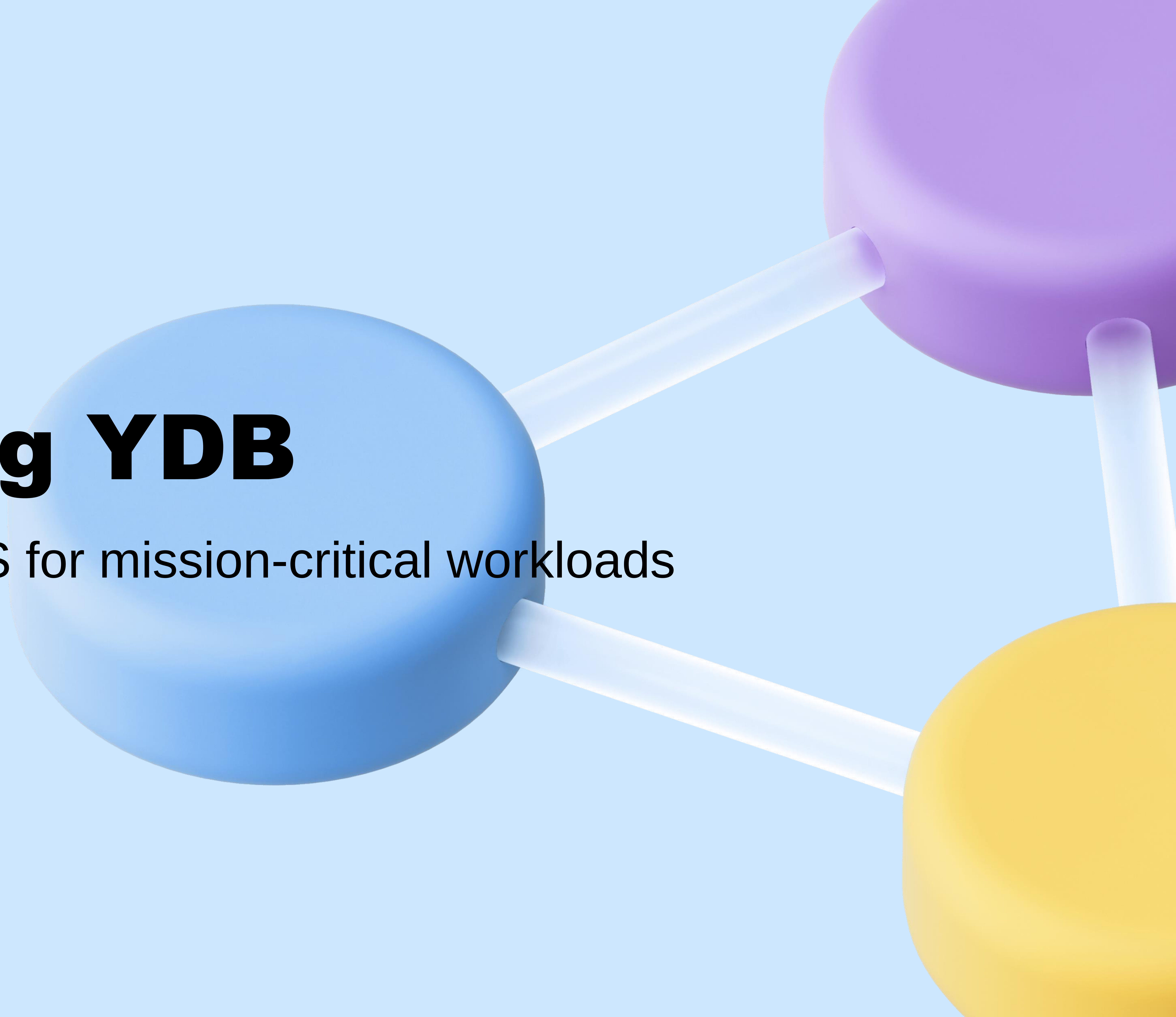
YDB



Introducing YDB

Distributed SQL DBMS for mission-critical workloads

Ivan Blinkov
VP, Product and Open-Source

Ivan Blinkov

- Over a decade of experience in the database management systems (DBMS) development industry
- Talked with countless DBMS users and stakeholders to understand how and why they ended up with a specific solution
- Worked on a handful of DBMS products, including two open-source ones:

ClickHouse YDB

Agenda

1

Technology overview

2

Design and architecture

3

Open-source

YDB technology overview

YDB: Open-Source Distributed SQL Database

Mission critical

- Designed for services with 24×7 uptime requirements
- Serializable consistency
- Adapts to workloads
- Security features

Highly available

- Survives AZ plus rack failure without human intervention
- Seamless upgrades
- Self-healing
- Smart SDKs

Data platform

- Row-oriented tables (OLTP)
- Column-oriented tables (OLAP)
- Topics (persistent queues)
- Federated queries
- Multitenancy

