pod:
a latency-optimal layer 1

Dionysis Zindros

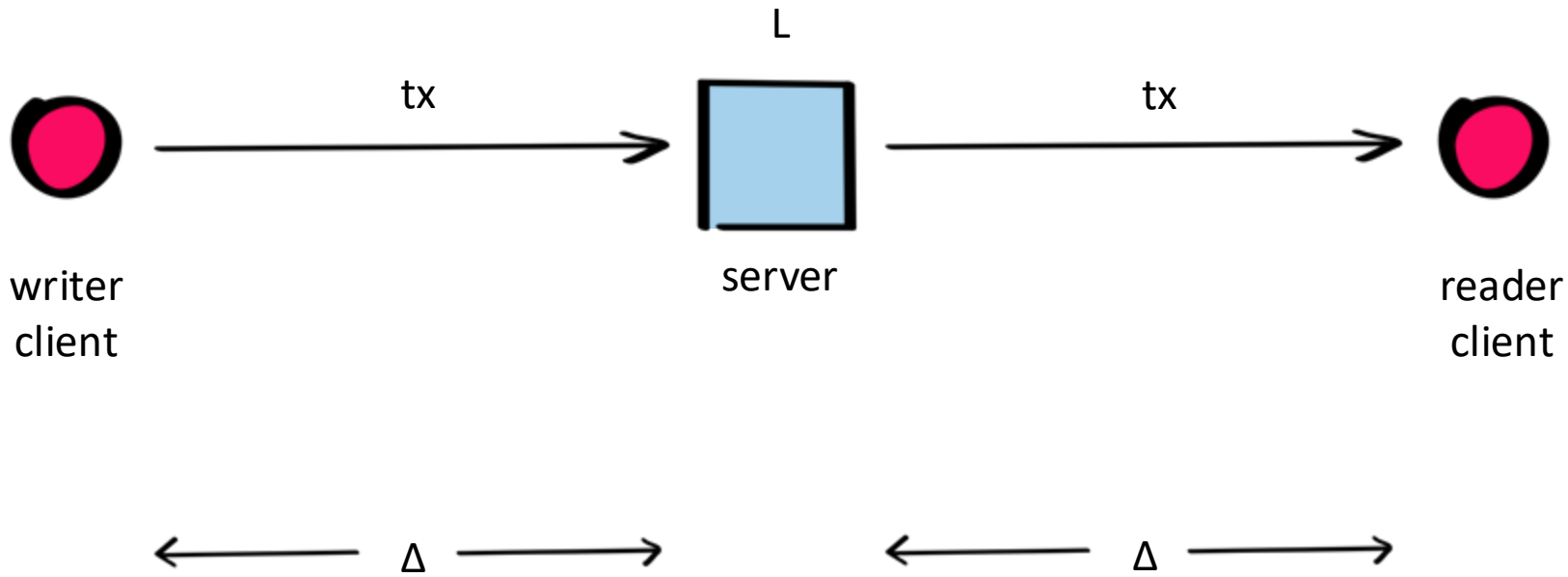
Common Prefix

Orestis Alpos

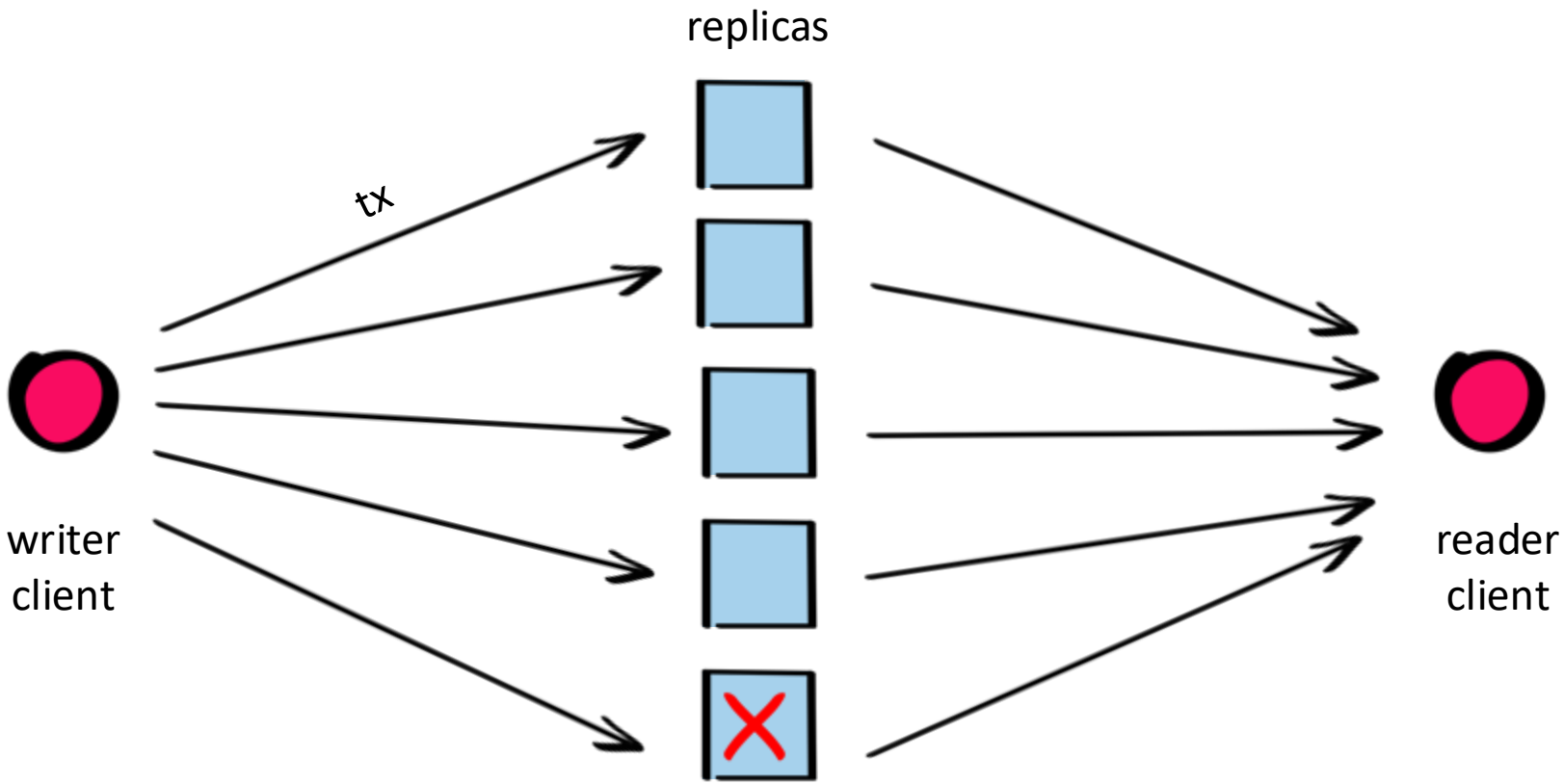
Bernardo David

May 27th, 2025. TUM Blockchain & Cybersecurity Salon

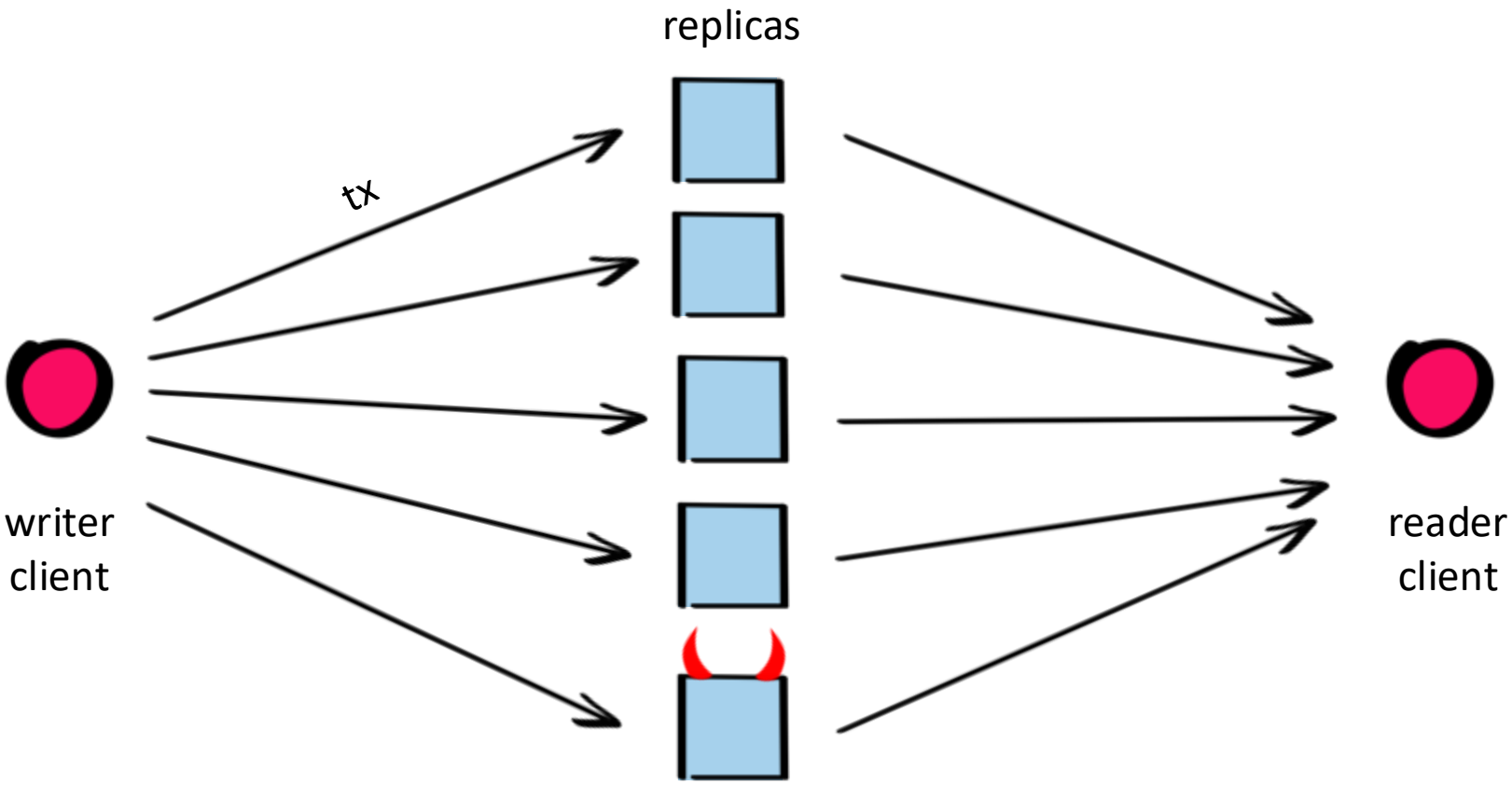
Optimal Latency



Crash Resilience



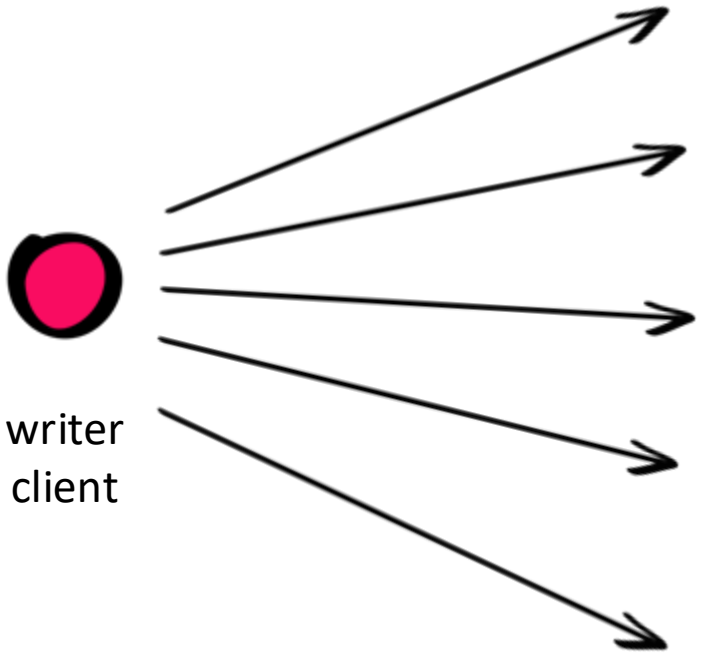
Byzantine Resilience



pod design principles

1. Optimal latency of 2Δ
2. Replicas do not communicate
3. Byzantine resilient
4. Replicas are lazy: log but do not execute
