Distributed Systems
CS6421
Consistency and Replication

Prof. Tim Wood and Prof. Roozbeh Haghnazar

Includes material adapted from Van Steen and Tanenbaum's Distributed Systems book

Final Project

Questions?

- Design Document
  - Proposed Design
  - UML Diagrams describing architecture and communication
  - Work timeline with breakdown by team member
- Schedule meetings with us!

- Timeline
  - Milestone 0: Form a Team - 10/12
  - Milestone 1: Select a Topic - 10/19
  - Milestone 2: Literature Survey - 10/29
  - **Milestone 3: Design Document - 11/5**
  - Milestone 4: Final Presentation - 12/14

https://gwdistsys20.github.io/project/

Last Time…

- Fault Tolerance
  - Types of Failures
  - Two Generals Problem
  - Fault Tolerance Algorithms
  - Centralized FT: Raft/Paxos

This Time…

- Replication and Consistency
  - Why replicate
  - What is consistency?
  - Consistency Models
  - Quorum Replication

Next Time: Exam!

DistSys Challenges

- **Heterogeneity**
- Openness
- **Security**
- **Failure Handling**
- Concurrency
- **Quality of Service**
- **Scalability**
- **Transparency**

Any questions about these? You will need to relate your project to them and they will be on the exam!

Problem

- Given that synchronization and locking is so difficult, do we really need it in a distributed system?
- Is there a better way?

Reasons for Replication

- Data are replicated to increase the reliability of a system.
- Replication for performance
  - Scaling in numbers
  - Scaling in geographical area
- Caveat
  - Gain in performance
  - Cost of increased bandwidth for maintaining replication

Reasons for Replication

- Reliability.
- Performance.

- Replication is the solution.

How do we keep them up-to-date?
How do we keep them consistent?

More on Replication

- Replicas allows remote sites to continue working in the event of local failures.
- It is also possible to protect against data corruption.
- Replicas allow data to reside close to where it is used.
- This directly supports the distributed systems goal of enhanced *scalability*.
- Even a large number of replicated "local" systems can improve performance: think of clusters.

- So, what's the catch?

- It is not easy to keep all those replicas *consistent*.

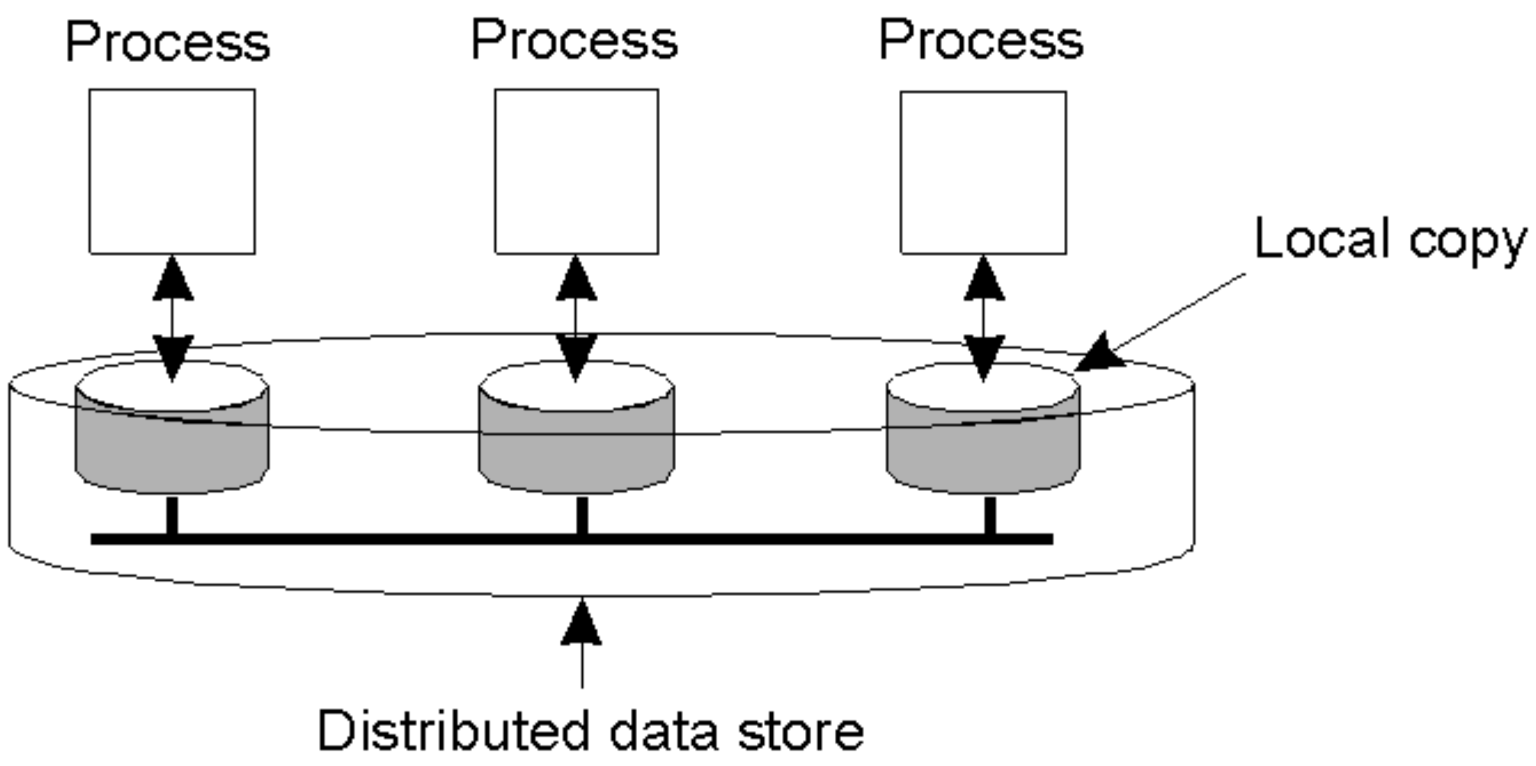
CONSISTENCY MODELS

- What is a consistency model?
  - It is an agreement and contract between a distributed data store and related processes.

