

# Paxos / Multi-Paxos

Admin

- Midterm

+ DURING CLASS

+ OPEN BOOK

- Papers } No internet connected  
- Notes } device.  
- ... }

+ COVERS EVERYTHING UP TO THIS CLASS

- Know what the async model is

- Fairness

- Linearizability, seq cst

- Raft, Paxos

+ QUESTIONS WE DISCUSSED IN CLASS OR SHOW UP IN NOTES  
ARE A GOOD BENCHMARK

- FINAL PROJECT PROPOSAL

↳ CAMPUSWIRE POST

Where we are

- RAFT

- BUILDING BLOCKS

## QUORUM INTERSECTION!

- Replication

↳ Commit entry by replicating to at least a quorum

Goal: Committed entries will never be lost

- Leader election

\* → Leader's log is authoritative

Goal: Any node elected leader has a log containing all committed entries

How

- Check log completeness when voting for a leader

- Require votes from a quorum

MULTI PAXOS

BUILDING BLOCKS: QUORUM INTERSECTION

- Replication

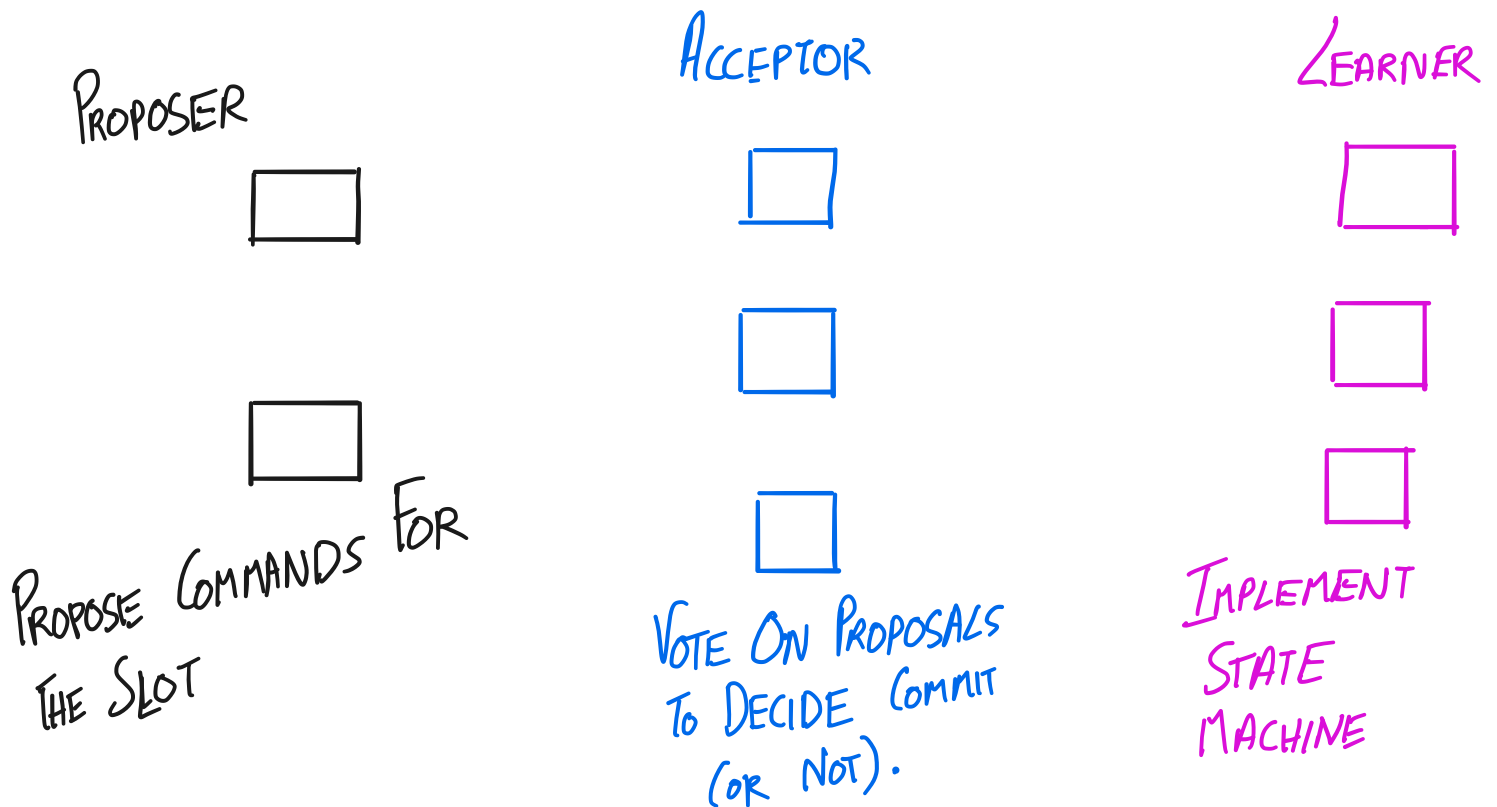
↳ Commit entry by replicating to at least a quorum

