

The Apache Cassandra storage engine

Sylvain Lebresne
([sylvain@!\[\]\(c8d96c8885d3000a912c2582004aed63_img.jpg\)DataStax.com](mailto:sylvain@DataStax.com))

FOSDEM '12, Brussels



1. What is Apache Cassandra

2. Data Model

3. The storage engine

1. What is Apache Cassandra

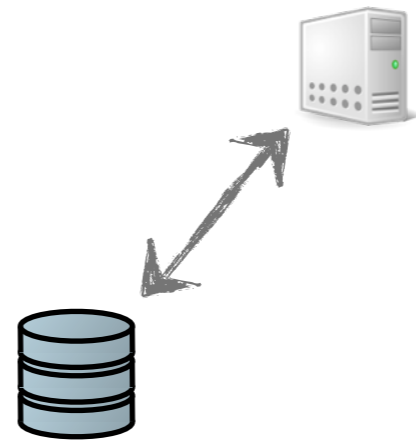
2. Data Model

3. The storage engine

about:project

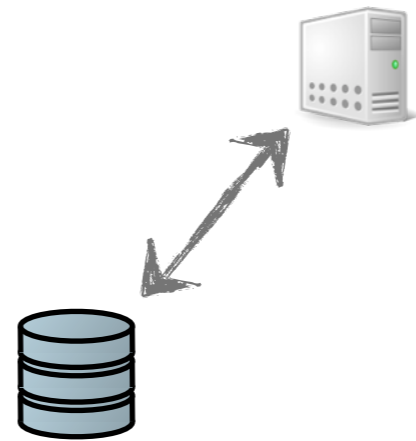
- Distributed data store aimed at big data.
- Apache project since 2010.
- Version 1.0 released last October.
- Proven in production (Netflix, Twitter, Reddit, Cisco, ...). Largest know cluster has over 300TB in over 400 machines.

Apache Cassandra



Apache Cassandra

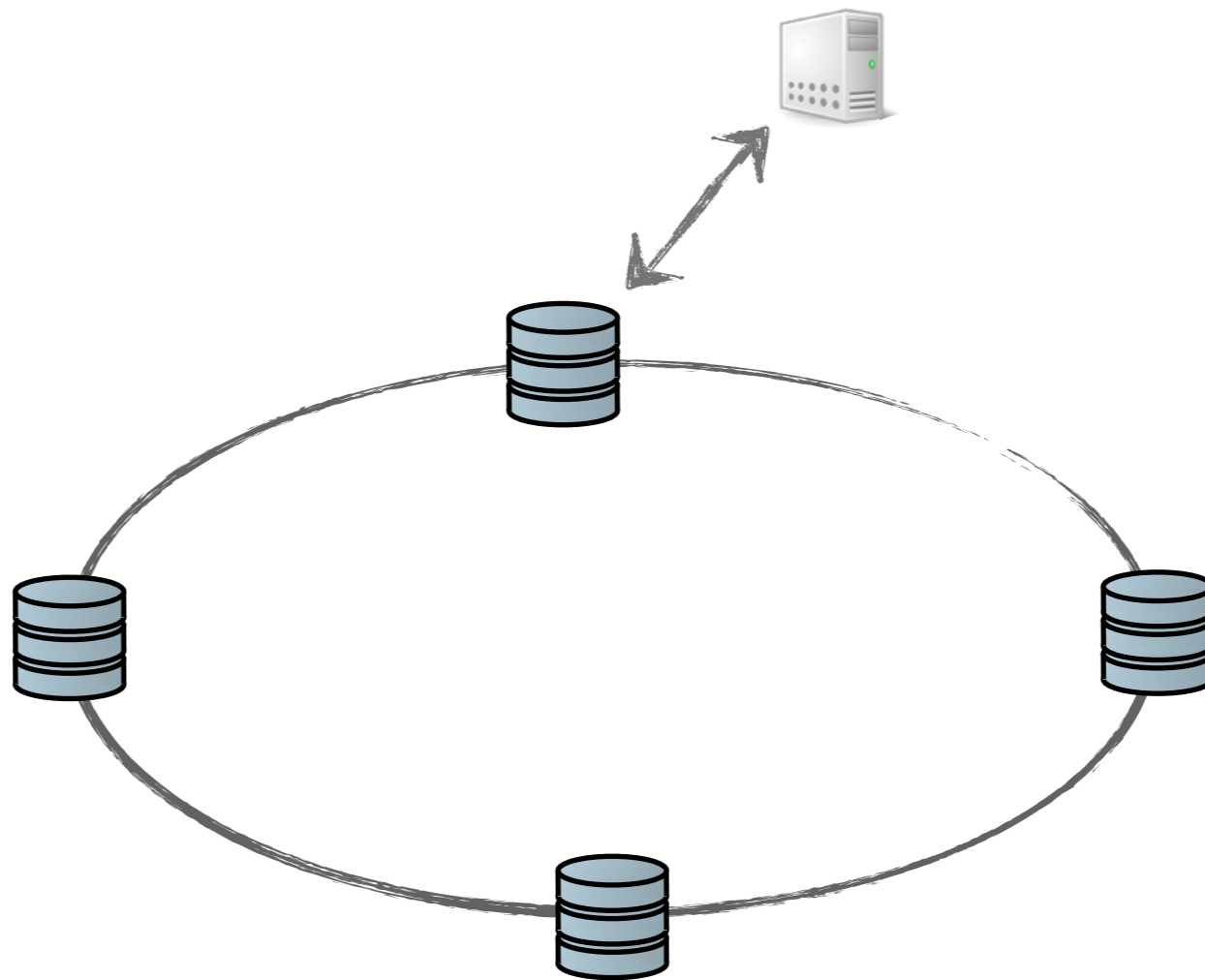
A database:



Apache Cassandra

A database:

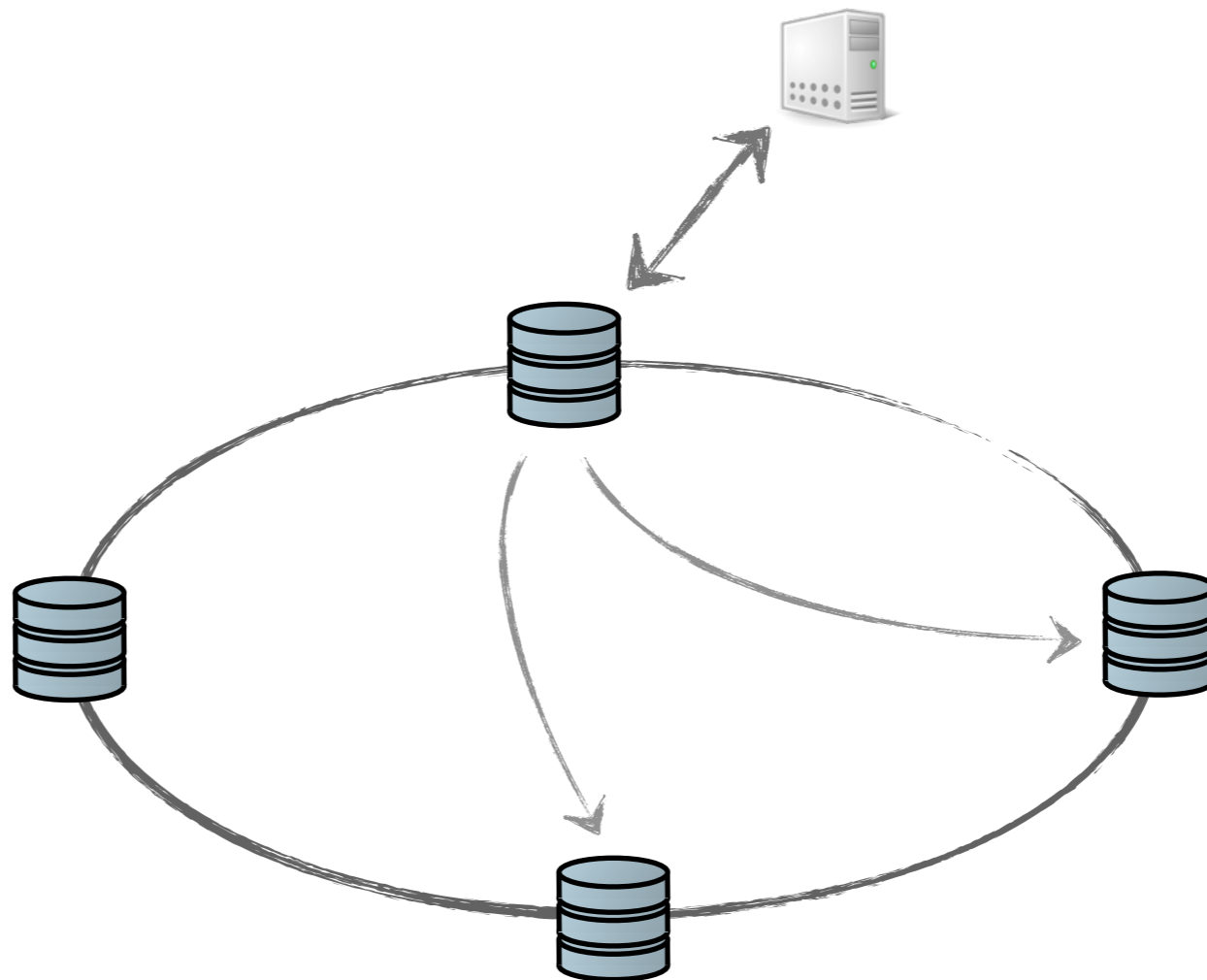
- distributed / decentralized



Apache Cassandra

A database:

