



Build Next Generation Cloud Native Applications with the SMOKE Stack

JANUARY 2022



ABSTRACT

For enterprises to succeed in a hybrid world that moves at breakneck speed, they need platforms that are:

Serviceful
Multicloud
Open
Kubernetes-based
Event-driven

We call this combination the SMOKE stack and we think it provides the best foundation for next generation enterprise architectures.

CONTENTS

WHY SMOKE	4
SERVICEFUL	5
Serviceful architectures in action	5
Where the industry is headed	6
MULTICLOUD	7
Multicloud in action	7
Where the industry is headed	8
OPEN	9
Open Source in action	9
Where the industry is headed	9
KUBERNETES	10
The Benefits of Kubernetes	10
Kubernetes in action	10
Where the industry is headed	11
Knative expands Kubernetes serverless capabilities	12
EVENT-DRIVEN	13
Where the industry is headed	13
EDA in action - TriggerMesh case study	14
CONCLUSION	15
TRIGGERMESH CLOUD NATIVE INTEGRATION PLATFORM	15



WHY SMOKE

Businesses today are deploying to more clouds and consuming more cloud services than ever. Quickly integrating all these cloud endpoints and bringing on-premises apps to this mix is a huge challenge, and one for which traditional Enterprise Service Bus (ESB)-based integration systems are not well-suited. Instead, what's needed is an event-driven architecture that can react to fast-paced changes built by harnessing the power of Kubernetes and other systems like Knative.

The advent of serverless deployment models for businesses has been a revolution. The ability to run applications without managing servers allows teams the flexibility they need while helping keep down overhead. Because of these reasons and more, enterprises have continued to embrace serverless (or, as we prefer to say, serviceful) and build it into their real-time event-driven capabilities. They manage everything with Continuous Integration/Continuous Delivery (CI/CD), and they are using declarative APIs. At a foundational level, modern applications are containerized, run on Kubernetes, and managed via CI/CD.

For enterprises to succeed in this modern cloud native world, they need platforms that harness the power of the following five trends:

Serviceful
Multicloud
Open
Kubernetes-based
Event-driven

We call this combination the SMOKE stack, and we think it will serve as the foundation for successful cloud native enterprises. The SMOKE stack gives developers the flexibility to respond fast to changing markets without sidelining any of the on-premises, hybrid, and multicloud applications their business depends on to be successful. Each section in this paper gives our thoughts on:

1. Why each trend is a critical technology to enable enterprise application modernization
2. Where you can see it in action
3. Where we see the industry headed



In a cloud native world, applications are a combination of services—think of these services as similar to libraries in monolithic applications. Because serverless may hide the complexities of infrastructure, it allows developers to focus on building applications comprised of event-driven functions that respond to various triggers.

A Function as a Service (FaaS) platform (such as Amazon Lambda or Google Cloud Functions) lets users write small pieces of code that are executed based on a change in another system. The FaaS platform takes care of the trigger-to-function logic, passing information from one function to another, auto-provisioning of containers and runtime (when, where, and what), auto-scaling, identity management, and more. It also transparently handles the life cycle of functions and security.

Notable characteristics of serverless computing include:

- **Autoscaling, including scaling to zero:** Traditional cloud or on-premises applications run—and consume compute, storage, and networking resources—even when they are not in use. With serverless, when the function is not called, all compute, and other unused resources go idle and are free for reallocation. This is particularly valuable to organizations as demand for their services fluctuates. In an era of unprecedented uncertainty, every business needs the ability to iterate on features quickly and have complete confidence that their infrastructure will be as flexible as necessary.
- **Usage-based pricing:** Hand in hand with scaling to zero, when a function is not being used, you pay nothing. Serverless providers charge per function call.
- **Event-driven:** Serverless enables developers to focus on applications that consist of event-driven functions that respond to various triggers. These may be a database change, an event on a connected IoT device, or some activity from a customer on a website.

Serviceful architectures in action

The cost savings, scalability, and reduction in operational complexity from serverless are real.

These are just a few of the more notable examples of event-driven architecture in action.