Goal: Committed entries will never be lost

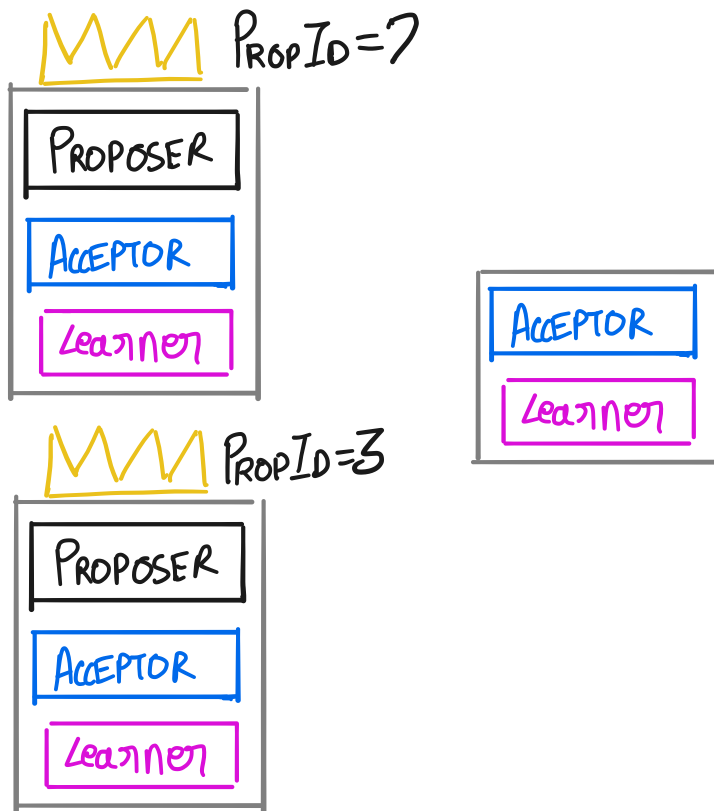
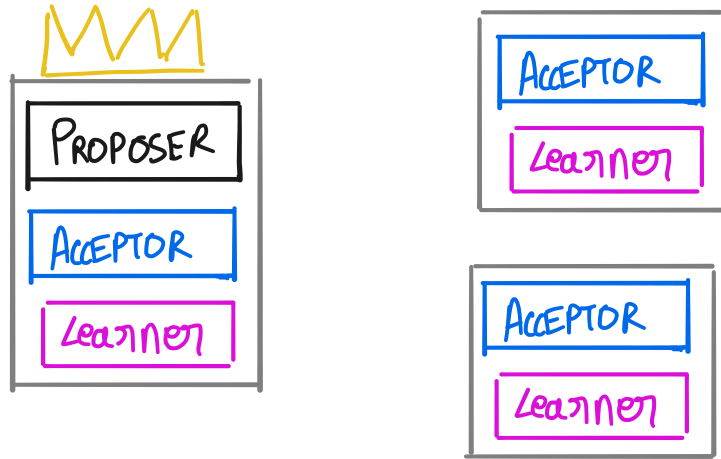
Do not depend on leader election to achieve this goal.

↳ Leader's log is not special.

