Introduction to TiDB

Mattias Jonsson, PingCAP

Athens, Greece 2023

PERCONA UNIVERSITY

In partnership with FerretDB

Safe Harbor Statement

This presentation has been prepared for general informational purposes only. All information contained in this presentation is provided in good faith, however we make no representation or warranty of any kind, express or implied, regarding the accuracy, adequacy, completeness of any information. The information may not be incorporated into any contract. The development, release and timing of any features or functionality described for PingCAP products remains at the sole discretion of PingCAP.

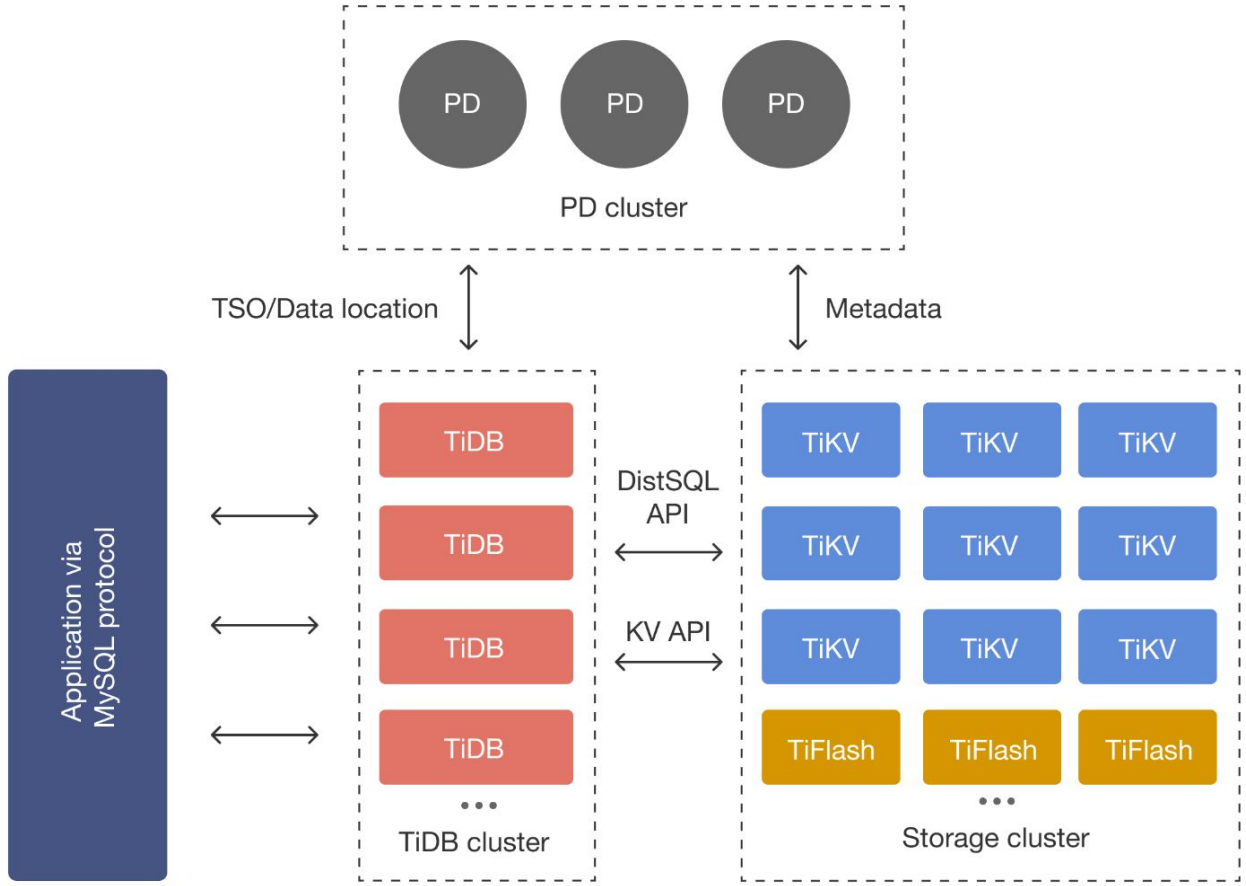
Introduction

Mattias Jonsson

- Working for PingCAP as Senior Database Engineer, developing TiDB
- Previous:
  - Senior Developer / Engineering manager at Booking.com
  - Senior Software Engineer at MySQL/Sun/Oracle

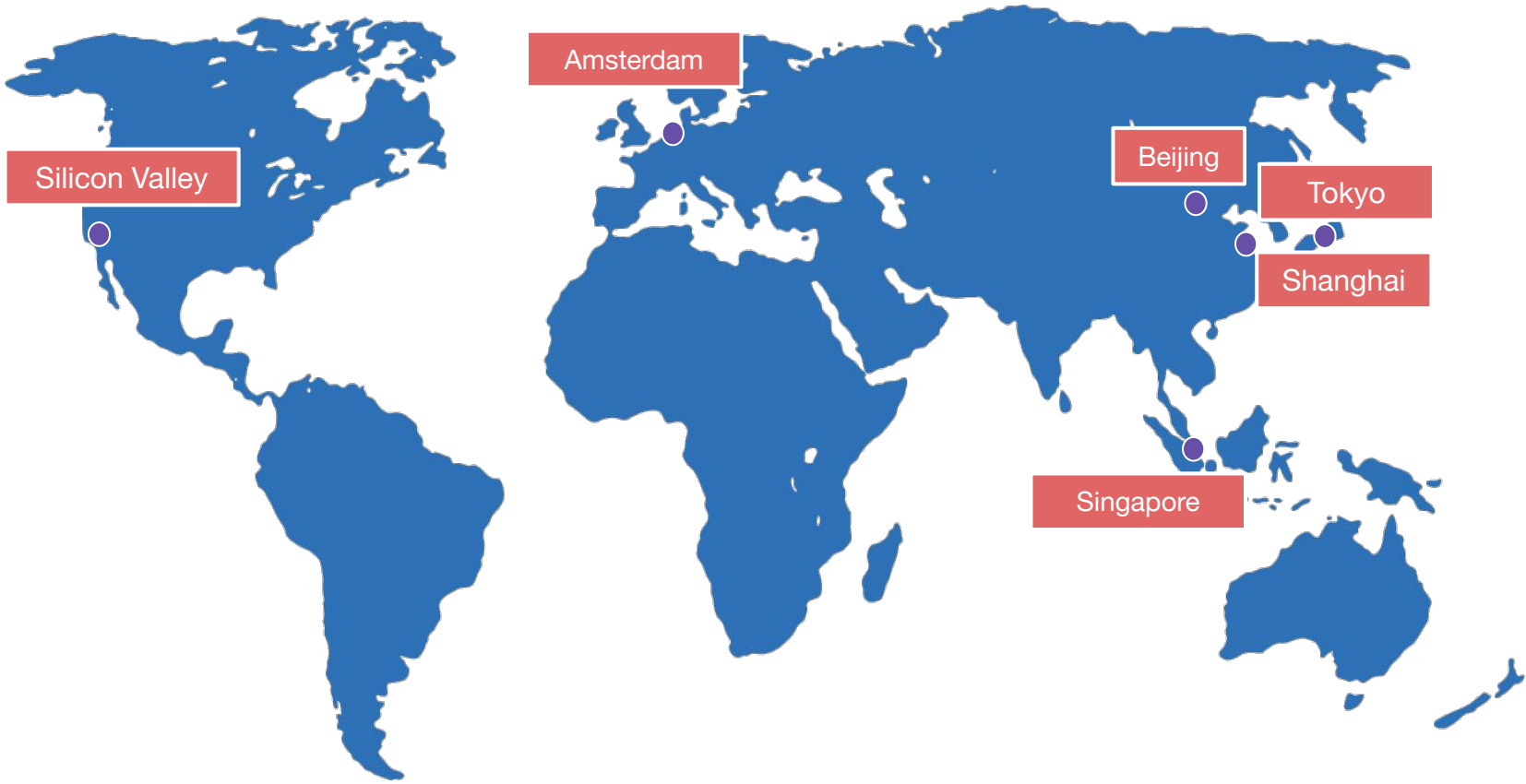
tl;dr

- Distributed SQL database
- Looks like a MySQL primary, no changes in application
- Scale as simple as adding more machines
- Database-as-a-Service or deploy on cloud or on-prem.
- Open source, Apache 2.0
- Optional analytics / columns store
- Migration tools
- Change Data Capture to Kafka/MySQL/TiDB



About PingCAP

- The company behind TiDB
- Founded in 2015
- Global
- Open source culture
- Strong investors
- 600+ employees

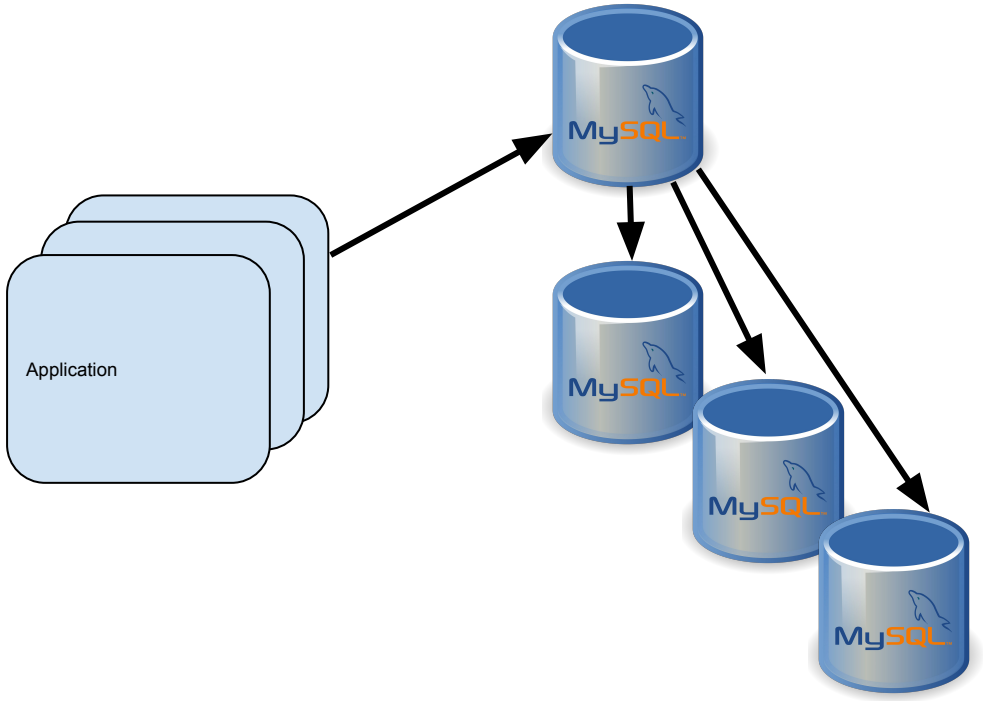


High availability with MySQL

Logical replications included since early days, so one primary and one or more replicas.

Then if the primary fails you promote a replica. How?

Not automated.
No leadership election.
Are the replicas up to date?
Service discovery?
Rejoining a recovered node?

