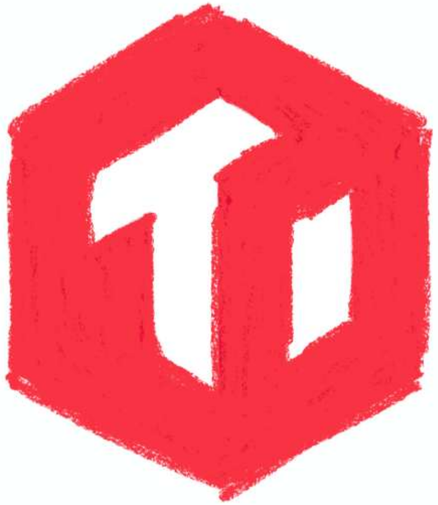
TiDB 101



fipar - Percona University MVD 2026

About me

DBRE @ Life360

First broke a prod DB in 2001



TiDB Overview

- Distributed DBMS
- HTAP
  - OLAP
  - OLTP
- MySQL protocol-compatible

TiDB?

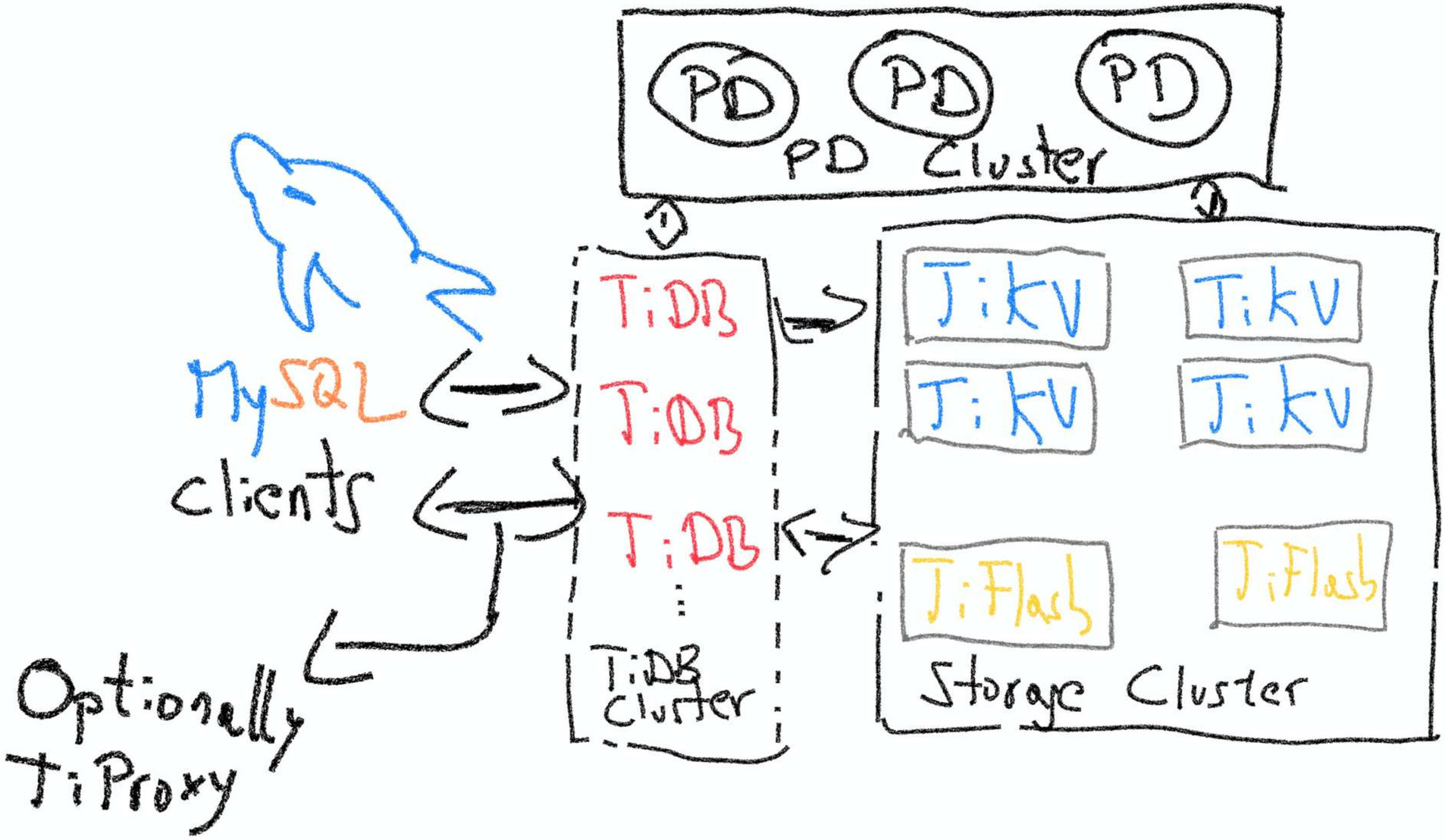
Yes

- VLDB
- Strong Consistency

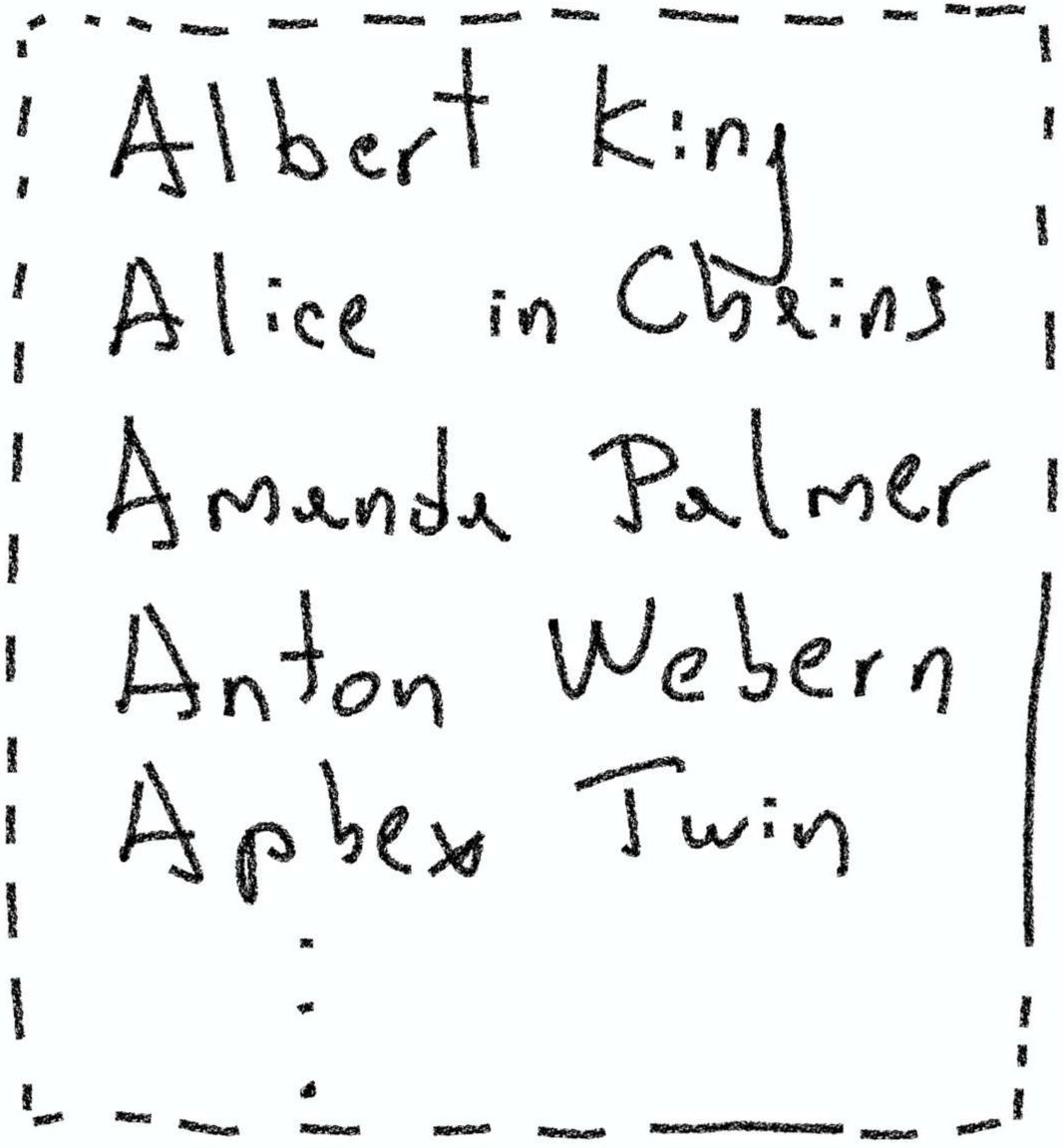
No

- Small DBs (< 10M rows)
- Relaxed Consistency

Architecture



