Linkfire Data Engineer Technical Task:

Task # 1:

What partitioning and clustering keys would you consider for storing data in cassandra?

Solution:

I would consider “sessionToken” as the partitioning key and “timestamp” as the clustering key. The reasons for the selection are as below.

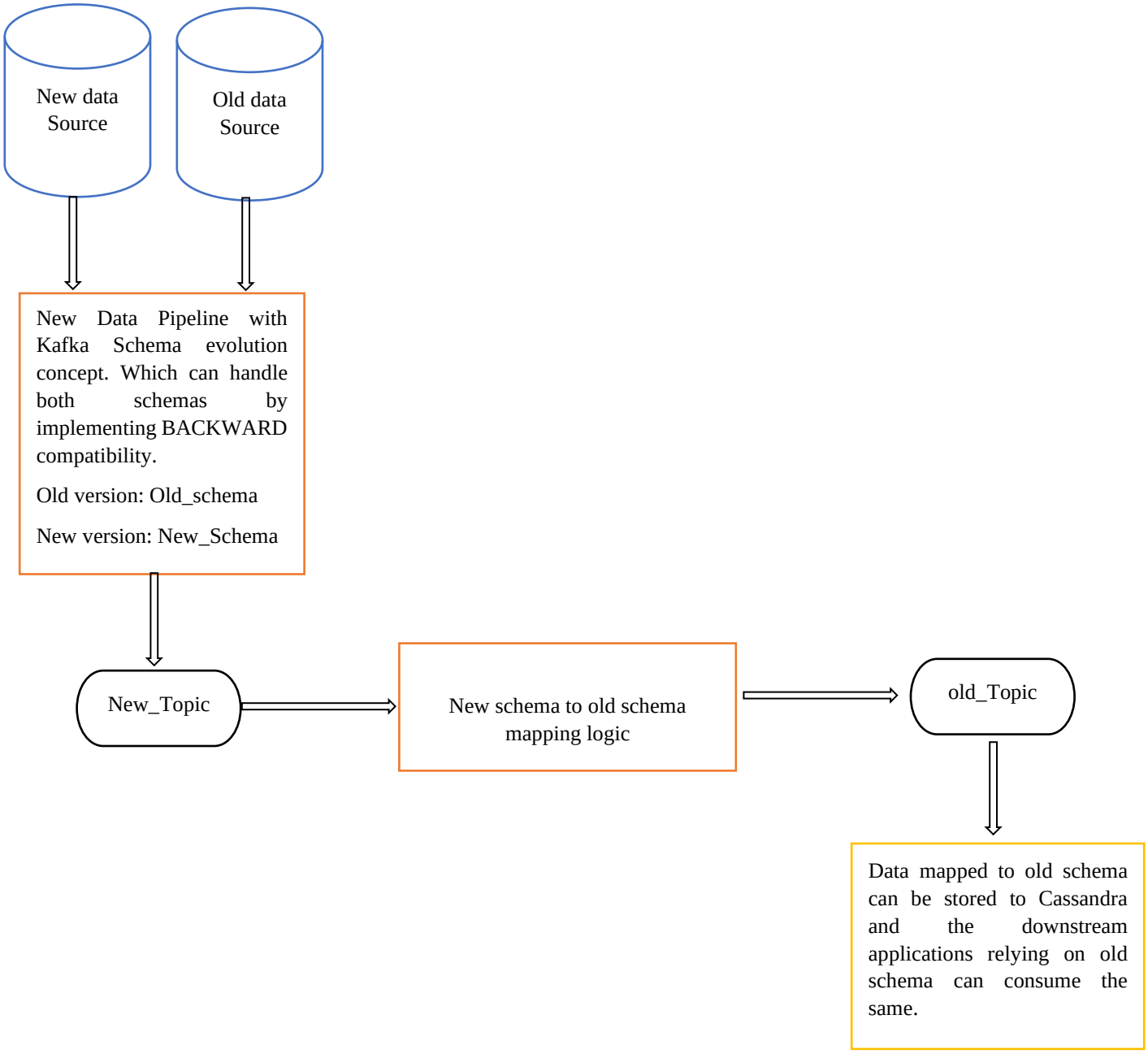
- Since we perform lookups on “sessionToken”, it will be efficient to use the same as partition key. Because Partition Key is responsible for data distribution across nodes. By keeping “sessionToken” as partition key all the rows having same sessionToken will be stored in same node, which makes the lookup efficient.
- The Clustering Key is responsible for data sorting within the partition. making “timestamp” as clustering key helps to efficiently fetch the recent data.
- sessionToken and timestamp is a good combination as a PrimaryKey.

Task # 2:

Line out a strategy to get rid of the old setup while keeping downstream applications running uninterrupted.

Solution:

- The goal is to get rid of the old pipeline, which means the new data pipeline should be capable of handling old and new schema data.
- We can make use of Kafka Schema Evolution concept with BACKWARD compatibility, such that consumers using the new schema can read data produced with the old schema as well.
- Since many applications rely on the old pipeline's output and its schema. The implementation of new data pipeline can cause application interruption, because of the data schema change. To resolve this issue, we may need to implement a logic to map data from new schema to old schema and the same can be utilized by the downstream applications relying on old schema data.


New data Source
Old data Source
New Data Pipeline with Kafka Schema evolution concept. Which can handle both schemas by implementing BACKWARD compatibility.
Old version: Old_schema
New version: New_Schema
New_Topic
New schema to old schema mapping logic
old_Topic
Data mapped to old schema can be stored to Cassandra and the downstream applications relying on old schema can consume the same.

Task # 3:

For most strategies, it would be necessary to map data from one schema to the other for a transition period. Check out the python module below that forwards records from a kafka topic to another (execution and configuration is handled outside), and modify it to map data from one format to the other (pick one direction) based on the two examples. It is not the purpose of this challenge to have a working application.

Solution:

- Schema mapping logic from new schema to old schema is coded in python and the same is available in python module “schema_mapping.py”.
- Also, I have coded a working test application just to represent the working of mapping logic. The test application consists of a kafka producer, which sends data with new schema to “new_topic”. Logic to consume the data from new_topic, mapping logic and pushing the mapped data to old_topic is written in the module kafka_to_kafka.py.