Synchronous Replication Use Cases

Seppo Jaakola
Codership

codership

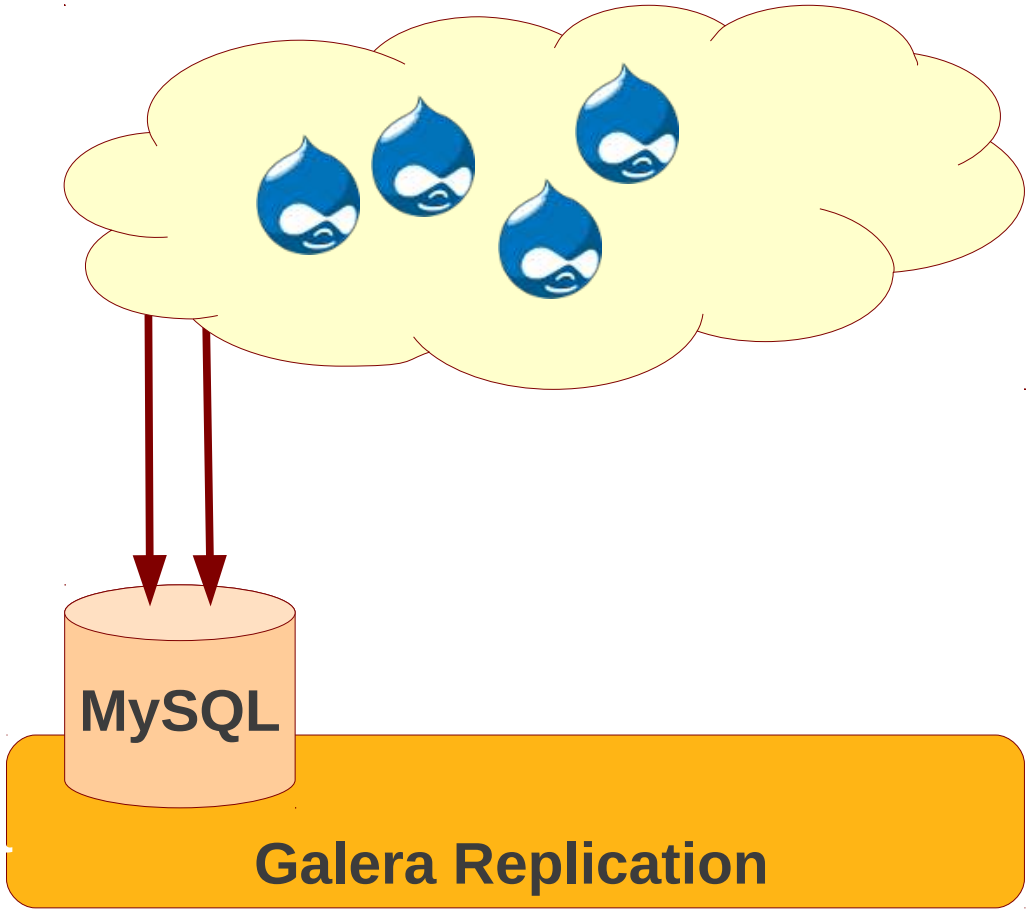
Agenda

- Galera Cluster Short Introduction
- Topologies
- Connecting to Galera Cluster
- Use Cases
- Galera Project

