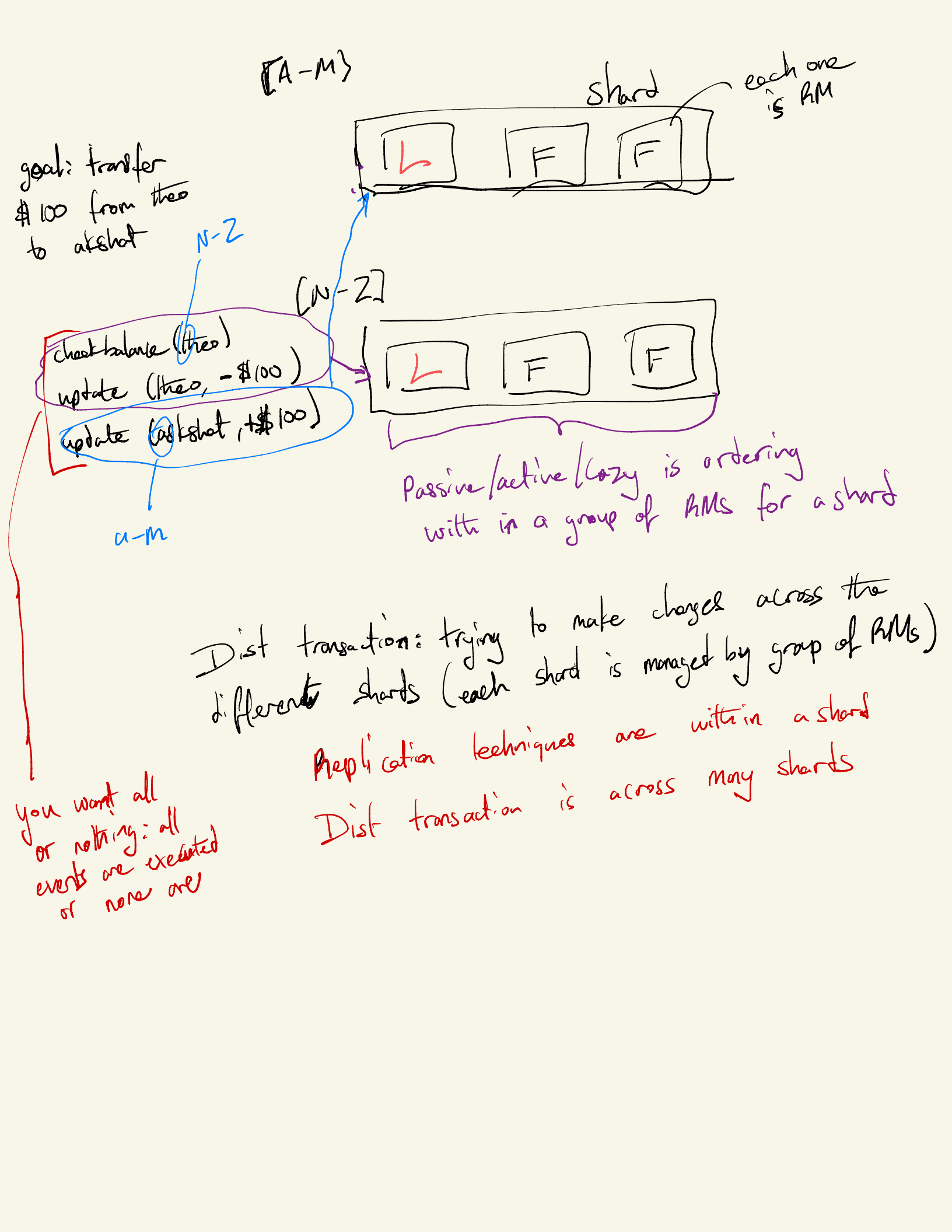
CSCI 1380 : Day 18 (Dist Transaction)

