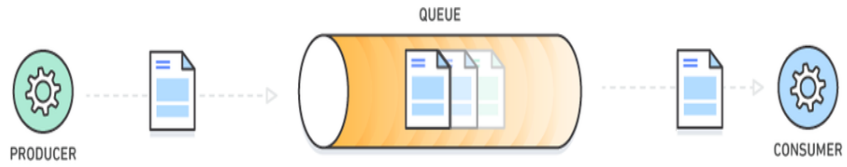
Scenario – Created new Kafka based streaming architecture and migrate key IDResolver activities to AWS Lambda in Java

- As IDResolver is essentially a hub for bringing together information from internal and external services and calculating a “lending answer” separating these into a streaming architecture will reduce latency.
- The following actions will be moved to AWS Lambda:
  - Retrieving Card Payment Information from the 3rd Party Gateway
  - Retrieving Postcode Data from the 3rd Party Gateway.
  - Retrieving the Credit Search Data available internally.
  - Finally, the resource intensive “lending answer” calculation.
  - These implementations will be written in Java, as opposed to the current php implementation.
- Kafka will be used as the event streaming system:
  - Kafka queueing will be used, rather than pubsub in order to be highly scalable.
  - Kafka Streams API will be used to transform data, mainly in terms of joins and aggregations to service the lending answer calculation.
  - Kafka is excellent for retaining data and being able to reprocess it later.



Queues, messages and topics are generally represented by these symbols, it is also good to represent producers of messages and consumers of messages