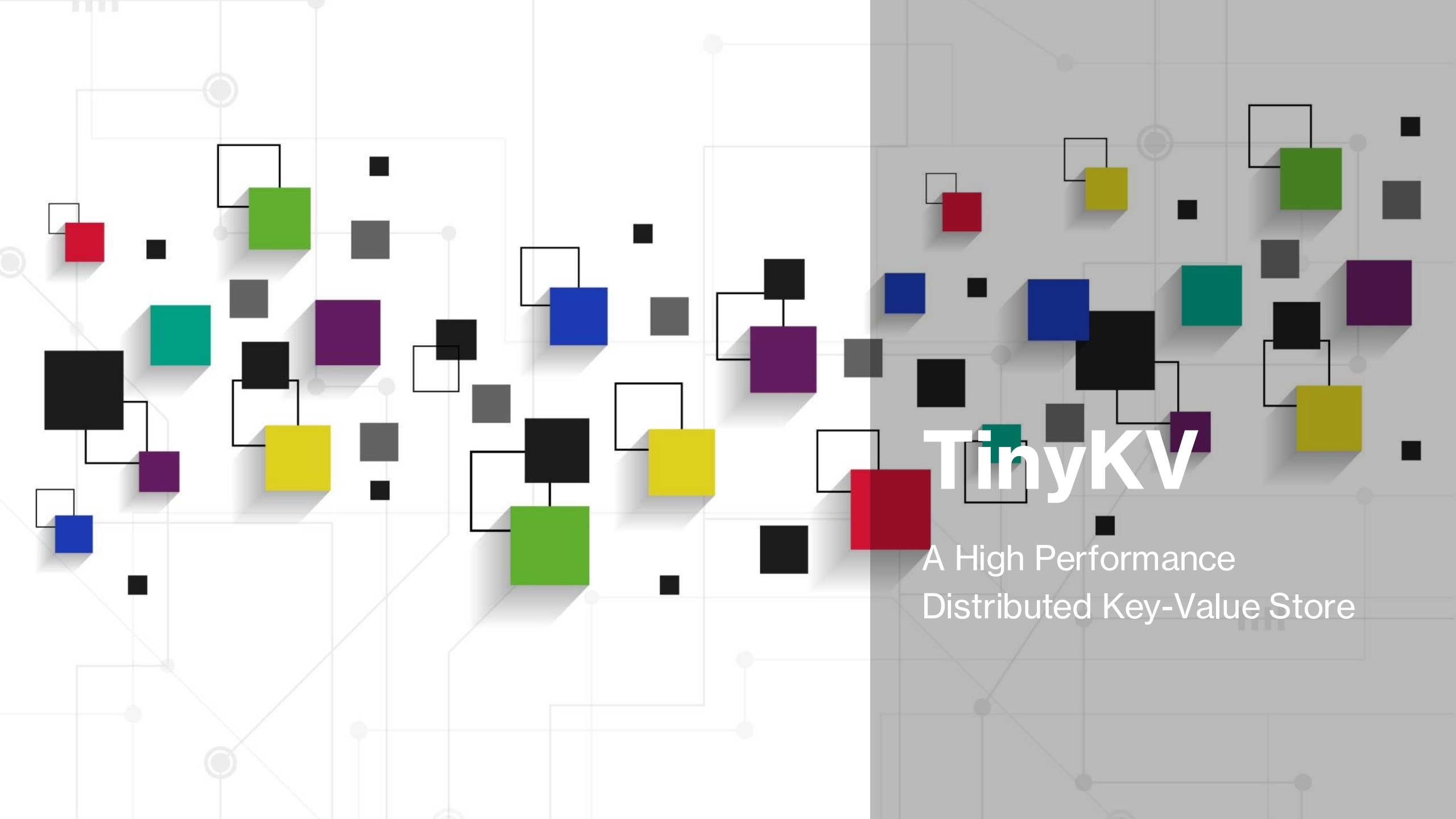
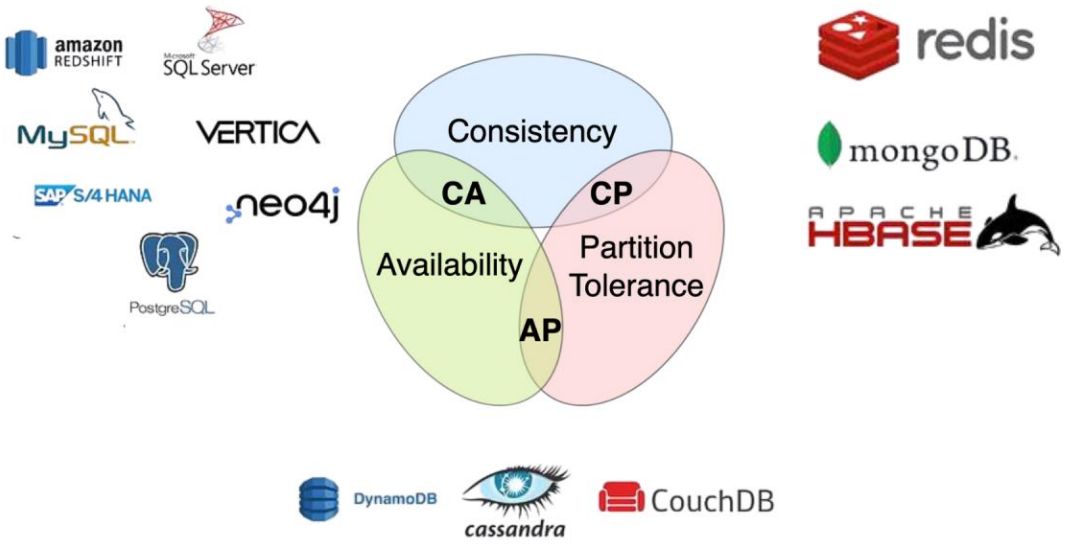

