

Jeppesen



a fsyncing
feeling

Kyle Kingstury
@Aphyr

I break
databases!



Public
API {



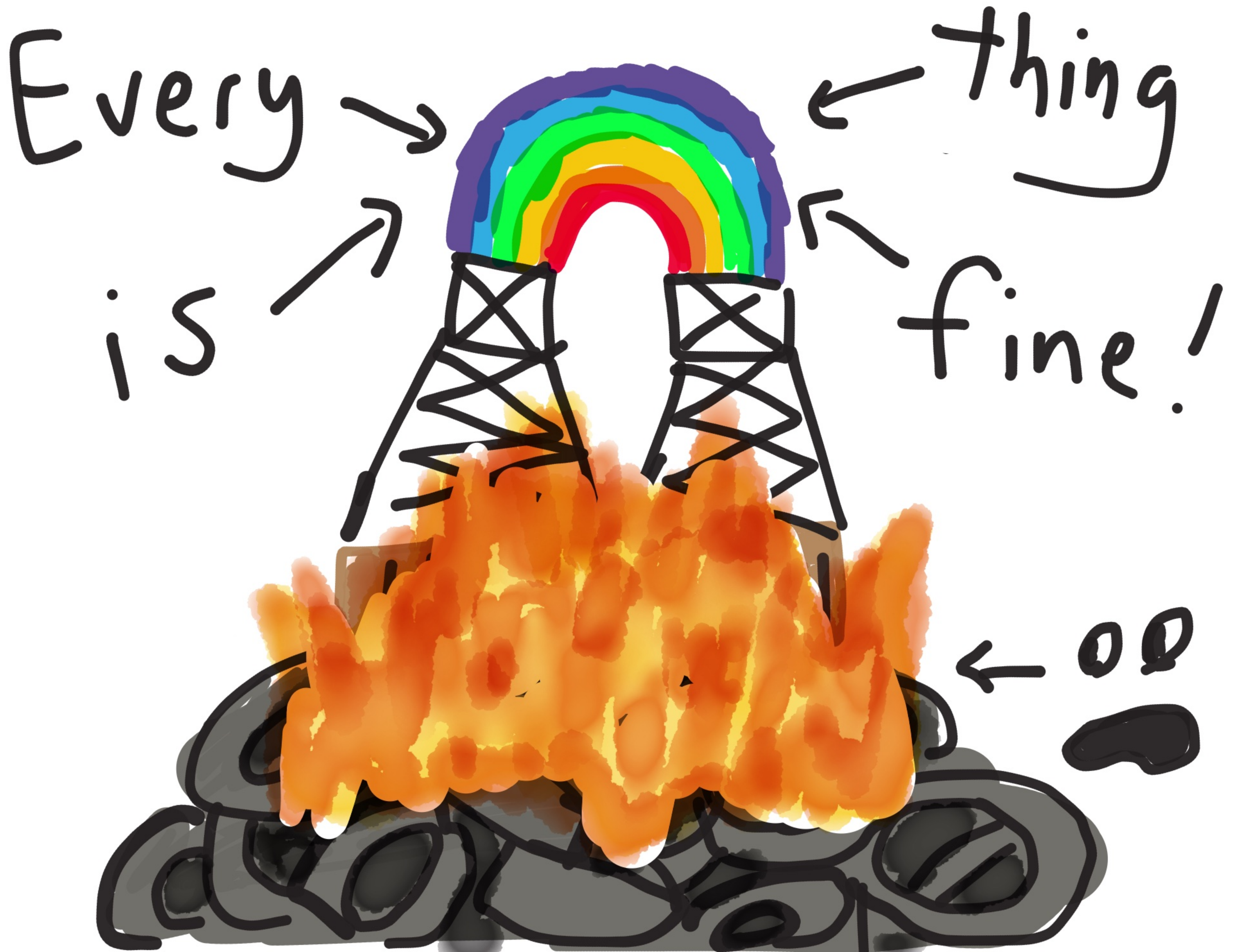
} API code

Ruby {



DBs



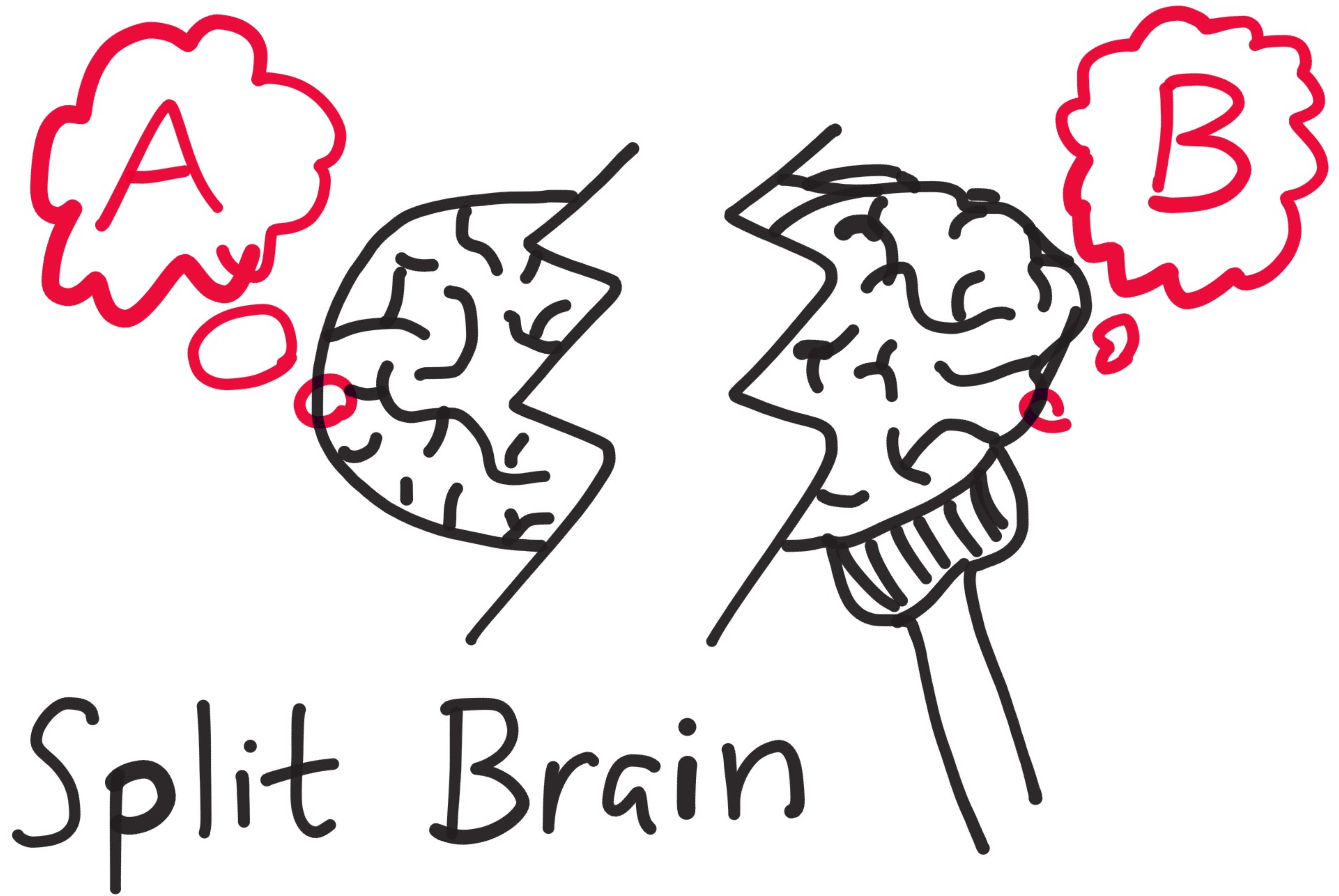


Databases!

Queues!

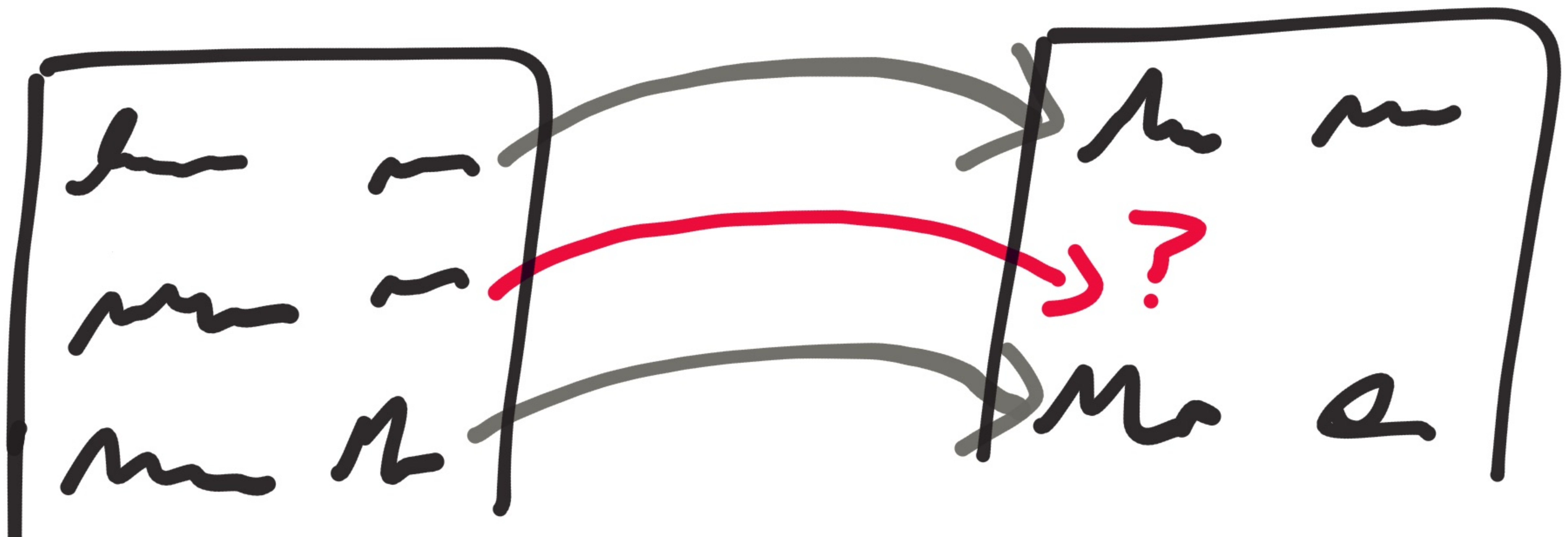
Discovery!

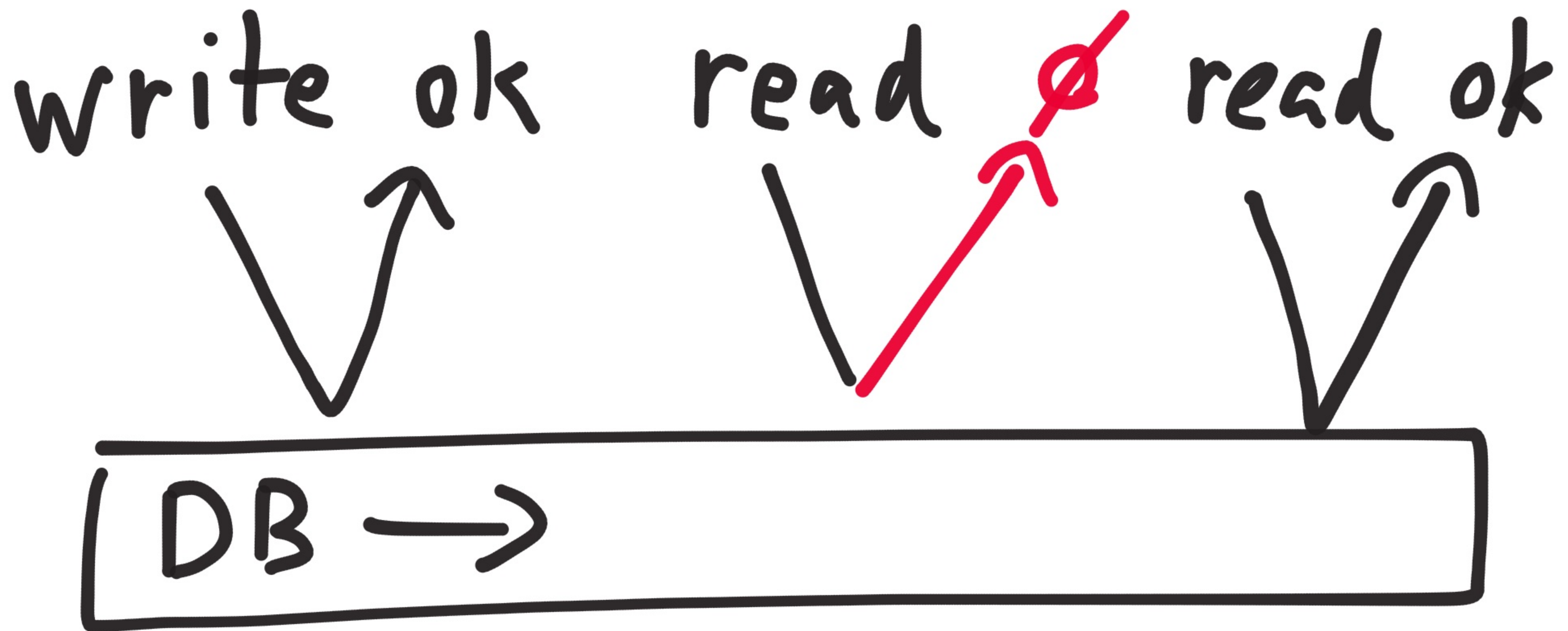
THE HORROR



Broken

Foreign Keys

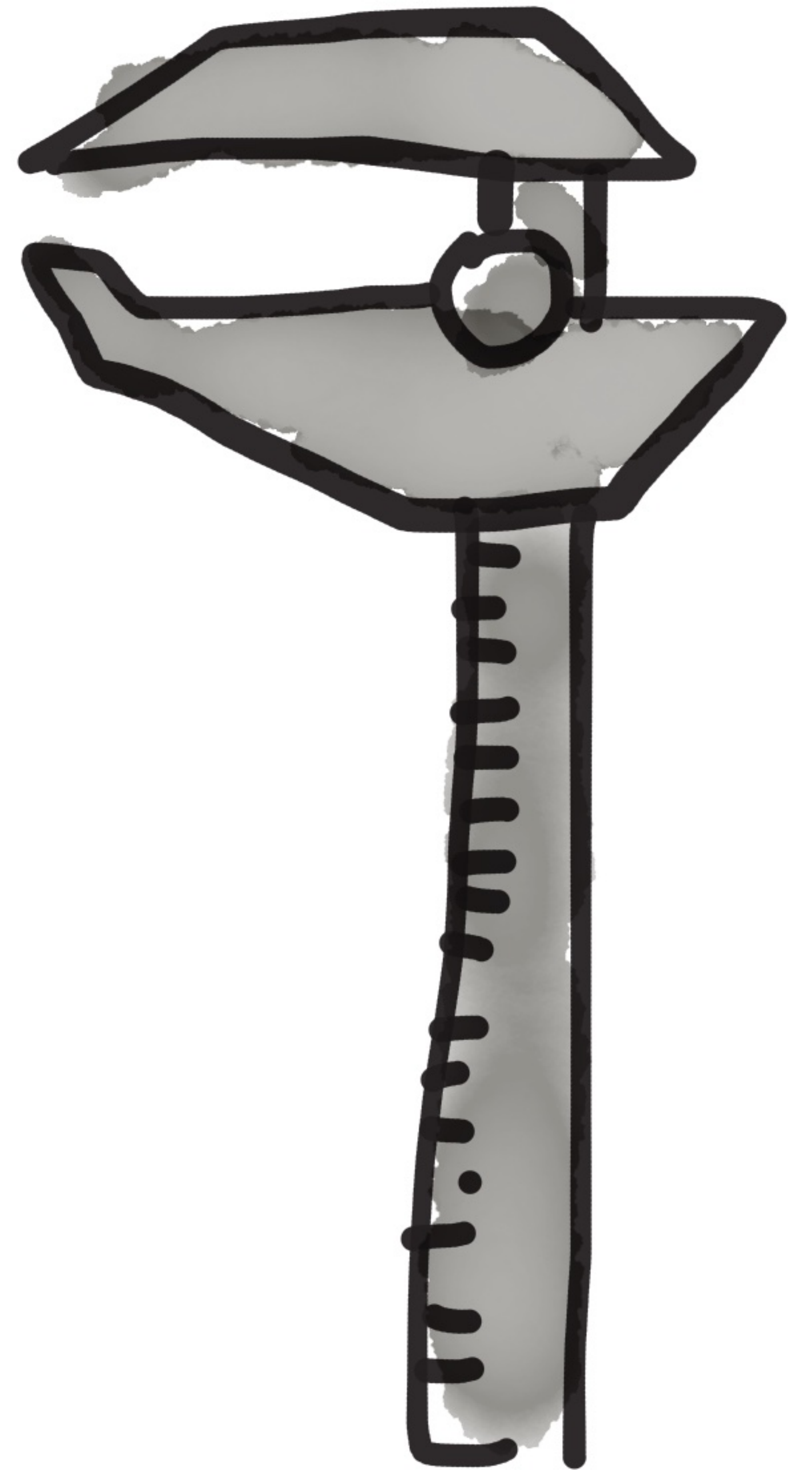




Anomalies

How do you
know if a
system is safe?

