QCon 全球软件开发大会
INTERNATIONAL SOFTWARE DEVELOPMENT CONFERENCE

BEIJING 2018

TiDB 架构及开源之路

申砾 @ PingCAP

主办方 Geekbang. 极客邦科技 InfoQ

About Me

- VP of Engineering @ PingCAP
- Netease/360/PingCAP
- Tech lead of TiDB

About PingCAP

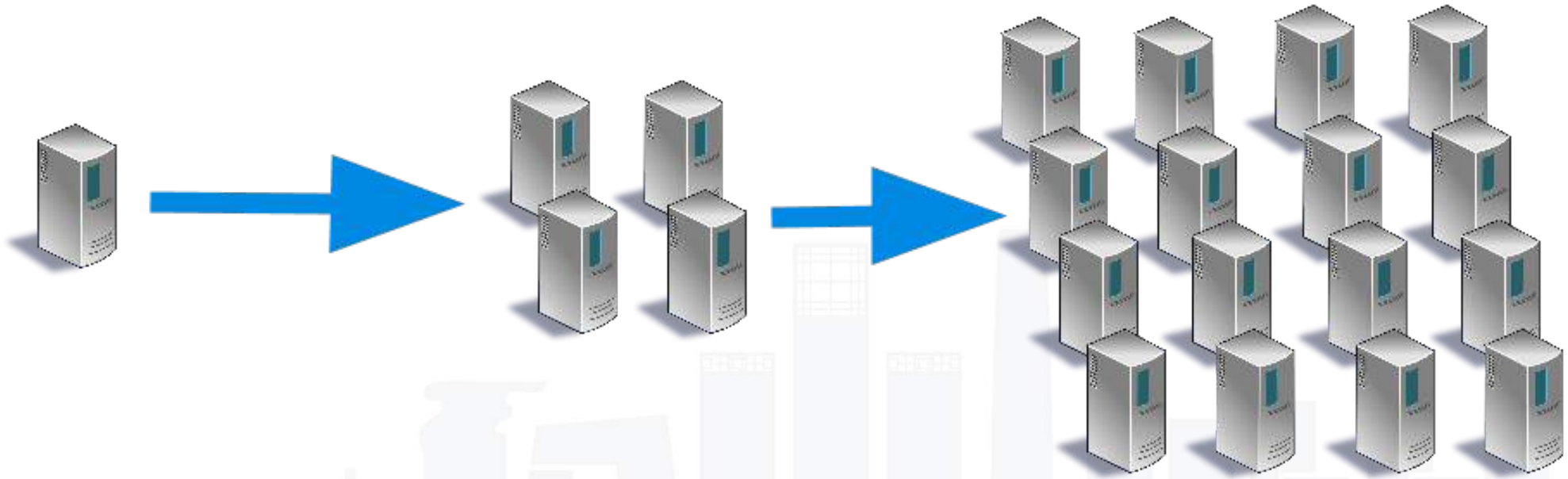
- Since 2015
- Beijing, Shanghai, Guangzhou, Silicon Valley, Home
- Open-Source infrastructure software: Codis, RebornDB
- TiDB, TiKV, TiSpark

Agenda

- Design Goals
- Architecture Evolution
- Lessons Learned
- Open Source Community

