Permissionless	Permissioned	Attendance
Anyone can be a consensus node	Only authorized parties can be consensus nodes	
N → variable over time ↓ anyone	D D D D N parties ↓ fixed	

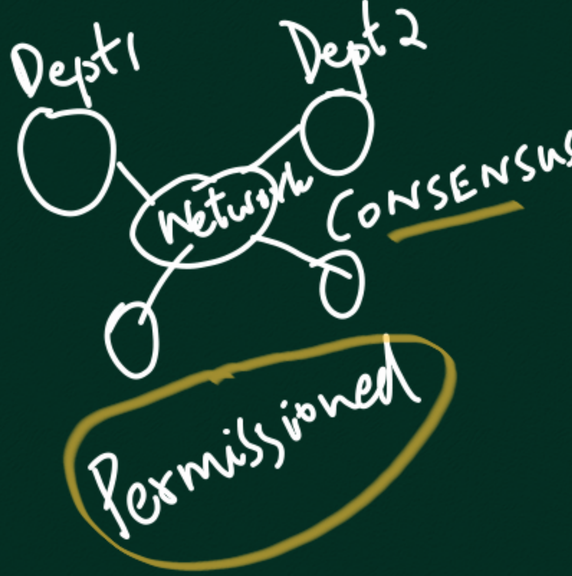
e-Gov

chameleon hashing

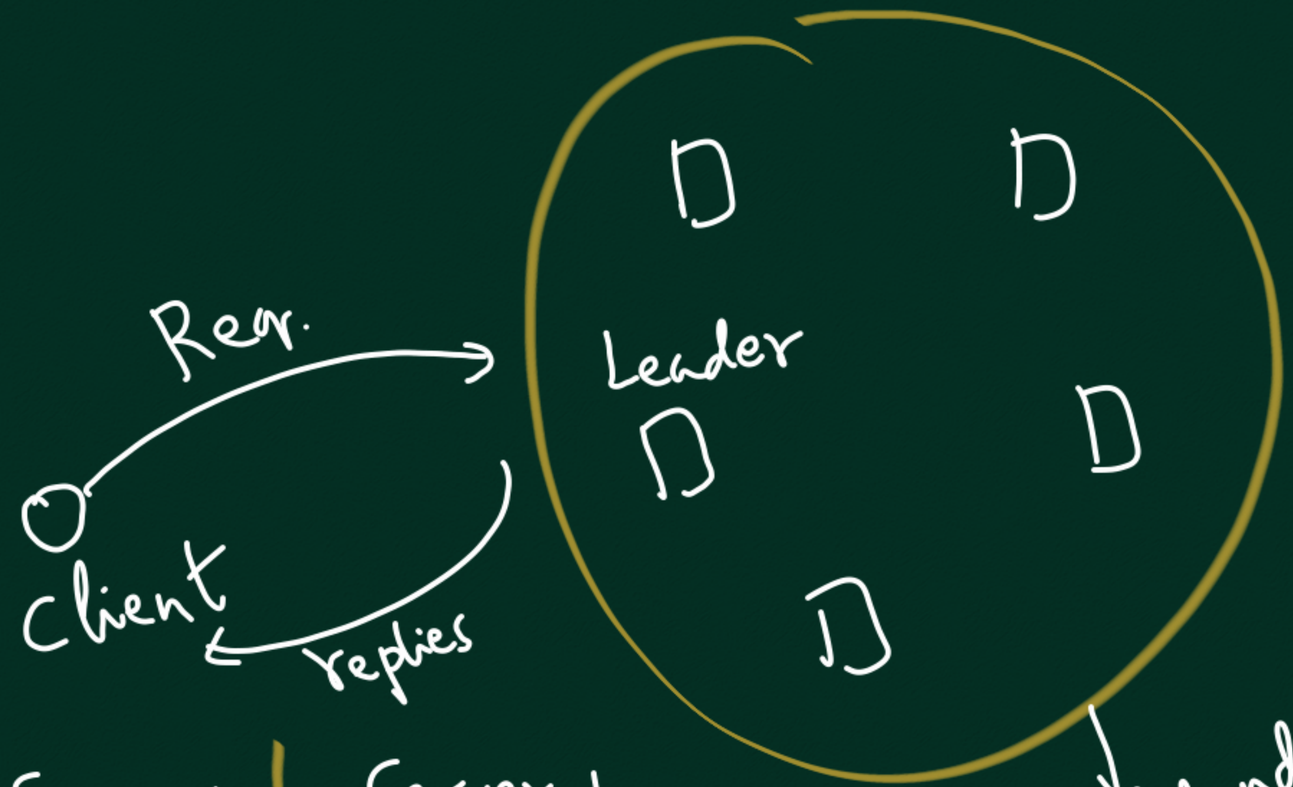
- Collecting tax
- Land records

Ledger available to public

Property: One Smart Contract



Lamport,
N. Lynch (?)
Liskov



x:=0
set x=5

Client → Server → Server-1
← Ack
→ Server-2
Ack
gave (read)
val. of x
x=5
x=0

Single server case

Distrib.

