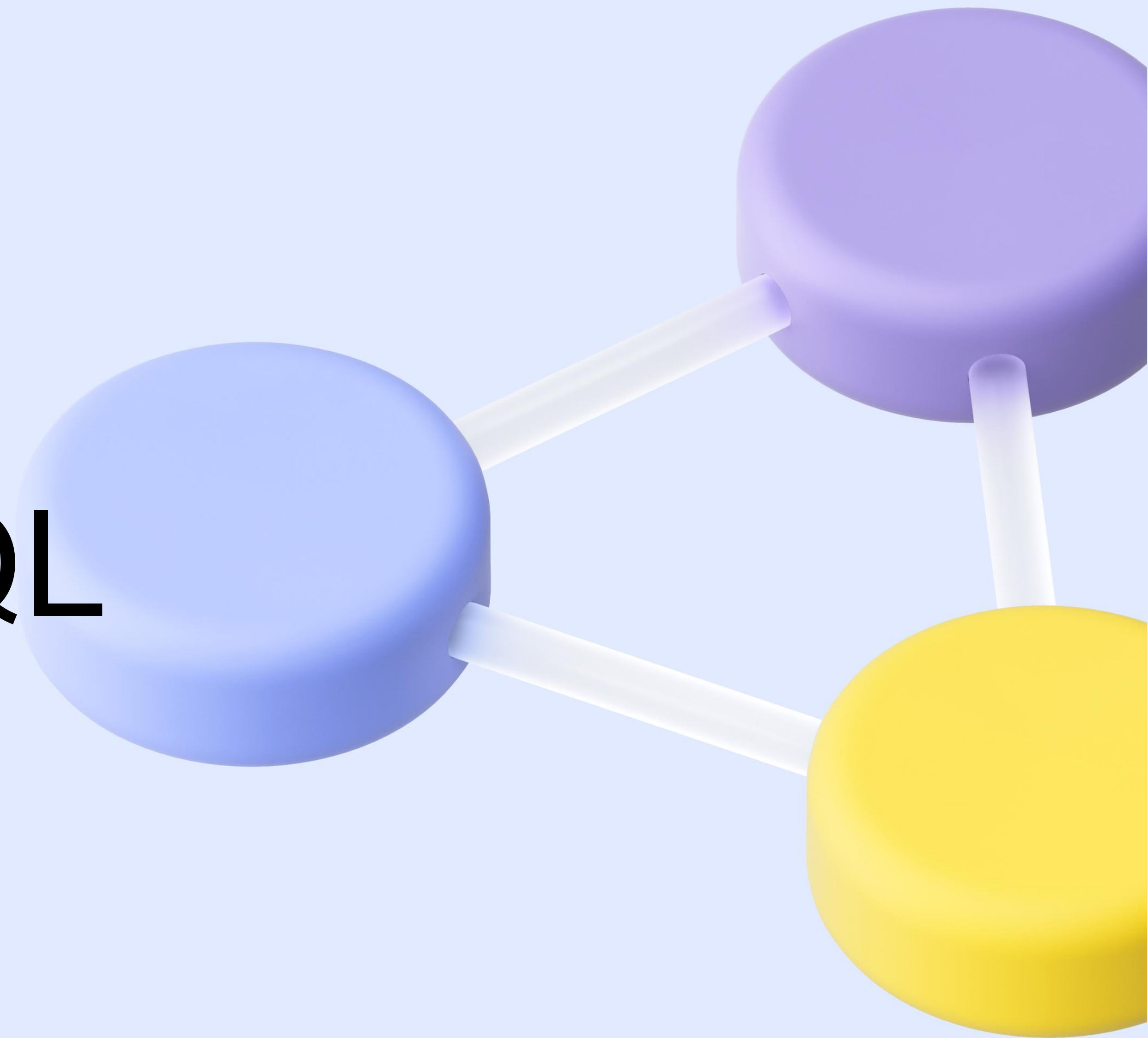




YDB — a Distributed SQL database

Anton Kovalenko,
YDB TPM





**Anton
Kovalenko**

YDB Technical Project Manager

- 2017 • Project manager in YDB
- 2014 CTO in TUI CIS travel operator
CEO
- 2009 CTO in various ecommerce projects
- 2000 Started working in software development

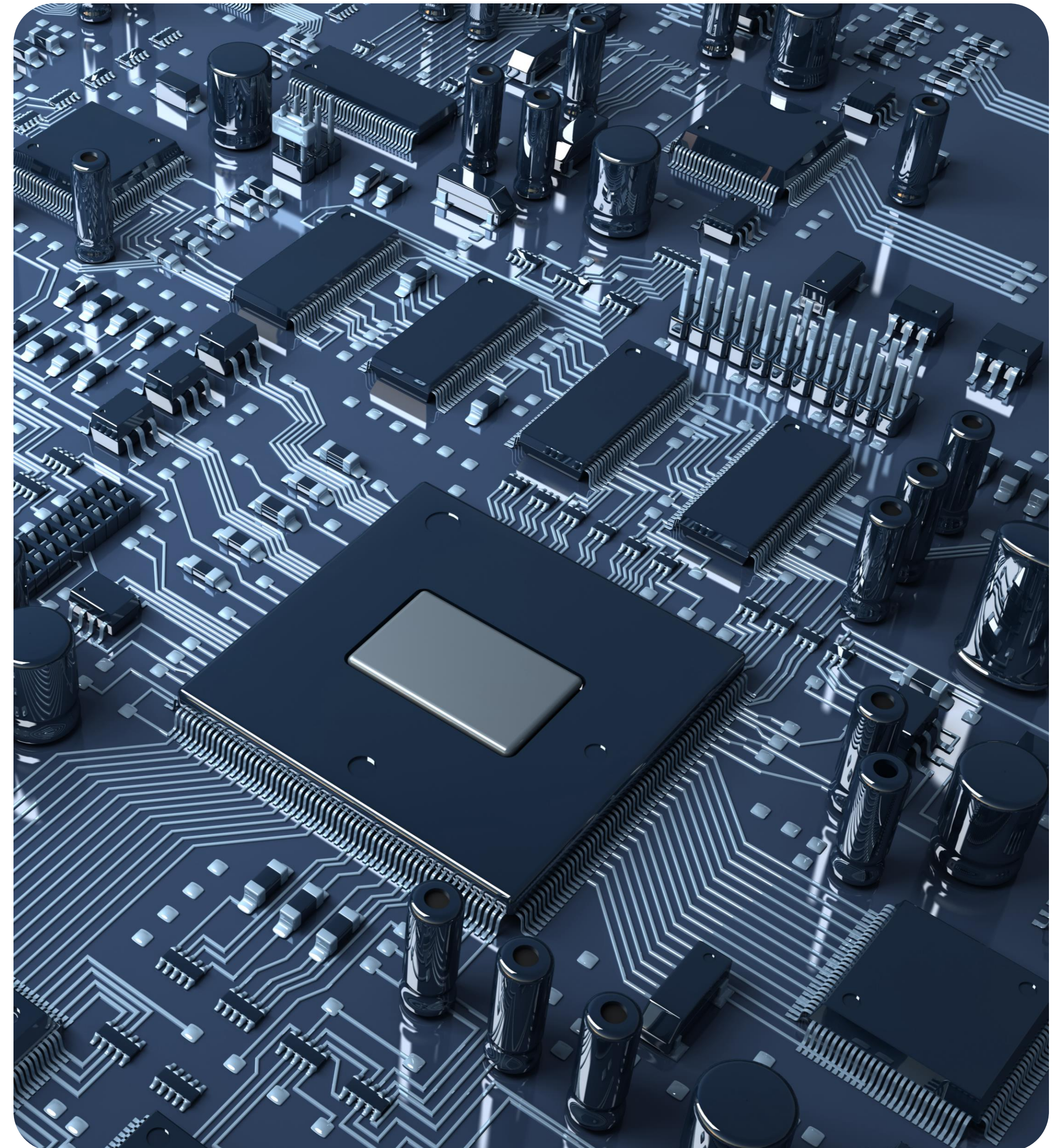
Agenda

1. Why one more database
2. What is a DistributedSQL database
3. YDB Users
4. YDB Layered Architecture
5. Availability and Failures
6. Challenges

