

From Scratch to Scalable: Building Smarter AI Agents with Frameworks



Jetro Coenradie




Daniël Spee

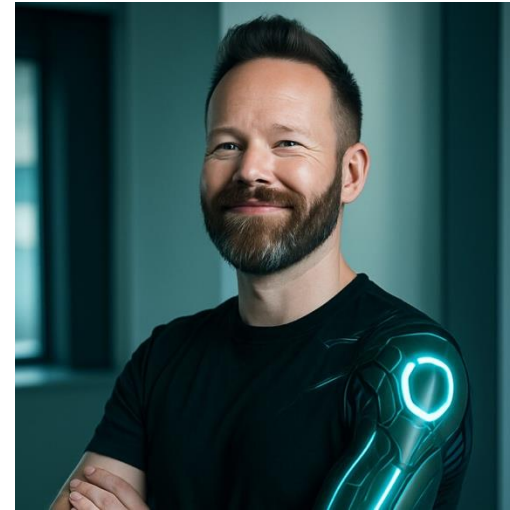
DEVOXX™

luminis. / part of yuma




Jetro Coenradie

 /in/jetro



Daniël Spee

 /in/dspee

What are you going to do?

Create an Agent using Java + Spring Boot.

The prompt, ReAct, tools, memory, and multi-agents

Create an Agent using Spring AI.

Tools, memory, MCP, Guardrails, and multi-agents

Work with Evaluations.

LLM Evals, Human Evals and Accuracy

What are you going to do? (OPTIONAL)

Create an Agent with Embabel.

Tools, MCP

Secure your MCP Server.

OAuth

Schedule

09:30 - 09:45: Introduction + What are agents?

09:45 - 10:30: Work on Java Agent

10:30 - 10:45: Multi-agent discussion + Spring AI Introduction + MCP server

10:45 - 11:30: Work on Spring AI Agent

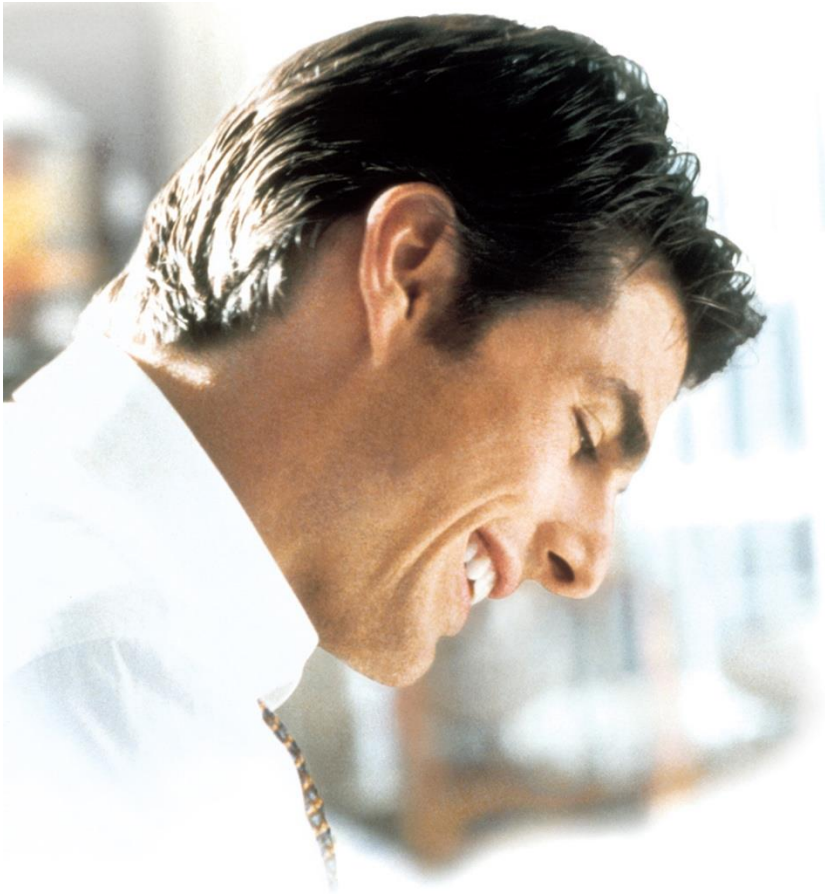
11:30 - 11:45: Evaluations + GuardRails

11:45 - 12:20: Work on Evaluations and Guardrails

Optional: Work on Embabel Agent and secure MCP with Oauth

12:20 - 12:30: Wrap-up

What is an Agent?



