



# What's Driving Event-Driven Architecture (EDA)?

4 critical business drivers of the real-time data revolution

*An Intellyx eBook, for Red Hat*

**Jason English**

Principal Analyst, Intellyx

February 2022



**Intellyx**<sup>™</sup>



**Red Hat**



*We're on our way.  
Where will event-  
driven architecture  
take our enterprise?*

The transformation of conventional point-to-point integration technology to ***event-driven architecture (EDA)*** is already well underway. Whether enterprises are ready or not, they must answer business imperatives for greater agility and innovation.

Understanding what is driving this real-time data revolution is the first step toward making your organization ready to embrace and take advantage of EDA's strengths. This guide will take you there.

## **What is different about an event-driven architecture?**

EDA is not a new concept. Broadly, an EDA could produce any computing system designed to consume and produce a stream of events or respond to changes in state. EDAs have existed for years, handling some of the most mission-critical integration scenarios supporting financial, retail, media, travel and government agency applications and networks.

The ***asynchronous*** nature of real-world events is what guides the paradigm of EDA technology. Think about a complex real-time system such as a news feed, or weather. While data points in the form of news events or weather signals may be time-stamped when they appear, they could have actually arrived in any order, and at any time interval.

The ***publish/subscribe (or pub/sub)*** model of a typical RSS feed offers a good simple example of EDA in action. Whenever subscribers choose to look at a podcast or blog feed, it may publish different items or topics as they become available, but no direct message needs to be sent by either party before the subscriber determines if they want to view the content or download it for consumption.



By contrast, architectural approaches such as REST (representational state transfer) and integration backbones do not naturally support asynchronous events, as they often exclusively communicate through APIs over HTTP.

