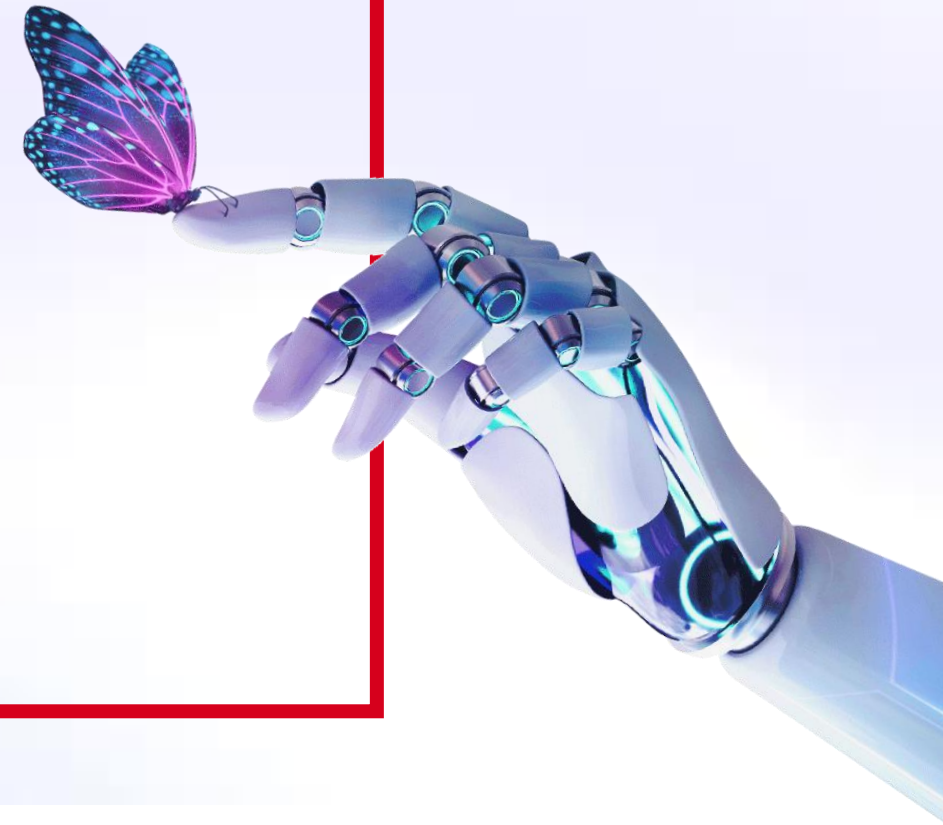




# Build AI Agent tools using CSharp MCP (Model Context Protocol) SDK



Thang Chung

May 10, 2025



**Nash  
Tech.**

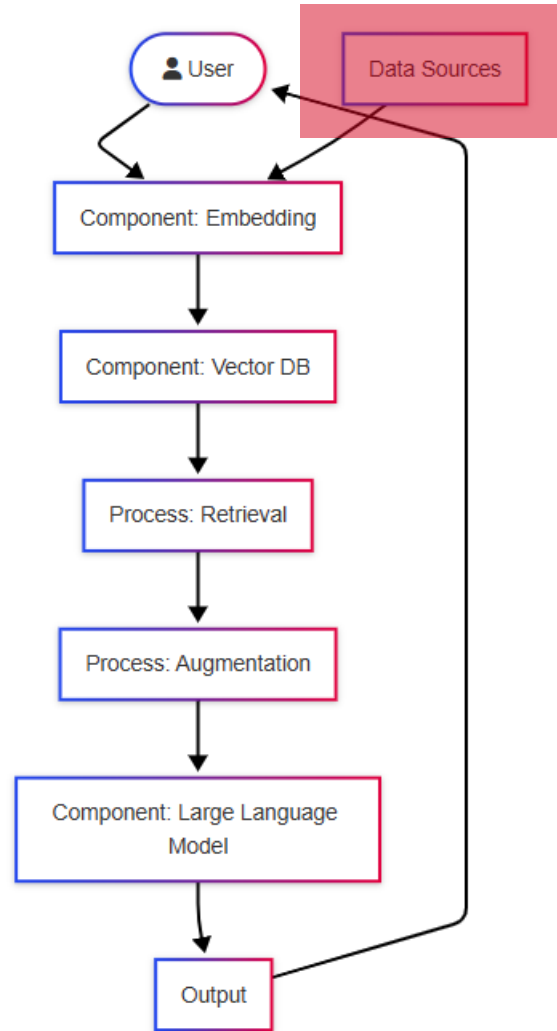
# Agenda

1. RAG, AI Agents, and Agentic RAG
2. MCP – Model Context Protocol with CSharp SDK
3. Demo
4. Appendix: MCP for enterprise applications.
5. Q&A

