MySQL Servers Working as a Team - Replication or Galera Cluster

Open Source Data Center Conference, April 26th - 28th, Berlin

Jörg Brühe

Senior Support Engineer, FromDual GmbH

joerg.bruehe@fromdual.com

FromDual GmbH

Support

Consulting

remote-DBA

Training

About Me

- **Development distributed SQL-DBMS**
  Porting mainframe -> Unix,
  Interface to archiver tools (ADSM, NetWorker)
- **MySQL Build Team**
  Release builds incl. tests, packaging, scripts, ...
- **DBA**
  MySQL running a web platform
  (master-master-replication)
- **Support-Engineer (FromDual)**
  Support + Remote-DBA for MySQL / MariaDB / Percona
  both with and without Galera Cluster

Contents

General Remarks

- Concepts rather than details:
  ”the forest, not the trees“
- MySQL 5.6 (established GA version)
- Also valid for Percona and MariaDB

- Not applicable to ”embedded“ MySQL
- Not considered: NDB = ”MySQL Cluster“

- **MySQL Server: Architecture**

  Binlog

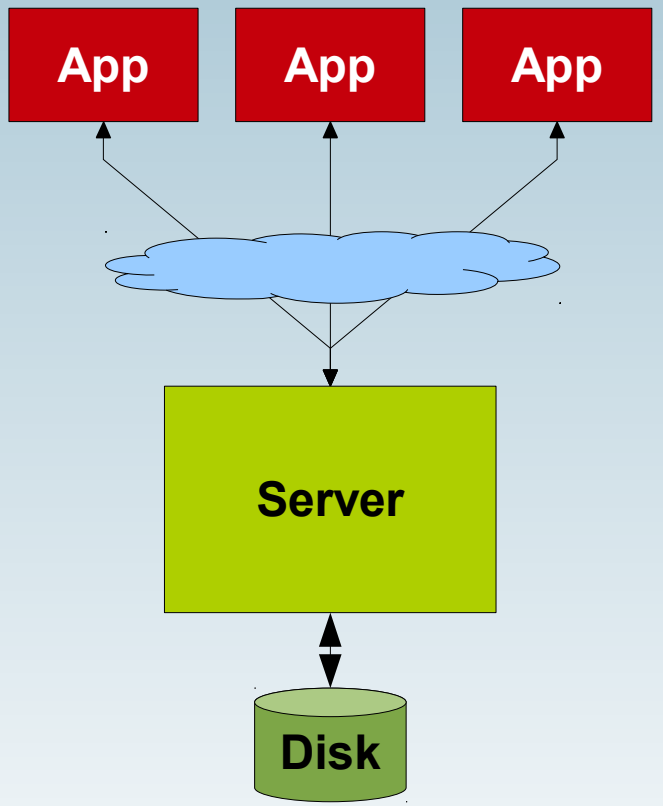
  Replication

  Galera Cluster

  Comparison

  Examples / When (not) to Choose Which

Client-Server-DBMS



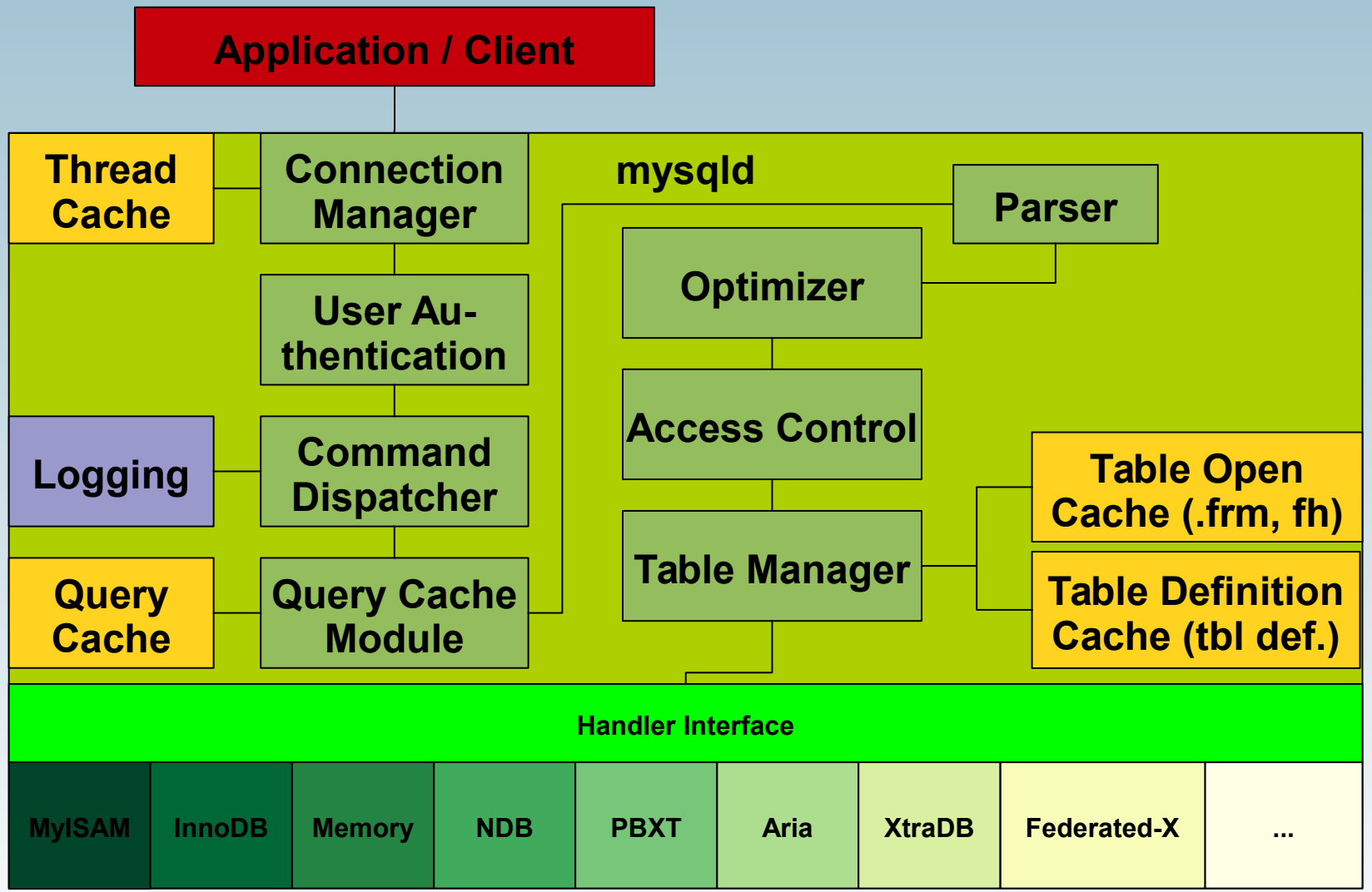
Client (application)
local or remote

Socket, LAN or internet

Server is separate process,
multi-threaded:
1 thread per user session

Disk / SSD, local or SAN

Inside the Server



MySQL Server: Architecture

➡ **Binlog**

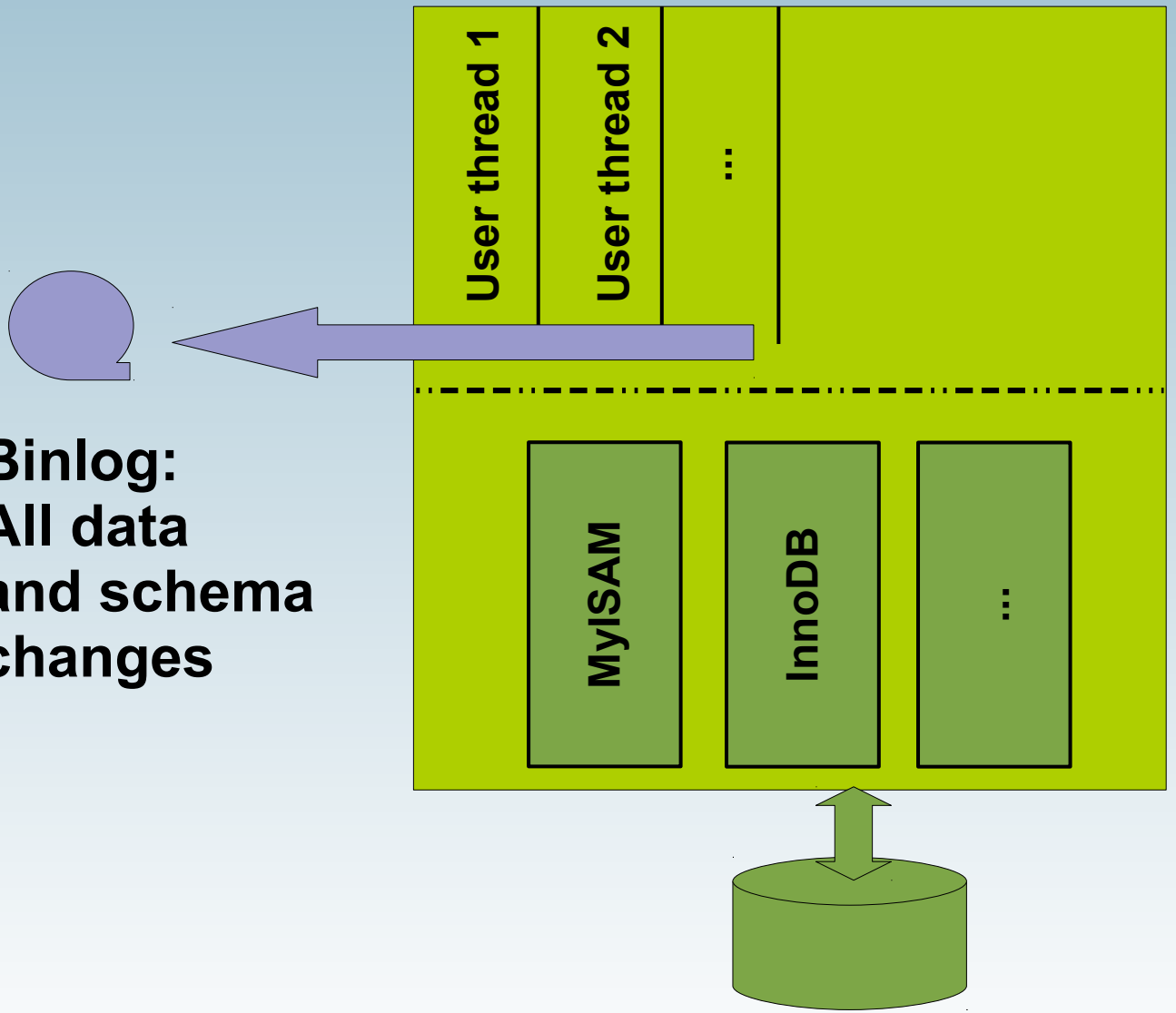
Replication

Galera Cluster

Comparison

Examples / When (not) to Choose Which

Layers + Binlog



Binlog:
All data and schema changes

SQL layer:

- Parser
- Optimizer
- Privileges
- Query Cache
- ...

