

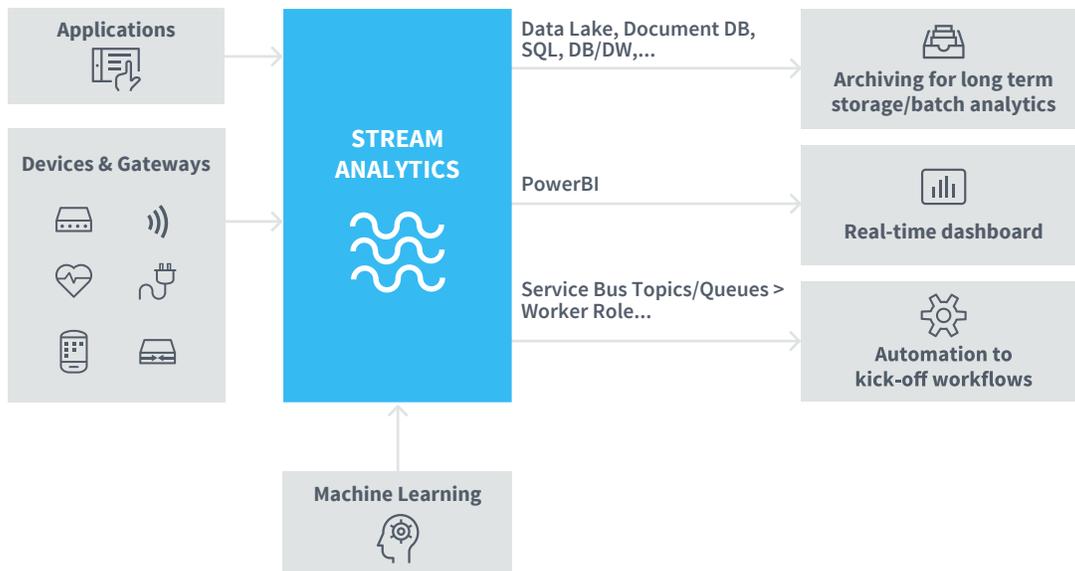
# Unravel<sup>®</sup> for Kafka

## Streaming Data

We live in a real-time world where several processes and events (e.g. IoT devices, financial transactions, social media, medical monitoring etc.) produce a continuum of meaningful data. Time-value of data is critical. It pays to capture and analyze real-time events in the form of streaming data to derive actionable insights and make timely data-driven business decisions. A new breed of real-time applications that can process this type of streaming data have taken advantage of tremendous advances in open-source big-data technologies, which provide both performance and scalability at extremely high throughput rates.

Distributed systems such as Kafka<sup>®</sup>, Spark<sup>®</sup> Streaming, Flink<sup>®</sup>, and HBase<sup>®</sup> have emerged as key components of the Big Data stack to support streaming data analytics. These systems provide a unified and high-performance architecture for processing real-time data streams. These individually distributed systems are connected to each other by a streaming data application. Enterprises now run a wide swath of streaming Big Data applications in production to provide their customers great value. Stream processing powers some of the most critical undertakings today such as algorithmic trading, autonomous and connected cars, transportation, manufacturing, and health monitoring, to name a few.

*A real-time streaming architecture*



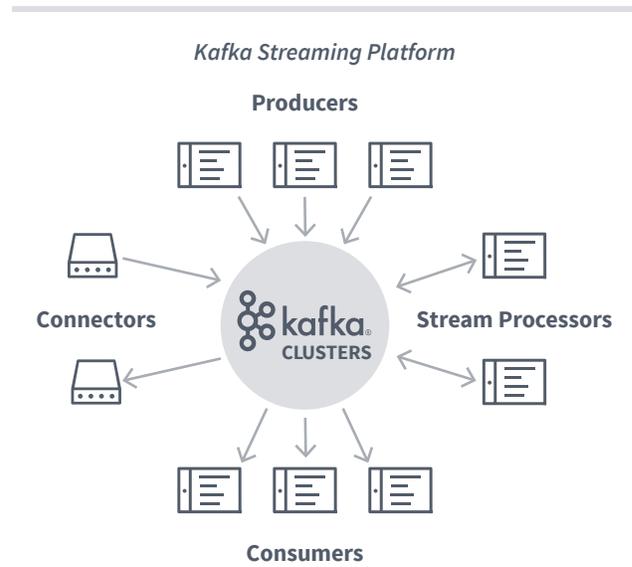
# Apache Kafka®

Apache Kafka is an open-source distributed stream processing platform to publish and subscribe, store, and process streams of real-time data feeds. It is used to build streaming data pipelines and applications. Kafka has four core APIs, viz., Producer, Consumer, Stream, and Connector. Producers send messages to a Kafka broker and are read by Consumers. Messages are associated by the producer with a topic (grouping), providing consumers higher abstraction of details. Kafka is run as a cluster on one or more servers and can be used as a messaging, storage, and stream processing system, all of which are integral to Kafka's key role as a streaming platform.

Kafka has seen a massive adoption by enterprises across diverse verticals, applications, and use-cases. Kafka has become the enterprise backbone for connecting various big-data stack components. Kafka is also used in conjunction with systems such as Hadoop, Spark, Cassandra, etc.

