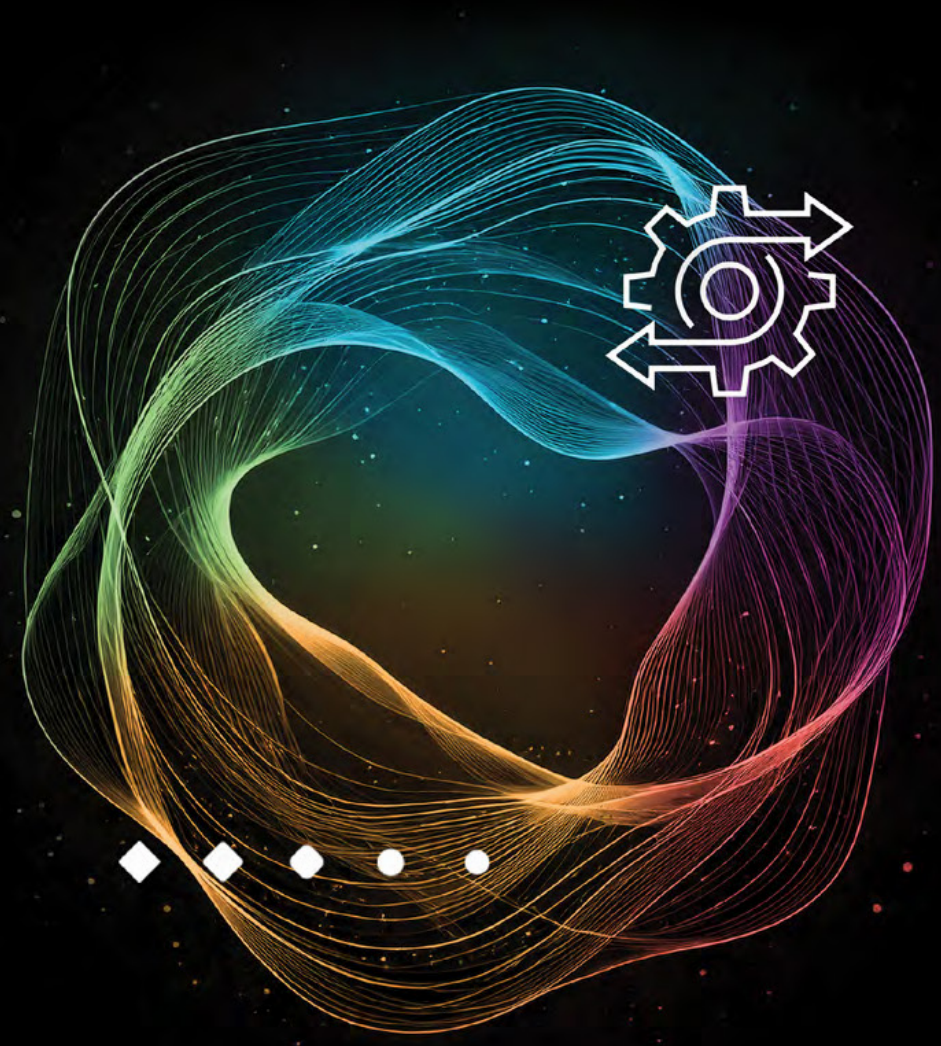
Deloitte.

