International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.3 (2025) pp. 6072-6082
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Model Context Protocol for Agentic AI: Enabling Contextual Interoperability Across Systems

Vallikranth Ayyagari*

DaVita Inc. USA

* **Corresponding Author Email:** vallikranth@gmail.com

Article Info:

DOI: 10.22399/ijcesen.3678
Received : 12 June 2025
Accepted : 22 August 2025

Keywords

Context
Agentic AI
Interoperability
MCP

Abstract:

The Model Context Protocol (MCP) is an open standard developed by Anthropic to facilitate seamless integration between large language models (LLMs) and external data sources or tools. By standardizing context exchange, MCP allows AI agents to interact with diverse systems in a secure and scalable manner. This paper explores MCP's architecture, components, and its role in enhancing agentic AI capabilities. The protocol introduces primitives such as tools, resources, prompts, and sampling to establish a structured interaction layer between models and environments. MCP's modular design supports real-time, bidirectional communication over common transports like HTTP and WebSocket. Its adoption has enabled AI systems to become more flexible, interactive, and autonomous in enterprise and developer workflows. As context-aware intelligence becomes foundational, MCP offers a scalable pathway for building the next generation of agentic applications.

1. Introduction

The landscape of artificial intelligence is undergoing a profound transformation with the emergence of Agentic AI, a paradigm where systems exhibit autonomy, adaptability, and the capacity for goal-driven behaviour [1]. These advanced systems, often composed of multiple interconnected AI agents, leverage the power of large language models (LLMs) and complex reasoning mechanisms to tackle complex, multi-step problems [4]. The relevance of Agentic AI is rapidly expanding across diverse sectors, including healthcare, finance, manufacturing, and customer service, promising to revolutionize workflows and decision-making processes [2].

A cornerstone of effective Agentic AI operation is the concept of context. The ability of an AI agent to understand its operational environment, including past interactions, relevant data, and available tools, is crucial for it to reason, plan, and act appropriately [1]. For AI agents to perform complex tasks and deliver accurate, context-aware responses, seamless access to relevant data repositories and a diverse array of tools is essential [1].

However, a significant hurdle in harnessing the full potential of Agentic AI lies in the challenge of integrating disparate AI systems, data sources, and tools. The current landscape is often characterized by fragmented integrations, hindering the scalability and interoperability of these advanced AI systems [15]. This situation is frequently described as the "M×N integration problem," where M AI applications require custom-built connectors for each of the N tools they need to interact with, leading to substantial development and maintenance overhead [17]. Frameworks like LangChain [71] and LlamaIndex [72] provide standardized interfaces, making it easier to integrate LLMs with external systems. Other AI providers, including Anthropic, Google, and Meta, introduced similar mechanisms, further driving adoption. Despite these advancements, integrating tools remains fragmented.

To address these challenges, Anthropic introduced the Model Context Protocol (MCP) in late 2024. This open standard is designed to provide a unified interface that enables AI systems, particularly LLMs, to connect with external data sources and tools in a standardized manner [17]. Often likened to a "USB-C for AI applications," MCP aims to establish a standardized "plug-and-play" ecosystem for AI integration, simplifying the process of connecting diverse AI components.[15] As the demand for context-aware applications grows, the MCP is positioned as a pivotal solution for achieving

interoperability and integration across multiple platforms. This research paper aims to provide a comprehensive analysis of the Model Context Protocol and its critical role in enabling contextual interoperability across Agentic AI systems.

2. Background: Understanding Agentic AI

Agentic AI represents a significant leap in the evolution of artificial intelligence, characterized by systems that exhibit autonomy, goal-driven behavior, and the ability to adapt to changing circumstances [1]. Unlike traditional AI models that operate within predefined constraints and often require direct human intervention, agentic systems can perform tasks with limited supervision, make decisions based on their understanding of the environment, learn from their interactions, and adjust their strategies in real-time [1]. Key attributes of Agentic AI include autonomy, the capacity for reasoning, the ability to set and pursue goals, the capability to make decisions, the execution of actions, and the continuous process of learning and adaptation [1]. Agentic AI often involves the coordinated efforts of multiple individual AI agents working together to tackle complex, multi-faceted goals that would be challenging for a single agent to achieve [4]. The transition from generative AI, which primarily focuses on creating new content, to agentic AI, which emphasizes autonomous action and decision-making, marks a crucial advancement in the field, highlighting the growing demand for AI systems that can actively engage with the real world and perform tasks independently [1].

The effective operation of Agentic AI systems is linked to their ability to access and process relevant context. These systems rely heavily on understanding the situation at hand to set appropriate goals and make informed decisions.[1] This necessary context often resides in external data repositories, encompassing a wide range of sources such as business applications and developer tools [17]. To translate their understanding into real-world impact, AI agents must also be capable of utilizing external tools, including APIs, databases, and various other services [14]. The ability to maintain context not only within a single interaction but also across multiple exchanges and tool calls is significant for enabling complex, multi-step workflows where AI agents need to remember previous actions and their outcomes to proceed effectively [14]. Ultimately, the capacity of an AI agent to act autonomously is fundamentally dependent on its ability to access and process relevant context and utilize appropriate tools, highlighting that an AI agent's effectiveness is directly limited by its access to information and its means of interacting with external systems.

