Distributed Systems } Paxos

But First... Administrivia

- Final Exam
  - Registrar Scheduled May 12
    - ↳ Does Not Work - Ask GSAs
  - Proposal: May 7
    - ↳ Reading Day :- No Classes

- Midterm: Next Week, In Class
  - Covers Everything Up To Today
  - Open Book
    - ↳ Papers, Notes, Etc.

→ ELECTRONIC DEVICES

↳ CAN USE IPAD ETC.

BUT

- MUST DISCONNECT FROM INTERNET: DOWNLOAD AHEAD OF TIME
- NO CHAT BOTS, ETC.

PAXOS

- Why? Reinforce ideas around correctness

- General form: 3 types of nodes.
  SYNOD: Log with 1 slot/index

① PROPOSE VALUES: Suggest command to set index to

② ACCEPT VALUES: CHOOSE WHAT VALUE TO ACCEPT & COMMIT

③ LEARN VALUES: EXECUTE COMMAND

f+1 | 2f+1 | f+1

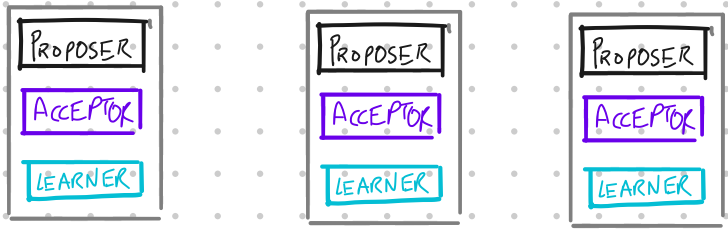
WHY SEPARATE THIS OUT?

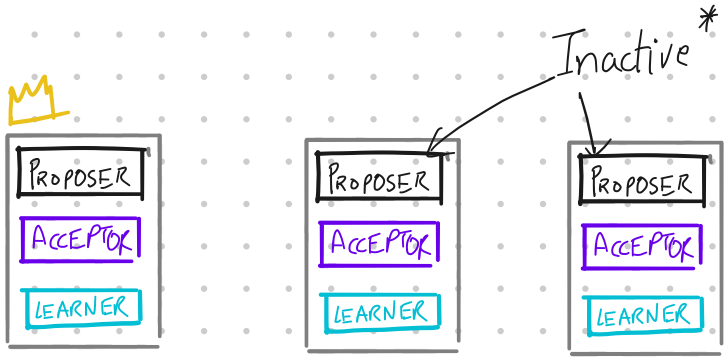
RECIPE FOR CONSTRUCTING RSM PROTOCOLS

↳ DIFFERENT TYPES OF PROPOSERS/ACCEPTORS/LEARNERS

RETURN TO THIS AT THE END OF CLASS.

MULTIPAXOS / CHUBBY : OUR SETTING





SYNOD: THE ALGORITHM FOR A SINGLE SLOT
(WE LOOKED AT THIS LAST CLASS)

GOES IN ROUNDS, EACH ROUND DENOTED BY A "PROPOSAL #"

REQUIREMENTS

COMMIT

- CHOOSE SOME VALUE, EVENTUALLY (Avoids trivial solutions)

P1 ↳ ACCEPTOR ACCEPTS FIRST VALUE

↳ Not the same as choosing!

P2 - ONCE A VALUE √ IS CHOSEN, NO OTHER VALUE

WILL BE CHOSEN [C.F., STATE MACHINE SAFETY FROM LAST CLASS]

↳ P2a. IF VALUE V IS CHOSEN AS PROPOSAL N, THEN ANY PROPOSAL WITH #>N ACCEPTED BY AN ACCEPTOR MUST HAVE VALUE V

[NOTE, INCLUDES VALUES ACCEPTED BY ACCEPTORS WHO HAVE MISSED ALL PROPOSALS/MESSAGES SO FAR]

P2b. IF VALUE V IS CHOSEN AS PROPOSAL N, ~~THEN ANY PROPOSAL WITH #>N ACCEPTED BY AN ACCEPTOR MUST HAVE VALUE V~~ ANY HIGHER # PROPOSAL ISSUED BY A PROPOSER HAS VALUE V.

[Note, must be true even for proposals issued by a different proposer in the future.]

