facebook

facebook

Binlog Server at Facebook

Santosh Banda
Teng Li
Database Engineering Team, Facebook, Inc.

Agenda

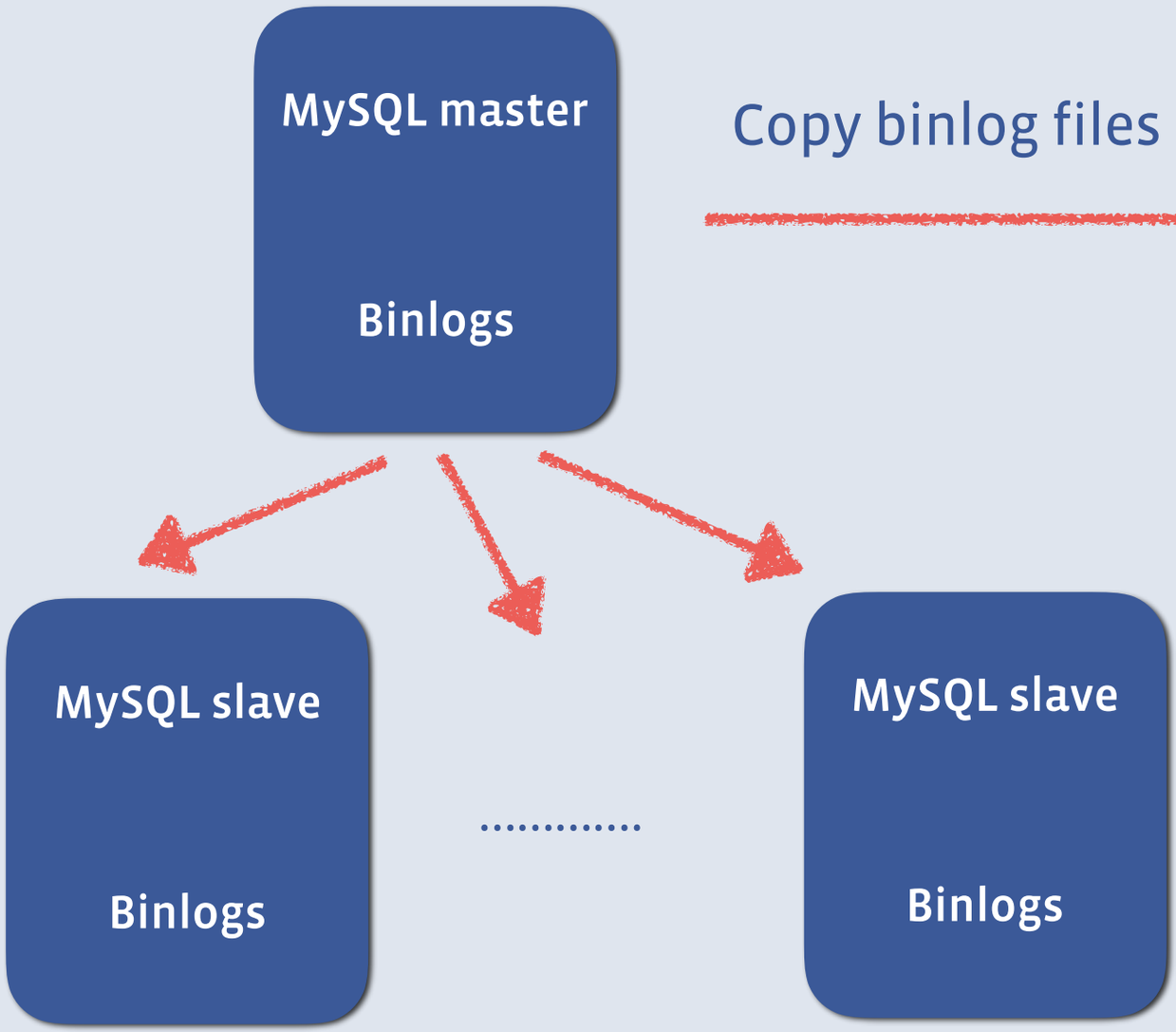
1 Motivation

2 Use cases

3 Design

4 Operational Commands

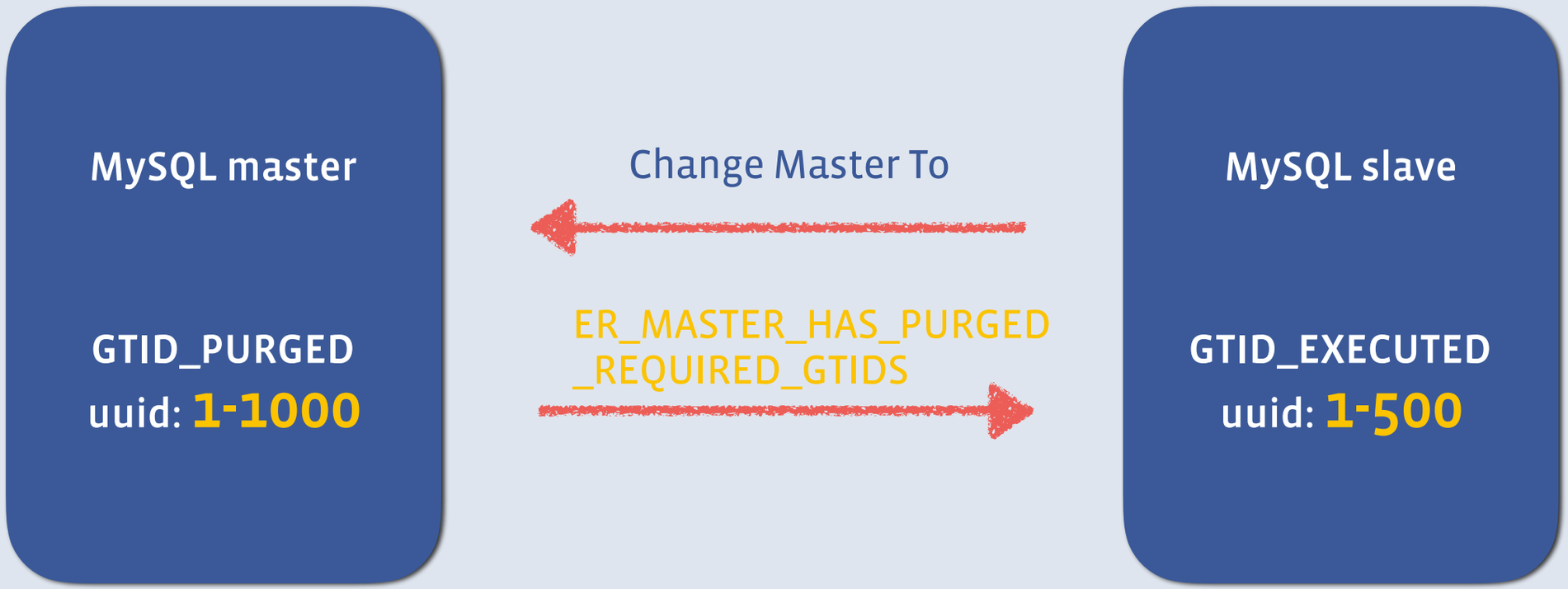
Binlog Storage at Facebook



- Shared by replica-set
- Always copied from current master
- GTID makes replay safer
- Retention time: Weeks

* Binlog retention time: Hours

Replication Catchup



Binlog retention time: Hours

Binlog Replay

Binlog retention time: Hours

Binlog Server

HDFS

Binlog Backups

Retention time: Weeks

Binlog Server

Change Master To / MySQLBinlog

Binlog data stream

Automation tools/ MySQL slave

Let's fetch old binary logs

Serves Binlogs Using MySQL Protocol

Motivation

- Unified solution for binlog retrieve and replay
- Reduce binlog partition size on MySQL machines

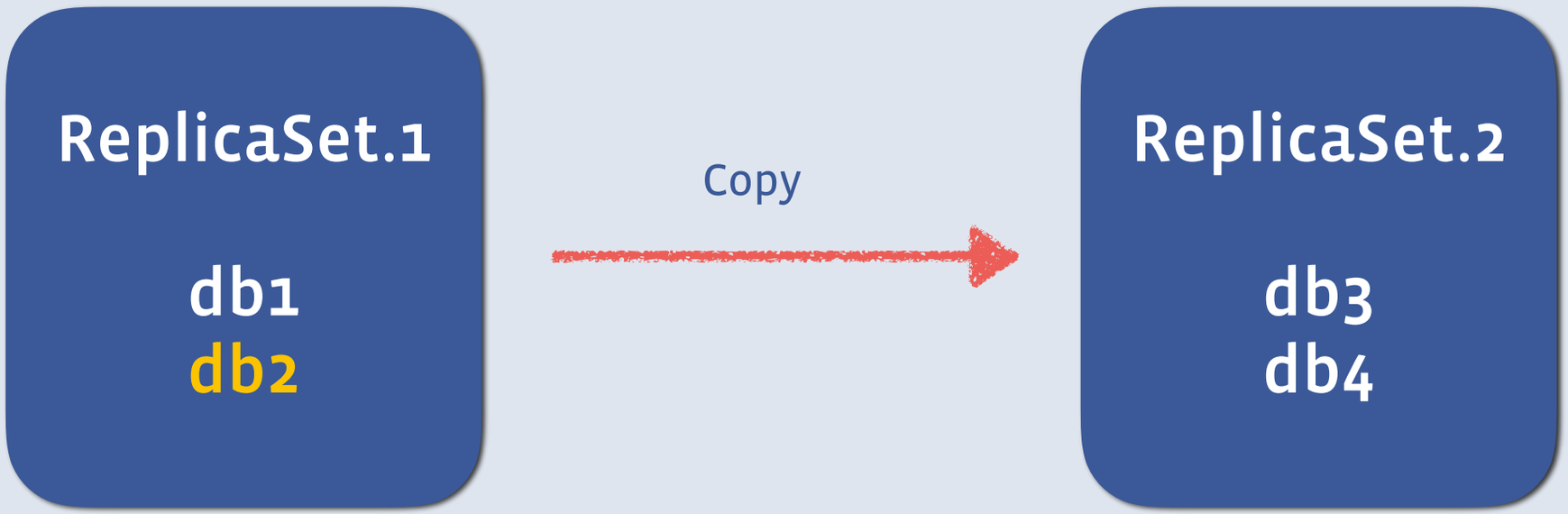
Facebook Vs MaxScale

	Facebook	MaxScale
Binlog proxy (Intermediate replica)	Yes	Yes
Easy Failover	Yes	Yes
GTID support	**Yes**	**No**
Pluggable storage systems	**Yes**	**No**
Open Source	**No**	**Yes**

