Cloud-Native Apache Kafka: Event-Driven Applications with Red Hat AMQ Streams

RTInsights.com

Red Hat

Trends in Event Processing, Integration, and Agile Development Practices

Successful enterprises have learned that business and technology initiatives must be closely aligned in order to deliver the greatest benefits of improvements in operational efficiency and customer experience. Events—or changes in a business situation that may yield useful information to business leaders—are the current focus of many enterprise software initiatives that seek to garner insight from information such as customer contacts, transactions, operations, and more. Traditional technologies, such as batch-processing ETL (extract, transform, load) for recording, collecting, and processing these events have historically placed restrictions on how quickly businesses could obtain the insights held within events.

Rapid advances in technology—such as real-time data collection, better connectivity and bandwidth, more scalable data stores, and improvements in computing performance—are transforming not only how businesses operate, but also how business leaders think. Businesses can now "see" themselves and their ecosystems accurately, and adjust operations rapidly and efficiently when needed. As businesses react faster and faster, IT staff must also react faster and faster, iterating rapidly through shorter software development cycles, so that applications remain aligned with business.

Today's businesses increasingly rely on microservices and containers to deliver high-performance and highly scalable event-driven applications. This architecture, combined with Agile practices, has brought sweeping change to the application development lifecycle.

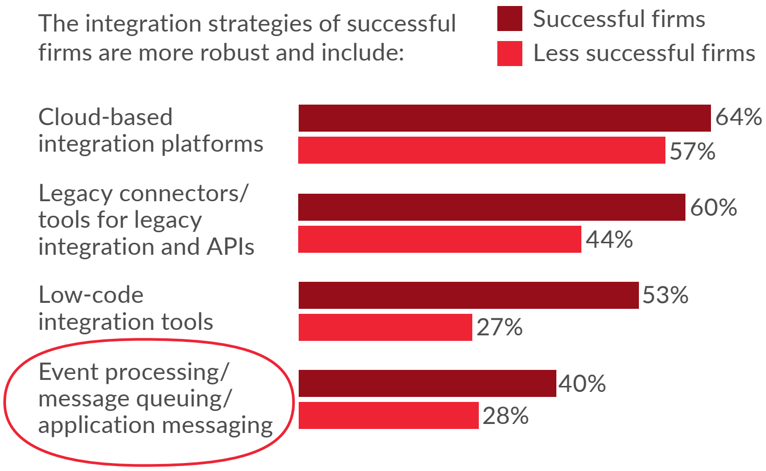
Today's businesses increasingly rely on microservices and containers to deliver high-performance and highly scalable event-driven applications. This architecture, combined with Agile practices, has brought sweeping change to the application development lifecycle. Development teams can now rapidly improve applications and integrations across organizational silos. Nimble technology platforms enable nimble businesses, and therefore drive competitive advantage.

Event-driven architecture (EDA) is coming to the forefront as a method for leveraging microservices for greatest impact. EDA is "a software architecture pattern promoting the production, detection, consumption of, and reaction to events."[1] Event-driven applications work with individual events, not batches, and run in parallel using push-based communications. Removing batch and polling technologies results in faster business processes and analyses. EDA is being used today to integrate applications and data on an enterprisewide basis to enable real-time decision making and situational awareness across the entire organization—and beyond.

[1] K. Mani Chandy Event-Driven Applications: Costs, Benefits and Design Approaches,California Institute of Technology, 2006

Event-driven architectures are driving innovative software of all kinds—from stock ticker feeds for trader dashboards to the way enterprise apps and new web/mobile apps integrate. EDA enables businesses and customers to respond in real time to a rapidly changing business environment.

