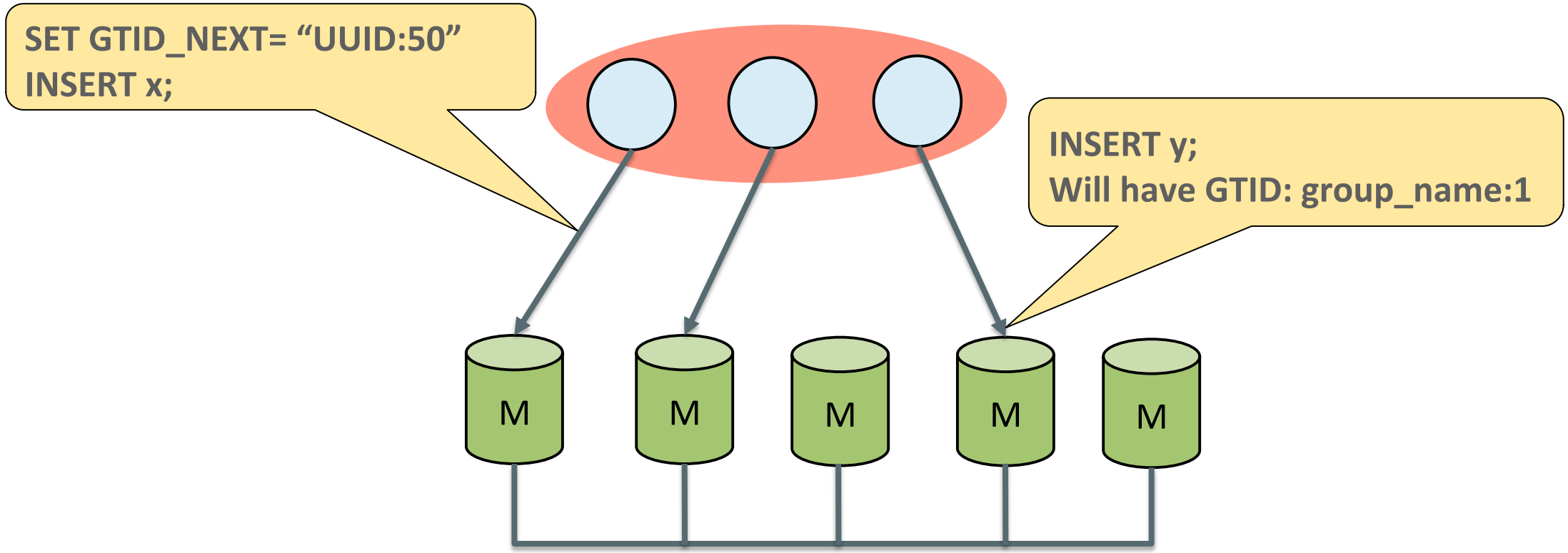
Full GTID support!

- All group members share the same UUID, the group name.



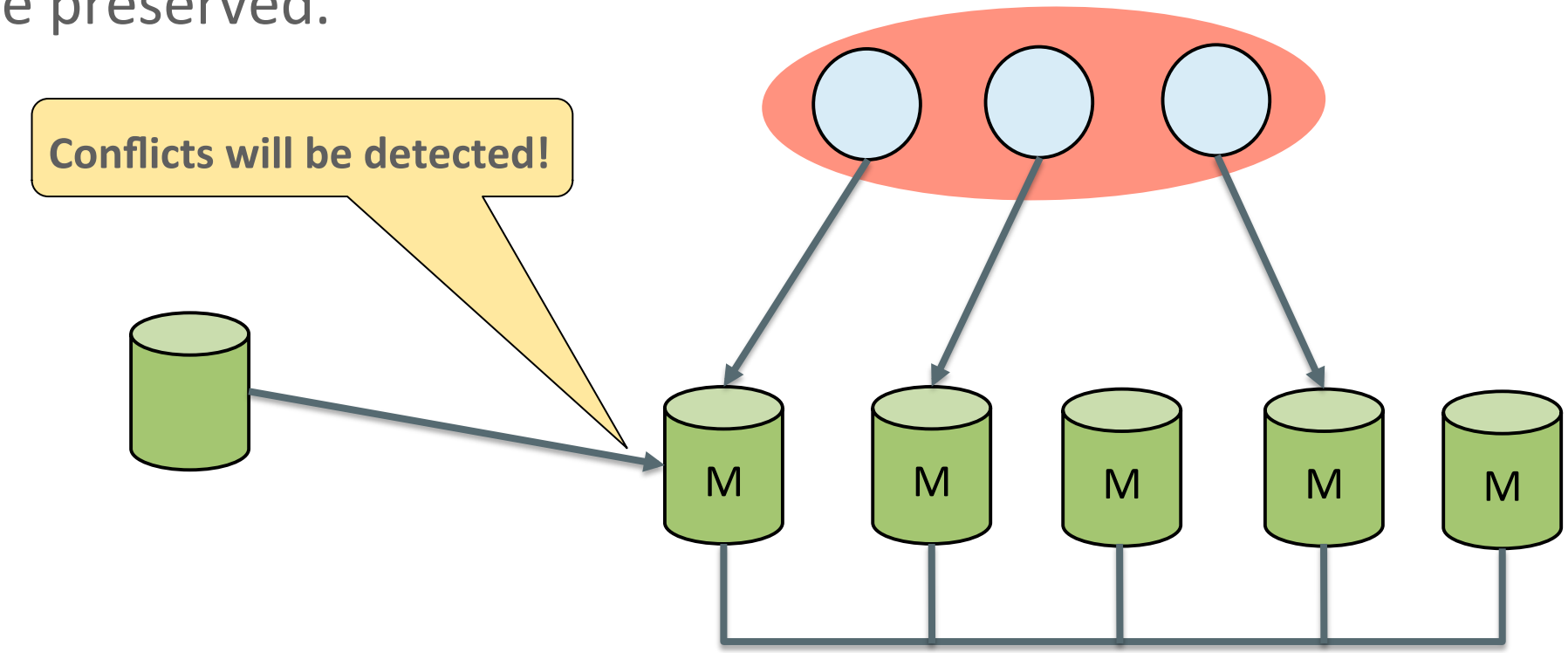
Full GTID support!

- Users can specify the identifier for the transaction.



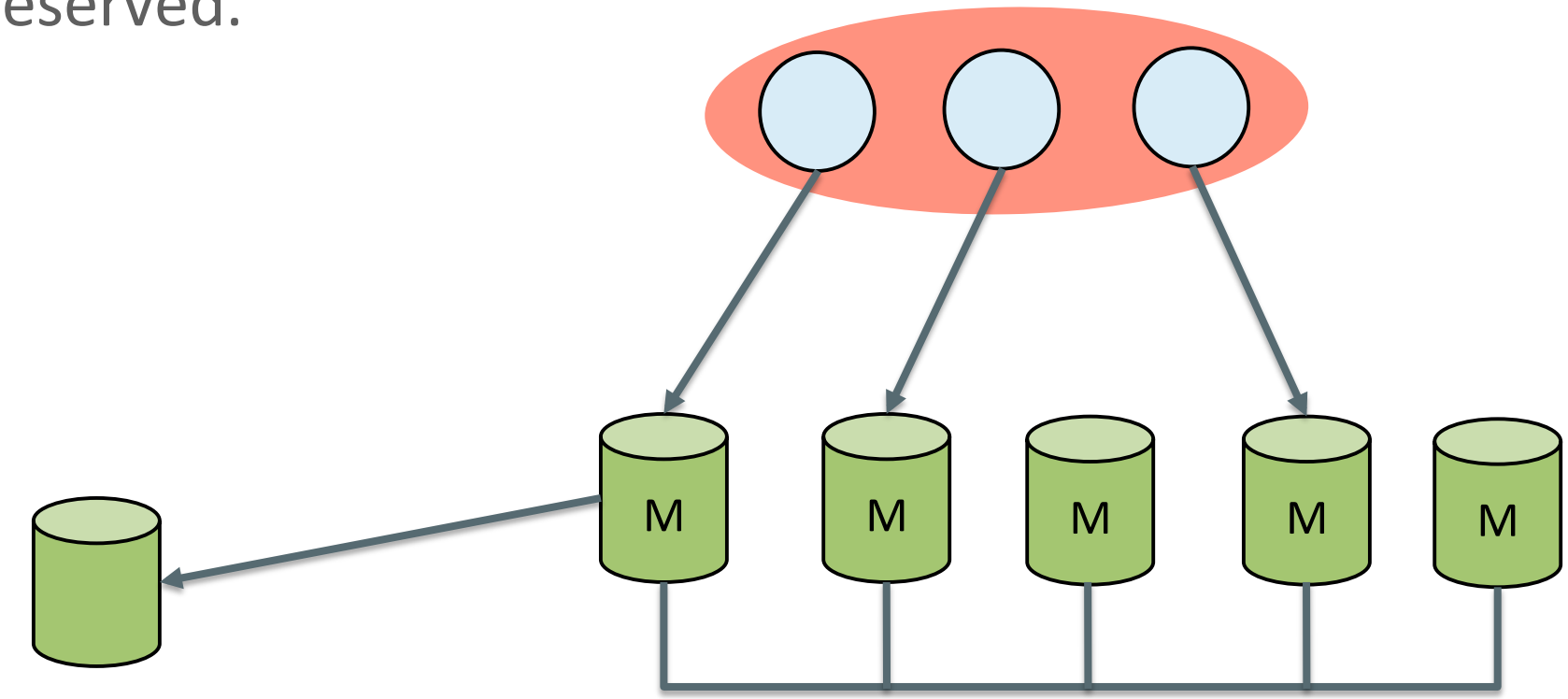
Full GTID support!