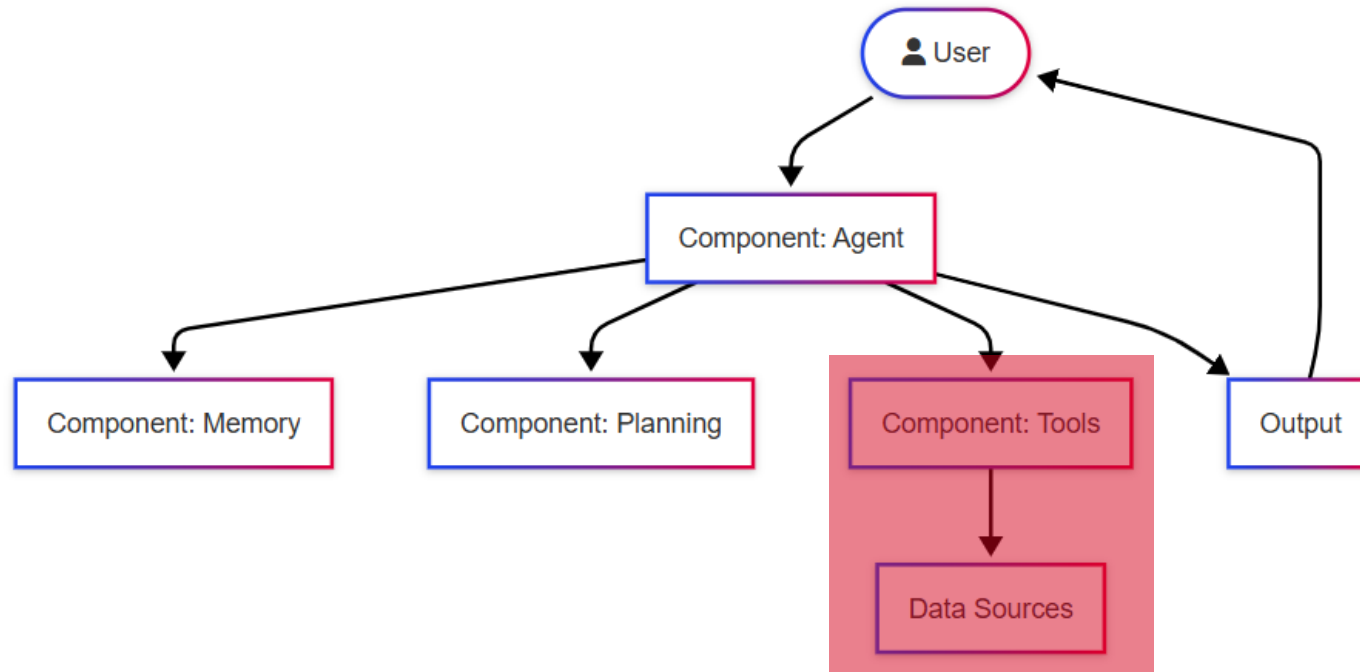
# RAG, AI Agents, and Agentic RAG



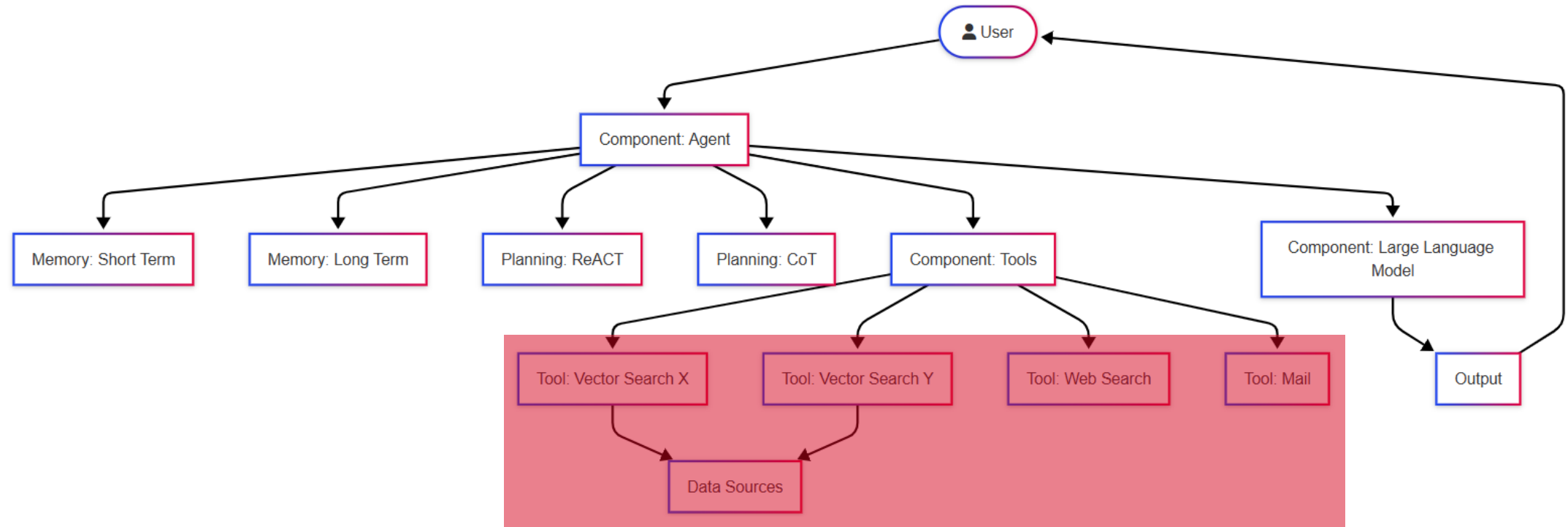
# Retrieval-Augmented Generation (RAG)



