

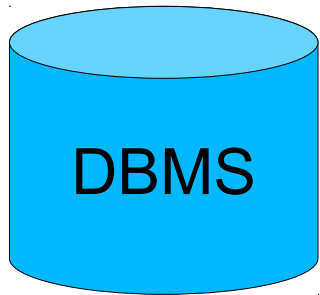
Galera Replication - Synchronous Multi-Master

Seppo Jaakola, Codership
Alexey Yurchenko, Codership

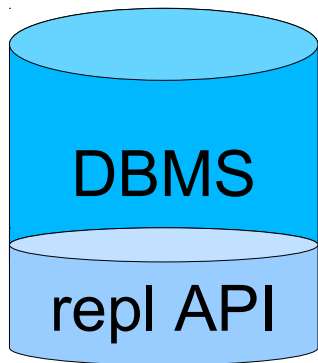
Contents

1. Galera Cluster
2. Replication API
3. Benchmarking
4. Installation & Management
5. Galera Project

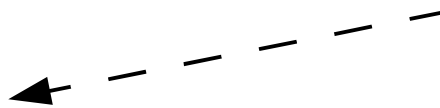
DBMS Replication



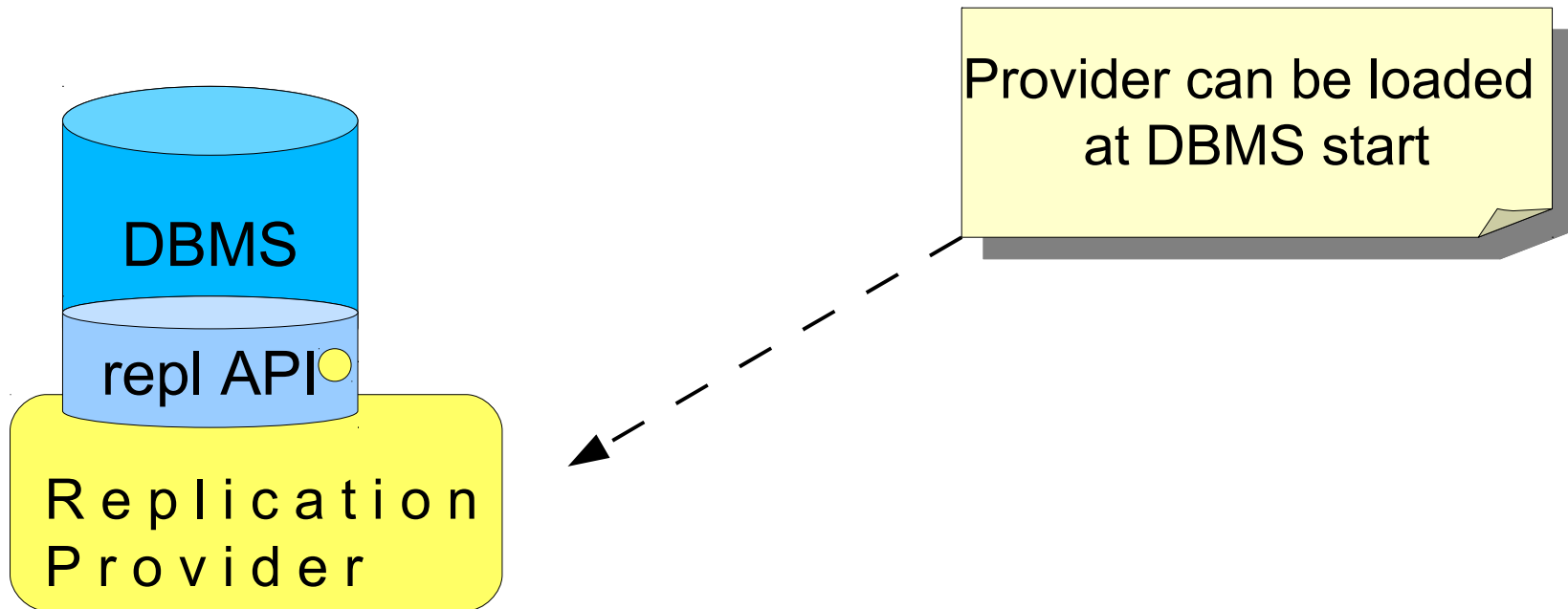
Replication API



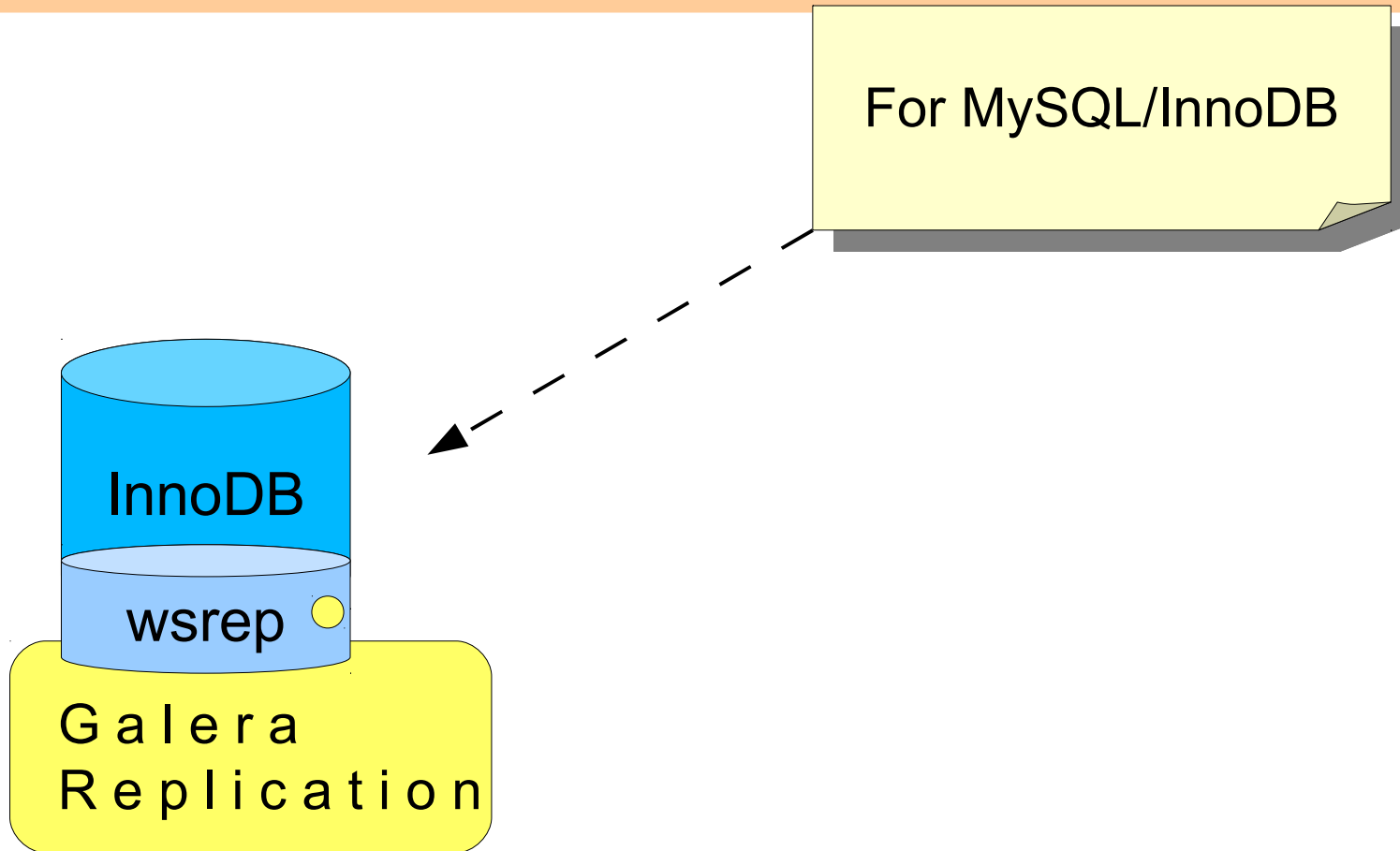
Interface for replication system
→ Calls for replication
→ Callbacks from replication
Plugin framework