- You can even replicate from a outside server to a group, global identifiers will be preserved.



Full GTID support!

- You can also replicate from a group to a outside server, global identifiers will be preserved.

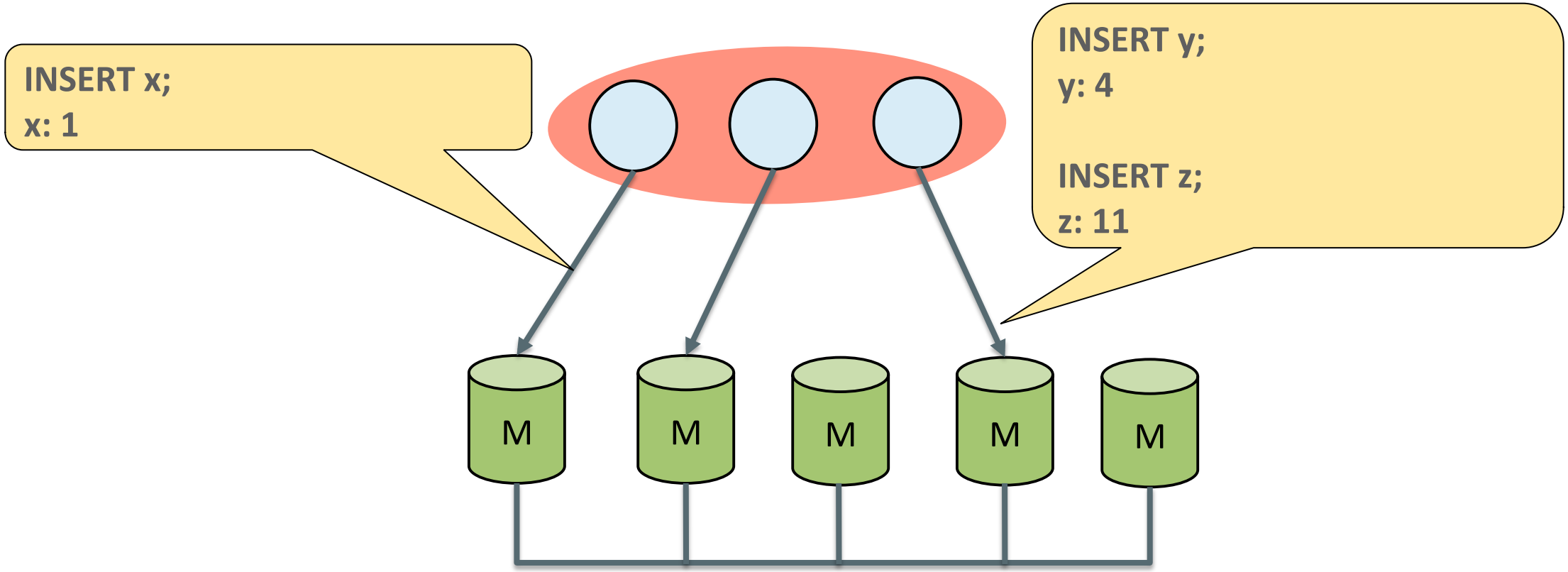


2 MySQL Group Replication

- 2.1 Multi-Master
- 2.2 Automatic distributed server recovery
- 2.3 MySQL Look & Feel
- 2.4 Full GTID support
- 2.5 Auto-increment configuration/handling

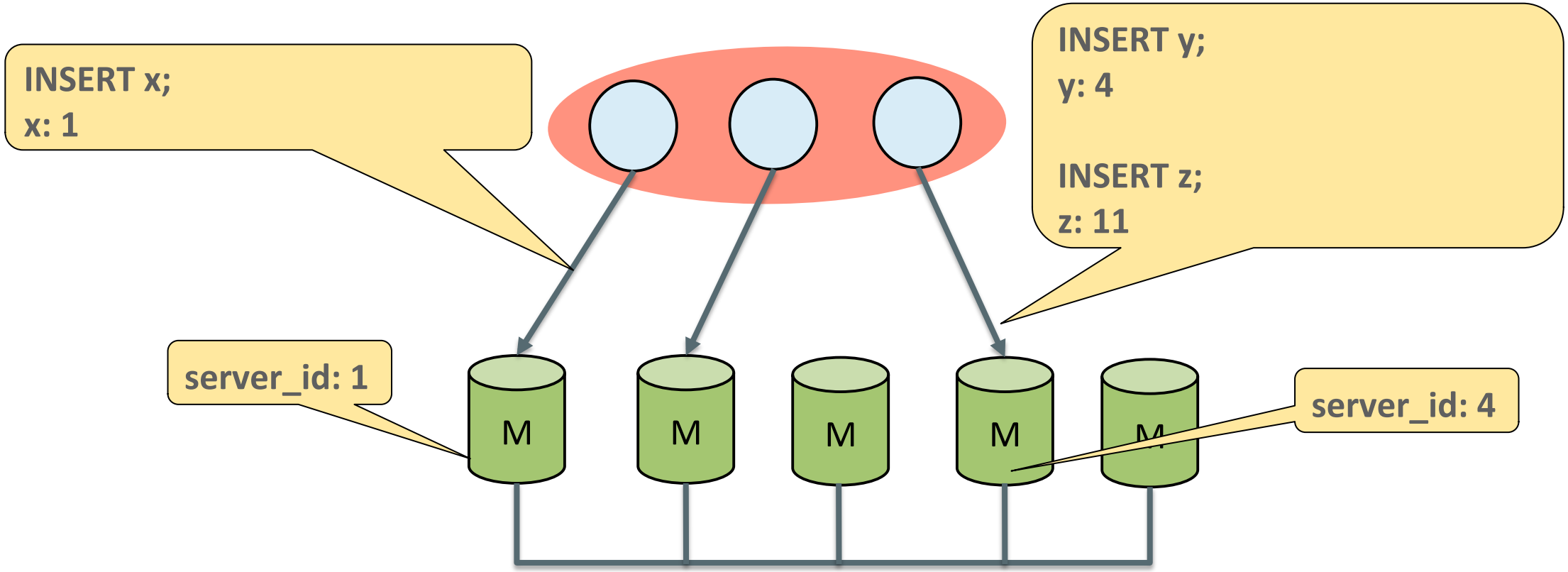
Auto-increment configuration/handling

- Group is configured to not generate the same auto-increment value on all members.



