TPM

TRANSACTION PROCESSING MONITOR

OVERVIEW OF TPM FOR DISTRIBUTED TRANSACTION PROCESSING

Peter R. Egli
INDIGOO.COM

Contents

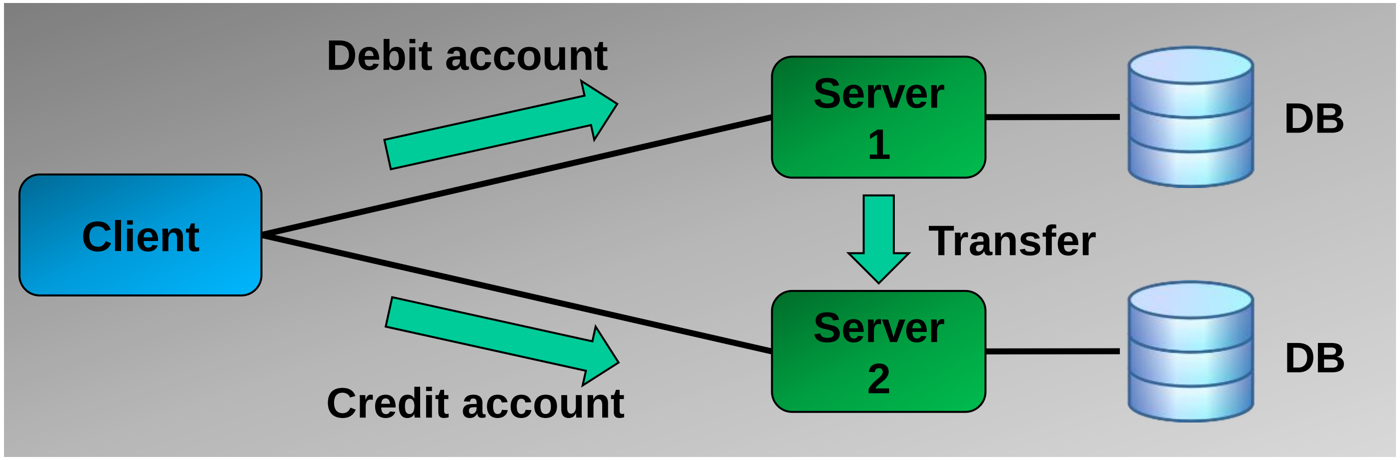
1. What are Transaction Processing Monitors?
2. Properties of DB transactions
3. Two-Phase Commit architecture
4. Two-Phase Commit sequence
5. TPM structure

1. What are Transaction Processing Monitors (TPM)?

Typical problem of distributed applications:

Operations involving multiple databases (DB) require transactional access.

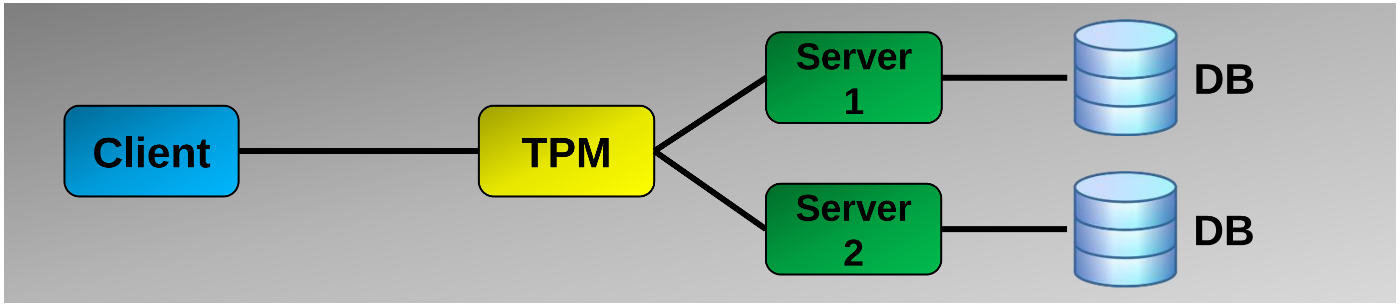
The operation is either successful or fails, but always has to leave all involved databases in a consistent state.



TPM:

TPMs are an early solution for distributed transactions (booking systems, bank account transfers etc.).

TPMs support distributed database transactions by coordinating multiple DB accesses.