3. The Imperative of Contextual Interoperability

A significant impediment to the widespread and efficient deployment of Agentic AI lies in the challenges associated with integrating the diverse array of systems these agents need to interact with. Traditional methods of connecting AI models to external resources often involve developing custom integrations tailored to each specific data source, leading to a fragmented and increasingly complex integration scenario [17]. This approach gives rise to the "M×N integration problem," where a multitude of AI applications must each establish unique connections with a multitude of tools, resulting in a complex web of custom connectors that are both difficult to manage and costly to maintain [17]. Also, the existing Application Programming Interfaces (APIs), while facilitating data exchange, may lack the semantic context that AI agents require for truly intelligent interaction and understanding [36]. The issue of context fragmentation, where relevant context is not consistently maintained across different interactions and systems, further compounds the problem, significantly limiting the ability of AI agents to effectively execute complex, multi-step tasks that necessitate a cohesive understanding of prior states and information [14]. The current reliance on bespoke integrations for every AI-tool connection is rapidly becoming unsustainable as the number of available AI models and specialized tools continues to proliferate, underscoring the critical need for a standardized approach to streamline these interactions and enhance both development efficiency and system scalability.

For multi-agent systems, a core component of many Agentic AI architectures, the ability of individual agents to share context seamlessly is essential for effective collaboration. This shared understanding enables agents to coordinate their actions, resolve conflicts, and collectively work towards achieving common objectives.[8] Maintaining context over extended periods and across the execution of complex workflows is crucial, allowing AI agents to retain vital knowledge and make more informed, contextually relevant decisions [14]. Achieving contextual interoperability allows AI agents to access and utilize a wide variety of data types and a diverse set of tools from different systems in a unified and coherent manner.[14] The benefits of such interoperability are manifold, including faster time-to-value for AI initiatives, the ability to reuse existing integration assets more effectively, and the development of more flexible and adaptable AI systems [34]. The capacity to share and maintain context transcends the mere exchange of data; it

empowers AI agents to reason, plan, and act coherently over extended durations and across multiple interconnected systems, mirroring the collaborative problem-solving capabilities observed in human interactions.

4. Model Context Protocol (MCP): Architecture and Functionality

The Model Context Protocol (MCP) is built upon a client-server architecture, a well-established model for distributed systems [17]. This architecture as shown in Fig 1 comprises three primary components: the MCP Host, the MCP Client, and the MCP Server.[15] The MCP Host is typically the main AI application itself, such as Anthropic's Claude Desktop or an AI-powered Integrated Development Environment (IDE). It is responsible for initiating connections to MCP Servers and coordinating the overall interaction between the AI model and the external tools and data sources.[15] Running within the Host is the MCP Client, which acts as a connector, establishing and managing a one-to-one communication pipeline with individual MCP Servers.[15] The MCP Server is a separate process or service that exposes specific functionalities, such as access to data, the ability to execute actions, or pre-defined conversational prompts, through the MCP protocol [15].

Communication between MCP Clients and Servers relies on JSON-RPC 2.0, a widely adopted and versatile protocol that enables the invocation of methods over various communication channels, including standard input/output (stdio), Hypertext Transfer Protocol (HTTP), and WebSocket [22]. This choice of protocol provides a well-established and flexible foundation for MCP, allowing it to be implemented across diverse platforms and to support both local and remote communication scenarios. Additionally, MCP supports stateful connections, which are crucial for maintaining persistent context across multiple messages exchanged between the client and the server [24].

MCP Servers make their capabilities accessible through three primary building blocks or primitives [17]. Prompts are reusable conversational workflows or templates that can be initiated by users to guide the AI model's response or behavior in specific ways [17]. Resources represent static or dynamic data that the AI application can access for contextual reference. These can include files, records from databases, or responses from APIs [17]. Tools are functions or actions that the AI application can dynamically invoke to perform specific tasks, such as analyzing data, sending emails, or interacting with external APIs and services [14]. In addition to these core primitives, MCP also supports features like Sampling, where MCP Clients can offer server-initiated agentic behaviors and recursive LLM interactions; Roots, a mechanism for managing and referencing specific contexts; Notifications, which are server-initiated messages sent to the client; and Pings, which allow the client to check the status of the server.[27] A key aspect of MCP is its support for dynamic tool discovery. AI agents can query MCP servers at runtime to understand the capabilities they offer, including the available tools, resources, and prompts.[18] This allows AI applications to adapt to the available functionalities without requiring prior knowledge or configuration for every possible server. The introduction of these standardized primitives provides a common language that enables AI models and external systems to interact more effectively, thereby simplifying the development of AI workflows.

