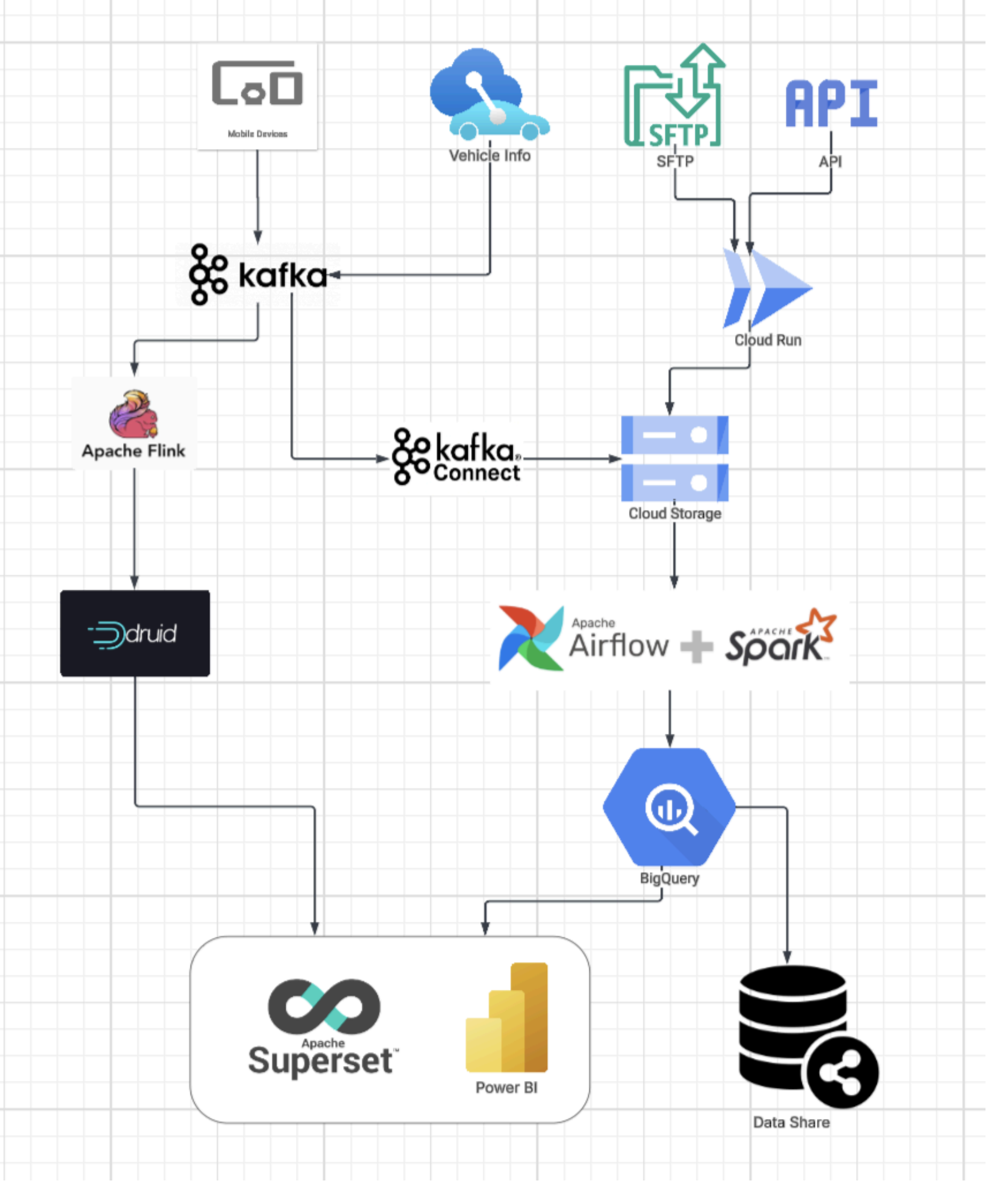
Part 1: Architecture Design

This document presents the architectural design for a logistics delivery company, covering the entire data pipeline, from initial data ingestion to the final report generation. The architecture is designed to support high data throughput, capable of processing approximately 100,000 events per minute (about 1,667 events per second) from applications and vehicles, as well as handling multiple daily file uploads. The proposed architecture is divided into two key components: **real-time analytics** for immediate event processing and monitoring, and **batch analytics** for in-depth, scheduled processing and reporting based on larger data sets. This system leverages a mix of open-source technologies and services provided by Google Cloud Platform.



