TECH VAULT

CONCURRENCY CONTROL

- Timestamp Ordering
- Optimistic CC

AMR ELHELW

Timestamp Ordering (TO)

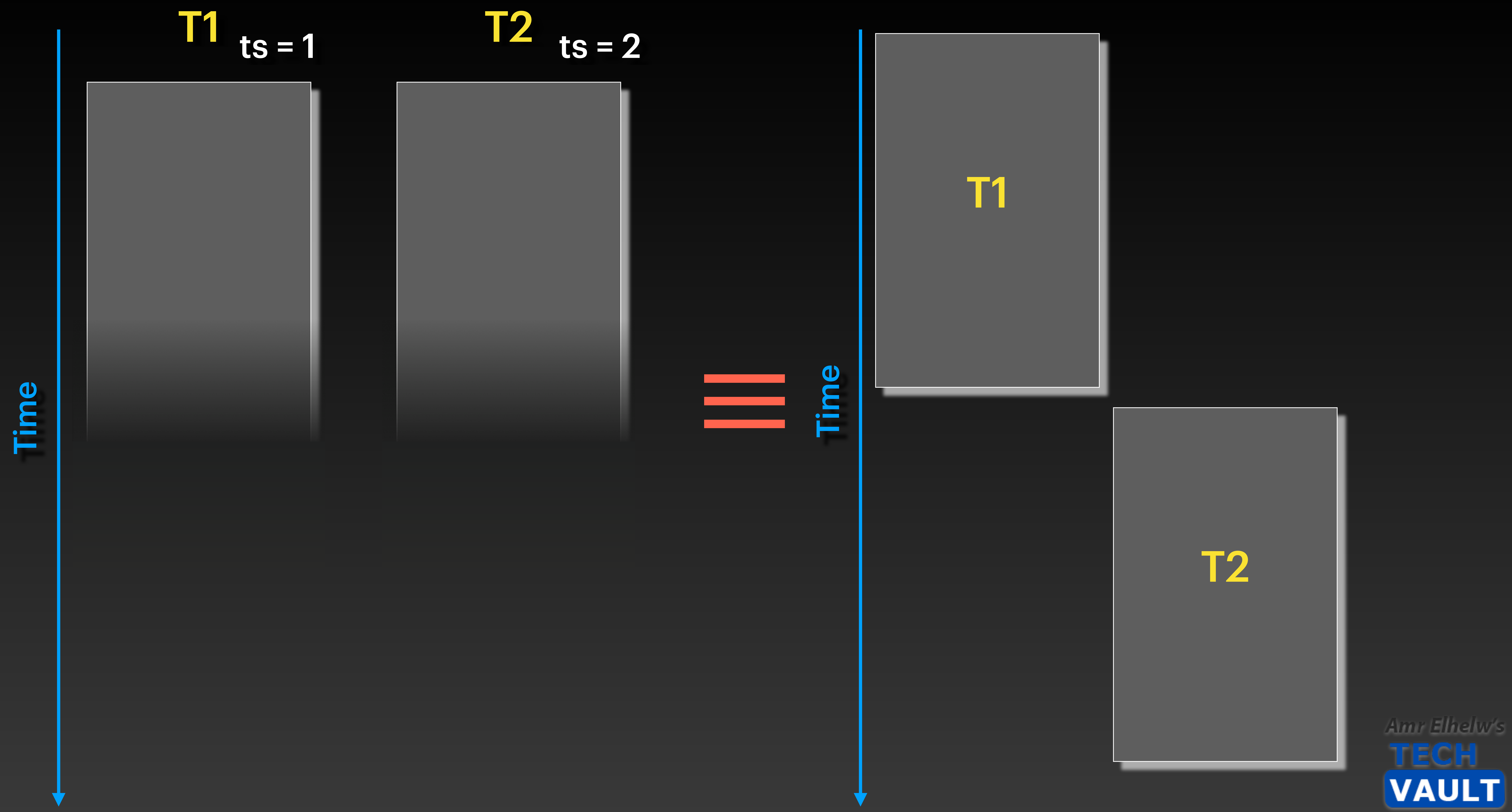
- Each transaction is assigned a timestamp *ts*
- Timestamps are monotonically increasing
- If $ts(T_i) < ts(T_j)$ then we want a schedule equivalent to the serial schedule where T_i runs before T_j

Amr Elhelw's TECH VAULT

Timestamp Ordering (TO)

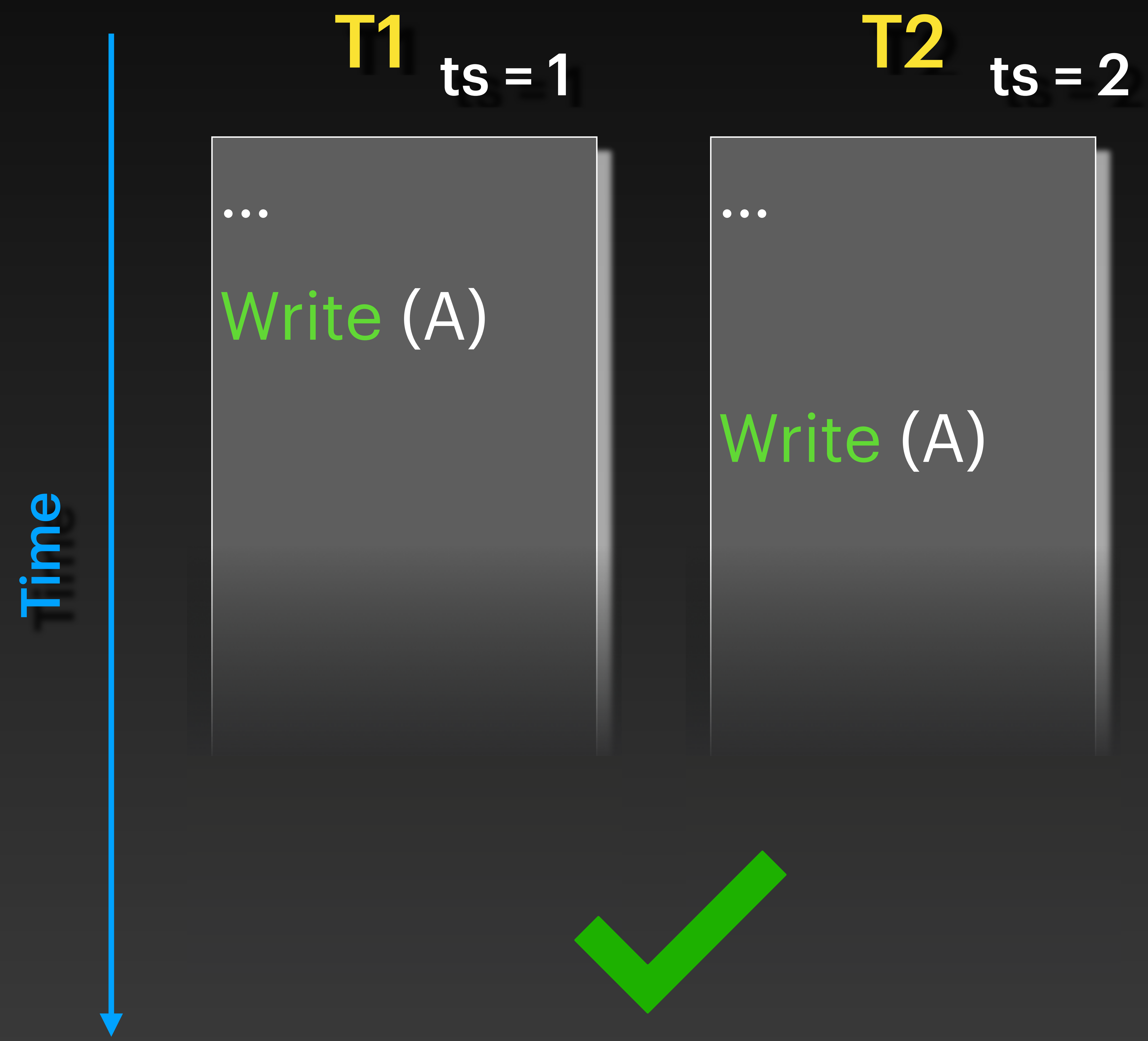
- How are timestamps assigned?
  - System/Wall Clock
  - Incremental Counter
  - …