Jerry Maguire ~ Sports Agent

One who is authorized to act for or in the place of another

~ Merriam Webster

A software agent is a program that acts for a user or another program.

~ Wikipedia

<https://www.merriam-webster.com/dictionary/agent>

https://en.wikipedia.org/wiki/Software_agent

What about AI Agents?

AI agents are software systems that use AI to pursue goals and complete tasks on behalf of users. They show reasoning, planning, and memory and have a level of autonomy to make decisions, learn, and adapt.

~ Google cloud

Key aspects of an Agent

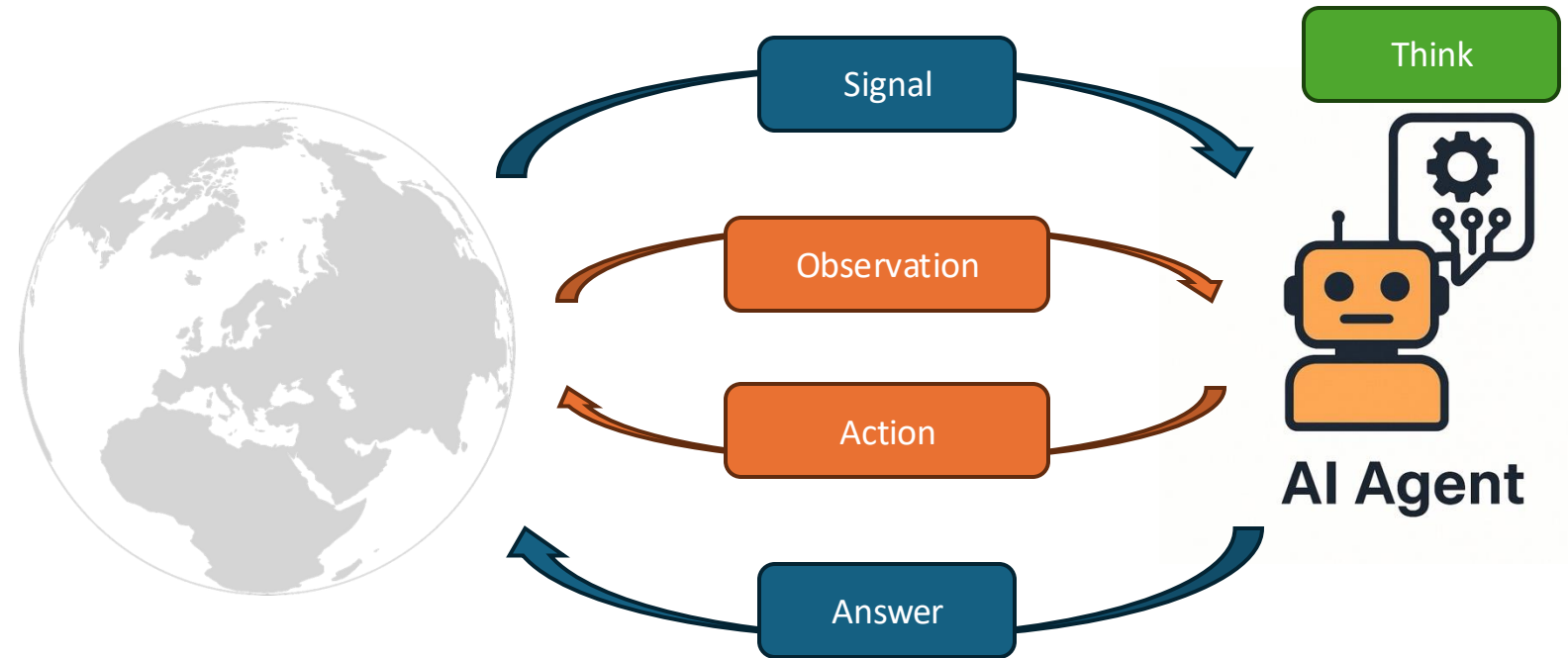
Reasoning

Acting

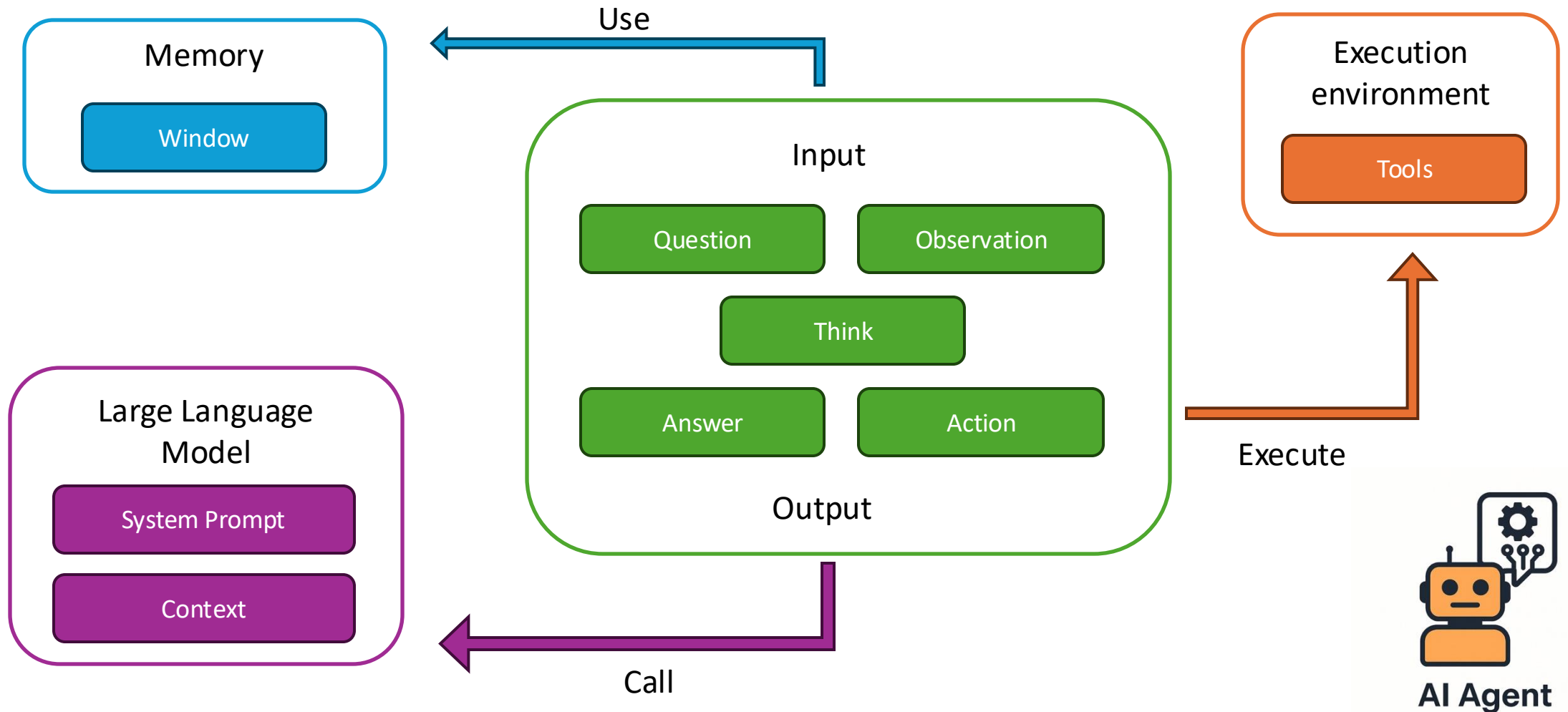
Observing

Planning

Collaborating

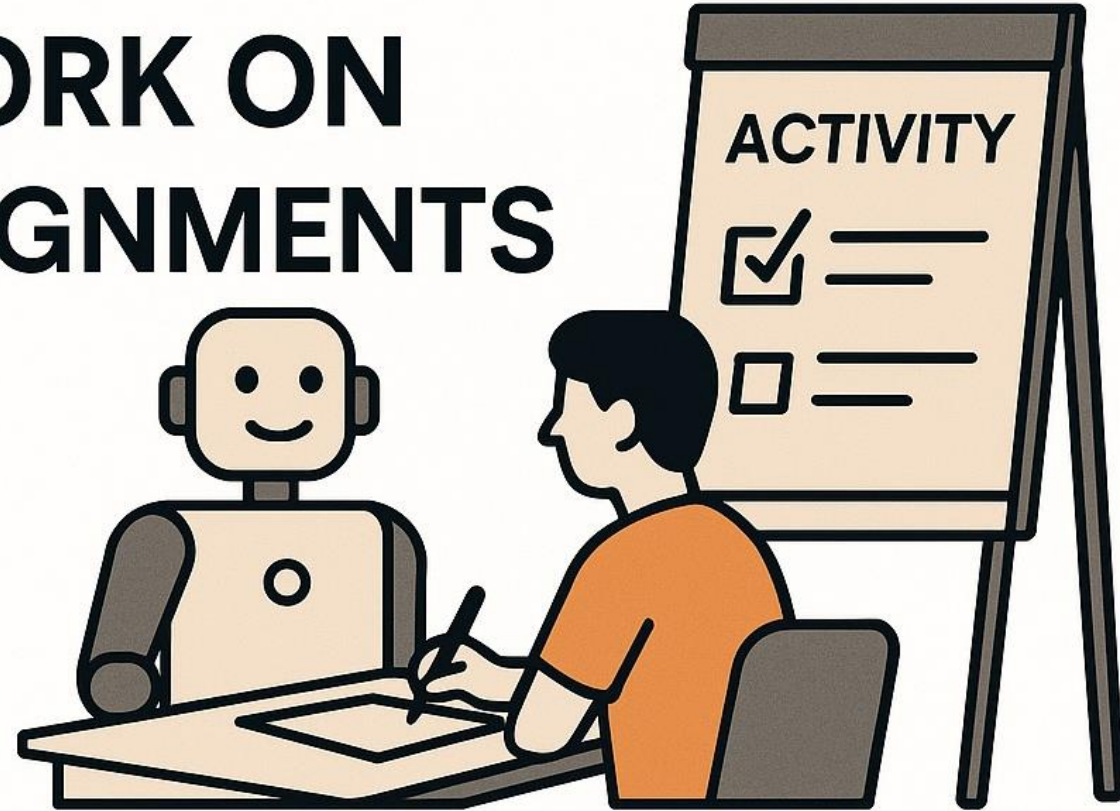


The Plain Java Agent



TIME TO WORK ON ASSIGNMENTS

Plain Java Agent



The assignments

- Check out the repository.
- Review the README.md file to become familiar with the project.
- The assignments are in the `Assignments` folder.
- Text blocks help you get familiar with the assignment.
- The code blocks tell you what to do.

<https://github.com/RAG4J/devoxx-2025-assignments>

Project layout

Assignments: assignments markdown files

Data: data files for favourites and evals

Logs: logs for the favourites mcp server

Core-agent: files shared between all agents

Web-app: runs the UI for all agents.