- Data-Centric
  - Continuous
  - Consistent ordering of operation
    - Sequential
    - Causal
- Client-Centric

DATA-CENTRIC CONSISTENCY MODELS

- A data-store can be read from or written to by any process in a distributed system.
- A local copy of the data-store (replica) can support "fast reads".
- However, a write to a local replica needs to be propagated to *all* remote replicas.



CONTINUOUS CONSISTENCY

- There are different ways for applications to specify what inconsistencies they can tolerate.
- Yu and Vahdat [2002] take a general approach by distinguishing three independent axes for defining inconsistencies:
  - deviation in numerical **values** between replicas
  - deviation in **staleness** between replicas
  - deviation with respect to the **ordering** of update operations
- They refer to these deviations as forming ***continuous consistency*** ranges.

Example of numerical deviations

Company	Stock price
Apple	$1100.00

+0.02
+0.03

Company	Stock price
Apple	$1100.00

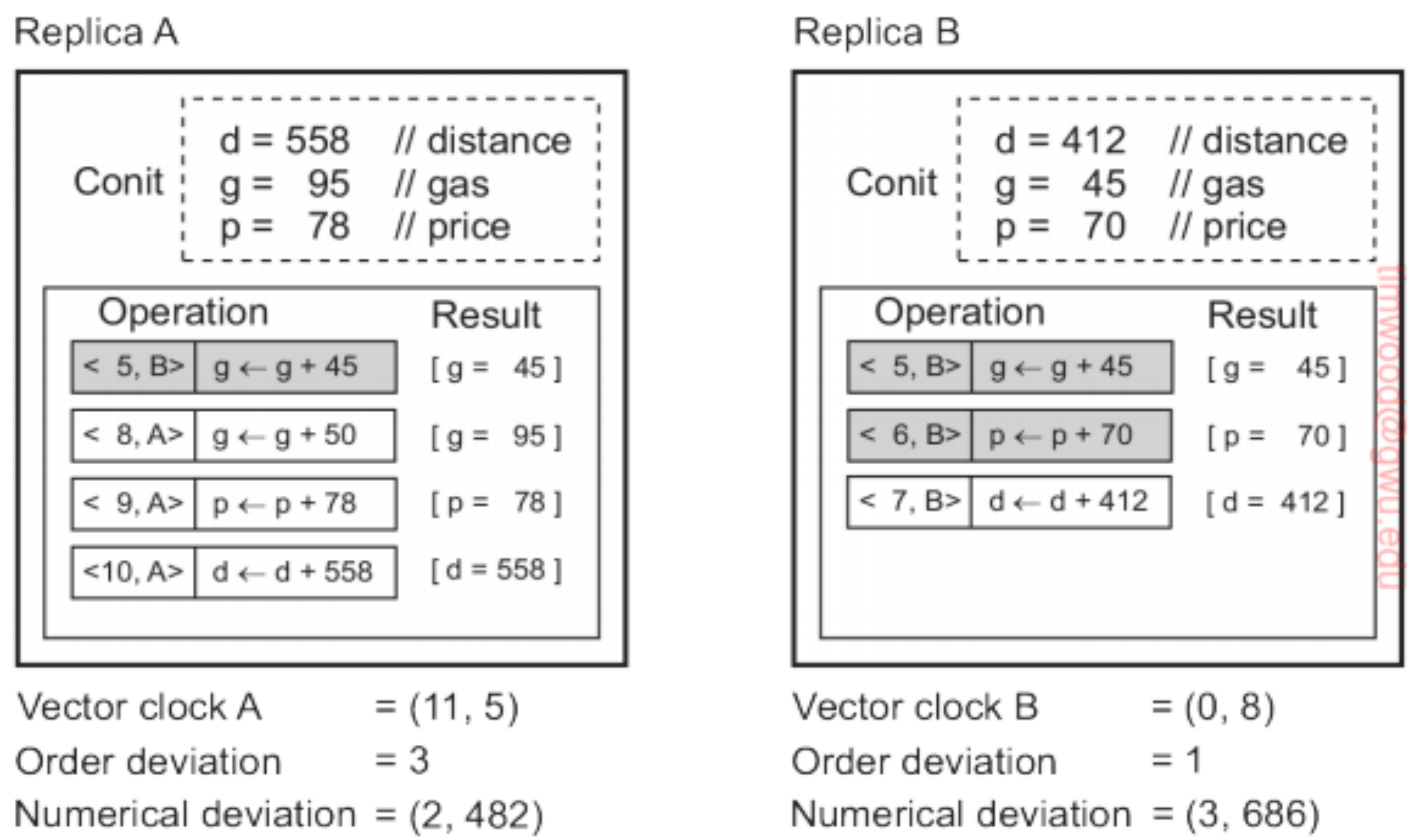
+0.03

Company	Stock price
Apple	$1100.00

+0.03

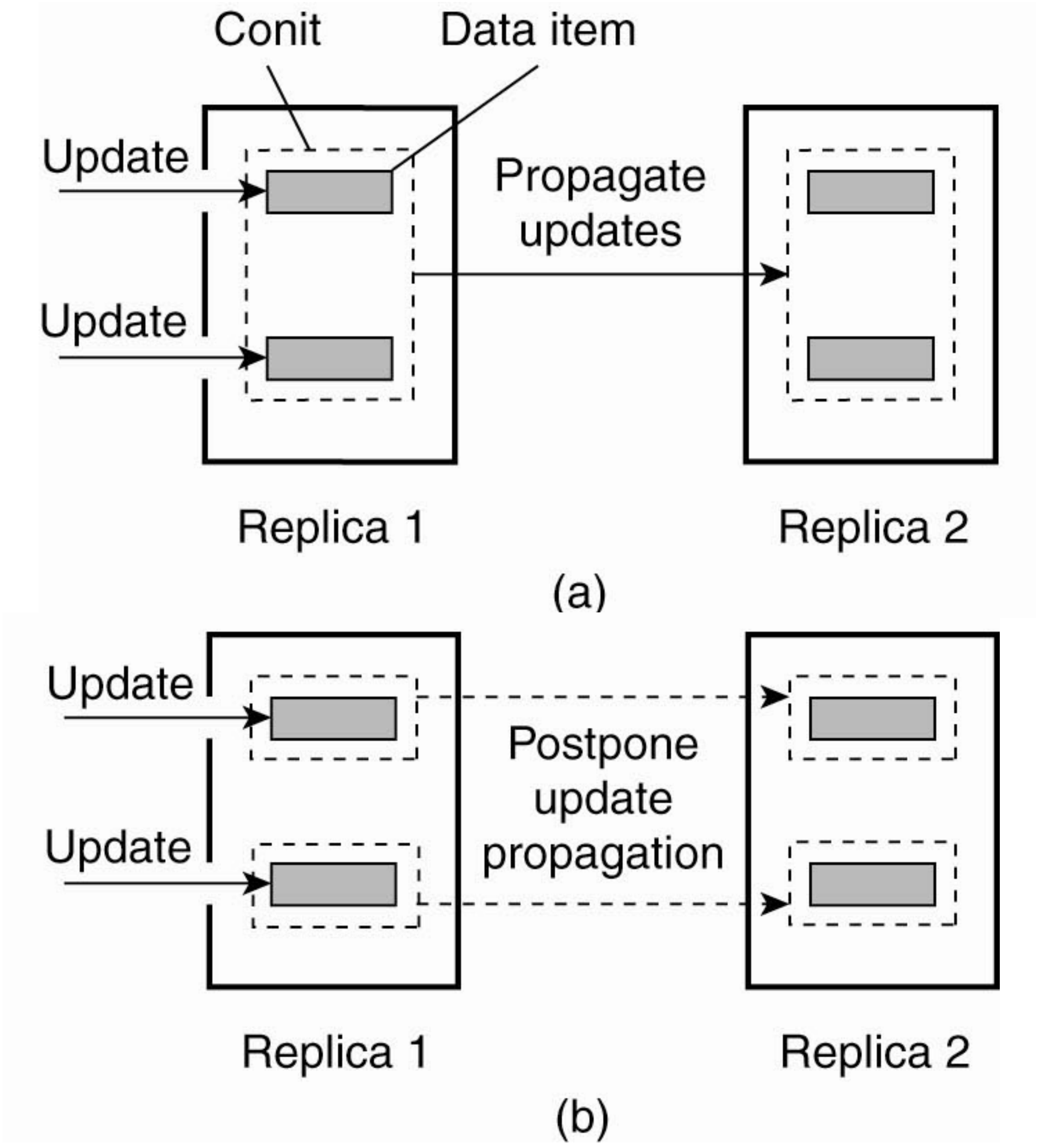
CONTINUOUS CONSISTENCY

- Each replica server maintains a two-dimensional vector clock



Continuous Consistency

- Choosing the appropriate granularity for a conit.
  - (a) Two updates lead to update propagation.
  - (b) No update propagation is needed



Consistent ordering of operations

- Sequential consistency
- Causal consistency
- Grouping operations

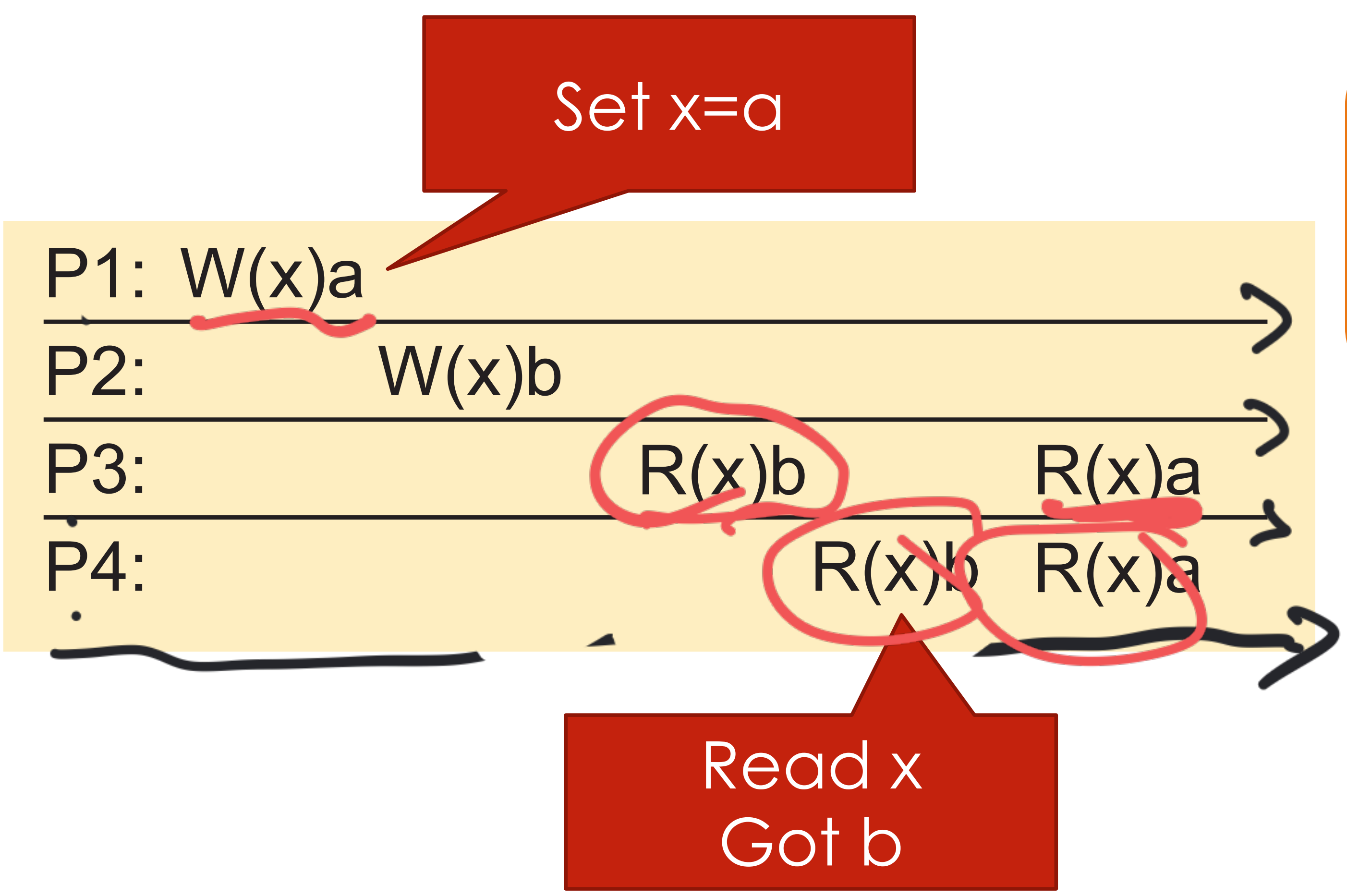
Prof. Tim Wood & Prof. Roozbeh Haghnazar