# AI Agents



# Agentic RAG

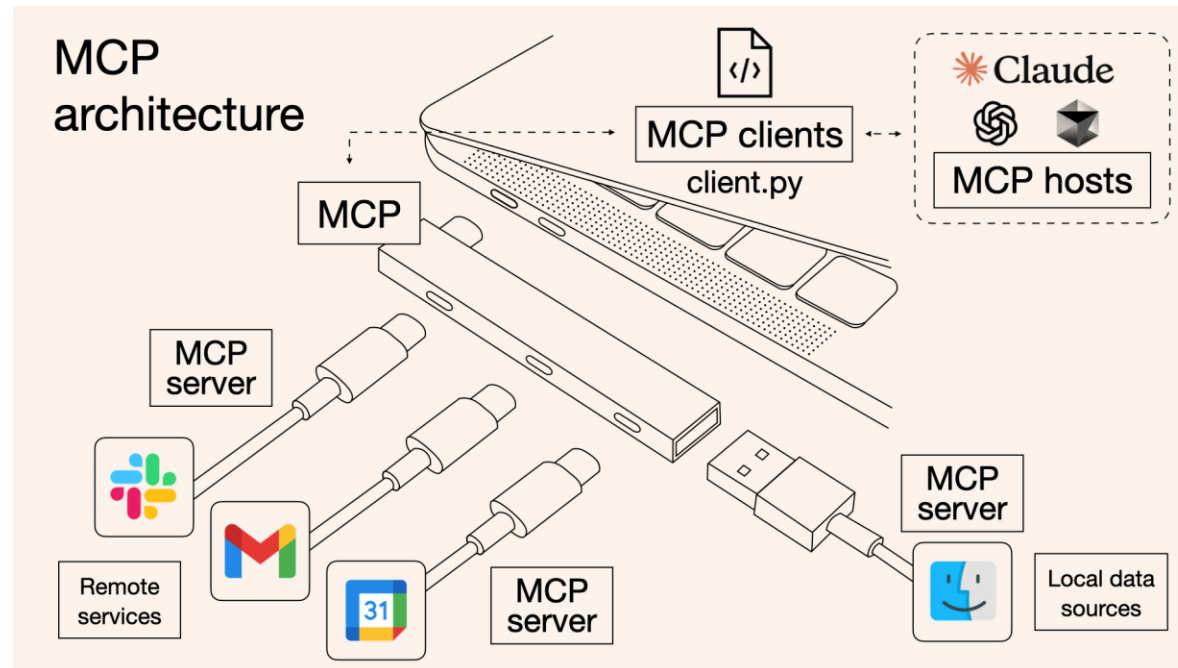


# MCP – Model Context Protocol