Use Cases

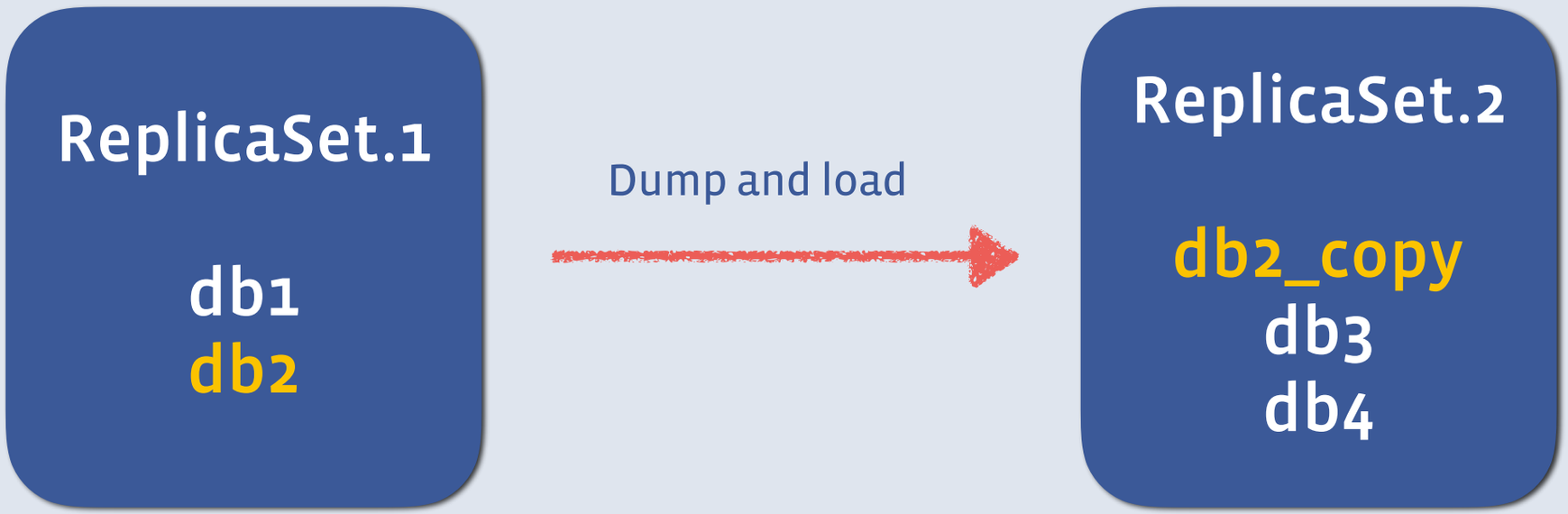
Online Shard Migration

- Moves shard across replica sets



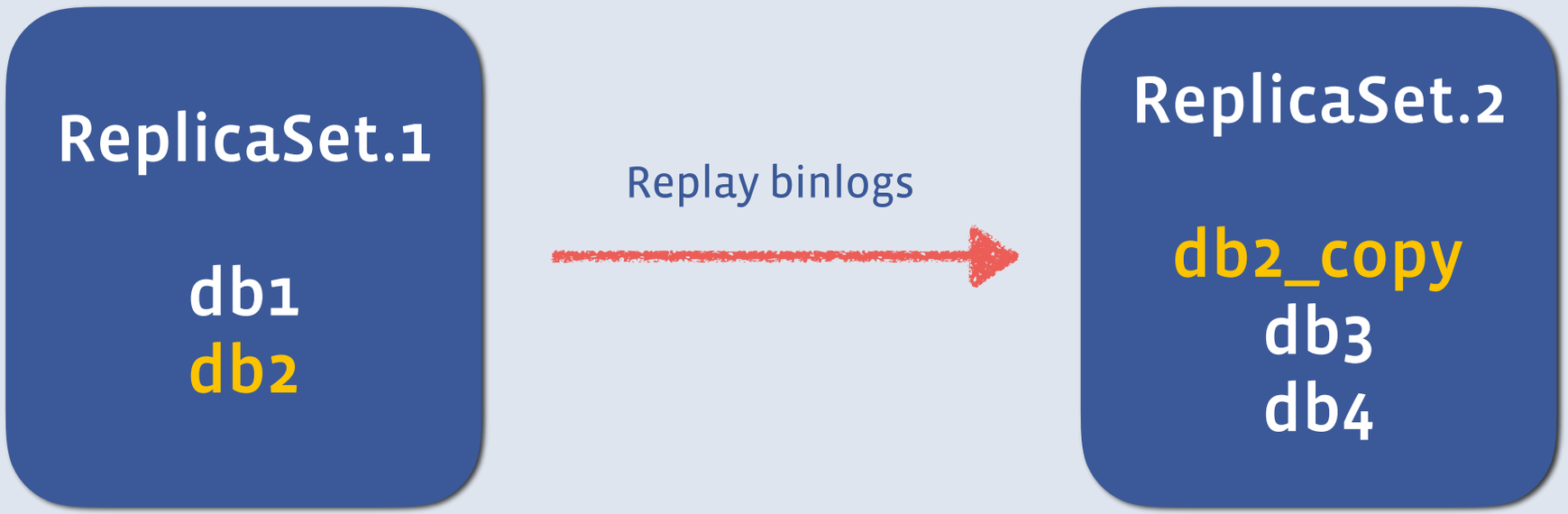
Online Shard Migration

- Copies the database using MysqlDump



Online Shard Migration

- Replay the binlogs using mysqlbinlog from ReplicaSet.1

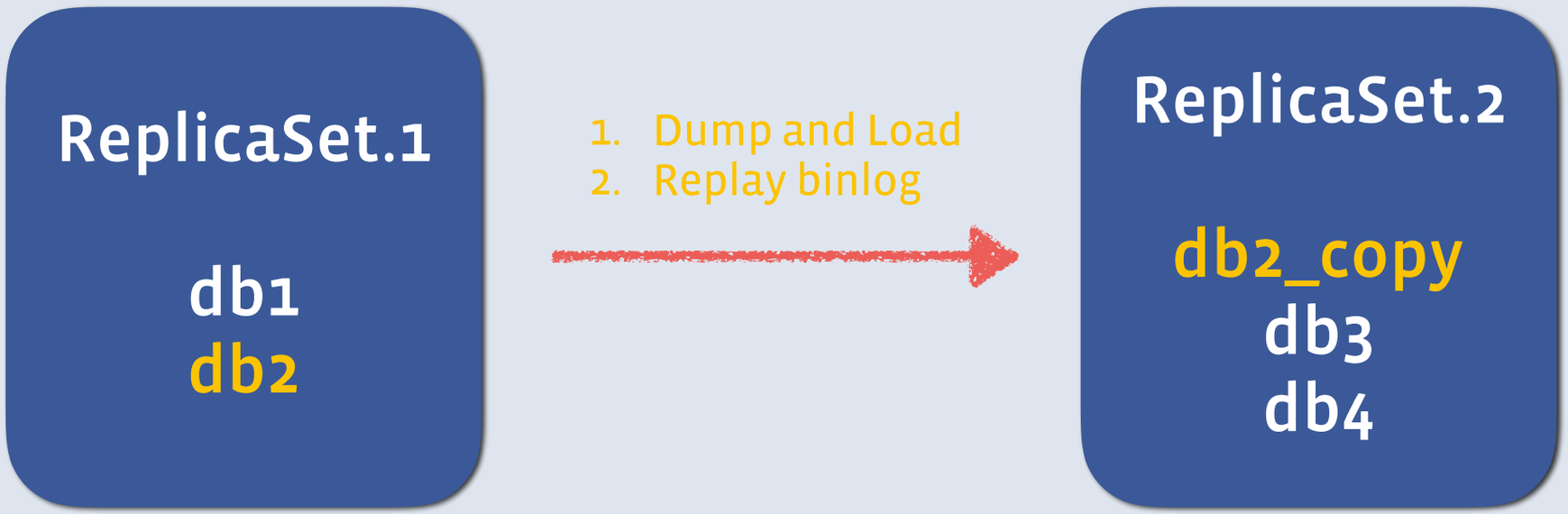


Online Shard Migration

- Copy time is greater than local binlog retention time
- Retry

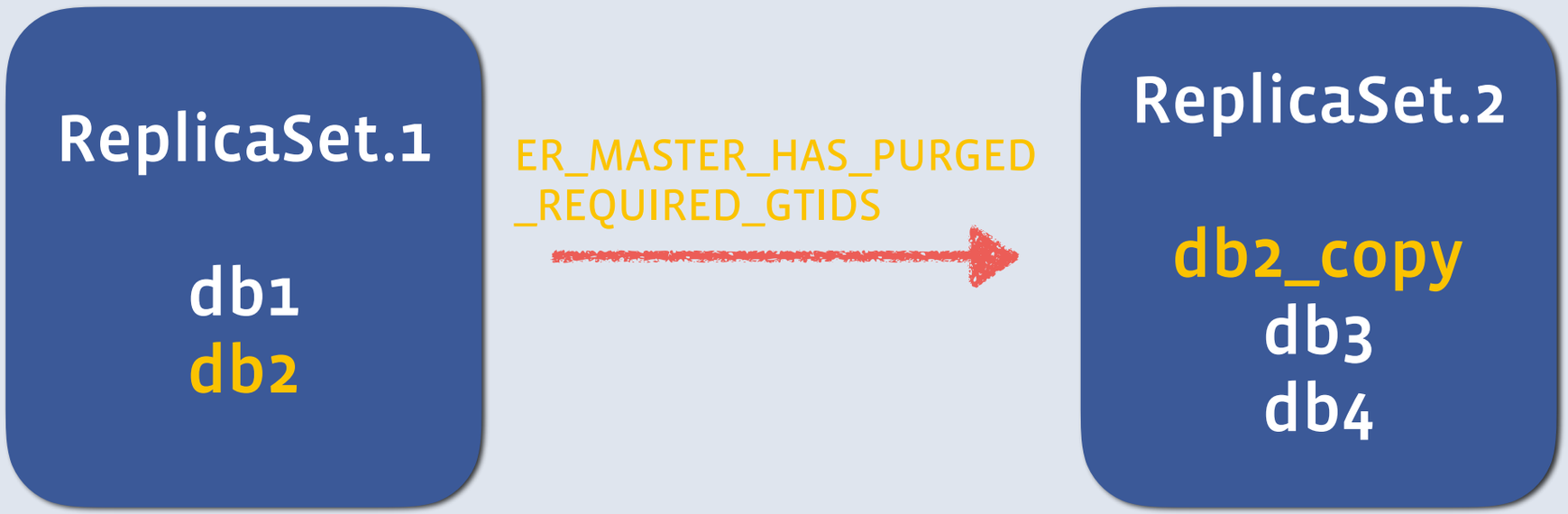
Online Shard Migration

- Retry OLM



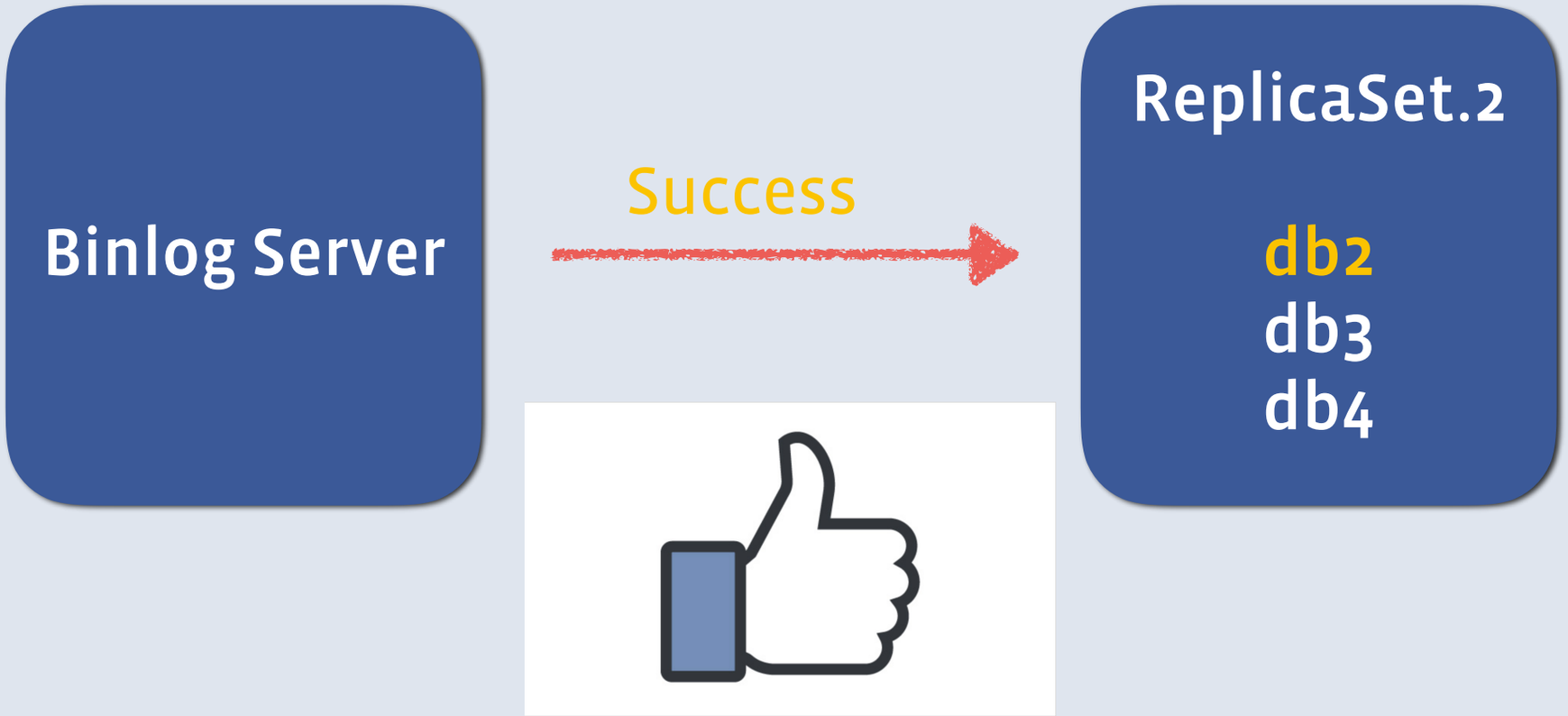
Online Shard Migration

- Retry... Retry...
- Failure !! Reach on-call



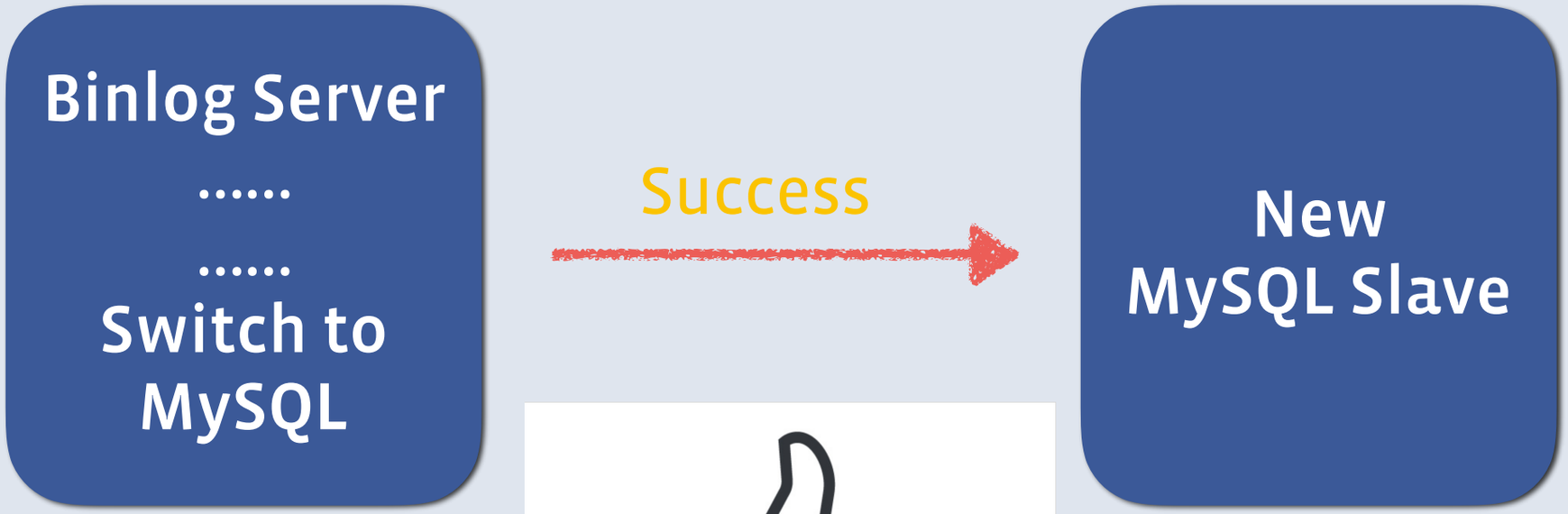
Online Shard Migration

- Replay using binlog server.
- Copy time doesn't affect migration



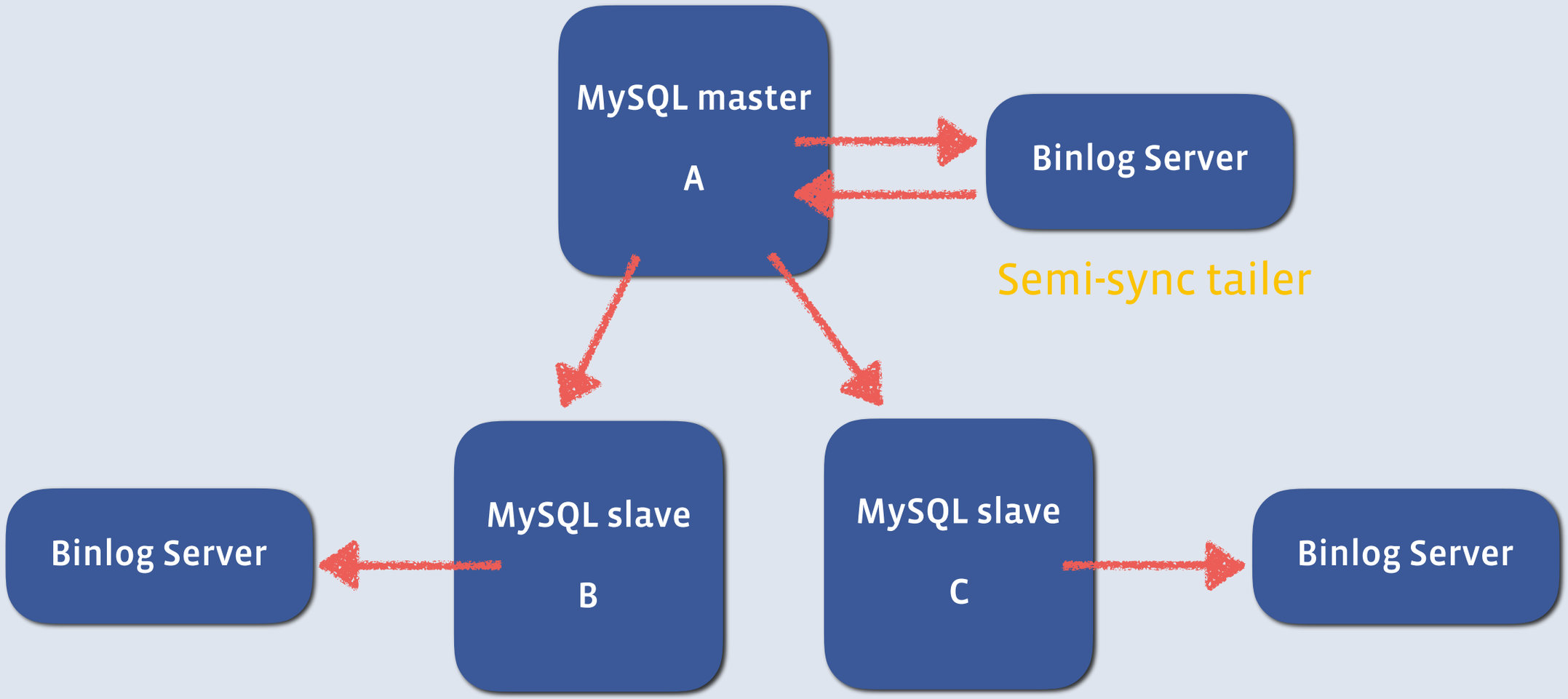
Creating New Replicas

- Replay using Binlog Server. Switch back to actual MySQL master
- No retries !!