Looks like single server

redundancy

Threat Model: Specifies what can go wrong

- FAIL-STOP: A node can fail and stop responding
- FAIL-STOP-RESTART: Fail, but later restarts with state at time of failure
- BYZANTINE FAULTS: Any behaviour deviating from specified protocols. (allows malicious behavior)

Network Model: Specifies delays (latencies) and message loss properties

- SYNCHRONOUS: Known upper bound on max. delay a msg will experience.
- PARTIALLY SYNCHRONOUS: There is an upper bound on delay, but this value is unknown to the protocol.
- ASYNCHRONOUS: No upper bound on msg delay

SAFETY: "Nothing bad can happen"

Ex: Once a log entry has been made
it will remain forever

LIVENESS: "Without much delay, there
will be a response to any request"
↙ from D.S.

"Good things happen often"

RAFT

↙

Fail-Stop-restart

PBFT

↙

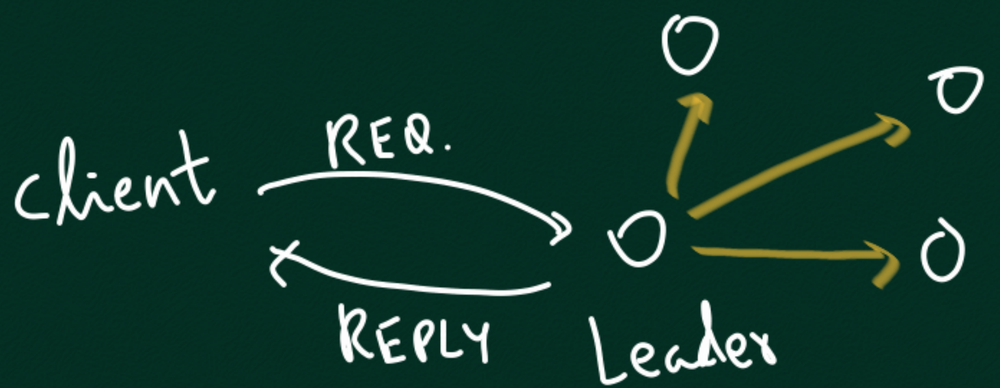
Byzantine Faults

Hyperledger Fabric	R3-CORDA	Quorum (from Ethereum)

CDAC

RAFT (2014, ONGARO et al.)

