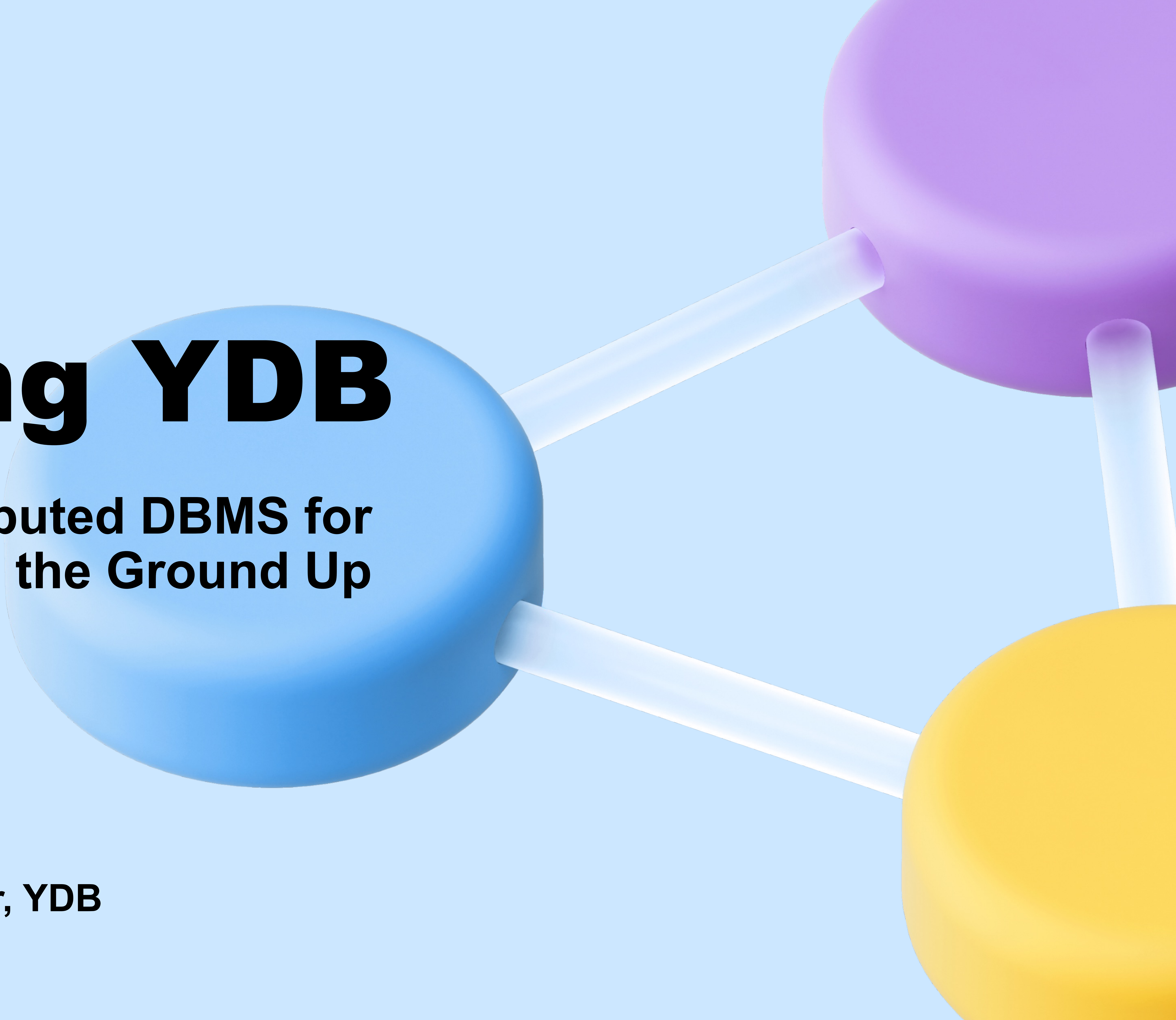
YDB foss asia



Designing YDB

Constructing ả Distributed DBMS for OLTP and OLAP from the Ground Up

Evgenii Ivanov

Principal Software Developer, YDB

About myself

- YDB developer
- Amateur speaker
- Outside YDB I enjoy spending time with my family, aerial photography, and reading

Rumors about YDB and YugabyteDB

- Many believe that YDB and YugabyteDB are the same thing
- Others say we once had a bar fight

The truth

- YDB and YugabyteDB are **different** distributed DBMSs
- We enjoy discussing topics related to benchmarking and distributed systems

YDB is a platform

1

Originally OLTP

2

YDB Topic Service (kafka like)

ACID transactions between topics and tables

3

OLAP

4

And more

Open-Source Distributed SQL Database

1 Relational DB: both OLTP and OLAP

2 Clusters with thousands of servers

3 Apache 2.0 license

4 Star ydb-platform on GitHub

Strictly consistent

1

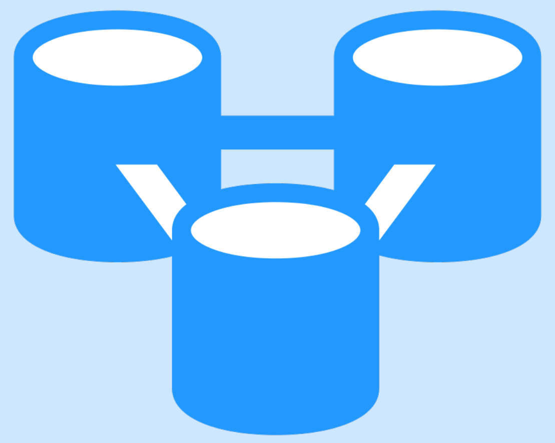
CAP-theorem —
YDB chooses CP

2

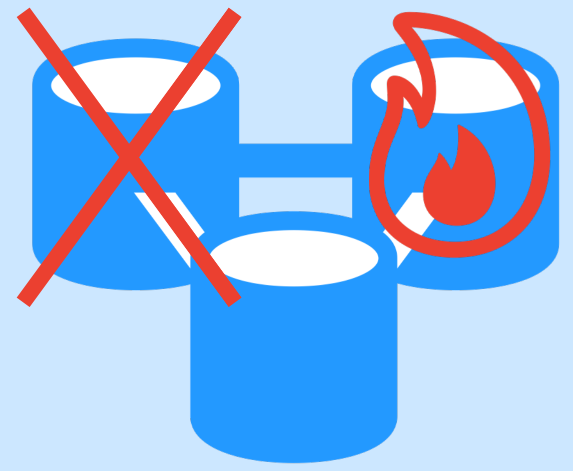
Serializable transaction
execution

Highly available and fault tolerant

Multiple availability zones (AZ): automatic recovery



YDB is read-write available even after losing an AZ and a rack simultaneously



A mission critical database

1

365x24x7 (366x24x7 when needed)

2

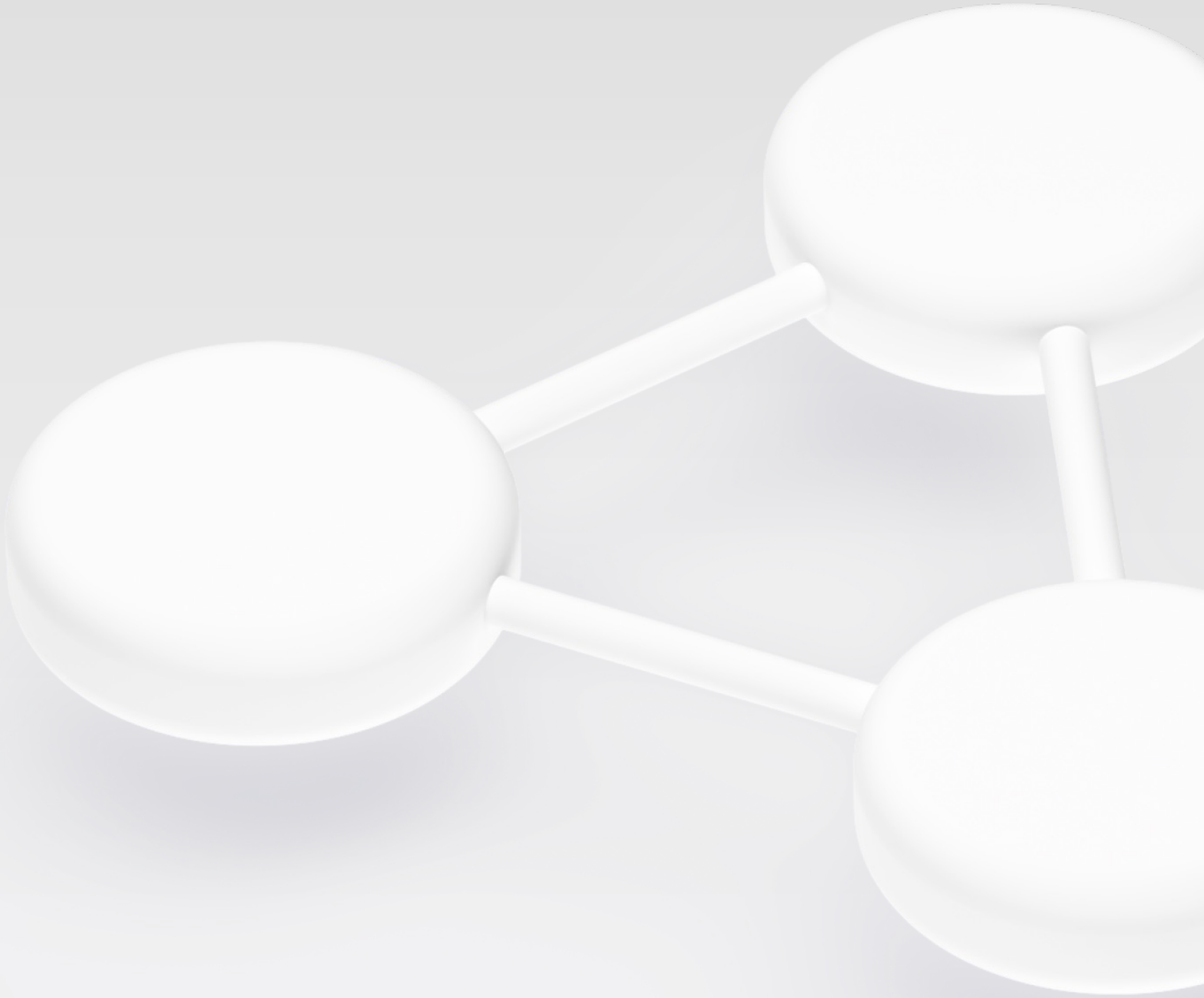
No downtime during a maintenance (e.g. to roll out a new YDB version)