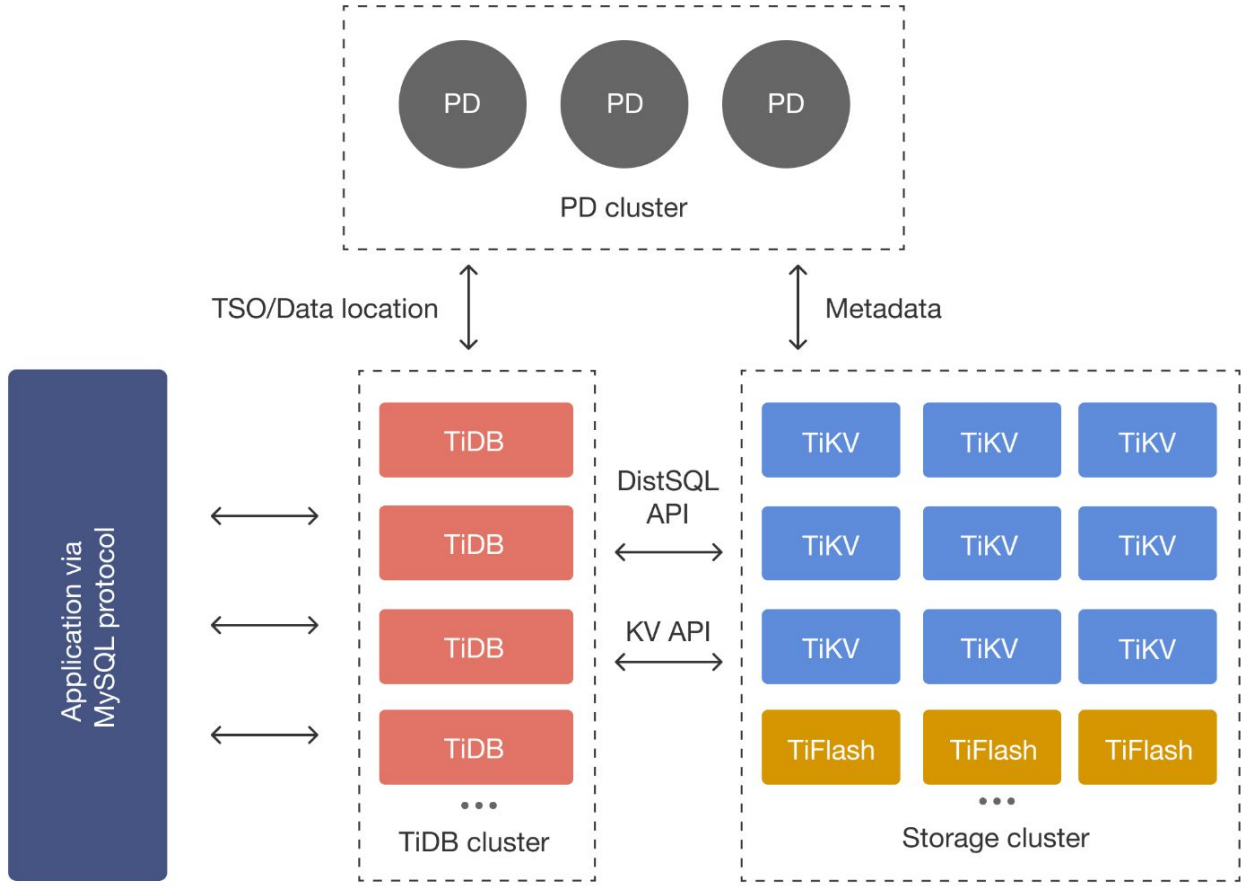


High availability with TiDB

All components are redundant.

Data are distributed and stored in smaller ranges of around 100MB ranges with copies in 3 different AZ.

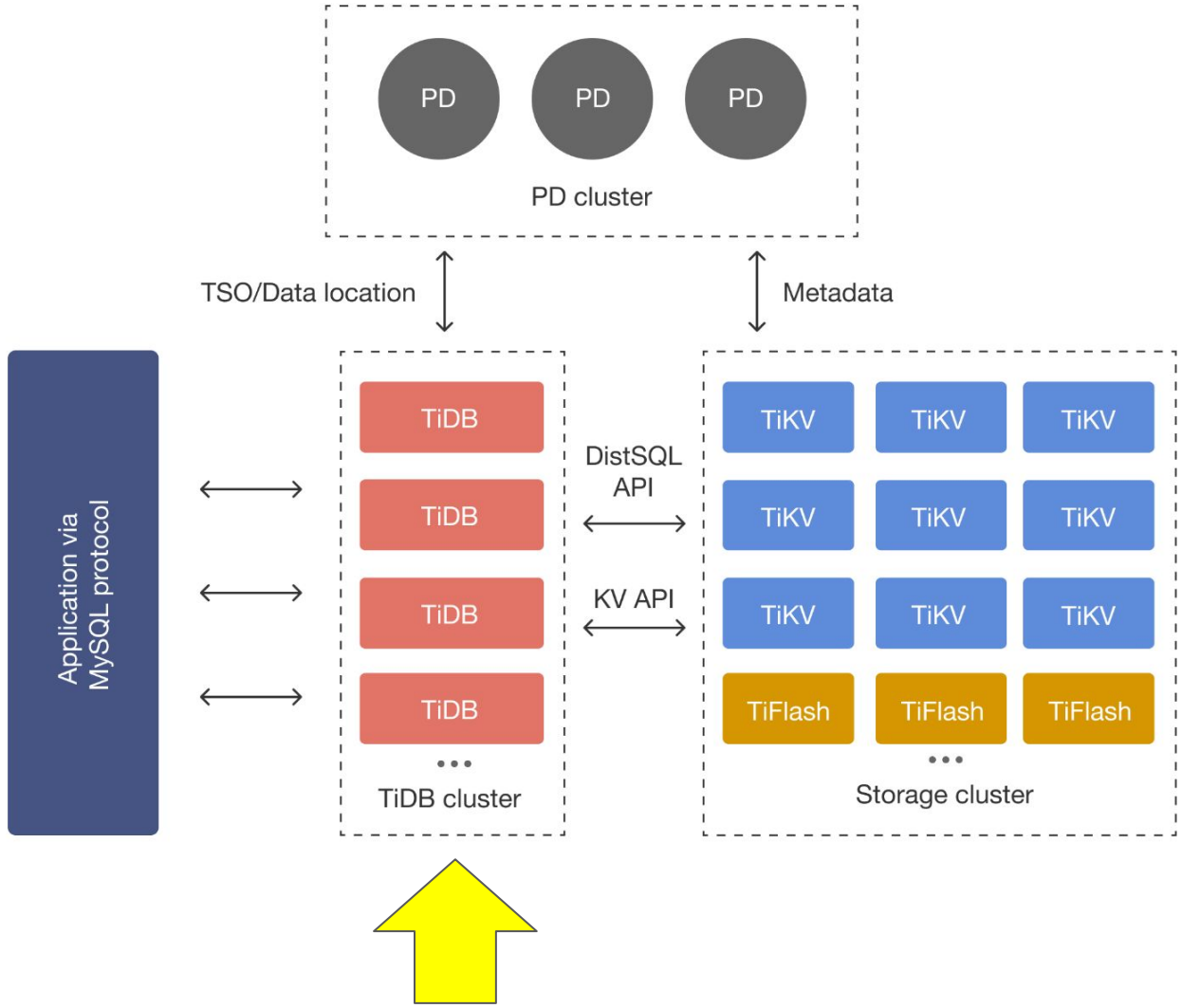
Rolling upgrades.



TiDB Server

- MySQL protocol
- Parser
- Optimizer
- Executor.

Stateless



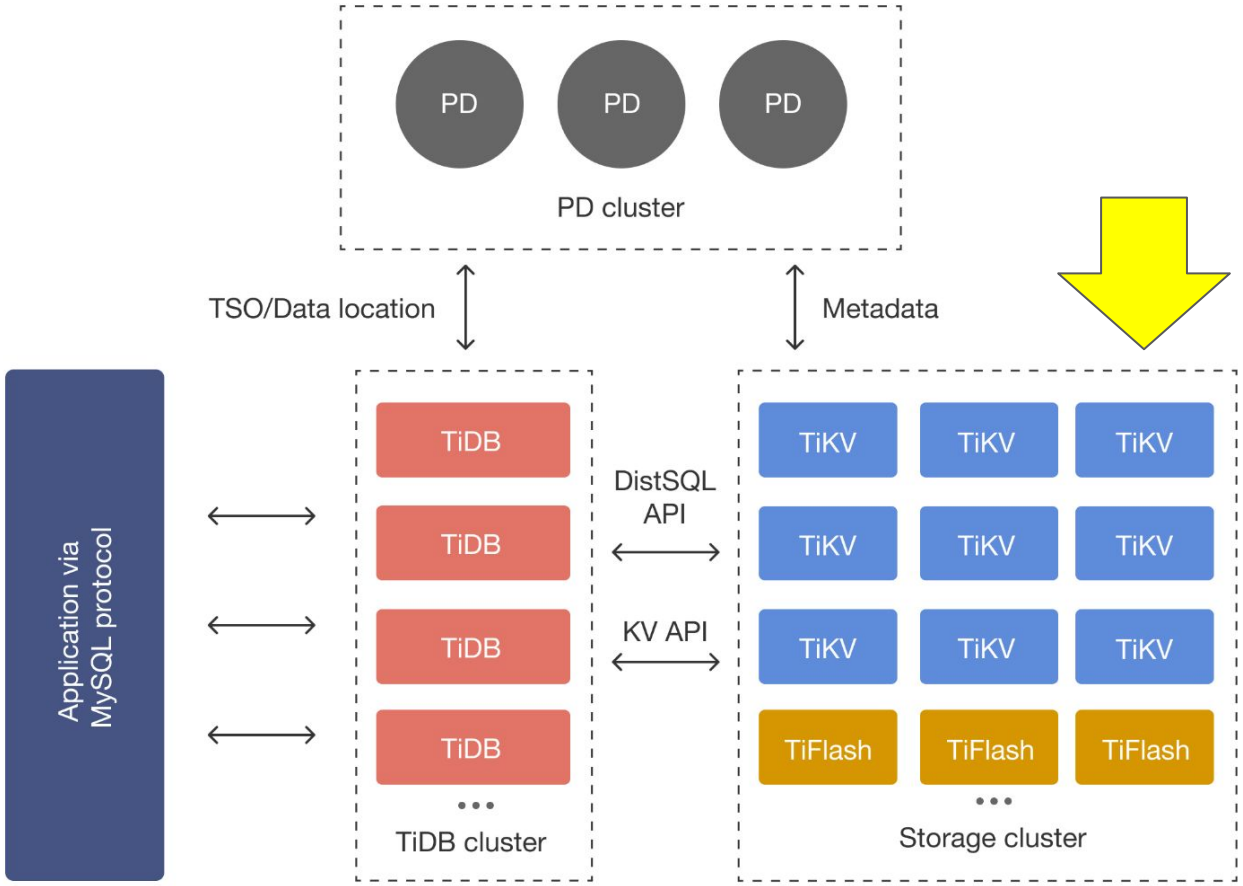
TiKV

TiKV is a distributed key-value store. This is a CNCF project.

