

# Data Streaming Applications and Their Usage in Healthcare

Mohan Krishna Mangamuri

Health Care Service Corporation, Chicago, United States of America

**Abstract** - We introduce a cutting-edge streaming technology that is scalable, efficient in transmitting real-time data to Data Lake applications. We explain the key components of Confluent Kafka, the brokers, the messages, the partitions, the topics etc. We further explain the best design practices and things to consider while designing Confluent Kafka based streaming applications. We have also added details on the differences between Apache Kafka and Confluent Kafka. We have included code snippet on how to configure the jaas config file with appropriate values. Finally, we conclude how the Confluent Kafka based streaming applications are used in healthcare industry, with real-world use cases.

**Keywords:** Healthcare Information Technology, Data Streaming, Healthcare.

## 1. Introduction

Data streaming applications and their usage in healthcare:

Streaming applications have become increasingly popular in recent years and Confluent Kafka has emerged as a leading platform for building these applications. Real-time streaming applications have significantly impacted the healthcare industry, transforming the way healthcare is delivered, monitored, and managed. In the healthcare industry, there are many use cases where streaming applications can be used to improve patient care and outcomes.

### What is Streaming Data?

A continuous stream of data being produced/generated by a source (called Producer) and in-real time being consumed by a target (called Consumer). For example, social media platforms like twitter, LinkedIn, continuously generate new data in the form of text messages, audio, video, and other formats as well. Data Analytics platforms need to consume this continuous stream of data to provide real-time analytics as opposed to batch processing.

We will look at how Kafka is useful to build streaming applications.

The major components in Kafka are

- Broker
- Topic
- Producer
- Consumer
- Partition
- Offset

A Broker is a server in the cluster where the Kafka software is installed. Each Broker is identified with an ID. Each Broker will have “Topics” configured.

Think of a “Topic” as a table in a Database. Each Record in the topic is referred to as a “Message”.

Each “Topic” is composed of “partitions”. Each partition is a combination of write ahead log. Data is written to the end of the Partition.

Clients/Users writing to Kafka are called Producers. Clients reading the data from the topics are called Consumers.

Each record (message) has an Offset assigned to it which denotes its order in the Partition. Offsets start at 0 and increment by 1 sequentially.

The producer and consumer clients keep running, and if there is no data to consumer, the Consumer would simply wait for new data to be written by producer to the topic.



Each Partition will have Offsets as mentioned above. Records are read sequentially from the initial offset position. Each consumer can be part of consumer group and the permissions can be given at a consumer group level, so any consumer that is part of the group will have access to the topic.



Each consumer group can be identified using the “group\_id” parameter. There can be multiple consumer groups reading from the same topic having their own independent state (offsets). Only one consumer per consumer group can be reading a partition at any point in time.

## 2. Difference between Kafka and Confluent Kafka

Confluent Kafka is built on top of Apache Kafka. One gets to use Kafka without having to setup on their own servers, i.e., Confluent provides Kafka as a managed service. Apache Kafka has a message size limit of 1 MB, while confluent do not impose this restriction. Moreover, Confluent Kafka provides wide range of features such as scalability, fault tolerance, high throughput, and low latency. While some of the Confluent Kafka offerings are free, some are only available through an enterprise license. Examples of the features included in the community license are pre-built connectors, schema registry, REST proxy, ksqldb etc. Confluent version 7.3 is a major release of Confluent platform that provides us with Apache Kafka 3.3 which is the latest stable version of Kafka. Confluent Kafka brokers (servers) can also be configured in Cloud.

Now that we have covered how Kafka works and how it is able to process the streaming data, let us look at the benefits of using Confluent Kafka for building streaming applications in healthcare.

## 3. Design Considerations Using Confluent Kafka

Confluent Kafka is more than a pub-sub (publisher/subscriber) platform, providing fast data processing and data storage capabilities. This is extremely useful for time-critical business applications as access to the data is provided in near real-time. When designing Confluent Kafka based streaming applications, below are the key points to consider:

- 1) Expected size of each message
- 2) Retention period
- 3) Number of partitions
- 4) Access control to the topics

**Message Size:** Apache Kafka have a message limit of 1 mb. While Confluent Kafka is also not built for extremely large

messages, nevertheless it can process files (XML, CSV, JSON, Audio/Video files) of 10 MB or more.