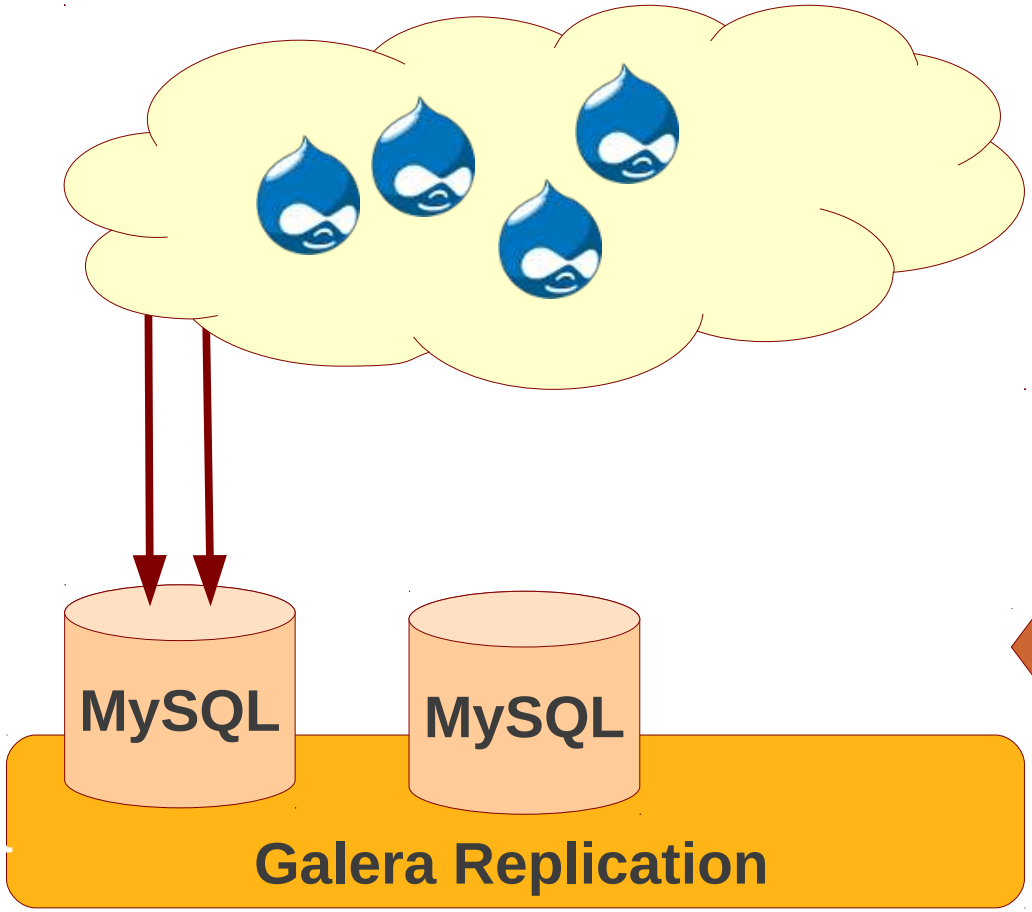
Galera Cluster

codership

Multi-Master Replication

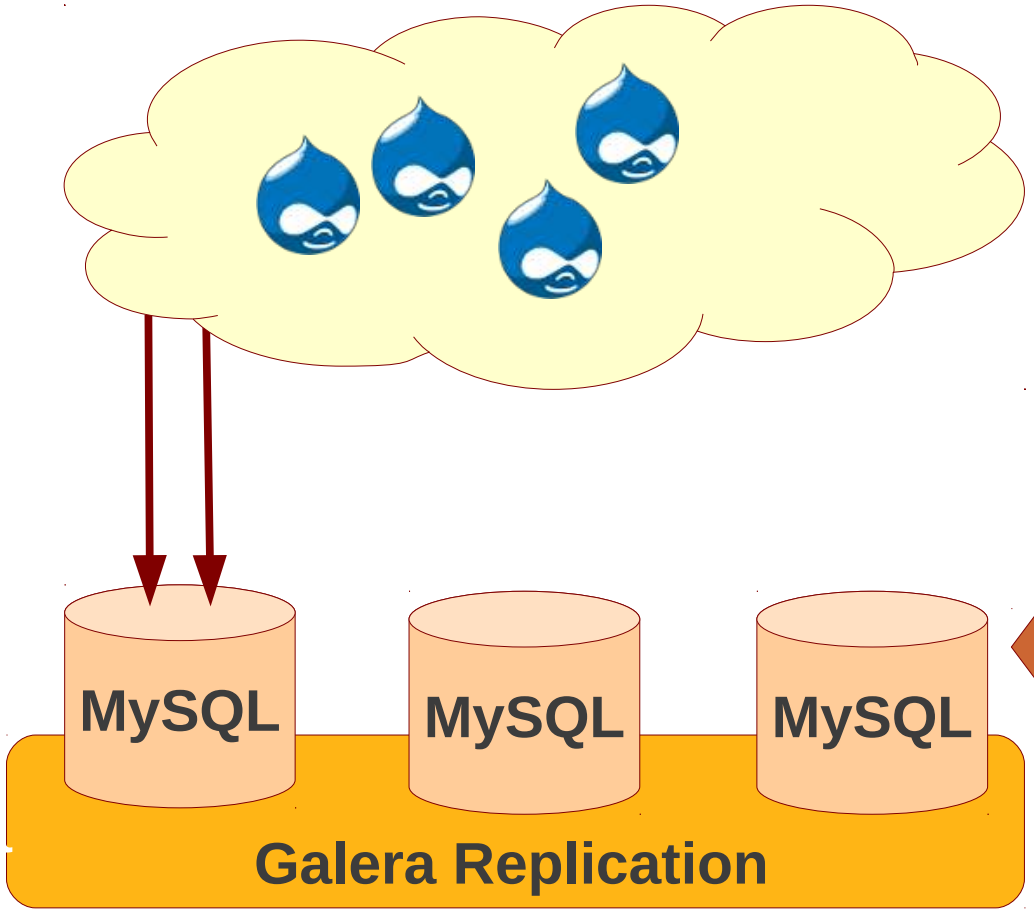


Multi-Master Replication



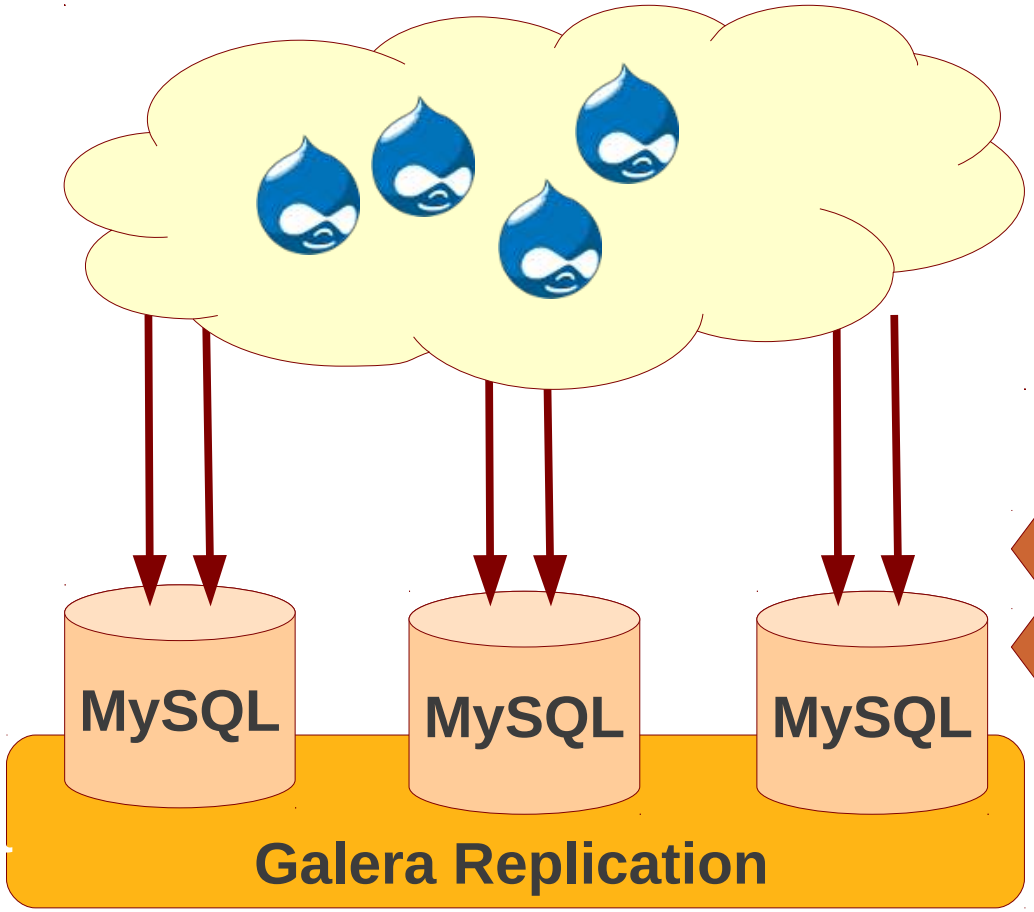
There can be several nodes

Multi-Master Replication



There can be several nodes

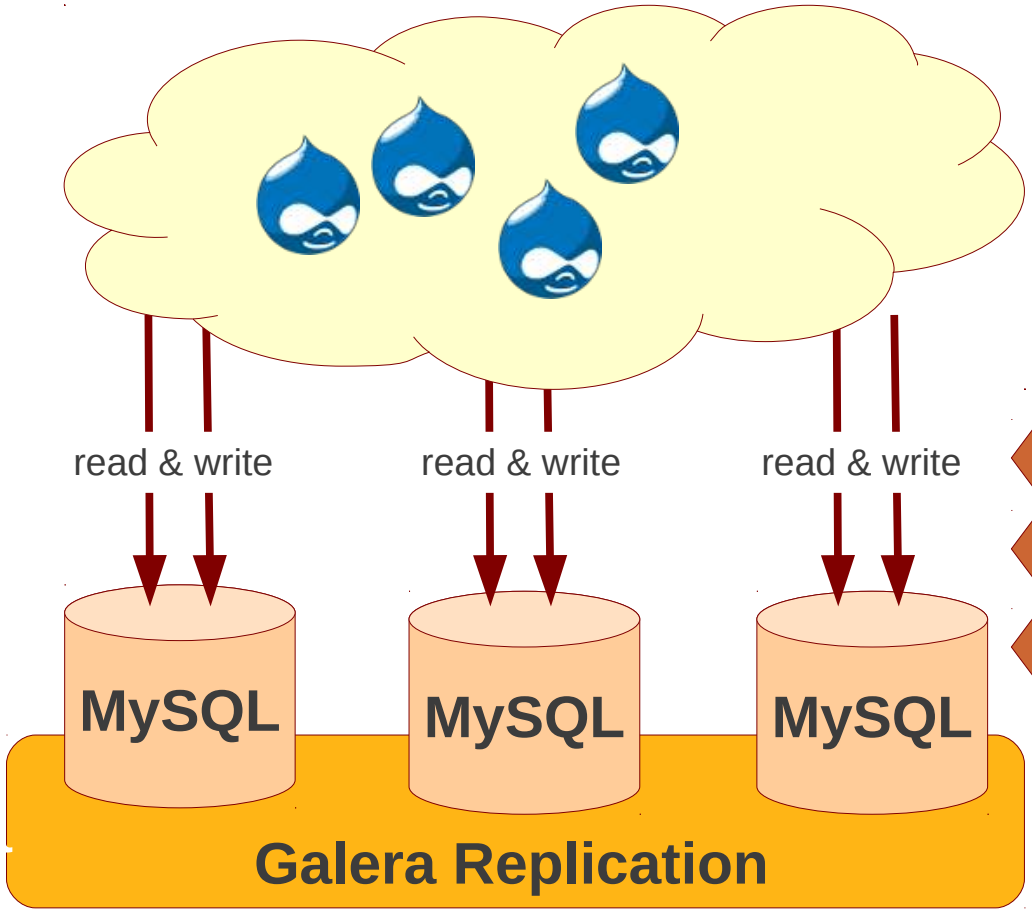
Multi-Master Replication



Client can connect to any node

There can be several nodes

Multi-Master Replication

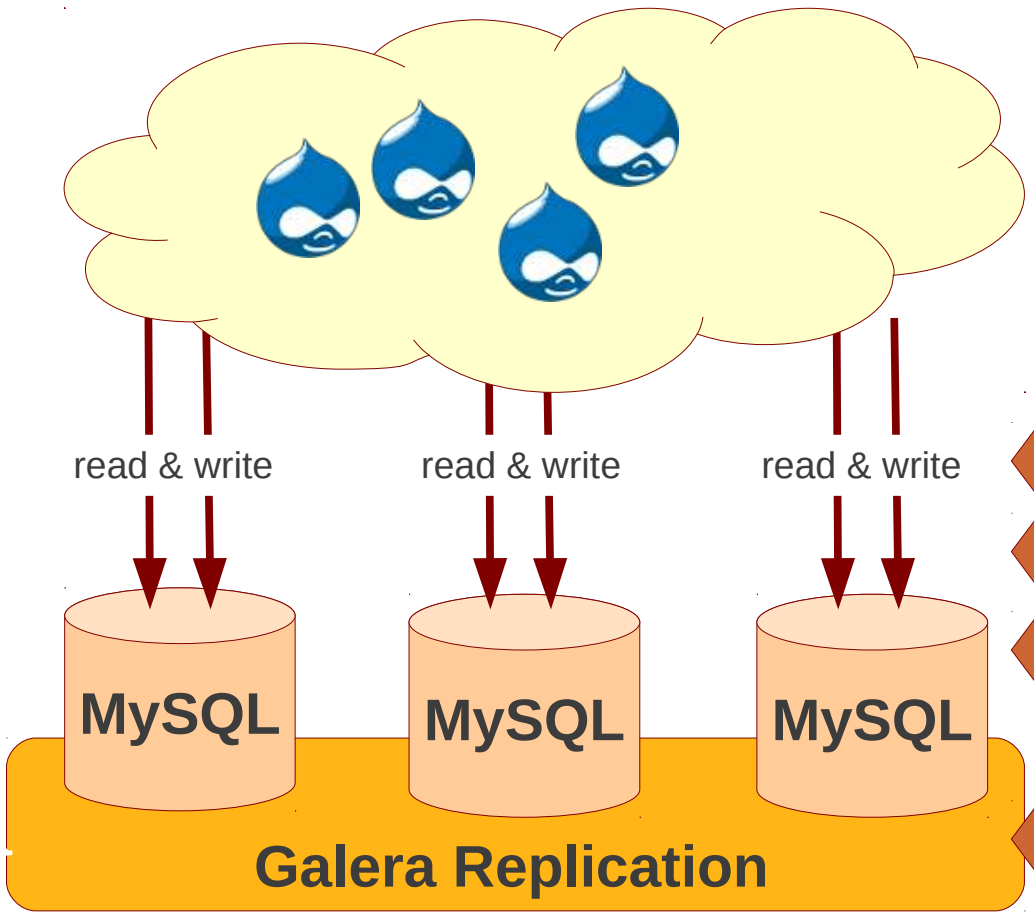


Read & write access to any node

Client can connect to any node

There can be several nodes

Multi-Master Replication



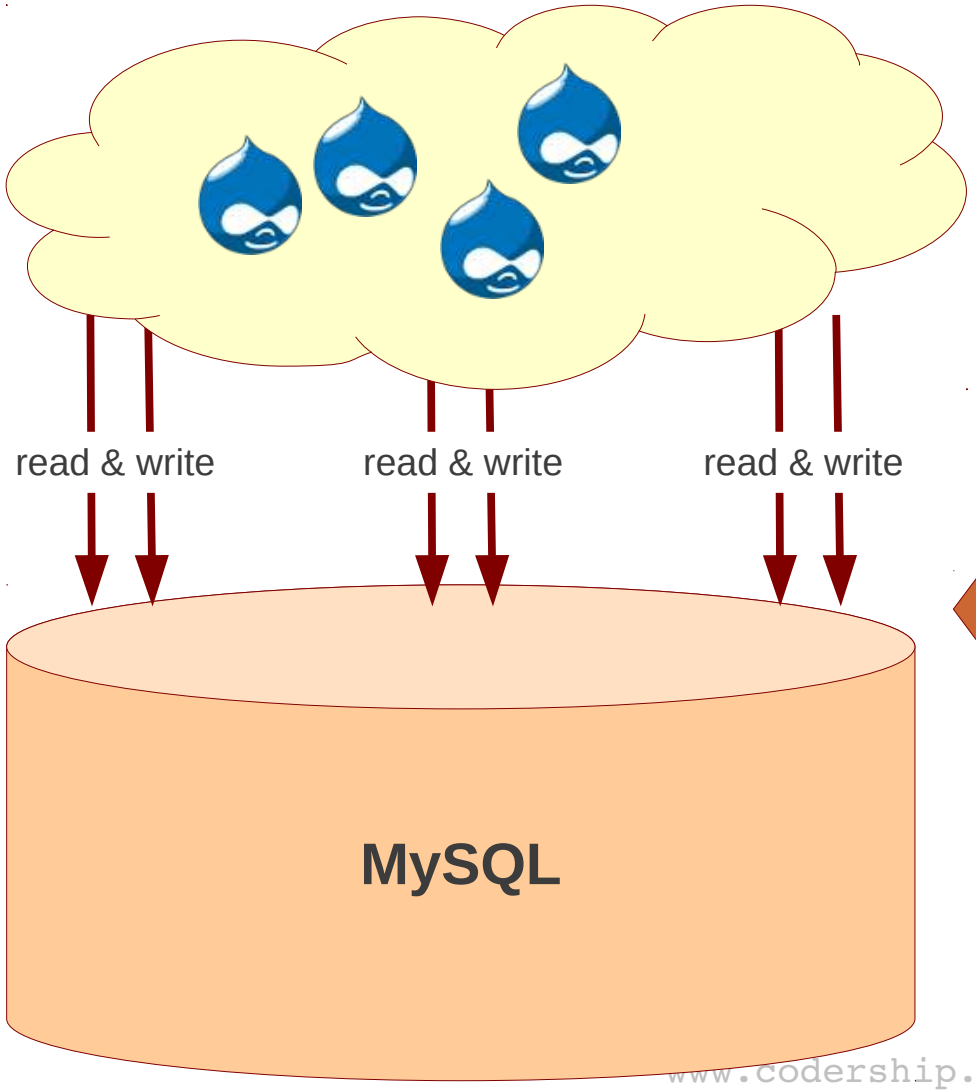
Read & write access to any node

Client can connect to any node

There can be several nodes

Replication is synchronous

Multi-Master Replication

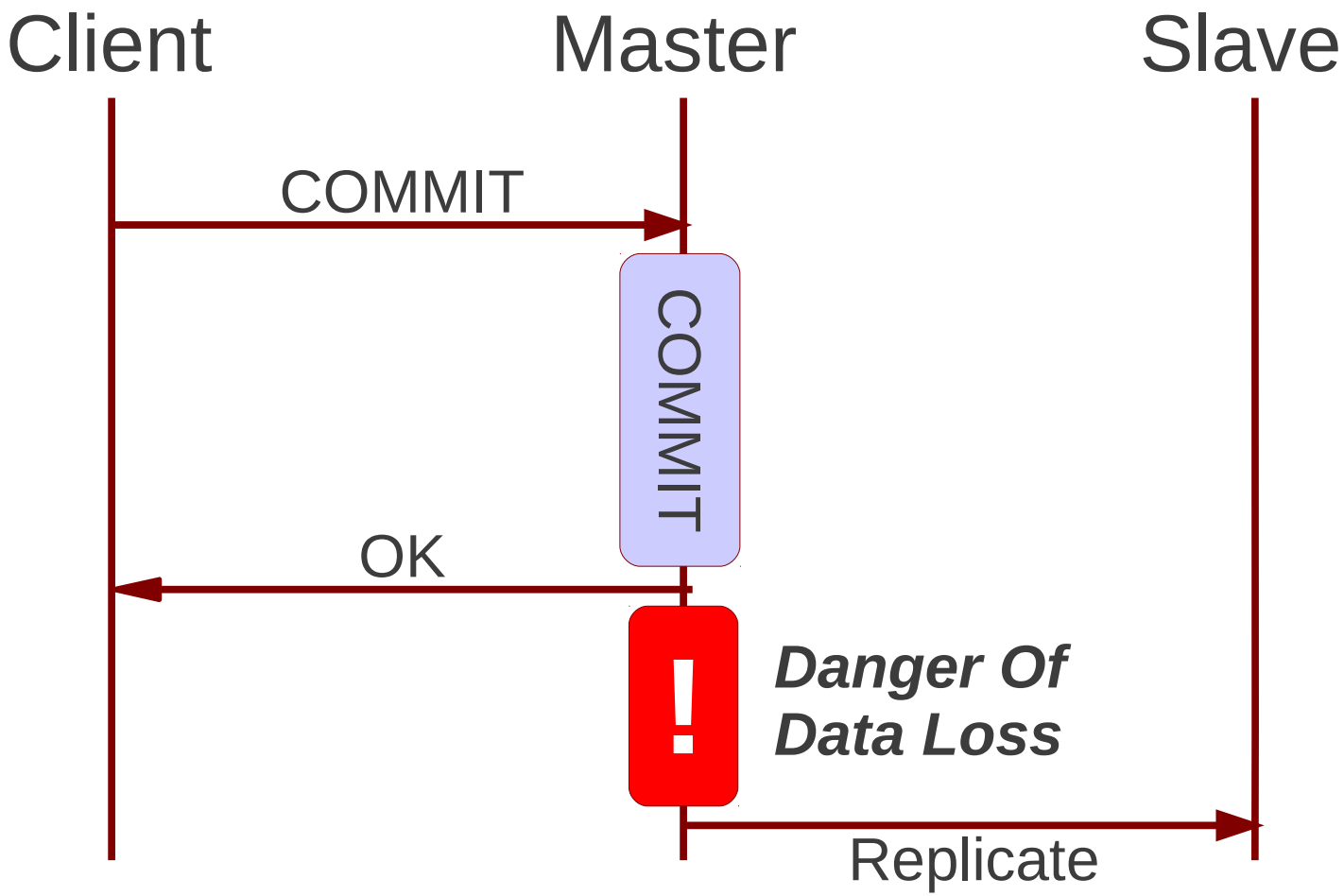


Multi-master cluster looks like one big database with multiple entry points

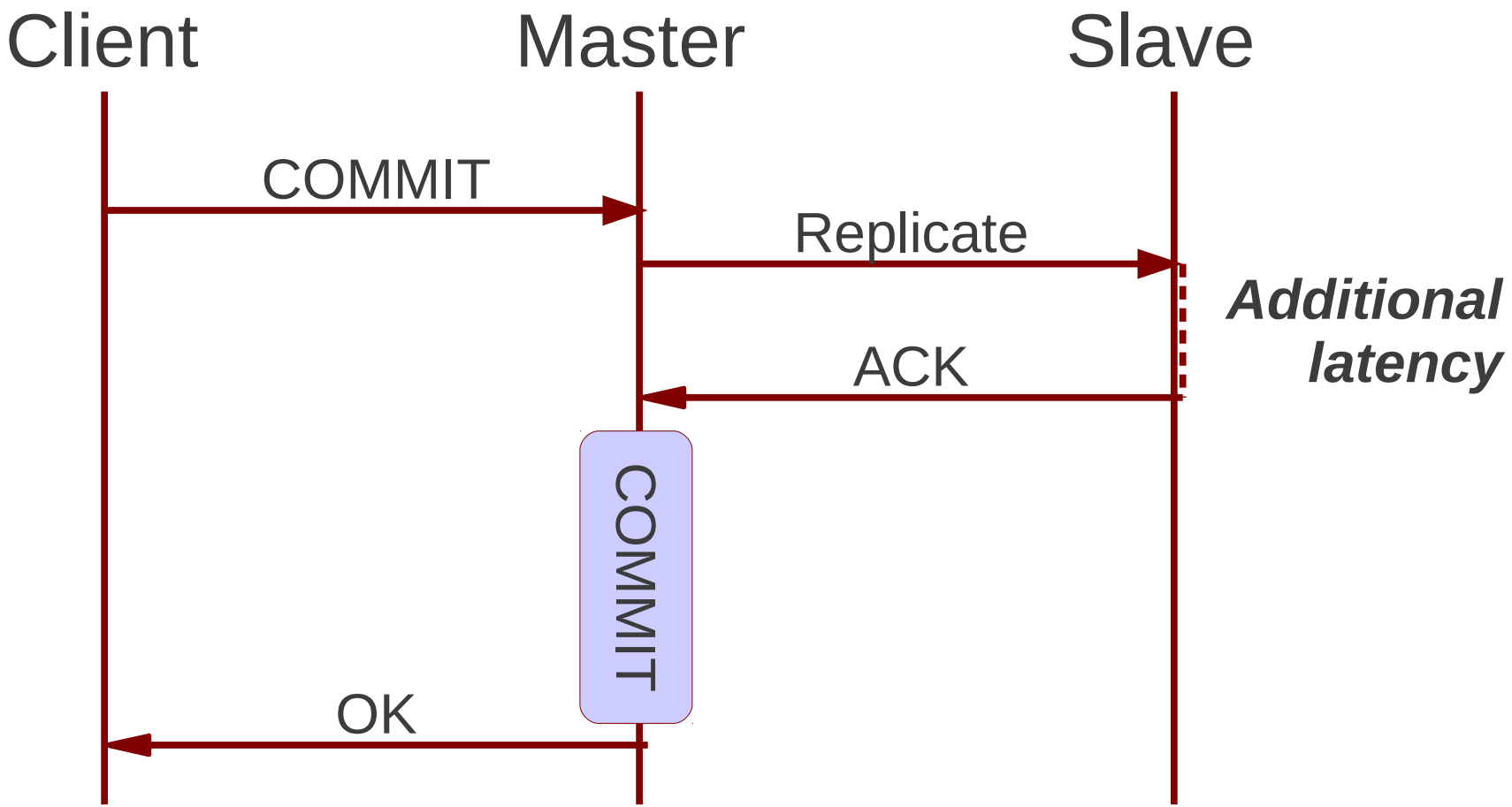
Galera Cluster

- Synchronous multi-master cluster
- For MySQL/InnoDB
- 3 or more nodes needed for HA
- Automatic node provisioning
- Works in LAN / WAN / Cloud

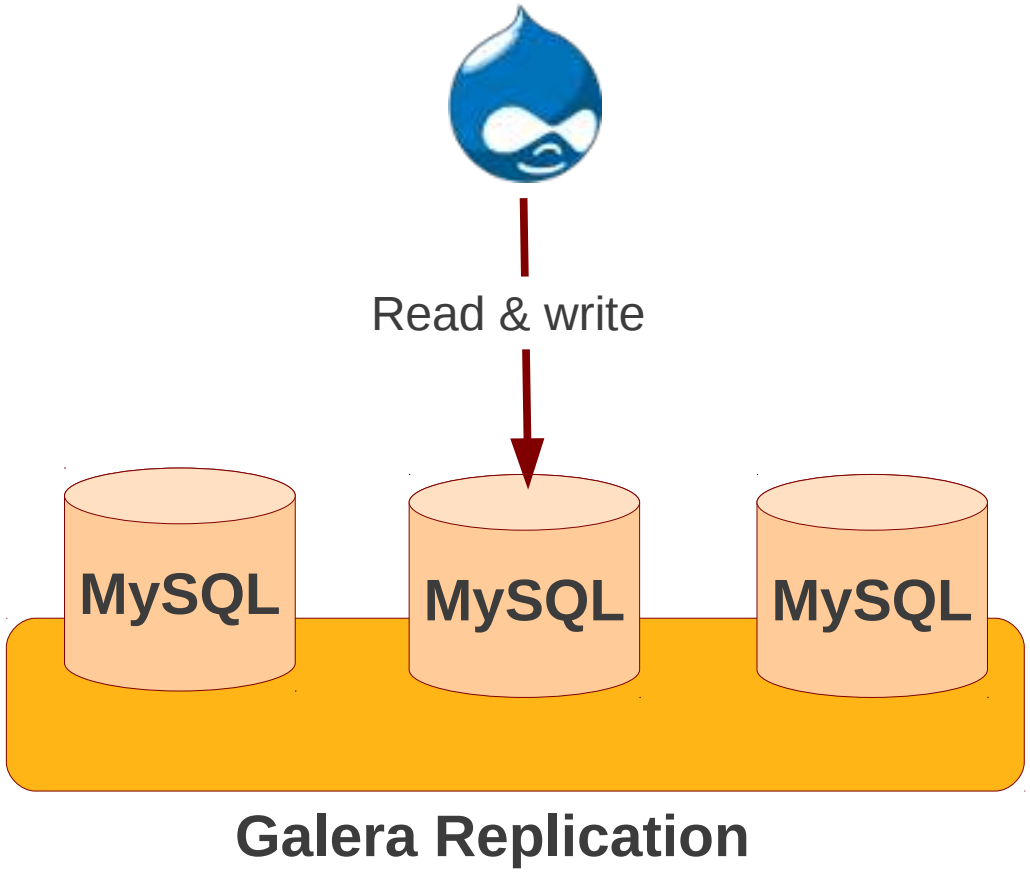
What Is Asynchronous Replication?



What Is Synchronous Replication?

