THE STATE OF AGENTIC AI REPORT

docker

EXECUTIVE SUMMARY

Key Takeaways

- **Rapid adoption, early maturity:** 60% of organizations already have AI agents in production[1], and 94% view building agents as a strategic priority, but most deployments remain internal and focused on productivity and operational efficiency.
- **Security and complexity are the top barriers:** 40% of respondents cite security as the #1 challenge in scaling agentic AI, with 45% struggling to ensure tools are secure and enterprise-ready. Technical complexity compounds the challenge. One in three organizations (33%) report orchestration difficulties as multi-model and multi-cloud environments proliferate (79% of organizations run agents across two or more environments).
- **MCP shows promise but isn't enterprise-ready:** 85% of teams are familiar with the Model Context Protocol, yet most report significant security, configuration, and manageability issues that prevent production-scale deployment.
- **Containerization remains foundational:** 94% use containers for agent development or production, and 98% follow the same cloud-native workflows as traditional software, establishing containers as the proven substrate for agentic AI infrastructure.
- **Long-term outlook:** Rather than a "year of the agents," the data points to a decade-long transformation. Organizations are laying the governance and trust foundations now for scalable, enterprise-grade agent ecosystems.

[1] Note: Adoption levels reflected in this research may appear high relative to the broader market. This survey intentionally sampled technical practitioners (developers, DevOps engineers, and technical leaders) across all industries, i.e.,the professionals most likely to be working with emerging technologies. These findings reflect the leading edge of enterprise adoption rather than the broader market average.

AI agent adoption is advancing quickly, but the ecosystem is still immature. Organizations across industries view AI agents[2] as a strategic priority–94% overall, with nearly half (42%) describing building agents as a "very high priority," signaling the depth of commitment beyond surface-level interest. Significantly, adoption today is focused on internal workflows that boost team productivity while minimizing business risk.

This early inward-oriented focus obscures the true degree of adoption, which has gone beyond mere experimentation (fully 60% of organizations say they already have AI agents in production). The significant increase in ink spent on contrarian, counter-hype articles decrying the shortcomings of AI, as recently described both by MIT and Harvard Business Review, misses the real story happening beneath the surface. While some voices are calling 2025 "the year of the agents," a growing number of researchers and industry leaders are urging a longer view–one that frames this as the decade of agents, not the year. AI researcher Andrej Karpathy has suggested that true agentic AI remains years away from matching the industry's hype[3]. **What's happening now is the crucial groundwork: organizations are building the infrastructure, governance, and trust required to sustain the next wave of intelligent, autonomous systems.** Agent adoption is advancing quickly, but the ecosystem is still young and laying foundations for a multi-year transformation rather than a momentary boom.

This inside-out approach mimics past technological adoption cycles, including how companies first built internal automation and orchestration systems before monetizing them externally, or how cloud infrastructure (IaaS / private clouds) was internalized before platforms were exposed to customers. Organizations built confidence, governance, and resiliency through operational use cases before extending into external or revenue-generating domains.

Interest is strong, but scaling agents even internally presents persistent challenges. Security remains the top barrier regardless of where an organization is on its AI adoption journey. Other major barriers include complexity in orchestration, fragile build processes, and distribution challenges—sharing practices remain fragmented, and no widely accepted standards for packaging or deployment have taken root. Fear of vendor lock-in compounds these risks, as enterprises worry about dependencies in core agent and agentic infrastructure layers such as model hosting, LLM providers, and even cloud platforms.

At the same time, clear patterns of adoption are taking shape. Containers and cloud native workflows have become foundational to agent development. Hybrid and multi-cloud deployments are now the default. These are not unfamiliar patterns. Indeed, an overwhelming majority (98%) of organizations are using the same approach to development and deployment for AI apps and agents as they do for more traditional cloud native apps. Tools from Google, AWS, Docker, and other cloud native providers top the list of critical infrastructure vendors and are already embedded in many agent stacks.

What emerges is a landscape marked both by promise and the need for flexible solutions. Organizations are already seeing measurable productivity gains, particularly in DevOps, security automation, and process automation. But realizing the full potential of agentic AI will take time. The coming years will be less about chasing hype cycles and more about building the trust layer, infrastructure, and interoperability standards needed for agents to operate safely and at scale. Rather than a "year of agents," what's unfolding is the decade of agents, or a long-term transformation in which today's early, inward-facing deployments lay the groundwork for tomorrow's interconnected ecosystems. The organizations that treat this as a sustained evolution, not a short-term sprint, will be best positioned to shape and benefit from the agentic future.

---

[2] Note on terminology. Throughout the document, we will often refer simply to "agents" and "agentic workflows." Unless specifically indicated, we are in all instances referring to "AI agents" and "agentic AI workflows."

[3] https://the-decoder.com/ai-researcher-andrej-karpathy-says-agentic-ai-is-years-away-from-matching-industry-hype/

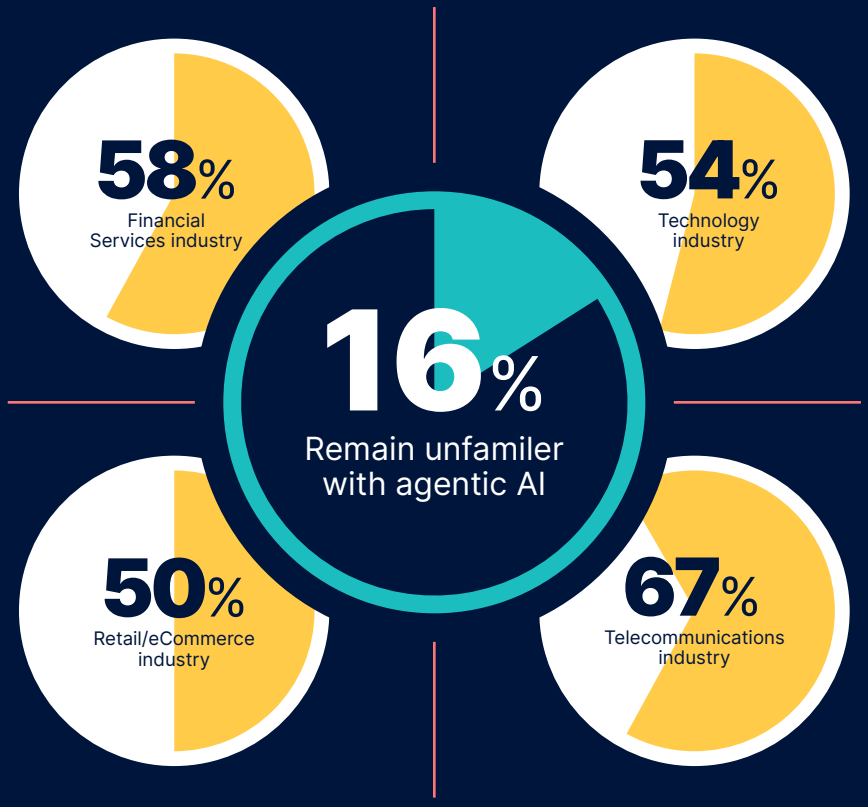
Section 1

State of Agentic AI Adoption: *FROM EXPERIMENTATION TO OPERATIONAL MATURITY*

AI agent adoption has moved beyond experimentation into early operational maturity. Adoption rates are especially high in financial services (58%), technology (54%), retail / eCommerce (50%), and telecommunications (67%) industries, each saying they have at least one or more agents in production. Across the globe, organizations are investing heavily in agents, especially for internal, productivity-focused use cases. Fully 60% of organizations already report having AI agents in production, though a third of those remain in early stages. At the same time, 16% of organizations are still unfamiliar with the term “agentic AI,” underscoring how new but fast-maturing this field is.

WHO'S USING **AGENTIC AI?**

The Great Divide: While adoption is strong among early movers, 16% of organizations remain unfamiliar with the term "agentic AI," highlighting the gap between leading-edge adopters and the broader market.



What's clear is that agent adoption today is driven by a pragmatic focus on productivity, efficiency, and operational transformation, not revenue growth or cost reduction. Most organizations are deploying agents internally to optimize workflows and augment teams, particularly within software, infrastructure, and operations functions where feedback loops are fast and risk is controlled. Building agents has become a strategic priority for 95% of respondents.