Fun fact: YDB is bootstrappable in the cloud

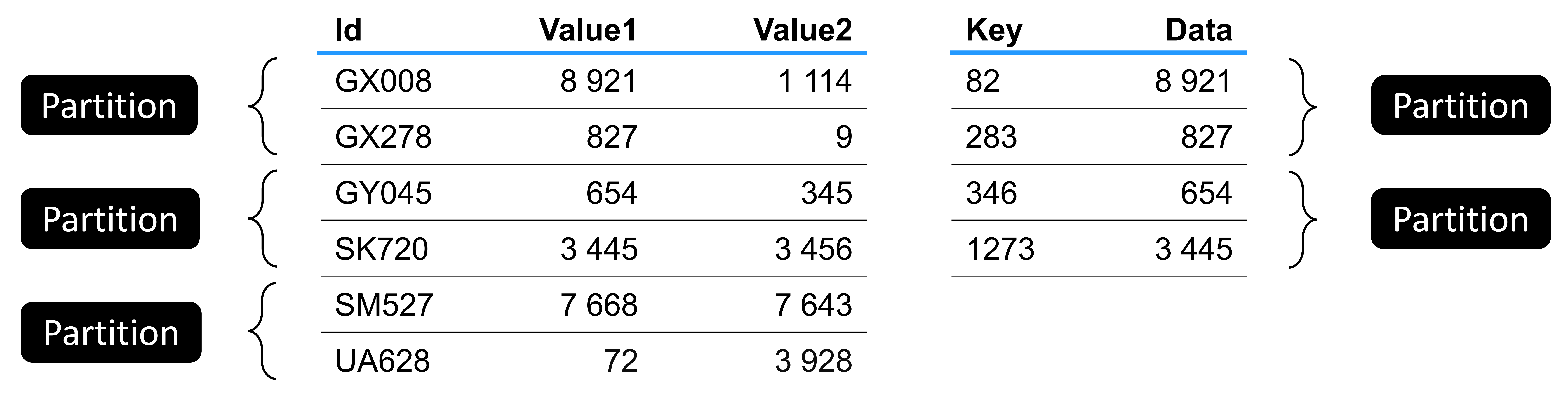
- Some clouds use YDB to store their metadata
- Often their Network Block Store is implemented over YDB
- When you get a YDB database as a service in the cloud, it is **YDB over YDB over YDB**

YDB

Spaceship View



Tables, Partitioning, Queries



Id	Value1	Value2
GX008	8 921	1 114
GX278	827	9
GY045	654	345
SK720	3 445	3 456
SM527	7 668	7 643
UA628	72	3 928

Key	Data
82	8 921
283	827
346	654
1273	3 445

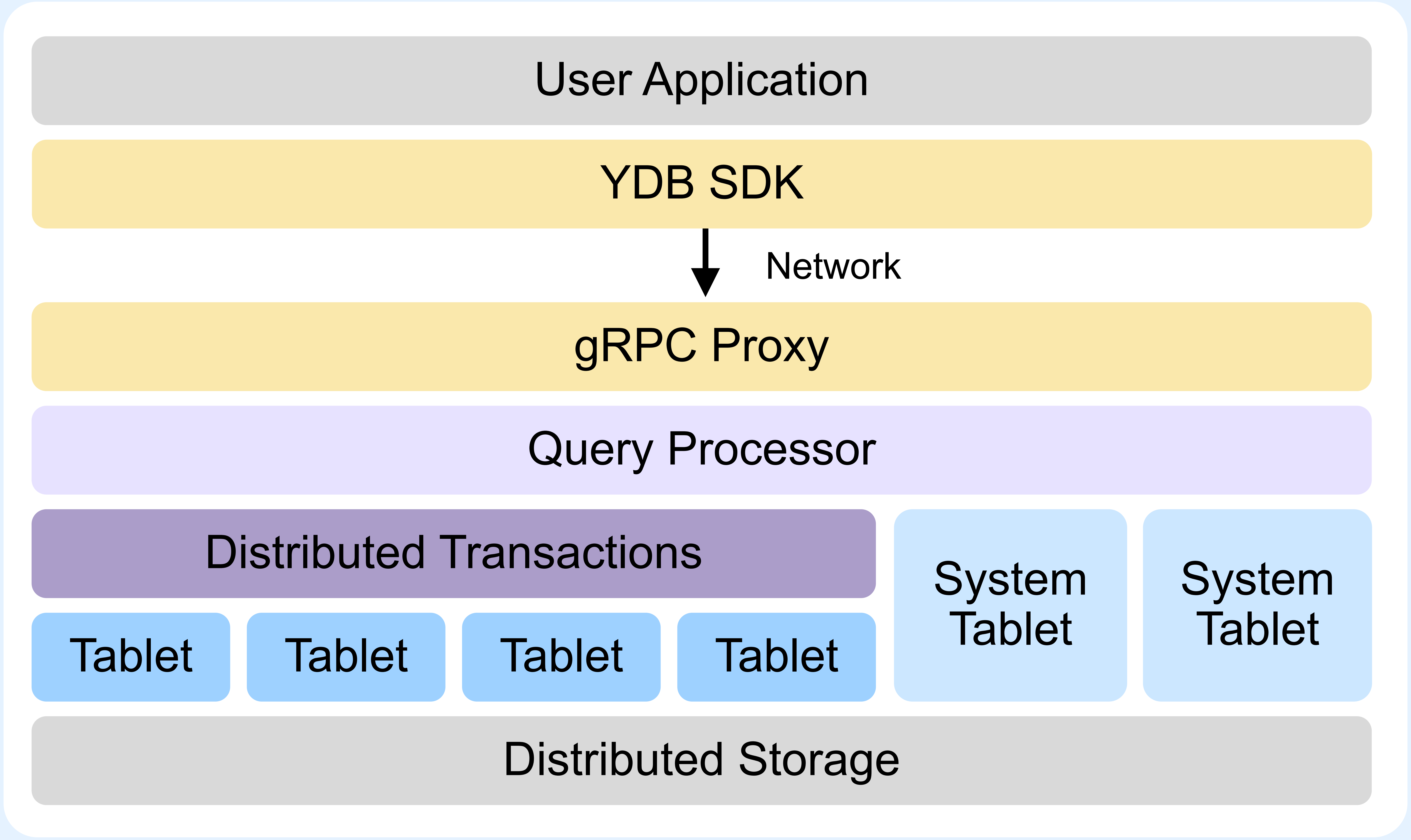
```
UPDATE table1 SET Value1=38 WHERE Id="GY045";
UPDATE table2 SET Data=Data+1 WHERE Key=346;
COMMIT;
```

Tables have a primary key (PK), tables
are sorted by PK.

All tables data are split into partitions,
partitions
are stored in Tablets.

