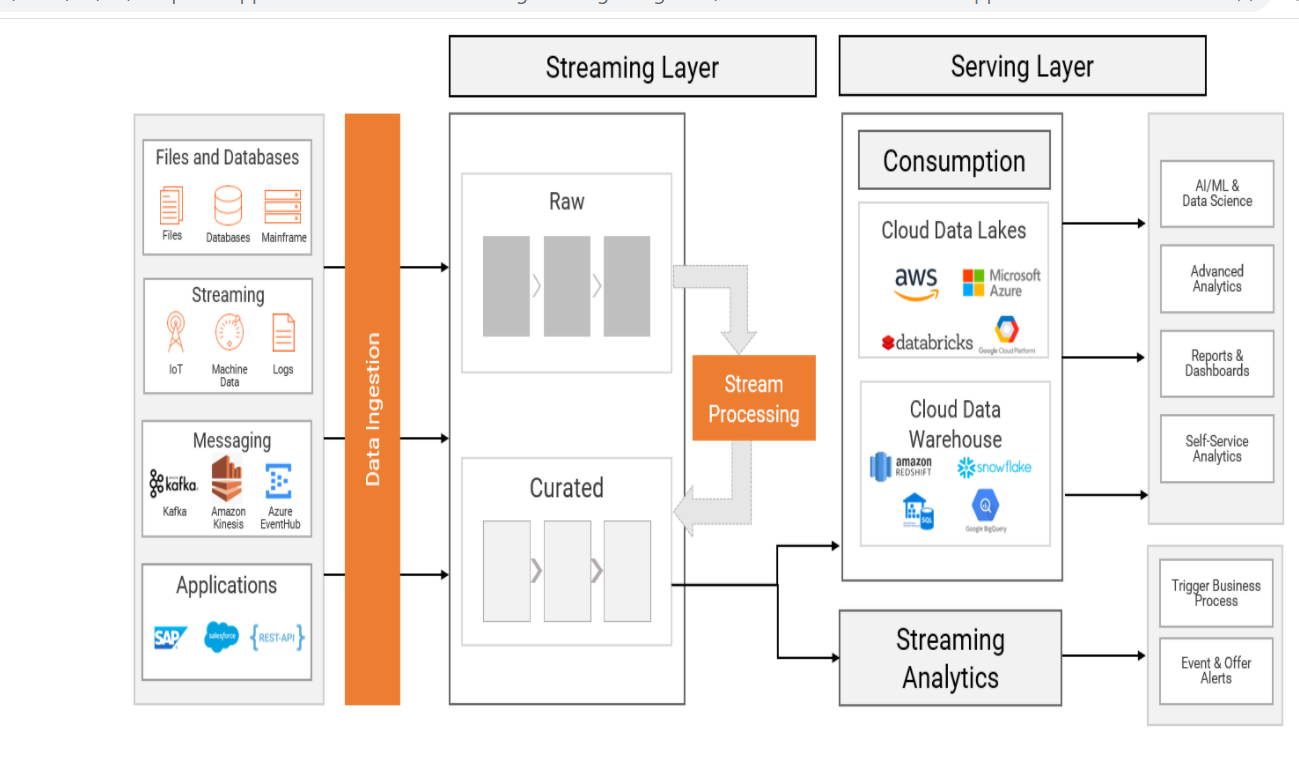
**What is Kappa architecture?**

Kappa architecture is a streaming-first architecture deployment pattern – where data coming from streaming, IoT, batch or near-real time (such as change data capture), is ingested into a messaging system like Apache Kafka. A stream processing engine (like Apache Spark, Apache Flink, etc.) reads data from the messaging system, transforms it, and publishes the enriched data back to the messaging system, making it available for real-time analytics. Additionally, the data is distributed to the serving layer such as a cloud data lake, cloud data warehouse, operational intelligence or alerting systems for self-service analytics and machine learning (ML), reporting, dashboarding, predictive and preventive maintenance as well as alerting use cases.



## How is Kappa different from Lambda architecture?

[Lambda architecture](http://lambda-architecture.net/) is a software architecture deployment pattern where incoming data is fed both to batch and streaming (speed) layers in parallel. The batch layer feeds the data into the data lake and data warehouse, applies the compute logic, and delivers it to the serving layer for consumption. The streaming layer makes use of the previous insights that are derived in the batch layer for processing new incoming data. It is important to note that Lambda architecture requires a separate batch layer along with a streaming layer (or fast layer) before the data is being delivered to the serving layer.

 It is, in fact, an alternative approach for data management within the organization. It can be used in architectures where the batch layer is not needed for meeting the quality of service needs of the organization as well as in the scenarios where complex transformations including data quality techniques can be applied in streaming layer.

## Use cases for adopting Kappa architecture

1. the oil and gas industry where there is a need to update the aggregated status of the machines in the assembly line for real-time reporting and dashboarding.
2. **Real-time rules processing and alerting**: This use case is seen in the retail and telecommunications industry where there is a need to operationalize complex event processing rules on the streaming data from point of sale or e-commerce websites. The result is an alert that is consumed by downstream business applications for triggering real-time campaigns or offer alerts.
3. **Real-time machine learning model operationalization**: This is seen in the financial services industry where pre-created fraud detection ML models are executed on the near real-time data from transactional systems to identify potentially fraudulent transactions and alert the customer so remedial actions can be taken.
4. **Real time operational Intelligence:**This is seen in asset heavy industries where there is a need to perform edge processing on the data before the enriched data is loaded into the cloud data lake or data warehouse. This helps reducing network traffic, operationalize predictive and preventive maintenance techniques as well as enabling IT/OT data integration for holistic analytics.

The **Kappa Architecture** is a software architecture used for processing streaming data. The main premise behind the Kappa Architecture is that you can perform both real-time and batch processing, especially for analytics, with a single technology stack. It is based on a streaming architecture in which an incoming series of data is first stored in a messaging engine like Apache Kafka. From there, a stream processing engine will read the data and transform it into an analyzable format, and then store it into an analytics database for end users to query

The Kappa Architecture supports (near) real-time analytics when the data is read and transformed immediately after it is inserted into the messaging engine. This makes recent data quickly available for end user queries. It also supports historical analytics by reading the stored streaming data from the messaging engine at a later time in a batch manner, to create additional analyzable outputs for more types of analysis.

The Kappa Architecture is considered a simpler alternative to the [Lambda Architecture](https://hazelcast.com/glossary/lambda-architecture/) as it uses the same technology stack to handle both real-time stream processing and historical batch processing. Both architectures entail the storage of historical data to enable large-scale analytics. Both architectures are also useful for addressing “human fault tolerance,” in which problems with the processing code (either bugs or just known limitations) can be overcome by updating the code and running it again on the historical data. The main difference with the Kappa Architecture is that all data is treated as if it were a stream, so the stream processing engine acts as the sole data transformation engine