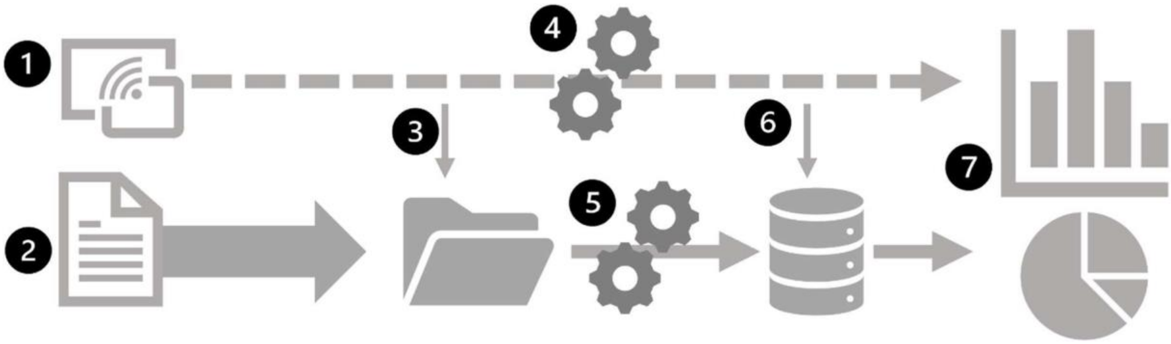
SPA Assignment –I Group –07

Q1. You need to introduce the client with several examples where streaming analytics has already been used. For that purpose, you need to formulate one example of each type of Real-time application scenarios mentioned in the white paper.

- The example should be different from the ones discussed in the document
- Narration should have
  - brief description of the use case scenario
  - short explanation about how it can leverage streaming analytics solutions / platforms
  - justification about how it falls under the particular category

Brief Description

- ***Batch processing***, in which multiple data records are collected and stored before being processed together in a single operation.
- ***Stream processing,*** in which a source of data is constantly monitored and processed in real time as new data events occur.



1. Data events from a streaming data source are captured in real-time.
2. Data from other sources is ingested into a data store (often a *data lake*) for batch processing. If real-time analytics is not required, the captured streaming data is written to the data store for subsequent batch processing.
3. When real-time analytics is required, a stream processing technology is used to prepare the streaming data for real-time analysis or visualization; often by filtering or aggregating the data over temporal windows.

4. The non-streaming data is periodically batch processed to prepare it for analysis, and the results are persisted in an analytical data store (often referred to as a *data warehouse*) for historical analysis.
5. The results of stream processing may also be persisted in the analytical data store to support historical analysis.
6. Analytical and visualization tools are used to present and explore the real-time and historical data.

Case Studies:

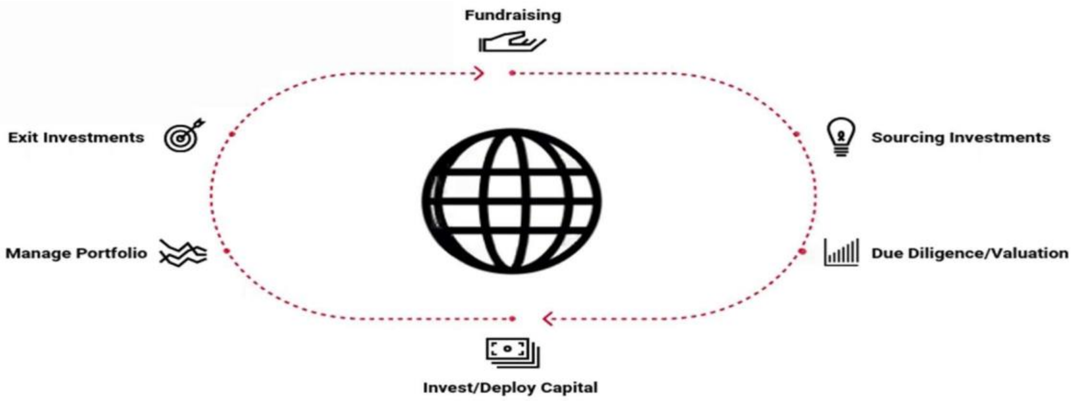
Use Case 1: Stock Pricing Alert

Brief description of the use case scenario

Real-time analytics can help to identify trends of manipulation in markets, especially insider trading and price manipulations that are done to gain profit in real time. In stock trading, it's common to gain profit by using dubious methods, such as insider trading or the artificial deflating/inflating of stock prices. Real-time analytics can be used to collect data from Twitter streams, newsfeeds, company announcements, and other external data streams to identify potential attempts to manipulate the market. Most traders put their trades within stop limits and if stock price is to breach that price, user should get the alert.

- **Short explanation about how it can leverage streaming analytics solutions / platforms**

Develop an event-driven trading platform using stream analytics so that it supports the real time data and high-speed trading which provides clients with knowledge to take decisions appropriately. So that companies can provide recommendations to consumers. Make ratings and analyse data available across systems, subscribers, and customers.



- **Justification about how it falls under the particular category**

  In capital markets, real-time data is a crucial tool to understand what is happening right now and take appropriate action. For example, a streaming analytics model might watch market data streams with instructions to take specific action if certain conditions are met. For example, if the spread between two stocks deviates by more than a certain percentage in any five second period, trade the stocks immediately.

