Understanding MySQL Group Replication

Libing Song(libing.song@oracle.com)
Software Engineer
MySQL Replication Team

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. Use cases
3. Deployment modes
4. Features
5. Performance
6. Architecture
7. Conclusion

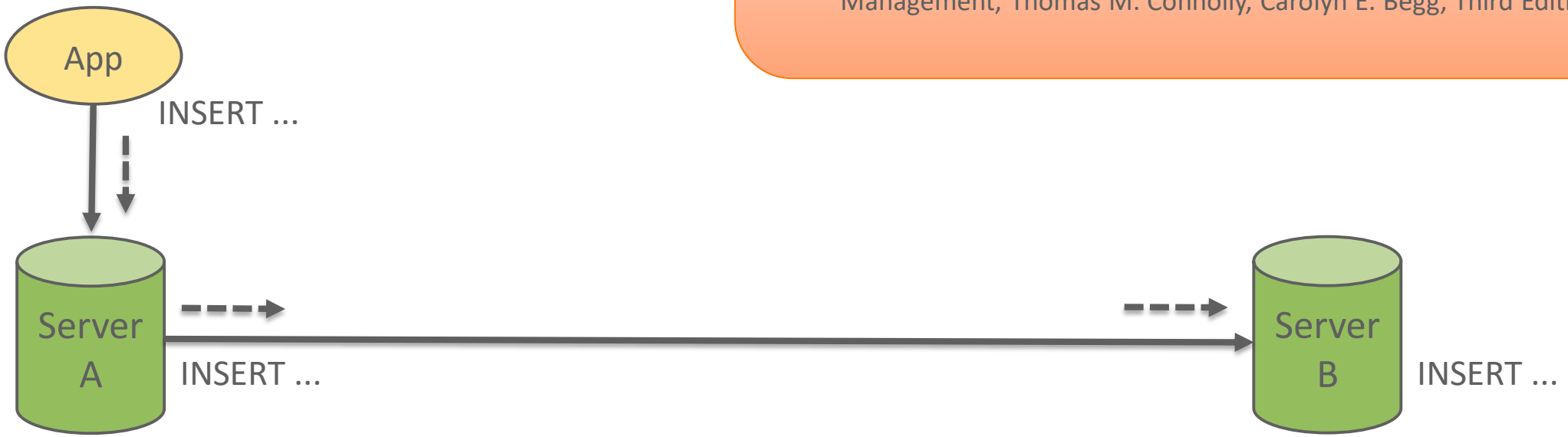
1 Background

Database Replication

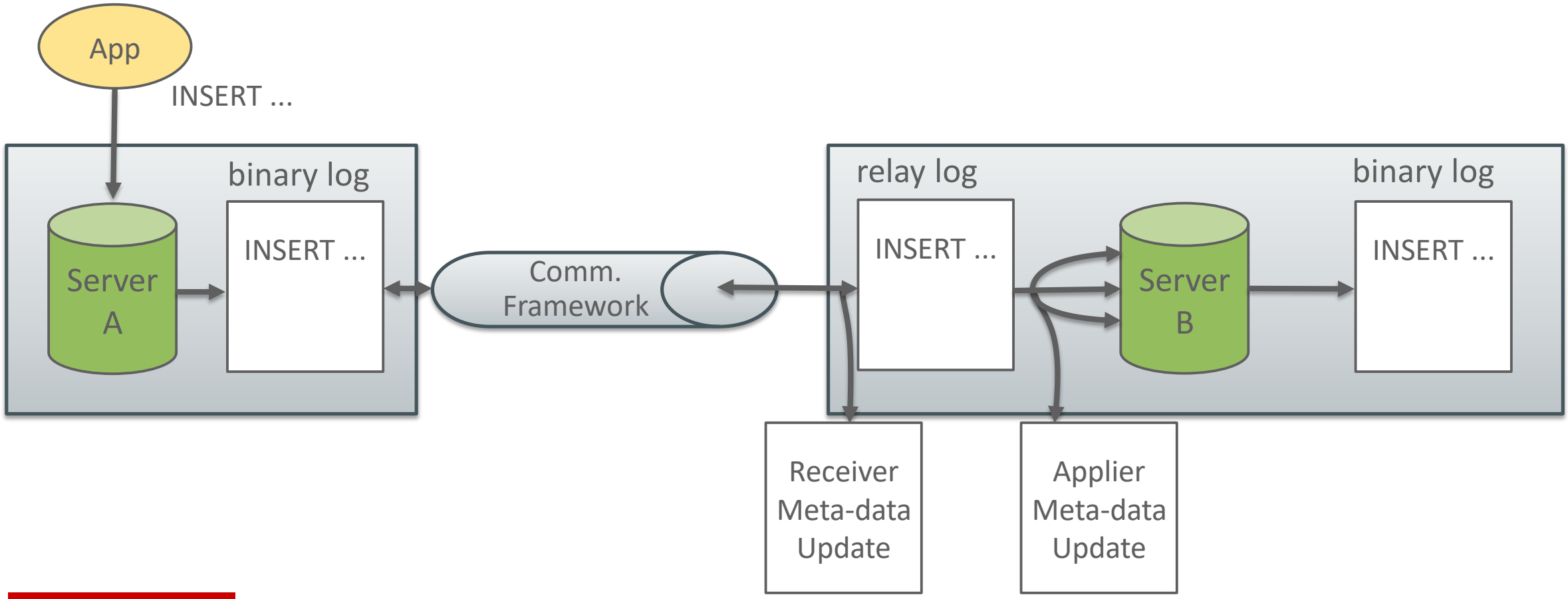
Replication

"The process of generating and reproducing multiple copies of data at one or more sites.",

Database Systems: A Practical Approach to Design, Implementation, and Management, Thomas M. Connolly, Carolyn E. Begg, Third Edition, 2002.



MySQL Database Replication: Overview

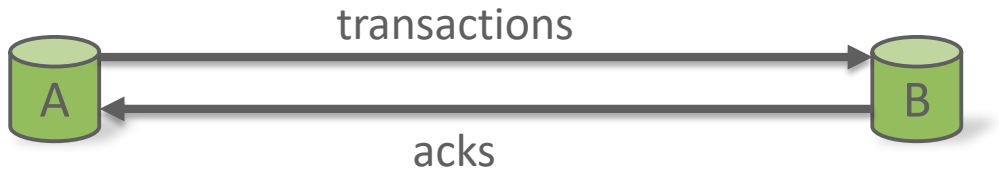


MySQL Database Replication: Some Notes

Coordination Between Servers

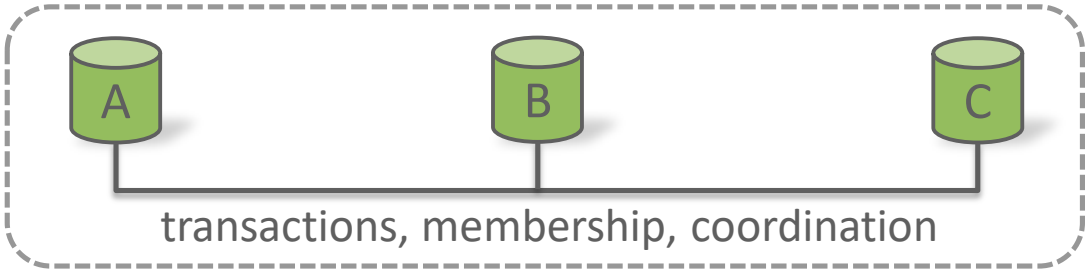
Since 3.23

asynchronous (native)



Since 5.5

semi-synchronous (plugin)



Since 5.7.17

And now in MySQL 8 as of 8.0.1

group replication (plugin)

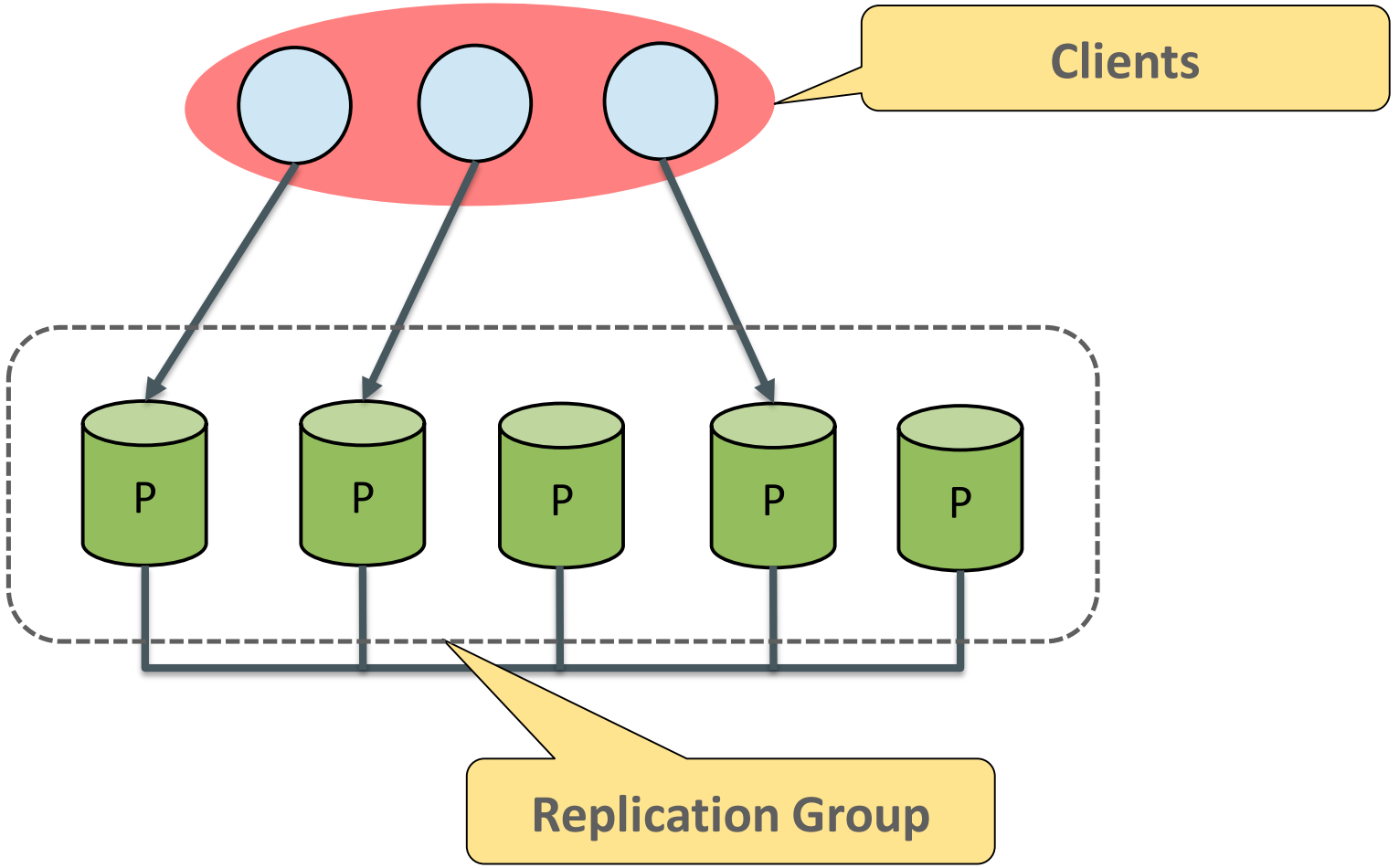
MySQL Group Replication

- **What is MySQL Group Replication?**

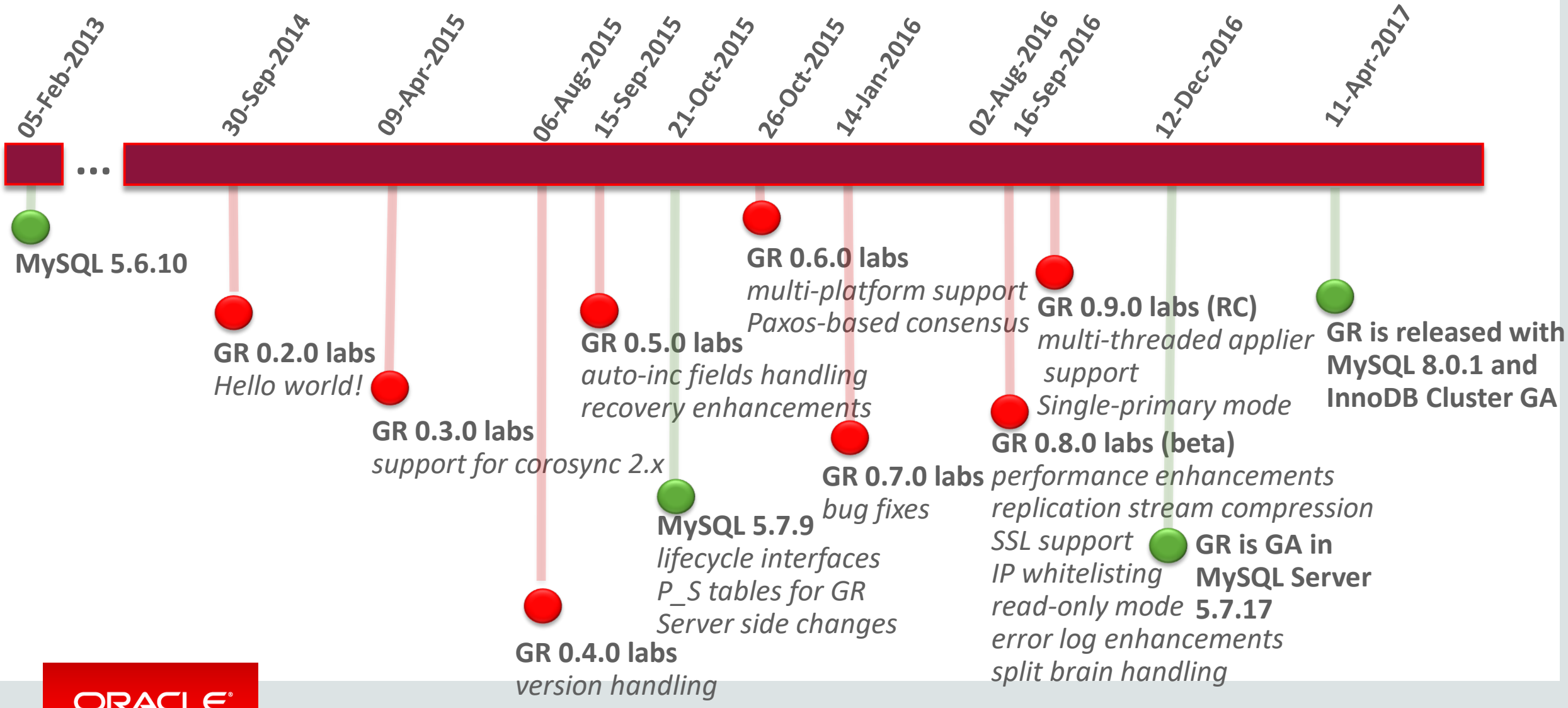
  "Single/Multi-primary **update everywhere** replication plugin for MySQL with built-in **automatic distributed recovery**, **conflict detection** and **group membership**."