Real-time Analytics

Data events from mobile devices and vehicle systems are streamed into **Kafka**, a reliable distributed streaming platform, for ingestion. For real-time analytics, the data is processed using **Apache Flink**, a high-performance stream processing engine, and then stored in **Apache Druid**, a real-time analytical database optimised for fast, interactive queries. **Apache Superset** connects to Druid to provide the Operations team with dynamic, real-time dashboards. In parallel, raw event data from Kafka is ingested into **Google Cloud Storage (GCS)** using **Kafka Connect**.

Batch Processing

Data received via **SFTP and APIs** is ingested into **Cloud Storage** using **Cloud Run functions**. From there, **Apache Airflow** orchestrates and schedules **Spark jobs** to process and refine the data before loading it into **BigQuery**, which serves as the central data warehouse for analytical needs. Then **BigQuery** is integrated with visualisation tools like **Power BI** to generate charts and dashboards for actionable insights.

Data Share

Internal dashboards for ops

Using **PowerBI** or alternative visualisation tools (such as **Tableau**, **Looker, Superset**) to deliver reports and dashboards to internal stakeholders/customers.

Push-Based Delivery

- **SFTP**: Automate weekly data exports from BigQuery to secure **SFTP** using data transfer jobs. Providing customers with credentials and a schedule to retrieve data, ensuring encryption for security.
- **API**: Building a RESTful API using Cloud Functions (**Python + FastAPI**) to query BigQuery and deliver data. Schedule weekly pushes via cron jobs, sending JSON or CSV payloads to customer endpoints. Implement authentication (e.g., OAuth) and rate limiting.
- **Dataset Share**: Using BigQuery authorised views to share specific datasets. Schedule updates with **Data Transfer Service** and grant customer access via IAM roles, enabling them to query data directly every week/day.

Customer-Managed Integrations

- **Customer Pull**: Providing authorised views or shared datasets in BigQuery. Sharing connection details (e.g., service account keys) and documentation for customers to query data using their tools (e.g., BI platforms) on a weekly schedule.
- **Bring Their Platform**: Allowing customers to connect their platforms (such as Tableau and Power BI) to BigQuery using ODBC/JDBC drivers or connectors, and setting up recurring queries or export scripts for weekly execution while managing dataset permissions and updates.

Security

- Securing data stored in Cloud Storage and BigQuery using Google Cloud’s built-in encryption.
- Ensuring data security during transmission by enabling TLS encryption for Kafka, Kafka Connect, and API connections.
- Protecting sensitive data in BigQuery or Druid by applying dynamic data masking.
- Controlling API access using OAuth 2.0 or service account keys, and regularly rotating those keys to maintain security.
- Limiting access to services such as BigQuery, Cloud Storage, and Cloud Run by configuring appropriate IAM roles and permissions.
- Granting service accounts only the minimum permissions required to perform the task.
- Implementing row-level security.
- Leveraging GCP's Secrets Manager for secure storage of access credentials and security tokens.

Part 4: Strategic Write – Team build out

1. What team members would you hire next and why?

I’ll start by bringing on a **Data Analyst/Business Analyst**. They'll be crucial for delivering immediate value by creating the reports our customers/stakeholders need.

Once those initial reporting needs are met and we have a clearer picture of the workload and management's priorities, I'll then focus on expanding the team with **Data Engineers**, **Data Scientists** and **Data Infrastructure Engineers,** hiring based on our evolving project requirements.

2. What are your top 3 priorities in your first 90 days

1. Understanding of our business model, operational workflows and company culture.
2. Developing an in-depth understanding of the existing infrastructure and upskilling where needed.
3. Identifying areas where I can contribute early value.