Synchronous Replication for MySQL

Kenny Gryp
<kenny.gryp@percona.com>
29 Oct 2013

Agenda

- Default asynchronous MySQL replication
- Percona XtraDB Cluster:
  - Introduction / Features / Load Balancing
- Use Cases:
  - High Availability / WAN Replication / Read Scaling
- Limitations
- Future

Percona

- Percona is the oldest and largest independent **MySQL** Support, Consulting, Remote DBA, Training, and Software Development company with a global, 24x7 staff of over 100 serving more than 2,000 customers in 50+ countries *since 2006* !
- Our contributions to the MySQL community include:
  - **Percona Server, Percona XtraDB Cluster**
  - **Percona XtraBackup: online backup**
  - **Percona Toolkit, Percona Playback…**
  - books, and research published on the MySQL Performance Blog.

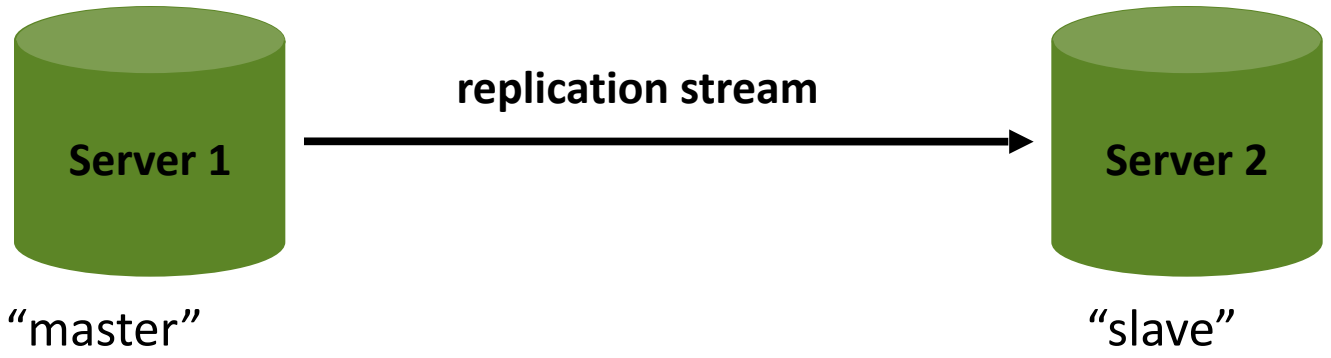
Agenda

- **Default asynchronous MySQL Replication**
- Percona XtraDB Cluster:
  - Introduction / Features / Load Balancing
- Use Cases:
  - High Availability / WAN Replication / Read Scaling
- Limitations
- Future

MySQL Replication

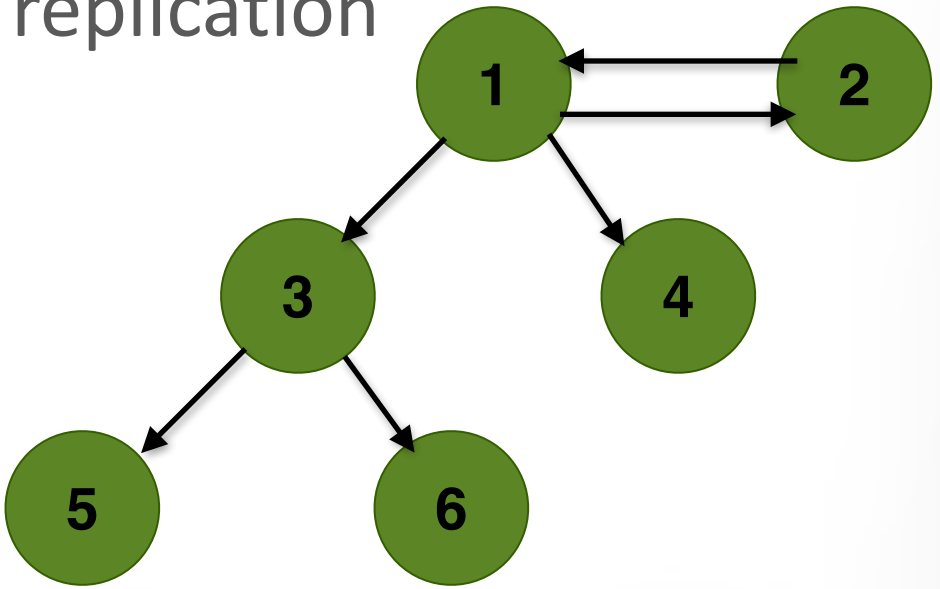
If your HA is based on MySQL Replication -
You may be playing a dangerous game !

Traditional Replication Approach



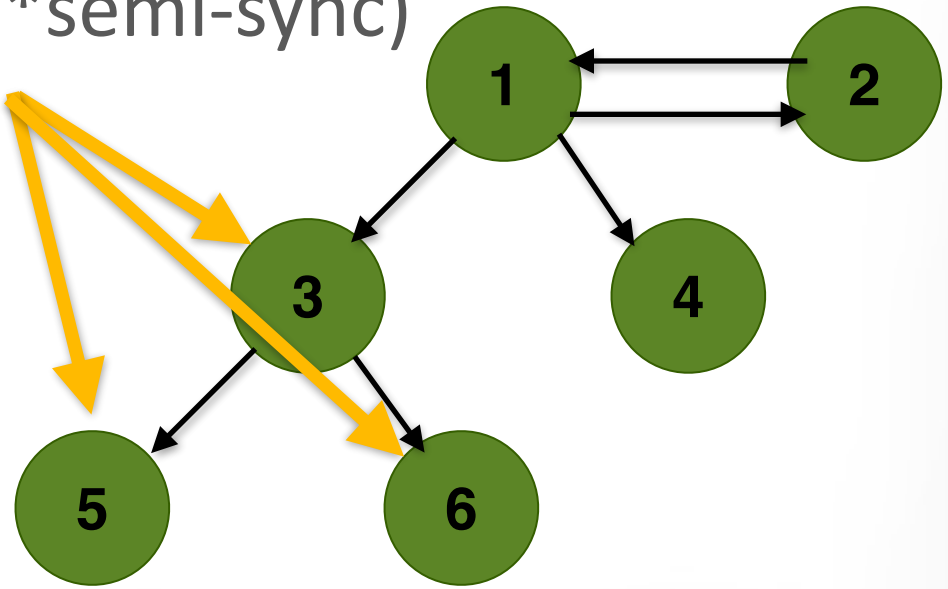
MySQL Replication

- Common Topologies:
  - Master-Master (Only 1 active master)
  - 1 or more layers of replication



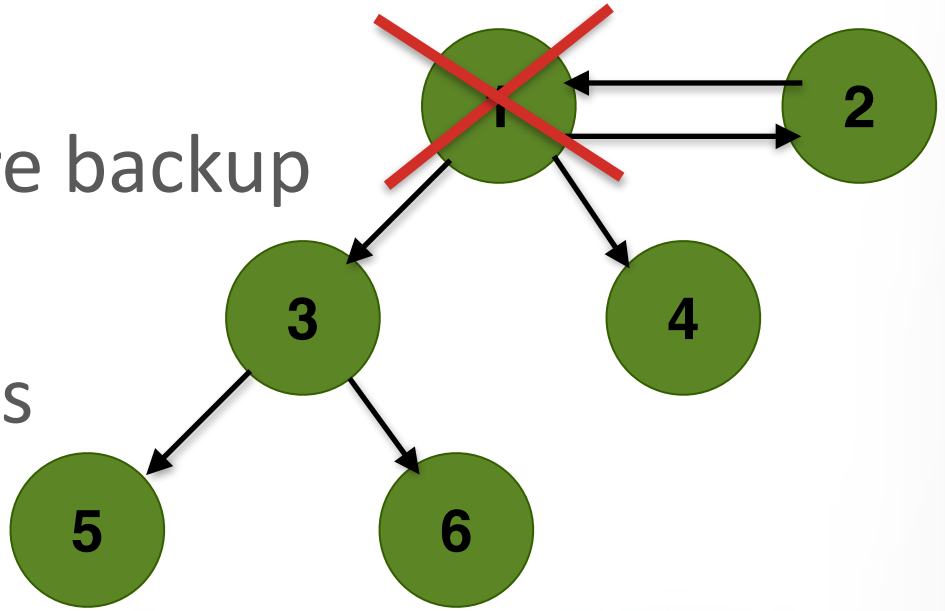
MySQL Replication

- Slaves can be used for reads:
  - asynchronous, stale data is the rule
  - data loss possible (*semi-sync)



MySQL Replication

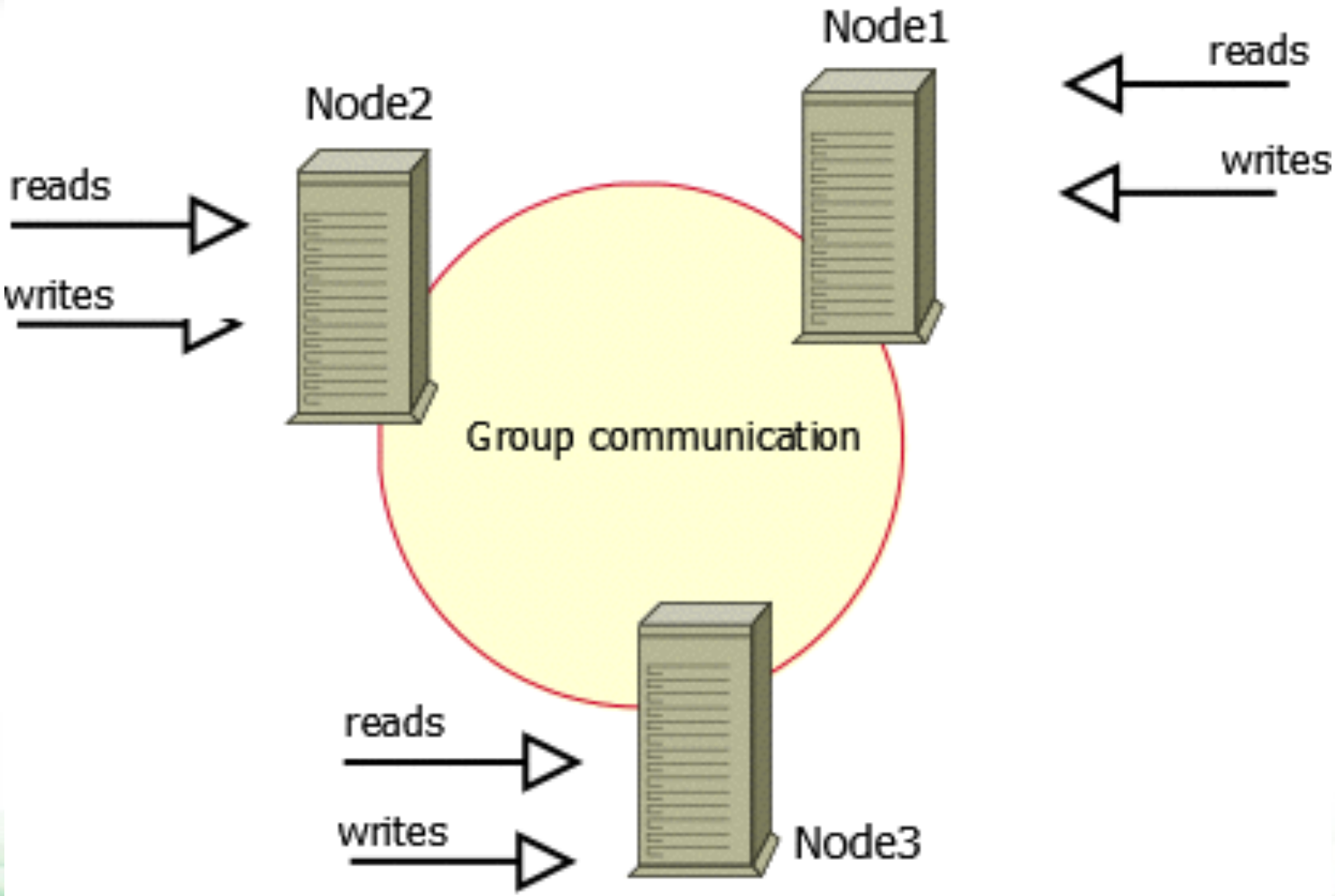
- non-trivial:
  - external monitoring
  - scripts for failover
  - add node == restore backup
  - much better in
    MySQL 5.6: GTIDs



Agenda

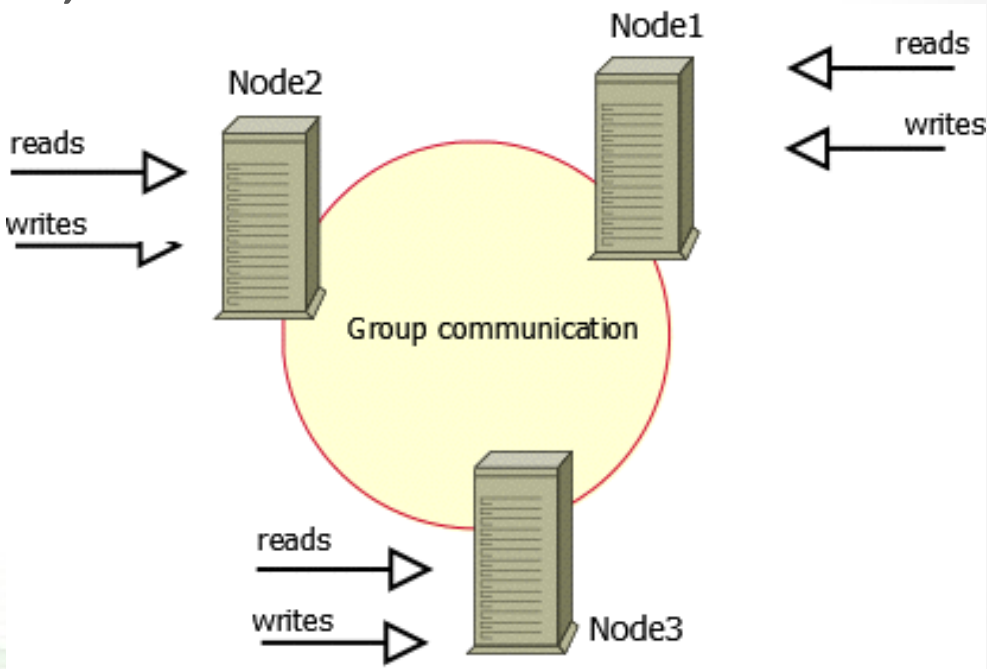
- Default asynchronous MySQL Replication
- **Percona XtraDB Cluster:**
  - **Introduction / Features / Load Balancing**
- Use Cases:
  - High Availability / WAN Replication / Read Scaling
- Limitations
- Future

