SPOTLIGHT

Sponsored by: BrainGu

Investing in a secure, scalable, and fully supported IDP promotes faster releases, tighter compliance, and increased developer satisfaction, avoiding the hidden costs of building an in-house solution.

Enhancing Developer Experience with a Resilient, Scalable, Reliable, and Secure Internal Developer Platform

March 2025

Written by: Katie Norton, Research Manager, DevSecOps and Software Supply Chain Security, and Matthew Flug, Research Manager, Cloud Application Deployment Platforms

Introduction

Software powers business growth, enhances customer experience, and provides a competitive edge; however, delivering the right software at the right time remains a significant challenge. Organizations must navigate a fragmented landscape of development tools, evolving security and compliance requirements, and the constant pressure to innovate while maintaining reliability and performance at scale.

At the same time, modern development models, such as DevOps and DevSecOps, have shifted more responsibilities to developers, increasing their cognitive load. IDC's *Developer View Worldwide Survey* found that in 2023, developers spent 60% of their time on operational responsibilities rather than writing code, up from 51% in 2022.

AT A GLANCE

KEY TAKEAWAYS

- IDPs have become essential for delivering secure, high-quality software at speed.
- Building an IDP requires heavy investment, with significant ongoing maintenance and talent needs.
- Buying an IDP offloads complexity, accelerates time to market, ensures built-in compliance, and frees teams to focus on core innovation.

Striking the right balance between speed and robust security, resilience, and compliance has become a formidable challenge. Organizations must simplify complexity, streamline workflows, and empower teams with enhanced visibility and automation to consistently deliver secure, high-quality applications at business pace. To address this challenge, platform engineering has emerged as a key strategy in modern software development.

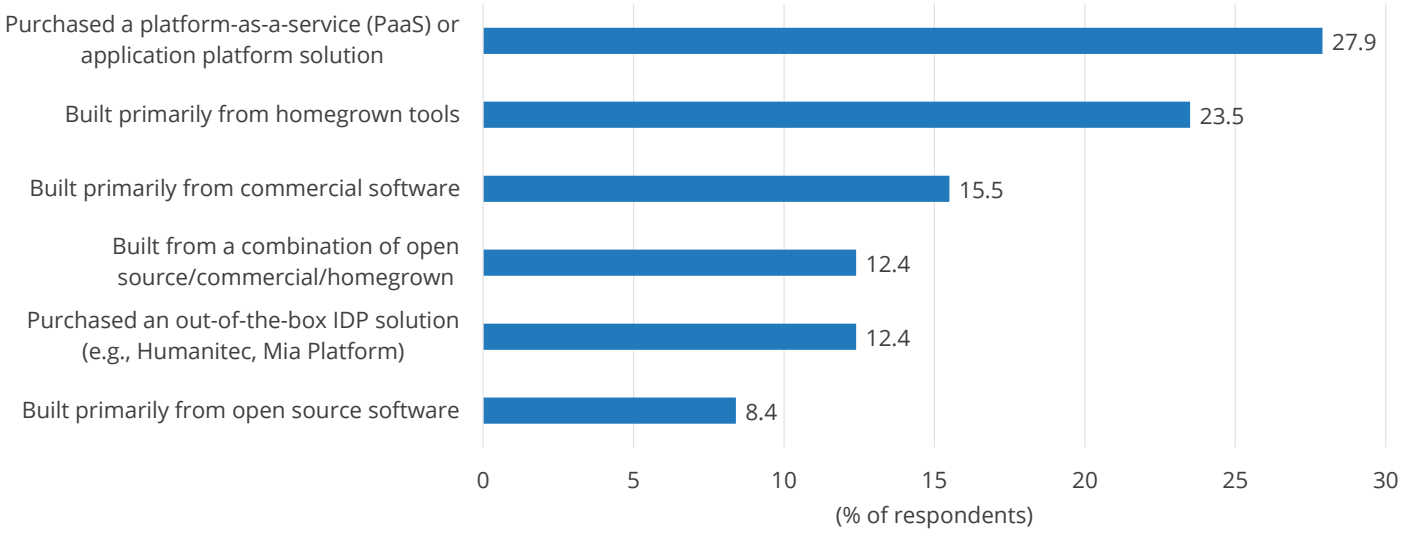
IDC defines platform engineering as designing, building, and maintaining a platform of curated tools, services, and knowledge, namely an internal developer platform (IDP), which enables development teams' self-service access to the resources necessary to build, test, and operate digital solutions. In IDC's 2024 *DevOps Practices, Perceptions, and Tooling Survey,* 80.8% of respondents indicated they're expanding the use of, using, or piloting an IDP. More organizations are using IDPs because they reduce friction in the development process by providing a clear, secure, and compliant path that removes complexity and reduces superfluous work.

