

What is a multi-agent system?

Introduction

A multi-agent system (MAS) consists of multiple [artificial intelligence \(AI\) agents](#) working collectively to perform tasks on behalf of a user or another system.

Each agent within a MAS has individual properties but all agents behave collaboratively to lead to desired global properties.¹ Multi-agent systems are valuable in completing large-scale, complex tasks that can encompass hundreds, if not thousands, of agents.²

Central to this idea are [artificial intelligence \(AI\) agents](#). An AI agent refers to a system or program capable of autonomously performing tasks on behalf of a user or another system by designing its workflow and using available tools. At the core of AI agents are [large language models \(LLMs\)](#). These intelligent agents leverage the advanced natural language processing techniques of LLMs to comprehend and respond to user inputs. Agents work through problems step-by-step and determine when to call on external tools. What differentiates AI agents from traditional LLMs is the use of tools and the ability to design a plan of action. The tools available to an agent can include external datasets, web searches and application programming interfaces (APIs). Similarly to human decision-making, AI agents can also update their memory as they acquire new information. The information-sharing, tool usage and adaptive learning allow AI agents to be more general purpose than traditional LLMs.

For more information about single agent systems, see our detailed [AI agent content](#).

Single agent versus multi-agent systems

Single agent intelligent systems engage with their environment to autonomously plan, call tools and produce responses. The tools made available to an agent provide information that is otherwise unavailable to the agent. As previously described, this information can be a database acquired through an API or another agent. There is a distinction here between single and multi-agent systems. When calling another agent as a tool, that secondary agent is part of the original agent's environmental stimuli. That information is acquired and no further cooperation takes place. Whereas multi-agent systems differ by involving all agents within the environment to model each other's goals, memory and plan of action.⁴ Communication between agents can be direct or indirect through altering the shared environment.

Each entity within a multi-agent system is an autonomous agent to some extent. This autonomy is typically seen by the agent's planning, tool calling and general reasoning. In a multi-agent system, agents remain autonomous but also cooperate and coordinate in agent structures.³ To solve complex problems, agent communication and distributed problem-solving are key. This type of agent interaction can be described as multi-agent [reinforcement learning](#). The information shared through this form of learning can include instantaneous information acquired through sensors or actions. Additionally, an agent's experiences in the form of episodic information can be shared. These episodes can be sequences of sensations, actions and learned policies. Finally, agents can share their experiences in real-time to prevent other agents from repetitively learning the same policies.⁵

Individual agents are powerful on their own. They can create subtasks, use tools and learn through their interactions. The collective behavior of multi-agent systems increases the potential for accuracy,

adaptability and scalability. Multi-agent systems tend to outperform single-agent systems due to the larger pool of shared resources, optimization and automation. Instead of multiple agents learning the same policies, one can share learned experiences to optimize time complexity and efficiency.⁵

Industry newsletter

The latest AI trends, brought to you by experts

Get curated insights on the most important—and intriguing—AI news. Subscribe to our weekly Think newsletter. See the [IBM Privacy Statement](#).

We use your email to validate you are who you say you are, to create your IBMid, and to contact you for account related matters.

Business email

Your subscription will be delivered in English. You will find an unsubscribe link in every newsletter. You can manage your subscriptions or unsubscribe [here](#). Refer to our [IBM Privacy Statement](#) for more information.

Your subscription will be delivered in English. You will find an unsubscribe link in every newsletter. You can manage your subscriptions or unsubscribe [here](#). Refer to our [IBM Privacy Statement](#) for more information.