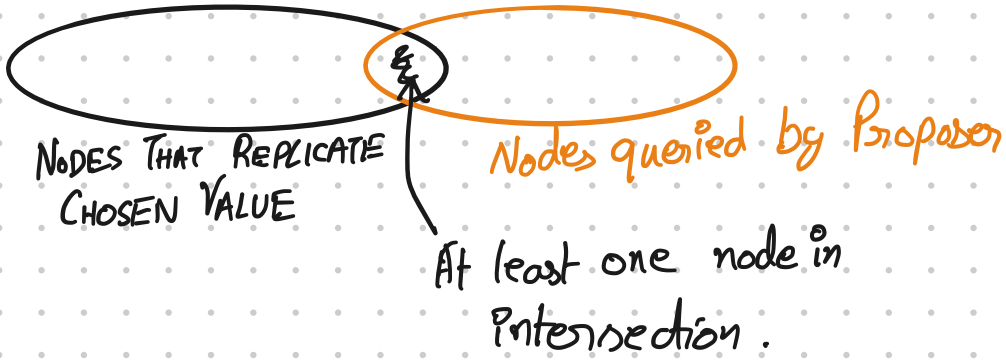
Observe - P2b ⇒ P2a

- For P2b to hold proposers need to be able to find any chosen (committed) value V

- CAN USE QUORUM INTERSECTION FOR THIS (P2C)

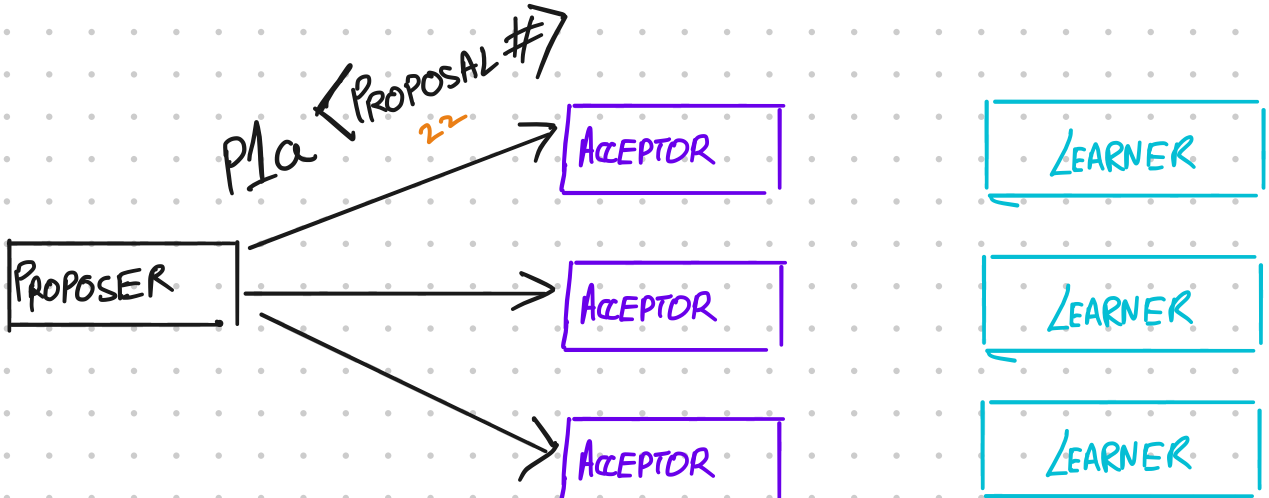
REQ: ① COMMITTED CHOSEN VALUES MUST BE ACCEPTED BY A QUORUM OF ACCEPTORS

② PROPOSER MUST CHECK WITH A QUORUM BEFORE PROPOSING A VALUE



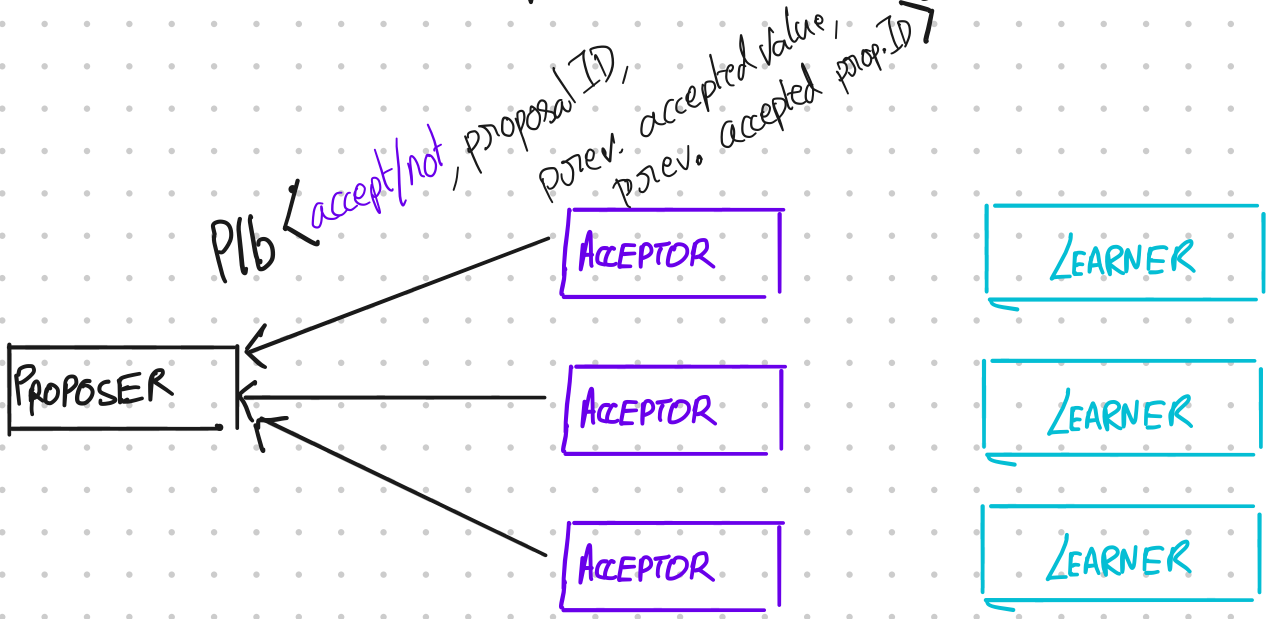
P2a/P2b ⇒ If a value V was committed w/ proposal # P, all accepted proposals w/ prop # p+1 or larger contain V

