

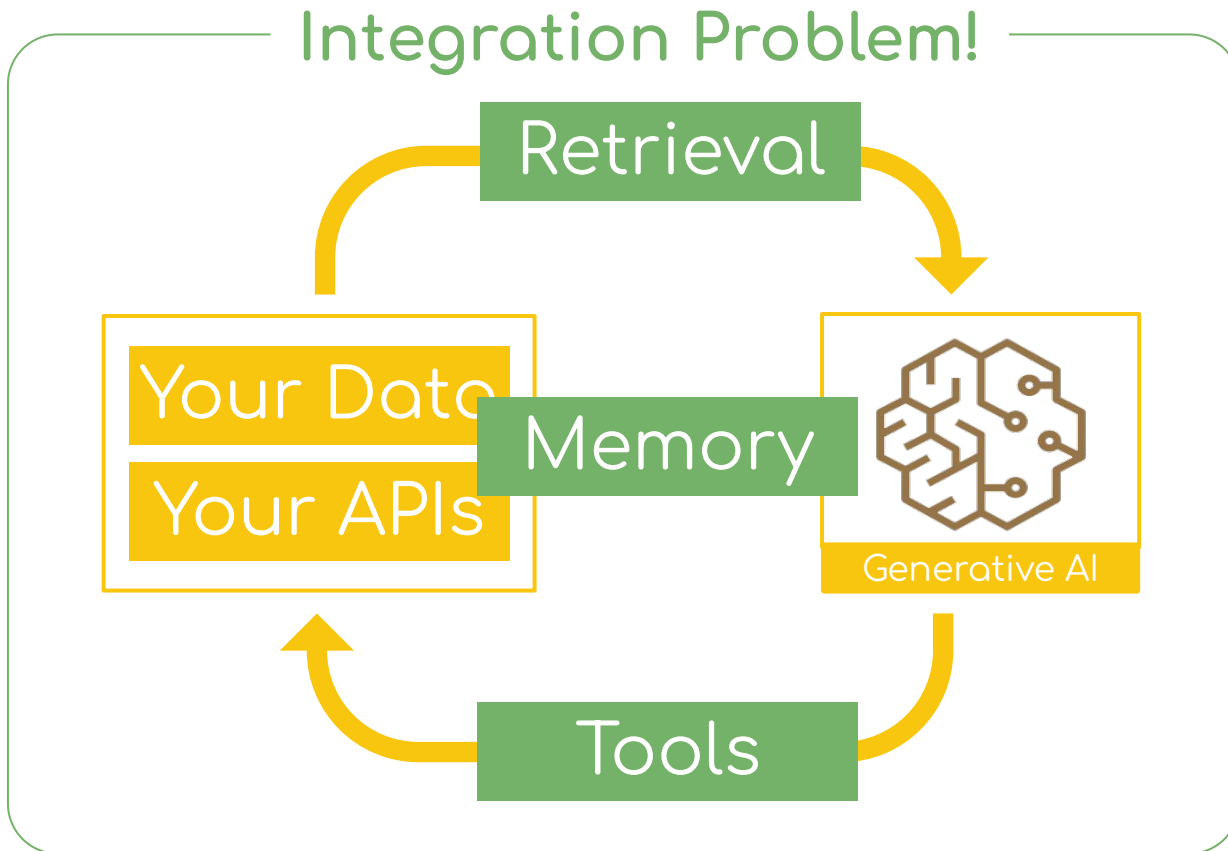


Building Agentic Systems with Spring AI & MCP

Christian Tzolov
Spring AI & MCP Java SDK

 @christzolov ,  @tzolov

Using Generative AI



spring[®] Ai

Models

 OpenAI

 Azure

 stability.ai

 Hugging Face

 ANTHROPIC

 MISTRAL AI

 Gemini Google



 Amazon Bedrock

 MINIMAX

Chat Client

Chat Model

Multimodality

Tool Calling

Struct. Out.

Prompt

Advisors

Memory

Embedding Model

Image Model

Audio Model

Moderation Model

VectorStore



 Pinecone



redis



 drant



Milvus



mongoDB

sandra

 elasticsearch



Azure



SAP HANA



Chroma

ETL

Modular RAG

...

Model Context Protocol (MCP)

Observability

Agentic System

“Leverages an **AI model** to **interact with** its **environment** in order to **solve** a **user-defined task**”

Combines **Planning** & **Actions** to fulfill the tasks



Brain (LLM)

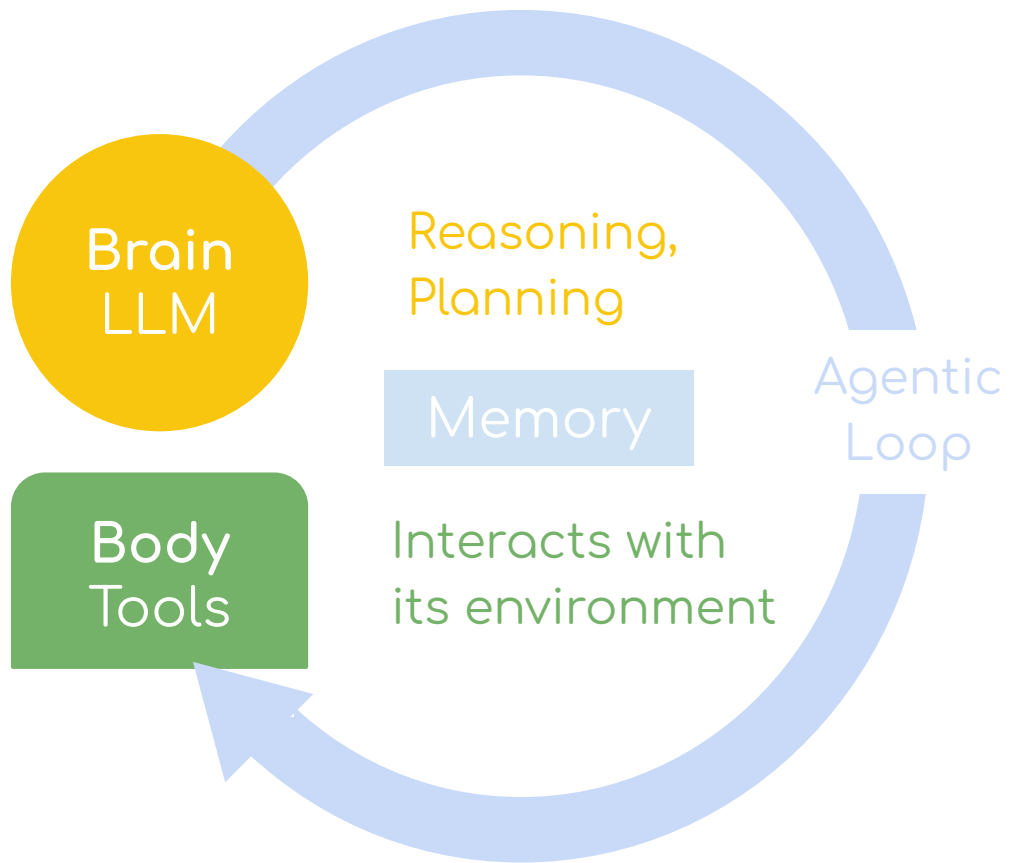
Handles reasoning and planning
Decides which Actions to take

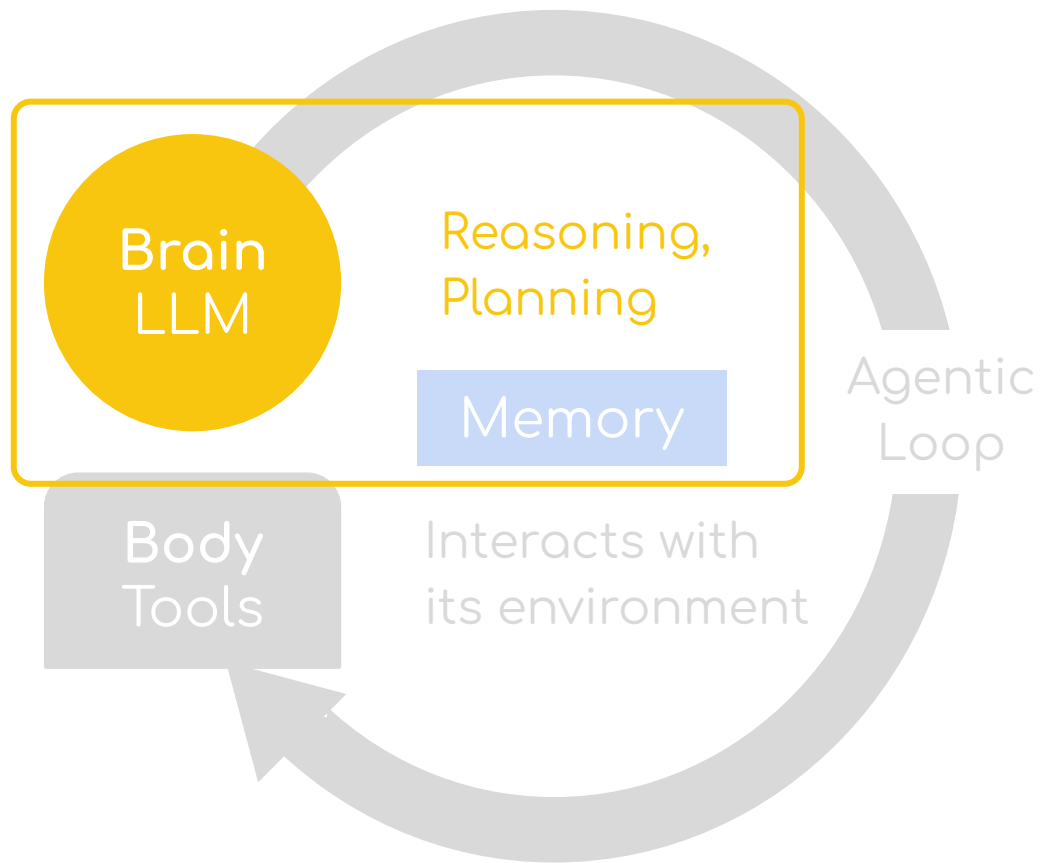


Body (Tools)

