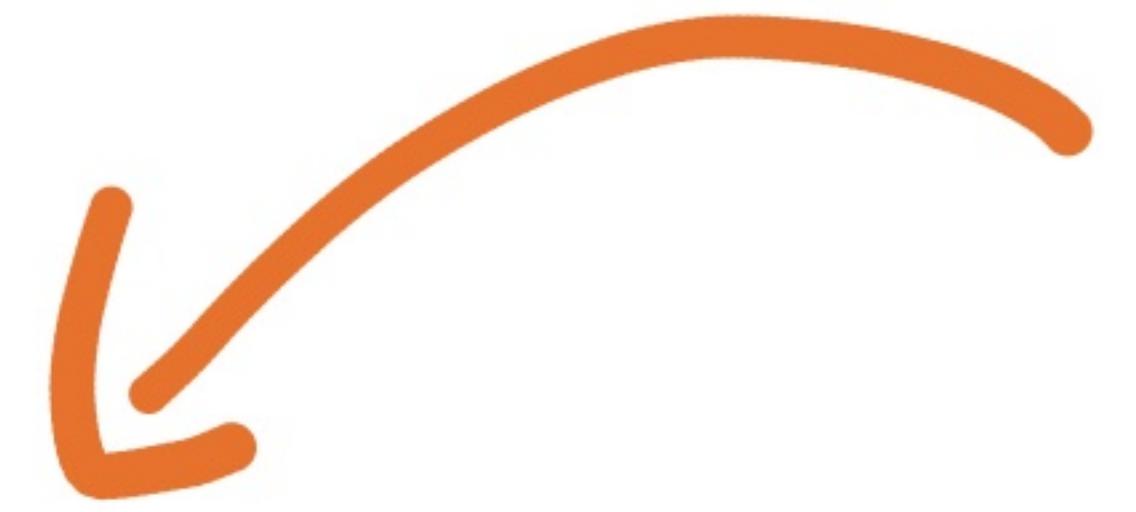


Jepsen

a syncing feeling

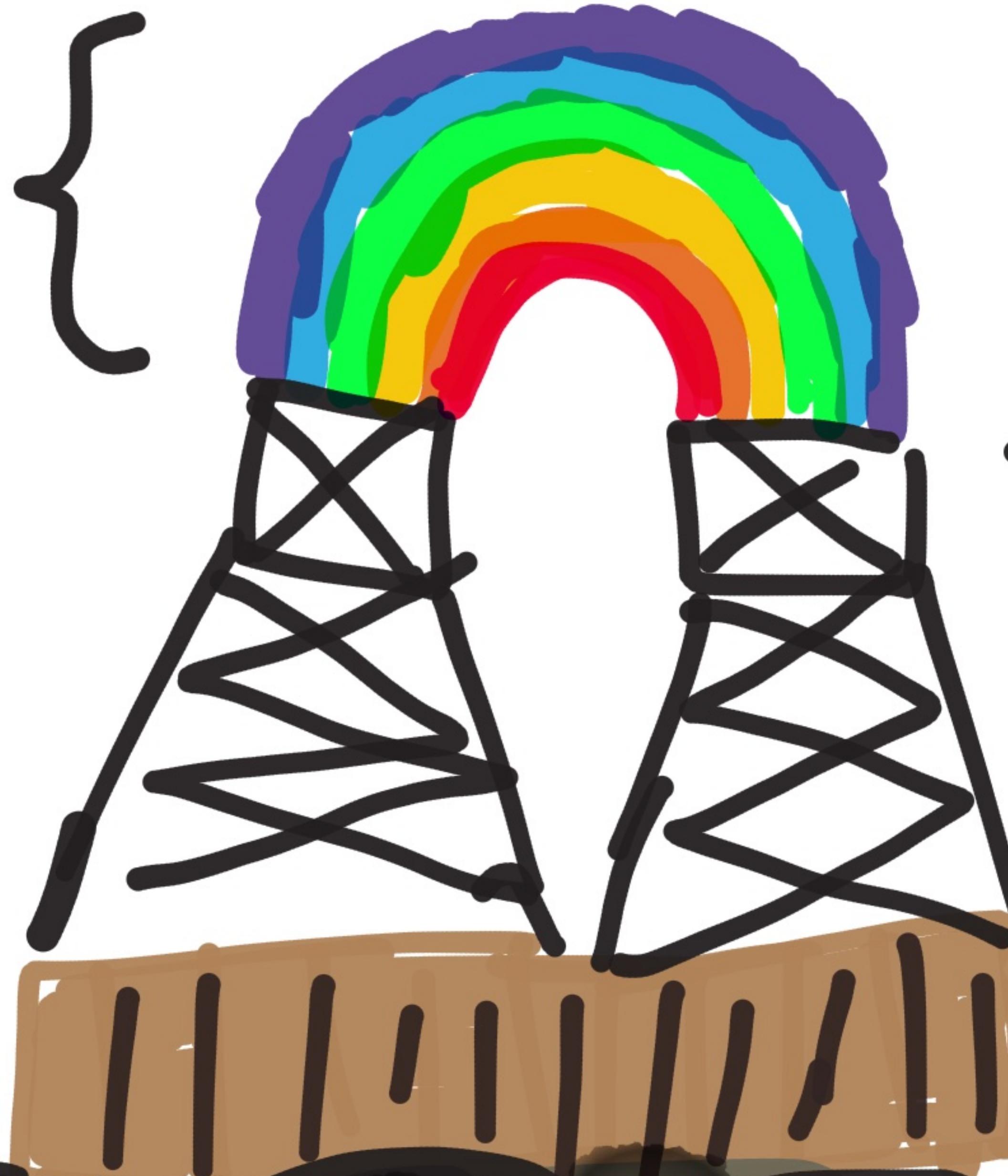


Kyle Kingsbury
@aphyr



I break
databases!

Public
API {



} API code

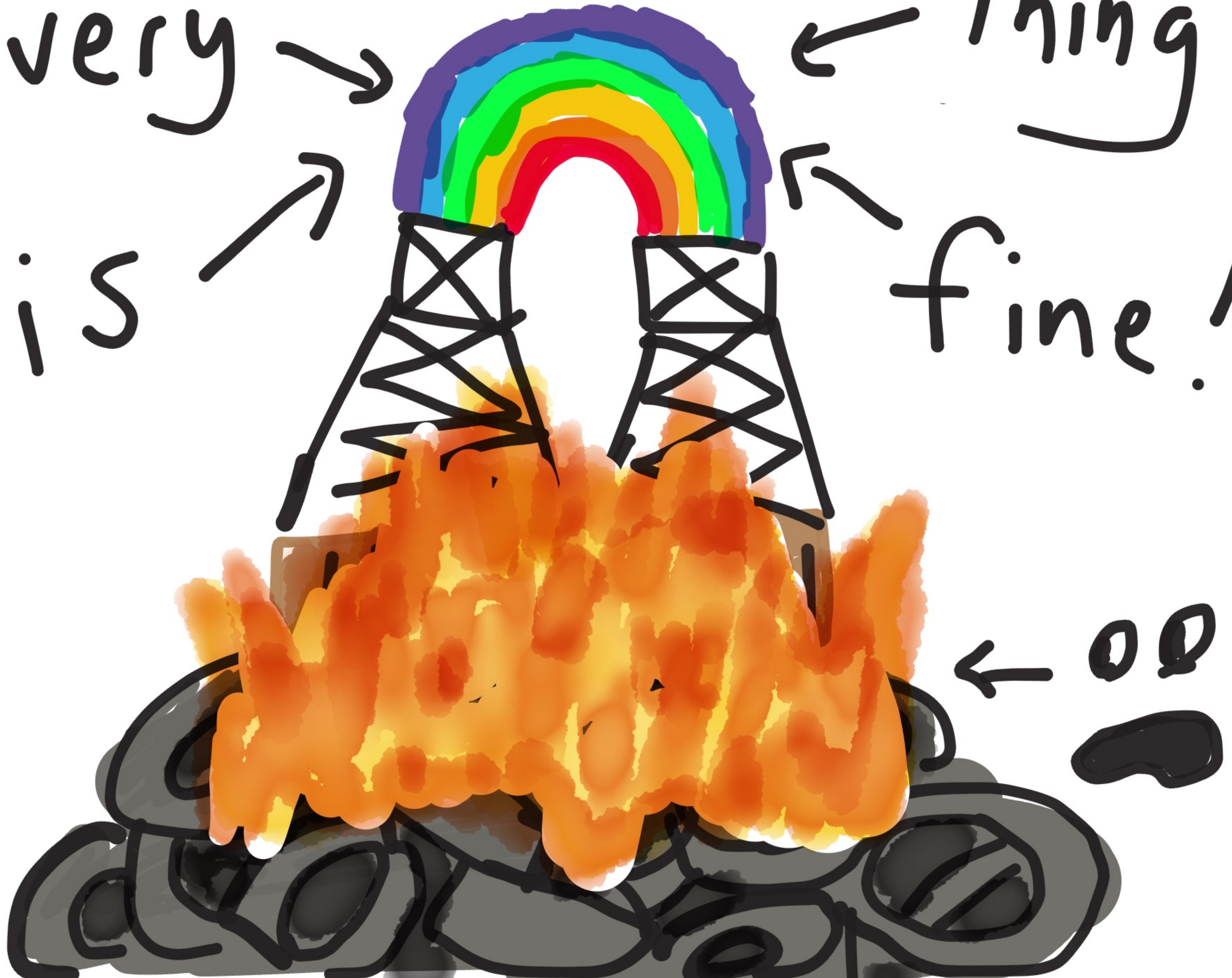
Ruby {



DBs

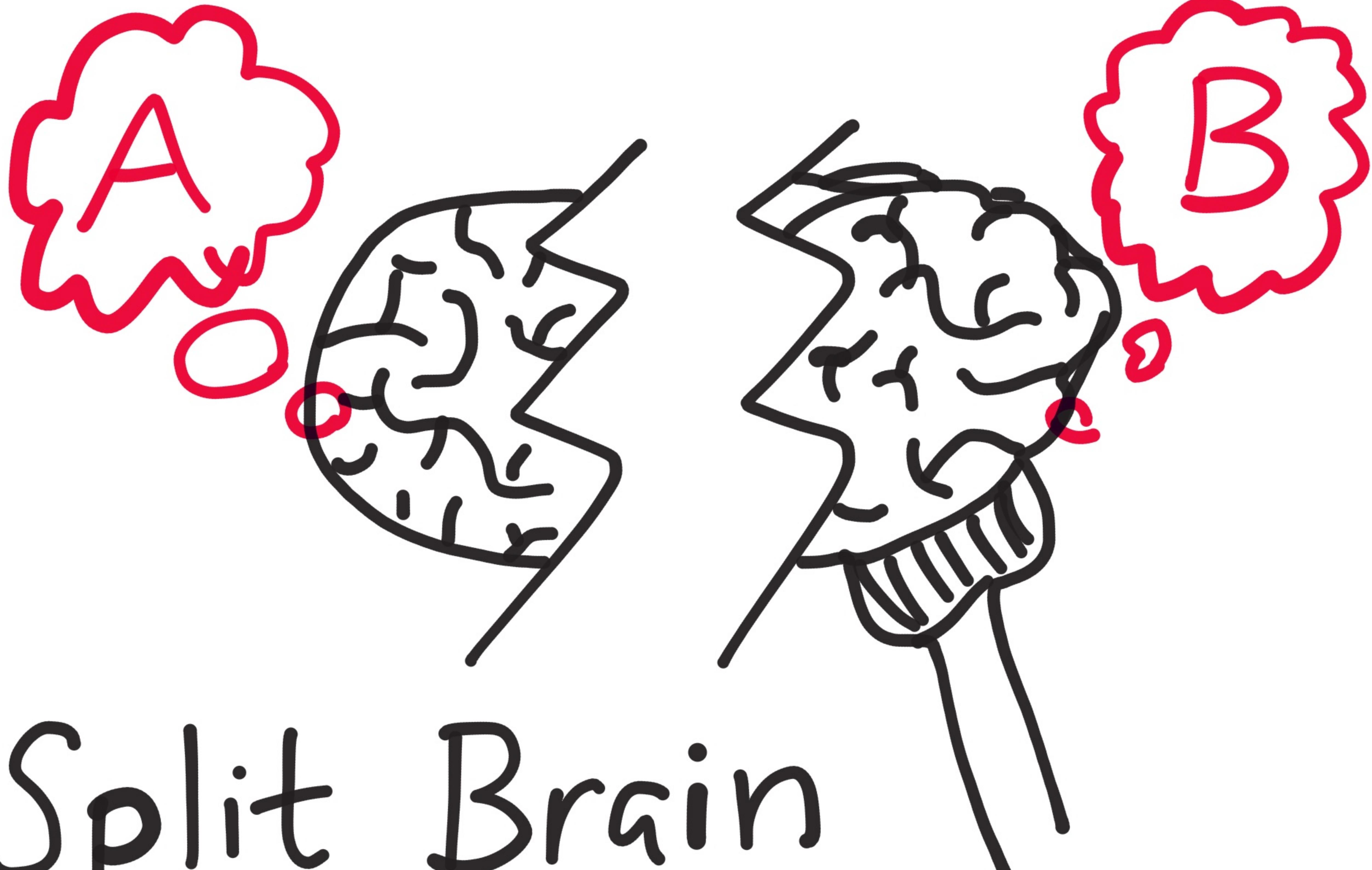
Every →
is →

← thing
fine !



← 00

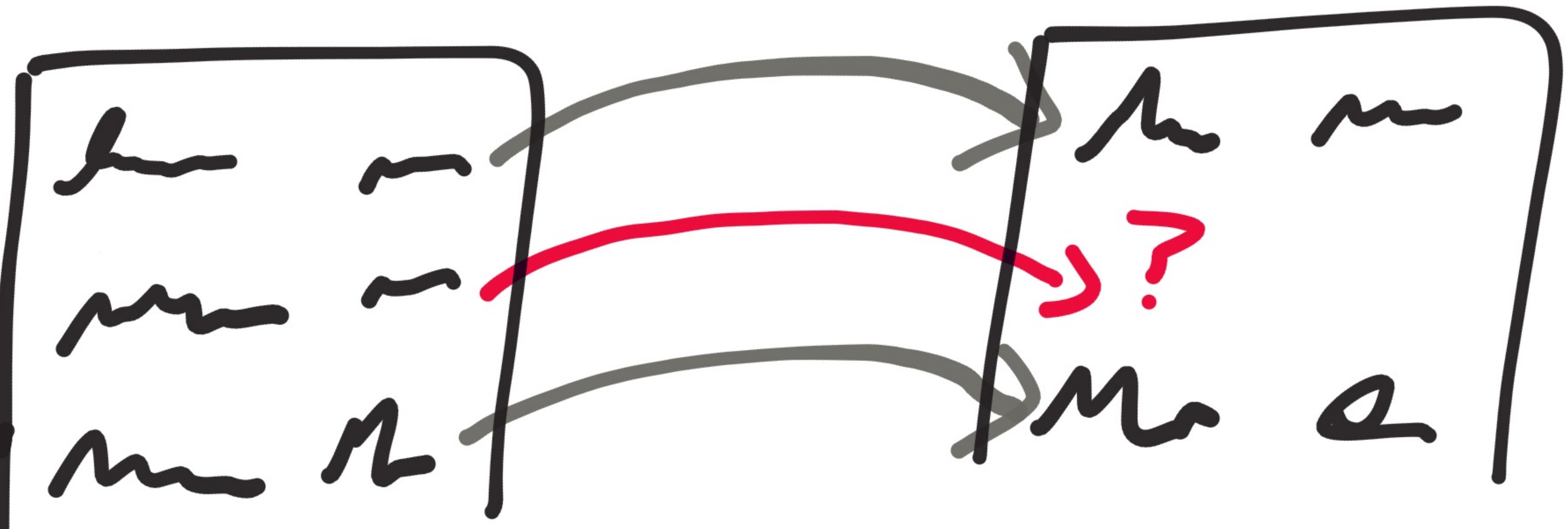
Databases! /
Queues! /
Discovery! /
THE HORROR



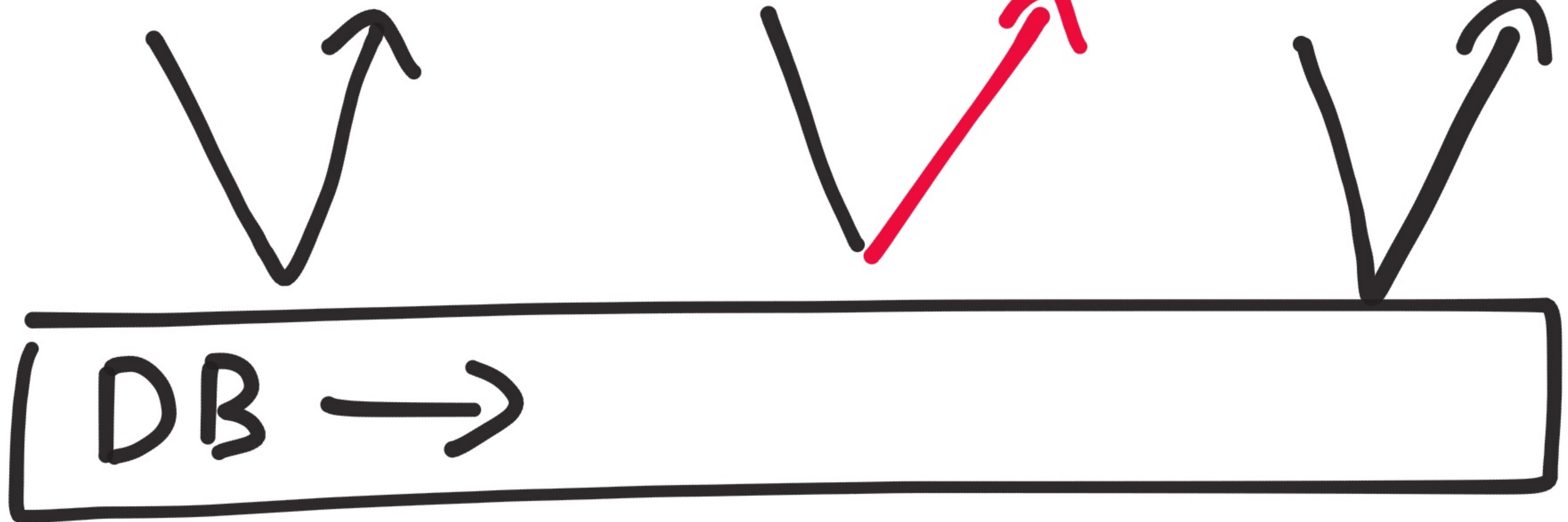
Split Brain

Broken

Foreign Keys



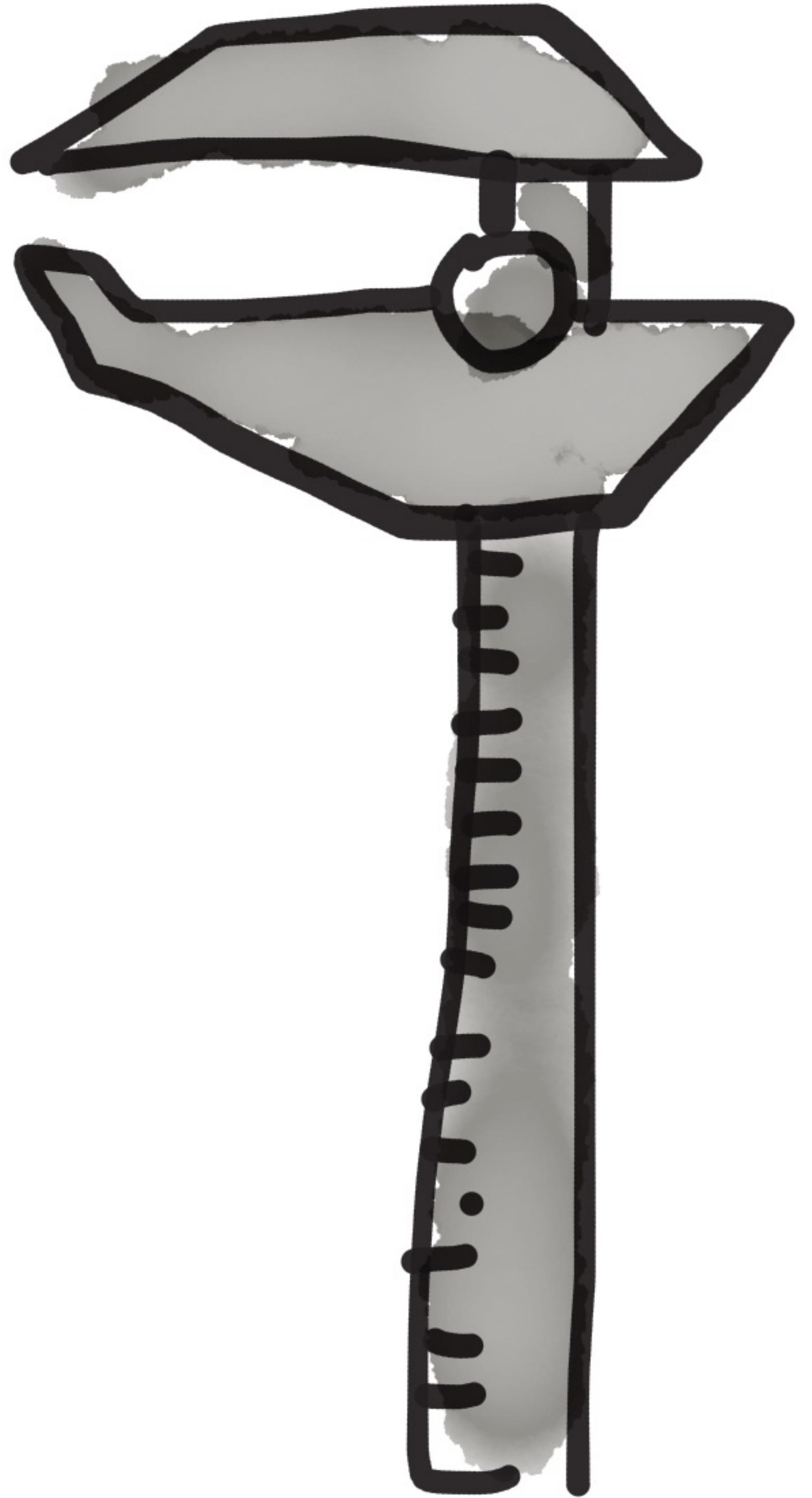
write ok read ↗ read ok



Anomalies

How do you
know if a
system is safe?