Architectures of multi-agent systems

Centralized networks

Multi-agent systems can operate under various architectures. In centralized networks, a central unit contains the global knowledge base, connects the agents and oversees their information. A strength of this structure is the ease of communication between agents and uniform knowledge. A weakness of the centrality is the dependence on the central unit; if it fails, the entire system of agents fails.⁶

Decentralized networks

Agents in decentralized networks share information with their neighboring agents instead of a global knowledge base. Some benefits of decentralized networks are robustness and modularity. The failure of one agent does not cause the overall system to fail since there is no central unit. One challenge of decentralized agents is coordinating their behavior to benefit other cooperating agents.⁷

Structures of multi-agent systems

There are also many ways of organizing agents within a multi-agent system including:

Hierarchical structure

A hierarchical structure is tree-like and contains agents with varying levels of autonomy. Within a simple hierarchical structure, one agent can have the decision-making authority. In a uniform

hierarchical structure, the responsibility can be distributed among multiple agents.⁸

Holonic structure

Within this architecture type, agents are grouped into holarchies. A holon is an entity that cannot operate without its components. For instance, the human body is a holon because it cannot function without working organs.⁹ Similarly, in holonic multi-agent systems, the leading agent can have multiple subagents while appearing to be a singular entity.⁸ These subagents can also play roles in other holons. These hierarchical structures are self-organized and created to achieve a goal through the collaboration of the subagents.

Coalition structure

Coalitions are helpful in cases of underperforming singular agents in a group. In these situations, agents temporarily unite to boost utility or performance. Once the desired performance is reached, the coalitions are disbursed. It can become difficult to maintain these coalitions in dynamic environments. Regrouping is often necessary to enhance performance.⁹

Teams

Teams are similar in structure to coalitions. In teams, agents cooperate to improve the performance of the group. Agents in teams do not work independently, unlike in coalitions. Agents in teams are much more dependent on one another and their structure is more hierarchical than coalitions.⁸

Behaviors of multi-agent systems

The behaviors of agents within a multi-agent system often reflect behaviors occurring in nature. The following agent behaviors can apply to both multisoftware and multirobot agents.

Flocking

The collective behavior seen in multi-agent systems can resemble that of birds, fish and humans. In these systems, agents share an objective and require some organization to coordinate their behavior. Flocking pertains to directional synchronization and the structure of these flocks can be described by these heuristics:¹⁰

- Separation: attempt to avoid collision with nearby agents.
- Alignment: attempt to match the velocity of nearby agents.
- Cohesion: attempt to remain close to other agents.

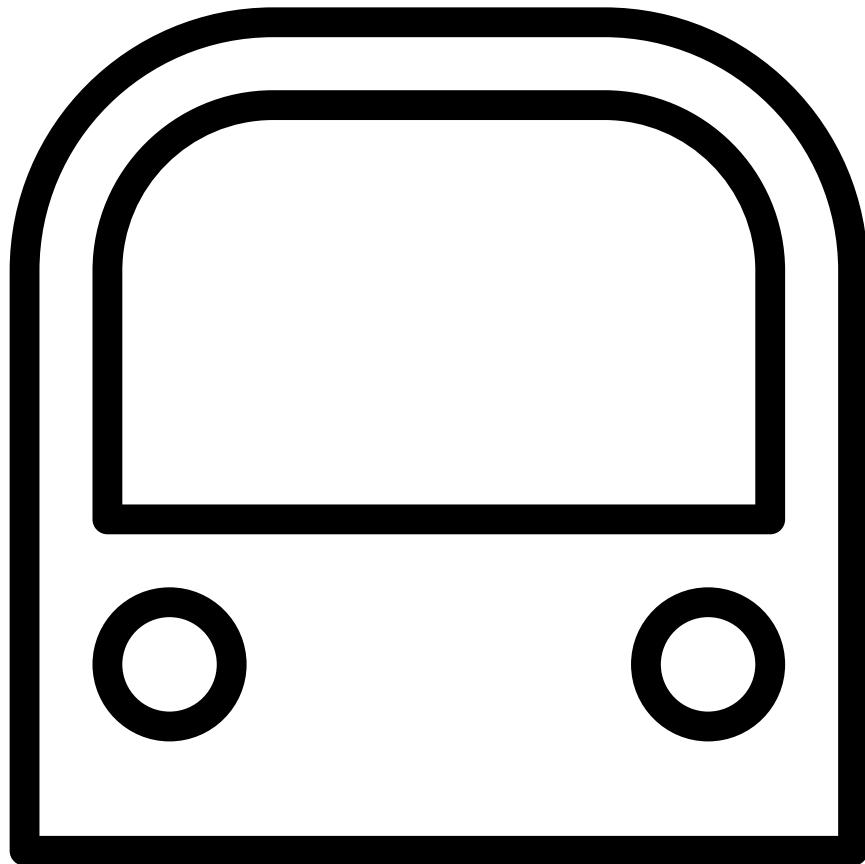
In the context of software agents, this coordination is crucial for multi-agent systems managing transportation networks such as railroad systems.