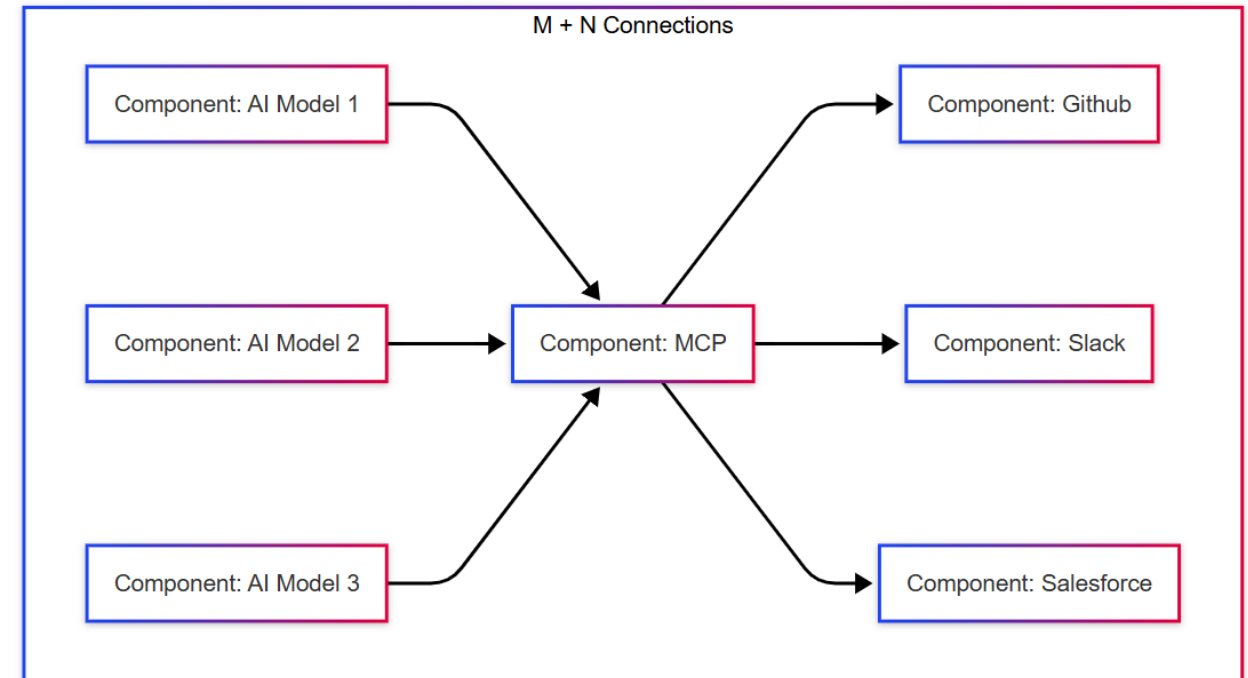
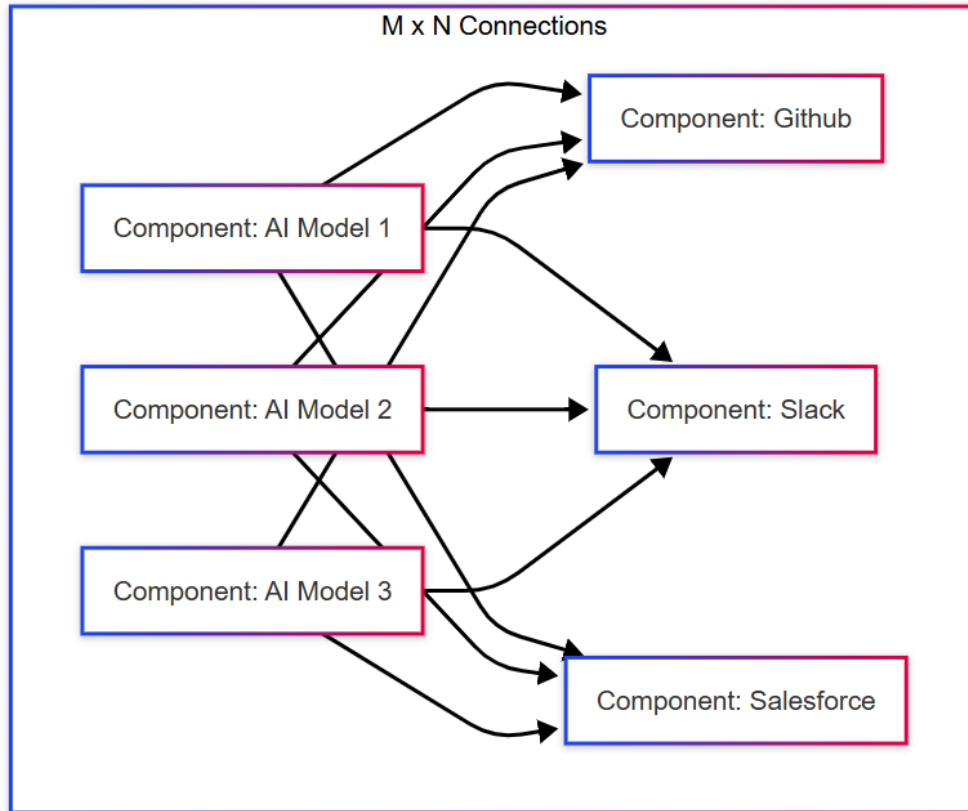
# What is MCP?