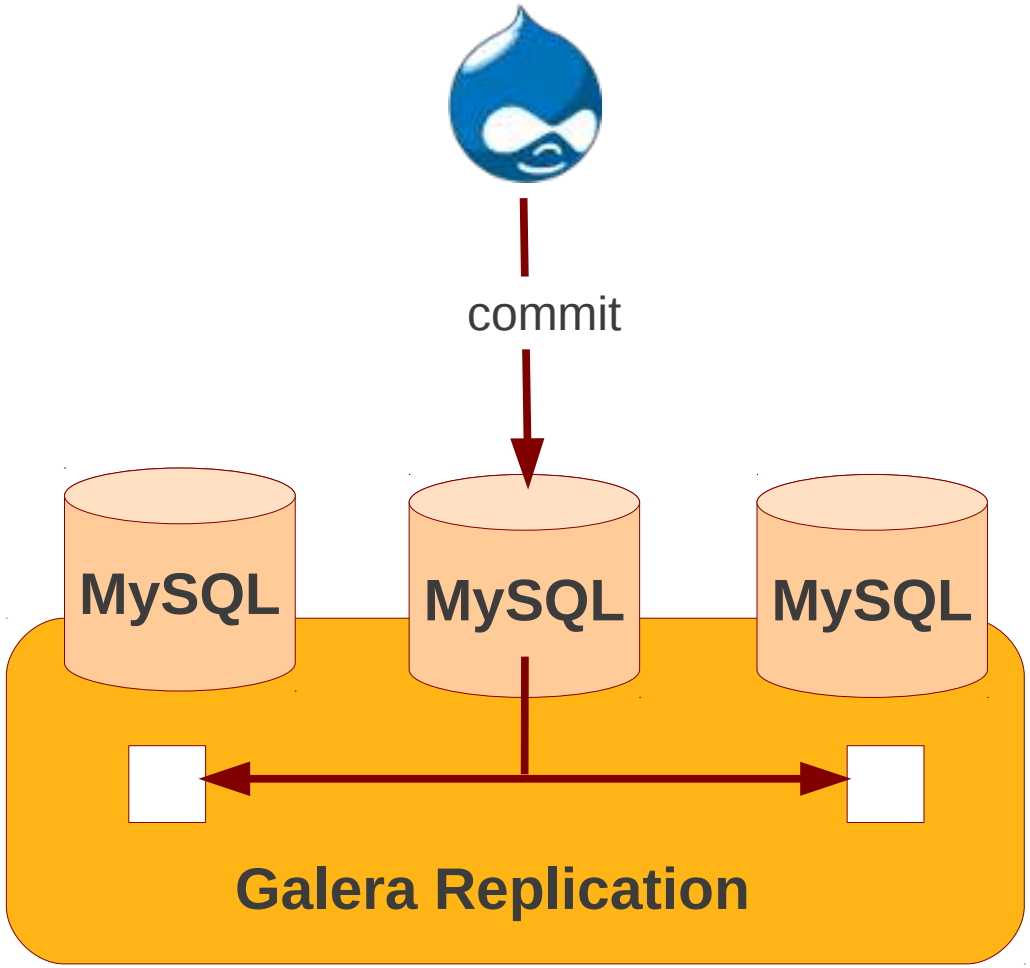


Synchronous Replication



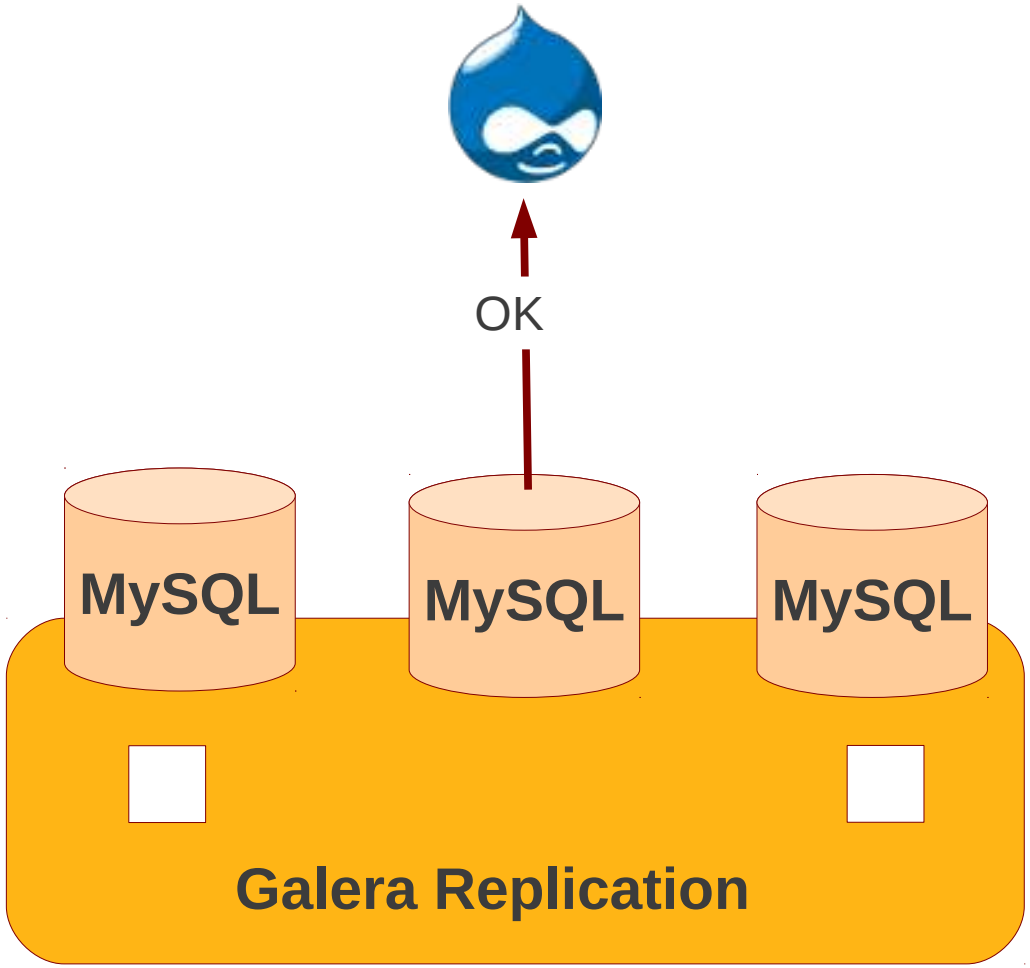
Transaction is processed locally up to commit time

Synchronous Replication



Transaction is replicated to whole cluster

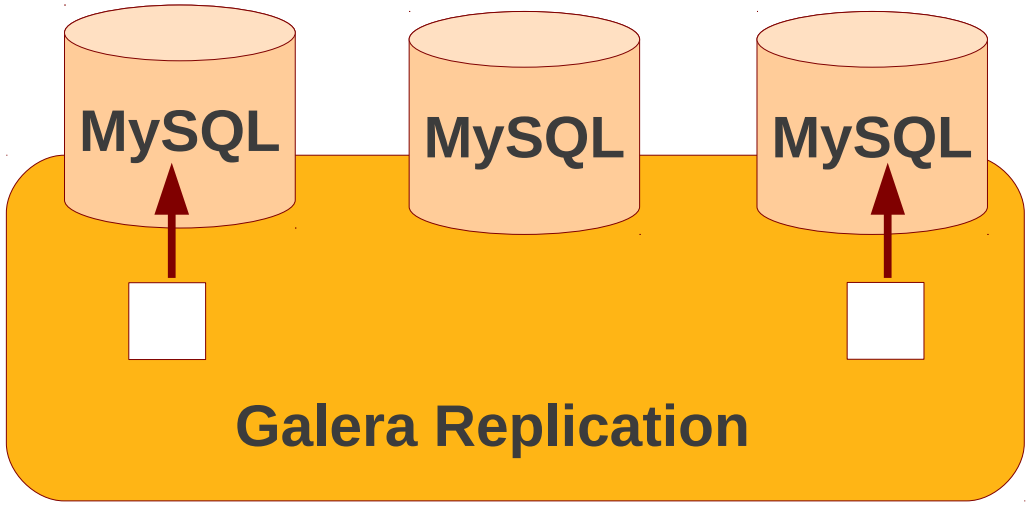
Synchronous Replication



Client gets OK status

Synchronous Replication

Transaction is applied in slaves



Galera Cluster

Each node is full representative of the cluster

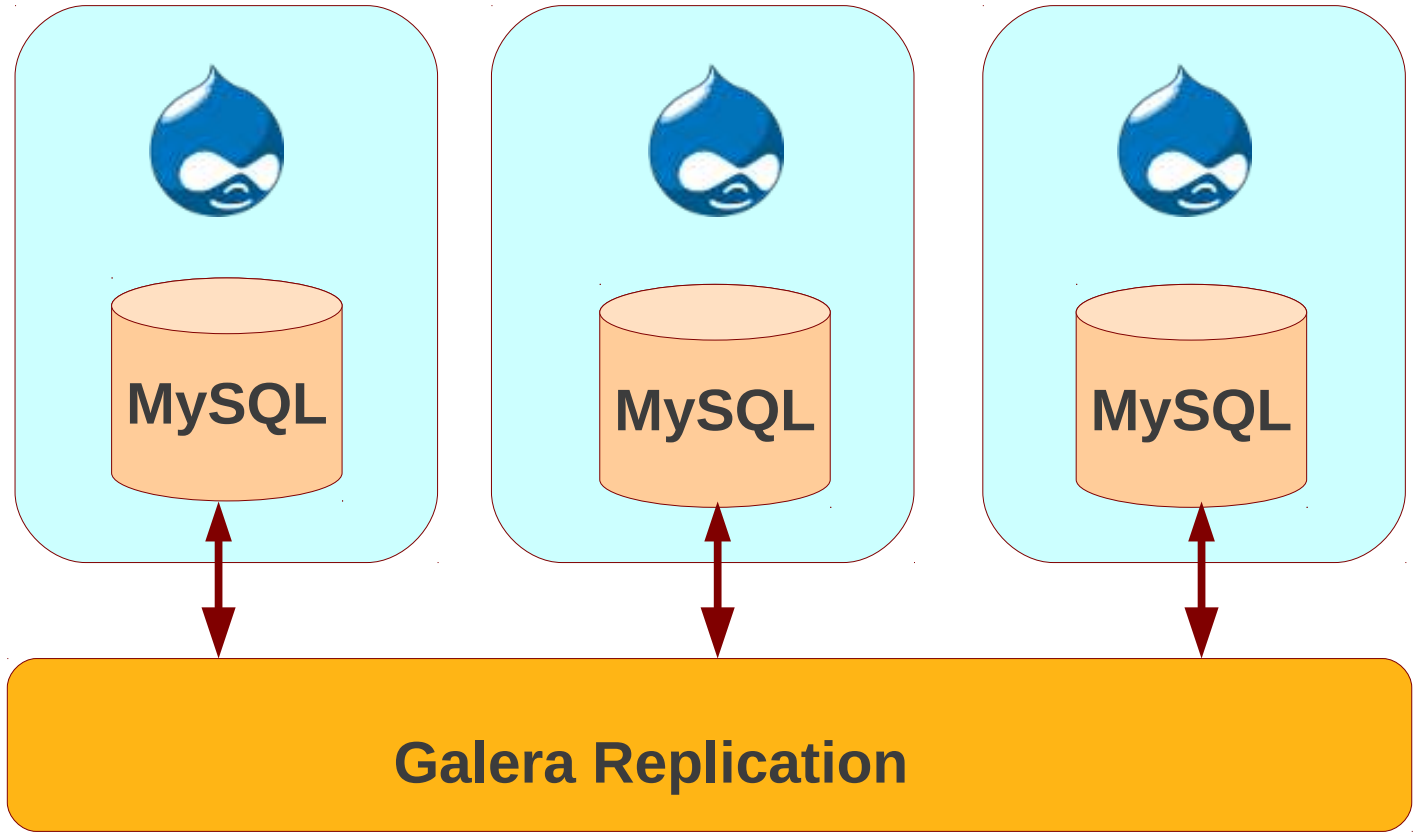
- No single point of failure
- Synchronous...
  - No Data Loss
  - No Slave Lag
  - No Master Failover
- 99.99% transparent
  - InnoDB look & feel

Cluster Topologies

Galera Cluster Topologies

- Application and Galera node in same server
- Separate Galera Cluster
- WAN Clustering
- Inter-operating with MySQL replication

Application + Galera in Same Server



Application+ Galera in Same Server

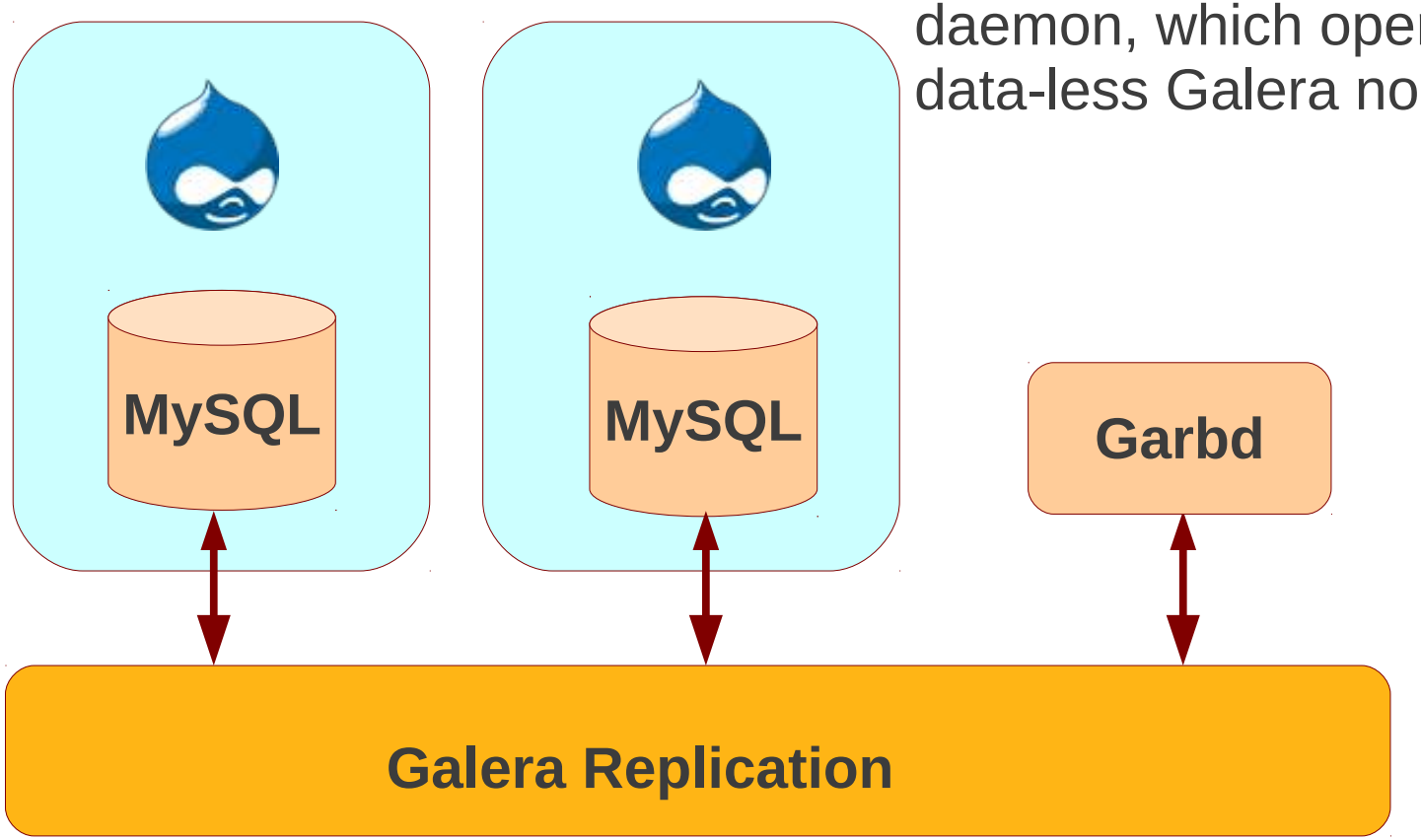
- Easy to install, all nodes are identical
- Application has local access to DB
- Easy to scale

- Application & MySQL compete of same resources
- There is upper limit for scalability
- MySQL could serve many application servers

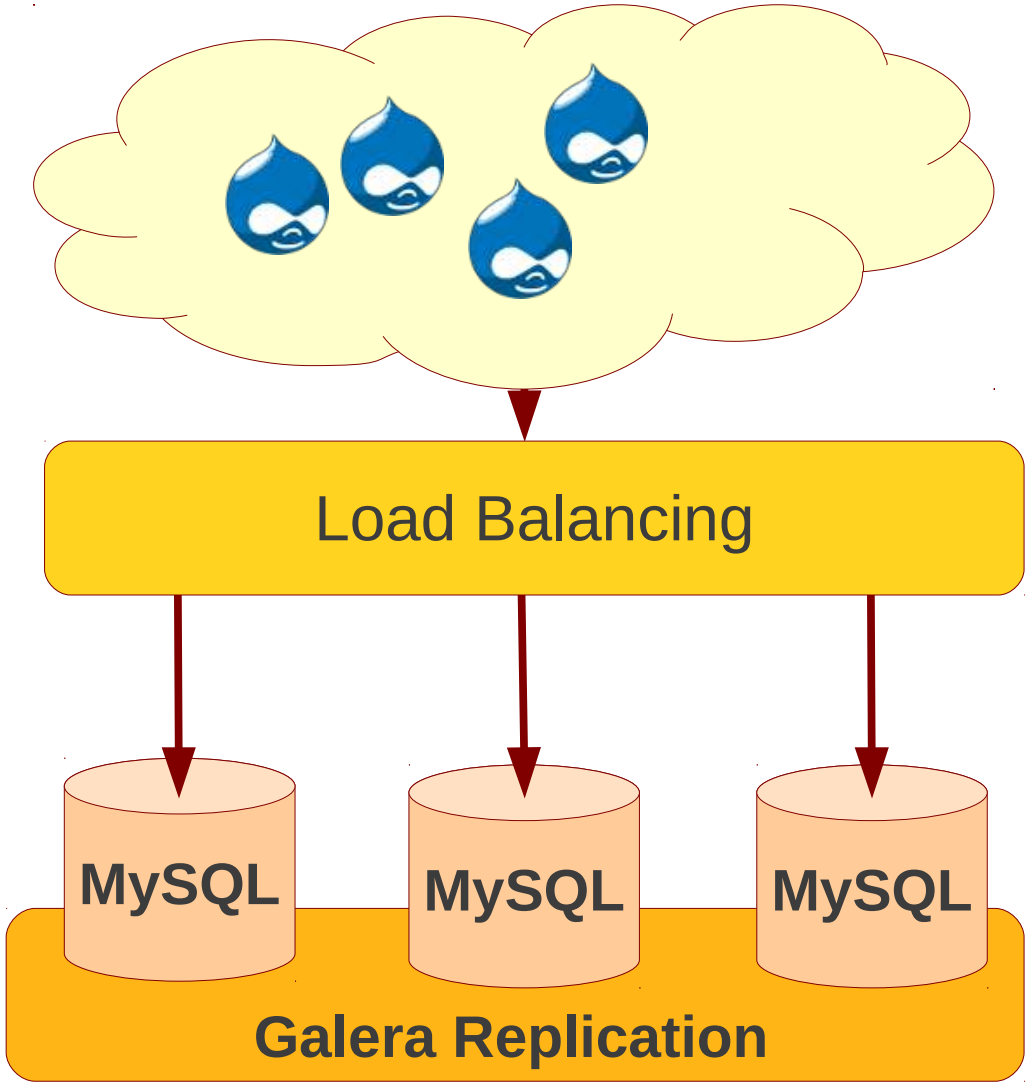
Minimal deployment, 2 Galera Nodes

HA requirement is to have 3 nodes minimum

Garbd is light weight arbitrator daemon, which operates as a data-less Galera node