- Digital publisher Bustle experienced approximately 84% cost savings by moving to a serverless architecture.
- Thomson Reuters processes 4,000 requests per second with serverless.
- Major League Baseball Advanced Media ingests, analyzes, and stores over 17 petabytes of data per second.

For us, the question is not “why serverless,” but rather, why aren’t enterprises using it more?

Why Serviceful?

Because TriggerMesh thinks serverless is much more than a FaaS offering. Serverless deployment methods can reach beyond the cloud to any application a business runs, including those on-premise and SaaS. This places even more focus on building application flows over infrastructure.¹

¹ <https://www.triggermesh.com/blog/follow-the-smoke-signals-to-enterprise-automation>

Where the industry is headed

We see serviceful as much more than FaaS. It is a key enabler of the agile and event-driven future we expect. We do not accept the notion that to take advantage of serverless, all your applications must be in the cloud or in a particular cloud. Our goal is to extend the reach of serverless to every application running anywhere—on-premises, cloud, and SaaS—so developers can worry less about infrastructure and focus on building the application flows their businesses need.



MULTICLOUD

It's easy to get caught up in the online war about whether multicloud is desirable or not. As VMware CTO James Waters recently explained, different operational contexts usually explain whether one is pro or anti-multicloud. Those who argue against multicloud are often very focused on data gravity, with heavy investments in one particular cloud data model. On the other hand, organizations investing in DevSecOps tend to care more about what James calls "Developer Gravity." For these companies, the goal is to have a consistent set of DevSecOps tools and processes across all clouds. Most larger organizations fall somewhere between the two extremes.

Enterprises focused on consistent developer tooling across all environments need a way to define an event flow that is agnostic to where the service is running.

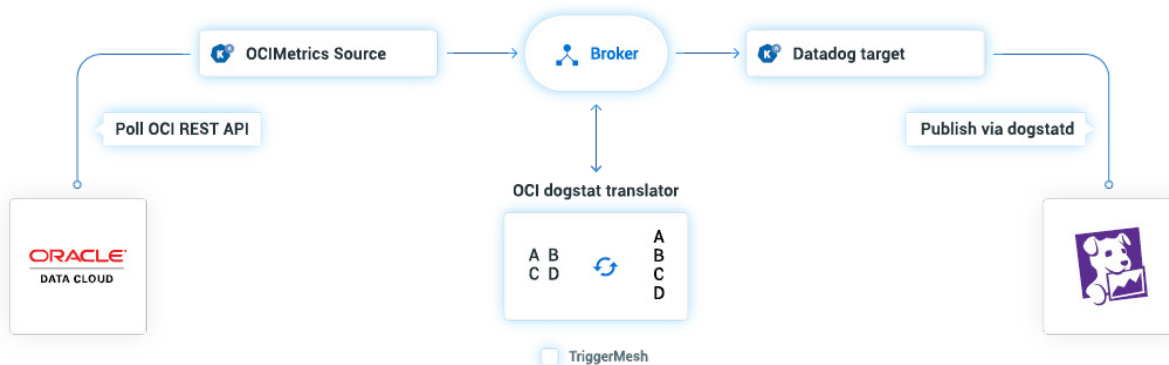
"Enterprises focused on consistent developer tooling across all environments need a way to define an event flow that is agnostic to where the service is running."

Multicloud in action

In naming TriggerMesh a [2021 Cool Vendor for Cloud Computing](#), Gartner notes that even while enterprises embrace multicloud, most are unprepared for the complexity it brings. Gartner further observes that "Recent innovations in containers and serverless platform technologies enable a new breed of providers to adopt a modern, scalable and portable approach to solving multicloud integration and management challenges."

Take the TriggerMesh customer, a leading risk and regulatory reporting provider. Although they rely heavily on AWS, they have several customers with databases in Oracle Cloud. This meant they needed to extract Oracle Cloud Metrics events and route them in real-time to their Application Performance Management system, Datadog. They turned to TriggerMesh to use CloudEvents to integrate Oracle Cloud and Datadog.