Percona XtraDB Cluster



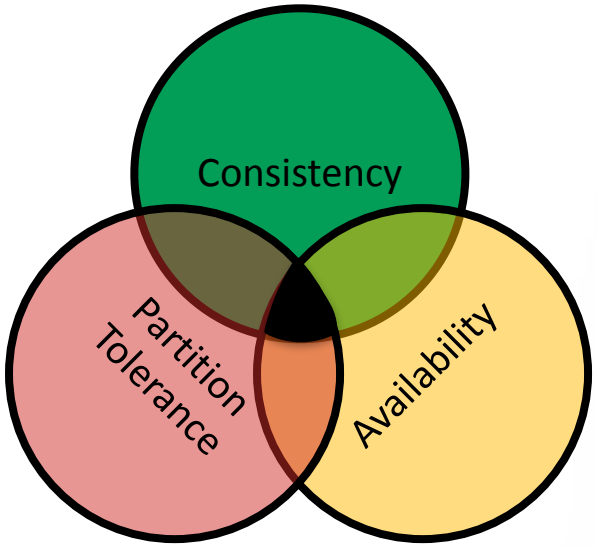
Percona XtraDB Cluster

- All nodes have a full copy of the data
- Every node is equal
- No central management, no SPOF



CAP Theorem

- MySQL (Asynchronous) Replication:
  - Availability
  - Partition Tolerance
- Percona XtraDB Cluster
  - Consistency
  - Availability



What is Percona XtraDB Cluster ?

- Percona Server
- + WSREP patches
- + Galera library
- + Utilities (init, SST and cluster check scripts)

Percona Server

- This is a ***free open source*** solution, Percona Server is a **MySQL alternative** which offers breakthrough **performance**, **scalability**, **features**, and **instrumentation**. Self-tuning algorithms and support for extremely high-performance hardware make it the clear choice for organisations that demand excellent performance and reliability from their MySQL database server.

WSREP and Galera

- **WSREP API** is a project to develop generic replication plugin interface for databases (**W**rite**S**et **Rep**lication)
- **Galera** is a wsrep provider that implements multi-master, synchronous replication

What is Percona XtraDB Cluster ?

What is Percona XtraDB Cluster ?

What is Percona XtraDB Cluster ?

Minimal efforts to return back to MySQL

Features

- Synchronous Replication
- Multi Master
- Parallel Applying
- Quorum Based
- Certification/Optimistic Locking
- Automatic Node Provisioning

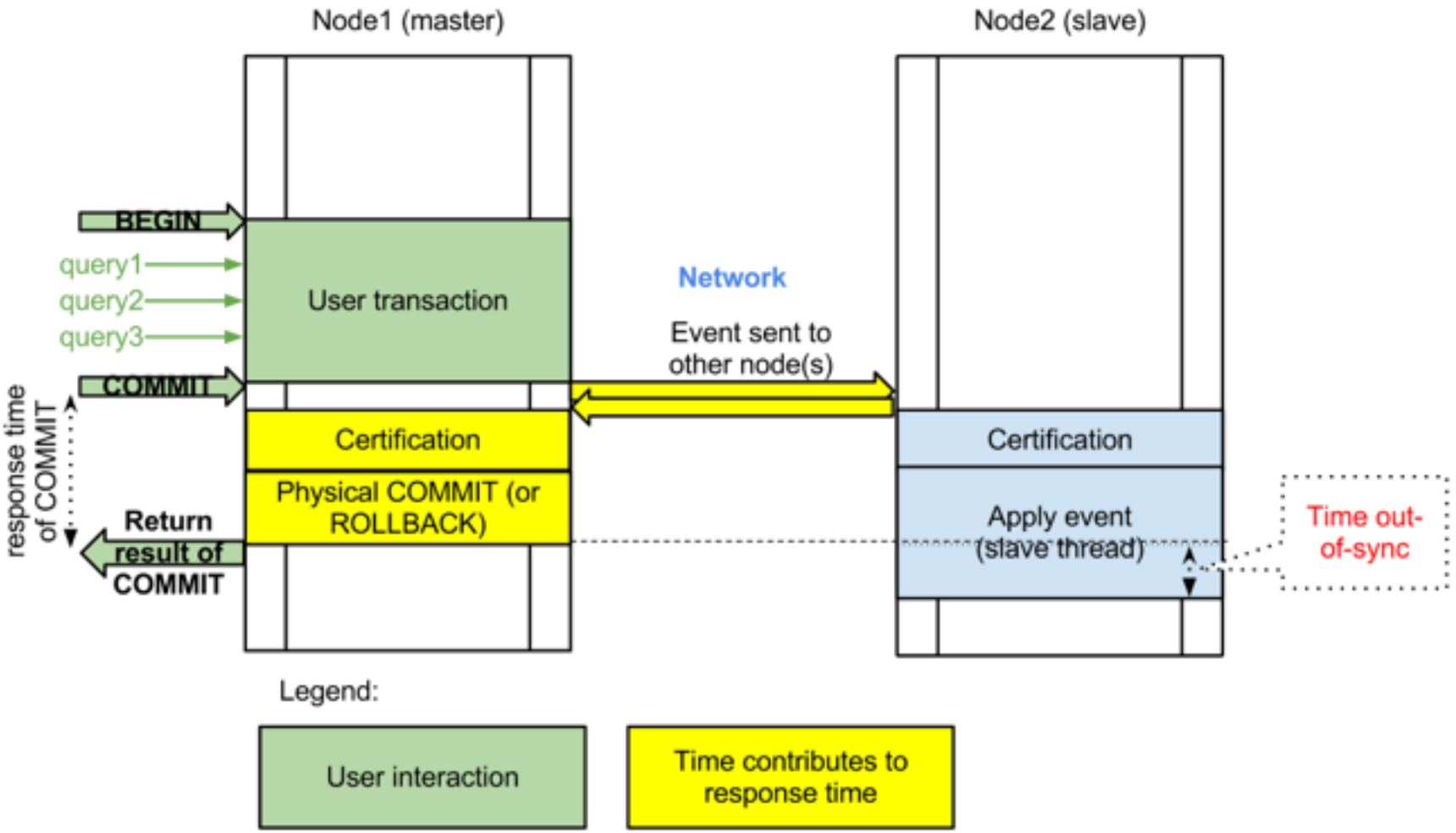
Features

- **Synchronous Replication**
- Multi Master
- Parallel Applying
- Quorum Based
- Certification/Optimistic Locking
- Automatic Node Provisioning

(Virtual) Synchronous Replication

- Writesets (transactions) are replicated to all available nodes **on commit** (and queued on each)
- Writesets are **individually "certified"** on every node, **deterministically**.Either it is committed on all nodes or no node at all (NO 2PC)
- Queued writesets are applied on those nodes independently and asynchronously
- **Flow Control** avoids too much 'lag'

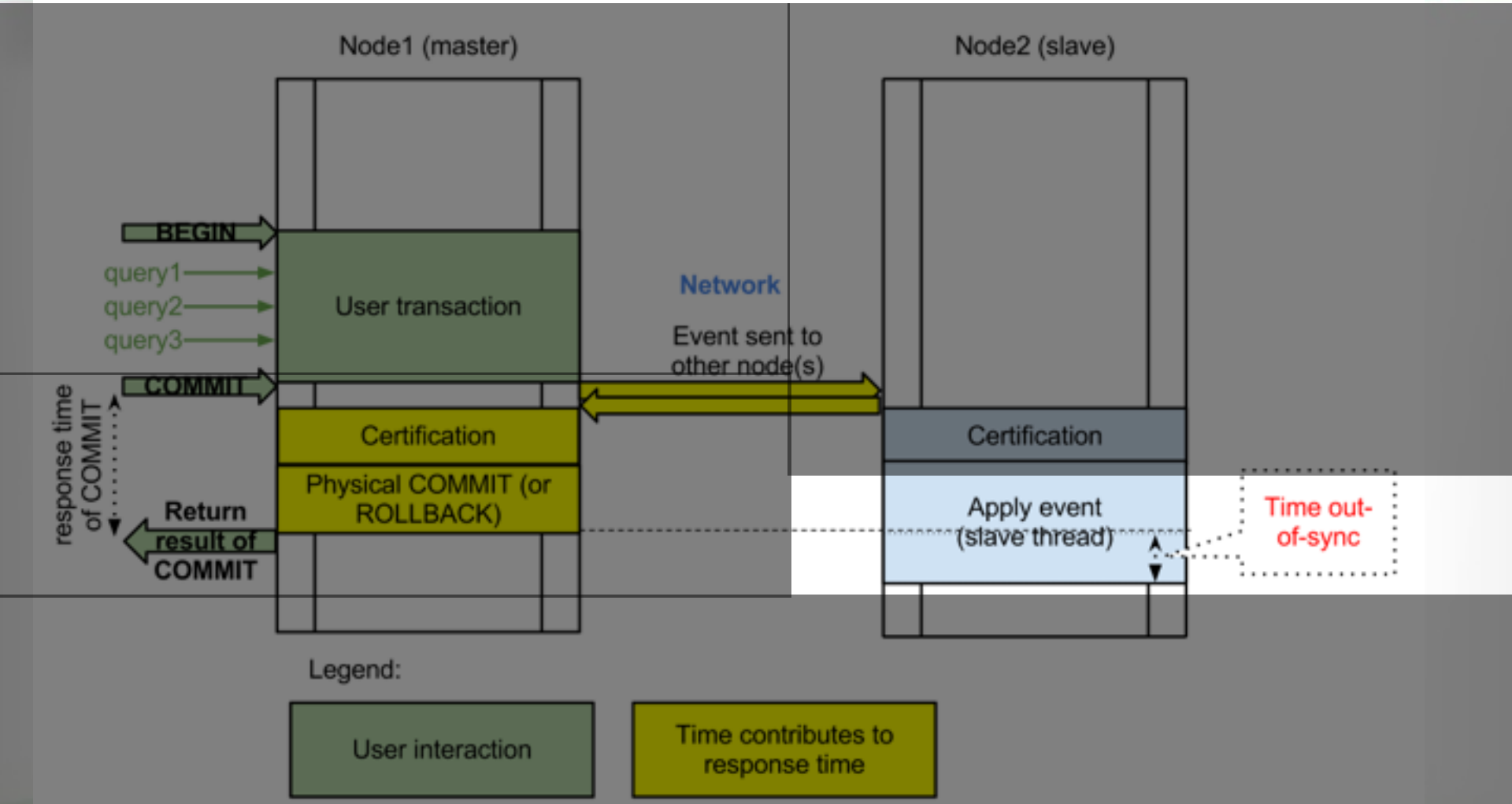
(Virtual) Synchronous Replication



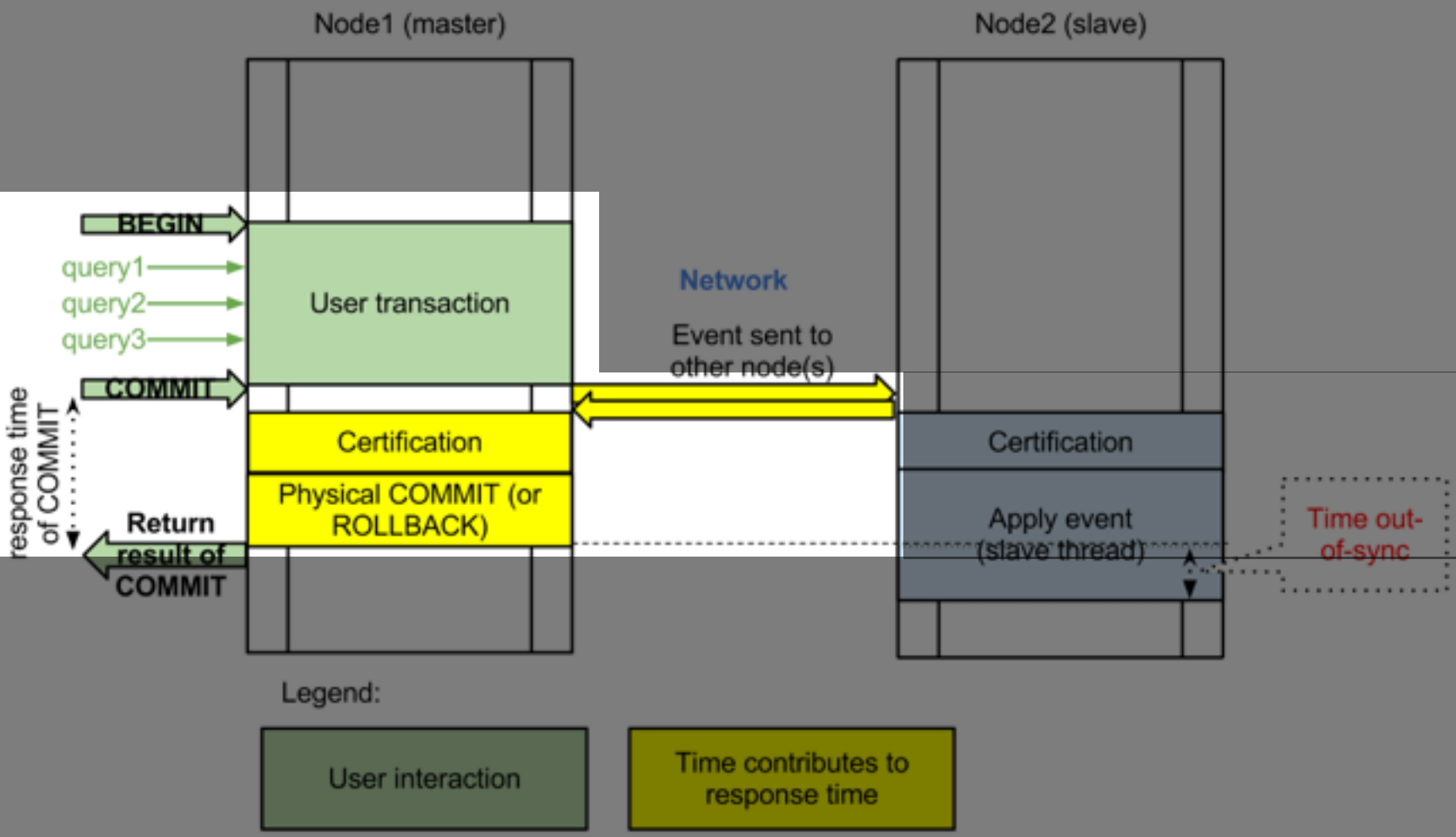
(Virtual) Synchronous Replication

- Reads can read old data
  - Flow Control (by default 16 trx) avoids lag
  - *wsrep_causal_reads* can be enabled to ensure full synchronous reads
- Latency: writes are fast, only at COMMIT, communication with other nodes happen

