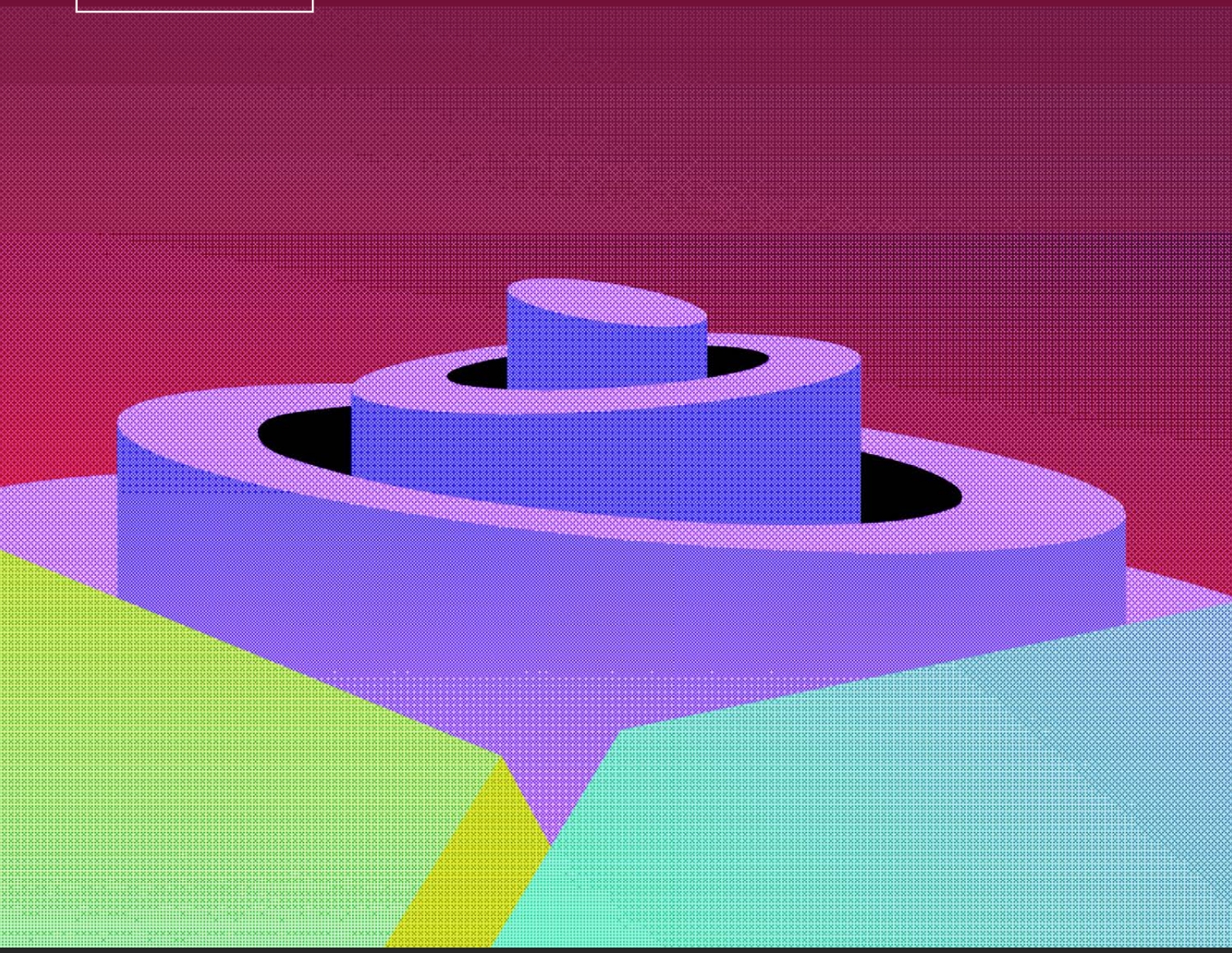
Weave Intelligence

State *of* Platform Engineering Report

VOLUME 4



WEAVE INTELLIGENCE PRESENTS
STATE OF PLATFORM ENGINEERING REPORT VOLUME 4
PUBLISHED IN 2025

State of Platform Engineering Report Volume 4

Table of contents

Our sponsor

Broadcom's VMware Cloud Foundation (VCF) is the platform for the modern private cloud, delivering a single unified platform for VMs and containers that supports all applications, traditional, modern, or AI, with a consistent operating model, governance, and controls spanning data centers, edge, and managed cloud infrastructure. VCF combines the agility and scalability of public cloud with the security, performance, architectural control and total cost of ownership (TCO) benefits of an on-premises environment. VCF frees development teams to focus on applications instead of infrastructure. Through the native VMware vSphere Kubernetes Service (VKS), an integrated CNCF-certified Kubernetes runtime, platform engineering teams can support agile modern app development directly from the private cloud. GitOps-driven, self-service infrastructure with guardrails balances developer autonomy with IT control and reduced risk, while multi-tenant, policy-driven delivery and built-in observability enable more secure and visible applications that are always in their desired state.

Executive *summary*

Platform engineering is eating the world. As the defining discipline of our time, platform engineering has rapidly evolved into the foundational operating system of the modern enterprise. Rather than a pure focus on infrastructure, or DevEx, modern platform teams are absorbing traditional silos like observability, security, data, and FinOps directly into their centralized paradigm. Thus the discipline has begun to move the industry completely from “shifting left” to actively “shifting down” by embedding essential capabilities like security, guardrails, and quality attributes directly into the platform. This approach aims to remove toil completely, rather than merely dumping it onto developers earlier in the software delivery lifecycle.

Platform engineering has begun to replace the manual, bespoke work of the "artisan software engineer," with a standardized model more reminiscent of an industrialized supply chain. No longer is an organization's success driven by the individual prowess of its "elite coders" but by the effectiveness of its software supply chain.

30% of platform teams still report that they do not measure success at all

This rapid institutionalization however exposes a set of critical tensions. While the industry is moving decisively away from the early "portal trap" toward focusing on robust core platform logic, measurement practices remain alarmingly weak. Despite significant improvement from volume 3, nearly 30 percent of platform teams still report that they do not measure success at all, crippling their ability to prove ROI or secure the investment needed for long-term growth. At the same time, platform adoption is still frequently driven by extrinsic push or mandate (36.6%) rather than the intrinsic value necessary for true user pull (28.2%).

Above all else however, the defining trend of this era is AI. The vast convergence of infrastructure and software has pushed us from the cloud-native era into the AI-native era, placing platform engineering at the absolute core of this transformation.

Platform teams face a dual mandate: they must both deliver AI-powered platforms (augmenting Internal Developer Platforms with tools to turbocharge developer productivity and automation) and construct robust Platforms for AI (building specialized ecosystems to support the deployment, training, and scaling of complex AI/ML workloads for data scientists). An overwhelming 94% of organizations see AI as critical to the future of platform engineering, validating the new dual role of the platform engineer as a driver, not merely a consumer, of AI.

The responsibility of the platform engineer has never been greater. We are the architects of the new digital factory, tasked with transforming fragmented experimentation into systematic, governed, and repeatable value. As we look toward the future, those organizations that embed FinOps into every workflow, treat observability as a fundamental default, and master the creation of golden paths for compliance, reliability, and AI, will define the next decade.

This challenge is not purely technical, it is fundamentally cultural. It requires dedication to continuous learning, strong product leadership, and unwavering commitment.

It requires experimentation, a willingness to take risks, and an acceptance of learning from failure. The future belongs to those who invest now in people, culture, and AI-native foundations; those who hesitate will be left behind, and inherit an insurmountable load of organizational debt. Those who embrace this new era of software engineering will ride a wave of innovation, the likes of which comes only once in a generation. The choice is yours to make.

> If not for AI, platform engineering would be the number 1 trend in the enterprise.

Rickey Zachary

GLOBAL PLATFORM ENGINEERING LEAD, THOUGHTWORKS

Platform engineering is eating the world

As Platform engineering has matured over the last few years, it has evolved from an emerging methodology nipping at the heels of DevOps and focused merely on DevEx and infrastructure to a foundation of modern software delivery. Much like software fundamentally reshaped nearly every industry, platform engineering is now transforming software development itself, streamlining operations, and redefining traditional IT roles at an unprecedented pace. This transformation is especially driven by the necessity for speed, scale, and security in the age of AI.

The shift is so profound that senior executives are acknowledging that platform engineering is "consuming all other functions", absorbing traditional roles such as observability, security, data, and more directly into the platform paradigm. Organisations that embrace this shift, moving away from artisanal workshops to structured digital factories, are achieving incredible results in improving DevEx, security, productive AI adoption, and more.

Shifting down

"Shifting down" is the platform engineering strategy that aims to maximize value and sustainability by embedding responsibilities, controls, and quality attributes directly into the platform rather than leaving them to developers. Rather than "shifting left", which simply moves the toil earlier in the software delivery lifecycle (often simply onto the developer), shifting down aims to remove the responsibility entirely by embedding it directly into the platform via automation or improved processes.

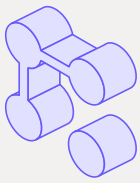
Shifting down eliminates manual toil and removes whole classes of errors by embedding security, reliability, and performance directly into automated platform capabilities. By building policy enforcement, secure defaults, and architectural guardrails into the platform, teams reduce developer burden and make work easier by default. This sociotechnical approach establishes a shared responsibility model that lets developers stay focused on product work while the platform absorbs the underlying complexity.

"Shifting down" has also driven a renewed focus on the foundational backend architecture of platforms rather than superficial interfaces. Early initiatives often fell into the "portal trap," chasing quick wins by building developer portals while neglecting the deeper orchestration and automation required for real impact. As the hype fades, the industry now recognizes that a platform is defined by its core logic - not its UI. Portals remain useful entry points, but they cannot replace the platform orchestrators, pipelines, and backend systems that deliver true value. This shift is evident not only in enterprise platform teams but also among portal vendors, who are increasingly expanding their backend capabilities to meet these more mature expectations.

New domains: Expanding beyond classical DevEx

Platform engineering's scope has expanded far beyond classical Developer Experience (DevEx) and container management. Platform teams are now becoming foundational enablers across critical new domains like AI, data, observability, security, and FinOps. Just as the ill-defined role of "DevOps" gradually grew to the absurdly defined "DevSecOps", and began to incorporate Observability and FinOps, so too is the obvious conclusion that platform engineering should engage with these domains as well.

As platform teams standardize workflows, automation, and governance, it becomes natural to extend these capabilities to adjacent domains where fragmentation and manual processes remain a bottleneck - especially those organizations suffering under the challenges of a poorly implemented policy of "shifting left".



AI, Data, and LLMOps

The intersection of Artificial Intelligence (AI) and platform engineering is the most significant new domain. Platform teams operate under a dual mandate regarding AI. Augmenting Internal Developer Platforms with AI and building new platforms that support and drive AI itself. This rapid emergence of AI is pushing enterprise architecture from the cloud-native era into the AI-native era. AI-native infrastructure demands a sophisticated, globally distributed foundation tailored for GPU-accelerated workloads, composable architecture, and advanced MLOps governance. Platform teams are already embracing this, with 75% of respondents to the State of AI in Platform Engineering report suggesting they are already hosting or preparing to host AI workloads.

Security platform engineering

Security is now a foundation, not a silo or a blocker with concepts like ("Security first" and "Secure by design") because they are key parts of platform engineering best practice. To integrate security directly into the platform, Security platform engineers drive:

Automated guardrails and policy enforcement

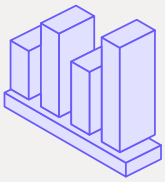
They enforce least-privilege access, encryption, and secure secret retrieval by default. Platform policy controls enforce these rules to manage risks, including AI agent hallucinations.

Secure golden paths

When secure practices are built into every golden path, developers can move fast without creating vulnerabilities, as the safest way naturally becomes the easiest and most productive.

Centralized secrets management and compliance

The platform manages secure secret storage and retrieval across the SDLC while the SPE automates security controls and compliance. Policy as Code can scan AI-generated code and enforce constraints automatically, providing scalable and transparent governance.



Observability platform engineering

The Observability Platform Engineer is responsible for turning system telemetry into a continuous operational feedback loop, with observability now designed as "observability by default," meaning it is built into the platform from the start. This begins with end-to-end telemetry: every component must emit metrics, logs, and traces by default, using standardized formats and centralized aggregation.

Thus, the most prominent reference architecture for an Internal Developer Platform in the industry received a large adoption to highlight these changes to the Observability Plane. This new domain of platform engineering spans monitoring, logging, tracing, and alerting capabilities and also incorporates SLO frameworks and reliability workflows, where Service Level Objectives are defined

per product and capability, with automated alerts tied to error budgets to support rapid root cause analysis and proactive incident detection. As the Observability Plane touches every other plane, it integrates closely with orchestrators and is often enhanced by AI-driven capabilities that automate log analysis and anomaly detection, surfacing issues before they impact production.

Together, the data collected through the Observability Plane ensures continuous, comprehensive visibility into system health, performance, and user impact at all times.

FinOps and platform engineering

The intersection of FinOps and platform engineering is one of the most recent strategically important areas for the industry as cloud usage accelerates, and platform teams grow larger and more encompassing. Platform teams are increasingly responsible for ensuring financial accountability at scale. FinOps is shifting from a detached reporting function into an embedded operating model within platforms themselves, where cost visibility, allocation accuracy, real-time forecasting, and automated optimisation can be built directly into the developer experience. This pushes enterprises from simple cost control to a more proactive cost-aware architecture, where tagging, rightsizing, lifecycle policies, and intelligent guardrails are directly integrated.

FinOps-enabled platforms give engineering organisations the ability to balance speed with financial discipline, which removes the constraint of cloud cost management, and when done right, drives significant operational advantage.

New platform engineering roles

The increasing complexity and expansion into new domains necessitate specialization within platform teams. This is leading to the emergence of dedicated, professionalized platform engineering roles that consolidate traditional silos.

Head of Platform Engineering (HOPE)

A senior leader overseeing the entire platform engineering function, setting strategic direction, aligning with business goals, and coordinating teams. The role requires broad skills across architecture, software development, security, and operations.

Platform Product Manager (PPM)

Acts as the bridge between platform engineering and the organization, structuring work and prioritizing features. The PPM balances technical understanding with user needs to maximize platform value.

Infrastructure Platform Engineer (IPE)

Defines default resource configurations and maintains the core infrastructure like servers, networks, databases. Ensures the platform is scalable, reliable, and efficient.

DevEx Platform Engineer (DPE)

Improves developer experience by streamlining workflows and reducing friction. Builds tools, templates, and documentation to make the platform intuitive and efficient documentation to empower developers.

Security Platform Engineer (SPE)

Implements and governs security policies within the development pipeline, embedding guardrails and maintaining robust security controls across the platform.