Handler Interface

File layer:

- Table Handler
- InnoDB:
  - Row Access
  - Row Locks
  - Recovery
  - ...

Binlog

- All data changes executed
- All schema changes executed
- Timestamps
- Essential for Point-in-Time-Recovery ”PITR“
- Independent of table handler
- Formats ”statement“, ”row“, and ”mixed“
- Segments of configurable size
- Numbered sequentially

MySQL Server: Architecture

Binlog

➡ **Replication**

Galera Cluster

Comparison

Examples / When (not) to Choose Which

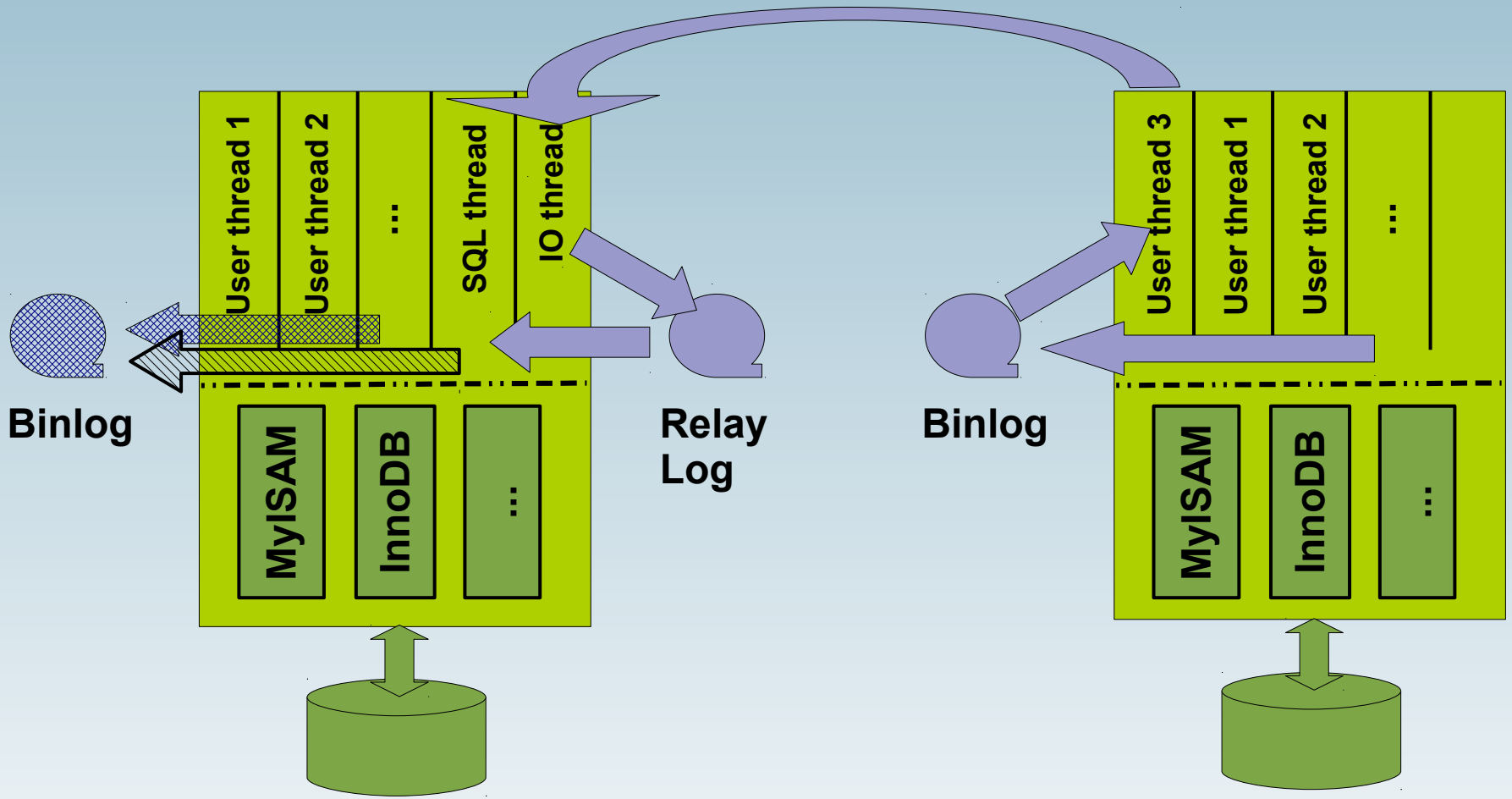
MySQL Replication

- Applications communicate with ”Master“
- ”Master“ logs all changes
- ”Slave“ has identical initial state