## Monitoring Challenges

Connecting performance and metrics from an application point of view is critical to maintain the health of these mission critical streaming applications. Tools such as Apache Ambari™, Cloudera Manager™, and Confluent Control Center™, etc. offer a good built-in system or service monitoring capabilities. However, being able to monitor and manage issues related to missed SLAs, performance, failures and resource utilization for big data streaming apps remains a huge gap in the ecosystem. DataOps teams today don't have a single pane of glass to monitor and manage all of the distributed components together (e.g., Kafka, Spark Streaming, etc.), and further rationalizing performance metrics to application usage makes it extremely tedious to troubleshoot problems. For instance, if the results for a streaming application are untimely, is the problem due to lag in the Kafka queues or is it caused by a resource contention in the analytics pipeline or an inefficient configuration, or poor partitioning of the data layout. Monitoring such blind spots become critical especially when scaling production grade multi-tenant applications. A comprehensive and easy to use

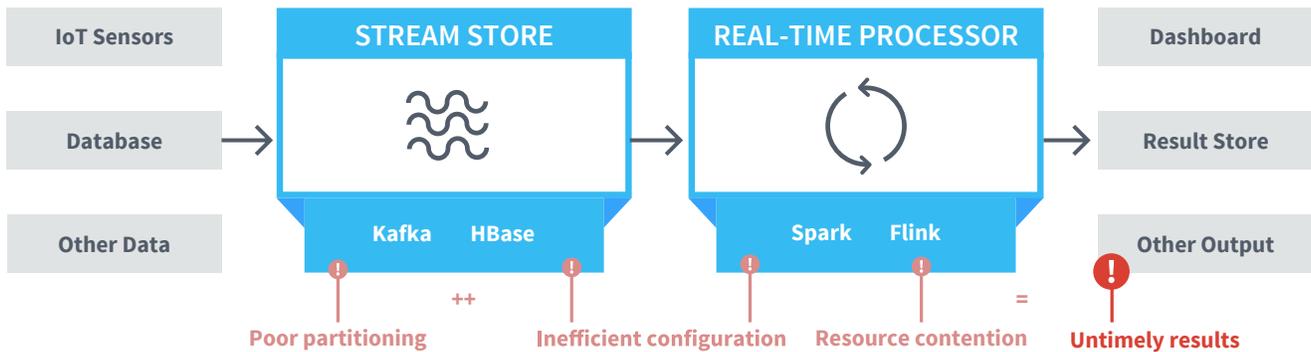


approach is necessary to capture the essence of big data streaming apps, analyze and provide recommendations to fix issues with performance, bugs, and inefficiencies.

Unravel for Kafka enables DataOps teams to comprehensively monitor and manage performance, predictability, and reliability, of Kafka based streaming data applications end-to-end (i.e., across all the components such as Kafka, HBase, Spark, that make up an application). For example, it correlates application KPIs with Kafka cluster and performance metrics, allowing rapid problem diagnosis and fast failure detection for Kafka-based applications.

The Unravel dashboard for Kafka shows key KPIs (e.g total fetch requests, # of under replicated partitions, # of offline partitions etc.) for every Kafka cluster it tracks.

### Potential problems in a streaming big data stack



Within a cluster, the dashboard tracks all the relevant “Topics” and the “Brokers” and “Consumer Groups (CG)” associated with them. The Unravel Intelligence engine provides insights into a cluster’s activities through the status of the Consumer Group’s (CG). A CG’s state is determined on a partition by partition basis, specifically by its commit offset relative to the producers log commit offset for a particular partition. The intelligence provided by Unravel can help easily identify bottleneck in the Kafka cluster that affect the overall throughput and efficiency of a streaming application. The KPI’s tracked by Unravel help in planning capacity and growth of the Kafka cluster to meet application demands.

Unravel is designed to automatically detect, diagnose and resolve Apache Kafka problems to make sure the data flows smoothly. Unravel ensures that streaming data applications are reliable in production, preventing data loss and applications from getting stuck. Some of the key benefits include:

#### Keep the data flowing smoothly

- Monitor all important Kafka metrics and automatically detect and alert on anomalous Kafka behavior at the cluster, topic and consumer level
- Remediate load imbalance in Kafka caused by factors such as poor data partitioning
- Accurately plan and allocate Kafka capacity to meet application SLAs

#### Guarantee reliability and performance of streaming and IoT applications

- Correlate application KPIs with Kafka performance metrics for rapid problem diagnosis
- Monitor Kafka-based applications end-to-end and quickly detect failures and slowdowns
- Troubleshoot and tune applications that are unable to keep up with input data rates in Kafka

#### The Unravel intelligence engine

- Provides insights into a Kafka cluster’s activities through the auto-detection of the status of relevant Consumer Groups (OK/Lagging/Stalled) and KPIs, such as number of topics and partitions
- Drill down easily from the cluster to determine where Topic/Consumer Group is lagging or stalled
- View configured Kafka clusters, cluster metrics, broker/consumer group KPIs, partition details/KPIs/status

## Business benefits of using Unravel

Unravel makes sure businesses can achieve their goals with streaming applications across their big-data stack. Some of the tangible benefits include:

- Improved reliability to meet SLAs, ensuring no disruption in revenues
- Optimized resource utilization, resulting in lower infrastructure and project costs
- Improved productivity for all stakeholders