Distributed databases building blocks

OREN EINI

WIZARD

HIBERNATING RHINOS

I write databases for a living

- Rhino DHT
- RavenDB
  - Voron

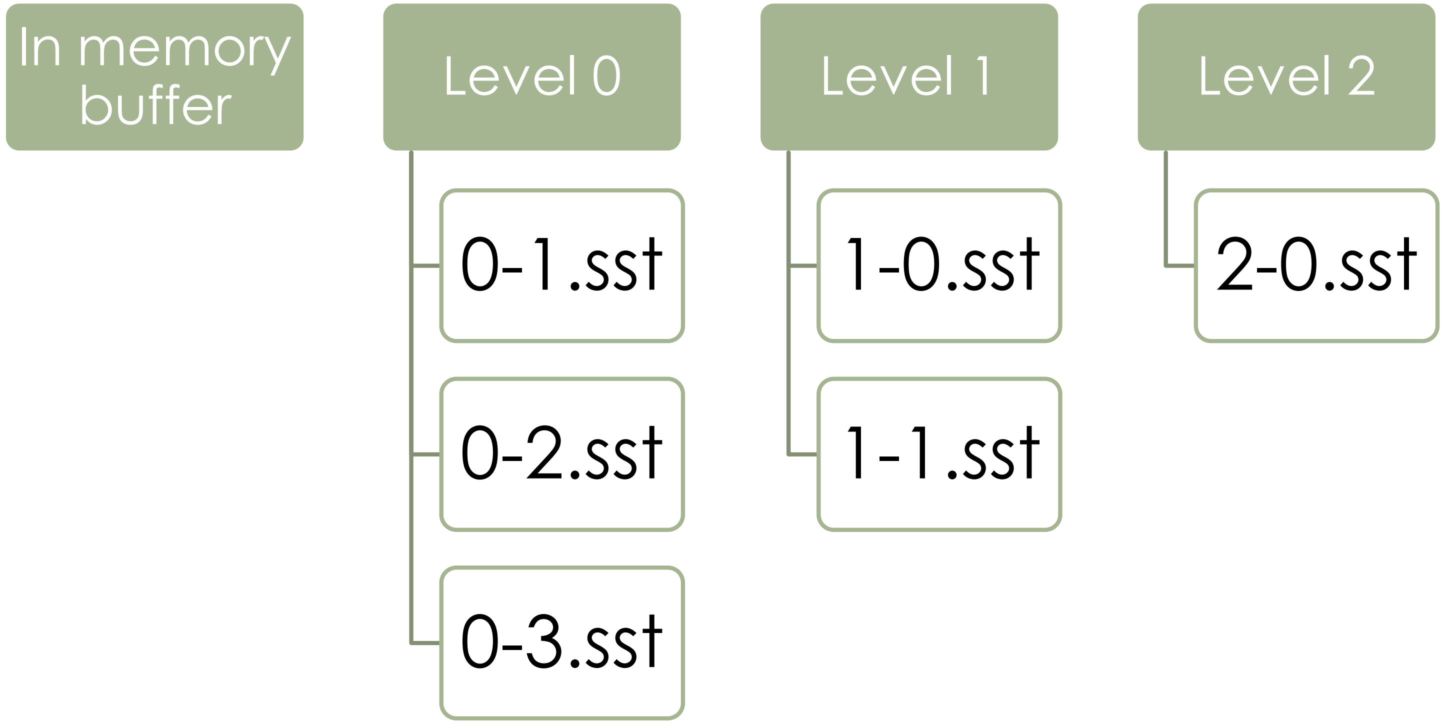
- This is hard ☺
  - Also fun

Storing data

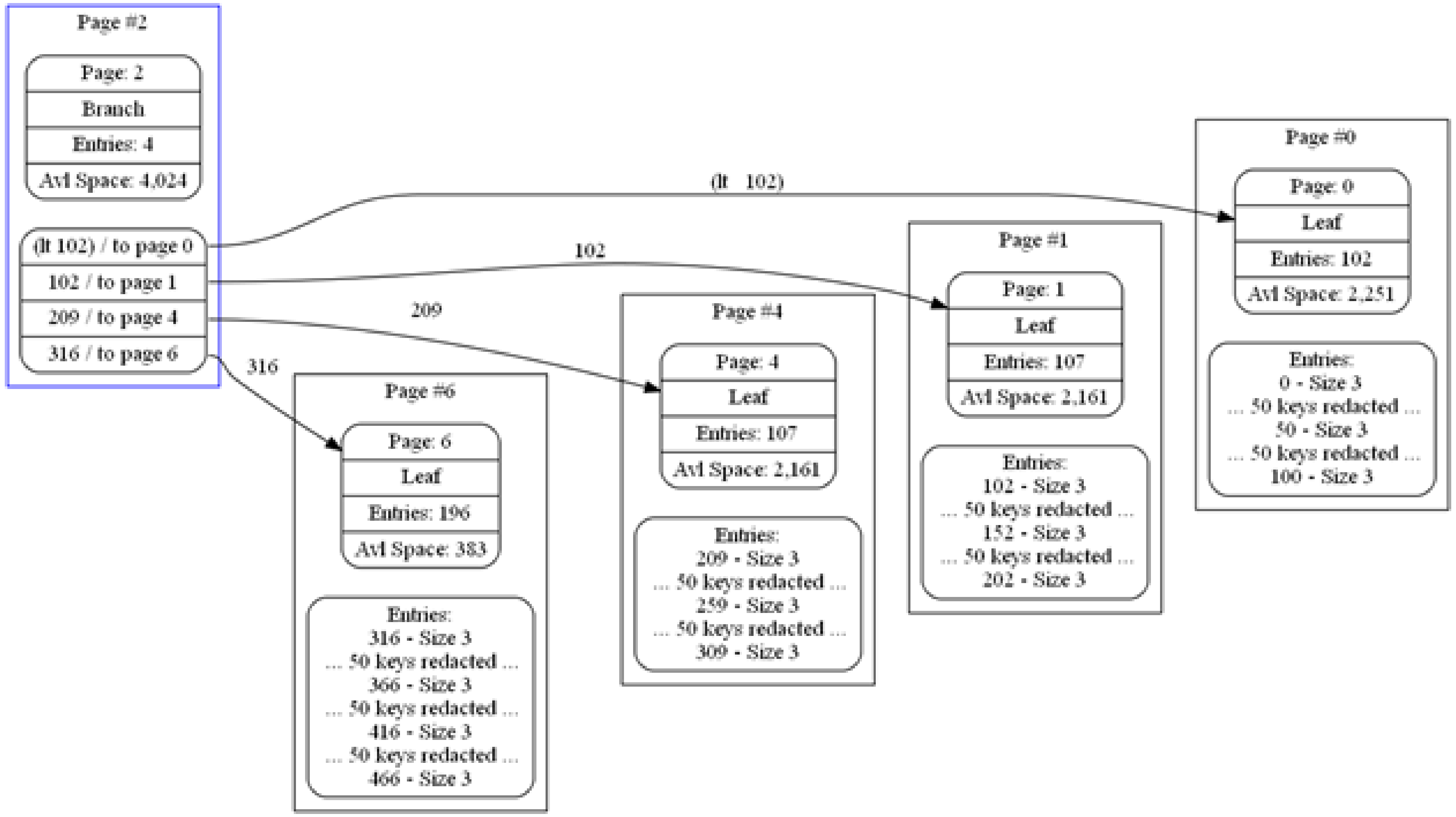
- What do we need?
- File.Open ?