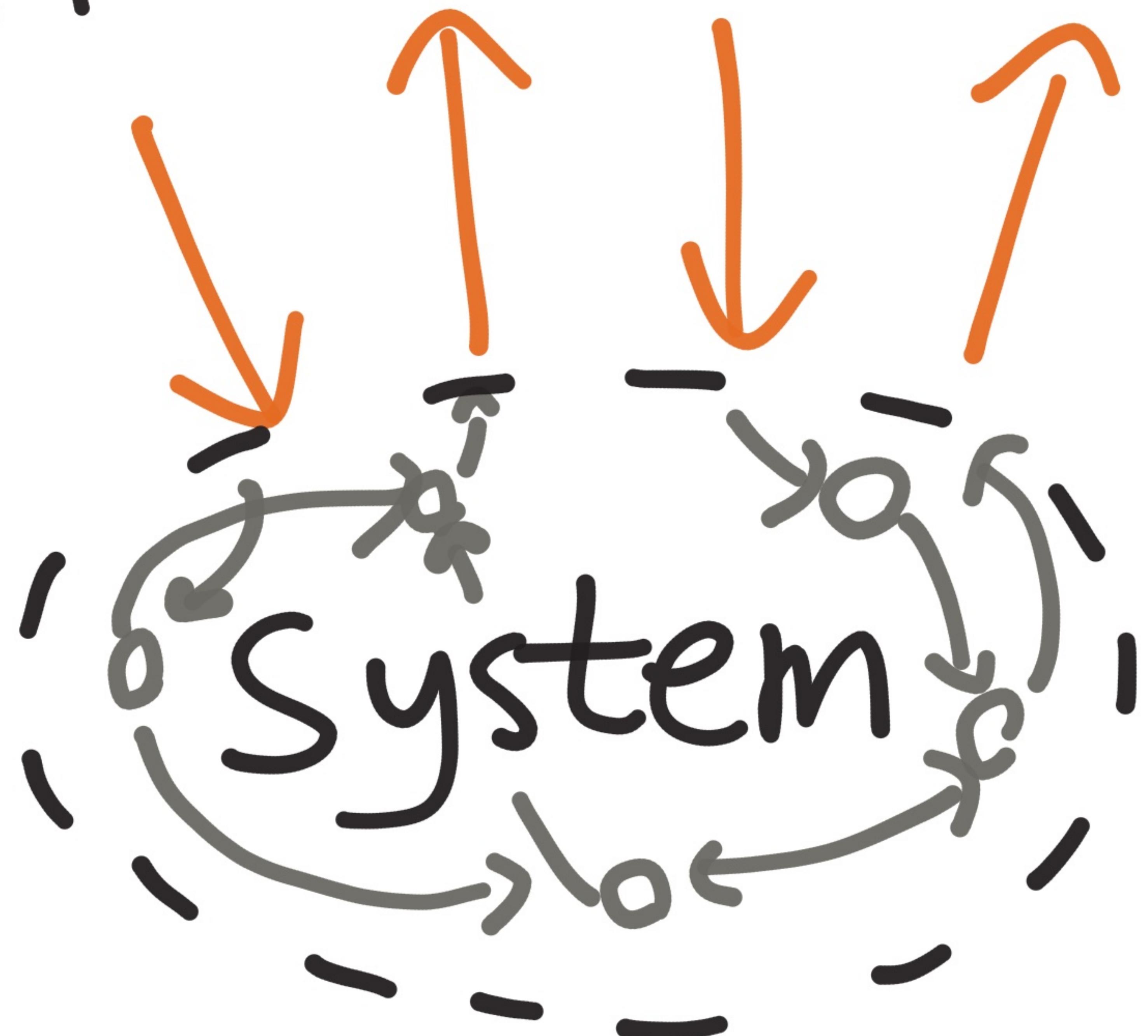
Measure
your
Systems



Jepsen

github.com/aphyr/jepsen

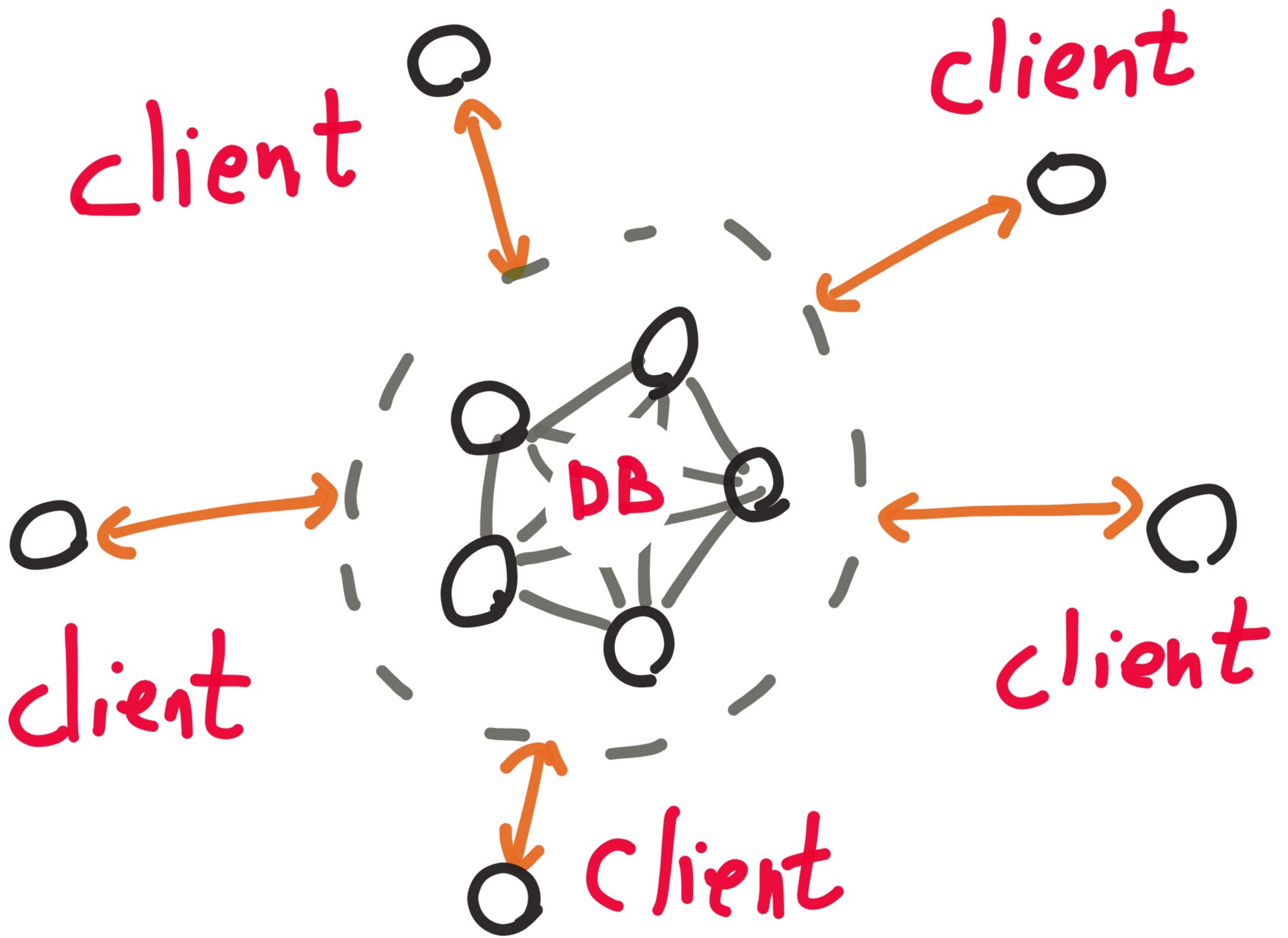
Environment





—INVARIANTS—





client: $w-w'$ $r-r'$ $w-w'$

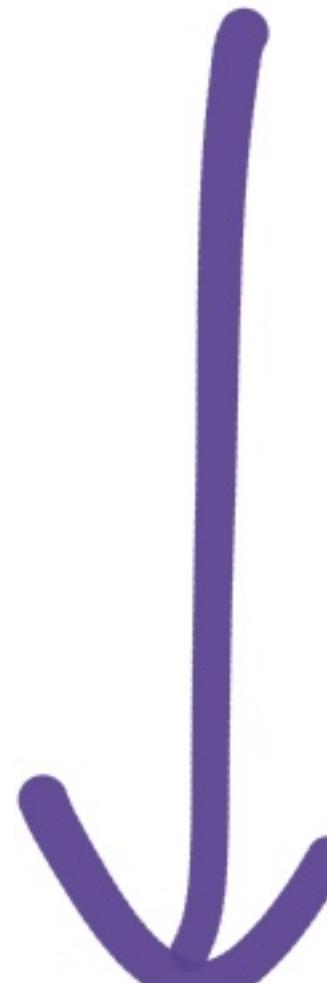
The diagram illustrates a client interacting with a central database (DB). Three arrows originate from the client and point to the DB. The first arrow is labeled $w-w'$, the second $r-r'$, and the third $w-w'$. The DB is represented by a central gray cloud-like shape.

DB:

client: w — w' w — \dots

The diagram shows a client sending a request w to a central DB. The DB returns multiple responses, indicated by three arrows pointing from the DB to the client, labeled w' , w , and \dots respectively. The DB is represented by a central gray cloud-like shape.

Clients Generate
random operations (w)
and apply them
to the system (w')



invoke

ok

invoke

fail

invoke

?

?

info

?

?

?

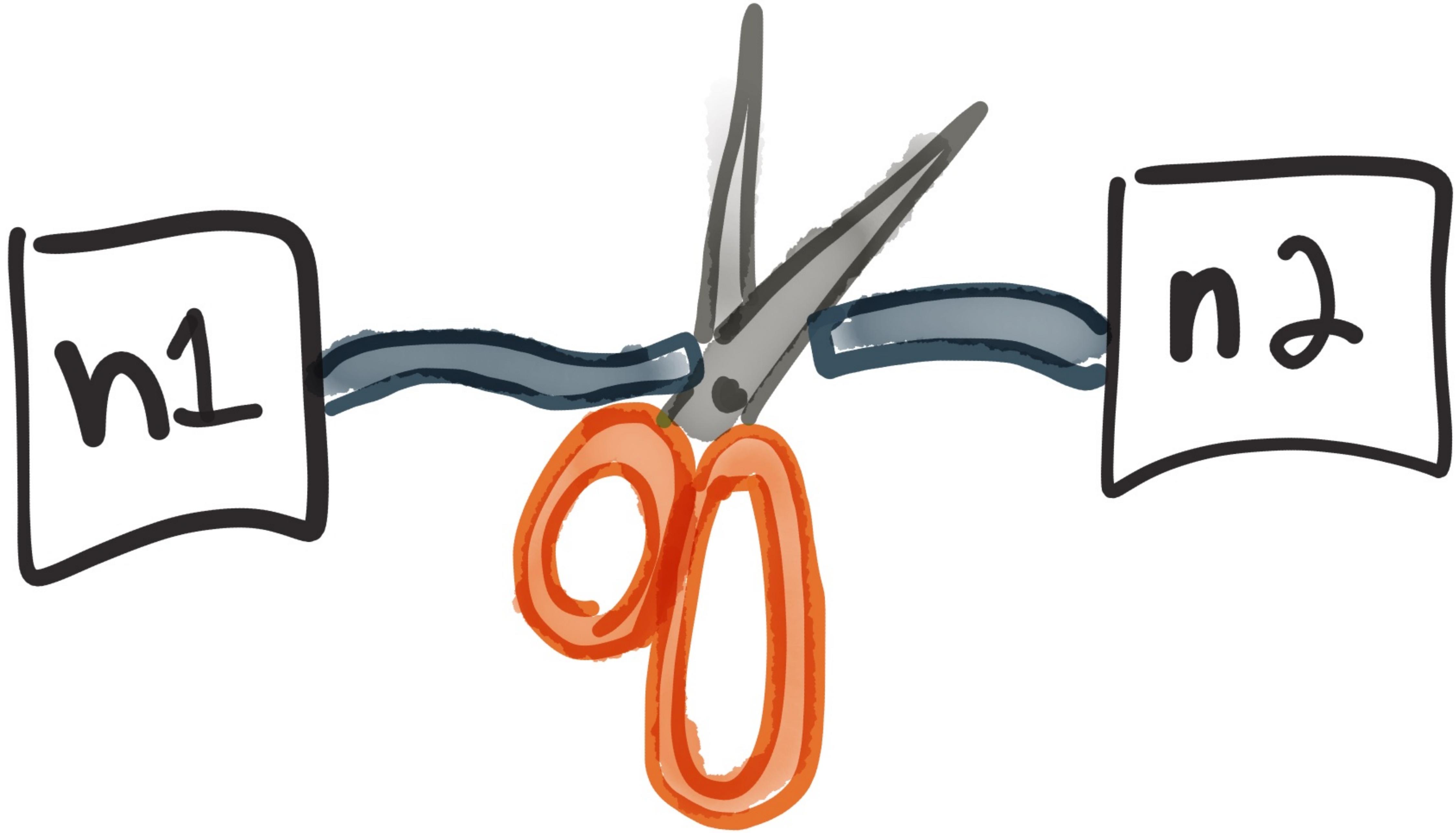
invoke ok

invoke fail

invoke ok

invoke info

1. Generate ops
2. Record history
3. Verify history is
consistent w/ model



Partitions!

So, what
have you
found?

Riak

LWW \rightarrow lost writes

CRDTs \rightarrow safe

5/13

Mongo

Data loss at all
Write Concerns

Redis Sentinel

5/13

Split brain,
massive write loss

Cassandra

9/13

- LWW write loss
- Row isolation broken
- Transaction deadlock
data less

NuoDB

Beat CAP by buffering all requests in IRAM during partition

Kafka

In-sync Replica Set

Could shrink to 0

nodes, causing msg loss.

9/13

Zookeeper

Works.

etcd / Consul

6/14

Stale reads

Elastic search

Loses documents in
every class of partition
tested.

RabbitMQ

Split brain, massive
message less

Aerospike

Claims "ACID", was
really LWW.

Elasticsearch

1.5.0

5/15

Still loses data
in every test case

MongoDB 2.6.7

5/15

stale reads

dirty reads

Chronos

8/15

- Breaks forever after losing quorum

Percona XtraDB/Galera 9/15

- "Snapshots" weren't
- First-committer-wins
not preserved
- Read locks broken

RethinkDB

1/16

- Basic tests passed
- Reconfiguration Could
destroy cluster in
rare cases

- Stale reads
- Dirty reads
- Lost writes

Crate.io

2016

- Stale reads
- Dirty reads
- Lost/corrupt updates
- Lost inserts

CockroachDB

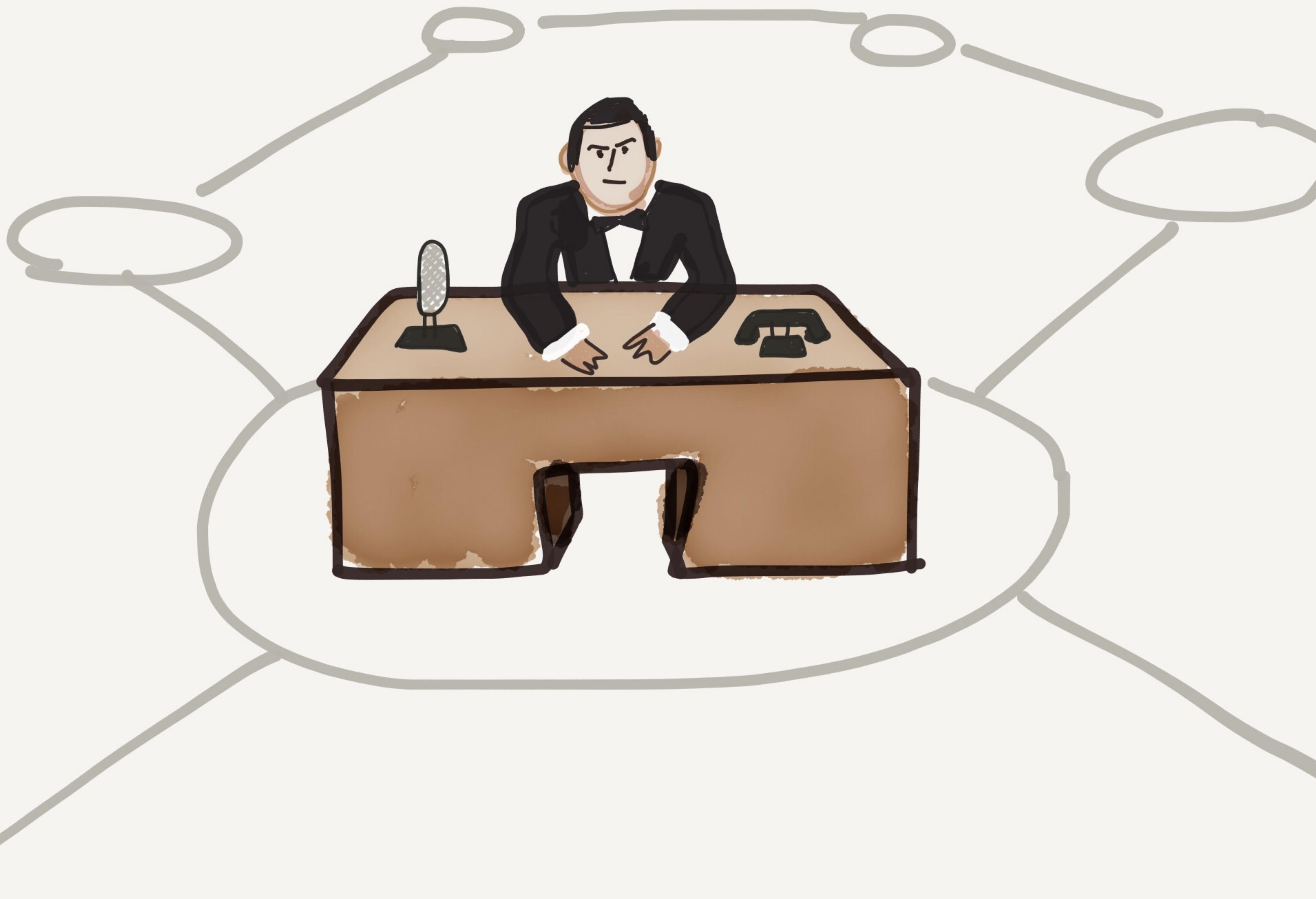
2017

- phantoms
- Double inserts

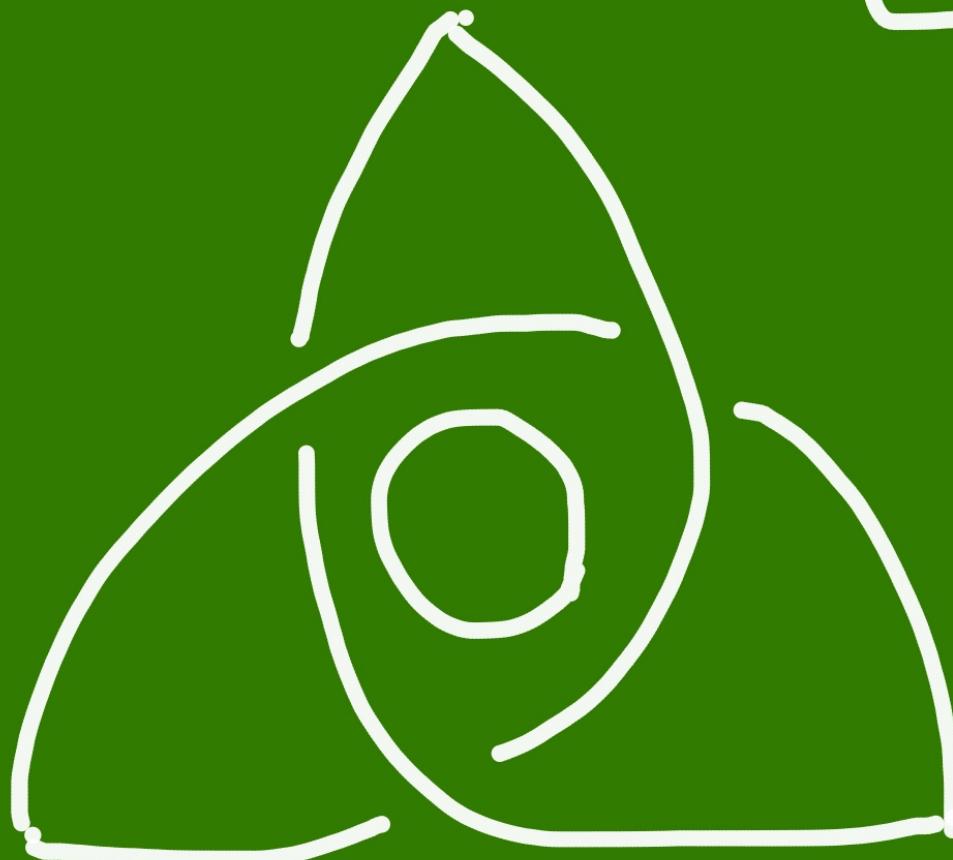
MongoDB

2017

- v0 protocol broken,
loses data
- Multiple data loss bugs
in v1 protocol



0.10.2



Tendermint

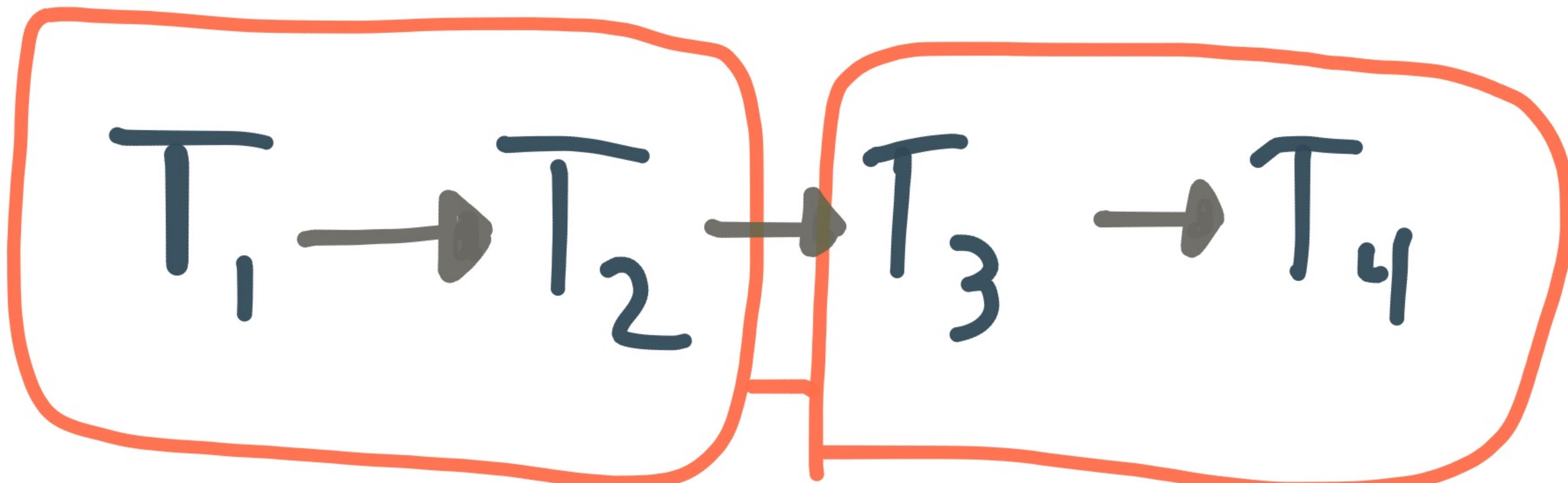
- Transaction replication
- Pluggable FSMs
- Byzantine fault tolerant

"Raft Meets
Blockchain"

Like e.g. Raft, forms a linearizable order of txns

$$T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4$$

Like Raft, provides a linearizable order of txns



blockchain!