Today

1. Distributed Transaction
   a. ACID
   b. Locks
   c. 2PC (two phase commit)
2. Ordering Guarantees


[A-M]
shard
each one is RM
L
F
F
goal: transfer $100 from theo to akshat
N-Z
[N-Z]
checkbalance (theo)
update (theo, -$100)
update (akshat, +$100)
L
F
F
a-m
Passive/active/Lazy is ordering with in a group of RMs for a shard
Dist transaction: trying to make changes across the different shards (each shard is managed by grop of RMs)
Replication techniques are within a shard
Dist transaction is across many shards
you want all or nothing: all events are executed or none are

ACID

semantics

atomicity

* all events execute or none
* 2PC (two phase commit)

Consistency

* you want each shard to go from safe state to safe state
  * log safety

isolation

* each transaction has perception that it's executing in isolation
* execute transactions in a serial order one after the other
* locks (pessimistic V optimistic)

Durability

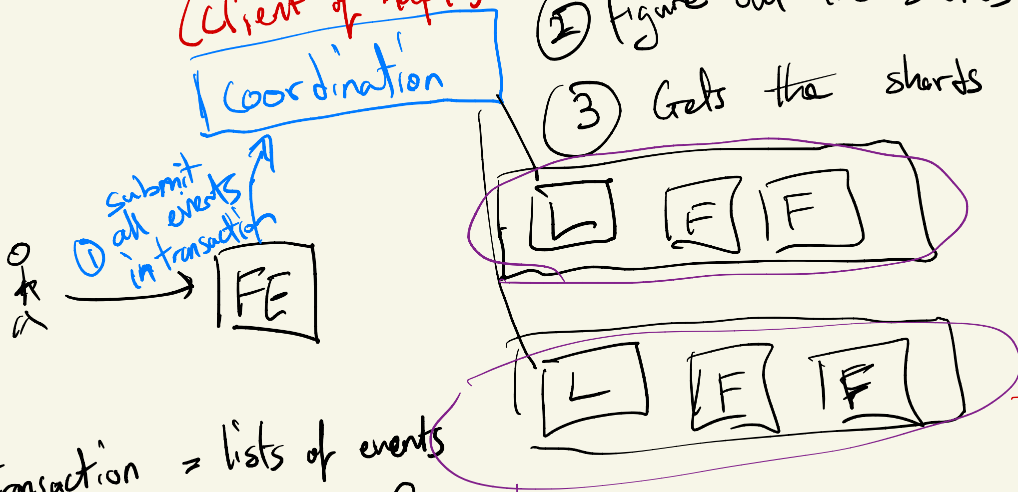
* changes from events last/persist after transaction(s)
* memory V. Disk
  * reset when server restarts
  * data on disk is persistent
* making sure things are stored to disk

Atomicity (1PC, 2PC, 3PC, 4PC)

(client of Raft)

② figure out the shards & leader for each shard

③ Gets the shards to execute events in the transaction



Transaction = lists of events to execute @ different shards

Participants are the leaders of each shard involved in a transaction

why would requests fail?

(why will a request carrying an event to a leader fail?)

