Transactions

Knowledge Objetives

1. Define ACID properties
2. Explain the four kinds of interferences
3. Explain how each standard isolation level avoids the corresponding interference by locking data
4. Explain the correlation between interferences and performance
5. Explain seven things we can do to improve performance, from the point of view of isolation
6. Give five reasons to need recovery
7. Explain the difference between restoration and reconstruction
8. Explain the need of restoration
9. Name five elements in a typical log file
10. Explain the write-ahead log protocol and justify its need
11. Name the four rebuilding steps
12. Give two reasons to place the log file in a dedicated disk
13. Explain why transaction should be as short as possible from the point of view of recovery as well as concurrency
14. Explain what chained transactions are

Understanding Objetives

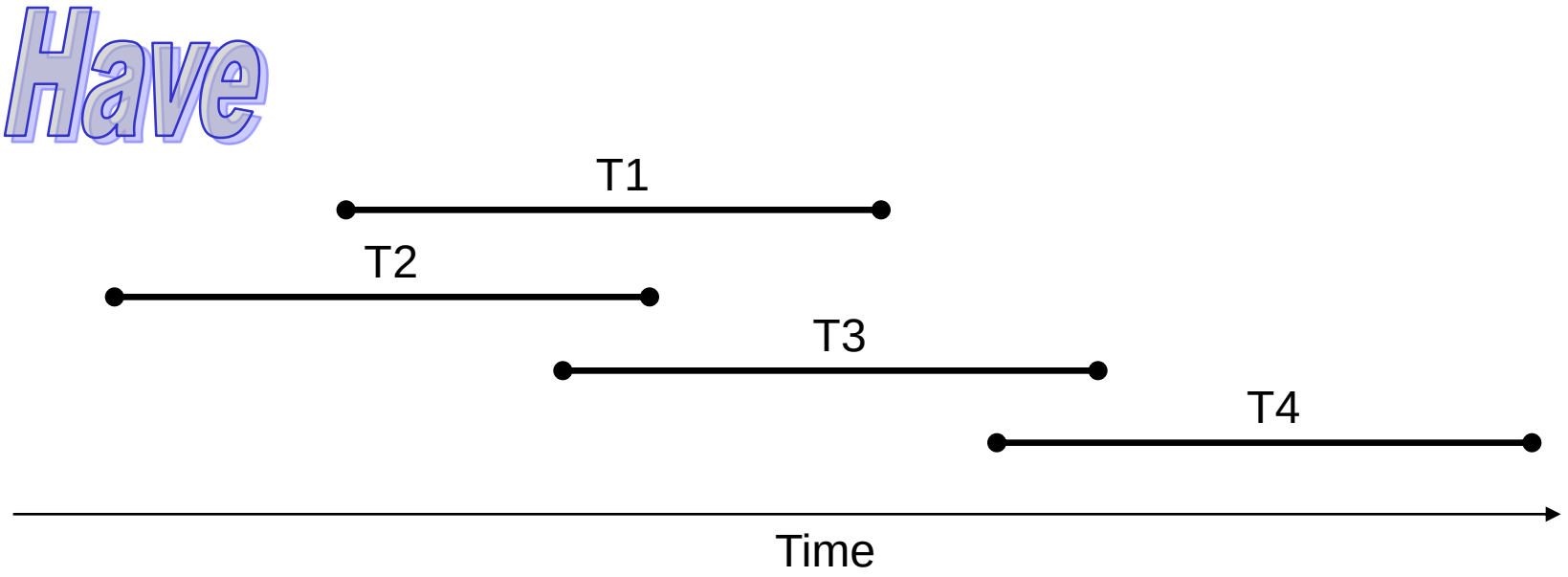
ACID properties

- *Atomicity*
- *Consistency*
- *Isolation*
- *Durability*

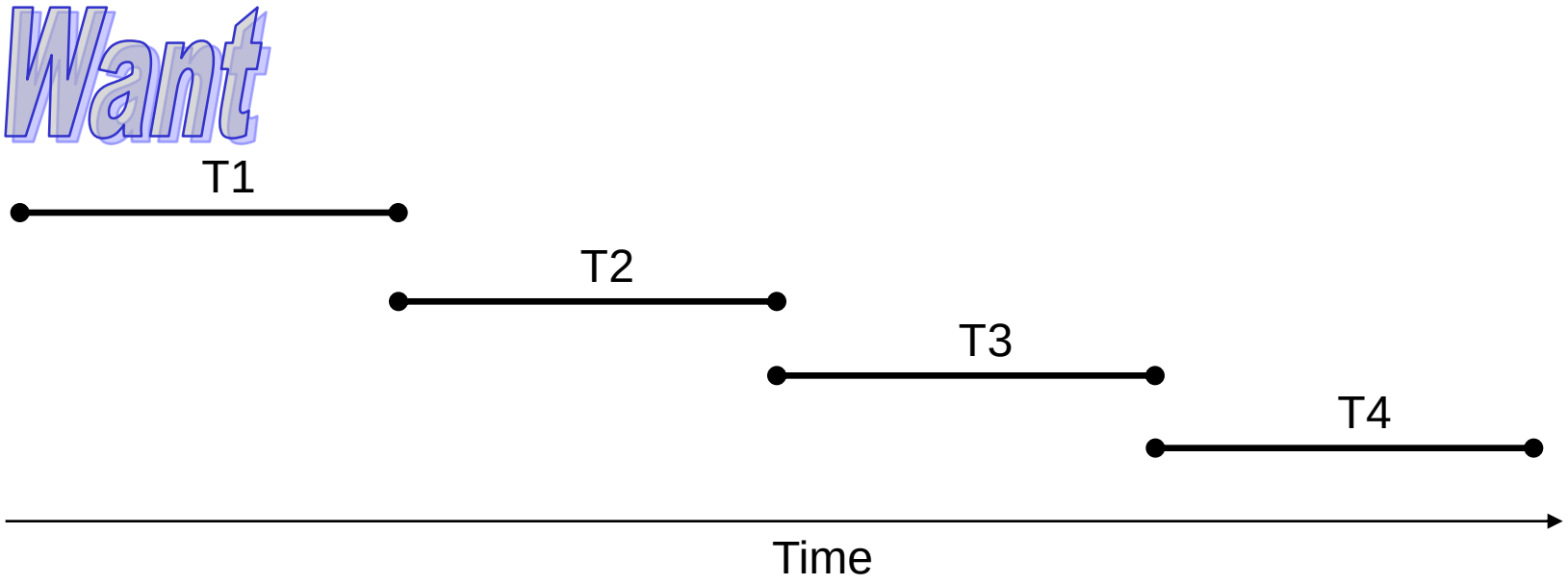
ACID properties

- *Atomicity*
- *Consistency*
- *Isolation*
- *Durability*

Objetive



Objetive



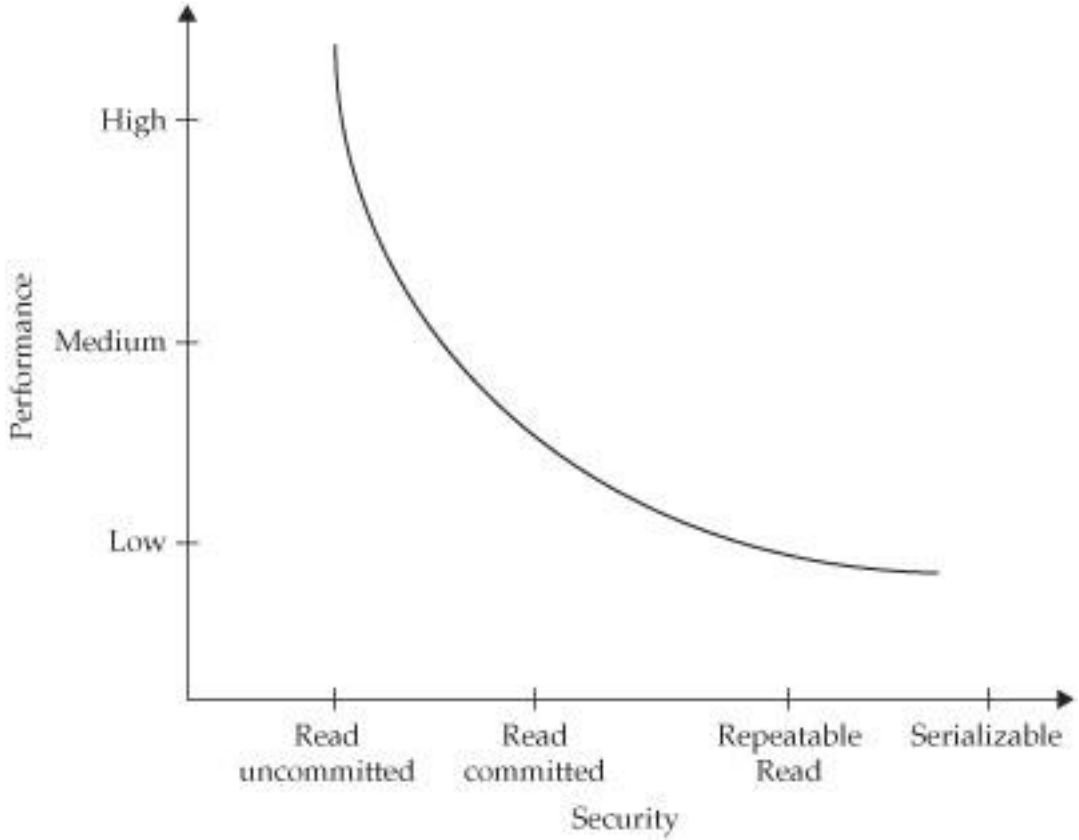
Standard Isolation levels

- READ UNCOMMITTED (avoids lost update)
  - Locks X for W (freed at the end of Tx)
  - No locking for R
- READ COMMITTED (avoids read uncommitted)
  - Locks S for R (freed as soon as read ends)
- REPEATABLE READ (avoids unrepeatable read)
  - Strict two phase locking protocol
- SERIALIZABLE (avoids phantoms)
  - Locks tables/indexes

