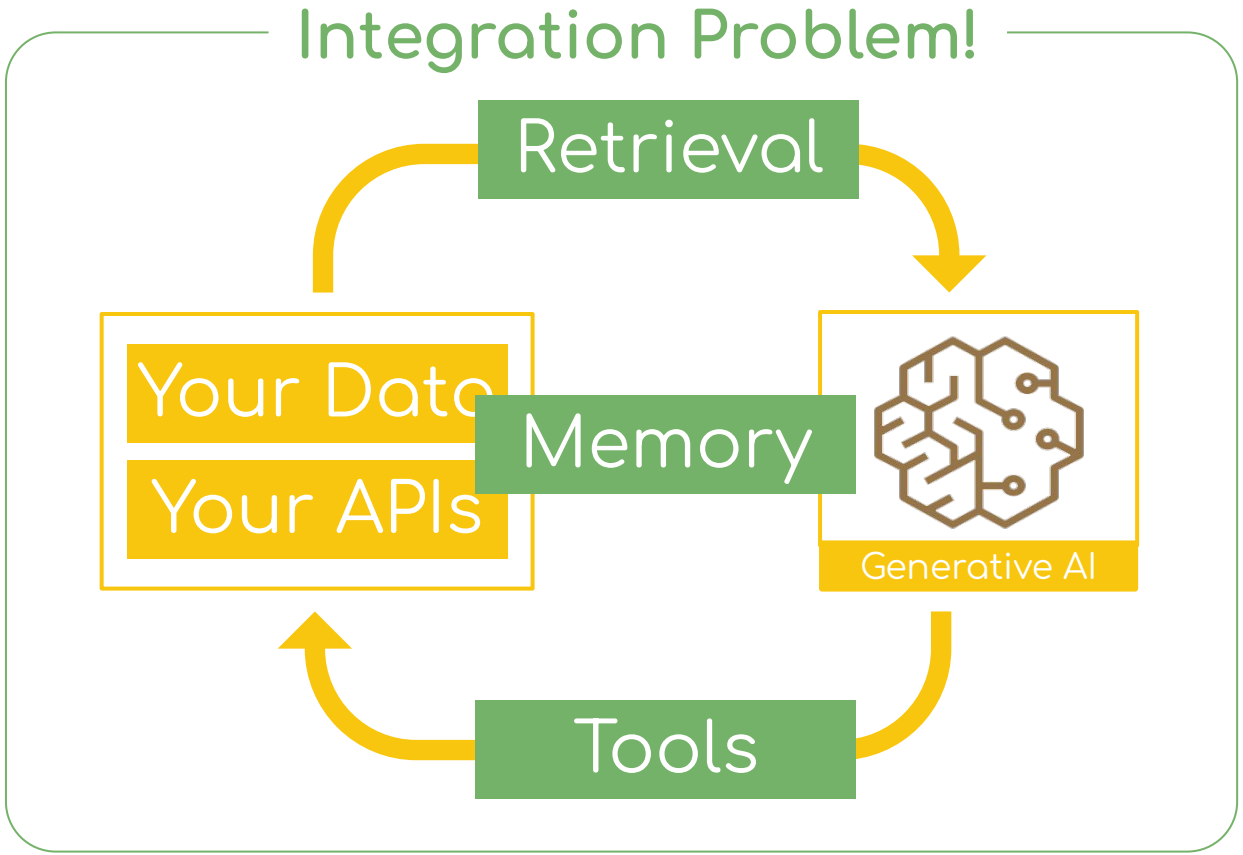
Building Agentic Systems with Spring AI & MCP

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

Using Generative AI



spring AI

Models

OpenAI
Azure
stability.ai
Hugging Face
ANTHROP\C
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
Weaviate
drant
Milvus
elasticsearch
mongoDB.
sandra
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

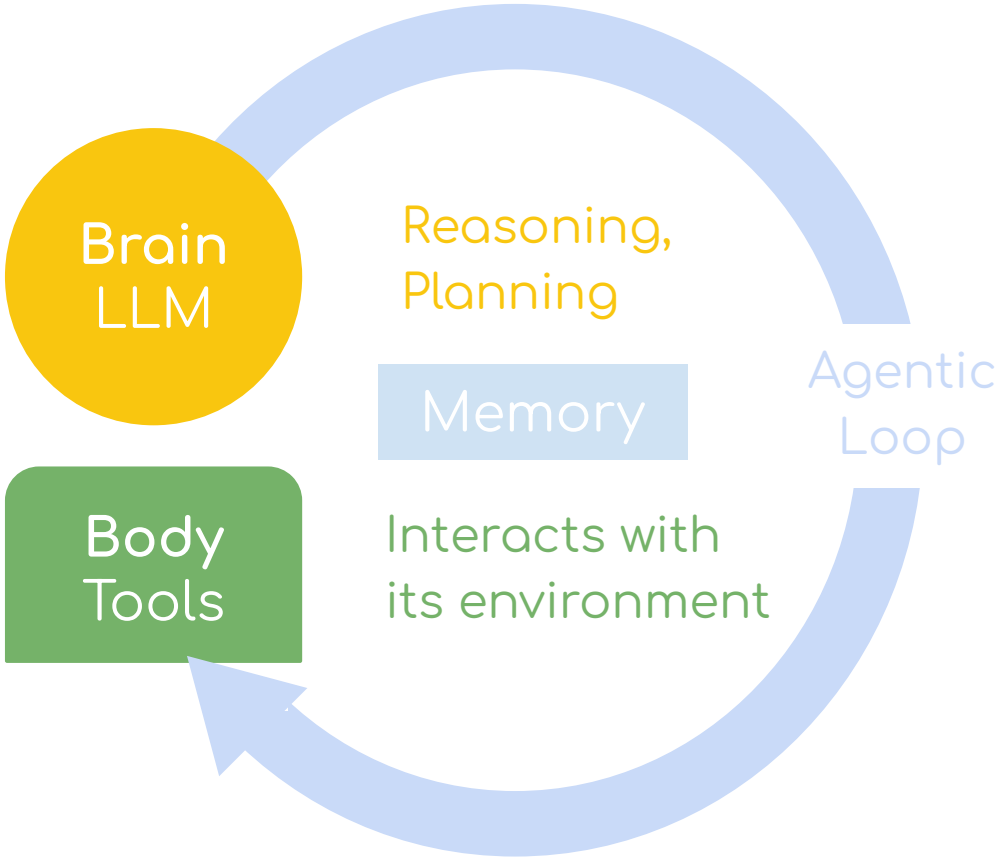
Decides which Actions to take

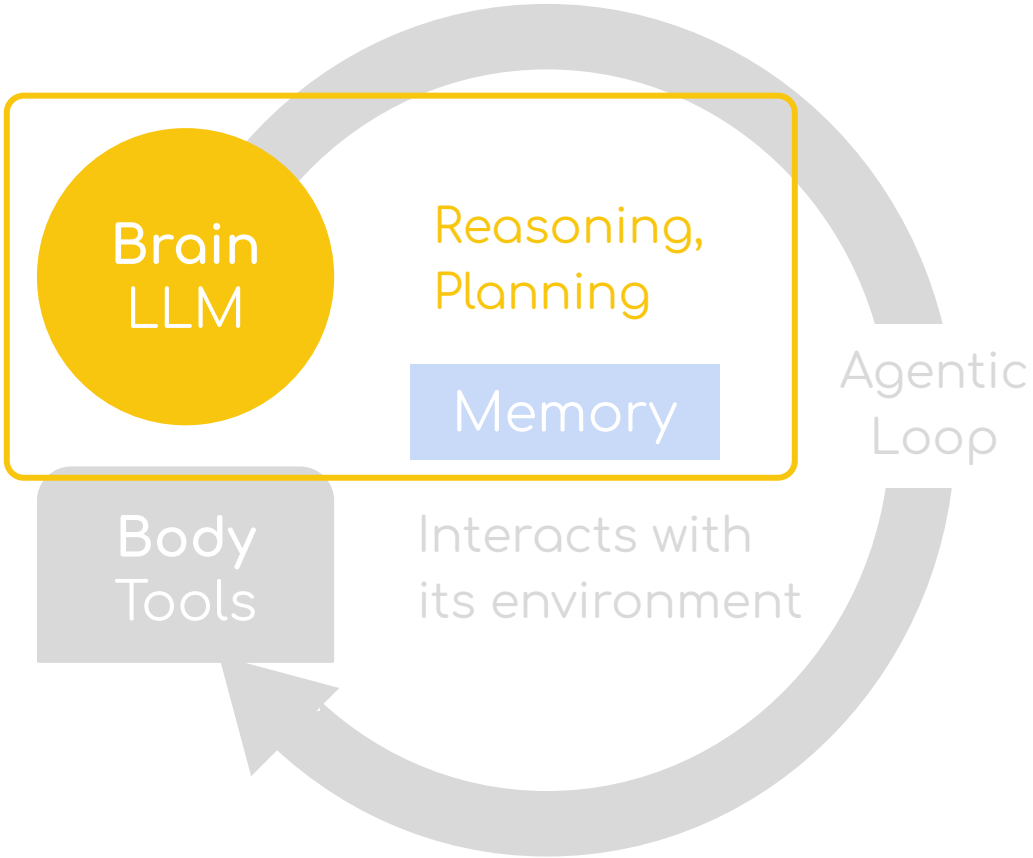
Body (Tools)