Java-agent: implementation for Java agent

Springai-agent: implementation for Spring AI

Embabel-agent: implementation for Embabel

Favourites-mcp: MCP server

Eval-web-app: runs the evaluation web app

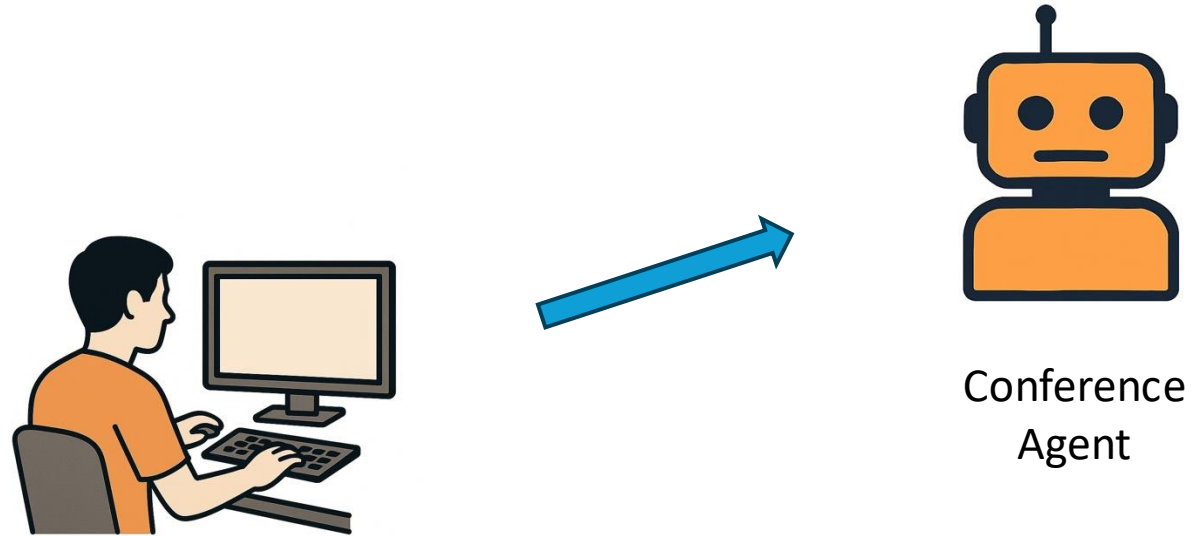
Connecting to OpenAI

You have two options to connect to OpenAI

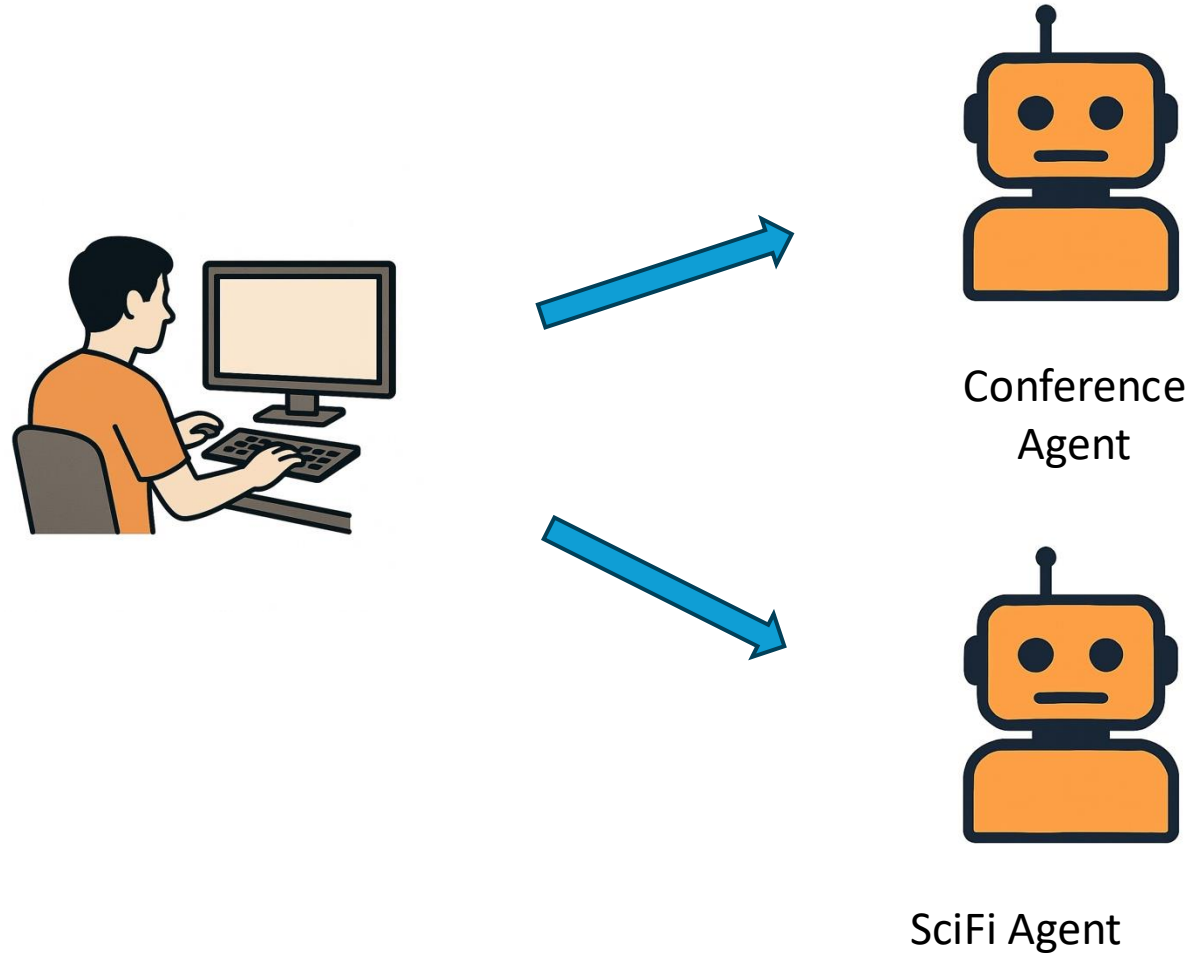
1. Use your own account, generate an API Key
2. Use our proxy. Fetch the key through the application token page