TinyKV
A High Performance
Distributed Key-Value Store

Concept

- Decentralized peer-to-peer architecture
- Availability and network partitioning
- Performance
- Tunable consistency

Motivation and Learning goals

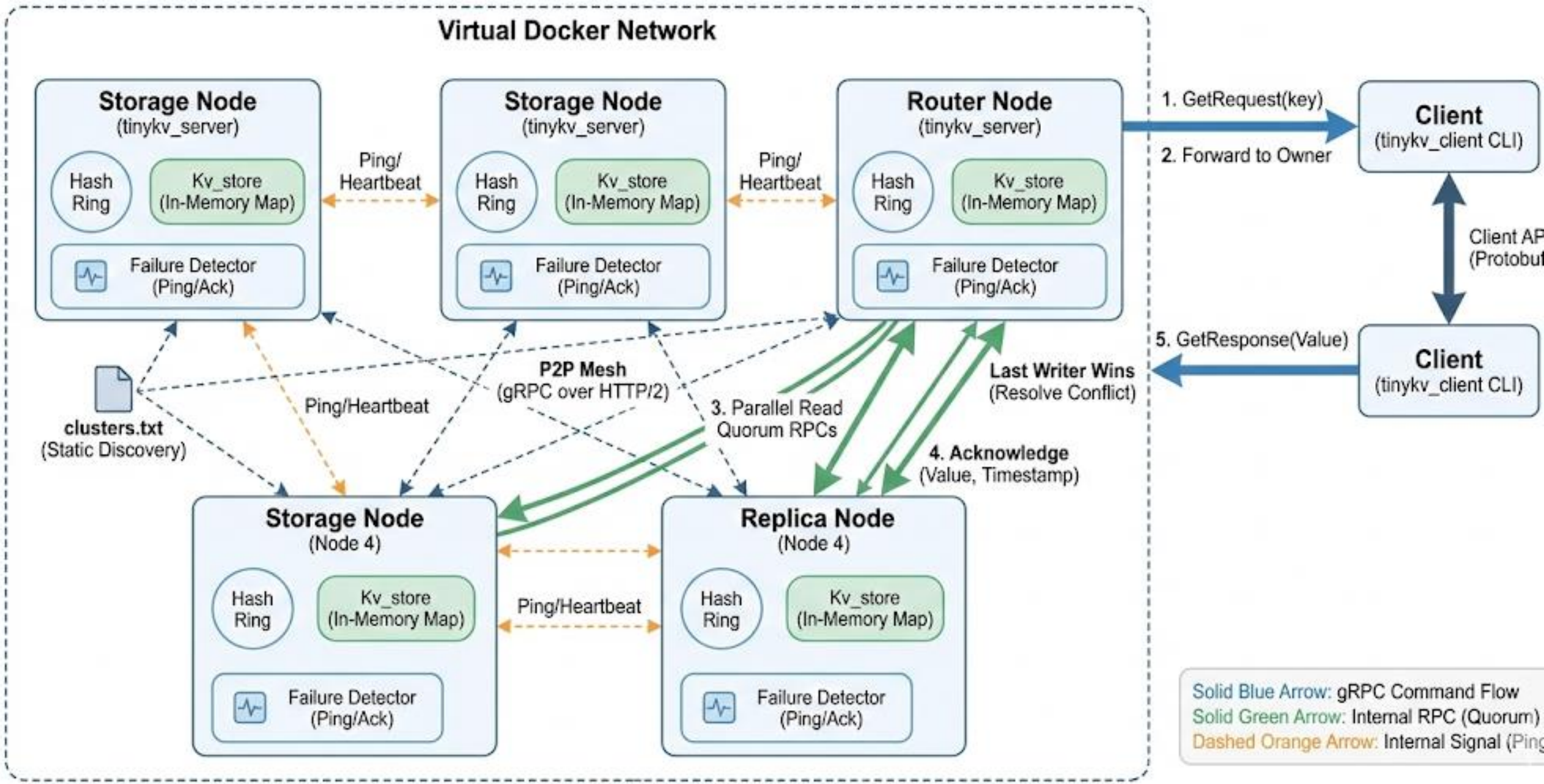
- CAP theorem in practice
- Performance and latency in distributed systems
- Practical experience with replication, partitioning, quorums and failure detection