Firms Need an Agile Integration Approach to Match the Speed of Business Change



3X **Successful firms with an agile integration strategy release/update APIs weekly or daily, nearly three times more frequently than less successful firms.**

Source: *Agile Integration Is Critical to Successful Digital Transformation*, a commissioned study conducted by Forrester Consulting on behalf of Red Hat, September 2018

Building a Robust Platform

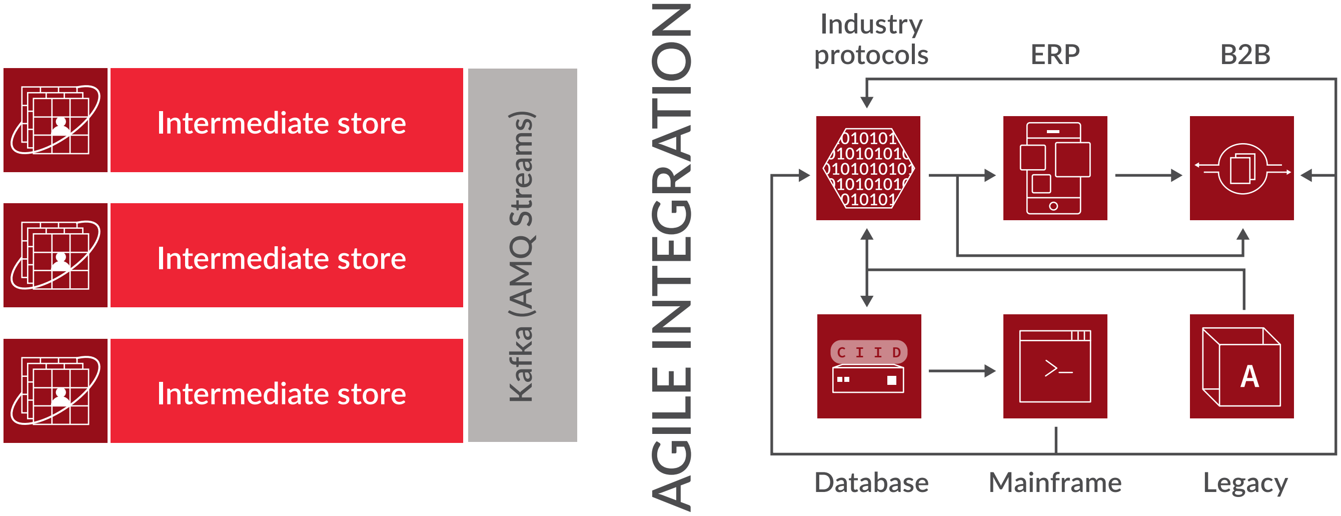
EDA drives a technology preference for Apache Kafka, a publish and subscribe event distribution system that is fault-tolerant, high-throughput, low-latency, and horizontally scalable. Kafka is the basis for event-streaming architectures and stream processing applications. Kafka is frequently combined with Kubernetes. As a result, Kafka and Kubernetes have risen to the top as state-of-the-art, event-streaming technologies.

Kubernetes is a container management system that was open-sourced by Google in 2014. It is used to automate deployment, scaling, and operations of containers across clusters of hosts. Using Kubernetes simplifies deployment and ongoing management of containers, including automatic health-checks for increased app resiliency.

Kafka and Kubernetes are frequently combined to build robust, event-streaming platforms that enable enterprise integrations via a microservices architecture. These platforms have proven to be flexible, resilient, performant, and scalable across many deployments. More than one-third of the Fortune 500 uses Kafka, including the top ten travel companies, eight of the top ten insurance companies, nine of the top ten telecom companies, and seven of the top ten banks.[3] More than a few companies process in excess of one trillion messages per day. The popularity of Kafka and Kubernetes has led to a significant increase in online training and university classes about them.

The combination of Kafka and Kubernetes gives businesses the ability to create event-driven, real-time data pipelines that stream and integrate data between applications. They encourage a model in which software is deconstructed into modules or services/microservices that can be independently developed by different teams, independently updated, and independently scaled.

Agile Integration Bridges the World of Microservices Development with Legacy IT



This eBook will explain the most common ways to build event-driven enterprise applications such as those used for stream processing, data integration, and website activity tracking, and how Red Hat AMQ Streams provides a flexible, enterprise-grade distribution of Apache Kafka (running on OpenShift) that serves as the foundation of a real-time software and business innovation.

[3] https://dzone.com/articles/what-is-kafka

Design Patterns & Usage Scenarios

AMQ Streams is a Red Hat enterprise distribution of Apache Kafka, consisting of the core components of the Apache Kafka project. Much of the extra value that AMQ Streams brings to Kafka is focused on the use of Kafka on Kubernetes, or OpenShift, which is the Red Hat distribution of Kubernetes.