Data is mapped to key/values
- <tableID,rowID/PK> -> <col1, col2…>

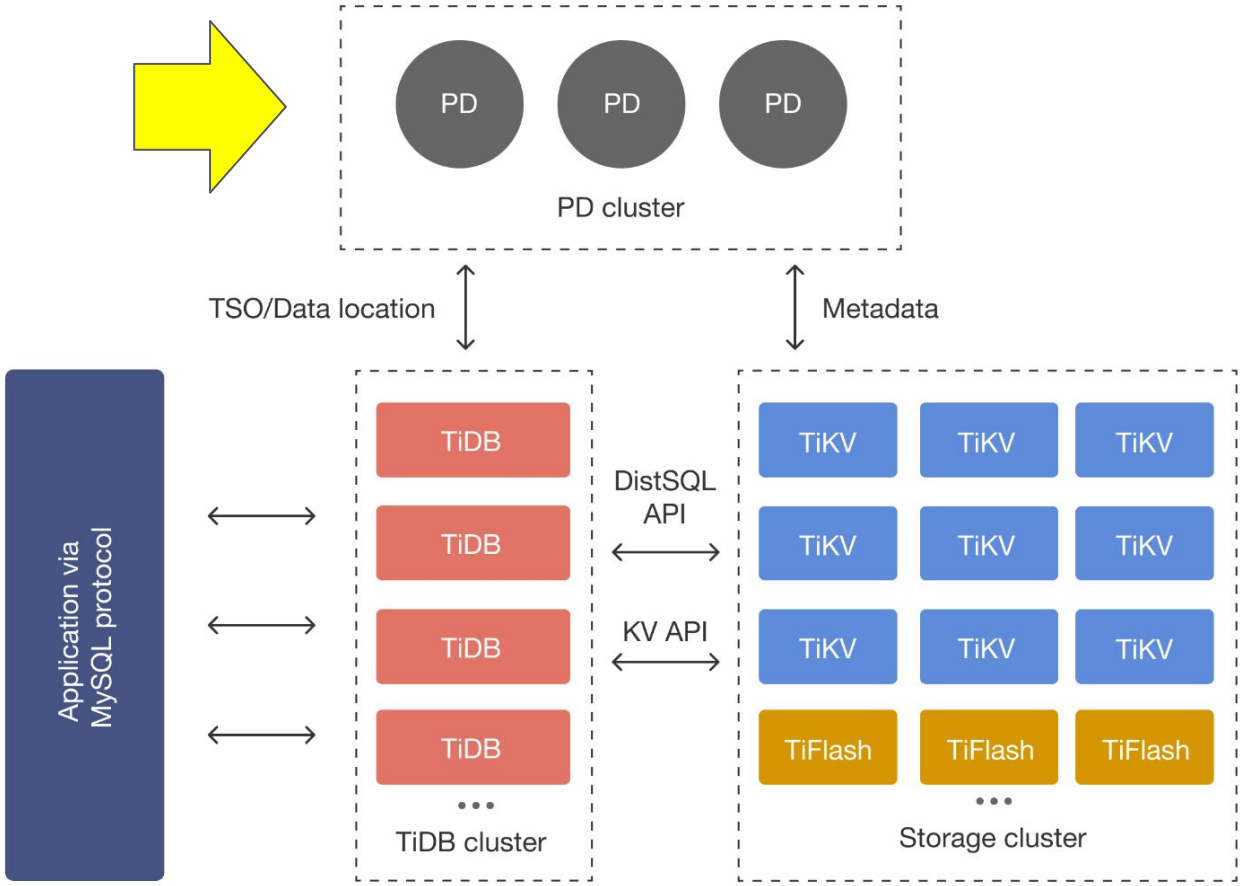
Data is split by ranges ~100MB in size
Each range forms a RAFT group keeping 3 copies of the data in sync in different AZ.

Physical storage is based on RocksDB.



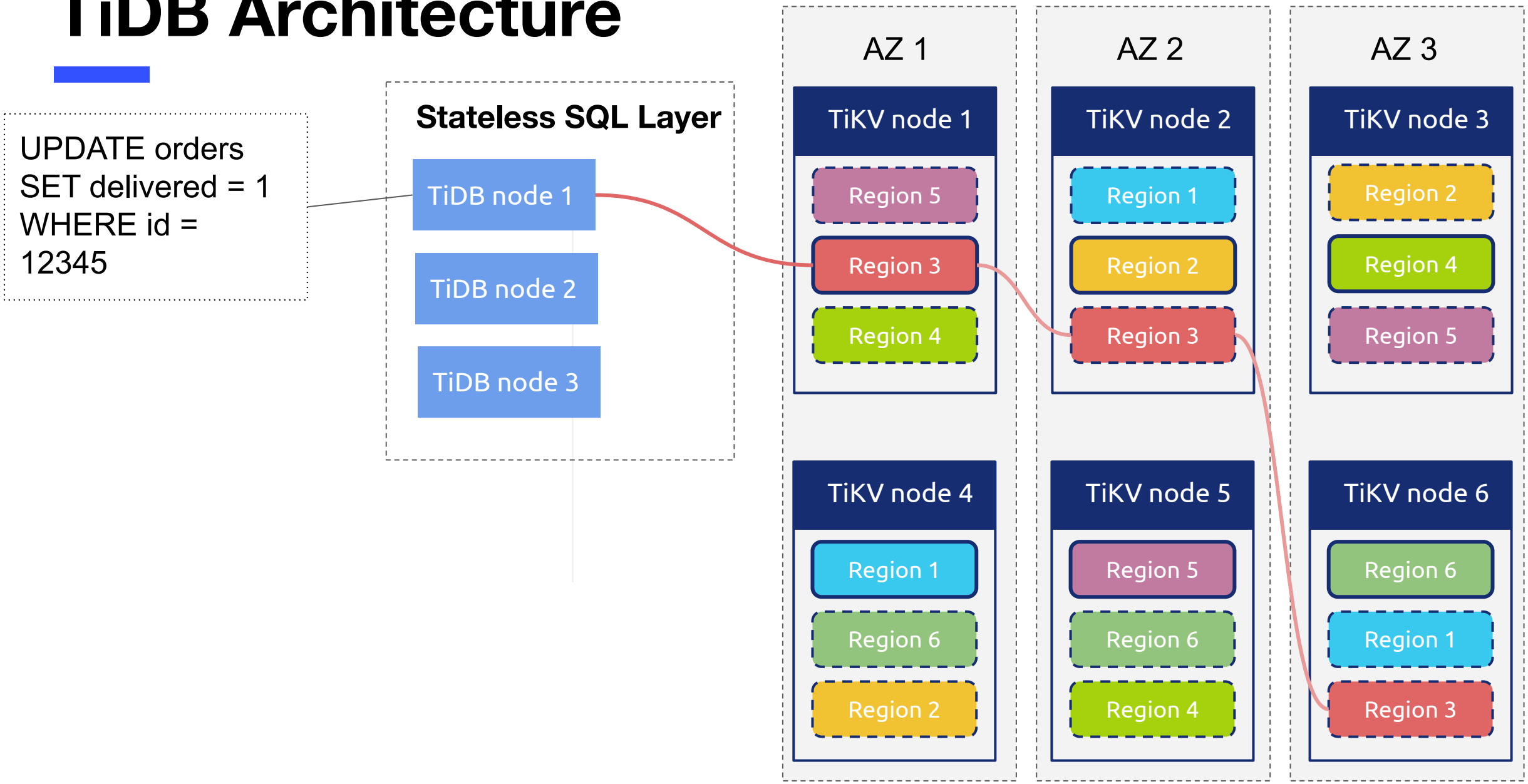
Placement Driver (PD)

- Time Stamp Oracle (TSO) for MVCC
- Cluster metadata
- Data placement:
  - One copy per AZ
  - Split and moves data ranges between nodes
    - Size (split and merge)
    - Load (both read and write)



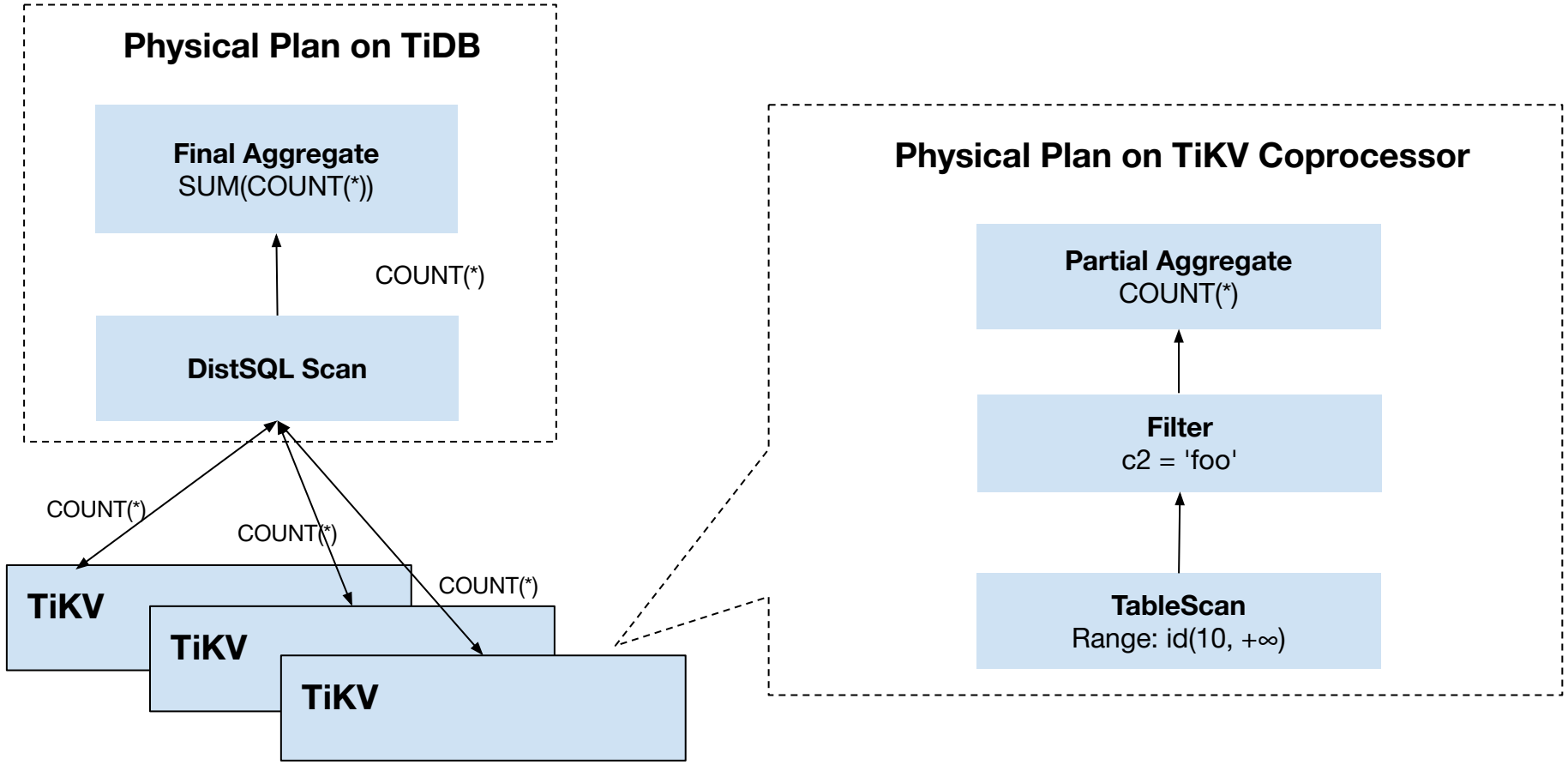
Result is even distribution of both data and load!

