Galera Cluster

Release 3 New Features

Seppo Jaakola, Alex Yurchenko
Codership Oy

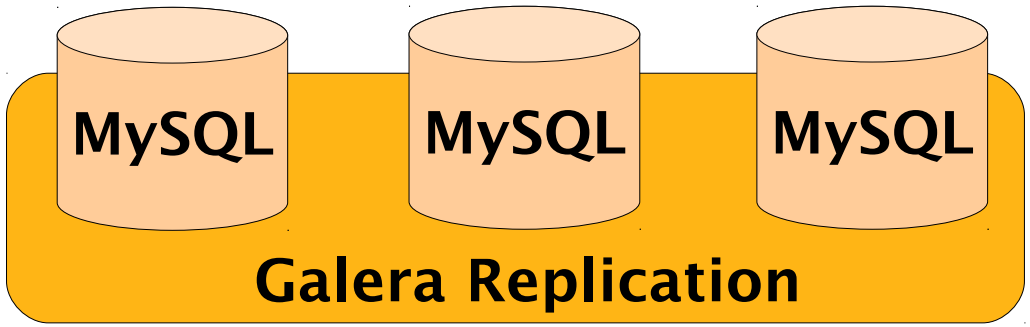
Agenda

1. Galera Cluster for MySQL
2. Release 3 New Features:
   - WAN Replication
   - 5.6 GTID Support
   - MySQL Replication Support
   - etc.
3. Galera Project

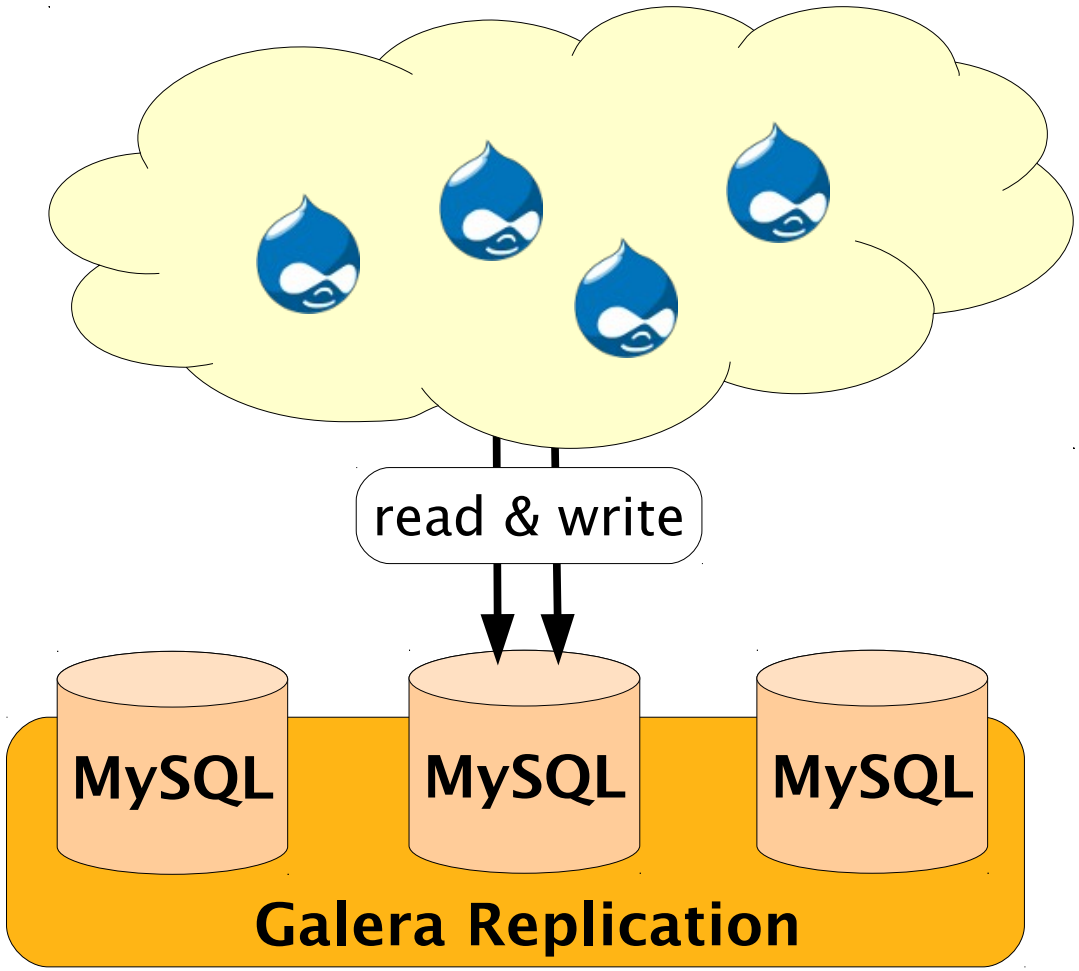
Galera Cluster

< No SPOF

all nodes are equivalent and fully representative of the cluster



Galera Cluster



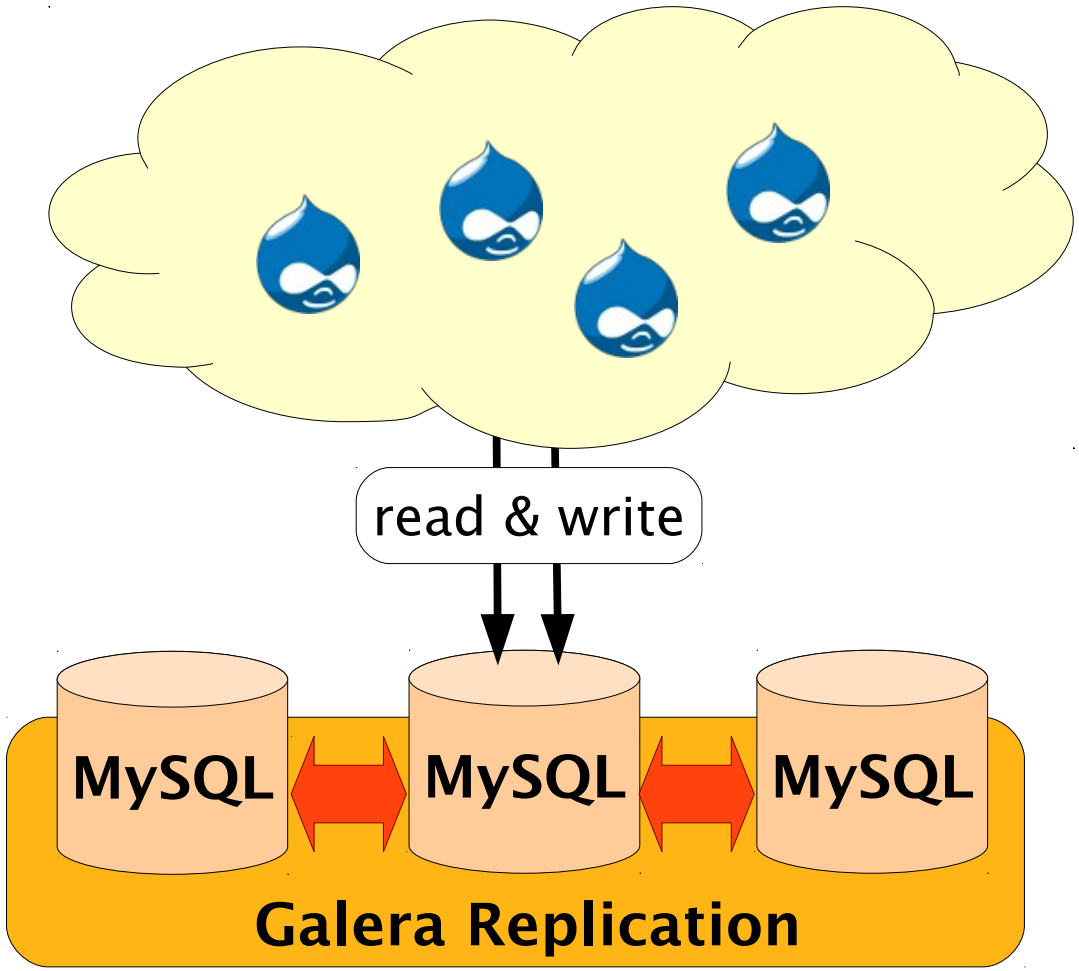
< No SPOF

all nodes are equivalent and fully representative of the cluster

< Transparent

clients connect directly to server

Galera Cluster



< No SPOF

all nodes are equivalent and fully representative of the cluster

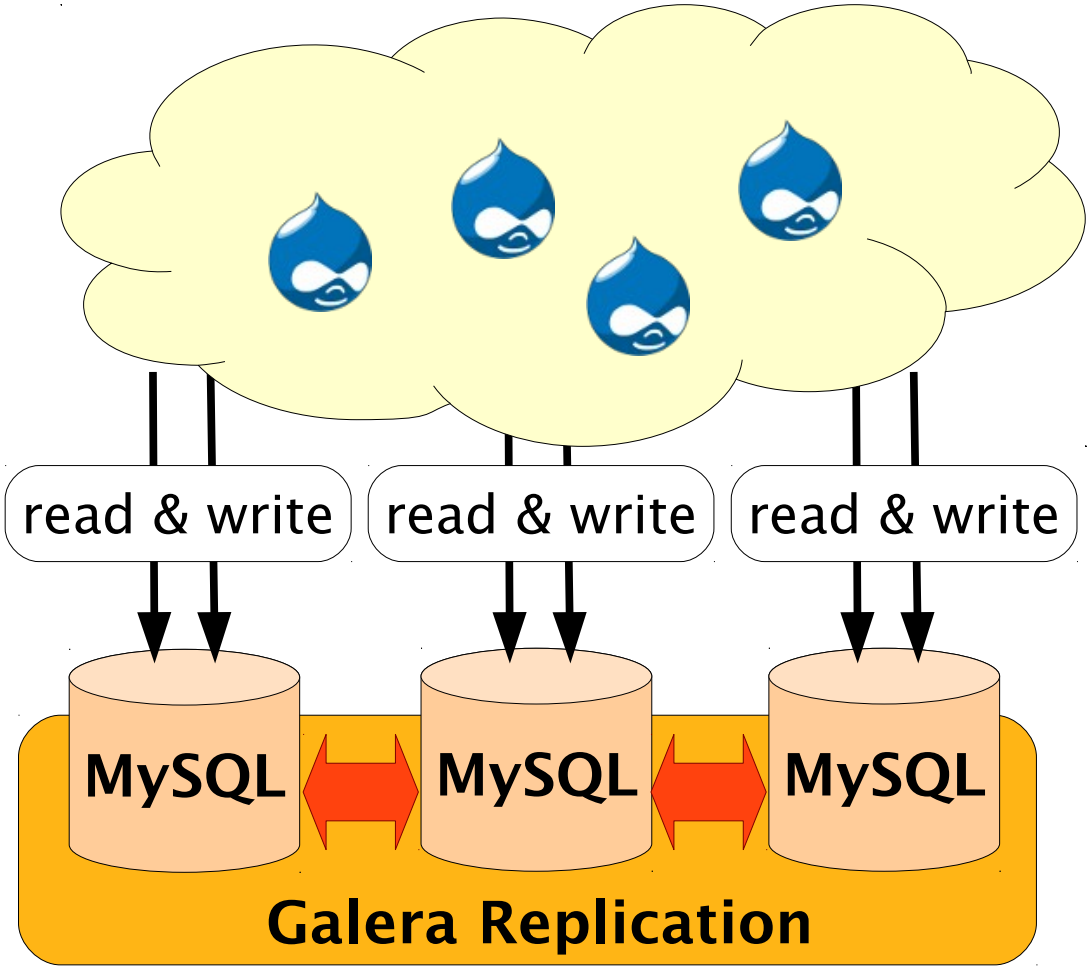
< Transparent

clients connect directly to server

< Synchronous

transaction is either committed on every node or not at all

Galera Cluster



< No SPOF

all nodes are equivalent and fully representative of the cluster

< Transparent

clients connect directly to server

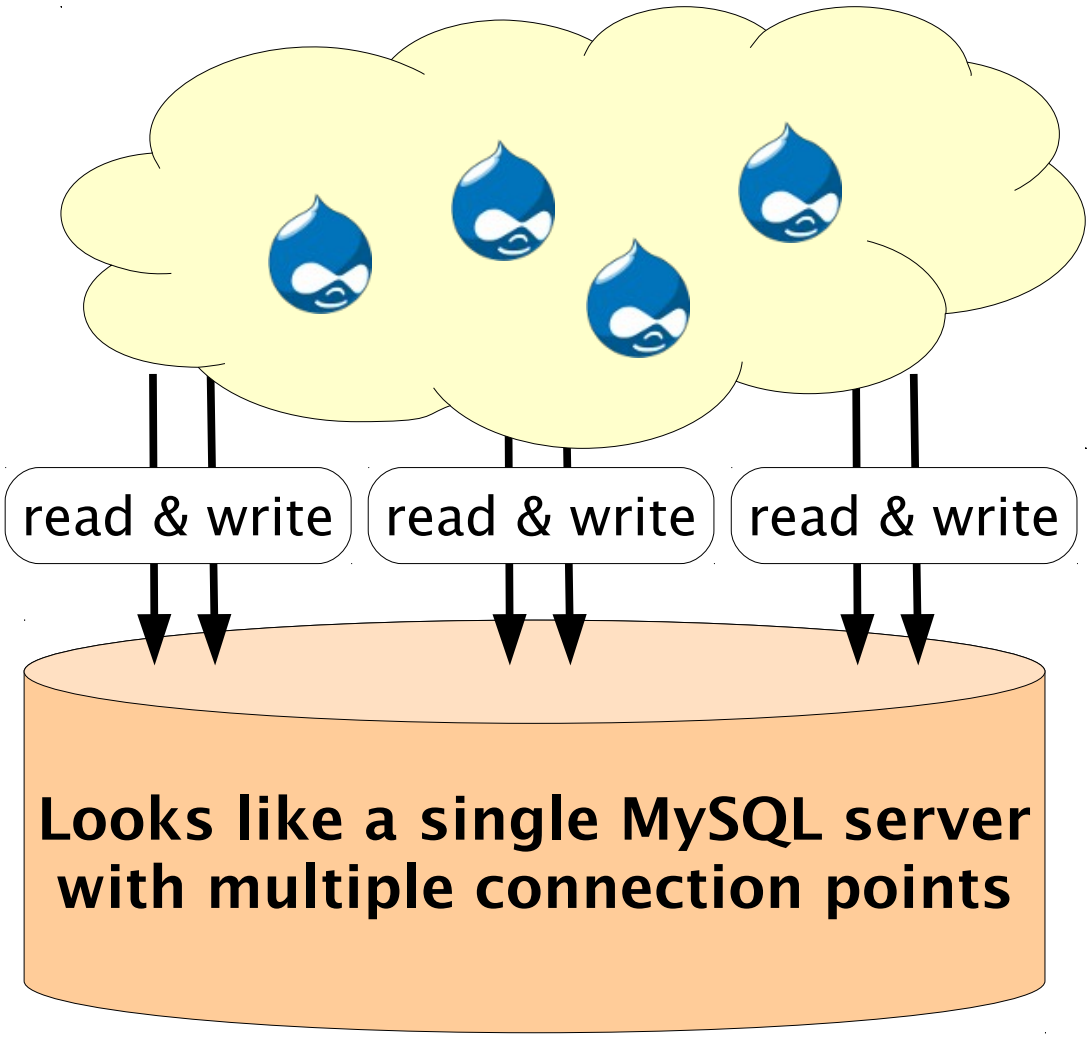
< Multi-master

unrestricted writes to every node

< Synchronous

transaction is either committed on every node or not at all

Galera Cluster



< No SPOF

all nodes are equivalent and fully representative of the cluster

< Transparent

clients connect directly to server

< Multi-master

unrestricted writes to every node

< Synchronous

transaction is either committed on every node or not at all

Also...