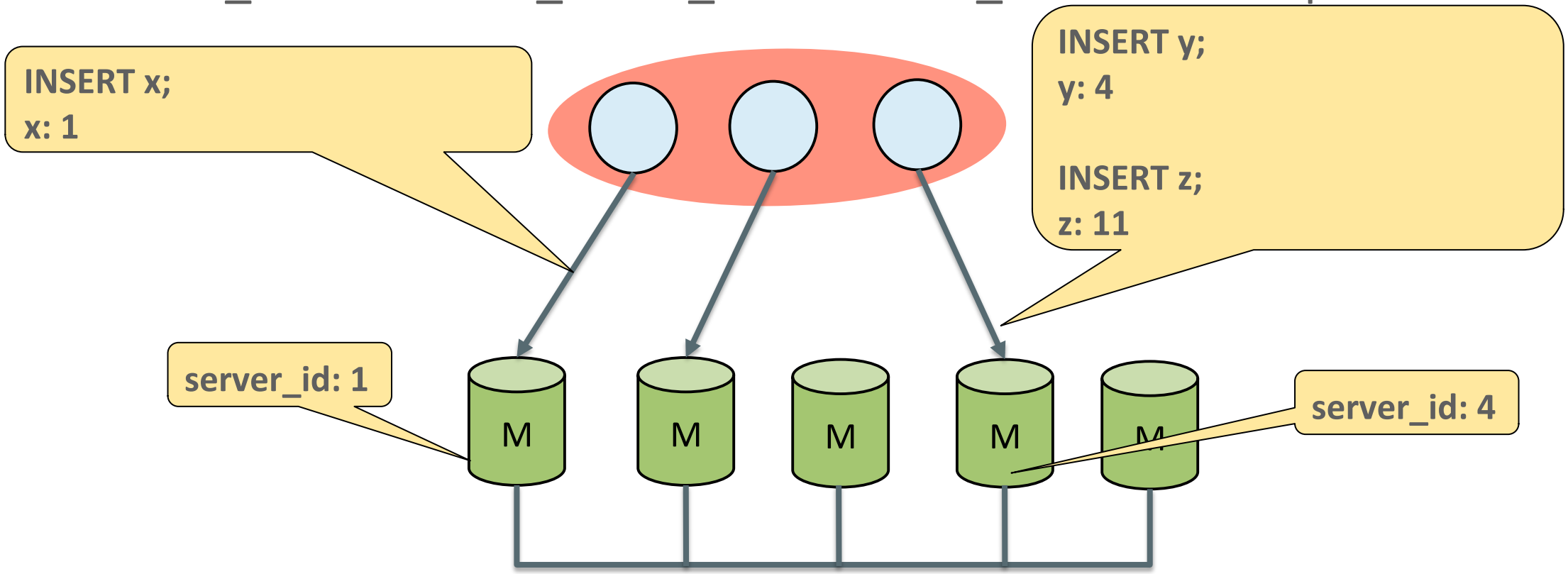
Auto-increment configuration/handling

- By default, the offset is provided by server_id and increment is 7.



Auto-increment configuration/handling

- Users can change the increment size to their needs using GROUP_REPLICATION_AUTO_INCREMENT_INCREMENT option.

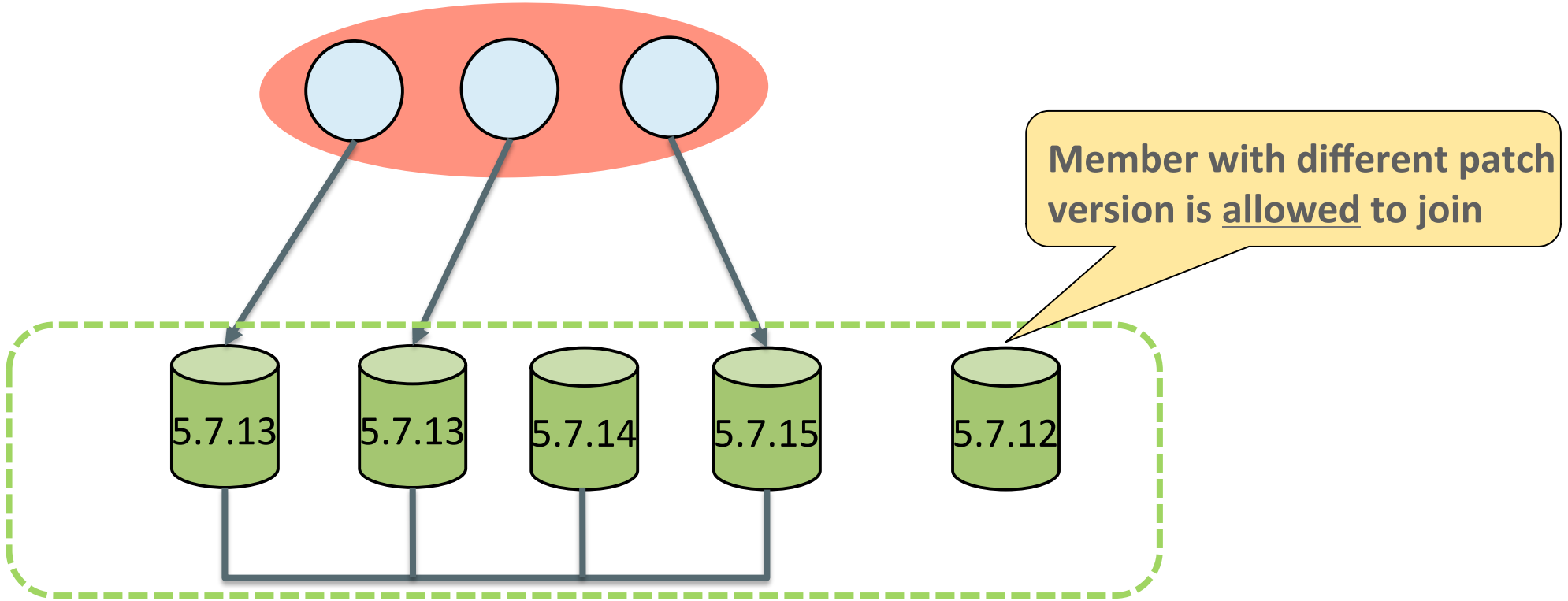


2 MySQL Group Replication

- 2.1 Multi-Master
- 2.2 Automatic distributed server recovery
- 2.3 MySQL Look & Feel
- 2.4 Full GTID support
- 2.5 Auto-increment configuration/handling
- 2.6 Plugin version access control