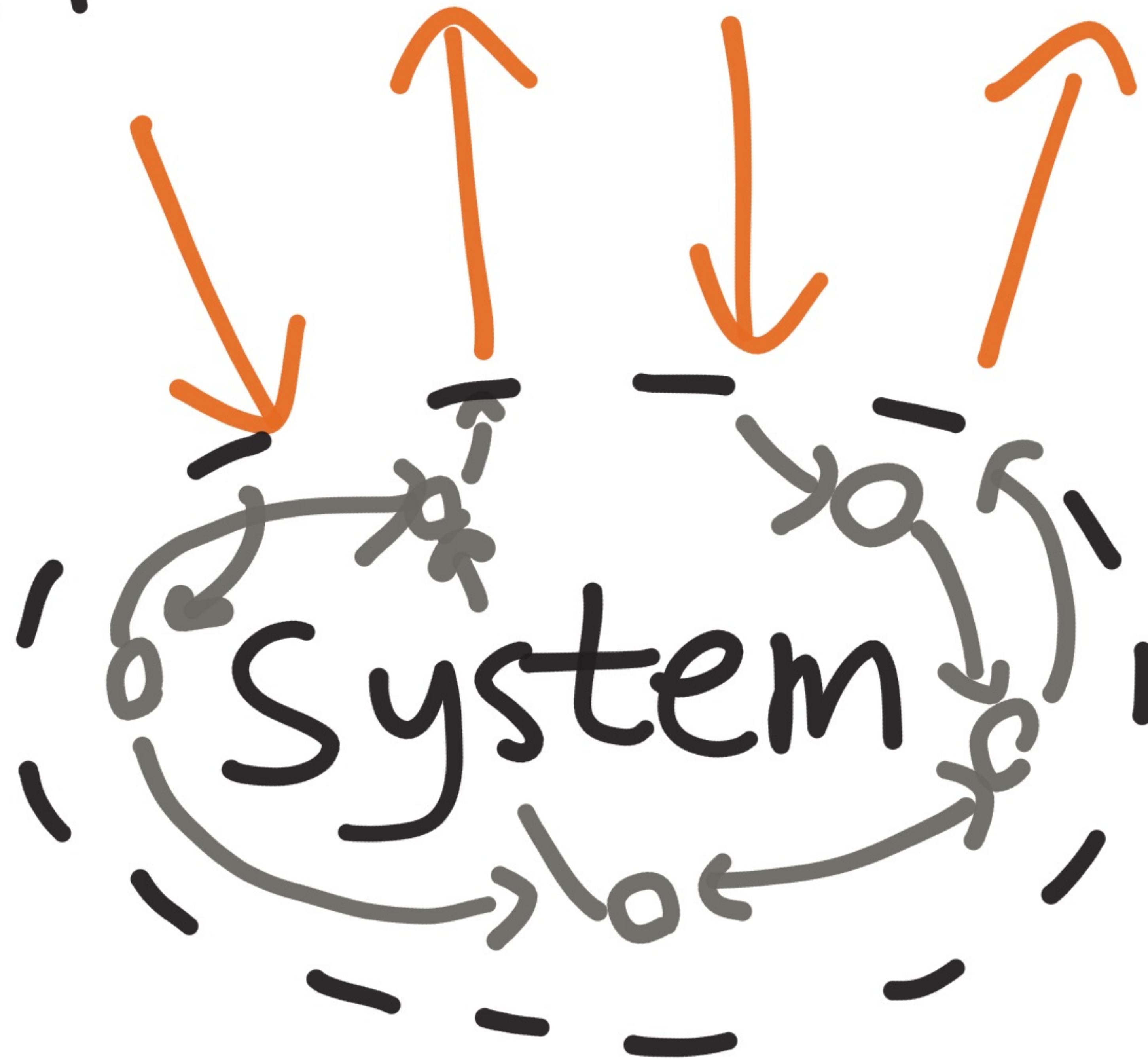
Measure
your
Systems



Jepsen

github.com/aphyr/jepsen

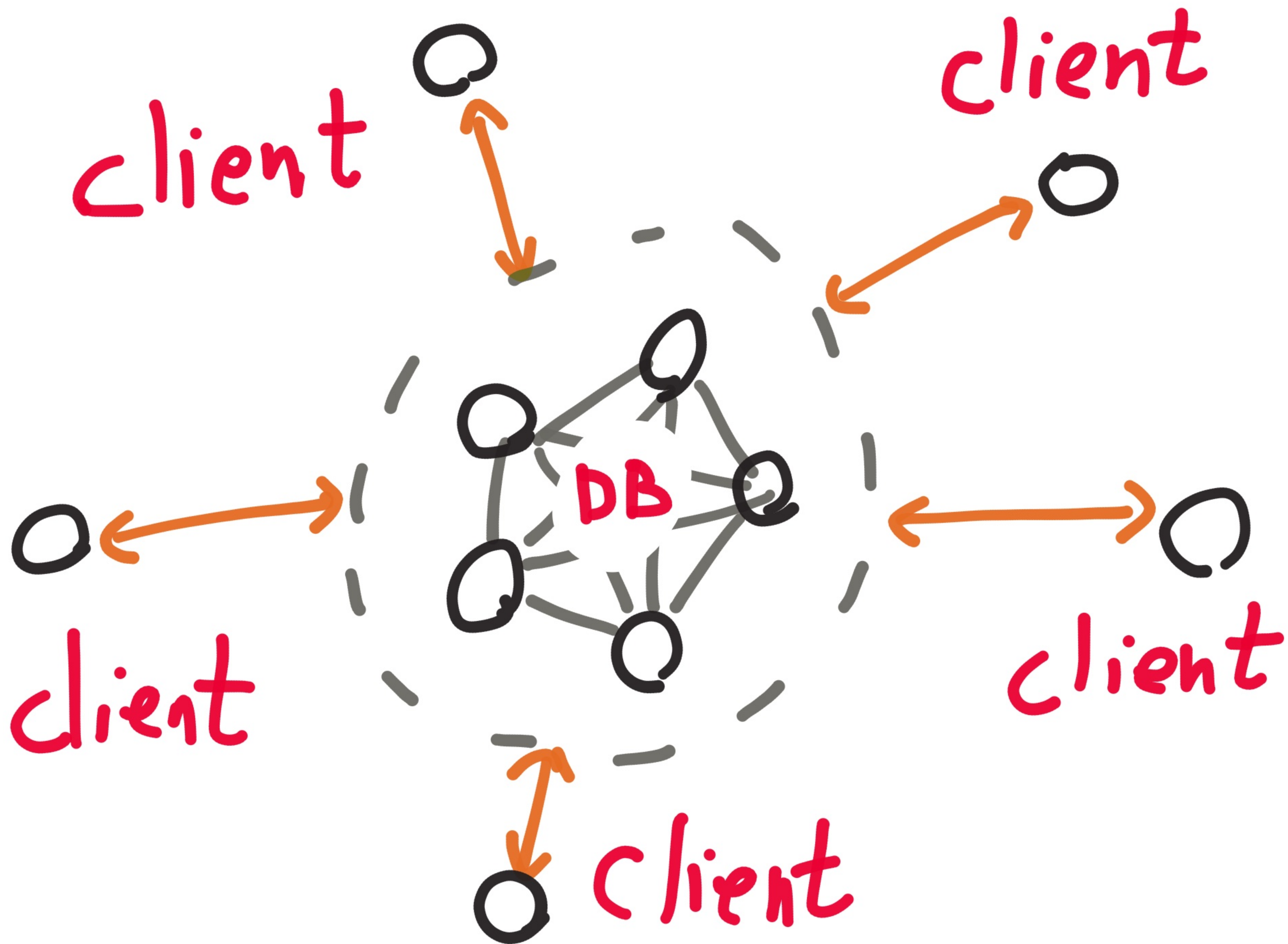
Environment





- INVARIANTS -





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

DB:

client: $w - w'$ $w - \dots$

Clients Generate

random operations (w)

and apply them

to the system (w')



invoke  ok

invoke fail

involve ? ? info ? ? ?

invoke ok

invoke fail

invoke ok

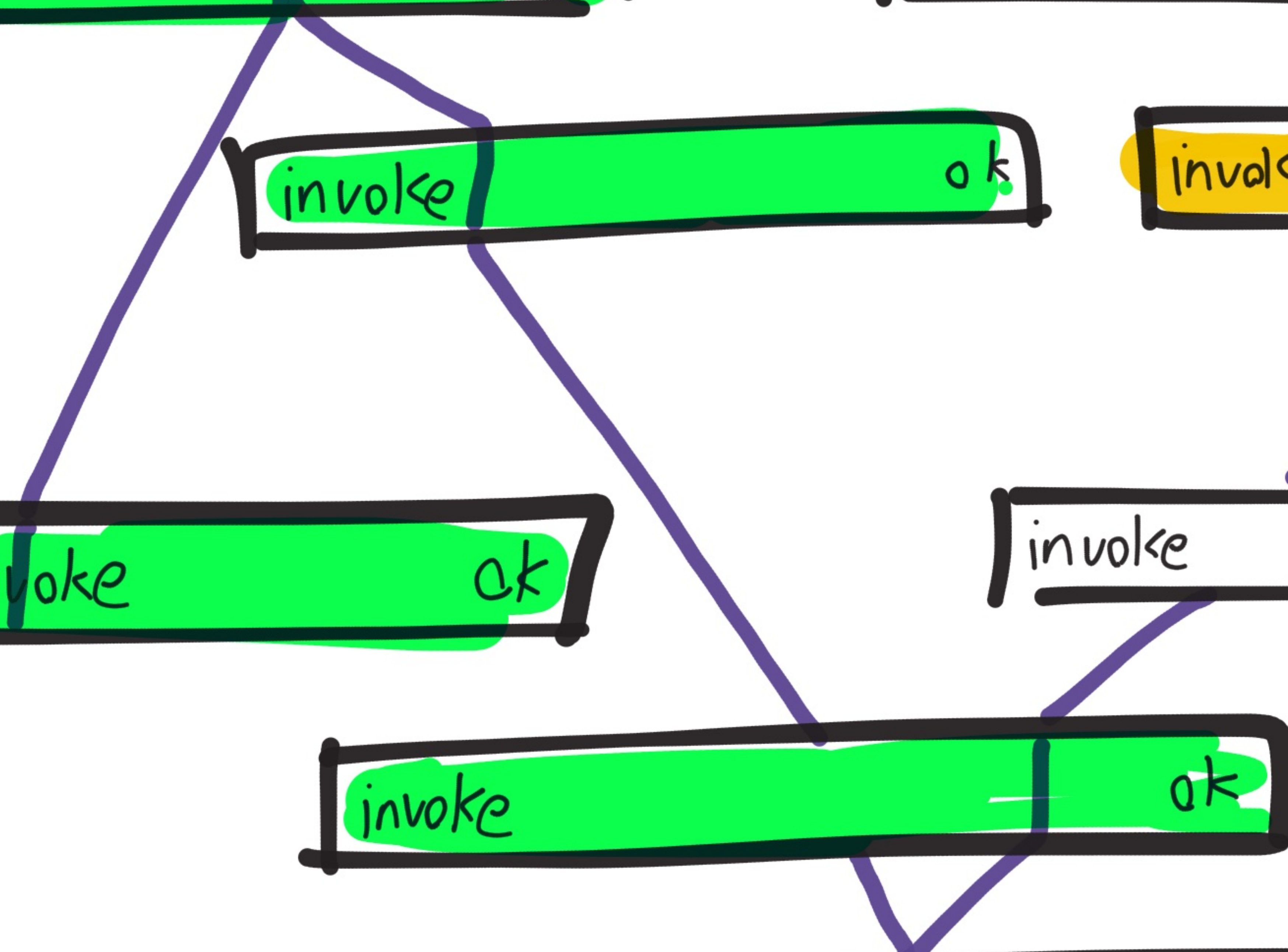
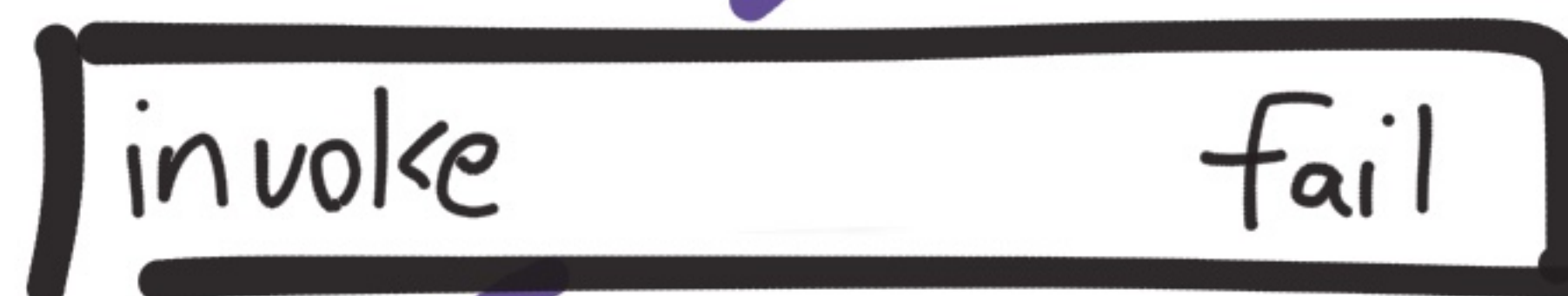
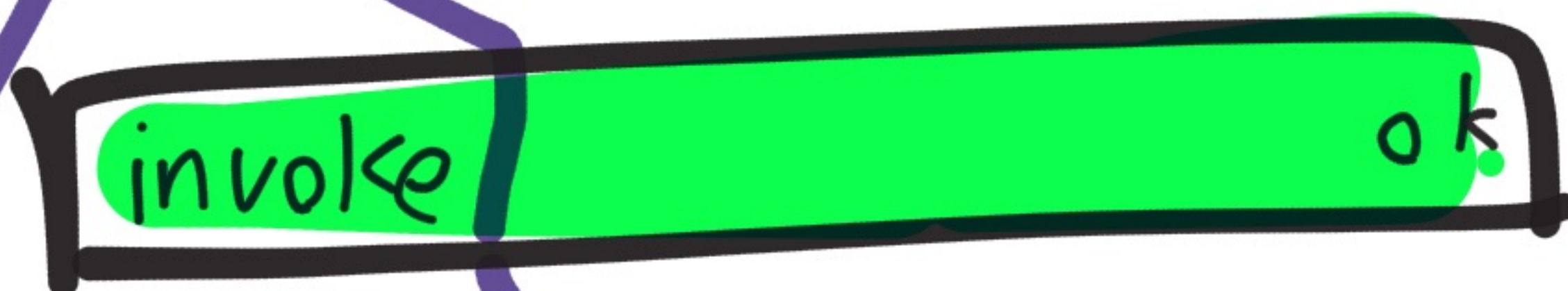
invoke info

invoke ok

invoke fail

invoke ok

invoke info

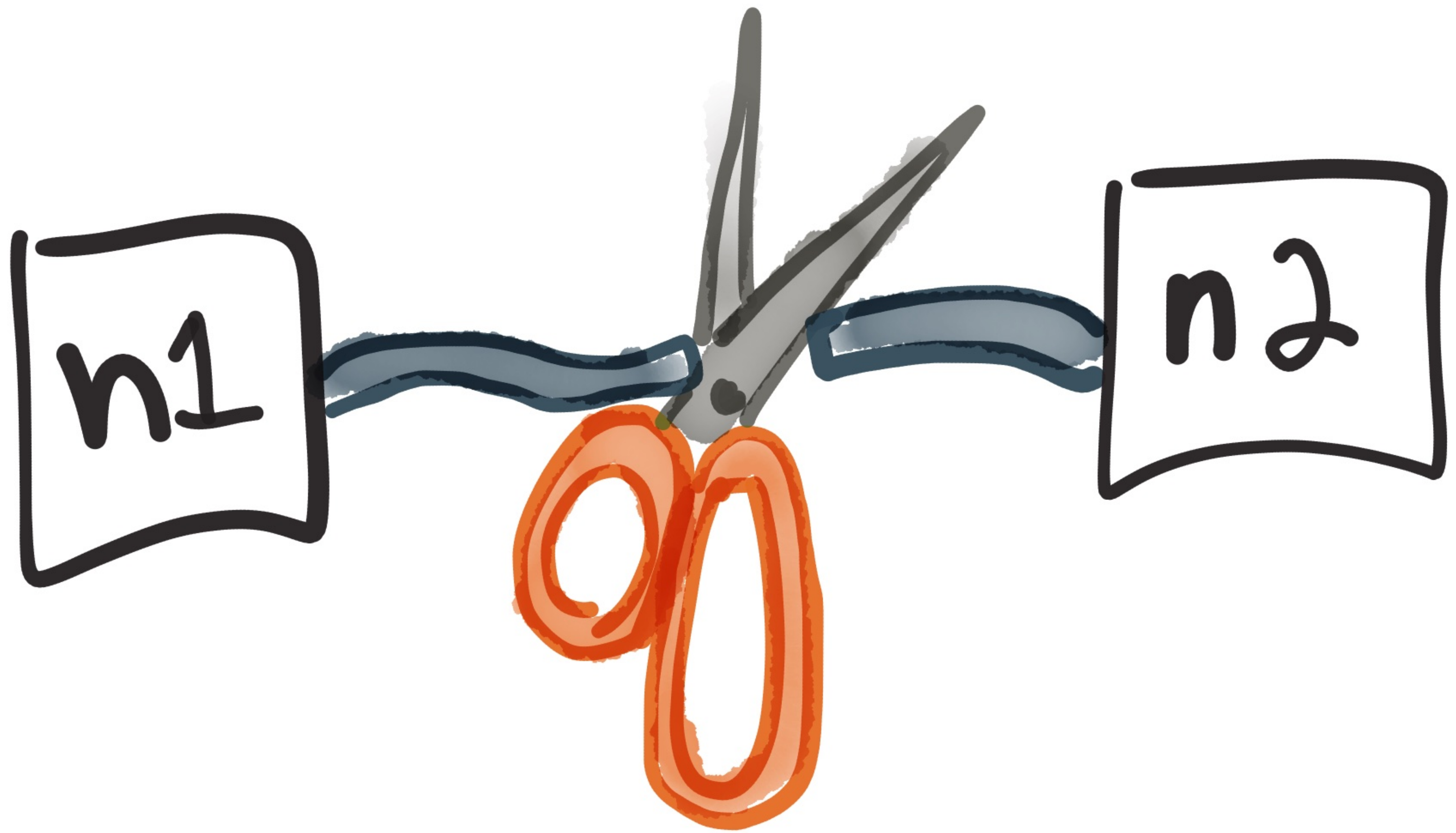


1. Generate ops

2. Record history

3. Verify history is

consistent w/ model



Partitions!

"So, what
have you
found?"



Riak

5/13

LWW $\rightarrow$ last writer

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 loss

NuoDB

9/13

Beat CAP by buffering all requests in RAM during partition

Kafka

9/13

In-sync Replica Set

could shrink to 0

nodes, causing msg

loss.

Zookeeper

9/13

Works.

etcd / Consul

6/14

Stale reads

Elastic search

6/14

Loses documents in
every class of partition
tested.