Use Case 2: Diabetes Management

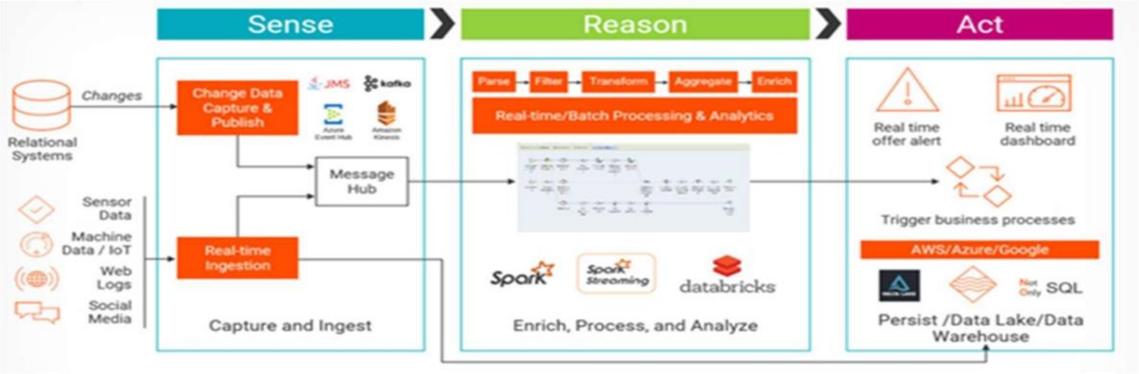
- **Brief description of the use case scenario**

Diabetes is one of the greatest global health threats. Individuals with diabetes take the brunt of it. They see their doctors for a few minutes every few months, so it's largely up to them to manage their conditions—finding a balance between not having enough sugar in their blood and having too much.

- **Short explanation about how it can leverage streaming analytics solutions / platforms**

  For example, if a patient's heart rate increases by five percent or blood pressure drops by 10 percent, those actions can trigger an alert for a nurse or a doctor to take immediate action. The solution can scale out horizontally and vertically to handle petabytes of data while honouring business service-level agreements.

- Informatica Data Engineering Streaming offers a **"sense-reason-act"** framework for real-time streaming analytics. The framework provides end-to-end data-engineering capabilities to ingest real-time sensor data coming from medical devices, apply enrichments on the data in real-time or in batches, and operationalize the actions on the data in a single platform using a simple and unified user experience.



- **Justification about how it falls under the particular category**

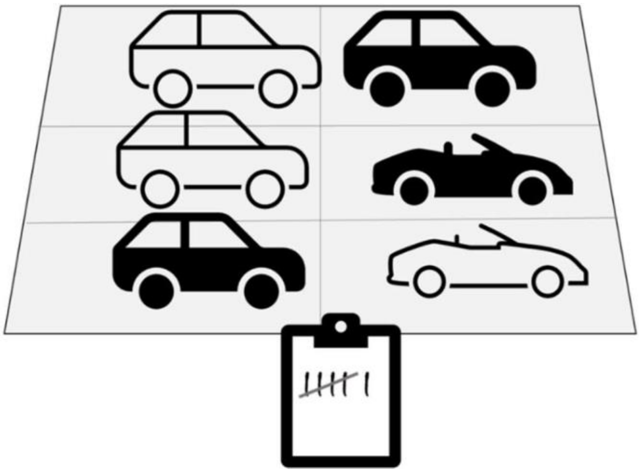
Healthcare providers can develop mobile personal assistant apps powered by streaming analytics that provide real-time actionable glucose insights and predictions for individuals with diabetes, helping to make it easier for them to manage the disease. Streaming analytics can help to analyse data from wearable

Devices (Apple watch) and use machine-learning models to assess the risk of patients' glucose levels falling outside the safe threshold.

Use Case 3: Car Study Systems:

- **Brief description of the use case scenario**

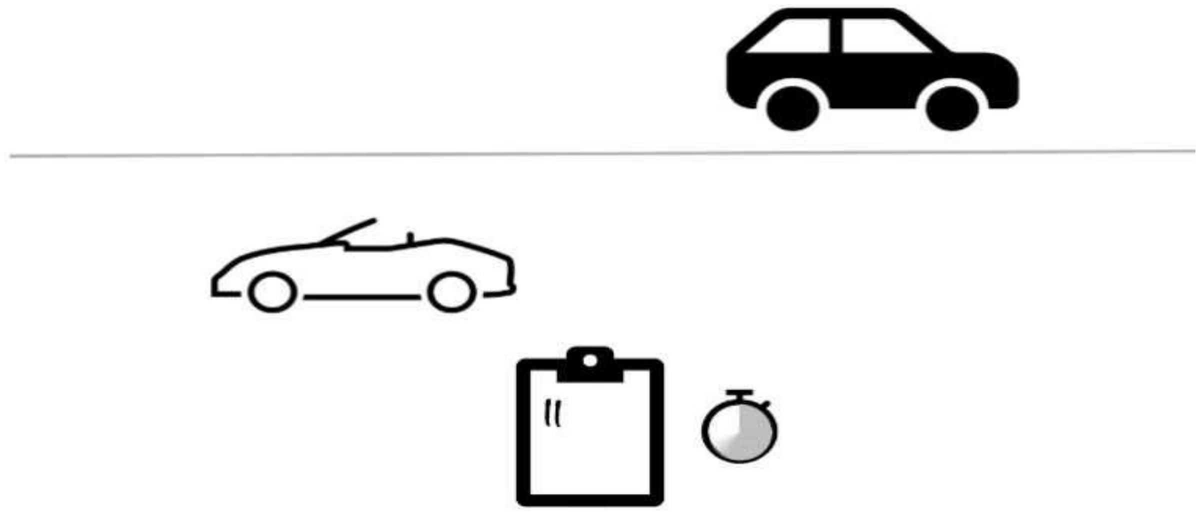
For example, suppose you want to analyse road traffic by counting the number of cars on a stretch of road. A batch processing approach to this would require that you collect the cars in a parking lot, and then count them in a single operation while they're at rest.