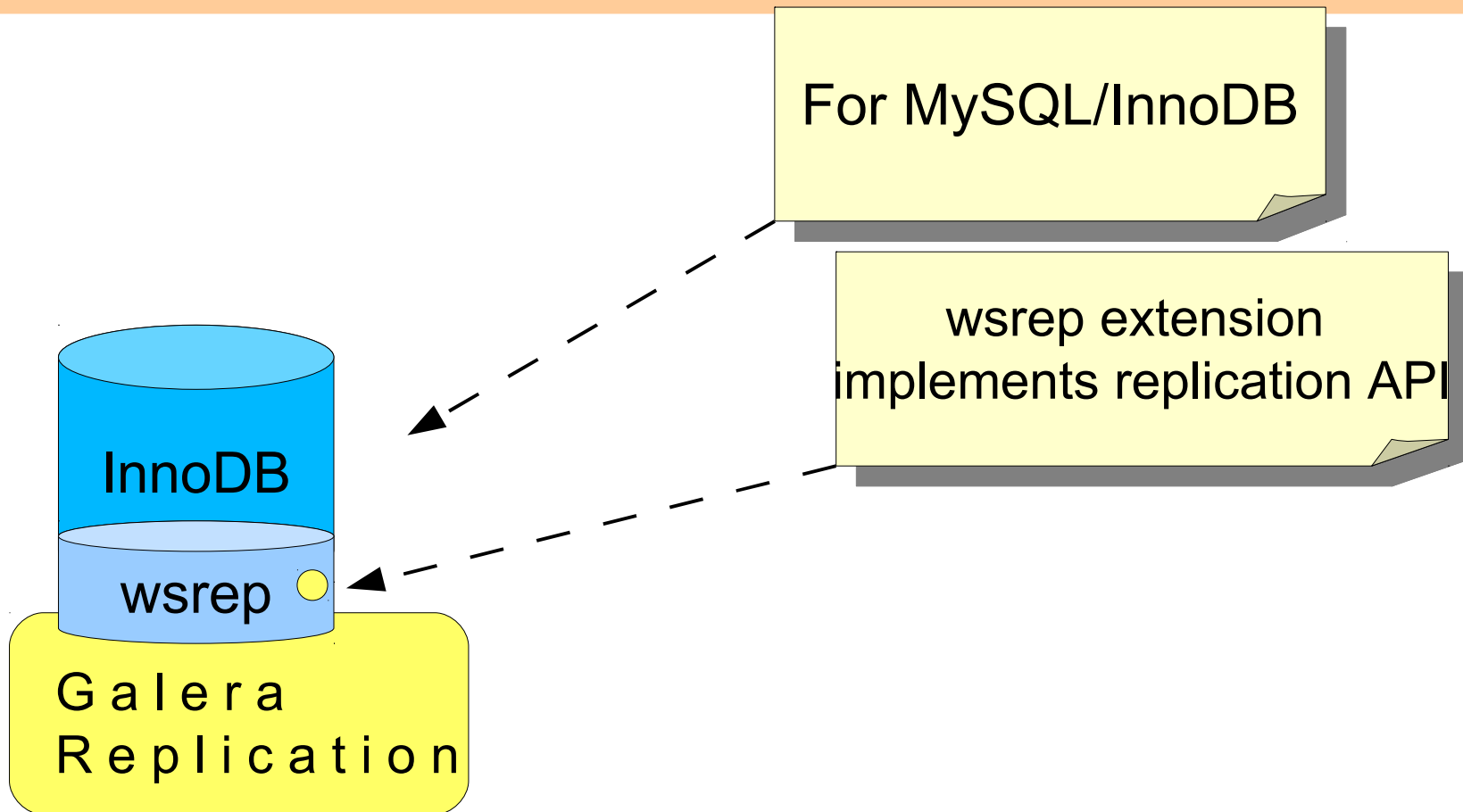
Pluggable Replicator



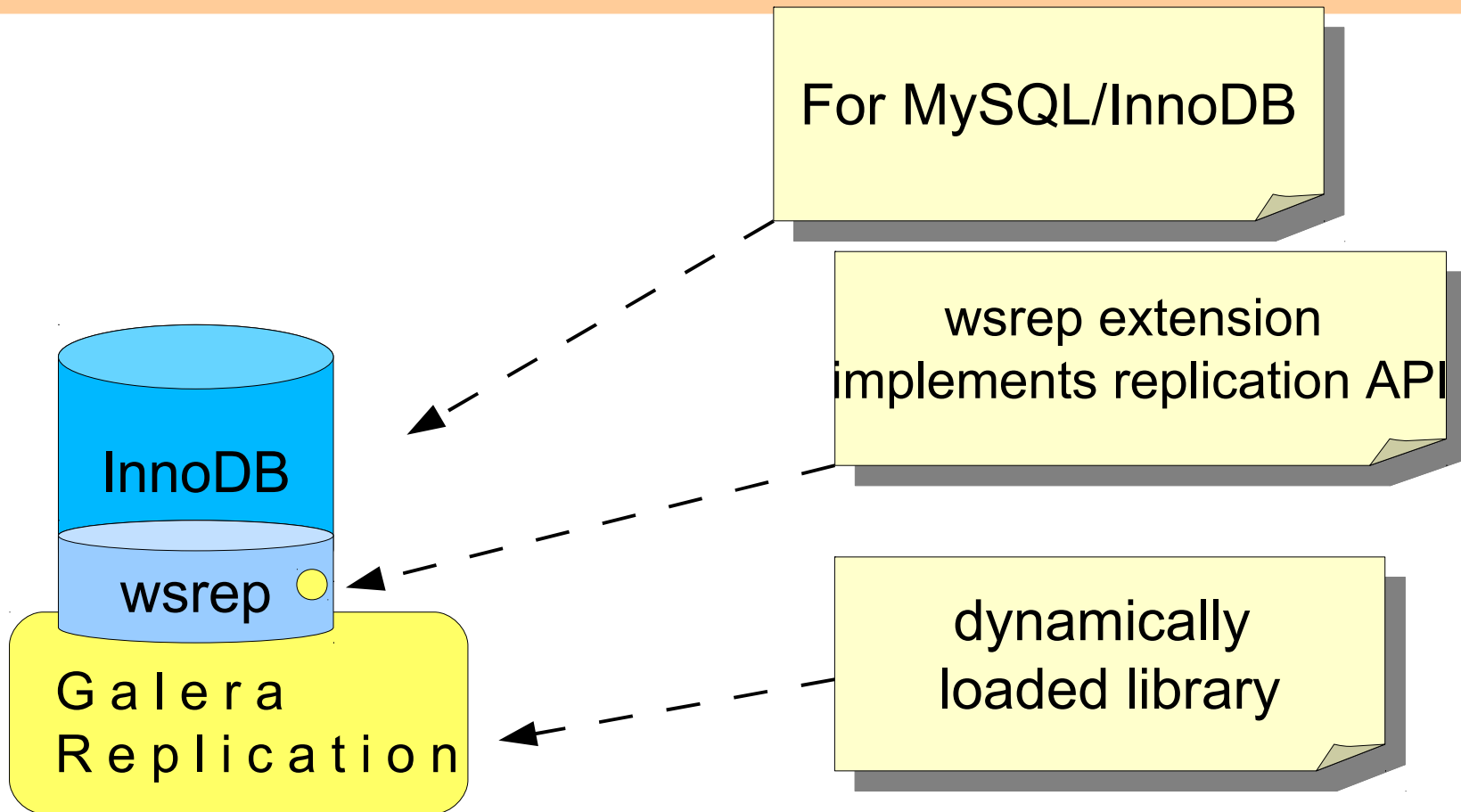
MySQL/Galera Cluster



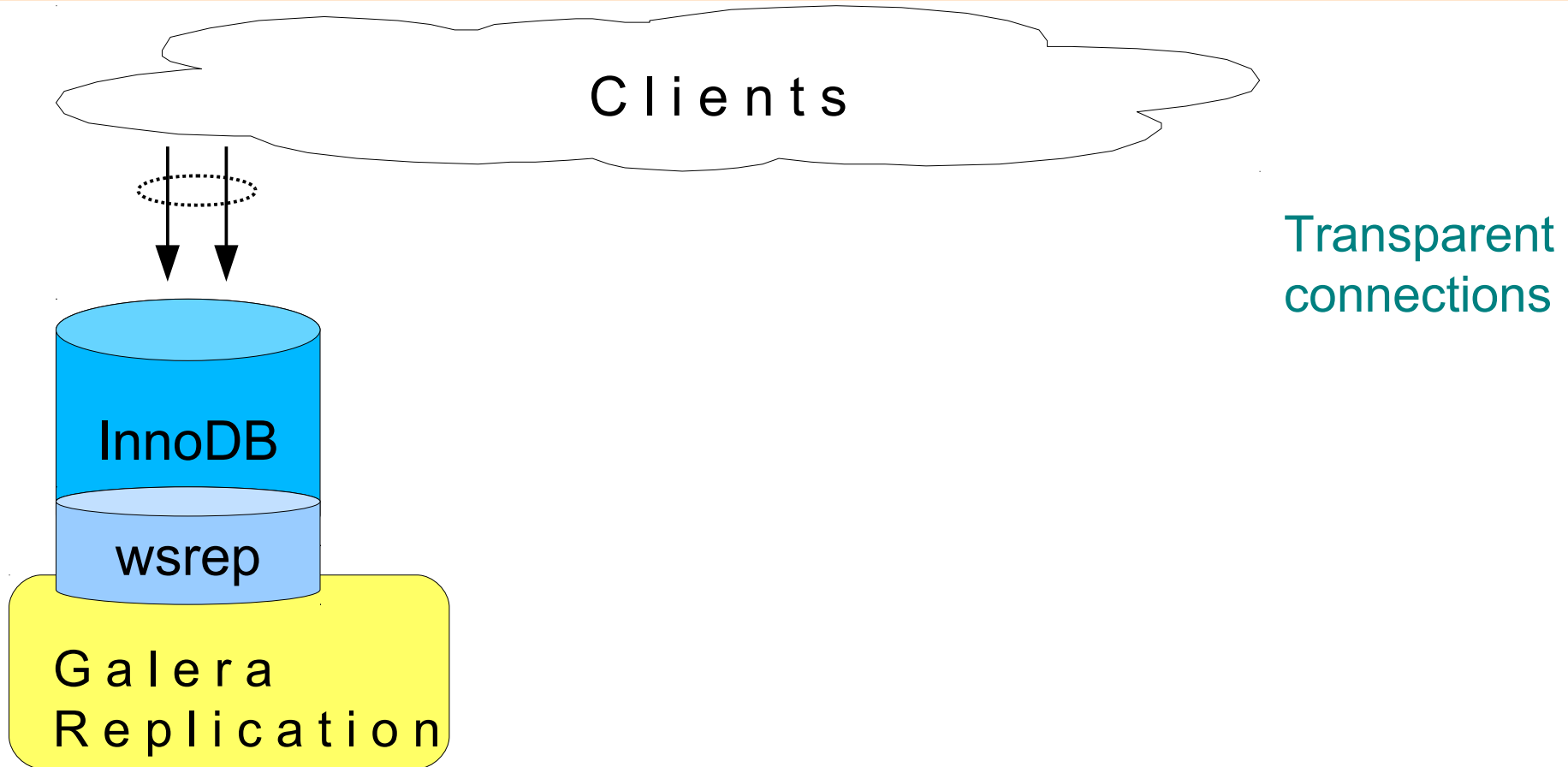
Galera Cluster



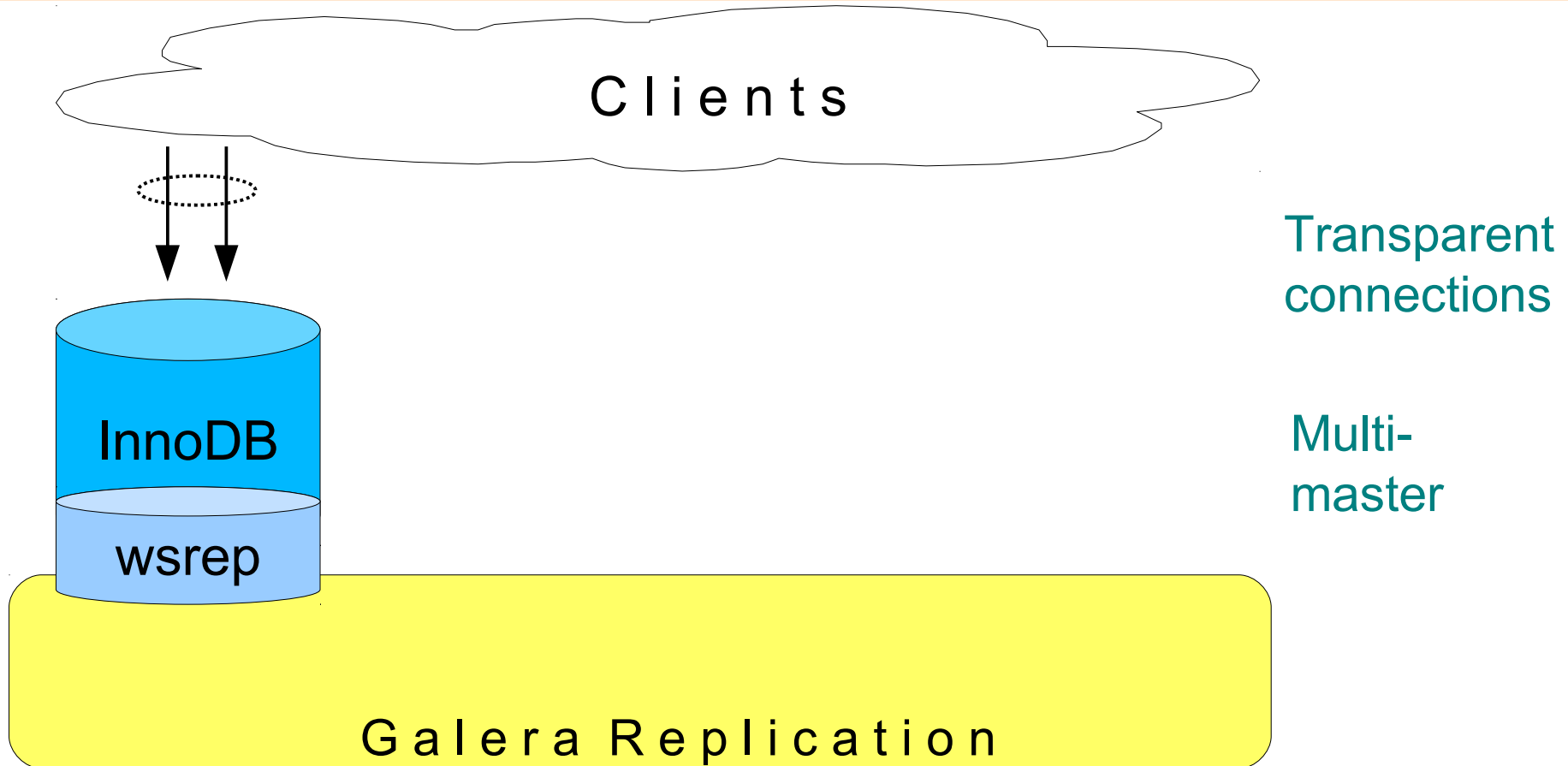
Galera Cluster



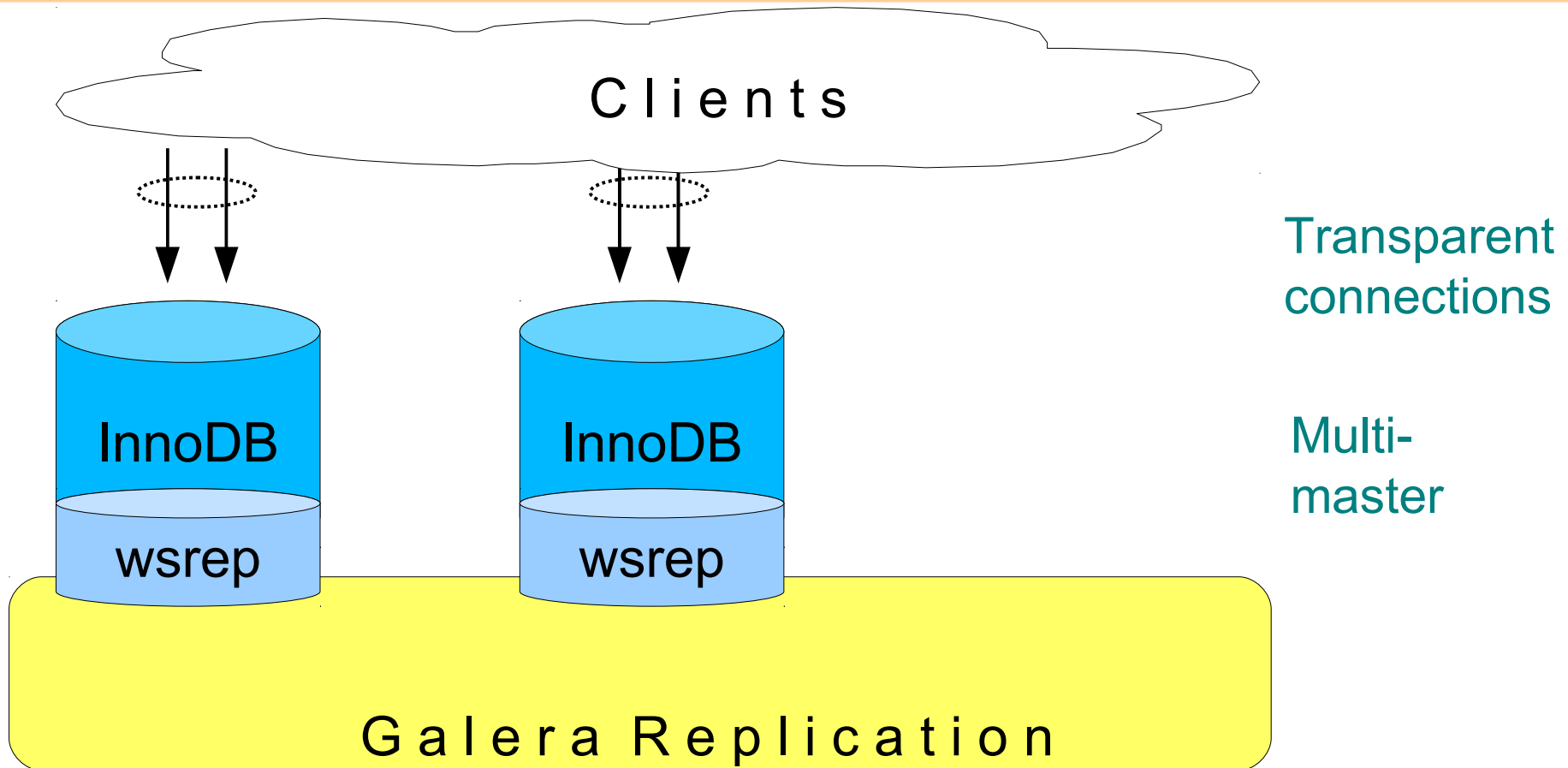
Galera Cluster



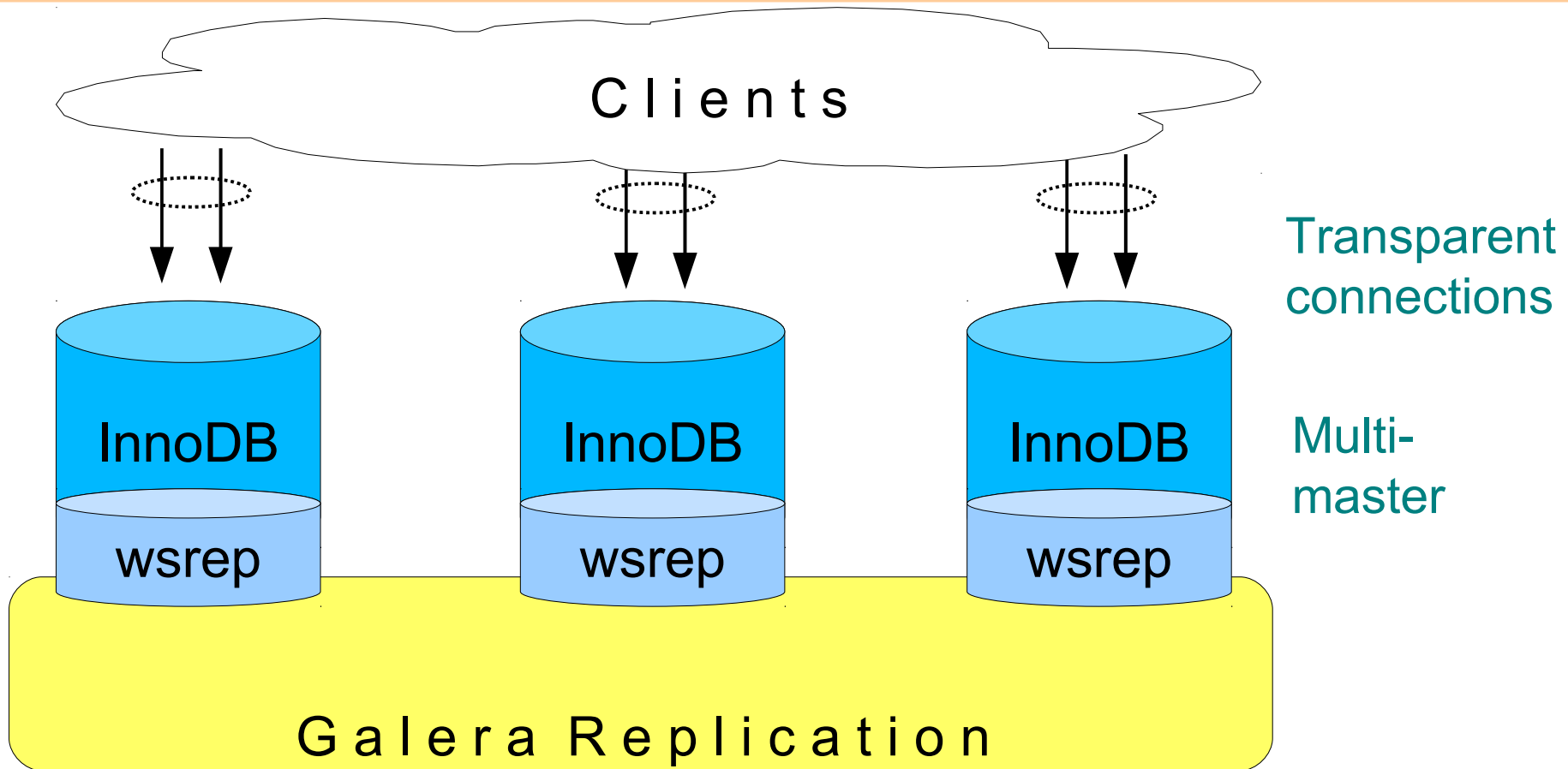
Multi-Master



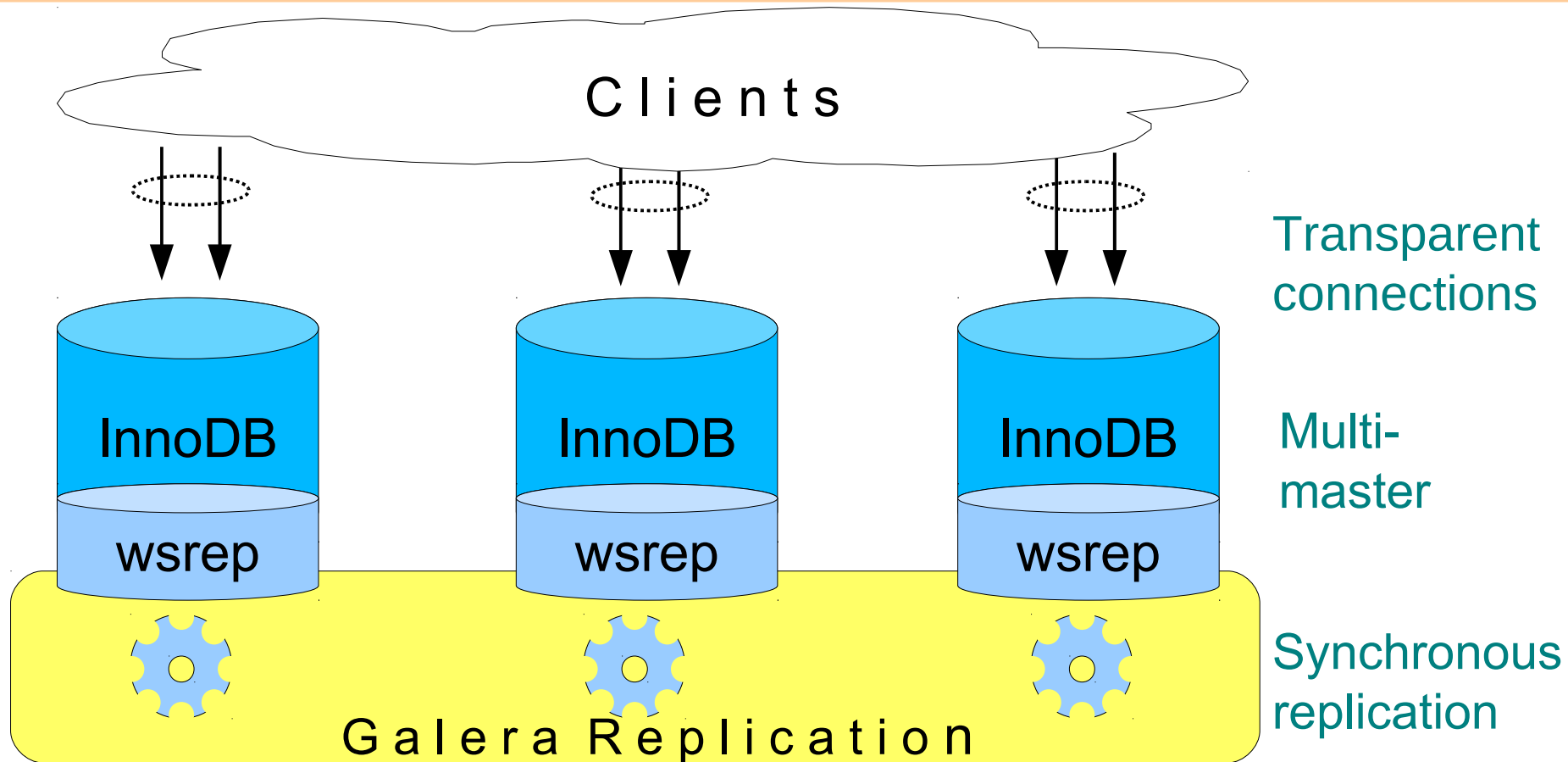
Multi-Master



Multi-Master



Synchronous Replication