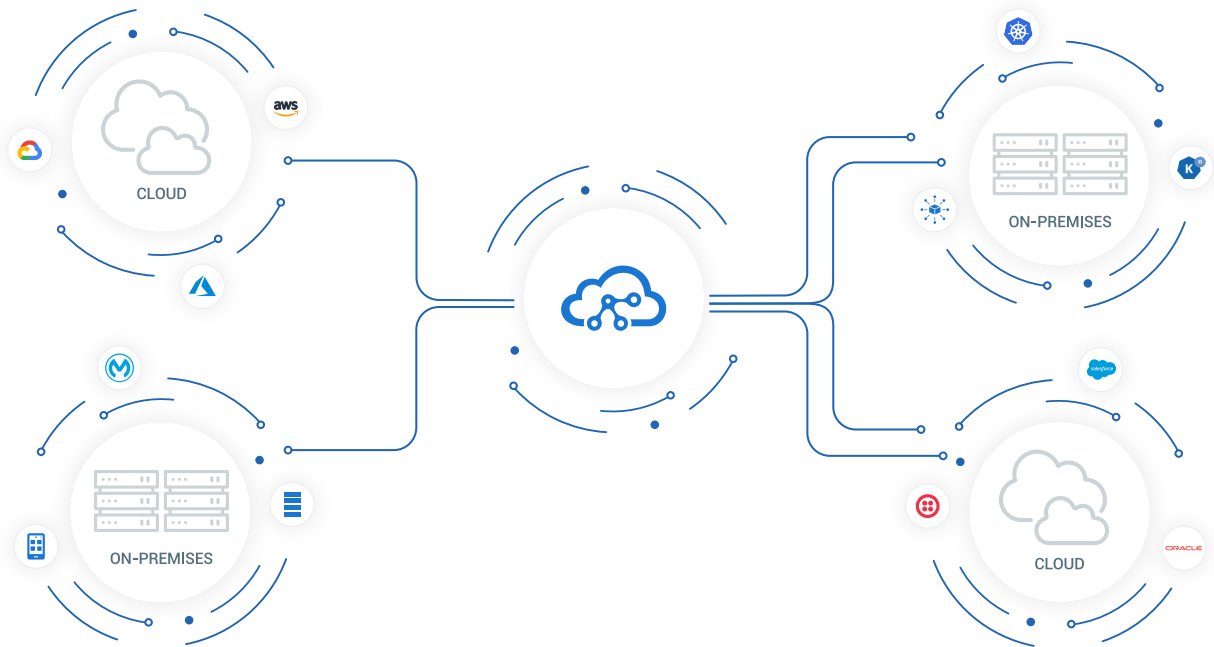
TriggerMesh allows this company to leverage the power behind Kubernetes and event-driven architecture to make sure they can meet more demands from customers.



AN OPEN SOURCE SMOKE STACK ROUTES EVENTS IN REAL-TIME FROM ORACLE CLOUD TO DATADOG

Where the industry is headed

Enterprises are finding multicloud both a benefit and a necessity. With that, integrations can no longer be single-point solutions but strategic and systematic. DevOps and product owners use TriggerMesh to create and easily maintain event-driven hybrid cloud integrations quickly. Our modern, self-service platform connects SaaS apps, cloud services, serverless functions, and even legacy applications. You can declaratively build integrations as code™ and deploy and manage those integrations with your existing GitOps workflow. We also make it easy for cloud operators to create integrations through our menu-driven low-code editor.



THE TRIGGERMESH EVENTS AND TRIGGERS MODEL IS FLEXIBLE AND EASY TO ADOPT. AS AN OPEN SOURCE SOLUTION, TRIGGERMESH IS FULLY VISIBLE AND CONTROLLED BY THE ENTERPRISE.

TriggerMesh Bridges

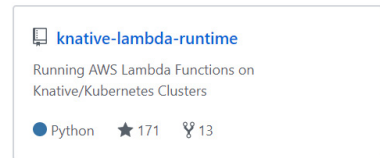
The cornerstone of the TriggerMesh solution is our serverless event bus that consumes real-time events from everywhere and uses them to trigger corresponding functions on any FaaS, Kubernetes cluster, or on targets like Salesforce, Twilio, or Slack. Bridges are connected application flows built on top of Kubernetes.