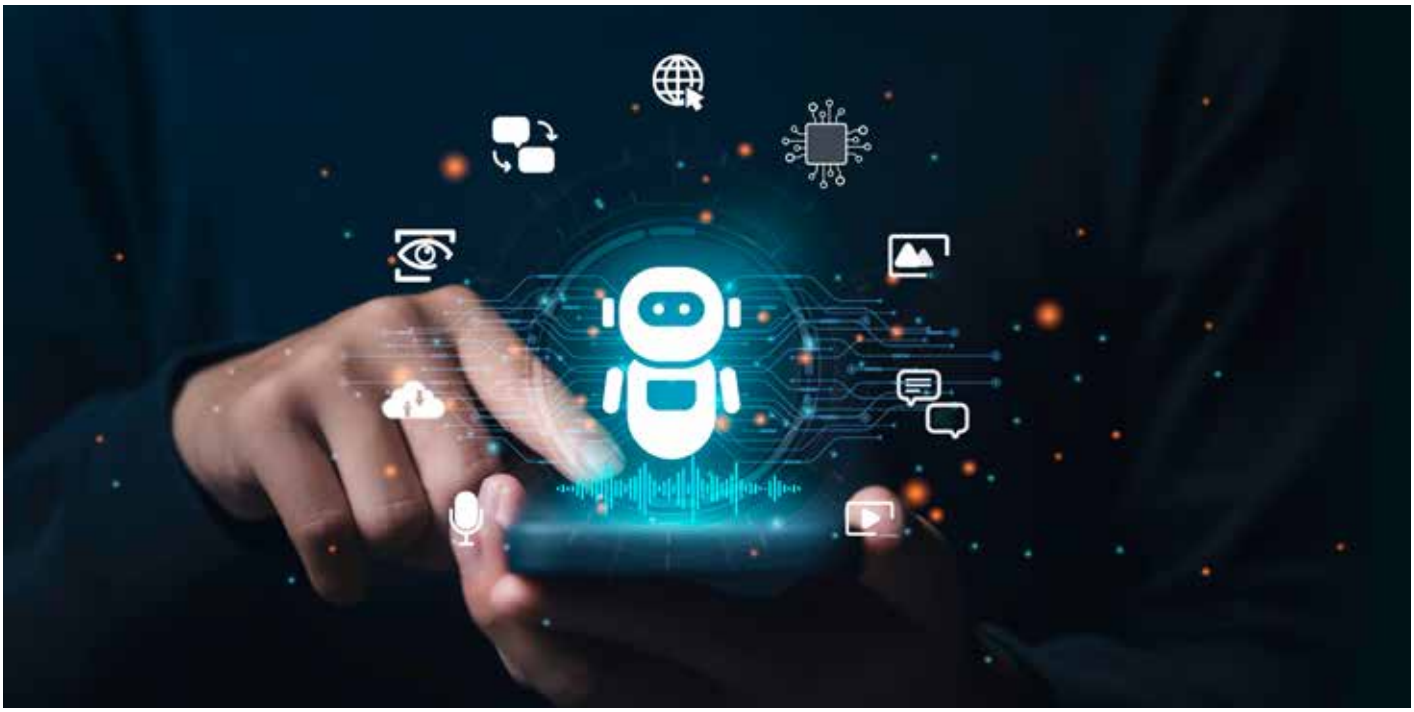


Evaluating agentic AI technology readiness

The business imperative for agentic AI

October 2025

Table of contents



Introduction

The agentic AI ecosystem has come a long way since the early days of Generative AI (GenAI), when outputs were generated by prompting a single Large Language Model (LLM). Today, it features a sophisticated architecture where autonomous agents collaborate, reason and interact seamlessly with enterprise systems.

Unlike traditional AI solutions bound by predefined tasks, agentic AI introduces a modular and extensible ecosystem. It comprises a layered stack of technologies, frameworks and communication protocols that enable enterprises to build goal-driven, context-aware and adaptive solutions. This ecosystem varies across organisations depending on their readiness to adopt and scale agentic AI. Depending on their current maturity, organisations face different technical requirements and adoption paths. For instance, organisations already using GenAI, automation and LLMs often operate in a multisystem.

These environments are supported by robust data pipelines, orchestration layers and integration frameworks that enable seamless operations and drive innovation.

In contrast, organisations that have yet to adopt GenAI or automation face a steeper path towards implementing agentic AI, as they must first build foundational capabilities and infrastructure to support such advanced systems.

