

Jepsen
a syncing feeling



Kyle Kingsbury

@Aphyr

I break
databases!

Public
API {



} API code

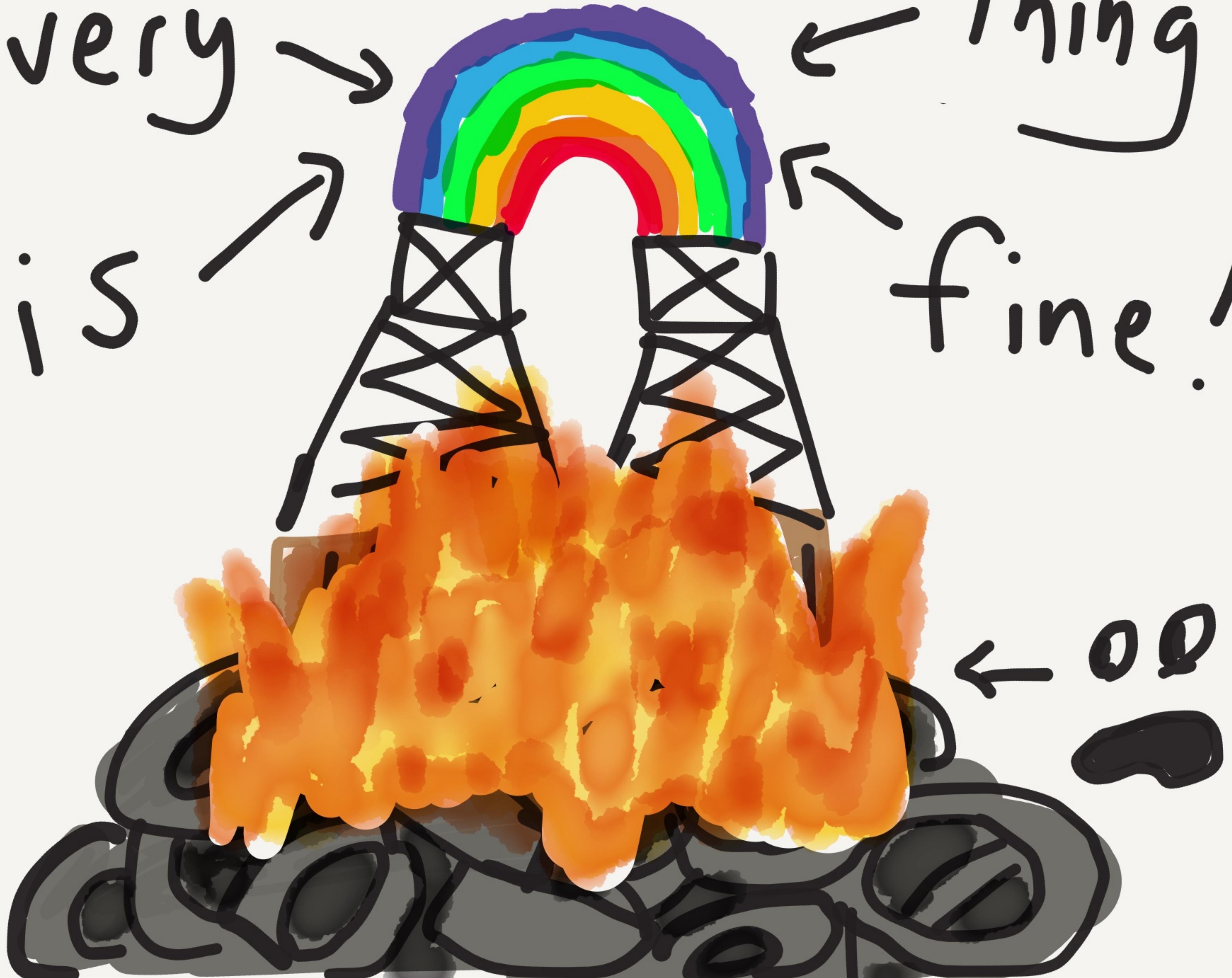
Ruby {



DBs

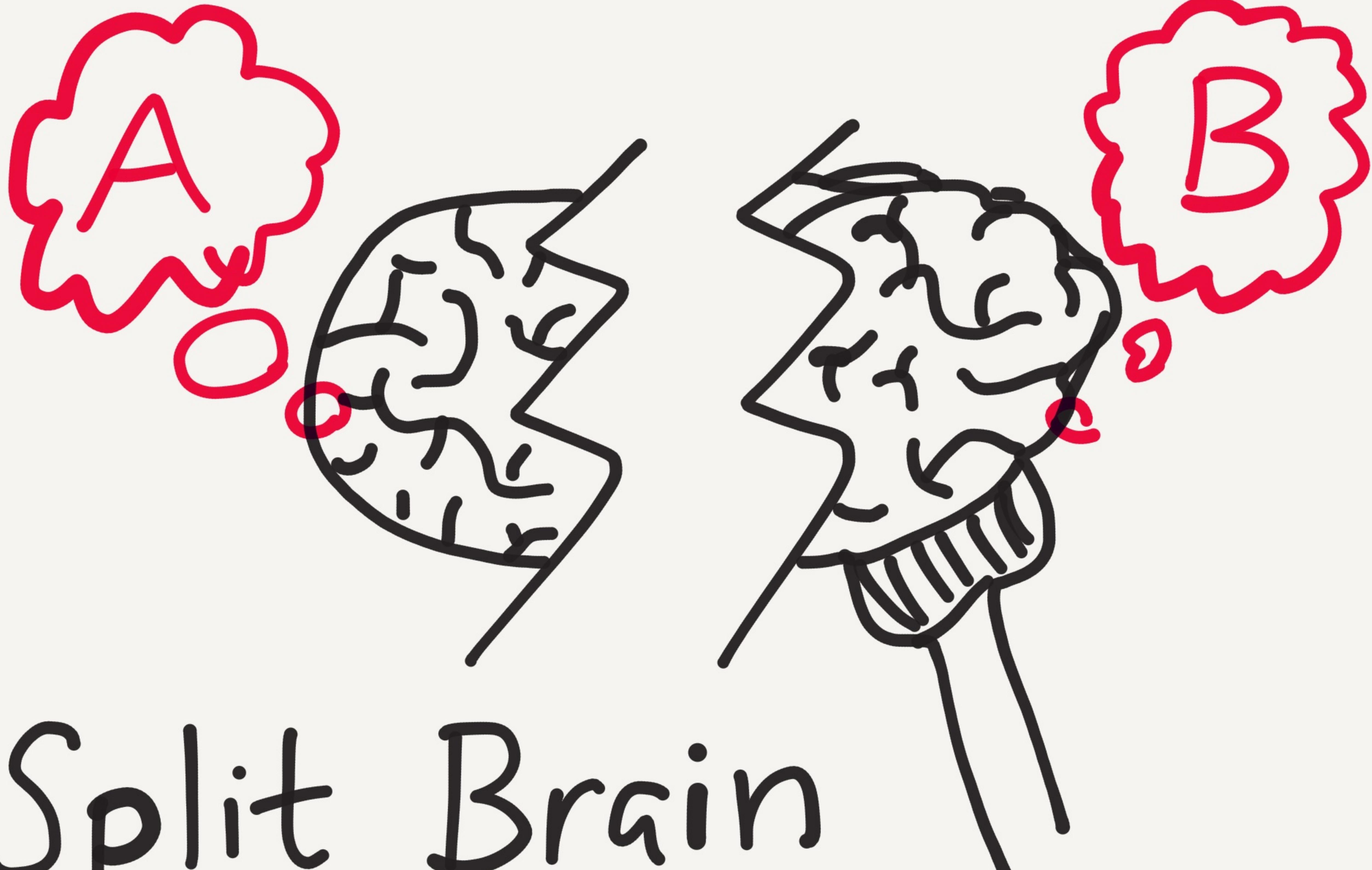
Every →
is →

← Thing
fine !



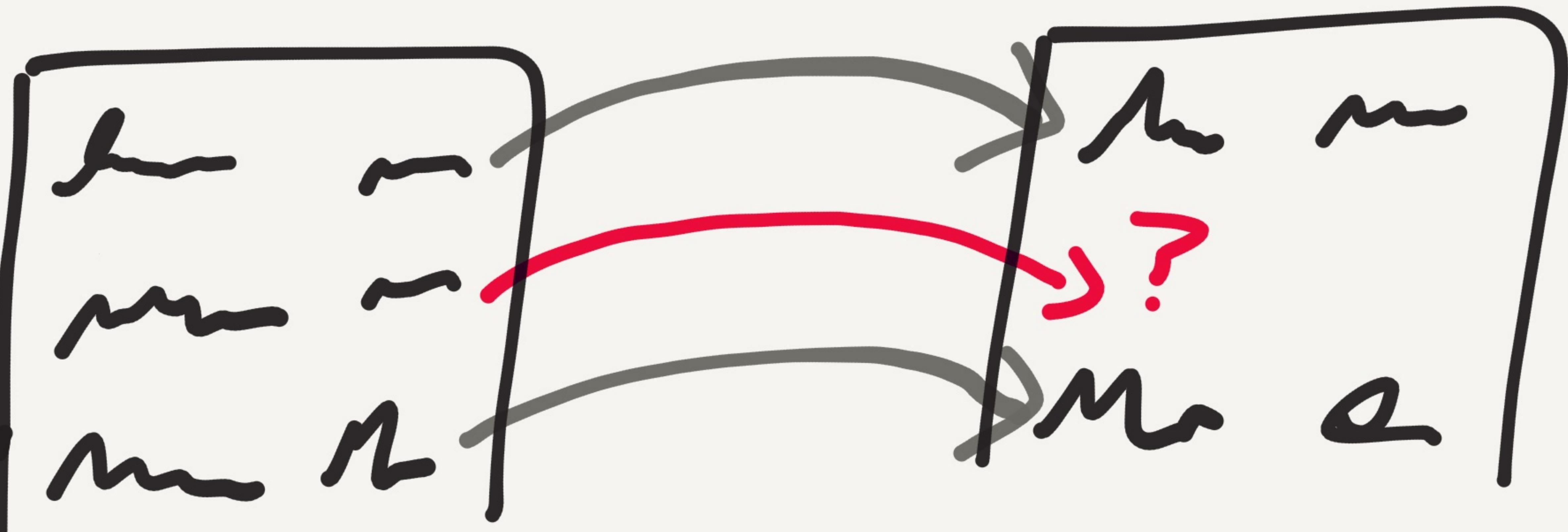
← 00

Databases! /
Queues! /
Discovery! /
THE HORROR

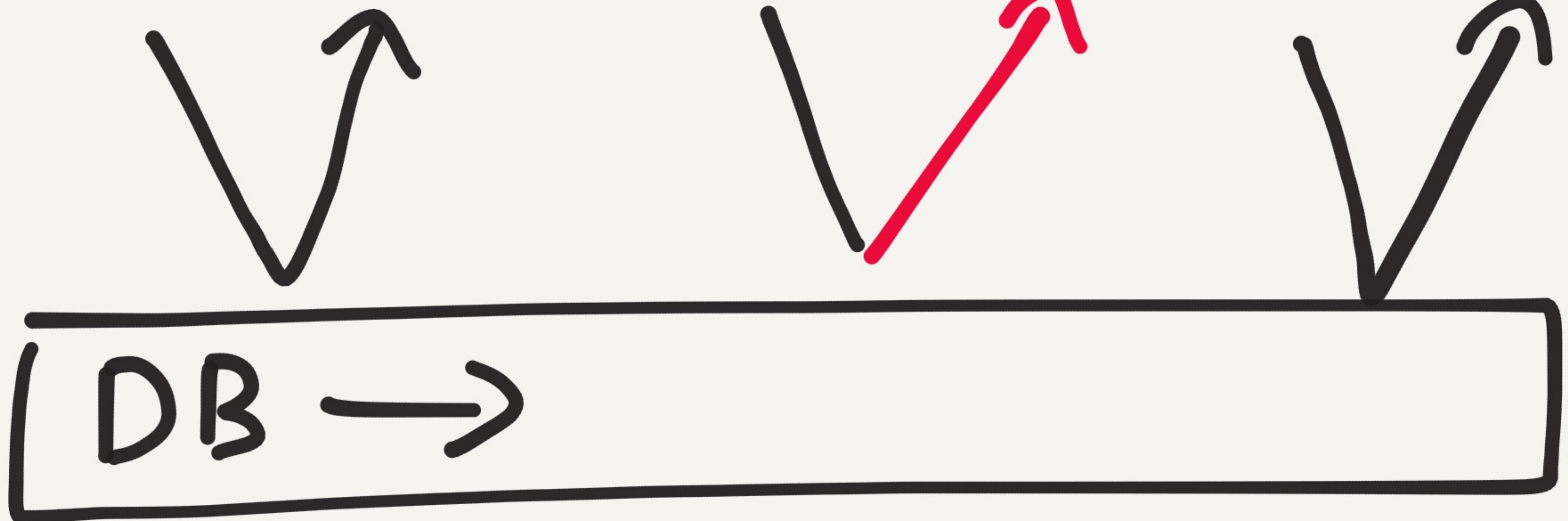


Split Brain

Broken Foreign Keys



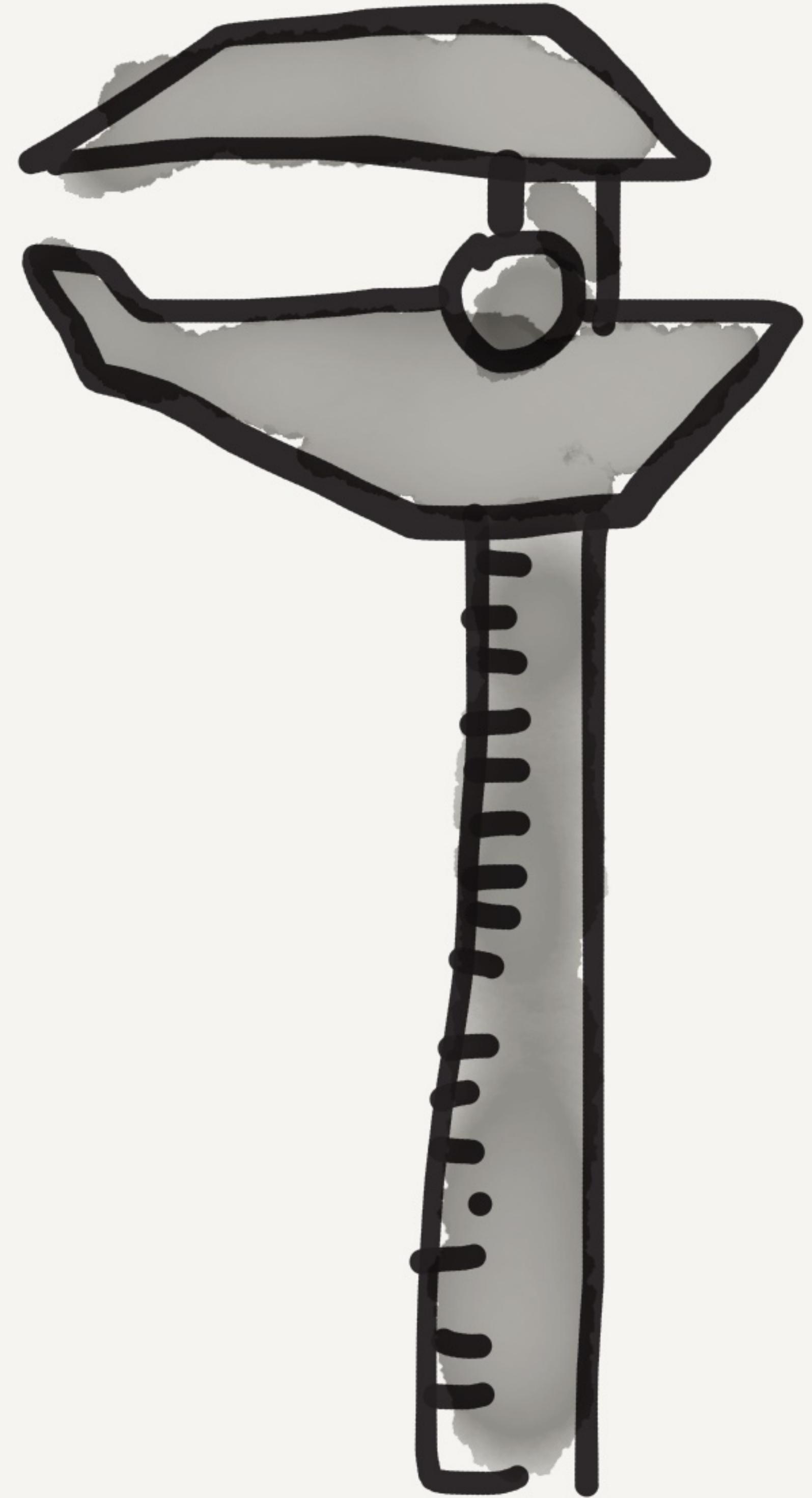
write ok read ↗ read ok



Anomalies

How do you
know if a
system is safe?