Event Sources for bridges can be legacy systems (VMware Vsphere, IBM MQ, Solace, OracleDB, etc.); public cloud services (AWS, Azure, GCP, etc.); custom applications (e.g., eCommerce platforms); and source control and other popular cloud apps, such as Twilio, Salesforce, and Apache Kafka. Developers use the TriggerMesh Controller to compose Bridges from our ever-expanding library of Sources, Brokers, and Targets, or they can easily define their own.

Bridges represent an extensible framework for building event-driven flows from any application to any other application or service and running across any cloud or on-premises. Built as an API extension to Kubernetes, a Bridge is a single manifest that contains an arbitrary number of components, each of which must be a valid Kubernetes object. Developers can use any of our dozens of existing sources and targets to create TriggerMesh Bridges. Or use our integration wrapper to turn any application into a TriggerMesh Source or Target easily.



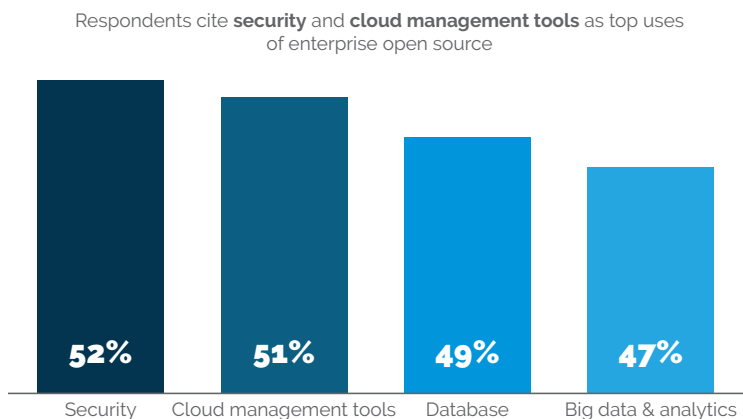
Solving the problem of cloud native integration depends on open specifications and open source. TriggerMesh is an open source solution built on the thriving ecosystem of open source projects. Our heritage is open source; TriggerMesh is one of the top 5 contributors to the upstream Knative project we depend on, alongside Google, IBM (incl. Red Hat), and VMware (incl. Pivotal). Our commitment to open source doesn't stop there. We have released several innovations under the Apache v2 license, such as our Knative Lambda Runtime.



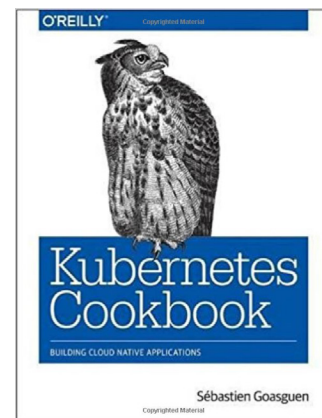
We also embrace open specifications like [CloudEvents](#). Industry-standard open specifications are critical to giving enterprise developers maximum flexibility when building serverless cloud native applications.

Open Source in action

Open source is everywhere, and enterprises have bought into the promising results already demonstrated by the movement. In Red Hat's [The State of Enterprise Open Source 2020 Report](#), 95 percent of respondents say open source is strategically important. After Security, Cloud Management Tools are the next most popular use for open source identified by IT leaders surveyed by Red Hat.



SOURCE: WWW.REDHAT.COM/EN/ENTERPRISE-OPEN-SOURCE-REPORT/2020



TRIGGERMESH CO-FOUNDER
SEBASTIEN GOASGUEN HAS BEEN
A PART OF THE KUBERNETES
REVOLUTION FROM THE START.

Where the industry is headed

Open source projects are giving enterprises capability, control, and confidence. The outlook for the future is that technology stacks will be “as-code” and free or proprietary lock-in. Integrations without capability, control, and confidence can cripple critical operations. Building on [Kubernetes](#), the open-source container orchestration system, and its serverless platform, Knative, enterprise technology stacks are extending the ability of organizations to control their own destiny. We believe these technologies are essential for businesses in the future with event triggered architectures. Some additional bona fides: TriggerMesh Co-founder Sébastien Goasguen is an Apache Software Foundation committer, a former VP of the Apache CloudStack project, founded Kubeless, one of the first Kubernetes-native open source serverless frameworks, and is a sought-after speaker and technical resource.