Byzantine

Fault
Tolerant!

shun!



shun!



shun!



Pluggable

State

Machines

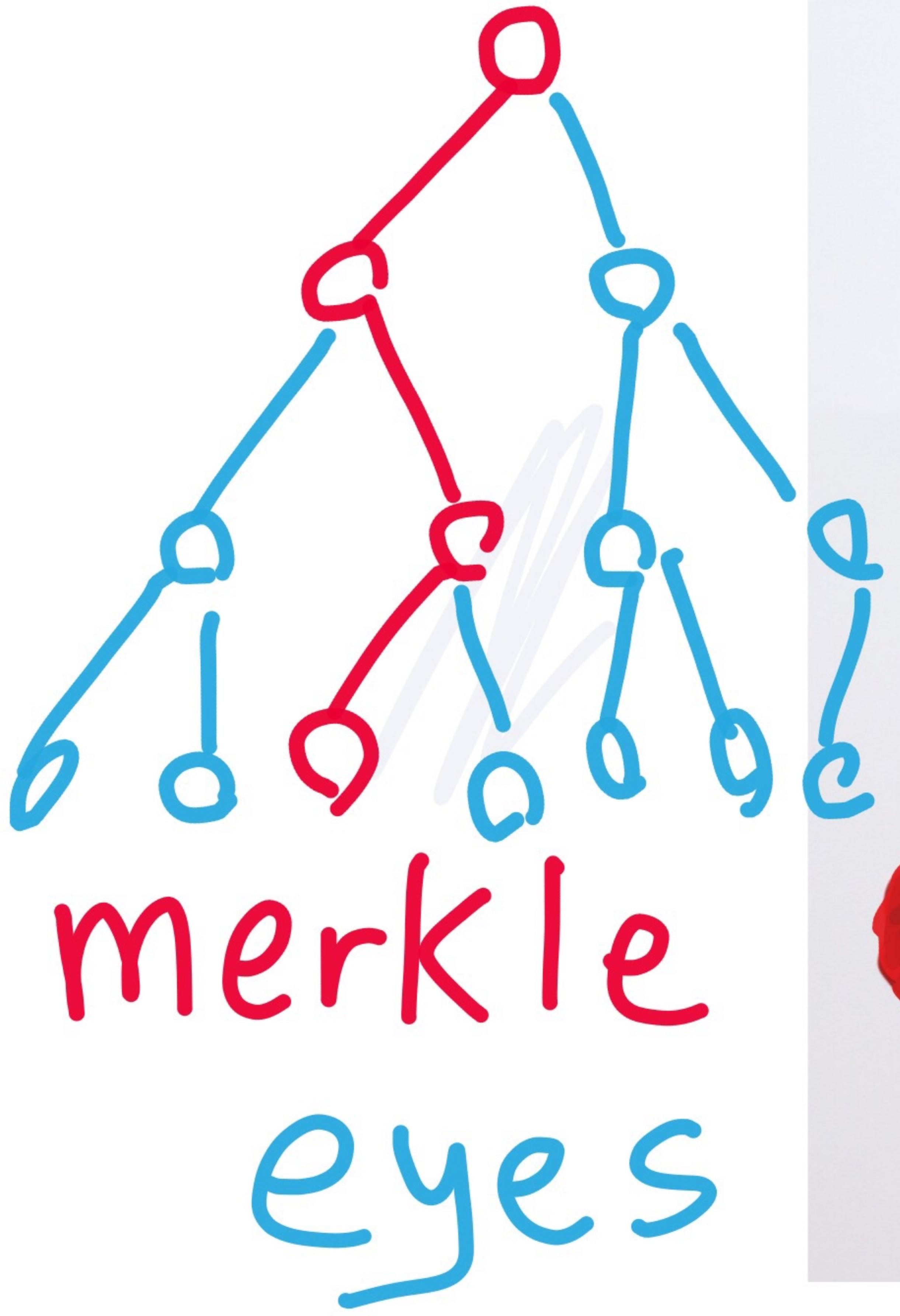
Validator

validator

State

State





CAS-register

- read(2)
- write(5)
- cas(1, 4)

Set

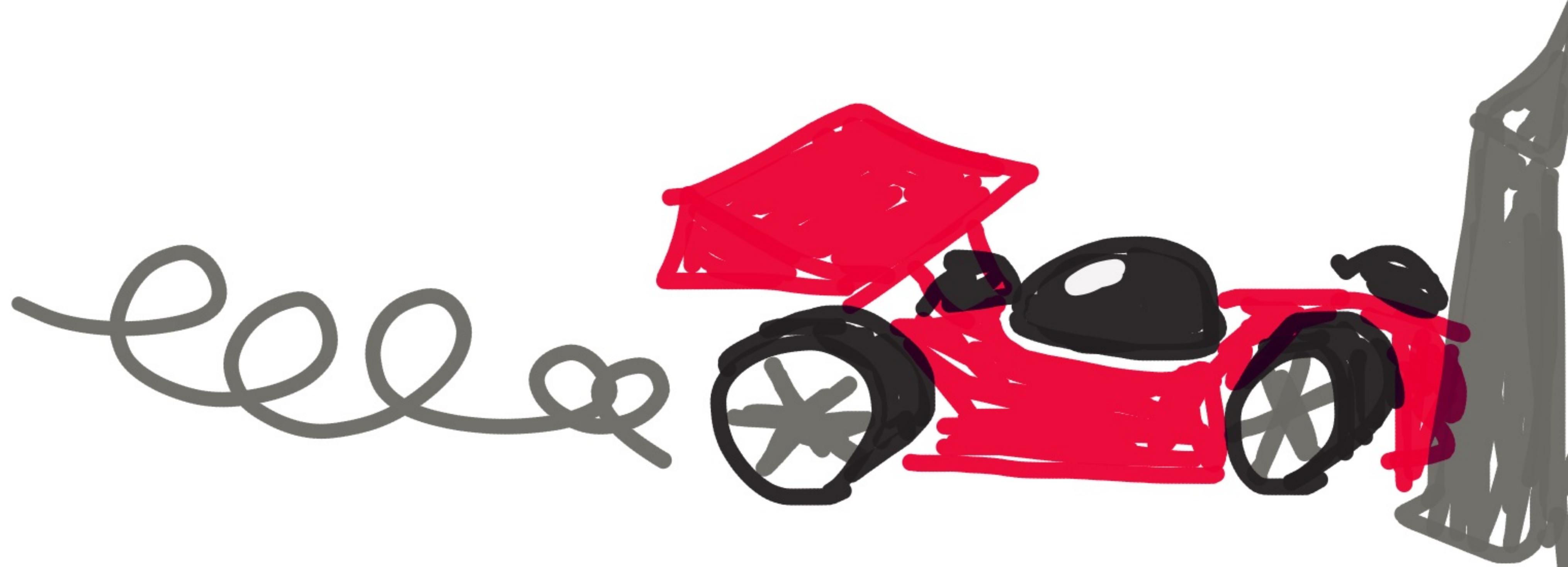
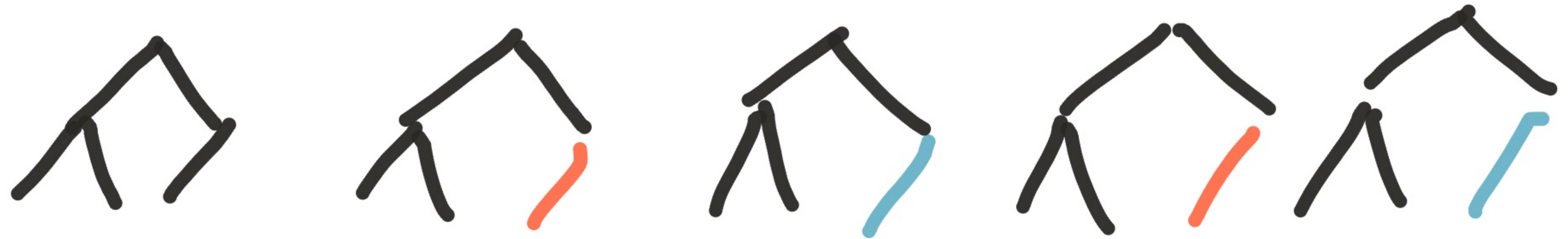
- add(6)

read({1,2,3})

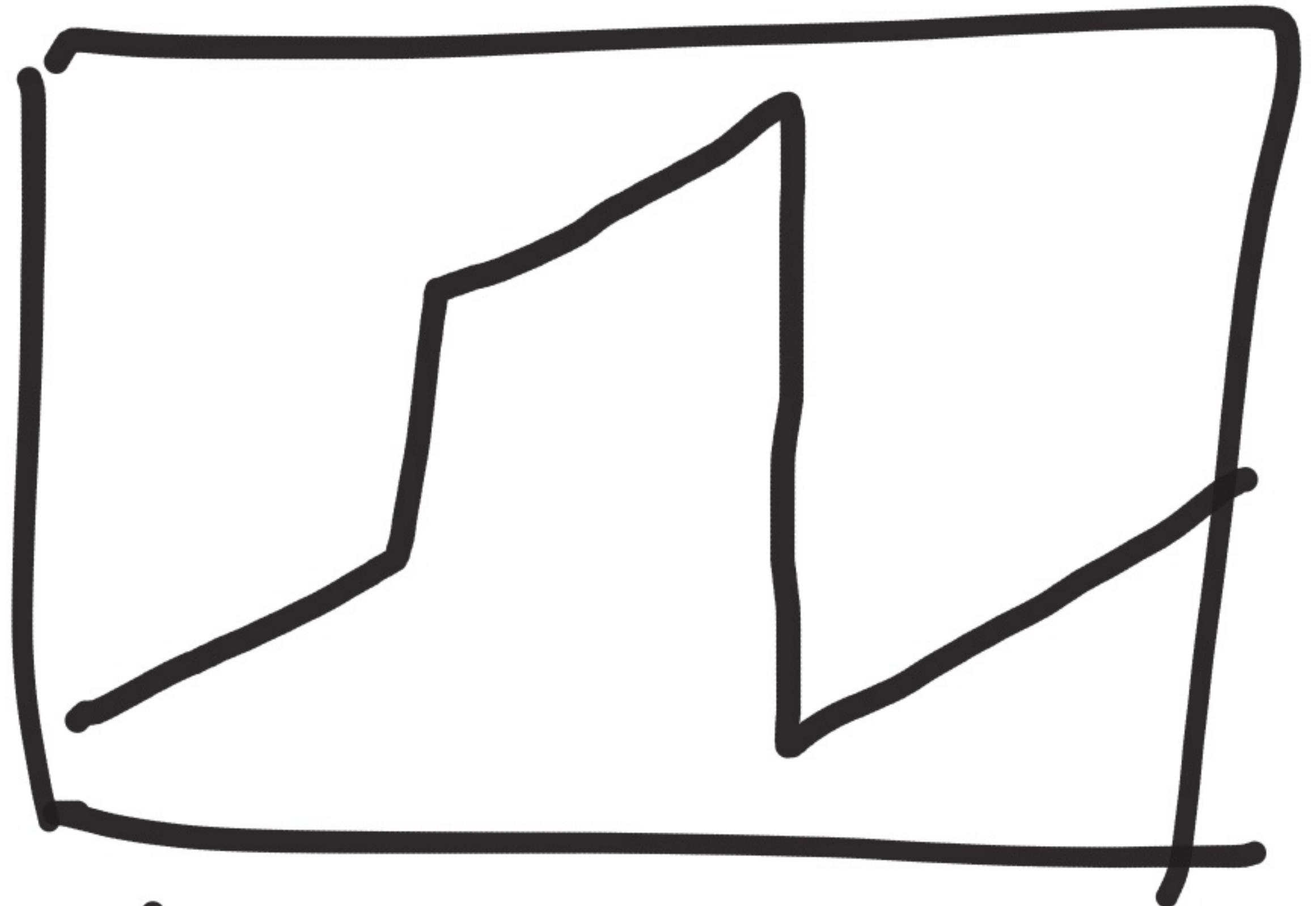
[cas({1,2,3}, {1,2,3,6})

- read({1,2,3,...})