As IDPs become fundamental, organizations must confront the timeless technology question: Build or buy? IDC's 2024 *DevOps Practices, Perceptions, and Tooling Survey* found that when it comes to IDPs, purchasing a platform-as-a-service (PaaS)/application platform solution was the top approach (27.9%) as seen in Figure 1.

While building a custom platform might seem attractive for teams wanting full control, it can quickly become a resource-intensive endeavor requiring deep expertise, ongoing maintenance, and significant opportunity costs. In contrast, purchasing a proven, ready-to-deploy IDP allows organizations to leverage the provider's specialized knowledge and established best practices, reducing initial setup time and long-term overheads. Moreover, a commercial IDP typically comes with dedicated support, regular feature updates, and compliance safeguards — factors that help ensure stability, security, and future scalability. By opting to buy, companies can redirect critical engineering resources to tasks that directly contribute to business differentiation, accelerating innovation and improving time to market.

FIGURE 1: ***Buying an Application Platform Is the Top Approach for IDPs***

Q *What approach are you taking to build your IDP?*



n = 251

Source: IDC's DevOps Practices, Perceptions, and Tooling Survey, February 2024

Core Components of an IDP

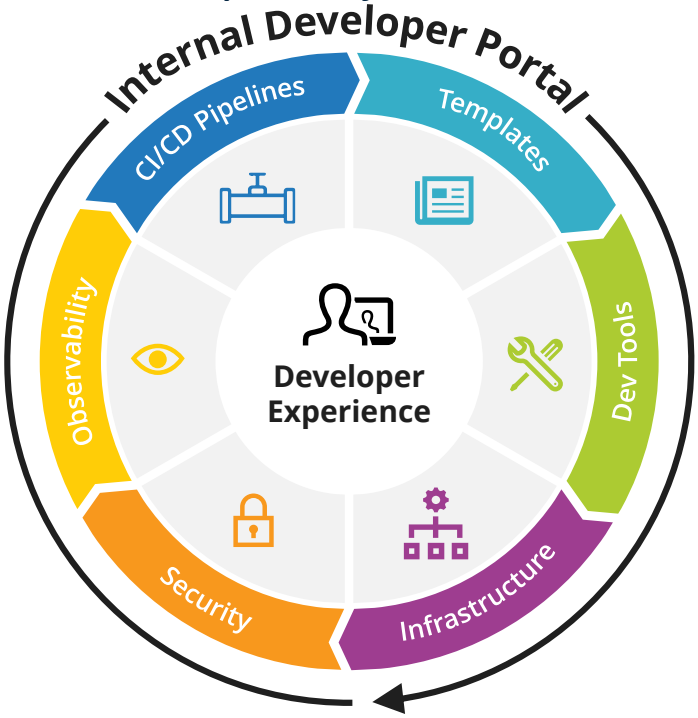
To effectively evaluate a product for its suitability as an IDP, organizations must first understand the key features of effective IDPs, including:

- **Comprehensive developer tooling:** IDPs should streamline everyday tasks and integrate with key tools such as linters, debuggers, source code repositories, and container registries.
- **Ease of use:** IDPs should offer a user-friendly interface, personalized views for various roles, and consistent "golden paths" across public cloud, private cloud, on-premises, and edge environments.

- **Standardized templates:** Predefined configurations help maintain consistency, reduce drift, and automate infrastructure setup across environments.
- **Infrastructure orchestration:** IDPs should automate deploying and managing resources (e.g., databases, storage, and networking) using infrastructure as a code and GitOps best practices.
- **Security and compliance guardrails:** IDPs should enforce security standards, auto-generate compliance artifacts (e.g., SBOMs and bodies of evidence [BoEs]), and provide built-in security policies and verifications.
- **Observability:** IDPs should centralize telemetry, integrate monitoring and logging, and provide real-time insights to quickly detect and resolve issues.
- **DevOps pipelines and workflows:** IDPs should deliver "golden paths" that automate continuous integration (CI)/CD, embedding security and compliance to streamline builds, tests, and deployments.

Figure 2 illustrates the core components of an IDP.

FIGURE 2: ***Components of an IDP***



Source: IDC, 2025

The Challenges of Building an IDP

Building an internal developer platform demands a substantial investment for initial development and ongoing maintenance — it is not a "set it and forget it" process. As complexity increases, so does the need for dedicated teams to address bugs, apply security patches, and maintain performance. Tailoring the platform to specific needs often involves

custom code that becomes harder to update as technologies evolve, and scaling a homegrown IDP may require repeated reengineering efforts.