- **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



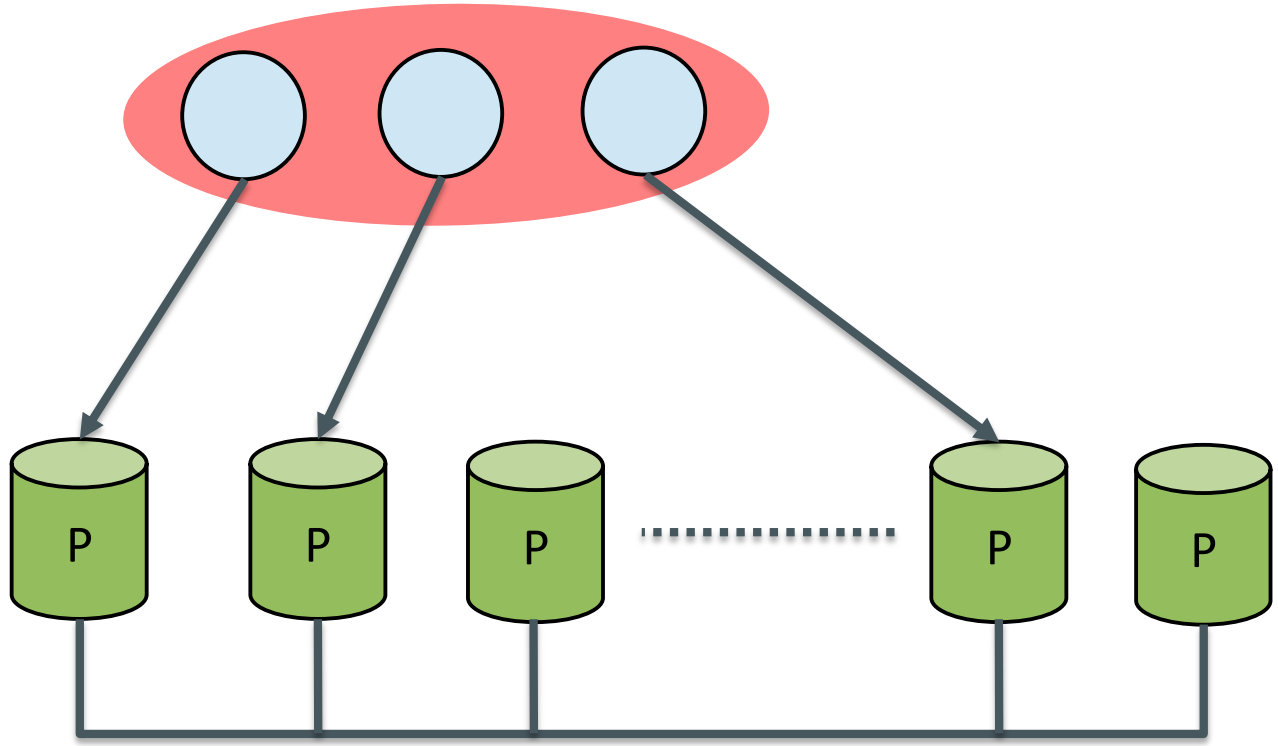
The Road to Group Replication in MySQL 8 and InnoDB Clusters



2 Use cases

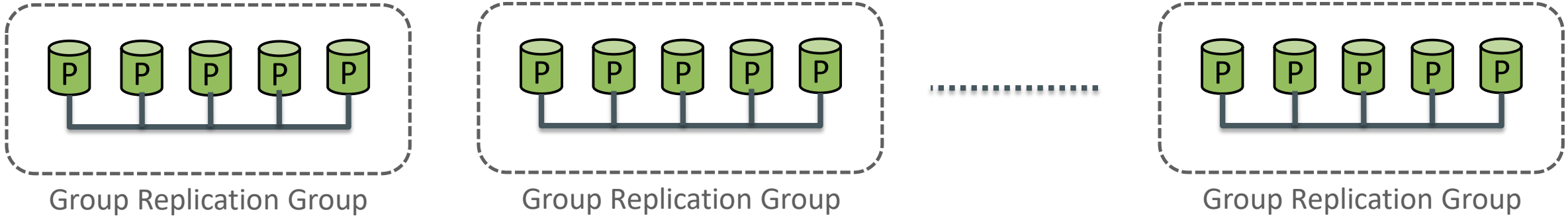
Use Cases

- **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

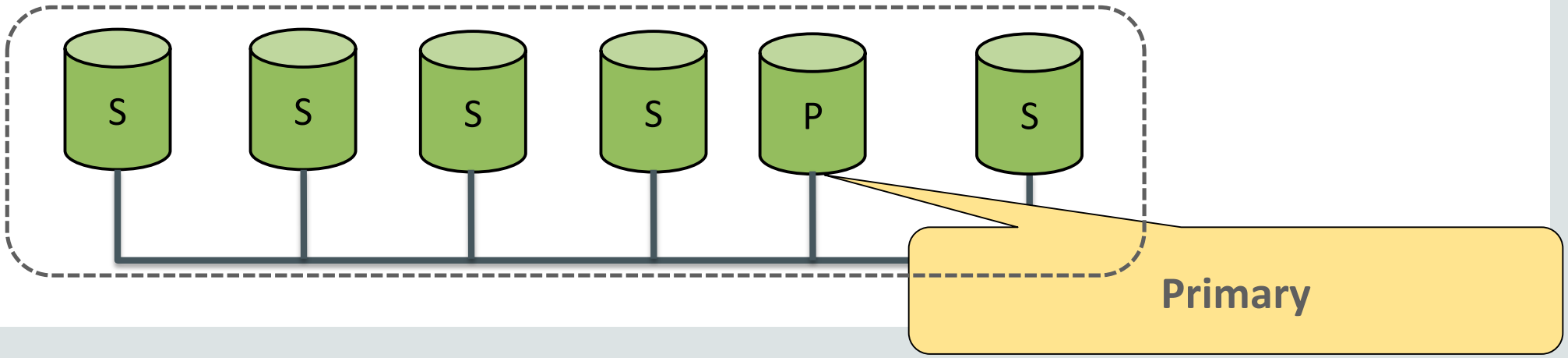
- **Highly Available Shards**
  - Sharding is a popular approach to achieve write scale-out. Users can use MySQL Group Replication to implement highly available shards. Each shard can map into a Replication Group.



Group Replication Group | Group Replication Group | Group 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



3 Deployment modes

3.1 Single-primary

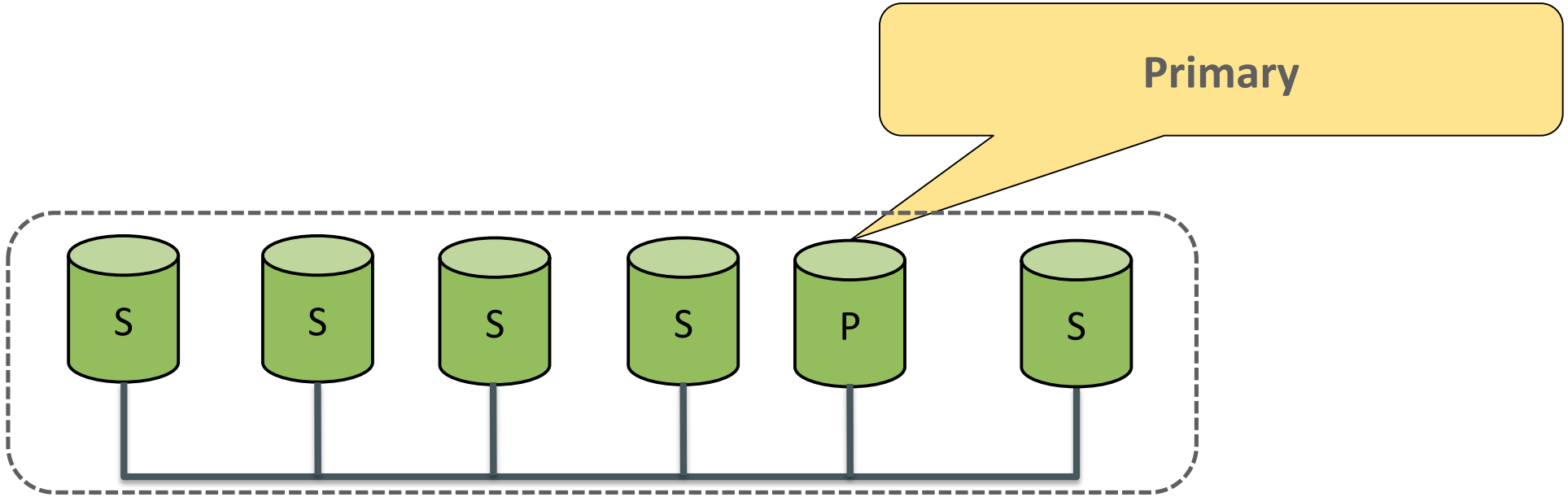
Single-primary mode

- 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-primary mode by default.