	S	X
Shared	OK	NO
e**X**clusive	NO	NO

Isolation & Performance

- Number of locks per Tx
- Kind of locks
- Time that a Tx keeps the locks



Tuning

- Relax isolation requirements if the application allows it
- Remove locking if it is **un**necessary (2 cases)
- Circumvent hot spots
  - Access them as late as possible
  - Partitioning
    - Specially useful for insertions
  - Use the mechanisms provided by the DBMS
    - Surrogates
    - Read-only transaction
- Select the proper deadlock interval
- Use DDL with few users logged in (or no user if possible)
- Chop transactions

ACID properties

- *Atomicity*
- *Consistency*
- *Isolation*
- *Durability*

ACID properties

- *Atomicity*
- *Consistency*
- *Isolation*
- *Durability*

Atomicity and Durability

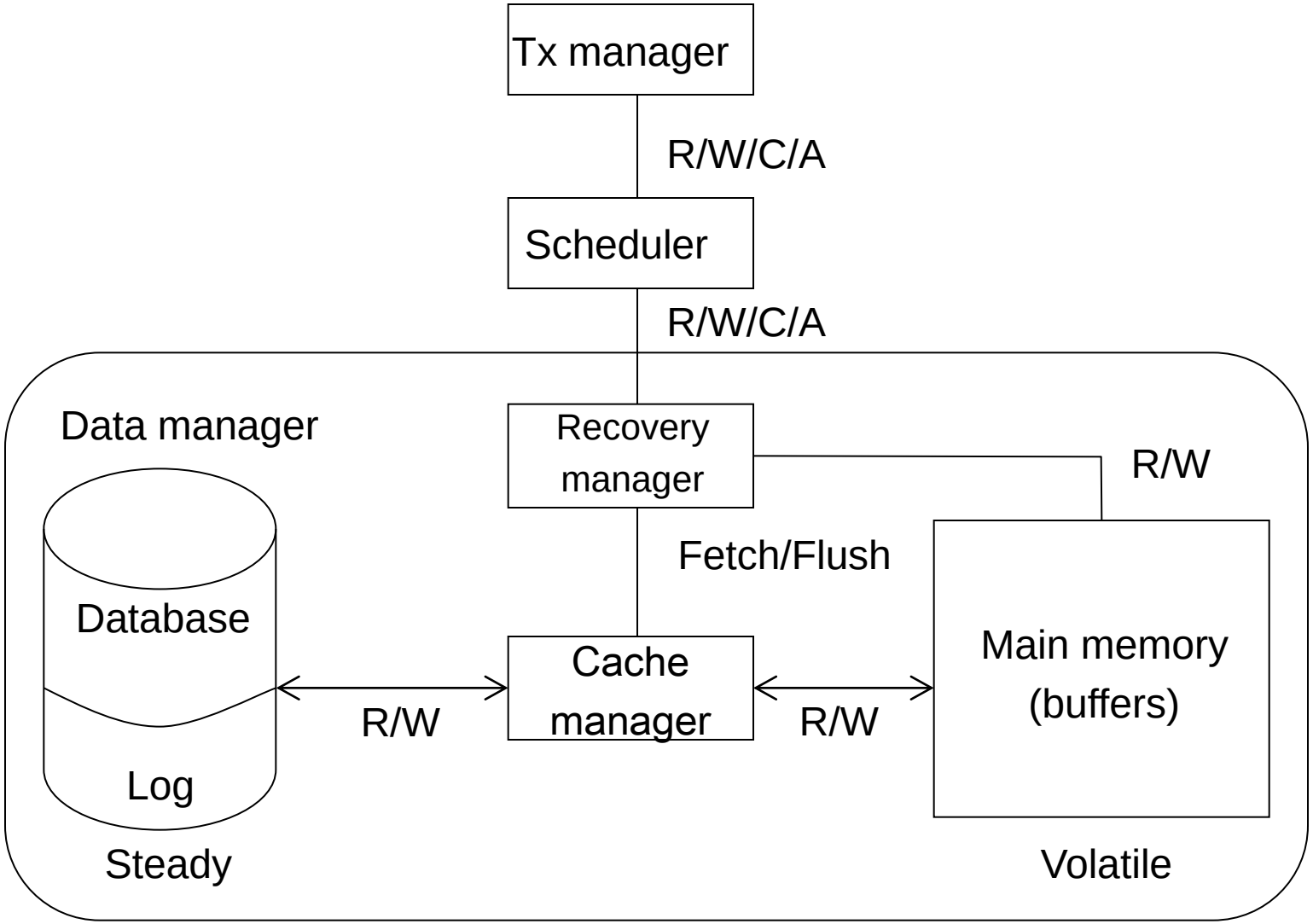
- Each transaction ends with either COMMIT or ROLLBACK

- Even in case of failure:
  - Effects of committed transactions remain
  - Effects of aborted transactions do not leave trace