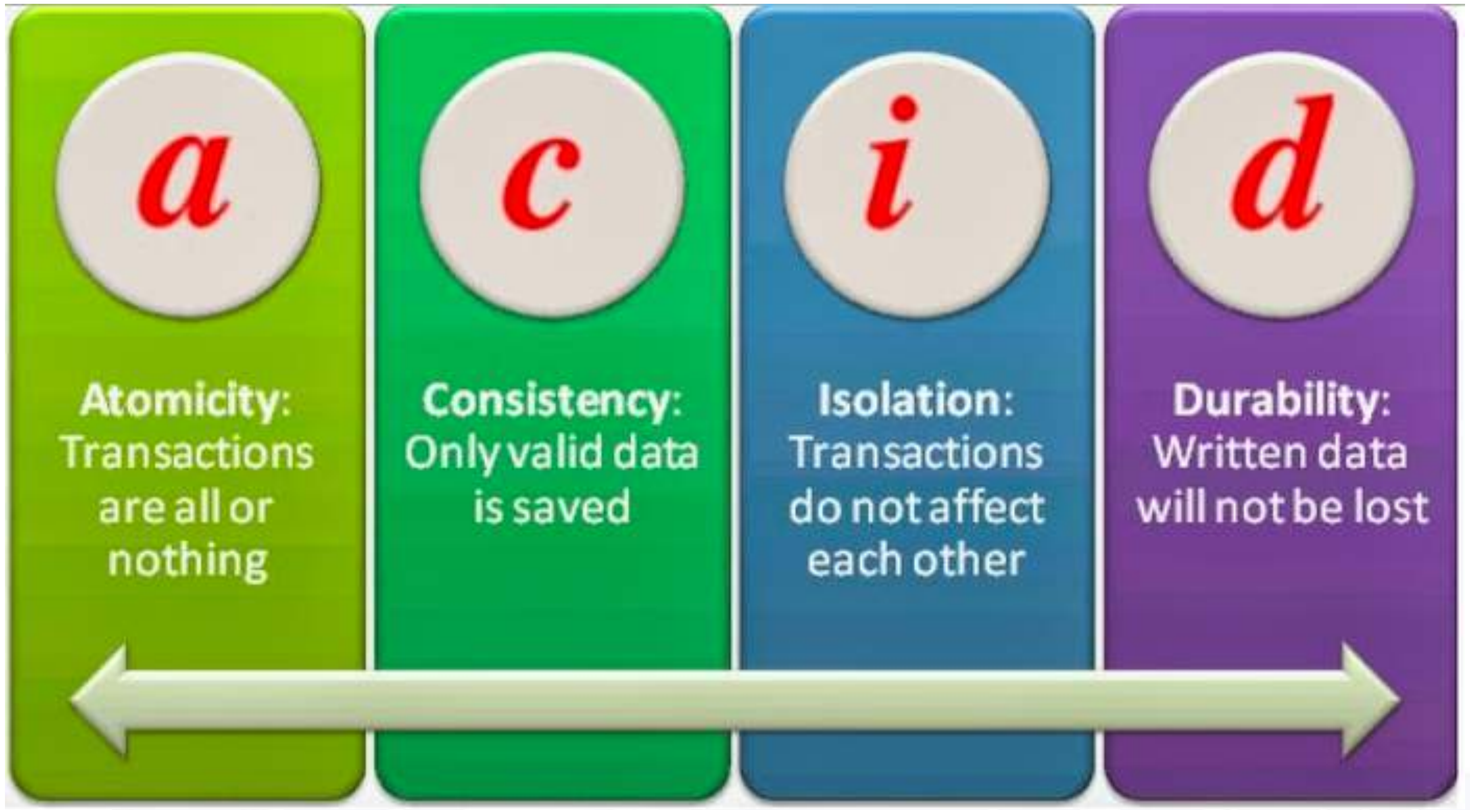
Design Goals

Goal 1: Horizontal Scalability



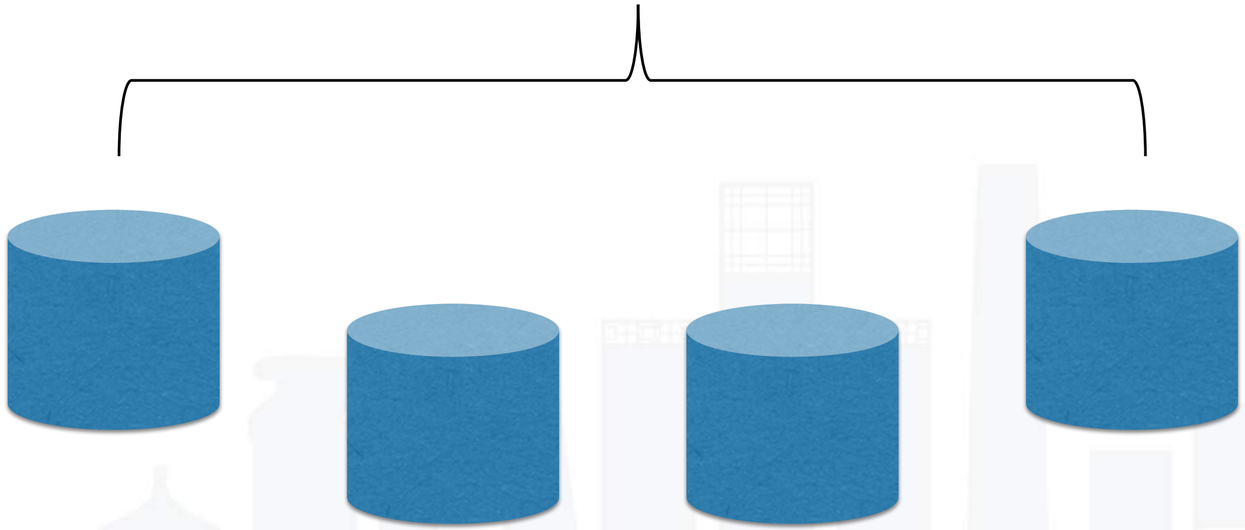
Goal 2: High Availability

Goal 3: ACID Transaction



SQL at Scale

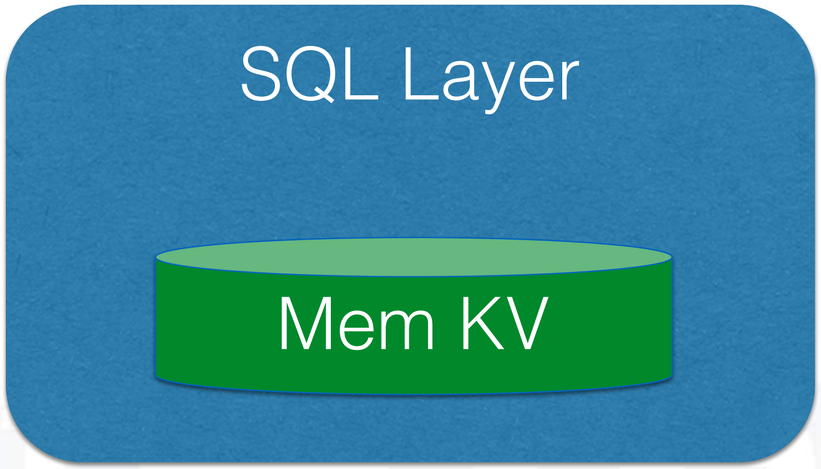
select count(*) from t;