* leader failures → during election system will not be able to process events

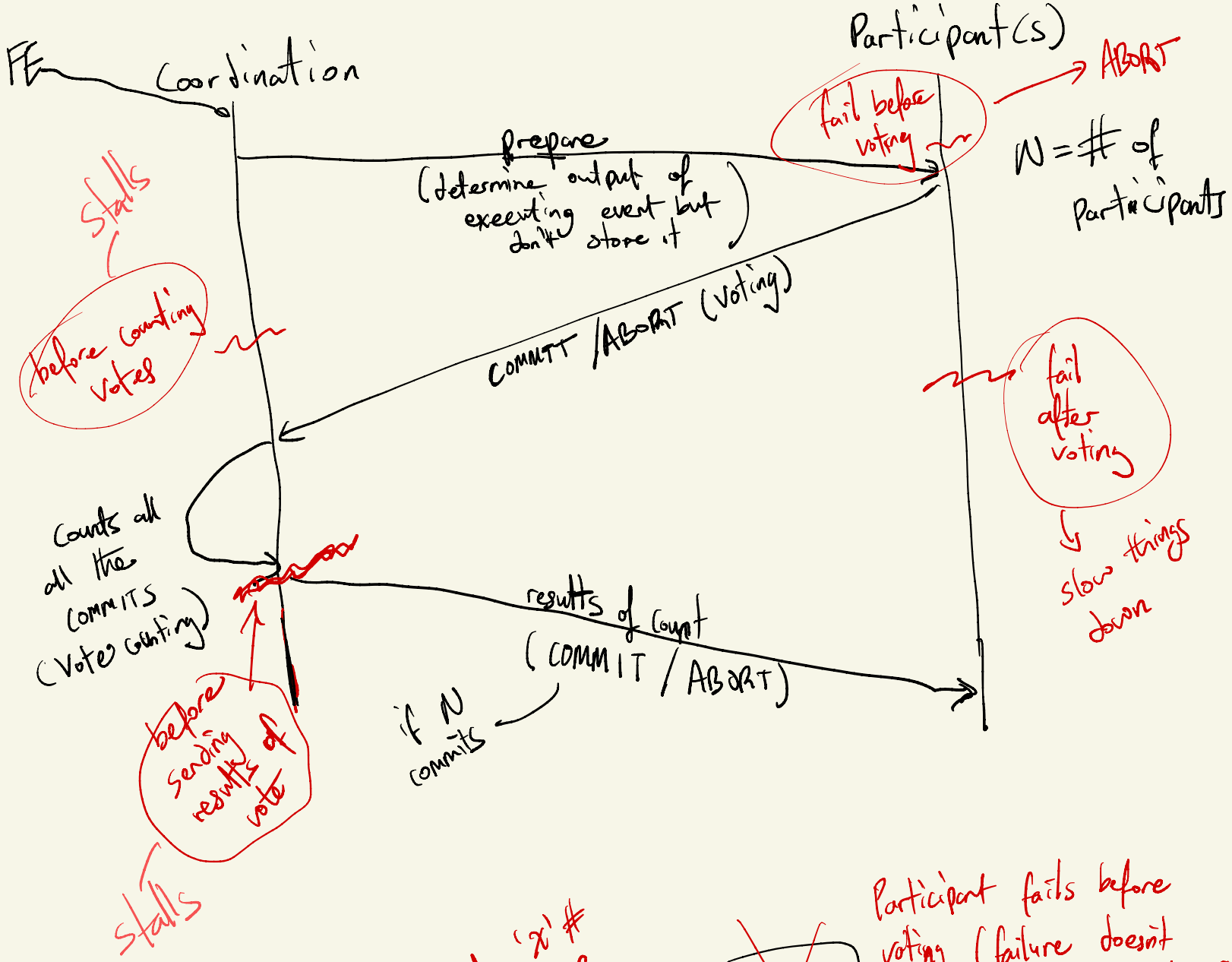
* network partition → if the coordination is in the minority partition then no events can be processed

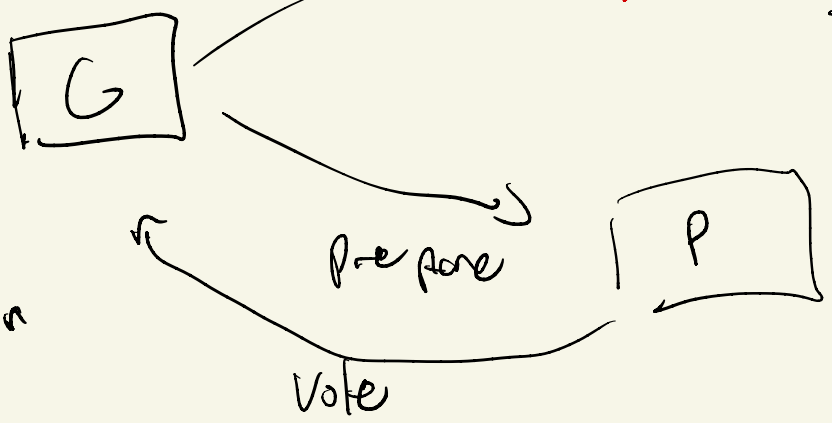
Two Phase Commit:

transaction succeeds (commits) → if all participants (leaders) commit their events

transaction fails (ABORTS) → if at least one participant fails (abort)