Stale Reads


Node1 (master)
Node2 (slave)
BEGIN
query1
query2
query3
User transaction
COMMIT
Network
Event sent to other node(s)
response time of COMMIT
Certification
Physical COMMIT (or ROLLBACK)
Return result of COMMIT
Certification
Apply event (slave thread)
Time out-of-sync
Legend:
User interaction
Time contributes to response time

Latency

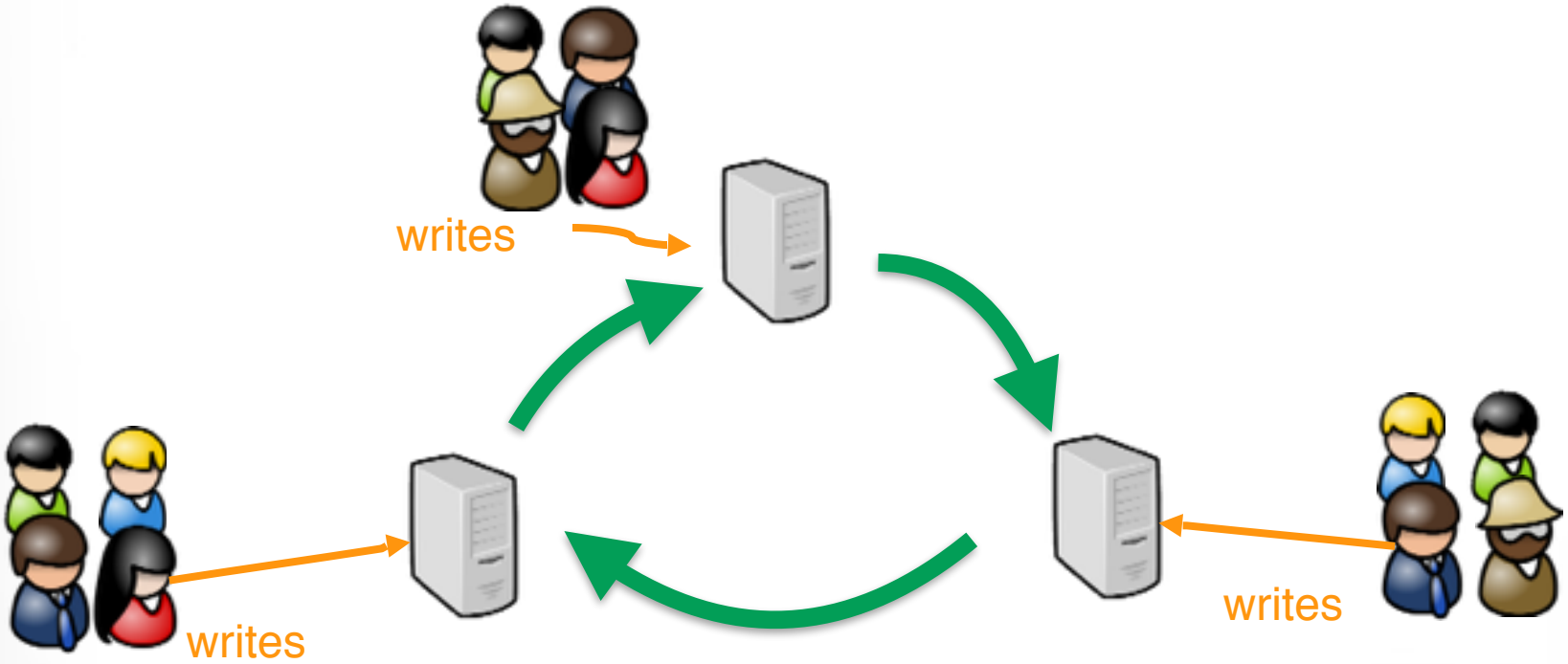

Node1 (master)
Node2 (slave)
BEGIN
query1
query2
query3
User transaction
COMMIT
Network
Event sent to other node(s)
Certification
Physical COMMIT (or ROLLBACK)
Return result of COMMIT
response time of COMMIT
Certification
Apply event (slave thread)
Time out-of-sync
Legend:
User interaction
Time contributes to response time

Features

- Synchronous Replication
- **Multi Master**
- Parallel Applying
- Quorum Based
- Certification/Optimistic Locking
- Automatic Node Provisioning

Multi-Master Replication

- You can write to any node in your cluster*
- Writes are ordered inside the cluster

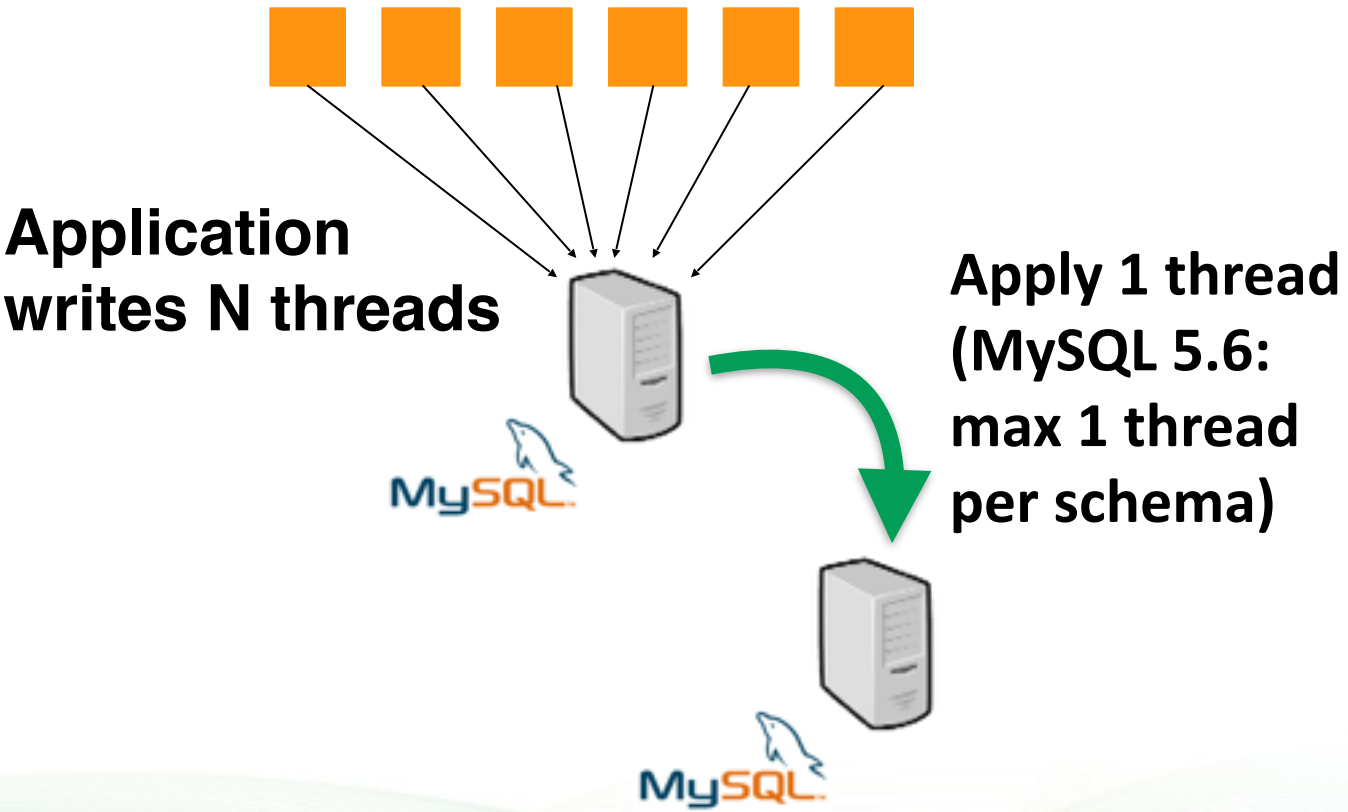


Features

- Synchronous Replication
- Multi Master
- **Parallel Applying**
- Quorum Based
- Certification/Optimistic Locking
- Automatic Node Provisioning