- Parallel slave applying
- Practically no slave lag
- Instant trivial failover
- Automatic node provisioning
- Works great in WAN

How is that possible?

"Cluster" vs. "replication"

- Global context:
  - Membership
  - Global State
  - Global Transaction ID

How is that possible? Limitations:

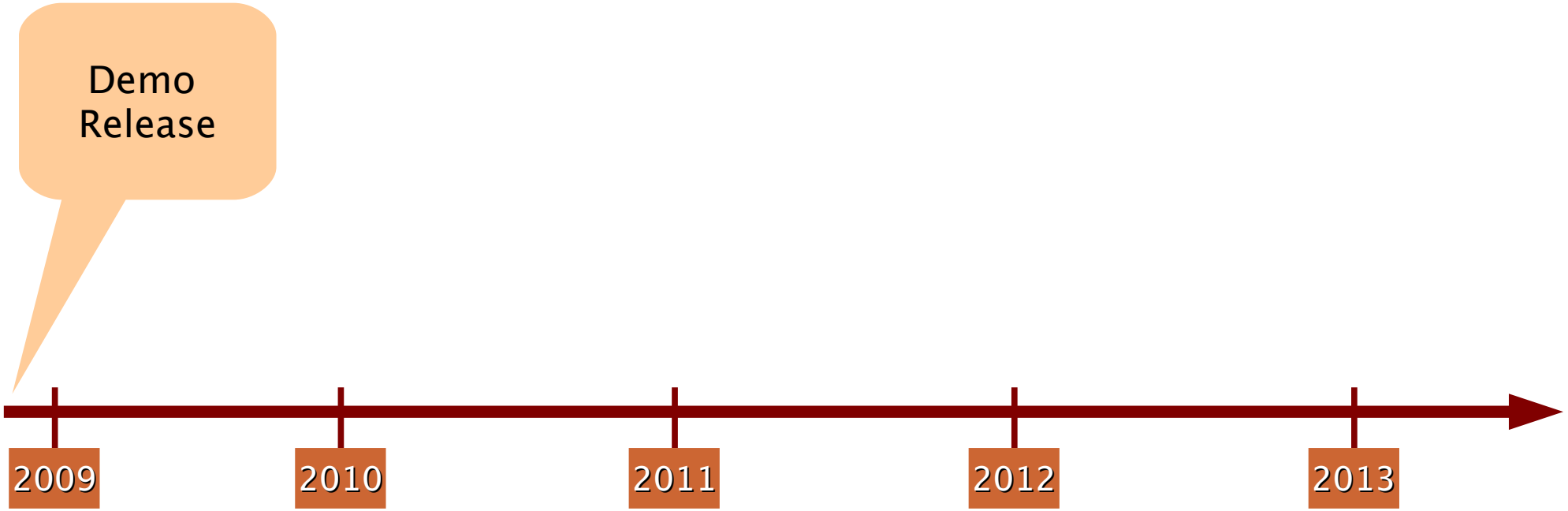
- InnoDB only
- Primary Keys is a must
- Commit latency
- Doesn't like huge transactions (max 2GB)
- DEADLOCK on COMMIT
- ...

Galera Cluster Releases

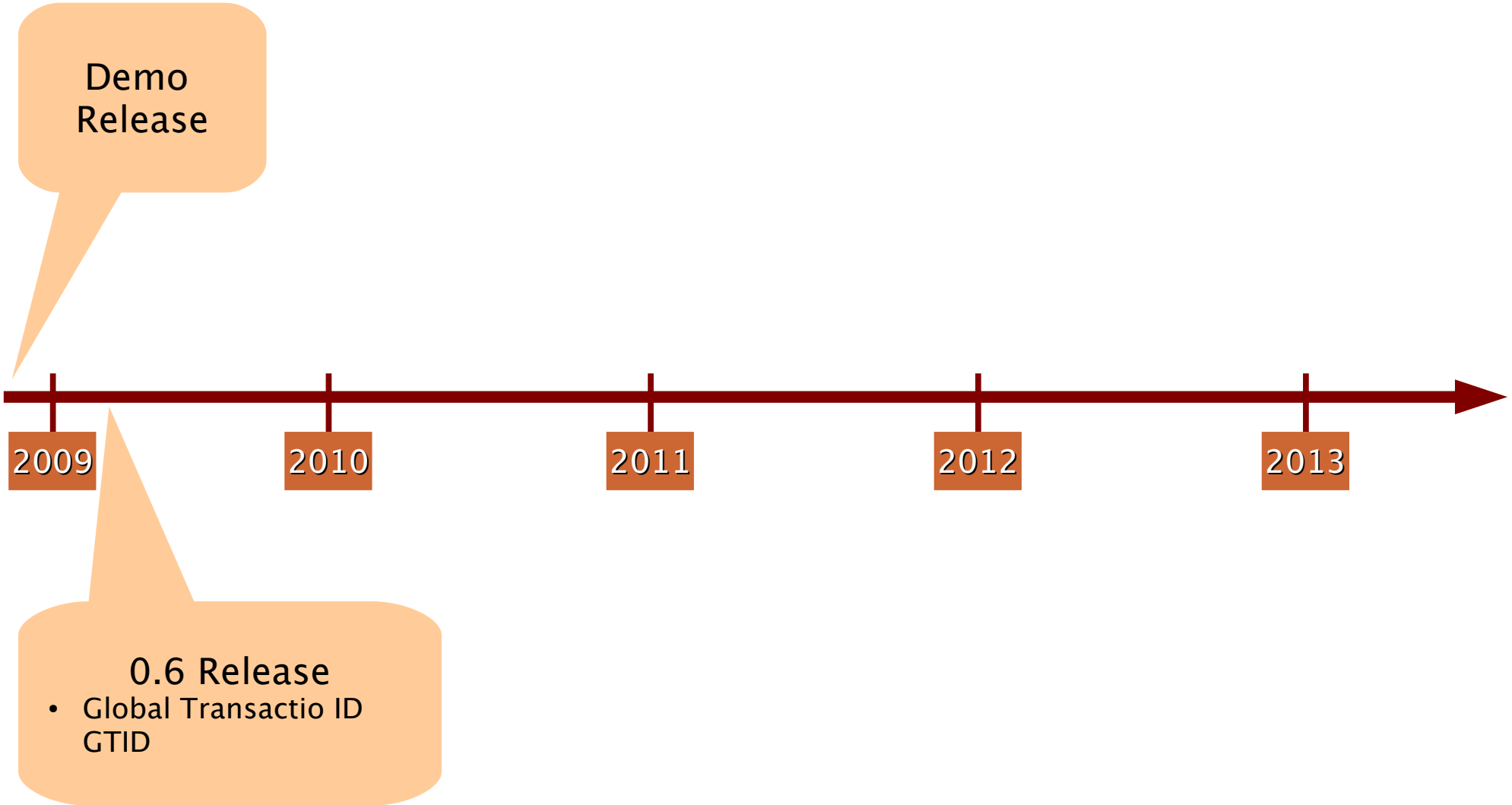
Galera projects:

- Universal Replication Interface (wsrep API)
  - Defines the interface between DBMS and replication plugin
  - Versioned by API level #1-25. Galera 3.1 is using #25
- Galera Replication Plugin
  - Latest API #23 version is 2.7
  - And API #25 version is 3.1
- Replication API implementation in MySQL server
  - 5.5.34-25.9 for MySQL 5.5
  - 5.6.14-25.1 for MySQL 5.6
- All open-source on Launchpad