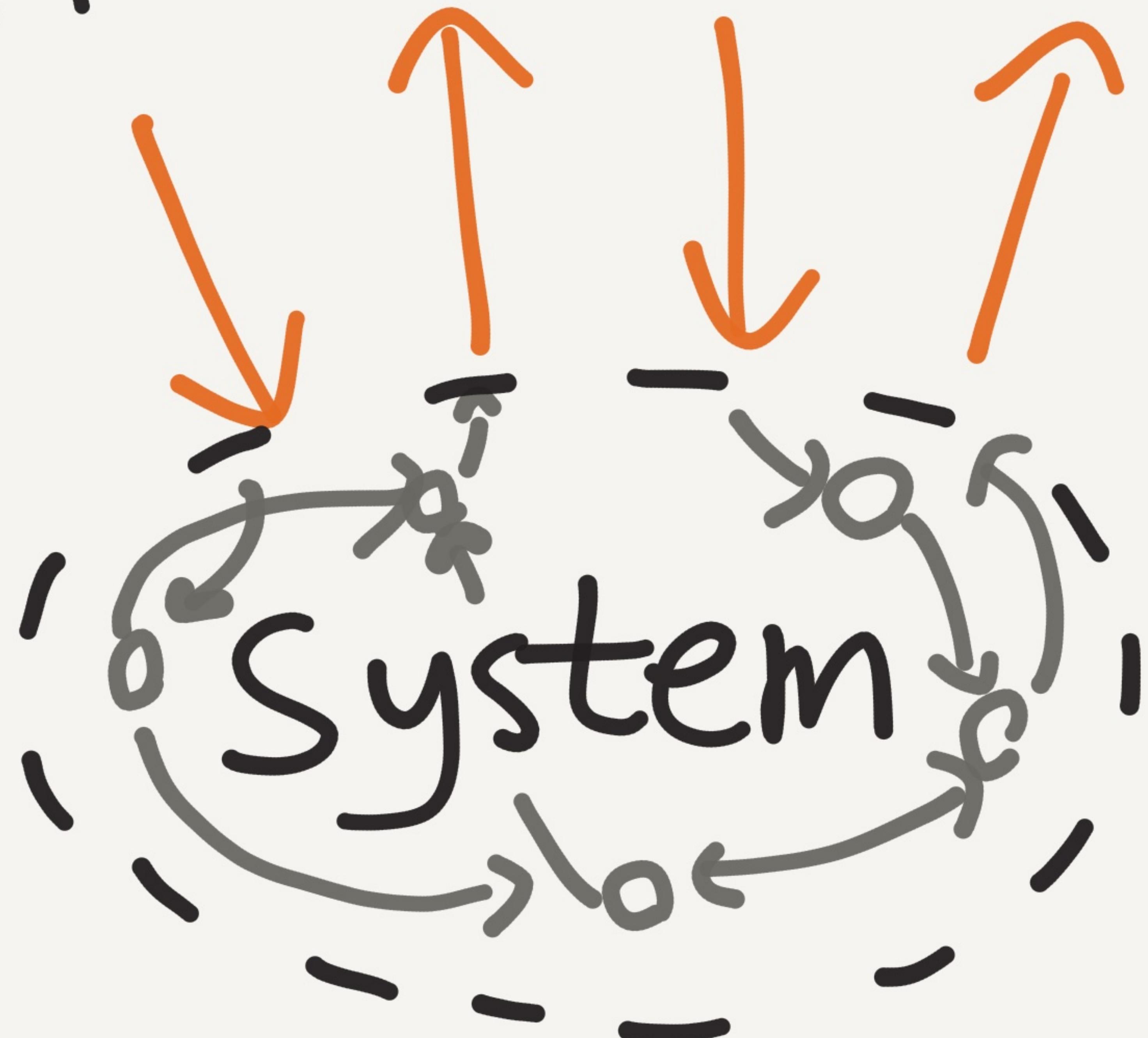
Measure
your
Systems



Jepsen

github.com/aphyr/jepsen

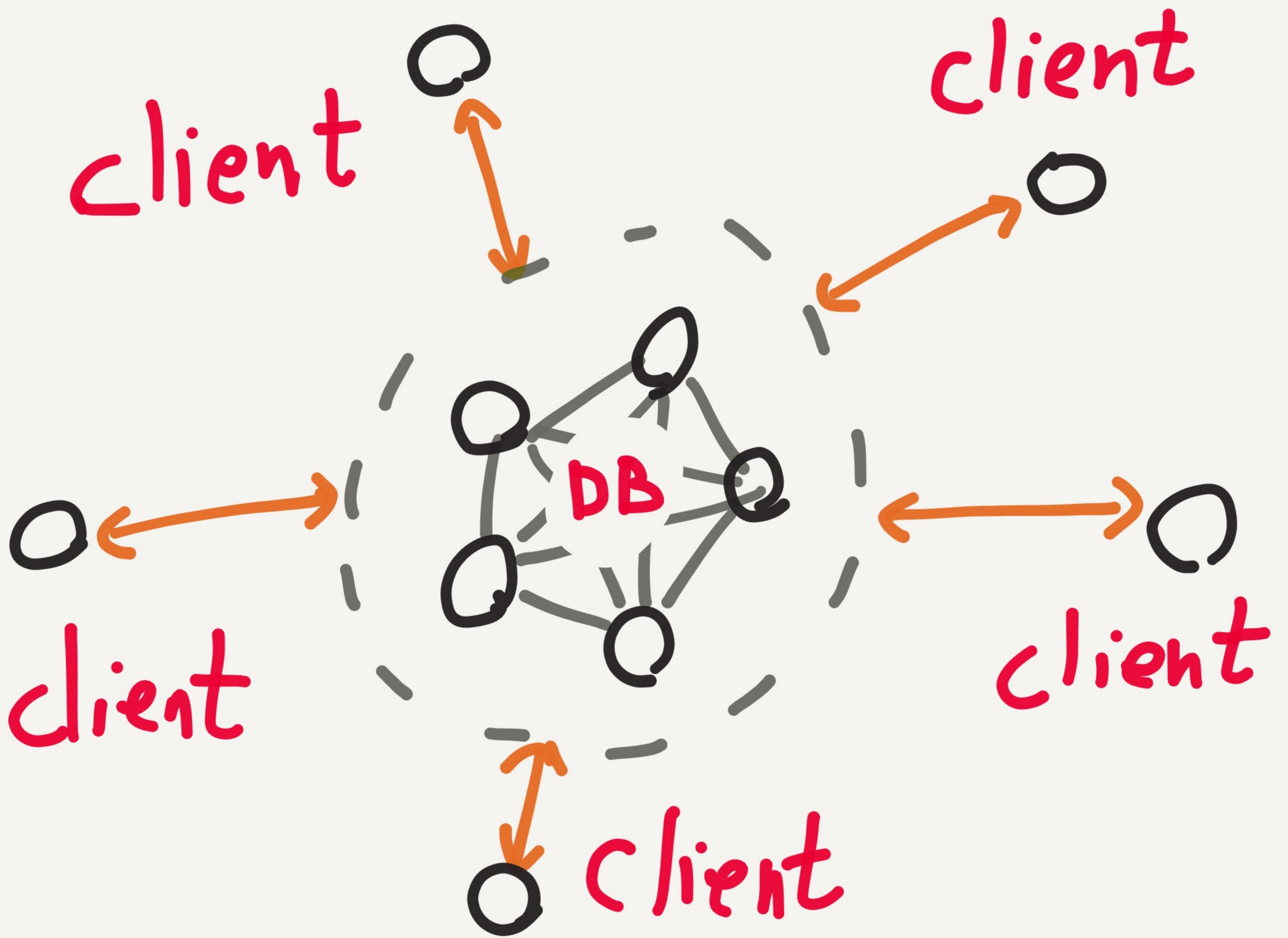
Environment





—INVARIANTS—





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

The diagram illustrates a network topology where three client nodes are connected to a central DB node. Each client node (w , r , and w') has two outgoing arrows pointing towards the DB node, indicating bidirectional communication or replication.

DB:

client: w — w' w — ---

The diagram shows a single client node (w) connected to a DB node. The connection is represented by a single solid arrow pointing from the client to the DB, indicating a unidirectional connection or a specific type of communication path.

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



invoke

ok

invoke

fail

invoke

?

?

info

?

?

?

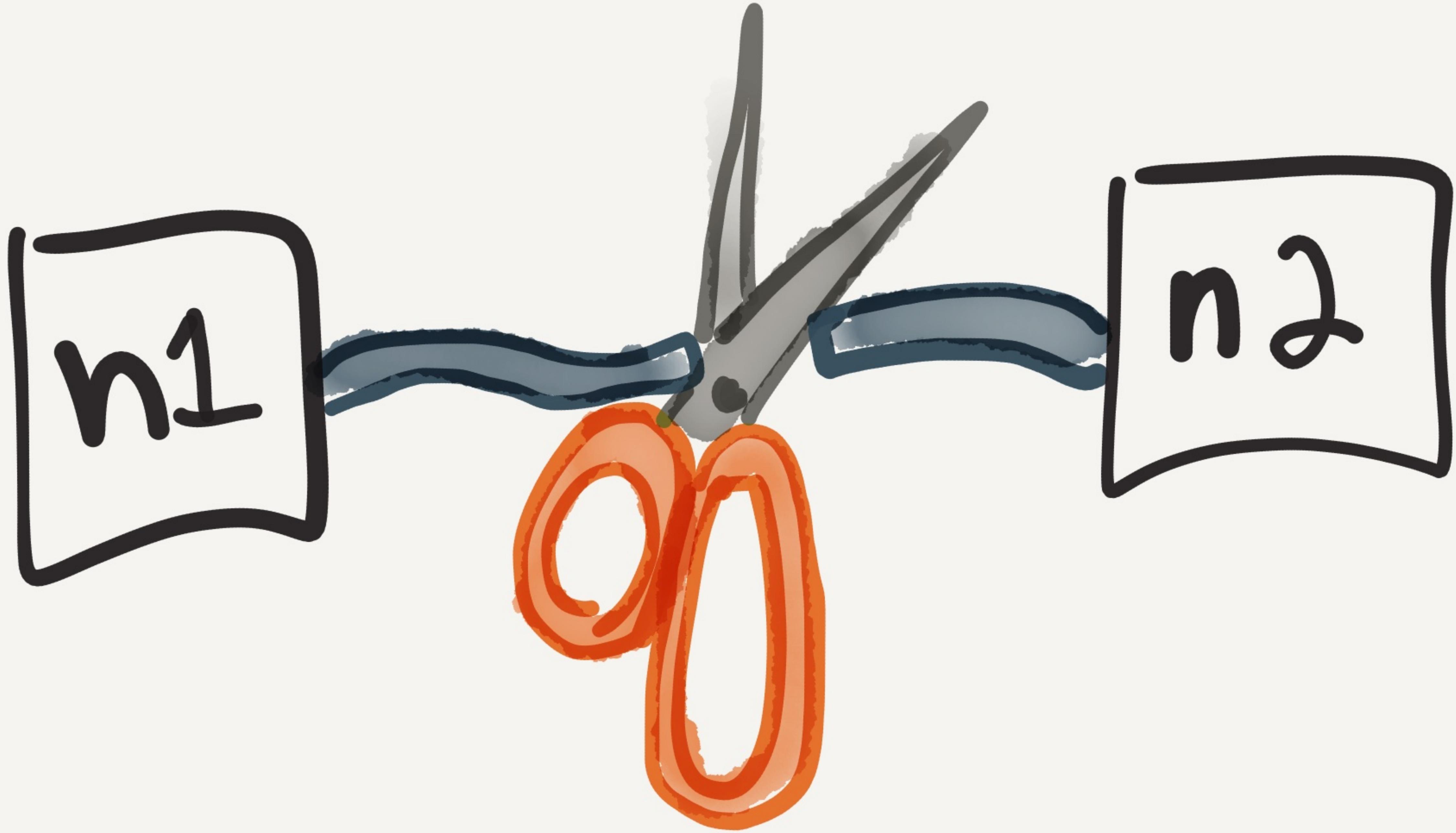
invoke ok

invoke fail

invoke ok

invoke info

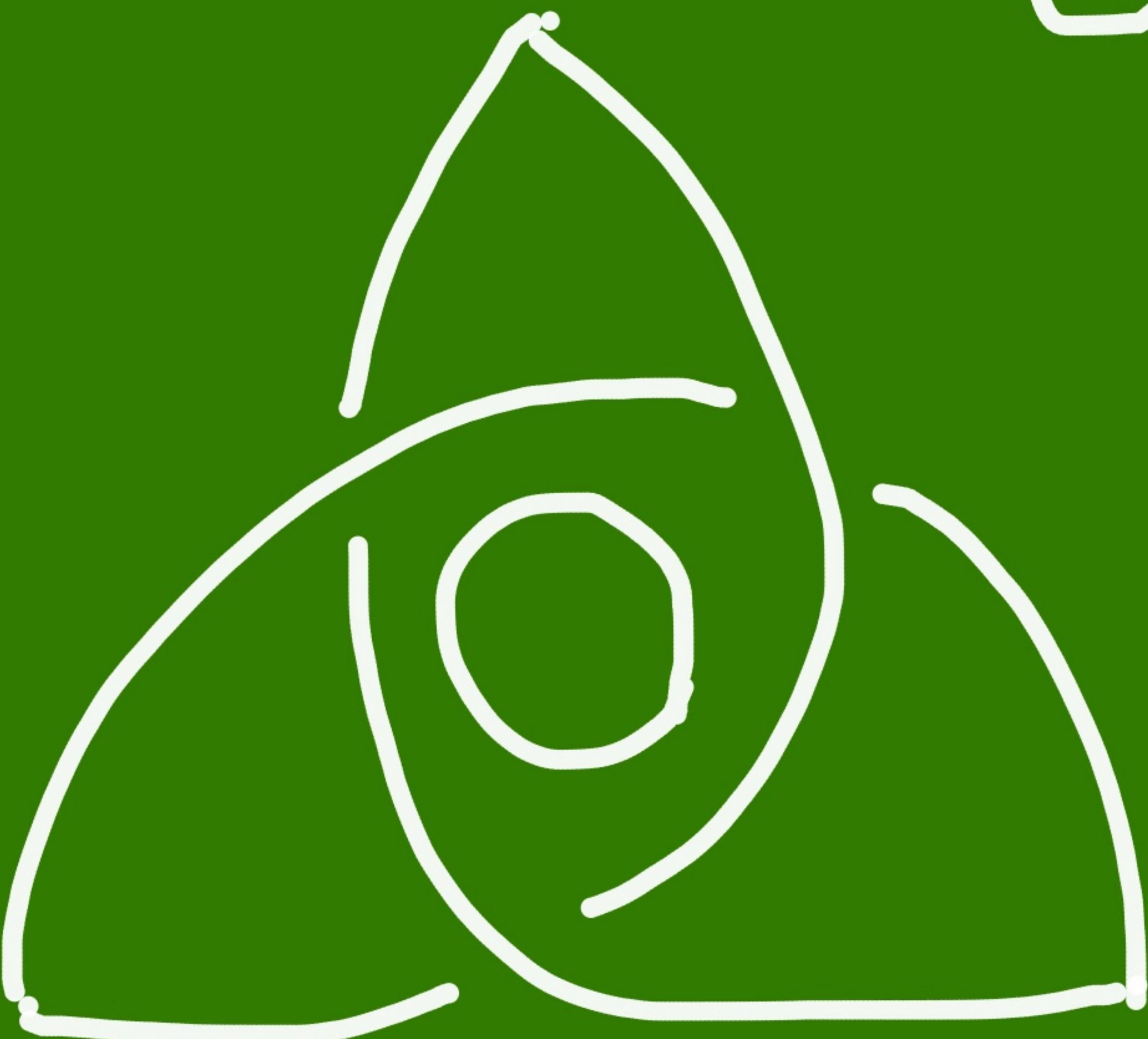
1. Generate ops
2. Record history
3. Verify history is
consistent w/ model



Partitions!

Recent
Research

0.10.2



Tendermint

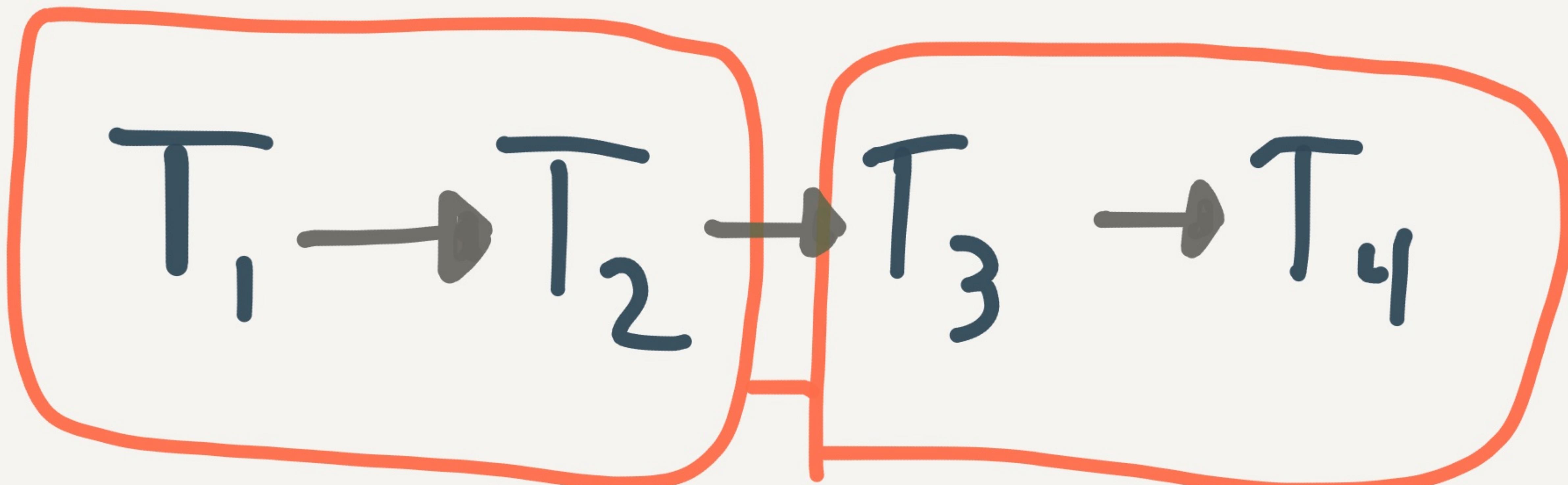
- Transaction replication
- Pluggable FSMs
- Byzantine fault tolerant

"Raft Meets
Blockchain"

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

$$\overline{T_1} \rightarrow \overline{T_2} \rightarrow \overline{T_3} \rightarrow \overline{T_4}$$

Like Raft, provides a linearizable order of txns



blockchain!

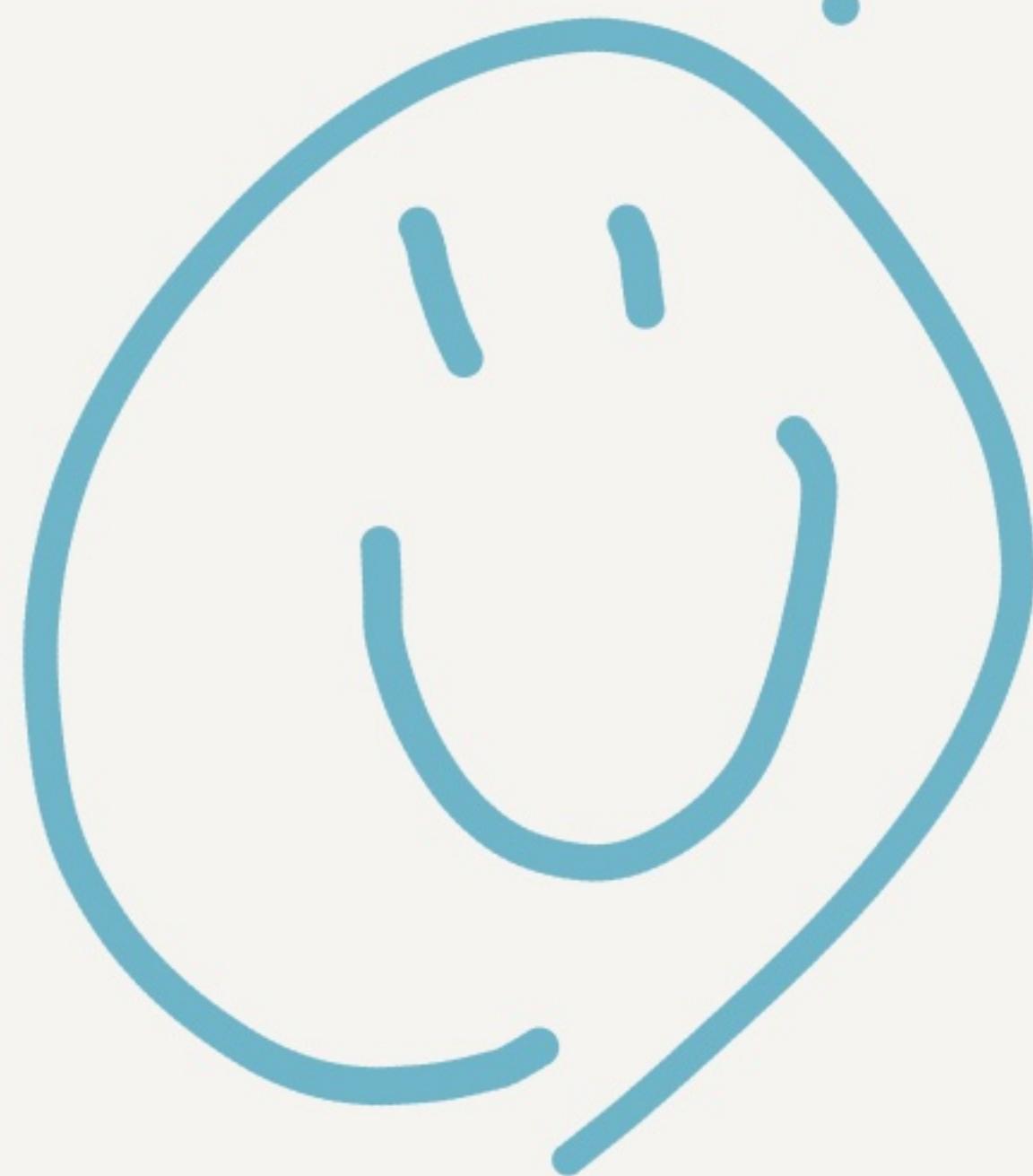
Byzantine

Fault
Tolerant!

shun!



shun!



shun!

shun!



