



API LIFECYCLE MANAGEMENT · THE PLATFORM ENGINEERING ERA

Stop paying a premium for features no one uses.

A strategic comparison of **MuleSoft**, **Kong**, and **Backstage + Envoy Gateway** – and why an open, cloud-native foundation delivers the capabilities your teams actually use, at a fraction of the cost.

THE CONTENDERS



MuleSoft

\$\$\$ Premium license



Kong

\$\$ Commercial license



RECOMMENDED

**Backstage + Envoy
Gateway**

\$0 Open source · no license



The question has changed.

API lifecycle management has become a critical capability for modern enterprises. As organizations scale, they must enable developers to discover, publish, secure, consume, and operate APIs efficiently – while maintaining governance and operational control.

Traditionally, that meant large API Management suites such as **MuleSoft** and **Kong**, with their extensive feature catalogs and integrated experiences. But the rise of Kubernetes, platform engineering, GitOps, and AI-assisted development is changing how organizations evaluate API platforms.

THE QUESTION IS NO LONGER

~~Which platform has the most features?~~

THE QUESTION THAT MATTERS

Which platform delivers the capabilities we actually use – while maximizing agility, minimizing cost, and reducing long-term vendor dependency?

Teams use a small slice of any API platform.

80%

of activity lives in 5 core capabilities

More than 80% of day-to-day API lifecycle activity falls into just five core categories. These are the foundation of any successful API program.

01

API Discovery & Ownership

Understanding what APIs exist, who owns them, and where they are documented.

✓ Faster reuse & accountability

02

API Publication & Exposure

Publishing APIs and securely exposing them to consumers.

✓ Accelerated API delivery

03

API Security & Governance

Authentication, authorization, policies, and compliance controls.

✓ Reduced security & compliance risk

04

API Consumption & Self-Service

Developers discover and consume APIs without ticket-driven processes.

✓ Improved developer productivity

05




API Observability & Operations

Monitoring API health, performance, traffic, reliability, and usage.

✓ Increased reliability & efficiency

All three check every core box.

Across the five capabilities that make up the bulk of daily work, MuleSoft, Kong, and Backstage + Envoy Gateway are at parity. The differences lie elsewhere.

CORE CAPABILITY	 MuleSoft	 Kong	 Backstage + Envoy Gateway
API Discovery & Ownership	✓	✓	✓
API Publication & Exposure	✓	✓	✓
API Security & Governance	✓	✓	✓
API Consumption & Self-Service	✓	✓	✓
API Observability & Operations	✓	✓	✓

KEY OBSERVATION

When every platform delivers the core, the discussion shifts from ~~feature availability~~ to **architecture, economics, flexibility, and long-term strategy.**

The features you're billed for — and rarely touch.

Enterprise suites advertise hundreds of additional capabilities. They can be valuable in specific industries — but most organizations use only a small fraction of them after implementation, while continuing to pay for all of it.

✓ Used every day

The capabilities that carry the program

- API Catalog
- API Gateway
- Security Policies
- Documentation
- Developer Self-Service
- Observability

✗ Largely unused

Paid for, seldom adopted

- API monetization
- API marketplaces
- Revenue management
- Subscription billing
- Partner ecosystems
- Advanced workflow engines
- Specialized governance modules
- Vendor-specific analytics
- Legacy integration tooling

Enterprises frequently pay for capabilities that provide little measurable business value — capabilities that keep contributing to licensing, support, and operational cost long after go-live.

Platform engineering and GenAI changed the math.

The old logic was simple: buy the big suite, because building capabilities yourself is expensive. Two shifts have dismantled that assumption.



The Platform Engineering Model

Instead of buying every conceivable feature, teams now prefer to:

- ✓ Implement all core capabilities immediately
- ✓ Extend only where business value exists
- ✓ Leverage open standards & cloud-native technologies
- ✓ Avoid long-term dependency on proprietary ecosystems



The GenAI Effect

Customization used to be expensive. With AI-assisted development:

- ✓ New workflows developed faster
- ✓ Governance processes customized quickly
- ✓ Developer experiences tailored efficiently
- ✓ Integration requirements addressed rapidly

~~months~~



days

Capabilities that once required months of engineering can now be delivered in days or weeks. Organizations no longer need to buy every potential feature up front.

Two comprehensive commercial suites.