RabbitMQ

6/14

Split brain, massive

message loss

5/15

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

VoltDB 6.3

7/16

-
- Stale reads
 - Dirty reads
 - Lost writes

Crate.io

2016

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

CockroachDB

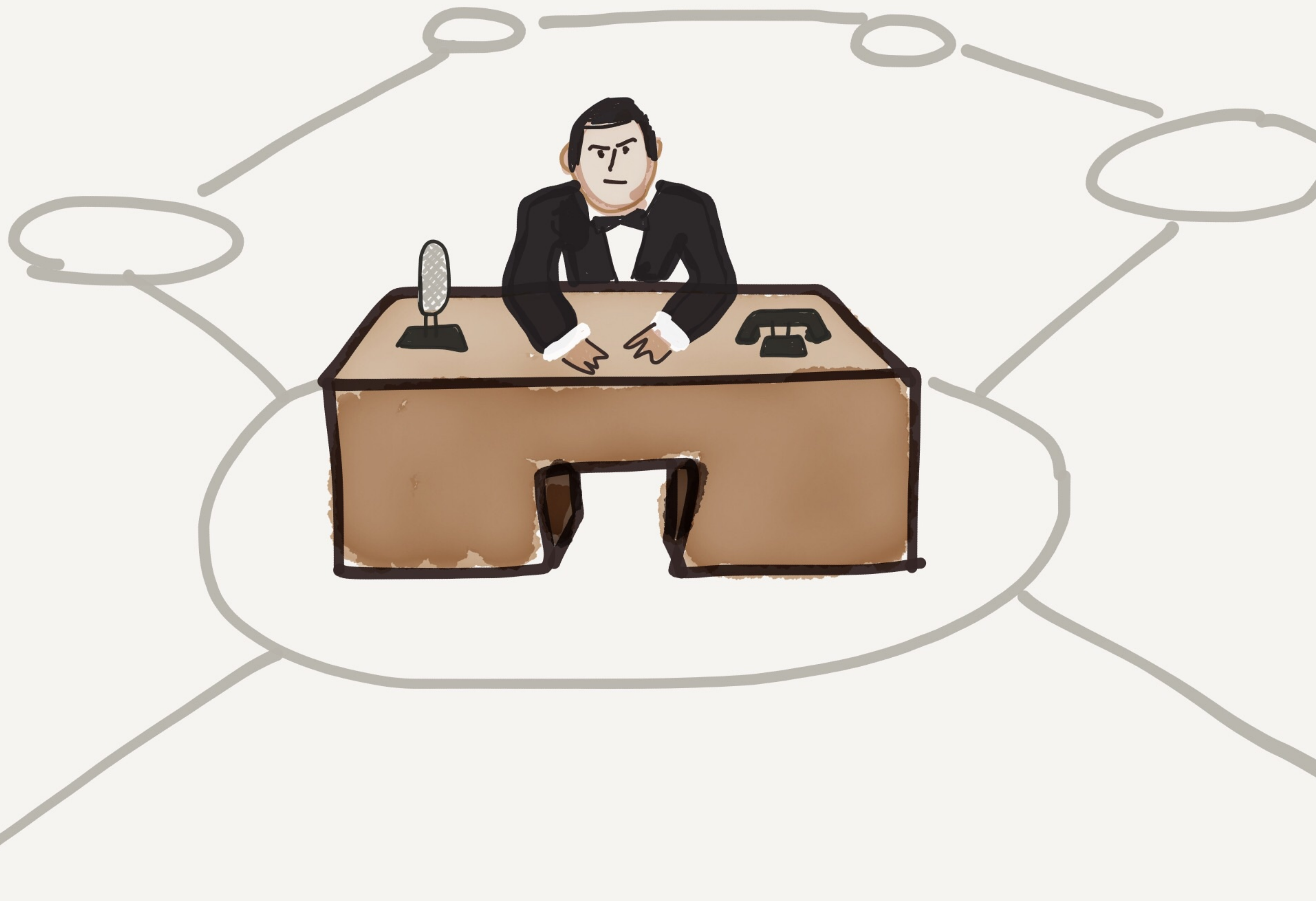
2017

- phantoms
- Double inserts

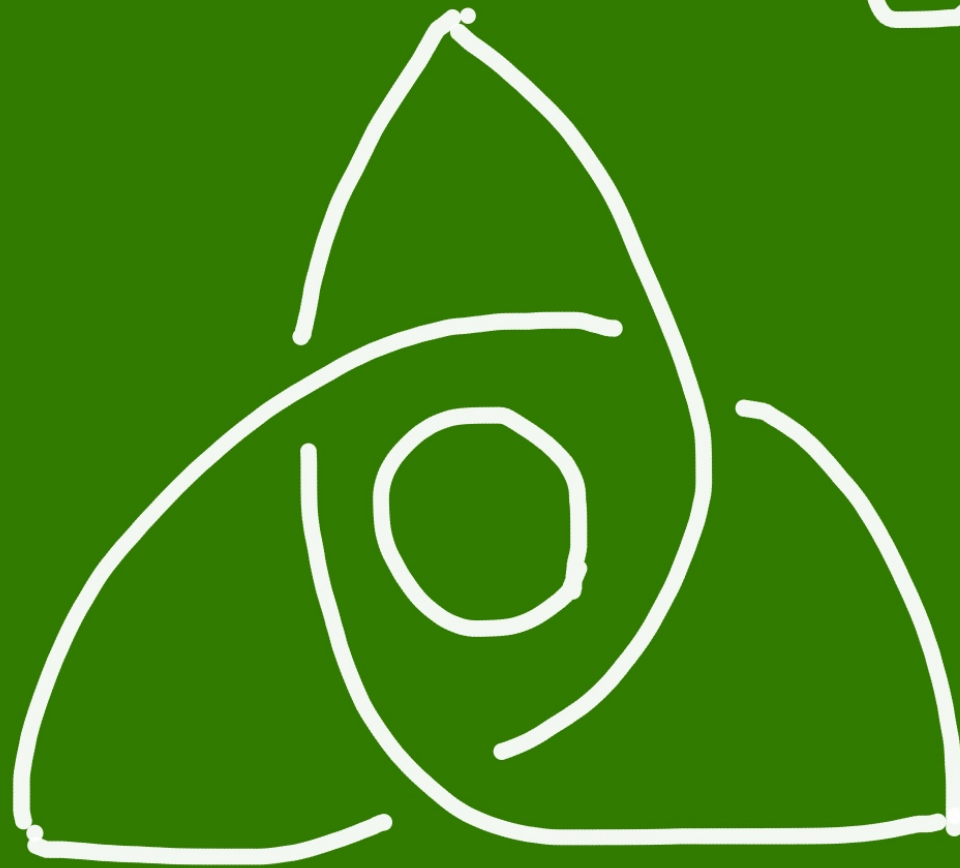
MongoDB

2017

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



0.10.2



Tendermint

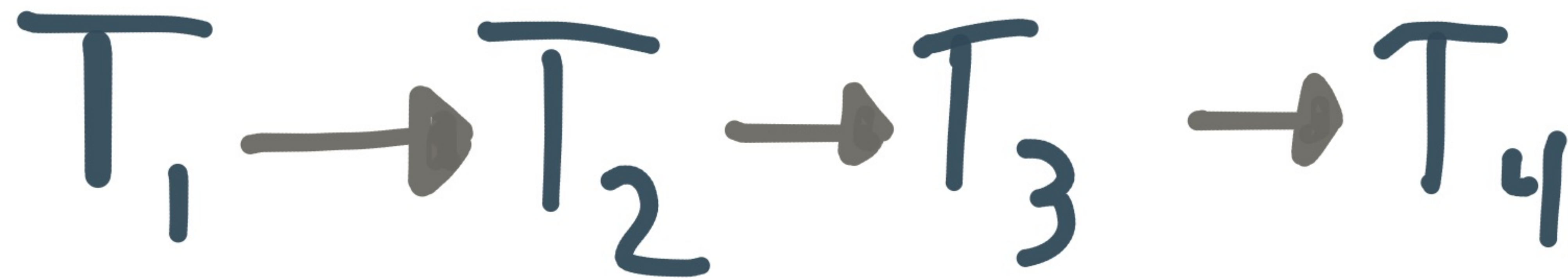
-Transaction replication

-Pluggable FSMs

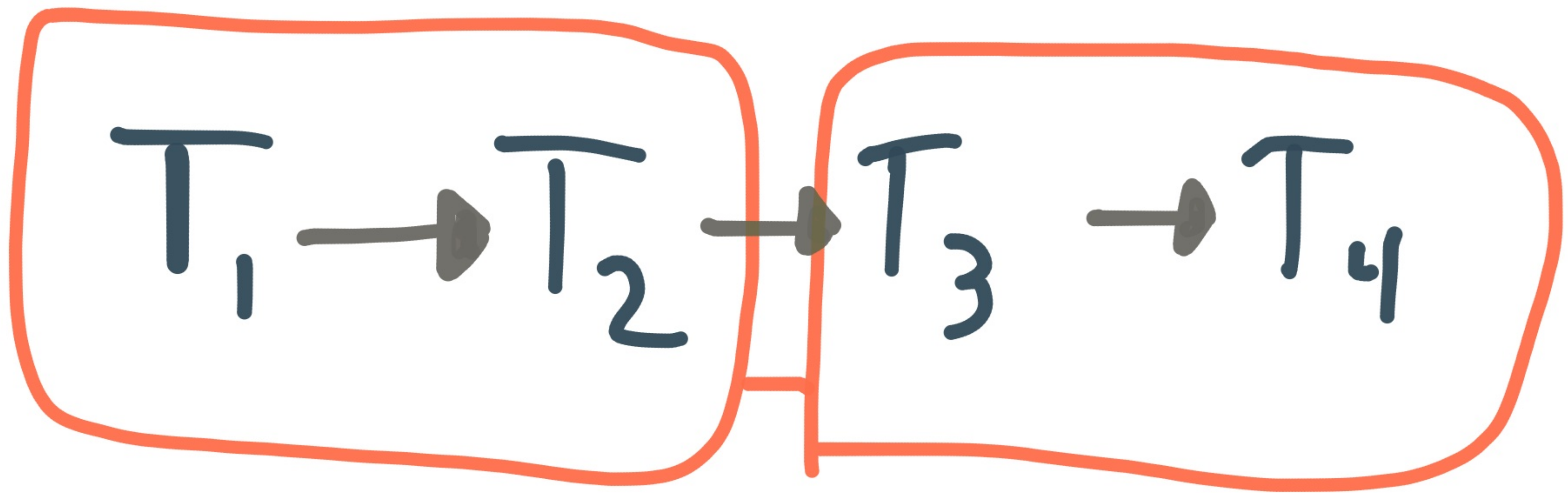
-Byzantine fault tolerant

"Raft Meets
Blockchain"

Like eg. Raft, forms a
linearizable order of txns



Like Raft, provides a linearizable order of txns



blockchain!

Byzantine

Fault

Tolerant!

shun!



shun!



shun!

shun!



Pluggable

State

Machines

validator

state

validator

state



CAS-register

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

set

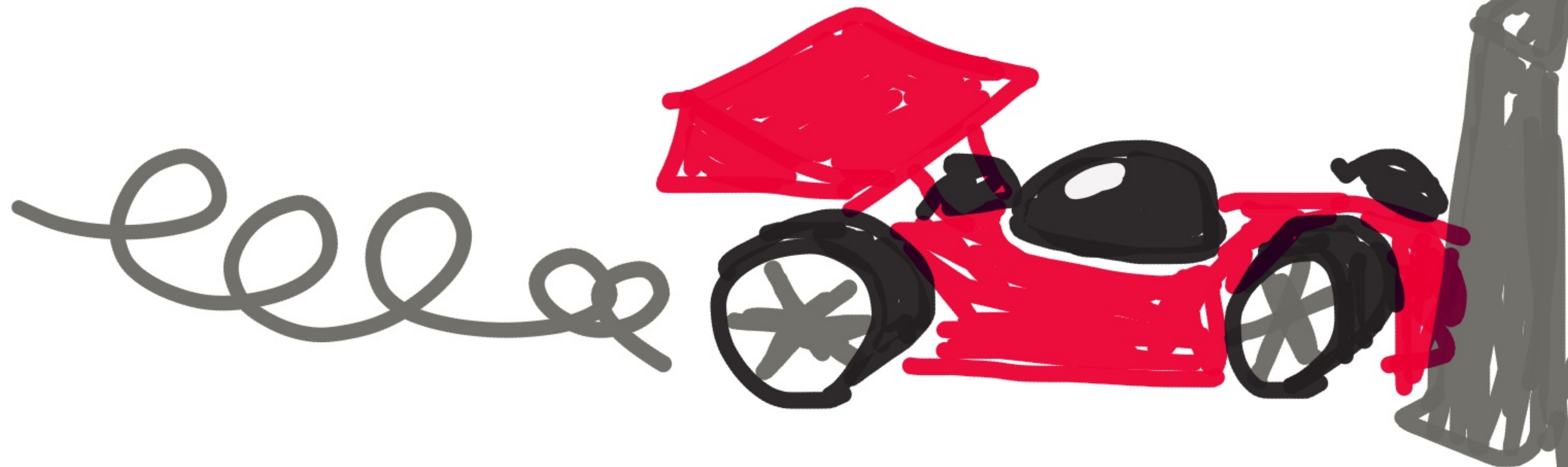
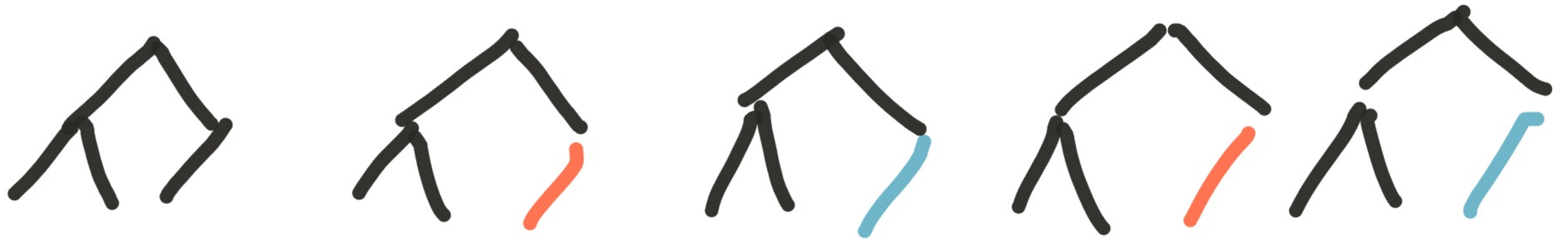
- add(6)

[read($\{1, 2, 3\}$)

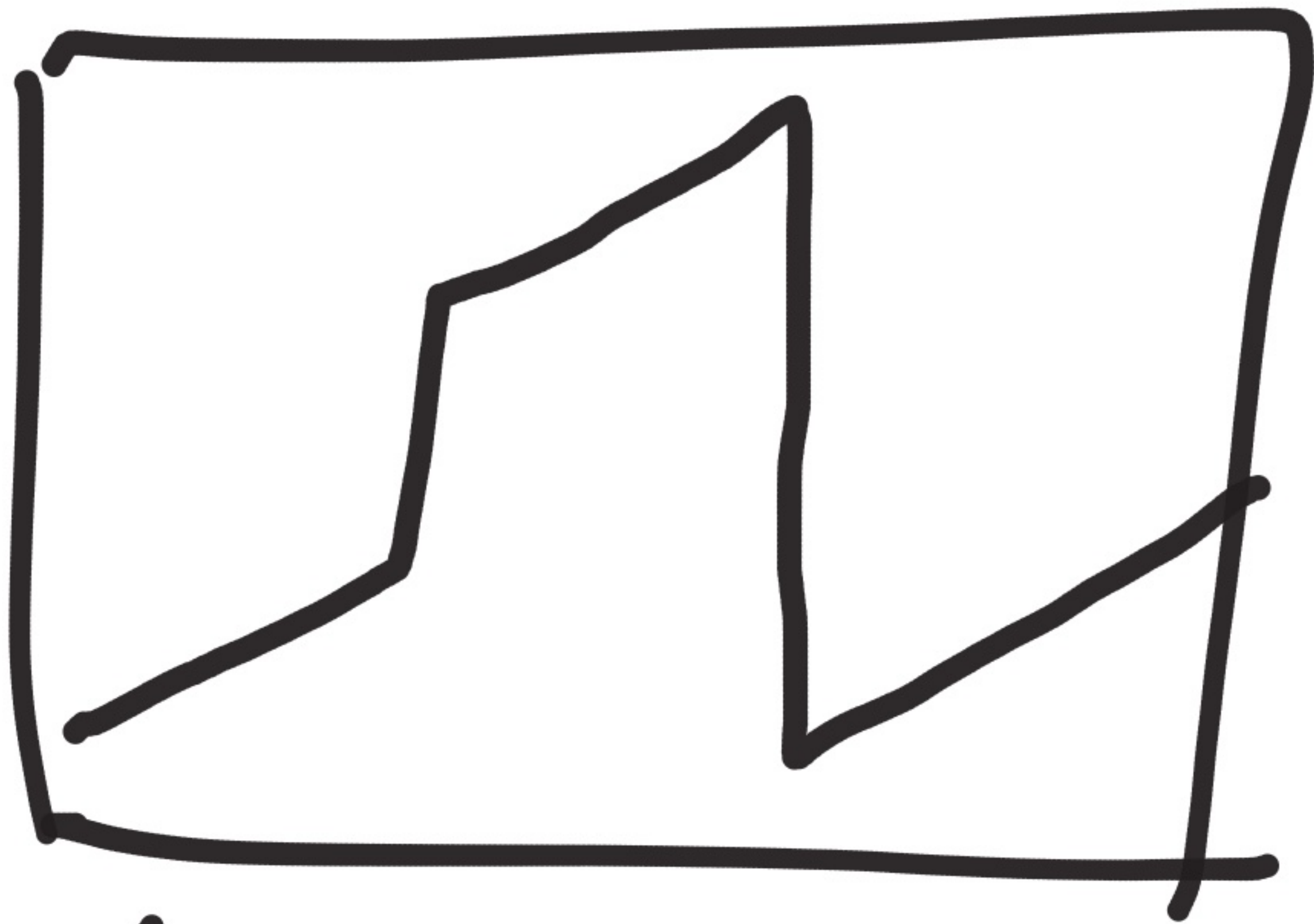
cas($\{1, 2, 3\}$, $\{1, 2, 3, 6\}$)

- read($\{1, 2, 3, \dots\}$)

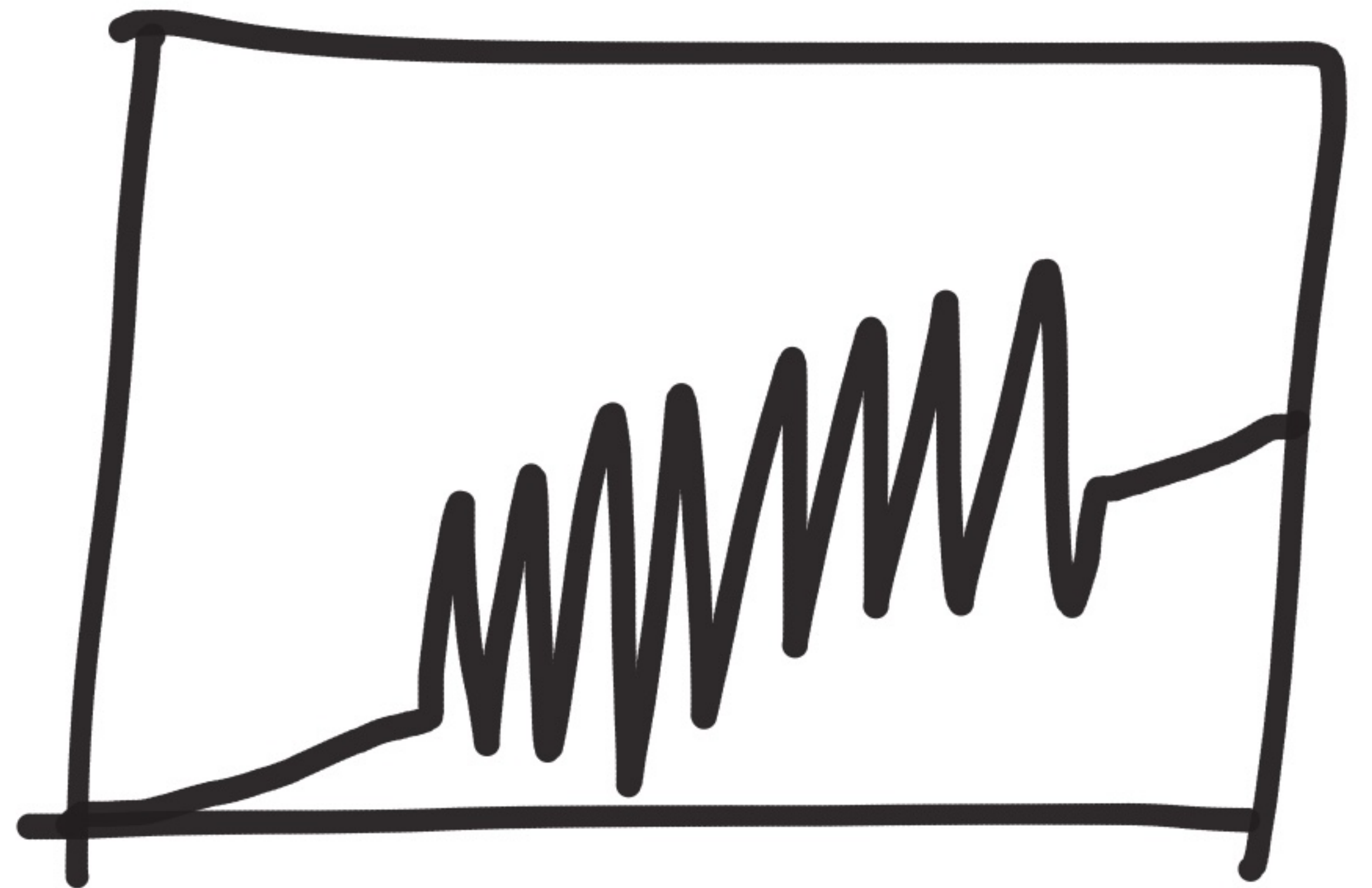
Merkleeyes Race



clock skew



bump



strabe

crashes

n1

———— x

—————

n2

———— x

—————

n3

———— x

—————

n4

———— x

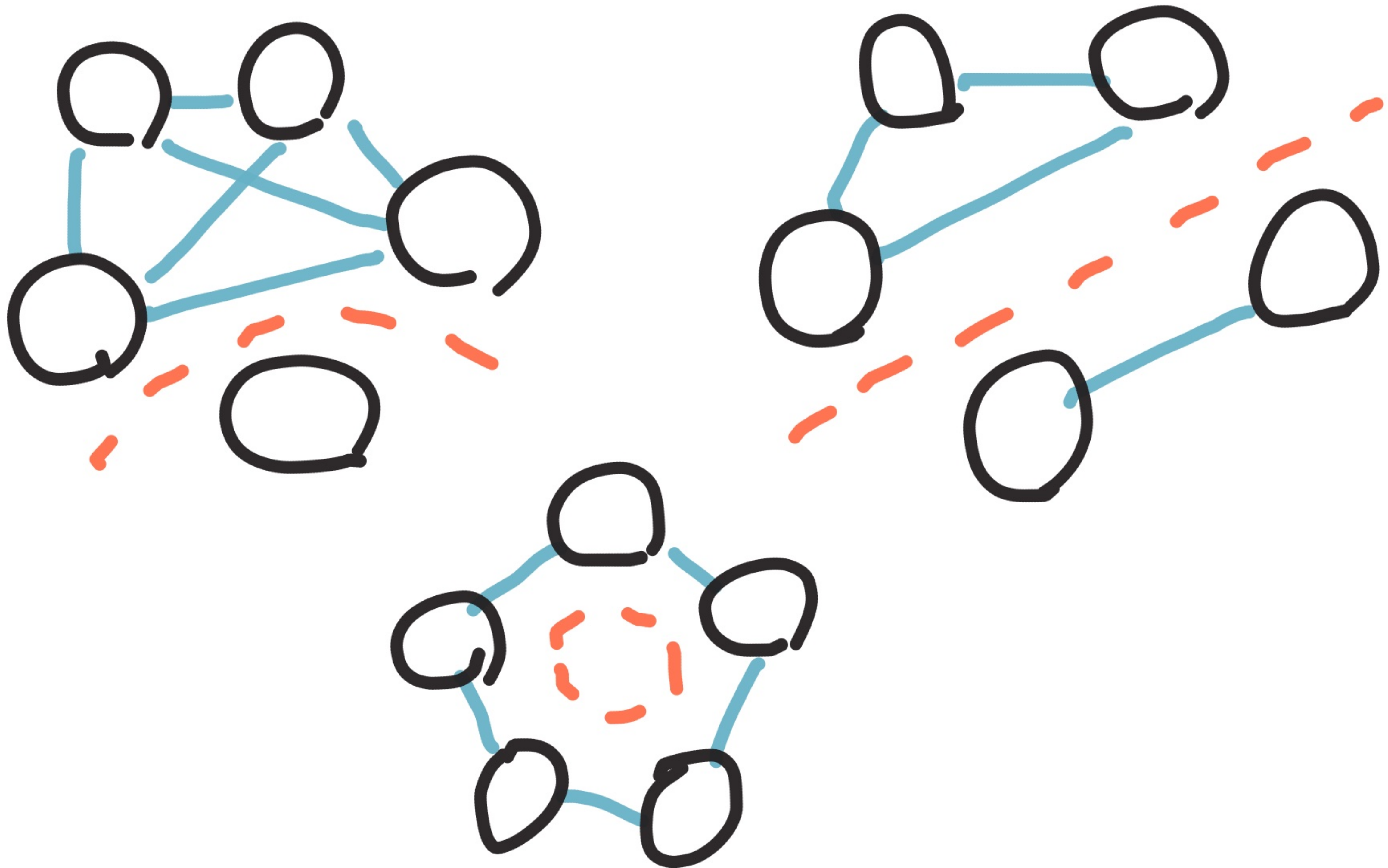
—————

n5

———— x

—————

partitions



Duplicate Validators

⁷A B²

⁷A' C²

$$7 + 2 + 2 = 11$$

Duplicate Validators

γA	B^2	$\gamma / \parallel$
$\gamma A'$	C^2	$\gamma / \parallel$

Duplicate Validators

'A

B²

'A'

C²



Lost

Documents



iff



$$\geq \frac{1}{3}$$

of votes

file truncation



merkleeyes crash!