Separate Galera Cluster

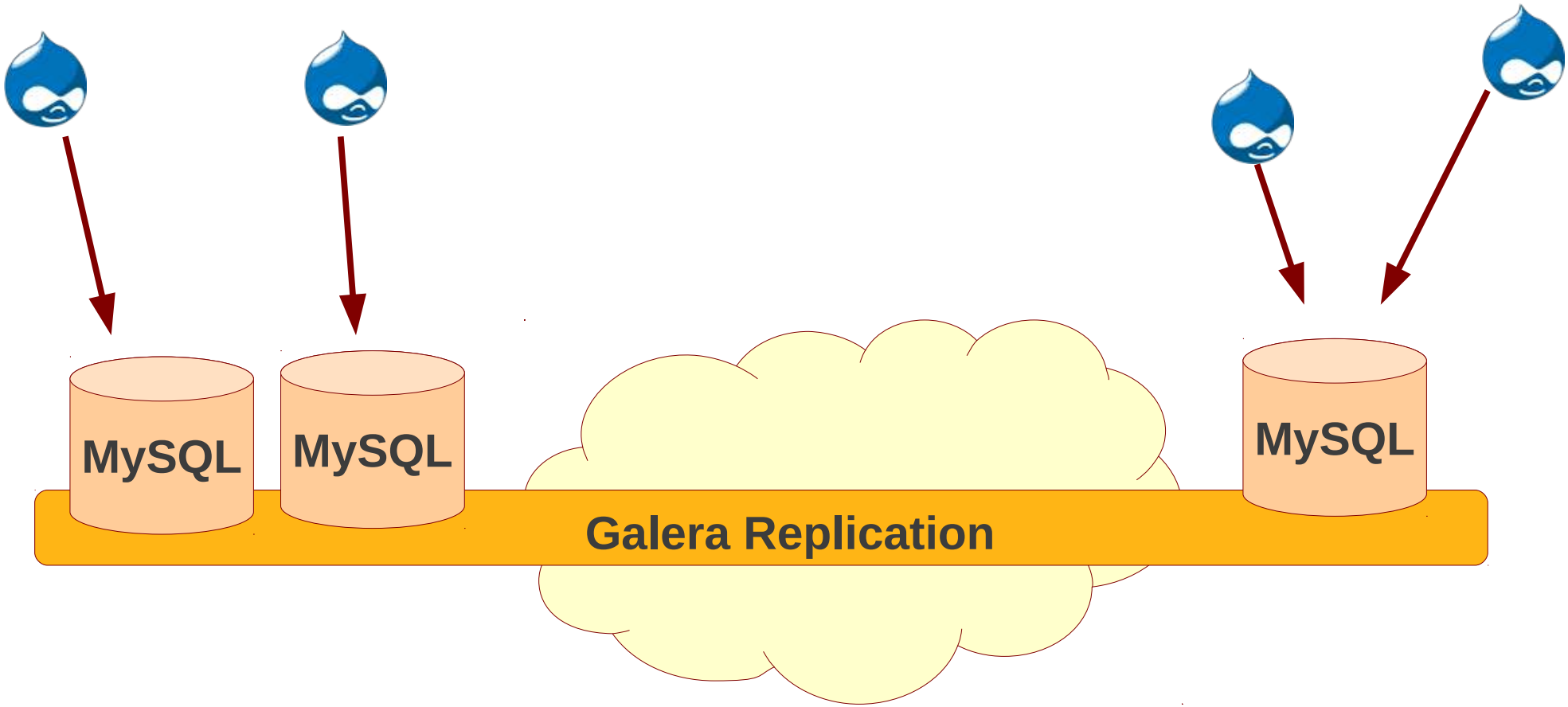


Separate Galera Cluster

- Best use of resources, Galera node can be optimized for MySQL
- Small Galera cluster can serve large application Farm
- Optimal for scalability

- One extra hop for MySQL
- Load balancing needed

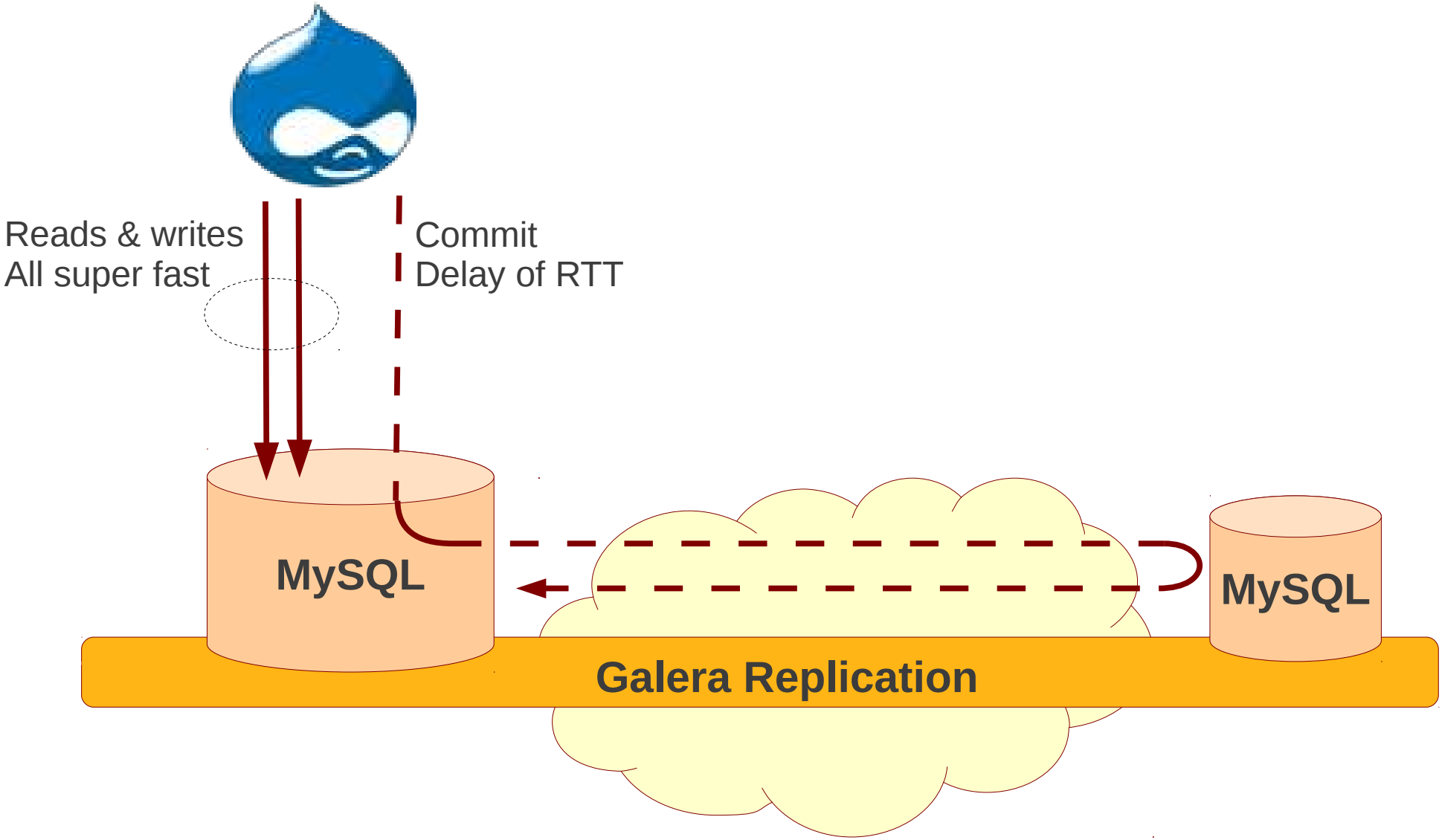
WAN Cluster



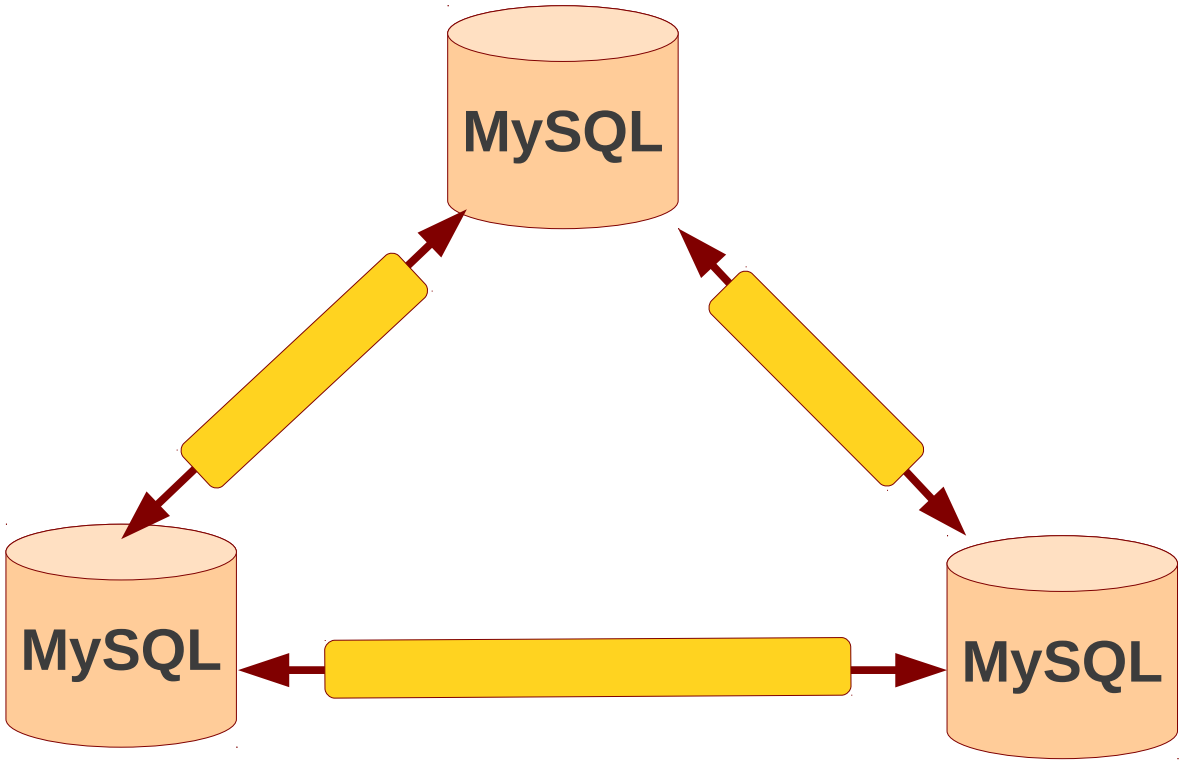
WAN Replication

- WAN latencies affect only commit operation
- Local reads and writes are fast
- Network instability can cause node drops
  - Disconnected node can join back fast with incremental state transfer

WAN Cluster



Replicating over SSL



Connecting to Galera Cluster

Connecting to Galera

- If application and Galera node reside in same server, connection can be just local
- Load balancer can work as single connection point for external Galera cluster
- Load balancing drivers:
  - For PHP: mysqlnd_ms
  - For java: connector/J

Load Balancers for Galera

- Any load balancer will do
  - However, better be runtime configurable
  - Even better if LB can monitor node status
- Hardware LB for performance
- Software, user land load balancer can spend a lot CPU
  - Haproxy, glbd

Load Balancer Location

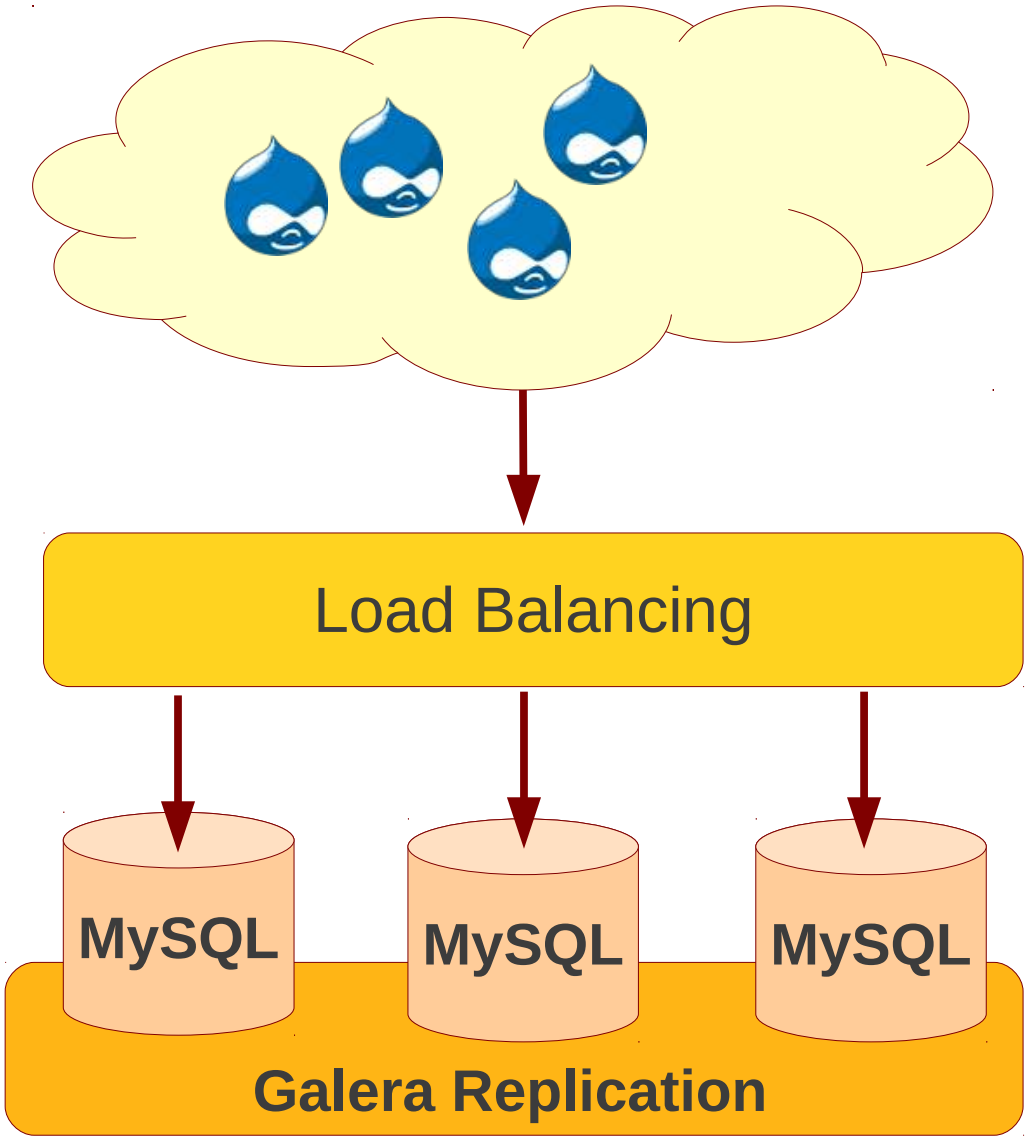
1) Together with application in same server

- No extra hop
- MySQL privilege system works
- Competes of CPU with application
- There may be a big number of LBs to configure

