MODEL CONTEXT PROTOCOL (MCP) PRINCIPLES

Module 6 — Harwell Prompt Engineering

LEARNING OBJECTIVES

By the end of this module you will be able to:

BRIDGE FROM MODULE 5

What we learned yesterday:

- **RAG** connects AI to documents

The connection:

- RAG: Documents (static)
- MCP: Files, repos, databases (live)
- Both solve “AI doesn’t know your context”

Frame explicitly:

- This section is **orientation** — understanding what MCP is
- Not adoption or rollout guidance
- MCP is still early/evolving

THE PROBLEM: THE SILO PROBLEM

The limitation:

- AI tools (browser, apps) can’t see your local files, repos, or databases
- ❌ You have to manually copy-paste code
- ❌ AI doesn’t know your project structure
- ❌ AI can’t access your database schema
- ❌ Context switching is slow and error-prone

The pain:

- Scenario: “Explain how this module works”
- ❌ You copy-paste files one by one
- ❌ Lose project structure context
- ❌ Can’t reference related files
- ❌ Slow, manual, risky (data privacy)

THE SOLUTION: MCP

MCP = Model Context Protocol

- Standardized protocol for context servers
- ✅ AI can request context through protocol
- ✅ Structured, safe access to files/repos/databases
- ✅ Less manual copy-paste

Key idea:

WITHOUT MCP VS. WITH MCP

Without MCP:

- AI tool can’t see your files
- ❌ You paste code manually
- ❌ No project structure awareness
- ❌ Can’t reference other files
- ❌ Context is lost

With MCP:

- AI tool requests context through MCP protocol
- Context server exposes files/repos/databases
- ✅ AI can “read” project structure
- ✅ Can reference specific files
- ✅ Structured, controlled access

HOW MCP WORKS: THE ARCHITECTURE

Context servers:

- Expose data sources (filesystem, git, database)
- Examples: Filesystem server, Git server, Database server
- Standardized way to access different data sources

Clients:

- AI tools that request context
- Examples: Claude Desktop, IDEs, AI applications
- Request context through MCP protocol

The protocol:

- Standardized request/response format
- Any client can work with any server
- Defines how to request files, list directories, query databases

EXAMPLE FLOW

User asks AI: “Explain this module”

1. AI client requests: “List files in src/main/java/module1”
2. Filesystem server responds: [“File1.java”, “File2.java”]
3. AI client requests: “Read File1.java”
4. Filesystem server responds: File contents
5. AI generates explanation based on retrieved files

Result: AI understands your project structure without manual copy-paste.

SIMPLE ANALOGY

Think of MCP like a library API:

- Client (AI tool) requests information
- Server (context server) provides it
- Protocol defines the request/response format
- Standardized, structured, safe

MCP enables structured access to your systems.

IMPLICATIONS: WHAT THIS MEANS

Implication 1: Safer access

- ✅ Structured, controlled access to files/repos
- ✅ Can limit what AI can see (e.g. “only this module”)
- ✅ Less risky than manual copy-paste

Implication 2: Less copy-paste

- ✅ AI can request context automatically
- ✅ No manual file copying
- ✅ Faster iteration

Implication 3: Still early

- ⚠️ MCP is still evolving
- ⚠️ Not all tools support it yet
- ⚠️ Expect changes and improvements
- ⚠️ This is orientation, not adoption guidance

WHEN TO CONSIDER MCP

Consider MCP when:

- ✅ You need AI to reason over many files
- ✅ You need structured access to repos/databases
- ✅ Manual copy-paste is too slow or risky
- ✅ Your tools support MCP

MCP is still early — watch for adoption and tool support.

SUMMARY

1. **Problem**: AI tools can’t access local files/repos/databases
2. **Solution**: MCP provides standardized protocol for context servers
3. **How it works**: Servers expose data, clients request through protocol
4. **Implications**: Safer, structured access; less copy-paste; still early
5. **Orientation only**: Understanding, not adoption guidance

BRIDGE TO MODULE 7

What we've learned:

- **MCP** connects AI to live systems (files, repos, databases)

What's next:

Module 7: AI APIs — the technical details of calling LLMs programmatically.

MCP uses APIs under the hood — let's understand how those APIs work.

QUESTIONS?

Module 6 — Model Context Protocol (MCP) Principles