Imagine that you are working on a data platform solution for a retail channel engagement customer, and your team is looking for the next important metric to deliver to end enterprise (the merchants/retailers).

The customer provided an insight that the most frequently requested metric by all retailers is an average of how long users spend on their flyers each day. This would allow them to measure the effectiveness of the flyer, as well as having an understanding of what resonates more with their consumers.

Part 1 - Algorithm

Background information for this exercise:

* Assume that the channel engagement platform is a mobile or tablet application.
* Flyers have unique identifiers
* Merchants have unique identifiers
* Merchants have multiple flyers

You have been given a sample dataset with the following schema: timestamp, user\_id, event, flyer\_id, merchant\_id

There are a number of events which the mobile app collects as the shoppers the app. These are the 'event" types in the preceding schema.

* flyer\_open - Whenever a flyer is opened by a user, to see the flyer contents
* item\_open - Whenever an item is opened by the user, to see the item details
* list\_flyers - Whenever a listing of the flyers is shown
* shopping\_list\_open - Whenever a user opens their shopping list
* favorite - Whenever a user adds a merchant to their favourites.

Your task is to:

* 1. Compute the average time on flyer per user.
  2. Next, generate an output that will back a Business Intelligence (Bl) report that will be shared with our merchant partners.
  3. Explain how your algorithm scales for:
     1. 1 Million Events (~10 MB of data)
     2. 1 Trillion Events (~10 TB of data)

To write your algorithm, you may use any of: Python, Java.

Part 2 - Infrastructure

During your work on Part 1, you discover that the mobile app is not sending all the necessary user behaviour events. You discuss it with the customer and they agree to collect and send the required, new user behaviour events.

Assume that:

* You are using a distributed pub-sub style technology cluster such as Apache Kafka/AWS Kinesis/Azure EventHubs
* Other groups at the customer will need streaming access to the data

Requirements:

* User behaviour event data needs to be consumable from the pub-sub Cluster within 120 seconds of being created by the mobile application
* Data Analysis team will need access to an environment to perform on-demand adhoc analysis
* Data Science team will need cleansed and backfilled data both on stream and batch capacity to build, train and re-train their models

Your task is to:

* Design a workflow to move the user behaviour event data from the application to a backend and provide insights into the data pipelines that you foresee
* Explain how the workflow would provide the data to the batch process in Part 1 Algorithm.
* Explain any adaptations that your work from Part 1 - Algorithm would need to work as a streaming process.
* Highlight any important design decisions you make, and describe briefly why you made those decisions. Important parts to address include:
  + Latency
  + Scalability & Data Volume
  + Robustness
  + Failure Modes
  + Delivery Guarantees
  + Technology selections