Tools and technologies

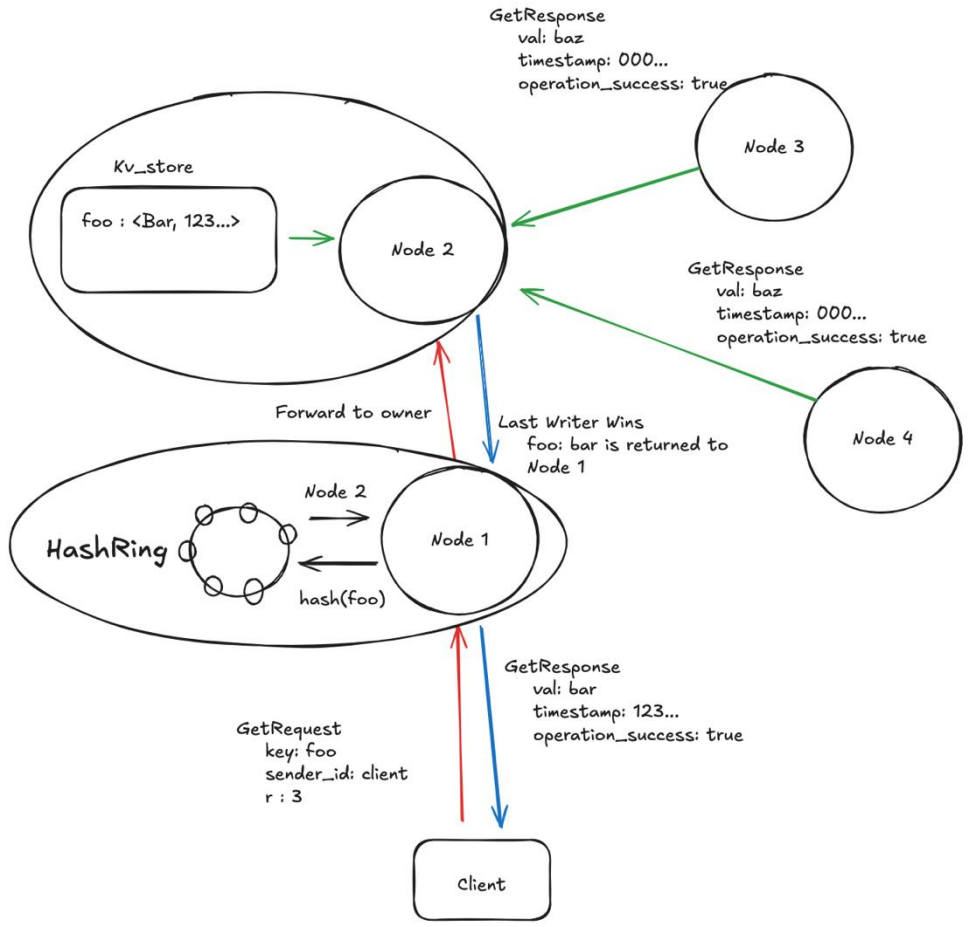
- C++20
  - High performance and fine grained control
- gRPC
  - Efficient communication and memory compact payloads
- Docker
  - Portability and containerization