If the road is busy, with a large number of cars driving along at frequent intervals, this approach may be impractical; and note that you don't get any results until you have parked a batch of cars and counted them.

- **Short explanation about how it can leverage streaming analytics solutions / platforms**

For example, a better approach to our hypothetical car counting problem might be to apply a *streaming* approach, by counting the cars in real-time as they pass:



- **justification about how it falls under the particular category**

In this approach, you don't need to wait until all of the cars have parked to start processing them, and you can aggregate the data over time intervals; for example, by counting the number of cars that pass each minute.

Q2. You are in a meeting with the firm’s management who are little bit concerned about the challenges associated with streaming analytics. The white paper describes few challenges faced while adapting the streaming analytics. In order to assist the client

- Briefly narrate the four critical challenges in your own words
- Identify the different tools that can be used to resolve / mitigate those challenges
- Address how each of the challenge is resolved with the tools / platforms identified

Answer for Q2:

Briefly narrate the four critical challenges in your own words

Solution: Critical challenges in adapting Streaming Analytics.

It is very complicated and resource-demanding to build and operate custom streaming data pipelines. Few of the challenges are listed below:

a. **Resource Intensive & High cost**-A system must be built to collect, prepare / transform, transmit data that is collected concurrently from large number of data sources.

Complicated, resource intensive, time consuming and very expensive to build and operate our own infrastructure for streaming data pipelines.

b. **Complex Processing**-Frequent fine-tuning and updating the resources for data storage and computation. Low throughput and high latency are big challenges in Streaming applications due to inefficiently tuned storage and compute resources.

c. **Recovery**-Recovering the system from server or network failures and catching up on data processing from appropriate points in the stream becomes very difficult. To avoid duplication of data and effectively recover from server & network failure, we need to have good recovery and monitoring strategies implemented.

d. **Fault tolerance system** – In order to enable the cluster to continue the data processing without interruption, we have to store duplicate copies of the data in the cluster which makes the whole system more complex. To handle the varying speeds of data to the servers, we need to have deployment and management strategies for a range of servers that are involved in stream processing.

Identify the different tools that can be used to resolve / mitigate those challenges

Solution:

- *Amazon Kinesis Firehose*
- *AWS Lambda*
- *Amazon Kinesis Agent*
- *Amazon CloudWatch*
- *Amazon S3 Storage*
- *Amazon Kinesis Streams*
- *Redshift DB*

Address how each of the challenge is resolved with the tools / platforms identified

Solution:

- ***Amazon S3 Storage***

Amazon S3

AWS DMS provides AWS S3 (Simple Storage System) to store and retrieve any amount of data at any time through a web interface. All event objects from data collections are stored in an object storage that offers scalable, reliable, available, secured and high-performance access. The cloud web & native applications store any amount serving millions of customer data to build platforms such as data lakes, applications, backups and archive and can use for any real-time websites & analytics, machine learning use-cases etc., S3 Storage can achieve high storage throughput and low latency when fine-tuned properly.

- ***Amazon Kinesis Firehose***

Amazon Kinesis Firehose

Amazon Kinesis Firehose helps in streaming by efficiently capturing, transforming the data and load streaming data from large number of data sources. It also provisions & scale compute, memory, and network resources without additional administration. Data transformation into different formats like Parquet helps in building our own processing pipelines.

- ***AWS Lambda***

AWS Lambda is a server less, event-driven computing service that lets us run code for virtually any type of application or backend service without provisioning or managing servers. It can execute code at the capacity we need and scale our data volume automatically and enable custom event triggers, powerful ML processing etc. AWS Lambda can achieve high compute throughput and low latency when fine-tuned properly.

- ***Amazon Kinesis Agent***

Kinesis Agent

Amazon Kinesis Agent helps in collecting logs, events and metrics at large volumes from different sources and adding necessary context to the collected data for organizing and storing in AWS infrastructures near real-time. The accuracy and completeness of the data being collected and streamed is confirmed by agent itself. The agent handles file rotation, check pointing, and retry upon failures. To avoid duplication of data and effectively recover from server & network failure.

- ***Amazon CloudWatch***

Amazon CloudWatch

Amazon CloudWatch helps in monitoring and maintaining actionable insights of our applications, system-wide performance and identify ways to optimize resource utilization. It performs Operational Excellence gaining complete transparency on AWS resources, and service running on AWS and on-premises. It can detect anomalous behaviour in application environments, set alarms, visualize logs and metrics to take automated actions and helps ensure the applications are running smoothly. CloudWatch helps us in effectively detecting server and network failure.

- ***Amazon Kinesis Data Streams***

Amazon Kinesis Data Streams is a fully managed, server less data streaming service that stores and ingests various streaming data in real time at any scale. You can use Amazon Kinesis Data Streams to collect and process large streams of data records in real time. You can create data-processing applications, known as Kinesis Data Streams applications. A typical Kinesis Data Streams application reads data from a data stream as data records. These applications can use the Kinesis Client Library, and they can run on Amazon EC2 instances. You can send the processed records to dashboards, use them to generate alerts, dynamically change pricing and advertising strategies, or send data to a variety of other AWS services.

- ***RedShift DB***

Amazon Redshift uses SQL to analyse structured and semi-structured data across data warehouses, operational databases and data lakes using AWS-designed hardware and machine learning to deliver the best price performance at any scale

---

Q3. The white paper discusses three different use cases which the toll station company has addressed using streaming data. But the solution is described in terms of various cloud services offered by AWS. The client does not have the knowledge about the cloud computing and AWS. In fact all the three use cases can be very well addressed with a general architecture used in the big data analytics and streaming analytics. You need to work upon helping client to understand those common architectures.