Figure 1. *MCP Protocol Architecture.*

5. Enabling Contextual Interoperability with MCP

One of the primary ways MCP enables contextual interoperability is by standardizing the integration process, effectively reducing the complexity of connecting AI applications with external resources [17]. By requiring each AI application and tool to implement the MCP protocol only once, it transforms the "M×N integration problem" into a more manageable "M+N problem" [17]. This standardization significantly lowers the integration effort and the ongoing maintenance burden compared to the traditional approach of building and maintaining custom connectors for every unique pair of AI application and tool [18]. MCP acts as a unified execution layer that sits above existing API strategies, such as REST and GraphQL, making these underlying functionalities more readily

accessible to intelligent AI systems through a consistent and standardized interface [34]. The core value of MCP lies in its ability to establish a standardized method for AI models to interact with a diverse range of external resources, thereby unlocking the potential for more powerful and versatile AI applications.

MCP facilitates contextual interoperability by enabling AI models to dynamically discover and utilize the available tools and resources based on the specific context of the task at hand [14]. Moreover, MCP supports bidirectional communication, allowing for not just the retrieval of data but also the invocation of tools to perform actions and the execution of those actions on behalf of the user or another system [20]. By providing a standardized way to access and utilize tools and data, MCP empowers AI agents to interact with their environment in a more intelligent and adaptive manner, moving beyond simple question-answering to performing complex actions based on real-time, contextually relevant information.

6. Benefits and Applications of MCP in Agentic AI

The adoption of the Model Context Protocol in Agentic AI systems offers a multitude of significant benefits as outlined in Table 2. MCP provides a standardized language that bridges the gap between AI systems and the vast ecosystem of external data and tools [22]. Its design is flexible and vendor-agnostic, ensuring that it is not tied to any specific AI model or provider [18]. As MCP gains traction, it promotes a growing ecosystem of reusable connectors, accelerating the adoption and expansion of AI capabilities [20]. A key advantage is the reduction in integration overhead; MCP significantly decreases the amount of custom code that developers need to write to connect AI models to external resources [18]. This standardization also enhances context awareness, improving the ability of AI models to access and utilize real-time and relevant data for more accurate and informed responses [14]. Also, MCP enables dynamic tool discovery and execution, allowing AI models to automatically detect and interact with new tools or services as they become available [18]. The protocol also provides a framework for improved security and access control, crucial for managing sensitive data and ensuring authorized tool usage [14]. By standardizing context access, MCP helps future-proof AI applications, allowing them to adapt to new AI models and services without requiring extensive rewrites.[18] This also leads to accelerated development cycles as integration efforts are streamlined[18]. The collective impact of these benefits points towards a more efficient, scalable, and adaptable AI ecosystem, allowing developers to focus on building innovative AI applications rather than managing complex integrations.

The potential applications of MCP in Agentic AI are vast and span numerous domains. In AI-powered code editors, MCP enables seamless access to relevant code context, repository structure, and documentation, enhancing developer productivity [18]. Customer support systems can leverage MCP to access CRM data, product information, and support tickets in real-time, providing more accurate and contextual help [18]. Data analysis workflows are enhanced by MCP's ability to dynamically query databases, generate visualizations, and explain insights, all while maintaining context across multiple analytical steps [18]. In enterprise AI search, MCP enables AI to search across diverse document stores, databases, and cloud storage, linking responses directly to their source documents [22]. Automated trading systems in finance can utilize MCP to analyze market trends in real-time and efficiently interact with various financial services.[40] Smart factories can leverage MCP to connect various machines and systems, enabling real-time monitoring and optimization of production lines [40]. E-commerce platforms can enhance personalized shopping experiences by using MCP to tailor product recommendations based on user data [40]. In healthcare, MCP can facilitate AI-driven diagnostic tools accessing electronic health records, improving the speed and accuracy of medical decisions [34]. MCP also enables the creation of multi-step, cross-system workflows, allowing AI agents to coordinate tasks across different platforms through a single, standardized interface [37]. Other potential applications include AI agents that understand and interact with their environment in smart homes and robotics, collaborating agents in multi-agent systems, personal AI assistants with deep integration into personal data, and enterprise governance and security systems that standardize AI access to internal tools with monitoring and control [37]. The growing list of integrations with platforms like Google Drive, Slack, GitHub, AWS, Stripe, Supabase, Grafana, Microsoft Playwright, Neo4j, Brave Search, and the support from major AI players like OpenAI and Google DeepMind further underscore the significant impact MCP is poised to have [18].

