Q. Dataset A is formed by storing streaming data from all the video camera sensors from the edge. Each row in Dataset A is an event. There are around 10000 events per seconds.

You can assume Dataset B, is a static reference table that is unchanged.

Suppose your PM wanted you productionize this architecture, by considering the following
requirements:
1. The PM wants to build a dashboard to visualize the results by joining Dataset A and B without any duplicates.
2. The join results of dataset A and B shall be available as soon as the events in Dataset A has been published.
3. Duplicate events may occur upstream due to retries, etc.

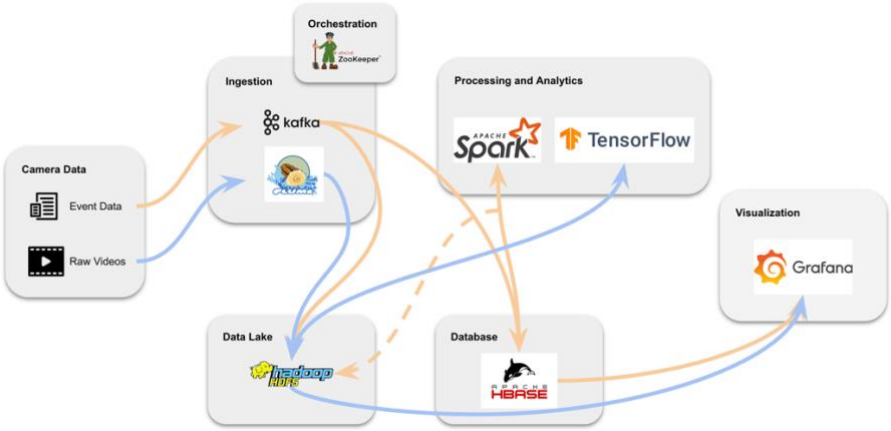
You are required to design an architecture to describe how you will be proposing the store, process and finally the considerations when accessing data. To score full points, you need to detail the questions you will ask the end user and considerations or assumption you will make in your design.

You can also assume the environment that you are using, i.e. Hadoop, Azure, GCP and etc.

You will also explain the tech stacks and storage tech stack that you are using why this tech stack shall be used.

- Ingest Dataset A (camera data) using **Kafka** (using quixstream for python script). Ingested data can be stores in **HBase** database.
    - Kafka can handle high velocity streaming ingestion.
    - Use zookeeper to be the orchestrator, managing the worker nodes.
    - HBase provides strong consistency.
    - Kakfa and HBase can also prevent duplicated data from being ingested using det_oid as the unique identifier.
- Reference table Dataset B can also be stored in **HBase** for low latency join processing.
- Process using **Spark Streaming** (join table A and B, extract top x items) for real time processing.
- Store in HBase (can store RDD) for fast querying. Convert to parquet and store into **HDFS** for future batch processing requirements.
- Access the data for visualization using **Kafka exporter + Prometheus + Grafana**

- If there is also a need to ingest raw video footage, we can use **Flume** to store the .mp4 (or .png/.jpeg if it's just a frame image) into **HDFS** which can then be processed and analysis through Spark and eventually visualized on Grafana.



Questions to ask:

- Is it also necessary to access the actual footages of the items? (will need Flume to help with the ingestion)
- Can the dashboard be refreshed near real-time (minimally once every 5-10 seconds) instead of real time (every second)
- My assumption is that it is more important to continuously capture accurate data (strong consistency) rather than continuously updates that may be inaccurate (high availability). But is there a case where high availability is more important? (e.g being notified to investigating a suspicious object wrongly is more important than not being notified at all). [consider switch to Cassandra]
- On premise or cloud deployment?
- Will there be a lot of people accessing the dashboard? (managers or only or all staff)