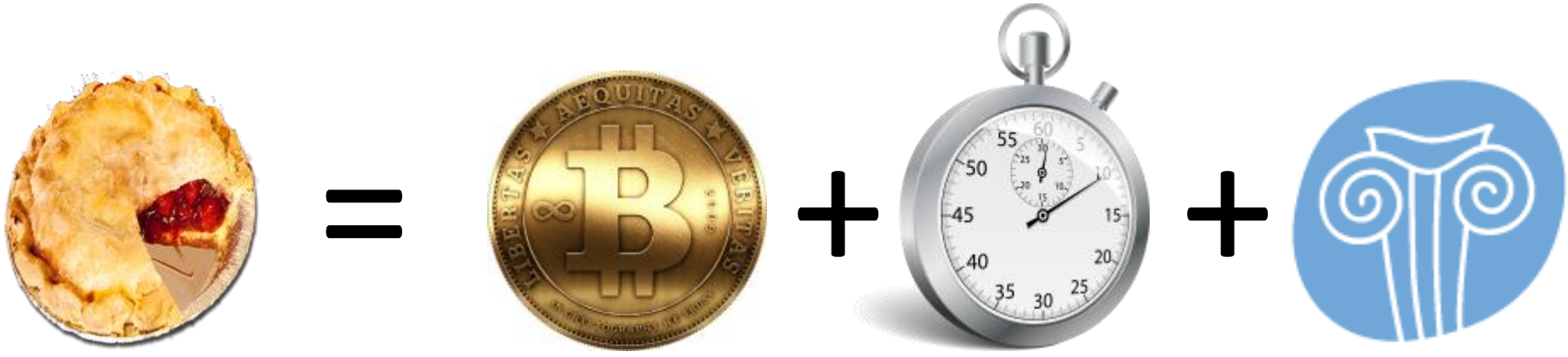
ETH Zurich – Distributed Computing Group