(goleveldb bug)

Tendermint

Doesn't fsync to
disk!

Dynamic Reconfig

_____X

____X

_____X

Data corruption &
crashes are to be
expected — Tendermint
is still in beta

Team working to
fix these issues





hazelcast

3.8.3

In-memory

D A T A
G R I D

Sets

Lists

IDs

Semaphores

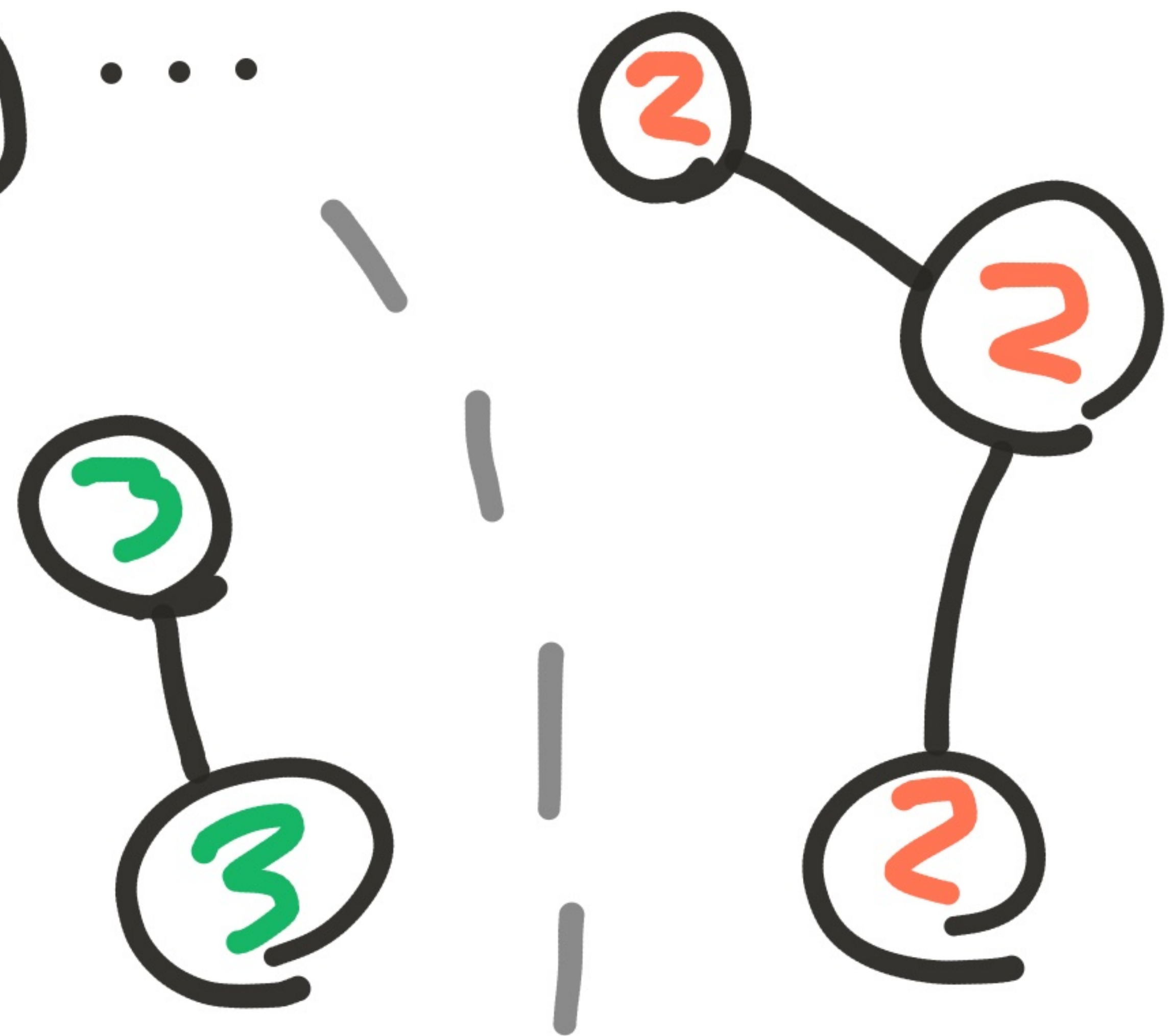
Queues

Maps

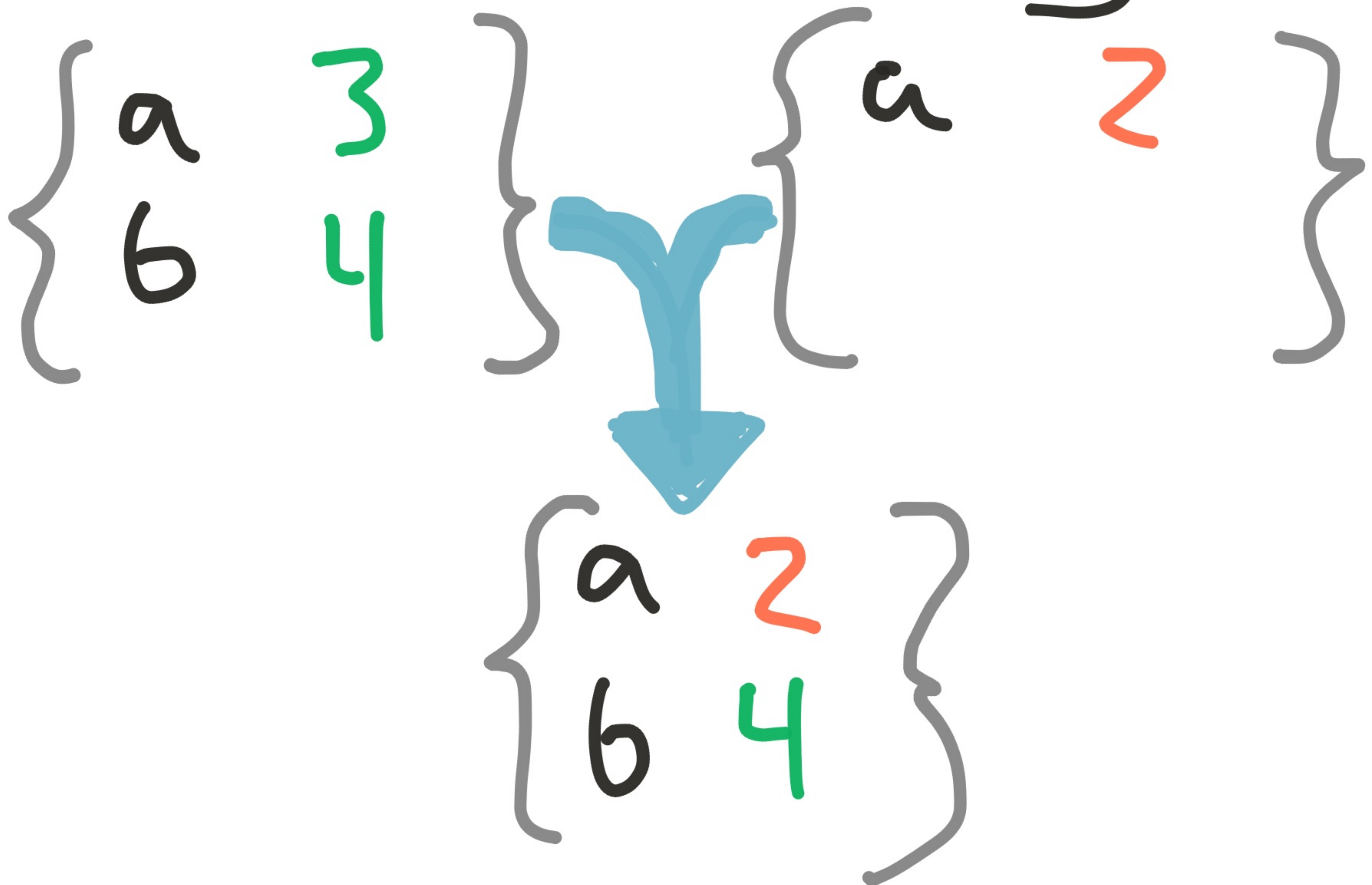
Locks

Atomic Longs

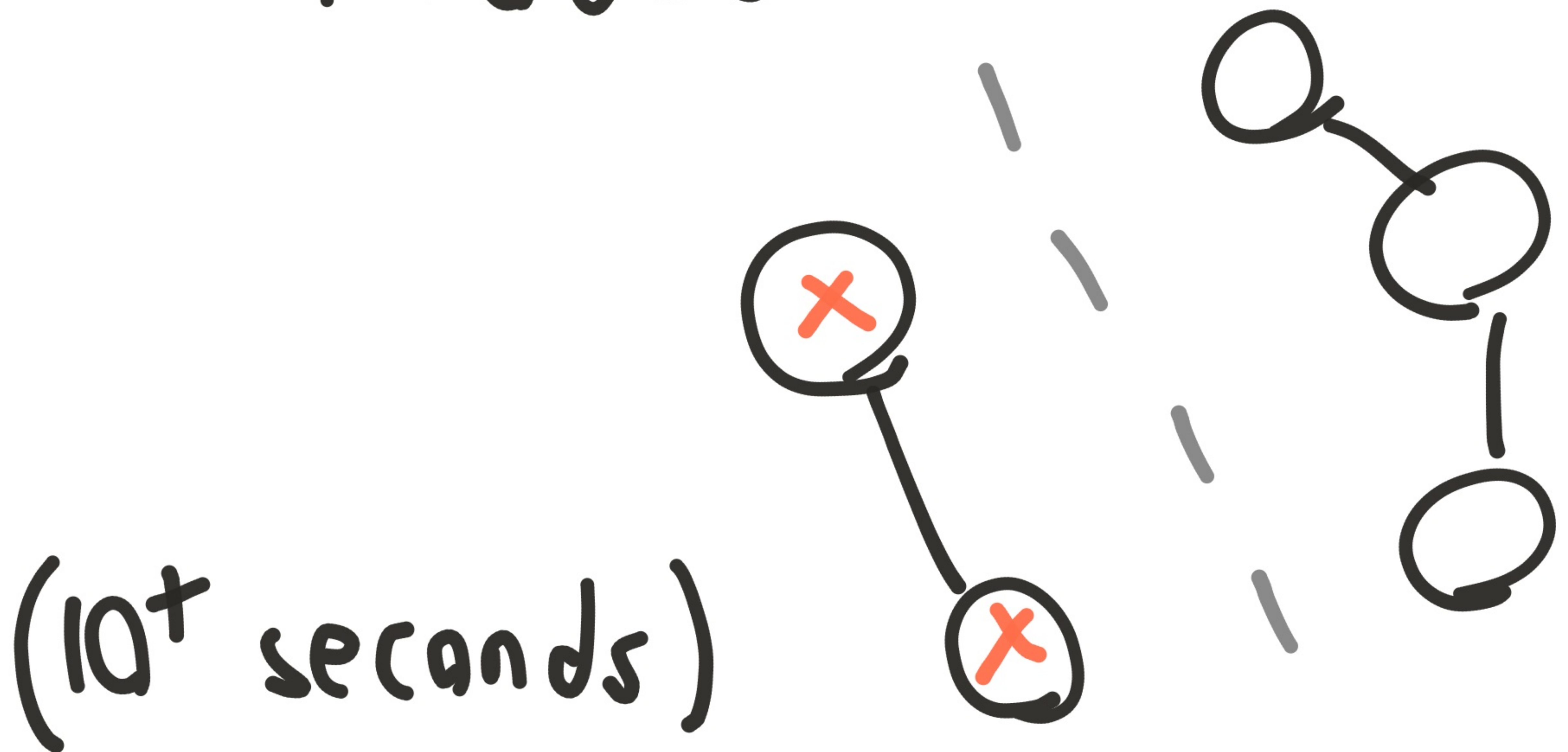
But in a partition,
components run
independently ...