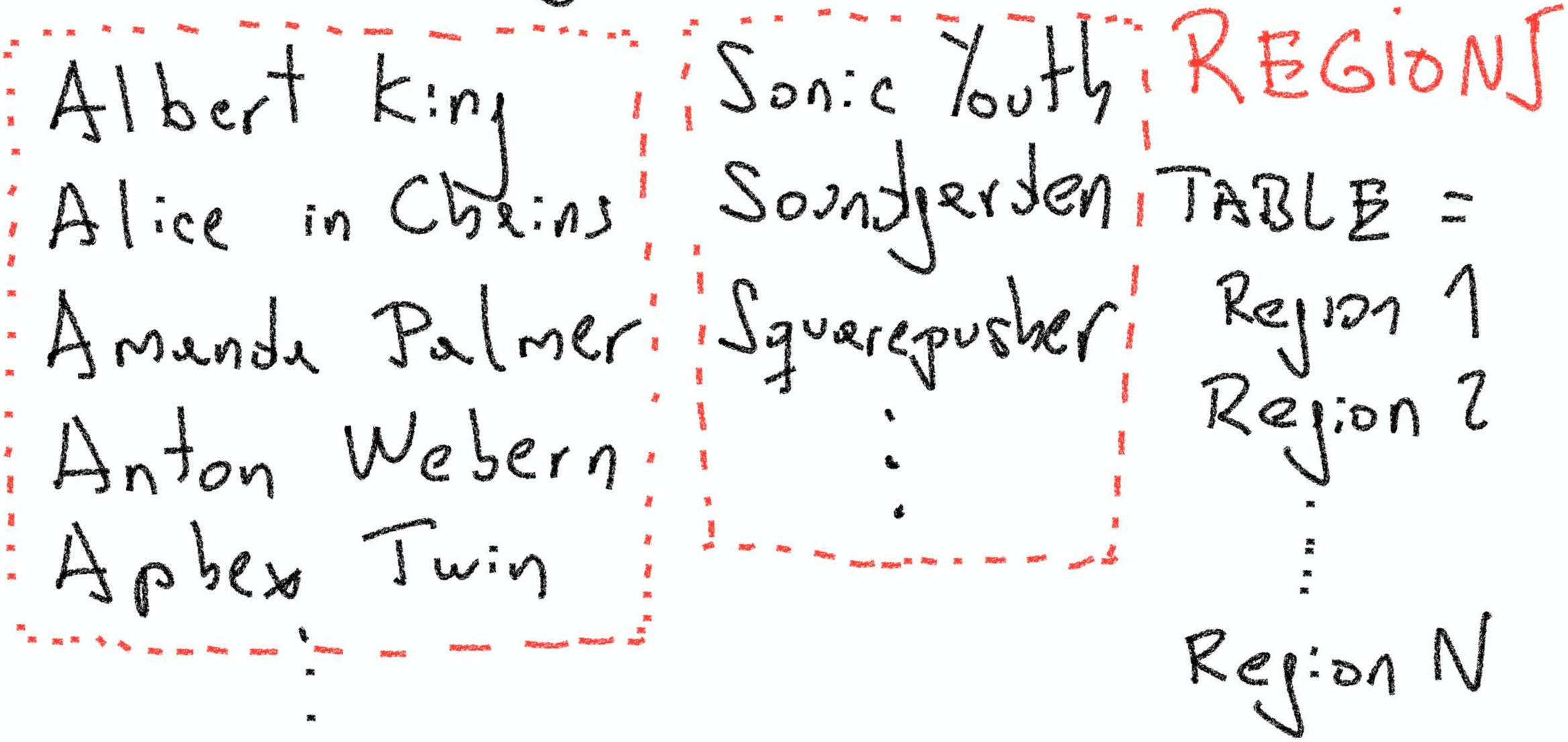
Date organization: MySQL



TABLE

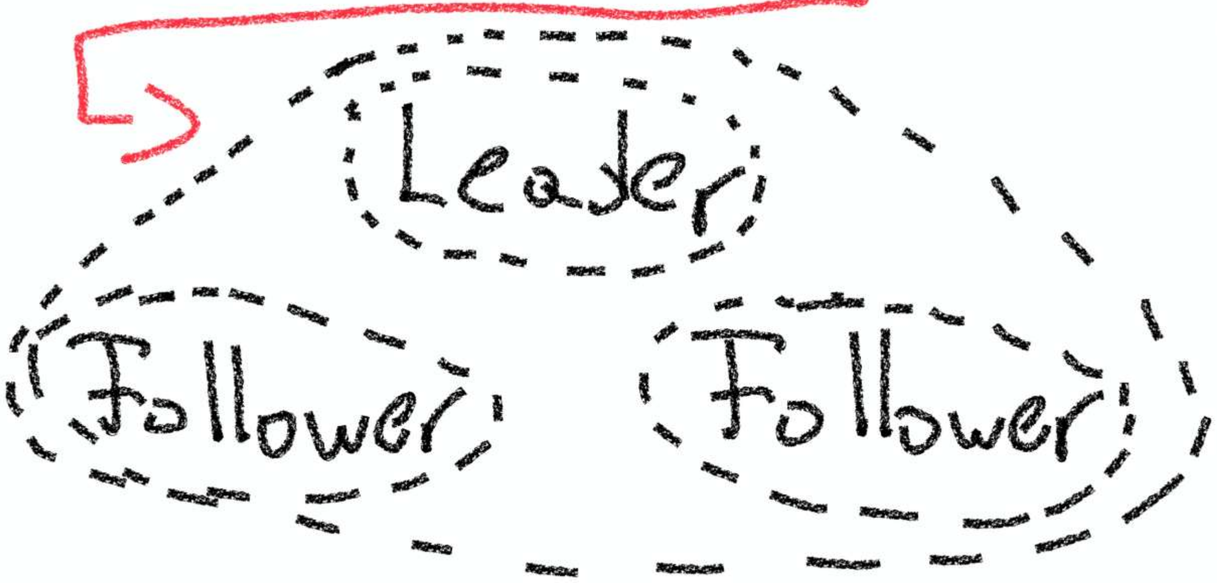
Maybe Partitions

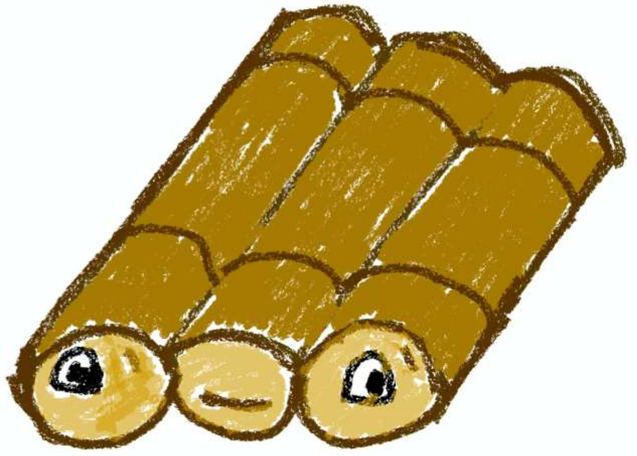
Data Organization : TiDB



A Region is a Raft Group responsible for a range of keys

CONSENSUS





Raft in TiDB

Role	Responsibilities
Leader	• writes and reads
Follower	• Ack writes • vote • can take reads (follower reads)
Learner	• Apply writes • don't vote or Ack