↳ Value associated with largest proposal # is the one likely committed



Goals (a) Query to find any previously accepted values.

(b) Prevent any pending proposals (with lower ID) from making progress

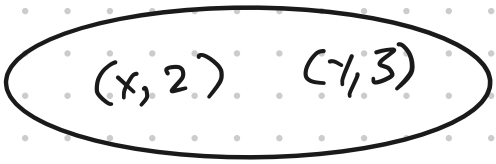


- FALSE, 24, ... ← 22 is too low

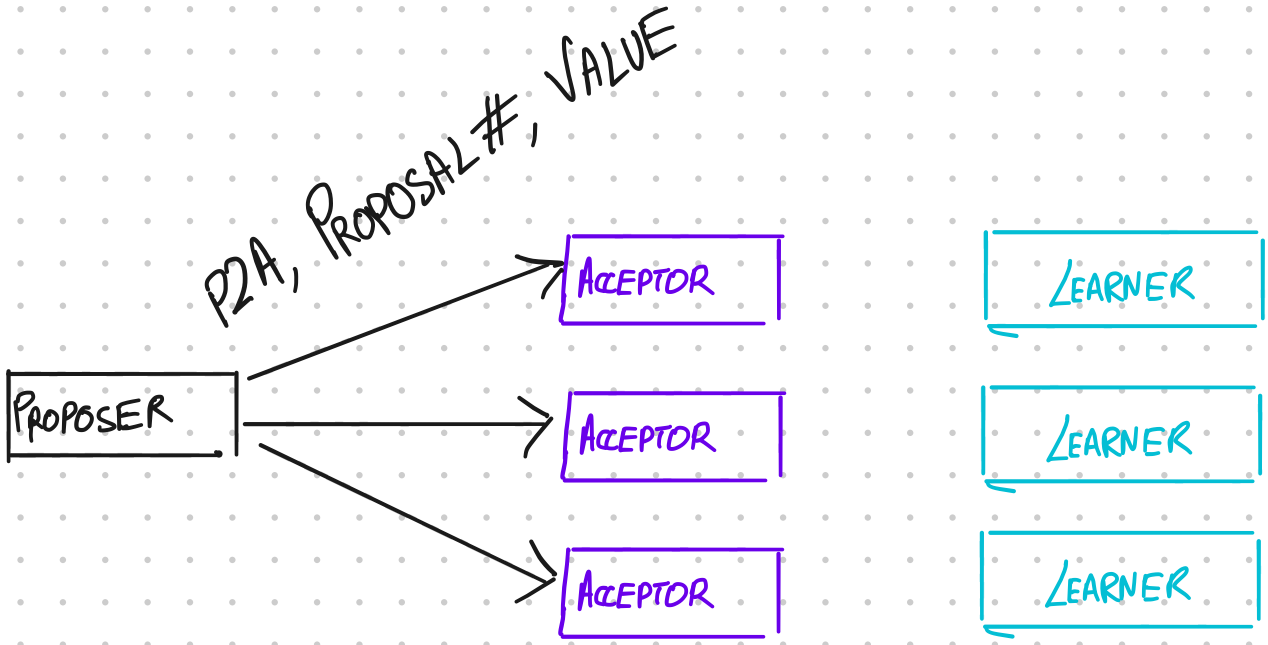
- True, 22, ⊥, – ← Nothing accepted

- True, 22, X, 2 ← x accepted as prop. ID 2

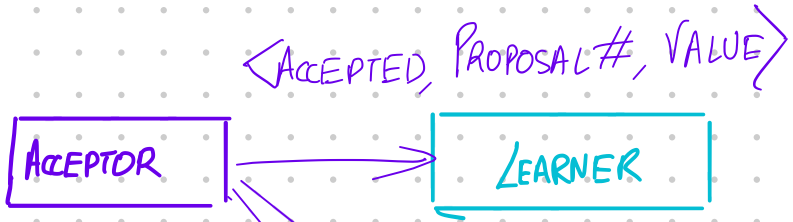
	Value to propose	Possibly committed
⊥, ⊥	Any	Nothing
(x, 7) ⊥	X	X or ⊥