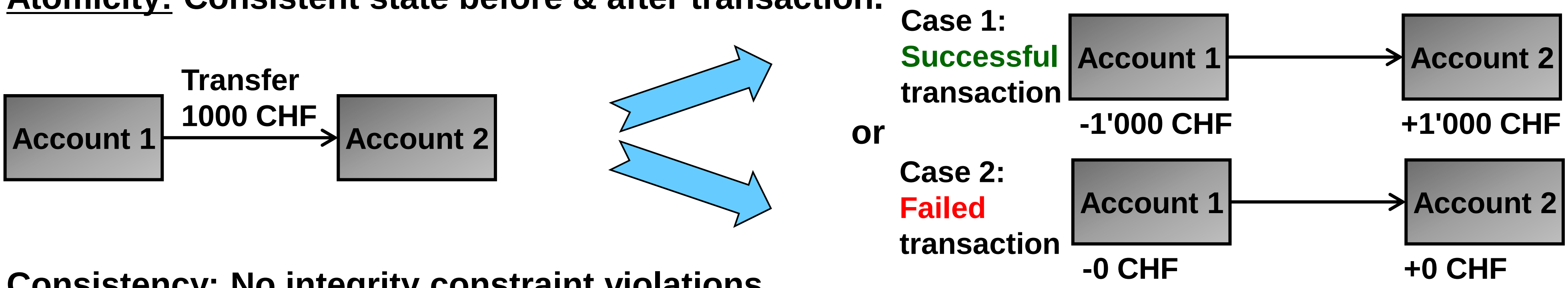


2. Properties of DB transactions (1/2)

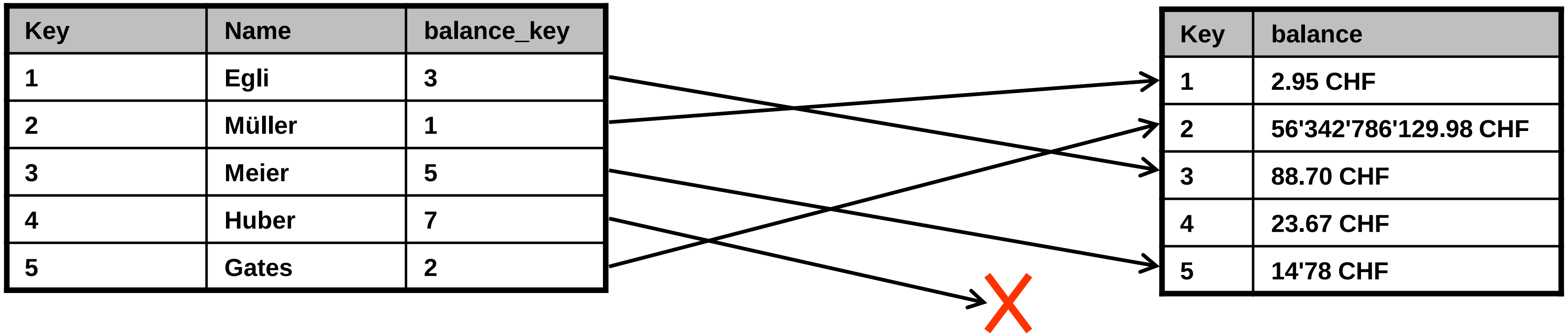
Relational databases support ACID properties:

1. Atomicity
2. Consistency
3. Isolation
4. Durability

Atomicity: Consistent state before & after transaction.