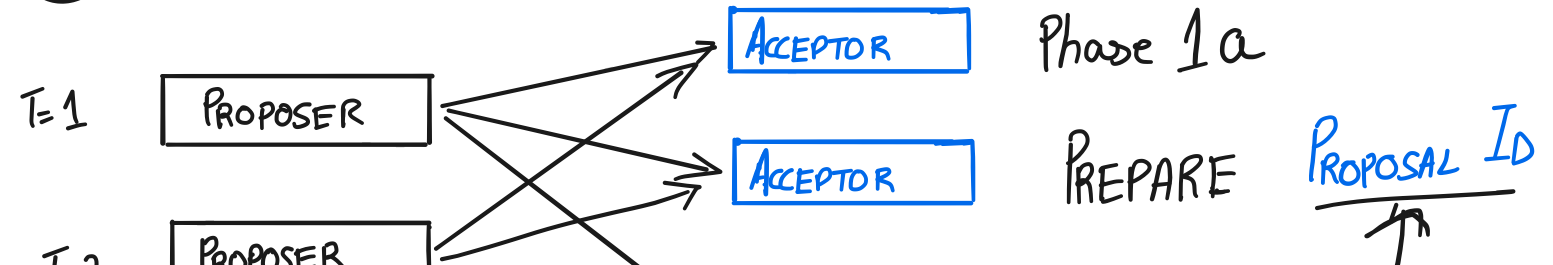
Paxos/Synod: Agreement on one log entry (slot/index)



Mapping this to a familiar picture

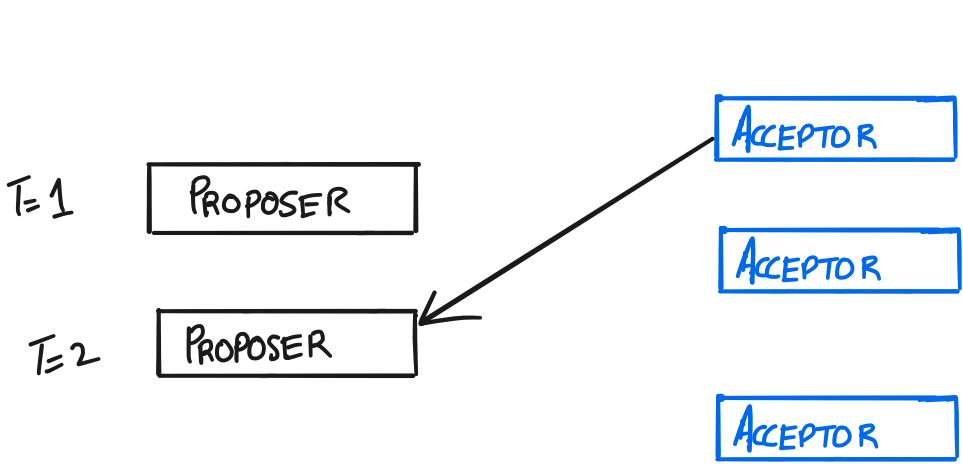


① PHASE 1<sup>o</sup>: Figure out what value to use





INCREASING NUMBER



Phase 1b

PROMISE / PREPARE

RESPONSE

- PROPOSAL ID
- PREV VALUE if any
- PROPOSAL ID for prev accepted value.

◦ Reminders: Only looking at 1 slot

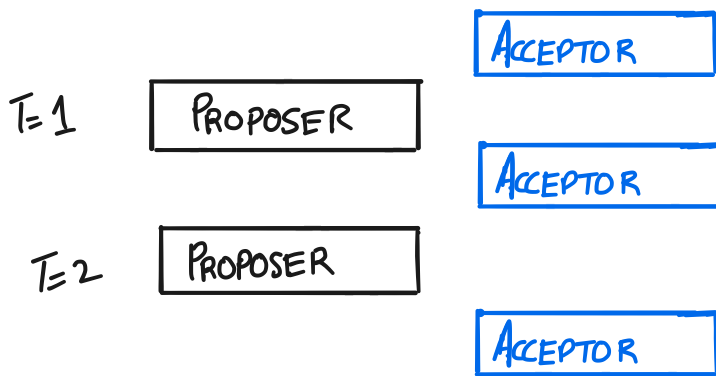
- ACCEPTOR sends PROMISE TO PROPOSER If

Has not previously seen PREPARE with higher proposal ID.