y

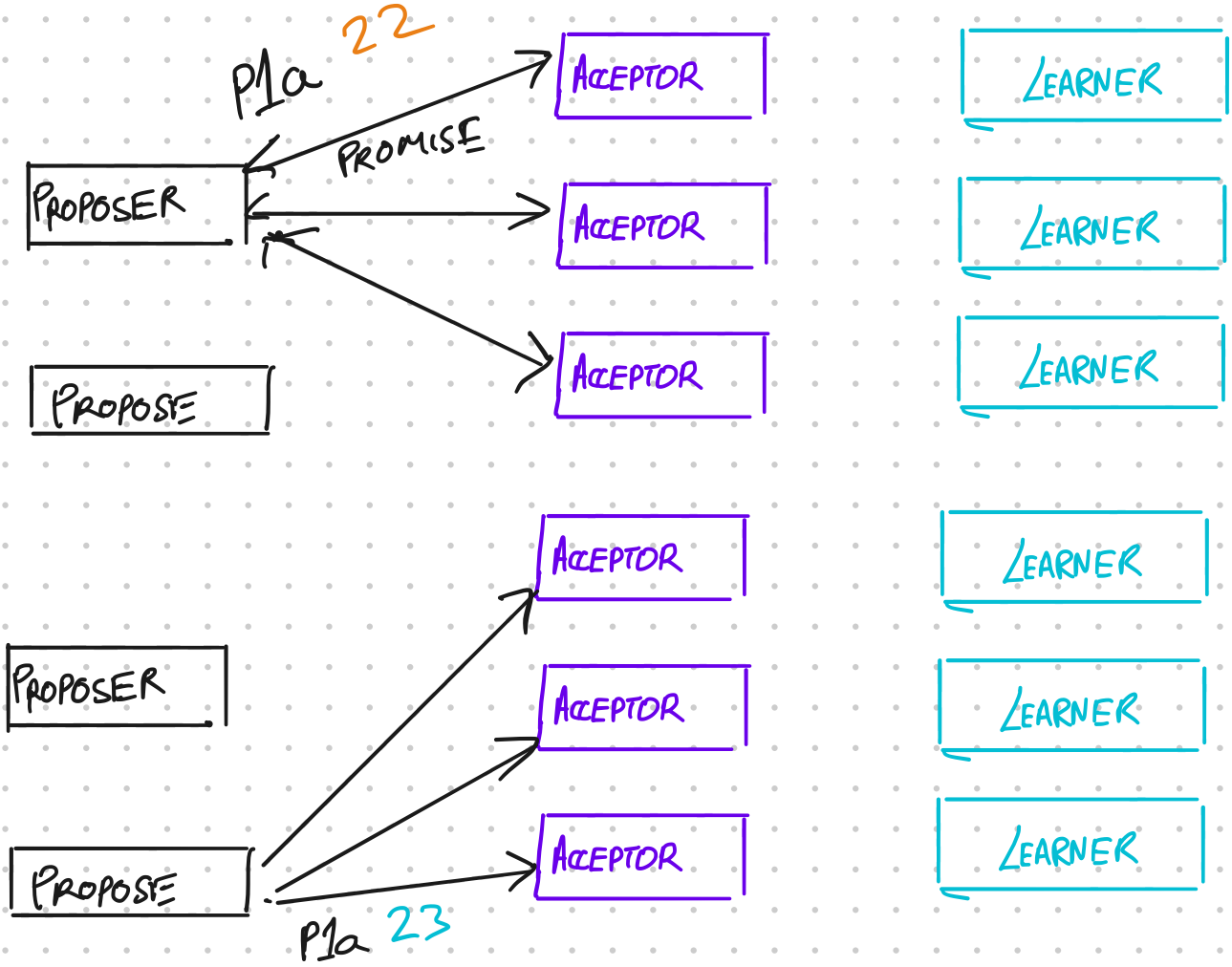


When should the ACCEPTOR not accept?



- When should learners execute/act on value?