- Identify the architecture that can be fitted well for capturing all three use cases

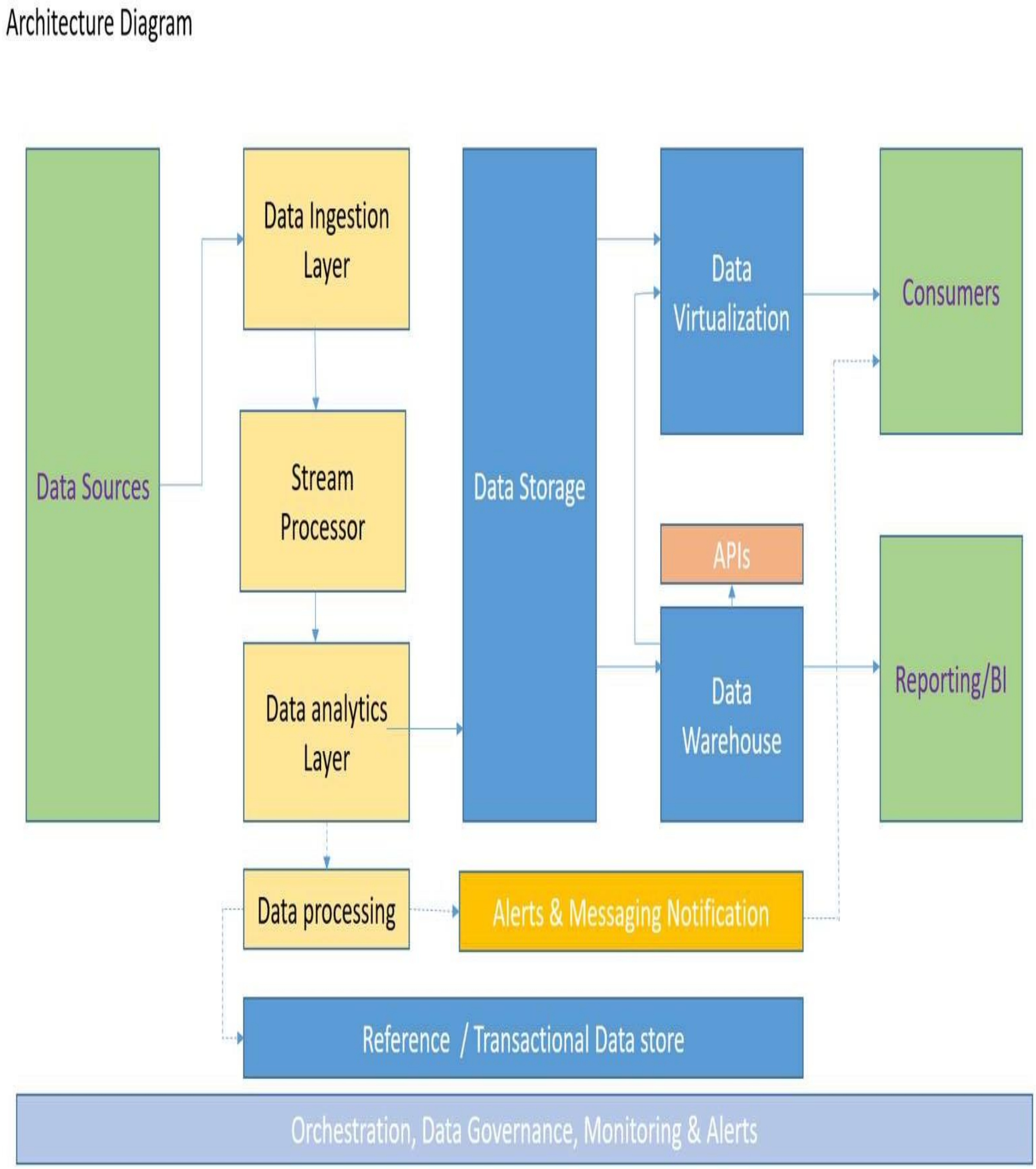
- Convert the final architecture diagram provided by AWS team into an architecture diagram based upon your answer to earlier question

- Take care that all three cases should be vividly coming out of the architecture diagram, if required add brief description about each flow

Answer Q3

- Identify the architecture that can be fitted well for capturing all three use cases

Overall solution architecture in general terms would be like the below .