Observability Platform Engineer (OPE)

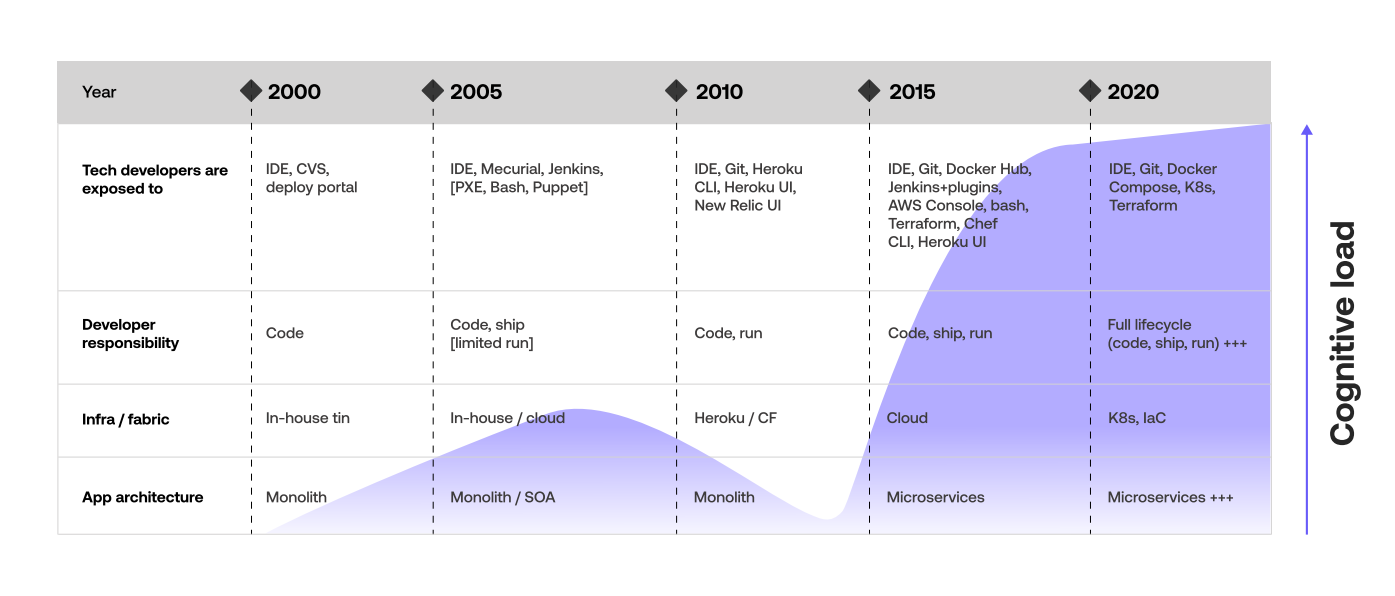
An evolution of SRE, responsible for reliability and observability standards, resource optimization, and overseeing monitoring and observability. Plays a key role in tuning production resources.

AI-focused platform engineers

The integration of AI has introduced new specialties to platform teams, including AI engineers, data engineers, and MLOps specialists. The specialized title Data and AI platform engineer has emerged to interface with data science teams.

The end of the artisan software engineer

For almost two decades, software development inside most organisations operated like a craft: manual, bespoke, and heavily dependent on individual expertise. Development relied on the knowledge and expertise of the individual. That senior engineer who knew your organization's codebase inside and out. This came with it the dominance of concepts like the "10x coder" or the "Full-stack engineer". This phenomenon was pushed to its limits as technological advancement and new philosophies like DevOps blurred the lines between domains. And so, engineers were (and in many cases still are) expected to understand everything from their increasingly extensive deployment tooling to infrastructure, security, and beyond.



Inspired by Daniel Bryant at PlatformCon 2022

This also now meant that previously simple requests like provisioning a database could trigger long, multi-step processes involving multiple teams, as the lines between roles became more and more blurred. This fragmentation led to slow delivery, inconsistent quality, and unnecessary cognitive load, making it difficult for organisations to scale engineering output reliably.

It also meant that the role of the software engineer was often that of the master craftsman. As they increasingly grew their knowledge base and skillset, organizations grew dependent on their understanding of a vast set of skills, languages, tools, and the growing complexity of an organization's IT.

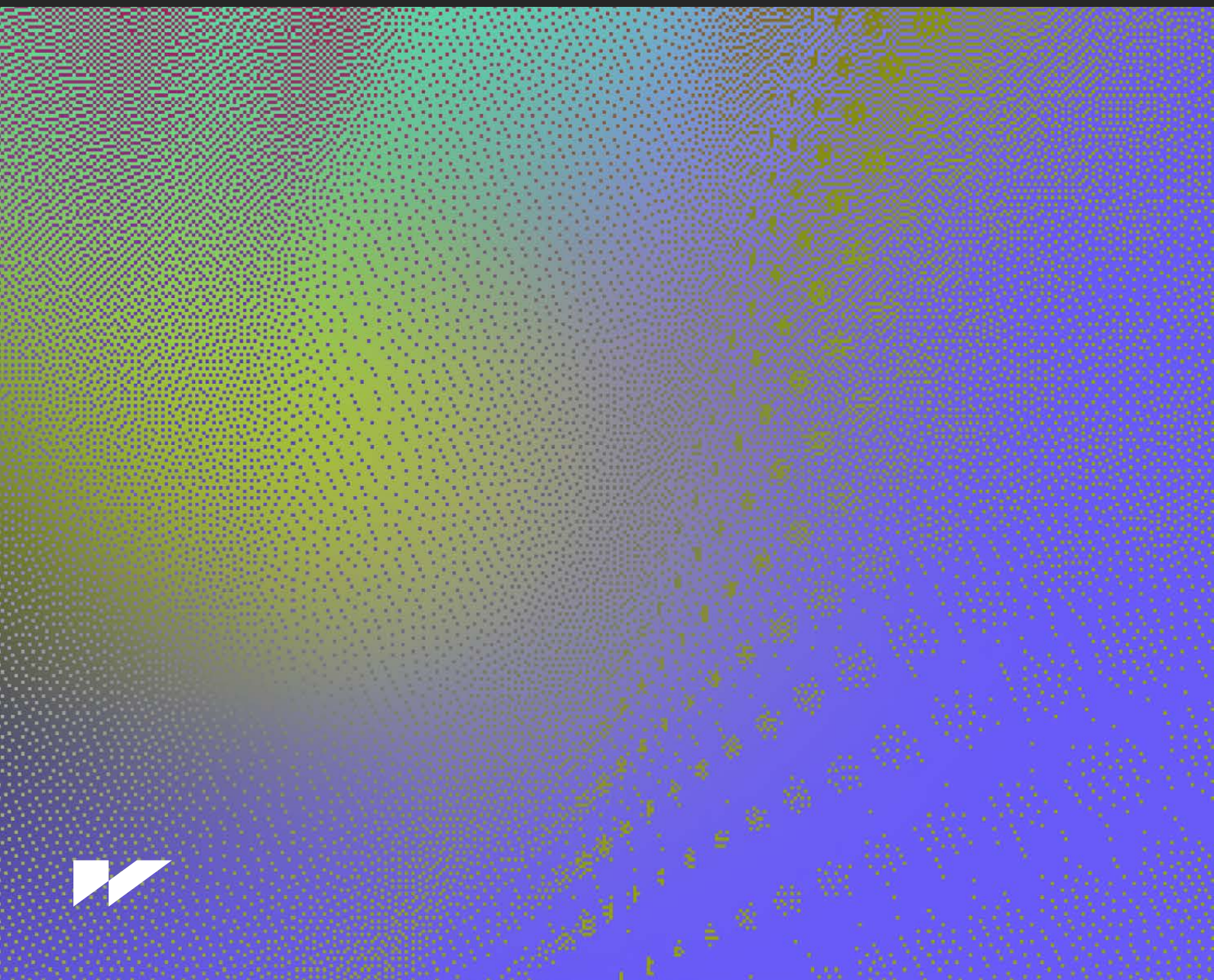
> For hundreds of years, the master craftsman made the perfect handwoven chair. It was expert quality and had a cost and waiting time to match. Now, all chairs are produced en masse by extensive industrialized supply chains. Software is going through the same process.

Luca Galante

CORE CONTRIBUTOR TO PLATFORM ENGINEERING COMMUNITY

Platform engineering is changing this. The automated and systemized approach platform engineering takes to software can be thought of as industrializing the entire software delivery process. It standardises core services, abstracts complexity behind clear interfaces, and enables developers to consume capabilities such as databases, infrastructure, and deployment pipelines through intuitive self-service workflows. By treating foundational components as reusable products rather than custom projects, organisations thus unlock a reliability, speed, and predictability that was previously dependent on the individual rather than the system itself. The result is a shift from artisanal, one-off engineering work to a streamlined assembly-line model where teams can build and ship software quickly, safely, and at scale.

Reference architecture *and* tooling landscape updates in 2025

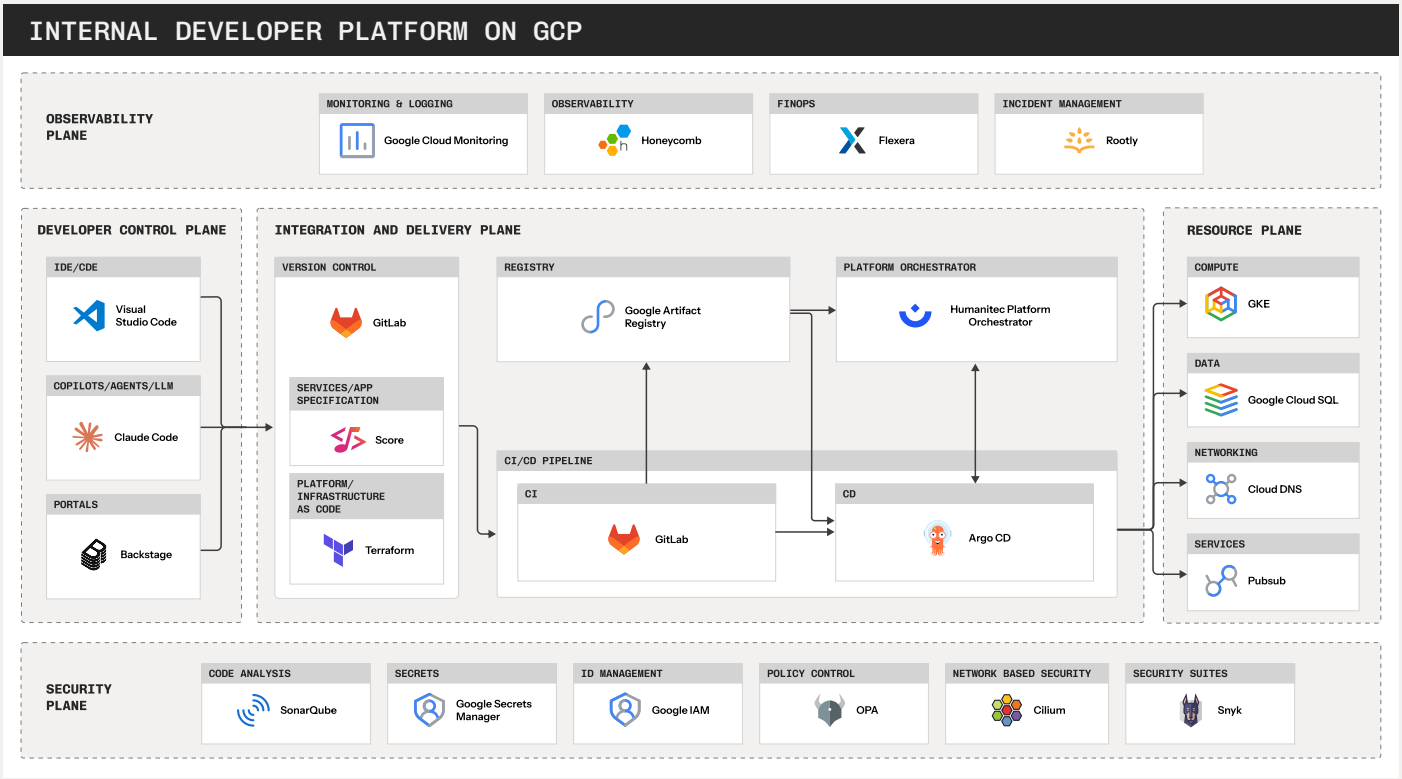


Since the first standard reference architecture for Internal Developer Platforms (IDPs) debuted at PlatformCon in June 2023, it has rapidly evolved from a simple blueprint into a widely adopted industry benchmark. With over 100,000 downloads and continuous use by hundreds of teams, it has helped organizations plan and implement effective platform engineering initiatives.

As the practitioner community has grown to more than 270,000 members, we've gained unparalleled insight into the full spectrum of platform programs, both successes and failures. Over the same period, AI has reshaped the Software Development Life Cycle, adding new complexity and opportunity. These developments, combined with data from real-world enterprises, hundreds of conversations, and 480 platform examples submitted through the communities platform engineering certification programs, have informed a new, modernized reference architecture.

Updated reference architecture of Internal Developer Platforms

Version 2.0 of the reference architecture evolves the original model based on two years of real-world adoption, expanded industry maturity, and the rapid emergence of AI in the SDLC.



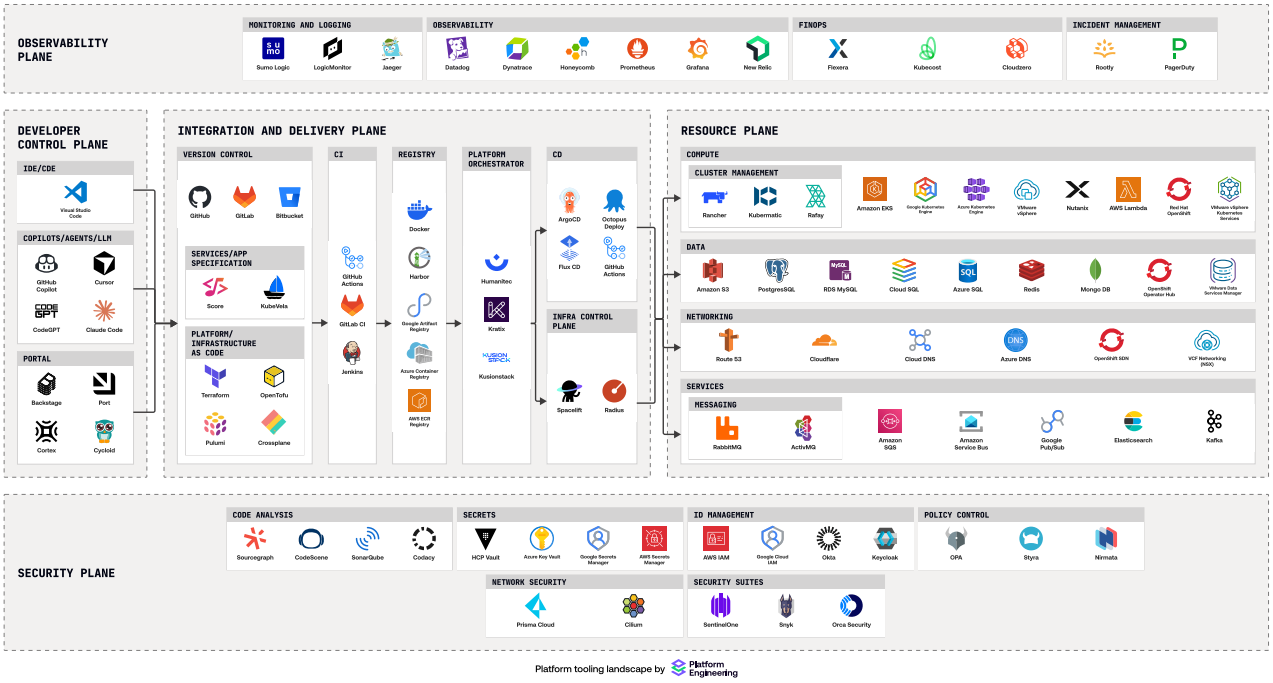
The biggest change is the recognition of a multi-platform reality: most enterprises now operate several platforms across backend, frontend, data/AI, and mobile, rather than a single unified system. The new model also shifts to code as the single source of truth, with GitOps-driven workflows ensuring every action is logged, versioned, and auditable.