Maps have merge

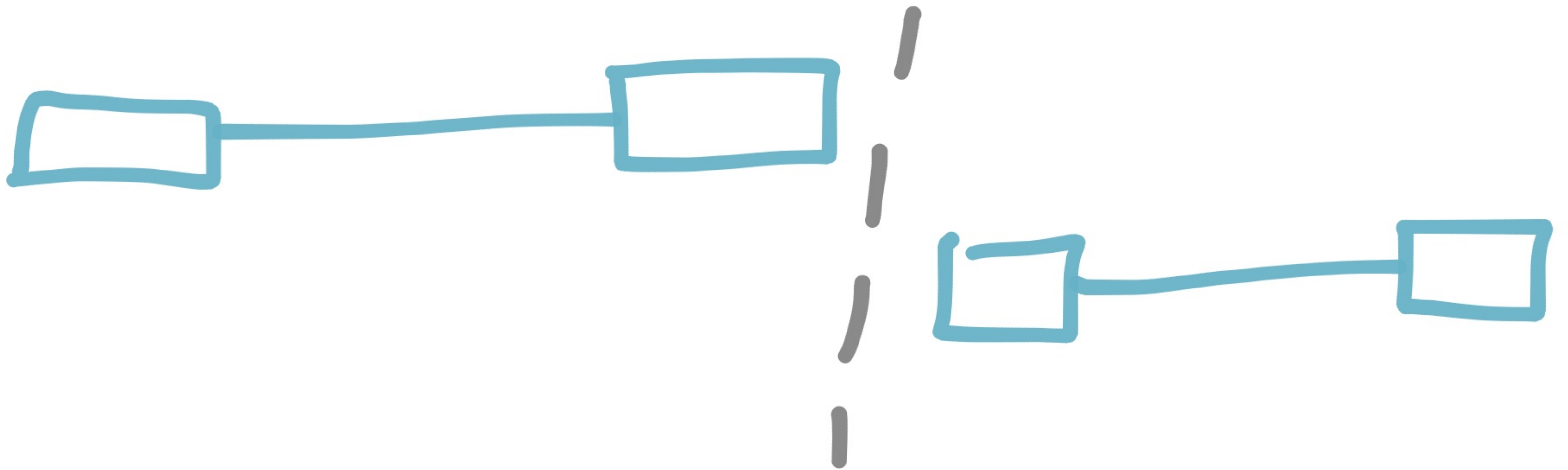


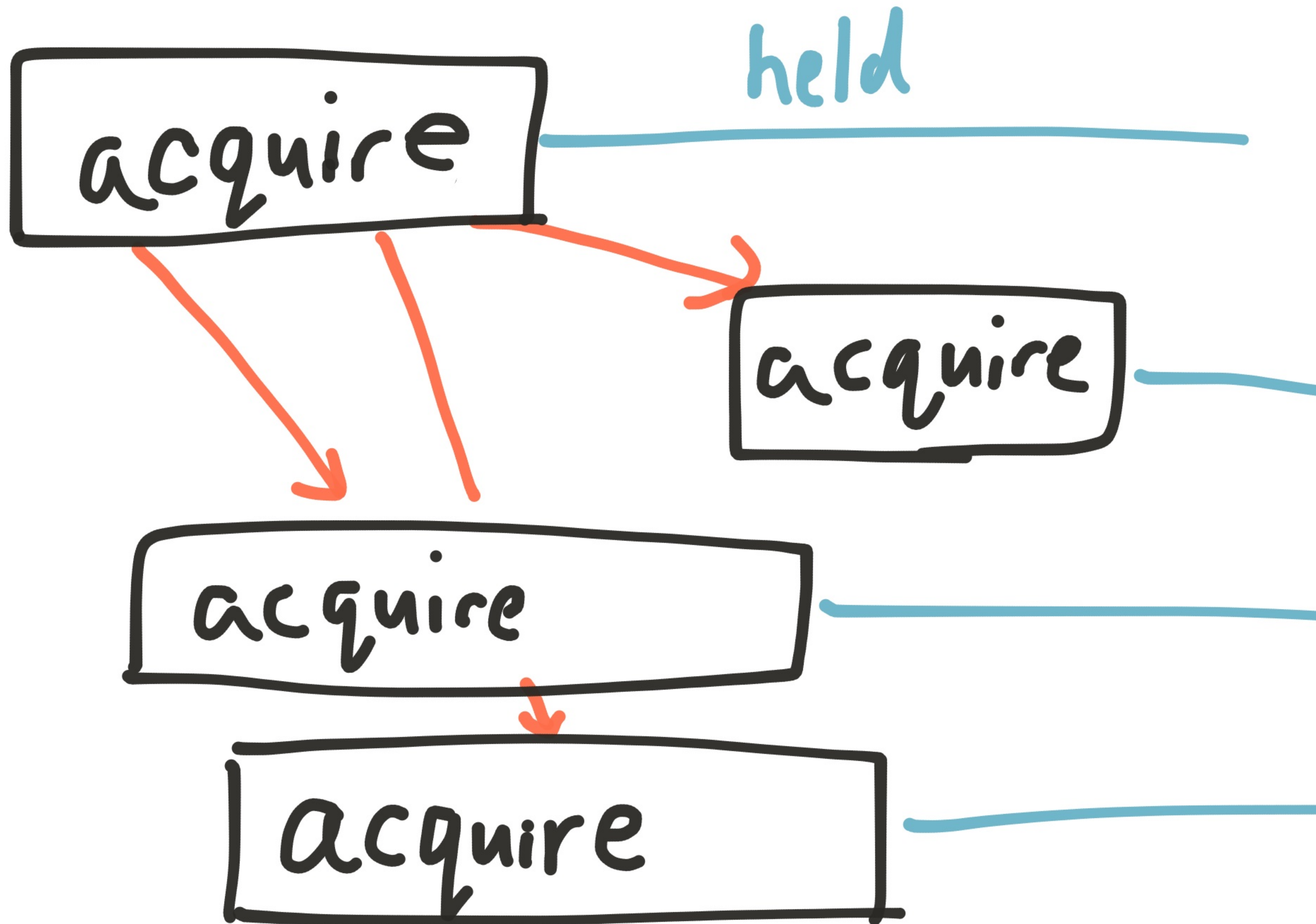
Split-Brain Protection



Lacks

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

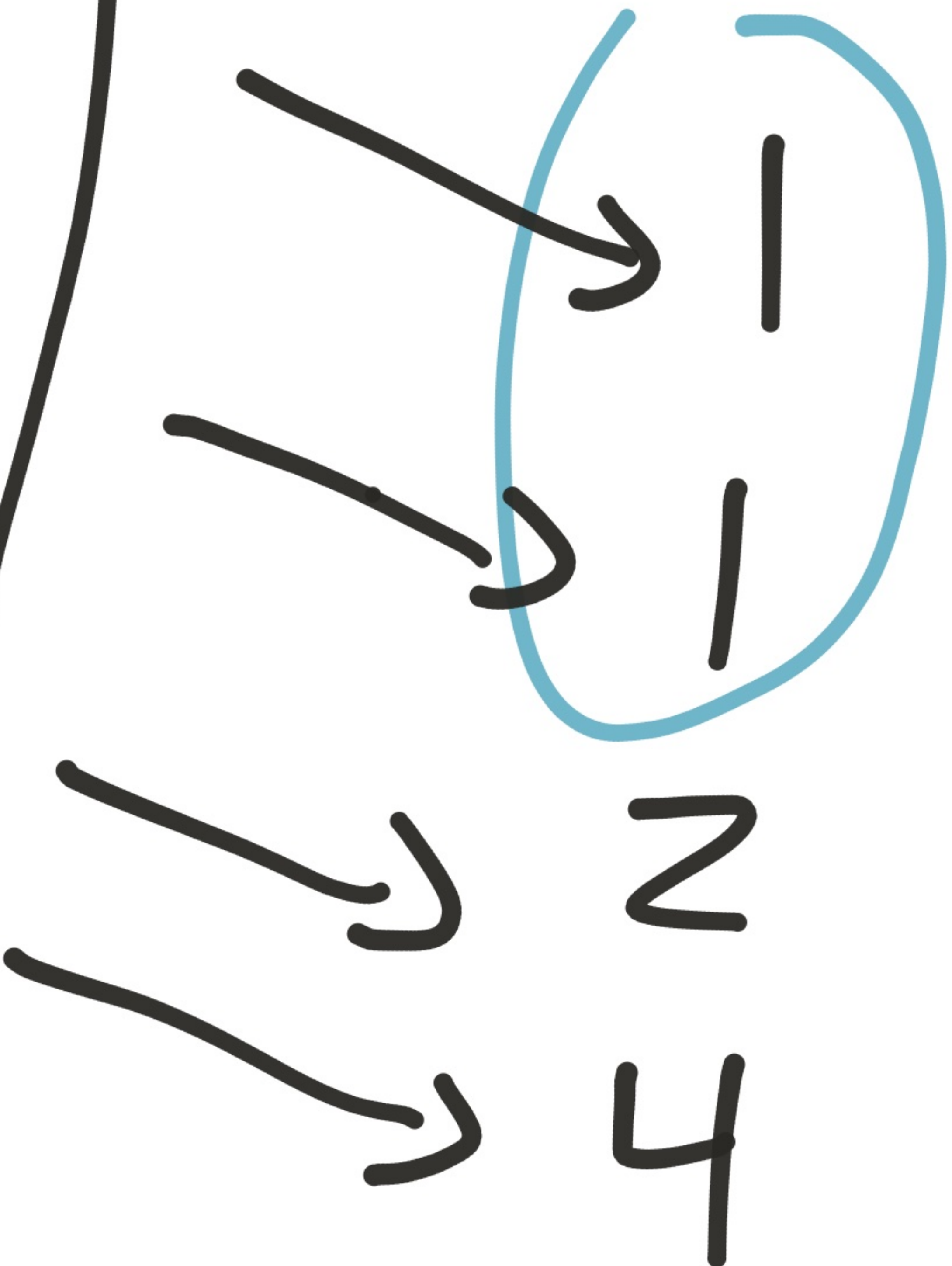
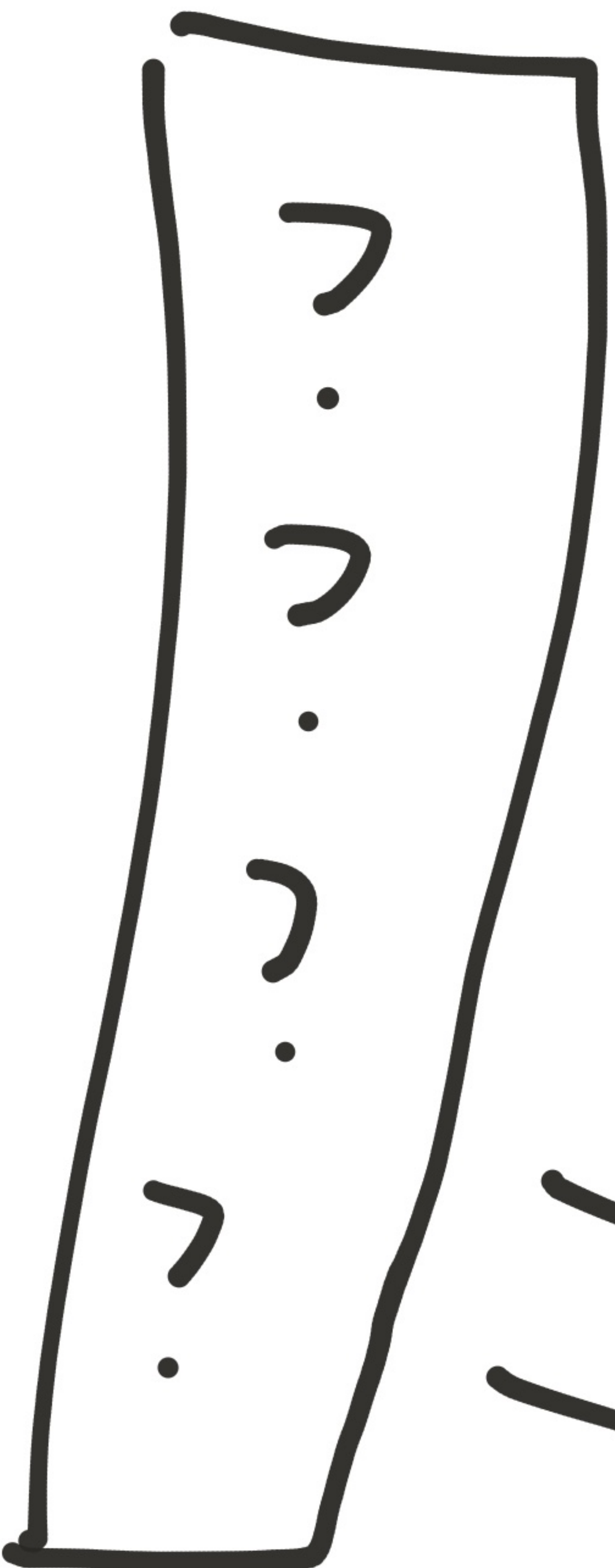
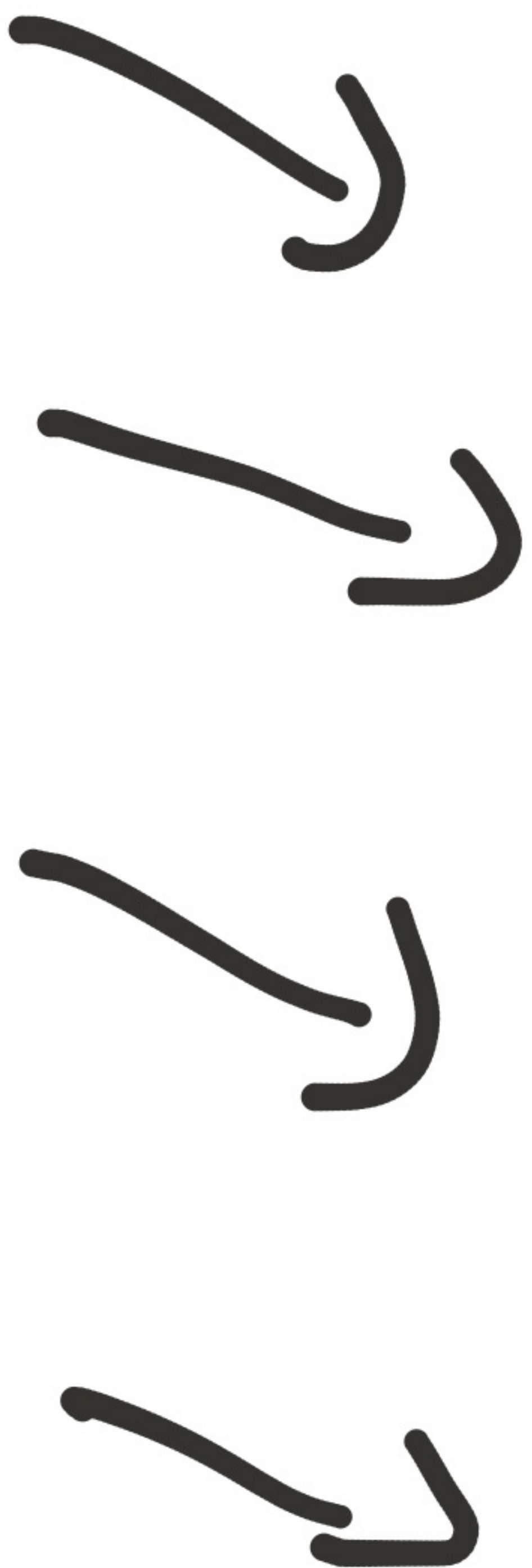




Queues

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

1
2
3
4



~ 2% duplicated

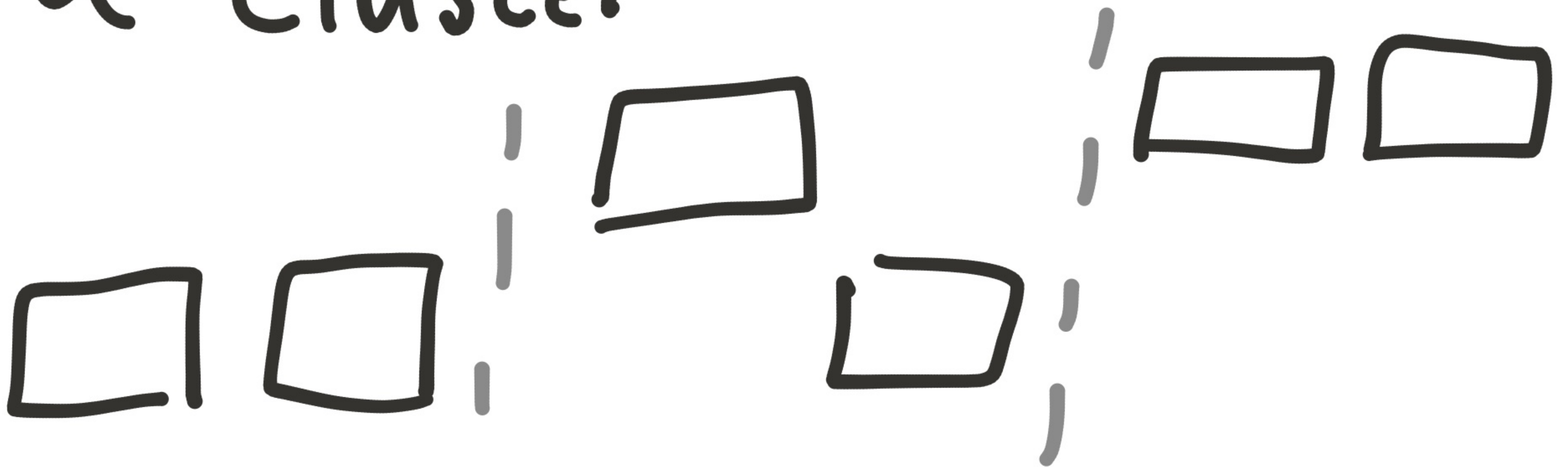
~ 2% lost

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

3 - 3

increment

$$2 + 1 = 3$$



increment

$$2 + 1 = 3$$

"still 2?"
"yes!"

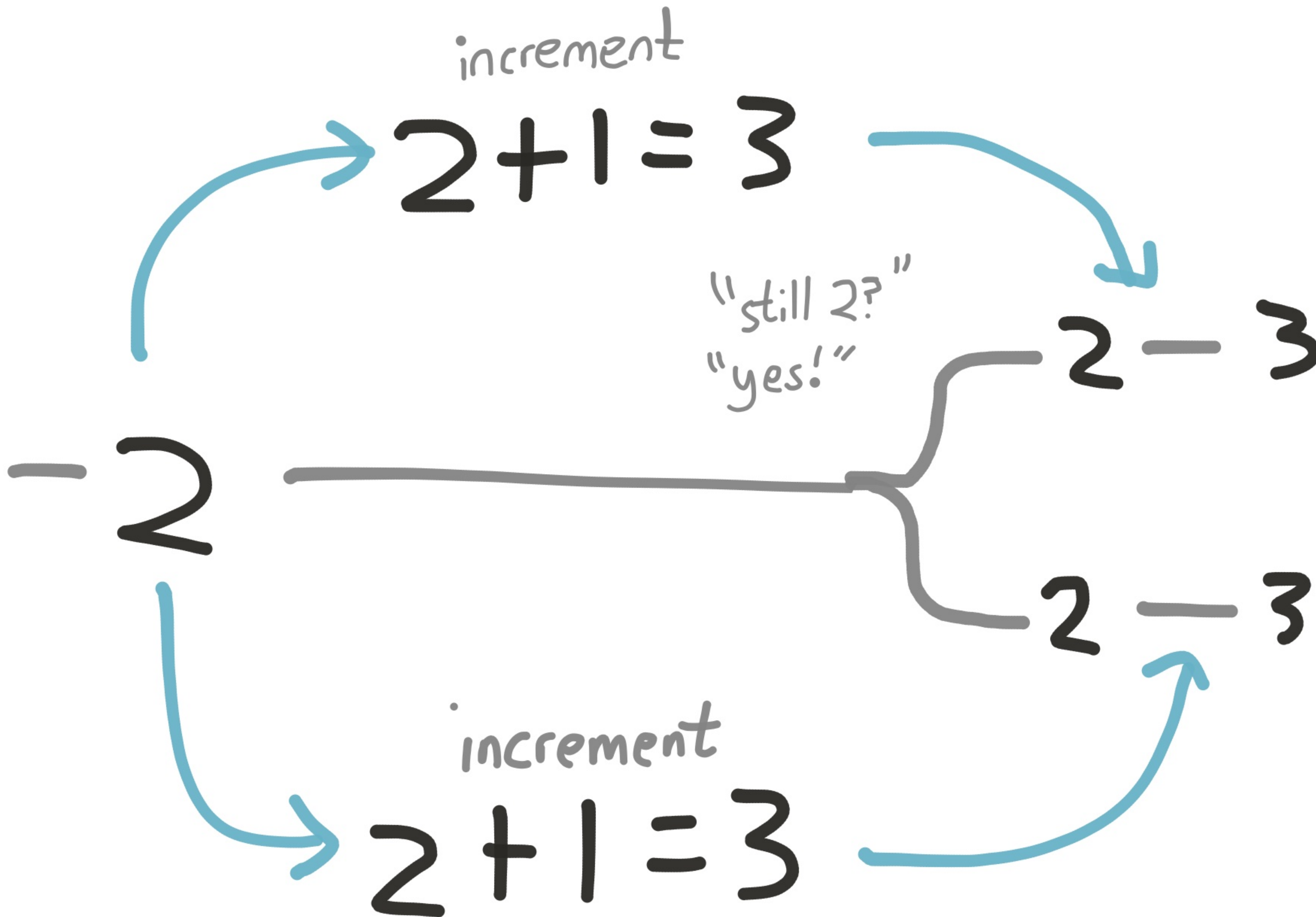
$$2 - 3$$

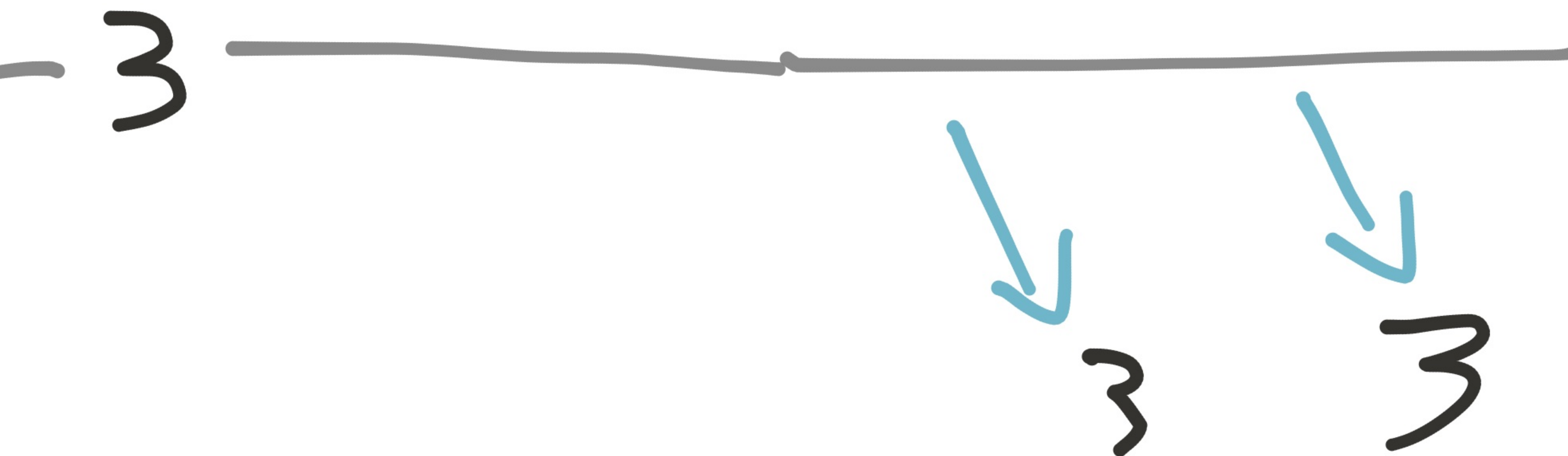
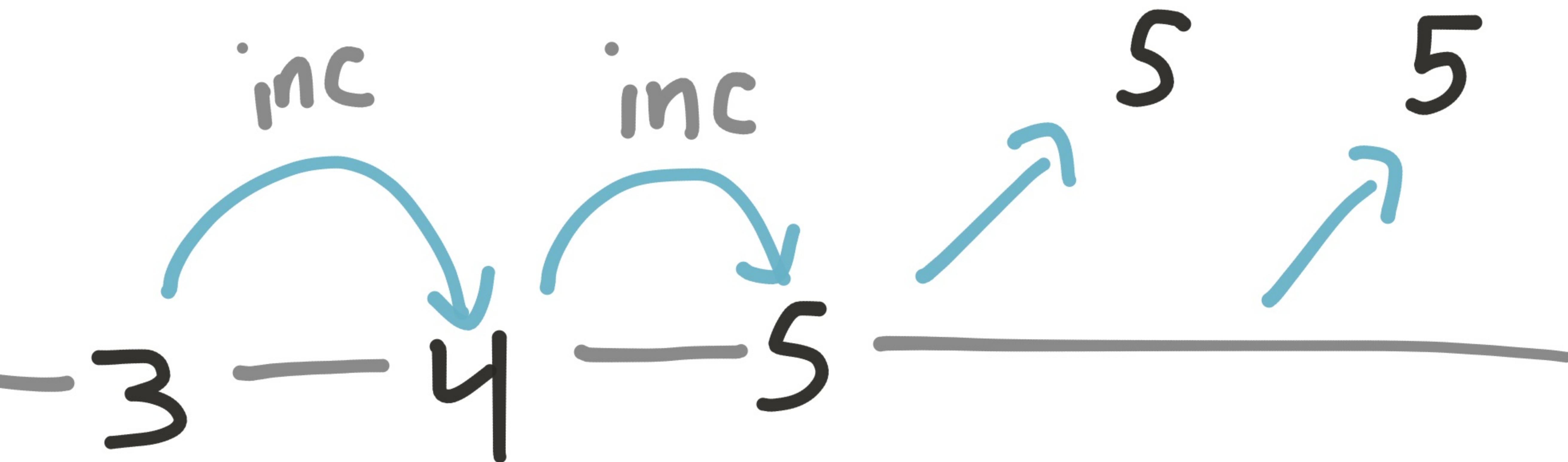
$$2 - 3$$