5. No blocks, no chains
6. Streaming: Push rather than pull

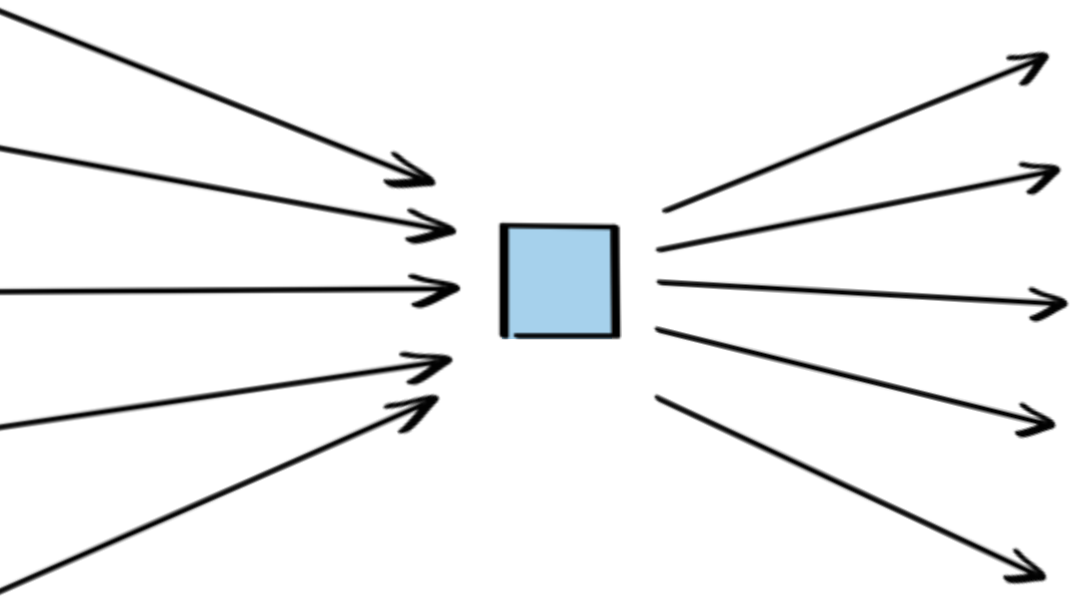
The Writer*



- Keep connection to all replicas
- Sign payment transaction
- Broadcast to all replicas

* In practice, all clients are readers and writers. We distinguish the two functionalities for simplicity.

The Replica



- Maintain connection to all clients
- Maintain local log L
- When tx is received from writer append it to the log:

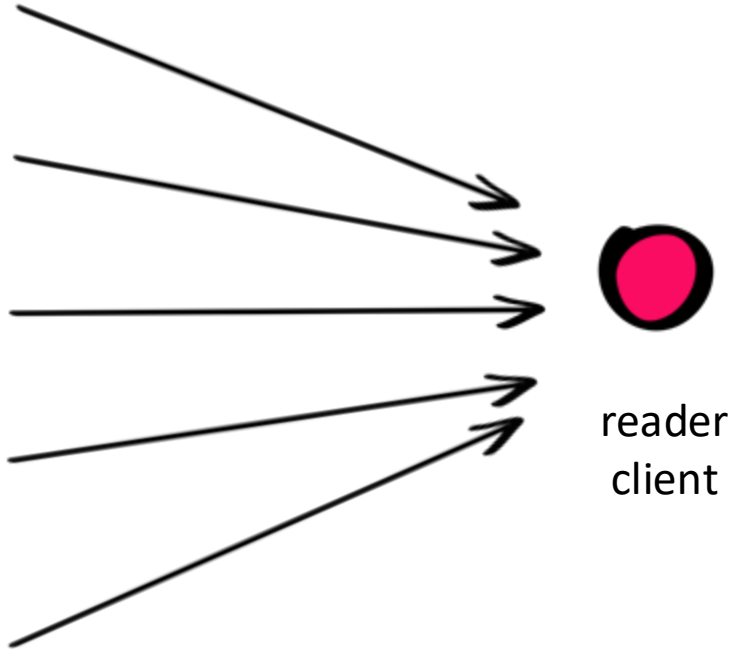
$L = L \,||\, (tx, ts)$

- Sign L and send it to readers:

$\sigma = sign(sk, L)$

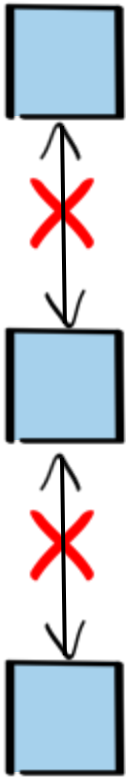
The Reader

- Keep connection to all replicas
- Receive signed logs
- Confirm transaction when: 4n/5 of replicas have included it in their logs



Resilience is f < n/5

incommunicado



- Replicas don’t communicate
- This allows us to avoid roundtrips & maintain 2Δ latency
- But this means that each log is different…

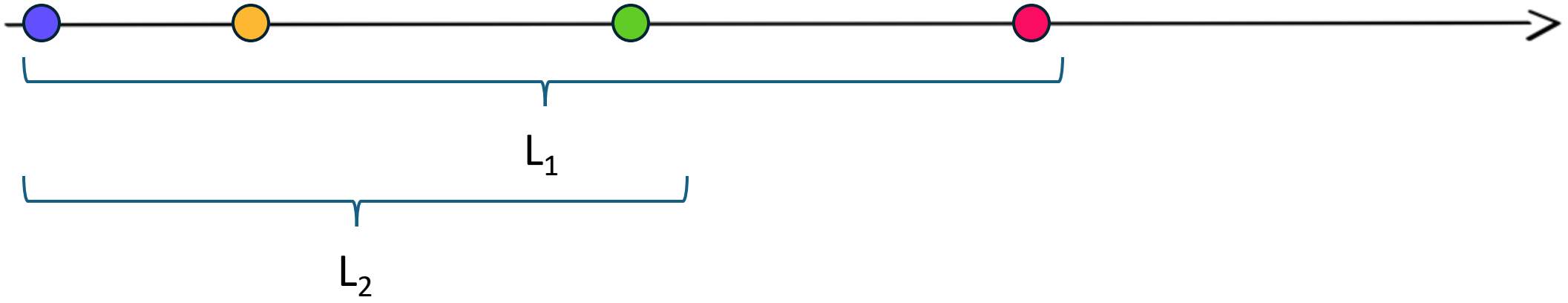
State Machine Replication (“consensus”)

- **Liveness**: An honest transaction gets eventually confirmed

We achieve this if $f < n/5$.

State Machine Replication (“consensus”)

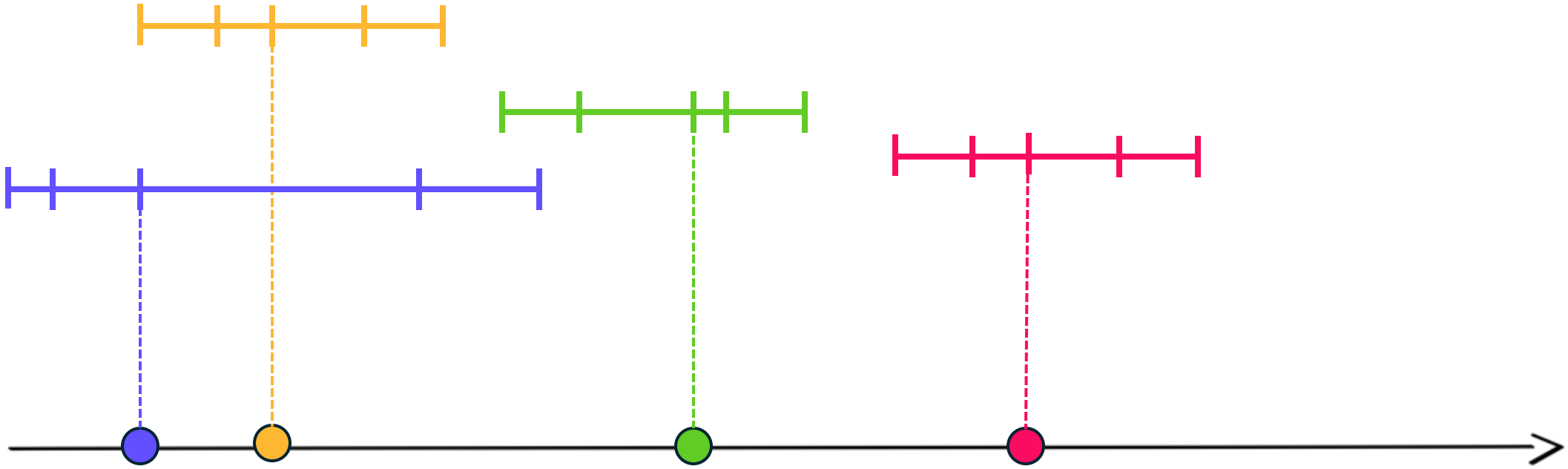
- **Safety**: Two honest replicas report logs that are prefixes of each other. $L_1 \preccurlyeq L_2$ or $L_2 \preccurlyeq L_1$.