Architecture Evolution

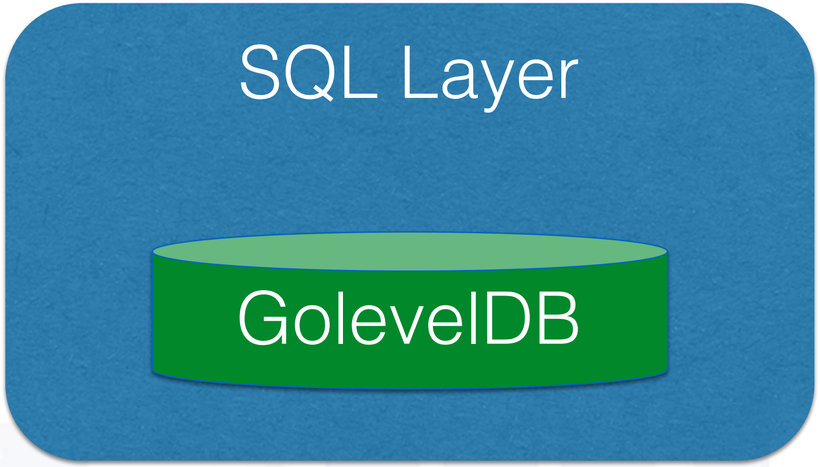
In ancient times

- Command-Line interface
- Basic CRUD
- AST ---> Executor

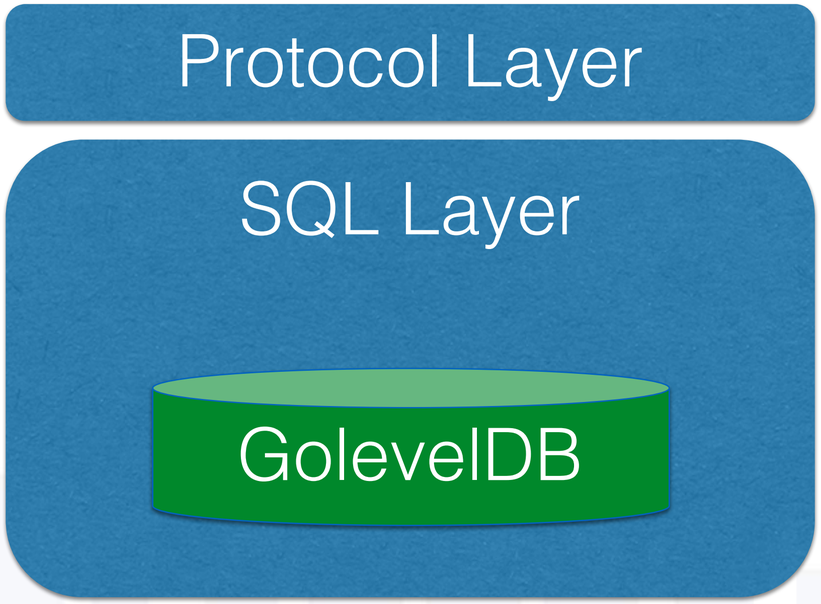


We get SQL and transaction!
But it is a toy.