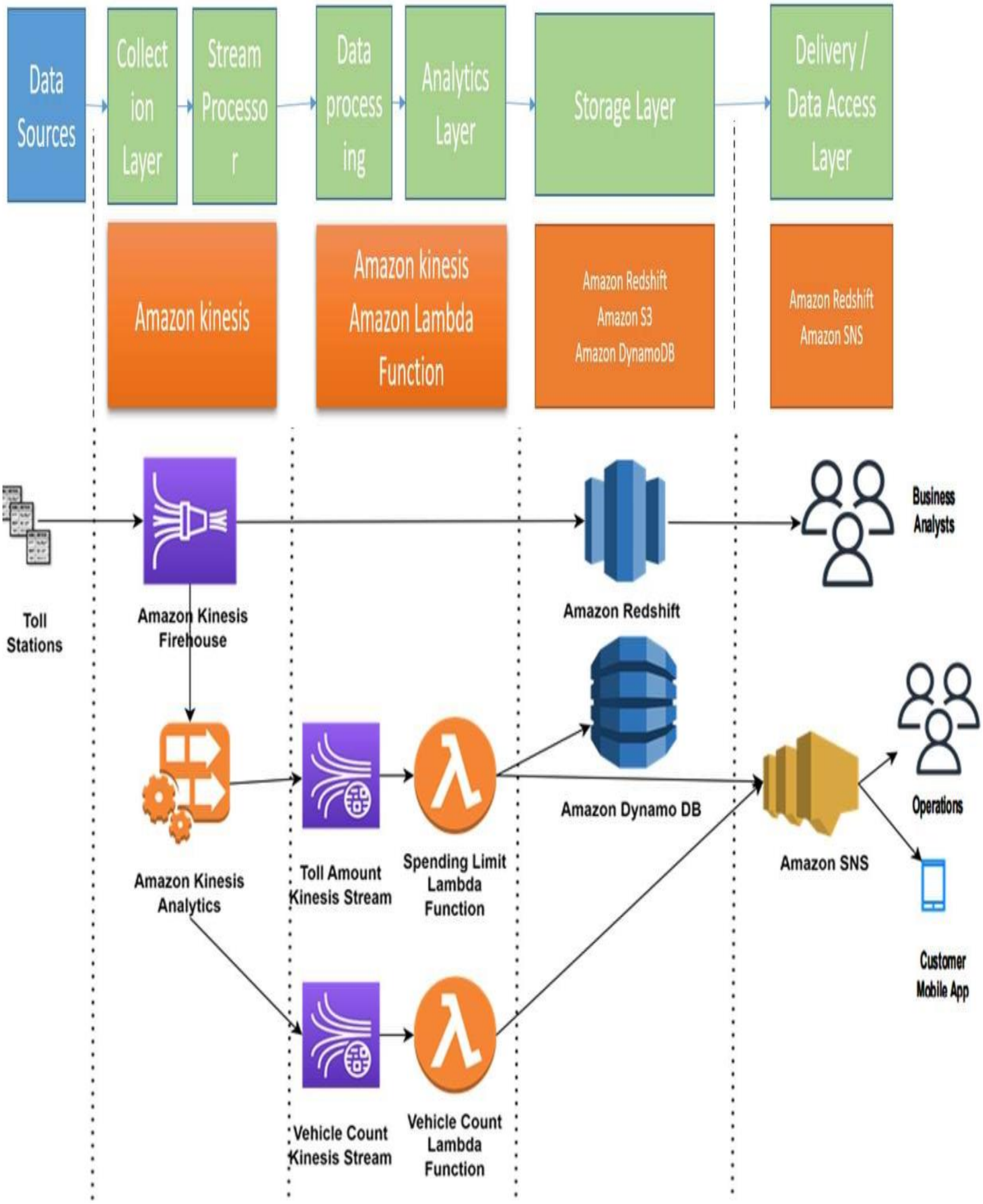
Lambda architecture and Kappa architecture are the two most widely used architecture designs for streaming. Kappa architecture has been taken into account for this solution given the relative simplicity of Kappa (just requiring a streaming layer and no separate batch layer).A typical streaming architecture based on Kappa is shown in the diagram above.

key components of this architecture as follows

Data Source : These are the sources of data that will be ingested and processed by the system. These can include both structured and unstructured data, and may include sources such as databases, log files, social media feeds, IoT sensors, and more.

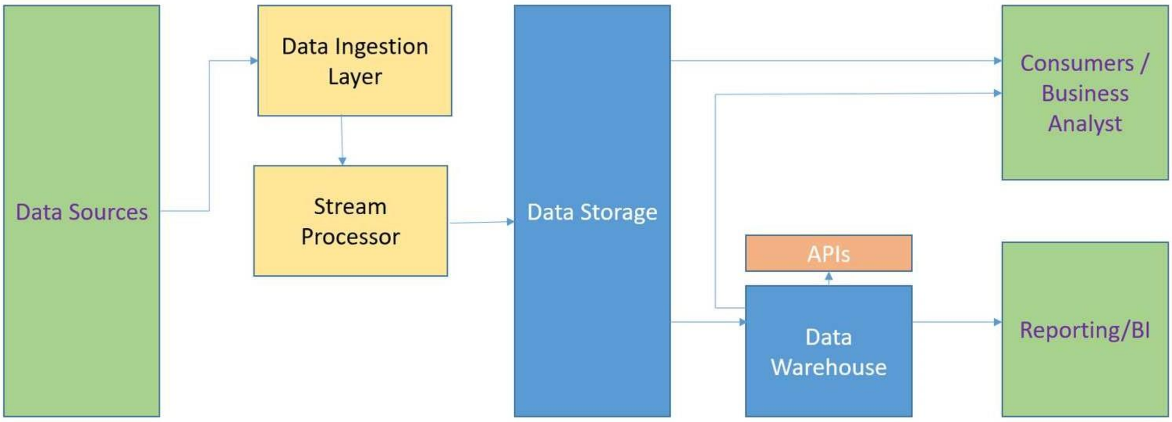
- **Data Ingestion Layer**: The data ingestion layer is responsible for collecting and transferring data from the various data sources into the analytics system. This may involve using tools such as Apache Kafka or Apache Flume to collect and stream the data in real-time.
- **Stream Processor**: An overall coordination system which is responsible to ensure data is catered from sources and made to sinks (destination). Also helps to provide temporary buffer due to high speed of data generation at sources but limited speed of data processing
- **Data analytics Layer**: Responsible for generating real time analytics on streamed data
- **Data processing**: Responsible for any data validation, messaging, transformation wrt business domain. May need to interact with other design components in the application architecture to fulfill its job requirements.
- **Data storage**: The data storage layer is responsible for storing the data in a format that can be easily accessed and queried by the analytics system. This may involve using a distributed file system such as HDFS or a distributed database such as Apache Cassandra.
- **Data warehouse**: Warehouse solution to create time series analysis of the data, trends for reporting and analytics
- **API** - APIs that make data from a data warehouse accessible to other applications
- **Data virtualization**: Application components designed to provide data at low latency speed. Typically in memory data stores for faster response times
- **Reporting / BI** : The data visualization and reporting layer is responsible for providing users with insights and information derived from the processed data. This may involve using tools such as Tableau or Power BI to create dashboards and reports.
- **Consumers**: Consumer can be Mobile App, Business Analyst or Internal Staff(Operators) who is interested in consuming real time data and analytics.
- **Orchestration** : Orchestration refers to the process of coordinating and managing the various components of a stream processing system. This can include tasks such as setting up and configuring the system, starting and stopping streams and jobs, monitoring the status of the system and its components, and managing the flow of data through the system. Orchestration is a critical aspect of stream processing and analytics, as it helps to ensure that the system is running smoothly and efficiently, and that data is being processed and analyzed in a timely and accurate manner. There are a number of tools and technologies available for orchestrating stream processing systems, including open-source solutions like Apache Kafka and Apache Flink, as well as proprietary platforms offered by vendors such as Google Cloud, Amazon Web Services, and Microsoft Azure.