…we do *not* achieve this! **(Impossible at 2Δ latency)**

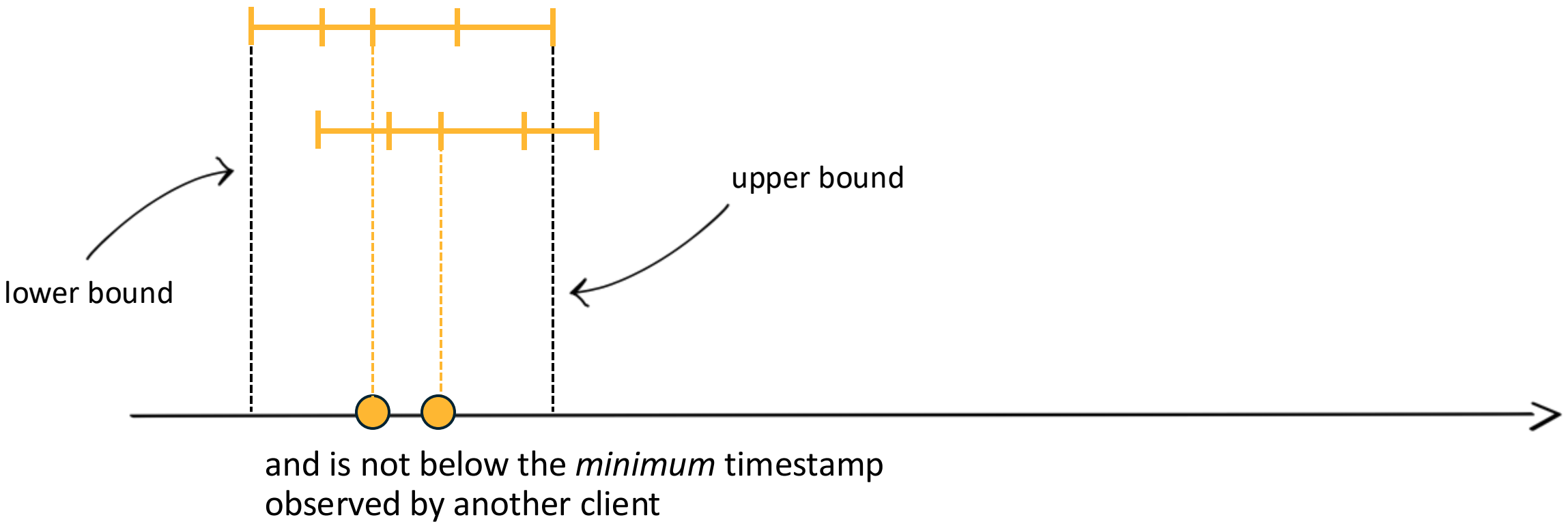
Generalized ledgers

Client determines position of each tx on timeline by taking *median* timestamp

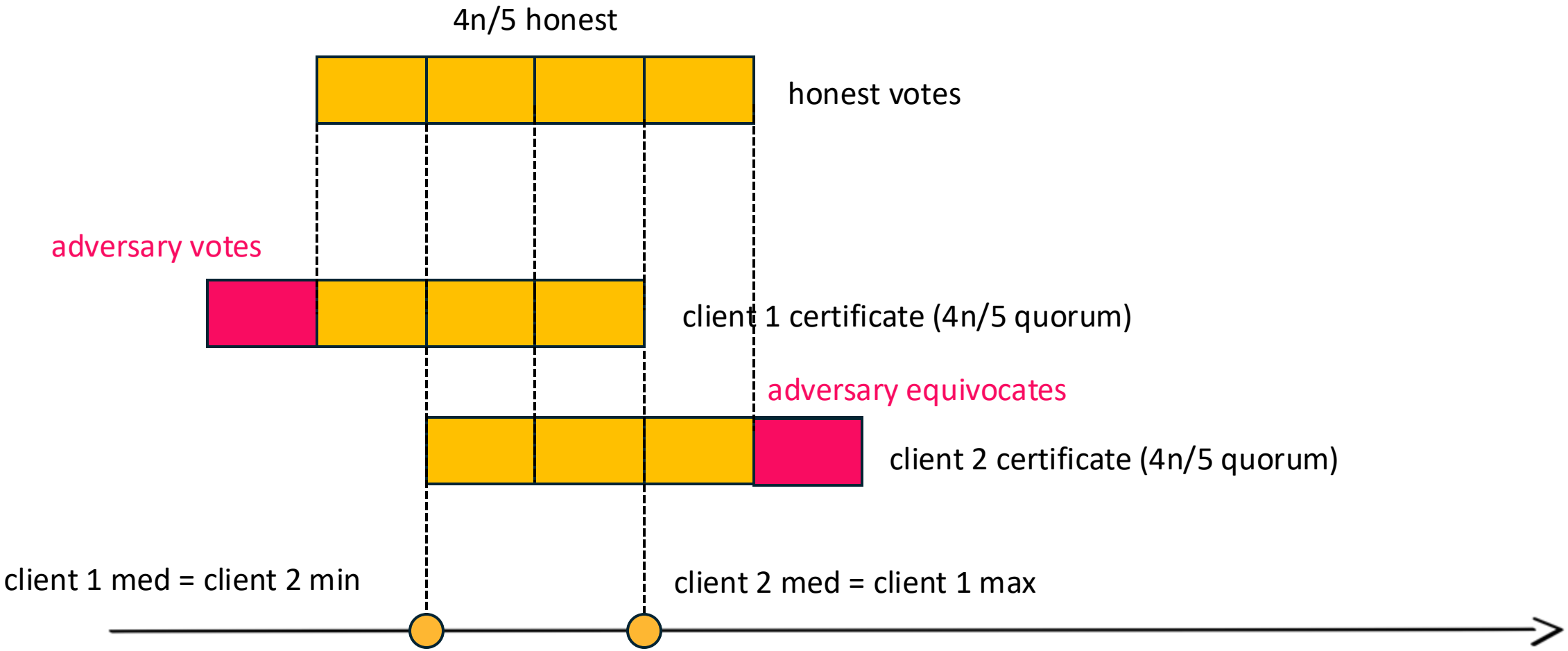


Generalized safety

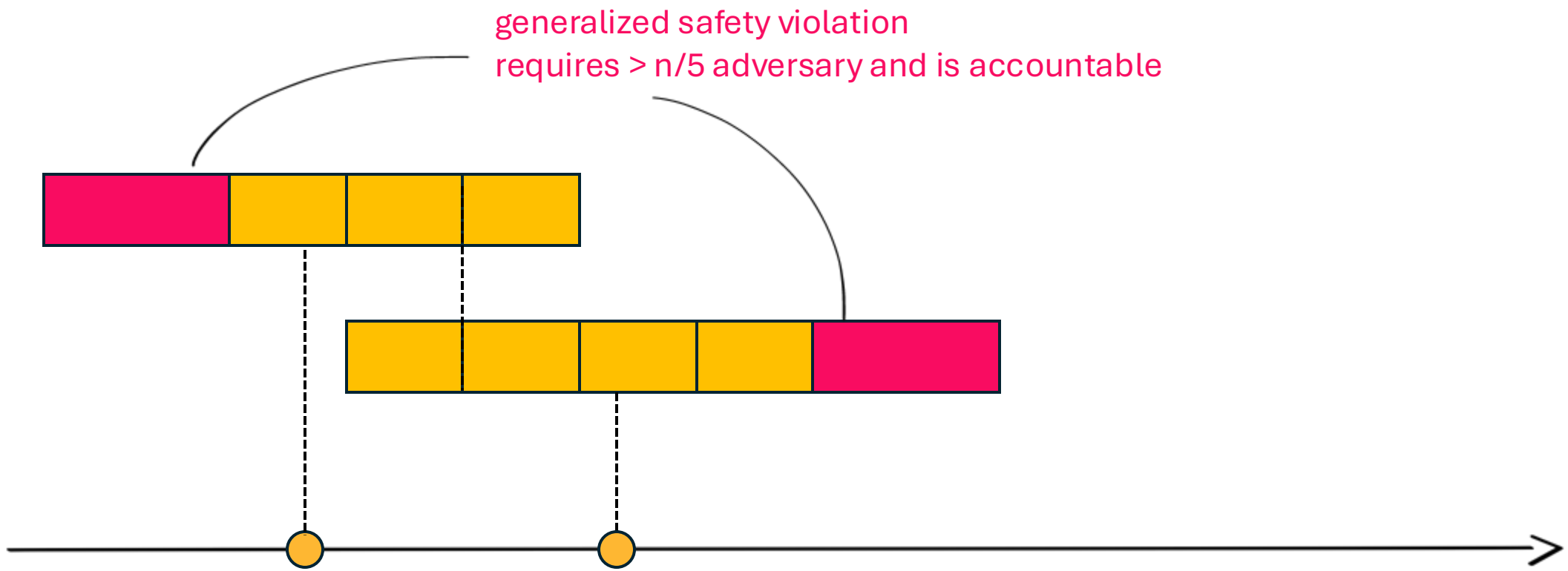
The *confirmation* timestamp of a tx in the view of one client does not exceed the *maximum* timestamp observed by another client



