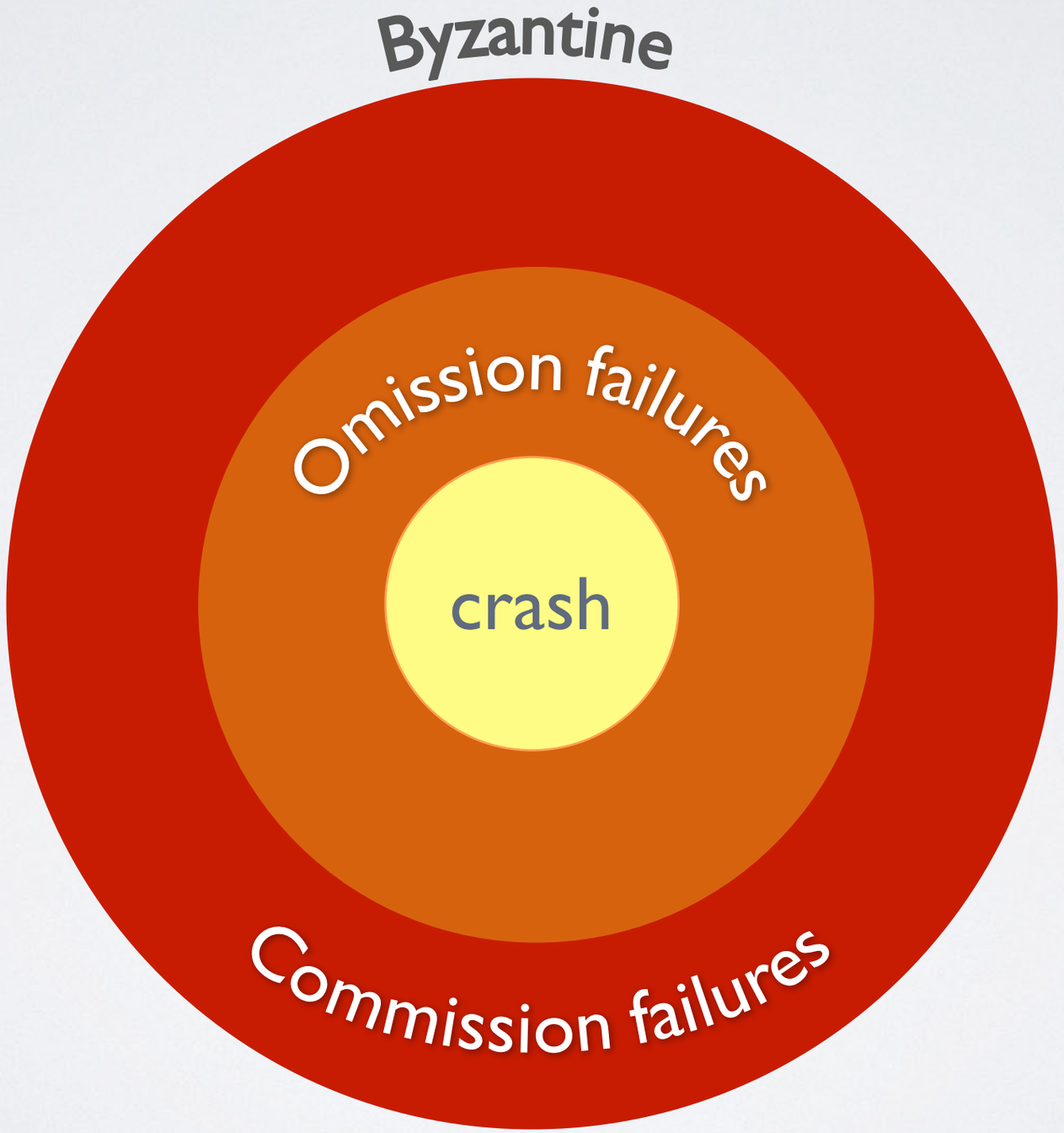
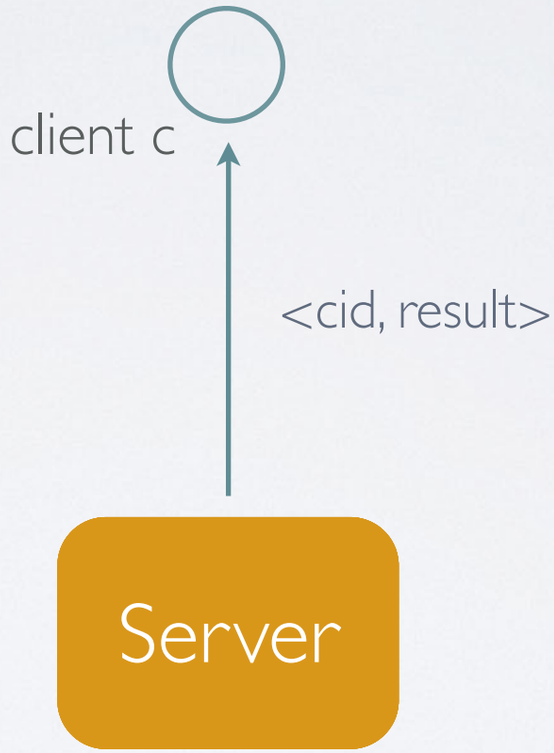
FAILURE MODELS