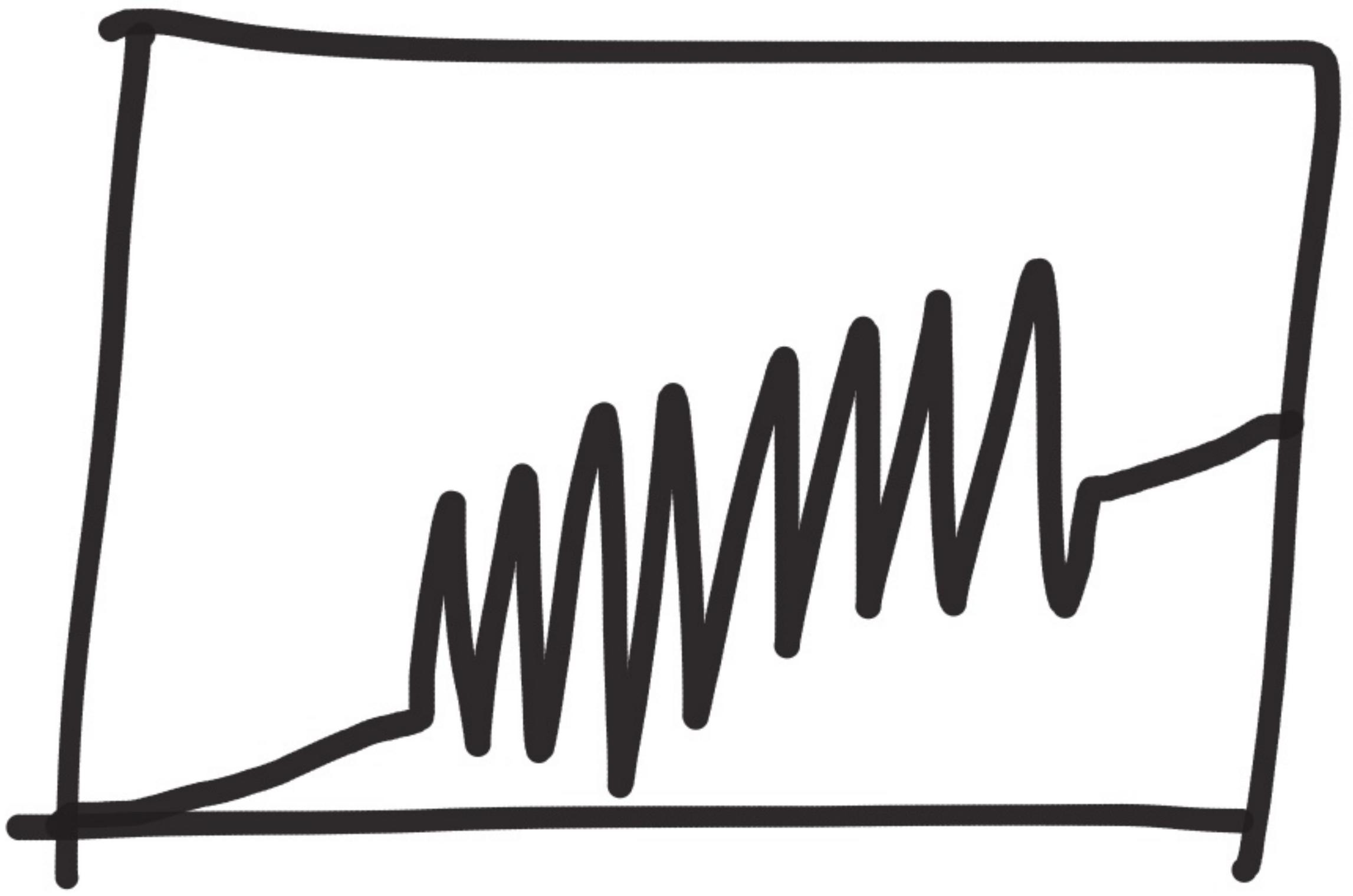
Merkleeyes Race



clock skew



bump



strobe

crashes

n_1 — 

n_2 — 

n_3 — 

n_4 — 

n_5 — 

—

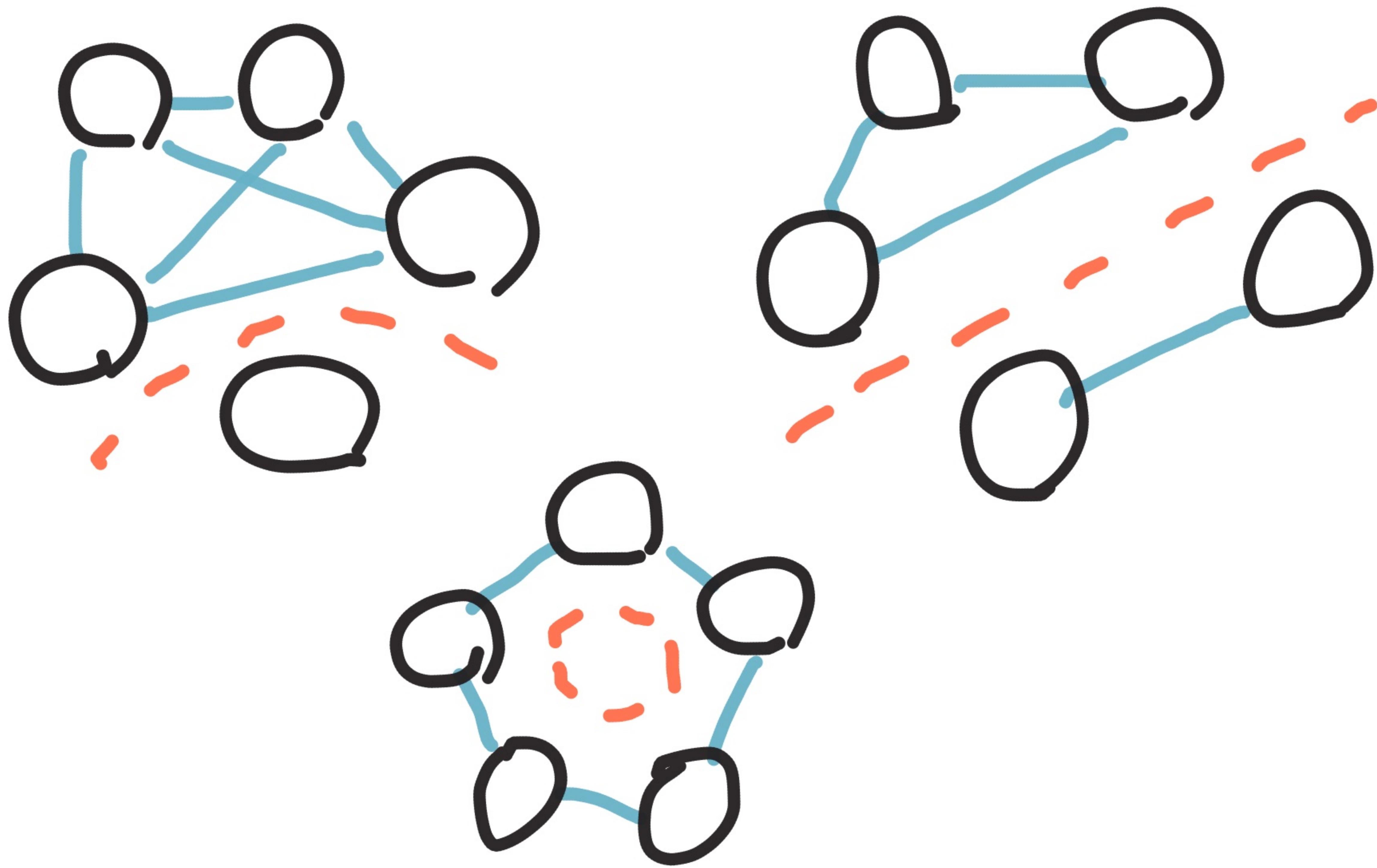
—

—

—

—

Partitions



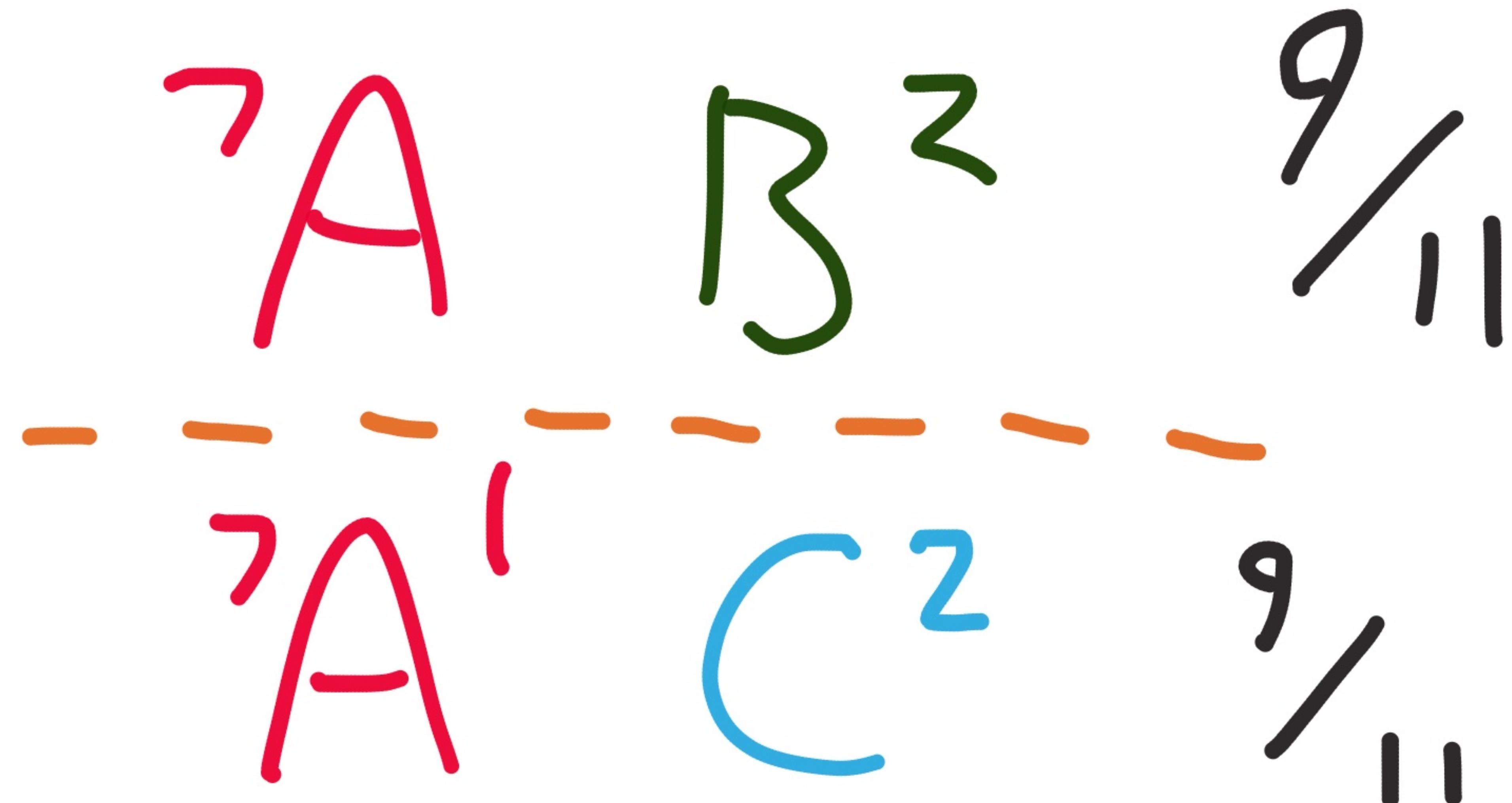
Duplicate Validators

'A' B²

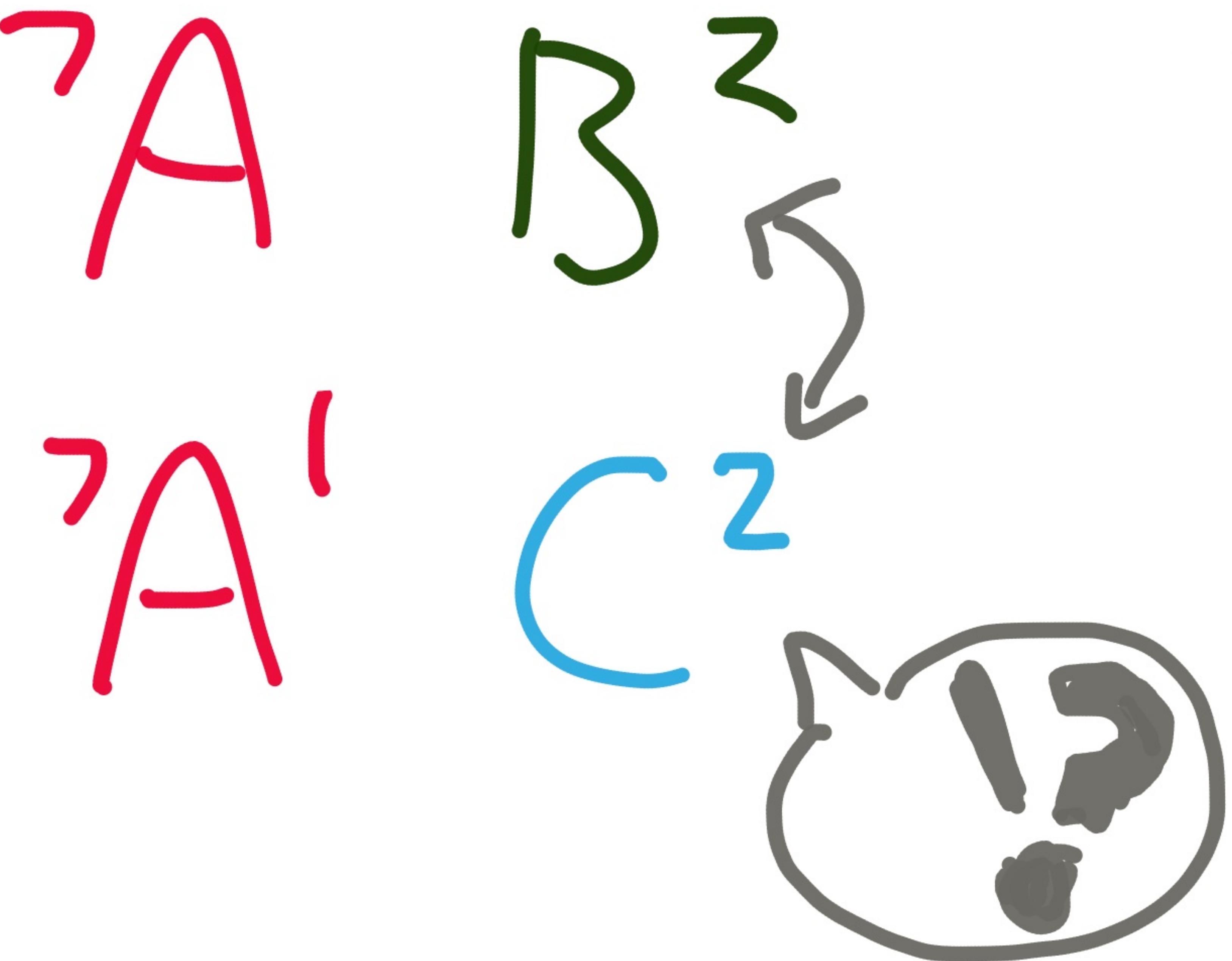
'A'¹ C²

$$7 + 2 + 2 = 11$$

Duplicate Validators



Duplicate Validators



lost

Documents



iff



> / 3

of votes

file truncation



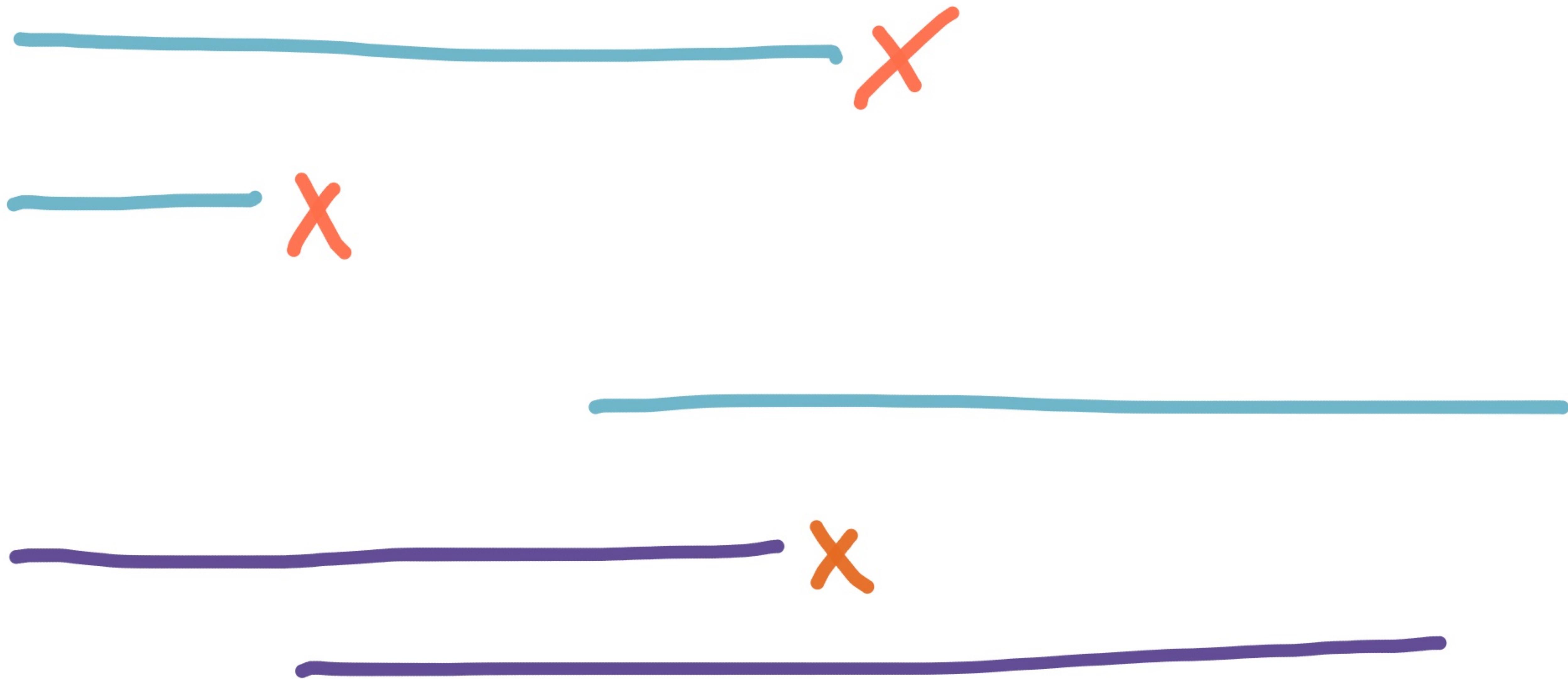
Merkleeyes crash!

(goleveldb bug)

Tendermint

Doesn't fsync to
disk!

Dynamic Reconfig



Data Corruption &

Crashes are to be
expected — Tendermint

is still in beta

Team working to

fix these issues





hazelcast

3.8.3

In-memory

DATA
GRID

Sets

Lists

IDs

Semaphores

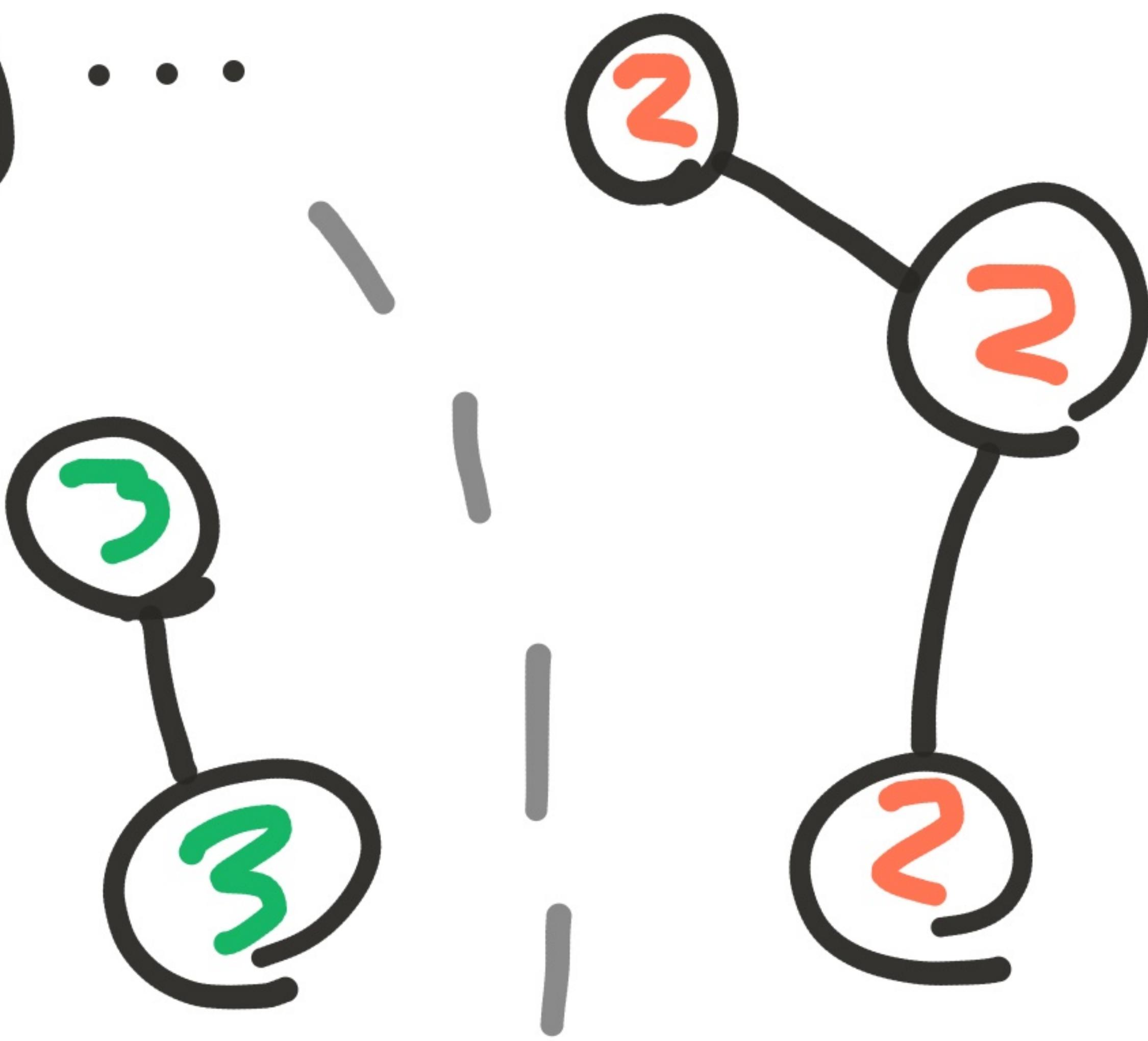
Queues

Maps

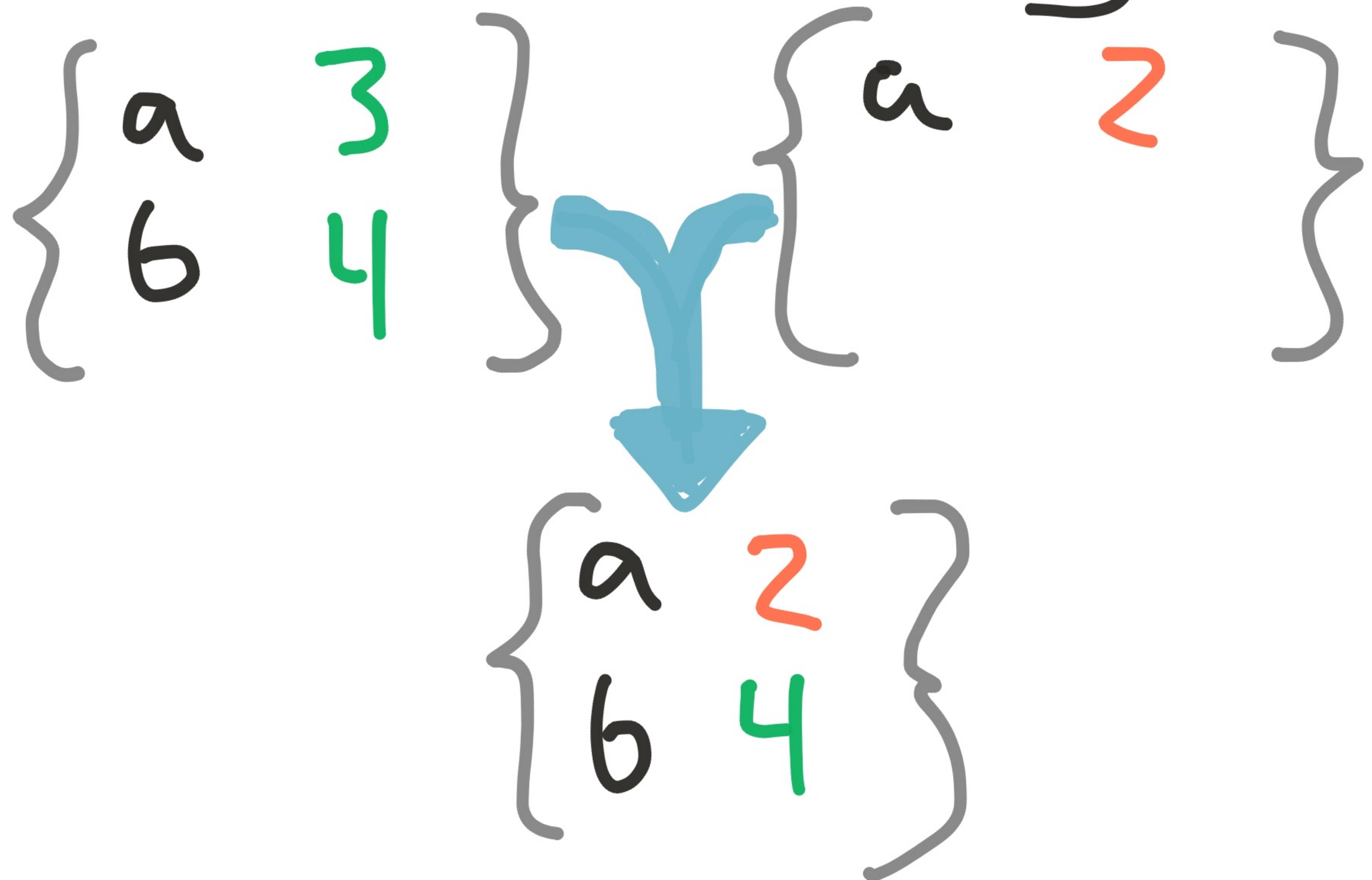
Locks

Atomic Longs

But in a partition,
Components run
independently ...