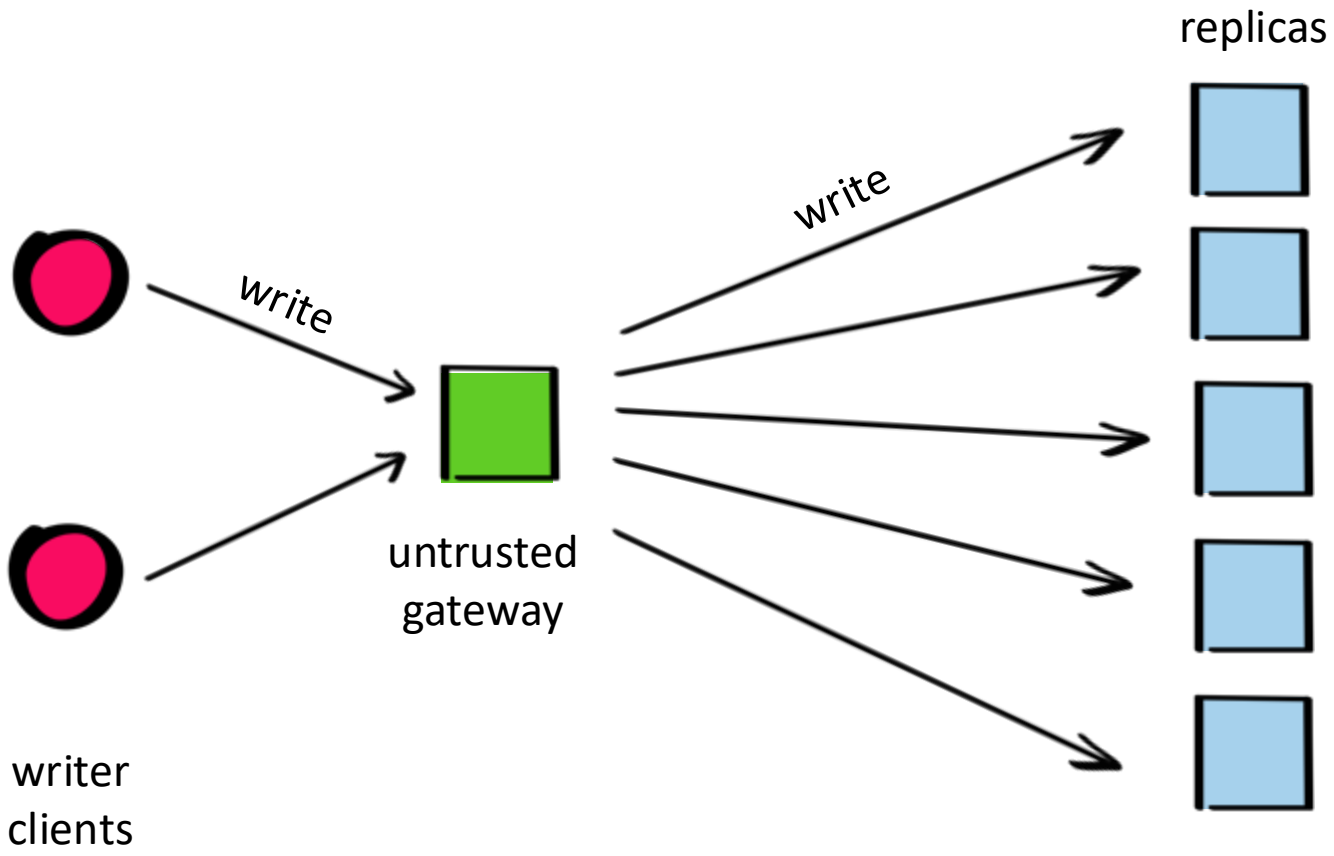
Proof sketch



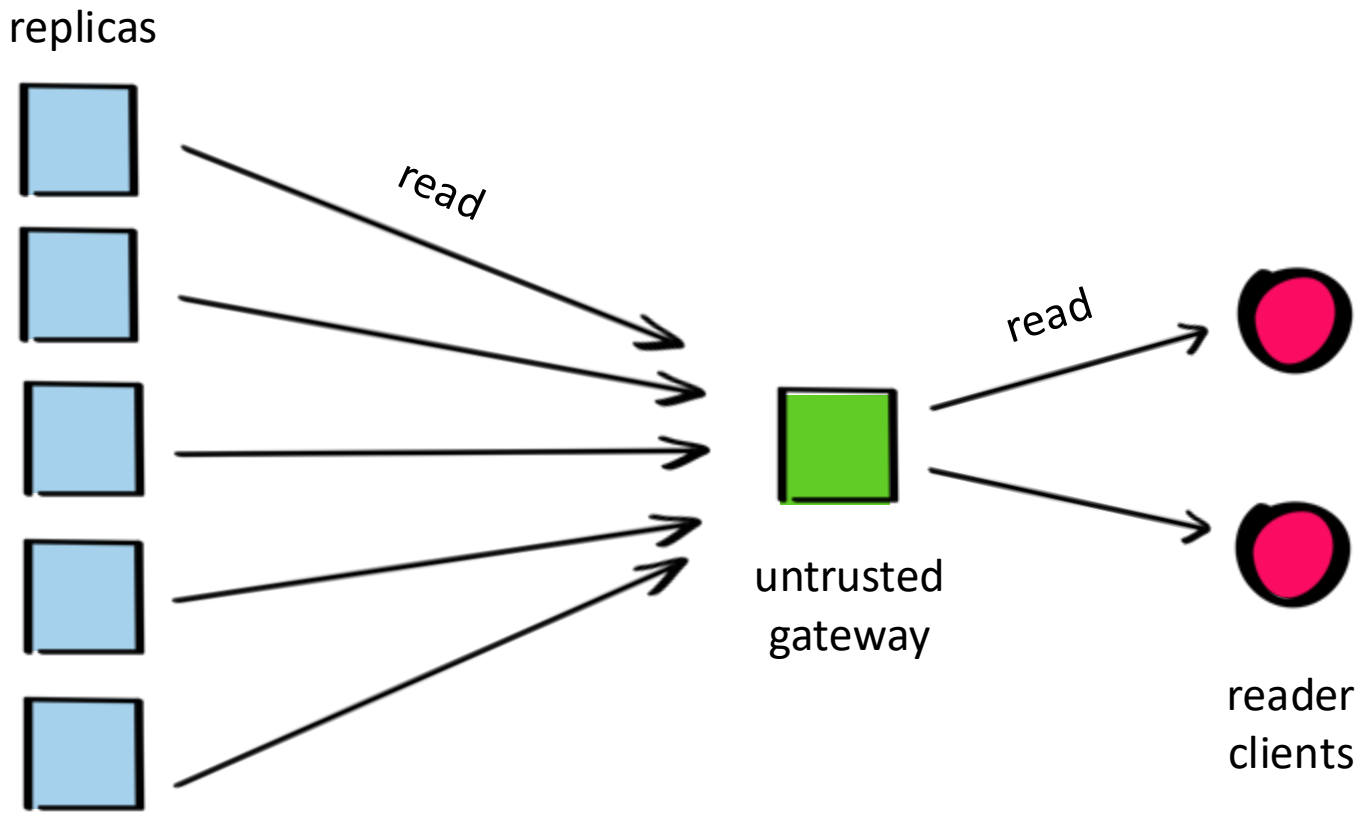
Accountability



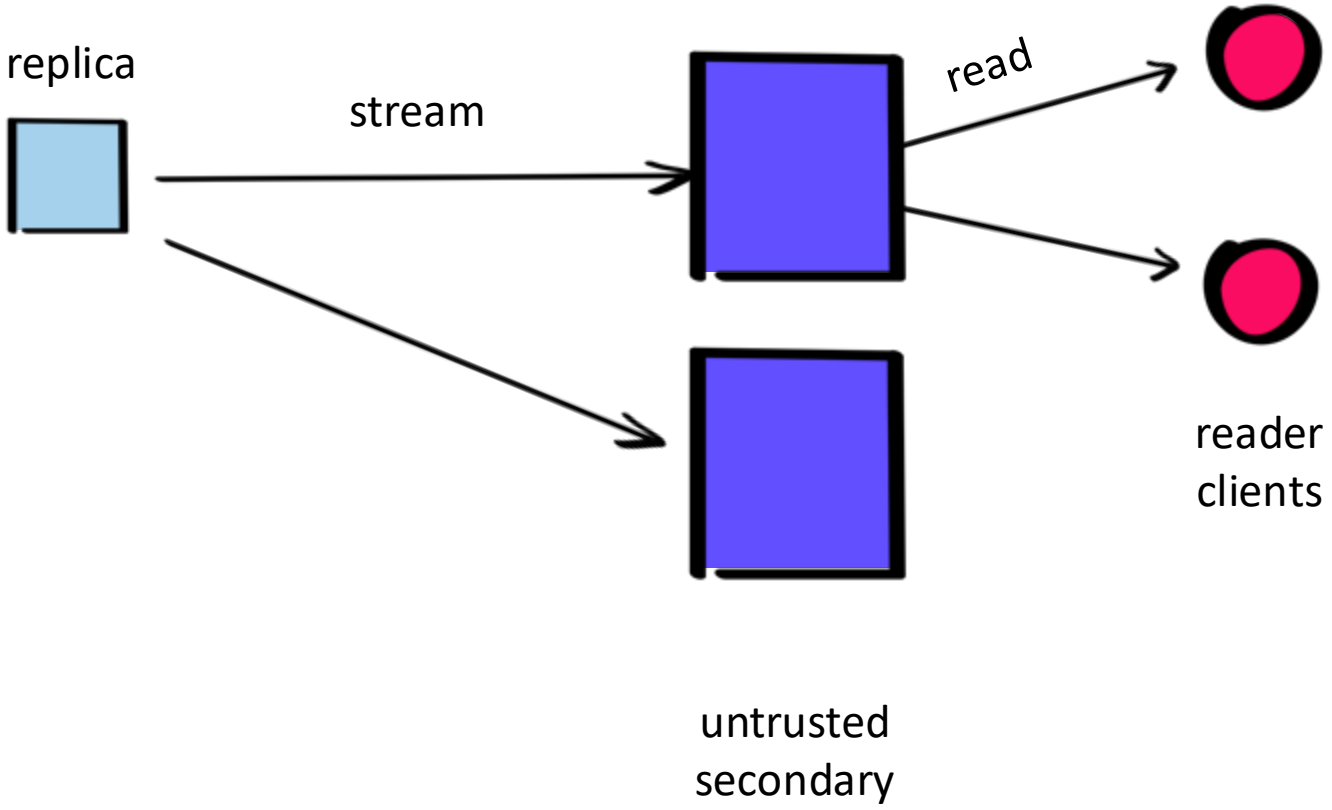
Gateways



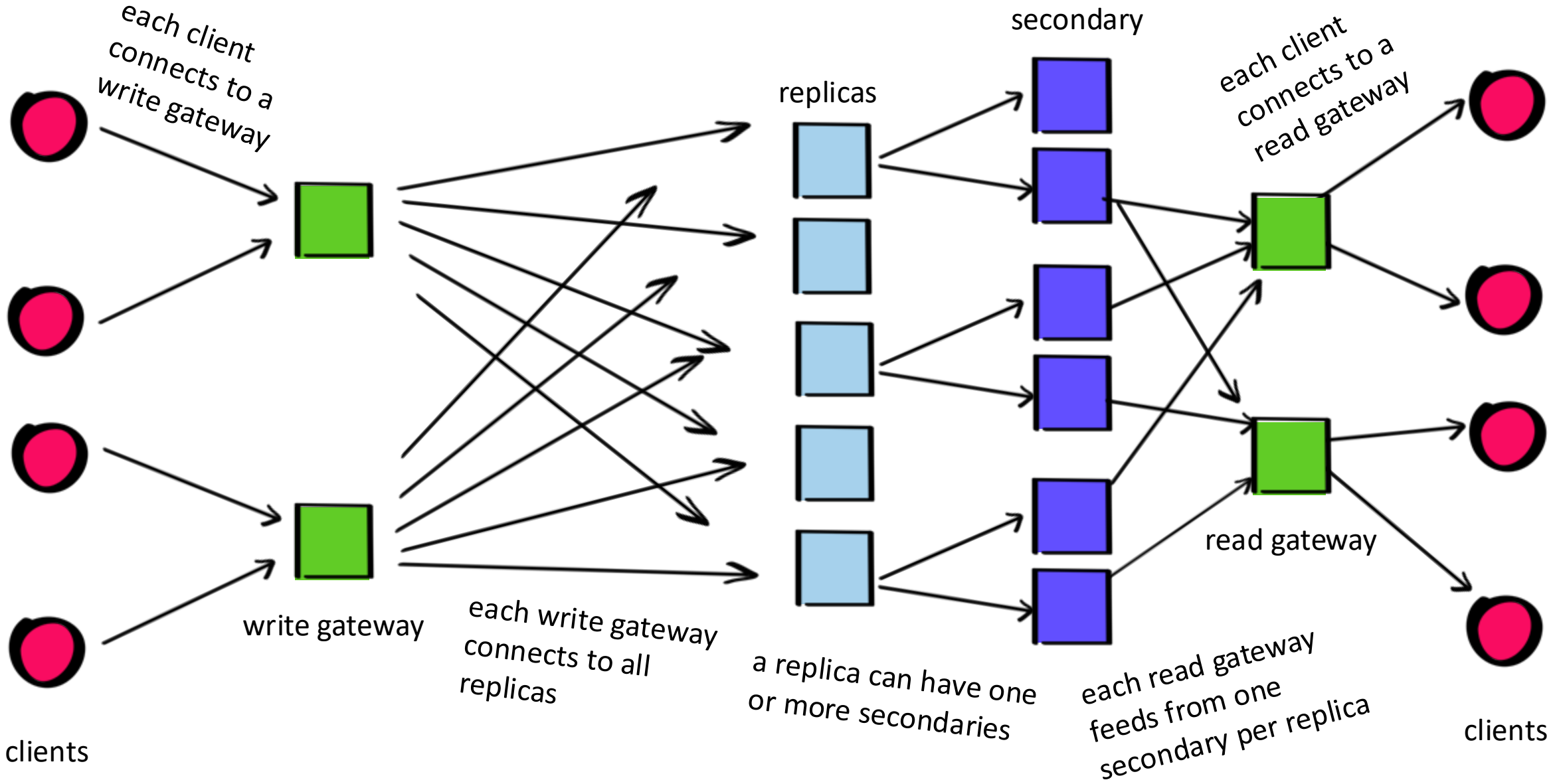
Gateways



Read-only Secondaries



