#ASLI ENGINEERING

Distributed Transactions Two Phase Commit

SWIPE

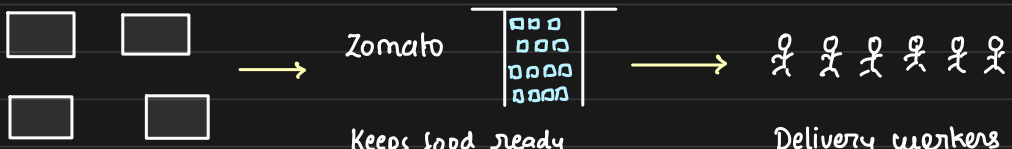
BY

ARPIT BHAYANI

Distributed Transactions: 2 Phase Commit

Distributed Transaction: A transaction that span over multiple physical systems, machines or computers.

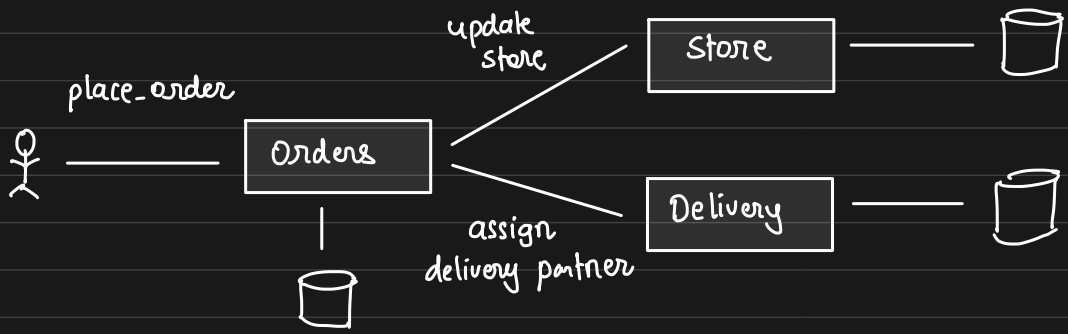
Scenario: 10 min food delivery by Zomato



To guarantee that the food is delivered under 10 mins

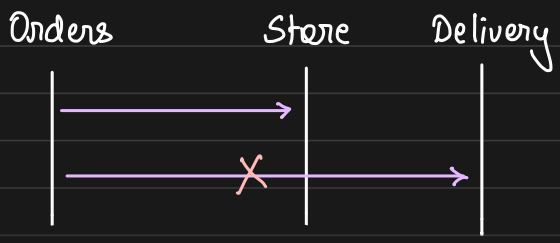
Zomato should accept order only when

1. Food is available in the store
2. Delivery partner is available to deliver

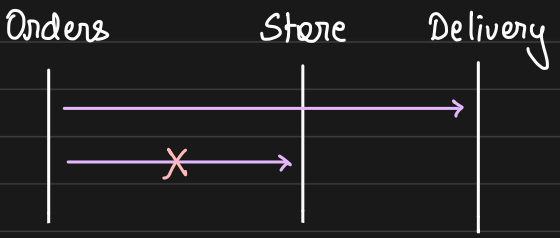


Place order only when there is food available in the store and delivery partner is assigned

Atomicity: In a transaction, either all steps happen or none



Store got the order ✓
Delivery agent not booked ✗



Delivery agent booked ✓
Store did not get the order ✗

If either of them fails, we revert the other

↳ Poor UX to delivery partner
↳ Loss because store spent time heating & packing

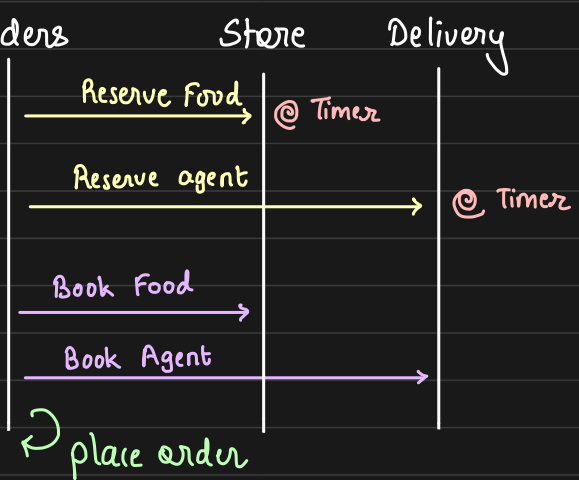
This is a classic case of Distributed Transactions

and we now implement with 2PC

Two Phase Commit :

Split the entire flow in 2 phases

1. Prepare – Reserve
2. Commit – Book



Reservation: if both fails, transaction fails, we abort

if only one succeeds, we cancel the reservation & abort

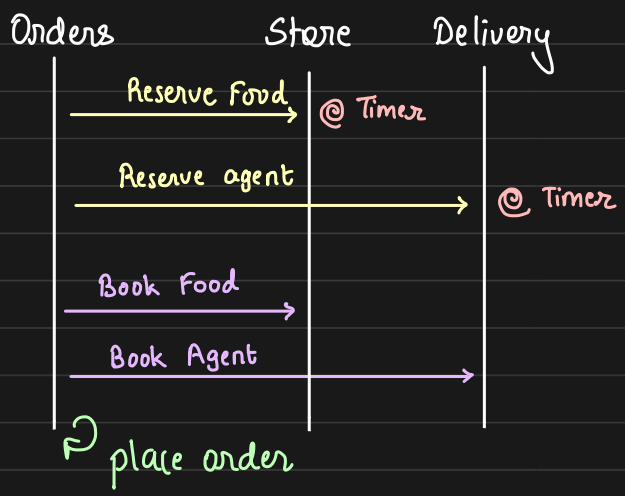
or the timer will auto cancel

if both succeds, we move forward to Commit phase

Commit: if both succeeds, order is placed

if any one fails, we cancel the reservation and abort

if order service fails, timer cancels the reservation



Advantages:
- guarantees ATOMIC transactions
- guarantees Isolation

Disadvantages:
- slow
- prone to deadlock