Consistency: No integrity constraint violations



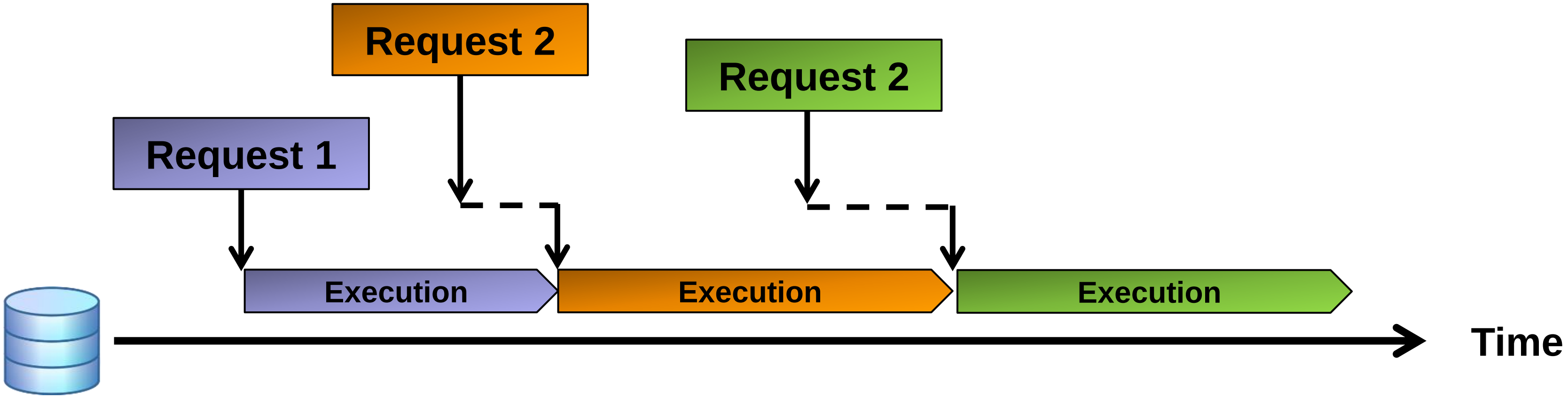
Key	Name	balance_key
1	Egli	3
2	Müller	1
3	Meier	5
4	Huber	7
5	Gates	2

Key	balance
1	2.95 CHF
2	56'342'786'129.98 CHF
3	88.70 CHF
4	23.67 CHF
5	14'78 CHF

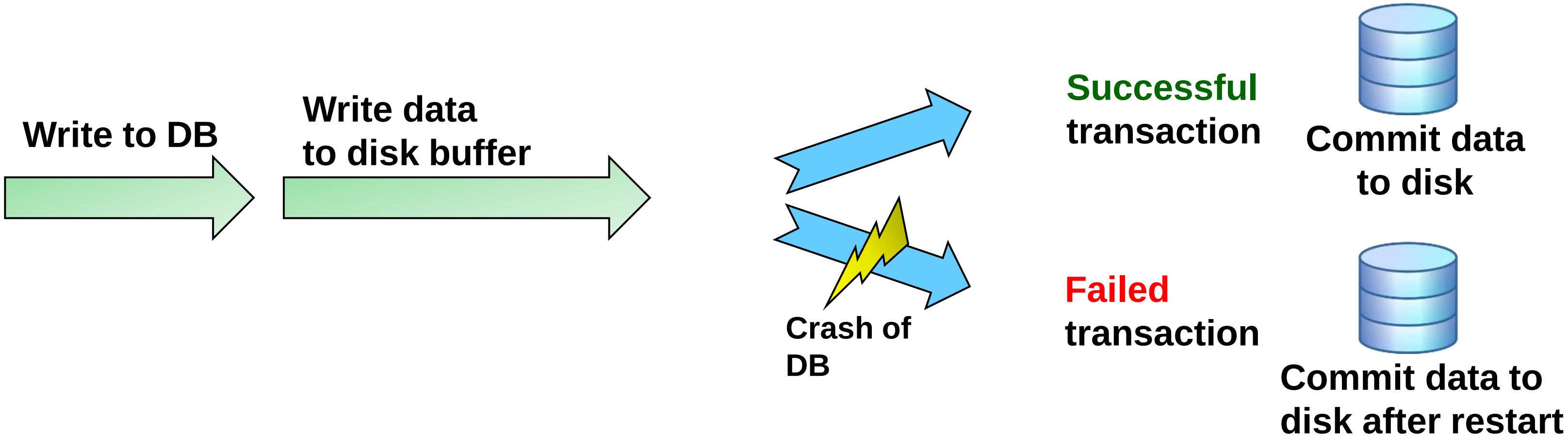
Integrity violation (missing entry with key=7)

2. Properties of DB transactions (2/2)

Isolation: DB accesses are isolated from each other so they do not impact each other. Typically this is done with some sort of locking of resources (tables in DB) and serialization of the requests.



Durability: Transaction is persistent, even in case of errors (crash of DB, error etc.)