TiDB Architecture



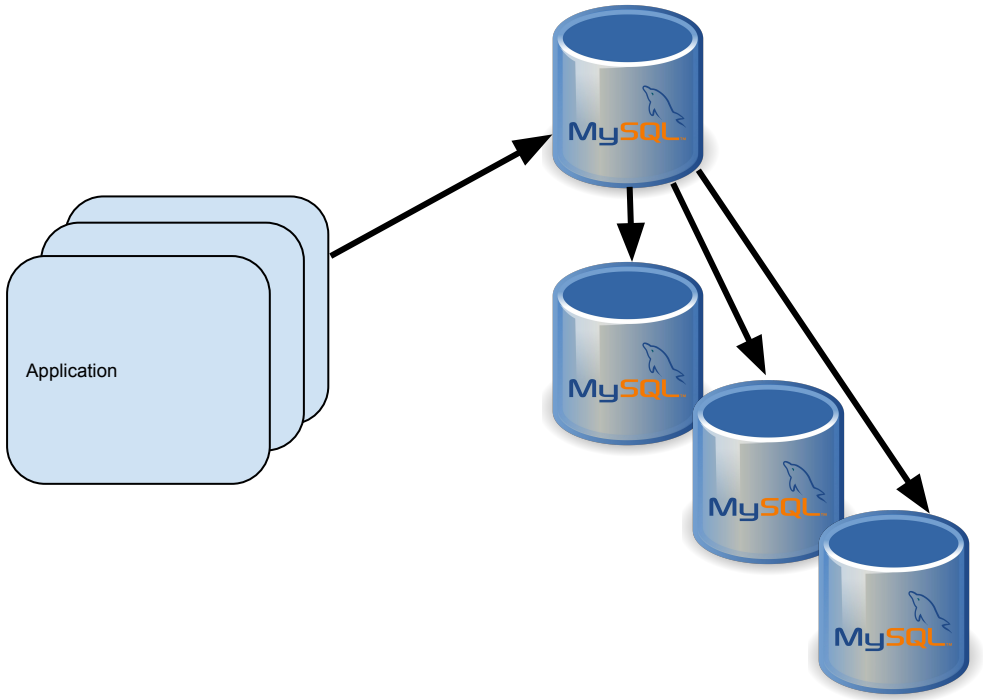
Query Execution/Distributed Computing

```
SELECT COUNT(*) FROM t WHERE id > 10 AND c2 = 'foo';
```



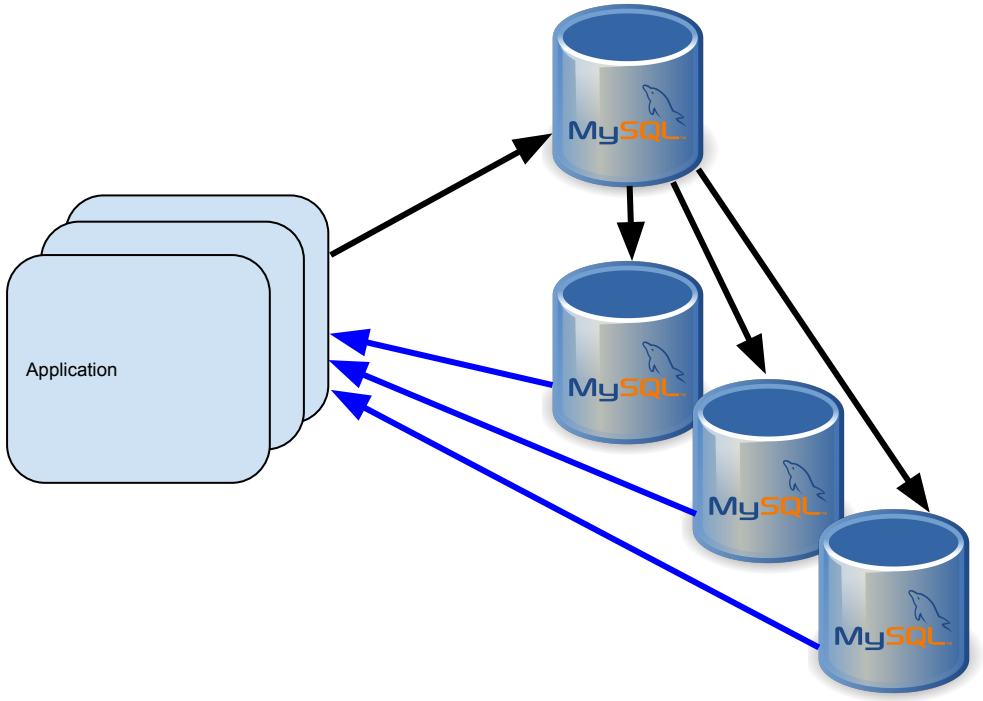
Scalability with MySQL

With MySQL all writes goes first to the primary, then replicated to all others.
All nodes store a complete copy of the data and applies all the changes.



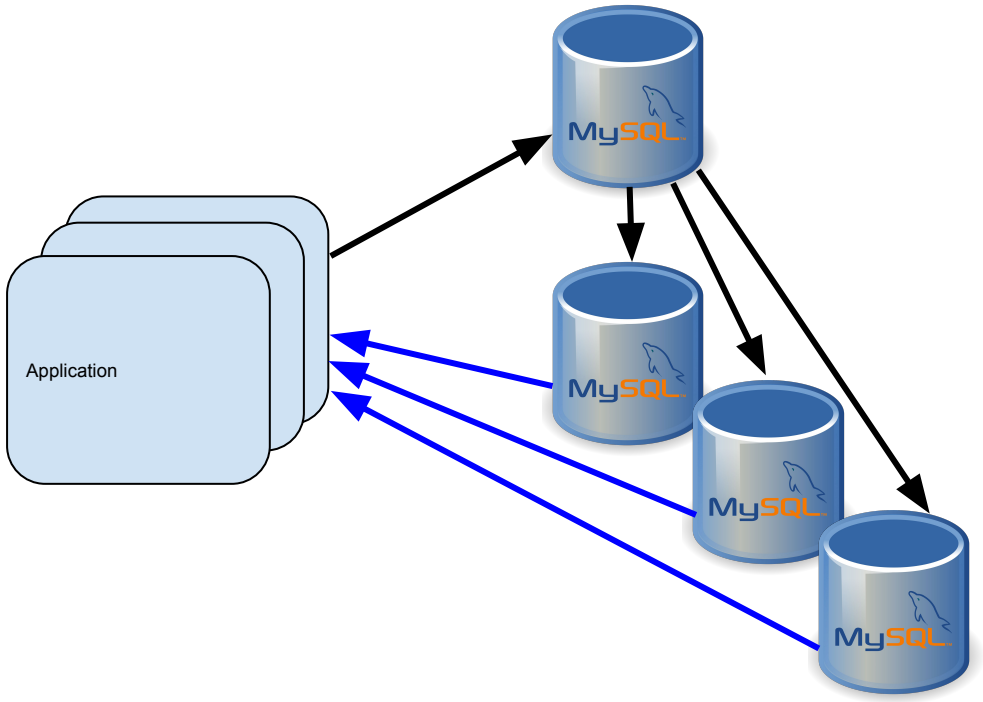
Read scalability with MySQL

Requirement: Split reads and writes in the application or use a smart proxy and be aware of replication delay
Then: Add more replicas



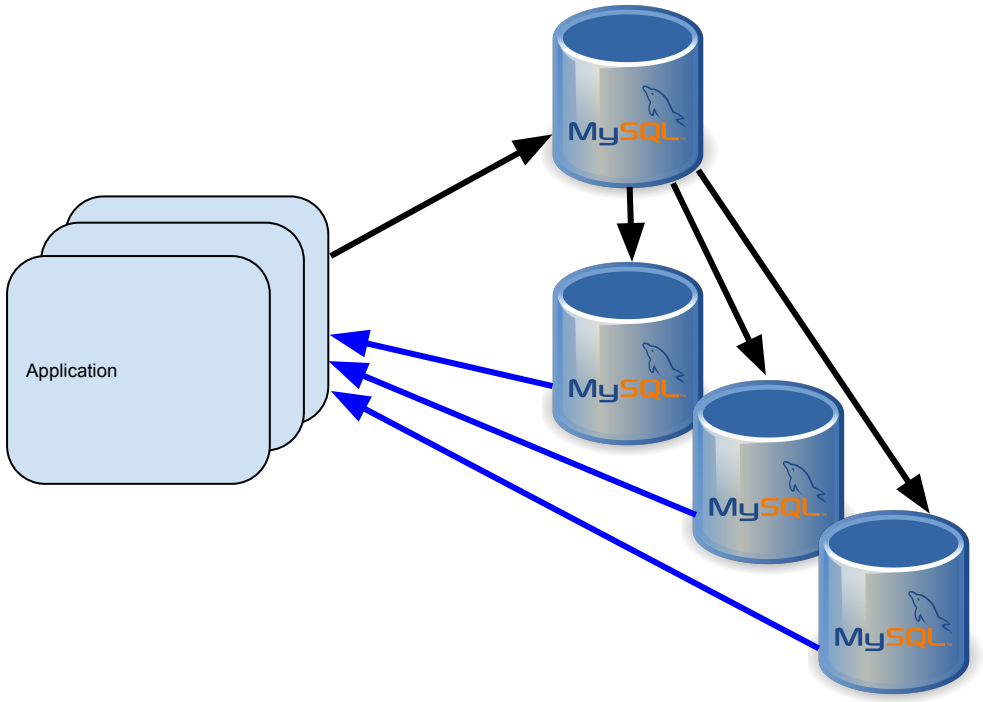
Write scalability with MySQL

Bigger machines for all instances, primary and all replicas!



Data size scalability with MySQL

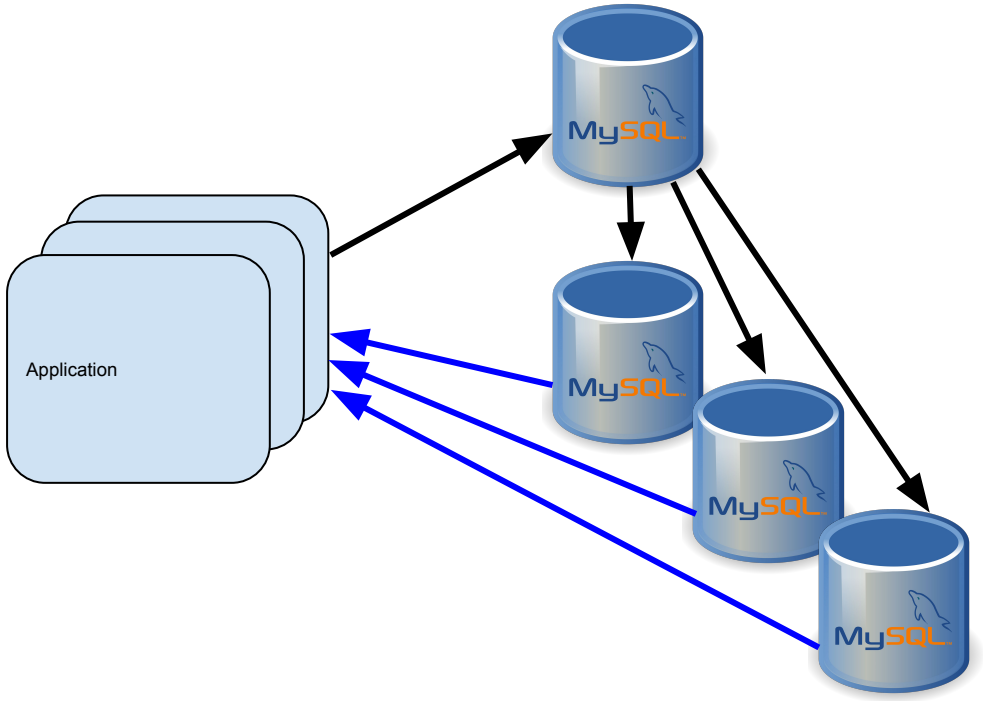
Bigger disks to all instances!



More scalability with MySQL, Sharding?

Need to scale more?

Shard on the application side and implement more DB logic in the application…



Scalability with TiDB

Both reads and writes can go to any of the TiDB nodes.

