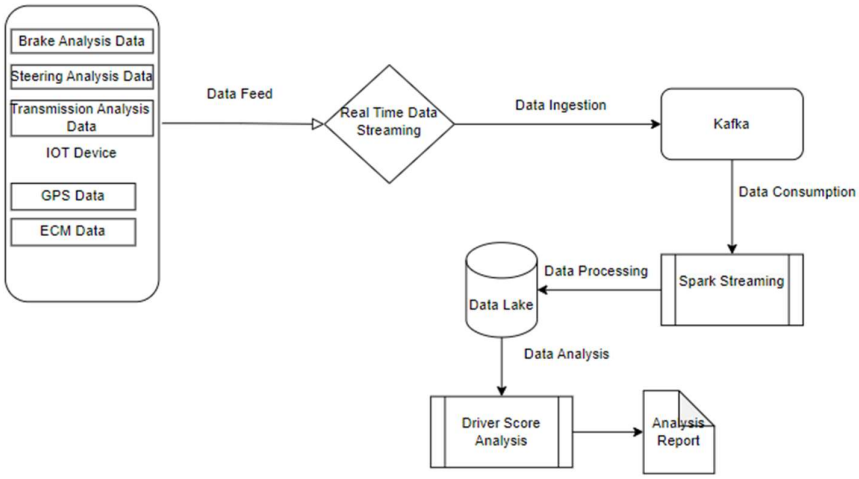
Architecture



Architecture Overview

1. Data Acquisition & Ingestion:

- **IoT Devices:** Vehicles equipped with telematics devices continuously stream data, including GPS coordinates, speed, acceleration, braking patterns, engine status, etc.
- **Kafka for Data Ingestion:**
    - **Producer:** Telematics devices act as producers, sending real-time data to Kafka topics.
    - **Topics:** Data is categorized into different Kafka topics based on the data type (e.g., GPS, Speed, Acceleration).
    - **Consumer:** A Spark Streaming application consumes data from these Kafka topics in real time for further processing.

2. Data Transformation & Processing:

- **Spark Streaming:**
    - **Data Cleansing:** Spark processes the raw data to remove noise, handle missing values, and correct inaccuracies.
    - **Feature Engineering:**
        - **Harsh Braking Detection:** Analyzing deceleration patterns to detect instances of harsh braking.
        - **Idle Engine Detection:** Identifying periods where the engine is on but the vehicle is stationary.
        - **Over Speeding Detection:** Comparing speed data against predefined speed limits.
        - **Right Turn Detection:** Calculating turn angles using GPS data to detect and analyze right turns.
    - **Threshold & Weightage Application:** Configurable logic applies predefined thresholds and weightages to the features, preparing them for scoring.

3. Data Storage & Management:

- **Scalable Storage Solutions:**
    - **S3:** Processed data is stored in as s3 files for both real-time access and long-term storage.
    - **Data Lake:** A data lake architecture can be implemented to manage both raw and processed data, ensuring efficient data retrieval for analysis.
- **Data Consistency:** Delta Lake can be employed to maintain consistency across storage layers, enabling accurate and timely data retrieval.

4. Driver Score Calculation:

- **Scoring Algorithm:**
    - Implement a scoring algorithm in Spark that combines weighted factors into a single driver score. This score is calculated in near real-time as data is ingested and processed.
    - The algorithm is optimized to handle large volumes of streaming data, ensuring performance and scalability.
- **Customization & Flexibility:**
    - The scoring model allows for customization to accommodate different driving contexts or fleet-specific requirements, ensuring flexibility in score calculations.

Deliverables

1. **Architecture Diagram:** This diagram will visualize the entire pipeline, highlighting the flow from data ingestion to driver score calculation.
2. **Running Code:**
    - **Kafka Producer:** Simulates telematics devices sending data to Kafka.
    - **Spark Streaming Job:** Consumes data from Kafka, processes it, and calculates driver scores.
    - **Scoring Module:** Implements the driver scoring algorithm.
    - **Output:** Driver scores are outputted to the console or terminal for verification.

Tech Stack

- **Data Ingestion:** Apache Kafka
- **Data Processing:** Apache Spark (using Python)
- **Data Storage:** S3 and Delta Lake
- **Scoring:** Custom Spark-based algorithm