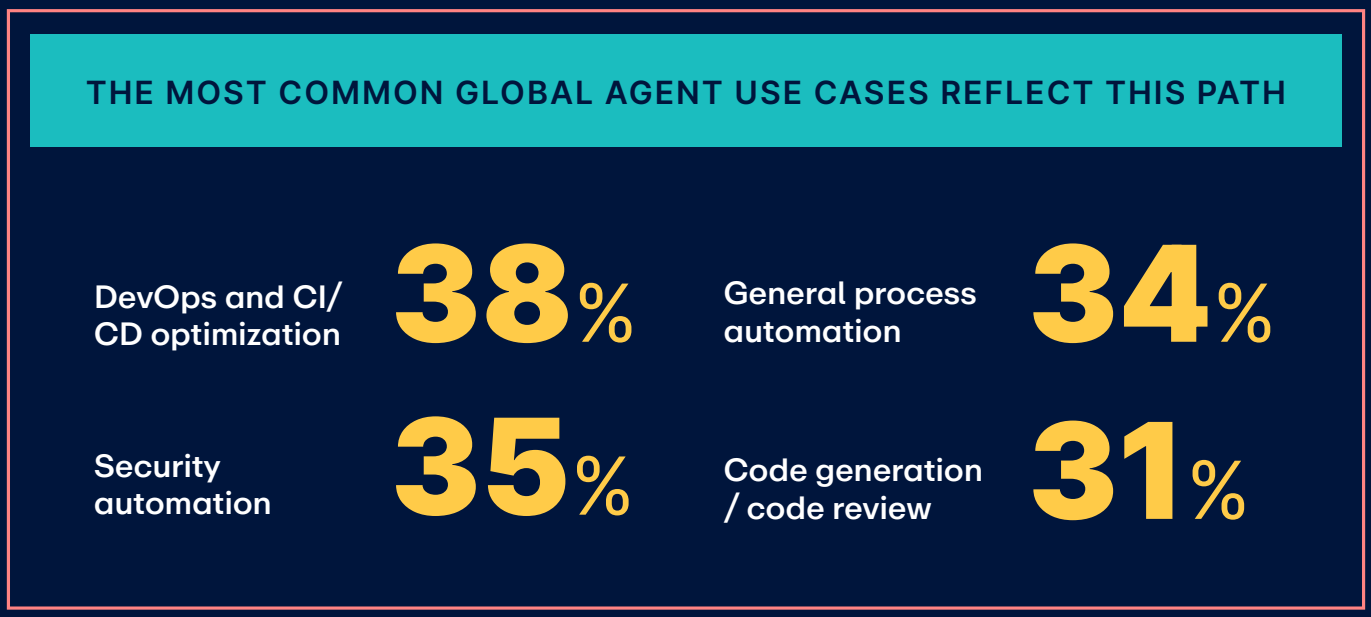
This focus on internal, operational domains is not new; it mirrors the early adoption patterns of nearly every major enterprise-tech transformation. As with cloud infrastructure, where private cloud preceded hybrid and public cloud expansion, or internal automation platforms, which came before customer-facing APIs and services, organizations often begin by solving their own problems first.

Internal deployment creates a safer space for experimentation and gives teams time to build trust, governance, and familiarity with new patterns before scaling to customer-facing, innovative, or high-risk domains.

The most common agent use cases reflect this path:

- DevOps and CI/CD optimization (38%)
- Security automation (35%)
- General process automation (34%)
- Code generation / code review (31%)

These are the same domains where early platform engineering and cloud-native adoption first gained traction, and it's no surprise that developers and infrastructure teams are once again leading the curve.



Era of Productivity Agents
Top Use Cases (Global + Regional)

REGION	UNITED STATES	UNITED KINGDOM	GERMANY	JAPAN	SINGAPORE
DevOps and CI/CD optimization	36%	29%	49%	32%	41%
Security automation	31%	35%	31%	31%	32%
General process automation	35%	35%	27%	33%	41%
Code generation / code review	31%	30%	27%	35%	32%

These early deployments are largely built on top of existing cloud-native foundations, extending the same workflows and operational practices already in use across DevOps and infrastructure teams.

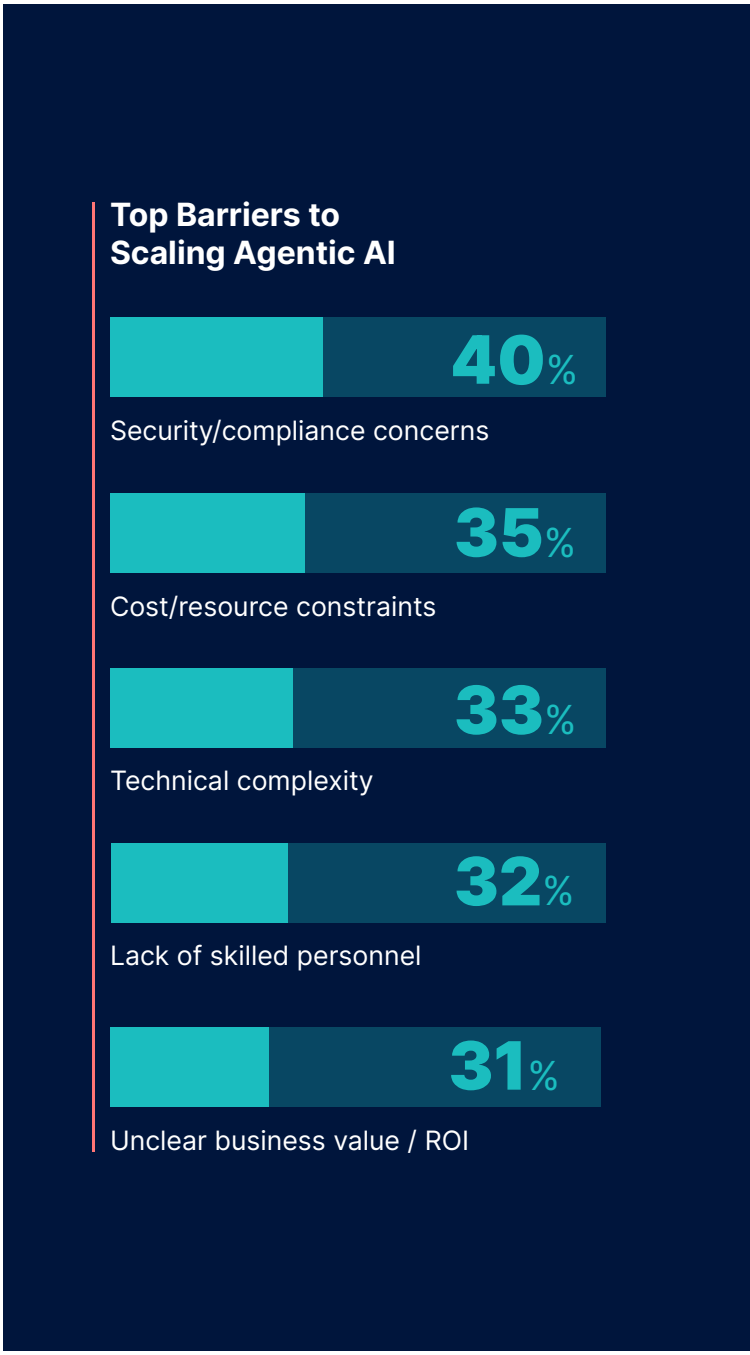
Still, the state of adoption today remains primarily inward-facing. Organizations are laying the groundwork, experimenting safely within operational domains, refining governance, and learning what reliable agent behavior looks like in production. These early experiences are shaping the next phase, when agents move from isolated productivity boosters to interconnected systems that drive business transformation.

The takeaway is a nuanced one. AI-agent adoption has entered a formative but fast-maturing phase. Organizations are embracing agents as a practical tool for improving productivity and operational efficiency, building confidence and capability before expanding to external or customer-facing applications. This growing foundation of real-world use sets the stage for how enterprises will address the next challenge of scaling agents securely and reliably at enterprise scale.

Section 2

The Roadblocks to Scale ***SECURITY, COMPLEXITY, AND ENTERPRISE READINESS***

As organizations move from experimenting with AI agents to scaling them in production, the challenges shift from feasibility to enterprise reality. The barriers are no longer conceptual; they define how far and how fast agentic systems can grow. Across the global sample, two obstacles dominate this next phase: **security and technical complexity**, each amplified by the growing diversity of models, tools, and deployment environments.



Security: The Persistent Gatekeeper

Security remains the top blocker for scaling agents. Across industries and maturity levels, organizations cite security and lack of enterprise readiness as the most significant limitations in today's agent tooling. In many ways, agent development is encountering the same challenges cloud-native technologies faced in their early days, namely rapid innovation outpacing the maturity of secure, enterprise-grade tooling. Without hardened security and governance frameworks, adoption risks stalling at the pilot phase, just as early cloud-native efforts did.

THE SECURITY TRUST GAP

Security and trust create a dual barrier to production deployment:

45%

struggle to ensure agentic tools are secure, trusted, and enterprise-ready

Financial Services faces the steepest hurdles:

52%

cite tool security as a major challenge

52%

struggle to identify which tools merit trust

The sector's strict compliance and risk standards make vetting new tools particularly complex.