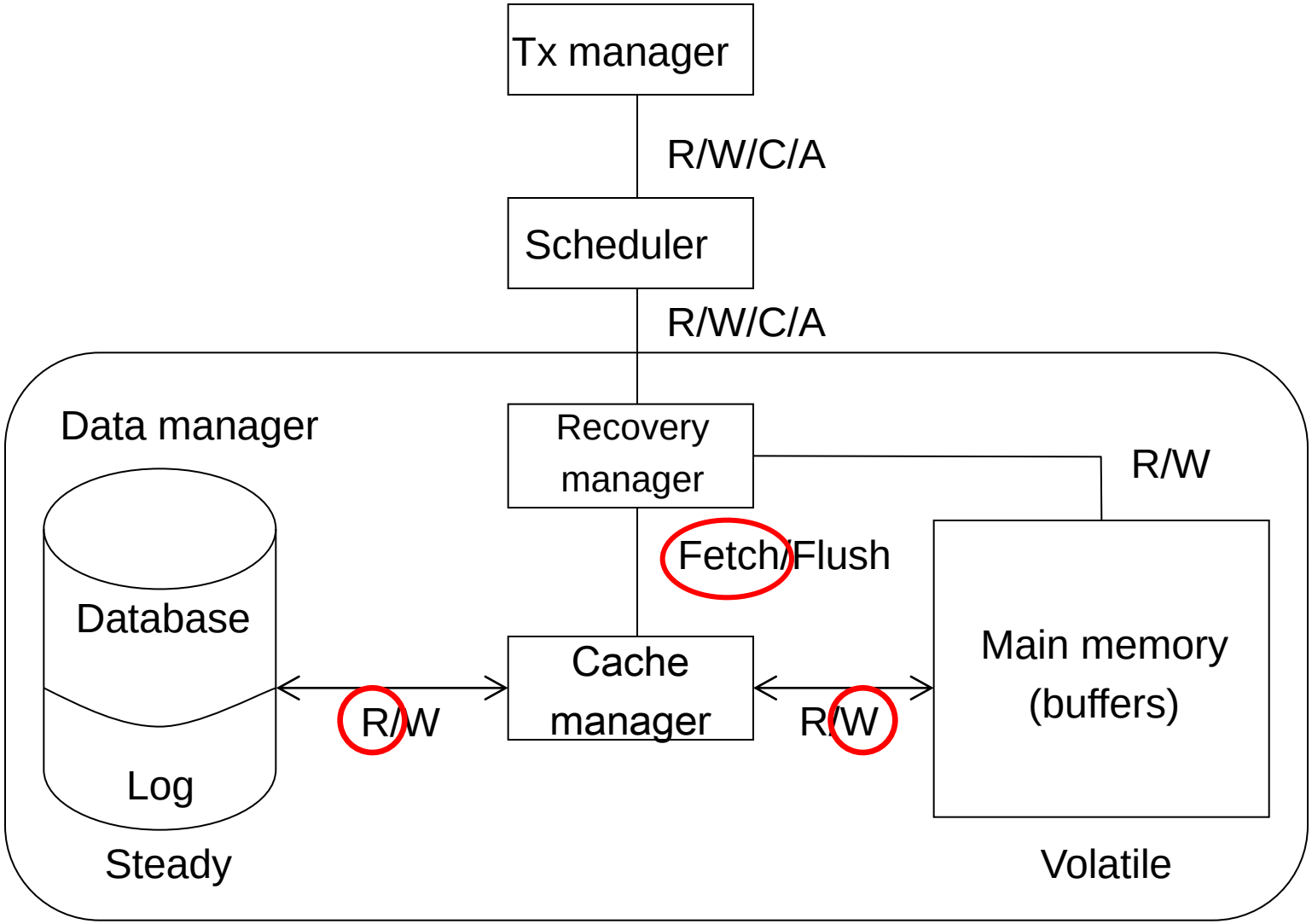
Reasons to need recovery

- User cancels
- A deadlock is detected
- Software failure (or virus)
- Hardware failure
- External factors (earthquake, fire, etc.)