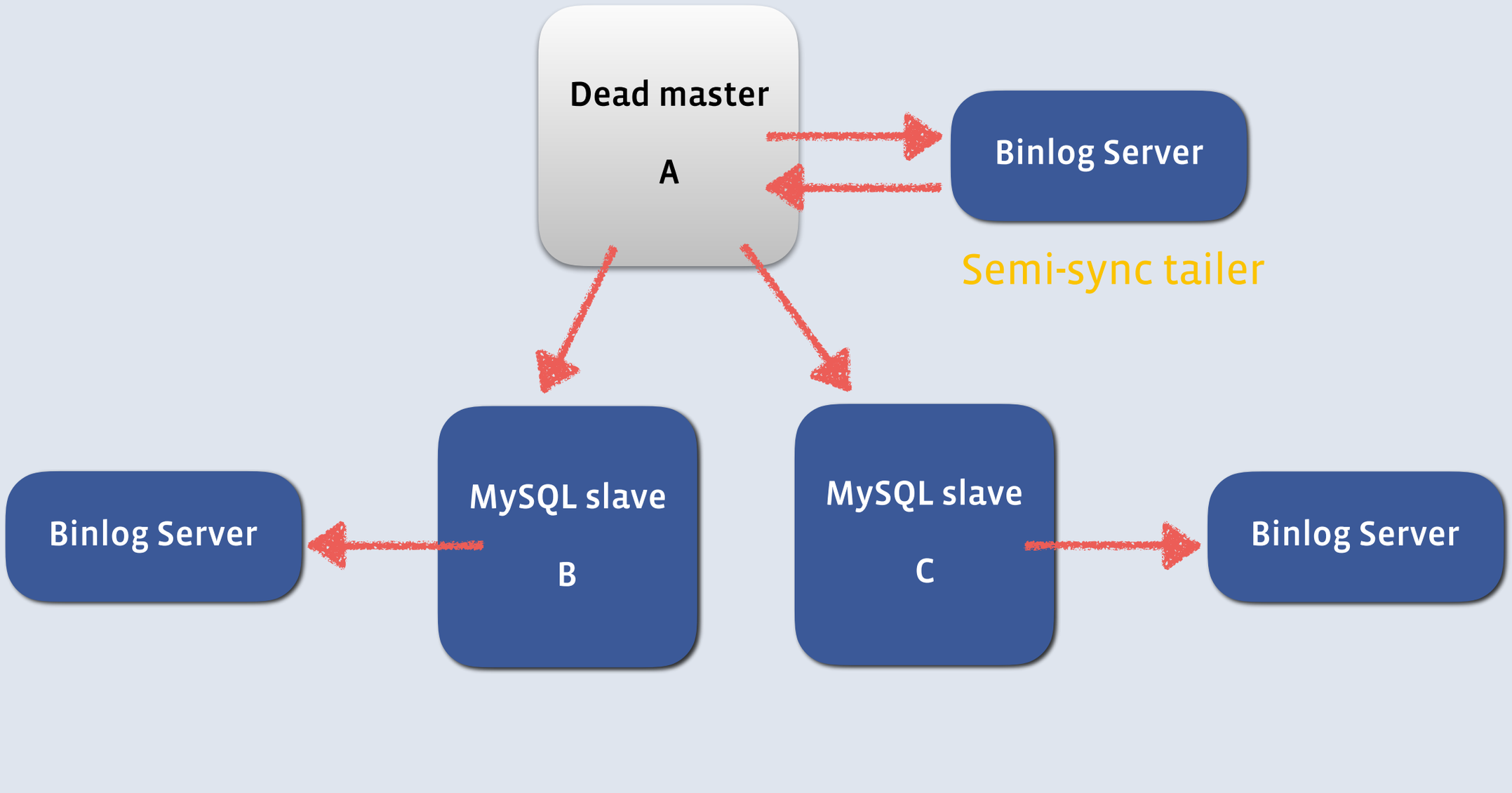
Binlog Server in Failover

- Binlog server used as semi-sync log tailers



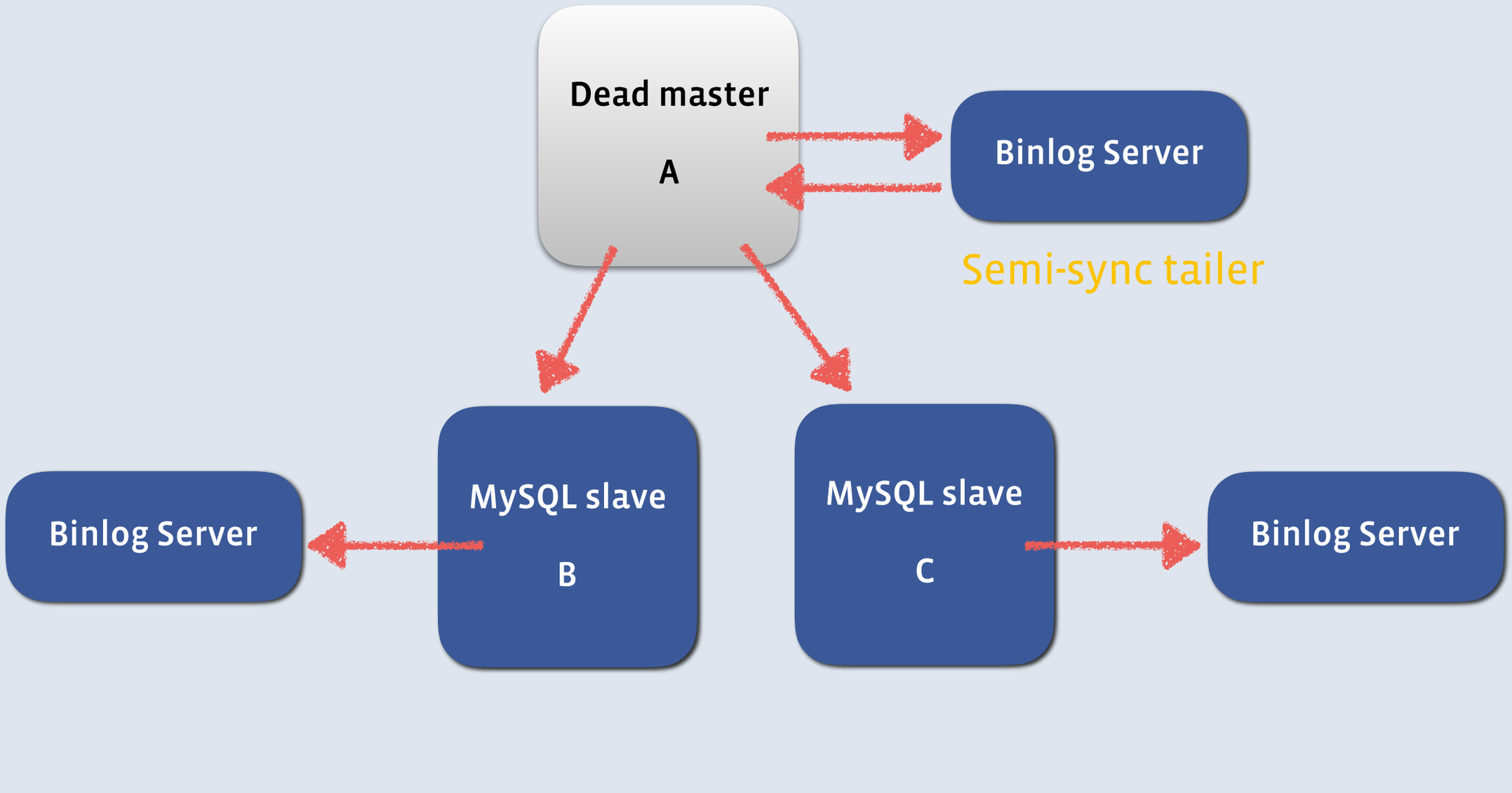
Binlog Server in Failover

- Dead master promotion is triggered



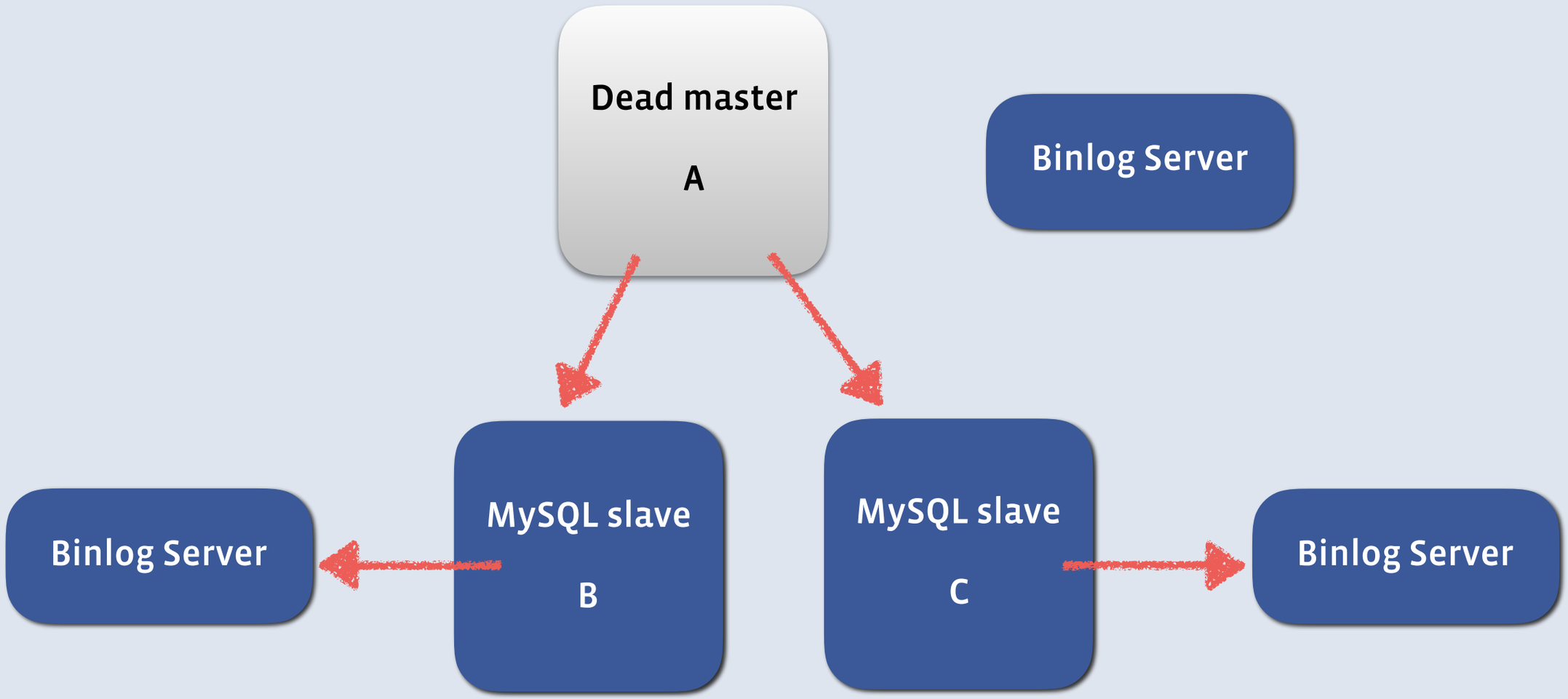
Binlog Server in Failover

- Stop binlog server's tailing to node fence dead master



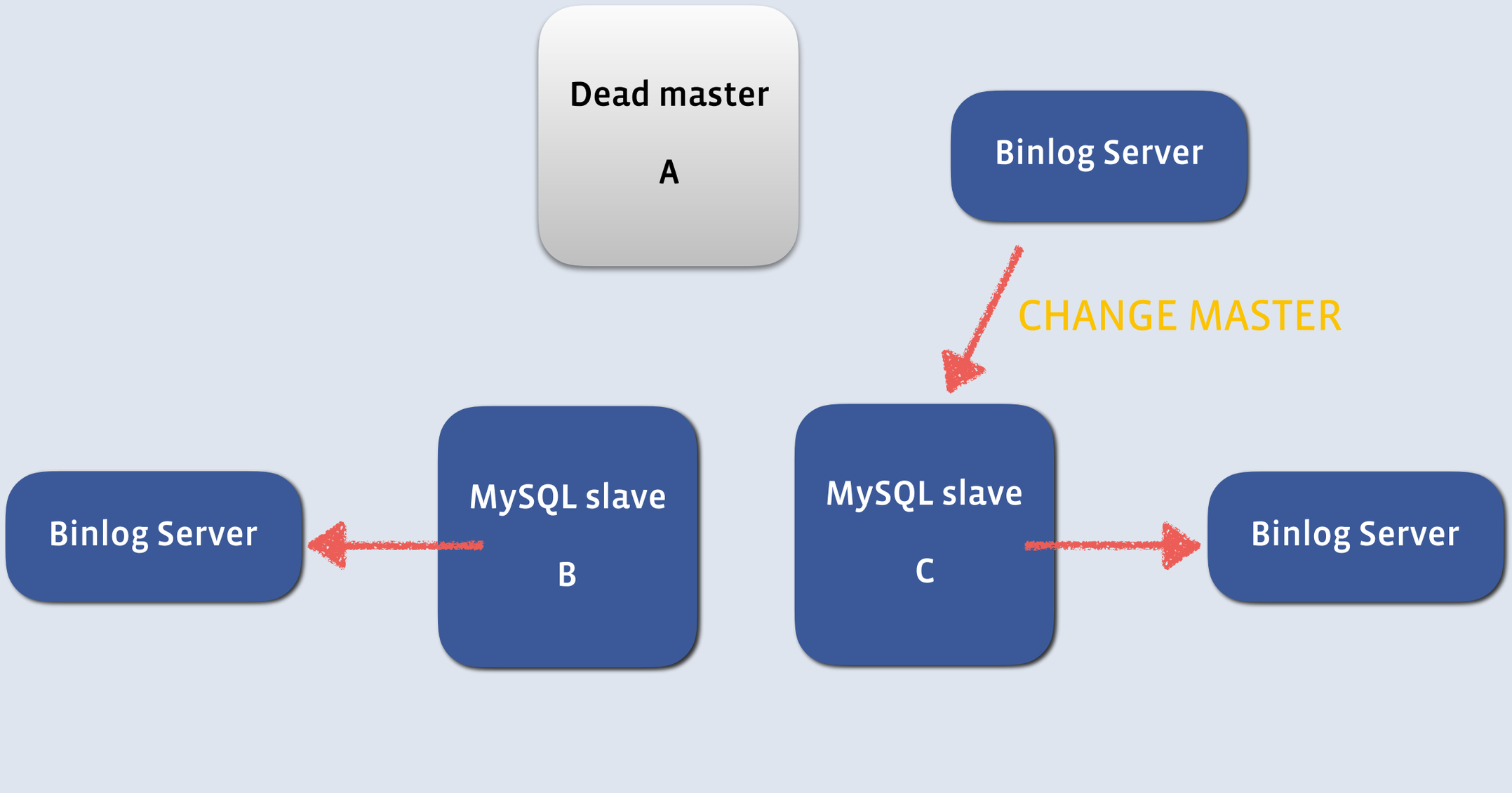
Binlog Server in Failover

- Pick a MySQL slave to promote



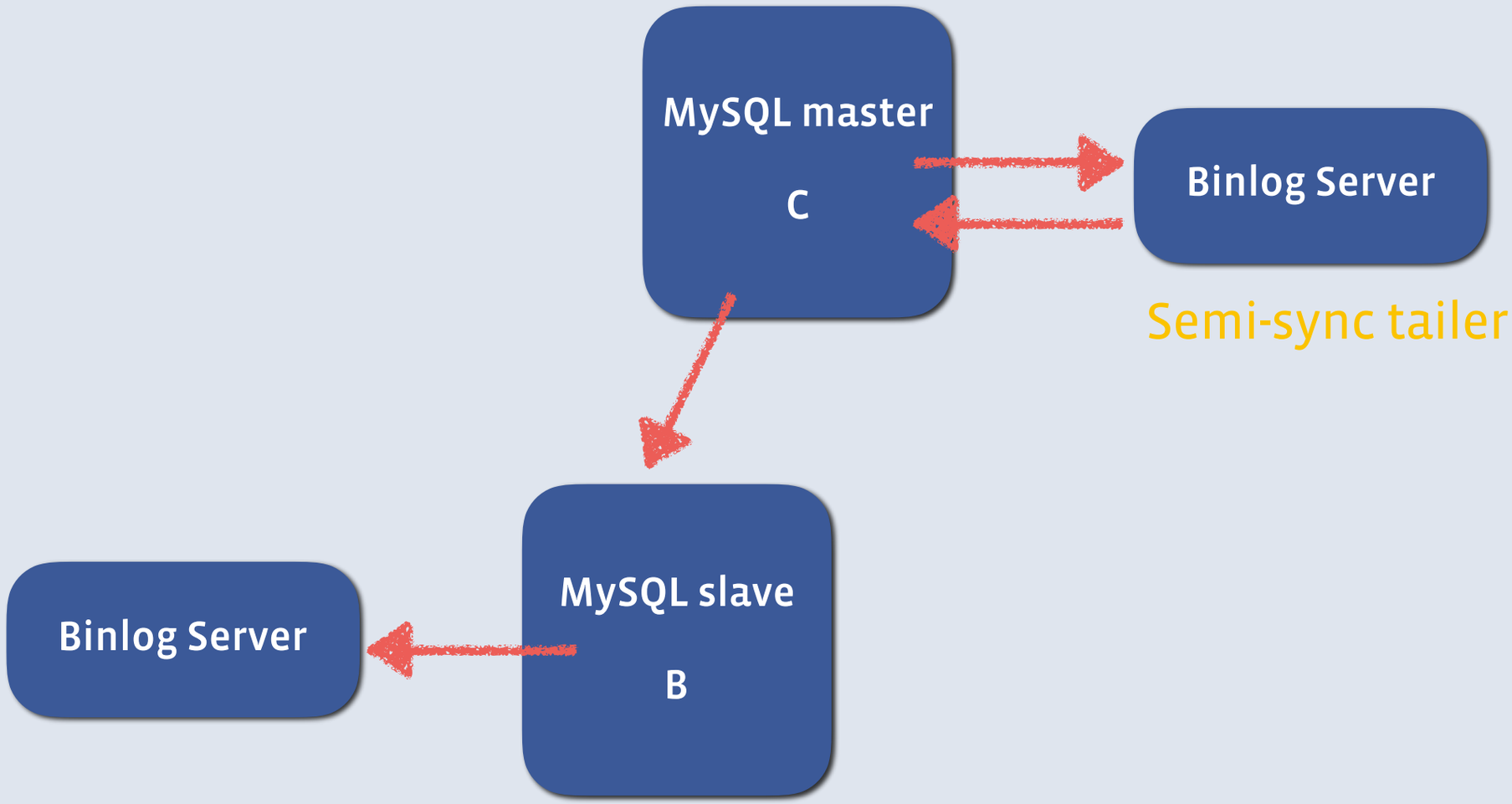
Binlog Server in Failover

- Catchup server C from binlog server using CHANGE MASTER