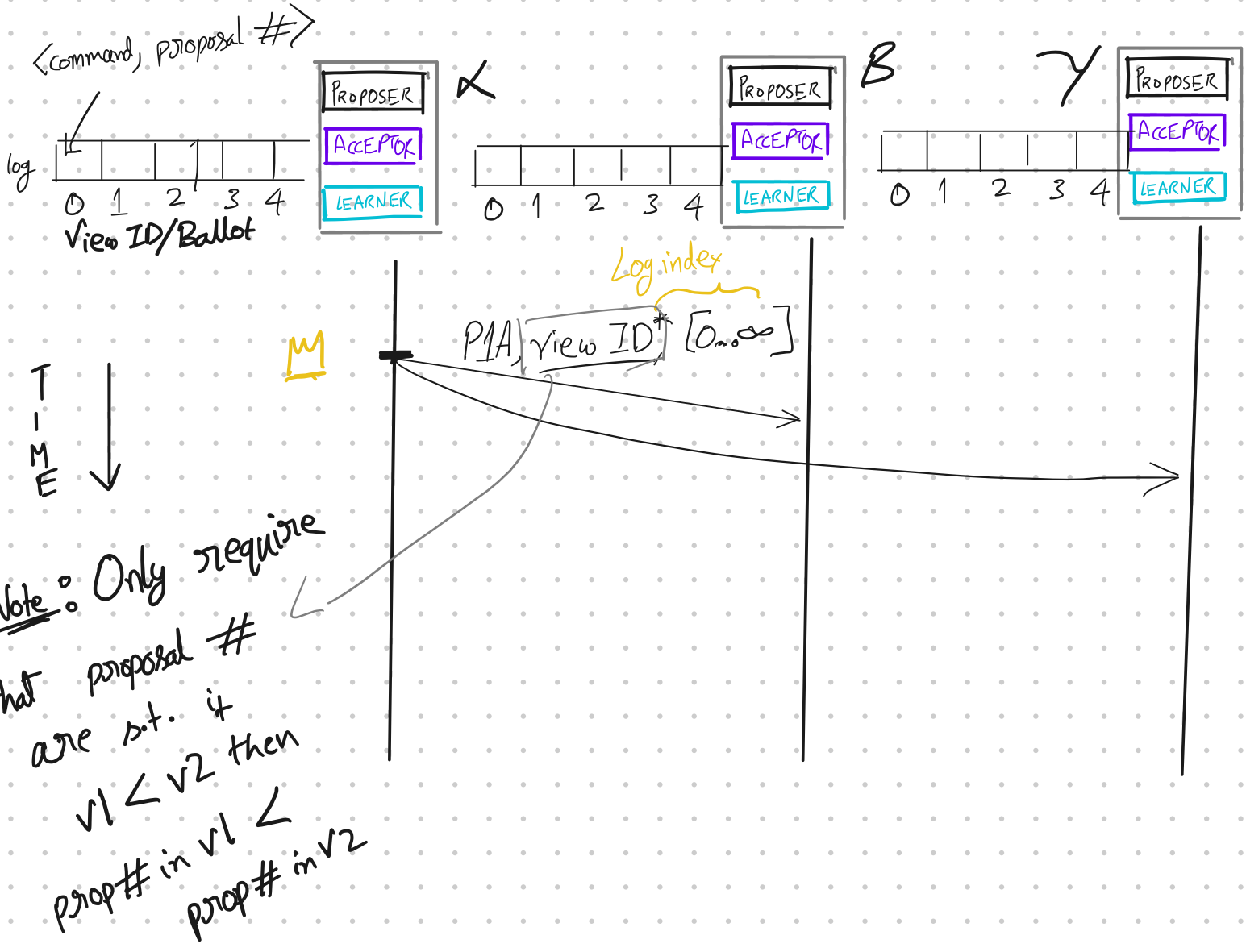
Why leader?



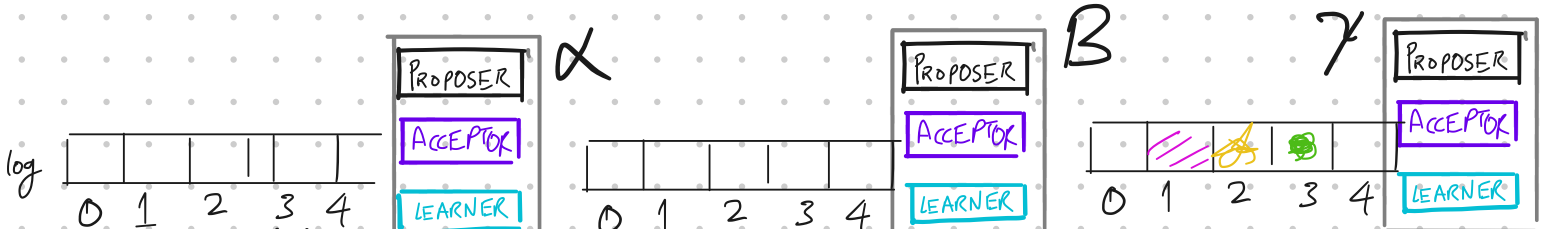
Duelling proposers are a problem!