- Slave fetches all changes from master and applies them locally

- Replication is running asynchronous
- Slave stops replication on difference

Slave fetches binlog



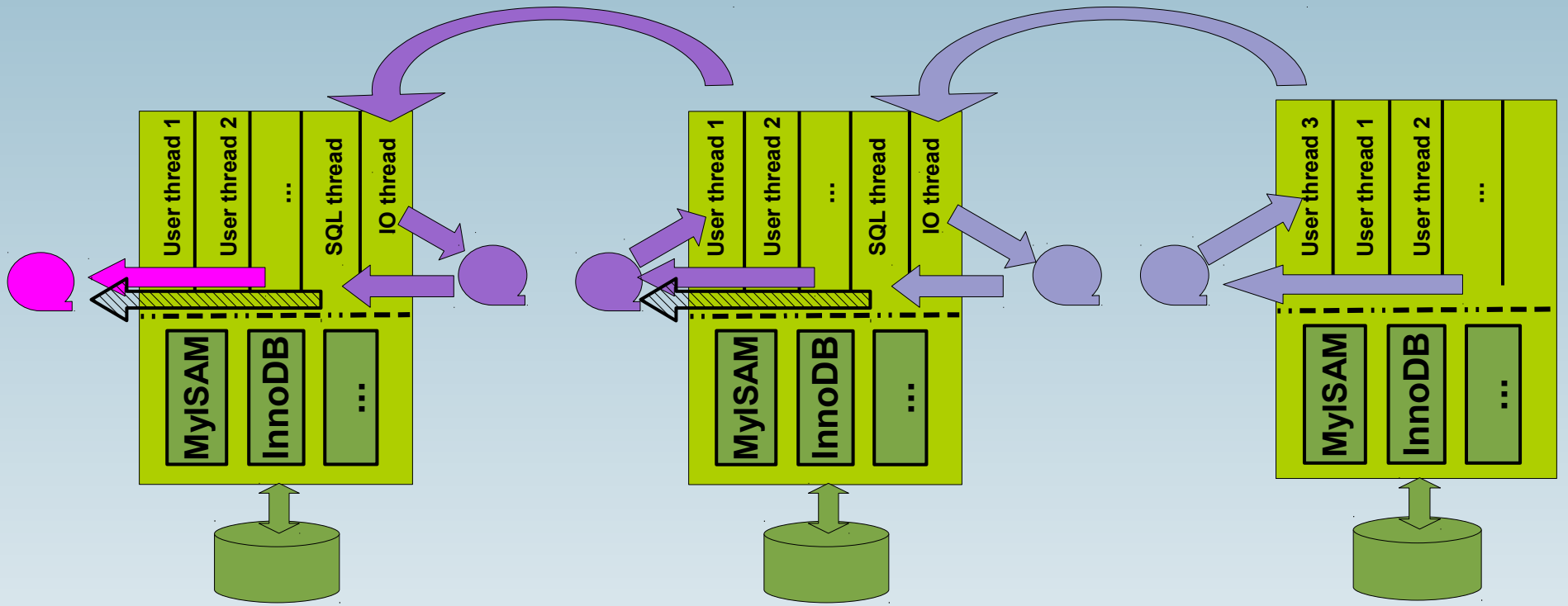
Slave:
“log-bin = FILE”, or else no binlog
“log_slave_updates = 1” for forwarding

Master:
“log-bin = FILE”, or else no binlog
(no master function)

Typical Usage

- ”High Availability“
- Geographic redundancy
- Support higher read load
  (= ”read scale-out“)
- Read-only instance(s)
  e.g. for backup or reports
- Intentional delay is possible
- Filtering (by DB or table) is possible

Replication Cascade



- Recommended: ”read-only = 1“ on slave ”log_slave_updates = 1“
- Multiple slaves per master are possible

Entries in Binlog

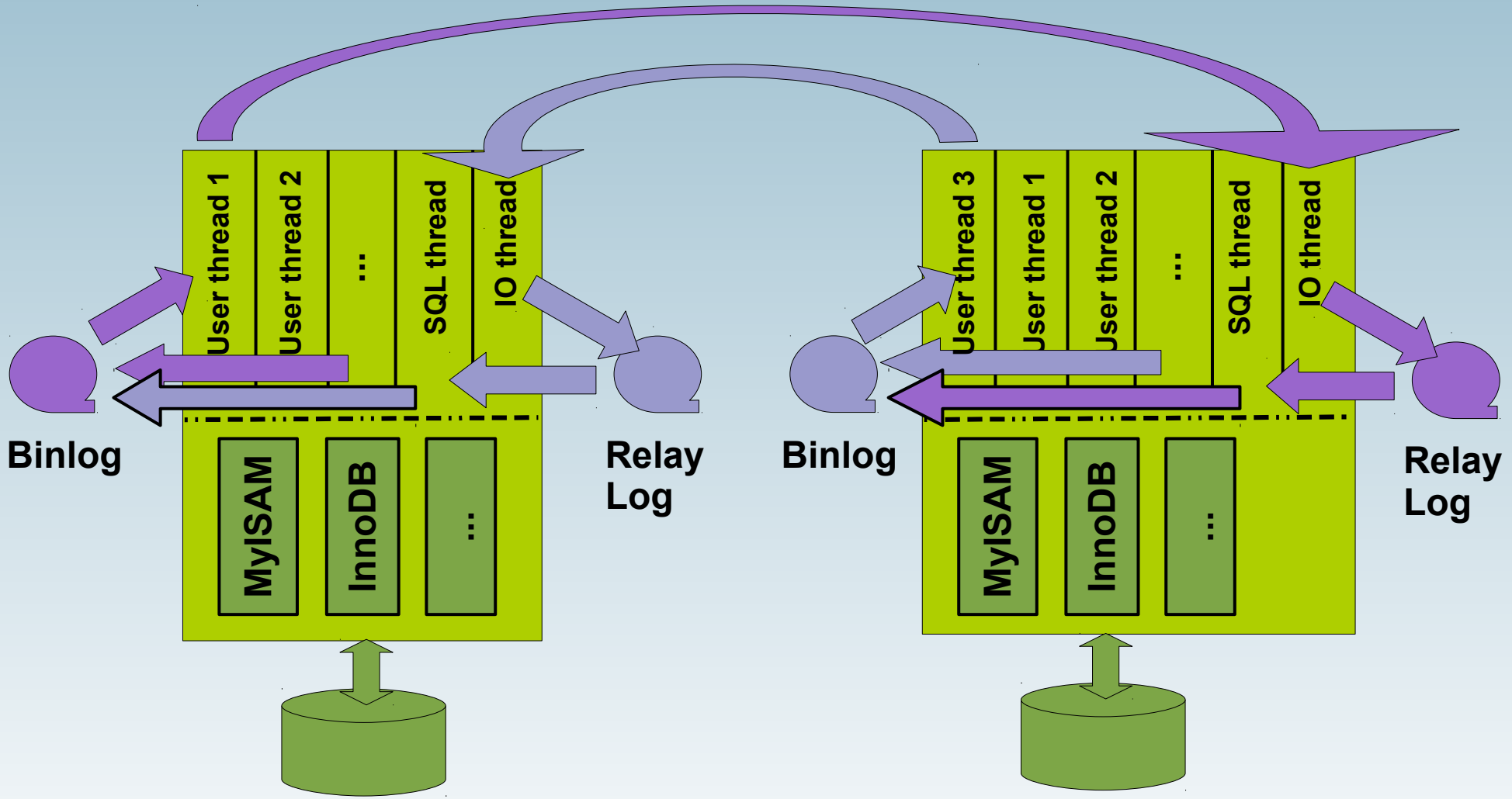
Original:

- **Identification by file name and position**
- **Replication: ”`change master to ...`“ specifying host, port, user, password, file, position**
- **See also: ”`mysqldump --master-data`“**

From MySQL 5.6 also:

- **GTID = ”Global Transaction ID“**
- **Replication: ”`change master to ...`“ specifying host, port, user, password, ”`auto_position = 1`“**

Master-Master-Replication



- **Overlapping changes are fatal!**

Notes about Replication

- Master-Master is controversial, be careful!
- Replication increases read throughput, but not/barely write throughput
- Replication causes file IO und network load
- Format ”row“ is more efficient, but less readable
- Multi-threaded replication since MySQL 5.6, multi-master (”multi-source“) coming in MySQL 5.7
- Big installation: booking.com
- Recommended: `datacharmer.blogspot.de` (Giuseppe Maxia, August 2015)

MySQL Server: Architecture

Binlog

Replication

➡ **Galera Cluster**

Comparison

Examples / When (not) to Choose Which

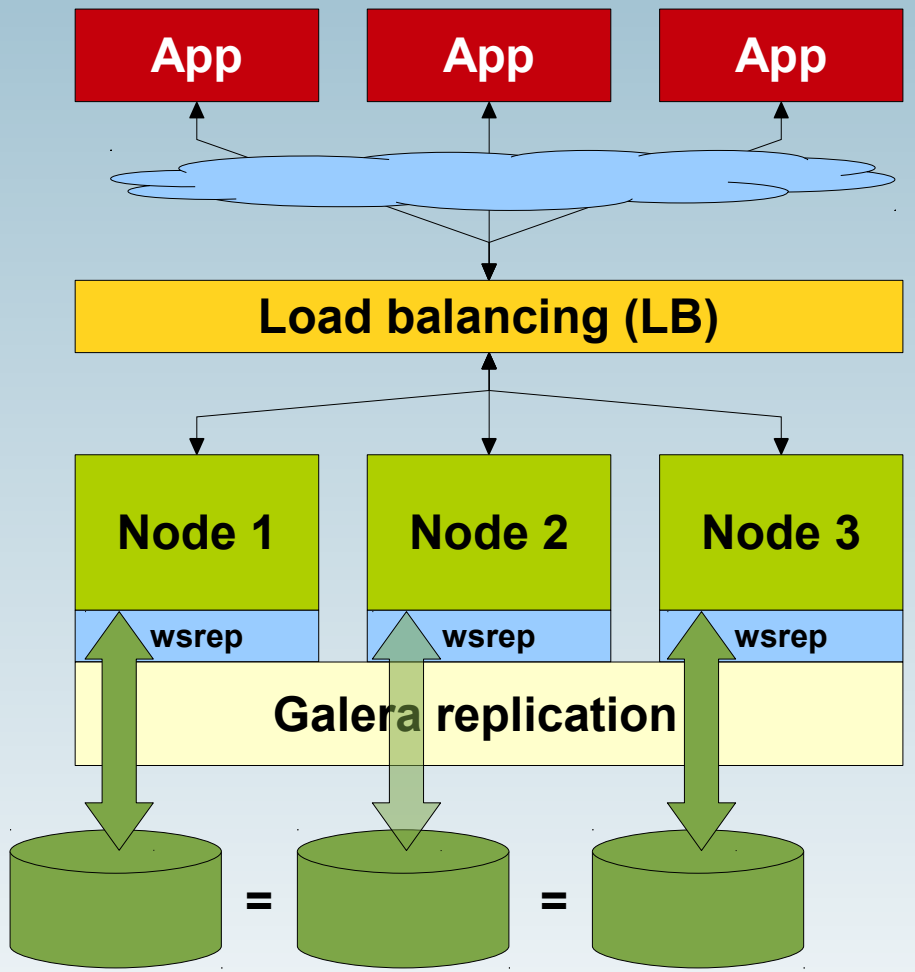
Replication Weaknesses

- Asynchronous
- Asymmetrical
- Only one write node
- Parallel writes may cause breakage
- HA needs failover after node crash
- Each node is SPOF for its slaves, breakdown requires structure change
- Dynamic changes are complicated