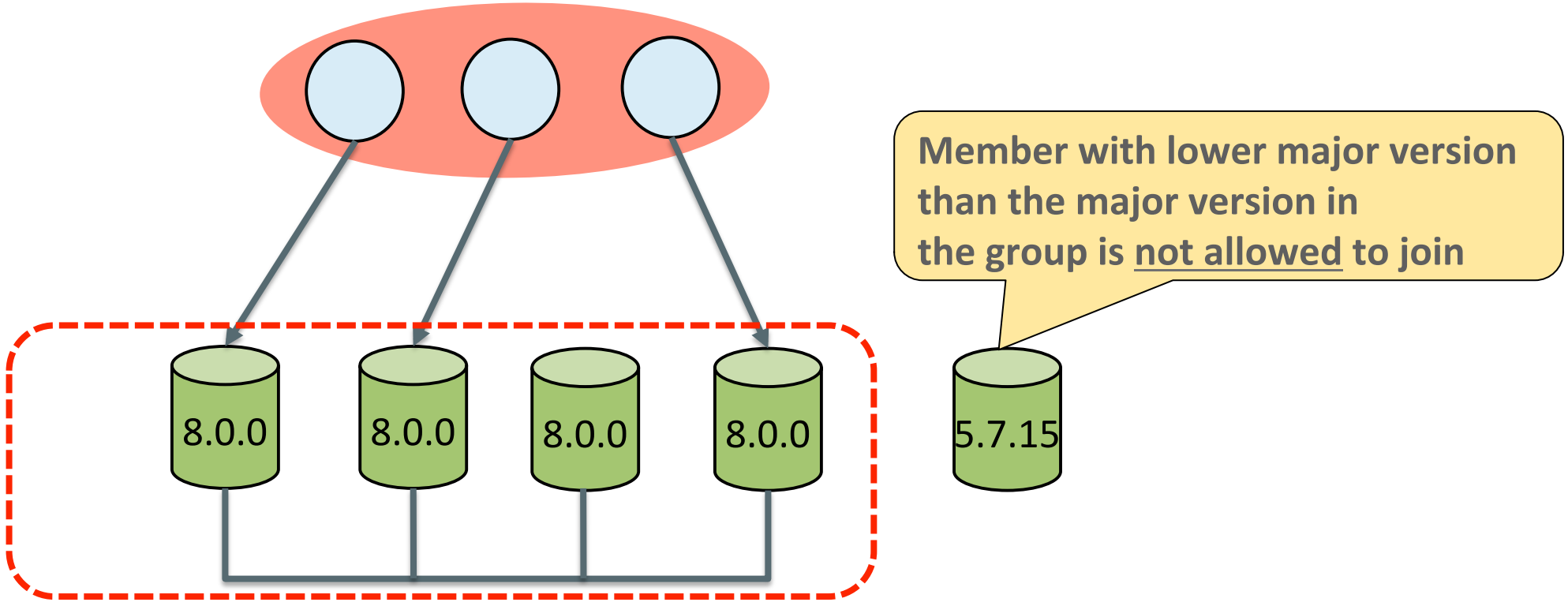
Plugin Version Access Control

- When joining, versions are crucial when determining if a member is compatible with a group.



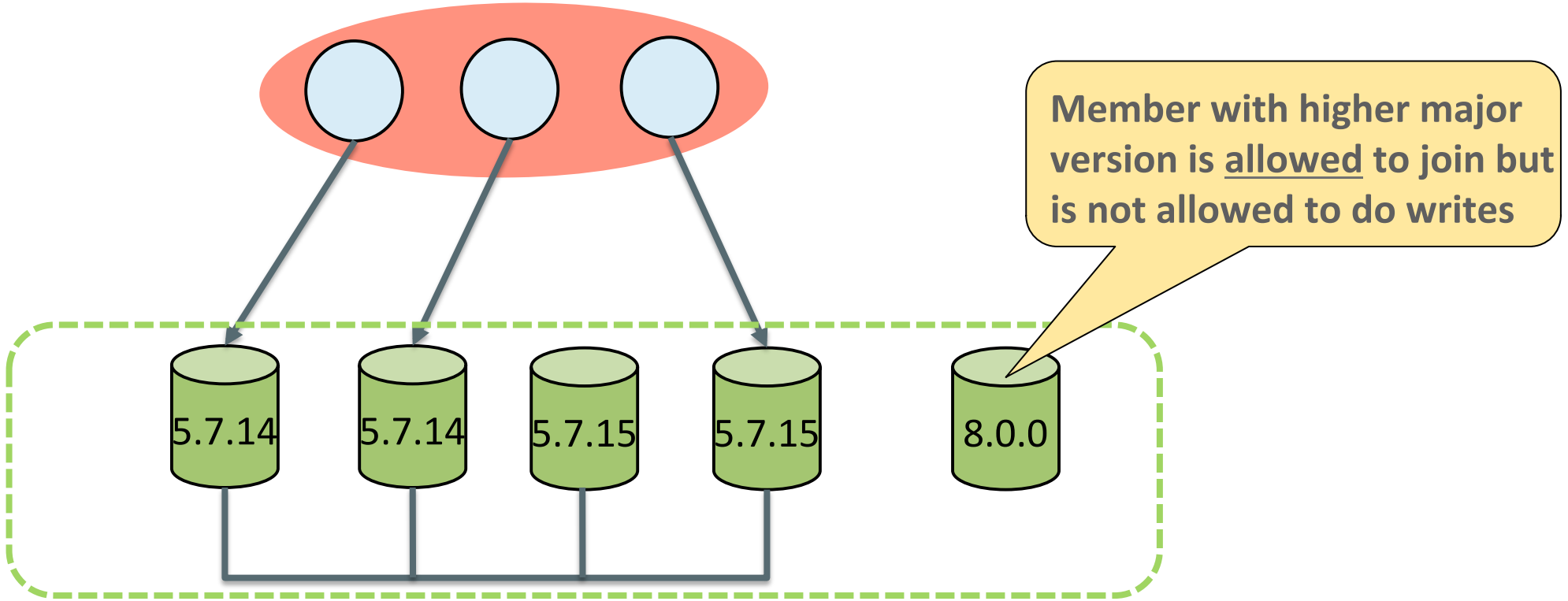
Plugin Version Access Control

- When joining, versions are crucial when determining if a member is compatible with a group.



Plugin Version Access Control

- When joining, versions are crucial when determining if a member is compatible with a group.



2 MySQL Group Replication

- 2.1 Multi-Master
- 2.2 Automatic distributed server recovery
- 2.3 MySQL Look & Feel
- 2.4 Full GTID support
- 2.5 Auto-increment configuration/handling
- 2.6 Plugin version access control
- 2.7 Built-in communication engine

Built-in Communication Engine

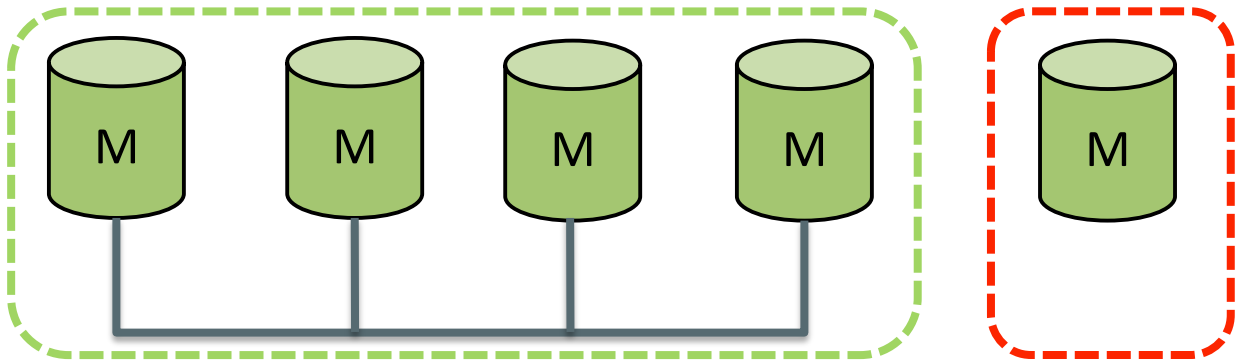
- Feature rich new replication plugin based on proven distributed systems algorithms (Paxos).
  - Compression, multi-platform, dynamic membership, distributed agreement, quorum based message passing, SSL, IP whitelisting.
- No third-party software required.
- No network multicast support required.
  - MySQL Group Replication can operate on cloud based installations where multicast is unsupported.

2 MySQL Group Replication

2.8 Read-only mode

Read-only mode

- When a member joins the group, during distributed recovery, read-only mode is automatically set.
- On the unlikely event of a member failure, read-only mode is set automatically to prevent inconsistency with the group and member state changes to ERROR.