Complete Architecture



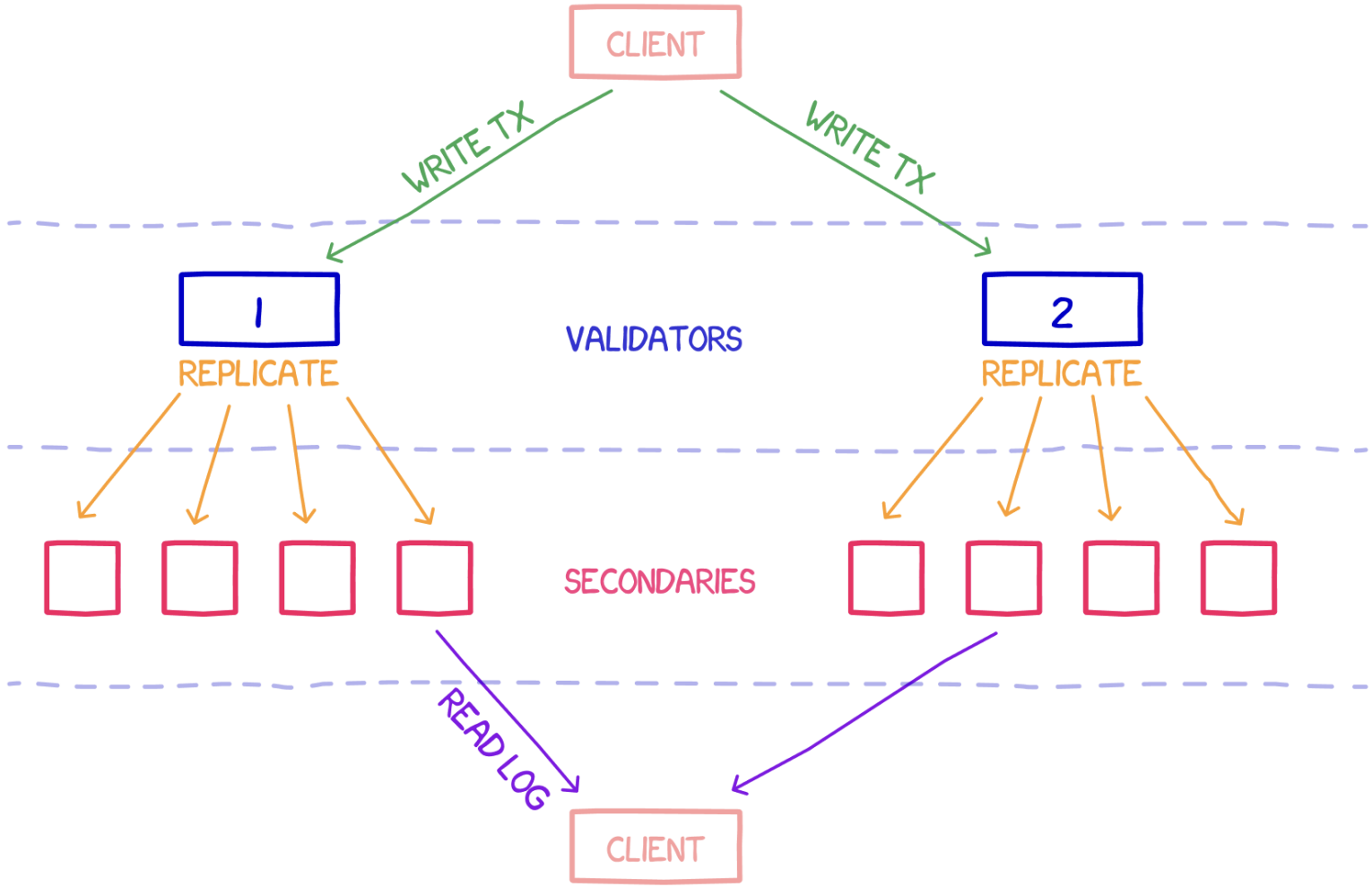
Key Takeaways

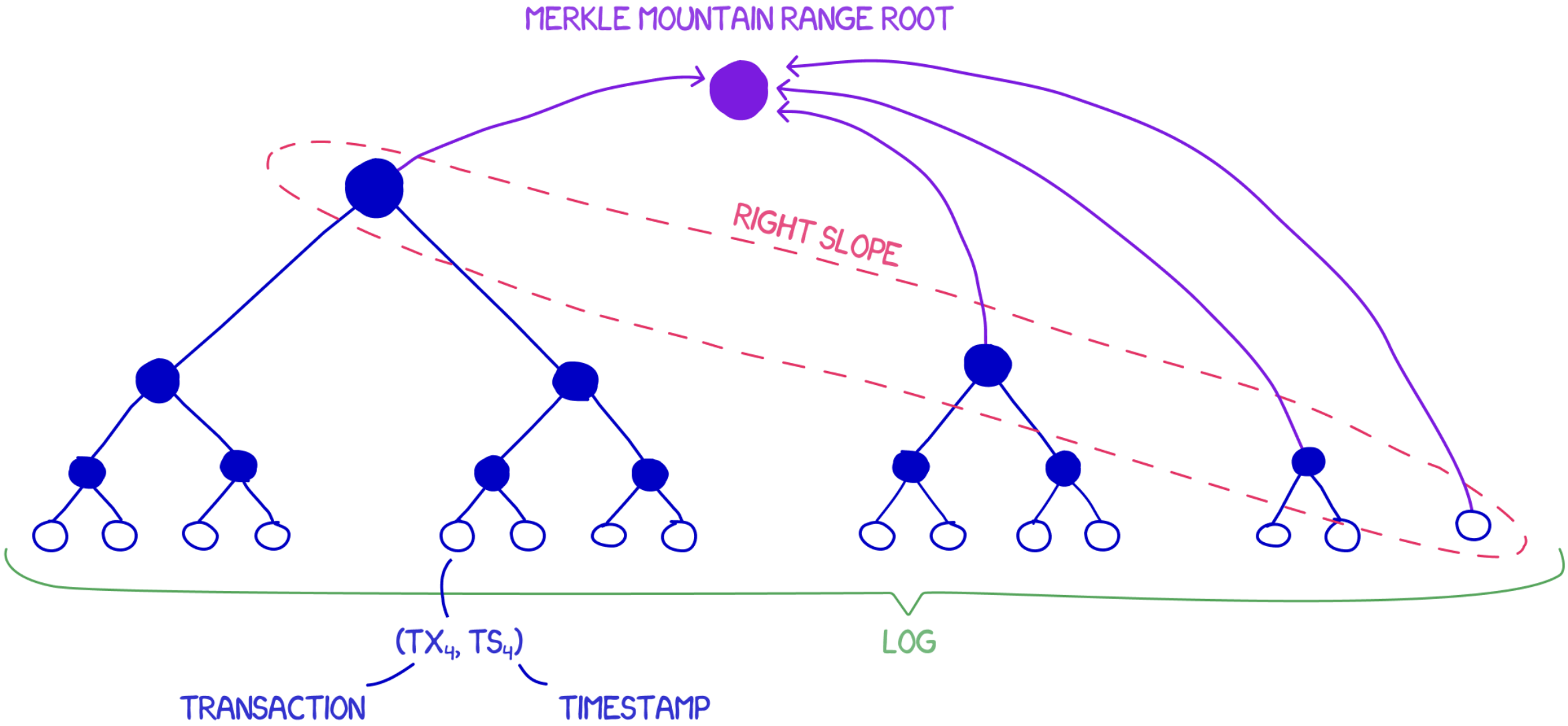
- pod: new protocol achieving **optimal latency**
- Simple design inspired by **traditional databases**
