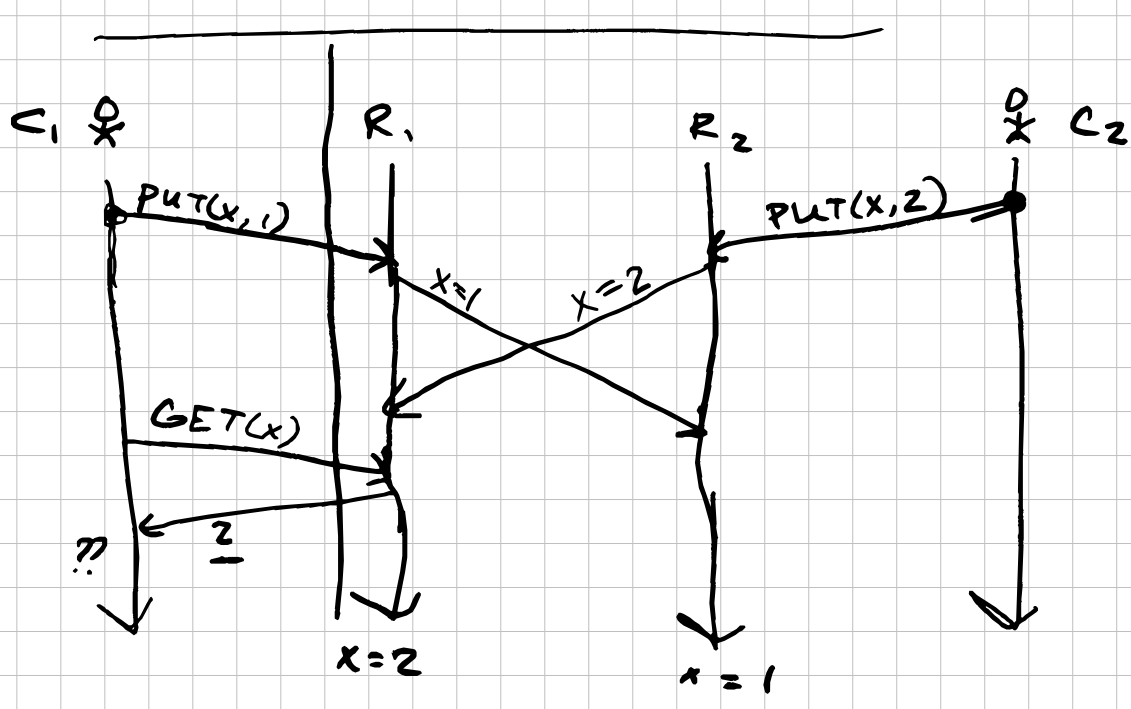
# CSE138 Lecture 10

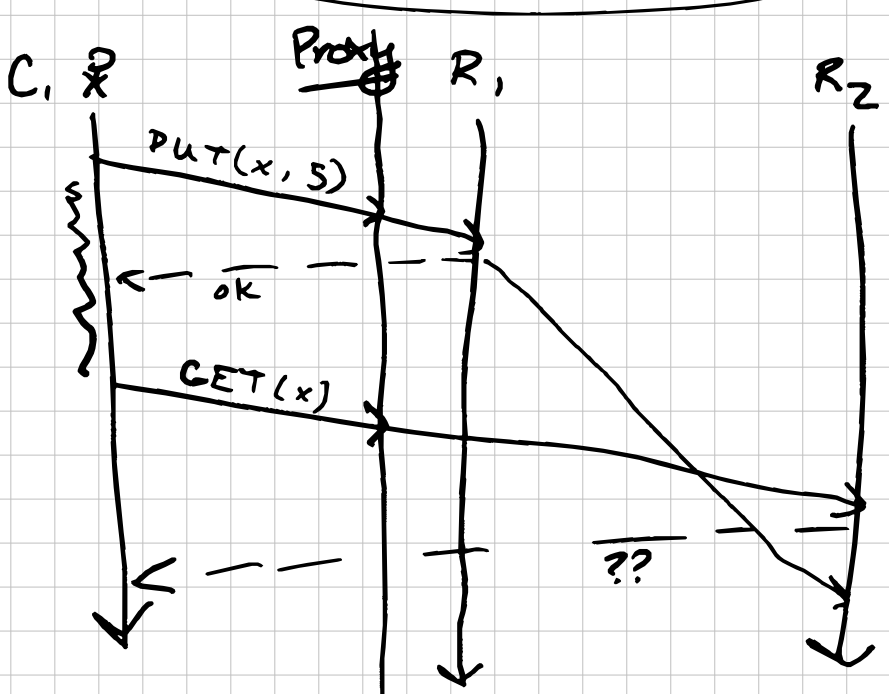
this time:

✓ - exam review
- recap: strong consistency
- other (weaker) consistency policies:
  - a small sampling: { - causal consistency; - FIFO consistency; - "read-your-writes" consistency. }
  - if time: { - mechanisms for implementing strong consistency: - primary-backup; - chain replication; - latency and throughput }

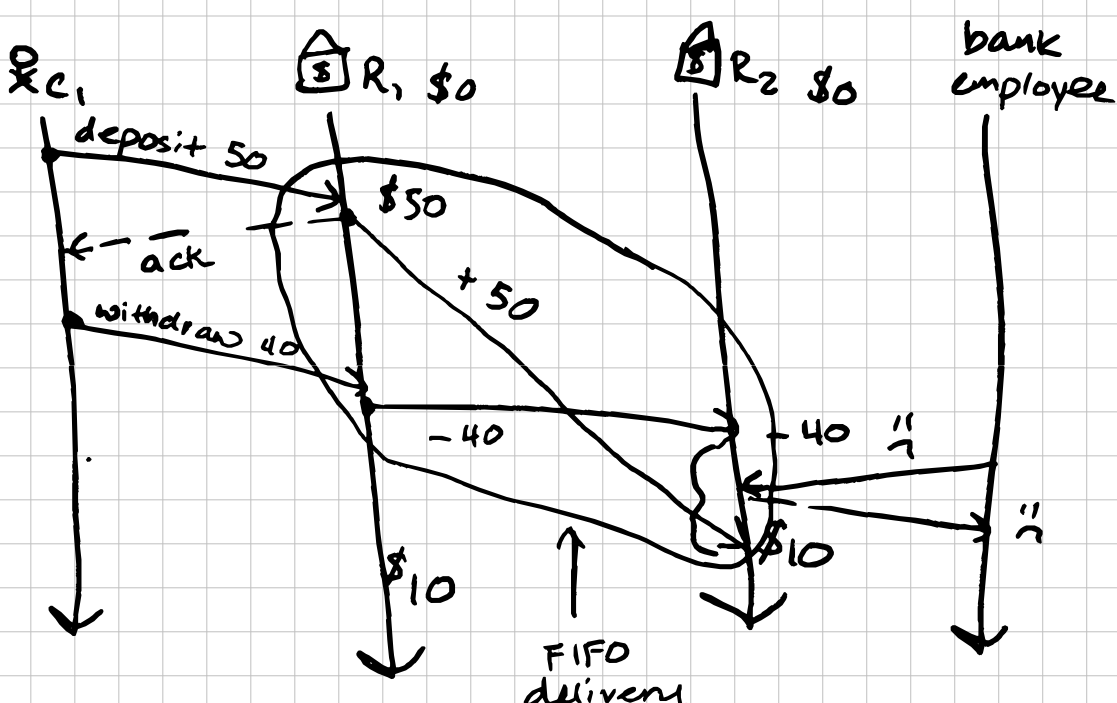



violation of totally-ordered message delivery leads to...
violation of strong consistency.

informally: clients can't tell the data is replicated.




violation of RYW (read your writes)