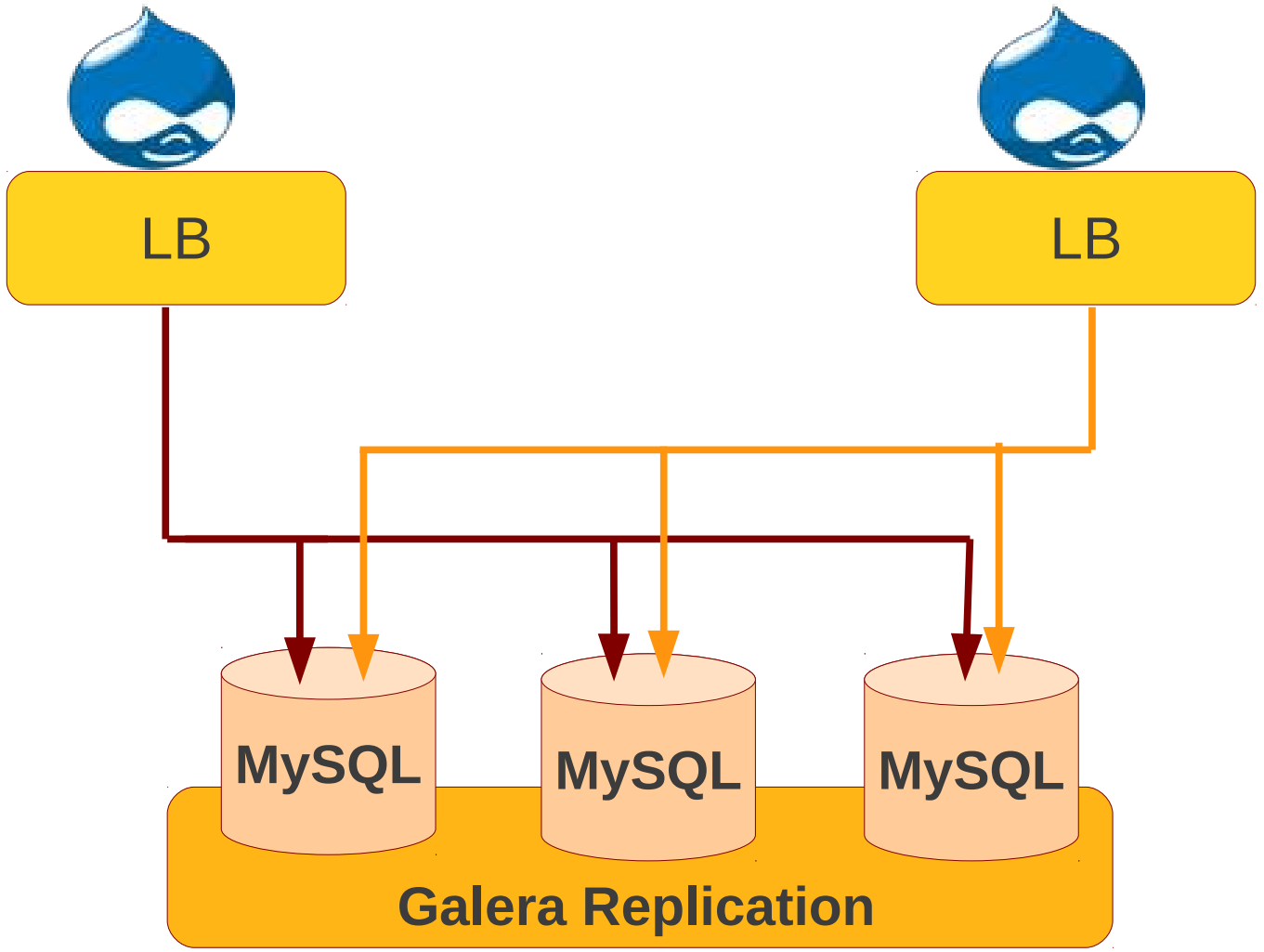
2) In separate server

- Can serve several application servers
- Extra hop needed
- Problem with privileges

Load Balancer Location



Load Balancer Location



Cluster Notifications

- Cluster can trigger notifications
- Script API for notification syntax
- wsrep_notify_cmd defines the script to handle notifications
- Use for:
  - load balancer configuration
  - monitoring

PHP and Java Connectivity

PHP: mysqlnd_ms

- MySQL native driver for PHP
  - Replacement for MySQL client library
- Read/write splitting
- Multi-master mode
- Several load balancing strategies

Java: Connector/J

- Oracle driver for Java
- Happens to support multi-master mode
  - jdbc:mysql:loadbalance://host-1,host-2,...host-n/database...

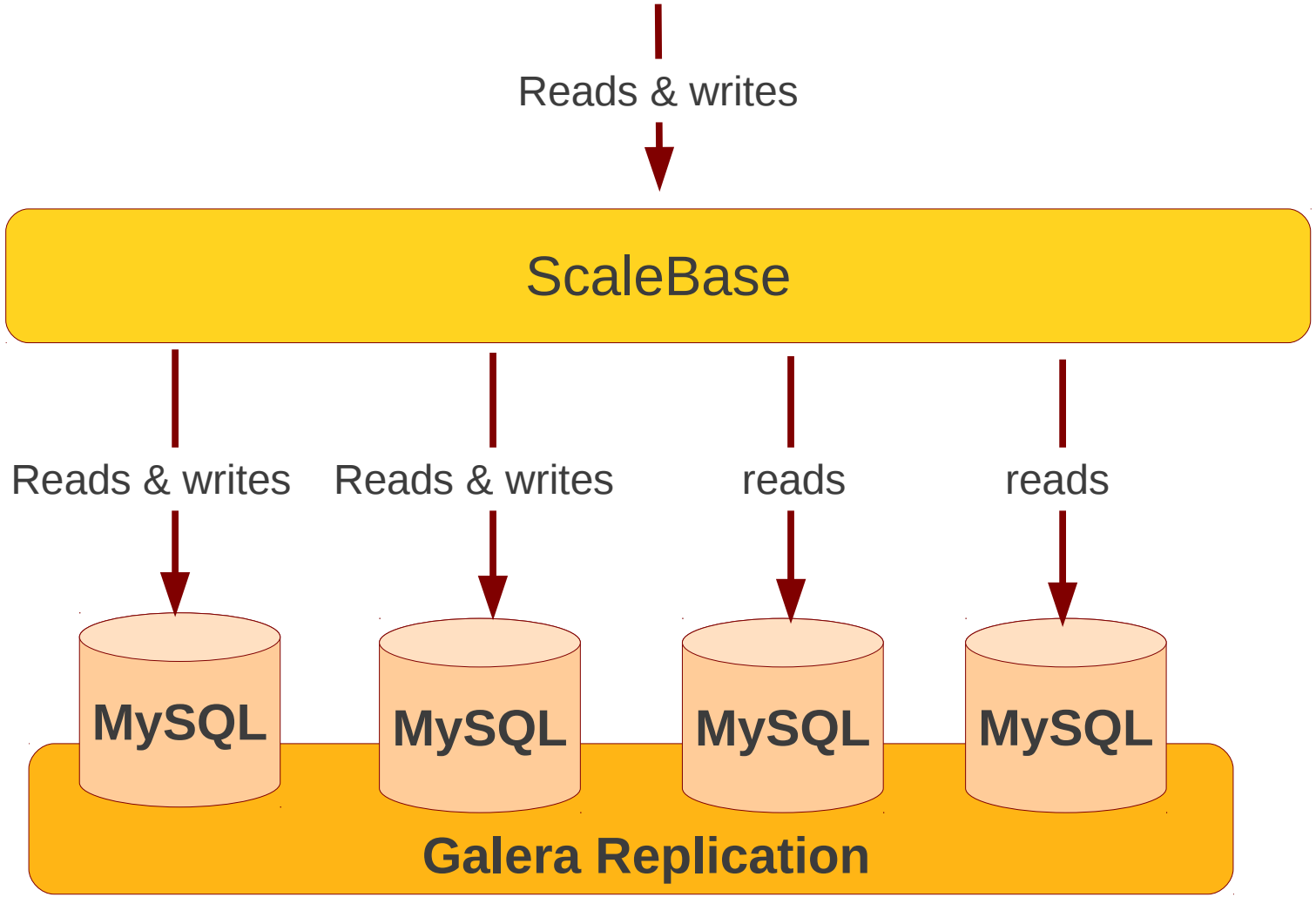
Connecting though Proxy

- Proxy is a middleware component which can react / interact with client protocol
- Generally somewhat slower than load balancing
- Can add extra intelligence for balancing
- MySQL Proxy
  - LUA scripting
  - Alpha/beta level
  - High overhead
- ScaleBase
  - efficient, very low overhead (8-18%)
  - Integrated with Galera
  - Can do sharding as well

ScaleBase

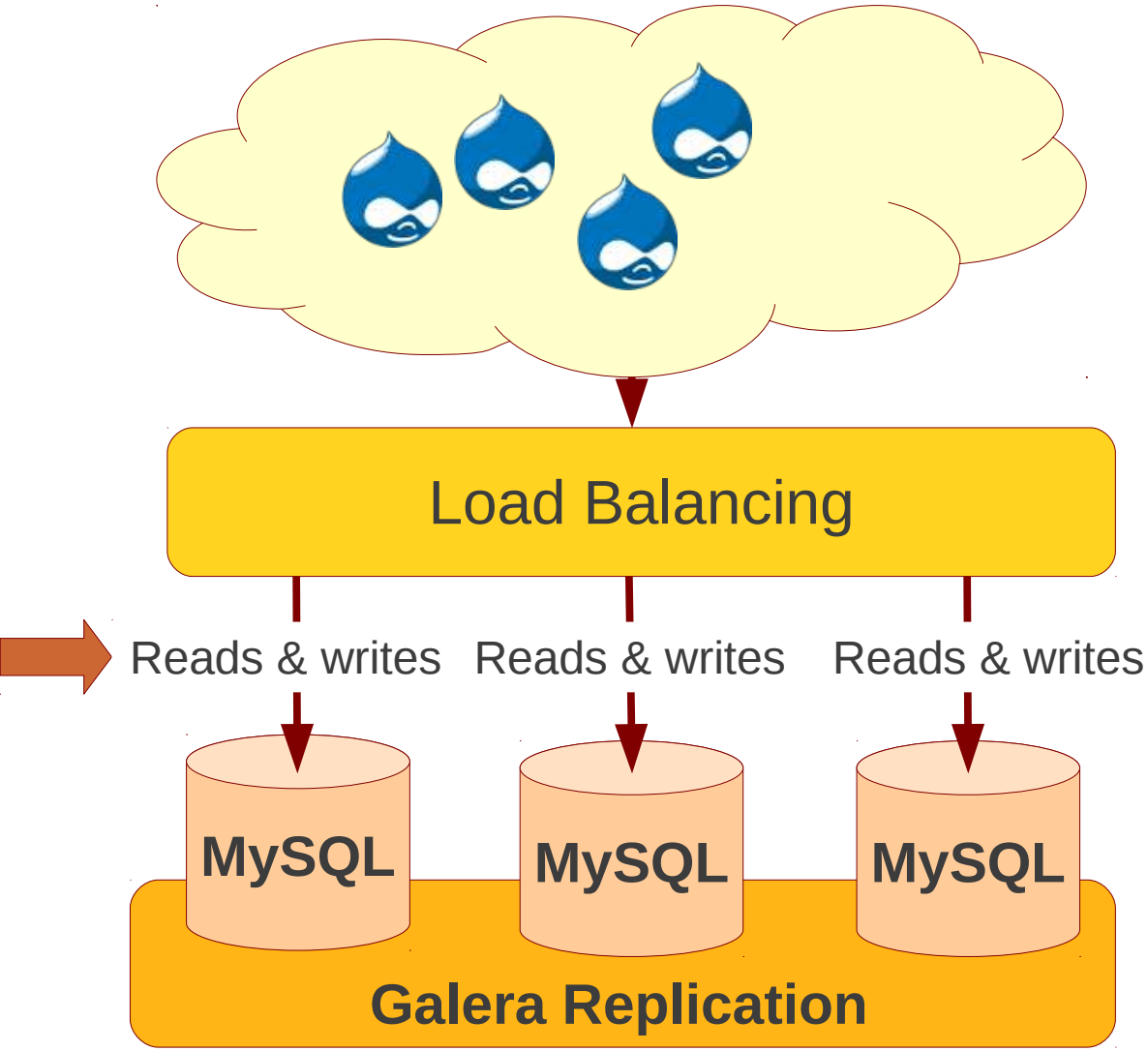
- Technology partnership between ScaleBase & Codership
- Read-write splitting
- sharding
- Can do hot-spot preventing

ScaleBase


Reads & writes
ScaleBase
Reads & writes
Reads & writes
reads
reads
MySQL
MySQL
MySQL
MySQL
Galera Replication

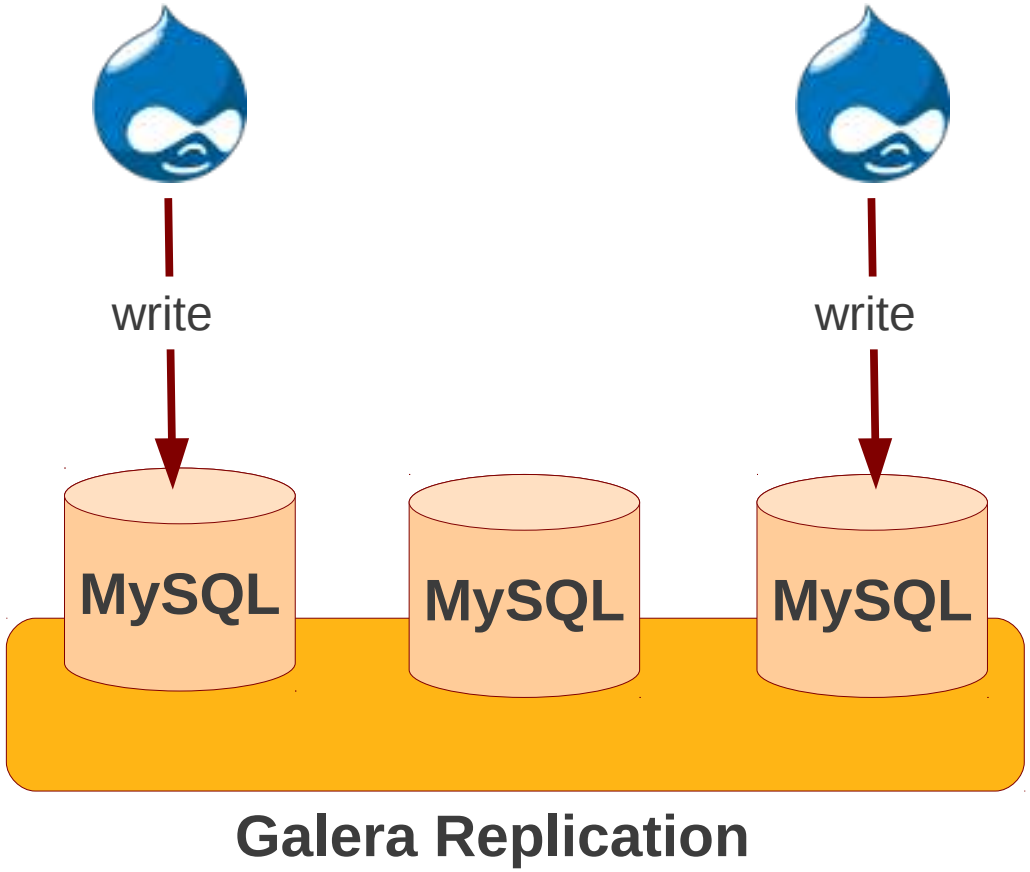
Use Cases

Multi-Master

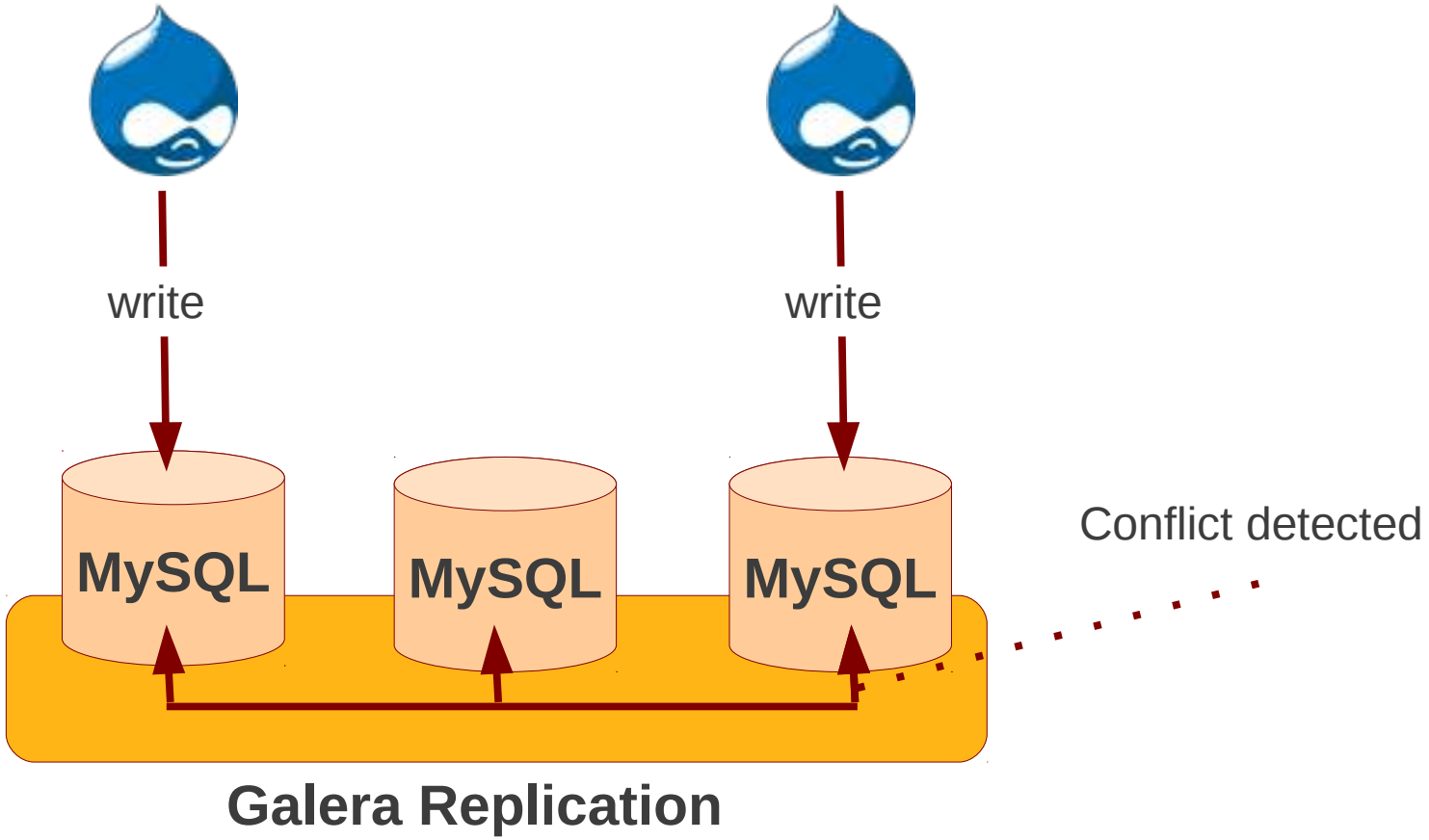


- Read & write access to any node
- Simple for LB
- Can scale writes
- Prepare for write-write conflicts