Pluggable

State

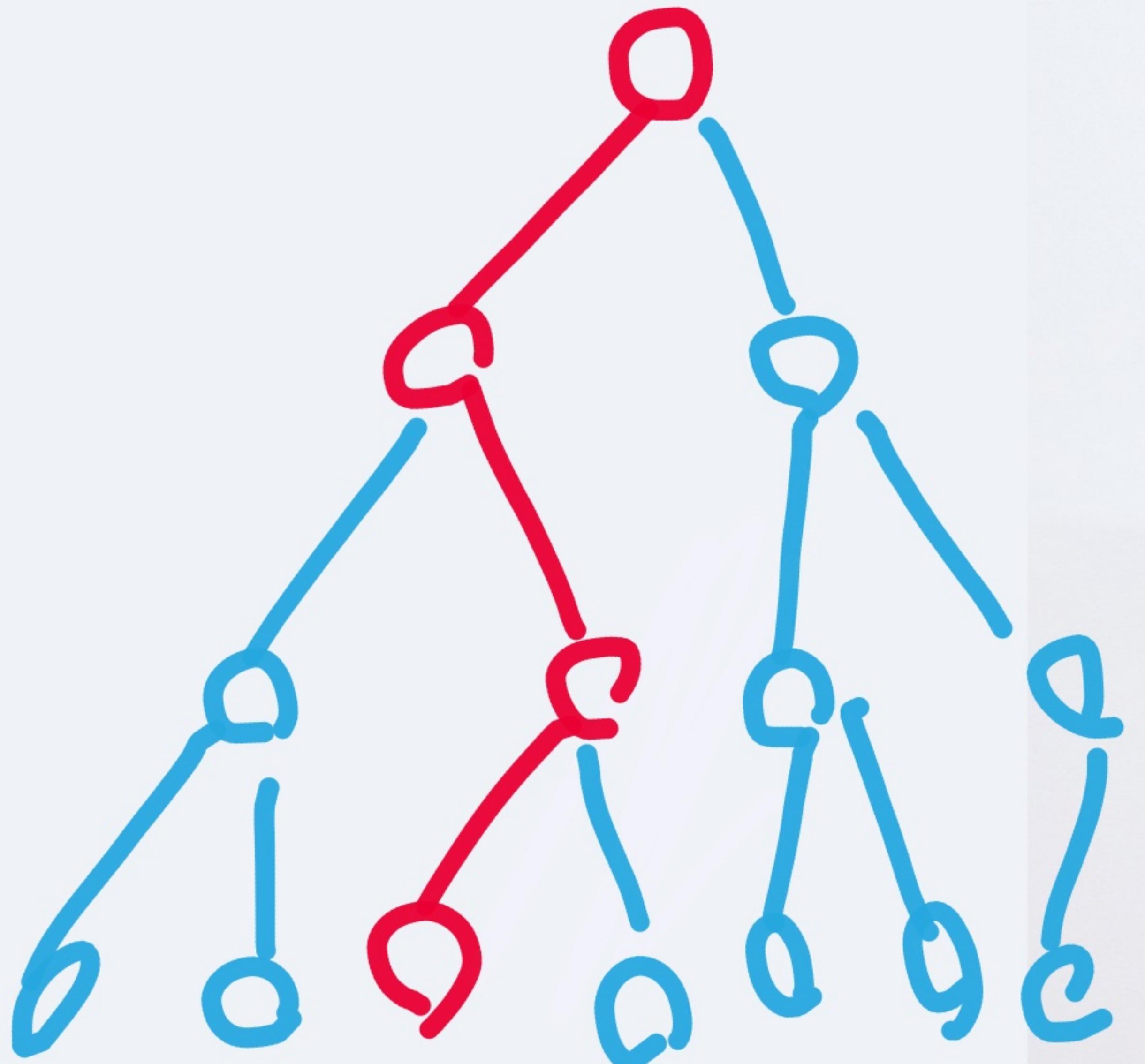
Machines

validator

validator

State

State



merkle
eyes



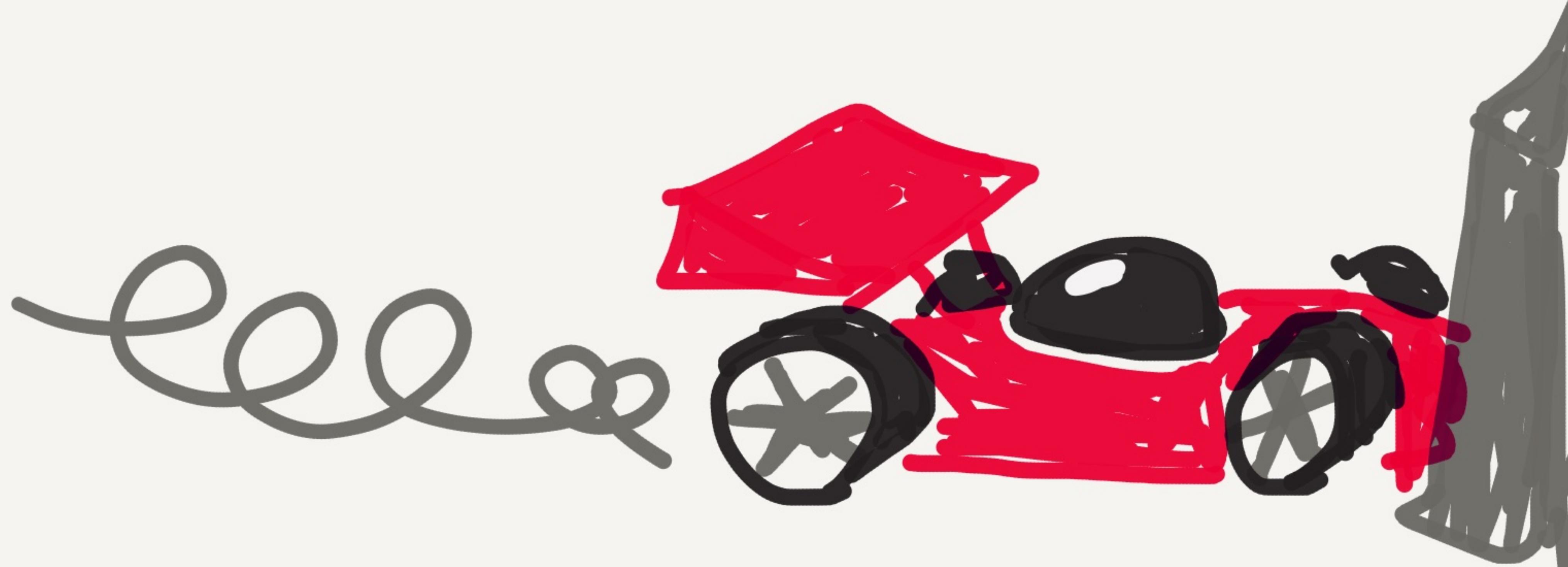
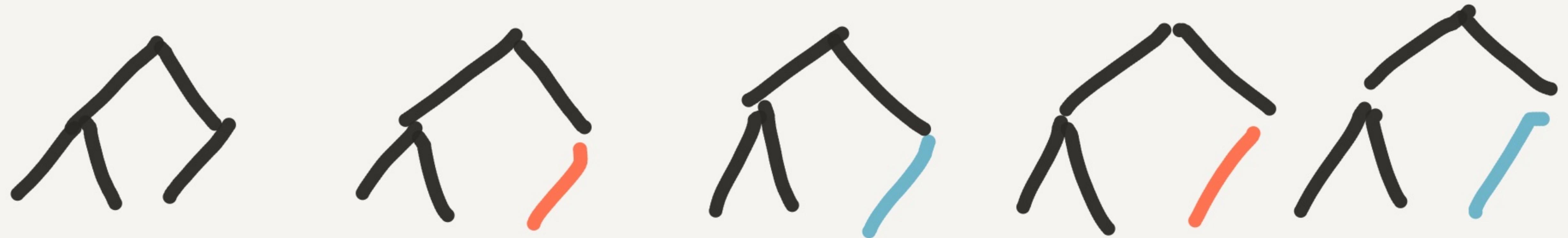
CAS-register

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

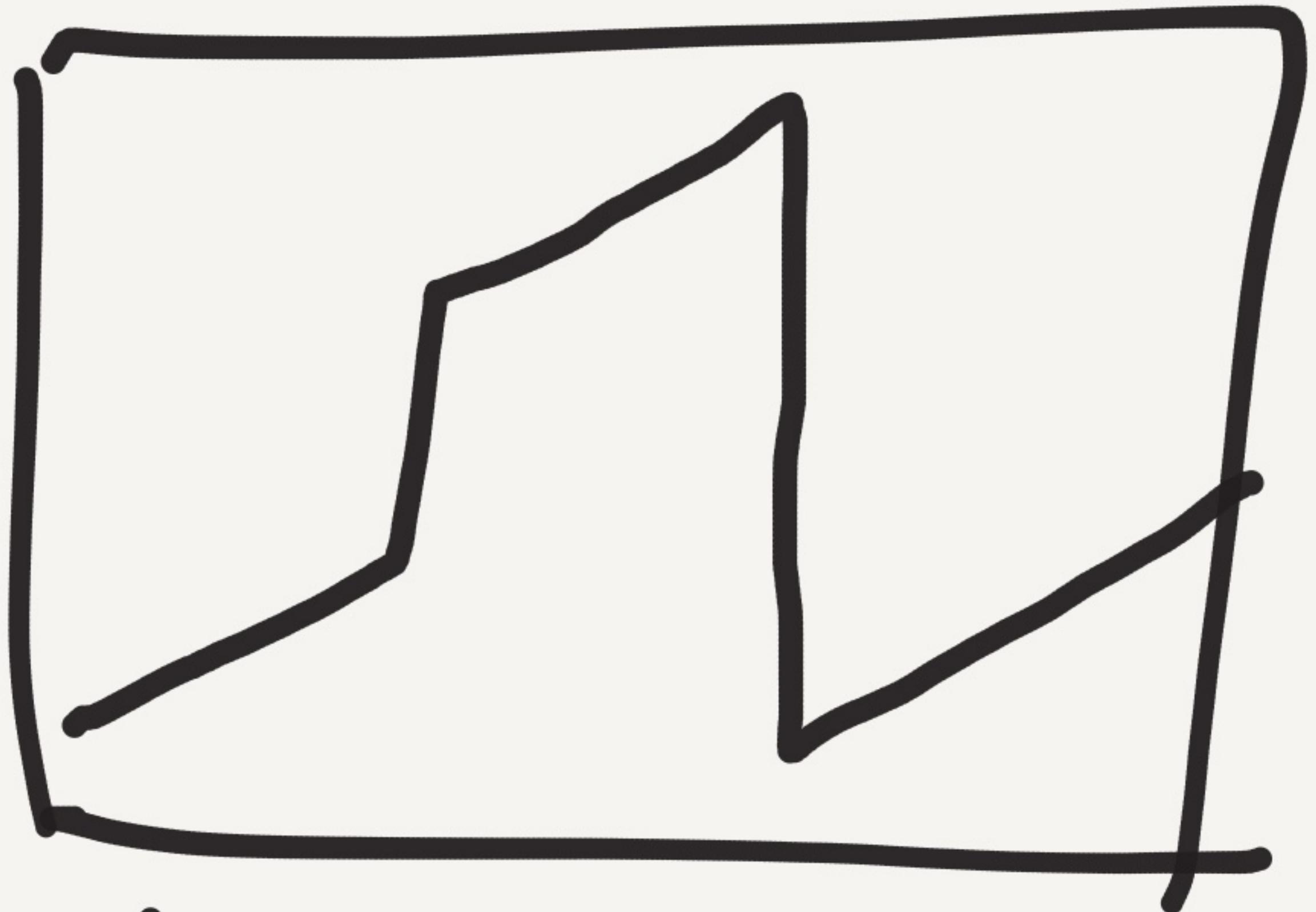
Set

- add(6)
 - read({1,2,3})
 - [cas({1,2,3}, {1,2,3,6})]
- read({1,2,3,...})

