Apache Kafka:

Type: Distributed event streaming platform capable of handling trillions of events a day.

Infrastructure: Self-managed, though there are managed versions available like Confluent Cloud.

Scalability: Highly scalable through partitions and replication across a Kafka cluster.

Performance: Optimized for high throughput and reliable latency, can handle high volumes of data with low latency.

Use Cases: Best for high-volume event recording, processing logs or metrics, event sourcing, website activity tracking, operational metrics, etc.

Customization and Control: Offers strong control over configurations, which can be crucial for complex applications.

Integration: Broad integration with various logging and monitoring tools, as well as a wide range of consumers due to Kafka's widespread adoption.

Google Cloud Pub/Sub:

Type: Fully managed real-time messaging service that allows messages to be exchanged between applications.

Infrastructure: Fully managed on Google Cloud Platform.

Scalability: Automatically scales to meet the demands of your applications without management overhead.

Performance: Provides global messaging for high-volume applications, ensuring at-least-once message delivery.

Use Cases: Ideal for integrating disparate systems, real-time analytics, distributed event-driven systems, and simplifying messaging infrastructure.

Customization and Control: Less control over the underlying infrastructure since it's a managed service, but easier to use and maintain.

Integration: Natively integrated with other Google Cloud services, making it ideal for projects already tied to Google's ecosystem.

Case Study: Streaming Data Processing for Real-time Analytics

Scenario:

An e-commerce company wants to analyze user interactions on their website in real-time to personalize user experience and improve operational efficiency.

Use Case with Apache Kafka:

Deployment: Set up a Kafka cluster to capture clickstream data, product views, cart updates, and purchases.

Processing: Use Kafka Streams for real-time processing and aggregation of events.

Output: Feed processed data into various systems like a recommendation engine, real-time dashboards, or trigger automated marketing emails based on user behavior.

Advantages:

Kafka's robust handling of high-volume throughput allows processing of millions of messages without noticeable lag, making it ideal for scenarios where performance and customization are critical.

Use Case with Google Cloud Pub/Sub:

Deployment: Implement Pub/Sub to ingest the same user interaction events.

Processing: Utilize Google Cloud Dataflow to process and analyze data streams.

Output: Push insights to BigQuery for real-time analytics and to other Google services for immediate action.

Advantages:

Fully managed services reduce operational overhead and integrate seamlessly, allowing for quick setup and easier scalability without deep expertise in infrastructure management.

Conclusion

Choosing between Apache Kafka and Google Cloud Pub/Sub largely depends on your specific requirements:

Kafka is more suitable for complex, high-throughput systems where fine-grained control over the environment and data pipelines is necessary.

Google Cloud Pub/Sub offers ease of use and integration within the Google Cloud ecosystem, ideal for projects requiring less customization and rapid scaling.

Your tech lead can use this information to weigh which solution fits best with the current and future needs of your infrastructure and business goals

Pub/Sub don’t provide streaming api like Kafka.   
<https://cloud.google.com/java/docs/reference/google-cloud-pubsub/latest/com.google.cloud.pubsub.v1>

We have to go via Dataflow pipeline to work it like streaming

Example Workflow: Streaming Analytics Pipeline

Let's say you want to analyze user activities on your website in real-time to generate instant metrics and insights.

Data Collection: User activities (like page views, clicks, and interactions) are sent to Google Cloud Pub/Sub from various sources, perhaps from a user's browser or server logs.

Stream Processing:

The messages (events) are pulled from Pub/Sub by a Dataflow pipeline.

Dataflow processes these events in real time—filtering, aggregating, and otherwise transforming the data based on the requirements.

For instance, you might count the number of clicks per article on your site in a given time frame using windowing functions in Dataflow.

Output: The processed data can be pushed to various destinations:

Real-time dashboards (using Google Data Studio or a similar tool).

Databases or Big Data warehouses like Google BigQuery for further analysis or permanent storage.

Trigger other downstream actions, such as sending alerts or automated emails.

<https://biztechmagazine.com/article/2022/03/how-financial-institutions-can-build-better-customer-profiles-cloud>

<https://cloud.google.com/dataflow/docs/concepts/streaming-with-cloud-pubsub>

Using Google Cloud Pub/Sub, like any cloud service, comes with specific technical risks that need to be managed to ensure the reliability and security of your systems. Here are some potential technical risks associated with using Cloud Pub/Sub:

Data Loss and Inconsistency: There is always a risk of data loss in any distributed system, including cloud-based pub/sub services. In scenarios where message delivery is critical, the loss of messages due to network failures or software bugs can lead to data inconsistency.

Security Vulnerabilities: As with any cloud service, there is a risk of unauthorized access to data. If not properly secured, sensitive data might be exposed to unauthorized users. This includes risks from inadequate access controls or misconfigured security settings, such as open permissions on Pub/Sub topics and subscriptions.

Vendor Lock-in: Using a proprietary service like Google Cloud Pub/Sub can lead to vendor lock-in, making it challenging and potentially costly to switch to another service in the future. This might limit your ability to adapt to new requirements or to take advantage of better offerings from competitors.

Scalability Limits: While Google Cloud Pub/Sub is designed to scale automatically, there can be practical limits based on resource quotas, the geographic distribution of your data, and the architecture of your solution. These limits might affect the maximum throughput or the latency of message delivery under high load conditions.

Compliance and Regulatory Risks: Ensuring compliance with data governance and privacy regulations can be more complex when using a cloud service. Data residency and compliance with standards such as GDPR, HIPAA, or others might require specific configurations and controls in your Pub/Sub implementation.

Dependency on External Network Infrastructure: Cloud services rely on the internet, which can introduce latency or reliability issues that are out of your control. Network outages or poor connectivity can delay message delivery or disrupt your service.

Monitoring and Management Complexity: Effectively monitoring a large-scale implementation of Cloud Pub/Sub can be challenging. The complexity increases as the system scales and integrates with other services. Failure to adequately monitor the system can lead to undetected issues, affecting system performance and reliability.

<https://statusgator.com/services/google-cloud/google-cloud-pubsub>

<https://github.com/googleapis/google-cloud-python/issues/4600>

<https://www.trustradius.com/compare-products/apache-kafka-vs-google-cloud-pub-sub>