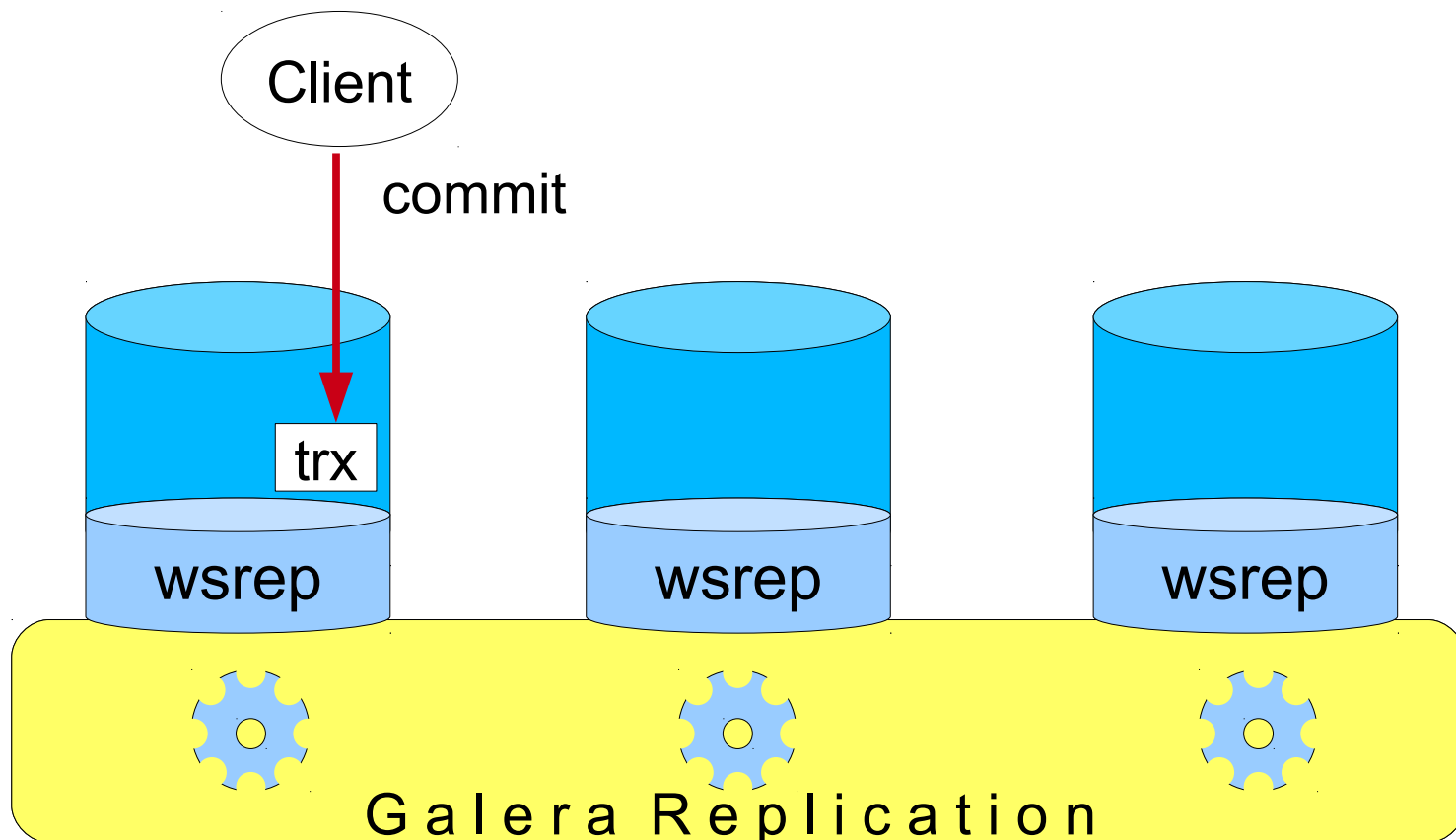
Galera Replication

- Synchronous multi-master replication
 - High Availability
- No middle-ware, direct DBMS connections
 - Transparency
- Row events, row level locking
 - Write scalability
- Certification based replication method

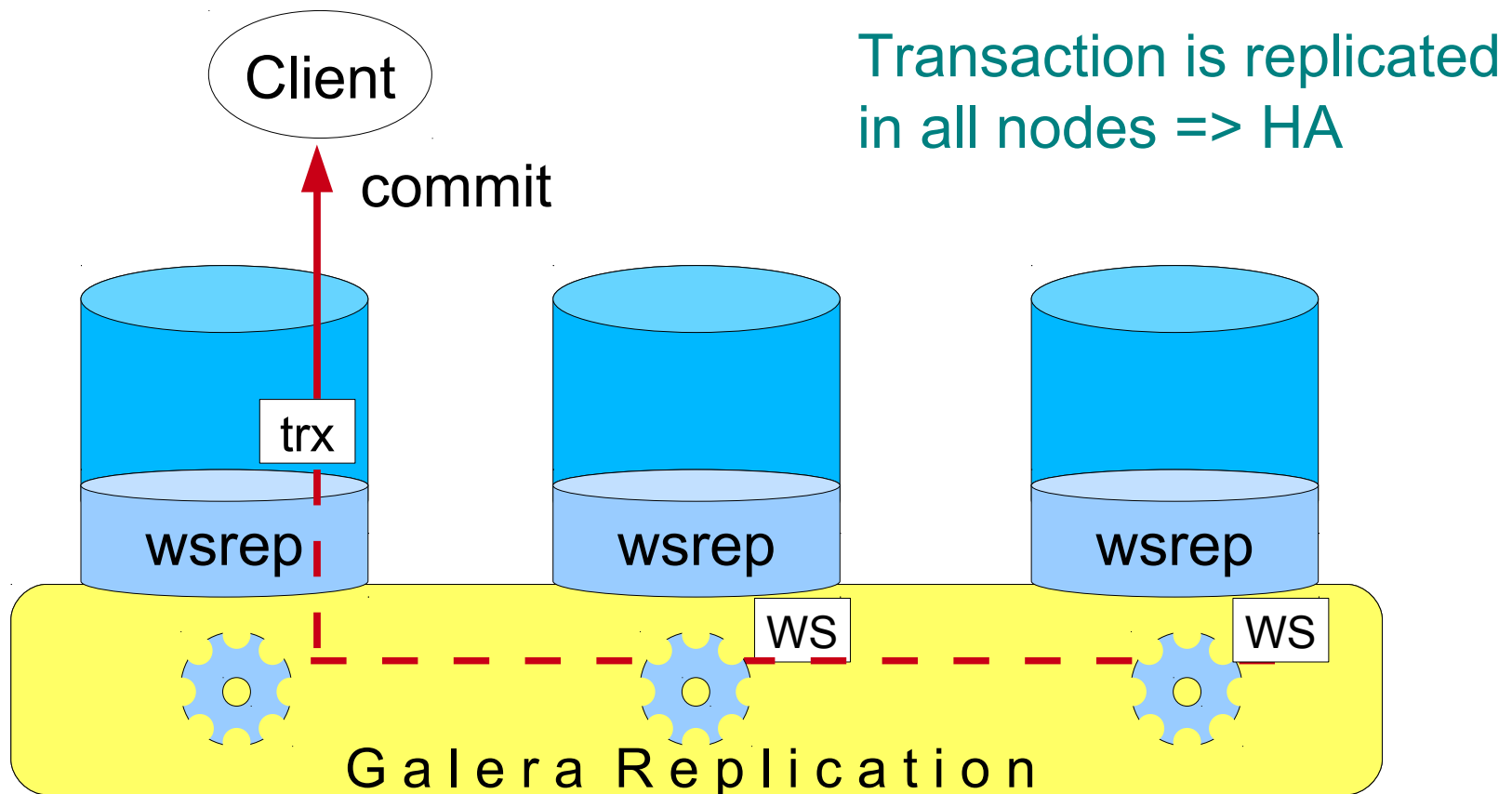
Galera Replication

- Global Transaction ID
 - No lost transactions
- Parallel Applying
 - Effective Replication
- Automatic node join/provisioning
 - Simple cluster management