Architecture proposed in the Whitepaper:



Requirement #1: Availability of More recent Data in the Datawarehouse.

Streaming Architecture – Use case #1 More Recent Data in the DataWarehouse

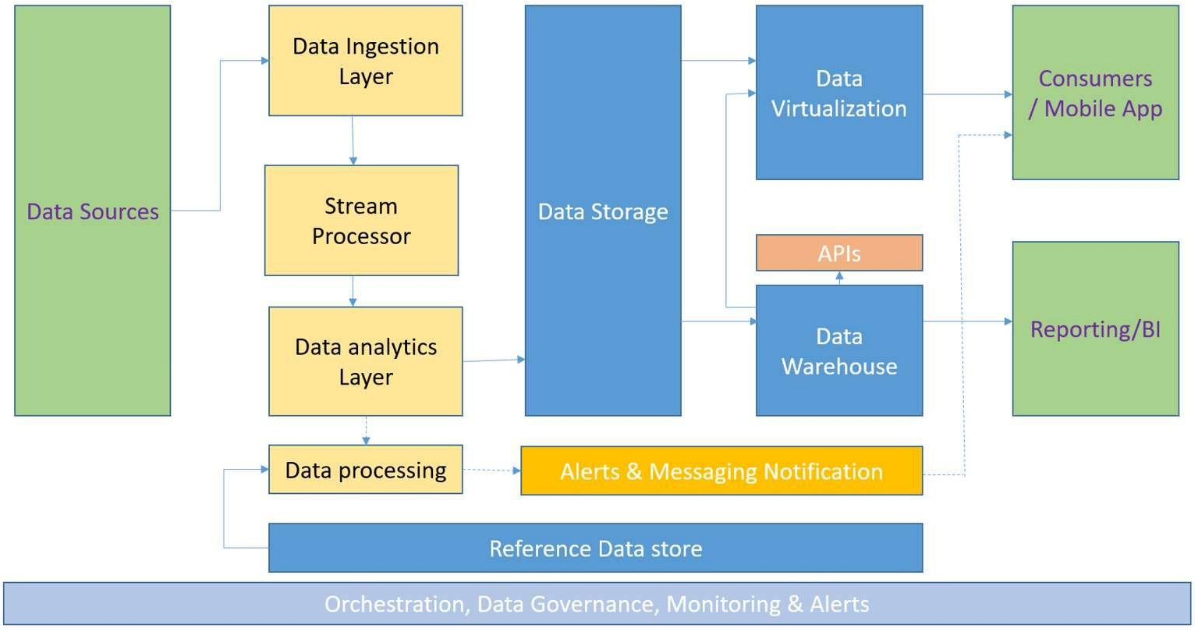


The information flow shown in the above diagram corresponds to the use case# 1.

Real-time data is consumed into the streaming architecture as it is generated by data sources. To offer high availability and fault tolerance, the same can be processed or made available to data storage using a data flow layer.

Requirement #2: Billing Threshold check.

Streaming Architecture – Use case #2 Billing Threshold Alerts

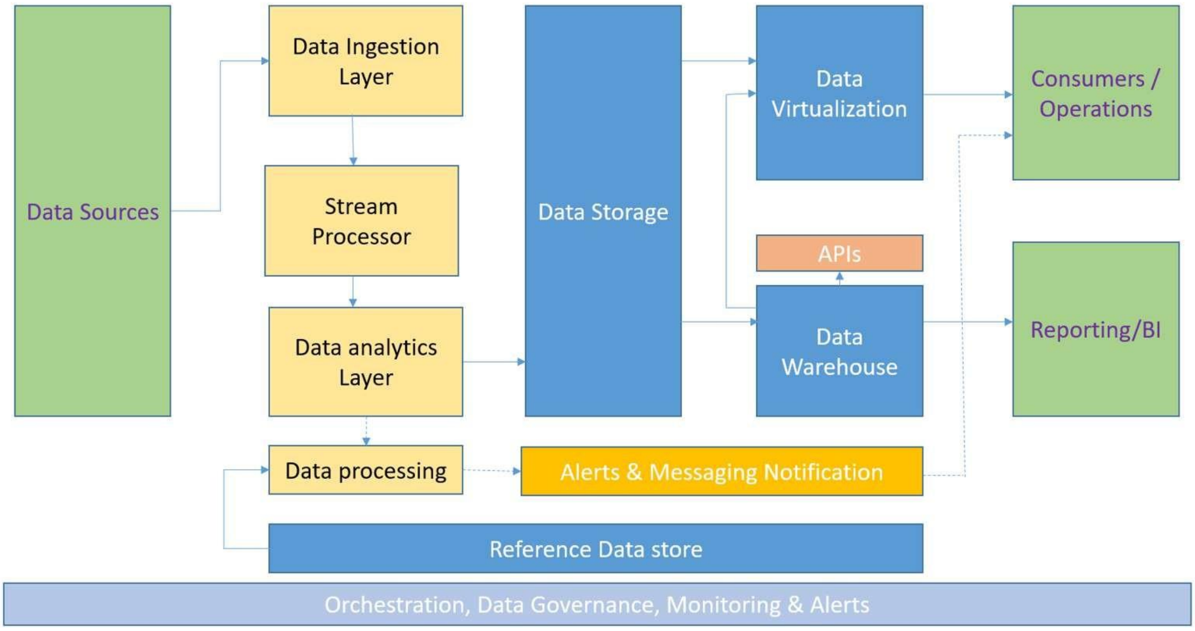


The general architecture for the use case # 2 is as shown above.