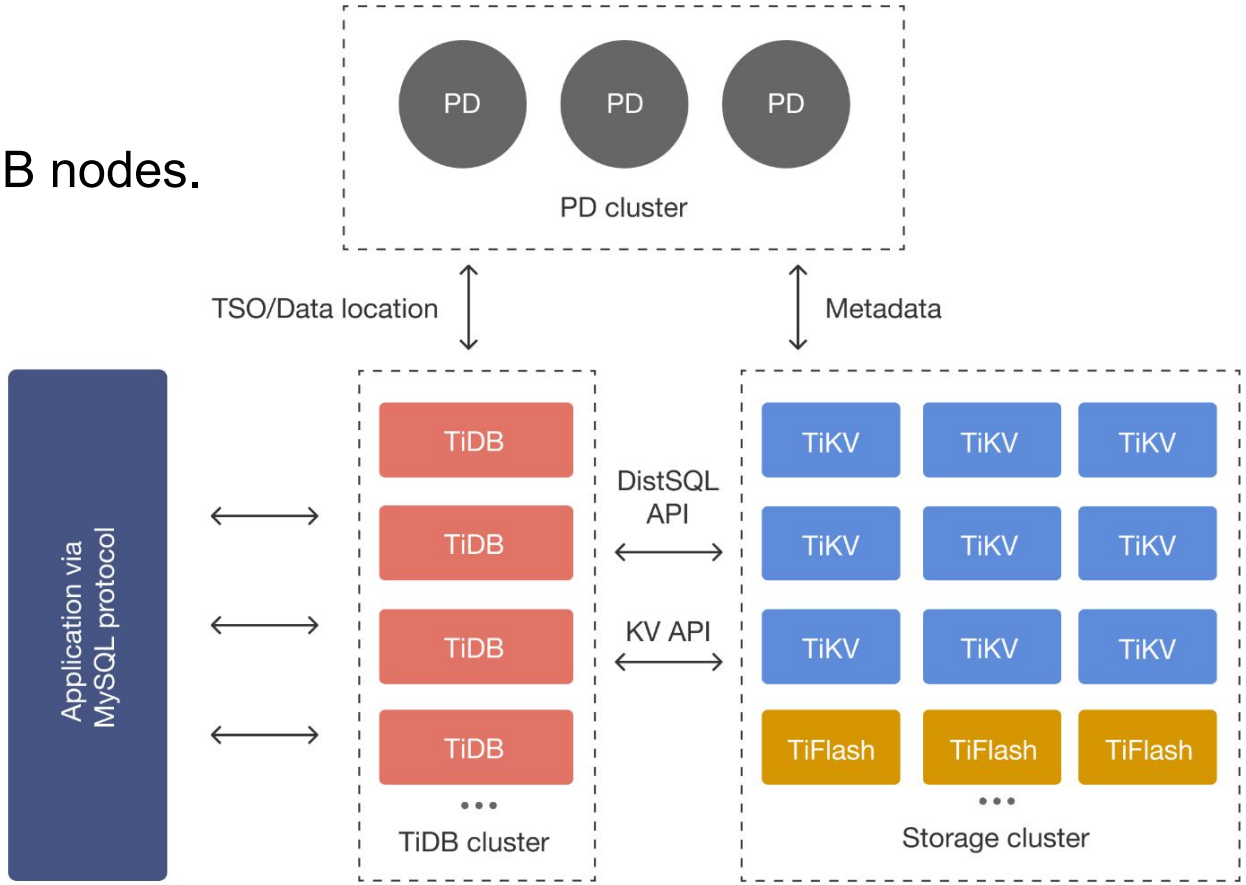
Scaling reads?
Add more nodes

Scaling writes?
Add more nodes

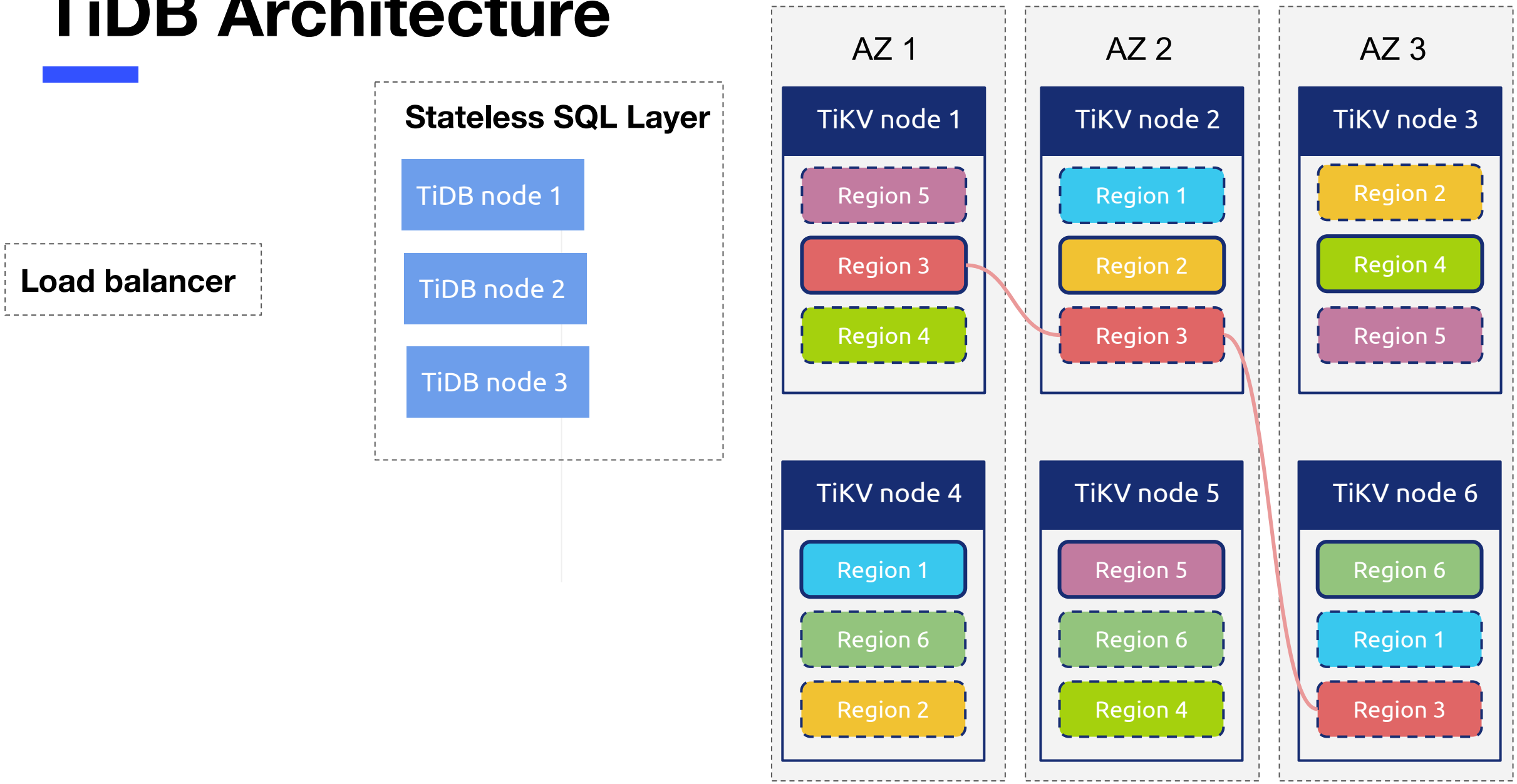
Scaling data volume?
Add more nodes

Need to scale more?
Add more nodes

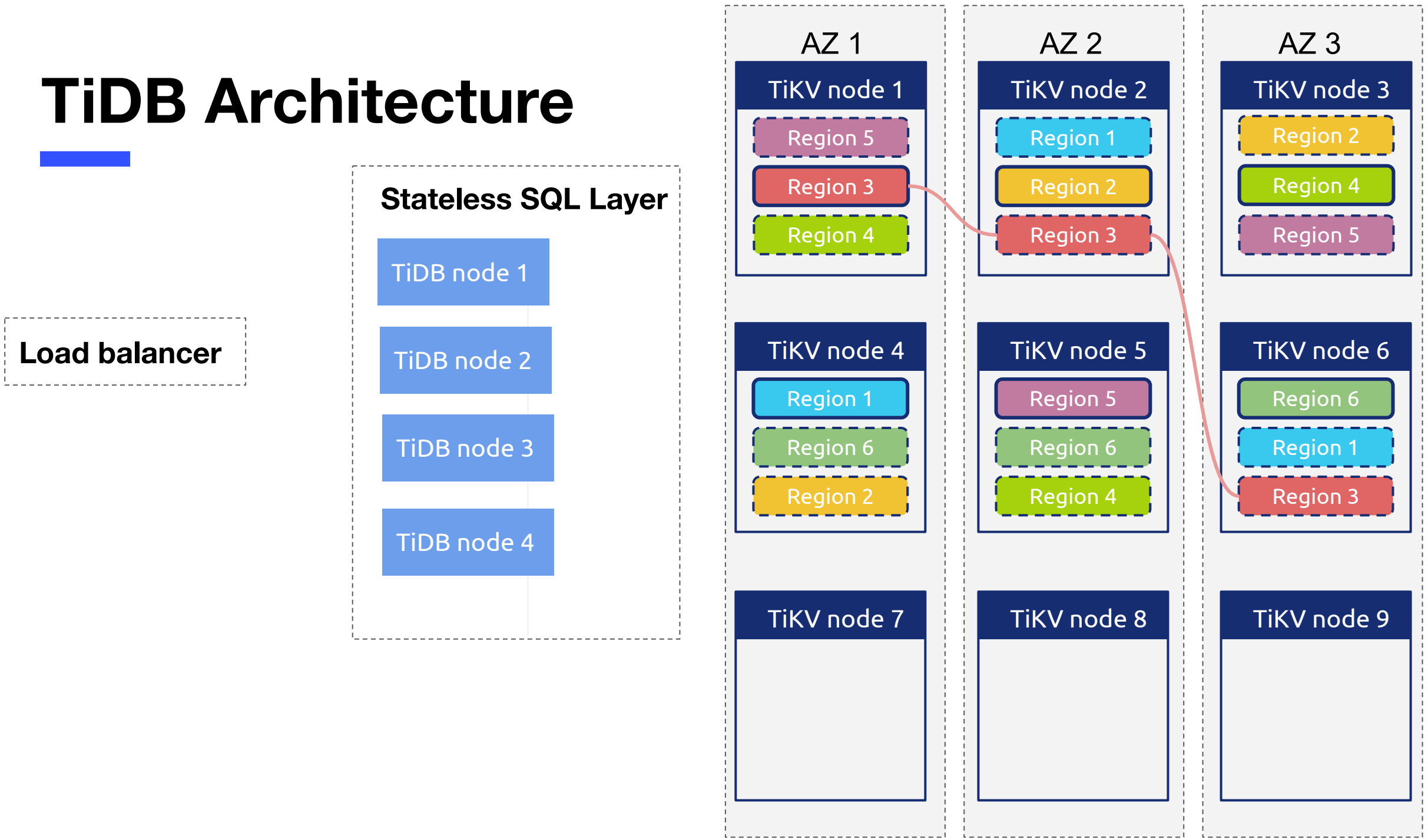
No need for read/write splitting OR explicit sharding!