Synchronous Replication



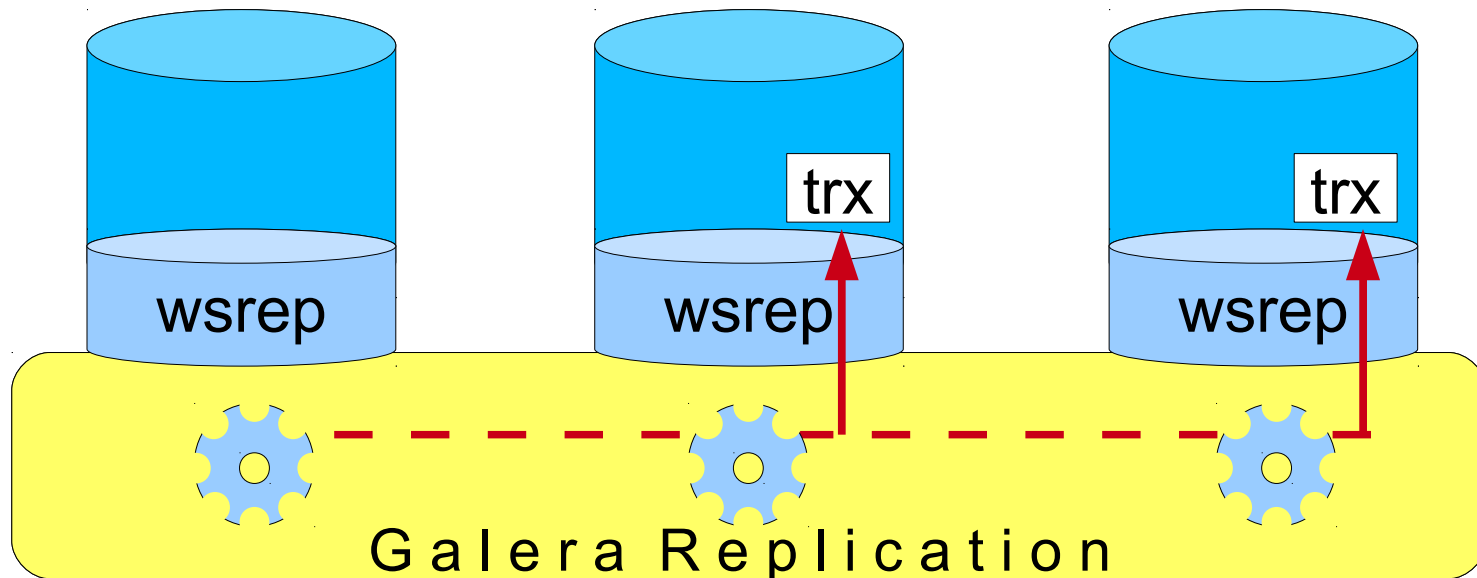
Synchronous Replication



Synchronous Replication

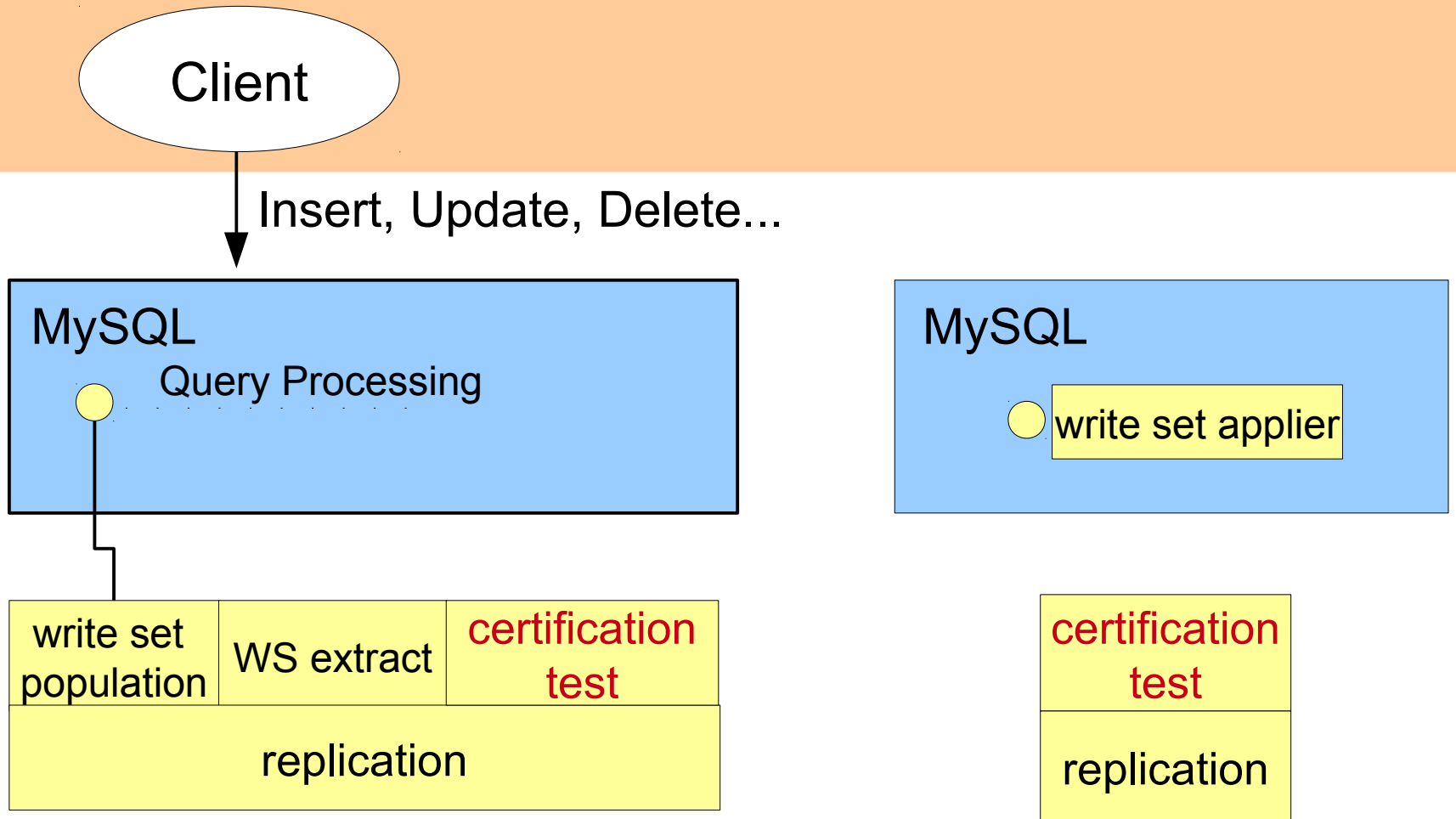
Client

Transaction is applied at later time
=> virtually synchronous

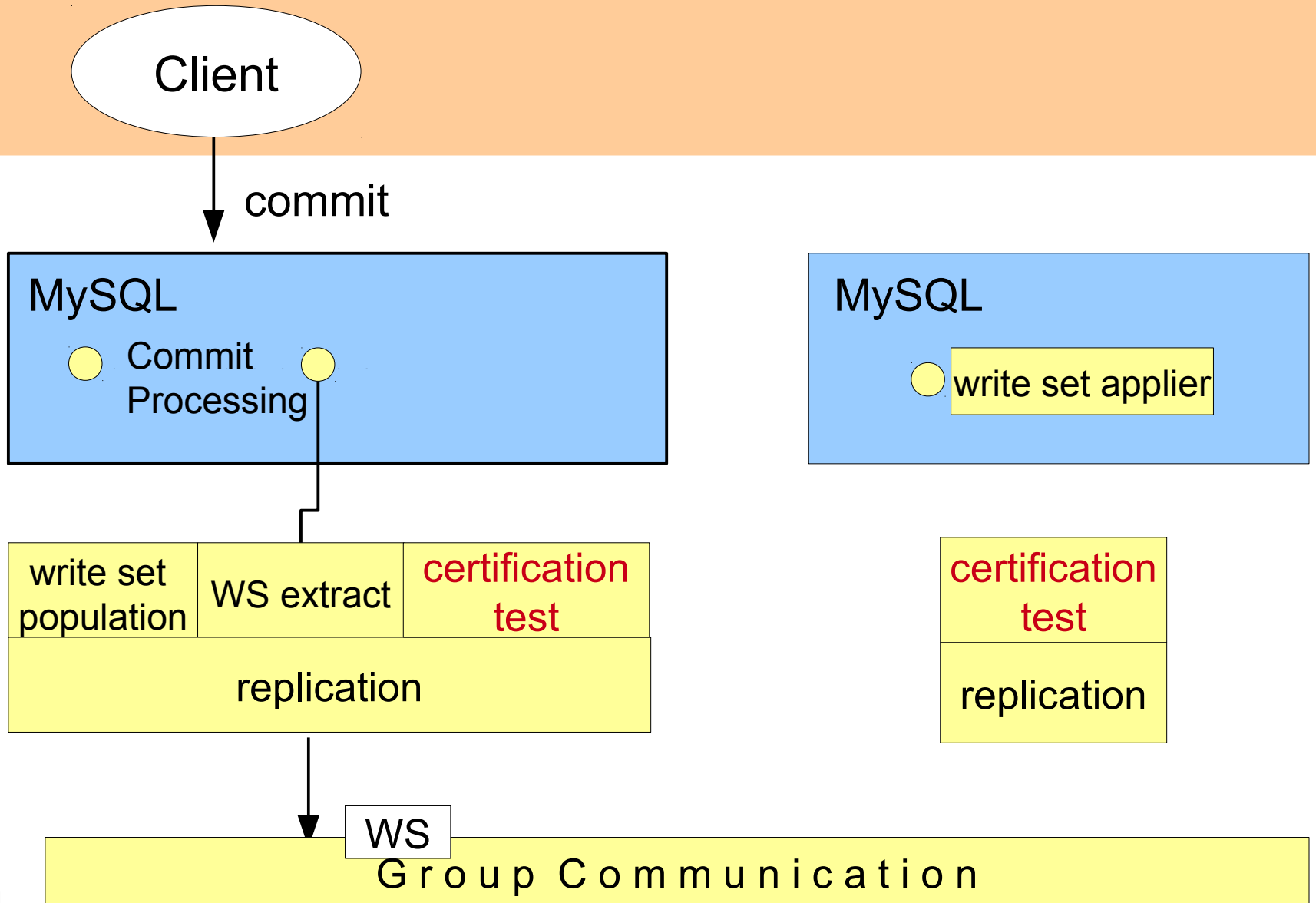


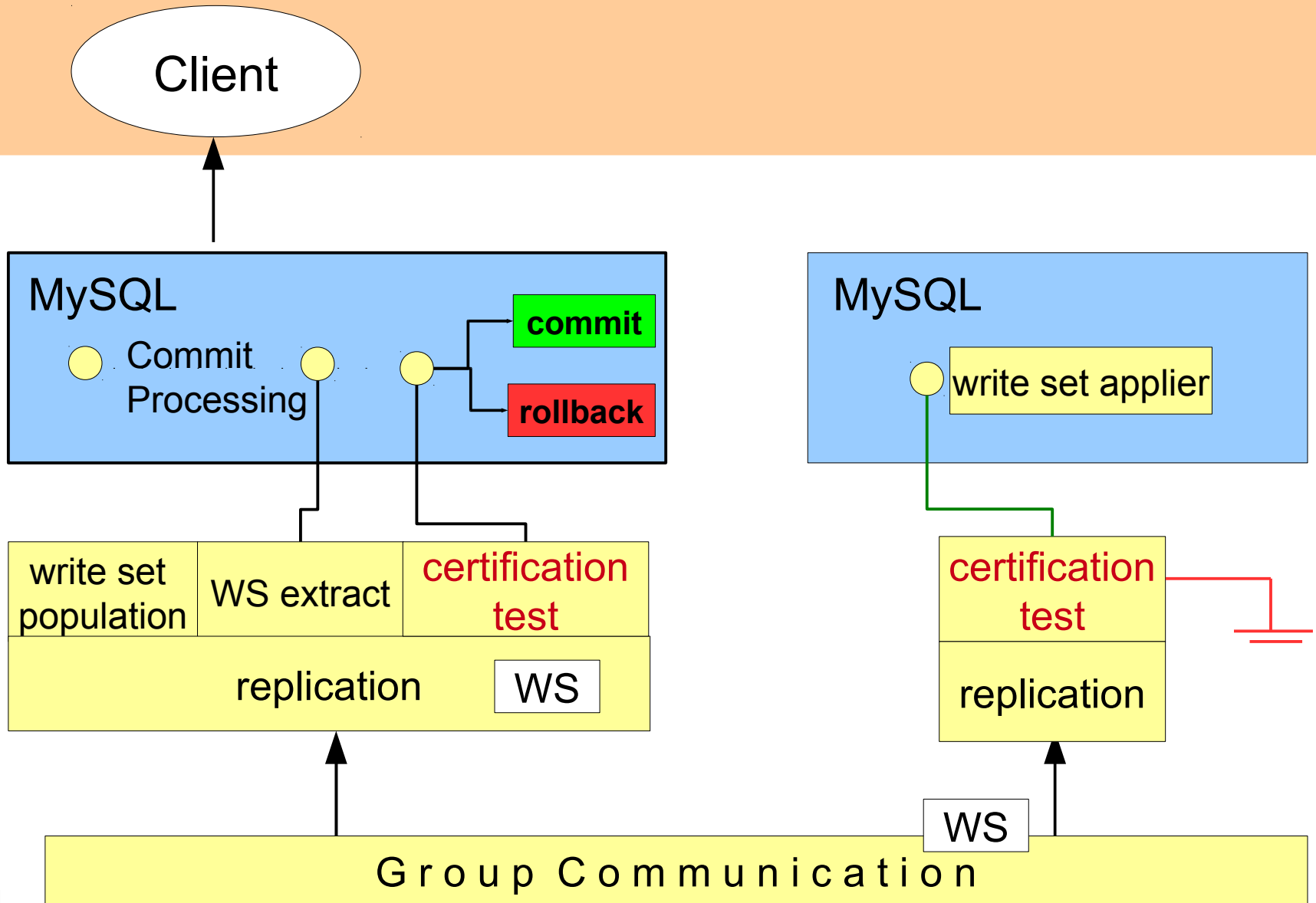
Certification Based Replication

- Transactions process independently in each cluster node
- Transaction write sets will be replicated at commit time
- Cluster wide conflicts resolved by certification test



Group Communication

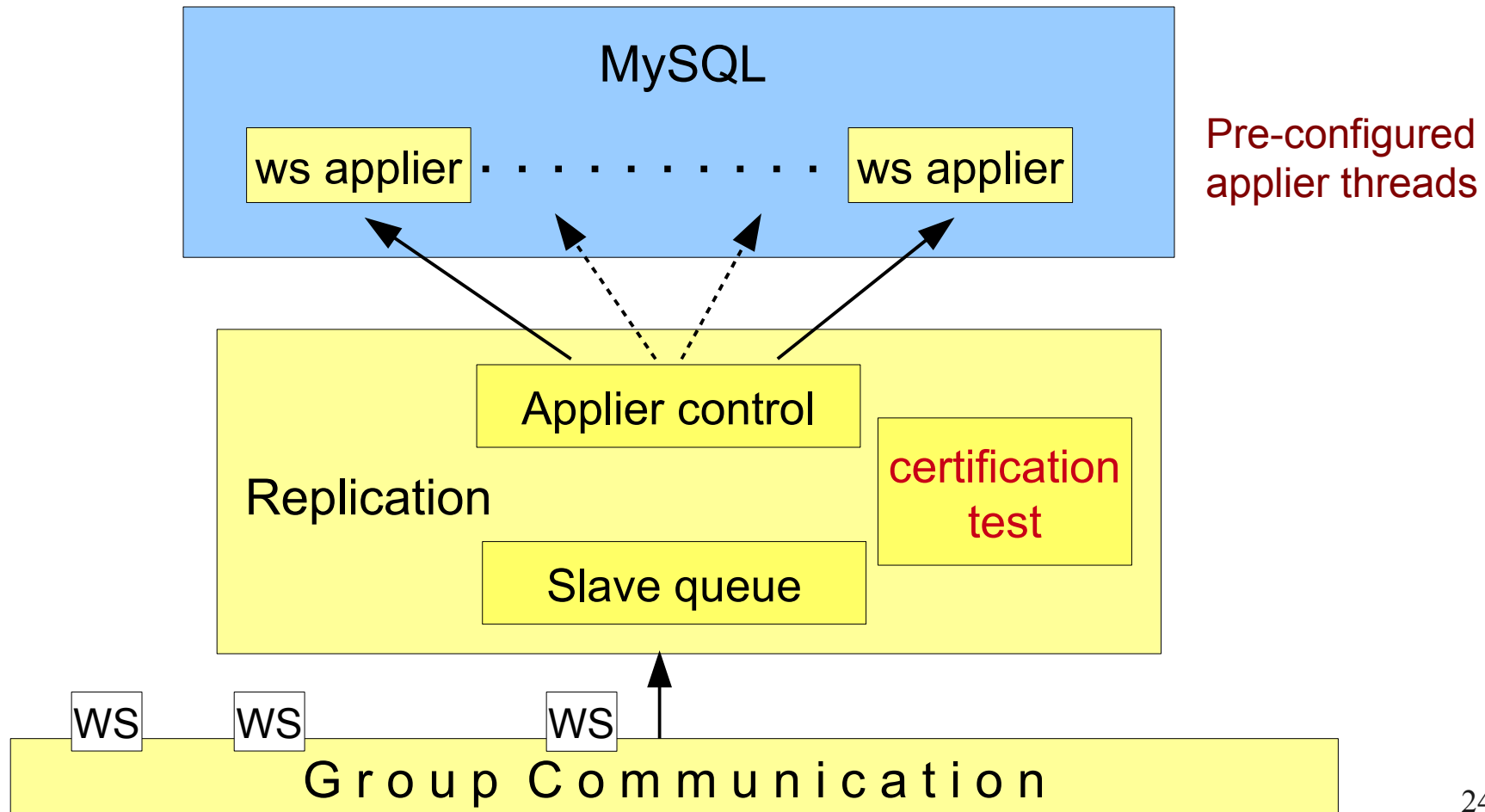




Parallel Applying

- Galera assigns non-conflicting WS tasks to parallel appliers
- Applier threads launched at MySQL startup
- 1-512 appliers can be launched
- Optimal applier count depends on work load

Parallel Applying



Global Transaction ID

- Galera assigns Global Transaction ID for all replicated transactions
- Transactions can be uniquely referenced in any node
- Helps in provisioning new nodes
- PITR

Node Provisioning

- Automatic node joining
- Cluster chooses 'donor' for the 'joiner'
- State Snapshot Transfer
- Scriptable interface, currently implemented:
 - `mysqldump`
 - `rsync`
 - ...more to come

Replication API

Replication API

- Galera integrates closely in DBMS transaction processing
- There must be an interface between DBMS and replication system

wsrep API

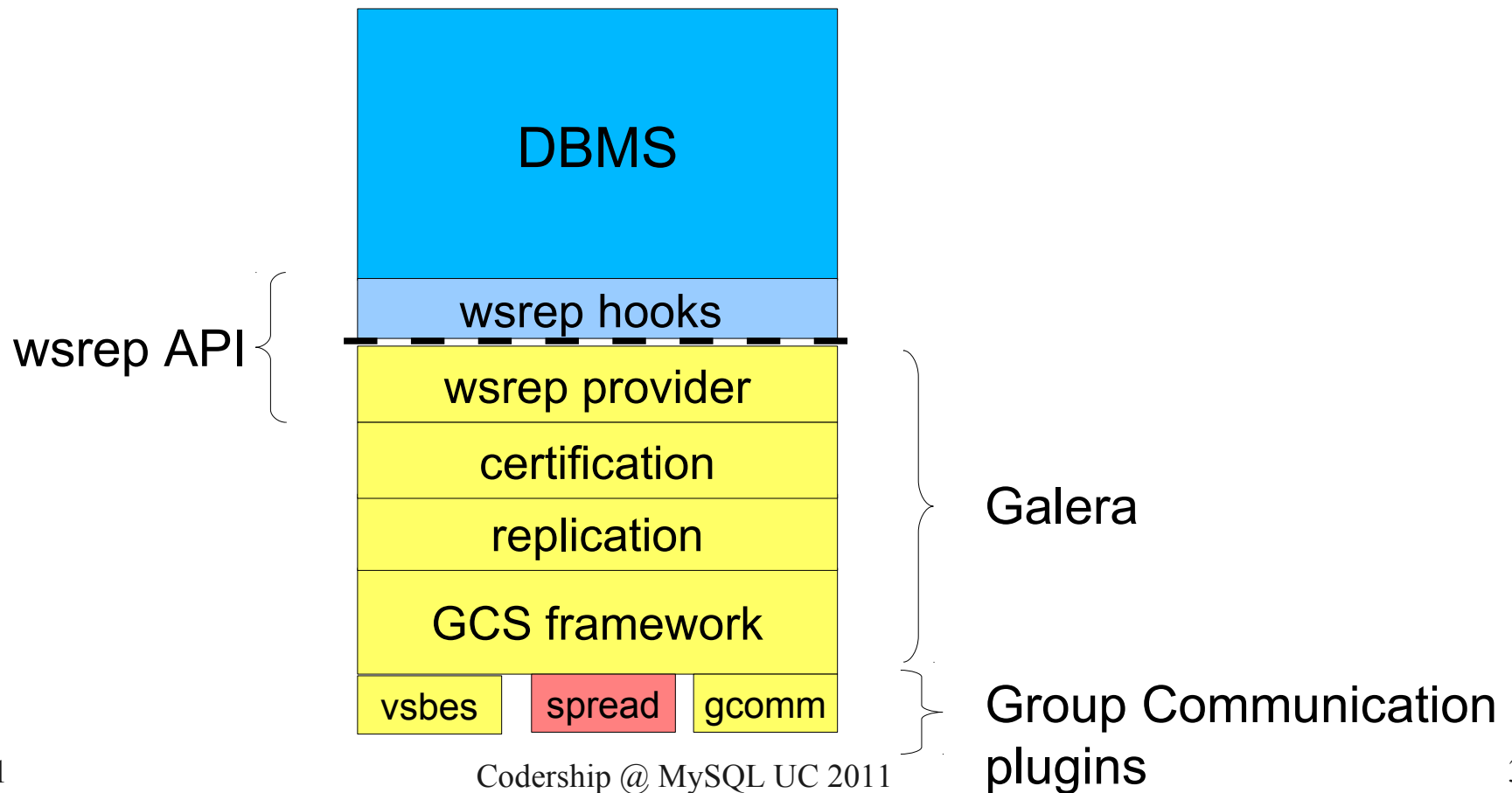
- DBMS agnostic replication interface
- Defines:
 - **Write Set** replication for transactions
 - **TO isolation** for replicating DDL
- Suitable for different replication modes (sync/async, multi-master, master/slave, PITR...)
- <https://launchpad.net/wsrep>



wsrep API Implementation

- Replication provider library
 - load/unload
 - configuration
 - status
- Write set
 - population calls
 - replication calls (at commit)
- Prioritized transactions
 - Lock queue modified
 - Aborting local victims
- TO isolation for DDL queries

Galera Library



Benchmarking

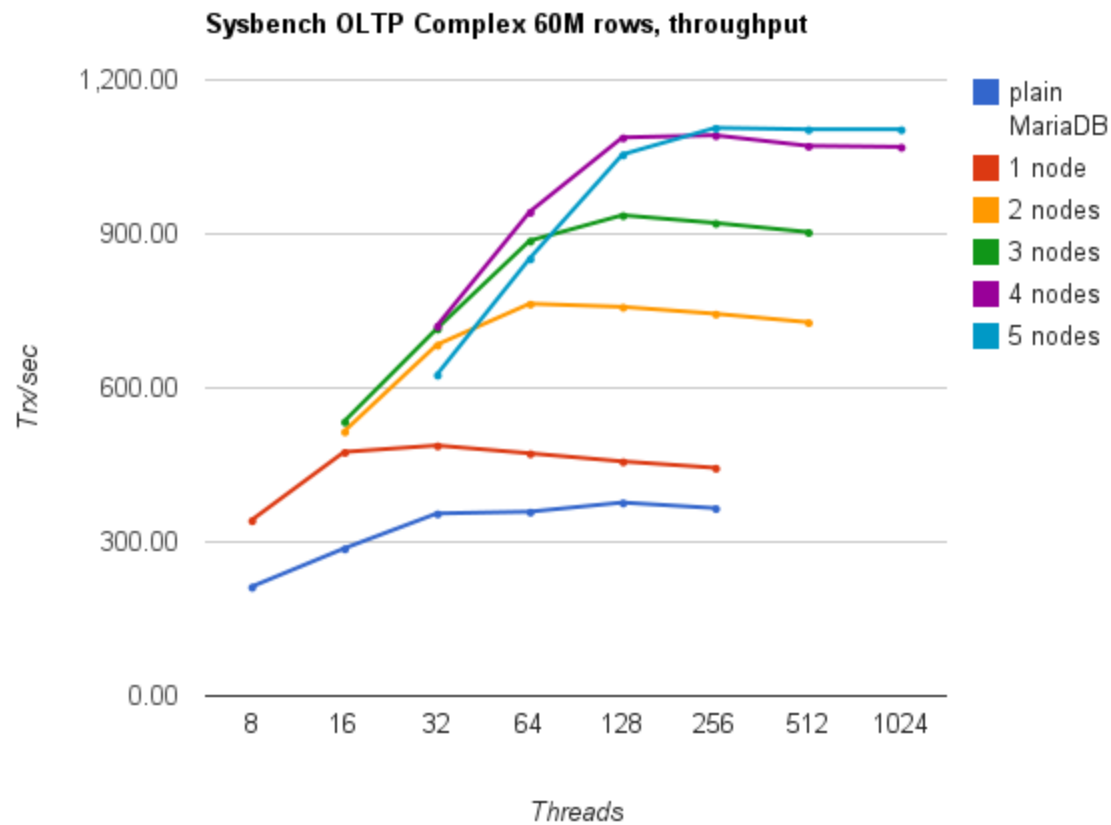
Benchmarking

- Tested with several benchmarks
 - Sysbench, dbt2, DOTS, osdb, jmeter, sqlgen...
- Tested with 'physical hardware' and with Amazon EC2 instances
- In general, shows good scalability even with write intensive work loads