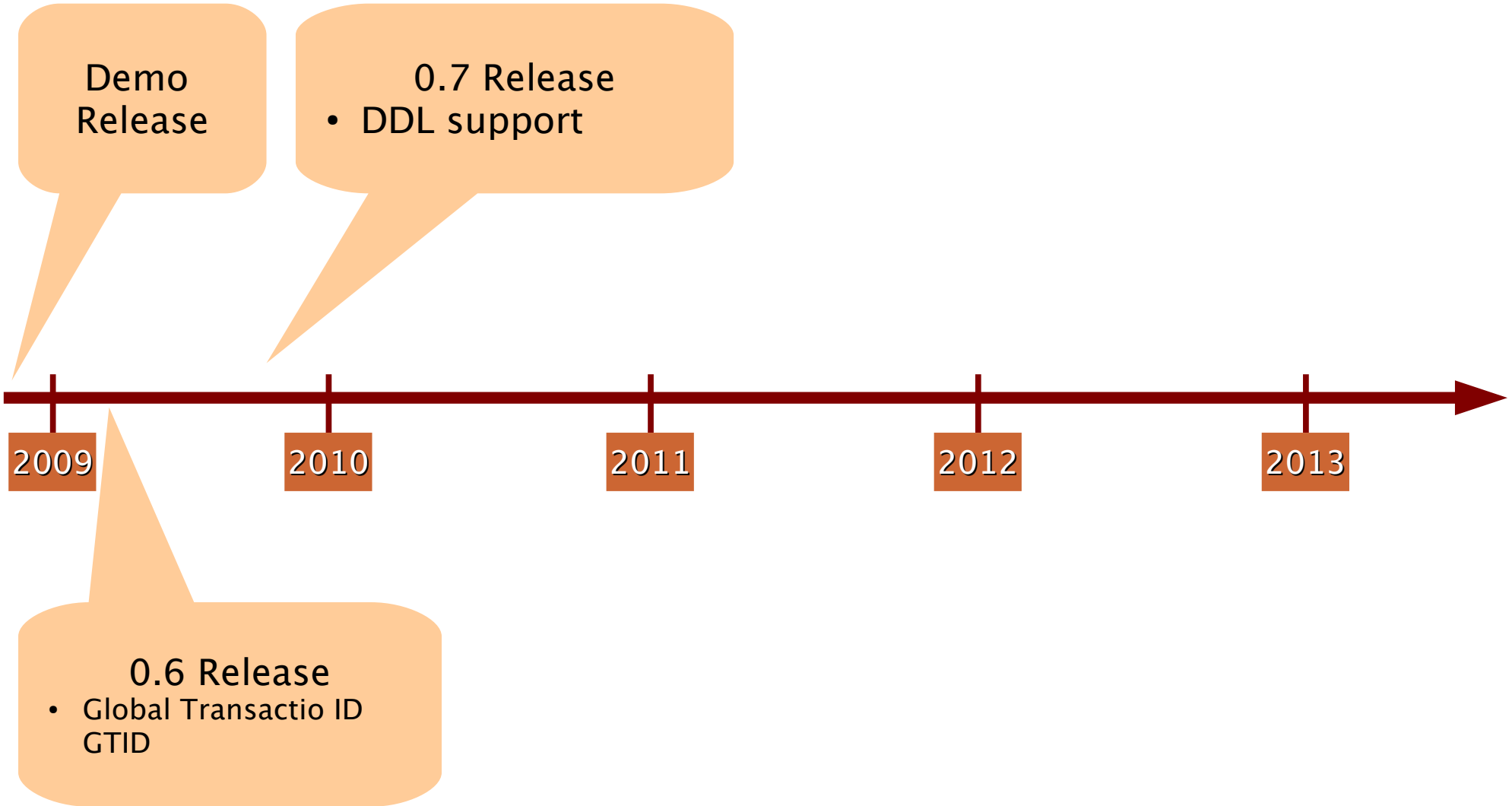
Roadmap to 3.0



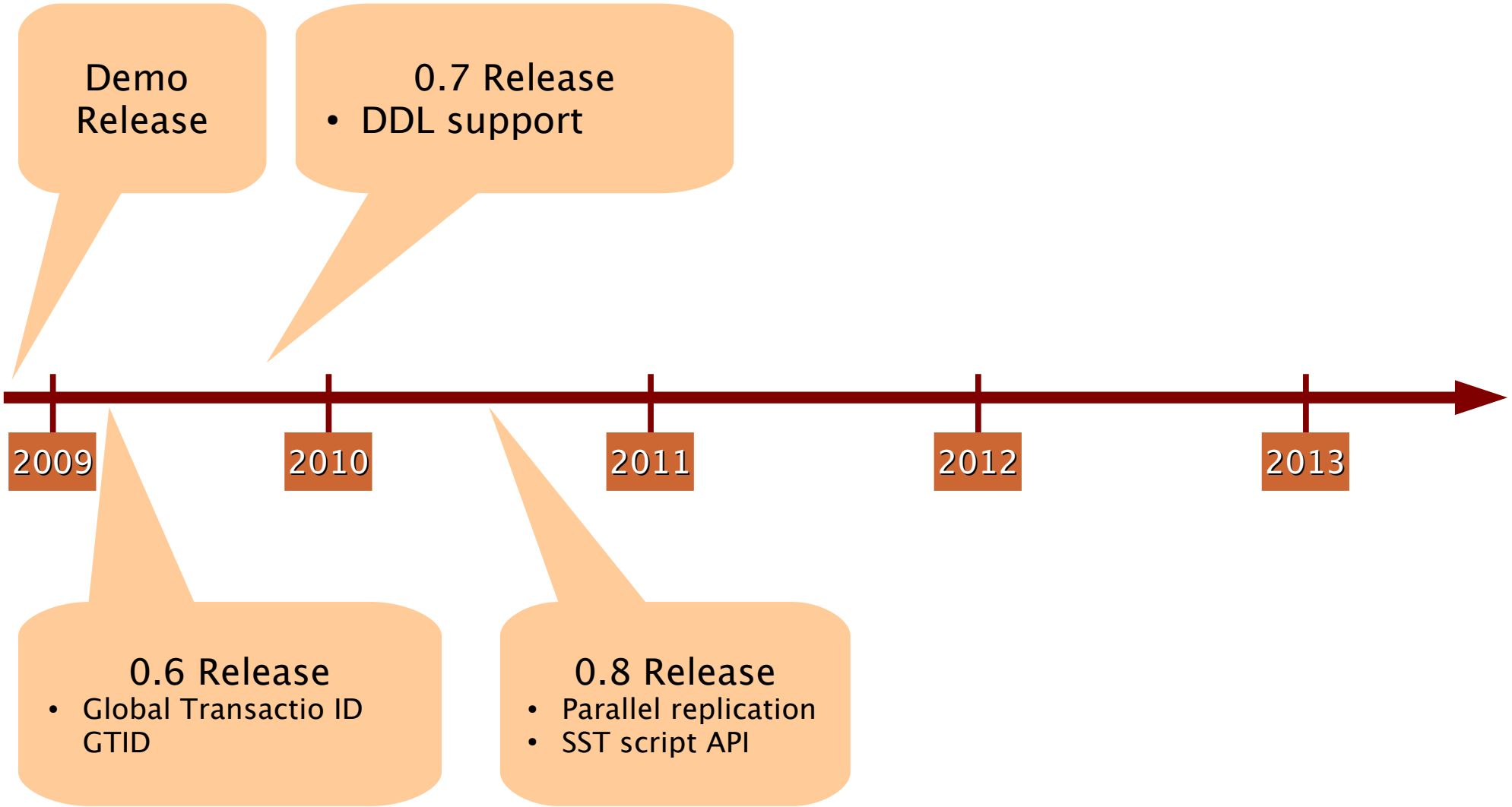
Roadmap to 3.0



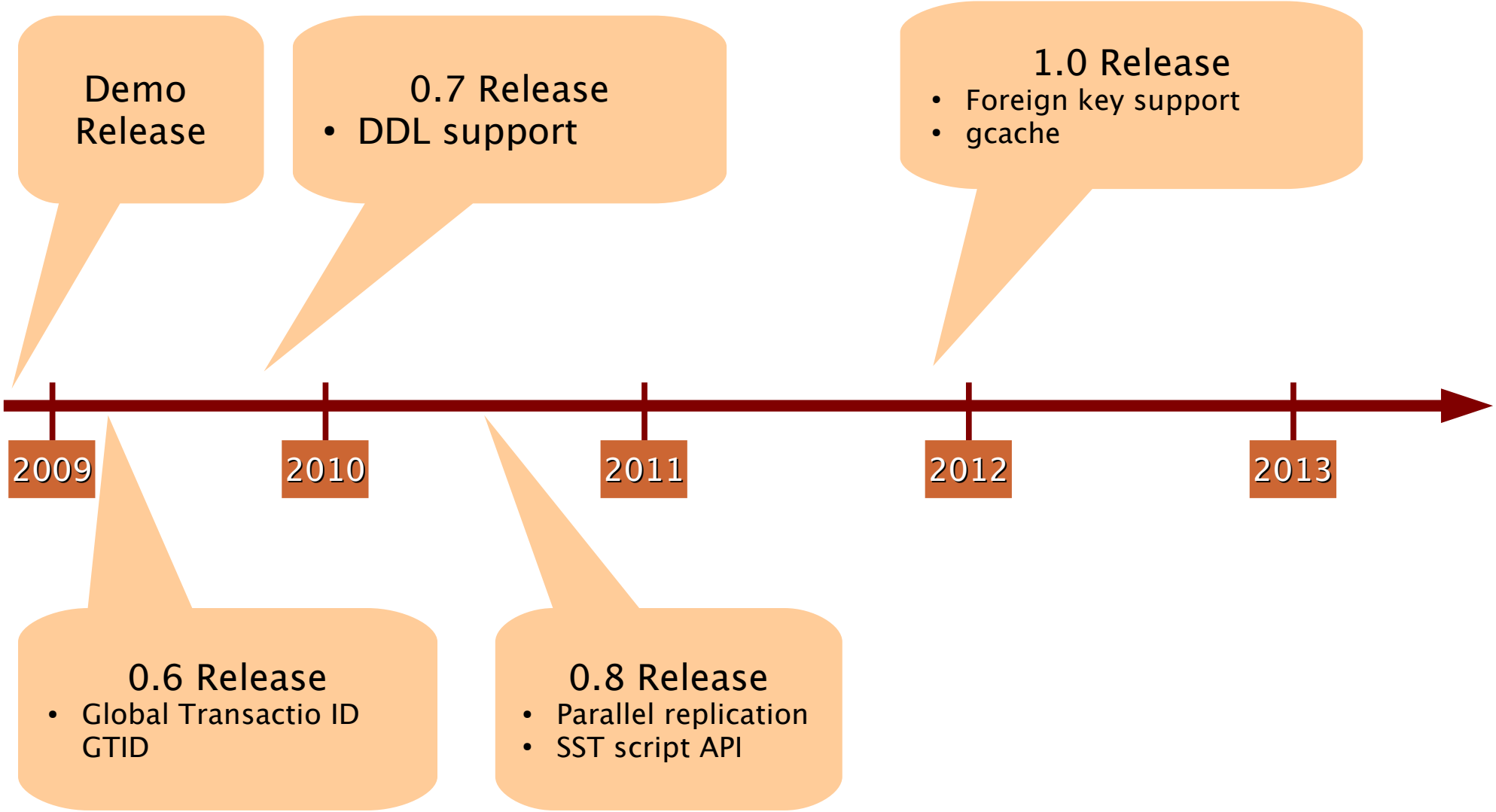
Roadmap to 3.0



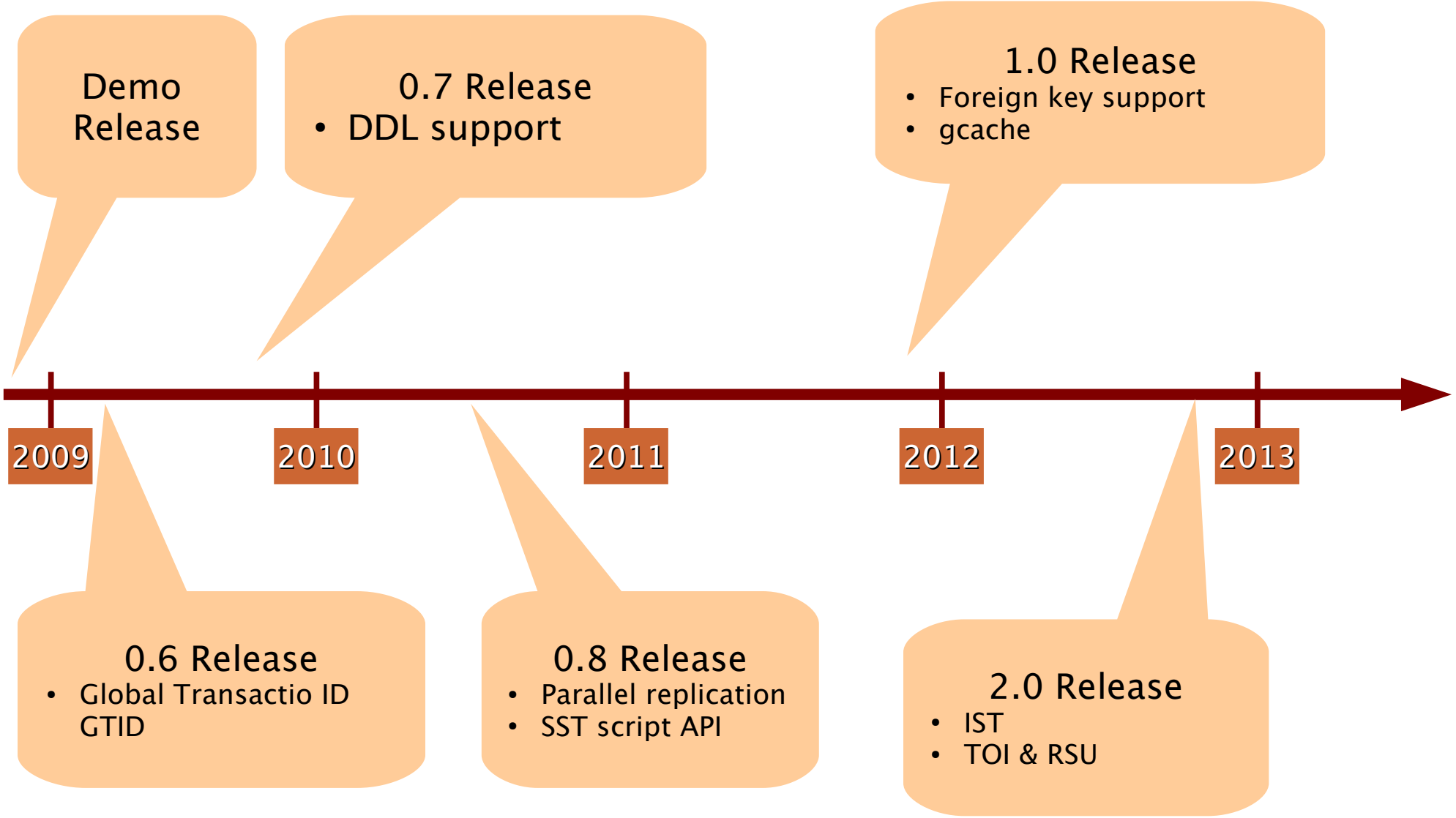
Roadmap to 3.0



Roadmap to 3.0



Roadmap to 3.0



Galera Cluster 3

Galera Cluster 3

Galera Cluster 3.1 GA released Nov 12

(Galera 3.1, MySQL-wsrep 5.6.14)

Focus on:

- WAN replication
- Asynchronous replication topologies

Galera Cluster 3 (Galera-3.x, MySQL-5.6.x)

Featuring:

- MySQL 5.6 Support:
  - Support for MySQL 5.6 GTID
  - Native replication can be interleaved with Galera replication
- Optimization for WAN replication:
  - Cluster can be divided in segments to indicate network proximity
- New writeset format:
  - Optimized certification key format
  - 128-bit checksums
  - Direct IO cache → writeset copy.
- A number of bug fixes and minor improvements:
  - Hardware-accelerated CRC32 on network packets, etc.

MySQL 5.6 Support

MySQL 5.6 Support

Both MySQL-wsrep 5.5 and 5.6 have support for Galera 3.x

From Galera perspective, it does not matter much what the DBMS is like, as long as it supports transactions.

In MySQL 5.6 interesting new features are:

- MySQL GTID
- InnoDB full text search
- Overall performance improvements

MySQL 5.6 Support

- We will also support MariaDB 10, work is ongoing.
- Percona XtraDB Cluster will also have 5.6 based version coming soon.

MySQL 5.6 Support

- Support for MySQL 5.1 is dropped.
- Future releases are for MySQL 5.5 and 5.6 only.
- Now can be built on Linux, Solaris, FreeBSD, MacOS.

MySQL 5.6 GTID Support

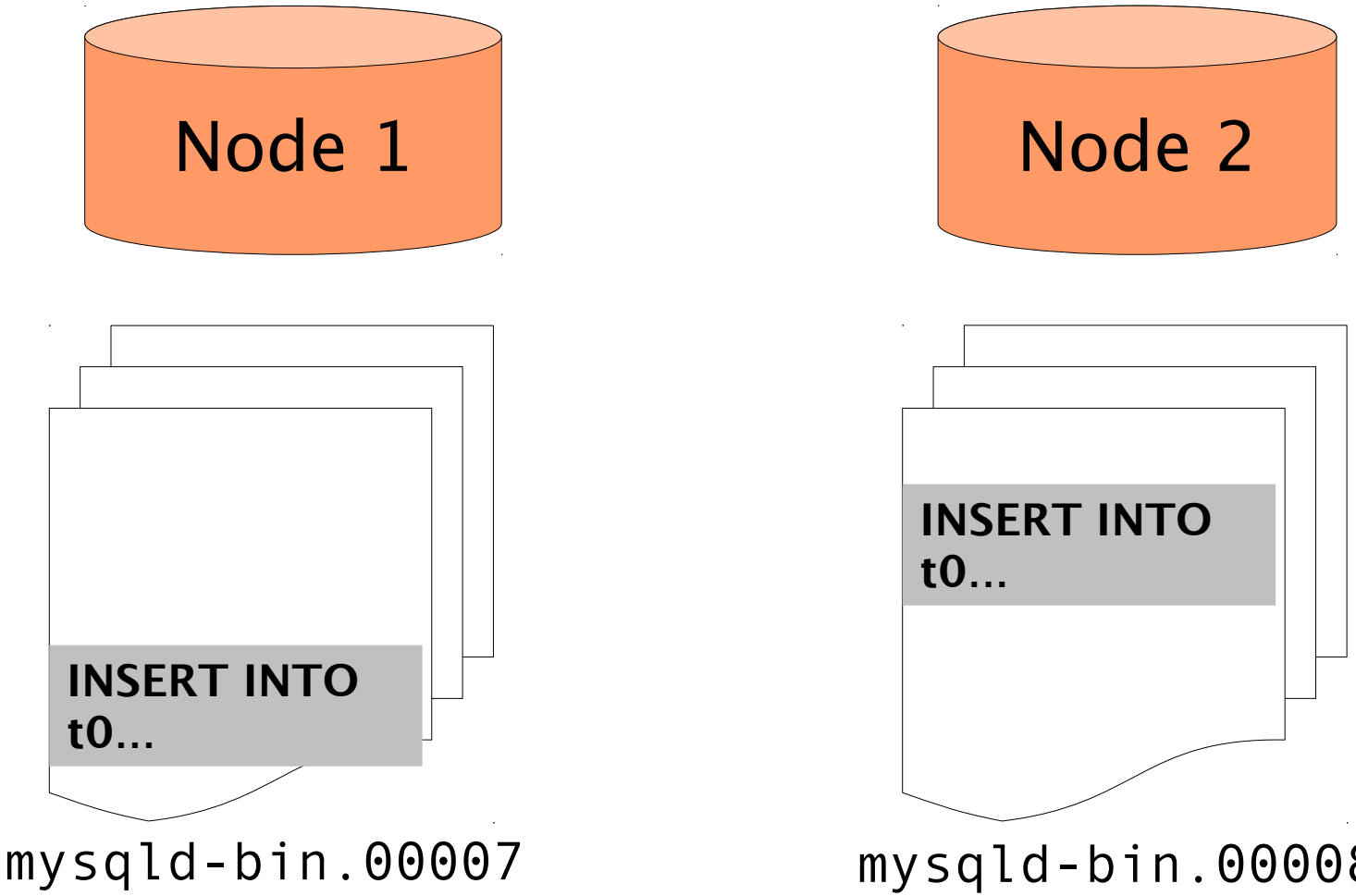
GTID Support

MySQL 5.6 introduces global transaction identifier (GTID), which can greatly ease up the MySQL replication master fail over.

In Galera Cluster, all nodes will generate different binlog files:

- Binlog events are same and in same order, but binlog file offsets may vary
- Galera community has developed miracles to cope with this, it is amazing what a single line of perl code can do

Galera Binlogging



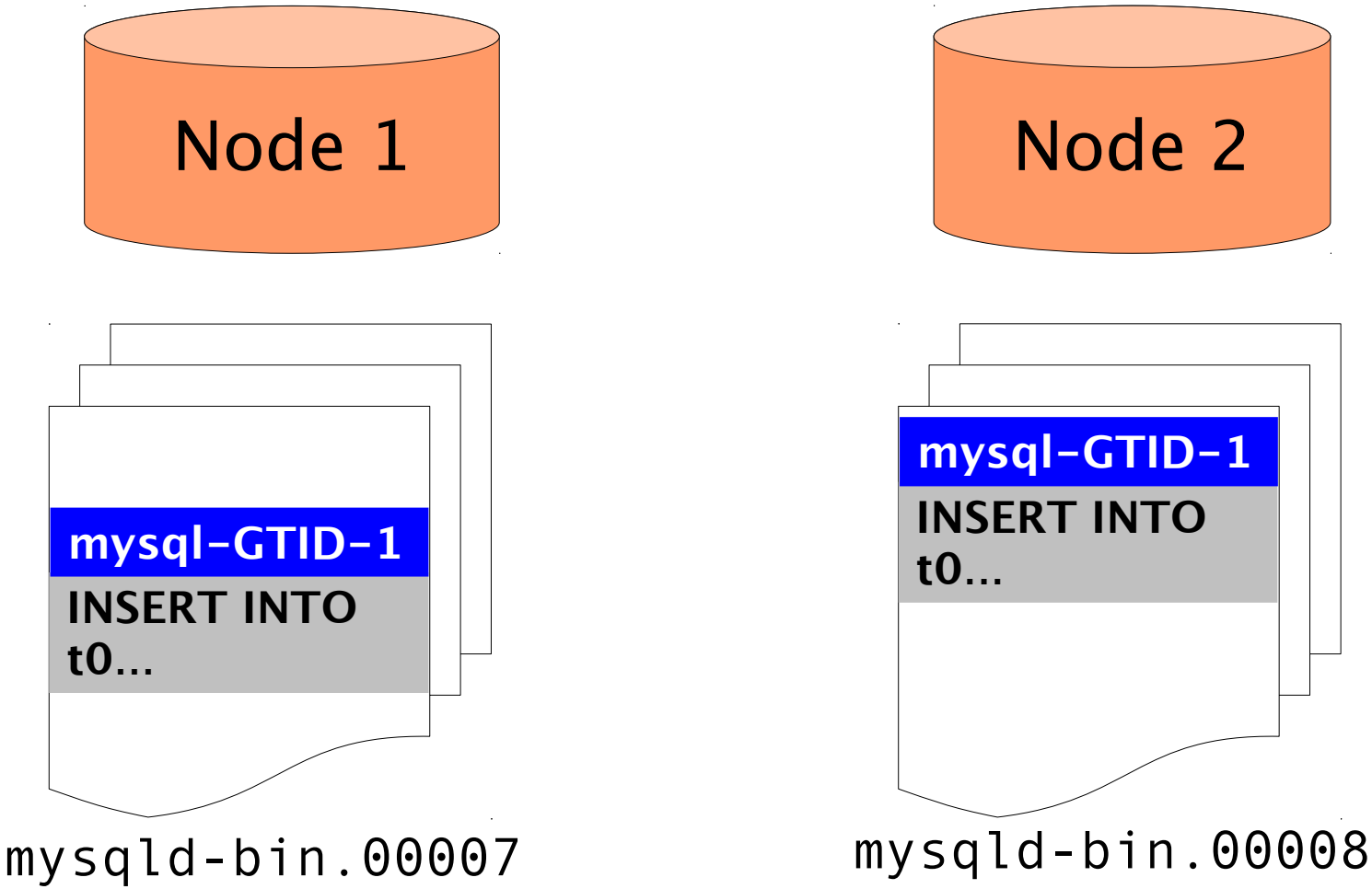
GTID Support

Galera will preserve the MySQL GTID events both in “galera replicated mysql slave” and “pre-ordered replication” methods.