- The disk goes round & round, round & round, round & round

Log Structure Merge



B+Trees


Page #2
Page: 2
Branch
Entries: 4
Avl Space: 4,024
(lt 102) / to page 0
102 / to page 1
209 / to page 4
316 / to page 6
(lt 102)
102
209
316
Page #6
Page: 6
Leaf
Entries: 196
Avl Space: 383
Entries:
316 - Size 3
... 50 keys redacted ...
366 - Size 3
... 50 keys redacted ...
416 - Size 3
... 50 keys redacted ...
466 - Size 3
Page #4
Page: 4
Leaf
Entries: 107
Avl Space: 2,161
Entries:
209 - Size 3
... 50 keys redacted ...
259 - Size 3
... 50 keys redacted ...
309 - Size 3
Page #1
Page: 1
Leaf
Entries: 107
Avl Space: 2,161
Entries:
102 - Size 3
... 50 keys redacted ...
152 - Size 3
... 50 keys redacted ...
202 - Size 3
Page #0
Page: 0
Leaf
Entries: 102
Avl Space: 2,251
Entries:
0 - Size 3
... 50 keys redacted ...
50 - Size 3
... 50 keys redacted ...
100 - Size 3

Concurrency & Isolations

Tx 1

- Begin Tx
- Read X (X = 1)
- Write X = 2
- Commit

Tx 2

- Begin Tx
- Read X (X = 1)
- Read X (X = ?)

Tx 3

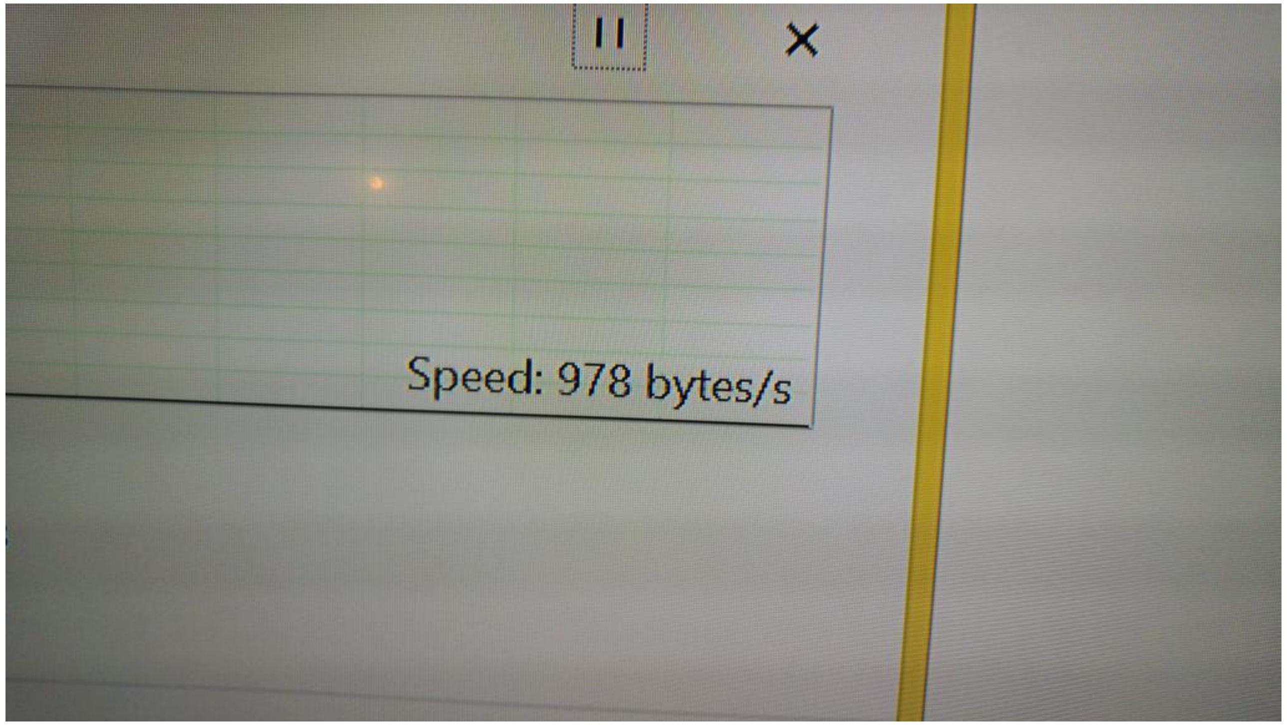
- Begin Tx
- Read X (X = 2)