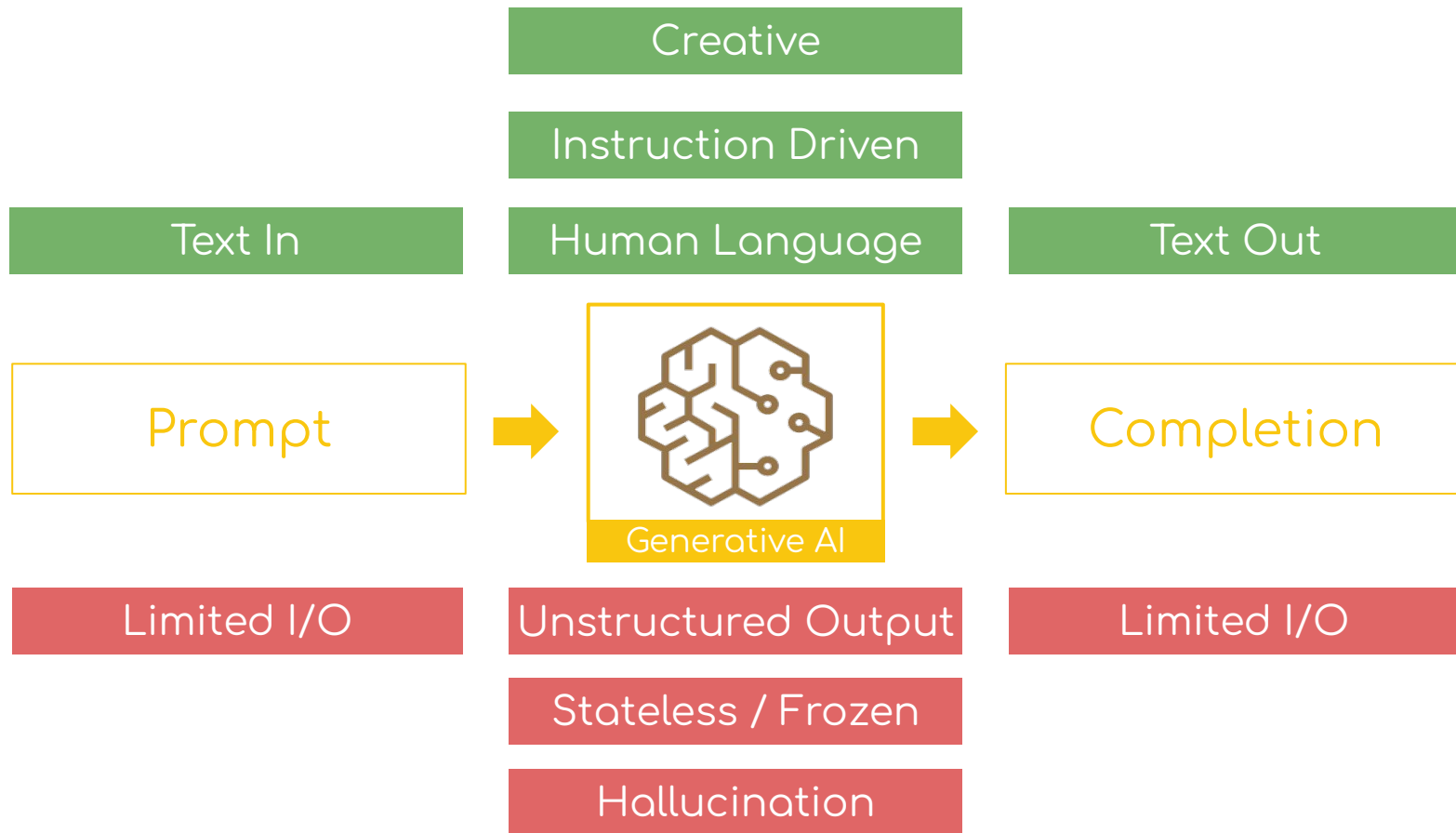
What Agent is equipped to do
Interact with its environment