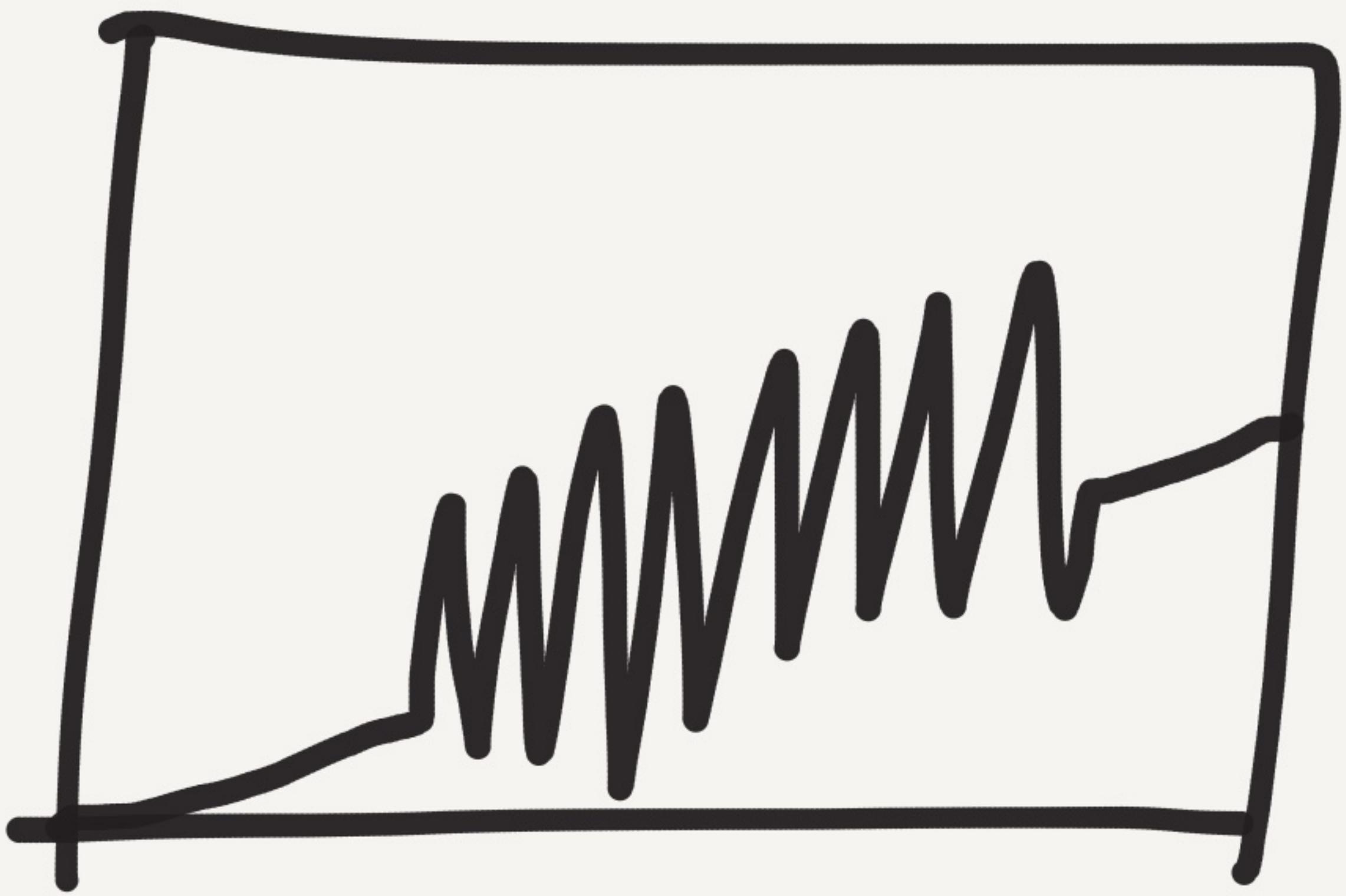
Merkleeyes Race



clock skew



bump



strobe

crashes

n_1 — 

n_2 — 

n_3 — 

n_4 — 

n_5 — 

—

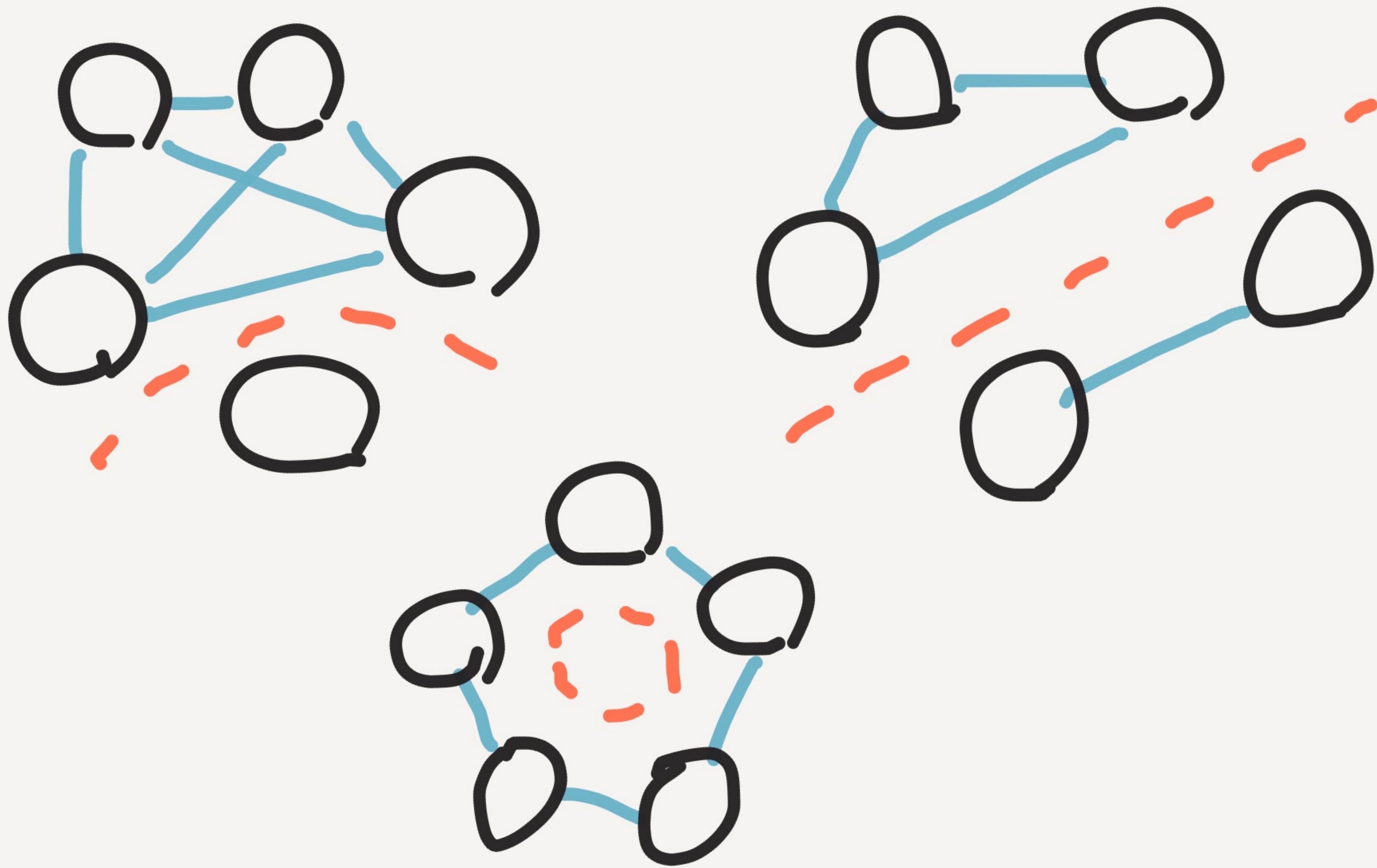
—

—

—

—

Partitions



Duplicate Validators

'A' B²

'A' C²

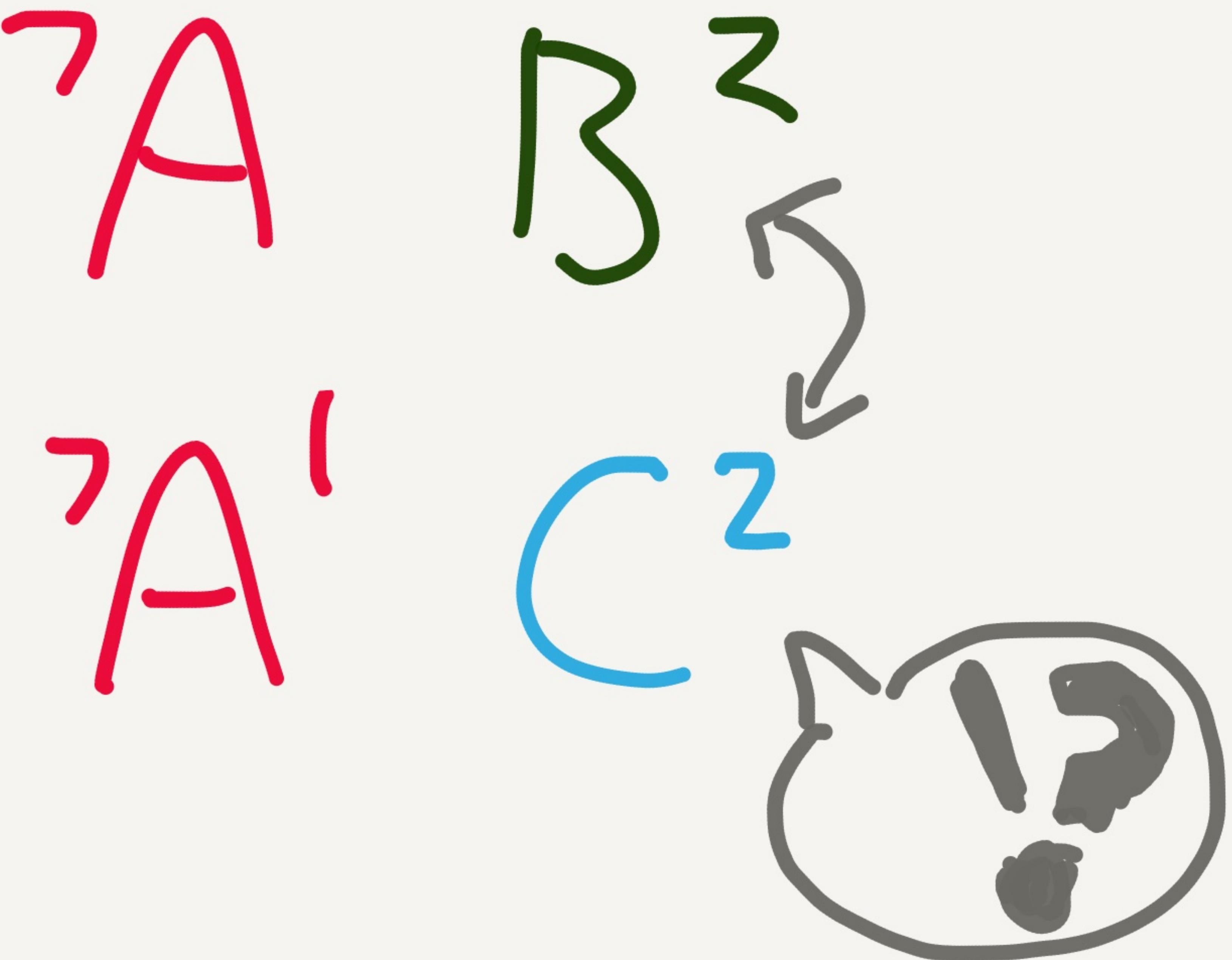
$$7 + 2 + 2 = 11$$

Duplicate Validators

A B %

A' C %

Duplicate Validators



lost

Documents



iff



> / 3

of votes

file truncation



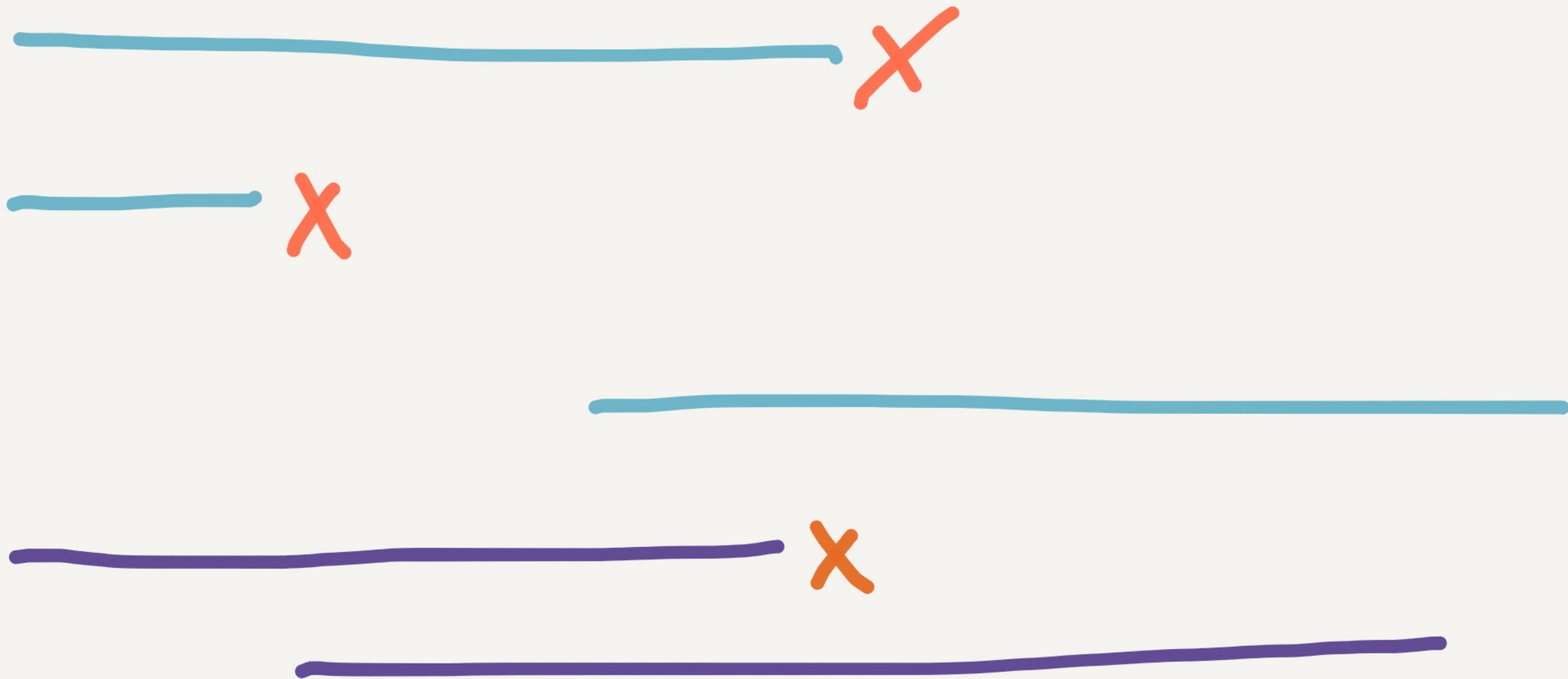
Merkleeyes crash!

(goleveldb bug)

Tendermint

Doesn't fsync to
disk!

Dynamic Reconfig



Data Corruption &

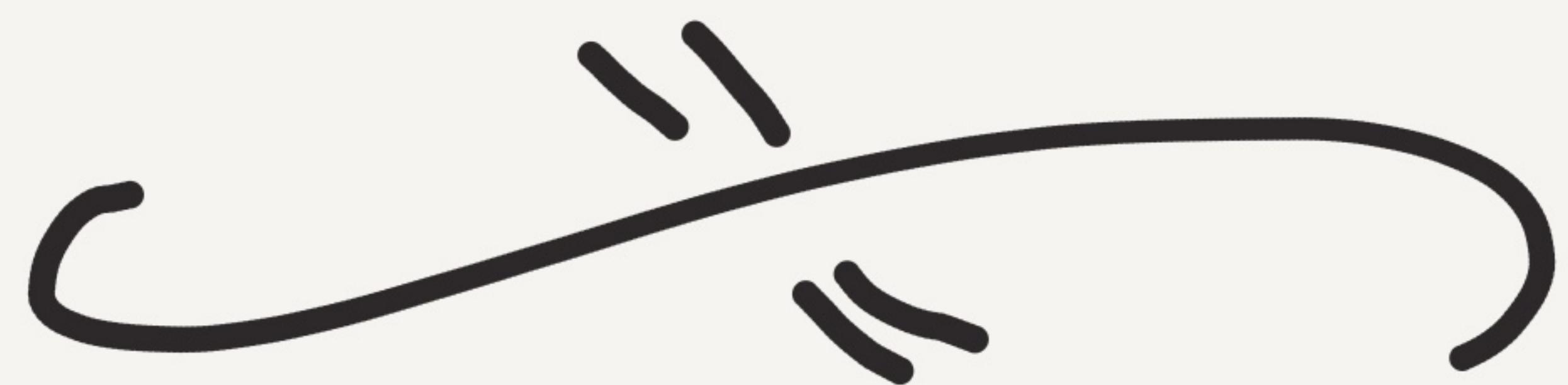
Crashes are to be

expected — Tendermint

was still in beta

Team working to

fix these issues





hazelcast

3.8.3

In-memory



Sets

Lists

IDs

Semaphores

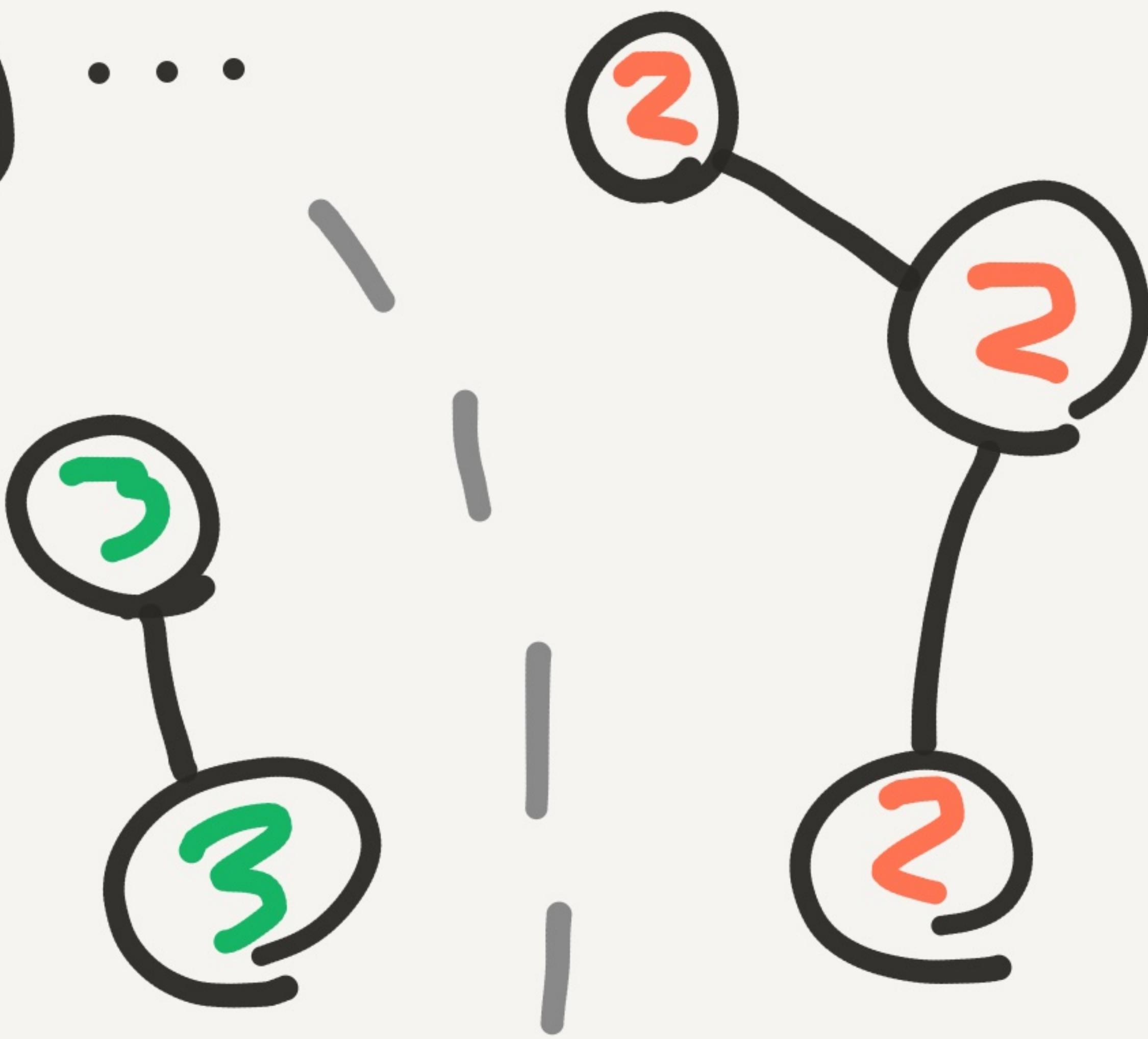
Queues

Maps

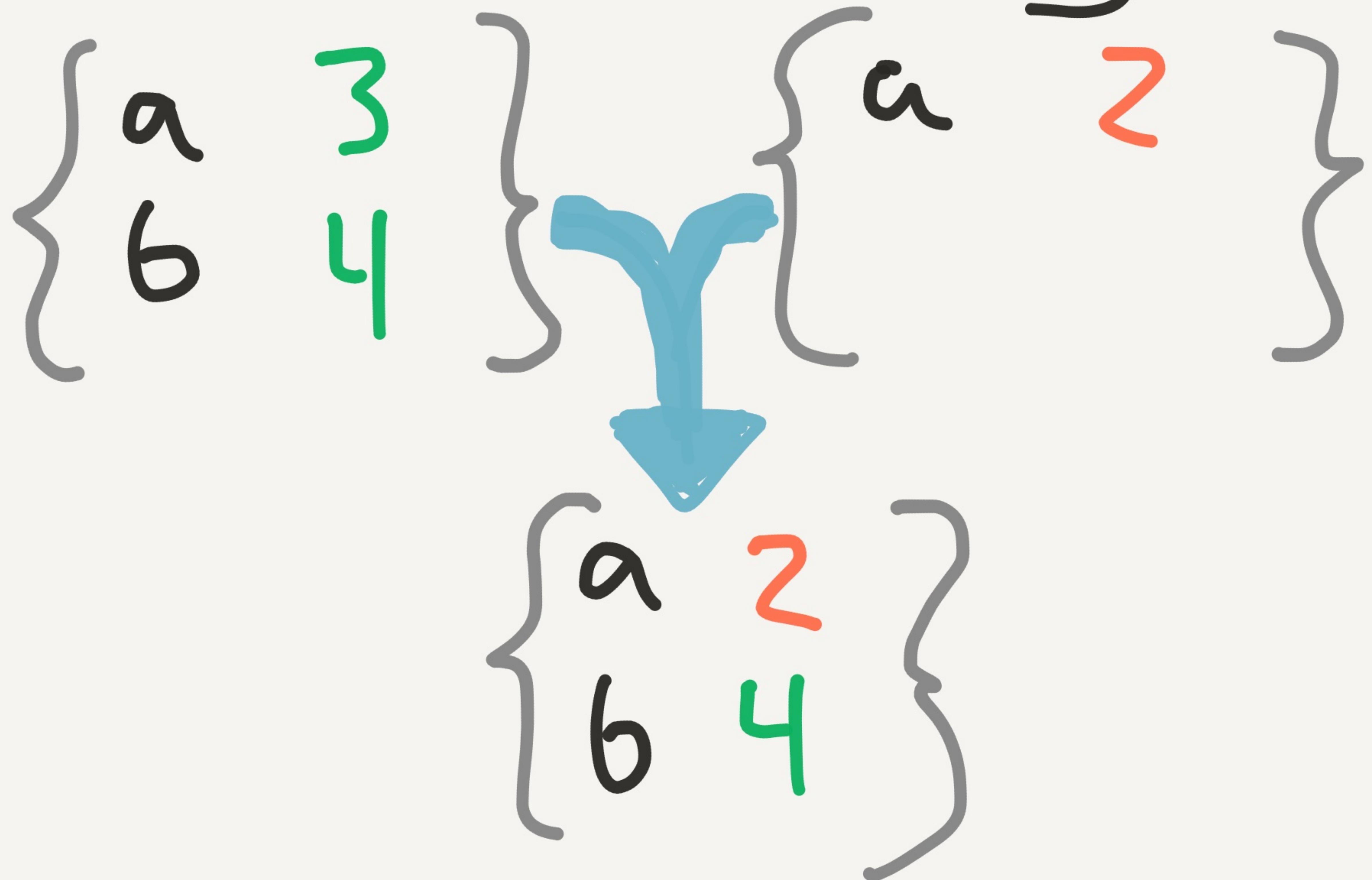
Locks

Atomic Longs

But in a partition,
Components run
independently ...

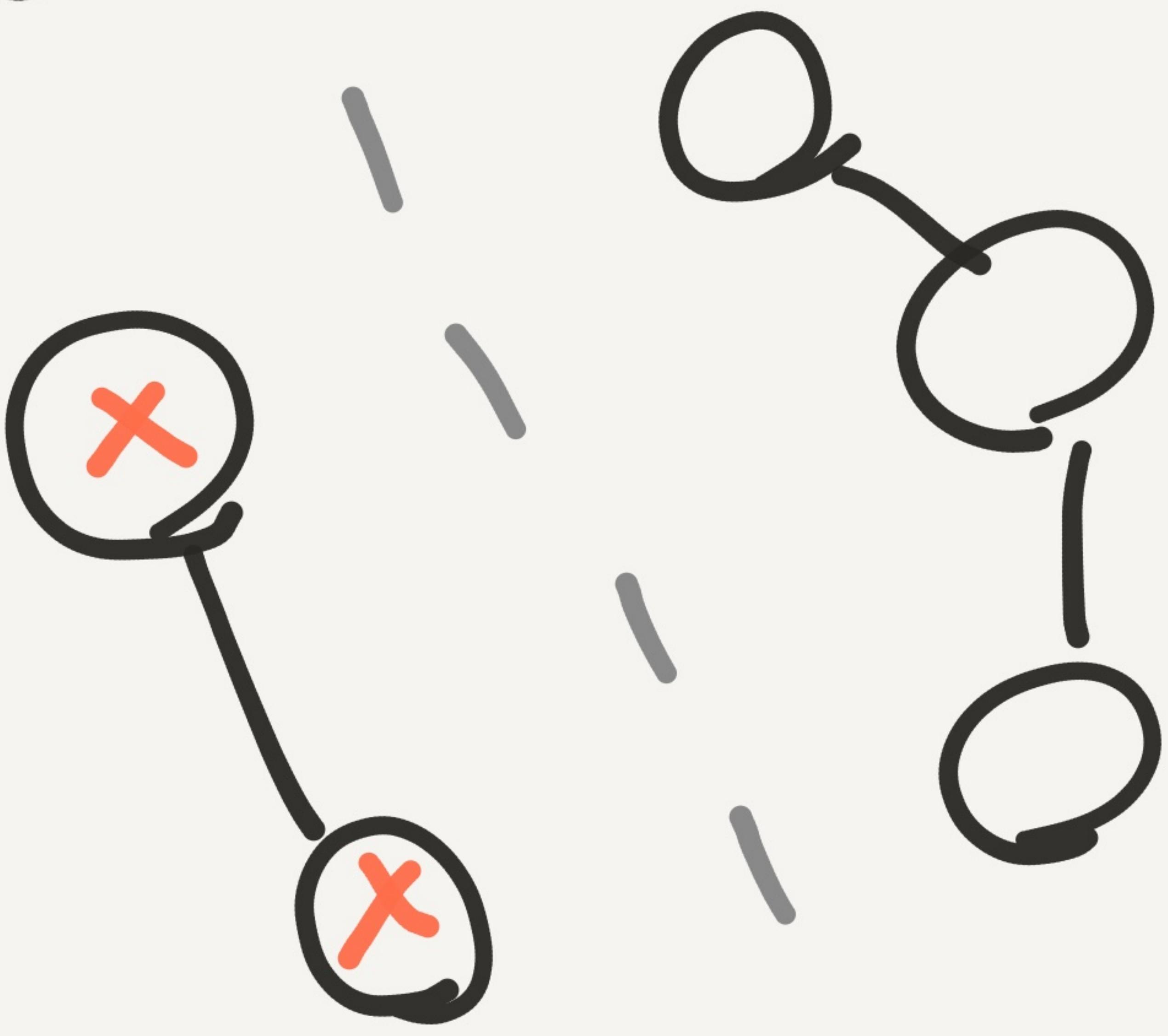


Maps have merge



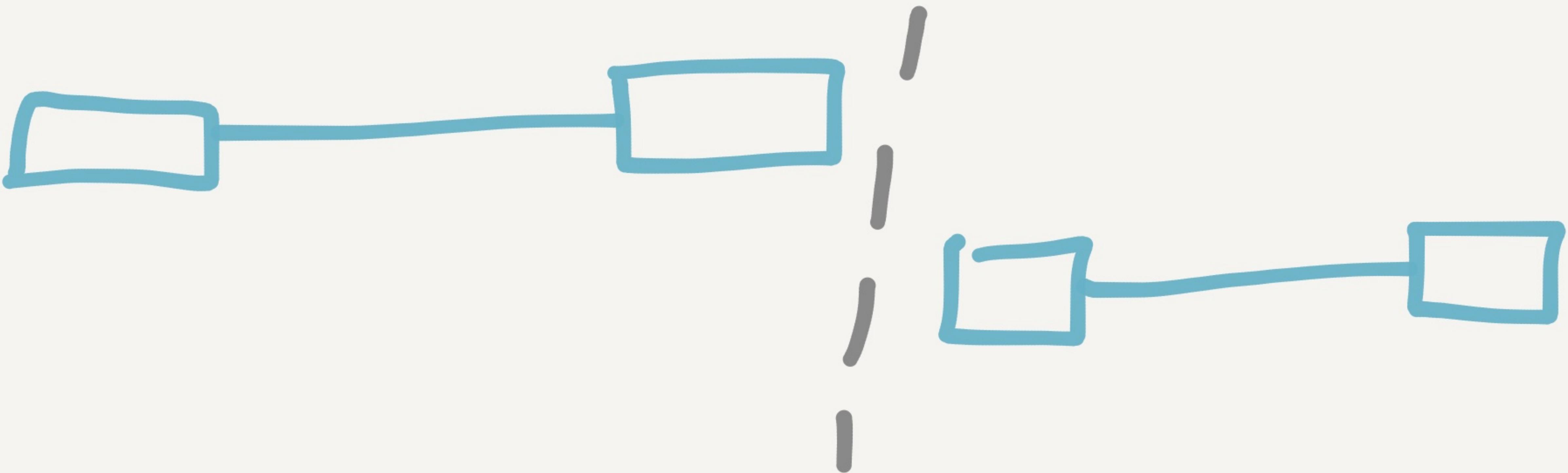
Split-Brain Protection

(10^+ seconds)



