

REAL-TIME DATA STREAMING ARCHITECTURES WITH KAFKA AND PUB/SUB: BEST PRACTICES AND USE CASES

Jagadeesh Thiruveedula¹ & Vikhyat Gupta²

¹Technological University, Kakinada, Andhra Pradesh 533003, India

²Chandigarh University, Punjab, India

ABSTRACT

Real-time data streaming architecture has become crucial for contemporary applications that demand instantaneous insights and actions, like in financial services, e-commerce, and the Internet of Things (IoT). Among the most well-known systems in this space are Apache Kafka and Pub/Sub messaging systems like Google Cloud Pub/Sub. Though both offer high-level solutions for event-driven architecture, they address different needs in scalability, fault tolerance, and message processing models. Apache Kafka, with its persistent messaging and complex event processing features, is best suited for situations demanding high durability and replaying data. Pub/Sub systems, on the other hand, shine in cloud-native use cases, providing high scalability and flexibility, making them applicable to dynamic applications with distributed aspects. Although they have been widely used, there is a knowledge gap in the comparative performance, integration, and security issues of Kafka and Pub/Sub systems in real-time data streaming. This gap is particularly evident when scaling these systems to process large volumes of data and achieve low latency in global distributed systems. Additionally, although both systems are central to real-time streaming, their integration with new technologies such as edge computing, microservices, and serverless architectures has not been well researched. This review seeks to bridge this gap by consolidating research from 2015 to 2024 about Kafka and Pub/Sub systems. This review concentrates on their performance for different use cases, implementation issues, security practices, and scalable fault-tolerant real-time data streaming architecture best practices. From the findings, it appears that although both systems possess unique strengths, the choice of using Kafka or Pub/Sub highly relies on the application's particular demands, such as scalability, latency, and fault tolerance.

KEYWORDS: *Real-Time Data Streaming, Apache Kafka, Pub/Sub, Event-Driven Architectures, Scalability, Fault Tolerance, Message Processing, Cloud-Native Systems, IoT, Complex Event Processing, Microservices, Serverless Architectures, Security, Data Replay, Real-Time Analytics, Distributed Systems..*

Article History

Received: 17 Jan 2025 | Revised: 24 Jan 2025 | Accepted: 31 Jan 2025