TiDB Architecture



TiDB Architecture

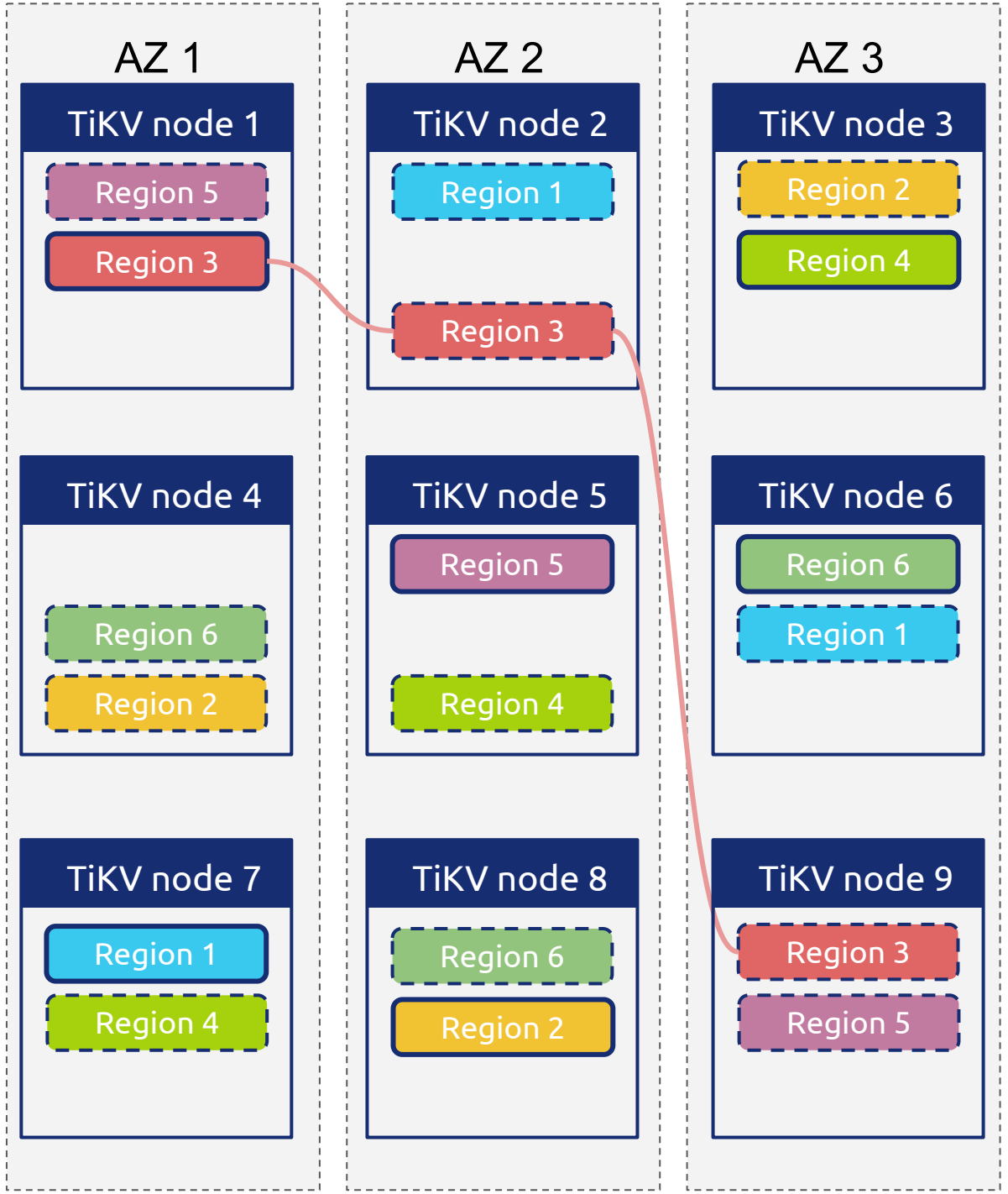


TiDB Architecture

Load balancer

Stateless SQL Layer

- TiDB node 1
- TiDB node 2
- TiDB node 3
- TiDB node 4



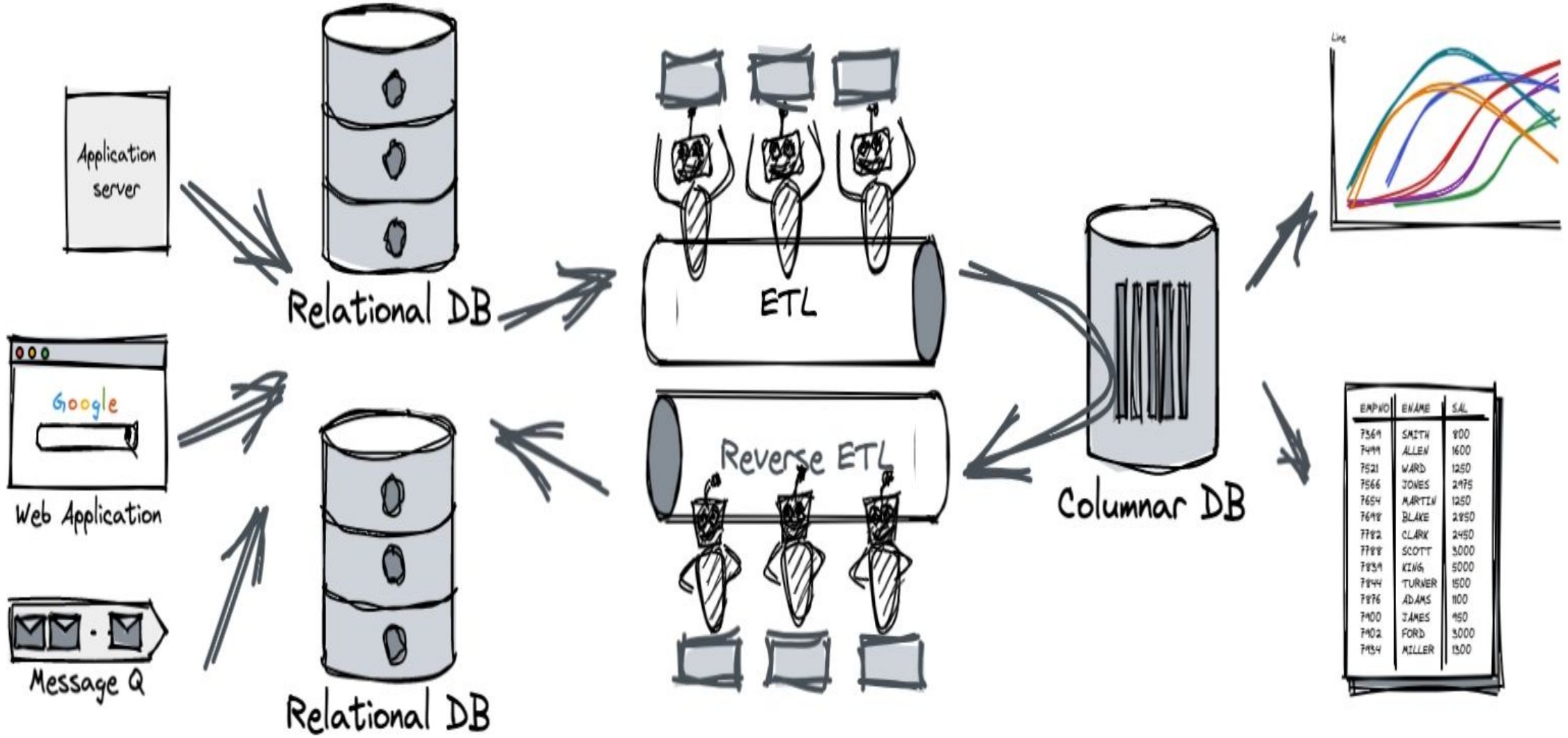
Deployment options

- Database as a Service
  - TiDB Cloud Dedicated cluster
  - TiDB Serverless
  - ➢ AWS
  - ➢ GCP
- Self-Deploy and manage
  - TiDB Enterprise Edition
  - TiDB Community Edition
  - ➢ kubernetes
  - ➢ tiup

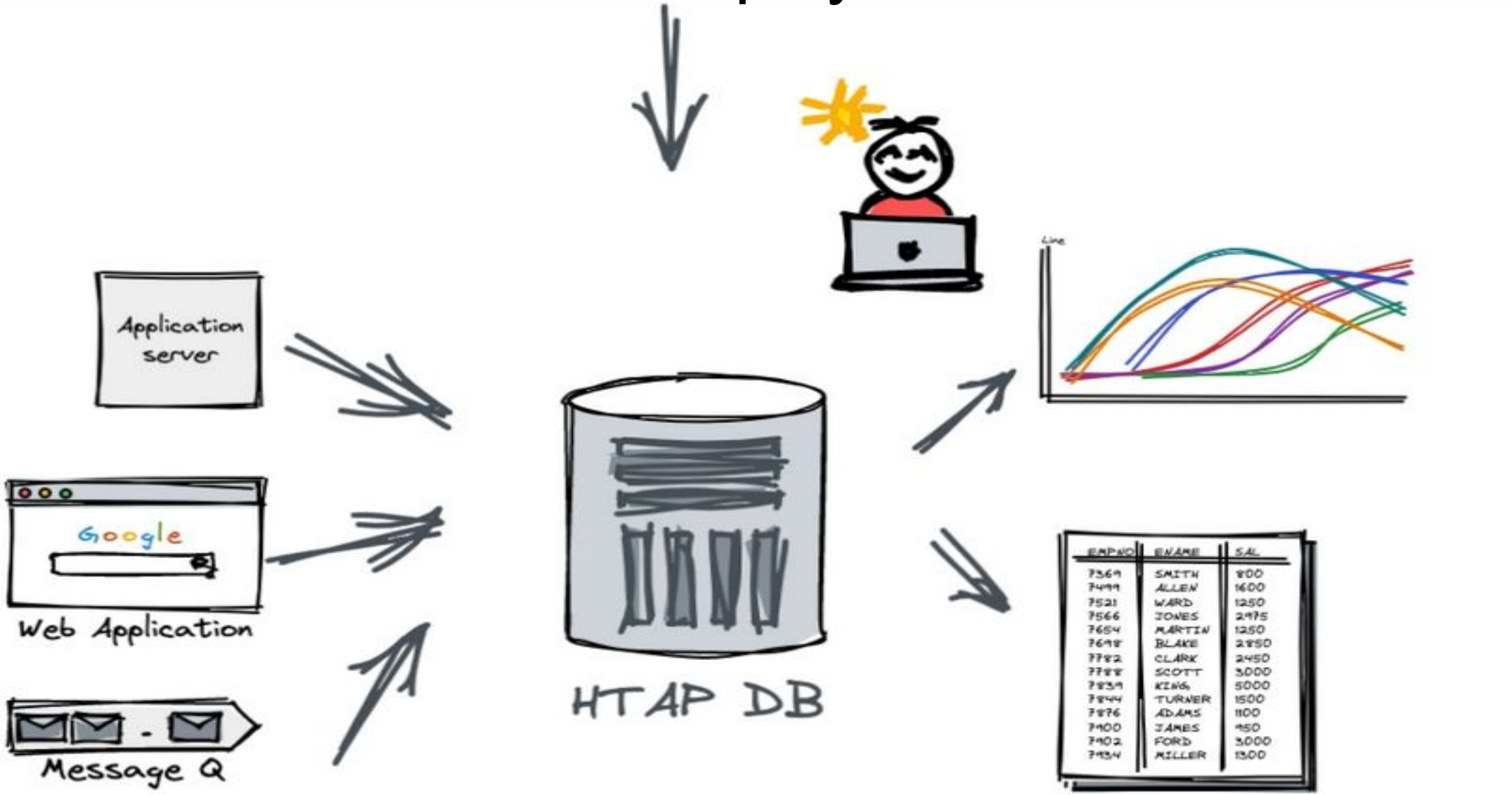
TiDB Serverless - tidbcloud.com

- Simplest way of running TiDB!
- Seconds to get a new serverless cluster up and running!
- High available elastic clusters with free forever option.
- Full access to HTAP functionality
- Pay as you go
- Free options, no credit card registration required
  - 5GiB Storage FREE forever
  - 50M Request Units FREE each month