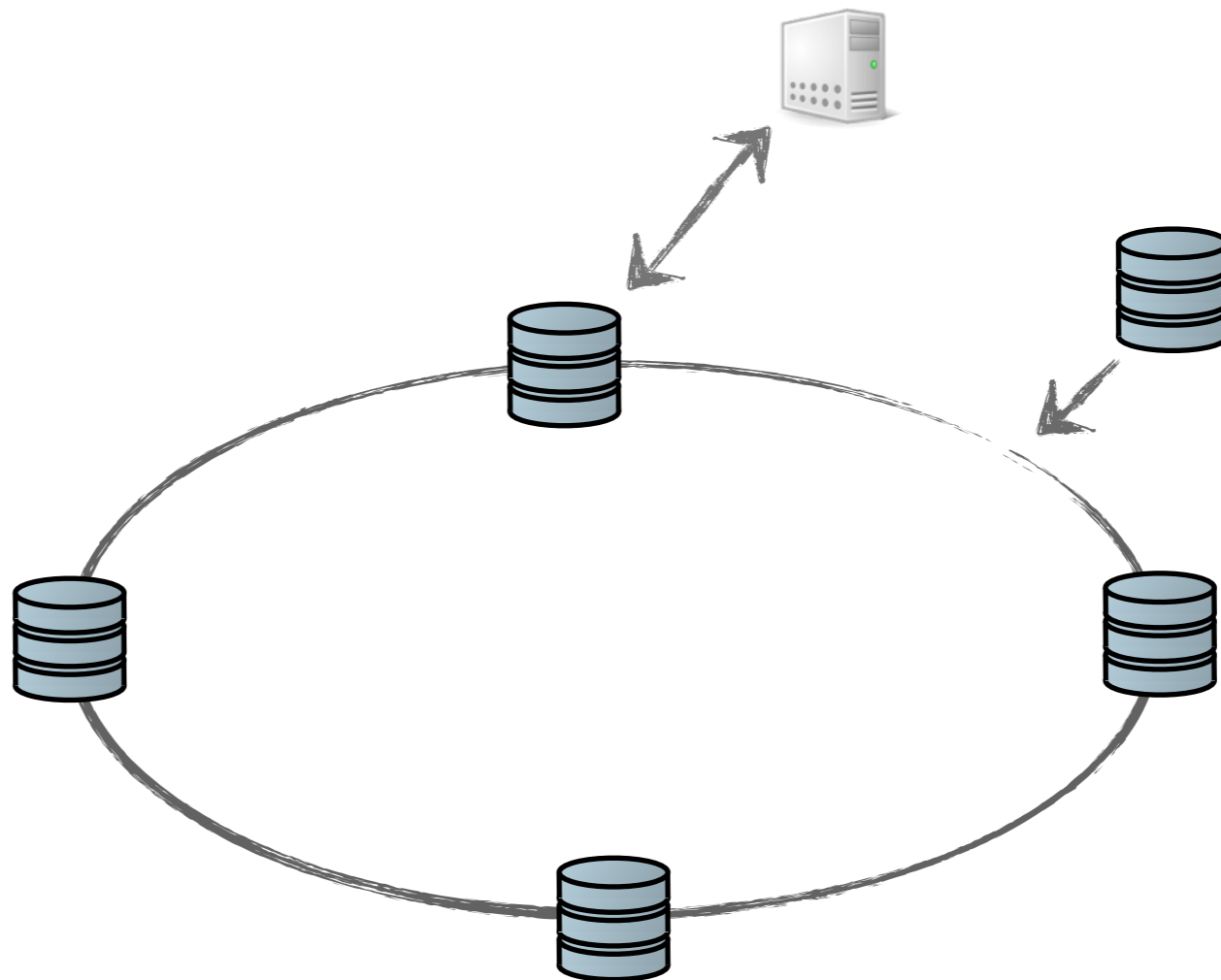
- distributed / decentralized
- replicated & durable



Apache Cassandra

A database:

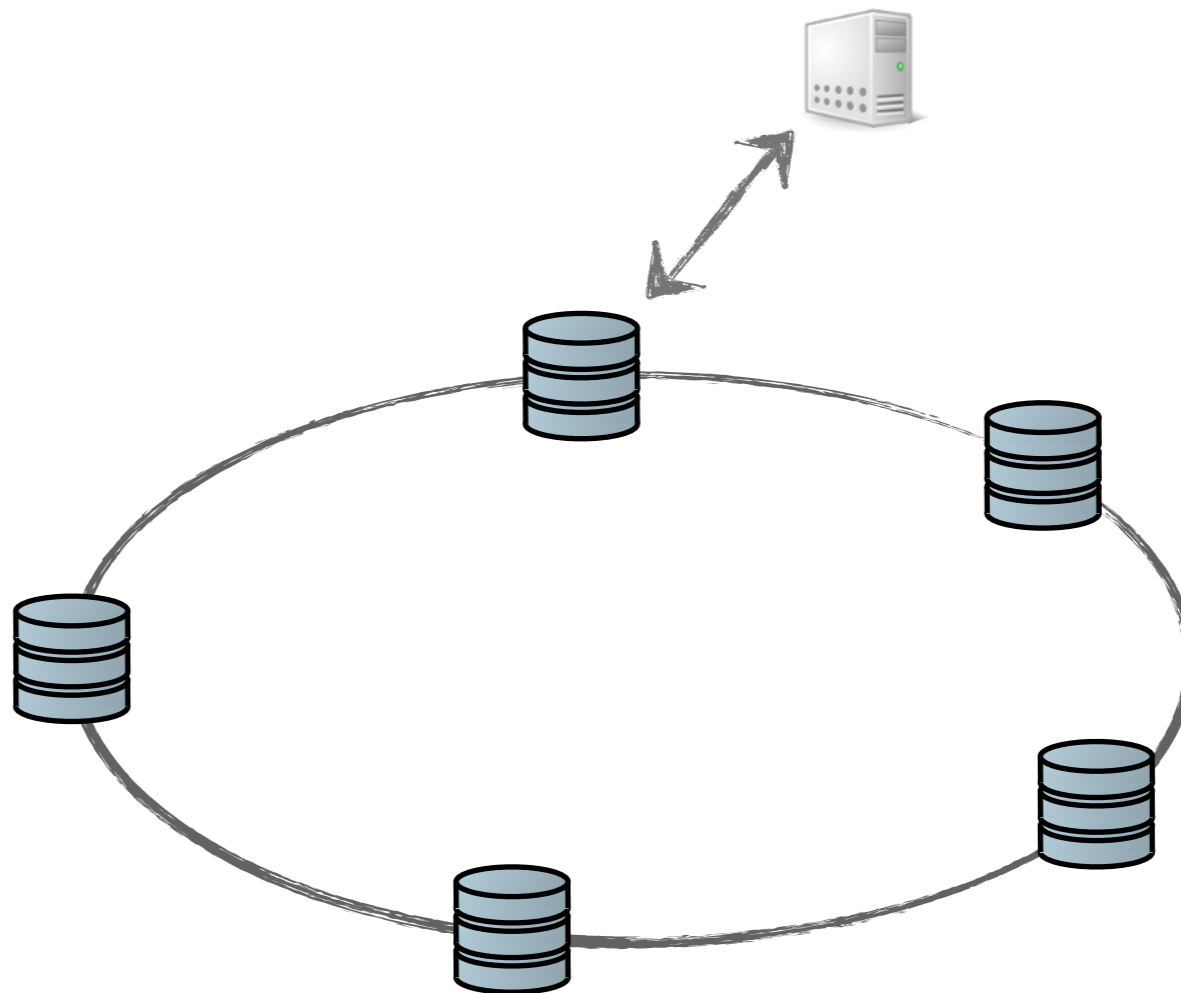
- distributed / decentralized
- replicated & durable
- scalable / elastic



Apache Cassandra

A database:

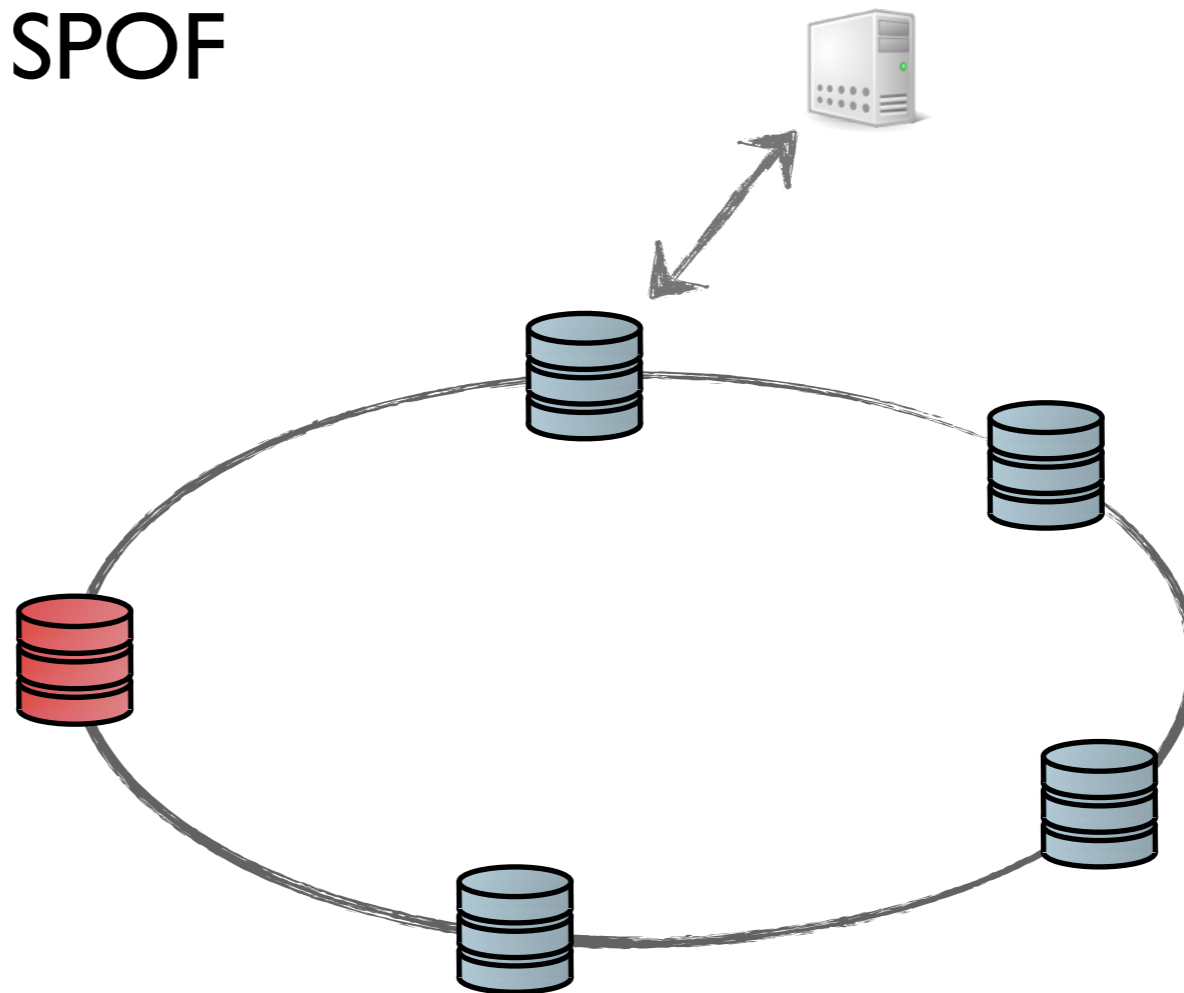
- distributed / decentralized
- replicated & durable
- scalable / elastic



