MCP in Action → Building Smarter AI-Driven Applications with LLMs and Agents.

Discover how MCP enables LLMs and agents to interact with your application logic.

We'll break down the basics of LLMs and agents, introduce the MCP model, and showcase a practical demo.

Agenda

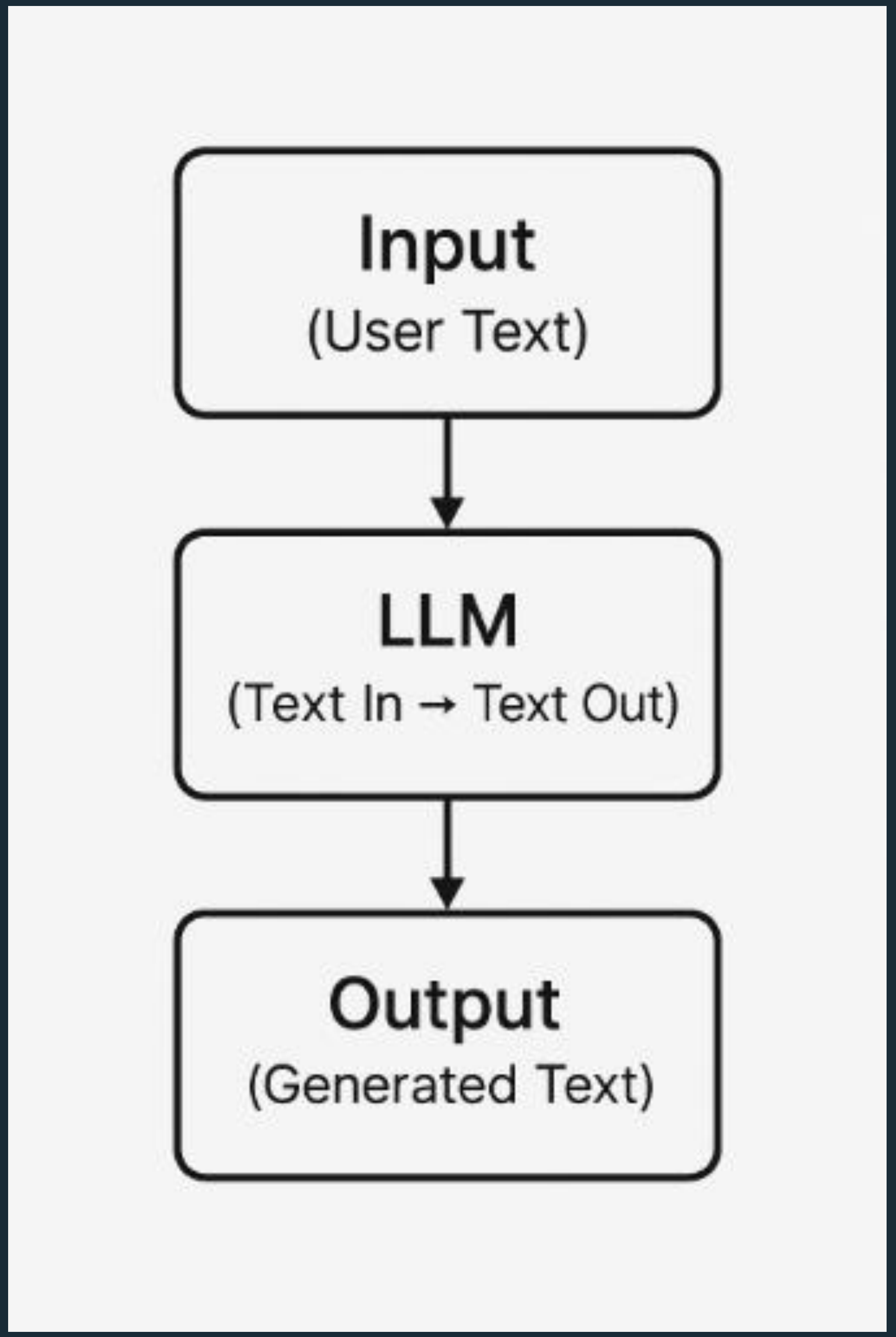
Chapter

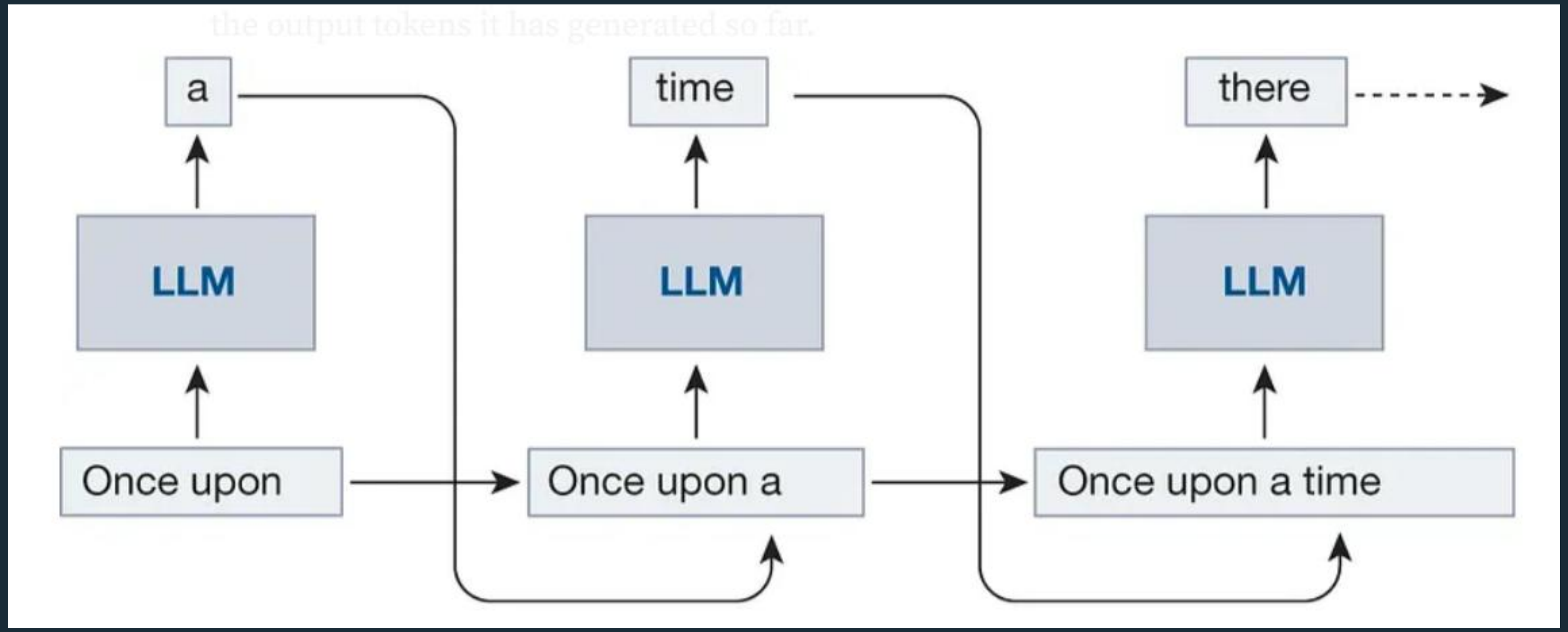
01 The History That Led to MCP

Early LLM Capabilities

From Static Models to Interactive AI

1. **LLMs are enormous neural networks (probabilistic model) that predict the next token/word**
2. **LLMs were originally "text-in, text-out"**
3. **No way to access live data**
4. **No awareness of external systems**
5. **No way to take actions**





