



International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 21 No. 4 (2025)



ijerst.editor@gmail.com
editor@ijerst.com

Research Paper**MODEL CONTEXT PROTOCOL****M V NARAYANA¹, M. VEERAMANIDRA CHARY², B. SAI³**¹Professor, Dept. of CSE, Guru Nanak Institutions Technical Campus, Ibrahimpatnam, Telangana, India^{2,3}B.Tech, Dept. of CSE, Guru Nanak Institutions Technical Campus, Ibrahimpatnam, Telangana, India

ABSTRACT— The Model Context Protocol (MCP) is a framework designed to enhance the functionality of AI models by enabling structured and secure access to external data sources, tools, and services. Unlike traditional AI systems that rely solely on pre-trained knowledge, MCP allows models to dynamically interact with real-time information, APIs, databases, and domain-specific applications. This interaction facilitates context-aware decision-making, improves response accuracy, and extends the operational capabilities of AI-driven systems. MCP emphasizes modularity, interoperability, and security, ensuring that models can safely integrate with external resources while maintaining performance and data privacy. Overall, MCP bridges the gap between static model intelligence and dynamic real-world data, supporting the development of smarter, adaptive, and more reliable AI applications.

Index Terms – Model Context, Protocol, Tool Integration, Memory, Context Management

Received: 29-09-2025

Accepted: 01-11-2025

Published: 08-11-2025

I. INTRODUCTION

The Model Context Protocol (MCP) is a framework developed to enhance the capabilities of AI models by enabling them to access and utilize external tools, databases, APIs, and real-time data in a structured and secure manner. The primary objective of the Model Context Protocol (MCP) project is to enhance AI models by enabling them to access and utilize external data sources, tools, and services in a structured and secure manner. The system aims to provide context-aware and adaptive responses, improving the accuracy and relevance of AI outputs. It seeks to bridge the gap between static pre-trained knowledge and dynamic real-world information, allowing AI to perform tasks beyond its native capabilities. Additionally, the project emphasizes a modular, scalable, and secure architecture, ensuring seamless integration with external systems while maintaining data privacy and system reliability. By achieving these objectives, MCP supports the development of smarter, more versatile, and reliable AI applications across diverse domains

such as healthcare, finance, education, automation, and research. Traditional AI models rely mainly on pre-trained knowledge, which limits their ability to provide accurate and context-specific responses, particularly in dynamic environments. MCP addresses this limitation by allowing AI systems to interact dynamically with external data sources and tools, enabling context-aware decision-making, improved accuracy, and extended functionality. The protocol emphasizes modularity, security, and scalability, ensuring that AI systems can safely communicate with external resources while maintaining data privacy and system integrity. By bridging the gap between static model intelligence and live external information, MCP supports the development of smarter, adaptive, and reliable AI applications suitable for a wide range of domains, including healthcare, finance, education, automation, and research.

II. RELATED WORK

A. Enabling Dynamic and Context-Aware AI

Recent advancements in artificial intelligence have focused on enhancing model adaptability, contextual understanding, and real-time data interaction. Traditional AI systems, including large language models like GPT and BERT, are primarily trained on static datasets and lack the ability to integrate dynamic, real-world information during inference. To address this limitation, researchers have explored various frameworks and architectures aimed at improving context awareness and external data access. Studies on Retrieval-Augmented Generation (RAG) and Tool-Using Models have introduced mechanisms where AI systems retrieve relevant documents or execute external functions to enhance their outputs. These approaches laid the foundation for protocols like the Model Context Protocol (MCP), which standardizes communication between AI models and external resources such as databases, APIs, and software tools. MCP builds upon earlier work in AI modularization, context management, and secure data integration, providing a unified method for enabling real-time reasoning and dynamic decision-making. The development of MCP aligns with ongoing research in contextual AI, autonomous agents, and knowledge-augmented systems, all of which aim to create models that are not just responsive but also adaptive to the evolving information environment. Through this integration, MCP represents a significant step toward making AI systems more intelligent, interactive, and practically useful across diverse domains.

B. Enhancing AI Decision-Making via the Model Context Protocol

Research into improving AI decision-making has increasingly emphasized grounding model outputs in external context and actionable tools. Early work on retrieval augmented methods showed that conditioning models on retrieved documents reduces factual errors and improves

relevance, while subsequent research into tool-using LLMs demonstrated that enabling models to call calculators, search APIs, or execution engines expands their practical capabilities and correctness on complex tasks. Modular and hybrid architectures which separate planning/reasoning, retrieval, and execution components have proven effective for interpretability and error isolation, and several studies highlight the value of chaining tool calls with controlled policies to solve multi-step problems.