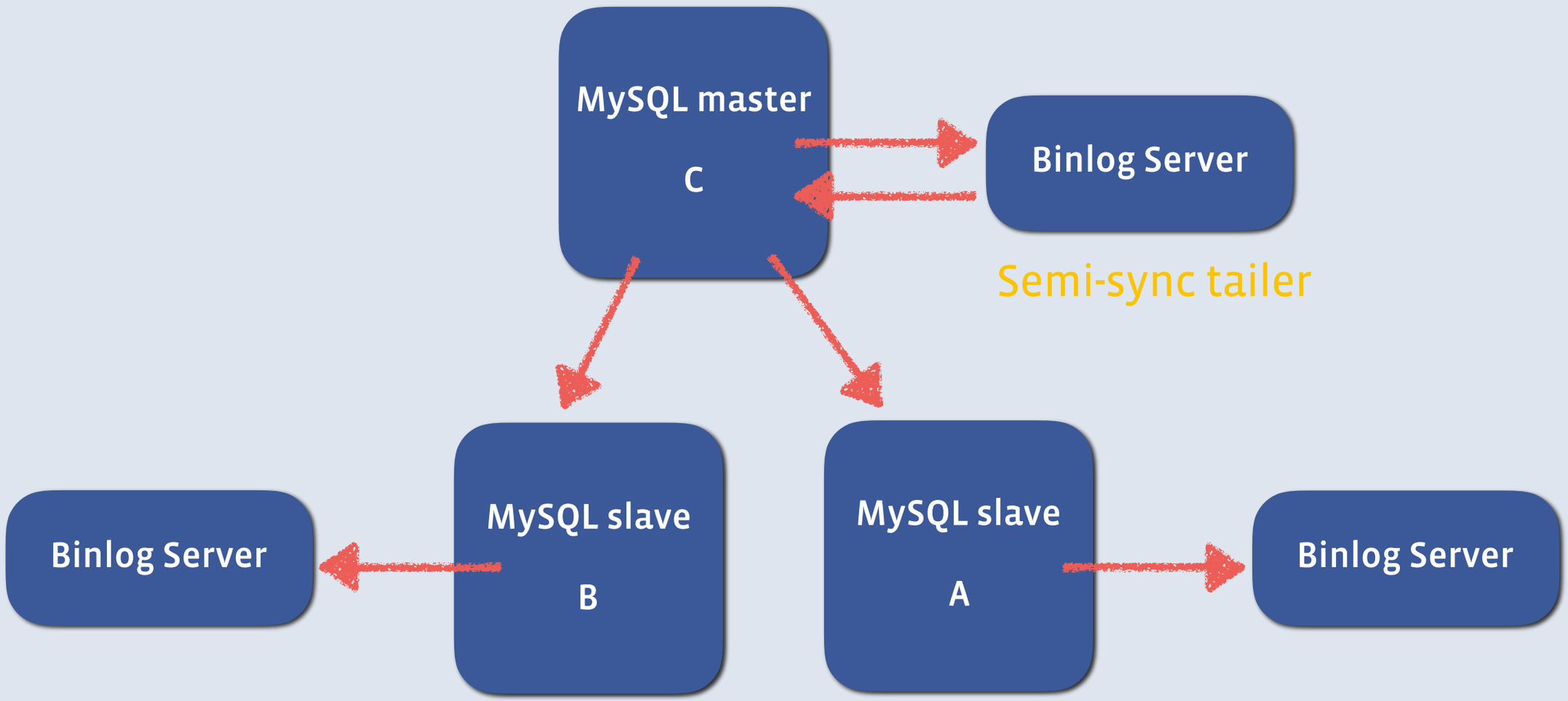
Binlog Server in Failover

- Promote server C as the new master



Binlog Server in Failover

- Recover dead master



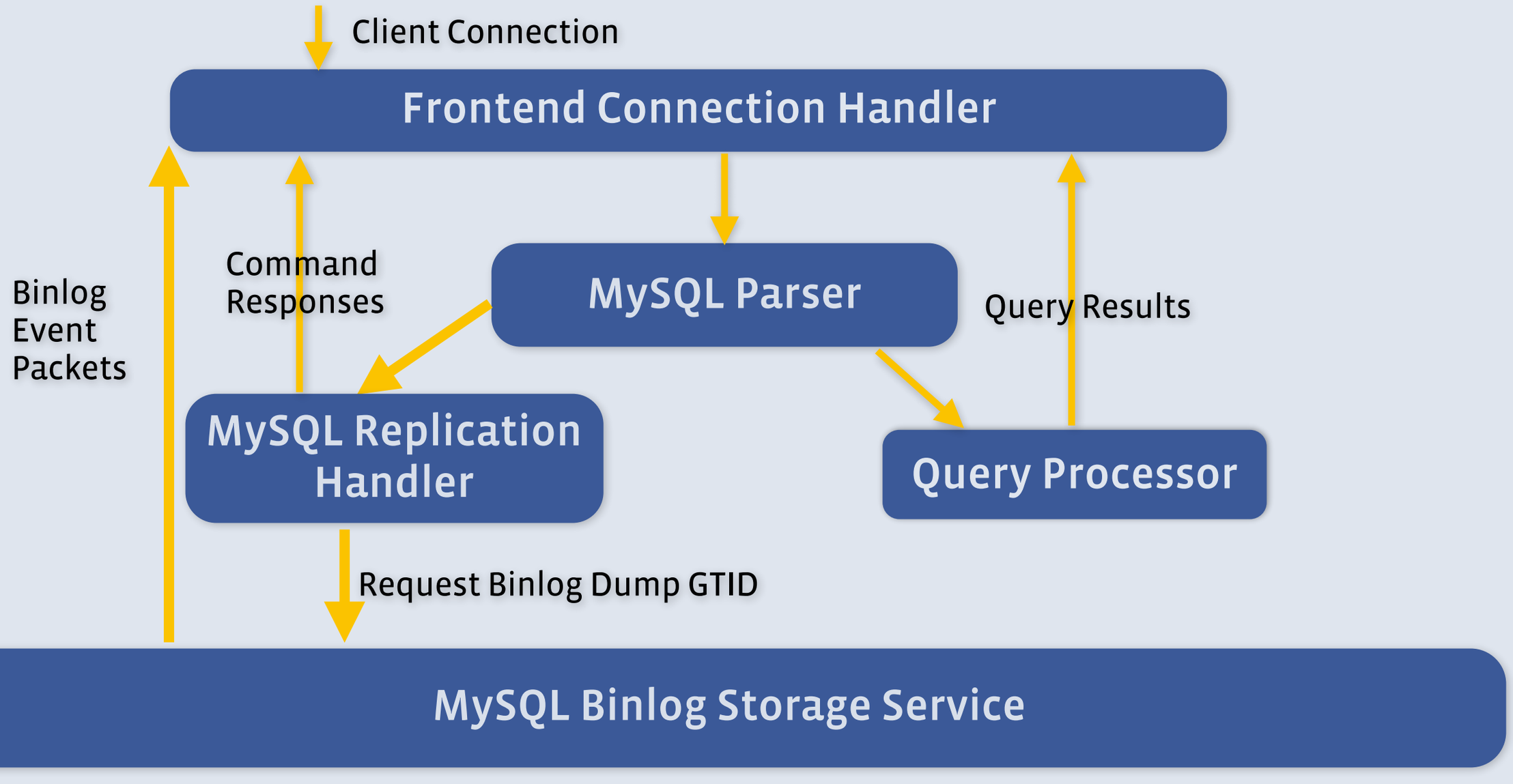
And more ...

- Point in time recovery of a single shard
- Disaster recovery of full MySQL instances
  - Binlog replay through replication is simpler, safer and reliable
- Binlog replay during Online Schema Change
  - Currently we are using table triggers to track deltas. With RBR, it is possible to replay per table binlog updates