2PC (two phase commit)





Participant fails before voting (failure doesn't recover after x retries)

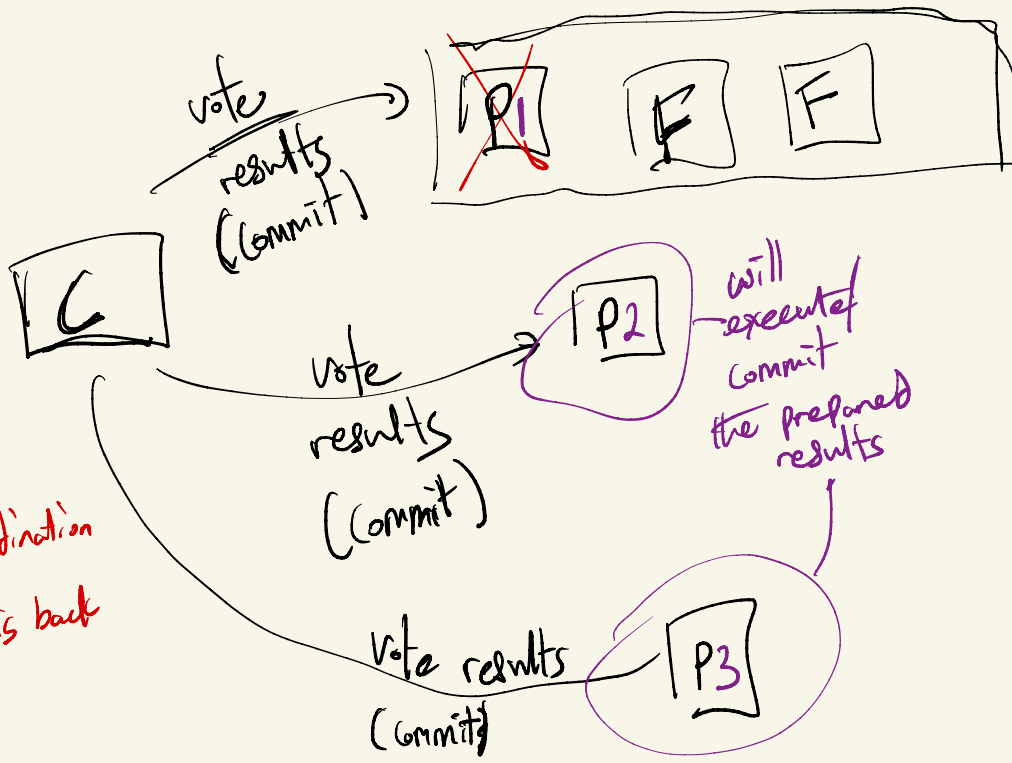
Participants that is unable to get prepare will not participate & thus the events on that shard are not going to happen