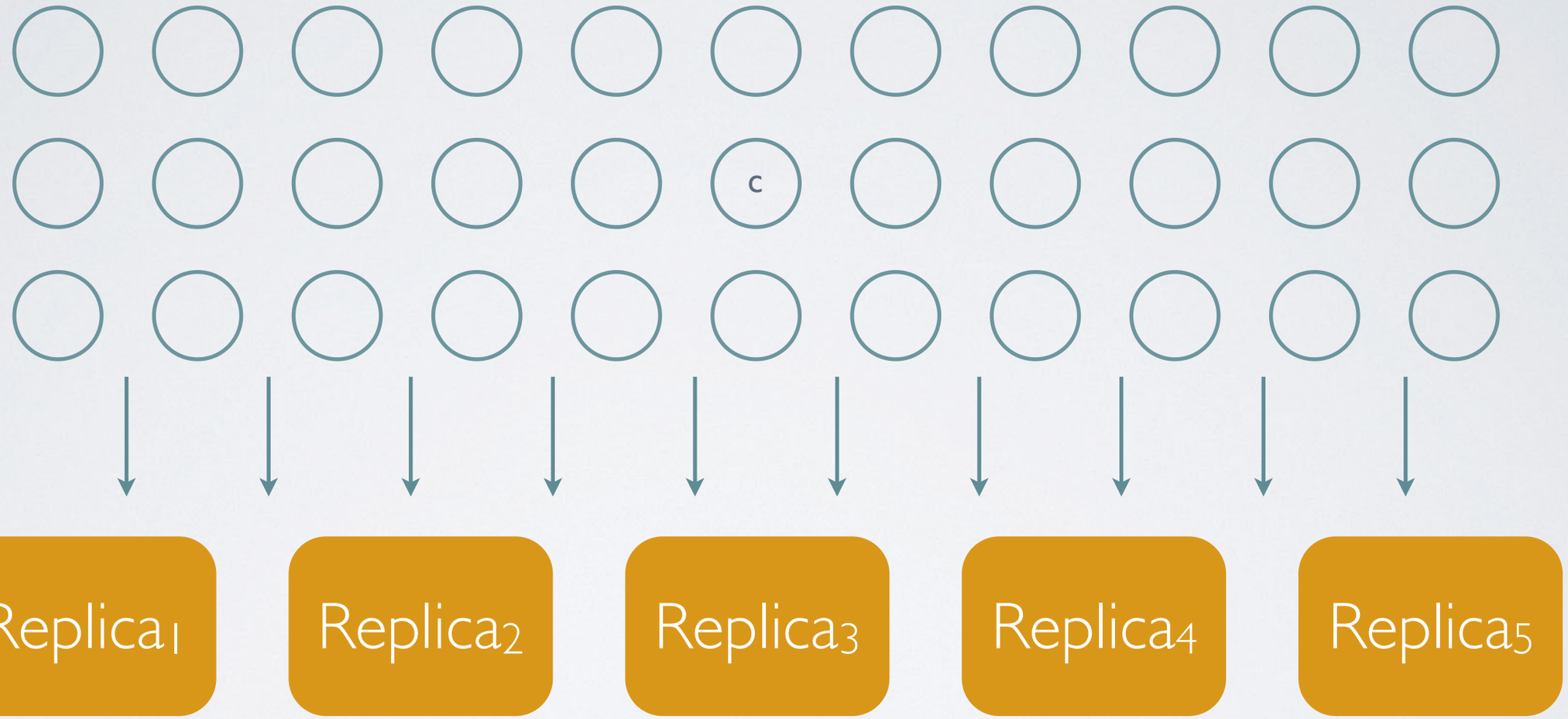
THE BIG PICTURE



THE BIG PICTURE

$f + 1$ Replica$_1$ Replica$_2$ Replica$_3$ Replica$_4$ Replica$_5$

THE BIG PICTURE

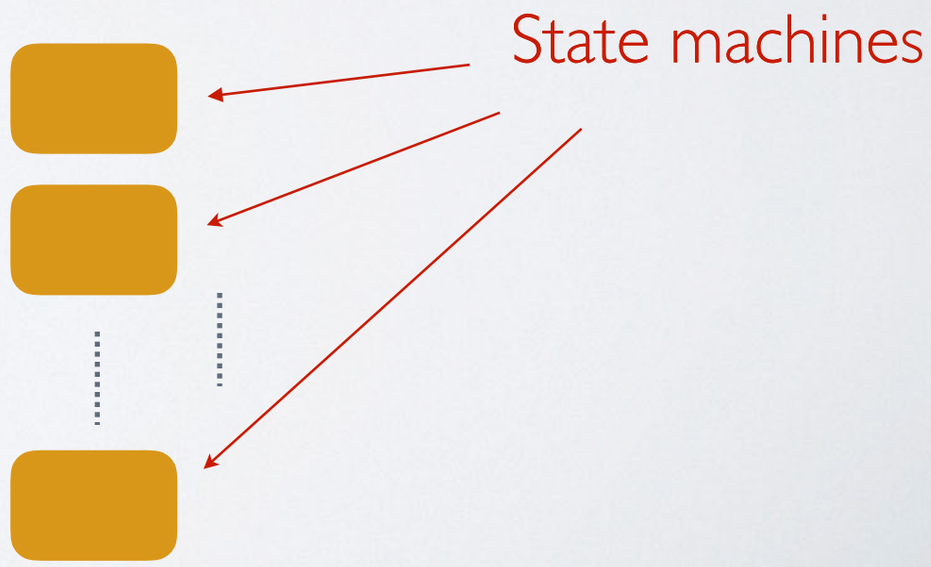


STATE MACHINE REPLICATION

1. Make server deterministic (state machine)

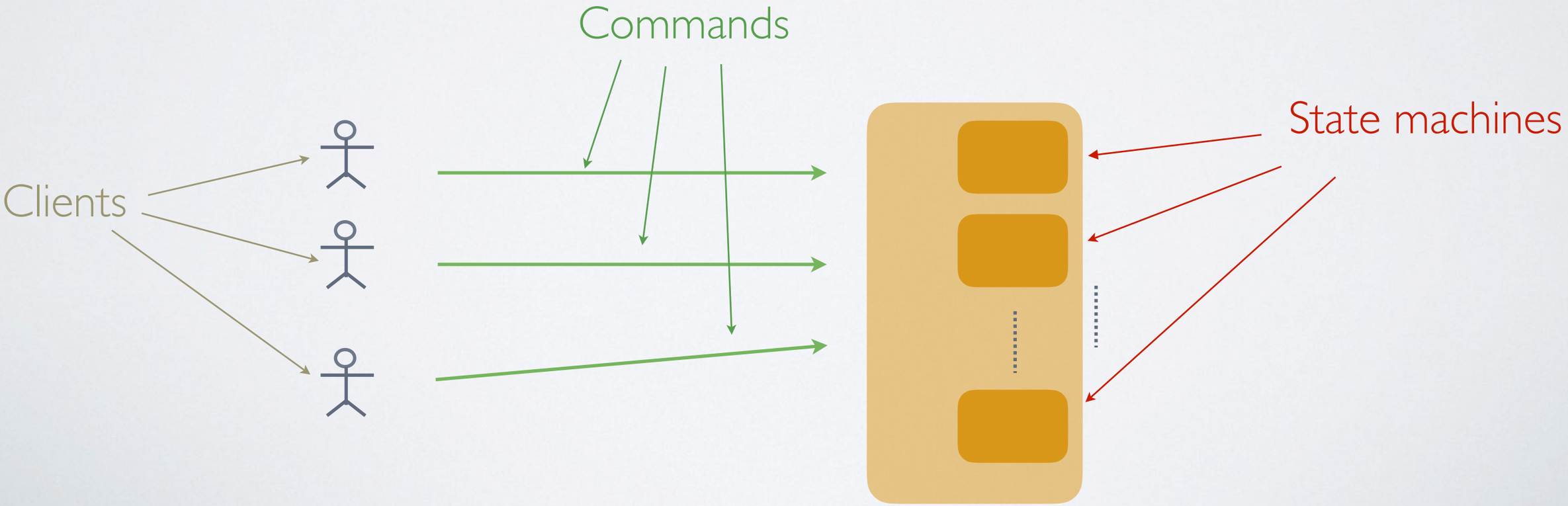
