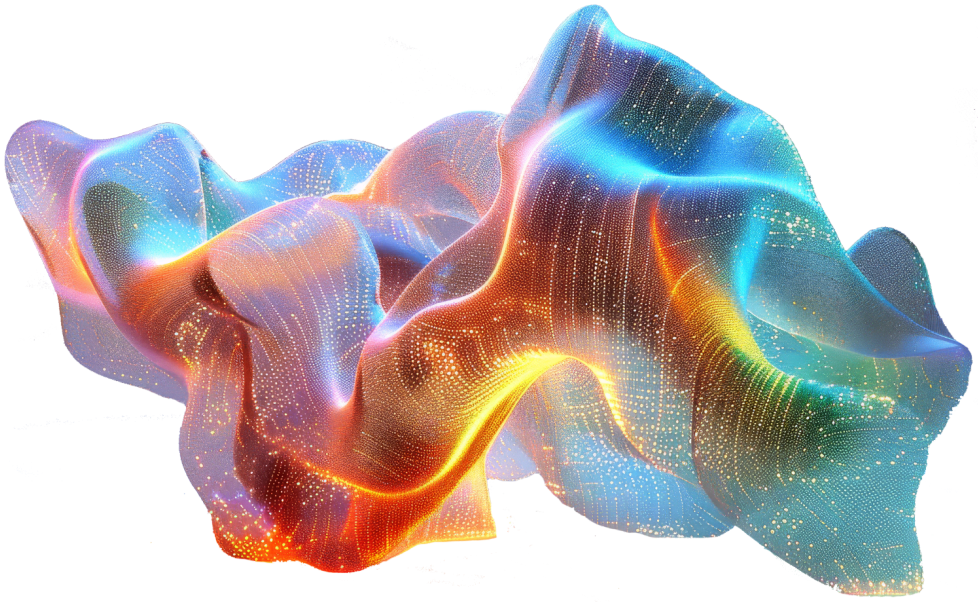
Google Cloud

Cloud Infrastructure in the Agent-Native Era



Introduction

Governing the internet of agents

The ability to deploy agents effectively is no longer a luxury—it is a core competitive differentiator for enterprises. However, as organizations move from experimental pilots to production-grade ecosystems, they are encountering a phenomenon known as "agent sprawl." Agents are being developed at an unprecedented rate across heterogeneous frameworks, protocols, and locations—from the cloud core to the local workstations of "invisible insiders."

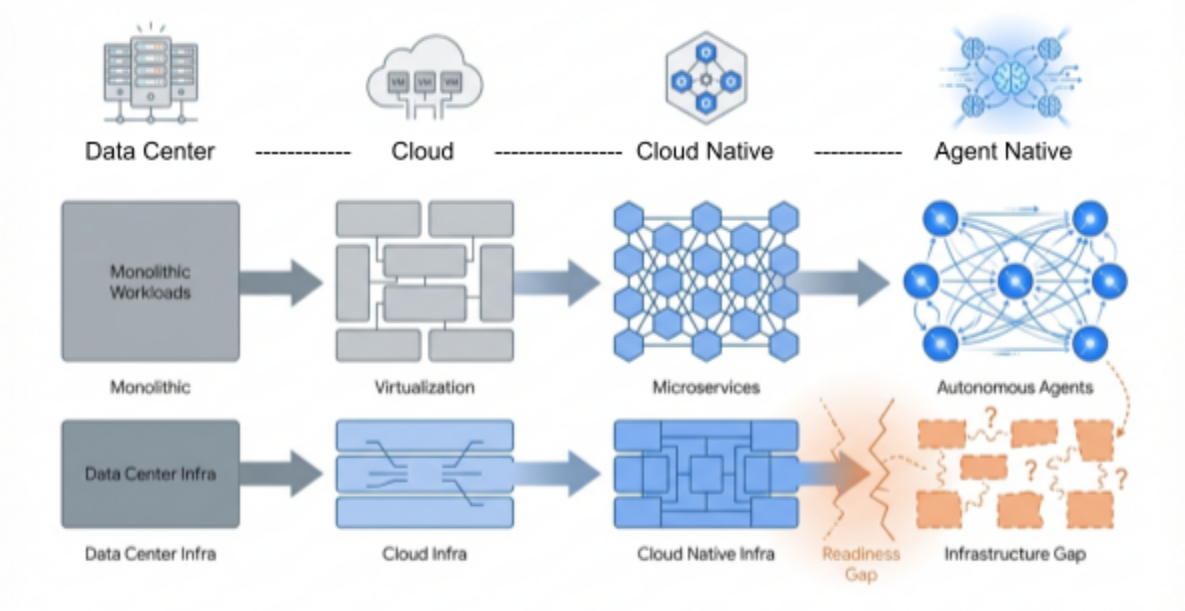
This sprawl introduces a non-deterministic complexity that traditional cloud-native infrastructure was never designed to handle. As agents and tools combine data access, external connectivity, and execution power, new risk vectors arise and drive new governance requirements. Framework-specific governance becomes an impossible task to scale in an environment where framework silos and incompatibility constantly emerge. Governance of agent sprawl must leverage an infrastructure-led approach that is ubiquitous, framework agnostic, standards-based and extensible.