Typical YDB use cases

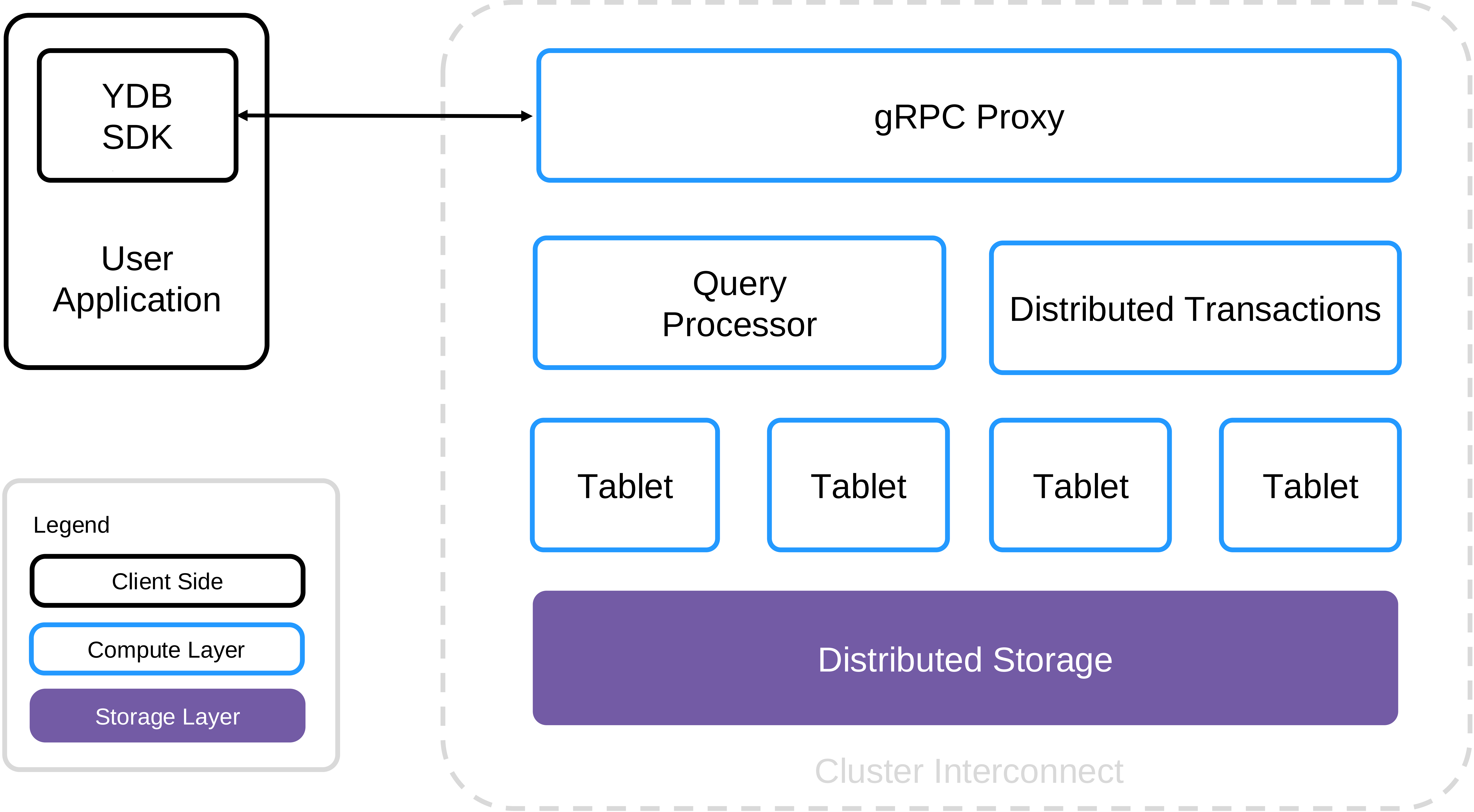
- Finance
- E-commerce
- Ride-hailing
- Advertisement
- Logistics
- AI services
- Infrastructure

Summary of YDB history

2014	Started as an in-house infrastructure technology
2020	Provided as a managed cloud service
2021	Kubernetes and Ansible deployment options
2022	Published to open-source under Apache 2.0 license
2024	PostgreSQL and Apache Kafka compatibility