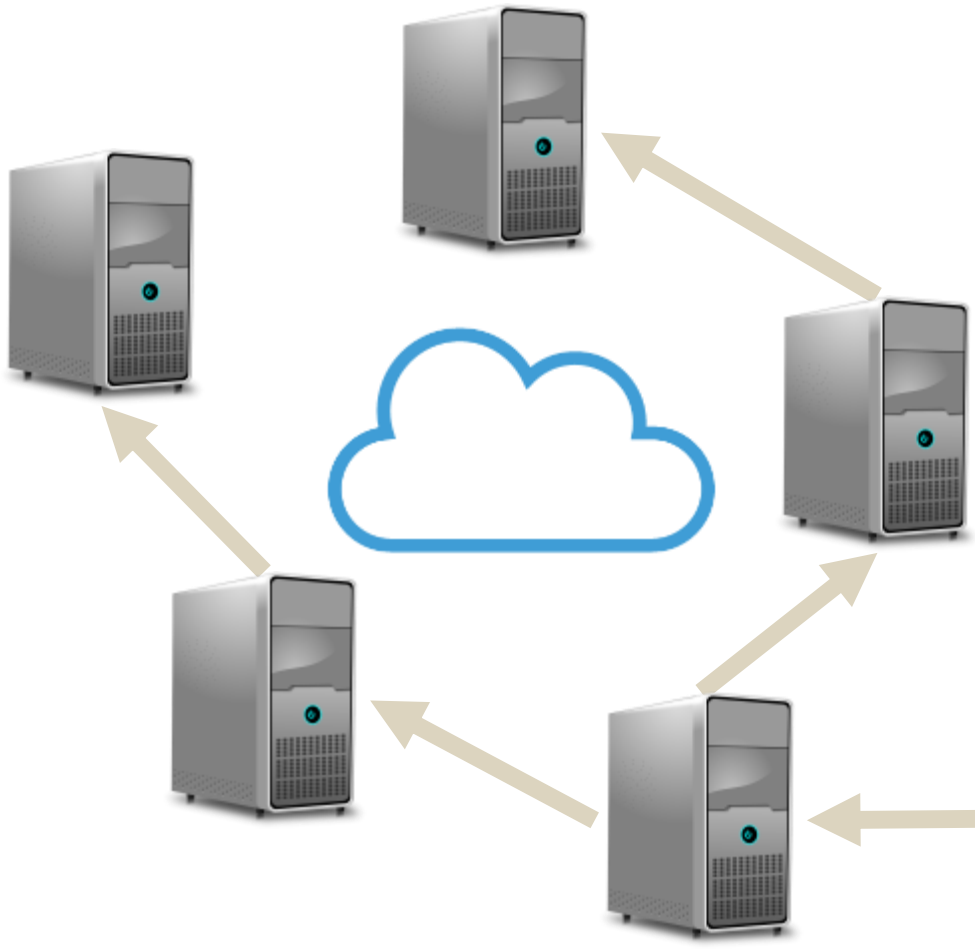
piChain: When a Blockchain Meets Paxos

=

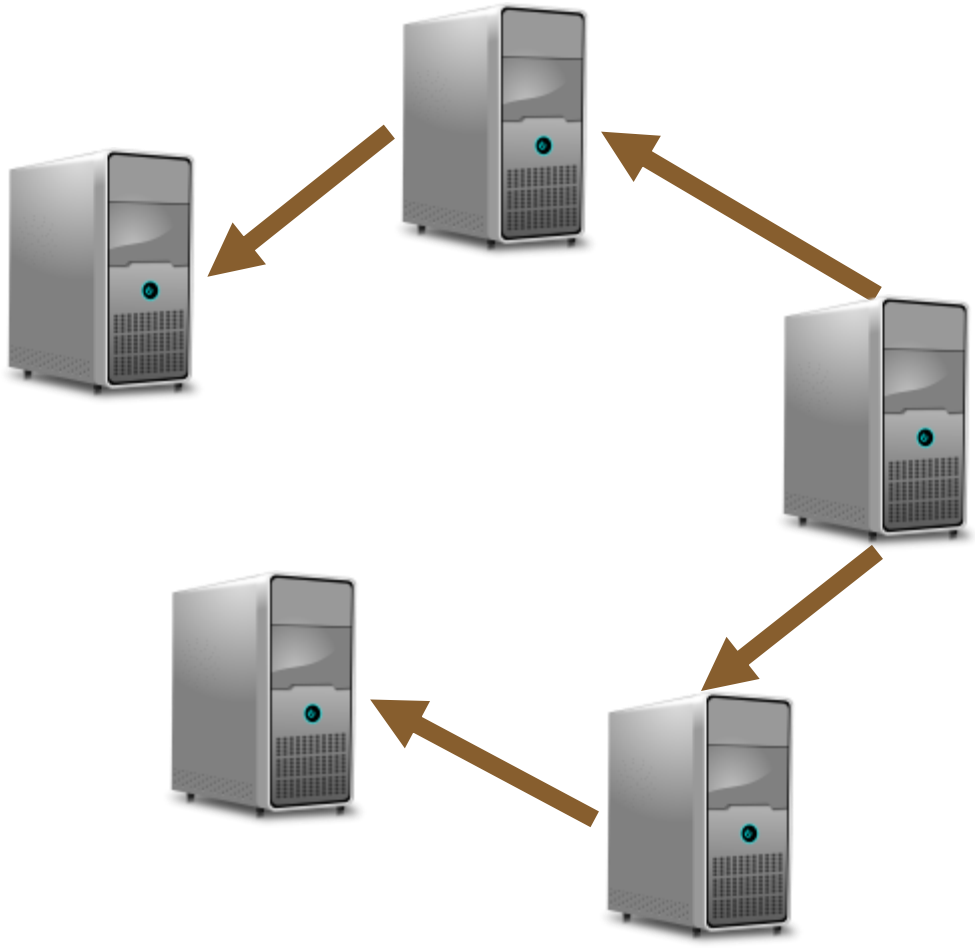
+

piChain: When a Blockchain Meets Paxos

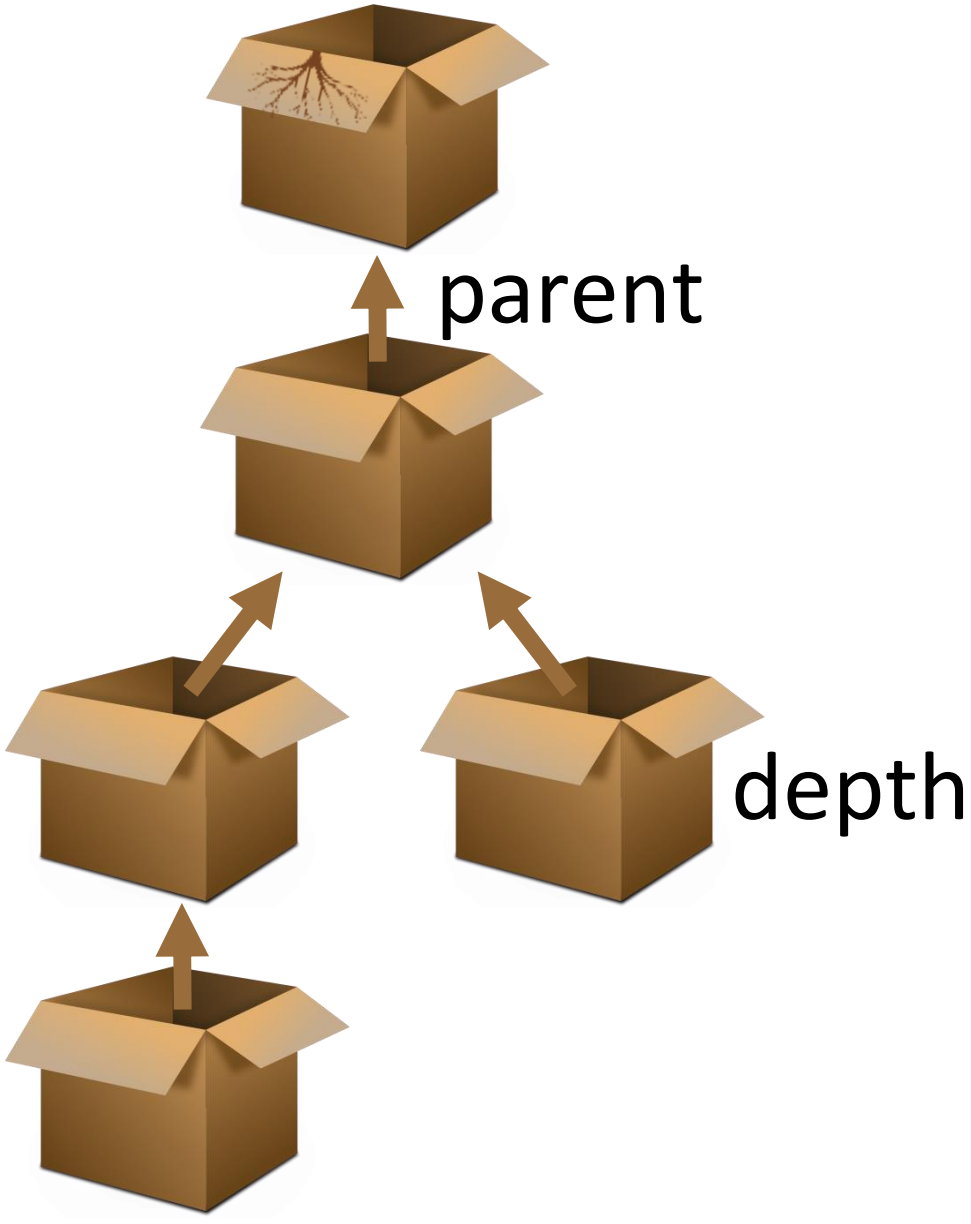