Password to get a key: jeda@devoxx

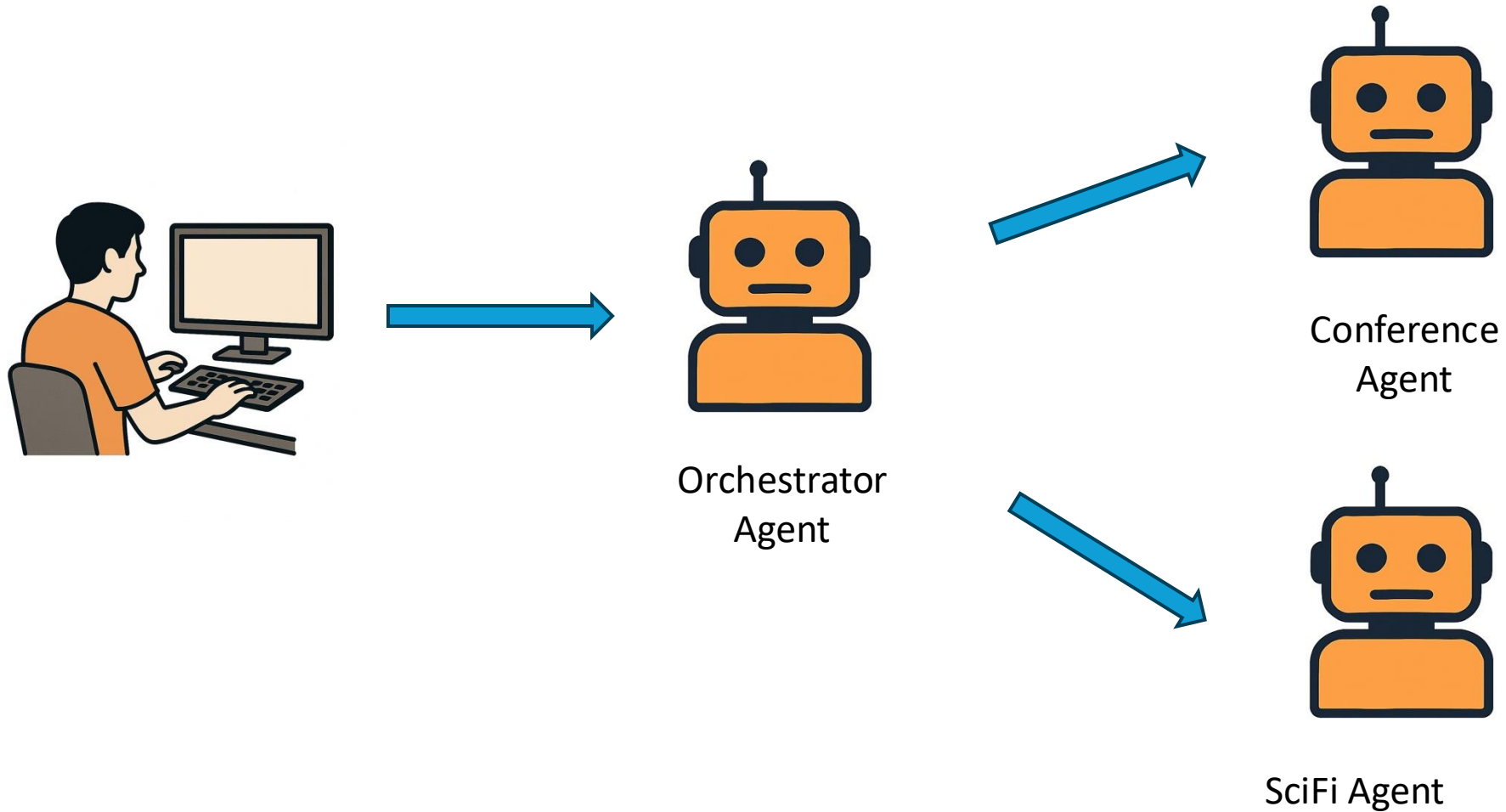
How do Agents work together?