YDB design and architecture

YDB high-level architecture



Distributed storage topologies

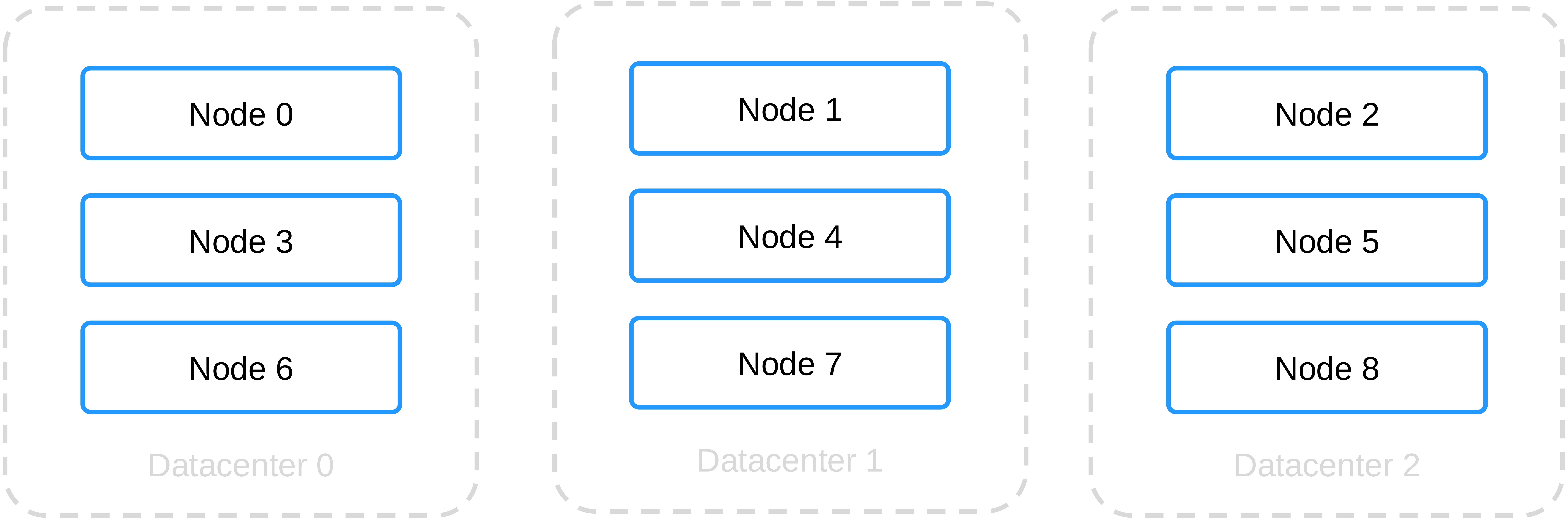
Mirror-3-DC

- Cross-datacenter
- Synchronous replication
- Survives failure of DC + rack
- ×3 space amplification

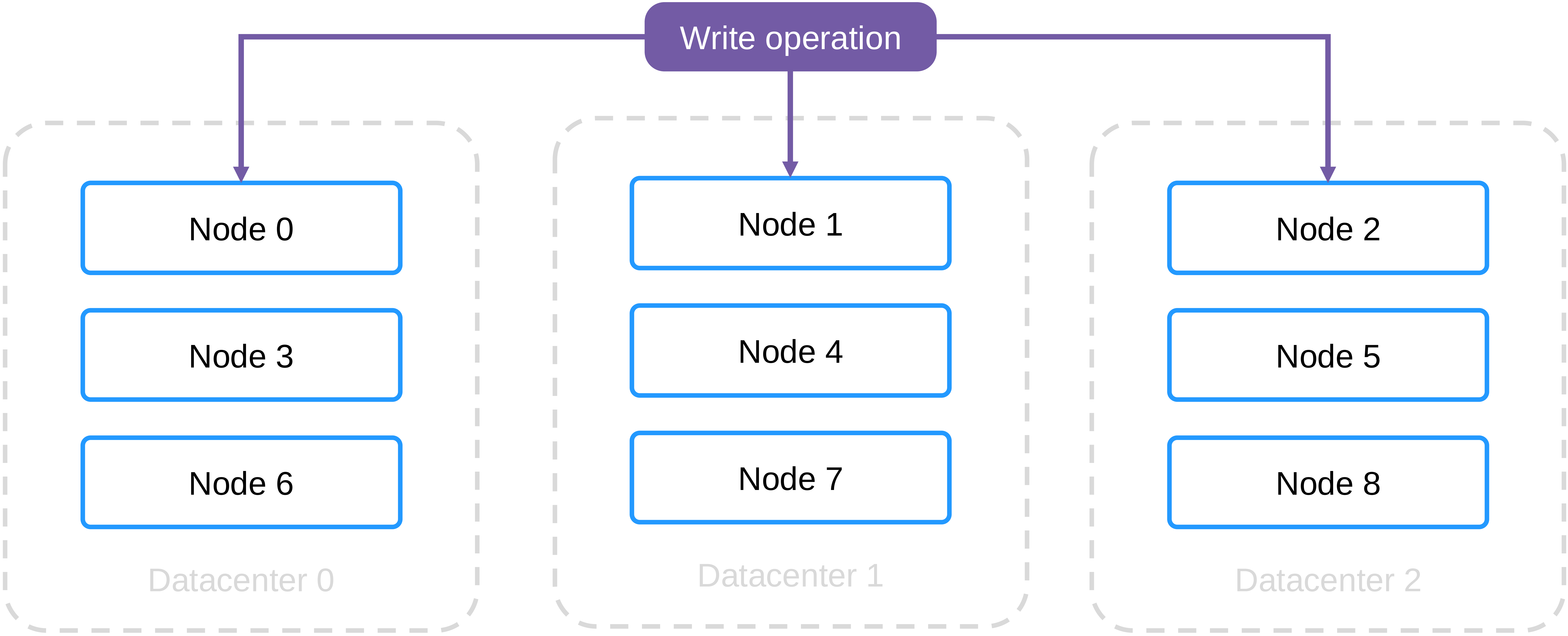
Block-4-2

- Single-datacenter
- Erasure coding
- Survives failure of 2 racks
- ×1.5 space amplification

