How Platform Engineering Is Quietly Killing the Monolith

The API-First Shift

by Derek Ashmore

This Whitepaper examines how platform engineering and an API-first mindset are enabling enterprises to replace legacy applications with modern, resilient services, using secure APIs, CI/CD, observability and internal developer platforms to reduce risk, minimize disruption and turn modernization into a continuous driver of innovation and business agility.

For decades, enterprises have struggled with the burden of legacy systems, critical to daily operations yet brittle, outdated, and costly to maintain. The traditional approach to modernization was the dreaded "big-bang rewrite," a risky endeavor that often ran over budget, missed deadlines, and disrupted the business.

Today, a quieter revolution is underway. Platform engineering and the API-first mindset are enabling organizations to modernize gradually, safely, and at scale. Instead of tearing out the monolith in one dramatic motion, enterprises are strangling it piece by piece, wrapping legacy systems in APIs and incrementally replacing them with modern services.

This approach not only reduces risk but also accelerates innovation. APIs become the connective tissue, while platform engineering provides the tools and guardrails that make the transition reliable. The result: modernization that is continuous, business-aligned, and resilient.

Platform Engineering as the Backbone of Modernization

At the heart of this shift is platform engineering, which provides the infrastructure and tooling to make incremental change safe, fast, and repeatable. Teams use secure API gateways to expose legacy functionality behind stable, decoupled interfaces, ensuring business continuity while preparing for gradual replacement. CI/CD pipelines automate testing and delivery, enabling frequent deployments of newly extracted services without destabilizing the environment.

Observability is embedded across both legacy and modern components, giving teams the ability to monitor performance and diagnose issues in distributed systems. Meanwhile, infrastructure-as-code ensures that environments are consistent, version-controlled, and reproducible across the entire stack.

Together, these capabilities make the Strangler Fig Pattern, the progressive replacement of legacy systems, not only viable but scalable. Rather than betting the business on one high-stakes rewrite, platform engineering transforms modernization into an ongoing process that reduces disruption and maintains uptime.

Why Secure, Well-Governed APIs Are the Game Changer

APIs sit at the center of this transformation. Secure, well-governed APIs create clear service boundaries, enforcing explicit contracts between services. This helps define ownership, reduce hidden dependencies, and modularize functionality during the breakup of a monolith.

Governed APIs also embed security and compliance from the start. Gateways enforce authentication, rate limiting, data policies, and auditability, ensuring that modernization is both safe and compliant. Once legacy functionality is exposed through APIs, teams across the enterprise can reuse services instead of duplicating effort, accelerating delivery and enabling consistency.

APIs also simplify testing and observability. Services can be monitored in isolation, reducing the risk of cascading failures and improving confidence in incremental releases. In short, APIs turn messy legacy environments into structured, modular systems that can evolve safely over time.

Minimizing Disruption with CI/CD and Service Discovery

One of the greatest fears of modernization is disruption to critical business systems. Platform teams address this by combining CI/CD pipelines with service discovery to create a resilient foundation for change.

CI/CD pipelines, bundled with automated testing, validate changes early and often. This reduces risk by catching errors before they reach production and builds trust with the business that new functionality will not break existing systems. Service discovery then decouples service location from implementation, dynamically routing traffic between legacy and modern components. As workloads shift from monolith to microservices, or from on-prem to cloud, service discovery ensures smooth transitions without requiring changes from upstream consumers.

Together, CI/CD and service discovery enable the Strangler Fig Pattern in practice, allowing old and new systems to operate side by side with minimal disruption.

Observability: The Safety Net for API-First Transitions

As enterprises embrace API-first modernization, architectures become more distributed and failure modes more complex. Observability is the safety net that ensures reliability throughout the transition.

TRACK

API Platforms & Business

The Power of API Integration and AI Innovation

Discover how APIs drive business growth through seamless integration and AI-powered strategies, transforming operations and fostering innovation. Learn how APIs modernize systems, enable agility, and create new revenue streams, driving successful business transformations.