Red Hat AMQ Streams on OpenShift delivers Kafka on Kubernetes to enable enterprise-grade, event-driven architectures that support distributed data streams and stream-processing microservices-based applications. AMQ Streams is particularly well-suited for high-scale, high-throughput scenarios because the inherent partitioning in Kafka helps address scalability requirements.

Following are examples of design patterns (reusable best practices that developers utilize when approaching common problems in application design) that have emerged which demonstrate the power of Red Hat AMQ Streams as a foundation for the event-driven enterprise.

STREAM PROCESSING

- Applications are composed of sets of event-driven services that are connected via AMQ Streams.
- Enables continuous, real-time applications built to react to, process, or transform multiple streams of events at scale.

USE CASE:
IoT-based environmental monitoring in manufacturing plants relies on machine-learning microservices to process events and proactively replace mission-critical machinery when needed.

USERS: Plant Operations Managers, Data Scientists, Business Analysts, Line Managers

BUSINESS BENEFITS: Reduced plant downtime, increased units manufactured, reduced risk of operating machinery under hazardous conditions, improved product quality

Why is this better than alternatives? Reliable, low-latency event-streaming architecture makes real-time processing of machine learning models possible.

USE CASE:
Financial fraud detection based on historical and real-time analysis of customer behavior across multiple systems such as credit, banking, and trading, while demonstrating rigorous compliance with industry regulations.

USERS: CFO, COO, Data Scientists, Business Analysts

BUSINESS BENEFITS: Decreased anomaly detection time from weeks to seconds, reduced costs, reduced fraud

Why is this better than alternatives? Kafka and Kubernetes enable shared, reliable, low-latency event distribution so all business units can now work with the same events to achieve relevant and consistent insights.

DATA INTEGRATION

- Captures streams of events or data changes and feeds these to other data systems.
- Widely used to process streams of events in real time while publishing into another database for historical analysis and retention.
- Change Data Capture (CDC) replaces ETL batches with more timely and granular updates between separate databases.

USE CASE:

Retail dashboard surfaces real-time trends for decision makers simultaneously as transactions and operational data are saved to enterprise data lakes for combined historical analysis and greater insight.

USERS: Business Analysts, Data Scientists, COO, CFO, CEO, Regional Managers, Store Managers, Buyers

BUSINESS BENEFITS: Price and inventory events and changes across all digital and brick-and-mortar entities are reconciled within seconds instead of days, increased customer satisfaction, increased sales, increased repeat sales

Why is this better than alternatives? Process billions of pricing and inventory events per day, create operational tools and alerts to optimize pricing and inventory. Kafka enables event sharing between environments, A/B testing, sophisticated real-time analytics.

USE CASE:

Multinational insurer transforms from mainframe to a microservices-based EDA, and preserves data integrity and consistency between internal and web/mobile customer-facing operations.

USERS: COO, Agents, Business Analysts, Unit Managers

BUSINESS BENEFITS: Connect core legacy data systems with new business applications and channels, roll out new enterprise-scale EDA in less than one year, event replication and streaming times reduces from days to seconds

Why is this better than alternatives? Event streaming and replication across systems empowers IT teams to work faster and with greater efficiency, rolling out modern applications that provide real-time customer insights.

ADDITIONAL DESIGN PATTERNS

Website Activity Tracker

- Rebuild the user activity tracking pipeline as a set of real-time publish-subscribe feeds to enable greater scalability, expose events as streams to applications for complex processing and analysis.
- Website navigation is comprised of events such as page views and object views. Activity is published to central topics with one topic per activity type.
- In practice: Website uses real-time customer clickstream analysis to suggest add-ons and additional products on the fly, while events are sent to other apps to forecast product demand.

Event-Driven Enterprise

- Replacement of traditional message broker and messaging architectures.
- High scale, high throughput, built-in partitioning, replication, and fault tolerance.
- In practice: Bank incrementally adds event-driven microservices to a monolithic mainframe application to update transaction processing and replace batch jobs with real-time streaming applications for fraud detection and reporting.

Centralized Business Metrics

- Aggregation of statistics from distributed applications to produce centralized feeds of operational data.
- Operational data can be analyzed and statistics provided to distributed applications as and AMQ Streams topic. This provides an effective way of centralizing that view and processing that data in different ways.
- In practice: Previously isolated silos of supply chain data are combined, used to more accurately predict the timing of dependencies, and used to automatically optimize production schedules across multiple locations.