1. Why one more database?
2. YDB Story
3. YDB Layered Architecture
4. Availability and Failures

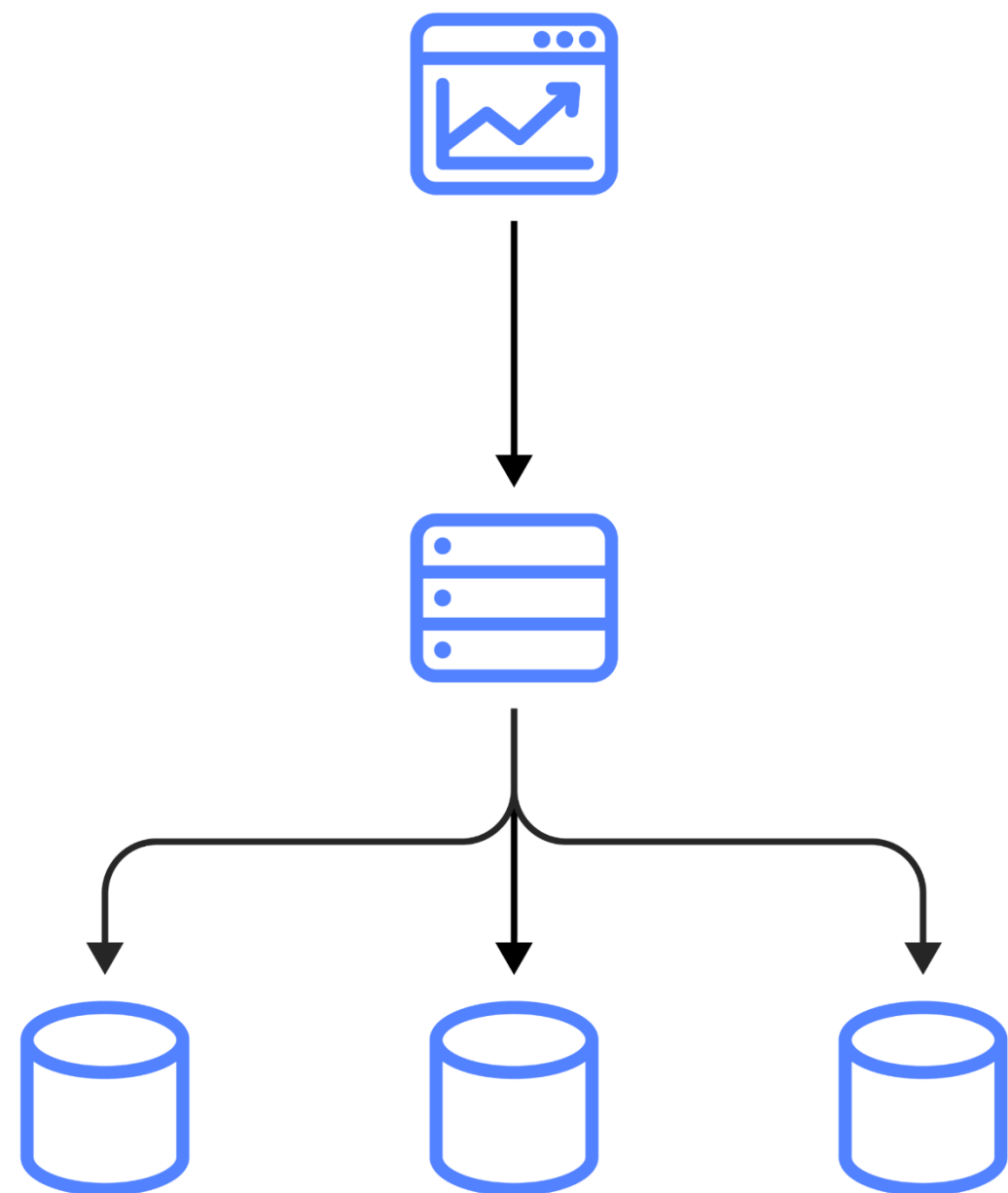
Early 2000s: rapid growth of the internet

- Open-source RDBs are getting more popular
- Commercial RDBMS and required equipment are expensive
Support of vertical scalability in most cases



Mid 2000-s — RDBMS sharding at logical level

- Increases application complexity
- Support costs are getting higher
- Logical sharding has limitations



End of 2000s — NoSQL

- Highly available
- Web-scale
- Relaxed scheme,
or schemeless
- No ACID-transactions
- No JOIN
- Eventual consistency is tricky

Problem statement

- Transactions are difficult
- Eventual consistency strikes back
- SQL is popular

Birth of DistributedSQL Databases

2012



Google Spanner

2015



CockroachDB

2018



Managed YDB
In Yandex Cloud

PolarDB
by Alibaba

2022



YDB goes opensource

What is a distributed SQL database?



“ A category of relational DBMSs designed to support scalable workloads for operational applications

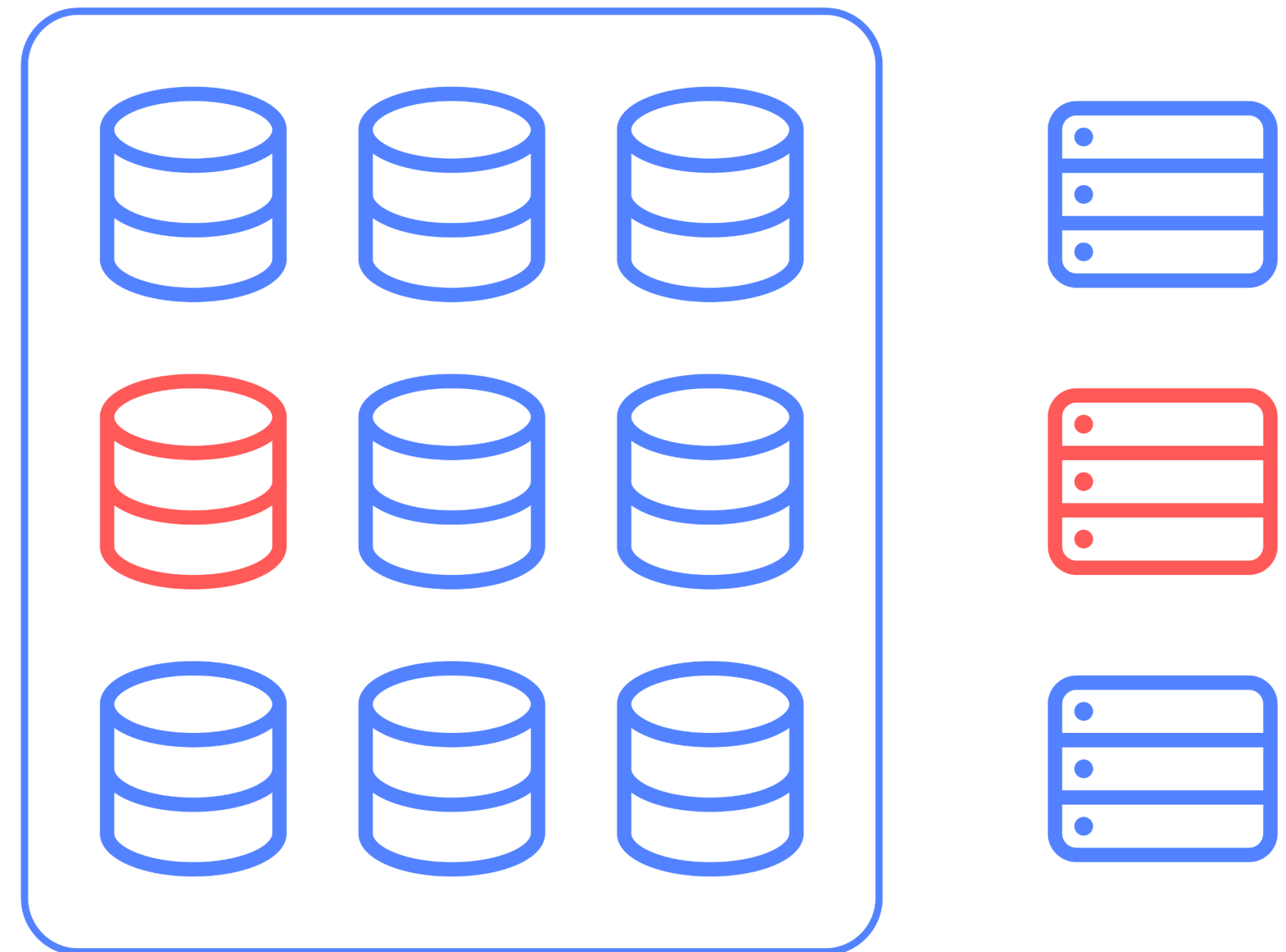
Andy Pavlo's talk at Hydra 2021

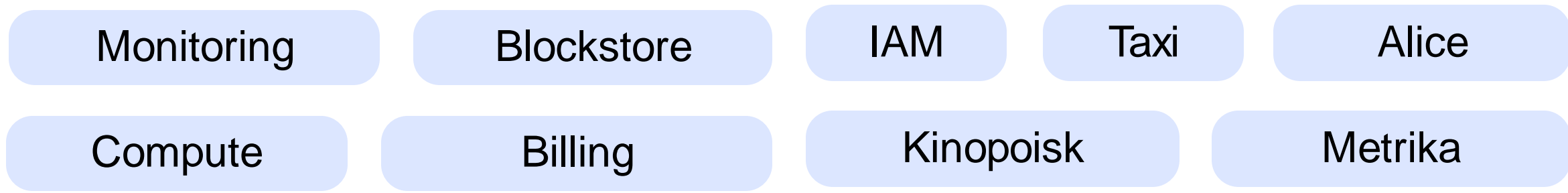
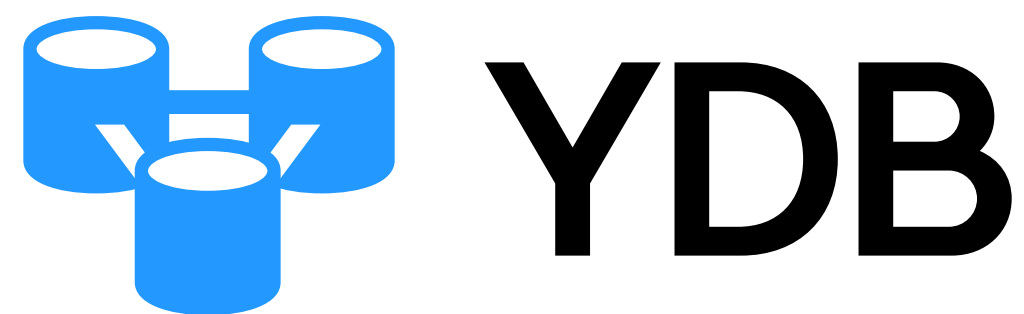
- SQL
- ACID-transactions across availability zones
- Horizontally scalable
- Automatic recovery after failures