Agentic System



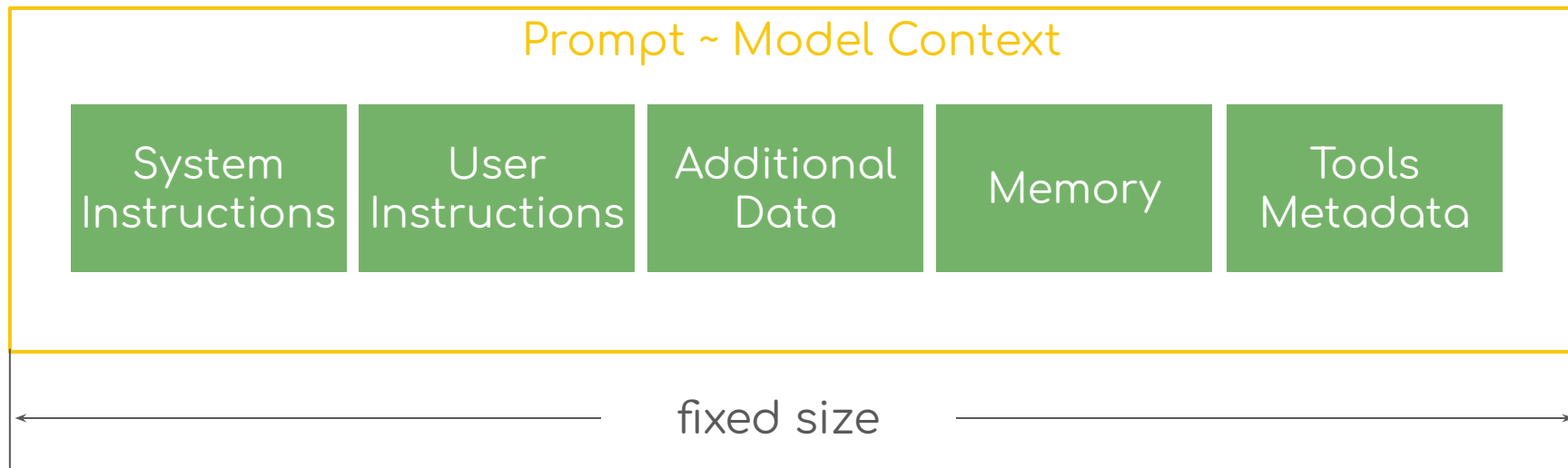


LLM as a System



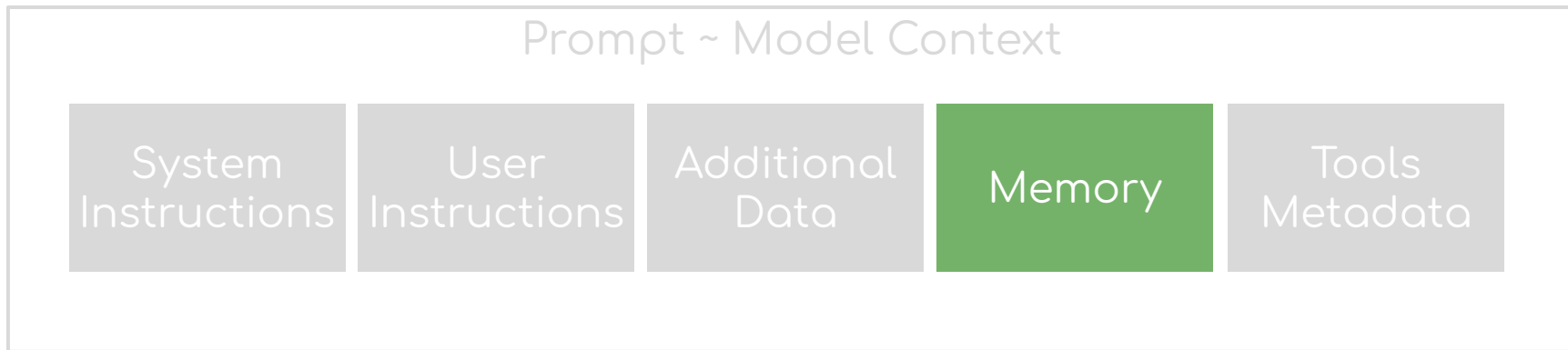
Prompt ~ Model Context

AI Models are only as good as the **Model Context** provided to them

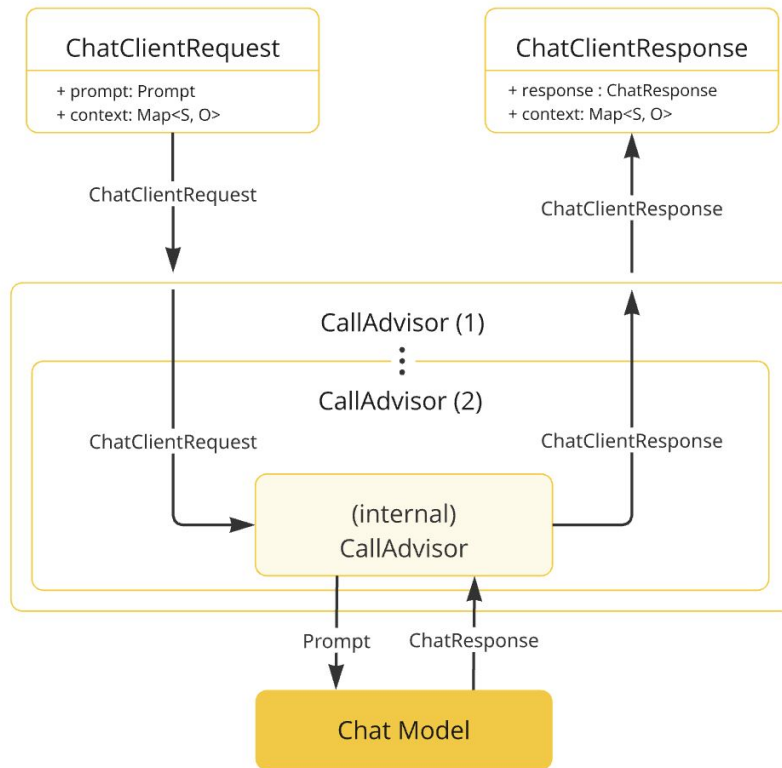


Prompt ~ Model Context

AI Models are only as good as the **Model Context** provided to them

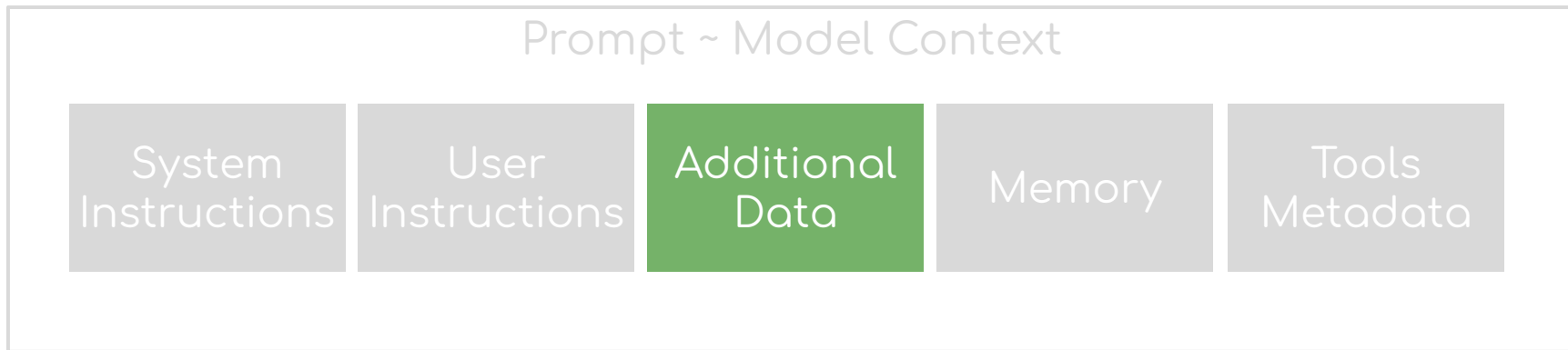


(LLM) Advisors

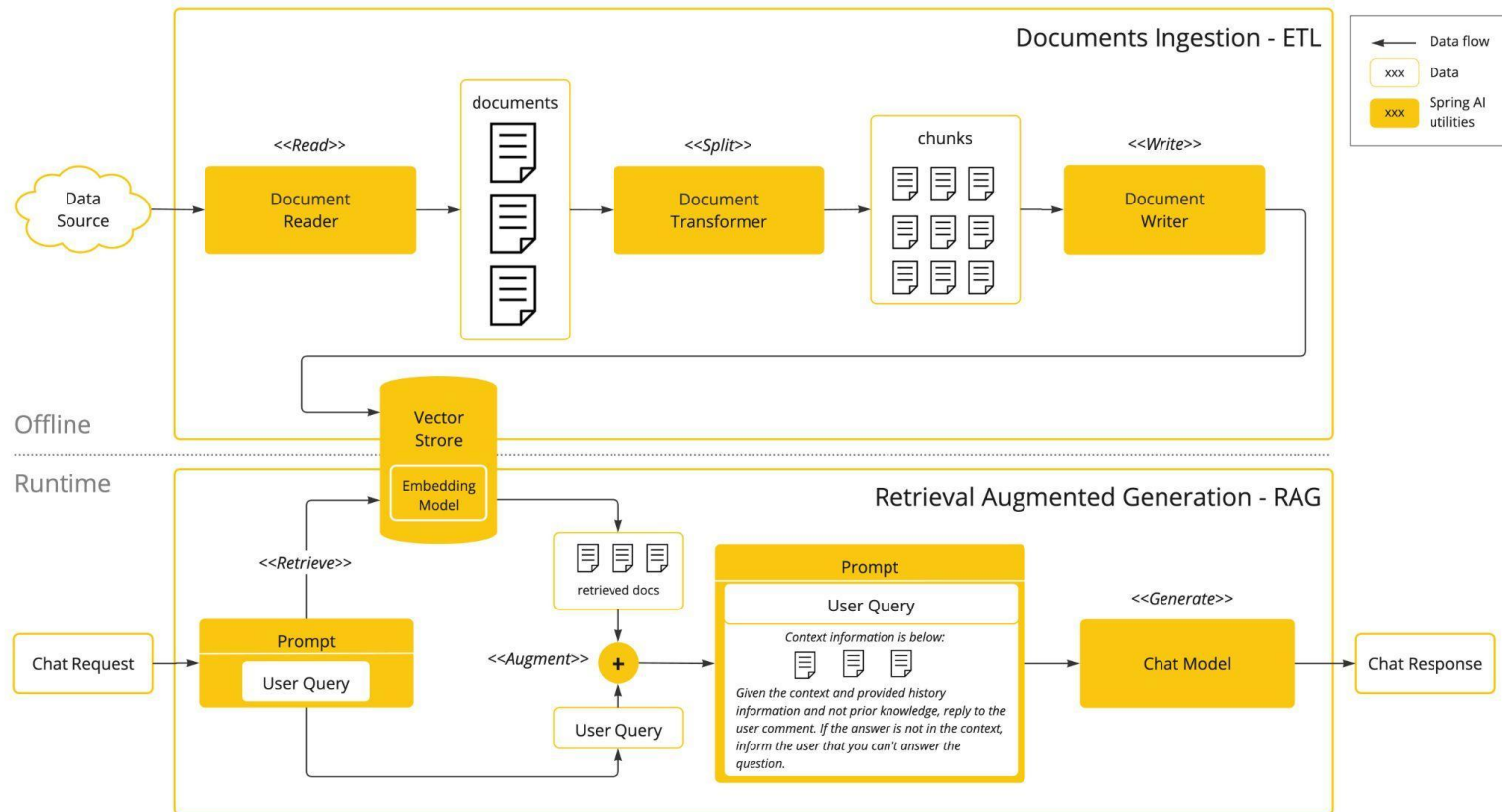


Prompt ~ Model Context

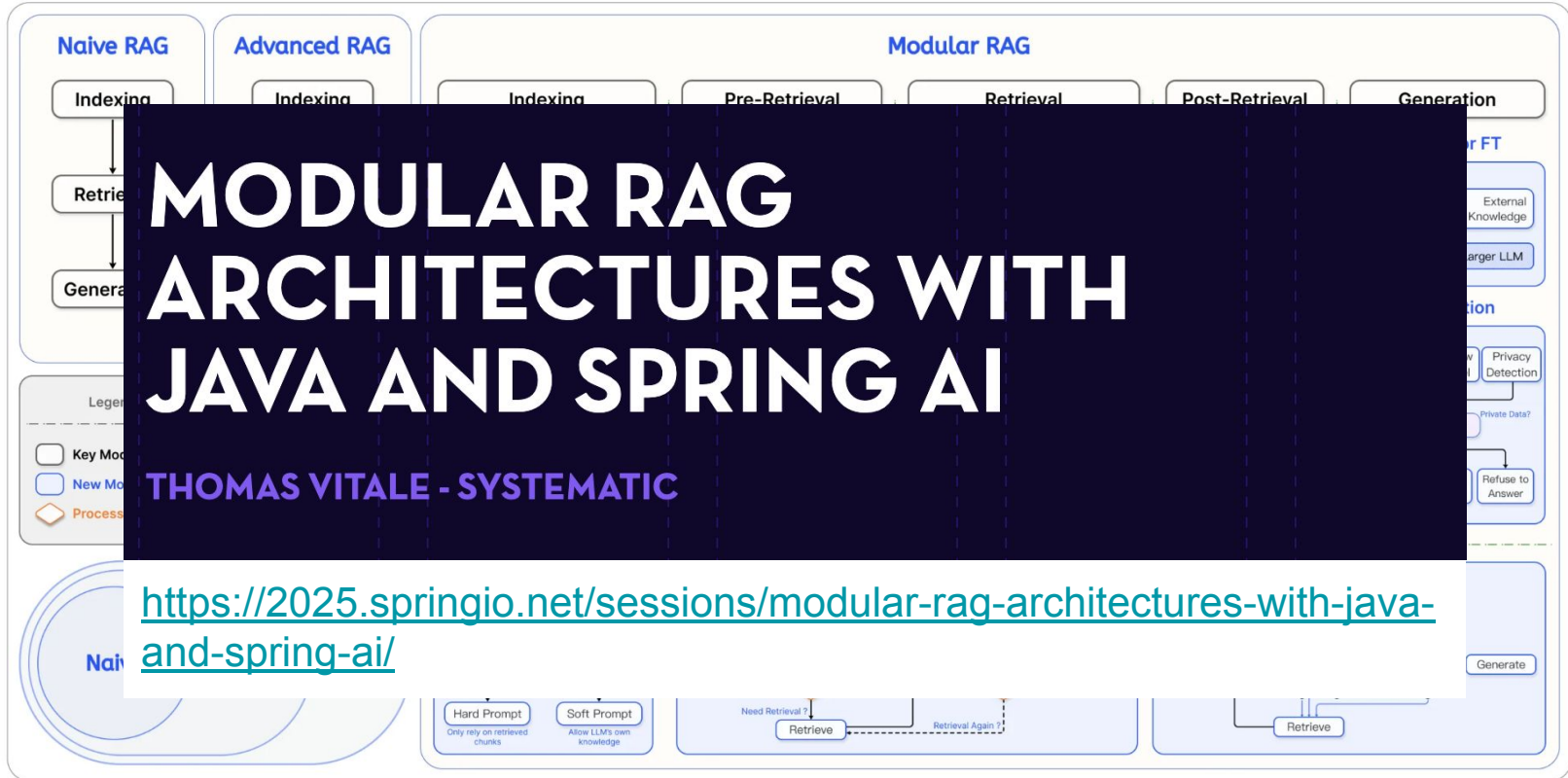
AI Models are only as good as the **Model Context** provided to them