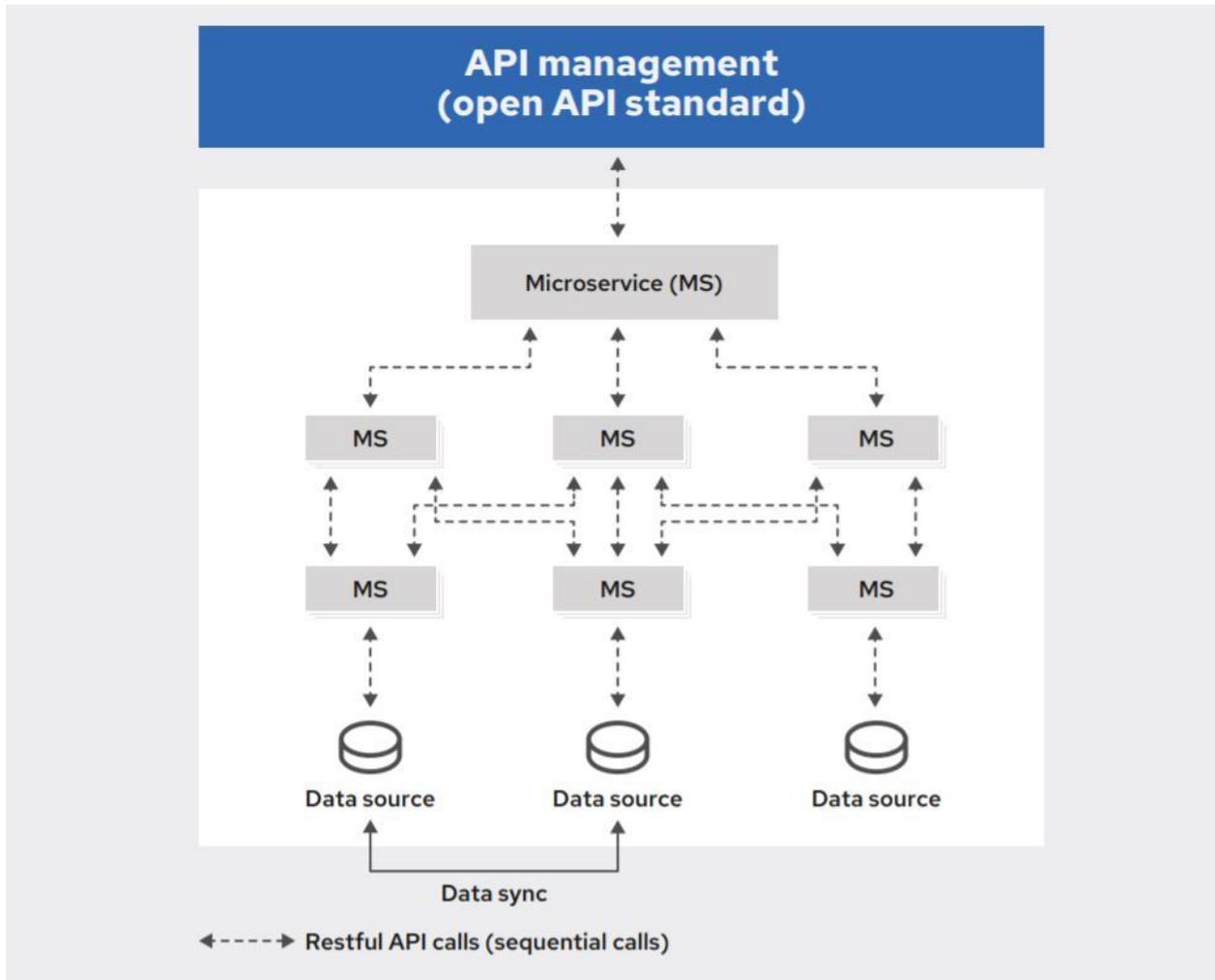


Figure 1. API management application architectures such as REST communicate message data through synchronous API request/response pairs over HTTP. (Diagram courtesy of Red Hat)

EDA however, can also encompass such **synchronous** protocols and **message-driven** approaches evident in REST. A sender can send an AMQP message with a request and data payload, and another system, for instance a JMS broker, receives the request, which triggers a response of a message back to the requestor with another payload. To an EDA application, an event is an event – however it is generated or responded to!

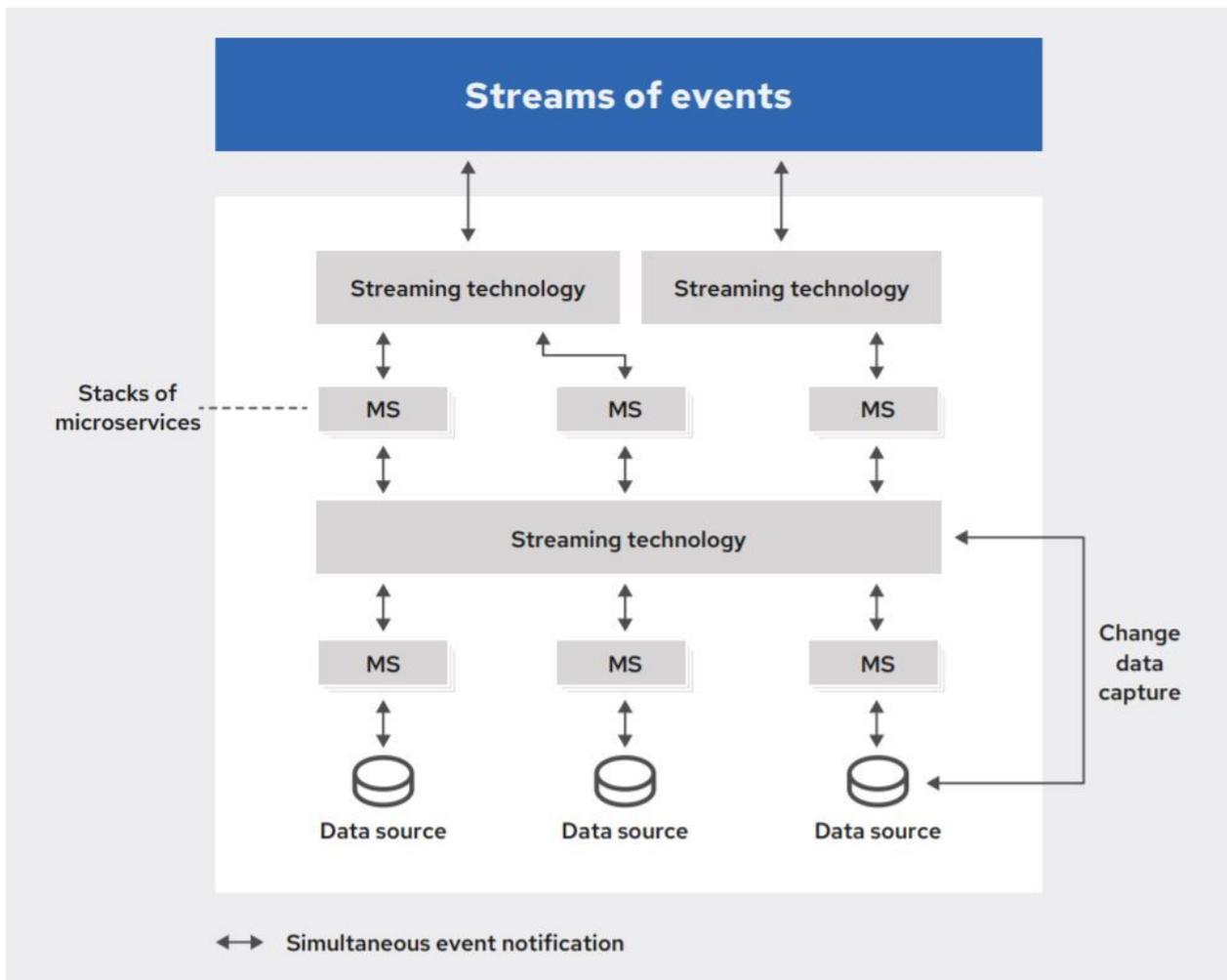


Figure 2: By contrast, EDA architectures can take in asynchronous events such as sensor updates and data changes, along with synchronous requests through event streaming technologies. (Diagram courtesy of [Red Hat](#))