YDB

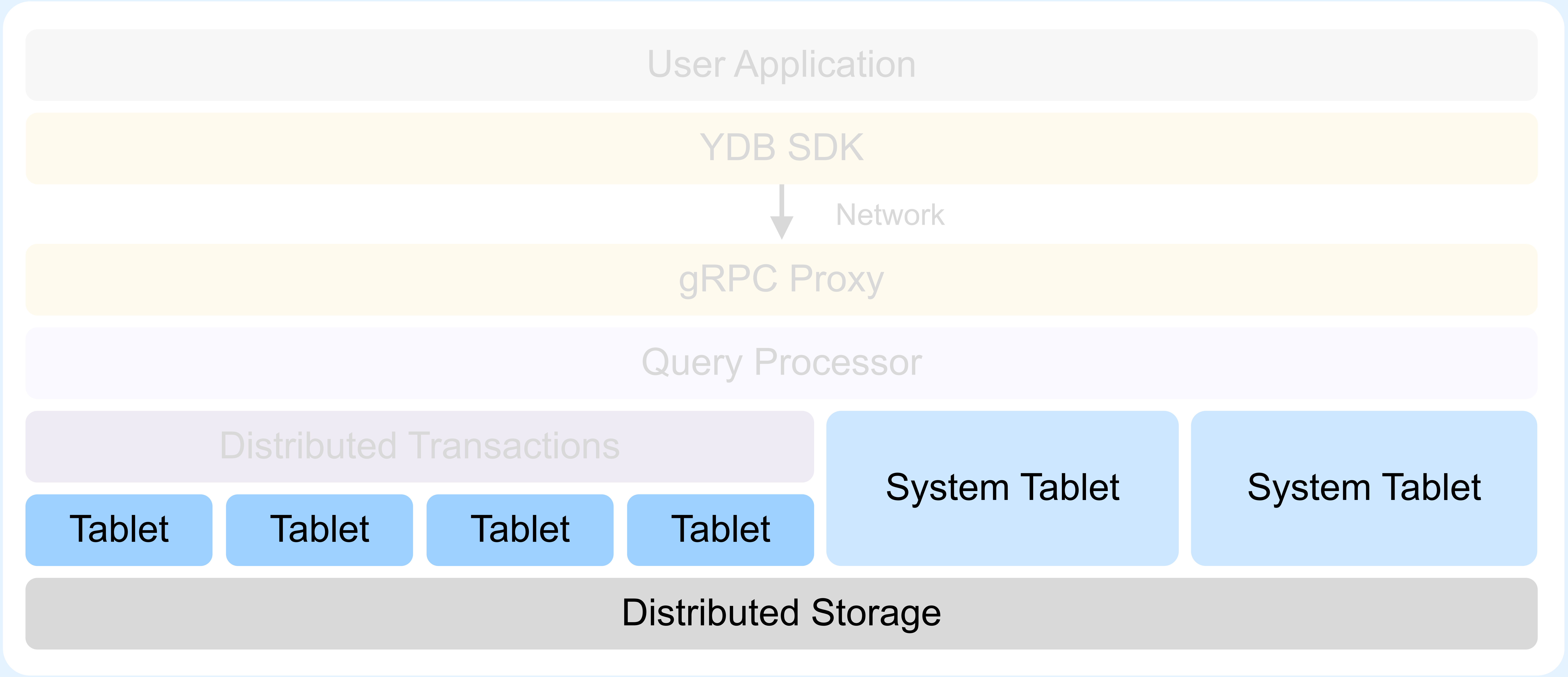
YDB Logical Architecture



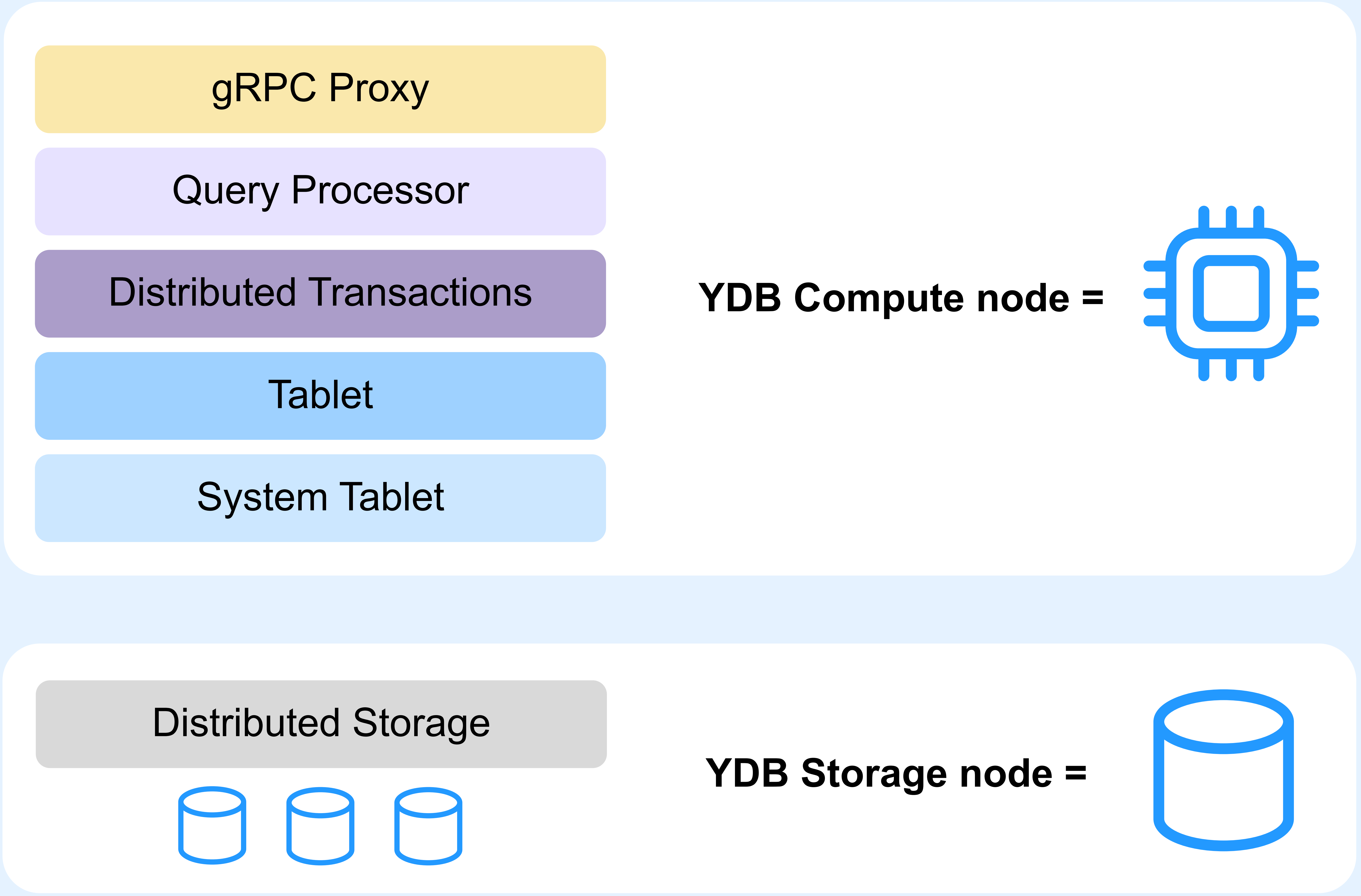
Layered Architecture

- Distributed Storage: data redundancy/ replication and consensus
- Tablet is a reliable component
- ACID distributed transactions between tablets

YDB platform components

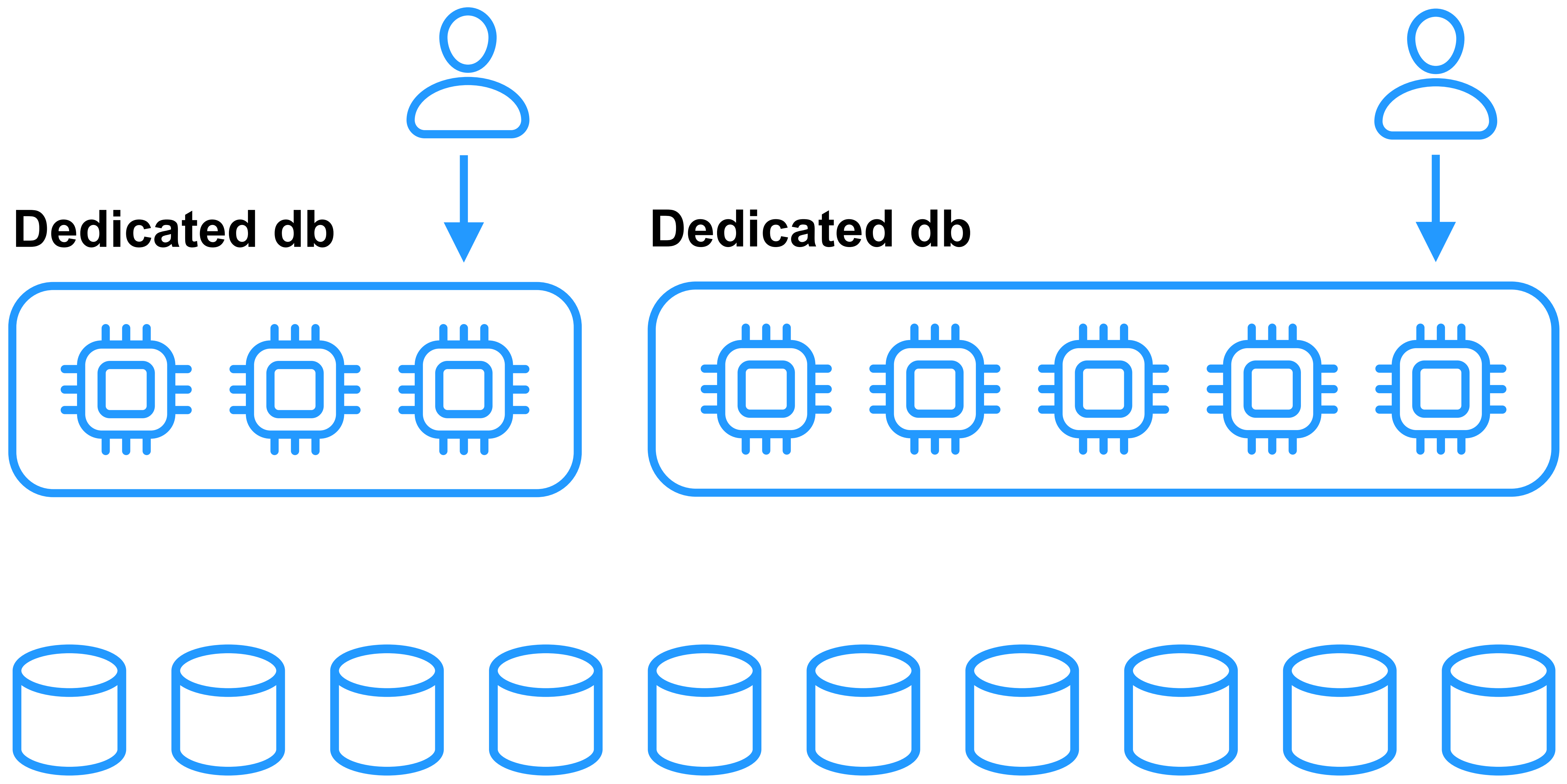


Separate compute and storage



- Share nothing architecture
- Commodity hardware
- Compute and storage scale independently
- Run in virtual machines or containers or bare metal

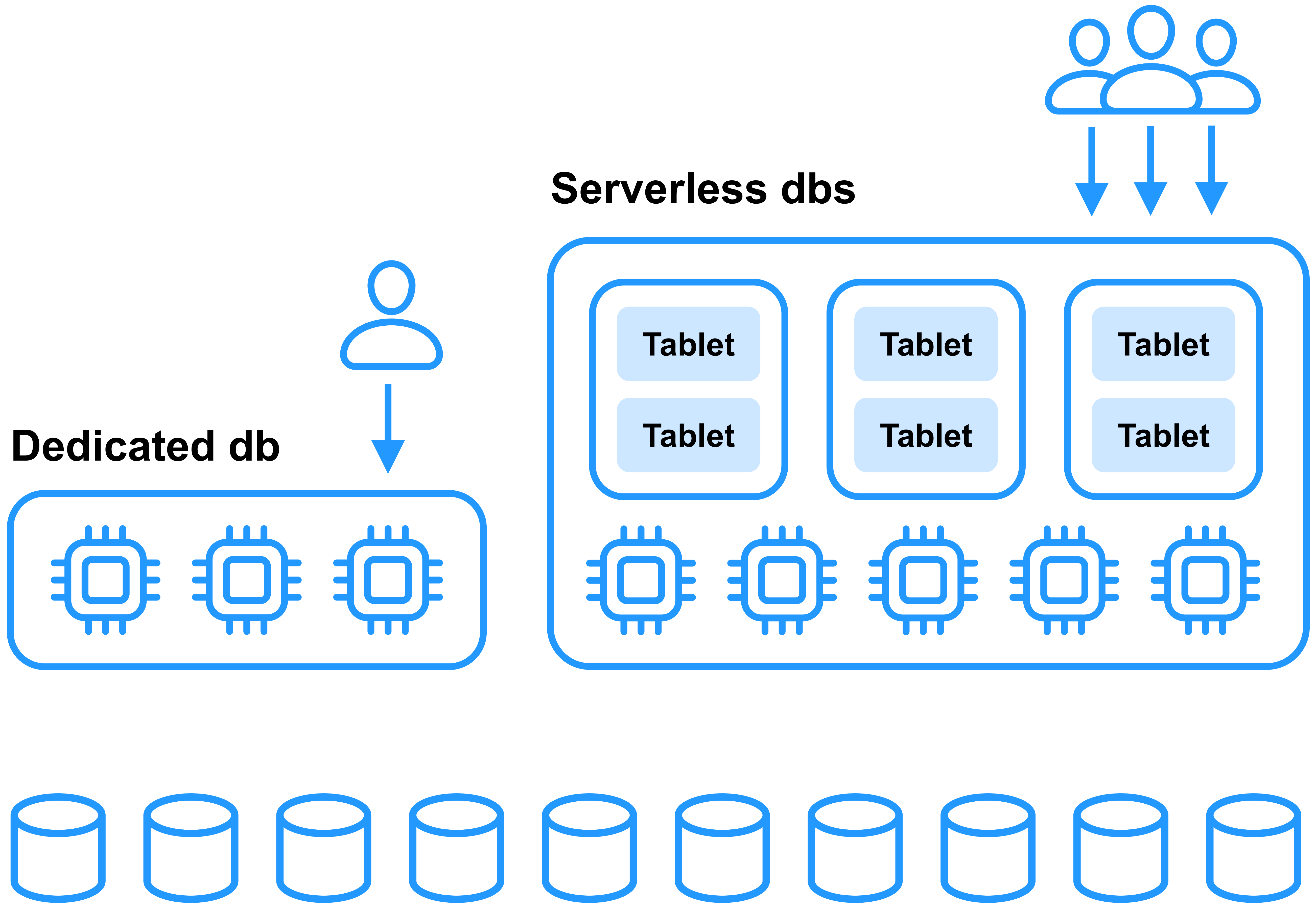
YDB Cluster



A database has dedicated compute nodes, large YDB clusters have thousands of databases