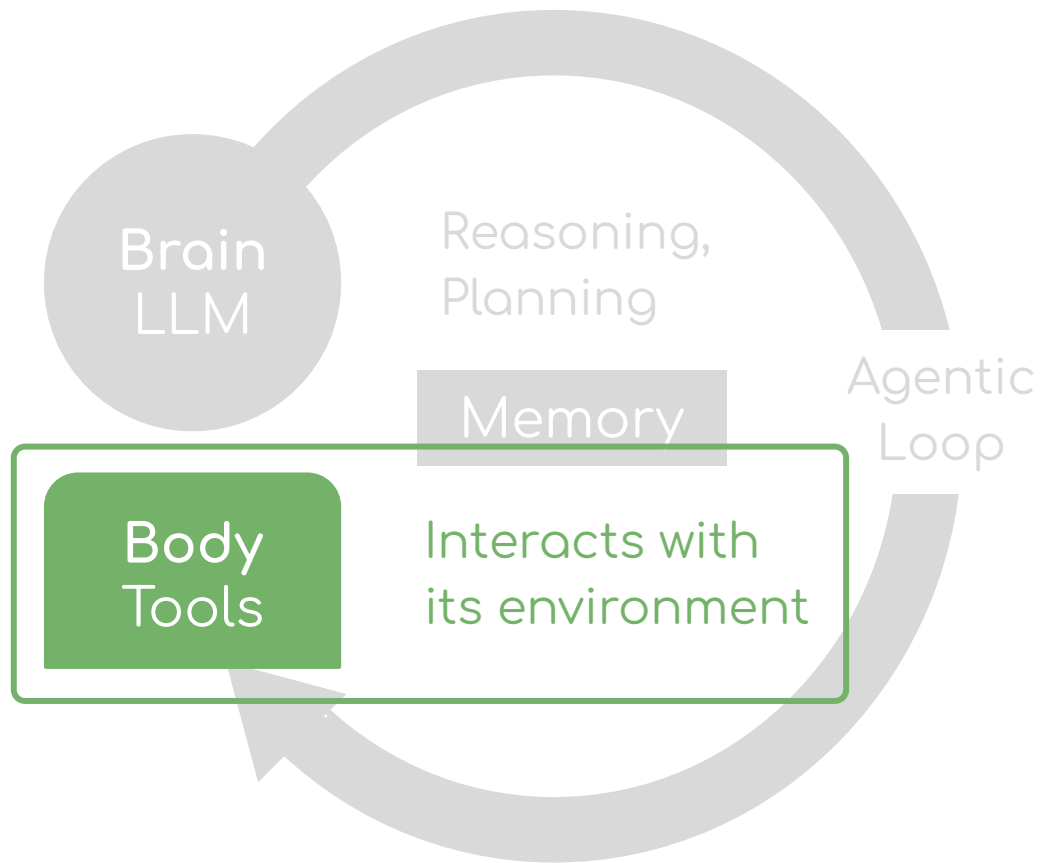


(LLM) Prompt Stuffing & RAG



(LLM) Prompt Stuffing & Modular RAG



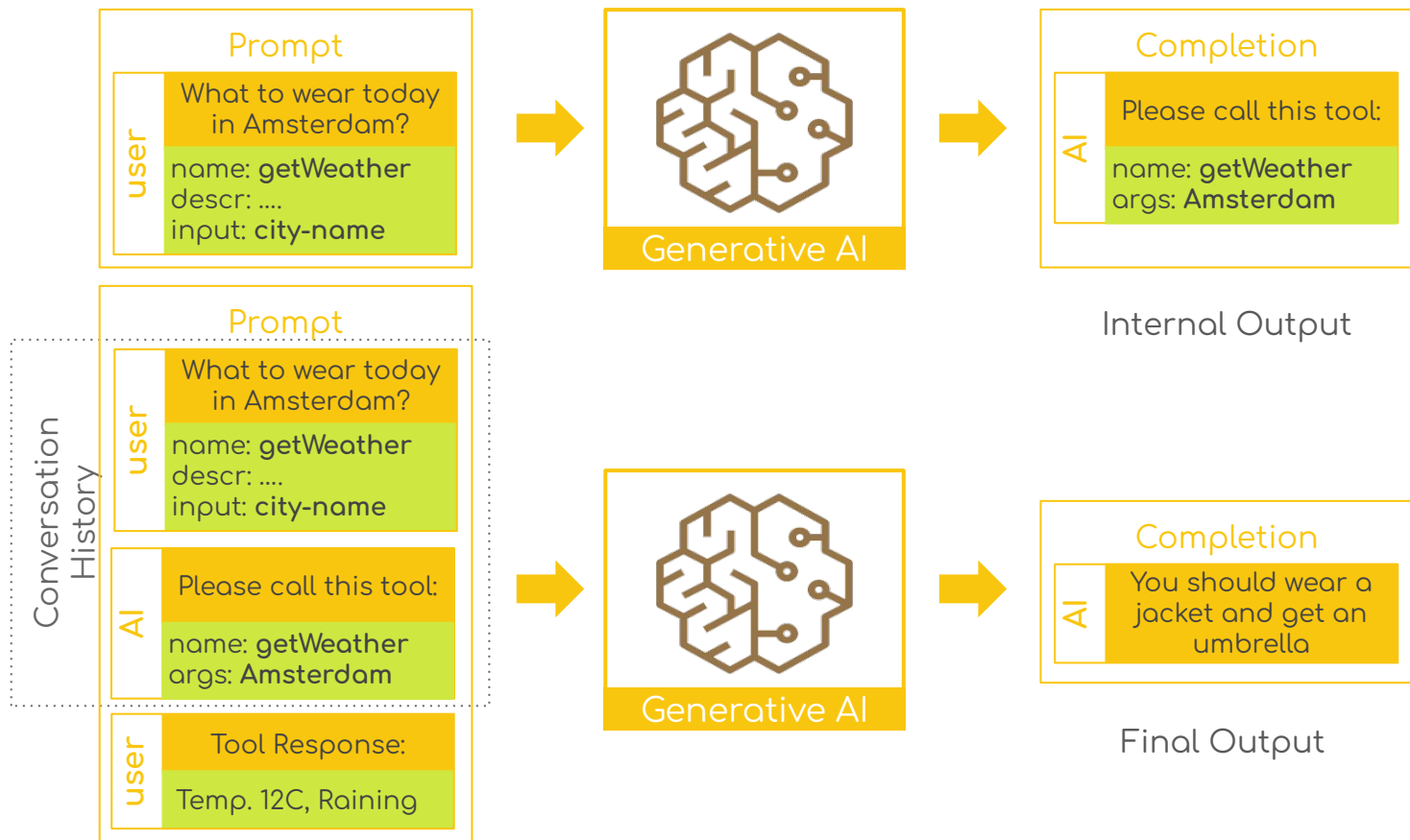


Prompt ~ Model Context

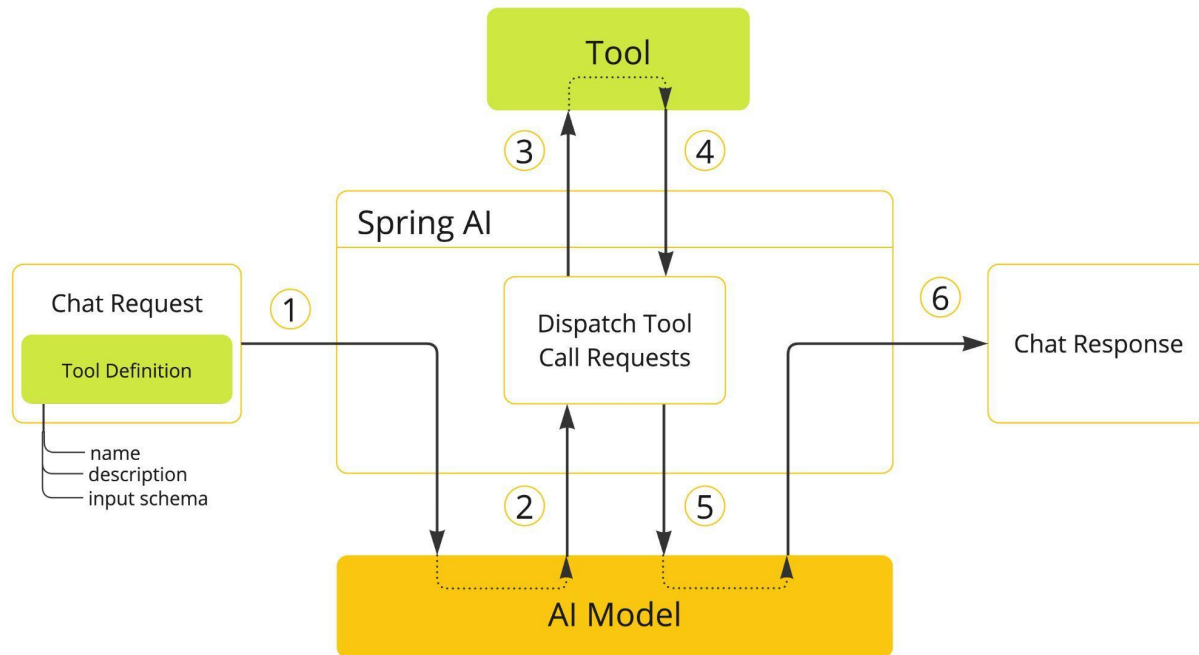
AI Models are only as good as the **Model Context** provided to them