For organisations with an existing tech ecosystem, the requirements may range from ensuring hyperscaler readiness to the ability to host serverless agent workloads to ensuring secure API gateways and CI/CD pipelines for agents. For those starting from scratch, the work must begin with building an in-house tech stack, investing in middleware and LLMOps, establishing data lakes and building APIs for core systems.

Agentic AI readiness checklist

Existing tech ecosystem	Requirements to be agentic AI-ready (Non-exhaustive)
If the organisation is already using automation or GenAI	• Hyperscaler readiness with enterprise-grade compute and auto-scaling • Container orchestration platform for agent workloads • Event-driven architecture with message queues and real-time streaming • API-first design with comprehensive gateway and security management • Vector databases for embeddings with session-based long-term memory • Zero-trust security model with IAM and API threat protection • Observability stack with AI-specific monitoring and audit trails • Multi-cloud strategy for resilience and compliance requirements • Unified data platform enabling seamless agent-to-system integration

Existing tech ecosystem	Requirements to be agentic AI-ready (Non-exhaustive)
If starting from scratch	• Cloud-native platform adoption with integrated AI services • AgentOps pipeline establishment with CI/CD for agent deployments • Data governance framework with quality management and lineage tracking • Enterprise security architecture with encryption and privacy controls • AI Centre of Excellence with cross-functional expertise and governance • Change management programme for workforce GenAI/agentic AI literacy and adoption • Scalable infrastructure design with microservices and API management • Risk management framework addressing AI ethics and compliance • Pilot programme strategy with measurable success metrics and KPIs

Key considerations for agentic AI technology readiness

Enterprises must focus on three essential aspects to prepare their technology ecosystem for agentic AI deployment. These include core components, key architectural patterns and agentic AI communication modes. The priority for enterprises at the early stages of their agentic AI journey should be strengthening the foundational core components and having basic architectural frameworks ready. In contrast, enterprises that have already achieved a degree of automation should shift their focus towards establishing robust agent communication protocols and implementing advanced/optimised architectural patterns.

1. Core components of the ecosystem

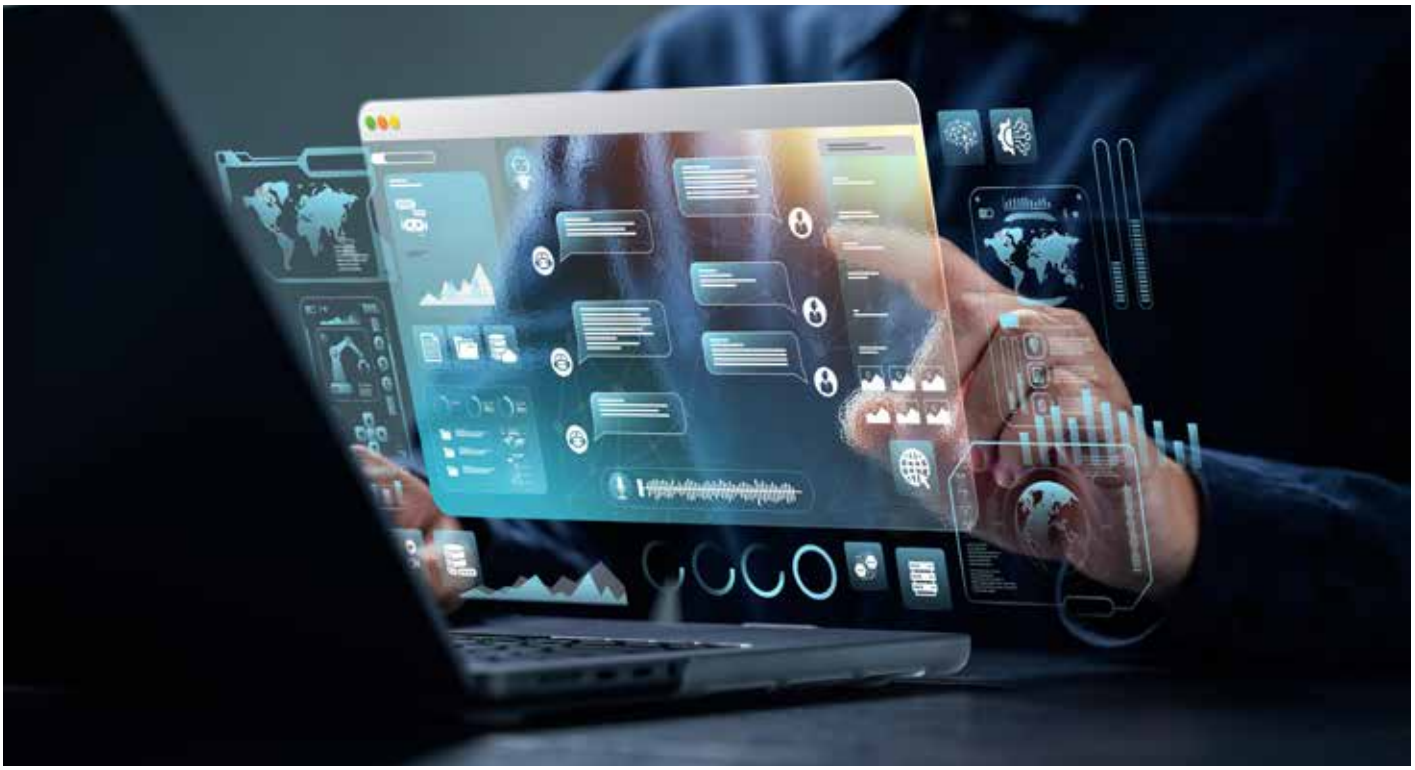
Foundation models: These models should be at the core of LLMs, which deliver the reasoning and generative capabilities that power agents. Increasingly, organisations at the early stage of the agentic AI journey have to fine-tune or adapt these models to reflect domain-specific language, data and governance requirements.