2 MySQL Group Replication

2.8 Read-only mode

2.9 Full stack secure connections

Full stack secure connections

- Group Replication supports secure connections along the complete stack:
  - Distributed recovery connections
  - Connections between members
  - Client connections
- IP Whitelisting
  - Restrict which hosts are allowed to connect to the group
  - By default it is set to the value AUTOMATIC, which allows connections from private subnetworks active on the host

2 MySQL Group Replication

2.8 Read-only mode

2.9 Full stack secure connections

2.10 Parallel applier support

Parallel applier support

- Reduces applier lag and improves replication performance considerably.
- The same configuration options as asynchronous replication.

```
--slave_parallel_workers=NUMBER
--slave_parallel_type=logical_clock
--slave_preserve_commit_order=ON
```

2 MySQL Group Replication

2.8 Read-only mode

2.9 Full stack secure connections

2.10 Parallel applier support

2.11 Single Primary Mode

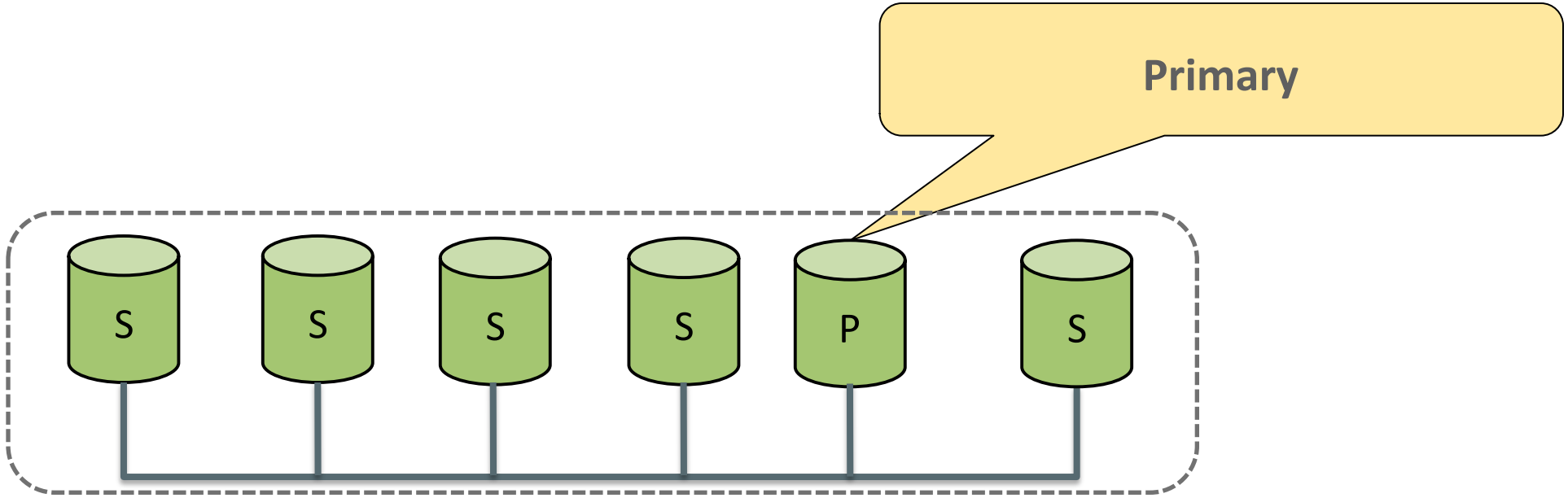
Single Primary Mode

- Configuration mode that makes a single member act as a writeable master (PRIMARY) and the rest of the members act as hot-standbys (SECONDARIES).
  - The group itself coordinates automatically to figure out which is the member that will act as the PRIMARY, through a primary election mechanism.

- Single Primary Mode is the Default mode
  - Closer to classic asynchronous replication setups, simpler to reason about from the beginning.
  - Avoids some of the limitations of multi-master mode by default.

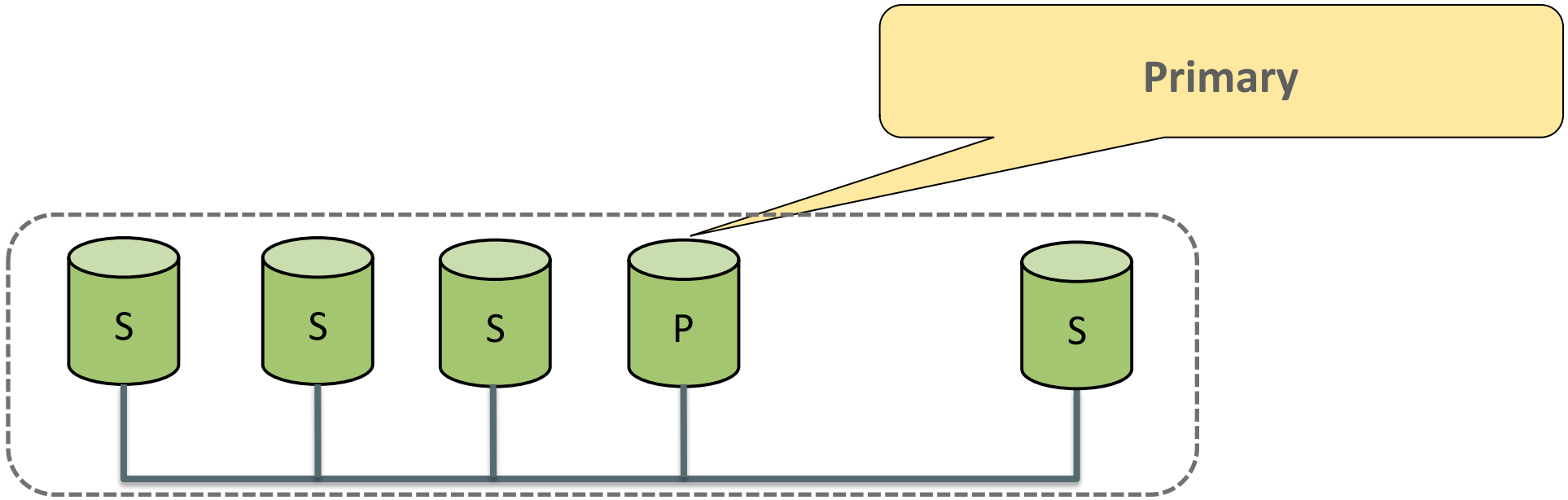
Single Primary Mode

- Automatic primary promotion election.
- Secundaries are automatically set to read-only.



Single Primary Mode

- Automatic primary election mechanism.



Single Primary Mode

- The current primary member UUID can be known by executing the following SQL statement.

```
mysql> SELECT * FROM performance_schema.global_status WHERE
VARIABLE_NAME='group_replication_primary_member';
VARIABLE_NAME                       VARIABLE_VALUE
group_replication_primary_member  dcd3b36b-79c5-11e6-97b8-00212844d44e
```

2 MySQL Group Replication

2.8 Read-only mode

2.9 Full stack secure connections

2.10 Parallel applier support

2.11 Single Primary Mode

2.12 Requirements

Requirements (by design)

- Requires InnoDB storage engine
- Primary key is required on every table
- Requires global transaction identifiers turned on
- Requires binary log turned on
- Requires binary log row format
- Optimistic execution: transactions may abort on COMMIT due to conflicts with concurrent transactions on other members
- Up to 9 servers in the group

Forbidden

- Serializable (on multi-master)
- Cascading Foreign Keys (on multi-master)
- Transaction savepoints
- Binary log events checksum

Warnings

- Concurrent DDL (on multi-master)