This is a requirement where the company wants to keep a track of balance on each customer and then send an alert to customer whenever there is breach of this threshold value so that the customer can recharge/refill his account.

Requirement #3: Other Threshold Alerts.

Streaming Architecture - Use case#3 Other Threshold Alerts



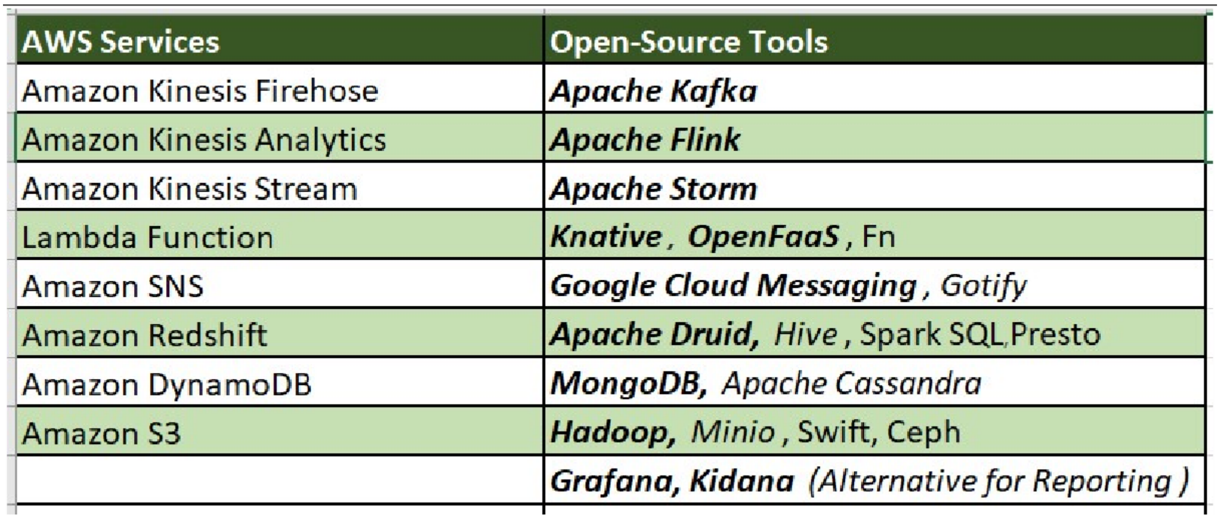
The general architecture for the use case # 3 is as shown above.

The company also wants to keep a track of the traffic at a particular toll station for each time segment. They have reference data with which the streaming data needs to be compared and if the set conditions are not met, an alert needs to be sent to the toll station for verification of the cause.

Q4. The client is now impressed with the capabilities of the AWS and how it's streamlining the application development and deployment. But they also want to discover more on the open-source tools / platforms that can be leveraged. As a result, you need to work upon identifying the open-source tools for each of the use case.

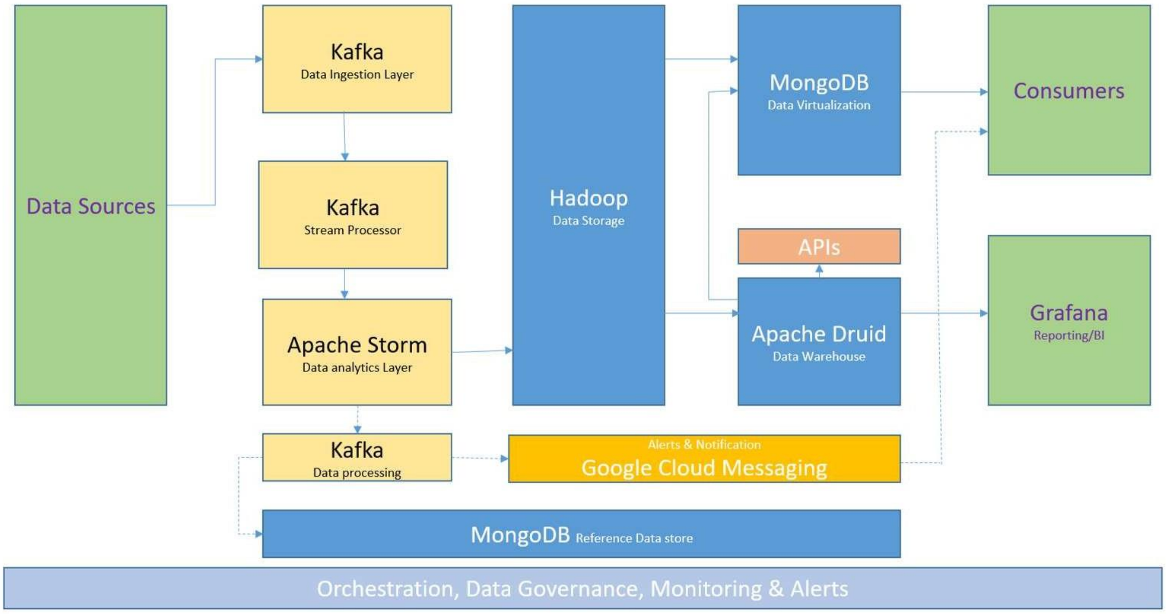
- For each of the use case
  - Identify the tools / platforms that can be used to solve it
  - Draw a solution diagram using the tools identified in earlier question the flow should come out clearly from the solution diagram