Fine-Tuning

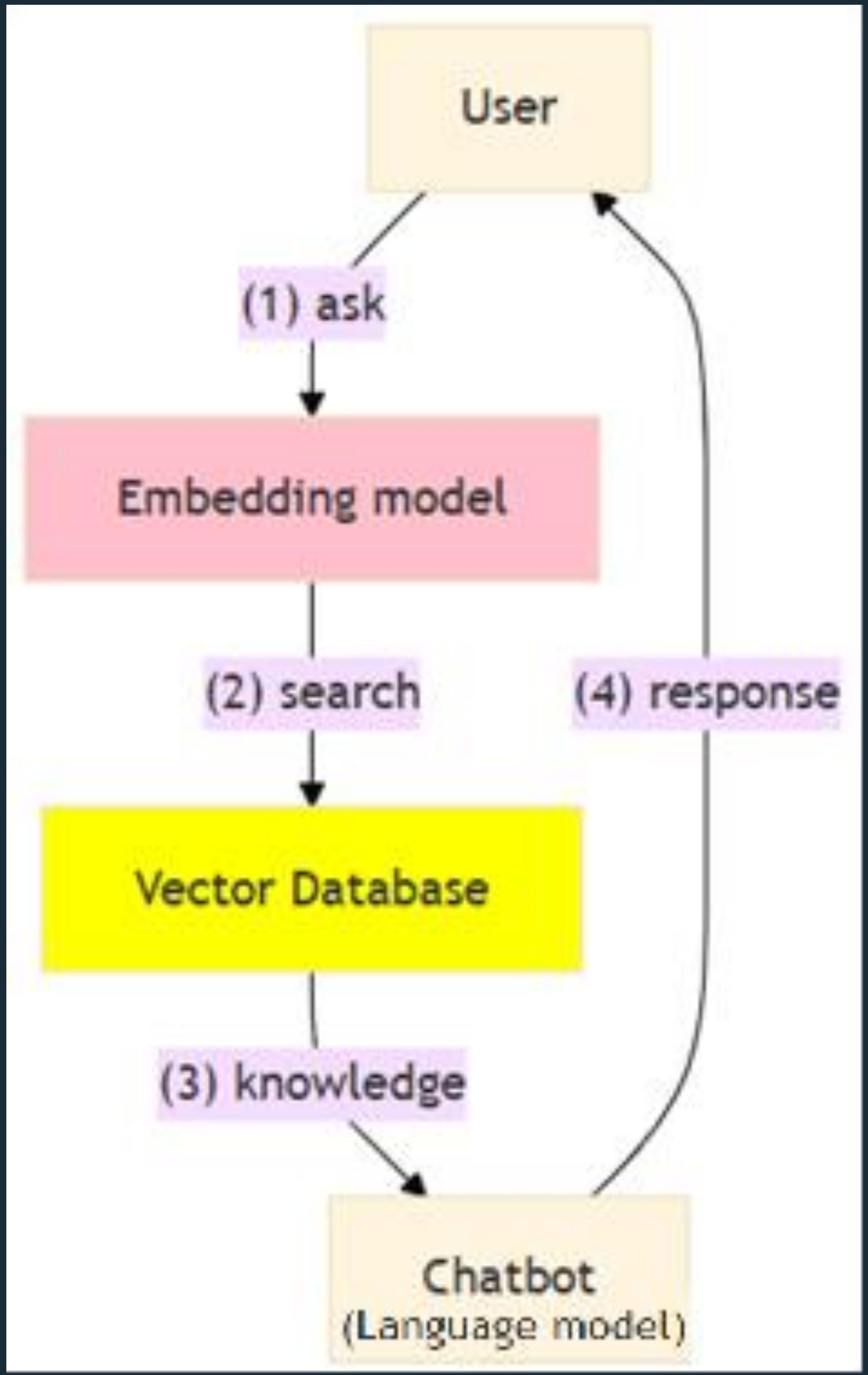
Trains the pre-trained LLM model with our specialized dataset: Helpful, but Limited

1. **Customization via training**
2. **Expensive, slow iteration**
3. **Still no access to external systems**
4. **Still no way to take actions**