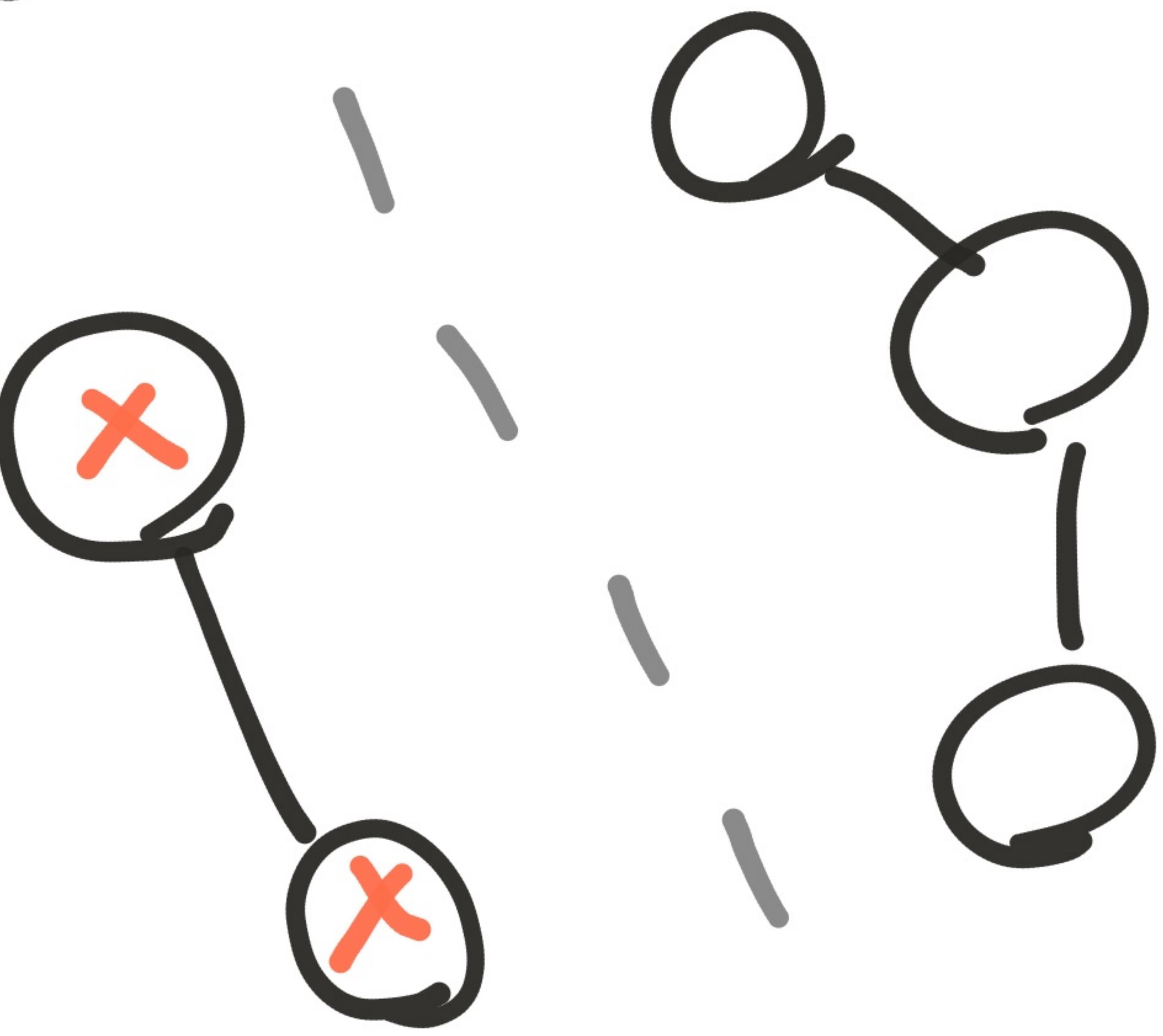


Maps have merge



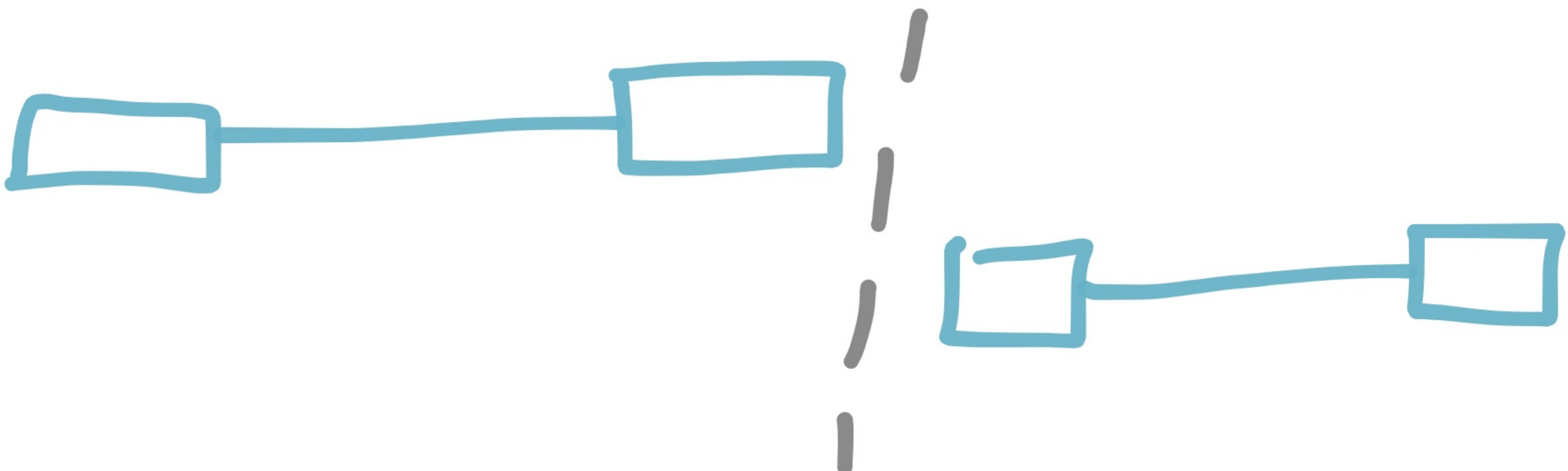
Split-Brain Protection

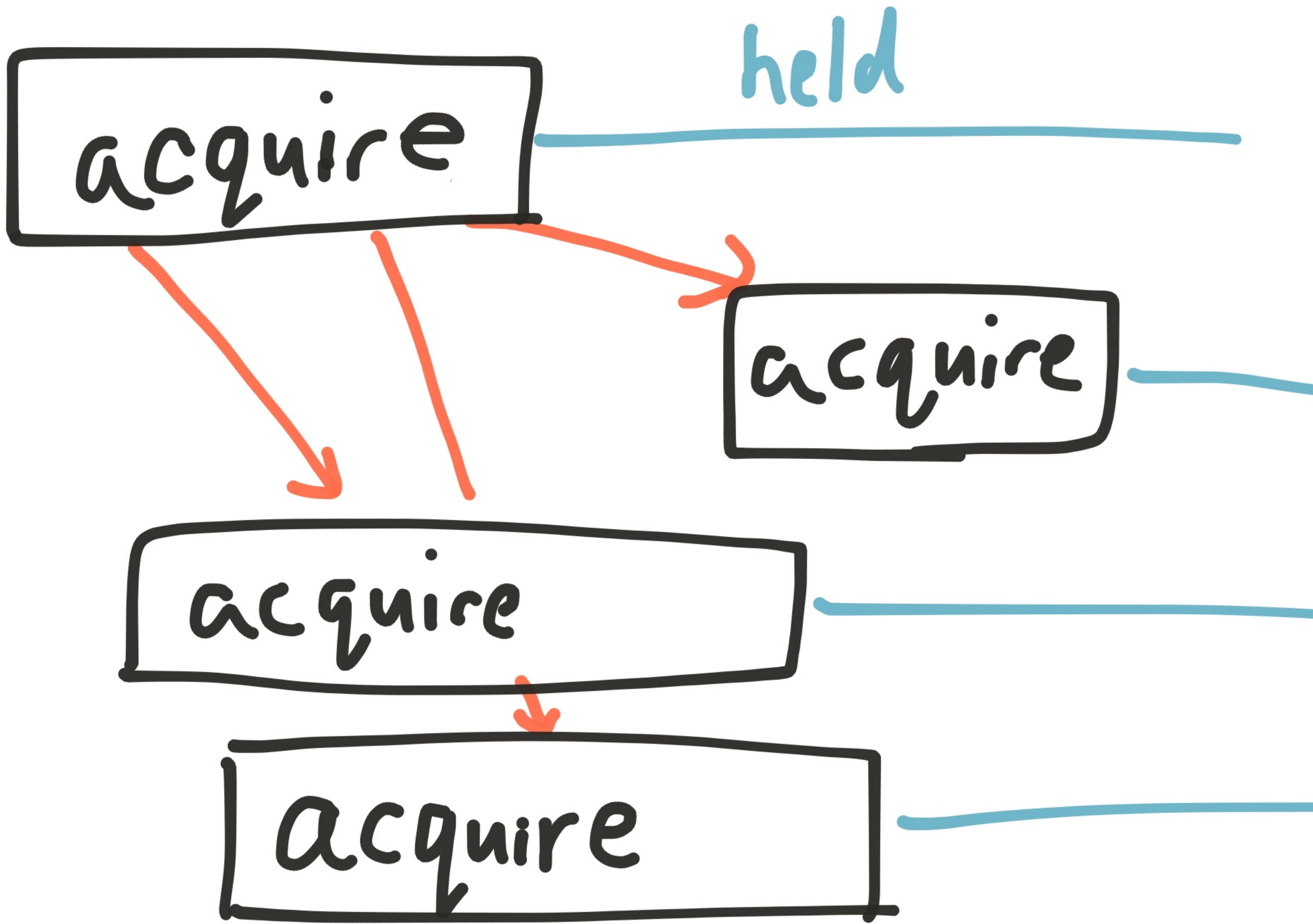
(10^+ seconds)



Locks

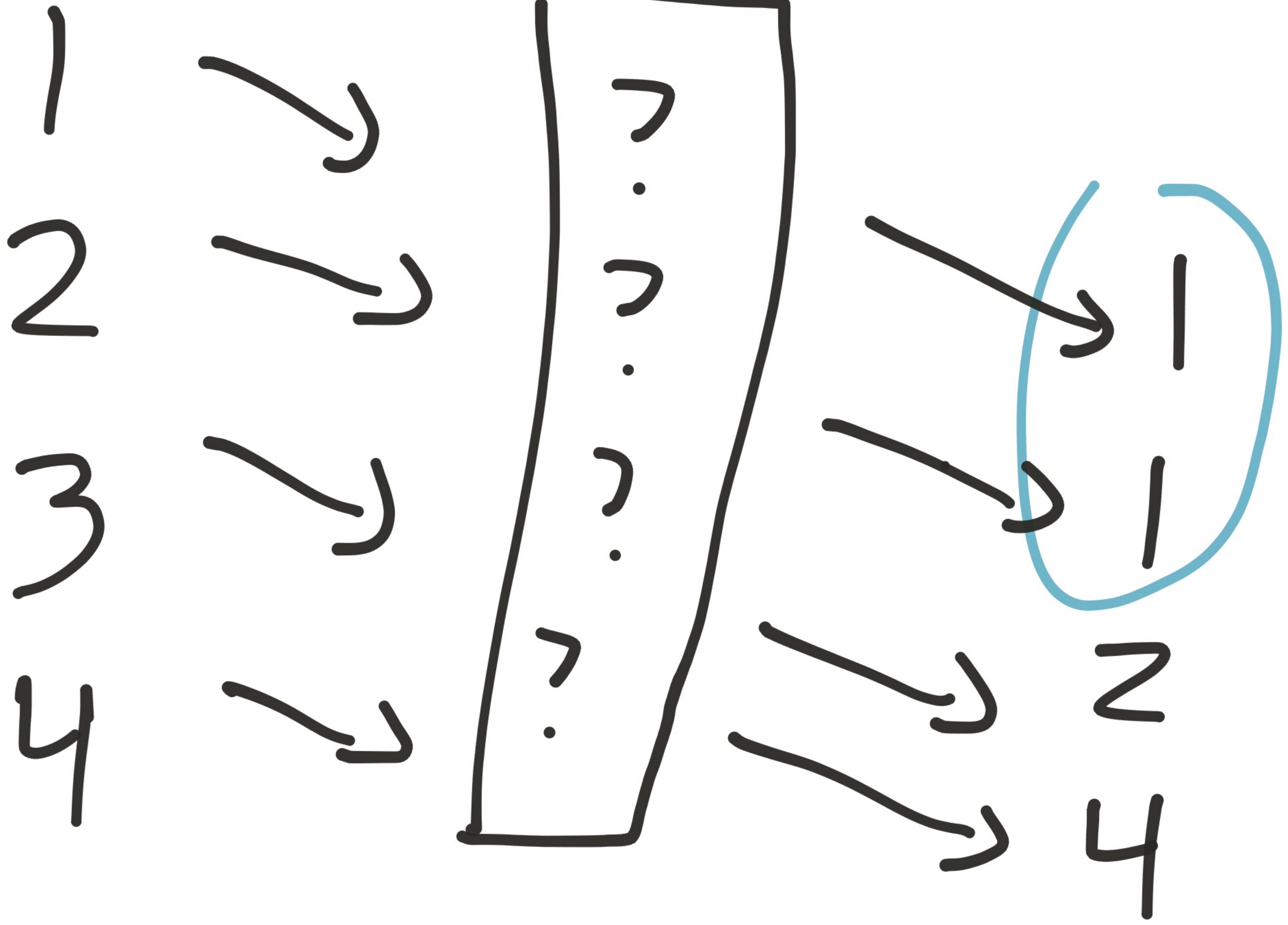
"Guaranteed to be executed by only one thread in the cluster"





Queues

"...you can add an item
in one machine and
remove it from
another one"



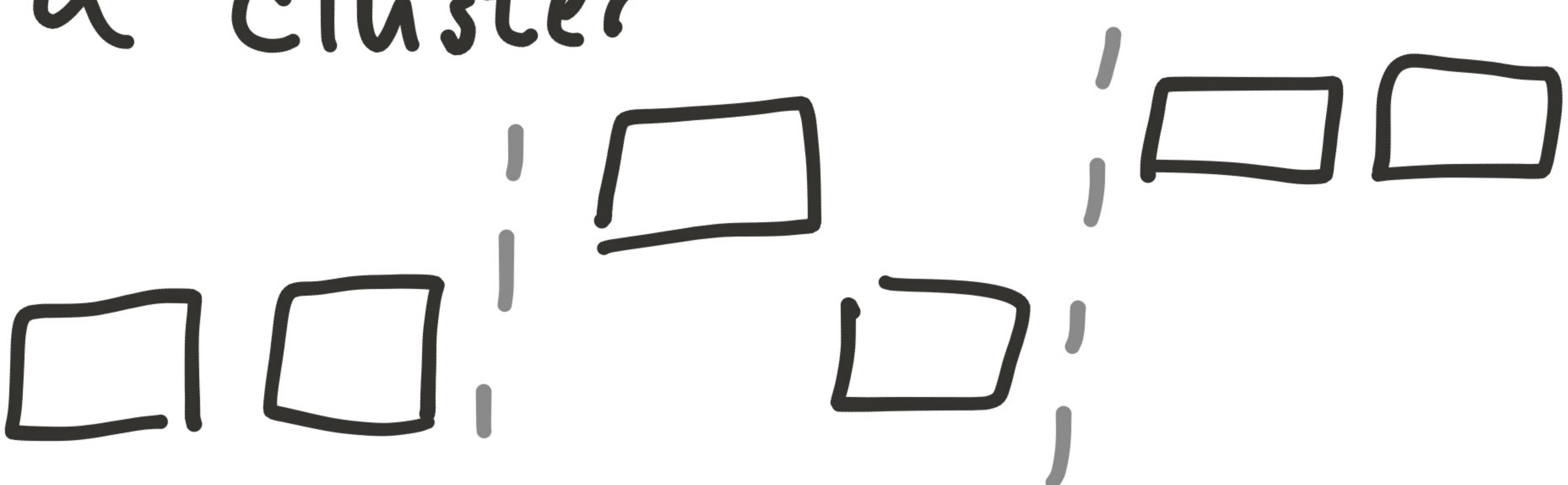
$\sim 2\%$ duplicated

$\sim 2\%$ lost

worse with default
timeouts

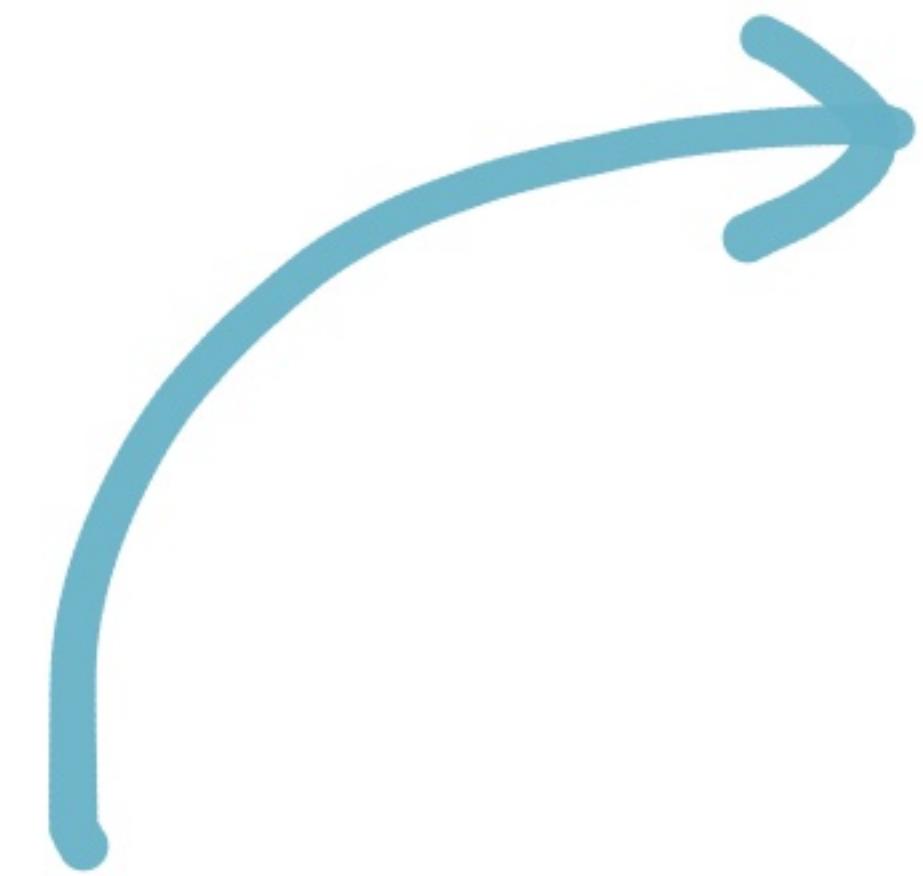
Atomic Reference

"Guaranteed atomic
Compare and set across
a cluster"



increment

$$2 + 1 = 3$$

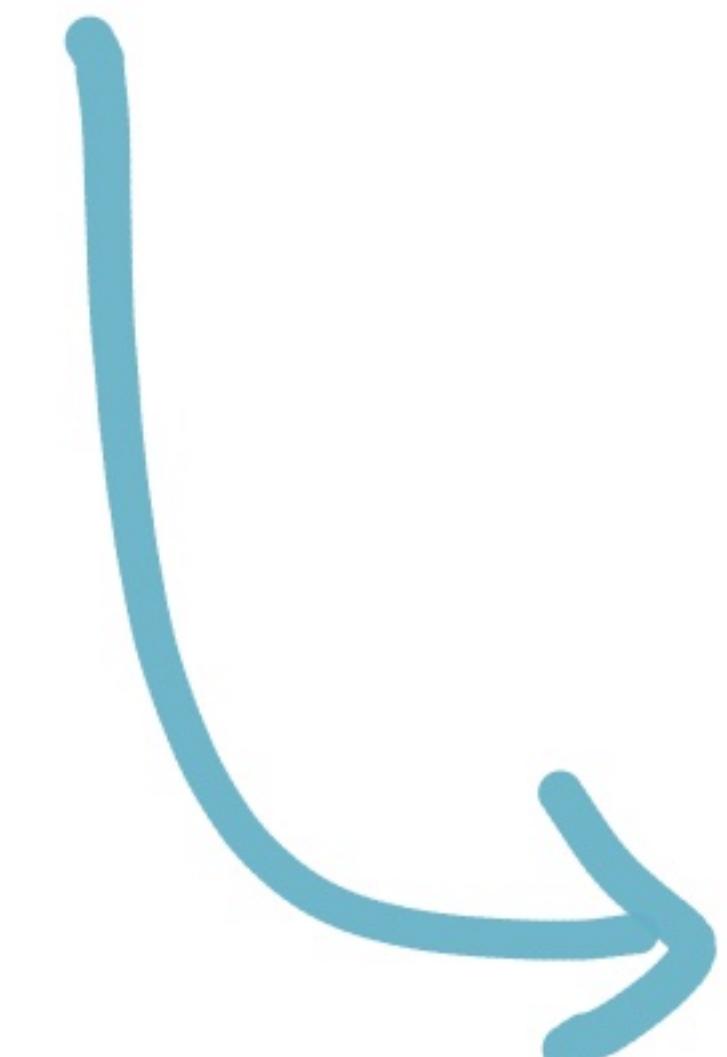


"still 2?"
"yes!"

-2

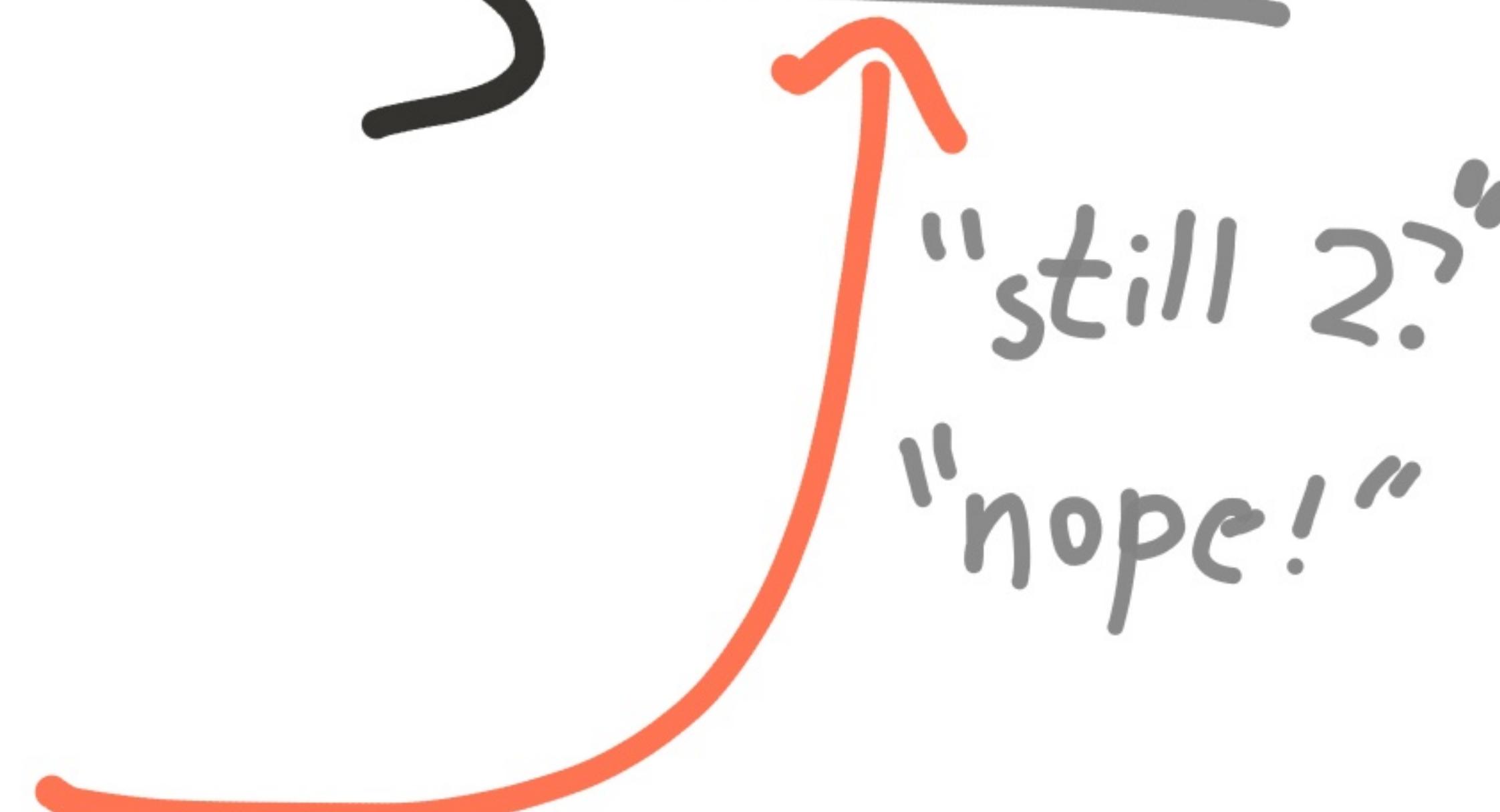


3



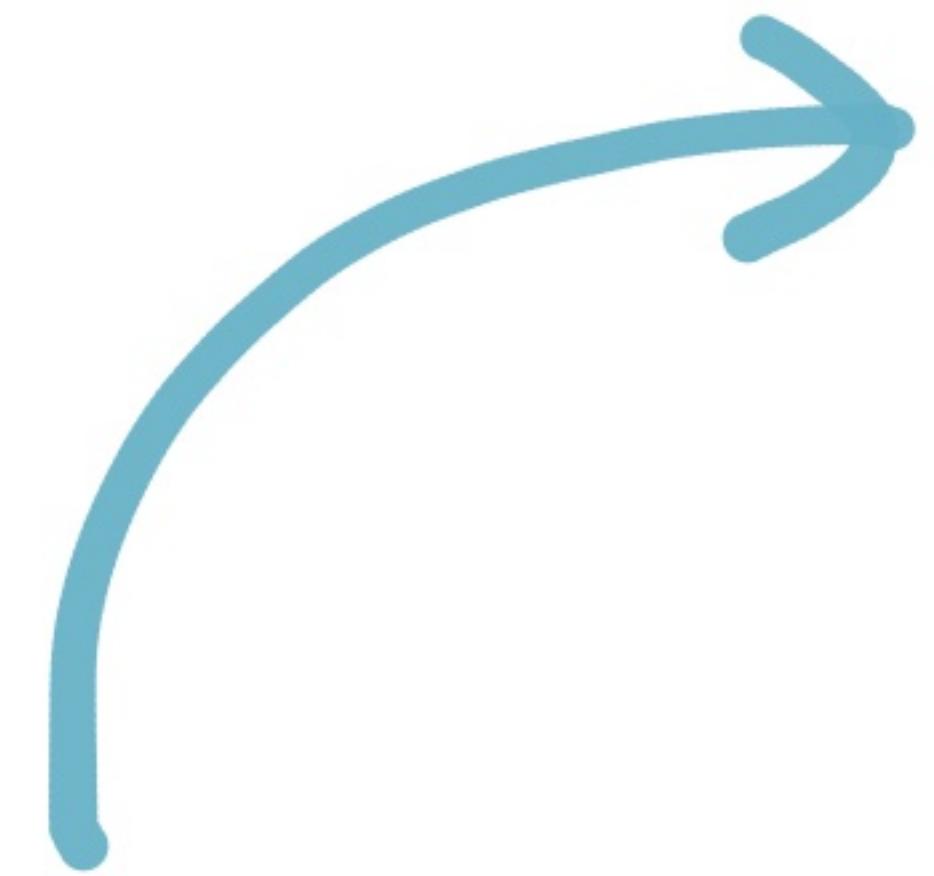
increment

$$2 + 1 = 3$$



increment

$$2 + 1 = 3$$



"still 2?"
"yes!"