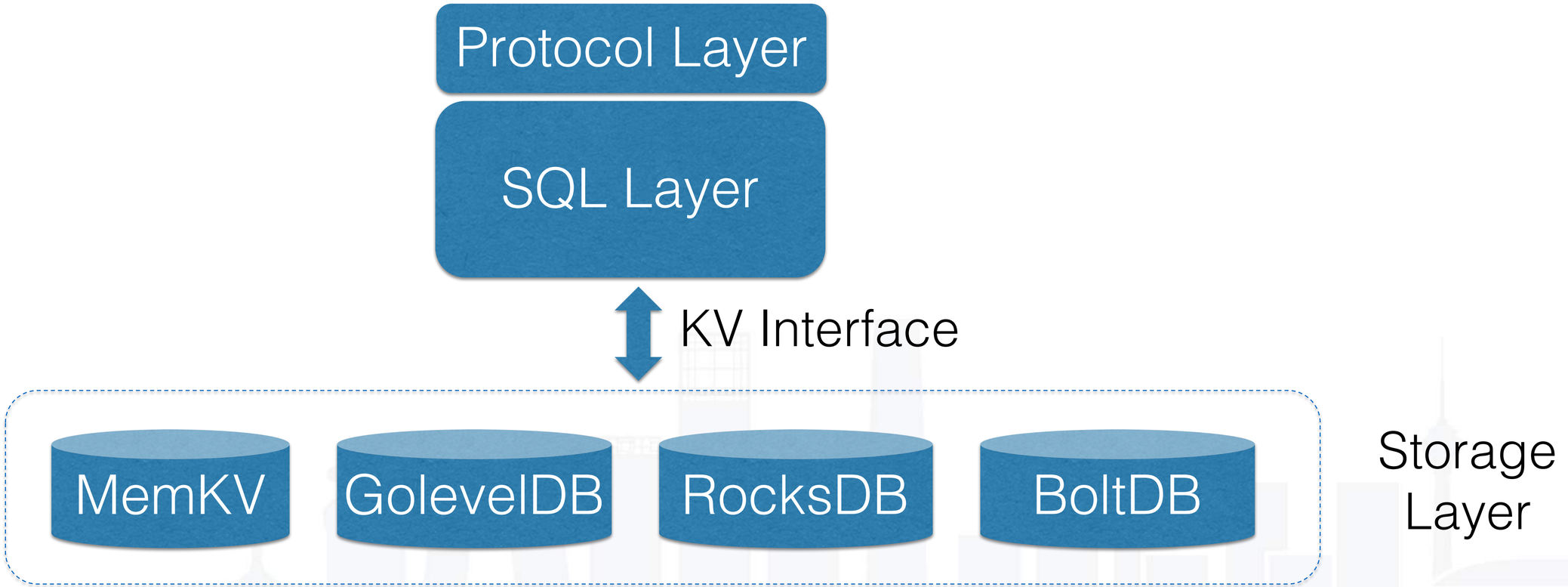
Persistent Storage



MySQL Protocol

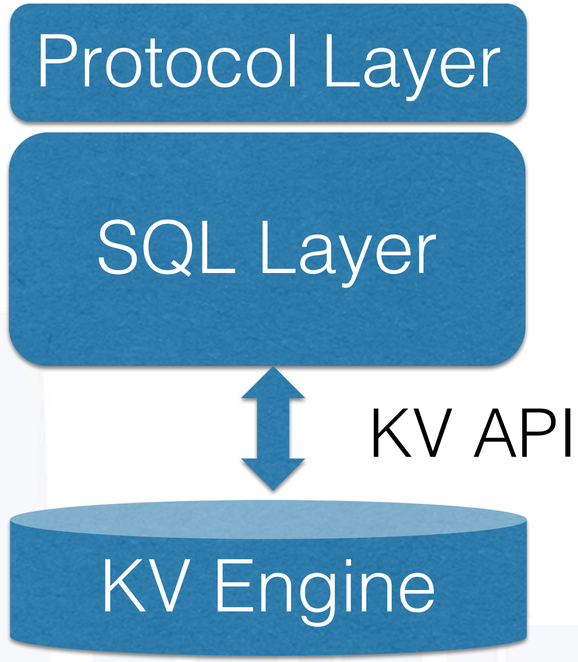


Pluggable storage engine



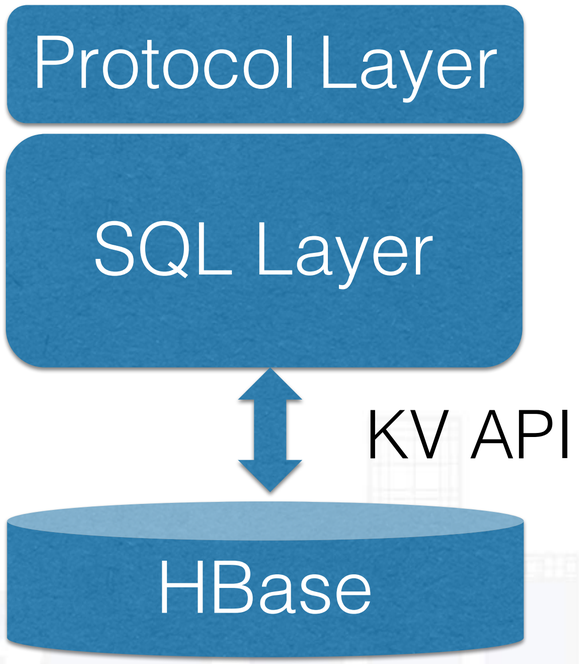