Implementing concurrency

Locks

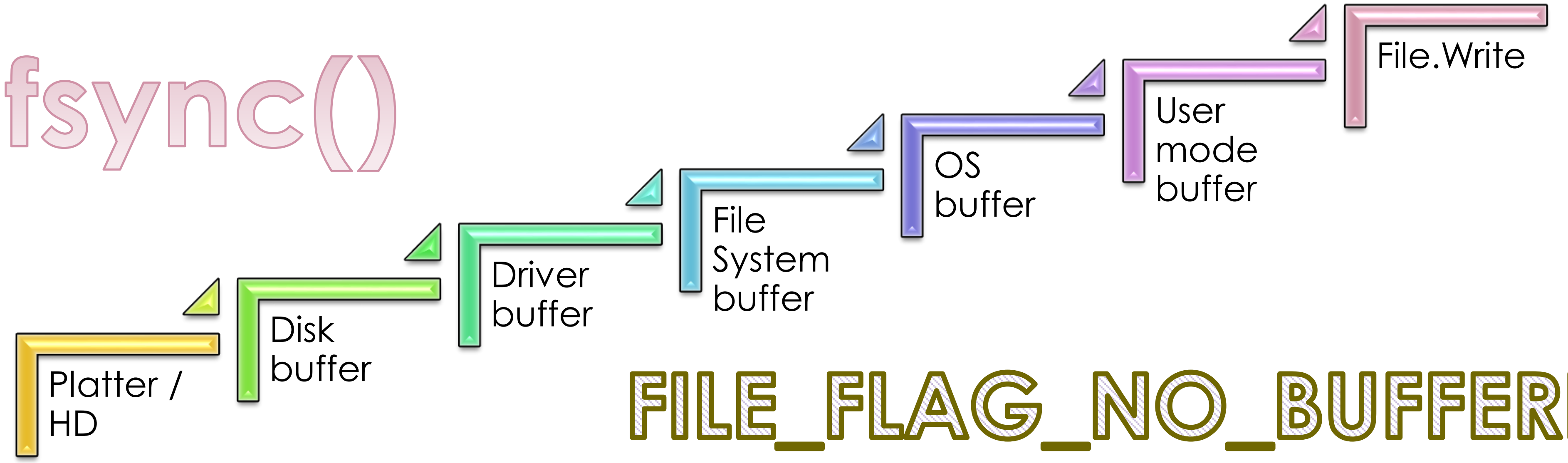
MVCC

I/O IS **SLOW**


Speed: 978 bytes/s

Durability / transactions / fsync

fsync()