Tool Calling



Tool Calling - Spring AI

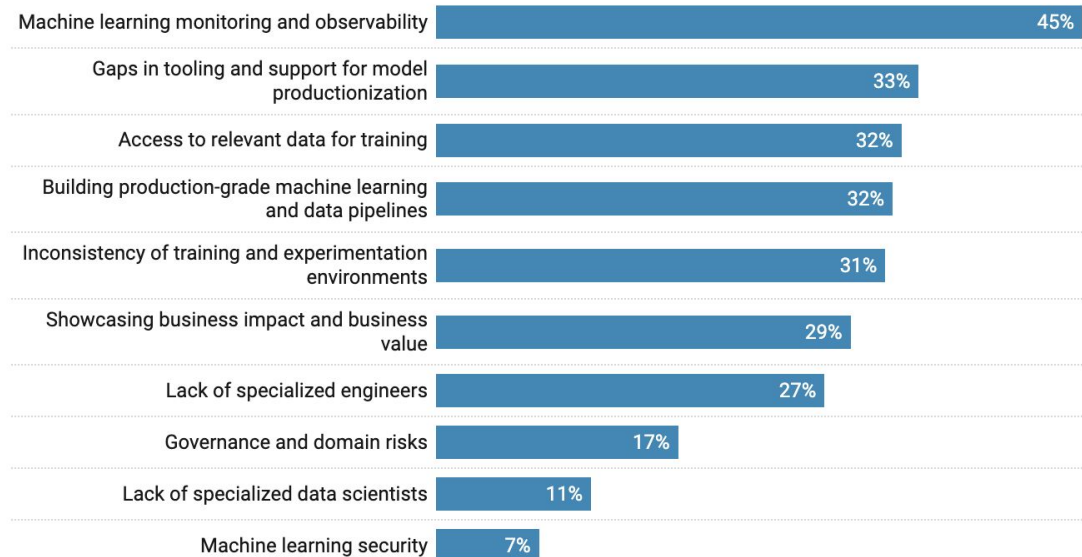


ML and LLM Observability



Observability and Monitoring Is the Biggest Challenge to Moving ML Models Into Production

Select the top 3 biggest challenges that you face when productionizing your machine learning models.



Spring AI Observability



Micrometer

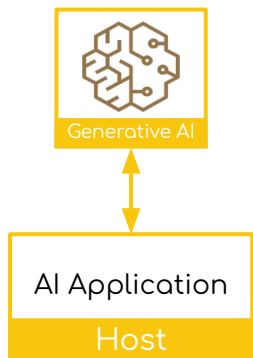


Wouldn't it be nice ...

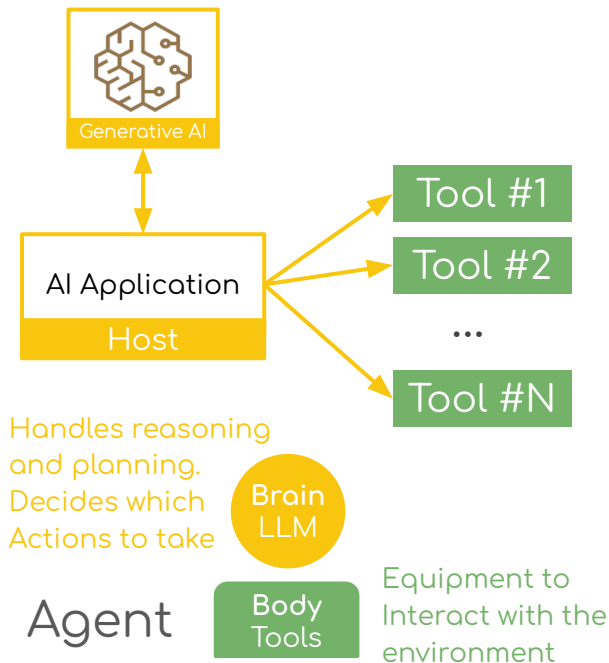
- To be able to use Tools written in other languages?
- To have Spring AI Tools used in non-Java envs?

Model Context Protocol (MCP)

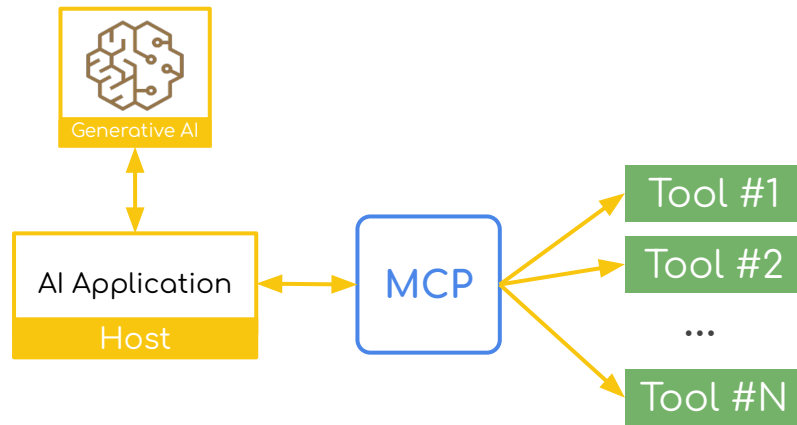
1 LLM



2 LLM + Tool Calling



3 LLM + MCP



Unified way to connect AI
applications to different data
sources, tools, ...

MCP Client & Server Ecosystem

MCP Hub

Model Context Protocol

A protocol for extending AI capabilities through local servers

- Model Context Protocol
- Universal Standard
- Semantic Understanding
- Context Aware
- Secure Connection

313 Live

Search by name... 

Personal MCP Server

AI-assisted personal health tracking server

2 months ago

YouTube Watch Later MCP Server

Access YouTube Watch Later playlist via MCP

2 months ago

MCP Knowledge Graph

A server enabling persistent memory for Claude.

2 months ago

Dify MCP Server (TypeScript)

A TypeScript MCP server for Dify workflows.

2 months ago

COLUMBIA-MCP-SERVERS

Deployment infrastructure for Columbia's MCP servers

2 months ago

MCP Tool Builder

A server for dynamically creating tools

2 months ago

MCP Official Java SDK

The image shows a composite of three overlapping screenshots related to the MCP Official Java SDK.

Top Left Screenshot: A web browser view of the `modelcontextprotocol.io/sdk/java/mcp-overview` page. The left sidebar lists SDKs for Python, TypeScript, Java, Kotlin, C#, and C. The main content area is titled "Features" and lists:

- MCP Client and MCP Server implementations supporting:
 - Protocol
 - Tool disc
 - Resource
 - Roots lis

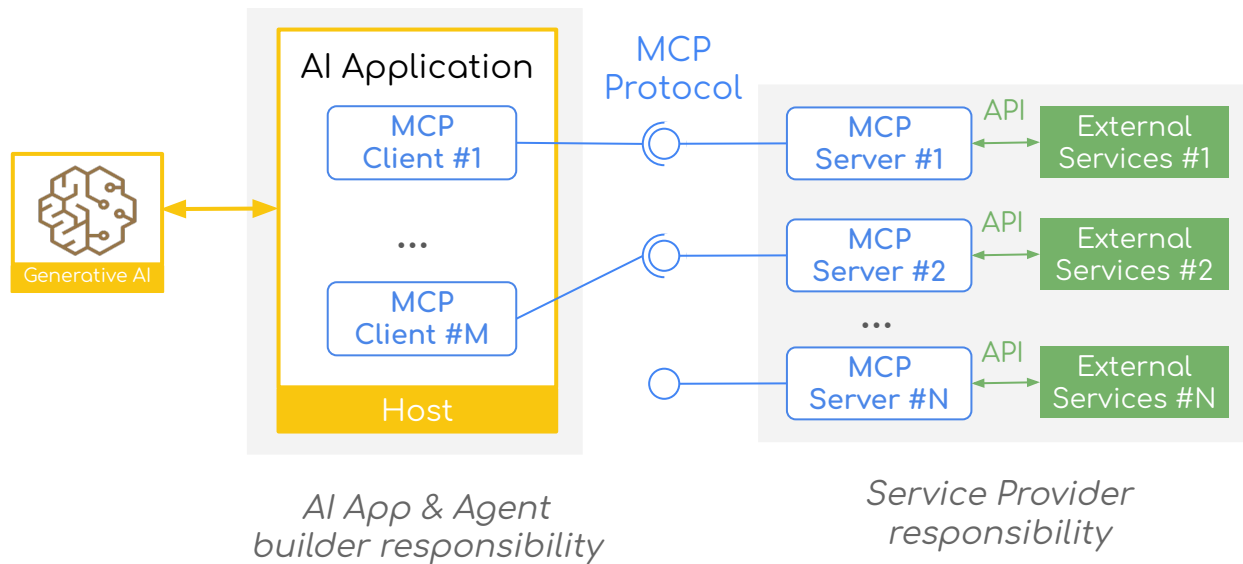
Top Right Screenshot: A GitHub repository view for `modelcontextprotocol / java-sdk`. It shows the repository name, public status, and statistics: 22 Issues, 12 Pull requests, 19 Watchers, 144 Forks, and 750 Stars. The "About" section states: "The official Java SDK for Model Context Protocol servers and clients. Maintained in collaboration with Spring AI".