To move past the current stagnation in AI adoption, the cloud must evolve from a passive transport mechanism into an active, intelligent participant that understands agents as a core primitive. By "shifting down" the complexities of identity, discovery, and authorization into the underlying platform, we remove the burden from the developer and place governance where it is most effective: in the ubiquitous fabric of the infrastructure itself.

Throughout this paper we outline the state of the industry and the requirements of agentic applications, we then discuss the technical imperatives and open source standards for an agent-native infrastructure, providing a vision for turning a chaotic sprawl of autonomous agents into a standardized, secure, performant, and governed "internet of agents."

The pivot from cloud-native to agent-native

Enterprise technology is pivoting from the determinism of cloud-native microservices—characterized by predictable HTTP request-response cycles—to the probabilistic, autonomy of **agentic AI**. This transition redefines how software is built, secured, connected, and operated. The industry is moving from cloud-native to agent-native.



Enterprises envision a future workforce composed of autonomous digital agents capable of reasoning, planning, and executing complex, multi-step tasks. The 2026 IDC Special Report on AI in Networking, reveals that **44% of North American organizations plan to deploy 21 or more AI applications in the coming year alone**[1]. This figure indicates a move from experimental applications to scaled, production-grade ecosystems that are integral to business operations.

The ability to deploy agents effectively is now seen as a core competitive differentiator, with organizations recognizing that the "Agentic" advantage will define market leadership in the latter half of the decade. The pivot is broad-based, with **32% of organizations already classifying themselves as having "substantial use" of AI**, and another **51% in "select use"**[1]. This maturity distribution suggests that the market has moved past the initial hype cycle and is now grappling with the hard realities of implementation.

The actual movement of AI projects from select to substantial use has **stagnated**. Projects that succeed in the sandbox fail in production. They remain blocked by infrastructure that simply wasn't designed for the unique behaviors of autonomous agents. **We are attempting to run agentic AI workloads on cloud-native infrastructure.**

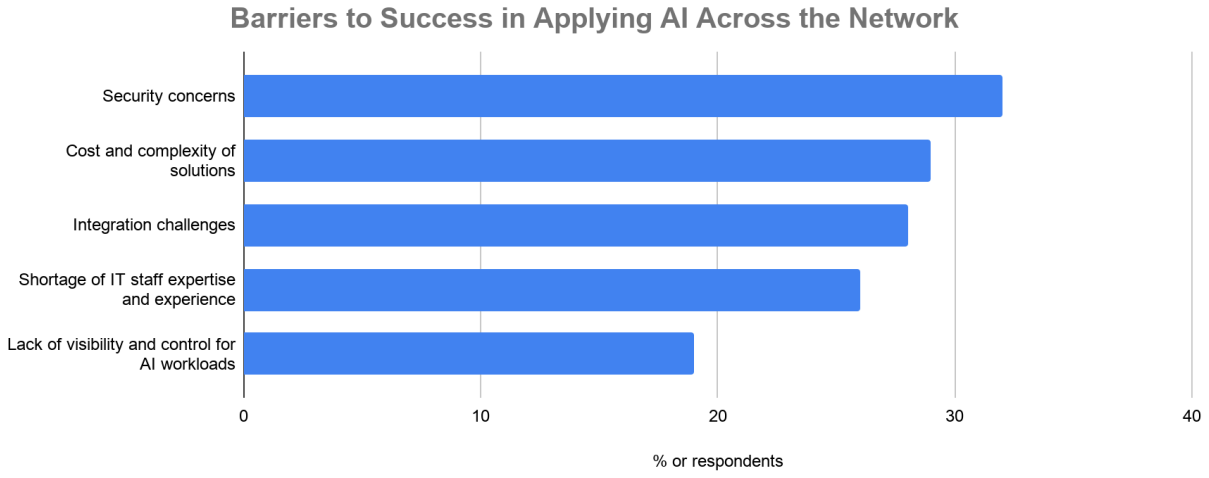
- **Cloud-native** was built for **deterministic microservices**. These applications are stateless, ephemeral, and communicate via predictable HTTP/REST pathways. The infrastructure assumes

that if a service is healthy, the request will succeed. Security is perimeter-based, and traffic patterns are largely between users and the application front-end.
- **Agent-native** involves **probabilistic agentic workflows**. Agents maintain state (memory), engage in complex reasoning loops, and generate unpredictable traffic patterns as they autonomously decide which tools to call. A "healthy" agent may still hallucinate, enter an infinite loop, or decide to exfiltrate data based on a malicious prompt.

Agents represent a fundamentally new class of workload with unique requirements for state, identity, connectivity, and governance. The cloud infrastructure must evolve to address these new requirements and enable agentic applications to move from pilot to production.