Storage is shared between databases

YDB Cluster

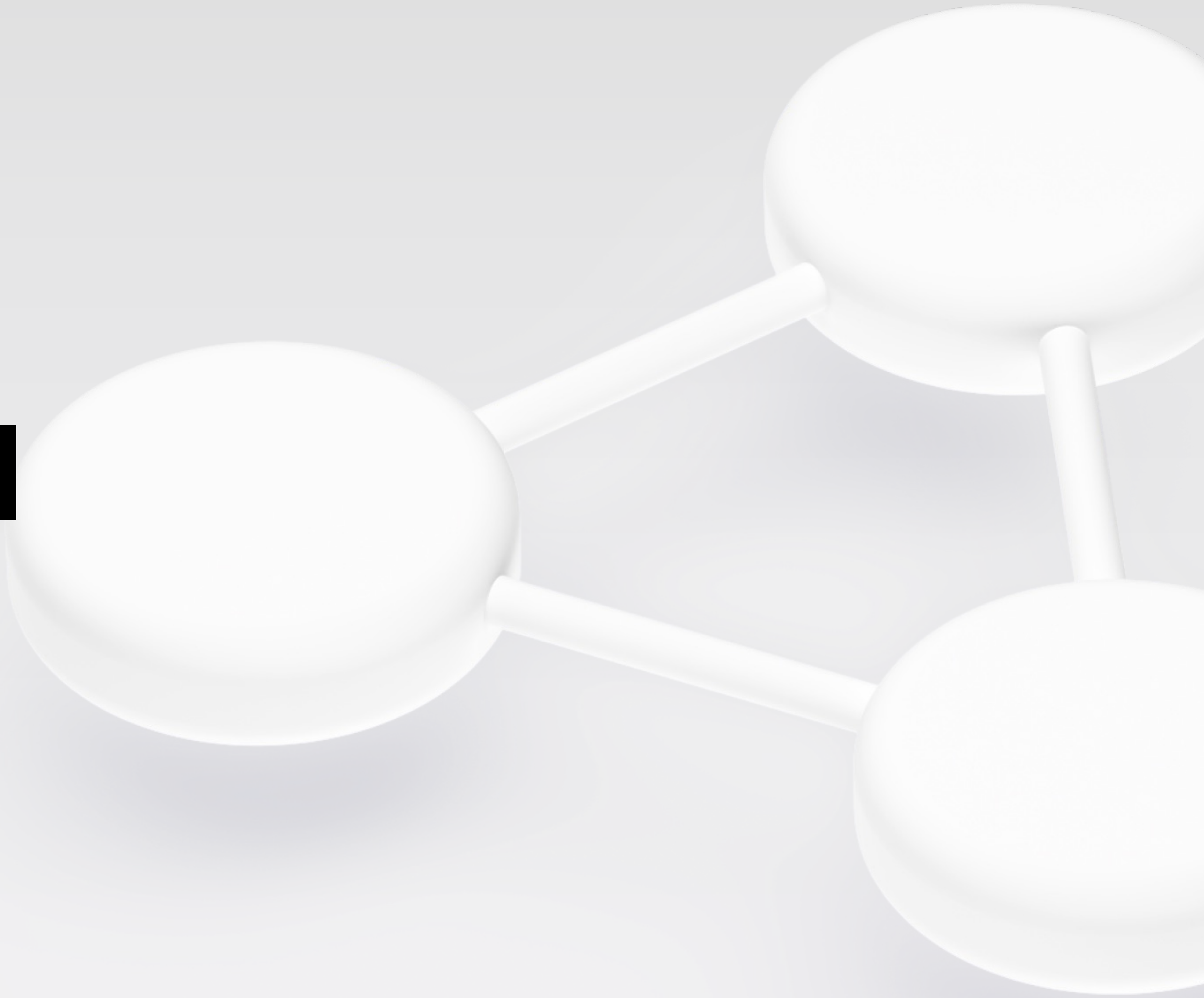


Dedicated database can host several serverless databases. In this case dedicated database is called a shared database.

Storage is shared between databases

YDB

Distributed Storage



Distributed Storage (Blob storage)

YDB Distributed Storage
is a special purpose distributed key value store for immutable blobs

From 1B to 10MB
stores immutable blobs

Tablets use Distributed Storage for:

1. Writing log records, i.e. heavy writing and rare reading (range based)
2. Storing standalone blobs or parts of tablet’s LSM tree

Redundancy schemes

Erasure coding

- Single AZ
- Block4-2: 4 parts + 2 parity
- Just 1.5x redundancy

Replication

- Three AZ (Mirror-3-DC)
- 3 replicas
- x3 redundancy

Other

More could be added

”Special Purpose KV-store” Means

```
Key = [TabletId, Generation, Step,…]
Value = <ArbitraryBlob>
```

Distributed Storage API

`Put(Key, Blob)`

`Get((Key, offset, size), …)`

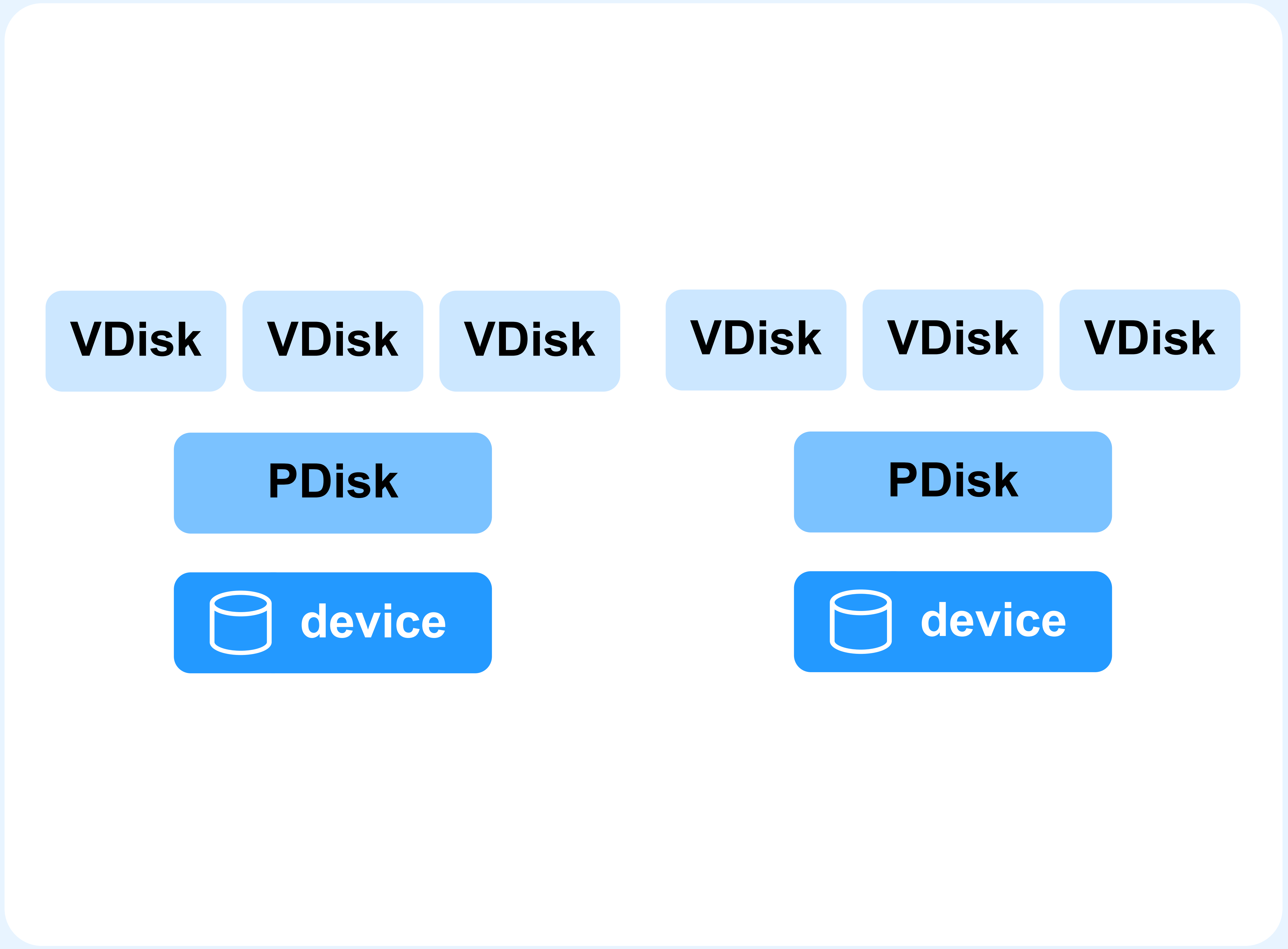
`Block(TabletId, Generation)` — write to storage, gather a quorum to become a tablet leader

`Discover` — find a last written to log record, make sure it is written in all replicas

`CollectGarbage(TabletId, Generation, Step)` — used to remove old blobs by moving garbage collection barrier ahead