1. Why one more database?
- 2. YDB Story**
3. YDB Layered Architecture
4. Availability and Failures

YDB — Open-Source Distributed SQL Database

- **Relational**
ACID OLTP transactions
- **Consistency**
Strongly consistent
Serializable transaction isolation level
- **Mission critical database**
Works for projects with 24x7 requirements
- **Highly available**
Survives AZ plus rack failure w/o human intervention, available for read/write





Yandex Cloud

3 200

Databases

60PB

Storage

26 000

Nodes

Yandex

1 000

Databases

5.7PB

Storage

10 000

Nodes

Open source under Apache 2.0



github.com/ydb-platform/ydb

 [ydb-platform / ydb](#) Public

 Edit Pins ▾

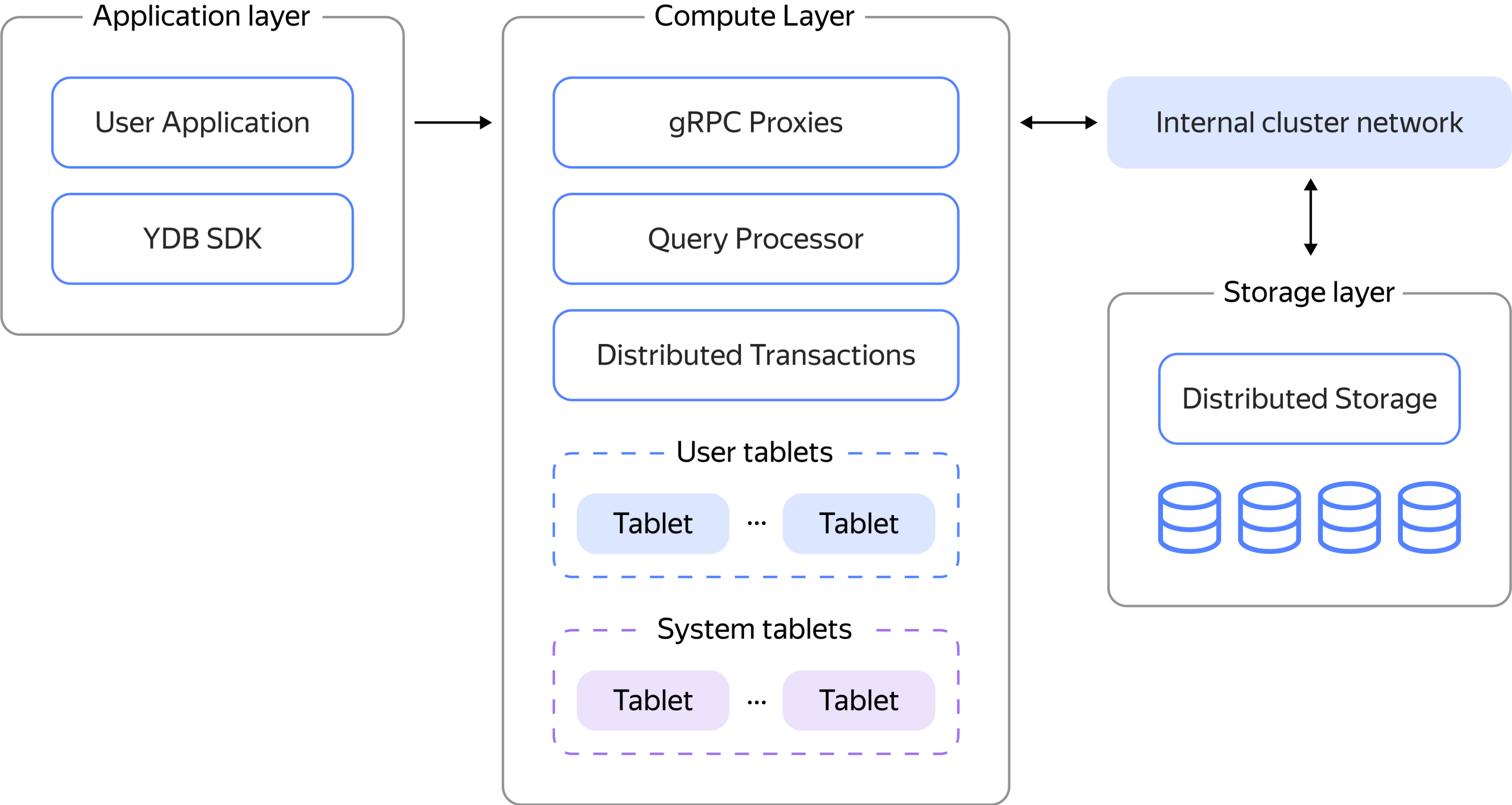
 Unwatch 43 ▾

 Fork 215 ▾

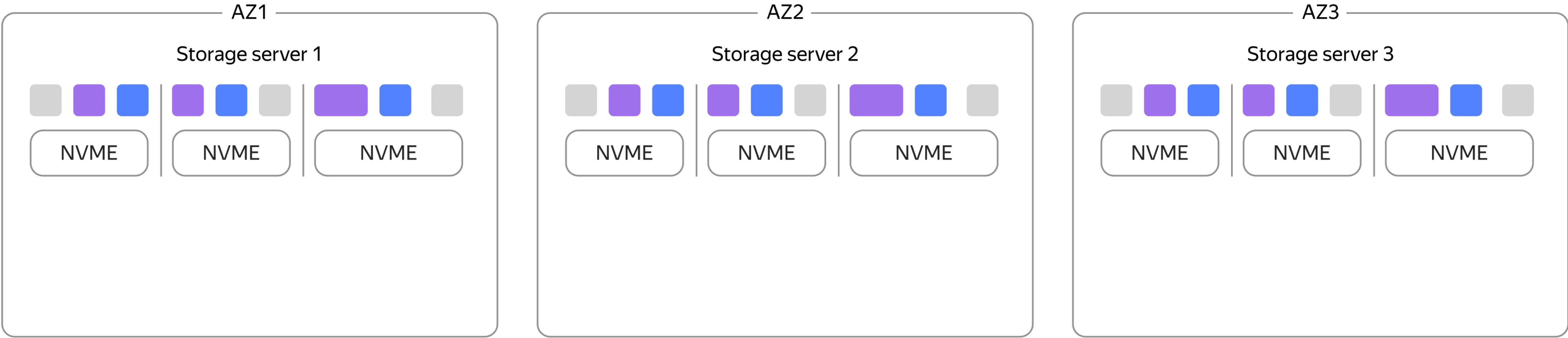
 Starred 2.8k ▾

1. Why one more database?
2. YDB Story
- 3. YDB Layered Architecture**
4. Availability and Failures

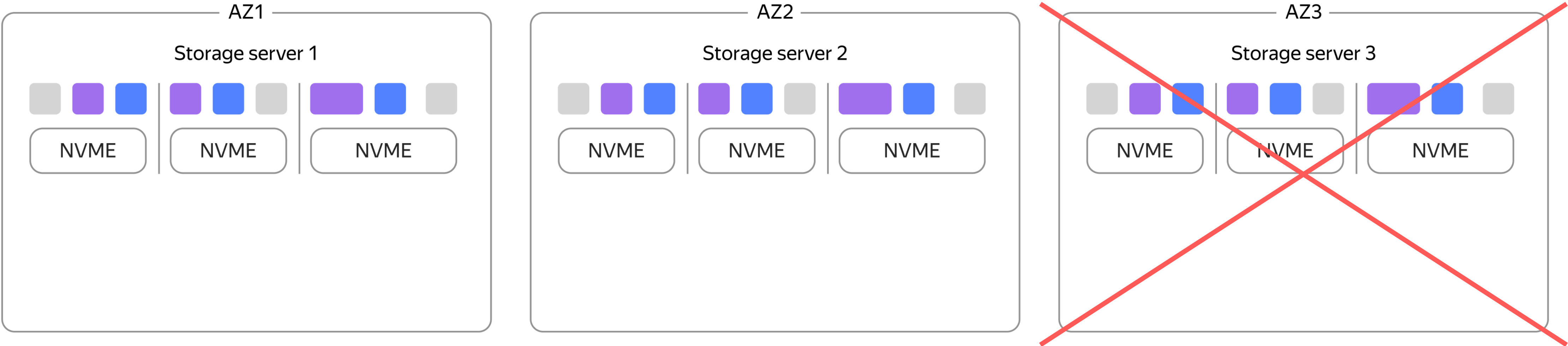
YDB Architecture



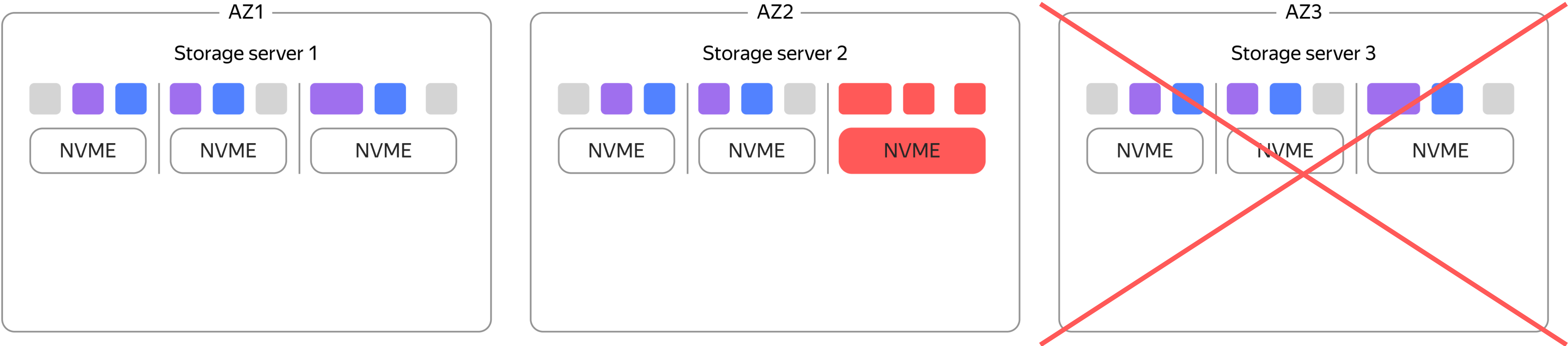
YDB storage availability model



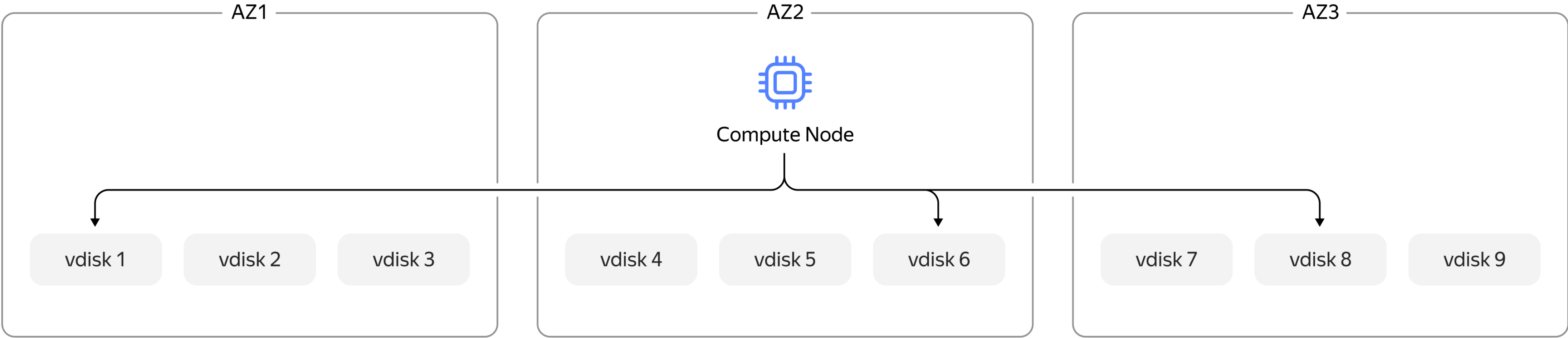
YDB storage availability model



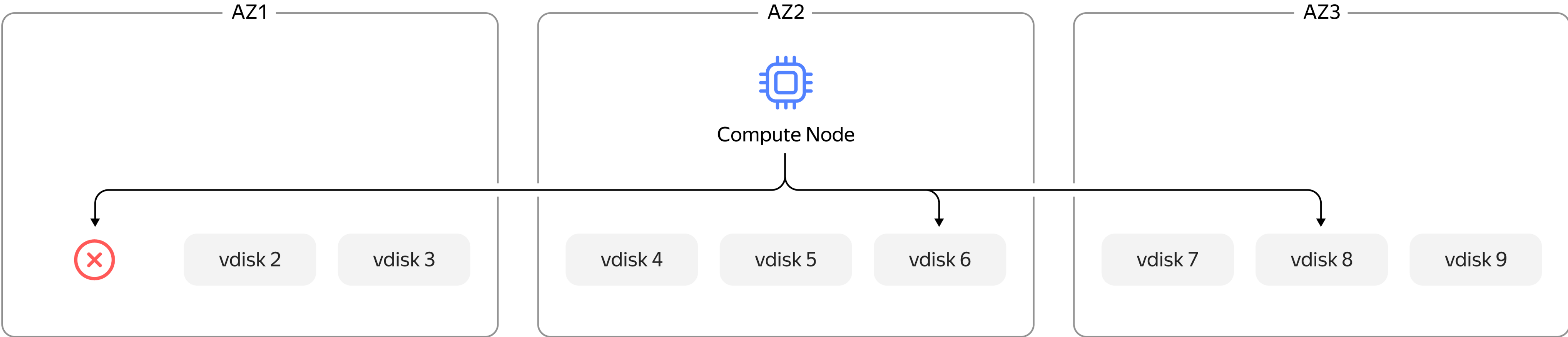
YDB storage availability model



