TRIGGERMESH
CLOUD NATIVE INTEGRATION FOR KUBERNETES

Build Next Generation Cloud Native Applications with the SMOKE Stack

JULY 2021

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

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. What's needed instead is a cloud native approach to integration, built using the same modern "as-code" approaches that enterprises are adopting for their other mission-critical workloads.

Enterprises are embracing serverless (or, as we prefer to say, serviceful) and building it into their real-time event-driven capabilities; they are managing everything with Continuous Integration/Continuous Delivery (CI/CD); and they are using declarative APIs. At a foundational level, this means that modern applications are containerized, run on Kubernetes, and are managed via CI/CD.

For enterprises to succeed in this hybrid and multicloud 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.

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

SERVICEFUL

In a cloud native world, applications are a combination of services—think of these services as similar to libraries in monolithic applications. Because serverless abstracts the complexities of infrastructure, it allows developers to focus on building applications comprised of event-driven functions that respond to a variety of 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 run-time (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 full confidence that your 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 a variety of 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. Digital publisher Bustle experienced approximately 84% cost savings by moving to a serverless architecture.[1] 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. And these are just a few of the more notable highlights.

For us, the question is not "why serverless", but rather, why aren't enterprises using it more?

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.

1 https://aws.amazon.com/solutions/case-studies

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.[2] 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.

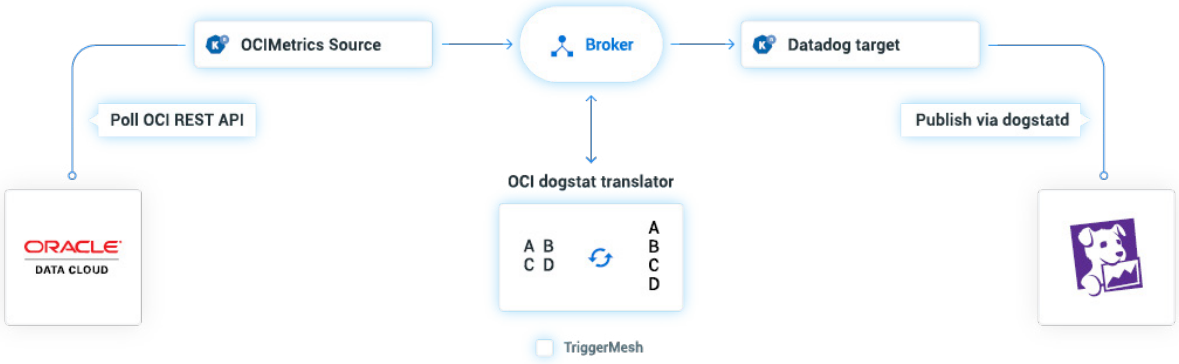
"Enterprises focused on consistent developer tooling across all environments need a way to define an event flow that is agnostic to where the service is running."

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 multi-cloud integration and management challenges."

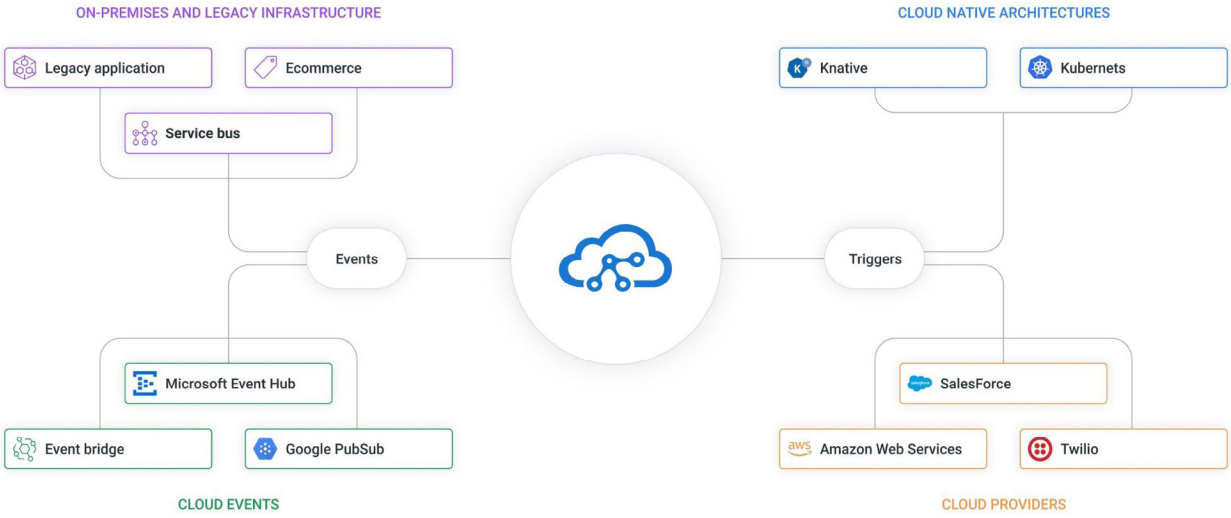
Take TriggerMesh customer AxiomSL, a leading risk and regulatory reporting provider. Although they rely heavily on AWS, they have several important customers with critical databases in Oracle Cloud. This meant they needed to extract Oracle Cloud Metrics events and route them in real time to their Application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