Developer interfaces like IDEs, CDEs, portals, CLIs, and AI-driven conversational tools are elevated as the primary interaction layer. Security and observability become foundational architectural planes rather than add-ons, reflecting both growing platform complexity and AI-related risks.

Together, these changes create a more realistic, secure, and scalable framework for modern enterprise platforms. This example below is that of an Internal Developer Platform based on GCP, and using the highlighted tools. However, all cloud providers and tools are interchangeable. You can download the full reference architecture whitepaper here. You can also see a reference architecture for Azure.

Platform tooling landscape in 2025

This newly created reference architecture has also resulted in a redefined platform tooling landscape on platformengineering.org.



Platform tools

Type to search

Results 141 Showing items 1 to 21 of 141

Sorted by A-Z

Planes Clear

- Monitoring & Logging Plane 13
- Developer Control Plane 25
- Resource Plane 34
- Integration & Delivery Plane 28
- Security Plane 17
- Resource Plane 34
- Other Tools 22

AWS ECR Registry
RESOURCE PLANE
REGISTRY
Amazon Elastic Container Registry (ECR) is a fully manage...

AWS RDS
RESOURCE PLANE
DATA
Amazon RDS is a managed relational database service that...

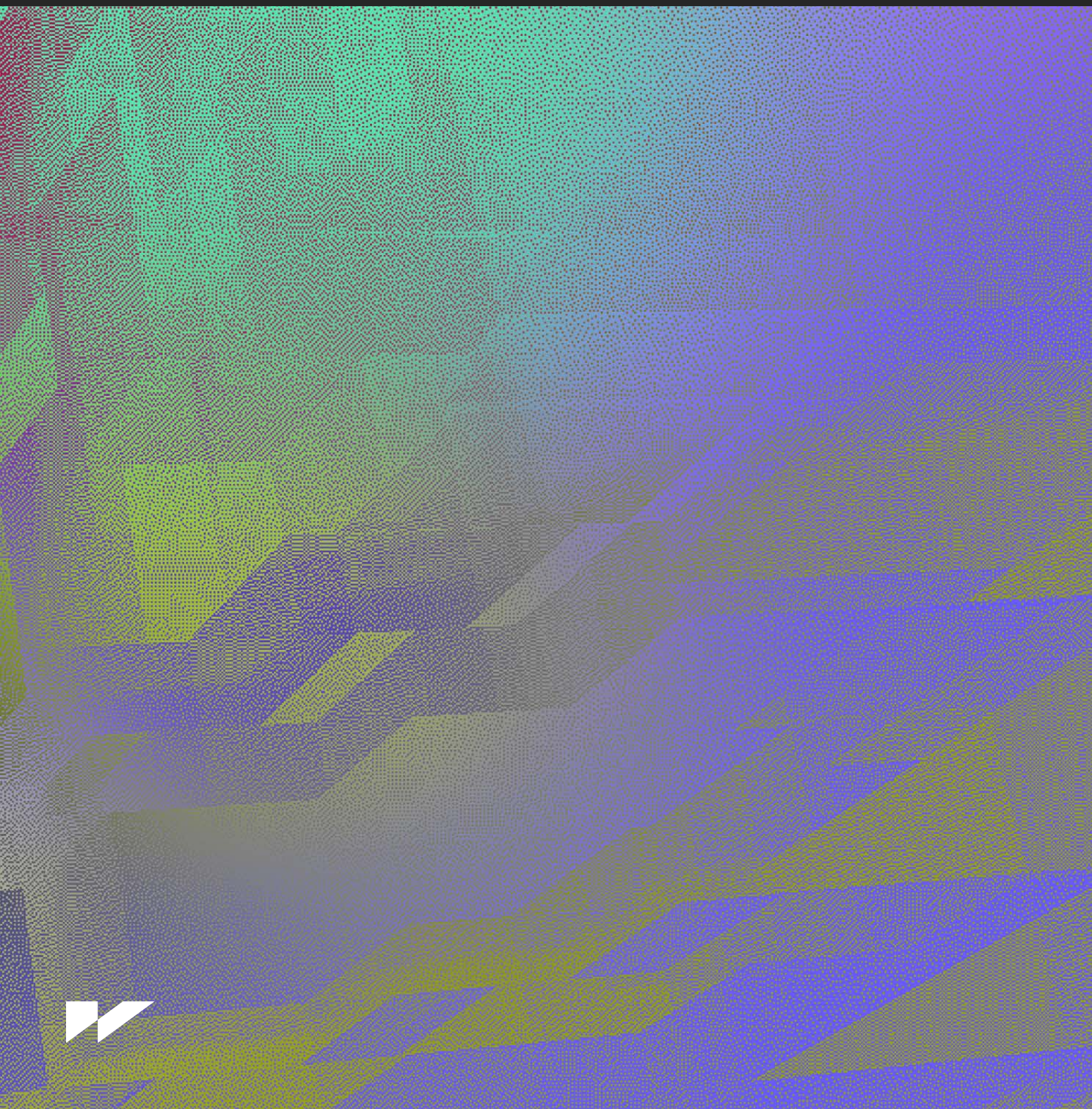
ActiveMQ
INTEGRATION & DELIVERY PLANE
MESSAGING
Apache ActiveMQ is an enterprise-grade message...

Aiven
RESOURCE PLANE
DATA

Akuity
INTEGRATION & DELIVERY PLANE
CD PIPELINE

Amazon S3
RESOURCE PLANE
DATA

AI *and* platform engineering



AI has fundamentally redefined the future of the enterprise, pushing us beyond the cloud-native era and into the AI-native era. This profound shift has necessitated that platforms and governance structures evolve. Platform teams now need to efficiently handle GPU-accelerated workloads, specialised agentic systems, and new operational patterns. This transformation has also introduced to platform engineering teams a dual mandate:

01 — AI-powered platforms

Integrating AI tools to augment Internal Developer Platforms (IDPs). The objective is to turbocharge core platform benefits by enhancing developer productivity, improving compliance, and driving automation through features like intelligent troubleshooting or automated security scanning.

02 — Platforms for AI

Building highly specialized infrastructure and tooling specifically designed to enable the efficient deployment, training, and scaling of AI/ML workloads. This requires platform teams to embrace new customers, including data scientists and ML engineers, and provide them with the foundational ecosystems needed for complex model development.

This evolution establishes platform engineering not merely as a consumer of AI, but as the critical architect and enabler of enterprise-wide AI value.

New research on AI in platform engineering

The hypothesis that a strong platform foundation is essential for scalable AI adoption is validated by both the 2025 DORA report and our State of AI in Platform Engineering findings. DORA confirms that AI's impact depends less on individual tools and far more on the quality of the underlying organizational system. With platform engineering now adopted by nearly 90% of organizations, DORA identifies a Quality Internal Platform as one of seven capabilities that significantly amplify AI's positive effect on performance.

Our State of AI in Platform Engineering report reinforces this. An overwhelming 94% of organizations view AI as critical or important to the future of platform engineering, while 86% believe platform engineering is essential to realizing AI's business value. Three-quarters (75%) are already hosting or preparing to host AI workloads, prompting new roles such as AI engineers and MLOps specialists. Yet challenges persist, including skill gaps (57%), hallucination risks (56%), and heightened governance friction.

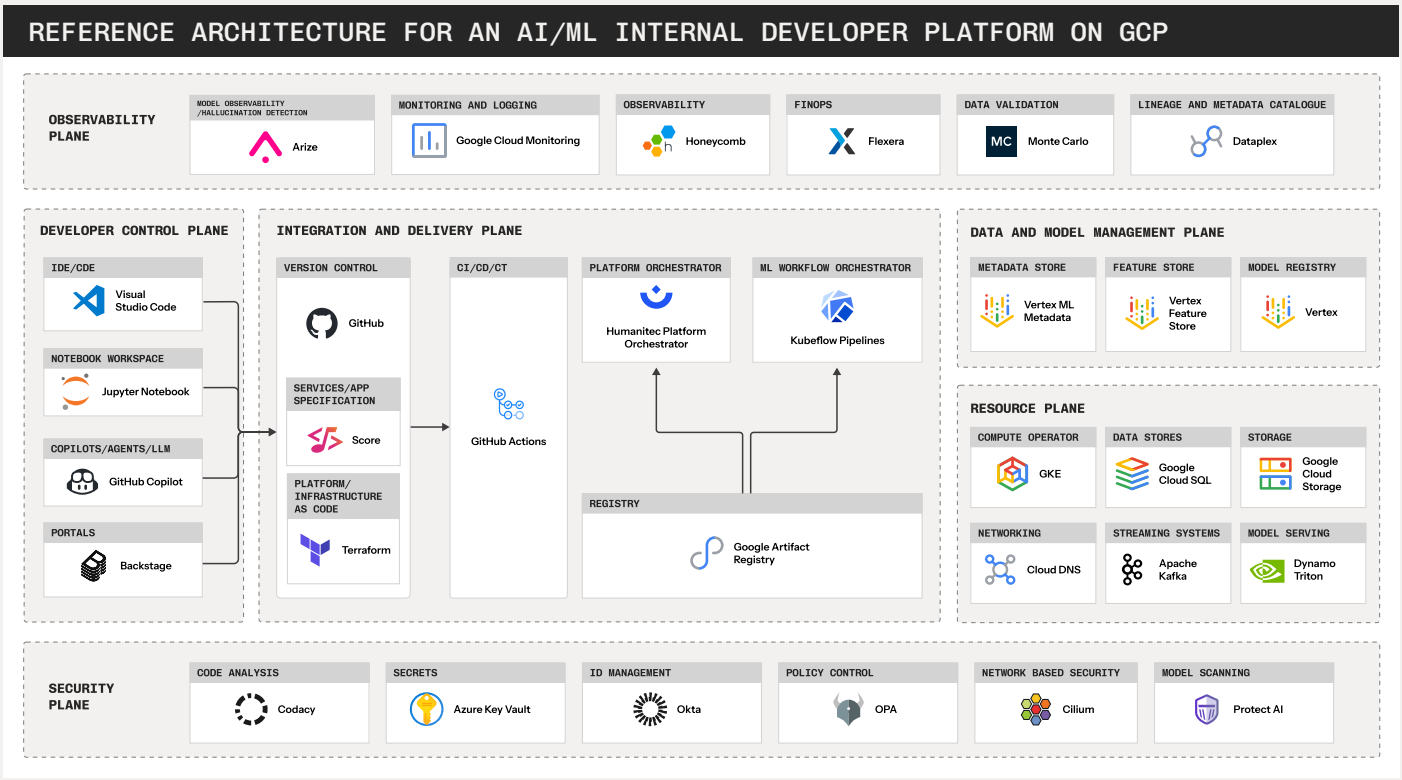
> The data confirms what we all already knew. Everyone is using AI (89% reported), but the ones who are truly succeeding with AI and achieving the ROI dreams that AI promises are those who have a strong platform engineering foundation to supercharge it.

Sam Barlien

AUTHOR OF THE STATE OF PLATFORM ENGINEERING VOL 4

Reference architecture for an AI/ML Internal Developer Platform on GCP

While standard IDPs focus on application delivery, the AI/ML IDP is designed to support the entire lifecycle of data, AI, and ML workloads, bringing new architectural requirements, new personas, and more rigorous governance requirements. This specialised reference architecture (labeled version 0.1) reflects the current immaturity of the industry as AI and platform engineering rapidly converge.



It adds a dedicated Data & Model Management Plane for feature engineering, experiment tracking, metadata, lineage, and model registry functions. The Developer Control Plane expands to include notebooks, LLM copilots, and interfaces tailored to data scientists and ML engineers. The Integration & Delivery Plane adopts a dual-orchestrator model combining platform orchestration with ML workflow automation.

The Resource Plane introduces GPU orchestration, streaming systems, and high-performance model serving, while the Observability Plane adds model monitoring, data validation, drift detection, and lineage tracking. Collectively, these capabilities enable the delivery of governed, production-grade AI systems rather than traditional applications.

Platform Engineering community: What's new?

Launched in 2022, the Platform Engineering community has always been a clear barometer for the industry at large. Now nearing 270,000 members across all manner of roles, industries, and geographies, its expansion has closely mirrored the rise of platform engineering itself.

What began with adoption from DevOps, SREs, and classic operations roles has accelerated as observability, security, AI, and data increasingly fold into platform engineering.

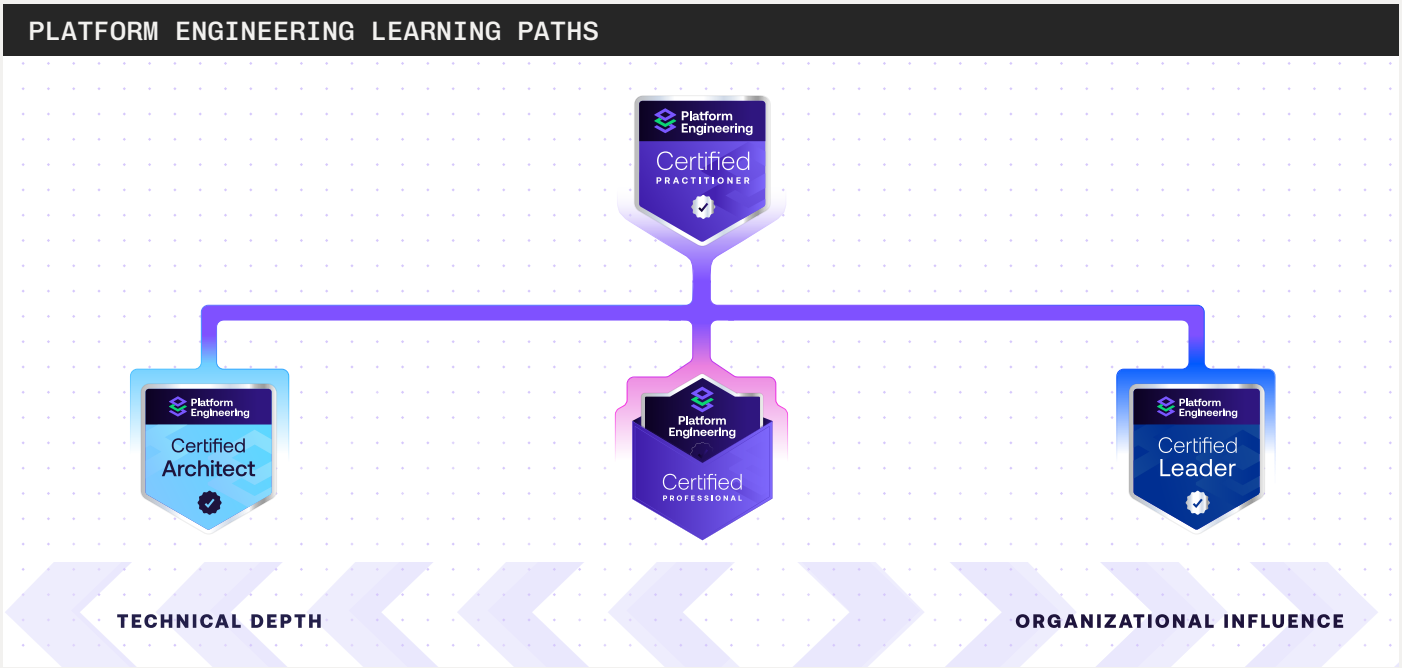
Security and observability are currently the fastest-growing demographic in the platform engineering community.