Swarming

The spatial positioning of agents in a multi-agent system can be compared to the swarming that occurs in nature. For instance, birds fly in sync by adjusting to neighboring birds. From a technical perspective, swarming is the emergent self-organization and aggregation among software agents with decentralized control.¹¹ A benefit of swarming is that one operator can be trained to manage a swarm of agents. This method is less computationally expensive and more reliable than training an operator for each agent.¹²

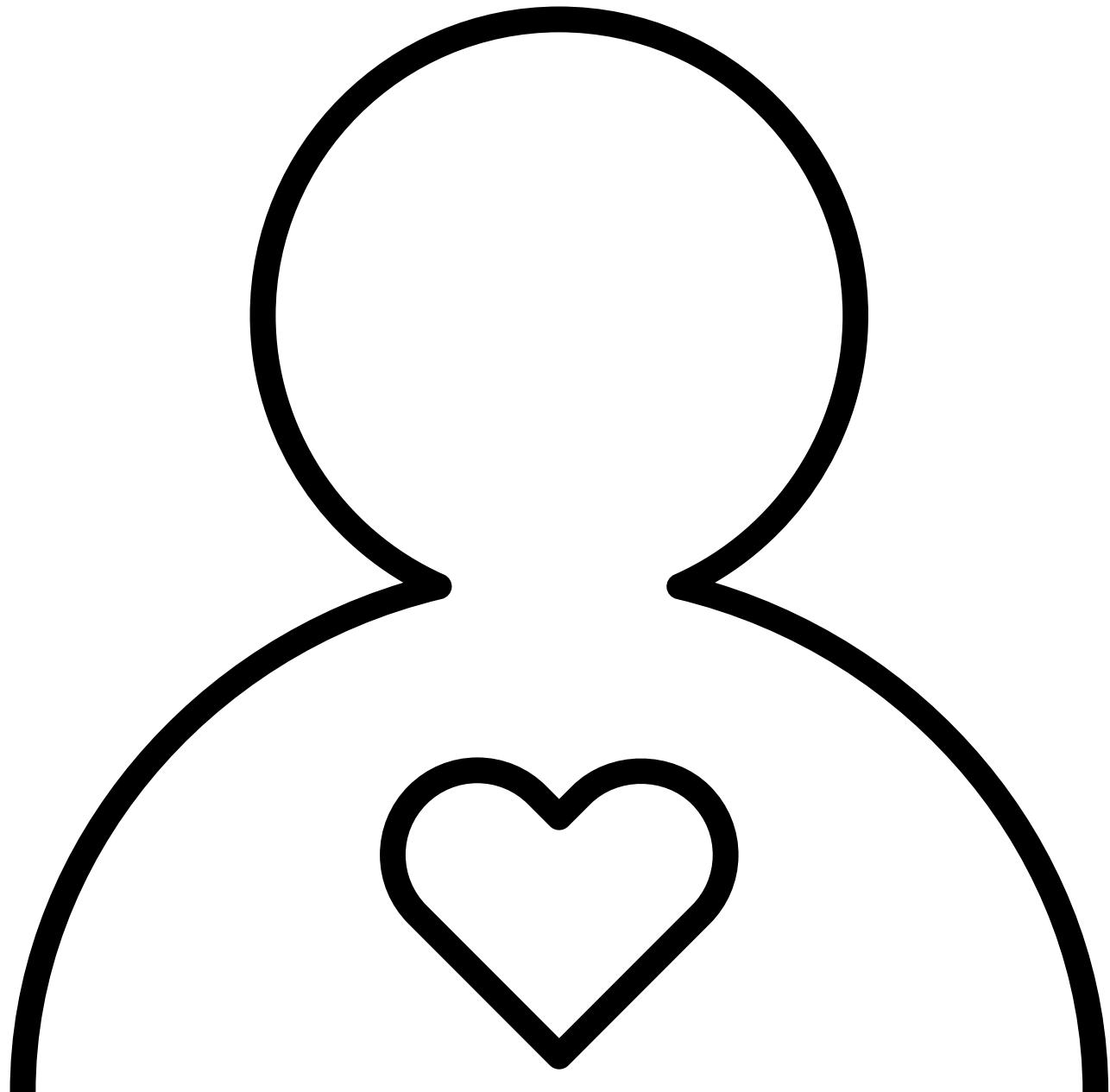