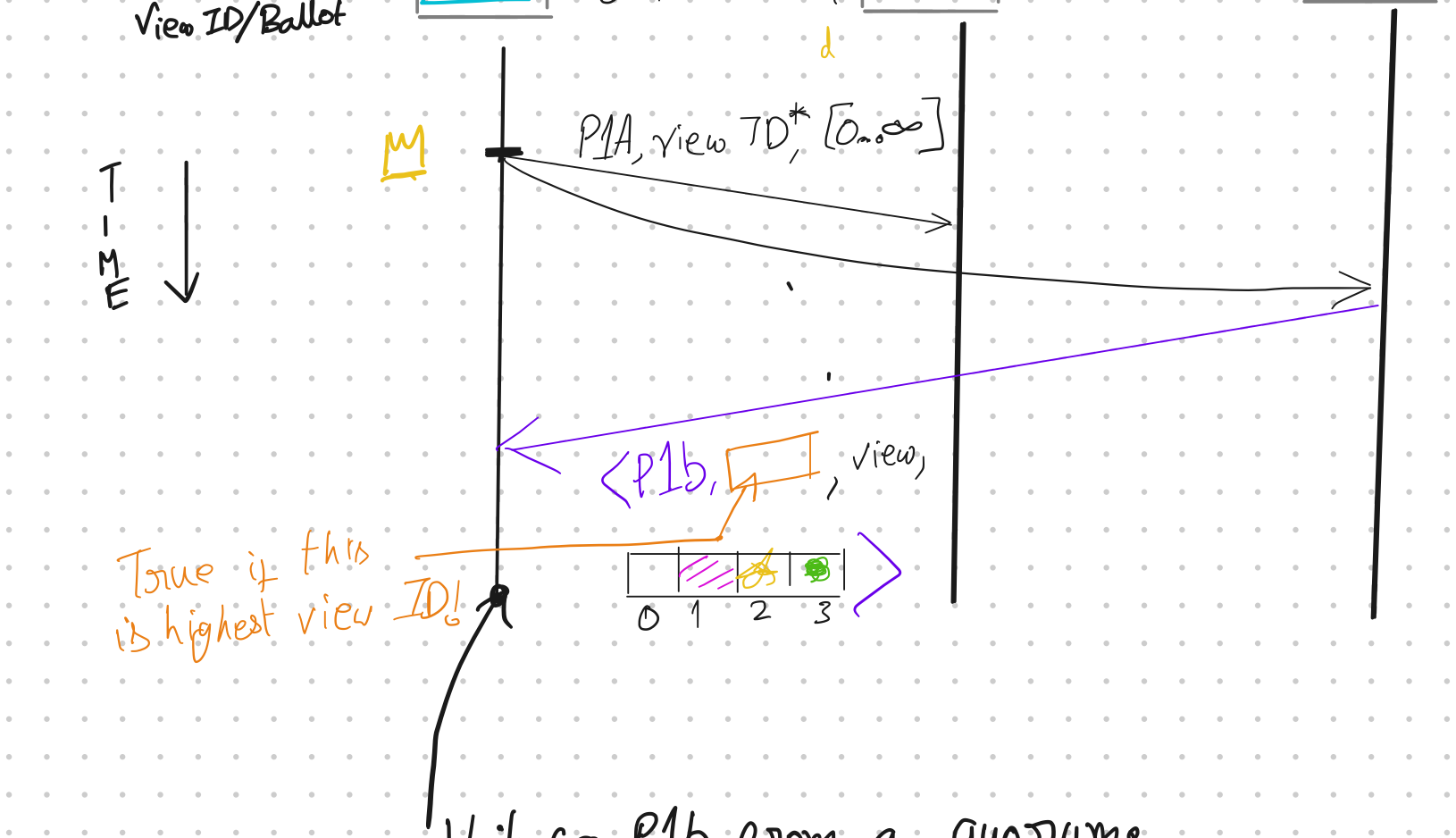
Putting it all together

So far: Focus on a single log idx. But what about RSMs? "Fast Paxos" Lamport '05



Note: Only require that proposal # are s.t. if v1 < v2 then prop# in v1 < prop# in v2