RAG

Retrieval-**A**ugmented **G**eneration: Giving LLMs Knowledge

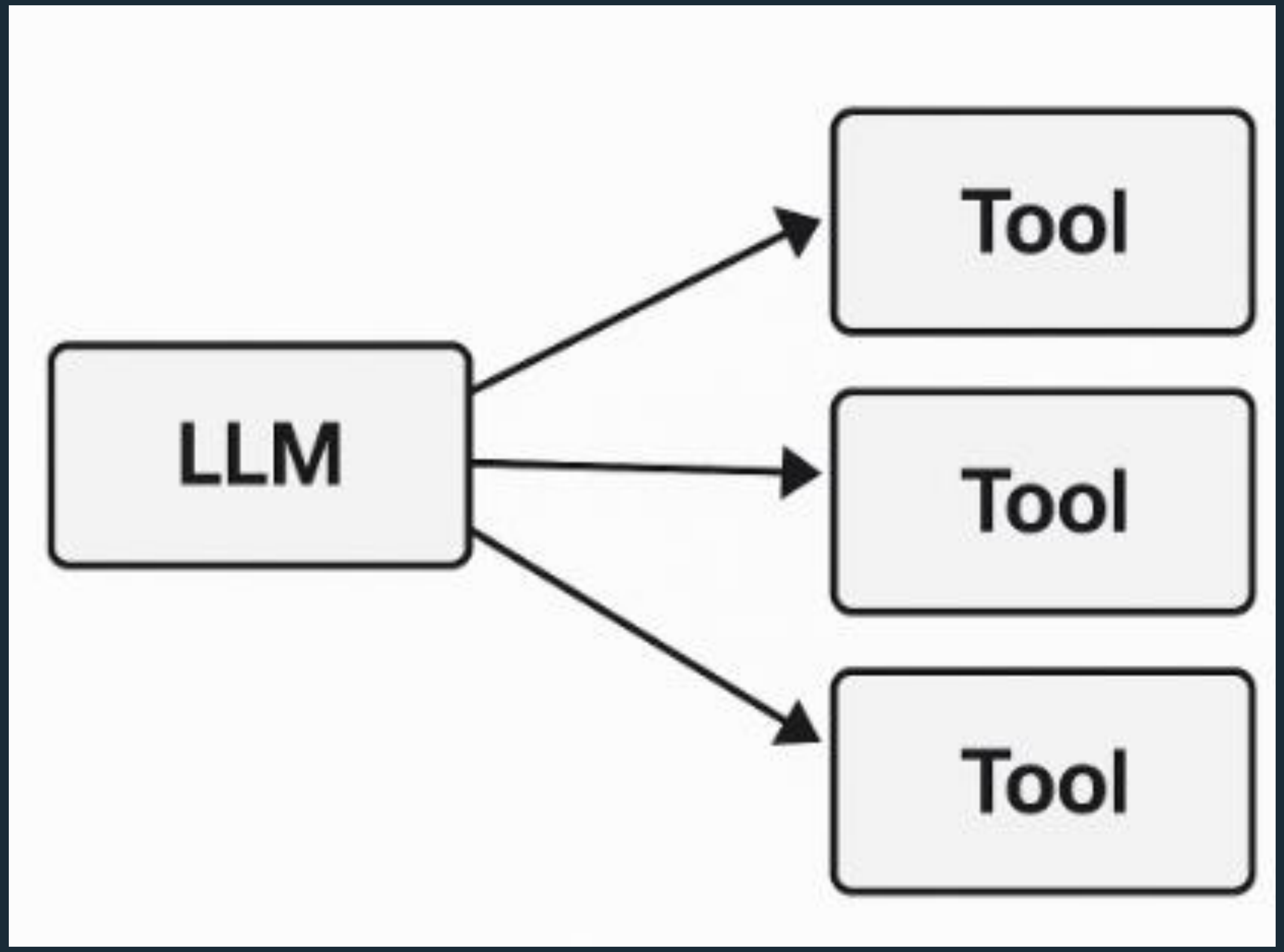
1. **Adds knowledge beyond training**
2. **Still limited to passive retrieval**
3. **Read-only access**
4. **Still cannot act**