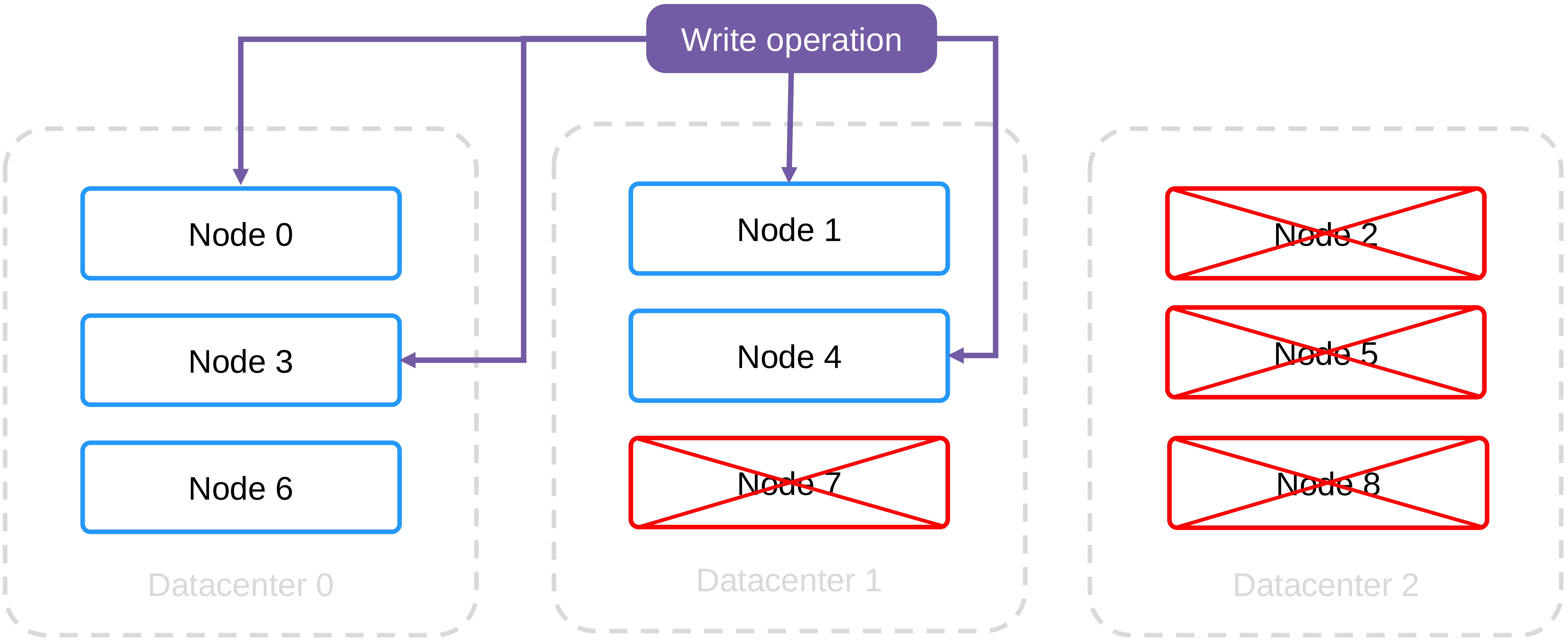
Mirror-3-DC topology



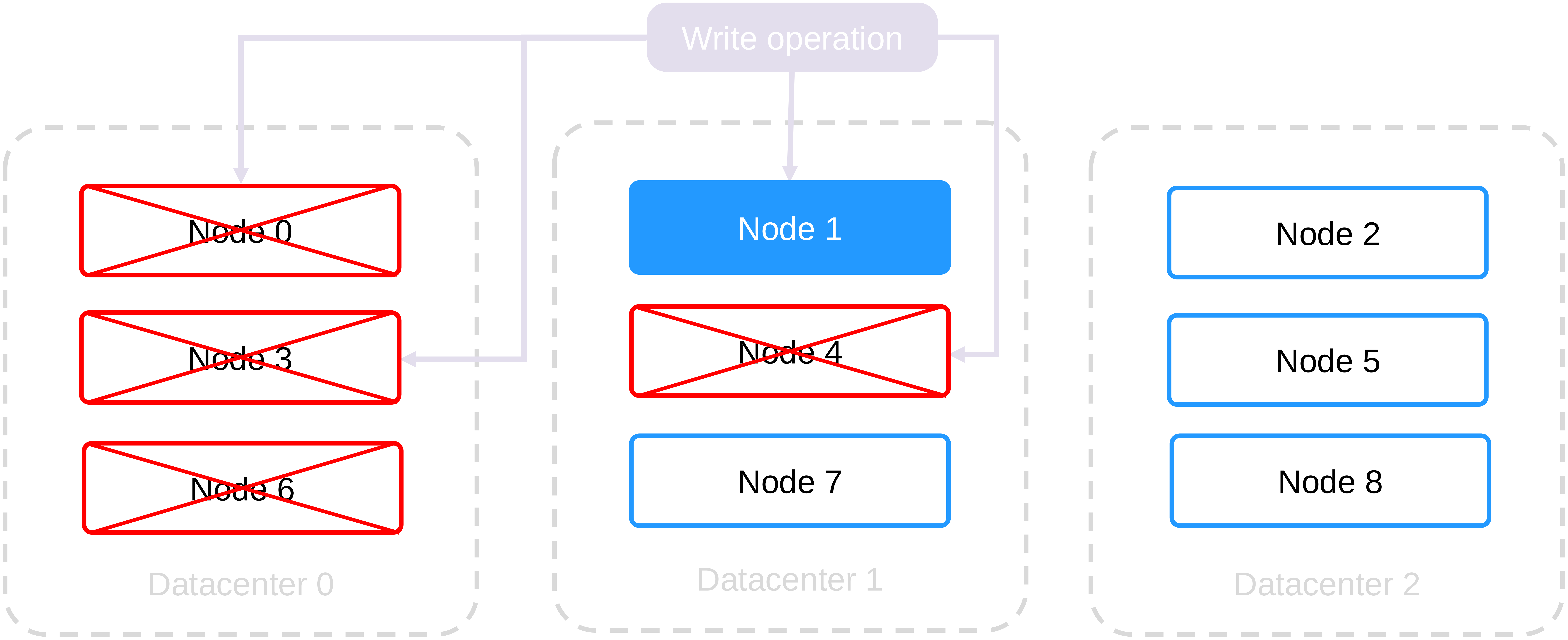
Mirror-3-DC topology: normal case



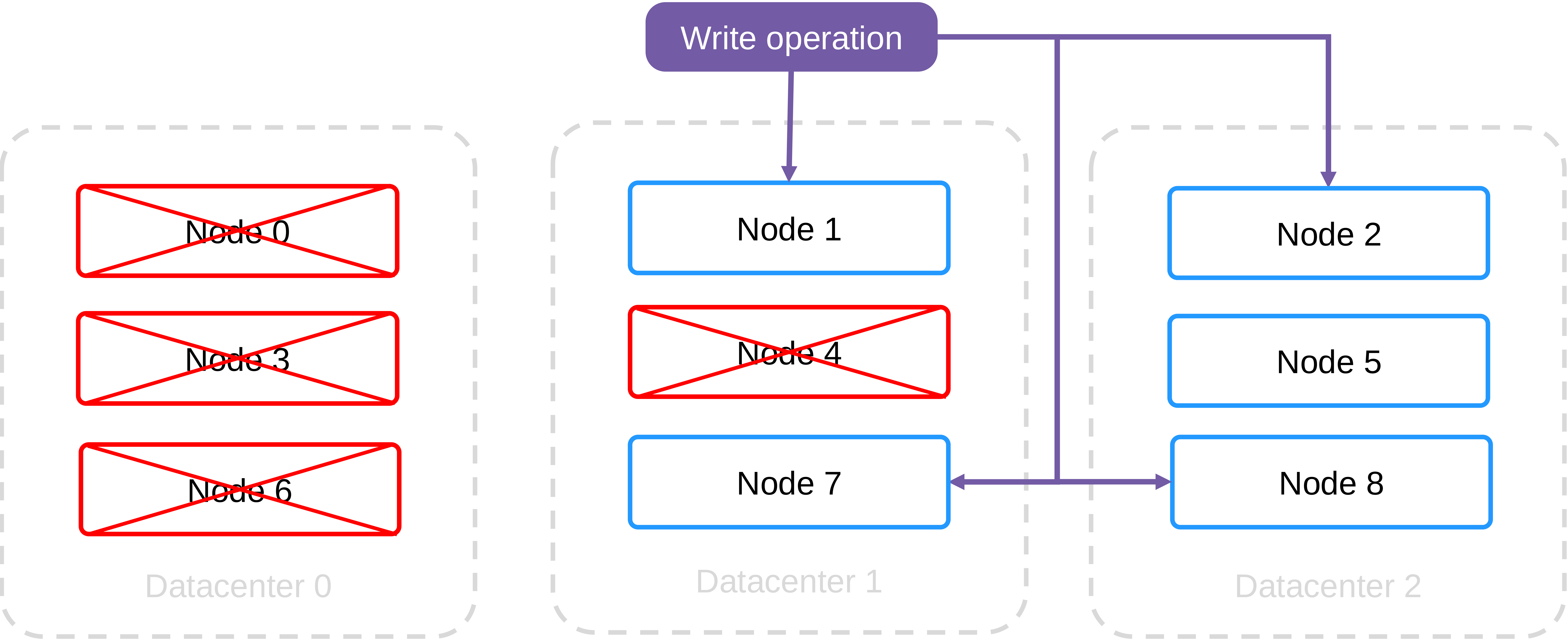
Mirror-3-DC topology: worst case