This growth has also coincided with a massive expansion in community initiatives like it's course ecosystem, ambassador program, work with enterprises, and it's flagship conference, PlatformCon.

Platform Engineering University

The platform engineering certification ecosystem has grown far beyond its original fundamentals track. It now offers a full suite of instructor-led certifications and on-demand courses for practitioners, leaders, and specialists across the discipline.

Platform engineering learning paths includes four instructor-led programs such as the Certified Practitioner, Certified Professional, Certified Leader, and Certified Architect. Each course building deeper skills in platform strategy, design, and architecture through hands-on, expert-led instruction.



Alongside these, the course ecosystem features a growing library of free, on-demand modules covering foundational and emerging topics such as Introduction to Platform Engineering, Intro to AI in Platform Engineering, Cloud Development Environments, Kubernetes Cluster Lifecycle Management, and Observability for Platform Engineering. These shorter courses help teams skill up quickly and stay current with best practices.

PLATFORM ENGINEERING COURSE ECOSYSTEM

Together, this expanded curriculum forms the most comprehensive learning pathway in platform engineering today, with additional courses on portals, security, AI, and tooling planned through 2026.

Ambassador program

As a collective effort to grow and nurture the platform engineering space, the community launched the Platform Engineering Community Ambassadors program in late 2024. It continues to serve as a way for practitioners and thought leaders to come together, exchange ideas, discuss best practices, and share lessons learned while receiving support to bring that knowledge to the wider community. Today, more than 90 ambassadors actively publish research, create content, and help elevate the discipline of platform engineering.

Training and advisory

The community's extensive learning and research capabilities are applied directly to support enterprises through two flagship offerings: Training and Advisory. Through the training programme organisations receive tailored workshops whether in-person or virtual that equip platform teams, product managers, architects and other stakeholder groups with the frameworks and skills needed to execute a shared platform vision. Meanwhile, the advisory service provides expert diagnosis of adoption blockers, alignment of platform strategy with business goals, and hands-on deployment support from Minimum Viable Platform to full production scale.

Together, these services have transformed raw practitioner insight and community research into enterprise-grade enablement, enabling structured upskilling, measurable adoption, and lasting organisational change for a number of enterprises in the platform engineering industry.

PlatformCon 2025

The world's largest platform engineering conference, PlatformCon 2025 marked a new milestone in the maturity and scale of the platform engineering industry. This year's edition brought together an unprecedented volume of knowledge with 255+ talks, 30 virtual workshops, and 20+ live day sessions - over 160 hours of platform engineering goodness.

Alongside these live events, 9,500 people participated in the virtual workshops, while nearly 40,000 more engaged with PlatformCon's wider universe of talks, panels, and community conversations. Beyond its scale, PlatformCon 2025 also reflected a clear shift: the conversation has moved decisively from "What is platform engineering?" to "What does great platform engineering look like?".

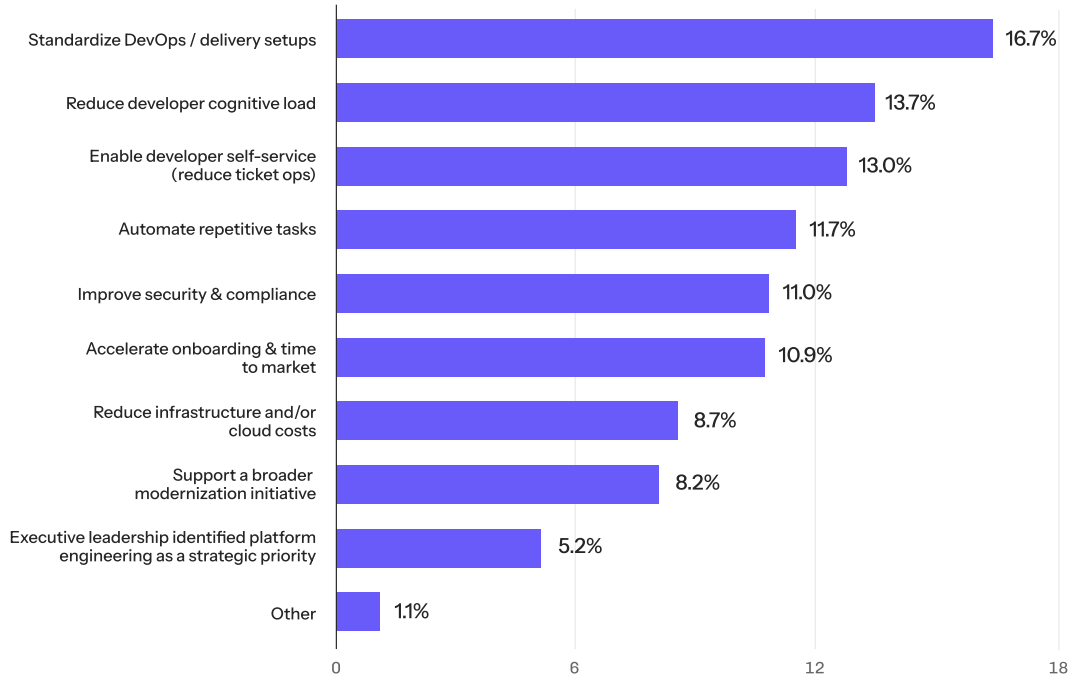
Survey results

The data for this State of Platform Engineering report was gathered across the platform engineering community. We spoke to 518 engineers across industries, geographies, job titles and levels of seniority. The questions ranged from the individual on job title, years experience, salary, and area of focus, to the organizational like platform engineering maturity, platform budget, time to ROI and far more.

Main reasons to set up a platform engineering team

The main motivators for creating a platform engineering team center on improving developer experience and streamlining the delivery pipeline. The most cited reason is standardizing DevOps and delivery setups, closely followed by reducing developer cognitive load and enabling self-service to move away from ticket-driven "TicketOps."

MAIN REASONS TO SET UP A PLATFORM ENGINEERING TEAM

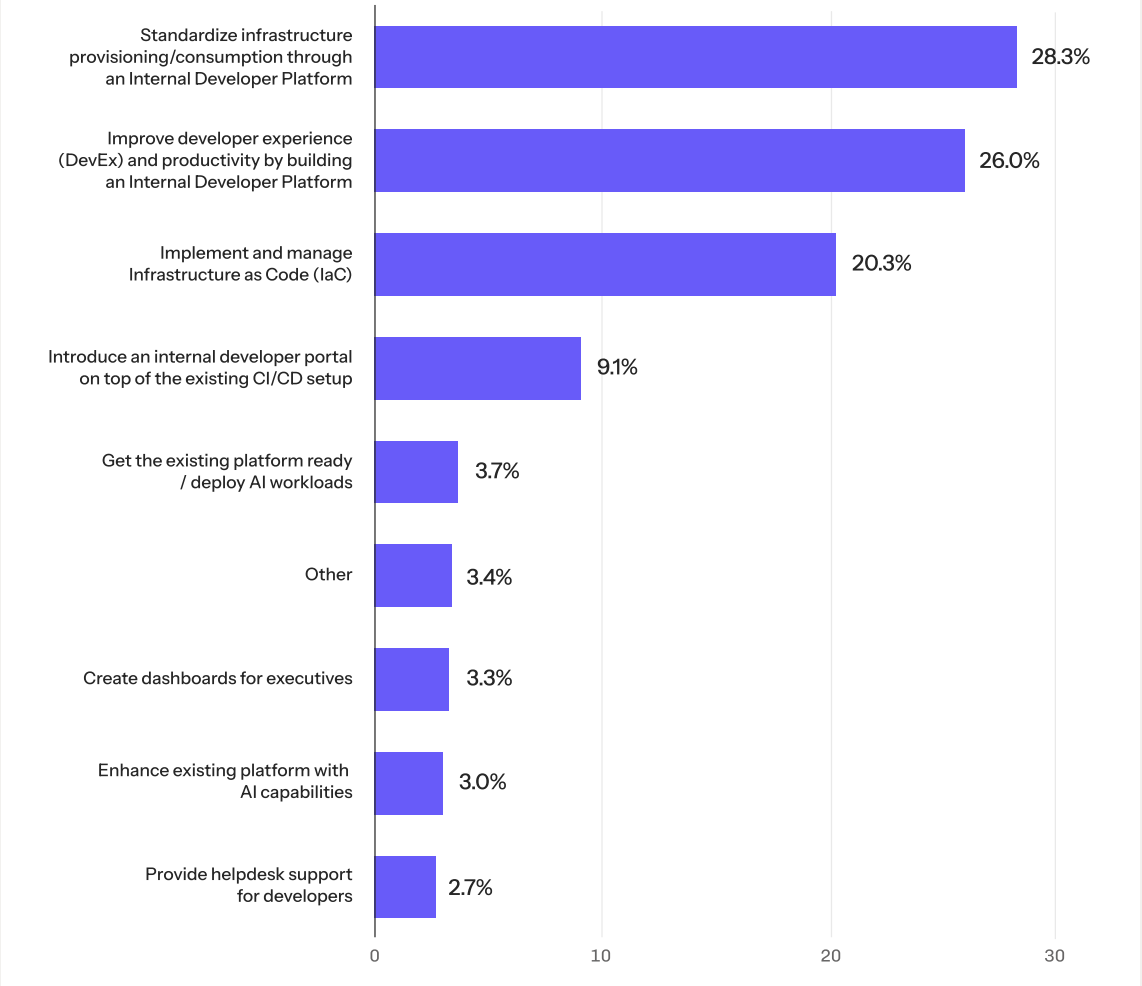


Together, these highlight a strategic shift toward greater efficiency and developer autonomy. Automating repetitive tasks and strengthening security and compliance remain significant drivers, balancing efficiency with risk reduction. While accelerating time to market and lowering costs matter, the overarching theme is the strategic push to standardize, simplify, and support developer flow.

Main focus of the platform engineering team

Survey data shows platform engineering teams remain focused on foundational enablement and developer experience. The top priorities are standardizing infrastructure provisioning through an Internal Developer Platform (28.3%) and improving developer experience and productivity (26%). Implementing and managing Infrastructure as Code (20.3%) is also a major focus, reinforcing the trend toward automated, codified infrastructure.

THE MAIN MOTIVATORS FOR CREATING A PLATFORM ENGINEERING TEAM

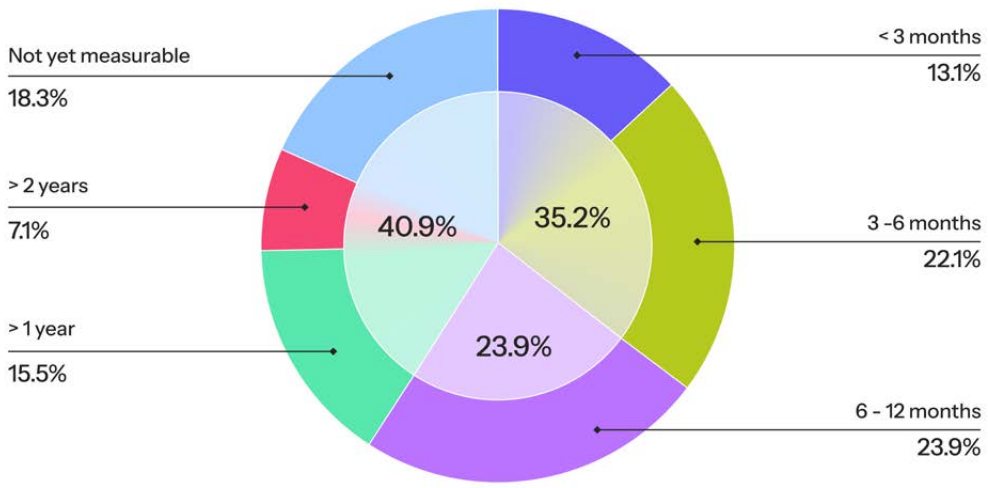


Notably, adding a developer portal on top of an existing CI/CD setup represents only 9.1%, indicating the industry has moved beyond the earlier "portal trap." AI-related efforts like preparing platforms for AI workloads (3.6%) and adding AI capabilities (3%) form a small but emerging area, supporting the report's theme of AI integration while showing that core platform foundations remain the top priority for most teams.

Time to deliver value

The time to value delivered by platform engineering initiatives shows a bimodal distribution, indicating that early success is achievable but sustained value creation often takes longer. A significant portion of teams (35.2%) begin delivering measurable value within the first six months, demonstrating the effectiveness of an iterative, Minimum Viable Platform (MVP) approach.

DISTRIBUTION OF TIME TO DELIVER MEASURABLE VALUE



Specifically, 11.3% report seeing value in less than three months, and 23.9% achieve it within three to six months. 40.9% of platform initiatives are unable to demonstrate measurable value within the first twelve months. For these initiatives, particularly the 18.3% that report having no measurable results at all, there is a substantial risk that they will remain underfunded, fail to secure necessary executive sponsorship, or face outright deprecation. The inability to quickly and clearly prove ROI is a major vulnerability, especially in environments where platform teams are already struggling with insufficient resources and the need to establish a clear business case.

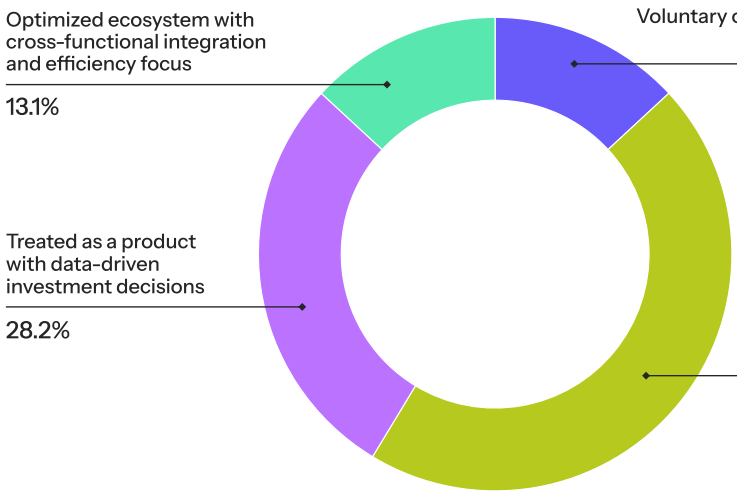
Platform engineering maturity

Similar to our methodology in the previous report, this analysis leveraged the CNCF Platform Engineering Maturity Model. This model provided the framework for questions centered on key areas: investment, adoption, interfaces, operations, and measurement.