Design of Binlog Server

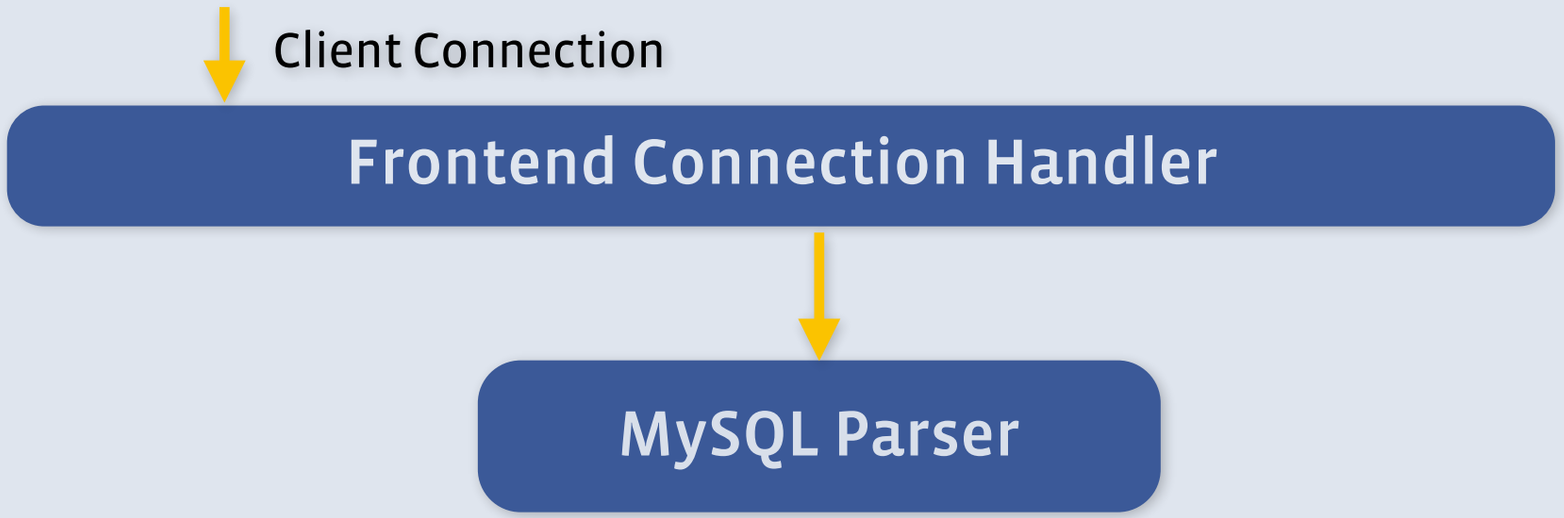
Binlog Server Design

Binlog Server Architecture



Binlog Server Design

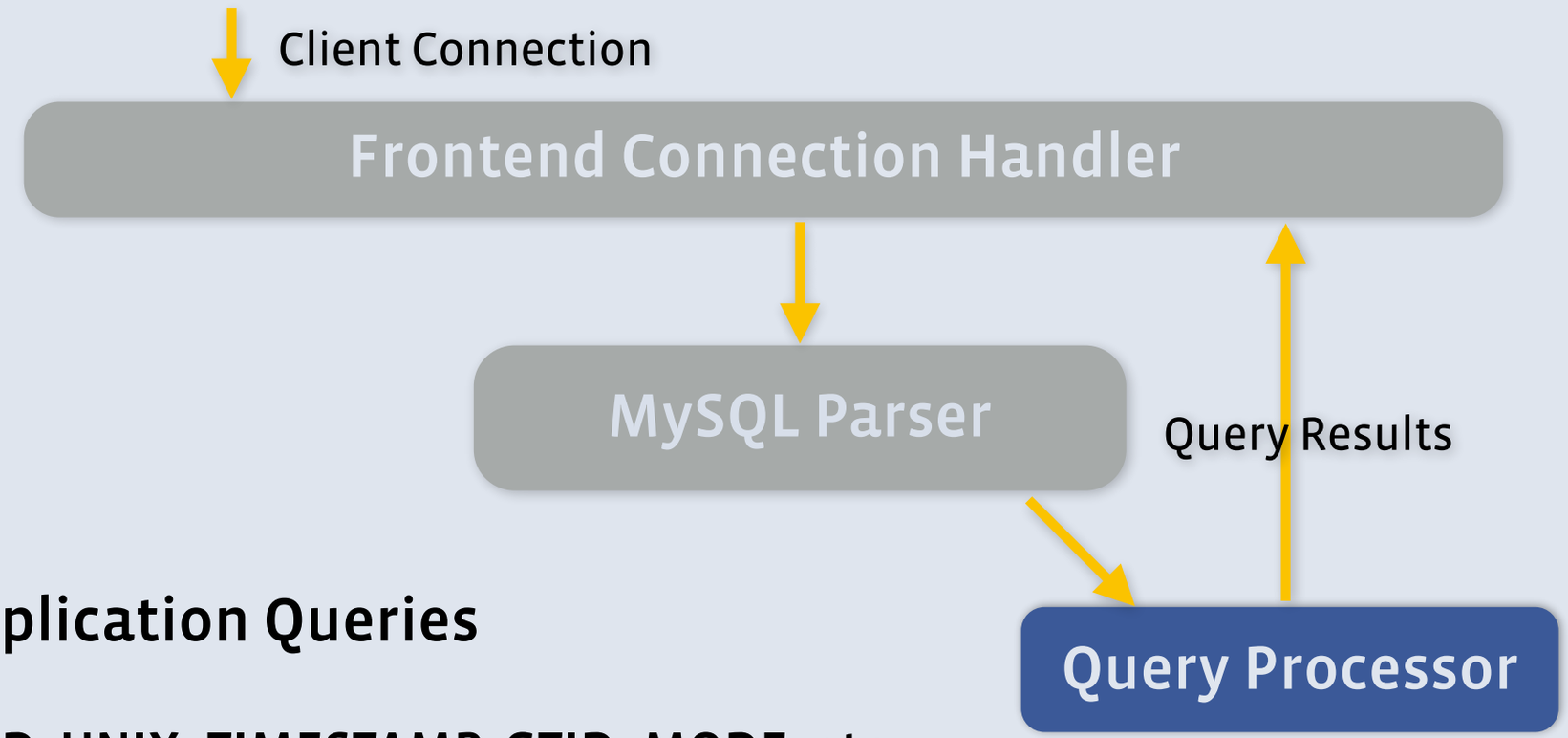
Handling MySQL Client Connections



- Built on the existing framework
  - MySQL connection/handshake handler
  - A compact MySQL parser

Binlog Server Design

Processing Replication Queries

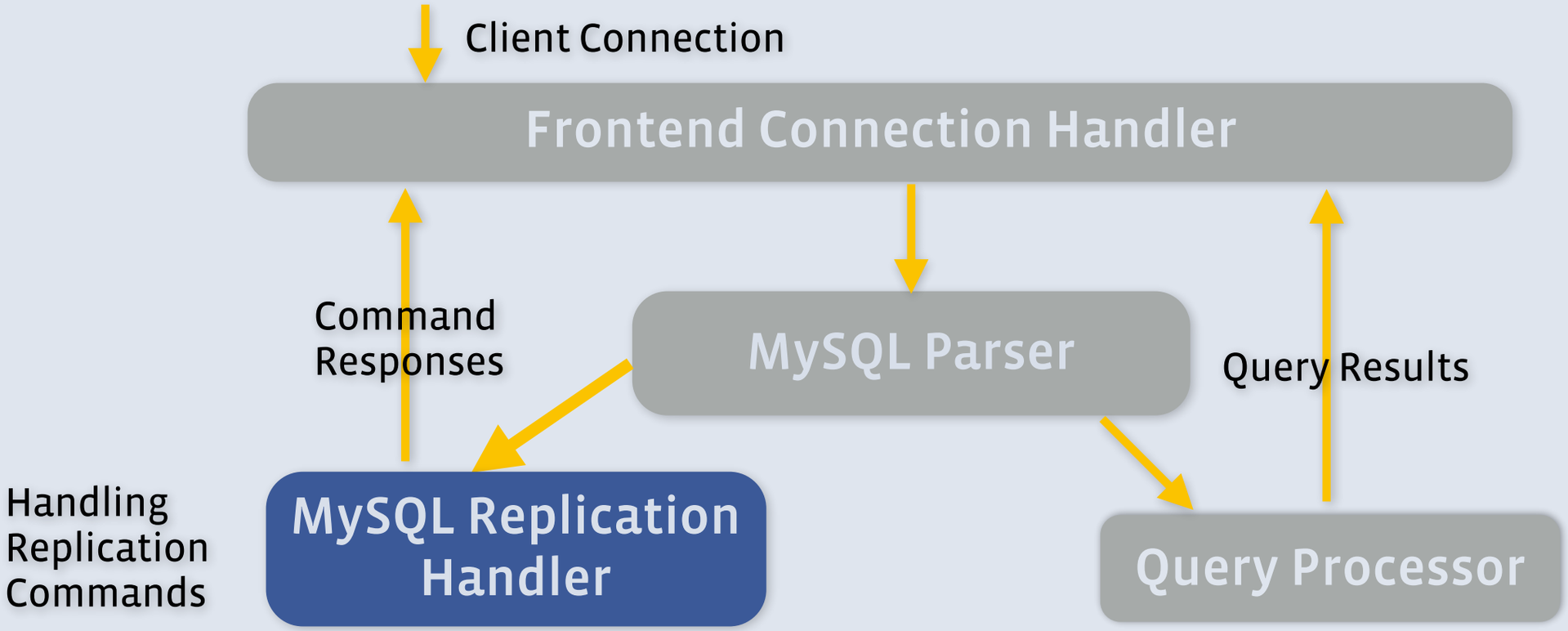


Process Replication Queries

- SELECT
  - SERVER_ID, UNIX_TIMESTAMP, GTID_MODE, etc...
- SHOW
  - rpl_semi_sync_master_enabled, SERVER_UUID
- SET
  - SLAVE_UUID, MASTER_HEARTBEAT_PERIOD

Binlog Server Design

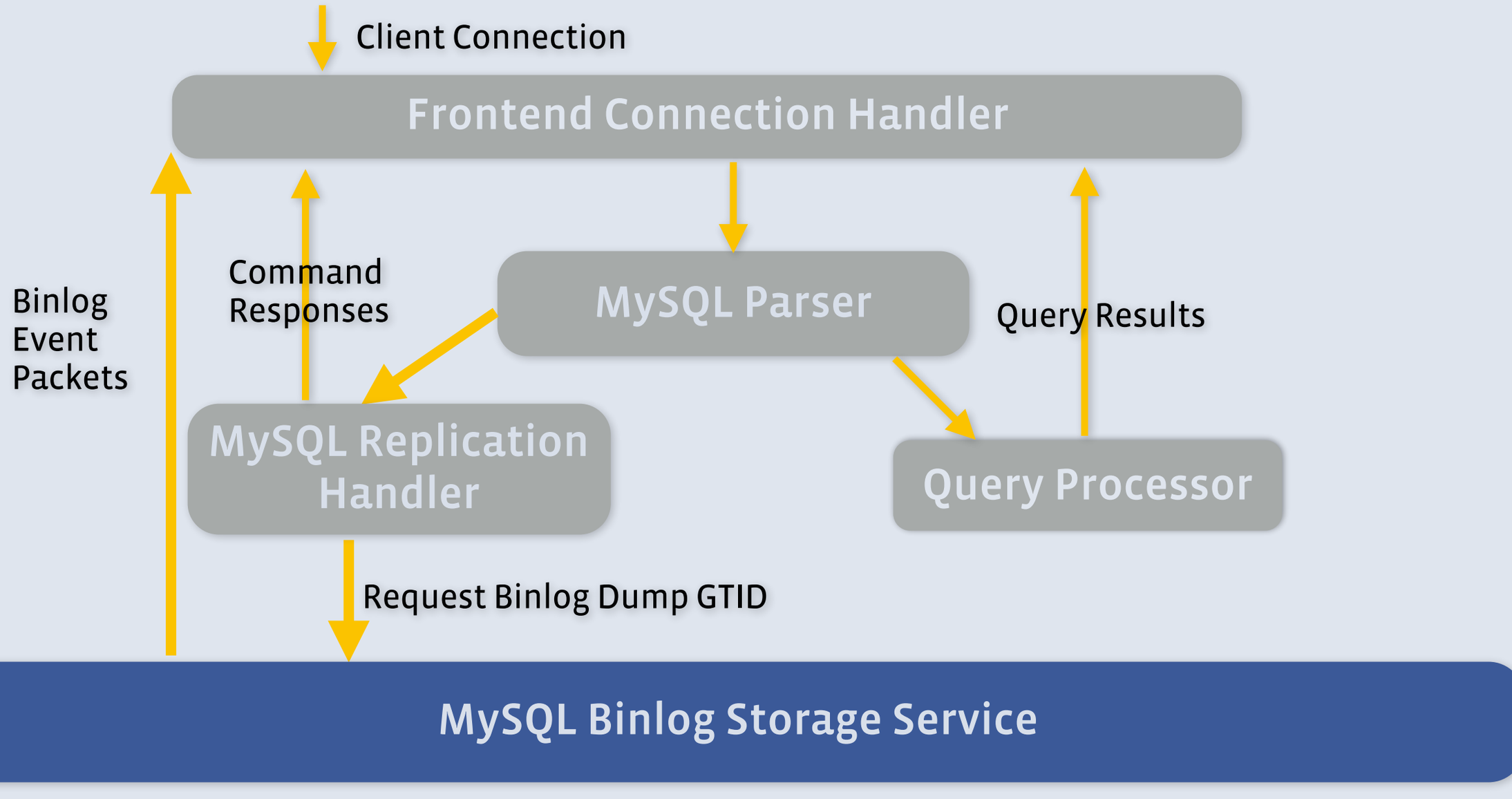
Processing Replication Commands



- Enabling MySQL replication protocol
  - COM_REGISTER_SLAVE, and COM_BINLOG_DUMP_GTID

Binlog Server Design

Handling Binlog Dump Requests



Binlog Server Design

MySQL Binlog Storage Service

- A library to plug binlog storage features
- Implemented the majority of MySQL replication protocol in GTID mode
- Components:
  - Binlog reader to fetch binlogs on different storage medias
  - Binlog locator
  - Binlog writer in semi-sync/async mode

Binlog Server Design

Binlog Server Operation Modes

- HDFS mode
  - Binlogs are backed up to HDFS with long retention time
  - Serving binlog backups on HDFS as a master
- Log-tailer mode
  - Backing up each MySQL instance's binlogs as a semi-sync tailer
  - Serving log-tailer's binlogs as a master