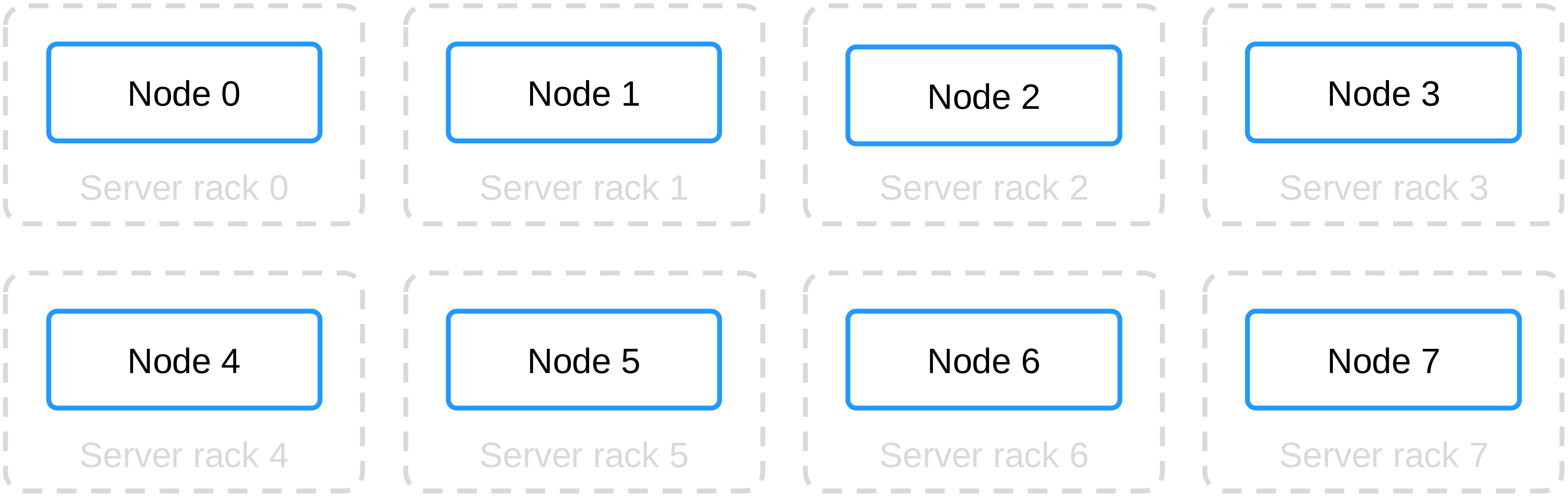
Mirror-3-DC topology: worst case



Mirror-3-DC topology: worst case



Block-4-2 topology



Node 0

Server rack 0

Node 1

Server rack 1

Node 2

Server rack 2

Node 3

Server rack 3

Node 4

Server rack 4

Node 5