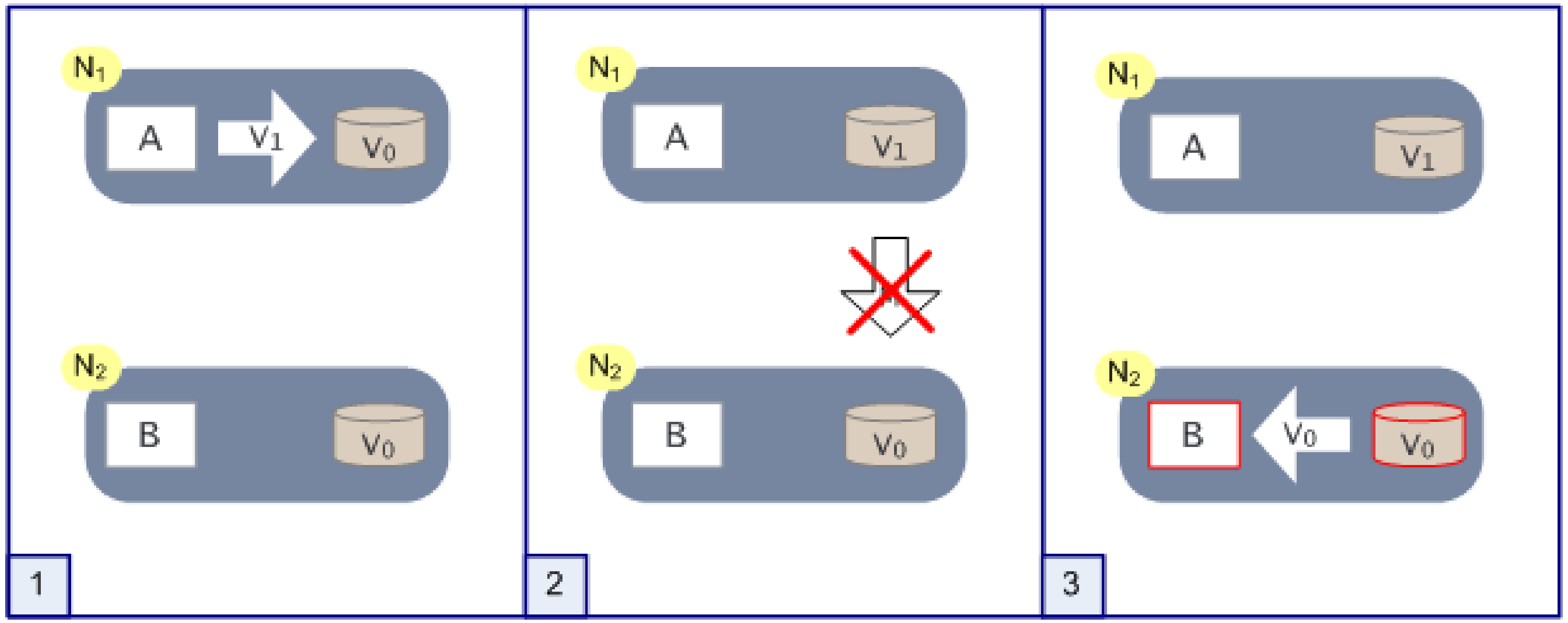


O_SYNC

FILE_FLAG_NO_BUFFERING | FILE_FLAG_WRITE_THROUGH

The curse of the single node…

What about the distributed part?



Before that…

- What is the distribution model?
  - Consensus?
  - Collaborative?
  - Repair?
- Partition model

Consensus

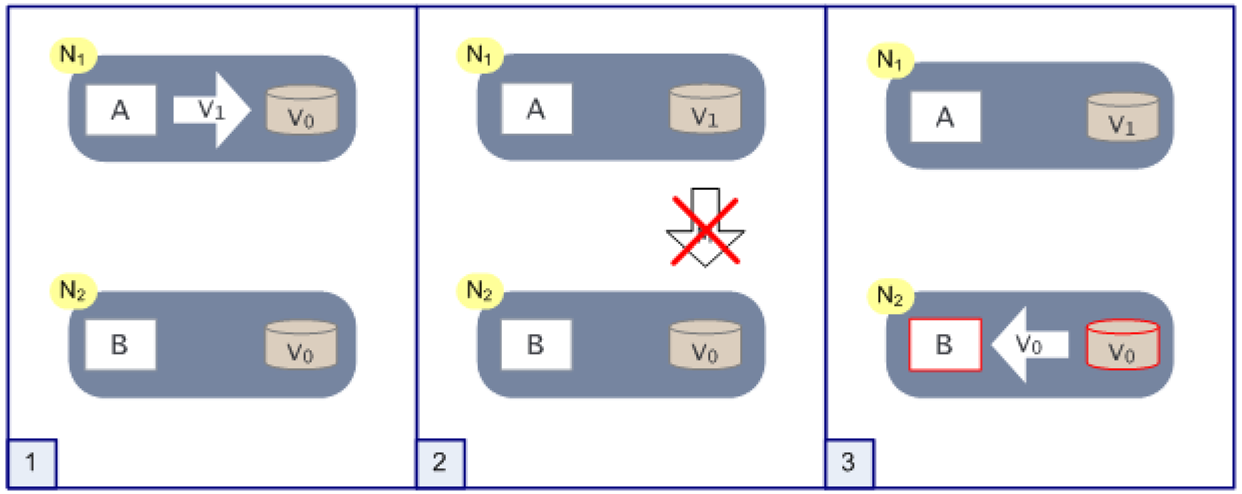
Paxos ???