Orchestration and agent frameworks: Frameworks are required to harness the power of models. These frameworks can create, configure and connect agents, define workflows, manage agent memory and ensure agents can collaborate towards shared goals. Orchestration is where autonomy is transformed into purposeful action.

Knowledge and data layer: Agents require access to structured and contextual knowledge to operate effectively. Vector databases support this need by enabling reasoning across relationships and hierarchies within data. In addition, cloud-native data platforms are essential for ensuring the scalability, lineage and governance of enterprise datasets.

Tooling and integration layer: The true value of agentic AI emerges when agents interact with enterprise systems. APIs, RPA connectors and native agentic platforms bridge this gap. It might require hyperscalers who bring capabilities such as orchestration into business process automation, native agentic frameworks and embedding AI agents into core processes, thereby increasing its functionality.

Development and operations layer: Agentic AI is not a "build once" capability. Continuous development, deployment and monitoring are essential for building model pipelines and are increasingly important for maintaining operational oversight as AI systems evolve. Complementing these are evaluation and observability tools that measure LLM quality, reliability and drift. These capabilities ensure agentic AI systems remain trustworthy and effective at scale.



2. Architectural patterns in agentic AI

As enterprises experiment with Agentic AI, several architectural patterns are beginning to crystallise. It is important to employ:

Multi-agent systems: Instead of relying on a single large agent, organisations need to distribute tasks into specialised agents, such as extraction agents, vetting agents or orchestration agents, that collaborate to complete complex workflows.

Human-in-the-loop: Even with automation, many enterprise scenarios still require human oversight. Human-in-the-loop design ensures that exceptions, ambiguity or ethical decisions are escalated to people, helping build trust in AI-driven processes.

Memory management: Effective agentic AI systems need to manage both short-term and long-term information. Short-term context windows allow agents to respond accurately within a single interaction, while long-term enterprise memory ensures continuity across sessions, preserving knowledge and enabling more intelligent, consistent decision-making over time.