Locks

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



acquire

held

acquire

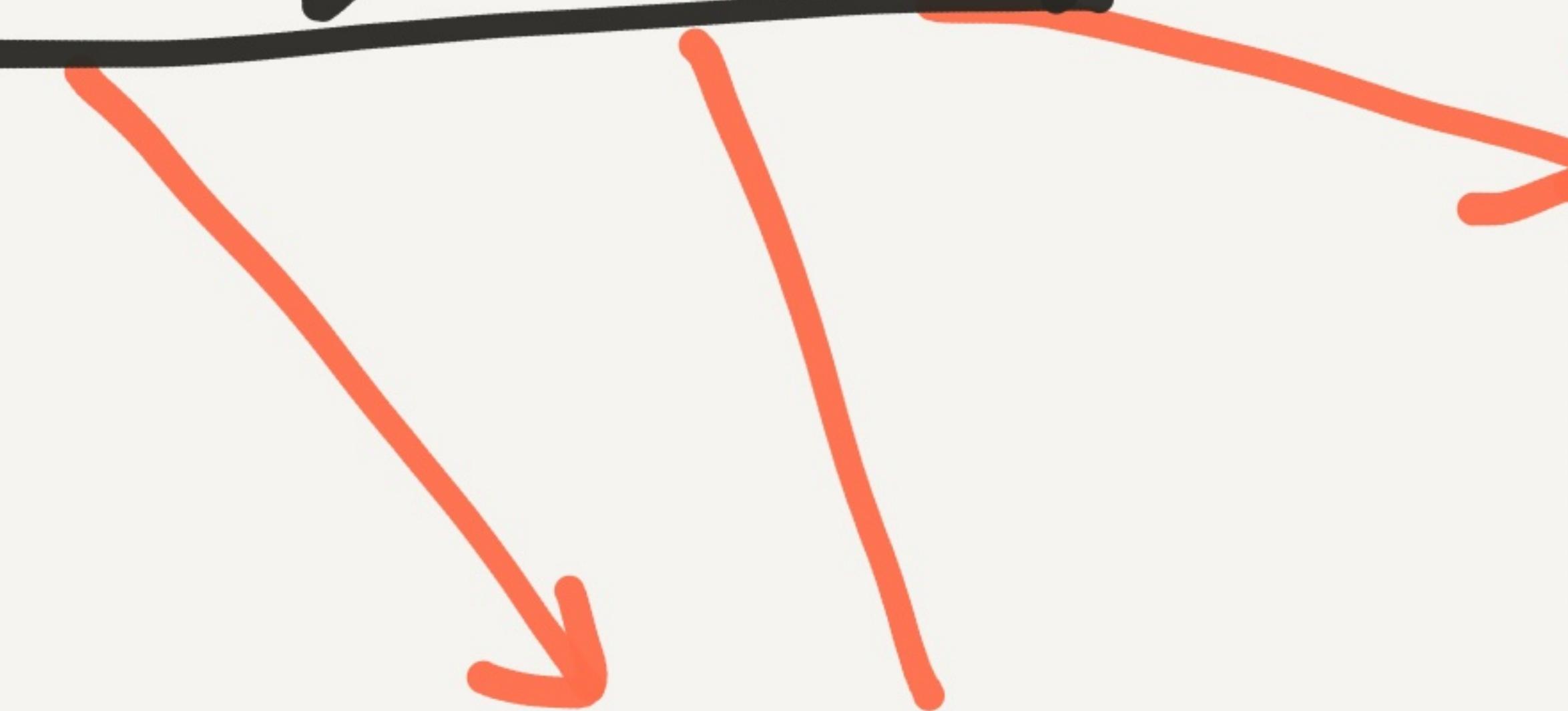
acquire

acquire

acquire

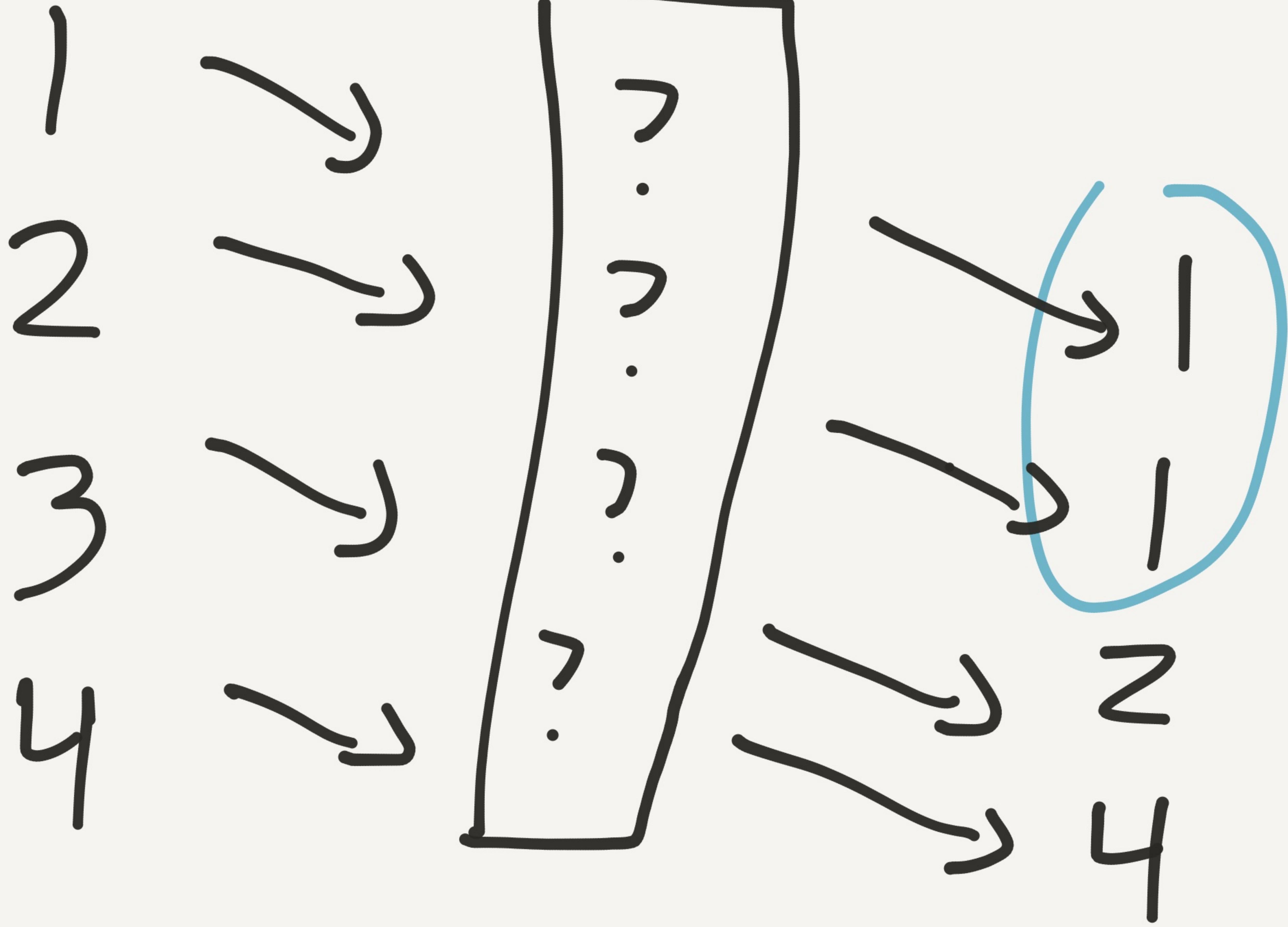
acquire

acquire



Queues

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



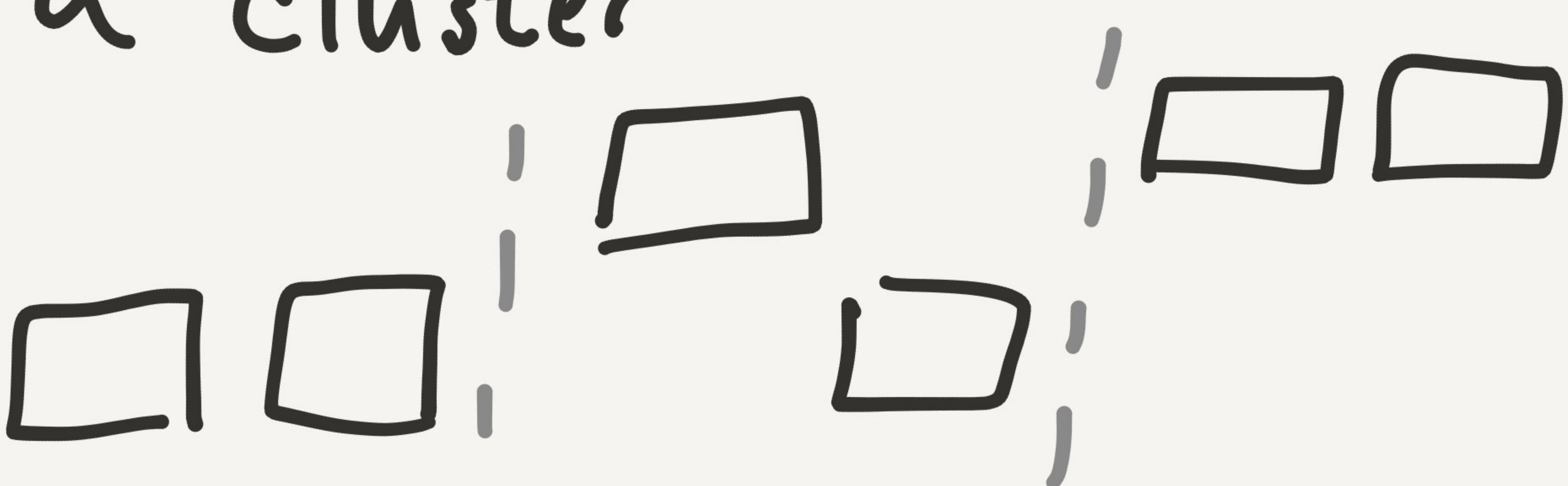
$\sim 2\%$ duplicated

$\sim 2\%$ lost

worse with default
timeouts

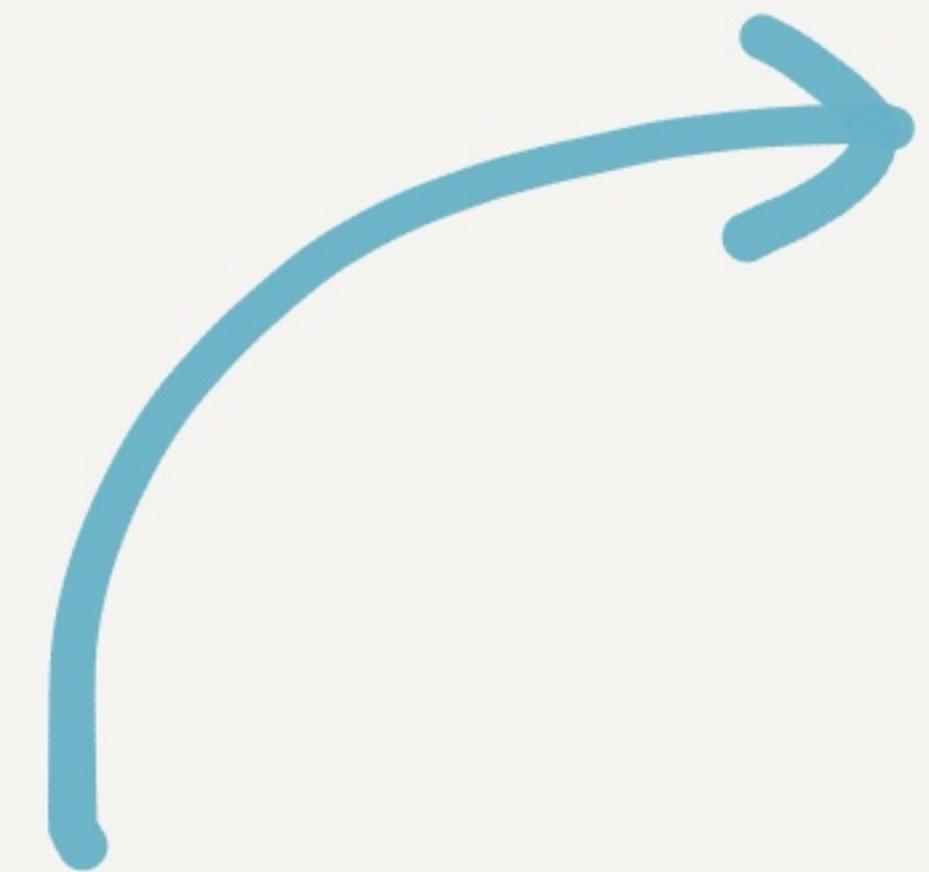
Atomic Reference

"Guaranteed atomic
Compare and set across
a cluster"



increment

$$2 + 1 = 3$$

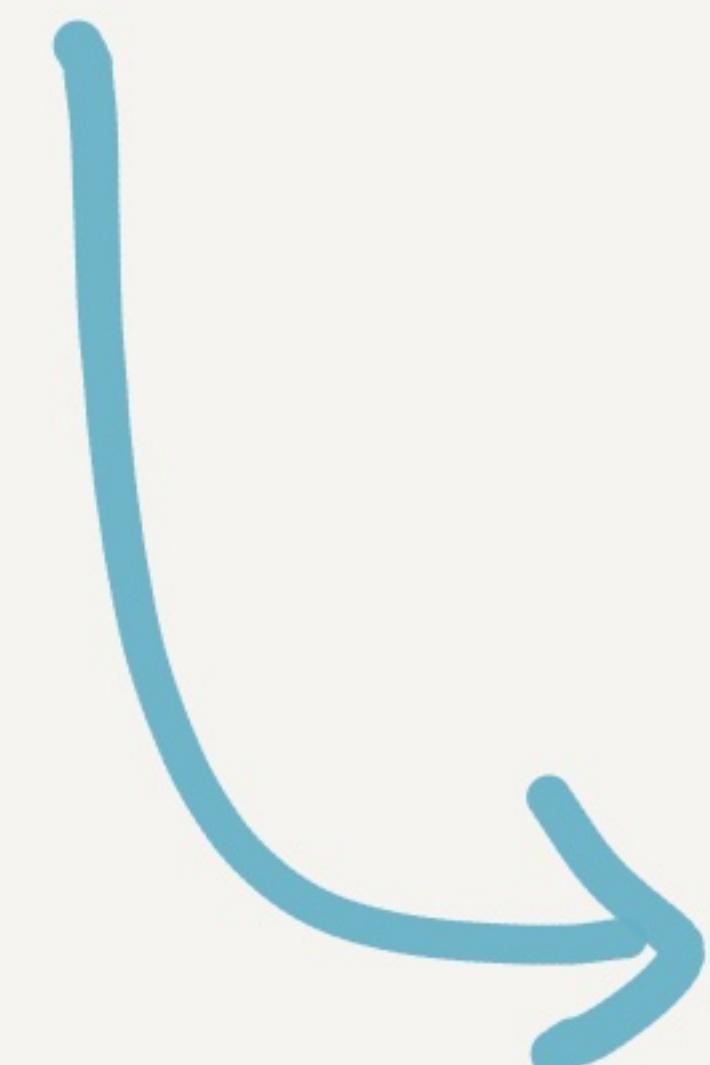


"still 2?"
"yes!"

-2



3



increment

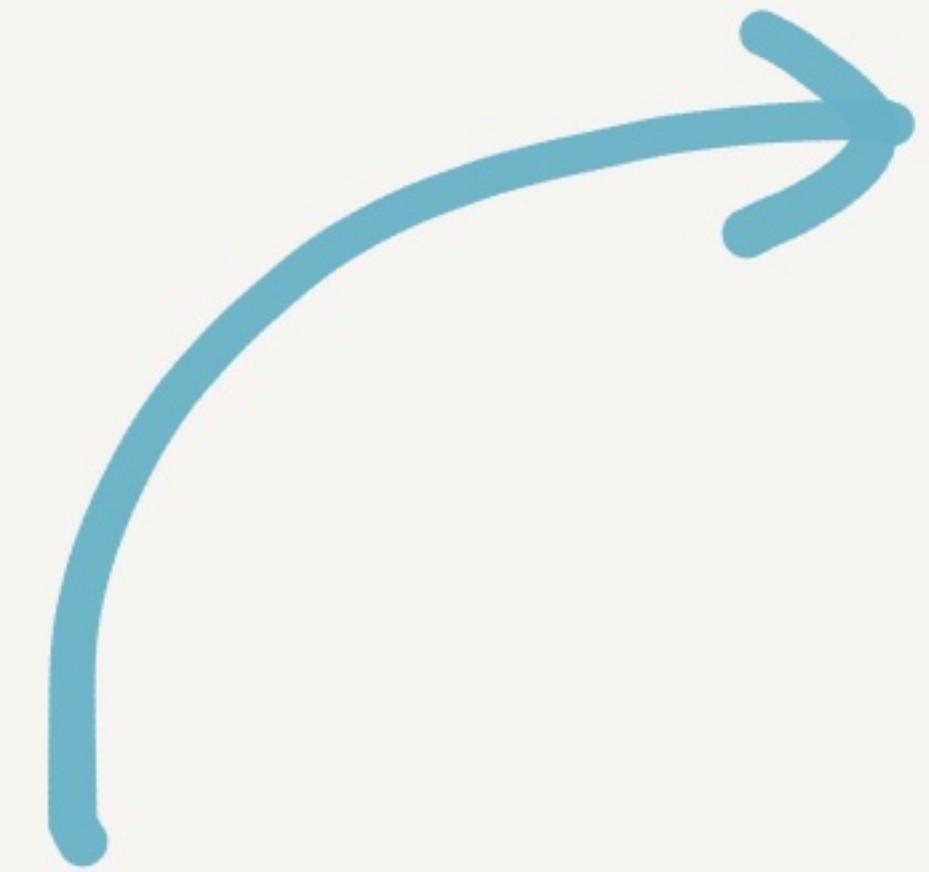
$$2 + 1 = 3$$



"still 2?"
"nope!"

increment

$$2 + 1 = 3$$



"still 2?"
"yes!"