Investment

The distribution of staff and funding models shows that most organizations are still in the early stages of platform maturity. The largest share, at 45.5%, have a dedicated, budgeted team that remains mostly reactive, which suggests that platform functions are established but not yet strategic.

INVESTMENT: ALLOCATION OF STAFF AND FUNDS TO PLATFORM CAPABILITIES

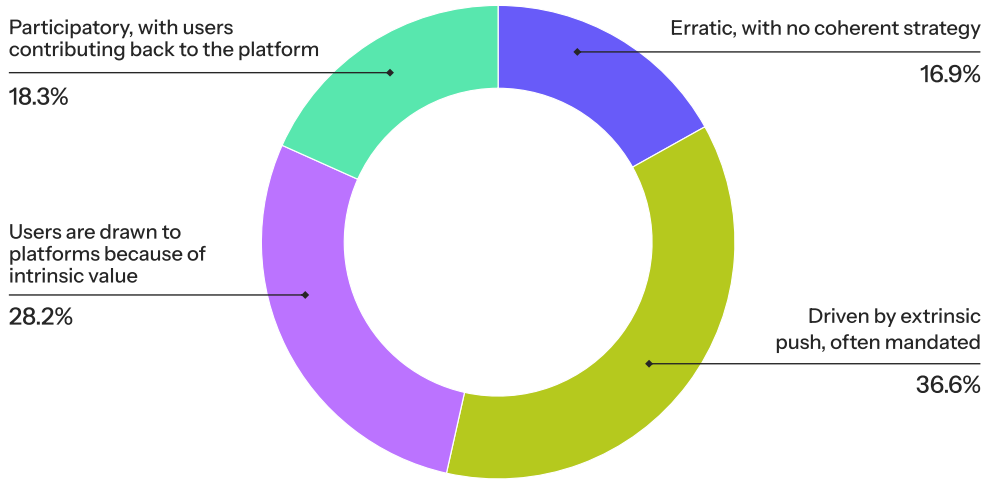


Another 13.1% rely on voluntary or temporary, unfunded assignments, showing that some teams still operate in an ad hoc way. More advanced maturity appears in the 28.2% who treat the platform as a product with data-driven investment decisions. Only 13.1% report an optimized, cross-functional ecosystem focused on efficiency, indicating that although progress is clear, fully mature models remain uncommon.

Adoption

The data shows that platform adoption varies widely in maturity. The largest segment, at 36.6%, adopts platforms through extrinsic push, which indicates reliance on top-down mandates. Another 16.9% experience erratic adoption with no clear strategy, suggesting fragmented communication or offerings.

ADOPTION: WHY AND HOW USERS DISCOVER AND USE INTERNAL PLATFORMS AND PLATFORM CAPABILITIES

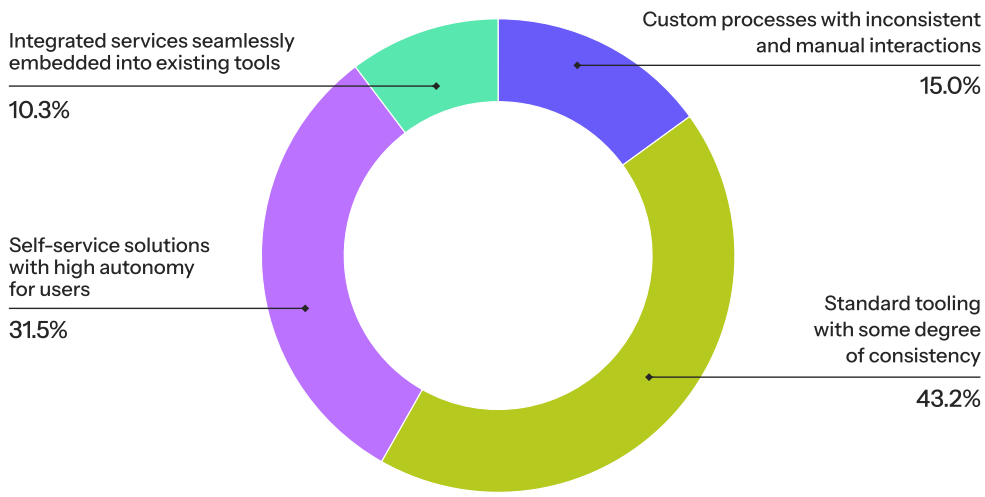


More positive patterns appear in the 28.2% who adopt platforms because of their intrinsic value, showing that users respond when the platform is genuinely useful. A further 18.3% report participatory adoption in which users contribute back, reflecting early but still limited signs of community-driven engagement.

Interfaces

The data on how users interact with platform capabilities shows that most teams rely on familiar and moderately consistent interfaces. Standard tooling is the most common approach at 43.2%, suggesting that many organizations prioritize predictability and familiarity. Another 31.5% provide self-service solutions that offer users a high degree of autonomy, indicating growing investment in more advanced platform interactions.

INTERFACES: HOW DO USERS INTERACT WITH AND CONSUME PLATFORM CAPABILITIES?

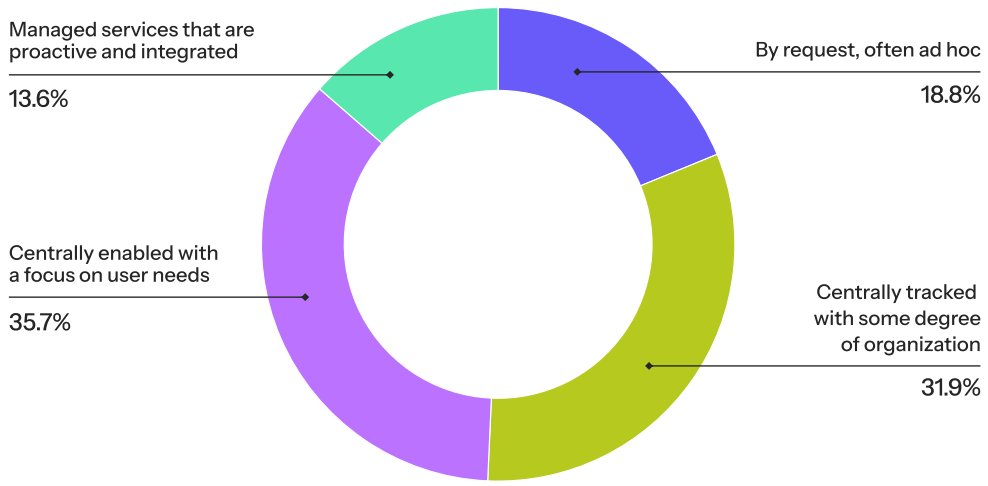


Custom processes with inconsistent and manual steps still account for 15.0%, showing that some teams continue to depend on bespoke or ad hoc workflows. Only 10.3% report fully integrated services seamlessly embedded into existing tools, which highlights that while integration is a clear aspiration, it remains relatively rare in practice.

Operations

The data shows that most organizations manage platform operations with some level of central structure, although maturity varies. The largest group, at 35.7%, centrally enables platform work with a strong focus on user needs, indicating a shift toward more service-oriented operations.

OPERATIONS: HOW ARE PLATFORMS AND THEIR CAPABILITIES PLANNED, PRIORITIZED, DEVELOPED, AND MAINTAINED?

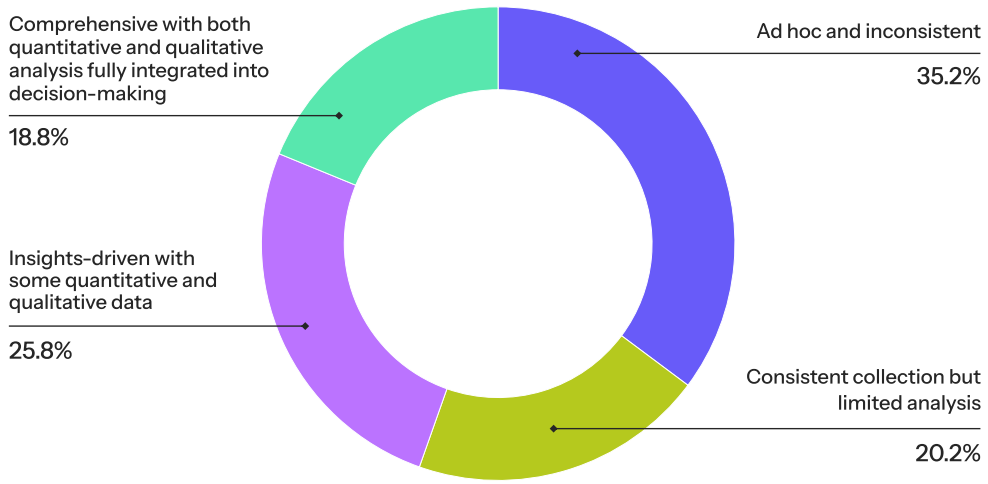


Another 31.9% track work centrally with only partial organization, which suggests that processes are improving but not yet fully aligned. At the same time, 18.8% still operate by request in an ad hoc manner, reflecting environments where prioritization is inconsistent. Only 13.6% report having proactive and integrated managed services, showing that fully mature operational models remain uncommon.

Measurement

The measurement practices reveal a similar spread in maturity. The largest share, at 35.2%, relies on ad hoc and inconsistent feedback, which limits continuous improvement. Another 20.2% collect feedback consistently but apply limited analysis, indicating that data is gathered but not fully leveraged.

MEASUREMENT: WHAT IS THE PROCESS FOR GATHERING AND INCORPORATING FEEDBACK AND LEARNING?



More advanced practices appear in the 25.8% who use both quantitative and qualitative insights to inform decisions. Only 18.8% achieve comprehensive measurement with fully integrated analysis, showing that while some organizations have strong data-driven loops, most are still developing robust learning processes.

Comparison to platform engineering maturity results from previous year

The CNCF Platform Engineering Maturity Model highlights a clear pattern of steady, incremental progress across every area of platform engineering compared with last year's survey. Investment has shifted slightly further toward product thinking and early signs of ecosystem enablement, showing that more organizations are beginning to treat their platforms as strategic assets rather than reactive functions. Adoption has also improved, with fewer teams stuck in erratic or mandate-driven usage and more reporting genuine pull from developers who see real value in the platform. Interfaces show a similar upward trend, with gradual movement away from manual, custom processes and toward more consistent tooling and early self-service experiences.

2024 2025

PLATFORM ENGINEERING MATURITY

ASPECT		PROVISIONAL	OPERATIONAL	SCALABLE	OPTIMIZING
Investment	How are staff and funds allocated to platform capabilities?	Voluntary or temporary 43.3% 45.5%	Dedicated team	As product	Enabled ecosystem 12.2% 13.1%
Adoption	Why and how do users discover and use internal platforms and platform capabilities?	Erratic 35.8% 36.6%	Extrinsic push	Intrinsic pull	Participatory 17.3% 18.3%
Interfaces	How do users interact with and consume platform capabilities?	Custom processes 42.1% 43.2%	Standard tooling	Self-service solutions	Integrated services 9.1% 10.3%
Operations	How are platforms and their capabilities planned, prioritized, developed and maintained?	By request	Centrally tracked 39.3% 35.7%	Centrally enabled	Managed services 10.7% 13.6%
Measurement	What is the process for gathering and incorporating feedback and learning?	Ad hoc 42.5% 35.2%	Consistent collection	Insights	Quantitative and qualitative 10.4% 18.8%

Operations and measurement likewise demonstrate modest but meaningful gains, with more teams introducing structured processes, user-focused prioritization, and data-informed learning loops.

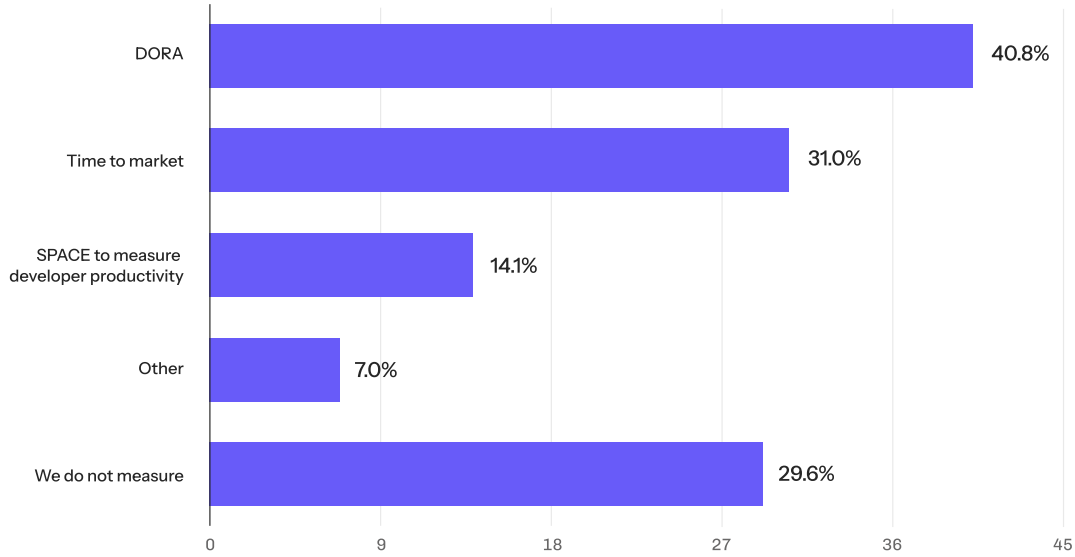
While none of these shifts are dramatic, the collective pattern points to an industry that is maturing year over year. At the same time, the improvements remain incremental, and the overall landscape still sits far from the fully optimized, ecosystem-level maturity that platform engineering aims for. We've still got a lot of room for improvement.

Metrics

Which metrics to measure to prove success

The data on how teams measure the success of their platform engineering initiatives reveals a striking split between those using established metrics and those still lacking any measurement practice. DORA metrics remain the most widely used approach, cited by the largest share of respondents, followed by time to market and, to a lesser extent, SPACE metrics to assess developer productivity. These signal a growing reliance on structured, industry-recognized methods for evaluating platform impact.

WHICH METRICS TO MEASURE TO PROVE SUCCESS

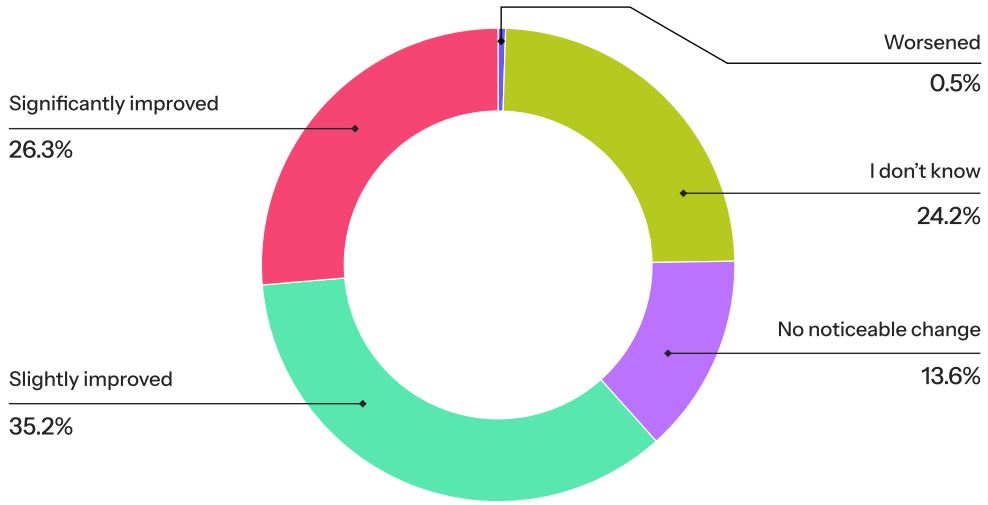


However, the most concerning finding is that 29.6% of teams do not measure success at all, which underscores a major gap in accountability across the industry. This absence of measurement limits the ability to demonstrate value, guide investment decisions, and refine platform capabilities. Taken together, the results show that while measurement practices are improving, a significant portion of organizations still lack the feedback loops necessary for sustained, data-driven platform progress.