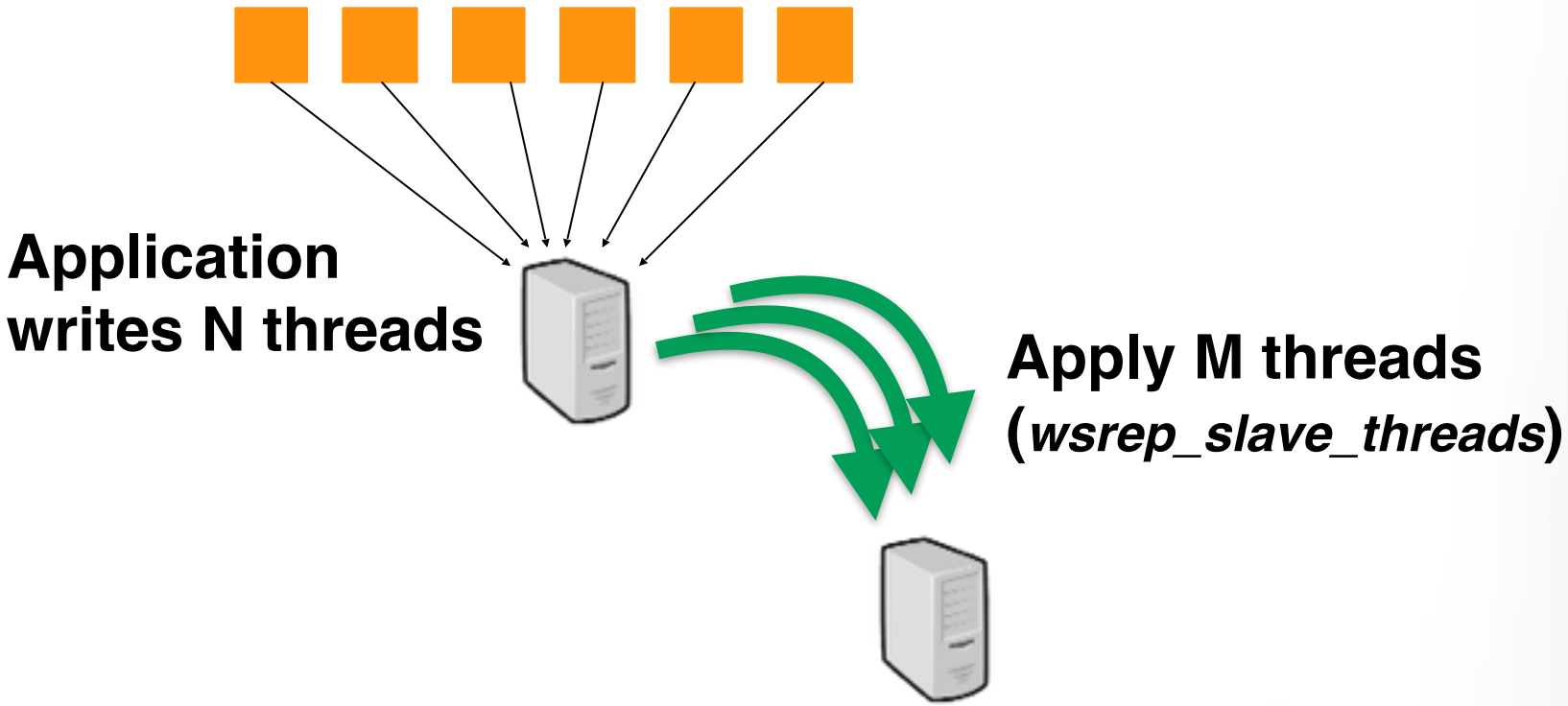
Parallel Replication

- Standard MySQL



Parallel Replication

- PXC / Galera



Features

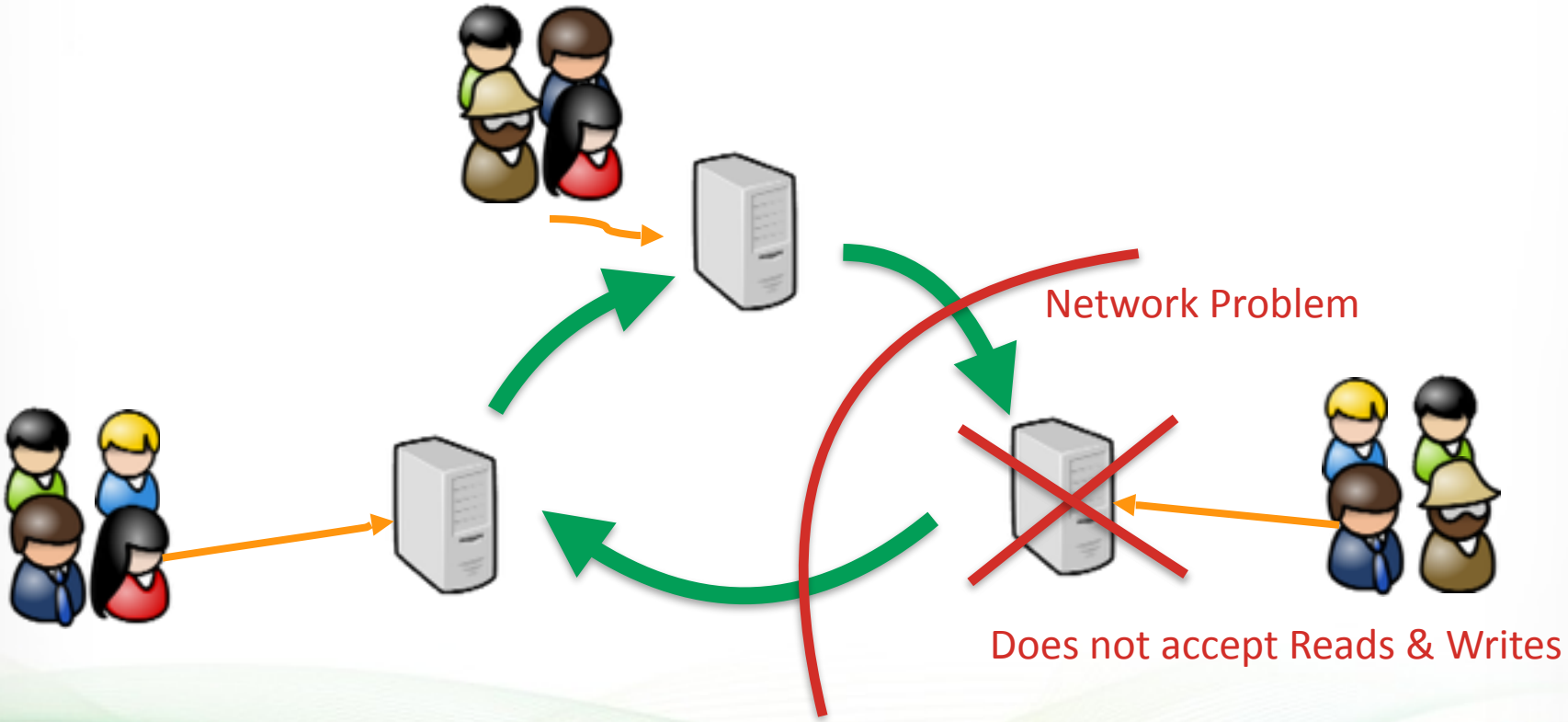
- Synchronous Replication
- Multi Master
- Parallel Applying
- **Quorum Based**
- Certification/Optimistic Locking
- Automatic Node Provisioning

Quorum Based

- If a node does not see more than 50% of the total amount of nodes: reads/writes are not accepted.
- Split brain is prevented
- This requires at least 3 nodes to be effective
- a node can be an arbitrator (**garbd**), joining the communication, but not having any MySQL running
- Can be disabled (but be warned!)

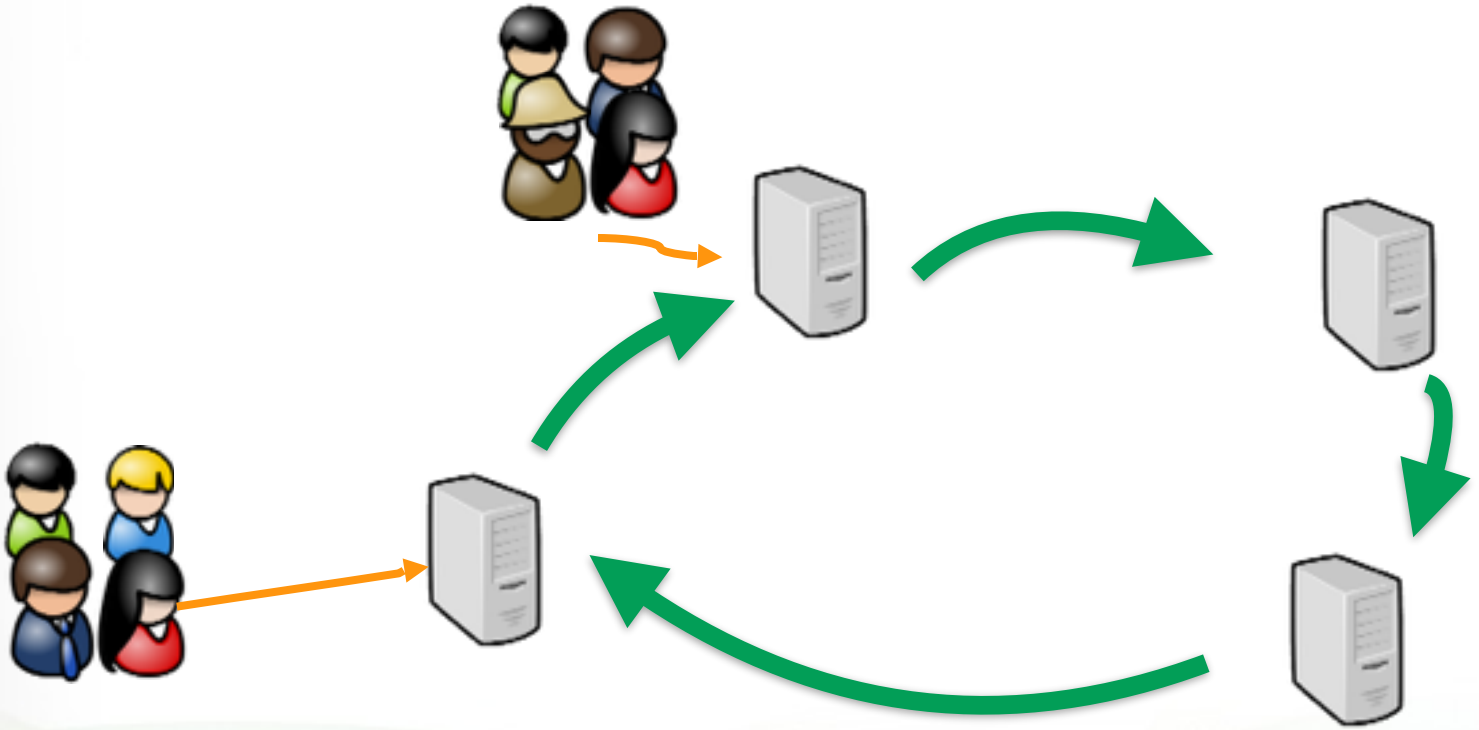
Quorum Based

- Loss of connectivity



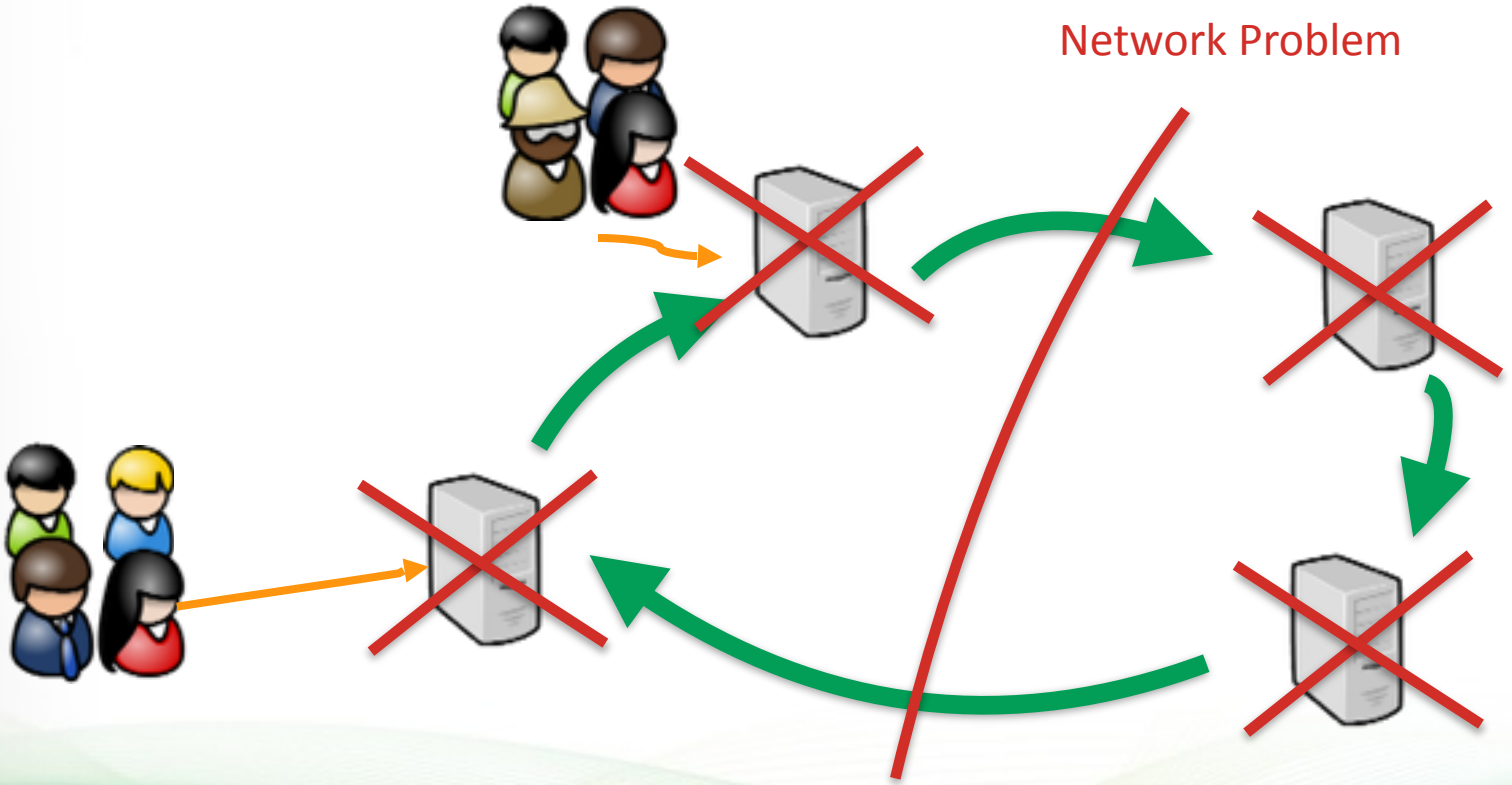
Quorum Based

- 4 Nodes



Quorum Based

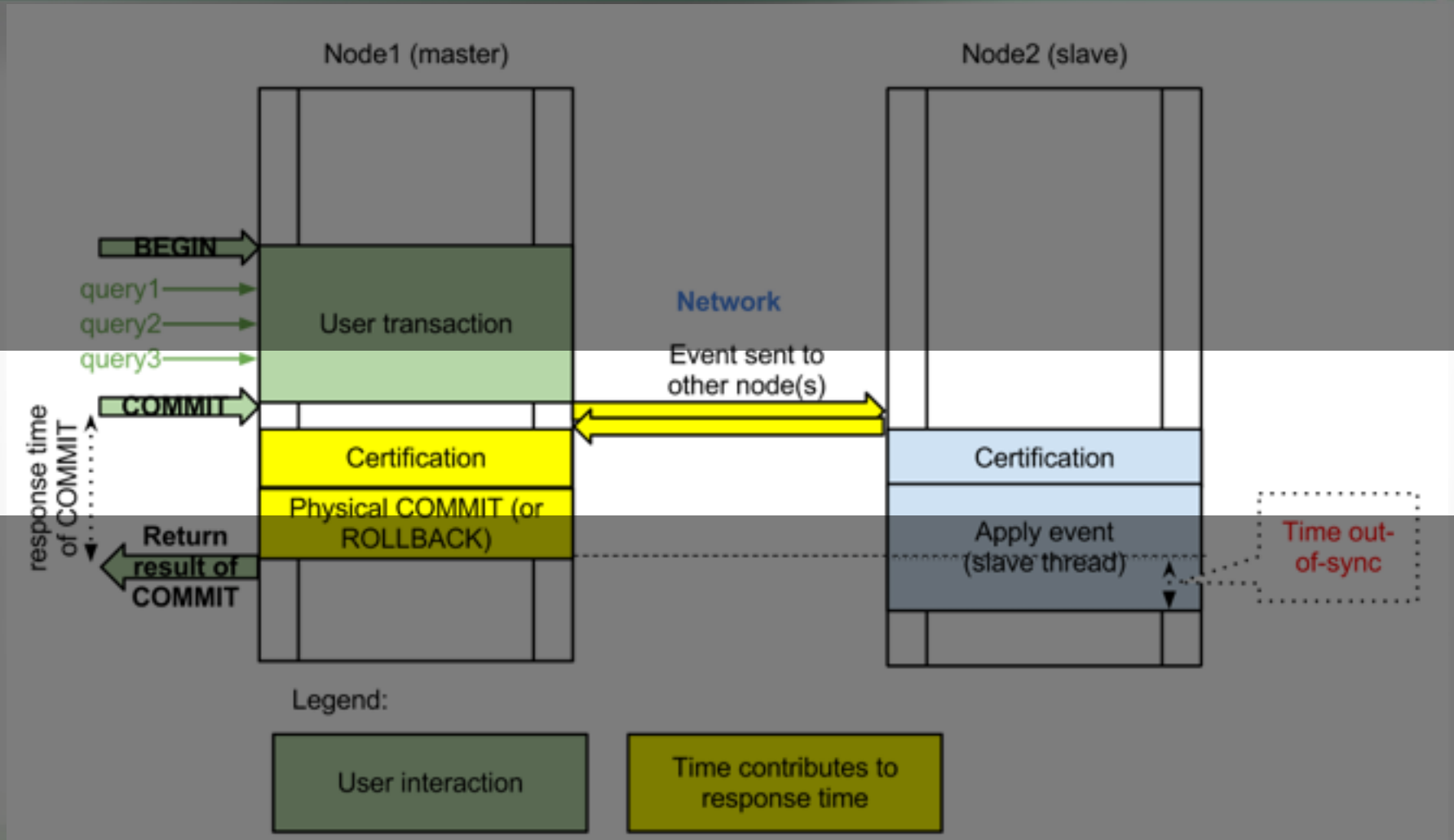
- Default quorum configuration:
  4 Nodes, 0 Nodes have quorum