Security concerns arise across every layer of deployment, with **40% of respondents citing it as their top blocker** when building agents. It isn't confined to any one layer of the stack, but **arises across every phase of deployment**, from infrastructure to governance to operations. And the **challenges compound** as teams scale from pilots to production.

INFRASTRUCTURE

As organizations expand agent deployments, teams emphasize the need for secure sandboxing and runtime isolation, even for internal agents.

OPERATIONS

Complexity and orchestration sprawl are introducing new security exposures. In our data, over a third of respondents cite challenges coordinating multiple tools, and a comparable share report that integration itself introduces security or compliance risks, which are clear indicators of operational fragility as agents move from pilots to production.

GOVERNANCE

There is a strong demand for clear guardrails, policy enforcement, and auditability at enterprise scale to ensure consistency and trust across distributed agent workflows. With current tooling, the biggest challenges are ensuring those tools are secure, trusted, and enterprise-ready, which is the top concern for 45% of organizations.

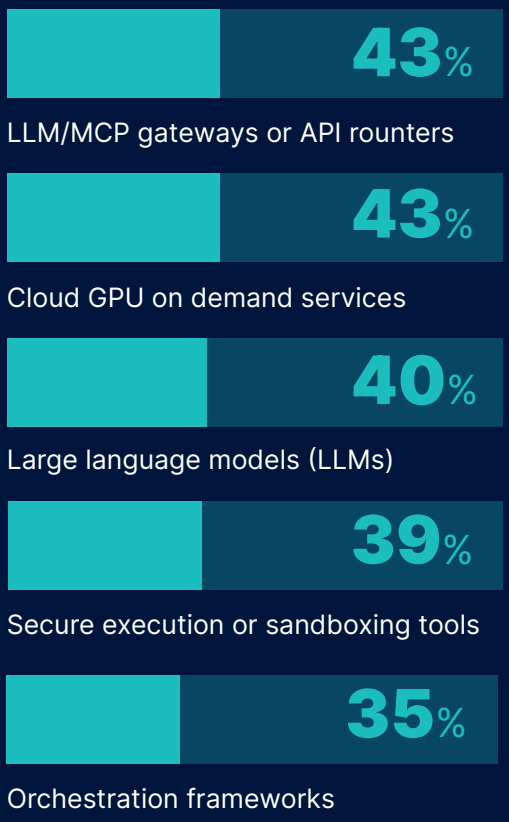
Together, these layers form a single story: security is not just one barrier among many. It is the defining constraint shaping how far and how fast enterprises can scale agentic AI.

Technical Complexity: The Expanding Challenge

While security defines whether agents can scale, technical complexity determines how easily they do. One in three organizations (33%) cite technical complexity as a top barrier, encompassing orchestration, model diversity, and infrastructure fragmentation.

Teams are working with foundational primitives, such as model endpoints, GPUs, and orchestration scripts, rather than integrated, production-grade platforms. They are looking for reliable infrastructure and centralized control, such as unified gateways and GPU-on-demand services. Yet orchestration consistently ranks as the hardest part of the agent lifecycle.

Top Tools for Building Agents



THE COMPLEXITY PARADOX

Even the most technically sophisticated industries struggle with complexity:

40% Technology industry

41% Retail/eCommerce

The irony? Aggressive adoption and intricate integration requirements create friction even for teams best equipped to handle it.

Complexity manifests in several dimensions:

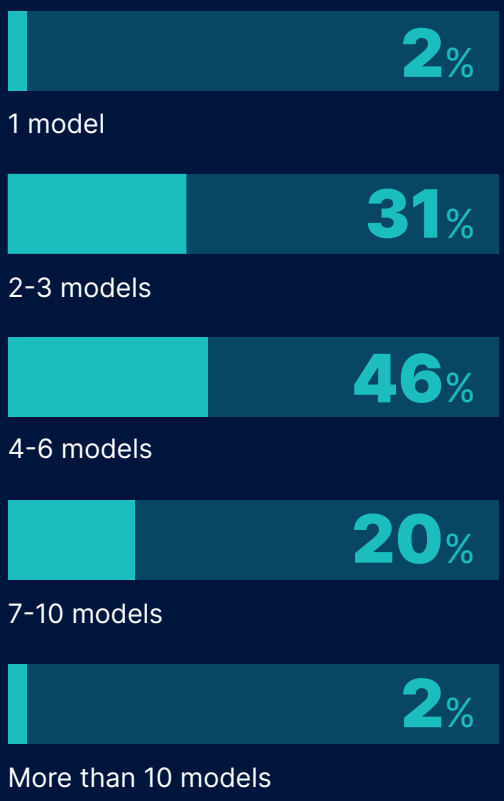
1. Multi-model ecosystems: Nearly two-thirds of organizations (61%) combine cloud-hosted and local models. This strategy increases integration effort and performance tuning complexity, but it is also intentional. Complexity grows further within the model layer itself: 46% of organizations report using between four and six models within their agents, while only 2% rely on a single model. Enterprises are adopting multi-model and multi-cloud architectures to give teams greater control over performance, customization, privacy, and compliance, reflecting the practical, use-case-driven nature of today's agentic ecosystems.

2. Hybrid and multi-cloud deployments: To maintain flexibility, 79% of respondents now operate agents across two or more environments–51% in public clouds, 40% on-premises, and 32% on serverless platforms. This approach enables greater control over performance, privacy, and compliance but also multiplies orchestration and governance demands, adding to the overall complexity of agent operations.

3. Orchestration and workflow management: Coordinating multiple models, tools, and frameworks is consistently identified as one of the hardest aspects of building agents. Ensuring reliability across heterogeneous systems requires new orchestration patterns, observability layers, and runtime policies.

4. Lack of standardization: With no universal framework for packaging or sharing agents, teams are forced to create custom processes, increasing maintenance costs and slowing deployment.

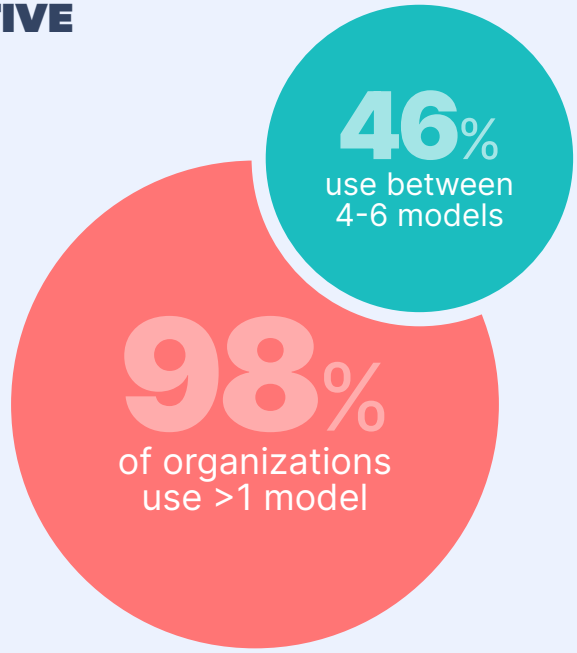
Number Of Models Actively Used Within Agent