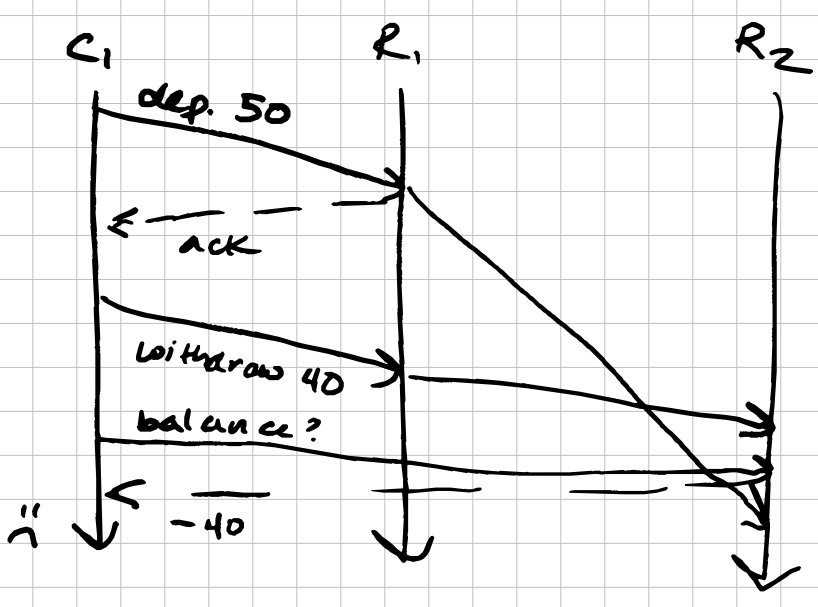


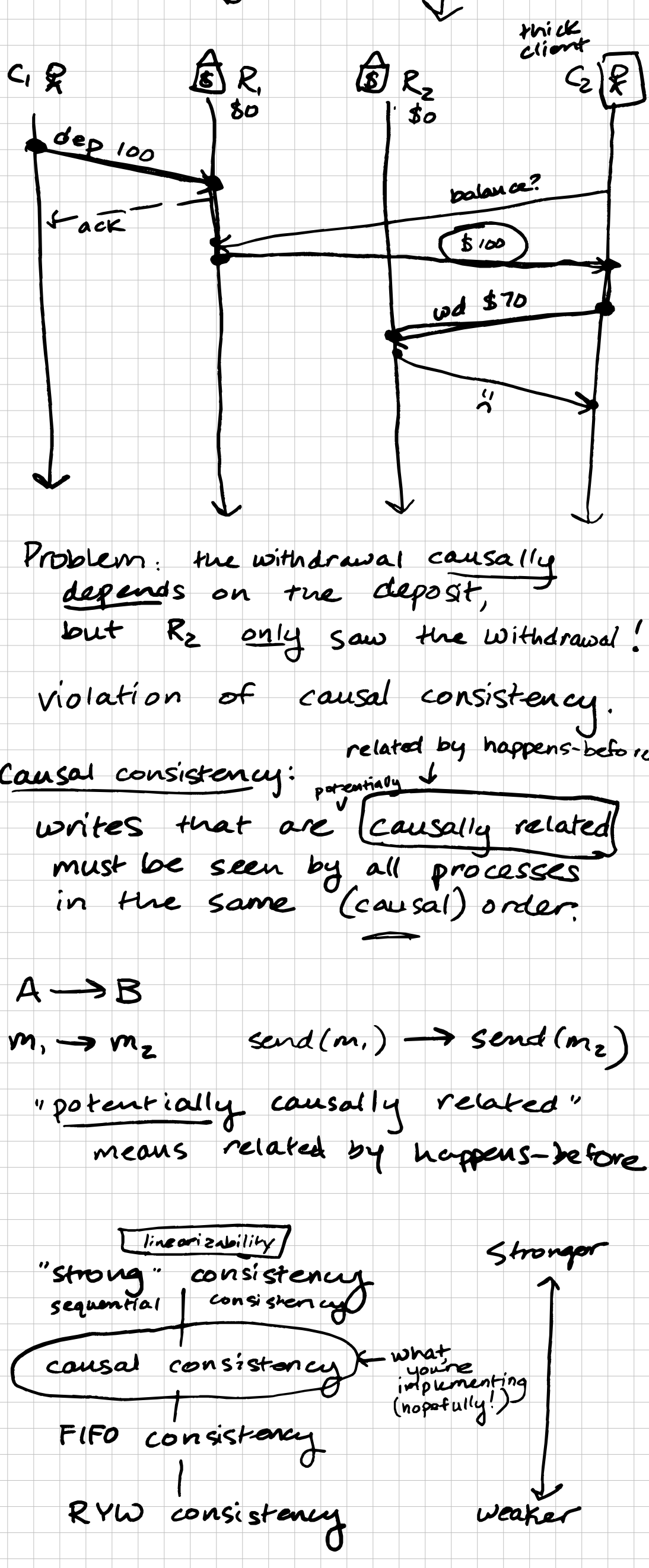

violation of FIFO consistency

FIFO consistency:

writes issued by a single process are seen by all processes in the order they were issued.

(A message would need to be delivered to be "seen")




Problem: the withdrawal causally depends on the deposit, but $R_2$ only saw the withdrawal!

violation of causal consistency.

Causal consistency: (potentially) related by happens-before

writes that are causally related must be seen by all processes in the same (causal) order.