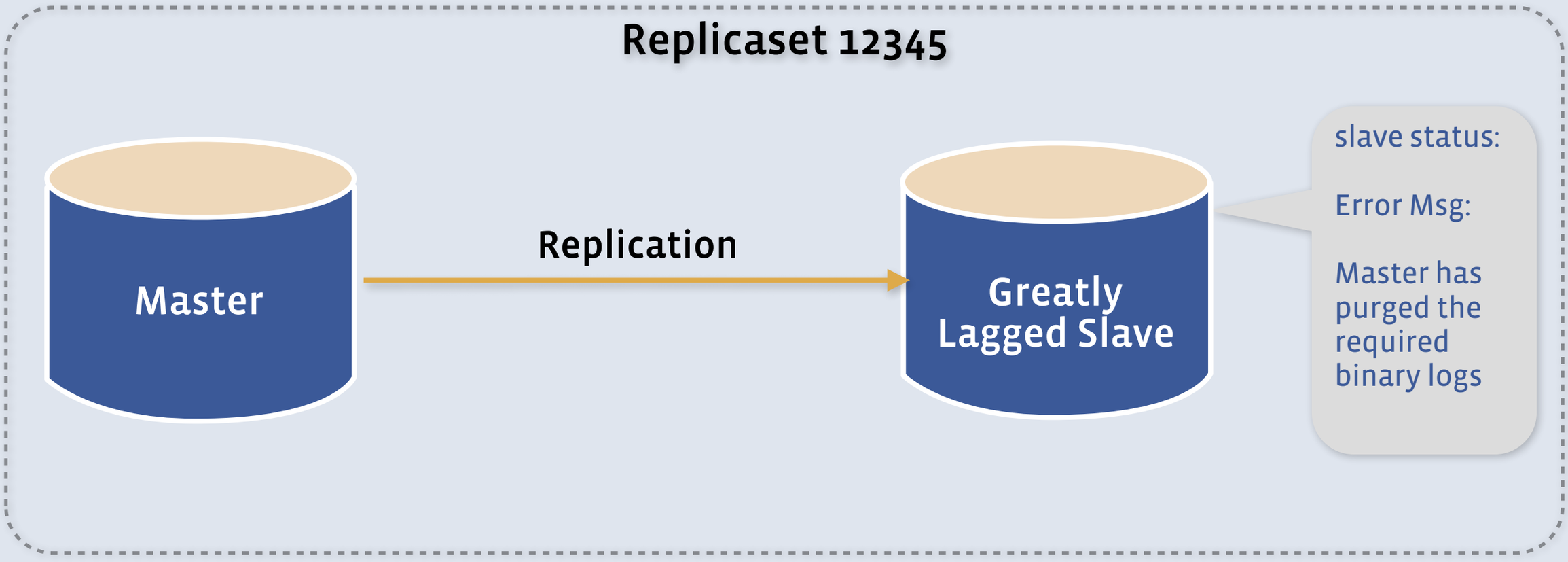
Binlog Server Design

Components in HDFS mode

- Binlog reader/sender from HDFS
  - A customized HDFS version of “binlog dump thread”
- HDFS binlog locator
  - Uses info stored in locator DB for each replicaset
    - HDFS binlog paths
    - Previous GTID sets of each binlog
  - Locates the list of required HDFS binlogs
    - With a given GTID set