Apache Cassandra

A database:

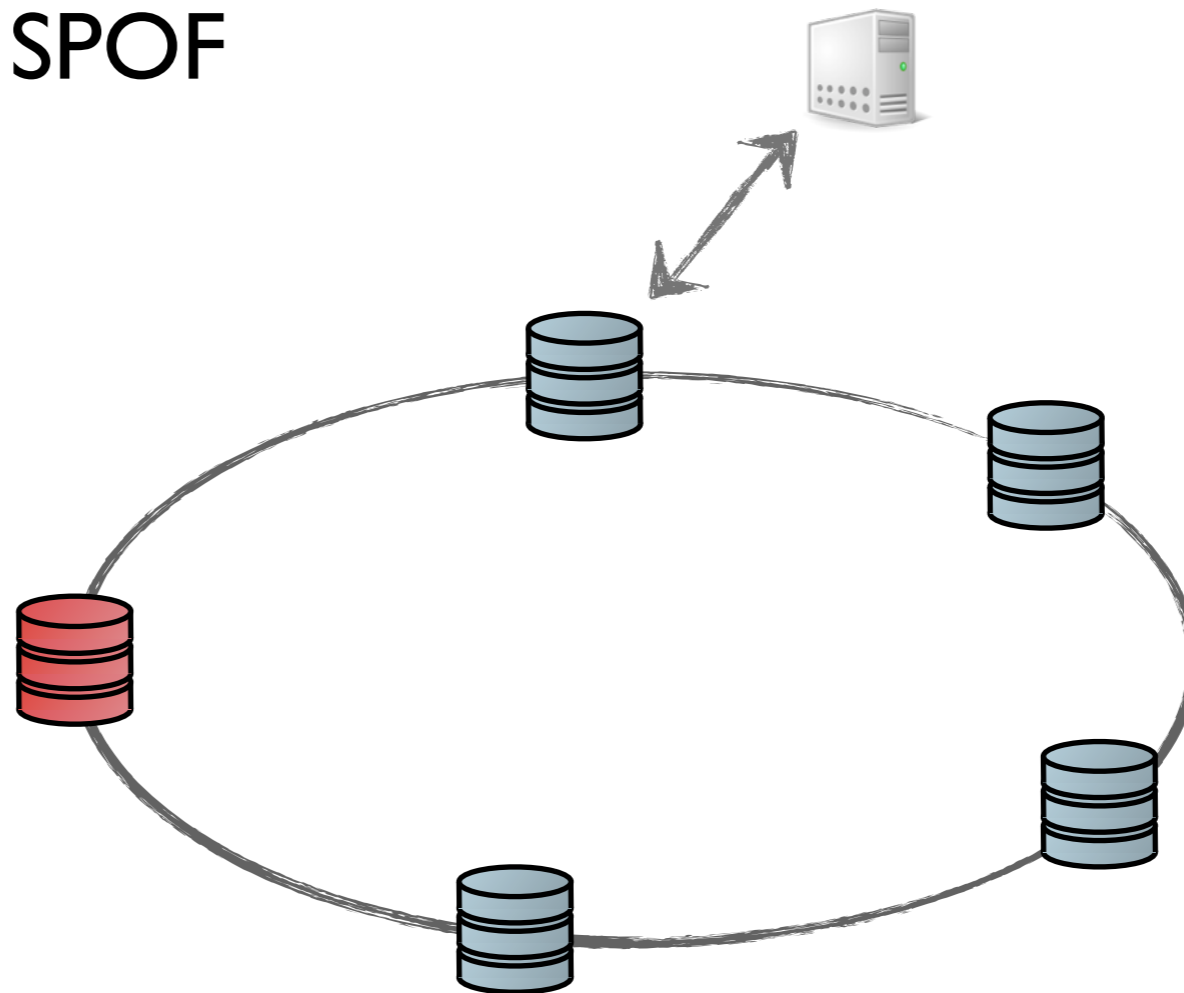
- distributed / decentralized
- replicated & durable
- scalable / elastic
- fault-tolerant / no SPOF



Apache Cassandra

A database:

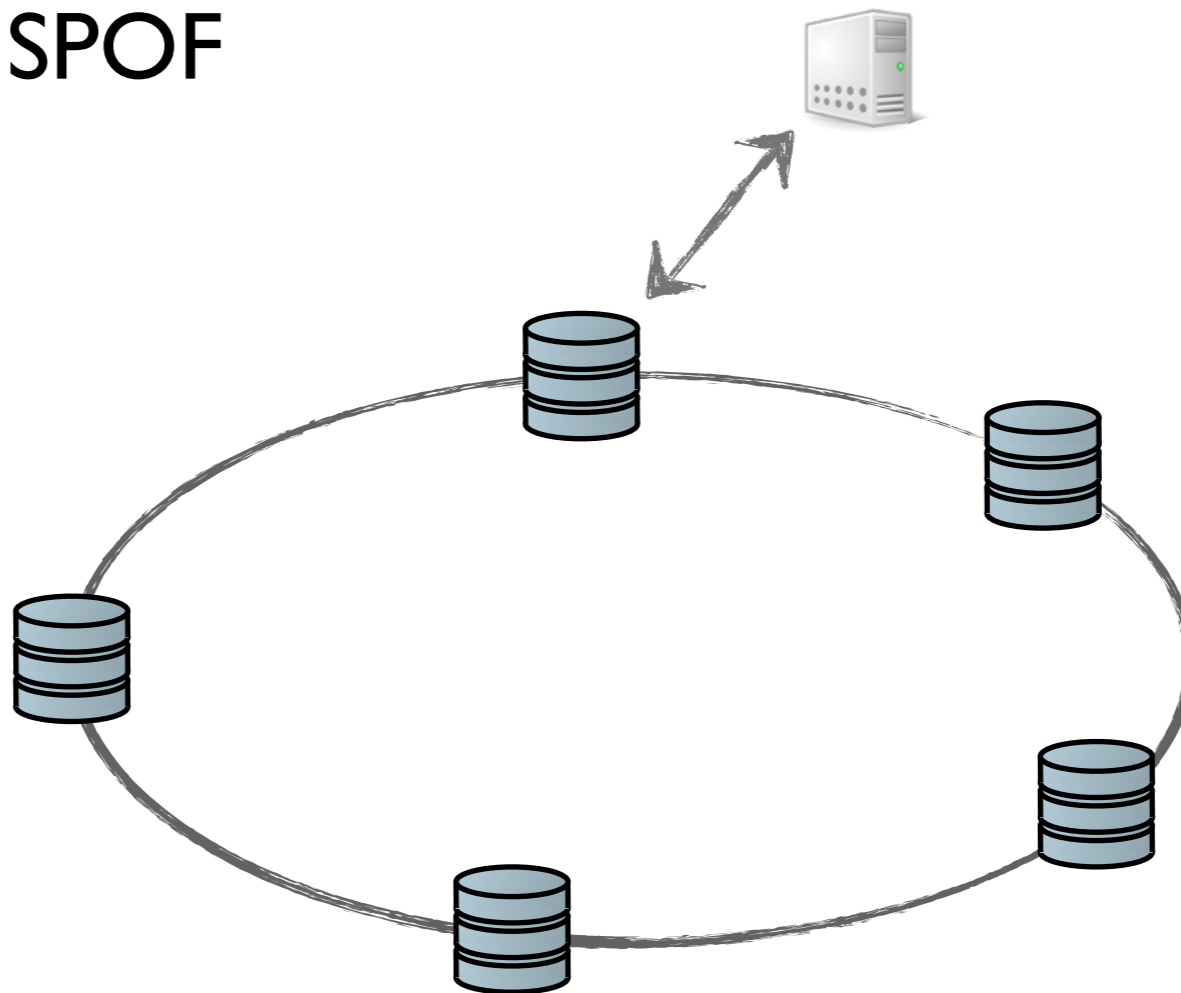
- distributed / decentralized
- replicated & durable
- scalable / elastic
- fault-tolerant / no SPOF
- highly available



Apache Cassandra

A database:

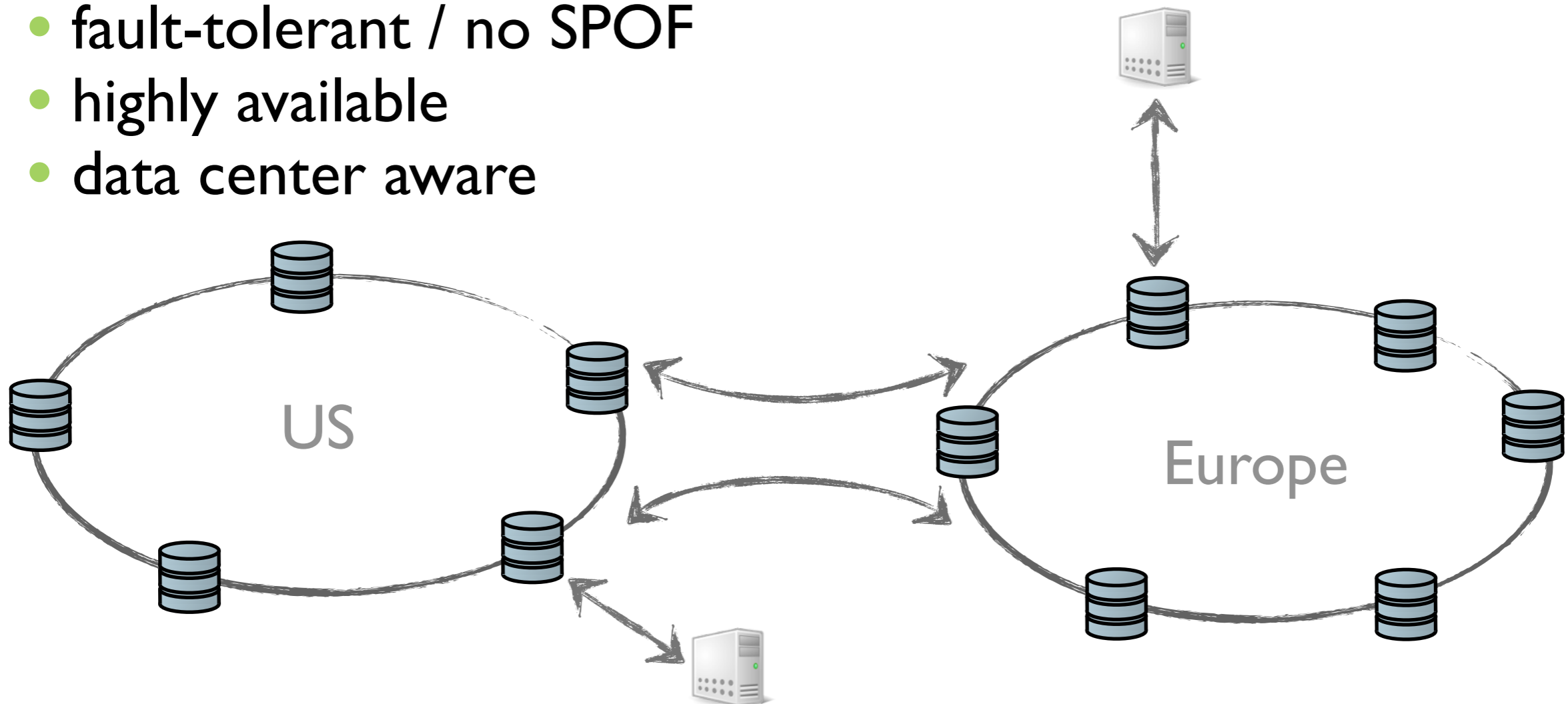
- distributed / decentralized
- replicated & durable
- scalable / elastic
- fault-tolerant / no SPOF
- highly available



Apache Cassandra

A database:

- distributed / decentralized
- replicated & durable
- scalable / elastic
- fault-tolerant / no SPOF
- highly available
- data center aware



1. What is Apache Cassandra

2. Data Model

3. The storage engine

Data Model

- Not SQL (no transaction, nor joins) but more than Key/Value.
- Inspired by Google BigTable
- Column families based.

Ex: user profiles

“For each user, holds profile infos”

50e8-e29b	
birth_year	1994
fname	Justin
lname	Bieber

Users

Ex: user profiles

“For each user, holds profile infos”

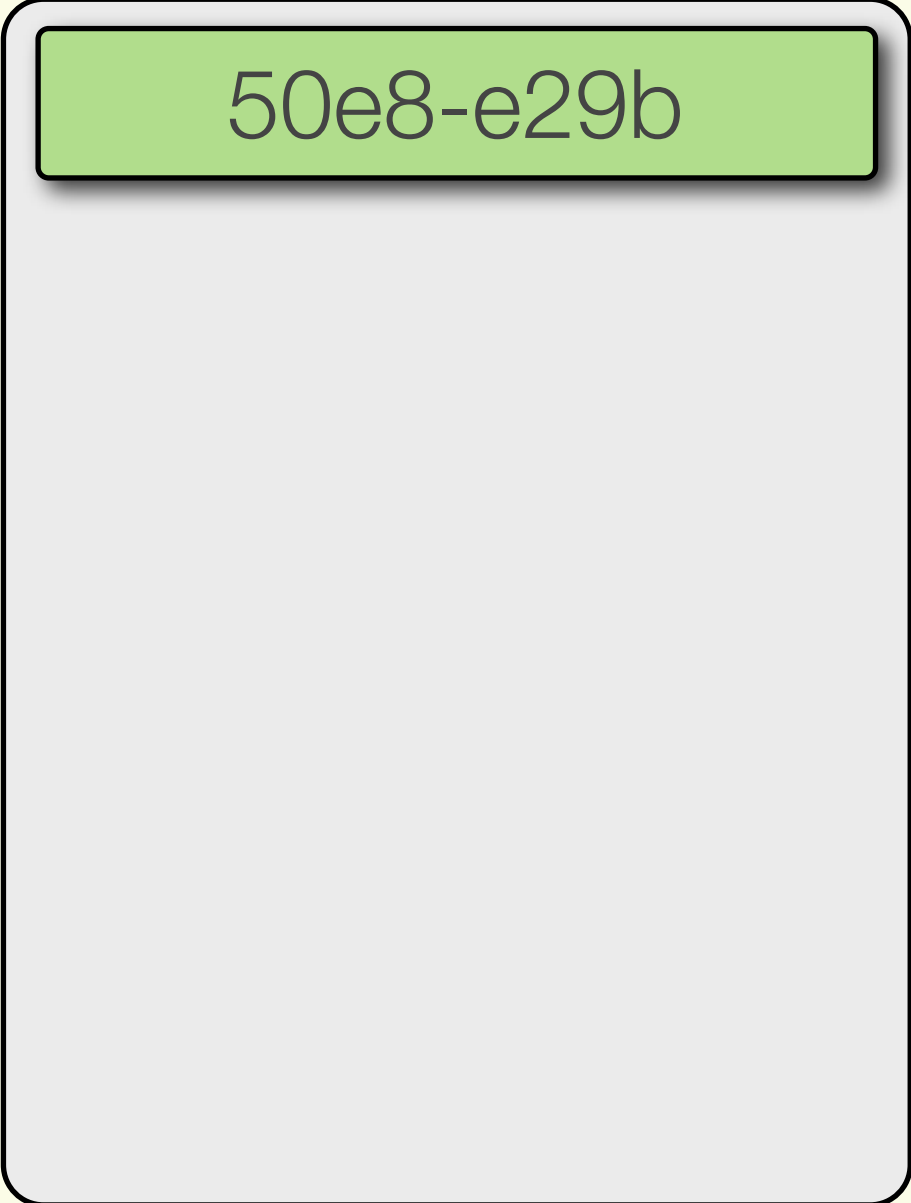
50e8-e29b	
birth_year	1994
fname	Justin
lname	Bieber

2ab1-f1b7	
birth_year	1978
email	a@kutcher.com
fname	Ashton
lname	Kutcher

Users

Ex: user's Tweets

“For each user, tweets he has made”



50e8-e29b

Timeline

Ex: user's Tweets

“For each user, tweets he has made”

50e8-e29b	
0	@LiveLoveKary glad you had a good birthday #muchlove

Timeline

Ex: user's Tweets

“For each user, tweets he has made”

50e8-e29b	
1	@NickDeMoura happy bday my dude.
0	@LiveLoveKary glad you had a good birthday #muchlove

Timeline

Ex: user's Tweets

“For each user, tweets he has made”

50e8-e29b	
2	@MickyArison @miamiHEAT thanks for the gam tonight
1	@NickDeMoura happy bday my dude.
0	@LiveLoveKary glad you had a good birthday #muchlove

Timeline

Ex: user's Tweets

“For each user, tweets he has made”

50e8-e29b	
3	still a little tired. back in the studio today with Timbaland
2	@MickyArison @miamiHEAT thanks for the gam tonight
1	@NickDeMoura happy bday my dude.
0	@LiveLoveKary glad you had a good birthday #muchlove