What Agent is equipped to do

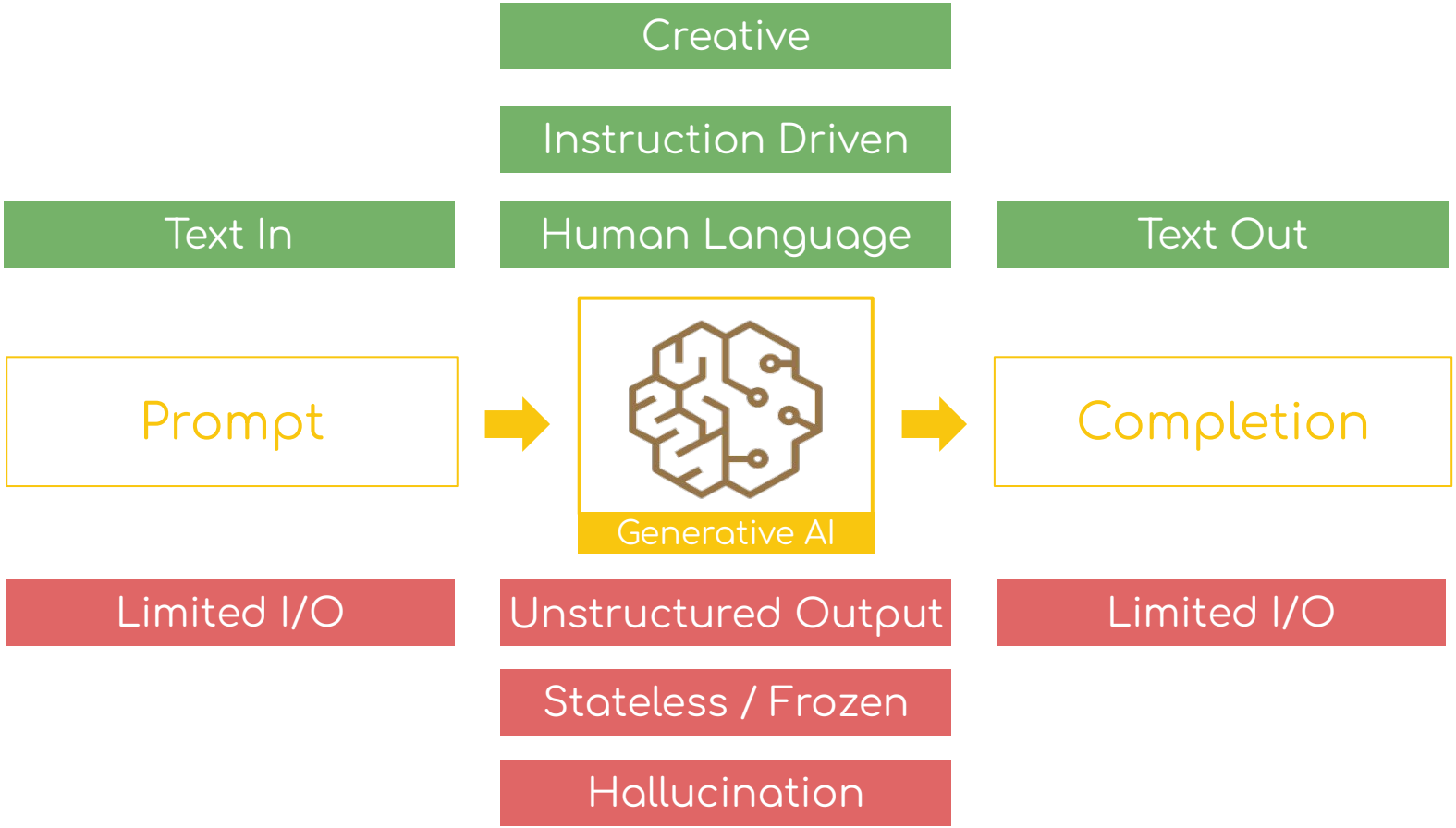
Interact with its environment

Agentic System



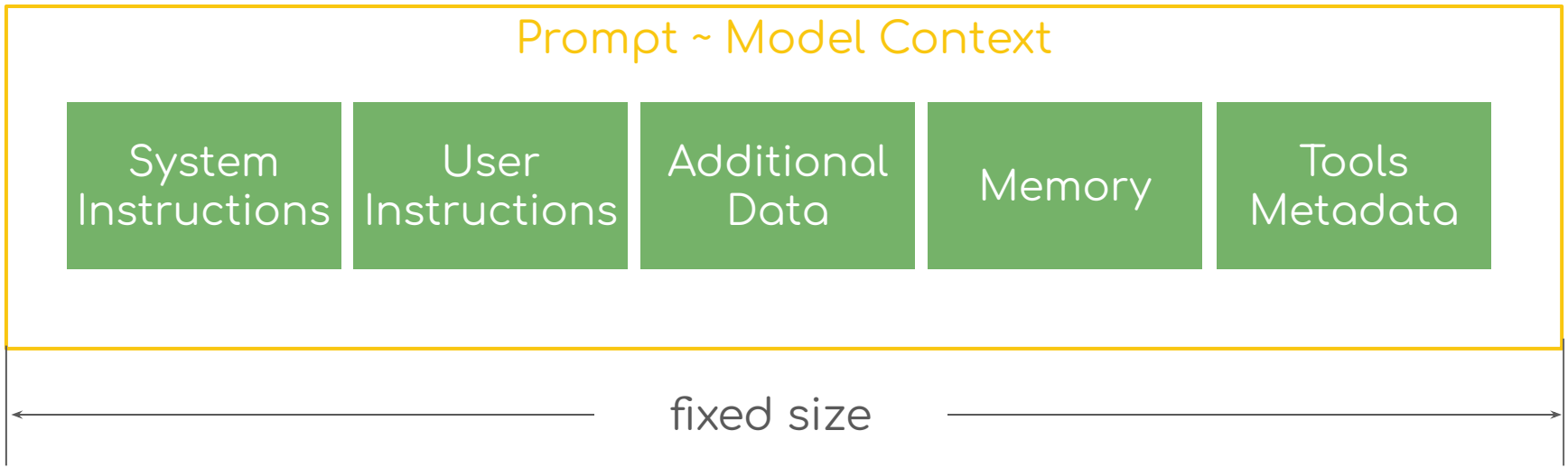

Brain
LLM
Reasoning,
Planning
Memory
Agentic
Loop
Body
Tools
Interacts with
its environment

