Session 7 – Tools & MCP

Learning Objectives

- Understand why MCP (Model Context Protocol) is needed.
- Learn how MCP standardizes tool integration for agents.
- Explore the architecture: MCP Server ↔ MCP Client.
- See a demo: Python MCP server + Node orchestrator + frontend.
- Connect the concept back to enterprise needs and Agentic RAG.

Beginner-Level Explanation

- Without MCP, every agent integration is custom: messy, fragile, and insecure.
- MCP is like USB for AI tools: a common standard that makes tools plug-and-play.
- An MCP server is a service that exposes tools (like list_files, read_file).
- An MCP client connects to the server and lets an agent call those tools safely.
- This avoids writing new glue code for every tool integration.

Expert-Level Explanation

- MCP is a protocol that decouples tool execution from model reasoning.
- Servers expose schemas for tools: their name, inputs, outputs.
- Clients enforce protocols: ensuring correct calls, input validation, and error handling.
- This reduces attack surface and ensures reproducibility across environments.
- In enterprise settings, MCP provides compliance, monitoring, and standard governance.

Demo Flow

- 1. Start MCP server exposing three tools: list_files, read_file, grep_text.
- 2. Orchestrator connects using mcp-use (Node client).
- 3. UI sends query → Orchestrator decides route.
- 4. If general knowledge: plain LLM response.
- 5. If file/search-related: Orchestrator calls MCP tool and streams result back to UI.

Enterprise Implications

- MCP lets companies wrap internal systems (databases, Git, APIs) as standardized servers.
- Agents can safely interact with these tools without bespoke integrations.
- Audit logs and monitoring are centralized at the MCP layer, improving compliance.
- This approach reduces maintenance overhead and improves portability of AI solutions.

Wrap-Up

- Tools are how agents act; MCP makes tool use safe and standardized.
- Demo showed how MCP server and client connect seamlessly.
- Next: Agentic RAG – combining retrieval with MCP-enabled actions.