Galera nodes will generate Galera Cluster GTID for Mysql replication. MySQL slaves will see the Galera Cluster as one MySQL master.

Master fail over in Galera Cluster can happen by using the GTID.

Galera Binlogging



Async Replication in/out Galera Cluster

Galera + MySQL Replication

Galera Cluster can operate both as MySQL master and MySQL slave, but

- MySQL master fail over in Galera Cluster has been challenging until now
- MySQL Slave operation has performance bottleneck

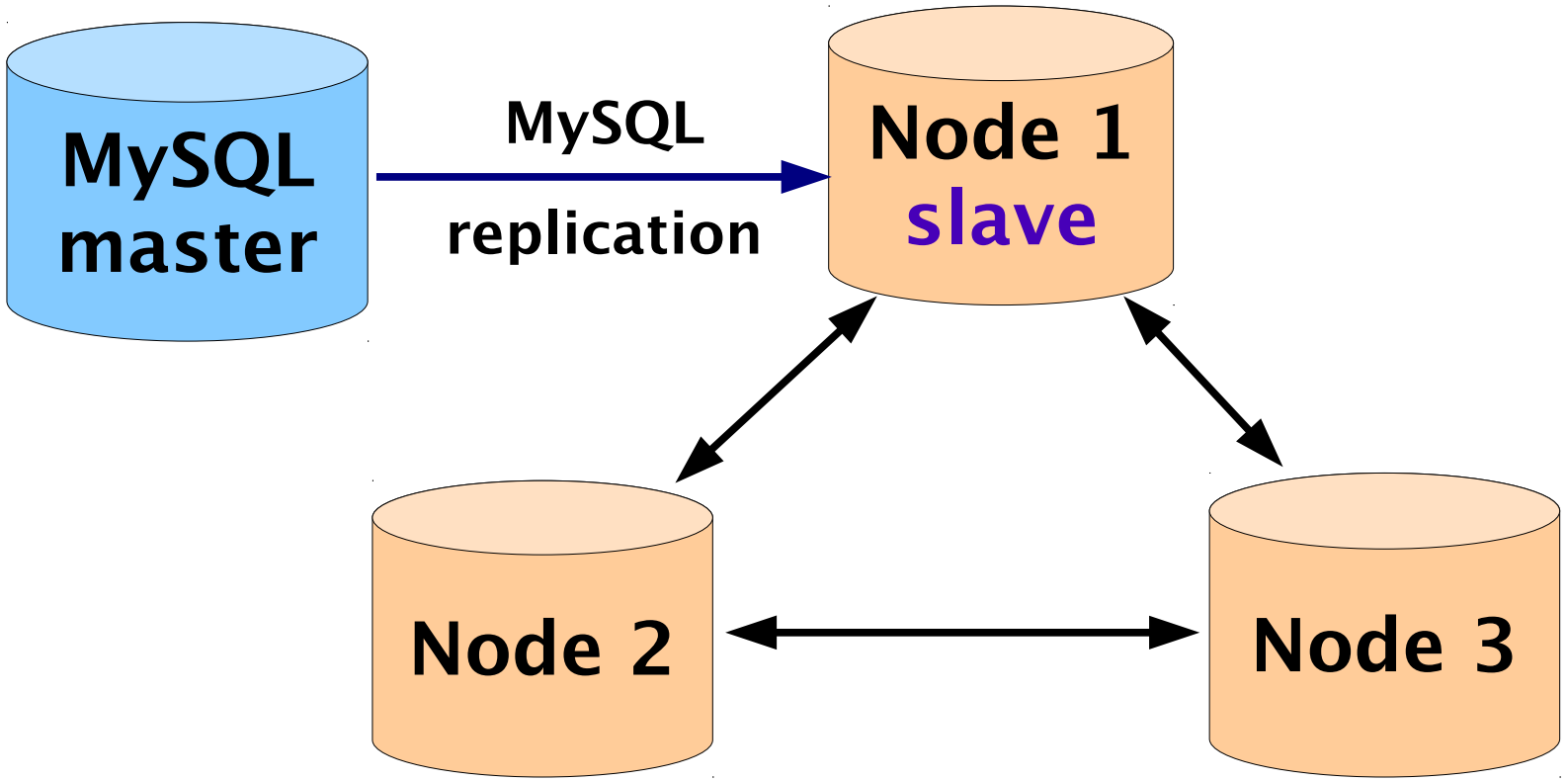
Galera 3.x addresses both problems

Galera as MySQL Slave

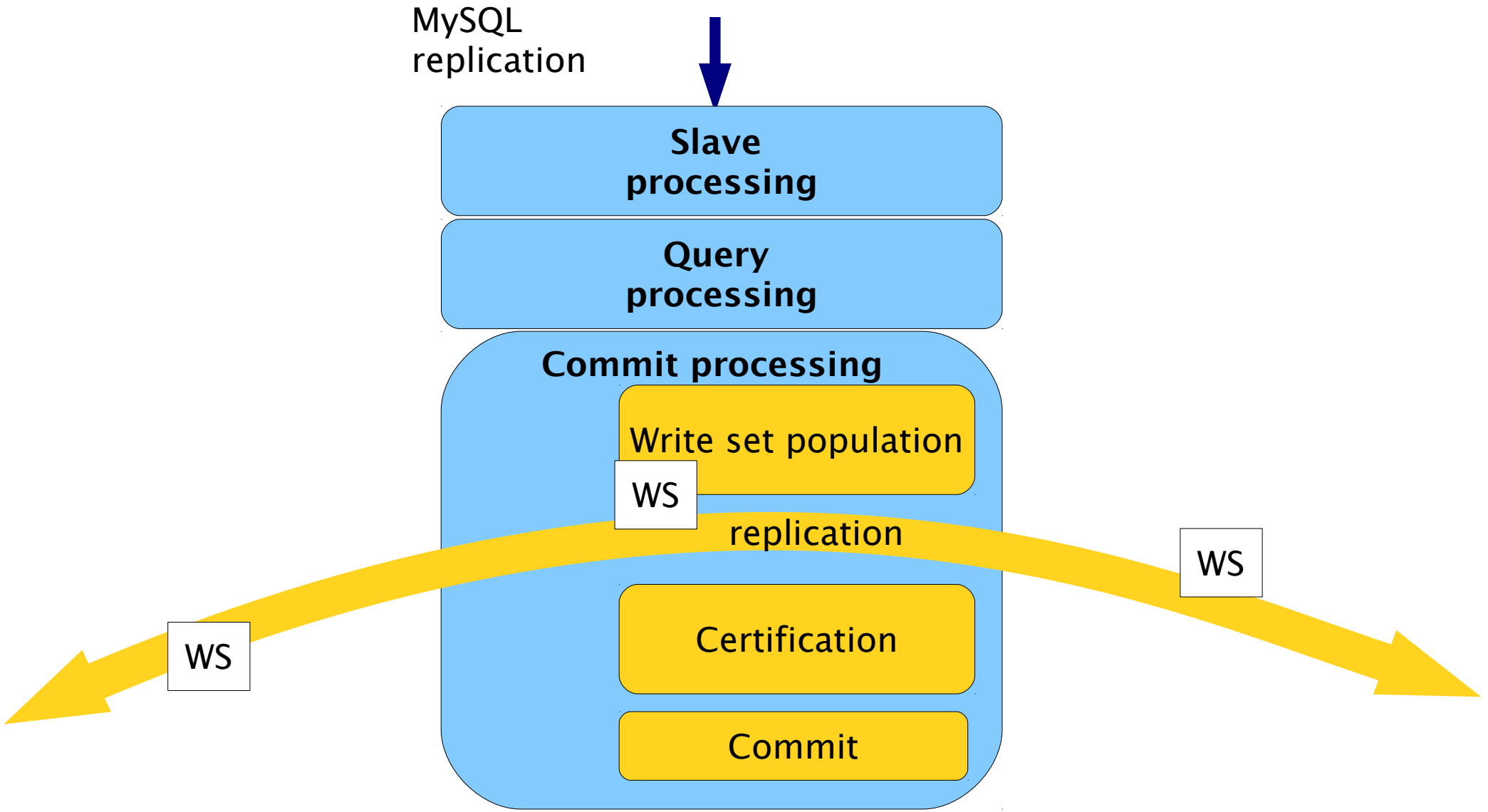
Galera Cluster can operate as MySQL slave in two ways:

1. MySQL master is like client connection to Galera node (version 2 and 3)
2. MySQL replication events will be delivered as pre-ordered events interleaved in Galera replication (new version 3 feature)

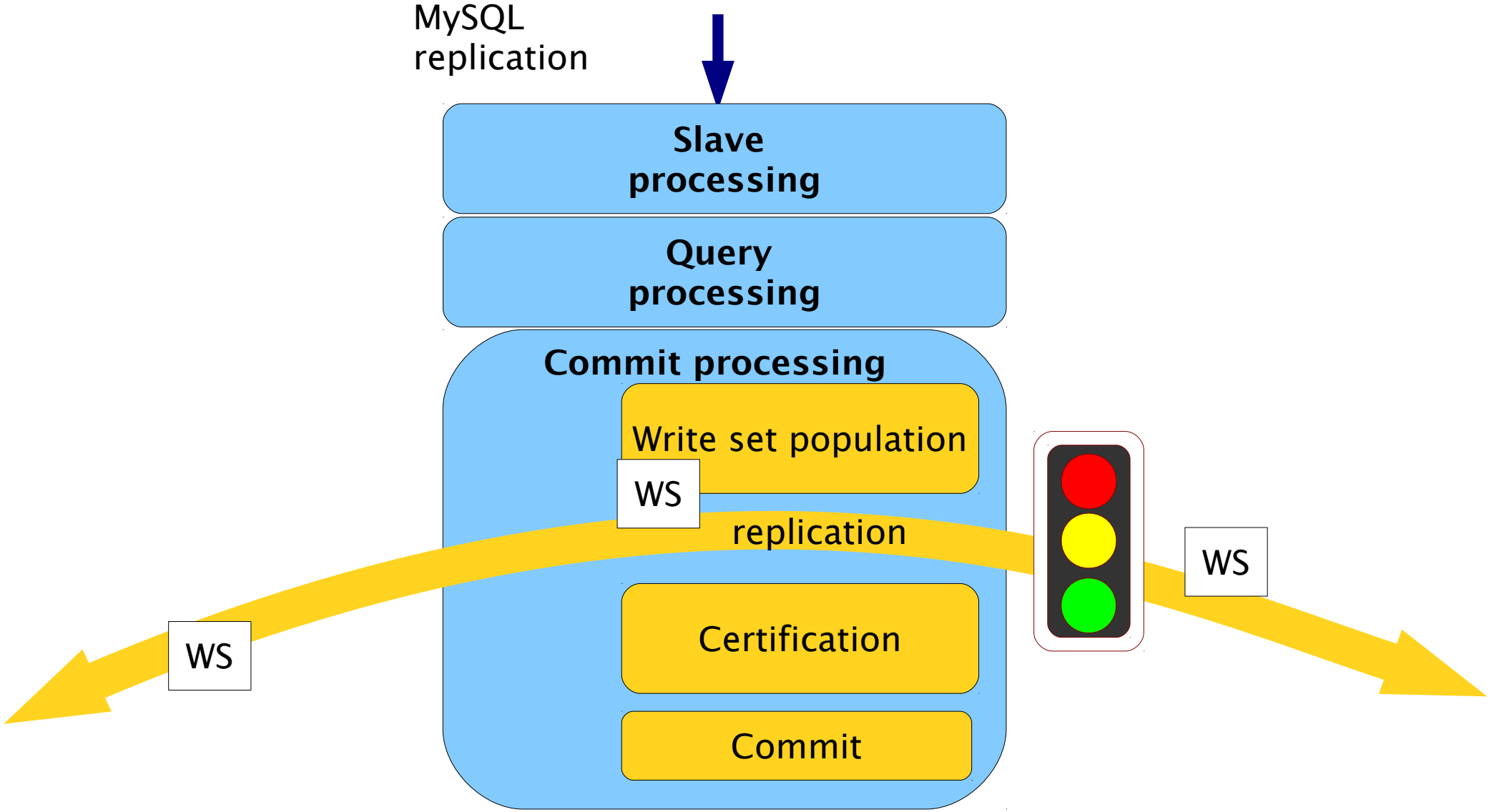
Galera as MySQL Slave



MySQL Slave through Galera Replication



MySQL Slave through Galera Replication



Async Replication Bottleneck

- As MySQL replication events are treated as regular client transactions, they must go through Galera replication pipeline at commit time
- … and this adds some delay before commit
- MySQL replication is single threaded

→ Galera Cluster will not apply replication as fast as native MySQL Slave

- MySQL 5.6 “parallel replication” may help, if you have several schemas
  - But only, if you have several schemas
  - MySQL 5.7 and MariaDB 10 will have proper parallel replication

MySQL Replication Bundling

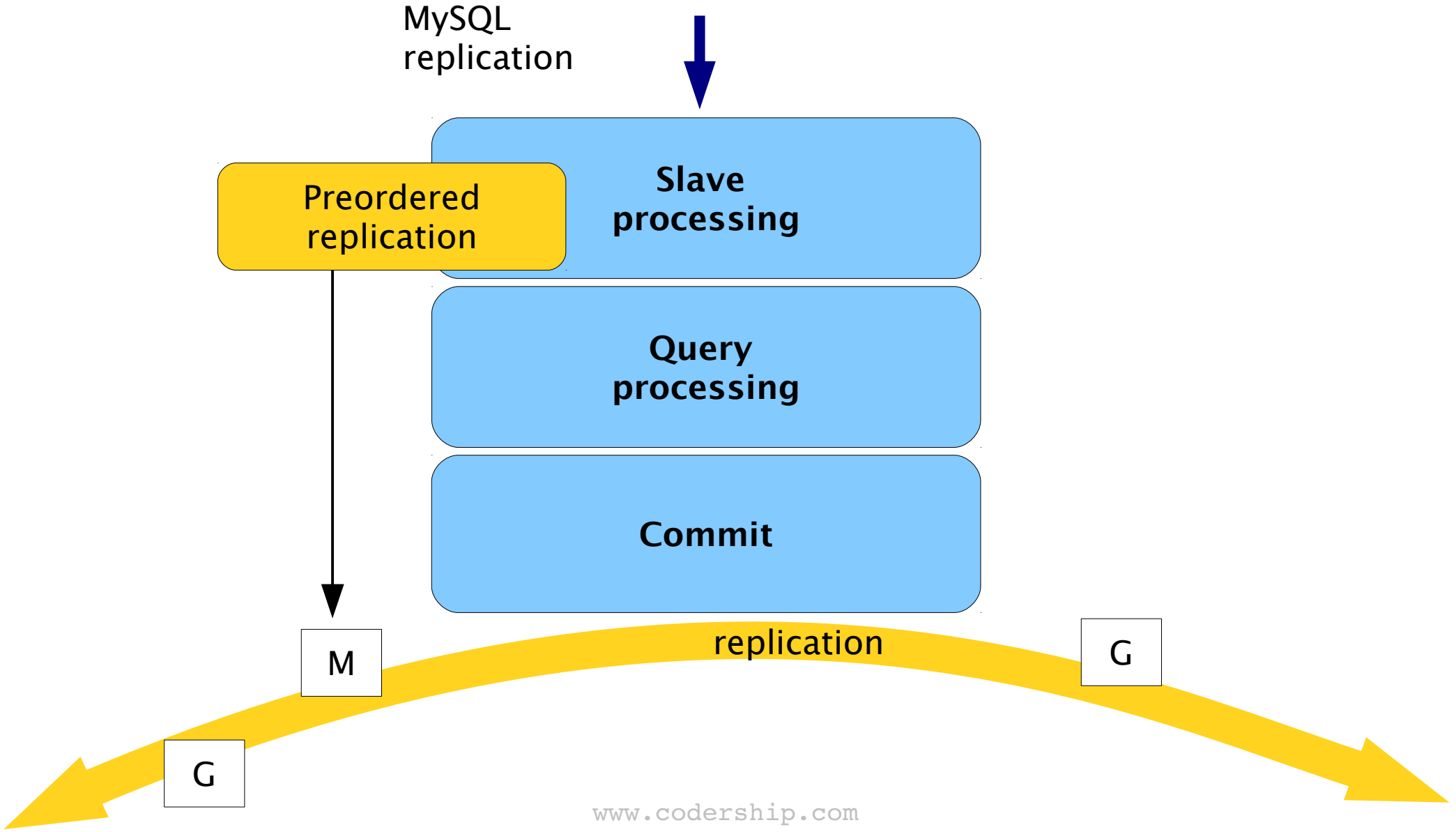
- Replication events can be bundled to commit as a single group