$$-2$$

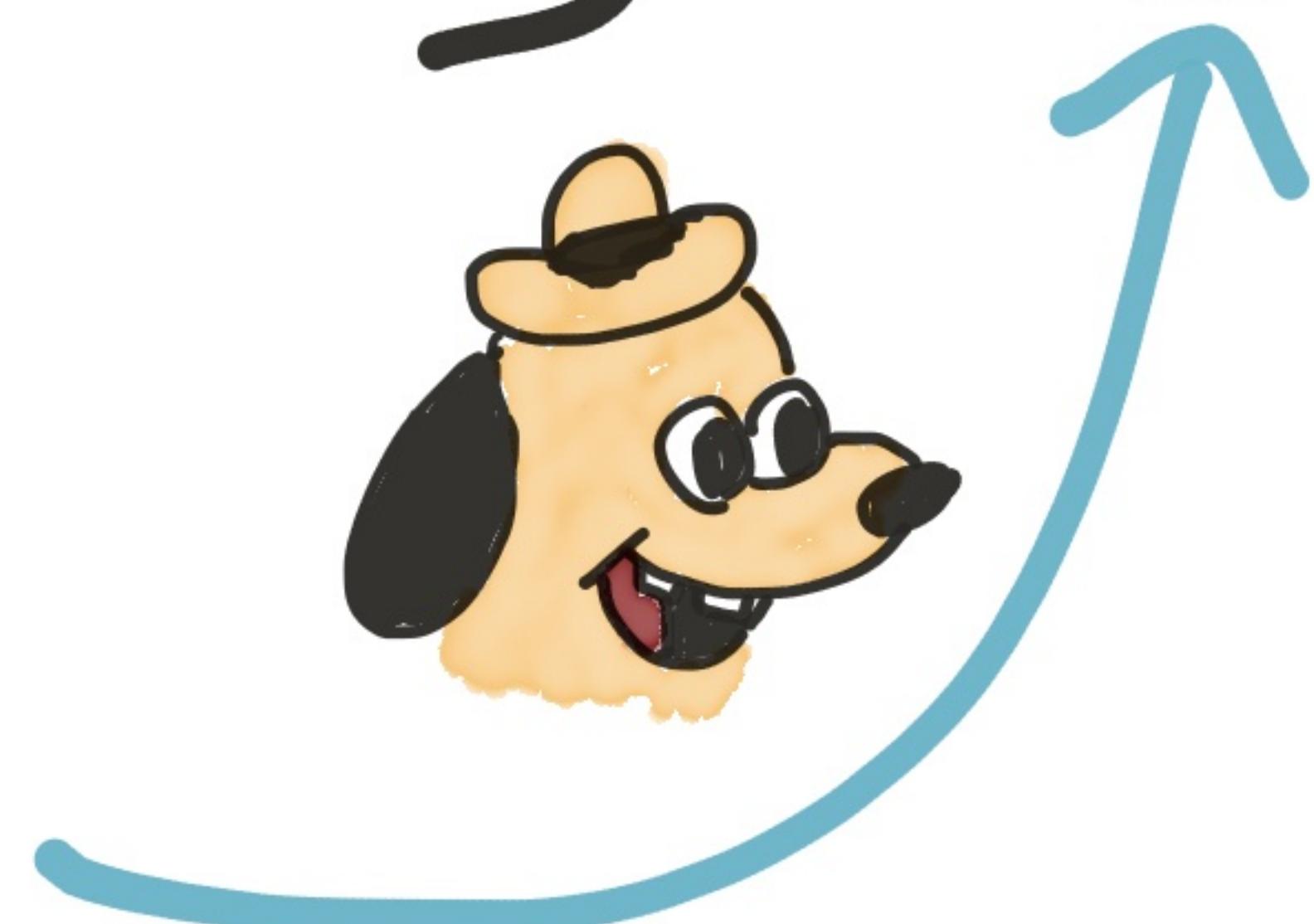
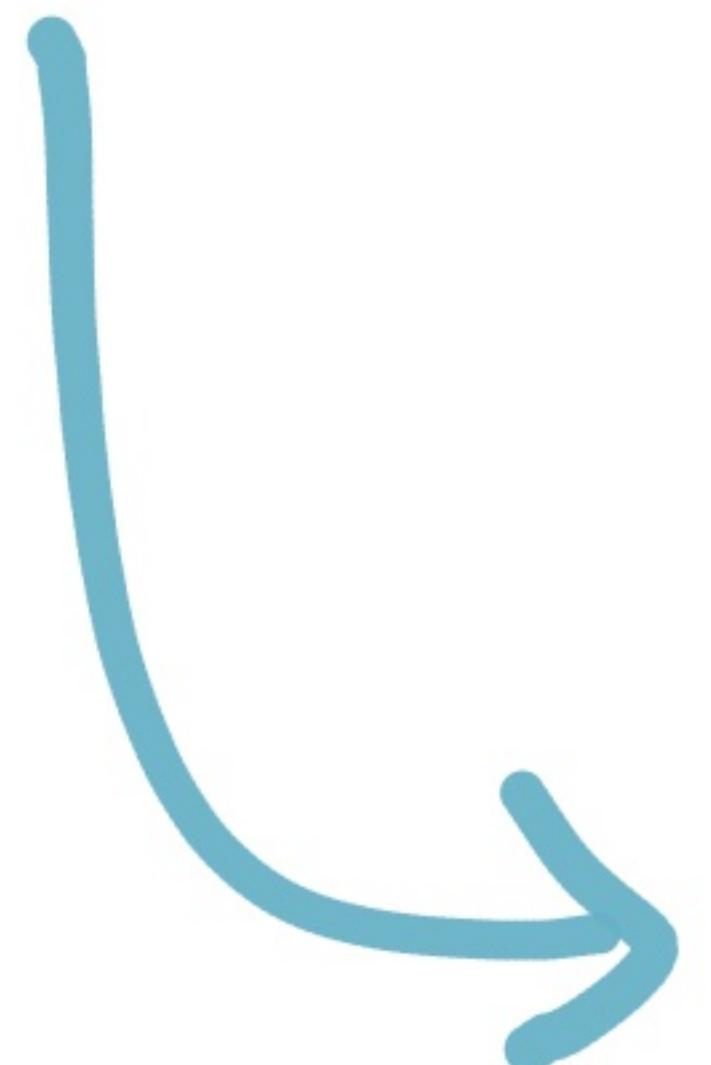


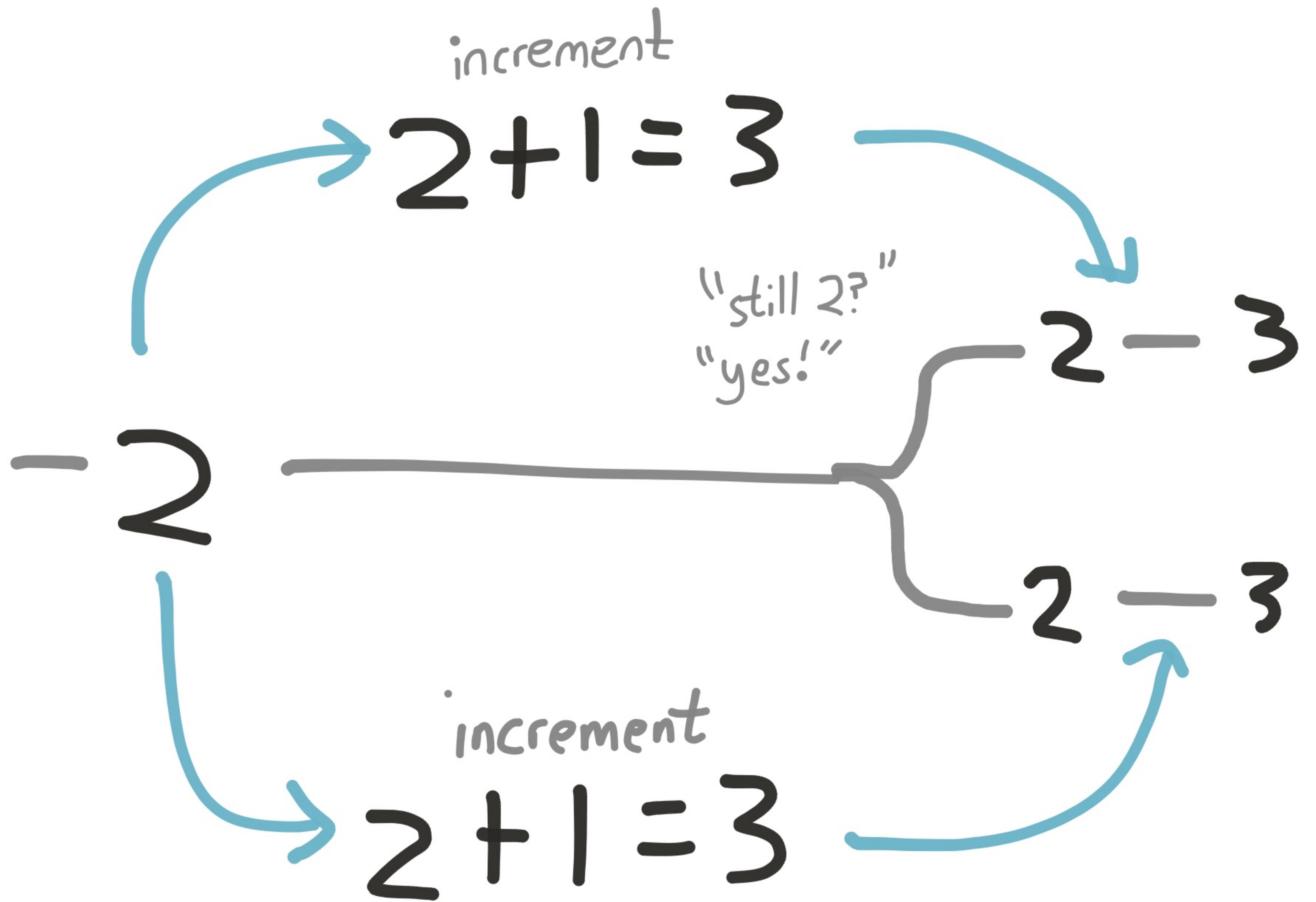
$$3 - 3$$

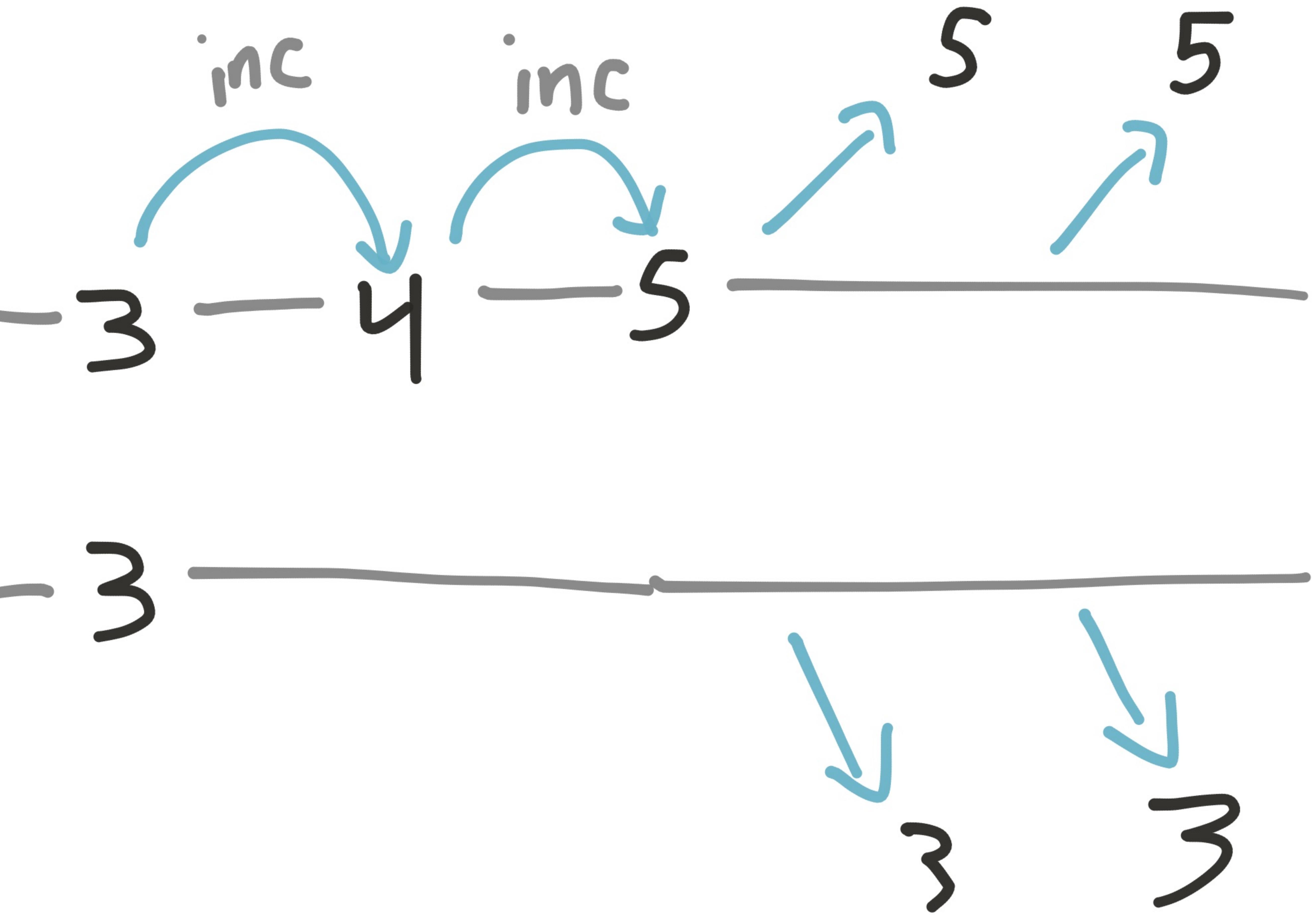


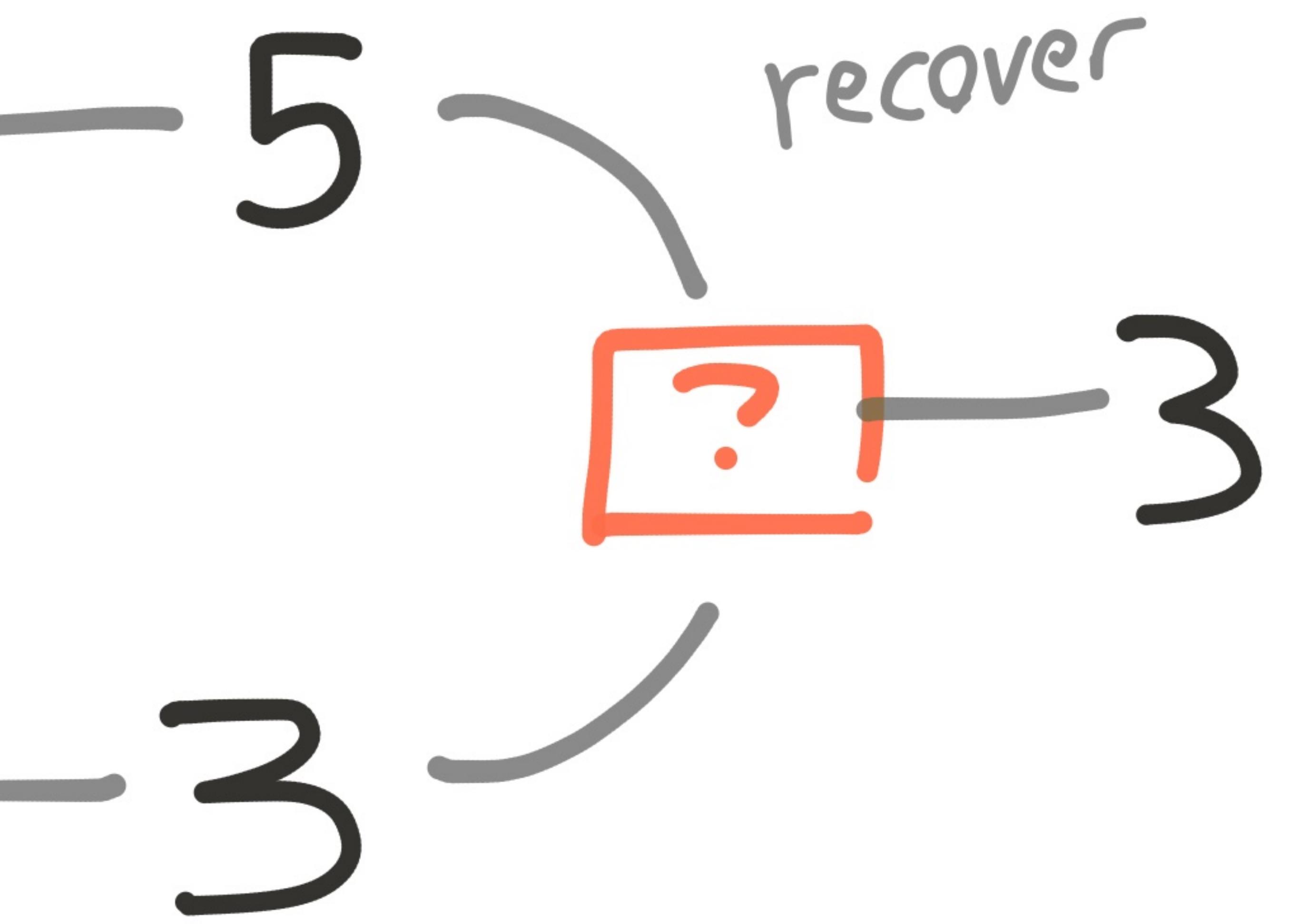
increment

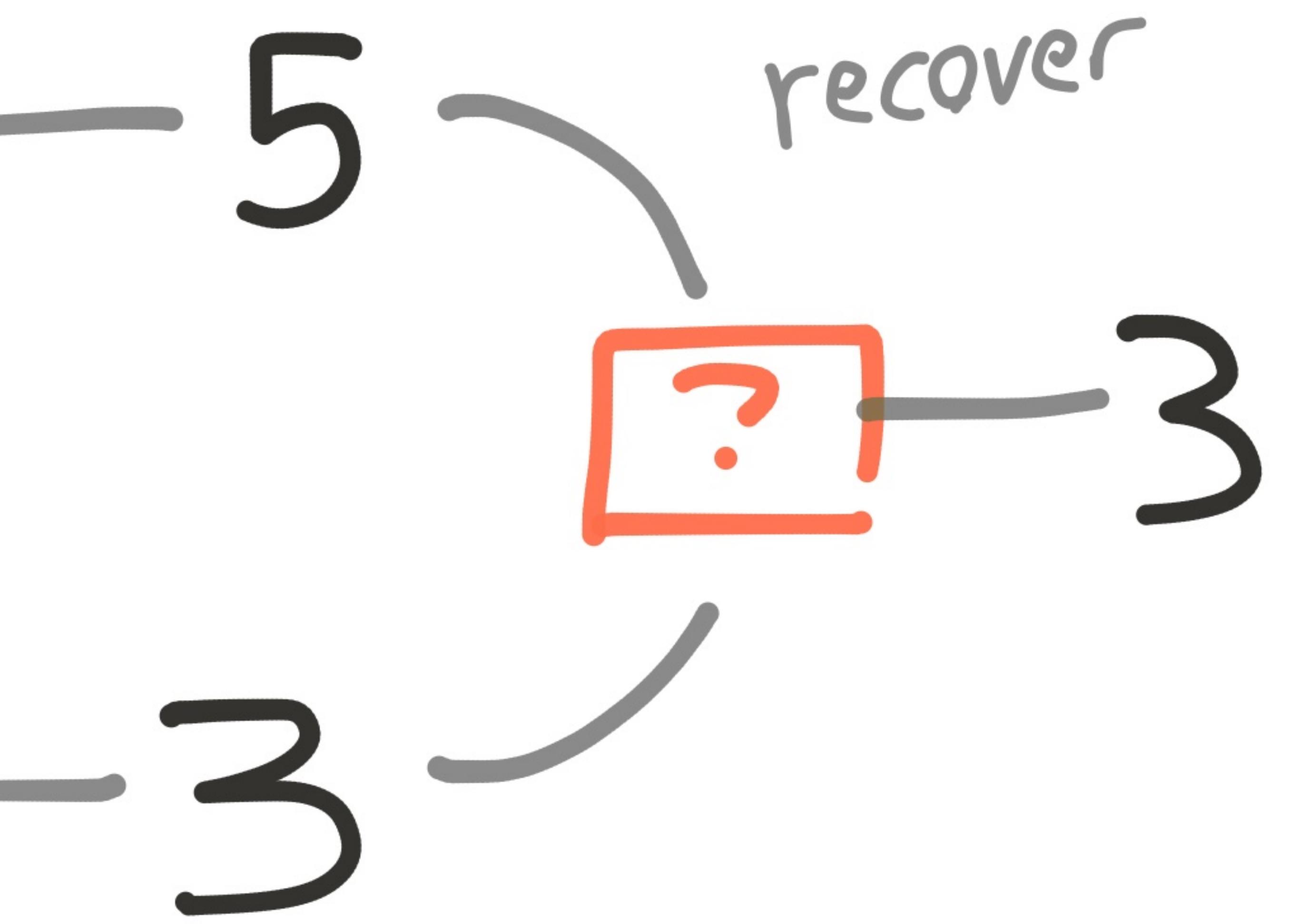
$$2 + 1 = 3$$

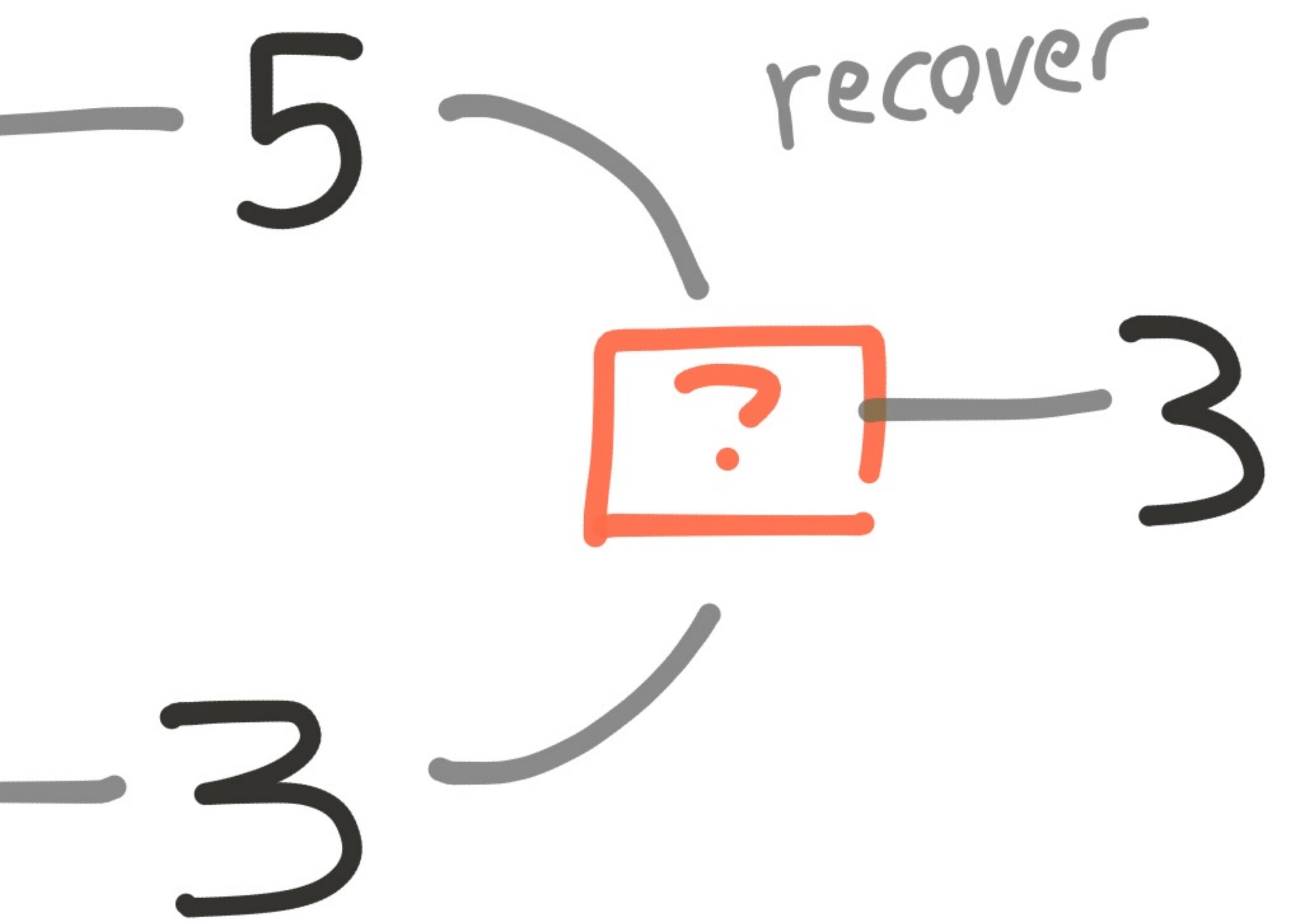












LOOK AT
—your—
LIFE



LOOK AT
—your—
LIFE

—5 — recover
—3 — ? — 3

—3 —
LOOK at your
CHOICES



ID Generator

- Same deal

~91,000 dup ids in 30 s

834,000 generated

Maps

- putIfAbsent (k, v)
- replace (k, v, v')



Lost updates!