Impact on metrics

The impact of platform engineering on organizational metrics shows a generally positive trend, though with clear signs that many teams are still early in their journey. A combined majority reports improvement, with the largest share seeing slight gains and another substantial group noting significant improvement, indicating that platform initiatives are delivering measurable benefits for most adopters.

TO WHAT EXTENT HAVE METRICS IMPROVED SINCE YOUR ORGANIZATION INTRODUCED PLATFORM ENGINEERING?



However, 13.6% report no noticeable change, and nearly a quarter of respondents say they do not know whether metrics have improved at all, which suggests gaps in visibility, measurement, or communication. Only a very small fraction reports any worsening of metrics. Overall, the results show that platform engineering is moving the needle in the right direction, but the mixed levels of awareness and the persistent absence of measurable impact for some teams point to the need for stronger feedback loops and clearer outcome tracking.

AUTHOR'S NOTE

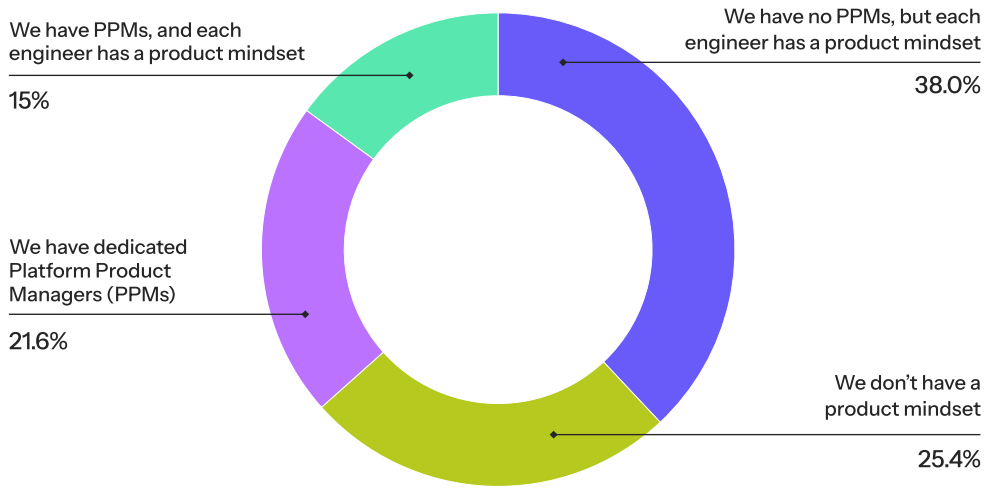
Last year's survey revealed that 45% of respondents "Do not measure", so, there has been a large improvement compared to last year. However, one key problem still remains. Only 24.4% of respondents this year reported that they "do not know" if metrics have improved, despite 29.6% answering us that they "do not measure". This gives us a 5% delta of liars.

How teams approach Platform as a Product

The data on how teams approach a Platform as a Product mindset shows that while awareness is growing, true product thinking is still a challenge for teams. This data also begins to answer one of the largest questions in the platform engineering industry from the last few years, “does your organization need a dedicated Platform Product Manager?”.

The largest group, at 38%, reports having no dedicated Platform Product Managers but says engineers operate with a product mindset, indicating the distributed approach to product ownership that was thought to be best practice from 2022-24. Another 25.4% say they do not have a product mindset at all, which highlights a significant barrier to achieving platform maturity. More structured models appear in the 21.6% who have dedicated Platform Product Managers, and an additional 15% combine PPMs with engineers who also think in product terms.

HOW DOES YOUR TEAM ENSURE A PLATFORM AS A PRODUCT (PAAP) MINDSET?

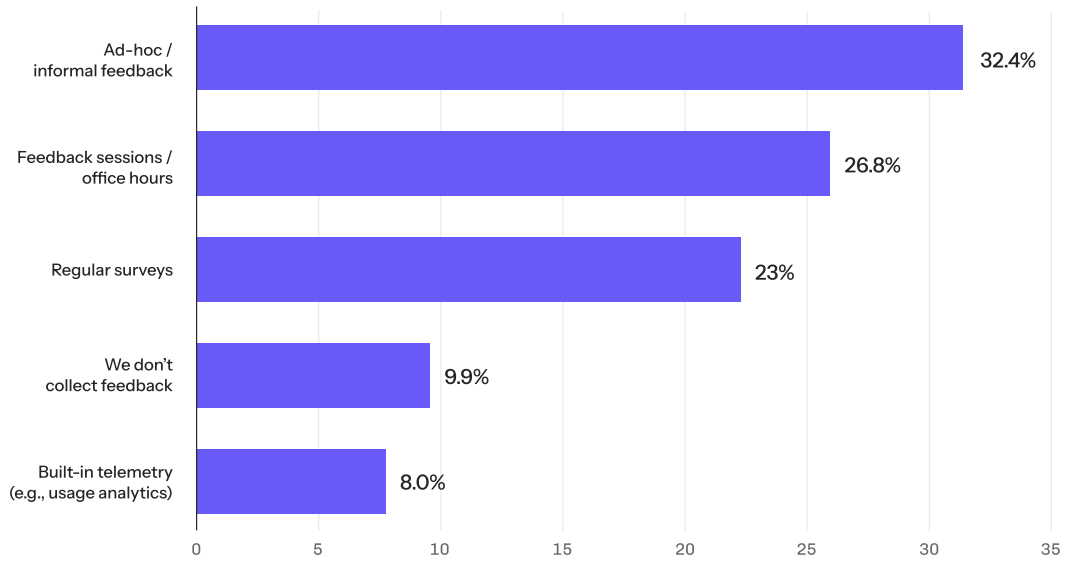


Together, the results suggest that teams are beginning to adopt product-oriented behaviors, but dedicated product leadership and a consistently applied mindset remain far from universal.

How platform engineers gather feedback about the platform

The data shows that platform engineers rely mostly on informal methods to gather feedback. Ad hoc conversations are the most common approach, with feedback sessions and office hours close behind, indicating that real-time discussions continue to drive most insights.

HOW DO PLATFORM ENGINEERS GATHER FEEDBACK ABOUT THE PLATFORM

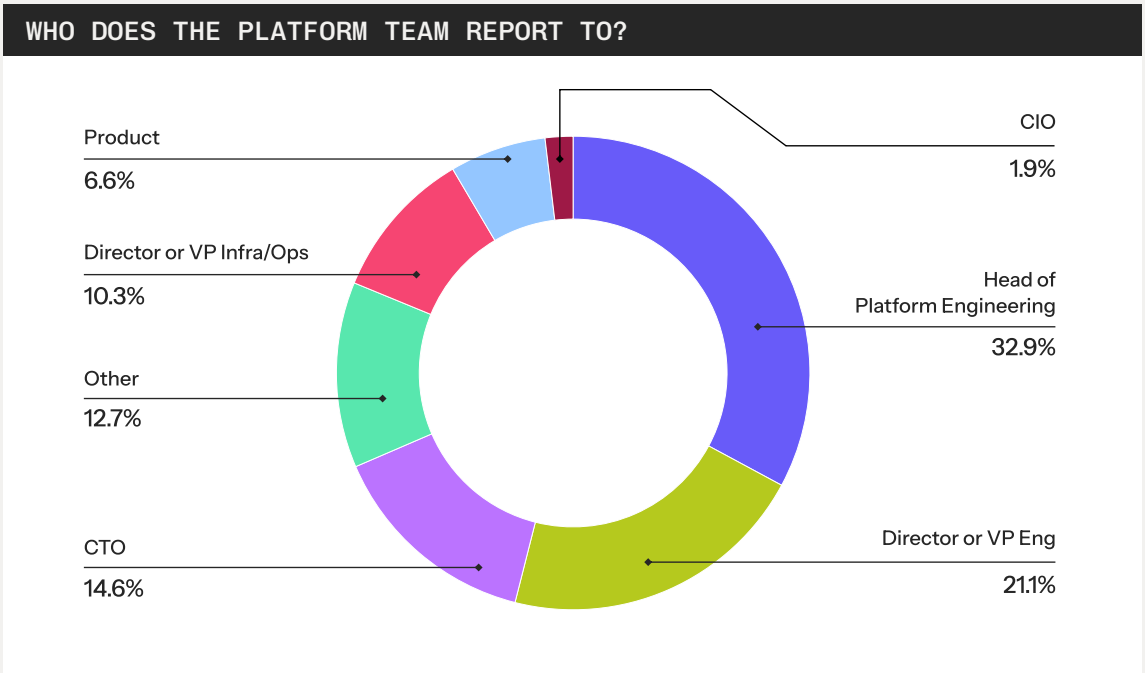


Regular surveys add some structure, though they are used less frequently. Nearly 10% of teams do not collect feedback at all, revealing a notable gap in understanding platform performance. Built-in telemetry and usage analytics are the least used method, showing that automated, data-driven feedback is still early in adoption.

Nearly *10%* of teams do not collect feedback at all, revealing a notable gap in understanding platform performance

Reporting lines

The data on reporting structures shows that platform teams most commonly report to a Head of Platform Engineering, which reflects the growing formalization of platform leadership. The next largest groups report into Director or VP Engineering roles and the CTO, indicating that platform engineering remains closely aligned with core engineering priorities in many organizations.

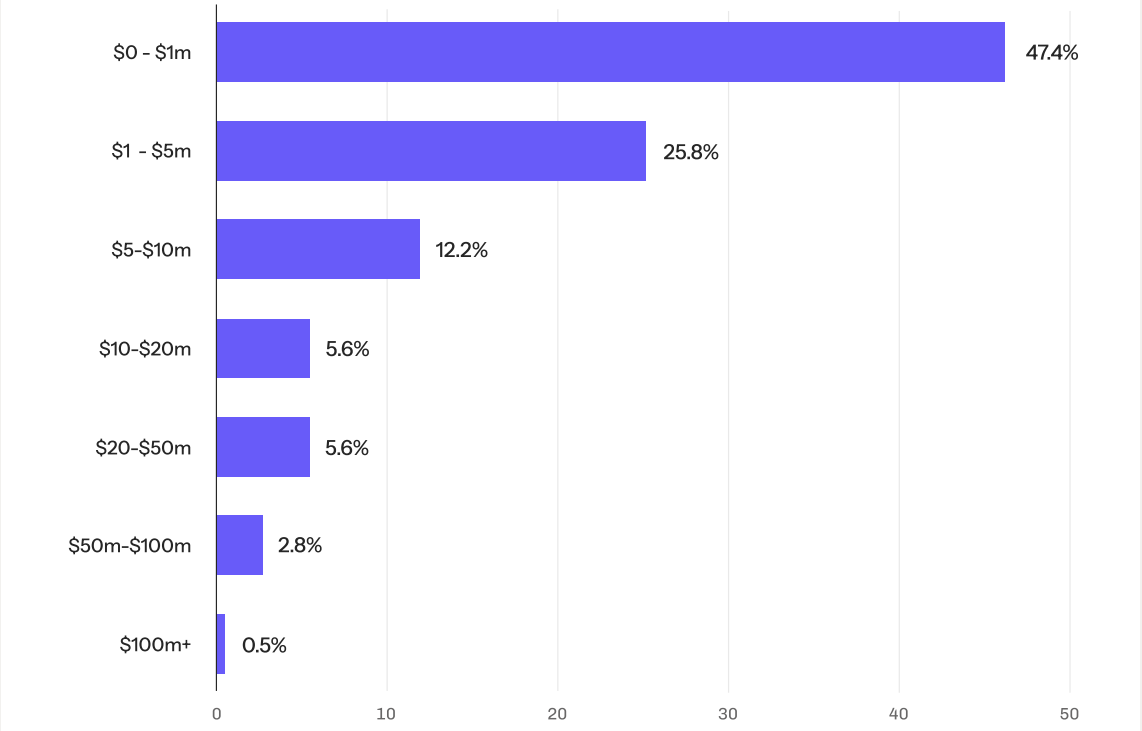


Smaller portions report into infrastructure or operations leadership, product teams, or other structures, highlighting the ongoing variability in how companies position platform work. The distribution suggests that while dedicated platform leadership is becoming more common, there is still no single dominant organizational model, and reporting lines continue to evolve as teams mature.

Annual budget of platform engineering initiative

The data on annual platform engineering budgets shows that most initiatives continue to operate with very limited funding. Nearly half fall into the zero to one million dollar range, highlighting that many teams are expected to deliver broad organizational impact with modest resources.

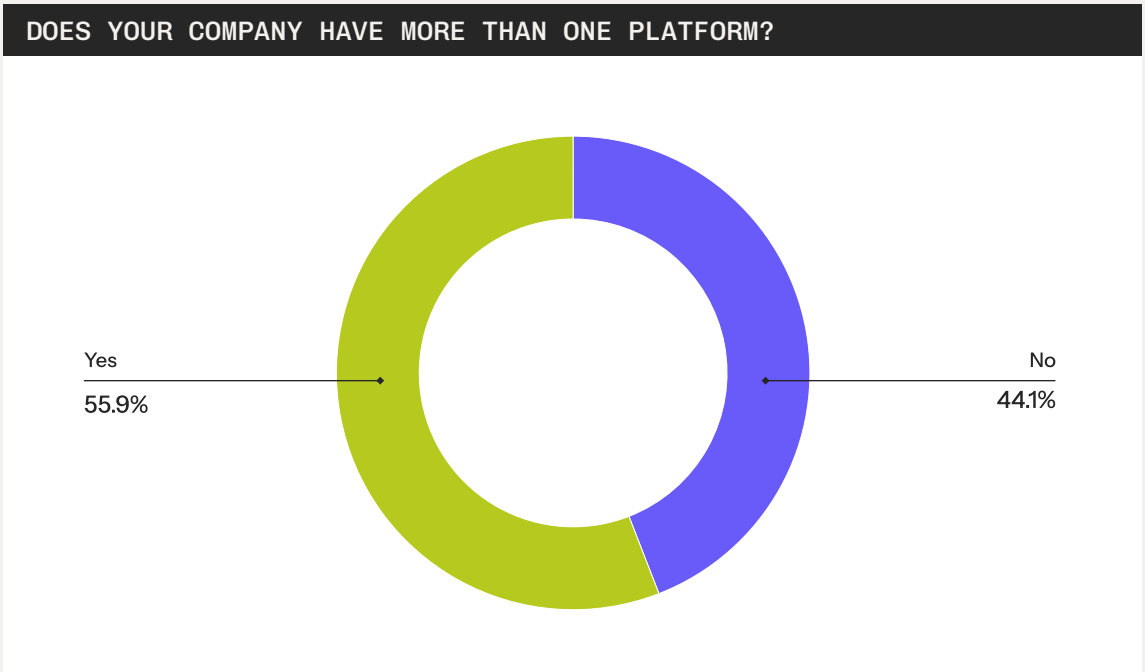
ANNUAL BUDGET OF PLATFORM ENGINEERING INITIATIVE



Another sizeable portion sits in the one to five million dollar range, while only a small minority report budgets above ten million dollars, and an even smaller share exceeds fifty million. This distribution makes it clear that although platform engineering is gaining strategic recognition, financial investment has not yet caught up, and most initiatives remain underfunded relative to their scope and expectations.