Tool Use / Function Calling

LLMs Begin Taking Actions

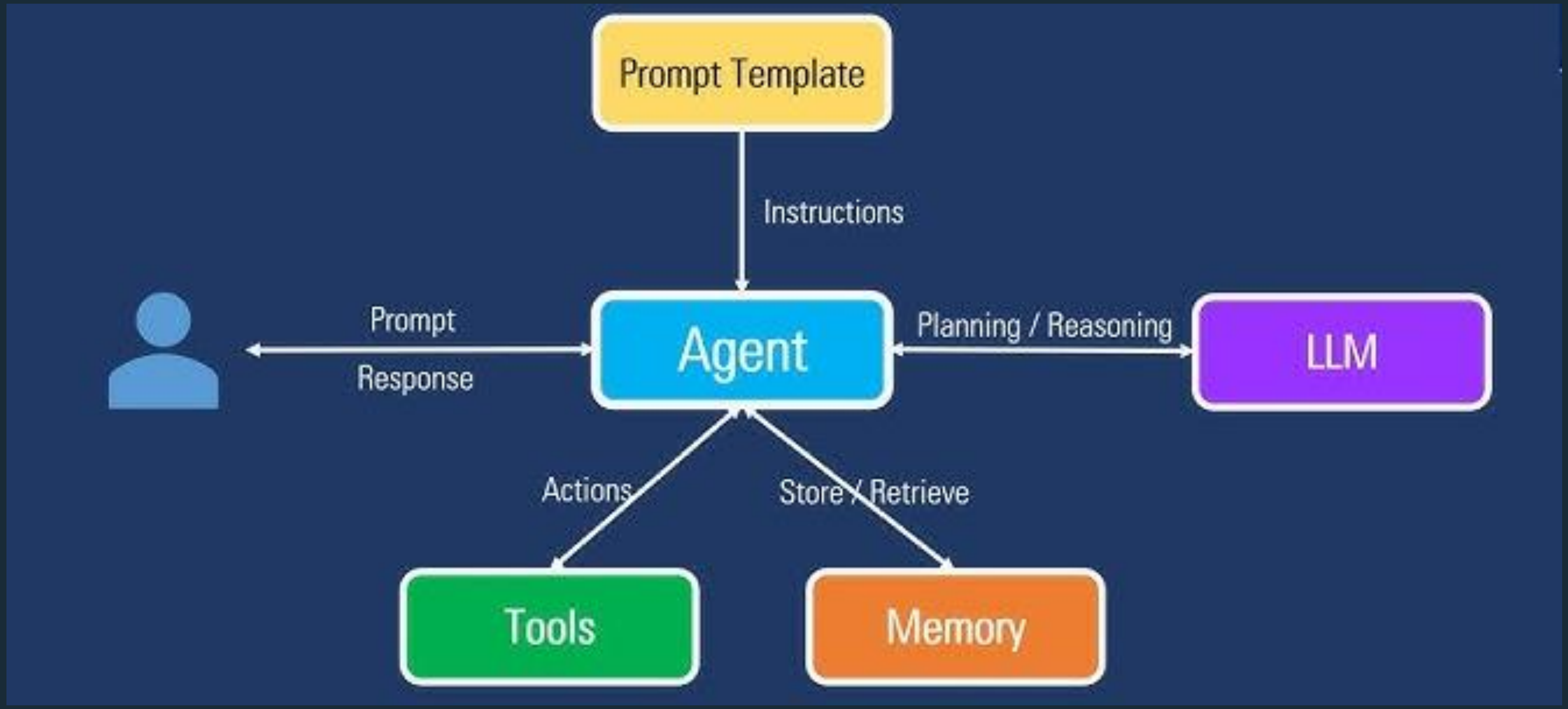
1. **LLMs call functions/tools via JSON schemas**
2. **Enabled real actions (APIs, DB calls, file ops)**
3. **But… every system defined tools differently**
4. **No standard, no permissions, no portability**



Agents

Reasoning + Tools

1. **Multi-step decision-making**
2. **Chains of tool calls**
3. **Increased power = increased integration chaos**
4. **Every tool → custom integration (tools lacked standards)**



The Problem

The Integration Mess

1. **No shared protocol for tools**
2. **No standard permissions**
3. **Hard to build, hard to secure**
4. **Hard to reuse tools across platforms**

