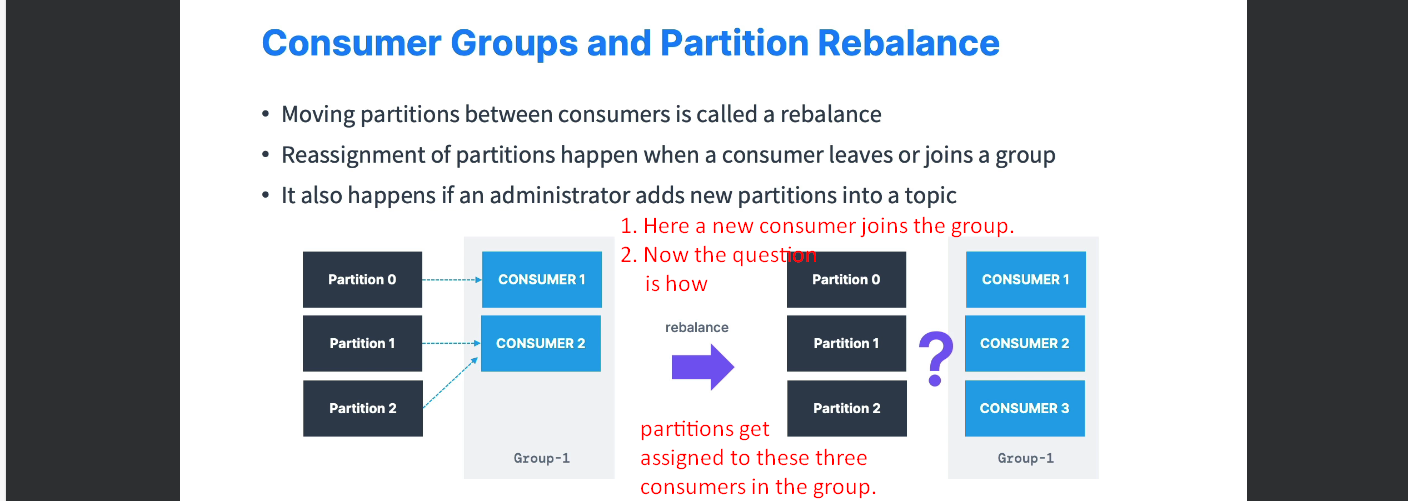
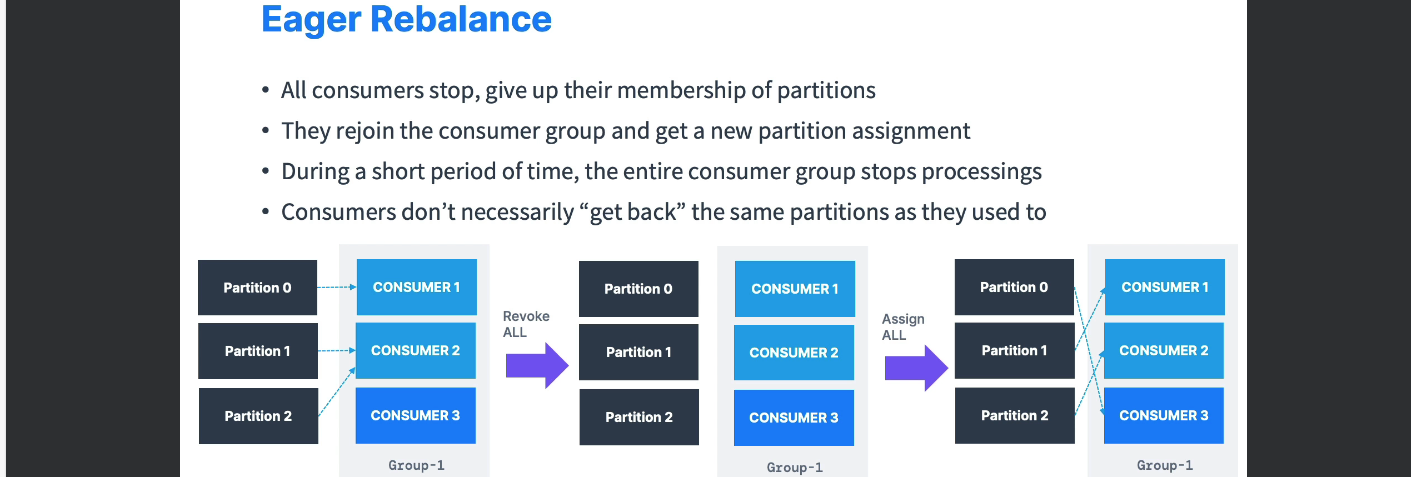
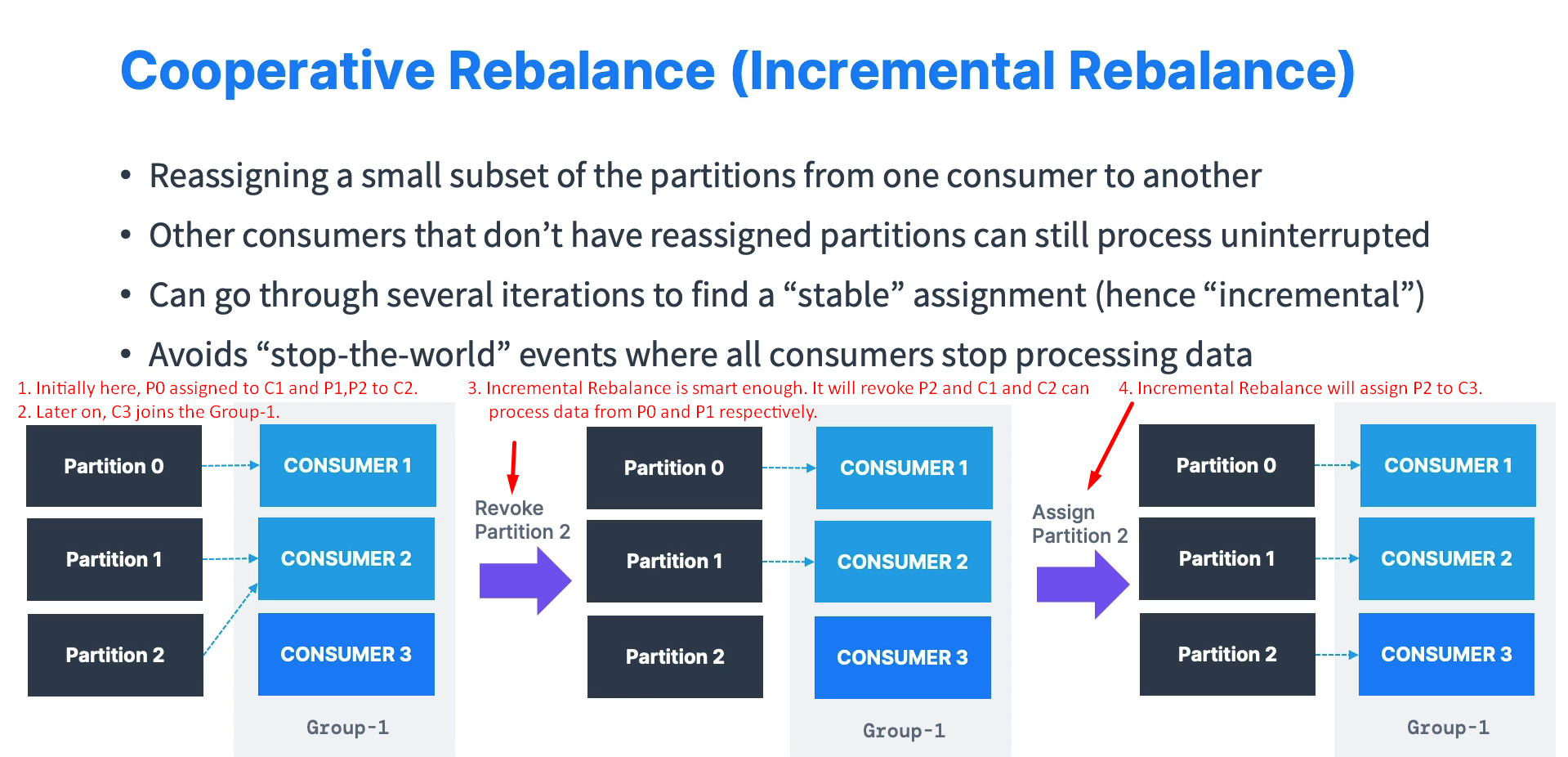
1. Depending on the partition assignment strategy, the output can be different.
2. **Eager Rebalance**:
   1. Default Behavior.
   2. As soon as a new consumer joins the group, all the existing consumers will stop (that is called eager) and they will give up their partition membership.
   3. Then they will rejoin the same group and get the partition assignments.  
      This is quit random means the order of partition assignments may be completely different this time.
   4. For a short period of time, the entire consumer group will stop processing and it is called “**Stop the World Event**”.
   5. Two problems that is why we don’t want “Stop the World Event”.
      1. We may not get same partitions back.
      2. If getting same partition, then the consumer must not have stopped reading.
   6. 
3. Therefore, we have **Cooperative Rebalance** which is quit recent in Kafka also called **Incremental Rebalance**.  
   
   1. Instead of reassigning all the partitions to all the consumers, it reassigns a subset of partitions from one consumer to another.
   2. In the above slide, C1 and C2 will keep on processing data from P0, P1 even during rebalancing process.   
      So, this is less destructive as it allows us to keep on reading from P0, P1.
4. How to use Cooperative Rebalance?
5. A close-up of a text

   Description automatically generated
6. Before going into practice, lets discuss about a topic “Static Group Membership”.   
   Actually, when a consumer leaves its group, the Kafka says okay **all consumers** must consumer **all partitions**.
7. A screenshot of a website

   Description automatically generated