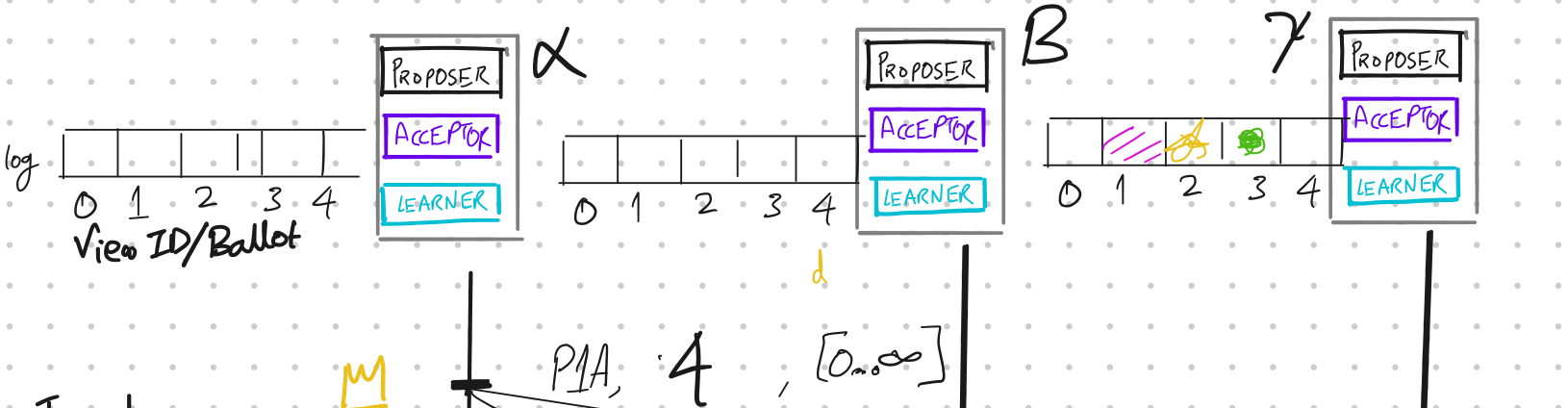


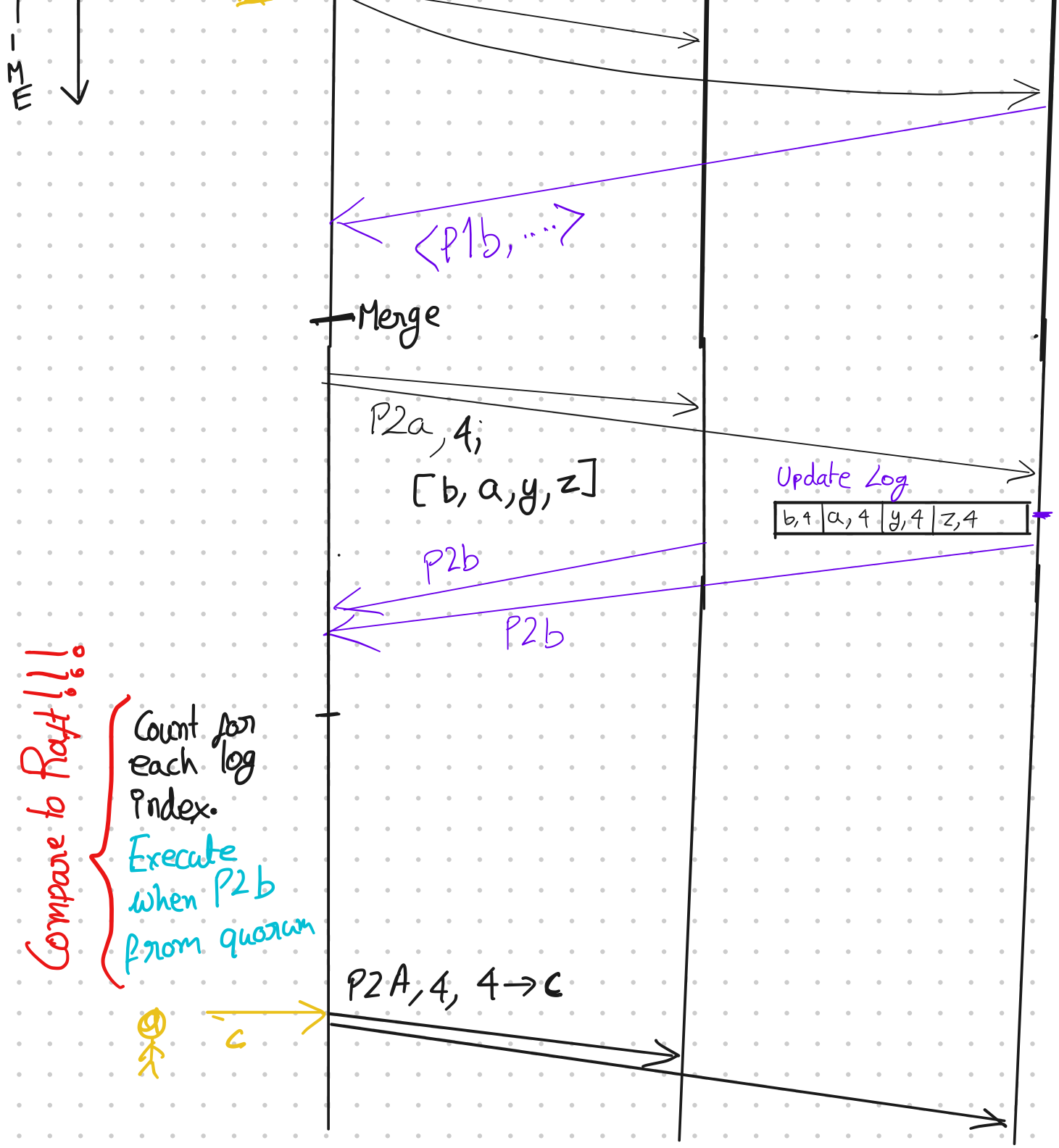


Wait for P1b from a quorum.