FAIL-STOP-RESTART



Replicated State Machine

x: 12

y: 9

Log

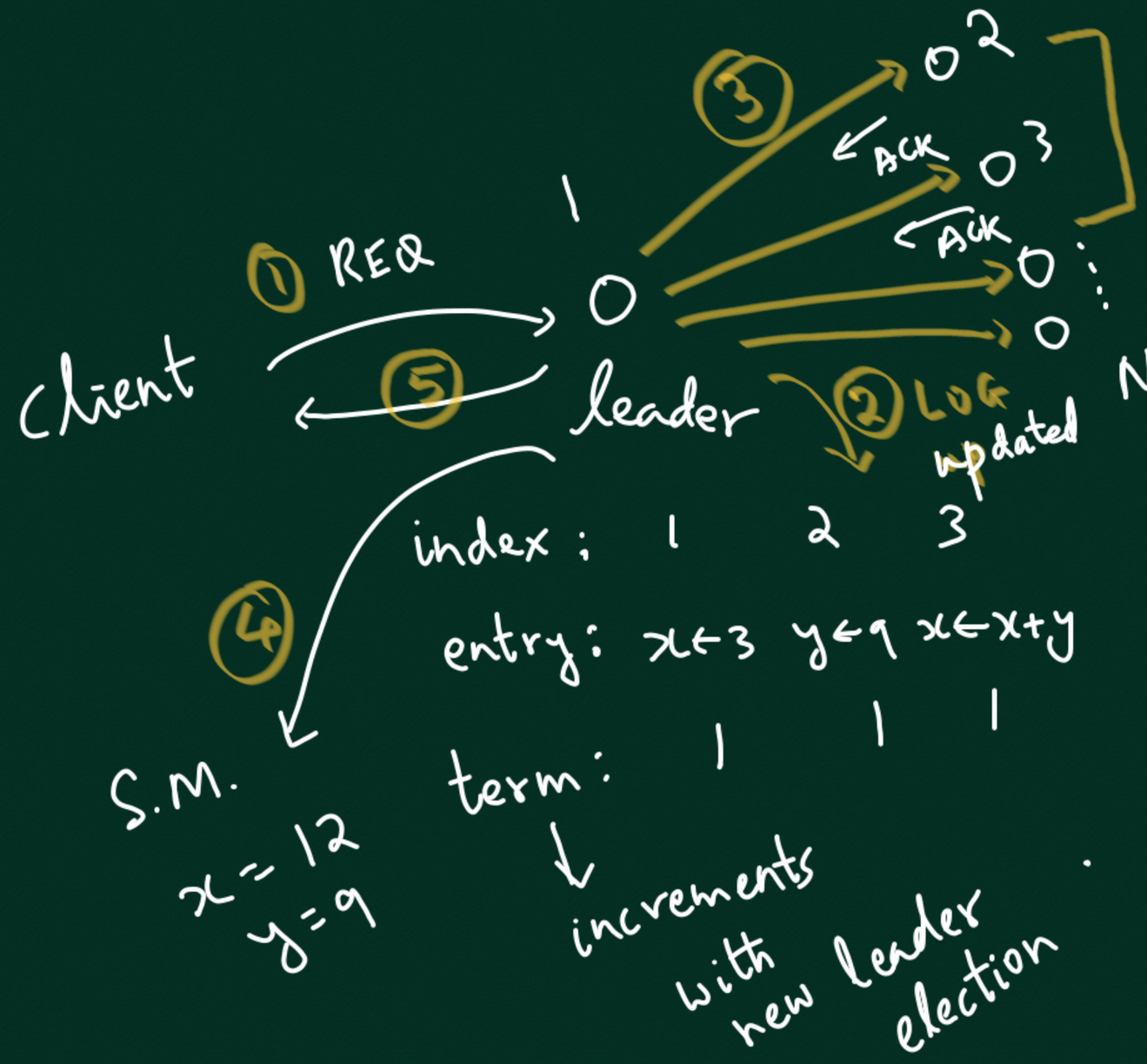
index	1	2	3	4
entry	x←3	y←9	x←x+y	...

last commit

SAFETY : Never return an incorrect result

(from RAFT paper) under threat model, network partition, packet loss, message reordering, message delays

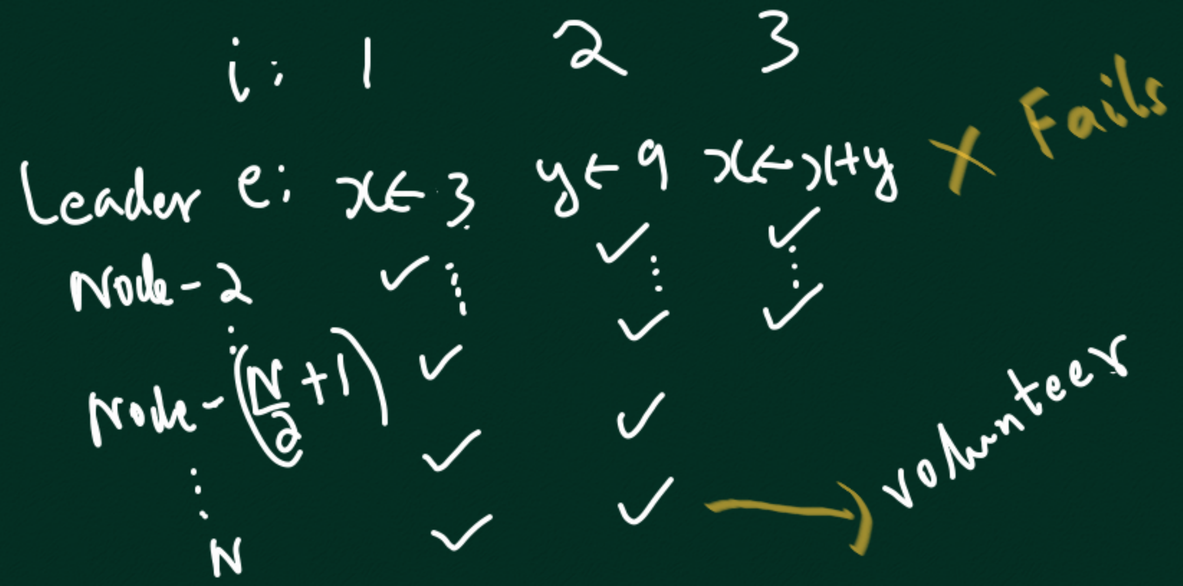
AVAILABILITY (LIVENESS): Fully functional as long as a MAJORITY of servers are operational and Can communicate with each other and the client



Intuition: Wait for $> \frac{N}{2}$ nodes to ACK, then commit, update state machine (local) and reply to client

How to elect new leader with correct state?