3 Architecture

3.1 Introduction

MySQL Group Replication is

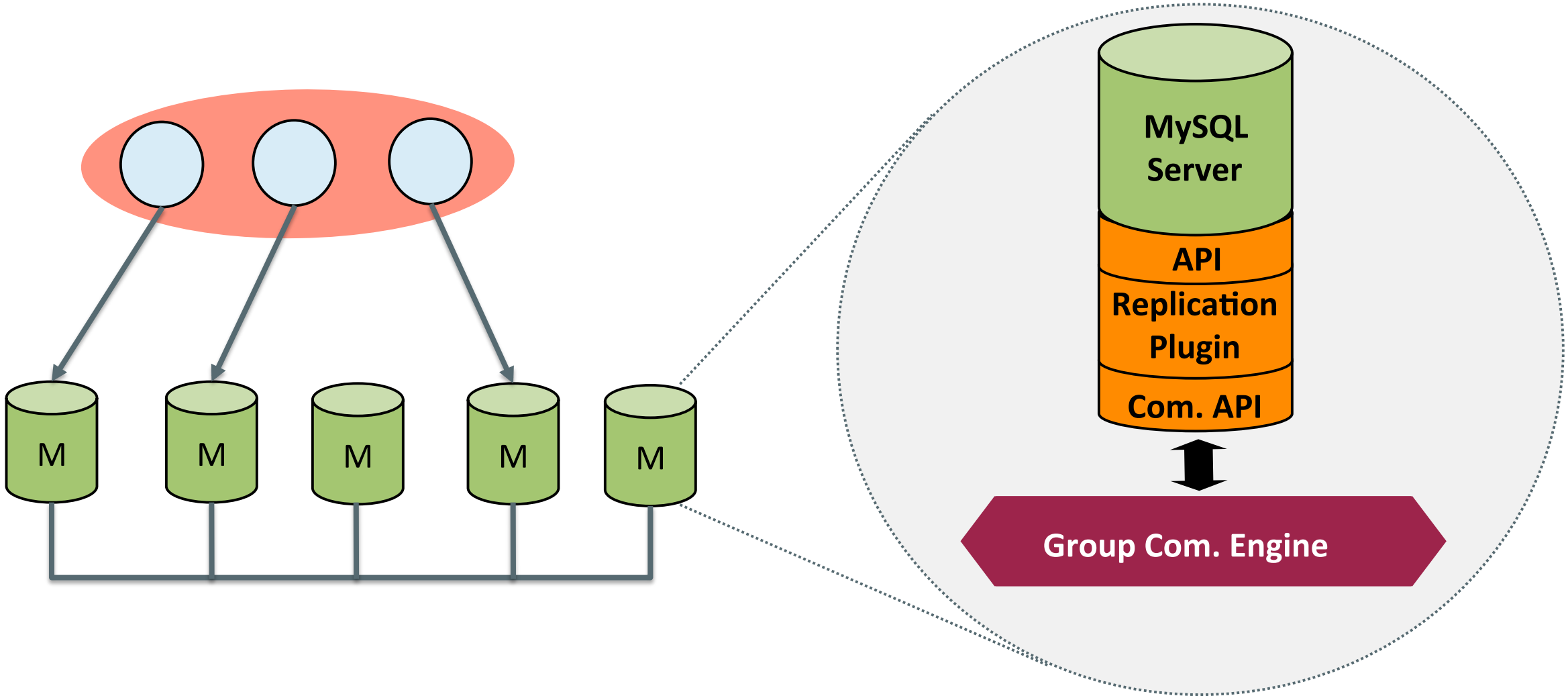
- **Built on top of proven technology!**
  - Shares many pieces of MySQL Replication.
  - Multi-Master approach to replication.
- **Built on reusable components!**
  - Layered implementation approach.
  - Interface driven development.
  - Decoupled from the server core.
  - The plugin registers as listener to server events.
  - Reuses the capture procedure from regular replication.
  - Provides further decoupling from the communication infrasctructure.