LLM as a System



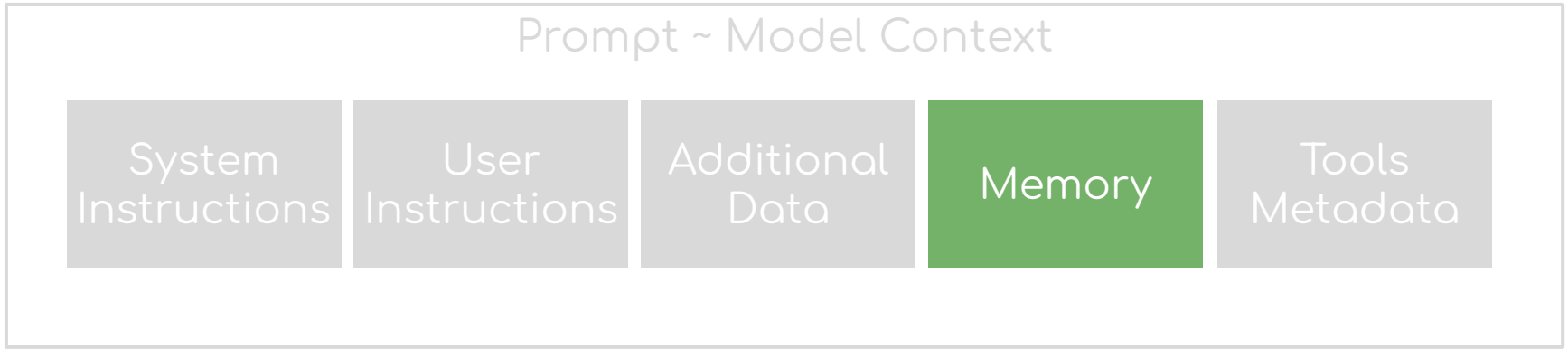
Prompt ~ Model Context

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

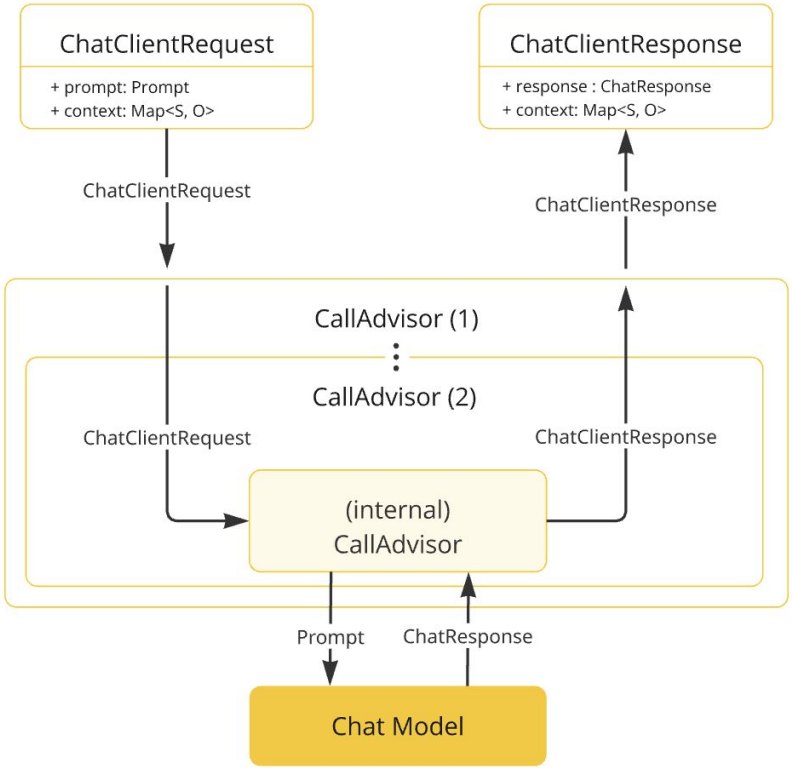


Prompt ~ Model Context

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

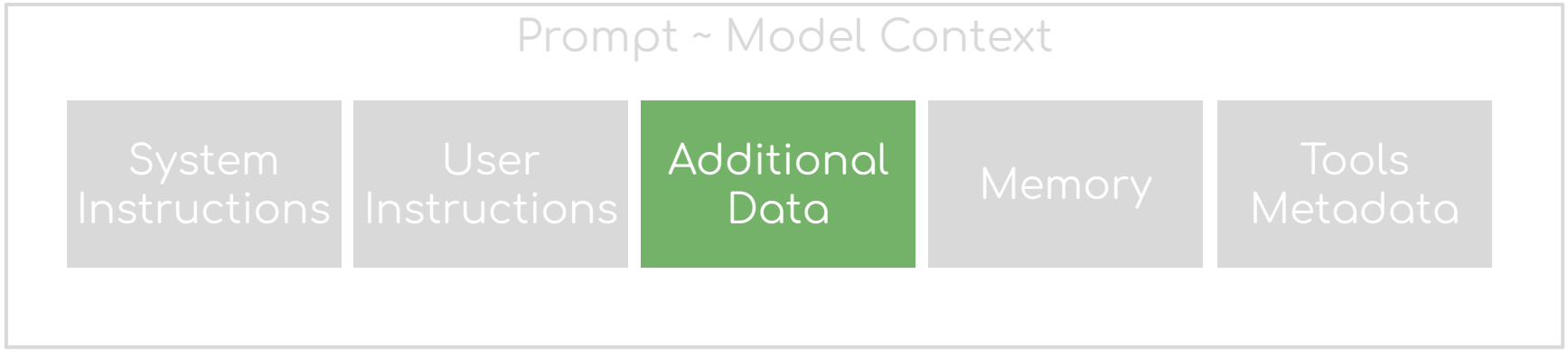


(LLM) Advisors

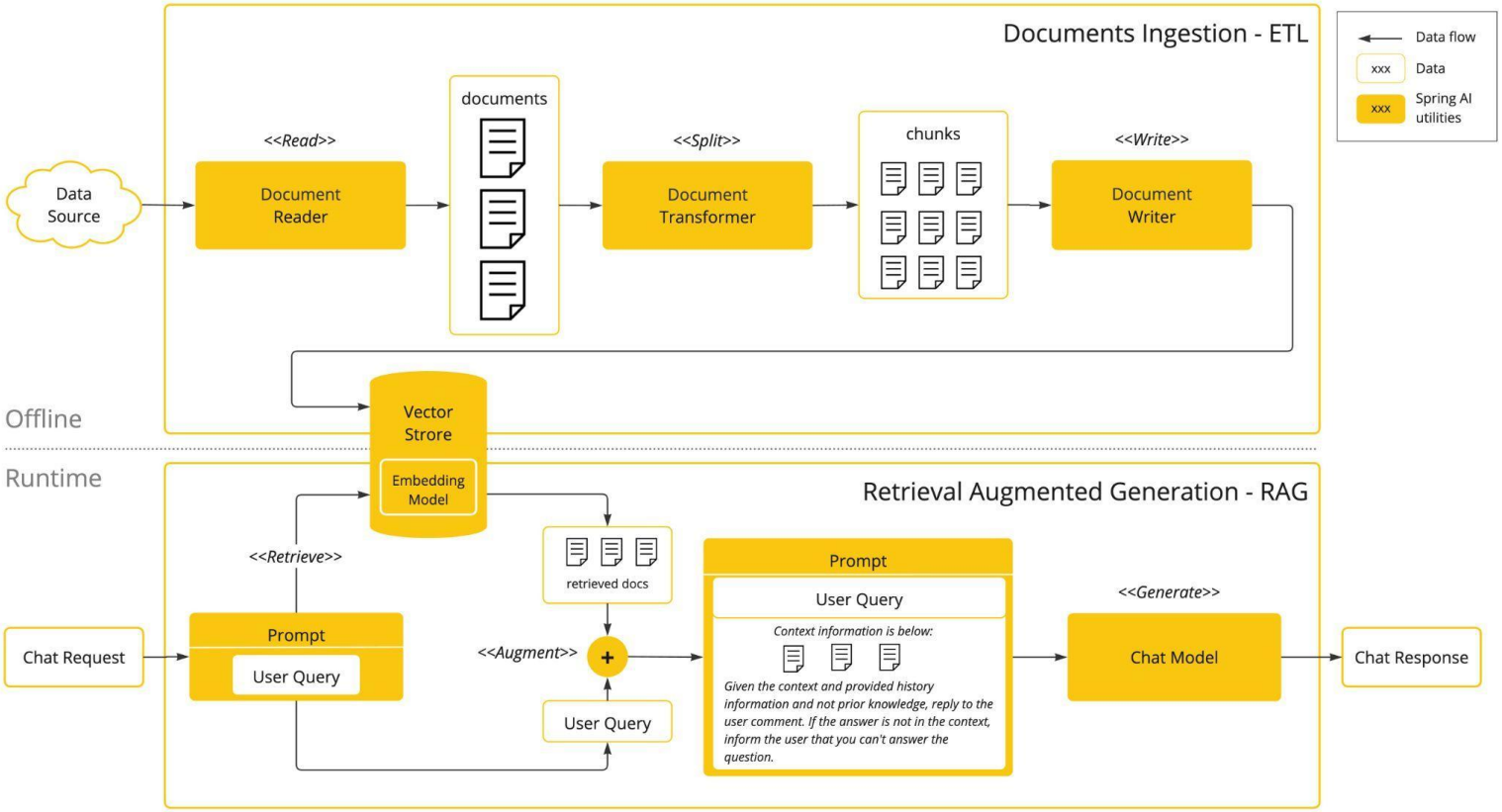


Prompt ~ Model Context

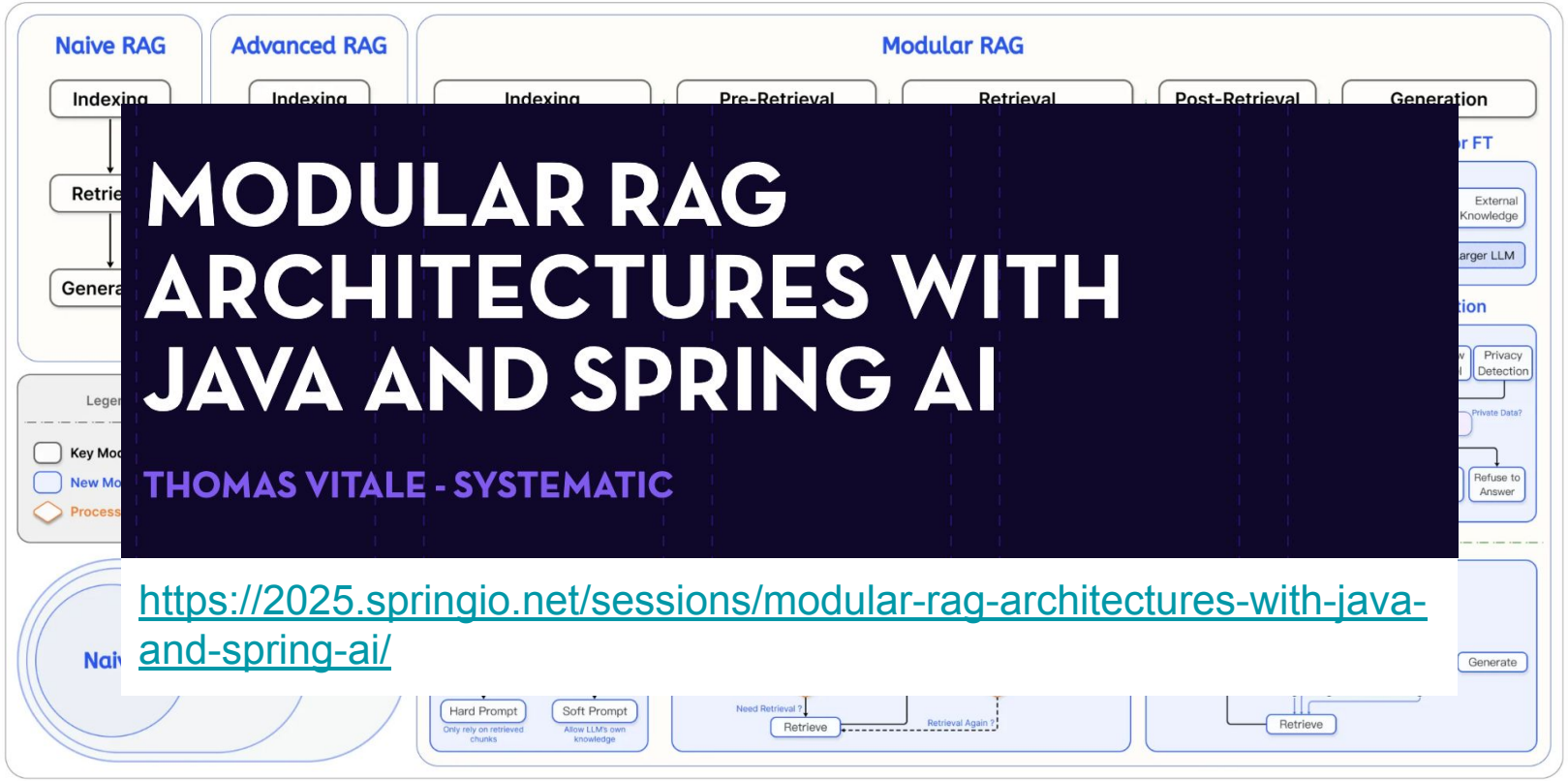
AI Models are only as good as the Model Context provided to them