Each **request/response** integration pair was once described as a 'connection' or 'pipeline' which offers a good representation of how data was synchronously moved between systems once a firm handshake is established. Old-school ETL (extract, transfer, load) data migration tasks operated in this way, but so do many modern applications, with direct API calls and point-to-point integrations from web and mobile apps to cloud-and-SaaS based data services.

REST-based, API-driven integration approaches appeared to be moving into fashion and replacing many earlier generations of EDA over the past decade or so, and newer 'born in the web' companies tended to design their systems in this direction.



However, while EDA's 'loose handshake' between systems may forgo some predictability, newer generations of EDA are proving to be very lightweight, flexible and highly performant integration methods indeed, which means we're likely to have both approaches active in the enterprise for the foreseeable future.

*Here's the four drivers propelling EDA's big comeback...*



## Driver 1: Rapid innovation and momentum

Enterprises can take advantage of the rapid rate of open source innovation to leapfrog earlier architectural constraints, and get a headstart on a battle-tested EDA application.

Who could have predicted that the little Linux OSS kernel would one day seed the clouds for an expansive ecosystem of innovative projects, partners and technologies? We're seeing open source take hold in every corner of the enterprise, from integration and networking standards to containerization, which ultimately led to today's microservices architectures and cloud-native development.

The introduction of the open source [Apache Kafka](#) in 2011 created an ideal scalable data platform for applications. Streaming data is ideal for EDAs, and especially relevant as a scalable source of state for stateless, ephemeral microservices as orchestrated by Kubernetes in conjunction with [Red Hat OpenShift](#). Quietly, technologists in the know started returning to EDA approaches to support real-time applications.

Now, with millions of person-hours contributed to open source software (OSS), enterprise customers often prefer OSS to commercial solutions because it mitigates vendor lock-in and de-risks investments in new technologies, as the most successful ones are more likely to continue to receive community support and improve.



*Open source projects propel the EDA revolution.*

When an OSS project is incubated or graduated by a foundation, that means it has already been battle-hardened through code reviews, experimentation, refinement and



real-world stress testing. In addition to Kafka, there are several OSS projects supporting other essential aspects of EDA, including event processing, application connectivity, infrastructure notification, change data capture and more in deployment today:

- **Infrastructure:** [Kubernetes](#), [Knative](#)
- **Storage:** [Ceph](#)
- **API/Applications:** [AsyncAPI](#), [Apicurio](#), [Camel](#), [Quarkus](#)
- **Messaging/streaming:** [Kafka](#), [Strimzi](#), [Debezium](#), [APIcurio](#), and more...

There's a free open source license on the software – but since EDAs are trusted for critical real-time business needs, customers still seek out value from vendors and service partners that can offer commercial support, certified packages, compliance coverage and more cost efficient deployed or hosted instances. This robust market for services ensures that vendors of all sizes will stay involved in supporting EDA.



## Driver 2: Real-time analytical intelligence

The best architectures shape technology to match the business intent it is used for, and nowhere is this more evident than in using EDA for many types of analytics work.

Earlier visibility into real-time data provides early warnings for changing conditions, ultimately powering better decisions and a competitive advantage.

**Take for instance a supply chain analysis scenario.** A major consumer goods manufacturer needs to interpret incoming retail order signals to forecast near term demand, while simultaneously planning manufacturing and logistics according to available supplier inventory and network capacity.

A supply/demand mismatch in analytics work could not only cause lost orders, it could cause inventory backlogs and the dreaded 'bullwhip effect' of a factory producing the wrong goods at the wrong time.

Fortunately, EDA with a data-centric design allows analytics functionality and data to scale faster, especially in complex multi-party environments, as consumers and producers of topic data can be added with minimal impact on existing applications. Kafka is perfectly suited for highly scalable, complex analytics workloads, whether for supply chains, financial analysis or system performance metrics.

Data scientists also use real-time data analytics to train and augment machine learning capabilities. Events are often first recorded in the application's systems of record and cloud data warehouses, so detecting changes within Db2, MongoDB, MySQL, Microsoft SQL, etc. becomes essential to feeding decision support.

Red Hat's [change data capture capability](#), based on the open source [Debezium project](#), identifies changes across an application's many data sources, and automatically publishes those changes to the Kafka event-streaming backbone, so AI/ML-based analytics programs gain deeper insights and make better recommendations based on current conditions.



