What Are Multi-Agent Systems?

Multi-agent systems—also known as teams of agents—are a collection of specialized AI agents that work together to solve a complex problem. Each agent has a specific role in executing varied tasks that contribute to achieving a common goal.

How Do Multi-Agent Systems Work?

Agentic AI is the next evolution of AI, introducing key features like planning, reasoning, contextual memory, and the use of tools to autonomously facilitate complex workflows with minimal human input. Also called AI agents, this technology relies on advanced reasoning to successfully navigate complex business scenarios.

Multi-agent systems work by having multiple AI agents collaborate to achieve common goals. Each AI agent has a level of autonomy, specialized capabilities, and a local view of the system. More notably, these systems are specifically designed to handle intricate tasks while balancing multiple dependencies.

How a team of AI agents function when a user request is inputted.

Autonomous agent operations can be integrated into intertwined workflows that involve human touchpoints, decision trees, and parallel workstreams.

For example, in end-to-end software development, developers, technical support, and technical documentation writers must support customers with questions and requests related to production-grade applications, while balancing their time to meet service-level agreements (SLAs). For maximum productivity gains, a team of agents can be designed to:

- Respond to bug requests using natural language processing and ask the user clarifying questions to reduce the resources required to review high volumes of bugs

- Reference and analyze past bugs with similarity matching and automatically create a new bug ticket for human review to streamline priority assignments
- Provide engineering assistance by generating code suggestions, orchestrating code review, and reproducing test cases for a human to verify and integrate

Multi-agent systems can be safeguarded by adding AI guardrails to prevent unexpected results. This closely models how development teams typically operate within the modern workplace.

💡**Key Takeaway:** Multi-agent systems work by performing higher-order planning, reasoning, and orchestration. Teams of AI agents engage in natural language conversations, handle complex tasks, and support human teams with decision making and task completion.

What Are the Benefits of Using Multi-Agent Systems?

While a single AI agent can execute many different tasks, a team of AI agents can achieve far more by communicating information and taking appropriate action to achieve a common goal.

As demonstrated in the bug-management example, multi-agent systems have the greatest impact on organizations that:

- Experience demanding, growing workloads, such as managing end-to-end software workflows, providing customer service in global telecom operations, and managing patient healthcare
- Face rapidly changing environments, such as market volatility in financial services, retail distribution, and supply chains
- Must monitor and distribute control intelligence, such as autonomous vehicles for safety and intelligent traffic systems in smart cities
- Require essential fault tolerance, such as coordinating disaster response and managing power supply grids

By using the combined knowledge and decision-making of multiple AI agents, organizations can become more efficient at solving complex problems while maintaining accuracy and security. This holds true for organizations that must tackle issues that are too sophisticated for traditional, centralized systems.

💡**Key Takeaway:** Multi-agent systems consist of many agents tailored for specialized or niche tasks, leading to overall greater efficiency and performance. These AI agents can also be customized and fine-tuned to adapt to changing requirements. Multi-agent systems are scalable and transparent, as systems don't require a complete overhaul or retraining—individual agents can be replaced or updated.

What's the Role of AI Agent Orchestration in Multi-Agent Systems?

AI agent orchestration is the process of enabling multiple agents or tools that would typically operate independently to work together toward a common goal. This coordination allows the multi-agent system to manage and execute more complex tasks efficiently.

There are several ways to orchestrate a team of AI agents:

Orchestration Type	Description	Advantages	Challenges	Use Case Example
Centralized	A single supervisor agent coordinates tasks, data flow, and decision-making.	Clear control Simplified management Consistency in decisions	Potential bottlenecks Less adaptable to dynamic systems	Customer relationship management (CRM)

Decentralized	Each agent operates autonomously, sharing information with others.	High flexibility Adaptable to dynamic environments	Requires sophisticated communication protocols Higher complexity	Swarm drones for real-time deliveries
Federated	Multiple agent systems collaborate across organizations with shared protocols.	Facilitates cross-system collaboration Leverages system strengths	Relies heavily on interoperability and shared standards	Supply chain collaboration between firms
Hierarchical	Higher-level agents supervise lower-level agents in a tiered structure.	Balances flexibility and oversight Ideal for complex systems	Coordination across layers can be complex Potential dependency delays	Industrial automation with layered control

Think of orchestration as a control framework for multi-agent systems. Orchestration is foundational for achieving scalability, efficiency, and adaptability in multi-agent systems. By enabling agents to collaborate and share resources effectively, orchestration supports:

- **Dynamic Problem-Solving:** Adapting to changing conditions or unexpected challenges
- **Improving Resource Utilization:** Optimizing how agents access and use tools and data

- **Enhancing System Reliability:** Reducing conflicts and ensuring consistent outcomes
- **System Access Controls:** Limiting actions agents can execute on to reduce the risk of errors and misuse
- **Protecting Information:** Ensuring only authorized agents and users can access or handle sensitive data

Agent orchestration is critical for industries such as logistics, autonomous systems, cybersecurity, and enterprise automation, where seamless multi-agent collaboration is a key to success.