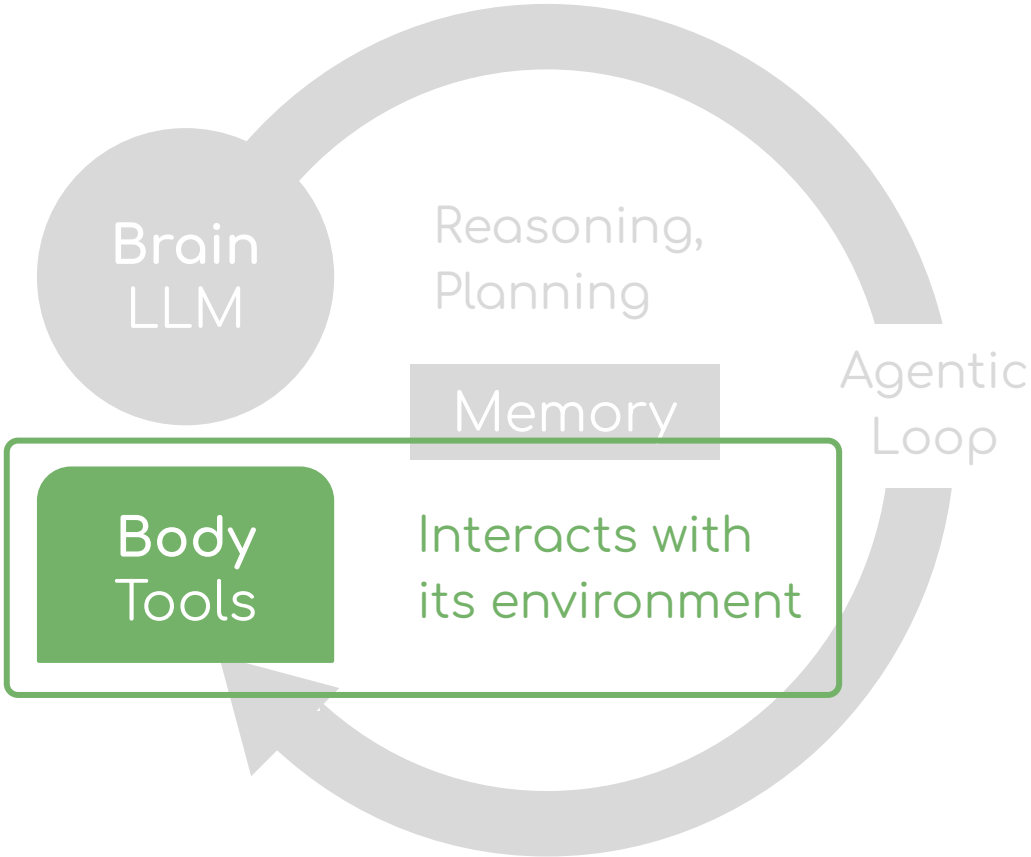


(LLM) Prompt Stuffing & RAG



(LLM) Prompt Stuffing & Modular RAG

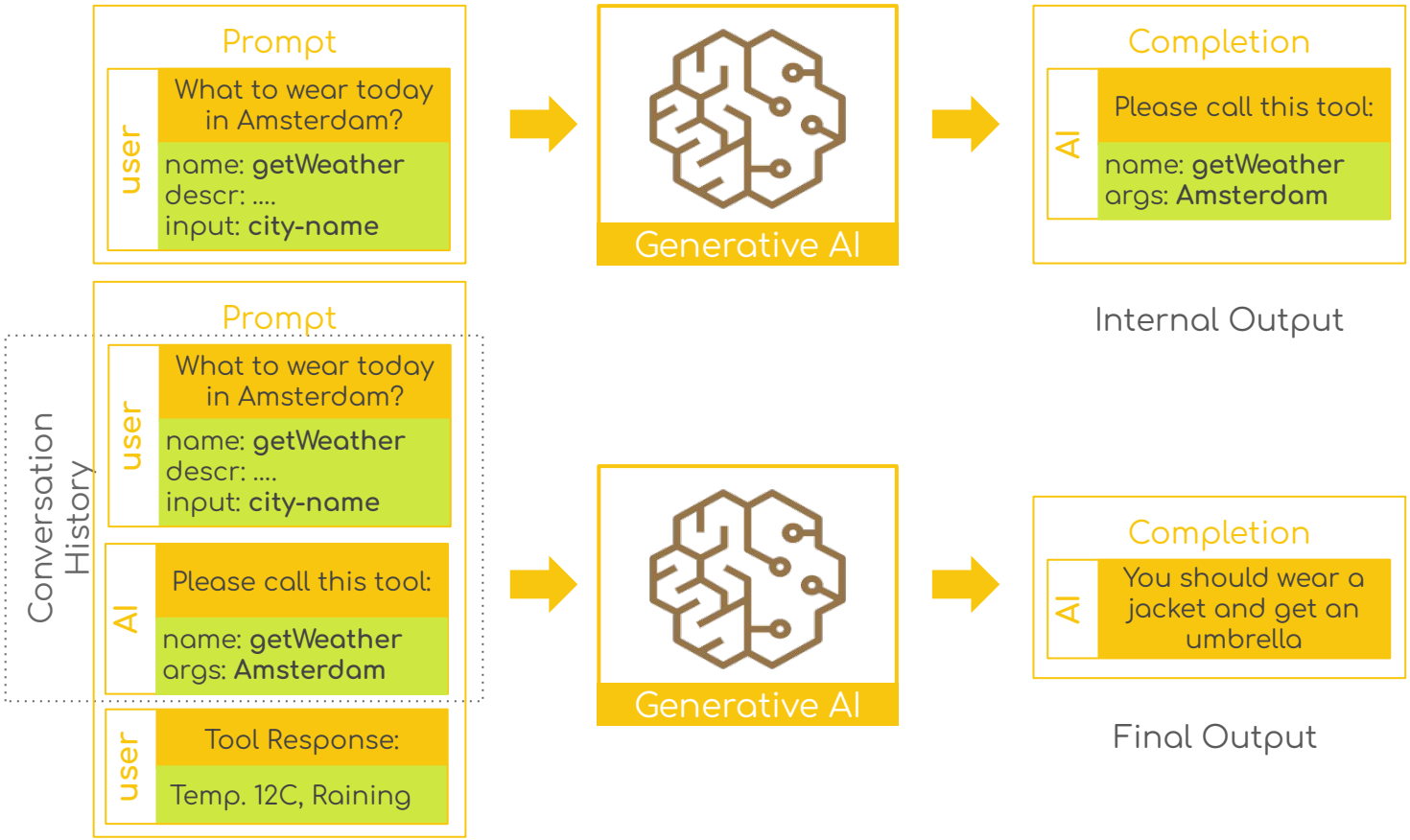



Brain
LLM
Reasoning,
Planning
Memory
Agentic
Loop
Body
Tools
Interacts with
its environment