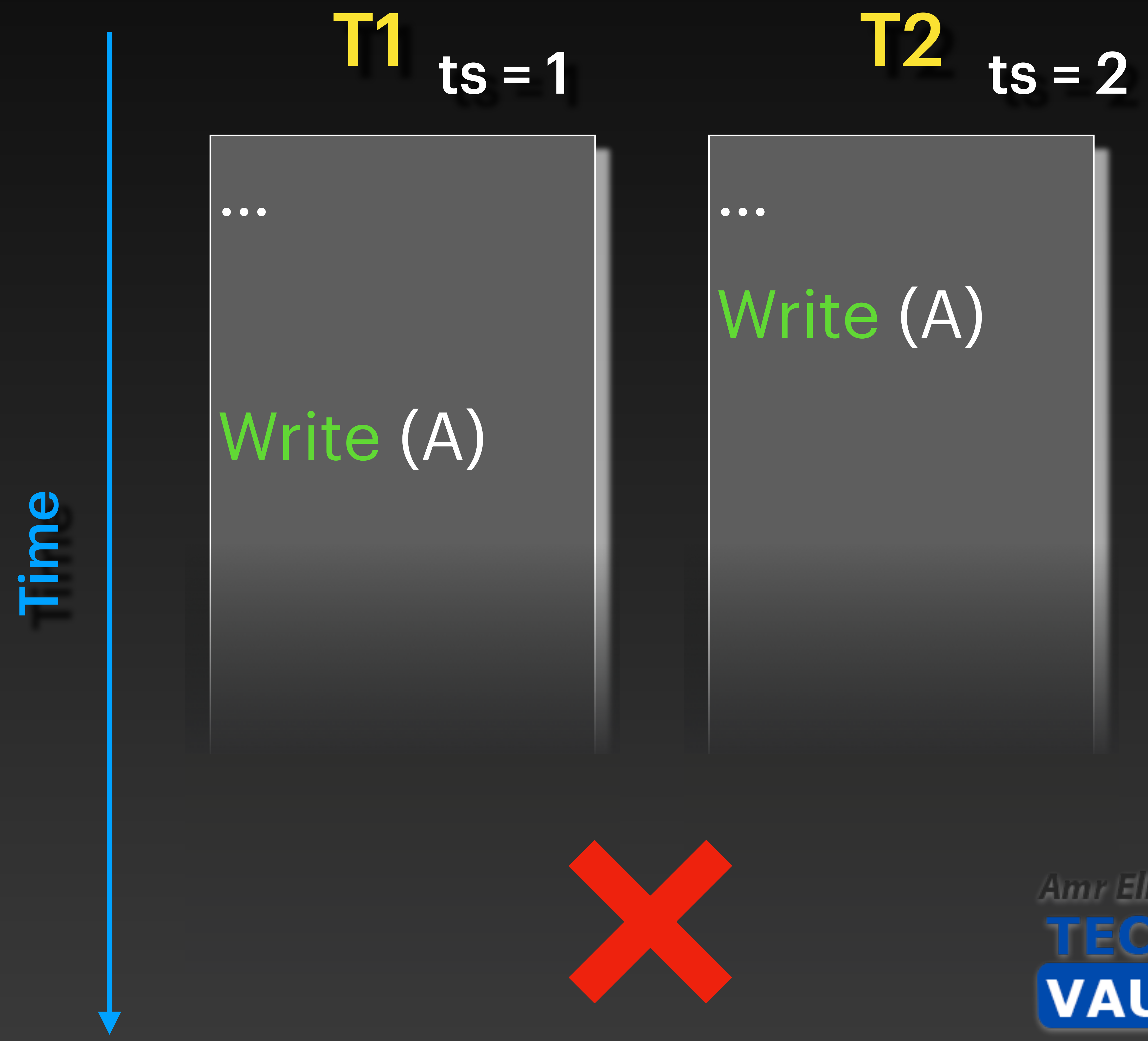
Amr Elhelw's TECH VAULT


T1
ts = 1
T2
ts = 2
Time
Time
T1
T2
Amr Elhelw's
TECH
VAULT


T1 ts = 1
T2 ts = 2
...
Write (A)
...
Read (A)
Time
Time
T1
T2
Amr Elhelw's
TECH
VAULT


T1 ts = 1
T2 ts = 2
...
Read (A)
...
Write (A)
Time
Time
T1
T2
Amr Elhelw's
TECH
VAULT