Chapter

02 Why MCP

Why MCP Exists

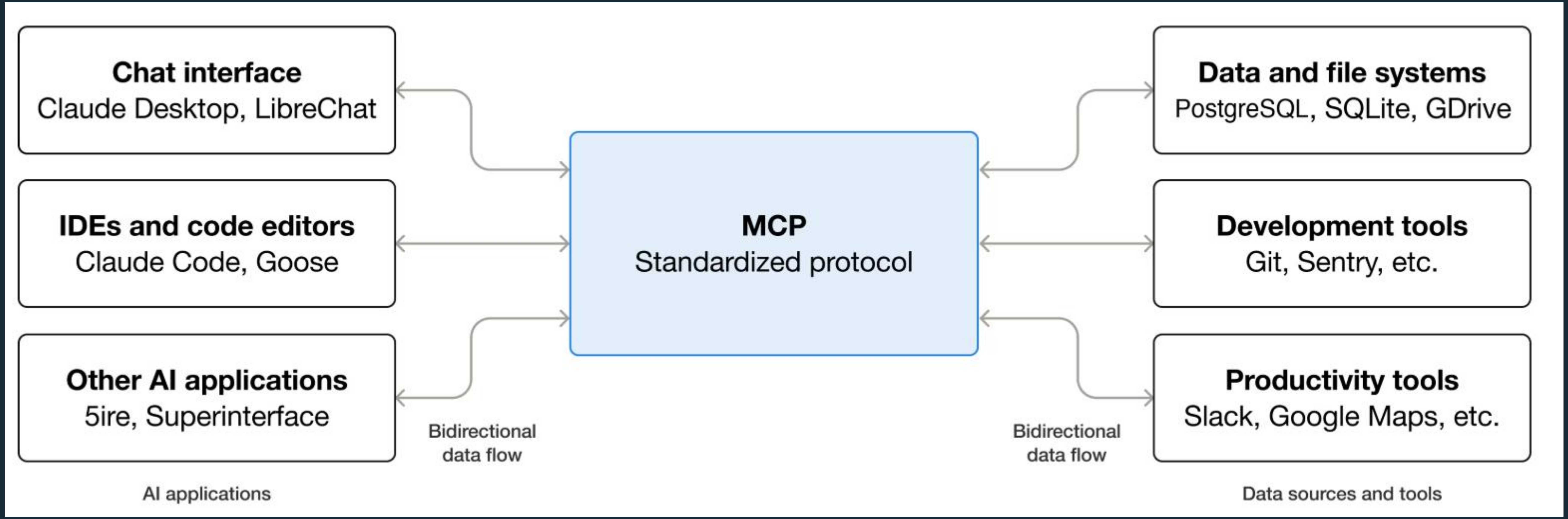
Model **C**ontext **P**rotocol: A Standard for Agent–Tool Interaction
Created by Anthropic: https://www.anthropic.com/news/model-context-protocol

1. **Uniform protocol for tools**
2. **Strong permission + security model**
3. **Works locally or remotely**
4. **Clear access control**
5. **Designed for LLMs and agents**
6. **Portable integrations (one server, many clients)**

MCP Architecture

The MCP Model

1. **Servers expose tools, resources, prompts**
2. **Clients (LLMs/Agents) interact over a shared protocol**



MCP in Practice

How MCP Changes Development

1. **Build once → use across platforms**
2. **Safe control of system resources**
3. **Makes apps "AI-ready"**

Chapter

03 Demo

Demo 1

Using an Existing Node.js MCP Server

1. **Walkthrough Clude Desktop**
2. **MCP Server config JSON file**
3. **Import AirBnb MCP server**

- **All available MCP servers:** https://github.com/modelcontextprotocol/servers
- **AirBnb MCP server:** https://github.com/openbnb-org/mcp-server-airbnb

Demo 2

Create a simple .NET MCP server that shows the local machine time

1. **Code walkthrough**
2. **Install Microsoft.Extensions.AI.Templates template : https://learn.microsoft.com/en-us/dotnet/ai/quickstarts/build-mcp-server**
3. **Show the template in the Visual Studio**
4. **MCP Server config JSON file – dotnet command**
5. **Import LocalTime MCP server**