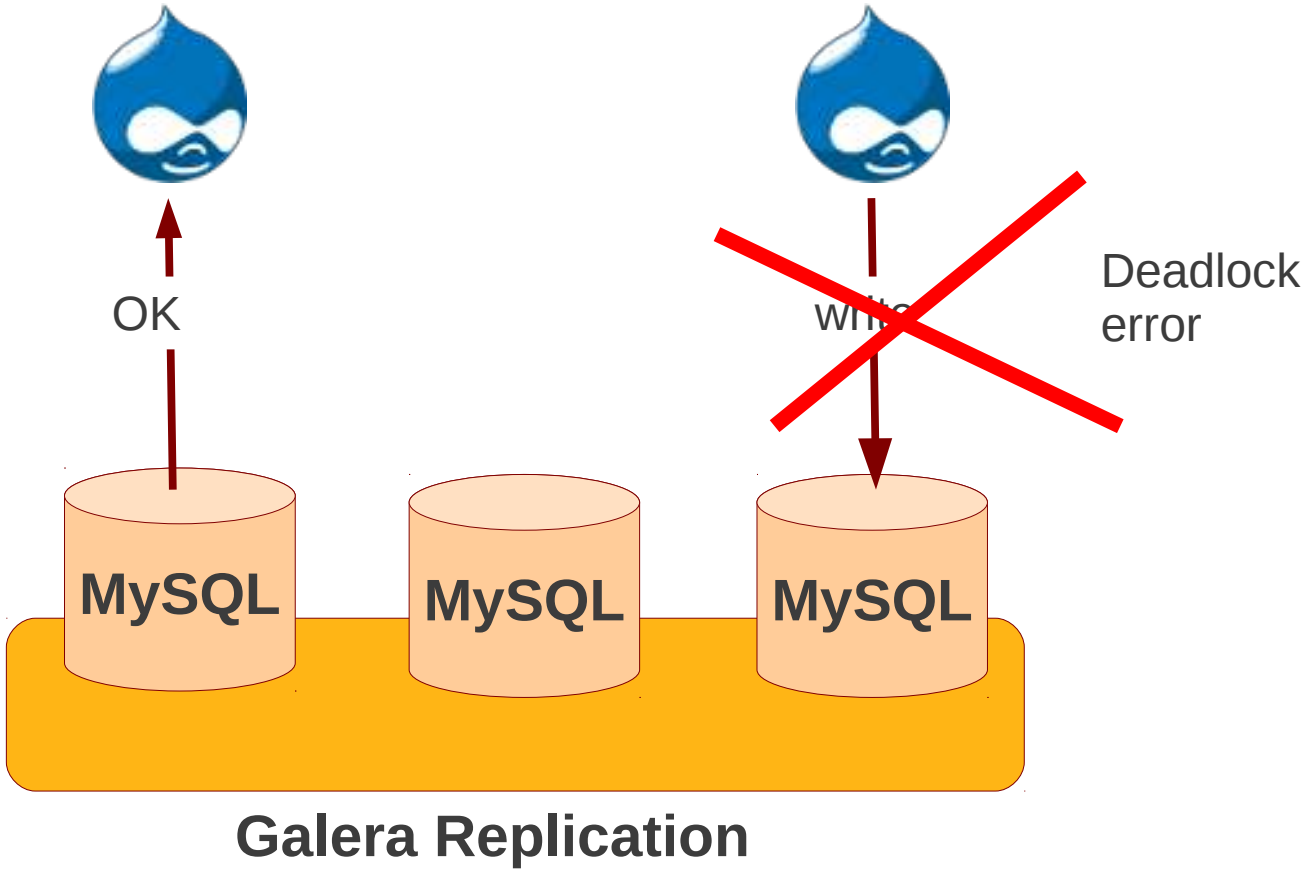
Multi-master Conflicts



Multi-master Conflicts



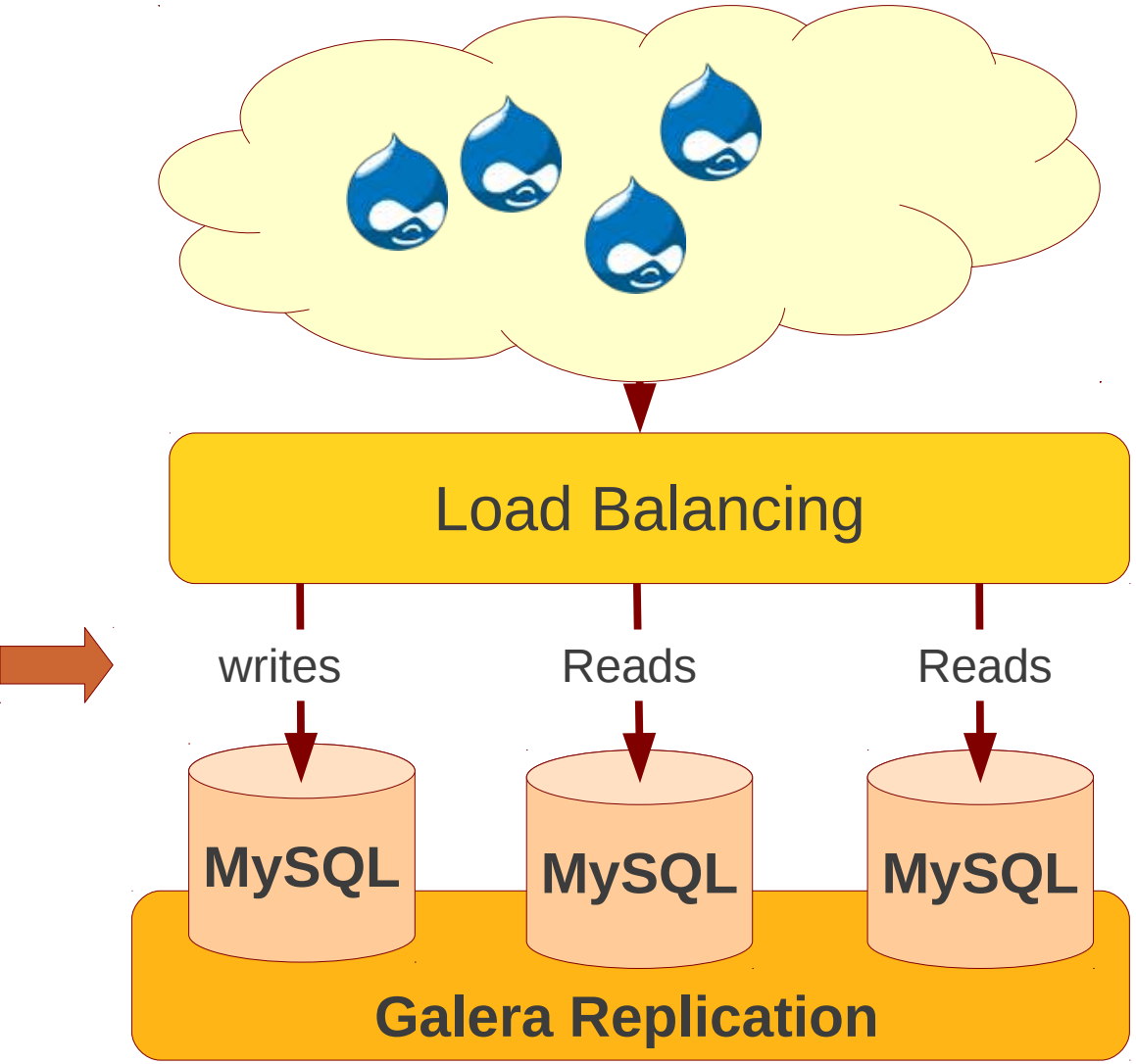
Multi-master Conflicts



Multi-Master Conflicts

- Galera uses optimistic concurrency control
- If two transactions modify same row on different nodes at the same time, one of the transactions must abort
  - → Victim transaction will get deadlock error
- Application should retry deadlocked transactions, however not all applications have retrying logic inbuilt

Master-Slave

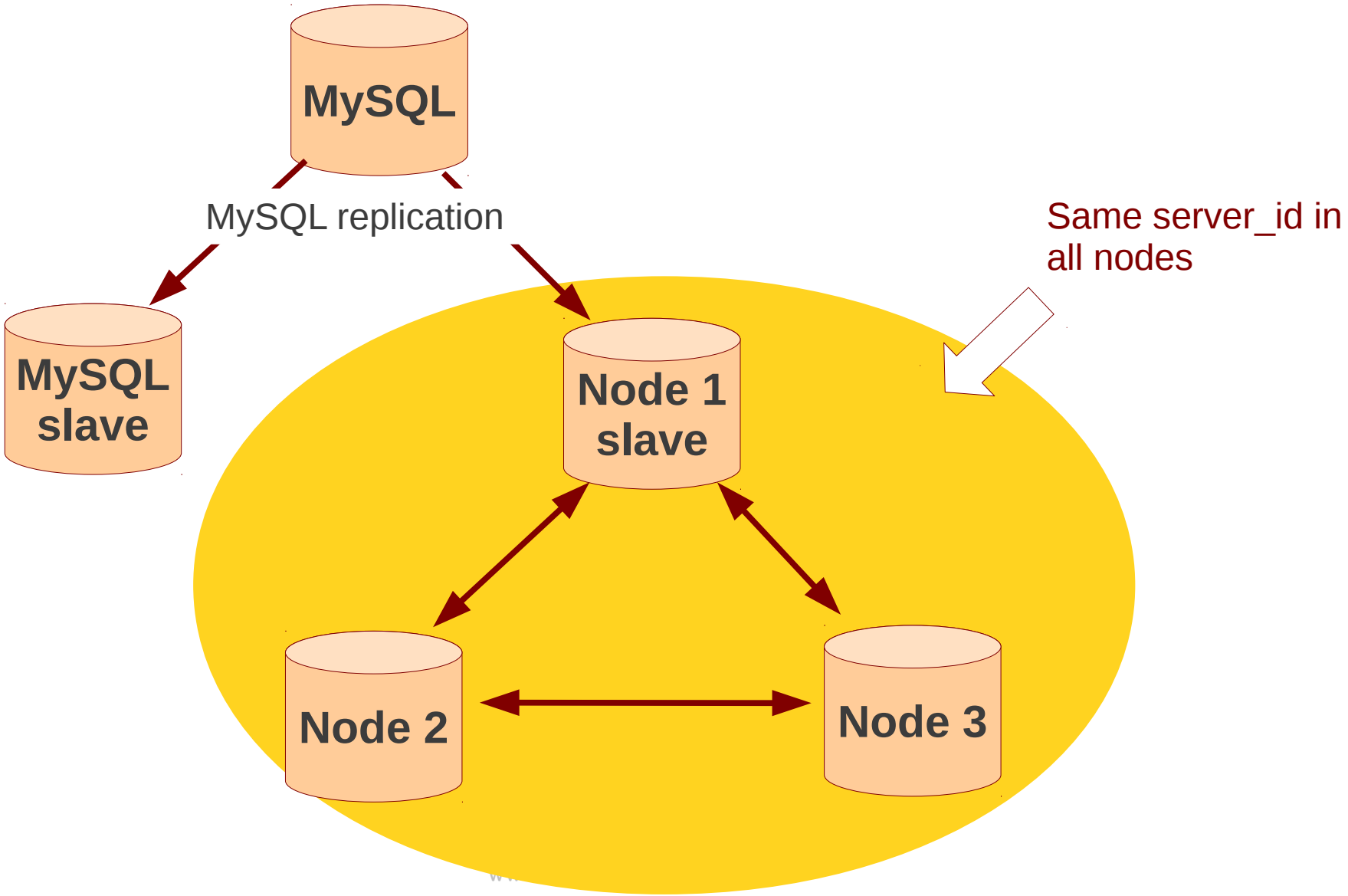


- Writes to dedicated master node
- Any node can be picked as master, any time
- Reads from any node
- Read scalability
- No data loss
- No slave lag

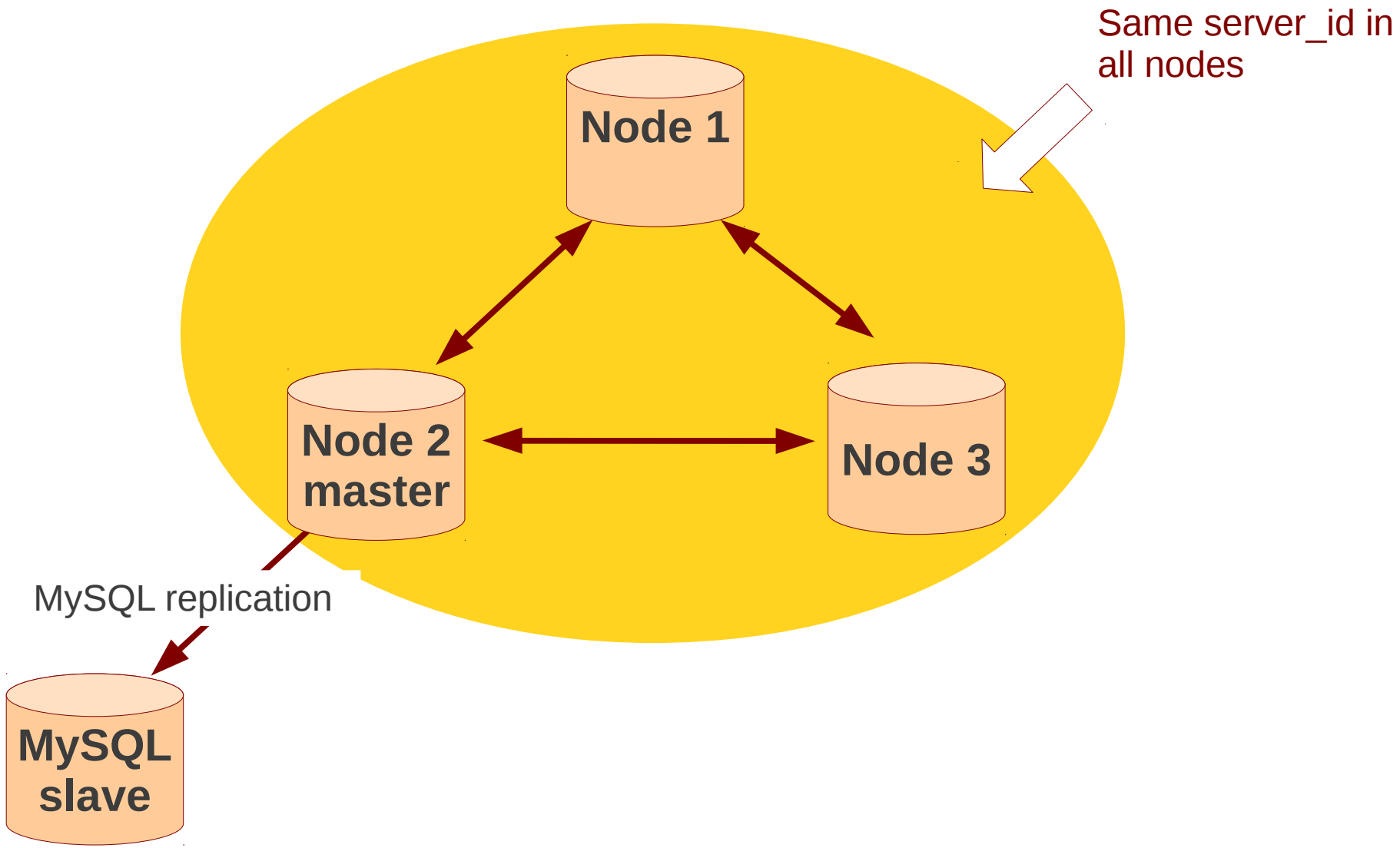
MySQL Replication Use Cases

- Galera node can be used both as MySQL slave and master
- Use:
  - log_slave_updates
  - same server_id in Galera cluster
- Galera cluster in slave role is slower than MySQL slave