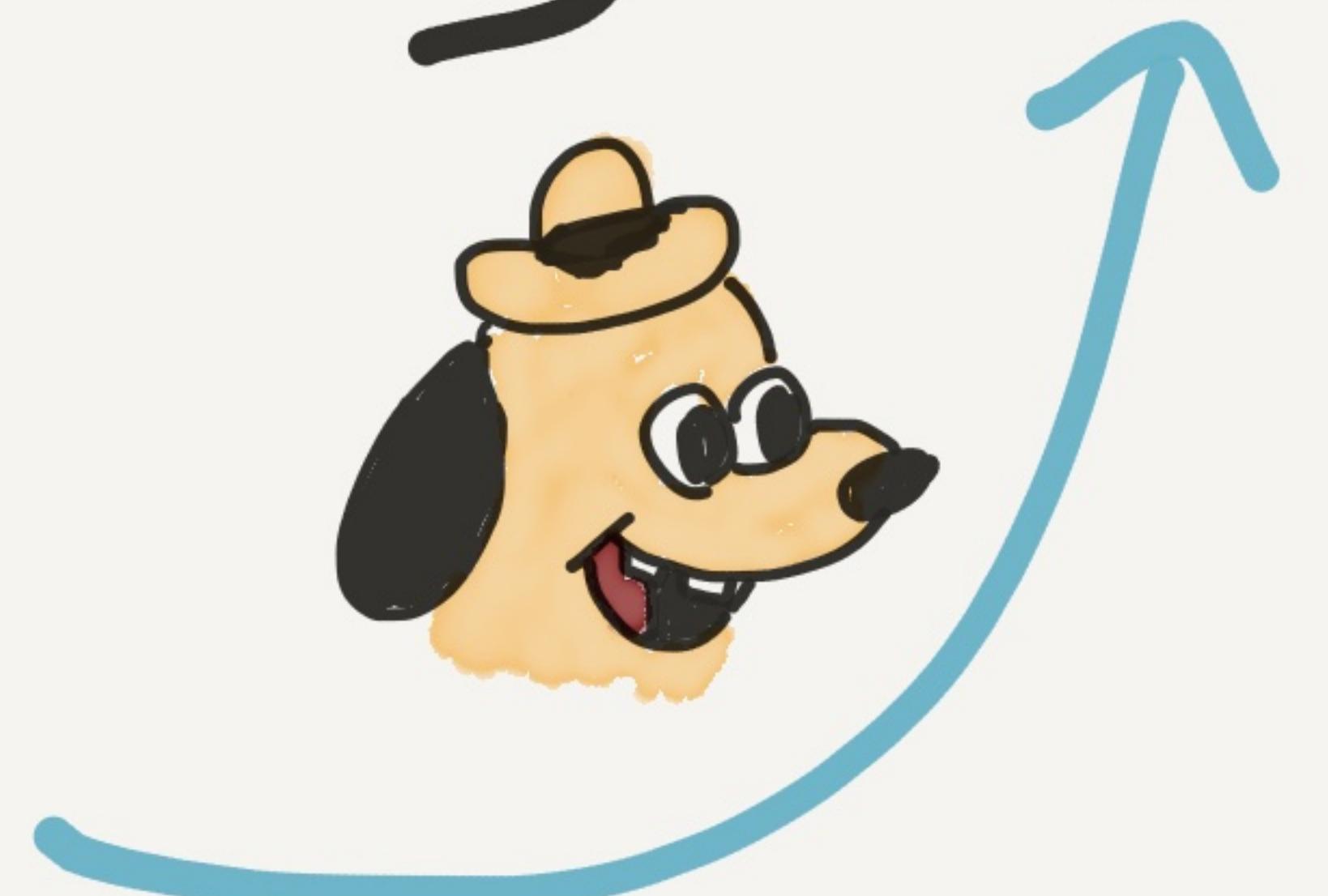
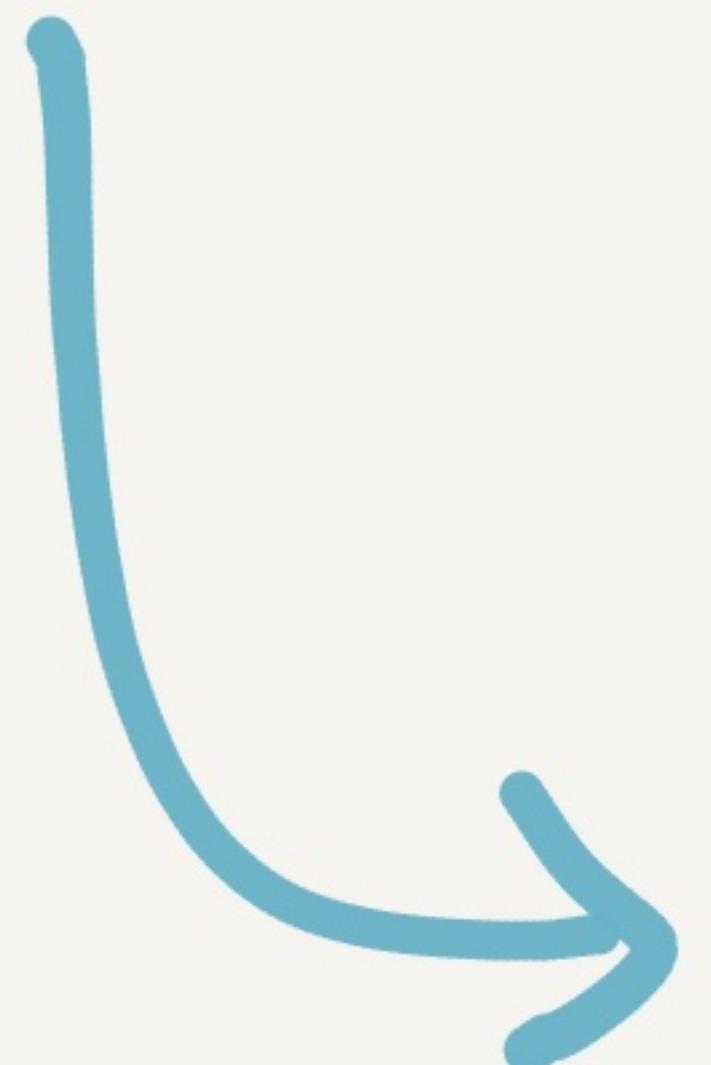
$$-2$$

$$\xrightarrow{\hspace{1cm}} 3 - 3$$



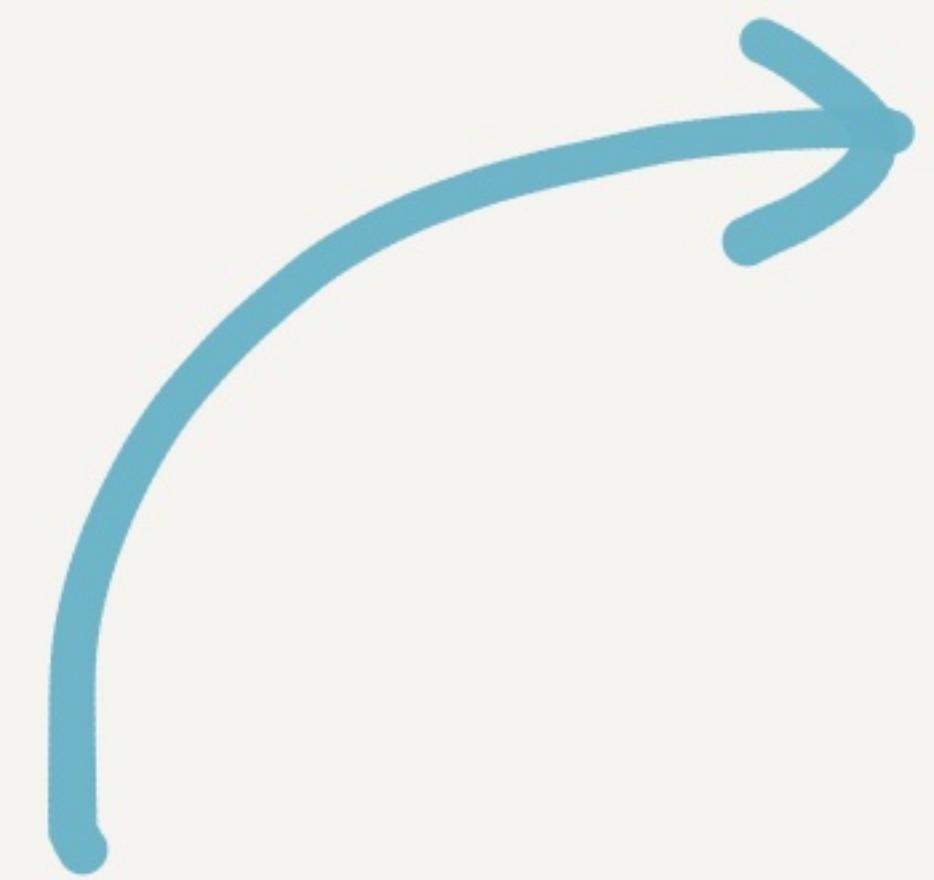
increment

$$2 + 1 = 3$$



increment

$$2 + 1 = 3$$



"still 2?"
"yes!"

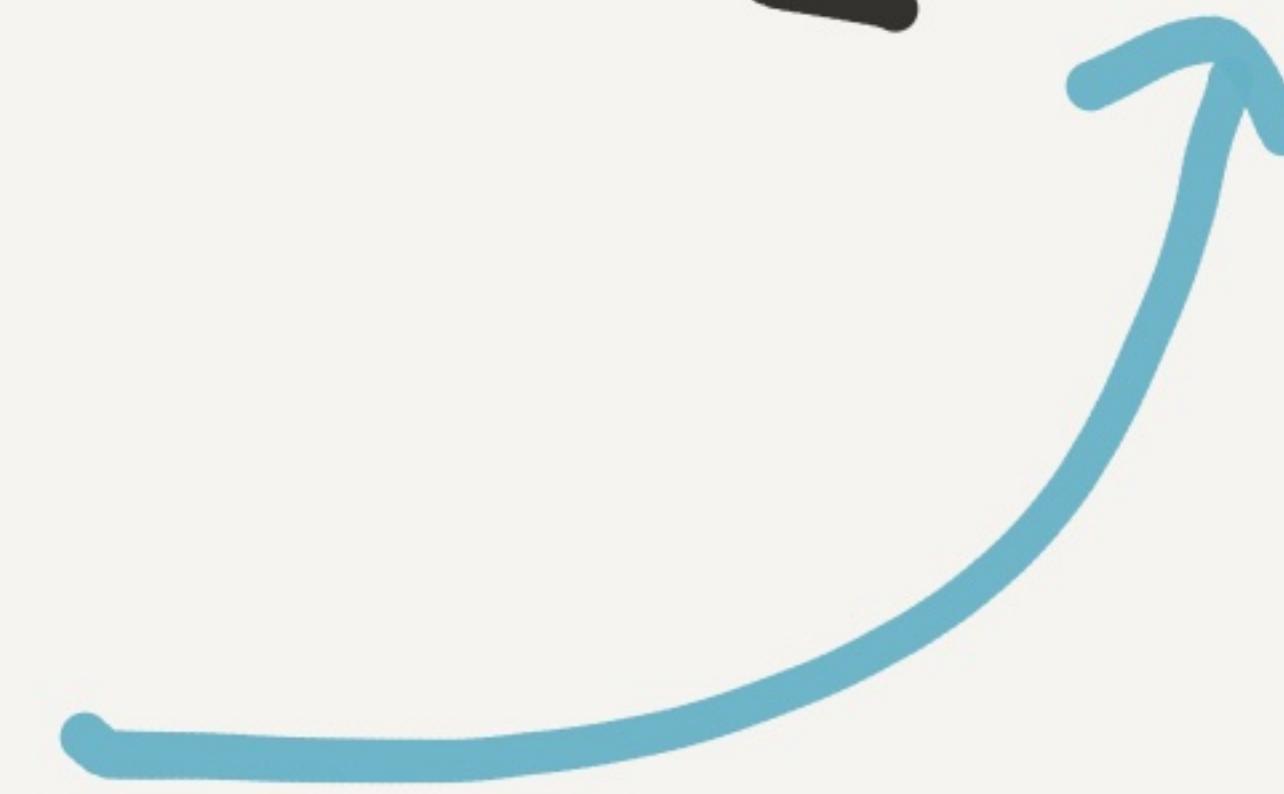
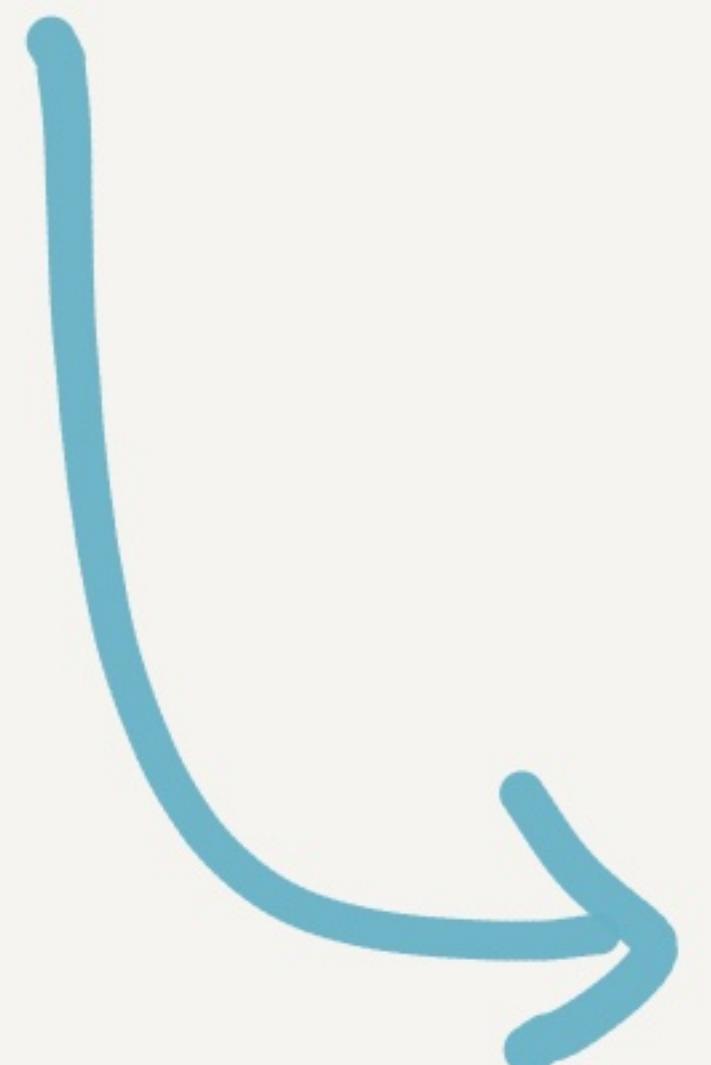
$$-2$$