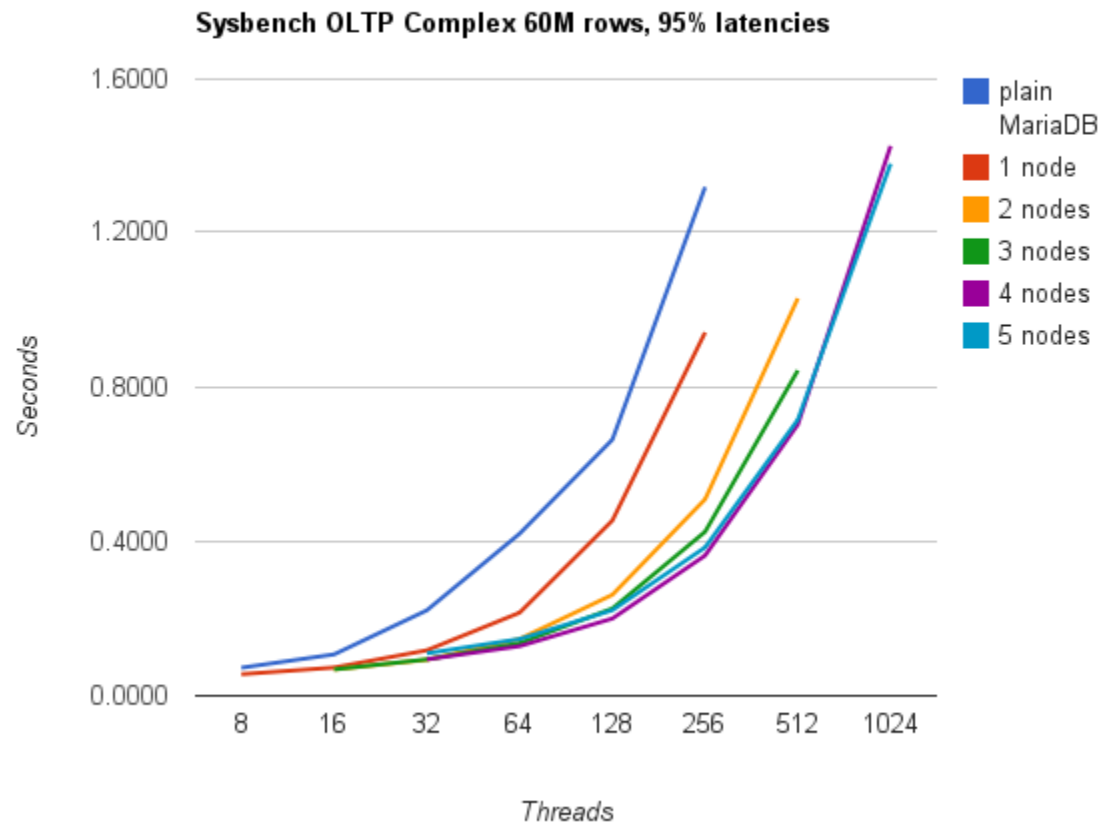
SysBench Benchmarks

- Sysbench OLTP, complex mode
- 60M rows (database size ~20 Gb)
- EC2 m1.large instances, 6Gb buffer pool size
- 24% writes

Scale Out: Throughput



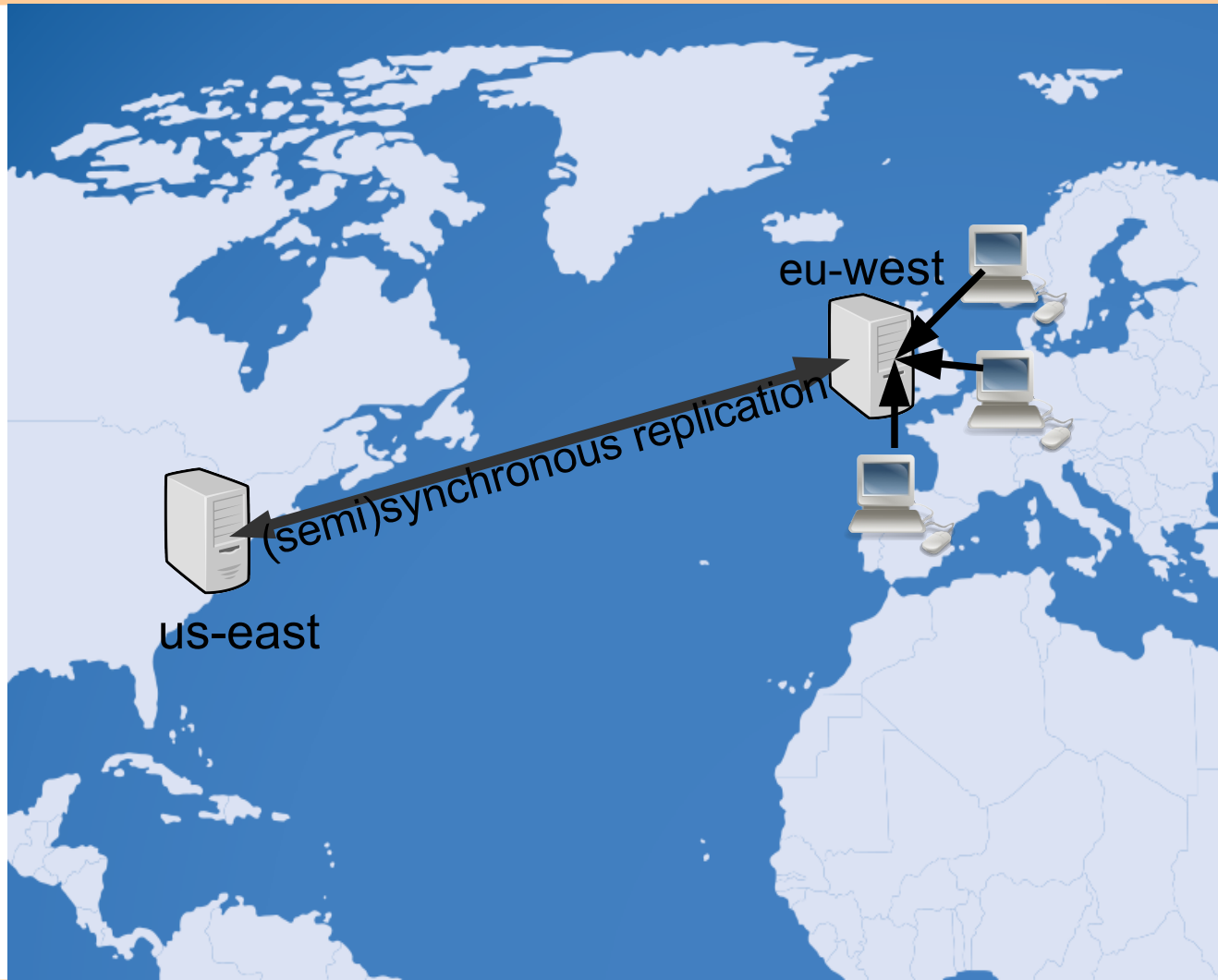
Scale Out: Latencies



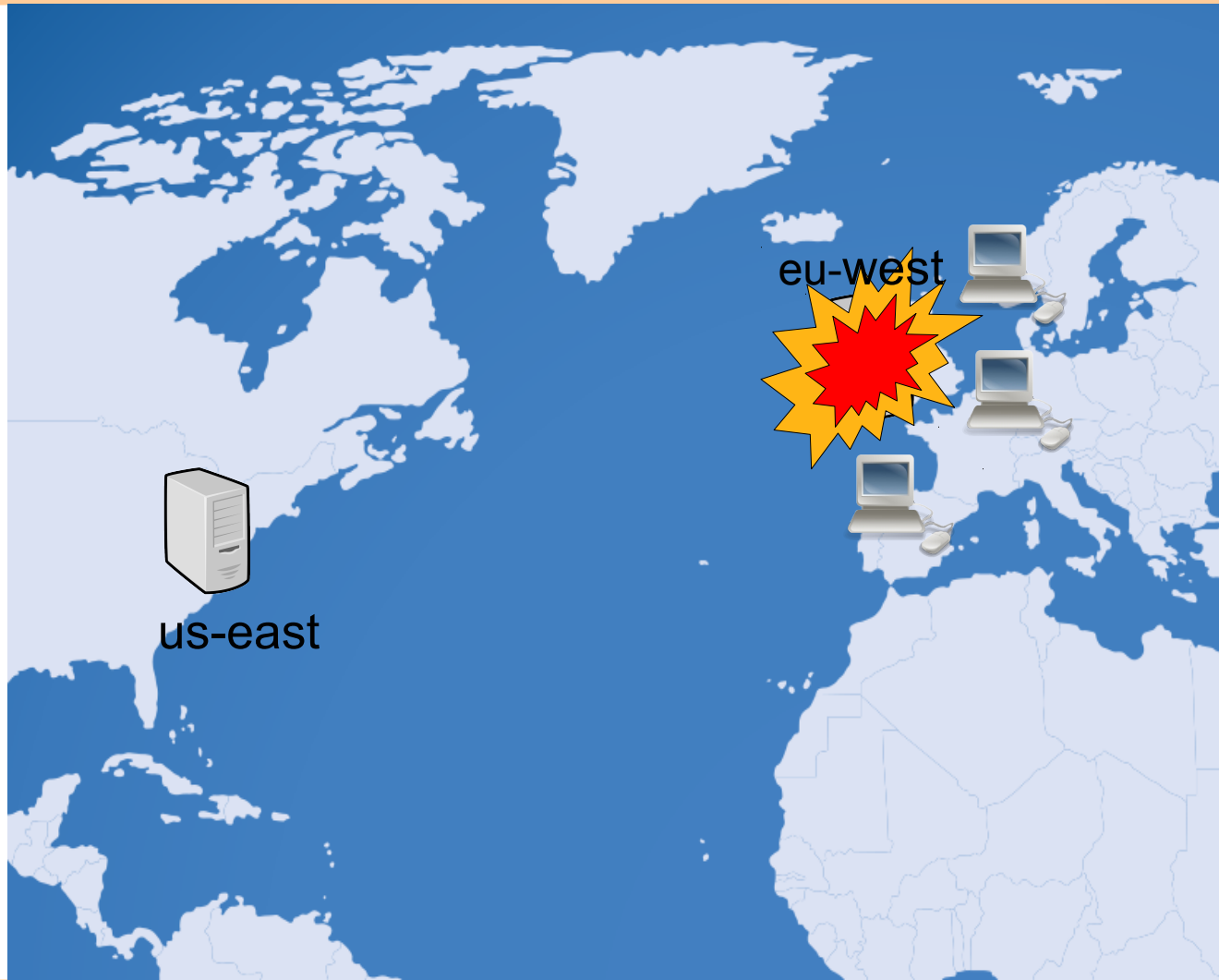
Synchronous WAN Replication

- Sysbench OLTP, complex mode
- 60M rows
- Amazon EC2 m1.large instances in US and EU zones
- Ping RTT ~90ms