Table 1. *Comparison of MCP features*

Feature	Model Context Protocol (MCP)	Traditional APIs (REST, GraphQL)	Agent Communication Languages (ACLs) (e.g., FIPA-ACL, KQML)
Primary Goal	Standardize context and tool access for AI agents	Facilitate data exchange between applications	Standardize the format and semantics of messages exchanged between agents
Context Handling	Built-in, supports stateful connections and rich contextual primitives (Prompts, Resources, Tools)	Typically stateless, requires manual handling of context across requests	Focuses on the content of the message and the communicative act
Capability Discovery	Dynamic discovery of available tools and their capabilities at runtime	Requires prior knowledge of API endpoints and specifications	Agents typically need to advertise their capabilities explicitly
AI Agent Focus	Specifically designed for the needs of AI agents and LLMs	General-purpose, not specifically optimized for AI agent interactions	Designed for communication and coordination between autonomous agents
Abstraction Level	Sits above transport protocols like HTTP, provides a unified execution layer for AI	Operates at the application layer, defines how applications request and exchange data	Operates at a higher level, focusing on the meaning and intent of communication
Integration Complexity	Aims to reduce the "M×N" problem to "M+N" by providing a standardized protocol	Requires custom connectors for each application-tool combination	Can involve complex semantic frameworks and ontologies

Table 2. *Explanation of MCP Protocol benefits*

Benefit	Description/Explanation
Standardization and Interoperability	Provides a common language for AI systems and external resources [22]
Flexibility and Vendor-Agnostic Design	Works with various AI models and vendors [18]
Ecosystem and Reusability	Promotes a growing collection of reusable connectors [20]
Reduced Integration Overhead	Simplifies connecting AI models to external systems [18]
Enhanced Context Awareness	Improves AI's ability to access and use relevant data [14]
Dynamic Tool Discovery	Enables AI to find and use new tools automatically [18]
Improved Security	Offers a framework for managing access to data and tools [14]

7. MCP in Relation to Existing Protocols and Frameworks

When considering the role of MCP, it is important to understand its position relative to existing technologies as explained in Table 1. Traditional APIs, such as REST and GraphQL, often handle individual, stateless requests, which can make maintaining context across multiple interactions challenging for AI agents [36]. The specifications for these APIs are typically static, which limits the ability of AI agents to dynamically discover and utilize new capabilities [18]. In contrast, MCP is designed with the specific needs of AI agents in mind, prioritizing the exchange of rich, structured context and supporting features like dynamic tool discovery [30]. MCP operates at a higher level of abstraction than these traditional APIs, sitting above transport protocols like HTTP and providing a unified execution layer that is specifically tailored for AI interactions [34]. While traditional APIs remain crucial for general system-to-system communication, MCP offers an AI-centric approach that more directly addresses the requirements of Agentic AI for contextual awareness and the dynamic utilization of tools.

Agent Communication Languages (ACLs), such as FIPA-ACL and KQML, emerged earlier in the field of multi-agent systems, providing standardized message structures with defined performatives to facilitate communication between autonomous agents [32]. These languages often focused on explicitly conveying the semantic intent behind messages and sometimes required significant effort in developing shared ontologies [32]. While MCP also enables communication, its primary focus is on

the exchange of comprehensive, structured context and the invocation of tools, rather than on formal communicative acts between agents [32]. Notably, emerging protocols like Google's Agent-to-Agent (A2A) Protocol appear to have a complementary relationship with MCP. While MCP focuses on enabling an agent to interact with tools and data, A2A is designed for direct communication and collaboration between different AI agents [32]. This suggests a trend towards specialized protocols that address different facets of interaction within complex Agentic AI systems.

In the context of other agent interaction frameworks, such as LangChain and LlamaIndex, MCP occupies a distinct but potentially complementary role. These frameworks typically provide developer-centric tools and standards for integrating external tools and data sources into the code of an AI agent [37]. MCP, on the other hand, establishes a model-centric standard, allowing the AI agent itself to discover and utilize any tool that adheres to the MCP specification at runtime [37]. Therefore, MCP can be viewed as providing a standardized interface for the underlying implementation of tools that these higher-level frameworks can then leverage within their agent architectures. This layered approach suggests that MCP is not intended to replace these frameworks but rather to provide a foundational layer that enhances their ability to connect with a wider range of external capabilities in a more standardized and dynamic way.

8. Challenges, Security, and Scalability of MCP Implementation

Despite the numerous advantages offered by MCP, its implementation and adoption are not without potential technical challenges. Developers may encounter an initial learning curve associated with understanding and implementing the protocol.[55] Integrating MCP with existing heterogeneous data sources and legacy systems can still present complexities, potentially requiring custom solutions to bridge the gaps between different technologies [56]. Ensuring consistency in metadata and achieving accurate semantic interpretation across diverse systems that are connected through MCP is crucial for its effectiveness, but it can also be a non-trivial task [56]. Moreover, developers might face practical obstacles such as the need for writing custom code to handle specific integration scenarios and managing the translation of context between different formats and systems [56]. While MCP aims to simplify the integration landscape, the inherent complexities of existing IT infrastructures mean that initial implementation and the integration with a wide variety of systems may still require significant technical expertise and careful planning.