Better SQL Layer

- MySQL Protocol interface
- More DDL/DML
- More Data Types
- Simple Planner

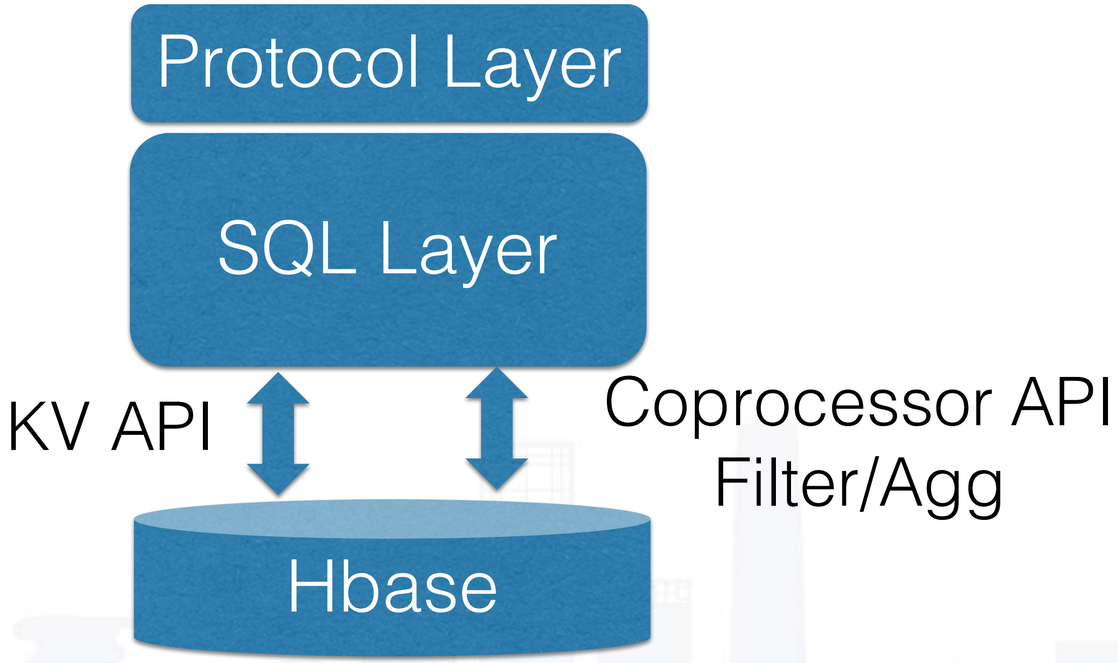


We get another MySQL.
But it is much weaker.