While the list of challenges is wide-ranging, these specific barriers to production consistently top the charts across the survey data:

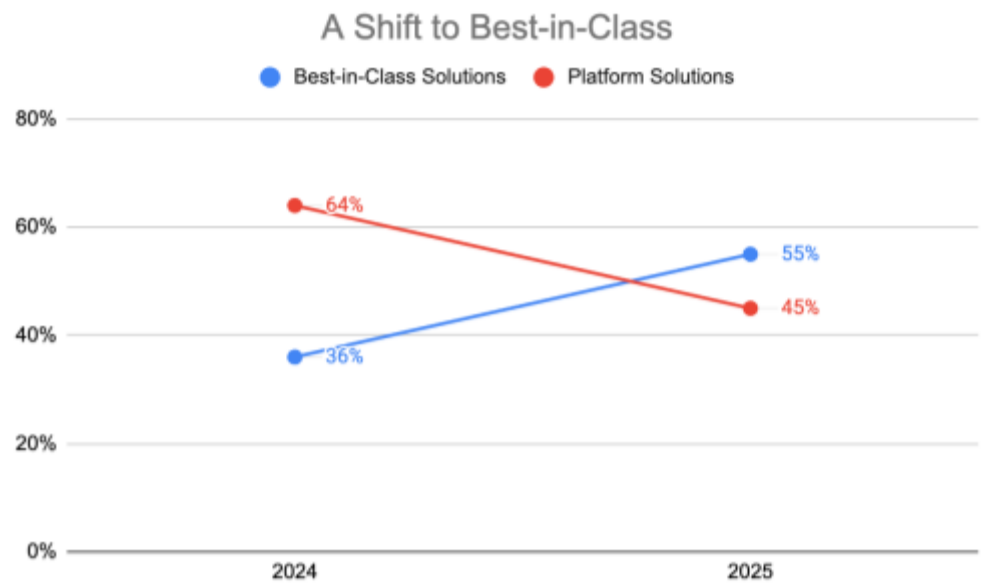
- **Security and governance**: This is the single biggest barrier to success. As reported by IDC[1], security is the top-ranked impediment for almost every industry vertical. It is also the leading concern regarding distributed AI workloads at the edge and agentic AI.
- **Observability**: A new barrier noted by IDC in their report[1] is the "lack of visibility and control for AI workloads." As AI traffic increases, networking teams are struggling to see what is happening inside the pipes, which impedes control and optimization.
- **Cost and complexity**: As organizations attempt to scale, the "cost and complexity of solutions" has become a primary blockade. This complexity is a major driver behind the market's pivot away from platforms toward best-in-class solutions, as buyers seek specialized tools to simplify specific operational headaches.
- **Integration**: "Integration challenges" rank nearly as high as cost, reflecting the difficulty of melding new AI workloads with legacy environments. This is further complicated by the "shortage of IT staff expertise," which remains a significant hurdle, though it has lessened slightly compared to previous years.



Data Source: IDC 2026 AI in Networking Special Report[1]

Platforms vs. point solutions

Unified platforms are well positioned to address cost, complexity, and integration challenges, and are usually preferred. However, preference for platforms has given way to best-in-class point solutions focused on the rapidly evolving requirements of agentic applications. The number of respondents choosing best-in-class point solutions in the 2026 IDC Special Report on AI in Networking rose to 55%, compared to 36% in the prior 2024 report[1].



Buyer preferences have inverted in a single year. While enterprises historically favor the simplicity of platforms, the lack of readiness for agentic workloads is driving a migration toward specialized, best-in-class point solutions.

Data source: IDC 2026 AI in Networking Special Report[1]

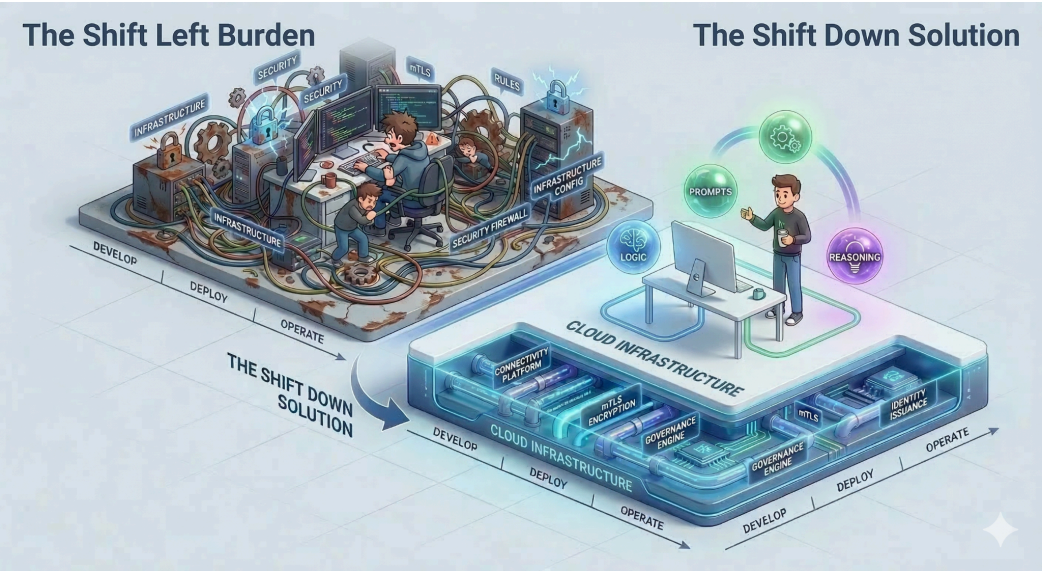
This shift stems from a perceived lack of readiness and velocity in current platforms to manage complex agentic workflows. Buyers now prioritize innovative, dedicated tools over integrated platforms to secure reasoning loops and optimize token usage, despite increased integration complexity. Nevertheless, the benefits of a platform approach remain top of mind and access to an extensible platform that can integrate best-of-breed pure play solutions is highly desirable.