Even with
quorum protection

However...

w/ custom merge

function, can build

safe CRDTs

Recommendations

- Choose replication algo appropriate for datatype
- Flake IDs, PN-Counters, Paxos

- Don't rely on Hz for safety!
- Bagri DB txn IDs
- Orient DB cluster state
(used to be txns!)

DISTRIBUTED



aren't

R E A I



3.99.0.3

Distributed, sharded

{ Document }

store

3.5.4:

- Trivial split-brain
- Lost updates
- Stale & Dirty reads

2 Years of
fundamental redesign
work . . .

4.0: Linearizable Mole

- Custom Consensus protocol
- Reduced loss in AP mode

(start
of
test)

read ↗

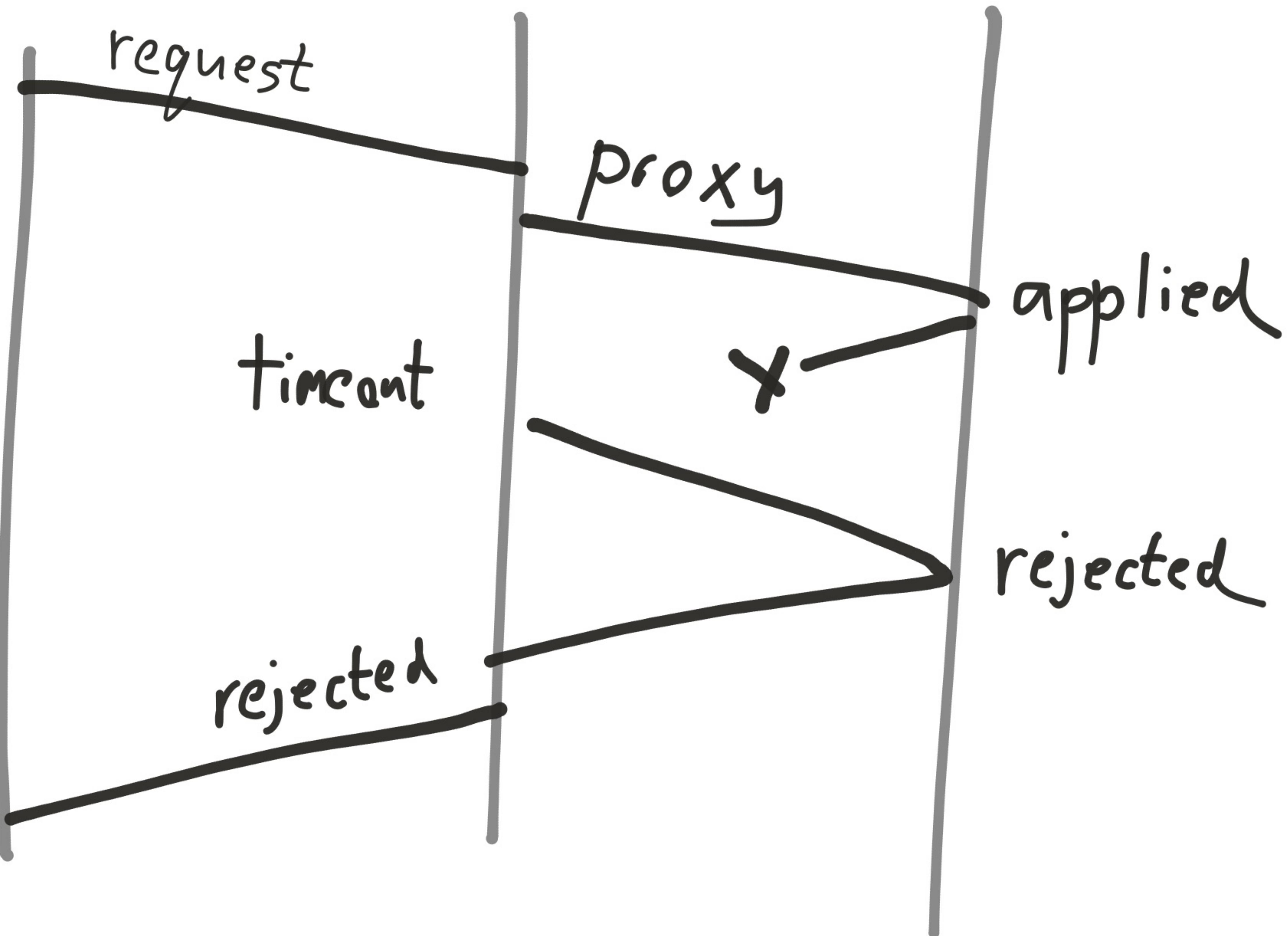
fail



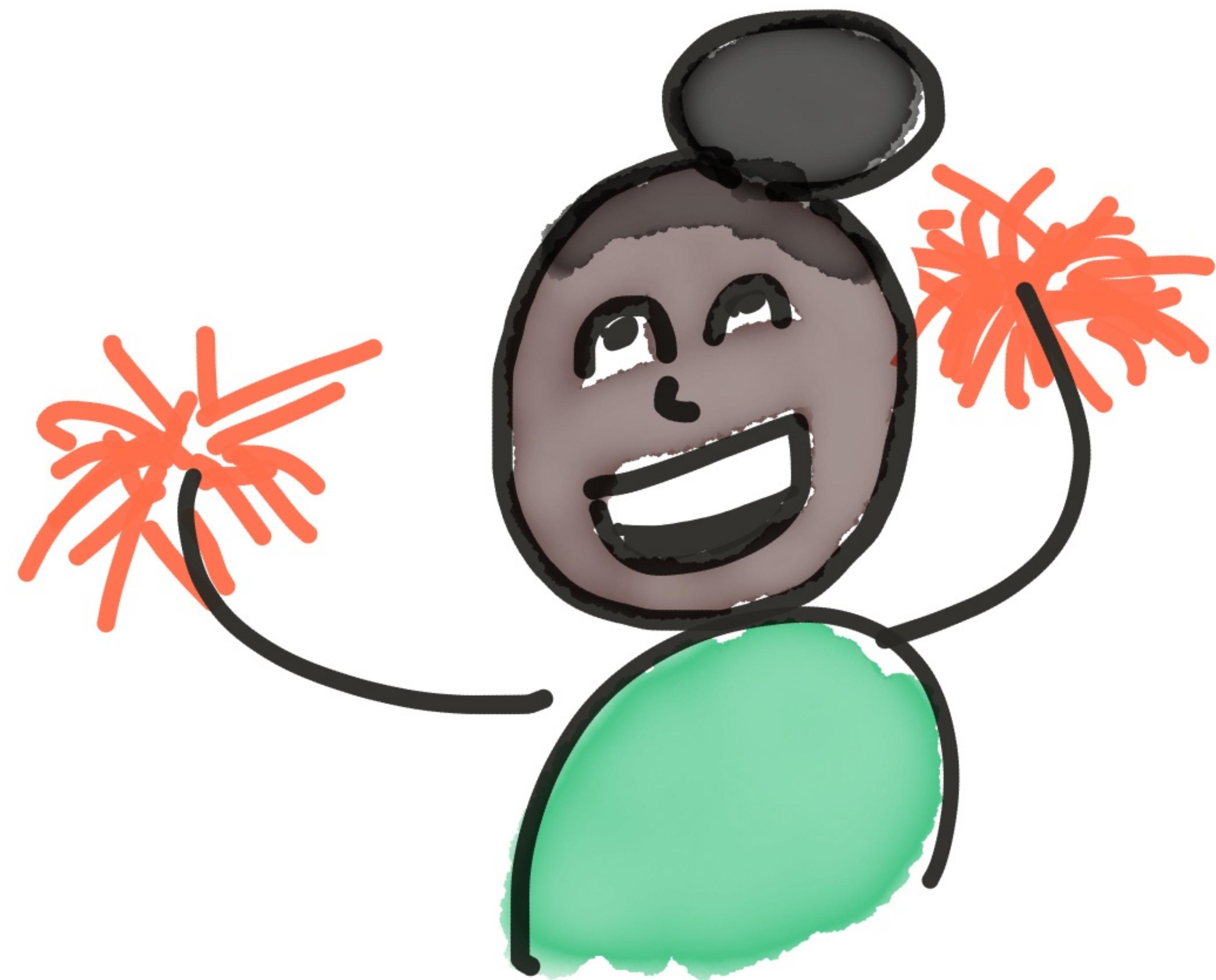
ok



client server server



fixed in 3.99.1.5



Node crashes



client

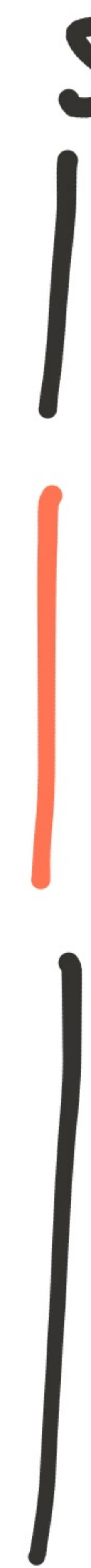
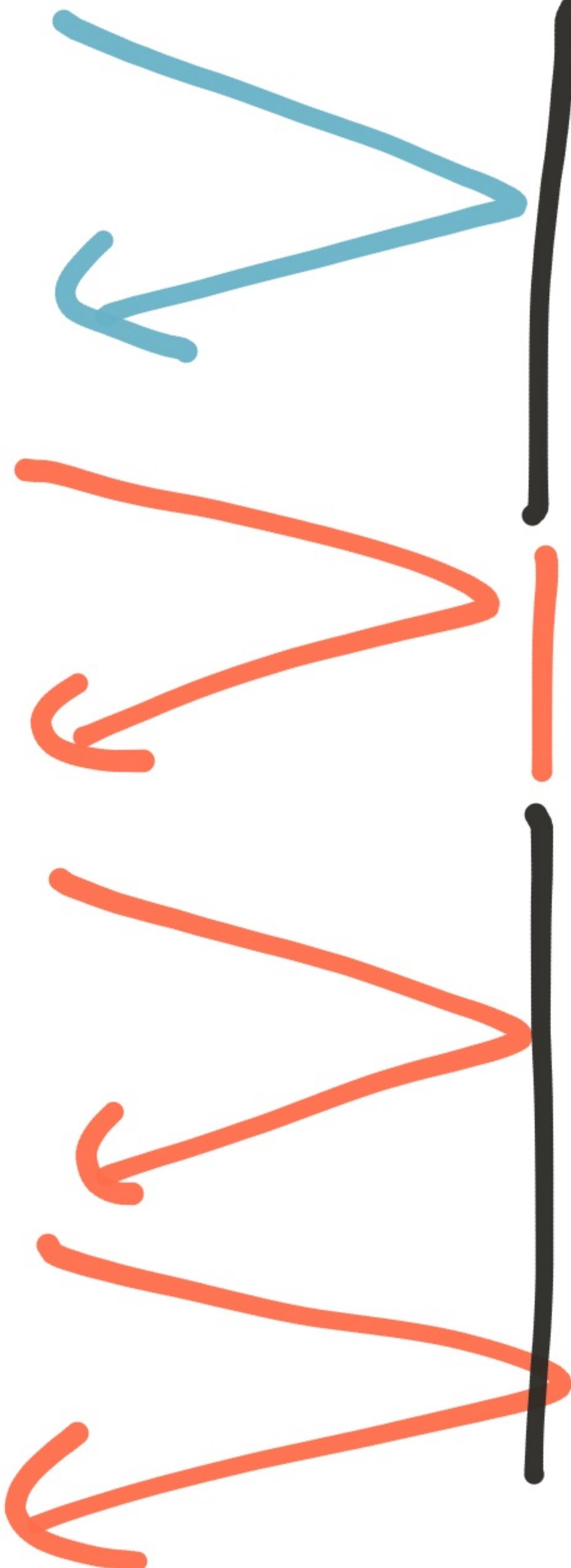
s

s

s

s

s



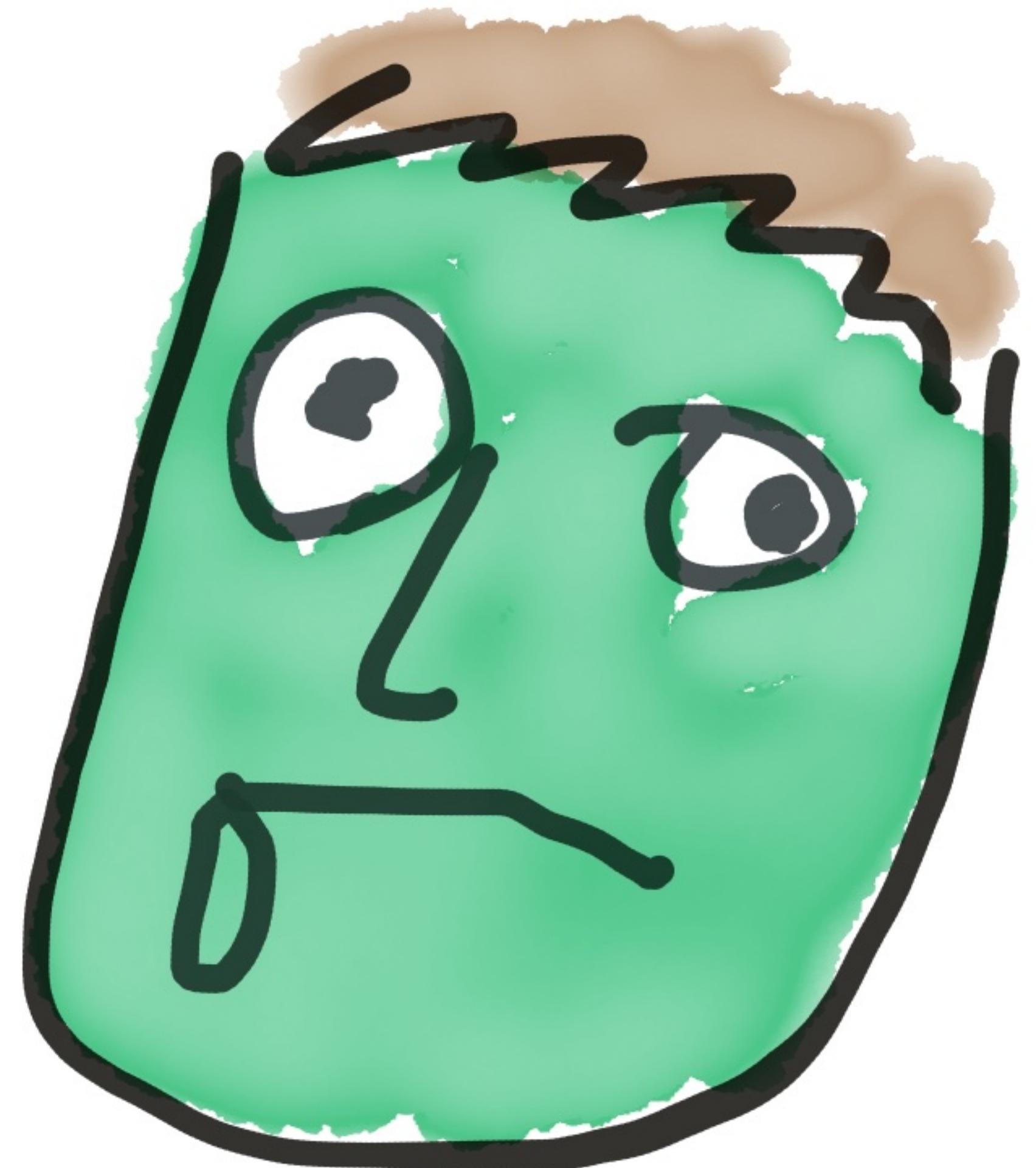
[press E to revive]



Dead partitions

may not be all

there...



AS didn't flush

writes to disk

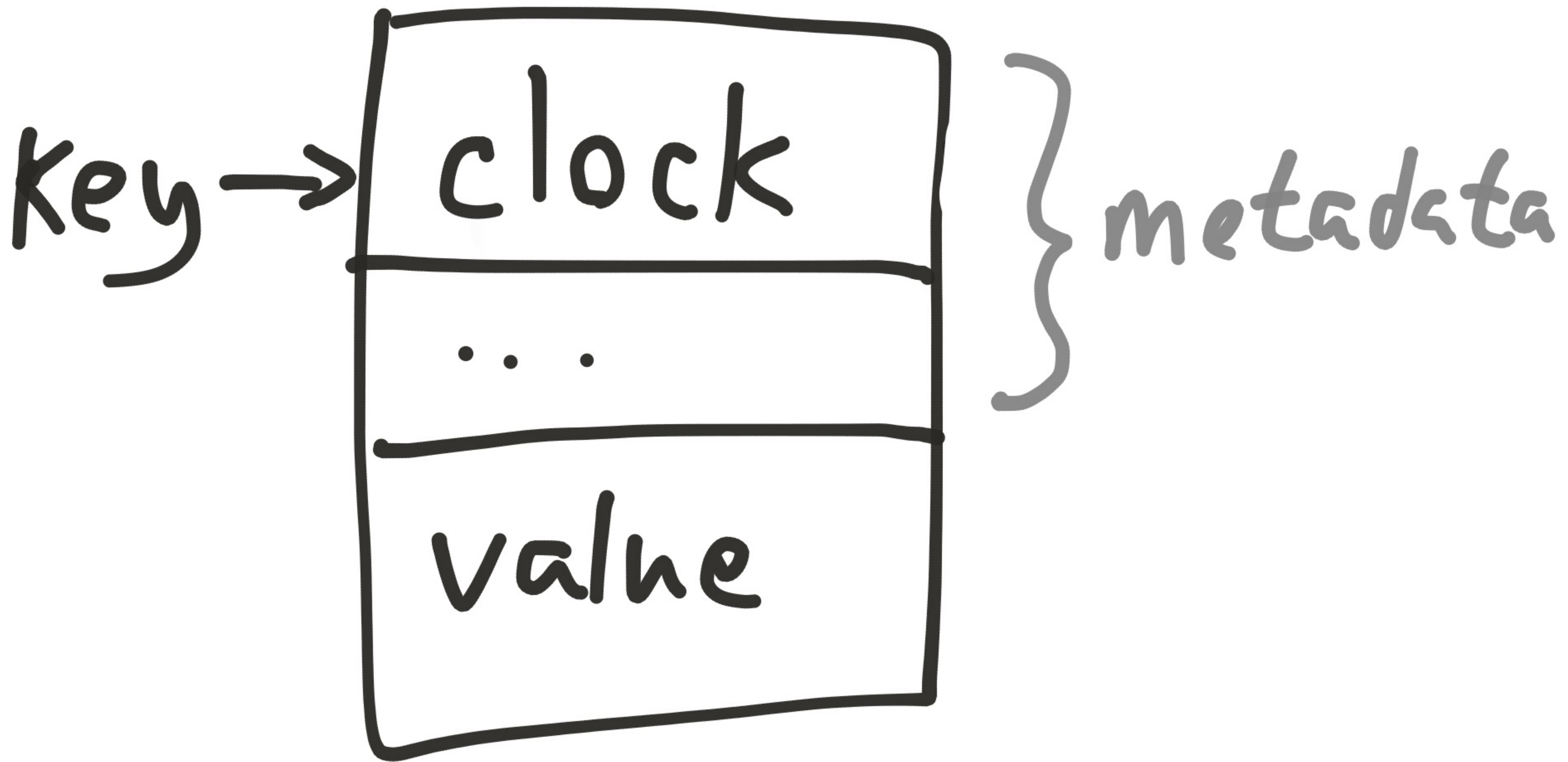
before acknowledging

(fixed in 3.99.2.1)

Clock Skew can

lead to Lost

updates



Clock

regime

wall-clock

Counter

increment
on
election

node
local

for mult.
updates in
same clock

fits in cache line

regime

wall-clock

Counter

6 bits

w_1

A (primary for regime 1)

! pause!

