Financial Services' Use of EDA to Migrate to Real-Time

RTInsights.com
Accelerate Your Business with Real-Time Insights

Red Hat

The financial services industry is under great pressure to modernize the services they offer. They face increased competition from nimbler FinTechs, the growing risk from cyber threats and fraud, evolving regulatory requirements, and ever-rising customer demands for highly responsive innovative digital services.

At the heart of all these matters is a need for a cloud platform that can work with events and streaming data while integrating existing applications and systems into new processes and services. Increasingly, financial institutions are moving to cloud-native applications and event-driven architectures (EDAs) to address these issues. Such a cloud platform can position a financial institution to address today's issues and meet future challenges. Below are some of the specific areas where an EDA can help.

Moving to real-time services

As businesses strive to be more responsive and react in real time, they must incorporate events data and develop and deploy applications based on microservices and cloud-native architectures. When compared with alternative approaches such as batch processing, an event-driven approach seeks to capture an event and process immediately.

Digital consumers today demand that financial services firms revisit the onboarding process — to open a new banking account or buy an insurance policy. Stages involved are extremely important to start the relationship with the customer, requiring an ability to verify the customer's identity, assess their suitability as a customer, and verify the potential risk of fraud. It is vital to get it right and eliminate frictions and obstacles for the customer. They need to reduce the full onboarding digital service time from days to hours or minutes. However, most existing steps in that process operate in batch and are not able to respond immediately.

Event-driven solutions enable responsive applications, such as loan approval or payment processing, that batch systems do not. Event producers publish events, ordered by time of creation, forming an event stream. The stream can be distributed across the enterprise, and consumers may subscribe to the various event streams and use all or only the events they're interested in.

Organizations can make use of event streams in a couple of ways. They can perform stream processing, which is the continuous processing of an event stream (usually in real time and focused on a defined time window). They also can perform streaming analytics, a type of event stream processing that leverages machine learning to detect patterns, trigger actions, or produce other events. Use cases include real-time fraud detection and protection, customer onboarding, credit approval, and more.

Bottom line: An event-driven architecture facilitates the goal of capturing events as soon as they're raised, received, or available. However, some platforms are not adequate for real-time operations. They do not work in the cloud-native world. Right now, the most commonly used platform for an event-driven architecture is Apache Kafka. It is a popular approach to event-streaming backbones upon which an organization can build an event-driven architecture.

Such an architecture can support a broad range of applications across the entire FinServ spectrum.

Applying event-driven architectures in the payments industry

Real-time payment networks are now the norm in many markets, and they have made it possible to process payments almost instantly. It is expected to become the norm for many types of payments. As such, real-time payments can help financial services organizations provide new value to customers by eliminating the delay in payment processing. They also can act as an enticement for customers to move away from checks and other more costly payment methods.

However, many organizations find their existing payment infrastructures prevent them from achieving the full value of real-time payments. Often, it is difficult to scale their infrastructure to support ever-growing payment volumes. In many instances, their existing payment processing capabilities were developed for batch processing and cannot sustain the throughput to support a real-time service.

Moving to an EDA helps to decouple systems and bring agility to core business capabilities. First, it means different parts of the financial services organization can operate independently of one another, plugging new digital services into the real-time stream of events. That means that organizations can tap into the distributed nature of microservices architecture and benefit from the elasticity of the underlying cloud platform while improving the processing speed across the systems that participate in the payment transaction. The increased processing speed can reduce processing exceptions with instant feedback when the payment is initiated, creating a competitive difference between organizations that have invested and those that have not.

Second, because of the loosely coupled nature of an EDA, payment processing enables organizations to respond to industry changes quickly. They can make changes to relevant parts of an application while leaving everything else unchanged. One example where this comes in handy is an organization's support of ISO 20022, the common language and model for exchanging financial messages worldwide. SWIFT estimates that 80% of global, high-value payments by volume will be processed through ISO 20022. The standard is nearly two decades old and continues to produce updates every year. Keeping pace with such frequent changes would require great amounts of resources in a monolithic application architecture.

Similarly, many financial services institutions are impacted by the Revised Payment Services Directive (PSD2), a European Union Directive to regulate payment services and payment providers. The directive replaced the original Payment Services Directive (PSD) and will undoubtedly evolve over time.

In both cases, existing capabilities can make it difficult for organizations to quickly implement changes in message formats that enable real-time payment processing. Applying cloud-native technology to an event-driven architecture allows organizations to make changes as requirements evolve. Additionally, cloud-native integration means organizations can run multiple versions at the same time, gradually switch over to the new version, and avoid delays in rolling out the changes. And an EDA gives organizations the ability to quickly incorporate new technologies, such as the use of AI/ML on streaming data to improve fraud detection and anti-money laundering efforts, into their payment processes.