➡ Less waits for replication synchronization

- wsrep_mysql_replication_bundle=n
  - Groups **n** mysql replication transactions in one large transaction
- Helps with the latency, you can go up to 5K trx/sec

Galera 3 Pre-ordered Replication

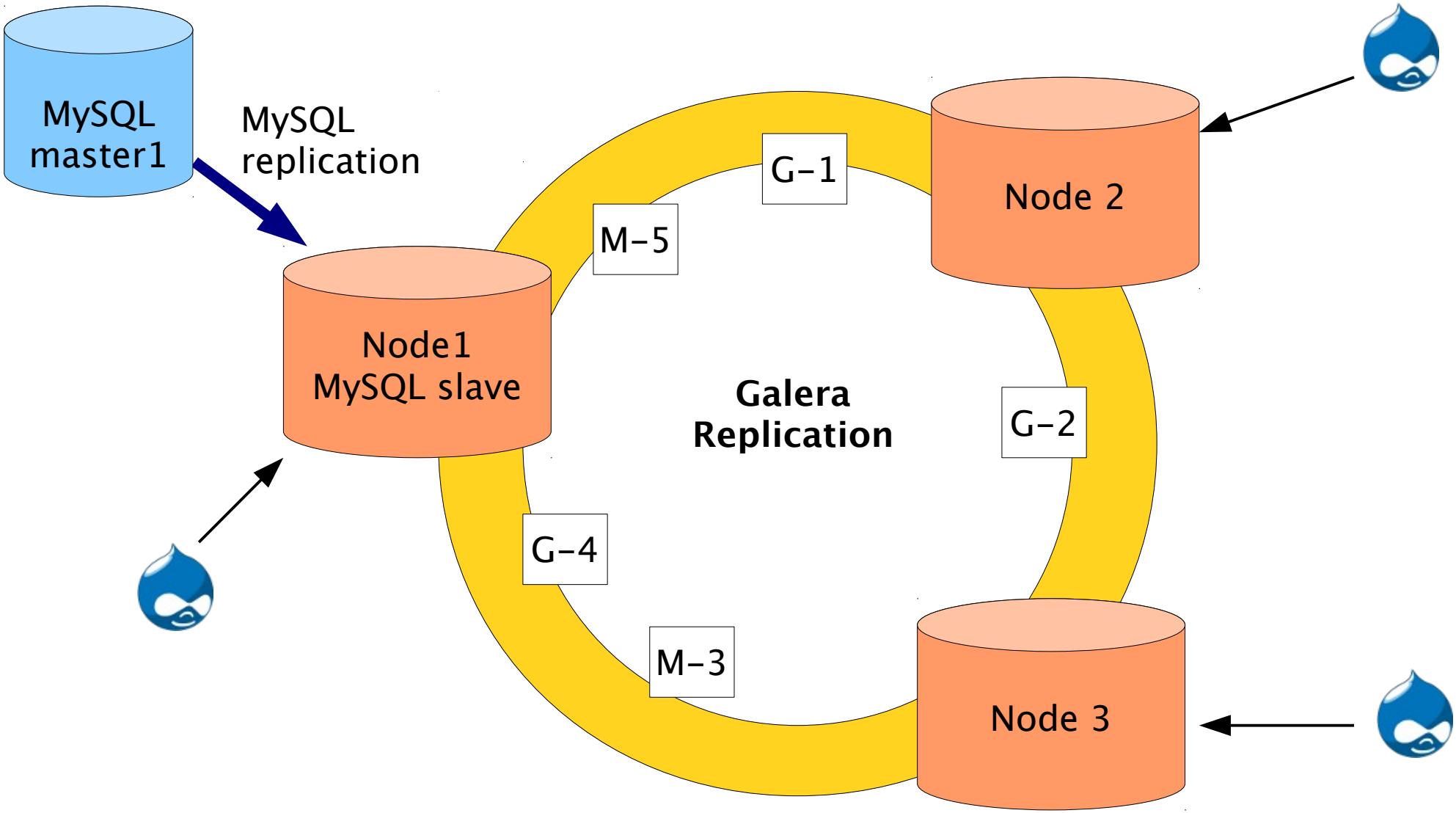
Pre-ordered replication



Pre-ordered Replication

- MySQL replication stream constitutes partial order, which can be interleaved with cluster replication
- SQL slave thread broadcasts MySQL replication events before Galera slaves apply them
- Applying does not slow down SQL slave thread
- Note, we expect MySQL replication stream to be conflict free

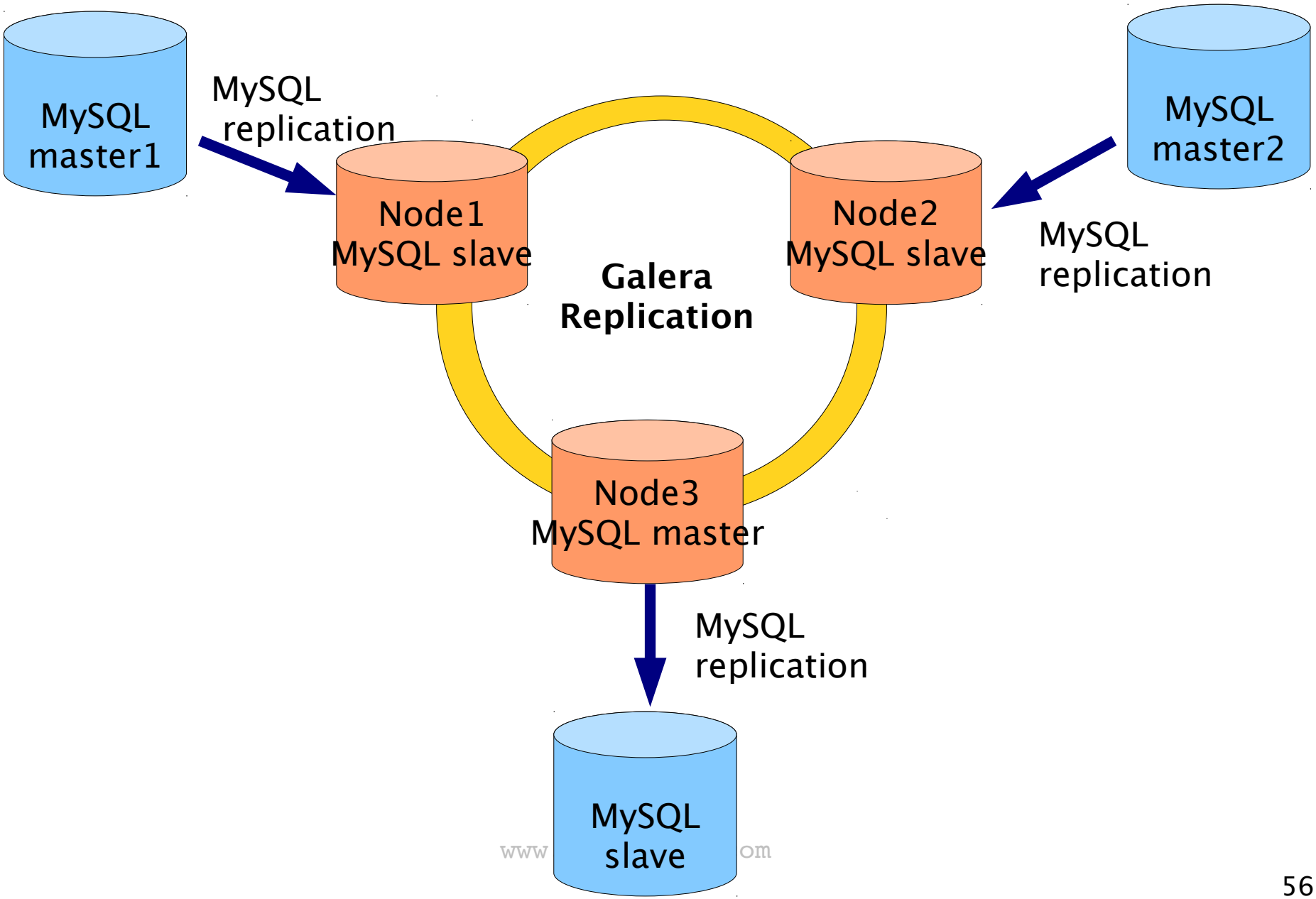
Pre-ordered Replication



Pre-ordered Replication

- Galera nodes will receive and apply MySQL replication events in same order and interleaved with local Galera replication events
- If MySQL replication has conflicts with Galera events, node will do emergency shutdown
  - A better approach for dealing with conflicts is under works
  - The usual multi-master conflict scenarios apply here:
    - Delete conflict
    - Update conflict
    - Uniqueness conflict

MySQL Replication – Multi Source



MySQL Replication – Multi Source

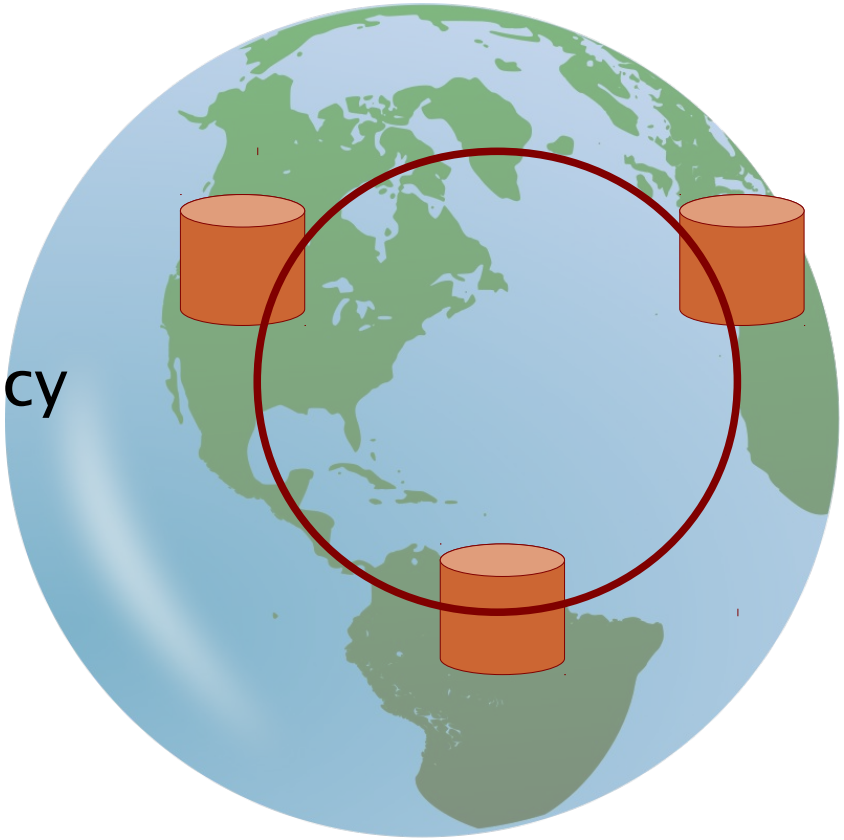
Works with both MySQL replication strategies

- "Slave through Galera Replication", provides certification for MySQL masters
  - Consistency is guaranteed
- "Pre-ordered Replication", is high speed but requires consistency guarantee from application

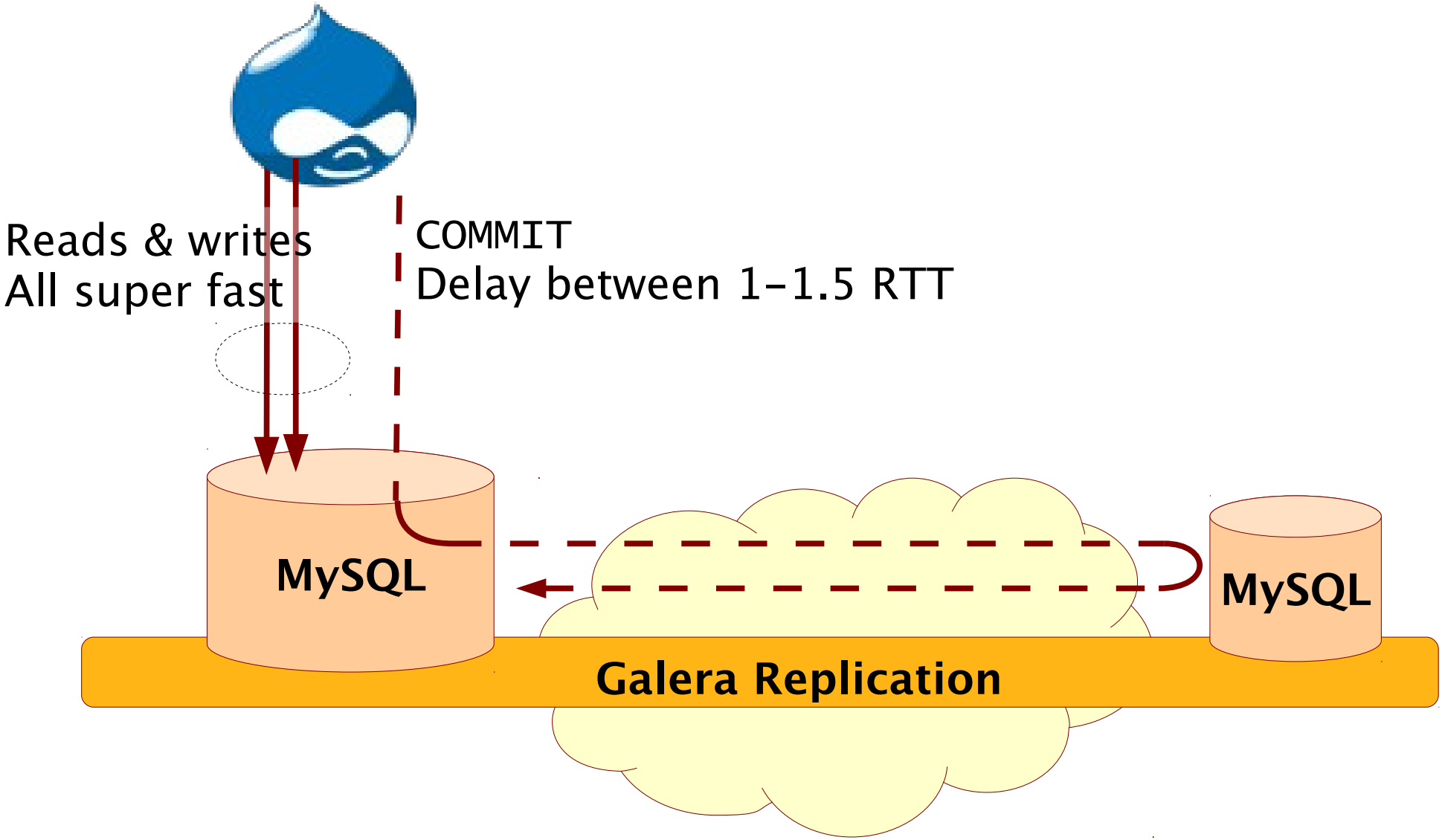
WAN Replication

WAN replication

- Works fine
- Use higher timeouts
- No impact on reads
- No impact within a transaction
- adds 0-300 ms to commit latency