Amr Elhelw's TECH VAULT

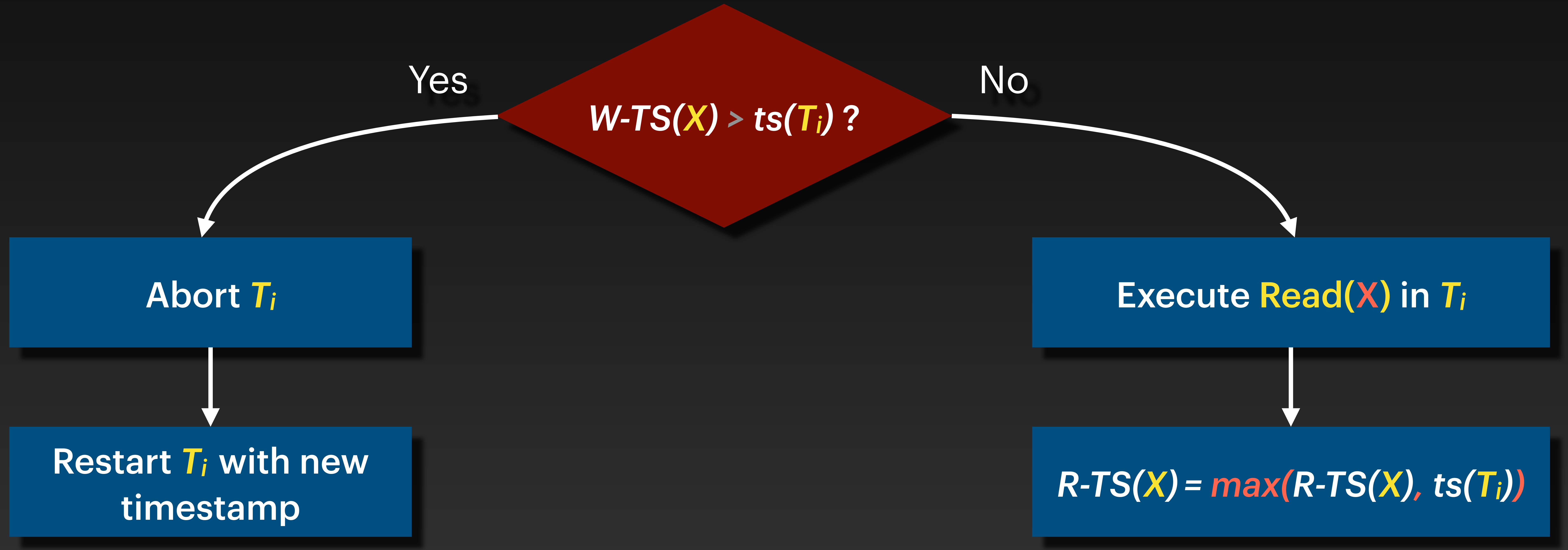
Timestamp Ordering (TO)

- No locks
- Each object (row, table, etc.) is tagged with:
  - *Read timestamp* R-TS
    - Largest timestamp of a transaction that read object
  - *Write timestamp* W-TS
    - Largest timestamp of a transaction that wrote object

Amr Elhelw's
TECH
VAULT

Timestamp Ordering: Reads

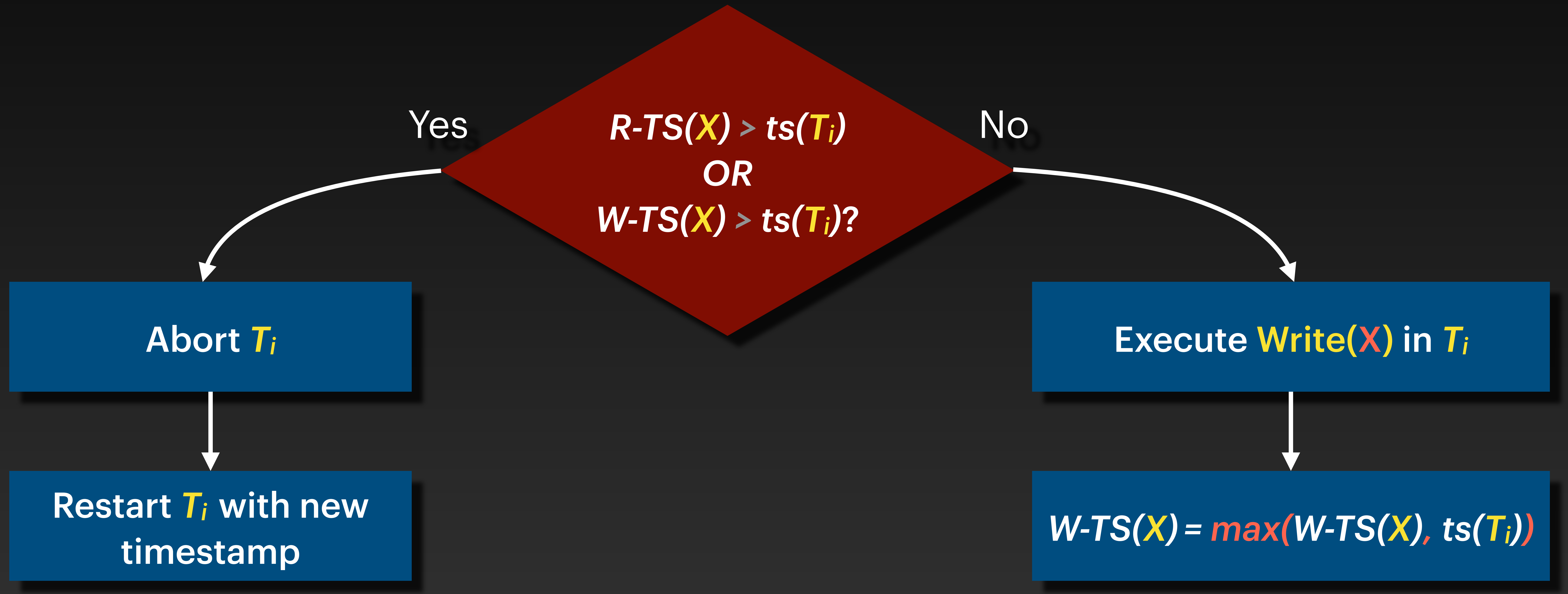
- Transaction T_i wants to read object X