Galera as MySQL slave



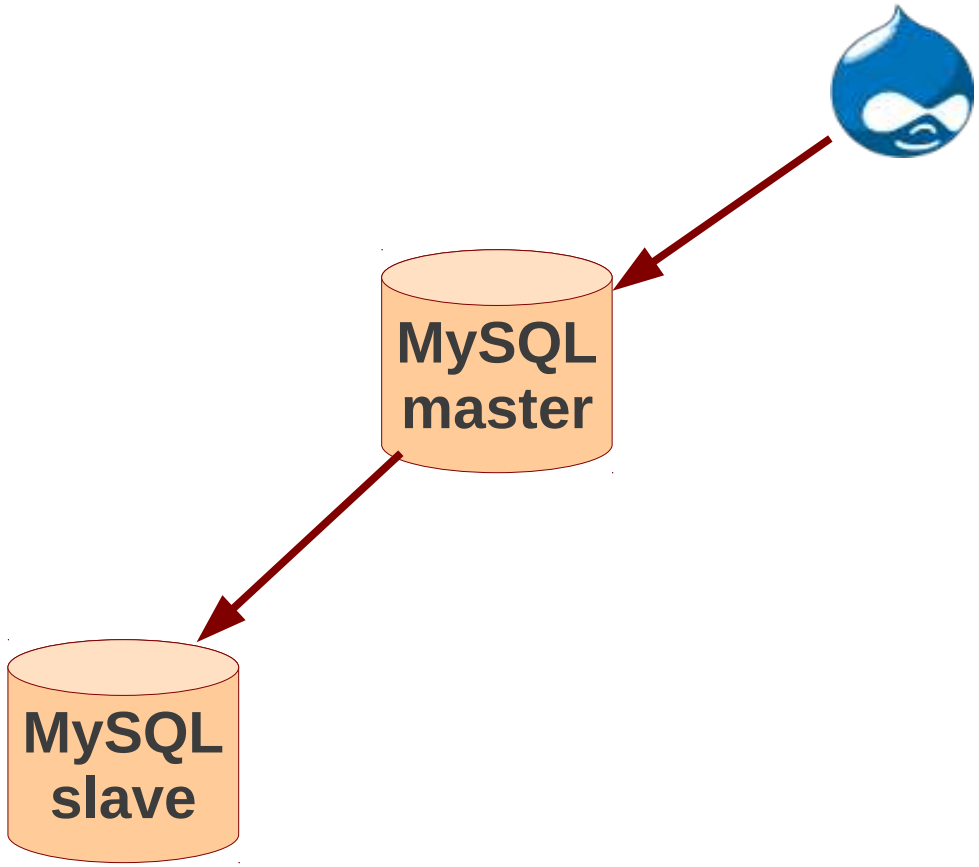
Galera as MySQL Master



Galera as MySQL Master

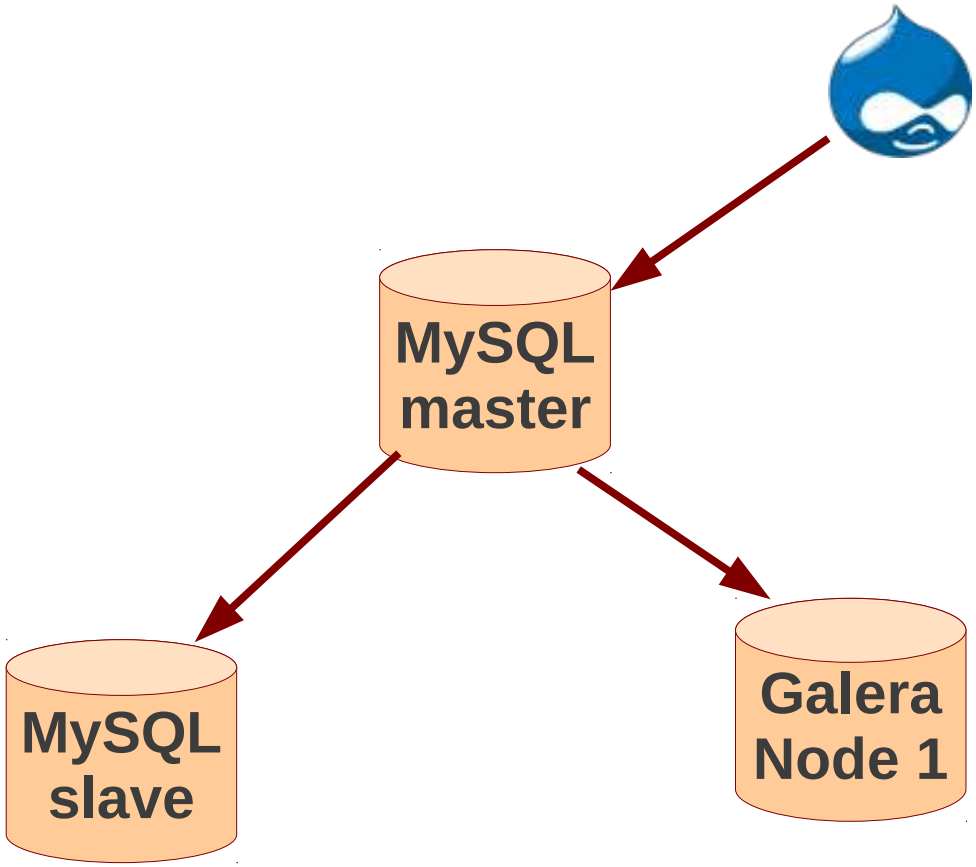
- Fail over to new Galera master is manual operation
- MySQL slaves should use ROW format (same as Galera Cluster)