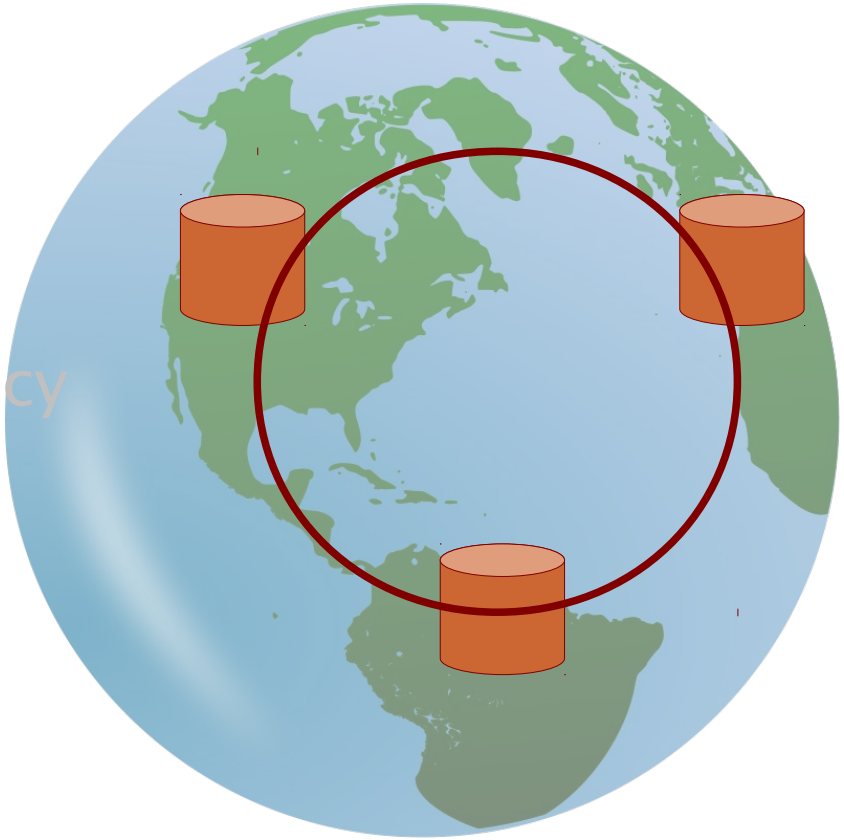


WAN Cluster



WAN replication

- Works fine
- Use higher timeouts
- No impact on reads
- No impact within a transaction
- adds 0–300 ms to commit latency
- No major impact on tps
- Quorum between data centers
  - 3 data centers
  - Weighted quorum calculation
  - Garbd arbitrator

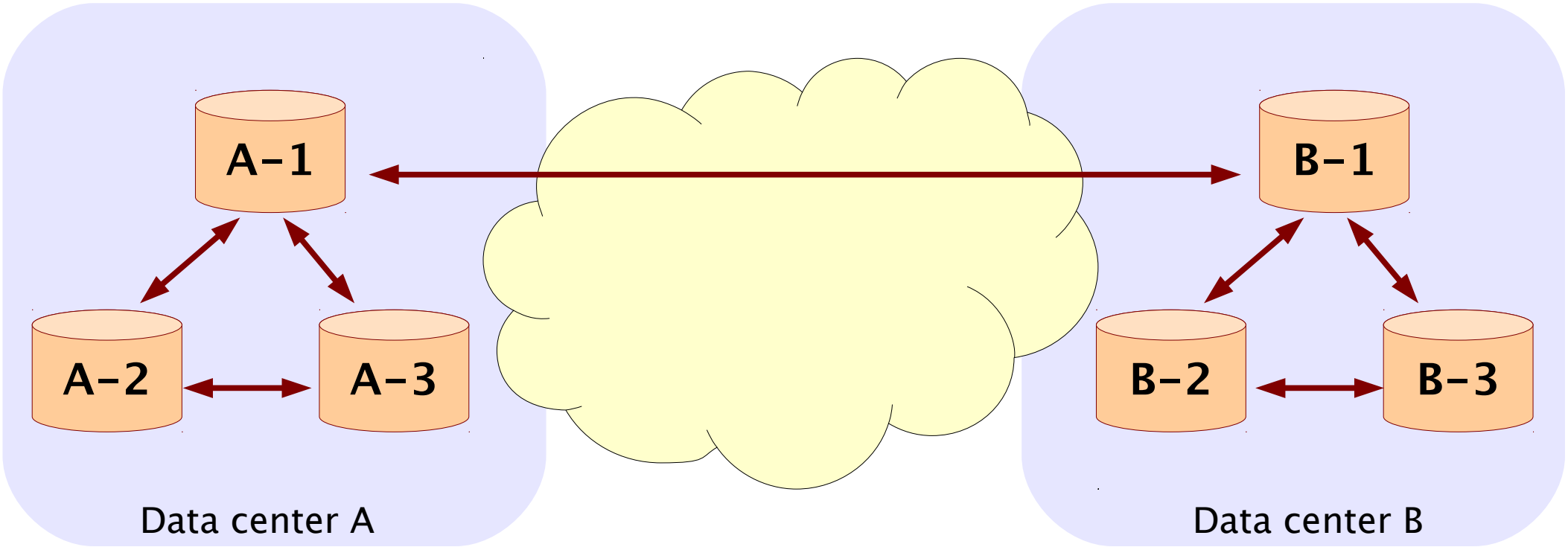


WAN Replication 2.0

Galera 3.0 has new replication mode, which is optimized for WAN networks (or any network with high latencies in general).

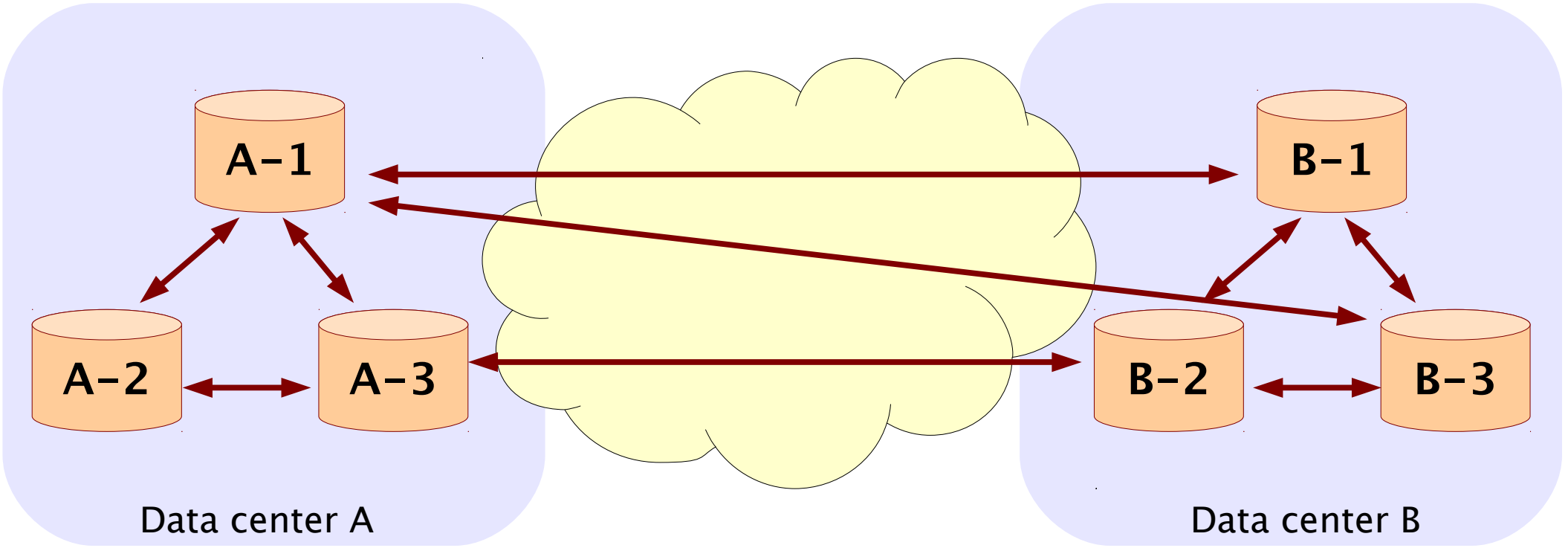
But, let's first take a look how Galera 2.0 replication looks between data centers.

WAN Replication 2.0



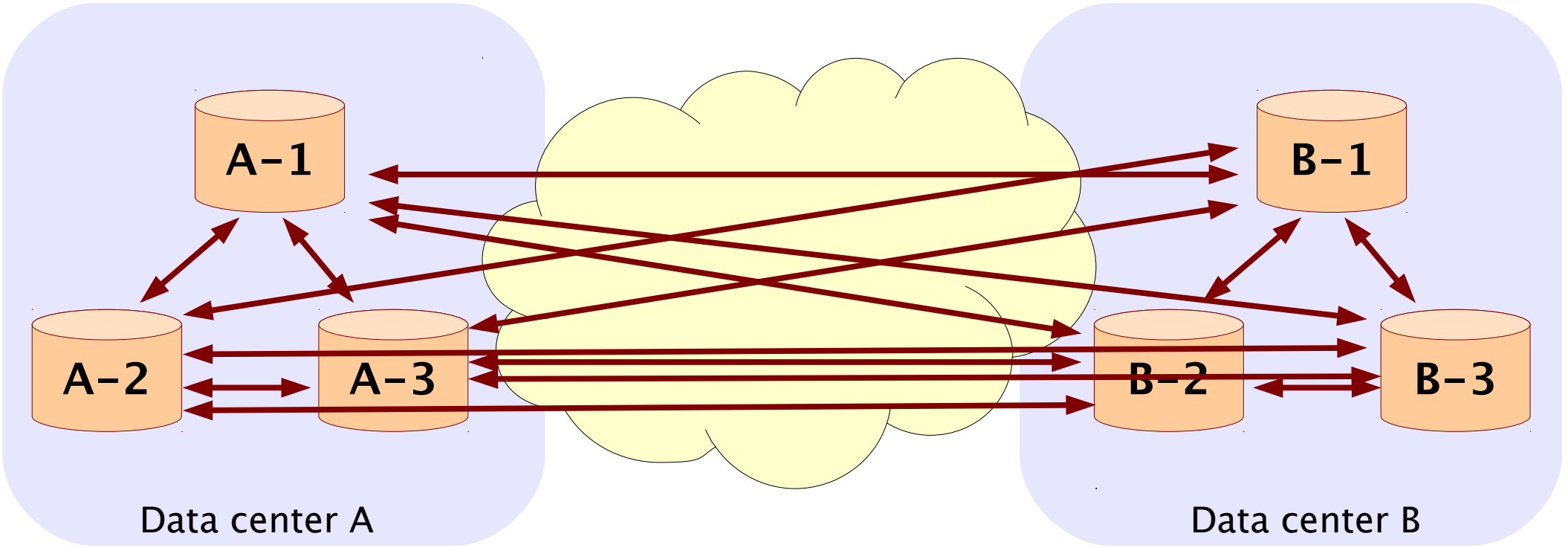
- All point-to-point connections will be needed for replication

WAN Replication 2.0



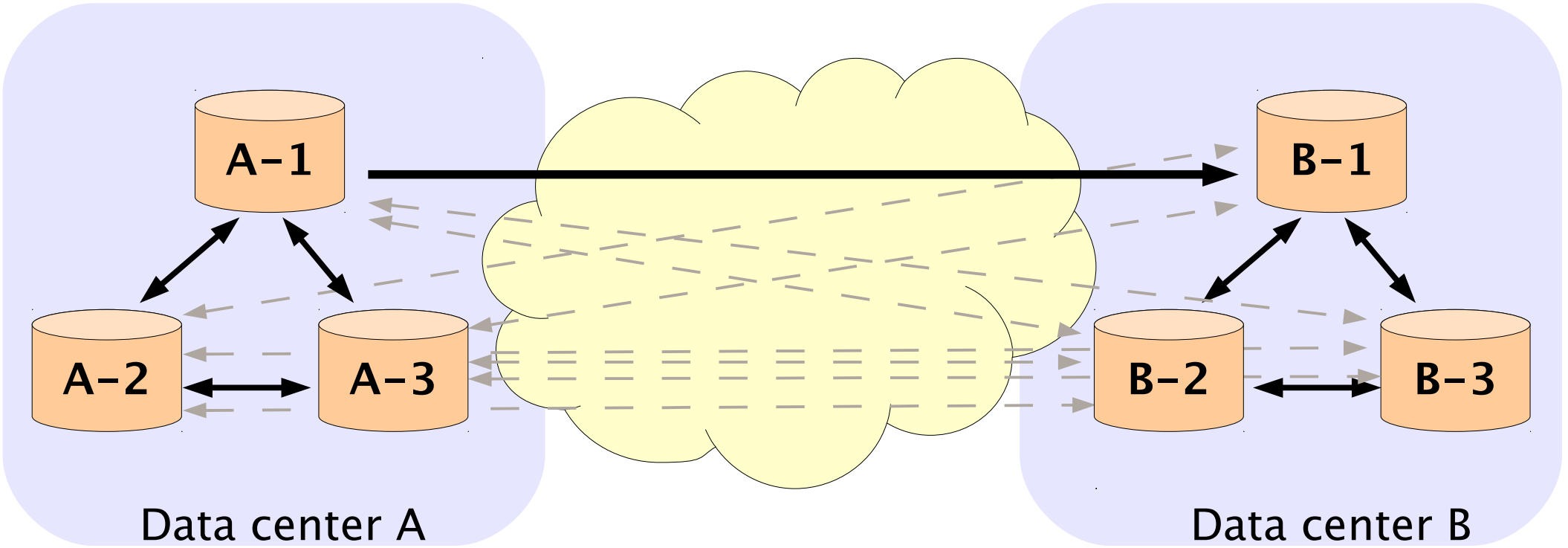
- All point-to-point connections will be needed for replication

WAN Replication 2.0



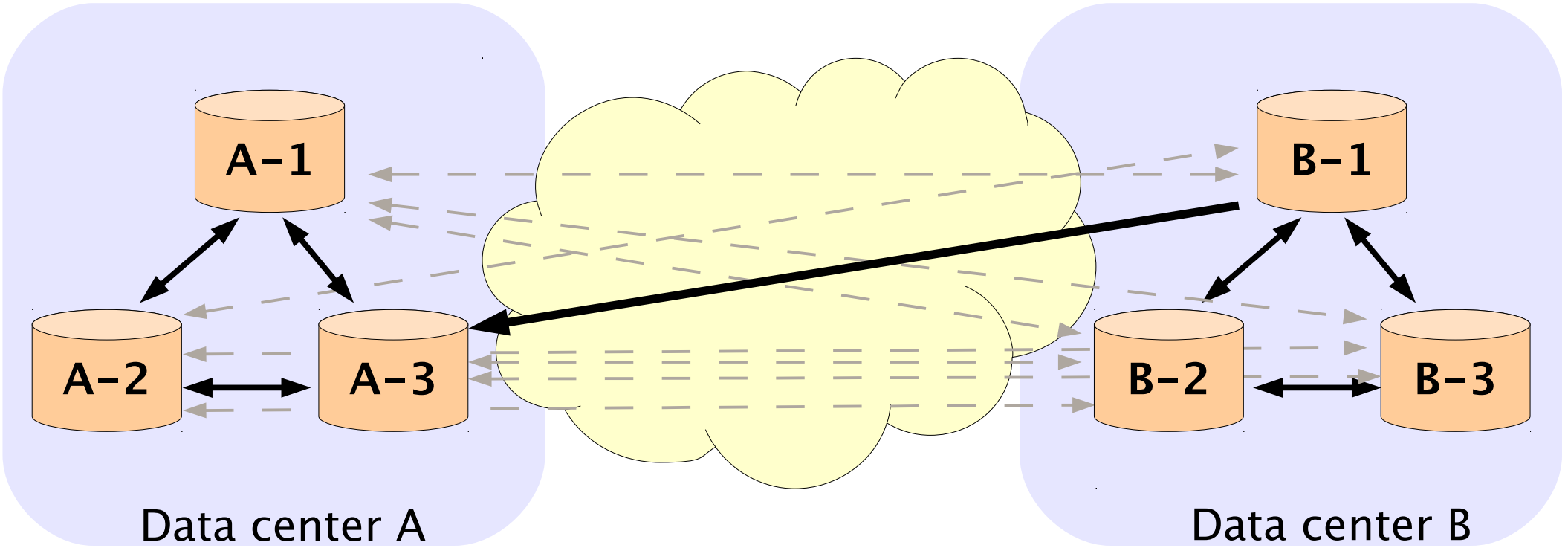
- All point-to-point connections will be needed for replication
- I said **ALL**