Amr Elhelw's TECH VAULT

Timestamp Ordering: Writes

- Transaction T_i wants to write object X



Amr Elhelw's TECH VAULT


Time
T1
T2
Read (B)
Read (B)
Write (B)
Read (A)
Read (A)
Read (A)
Write (A)
Commit
Commit
A: R-TS = 0
W-TS = 0
B: R-TS = 0
W-TS = 0
Amr Elhelw's
TECH
VAULT

Time	T1 ts = 1	T2
	Read (B)	
		Read (B)
		Write (B)
	Read (A)	
		Read (A)
	Read (A)	
		Write (A)
	Commit	
		Commit

A: R-TS = 0

W-TS = 0

B: R-TS = 1

W-TS = 0

Amr Elhelw's TECH VAULT

Time

T1 ts = 1

Read (B)

Read (A)

Read (A)

Commit

T2 ts = 2

Read (B)

Write (B)

Read (A)

Write (A)

Commit

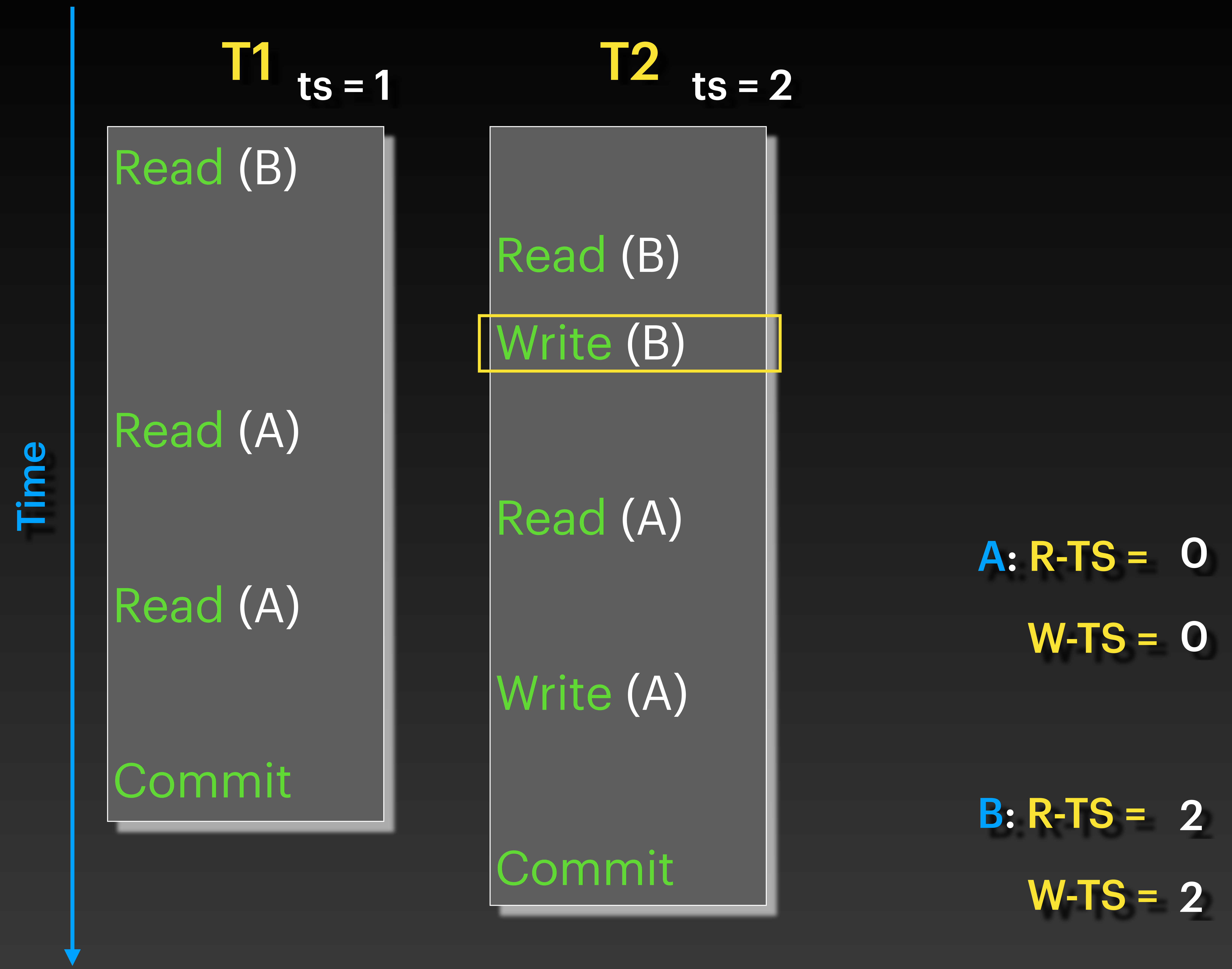
A: R-TS = 0

W-TS = 0

B: R-TS = 2

W-TS = 0

Amr Elhelw's TECH VAULT



Amr Elhelw's TECH VAULT

Time	T1 ts = 1	T2 ts = 2
	Read (B)	
		Read (B)
		Write (B)
	Read (A)	
		Read (A)
	Read (A)	
		Write (A)
	Commit	
		Commit

A: R-TS = 1

W-TS = 0

B: R-TS = 2

W-TS = 2

Amr Elhelw's TECH VAULT

Time

T1 ts = 1	T2 ts = 2
Read (B)	
	Read (B)
	Write (B)
Read (A)	
	$\boxed{\text{Read (A)}}$
Read (A)	
	Write (A)
Commit	
	Commit

A: R-TS = 2

W-TS = 0

B: R-TS = 2

W-TS = 2

Amr Elhelw's
TECH
VAULT

Time

T1 ts = 1	T2 ts = 2
Read (B)	
	Read (B)
	Write (B)
Read (A)	
	Read (A)
$\boxed{\text{Read (A)}}$	
	Write (A)
Commit	
	Commit

A: R-TS = 2

W-TS = 0

B: R-TS = 2

W-TS = 2

Amr Elhelw's TECH VAULT

Time

T1 ts = 1	T2 ts = 2
Read (B)	
	Read (B)
	Write (B)
Read (A)	
	Read (A)
Read (A)	
	Write (A)
Commit	
	Commit