STATE MACHINE REPLICATION

1. Make server deterministic (state machine)
2. Replicate server



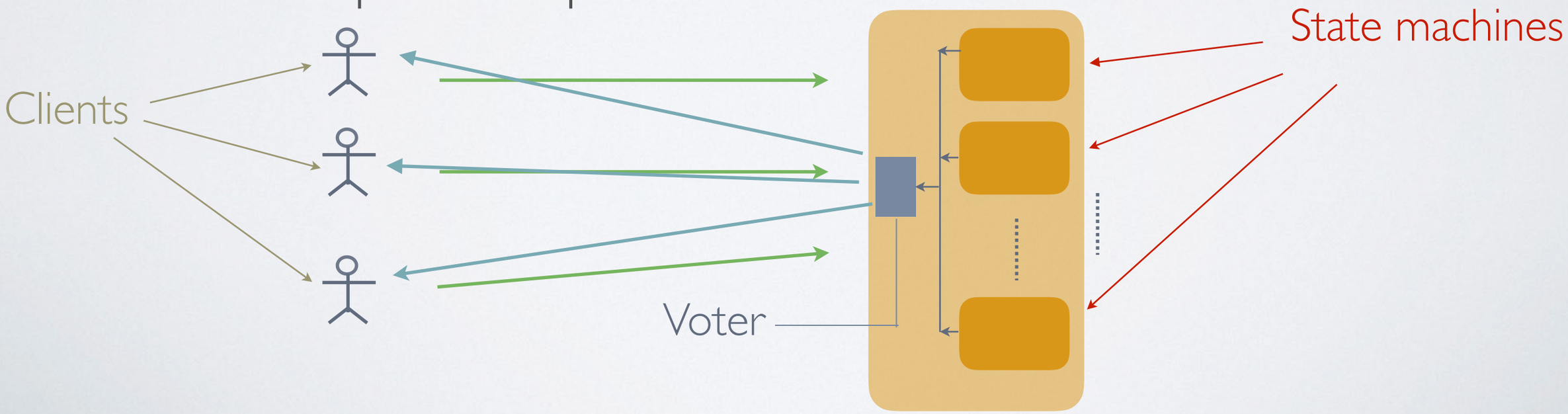
STATE MACHINE REPLICATION

1. Make server deterministic (state machine)
2. Replicate server
3. Ensure correct replicas step through the same sequence of state transitions