In this failure coordinator automatically aborts the transaction

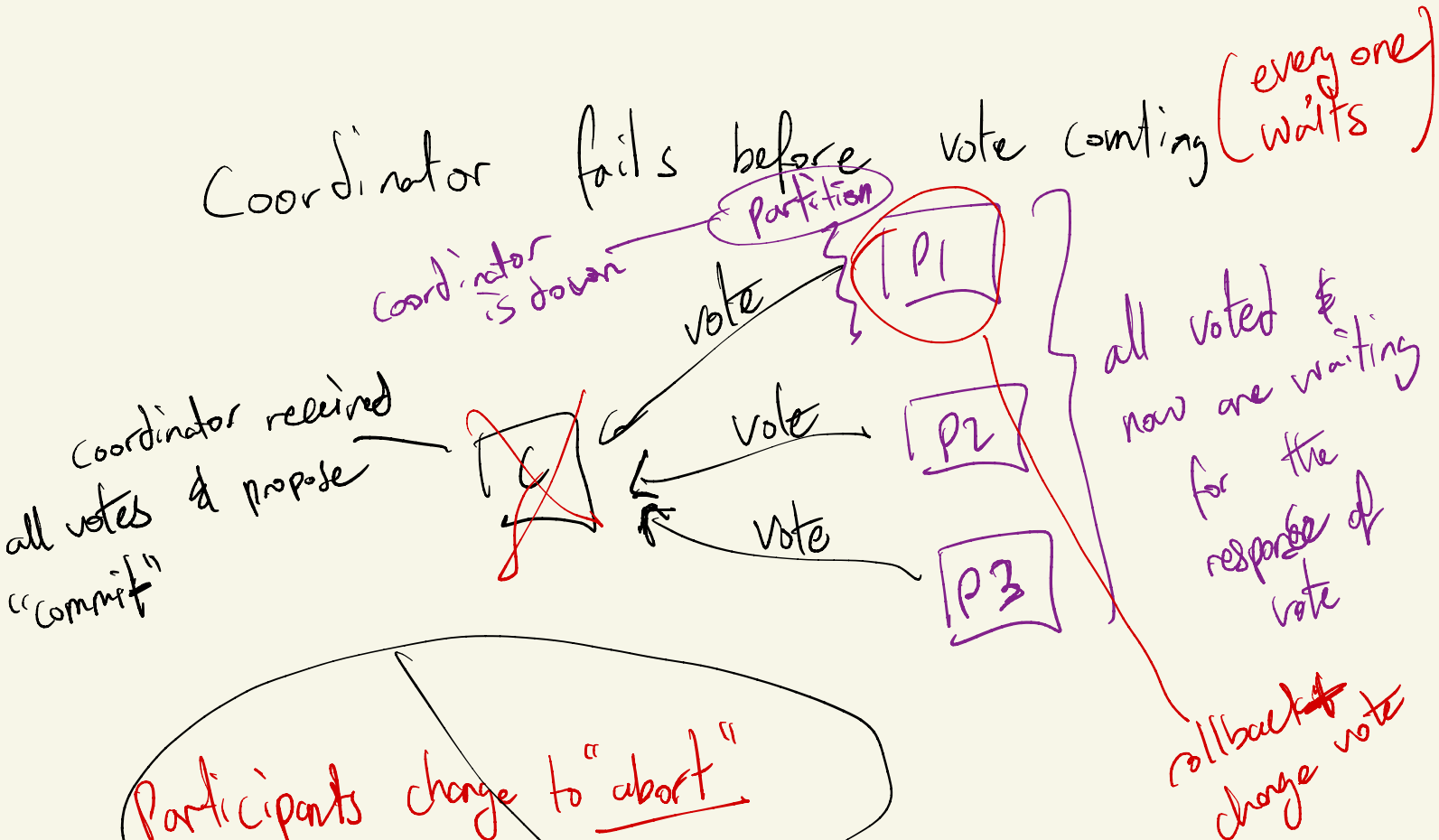
Participant fails after voting

① try to do a rollback
but eventually participant will come back up with "prepared" information still store

② Instead of rollback, the coordination will wait / retry until P1 is back up



Coordinator fails before vote counting (every one waits)



2PC does not allow participants to change votes

Coordinator

fail — stalls the transaction & prevents forward progress toward termination

after sending prepare but before sending out votes

→ no progress can be made

Participants

failure before voting → then abort everything

if participant fails after voting → then just retry until participant acknowledges receiving results

Today

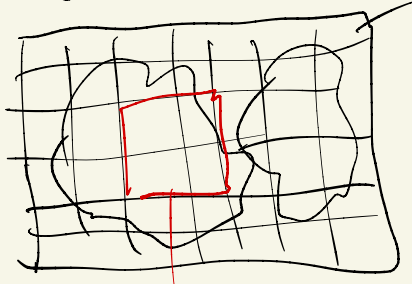
1. ACID semantics
2. differences btw dist transaction & replication strategies
3. Atomicity ⇒ 2PC

Next Class

4. Isolation
5. Distributed file systems

When to used dist Transactions?

① money transfer btw users

② 

③ online network place closely interacting users in same shard so that shard interactions are low