A successful agent-native infrastructure requires a "ready" platform that balances simplicity with sophistication. Such a platform must "shift down" integration burdens into the infrastructure, allowing for composable deployment of best-of-breed tools while simplifying operations as technology evolves.

Shift down just-in-time

In the context of agentic AI, where complexity explodes due to the non-deterministic nature of the workloads, shift down operations are a must. Shift down operations push the immense complexities of authorization, security, and networking down into the infrastructure, effectively simplifying operations that would otherwise be hard to sustain. The platform automatically handles the "plumbing"—identity issuance, secure connectivity, policy, and observability.

Scaling the large number of ephemeral connections and services involved in agentic applications requires a just-in-time approach to auto provisioning the infrastructure. Just-in-time provisioning enables agents to deploy their own code and services without a human in the loop.



By shifting these complexities down, and automatically binding services to the right infrastructure resources, the agent-native cloud resolves the critical bottlenecks that are currently stalling adoption: **security risks** (the #1 barrier) and **integration complexity** (the #3 barrier).

The internet of agents: governing agent sprawl

Agents are being developed everywhere and at unprecedented rates. Agents in turn rely on a sprawling ecosystem of heterogeneous frameworks, protocols, tools, providers and other agents. The phenomenon is known as Agent sprawl and results in non-deterministic communication patterns that introduce unprecedented challenges for security, governance, and operations. Managing these "conversational communications" is further complicated as agents and tools are deployed in a variety of form factors (serverless, VMs, managed runtimes, etc.).

AI workloads are highly distributed. Agents, tools and models may be deployed on-prem, across a multitude of cloud providers or on hosts distributed across the Internet. This surface continues to expand with agents increasingly being deployed on client workstations beyond the footprint of the data center infrastructure, effectively creating an Internet of agents.

Client agents, often referred to as "agentic clients," are autonomous AI assistants that operate locally on edge devices, mobile platforms, and developer workstations using desktop tools like Cursor, Claude Desktop, Gemini CLI, and VSCode. As coding and productivity agents running locally on corporate devices become some of the most widely adopted AI tools, they increasingly function outside the standard enterprise network perimeter, directly assisting users at the endpoint. Because these agents are ephemeral, use arbitrary frameworks, and run locally and autonomously, they are incredibly difficult to govern. The mechanisms used by the cloud infrastructure must extend to these clients on the internet.

Agents, and in particular client agents, possess what is described as a "lethal trifecta" of risk: they have access to corporate data, they connect to third-party tools and external large language models (LLMs) on the public internet, and they can execute actions, sometimes via untrusted local tools. Without robust governance, these client agents operate as "invisible insiders" that can exfiltrate sensitive data or perform unauthorized actions using the full privileges of the employee.

New risk vectors arise with agent sprawl with the potential for impact to brand reputation, leakage of sensitive data, exposure of personally identifiable information, regulatory compliance issues and the execution of destructive or unintended actions. Governance based on specialized security and observability with fluency in natural language and emerging AI protocols is required ubiquitously.

Pursuing the governance of agentic applications in each of the diverse development frameworks quickly creates silos with bespoke and inconsistent governance policies that must somehow be unified. A governance model disaggregated across a multitude of frameworks would be very hard to scale for the enterprise. Governance in the frameworks also places the responsibility for enabling the right controls on the developers, creating concerns around trust and compliance. In contrast, the cloud infrastructure offers a common surface for the deployment of agents and the homogeneous implementation of governance controls with the appropriate separation of concerns.

The agent-native infrastructure: imperatives for agentic apps

An infrastructure led approach is critical to enable agentic applications and realize the vision of the internet of agents with effective governance, security and visibility. The cloud infrastructure provides the unified platform to which a sprawling mix of agents, tools, models and resources can register and authenticate. The ubiquity of the cloud infrastructure makes it the ideal medium to govern, optimize and observe agentic resources developed and deployed on a wildly heterogeneous mix of frameworks, form factors and locations, effectively bringing structure to what is a naturally non-deterministic set of interactions. Discovery is fundamental to any Internet-like system as evidenced by the core role of DNS in the World Wide Web; a federated authority for identity and attested registration of agentic services, delivered as an infrastructure component, enables the automated discovery of agents and tools in the context of their capabilities. By structuring the identity, registration and discovery of agentic resources, an ubiquitous infrastructure also offers the ideal surface for the enforcement of governance policies through the enablement of authentication, authorization and session establishment inspection. Enabling these functions in the infrastructure is critical to enabling the separation of concerns that is fundamental to IT governance, and also decouples the governance mechanisms from specific frameworks or developer preferences. Extensibility is also a key attribute of the infrastructure, allowing the governance framework to evolve as the supporting technologies mature without altering the infrastructure foundation on which these mechanisms rely.