Features

- Synchronous Replication
- Multi Master
- Parallel Applying
- Quorum Based
- **Certification/Optimistic Locking**
- Automatic Node Provisioning

Certification



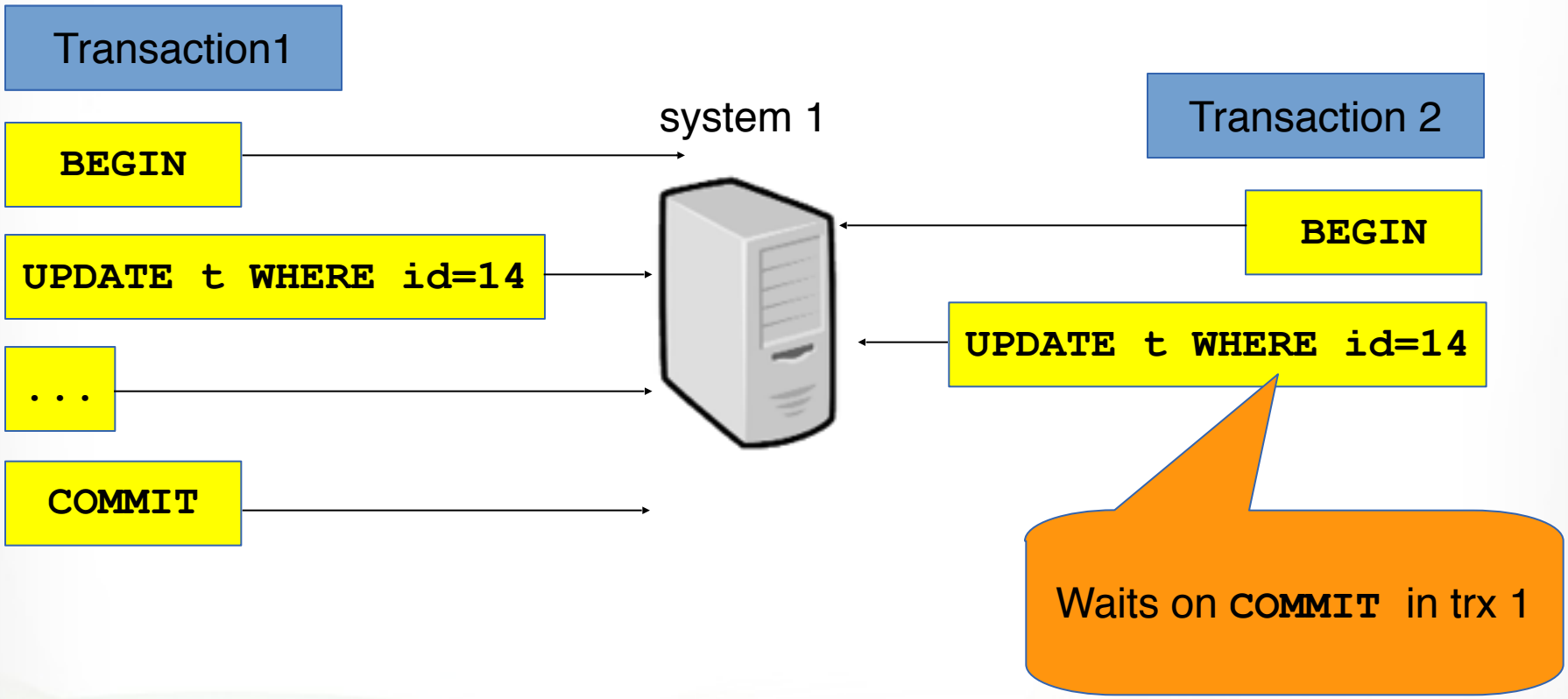
Optimistic Locking

- Communication to the other nodes of the cluster only happens during COMMIT, this affects locking behavior.
- Optimistic Locking is done:
  - InnoDB Locking happens local to the node
  - During COMMIT/Certification, the other nodes bring deadlocks

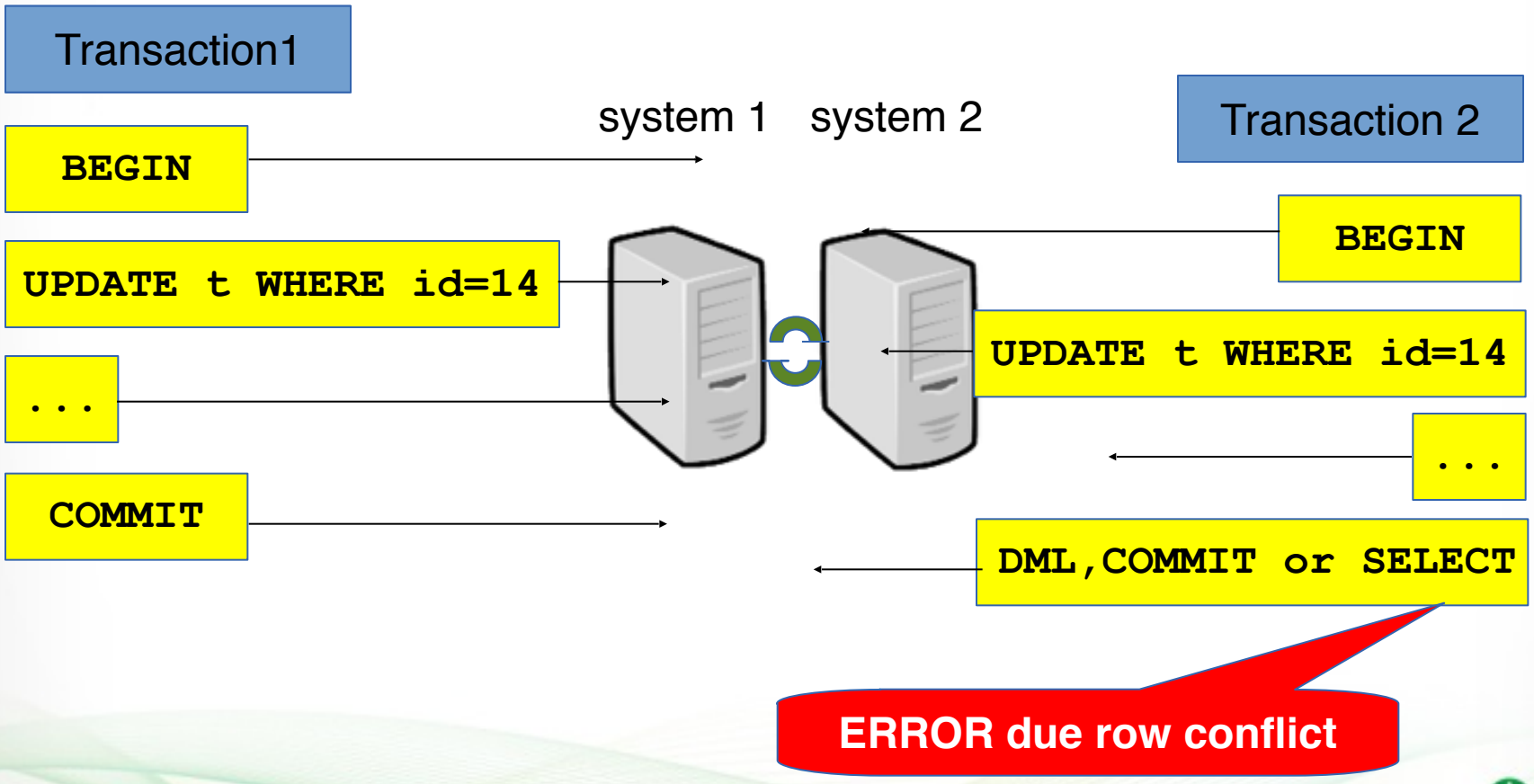
Optimistic Locking

- Some Characteristics:
  - also COMMIT and SELECT's can fail on **deadlock**
  - Might require application changes:
    Not all applications handle this properly