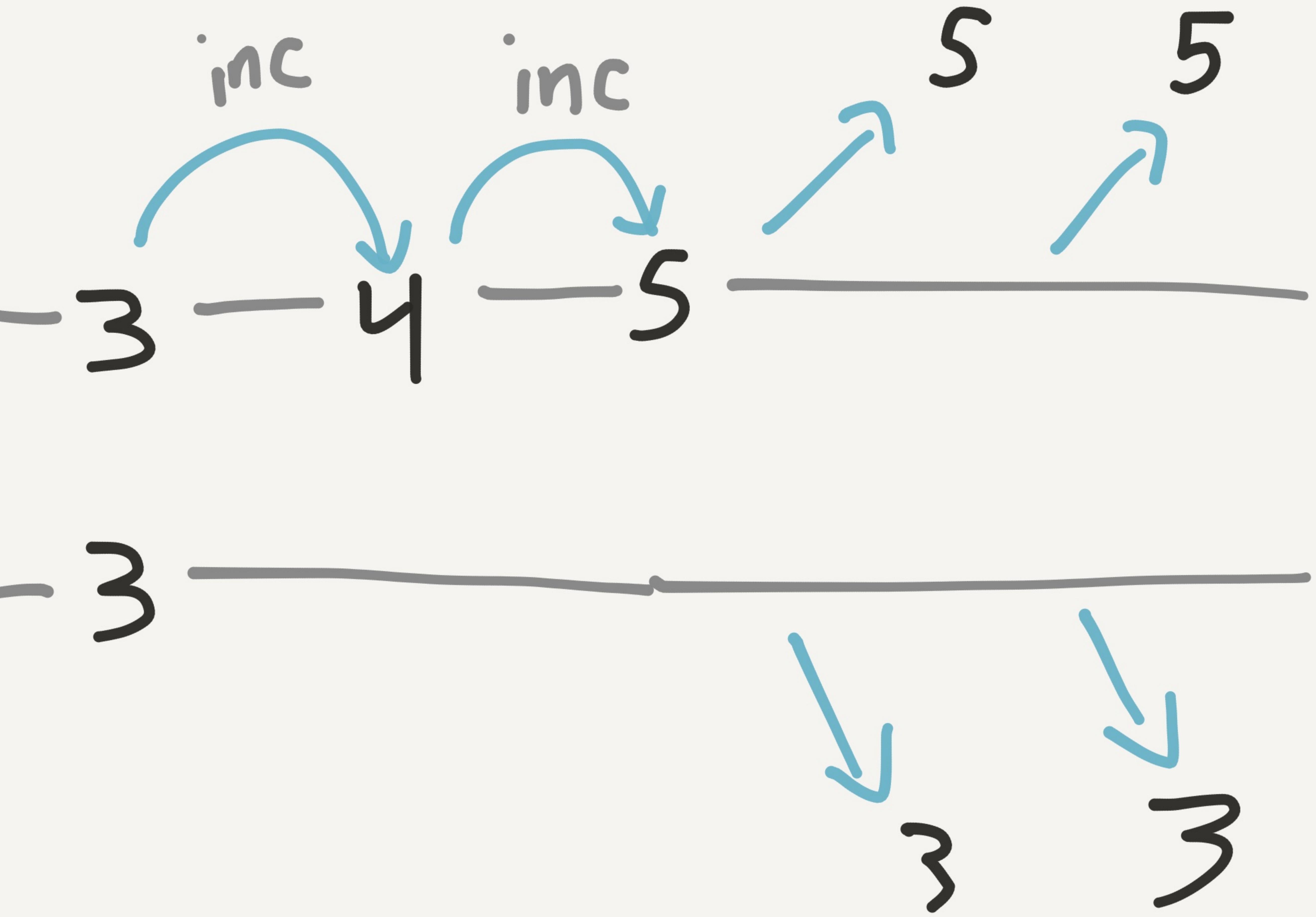


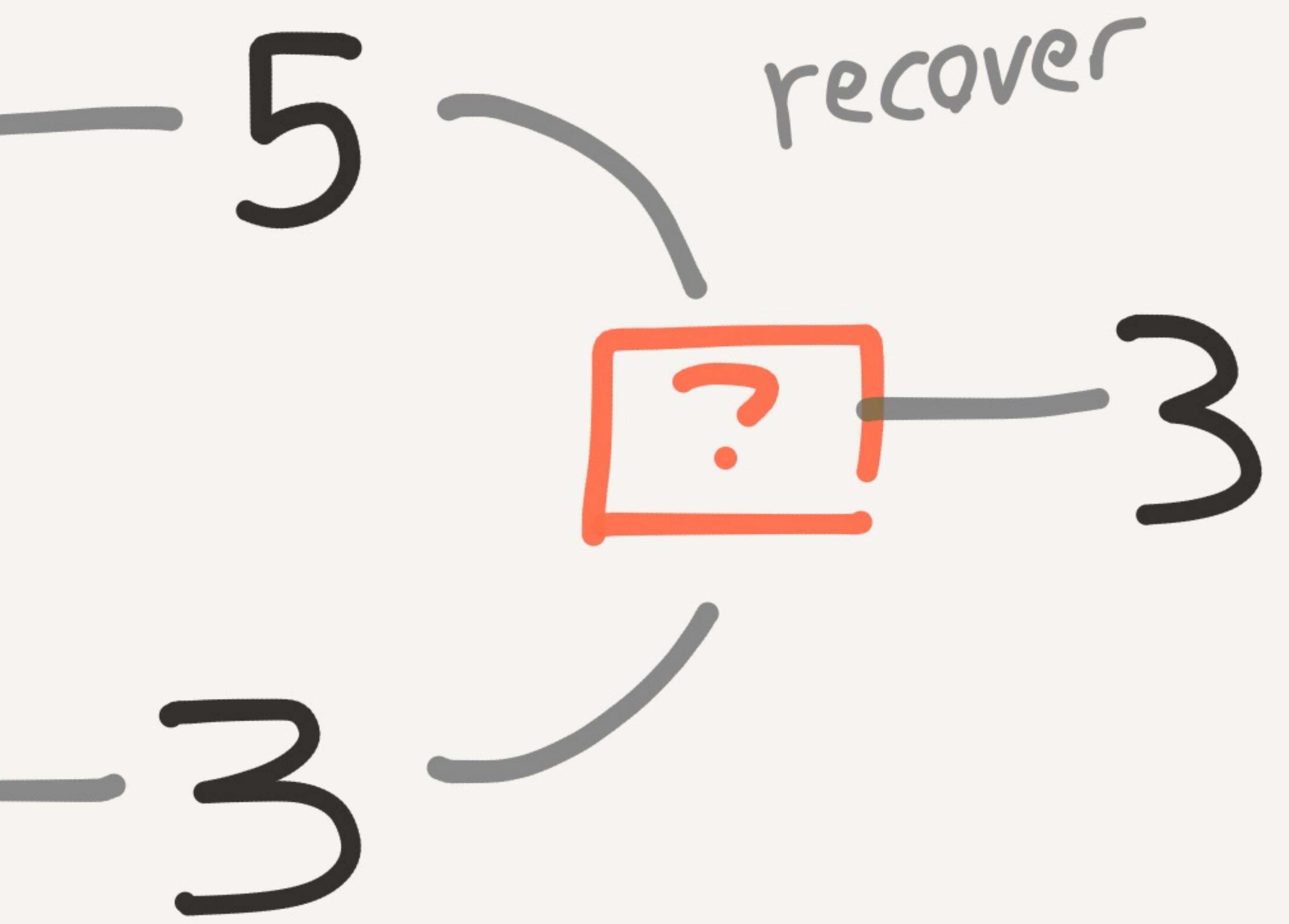
$$2 - 3$$

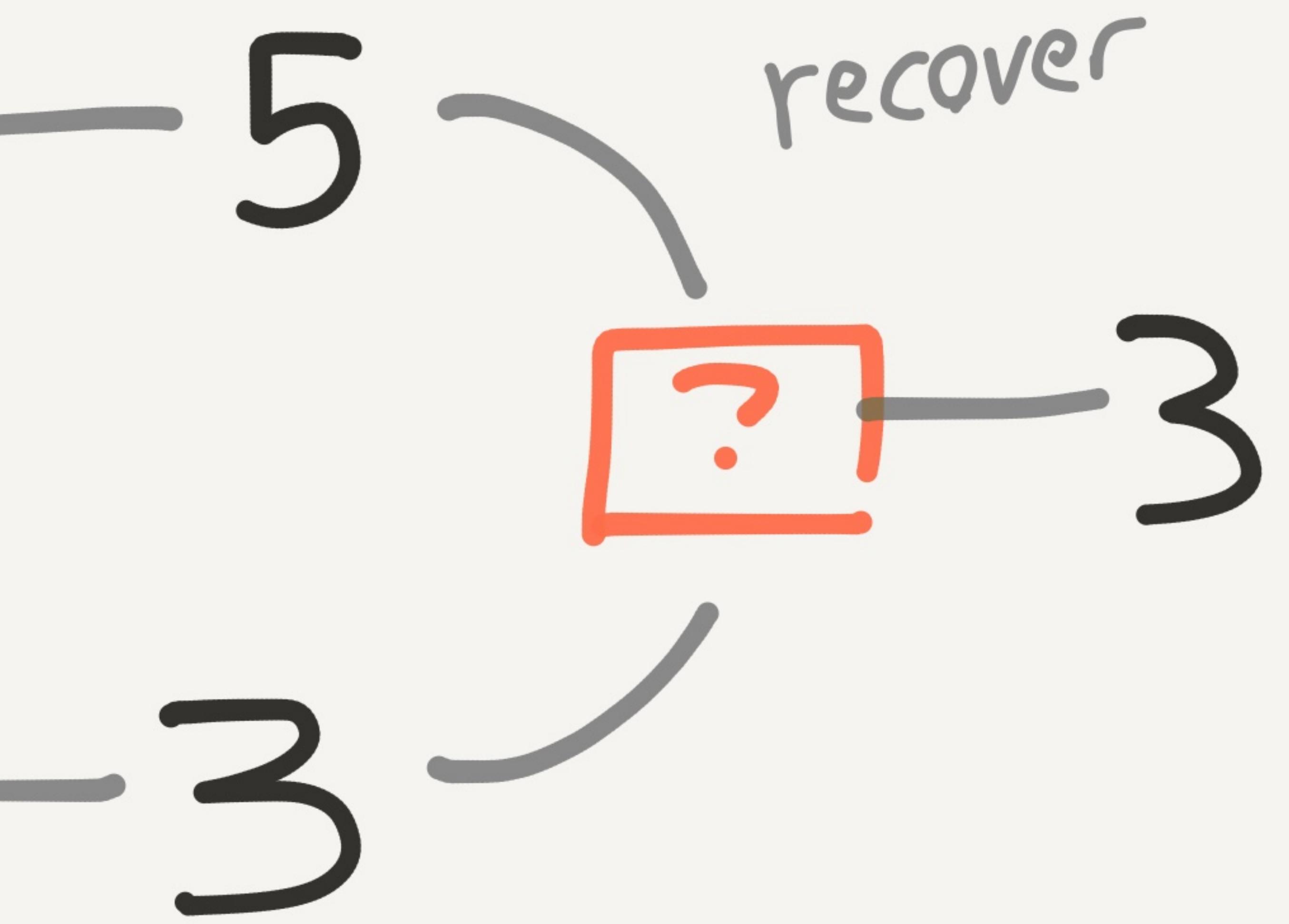
increment

$$2 + 1 = 3$$









—5 —3

recover

?

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

Hazelcast added flake

IDs, PN counters in

3.10 (April 2018)

KEROSPIKE

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)

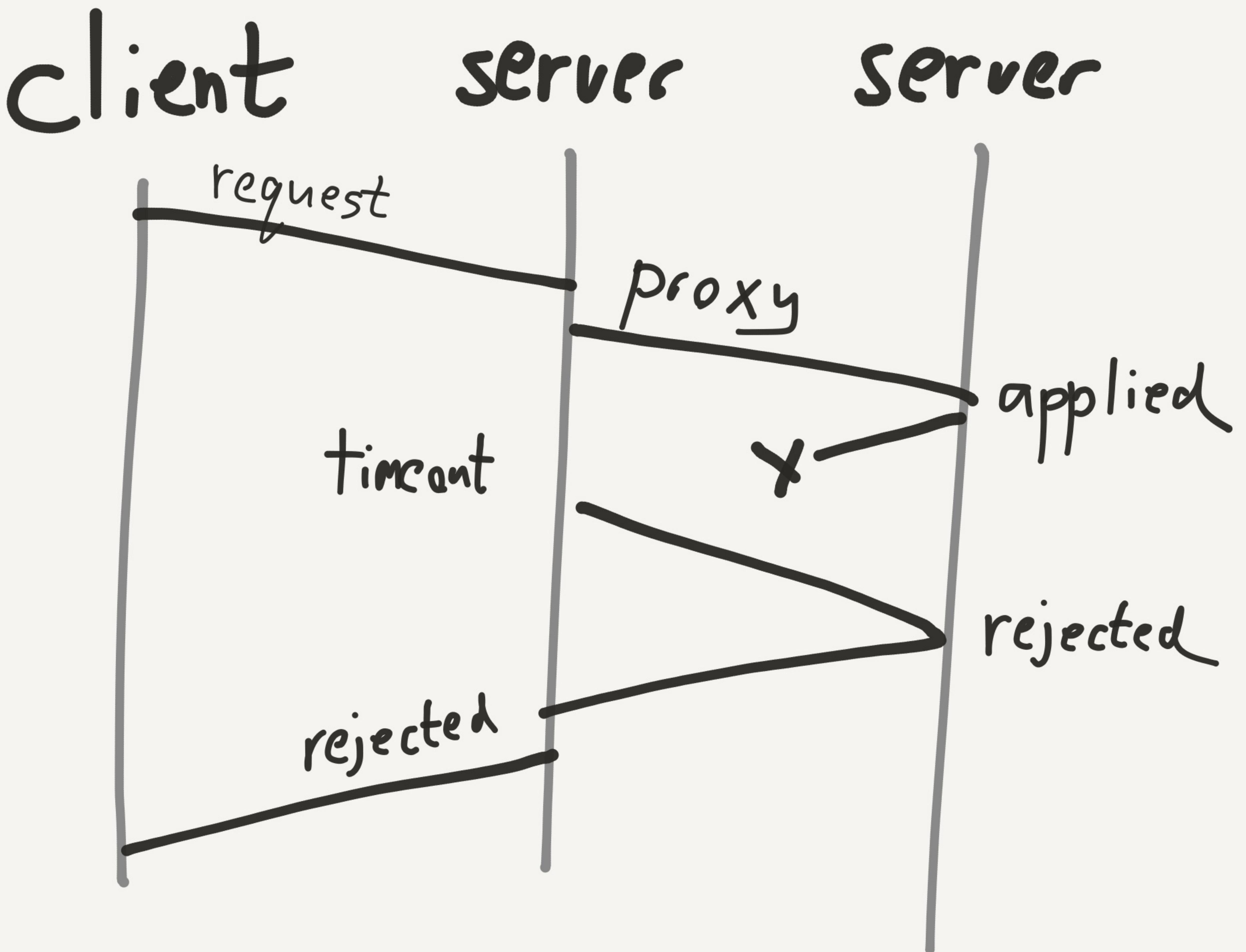


fail

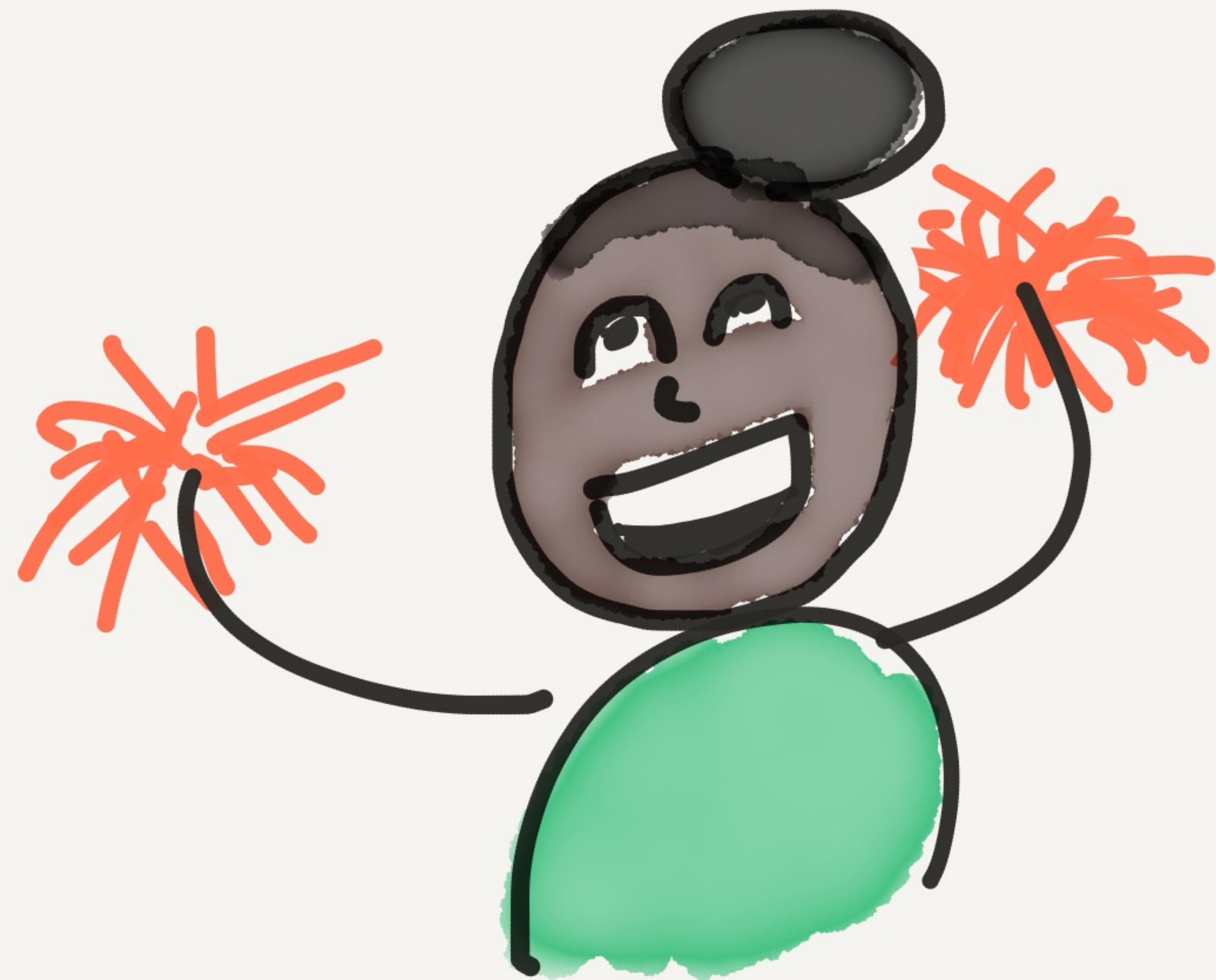


ok





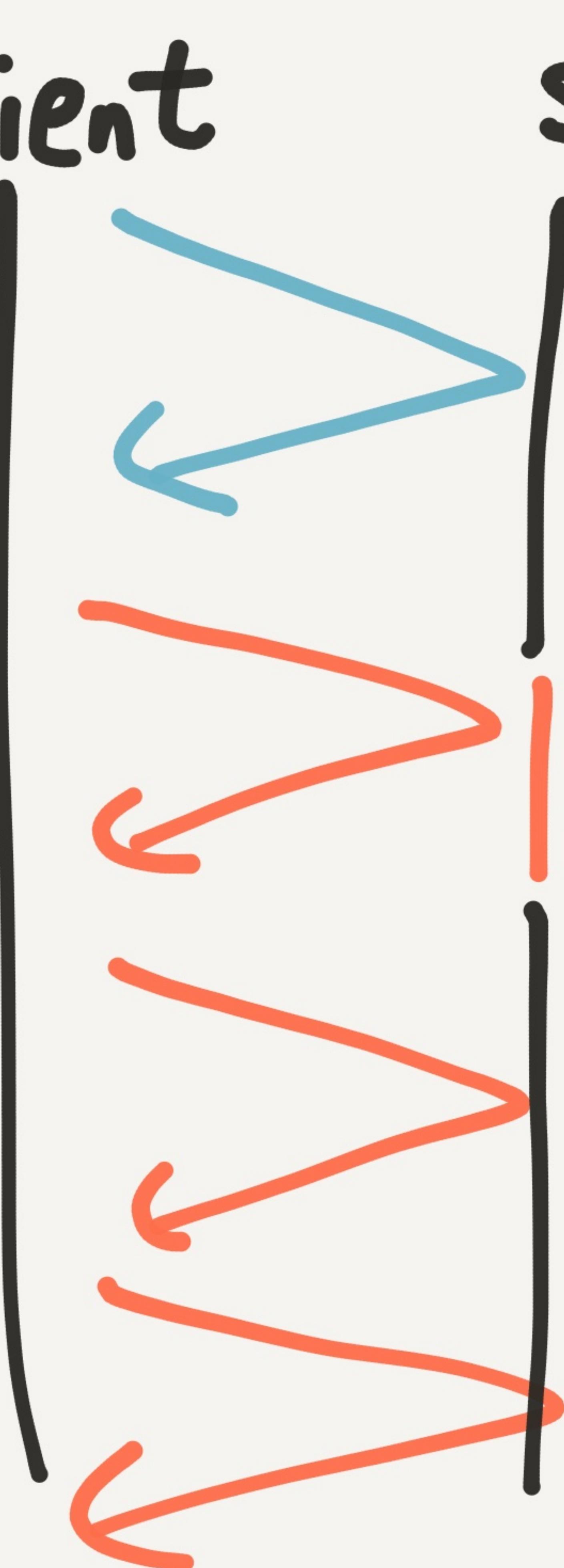
fixed in 3.99.1.5



Node crashes



client



s



s



s



s



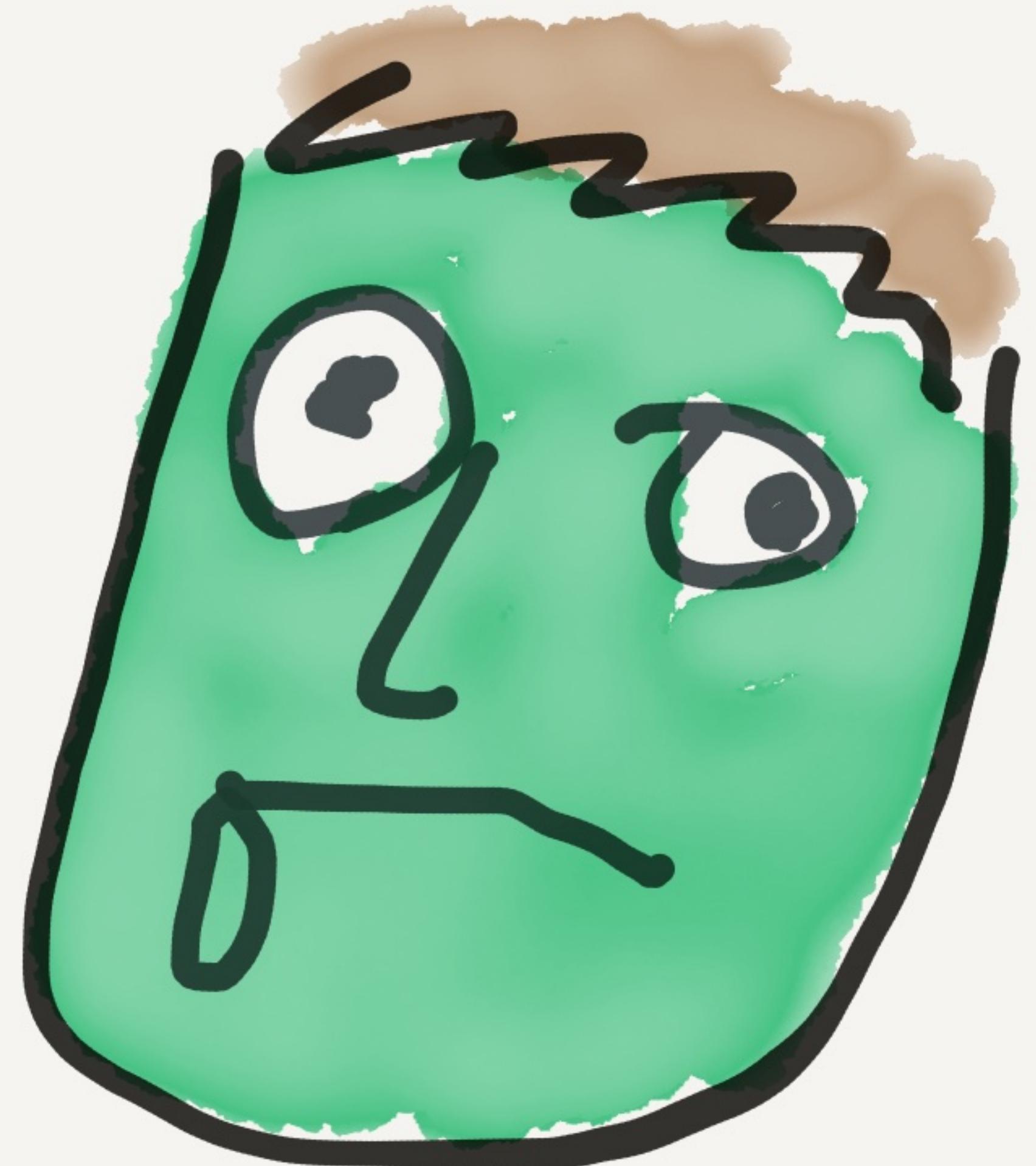
[press E to revive]