Given that MCP facilitates powerful interactions between AI models and external data and tools, security is a important consideration [14]. The protocol's design emphasizes key security principles, including the necessity of explicit user consent and control over data access and actions, the importance of maintaining data privacy, the need to ensure the safety of invoked tools, and the requirement for controls over LLM sampling [21]. Potential security risks associated with MCP implementation as detailed in Table 3 include the exposure of sensitive data, the possibility of unauthorized access to systems and resources, the threat of content poisoning through malicious data or tools, and vulnerabilities that could be exploited during the creation, operation, or updating of MCP servers.[14] To mitigate these risks, it is essential to implement robust consent and authorization mechanisms, enforce strict access controls, employ comprehensive data protection measures, and adhere to established security best practices [14]. A multi-layered security approach, incorporating end-to-end encryption for data transmission, granular access control mechanisms, and comprehensive audit trails of all interactions, is recommended to ensure the secure and responsible use of MCP.

As Agentic AI systems that utilize MCP become more prevalent and are tasked with handling increasing volumes of data and interactions, ensuring the scalability of MCP implementations will be critical for maintaining optimal performance and responsiveness [18]. The process of context handling and data exchange can become resource-intensive, especially as the complexity and volume of data increase [56]. High latency in retrieving context can negatively impact the real-time responsiveness expected of many AI applications.[56] Therefore, efficient mechanisms for transmitting and managing context are important for achieving scalability [55]. Various techniques can be employed to address these challenges, including parallel query routing to process multiple requests simultaneously, distributed caching to reduce redundant data retrieval, and elastic resource allocation to dynamically adjust computing resources based on demand [55]. Additionally, designing MCP implementations with stateless request processing and implementing context-aware load balancing can further contribute to achieving scalability in high-throughput environments.

Table 3. *Security Risks with MCP Protocol*

Security Risk	Phase of Lifecycle	Brief Description/Mitigation Strategies
Data Exposure	Operation	Potential unauthorized access during context transmission; Implement end-to-end encryption and granular access controls [21]
Tool Name Conflicts	Operation	Conflicts between tool names from different servers; Implement clear naming conventions and namespacing [45]
Code Injection/Backdoor	Creation	Malicious code introduced during server creation; Implement secure development practices and code reviews [45]
Installer Spoofing	Creation	Attackers distributing malicious servers disguised as legitimate ones; Use trusted sources and verify signatures [45]
Slash Command Overlap	Operation	Overlapping slash commands from different tools causing unintended actions; Implement clear command prefixes and user education [45]
Sandbox Escape	Operation	Attackers breaking out of the server's sandbox to access the host system; Employ robust sandboxing techniques and regular security audits [45]
Post-Update Privilege Persistence	Update	Vulnerabilities allowing attackers to maintain elevated privileges after a server update; Ensure secure update mechanisms and principle of least privilege [45]
Re-deployment of Vulnerable Versions	Update	Rolling back to or re-deploying older, vulnerable server versions; Maintain secure version control and prevent deployment of known vulnerabilities [45]
Configuration Drift	Update	Unauthorized or unintended changes to server configurations; Implement configuration management and monitoring [45]

9. Future Directions and Research Opportunities

As a relatively new technology, MCP is expected to undergo further evolution and play an increasingly significant role in the future of Agentic AI [8]. There are capabilities being built on top of MCP protocol such as the recent MCP bridge [69] which looks promising. Future developments may include enhanced support for more intricate, multi-agent workflows, often referred to as "Agent Graphs," where complex interactions and collaborations between multiple AI agents are orchestrated through MCP.[8] Another promising direction is the addition of multimodal capabilities, allowing AI agents to exchange and process information beyond text, such as images, audio, and video, through the MCP framework [18]. The ongoing enhancement of security features and the development of more granular and flexible permission models will be crucial for ensuring the safe and responsible use of MCP in various applications [18]. The establishment of formal governance structures and industry-wide standardization efforts will also be important for promoting broader adoption and interoperability across different AI ecosystems [18]. We can anticipate the development of more sophisticated tools and platforms for facilitating the discovery, management, and monitoring of MCP servers and clients [43]. Improvements in remote deployment and management capabilities will also be essential for enabling the widespread use of MCP in enterprise environments and cloud-based AI solutions [39]. Finally, exploring the integration of MCP with other emerging AI technologies and

paradigms, such as specialized AI models and advanced reasoning frameworks, holds the potential to unlock even more powerful and innovative applications [8].

The nascent stage of MCP also presents numerous opportunities for future research and development. A critical area of focus should be on conducting thorough security analysis and developing robust threat models specifically for MCP implementations to identify and mitigate potential vulnerabilities [14]. Further research is needed to evaluate and optimize the performance of MCP in various use cases and at different scales, particularly in high-throughput scenarios involving complex data and interactions.[55] The development of best practices, comprehensive guidelines, and standardized tools for building, evaluating (eg: MCPBench [70]) secure and scalable MCP servers, clients will be invaluable for the developer community [39]. Exploring novel applications and use cases for MCP across a wider range of industries and domains beyond the initial set of early adopters also presents a significant opportunity [40]. Investigating the interoperability of MCP with other emerging agent communication protocols and frameworks, such as Google's A2A, could lead to more seamless and powerful multi-agent systems.[32] Research into the design of effective human-in-the-loop mechanisms and intuitive user interfaces for managing MCP interactions and permissions will be crucial for ensuring user trust and control over these powerful AI systems [26]. Finally, as Agentic AI systems become more autonomous, research into the ethical implications of using MCP, particularly concerning issues of accountability, transparency, and potential bias, will be essential for guiding its responsible development and deployment [11].