C. Towards Smarter and Adaptive AI Applications

Artificial Intelligence (AI) has undergone a remarkable transformation from rule-based systems to adaptive, context-aware frameworks capable of real-time learning and decision-making. Early research in AI primarily emphasized algorithmic efficiency and pattern recognition within fixed datasets, which limited the adaptability of models in dynamic environments. Recent developments, however, focus on creating smarter and more adaptive AI systems that can continuously learn, adjust, and respond to evolving data and user contexts. Studies on reinforcement learning, transfer learning, and context-aware reasoning have played a crucial role in enhancing AI's ability to make intelligent and situation-specific decisions. Researchers have also explored integrated AI architectures that combine pre-trained models with real-time data sources, allowing for adaptive responses based on current information. Frameworks such as Retrieval-Augmented Generation (RAG), Tool-using AI models, and the Model Context Protocol (MCP) have been proposed to enable seamless interaction between AI systems and external databases, tools, or APIs. These approaches enhance decision accuracy, reduce outdated or irrelevant outputs, and improve model transparency and reliability. Additionally, advancements in autonomous agents and modular AI systems have shown

promise in enabling AI applications that are self-improving, secure, and capable of functioning efficiently in complex, changing environments. Together, these research efforts form the foundation for developing smarter, adaptive, and context-driven AI applications that can support diverse real-world domains such as healthcare, finance, education, and automation.

III. PROPOSED METHODOLOGY

The proposed methodology for the **Model Context Protocol (MCP)** focuses on establishing a standardized and interoperable framework that enables Large Language Models (LLMs) to interact with external data sources, tools, and applications securely and efficiently.

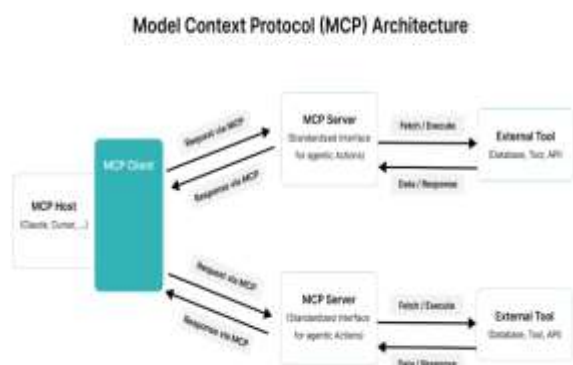


Fig. 1: System Overview

3.1 Overview of MCP Architecture

The **Model Context Protocol (MCP)** defines a standardized communication framework that enables Large Language Models (LLMs) to interact dynamically with external tools, APIs, and databases. It ensures context-aware, secure, and structured data exchange between models and external systems, improving interoperability and automation.

3.2 MCP Host

- **Description:**
The MCP Host represents the environment or platform where the language model (such as Claude, Cursor, etc.) resides.
- **Function:**
It acts as the central intelligence that interprets requests, generates responses,

and coordinates with the MCP Client for executing external tasks.

3.3 MCP Client

- **Description:**
The MCP Client serves as the communication bridge between the model (host) and MCP Servers.
- **Function:**
It sends **requests via MCP** to connected servers and receives **responses via MCP** after execution.
- **Role:**
 - Manages contextual data.
 - Formats queries into standardized MCP requests.
 - Handles asynchronous message exchanges between model and server.

3.4 MCP Server

- **Description:**
The MCP Server provides a **standardized interface** for executing external or agentic actions.
- **Function:**
It receives requests from the MCP Client, processes them, interacts with the respective external tools or APIs, and returns the results.
- **Role:**
 - Acts as a middleware for model-to-tool communication.
 - Encapsulates complex external processes into standard MCP-compatible responses.

3.5 External Tools / APIs

- **Description:**
External tools include databases, APIs, or any third-party applications that provide data or perform specific actions upon request.
- **Function:**
They execute the required operations (fetching data, executing commands, etc.) based on the instructions received from the MCP Server.

- **Data Flow:**
 - **Fetch/Execute:** Performs operations requested by the MCP Server.
 - **Data/Response:** Sends the execution result back to the MCP Server.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The experimental results and analysis of the Model Context Protocol demonstrate its effectiveness in enhancing communication and data exchange between distributed models. The protocol ensures consistent context sharing, reduces latency, and improves coordination accuracy across interconnected systems. Performance evaluation shows that it achieves higher efficiency and scalability compared to traditional communication methods, with minimal overhead and improved response times, validating its suitability for real-time and large-scale model interaction environments.