Use cases of multi-agent systems

Multi-agent systems can solve many complex, real-world tasks. Some examples of applicable domains include:



Transportation

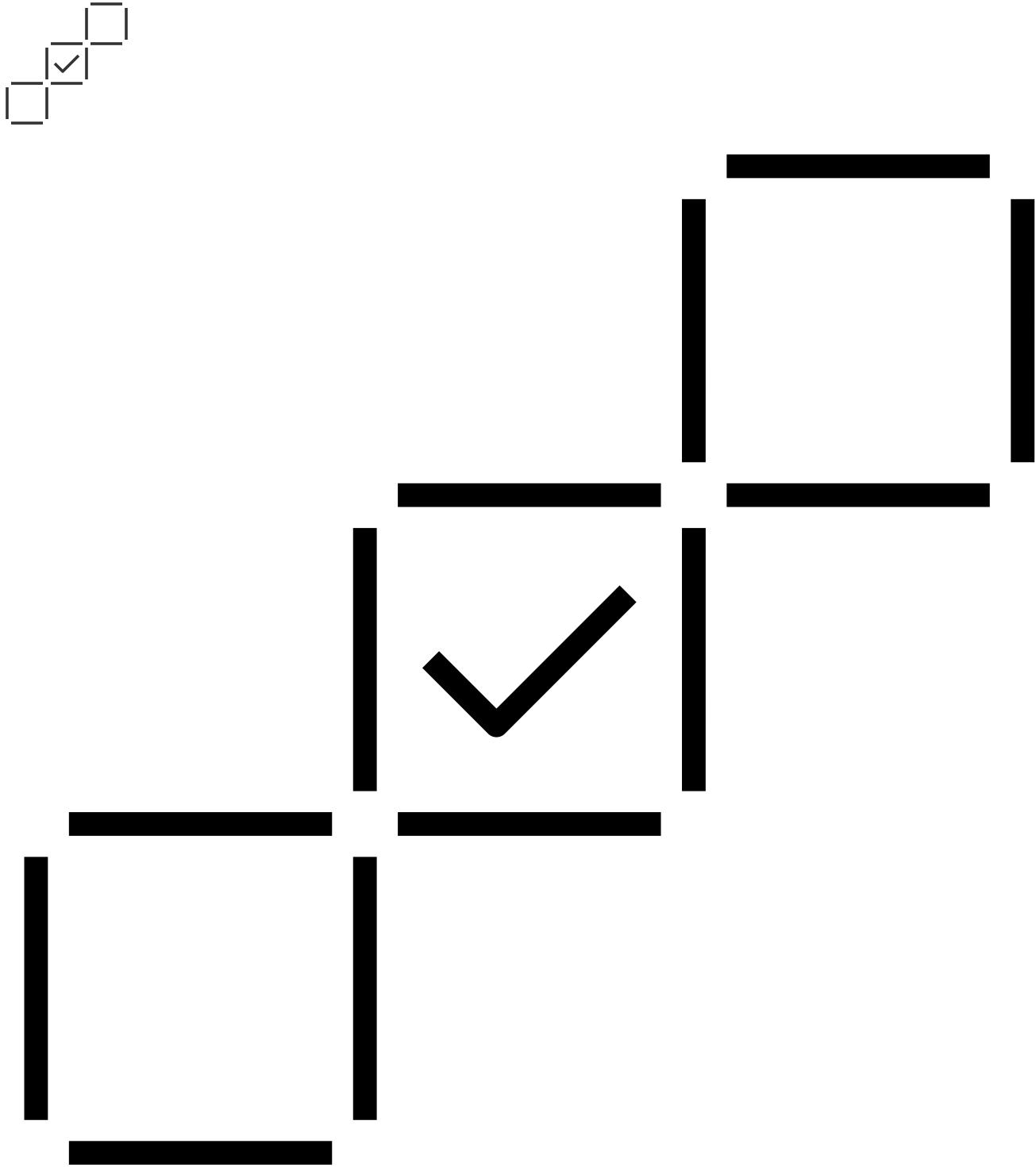
Multi-agent systems can be used to manage transportation systems. The qualities of multi-agent systems that allow for the coordination of complex transportation systems are communication, collaboration, planning and real-time information access. Examples of distributed systems that might benefit from MAS are railroad systems, truck assignments and marine vessels visiting the same ports.¹³



