SENTIMENTAL ANALYSIS PROJECT ARCHITECTURE

Data pipeline is a process for moving data between source and target systems, the pipeline architecture is the broader system of pipelines that connect disparate data sources, storage layers, data processing systems, analytics tools, and applications.

Architecture Broadly classified into :

- The logical architecture that outlines the process and transformations a dataset undergoes, from collection to serving (see data architecture components).

- The specific set of tools and frameworks used in a particular scenario, and the role each of these performs.

Why Does Pipeline Architecture Matter?

Business appetite for data and analytics is ever-increasing. The need to support a broad range of exploratory and operational data analyses requires a robust infrastructure to provide the right data to the right stakeholder or system, in the right format.

Even a small company might develop a complex set of analytics requirements. Let’s take the example of a company that develops a handful of mobile applications, and collects in-app event data in the process. Multiple people in the organization will want to work with that data in different ways:

- Data scientists want to build models that predict user behavior and to test their hypotheses on various historical states of the data

- Developers want to investigate application logs to identify downtime and improve performance

- Business executives want visibility into revenue-driving metrics such as installs and in-app purchases

Components And Building Blocks

Data infrastructure addresses the full scope of data processing and delivering data from the system that generates it to the user who needs it, while performing transformations and cleansing along the way. This includes:

- **Collection**: Source data is generated from remote devices, applications, or business systems, and made available via API. Apache Kafka and other message bus systems can be used to capture event data and ensure they arrive at their next destination, ideally without dropped or duplicated data.

- **Ingestion**: Collected data is moved to a storage layer where it can be further prepared for analysis. The storage layer might be a relational database like MySQL or unstructured object storage in a cloud data lake such as AWS S3. At this stage, data might also be cataloged and profiled to provide visibility into schema, statistics such as cardinality and missing values, and lineage describing how the data has changed over time.

- **Preparation**: Data is aggregated, cleansed, and manipulated in order to normalize it to company standards and make it available for further analysis. This could also include converting file formats, compressing and partitioning data. This is the point at which data from multiple sources may be blended to provide only the most useful data to data consumers, so that queries return promptly and are inexpensive.

- **Consumption**: Prepared data is moved to production systems – analytics and visualization tools, operational data stores, decision engines, or user-facing applications.

Data pipeline architecture can refer to both a pipeline's conceptual architecture or its platform architecture.

Batch vs. Streaming vs. Combination

1. **Batch Architecture**

A batch pipeline is designed typically for high volume data workloads where data is batched together in a specific time frame. This time frame can be hourly, daily, or monthly depending on the use case. The data formats consumed might consist of files like CSV, JSON, Parquet, Avro and files might be stored in different cloud object stores or data stores of some kind.

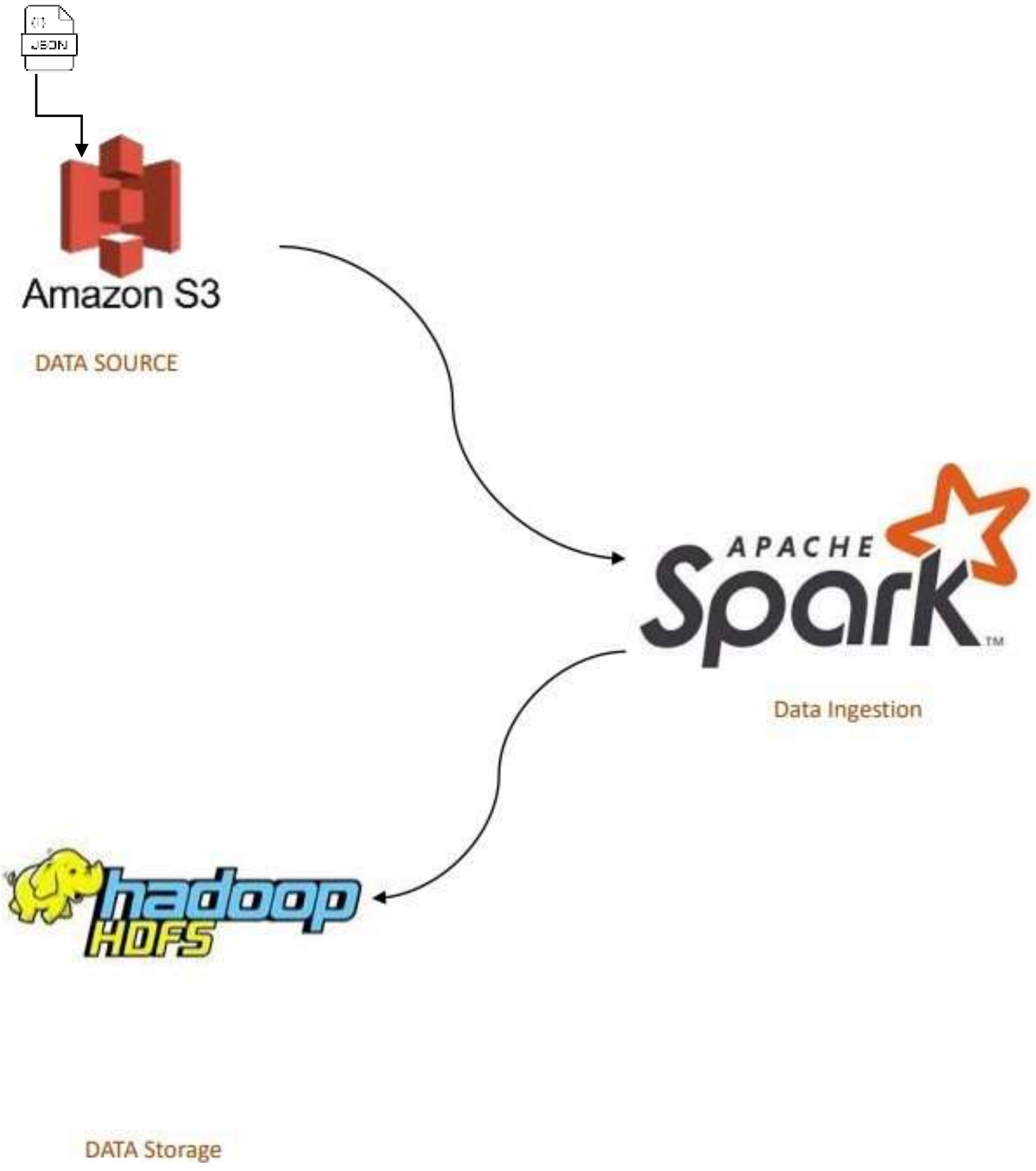
2. **Streaming Architecture**

A streaming pipeline is designed for data that gets generated in real time or near real time. This data is crucial in making instantaneous decisions and can be used for different IoT devices, fraud detection, and log analysis.

3. **Streaming/Batch Architecture**

This architecture is called lambda architecture and is used when there is a need for both batch and streaming data. Although a lot of use cases can be handled by using one or the other, this gives us more flexibility when designing our overall architecture.

PLATFORM ARCHITECTURE


{ }
JSON
Amazon S3
DATA SOURCE
APACHE
Spark
Data Ingestion
hadoop
HDFS
DATA Storage