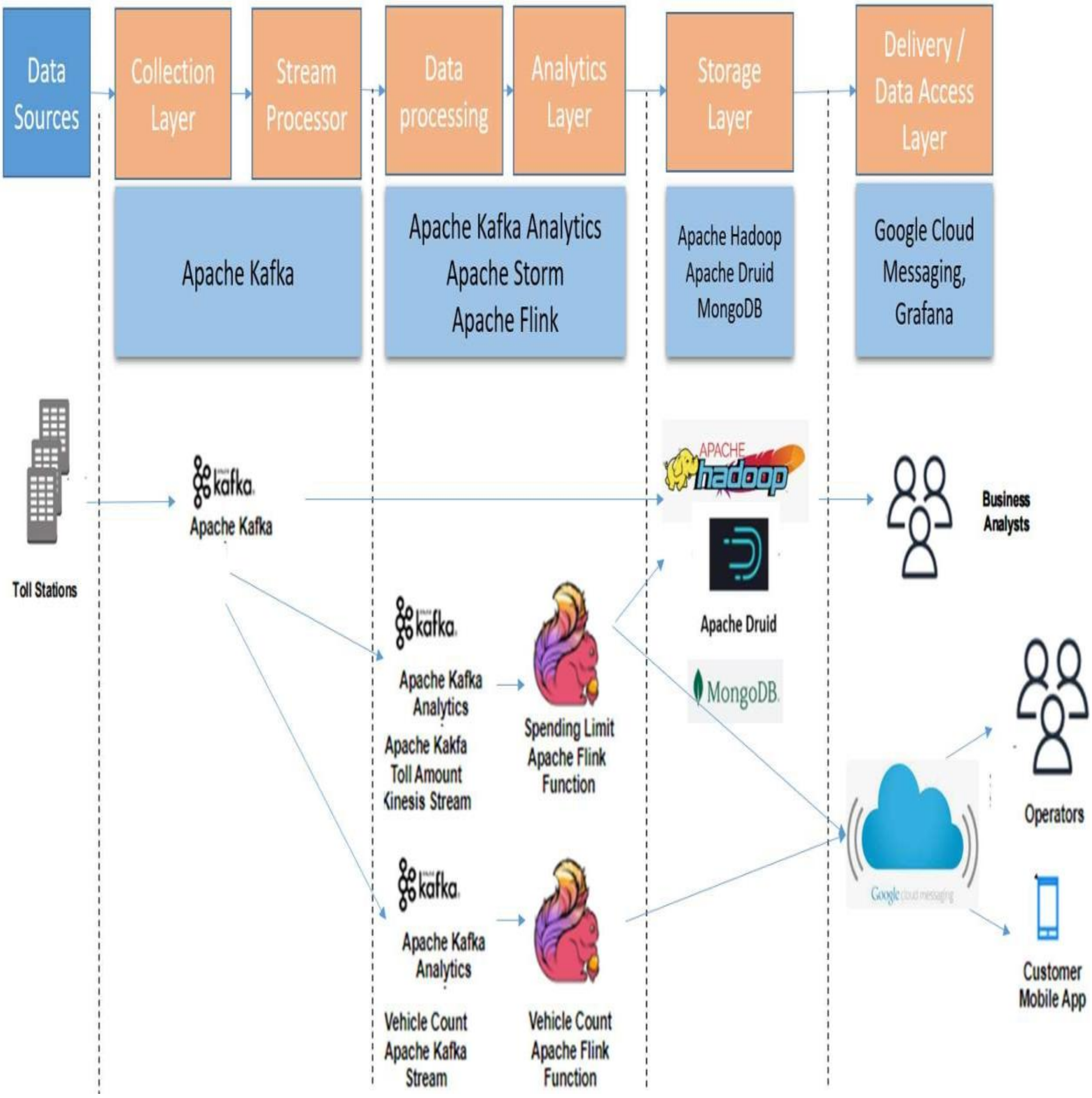
Answer: For the above use cases the alternative open-source tools to the AWS tools are as below and the architecture diagrams are recreated using the Open Source Tools.



AWS Services	Open-Source Tools
Amazon Kinesis Firehose	***Apache Kafka***
Amazon Kinesis Analytics	***Apache Flink***
Amazon Kinesis Stream	***Apache Storm***
Lambda Function	***Knative*** , ***OpenFaaS*** , Fn
Amazon SNS	***Google Cloud Messaging*** , *Gotify*
Amazon Redshift	***Apache Druid,*** *Hive* , Spark SQL,Presto
Amazon DynamoDB	***MongoDB,*** *Apache Cassandra*
Amazon S3	***Hadoop,*** *Minio* , Swift, Ceph
	Grafana, Kidana *(Alternative for Reporting)*

Solution Diagram with Tools




Data Sources
Collection Layer
Stream Processor
Data processing
Analytics Layer
Storage Layer
Delivery / Data Access Layer
Apache Kafka
Apache Kafka Analytics
Apache Storm
Apache Flink
Apache Hadoop
Apache Druid
MongoDB
Google Cloud Messaging,
Grafana
Toll Stations
Apache Kafka
Apache Kafka Analytics
Apache Kakfa Toll Amount Kinesis Stream
Spending Limit Apache Flink Function
Apache Kafka Analytics
Vehicle Count Apache Kafka Stream
Vehicle Count Apache Flink Function
Apache Druid
MongoDB
Business Analysts
Operators
Customer Mobile App

****