**Technical Design Document**

**Sales Statistics Dashboard**

**Submitted By:  
Group 8**

Ramanan Thayalan - 200565744

Niki Patel - 200545936

Anoushka Rastogi - 200593018

Ishan Patel - 200528548

Hellyben Vaghasiya – 200544282  
  
Sahil Kashyap - 200569298

**Project Name: Sales Statistics Dashboard**

**Coding Languages and Tools**

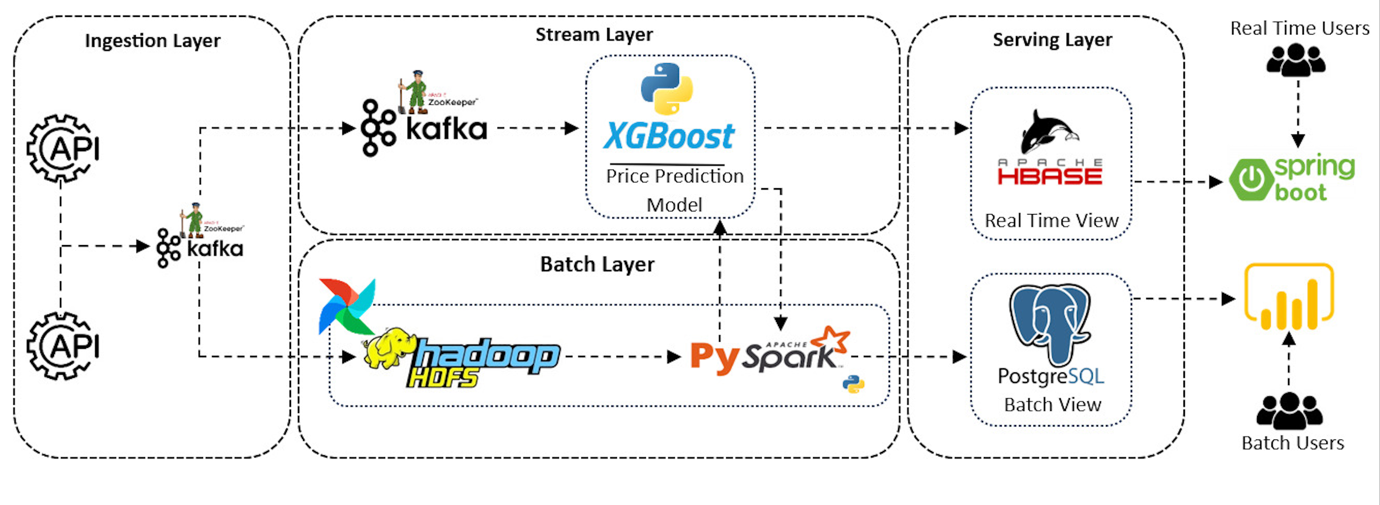
* **Programming Languages**: Python and SQL
* **Frameworks and Tools**:
  + **Apache Spark**: For large-scale data processing and transformation.
  + **Apache Hadoop**: As the data lake for storing large datasets.
  + **Apache Airflow**: For orchestrating complex data pipelines.
  + **PostgreSQL**: As the data warehouse to store processed and transformed data.
  + **Apache Kafka**: For real-time data simulation.
  + **Power BI**: For visualizing the processed data in a user-friendly dashboard.

**Infrastructure/System**

* **Deployment Infrastructure**: The project is deployed on an in-house local system using Docker for containerization and environment management.
* **Data Collection and Processing**:
  + **Historical Data**: Collected from Kaggle and stored in Apache Hadoop as a data lake.
  + **Real-Time Data**: Simulated using Apache Kafka and processed through Apache Spark for real-time analytics.
* **Data Storage**: After transformation by Apache Spark, data is stored in PostgreSQL, a robust data warehouse solution.
* **Visualization**: Power BI is used to create dashboards that visualize the processed data, providing insights into sales patterns and performance.

**Data Flow**

1. **Data Collection**:
   * Historical data is obtained from Kaggle and uploaded to the Hadoop data lake.
   * Real-time data is simulated via Apache Kafka and also sent to the Hadoop data lake.
2. **Data Processing**:
   * Apache Spark processes the data from Hadoop, performing necessary transformations.
   * The transformed data is then stored in the PostgreSQL data warehouse.
3. **Data Visualization**:
   * Data from PostgreSQL is pulled into Power BI for dashboard creation and visualization.
   * Stakeholders can interact with the dashboard to explore sales patterns and generate reports.