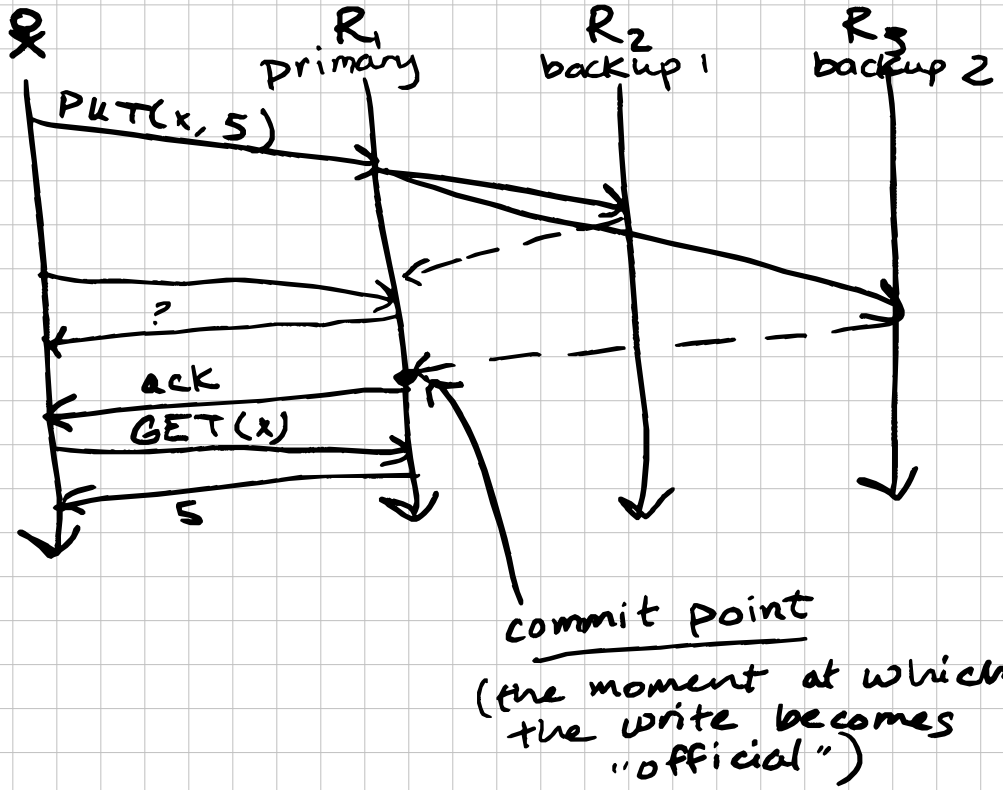
$A \rightarrow B$

$m_1 \rightarrow m_2$ $\quad send(m_1) \rightarrow send(m_2)$

"potentially causally related" means related by happens-before

# Mechanisms for implementing strong consistency.

## primary-backup replication




Pros and Cons:

- Strong consistency! (pro)
- primary is a bottleneck! (con)

Replication should ideally give you

- ☑ fault tolerance
- ☒ dividing up the work ("scalability")
- ☒ data locality

primary-backup replication is only 1 for 3.