Demo 3

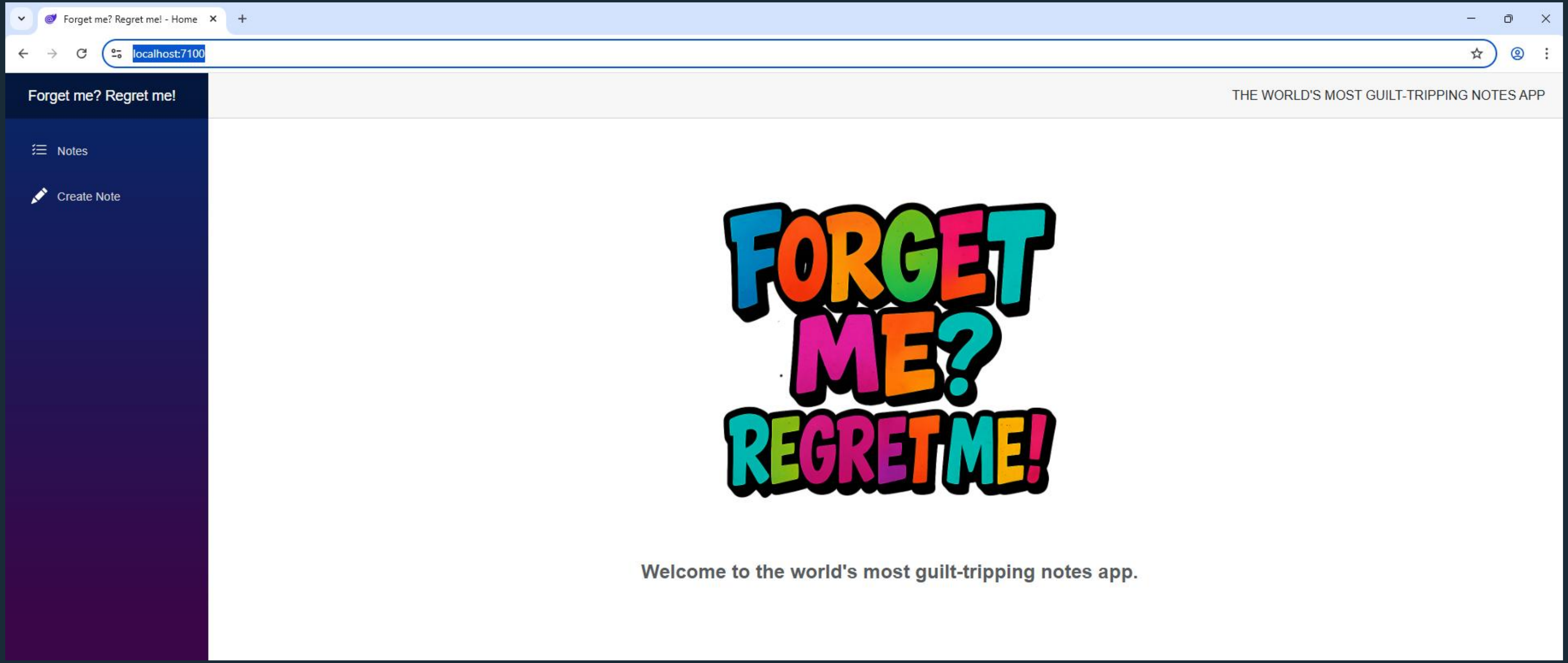
Create a simple .NET MCP server that manipulates the files on the local machine time

1. **Code walkthrough**
2. **Import LocalFile MCP server**

Demo 4

Create a .NET MCP server that uses our application

1. **Application walkthrough**
2. **Code walkthrough**
3. **Import LocalFile MCP server**
4. **Import one more MCP server and see how these 2 are working together**



Welcome to the world's most guilt-tripping notes app.

Resources

1. **Official MCP documentation:** https://modelcontextprotocol.io/docs/getting-started/intro
2. **MCP Implementation:** https://github.com/modelcontextprotocol
3. **MCP Servers:** https://github.com/modelcontextprotocol/servers **or** https://mcp.so/servers
4. **MCP Clients:** https://mcp.so/clients
5. **MCP Csharp SDK:** https://github.com/modelcontextprotocol/csharp-sdk

Thank you!