Primary Reasons For Running Models Locally

CONTEXT ENGINEERING: **THE MULTI-MODEL IMPERATIVE**

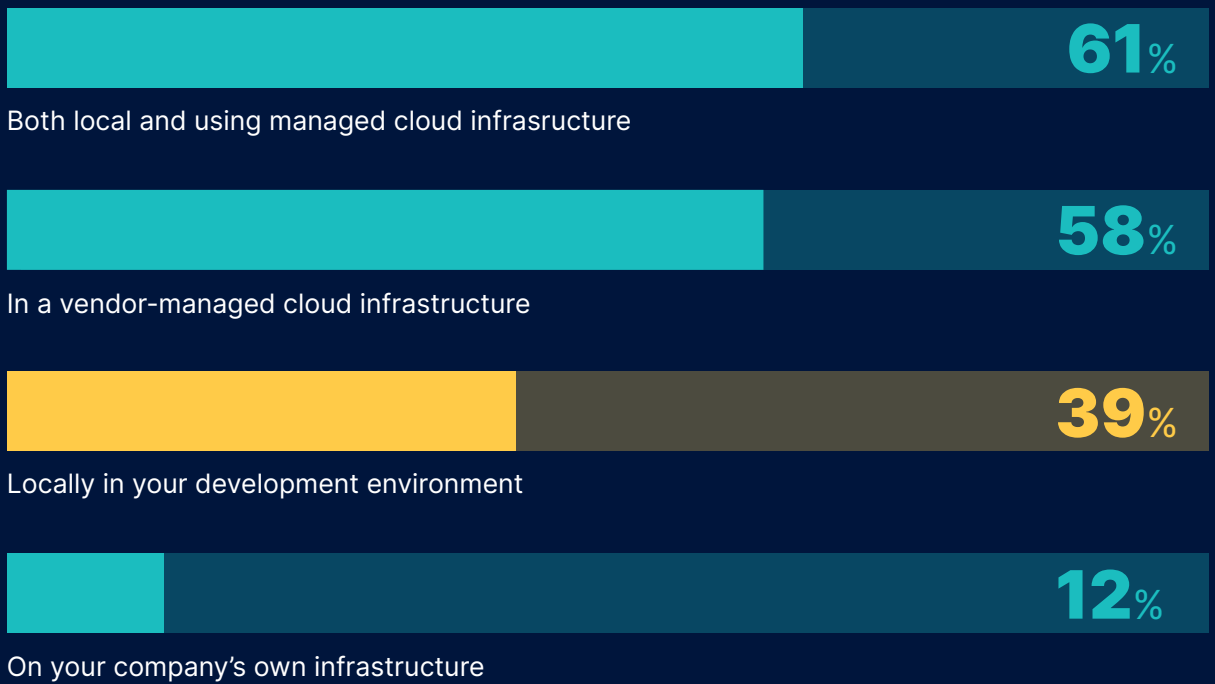
Agents are multi-model by design:



The reason? Effective agent development is about context engineering. This means giving the right prompt, with the right tools and data, to the right LLM.

Frontier cloud models offer cutting-edge capabilities when you need them. Open and local models deliver advantages when cost, latency, or privacy constraints matter most. Leading organizations build flexible architectures that strategically mix and match rather than lock into a single approach.

Where AI Agent Models Are Executed

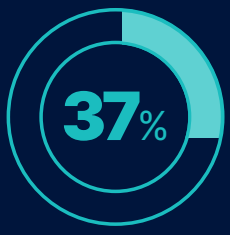


THE ORCHESTRATION CHALLENGE

Operational complexity from orchestrating multiple components is the #1 challenge in building agents (48%). The problem has three dimensions:

1

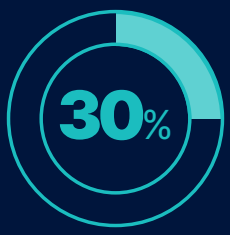
IMMATURE TOOLING



report orchestration frameworks are too brittle or immature for production use

2

TESTING & VISIBILITY GAPS



struggle to test, debug, and monitor complex orchestrations

3

INTEGRATION COMPLEXITY

Coordinating multiple

models, tools, & frameworks

across heterogeneous systems

The need: Standard orchestration layers that provide production-grade reliability, built-in observability, and simplified integration patterns, which is the missing infrastructure for scaling agents beyond pilots.

Coordinating across this expanding ecosystem is already stretching existing workflows beyond their limits. Nearly half of global respondents (48%) cite operational complexity in coordinating multiple components as their top challenge, while 43% point to increased security exposure stemming from orchestration sprawl. In certain markets, the burden is even heavier: 65% of organizations in India, 55% in Germany, and 53% in Singapore identify orchestration as the most acute pain point in their agent development pipeline. **Yet orchestration is only one facet of a broader problem: integrating diverse models, tools, frameworks, and deployment environments adds further layers of difficulty that test even the most mature cloud-native operations.**

What's missing is a standard orchestration layer that can abstract complexity, ensure interoperability, and simplify coordination between agent components. Still, a few solutions are beginning to gain traction. For example, among teams already building agents with Docker, 40% are using Compose as their orchestration layer, signaling early momentum toward more standardized, container-based approaches to coordination.

In short, the very strategies enterprises use to increase flexibility, like multi-model, multi-cloud, multi-runtime architectures, also intensify operational complexity. Flexibility comes at the cost of coordination.

But beneath that complexity lies an even more fundamental question: who governs it all?

From Complexity to Governance: The Price of Freedom

As organizations diversify across models, tools, and deployment environments to gain flexibility, that same complexity is giving rise to a new kind of challenge: governance. What began as a pursuit of freedom from single-vendor dependencies has evolved into a broader need for control, consistency, and accountability across an expanding ecosystem. Each additional model, orchestration tool, or cloud platform adds autonomy but also enlarges the surface area. The “price of freedom” is greater coordination overhead and heightened security exposure, making governance not just a policy question but an architectural one.

This fragmentation reflects the ecosystem’s immaturity and the difficult balancing act organizations face. Teams must experiment with new capabilities while simultaneously meeting regulatory requirements, addressing specific use-case demands, and maintaining operational standards. The result: organizations combine multiple models, orchestration tools, and cloud environments not by choice, but out of necessity. They are stitching together immature building blocks while navigating compliance constraints. Each component added expands autonomy and capability but multiplies the coordination burden. The operational complexity isn’t a side effect; it’s the price of progress in an ecosystem still finding its structure.

Across the survey, enterprise readiness and security consistently emerge as the top blockers to adoption–symptoms of this same fragmentation. The more components, frameworks, and models organizations assemble, the harder it becomes to secure and standardize them. Respondents draw a widening

FROM FREEDOM TO TECHNICAL DEBT: **WHY GOVERNANCE CAN'T WAIT**

LEADING ORGS EMBED GOVERNANCE DAY ONE:

STANDARDIZED ORCHESTRATION POLICIES

CONTROLLED RUNTIME

SECURE-BY-DEFAULT TOOLCHAINS

The implication is clear: without consistent guardrails and interoperability, today's creative freedom becomes tomorrow's technical debt.

They've redefined "enterprise readiness," not as a final certification step, but as an architectural principle that guides every decision.

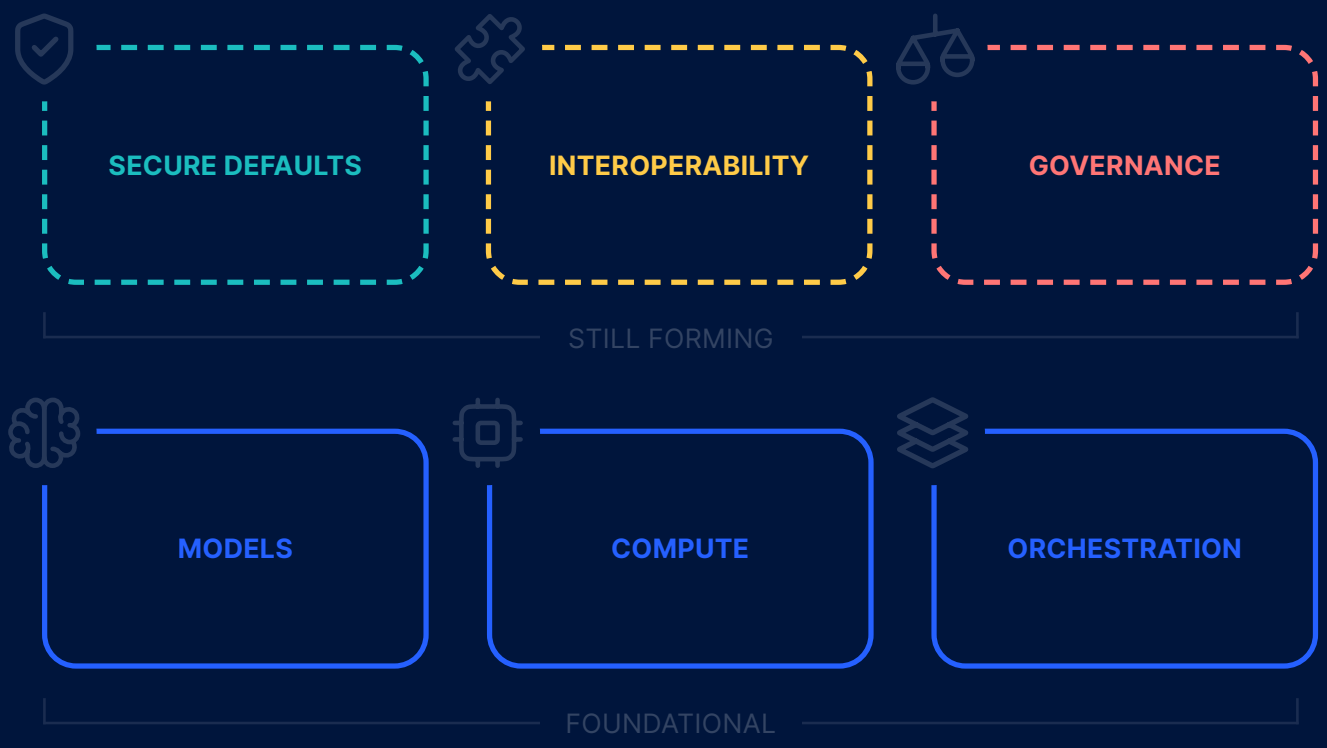
gap between experimentation and safe, repeatable deployment. The implication is clear: governance isn't just a compliance function; it's what transforms experimentation into enterprise reliability. Without consistent guardrails and interoperability, today's creative freedom risks solidifying into tomorrow's technical debt and risk.