- MCP is an **open protocol** that **standardizes how applications provide context to LLMs**. Think of MCP like a USB-C port for AI applications. Just as USB-C provides a standardized way to connect your devices to various peripherals and accessories, MCP provides a standardized way to connect AI models to different data sources and tools.
- GitHub: <https://github.com/modelcontextprotocol>
- Home website: <https://modelcontextprotocol.io>





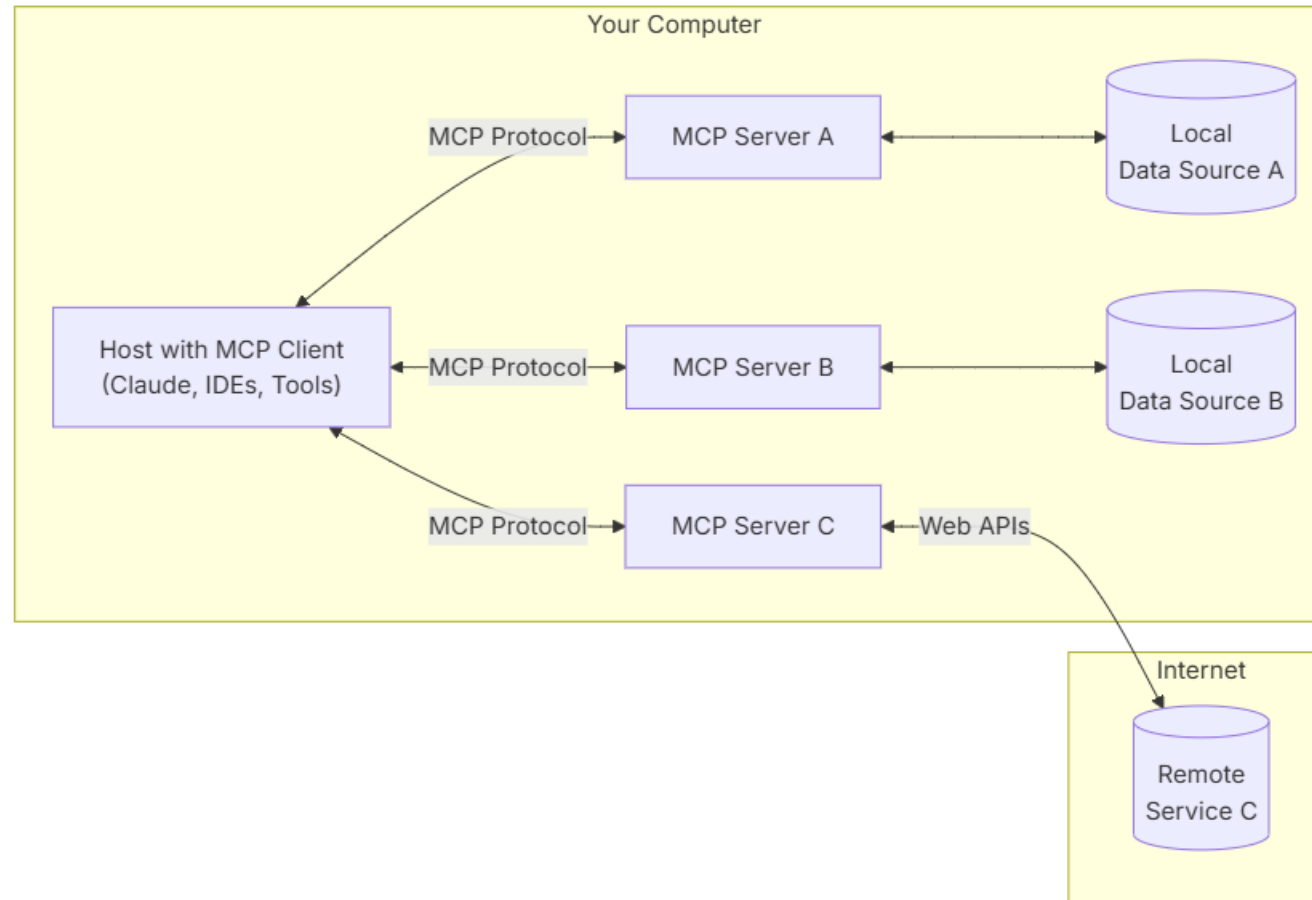