## Driver 3: Modernizing responsive business operations

Digital transformation initiatives align the entire enterprise's technology stack around meeting customer needs. Within that, modernization of an organization's entire stack around EDA can bring high-value customer-facing services, better employee experiences, and operational efficiency 'features' to market faster.

Modernization shouldn't present an insurmountable up-front hurdle of rip-and-replace projects with tight, hardwired integration protocols. Since current business operations can seldom afford a pause for replatforming, you need to bring your own data and infrastructure, and merge in event-driven adaptations as you go.

The EDA application design pattern is well-suited for exactly this type of agile adaptation. Both new real-time data streams and changes to conventional systems of record produce events in the eyes of EDA, so this makes it easier to start layering on new event-driven business functionality.

Italy's national postal service, [Poste Italiane](#) saw the need to modernize its entire business stack in a hurry, due to the combined pressures of COVID and the need to transition a massive operation of thousands of offices servicing 35 million customers into a digital-first business.

Leveraging Kubernetes-based orchestration and containerization to encapsulate existing application services 'one bite at a time' alongside an EDA approach, agile teams have already brought forward 108 digital features such as sales front-ends and QR payment codes, and more than 200 new projects are currently underway. Most importantly, the team reduced each project's development time by a 3X factor, from 6-12 months to 2-4 months.

Many companies opt to avoid stream-based operations and scalability challenges altogether, by taking advantage of a cloud service, such as [Red Hat OpenShift Streams for Apache Kafka](#). Leveraging this SaaS model gives teams a fully hosted and supported EDA management platform with 'quick start' workflows built in to support common tasks, and a user interface, API, CLI, service registry, metrics and monitoring—all ready to run.



## Driver 4: The portable enterprise edge

Events happen all around us – and we are surrounded by the data events produce. A great migration of digital capacity is underway, away from centralized systems and data silos and even cloud infrastructures, out to the enterprise [edge](#).

The rapid adoption of 5G networks, increased miniaturization and IoT (Internet of Things) devices coming online is fueling this movement of business functionality closer to events where they happen. Retailers are replacing centralized systems and server closets with a portable 'local cloud-to-edge' unit that interacts with store registers and supports employees with on-the-spot services on their own devices.

Cloud-native doesn't mean cloud-infrastructure-only – it means teams can situate application workloads and data wherever they most ideally located. Kubernetes itself provides an ideal orchestration engine for managing containers that encapsulate an entire application estate, from the data center to cloud and the enterprise edge.

Portable edge cases of EDA are especially promising because they respect data gravity. Since change capture, event data processing and application decisions and execution are made closer to the events themselves, much lower event processing latencies and higher performance are well within reach. In smart cars, the on-board processing of real-time sensory and traffic event data must literally assist drivers with real-time responsiveness.

We are seeing excellent examples of portable EDA on the edge in [Industry 4.0](#) examples of connected planning and execution – where every piece of equipment on a factory floor, and every physical container and truck in transit becomes an IIoT (Industrial IoT) source of streaming event data that powers local operations while feeding global strategic planning work.



## The Intellyx Take

Is your enterprise driving the events, or are the events driving your enterprise?

As it turns out, we will continue to see innovation and growth in event-driven architectures far into the future. EDA is taking its turn in the limelight again due to the chaotic, fast-changing nature of real-world conditions, and the explosion of data that creates, all the way to the remote edges of the enterprise.

With so many strong use cases coming into focus, organizations are rethinking their entire application estate around event-driven architectures. This doesn't mean our existing REST-based services and core systems will go out of style, but it does mean we'll continue to find better ways to abstract them with microservices and event mesh.

There's much more to the EDA story than Kafka. Fortunately, customers have many ready options to get started down an event-driven path, from a suite of robust, proven open source technologies in play, as well as cloud service platforms offering out-of-the-box EDA functionality without operations headaches.



## About 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.

**Red Hat**

## About the Author (optional)

Jason "JE" English ([@bluefug](#)) is Principal Analyst and CMO at [Intellyx](#), a boutique analyst firm covering digital transformation. His writing is focused on how agile collaboration between customers, partners and employees can accelerate innovation.



In addition to several leadership roles in supply chain, interactive and cloud computing companies, Jason led marketing efforts for the development, testing and virtualization software company ITKO, from its bootstrap startup days, through a successful acquisition by CA in 2011. JE co-authored the book [Service Virtualization: Reality is Overrated](#) to capture the then-novel practice of test environment simulation for Agile development.

*©2022 Intellyx, LLC. Intellyx retains full editorial control of the contents of this document. At the time of writing, Red Hat is an Intellyx customer. Image sources: Cover composite by Jason English, using open source wikiCommons images, Mile1 licensed from [iStockphoto](#). Diagrams courtesy of Red Hat.*