Why does MCP exist?

Welcome to our exploration of why **MCP (Model Context Protocol)** exists and the reasons behind its development. In this lesson, we dove into four main challenges faced by LLMs (Large Language Models) and how MCP addresses these. Here's what we covered:

Limitations of LLMs

- **Interactivity Shortcomings**: LLMs excel at generating content but fall short in interacting with the outside world, such as sending emails or executing marketing strategies without external intervention.
- **Tool Calling Evolution**: To bridge this gap, tool calling emerged, allowing LLMs to utilize external functions or 'tools.' However, inconsistencies exist with each framework adopting different mechanisms.

Fragmentation in Tool Development

- **Separate Developments**: Various tool calling scripts are being built independently and don’t have universal API endorsements. This results in duplication of efforts, such as everyone creating their Gmail-LLM integration.
- **Lack of Standardization**: The absence of a standardized approach means created solutions are often non-interoperable or fragile when APIs update.

Server-Dependent Operations

- **External Server Limitation**: Most tool calling mechanisms depend on web servers, inhibiting LLMs from effectively interacting with local systems, limiting their ability to perform tasks like organizing local files or assisting in code development.

MCP as a Solution

- **Standardized Protocol**: MCP serves as a unifying standard, enabling seamless interaction between AI systems and external systems. It simplifies creating connections across different APIs without individual custom scripts.
- **Local and Server Integration**: By facilitating not just server interactions but also local machine operations, MCP expands the functionality and accessibility of AI tools.

Conclusion

In essence, MCP isn't introducing a groundbreaking technology but aligning developers to a common standard that enhances compatibility and promotes widespread adoption. As we move forward in this course, we'll explore MCP's history, architecture, and practical implementation. This foundational understanding prepares us for building our very own MCP servers and clients.

Stay tuned as we dive deeper into the world of MCP. We’re glad you’re here with us and excited to embark on this journey together!