WAN Replication 3.0



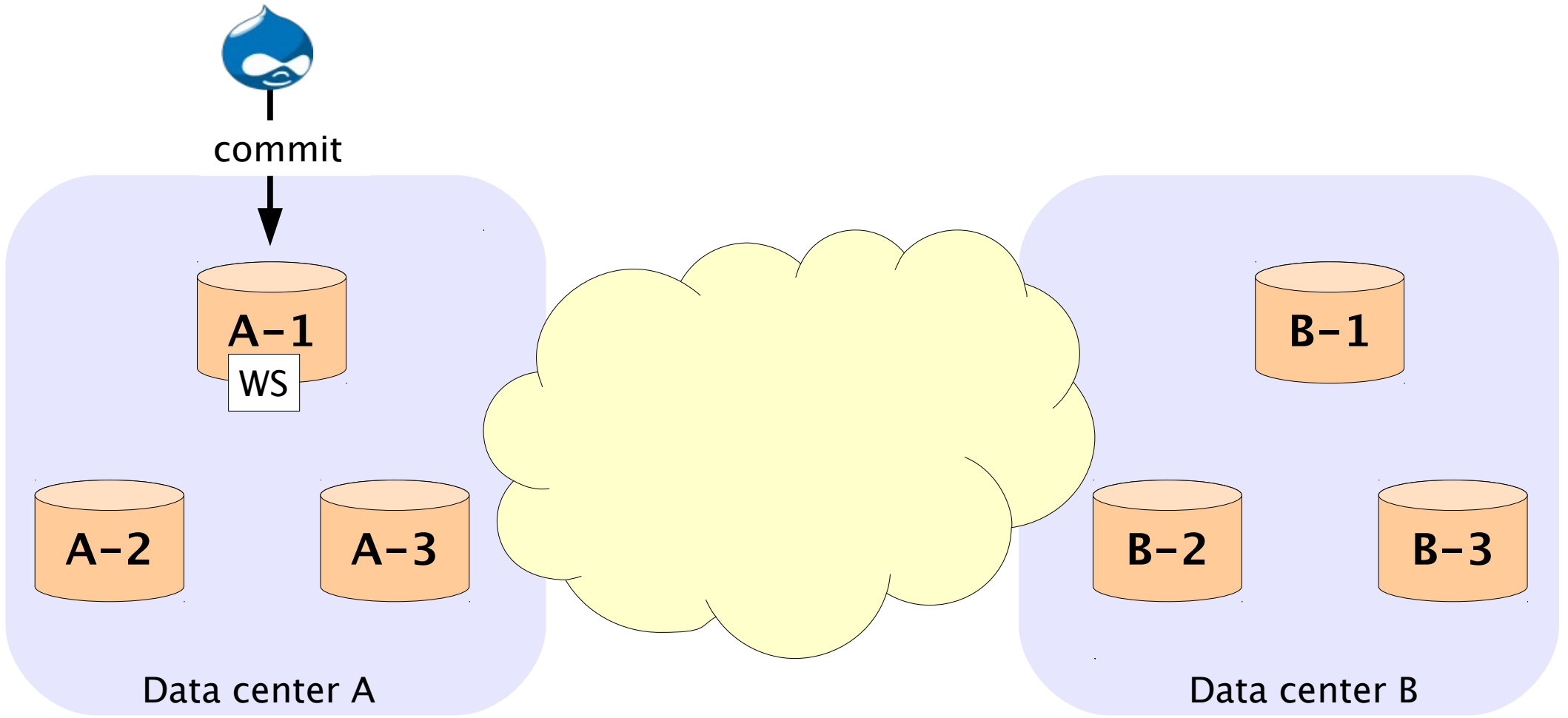
- Data messages between cluster segments is sent only once
- Replication events will be distributed within each segment p2p

WAN Replication 3.0

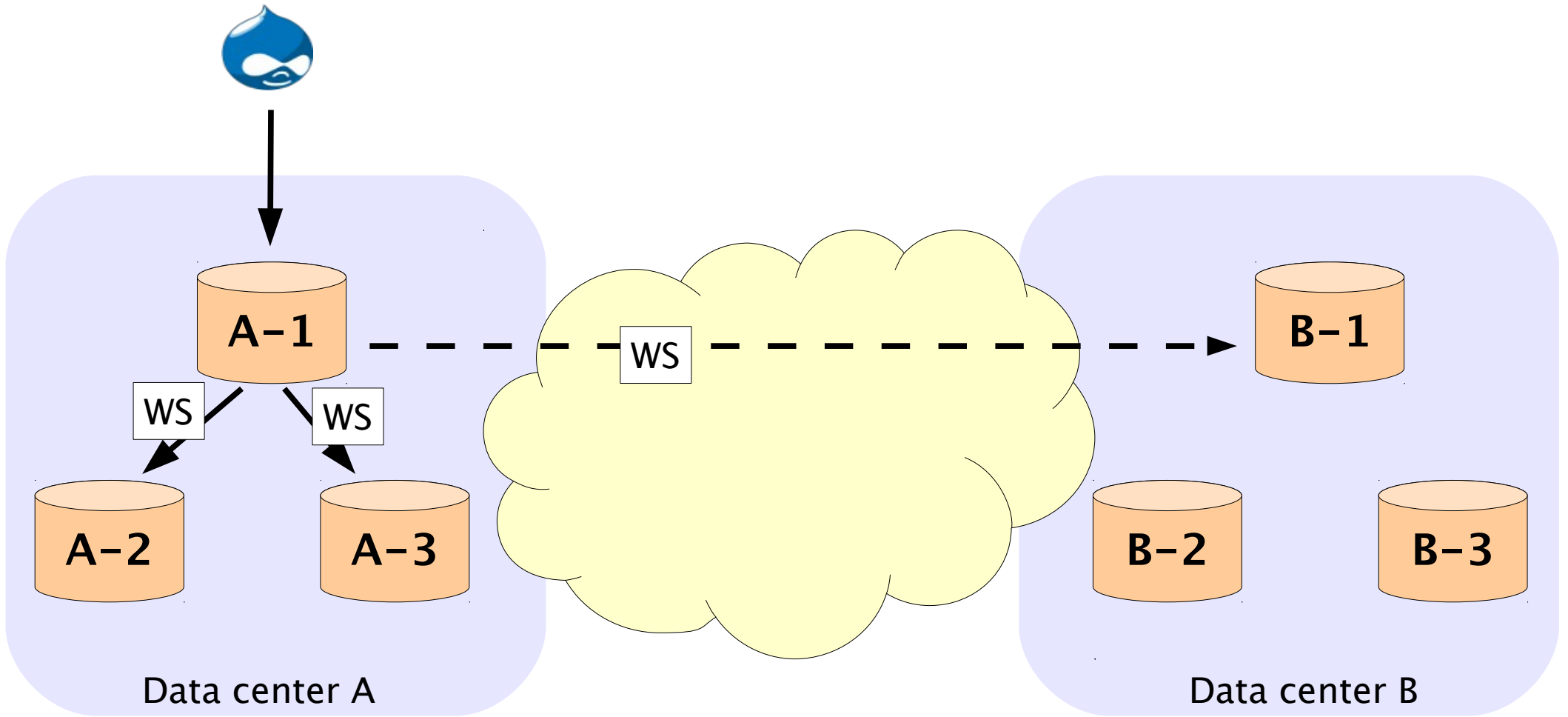


- Replication between segments go over one link only
- Replication events will be distributed within each segment p2p
- Segment gateways can change per transaction

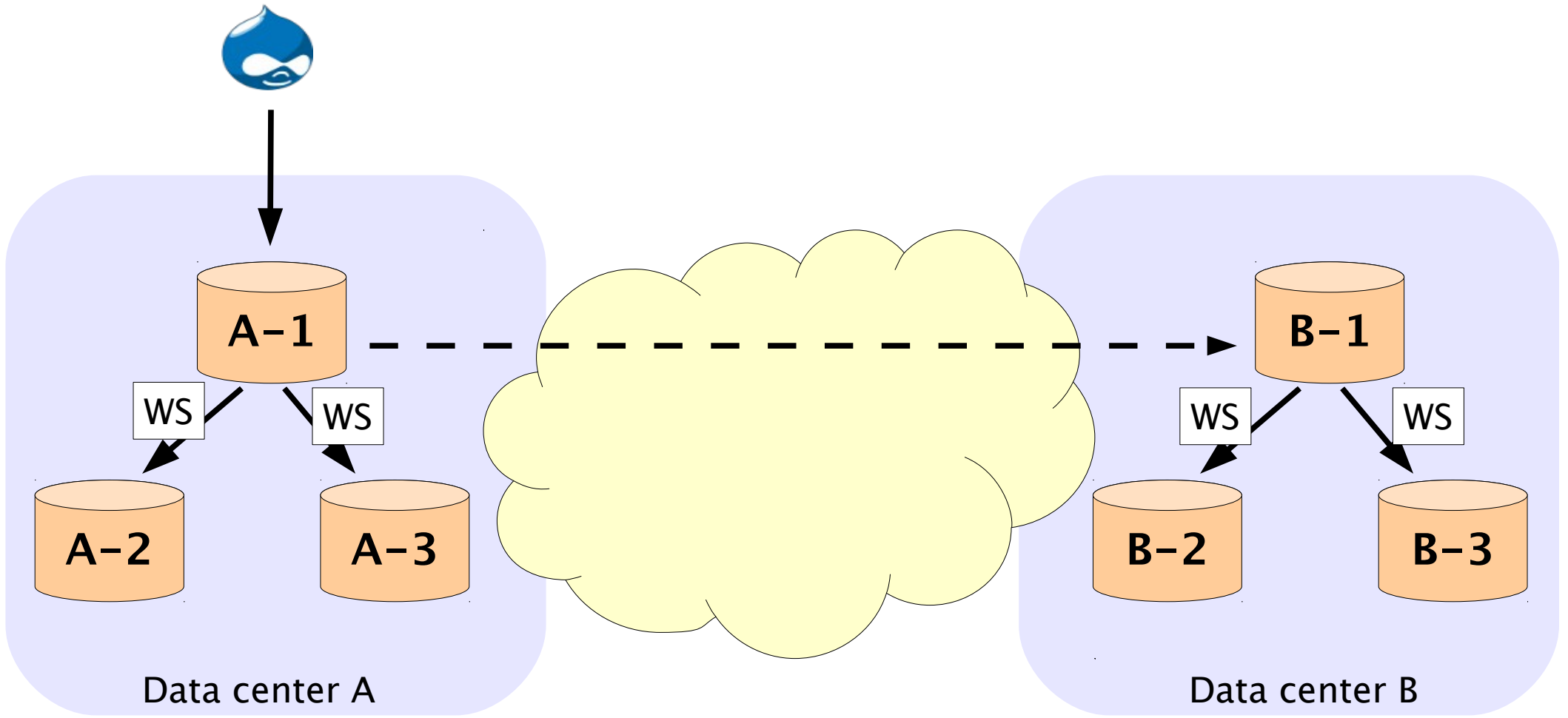
WAN Replication 3.0



WAN Replication 3.0



WAN Replication 3.0



WAN Replication 3.0

Define cluster segments up front by node location

```
gmcast.segment = 1..255
```

SST in WAN

Galera will choose SST/IST Donor from same segment, if possible

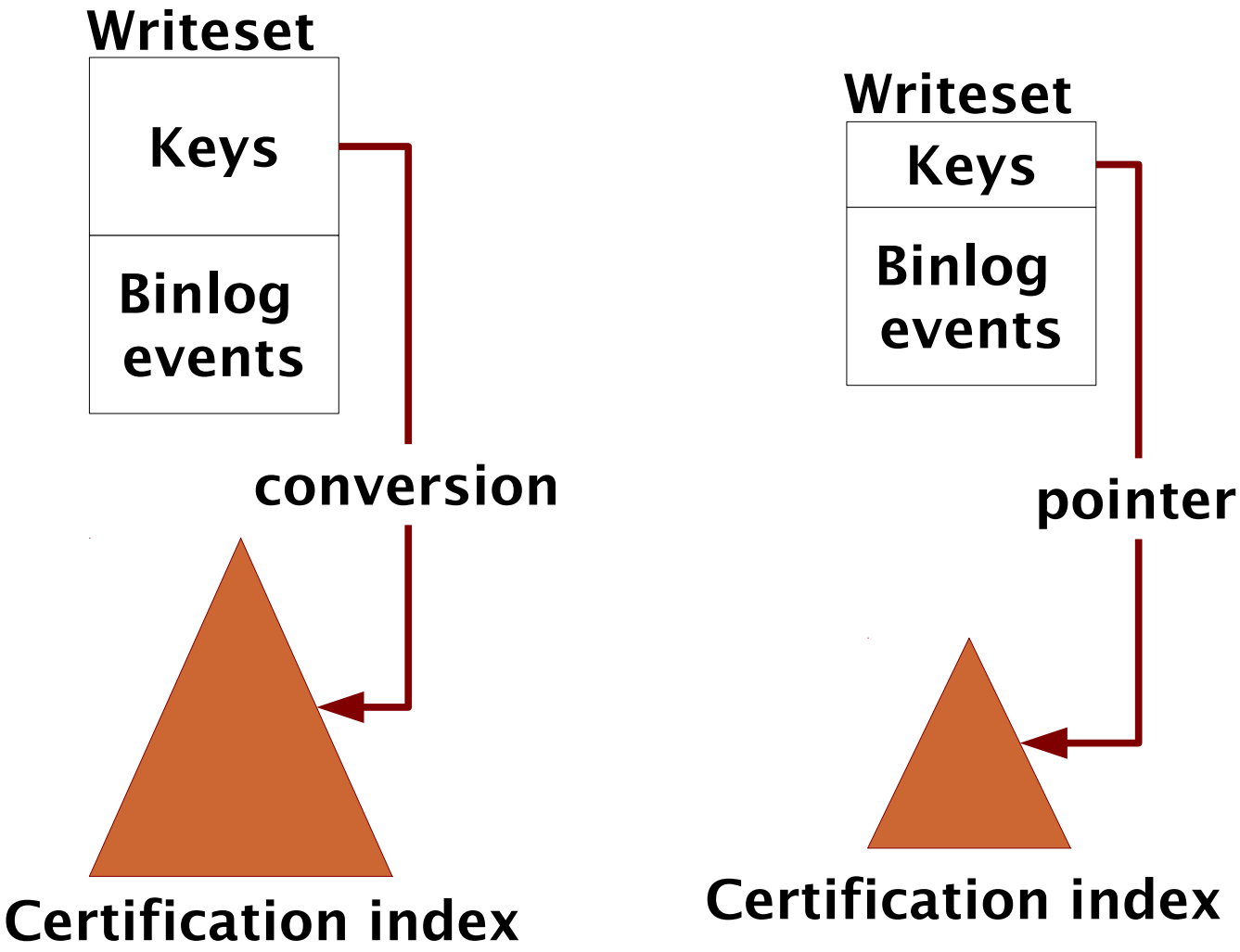
Optimized Writeset Format

New writeset format: Certification Keys

Writeset contains references for all affected primary, unique and foreign keys.