Bottom Screenshot: A file explorer view showing the directory structure of the SDK. It includes two circular profile pictures of the maintainers, **Christian Tzolov** and **Dariusz Jędrzejczyk**. The file list shows:

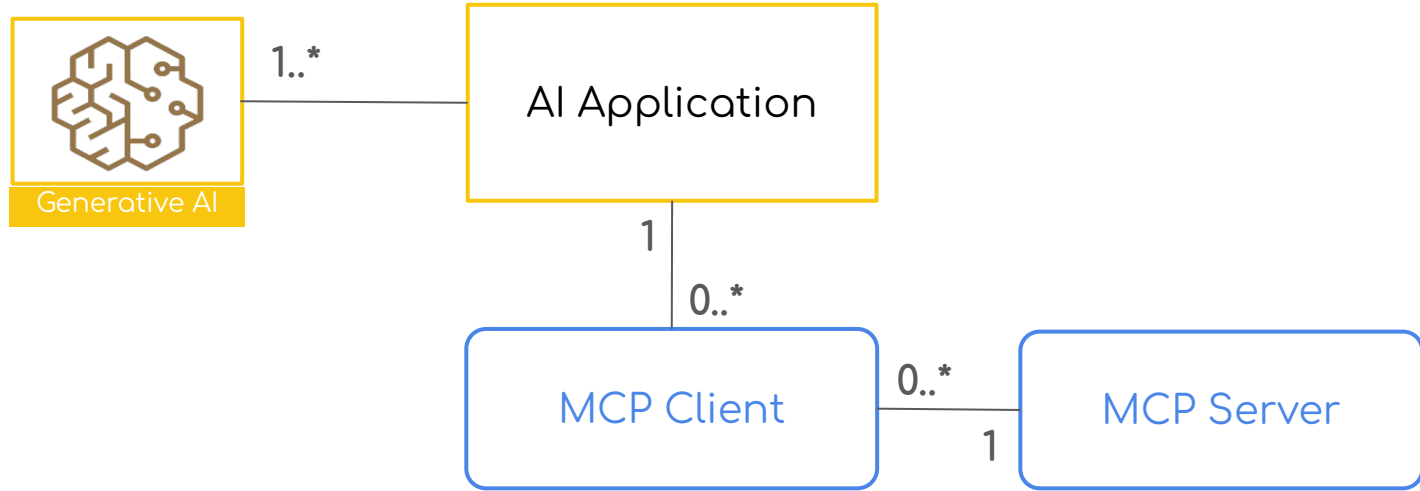
- `mcp-test`
- `mcp`

The bottom right of this screenshot shows a snippet of a commit message: "Failed to start process with..." followed by a green checkmark and the commit hash `79ec5b5` from 4 days ago.