Learn from Industry Leaders about:

- **Business Potential with APIs:** Unlock the untapped business potential with APIs by seamlessly integrating them into your products and platforms. Delve into AI-powered API management strategies and future trends to drive innovation, efficiency, and business growth. Understand how APIs can create new revenue streams, enhance customer engagement, and streamline business operations.
- **APIs for Digital Transformation:** Understand how APIs can drive digital transformation within your organization. Learn about the role of APIs in modernizing legacy systems, enabling new digital services, and supporting agile business practices. Explore case studies of companies that have successfully used APIs to transform their business models.
- **Monetizing APIs:** Discover strategies for monetizing your APIs effectively. Learn about different API monetization models, including subscription-based, pay-per-use, and freemium models. Understand how to create value propositions that attract developers and business partners, and explore case studies of successful API monetization.
- **API Marketplaces and Ecosystems:** Explore the role of API marketplaces in expanding your business reach and fostering innovation. Understand how to participate in or create an API marketplace to expose your APIs to a broader audience. Discuss the benefits of building and participating in API ecosystems to drive business growth.
- **Using API Platforms Effectively:** Discover best practices for using API platforms to maximize their potential. Learn how to efficiently onboard developers, utilize platform features for monitoring and analytics, and leverage built-in tools for security and compliance. Explore case studies showcasing successful API platform utilization, and understand how to integrate these platforms into your business processes to enhance productivity and drive innovation.
- **Leveraging AI and LLMs:** Explore integrating AI and large language models (LLMs) into APIs to enhance business applications. Build AI-powered applications and AI-assisted organizational bots using tools like LangChain, LlamaIndex, and vector databases like Pinecone or Chroma. Discuss the business implications and opportunities of leveraging AI through APIs.

Monitoring API performance and usage provides assurance that new services behave as expected and meet SLAs. Tracing requests across old and new components allows teams to diagnose issues that span both environments, a critical capability during partial migrations. Observability also enables proactive detection of anomalies, catching error spikes or latency increases before they impact end users.

Most importantly, observability provides data-driven feedback loops. Teams use metrics and logs to refine CI/CD pipelines, adjust rollback strategies, and optimize migration pacing. With observability in place, modernization becomes not only possible but measurable, traceable, and reversible when needed.

Internal Developer Platforms: Enforcing Consistency at Scale

Even with APIs as the foundation, consistency at scale can quickly become a challenge. This is where Internal Developer Platforms (IDPs) come in. Designed to reduce cognitive load and accelerate developer productivity, IDPs embed governance and automation directly into the developer workflow.

Through scaffolding and templates, IDPs ensure that new APIs are generated with consistent authentication, versioning, and documentation patterns. Automated API linting and contract checks in CI/CD enforce compliance with naming conventions and OpenAPI standards before release. IDPs also standardize service registration and gateway integration, ensuring that all services are discoverable, observable, and governed.

Golden paths guide developers through approved workflows, minimizing misconfiguration and boilerplate while accelerating delivery. Importantly, IDPs act as enablement layers rather than control points. Security enforcement still resides in systems like API gateways and service meshes, while IDPs streamline and scale the developer experience.

Looking ahead, the role of IDPs may evolve. As AI-assisted coding becomes mainstream, many of the consistency checks and golden paths that IDPs provide may be handled directly by AI agents. For now, however, IDPs remain critical for managing complexity at scale.

The Strategic Imperative of API-First Modernization

The API-first shift is not just a technical evolution, it is a strategic imperative. Enterprises that cling to monoliths risk being outpaced by competitors who can innovate, scale, and adapt faster. By embracing platform engineering, CIOs and technology leaders can modernize without gambling on risky rewrites.

Secure APIs create modularity and reusability. CI/CD and service discovery minimize disruption. Observability ensures reliability. IDPs enforce consistency at scale. Together, these capabilities turn modernization from a one-time event into a continuous capability.

The API-first shift is ultimately about business resilience. It allows enterprises to adapt at the speed of change, delivering new functionality quickly while maintaining the reliability of core systems. Instead of being held hostage by their legacy systems, organizations gain the freedom to evolve.

From Monoliths to Modern Platforms

The death of the monolith will not happen overnight. But with platform engineering and the API-first approach, it no longer needs to. The Strangler Fig Pattern offers a safer path forward, one that prioritizes uptime, reduces disruption, and delivers incremental value along the way.

For CIOs, the opportunity is clear. API-first modernization is not just about surviving legacy debt; it’s about creating a platform for innovation that can scale with the future. Those who embrace it will turn their IT foundations into engines of agility. Those who delay will find themselves trapped in systems too slow, too brittle, and too costly to compete.

The monolith’s days are numbered. The only question is whether enterprises will strangle it on their own terms or be strangled by it.

Derek Ashmore is an Application Transformation Principal at Asperitas. He helps companies use cloud technologies more cost-effectively, securely, and with better availability and performance, to gain the advantage over their competitors.

API CONFERENCE

THE CONFERENCE FOR **WEB APIS, API DESIGN & MANAGEMENT**

November 23 – 27, 2026

BERLIN

May 11 – 15, 2026

LONDON

NEW YORK

September 28 – October 02, 2026