Does your company have more than one platform?

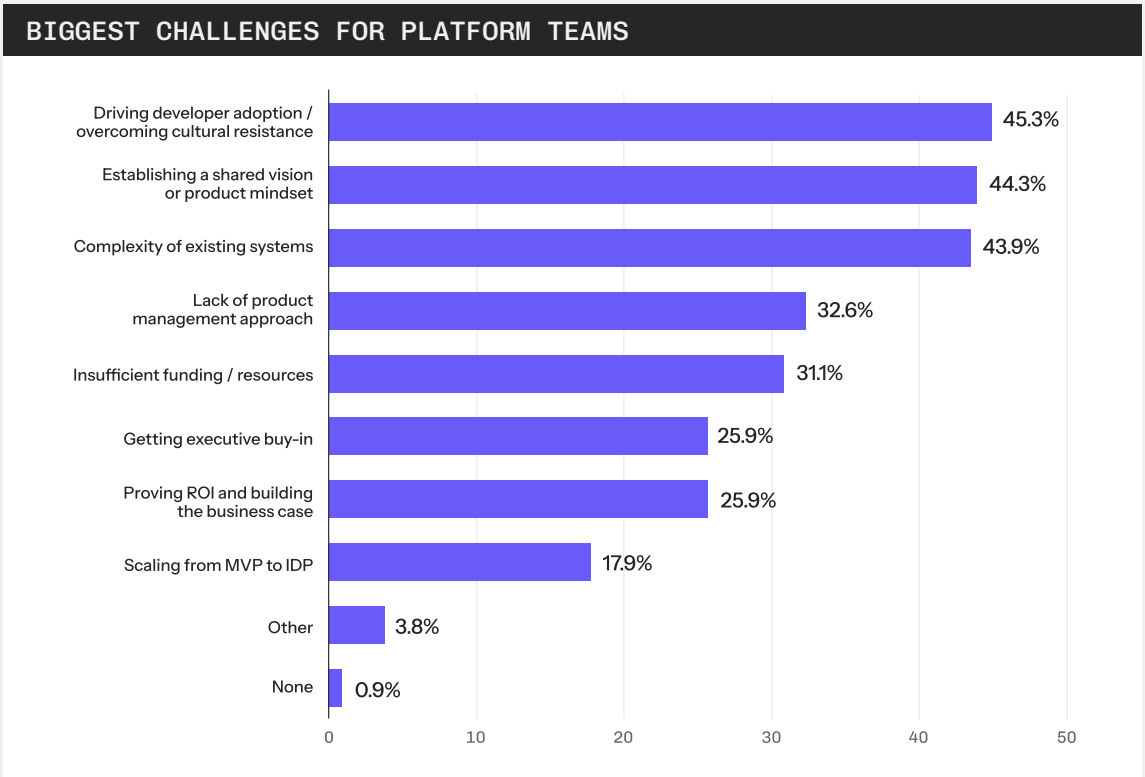
The data shows that most organizations now operate more than one platform, a notable shift from earlier industry assumptions that multiple platforms signaled fragmentation or low maturity. With 55.9% reporting more than one platform, it is increasingly clear that platform plurality often reflects intentional design rather than organizational disconnect.



As AI, data, and specialized workloads grow in importance, different teams require purpose-built platforms that address distinct needs and constraints. While a single, unified platform was once viewed as the ideal, today's landscape shows that multiple platforms can coexist as part of a cohesive strategy, provided they are well aligned and not duplicating effort.

Biggest challenges for platform teams

The data on current challenges for platform teams shows that cultural and organizational barriers continue to outweigh technical ones. Driving developer adoption remains the top challenge, closely followed by establishing a shared vision or product mindset, which reflects the ongoing complexity of shifting teams toward platform-oriented ways of working. The complexity of existing systems is another major hurdle, reinforcing how difficult it is to modernize fragmented architectures



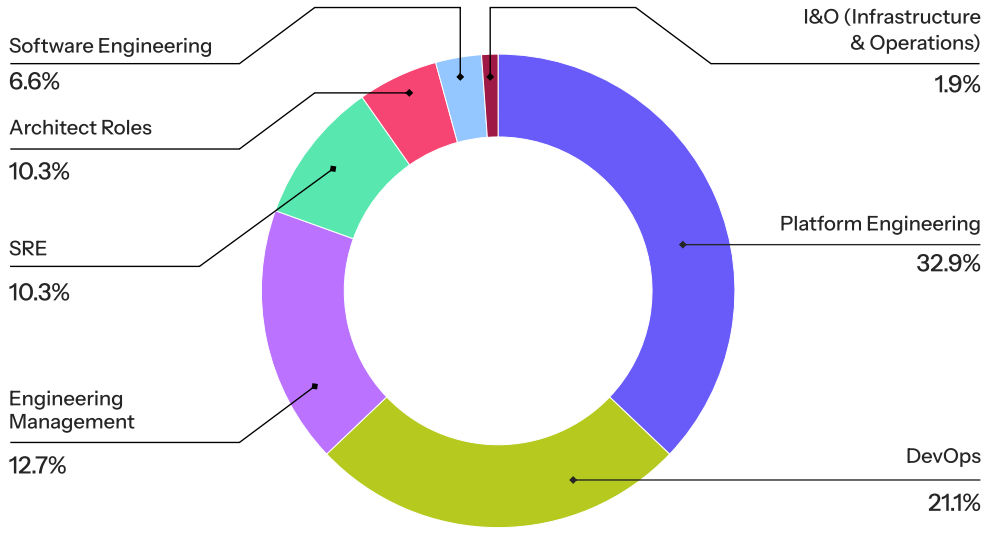
Many teams also struggle with limited product management capacity, insufficient funding, and securing executive buy-in, all of which slow progress and reduce long-term impact. Challenges like proving ROI and scaling from MVP to a full Internal Developer Platform round out the list, showing that even with growing industry maturity, platform teams still face significant structural and cultural obstacles.

The individual experience

Job titles

The distribution of job titles shows a clear shift toward platform engineering as the dominant identity for this type of work. Platform Engineering is now the largest cluster by a wide margin, while DevOps titles remain common but increasingly represent legacy naming rather than actual responsibilities.

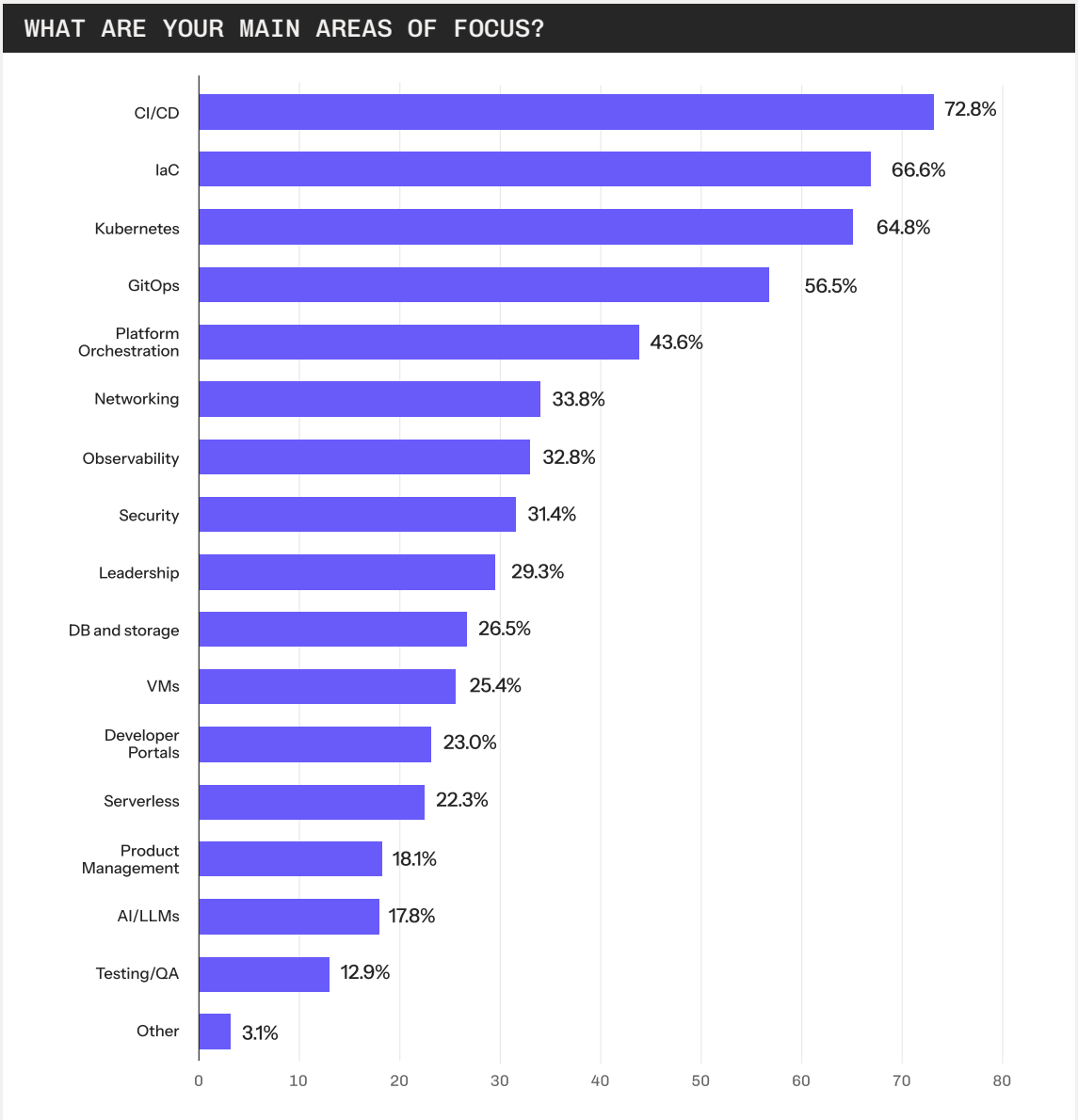
JOB TITLES (GROUPED)



Many respondents with DevOps titles indicated that their day-to-day work is platform engineering, suggesting that title changes have not yet caught up with the evolution of the role. Engineering management, SRE, and architect roles appear in smaller proportions, reflecting the adjacent backgrounds from which many platform practitioners emerge. Overall, the data highlights that while job titles are still mixed, the industry is converging on platform engineering as the accurate description of this discipline.

Primary work focus

The survey data reveals that the core focus areas for Platform Engineering teams remain heavily concentrated on foundational infrastructure and delivery mechanisms. CI/CD leads with a commanding 72.8% of mentions, followed closely by Infrastructure as Code (IaC) at 66.6% and Kubernetes at 64.8%. This triumvirate clearly indicates that the primary mandate of Platform teams is establishing automated, reliable, and standardized ways to build, provision, and deploy applications on modern containerized infrastructure.



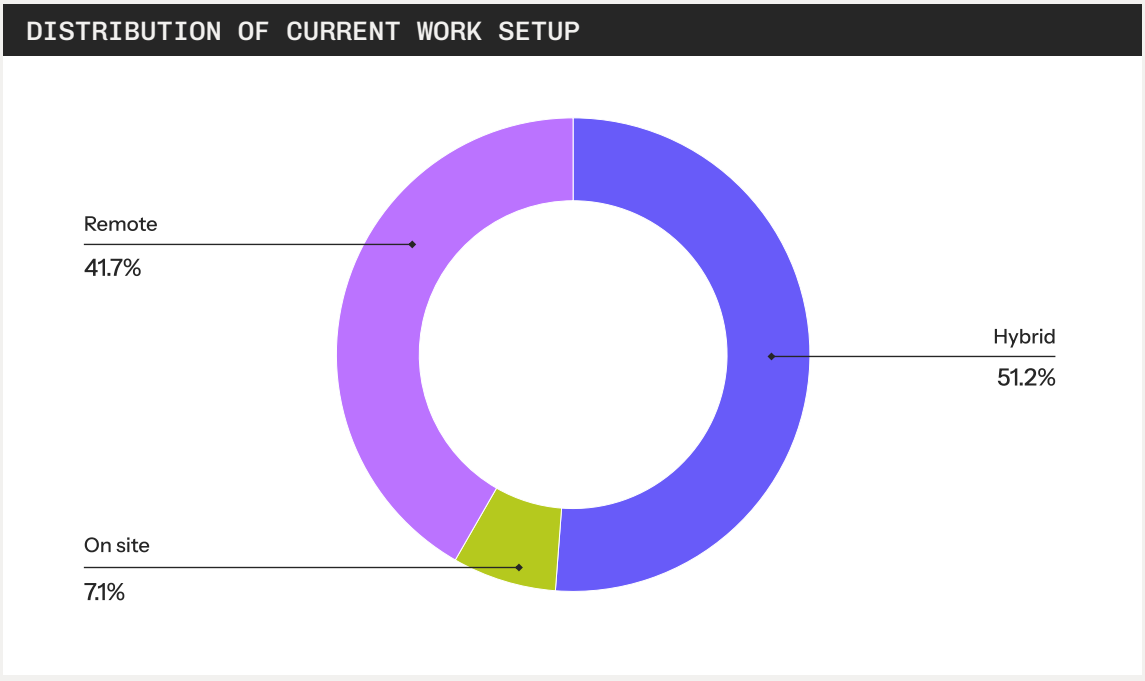
GitOps (56.5%) and Platform Orchestration (43.6%) further reinforce this trend, underscoring the shift towards declarative, system-driven application lifecycle management. While traditional Virtual Machines (VMs) still garner a significant 25.4% of attention, and Serverless is at 22.3%, the top responses confirm that the majority of platform effort is being directed toward maturing the cloud-native continuous delivery experience.

Beyond the immediate infrastructure concerns, Observability and Security stand out as critical, high-priority cross-cutting concerns, though they trail the top infrastructure items. Observability is a key focus for 32.8% of respondents, indicating a strong recognition that platforms must provide not just deployment pipelines, but also the crucial visibility required for developers to troubleshoot and monitor their applications in production. Security, mentioned by 31.4% of respondents, is similarly high on the agenda. This suggests that platform teams are increasingly enabling a "shift down" security posture, integrating tools and processes into the platform itself to ensure compliance and robust security from the earliest stages of development, rather than treating it as a final-stage gate.