- Includes value if previously **ACCEPTED** a value



# \* COMPUTE VALUE FOR PHASE 2



- Wait for Phase 1b (PROMISE / PREP RESP) responses from QUORUM of acceptors

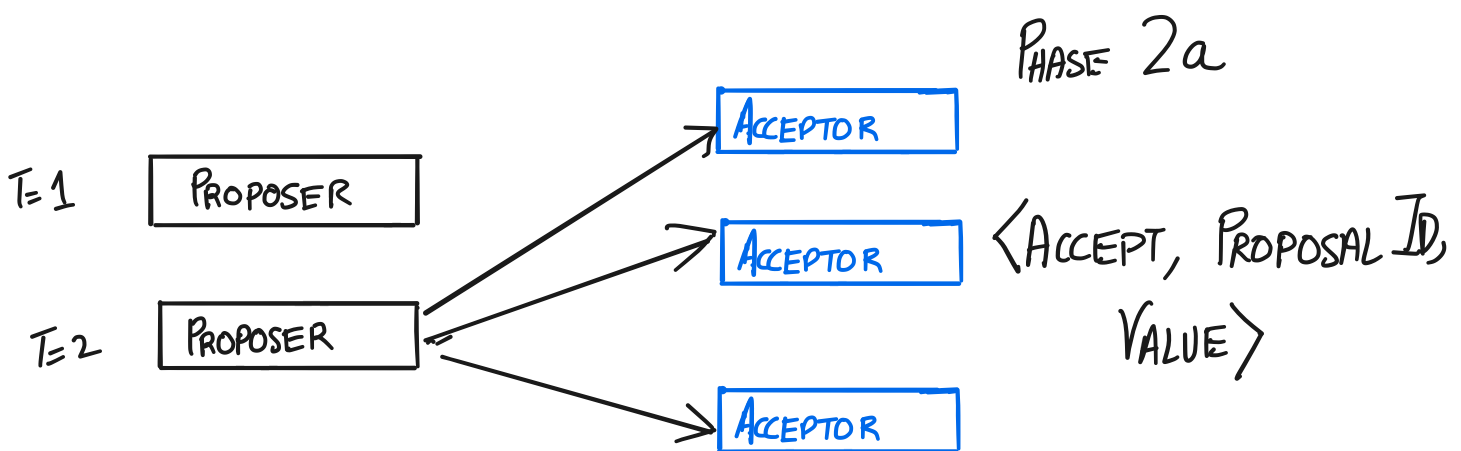
< PROMISE, PROPOSAL ID, PREV. ACCEPTED VALUE, PROP ID WHEN ACCEPTED >

- If no phase 1(b) message contains a value :- Propose any value
- If one or more phase 1(b) message contains a value :-  
 Must use value accepted w/ highest prop ID

CLAIM: A committed value will always be used for phase 2.

Why?

## II PHASE 2: REPLICATE CHOSEN VALUE



At acceptor:

If PROPOSAL ID = Prop ID of last  
last promise

ACCEPT Value

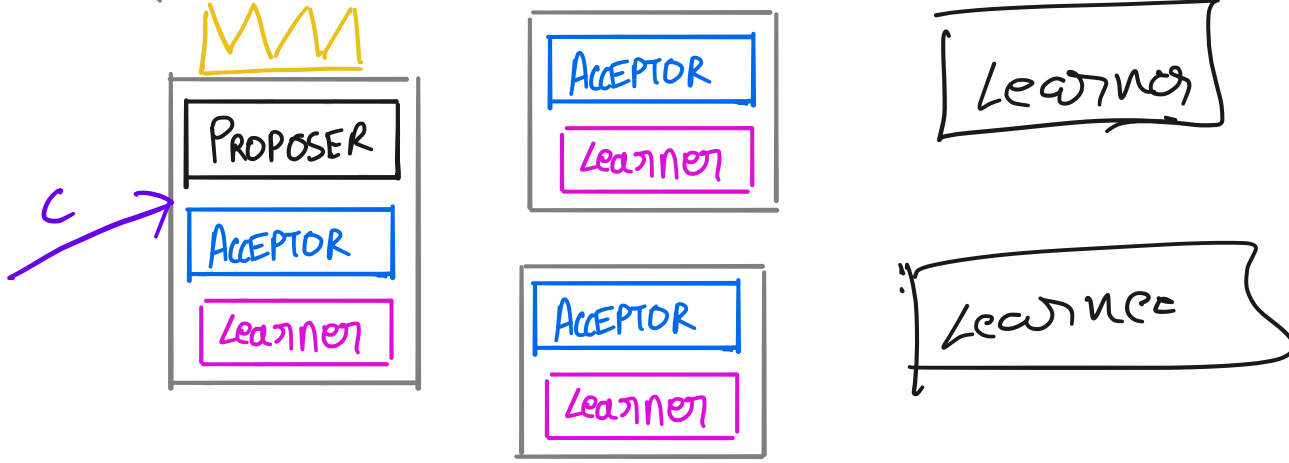
Tell learners: Phase 2b message

Note: Need to count # of accepts

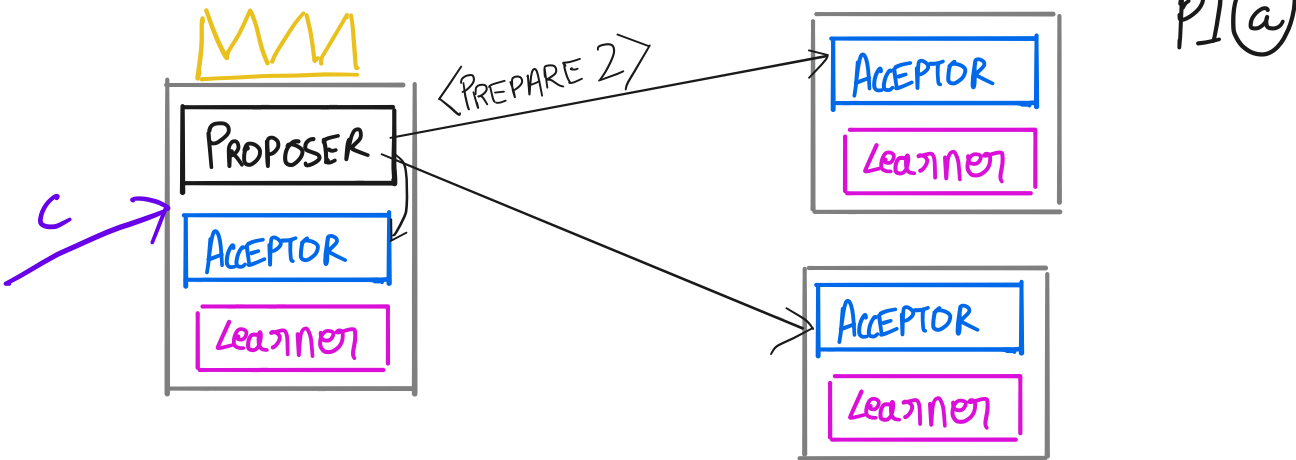
(P2b msgs) to decide if committed.