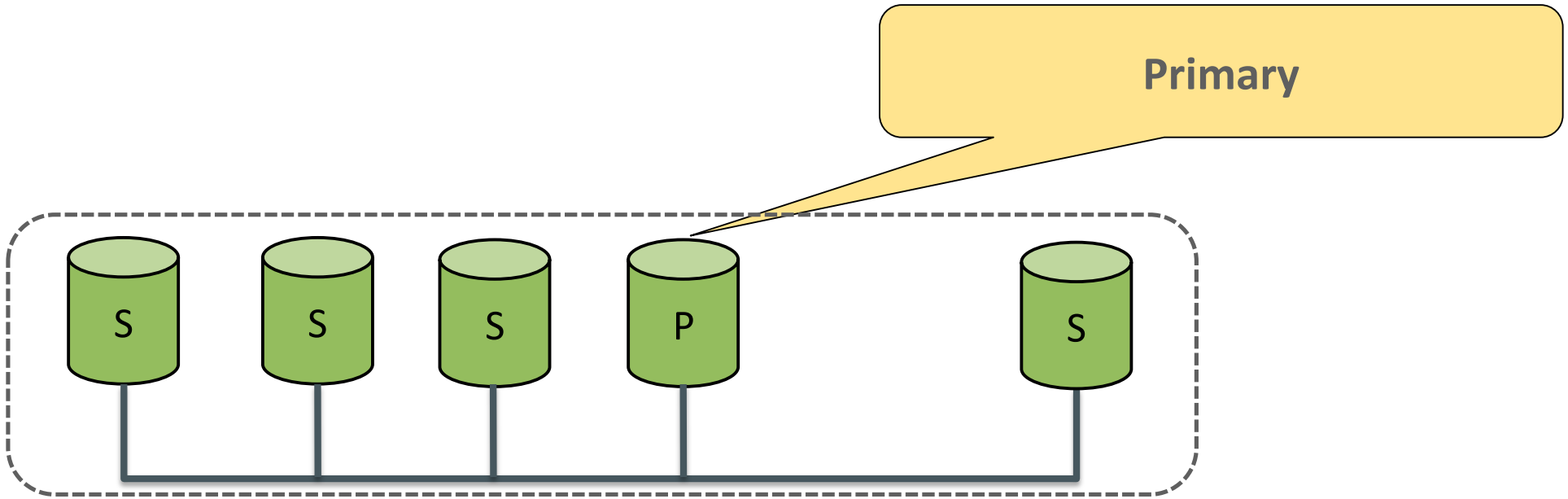
Single-primary mode

- Automatic primary promotion election.
- Secondaries 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
```

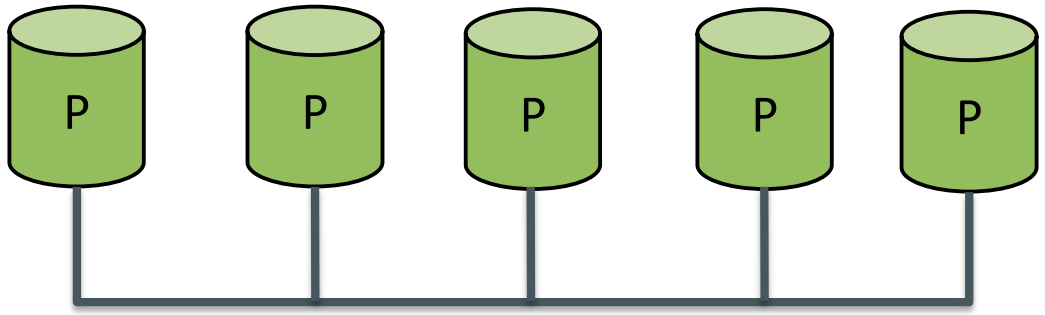
3 Deployment modes

3.1 Single-primary

3.2 Multi-primary

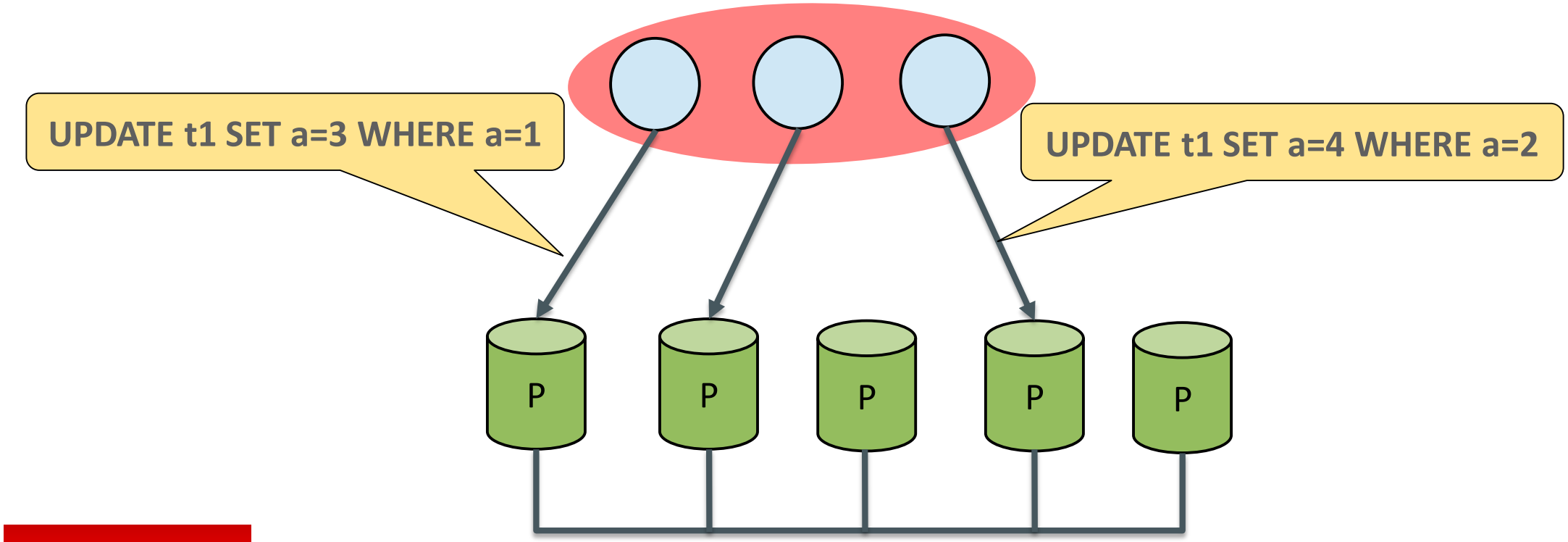
Multi-primary update everywhere!

- Configuration mode that makes all members writable
  - Enabled by setting option --group_replication_single_primary_mode to OFF



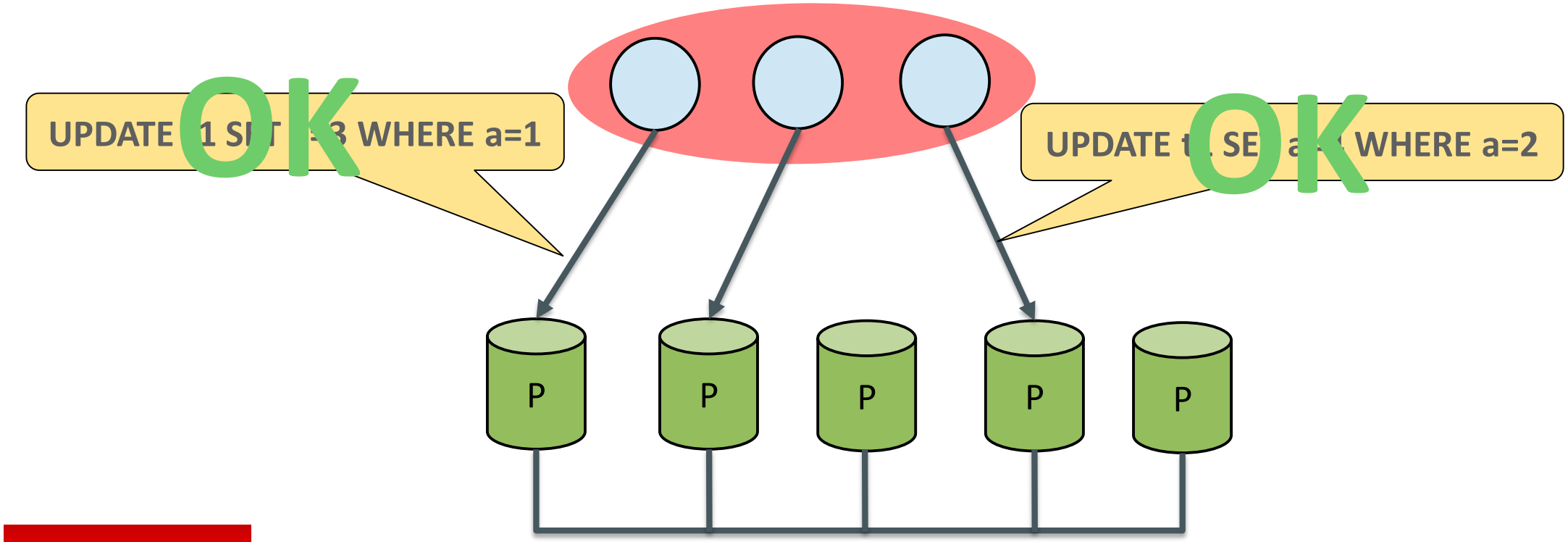
Multi-primary update everywhere!

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



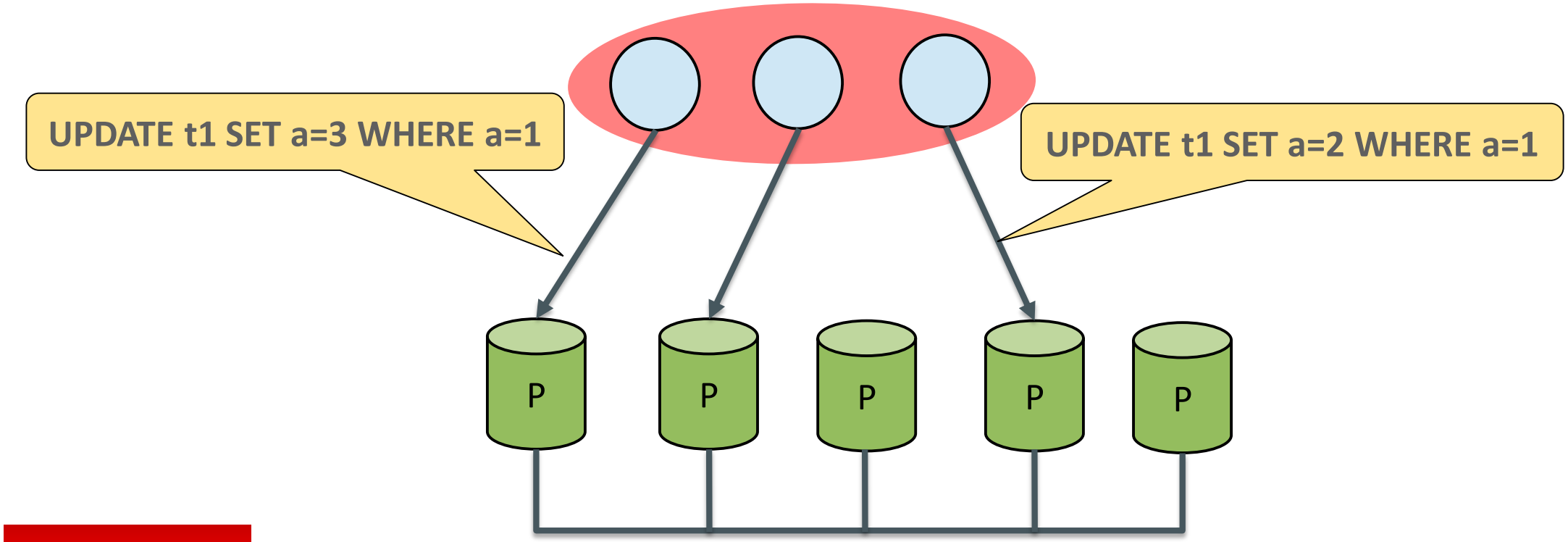
Multi-primary update everywhere!

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



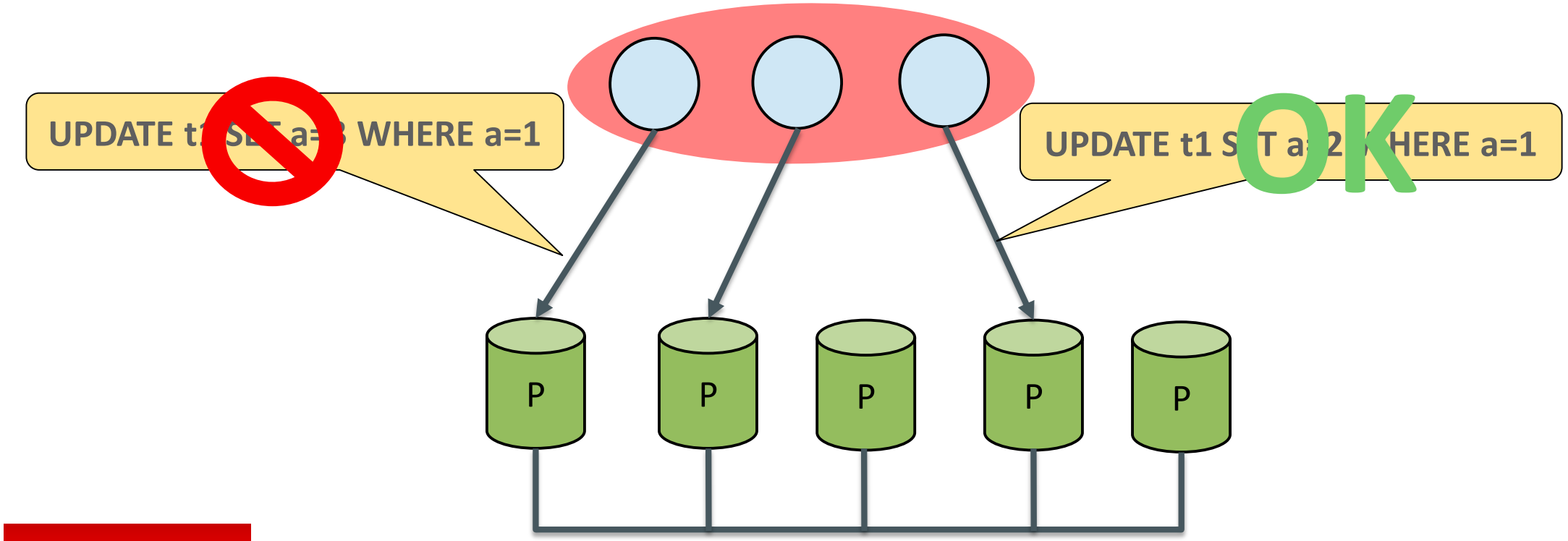
Multi-primary update everywhere!

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



Multi-primary 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.

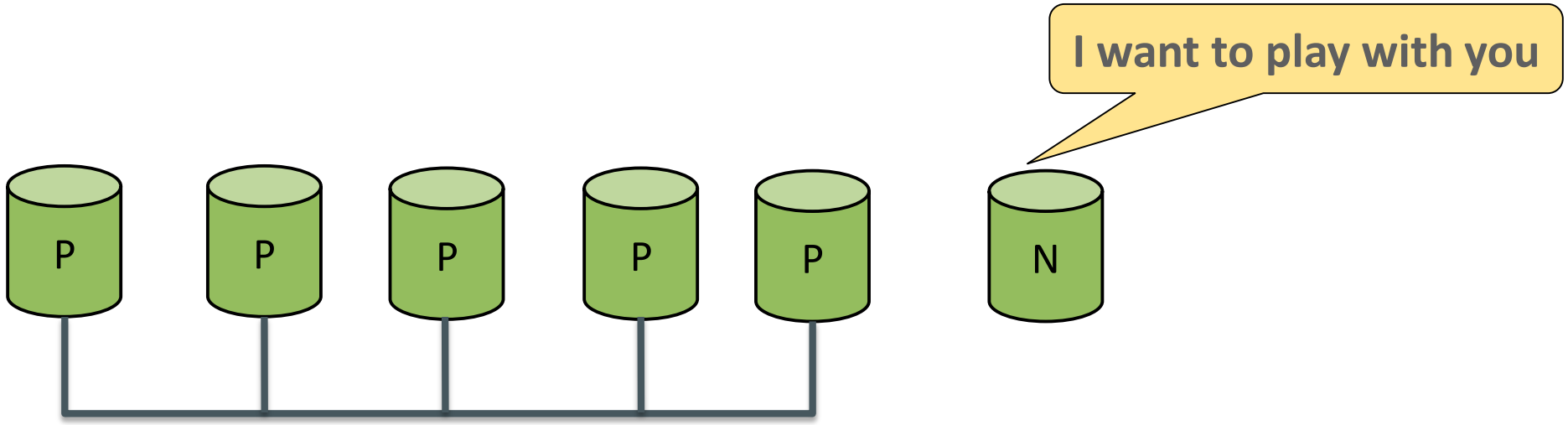


4 Features

4.1 Automatic distributed server recovery

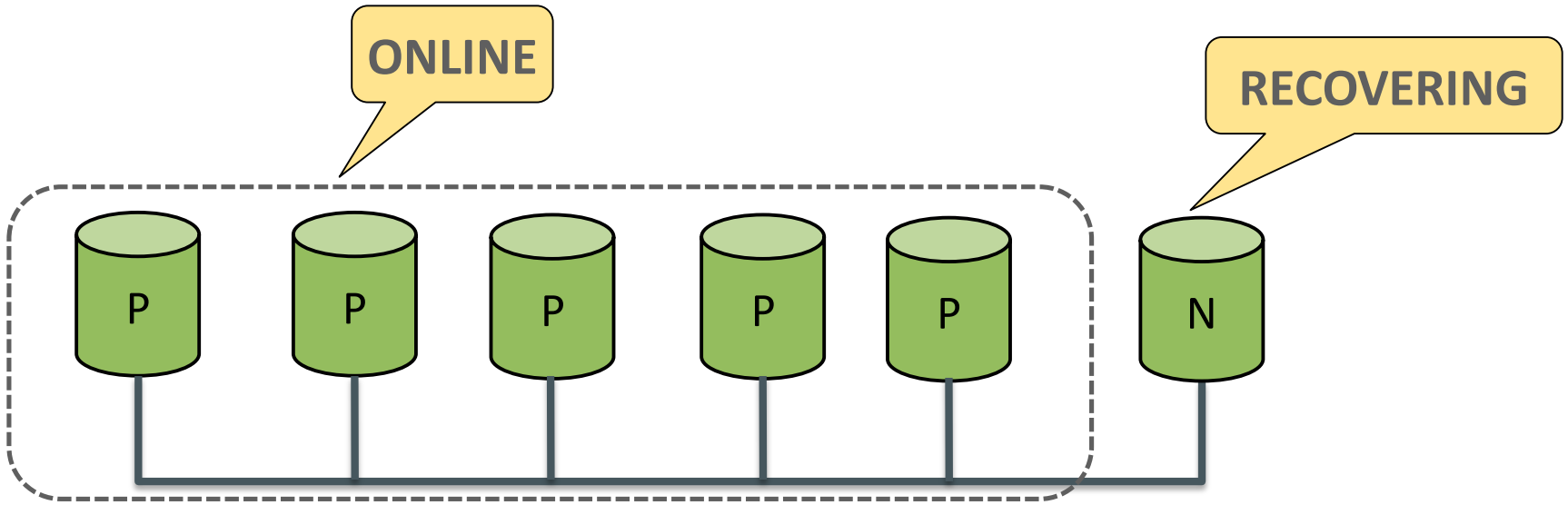
Automatic distributed server recovery!

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



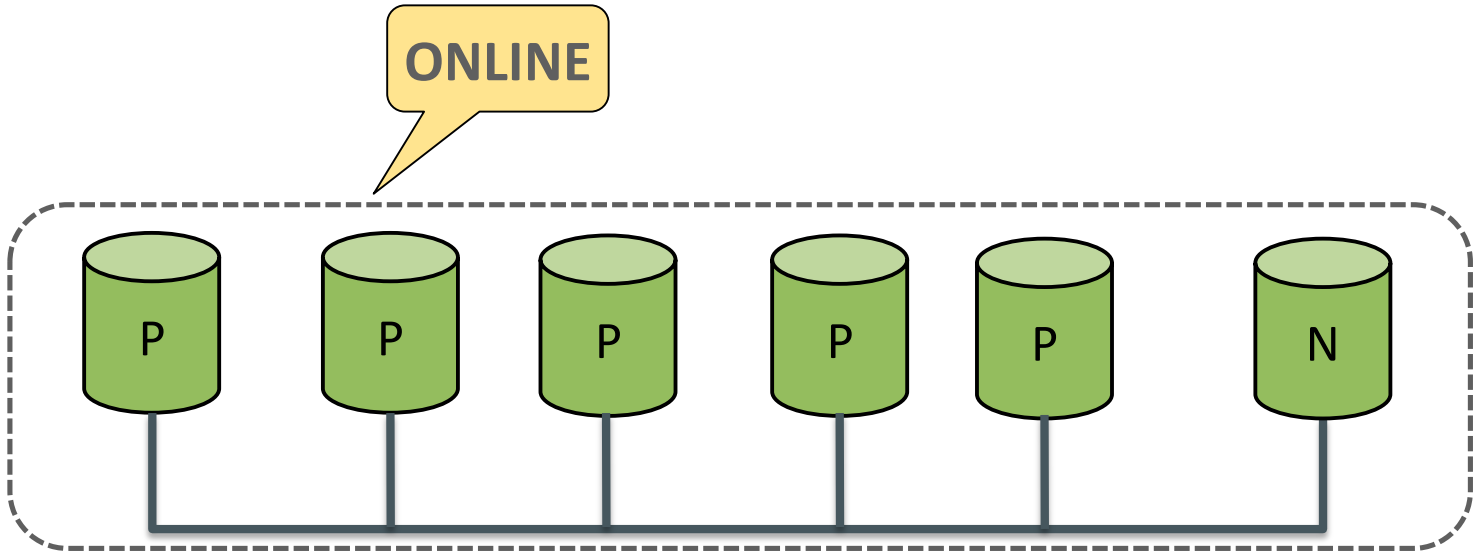
Automatic distributed server recovery!

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



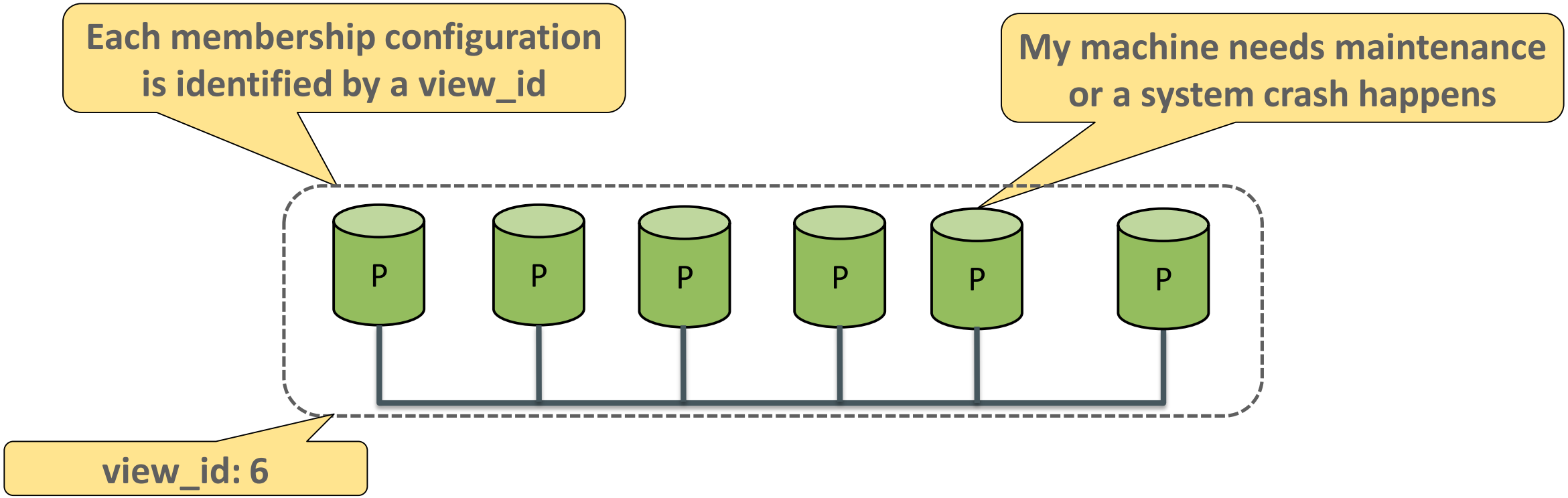