# Before and after MCP



- Each model requires custom integration with each tool
- Inconsistency implementations across models
- Adding new tools/models requires significant effort

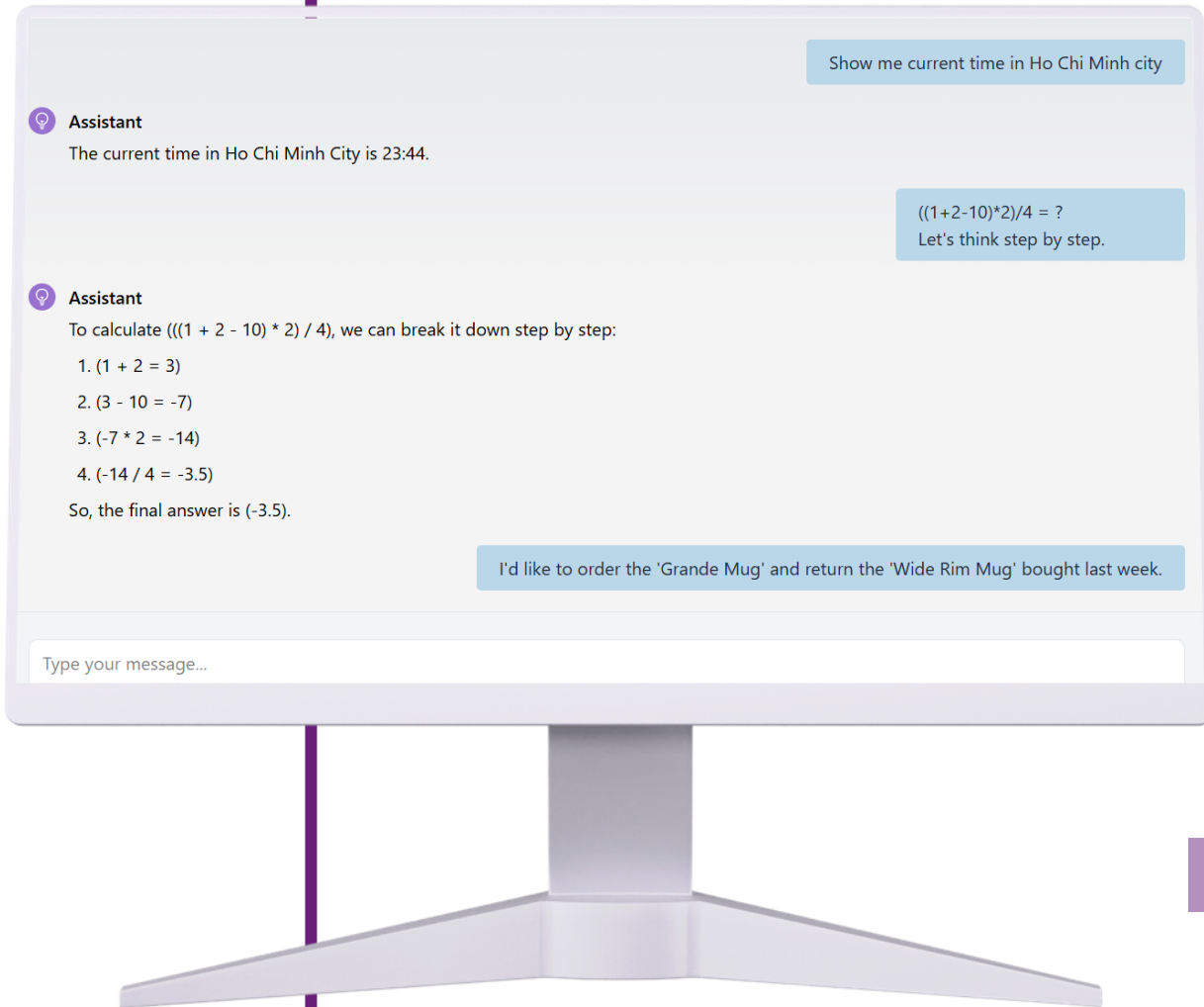
- Single protocol for all model-tool interactions (easy to switch between LLM providers and vendors)
- Consistent implementation across models & tools
- Easy tool discovery and extensibility
- Best practices for securing your data within your infrastructure