Timeline

There's more

- Secondary indexes
- Distributed counters
- Composite columns

1. What is Apache Cassandra

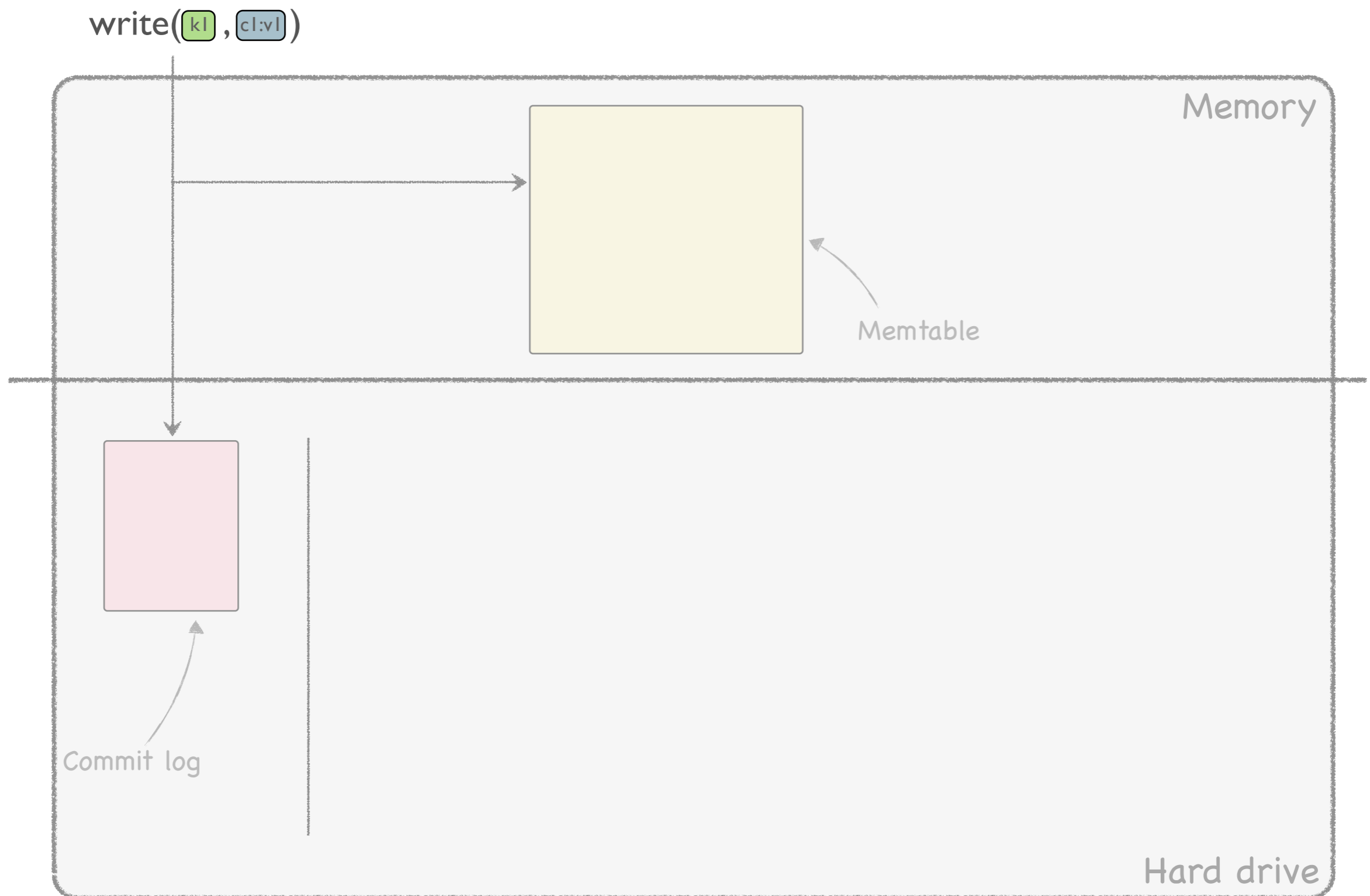
2. Data Model

3. The storage engine

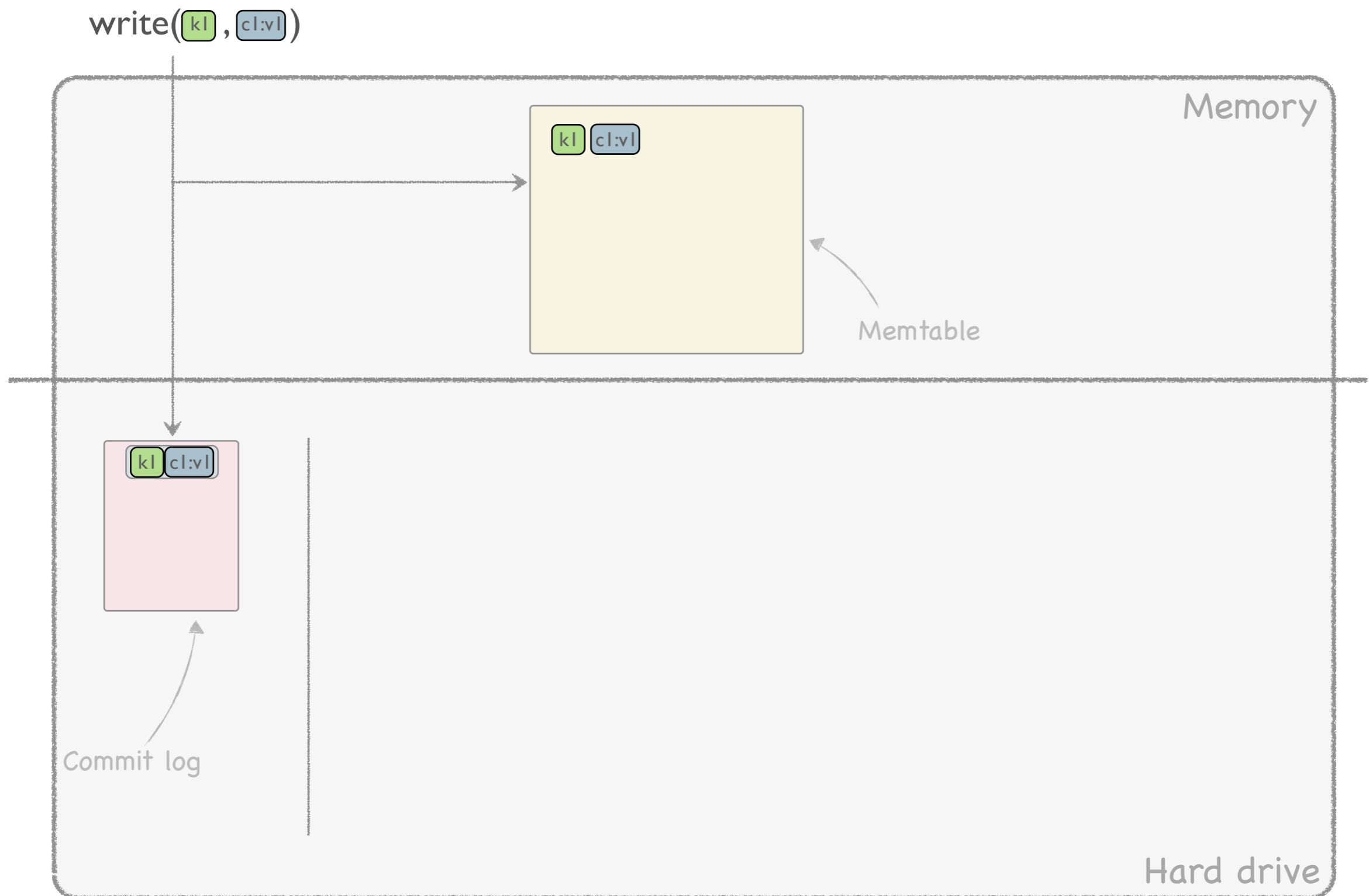
Goal

- Writes are harder than reads to scale
- Spinning disks aren't good with random I/O
- Goal: minimize random I/O