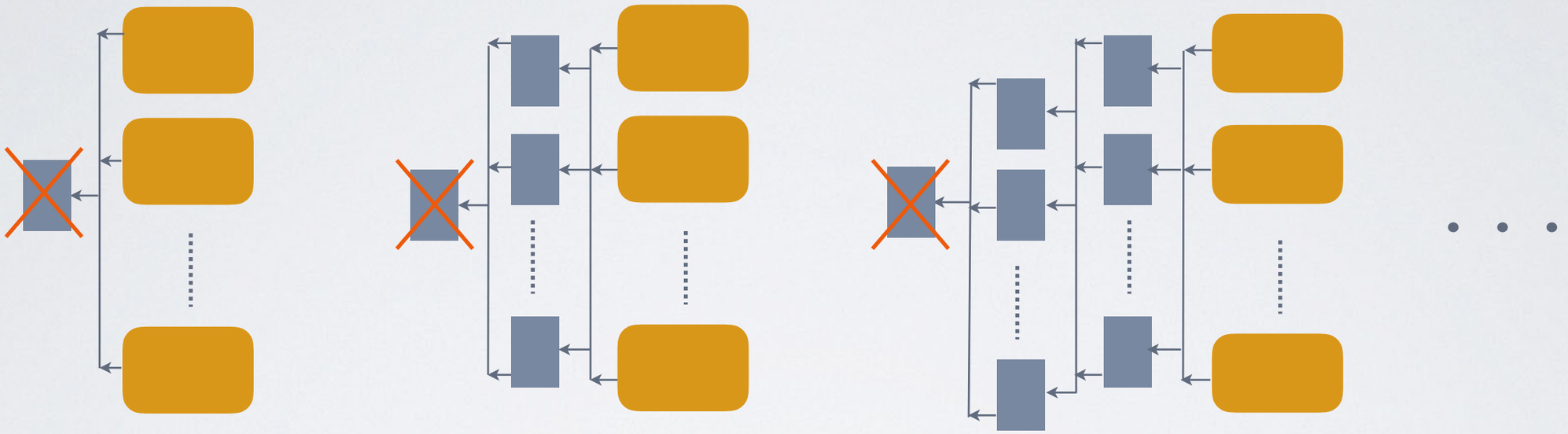


STATE MACHINE REPLICATION

1. Make server deterministic (state machine)
2. Replicate server
3. Ensure correct replicas step through the same sequence of state transitions
4. Vote on replica outputs for fault-tolerance

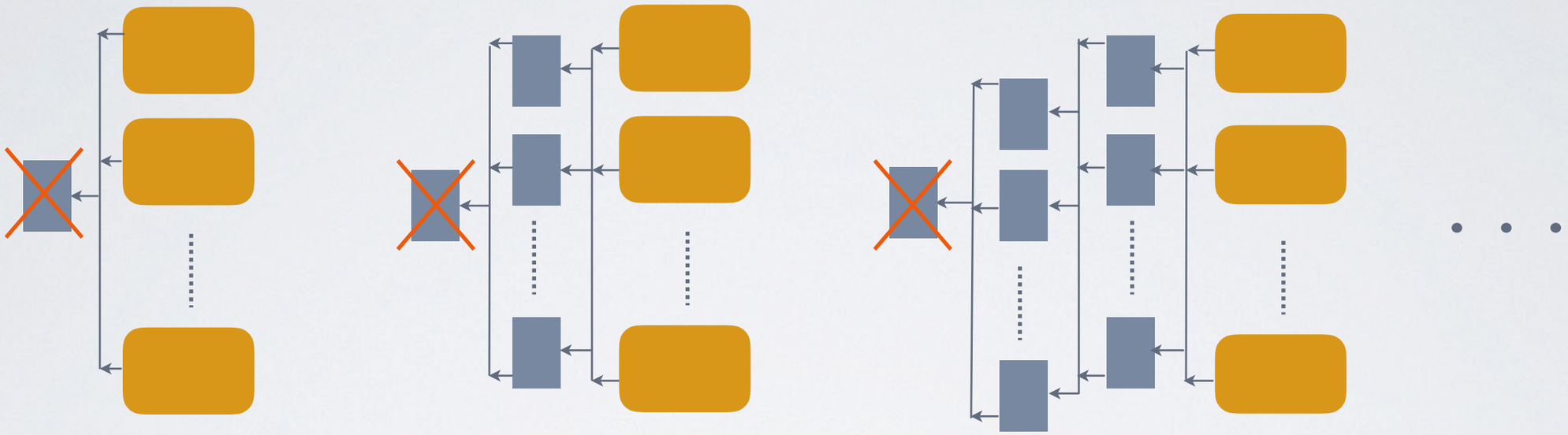


A CONUNDRUM

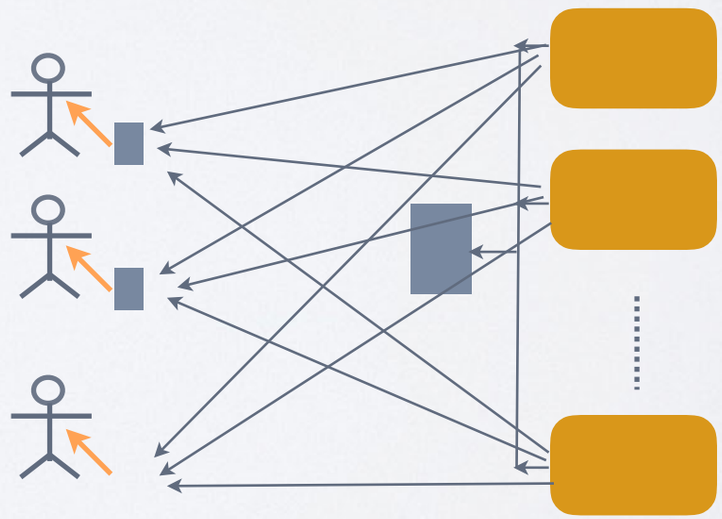


A: voter and
client share fate!

A CONUNDRUM



A: voter and
client share fate!



REPLICA COORDINATION

All non-faulty state machines receive all commands in the same order

- Agreement: Every non-faulty state machine receives every command
- Order: Every non-faulty state machine processes the commands it receives in the same order

HOW?