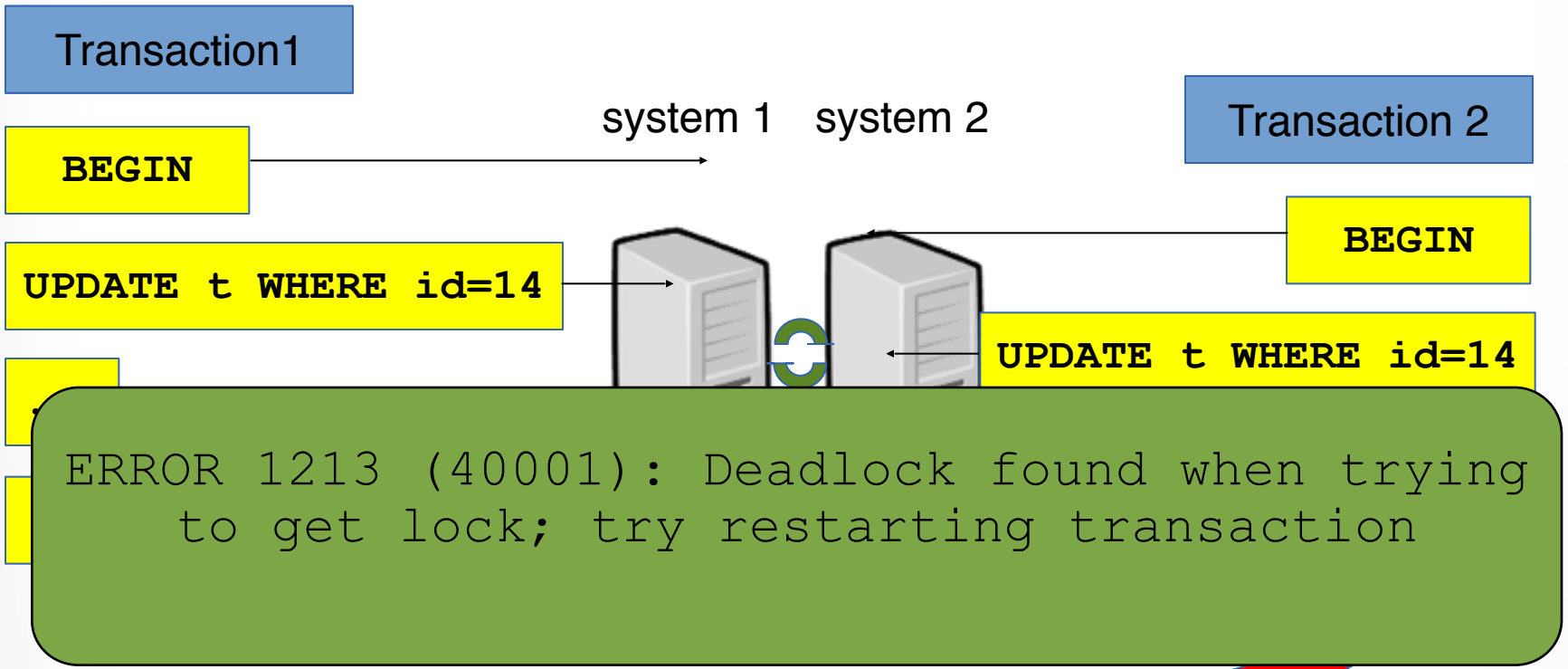
Traditional InnoDB Locking



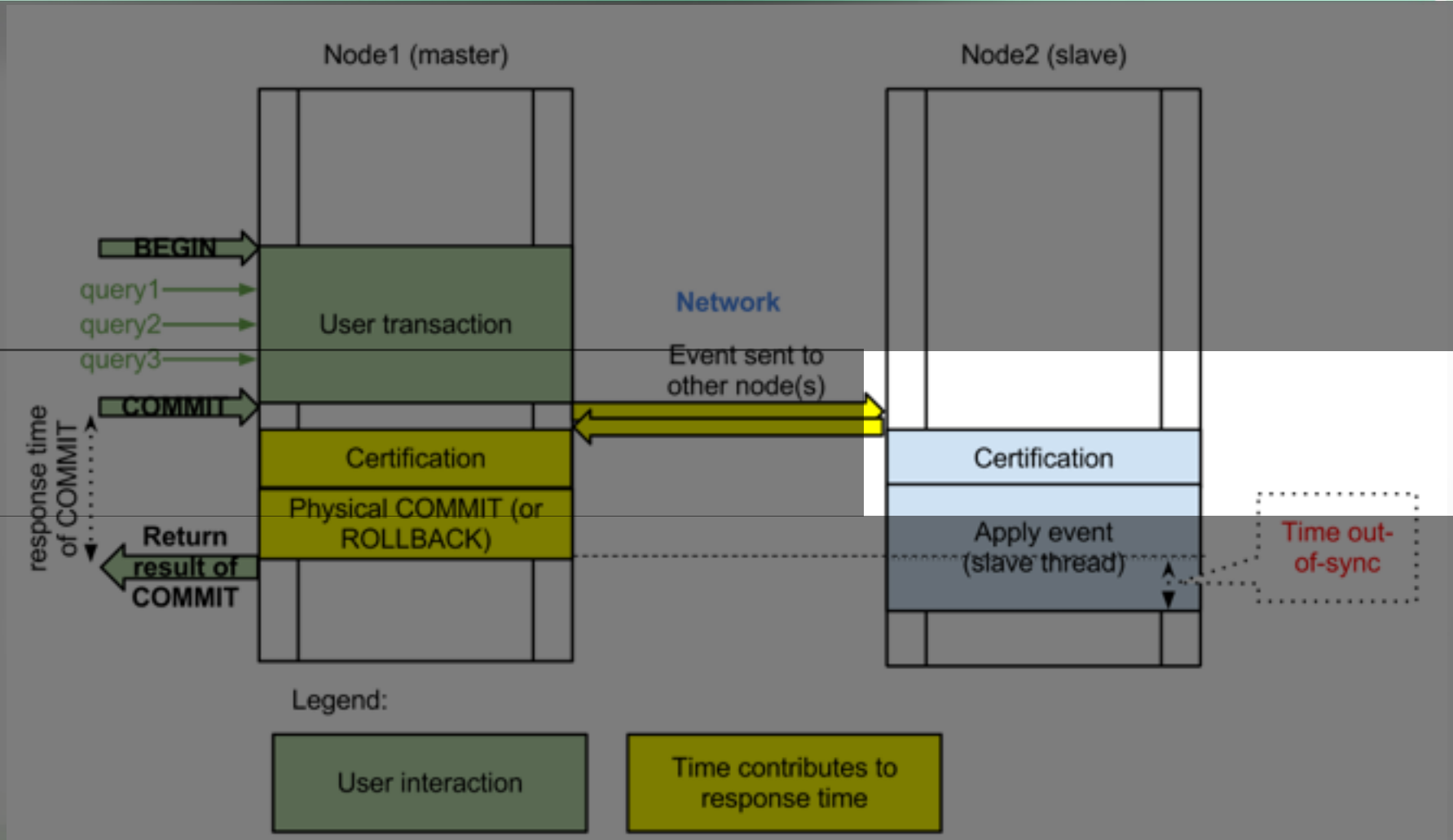
Traditional InnoDB Locking



Traditional InnoDB Locking



Optimistic Locking



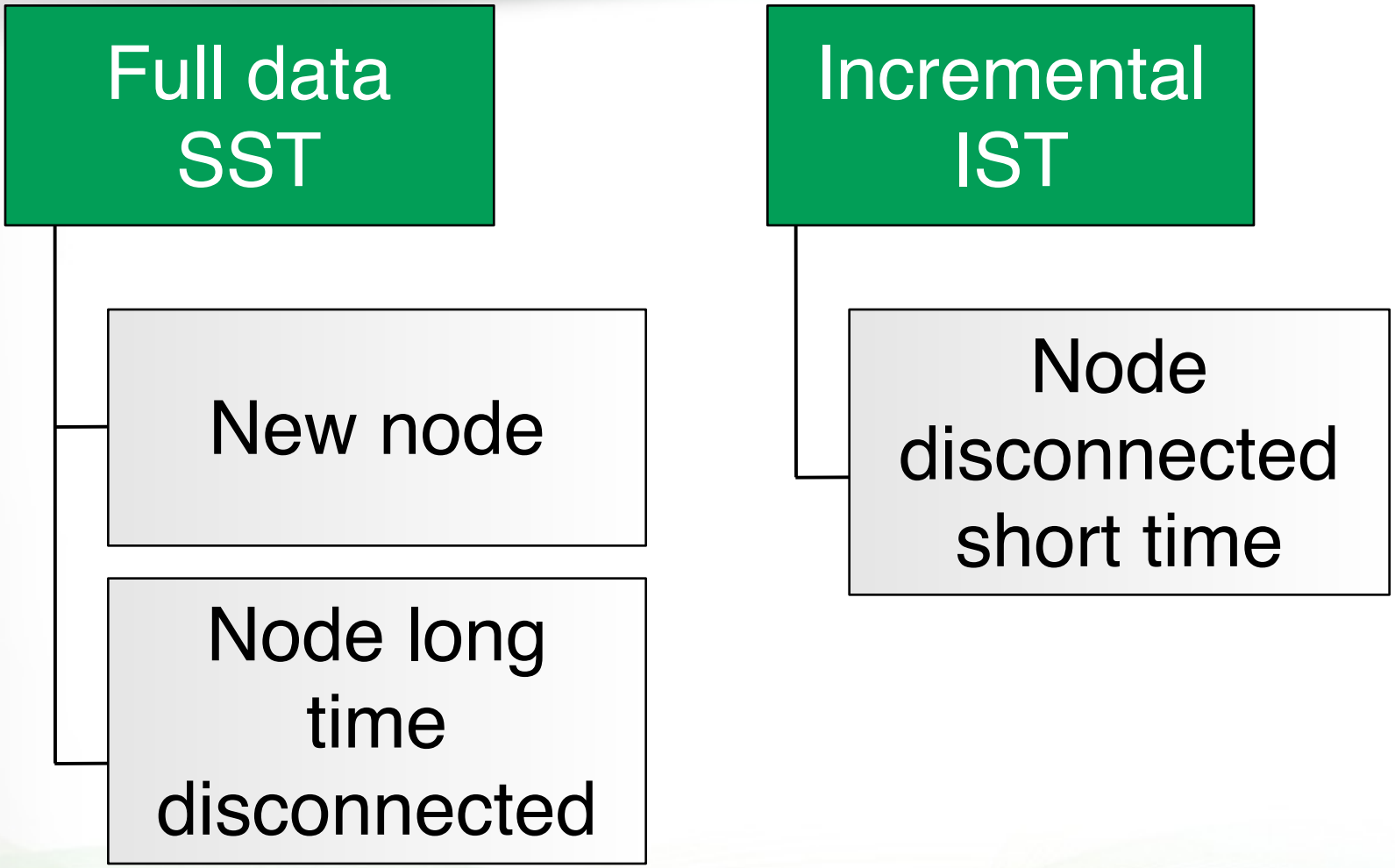
Features

- Synchronous Replication
- Multi Master
- Parallel Applying
- Quorum Based
- Certification/Optimistic Locking
- **Automatic Node Provisioning**

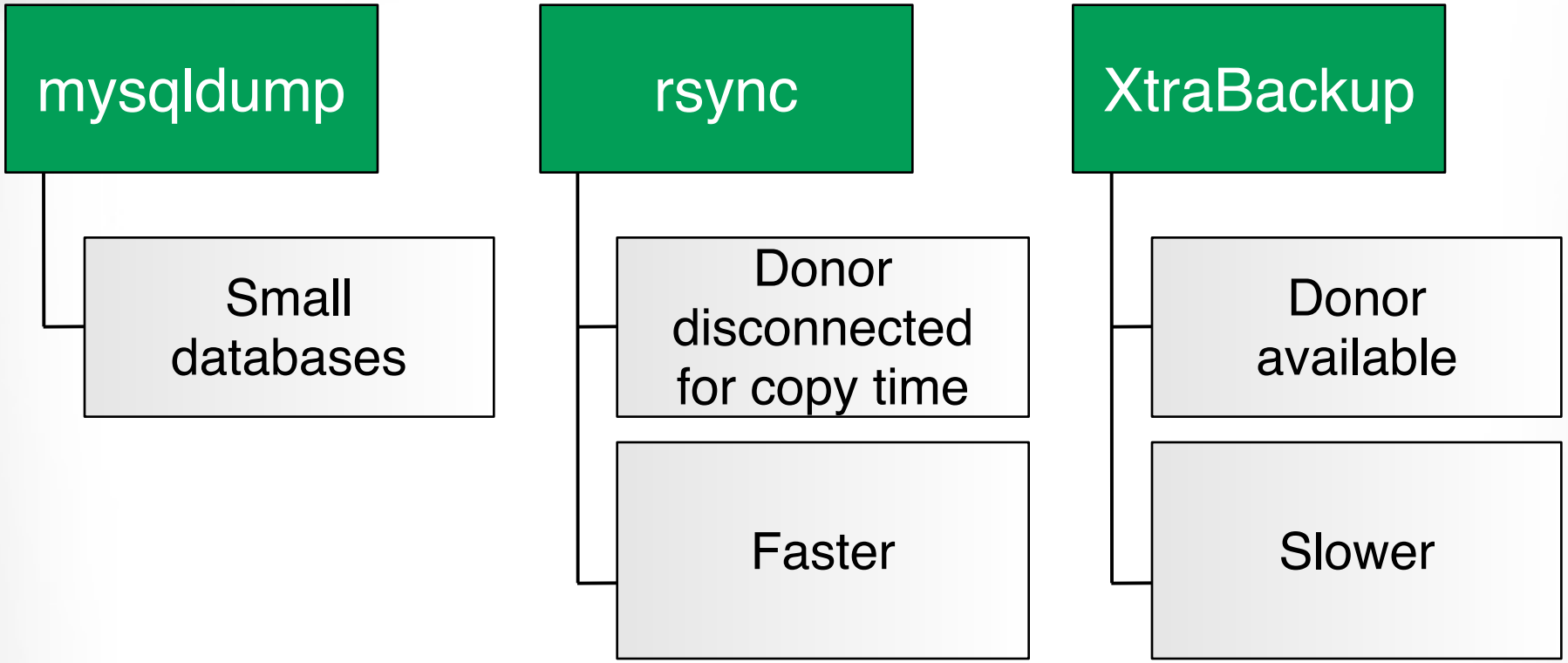
Automatic Node Provisioning

- When a node joins the cluster:
  - the data is automatically copied
  - when finished: the new node is automatically ready and accepting connections
- 2 different types of joining:
  - **SST** (*state snapshot transfer*): full copy of the data
  - **IST** (*incremental state transfer*): send only the missing writesets *(if available)*