TinyKV System Architecture Diagram

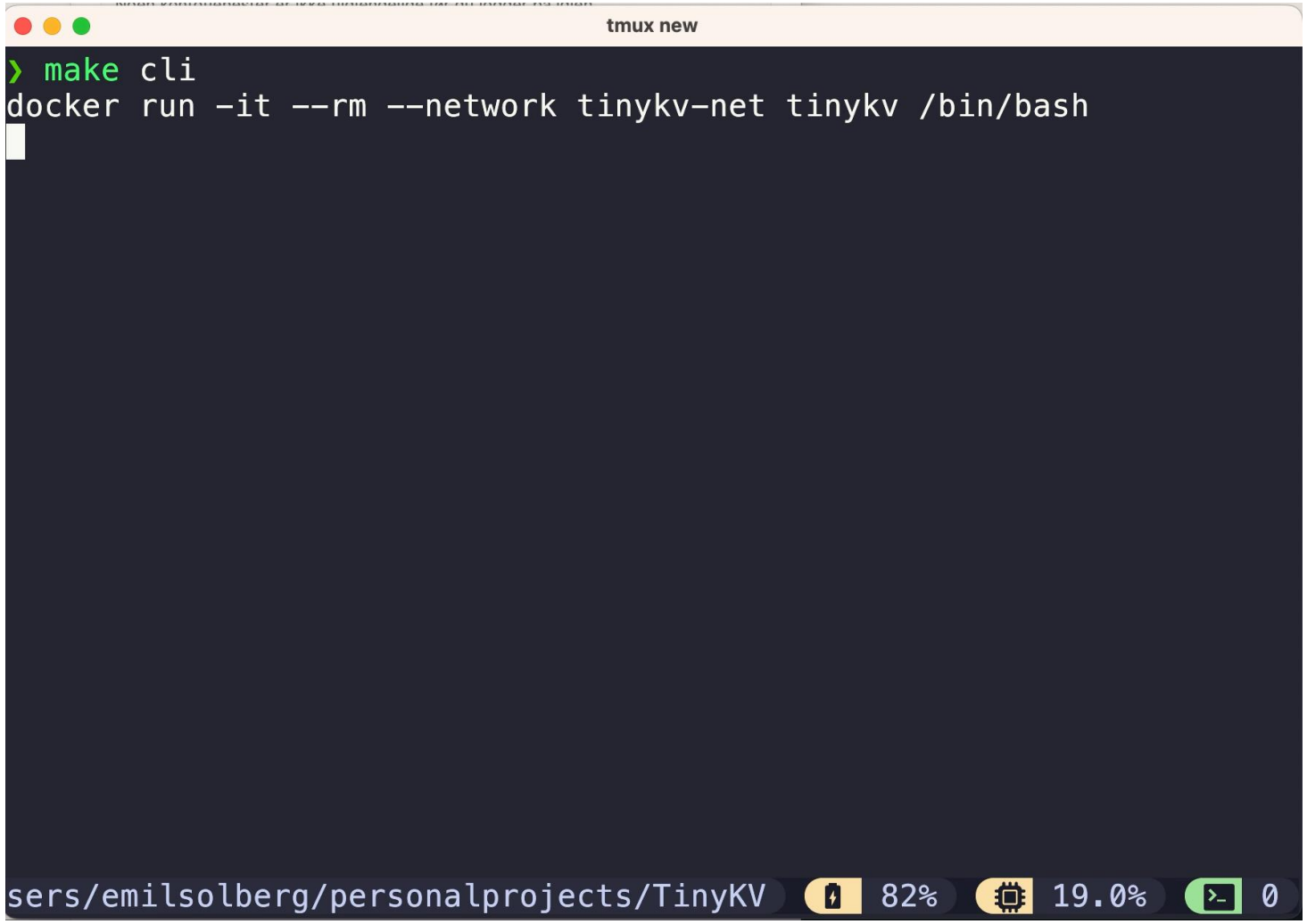


How it works