2 James shared his thoughts with Google Cloud's Richard Seroter

Where the industry is headed

We call TriggerMesh a cloud native integration platform. DevOps and Product Owners use TriggerMesh to quickly create and easily maintain event-driven hybrid cloud integrations. 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.



TriggerMesh Bridges

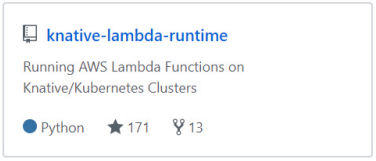
The cornerstone of the TriggerMesh solution is our serverless event bus that consumes real-time events from everywhere and uses them to trigger corresponding functions on any FaaS, Kubernetes cluster, or on targets like Salesforce, Twilio, or Slack. Bridges are connected application flows built on top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

OPEN

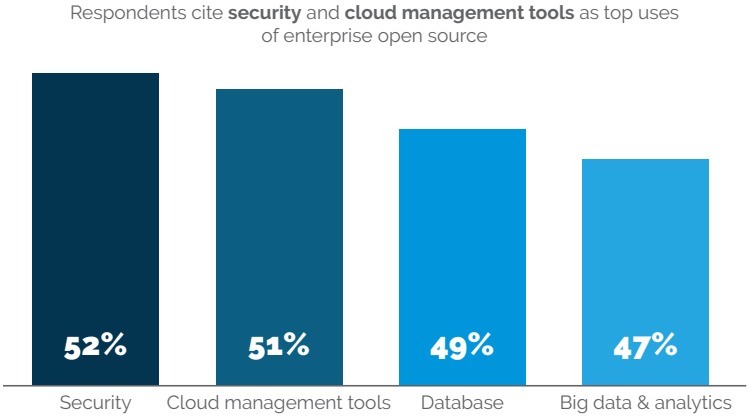
Solving the problem of cloud native integration depends on open specifications and open source. This is why TriggerMesh is one of the top 5 contributors to the upstream Knative project on which we depend, alongside Google, IBM (incl. Red Hat), and VMware (incl. Pivotal). Our commitment to open source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



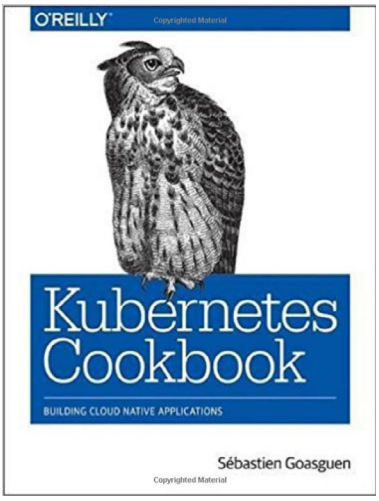
We also embrace open specifications like CloudEvents. Industry-standard open specifications are critical to giving enterprise developers maximum flexibility when building serverless cloud native applications.

Open Source in action

Open source is everywhere, and enterprises are sold on it. 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



Where the industry is headed

We are all-in on open source cloud native technologies. We are fully behind Kubernetes, the open source container orchestration system, and its serverless platform Knative. Some additional bona fides: TriggerMesh Co-founder Sebastien 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.

kubernetes

CLOUD NATIVE COMPUTING FOUNDATION

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.

We have open sourced much of our supporting technology, including our Sources for AWS (SAWS), and our Knative Lambda Runtime (KLR) that lets you deploy Lambda functions on Google Kubernetes Engine. SAWS is available as part of VMware's Cloud Native Runtimes for Tanzu Advanced VMware, so it's a snap to consume events from AWS services and send them to workloads running within a Tanzu cluster.