StateTransfer Summary

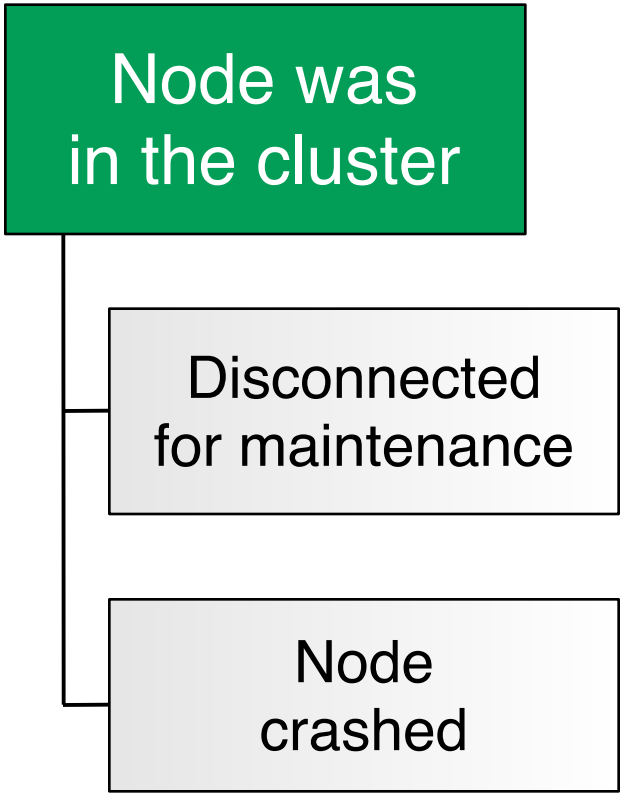


Snapshot State Transfer

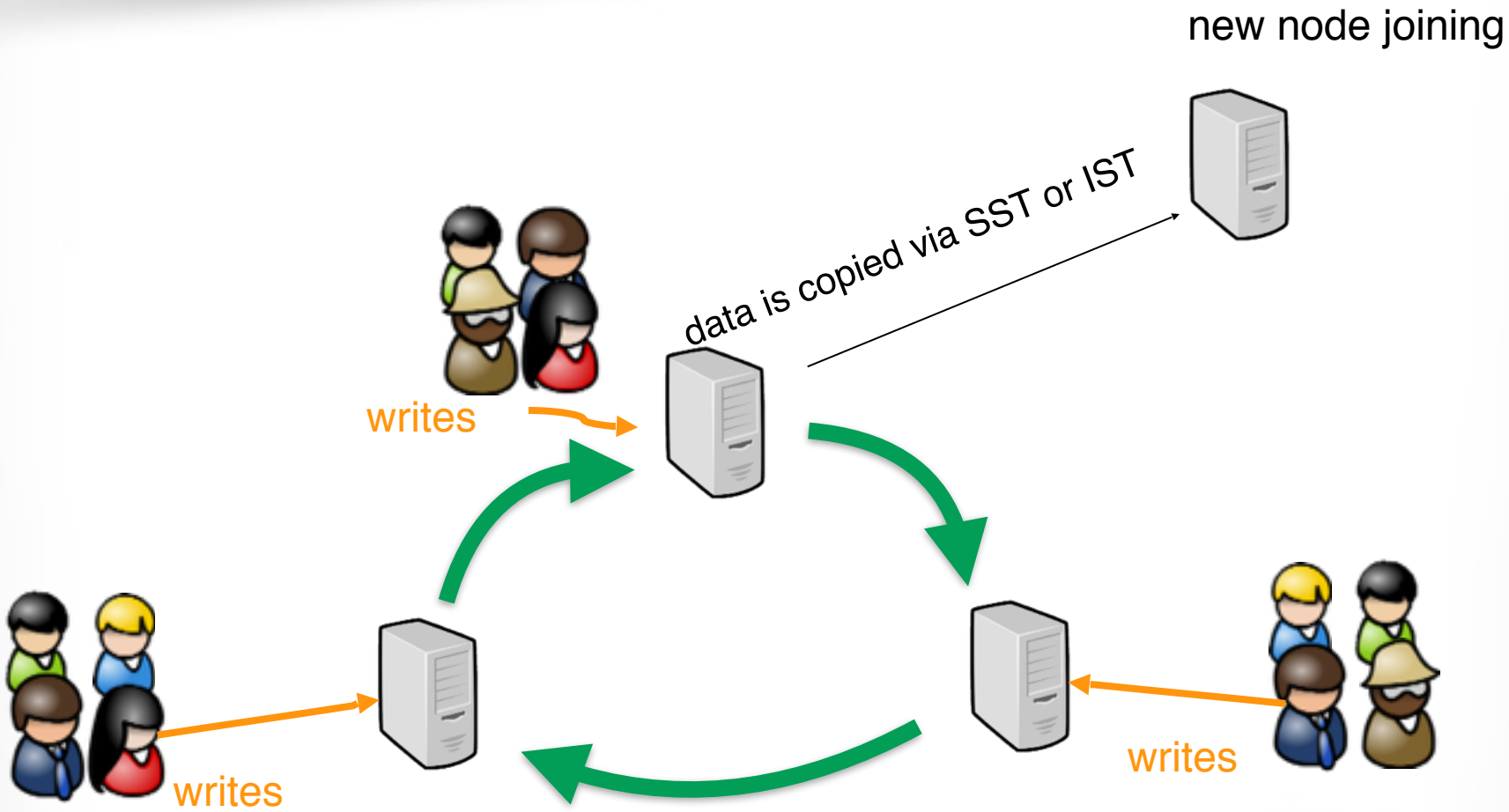


PERCONA XTRABACKUP

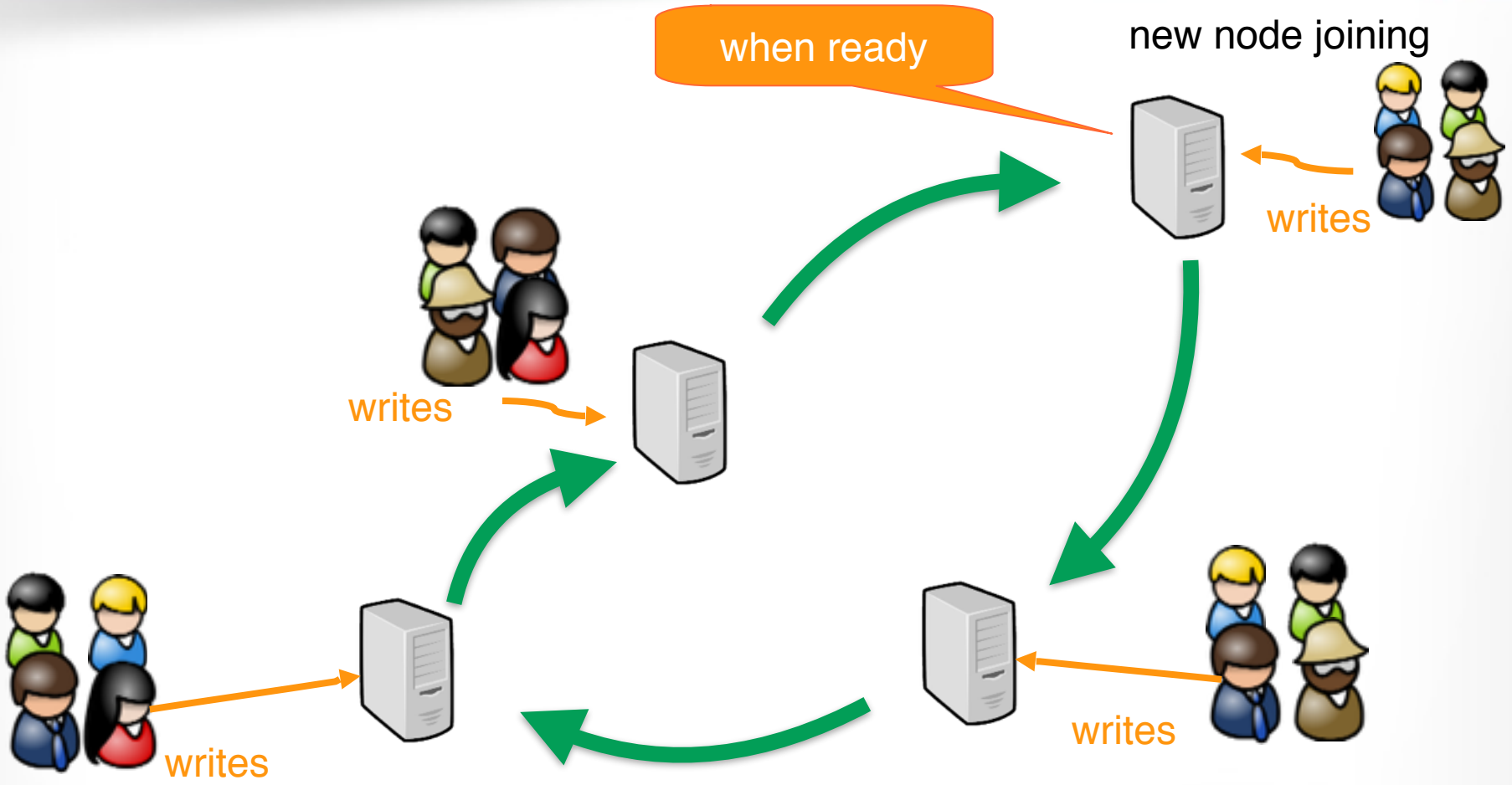
Incremental State Transfer



Automatic Node Provisioning



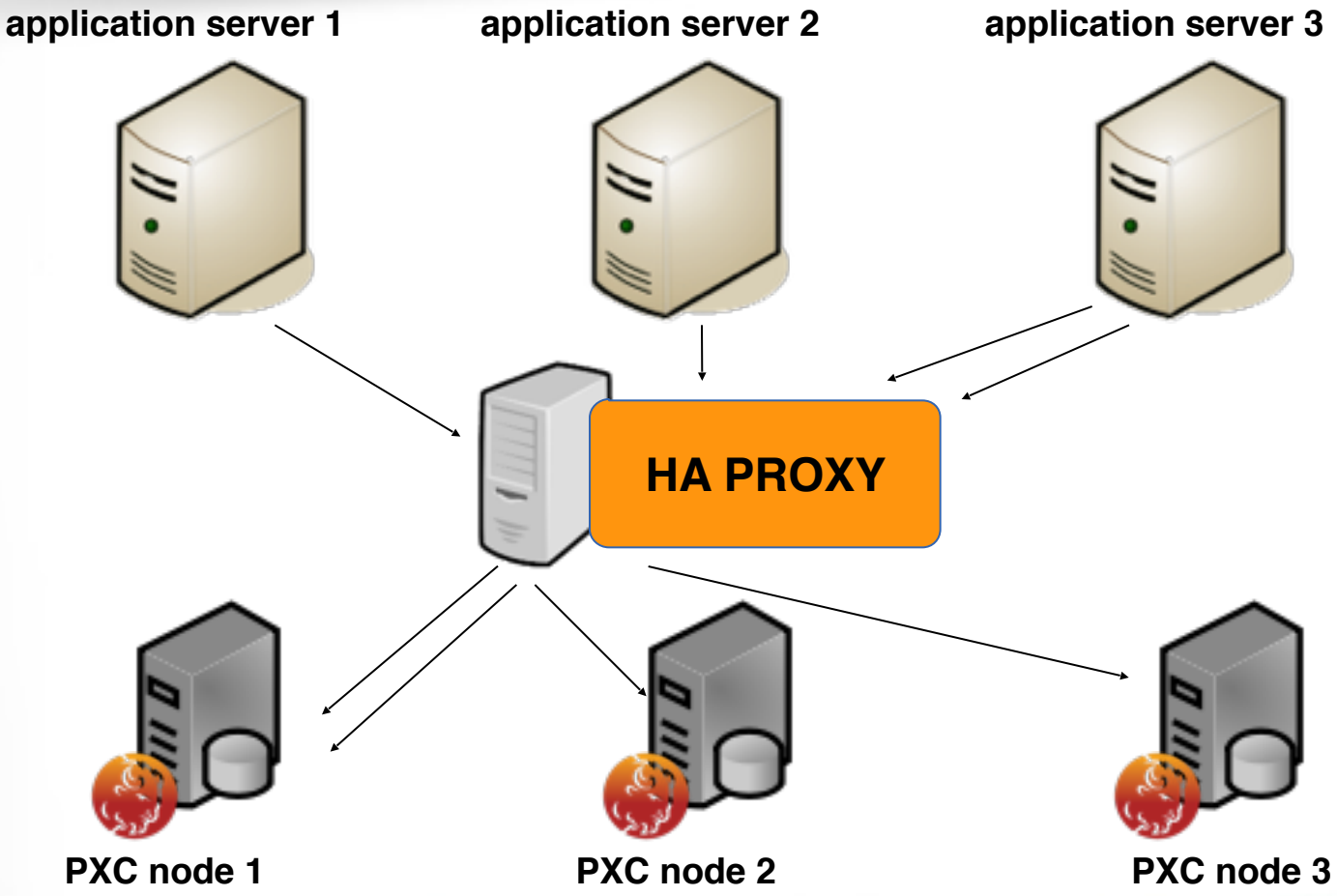
Automatic Node Provisioning



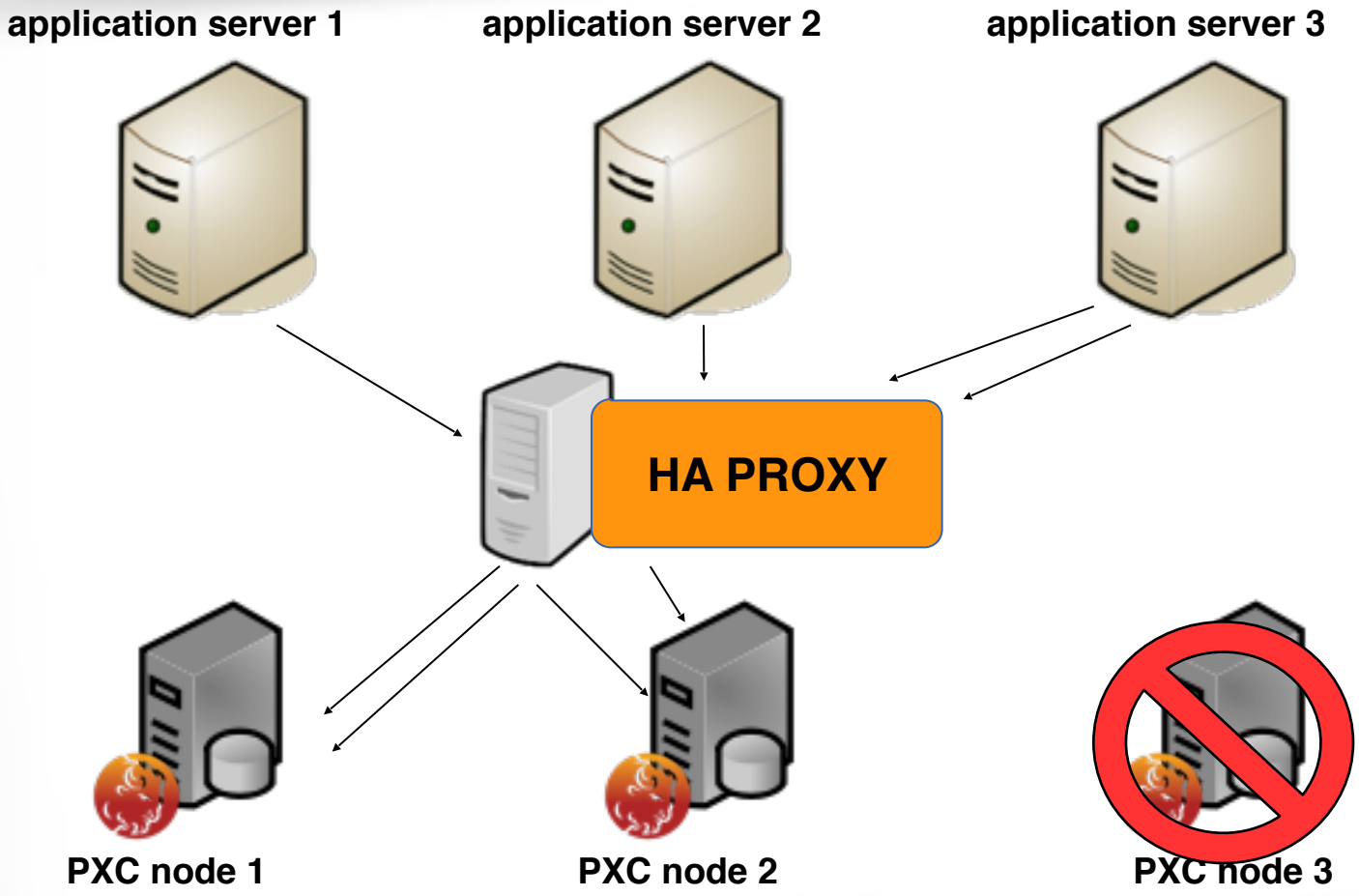
PXC with a Load balancer

- PXC is often integrated with a load balancer
  - service can be checked using *clustercheck* or *pyclustercheck*
- The load balancer can
  - be a dedicated layer
  - integrated at application layer
  - integrated at database layer