Organizations must treat an IDP like a product, where developers act as customers that rely on the platform's capabilities. Evolving this platform in step with shifting technology and user preferences demands a steady stream of resources that could otherwise fuel core innovation, potentially affecting competitive advantage. In addition, the timeline for delivering a custom IDP can be extensive — developers may have to wait for critical features and updates, slowing the organization's ability to release software quickly.

Finding and retaining platform engineers with the necessary skills to build and manage an IDP adds another challenge. *DevOps Practices, Perceptions, and Tooling Survey, 2024: Platform Engineering* (IDC #US51622924, February 2024) indicates that engineering teams typically consist of 9–10 engineers, making it difficult to assemble a group with the right expertise. Holding on to these experts and the specialized institutional knowledge they develop can prove even more challenging, as they are often in high demand industrywide.

Integration further complicates matters, as a custom IDP must work seamlessly with various systems — CI/CD pipelines, version control, and observability tools — many of which are open source. Although open source solutions enable flexibility and community-driven innovation, integrating disparate projects can be difficult. This challenge is evident in the limited number of organizations (8.4%) that build IDPs entirely on open source DevOps tools.

Security remains a paramount concern. Because the internal team takes full responsibility for identifying, developing, and deploying security patches, they must remain vigilant against potential vulnerabilities that might arise in any custom elements of the platform. Data protection requirements call for robust encryption, access controls, and continuous monitoring, which can be complex to implement effectively. Meanwhile, comprehensive identity and access management must be in place to ensure that only authorized personnel have access to the platform's sensitive aspects. Without vendor support or external security updates, organizations building an IDP in-house shoulder the entire burden of keeping the platform secure throughout its life cycle.

Benefits of Buying an IDP

A Common Developer Experience

Purchasing a platform frees developers to focus on the organization's core innovations rather than platform building and maintenance. IDC's 2023 *PaaS Decision-Maker and Business Value Survey* found that over one-third of organizations with application platforms cite developer empowerment for differentiated work as their primary source of ROI. An out-of-the-box platform also accelerates the adoption of platform engineering practices through its quick implementation and availability for use. Furthermore, IDPs fail when they become outdated and no longer help users complete daily tasks. By regularly delivering new features and capabilities, vendors ensure their platforms keep pace with evolving technology and developer trends, sustaining long-term value.

Resiliency and Scalability

Vendor-provided platforms undergo extensive testing in diverse environments, delivering reliability and performance that can be difficult to match in a custom-built solution. The vendor largely handles maintenance, reducing the skill sets and number of platform engineers organizations need in-house and ensuring that dedicated support is available for issues beyond internal expertise. These commoditized platforms also enable scalability, allowing organizations to scale

horizontally or vertically without costly re-architecting. As the platform expands, vendors can pass on cost savings from economies of scale, making these solutions more cost-effective than custom-built platforms for large workloads.

Reliability and Security

IDP reliability stems from rigorous testing, industry-standard practices, and ongoing vendor support typical of prebuilt solutions, minimizing disruptions and operational overheads. Real-time monitoring, observability tools, and proactive updates help maintain stable operations and enable swift issue resolution. Many vendor offerings also simplify compliance by meeting common industry regulations and frequently holding security certifications that can be costly and time-consuming to obtain with a custom-built platform. By providing timely bug fixes, security patches, and dedicated support, these solutions help reduce vulnerabilities and sustain a robust security posture.

Additional Benefits

- **Cost efficiencies:** Building an IDP from scratch requires significant up-front costs for development, infrastructure, and talent acquisition.
- **Faster time to market:** Purchased platforms are ready to deploy, significantly reducing the time necessary to start leveraging their capabilities.
- **Vendor expertise:** Vendors bring deep expertise in DevSecOps, platform design, and scalability, which they embed into their products via continuous integration centered around customer centricity.
- **Ecosystem and integrations:** Vendors offer prebuilt integrations for popular tools (e.g., Kubernetes, CI/CD systems, and monitoring tools) and support plug-ins and add-ins, reducing the complexity of setting up functional environments.
- **Risk mitigation:** Vendors share the responsibility for maintaining uptime, security, and compliance and offer SLAs guaranteeing performance, offering a safety net that custom-built solutions lack.

Considering BrainGu

BrainGu began operations in 2012, with its headquarters in Grand Rapids, Michigan. BrainGu's IDP solution, SmoothGlue, offers organizations the capabilities to build, deploy, run, and monitor cloud-native, on-premises, hybrid, and edge applications. Focused on secure DevSecOps practices and compliance, SmoothGlue is built to exceed the requirements of mission-critical enterprises and integrates dozens of best-of-breed components for a cohesive experience, securely serving teams of all sizes.