Leading teams are closing this gap by building governance and interoperability directly into their architectures, embedding standardized orchestration policies, controlled runtimes, and secure-by-default toolchains. These teams are redefining "enterprise readiness," not as a certification step at the end of deployment, but as an architectural principle from the start. In this sense, the next phase of maturity isn't just about building agents that work; it's about building ecosystems that behave.

Bridging the Gap

Together, security and complexity define the next phase of agentic maturity. Enterprises have the foundational primitives, like models, compute, and orchestration, but the connective tissue of secure defaults, interoperability, and enterprise governance is still forming. The following section examines this through the lens of Model Context Protocol (MCP), which sits at the intersection of these themes: a promising foundation for interoperability that must evolve rapidly to meet enterprise security and scalability requirements.

Enterprise AI Integration Gap



Section 3

Model Context Protocol (MCP): *A PROMISING FOUNDATION THAT'S NOT YET SECURE AND ENTERPRISE-READY*

Model Context Protocol (MCP) is emerging as the de facto standard for connecting agents to external tools and data sources, essentially forming the backbone of modern agent ecosystems. MCP enables agents to communicate across diverse environments, invoking multiple tools, querying knowledge bases, and connecting to enterprise systems. By providing a common protocol, it aims to reduce the fragmentation and ad-hoc integrations that currently create complexity and boilerplate.

Adoption of MCP is high in principle among survey participants who are further along in their agent journey. Eighty-five percent of global respondents say they're familiar with MCP, and two-thirds say they actively use it across both personal and professional projects. But implementation remains fragile, particularly in enterprise environments. While the MCP shows early promise, MCP security evaluation remains shallow. This suggests that most teams are operating in what could be described as "leap-of-faith mode" when it comes to MCP–adopting the protocol without security guarantees and operational controls they would demand from mature enterprise infrastructure. As with early container and microservice adoption, gaining real enterprise trust will require secure-by-default deployment patterns, including trusted content and components, verified runtime behavior, and integrated governance controls.

The infrastructure burden is already showing. Among all organizations using MCPs, 42%

THE MCP ADOPTION PARADOX

Despite widespread familiarity (85%), MCP adoption remains hindered by fundamental enterprise readiness gaps.

1. **42% OPERATIONAL OVERHEAD**
2. **41% SECURITY *or* COMPLIANCE CONCERNS**
3. **41% INSTALLING & CONFIGURING ISSUES**

Security stands out as the primary barrier.

Critical security barriers include:

1. **46% VULNERABILITY DETECTION**
   Detecting and mitigating security vulnerabilities including indirect prompt injection, tool poisoning, and rug pull attacks
2. **40% ACCESS CONTROL & CREDENTIALS**
   Managing access controls, credentials, and authentication without standardized approaches
3. **36% ISOLATION**
   Isolating MCP servers from host systems and other workloads

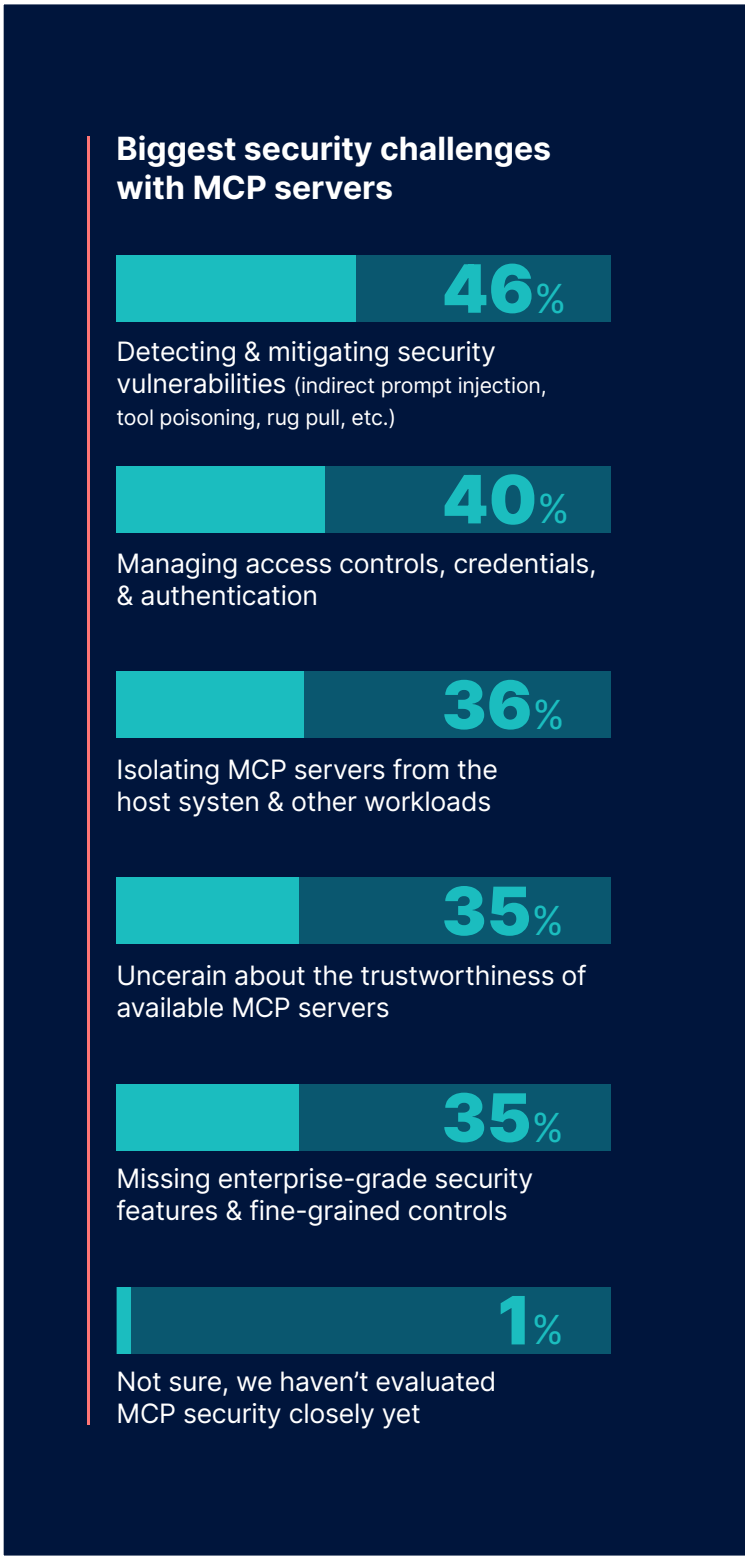
The paradox: teams recognize MCP's potential but can't justify production use without the security infrastructure that would make it enterprise-grade.

cite operational overhead in managing servers and clients, while 41% report difficulty with installation and configuration, and an equal 41% raise concerns about security and compliance. These challenges become even more pronounced for teams earlier in their agentic journey: 46% of those in the early stage worry about security and compliance—a clear signal that usability, maturity, and trust gaps are still holding the ecosystem back.

Meanwhile, organizations report rising concerns about security challenges, such as prompt injection, tool poisoning, and rug pulls (46%), as well as access controls, credentials, and authentications (40%).

For MCP to scale, it must improve in the areas of discovery, manageability, and security and governance. Without these improvements, MCP risks falling into the same traps that plagued early service meshes and API gateways, promising flexibility but delivering friction. Because MCP underpins how agents get access to external data and tools, its maturity is critical to ensuring scaling and moving to production can be done securely inside enterprise environments. In fact, nearly half (45%) of all organizations find guaranteeing tools are secure, trusted, and enterprise ready to be the greatest challenge with agentic build tools.

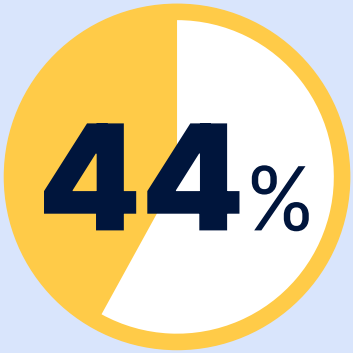
The immaturity and security challenges of current MCP tooling make for a fragile foundation at this stage of agentic adoption. MCPs play a critical role in supercharging agents, enabling them to connect securely to tools, data, and external systems, but today's implementations remain early and uneven. Unlocking their full potential will require a new generation of MCP platforms built for enterprise scale, with secure-by-default architectures, robust governance, and integrated policy enforcement.