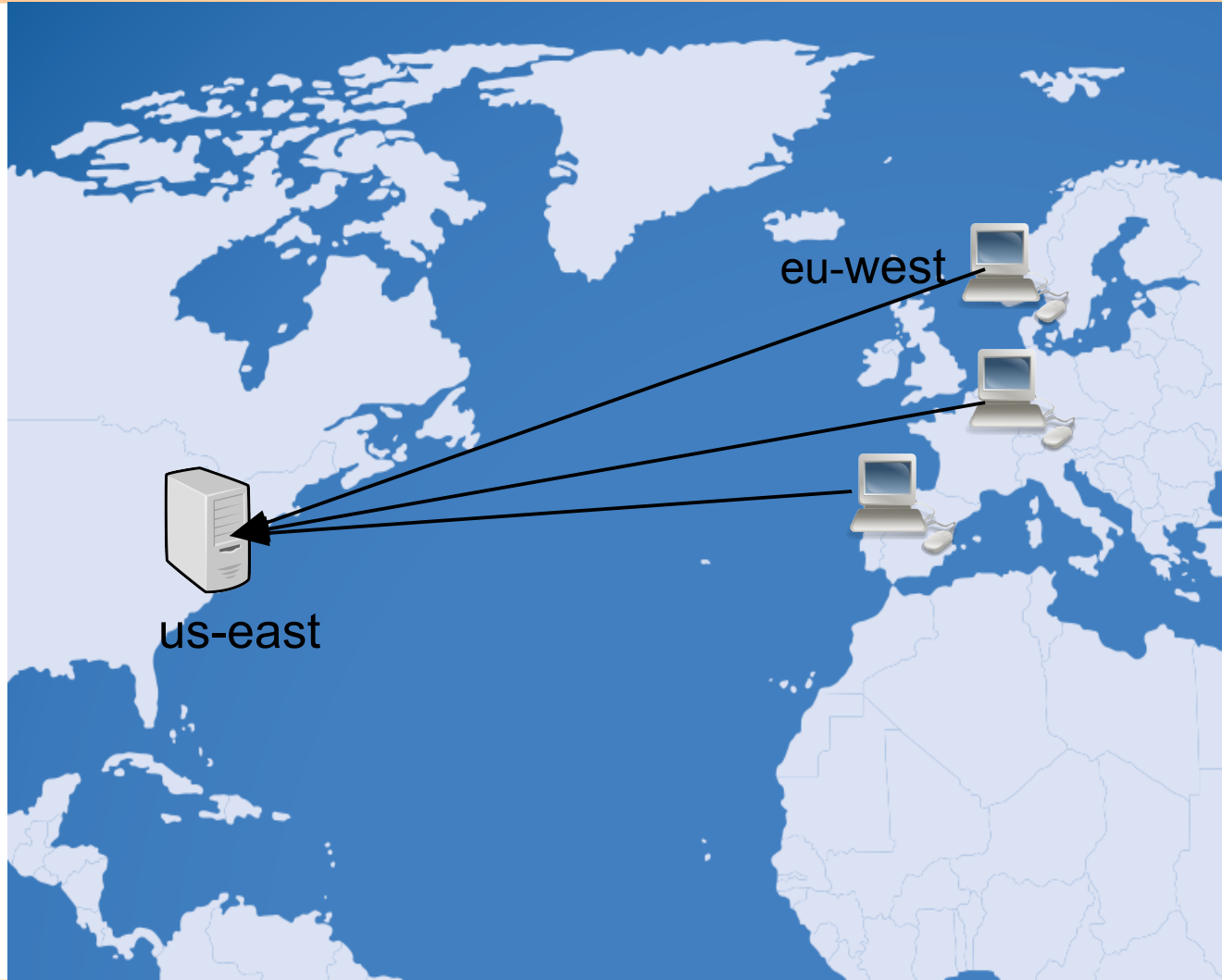
Disaster Recovery



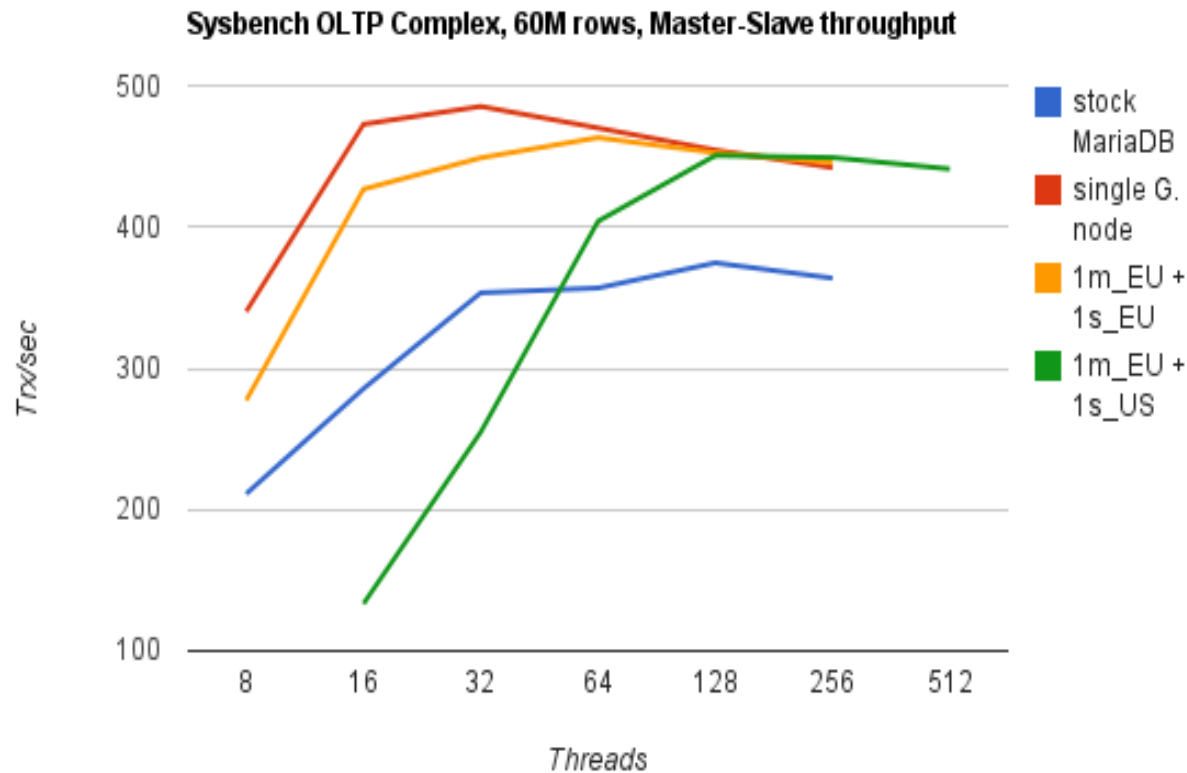
Disaster Recovery



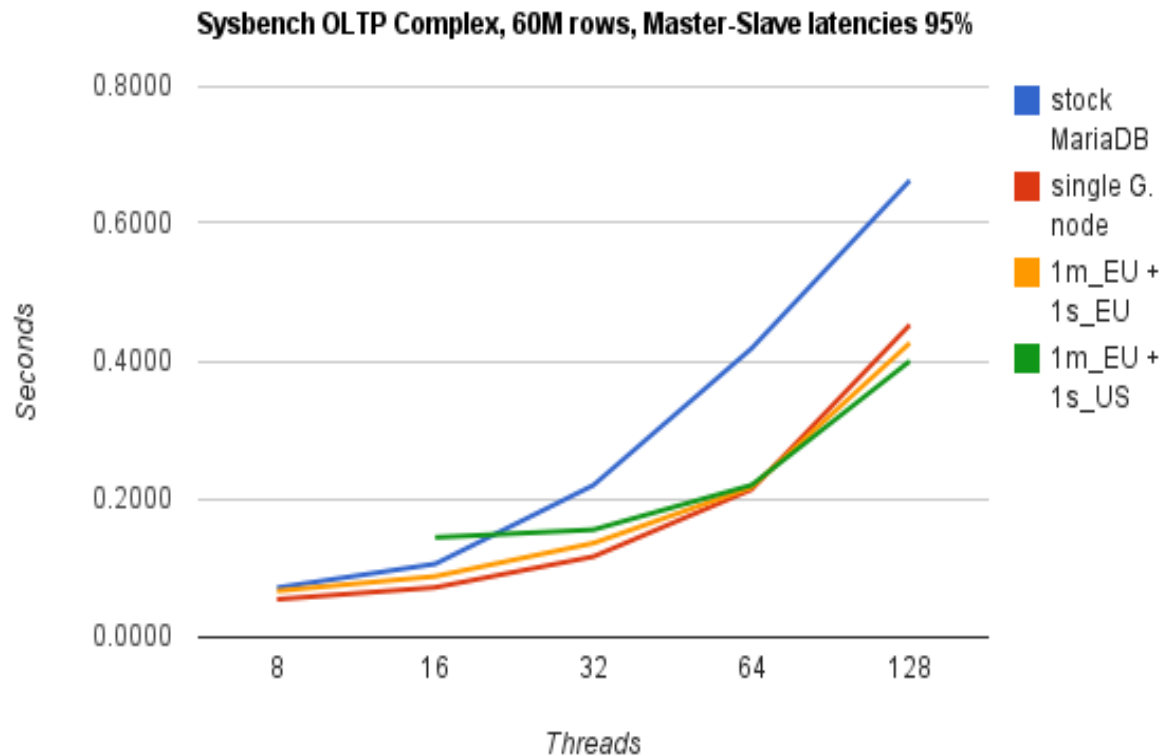
Disaster Recovery



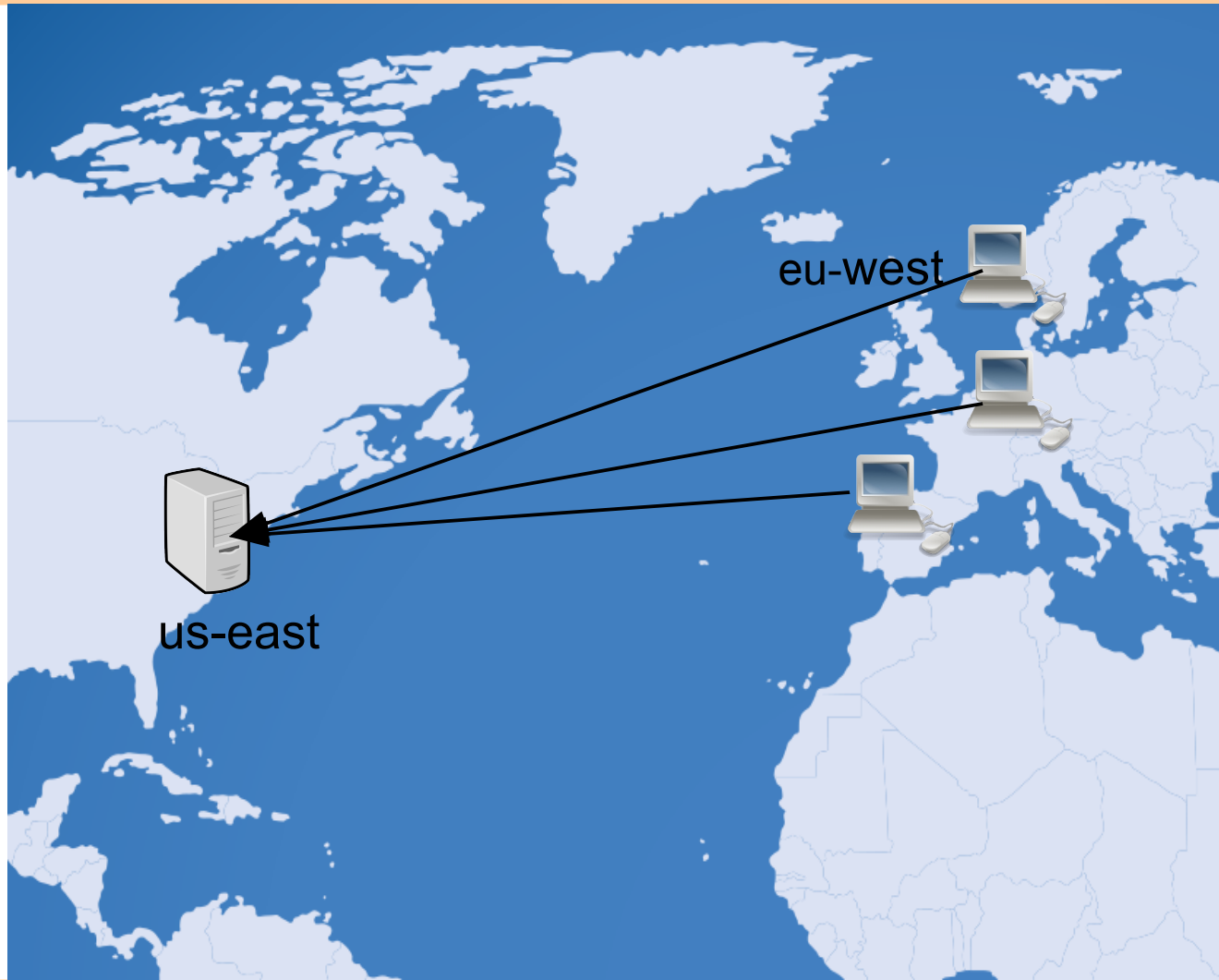
Synchronous Master-Slave WAN Throughput



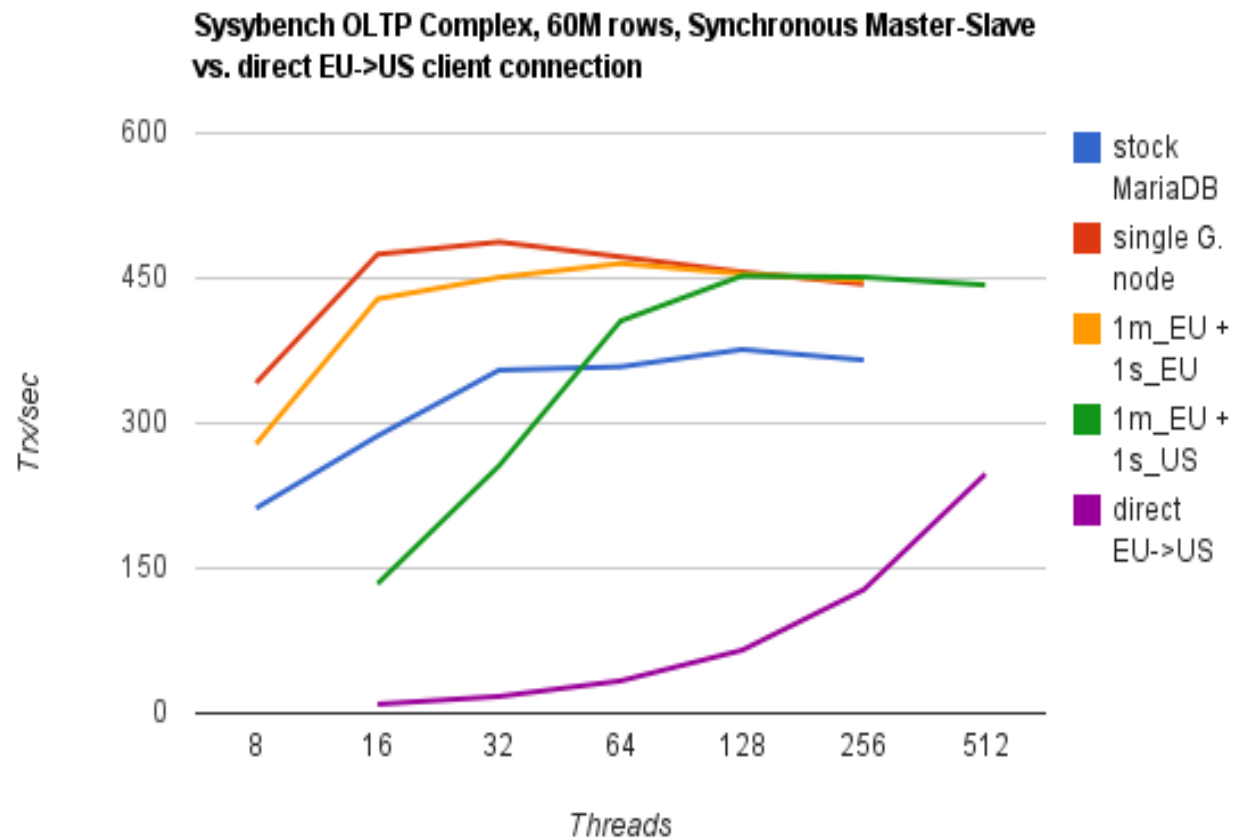
Synchronous Master-Slave WAN Latencies



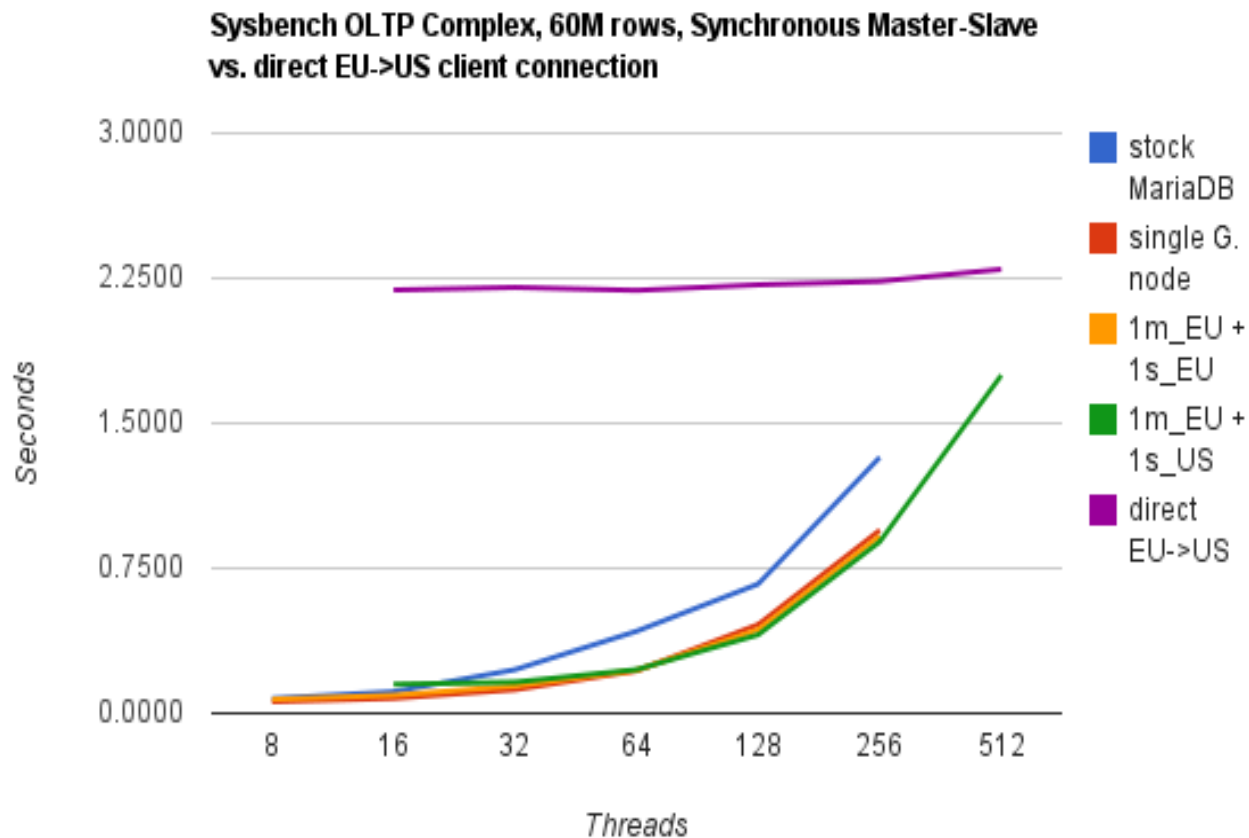
Disaster Recovery



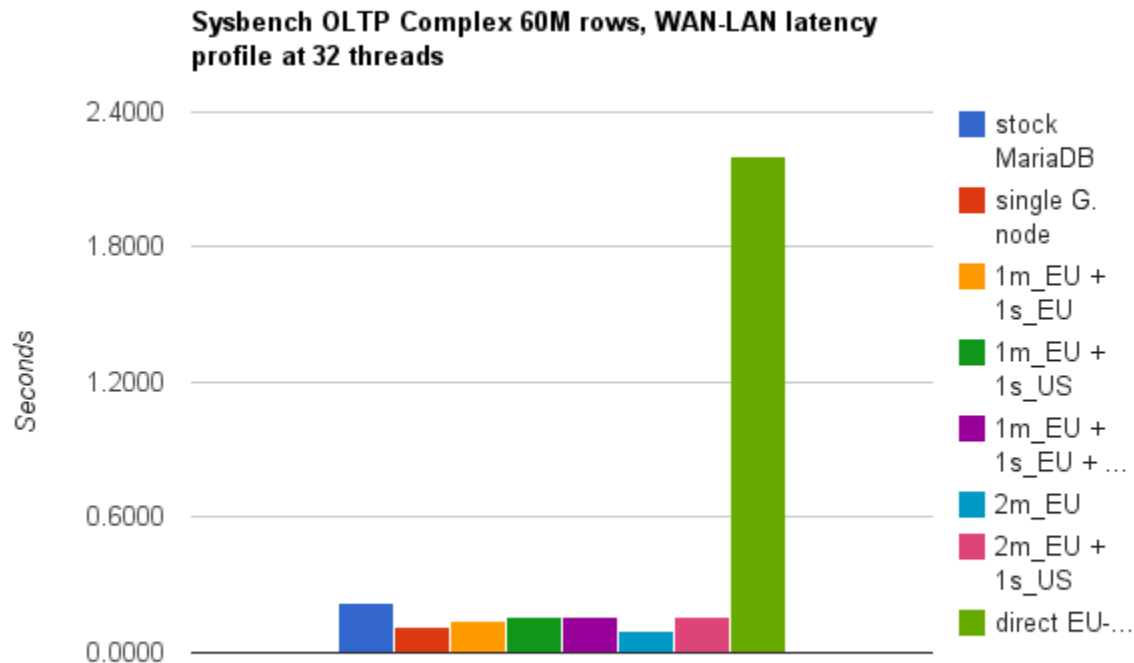
Synchronous Master-Slave vs. Direct Connections



Synchronous Master-Slave vs. Direct Connections



WAN Latencies



Installation

Installing MySQL/Galera

- Download from www.codership.com
- Distributions choices:
 - 1.Pre-built RPM or Debian package
 - 2.demo tar distribution
 - 3.Source build

Demo Distribution

- Pre-built 32/64 bit linux binaries
- Installs in one directory path
- Contains a sample database
- Good for testing/evaluation

Demo Distribution

- Install as regular user (not root)

```
$ tar xzf mysql-5.1.53-galera-0.7.6-x86_64.tgz
```