YDB

Distributed Storage Node

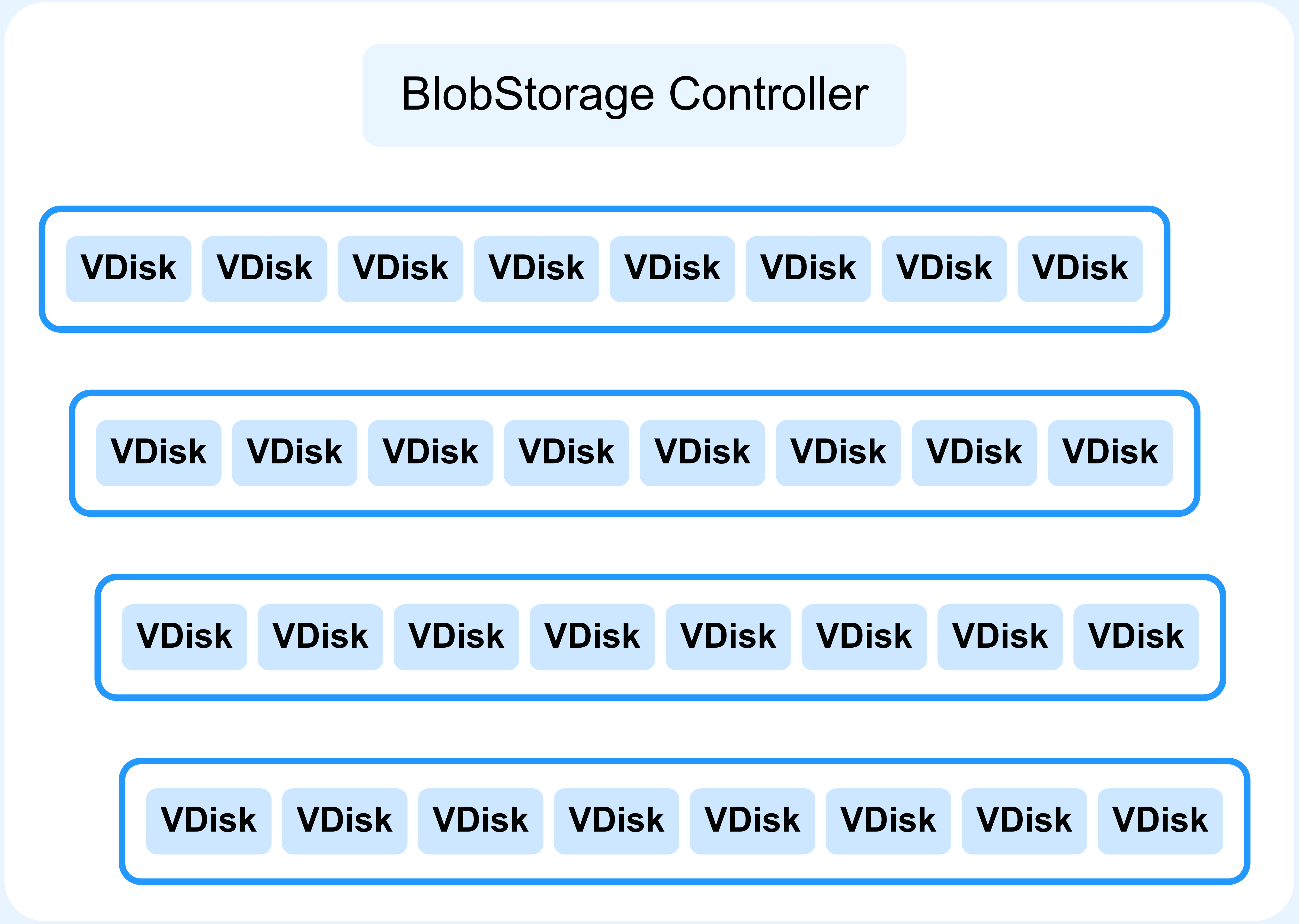


PDisk owns block device and

- Manages chunks of fixed size
- Optimized for log writing
- Has a scheduler that allows to distribute disk throughput evenly between VDisks

Several VDisks usually run over a single PDisk

Distributed Storage Structure


BlobStorage Controller
VDisk VDisk VDisk VDisk VDisk VDisk VDisk VDisk
VDisk VDisk VDisk VDisk VDisk VDisk VDisk VDisk
VDisk VDisk VDisk VDisk VDisk VDisk VDisk VDisk
VDisk VDisk VDisk VDisk VDisk VDisk VDisk VDisk

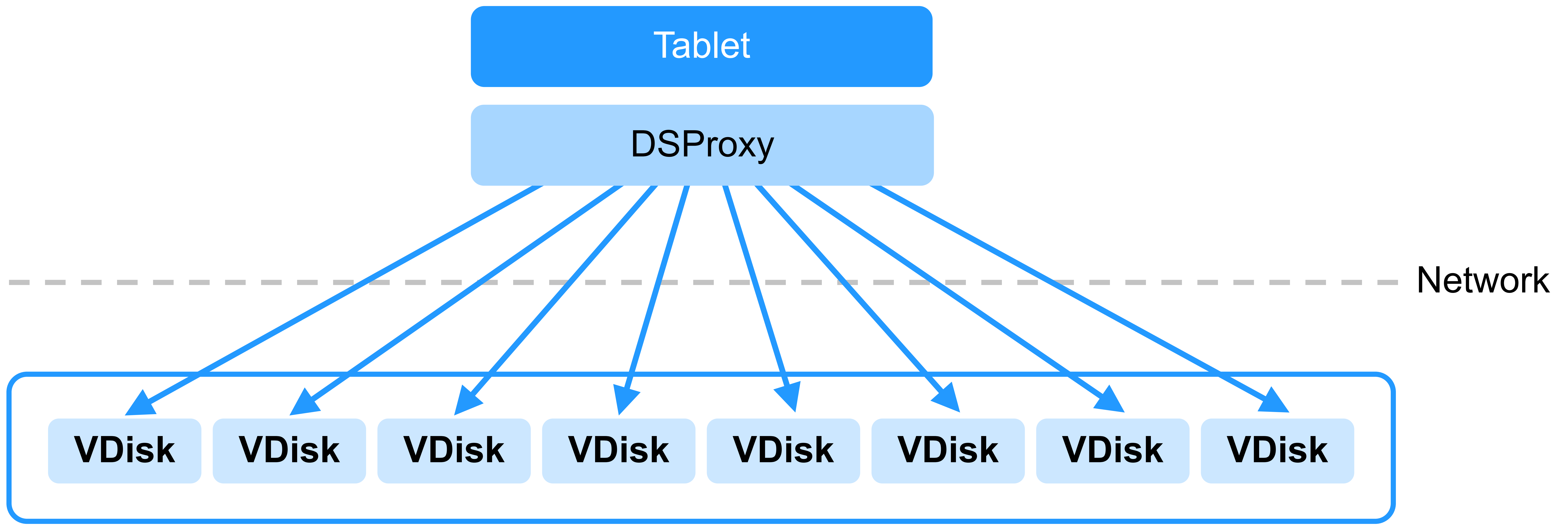
Distributed Storage is build of

- several BlobStorage Groups
- and BlobStorage Controller – a special tablet that manages Distributed Storage metadata

BlobStorage group is a reliable storage entity built from unreliable VDisks

The easiest way to understand BlobStorage group is to think about it as a **Distributed RAID**