Consistency Models

CONSISTENCY VERSUS COHERENCE

- A consistency model describes what can be expected when multiple processes concurrently operate on a set of data. The set is then said to be consistent if it adheres to the rules described by the model.
- Where data consistency is concerned with a set of data items, coherence models describe what can be expected to hold for only a single data item [Cantin et al., 2005].
- In this case, we assume that a data item is replicated; it is said to be coherent when the various copies abide to the rules as defined by its associated consistency model.

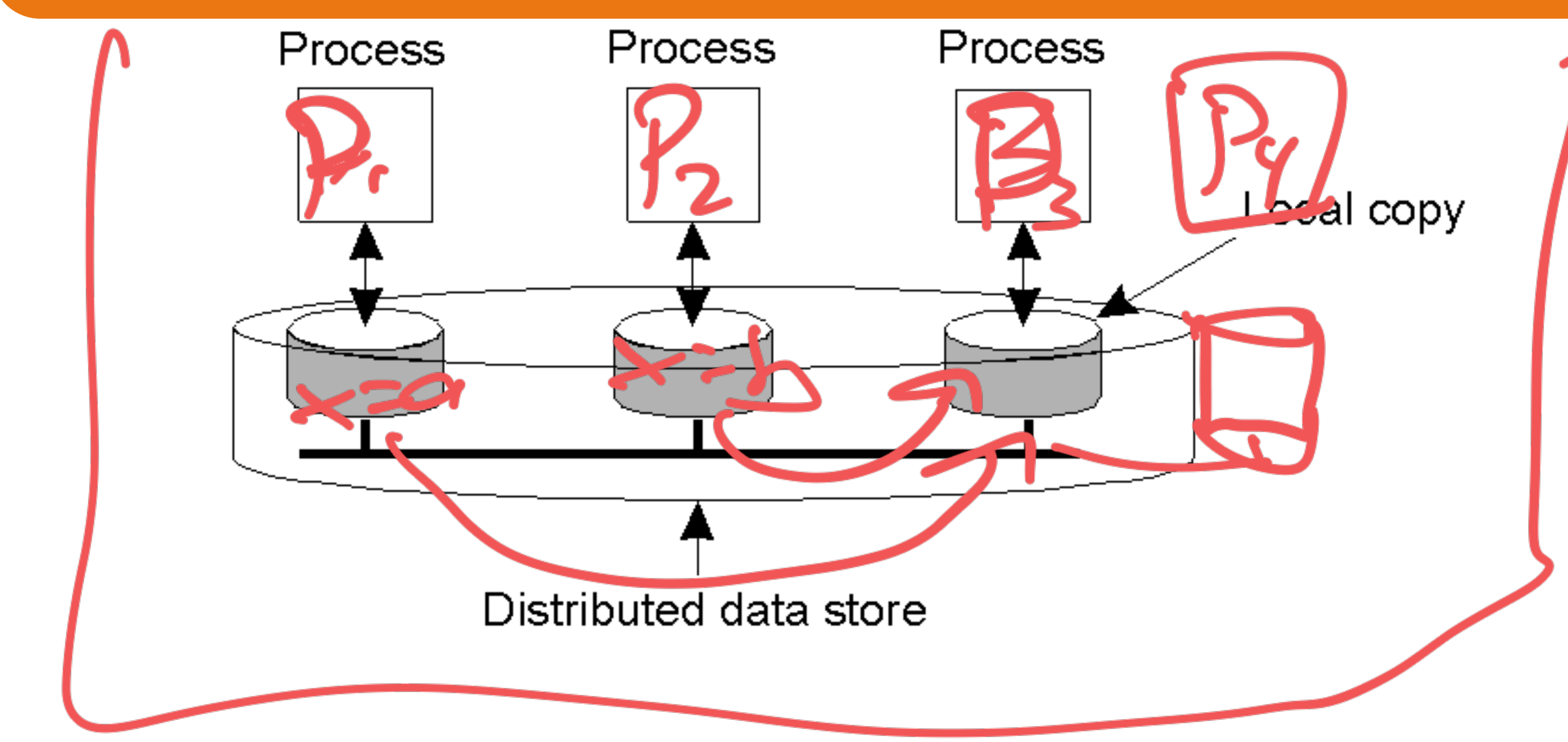
Consistency Model Diagram Notation



$W_i(x)a$ – a write by process 'i' to item 'x' with a value of 'a'. That is, 'x' is set to 'a'.

$R_i(x)b$ – a read by process 'i' from item 'x' producing the value 'b'. That is, reading 'x' returns 'b'.

Time moves from left to right in all diagrams.



Sequential Consistency

- The result of any execution is the same as if the operations of **all processes** were executed in some sequential order, and
- The operations of each individual process appear in this sequence in the order specified by its program.