Automatic distributed server recovery!

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



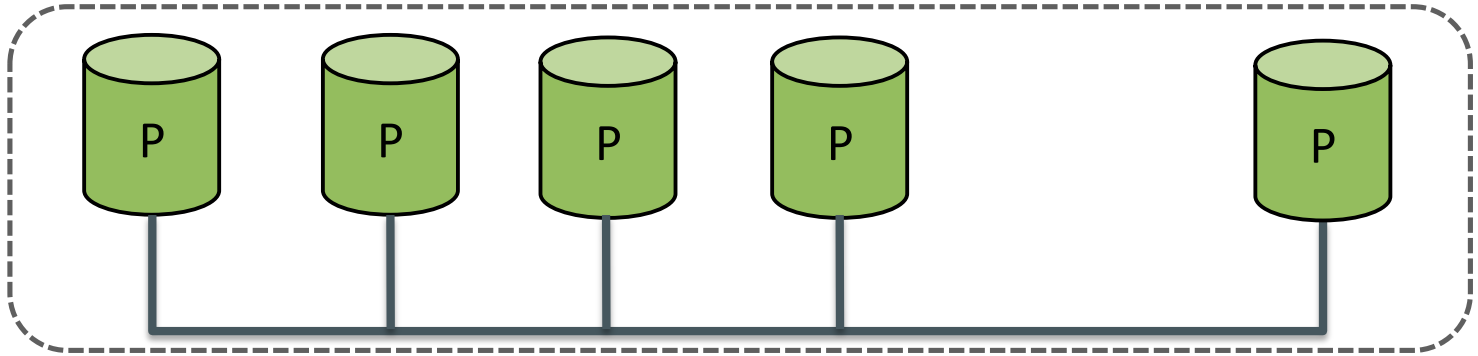
Automatic distributed server recovery!

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



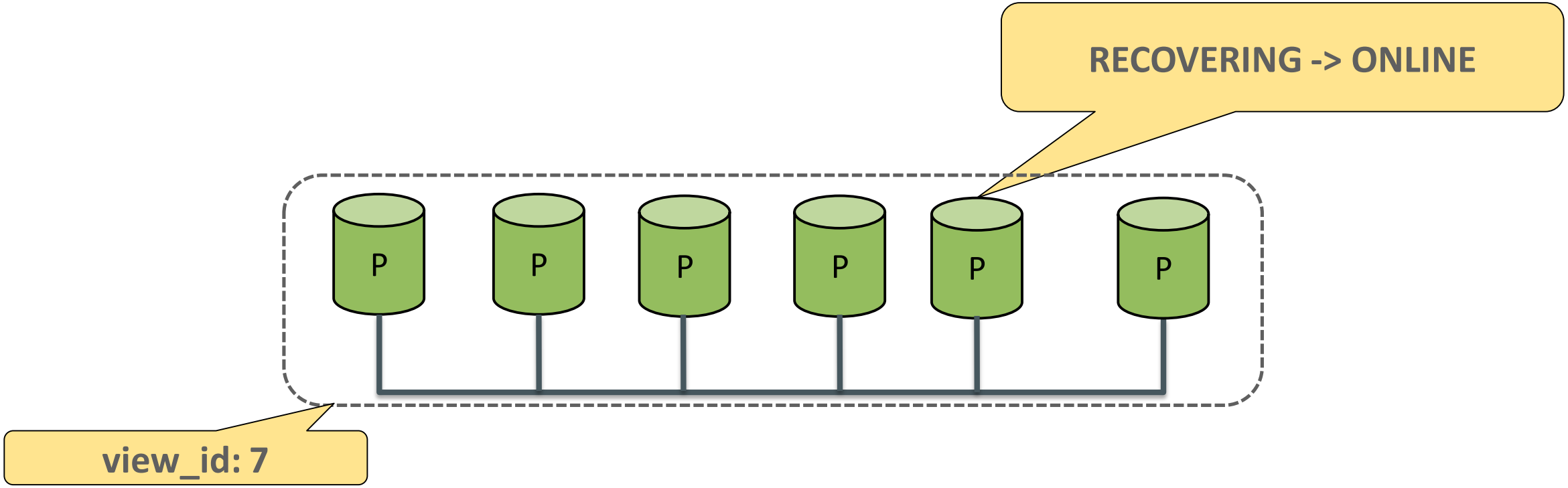
Automatic distributed server recovery!

- 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.



4 Features

4.1 Automatic distributed server recovery

4.2 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.

4 Features

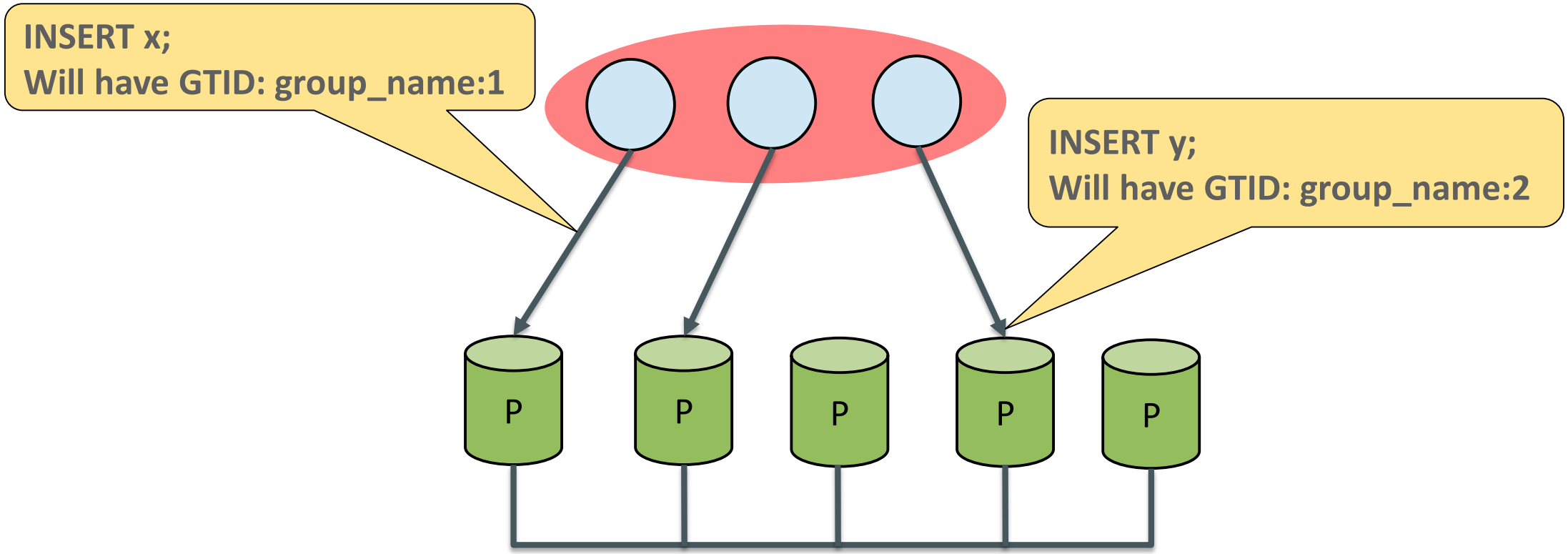
4.1 Automatic distributed server recovery

4.2 MySQL Look & Feel

4.3 Full GTID support

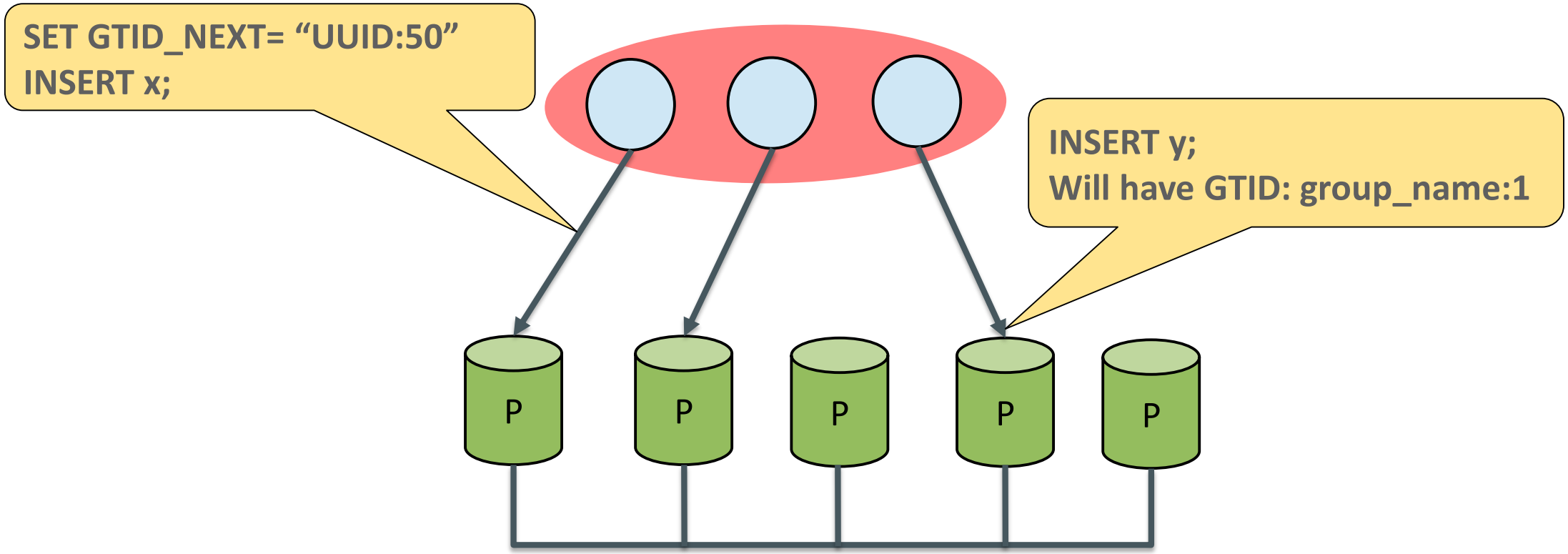
Full GTID support!

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



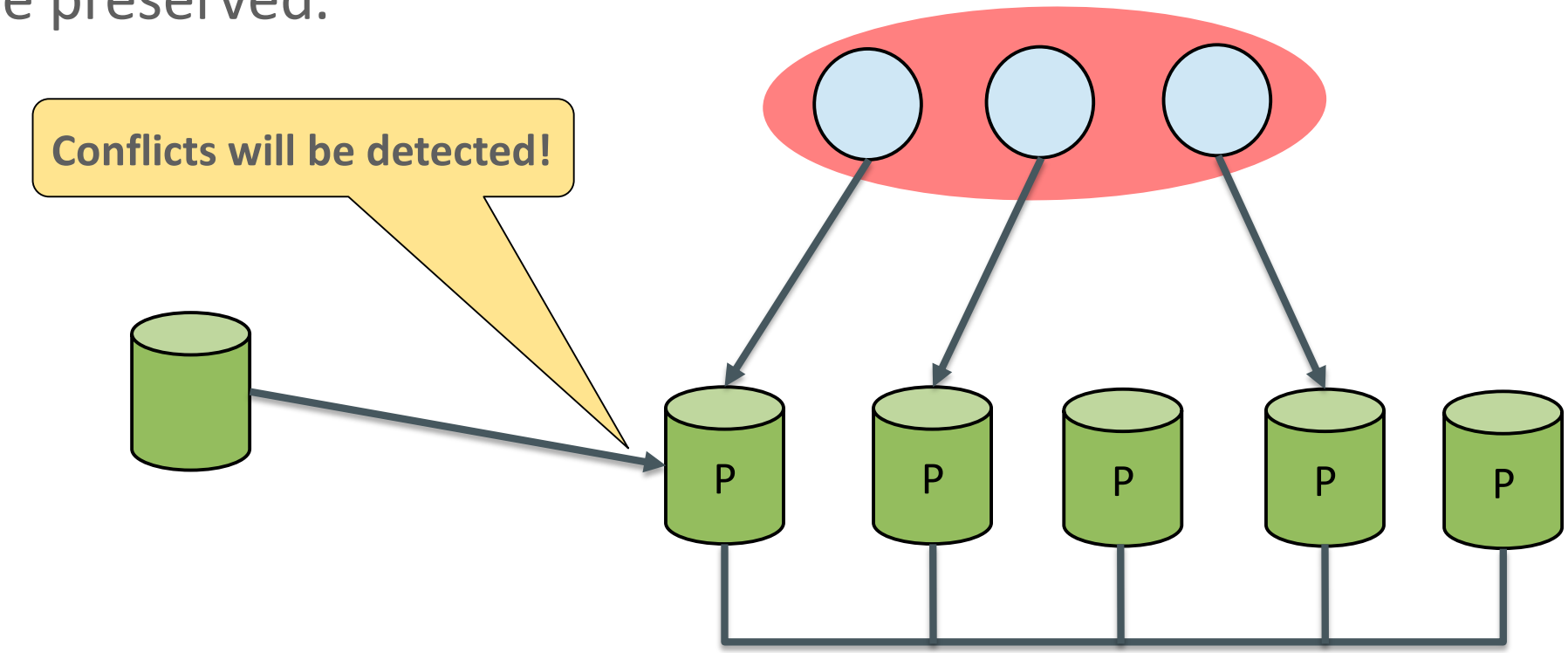
Full GTID support!

- Users can specify the identifier for the transaction.



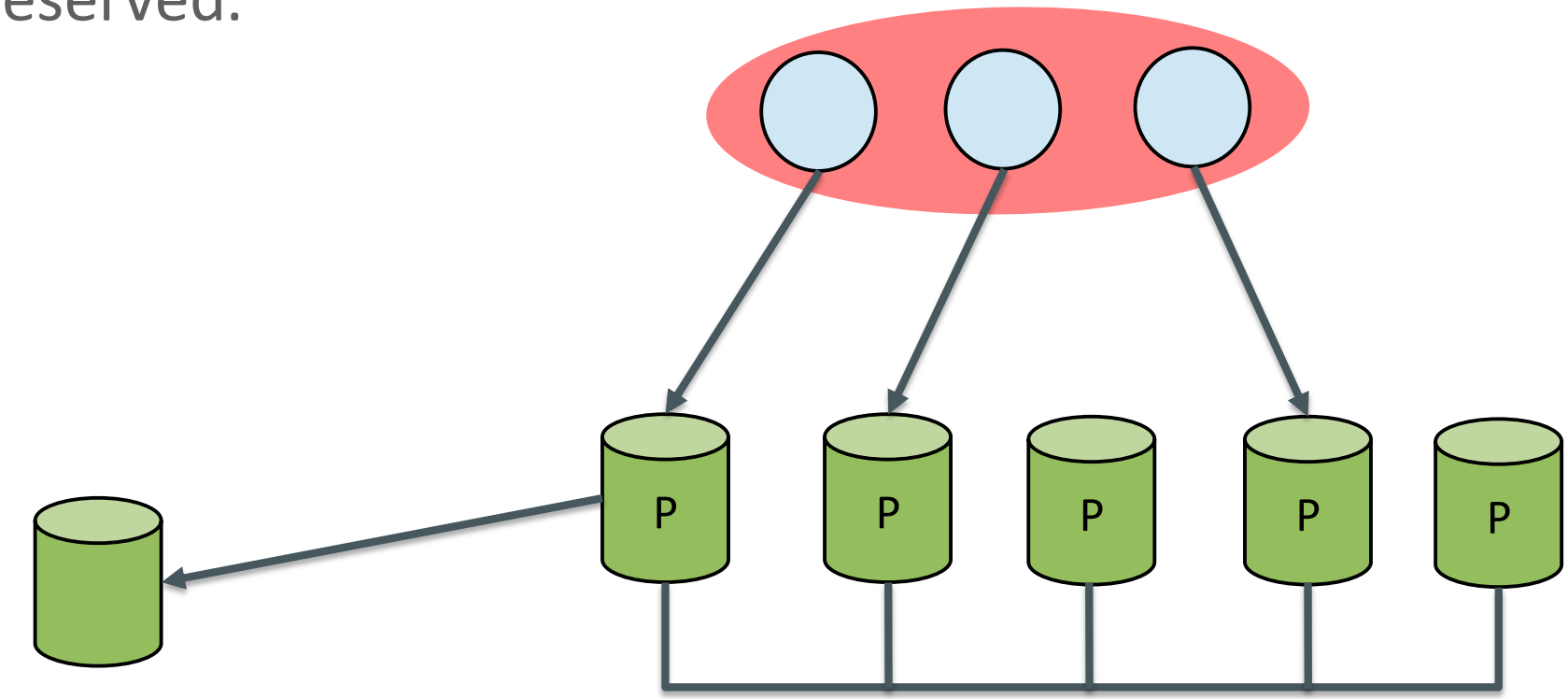
Full GTID support!

- 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.



4 Features

4.1 Automatic distributed server recovery

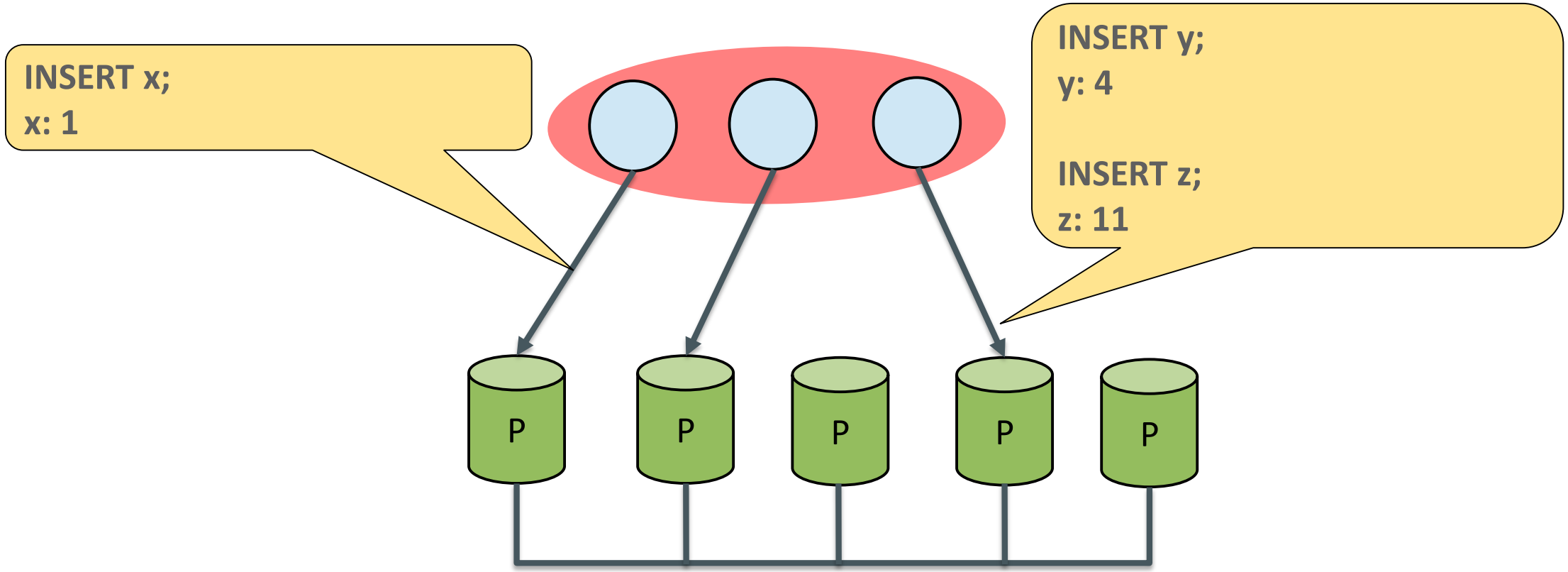
4.2 MySQL Look & Feel

4.3 Full GTID support

4.4 Auto-increment configuration/handling

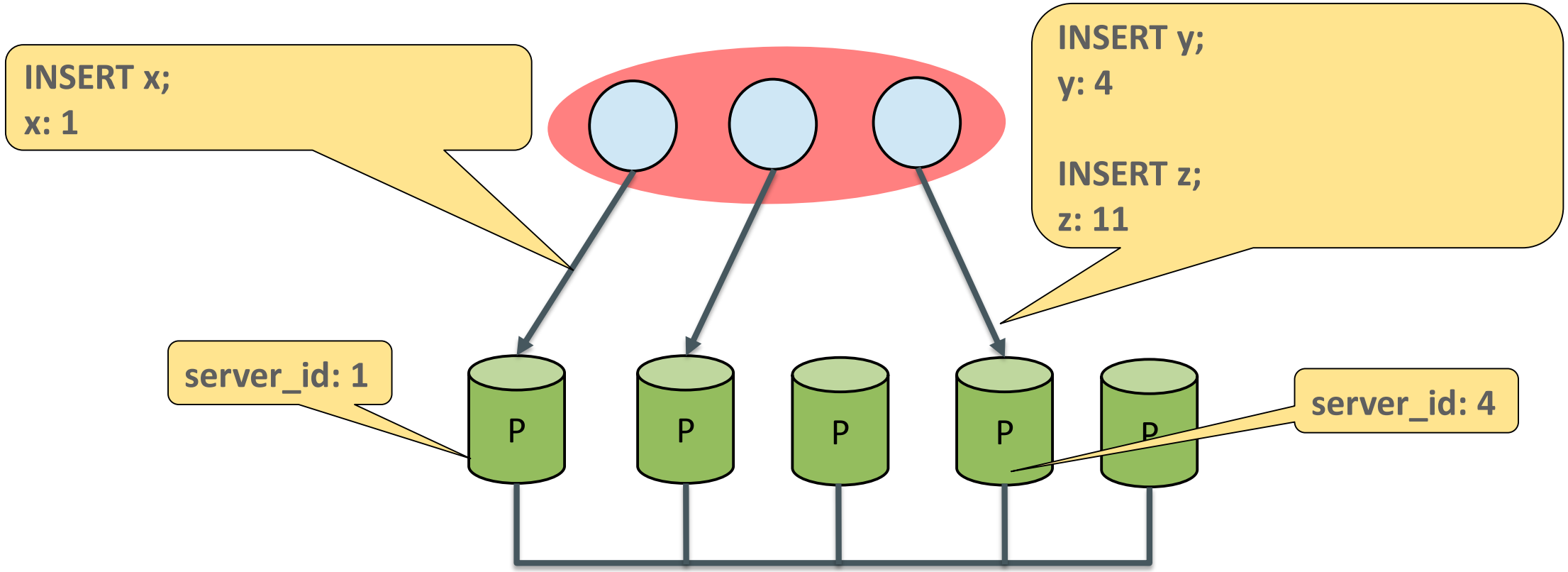
Auto-increment configuration/handling

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