MuleSoft

Anypoint Platform

Integrated API & Integration Suite

STRENGTHS

- ✓ Comprehensive feature catalog
- ✓ Large integration ecosystem
- ✓ Unified vendor experience
- ✓ Mature enterprise support model

CONSIDERATIONS

- Premium licensing model
- Proprietary ecosystem
- Higher long-term vendor dependency

BEST FOR

Organizations prioritizing integration capabilities and a single-vendor approach.



Kong

Konnect / Gateway

Integrated API Management Platform

STRENGTHS

- ✓ Mature API gateway
- ✓ Strong runtime security
- ✓ Developer portal capabilities
- ✓ Rich API management features

CONSIDERATIONS

- Commercial licensing model
- Additional platform tooling often required
- Focused on API management, not broader platform engineering

BEST FOR

Organizations seeking a commercial API management platform with strong gateway capabilities.

Both are capable, mature platforms. Both also carry commercial licensing, a degree of proprietary lock-in, and a feature surface far larger than most teams operate.

One composable, cloud-native foundation.



Backstage + Envoy Gateway

Cloud-Native API Lifecycle Management

\$0

open-source license

STRENGTHS

- ✓ Kubernetes-native architecture
- ✓ Self-service developer experience
- ✓ High extensibility
- ✓ GitOps-native operations
- ✓ Open ecosystem
- ✓ Reduced vendor lock-in

CONSIDERATIONS

- Requires platform-engineering ownership
- Composable architecture rather than a single packaged suite

BEST FOR

Organizations embracing Kubernetes, cloud-native architectures, platform engineering, and developer self-service.

The trade-off is honest: composability asks for platform-engineering ownership. For an organization already running Kubernetes at scale, that ownership already exists – and it converts into control, not cost.

Measure value realized, not features shipped.

THE MOST IMPORTANT FINOPS QUESTION

How much of the platform are we actually using?



Evaluate platforms on

- ✓ Cost per active capability
- ✓ Developer productivity gains
- ✓ Operational efficiency improvements
- ✓ Time-to-market acceleration
- ✓ Vendor dependency risk

Rather than

- ✗ ~~Total number of available features~~

A larger catalog is not a larger return. Value is what teams actually put to work.

For organizations already running Kubernetes at scale

A cloud-native approach reuses infrastructure and expertise you've already paid for:

✓ Reduced platform licensing costs

✓ Improved utilization of existing infrastructure

✓ Faster delivery of custom workflows

✓ Lower vendor dependency

✓ Greater flexibility for future requirements

Cloud-native approaches can significantly reduce platform costs versus traditional enterprise suites — particularly where Kubernetes and platform-engineering expertise already exist. Actual savings vary with deployment scale, licensing, support requirements, and organizational maturity.

Choose deliberately.

Each platform has a profile where it is the right answer. The question is which profile matches your operating model.



Choose MuleSoft when

- Integration is a primary business requirement
- A comprehensive commercial suite is preferred
- Vendor consolidation is a strategic objective



Choose Kong when

- API management is the primary focus
- A mature commercial API platform is desired
- Out-of-the-box API management features are prioritized



Choose Backstage + Envoy Gateway when

- ✓ Platform engineering is a strategic initiative
- ✓ GitOps is part of the operating model
- ✓ Architectural flexibility is valued
- ✓ Kubernetes is the standard operating platform
- ✓ Developer self-service is a priority
- ✓ You want to avoid paying for rarely used features

What to walk away with.

1 All three deliver core API lifecycle management

Discovery, publication, security, self-service, and observability are available across MuleSoft, Kong, and Backstage + Envoy Gateway.

2 The real decision is architectural

Not feature parity – but platform strategy, operating model, total cost of ownership, developer experience, and long-term flexibility.

3 The future is composable

Modern enterprises favor strong core capabilities, open standards, extensibility, and cloud-native operations over large suites of rarely used features.



OUR RECOMMENDATION

For a Kubernetes-first, platform-engineering organization, **Backstage + Envoy Gateway** delivers every core capability your teams use daily – with open standards, full extensibility, and no premium license for features that go untouched.

FINAL THOUGHT

The most successful API programs aren't built on the largest feature catalog. They're built on platforms that deliver the capabilities developers use every day – while staying flexible enough to evolve with the business. The question is no longer which platform has the most features. It's which platform delivers what we need today and the freedom to innovate tomorrow.