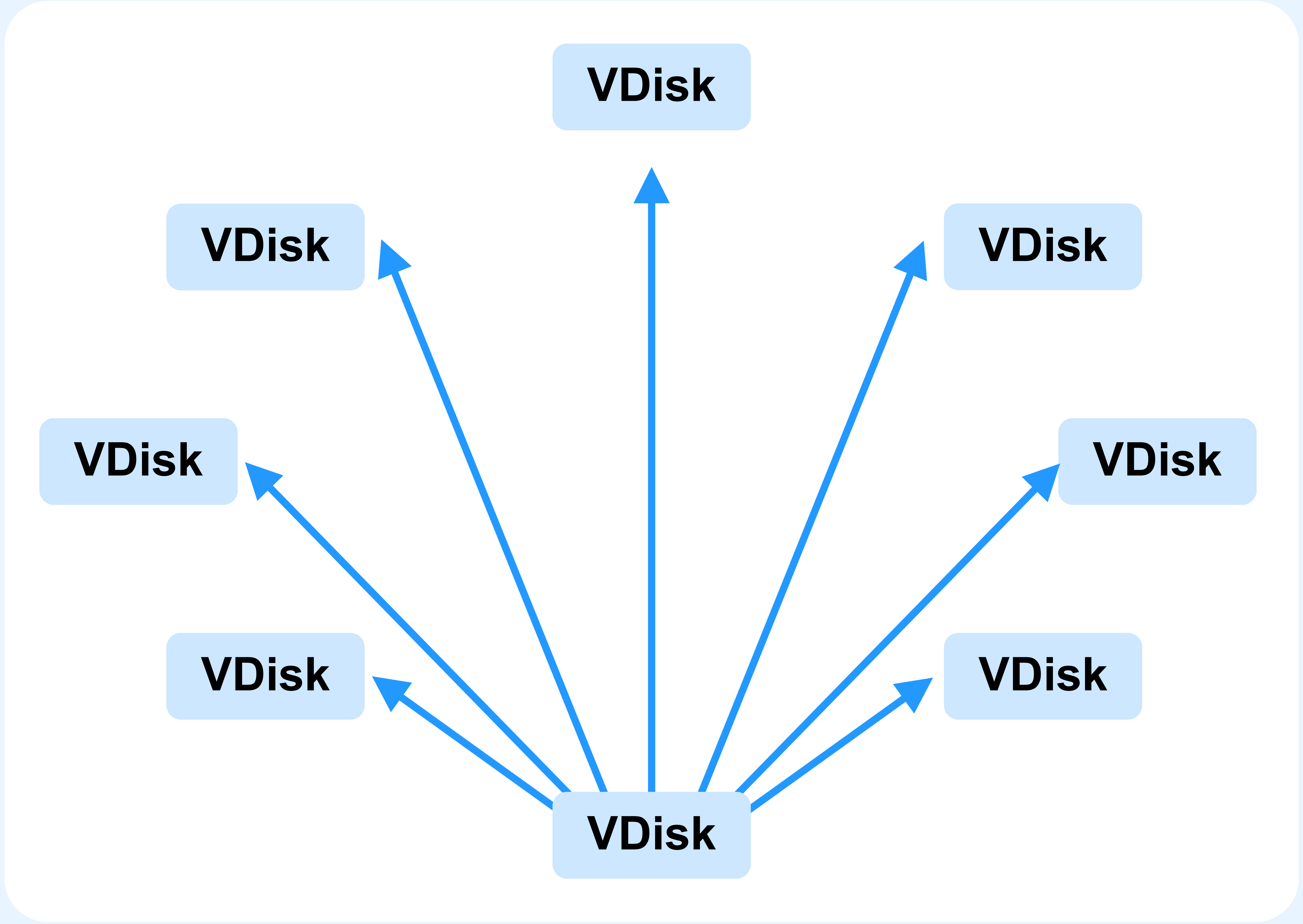
Distributed Storage Group



DSProxy communicates with remote VDisks, handles network and disk failures

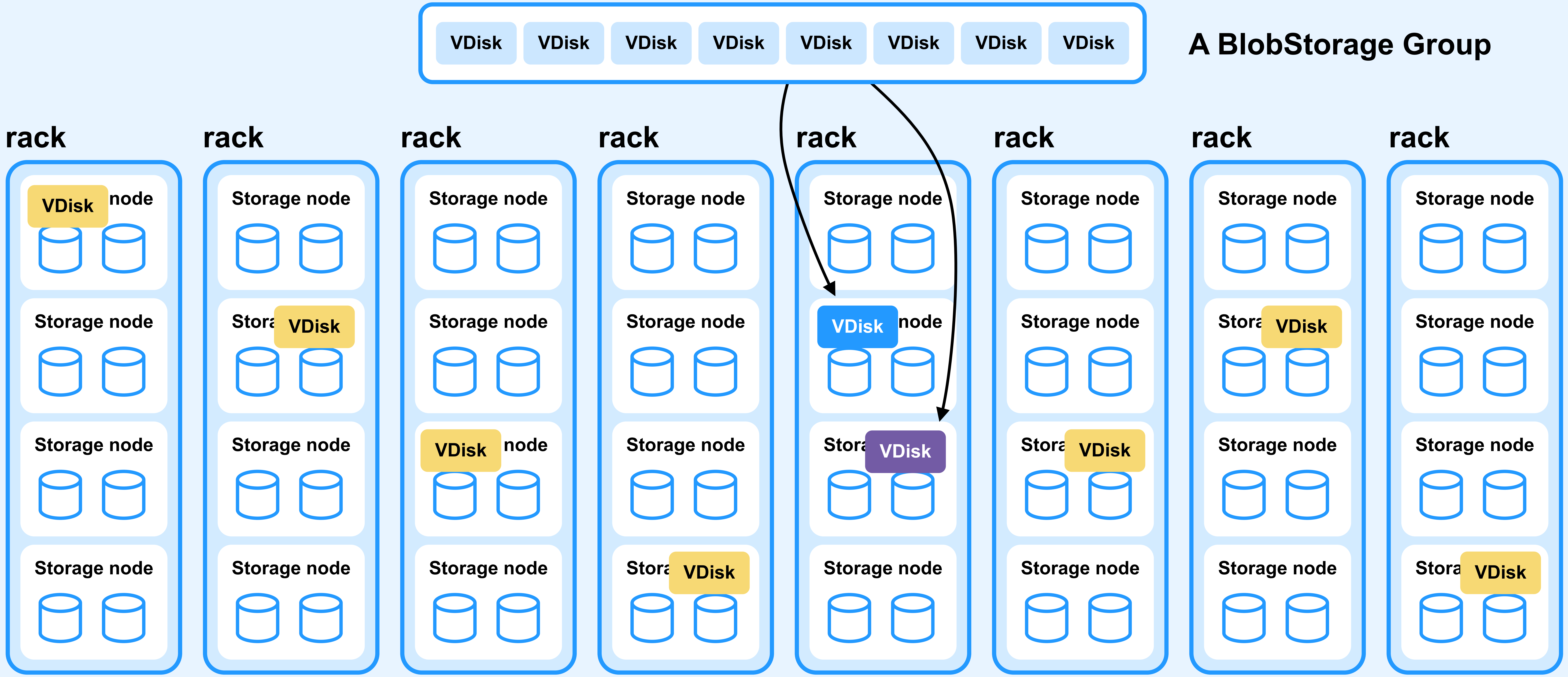
Tablet is attached to one or more BlobStorage Groups. Tablet works with BlobStorage group via local DSProxy component, which provides Distributed Storage API to the tablet

VDisk



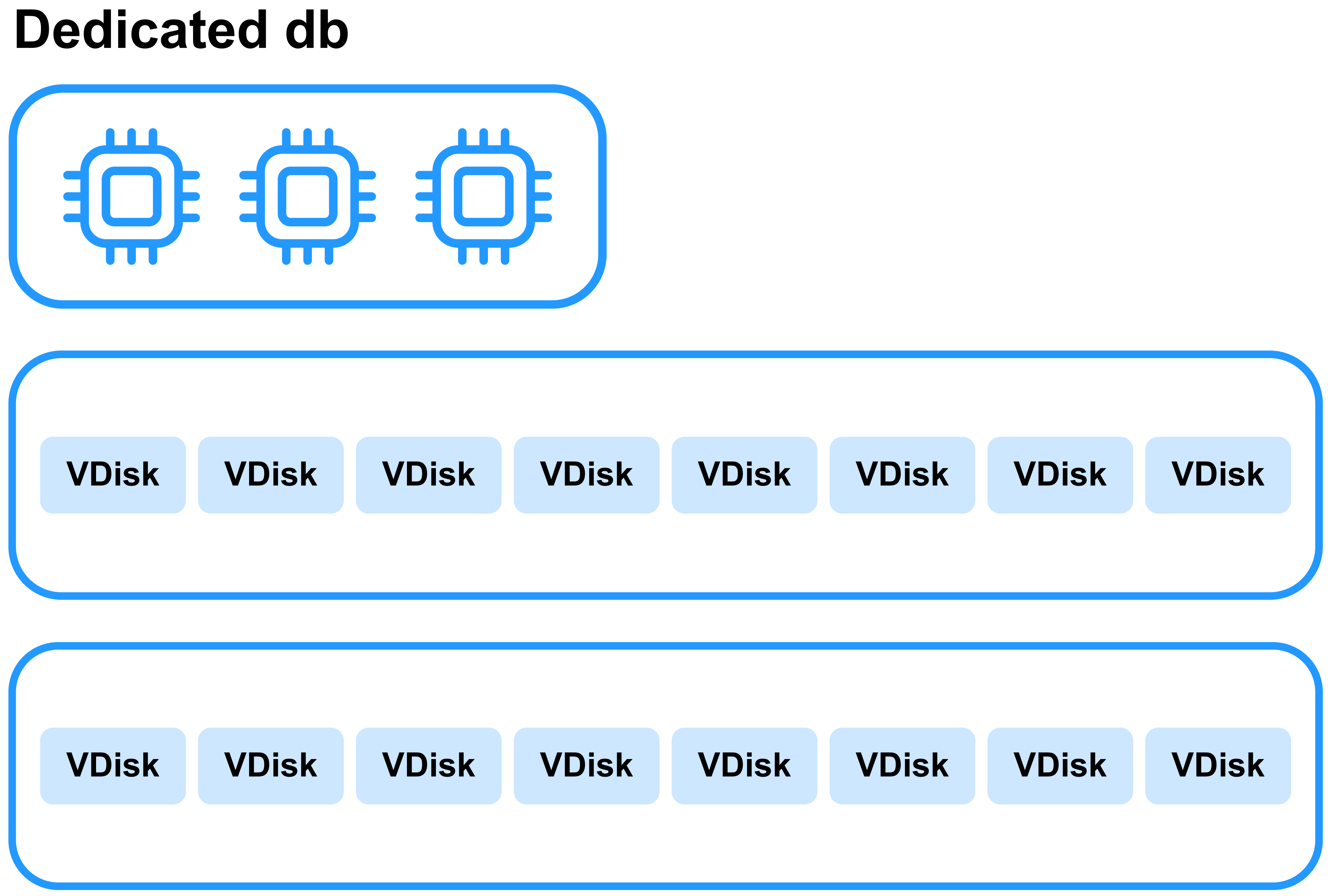
- Stores blobs locally on disk
- Built as a local KV-store
- Communicates **peer-to-peer** to other VDisks in group for synchronization
- In case of device failure automatically replicates data from other VDisks in the BlobStorage Group

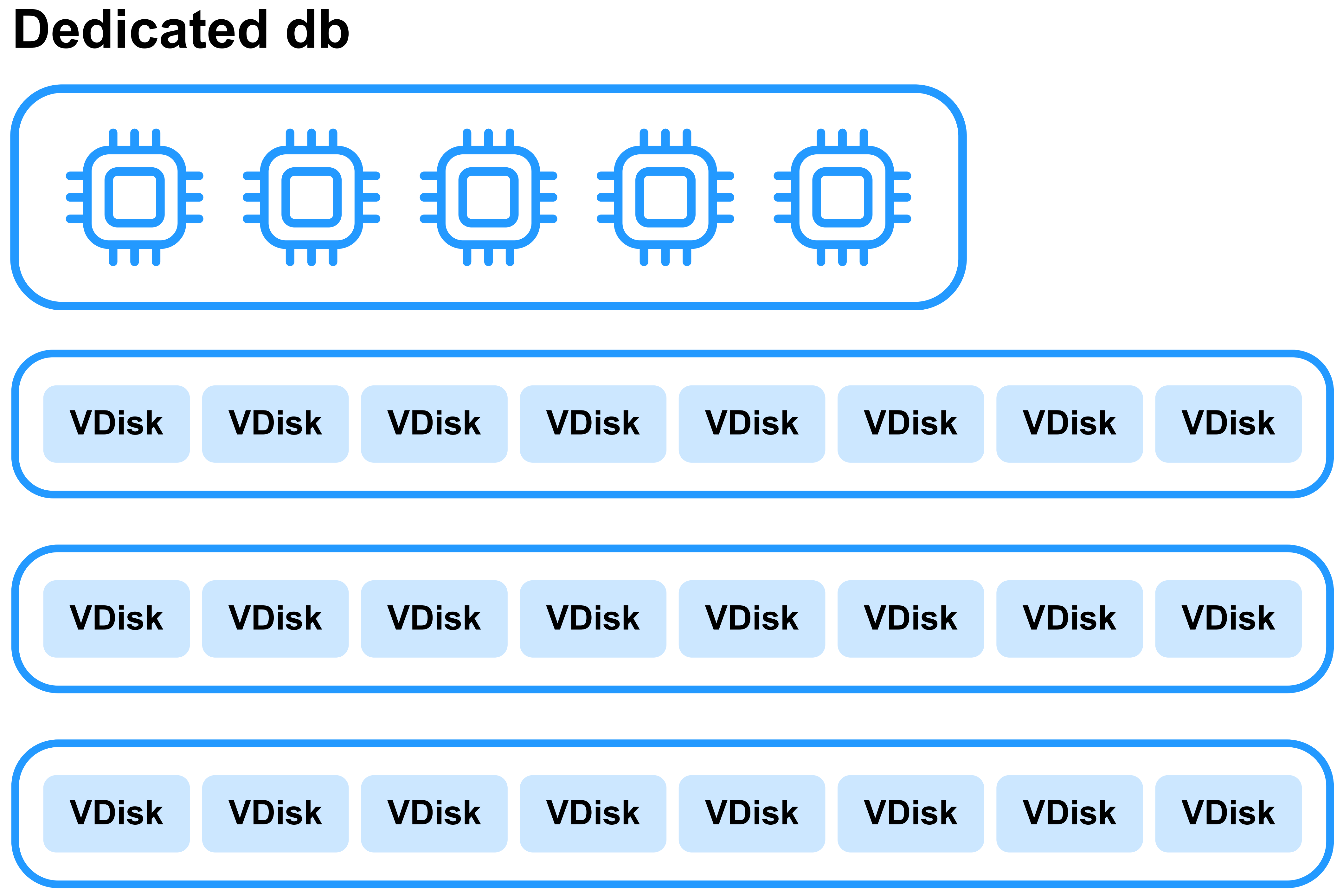
BlobStorage Group Reconfiguration



YDB

Distributed Storage Users Isolation


Dedicated db
VDisk VDisk VDisk VDisk VDisk VDisk VDisk VDisk
VDisk VDisk VDisk VDisk VDisk VDisk VDisk VDisk