Hybrid reasoning: Combining LLMS's probabilistic reasoning with deterministic rules and symbolic knowledge graphs enhances accuracy and reliability for enterprise-grade applications.

Overall, agentic AI is no longer confined to proof-of-concept pilots. Enterprises across industries are now building concrete applications by using core components and architectural patterns. AI agents are actively orchestrating incident resolution in IT and cloud operations, managing playbooks within ITSM systems, and driving automation in DataOps pipelines.

3. Agentic AI communication modes

While all the components and architectural patterns can be in place, how they coordinate and communicate with each other is an important aspect that needs to be considered while ensuring tech readiness.

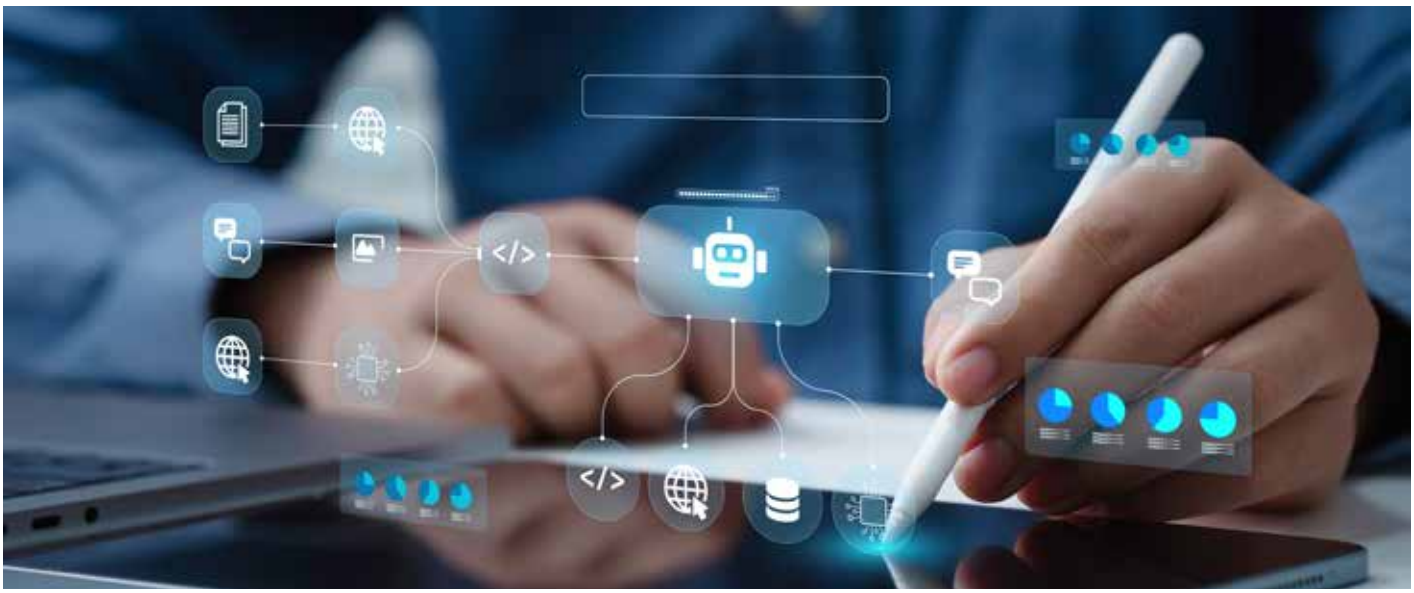
Some ways agentic AI communication can be enabled are Model Context Protocol (MCP), Agent-to-Agent Communication (A2A) and Agent Communication Protocol (ACP).

MCP
One of the most critical emerging standards in the ecosystem is the MCP. MCP is designed to simplify and standardise how models communicate with external tools, databases and APIs. Instead of requiring bespoke integrations for every new service, MCP creates a consistent layer that allows models to access external functionality in a controlled and auditable manner.