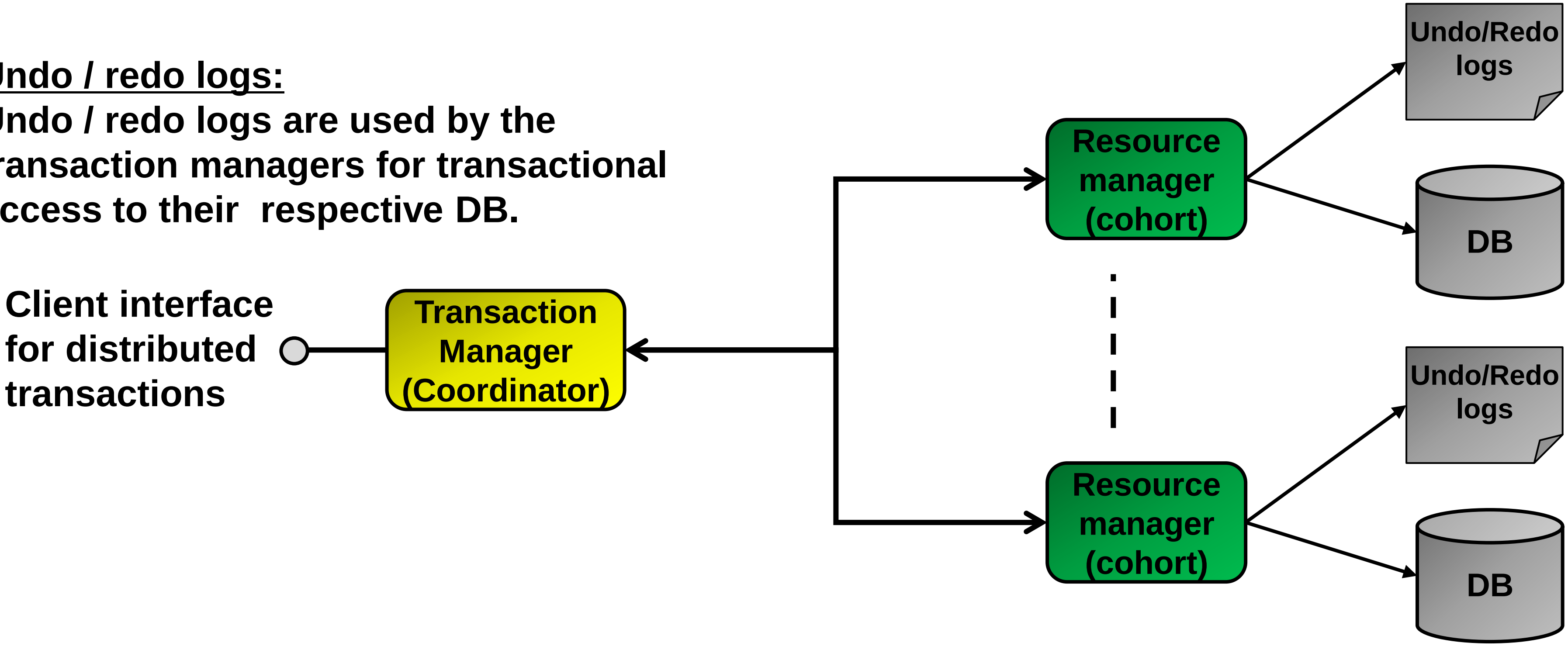
3. Two-Phase Commit architecture

TPMs employ a mechanism called Two-Phase Commit to support distributed DB transactions.

Transaction manager (Coordinator):
The coordinator serves as coordination point in the distributed transaction.

Resource manager:
The resource managers (aka "Cohort") perform the individual transactional DB accesses and are able to either commit or rollback an individual DB access.