A write's journey



A write's journey

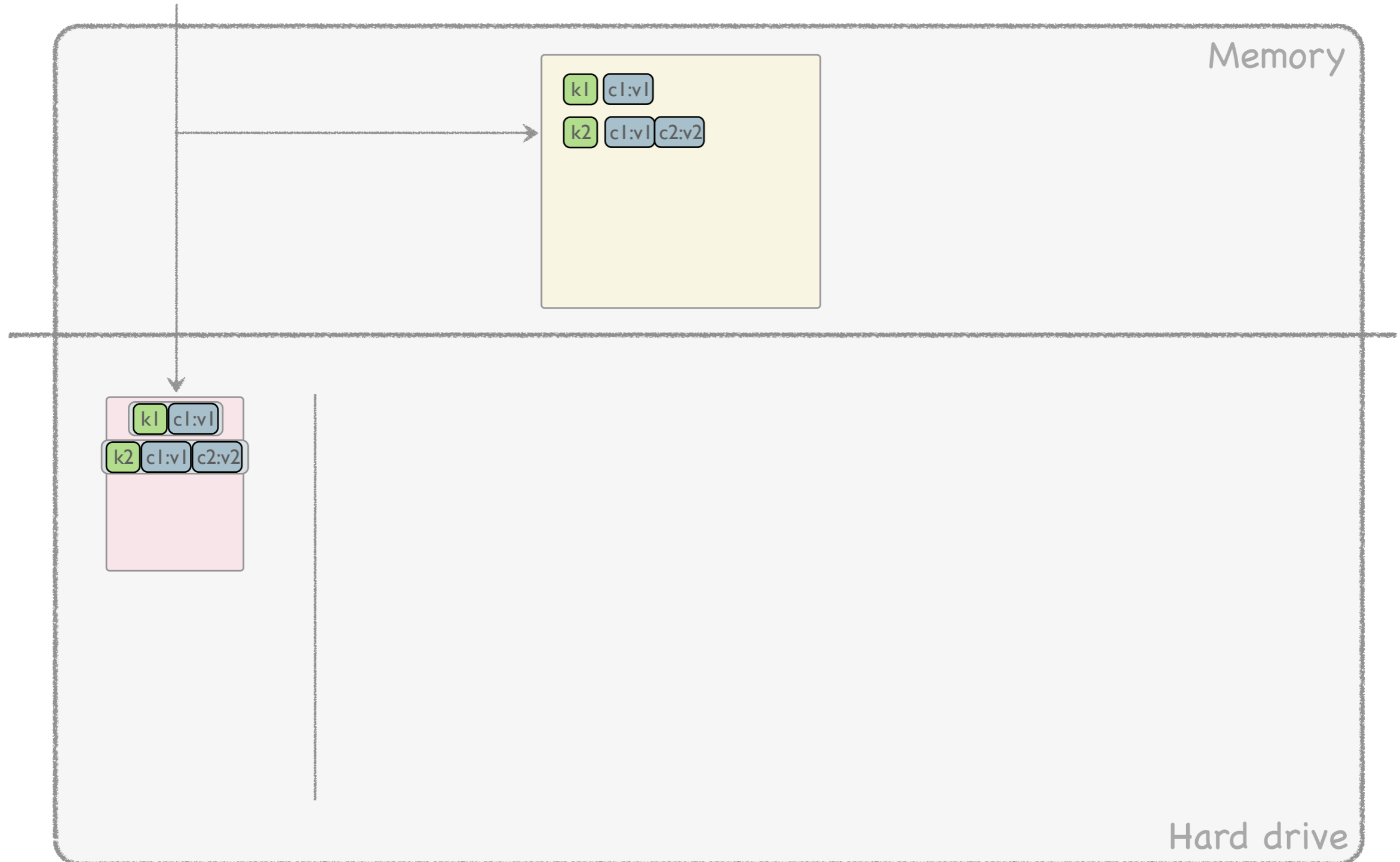


A write's journey



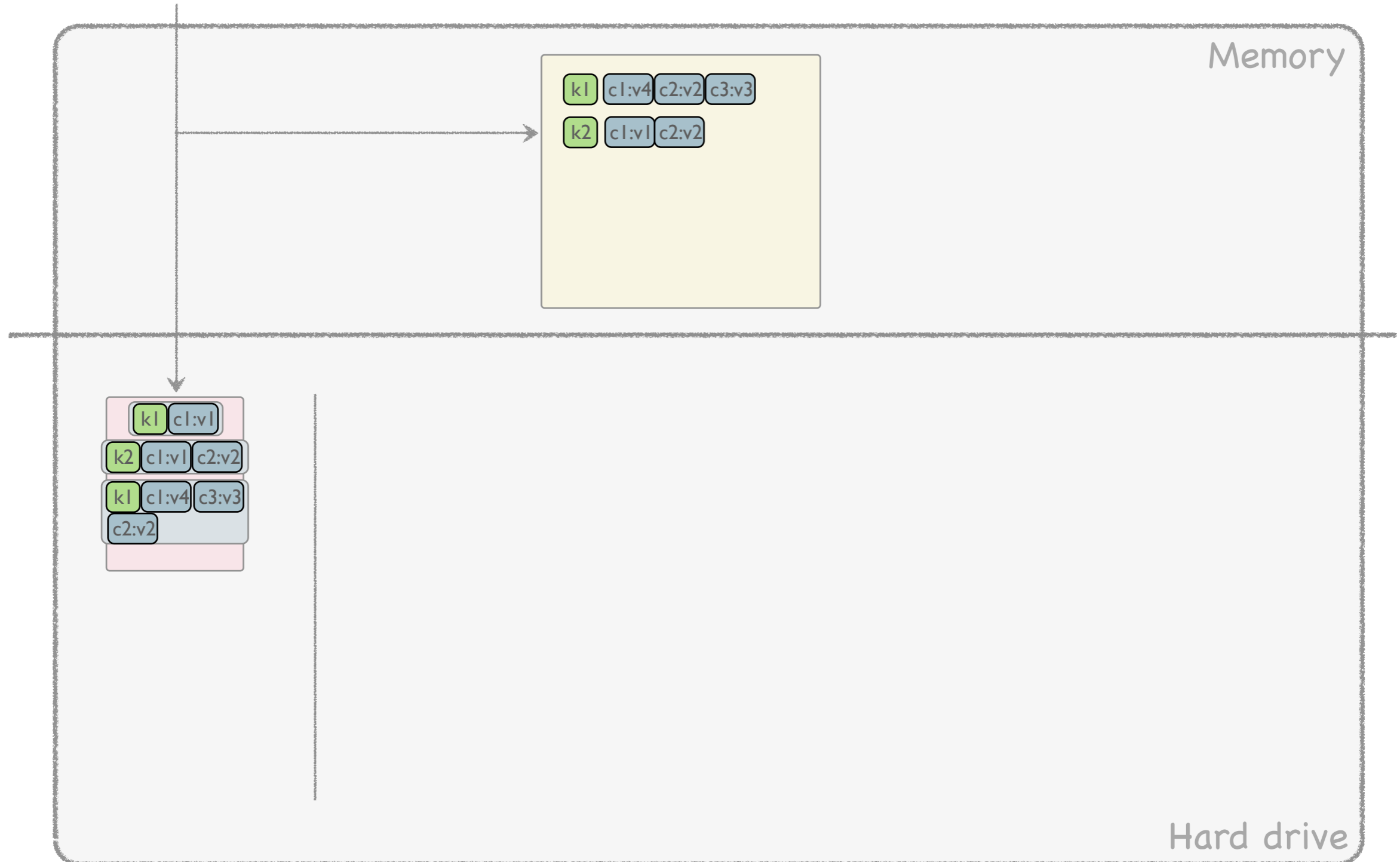
A write's journey

write(k2, c1:v1 c2:v2)

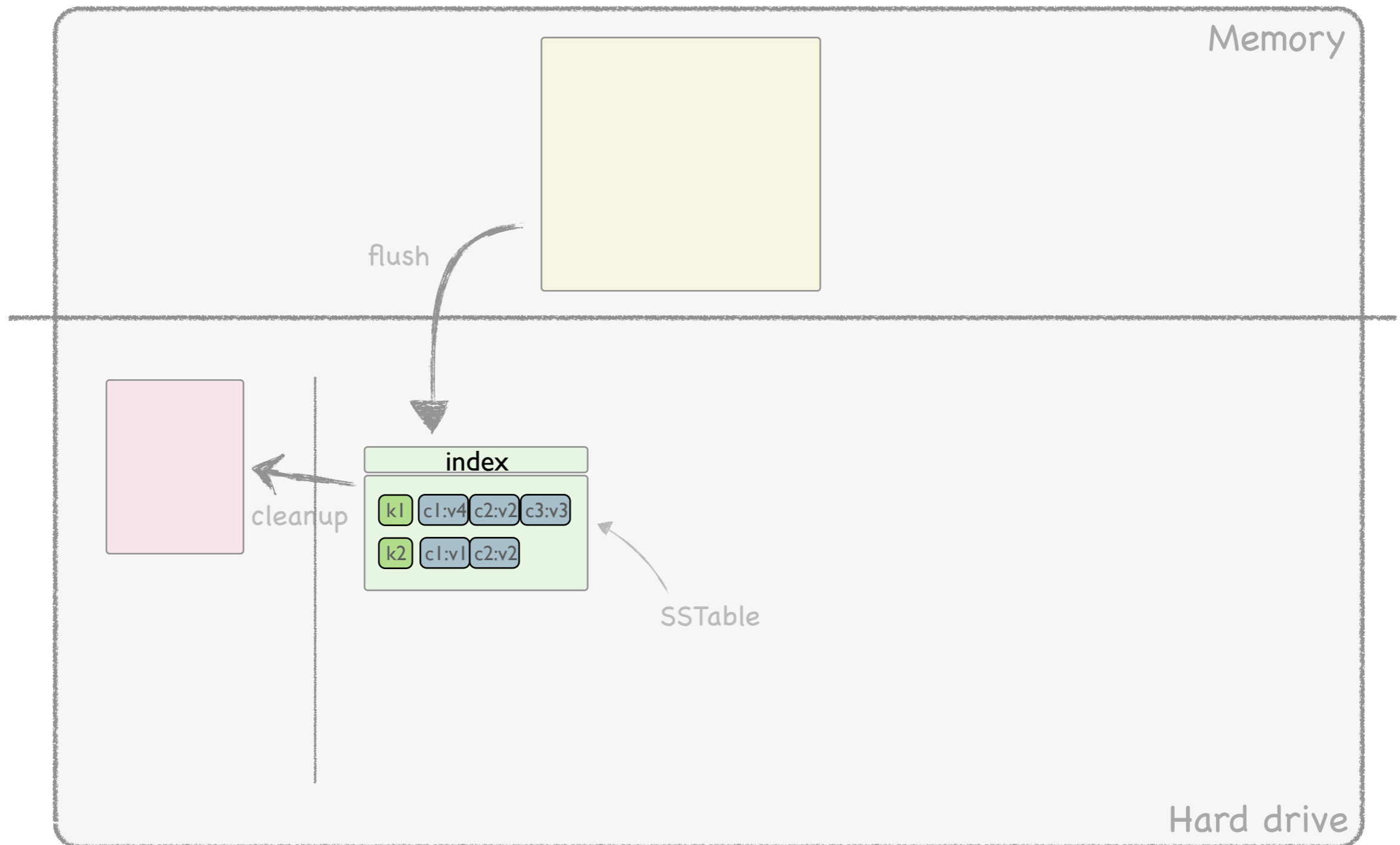


A write's journey

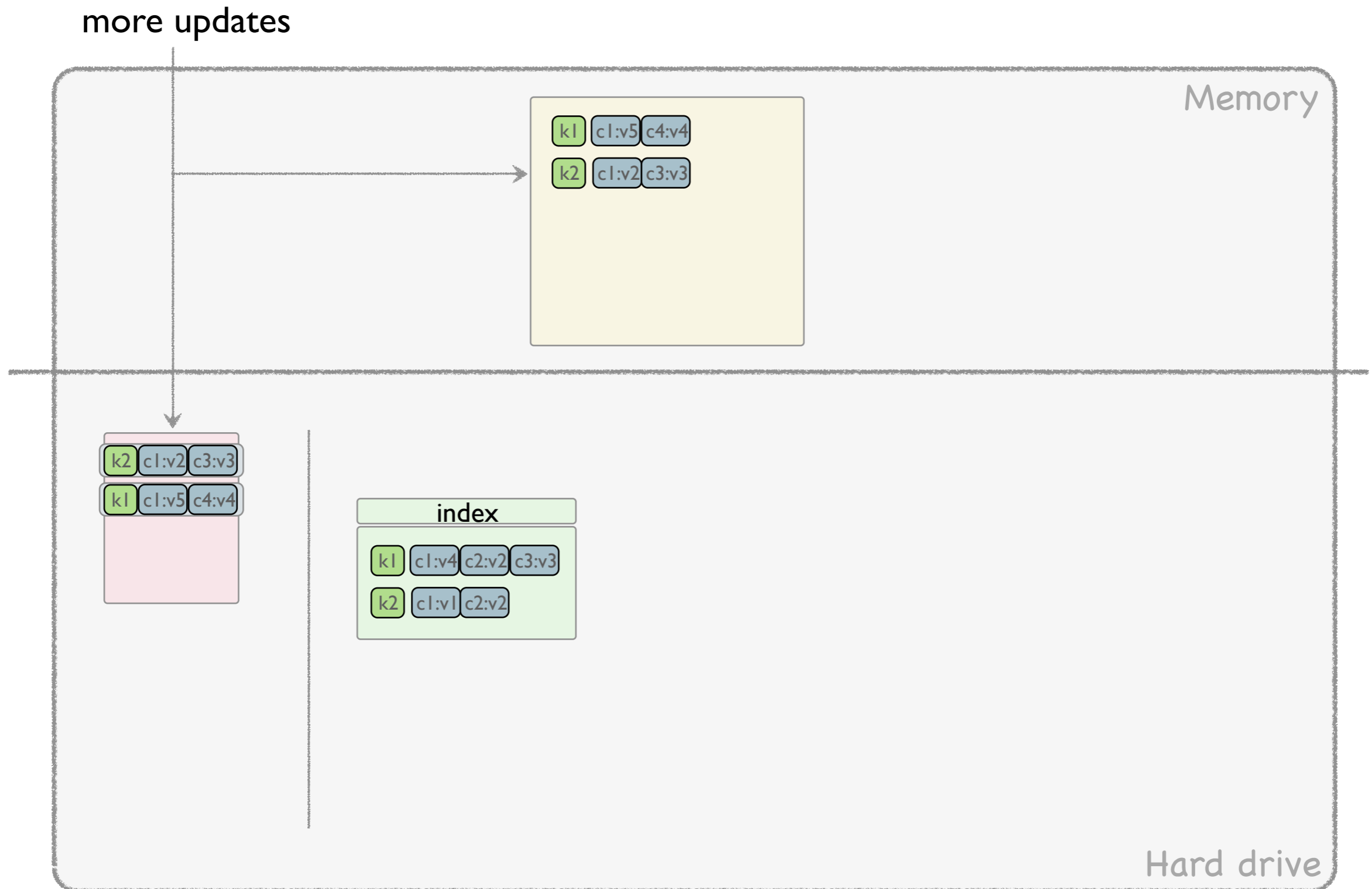
write(k1, c1:v4 c3:v3 c2:v2)