While MCP is sometimes confused with agent-to-agent communication, its role is distinct. It focuses on model-to-tool interoperability, for example, enabling an LLM-powered agent to seamlessly retrieve information from a corporate database, invoke a financial calculation API or call an external workflow. In doing so, MCP reduces development complexity, enhances extensibility and accelerates enterprise adoption. As enterprises increasingly work across multiple vendors and ecosystems, MCP will play a pivotal role in ensuring interoperability and avoiding lock-in.

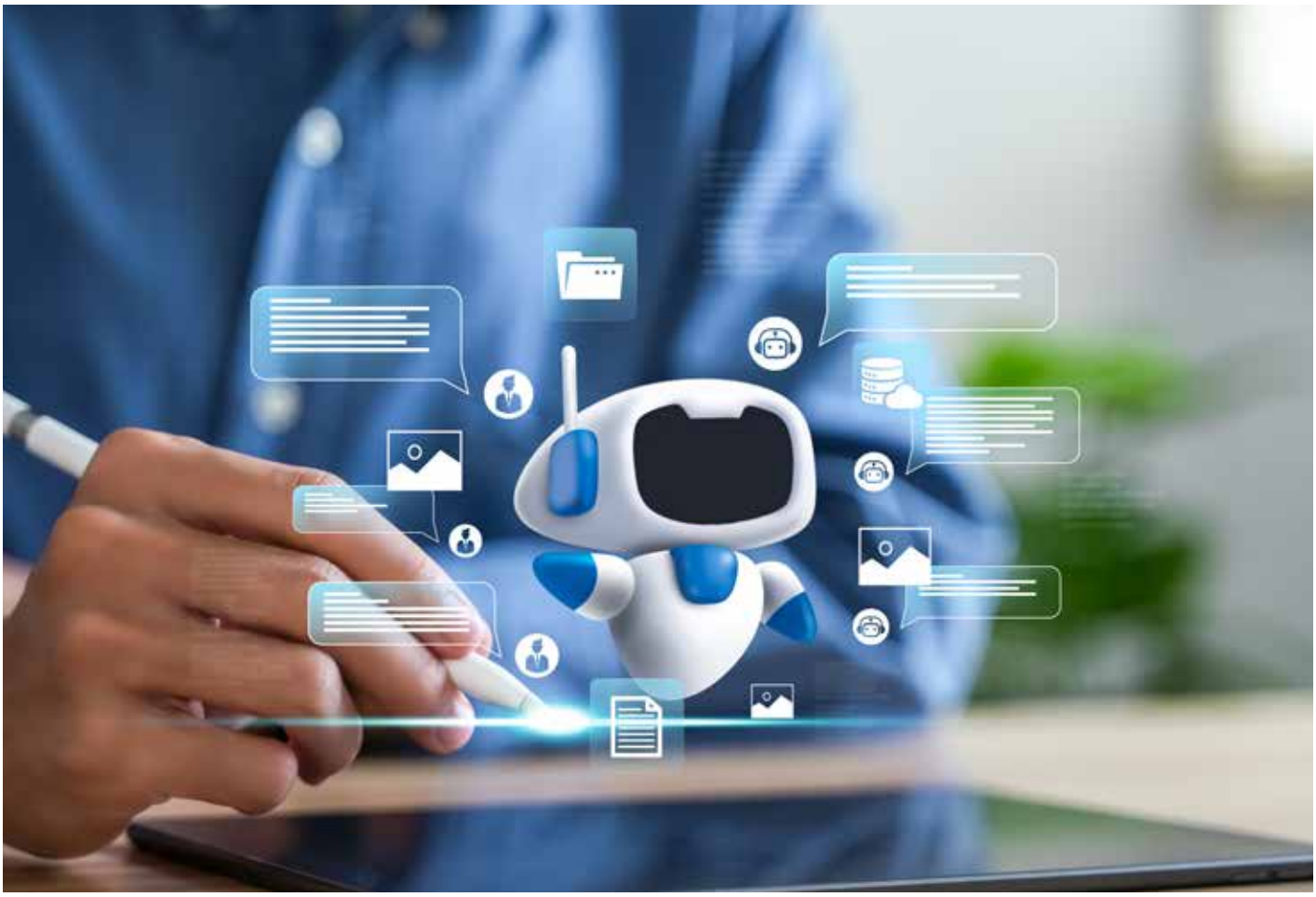
A2A
Agent-to-agent communication allows the exchange of information between two autonomous digital entities or systems. These agents collaborate to perform tasks, share data or coordinate actions, typically without direct human intervention. This form of communication is foundational in distributed systems and automated workflows.