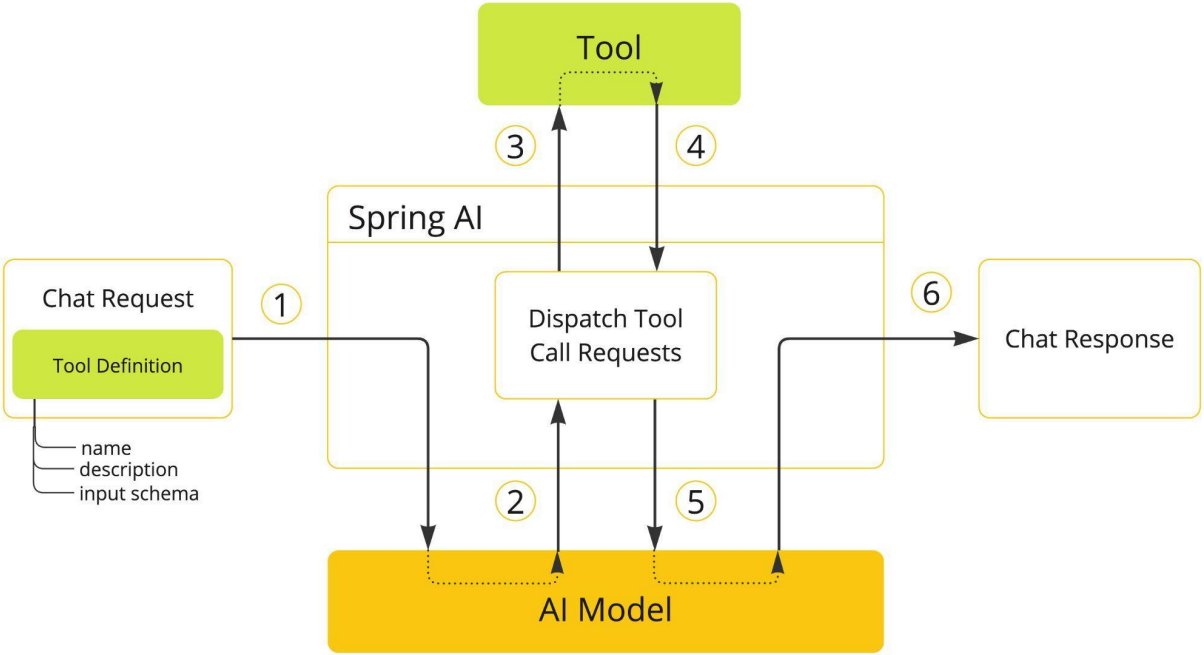
Prompt ~ Model Context

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

Tool Calling

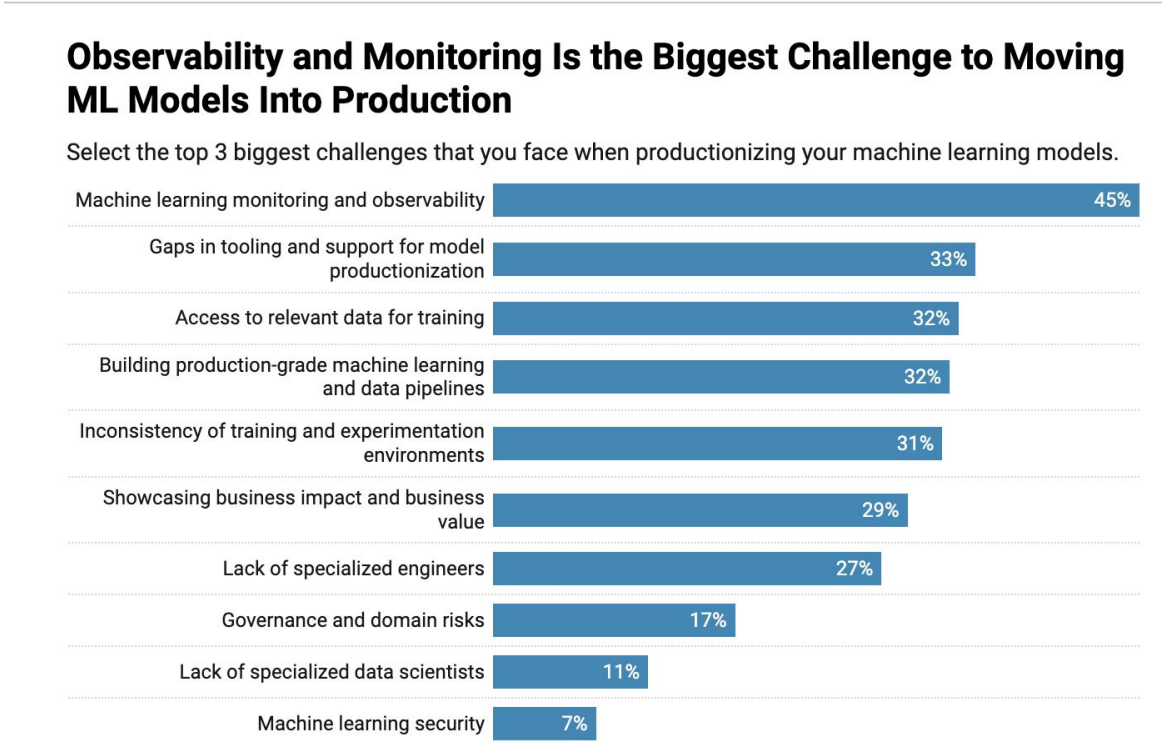


Tool Calling - Spring AI



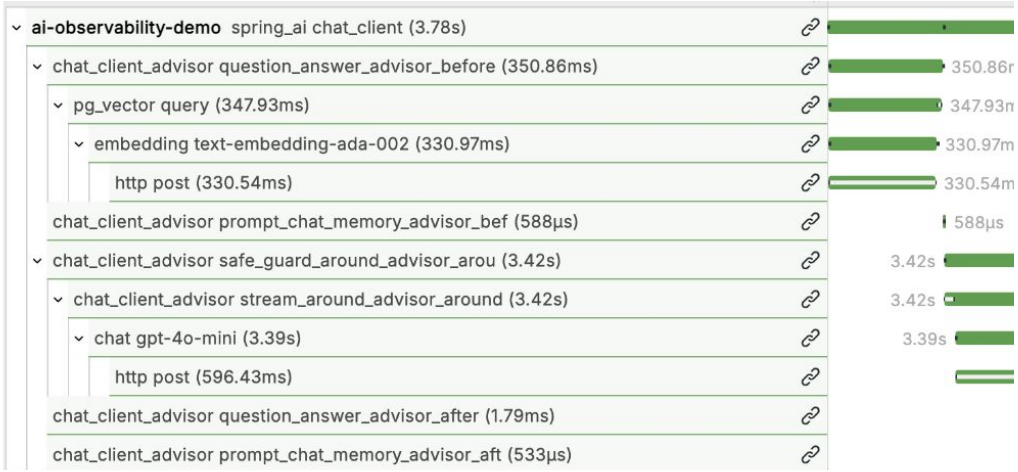
ML and LLM Observability

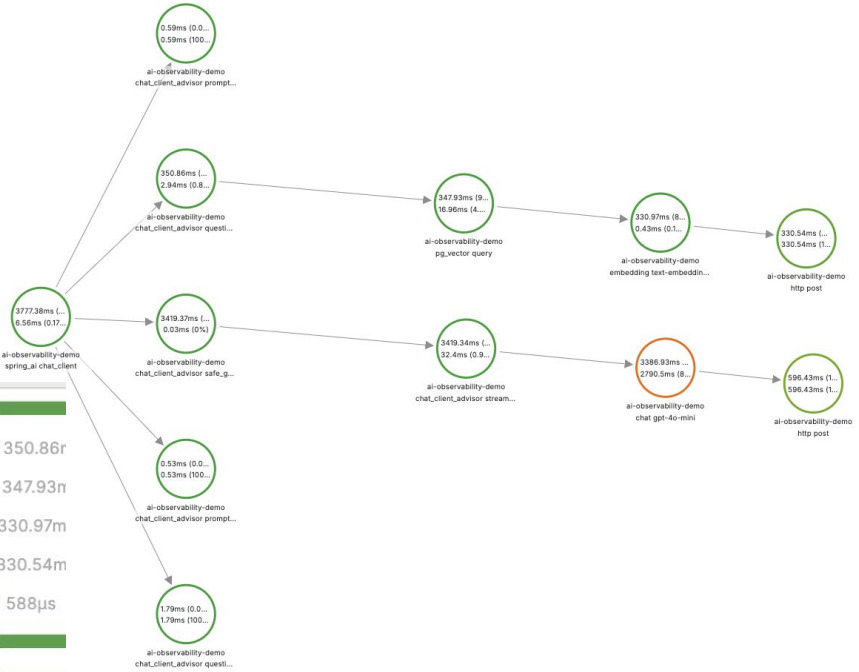
THENEWSTACK



Spring AI Observability

Micrometer





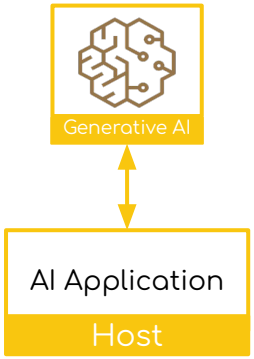
Wouldn’t it be nice ...

- To be able to use **Tools** written in other languages?