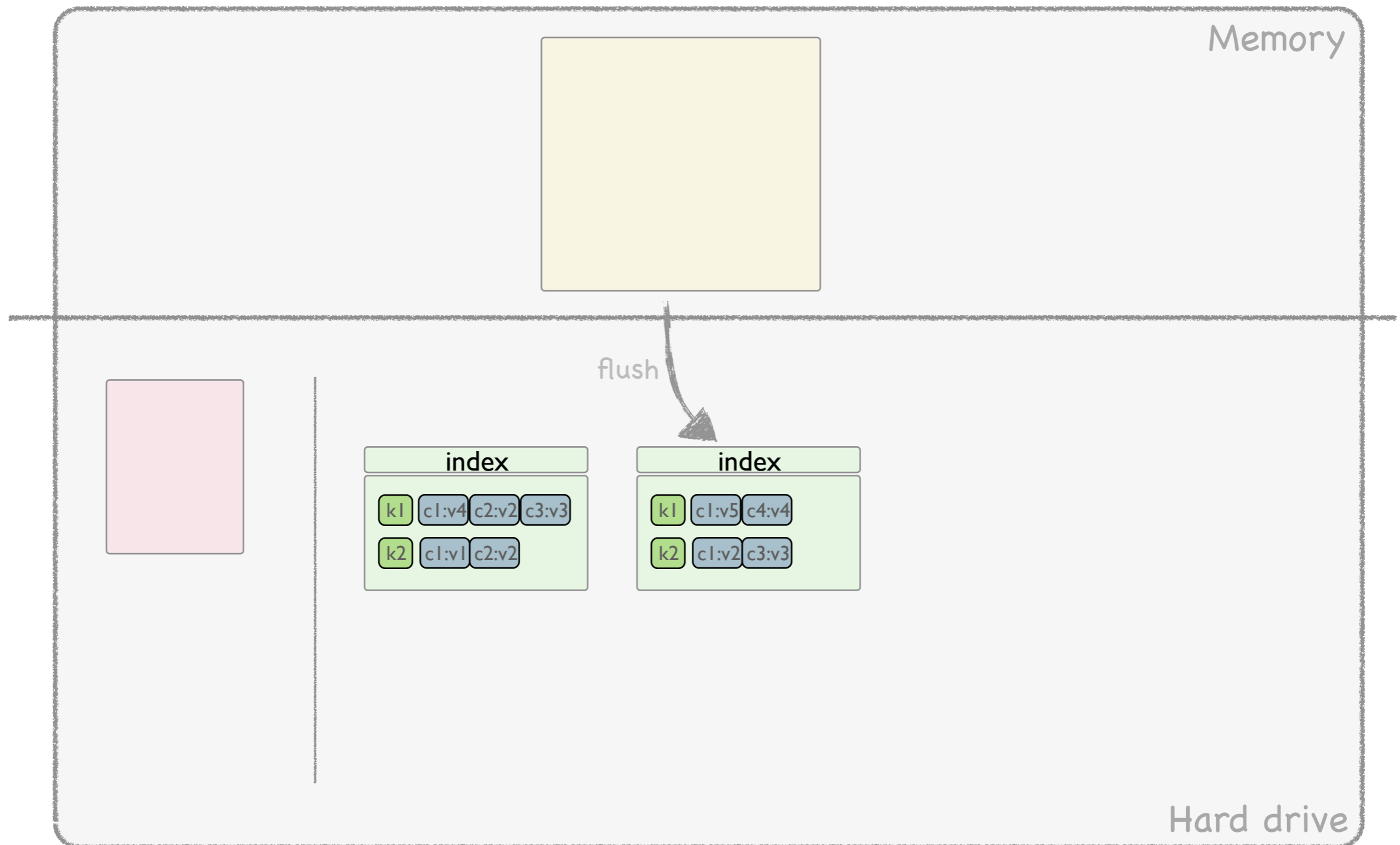
A write's journey



A write's journey



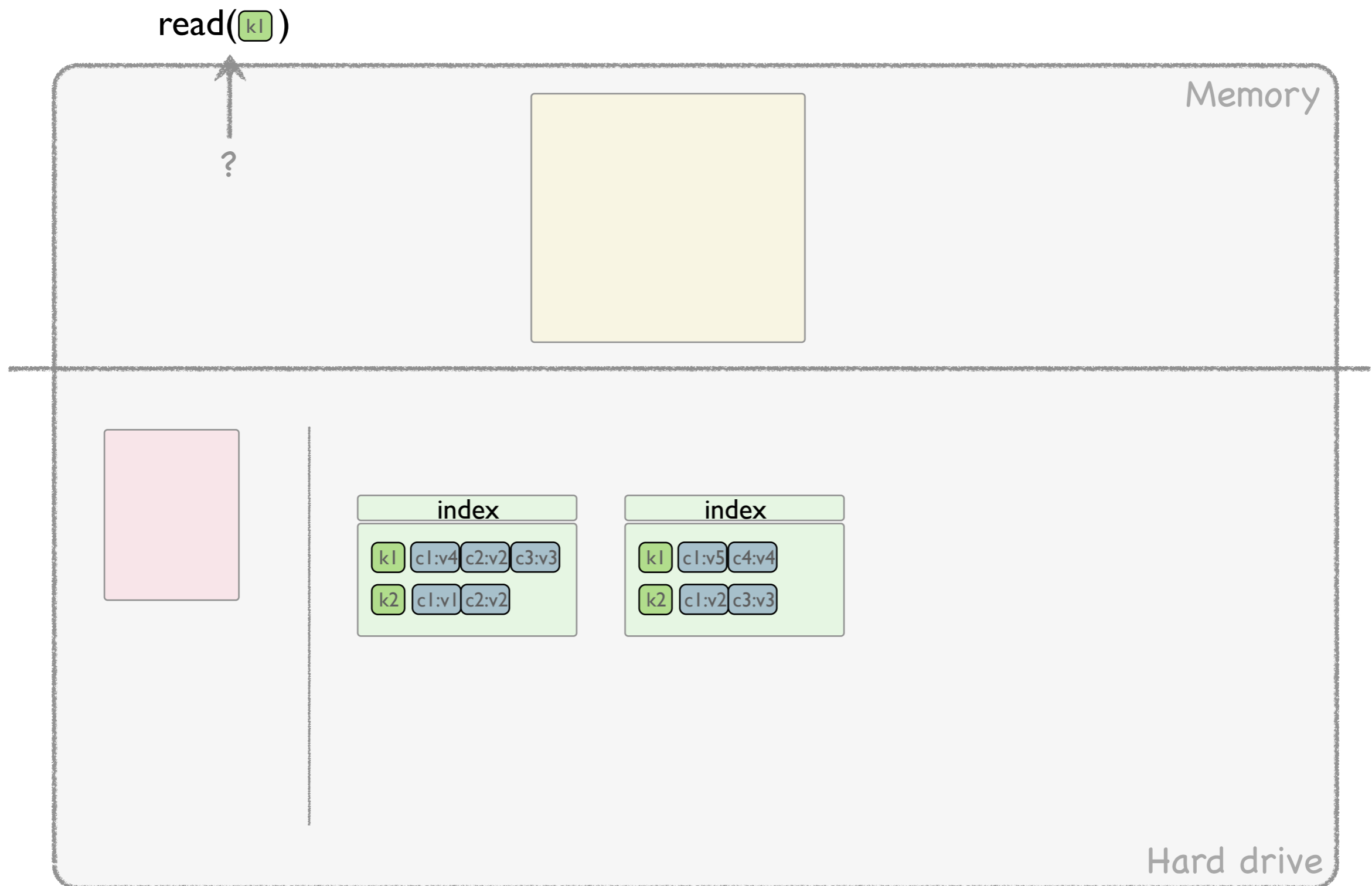
A write's journey



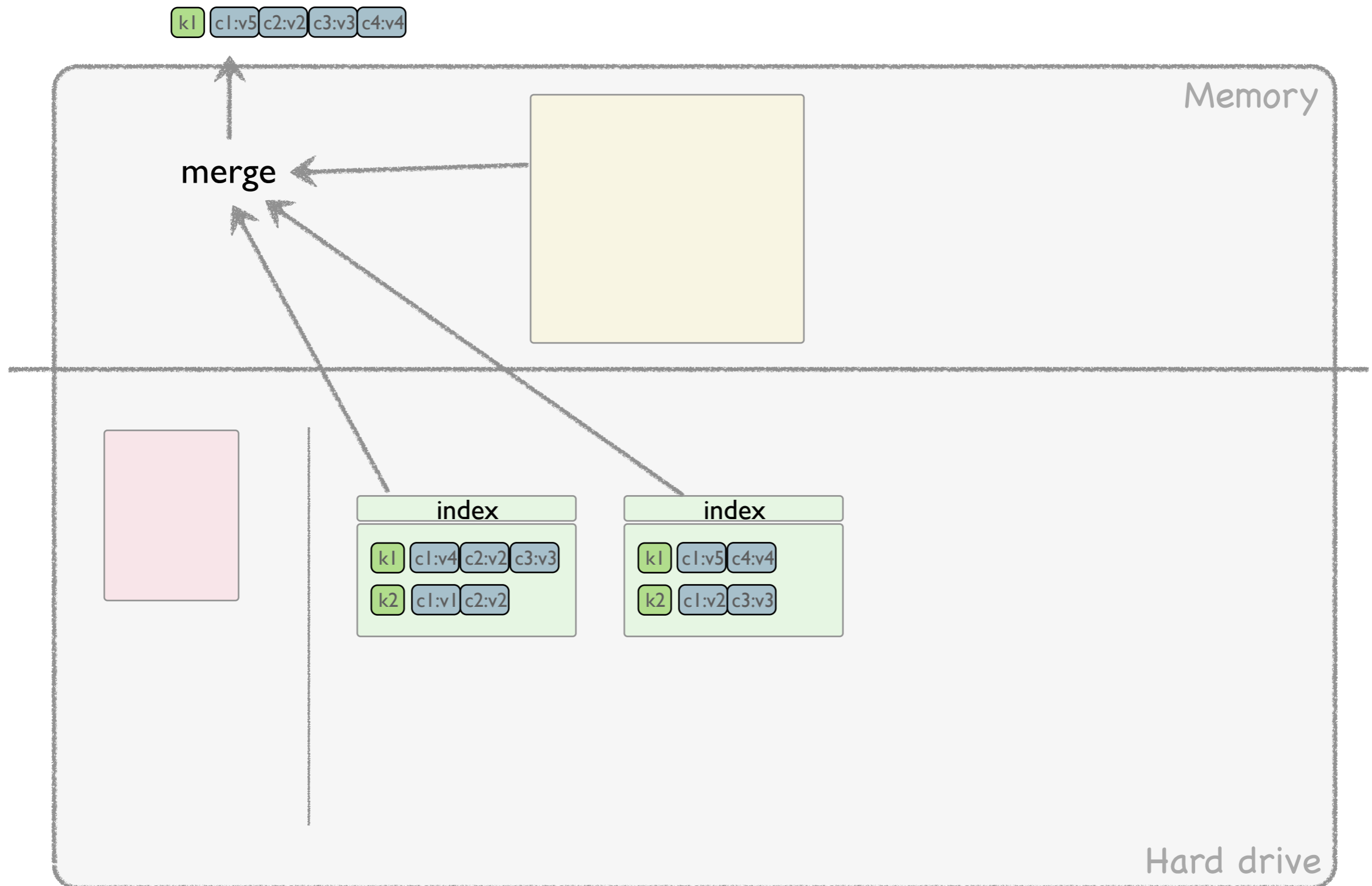
Writes properties

- No reads or seeks
- Only sequential I/O
- Immutable SSTables: easy snapshots

A read's journey



A read's journey

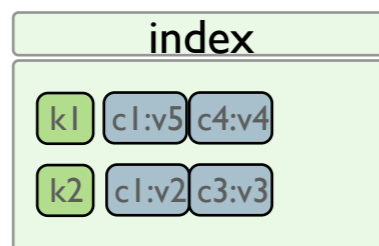
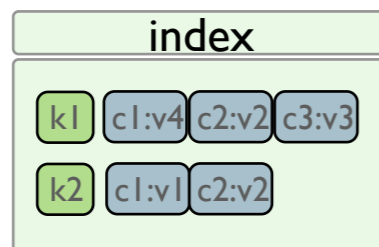


Compaction

- Goal: keep the number of SSTables low
- Merge sort against multiple sstables
- Sequential I/O

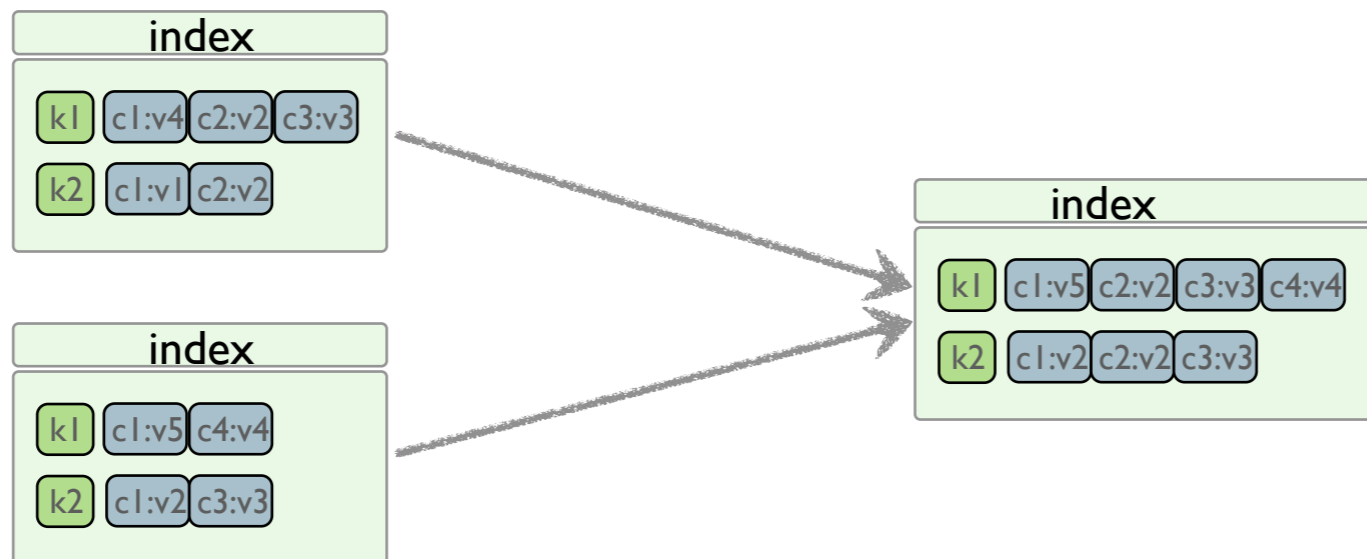
Compaction