Conclusion

Contextual interoperability stands as a critical requirement for the continued advancement and widespread adoption of Agentic AI. The ability for autonomous AI systems to seamlessly exchange and understand not just data, but also the rich context surrounding that data, is crucial for enabling them to perform complex tasks, collaborate effectively, and ultimately realize their full potential. The Model Context Protocol (MCP) emerges as a promising solution to address this imperative, offering a standardized framework for connecting AI systems with the external world of data and tools.

MCP's key features, including its client-server architecture, the introduction of standardized primitives like Prompts, Resources, and Tools, and its support for dynamic tool discovery and stateful connections, provide a robust foundation for enabling contextual interoperability. The benefits of adopting MCP are numerous, ranging from the standardization of integration processes and the reduction of development complexity to the enhancement of context awareness and the facilitation of dynamic tool usage. These advantages collectively pave the way for a more efficient, scalable, and adaptable AI ecosystem.

MCP is not just a protocol, it's a rapidly evolving ecosystem. Future versions of MCP will likely expand beyond text to support other data modalities like video, audio and images, enabling richer AI interactions. Initiatives like MCP registry will improve the discoverability and sharing of MCP servers.

While MCP holds significant promise, its successful adoption necessitates careful consideration of the challenges associated with its implementation, particularly concerning security and scalability. Addressing these aspects through robust security measures, efficient resource management, and ongoing research and development will be crucial for ensuring the reliable and responsible use of MCP in a wide range of applications.

In conclusion, the Model Context Protocol represents a significant step forward in enabling contextual interoperability for Agentic AI systems. By providing a standardized and flexible framework for AI models to interact with external resources, MCP has the potential to revolutionize the way AI agents operate and collaborate. As the protocol continues to evolve and its ecosystem expands, it is poised to play a transformative role in shaping the future of Agentic AI and the broader artificial intelligence spectrum.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data

are not publicly available due to privacy or ethical restrictions.

References

[1] What Is Agentic AI?, (2025), *IBM*. https://www.ibm.com/think/topics/agentic-ai

[2] What is Agentic AI?, (2025). Salesforce US. https://www.salesforce.com/agentforce/what-is-agentic-ai/

[3] What is Agentic AI?, (2025). Key Trends in 2025 - *Aisera*, https://aisera.com/blog/agentic-ai/

[4] Agentic AI Vs AI Agents: 5 Differences and Why They Matter, (2025). Moveworks, https://www.moveworks.com/us/en/resources/blog/agentic-ai-vs-ai-agents-definitions-and-differences

[5] What is agentic AI? (2025). *Red Hat*, https://www.redhat.com/en/topics/ai/what-is-agentic-ai

[6] What Is Agentic AI?, (2025). NVIDIA Blog, https://blogs.nvidia.com/blog/what-is-agentic-ai/

[7] *aisera.com*, (2025), https://aisera.com/blog/agentic-ai/#:~:text=Agentic%20AI%20is%20an%20AI,(LLMs)%20and%20complex%20reasoning.

[8] Multi-Agent AI Systems, (2025). *Aisera*, https://aisera.com/blog/multi-agent-ai-system/

[9] Multi-Agent Collaboration Mechanisms: A Survey of LLMs, (2025) *arXiv*, https://arxiv.org/html/2501.06322v1

[10] What is Agentic AI?, (2025). *UiPath*, https://www.uipath.com/ai/agentic-ai

[11] What is Agentic AI?, (2025). An Easy Explanation for Everyone-YouTube, https://www.youtube.com/watch?v=-pqzyvRp3Tc

[12] From Experimentation to Deployment: Crafting an Agentic AI Strategy for Your Business, (2025). https://www.arionresearch.com/blog/h8qt2olb6jx4el4dirn0eakjmz6eob

[13] Agentic Workflows in Healthcare: Advancing Clinical Efficiency through AI Integration, (2025). https://www.researchgate.net/publication/389664995_Agentic_Workflows_in_Healthcare_Advancing_Clinical_Efficiency_through_AI_Integration

[14] How Model Context Protocol (MCP) Transforms Your AI into a Powerful Digital Assistant, (2025). *Kanerika*, https://kanerika.com/blogs/model-context-protocol-mcp/

[15] AI Spotlight: MCP (Model Context Protocol) and Agentic AI systems. *Gravitee.io*, (2025), https://www.gravitee.io/blog/mcp-model-context-protocol-agentic-ai

[16] Model Context Protocol (MCP) - A Deep Dive-WWT, (2025), https://www.wwt.com/blog/model-context-protocol-mcp-a-deep-dive