THREE REQUIREMENTS FOR MCP TO SCALE

If Model Context Protocol is to become the backbone of agents, it must mature quickly in three critical areas:

1 DISCOVERY



of organizations struggle to find trustworthy MCP servers.

SOLUTION

Stronger registries and standardized validation mechanisms.

2 MANAGEABILITY

Teams need simplified configuration and deployment patterns that integrate cleanly into existing environments—not add more overhead.

3 SECURITY & GOVERNANCE

Enterprise adoption demands secure-by-default architectures with clear visibility, auditability, policy enforcement, and alignment with established security models.

Section 4

Distribution and Sharing: *THE MISSING LINK TO SCALING AGENT ADOPTION*

Building agents is only half the challenge; sharing and reusing them is what enables scale. Without effective distribution mechanisms, every team rebuilds the same capabilities, knowledge stays siloed, and agent adoption remains confined to isolated pockets of experimentation. Organizations can't achieve enterprise-wide impact if successful agents remain trapped in individual repositories or shared through informal channels.

From an organizational perspective, distribution solves a fundamental scaling problem: how do you enable teams to discover, trust, and reuse agents built by others? This isn't just about technology; it's about creating the cultural and operational conditions for collaboration. Teams need confidence that shared agents are secure, maintained, and compatible with their environments before they'll adopt them over building from scratch.

Yet the technology infrastructure to support this vision remains immature. Despite strong interest and early momentum, the mechanisms for distributing and sharing agents remain immature and fragmented. Developers and enterprises alike are running into the same problem: once an agent has been built, there's no standard path to production-scale sharing or deployment.

Across the global sample, commercial platforms and marketplaces are currently the most relied-upon distribution channel, used by 66% of respondents. Git-based source repositories follow closely at 51%, and nearly half of respondents (48%) rely on internal documentation, wikis, or tribal knowledge within their teams. Containers, which have become the standard in broader cloud-native software delivery, are used in only 38% of agent distribution workflows, though that number is significantly higher among more advanced teams, suggesting this is the trend.

The picture that emerges is one of patchwork distribution, where different teams adopt different approaches depending on their maturity, risk tolerance, or tooling preferences. But the impact is universal: without consistent packaging semantics, version control becomes fragile, integration into enterprise systems breaks down, and agents become difficult to audit, govern, or reproduce.

Security looms large over this distribution problem. One-third of respondents cite it as the single biggest barrier to sharing agents across teams or business units. Enterprises face a daunting set of compliance concerns, such as data privacy, auditing, runtime integrity, access controls, and most current distribution methods lack the visibility and enforcement to meet enterprise thresholds. In particular, governance becomes brittle when agents are passed informally or documented only in local wikis or Git repositories.

The result is an ecosystem that feels eerily familiar. Sharing agents today resembles the pre-container microservices era: chaotic, inconsistent, and highly manual. Teams are building promising prototypes but lack the infrastructure and packaging standards to share them safely, at scale, with consistency and reliability. Versioning is done by hand. Configuration is bespoke. Runtime behavior is unpredictable.

Just as containerization revolutionized the way microservices were packaged, distributed, and deployed, the agent ecosystem now needs a parallel shift. Common formats, portable definitions, and standardized lifecycle tools will be critical to unlocking true agent scalability. Without them, enterprises will remain mired in bespoke pipelines, constrained by elevated risk, and unable to share agents at the speed innovation requires, limiting their ability to scale.

MAKING AGENT SHARING SEAMLESS: **WHAT NEEDS TO HAPPEN**

For agent adoption to scale, organizations must move beyond isolated experiments to true sharing and reuse. Today's distribution challenges reveal what's needed:

Top 5 blockers to seamless agent sharing

1. SECURITY CONCERNS

31% **REQUIRE** Signed, scannable agent packages with provenance tracking

2. INTEGRATION WITH EXISTING INFRASTRUCTURE

29% **REQUIRE** Standardized interfaces and enterprise system compatibility

3. COMPLIANCE AND GOVERNANCE

28% **REQUIRE** Built-in policy enforcement, audit trails, and data privacy controls

4. VERSIONING AND MAINTENANCE ACROSS TEAMS

27% **REQUIRE** Centralized registries with dependency management and rollback capabilities

5. PERFORMANCE VARIABILITY ACROSS ENVIRONMENTS

27% **REQUIRE** Portable packaging that ensures consistent behavior regardless of deployment context

The bottom line: Agent distribution needs the same infrastructure maturity that made container adoption successful: secure registries, standard packaging formats, and tooling that makes sharing easy and trustworthy.

Section 5

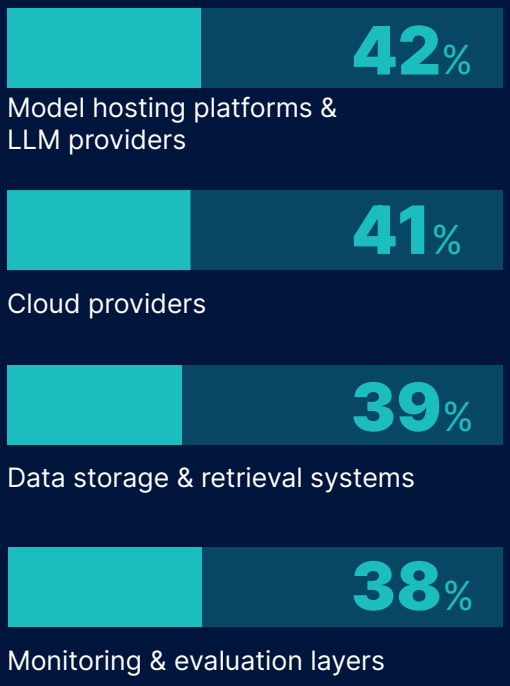
Lock-In Fears Are Real: ***PORTABILITY AS THE FOUNDATION OF RESILIENT AGENT ARCHITECTURES***

If orchestration is the technical bottleneck for agentic AI, vendor lock-in is the strategic one. Even as organizations invest heavily in agents, many are sounding the alarm about the fragility of their supply chains. Seventy-six percent of global respondents report active concerns about vendor lock-in—rising to 88% in France, 83% in Japan, and 82% in the UK. And these aren't theoretical anxieties. They center on the very layers of the stack that power agentic systems today.

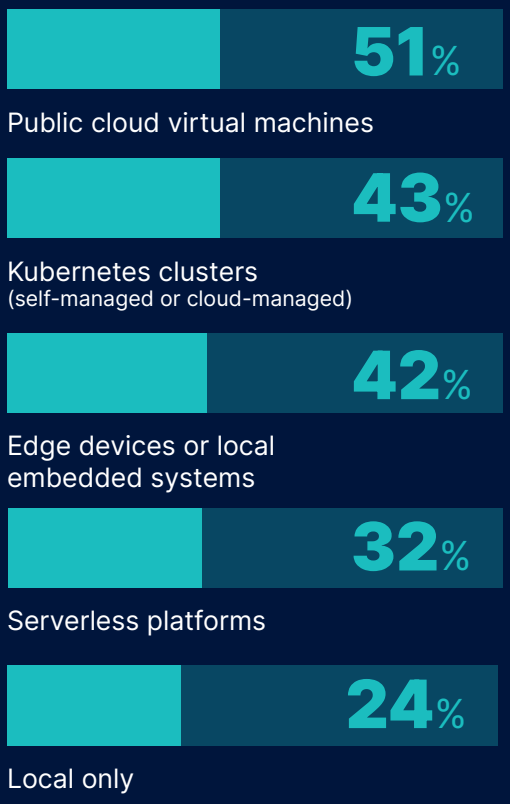
SEVENTY-SIX PERCENT OF GLOBAL RESPONDENTS REPORT ACTIVE CONCERNS ABOUT VENDOR LOCK-IN

These concerns center on the layers where inference meets infrastructure. Model hosting platforms and LLM providers (42% each) top the list of lock-in risks, closely followed by cloud providers (41%), data storage and retrieval systems (39%), and monitoring and evaluation layers (38%). Enterprises fear that today's rapid adoption could translate into long-term dependency, limiting flexibility and innovation down the road.