The agent-native cloud must adopt an evolved architecture where the cloud infrastructure natively understands agents as a core primitive. The infrastructure is the foundation on which the cloud relies for the delivery of enhanced functionality that understands agents to address the agentic application requirements. The infrastructure must evolve from a passive transport and hosting mechanism into an active, intelligent participant in the agentic workflow. The evolved cloud infrastructure must address agent sprawl and the many-to-many scale challenge of connecting agents to thousands of potential tools and data sources. In order to fulfill this mission, the infrastructure must address the following imperatives:

Governance: Governance is the key imperative to unlock the production deployment of agentic applications at scale. Without a revised governance paradigm that aligns to the autonomous and non-deterministic nature of agentic applications these cannot be delivered safely. Governance relies on

agent/user-delegated identities and validated registries for the creation of policies, and relies on extensible proxy data planes for authorization, access controls, and the consistent insertion of advanced protection services.

Discovery: To combat the sprawl of highly distributed, ephemeral agents and tools, the infrastructure must include standards based authoritative registries that provide a federated single source of truth for the discovery of agents, tools, skills, and their capabilities.

Identity: Because agents operate autonomously—often forming transient partnerships or acting on behalf of human users—traditional static credentials and API keys fail. The platform must automatically provision standards-based identity (e.g. SPIFFE) to strongly attest the identity of each workload, supporting delegation of user identity, enabling zero-trust architectures and granular access control without burdening the developer.

Authorization and policy enforcement: The infrastructure must provide ubiquitous in-line mechanisms to authenticate and authorize non-deterministic agent connections. The infrastructure must extend to deliver rich identity paradigms such as SPIFFE and also achieve protocol fluency, natively understanding and parsing agentic protocols like the Model Context Protocol (MCP), the Agent2Agent (A2A) protocol, and gRPC. In order to effectively enforce policies based on agentic semantics, the infrastructure must deeply inspect JSON-RPC and gRPC payloads to understand the actual intent of an agent, such as executing a specific tool.

Protection: The non-deterministic nature of agents introduces novel risks, such as tool poisoning, indirect prompt injections, and infinite execution loops. To counter this, the infrastructure must feature extensibility for in-line service insertion, allowing the dynamic injection of extensible security with support for native and best-of-breed pure play services (such as Google Cloud's Model Armor or partner ISV solutions) directly into the data path. This allows the platform to inspect and sanitize prompts at runtime without altering application code.

Observability: Because probabilistic models can hallucinate or fail unexpectedly, robust observability is a key imperative. Infrastructure must shift from tracking basic network packets to tracing data connections and lineage, logging token consumption and tracking model latencies, to render visualizations of the agent's complete reasoning and tool-calling trajectory.

Connectivity: Agentic applications put renewed pressure on network capacity, footprint, latency, and reliability. Connectivity must also evolve to understand AI semantics and support specialized routing that can optimize AI accelerator resource utilization for inference and agentic connections to minimize latency for computationally expensive reasoning loops.

Addressing the imperatives in the infrastructure

Together, these capabilities operate as a cohesive fabric, ensuring that highly autonomous systems remain standardized, secure, performant, and governed.

Governance

Effective governance of the agentic communications is the key imperative for the successful deployment of production grade agentic applications at scale.

Governance for agentic applications is achieved using a combination of evolved registries, identity, authentication, authorization, protection, and visibility.

The required governance architecture has an authoritative agent and tool registry at its core. Policies rely on this registry and strongly attested standard agent and user-to-agent delegated identity to effectively authenticate agentic connections that are authorized leveraging on-path mechanisms enabled in the infrastructure. This framework, centered around agent registries, identity, and infrastructure enforced authorization, is augmented with the use of in-line AI guardrail checks to provide a comprehensive governance approach to agentic communications.

Standardized discovery: registries as sources of truth

At the heart of the governance model required to secure agentic applications are the registries of tools, agents, and models that enable resource discovery for agentic applications. A registry system where all A2A cards and MCP capabilities are registered enables the scalable auto-discovery of agents that the internet of agents requires. Such a registry would also provide the main source of truth and trust upon which governance policies rely. The discovery of agents and tools is the starting point for the rich mix of non-deterministic connections that compose the autonomous agentic application. By governing the discovery mechanisms, the agentic applications can be effectively secured while allowing them to retain their autonomy and agile non-determinism. Different registries are likely to emerge in different environments, to succeed in enabling agentic applications these registries should follow a common standard definition that would allow them to federate and provide a reliable and consistent source of truth.

Google is leading industry collaboration in many fronts of the open source community to standardize the mechanisms for agent, tool, and skill discoverability. Google is an active contributor to the definition of the A2A and MCP protocols, as well as the open source definition of agent registry APIs to bring governance and control to AI artifacts and infrastructure. An open source agent registry API enables a secure, authoritative registry federation where teams can publish, discover, and share AI services, including MCP servers, agents, and skills, and deploy them easily to any environment.