Our CEO and Co-founder, Mark Hinkle, ran the open-source office at Citrix, and served as a VP at the Linux Foundation as well as Executive Director of the Node.js Foundation.





In the push to satisfy customers' unpredictable demands, microservices architectures are one of your greatest allies. Microservices are independently packaged and deployed via containers. As microservices applications grow, orchestrating the containers and their dynamic access to compute, storage, and networking can become complex. The leading industry solution to handle this complexity is [Kubernetes](#), a solution supported by every major cloud and infrastructure vendor. Many observers believe, and we agree, that Kubernetes is fast becoming the operating system for the cloud.

Kubernetes is an open source container orchestration system for automating deployment, scaling, and managing containerized applications. Initially designed by Google and released as open-source in 2014, the Cloud Native Computing Foundation now maintains it.

The Benefits of Kubernetes

The popularity of Kubernetes has grown steadily over the last couple of years. A [CNCF 2020 report](#) found an extraordinary 300% increase in its production usage from 23% in March 2016 to 92% in July 2020. This continues a steady rise from 78% last year and 58% in 2018.

Kubernetes provides out of the box capabilities that allow organizations to accelerate delivery cycles and rapidly scale their operations:

- Application resiliency and responsiveness by replicating services under load with selective burst application components
- Scaling the application across many servers in a cluster based on the dynamic user traffic. Kubernetes' autoscaler provides horizontal scaling by replicating pods to multiple nodes
- Creation of symmetric environments to support various needs. Kubernetes makes the application easily portable across these environments
- Application reliability and stability with liveness and readiness checks
- Application security with data encryption, vulnerability scanning, mutual TLS, and other capabilities
- Rolling updates which can offer zero downtimes

Kubernetes in action

According to the [VMware State of Kubernetes 2021 report](#), production use of Kubernetes is growing from 59% in 2020 to 65% in 2021. Companies with more than 500 developers were more likely to be running all or mostly containerized workloads in production (78%).

The companies with the most clusters had the most nodes per cluster and were also more likely to be using Kubernetes in production. In companies with more than 10 clusters, 41% reported having 20+ nodes per cluster.

And while the majority of VMware's findings can be viewed as positive, integration challenges impede progress. "Integrating new technologies with existing systems" jumped from 35% in their 2020 report to 42% for 2021. It was the only impediment to increase significantly.

Where the industry is headed

Kubernetes is undoubtedly going to be employed more and more in the coming years. Integration will be a growing concern for each new implementation. TriggerMesh provides developers with automation to simplify building application integration flows on Kubernetes. By providing a bridge from legacy applications to Kubernetes, we streamline many low value activities so developers can easily act in a cloud native fashion in hybrid and multicloud environments.

For example, Red Hat OpenShift customers can use the TriggerMesh Operator to configure their OpenShift workloads to respond to events from practically any SaaS, cloud service, or on-premises application. [VMware Cloud native Runtimes for Tanzu Advanced](#) includes the open-source TriggerMesh Sources for Amazon Web Services to extend their Knative serving and eventing. Using our [Knative Lambda Runtime \(KLR\)](#), you can deploy Lambda functions on [Google Kubernetes Engine](#). In an excellent Medium article (reposted with permission on the TriggerMesh blog), Suganya, a Cloud Architect at Search Engineering, provides a [detailed technical write-up](#) of how she and her team used the KLR to help migrate a client from AWS to Google Cloud.

“Suganya, a Cloud Architect at Searce Engineering, provides a [detailed technical write-up](#) of how she and her team used TriggerMesh to help migrate a client from AWS to Google Cloud.”

Knative expands Kubernetes serverless capabilities

Knative is an open source Kubernetes-based serverless platform that provides a set of middleware components to build modern, source-centric, and container-based applications that can run anywhere: on-premises, in the cloud, or even in a third-party data center.

Knative does two things: serving and eventing.

