Model Context Protocol (MCP) Server

---

1. Introduction

The **Model Context Protocol (MCP)** is an open protocol that standardizes how Large Language Models (LLMs) interact with external tools, data sources, and services. Instead of tightly coupling an AI application with APIs, databases, or business logic, MCP introduces a **clean client–server abstraction**.

An **MCP Server** exposes capabilities (called *tools*) that an LLM or AI agent can discover and invoke in a structured, secure, and predictable way.

MCP is especially relevant in **LLM, AI, and GenAI systems** where models must:

- Call APIs
- Query databases
- Execute business logic
- Access enterprise systems

---

2. Why MCP Exists

Before MCP, integrations were handled using:

- Custom function-calling logic
- Plugins tied to a specific LLM vendor
- Hard-coded API wrappers

This caused:

- Vendor lock-in
- Tight coupling
- Difficult maintenance

MCP solves this by:

- Creating a **vendor-neutral protocol**
- Separating *AI reasoning* from *tool execution*
- Allowing tools to be reused across models and platforms

---

3. MCP Architecture Overview

High-Level Components

1. **MCP Server**
    - Hosts tools
    - Implements business logic
    - Runs independently
2. **MCP Client**
    - Used by LLM applications
    - Sends structured requests
    - Receives structured responses
3. **LLM / AI Agent**
    - Decides *when* to call a tool
    - Uses tool results to generate responses

---

4. What Is an MCP Server

An **MCP Server** is a process that:

- Registers tools
- Defines input and output schemas
- Listens for requests from MCP clients
- Executes logic and returns results

Each tool is:

- Strongly typed

- Self-describing
- Independent of the LLM

---

5. Core Concepts

5.1 Tools

A **tool** is a callable function exposed by the MCP server.

Example:

- get_customer_balance
- search_documents
- create_ticket

Each tool has:

- Name
- Description
- Input parameters
- Output

---

6. Real-World Use Cases

6.1 Enterprise Data Access

- HR systems
- CRM (Salesforce)
- ERP (SAP)

LLMs can securely retrieve data without direct DB access.

6.2 AI Assistants

- IT helpdesk bots
- Finance assistants
- Student information systems

Tools encapsulate business rules.

6.3 GenAI Workflows

- Document processing
- Approval workflows
- Automated reporting

MCP tools act as workflow steps.

6.4 Multi-Model Strategy

Same MCP server works with:

- GPT-4o
- Claude
- Open-source LLMs

No rewrite needed.

---

7. MCP vs RAG vs Plugins

Aspect	**MCP**	**RAG**	**Plugins**
Purpose	Tool execution	Knowledge retrieval	Vendor-specific tools
Vendor Neutral	Yes	Yes	No
Structured Calls	Yes	No	Partial
Enterprise Ready	Yes	Yes	Limited

---

8. Benefits of MCP

- Loose coupling
- Strong typing
- Security isolation

- Reusability
- Vendor neutrality

---

9. When to Use MCP

Use MCP when:

- LLM must perform actions
- Business logic must stay outside the model
- Multiple AI platforms are involved

Do NOT use MCP for:

- Static knowledge (use RAG instead)
- Simple prompt-only tasks

---

10. Summary

An MCP Server:

- Acts as a bridge between LLMs and real systems
- Enables scalable, secure, and maintainable AI applications
- Is a foundational building block for enterprise GenAI

MCP is not a replacement for RAG or prompts — it **complements them**.

---

11. Key Takeaway

LLMs think. MCP servers act.