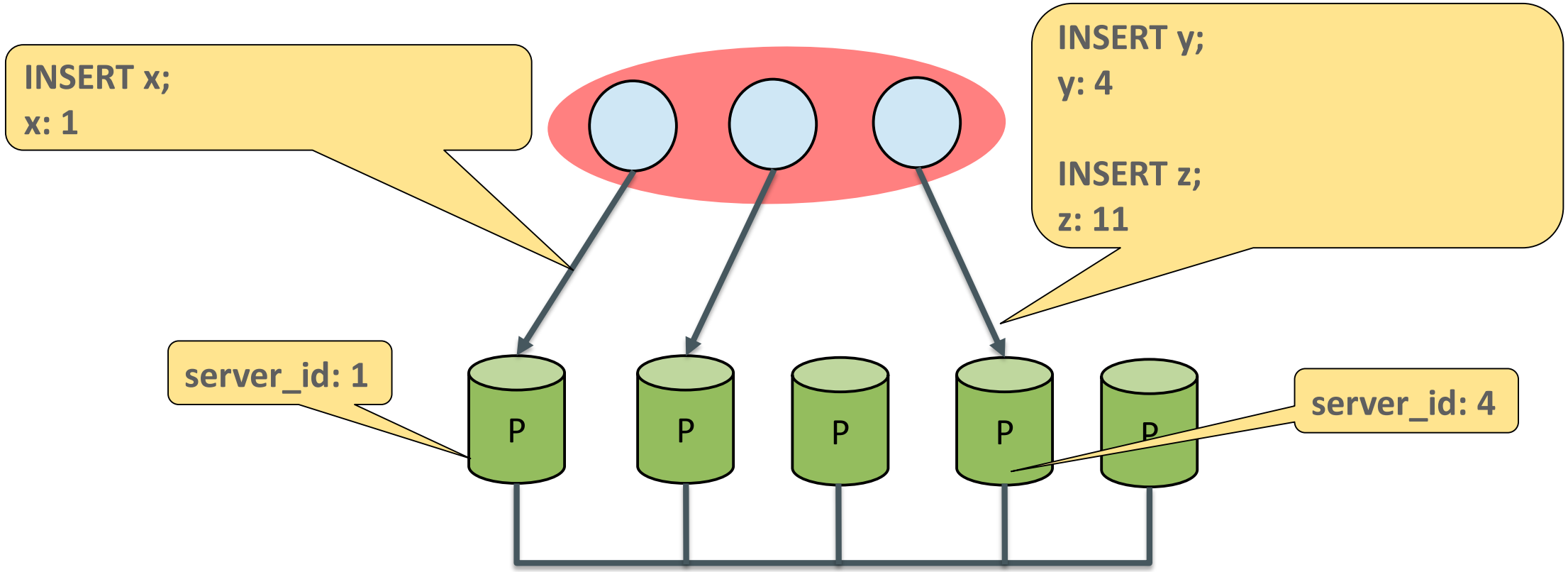
Auto-increment configuration/handling

- 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.



4 Features

4.1 Automatic distributed server recovery

4.2 MySQL Look & Feel

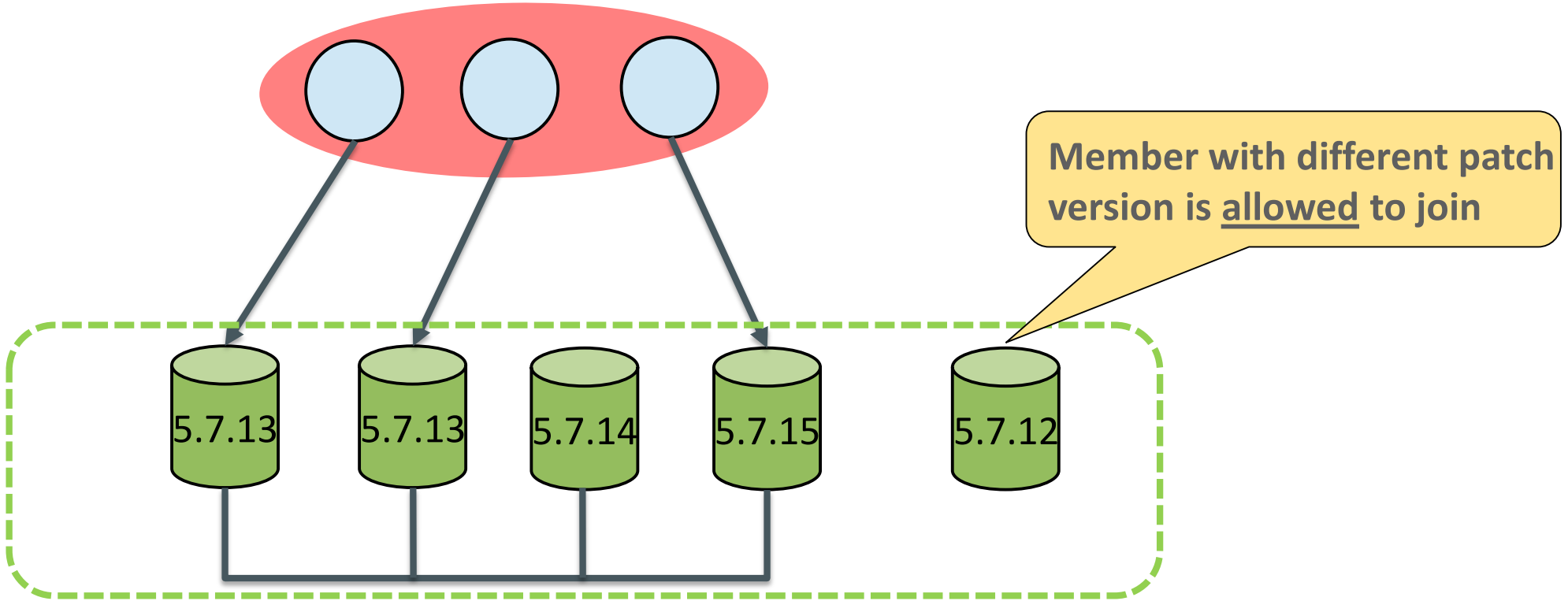
4.3 Full GTID support

4.4 Auto-increment configuration/handling

4.5 Plugin version access control

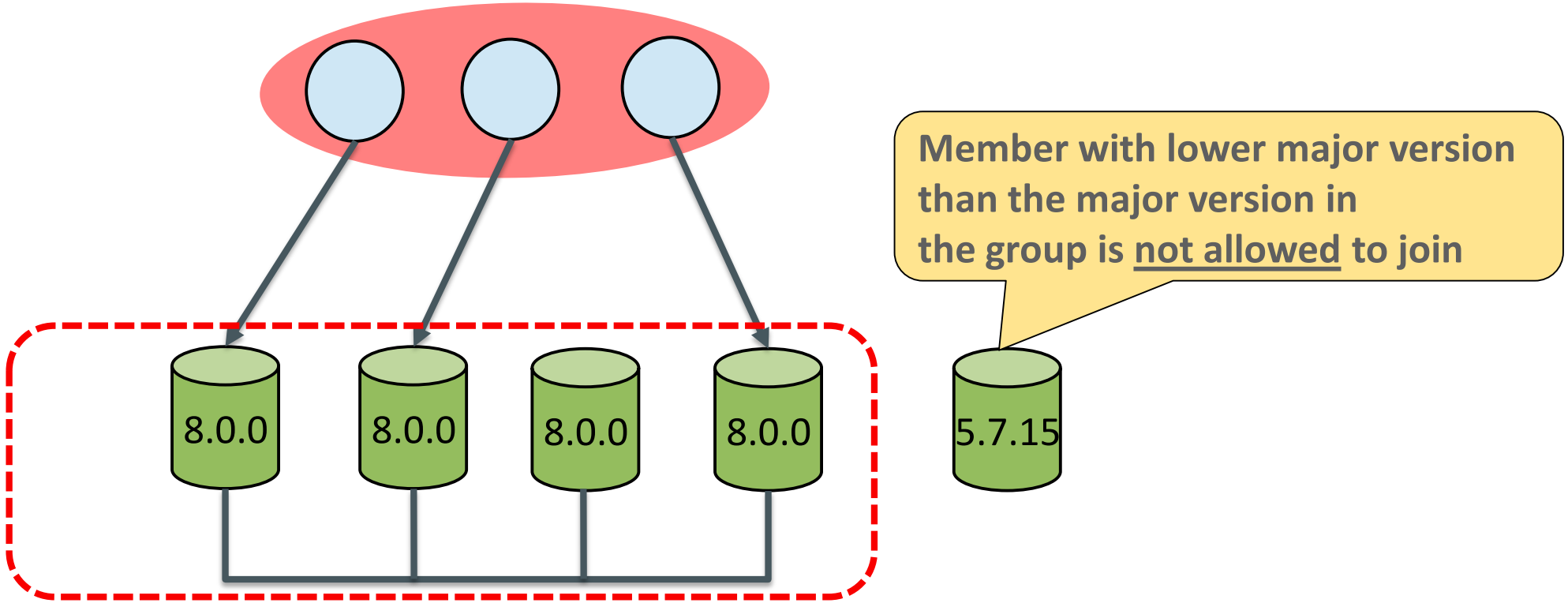
Plugin Version Access Control

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



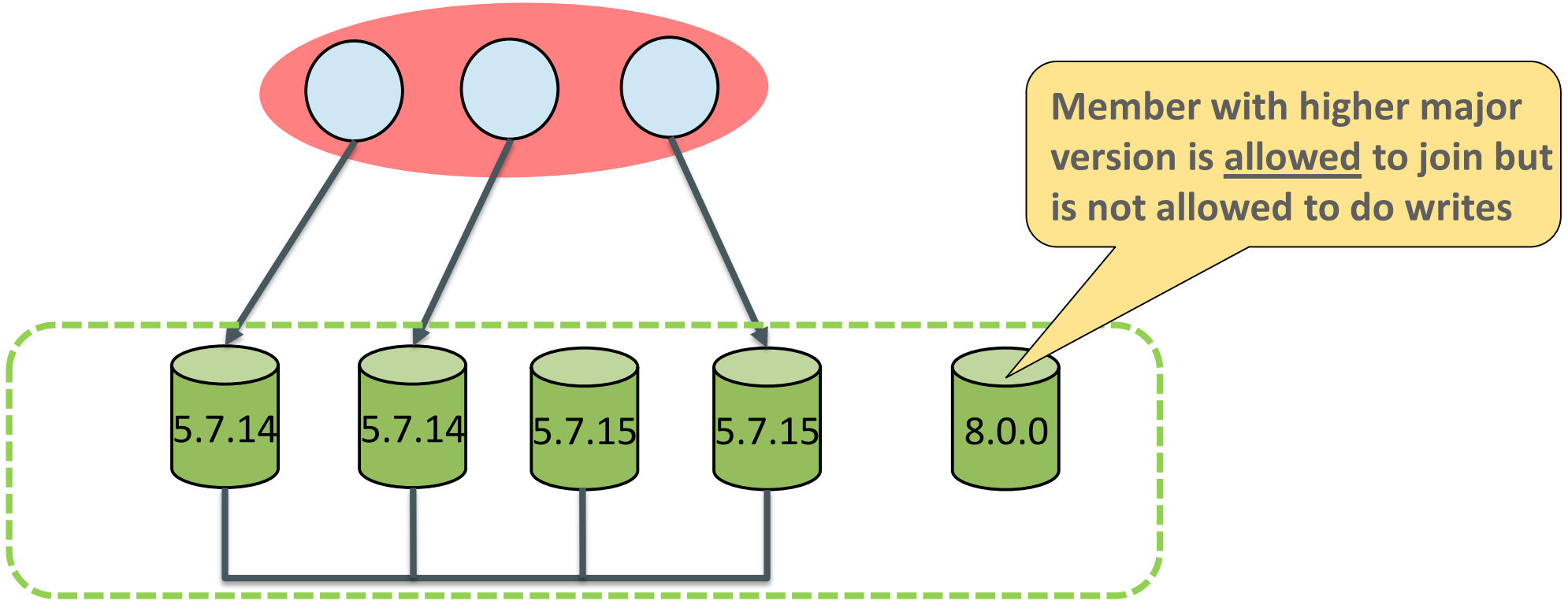
Plugin Version Access Control

- 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.



4 Features

4.1 Automatic distributed server recovery

4.2 MySQL Look & Feel

4.3 Full GTID support

4.4 Auto-increment configuration/handling

4.5 Plugin version access control

4.6 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.

4 Features

4.1 Automatic distributed server recovery

4.2 MySQL Look & Feel

4.3 Full GTID support

4.4 Auto-increment configuration/handling

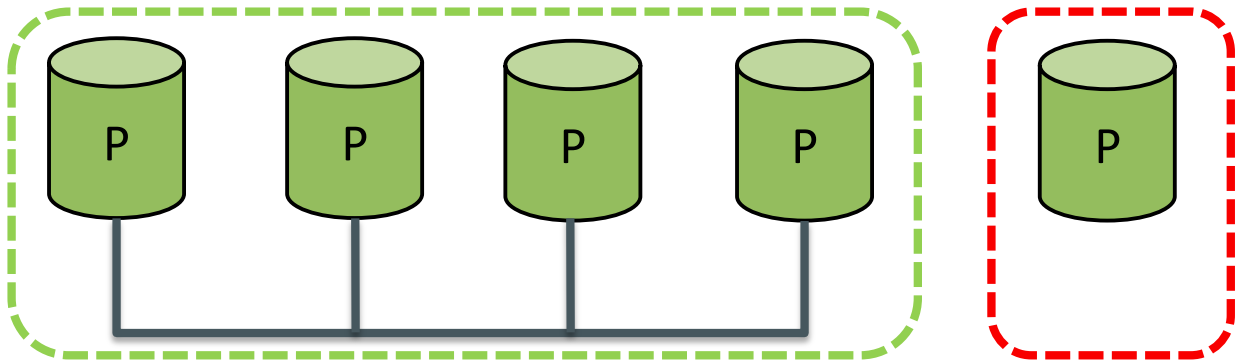
4.5 Plugin version access control

4.6 Built-in communication engine

4.7 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.



4 Features

4.8 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

4 Features

4.8 Full stack secure connections

4.9 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
```