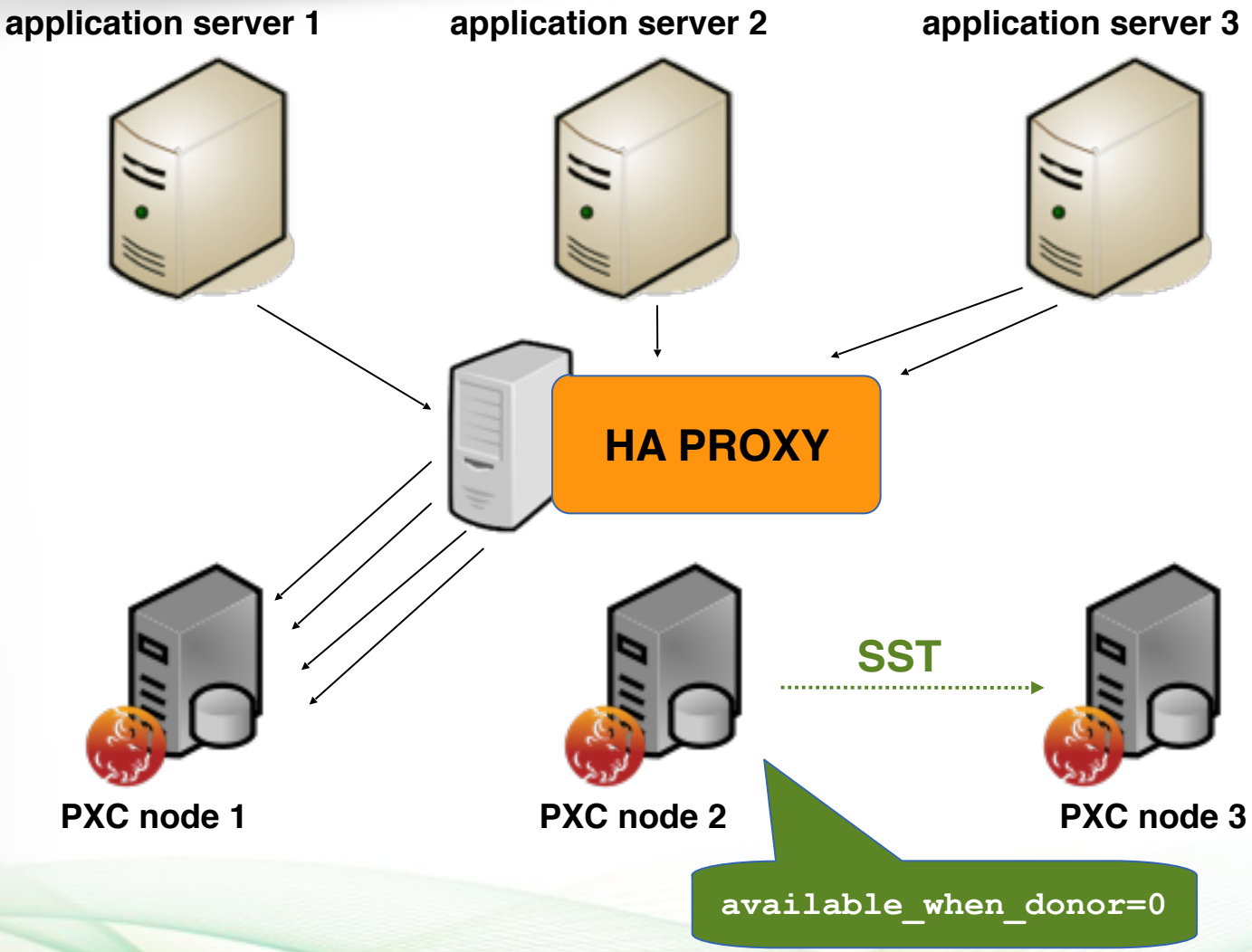
Dedicated shared HAProxy



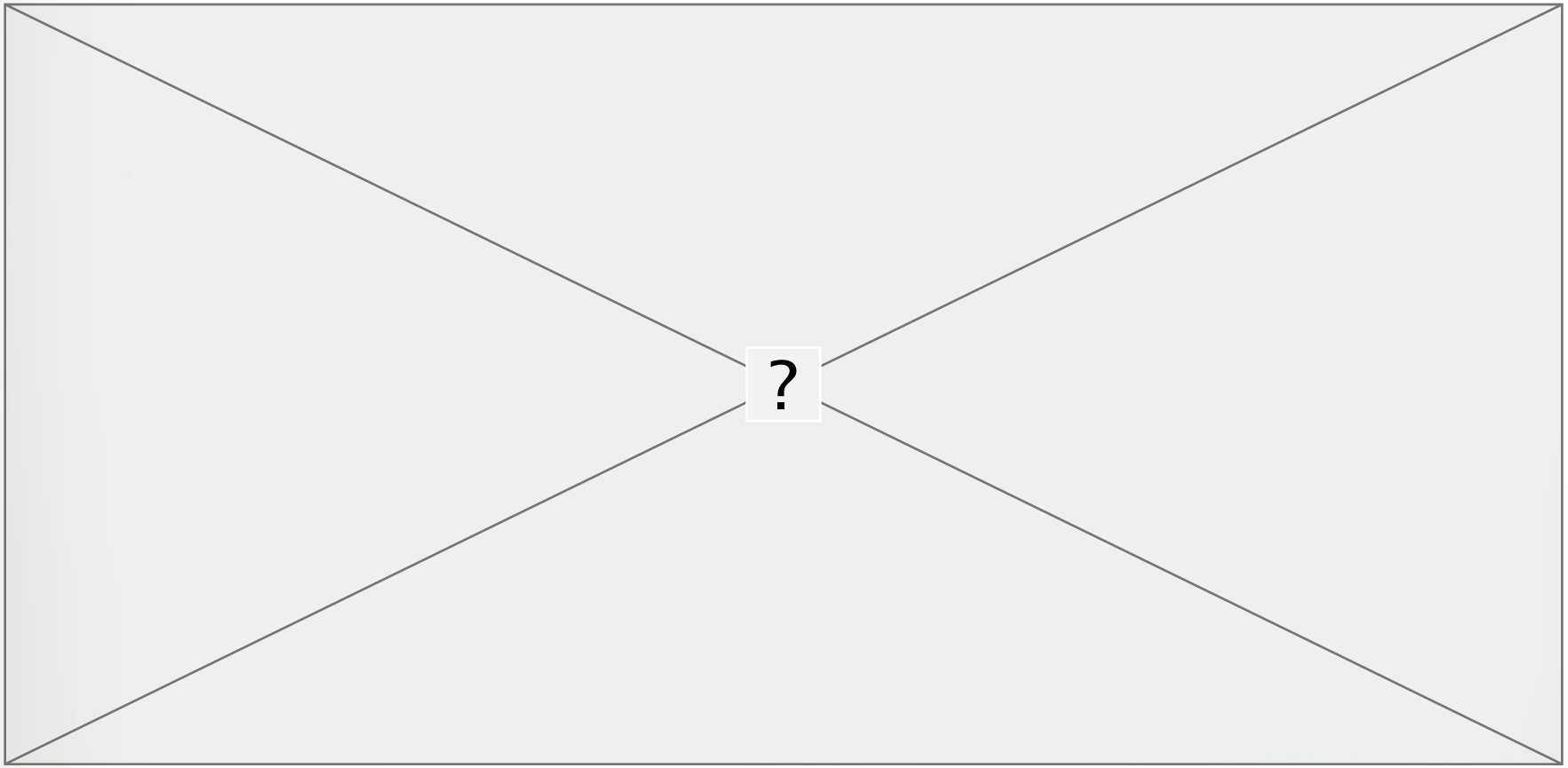
Dedicated shared HAProxy



Dedicated shared HAProxy



HAProxy on application side



Agenda

- Default asynchronous MySQL Replication
- Percona XtraDB Cluster:
  - Introduction / Features / Load Balancing
- **Use Cases:**
  - **High Availability / WAN Replication / Read Scaling**
- Limitations
- Future

Use Cases

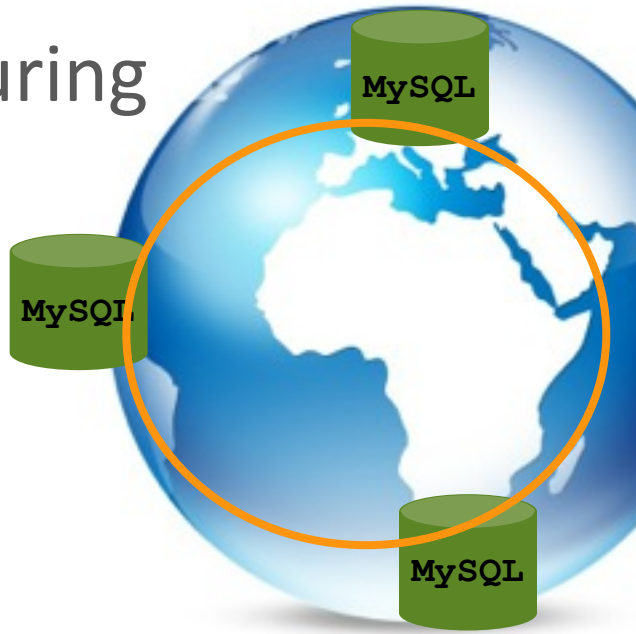
- High Availability
- WAN Replication
- Read Scaling

High Availability

- Each node is the same (no master-slave)
- Consistency ensured, no data loss
- Quorum avoids split-brain
- Cluster issues are immediately handled on
- no ‘failover’ necessary
- no external scripts, no SPOF

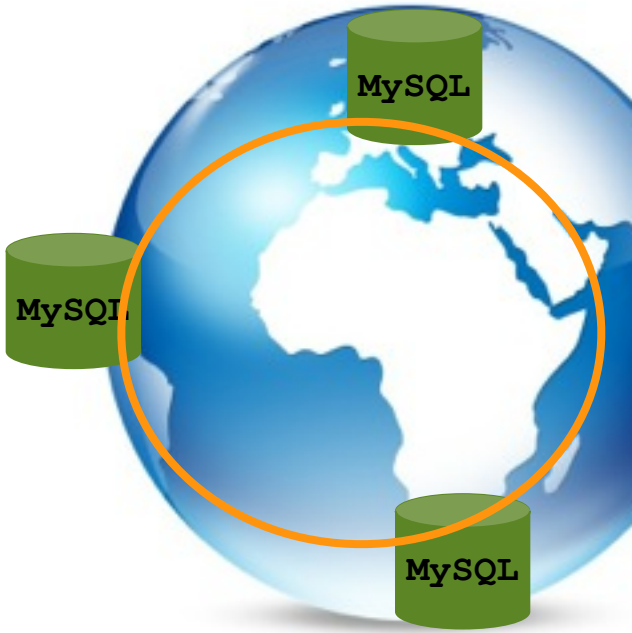
WAN replication

- No impact on reads
- No impact within a trx
- Communication only happens during COMMIT (or if autocommit=1)
- Use higher timeouts and send windows



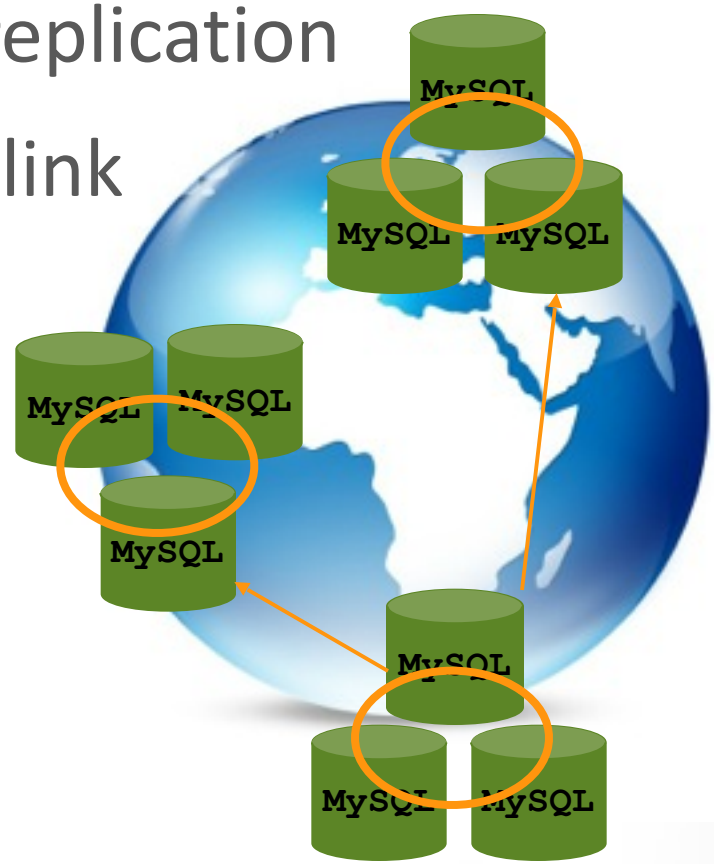
WAN replication - latency

- Beware of increased latency
- Within EUROPE EC2
  - COMMIT: 0.005100 sec
- EUROPE <-> JAPAN EC2
  - COMMIT: 0.275642 sec



WAN replication with MySQL asynchronous replication

- You can mix both types of replication
- Good option on slow WAN link
- Requires more nodes
- If binlog position is lost, full cluster must be reprovisioned (*)



Agenda

- Default asynchronous MySQL Replication
- Percona XtraDB Cluster:
  - Introduction / Features / Load Balancing
- Use Cases:
  - High Availability / WAN Replication / Read Scaling
- **Limitations**
- Future

Limitations

- Supports only **InnoDB** tables
  - MyISAM support will most likely stay in *alpha*.
- The **weakest** node **limits** **write performance**
- All tables must have a **Primary Key**!

Limitations

- **Large Transactions** are not recommended if you write on all nodes simultaneously
- **Long Running Transactions**
- If the workload has a **hotspot** then (frequently writing to the same rows across multiple nodes)
- Solution: Write to only 1 node

Limitations

- WAN Replication: All nodes connect to all nodes, causing some network overhead
- Mixing Galera with asynchronous replication is hard to manage (no GTID support)

Agenda

- Default asynchronous MySQL Replication
- Percona XtraDB Cluster:
  - Introduction / Features / Load Balancing
- Use Cases:
  - High Availability / WAN Replication / Read Scaling
- Limitations
- **Future**

Galera 3.0 - Currently BETA

- MySQL 5.6 Support
- GTID: solves many issues with mixing asynchronous replication.
- Improved WAN support (cluster segmentation)
- Performance improvements
- Better large TRX handling

Credits

- WSREP patches and Galera library is developed by Codership Oy
  http://www.codership.com
- Percona & Codership will present on Percona Live UK 2013, Nov 11-12
  http://www.percona.com/live/london-2013/

Summary

- Default asynchronous MySQL Replication
- Percona XtraDB Cluster:
  - Introduction / Features / Load Balancing
- Use Cases:
  - High Availability / WAN Replication / Read Scaling
- Limitations
- Future

Resources

Kenny Gryp
<kenny.gryp@percona.com>

Questions?

- Percona XtraDB Cluster website:
  http://www.percona.com/software/percona-xtradb-cluster/
- Codership website:
  http://www.codership.com/wiki/doku.php
- PXC articles on mysqlperformanceblog:
  http://www.mysqlperformanceblog.com/category/percona-xtradb-cluster/
- Test it now using Vagrant !
  https://github.com/grypyrg/vagrant-percona-playground
  https://github.com/lefred/percona-cluster
  https://github.com/percona/xtradb-cluster-tutorial/tree/v2