# MCP - High level architecture



# MCP C# SDK

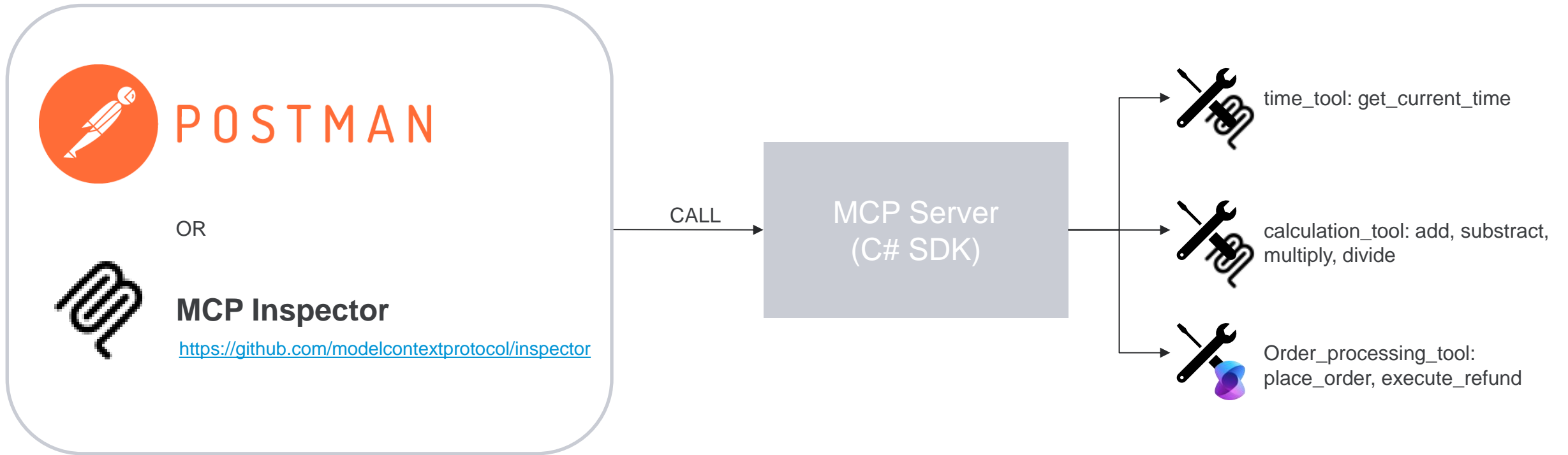
- With MCP C# SDK, developers can now easily **build both servers and clients** that leverage this protocol. This SDK simplifies the implementation process, allowing you to focus on your application's unique features rather than the complexities of protocol handling. Additionally, the SDK includes support for **consuming MCP servers**, enabling developers to **create robust client applications that interact seamlessly with MCP servers**.
- Announcement:
  - <https://devblogs.microsoft.com/dotnet/build-a-model-context-protocol-mcp-server-in-csharp/>
  - <https://devblogs.microsoft.com/blog/microsoft-partners-with-anthropic-to-create-official-c-sdk-for-model-context-protocol>
- Official C# SDK: <https://github.com/modelcontextprotocol/csharp-sdk>