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, which is 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 management of containerized applications. It was originally designed by Google and released as open source in 2014, and is now maintained by the Cloud Native Computing Foundation.

The Benefits of Kubernetes

The popularity of Kubernetes has grown many fold 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.

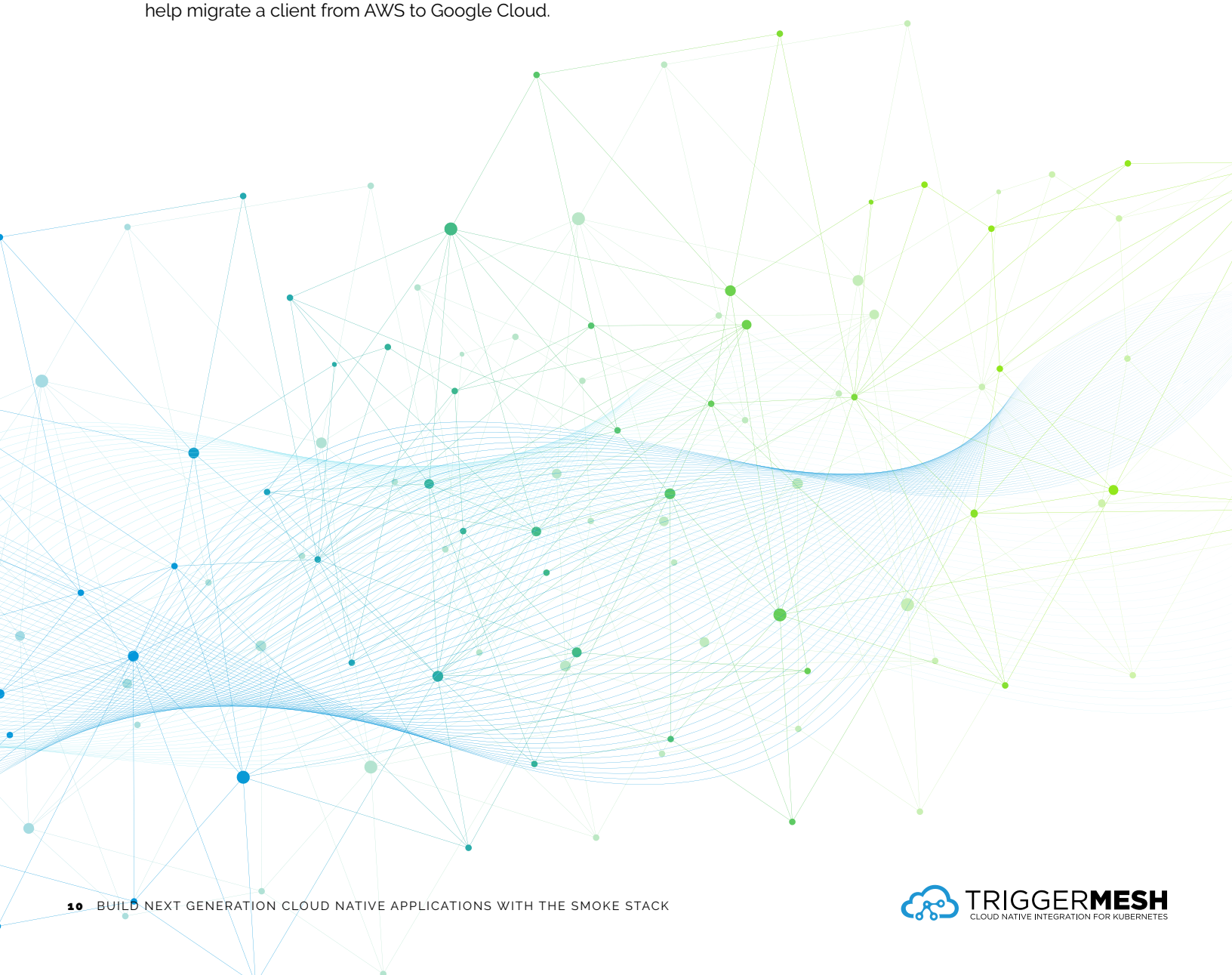
And while the majority of VMware's findings can be viewed as positive, integration challenges impede progress. Integrating new technologies with existing systems jumped from 35% in their 2020 report to 42% this time. While this number is headed in the wrong direction, it may also be the result of continued momentum.