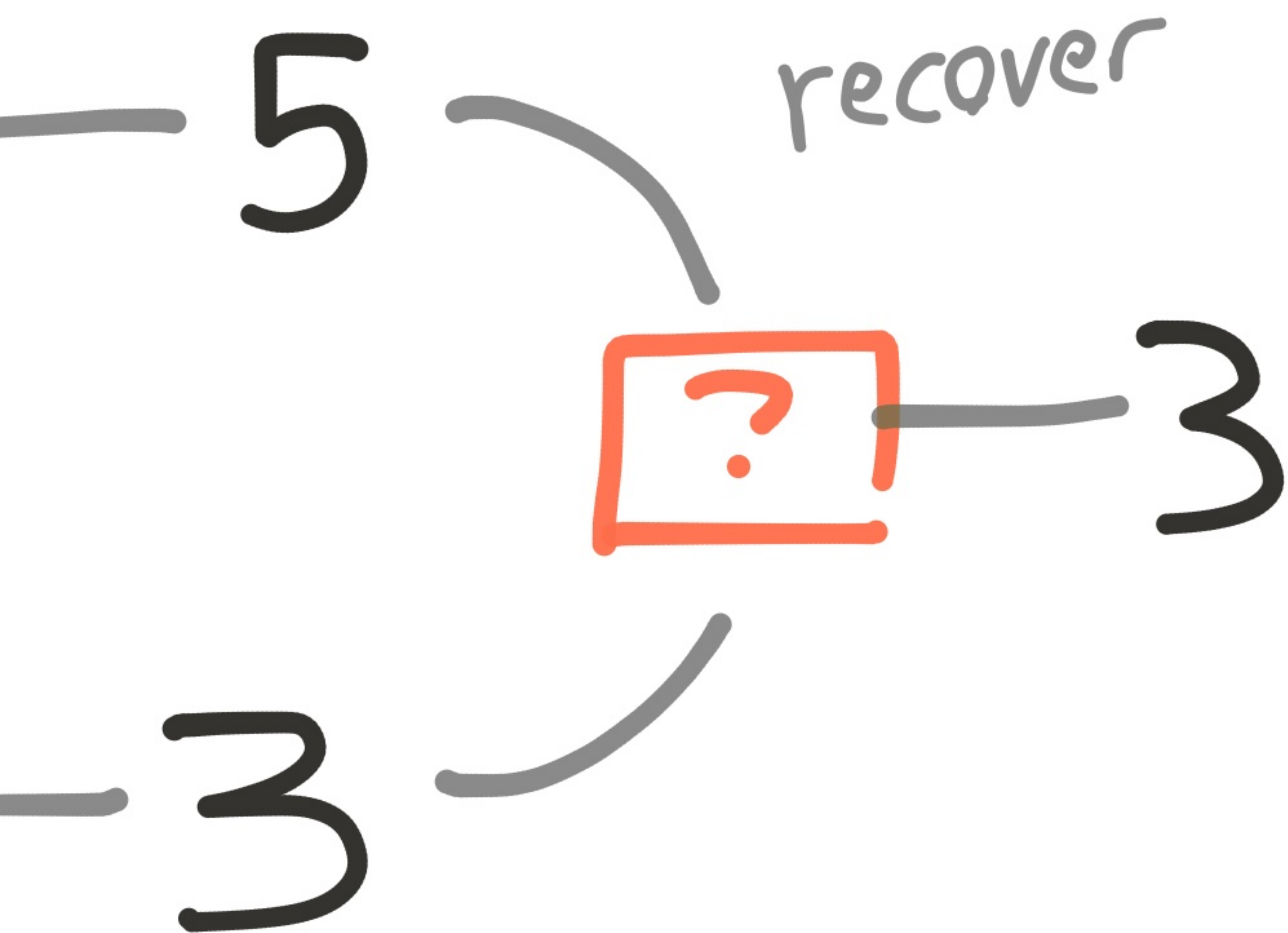
increment

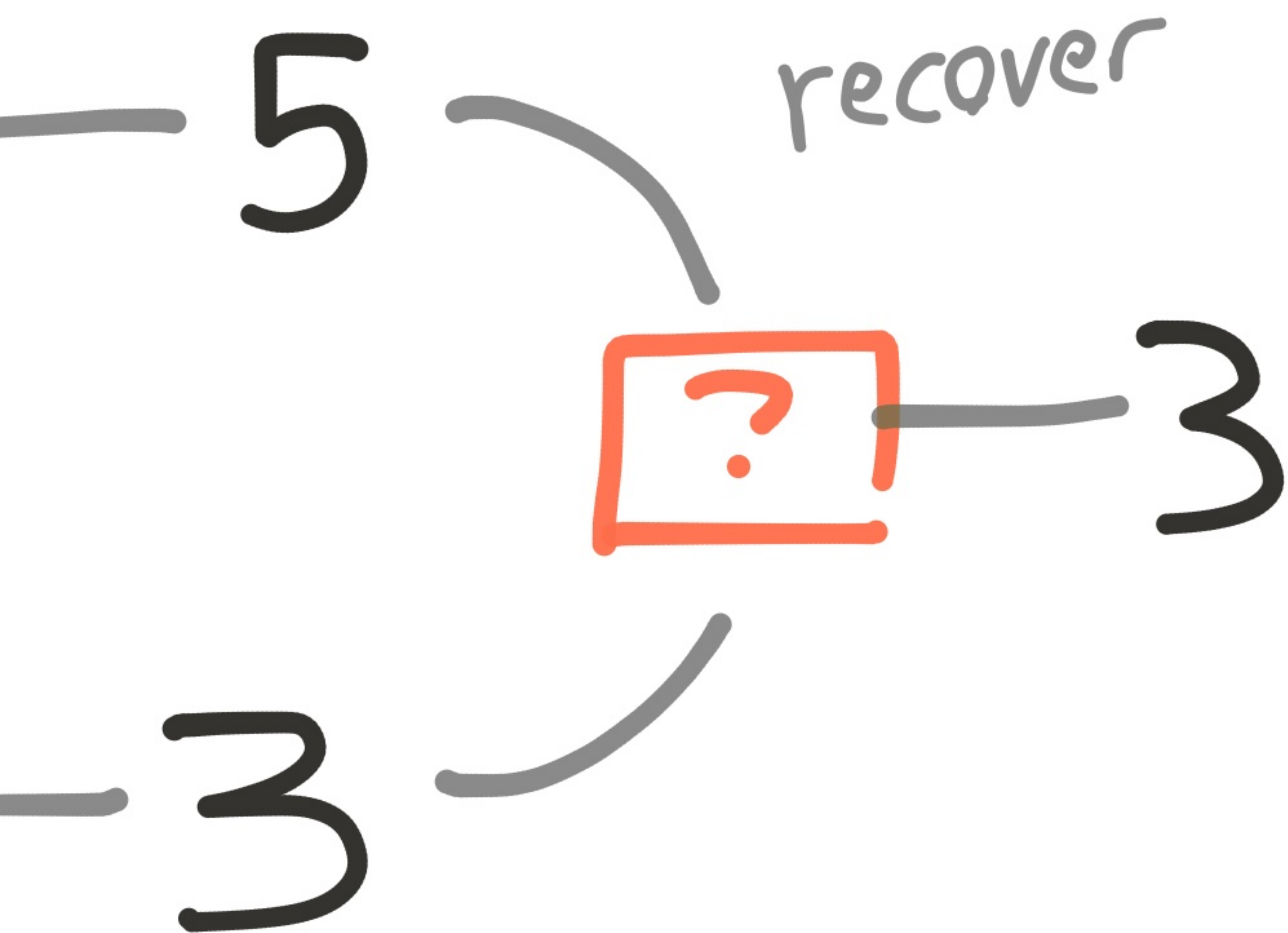
$$2 + 1 = 3$$

- 2

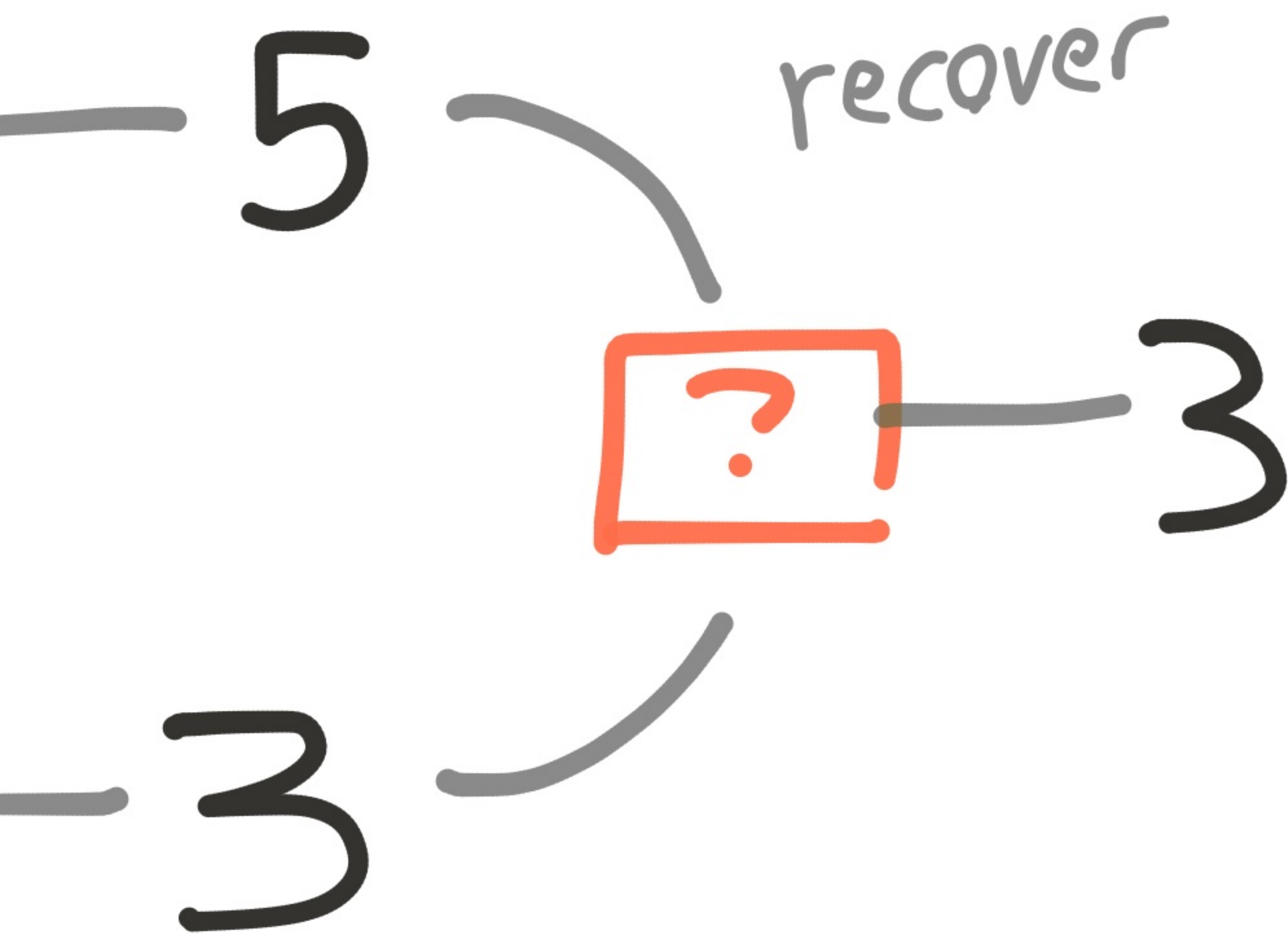


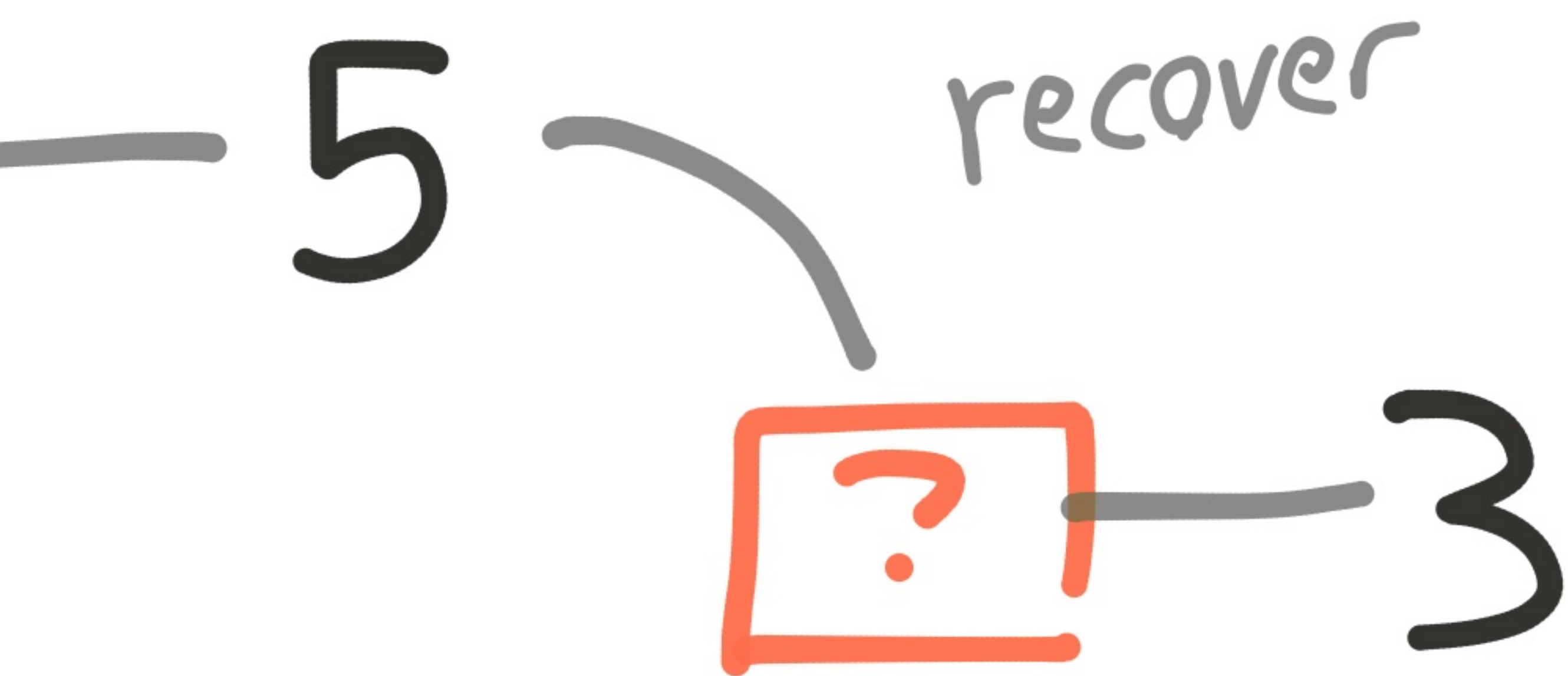






LOOK AT
—your—
LIFE





LOOK AT
—your—
LIFE



ID Generator

- Same deal

~91,000 dup ids in 30s

834,000 generated

Maps

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

nil $\xrightarrow{\text{putIfAbsent}}$ 3 $\xrightarrow{\text{replace}}$ 5

Last 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

REAL



3.999.0.3

Distributed, sharded
{ Document }

store

3.5.4:

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

2 Years of
fundamental redesign
work...

4.0: Linearizable Mode

- Custom Consensus protocol
- Reduced loss in AP mode

(start
of
test)