SmoothGlue is designed to integrate seamlessly with a wide range of existing tools and systems, allowing organizations to leverage their current technology stacks while enhancing them with platform engineering and management capabilities from a single, centralized location. It includes essential features for application developers and managers, such as code version control, CI/CD pipelines, agile software life-cycle management, SBOMs and cybersecurity artifacts, production deployments, and product management integration.

The SmoothGlue platform is built on well-known and trusted open source projects, many supported by the Cloud Native Computing Foundation, such as Kubernetes, Crossplane, Big Bang, Istio, Grafana, Argo, and Tekton. The platform leverages popular collaboration tools, such as Confluence, GitLab, and Jira, and includes SmoothGlue Console, a

developer portal that provides a single point of entry Keycloak into SDLC tools. It automates the creation and syncing of teams, users, and projects; streamlines role creation, ensuring adherence to the principle of the least privilege principle; and provides a centralized SSO solution based on Keycloak.

SmoothGlue's design enables rapid deployment, allowing organizations to quickly onboard developers. The platform comes with preconfigured workflows and templates that streamline the development and deployment processes, permitting faster delivery of applications and services. It is customizable to enable organizations to tailor the platform to their specific needs. BrainGu provides support for SmoothGlue users, helping them overcome skill gaps and accelerate platform adoption.

BrainGu prioritizes security. It adheres to the highest U.S. Department of Defense standards to ensure that control is not sacrificed while accelerating deployments and has received ATOs on IL4, IL6, and Top Secret. The platform auto-generates SBOMs, enabling customers to quickly produce an information assurance and security body of evidence. It dynamically generates CI/CD pipeline artifacts and stores them within the software BoE for third-party inspection and validation.

Challenges

BrainGu's biggest challenge is fending off internal platform engineering efforts. Organizations may invest in building a platform themselves, instead of investing in an IDP that is built to meet the general needs of many organizations when:

- Technical or business needs are highly specific or unique and not satisfied by commercial technologies
- Off-the-shelf platforms do not meet their technical, compliance, security, or cost requirements
- Resources and talent acquisition are not an issue

Moreover, when organizations do choose to purchase an IDP, they may look to PaaS solutions from hyperscalers to meet their platform needs. These vendors boast the ability to reduce tool sprawl with trusted offerings across the SDLC that provide consolidation benefits. However, these tools are often expensive, limiting the ability to customize the platform to meet an organization's particular needs.

Conclusion

Organizations should carefully assess the long-term resource, security, and innovation implications of building an IDP in-house versus purchasing a proven solution. While custom platforms may promise control, they also demand sustained engineering investments and continuous maintenance to remain effective. In contrast, buying an IDP provides immediate access to vendor expertise, dedicated support, and built-in best practices for security, compliance, and scalability — freeing teams to focus on core innovations and expediting the delivery of business-critical applications. Investing in a vendor-provided IDP accelerates time to market, streamlines operations, and strengthens an organization's competitive edge by enabling developers to channel their efforts into driving differentiated value.

About the Analysts

Katie Norton, *Research Manager, DevSecOps and Software Supply Chain Security*

Katie's core research areas include how security is integrated into the software development life cycle, exploring how development teams take ownership of security and collaborate with AppSec teams, and examining the drivers of DevSecOps adoption. She also explores buying patterns and trends for DevSecOps and software supply chain security tooling.

Matthew Flug, *Research Manager, Cloud Application Deployment Platforms*

Matt's research focuses on software vendors, cloud providers, and end customers' software development/deployment plans and experiences. He is responsible for insights and analysis on emerging platform-as-a-service (PaaS) solutions that support developers deploying highly performant, modern, and cloud-native applications.

MESSAGE FROM THE SPONSOR

BrainGu believes that developing and delivering software in highly regulated spaces shouldn't be hard or slow; the right way to ship code to production should be the easy way. Our internal developer platform, SmoothGlue, is a complete software development, delivery, and management platform that integrates best-of-breed tools and exceeds the usability and cybersecurity requirements of mission-critical enterprises. Our platform provides a cohesive experience, ensuring secure, compliant, and reliable software delivery from development to production and beyond. SmoothGlue supports self-service deployment of secure, cloud-native applications across public and private clouds, on-premises, and edge environments. It has built-in guardrails for compliance, security, and reliability, allowing developers to quickly build and deploy applications while meeting the highest compliance standards. Start building today and experience seamless, secure software delivery with SmoothGlue.

IDC Custom Solutions

The content in this paper was adapted from existing IDC research published on www.idc.com.

IDC Research, Inc.
140 Kendrick Street
Building B
Needham, MA 02494, USA
T 508.872.8200
F 508.935.4015
blogs.idc.com
www.idc.com