Transaction



Block


parent
depth

slow

medium

quick

New Transaction: Reaction Time

wait

wait

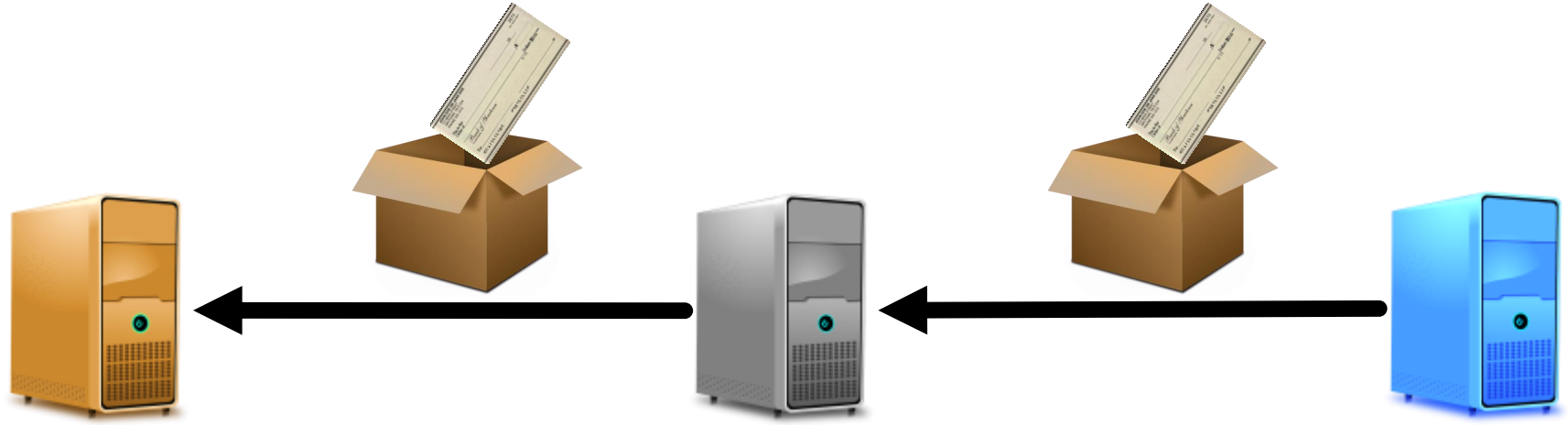
quick

medium

slow

time

State Transitions