A: R-TS = 2

W-TS = 2

B: R-TS = 2

W-TS = 2

Amr Elhelw's TECH VAULT

Time
T1
Read (A)
Write (A)
Read (A)
Commit
T2
Write (A)
Commit
A: R-TS = 0
W-TS = 0
Amr Elhelw's
TECH
VAULT

Time

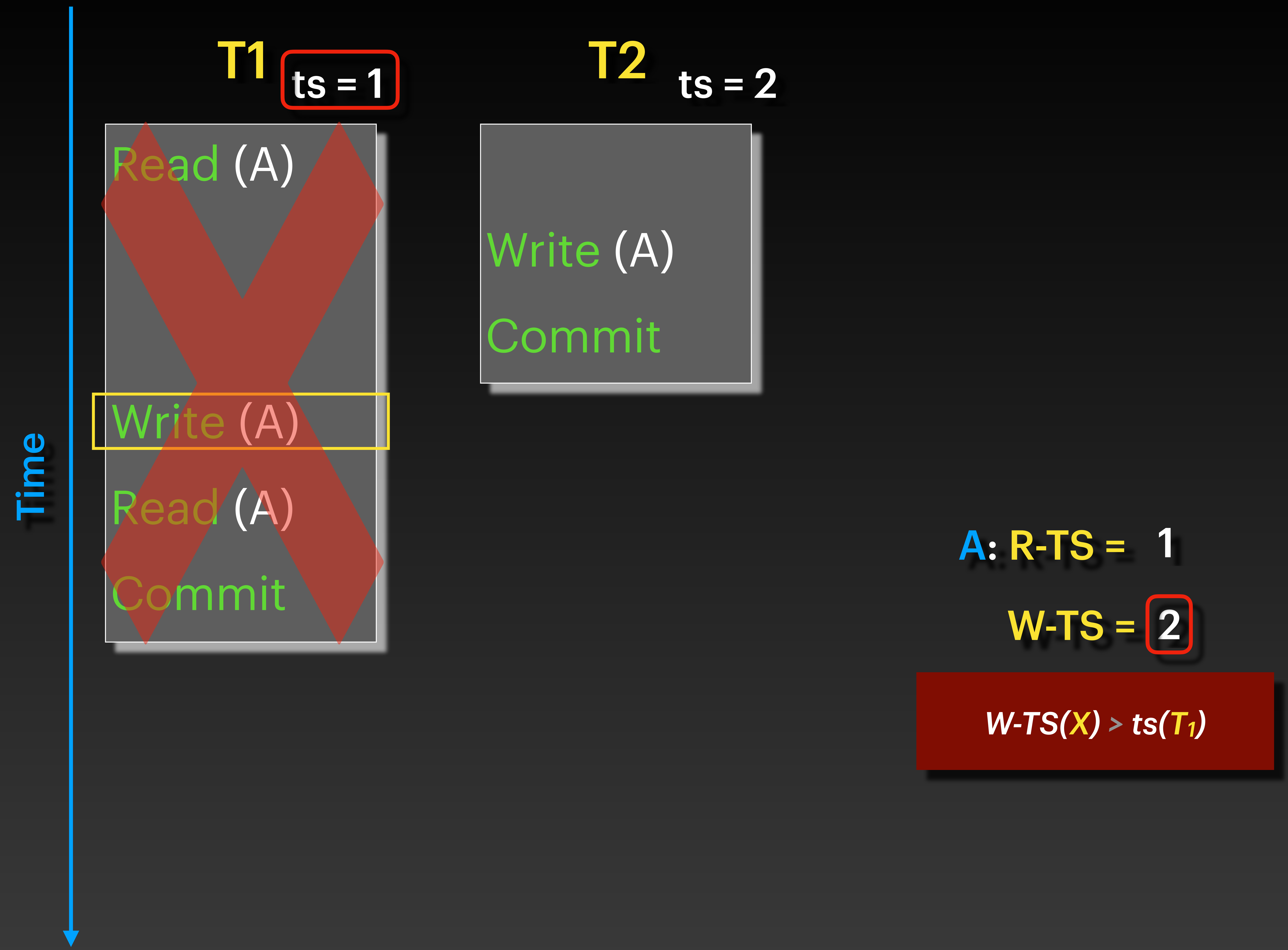
T1 ts = 1	T2
Read (A)	
	Write (A)
	Commit
Write (A)	
Read (A)	
Commit	