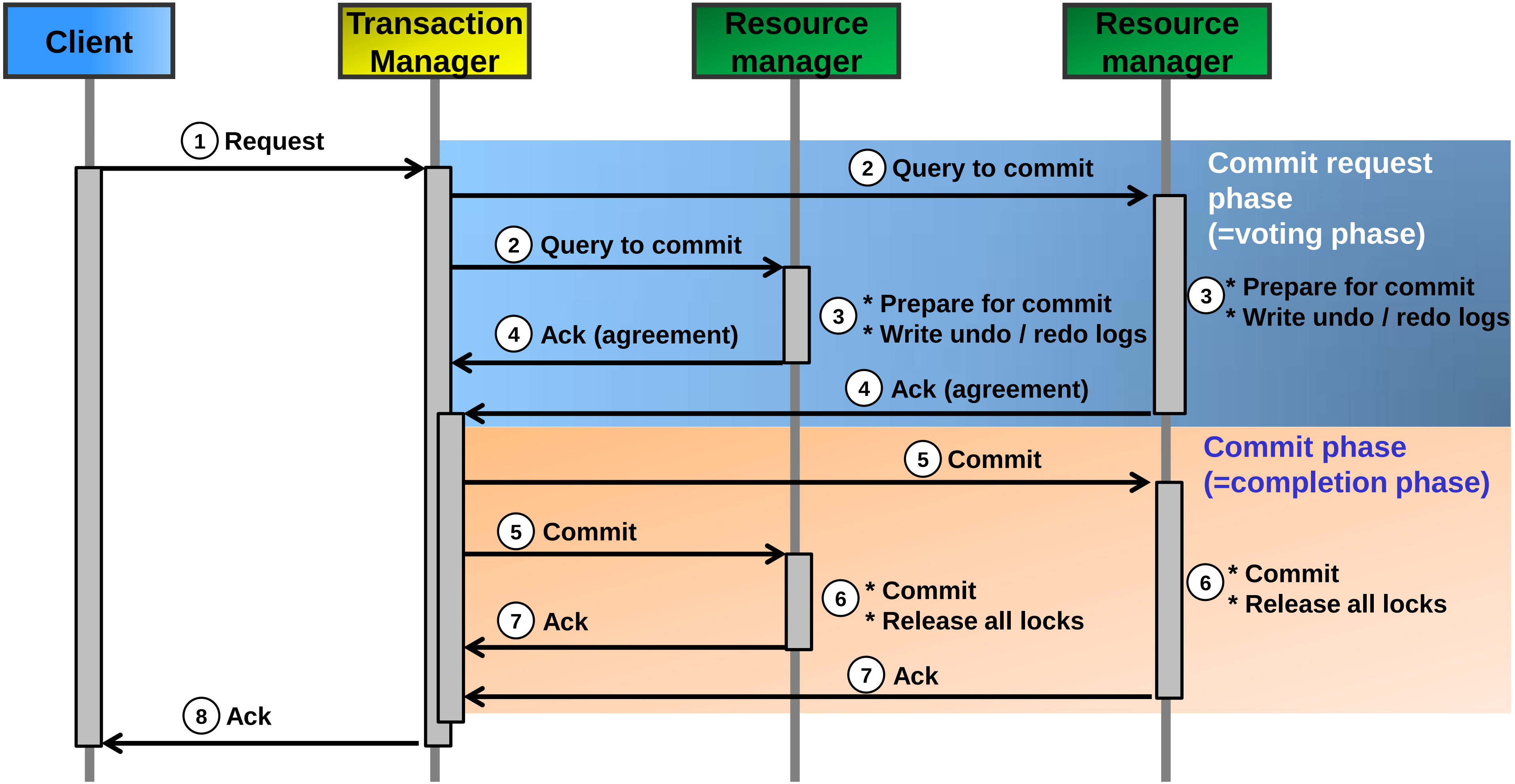
Undo / redo logs:
Undo / redo logs are used by the transaction managers for transactional access to their respective DB.


Client interface for distributed transactions
Transaction Manager (Coordinator)
Resource manager (cohort)
Undo/Redo logs
DB
Resource manager (cohort)
Undo/Redo logs
DB