Agentic identity

We have discussed the autonomous and non-deterministic nature of agents. Furthermore, agents can be short lived and ephemeral. Conventional security practices (such as network policies that only allow traffic between particular IP addresses) struggle to scale under this complexity. Agent identity requires agile authentication mechanisms that leverage identities that are decorrelated from IP addresses or hosts. Agent Identity is a new principal type distinct from traditional service accounts or user identities. Agent identities should feature:

- Auto-provisioning: Automatically tied to the runtime instance with strong certificate based attestation.
- Granular access control: Enables fine-grained IAM policies for specific resources or tools.
- Delegated authority: Allows agents to act on behalf of human users with OAuth tokens.

Agent identity requires an advanced framework capable of delivering strong attestation for IP independent identities. The industry is driving consensus on how to achieve this level of identity through the SPIFFE open source set of specifications. SPIFFE, the Secure Production Identity Framework for Everyone, automates and shifts down the identity work required from developers and also gives operations teams the visibility they require to support developers.

The SPIFFE framework consists of open-source specifications designed to bootstrap and issue identities to services across organizational boundaries and varied environments. At its core, the framework defines SVIDs—short-lived cryptographic identity documents—accessible through a simple API. Workloads leverage these documents to authenticate with one another, such as by signing JWT

tokens or establishing TLS connections. This architecture is natively extensible, supporting trust federation with external Certificate Authorities (CAs) and on-premises PKI infrastructure through trust bundles. These bundles allow the platform to aggregate multiple trust anchors, enabling integrated, secure mutual authentication for agentic workloads distributed across hybrid and multicloud environments.

Authorization rules for agentic flows must combine agent and user identity to provide granular access controls to the API. The agent may have permissions to access tools, meanwhile the information being accessed by the tools may be user specific. It is this combination of identity that defines the actual access rights for the agentic communication. The AI era demands a decisive shift toward an identity-first model centered on the AI application perimeter. This perimeter is an access security boundary that simplifies security by focusing on business outcomes rather than individual resource access controls. Oauth provides the standard mechanisms by which this delegated identity can be achieved.

Under this model, security is governed by three distinct identity types:

Identity Type	Purpose	Issuing System
User Identity (ID-1)	Human identity used to access the agent session	Human IdP (e.g., Cloud Identity, Entra, Auth0)
Agent Identity (ID-2)	Identity used by the agent to access APIs as itself Uses SPIFFE standards	Google Cloud (Auto-provisioned on creation)
User Delegated Identity (ID-3)	Identity used by the agent to access third-party tools on behalf of the user	Third-party OAuth server or tool registry (via OAuth dance)

Beyond user delegation, identity frameworks must evolve to identify and maintain context along the chains of services that call each other in multi-agent applications. A simple example may be an agent that is generally allowed to connect to the Internet, but assumes a different identity after having consulted an internal database. The chained series of events should result in a chained identity for which the action to connect to the internet is denied.

Authorization and policy enforcement

As agents proliferate in any location and on a variety of form factors (serverless, kubernetes, managed runtimes), transactions and connections must be authenticated and authorized in the infrastructure as connections may follow virtually any path possible and the responsibility of authenticating and authorizing connections must be shifted-down and taken out of the hands of the developers. An extensible data plane is key to enabling this ubiquitous authentication/authorization and policy enforcement surface. Additionally, an extensible data plane provides a surface of connectivity that allows the inspection of MCP and A2A calls and the enforcement of governance policies related to such calls.

The industry looks at Envoy to provide the standard for an extensible data plane for the agentic era. This foundation has evolved from a traditional Layer 7 proxy into an "agentic layer proxy". The use of the Envoy proxy is fundamental in enabling the ability of the network to insert the ecosystem of services required to secure and govern the agentic communications.

An Envoy infrastructure enables authentication via JWT tokens or TLS termination handling using external filters, the Envoy proxy external authorization filter (ext_authz) calls an extensible authorization service to check if the incoming request is authorized or not. These external filters are extensible and set to accommodate agentic semantics and SPIFFE identities today and as they evolve.

Data plane extensibility is a critical aspect of the infrastructure required to support agentic applications. As the industry matures so does the understanding of how to best secure, govern, and observe these agentic applications. The set of functionality and tools that can address these requirements will vary over time. An elastic platform that relies on an extensible data path can make operational simplicity the one constant in the cloud infrastructure along the journey of supporting AI agentic applications today and as they evolve to technical maturity.

Runtime protection: guardrails

Since agents are probabilistic, AI guardrails act as an "AI firewall" in the network path to inspect semantic transactions. AI guardrails scan for prompt injection, PII leakage, and malicious URLs across user-to-agent and agent-to-tool paths. Security administrators can set mandatory "floor settings" that developers cannot override, effectively implementing separation of concerns in governance.

Envoy filters to call out external processing (ext_proc) are available to easily insert guardrails as data plane extensions. This effectively shifts down the infrastructure toil required to include AI guardrails on specific connections by making the availability of guardrail services ubiquitous and consistent.