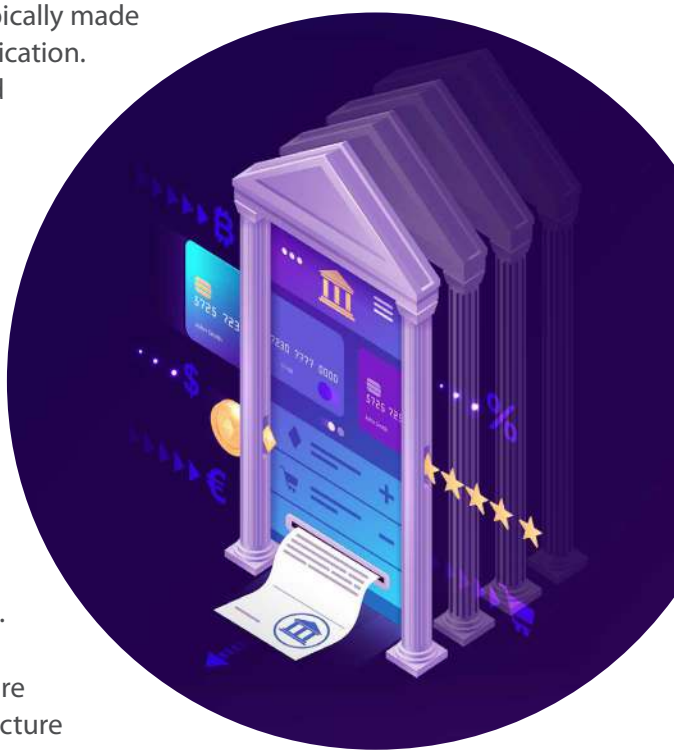
The Central Bank of Brazil (BCB) is a good example of the benefits of deploying an event-driven architecture at scale. The bank undertook a program to modernize the Brazilian Payments System (SPB), which manages the transfer of financial resources between banks for payments between individuals, businesses, and the government. That effort included Pix, a new instant payment network that uses ID keys or QR codes that are read by mobile devices. Using the network, individuals or companies can make or receive payments in seconds without needing debit or credit cards.

The payments network uses a distributed, highly scalable architecture based on Red Hat OpenShift, the leading enterprise Kubernetes platform, Apache Kafka with Red Hat AMQ. AMQ Streams, a component of Red Hat AMQ, offers distributed data streaming with high throughput and low latency for running, scaling, and managing applications. The platform cuts transaction processing time from hours to seconds and will serve as the foundation for anticipated future needs.

Applying event driven in the banking industry

Banks are under increased pressure to modernize operations due to higher customer expectations and market erosion from FinTech companies. An EDA and a cloud-native architecture enable financial institutions to expose existing core banking functions so they can be used with new front-end applications (e.g., consumer smartphones) and used as part of new applications. Such functionality allows banks to extend real-time capabilities to a wide range of operations, increasing efficiencies and improving customer experience.

Building an application that uses legacy applications or systems typically made use of middleware to integrate the existing system with a new application. The problem with such an approach was that the middleware could become a concoction of enterprise service buses (ESBs), object request brokers, and APIs over time. Anytime something needs to be updated or changed required extensive re-working and re-coding.



In contrast, modern approaches to core modernization are composed of independent business capabilities that work together. Elements of applications are services (microservices) that are loosely coupled, so changes can be made to one without making changes in others. Applications or processes run in software containers as isolated units, so they can be reused and do not have to be updated if other elements of an application change. And processes are managed by a central orchestration manager to improve resource usage and reduce maintenance costs.

An EDA complements the benefits of a core cloud-native architecture to provide support for events and streaming data. Such an infrastructure allows three types of modernization. Financial services organizations can extend, renew, or reinvent core services.

Extend

The extend approach adds new interface layers to existing core banking services, making them easily accessible to other cloud- and microservices-based applications through APIs. The architecture of the existing service remains largely unchanged, allowing a bank to protect existing investments while applying innovation and new processing and analysis methods to data.

Many organizations use this approach to provide customers with fast, easy, and real-time access to their accounts and services via the platform of their choice. Such undertakings might include linking back-end and event-driven systems to a new website, smartphone app, or voice system.

In each case, an EDA can act as the intermediary between the back-end applications and systems that create streams of data and the front-ends that consume them. The real benefit of a modern cloud-native application architecture is that the streams of data can be used by all front-ends using whatever technology they need. So, a web app developer can employ the latest design techniques and include such things as intelligent chatbots. Mobile apps can leverage advances in smartphone technology (e.g., location information, scanning capabilities, etc.). And phone systems can add automated voice assists, natural language processing, and other features.