4. Two-Phase Commit sequence (1/2)

A. Successful transaction:

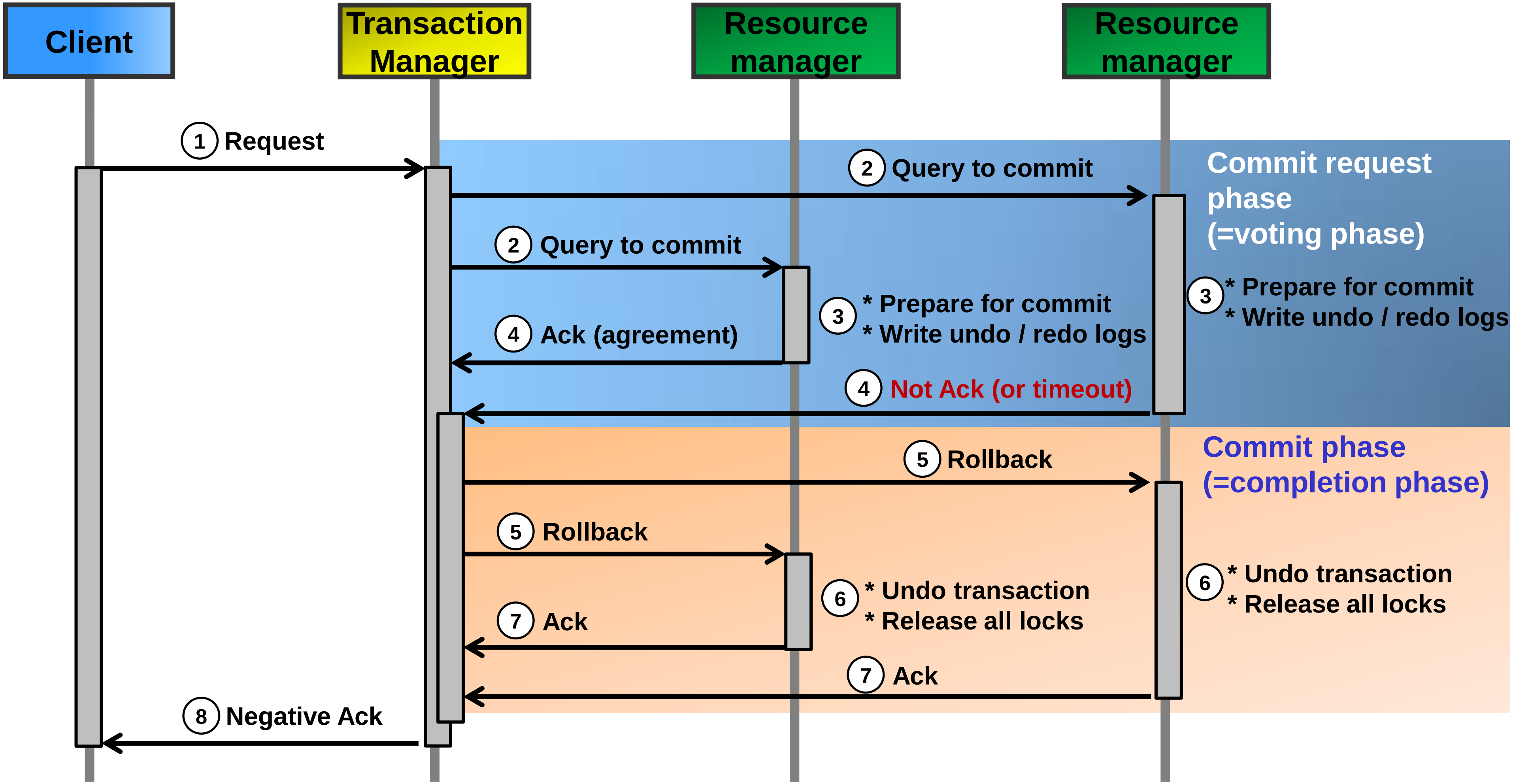
When all transaction managers agree with the commit (vote "yes"), the transaction can be successfully committed.


Client
Transaction Manager
Resource manager
Resource manager
1 Request
2 Query to commit
Commit request phase (=voting phase)
2 Query to commit
3 * Prepare for commit * Write undo / redo logs
3 * Prepare for commit * Write undo / redo logs
4 Ack (agreement)
4 Ack (agreement)
5 Commit
Commit phase (=completion phase)
5 Commit
6 * Commit * Release all locks
6 * Commit * Release all locks
7 Ack
7 Ack
8 Ack