The Dear Leader

The Parliament

PRIMARY-BACKUP

- Clients communicate with the Dear Leader (the Primary)
- The Primary:
  - sequences clients' requests
  - updates as needed other replicas (backups) with sequence of client requests or state updates
  - waits for acks from all non-faulty clients
- Timeouts detect failure of primary
- On primary failure, a backup is elected as new primary

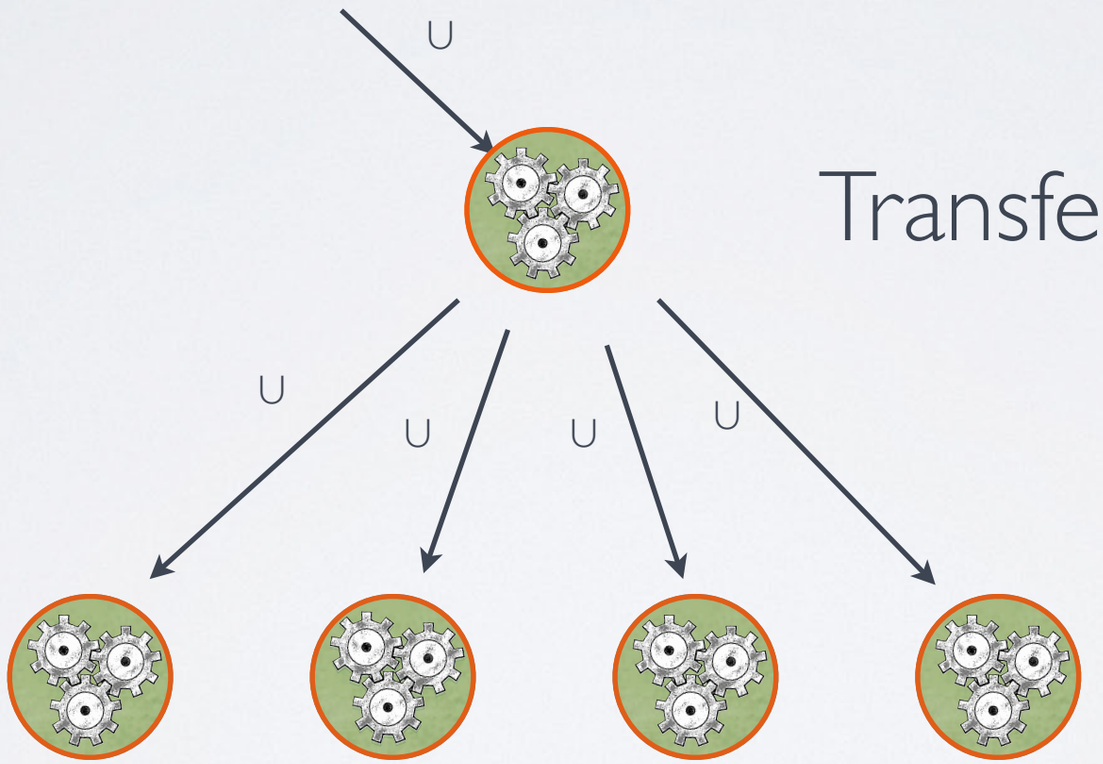
PRIMARY-BACKUP vs PARLIAMENT SYSTEM

1. More fault tolerance: $f < N$ vs $f < N/2$
2. Easier to develop, debug, tune, and maintain
3. Less communication and computation
4. Can handle non determinism (!)
5. Needs a stronger failure model (fail stop)

SOME LIKE IT HOT

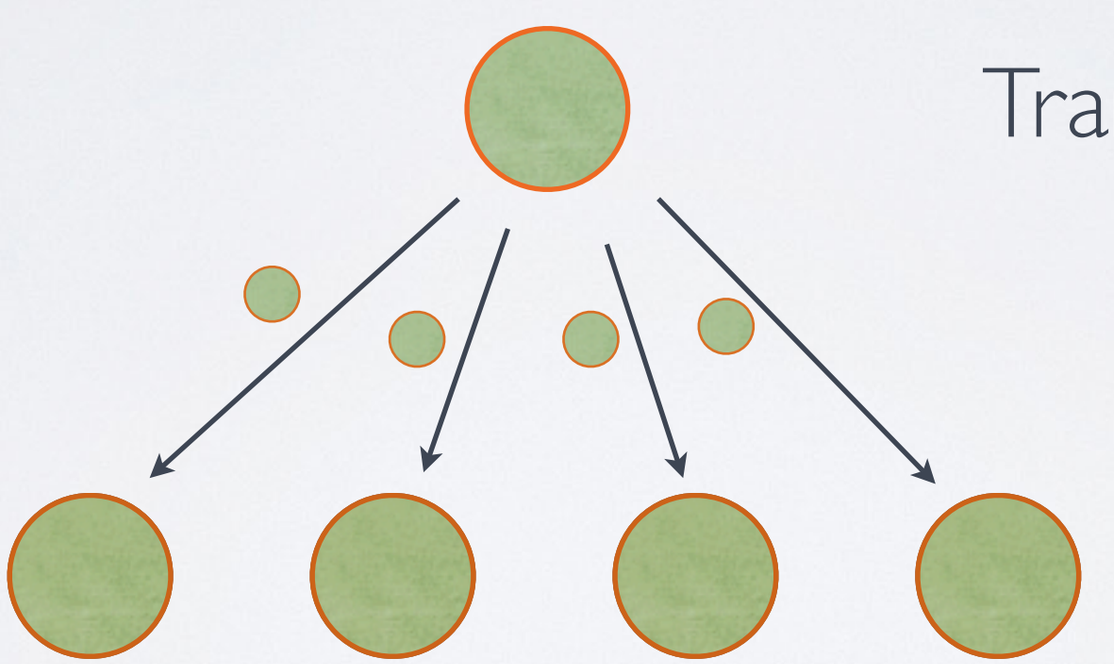
- Hot Backups process information from the primary as soon as they receive it
- Cold Backups log information received from primary, and process it only if primary fails
- Rollback Recovery implements cold backups cheaply:
  - Primary logs information needed by backups directly to stable storage
  - Backups are generated "on demand" upon primary's crash