Parallel applier support

- Write set Based Transaction Dependencies
  - Already used on Group Replication from the beginning
  - Speedup distributed recovery time

```
master> SET @@GLOBAL.binlog_transaction_dependency_tracking=WRITESET;
Query OK, 0 rows affected (0,00 sec)

master> SET @@GLOBAL.binlog_transaction_dependency_tracking=WRITESET_SESSION;
Query OK, 0 rows affected (0,00 sec)

master> SET @@GLOBAL.binlog_transaction_dependency_tracking=COMMIT_ORDER; -- default
Query OK, 0 rows affected (0,00 sec
```

4 Features

4.8 Full stack secure connections

4.9 Parallel applier support

4.10 Transaction SAVEPOINT support

Transaction SAVEPOINT support

```
mysql> BEGIN;
Query OK, 0 rows affected (0,00 sec)

mysql> INSERT INTO t1 VALUES(1);
Query OK, 1 row affected (0,00 sec)

mysql> SAVEPOINT S1;
Query OK, 0 rows affected (0,00 sec)

mysql> INSERT INTO t1 VALUES(2);
Query OK, 1 row affected (0,00 sec)

mysql> ROLLBACK TO S1;
Query OK, 0 rows affected (0,00 sec)

mysql> COMMIT;
Query OK, 0 rows affected (0,00 sec)
```

4 Features

4.8 Full stack secure connections

4.9 Parallel applier support

4.10 Transaction SAVEPOINT support

4.11 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

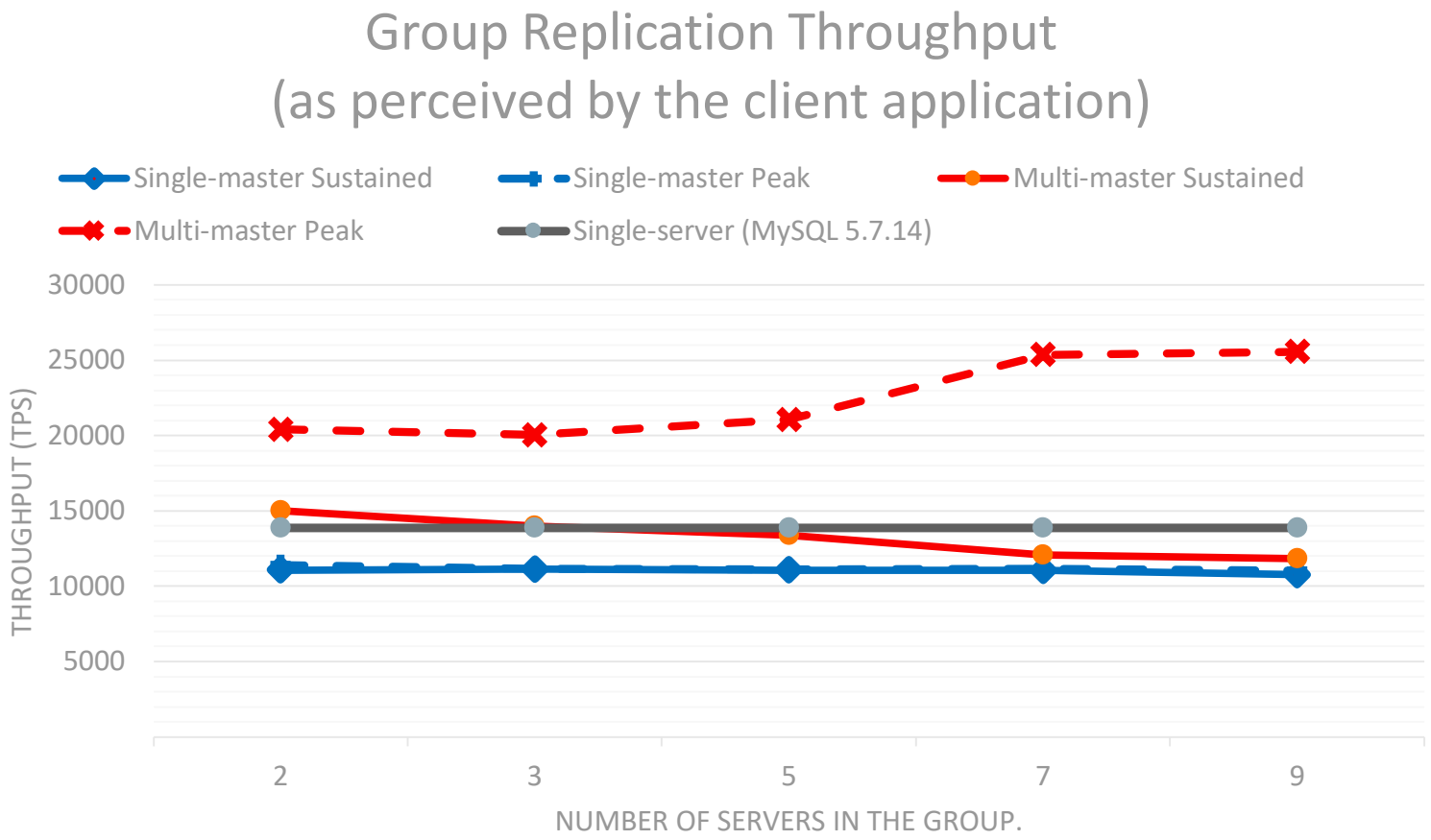
- Serializable (on multi-primary)
- Cascading Foreign Keys (on multi-primary)
- Binary log events checksum

Warnings

- Concurrent DDL (on multi-primary)
- SELECT *** FOR UPDATE does not have group locking (on multi-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

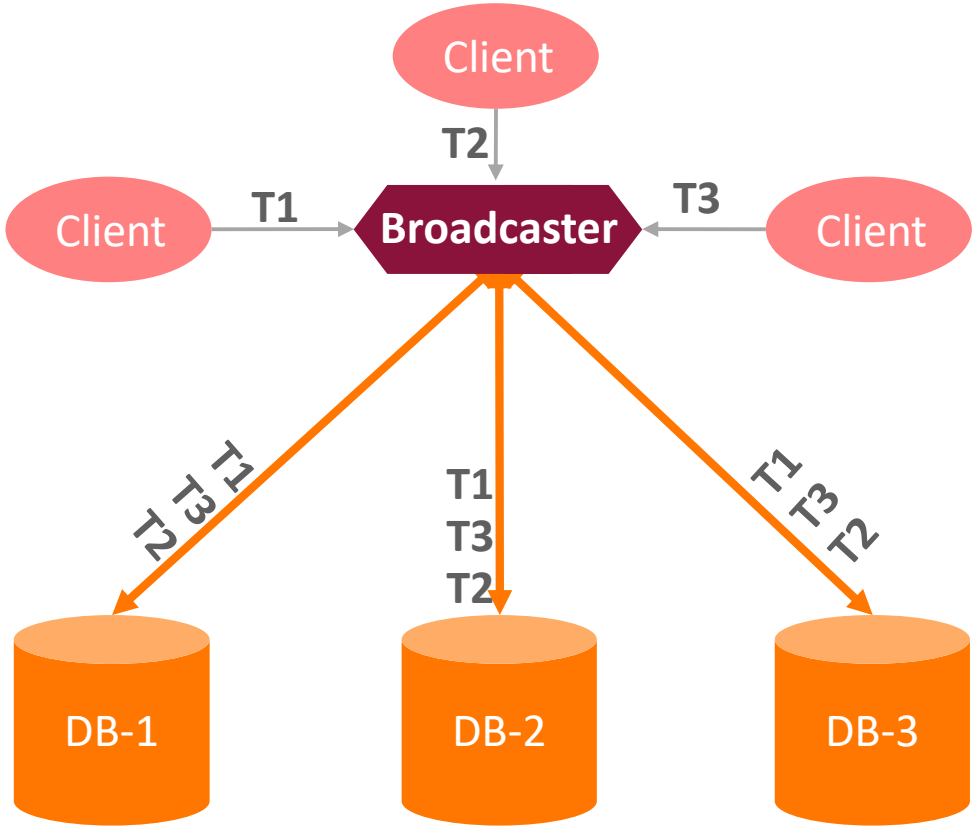
- On a sustained throughput:
  - Multi-primary performance **degrades gracefully** while going from a group with 2 servers to a group with 9 servers.
  - Single-primary performance **degrades marginally** when growing the group size.
- On a peak throughput:
  - Multi-primary exhibits **1.8X speedup** 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 Architecture

6.1 The Theories of Group Replication

The Theories of Group Replication

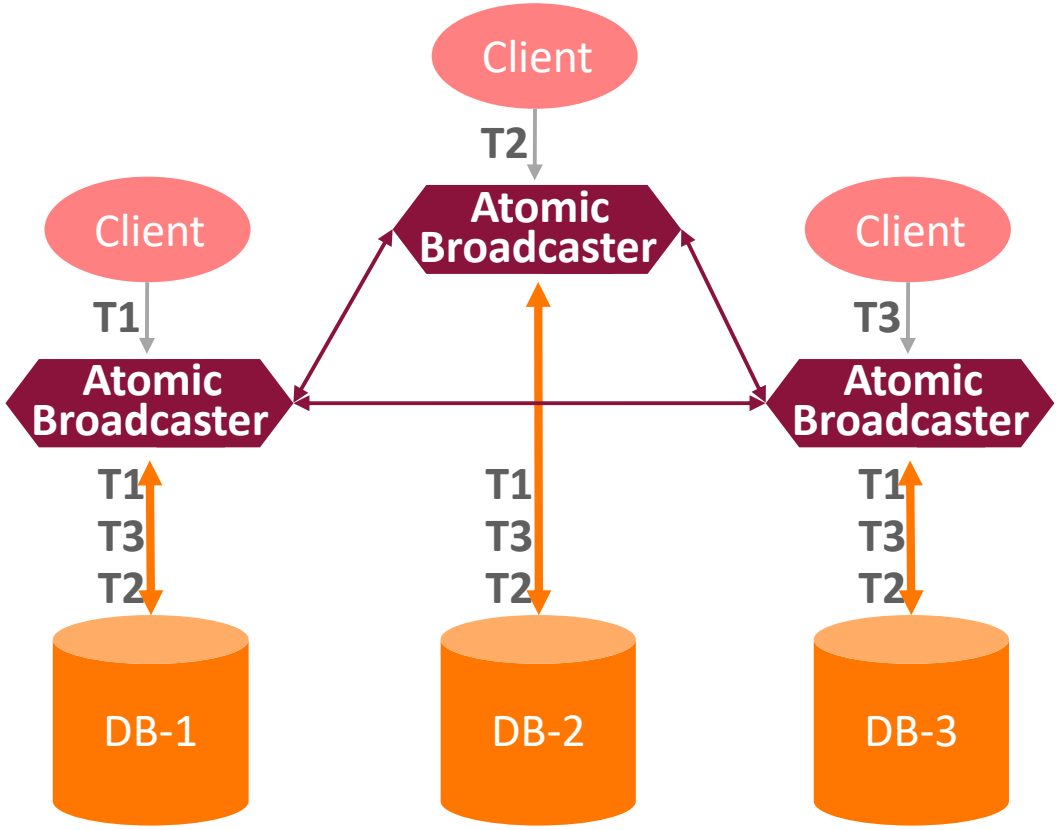
- **State Machine Replication**
  - All servers are initialized at same state.
  - Same inputs in same order generate same output state.



State Machine Replication Model

The Theories of Group Replication

- **State Machine Replication**
  - All servers are initialized at same state.
  - Same inputs in same order generate same output state.
- **Atomic Broadcast System**
  - Messages are totally ordered
  - All servers receive same messages in same order



Distributed State Machine Replication Model

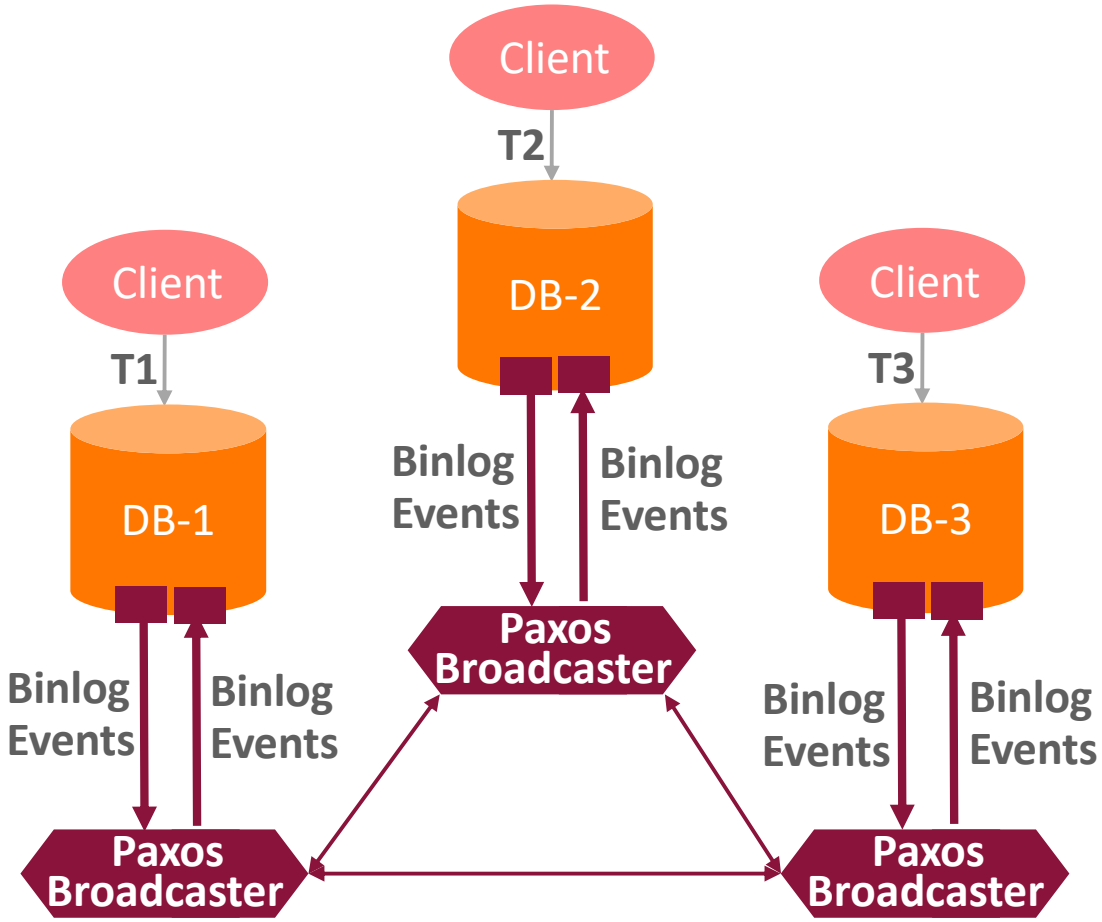
6 Architecture

6.1 The Theories of Group Replication

6.2 Group Replication Architecture

Group Replication Architecture

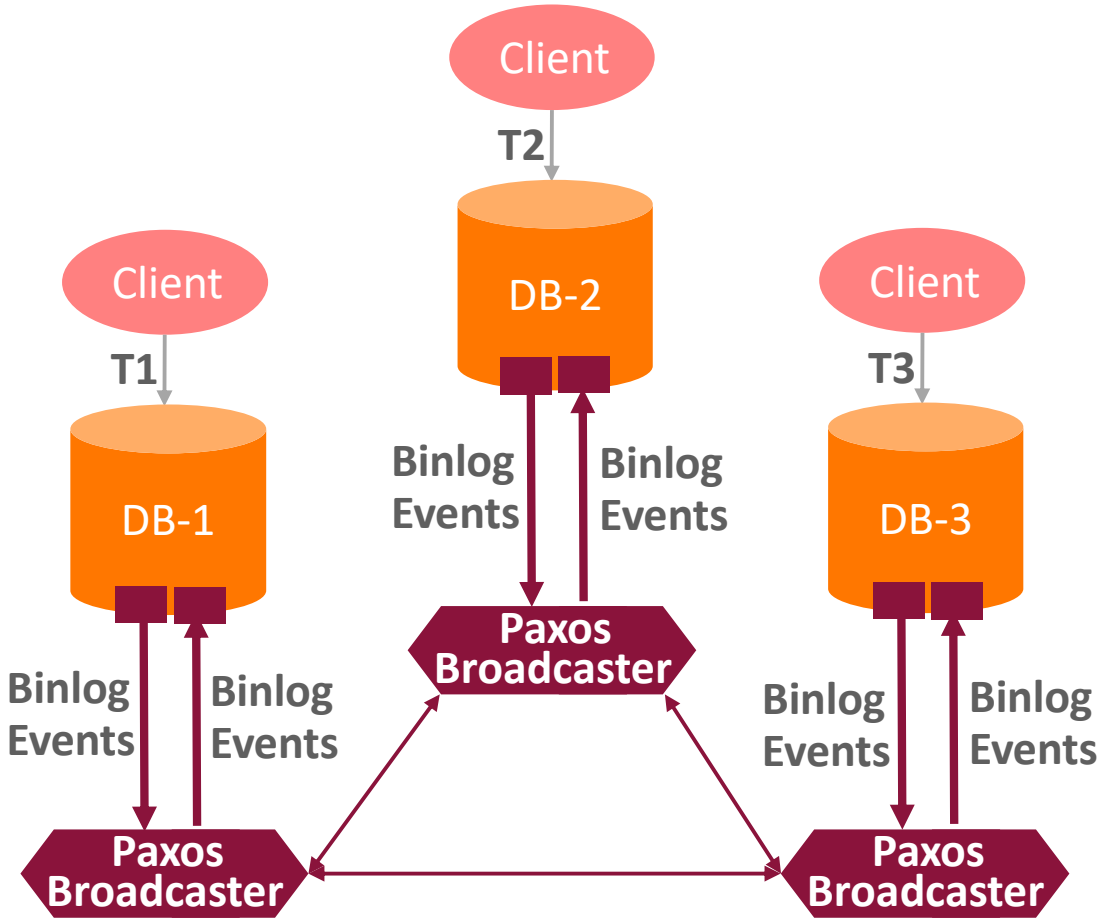
- **Broadcast Binary Log Events**
  - Row based data log
  - Captures and broadcasts binary log events after transaction execution
- **Certification(Conflict Detection)**
  - Execution before atomic broadcasting causes conflict
  - All conflicting transactions are broadcasted
  - But only the first broadcasted transaction can commit, others are rolled back