⋮

⋮

⋮

B

⋮

w_1

A

(primary for regime 1)

! pause!

|

|

|

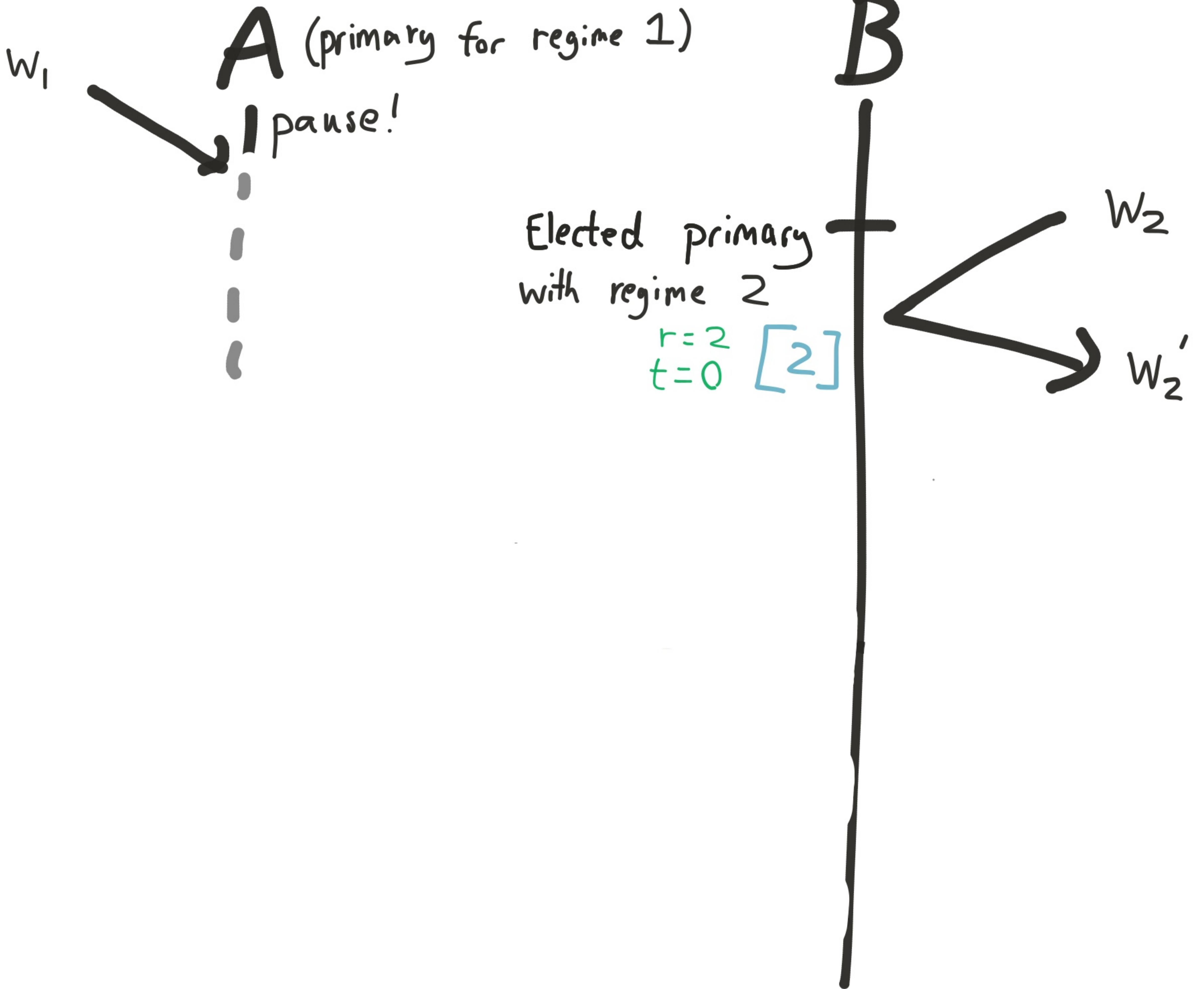
Elected primary
with regime 2

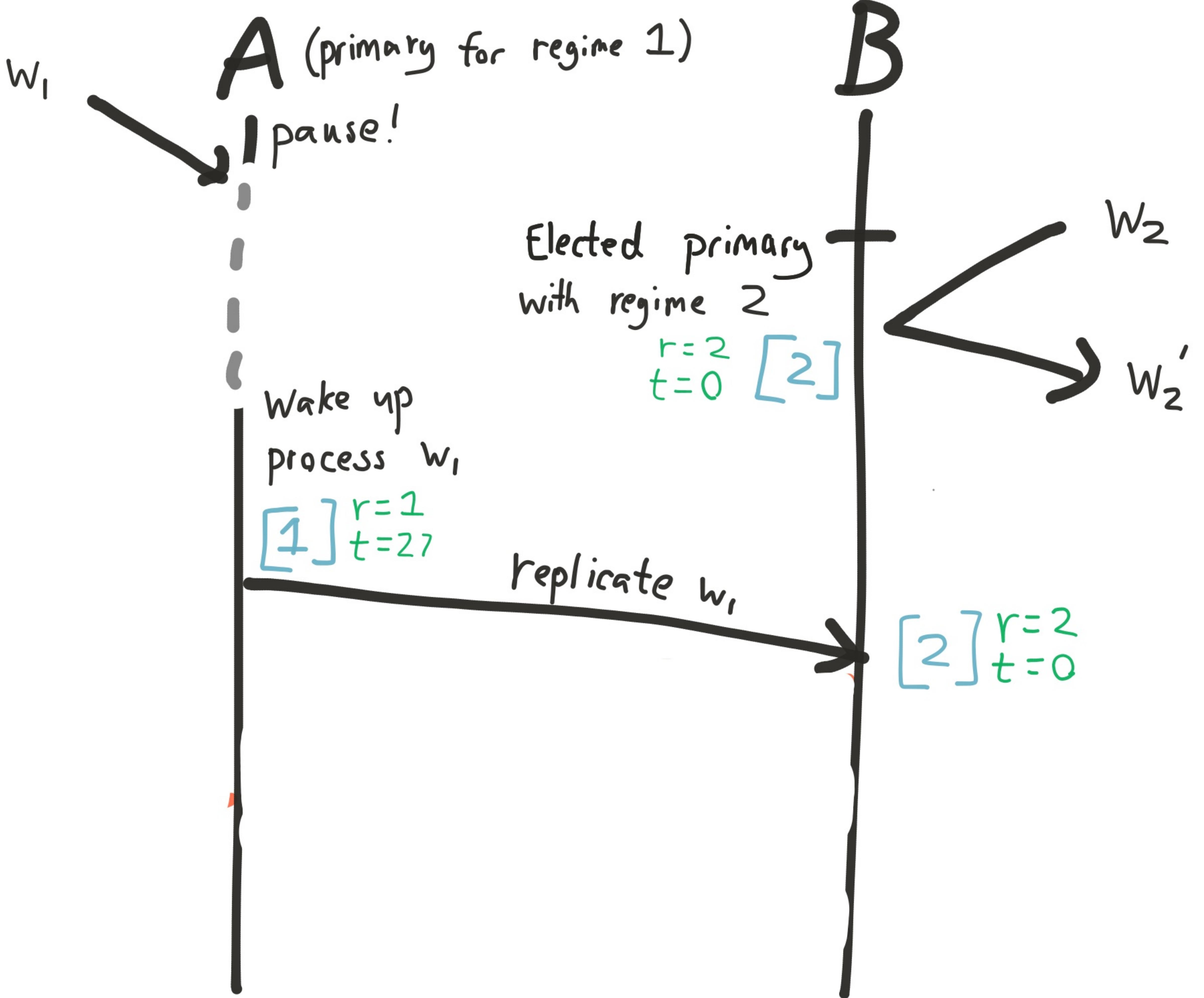
B

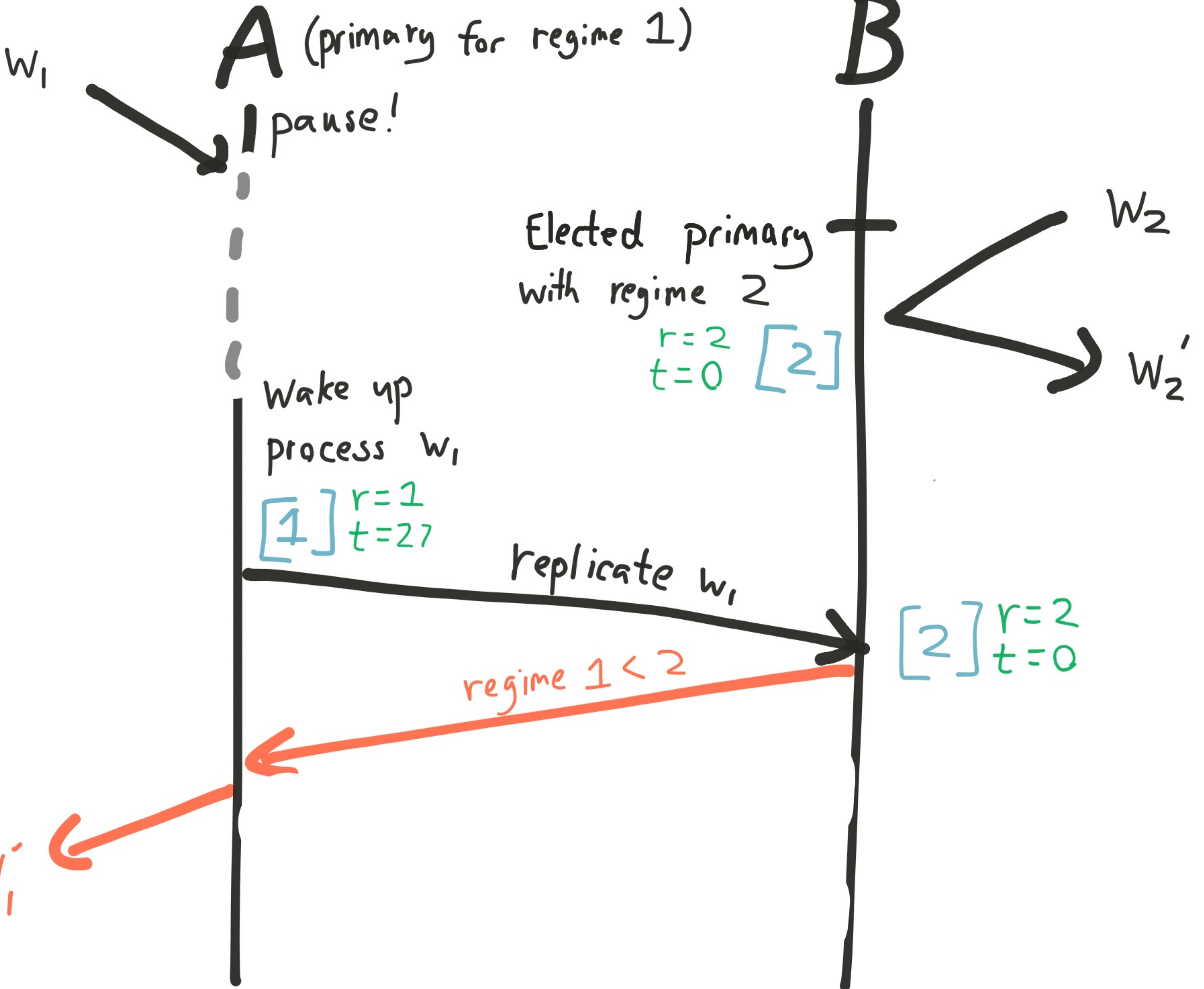
|

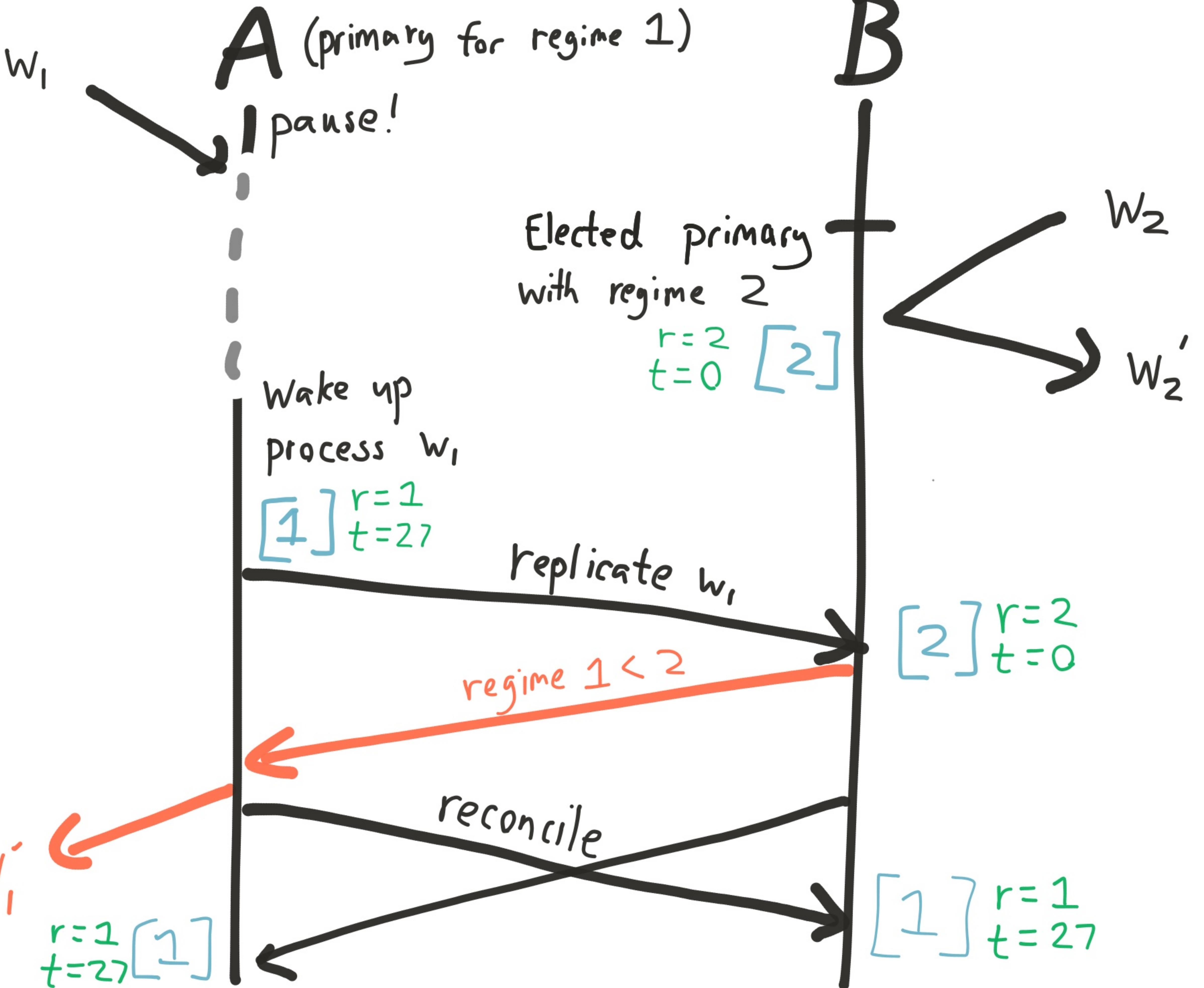
+

|











REALLY?

Beth clock skew ≠

hot using fSync were

Known failure modes

Stay in clock-skew
bands & use durable
flag . . .

looks |inearizable!

But XDR, scans, UDFs
don't go through the SC
mechanism yet

future work!

Recap



Read the docs!



Then test it

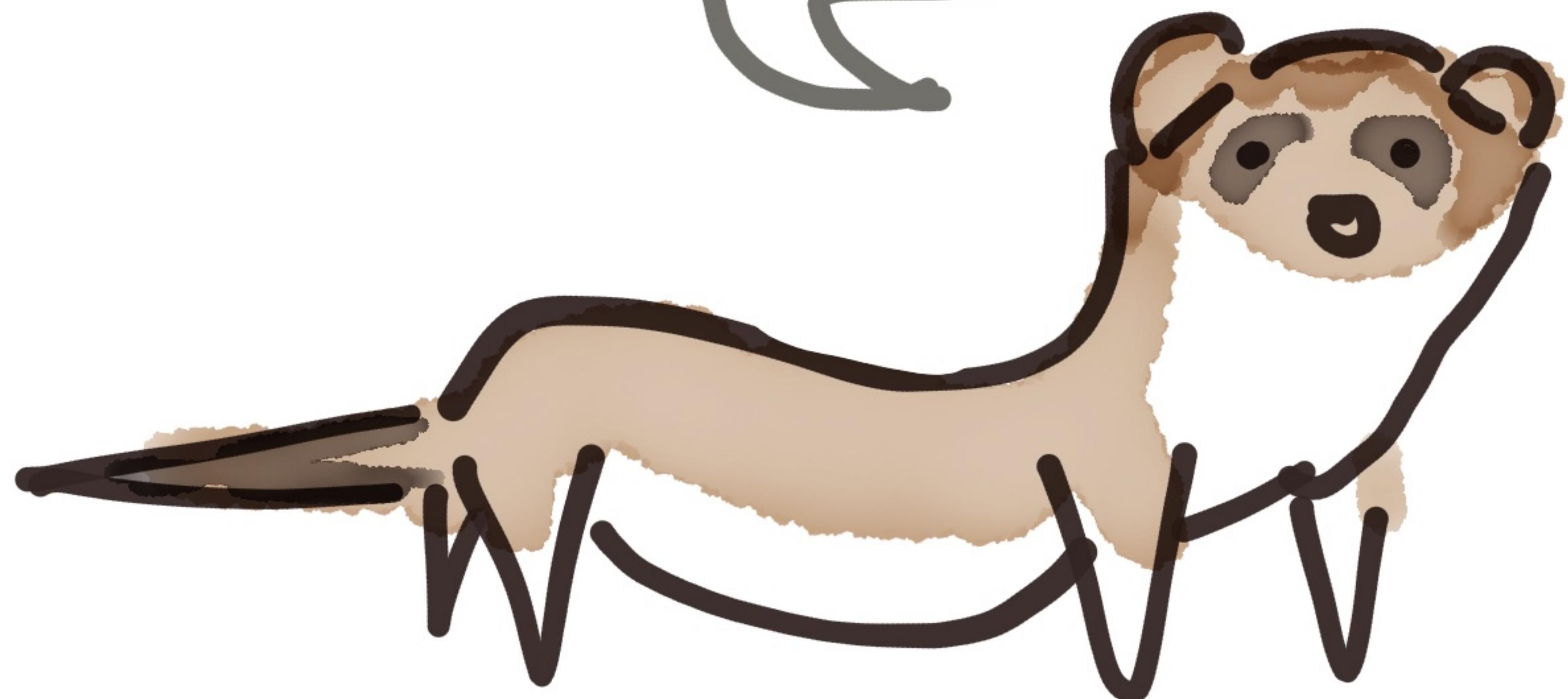
— for —

yourself

"Strict"

"ACID"

"Strong"



Be Formal

Be Specific.

*Figure out the
invariants
your system needs*

Consider
your
failure modes

Processus Crash

#kill -9 1234

Node failure

- AWS terminate
- Physical power switch

Clock Skew

```
# date 10 28 0000  
# fake time ...
```

GC/IO Pause

killall -s STOP foo

killall -s CONT foo

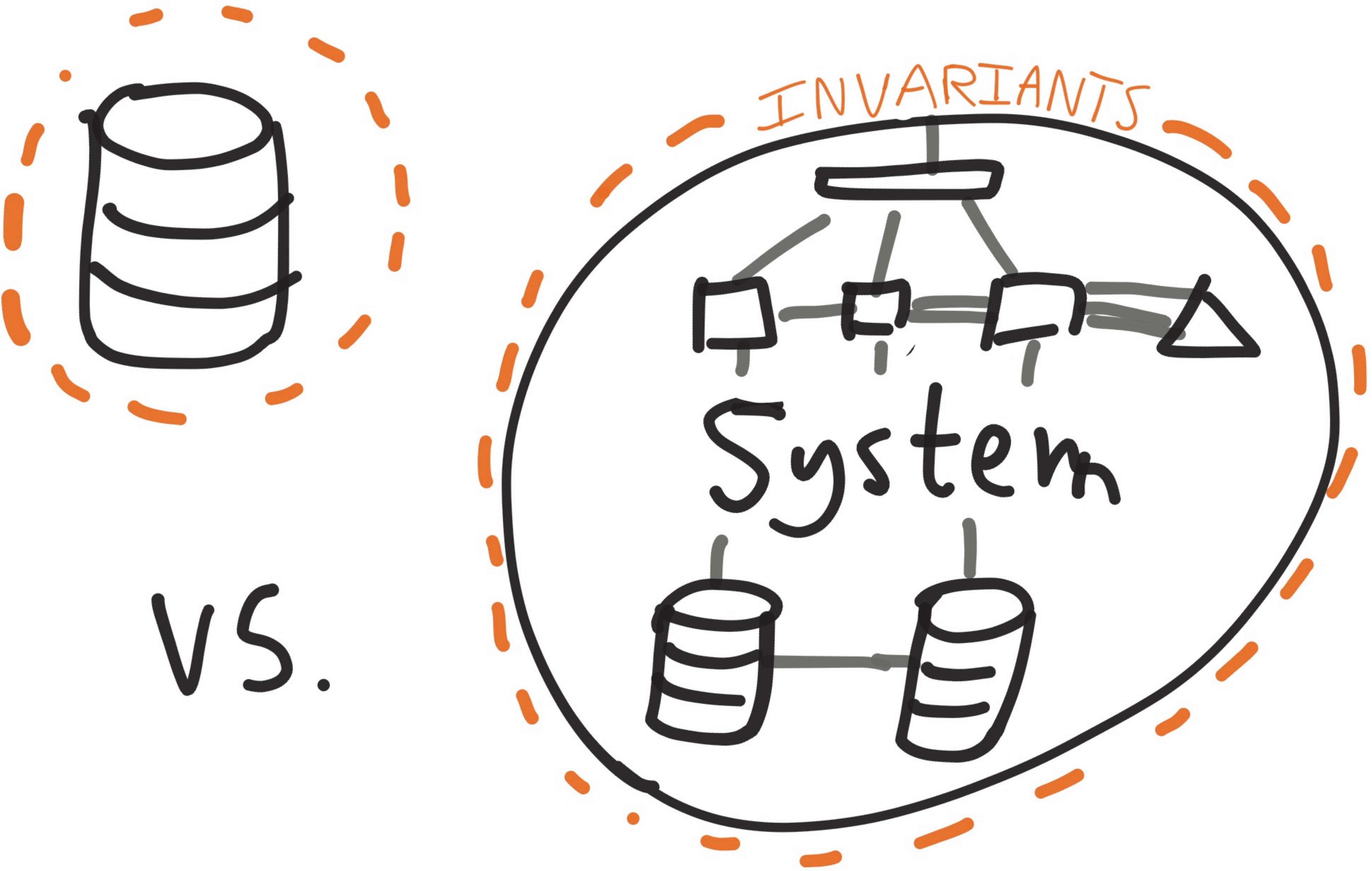
Network Partition

```
# iptables -j DREP  
# tc qdisc ... delay...  
      drop...
```

Test your

Systems

end ————— to —————→ end



If you look for

+ *perfection* +

you will never be
content

- Property testing
- High-level invariants
- With distsys failure Modes

Thanks

Tendermint 

Hazelcast

Jordan Halterman

Luigi Dell'Aquila

Luca Galli

Thanks

Denis Sukhoreslev

Aerospike



Peter Alvara



<http://jepsen.ie>