PRIMARY-BACKUP: UPDATES



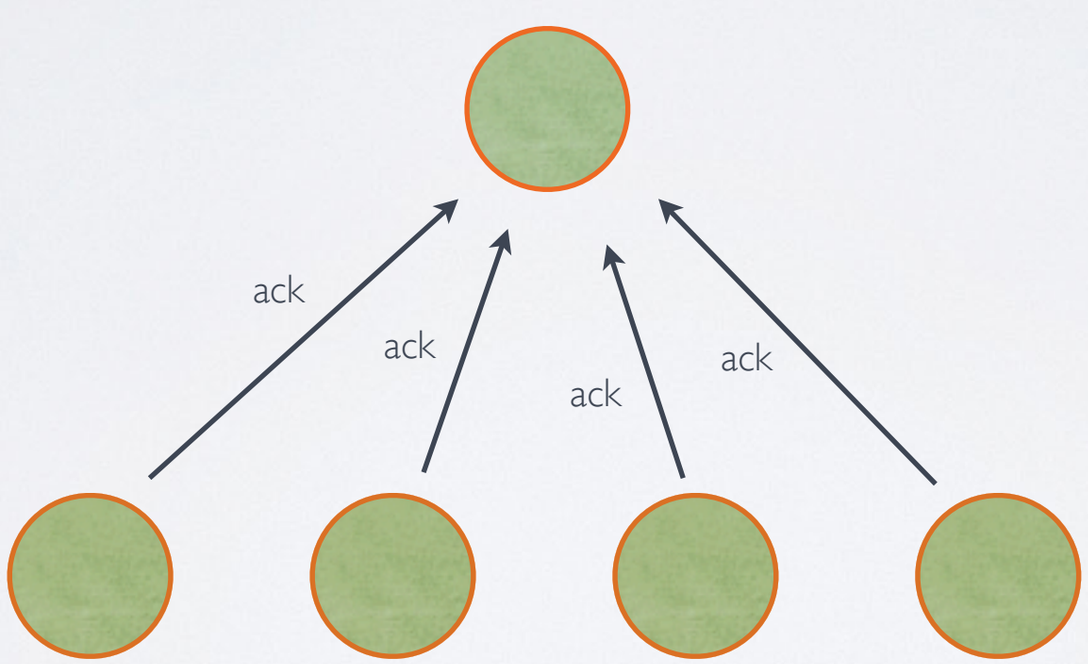
Transfer update

PRIMARY-BACKUP: UPDATES

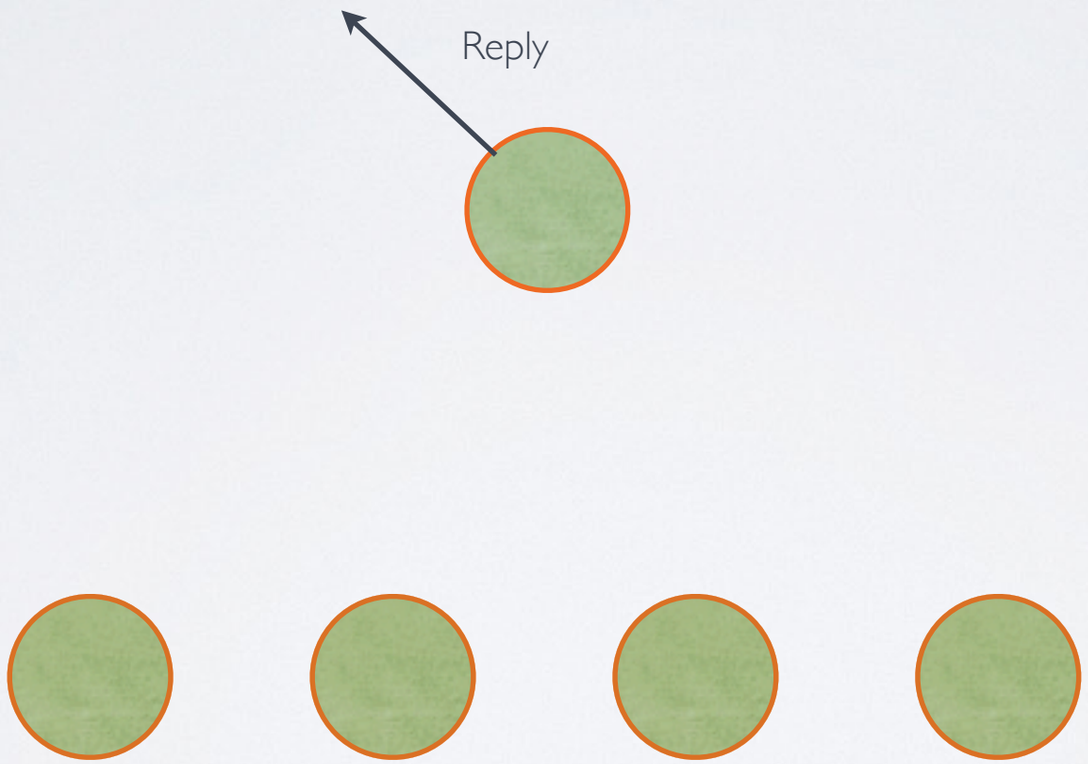


Transfer state