Top Lock-In Concerns Across the Agentic AI Stack



Typical deployment & run platforms



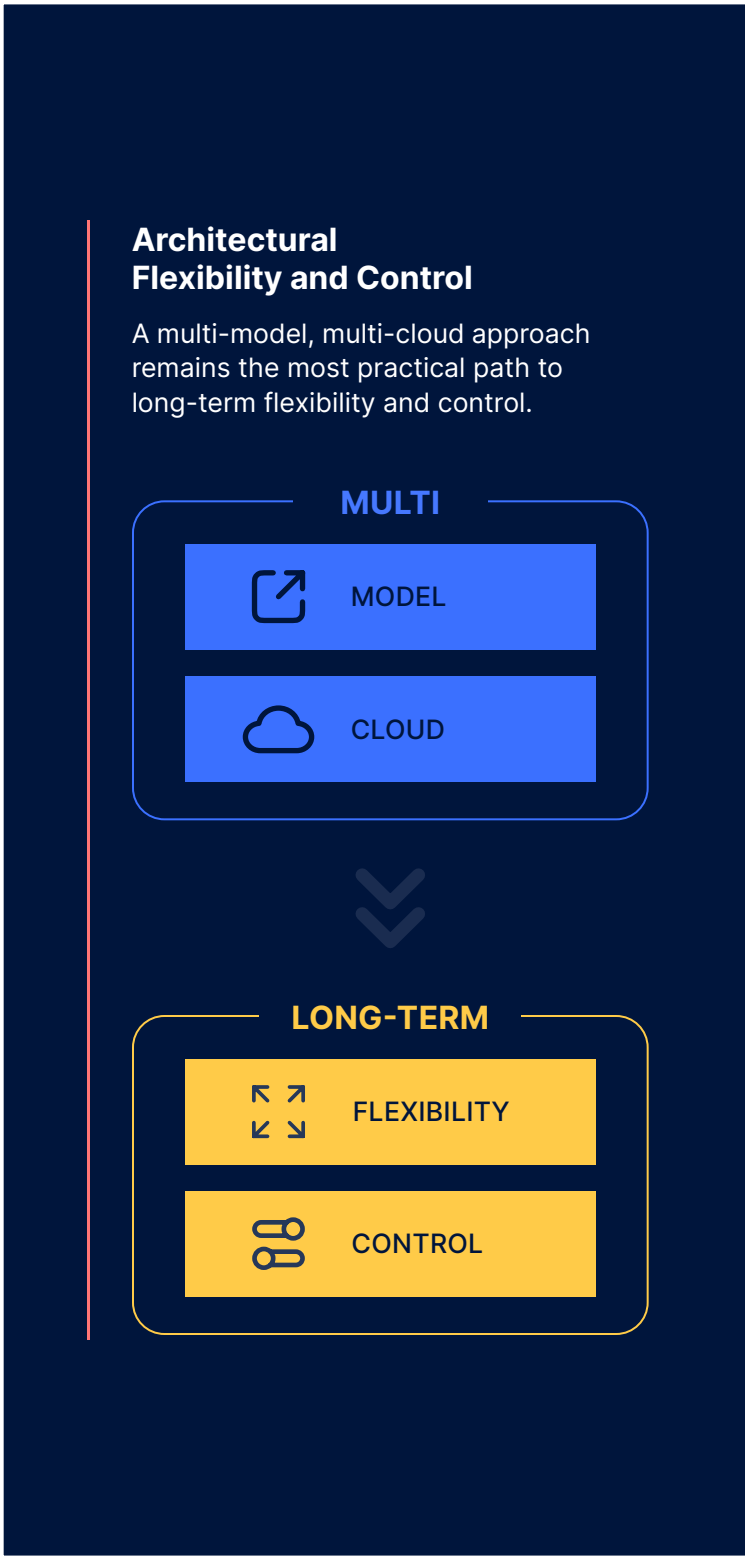
To mitigate that risk, organizations are diversifying rather than consolidating. They are spreading workloads across multiple models, tools, and cloud environments. Among the 61% of organizations that use both cloud-hosted and locally hosted models, the leading drivers are control (64%), data privacy (60%), and compliance (54%), with cost being far less influential (41%).

This strategy comes with trade-offs. Each additional platform, model, or runtime adds coordination overhead and security exposure, creating what can be described as “dependency management by distribution”. Still, the consensus is clear: a multi-model, multi-cloud approach remains the most practical path to long-term flexibility and control.

Deployment patterns reflect this mindset. Over half (51%) of organizations run agents in the public cloud, 40% on-premises, 32% on serverless platforms, and 24% locally. In total, 79% operate across two or more environments, with common pairings such as Kubernetes clusters with public cloud VMs (48%) or hybrid on-prem/cloud setups (35%).

Amid this diversity, containers provide the connective tissue–a consistent, portable layer that enables agents to move securely between environments. They remain one of the few technologies capable of mitigating lock-in risks while maintaining governance, reproducibility, and scale.

In short, the agentic future will not be monolithic. It will be multi-cloud, multi-model, and multi-environment, making open standards and portable infrastructure essential to sustaining enterprise trust and flexibility.



Section 6

Agents as the New Microservices: ***CONTAINERS AS THE FOUNDATION OF AGENTIC INFRASTRUCTURE***

As organizations navigate lock-in risks and increasing complexity, one pattern stands out: containers have become the foundational unit of agentic infrastructure. Their role is not merely theoretical or aspirational, rather it is deeply operational. Nearly all organizations surveyed (94%) already use containers in their agent development or production workflows, and the remainder plan to adopt them.

As organizations ramp up agent adoption, they are extending the same cloud-native workflows that already power their application pipelines–like microservices CI/CD, and container orchestration–to support these new workflows. In fact, ninety-four percent of all teams building agents rely on containers. This approach has real advantages: it leverages familiar tooling, pipelines, and operational patterns, accelerating time-to-market and reducing overhead. Containers provide built-in portability, version control, and environmental consistency, while cloud-native architectures offer ecosystem compatibility and cost control. In this way, agent development is evolving as a natural extension of cloud-native maturity, not a departure from it.

What's new is how containers are being adapted for agentic workloads. In fact, 98% of organizations report that they largely or mostly use the same development and deployment workflows for agents as they do for traditional cloud-native applications. This continuity underscores how containerization is not being reinvented for agentic AI, but extended. Teams are leveraging the same CI/CD pipelines,

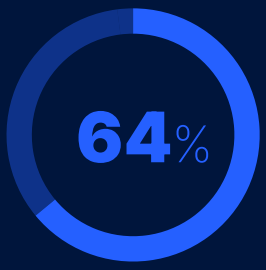
Container Use in Agentic Stack

94%

of surveyed organizations already use containers in their agent development or production workflows, and the remainder plan to adopt them.

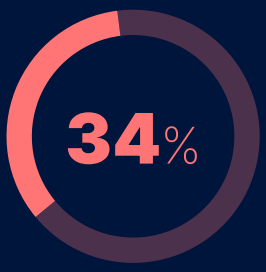
Agents and Cloud Native Software Overlap

98% of organizations report that they largely or mostly use the same workflows for agents as they do for traditional cloud-native apps



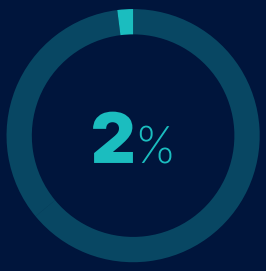
YES

our workflows are largely the same



MOSTLY

but added some AI-specific tooling or processes



NO

we've had to significatly change how we build and deploy

orchestration layers, and runtime standards that power their microservices to now support agentic workflows.

Beyond portability and rollback, teams now rely on containerization to provide sandboxed execution, version control, and predictable environments for probabilistic or autonomous behavior. These capabilities make containers the natural substrate for scaling agents securely and repeatedly.

While most organizations continue to use familiar CI/CD and microservice workflows, they are layering new agent-specific capabilities on top–features that directly address the ecosystem's twin challenges of complexity and security. These include dynamic orchestration, which manages multiple interacting components in real time; model isolation, which strengthens security and reliability through sandboxed and controlled execution environments; and contextual data management, which ensures agents can access relevant information securely and efficiently. In many ways, agents are becoming the new microservices–autonomous, composable units that extend across distributed environments. **But while they share some of the same modular benefits, agents also introduce new layers of complexity**. They have more moving parts to coordinate, more surface areas to secure, and an even stronger need for enterprise control and governance, especially as agents get access to sensitive tools and data and can act on their own. The same challenges that once defined microservice adoption, like dependency management, observability, and security are now resurfacing in agent development, amplified by the complexity of AI-driven behaviors.