Mapping this back

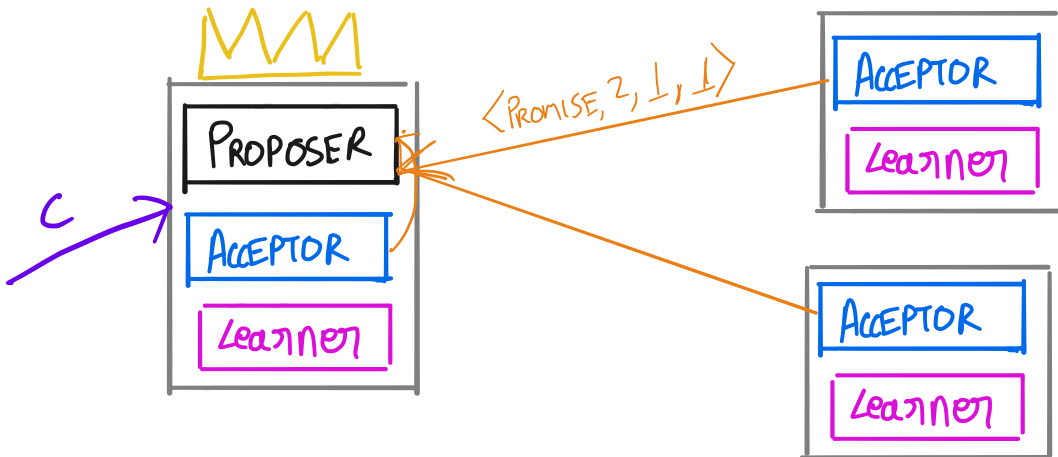
PROP ID = 2



PROP ID = 2

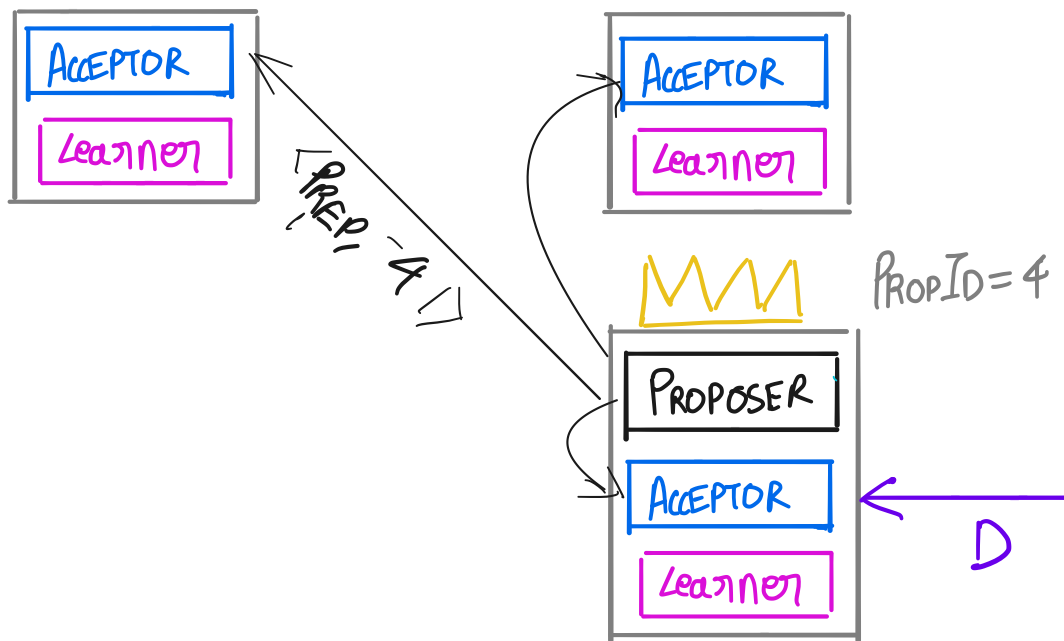
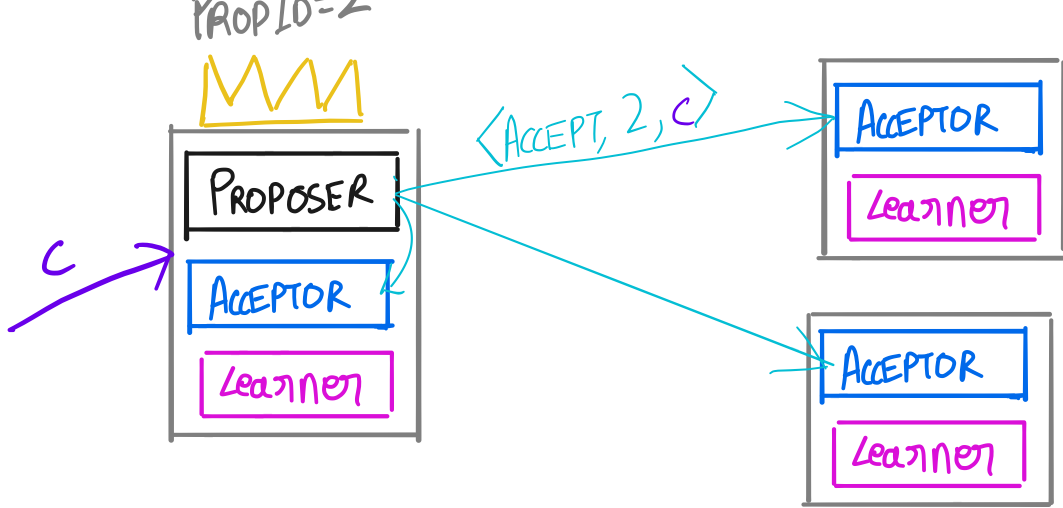


PROP ID = 2



P 1 - 2





Q: When is command committed?

Q: When can a learner apply a command?

Q: Requirements for Proposal ID:

# MULTI PAXOS: EXTENDING TO MANY SLOTS

- What is missing: Leader needs to know what index to use for new commands

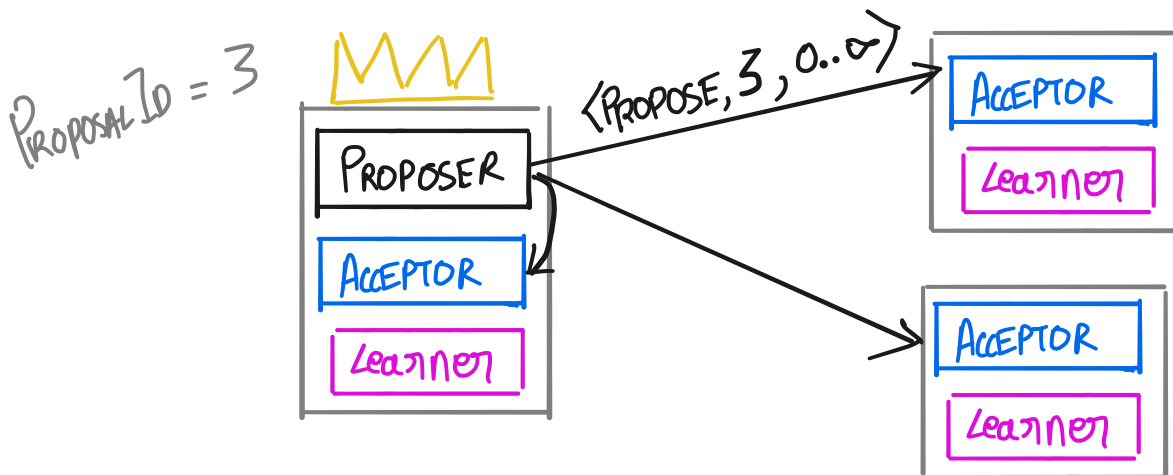
↳ Equivalently: Leader needs to know what indices are used

For any index  $i$

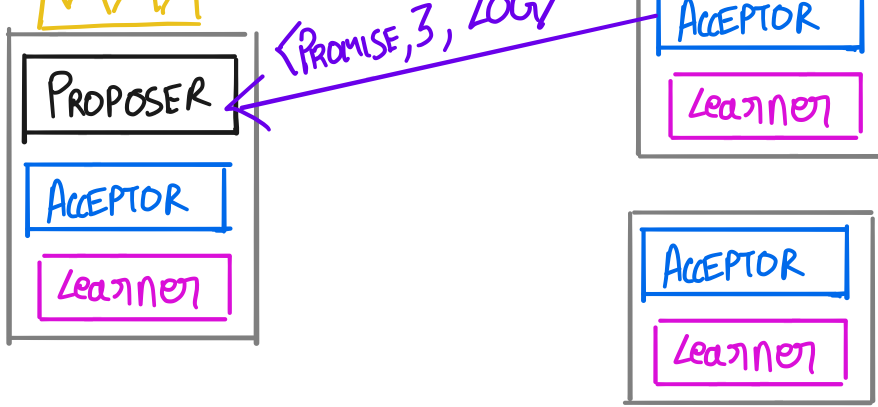
I(a)  $\langle \text{PROPOSE, } \underline{\text{PROPOSAL ID}} \rangle$

I(b)  $\langle \text{PROMISE, } \underline{\text{PROPOSAL ID}}, \text{PREV. ACCEPTED VALUE, PROP ID WHEN ACCEPTED} \rangle$