[17] How Model Context Protocol Connects LLMs to the Real World. (2025). *Label Studio*, https://labelstud.io/blog/how-model-context-protocol-connects-llms-to-the-real-world/

[18] Model Context Protocol: What You Need to Know, (2025). Gradient Flow. https://gradientflow.com/model-context-protocol-what-you-need-to-know/

[19] Introduction to Model Context Protocol (MCP)., (2025). *LearnOpenCV*. https://learnopencv.com/introduction-to-model-context-protocol/

[20] Introducing the Model Context Protocol-(2025). *Anthropic*. https://www.anthropic.com/news/model-context-protocol

[21] What is a MCP (Model Context Protocol) & How Does it Work?, (2025). *Use Cases + Examples*, https://www.shakudo.io/blog/mcp-model-context-protocol

[22] What is Model Context Protocol?, (2025). The emerging standard bridging AI and data, explained. https://www.zdnet.com/article/what-is-model-context-protocol-the-emerging-standard-bridging-ai-and-data-explained/

[23] *aisera.com*, (2025), https://aisera.com/blog/mcp-model-context-protocol/#:~:text=The%20Model%20Context%20Protocol%20(MCP)%2C%20launched%20by%20Anthropic%20in,tools%20via%20a%20unified%20interface.

[24] Model Context Protocol (MCP): The USB-C for AI?, (2025). *Aisera*. https://aisera.com/blog/mcp-model-context-protocol/

[25] Model Context Protocol (MCP), (2025). Anthropic API. https://docs.anthropic.com/en/docs/agents-and-tools/mcp

[26] A beginners Guide on Model Context Protocol (MCP), (2025). *OpenCV*. https://opencv.org/blog/model-context-protocol/

[27] Specification, (2025). *Model Context Protocol*. https://modelcontextprotocol.io/specification/2025-03-26

[28] A Survey of the Model Context Protocol (MCP): Standardizing Context to Enhance Large Language Models (LLMs), (2025).

Preprints.org, https://www.preprints.org/manuscript/202504.0245/v1

[29] Is Anthropic's Model Context Protocol Right for You?, (2025). *WillowTree Apps*, https://www.willowtreeapps.com/craft/is-anthropic-model-context-protocol-right-for-you

[30] Model Context Protocol (MCP), (2025). Everything You Need to Know, https://zencoder.ai/blog/model-context-protocol

[31] Anthropic's Model Context Protocol (MCP) is way bigger than most people think, (2025). *r/ClaudeAI*. https://www.reddit.com/r/ClaudeAI/comments/1gzv8b9/anthropics_model_context_protocol_mcp_is_way/

[32] Moving from monolithic to microservices architecture for multi-agent systems. (2025). *World Journal of Advanced Engineering Technology and Sciences*, https://journalwjaets.com/sites/default/files/fulltext_pdf/WJAETS-2025-0480.pdf

[33] Agentic AI vs. Generative AI, (2025). *IBM*. https://www.ibm.com/think/topics/agentic-ai-vs-generative-ai

[34] How to Use Model Context Protocol the Right Way. (2025). *Boomi*, https://boomi.com/blog/model-context-protocol-how-to-use/

[35] MCP: The Breakthrough Protocol Powering the Next Era of Agentic AI, (2025). *Fluid AI*. https://www.fluid.ai/blog/mcp-the-breakthrough-protocol

[36] Model Context Protocol - The Foundation for Truly Agentic AI, (2025). *Vinci Rufus*. https://www.vincirufus.com/posts/mcp-foundation-for-agentic-ai/

[37] #14: What Is MCP, and Why Is Everyone – Suddenly!-Talking About It?, (2025). *Hugging Face*. https://huggingface.co/blog/Kseniase/mcp

[38] Model Context Protocol (MCP): The Key To Agentic AI, (2025). *YouTube*, https://www.youtube.com/watch?v=VChRPFUzJGA

[39] Frequently Asked Questions About Model Context Protocol (MCP) and Integrating with AI for Agentic Applications, (2025). *Tenable*. https://www.tenable.com/blog/faq-about-model-context-protocol-mcp-and-integrating-ai-for-agentic-applications

[40] MCP for Automation: Real-World Applications and Case Studies, (2025). *Arsturn*. https://www.arsturn.com/blog/mcp-for-automation-real-world-applications-and-case-studies

[41] Model Context Protocol (MCP) and Its Impact on AI-Driven Startups, (2025). https://www.aalpha.net/blog/model-context-protocol-mcp-and-its-impact-on-ai-driven-startups/

[42] Model Context Protocol (MCP) Explained, (2025). *Humanloop*. https://humanloop.com/blog/mcp

[43] Everything a Developer Needs to Know About the Model Context Protocol (MCP), (2025). *Neo4j*, https://neo4j.com/blog/developer/model-context-protocol/