- Node startup by: mysql-galera script

- Commands: **start** | **stop** | **check**

- Specify cluster_address

- Start first node with address: `gcomm://`

- Start other nodes with `gcomm://<first-node-ip>`

```
$ mysql-galera -g gcomm:// start
```

```
$ mysql-galera -g gcomm://<other-IP> start
```

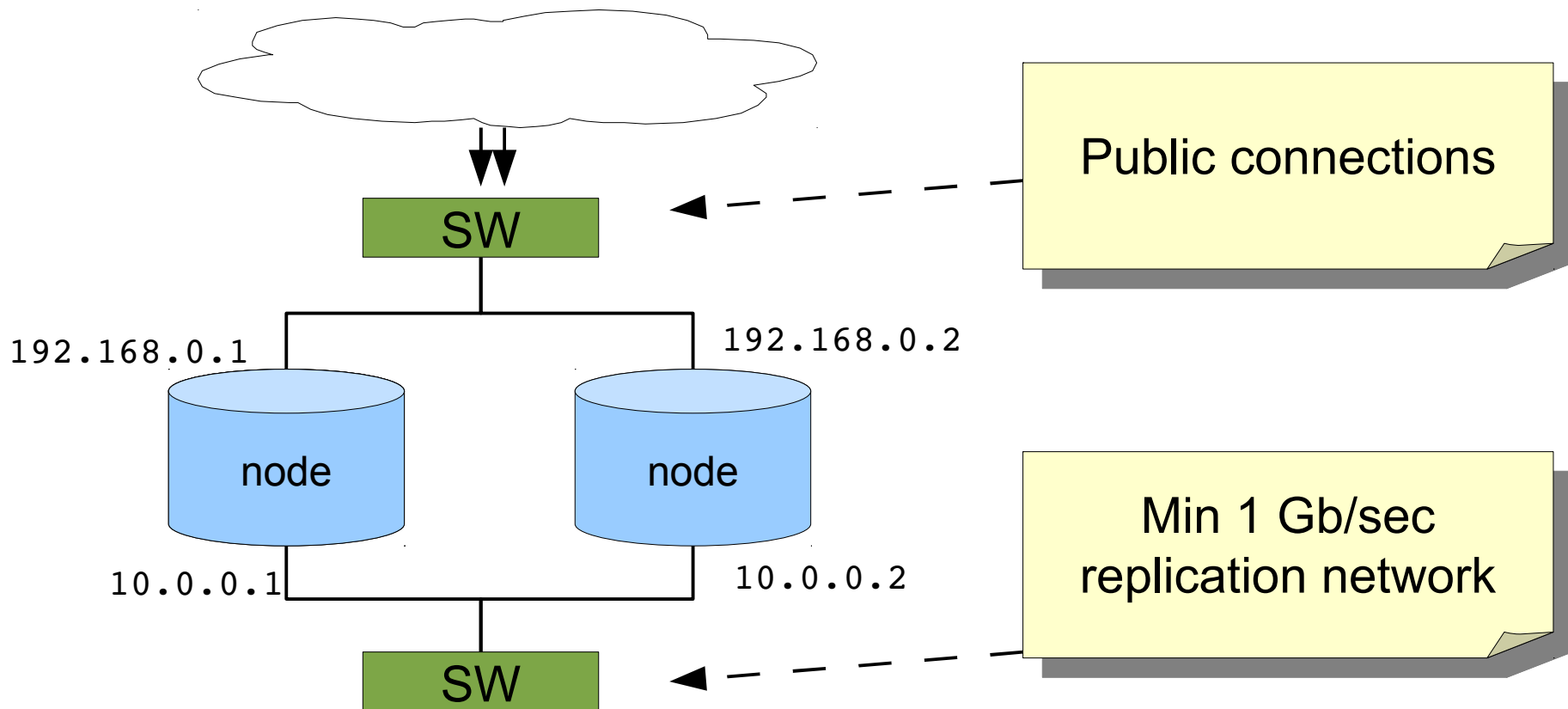
Galera in Cloud

- VPS.net
 - Nice new cloud computing solution
 - MySQL/Galera images available
- Amazon EC2
 - Extensively tested in EC2
 - Deploy .e.g. Ubuntu node and install MySQL/Galera manually
 - Pre-built image underway

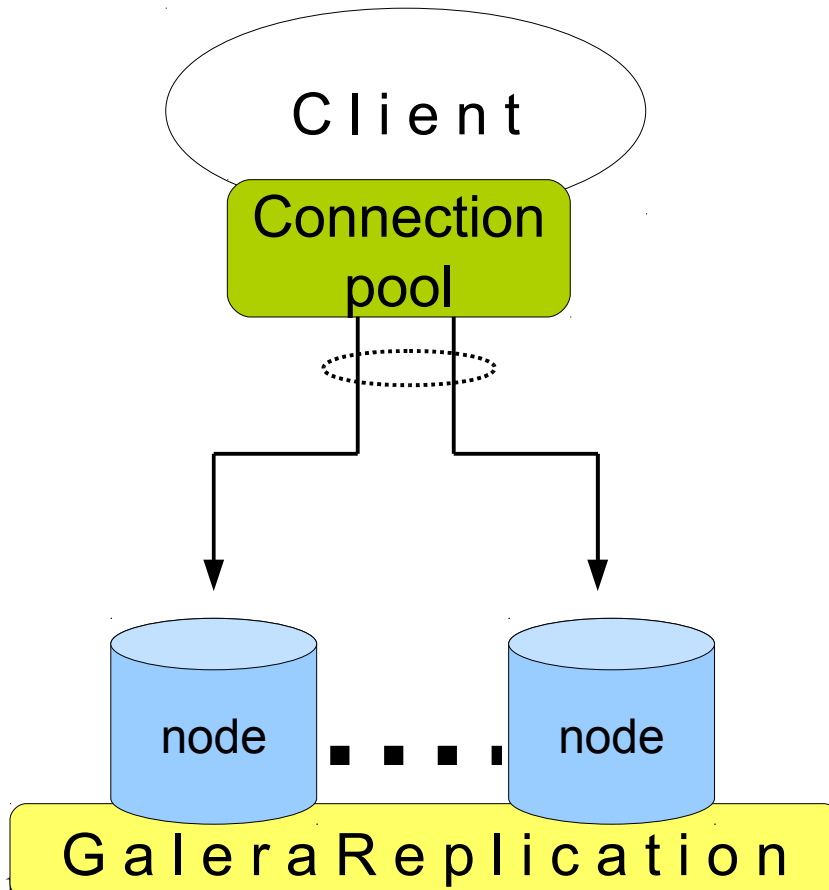
Cluster Topologies

- Use 3 or more nodes for HA
- Application load balancing gives best performance
- Use load balancer if a single connection point is needed
- Reference node can help in joining

Private Replication Network

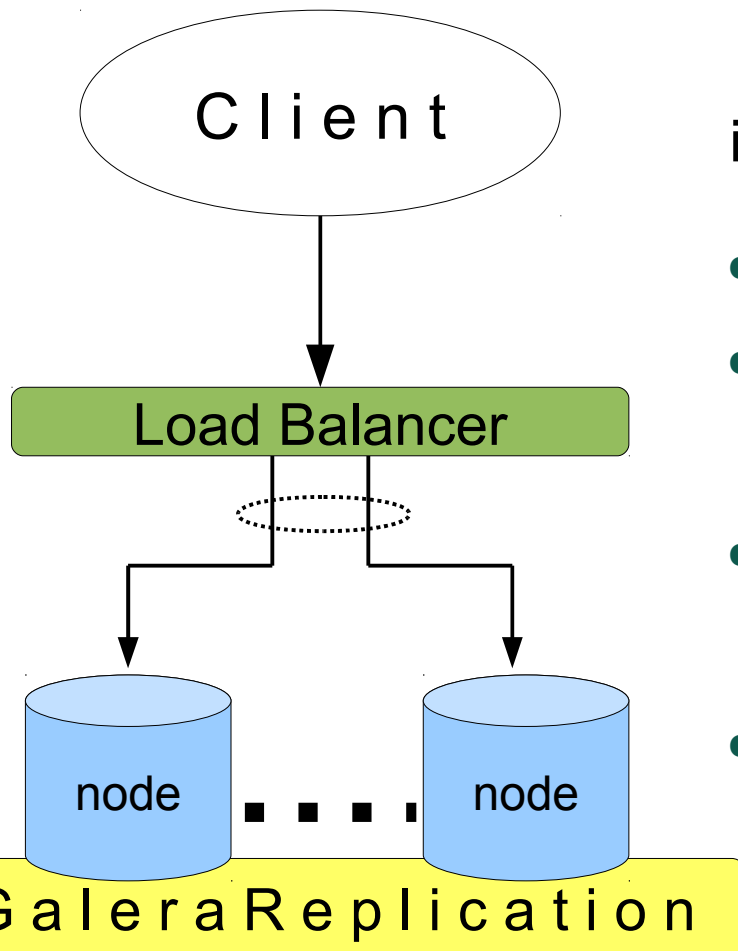


Application Load Balancing



- + Gives best performance
- Application must react to cluster changes

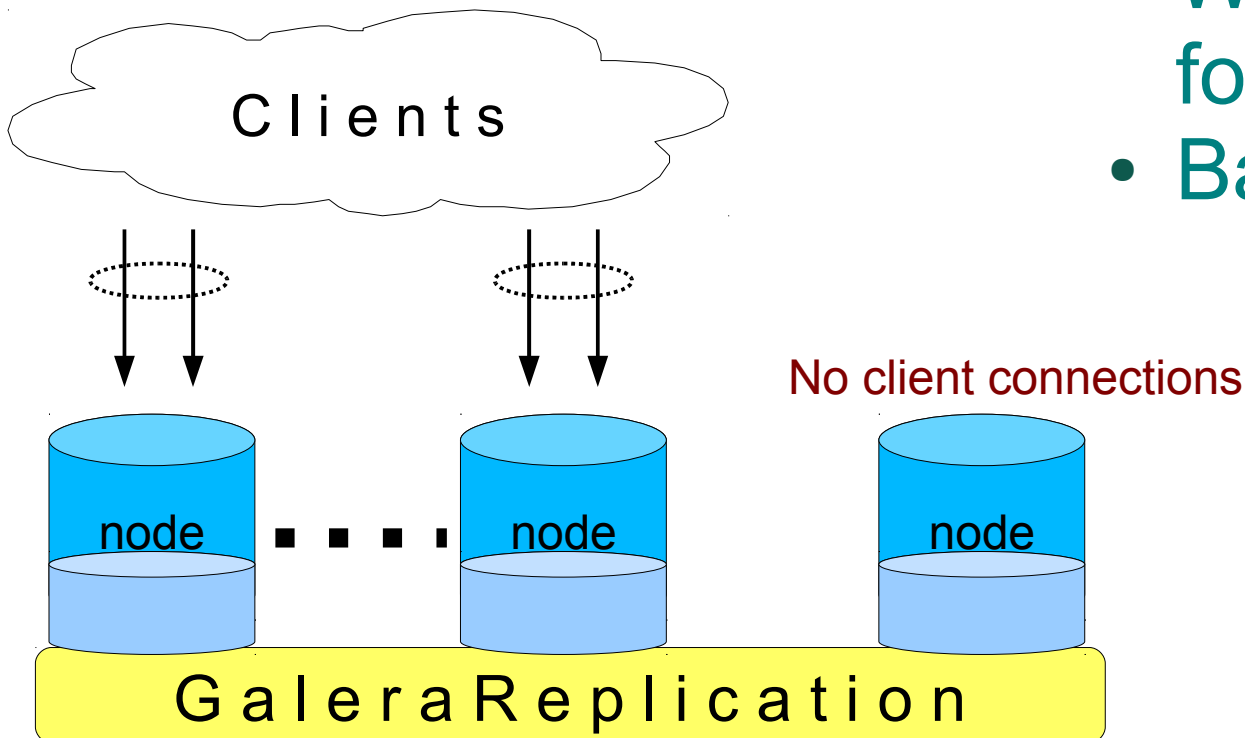
Load Balancer



in order of performance:

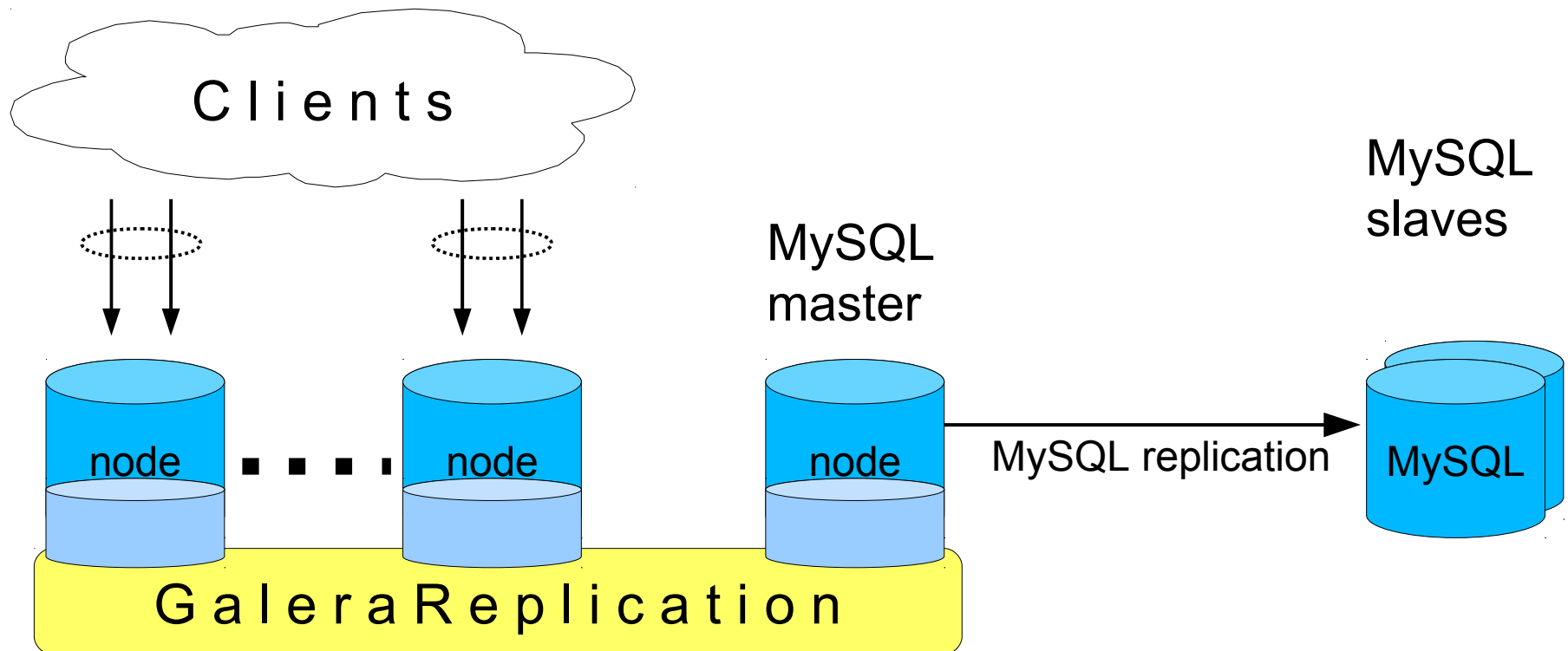
- HW balancer
- IP dispatching in kernel
e.g. LVS
- TCP/IP load balancers
.e.g. GLB, in user land
- Proxy (.e.g. MySQL Proxy)

Reference Node



- Works as donor for joining nodes
- Backups

Operating as MySQL Master



Management

wsrep Variables

```
mysql> show variables like 'wsrep%';
```

Variable_name	Value
wsrep_auto_increment_control	OFF
wsrep_certify_nonPK	OFF
wsrep_cluster_address	gcomm://?gmmcast.listen_addr=tcp://127.0.0.1:4568
wsrep_cluster_name	my_wsrep_cluster
wsrep_convert_LOCK_to_trx	OFF
wsrep_data_home_dir	/codership/data/galera-nod1/
wsrep_debug_option	
wsrep_debug	OFF
wsrep_drupal_282555_workaround	OFF
wsrep_local_cache_size	20971520
wsrep_max_ws_rows	65636
wsrep_max_ws_size	0
wsrep_node_incoming_address	10.1.198.1:3307
wsrep_node_name	nod1
wsrep_notify_cmd	
wsrep_on	ON
wsrep_provider	/codership/nod1/mysql-5.1.52/galera/lib/libmmgalera++.so
wsrep_provider_options	
wsrep_retry_autocommit	OFF
wsrep_slave_threads	1
wsrep_sst_auth	test:testpass
wsrep_sst_donor	
wsrep_sst_method	mysqldump
wsrep_sst_receive_address	AUTO
wsrep_start_position	00000000-0000-0000-0000-000000000000:-1
wsrep_ws_persistency	OFF

wsrep Variables

- **wsrep_provider**
 - Path to provider library
- **wsrep_cluster_address**
 - tells the connection point where node can join
 - 'gcomm://' for first node
 - 'gcomm://<IP address>', for joining nodes

wsrep Status

wsrep_local_state_uuid	a398eaf8-2aba-11e0-0800-432d0098b829
wsrep_last_committed	2989366
wsrep_replicated	122
wsrep_replicated_bytes	161514094
wsrep_received	0
wsrep_received_bytes	0
wsrep_local_commits	110
wsrep_local_cert_failures	0
wsrep_local_bf_aborts	0
wsrep_local_replays	0
wsrep_local_send_queue	0
wsrep_local_send_queue_avg	0.007752
wsrep_local_recv_queue	0
wsrep_local_recv_queue_avg	0.000000
wsrep_flow_control_paused	0.000000
wsrep_flow_control_sent	0
wsrep_flow_control_recv	0
wsrep_cert_deps_distance	1.750000
wsrep_apply_oooe	0.000000
wsrep_apply_ool	0.000000
wsrep_apply_window	1.000000
wsrep_local_state	4
wsrep_local_state_comment	Synced (6)
wsrep_cluster_conf_id	4
wsrep_cluster_size	2
wsrep_cluster_state_uuid	a398eaf8-2aba-11e0-0800-432d0098b829
wsrep_cluster_status	Primary
wsrep_local_index	1
wsrep_ready	ON