• Tie proposal ID to leadership term.  $\uparrow$  Part

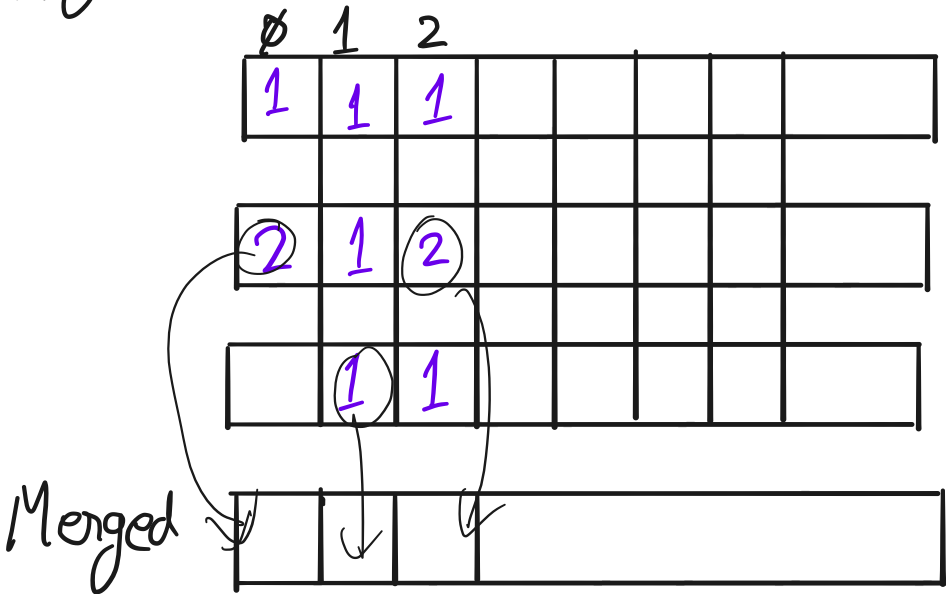


PROPOSAL ID

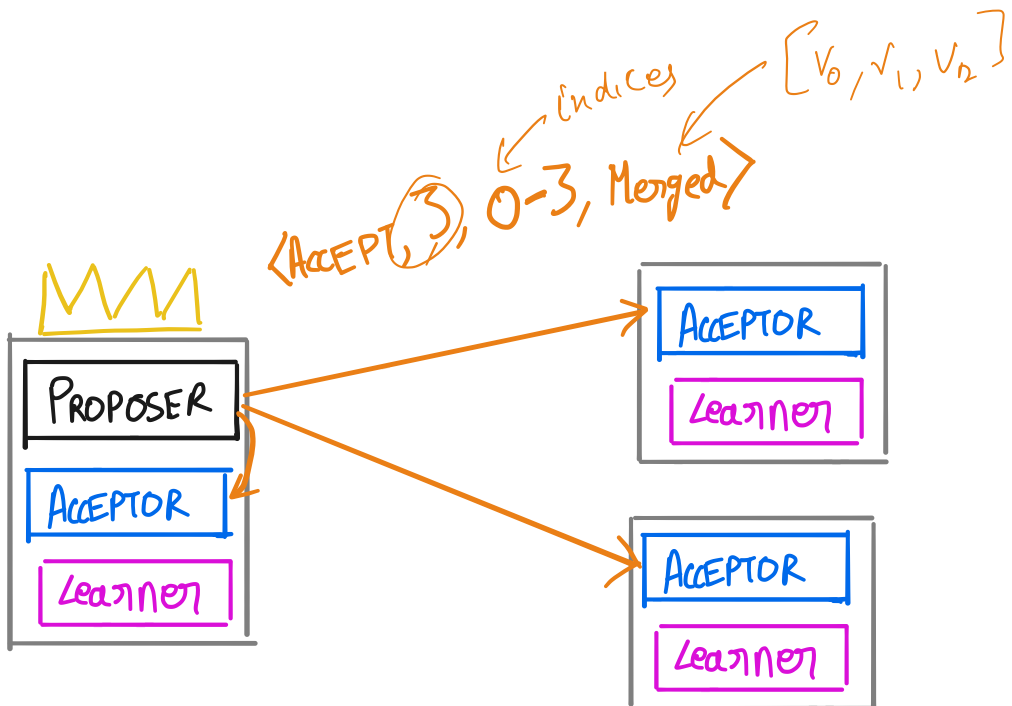


# Merging logs

- Apply rule from before



PROPOSAL ID=3



Log after  
accept is  
processed.

	0	1	2					
$v_0$	$v_1$	$v_2$						
$v_0$	$v_1$	$v_2$						
$v_0$	$v_1$	$v_2$						

Q. When is it safe to execute a command  
logged by a previous leader?

Raft<sup>o</sup>

Multi Paxos<sup>o</sup>

Why all this flexibility?

PROPOSER



ACCEPTOR



LEARNER



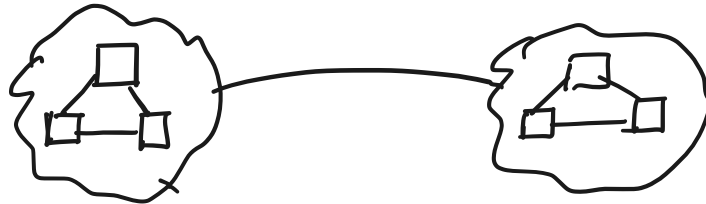
PROPOSE COMMANDS FOR THE SLOT

VOTE ON PROPOSALS TO DECIDE COMMIT (OR NOT).

IMPLEMENT STATE MACHINE

- Disk paxos: Use processors as proposers  
Processor + Disk as acceptors  
Disk as learners

- Mencius/ePaxos: Scale in asymmetric setting



- Vertical Paxos: Reconfig without stopping by changing acceptor sets

[Similar to Raft reconfig]

Things not mentioned

- Failure detection: knowing when a new leader should be elected

↳ PML: Leader leaves?

- Leader election: who becomes leader?

$P_0$   
 $P_1$