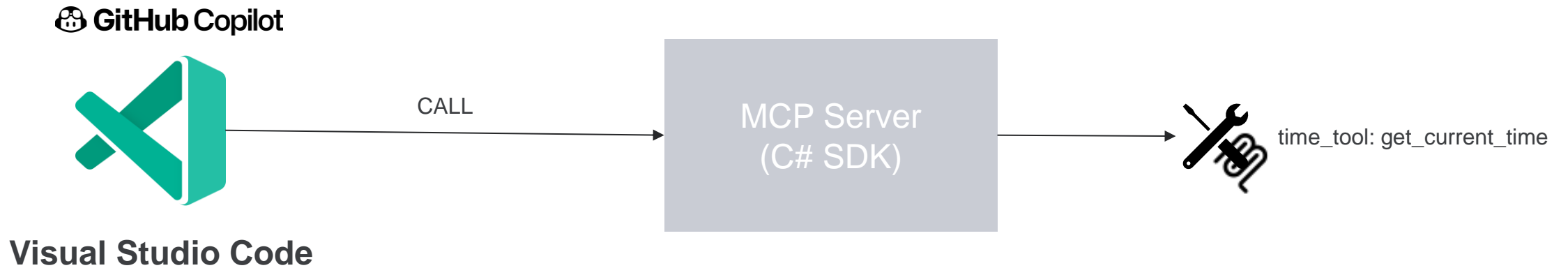
# DEMO

- Demo 1: Simple MCP – Server + Client
- Demo 2: Connect MCP Server with GitHub Copilot Agent mode (MCP Client)
- Demo 3: Create a ChatApp with .NET AI Template + Integrate with MCP Server
- References:
  - <https://github.com/thangchung/mcp-labs>
  - <https://dev.to/thangchung/visual-studio-code-model-context-protocol-mcp-servers-the-first-look-18nb>

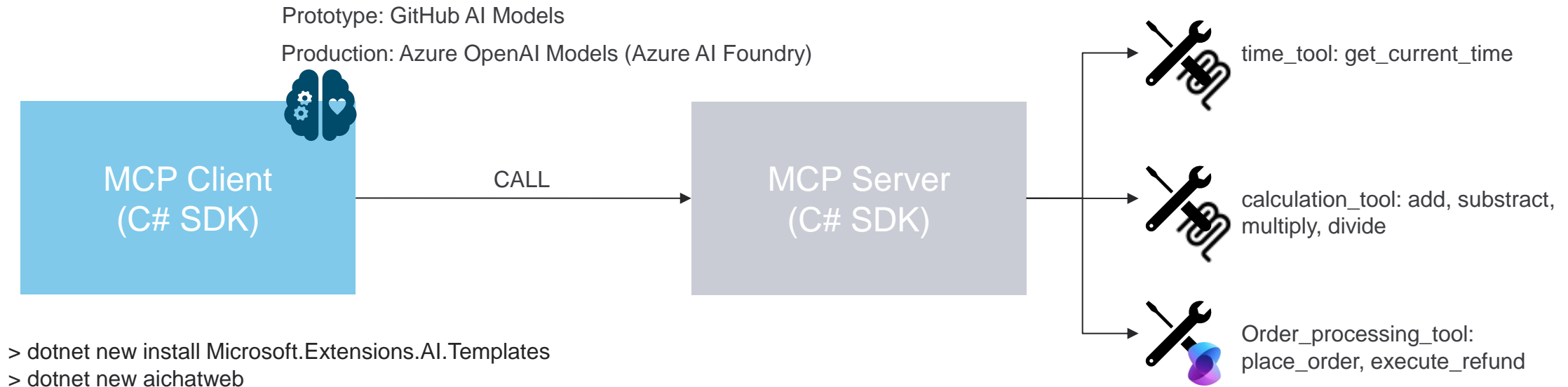
# Demo 1 - Simple MCP – Server + Client



# Demo 2: Connect MCP Server with GitHub Copilot Agent mode (MCP Client)



# Demo 3 - Create a ChatApp with .NET AI Template + Integrate with MCP Server



1. Query:  
Show me current time in Ho Chi Minh city

2. Query:  
 $((1+2-10)*2)/4 = ?$   
Let's think step by step.

3. Query:  
I'd like to order the 'Grande Mug' and return the 'Wide Rim Mug' bought last week.

# Appendix: MCP for enterprise applications

- Scalability: Streamable HTTP
  - <https://modelcontextprotocol.io/specification/2025-03-26/basic/transport>
- Security: OAuth 2.1, OAuth 2.0 Dynamic Client (RFC7591)., OAuth 2.0 Authorization Server Metadata (RFC8414)
  - <https://modelcontextprotocol.io/specification/2025-03-26/basic/authorization>
- Observability (like demo in .NET Aspire)





**Thank you**