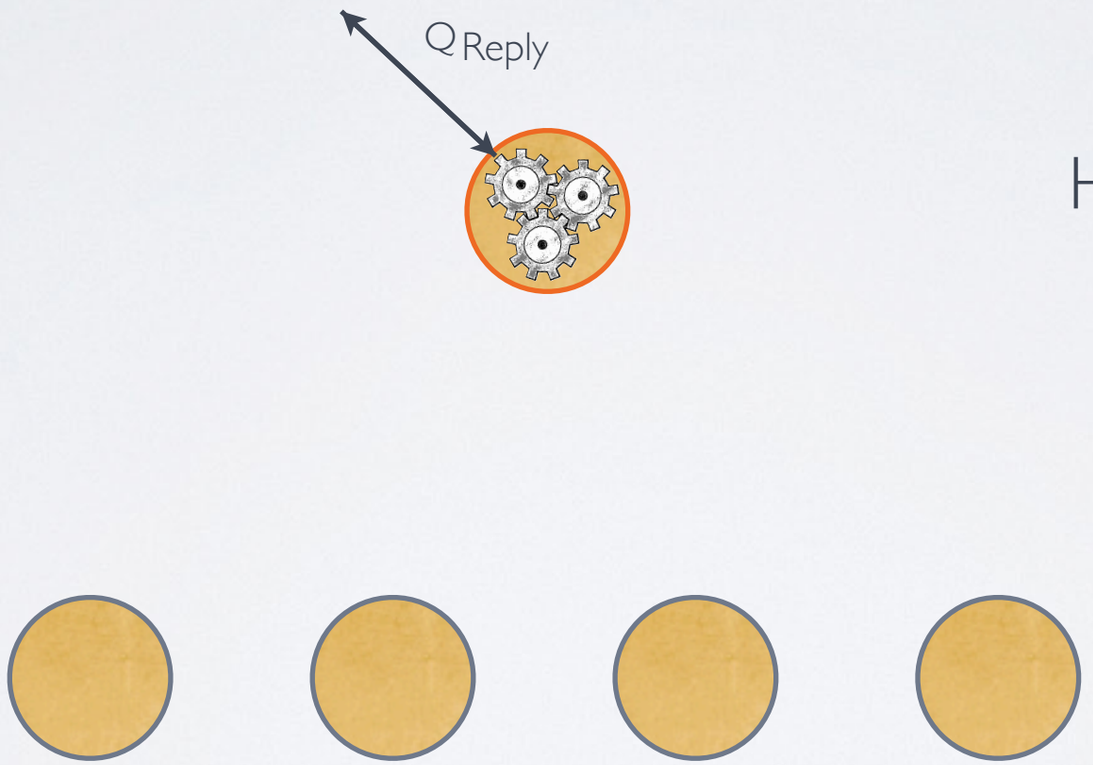
PRIMARY-BACKUP: UPDATES



PRIMARY-BACKUP: UPDATES

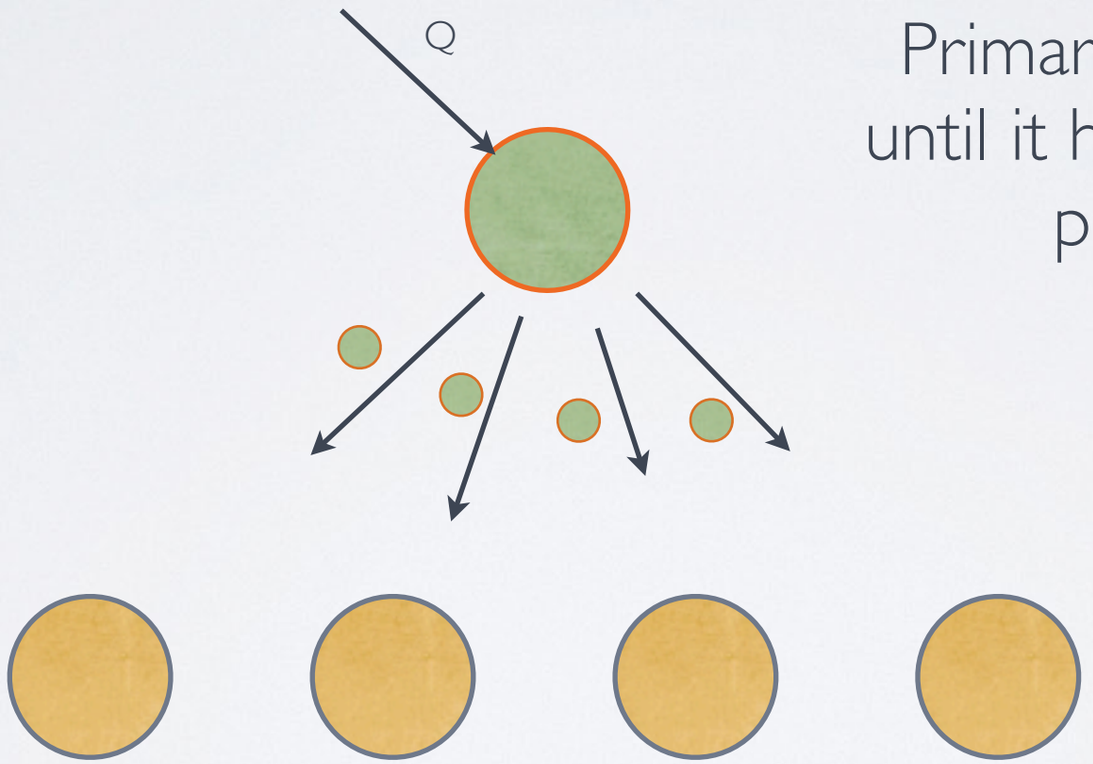


PRIMARY-BACKUP: QUERIES



However...