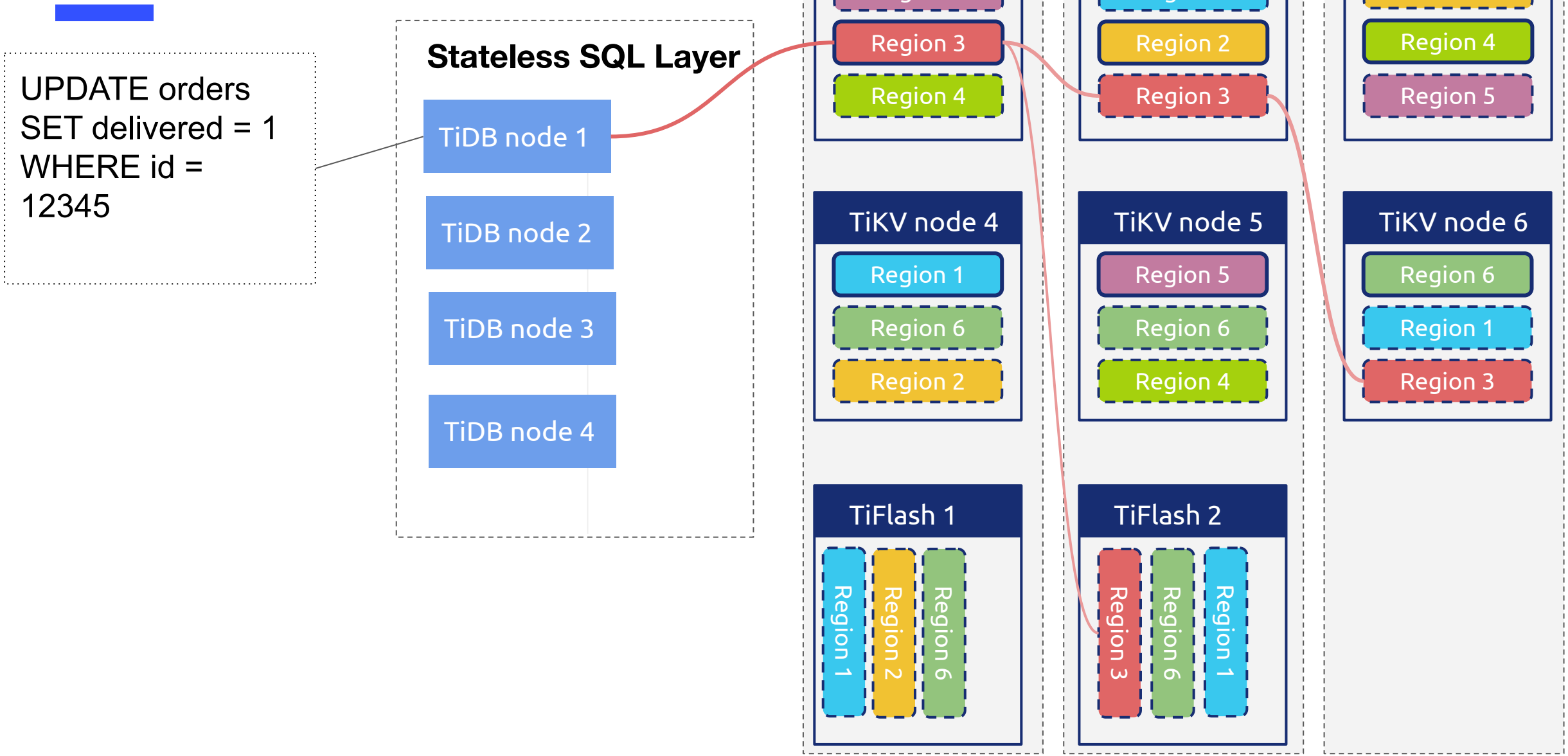
Mixing transaction and analytics?



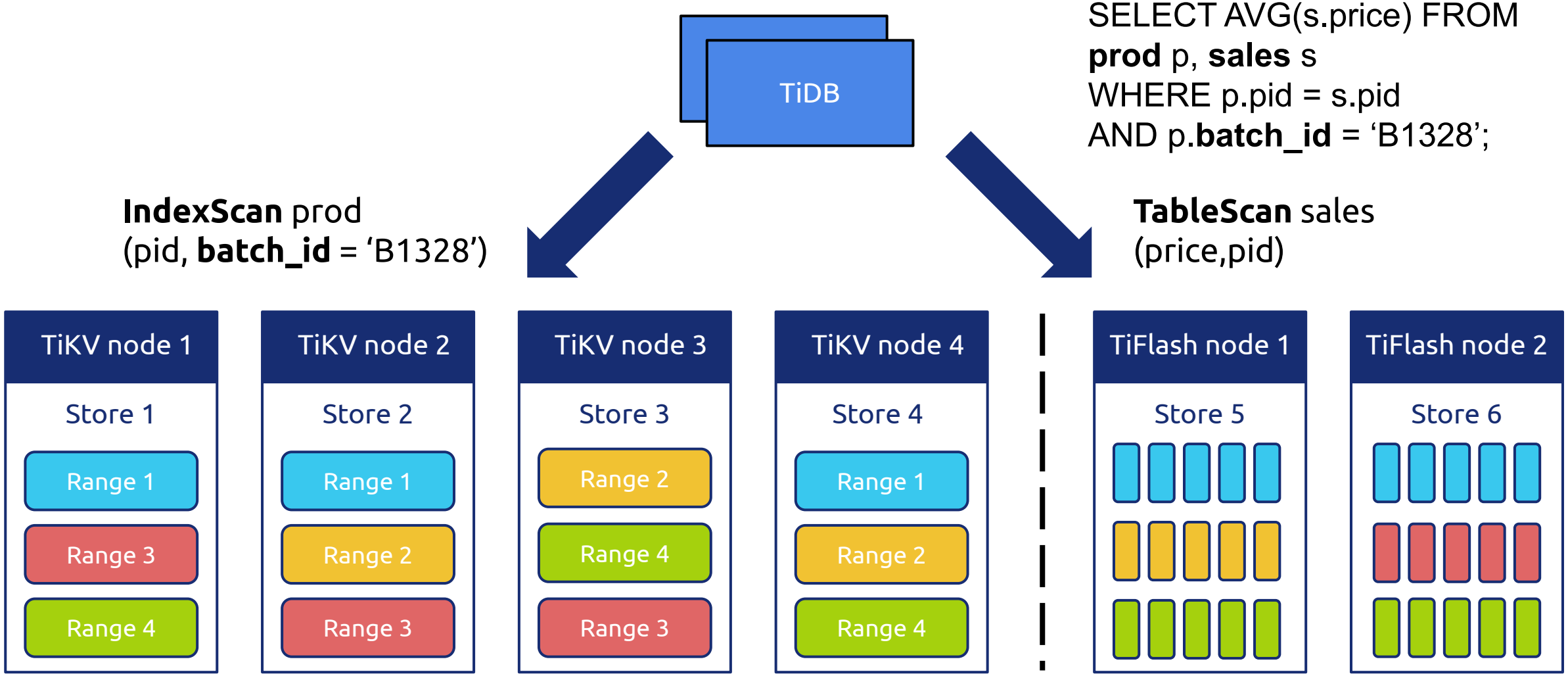
TiDB with TiFlash can simplify it!



TiDB Architecture with TiFlash



The optimizer can pick the best store for each case



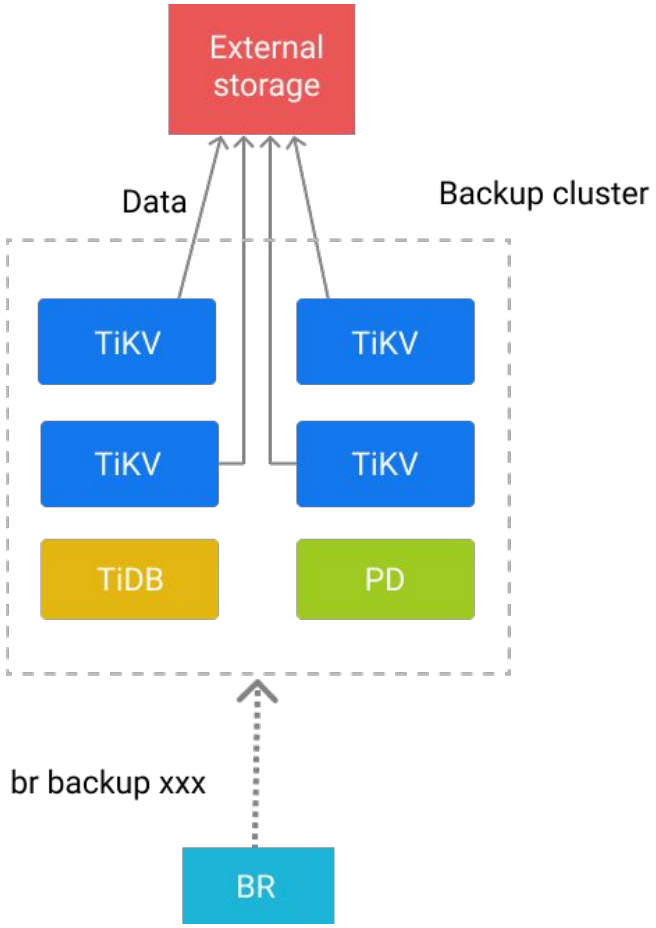
Schema changes

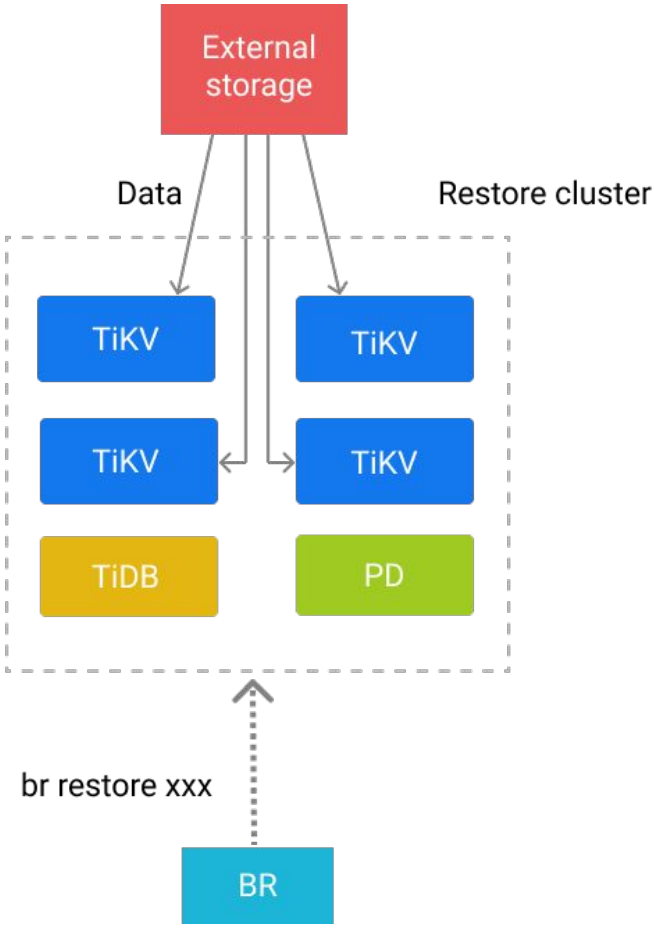
- All online in TiDB.
- Metadata only when possible.
- Adding index is very efficient.
- No external tools like pt-osc (Percona) or gh-ost (GitHub) needed.

Backup/Restore

Backup and Restore (BR)

- The TiKV servers backup to shared storage in parallel.
- The TiKV servers restore from shared storage in parallel.
- This makes backups fast and scalable.
- External storage could be S3, MinIO, GCS or a shared filesystem.
- Opensource.