What Are Tips for Building Multi-Agent Systems?

When designing a multi-agent system, factors such as telemetry, logging, and evaluation are imperative for increasing the accuracy of responses and improving business outcomes.

- Interoperability and agent communication so that agents can be written using the agent framework best suited for the task but can still work together as a team
- Monitoring and performance optimization, which involve system configuration for fine-tuning real-time data collection and processing and telemetry for reporting on system health metrics
- Evaluation and observability for debugging, traceability, and auditing

AI agent frameworks are specialized development platforms or libraries that streamline the process of building, deploying, and managing AI agents. To complement popular agent frameworks, NVIDIA's AI software solutions are open source.

By abstracting the complexity of creating agentic AI systems, developers can hone in on fine-tuning their applications and updating agent behaviors. Less time is spent on technical implementation, freeing developers to focus on refinement to meet business needs.

How Does Agentic RAG Enhance Your Multi-Agent System?

Data powers modern enterprise applications, but the magnitude and scale of the data have made it too expensive and time-consuming to use effectively. As a result, most generative AI applications leverage a corpus of data that is relatively small compared to the amount of proprietary knowledge being stored and generated.

To thrive in the AI era, workforces must be connected to enterprise knowledge, and doing so requires the use of vast amounts of data. This isn't possible with traditional computing and data processing techniques.

Every enterprise will need agentic retrieval-augmented generation (RAG). Implementing agentic RAG can connect teams of agents to enterprise knowledge. Multi-agent systems that can perceive, reason, and act will turn that knowledge into action to solve problems.

To give AI agents access to large amounts of diverse data, they need an accelerated AI query engine that efficiently processes, stores, and retrieves data to augment generative AI model inputs. RAG is commonly used today.

Unlike traditional metadata analysis, which only reveals surface-level details like authorship and time stamps, AI can ingest and interpret the full content of data. This enables a deeper understanding of context, meaning, and patterns within the information itself.

Agentic RAG Capabilities

Agentic RAG must be able to:

1. Access knowledge across structured, semi-structured, and unstructured data and metadata sources, including text, PDF, image, video, and specialty data types
2. Efficiently process data at petabyte scale, so all knowledge is quickly available to generative AI-powered applications and agents

3. Provide high-accuracy, high-performance retrieval and reranking of knowledge from multiple sources to efficiently augment inputs into generative AI models
4. Store and leverage learnings from AI-powered applications and agents in production, automatically increasing the knowledge of the enterprise and creating an AI data flywheel

Open-source and proprietary agent frameworks can be helpful when building agentic workflows, however, they may not work together seamlessly. NVIDIA NeMo Agent toolkit is an open-source library for connecting, evaluating, and accelerating complex agentic AI systems. It includes technology building blocks to ease the development and measurement of full-stack, enterprise-ready agentic systems that connect AI to data via a collection of reusable tools.

NVIDIA AI Blueprints provide a starting point for developing agents to address specific use cases, including RAG. The blueprints contain example applications, reference codes, sample data, tools, and documentation. Enterprises can build and operationalize custom AI applications—creating data-driven AI flywheels—using these blueprints.

Components of NVIDIA AI Enterprise that help you build agentic systems include:

- **NVIDIA NeMo™:** A set of microservices that help agentic AI developers easily curate data at scale, customize agents with popular fine-tuning techniques, evaluate them on standard and custom benchmarks, and guardrail them for appropriate and grounded outputs.
- **NVIDIA NeMo Retriever:** A collection of microservices for data ingestion, extraction, embedding, retrieval, and reranking that connect custom models to diverse business data and deliver highly accurate responses. NeMo Retriever embedding models are world-class, with multiple wins on the MTEB leaderboard for retrieval accuracy.
- **NVIDIA NIM™:** A set of inference microservices, optimized for leading open generative AI models, that provides up to 3x improved efficiency out of the box.