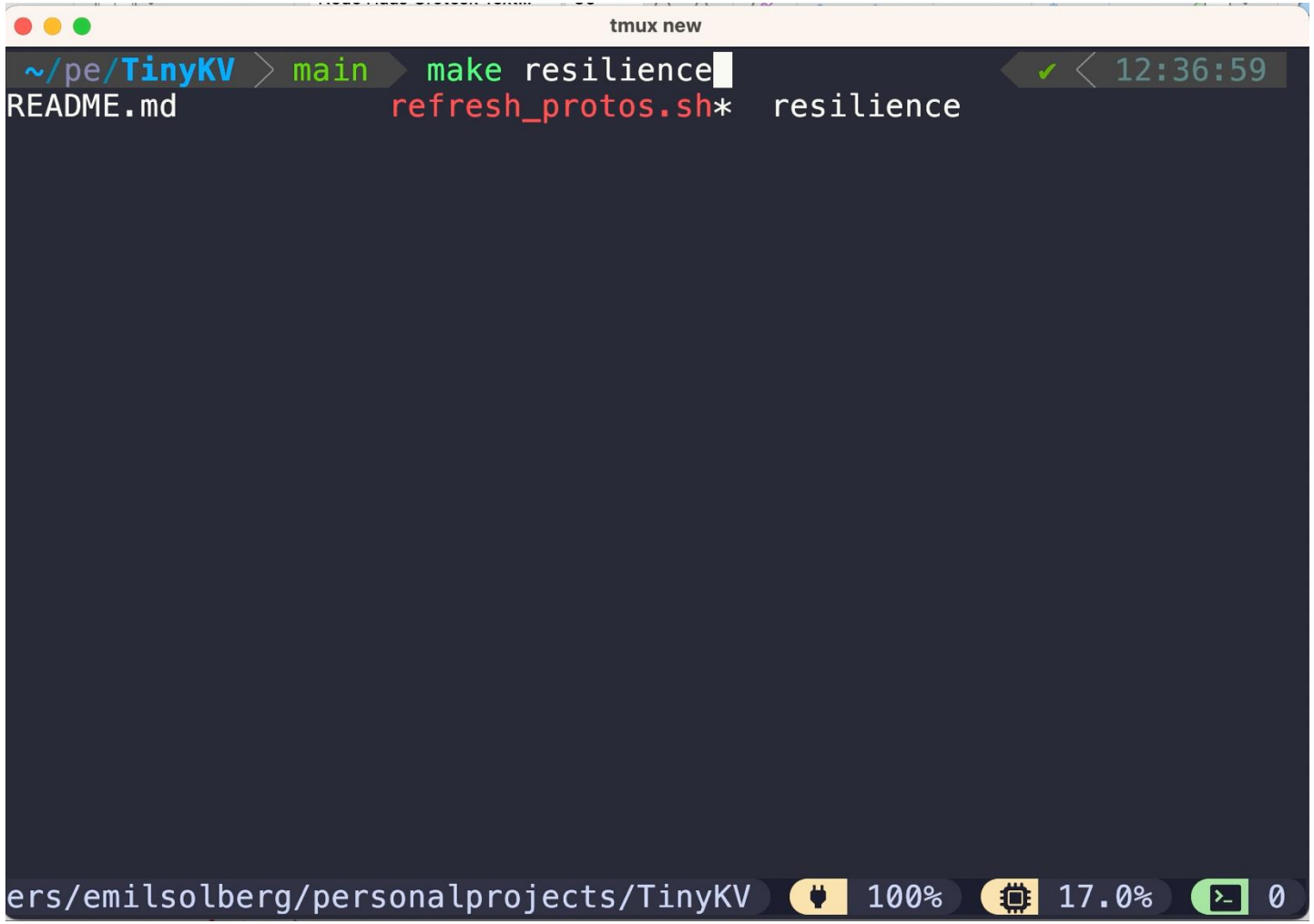
- Hash ring
- In memory hashmap with timestamped values
- Read and write quorums