read 2

fail

write 2

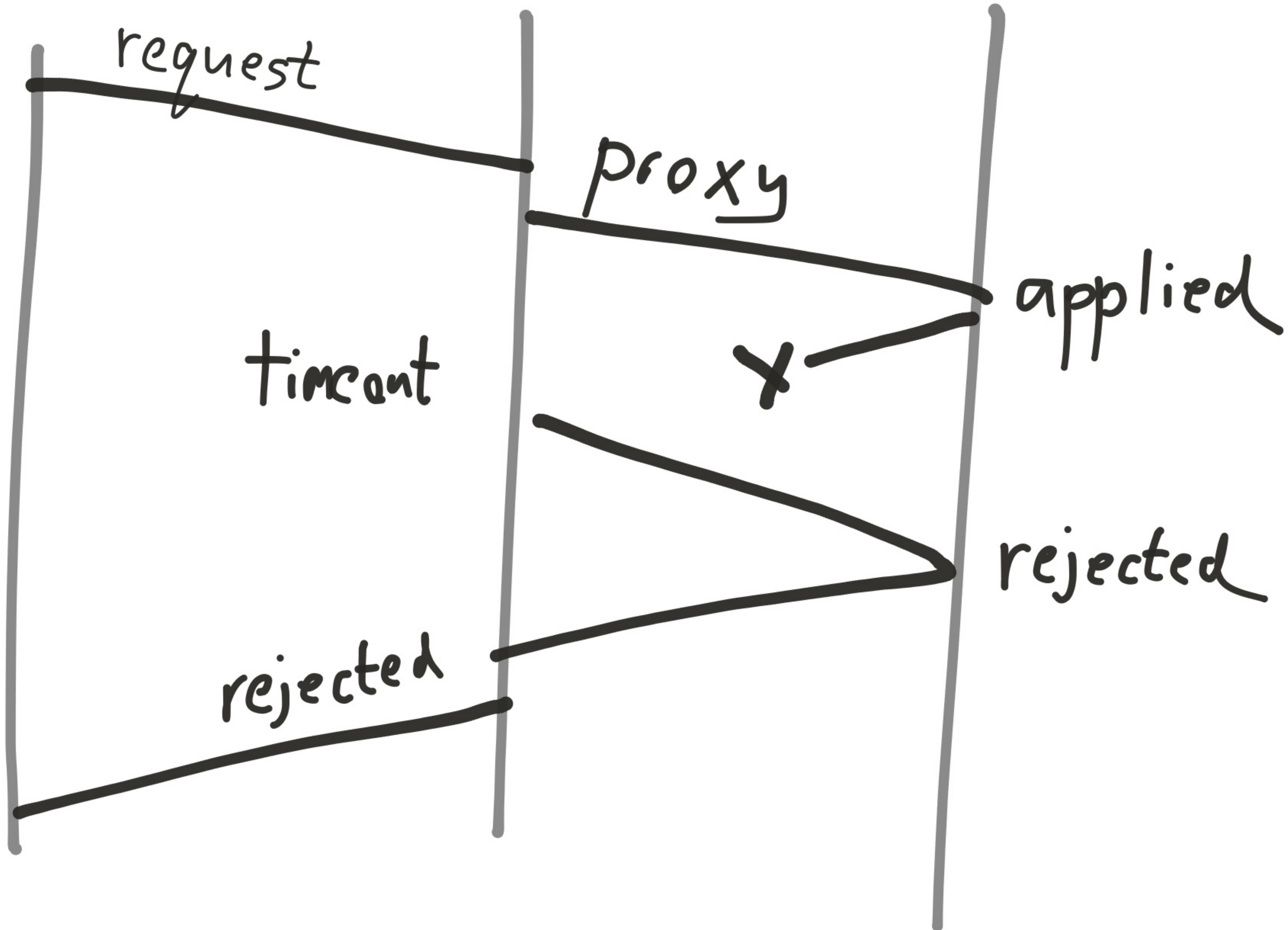
ok

read 2

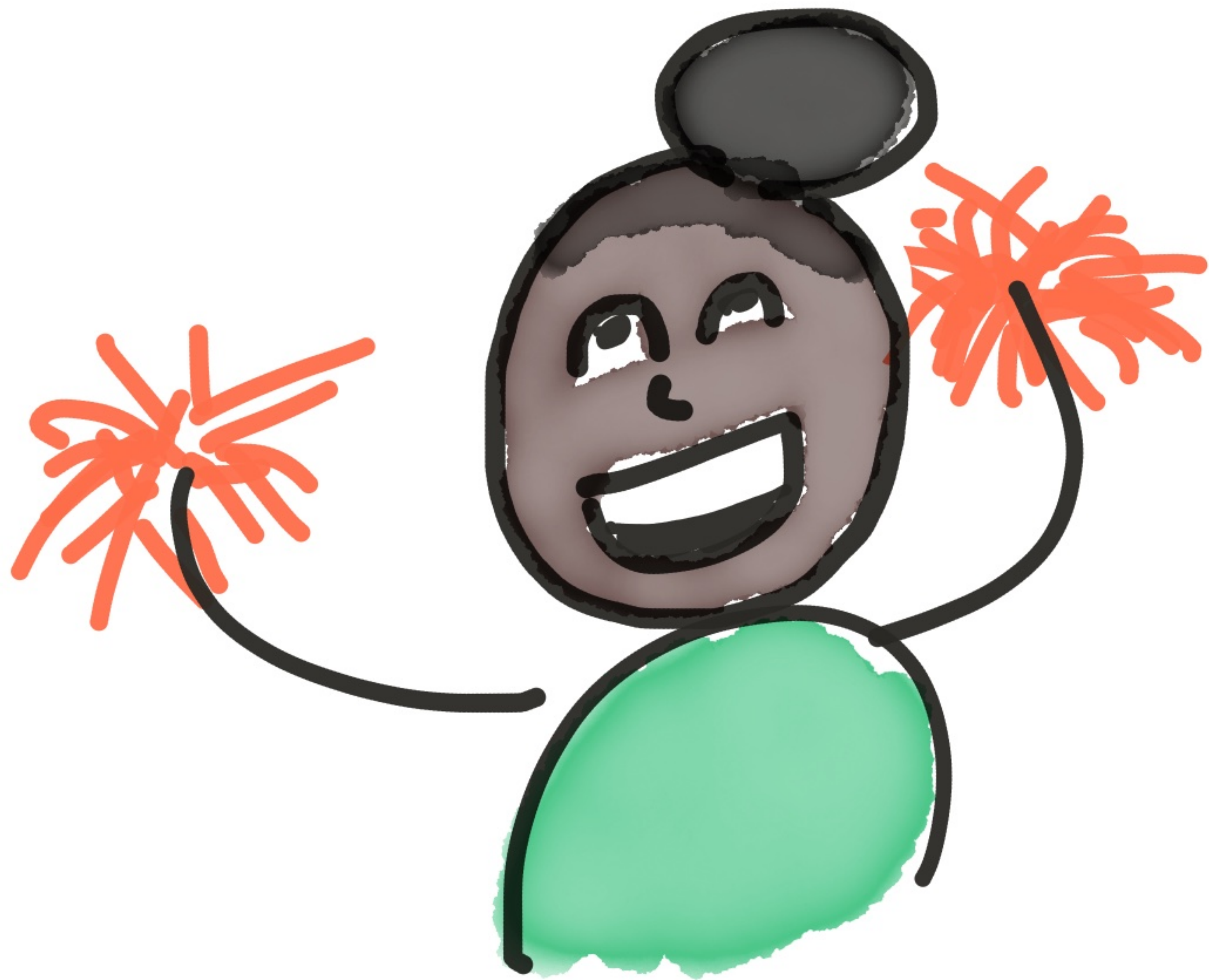
client

server

server



fixed in 3.99.1.5



Node crashes



client

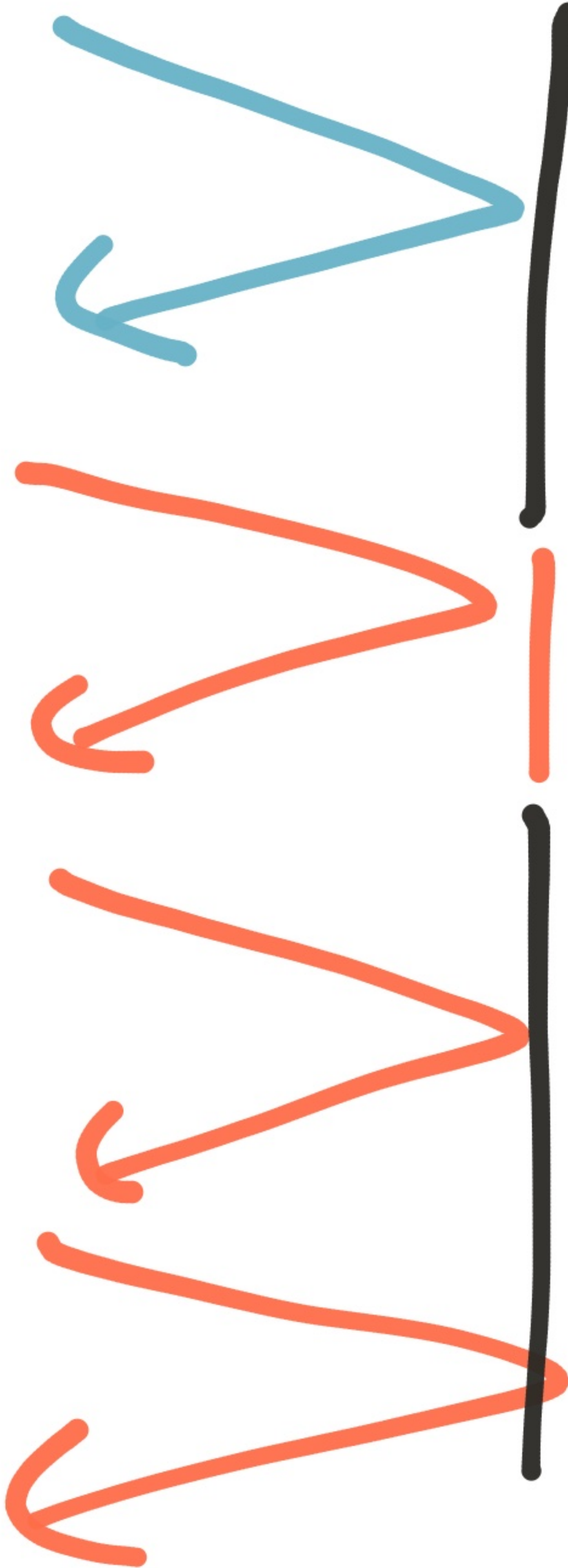
s

s

s

s

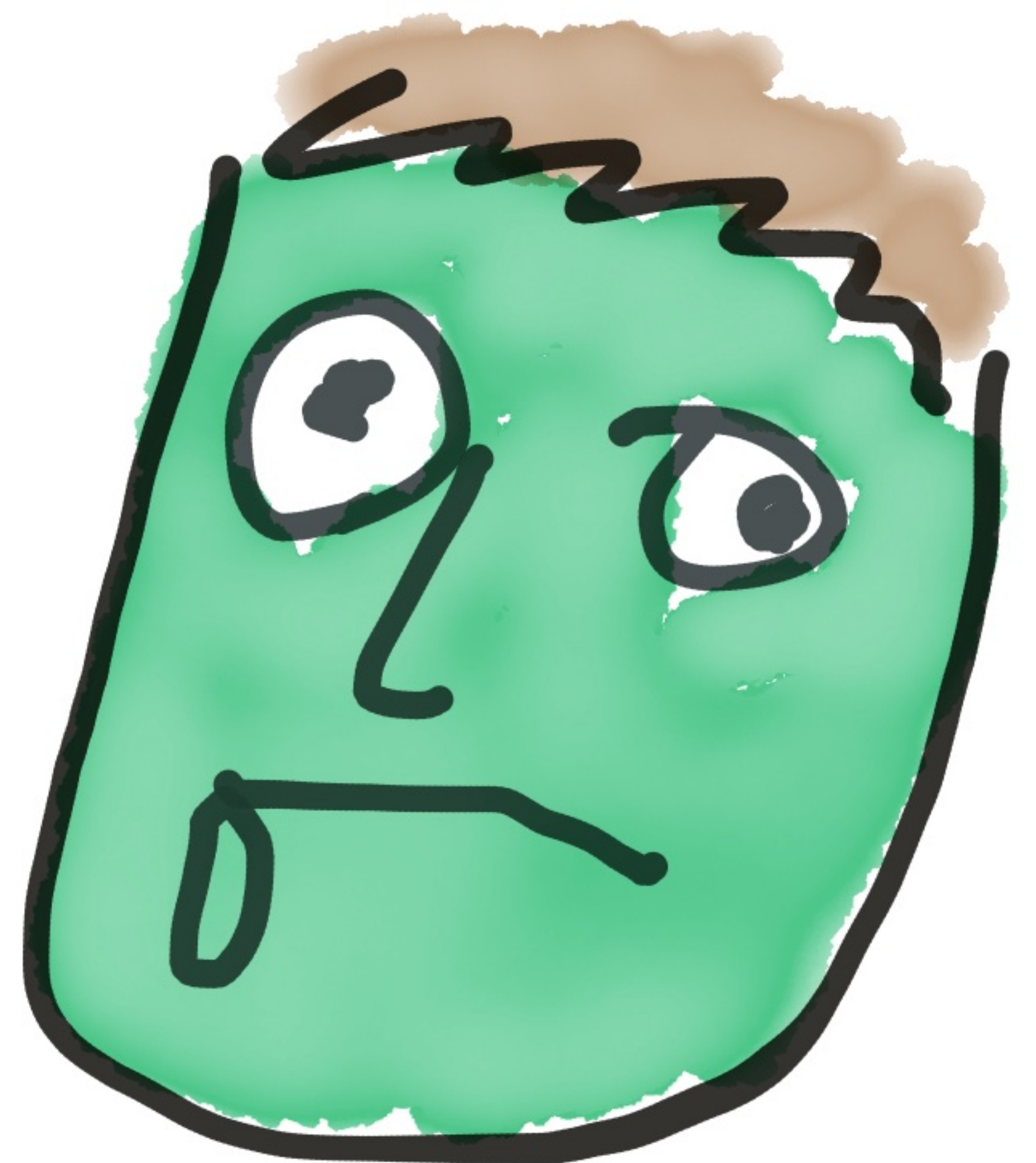
s



[press E to revive]

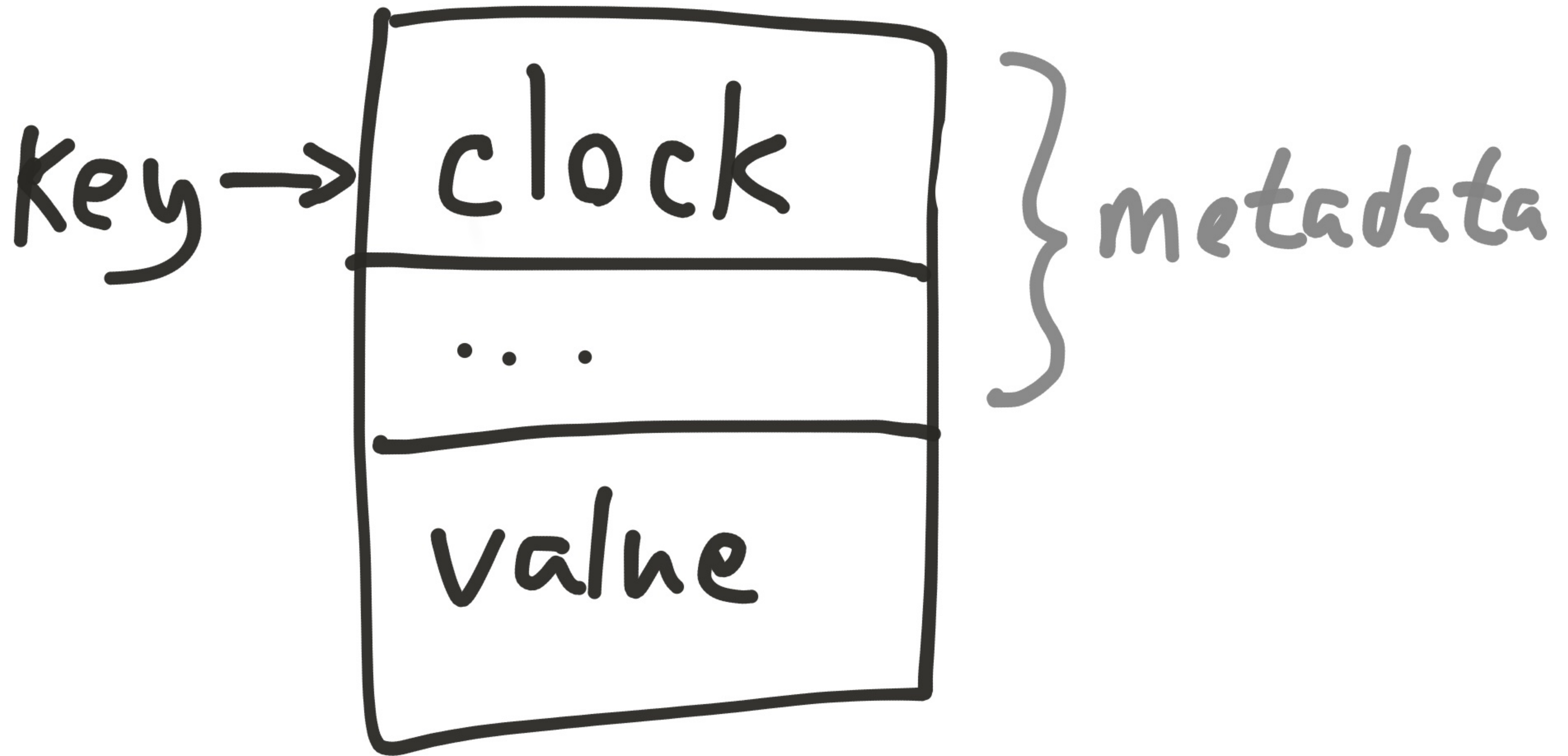


Dead partitions
may not be all
there...

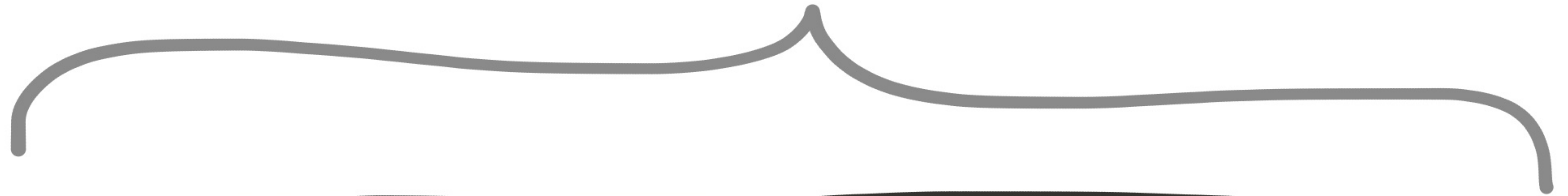


AS didn't flush
writes to disk
before acknowledging
(fixed in 3.99.2.1)