Distributed Storage Engine



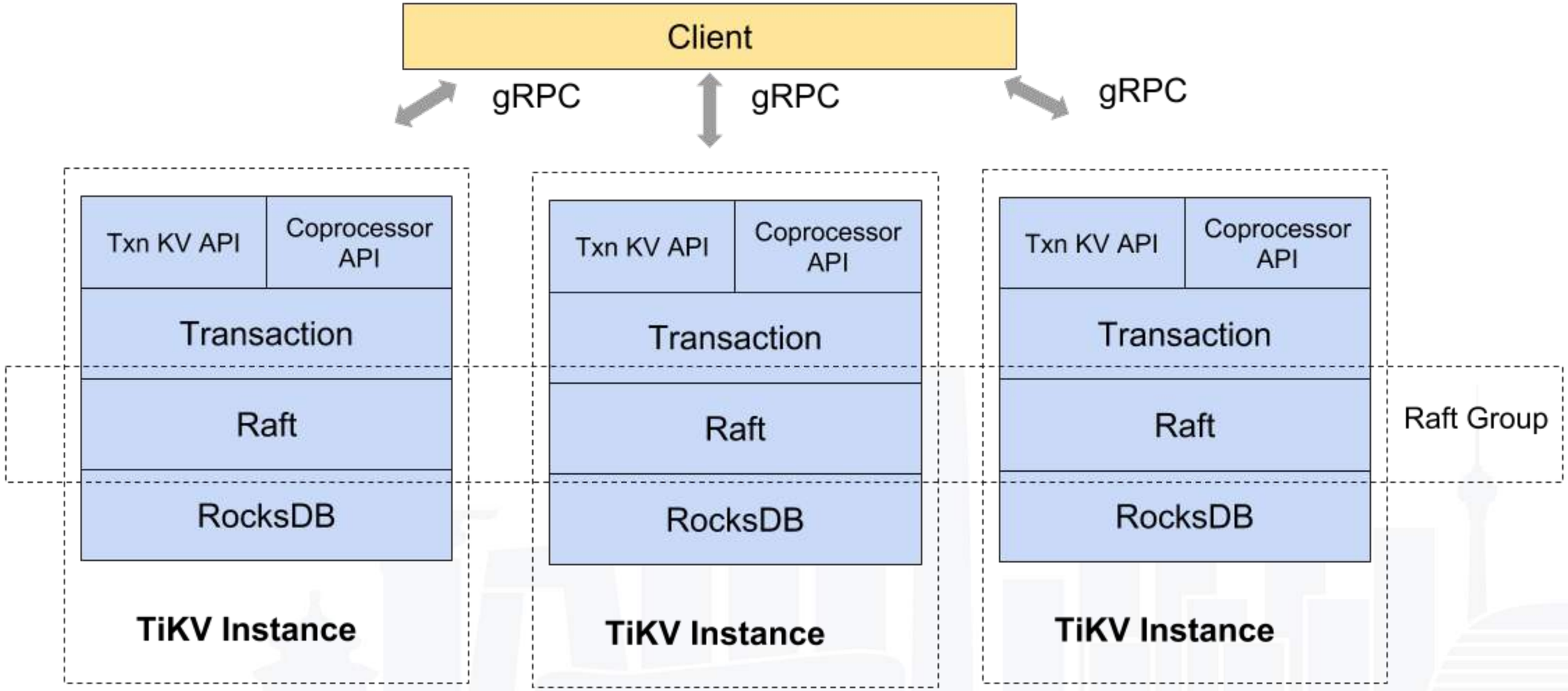
Operator Pushdown



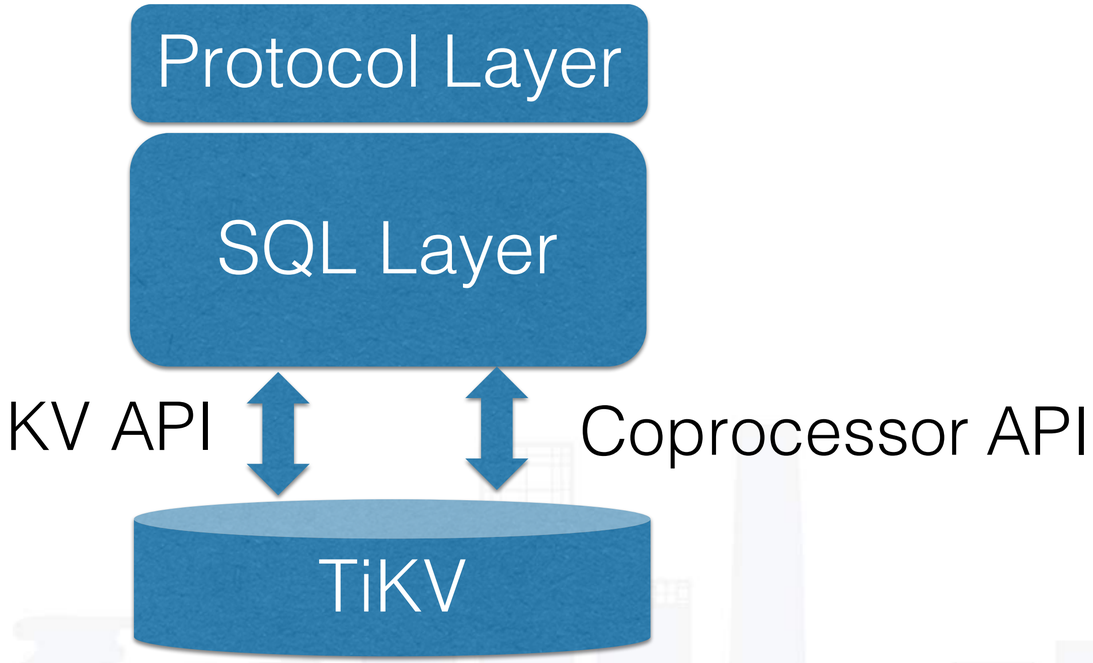
We get a distributed MySQL.
But it is too slow.