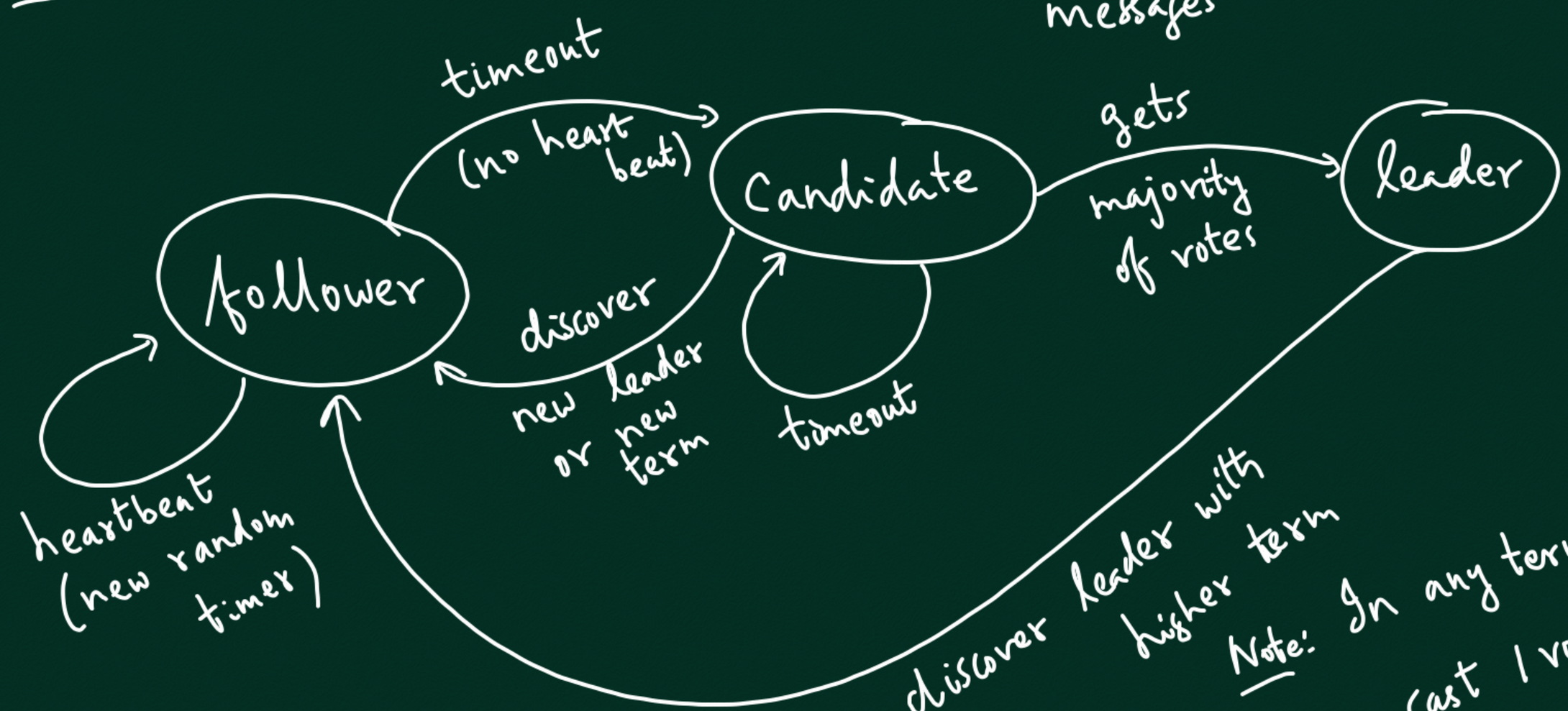
- Only consider FAIL-STOP (not FAIL-STOP-RESTART)
- Leader gets single req. from client every time, new request sent by client only after leader replies to client



Idea: Somene volunteers to be a candidate for leader and shares his log

Other node votes only if its own log entries are a prefix of candidate's log

State Diagram : Note: Leader sends periodic heart-beat messages



Note: In any term, a node can cast 1 vote at max.

EWRUHKQS