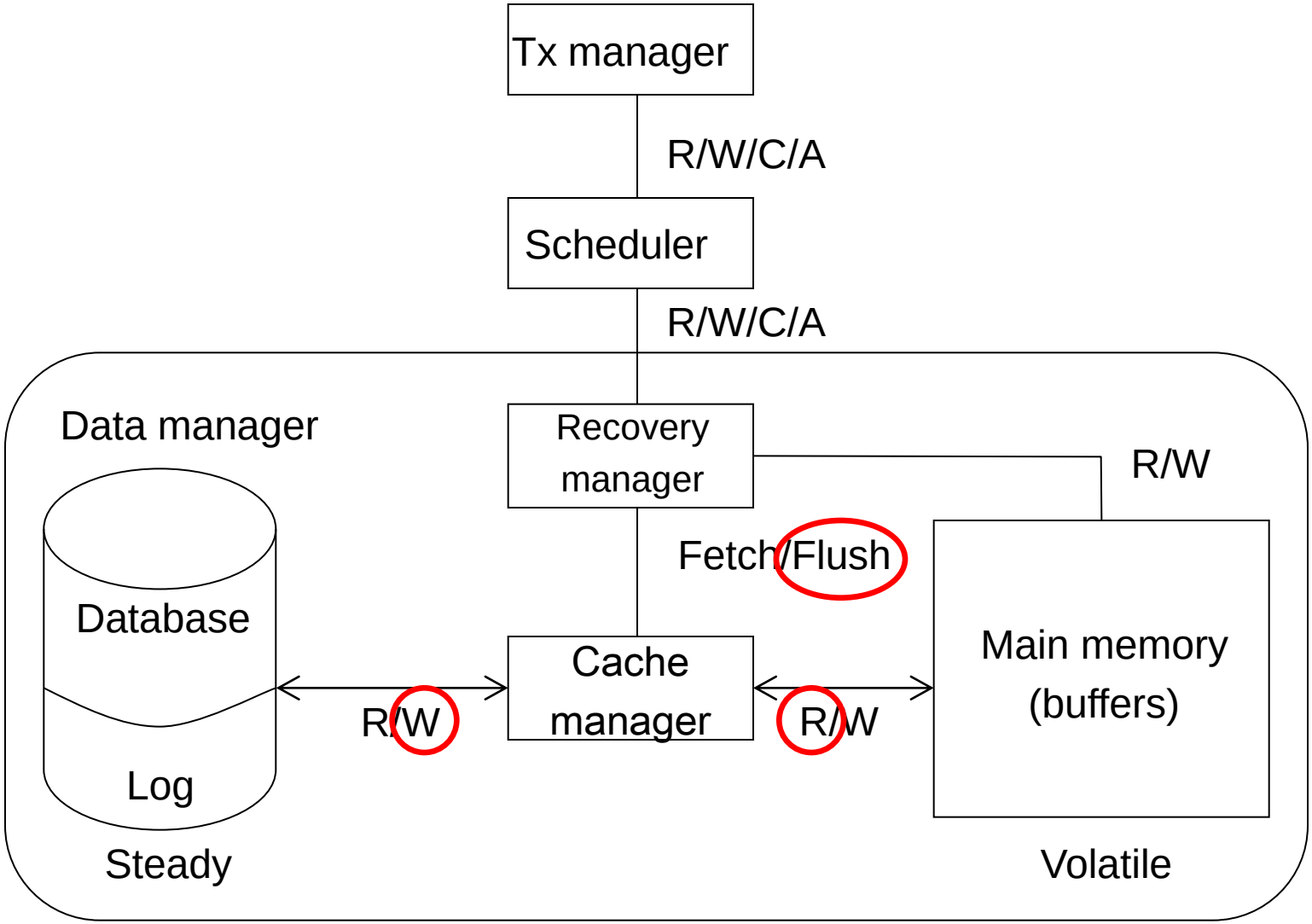
Transactions manager architecture



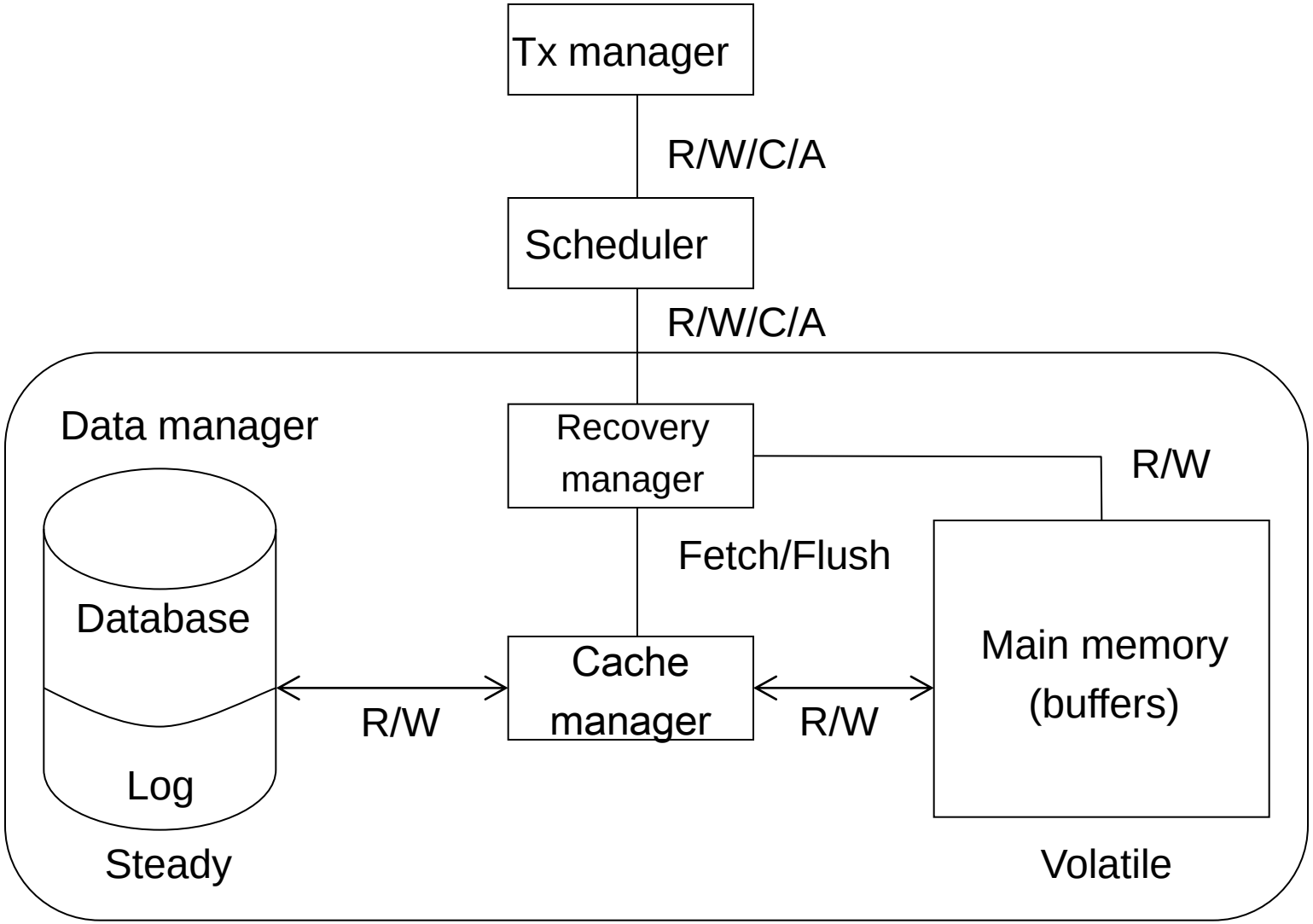
Transactions manager architecture



Transactions manager architecture



Transactions manager architecture



Restoration vs Rebuilding

- Restoration: Undo/Redo
  (Atomicity)

- Reconstruction: Hardware failure
  (Durability)

Kind of entries in the log

- BEGIN TRANSACTION
- Operation
  - Kind of operation
  - ID of the object
  - Old and new values
  - Pointer to the previous operation in the Tx
    - Used to Undo
  - Pointer to the next operation in the Tx
    - Used to Redo
- COMMIT/ROLLBACK

Log rules

- Transactions write in the buffers (not in disk)
- At some time, dirty pages are written to disk
  - At regular intervals
  - When the number of dirty pages is above a threshold
  - When the log is full
  - During a backup
- Write-Ahead Log protocol
  - The log is always written before the DB
- The log file is cyclic

Reconstruction steps

Current DB state = backup + log

Reconstruction steps

Current DB state = backup + log

1. Fix/Replace damaged hardware

Reconstruction steps

Current DB state = backup + log

1. Fix/Replace damaged hardware
2. Find the backup prior to the accident

Reconstruction steps

Current DB state = backup + log

1. Fix/Replace damaged hardware
2. Find the backup prior to the accident
3. Load the backup into the DB