Better Alternative

- Synchronous transfer
- Symmetrical cluster
- Write accesses on all nodes
- Distributed conflict analysis and handling
- HA by continuity after node outage
- Dynamic entry / exit of nodes supported

Galera Cluster



“shared nothing” architecture

Inclusiding outage detection and redirection for HA

“Working Set Replication”

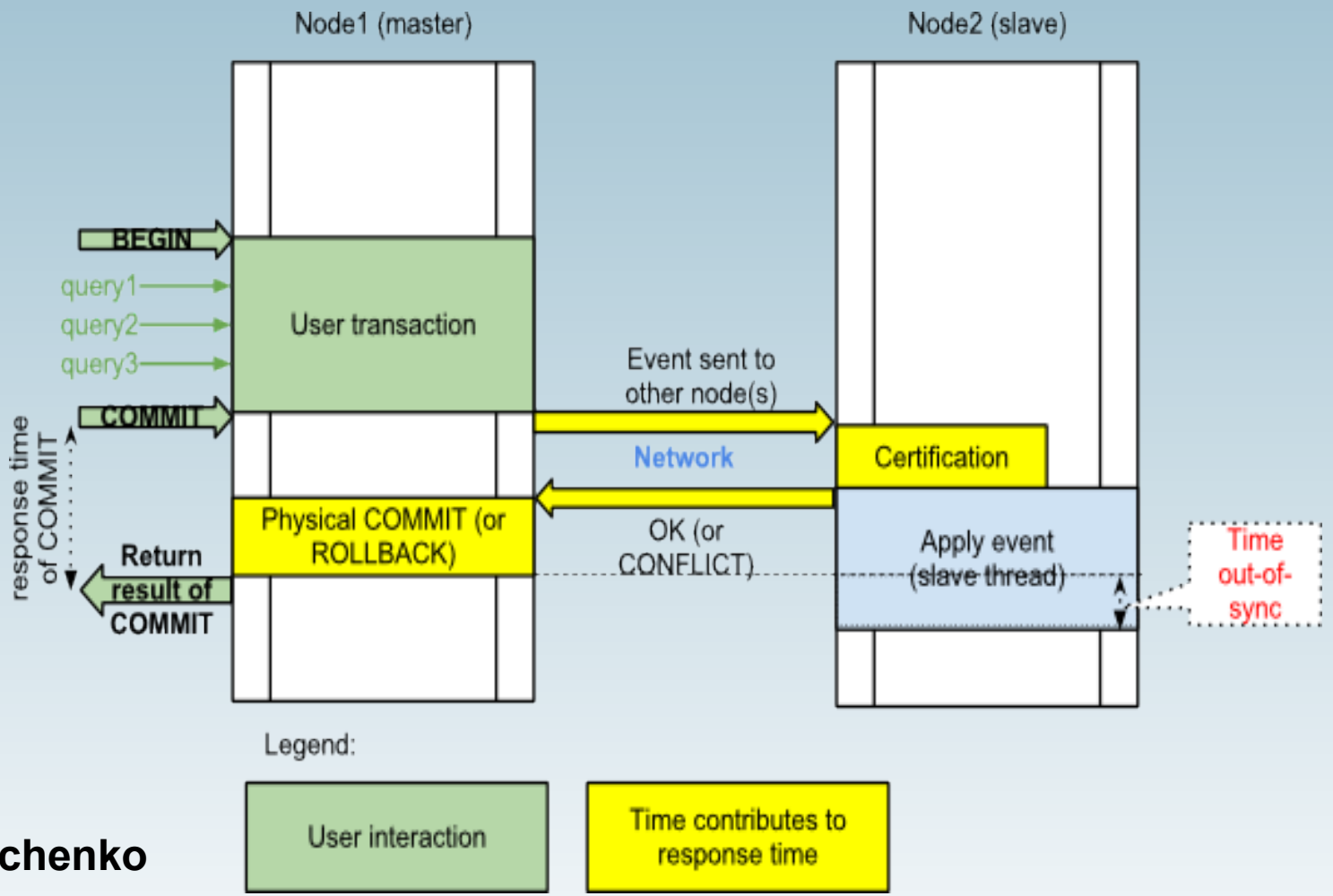
Dedicated network preferred

Locale disks, each holding all data

Galera Properties (1)

+ Based on InnoDB (due to transactions and rollback)
+ Also transfers user definitions, privileges, ...
+ Quasi-synchronous transfer on commit, check for conflicts, efficient
+ Symmetrical, HA without server failover, quorum
+ No loss of transactions
+ Brings read scale-out, also some write increase
+ Dynamical entry / exit possible, synchronisation is automated