These affected keys are also stored in certification index maintained in each node to find out possible multi-master conflicts.

New writeset format: Certification Keys



New writeset format: Certification Keys

Galera 3.0 has refactored these key representations so that same compact format can be used in write set and certification index.

- Memory footprint is minimal for each key and fixed length.
- Certification check runs faster on the new format.
- More and larger transactions can run simultaneously without memory issues.

New Writeset Format

In our tests we see

- 5-10% less CPU usage on both master and slave sides
- 10-15% better throughput with big transactions
- 2x smaller memory overhead

Compared to the latest 2.x release (25.2.8)

Huge Transaction Support

Writeset format refactoring is one step forward in huge transaction support.

However, more work remains for achieving that:

- We still replicate transaction as a singe batch at commit time. It needs to be fragmented.

LOAD DATA transaction can be split in a series of 10K row transactions

wsrep_load_data_splitting = ON | OFF

Galera Project

Galera Project

- Galera Cluster for MySQL
  - 6 years development
  - based on MySQL server community edition
  - Fully open source
  - Active community
- ~3 releases per year
  - Latest GA releases: 2.8, 3.1

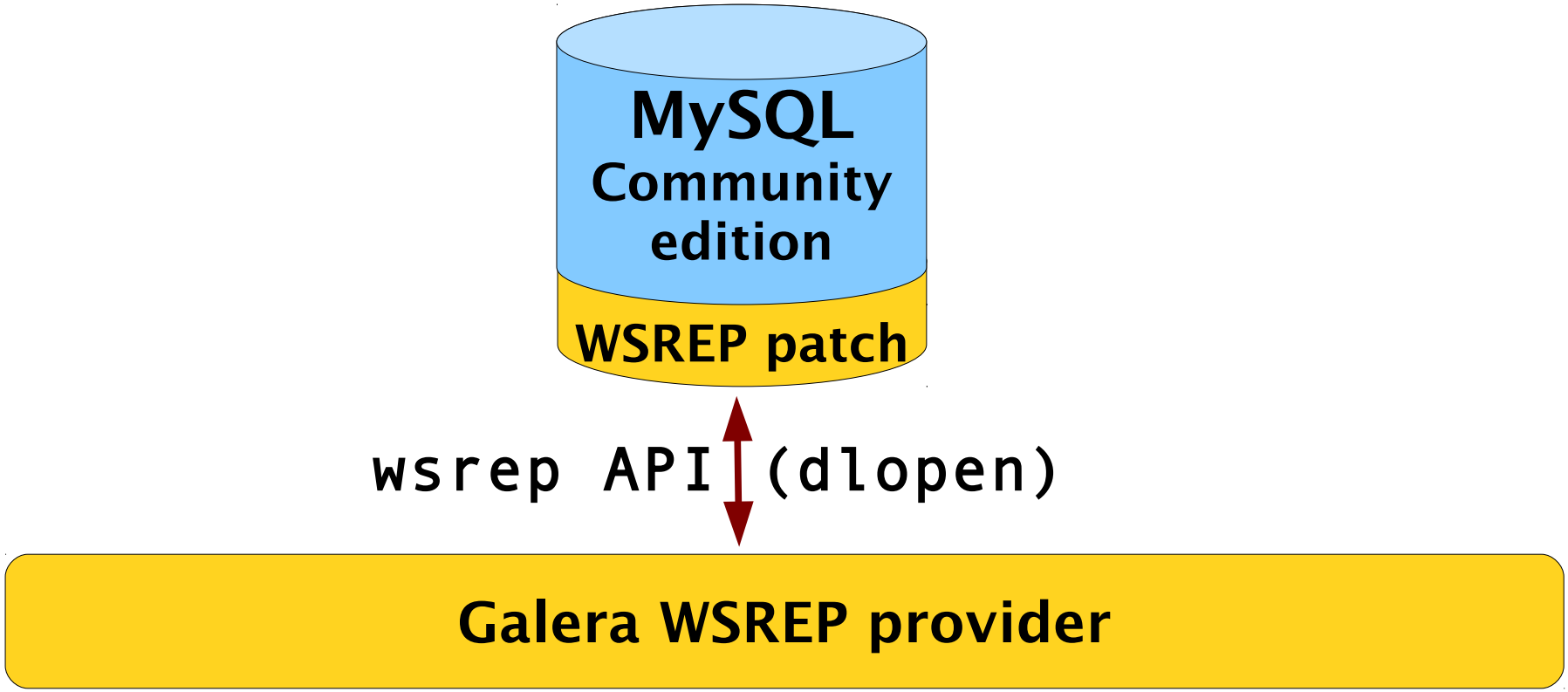
Galera Cluster

Galera Cluster

Galera Cluster for MySQL

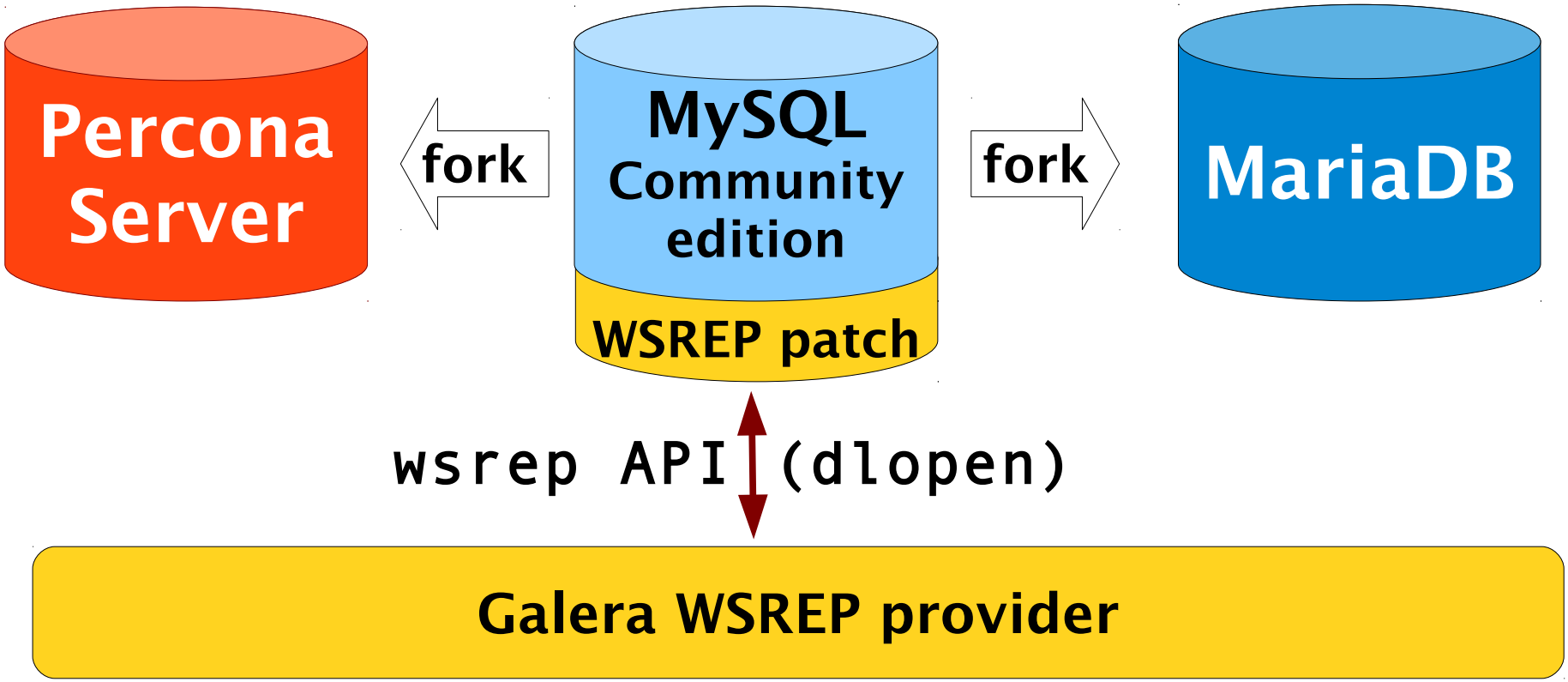
Galera Project

Galera Cluster for MySQL



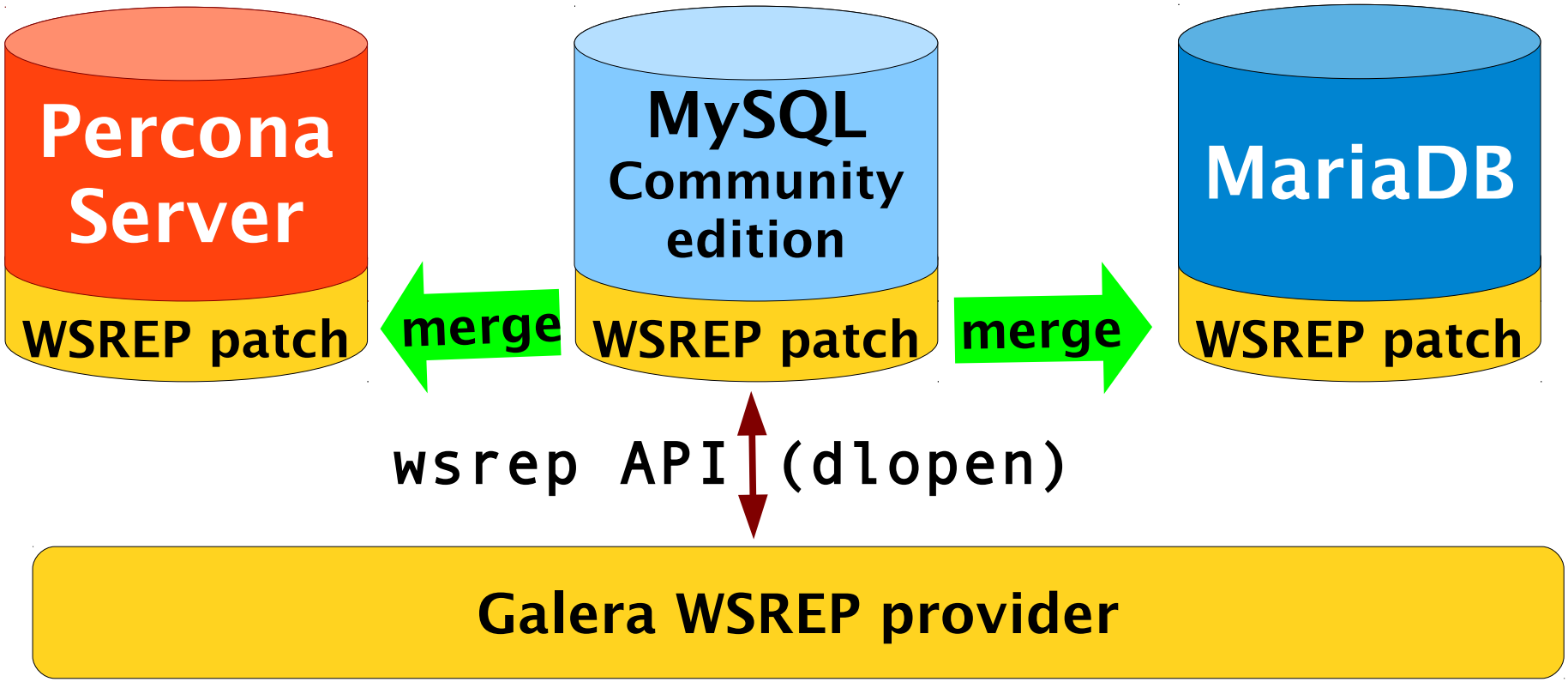
Galera Project

Galera Cluster for MySQL

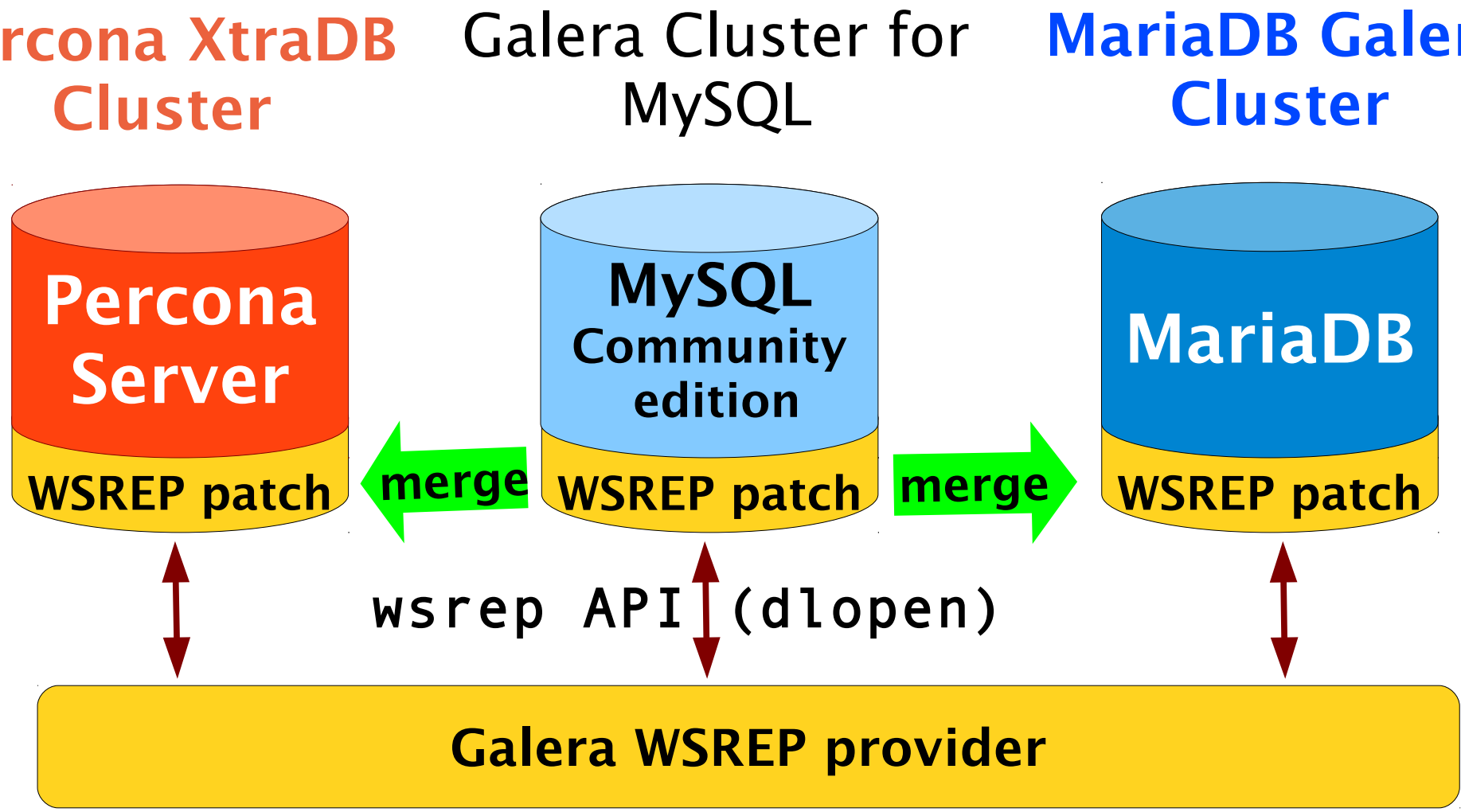


Galera Project

Galera Cluster for MySQL



Galera Project



Questions?

Thank you for listening!
Happy Clustering :-)


GALERA
3