Reconstruction steps

Current DB state = backup + log

1. Fix/Replace damaged hardware
2. Find the backup prior to the accident
3. Load the backup into the DB
4. Redo (based on log) all changes from the backup to now

Backup considerations

- Allows discarding of previous log entries
- In case of backup failure, we will lose data
- In case of log failure, we will lose data (even if we have a backup)
- Generates two problems:
  - Requires space
  - Increases response time while running
- It can be used to make decisions
- It is usually performed once or twice a day at most

Recovery tuning

- Place the log in a dedicated disk (2 reasons)
- Defer flushes as much as possible
- Weigh up recovery time vs performance free of failure
  - Backup requires space and time
- Chop long read-write transactions

Long transactions

- Problems
  - Concurrency control
    - Other transactions wait too much
    - This will probably have to wait
  - Recovery
    - It will take too long to recover
    - It is likely that a system failure occurs when it is half complete
- Solution
  - Chopping
    - Depends on the set of concurrent transaction
      - Could others interfere with this?
      - Could this interfere with others?
      - Does it matter?

Chained transactions in SQL'99

COMMIT AND CHAIN;

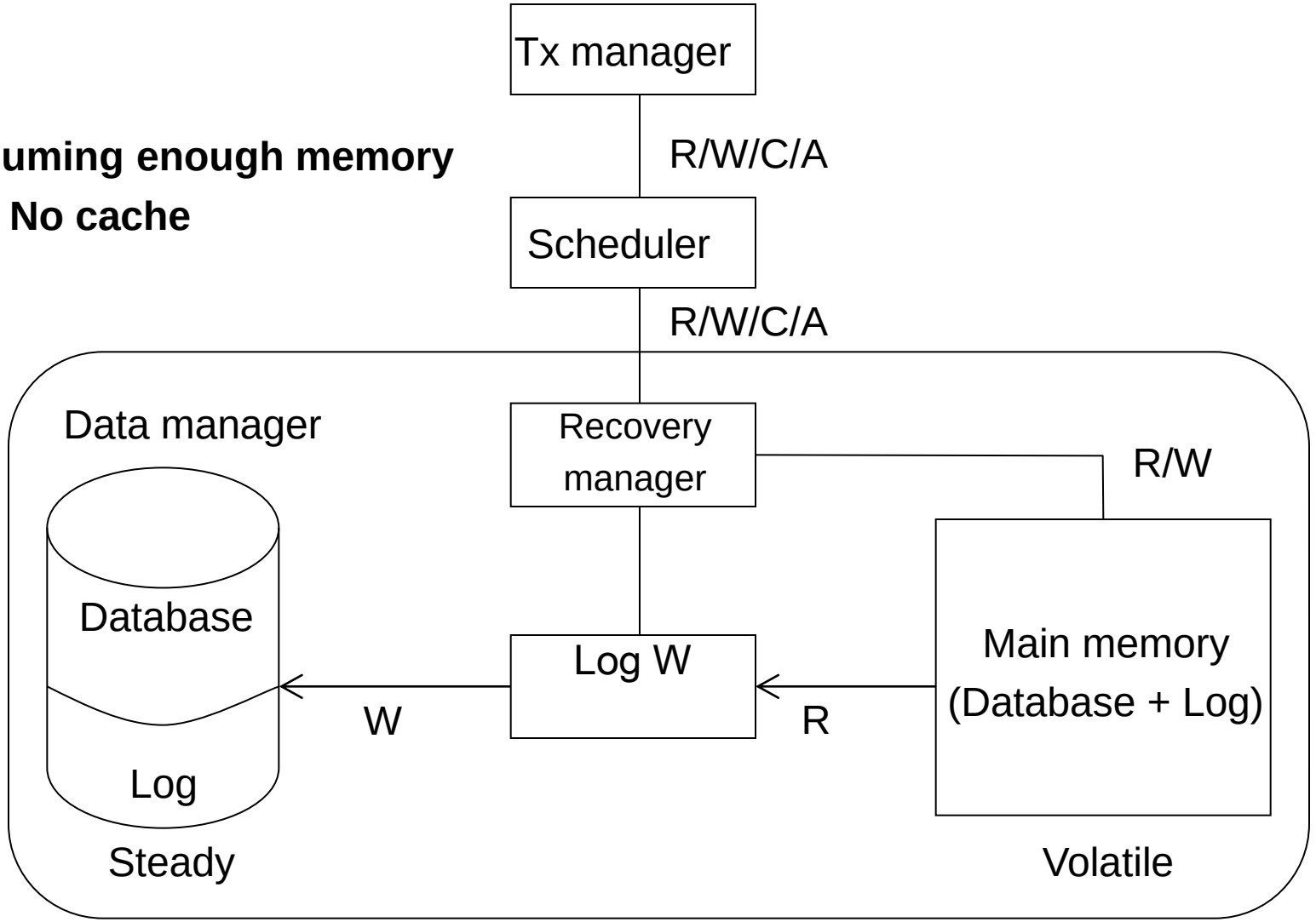
ROLLBACK AND CHAIN;