4. Two-Phase Commit sequence (2/2)

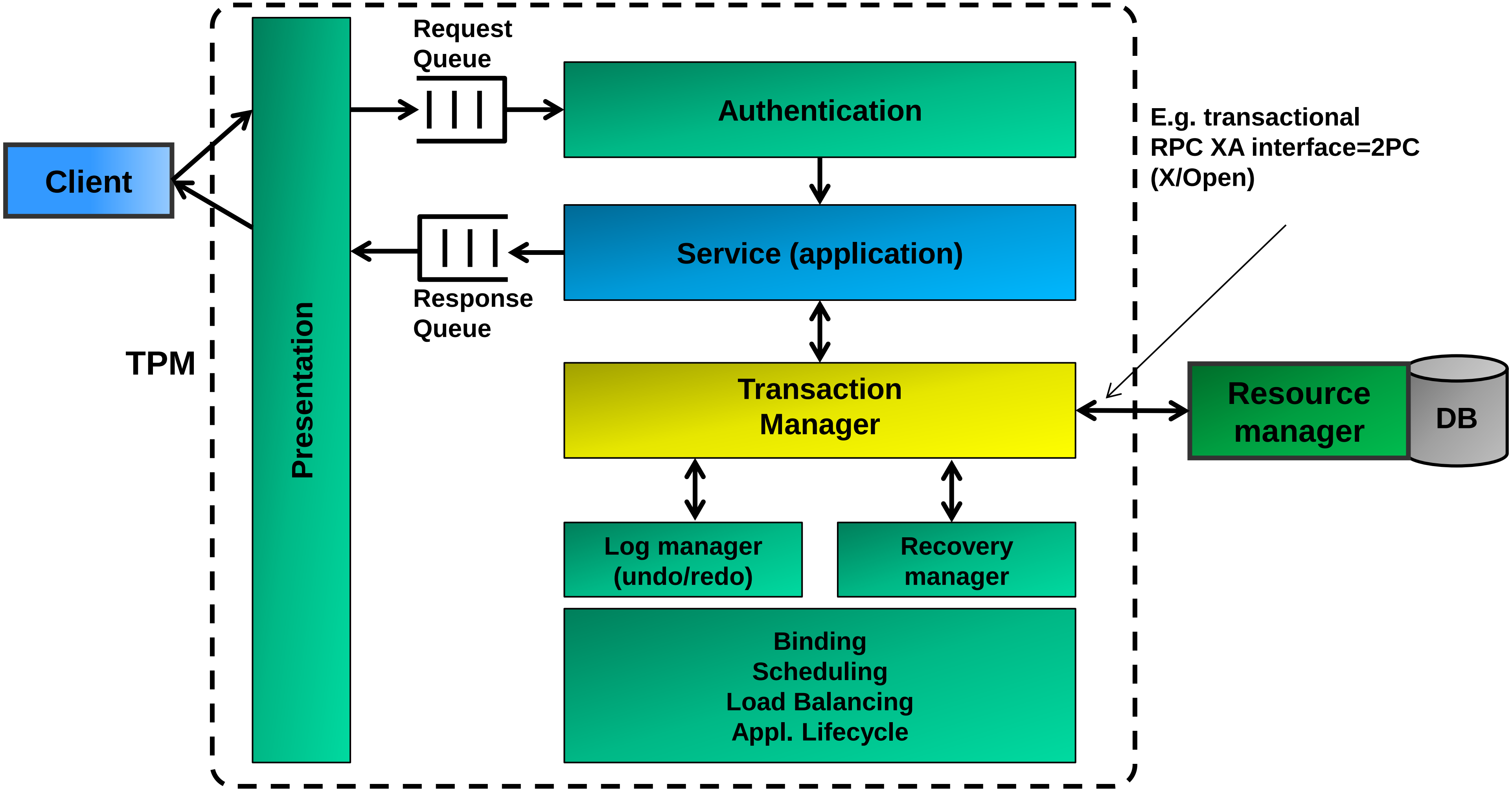
A. Failed transaction:

When either of the transaction manager does not positively acknowledge the transaction, the coordinator sends a rollback command to all transaction managers.


Client
Transaction Manager
Resource manager
Resource manager
1 Request
2 Query to commit
Commit request phase (=voting phase)
2 Query to commit
3 * Prepare for commit * Write undo / redo logs
3 * Prepare for commit * Write undo / redo logs
4 Ack (agreement)
4 Not Ack (or timeout)
5 Rollback
Commit phase (=completion phase)
5 Rollback
6 * Undo transaction * Release all locks
6 * Undo transaction * Release all locks
7 Ack
7 Ack
8 Negative Ack

5. TPM structure

TPMs represent complete systems akin to operating systems capable of running entire applications. Resource managers may be internal or external to the TPM.


Client
TPM
Presentation
Request Queue
Response Queue
Authentication
Service (application)
Transaction Manager
Log manager (undo/redo)
Recovery manager
Binding
Scheduling
Load Balancing
Appl. Lifecycle
E.g. transactional RPC XA interface=2PC (X/Open)
Resource manager
DB