Healthcare and public health

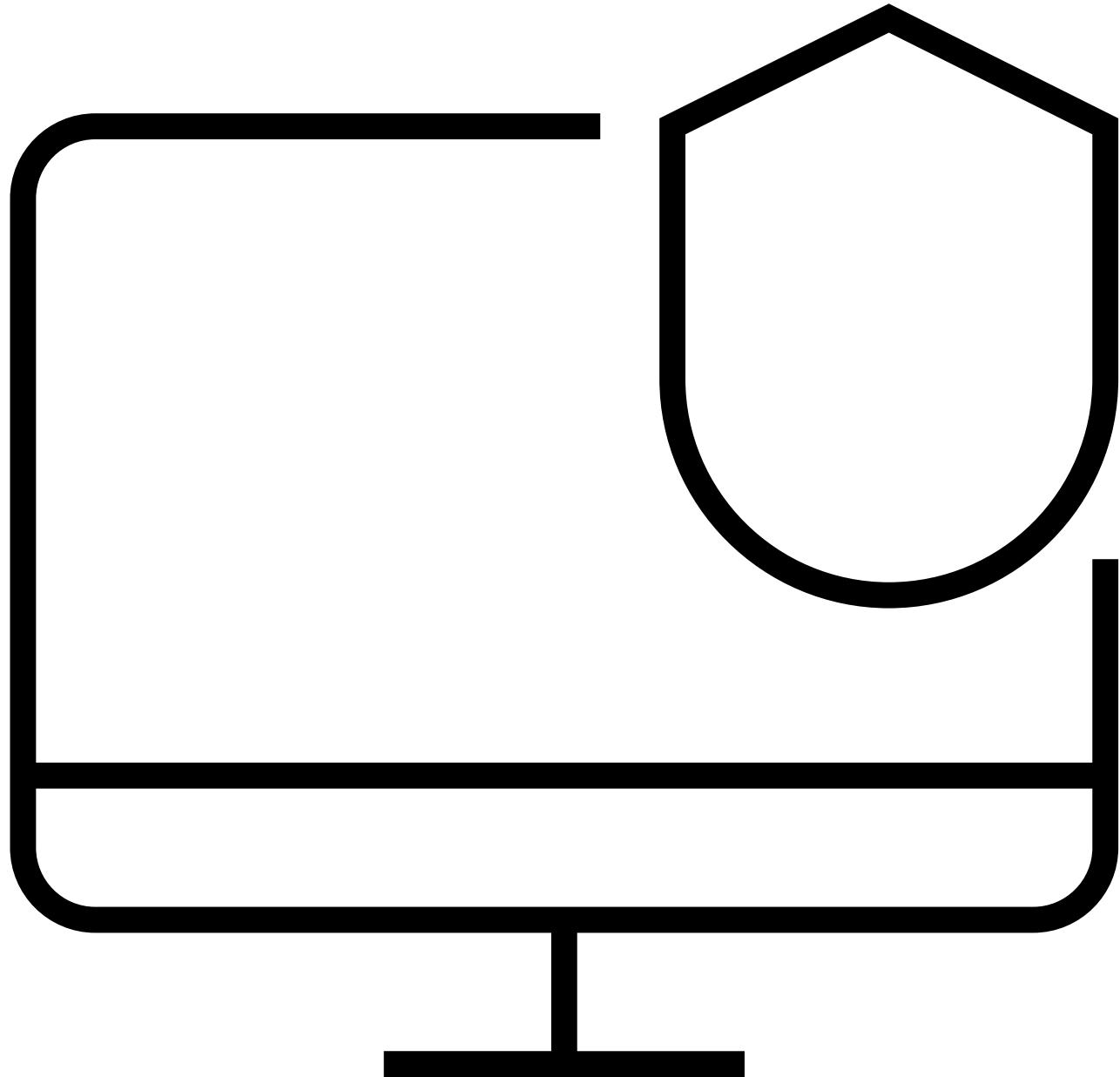
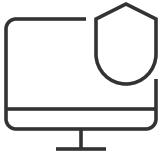
Multi-agent systems can be used for various specific tasks in the healthcare field. These agent-based systems can aid in disease prediction and prevention through genetic analysis. Medical research about cancer might be one application.¹⁴ In addition, multi-agent systems can serve as tools for preventing