A: R-TS = 1

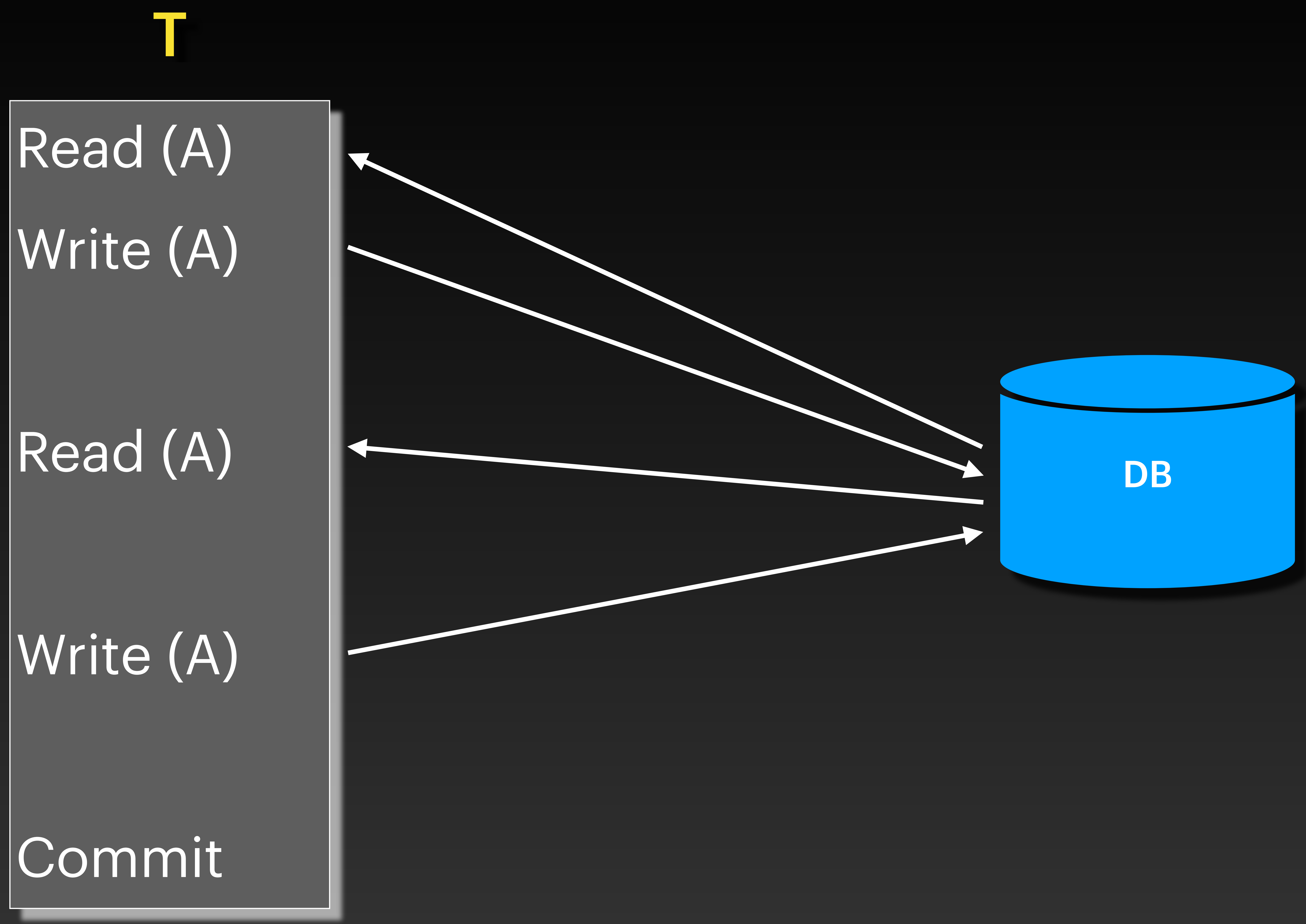
W-TS = 0

Amr Elhelw's TECH VAULT

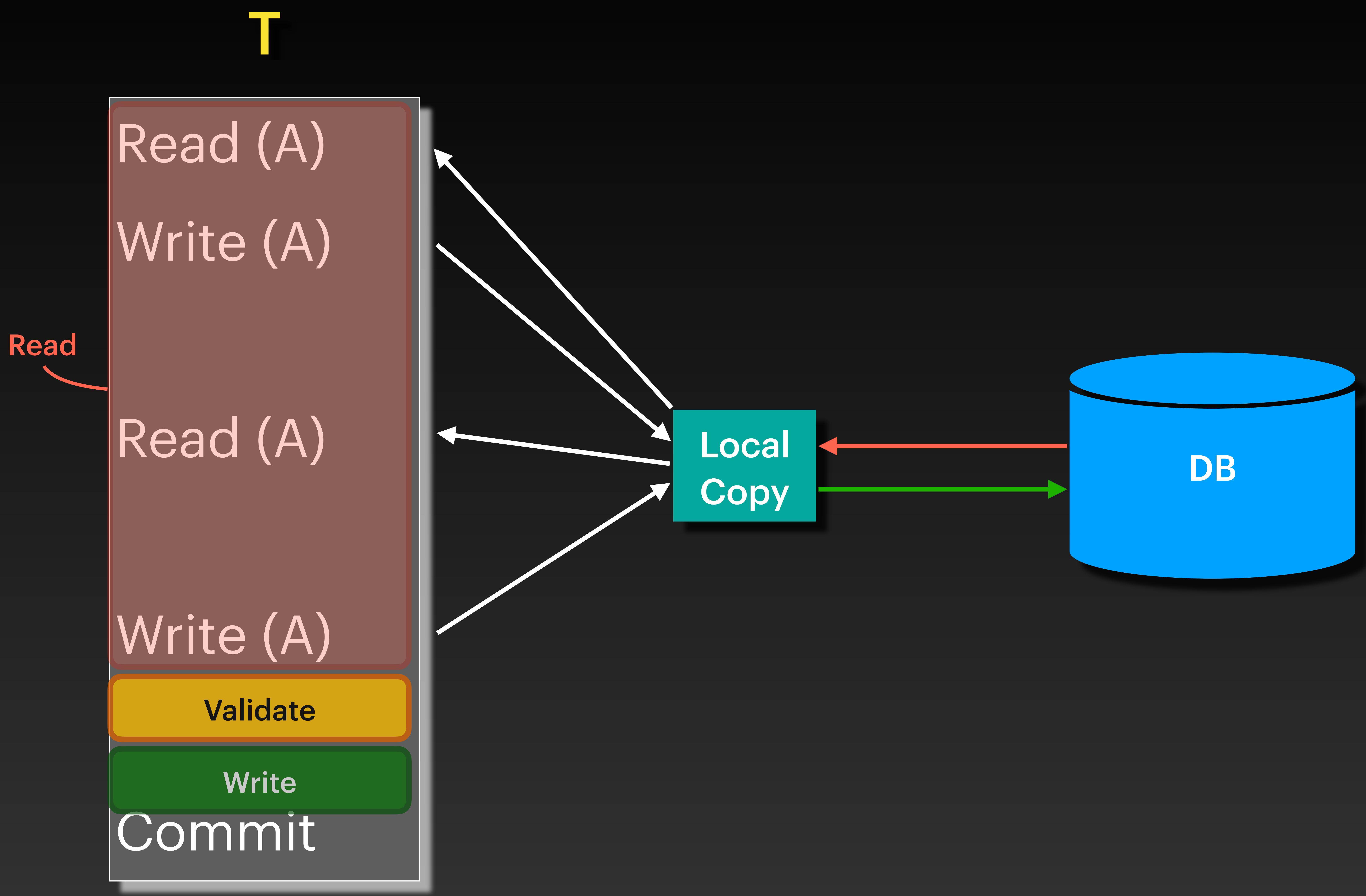

Time
T1 ts = 1
Read (A)
Write (A)
Read (A)
Commit
T2 ts = 2
Write (A)
Commit
A: R-TS = 1
W-TS = 2
Amr Elhelw's
TECH
VAULT



T1
ts = 1
Read (A)
Write (A)
Read (A)
Commit
T2
ts = 2
Write (A)
Commit
Time
A: R-TS = 1
W-TS = 2
W-TS(X) > ts(T1)
Amr Elhelw's
TECH
VAULT



Amr Elhelw's TECH VAULT



Amr Elhelw's TECH VAULT

Optimistic Concurrency Control (OCC)