Renew

The renew approach incrementally replaces existing core banking services with new, software-based versions from independent software vendors. The new version of the service is deployed on cloud-based infrastructure and may also use a microservices architecture.

For example, an organization might transparently combine industry-leading credit scoring services from a fintech with in-house risk controls and credit provisions capabilities running on a hybrid cloud platform. Some modern solutions can deliver the fast transaction rates needed. Many can also provide easy access to the transaction data needed to support new services and a more personalized customer experience.

An additional benefit is that the core services are now delivered via apps built for the cloud-native environment. That would mean they can run as independent microservices, which in turn can be consumed by other application elements.

Reinvent

The reinvent approach incrementally replaces existing core banking services with cloud-native, microservices-based versions running in an agile service mesh. Over time, cloud-native services will completely replace the traditional core banking systems, allowing an organization to take full advantage of the flexibility, control, and efficiency of containers and microservices.

Specifically, many of the systems used to provide core banking services such as deposit, loan, and credit processing are batch-oriented. A customer deposits a check, and it takes a day or two to clear. Similarly, with loan and credit processing, the customer fills out an online or mobile app form and applies. The information is passed to systems that batch process filings. The customer must wait for an answer.

Each of these core services can be replaced with modern versions that act on the data in real time and give the customer instant access to the deposited money or an on-the-spot loan approval or boost in their credit line. The modular nature of cloud-native apps and the real-time management or events data of an EDA make such transformations possible.

Focus on insurance

The insurance segment of the financial services industry faces similar pressures as its banking counterparts. Namely, FinTechs are encroaching on their markets, and customers have demanding expectations for instant everything. EDA and a cloud-native architecture can help speed the development of new applications and products, improve customer engagement, and migrate to real-time processing.

Core system modernization initiatives are the most critical component of their digital business strategies. With modernized core applications, insurers can work better, faster, and at the right cost point while also improving policyholder experience. These benefits increase as modernization efforts spread throughout processes and become better integrated with one another.

Major hurdles to modernization include data migration, integrations with up/downstream systems, cloud migration, and data conversion challenges. One of the most important aspects of EDA is that it fits well into existing environments. It is well-suited to common insurance industry core services applications that derive the most benefit from scalable and reliable real-time communication, such as stream processing and data integration.

EDA lets insurance industry developers build applications designed to react to, process, or transform multiple streams of events in real time. The advantages of EDA for stream processing applications include scalability to handle large amounts of streaming data and the reliability to ensure the lines of communication are kept open.

An example of what can be achieved in modernization is the work done by The Helvetia Group, an international insurance group with more than 5 million customers. The company ran applications on-premises on its servers but needed to enhance its development capabilities to stay competitive in the increasingly crowded insurance market.

It moved to a cloud-native software environment to build an engaging customer experience for new and existing applications and significantly enhance agility and time to market.

The scalability and availability of the cloud infrastructure solved another problem: The company's back-end systems lacked the availability to provide the instant response times expected by customers used to modern applications. With the move to a cloud-native and events-based structure, the company increased service uptime to over 99.9% and reduced time to market for new applications from months to weeks.

Integration advantages

Migrating from batch to real-time process via EDA and a cloud architecture brings additional benefits. The technologies make data available to a variety of systems.

For example, data and functions running on legacy systems can be made available as microservices consumed by other applications. For instance, a bank might use AI and ML on data streams to improve fraud detection in real time. Or, a customer loan application might draw on third-party data sources to expand the normal credit review and approval process.

With EDA, a stream engine can sit between an event generator and a variety of applications that might have use of that data. The apps that use that out of a stream engine do not have to be synchronized with the data generation side of operations. For example, a stream engine could accept continuous updates of a customer's account. Then different applications could make use of that data as needed.

The decoupled and asynchronous nature of an event-driven architecture enables the development of flexible, extensible, modern cloud-based applications. Such applications are essential for financial services organizations to move to the real-time operations they need to make customers happy and ward off competition from FinTech.

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 is the world s leading provider of enterprise open source software solutions, using a community-powered approach to deliver reliable and high-performing Linux, hybrid cloud, container, eventing, and Kubernetes technologies. Red Hat helps customers develop cloud-native applications, integrate existing and new IT applications, and automate and manage complex environments. A trusted adviser to the Fortune 500, Red Hat provides award- winning support, training, and consulting services that bring the benefits of open innovation to any industry. Red Hat is a connective hub in a global network of enterprises, partners, and communities, helping organizations grow, transform, and prepare for the digital future.