Raft in TiDB

The DB is a Replicated State Machine

1. Client sends (propose) write
2. Leader appends to its log
3. Leader replicates log
4. Followers ACK (or not...)
5. Majority ACK? => Commit
6. Apply to State Machine
7. Followes do the same

Writes in TiDB

- Always wait for majority Ack
- No majority? No write
- TiDB is C A P
- During P only a majority can continue
- The DB will always be C
- ∴ it won't always be A

Placement Driver Cluster

- Control Plane
- Brain of the Cluster
  - Stores metadata
  - PLACEs data
  - Rebalancing, etc.
  - No PD? No DB



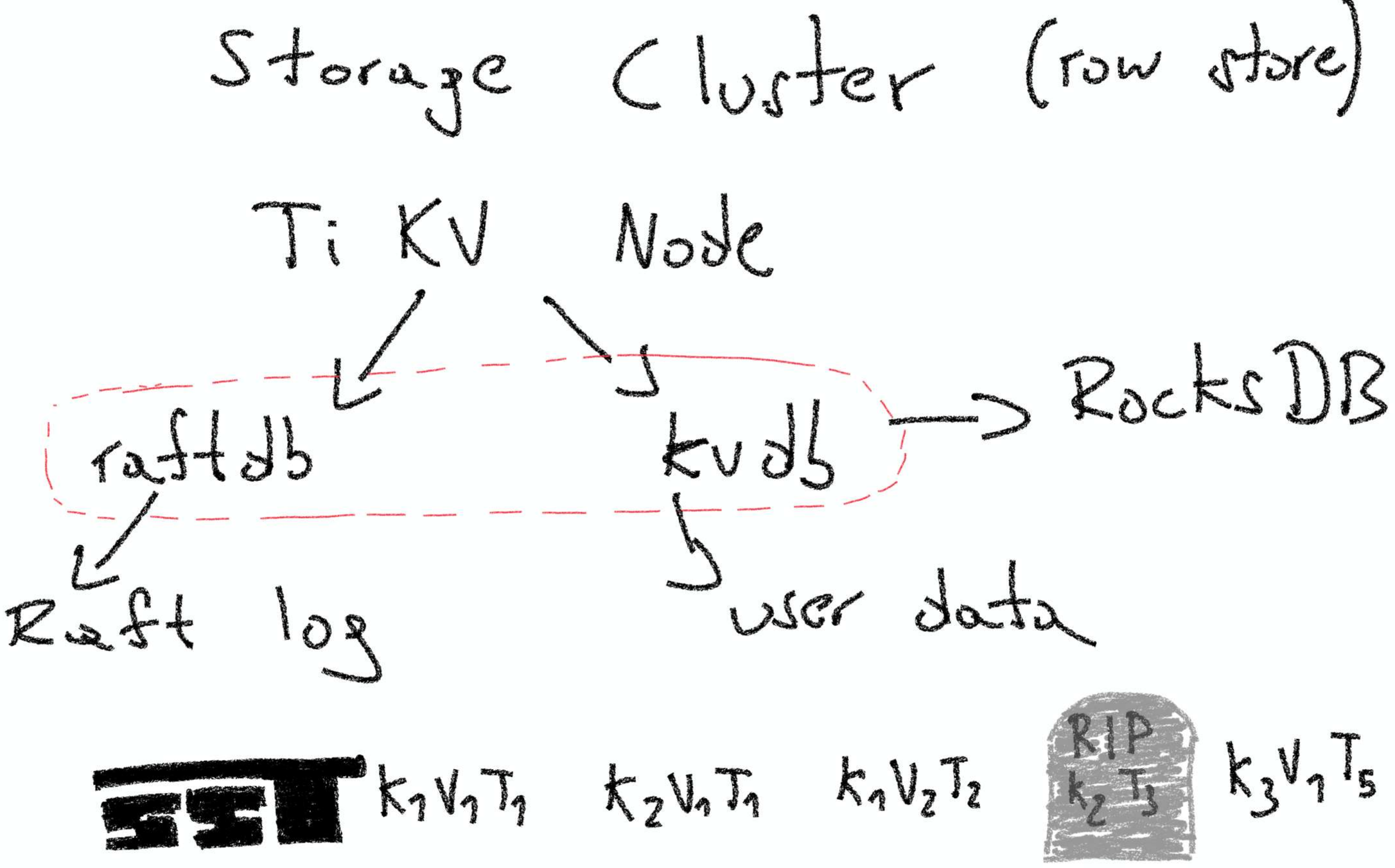
TiKV
TiKV
TiKV
TiKV

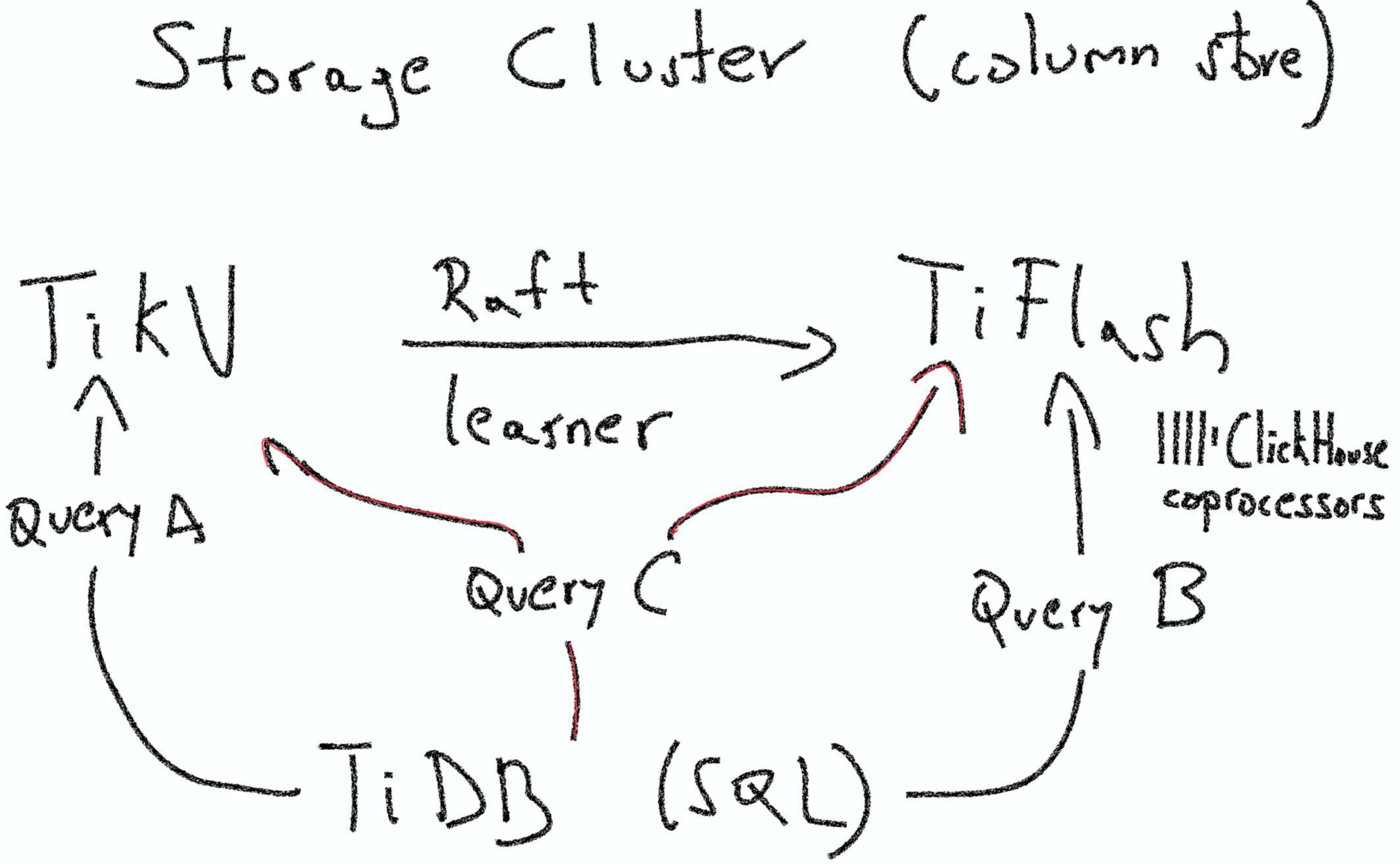
TiFlash
TiFlash

TiDB Cluster

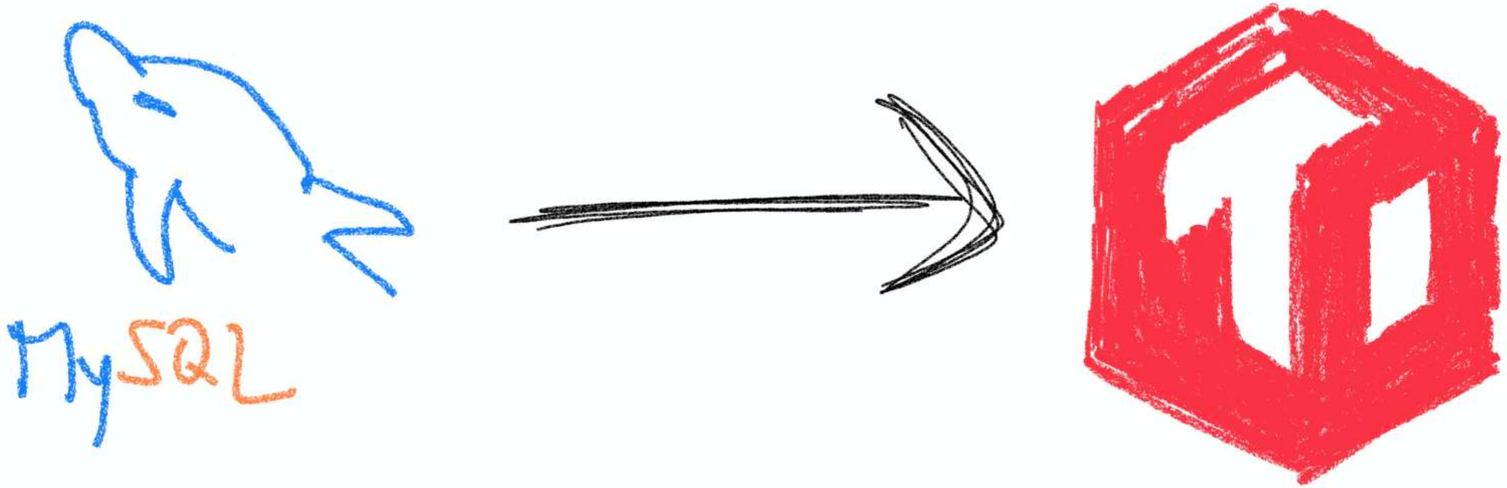
- Stateless SQL layer
- Easy to scale out/in
- MySQL protocol