- Applications: Payments, auctions, voting, decentralized social, notarization… but no general smart contracts

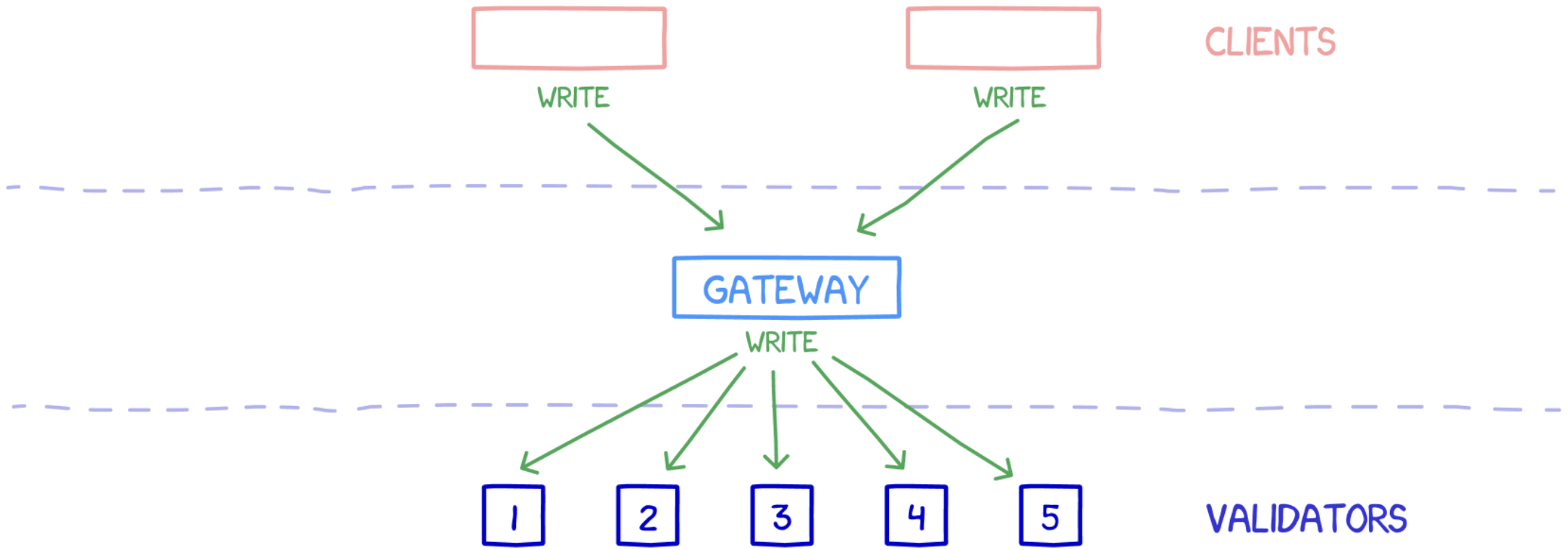
Come help us build the protocols of tomorrow.
We're hiring scientists & software engineers.