Where the industry is headed

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 Searce 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 the KLR to help migrate a client from AWS to Google Cloud."



Knative

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 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 allows 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 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 codeTM 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 is motivating 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 sources and volume. 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.

What is an Event?

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

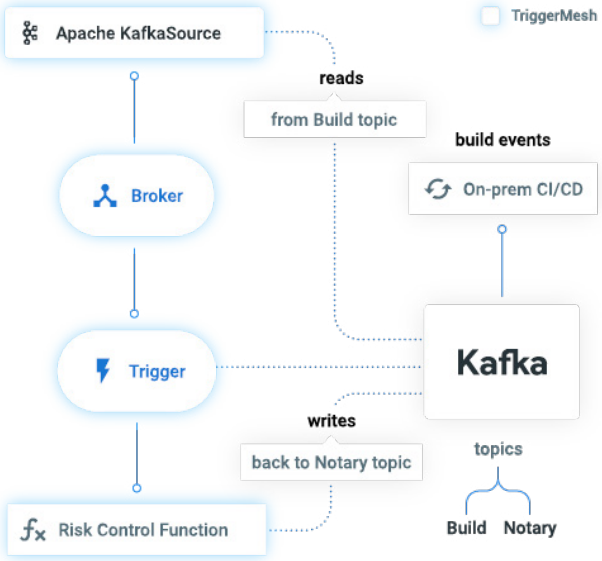
A data record expressing an occurrence and its context. Events are routed from an event producer (the source) to interested event consumers. The routing can be performed based on information contained in the event, but an event will not identify a specific routing destination. Events will contain two 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

EDA in action - TriggerMesh case study

The Enterprise Architecture team at one of the largest diversified financial services institutions in the United States needed to automate and accelerate the risk control evaluation process. With dozens of development teams around the company moving to CI/CD, they needed a way to deliver automated governance as a service so that all development teams could meet the business's need for new features while adhering to strict compliance requirements.

Using TriggerMesh's integration as code, the bank implemented a serverless event-driven architecture to automate risk control evaluation and attestation authoring. Events from multiple different sources such as BitBucket and Jenkins trigger the bank's Policy as Code (PaC) functions, with the results sent back to a notary service via Kafka.



"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."

Thanks to TriggerMesh, all event processing code runs as auto-scalable serverless functions. All results from the risk control evaluation are stored and available for audit. This allows the bank to move forward aggressively with real-time event-driven systems that increase operational efficiency by automating repetitive tasks and reducing application deployment time to production.

Where the industry is headed

There has been an increasing need for real-time data handling within the enterprise. Whether this need is a freshness of data issue, 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."

CONCLUSION

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 try TriggerMesh Cloud free for 30 days at www.triggermesh.com/trial

TRIGGERMESH CLOUD NATIVE INTEGRATION PLATFORM

The TriggerMesh cloud native integration platform allows you to *integrate* services, *automate* workflows, and *accelerate* the movement of information across your organization. TriggerMesh acts as a catalyst for digital transformation, enabling organizations to build event-driven applications out of any on-premises application or cloud service. With TriggerMesh, you can integrate any system, automate workflows, and respond faster to changing business needs.

For example, with TriggerMesh you can:

- **Stream data** from your cloud services to a central data lake
- **Route, transform, and split streams** from Apache Kafka
- **Sync** your on-premises ERP system with Salesforce
- **Add custom monitoring** in Datadog from cloud and on-premises sources
- **Run an event-driven serverless installation** on your on-premises Kubernetes infrastructure

For all the same reasons why enterprises are placing the cloud at the center of their application strategy—portability, scalability, and speed—we are building our solutions using modern cloud native architecture. This allows you to connect applications and services from the data center to the cloud, or clouds, of your choice with unrivaled speed and flexibility.

Benefits

- **Powerful, custom integrations made easy—**Easily connect SaaS, cloud, and on-premises applications with serverless and cloud-native architectures in a few simple steps
- **Modernization—**Bring your legacy applications to the cloud and leverage your existing IT investment
- **Programmatic—**Improve developer productivity and provide consistency by integrating services through a standard platform

CONTACT

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

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

triggermesh.com

TRIGGERMESH
CLOUD NATIVE INTEGRATION FOR KUBERNETES