3 Architecture

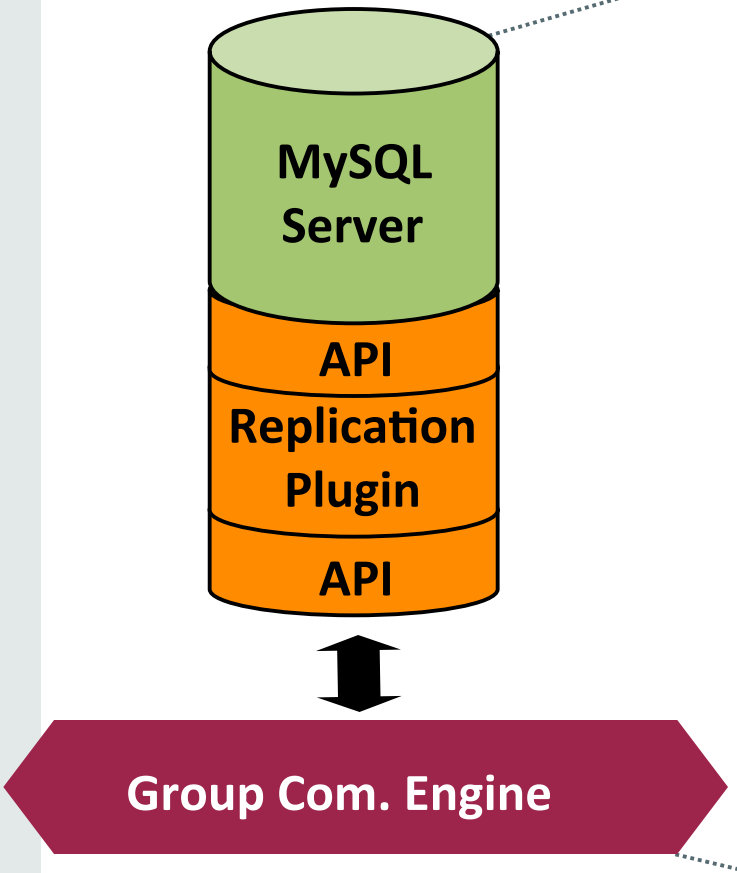
3.1 Introduction

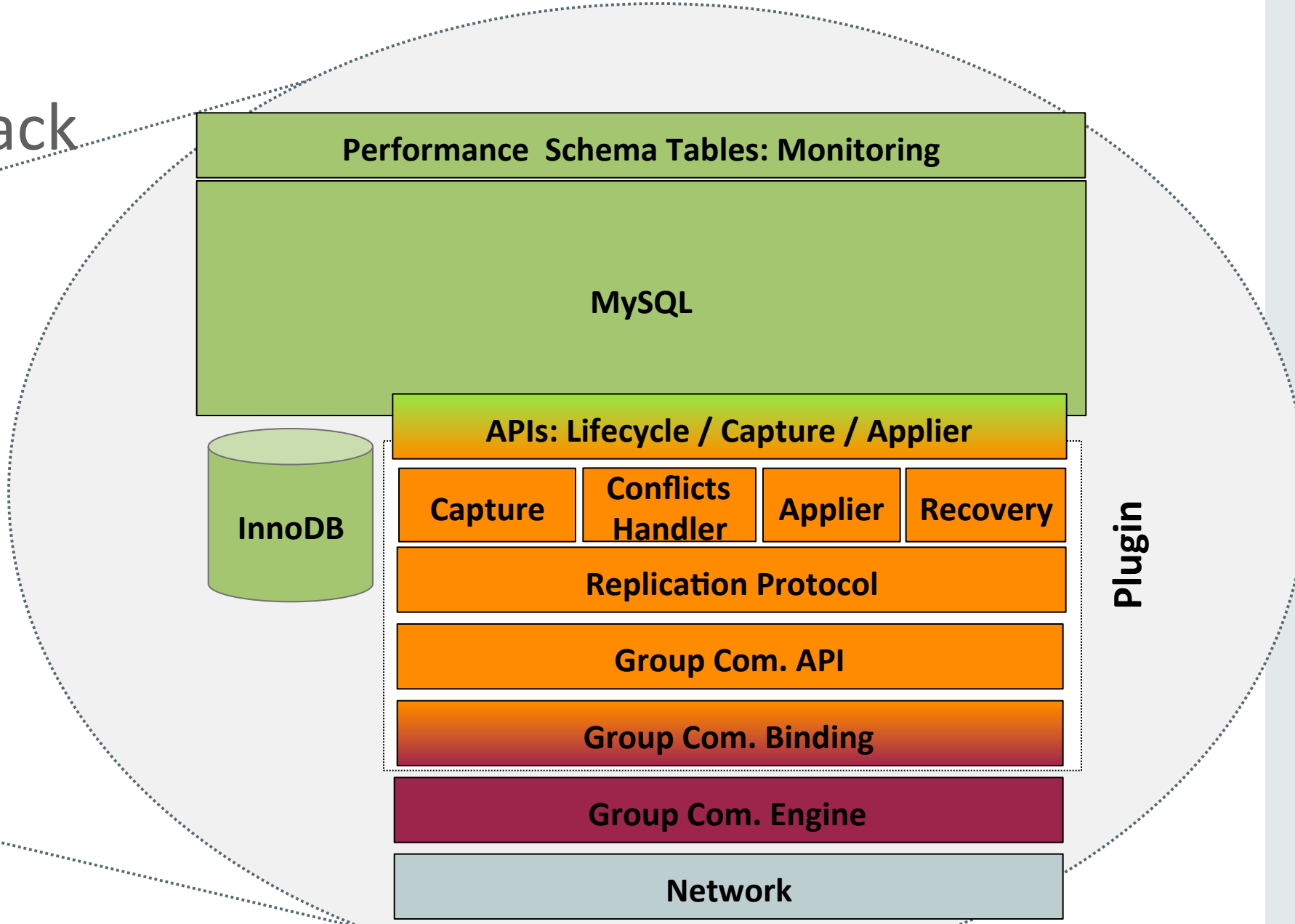
3.2 Major Building Blocks

Major Building Blocks



The Complete Stack

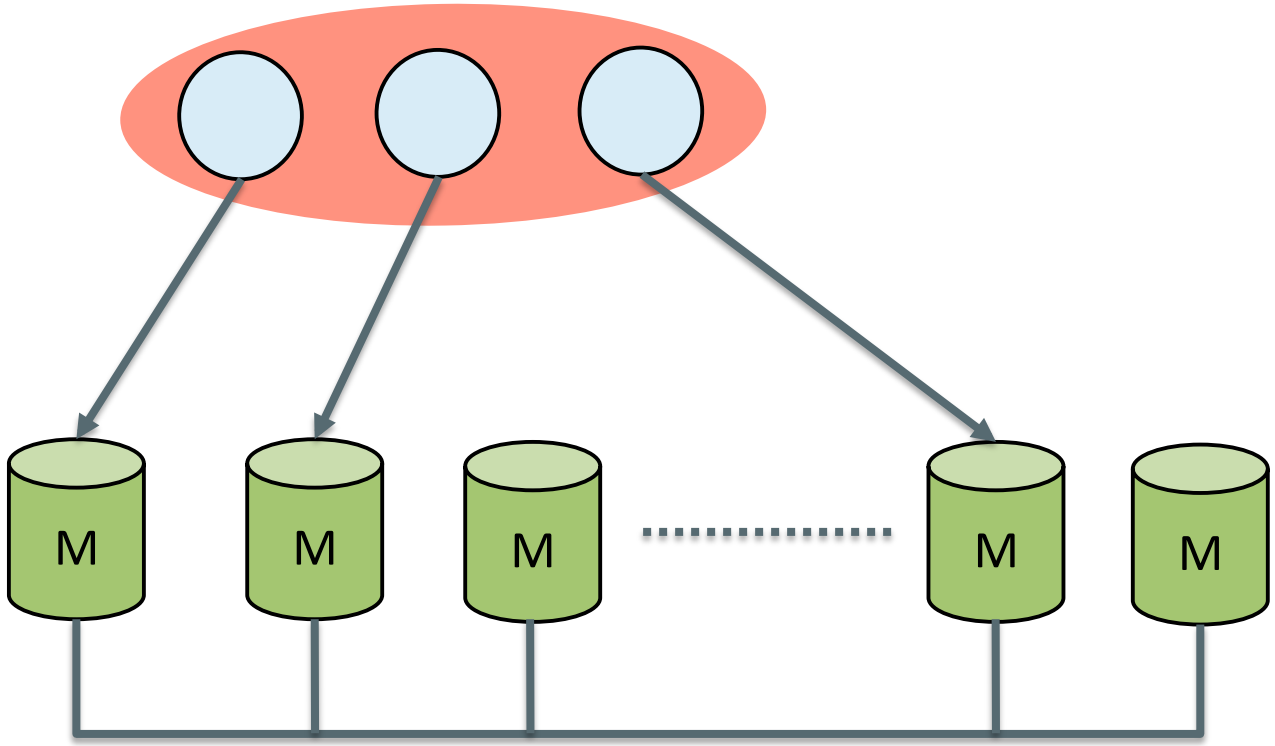




4 Use cases

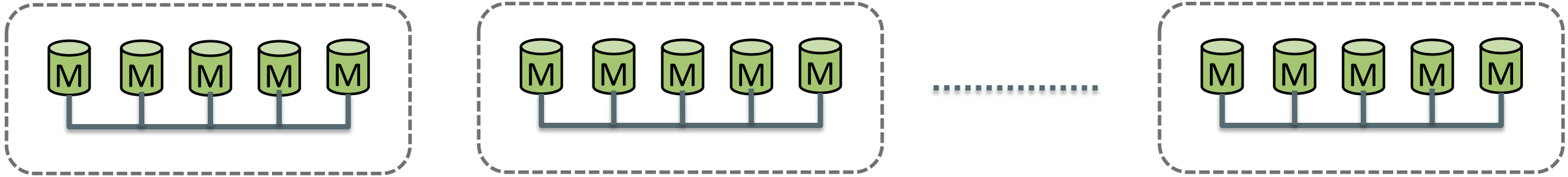
Use Cases

- **Elastic Replication**
  - Environments that require a very fluid replication infrastructure, where the number of servers has to grow or shrink dynamically and with as little pain as possible.



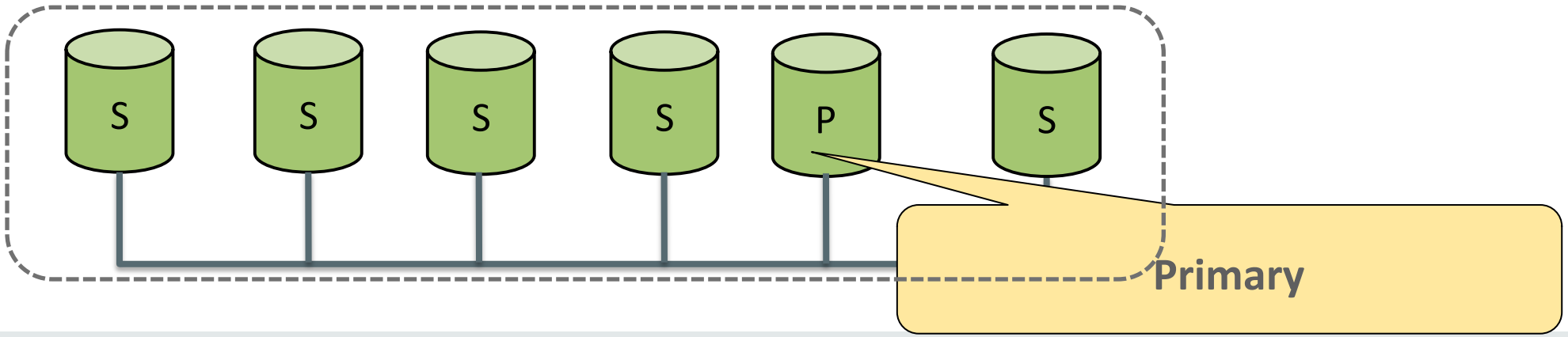
Use Cases

- **Highly Available Shards**
  - Sharding is a popular approach to achieve write scale-out. Users can use MySQL Group Replication to implement highly available shards. Each shard can map into a Replication Group.