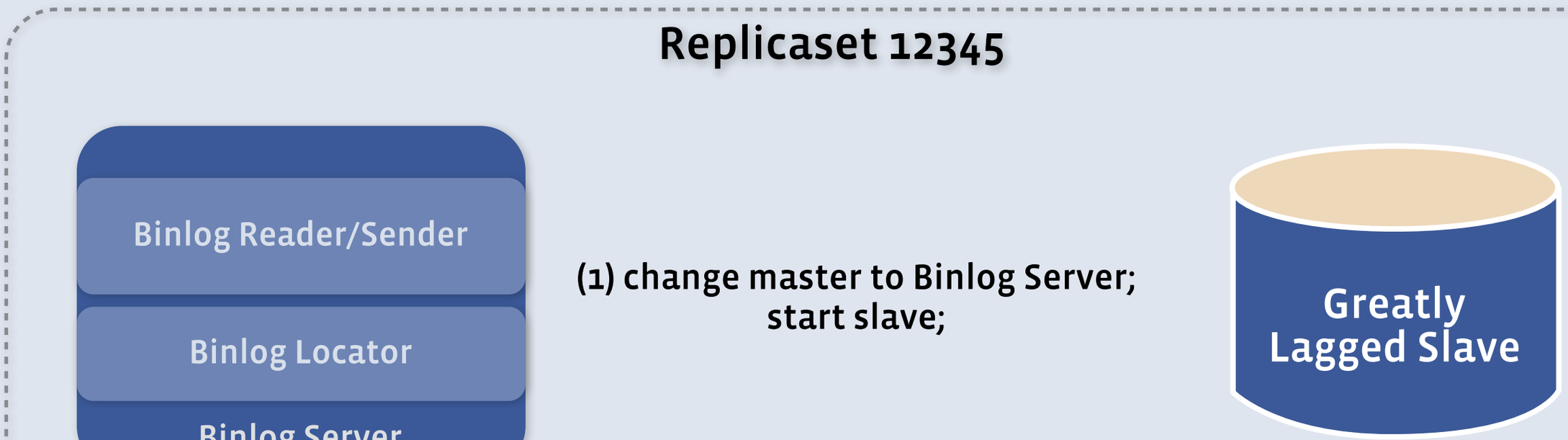
Binlog Server Design

Binlog Server in HDFS mode



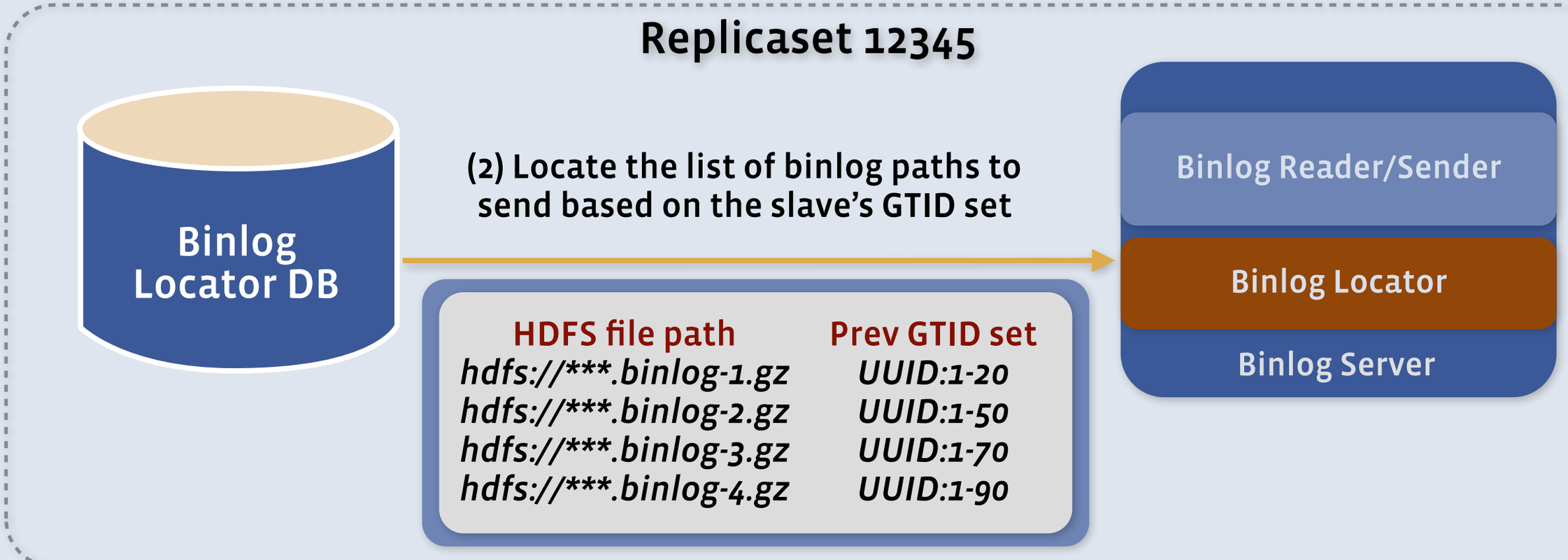
Binlog Server Design

Binlog Server in HDFS mode



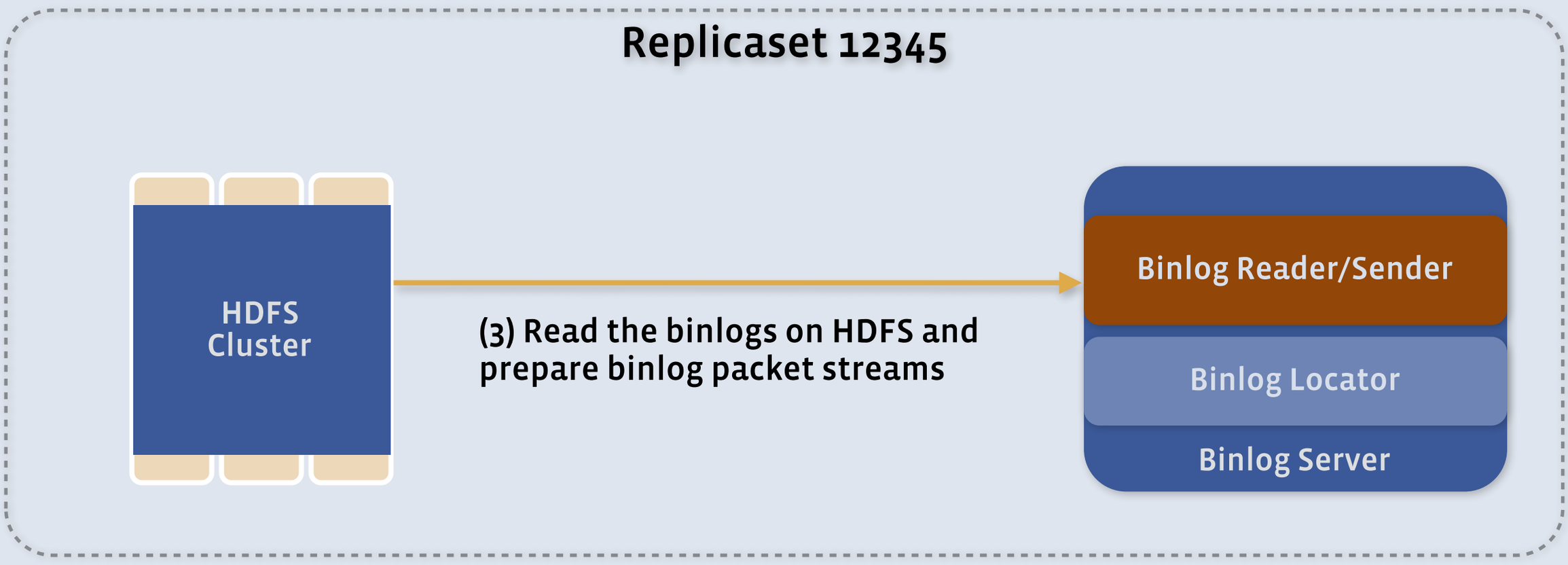
Binlog Server Design

Binlog Server in HDFS mode



Binlog Server Design

Binlog Server in HDFS mode



Binlog Server Design

Binlog Server in HDFS mode

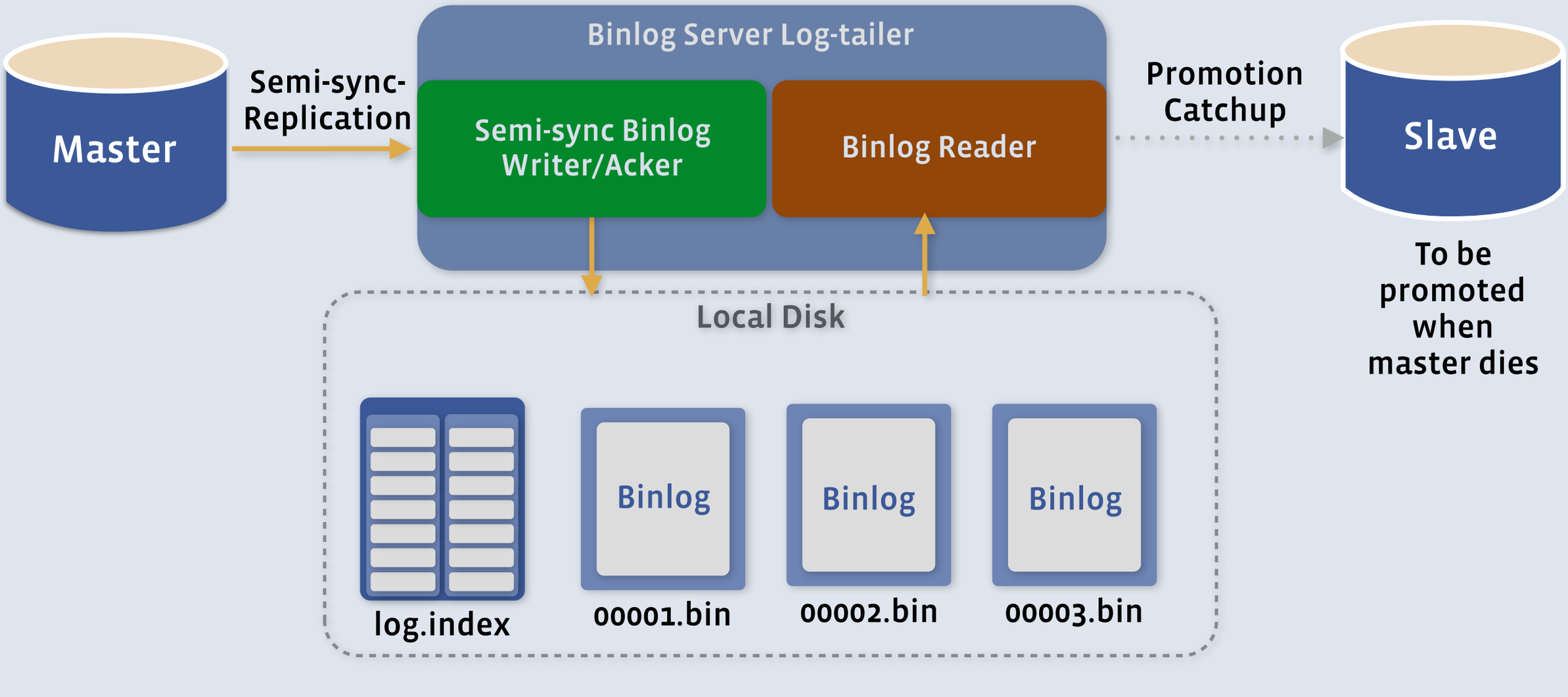
Binlog Server Design

Components in Log-tailer mode

- Binlog writer with acknowledgment capability
  - Connecting to the MySQL as a semi-sync slave
  - Writing binlogs to the Disk
  - Acknowledge the MySQL when requested by the master
- Binlog reader/sender from Disk
  - A customized version of “binlog dump thread”

Binlog Server Design

Binlog Server in Log-tailer mode



Operational Commands

Operational commands

Show Master Status

- HDFS mode

```
binlog_server> show master status\G
*************************** 1. row ***************************
             File: hdfs://******.binary-logs-xxxxxx.xxxxxx.gz
         Position: 4
Executed_Gtid_Set: 6c597fb0-d3a4-4aab-ba93-2286a75727ed:1-81669,
765a6781-d959-492b-8091-e6adeac313ee:1-53168
```

- Log-tailer mode

```
binlog_server>show master status\G
*************************** 1. row ***************************
             File: binary-logs-3306.007965
         Position: 13366
Executed_Gtid_Set: 49f5e0ca-80d2-4616-be83-d1aeb5e973bc:1-902909,
73707584-d9d1-49f1-b2bf-0ffb5e603b2d:1-81669
```

Operational commands

Show Slave Status in Log-tailer mode

```
binlog_server> show slave status\G
*************************** 1. row ***************************
         Slave_IO_State: Waiting for master to send event
            Master_Host: HOSTNAME
            Master_Port: PORT
          Connect_Retry: 0
        Master_Log_File: binary-logs-xxxxxx.007964
    Read_Master_Log_Pos: 97115
            Binlog_File: binary-logs-xxxxxx.007964
          Binlog_Pos: 97115
        Last_IO_Errno: 0
       Master_Server_Id: 3695980966
      Executed_Gtid_Set: ea4a5e01-b3e4-4273-a25e-88d06db8d1a5:1-902842,
b29a87bd-d60b-4455-9ab8-90d7b720f169:1-81669
       Mysql_Replicaset: REPLICA_SET_NAME
Replicaset_Tier_Version: VERSION_NUM
         Semisync_Slave: Yes
```

Operational commands

Show Master Logs in Log-tailer mode

```
binlog_server> show master logs;

+-------------------------+-----------+
