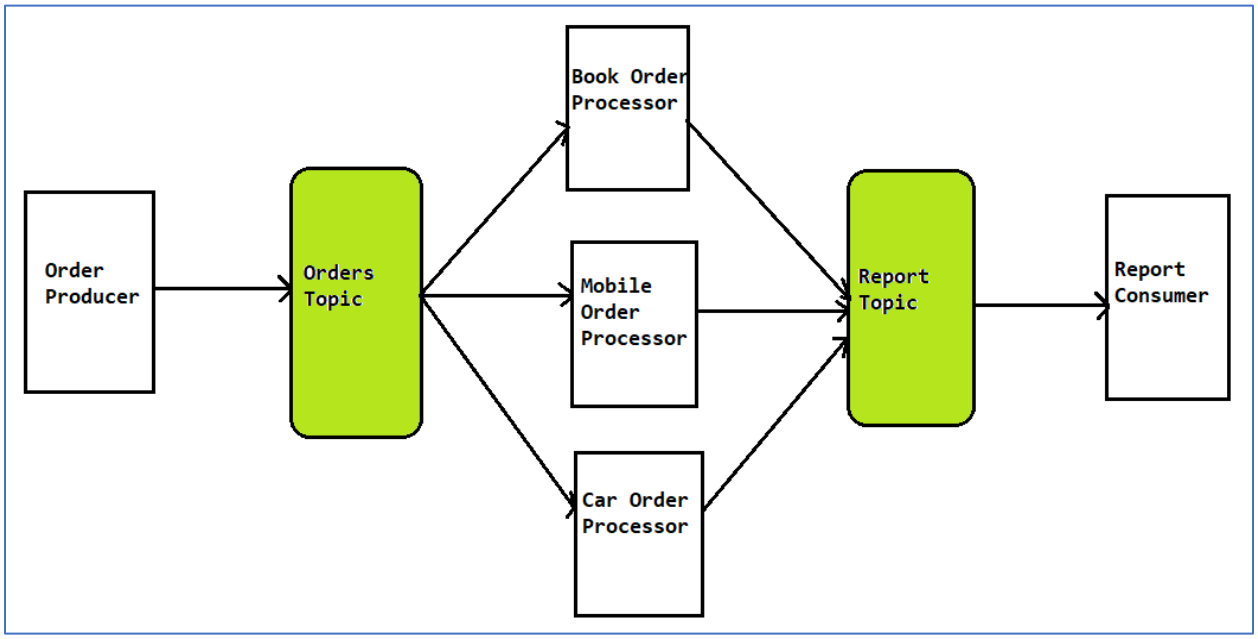
Flipkart Problem Statement

Flipkart is one of the most popular web portals that allows users to place online purchase order for books, mobiles & cars. Please find below the purchase order details:

Order Type	Order Details
Book purchase	id, book title, book author, price, order placement date, customer name, customer email, customer mobile
Mobile purchase	id, mobile vendor, model, price, order placement date, customer name, customer email, customer mobile
Car purchase	Id, car vendor, model, color, price, order placement date, customer name, customer email, customer mobile

The order processing is different for every order type. Hence, we need separate order processor for each order type. You need to solve this case study as desktop application i.e. using Core Java.



Solution:

1. Create a Kafka cluster with a topic ‘FlipkartOrdersTopic’ having separate partition for each order type. Make sure you keep at least one replica of every partition so that failure scenario can be handled.
2. Purchase order producer is continuously pushing orders into the topic & depending upon the order type, the message should automatically route to respective partition. Here, you need to write custom serializer & custom partitioner.
3. Create a consumer group having 3 consumers each one is consuming messages from the respective partition.
4. Consider all 3 consumers as Kafka stream processors where you need to generate aggregated values of all types of orders i.e. "Books_order_count": 230, "Books_total_transaction_amount": 2500000, "Mobiles_order_count": 410, "Mobiles_total_transaction_amount": 9000000, "Cars_order_count": 57, "Cars_total_transaction_amount": 65400000. Push these values into “ReportTopic”.
5. Write Report Consumer that prints aggregated values of all types of orders.