ACP
Agent Communication Protocol (ACP) is an open protocol for agent interoperability that solves the growing challenge of connecting AI agents, applications and humans. Modern AI agents are often built in isolation across different frameworks, teams and infrastructures. This fragmentation slows innovation and makes it harder for agents to work together effectively. ACP solves this by enabling agents to communicate through a standardised RESTful API.



Key differences between MCP, A2A and ACP

	MCP	A2A	ACP
Transport and communication	JSON-RPC is for protocol communication with stdio (local). HTTP + Server-Sent Events (SSE) or Streamable HTTP (remote)	JSON-RPC 2.0 inside HTTP POST requests, creating a multi-layered approach. All operations use Power-On Self-Test (POST)	REST-first approach, using standard HTTP verbs (GET, POST, DELETE) that make the protocol intuitive for developers
State management	Inherently stateless at the protocol level. Individual servers must manage their own state to maintain context and continuity	Agent and client-level state management	Three states: session level (context ID), agent level (internal) and task level (TaskStore)
Discoverability	No standardised discovery. Manual discovery through host application configuration files (e.g., claude_desktop_config. json)	Agent Cards are JSON metadata files hosted at known URLs, enabling both online and registry-based agent discovery	Agent metadata in Agent Decorator. Dedicated endpoints (online) and Docker Registry (offline)
Message structure	Structured function calls; JSON payloads for tool invocation	Structured with three explicitly defined message part types (TextPart, FilePart and DataPart), requiring protocol updates for new content types	MIME types for content identification working without protocol updates
Development complexity	Low; minimal server file with decorators for tools, resources or prompts	High; requires agent logic, an agent executor and a main server file, creating more initial complexity	Medium; needs just a base agent file with the @server. agent decorator, Docker images for offline discovery



Conclusion

The agentic AI ecosystem brings immense opportunities. Enterprises can use mature cloud platforms, SaaS providers and orchestration frameworks to build intelligent, autonomous systems that transform productivity and decision-making. Alliances with hyperscalers and automation vendors accelerate adoption by providing ready-made platforms and reducing integration friction.

Yet limitations remain. The ecosystem is still fragmented, with multiple frameworks competing for dominance. Interoperability challenges persist, making standards such as MCP critical. Observability and evaluation require continued investment, as enterprises must balance innovation with governance. Finally, the cost and complexity of orchestrating multi-agent workflows at scale demand robust architecture and disciplined operations.

The agentic AI technology ecosystem forms the foundation for real-world adoption. It includes models, frameworks, data layers, integrations and operational tools. Each is essential, but none is sufficient on its own.

As subsequent chapters will explore, technology is only part of the story. Measuring success, scaling initiatives, aligning employees and deploying responsibly are equally essential. However, without a robust technology ecosystem as described here, the promise of agentic AI cannot be fully realised.

Connect with us

Ashvin Vellody
Partner
Deloitte India
ashvinv@deloitte.com

Moumita Sarker
Partner
Deloitte India
msarker@deloitte.com

Contributors

Mohammed Riyaz Ahmed
Pradeep Surabattula
Sagarika Gupta
Shailesh Gaikar
Srishti Deoras

Acknowledgements

Neha Kumari
Ruchira Thakur

Deloitte.