Demonstration

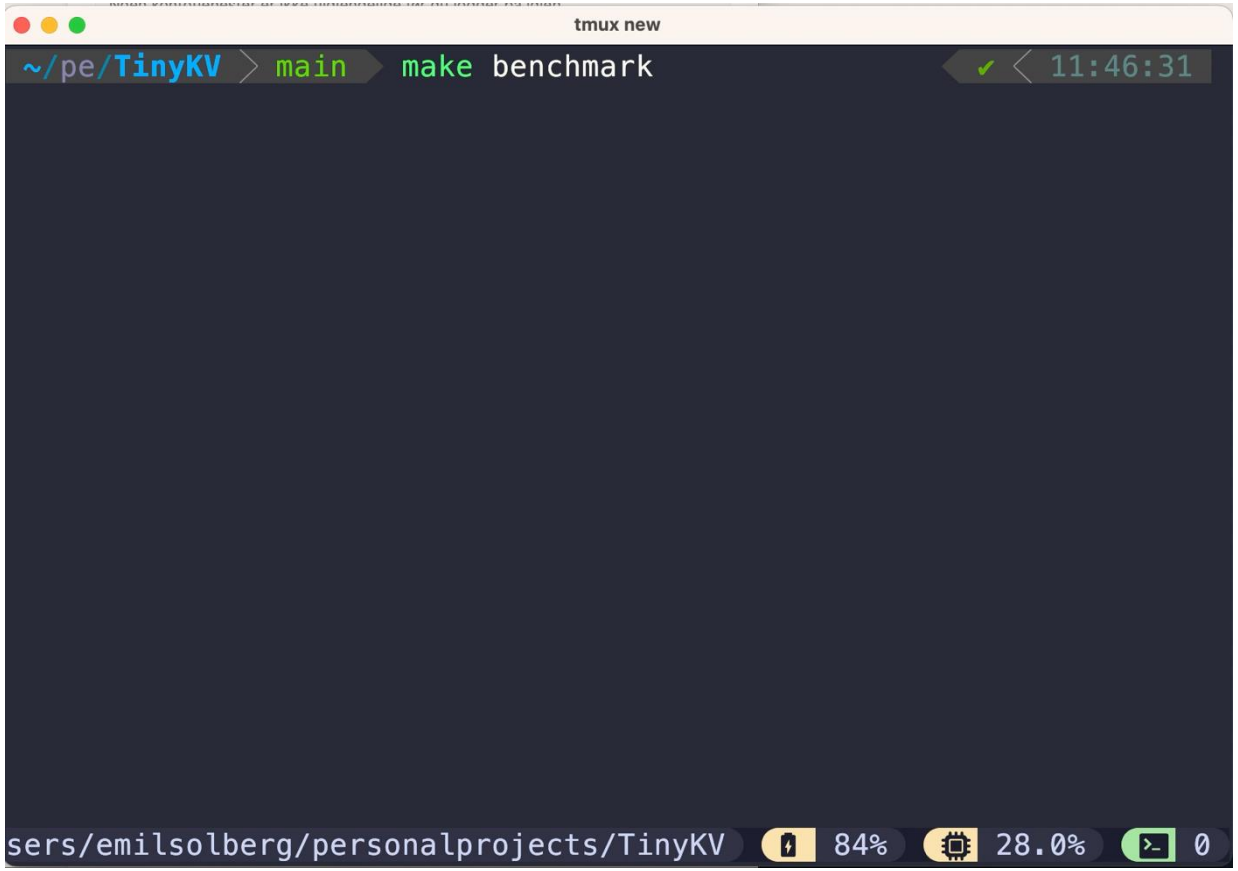


Tunable consistency



Performance

- 10´000 requests, 50 clients
- Up to 5000 OPS
- Achieved through parallelism
- Limited by terminal output, docker overhead and router node bottleneck



Improvements and future works

- Gossip protocol
- Asynchronous network requests
- Comprehensive test suite
- DELETE command
- Security and authorization
- Dynamic membership
- Load balancing