# Transation Management in Microservice spring Boot

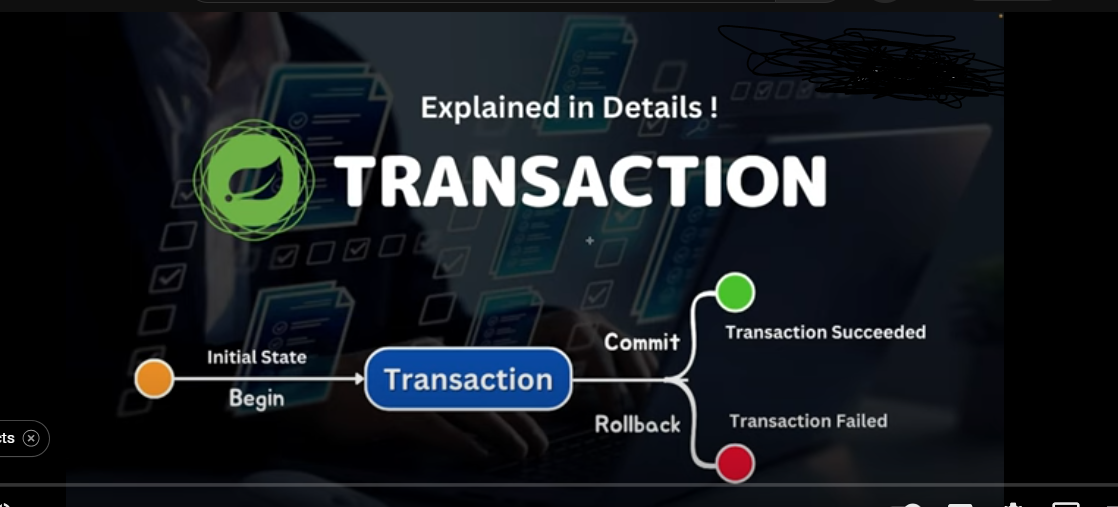
# Overview:

# Scenario : problem statement

Transaction are the backbone of any Enterprise Application.

Whether you are processing the payment or updating the inventory or managing user details ensuring that everything happens consistently is always critical.

In realworld application it is easy to execute multiple operation in single method or in a single usecase. But what will happen if something goes wrong . How we will ensure one failure does not break entire system . That’s where transaction management comes into the picture.



Here we will dive deep into springboot transaction manangement .exploring how it works and why

Its essential , covering different isolation and propogation examples with usecase.

## USECASE :

Imagine you are working on a ecommerce based project .

You have an “Inventory Service ” which have product table as below :

### ProductDB

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Price | Stock |
| 1 | Laptop | 25k | 10 |
| 2 | Mobile | 10k | 15 |
| 3 | Watch | 1500 |  |

### Order Db

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Pid | Qnt | Amount |
|  |  |  |  |
| 101 | 1 | 2 | 50k |
| 102 | 1 | 5 | 1.25L |
| 3 |  |  |  |

In the above scenario, if a user want to place order to purchase 2 Laptop below are the steps to be performed:

* placing order updating order table
* updating the inventory product db also , so further stock value will become
* update payment service db also

**so place an order we performing 3 db operation .**

further we received second request to purchase 5 laptop

* first place order updating order table
* updating the inventory product db also , so further stock value will become
* Imagine dueto network issue there is a failure in the inventory service to update product stock resulting we cannot perform step3.

But order db have saved record with 102 ID.

Problem , if any user search the product the count shown will be 8.

It means data in order service and inventory service is not consistent.

So need is not to keep unused data it means anything going wrong in my flow I should completely rollback it.

Also if all flow is successful commit the changes .