Log Aggregation

- Publish log messages as events that flow through AMQ Streams topics.
- Treat details previously hidden in text files as events and process them in streams using distributed microservices-based applications.
- Offers good performance, stronger durability guarantees due to replication.
- Builds a simplified environment where multiple services can process log data in real time or asynchronously.
- In practice: Enables real-time detection of potentially compromised corporate IP assets across multiple geographies, clouds, and ISPs.

How AMQ Streams Delivers Apache Kafka and Kubernetes to Event-Driven Enterprise Applications

Red Hat AMQ is a suite of event-streaming technologies that provides the flexibility to select the enterprise-grade components that meet your needs. AMQ Streams runs on top of OpenShift, Red Hat's Kubernetes, an enterprise-grade container orchestration platform. This combination enables modular, scalable, distributed, high-throughput event-streaming architectures.

As Kafka runs as multiple stateful services, it would ordinarily be non-trivial to operate Kafka on top of Kubernetes without the automation provided by AMQ Streams. Red Hat AMQ Streams provides automation that removes the typical Kafka and Kubernetes stumbling blocks such as installation, configuration, and upgrades. OpenShift enables AMQ Streams event-streaming applications with automated container scaling, load-balancing, and recovery, while providing the ability to meet production requirements through rolling updates and canary deployments.

This automation enables enterprise IT teams to build and deploy applications faster in order to achieve a nimble, event-driven enterprise. AMQ Streams jumpstarts EDA initiatives with better integrations with the many other apps and frameworks running on OpenShift. This deep integration means that topics and users can be configured as the cluster is rolled out, not as a separate, manual process. Red Hat AMQ Streams simplifies the complexities of microservices, data streaming, event sourcing, and event processing.

AMQ Streams, with simplified deployment on OpenShift, is the best technology for event-driven enterprises that want to leverage Kafka's power. AMQ Streams includes pre-built container images for Apache Kafka and Zookeeper to streamline deployment, as well as operators for managing and configuring Apache Kafka clusters, topics, and users on top of Kubernetes. The combination of operators and pre-built container images enables IT departments to stand up a production-ready event-streaming platform in minutes.

AMQ Streams as a Member of the AMQ Family

Red Hat AMQ provides fast, lightweight, and secure event distribution for enterprise applications. AMQ components communicate via industry-standard protocols and support numerous programming languages and operating environments. Enterprises rely on AMQ as the event-driven foundation upon which to build their real-time business-critical applications.

AMQ is comprised of the following components:

- **AMQ Streams**, based on Strimzi and Apache Kafka, provides a distributed backbone that enables microservices and other event-driven applications to share data with extremely high throughput and extremely low latency.

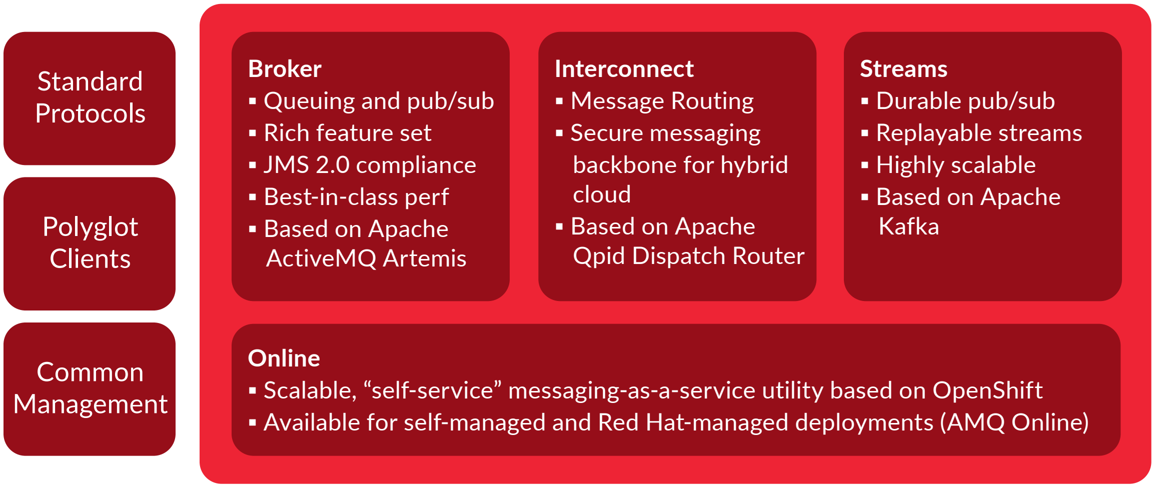
- **AMQ Broker**, based on Apache ActiveMQ Artemis, is a pure-Java multiprotocol event broker that is fast, efficient, highly available, and persistent. AMQ Broker is used as the messaging engine in the Red Hat Enterprise Application Platform, EAP. AMQ Clients is a suite of APIs that can be used to turn any application into an event-driven application. AMQ Clients is based on Apache Qpid JMS, Apache Qpid Proton, and Azure AMQP.NET Lite.