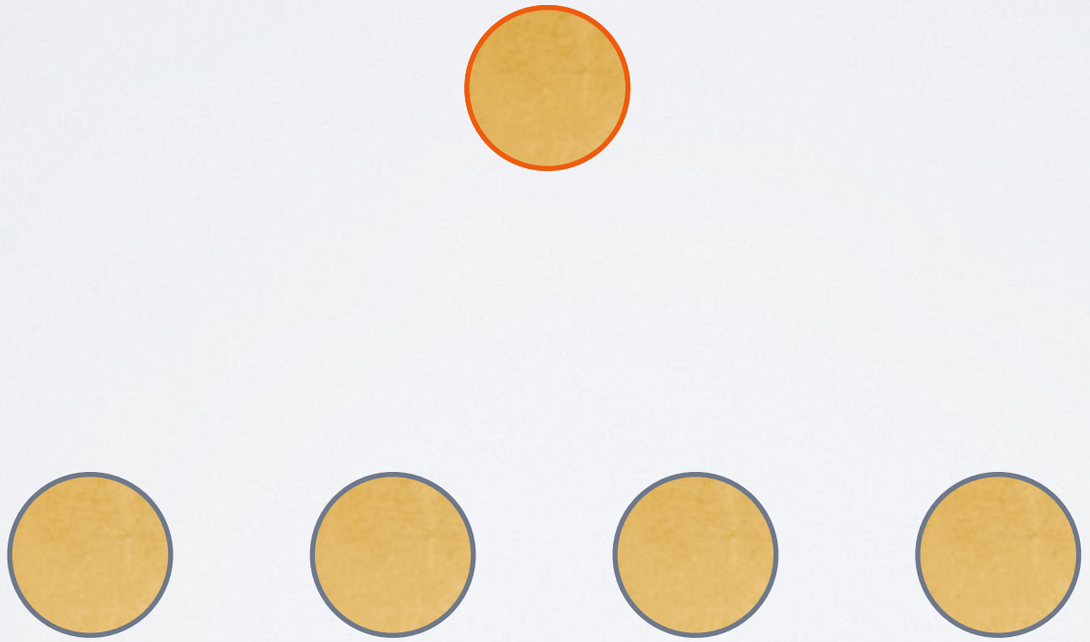
PRIMARY-BACKUP: QUERIES



Primary cannot respond until it has received ack for prior updates!

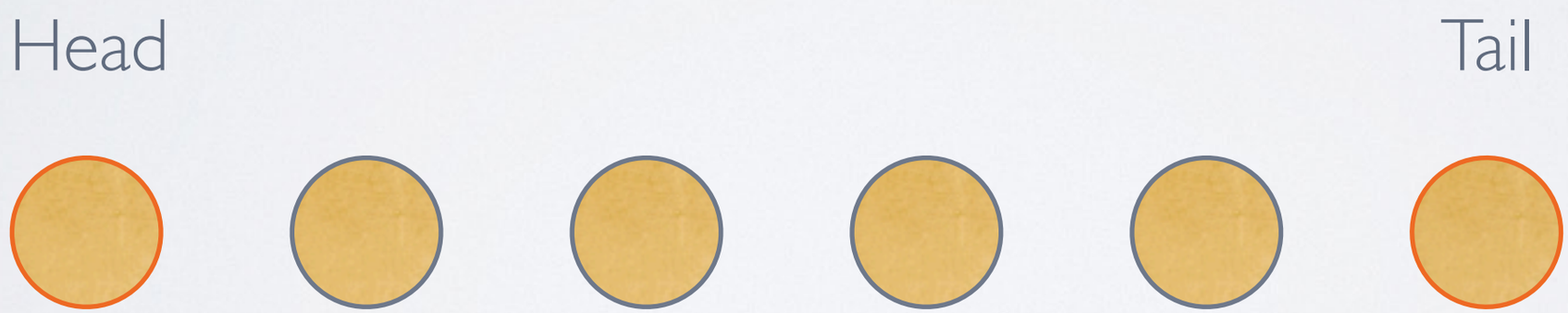
CHAIN REPLICATION

VAN RENESSE, SCHNEIDER, OSDI '04



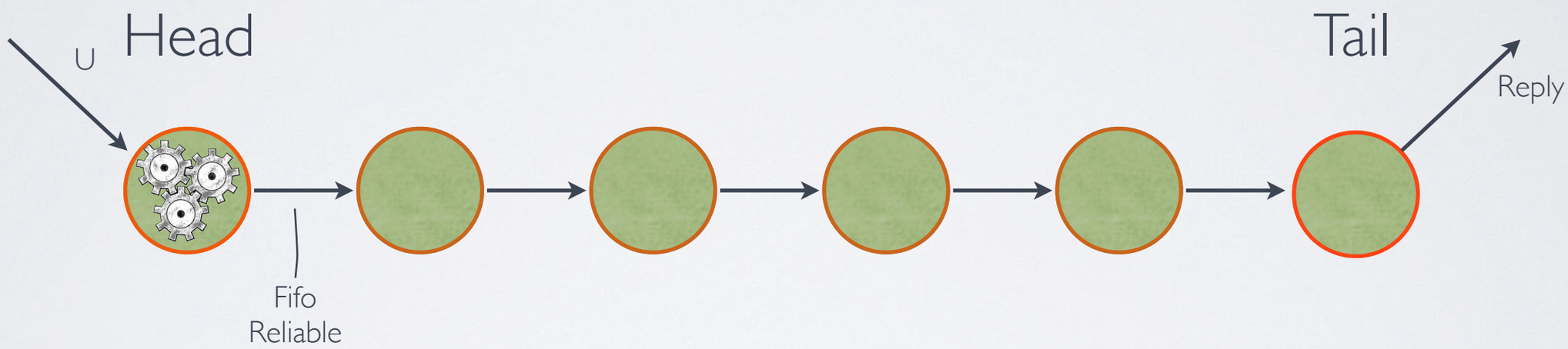
CHAIN REPLICATION

VAN RENESSE, SCHNEIDER, OSDI '04



CHAIN REPLICATION: UPDATES

VAN RENESSE, SCHNEIDER, OSDI '04



CHAIN REPLICATION: QUERIES

VAN RENESSE, SCHNEIDER, OSDI '04

Furthermore...

CHAIN REPLICATION: QUERIES

VAN RENESSE, SCHNEIDER, OSDI '04

Tail can respond immediately, without waiting for the new update