- Chains transactions
  - It does not free resources
- Preserves Tx configuration
- Advantages faced with a sequence of Tx:
  - Avoid unnecessary overload of freeing resources and immediately ask them again

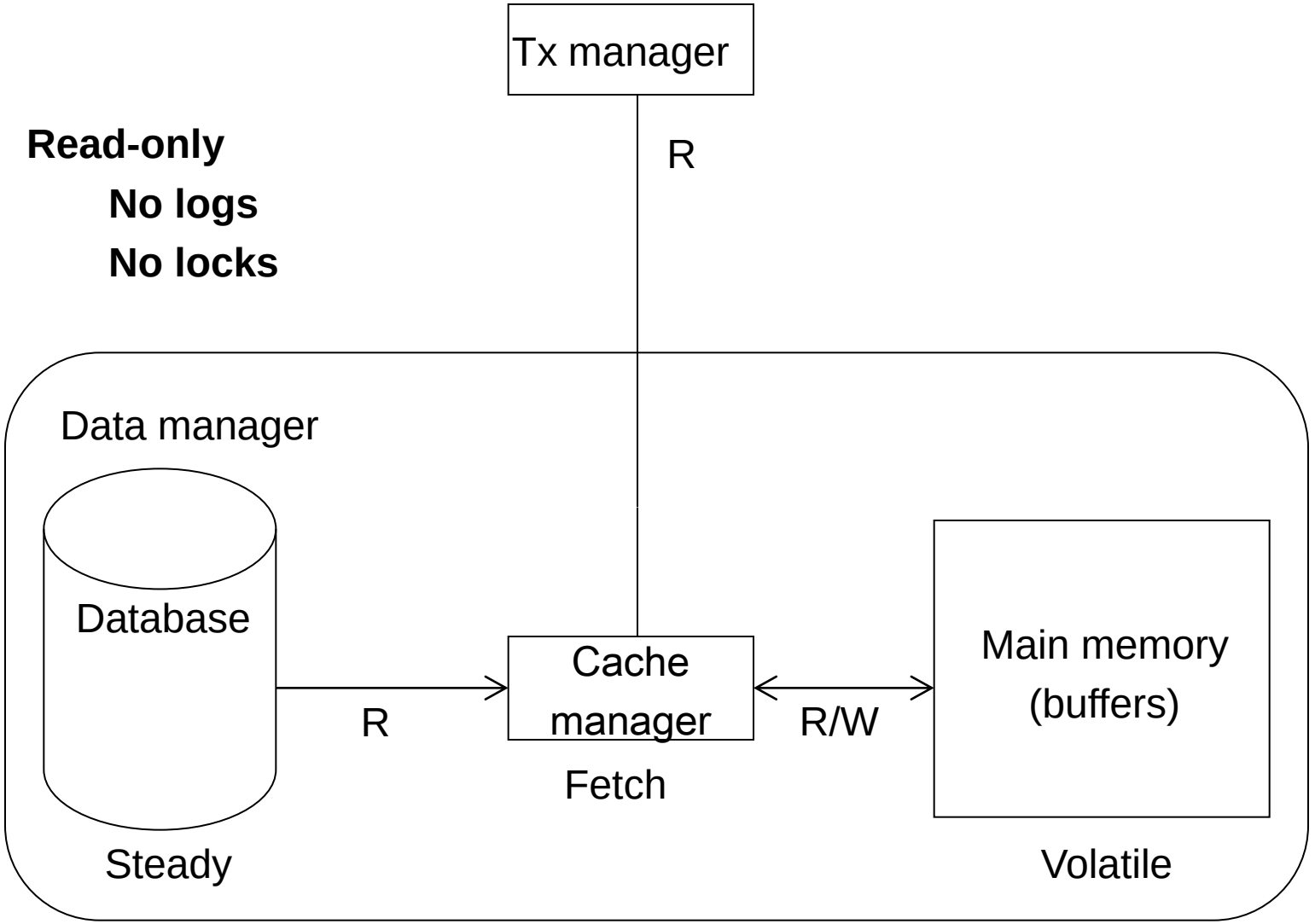
Alternative architecture (OLTP)