MySQL Group Replication Architecture

Group Replication Architecture

- **Binlog Event Applier**
  - Injects binlog events into relay log
  - Controls binlog event automatically
- **Paxos Based Atomic Broadcaster**
  - Concensus
  - Require majority servers available



MySQL Group Replication Architecture

6 Architecture

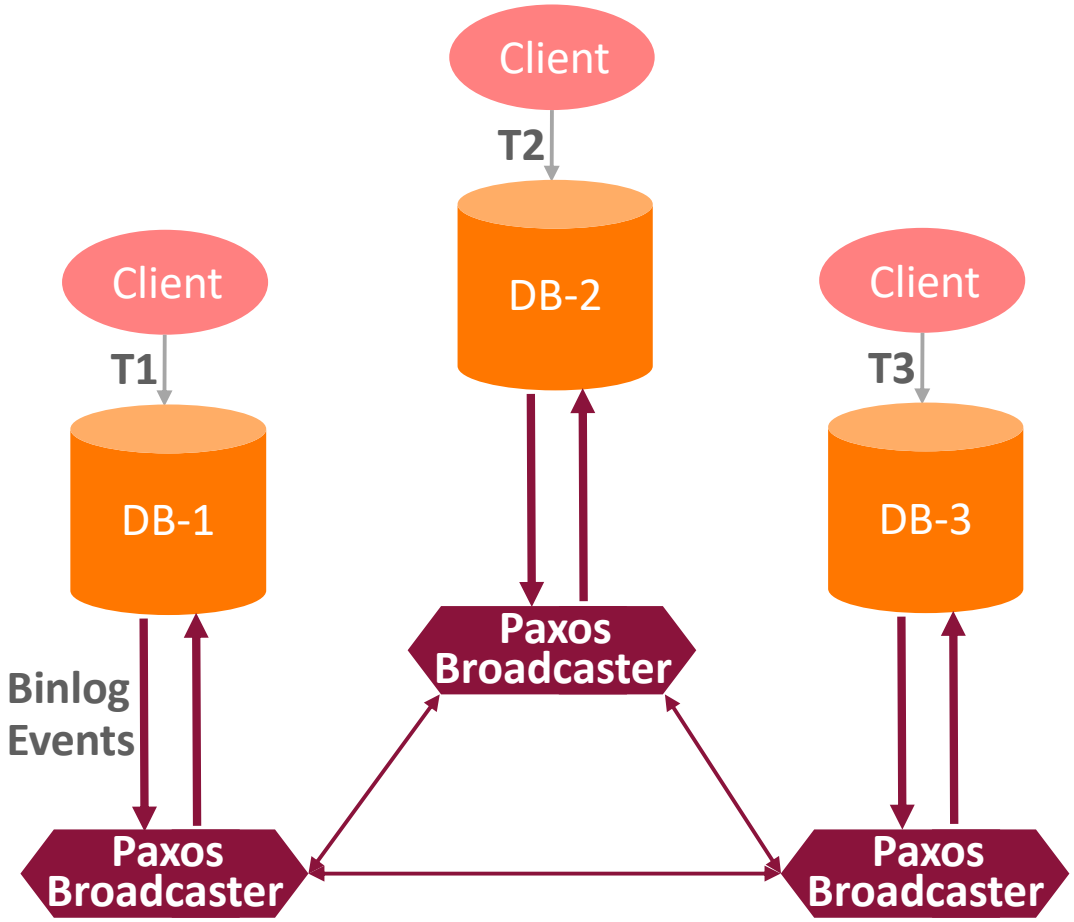
6.1 The Theories of Group Replication

6.2 Group Replication Architecture

6.3 Transaction Life Circle

Transaction Life Circle

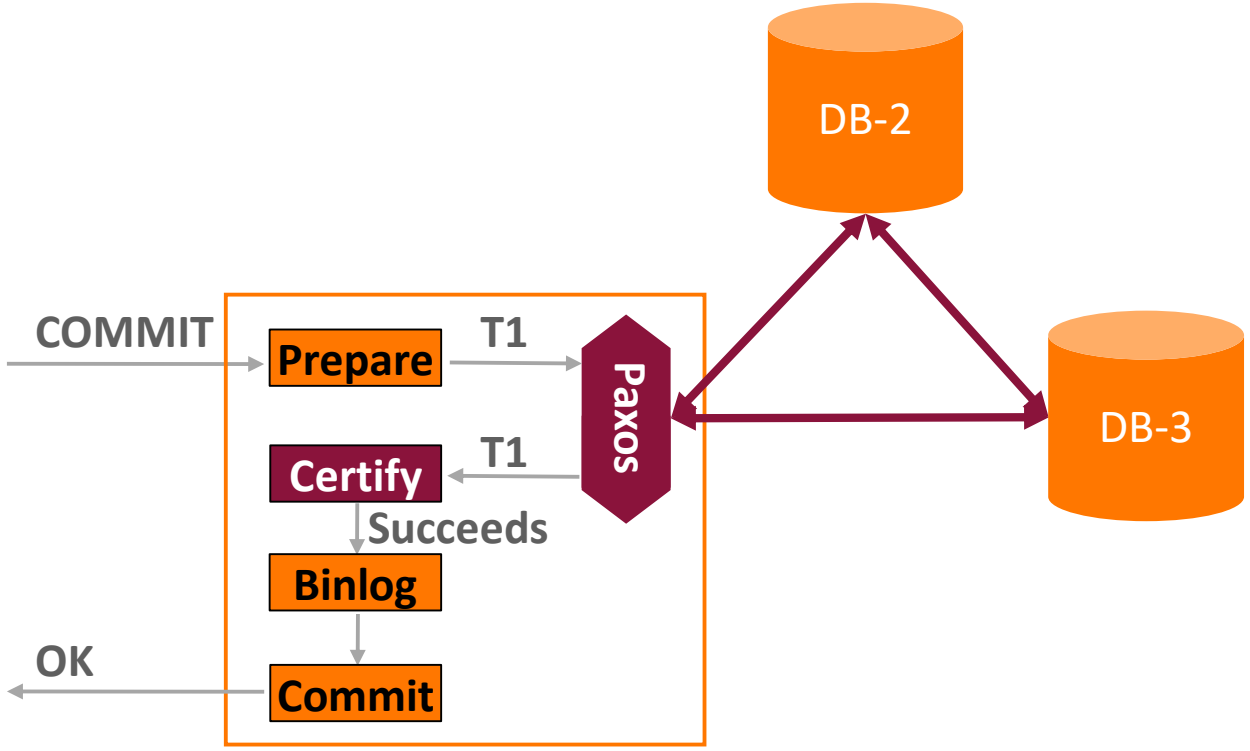
- **Local Transaction**
  - Executed by the client
  - T1 is local transaction on DB-1
- **Remote Transaction**
  - Replicated from other servers
  - T2,T3 are remote transaction on DB-1