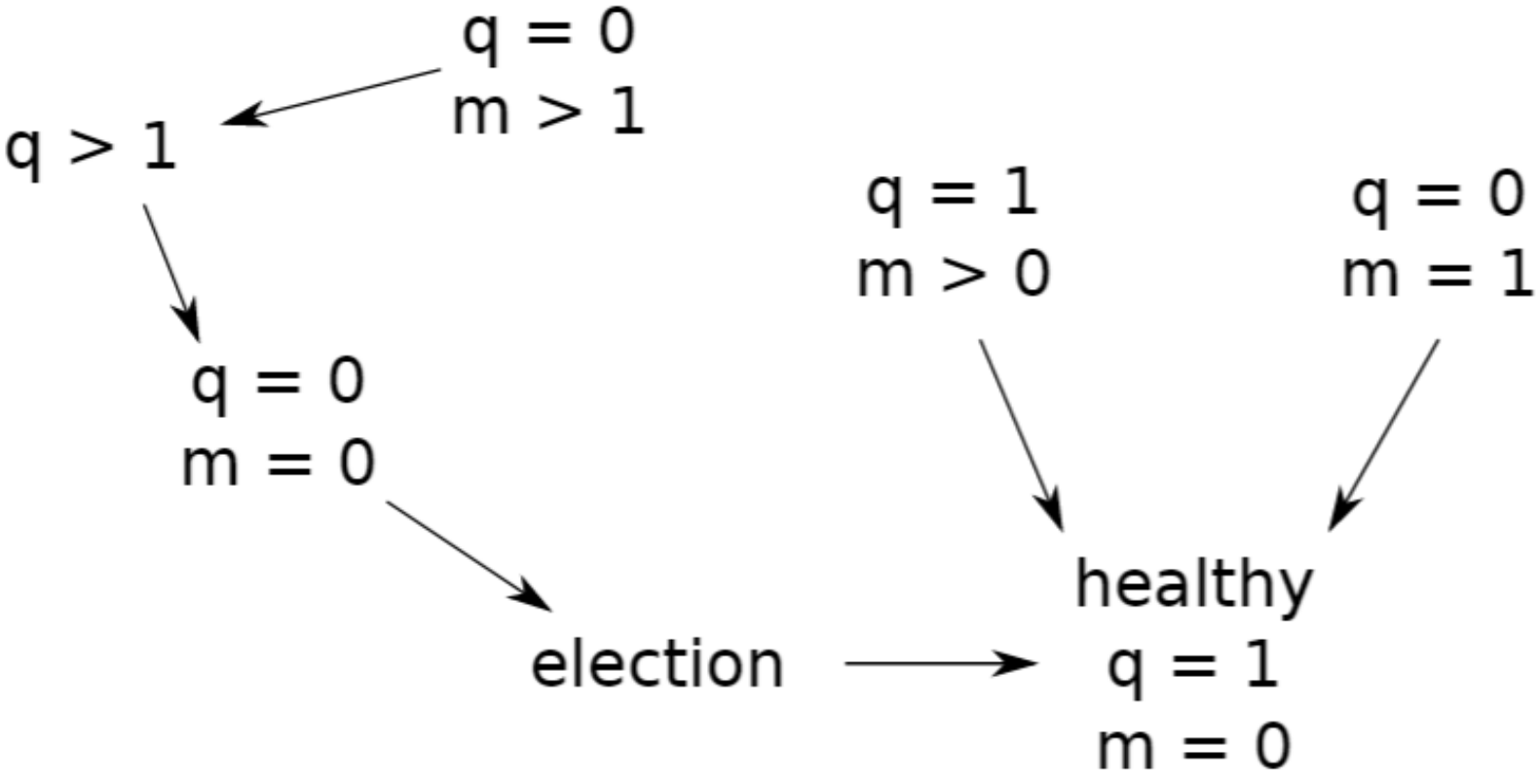


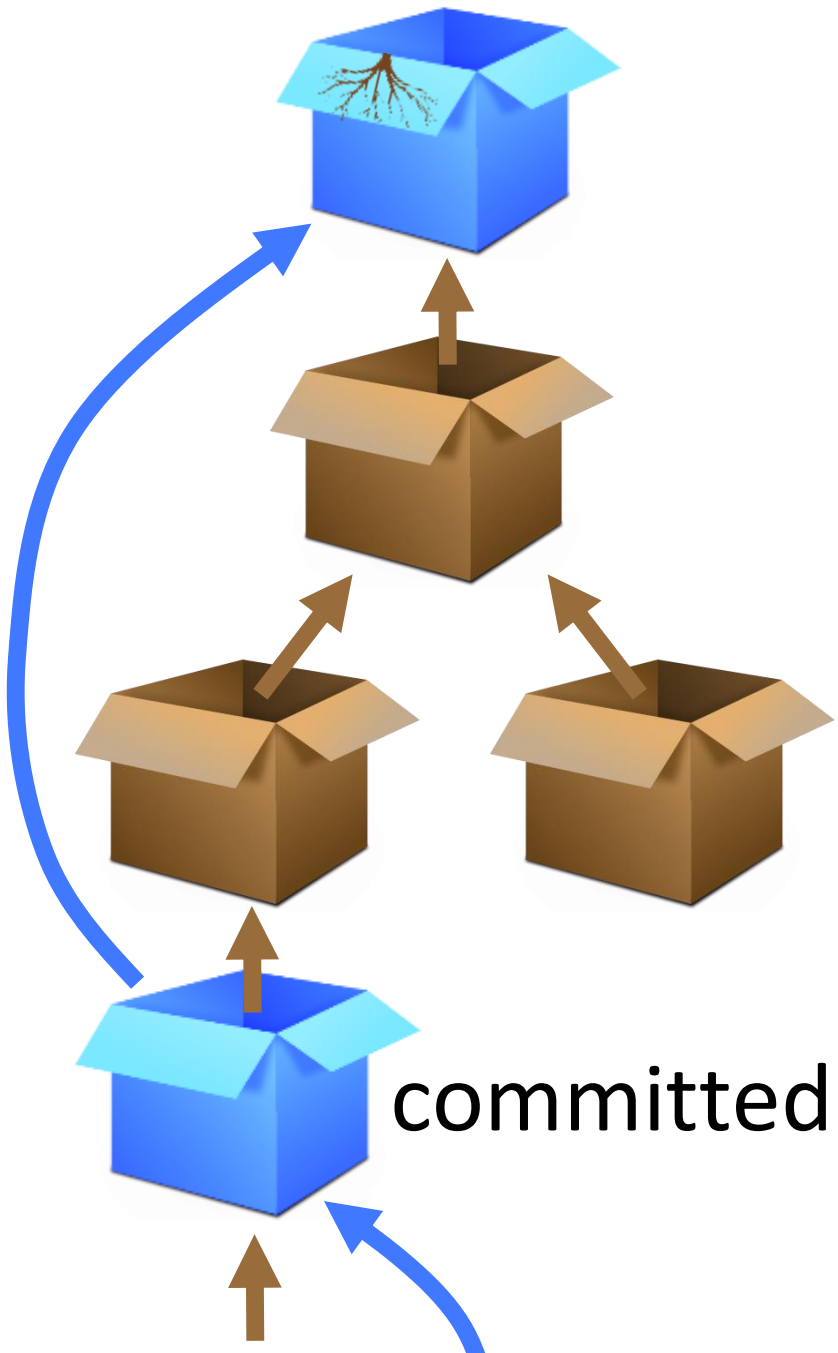
seen  : either deeper or by 

Self-Healing

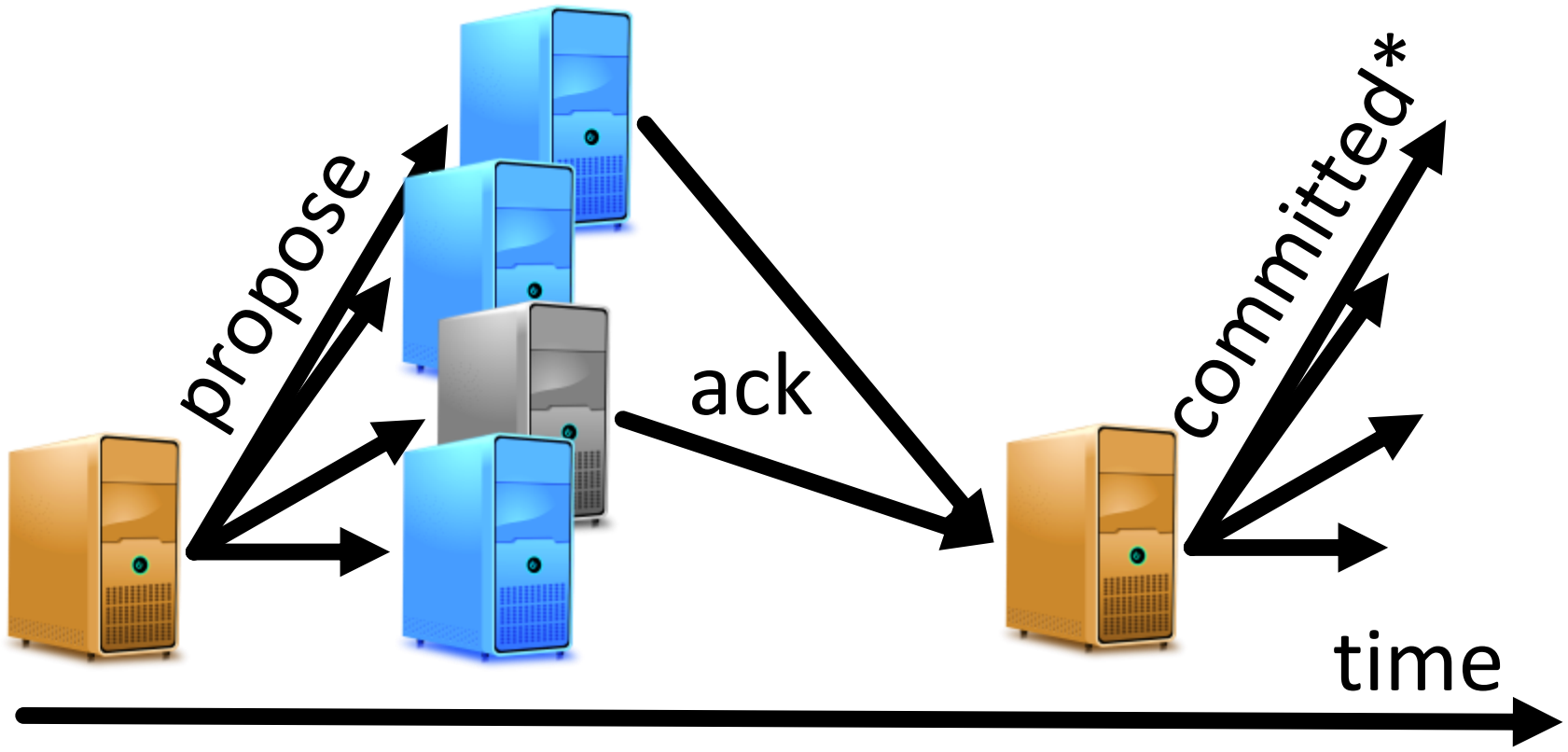
healthy

Self-Healing




committed

Truncated Paxos



*and next propose

Normal Paxos

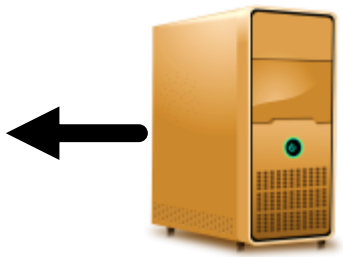
Round 1 .....

1: Quick node q sends "try b_{new}" to all nodes

2: On receiving a try message, all nodes:
3: **if** b_{new} deeper than b_{max} **then**
4: $b_{\text{max}} = b_{\text{new}}$
5: Answer q with "ok b_{prop}, b_{supp}"
6: **end if**

Round 2 .....

7: Node q: If majority responded with ok:
8: $b_{\text{com}} = b_{\text{new}}$
9: **if** some response included $b_{\text{prop}} \neq \perp$ **then**
10: $b_{\text{com}} = b_{\text{prop}}$ with deepest b_{supp}
11: **end if**
12: Node q sends "propose b_{com}, b_{new}" to all nodes