**Retention Period:** In general, it is a standard to use a retention period of 7 days in most cases, but if your use case demands more days/months, the retention period can be configured accordingly. Also, retaining 100s of GBs worth of data which is not usually referred after initial load is also not a good idea.

**Partitions:** Confluent Kafka uses partitions to facilitate parallel consumption of messages in the topics. Partitions are used to scale a topic across multiple servers. One can specify the partitions based on size of cluster, i.e., number of servers configured in the cluster. Too many partitions will also be a bottleneck for replication.

**Access control:** The access to the topic can be controlled, by creating specific role-based access entitlements and only the application IDs can be provided access to those entitlements.

## 4. Usage in Healthcare

### ▪ Patient Monitoring:

Streaming data from medical devices, such as ECG monitors or blood glucose monitors, can be processed and analyzed in real-time to detect any anomalies or changes in a patient’s condition. This can alert healthcare providers to potential issues before they become serious and allow for timely interventions.

### ▪ Prior-Authorizations:

Prior-authorization data from Utilization management vendors can be sent real-time to clinicians, so they can verify the prior-authorization status and adjust the status accordingly. It also enables the health advocates to quickly validate the data coming from the pre-auth vendors, as the streaming data is real-time.

### ▪ Clinical Trials:

Streaming data from clinical trials can be analyzed in real-time to detect any trends or patterns that may be relevant to the study. This can help researchers to make more informed decisions about the study design and treatment protocols, ultimately leading to better outcomes for patients.

Applications built using Confluent Kafka are scalable. The platform can handle large volumes of data in real-time, allowing for processing and analysis of data streams from various sources. This is very crucial in healthcare where there are often many data sources that need to be integrated and analyzed in real-time.

Confluent Kafka is also fault tolerant. It is designed to handle failures and ensure that data is processed and analyzed even in the case of a failure. This is critical in healthcare where data processing failures can have serious consequences for patient care.

## 5. Conclusion

Confluent Kafka has gained wide popularity in recent times because of the advanced features it offers for building streaming applications. In addition to those use cases mentioned above in this article, Streaming applications are also used to implement effective member enrollment, real time streaming of provider data updates. CareFirst is such company that leverages Confluent Kafka for many of the use cases mentioned here. Refer to <https://www.confluent.io/blog/data-streaming-public-sector/>. Similarly, Bankers Health Group leverages Confluent Kafka for several use cases. Please refer to item 4 in the references below. Cerner is a huge name in EMR/EHR world and it uses kafka for various use cases. Refer to item 6 below. This shows that there are growing use cases of streaming applications using Confluent Kafka in Healthcare.

It provides elastic scalability for the workloads instantaneously in Confluent Cloud. As Healthcare is evolving in a faster pace, it is more relevant than ever to use cutting-edge technologies to solve some of the challenges that healthcare faces today. As seen above, Confluent Kafka offers wide range of use cases in healthcare, to build scalable, easily managed, efficient streaming applications.

## REFERENCES

- [1] <https://docs.confluent.io/cloud/current/overview.html>
- [2] <https://kafka.apache.org/documentation/>
- [3] <https://www.confluent.io/blog/data-streaming-public-sector/>
- [4] <https://www.confluent.io/customers/bankers-health-group/>
- [5] <https://www.kai-waehner.de/blog/2022/03/28/apache-kafka-data-streaming-healthcare-industry/#:~:text=Apache%20Kafka%20to%20process%20data,Some%20are%20real%2Dtime.>
- [6] <https://engineering.cerner.com/blog/automated-deployment-with-apache-kafka/>
- [7] <https://drkumo.com/real-time-health-data-streaming-importance/>

### Citation of this Article:

Mohan Krishna Mangamuri, "Data Streaming Applications and Their Usage in Healthcare" Published in *International Research Journal of Innovations in Engineering and Technology - IRJIET*, Volume 7, Issue 7, pp 46-48, July 2023. Article DOI <https://doi.org/10.47001/IRJIET/2023.707006>

\*\*\*\*\*