The birth of TiKV

TiKV Stack

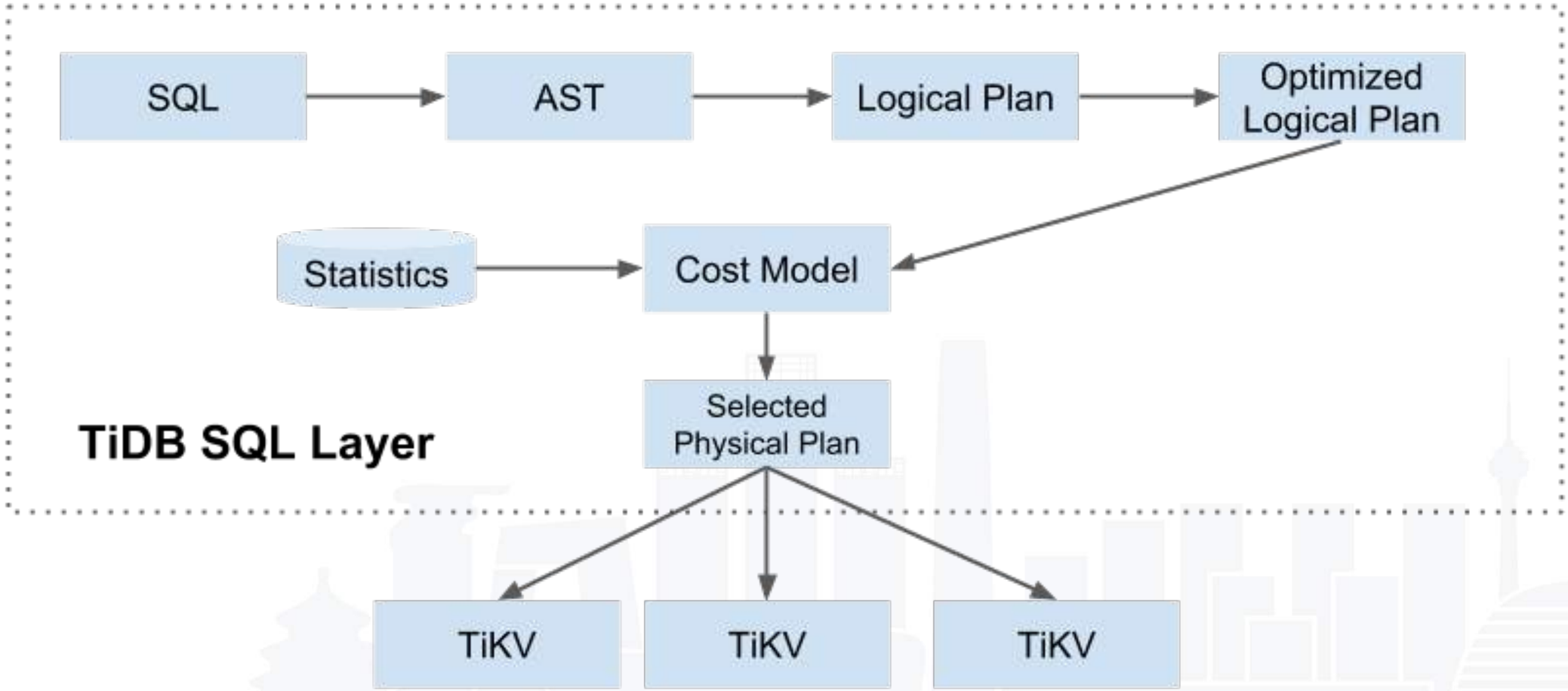

Client
gRPC
gRPC
gRPC
Txn KV API
Coprocessor API
Transaction
Raft
RocksDB
TiKV Instance
Txn KV API
Coprocessor API
Transaction
Raft
RocksDB
TiKV Instance
Txn KV API
Coprocessor API
Transaction
Raft
RocksDB
TiKV Instance
Raft Group