Observability: from traffic visibility to decision visibility

Traditional network monitoring tools look at packets, bytes, and TCP connections. This "traffic visibility" is insufficient for non-deterministic agents or debugging an agent that is stuck in a reasoning loop. Deploying autonomous systems introduces probabilistic uncertainty, challenging platform teams due to **non-deterministic execution paths** and complex **multi-agent coordination bottlenecks**. This also shifts the economic model from fixed infrastructure to variable intelligence expenses, resulting in an "unreliability tax" from high token costs and accumulated latency required to achieve enterprise-level accuracy. As noted in the IDC 2026 AI in Networking Report, "Lack of visibility and control for AI workloads" is a top barrier to adoption[1]. The agent-native cloud must shift observability to "decision visibility".

Tracing non-deterministic execution: the foundation

In traditional software, tracing a request involves following a strictly deterministic set of function calls. Agentic systems, however, exhibit immense variability in execution paths, tool selection, and memory retrieval patterns. A single user request may trigger 15 or more calls to various large language models, executing across dynamically chosen pathways and formulating intermediate logical steps that were never explicitly programmed. Without specialized decision tracking and reasoning path analysis, operations teams are left with opaque "black boxes" that are impossible to debug during production failures.

Multi-agent coordination bottlenecks

Modern production systems deploy variants of multi-agent architectures, such as the orchestrator-worker pattern—where a central intelligence delegates sub-tasks to specialized domain

experts—and reflexion, which incorporates meta-cognitive self-correction loops. While powerful, adding agents introduces severe coordination costs and complex failure modes. Critical observability challenges include monitoring for "silent worker failures" (where an agent crashes without reporting status, causing deadlocks) and "capability mismatches" (where an orchestrator assigns tasks that exceed a specific worker's tool scope).

To overcome these hurdles, the observability landscape is evolving:

- **Deep context tracing and state lineage** offer granular visibility and persistence for long-running workflows.
- **OpenTelemetry** standardization, specifically the gen_ai namespace, ensures consistent data analysis across platforms.
- **MCP** serves as an **observability bridge**, correlating behavior across tool execution and providing end-to-end visibility.
- **Evaluation-driven observability** and new **agent-specific SLOs** focus on runtime quality metrics like hallucination rates and token efficiency, moving beyond traditional uptime.

The open source community has driven OpenTelemetry (OTel) as the universal standard to mitigate the fragmentation caused by proprietary monitoring agents, the industry has coalesced around OpenTelemetry as the standard for AI telemetry collection.

- **Agentic AI relevant conventions**: OTel is actively being extended with several AI specific namespaces that provide a comprehensive ontology for the broad surface of interactions present in agentic applications. Gen AI semantic conventions (gen_ai.*) is the core namespace for AI-specific instrumentation that standardizes critical operational data such as the underlying model and model provider, operation names (e.g. chat vs. tool execution), and token usage for cost and performance monitoring. The AI agent namespace (ai.agent.*) is an emerging set of conventions for agentic frameworks geared towards capturing agent identity and context and tracking the steps in an agent's reasoning process. Tool and retrieval conventions track how agents interact with external tools and knowledge sources. These specialized namespaces are designed to be used along with general resource and context namespaces as well as common instrumentation libraries.
- **Cross-platform interoperability**: By standardizing data structures, OTel ensures that telemetry generated by diverse orchestration frameworks (like LangGraph or CrewAI) can be smoothly ingested and analyzed by various enterprise observability platforms (such as Datadog, Splunk, or Dynatrace) without data loss.

At the very foundation of achieving the right level of observability lies the infrastructure. Comprehensive OpenTelemetry instrumentation in the cloud infrastructure is critical to supporting the right level of observability to enable agentic applications.

- **Token metrics**: The cloud infrastructure should track AI-specific metrics like input vs. output token usage per request.
- **Cost attribution**: Requests should be tagged with metadata to track exactly which team or agent is responsible for inference costs, solving the "black box" spending problem.
- **Reasoning traces**: Observability tools visualize the agent's decision tree—showing not just that a call failed, but why the agent decided to make that call in the first place.

AgentOps and the quality flywheel

The cloud for the agentic era should adopt an "AgentOps" strategy that combines **observability** (capturing telemetry on what the agent did) with **evaluation** (assessing how well it did it). This creates a "quality flywheel" where production data feeds back into development for refinement.

Observability captures detailed telemetry from the agent's execution, offering insights into its call trajectory and decision-making, all while integrating with the rest of the application platform to be able to provide e2e agent/tools/models observability. Evaluation systematically assesses agent performance, quality, and safety against defined rubrics. This continuous observe-evaluate loop is fundamental throughout the agent lifecycle to refine prompts, advance context engineering, optimize cost, and maintain high-quality, reliable AI applications.

The impact on cost optimization of this quality flywheel should not be underestimated. Beyond enabling governance focused on optimizing application resource use, visibility into the agents' decisions enables the iteration required to fine tune the cost of running agentic applications. This process of fine tuning allows organizations to regain control of the economics of running the agentic applications and tame their perceived unpredictability.