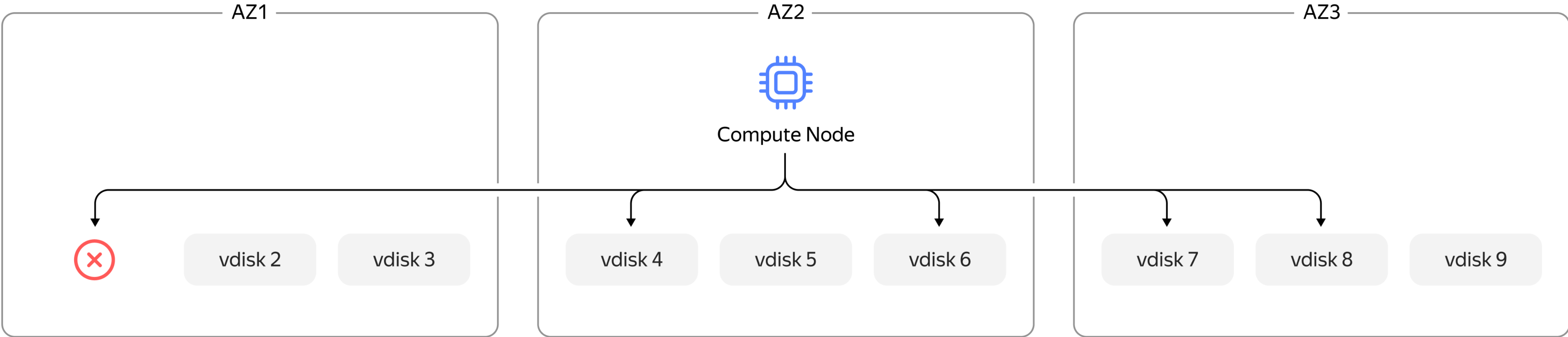
Storage group is a key to availability



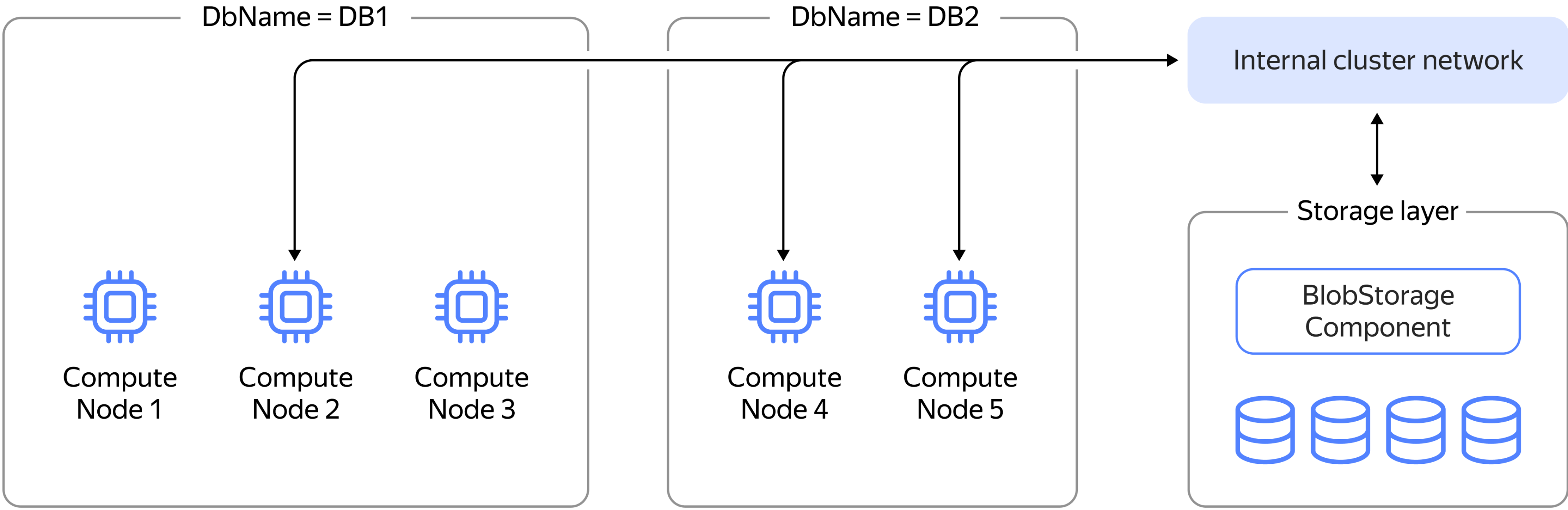
Storage group is a key to availability



Storage group is a key to availability

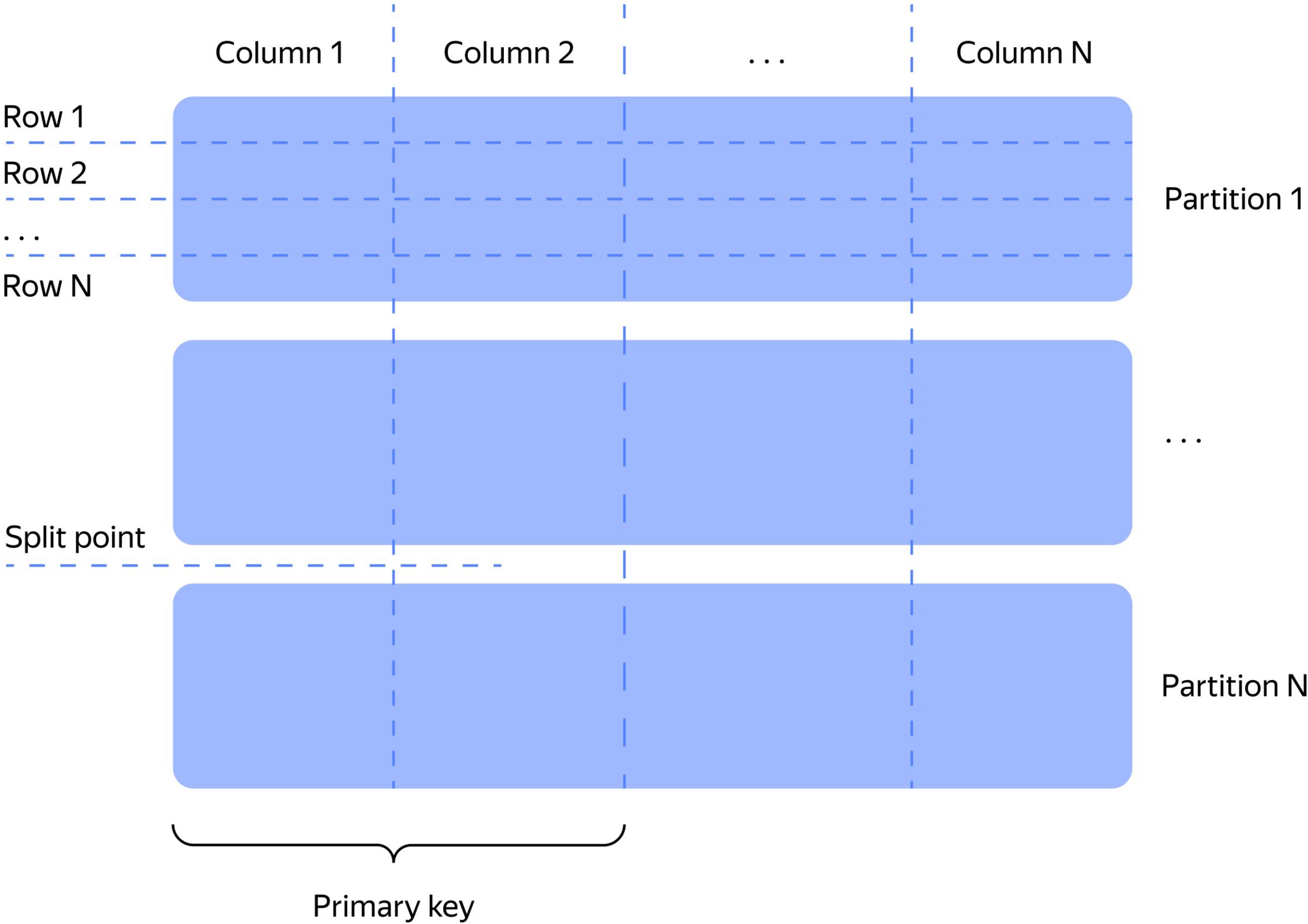


Compute node can run on any cluster node

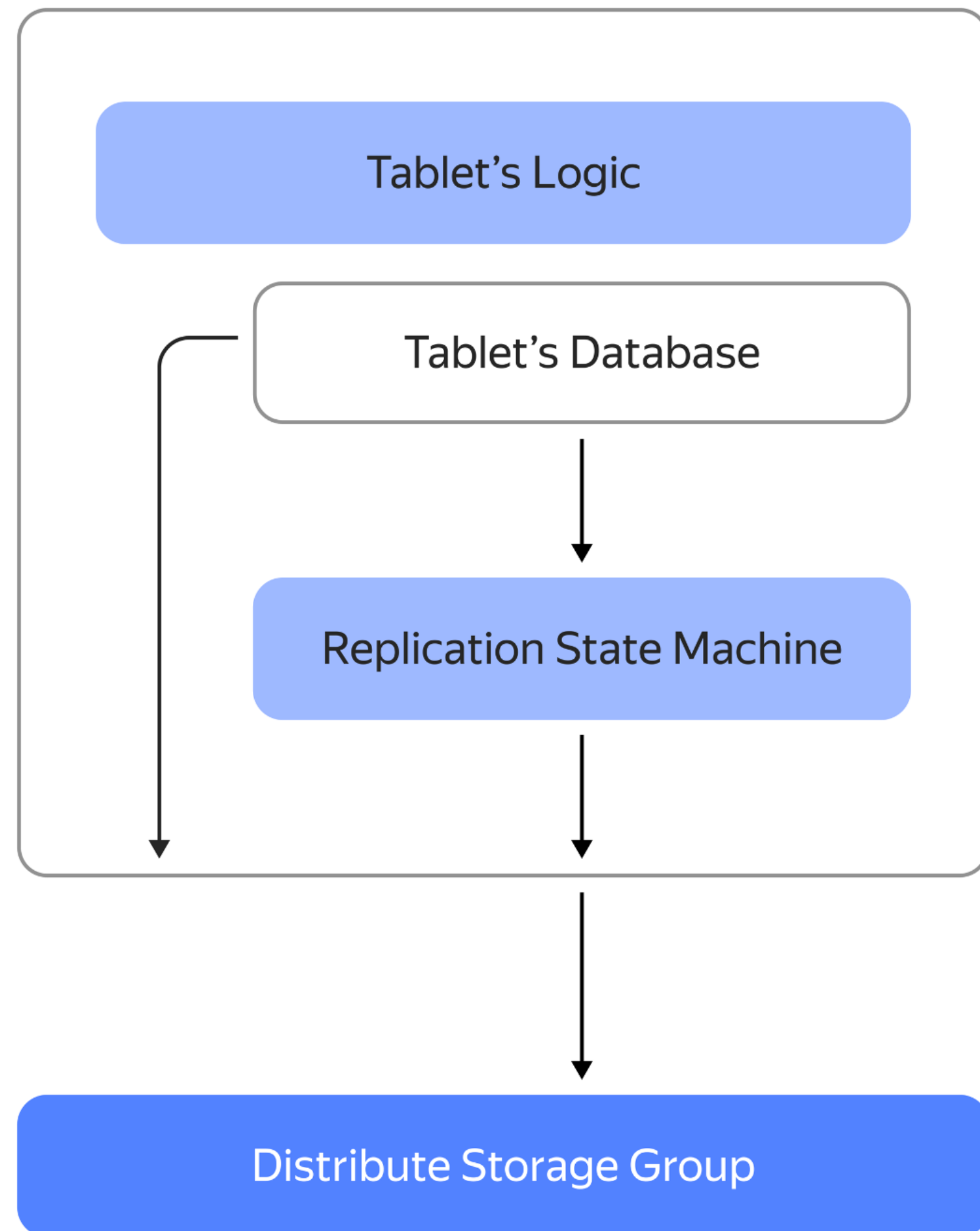


Users operate tables to read and write data

- Tables have a primary key (PK)
- Tables are sorted by PK
- Tables could grow up to petabytes of data
- Tables are automatically partitioned



Inside Partition



Replication State Machine (RSM)