- To have Spring AI Tools used in non-Java envs?

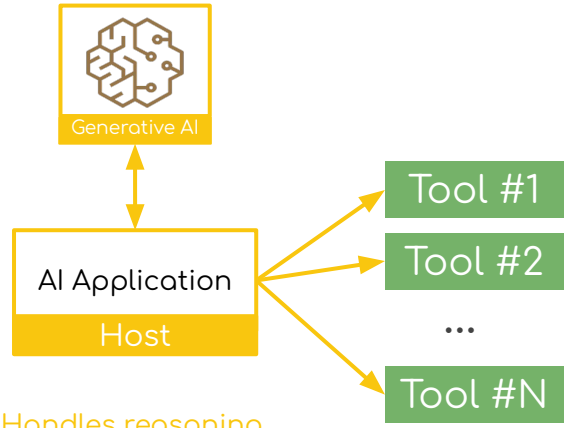
Model Context Protocol (MCP)

1 LLM



Text-In / Text-Out

2 LLM + Tool Calling



Handles reasoning and planning. Decides which Actions to take

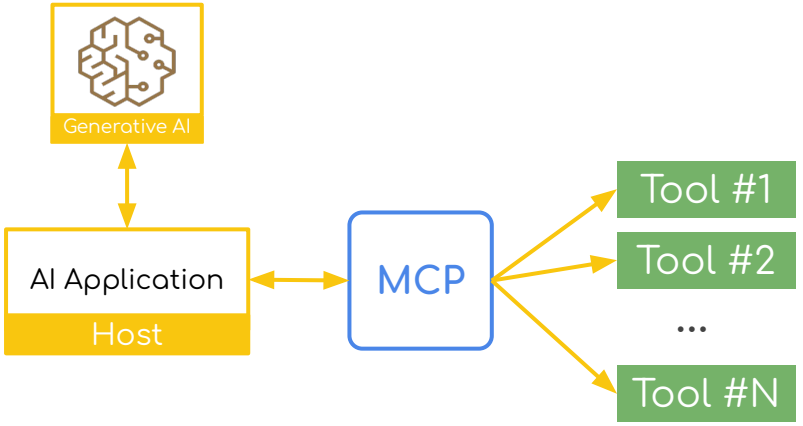
Brain LLM

Agent

Body Tools

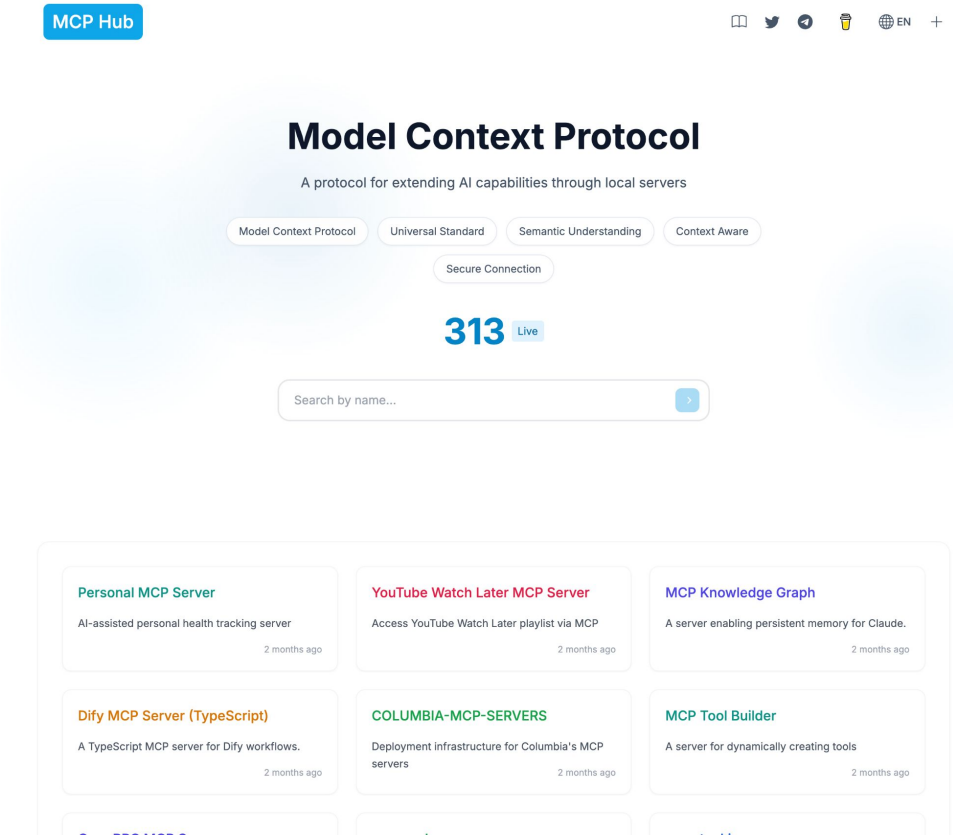
Equipment to Interact with the environment

3 LLM + MCP

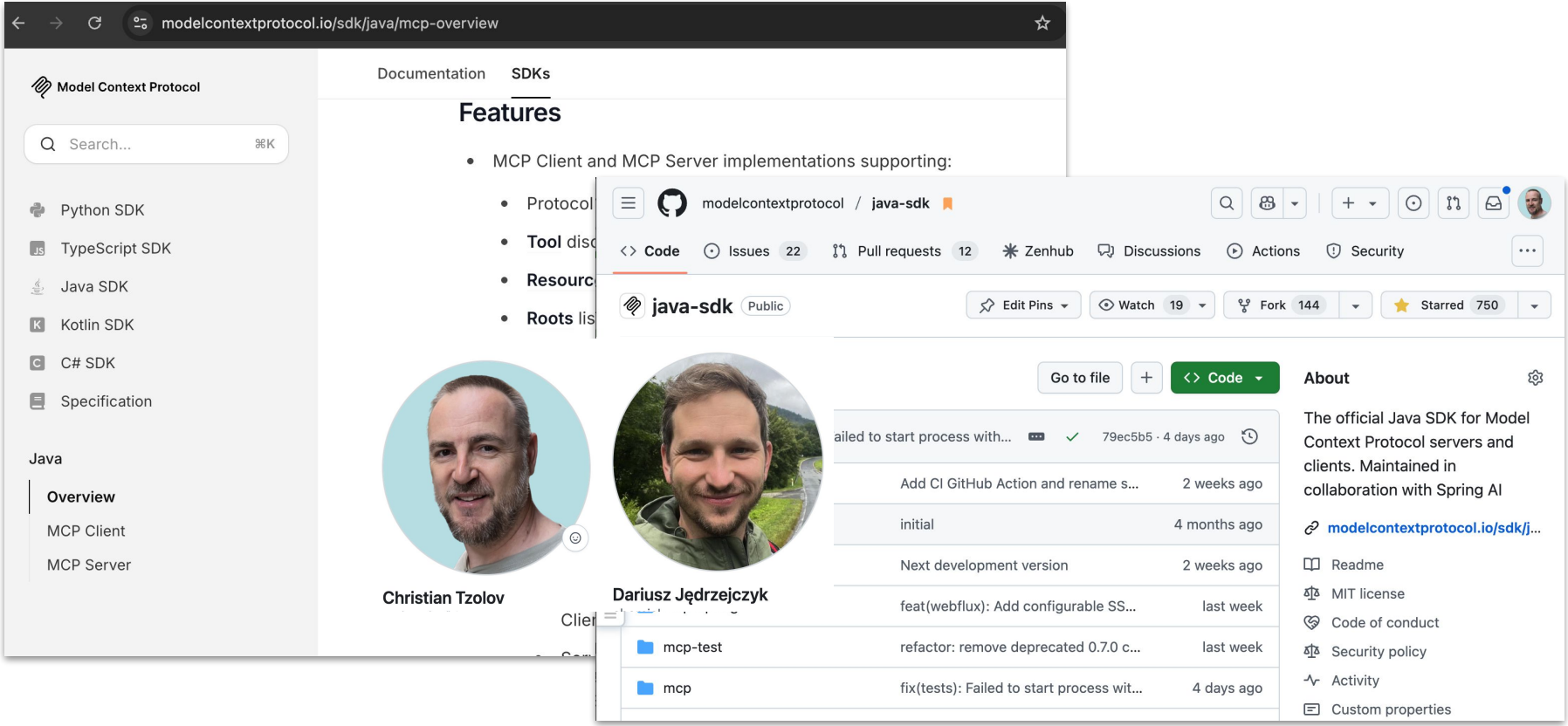


Unified way to connect **AI applications** to different **data sources, tools, ...**