Connectivity

Based on IDC's research[1], bandwidth needs at the network edge are expected to increase by 51% to support distributed AI workloads. The network must be prepared to handle this massive influx of data traffic generated by agents inferencing at the edge and communicating back to the core.

Cross-cloud considerations are critical because agentic applications are hosted across a variety of environments. Most organizations operate in a hybrid or multicloud model. The cloud network acts as a unifying overlay, providing cross-cloud abstraction consistency so that an agent in Google Cloud can securely call a tool on-premises or a model in another cloud without the developer managing the complex underlying networking plumbing.

To address the connectivity requirements for bandwidth and pervasive cross-cloud and edge connectivity, network connectivity must be ubiquitous, reliable, and deliver virtually unlimited capacity with predictable latency.

Latency is a critical consideration as it can create cognitive bottlenecks for agentic applications. Agents operate in loops: Perceive → Plan → Act → Reflect. Latency in any step of this loop—whether in inference or tool calling—accumulates, slowing down the agent's ability to reason and respond. The bulk of the latency is induced by the inference process and a smaller percentage is contributed by the network; the infrastructure must provide intelligent routing to minimize inference latencies. The infrastructure evolves to understand previously processed prompts to steer the traffic to accelerators where those prompt prefixes may already be cached and processing time can be minimized. The infrastructure also evolves to understand the prefill-decode stages of inference, accelerator specific metrics and route accordingly. For agents, tools, and models that may be distributed across different networks, connectivity latency may also contribute to the overall latency of the agent loop, however the latency due to connectivity is a small percentage of what is induced by the inference process. The infrastructure plays a critical role in addressing the most impactful sources of latency to optimize agent performance.

The open source community streamlines the efforts to optimize the performance of model serving through projects such as the Kubernetes Gateway API Inference Extension and the llm-d project. The infrastructure to deliver these optimizations relies on the combination of the open definition of these optimizations and open extensible data planes such as Envoy with its ext_proc filters and WASM plugins.

Google Cloud infrastructure: the starting point

The infrastructure delivered by Google Cloud today is built on the technologies and open source standards that will enable the imperatives for the successful production deployments of agentic applications moving forward. Google leverages many of these technologies and open source Standards to deliver AI services at scale today.

Google's planet scale network is designed, built, and ready to absorb the surge in bandwidth demand induced by the pursuit of agentic applications. The ubiquity of data center regions and points of presence around the globe guarantee minimal latency between agents, tools, and models across data centers that connect over a unique variety of interconnects and peerings offered by the Cross-Cloud Network.

Google Cloud employs Envoy pervasively as the dataplane to enable the service networking portfolio that includes Cloud Load Balancing, Secure Web Proxy, and Cloud Service Mesh.

The use of Envoy filters (ext_proc and ext_authz) enables the simple and extensible integration into the infrastructure of an ecosystem of first and third party functionality to enhance security, observability, and optimize traffic routing. Envoy filters (ext_proc) are instrumental in enabling managed implementations of serving optimization open standards proposed in the Kubernetes Gateway API Inference Extension and llm-d open source projects.

Google Cloud uses OpenTelemetry as the foundation for its instrumentation, telemetry, and observability. Adherence to an open standard in telemetry is fundamental to the successful integration of an ecosystem of observability tools and the standardization of operational practices that optimize performance, cost, and reliability.

Google Cloud managed workload identity is based on SPIFFE. In conjunction with Envoy filters authorization extensions (ext_authz), SPIFFE provides the necessary foundation for effective agent identity, authentication, and authorization across Certificate Authorities in different multicloud domains and on-prem PKI infrastructure.

Google Cloud is an active contributor to the different projects in the open source community where these standards are being defined. For instance, Google Cloud leads the charge to define Gateway APIs in the Kubernetes community to make infrastructure inference optimizations directly available to developers. These inference optimizations are also available as managed services in GKE.

The scale, ubiquity, and reliability of Google's existing network, coupled with the extensibility of its Envoy proxy infrastructure, make it the ideal crucible of connectivity to implement the level of IT governance that agentic applications require.

Conclusion

The stagnation in AI adoption—the "stalled engine"—is a clear signal that the current era of infrastructure has reached its limit. We cannot build the future of autonomy on primitives designed for static microservices. Buyers are acutely aware of this gap and are actively seeking solutions that are "ready" for the unique demands of agentic AI.

An augmented cloud infrastructure with native agentic capabilities is the answer to this imperative. By "shifting down" the complexity of connectivity, security, and governance into the platform, it frees developers to focus on the high-value work of reasoning and automation. The **cloud infrastructure**, with authoritative agentic resource registries, native fluency in agentic protocols like MCP and A2A, and a deep integration of identity and security, serves as the fundamental enabler of this new architecture.

For the enterprise, the transition to the agent-native cloud is enabled by an enhanced infrastructure. It is the bridge that spans the gap between the ambition of the agent-native enterprise and the reality of production-grade execution.

References

1. IDC. (2026). *2026 IDC Special Report on AI in Networking*.