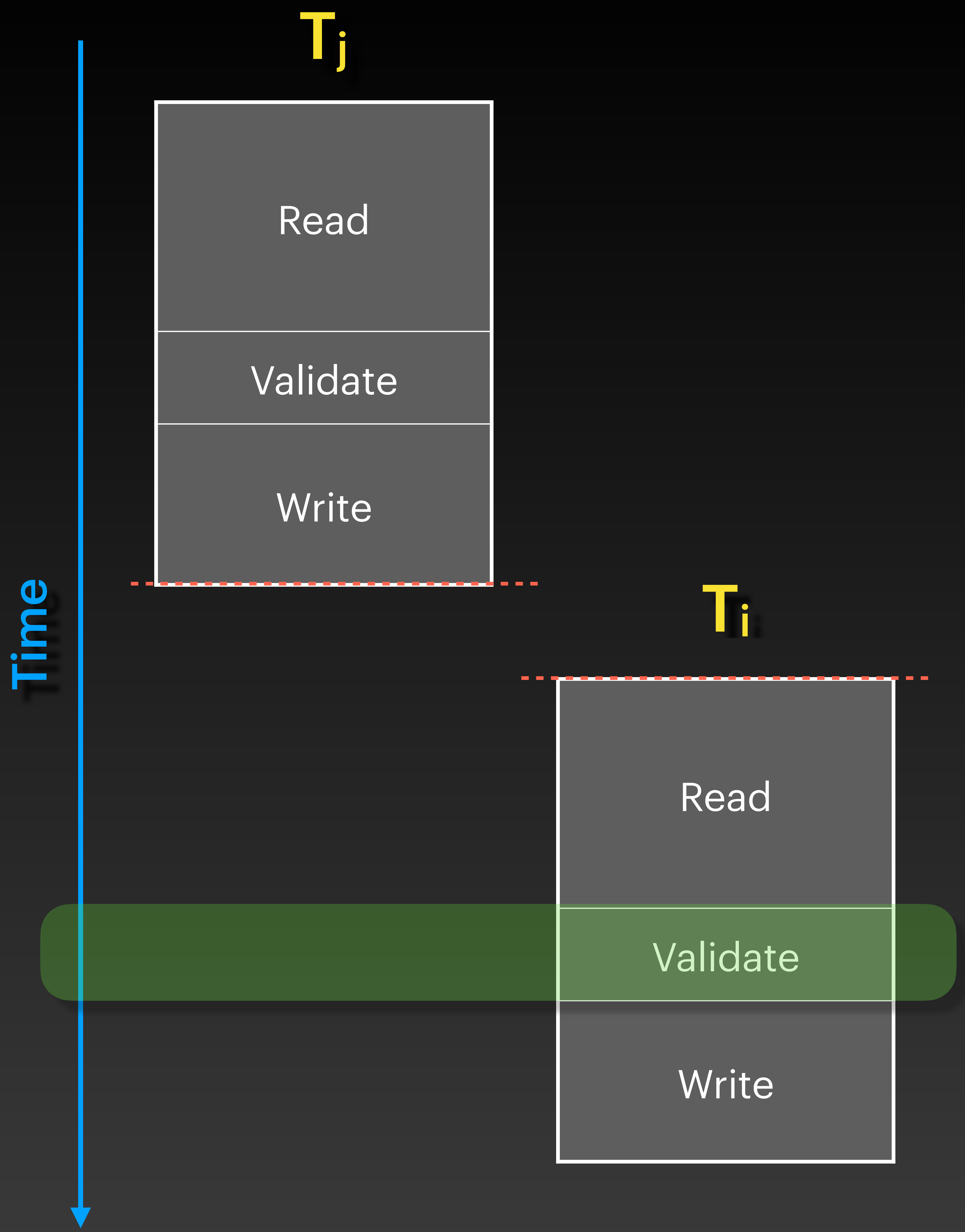
1. Read Phase
   - Read data objects from DB to local copy.
   - Any further reads/writes access the local copy.
2. Validation Phase
   - Check that serializability is not violated
3. Write Phase
   - If validation is successful, write local copy back to DB.
   - Otherwise, abort transaction and discard local copy.

Amr Elhelw's
TECH
VAULT

Validation Phase

- Check current transaction against other transactions in/past their validation phase (including committed transactions)
- Information required for validation:
  - *Transaction timestamp* - for individual stages
  - *ReadSet* - All objects read by a given transaction
  - *WriteSet* - All objects written by a given transaction