Server rack 5

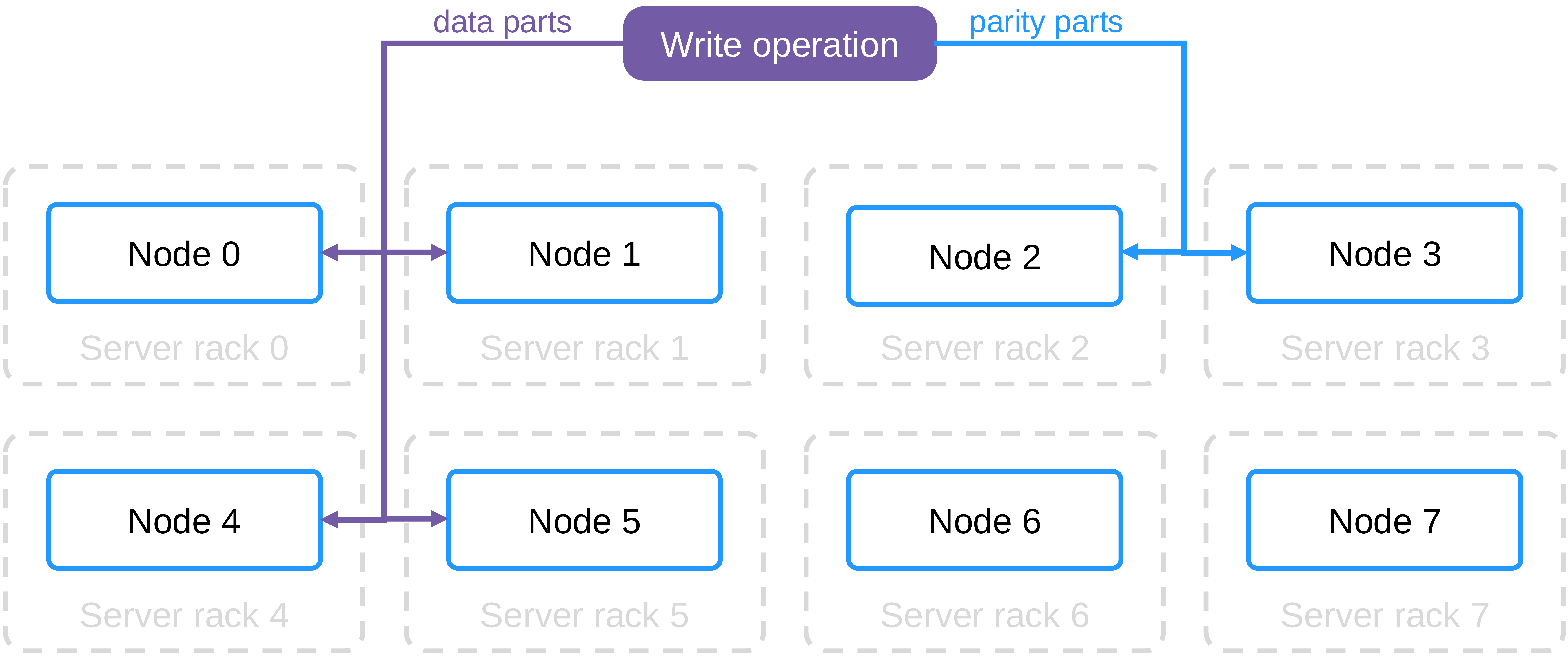
Node 6

Server rack 6

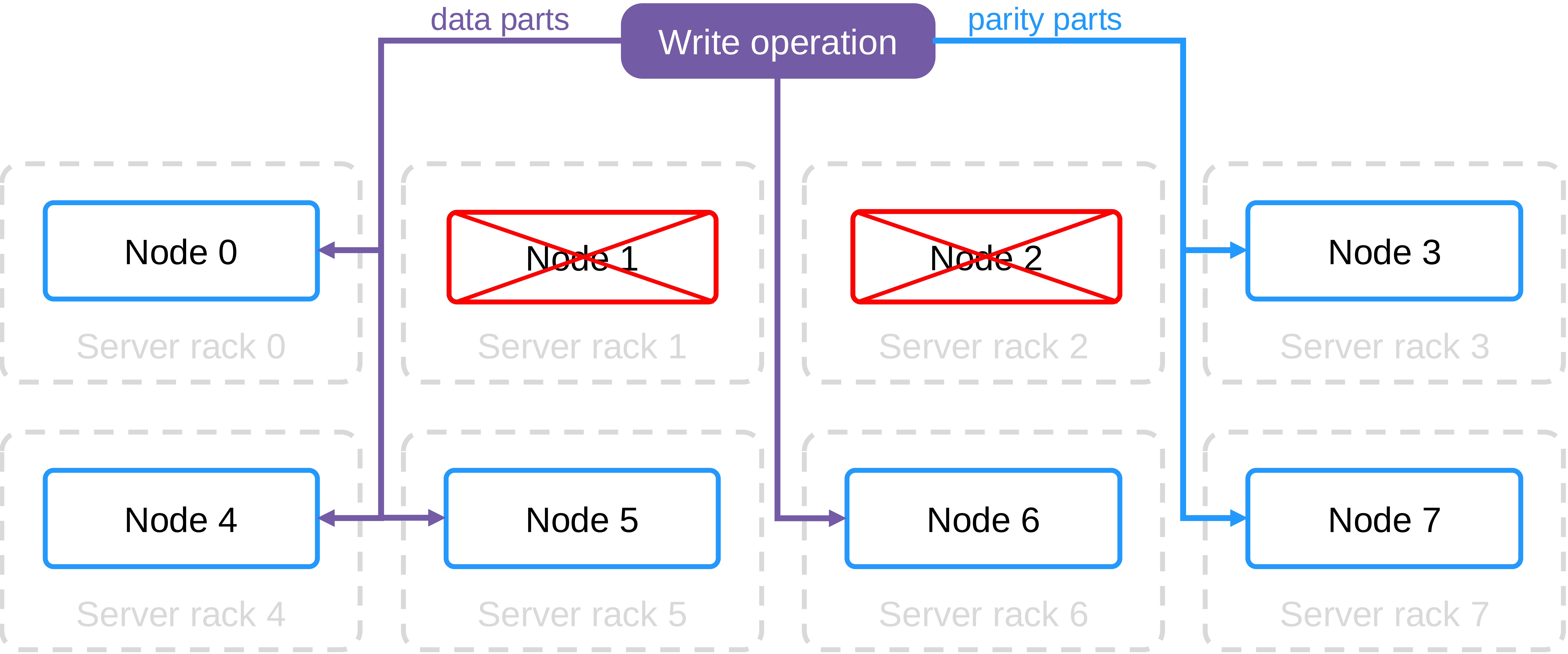
Node 7

Server rack 7

Block-4-2 topology: normal case



Block-4-2 topology: worst case



YDB distributed transactions

- **Serializable** isolation level by default
- Initially inspired by **deterministic transactions** from a Yale University research paper *
- Optimistic locking for conflict detection
- Multi-version concurrency control (MVCC)