Migrating to Galera Clustering



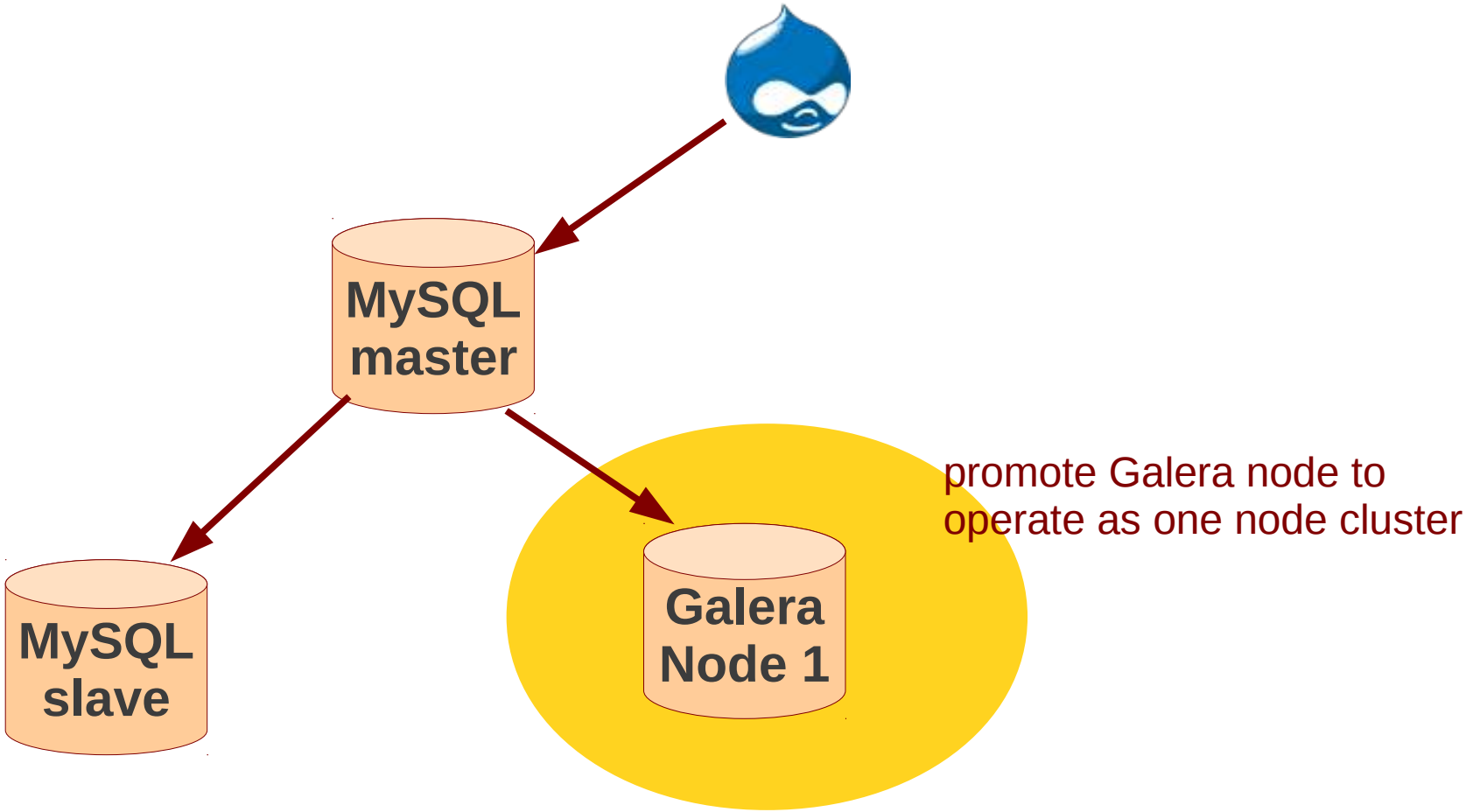
MySQL master slave hierarchy

Galera as MySQL slave

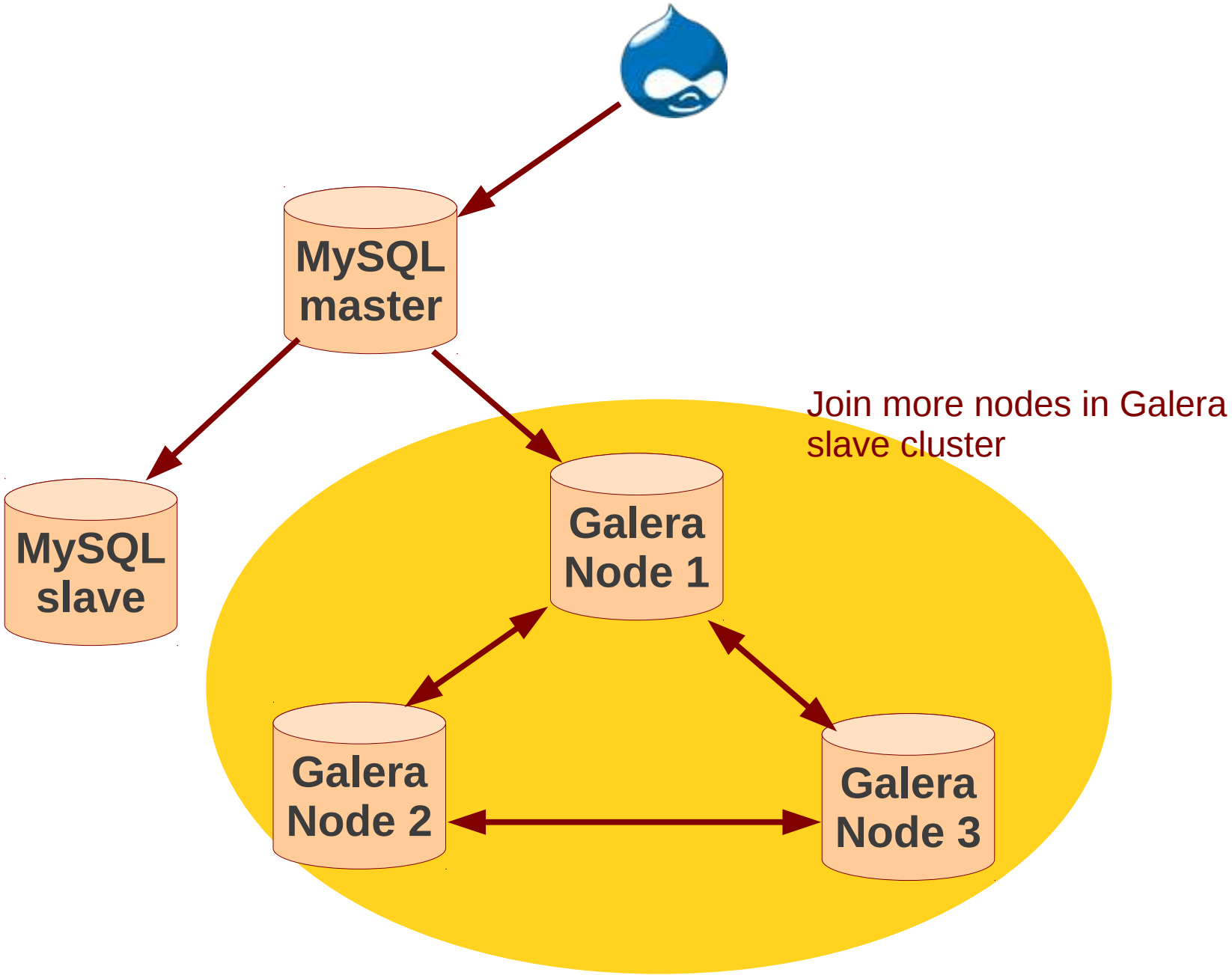


Install first Galera node to operate as MySQL slave

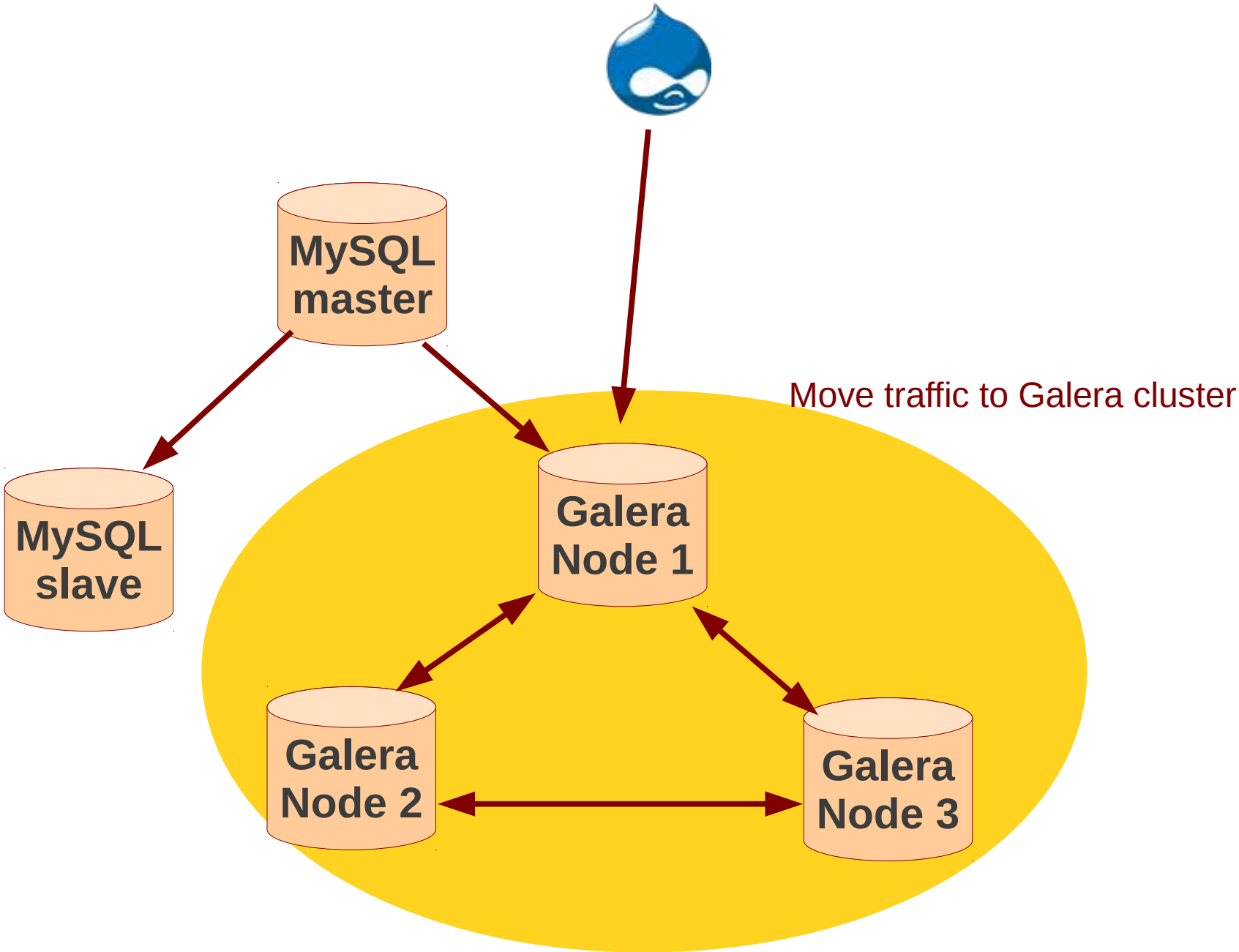
Galera as MySQL slave



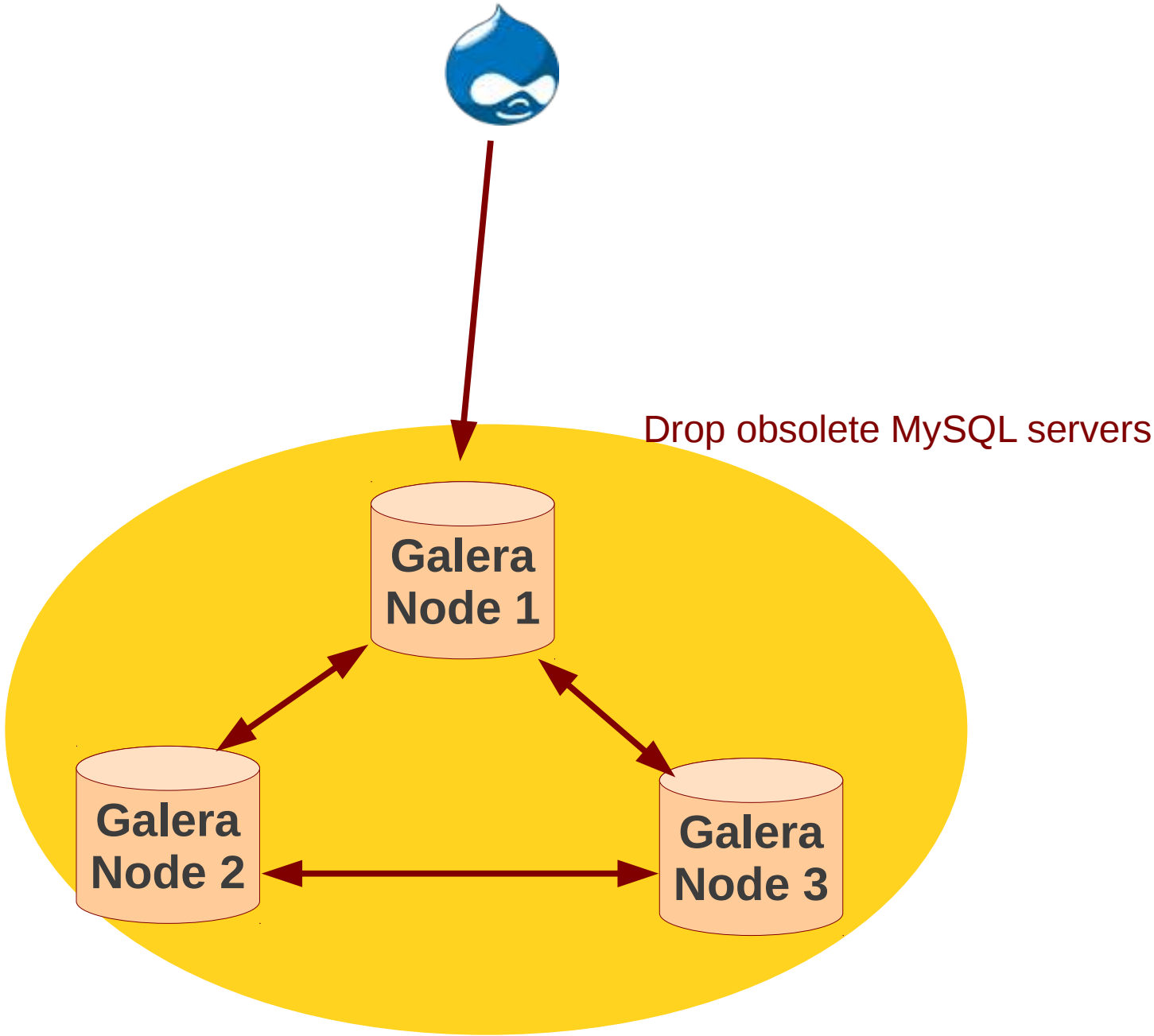
Galera as MySQL slave



Galera as MySQL slave



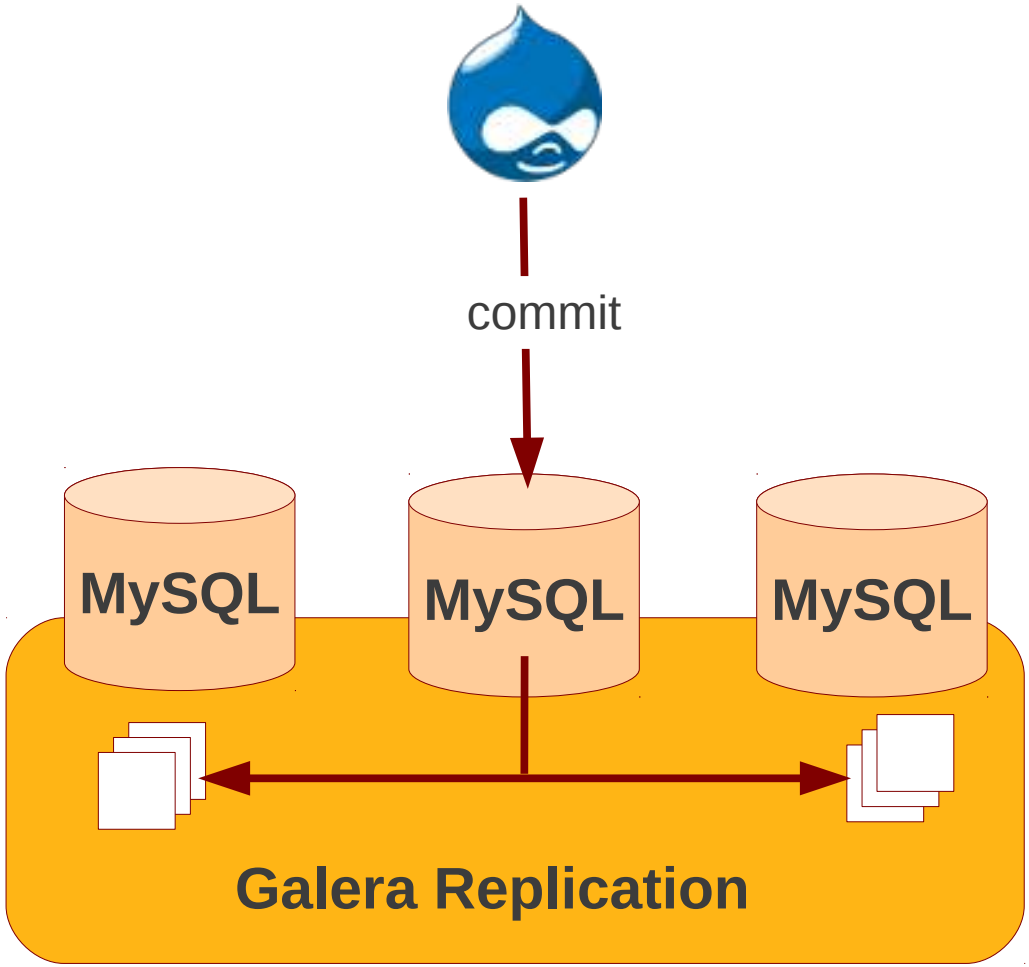
Galera as MySQL slave



Consistent Reads

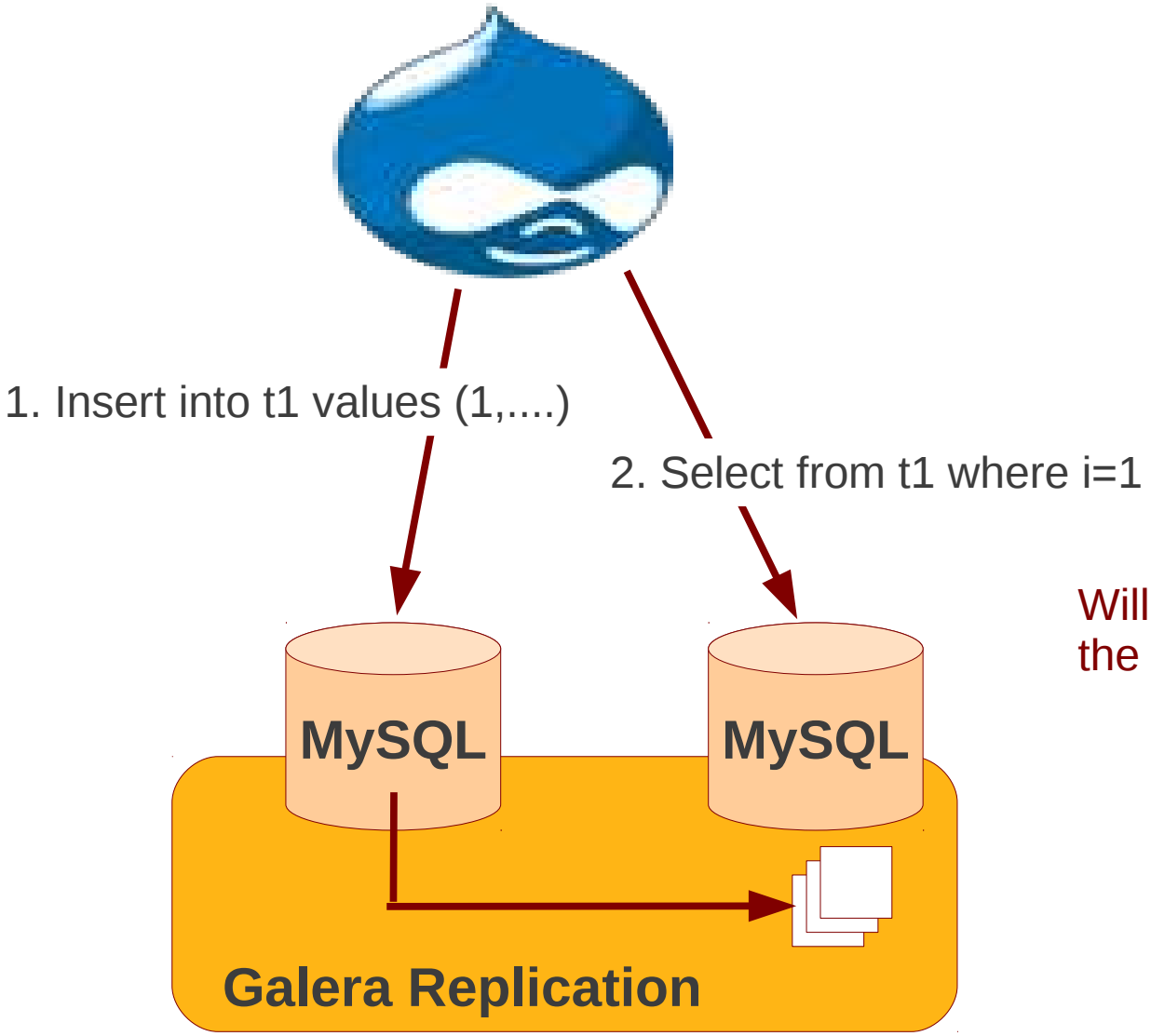
Consistent reads

Replication is virtually synchronous...



Transaction is replicated to whole cluster

Consistent reads



Will the select see the inserted row?

Consistent Reads

- Aka read causality
- There is causal dependency between operations on two database connections
  - Application is expecting to see the values of earlier write

Consistent Reads

- Use: wsrep_causal_reads=ON
  - ➔ Every read (select, show) will wait until slave queue has been fully applied
- There is timeout for max causal read wait:
  - replicator.causal_read_keepalive

UDP Multicast

- Configure with gmcast.mcast_addr
- Full cluster must be configured for multicast or tcp sockets
- Multicast is good for scalability
- Best practice is to go for multicast if planning for large clusters

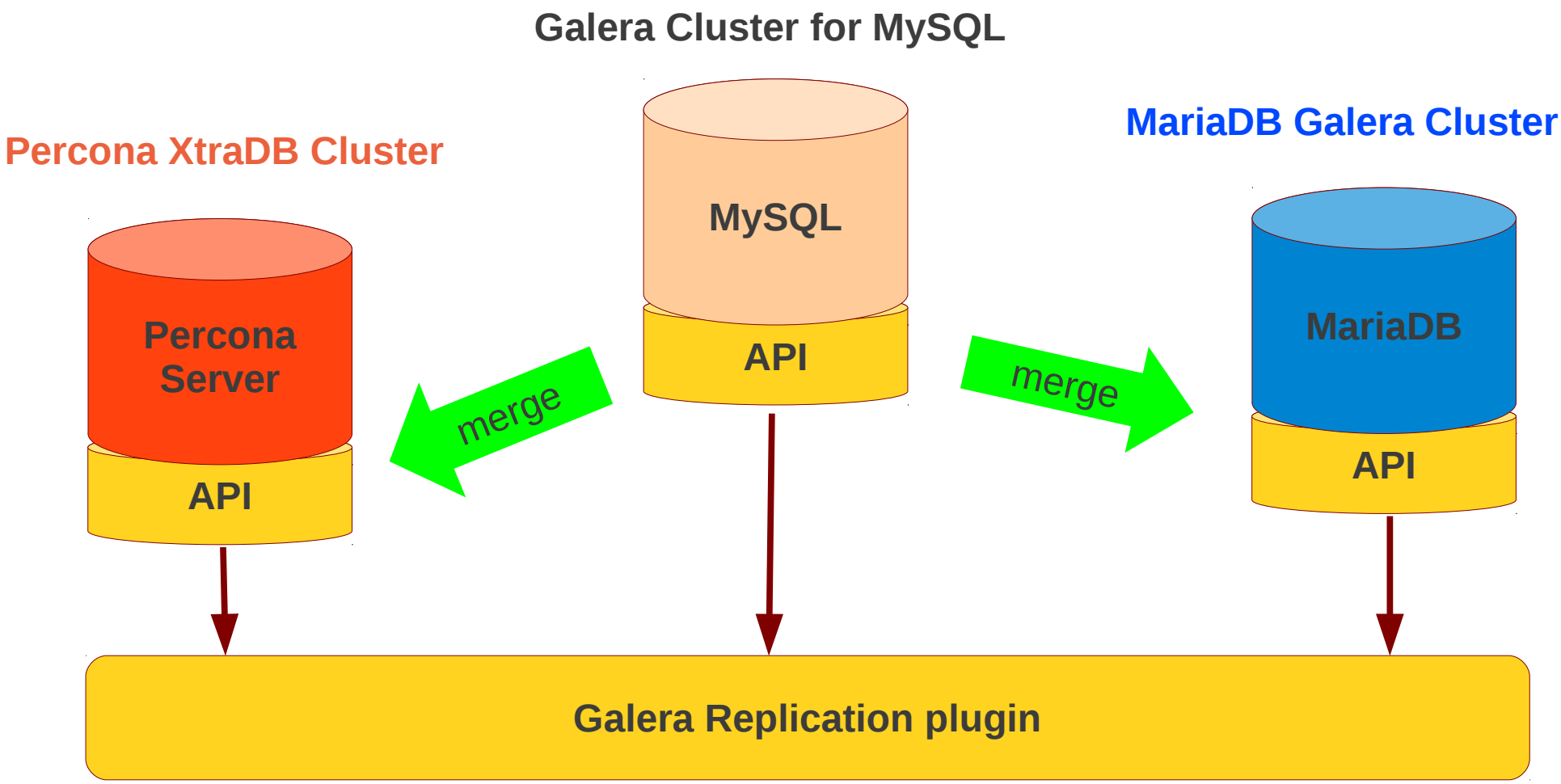
Galera Project

codership

Galera Project

- Galera Cluster for MySQL
  - 5 years development
  - based on MySQL server community edition
  - Fully open source
  - Active community
- ~3 releases per year
  - Release 2.2 RC out yesterday
  - Major release 3.0 in the works
- Galera Replication also used in:
  - Percona XtraDB Cluster
  - MariaDB Galera Cluster

Galera Project



Codership

- Consulting and support services for Galera
- Technology and support partners:
  - Percona
  - SkySQL
  - Monty Program
  - SeveralNines
  - FromDual
  - Capside

Questions?

Thank you for listening!
Happy Clustering :-)