TiDB + TiKV

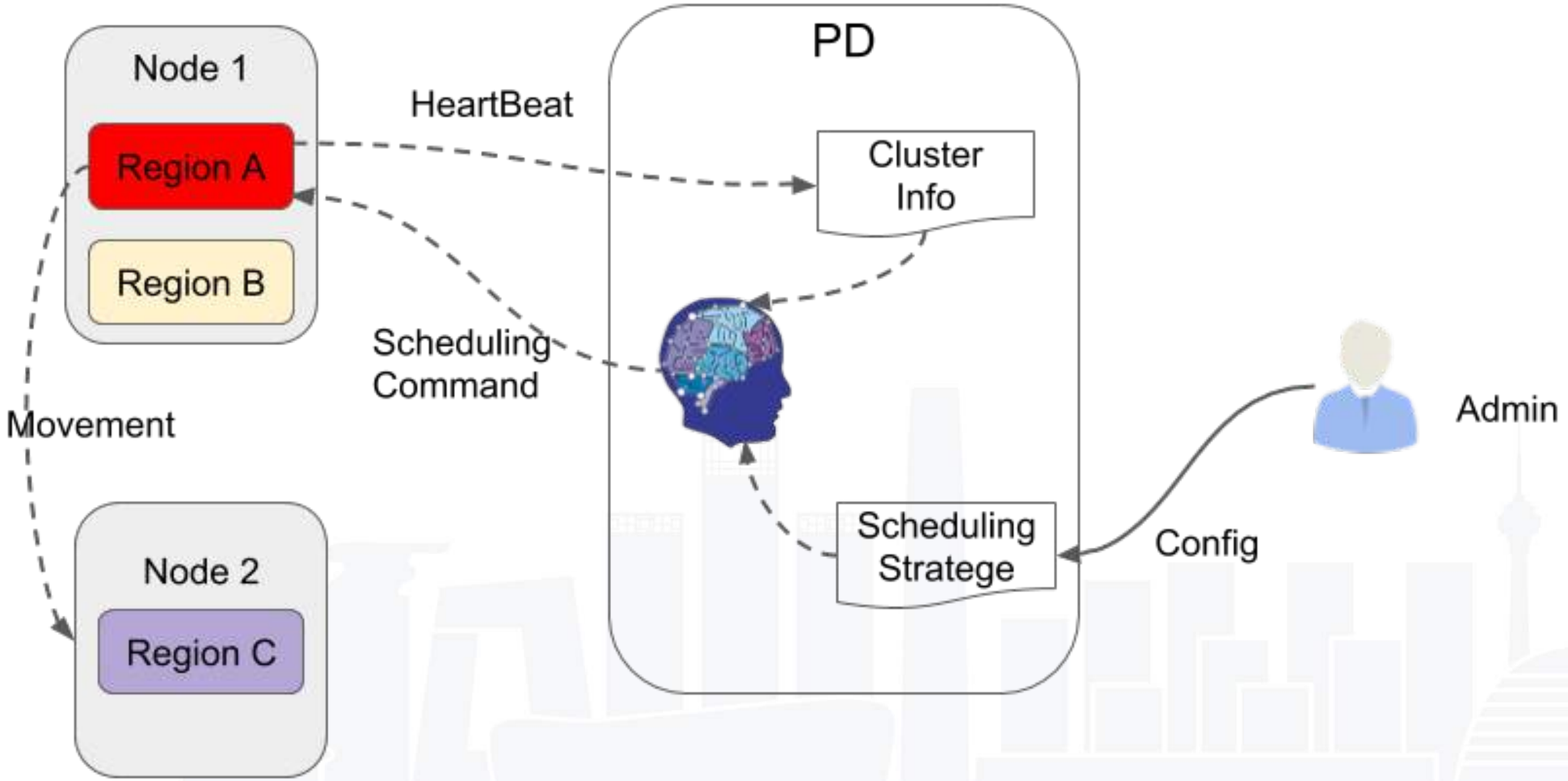


We get another distributed MySQL.
But it is not good enough.

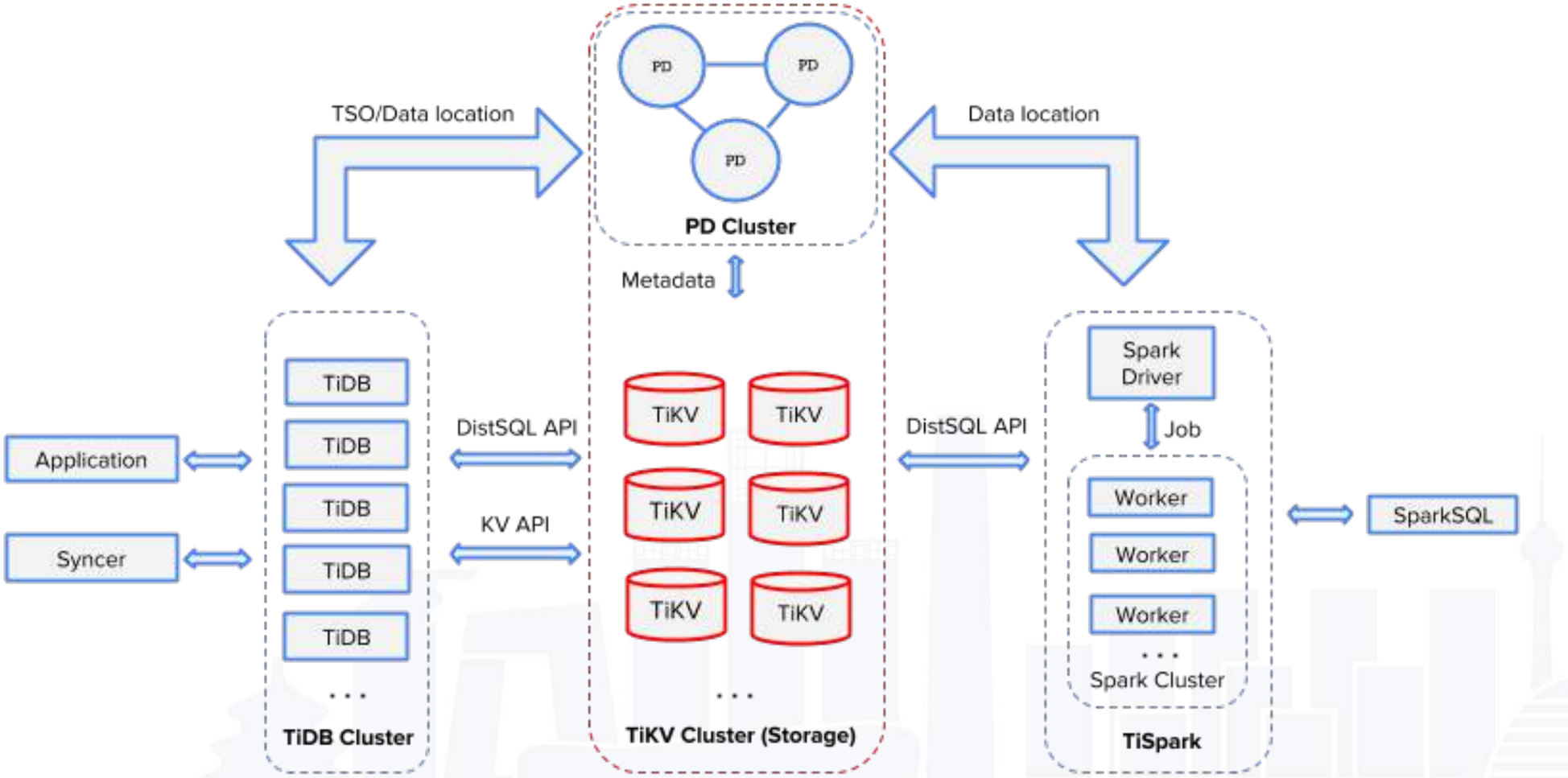
Better SQL Layer



Better KV Engine



The whole picture



Lessons Learned

Make it right, then make it fast