* *A. Thomson, T. Diamond, S.-C. Weng, K. Ren, P. Shao, and D. J. Abadi. Calvin: Fast Distributed Transactions for Partitioned Database Systems. In Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, SIGMOD '12, pages 1–12, New York, NY, USA, 2012. ACM.*

YDB in open-source

Leveraging open-source ecosystems

API compatibility

- gRPC
- PostgreSQL
- Apache Kafka

Deployment options

- Kubernetes
- Ansible

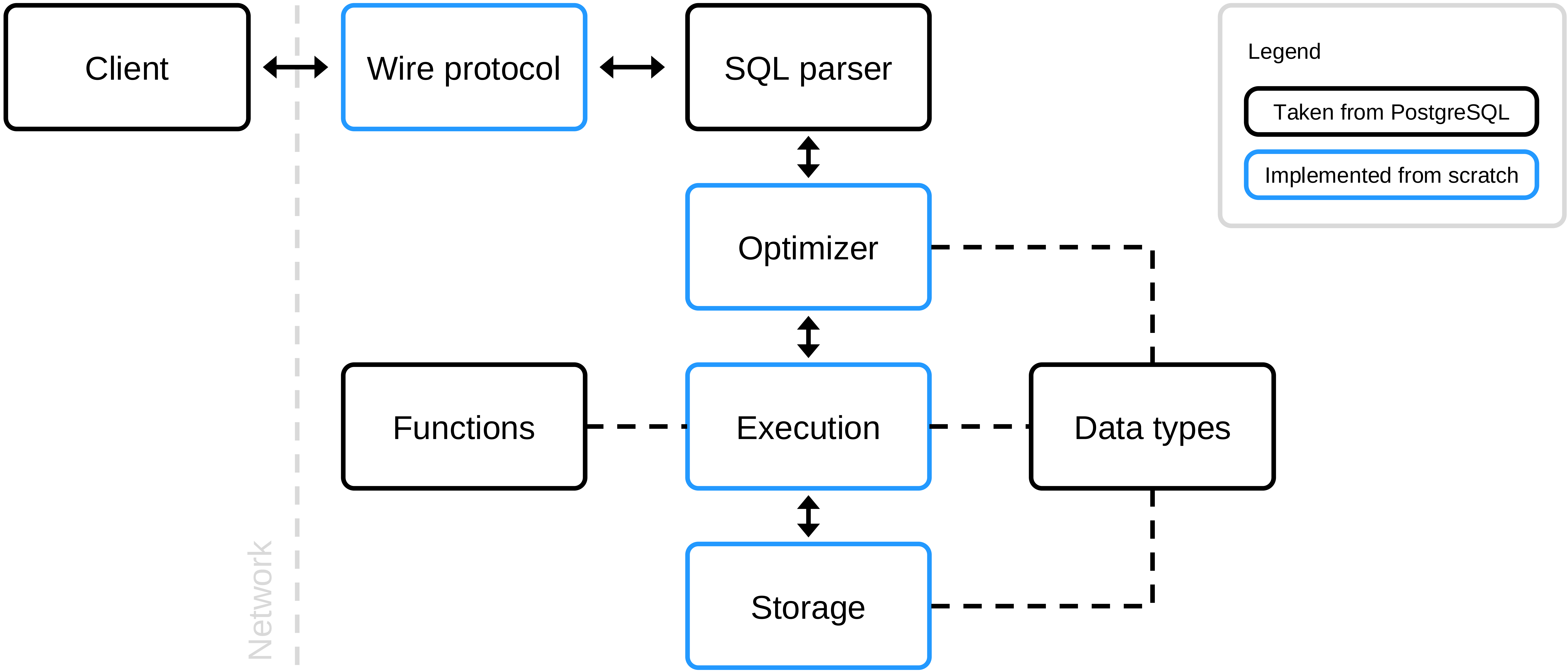
Data management

- Hibernate
- Apache Airflow
- DBeaver

Observability

- Grafana
- Prometheus
- Jaeger

YDB’s PostgreSQL compatibility mode



YDB is 100% open-source

Permissive Apache 2.0 License for:

- Core platform is built from scratch in C++
- Kubernetes operator in Go
- Ansible playbooks in Python 3
- SDKs in Java, Python, Go, Rust, Node.js, PHP, etc.
- Documentation in Markdown

https://github.com/ydb-platform/ydb

Contributors are welcome!

YDB

Thank you!

https://ydb.tech

YDB highlights:

- Strong consistency
- Resilience and self-healing
- Elastic scalability
- Various workloads
- PostgreSQL and Kafka compatibility
- 100% open-source under Apache 2.0