1. Writes a log of changes
2. Recovers from log on tablet crash
3. Provides guarantees analogous to RAFT and Paxos

Tablet's Database

1. Data is organized as an LSM-tree
Log Structured Merge tree
2. Guarantees ACID properties for the data it is in charge

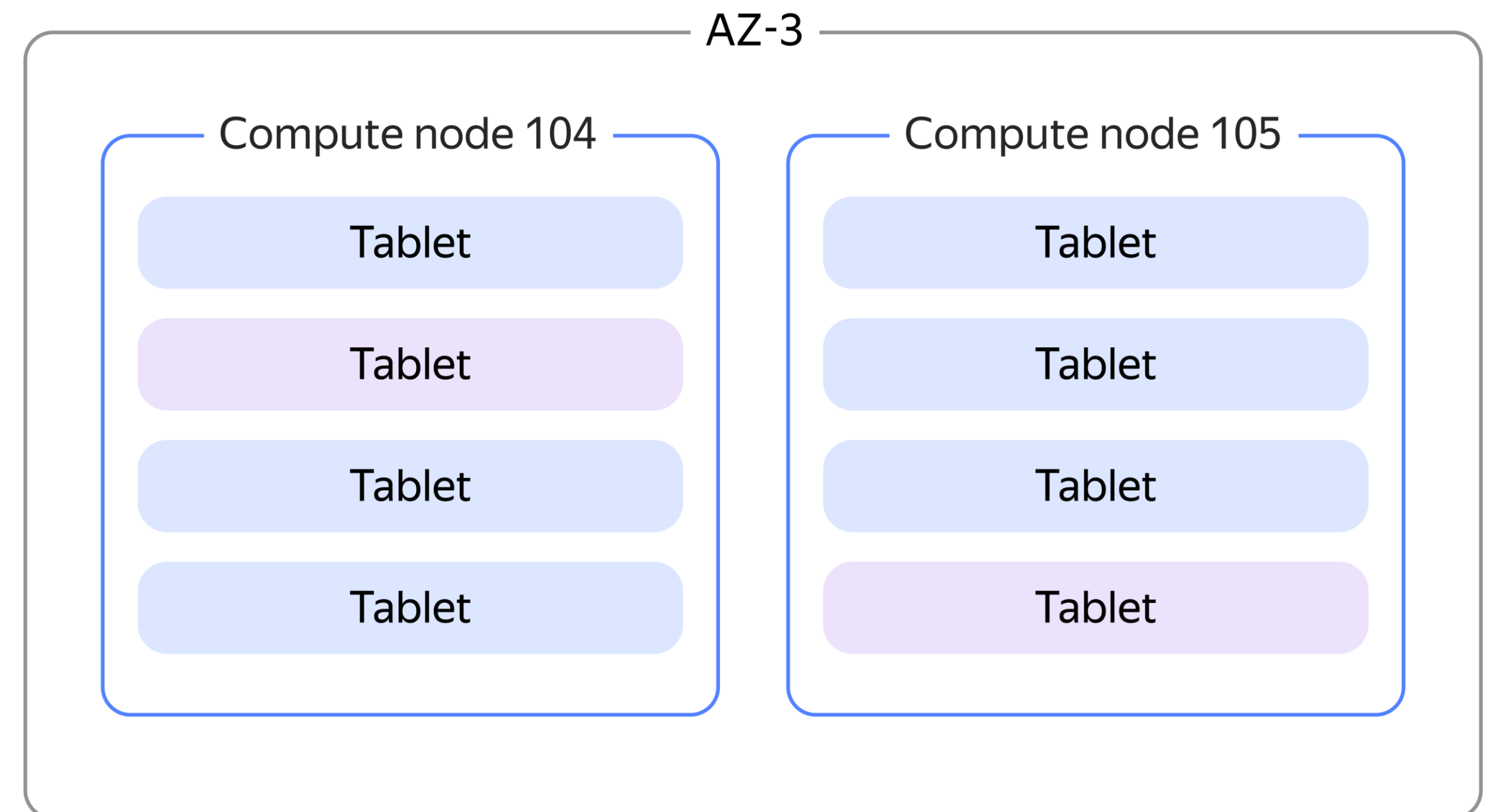
Tablet's Logic is specific for the Tablet type

1. Can implement some API
2. Can be active component that rebalance something in cluster

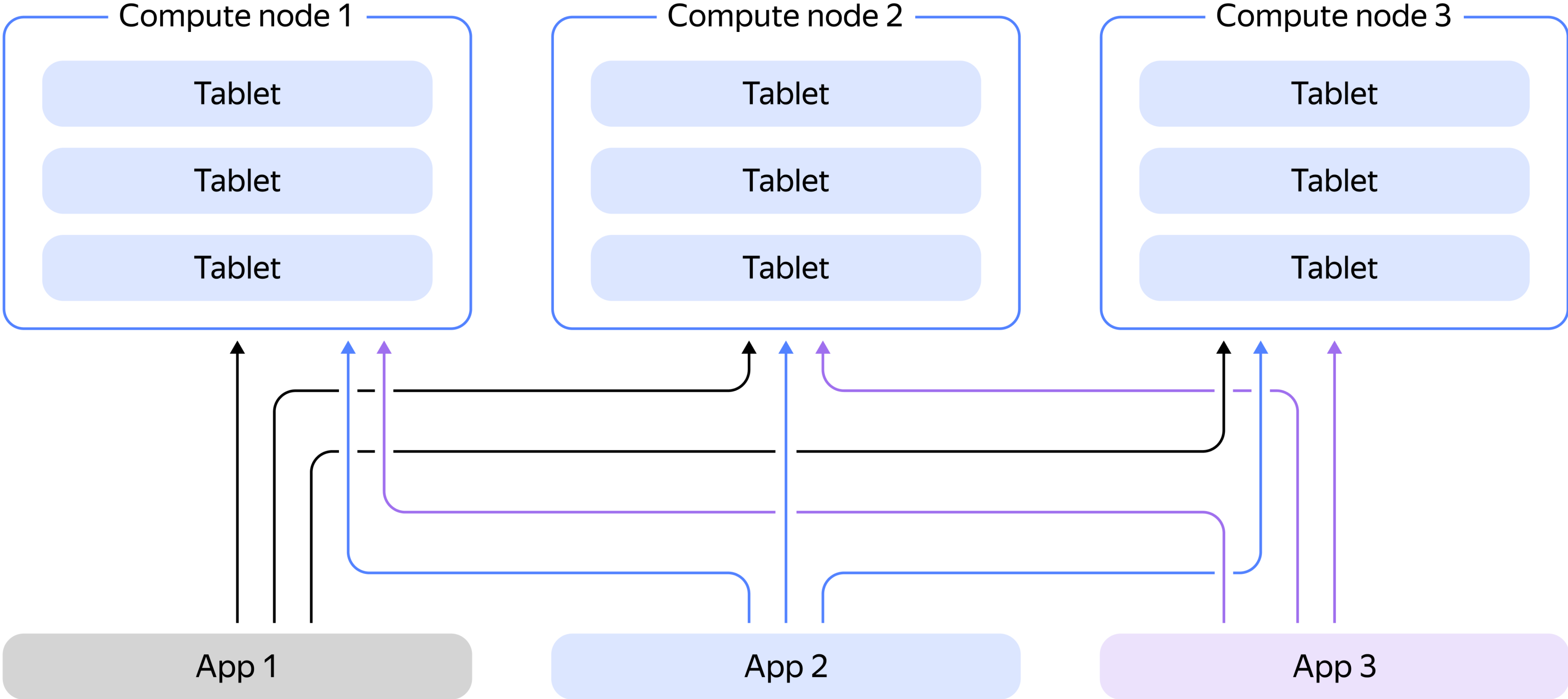
Distributed storage provides reliable data storage with redundancy

Tablets run on compute nodes

- Tablet can run on any compute node
- Can handle user requests
- Can perform system tasks
- Scalability block