Any ordering of reads/writes is fine, but all processes must see the same ordering

P1:	W(x)a			
P2:		W(x)b		
P3:	R(x)a		R(x)b	
P4:			R(x)a	R(x)b

P1:	W(x)a			
P2:		W(x)b		
P3:			R(x)b	R(x)a
P4:			R(x)b	R(x)a

P1:	W(x)a			
P2:		W(x)b		
P3:			R(x)b	R(x)a
P4:			R(x)a	R(x)b

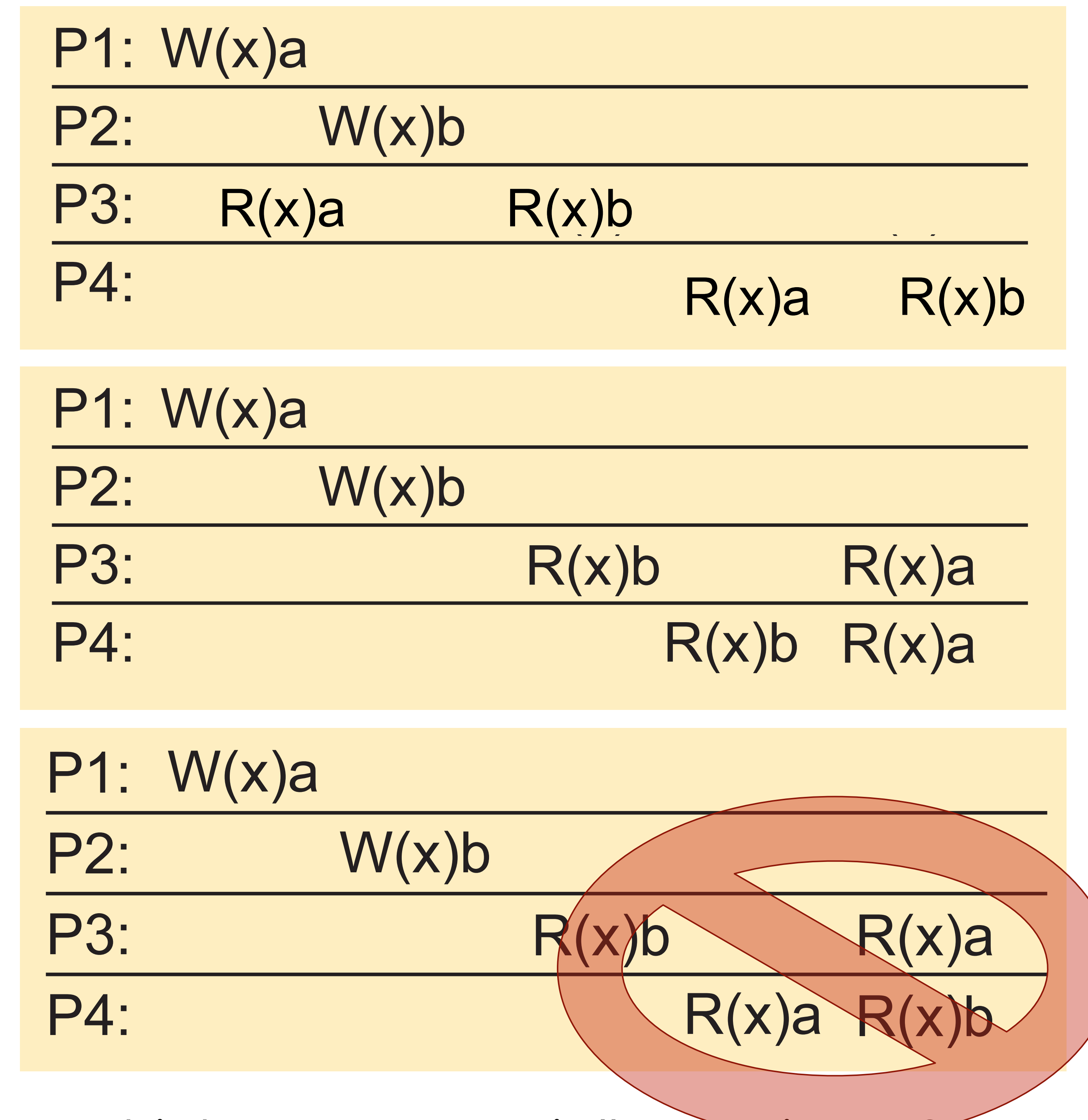
Which are sequentially consistent?

conflict

order for B → C → D → A → E → F

Sequential Consistency

- The result of any execution is the same as if the operations of **all processes** were executed in some sequential order, and
- The operations of each individual process appear in this sequence in the order specified by its program.



Which are sequentially consistent?

Any ordering of reads/writes is fine, but all processes must see the same ordering

if writes to the same variable are "connected" by a read

Causal Consistency

- Writes that are potentially causally related must be seen by all processes in the same order.
- Concurrent writes may be seen in a different order by different processes.

Reading a value means your future writes may be causally related to that operation!

P1:	W(x)a			
P2:		R(x)a	W(x)b	
P3:			R(x)b	R(x)a
P4:			R(x)a	R(x)b

P1:	W(x)a			
P2:		W(x)b		
P3:			R(x)b	R(x)a
P4:			R(x)a	R(x)b

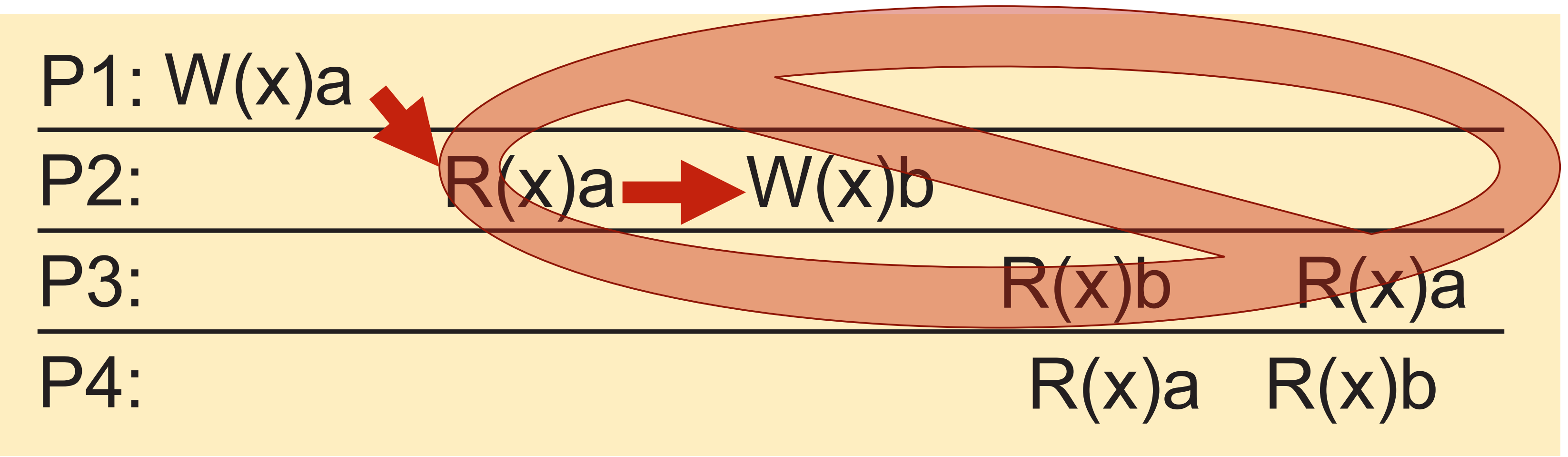
P1:	W(x)a				
P2:			W(x)b		R(x)c
P3:		W(x)c		R(x)b	R(x)a
P4:				R(x)a	R(x)b