Work setup

The data on work setups shows that hybrid models are now the dominant arrangement for platform teams, with just over half of respondents working in a mix of remote and on-site environments. Fully remote work remains substantial at more than forty percent, reflecting the continued suitability of distributed collaboration for engineering organizations.

DISTRIBUTION OF CURRENT WORK SETUP

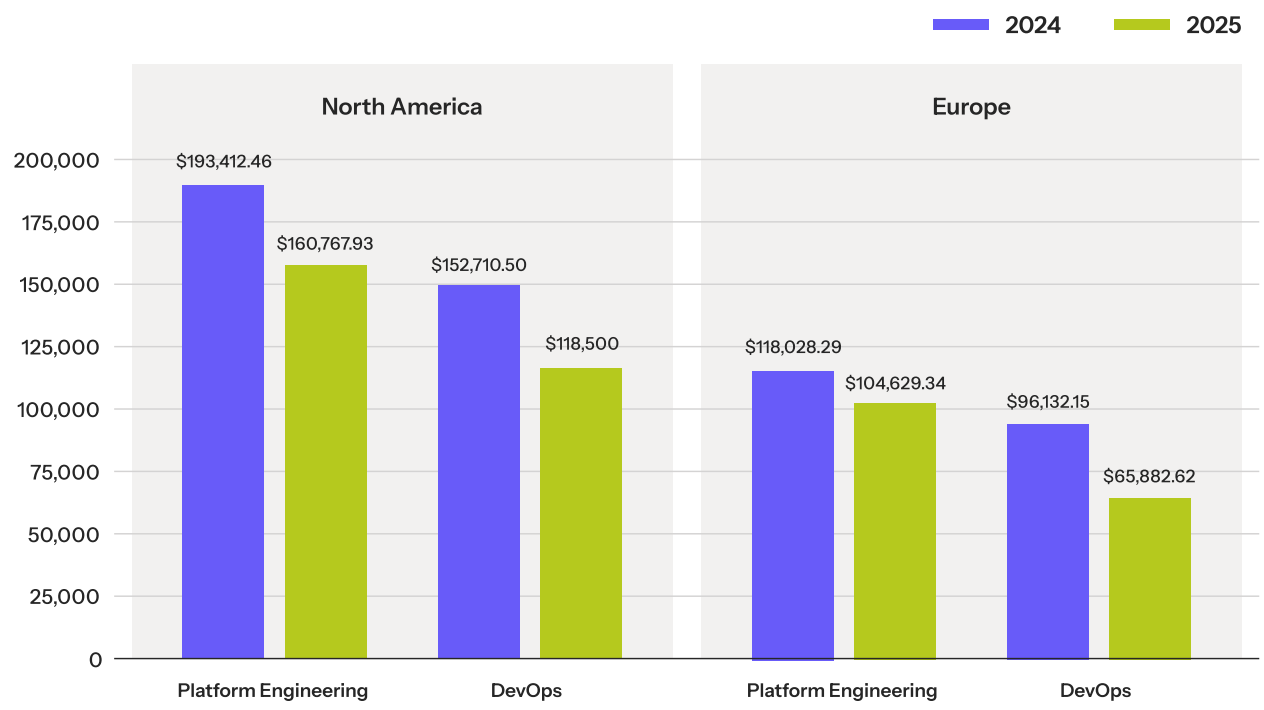


Only a small share operates fully on site, indicating that traditional office-centric models have become the exception rather than the norm. Overall, the distribution suggests that flexibility has become a standard part of platform engineering culture, enabling teams to collaborate effectively across locations while maintaining access to in-person interaction when needed.

Salary

The new salary data shows a clear decrease compared with last year, and the shift is closely tied to how the platform engineer title has evolved. In 2024, the role was dominated by highly senior engineers who had been early adopters of platform engineering practices, which pushed the average salary upward in both North America and Europe. In 2025, however, the title has become far more widespread, with many mid-level and junior engineers now classified as platform engineers as the discipline becomes mainstream.

AVERAGE SALARY

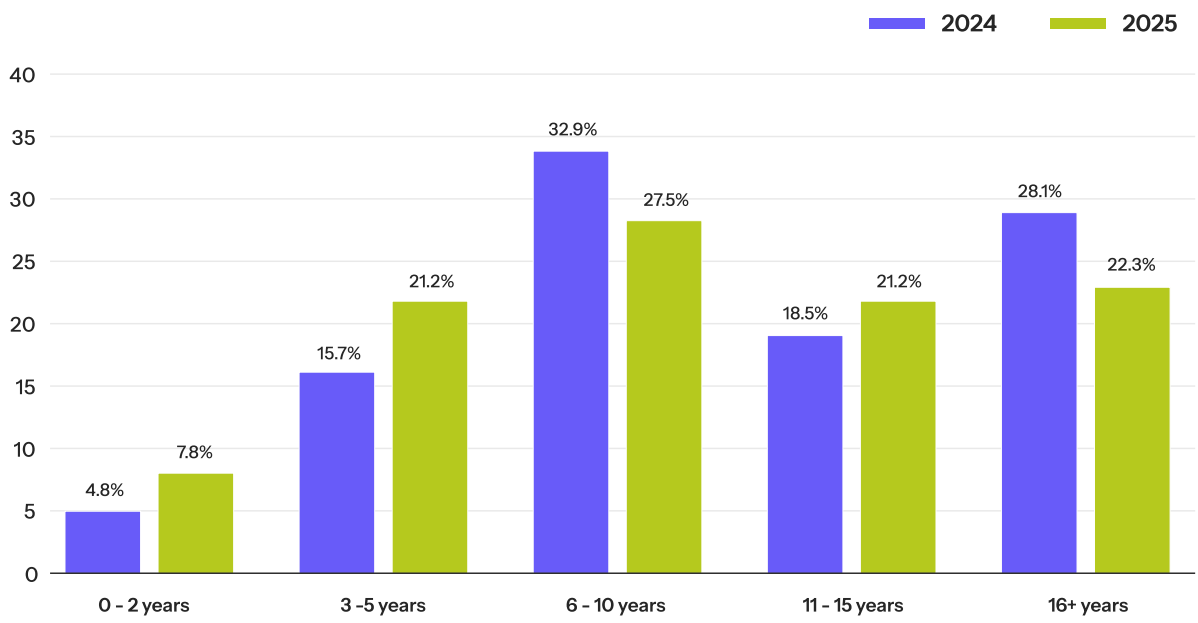


This broadening of the talent pool naturally lowers the average: North America dropped from about $193,000 to $160,000, and Europe from roughly $118,000 to $104,000. Rather than signaling reduced value, the decline reflects a maturing industry where platform engineering is no longer a niche practiced only by the most experienced specialists, but a standard function across teams with a much more diverse range of experience levels.

Working experience of platform engineers

The experience data reinforces the same trend seen in the salary numbers: platform engineering is no longer a discipline dominated by only the most senior and early-adopting practitioners. Compared with last year, the share of engineers with more than sixteen years of experience has fallen, and the proportion of those in the six-to-ten-year and three-to-five-year ranges has grown noticeably. Even the number of respondents with zero to two years of experience has increased.

AVERAGE SALARY



This shift reflects the rapid mainstreaming of platform engineering, where the role has expanded from a niche specialization held by highly seasoned engineers to a broad career path that now includes mid-level and junior talent. As more organizations formalize platform teams and adopt platform-as-a-product practices, the field is drawing in a wider range of experience levels, naturally lowering both the average tenure and average salary while signaling a maturing, scalable profession rather than an exclusive expert domain.

Main take aways *from* the results

The 2025 results show a platform engineering industry that is larger, more stable, and more mature than ever before. The growth in participation, the diversification of roles, and the widespread adoption of foundational practices all point to a discipline that has moved well beyond the hype train of the past.

One of the clearest signs of this maturity is the alignment around what platforms are for and how they should be built. This shared understanding did not exist a few years ago, when the industry was still debating what counted as "platform engineering."

Another major marker of growth is the acceptance of multiple platforms within an organization. Historically, multiple platforms were treated as evidence of low maturity or internal fragmentation. The new data tells a different story: platform pluralism is now the norm, reflecting the reality that AI, data engineering and more can often require purpose-built platforms. Instead of striving for a mythical single platform to rule them all, organizations are learning to design for interoperability and intentional separation of concerns.

This increased maturity also shows up in organizational behavior. Platform product management, once debated, is now broadly acknowledged as necessary. While not every team has a dedicated Platform Product Manager, far more organizations recognize the value of product leadership and are moving toward structured prioritization, feedback loops, and user-focused decision-making.

At the same time, the results highlight important gaps. Measurement remains the most significant. Despite improvement since last year, a large portion of teams still lack clear metrics or cannot articulate whether their metrics have improved. This signals an industry-wide challenge: even as platforms evolve, many organizations still struggle to quantify value, communicate impact, or secure the investment needed for long-term success. Platform operations and feedback collection show similar growing pains…progress is happening, but maturity remains uneven.

Finally, the workforce data reflects a major cultural shift. Salaries have decreased not because the discipline is shrinking, but because it is expanding. The title "platform engineer" is no longer held exclusively by senior experts; juniors and mid-level engineers now enter the field in large numbers. This democratization is a sign of success: platform engineering is no longer a fad or elite specialization, but a mainstream career path. However, it also means organizations must be thoughtful when interpreting titles and assessing expertise.

Taken together, the results reveal an industry in healthy expansion. Platform engineering is becoming standard practice but full maturity still lies on the road ahead.

Five key recommendations *for* platform teams

Across thousands of hours of platform engineering webinars, roundtables, discussions, and PlatformCon talks, and hundreds of published articles and research, the 5 core fundamental lessons for platform engineers going into 2026 will be very familiar to many readers. They however remain the key differentiator between successful and disastrous platform engineering initiatives.

FIVE KEY RECOMMENDATIONS FOR PLATFORM TEAMS

01 — Adopt the Platform as a Product mindset

Treat the platform as a continuous product, not a one-off project, by focusing on the holistic developer experience and iterating based on real user feedback. At the same time, ensure the platform's evolution is explicitly tied to strategic business goals, and translate technical improvements into clear, measurable business outcomes to secure lasting executive sponsorship and demonstrate value beyond technical correctness. The most successful enterprises drive this by having a dedicated Platform Product Manager in their platform teams, rather than relying only on your teams adoption of a Product mindset.

02 — Start small with the Minimum Viable Platform (MVP)

Use an iterative approach (ideally completed in weeks, not months) to de-risk the initiative and demonstrate measurable value quickly to key stakeholders. Define focused golden paths that cover 80% of common needs to standardize and automate fast.

03 — Prioritize culture and empathy for adoption

Understand that platform engineering success is fundamentally a cultural reframing. Success hinges on fostering psychological safety, transparency, and trust, and actively listening to the platforms customers to understand what hurts, rather than immediately prescribing technological fixes.

04 — Upskill yourself, your team, and your org

Platform engineering is a rapidly growing discipline now incorporate increasingly more domains, and requirements with AI. It is crucial that you are focused heavily on your own learning, the learning of your team, and the general knowledge of your organization, otherwise you will quickly fall behind, waste time and resources with mistakes, or simply fail to achieve the results you hope for.

05 — Master the dual relationship of AI and platform engineering

Recognize that a quality internal platform is a key enabler for magnifying the positive effects of AI on organizational performance. Shift the platform focus from managing granular complexity to providing intelligent orchestration for AI agents and ensure that AI proficiency is mandatory for platform teams.

Predictions *for* the future

Platform engineering is no longer an emerging discipline, it is becoming the operating model of the modern enterprise. As shown throughout this report, the industry has crossed a threshold. What began as an effort to improve developer productivity and right the wrongs of DevOps has expanded into a sociotechnical transformation touching every domain: AI, security, observability, FinOps, culture, and organizational design itself. The predictions outlined in this final section paint a future that is more interconnected, more automated, and more strategically critical than ever before.

The coming years will not simply demand better tooling or more mature architectures, they will demand new mental models. Platform engineering will become the backbone of AI-native enterprises, the arbiter of cost efficiency, the owner of security and compliance automation, and the driving force behind a new generation of developer experience standards. The shift to specialized roles, self-evolving platforms, and embedded intelligence marks a decisive departure from the artisanal era of software development. In its place emerges a systematized, data-driven, AI-augmented discipline capable of supporting entire ecosystems rather than individual teams.

Yet the greatest challenge ahead is not technical. It is cultural. As organizations adopt platform engineering at scale, the gap between early adopters and late movers will widen dramatically. Maturity, readiness, reskilling, and organizational debt will differentiate those capable of thriving in the AI-native era from those who find themselves constrained by outdated processes and brittle operating models. Platform engineering succeeds where culture supports it, and it fails where it does not.

The next phase of platform engineering will be defined by teams that embrace this new reality. Teams that embed FinOps into every workflow. Teams that treat observability as a default, not a feature. Teams that build golden paths not just for code, but for compliance, cost, reliability, and AI. Teams that understand that platform engineering is no longer merely an efficiency function, but a strategic differentiator.

As we look toward 2026, one conclusion is clear: platform engineering as crucial foundation of successful AI has become the most important force shaping how enterprises design, build, secure, and operate software. Those who invest now in people, in culture, in architecture, and in AI-native foundations will define the next decade of innovation. Those who hesitate will inherit a level of organizational debt that becomes exponentially more costly to unwind.

The future belongs to the organizations that treat platform engineering not as a project, but as a long-term capability. And it belongs to the teams who see what is coming and prepare accordingly.

The transformation is already underway. The question now is who transforms with it and who is left behind.

Appendix

Survey methodology snapshots

State of Platform Engineering Survey

- **Participants:** 518 professionals globally
- **Roles:** Platform engineers, DevOps, SREs, Architects, Consultants, and technical leaders
- **Timing:** Data collected August to October 2025
- **Scope:** 30 questions covering AI usage, challenges, organizational attitudes, and future expectations
- **Purpose:** Understanding current state and future trajectory of the platform engineering discipline and industry

References

DORA Research. State of AI-assisted Software Development Report 2025.

Luca Galante. AI and Platform Engineering. Platform Engineering Blog.

Luca Galante. New reference architectures for IDPs on AWS, GCP, and Azure: Version 2.0 now available.

Kaspar von Grünberg. Why Platform Engineering Will Eat the World. Platform Engineering Blog.

Weave Intelligence. State of AI in Platform Engineering 2025

Weave Intelligence. Reference architecture for an AI/ML Internal Developer Platform on GCP.

Authors

This report was written by Sam Barlien, with contributions from Luca Galante.

Sam Barlien

Sam Barlien is the Head of Ecosystem for the Platform Engineering Community. He is a tech nerd, and has been involved in tech communities for more than 10 years. He helps manage Platform Weekly, co-hosts PlatformCon, and drives the community Ambassador program, blog and Youtube channel. He speaks to 100s of platform engineers a year and translates their experience into articles, webinars and reports for the wider community.

Luca Galante

Luca Galante is the Core Contributor to the Platform Engineering community, the world's largest platform engineering community with over 270,000 members. He routinely speaks to dozens of engineering teams every month, and summarizes his learnings and takeaways from hundreds of setups into crisp, insightful content for everyone in the industry, from beginner-Ops to cloud experts. He is the host of PlatformCon, the world's largest platform engineering event, and writes to over 100,000 engineers every Friday in his newsletter, Platform Weekly.