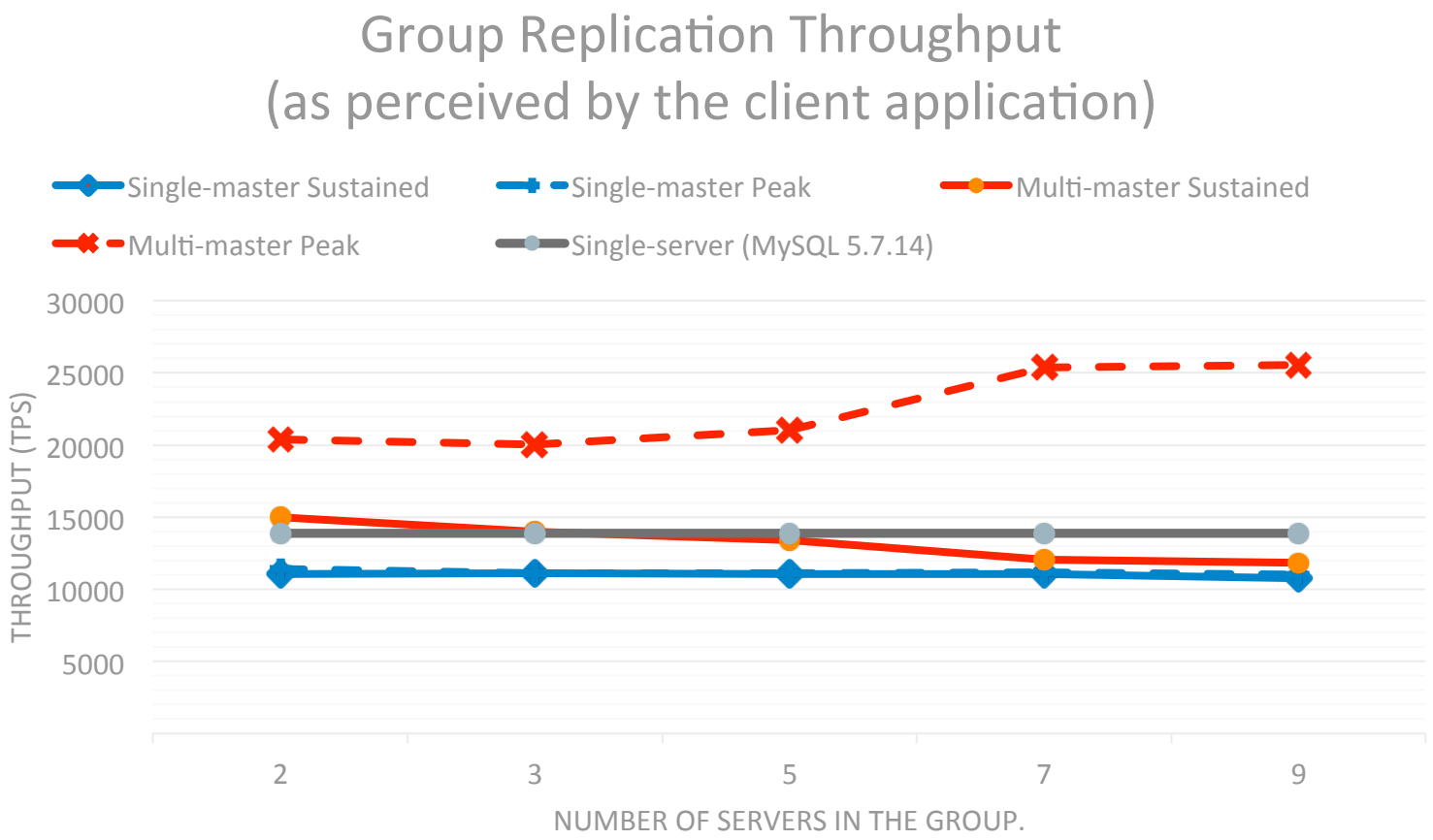
Use Cases

- **Alternative to Master-Slave replication**
- **Single Primary mode provides further automation on such setups**
  - Automatic PRIMARY/SECONDARY roles assignment
  - Automatic new PRIMARY election on PRIMARY failures
  - Automatic setup of read/write modes on PRIMARY and SECONDARIES
  - Global consistent view of which server is the PRIMARY



5 Performance

Performance



Peak Throughput (i.e., no flow control)

The number of transactions that writers can propagate to the group (per second).

Sustained Throughput (i.e., flow control)

The number of transactions that can be propagated to the group without increasing the replication lag on any member (per second).

Servers

9 Dual Xeon E5-2660-v3
Enterprise SSD Storage
10Gbps Ethernet Network

Client

1 Dual Xeon E5-2699-v3
10Gbps Ethernet Network
Sysbench 0.5 RW workload

More on this subject on the series of replication performance blogs at:
http://mysqlhighavailability.com/category/performance/

Performance

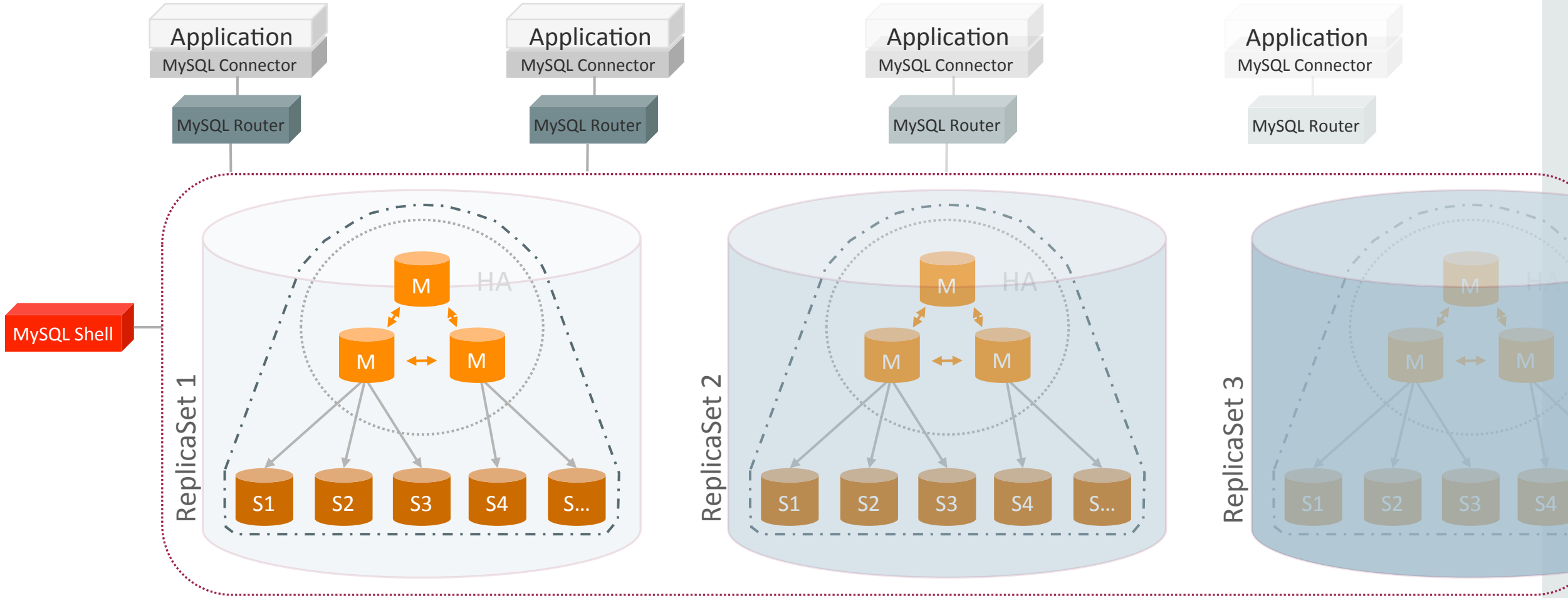
- On a sustained throughput:
  - Multi-master performance **degrades gracefully** while going from a group with 2 servers to a group with 9 servers.
  - Single primary performance **degrades marginally** when growing the group size.
- On a peak throughput:
  - Multi-master exhibits **1.8X speedup** when compared to the single server.
    - Read load is balanced across the servers in the group.
    - Write load is lower since execution is balanced across the group, whereas in single primary mode the primary becomes a bottleneck.
  - With a single primary there **is no lag** on the other members.

6 Conclusion

Summary

- **Cloud Friendly**
  - Great technology for deployments where elasticity is a requirement, such as cloud based infrastructures.
- **Integrated**
  - With server core through a well defined API.
  - With GTIDs, row based replication, performance schema tables.
- **Autonomic and Operations Friendly**
  - It is self-healing: no admin overhead for handling server fail-overs.
  - Provides fault-tolerance, enables multi-master update everywhere and a dependable MySQL service.
- Plugin **GA version** available with MySQL 5.7.17.

MySQL InnoDB Cluster: The End Goal



Where to go from here?

- Packages
  - http://www.mysql.com/downloads/
- Documentation
  - http://dev.mysql.com/doc/refman/5.7/en/group-replication.html
- Blogs from the Engineers (news, technical information, and much more)
  - http://mysqlhighavailability.com


ORACLE®