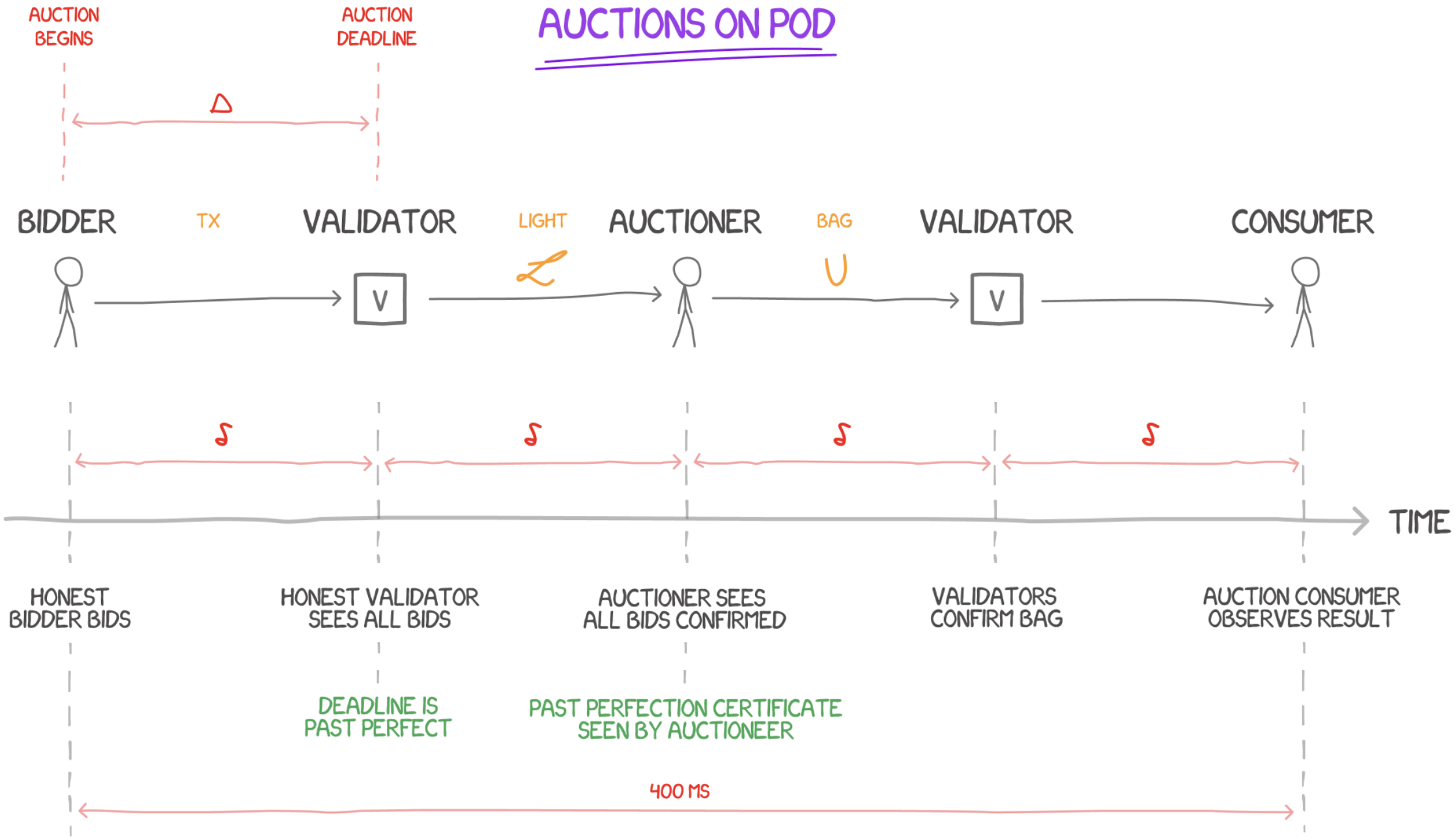
commonprefix.com/careers

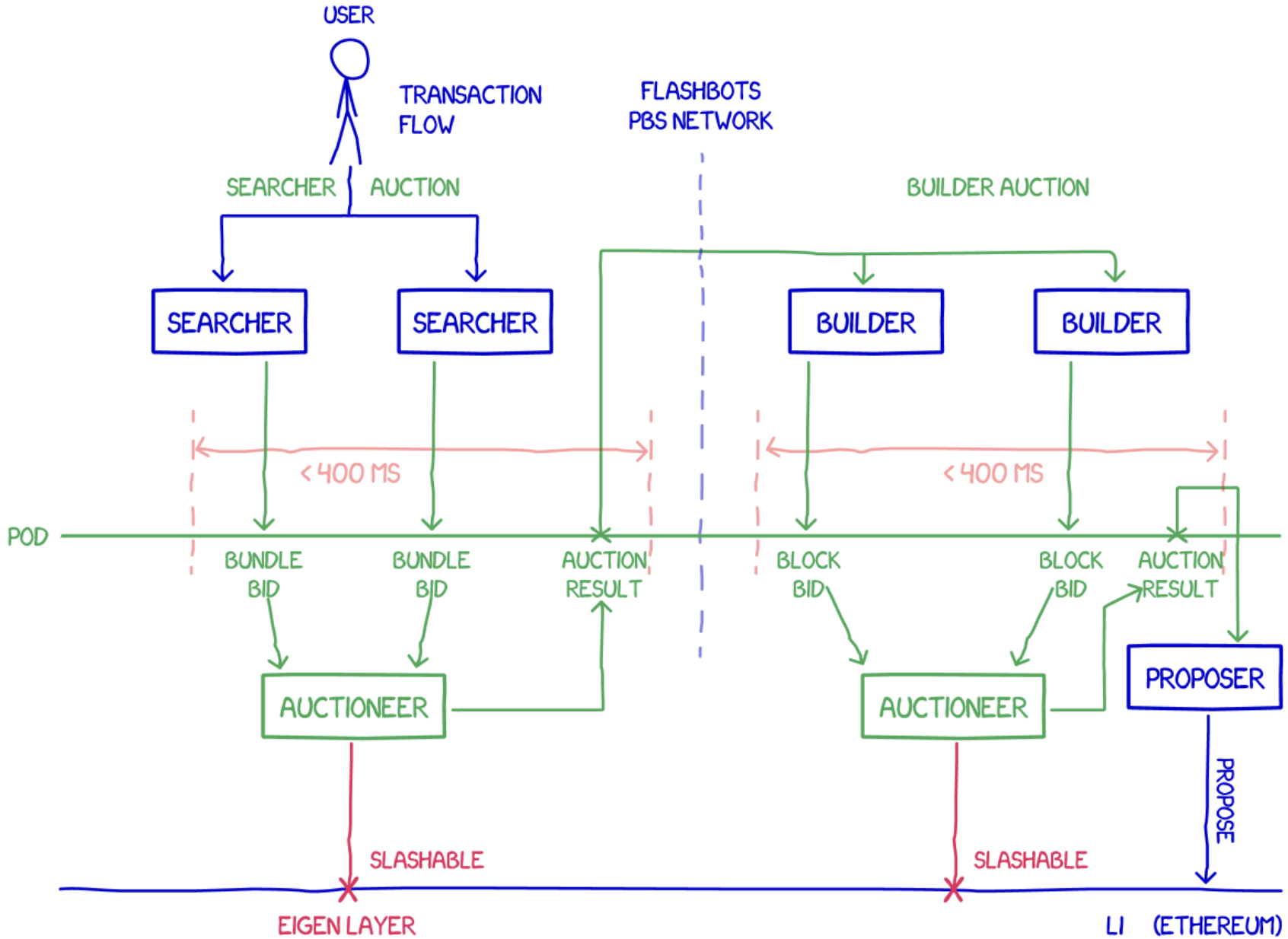
scan for paper


CLIENT
WRITE TX
WRITE TX
1
2
VALIDATORS
REPLICATE
REPLICATE
SECONDARIES
READ LOG
CLIENT


MERKLE MOUNTAIN RANGE ROOT
RIGHT SLOPE
(TX4, TS4)
TRANSACTION
TIMESTAMP
LOG


CLIENTS
WRITE
WRITE
GATEWAY
WRITE
1
2
3
4
5
VALIDATORS


AUCTION BEGINS
AUCTION DEADLINE
Δ
AUCTIONS ON POD
BIDDER
TX
VALIDATOR
LIGHT
AUCTIONER
BAG
VALIDATOR
CONSUMER
V
V
δ
δ
δ
δ
TIME
HONEST BIDDER BIDS
HONEST VALIDATOR SEES ALL BIDS
AUCTIONER SEES ALL BIDS CONFIRMED
VALIDATORS CONFIRM BAG
AUCTION CONSUMER OBSERVES RESULT
DEADLINE IS PAST PERFECT
PAST PERFECTION CERTIFICATE SEEN BY AUCTIONEER
400 MS



FLASHBOTS AUCTION ON POD