MySQL Group Replication Architecture

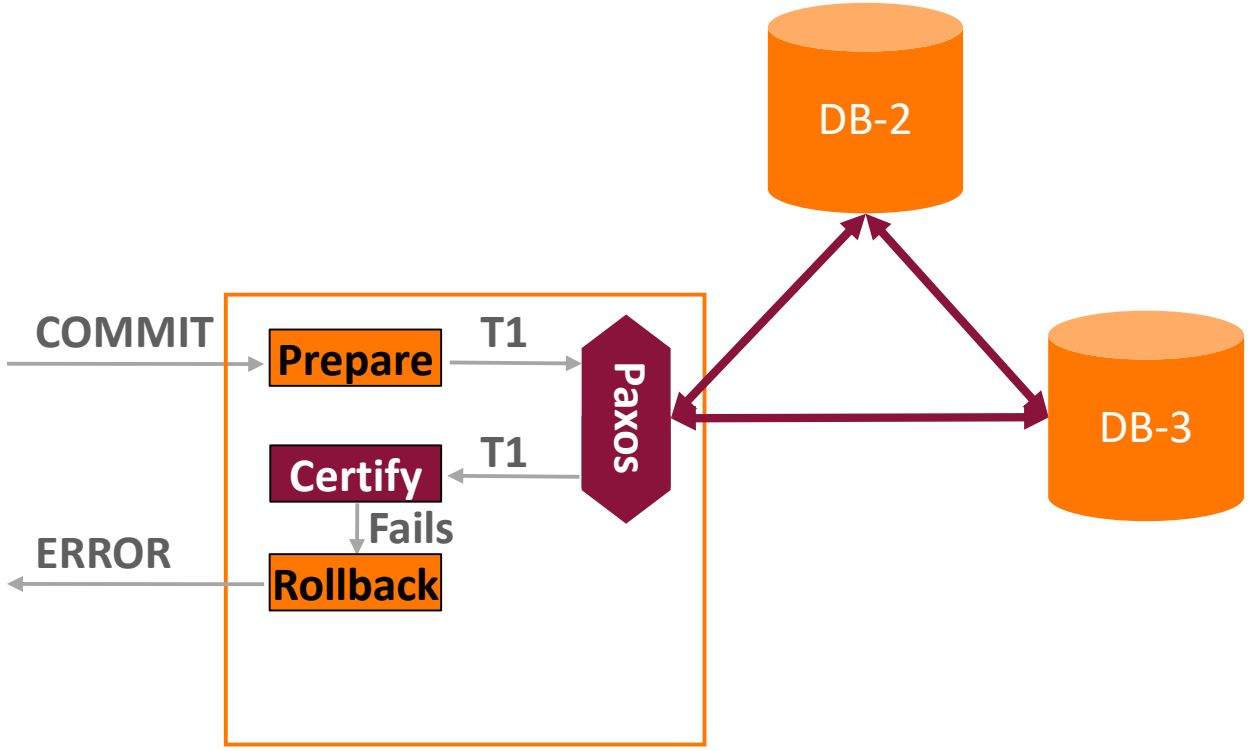
Transaction Life Circle

- Local Transaction Succeeds



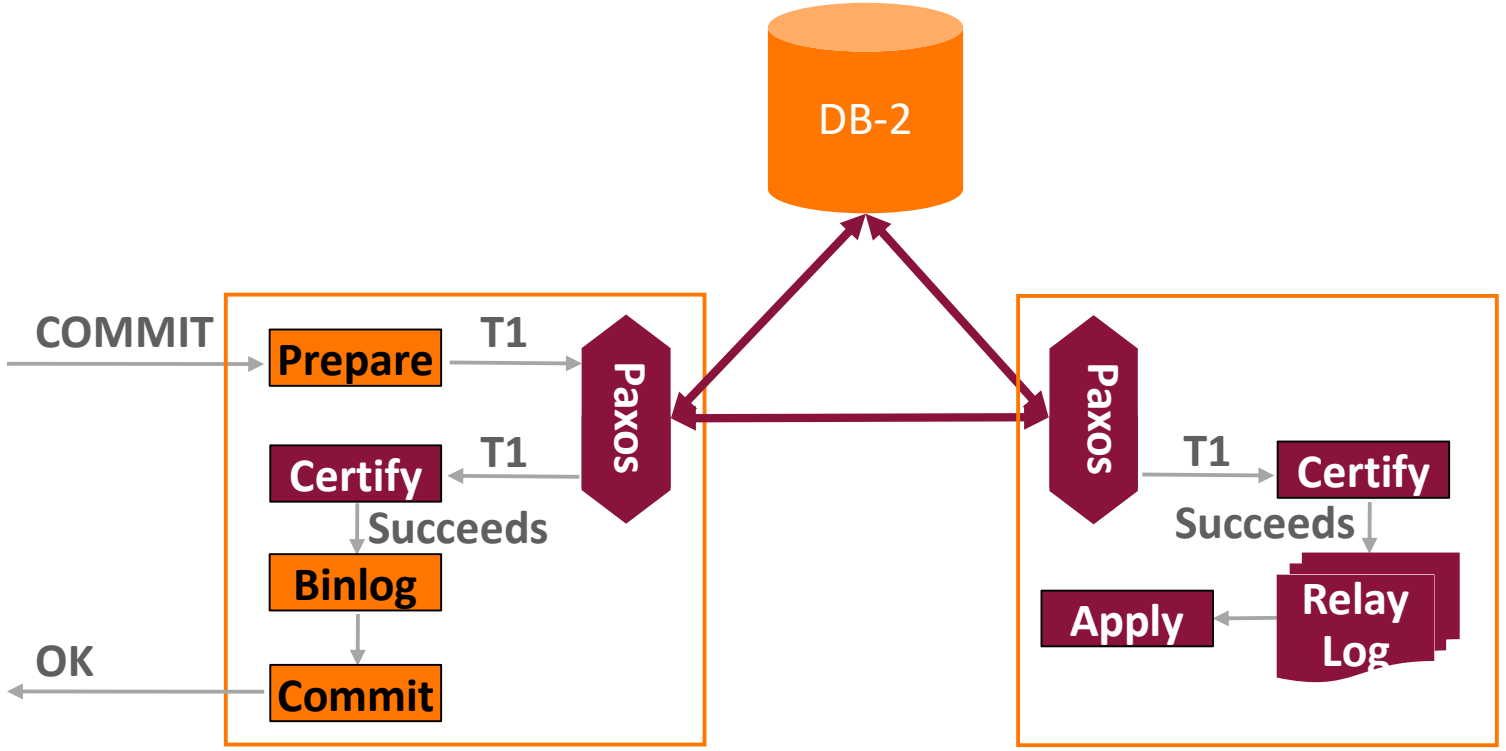
Transaction Life Circle

- Local Transaction Fails



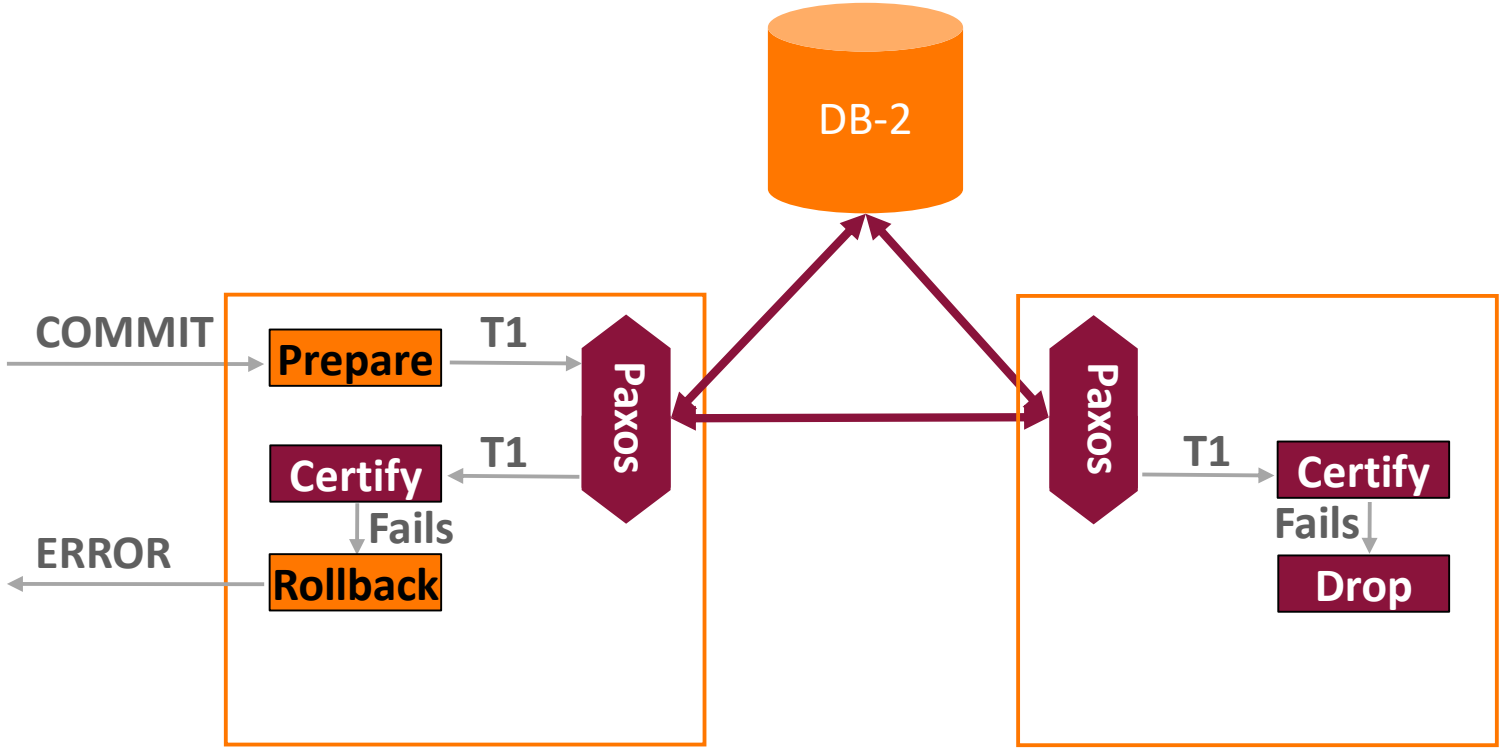
Transaction Life Circle

- Remote Transaction Succeeds



Transaction Life Circle

- Remote Transaction Fails



6 Architecture

6.1 The Theories of Group Replication

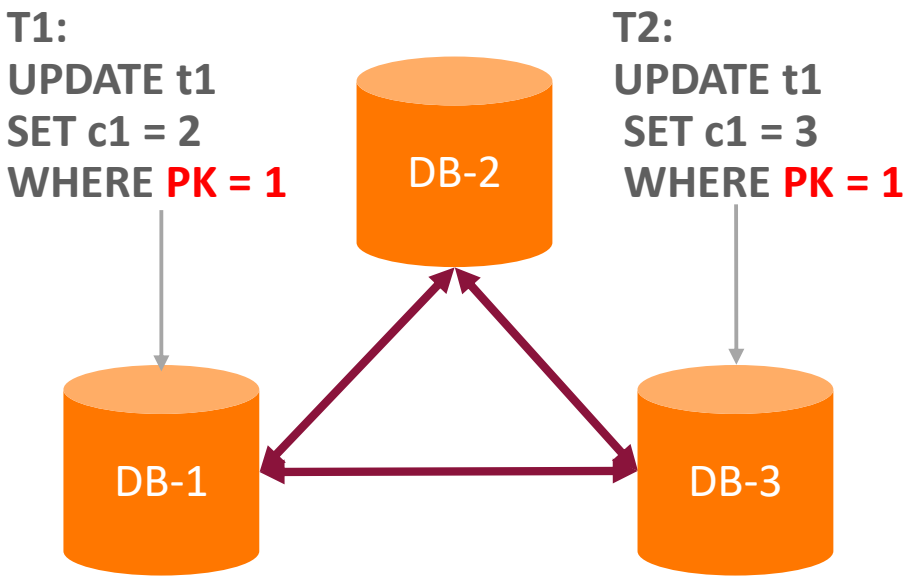
6.2 Group Replication Architecture

6.3 Transaction Life Circle

6.4 Certification(Conflict Detection)

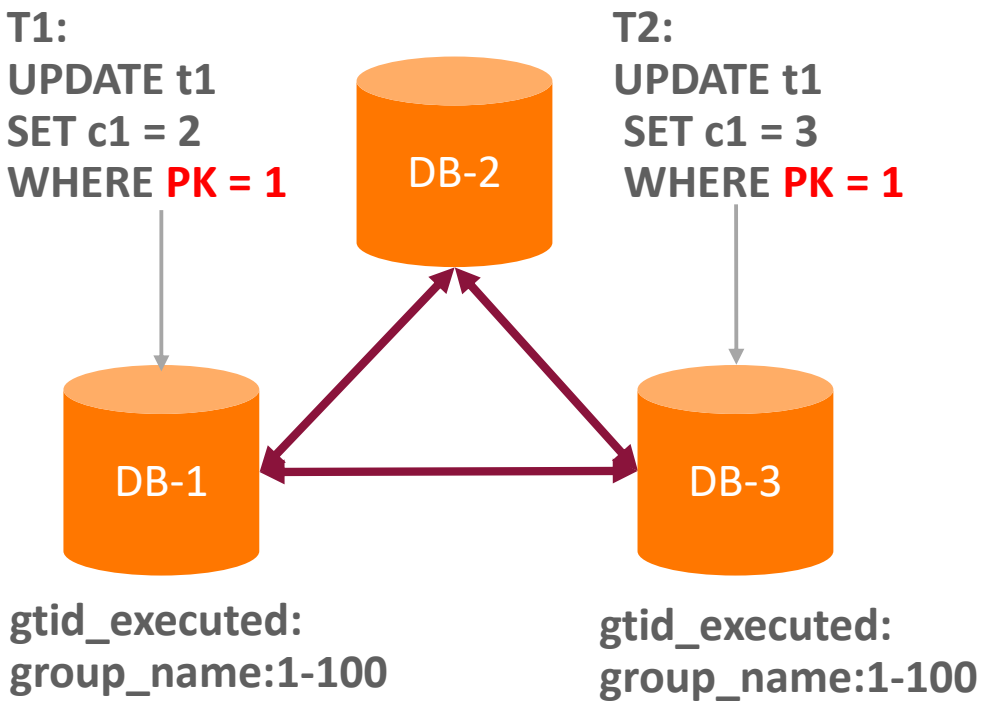
Certification

- Primary Key Based Conflict Detection
  - Possible conflict if two transactions updated same rows
  - Never conflict if two transactions updated different rows
  - Don’t support DDL yet



Certification

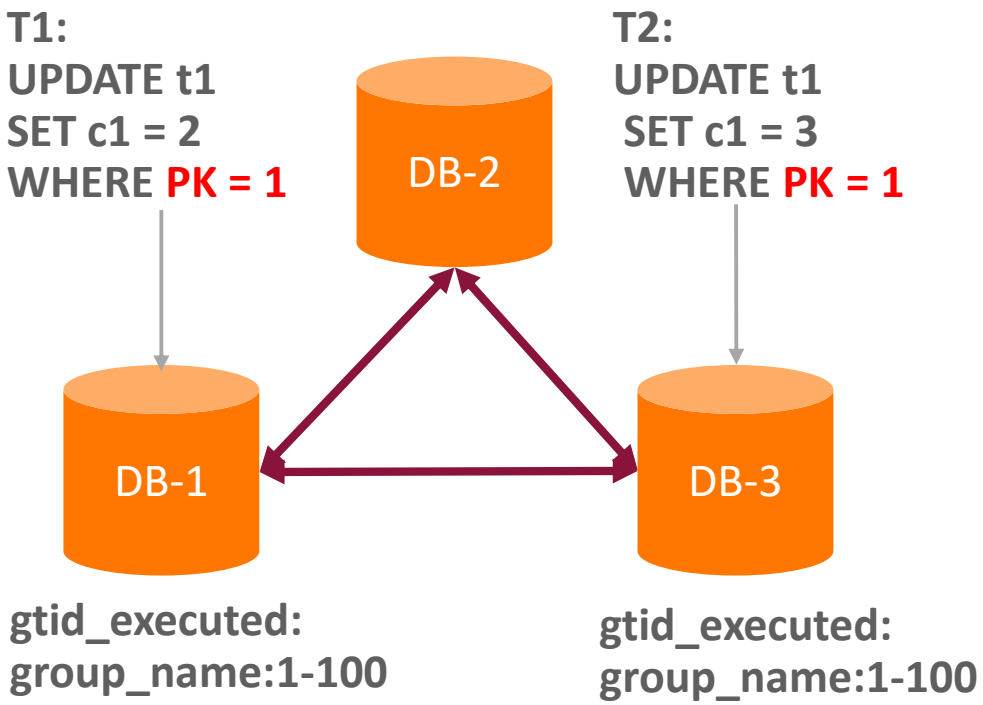
- Snapshot Version
  - Snapshot version is the value of gtid_executed variable before broadcasting the transaction
  - Snapshot of certifying transaction must include the GTIDs of the certified transactions which updated the same rows



Certification

- Conflict Detection Database
  - A list of certified transactions' PK HASH and GTID set pairs.
  - Every member has one.

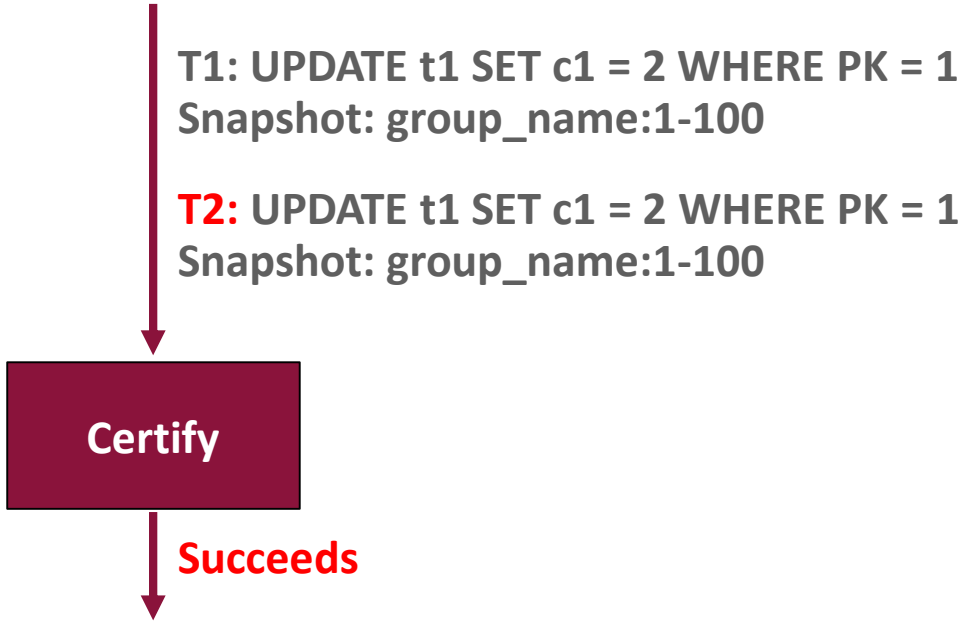
PK HASH	GTID SET
…	
db1.t1.pk=1	group_name:1-50
…	



Certification

- Conflict Detection
  - T2's snapshot includes the GTID set in conflict detection database, so certification succeeds.

    group_name:1-100 > group_name:1-50

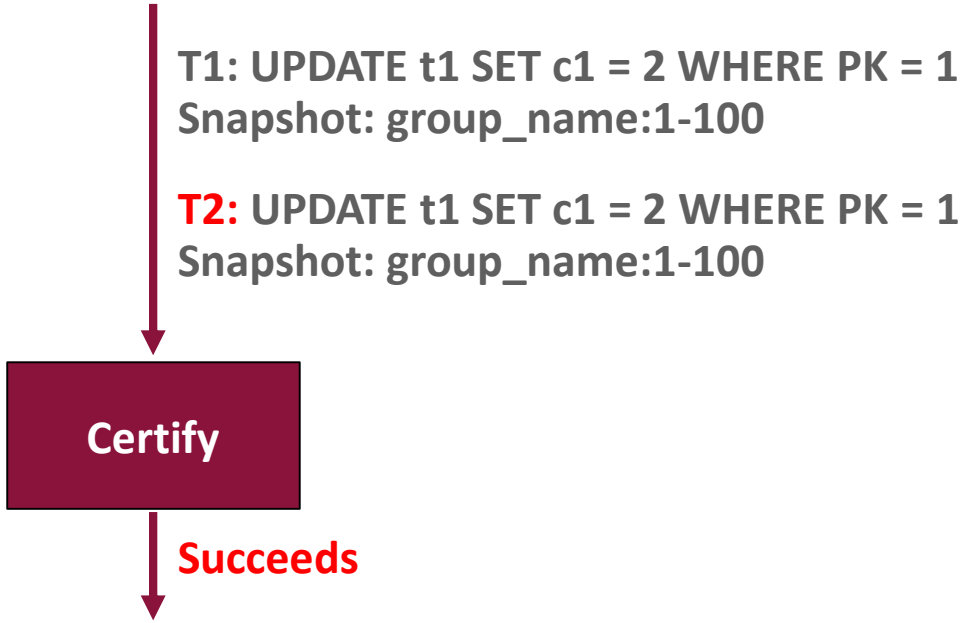


PK HASH	GTID SET
...	
db1.t1.pk = 1	group_name:1-50
...	

Certification

- Conflict Detection
  - T2's snapshot includes the GTID set in conflict detection database, so certification succeeds.

    group_name:1-100 > group_name:1-50
- Update Detection Database
  - Fill GTID SET with T2's snapshot and GTID
  - Suppose T2's GTID is group_name:101

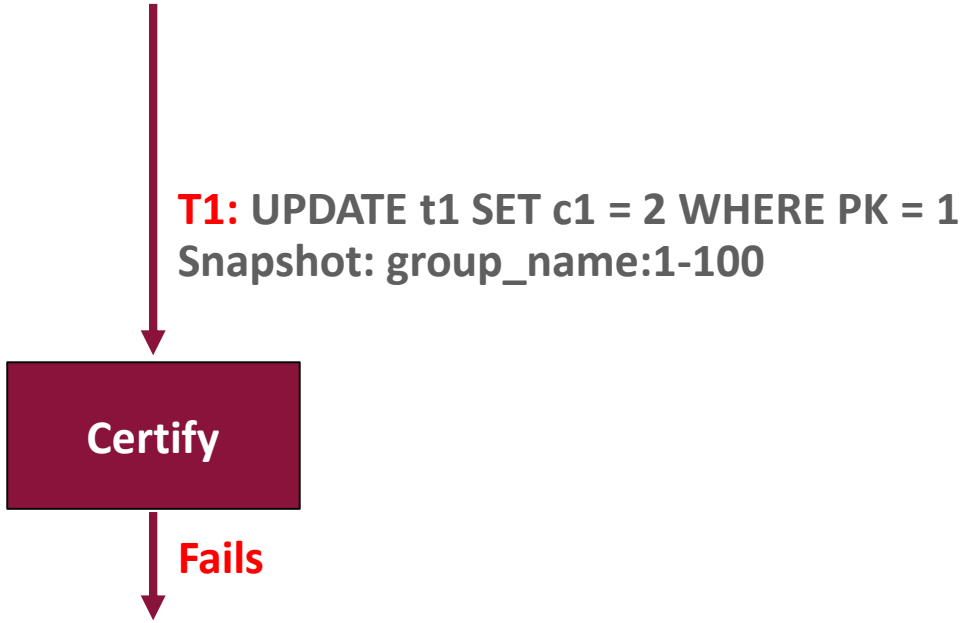


PK HASH	GTID SET
...	
db1.t1.pk = 1	group_name:1-101
...	

Certification

- Conflict Detection
  - T1's snapshot do NOT include the GTID set in conflict detection database, so certification fails.

    group_name:1-100 < group_name:1-101



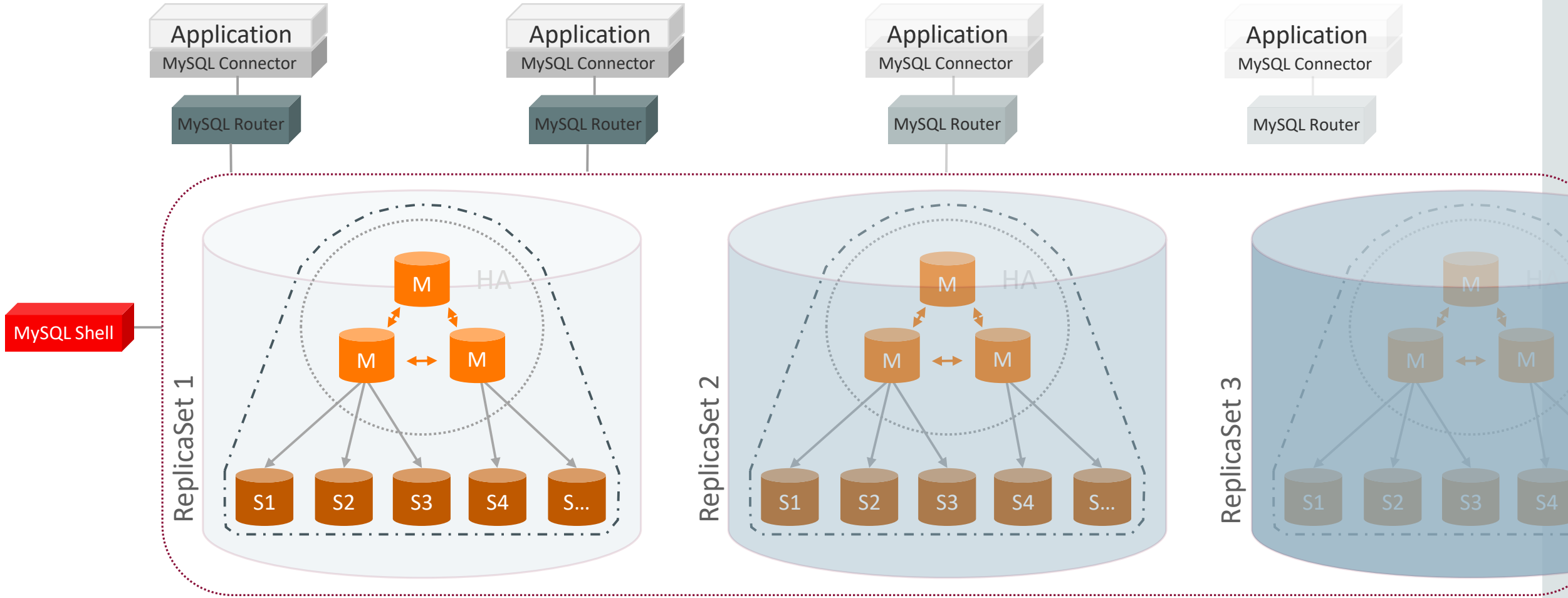
PK HASH	GTID SET
...	
db1.t1.pk = 1	group_name:1-101
...	

7 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-primary update everywhere and a dependable MySQL service.
- Plugin **GA version** available with MySQL 5.7.17+, available on 8.0.1+

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
  - http://dev.mysql.com/doc/refman/8.0/en/group-replication.html
- Blogs from the Engineers (news, technical information, and much more)
  - http://mysqlhighavailability.com


ORACLE®