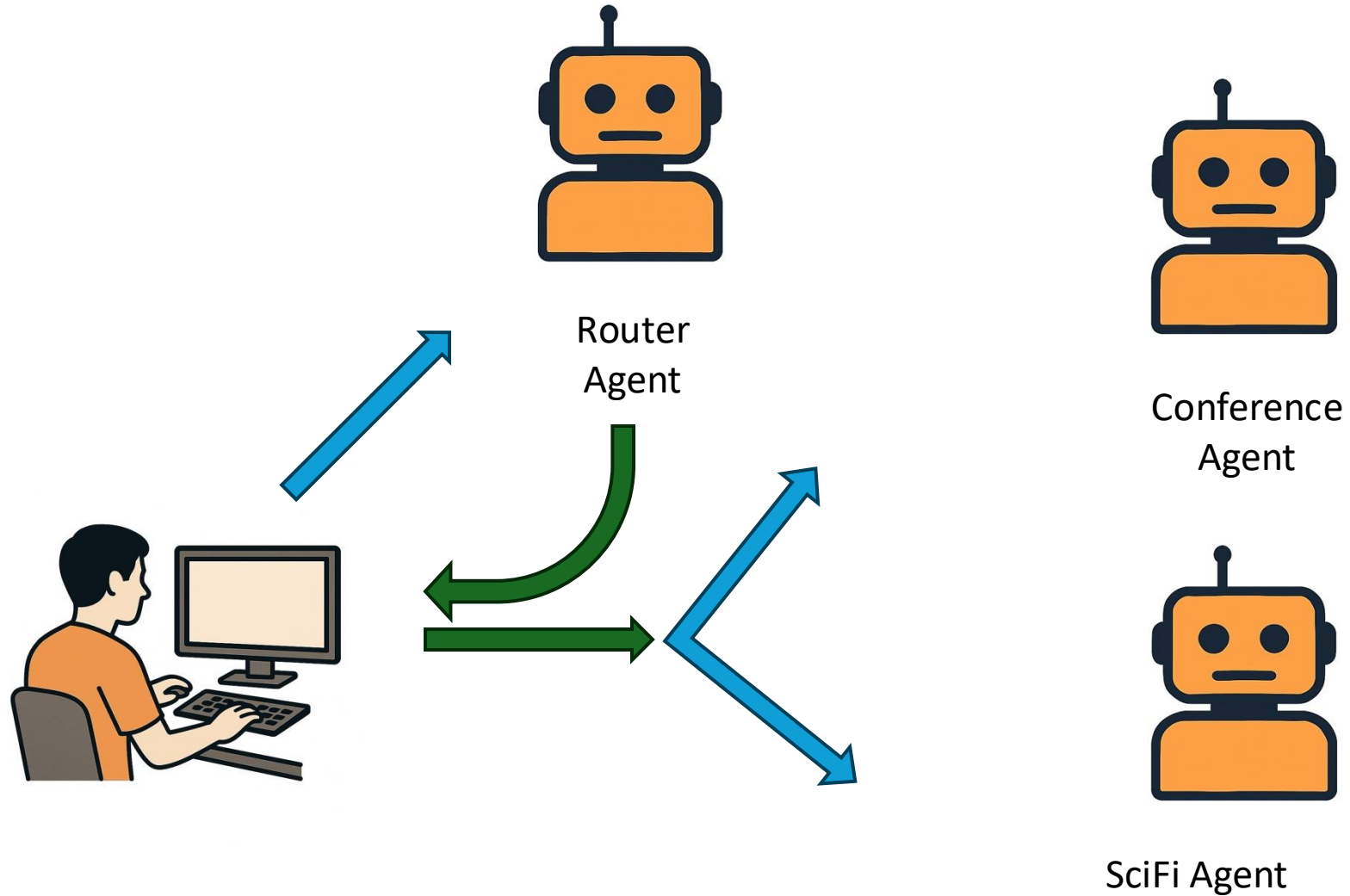
How do Agents work together?