Raft

Consensus – log of operations

- Set x = 1
- Set y = 2
- Set n = 1
- Set x = y + n

Problems?



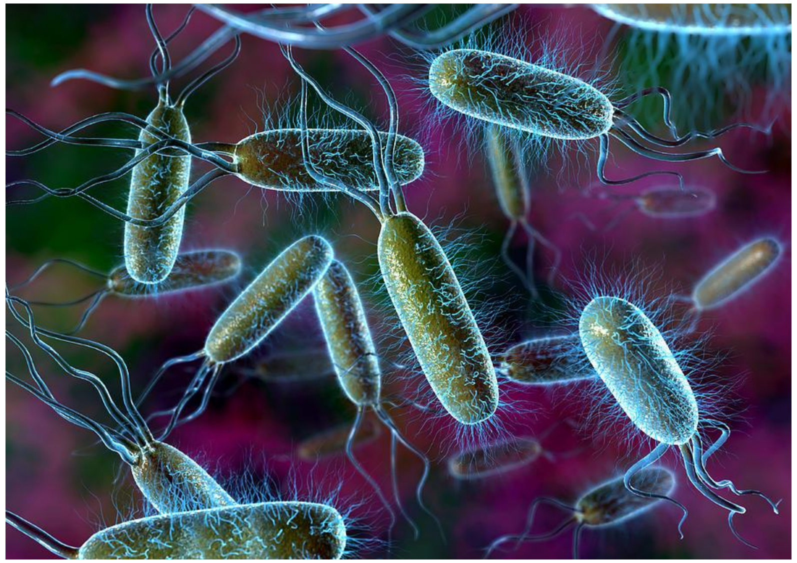
Collaborations (master / master)

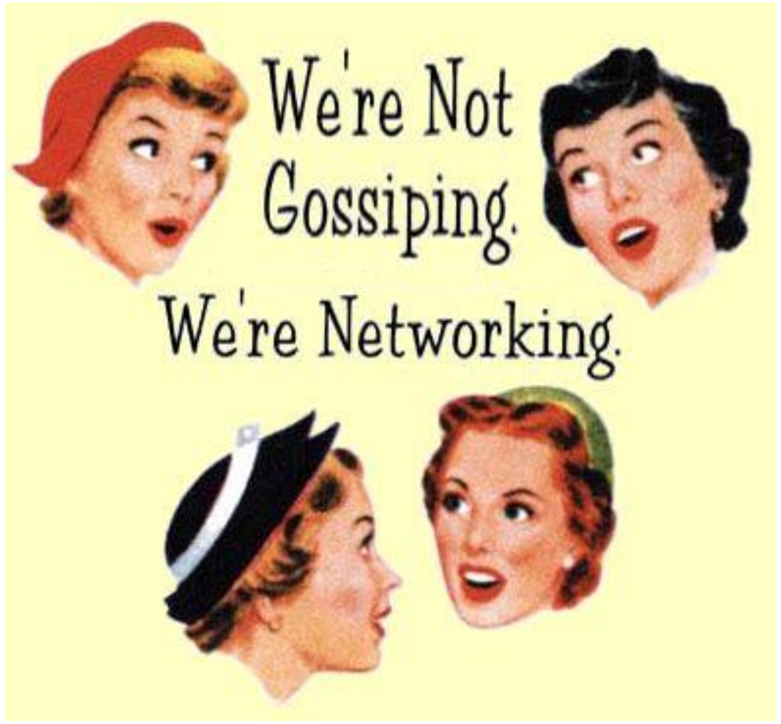
- Allow writes on any node
- Conflict-free replicated data types ?
- Merges