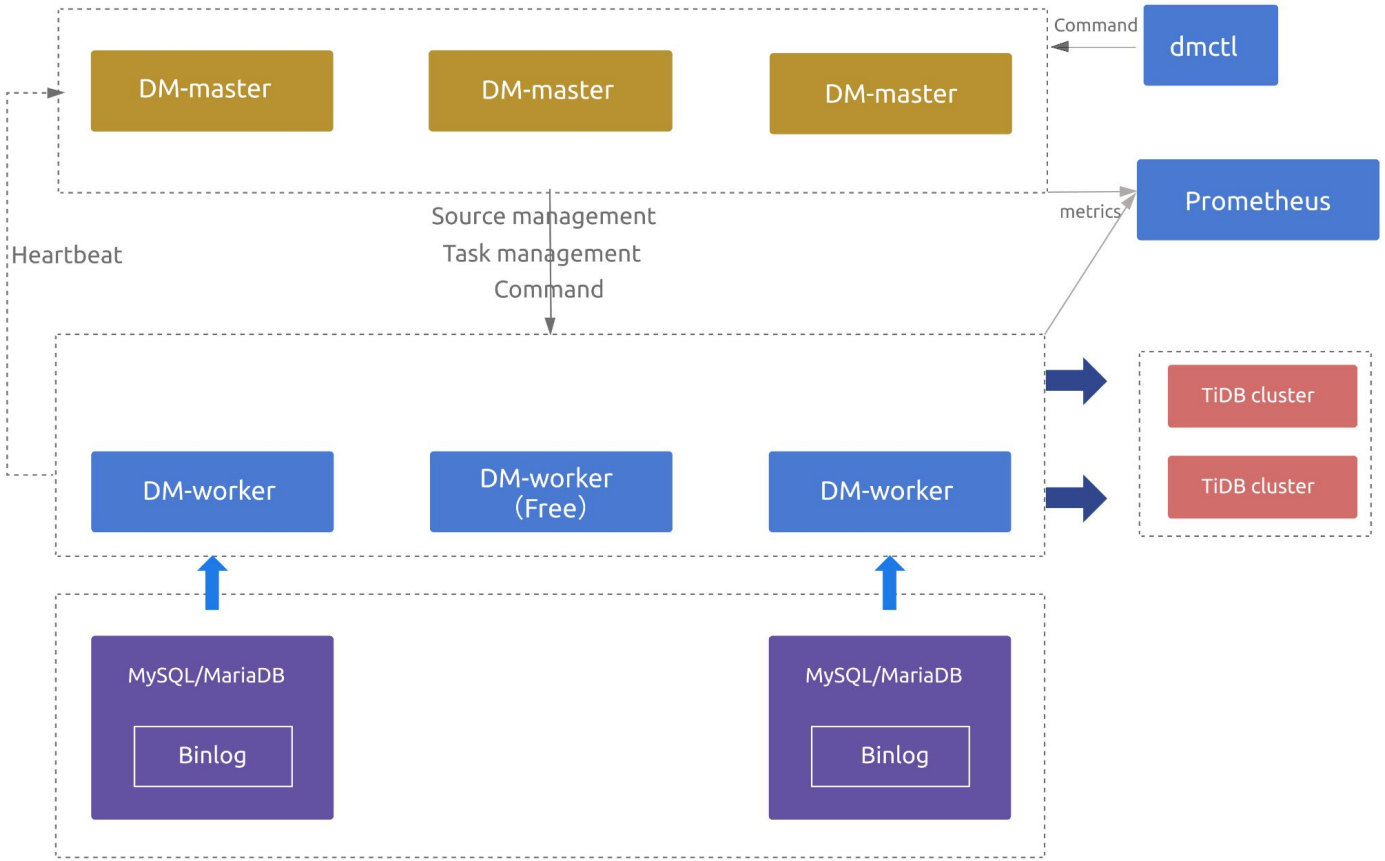




Data migration from MySQL

Data Migration (DM)

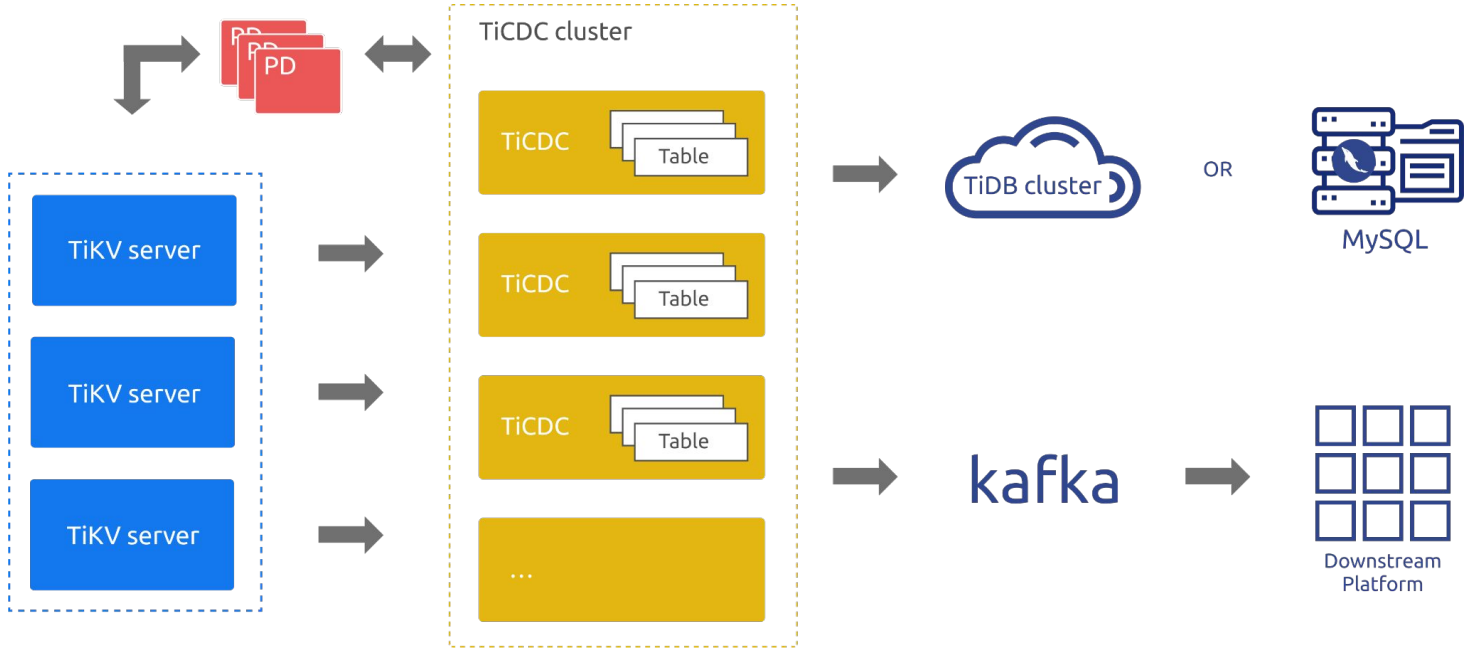
- Dumps and Loads initial data
- Replicate data from MySQL to TiDB
- HA



Change Data Capture

TiCDC

- Send events to Kafka, MySQL or another TiDB Cluster.
- Also high available



Limitations & Compatibility

Documented on
- https://docs.pingcap.com/tidb/stable/tidb-limitations
- https://docs.pingcap.com/tidb/stable/mysql-compatibility

We aim to be compatible with MySQL 8.0. Many MySQL 8.0 features have also been added already.

Row size is limited to 120 MiB.

Missing features in LTS, Long Term Support releases:
- Fulltext indexing
- Geospatial data types, functions, etc.
- Triggers, Stored procedures, Events
- XML Functions

Conclusion

- TiDB allows easy scaling
  - Less work needed by developers for
    - Dealing with read-write splitting
    - Dealing with replication delay
    - Dealing with sharding
- Deploy as you like
- Designed for scale and HA
- Cloud native

Best Fit

- Large Databases (> 5 TB)
- Hundreds of DB instances with Low CPU Utilization
    - *Consolidate into one TiDB cluster for significant cost savings*
- Multi-cloud - Fully Managed Service Offering
- Write Intensive Workload (TPS > 20K)
- Horizontally Scale Both Read and Write
- Read After Write with Strict Latency Requirement
- Distributed SQL or HTAP Database

Not The Best Fit

- Very Small < 100 GB
- Extremely Large Databases > 10 PB
- Primarily JSON/Document based data
- Where extremely low latency (i.e. < 5ms P99) transactions are required

Questions?

Mattias.Jonsson @ PingCAP.com or linkedin.com/in/mjonss

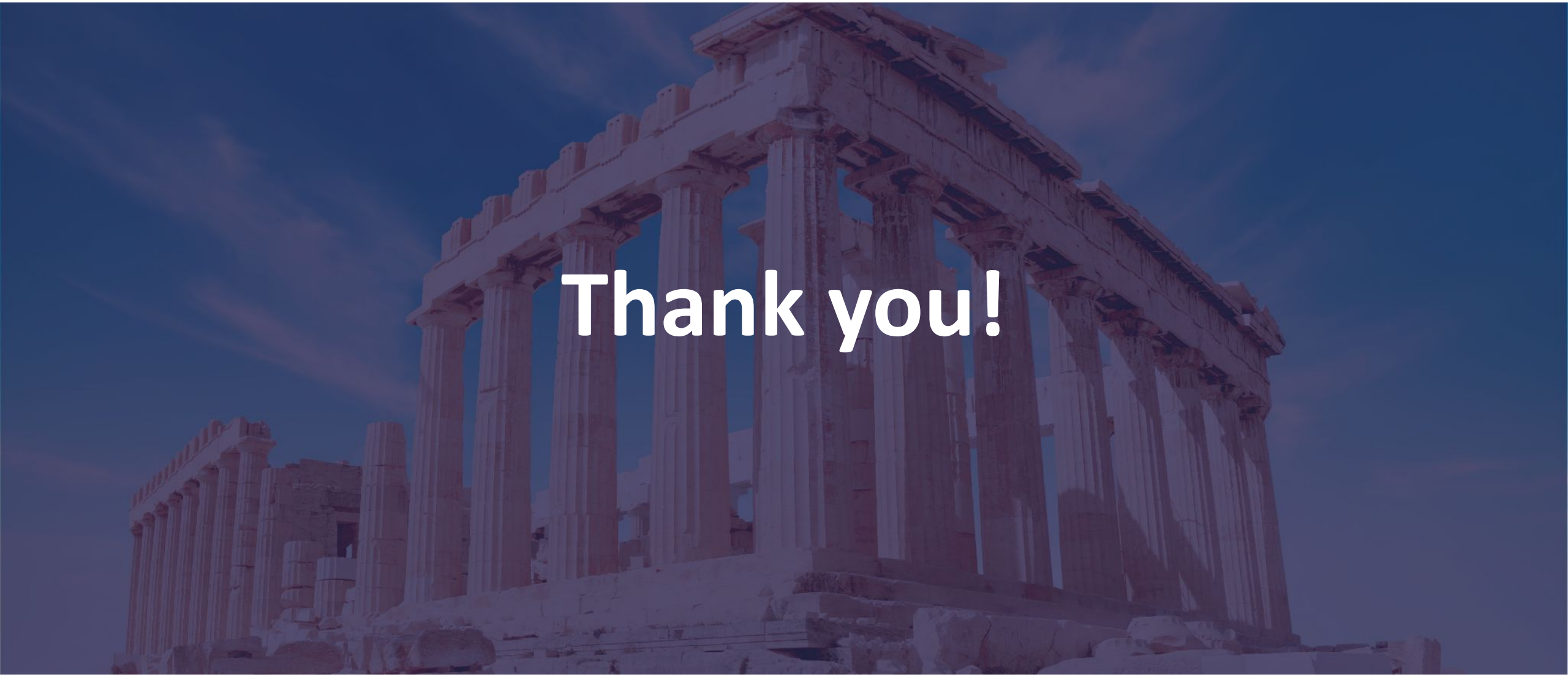
Slack: https://slack.tidb.io
GitHub: https://github.com/pingcap/tidb

Try TiDB:

- Local instance: https://tiup.io
- Free tier on TiDB Cloud: https://tidbcloud.com
- Kubernetes: https://docs.pingcap.com/tidb-in-kubernetes/stable/get-started

Demo of TiDB: OSSInsight.io

- 6 Billion GitHub events, growing by the second
- Compare different open source repositories
- Uses TiFlash for Analytics
- Use ChatGPT to generate SQL queries for analysing the data



In partnership with FerretDB