- **Knative Serving** builds on Kubernetes to support deploying and serving of serverless applications and functions. Serving provides the autoscaling—including scale to zero—feature of FaaS, as well as fine-grained traffic control using modern network gateways.
- **Knative Eventing** provides building blocks for consuming and producing events that adhere to the CloudEvents specification developed by the CNCF Serverless Working Group. It includes abstractions for event sources, and decoupled delivery through messaging channels backed by pluggable pub/sub broker services.

Knative in action

Google Cloud Run, Red Hat OpenShift Serverless, and VMware Cloud native Runtime are all based on Knative. TriggerMesh is also built on Knative. Preliminary emphasis and discussions of Knative centered on Knative Serving. This is what allows you to get nearly unlimited scalability for workloads in your Kubernetes clusters, including scale to zero. Perhaps more importantly, Knative Eventing enables you to weave together any application using events. The combined power of serverless and eventing allows developers to quickly integrate any application or data source running anywhere in a loosely-coupled, event-driven way.

Where the industry is headed

Open source and open specifications make it so you can copy and paste the exact same YAML manifest from Google Cloud Run directly into TriggerMesh and get the same running service. This is the power of portability and standardization that Knative makes possible. But it doesn't necessarily make it easy.

TriggerMesh is an open source solution that simplifies creating hybrid and multicloud application integration flows. You can select from our growing library of event Sources, Brokers, and Targets to compose application integration flows—what we call Bridges—regardless of where the Source and Target applications are running. And if we haven't yet added the source or target you need, you can create it once and then add it to your private repo for reuse by your colleagues. Alternatively, you can create integration as code™ using either our declarative API or our new TriggerMesh Integration Language (TIL). You can combine this with your existing CI/CD pipelines for a fully-automated GitOps approach to cloud native integration.

EVENT-DRIVEN

Think of event-driven architecture (EDA for short) as enterprise service bus meets microservices architecture.



Monolithic applications simply cannot keep up with the need to rapidly bring new features and services to market. Microservices represented the first step to providing a quicker way to develop and deliver business-aligned services. As the number of these services grows, the next step is to integrate them through event notifications. And this motivates enterprises to embrace EDAs.

EDAs are composed of loosely-coupled, distributed services connected by asynchronous events. These systems are perfect for today's uncertain times because new services and data sources can be added or removed dynamically without disrupting the existing application flows.

We believe EDAs are the best way to take advantage of the explosion of new cloud services and the exponential growth in data volume and sources. The opportunities for businesses to leverage data and services to build apps that deliver better experiences are only limited by the flexibility of their architecture.

What is an Event?

The CloudEvents specification,³ an industry-led effort, defines Event as:

A data record expressing an occurrence and its context. Events are routed from an event producer (the source) to interested event consumers. The routing can be performed based on information contained in the event, but an event will not identify a specific routing destination. Events will contain two types of information: the Event Data representing the Occurrence and Context metadata providing contextual information about the Occurrence. A single occurrence MAY result in more than one event.

³ <https://github.com/cloudevents/spec/blob/v1.0/spec.md#notations-and-terminology>

"The opportunities for businesses to leverage data and services to build apps that deliver better experiences is only limited by the flexibility of their architecture."

Where the industry is headed

There has been an increasing need for real-time data handling within the enterprise space. These needs are wide-ranging, whether these needs are freshness of data, a change data capture issue, or an event-driven workflow issue. Examples abound, such as [Shopify's](#) redesign of their data warehouse using [Debezium](#).

To address this need, most enterprises are turning to Apache Kafka, arguably the leader in message brokering and data streaming. These same enterprises often also use Kubernetes because it is the de-facto platform for managing containerized workloads. The question now becomes how to best unite Kafka and Kubernetes to meet urgent business requirements.

Uniting these two fundamental technologies requires a declarative API to describe what to do with your Kafka messages. Instead of writing your own Kafka client consumers, packaging that in your code, etc., it is simpler to define a Kafka source declaratively (i.e., A Kafka topic is the source of a message) and set the target for that message in a Kubernetes manifest. TriggerMesh makes integrating with Kafka as both the source or destination of events easy to deploy and manage.