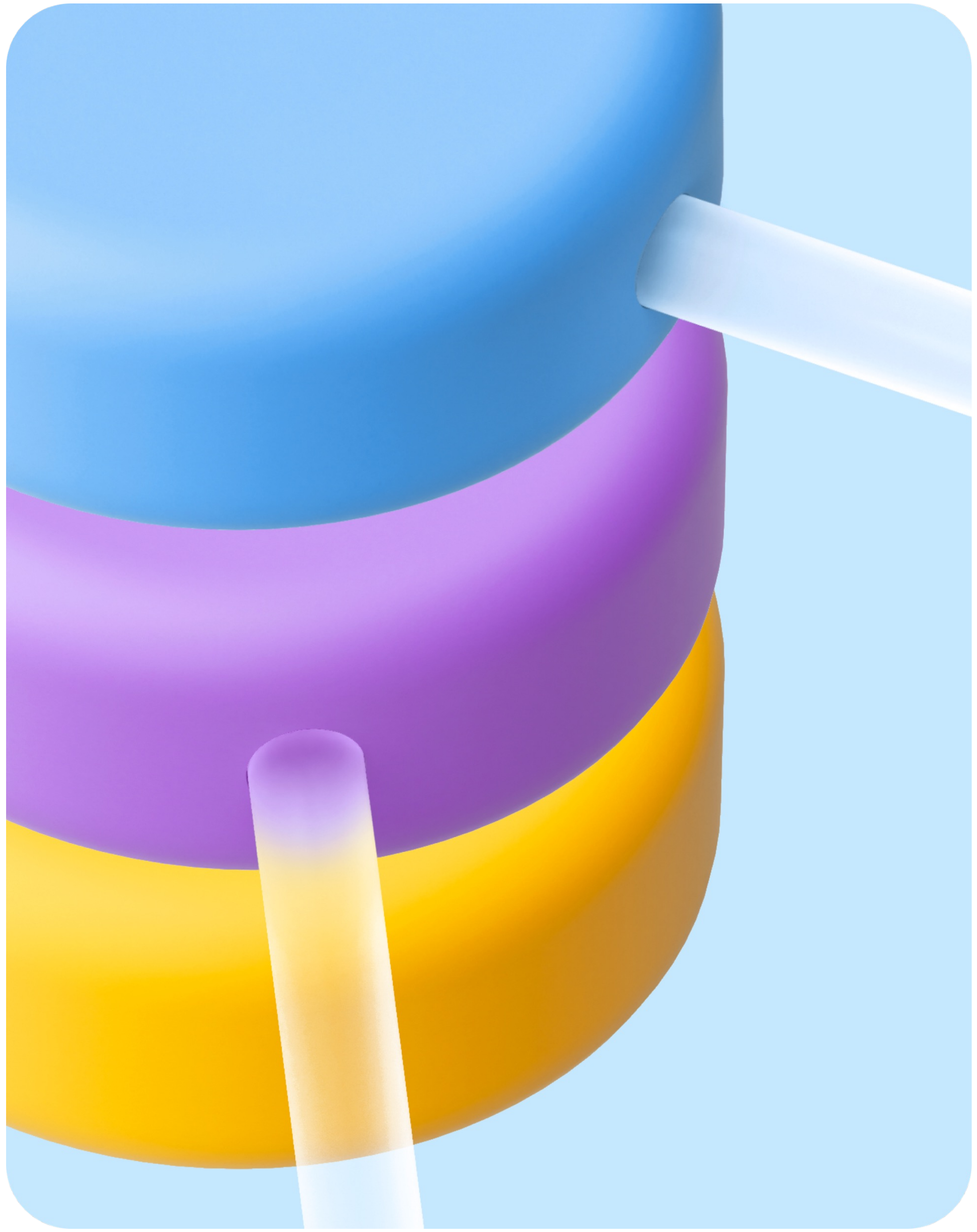

Dedicated db
VDisk VDisk VDisk VDisk VDisk VDisk VDisk VDisk
VDisk VDisk VDisk VDisk VDisk VDisk VDisk VDisk
VDisk VDisk VDisk VDisk VDisk VDisk VDisk VDisk

Dedicated databases have their own **pool** of BlobStorage Groups, databases can still share the same devices

There is a conception of **Box** that owns physical devices, pools from different boxes do not intersect by disks

Distributed Storage Fault Tolerance

- If a device is broken and replaced, replication starts automatically
- Self-heal tracks VDisk unavailability and runs BlobStorage Group reconfiguration, i.e. removes a broken VDisk from the group and adds a new one