- Goal: keep the number of SSTables low
- Merge sort against multiple sstables
- Sequential I/O



Compaction

- Goal: keep the number of SSTables low
- Merge sort against multiple sstables
- Sequential I/O

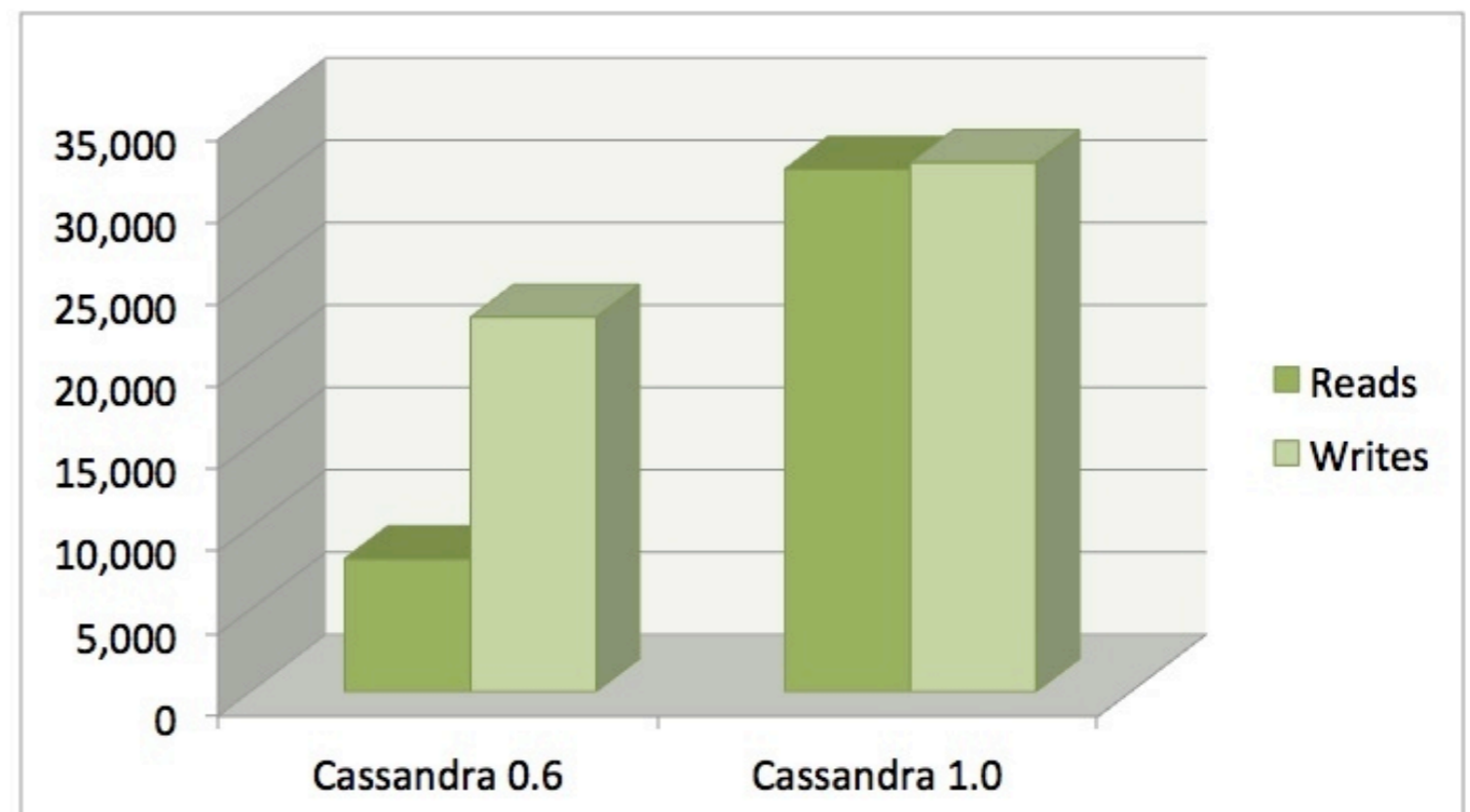


Optimizations

- Row Cache
- Bloom filters: eliminates whole SSTable
- Key Cache
- Rows & Columns Indexes
- ...

Other features

- Compression
- Checksums
- Time to live



Questions?

- Cassandra 1.1 scheduled for next month
- <http://cassandra.apache.org/>
- <http://wiki.apache.org/cassandra/>
- <http://www.datastax.com/docs/1.0>