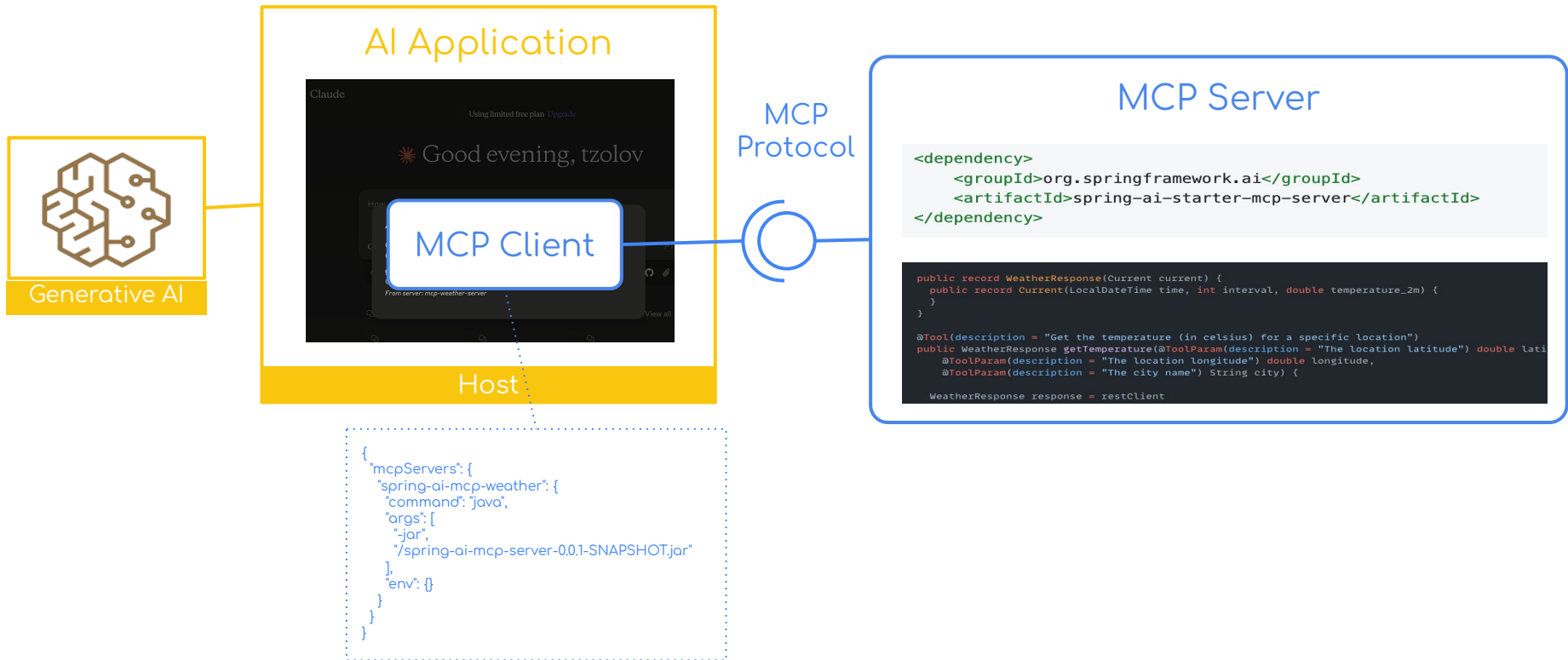
MCP Components



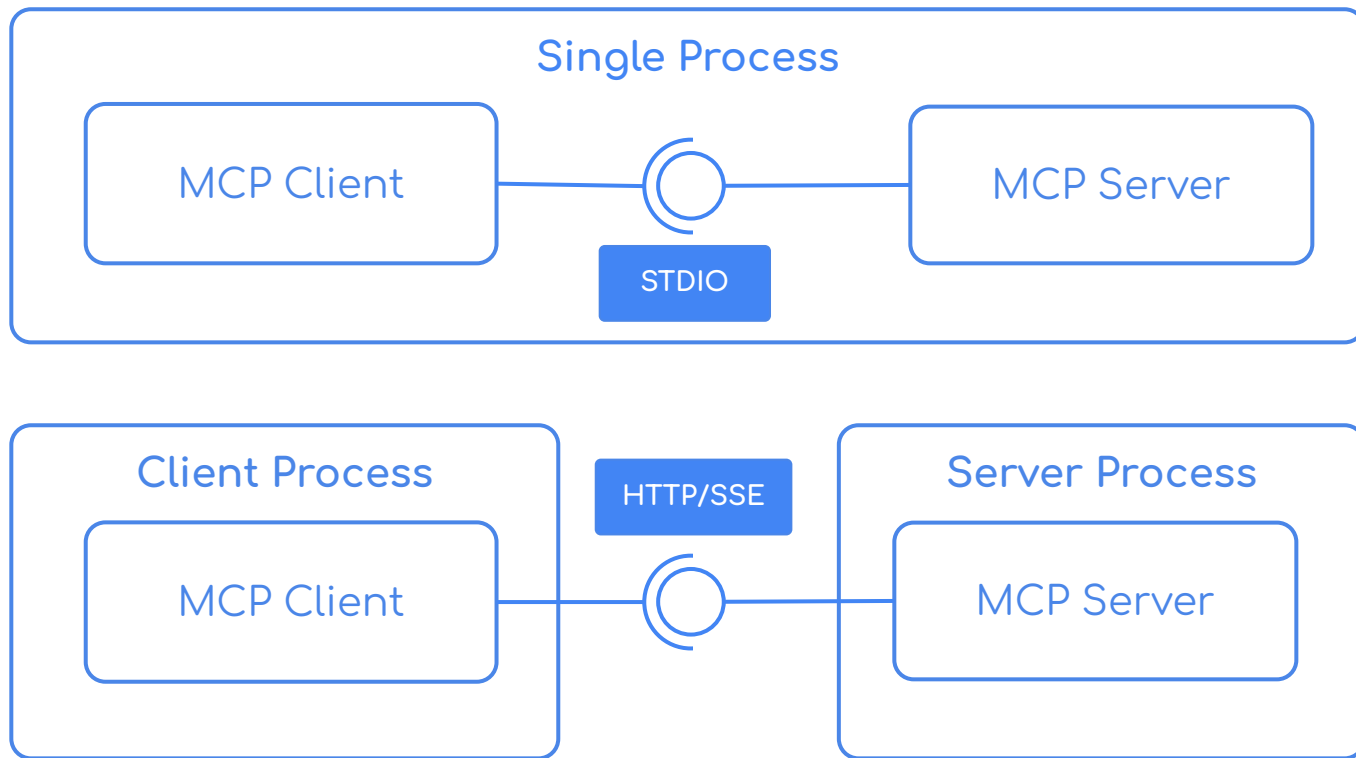
MCP Cardinalities



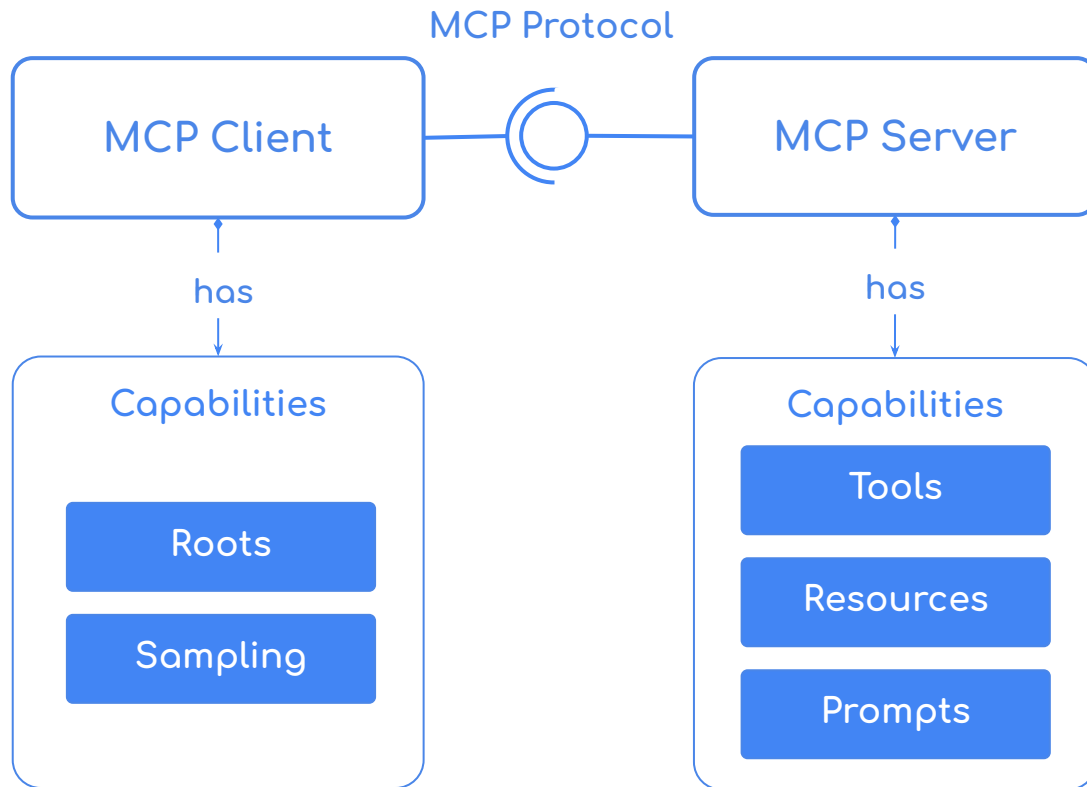
Spring AI MCP Server - Demo



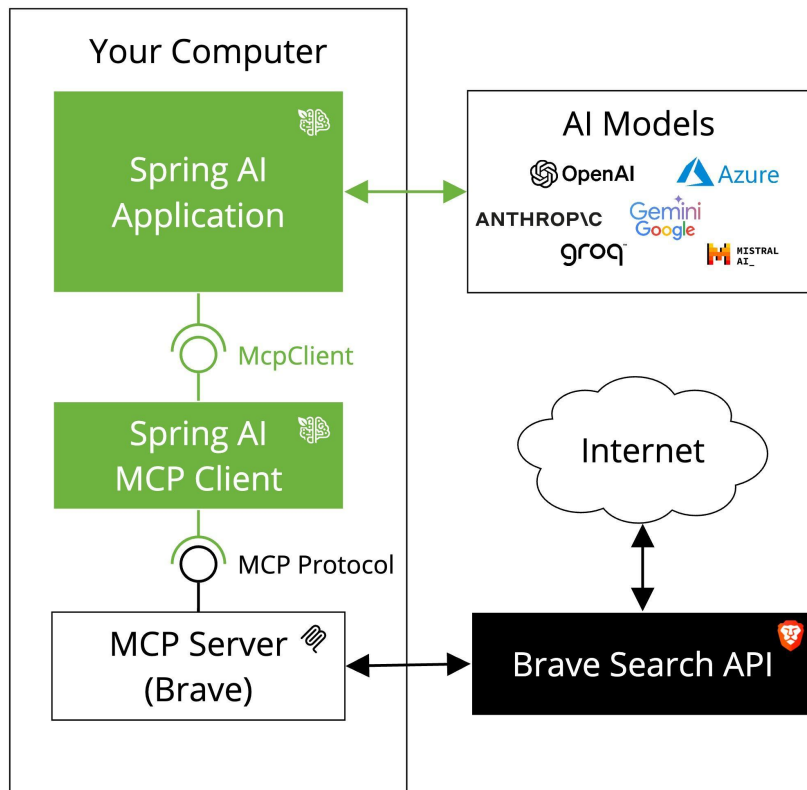
MCP Transports



MCP Capabilities



Spring AI + JavaScript WebSearch & Filesystem Tools



```
Application.java

@Bean
public CommandLineRunner chatbot(ChatClient.Builder chatClientBuilder, ToolCallbackProvider tools) {

    return args -> {
        var output = chatClientBuilder.build().prompt()
            .system("You are useful assistant and can perform web searches Brave's search API to reply")
            .user("Create a summary about the Talent Arena conference and save it as markdown talent-a")
            .tools(tools)
            .call()
            .content();
    };
}
```

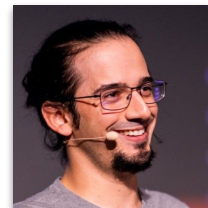
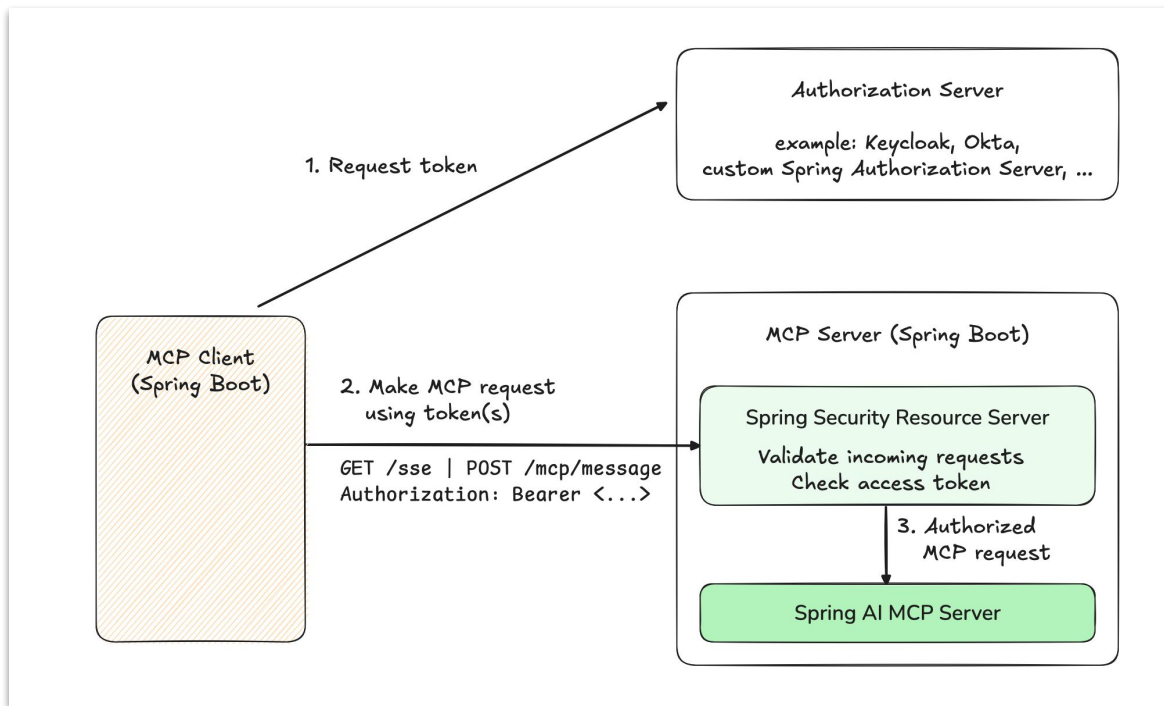
```
mcp-servers-config.json

{
  "mcpServers": {
    "brave-search": {
      "command": "npx",
      "args": [
        "-y",
        "@modelcontextprotocol/server-brave-search"
      ],
      "env": {
      }
    },
    "filesystem": {
      "command": "npx",
      "args": [
        "-y",
        "@modelcontextprotocol/server-filesystem",
        "/Users/christiantzolov/Desktop/tmp"
      ]
    }
  }
}
```

MCP Authorization

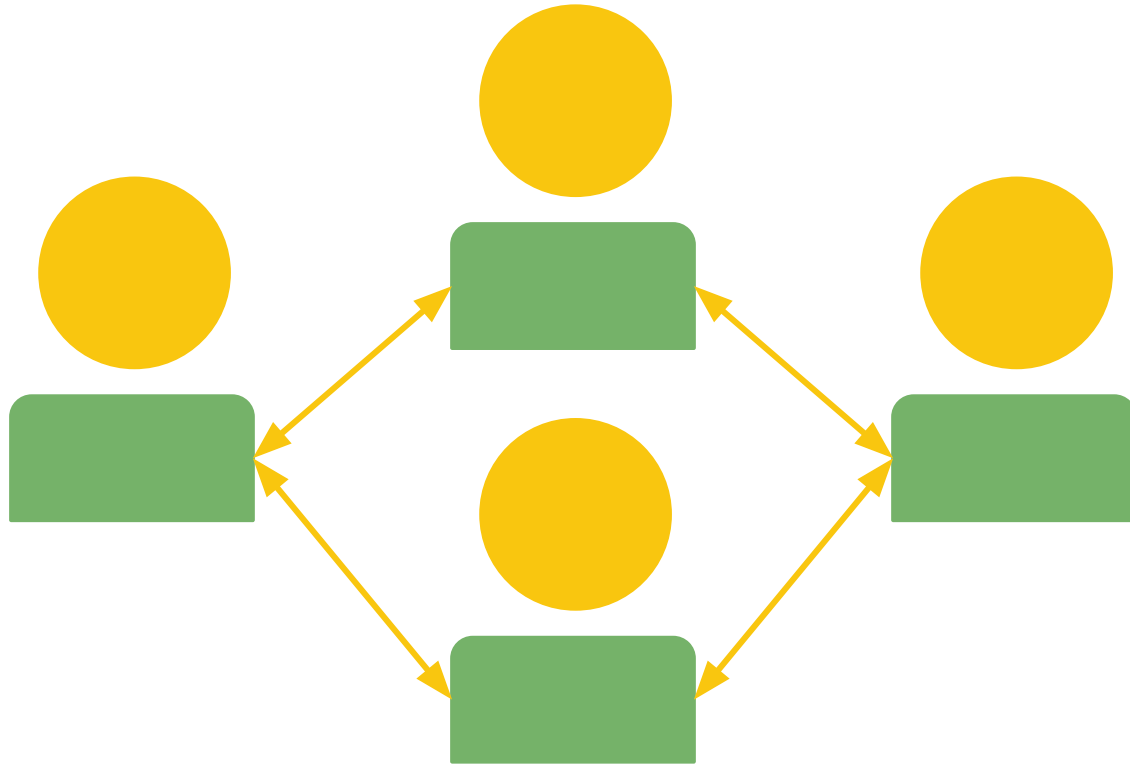
- New Protocol (2025-03-26) Authorization Feature
 - Authorization capabilities at the transport level
 - OAuth 2.1 compliant
 - Enabling MCP clients to make requests to restricted MCP servers on behalf of resource owners
 - For HTTP-based transports