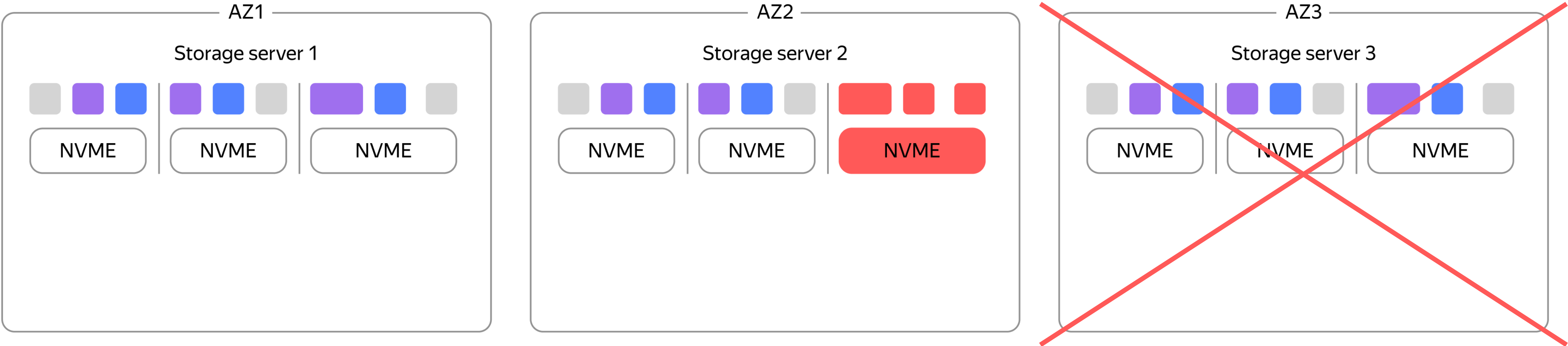


YDB SDK hides the complexity

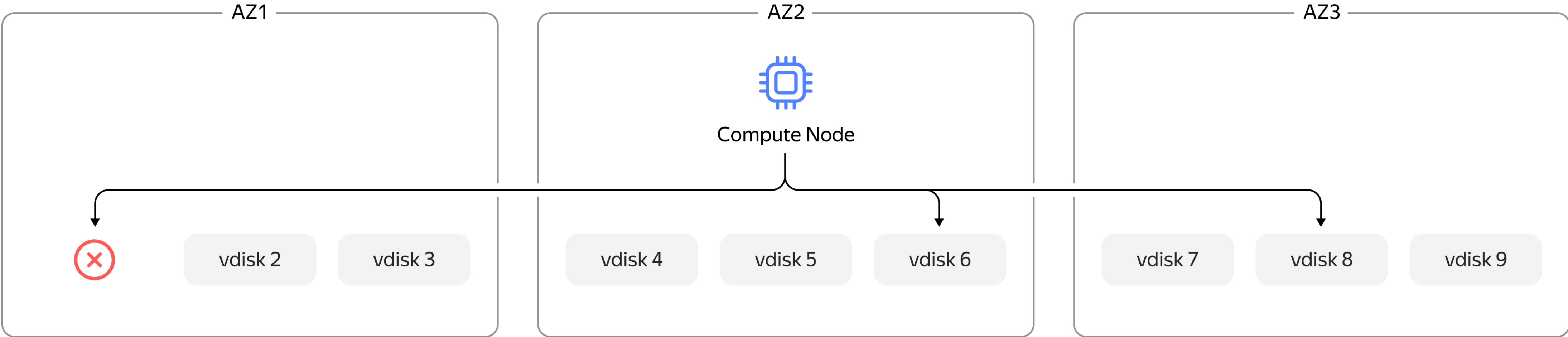


1. Why one more database?
2. YDB Story
3. YDB Layered Architecture
4. Availability and Failures

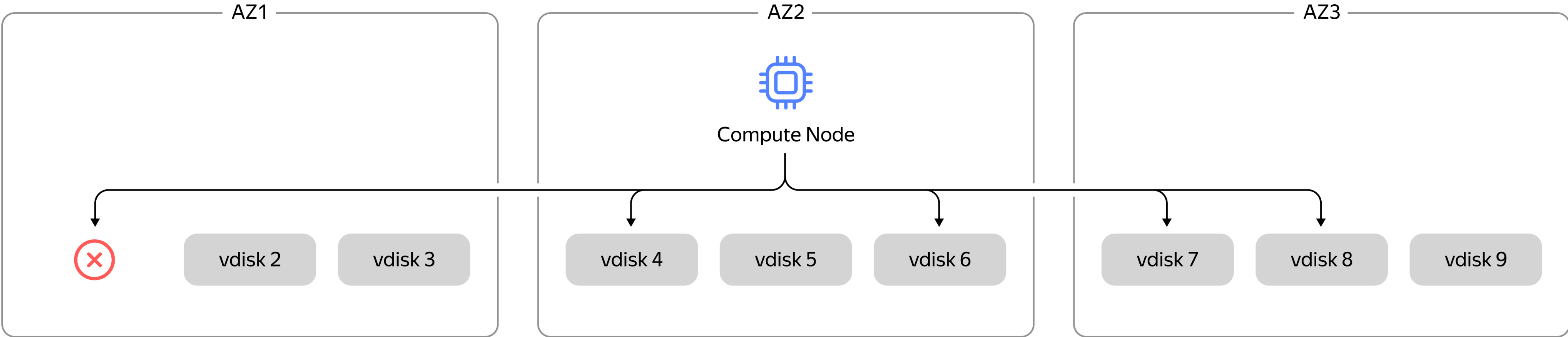
Storage availability model



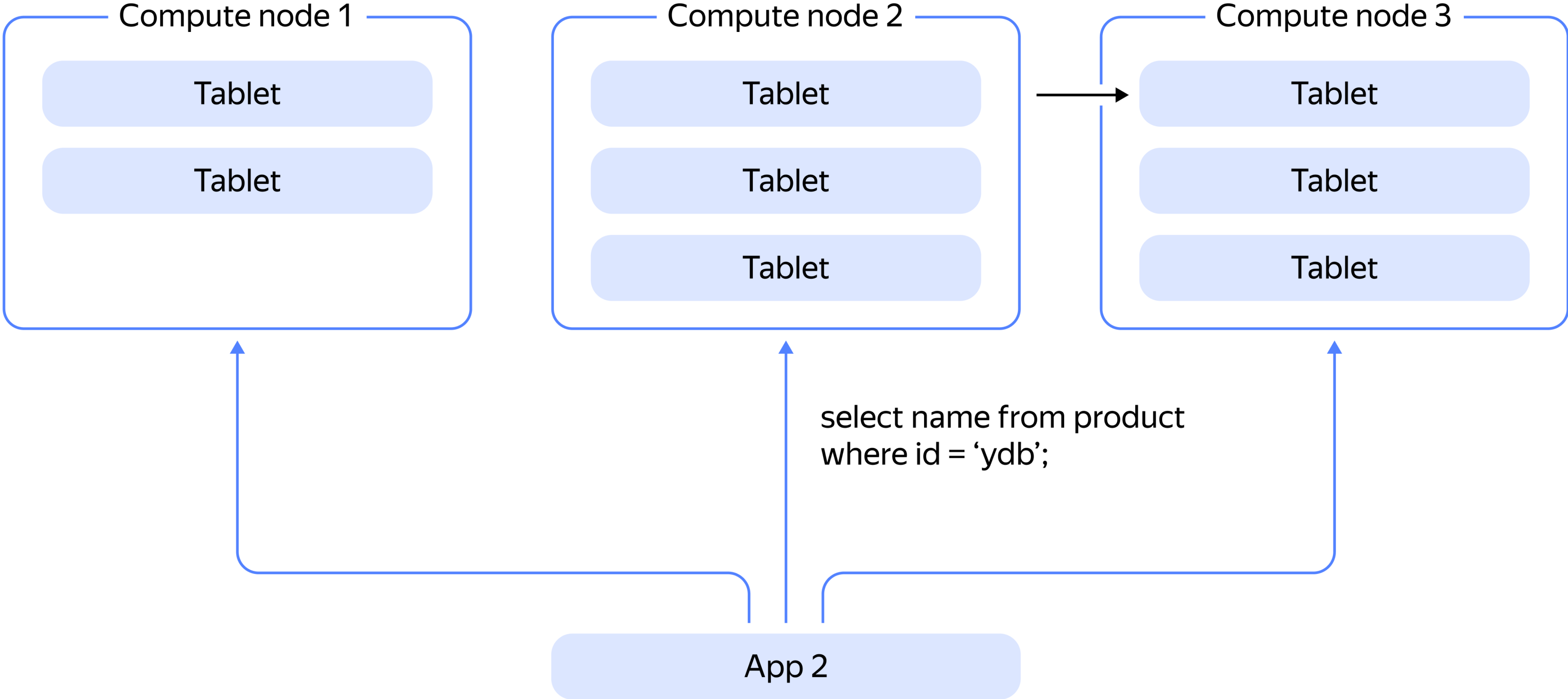
Storage group is a key to availability



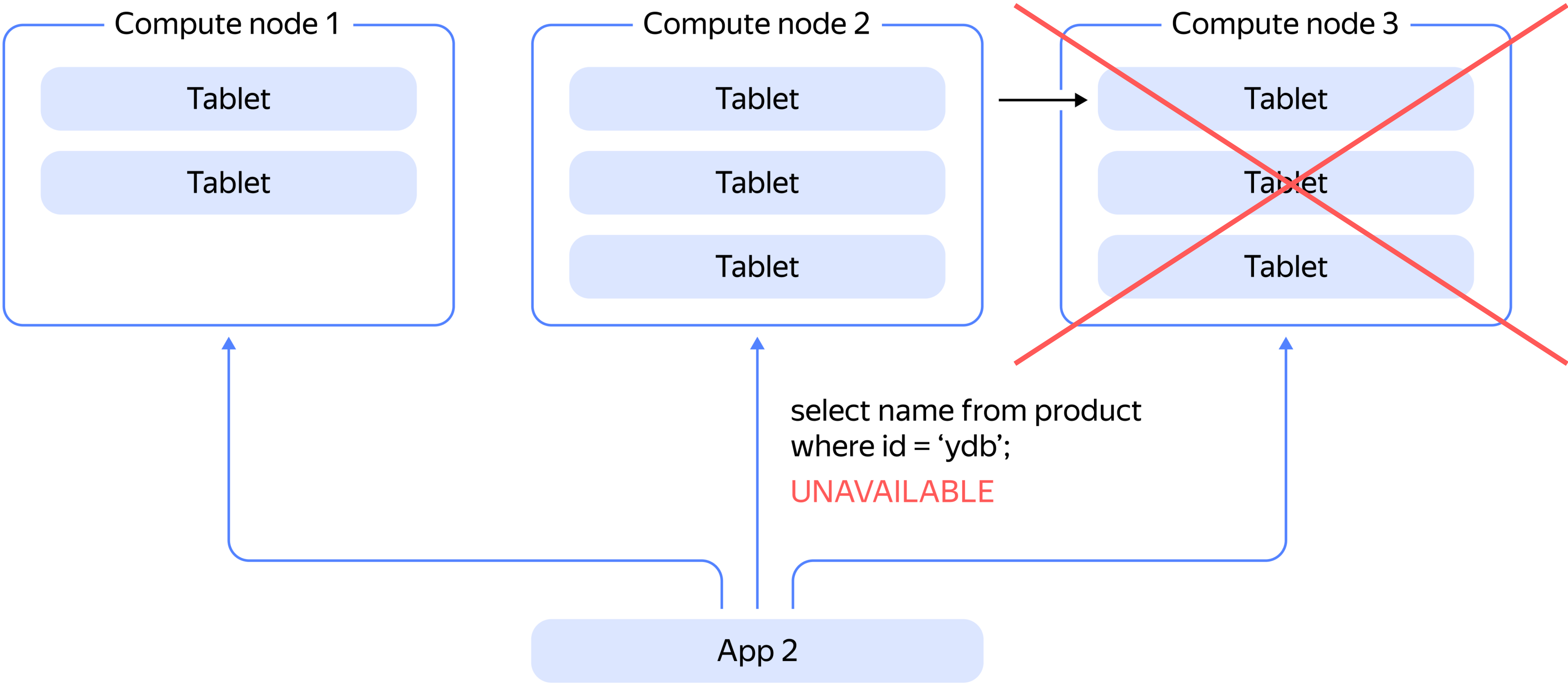
Write additional copies



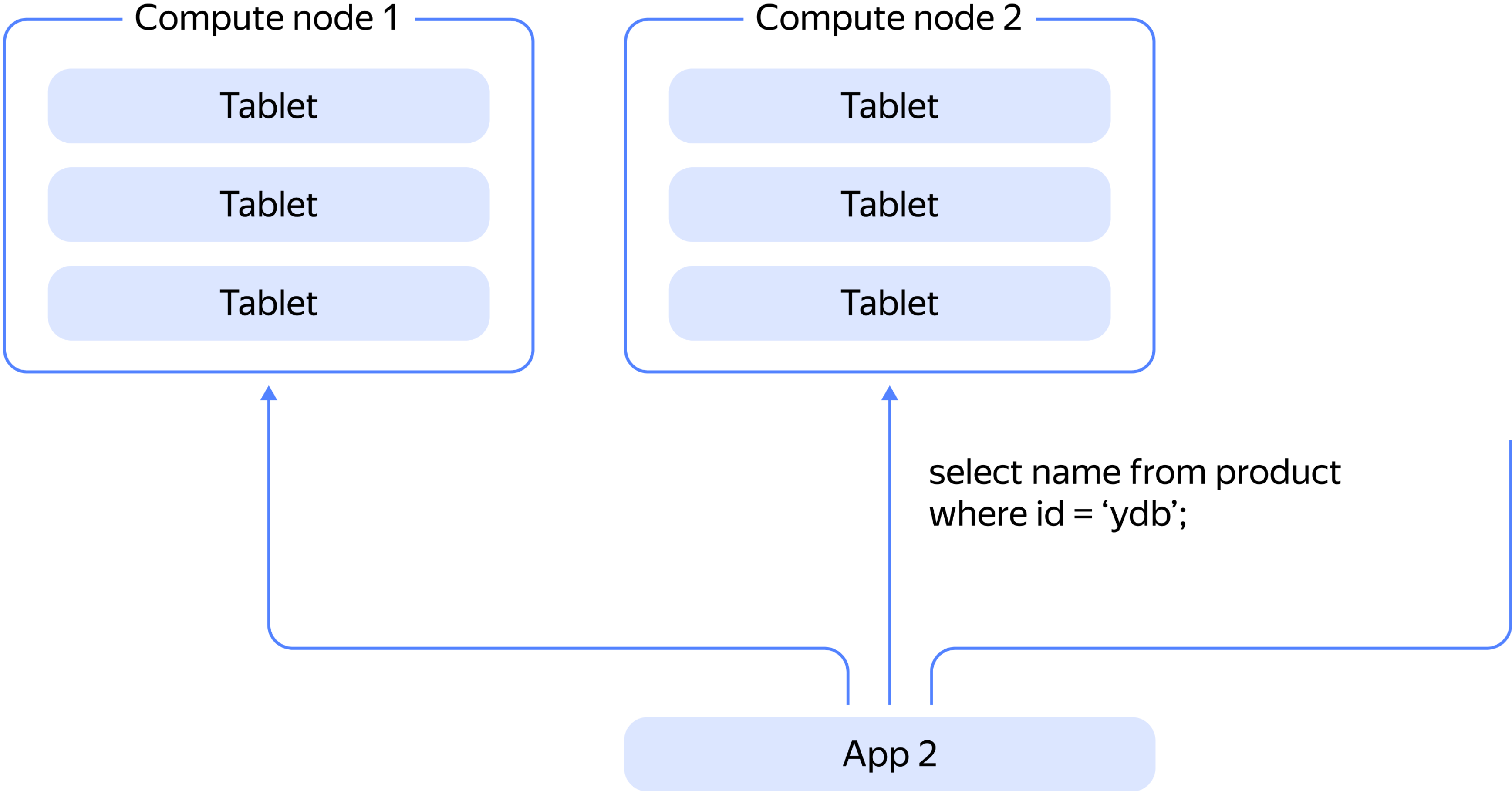
Compute node failure



Compute node failure



Compute node failure



DistributedSQL

Horizontal Scalability

Built-in scaling out capabilities

RDBMS Features

Tables. SQL, ACID transactions, strict consistency

Cloud-native

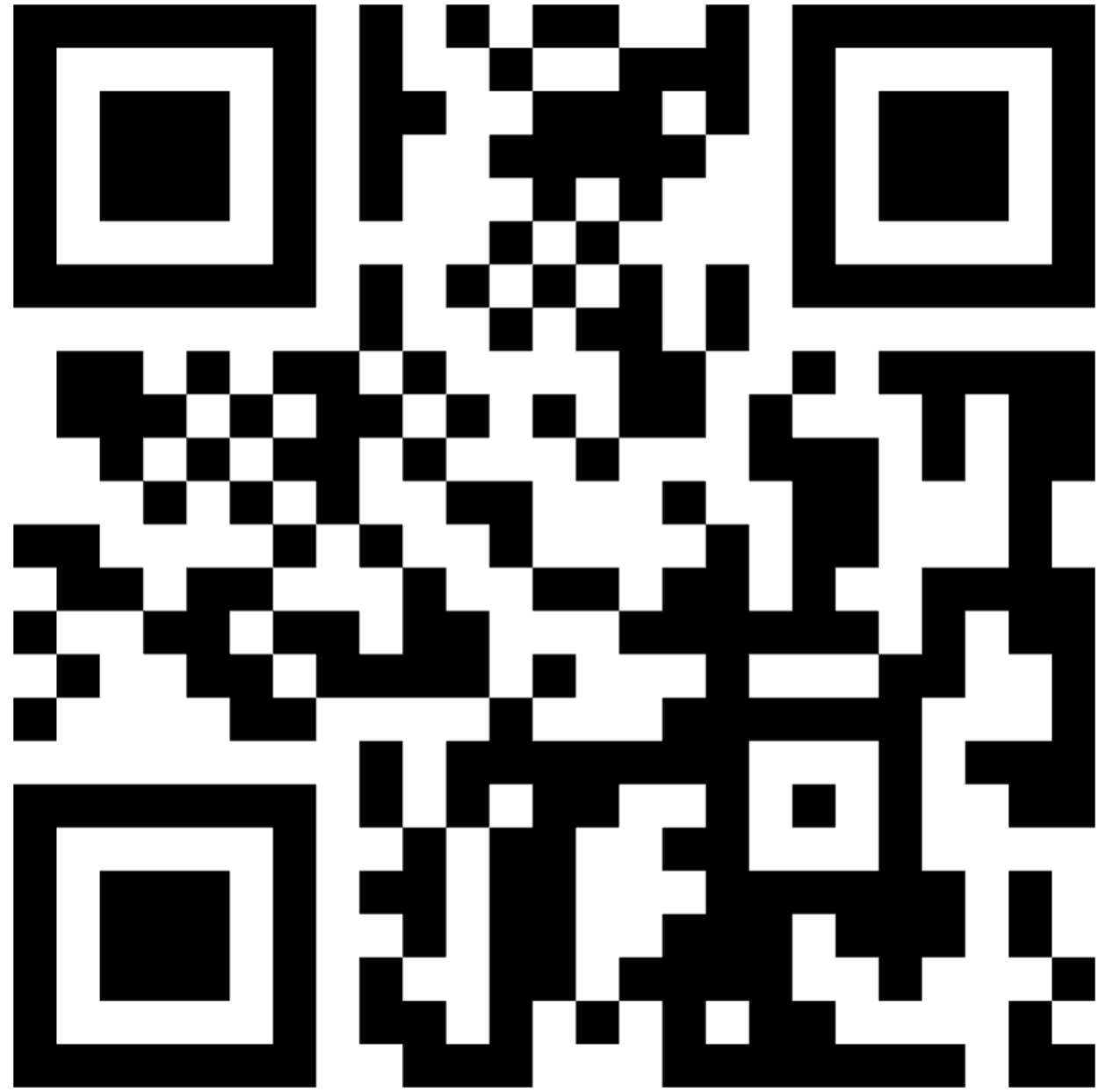
Has both managed options in different clouds and on-premise deployment options

Highly available

Automatic recovery after failures without human intervention

Current challenges

- **PostgreSQL wire protocol and syntax compatibility (C++)**
To simplify migration of our users we make PostgreSQL compatibility
- **OLAP engine (C++)**
We build OLAP engine to improve analytical experience
- **Column-organized tables (C++)**
We make column-organized tables to improve performance of analytical queries
- **K8S deployment (Go)**
We make our own k8s operator to make it easy to deploy our database in a cloud environment



ydb.tech



twitter.com/YDBPlatform



t.me/ydb_en

Thank you!



Anton Kovalenko
YDB Technical Project Manager
kovalad@ydb.tech