FROM MICROSERVICES TO AGENTS: **CONTAINER-POWERED EVOLUTION**

Same foundation, new capabilities:

MICROSERVICES
deterministic, request-response, explicitly orchestrated

AGENTS
LLM-driven, goal-oriented, self-coordinating

Containers bridge both eras:

- Isolation for unpredictable agent behavior
- Portability across hybrid model deployments
- Versioning for safe rollback

Agents are microservices with intelligence, and containers remain the substrate that makes distributed autonomy safe and scalable.

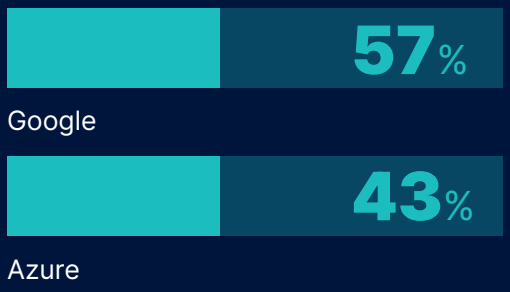
Cloud-native tools from leading providers remain central to this evolution. More than half of organizations use build tools from Google (57%), and 43% rely on Azure's container services. Open-source ecosystems continue to expand alongside these cloud platforms to support agent-centric development. Within that landscape, platforms like Docker (29%), Kubernetes (25%), and other container-orchestration frameworks play key roles as common operational primitives, rather than branded end points.

For many teams, container platforms are rapidly evolving to meet the demands of agentic development. Beyond serving as reliable runtimes, they are introducing services that make building, sharing and operating agents simpler and more secure: AI-powered assistants help users navigate platform operations, inference services, and local model execution for development flexibility, GPU-on-demand capabilities for scaling compute, and tools that simplify orchestration from development to production. At the same time, container platforms are beginning to bring structure and governance to MCP adoption, helping enterprises enforce security, consistency, and policy at scale.

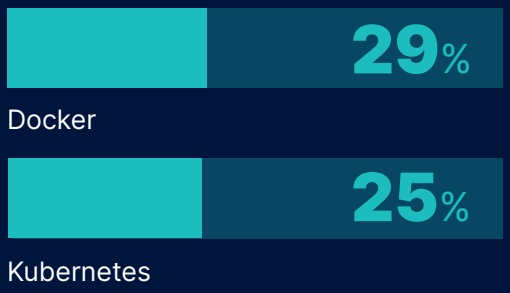
The principle remains the same, but the purpose has shifted. Container infrastructure is no longer just about uniform deployment across clouds. The future of agentic AI will not be a return to monoliths or black-box platforms. It will be built on the same foundations that transformed enterprise software a decade ago, and containers are once again at the heart of that transformation.

Top container tools organizations use for their agentic stack

Organizations use cloud-native tools from leading providers



Open-source ecosystems



Section 7

Conclusion: *FROM EARLY WINS TO ENTERPRISE-GRADE SCALE*

The signal is clear: agentic AI has moved out of the lab and into day-to-day operations. A majority of organizations report agents in production and building agents ranks as a high priority, but the center of gravity is still inward. Teams are deploying agents first where they can boost productivity with controlled risk (DevOps, security automation, and process automation) rather than in external, revenue-bearing experiences.

What's stalling broader impact isn't a lack of interest or use; it's trust, complexity, and uneven pathways to scale. Security remains the dominant barrier, with organizations struggling to ensure tools are enterprise-ready, implement proper access controls, and maintain secure isolation between agents and systems. **Orchestration and integration are the "silent killers," making promising pilots fragile as teams connect multiple models, environments, and clouds.**

MCP is poised to be the connective tissue for the agent ecosystem, but it's not enterprise-ready by default. While awareness and experimentation are high, teams are feeling the operational strain of managing a growing number of MCP tools, configurations, and permissions. Security and trust remain the most significant friction points: it's difficult to identify reliable MCP servers, and manage access controls, credentials, and authentication. Emerging threats like tool poisoning and prompt injection are further compounding the challenge. Until the ecosystem matures with standardized discovery, configuration, and policy enforcement, MCP will remain a source of both power and operational overhead, as well as risk.

At the same time, architectural patterns are converging. Most teams are extending familiar cloud-native practices to agents: containers are foundational, hybrid and multi-cloud are normal, and many organizations blend local and hosted models for control and compliance. This portability is strategic; three in four teams worry about model and cloud lock-in, yet the same diversification introduces new coordination and governance burdens.

Agent distribution is the least mature leg of the stool. Sharing patterns remain fragmented across marketplaces, repos, and internal wikis, making reproducibility, auditing, and policy enforcement fragile. The ecosystem needs secure, inspectable, and portable packaging semantics for agents–akin to what the Open Container Initiative (OCI) did for containers–to turn isolated wins into repeatable, governed deployments.

The path forward doesn't require reinvention so much as consolidation around a trust layer: access to trusted content and components that can be safely discovered and reused; secure-by-default runtimes; standardized orchestration and policy; and portable, auditable packaging.

Containers remain the practical substrate for that trust by providing isolation, flexibility, interoperability, versioning, and reproducibility, while MCP needs complementary controls for discovery, signing, sandboxing, and runtime policy to earn enterprise confidence.

Agentic AI's near-term value is already real in internal workflows; unlocking the next wave depends on standardizing how we secure, orchestrate, and ship agents. Teams that invest now in this trust layer, on top of the container foundations they already know, will be first to scale agents from local productivity to durable, enterprise-wide outcomes.

> *CONTAINERS REMAIN THE PRACTICAL SUBSTRATE FOR THAT TRUST BY PROVIDING* ***ISOLATION, FLEXIBILITY, INTEROPERABILITY, VERSIONING, AND REPRODUCIBILITY***

NEXT STEPS
FOR LEADING COMPANIES

Tame complexity with standard orchestration. Favor container-centric pipelines and unified gateways that abstract multi-model, multi-tool, multi-cloud sprawl.

Ensure access to trusted content and components. Use verified sources for models, MCP servers, and agents to reduce security exposure and build confidence in the ecosystem.

Adopt portable packaging for agents. Move toward container-like, signed, and inspectable artifacts to make sharing safe and repeatable.

Codify security as architecture, not a checklist. Treat sandboxing, credentials, and policy enforcement as first-class concerns across agent runtime and MCP tooling.

Diversify without drifting. Use multi-model and multi-cloud approaches for flexibility while centralizing governance, observability, and rollback paths.

APPENDIX

RESEARCH METHODOLOGY

The survey included 805 respondents sourced from a leading global online panel provider. They were selected from the panel based on geographic and role-based quotas, as well as screening questions based on role in IT, decision-making role, company size, and familiarity with agentic AI. Participants were IT decision-makers and purchase influencers with the ability to accurately understand agentic AI. Selected respondents were further screened based on self-reported agentic AI knowledge and attentiveness to survey questions.

ROLE QUOTAS

The survey divided respondents into three broad roles: Leadership 33%, DevOps 33%, App Devs 34%. Respondents were asked to select which role – from a list of 14 options – most closely described their primary responsibility, even if none were quite right or even if they performed more than one of these roles. Answers were consolidated into those three broad roles.

GEOGRAPHIC QUOTAS

The survey included respondents from North America (n=300), Europe (n=252), and APAC (n=253) countries.

RESPONDENT SCREENS

Role: All respondents were required to indicate that they were responsible for or had influence in evaluating and/or selecting IT or software for their organization.

Company size: All respondents must self-report their companies' number of employees. In total, the survey includes 1% of respondents from companies with 1-99 employees, 2% of respondents from companies with 100-249 employees, 8% of respondents from companies with 250-499 employees, 26% from companies with 500-999 employees, 41% from companies with 1,000 to 4,999 employees, 14% from companies with 5,000 to 9,999 employees, 5% from companies with 10,000 to 24,999 employees, 2% from companies with 25,000 to 49,999 employees, and 1% from companies with 50,000 or more employees.

Information level: In our experience, it is possible to have "qualifying respondents" who nevertheless prove to have too little information or knowledge about the space to provide useful data from which to draw insights. We therefore apply an "information" screen to respondents as well. Specifically, we ask whether or not respondents could explain certain terms to their colleagues if asked to do so. In order to qualify for this survey, a respondent must say "yes" to this question for the term "Agentic AI".

"Attention" level: It is easy for respondents to speed through surveys or not pay enough attention to provide useful data. We make an effort to exclude these respondents as well, as they generally provide less useful data. In this survey, respondents were screened out for "attention" reasons if they said they could explain the made-up term "Greenfield as a Service (GaaS)" to a colleague in the same question used for the Information Screen noted above.

RESPONDENT SCREENS

It is technically impossible and improper to list a margin of error for a survey of this type. The respondents for this sample were drawn from an online panel with an unknown relationship to the total universe, about which we also do not know the true demographics. As such, the exact representativeness of this, or any similarly produced sample, is unknown.