Spring AI MCP & Spring Security OAuth



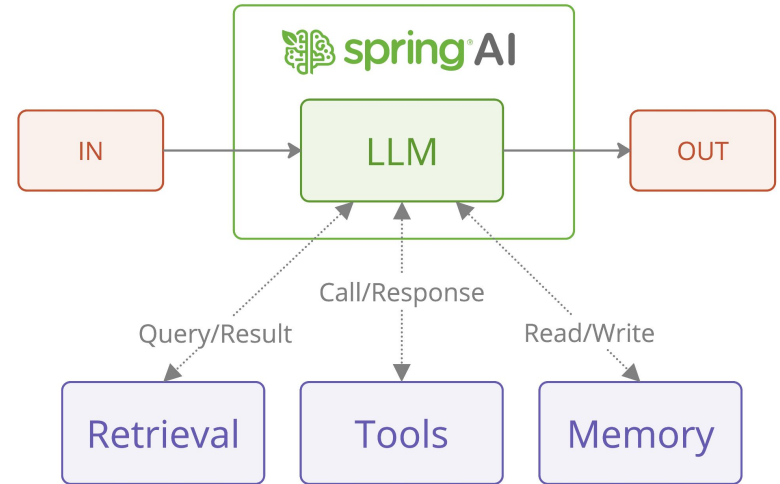
Blog: <https://spring.io/blog/2025/05/19/spring-ai-mcp-client-oauth2>

Agentic Systems



Agentic Systems - Prescriptive vs Autonomous

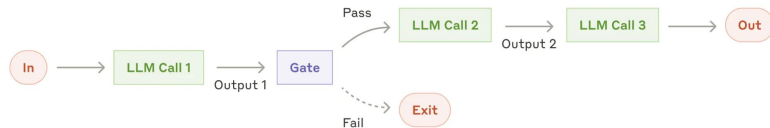
- Workflows - LLMs and Tools are orchestrated through predefined paths, e.g. prescriptive
- Autonomous Agents - LLMs autonomously plans and executes the processing steps toward accomplishing the tasks - e.g. autonomous



Agentic Workflows

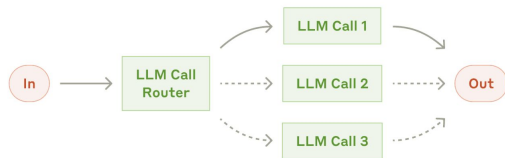
Chain Workflow

Break complex tasks down into simpler, more manageable steps



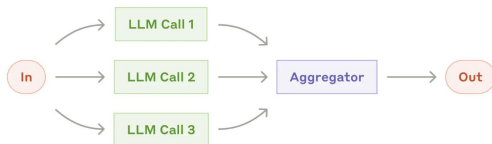
Routing Workflow

Complex tasks with different input types, handled by specialized processes. An LLM analyzes the input content and routes it to the specialized handler.



Parallelization Workflow

Work simultaneously on tasks and aggregate outputs



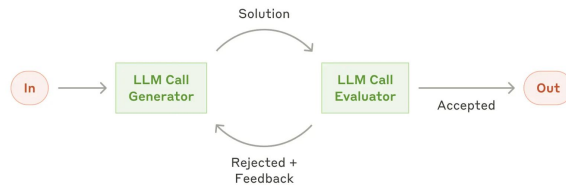
Orchestration - Worker Workflow

Central LLM orchestrates task decomposition. Specialized workers handle specific subtasks



Evaluator - Optimizer Workflow

Dual-LLM process - one LLM generates responses while another provides evaluation and feedback in an iterative loop



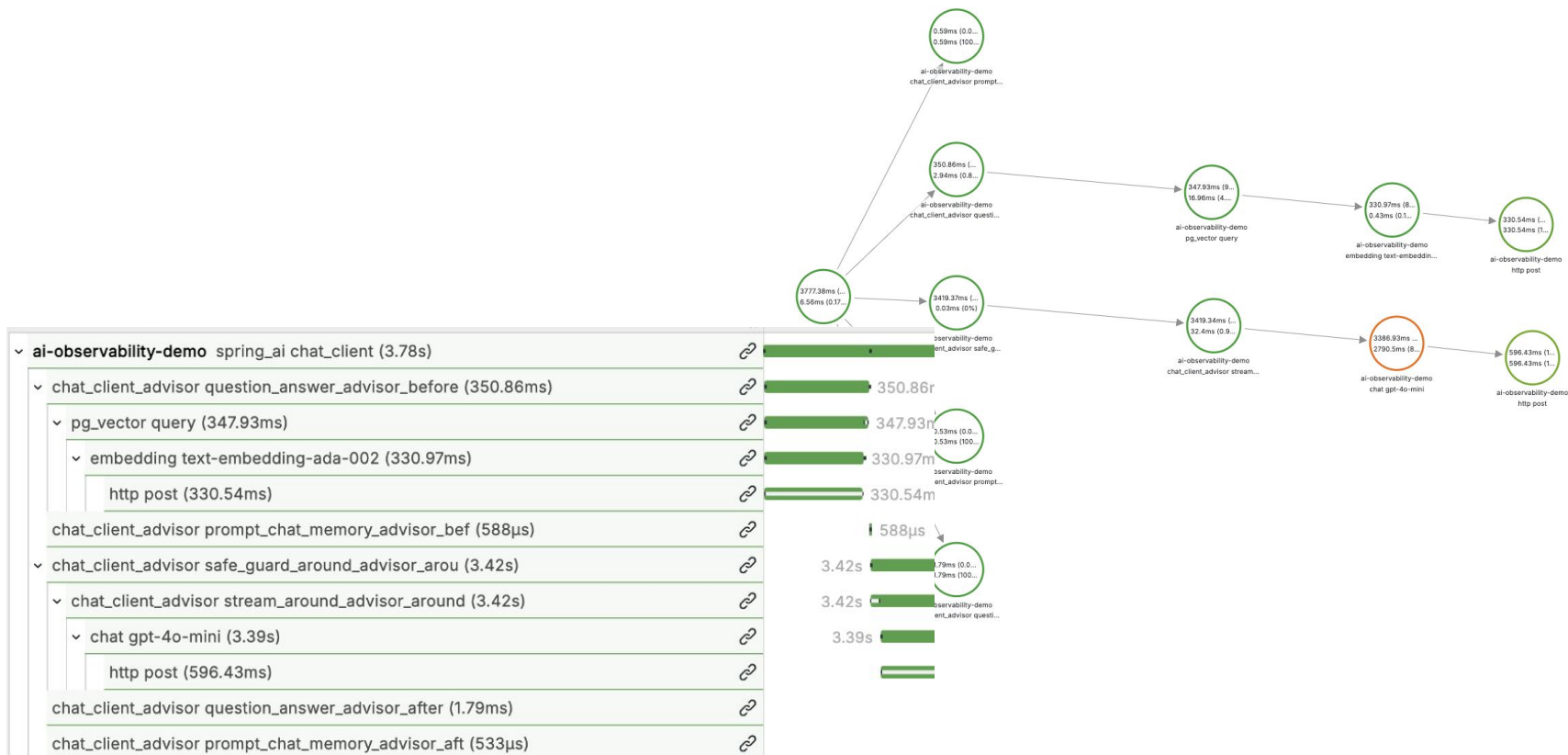
Building effective agents: <https://www.anthropic.com/engineering/building-effective-agents>

Building Effective Agents with Spring AI (Part 1): <https://spring.io/blog/2025/01/21/spring-ai-agentic-patterns>

GitHub Repo: <https://github.com/spring-projects/spring-ai-examples/tree/main/agentic-patterns>



Spring AI Observability



Agentic Workflows



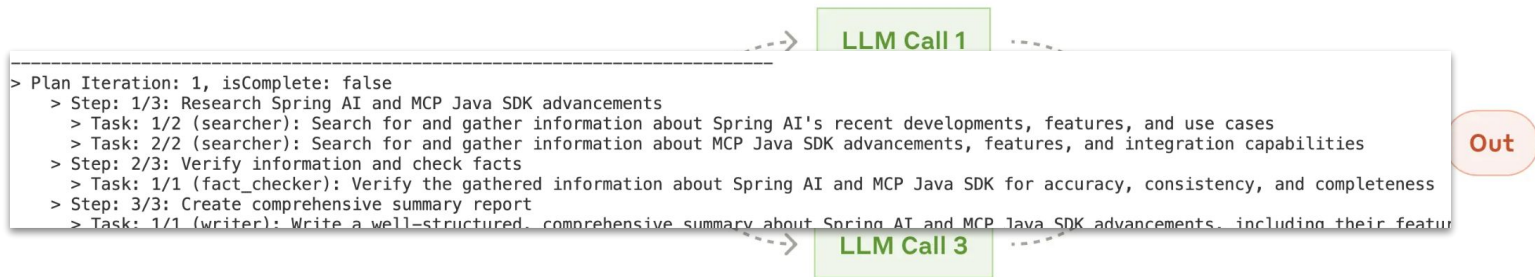
Build effective agents with Model Context Protocol using simple, composable patterns.

[Examples](#) | [Building Effective Agents](#) | [MCP](#)

pypi v0.0.14 open issues 31 chat 42 online pypi | downloads 12k license Apache-2.0

Orchestration - Worker Workflow

Central LLM orchestrates task decomposition. Specialized workers handle specific subtasks



When to Use:

- Complex tasks - subtasks can't be predicted upfront
- Tasks requiring different approaches or perspectives
- For adaptive problem-solving



Thank You

Christian Tzolov
Spring AI & MCP Java SDK
@christzolov , @tzolov