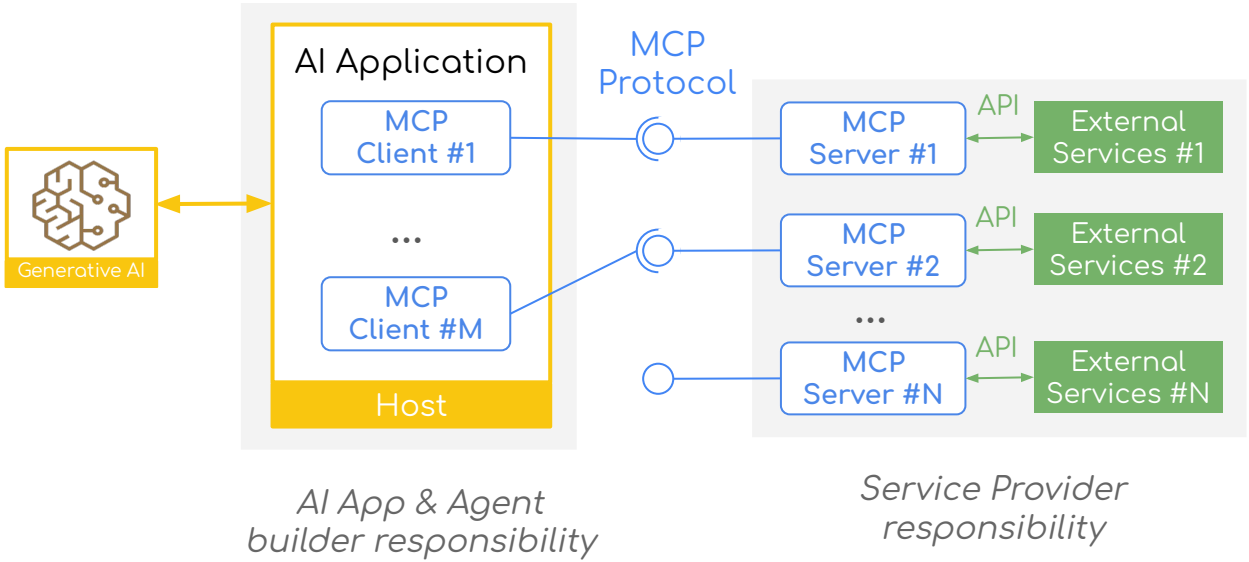
MCP Client & Server Ecosystem



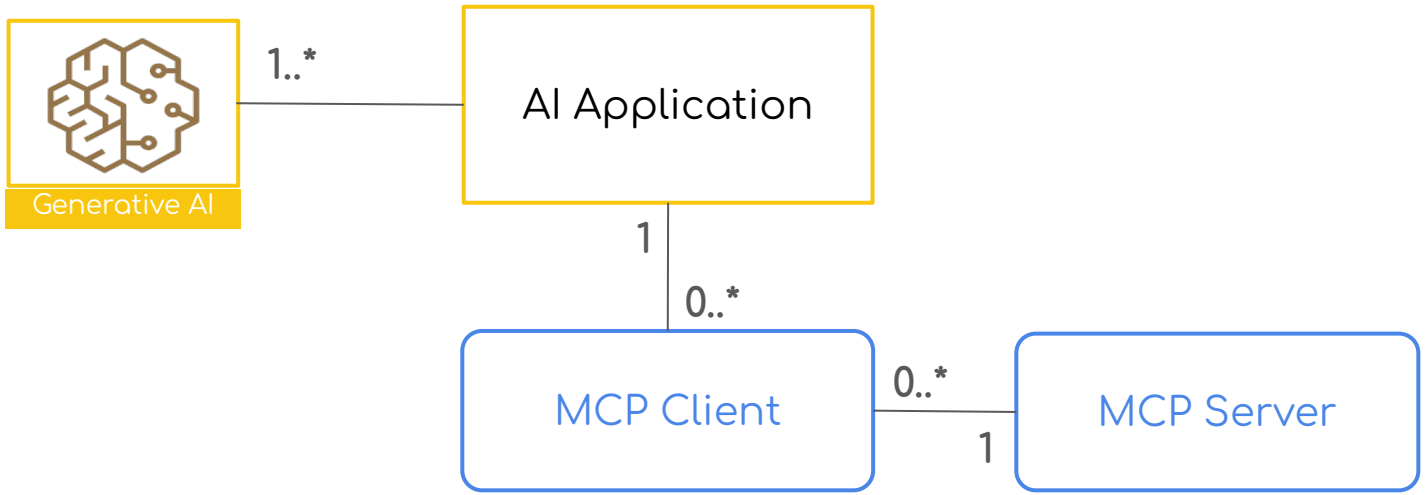
MCP Official Java SDK



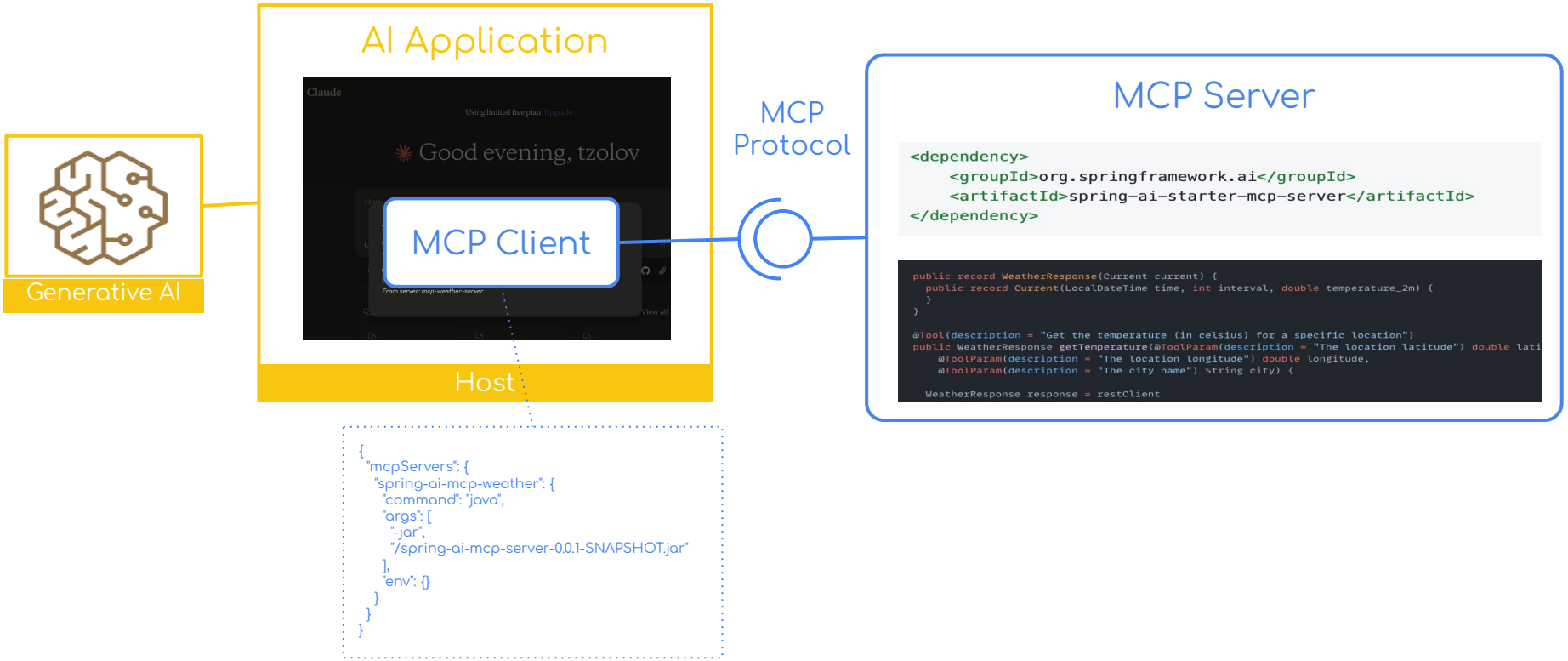
MCP Components



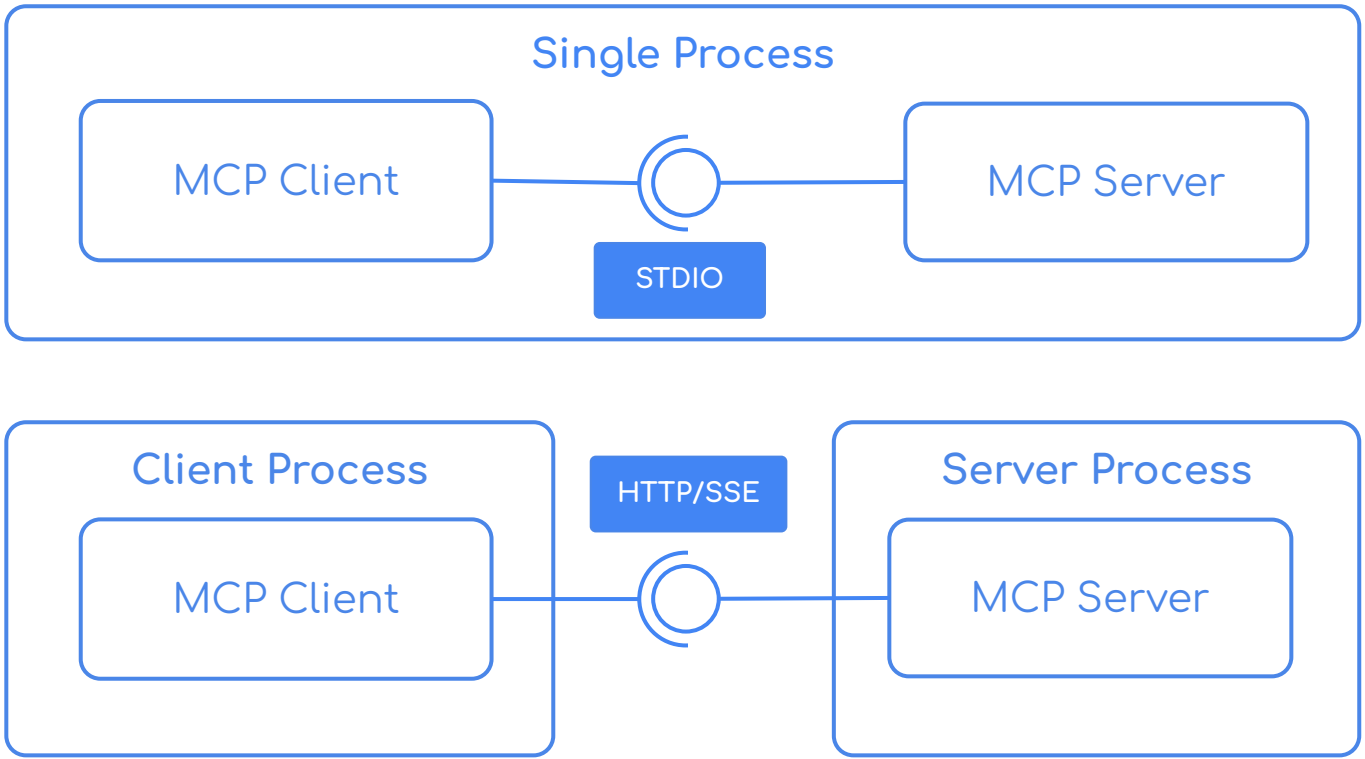
MCP Cardinalities



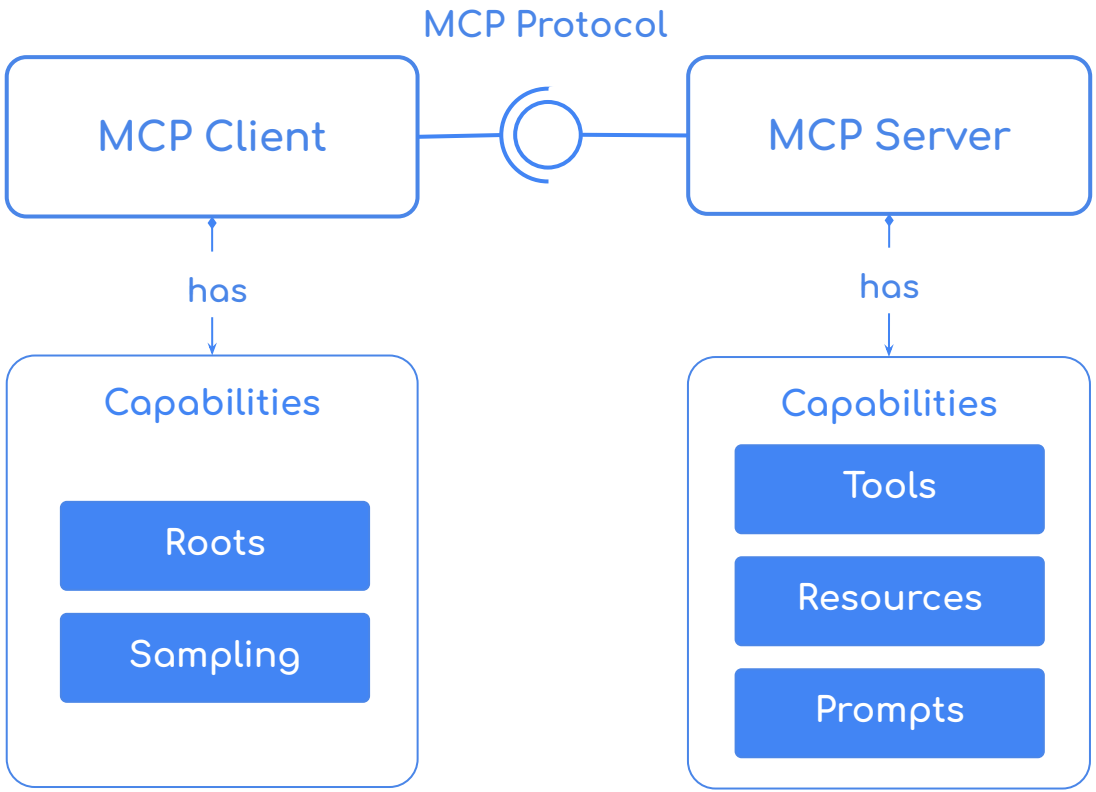
Spring AI MCP Server - Demo



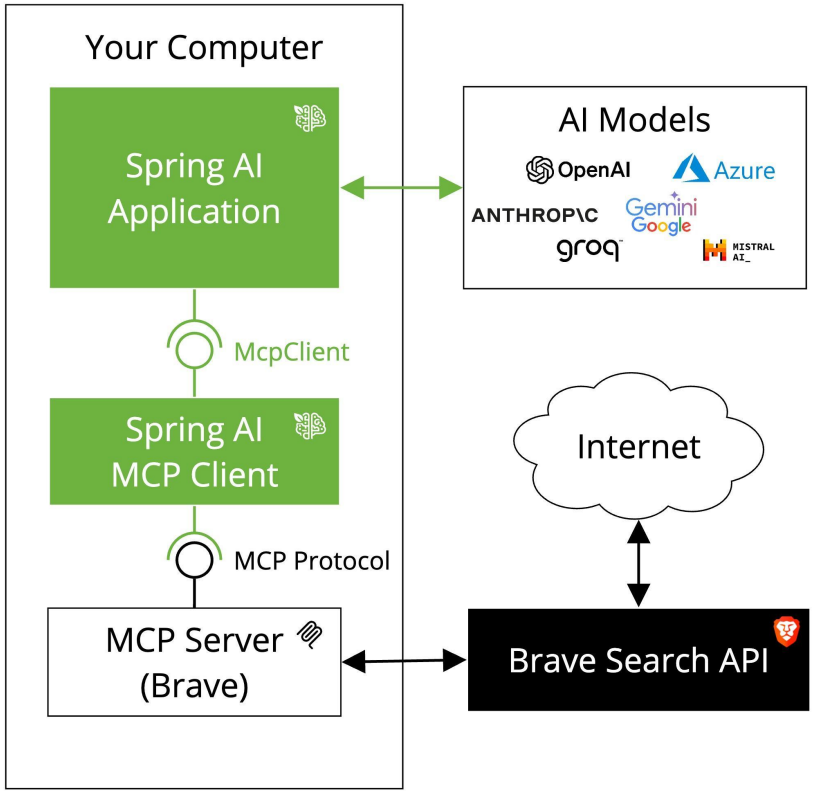
MCP Transports



MCP Capabilities



Spring AI + JavaScript WebSearch & Filesystem Tools



Application.java

```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();
  };
}
```