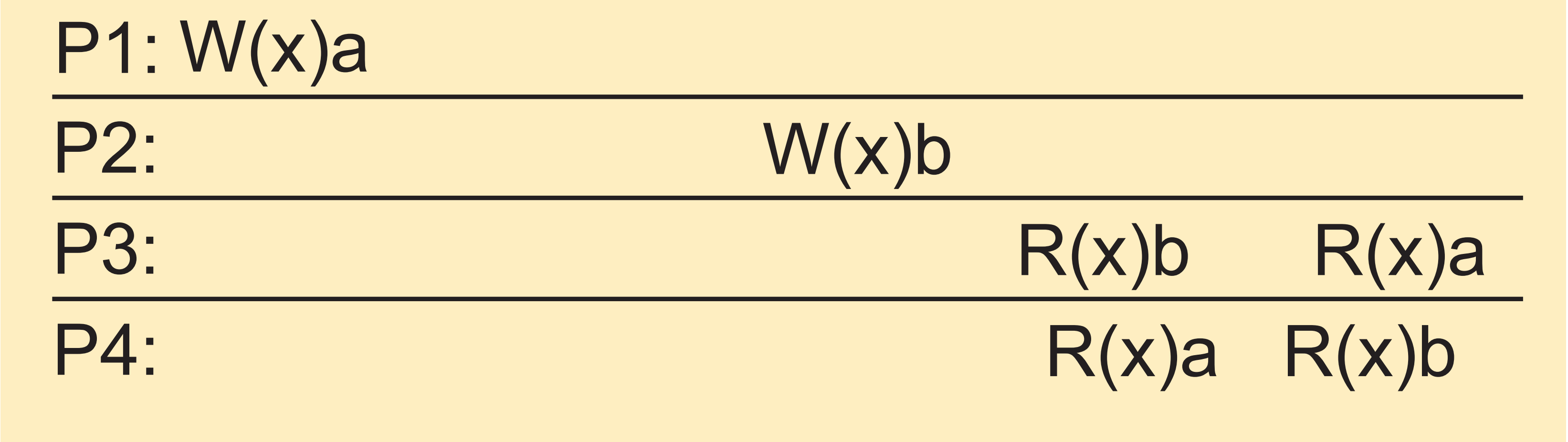
Prof. Tim Wood & Prof. Roozbeh Haghnazar

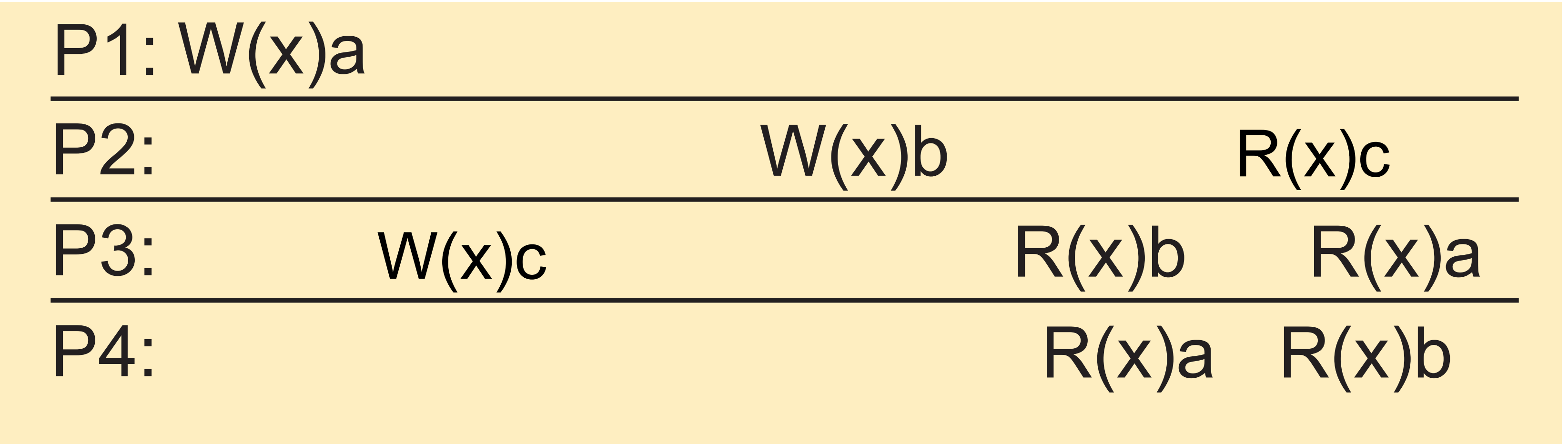
causal is less strict than seq

Causal Consistency

- Writes that are potentially causally related must be seen by all processes in the same order.
- Concurrent writes may be seen in a different order by different processes.