and simulating epidemic spread. This forecasting is made possible by using epidemiologically informed neural networks and **machine learning (ML)** techniques to manage large datasets. These findings can affect public health and public policy.¹⁵



Supply chain management

Numerous factors affect a supply chain. These factors range from the creation of goods to the consumer purchase. Multi-agent systems can use their vast informational resources, versatility and scalability to connect the components of **supply chain management**. To best navigate this **intelligent automation**, **virtual agents** should negotiate with one another. This negotiation is important for agents collaborating with other agents that have conflicting goals.¹⁶



Defense systems

Multi-agent systems can aid in strengthening defense systems. Potential threats can include both physical national security issues and cyberattacks. Multi-agent systems can use their tools to simulate potential attacks. One example is a maritime attack simulation. This scenario would involve agents working in teams to capture the interactions between encroaching terrorist boats and defense vessels.¹⁷ Also, by working in cooperative teams, agents can monitor different areas of the network to detect incoming threats such as **distributed denial of service (DDoS)** flooding attacks.¹⁸

Advantages of multi-agent systems

There are several characteristics of multi-agent systems that provide advantages including:

Flexibility

Multi-agent systems can adjust to varying environments by adding, removing or adapting agents.

Scalability

The cooperation of several agents allows for a greater pool of shared information. This collaboration allows multi-agent systems to solve more complex problems and tasks than single-agent systems.

Domain specialization

Single agent systems require one agent to perform tasks in various domains, whereas each agent in a multi-agent system can hold specific domain expertise.

Greater performance

Multi-agent frameworks tend to outperform singular agents.¹⁹ This is because the more action plans are available to an agent, the more learning and reflection occur. An AI agent incorporating knowledge and feedback from other AI agents with specialties in related areas can be useful for information synthesis. This backend collaboration of AI agents and the ability to fill information gaps are unique to agentic frameworks, making them a powerful tool and a meaningful advancement in artificial intelligence.

Challenges of multi-agent systems

There are several challenges in designing and implementing multi-agent systems including:

Agent malfunctions

Multi-agent systems built on the same [foundation models](#) can experience shared pitfalls. Such weaknesses might cause a system-wide failure of all involved agents or expose vulnerability to adverse attacks.²⁰ This highlights the importance of data governance in building foundation models and the need for thorough training and testing processes.

Coordination complexity

One of the greatest challenges with building multi-agent systems is developing agents that can coordinate and negotiate with one another. This cooperation is essential for a functioning multi-agent system.

Unpredictable behavior

The agents performing autonomously and independently in decentralized networks can experience conflicting or unpredictable behavior. Detecting and managing issues within the larger system might be difficult under these conditions.