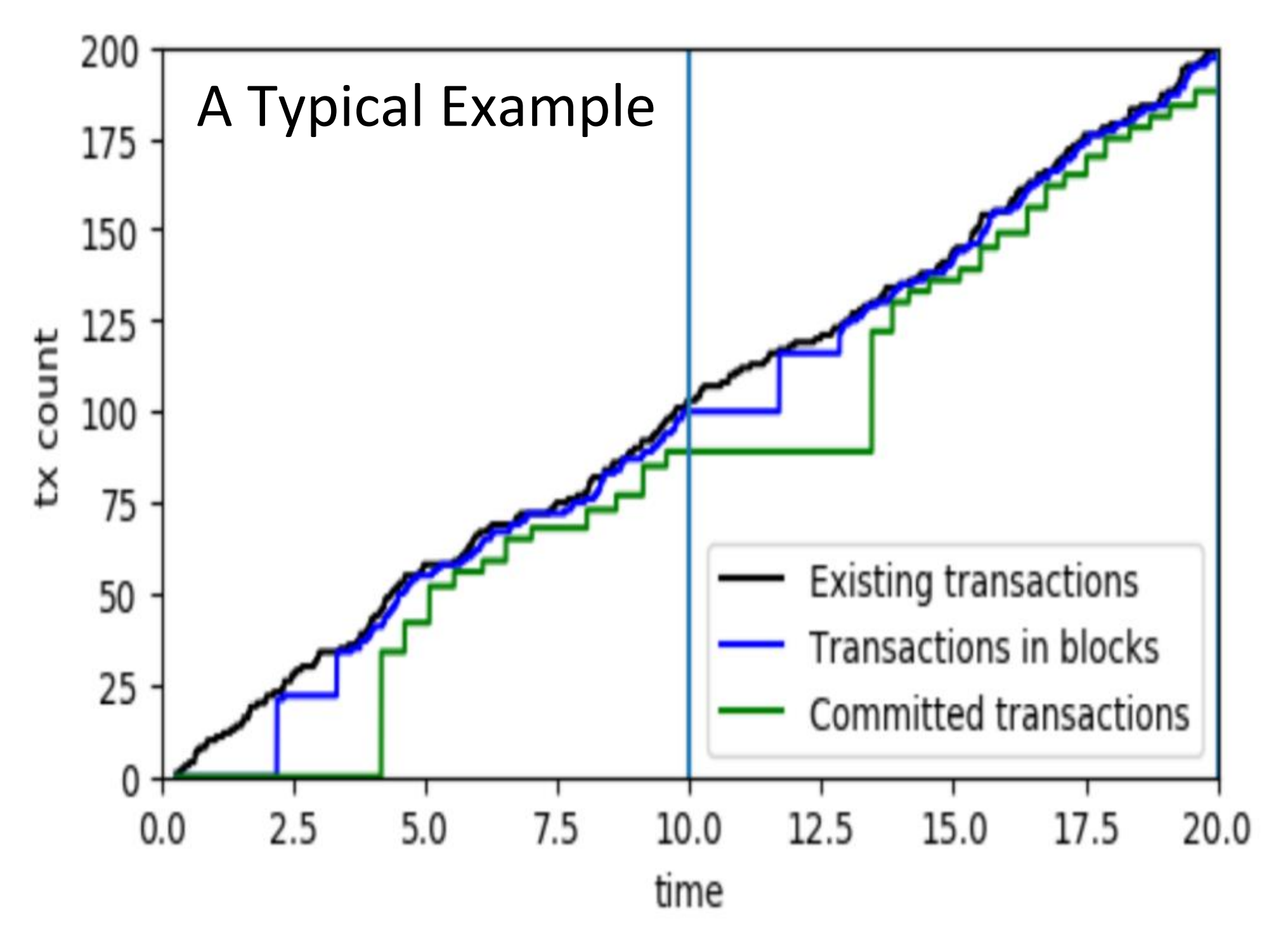


13: On receiving a propose message, all nodes:
14: **if** $b_{\text{new}} = b_{\text{max}}$ **then**
15: $b_{\text{prop}} = b_{\text{com}}$
16: $b_{\text{supp}} = b_{\text{new}}$
17: Answer q with "ack b_{com}"
18: **end if**

Round 3 .....

19: Node q: If majority responded with ack:
20: Node q sends "commit b_{com}" to all nodes

21: On receiving a commit message, all nodes:
22: Store b_{com} in their list of committed blocks


A Typical Example
200
175
150
125
100
75
50
25
0
tx count
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
time
Existing transactions
Transactions in blocks
Committed transactions

piChain vs. Raft

similar essentially same goals

simple e.g., no explicit leader election

silent no msg when no tx, no heartbeat

scalable O(1) msgs per node per tx


Thank You!
Questions & Comments?
www.disco.ethz.ch