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

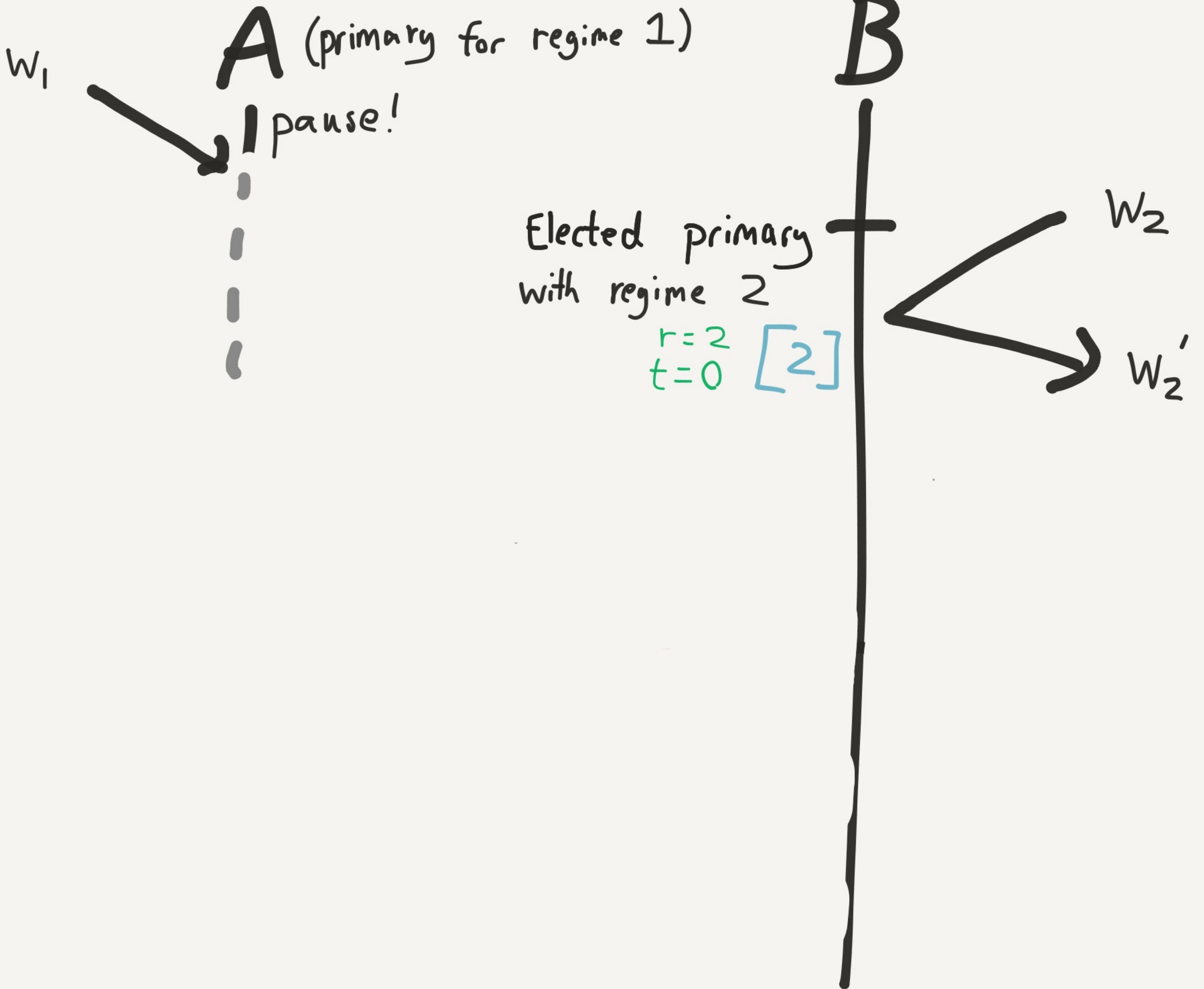
|

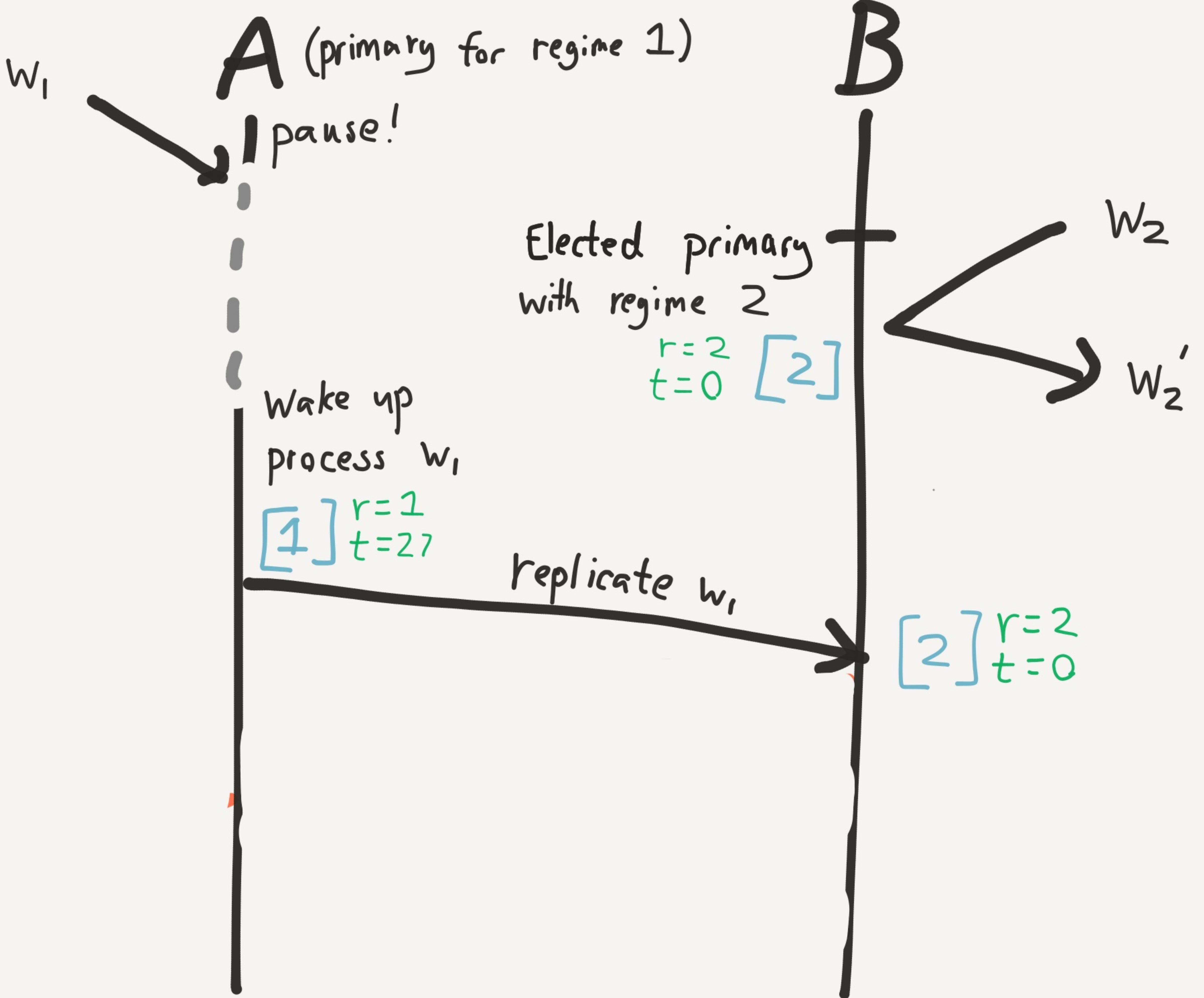
+

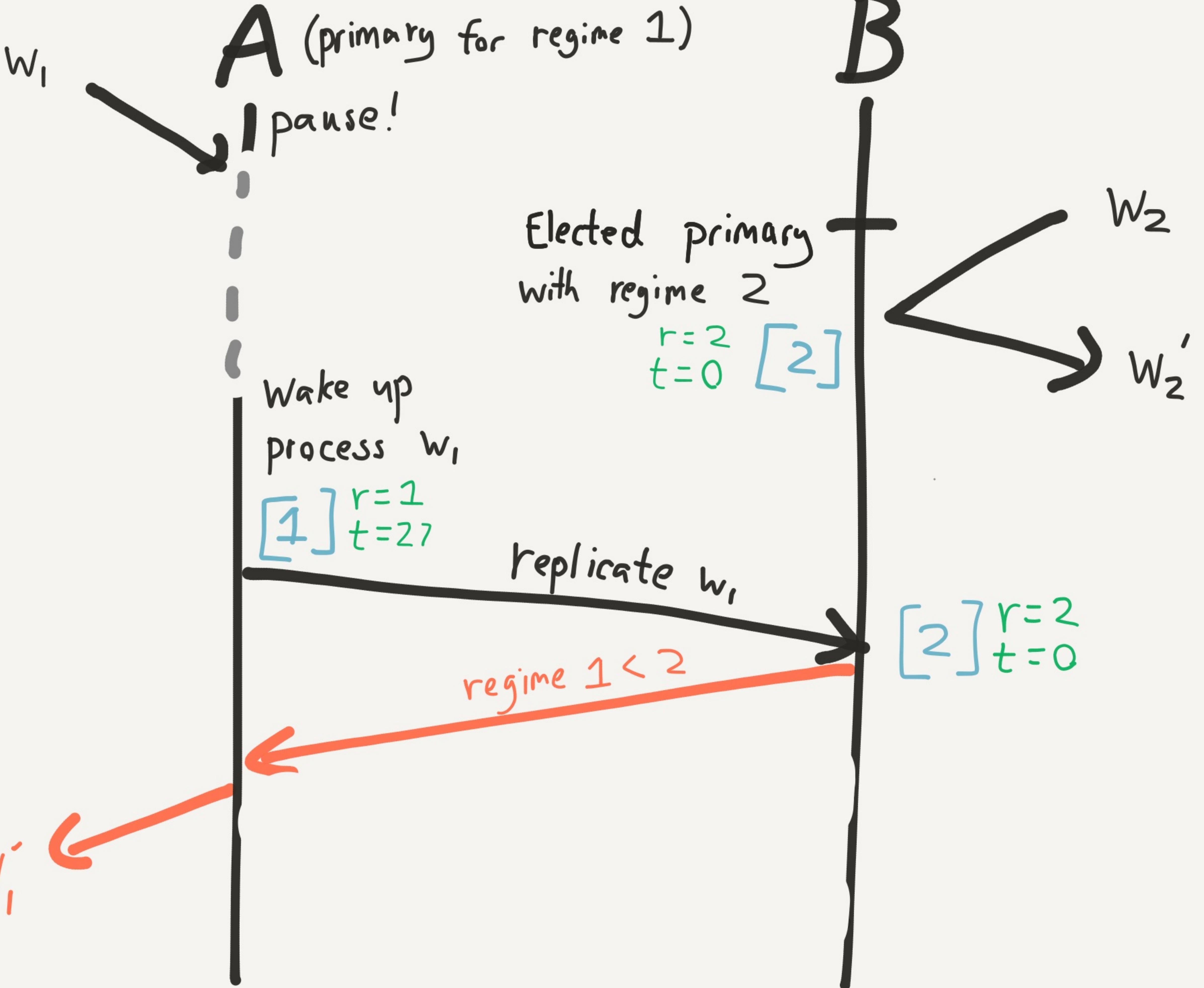
|

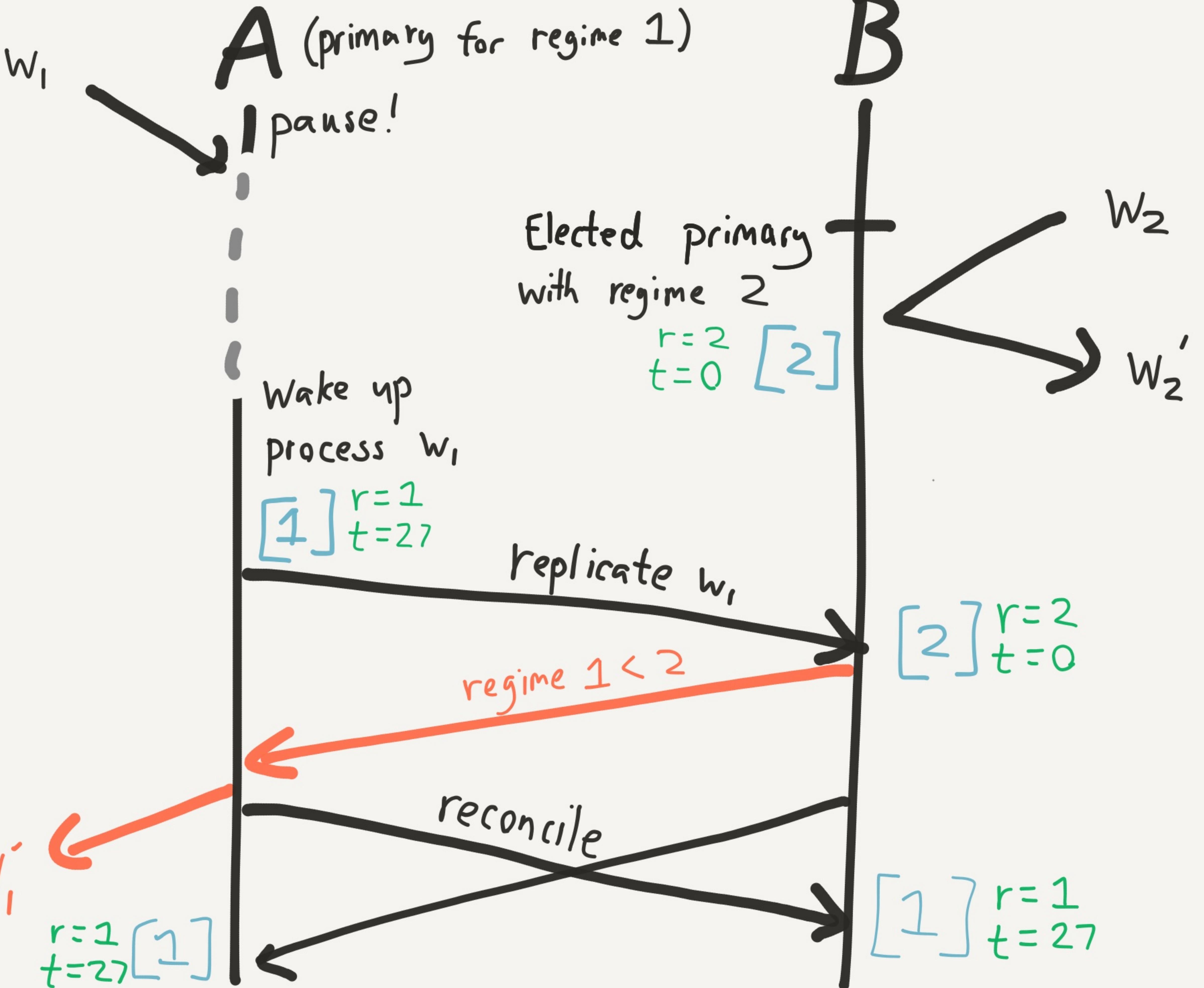
|

|









Beth clock skew ≠

hot using fSync were

Known failure modes

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

looks linearizable!

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

future work!

4.0.0 went live on

March 7, 2018

- SC is a commercial
feature

Recap



Read the docs!



CAREFULLY

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

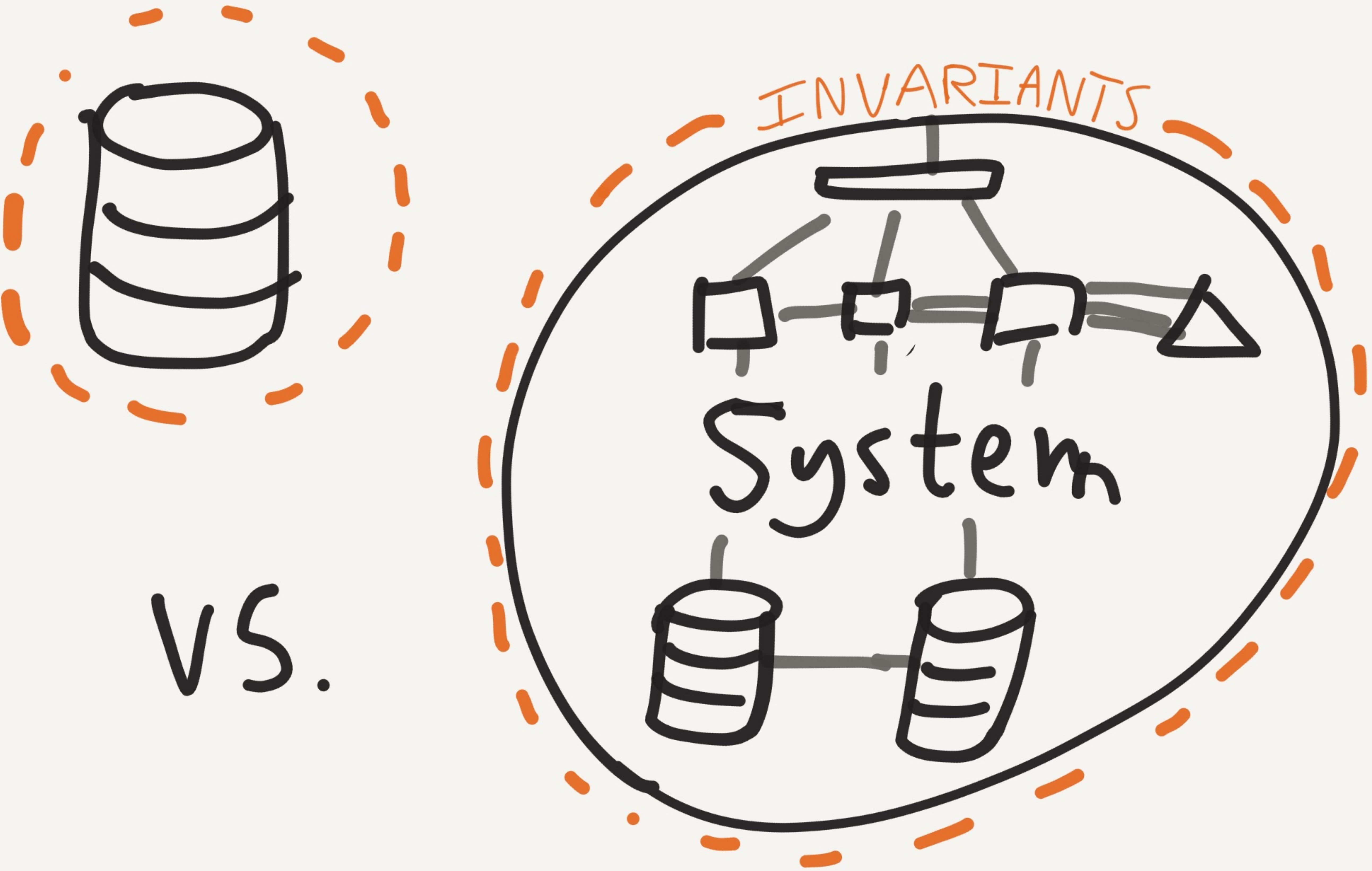
Network Partition

```
# iptables -j DROP  
# 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 Sukhoreslov

Aerospike



Peter Alvara



<https://jepsen.io>