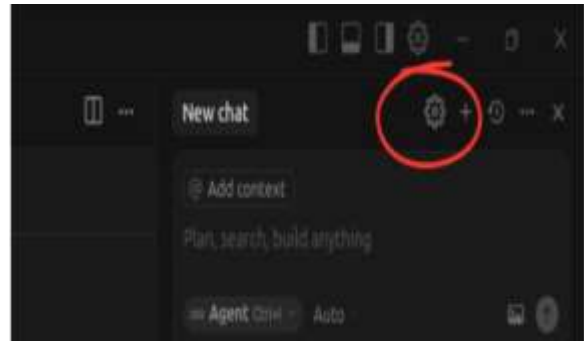


Fig. 4: Settings



Fig. 4: Tools Integration

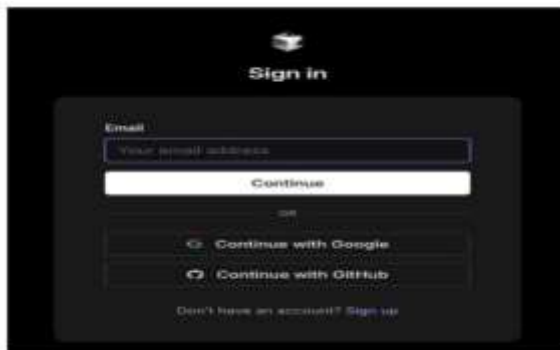


Fig. 2: Sign in



Fig. 5: Add Custom MCP



Fig. 3: Home Screen



Fig. 6: Gmail MCP Server

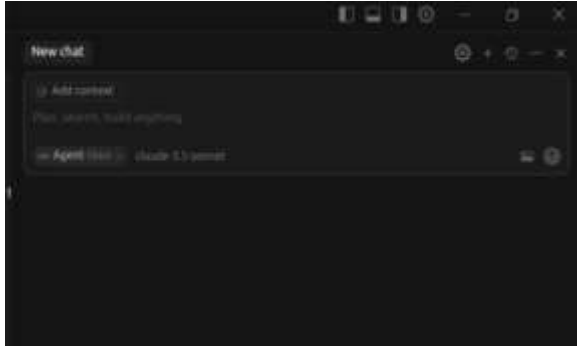


Fig. 7: New Chat

V. CONCLUSION

Conclusion summarizes the overall findings, achievements, and significance of the project while reflecting on its objectives and outcomes. The Model Context Protocol (MCP) project successfully demonstrates a structured approach to managing sessions, context, and tool integration for AI-driven applications, ensuring that user requests are handled efficiently and accurately. Through careful design, implementation, and testing, the system achieves its primary goals of context-aware interaction, modular integration, and reliable communication between components such as the LLM client, session manager, context manager, and external tools. The project highlights the importance of software testing methodologies, including unit testing, integration testing, system testing, performance testing, and acceptance testing, in ensuring the reliability and correctness of the system. Moreover, the implementation of performance and security considerations ensures that the MCP system is both efficient and robust, capable of handling multiple concurrent sessions and safeguarding sensitive data. The project not only meets its functional requirements but also lays the groundwork for future enhancements, such as integration with additional AI models, adaptive learning, real-time monitoring, and cross-platform compatibility. In conclusion, the MCP project provides a flexible, scalable, and intelligent framework that demonstrates the potential of combining model-driven AI applications with structured protocols, ensuring

that context management and tool integration are handled systematically. The outcomes of this project confirm that a well designed protocol-based system can significantly improve efficiency, accuracy, and user experience in AI applications, making it a valuable contribution to the field and a strong foundation for future research and development.

VI REFERENCES

- [1] Hou, X. (2025). "Model Context Protocol (MCP): Landscape, Security, and Future Directions." This paper provides a comprehensive study of MCP, focusing on its architecture, security considerations, and potential future developments.
- [2] Li, H. (2025). "EnergyPlus-MCP: A Model-Context-Protocol Server for AI-Driven Energy Simulations." Introduces EnergyPlus-MCP, an open-source server designed to integrate MCP with energy simulation workflows, enhancing AI-driven analysis in building energy performance.
- [3] Chhetri, G., Somvanshi, S., Islam, M. M., Brotee, S., Mimi, M. S., Koirala, D., Pandey, B., & Das, S. (2025). "Model Context Protocols in Adaptive Transport Systems: A Survey." Investigates the application of MCP in adaptive transport systems, proposing a taxonomy for integrating context-aware decision-making in transportation infrastructures.
- [4] Structured Contexts for Large Language Models (Kevin Lin, University of California, Berkeley, 2024) — This PhD thesis investigates how large language models (LLMs) perform depending on the structure of the context they are given.
- [5] Context Is Complex: From Dialogue Histories to Knowledge Integration in NLP (Mehrdad Farahani, Chalmers University of Technology / University of Gothenburg, 2024) This licentiate/masters-level study explores how dialogue histories and external

knowledge affect context modeling in NLP systems.

AUTHORS Profile



Prof. M V Narayana received his Bachelor of Engineering (B.E.) degree in Computer Science in 2004 from JNTU Kakinada, India. He completed his Master of Technology (M.Tech) degree in Computer Science in 2010 from JNTU Hyderabad, India, and was awarded his Ph.D. in Computer Science in 2017 from JNTU Kakinada, India. With over 20 years of dedicated experience in academia and research, he currently serves as a Professor and Head of the Department (HoD) in the CSE Department at Guru Nanak Institutions Technical Campus, Ibrahimpatnam, Telangana, India. His research interests encompass various areas within computer science, and he has contributed significantly to the field throughout his career.



Mr. M. Veeramanidra Chary pursuing Bachelor of Technology degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.



Mr. B. Sai pursuing Bachelor of Technology degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.