- **AMQ Interconnect**, based on Apache Qpid Dispatch, is a high-speed, low-latency event router. Enterprises use AMQ Interconnect to create fault-tolerant event networks that connect clients and brokers across the public and private cloud.

- **AMQ Online**, based on the enmasse project, is a self-service developer platform for provisioning event services when and where needed without the hassle of installation, configuration, and maintenance. Multiple development teams can provision the brokers and queues they need from a simple console.

Red Hat AMQ itself is a member of the Red Hat Integration product, allowing users to create integrations that make use of streams as well as synchronous interactions such as APIs from the same platform.

Red Hat AMQ Delivers Flexible Messaging for the Enterprise, Cloud, and Internet of Things



Standard Protocols

Polyglot Clients

Common Management

Broker
- Queuing and pub/sub
- Rich feature set
- JMS 2.0 compliance
- Best-in-class perf
- Based on Apache ActiveMQ Artemis

Interconnect
- Message Routing
- Secure messaging backbone for hybrid cloud
- Based on Apache Qpid Dispatch Router

Streams
- Durable pub/sub
- Replayable streams
- Highly scalable
- Based on Apache Kafka

Online
- Scalable, “self-service” messaging-as-a-service utility based on OpenShift
- Available for self-managed and Red Hat-managed deployments (AMQ Online)

Conclusion

Red Hat AMQ Streams, based on Apache Kafka, offers a distributed event-streaming infrastructure that enables microservices and distributed applications to share data with extremely high throughput and extremely low latency in a high-availability environment.

Running AMQ Streams on OpenShift provides a simplified and automated way to deploy, manage, upgrade, and configure a Kafka-on-Kubernetes ecosystem with continuous monitoring and high-availability features. This combination offers a resilient distributed backbone that allows microservices and other applications to share data with high throughput and low latency that serves as a powerful and flexible foundation for innovative and forward-looking enterprise software.

Enterprises can reap the benefits of an event-streaming application ecosystem built on AMQ Streams to achieve businesswide situational awareness and real-time, context-informed decision making. Event-streaming technologies are essential for building and maintaining real-time links with internal partners, business partners, suppliers, and customers. AMQ Streams on OpenShift is the foundation on which enterprises can build, deploy, and rapidly improve the software innovations that drive them forward. For more information, visit https://www.redhat.com/en/technologies/jboss-middleware/amq

RTInsights.com

Accelerate Your Business with Real-Time Insights

RTInsights is an independent, expert-driven web resource for senior business and IT enterprise professionals in vertical industries. We help our readers understand how they can transform their businesses to higher-value outcomes and new business models with AI, real-time analytics, and IoT. We provide clarity and direction amid the often confusing array of approaches and vendor solutions. We provide our partners with a unique combination of services and deep domain expertise to improve their product marketing, lead generation, and thought leadership activity.

Red Hat

Red Hat, a subsidiary of IBM, is the world's leading provider of open source software solutions, using a community powered approach to provide reliable and high-performing cloud, Linux, middleware, storage, and virtualization technologies. Red Hat also offers award-winning support, training, and consulting services. As a connective hub in a global network of enterprises, partners, and open source communities, Red Hat helps create relevant, innovative technologies that liberate resources for growth and prepare customers for the future of IT.