YDB

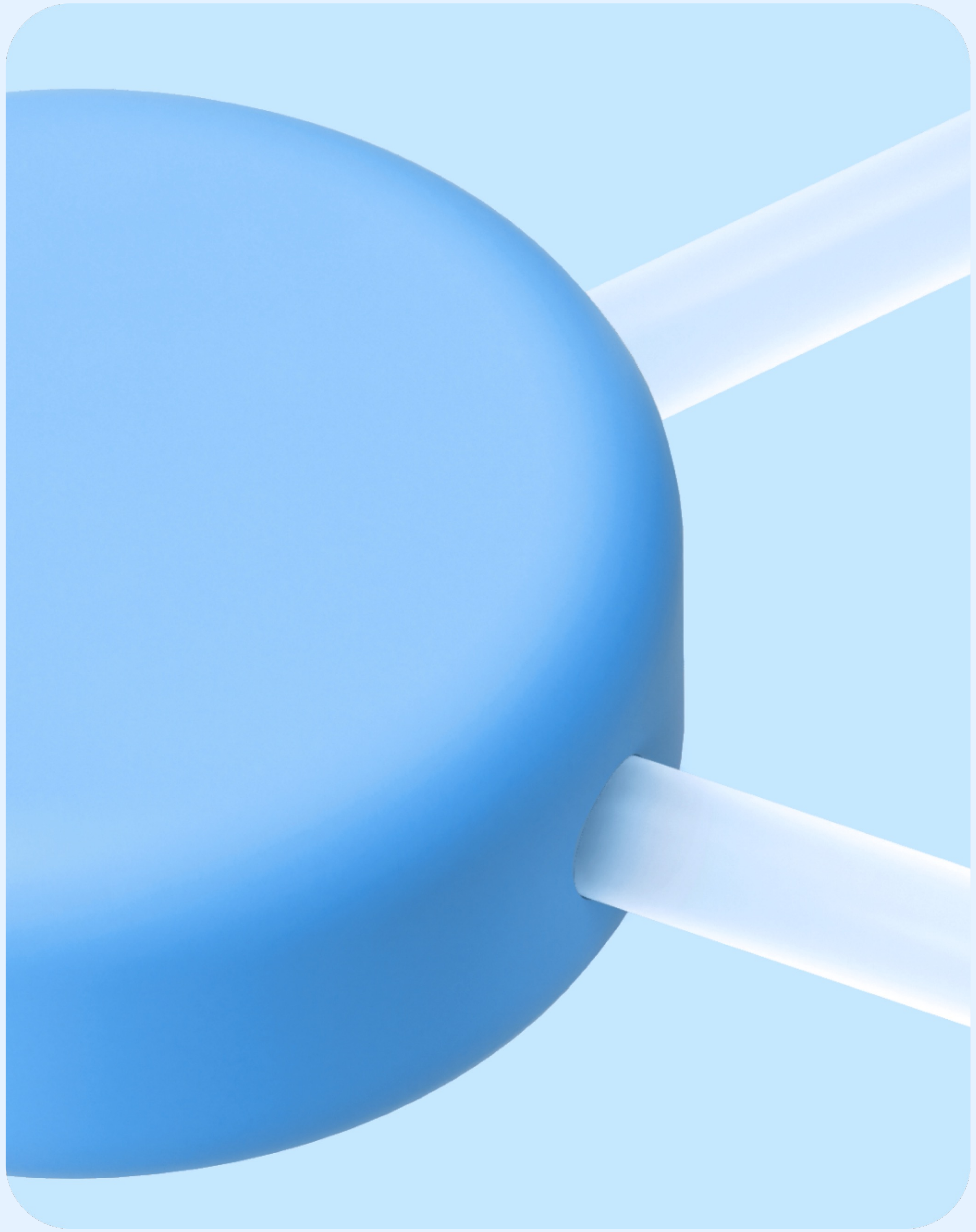
Distributed Storage Scalability

BlobStorage groups

are completely independent,
so could scale infinitely

BSC handles 10K

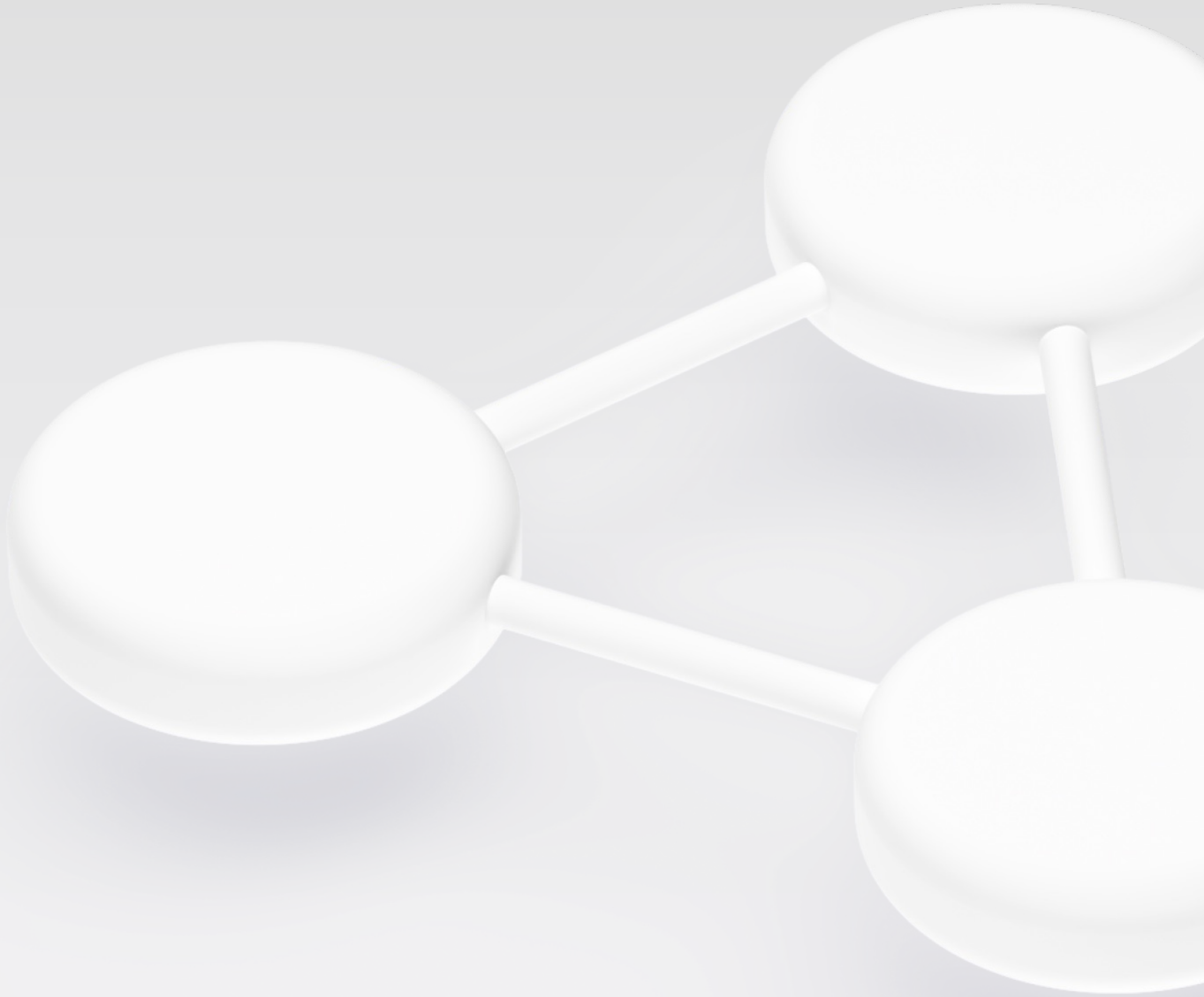
storage nodes without much CPU load



YDB

YDB

Tablets



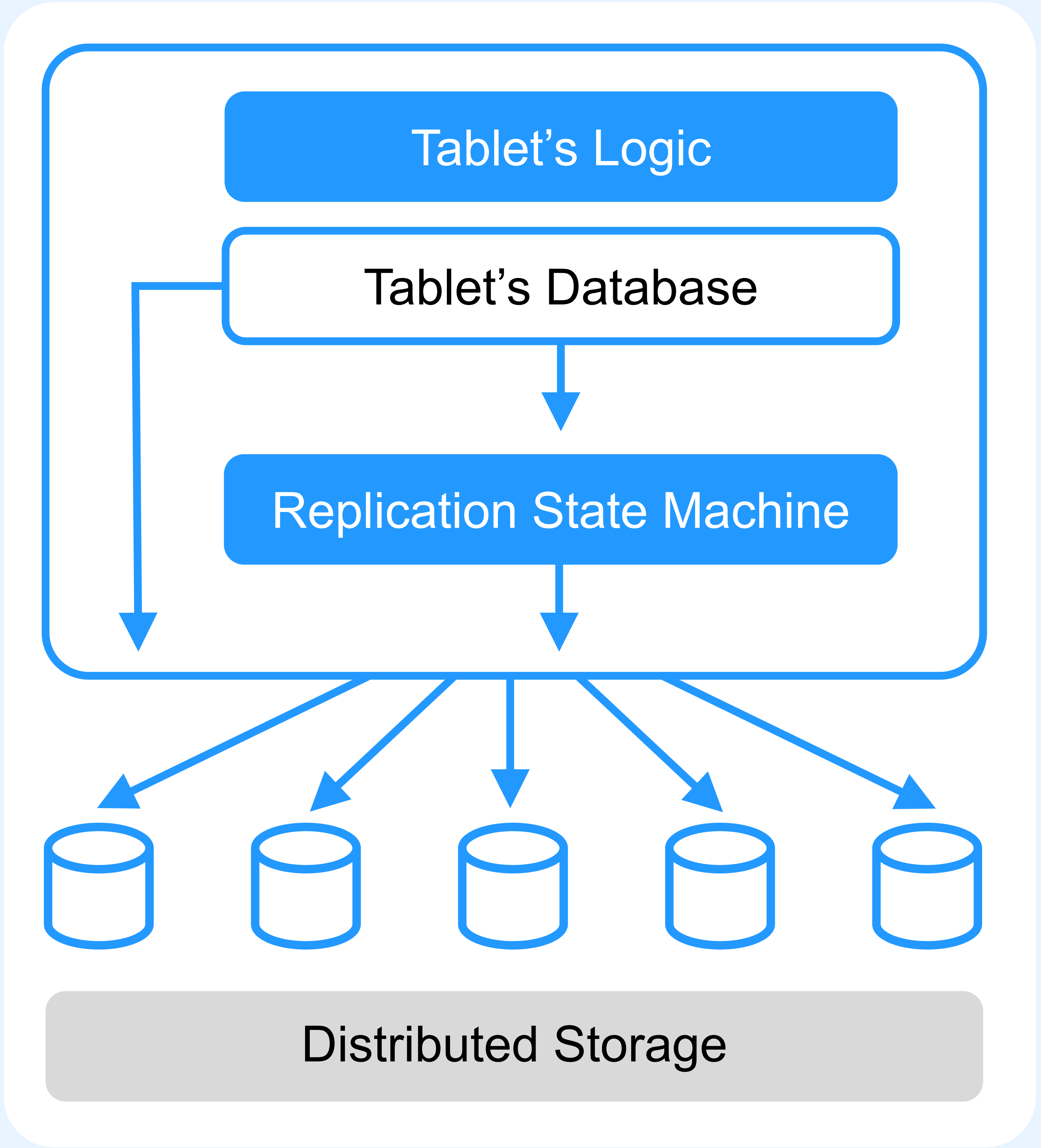
YDB Tablet

YDB Tablet
incapsulates
a solution for reliable
stateful building block

YDB Tablets run
in compute nodes

If a node that runs
a tablet dies,
YDB infrastructure
is responsible
for recovering
the tablet in exactly
same state

Inside Tablet



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

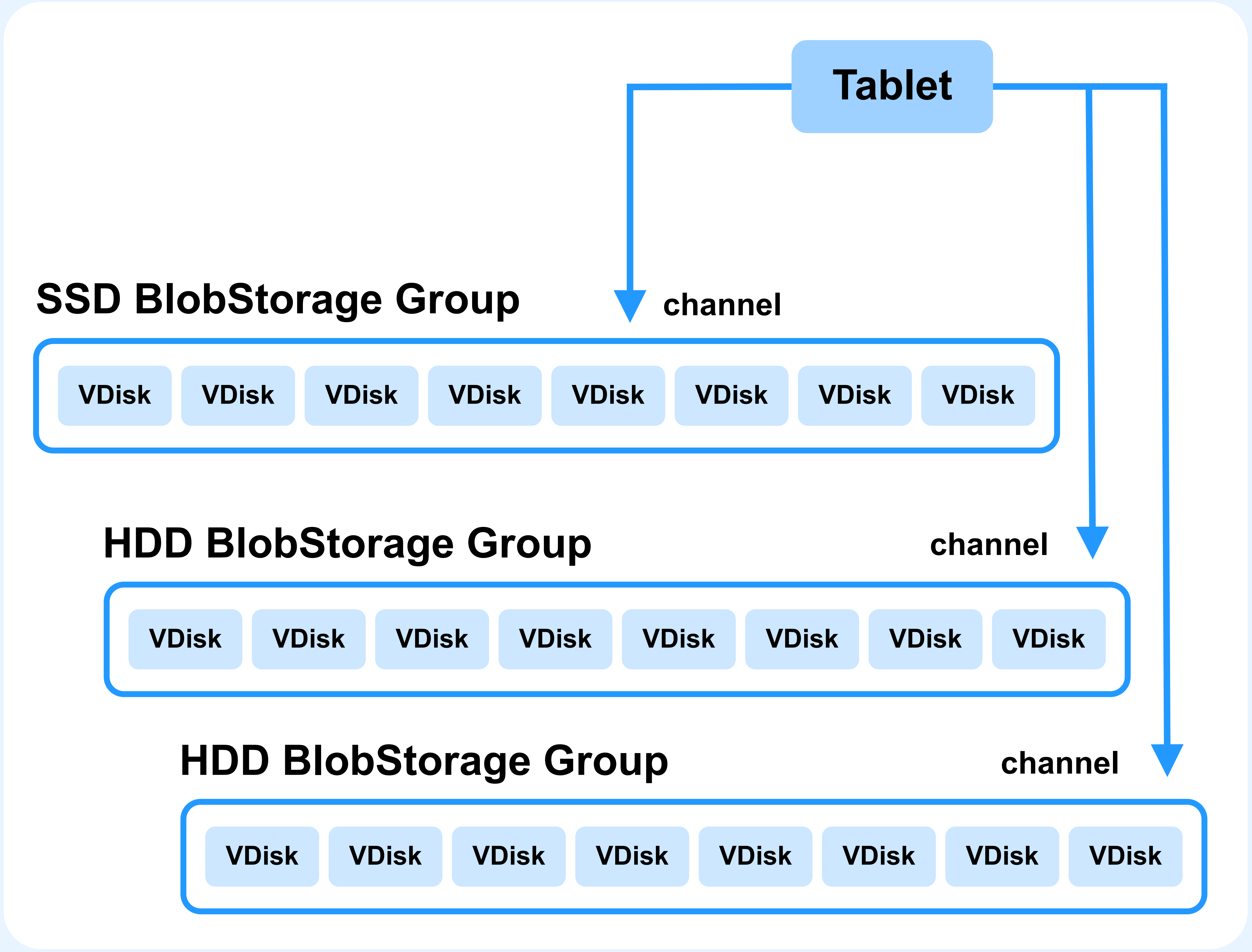
Tablet's Logic is specific for the Tablet type

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

Distributed storage provides
reliable data storage with redundancy

YDB

Tablet Channels



Tablet
SSD BlobStorage Group
channel
VDisk VDisk VDisk VDisk VDisk VDisk VDisk VDisk
HDD BlobStorage Group
channel
VDisk VDisk VDisk VDisk VDisk VDisk VDisk VDisk
HDD BlobStorage Group
channel
VDisk VDisk VDisk VDisk VDisk VDisk VDisk VDisk

Tablet has multiple channels that can be attached to the same or different BlobStorage Groups

Multiple channels give scalability and flexibility

- Amount of data stored by tablet
- Read/write throughput
- Support different media types in one tablet (for instance, table column groups are used to put some columns to SSD, while other columns to HDD)

Channel 0

always exists and reserved for Tablet's Log

Tablet Types

DataShard

A partition of a user table, supports SQL queries execution

ColumnShard

Our column store for OLAP workloads, supports SQL queries execution

SchemeShard

Stores user tables metadata

Hive

Manages other tablets in a database

Coordinator/Mediators

Used for distributed transaction scheduling

TxAllocator

Generates unique transaction identifiers

Cluster Management System

- Helps maintenance YDB cluster
- Answer the question «may I shut down this particular node»

SysView Processor

Manages system tables that provides statistics for user

YDB foss asia

Questions?

Evgenii Ivanov

Principal Software Developer, YDB

@eivanov89

ydb.tech

@YDBPlatform

@YDBPlatform