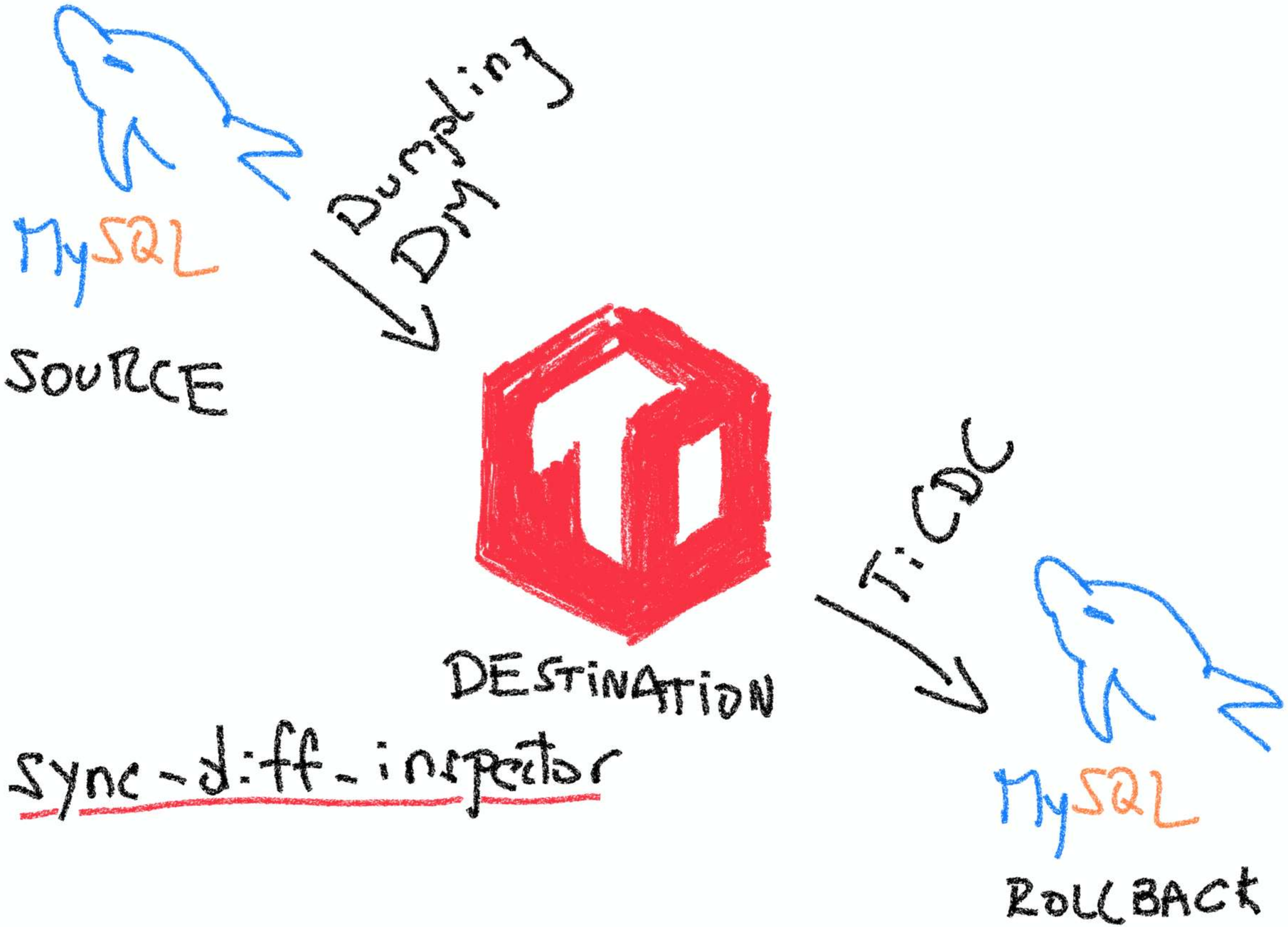
Storage Cluster (row store)




Storage Cluster (column store)
TiKV
Raft
learner
TiFlash
ClickHouse coprocessors
Query A
Query C
Query B
TiDB (SQL)

Migrating




MySQL
SOURCE
Dumping
Dm
DESTINATION
sync-diff-inspector
TiCDC
MySQL
ROLLBACK

Migrating

- AUTO INCREMENT → AUTO RANDOM
- WATCH KEY DATA TYPE OVERFLOW
- BINDINGS (DB-side index hints)
- ProxySQL query rewrite rules

Questions?

Answers here or over 

fiper@acm.org