Assuming enough memory

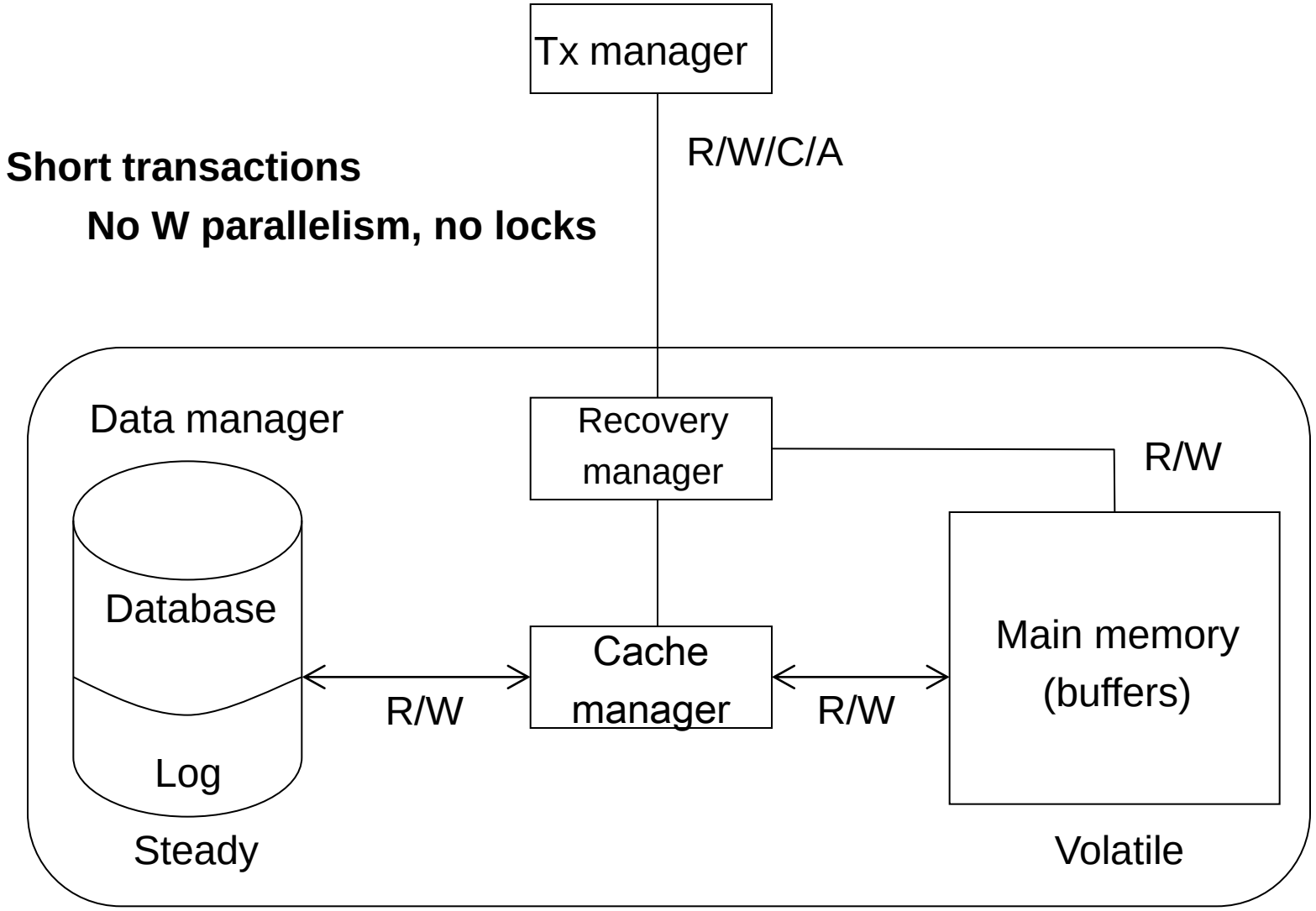
No cache



Alternative architecture (DW)



Alternative architecture (Big Data)



Summary

- ACID properties
  - Isolation
    - Isolation levels
  - Atomicity and durability
    - Restoration
    - Reconstruction

Bibliography

- J. Sistac. *Sistemes de Gestió de Bases de Dades*. Editorial UOC, 2002.
- D. Shasha and P. Bonnet. *Database Tuning*. Elsevier, 2003.
- R. Ramakrishnan and J. Gehrke. *Database Management Systems*. McGraw-Hill, 3rd edition, 2003.
- J. Melton and A. Simon. *SQL 1999*. Morgan Kaufmann, 2002.