Clock Skew can
lead to Lost
updates



clock



regime

wall-clock

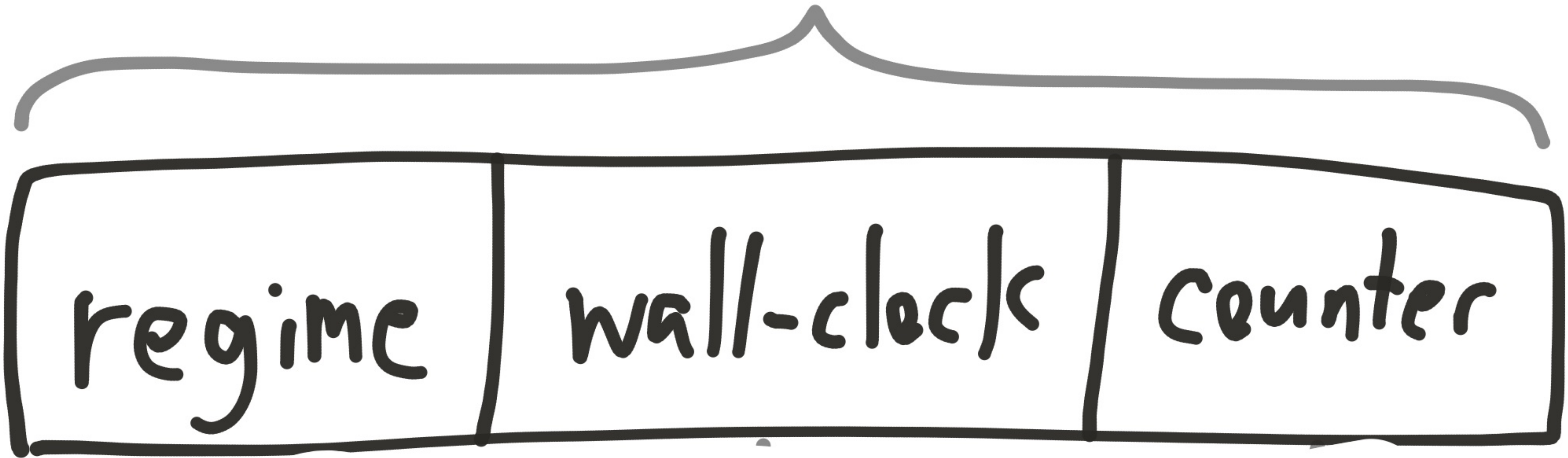
counter

increment
on
election

node
local

for mult.
updates in
same clock

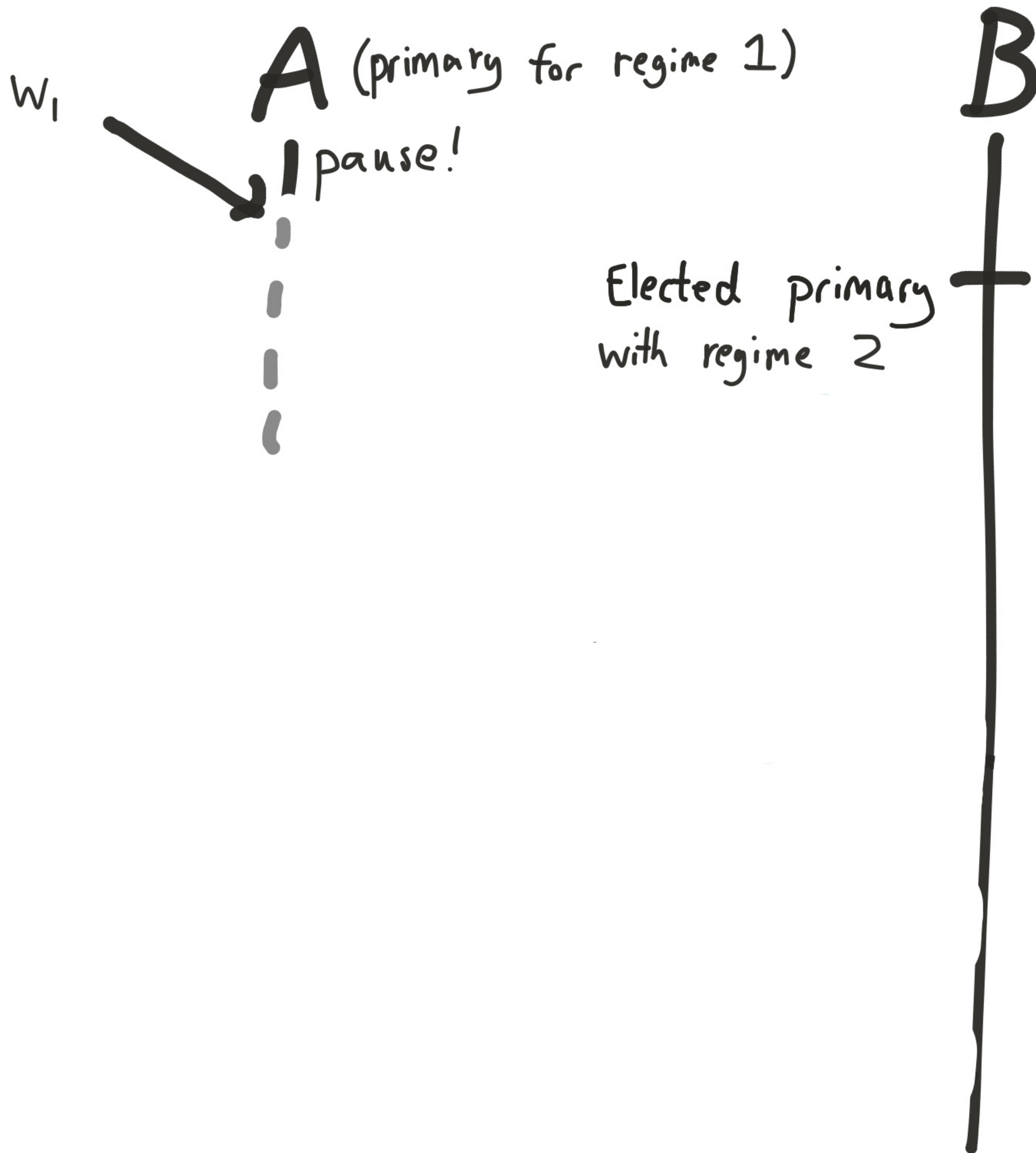
fits in cache line

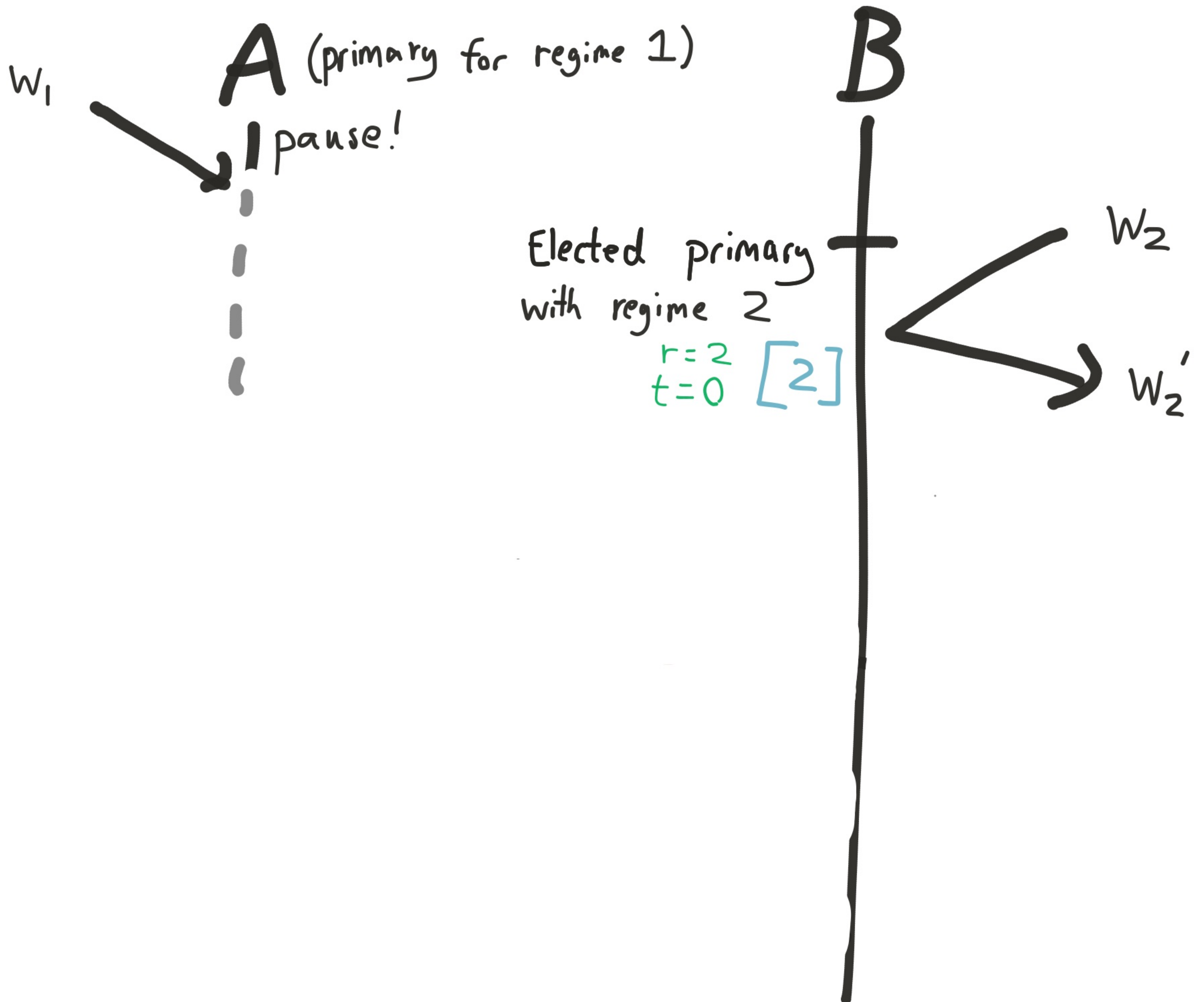


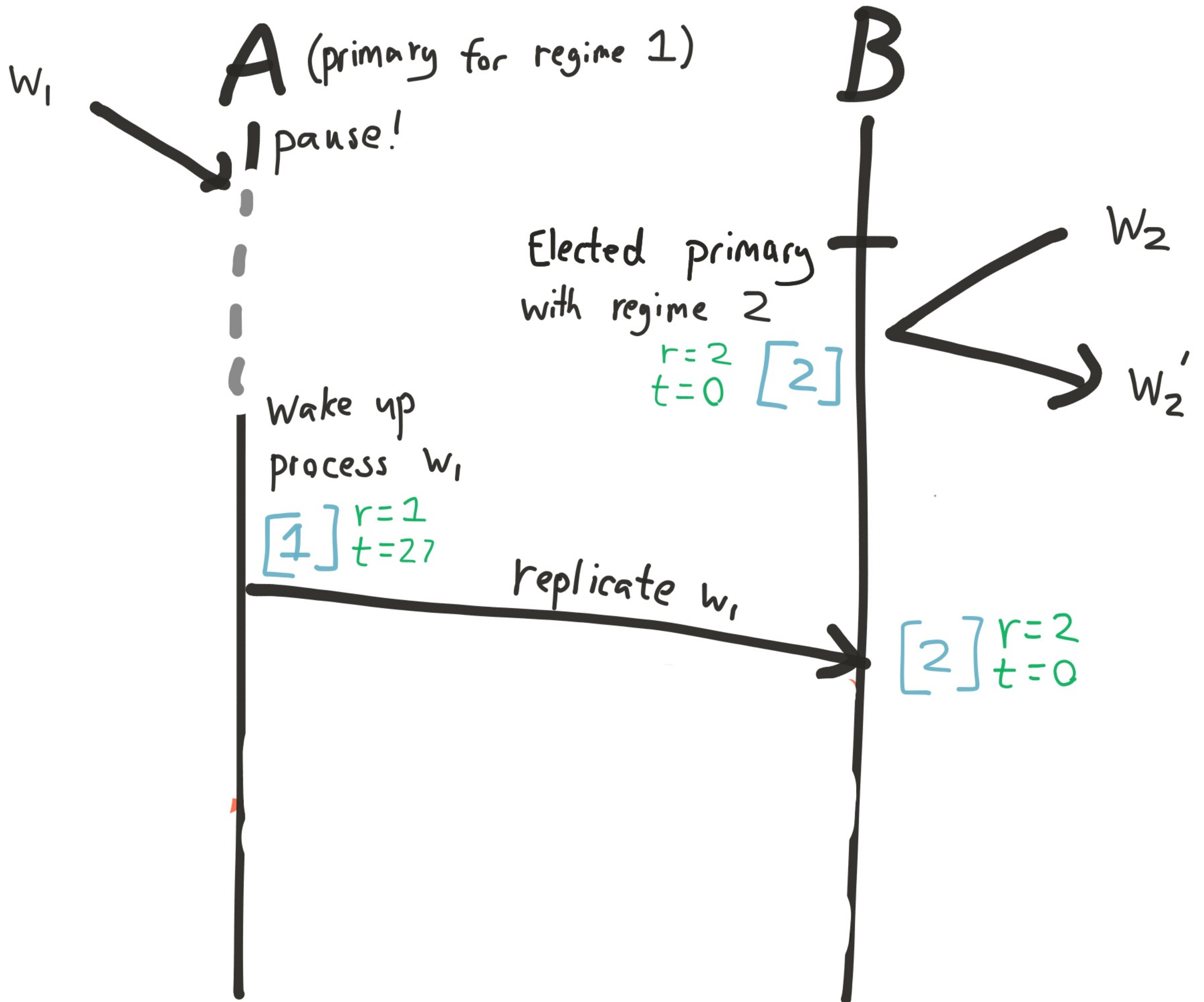
6 bits

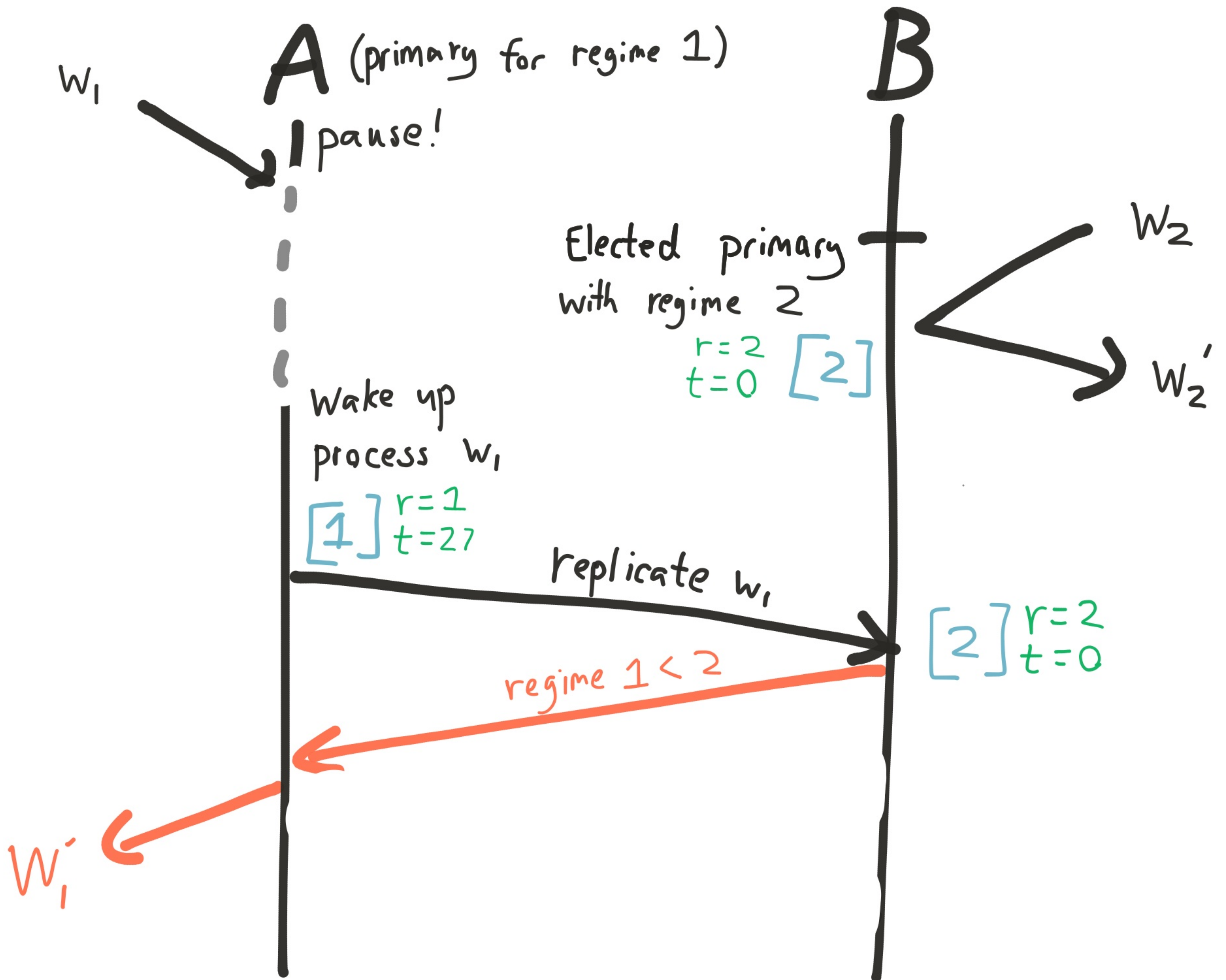
w_1 **A** (primary for regime 1)
! pause!
-

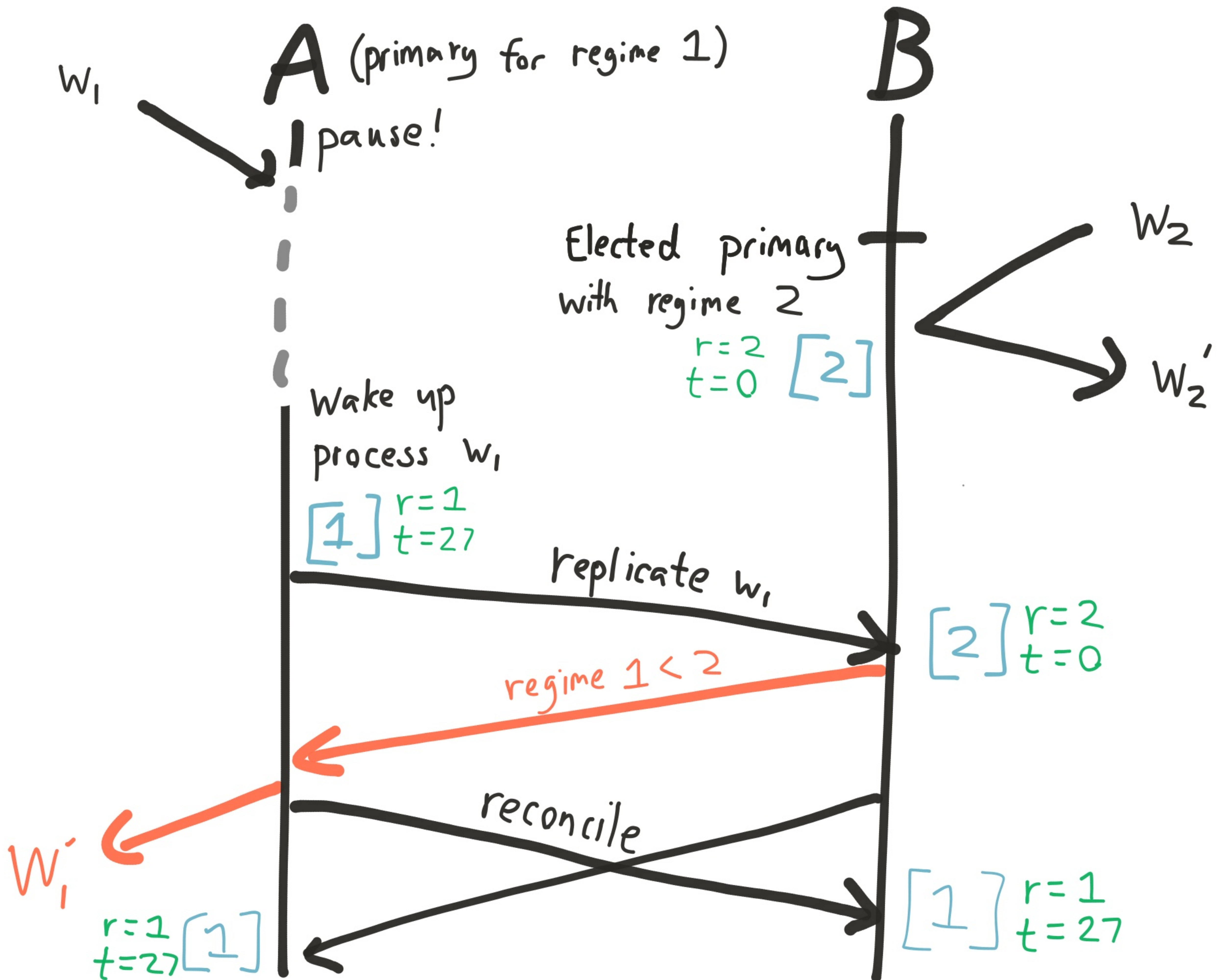
B
-











REALLY?



Both clock skew &

that using fsync were

known failure modes

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

looks linearizable!

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

future work!

Recap

Read the docs!



Then test it

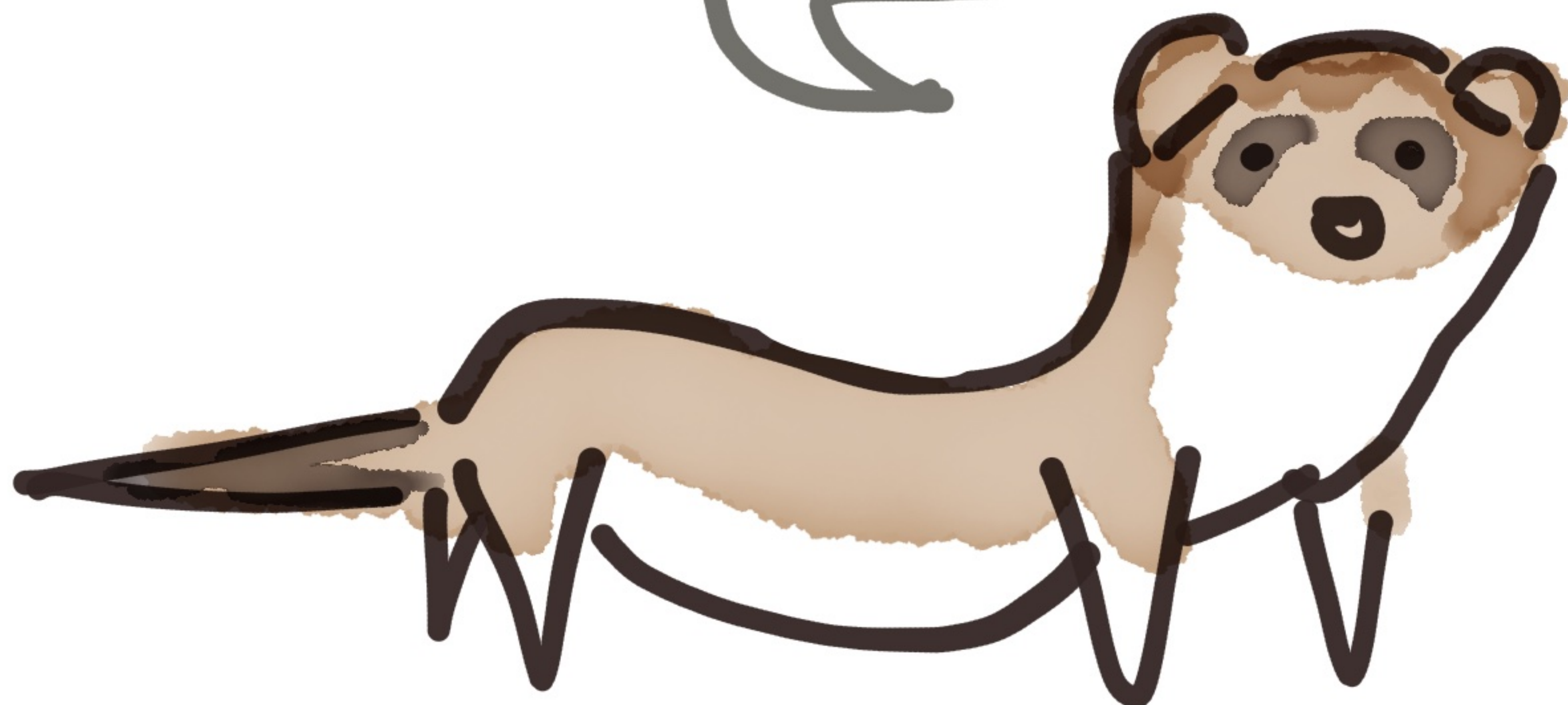
— for —

YOURSELF

"Strict"

"ACID"

"Strang"



Be Formal

&

Be Specific. !

Figure out the
Invariants
your system needs

Consider
your
failure modes

Process Crash

#kill -9 1234

Node failure

- AWS terminate
- Physical power switch

Clock Skew

date 10280000

fake time ...

GC/IO Pause

killall -s STOP foo

killall -s CONT foo

Network Partition

iptables -j DROP

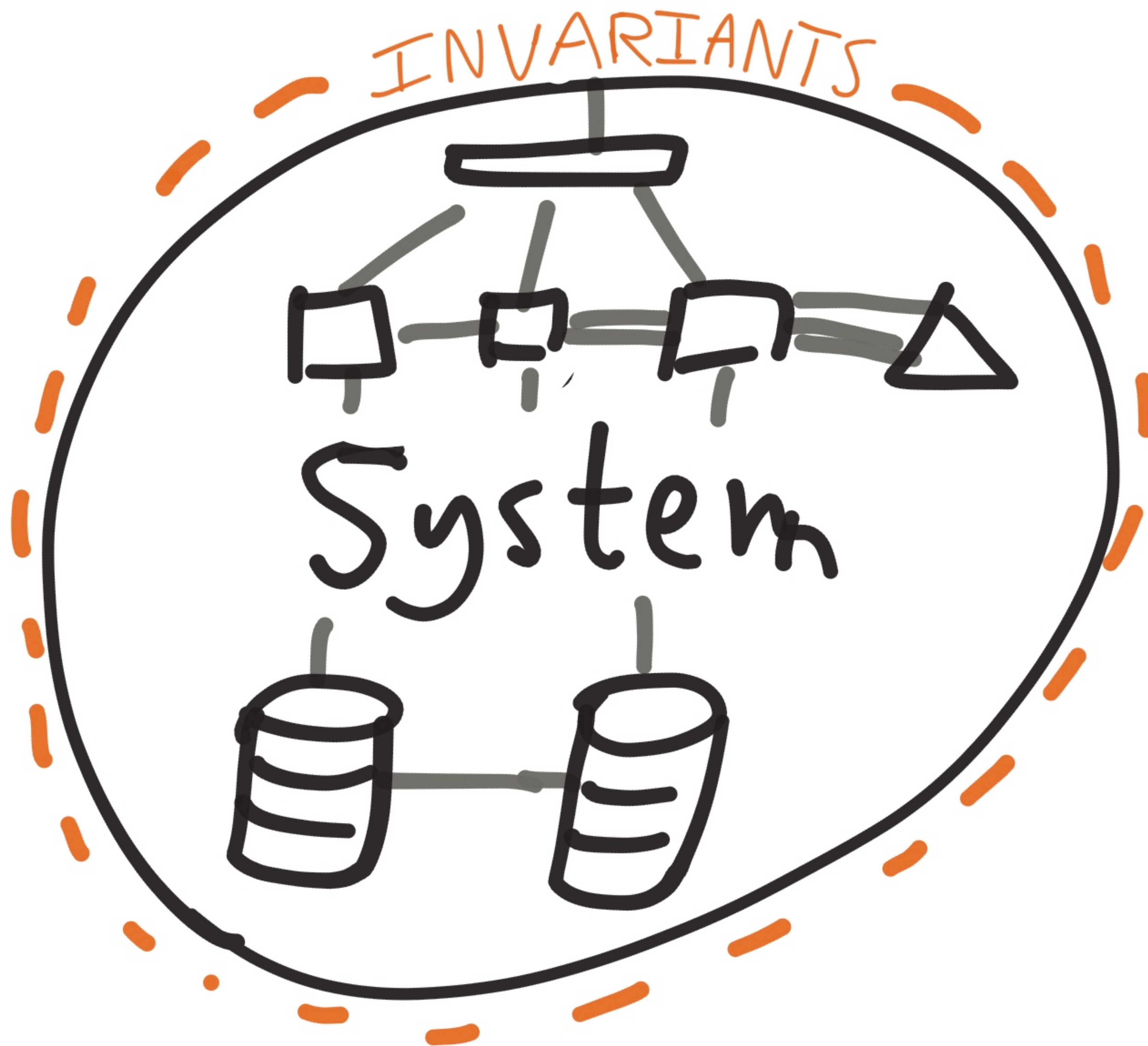
tc qdisc ... delay ...
drop ...

Test your
Systems

end ————— to —————> end



VS.



If you look for

✦ perfection ✦

you will never be
content

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

Thanks

Tendermint



Hazelcast

Jordan Haltermann

Luigi Dell'Aquila

Luca Garulli

Thanks

Denis Sukhoreslov

Aerospike



Peter Alvara



<http://jepson.io>