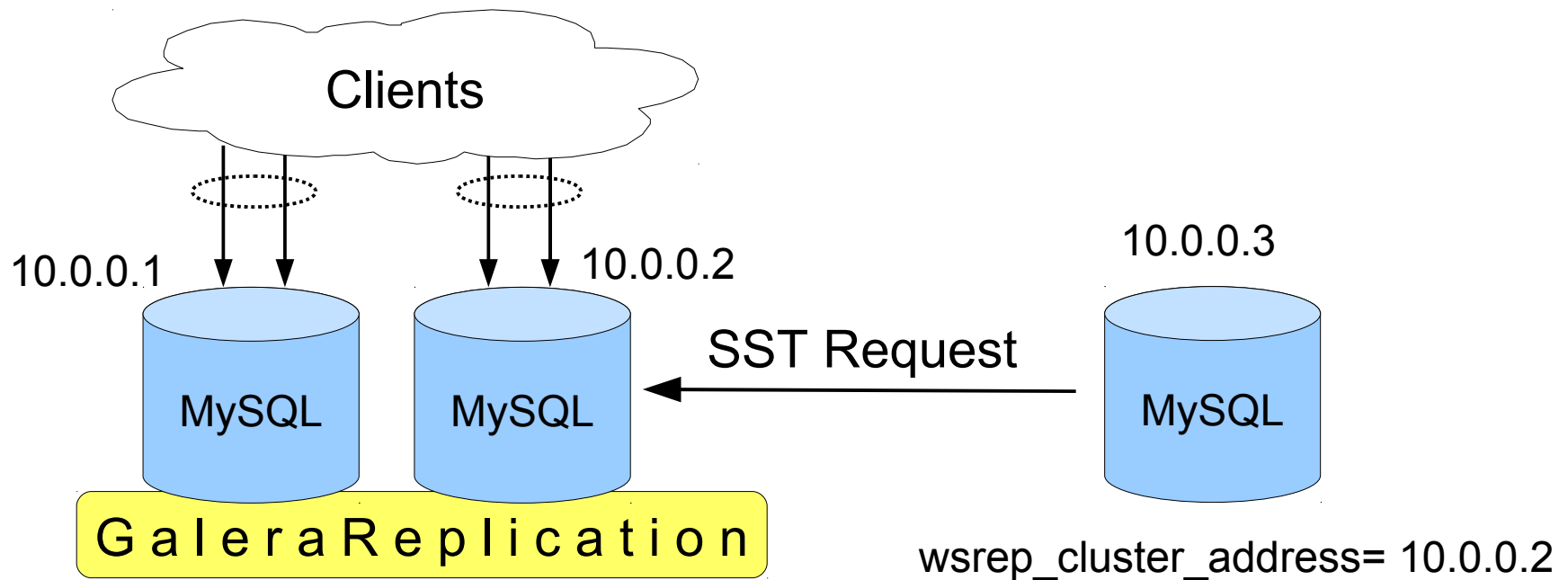
wsrep Status

- `wsrep_last_committed`
 - Tells which transaction has committed last
- `wsrep_local_cert_failures`
- `wsrep_local_bf_aborts`
 - How much cluster caused rollbacks
- `wsrep_flow_control_*`
 - How much wait for flow control

Backups

- No direct backup method in 0.7 release :(
- To get a backup:
 - Join/depart a node in a cluster
 - *Use reference node as MySQL master and fan out to a backup slave*
 - *Use .e.g. xtrabackup in reference node to get hot backup*

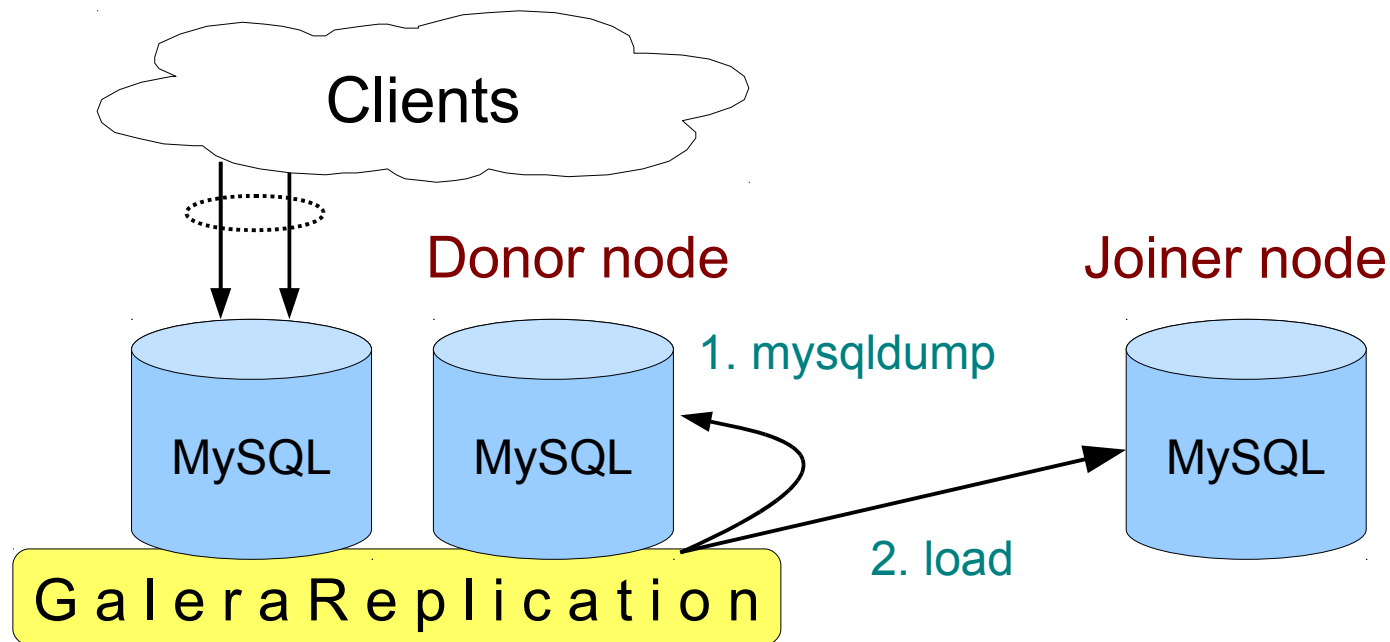
Joining New Nodes



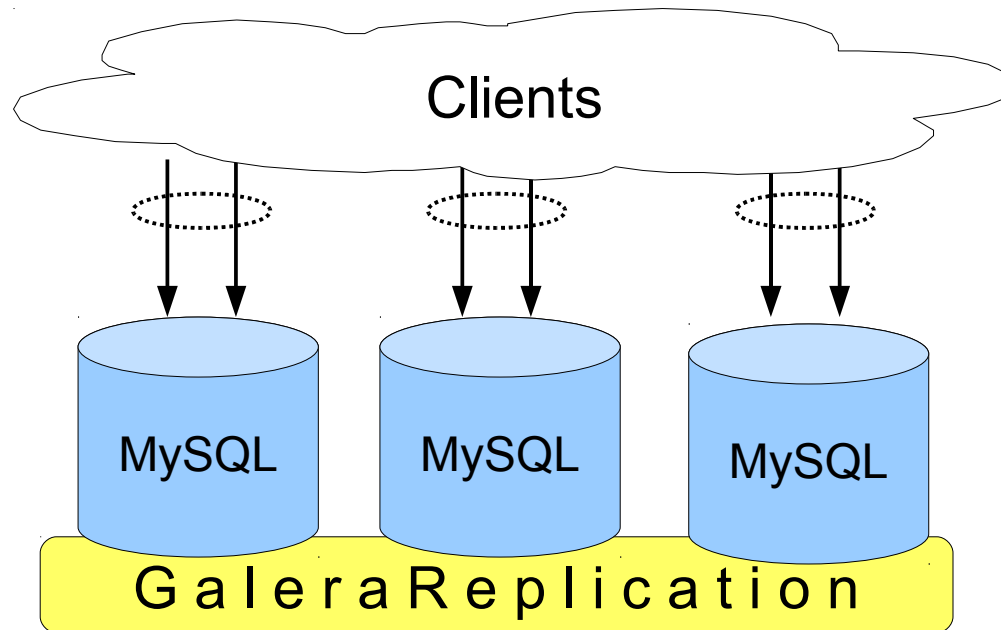
Active cluster

Joining node

Joining New Nodes



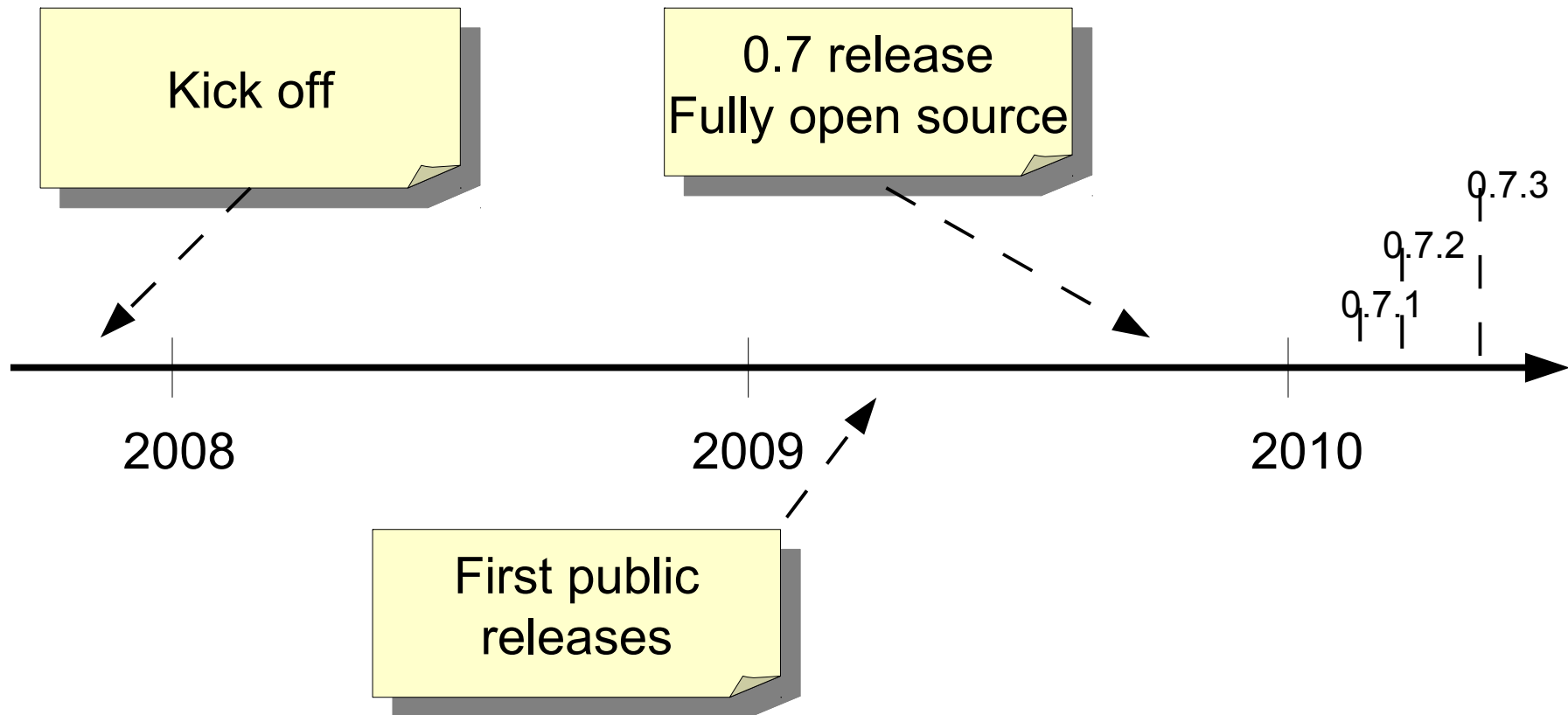
Joining New Nodes



A c t i v e c l u s t e r

Galera Project

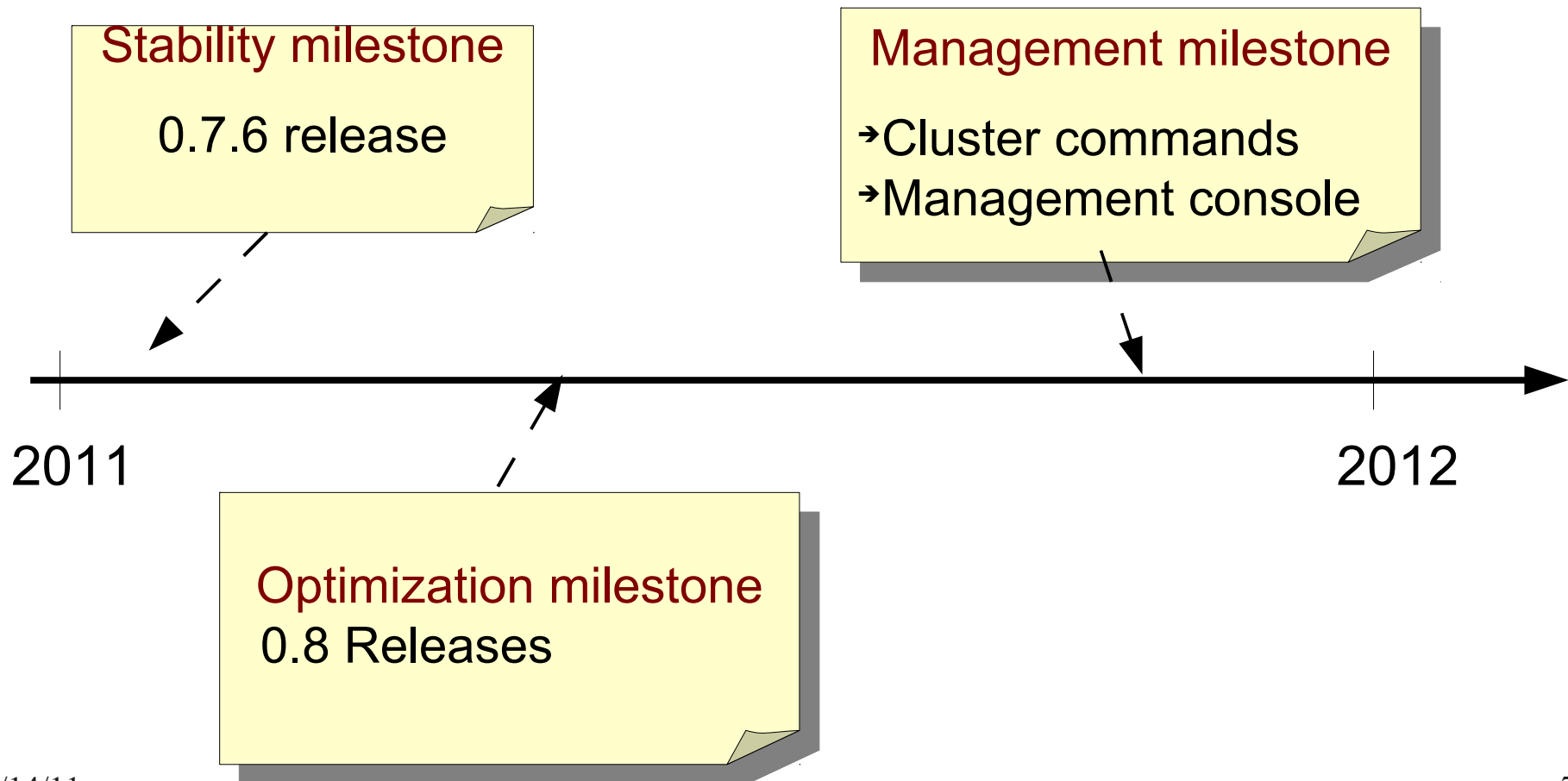
Galera Project



Releases

- Current release 0.7.6
 - Stable release
 - Production readiness
 - Open source
- 0.8 Release Mar 2011
 - Optimization Milestone
 - Rsync SST
 - UDP Multicast

Road Map



Summary

- Certification based replication turns out effective
 - High Availability
 - Transparency
 - Good scalability even with high write rates
- wsrep API is “not too hard” to implement
- Any (transactional) DBMS can leverage Galera replication

codership

- R&D consulting services
- Galera Support

- Web-site: <http://www.codership.com>
- Downloads: <https://launchpad.net/codership-mysql>
- Mailing list: codership-team@googlegroups.com