With TriggerMesh, developers can quickly build event-driven, cloud native integrations that link public cloud services with on-premises and private cloud workloads. We believe this is the most pragmatic way for companies to continue leveraging high-investment, on-premises systems while also accelerating the move to cloud for digital transformation.

"With TriggerMesh, developers can quickly build event-driven, cloud native integrations that link public cloud services with on-premises and private cloud workloads. We believe this is the most pragmatic way for companies to continue leveraging high-investment, on-premises systems while also accelerating the move to cloud for digital transformation."

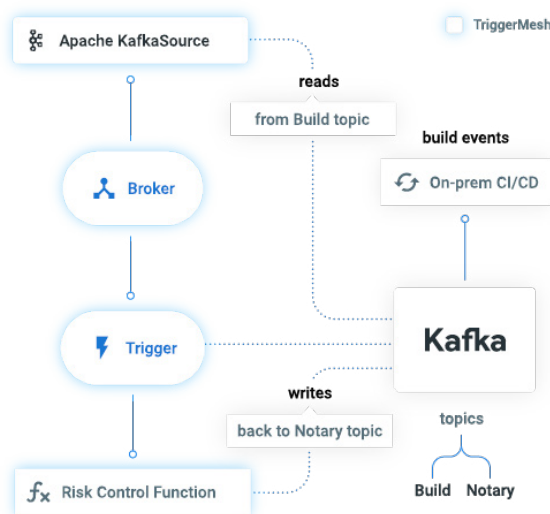
EDA in action - TriggerMesh case study

PNC Bank Automates Software Supply Chain Security with TriggerMesh Compliance audits shrink from one month to a few hours

The seventh largest US bank in terms of assets, PNC is a leader in applying DevOps, AI/ML, and cloud-native infrastructure to improve operational efficiency and speed their digital modernization.

The PNC Portfolio Management team needed to automate and accelerate the Software Supply Chain security and risk control process to reduce application deployment time and provide a governance audit trail. The formerly manual compliance processes took 30 days, slowing developer productivity and delaying feature releases.

PNC automated the previously manual processes by codifying them into serverless functions and leveraging TriggerMesh to build an on-premises event-driven application flow. The team uses the TriggerMesh declarative API to build an application flow between Apache Kafka and Jenkins that trigger the bank's serverless Policy-as-Code application to route pass/fail status back to Kafka. These event-driven application flows use TriggerMesh's integration-as code along with continuous delivery to build a high performance, reproducible EDA that provides real-time feedback for critical infrastructure.



Now, the previously 30-day process runs automatically and finishes in near real-time, allowing PNC to keep up with the surge of new applications and services while maintaining 100% security and compliance reporting.

CONCLUSION

Serviceful, multicloud, open, Kubernetes-based, and event-driven. Armed with the powerful technologies in the SMOKE stack, enterprises of every stripe can modernize their applications and tame multicloud complexity. TriggerMesh is fully aligned with each of the five trends in this paper, and we invite you to join us on [GitHub](#) or [schedule some time with our engineers to discuss your integration needs](#).

TRIGGERMESH CLOUD NATIVE INTEGRATION PLATFORM

TriggerMesh accelerates the move to an event-driven architecture. TriggerMesh is the leading Kubernetes-based integration platform to connect applications and build data flows. Our open source API helps you combine on-premises systems and cloud services in an automated manner. TriggerMesh helps you rein in cloud complexity while you build the event-driven systems your business needs to keep up with customers and ahead of competitors.

With TriggerMesh, enterprises can:

- **Enable Cloud Native Developers.** Our Knative-powered platform provides an easy, standard way to create cloud native applications.
- **Integrate the Modern Way.** Proprietary integration is slow, expensive, and relies on consultants. The TriggerMesh open source API and integration as code automate the combination of on-premises systems and cloud services.
- **Connect Real-time Data Flows.** For enterprises seeking to modernize applications and take action in real time, adding TriggerMesh to Apache Kafka simplifies the creation and maintenance of data integrations. The TriggerMesh API is much simpler than Kafka Connect.



CONTACT

Sebastien Goasguen
Co-Founder and Head of Product
Twitter: @sebgoa

Mark R. Hinkle
Co-Founder & CEO
Twitter: @mrhinkle

triggermesh.com