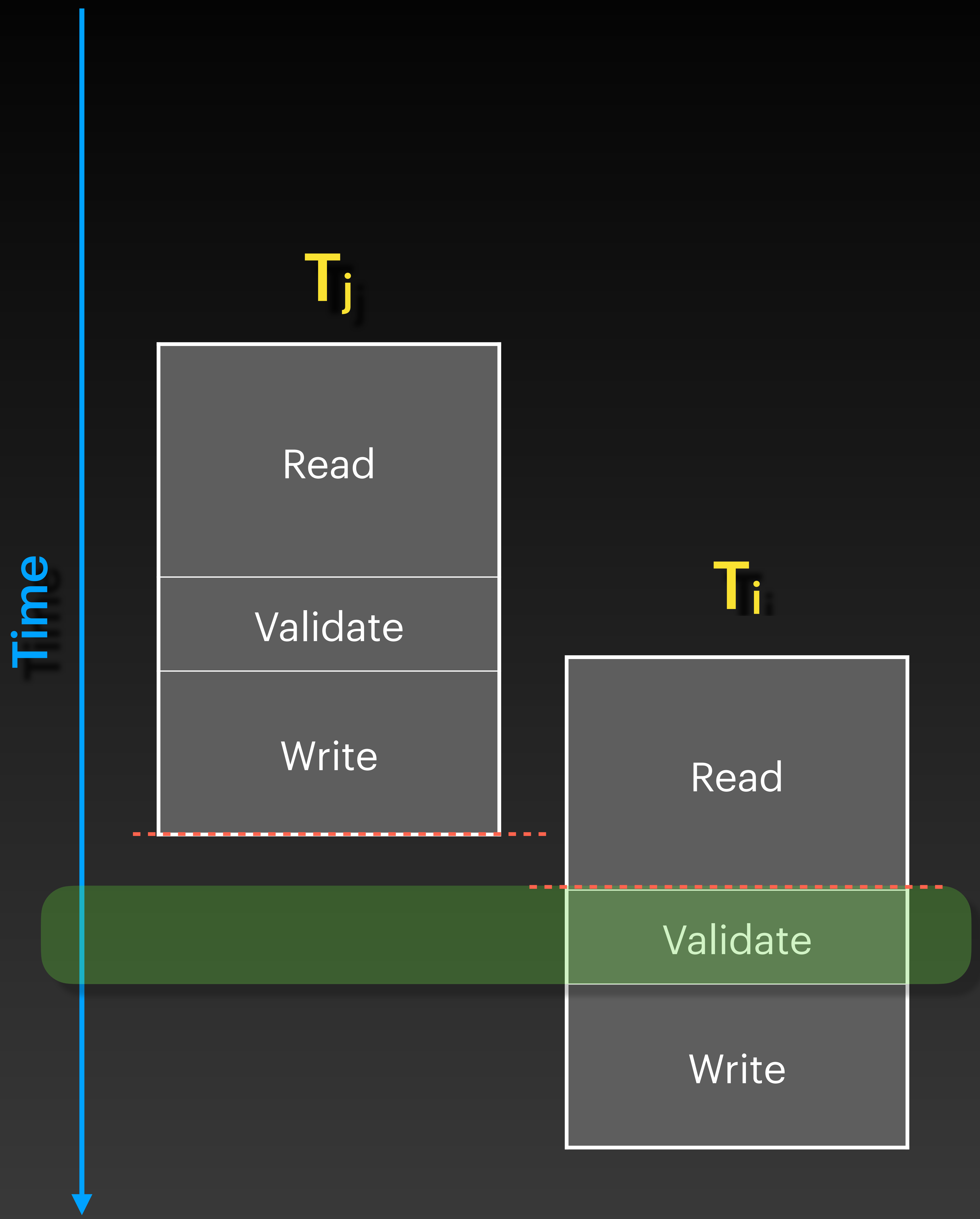
Amr Elhelw's TECH VAULT



Case 1

T_i starts its read phase after T_j completes its write phase.

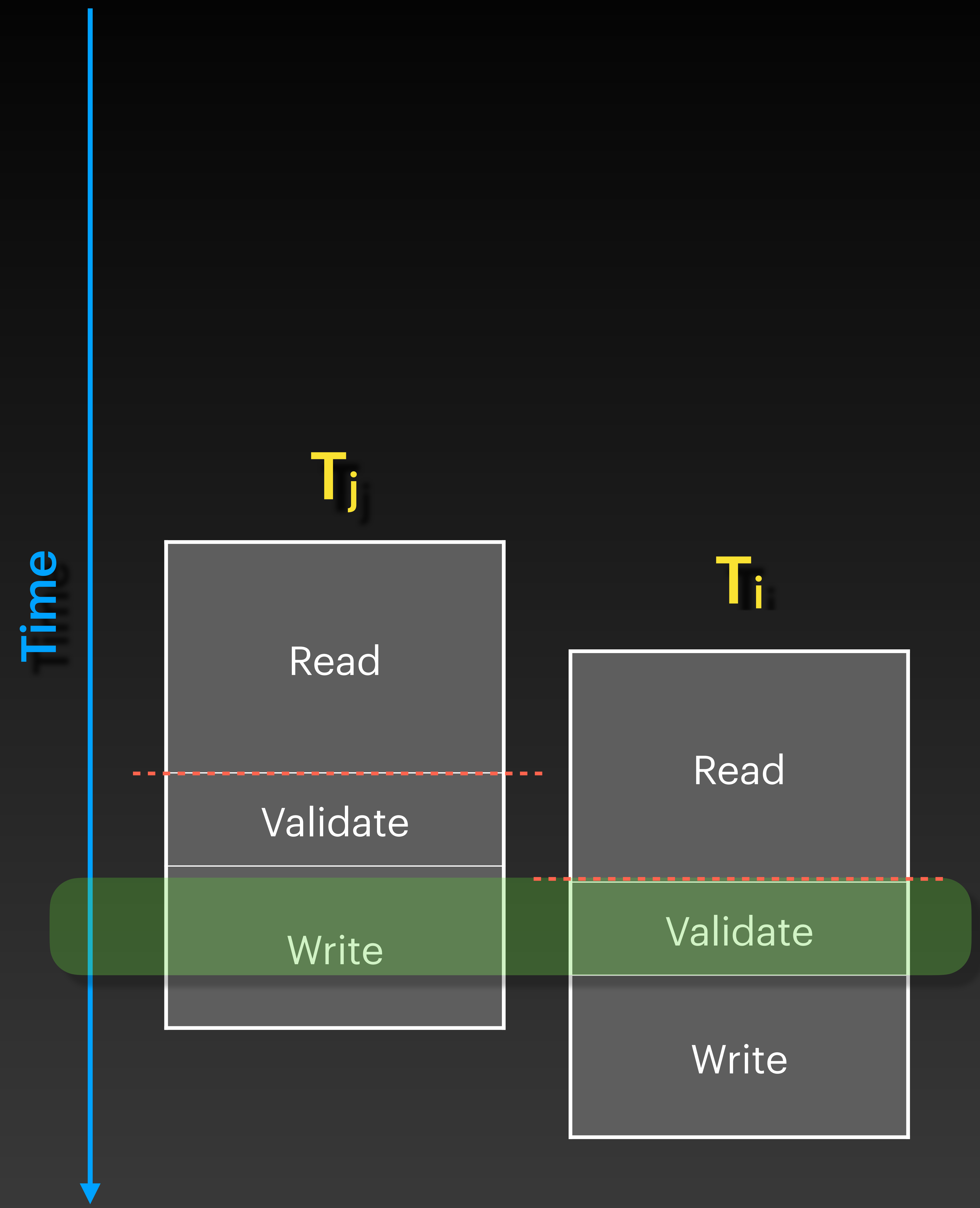
Amr Elhelw's TECH VAULT



Case 2

- T_i completes its read phase after T_j completes its write phase
- AND ReadSet(T_i) ∩ WriteSet(T_j) = ∅

Amr Elhelw's TECH VAULT



Case 3

- T_i completes its read phase after T_j completes its read phase
- AND ReadSet(T_i) ∩ WriteSet(T_j) = ∅
- AND WriteSet(T_i) ∩ WriteSet(T_j) = ∅

Amr Elhelw's TECH VAULT