Order of Events



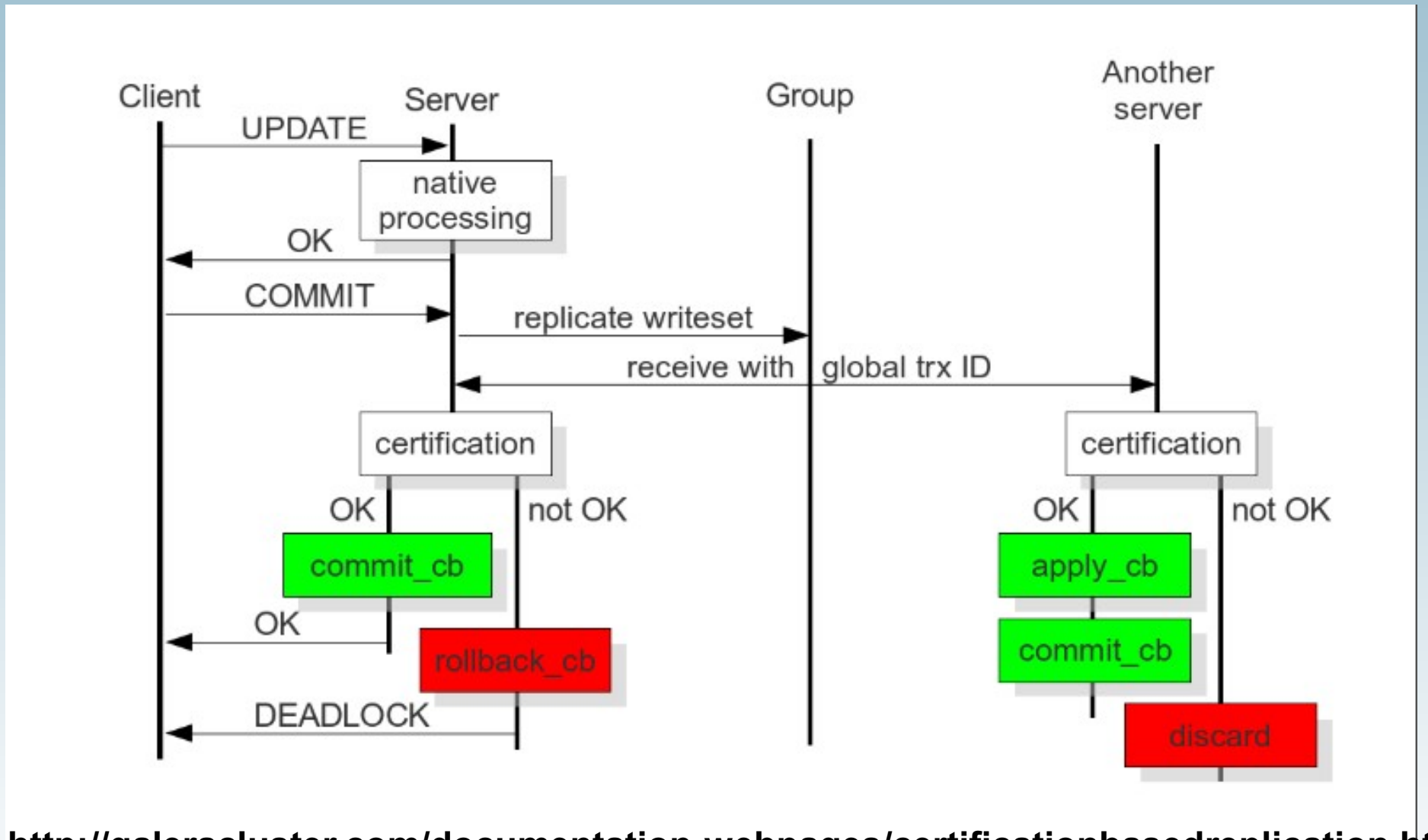
Graph by Vadim Tkachenko (Percona):

http://www.mysqlperformanceblog.com/2012/01/19/percona-xtradb-cluster-feature-2-multi-master-replication/

Galera Properties (2)

- MySQL sources need patching
  (Codership offers binaries, ditto MariaDB and Percona)
- Beware of hot spots (rows)
- Conflict detection is late, full rollback
  (Check postponed till commit)
- Minimum size is 3 nodes
- Synchronisation time for large DB
  (mysqldump -> xtrabackup or rsync)

Certifiation at Commit



http://galeracluster.com/documentation-webpages/certificationbasedreplication.html

MySQL Server: Architecture

Binlog

Replication

Galera Cluster

➡ **Comparison**

Examples / When (not) to Choose Which

Team of MySQL Servers

- Alternatives: Replication or Galera Cluster

- Redundancy of machine and storage
- HA
- Scale-out, esp. for read load
- Instances for reports, analysis, backup
- Data available locally (branch offices, ...)

Comparison (1)

Replication	Galera
Standard	Add-on product
All handlers	InnoDB only
Upwards compatible	Same versions
Minimum 2 nodes	Minimum 3 nodes
HA by failover	HA without changes
Communication:	
Hierarchical, chain	Symmetrical, parallel
Asynchronous	Quasi-synchronous
Delay is configurable	Immediate
Filtering is configurable	Complete

Comparison (2)

Replication	Galera
Read scale-Out	Read scale-Out
Write unchanged	Write increased
1 Master:	
1* write	1* write
Local conflict :	
Error on statement	Error on statement
n Master:	
n* write	n* write
Distributed conflict:	
Replication breakdown	Rollback on commit

Comparison (3)

Replication	Galera
Short interruption:	
Replication resumes	IST (incremental transfer)
Long interruption:	
Replication resumes	SST (full transfer)
Structure change:	
Manual / separate Tool	Automatic / dynamic
Initial setup:	
Snapshot, master remains available	Full transfer, donor may be blocked

CAP Theorem

”For a distributed computer system,
it is impossible to simultaneously provide
all three of the following guarantees:

- C = Consistency (identical data throughout)
- A = Availability (system is operational)
- P = Partition Tolerance (network outage)“

Eric Brewer, 1998 ff.