| Log_name                | File_size |
+-------------------------+-----------+
| binary-logs-3306.007962 | 124002    |
| binary-logs-3306.007963 | 131261    |
| binary-logs-3306.007964 | 15707     |
| binary-logs-3306.007964 | 110983    |
| binary-logs-3306.007965 | 127464    |
| binary-logs-3306.007966 | 135975    |
+-------------------------+-----------+
```

```
binlog_server> show master logs with gtid\G

*************************** 1. row
     Log_name: binary-logs-3306.007963
    File_size: 131261
Prev_gtid_set: 561d1725-ed2e-458a-a496-77c65701e6d7:1-902253,
1e407547-ca35-4838-a19c-e3c90e33ebd4:1-81669

*************************** 2. row
     Log_name: binary-logs-3306.007964
    File_size: 110983
Prev_gtid_set: 561d1725-ed2e-458a-a496-77c65701e6d7:1-902590,
1e407547-ca35-4838-a19c-e3c90e33ebd4:1-81669

                                      ......
```

Operational commands

Purging Logs in Log-tailer mode

```
binlog_server> show master logs;
+-------------------------+-----------+
| Log_name                | File_size |
+-------------------------+-----------+
| binary-logs-3306.007962 | 124002    |
| binary-logs-3306.007963 | 131261    |
| binary-logs-3306.007964 | 15707     |
+-------------------------+-----------+


binlog_server> purge logs to binary-logs-3306.007963;
Query OK, 0 rows affected (0.00 sec)


binlog_server> show master logs;
+-------------------------+-----------+
| Log_name                | File_size |
+-------------------------+-----------+
| binary-logs-3306.007963 | 131261    |
| binary-logs-3306.007964 | 69083     |
+-------------------------+-----------+
```

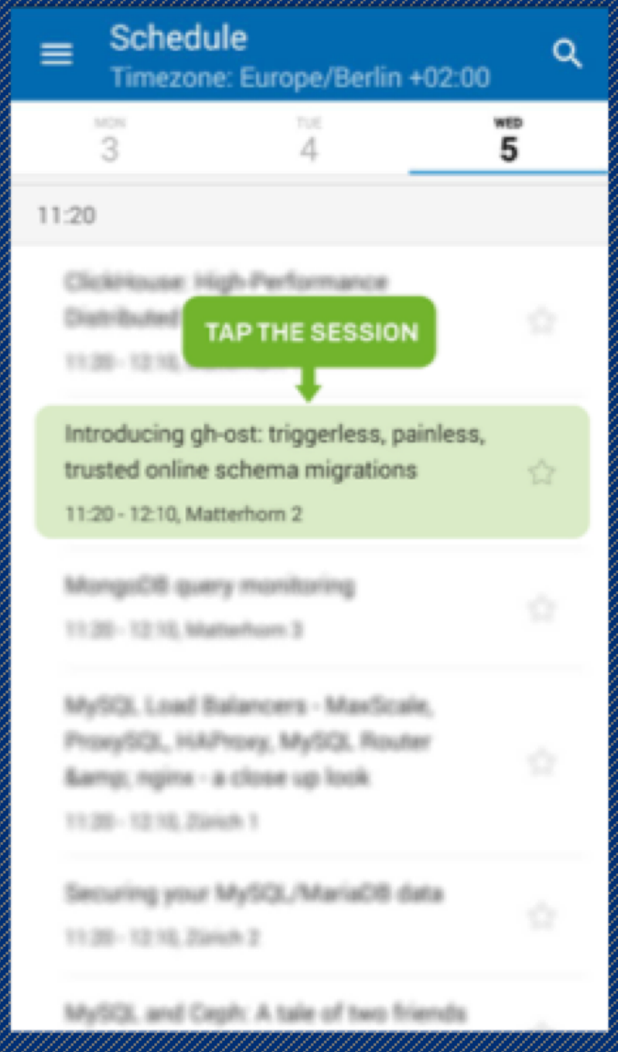
Operational commands

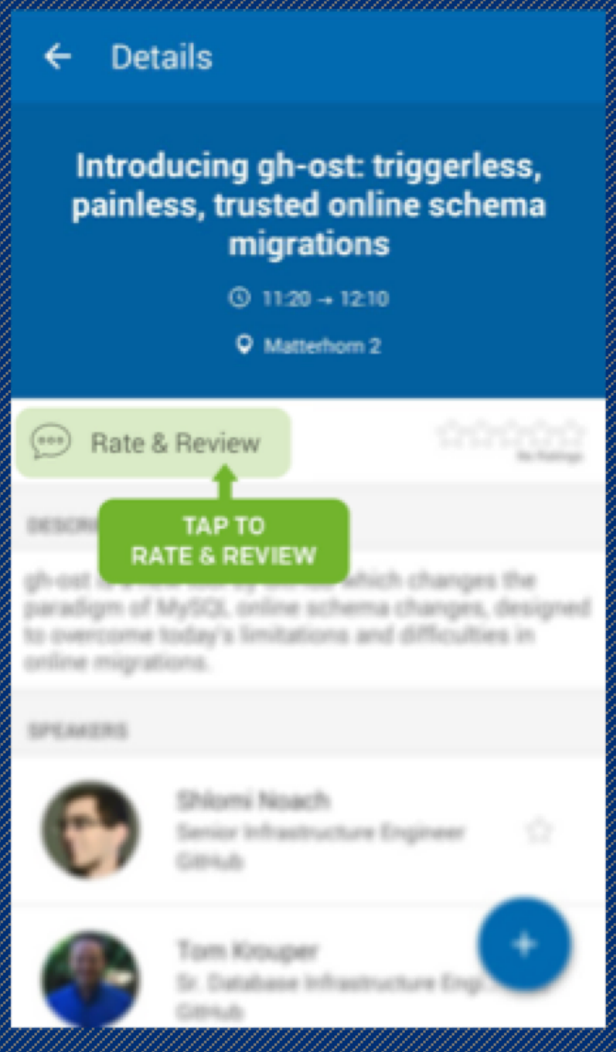
Start/Stop Slave in Log-tailer mode

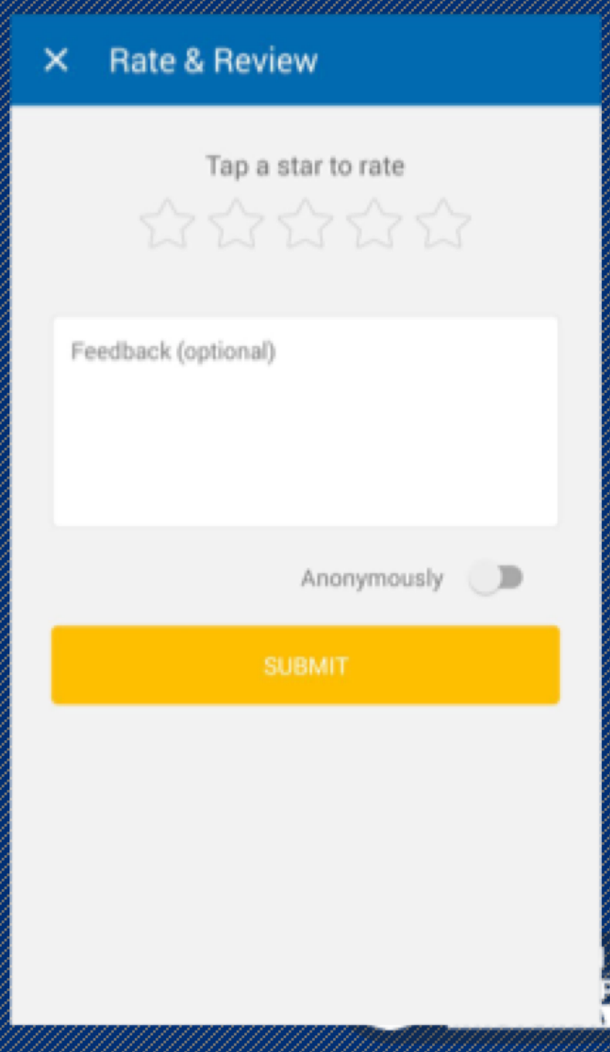
```
binlog_server> start slave
binlog_server> show slave status\G
*************************** 1. row
         Slave_IO_State: Waiting for master to send event
            Master_Host: HOSTNAME
            Master_Port: 3336
          Connect_Retry: 0

                  ......


binlog_server> stop slave
binlog_server> show slave status\G
*************************** 1. row
         Slave_IO_State: Stopped
            Master_Host: HOSTNAME
            Master_Port: 3336
          Connect_Retry: 0

                  ......
```

Questions?

Rate My Session!







facebook