Sharded

- Some data on some node
- Multiple consensus groups

Gossip

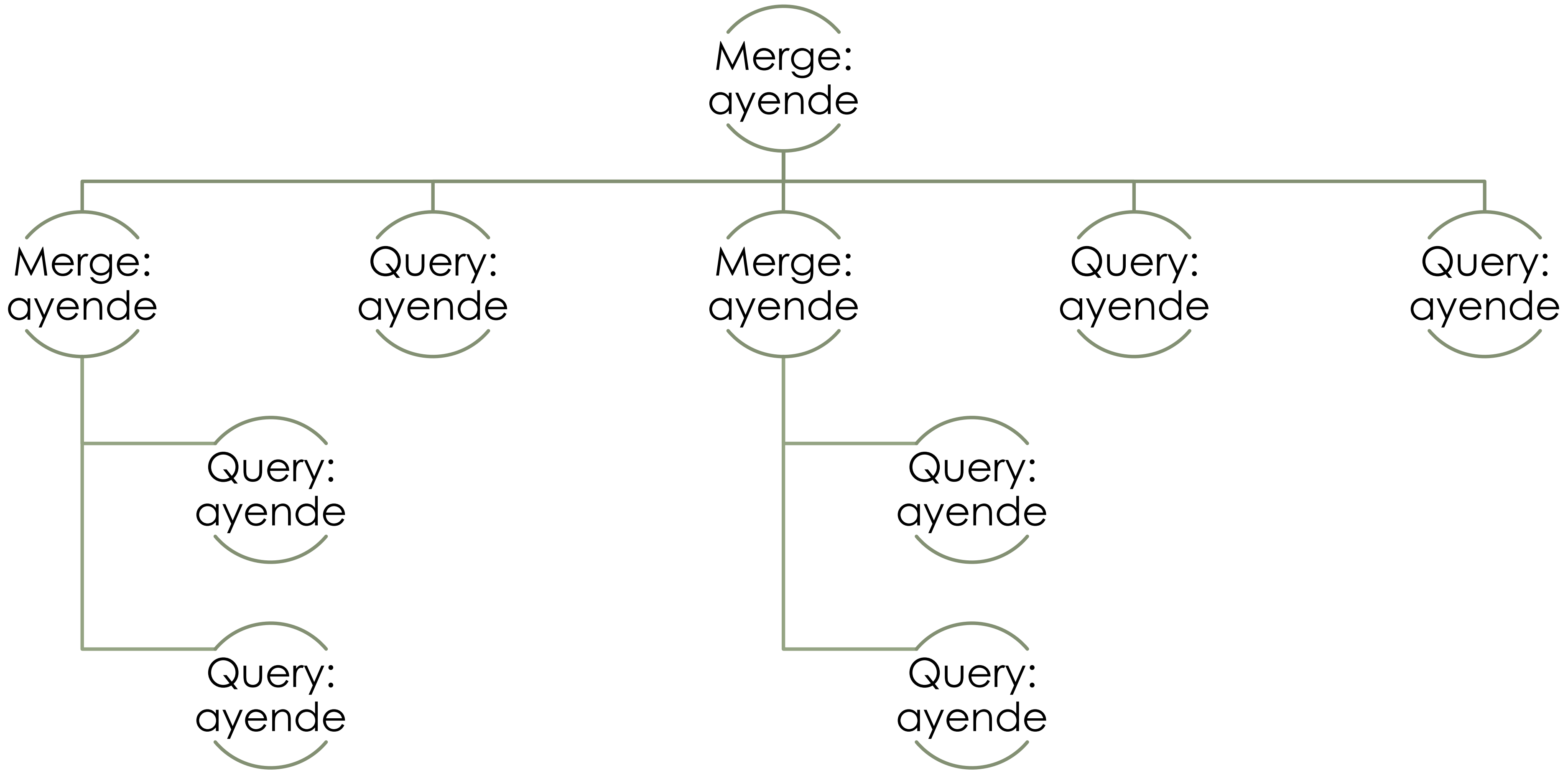




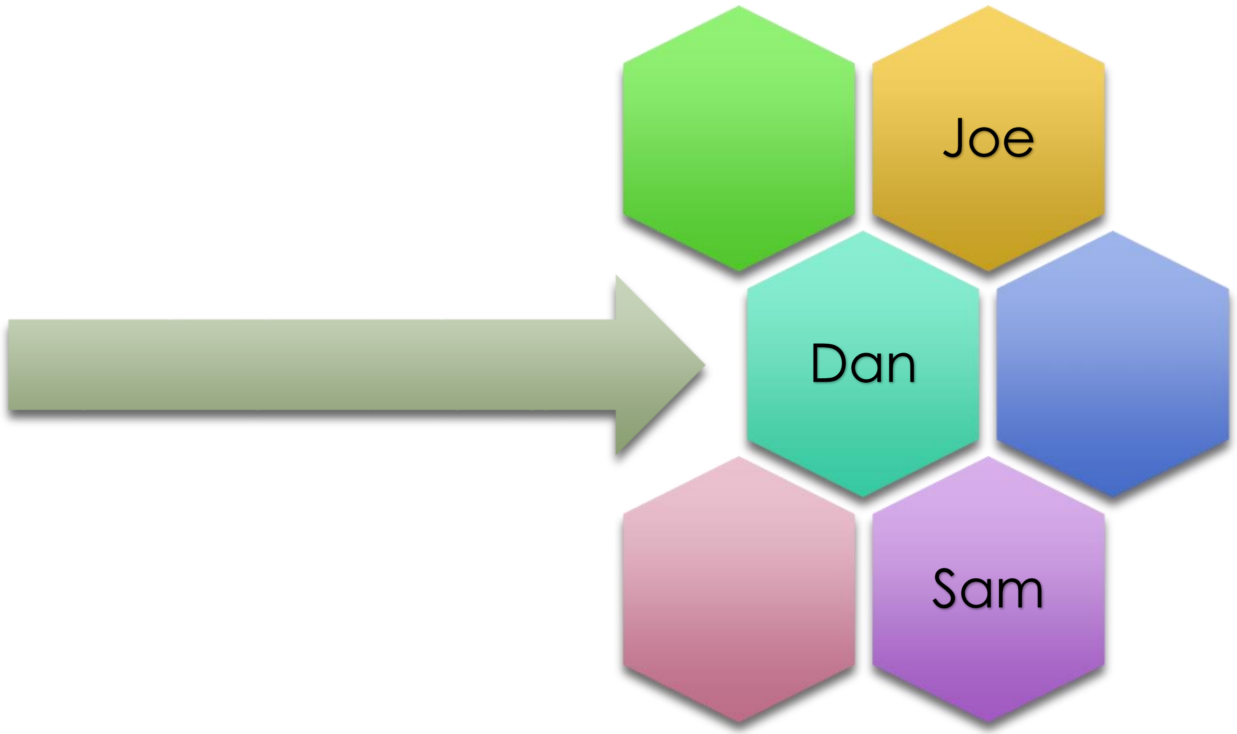
Data model

- So now you know how to store data (single node, multiple nodes)
- What do you *do* with this?

Entire dataset



Portioned data



Imagine a banking system…

- Core data model:
  - Accounts
  - Account operations (incoming, outgoing, fees, etc)

- New rule changes – gossip
- Reporting

Questions?