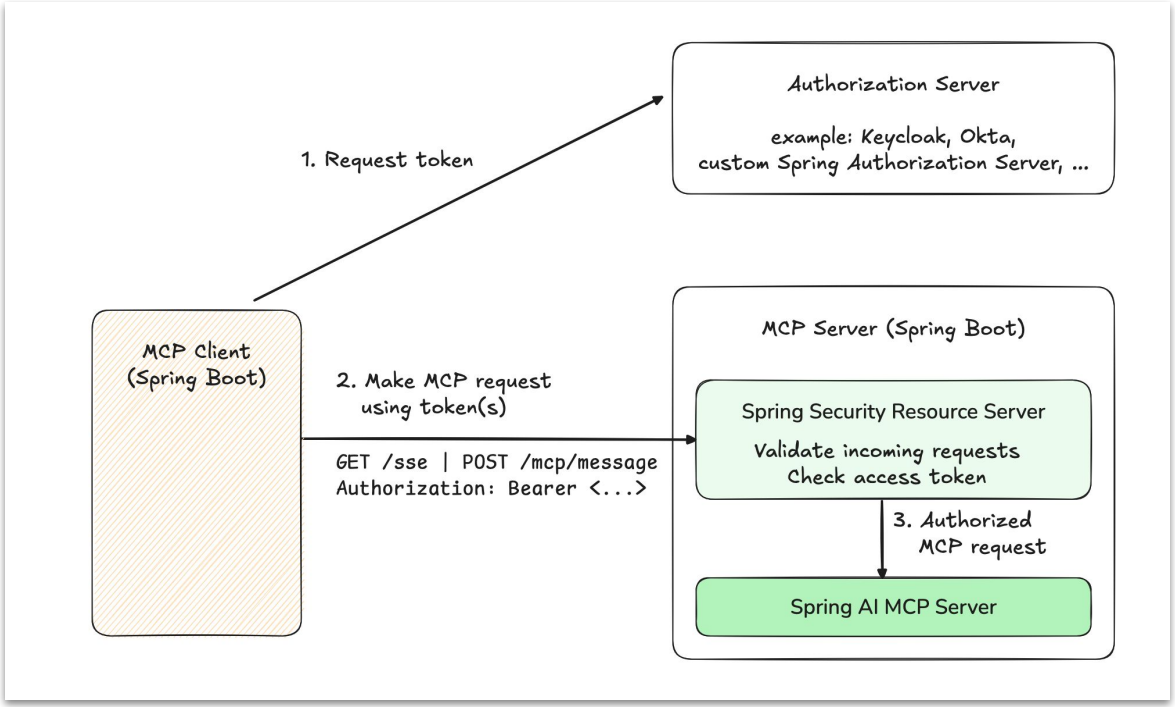
mcp-servers-config.json

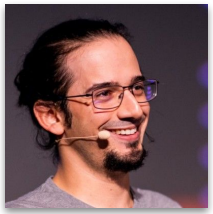
```json
{
  "mcpServers": {
    "brave-search": {
      "command": "npx",
      "args": [
        "-y",
        "@modelcontextprotocol/server-brave-search"
      ],
      "env": {
      }
    },
    "filesystem": {
      "command": "npx",
      "args": [
        "-y",
        "@modelcontextprotocol/server-filesystem",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

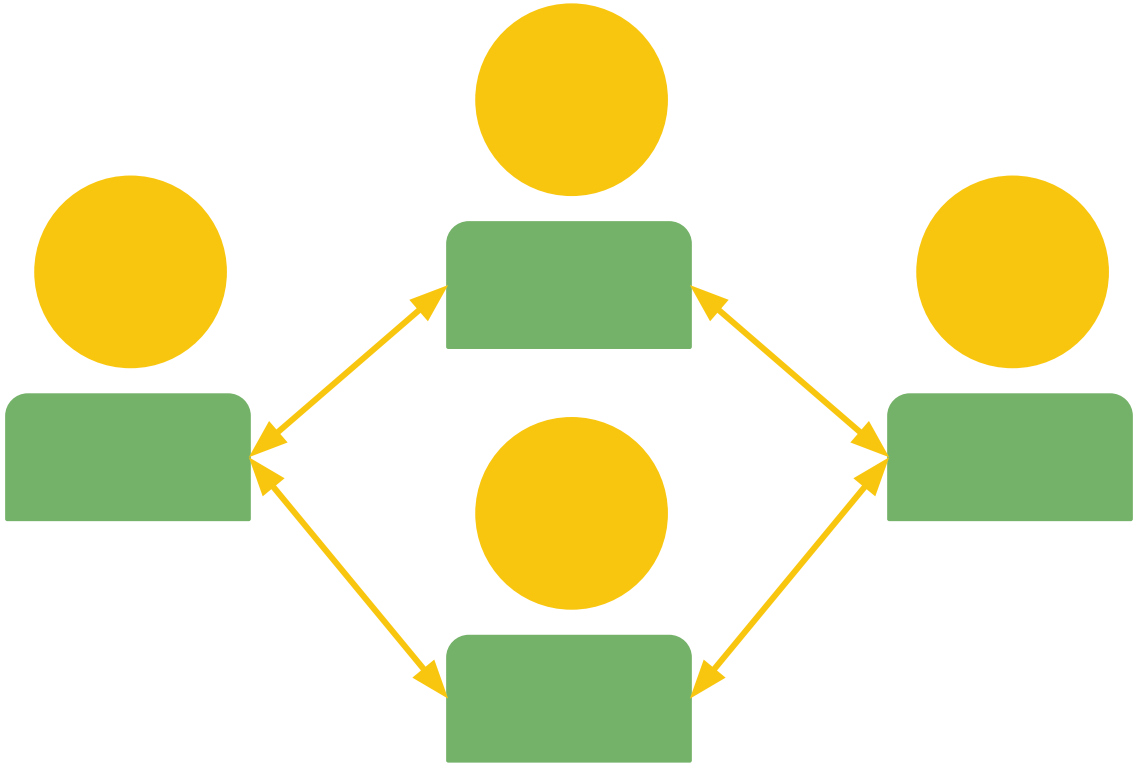
Spring AI MCP & Spring Security OAuth





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

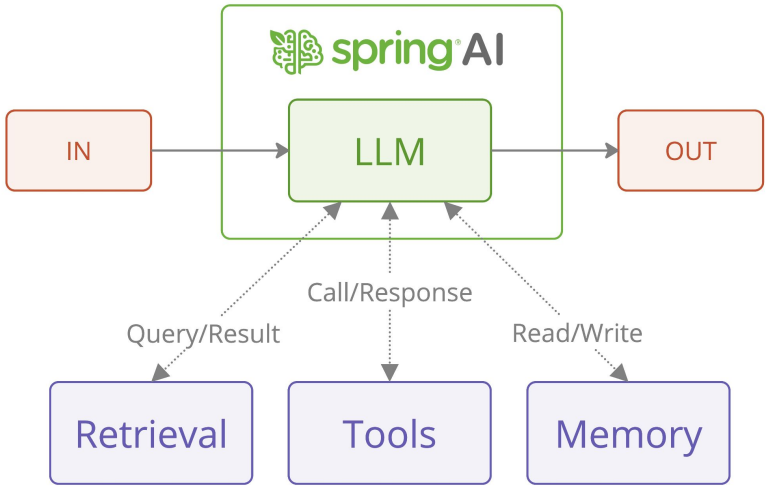
Agentic Systems



Agentic Systems - Prescriptive vs Autonomous

- **Workflows** - LLMs and Tools are **orchestrated through predefined** paths, e.g. **prescriptive**

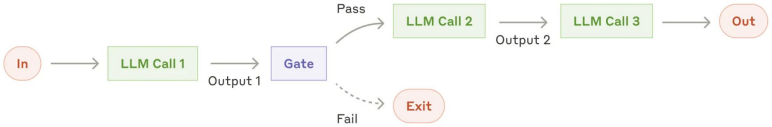
- **Autonomous Agents** - LLMs **autonomously plans** and executes the processing steps toward accomplishing the tasks - e.g **autonomous**



Agentic Workflows

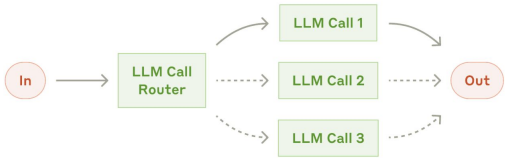
Chain Workflow

Break complex tasks down into simpler, more manageable steps



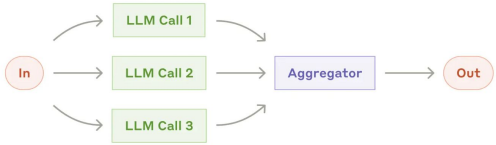
Routing Workflow

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



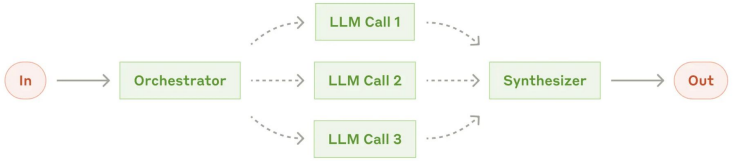
Parallelization Workflow

Work simultaneously on tasks and aggregate outputs



Orchestration - Worker Workflow

Central LLM orchestrates task decomposition. Specialized workers handle specific subtasks

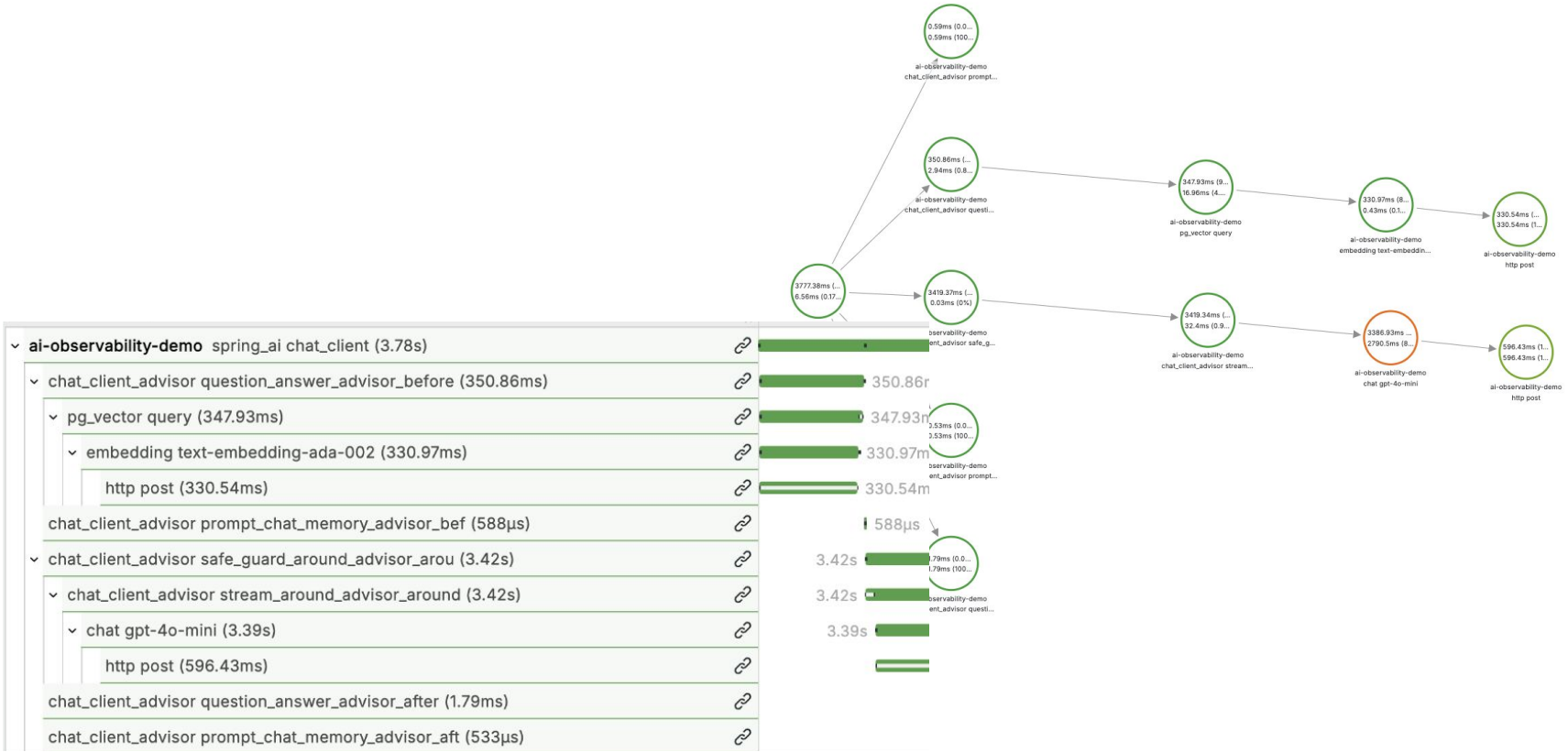


Evaluator - Optimizer Workflow

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



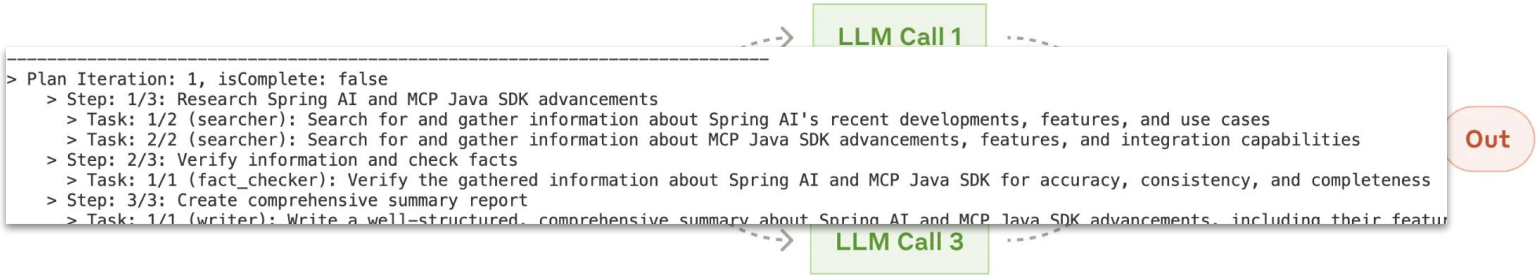
Agentic Workflows

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

Examples | **Building Effective Agents** | **MCP**

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