https://en.wikipedia.org/wiki/CAP_theorem

Prospect: MySQL 5.7

- **MySQL 5.7 is GA (5.7.9, 2015-Oct-21)**
- **Replication like in MySQL 5.6, added: multi-source replication (one slave reading from several masters)**
- **Codership is working on adding Galera Cluster to MySQL 5.7**
- **Oracle is working on ”Group replication“, currently available as ”labs release“ (= ”not fit for production“)**

MySQL Server: Architecture

Binlog

Replication

Galera Cluster

Comparison

➡ **Examples / When (not) to Choose Which**

Communication Outage (1)

Galera Cluster:

- Isolated node has no quorum
  => will not serve applications
- Quorum is at risk!
- Active nodes write ”gcache“ to files, storage period?
- Switch to SST threatens

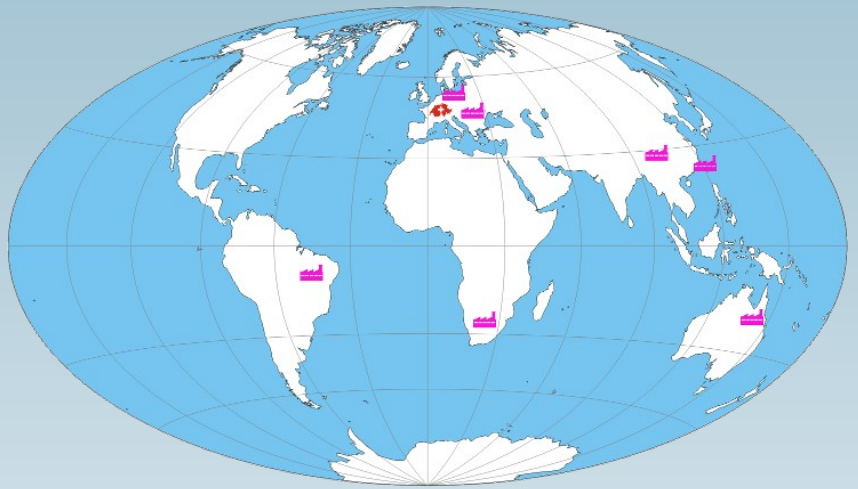
Communication Outage (2)

Replication:

- Master writes log segments to files
- IO-Thread asks to read from binlog position / GTID, retries periodical until successful
- Avoid ”purge log“!

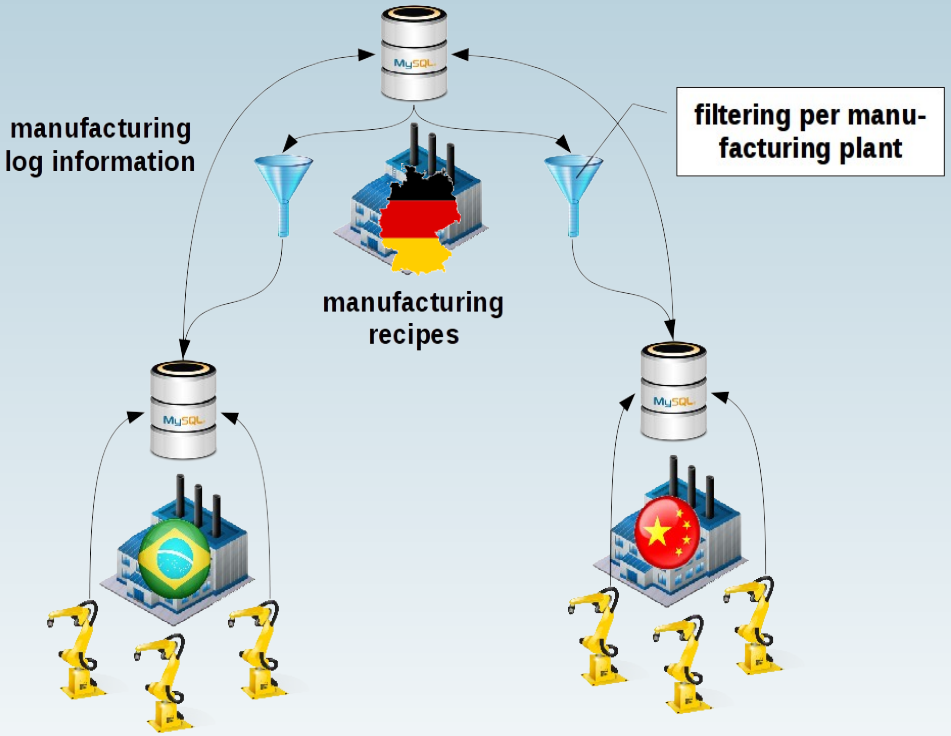
Replication is more tolerant than Galera Cluster !

Global Production



Requirement:
Head office (D) and factories (BR, CN, ...) with selective transfer

Solution: Replication with filtering



Parallel Writes + Conflict

Galera:

- Retry of autocommit statements configurable
- Transaction conflict causes rollback
  => Application repeats complete transaction

Replication:

- Slave detects conflict, no contact to application
  => Replication stops

Replication needs admin action on conflict !

Hot Spot

- Replication: frequent aborts
- Galera: frequent rollbacks

=> Agree on a single write node !

High Availability

Replication:

- Failover manual (reaction time) or automated (correct?)
- Slave lag, selection of new master

Galera:

- Symmetrical, no change of roles
- Virtually synchronous replication (no lag)

=> Advantage Galera

Q & A

Questions ?

Discussion?

- FromDual provides neutral and independent:
  - Consulting
  - Remote-DBA
  - Support for MySQL, Galera, Percona Server and MariaDB
  - Training

www.fromdual.com/presentations