Merge logs

	0	1	2	3
α	b,1	x,1	y,1	z,1
γ	1	a,2	y,1	z,1
ML:	b	a	y	z





Electing a leader?

- Only requirement is Election Safety; that is at most one leader

- Any protocol that meets this requirement

suffices

Changing Configuration

- Hard problem, for the same reason as last week: need to avoid quorums of nodes that are not up-to-date
  Vertical Paxos.

What does this generality help with

- Disk Paxos
  ↳ Acceptors and learners are networked disks
  ↳ Read, Write

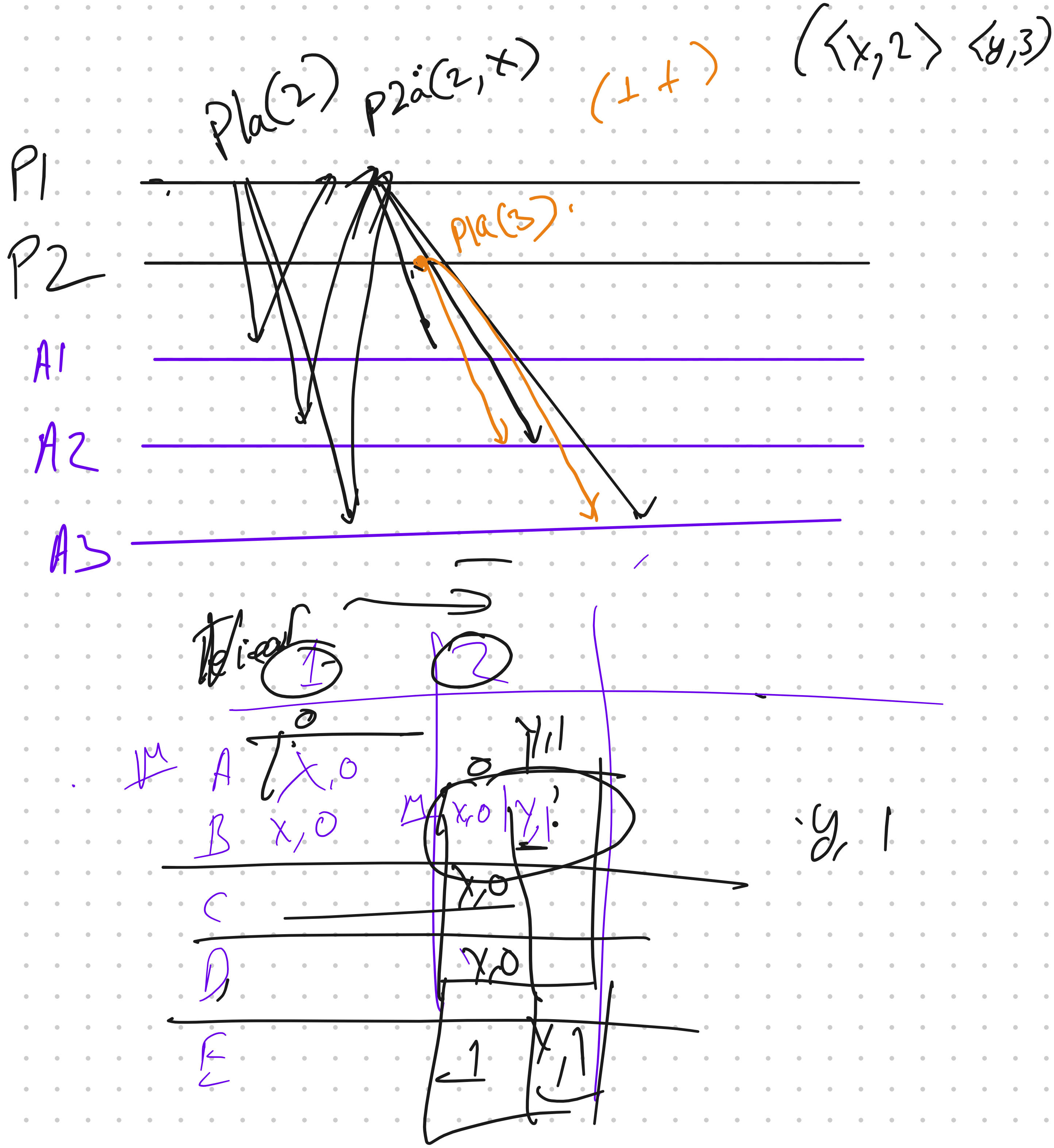
- Mencius
  ↳ Multiple leaders for multiple datacenters
  ⇒ Later improved by EPaxos

...

Taking a step back

- RSMs
- Quorum intersection

- Leader safety
- State machine safety



P1
P2
A1
A2
A3
P1a(2)
P2a(2,x)
p1a(3)
(⊥ ⊥)
(⟨x,2⟩ ⟨y,3⟩)
1
2
A
B
C
D
E
x,0
y,1
x,1