****

1. **Ingestion Layer**
   * **Tools Used**: Apache Kafka and APIs.
   * **Purpose**: This layer is responsible for ingesting data from various sources through APIs. Kafka acts as a message broker to ensure that the data is streamed efficiently to the subsequent layers.
   * **General Steps**:
     + **Step 1**: Set up Kafka as a message broker to capture and queue data from APIs.
     + **Step 2**: Ensure data is passed from the APIs to Kafka topics for distribution to the Batch Layer.
2. **Batch Layer**
   * **Tools Used**: Apache Hadoop HDFS, Apache Spark, and Airflow (for orchestration).
   * **Coding Language**: Python (with PySpark).
   * **Purpose**: This layer processes and transforms large volumes of data in batch mode, making it suitable for historical data analysis and long-term storage.
   * **General Steps**:
     + **Step 3**: Data from Kafka is stored in Hadoop HDFS for processing.
     + **Step 4**: Use Apache Spark (PySpark) to perform ETL (Extract, Transform, Load) operations on the data. This includes cleansing, aggregation, and transformation as needed.
     + **Step 5**: Orchestrate the ETL process using Apache Airflow, ensuring that all tasks are scheduled and executed in the correct order.
3. **Serving Layer**
   * **Tools Used**: PostgreSQL.
   * **Purpose**: This layer serves the processed data to batch users. The transformed data is stored in PostgreSQL, where it is accessible for further analysis or reporting.
   * **General Steps**:
     + **Step 6**: Load the transformed data from Apache Spark into the PostgreSQL database.
     + **Step 7**: Ensure data integrity and availability for batch users who access the data for reporting or analysis purposes.
4. **Batch Users**
   * **Tools Used**: Power BI (for visualization).
   * **Purpose**: Batch users, such as analysts or business stakeholders, access the data stored in PostgreSQL for generating reports, performing further analysis, or making business decisions.
   * **General Steps**:
     + **Step 8**: Connect Power BI to the PostgreSQL database.
     + **Step 9**: Create dashboards and reports in Power BI, providing users with insights into the processed data.

**Excluded Layers**

* **Stream Layer**: Although present in the diagram, this layer and its associated tools (XGBoost, HBase) are not used in your current implementation. The Stream Layer would typically handle real-time data processing and prediction, which are not part of your project flow.
* **Real-Time Serving Layer**: Similarly, the real-time serving layer and tools like Spring Boot are excluded from your implementation, as your project focuses on batch processing and not on real-time data processing.

**Project Flow**

* **Primary Flow**: Data is ingested via APIs into Kafka, processed in Hadoop/Spark, stored in PostgreSQL, and then accessed by batch users through Power BI.
* **Implementation Language**: The primary coding language for the ETL and processing tasks is Python, especially with the PySpark framework.

**Dataset Size – 1 GB**

* **Historical Data**: Multiple years of daily sales data, which could amount to several terabytes, depending on the size of the business.
* **Real-Time Data**: Generated continuously, the size is dynamic and can grow as more data is simulated.

**Cost Estimate of the Infrastructure**

Given the use of in-house systems and open-source tools (Hadoop, Spark, Kafka, PostgreSQL), the primary costs will be associated with:

* **Hardware**: Depending on the scale of the data, significant storage and processing power will be required.
* **Development and Maintenance**: The cost of developing, maintaining, and scaling the system.
* **Software Licensing**: Although most tools are open-source, Power BI may require licensing depending on the version used.

**Development Process**

The project follows an **Agile development process**, with iterative cycles that allow for continuous integration and testing. Each phase of the project involves specific tasks, tools, and languages:

* **Phase 1: Data Collection**
  + **Tasks**: Gather historical data, set up real-time data simulation.
  + **Tools**: Kaggle (for data), Apache Kafka (for real-time simulation).
* **Phase 2: Data Storage and Processing**
  + **Tasks**: Load data into Hadoop, process with Spark.
  + **Tools**: Apache Hadoop (data lake), Apache Spark (processing).
* **Phase 3: Data Transformation and Storage**
  + **Tasks**: Transform data in Spark, store in PostgreSQL.
  + **Tools**: Apache Spark, PostgreSQL.
* **Phase 4: Data Visualization**
  + **Tasks**: Create dashboards, visualize data.
  + **Tools**: Power BI.
* **Phase 5: Deployment**
  + **Tasks**: Deploy the dashboard, ensure system stability.
  + **Tools**: Docker (for containerization).