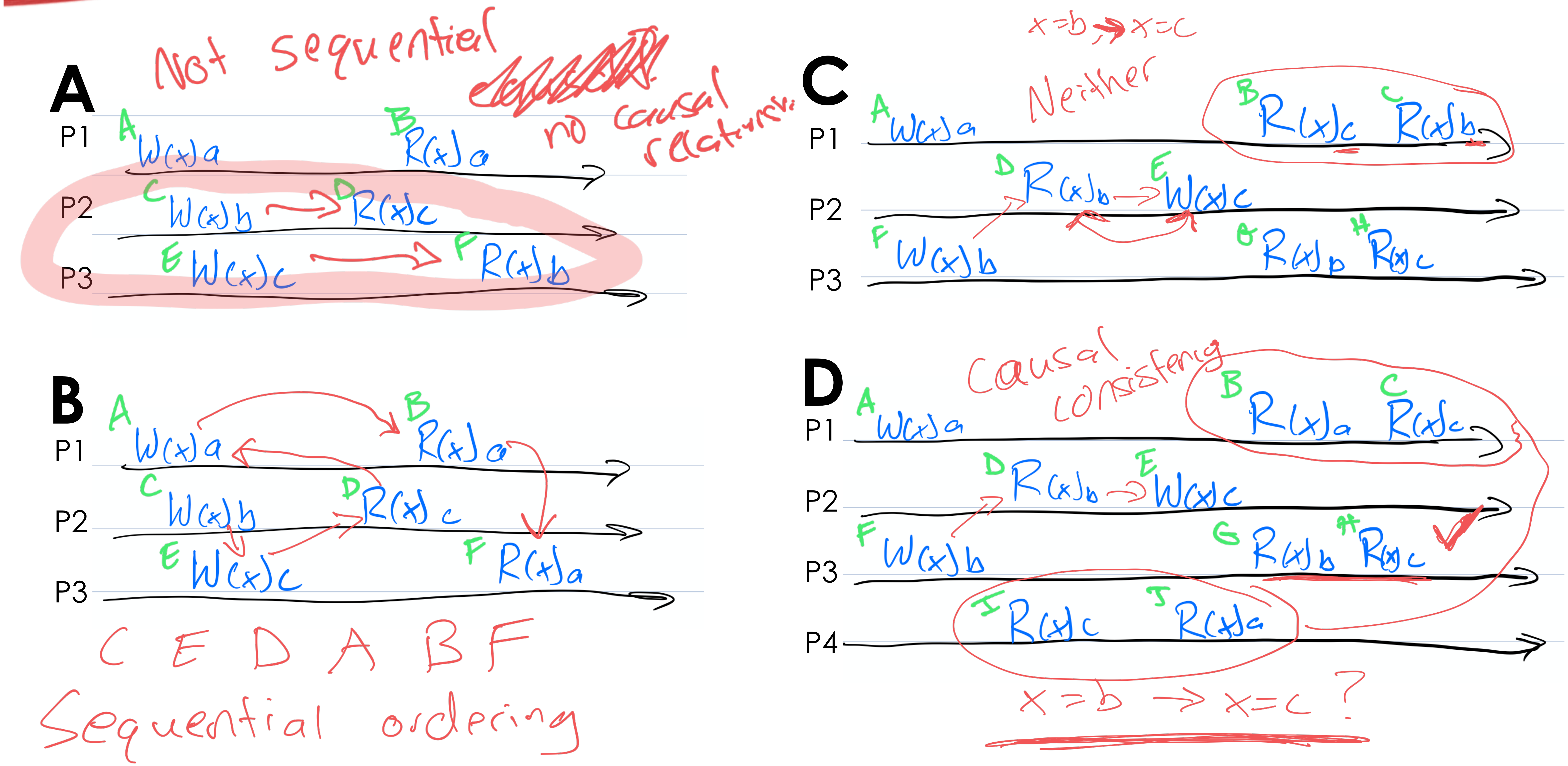




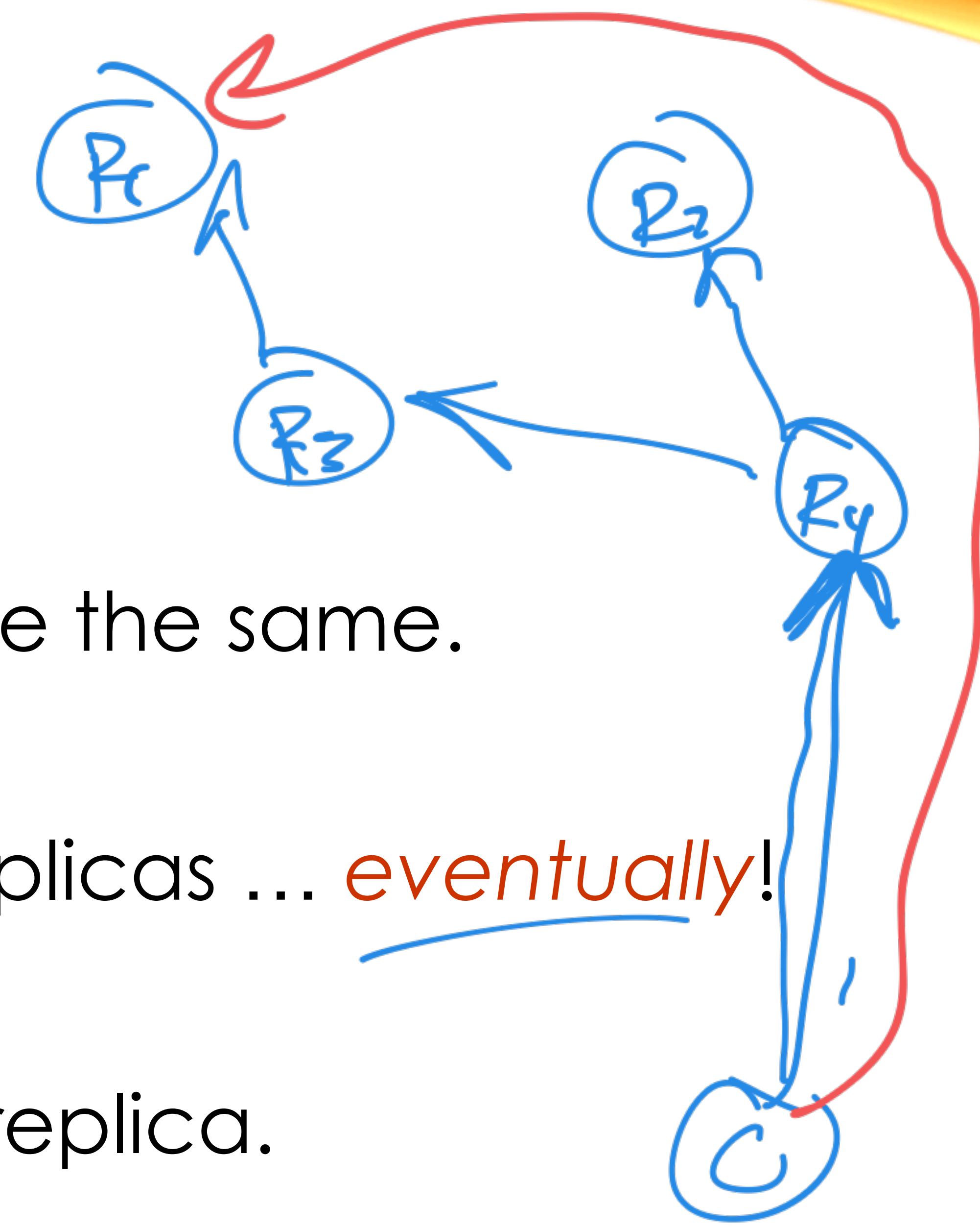
Reading a value means your future writes may be causally related to that operation!