[44] Advancing Multi-Agent Systems Through Model Context Protocol: Architecture, Implementation, and Applications, (2025). *arXiv*. https://arxiv.org/html/2504.21030v1

[45] Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions, (2025). *arXiv*. https://arxiv.org/html/2503.23278

[46] Model Context Protocol (MCP): A Developer's Guide, (2025). *Wallarm*. https://www.wallarm.com/what/what-is-model-context-protocol-mcp

[47] Announcing the Agent2Agent Protocol (A2A), (2025). *Google for Developers Blog*. https://developers.googleblog.com/en/a2a-a-new-era-of-agent-interoperability/

[48] Papers by Prashant Kulkarni, (2025). *AIModels.fyi*. https://www.aimodels.fyi/author-profile/prashant-kulkarni-6369b855-996d-499d-908f-a4a23741e13e

[49] Multi-Agent Environment Tools: Top Frameworks, (2025). *Rapid Innovation.* https://www.rapidinnovation.io/post/frameworks-and-tools-for-building-multi-agent-environments

[50] Our abstract model for interoperating software agents identies three, (2025). *ResearchGate*. https://www.researchgate.net/figure/Our-abstract-model-for-interoperating-software-agents-identiies-three-classes-of_fig1_221613402

[51] An Introduction to MultiAgent Systems, (2025). Request PDF. *ResearchGate*. https://www.researchgate.net/publication/200027549_An_Introduction_to_MultiAgent_Systems

[52] Turbo prolog 2.0 basics, PDF, (2025). *SlideShare*. https://www.slideshare.net/slideshow/turbo-prolog-20-basics/139319370

[53] Google A2A Protocol: Language Support & Capabilities Explained, (2025). *BytePlus*. https://www.byteplus.com/en/topic/551575

[54] Building A Secure Agentic AI Application Leveraging Google's A2A Protocol, (2025). *arXiv*. https://arxiv.org/html/2504.16902

[55] Model Context Protocol Benefits: Key

[56] MCP Model Limitations: Key Challenges & Solutions, (2025). *BytePlus*. https://www.byteplus.com/en/topic/542231

[57] [2503.23278] Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions, (2025). *arXiv*. https://arxiv.org/abs/2503.23278

[58] How MCP handles context management in high-throughput scenarios, (2025). *Portkey*. https://portkey.ai/blog/model-context-protocol-context-management-in-high-throughput/

[59] One AI Model Won't Fit All: Why Enterprise Workflows Need Multi-LLM and Contextual Interop, (2025). *Fluid AI*. https://www.fluid.ai/blog/why-enterprise-workflows-need-multi-llm-and-contextual-interop

[60] Interoperability Is Key to Unlocking Agentic AI's Future, (2025). *Forrester*. https://www.forrester.com/blogs/interoperability-is-key-to-unlocking-agentic-ais-future/

[61] Understanding Multi-Agent AI Frameworks, (2025). *Ema*. https://www.ema.co/additional-blogs/addition-blogs/understanding-multi-agent-ai-frameworks

[62] Agentic AI for Scientific Discovery, (2025). https://iclragenticai.github.io/

[63] MCP for Multimodal AI Applications: Benefits & Integration, (2025). BytePlus. https://www.byteplus.com/en/topic/541338

[64] RAG, AI Agents, and Agentic RAG: An In-Depth Review and Comparative Analysis, (2025). https://www.digitalocean.com/community/conceptual-articles/rag-ai-agents-agentic-rag-comparative-analysis

Advantages Explained, (2025). *BytePlus*. https://www.byteplus.com/en/topic/541182

[65] Model Context Protocol Case Studies, (2025). Real-World Examples. *BytePlus*. https://www.byteplus.com/en/topic/541353

[66] Moving from monolithic to microservices architecture for multi-agent systems, (2025). https://journalwjaets.com/content/moving-monolithic-microservices-architecture-multi-agent-systems

[67] Agentic AI for Ontology Grounding over LLM-Discovered Scientific Schemas in a Human-in-the-Loop Workflow, (2025). *Semantic Web Journal*. https://www.semantic-web-journal.net/system/files/swj3871.pdf

[68] The Co-Leadership Challenge: What Healthcare Can Learn from the AI CEO Buzz, (2025). https://www.vktr.com/ai-technology/the-co-leadership-challenge-what-healthcare-can-learn-from-the-ai-ceo-buzz/

[69] Ahmadi, Arash & Sharifi, Safura & Banad, Yaser. (2025). MCP Bridge: A Lightweight, LLM-Agnostic RESTful Proxy for Model Context Protocol Servers. 10.48550/arXiv.2504.08999.

[70] Luo, Zhiling, Xiaorong Shi, Xuanrui Lin and Jinyang Gao, (2025). Evaluation Report on MCP Servers.

[71] LangChain, (2022). LangChain: Framework for developing applications powered by language models. https://github.com/langchain-ai/langchain .

[72] Jerry Liu, (2022). LlamaIndex: A data framework for LLM applications. https://github.com/run-llama/llama_index .