Group Problems

Is each timeline **Sequential**, **Causal**, or **Neither**?



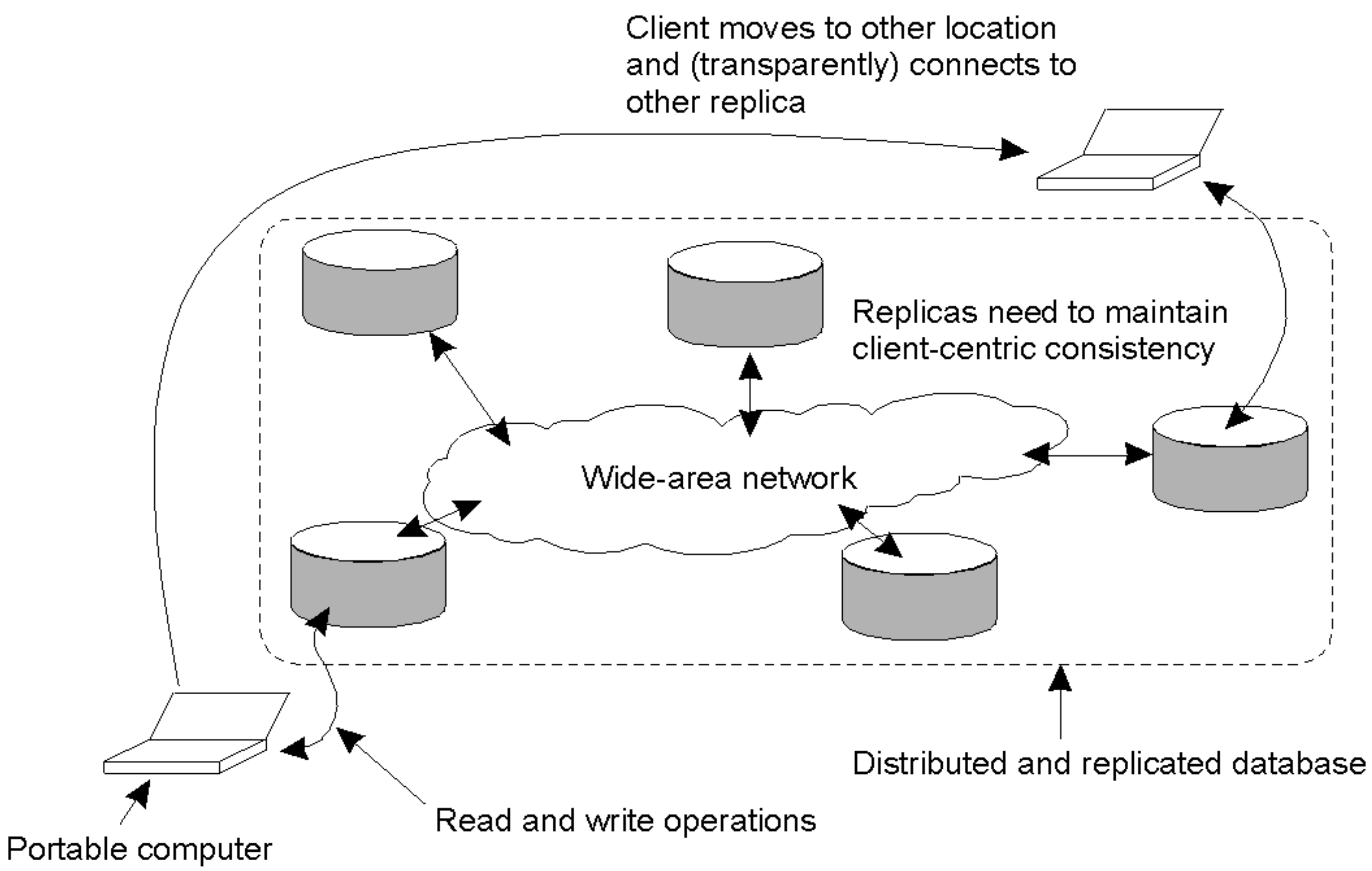
EVENTUAL CONSISTENCY



- The only requirement is that all replicas will *eventually* be the same.
- All updates must be guaranteed to propagate to all replicas … *eventually*!
- This works well if every client always updates the same replica.
- Things are a little difficult if the clients are *mobile*.

Eventual Consistency: Mobile Problems

- The principle of a mobile user accessing different replicas of a distributed database.
- When the system can guarantee that a single client sees accesses to the data-store in a consistent way, we then say that "client-centric consistency" holds.



Prof. Tim Wood & Prof. Roozbeh Haghnazar

Exam

Exam Details – 11/12

~60 minutes

- Exam will be during class Thursday November 12, 6:10-8:40PM
- Exam will be on Blackboard
- Exam will be open book and open notes as follows:
  - The Van Steen & Tanenbaum book
  - The slides presented in class
  - Any notes you wrote (typed/handwritten)
- You may **NOT** use any external websites
- You may **NOT** communicate with any other students/people outside class
- Sample questions are on website

my website

any material covered in class up to and including tonight