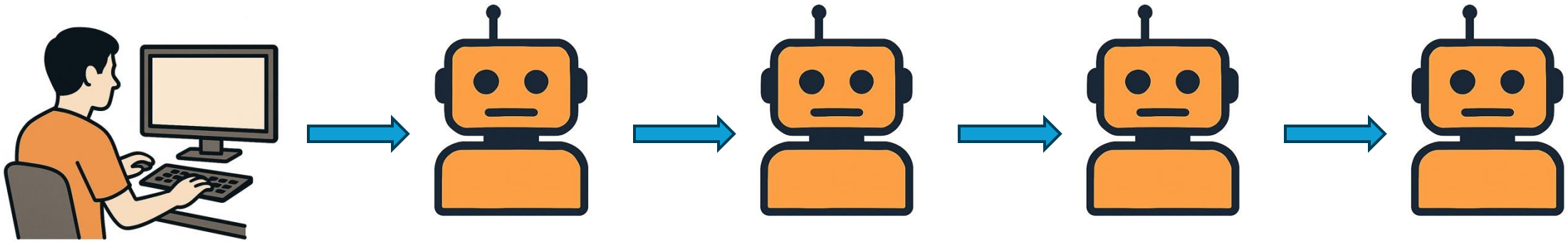
Orchestrator pattern



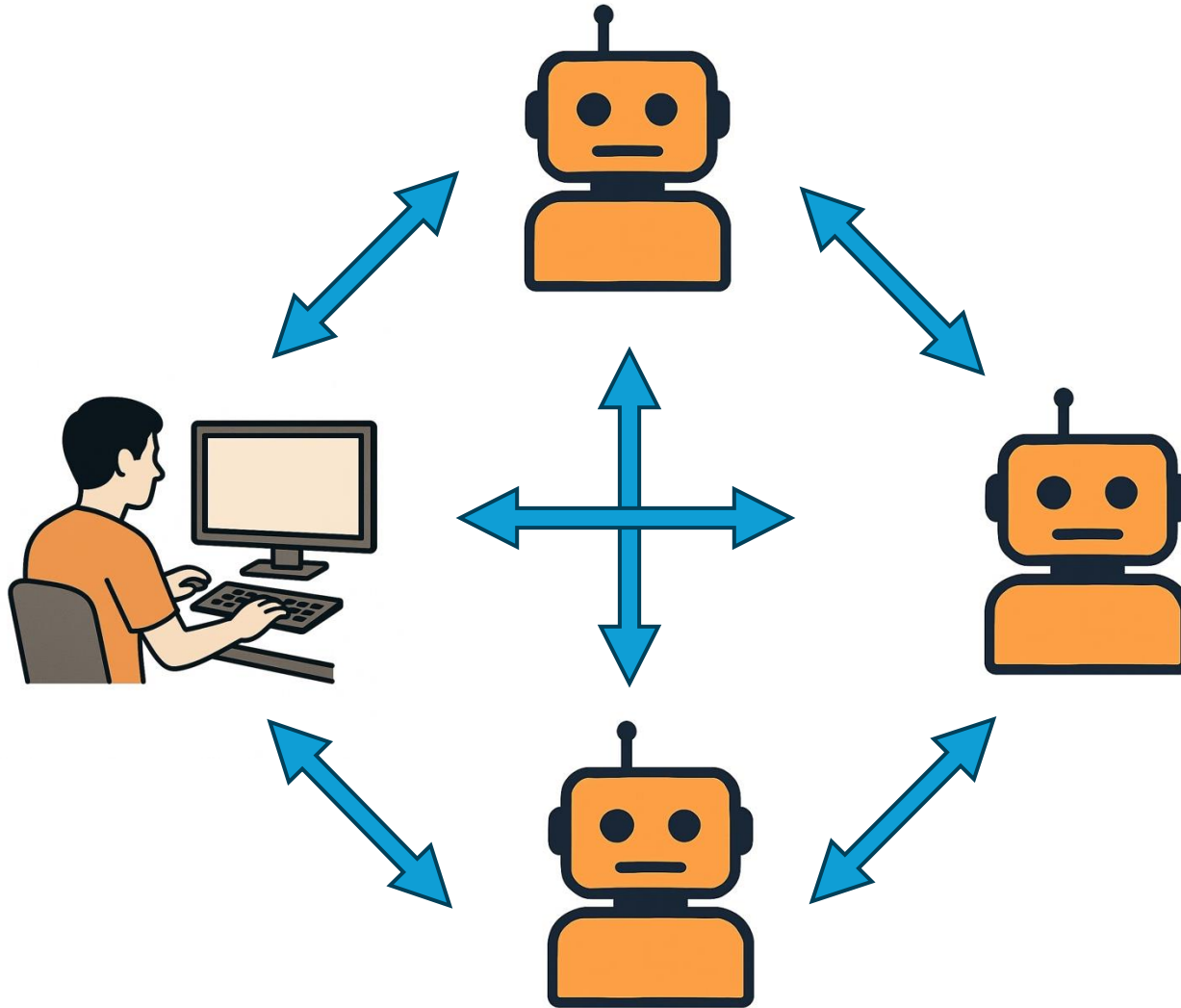
Router pattern



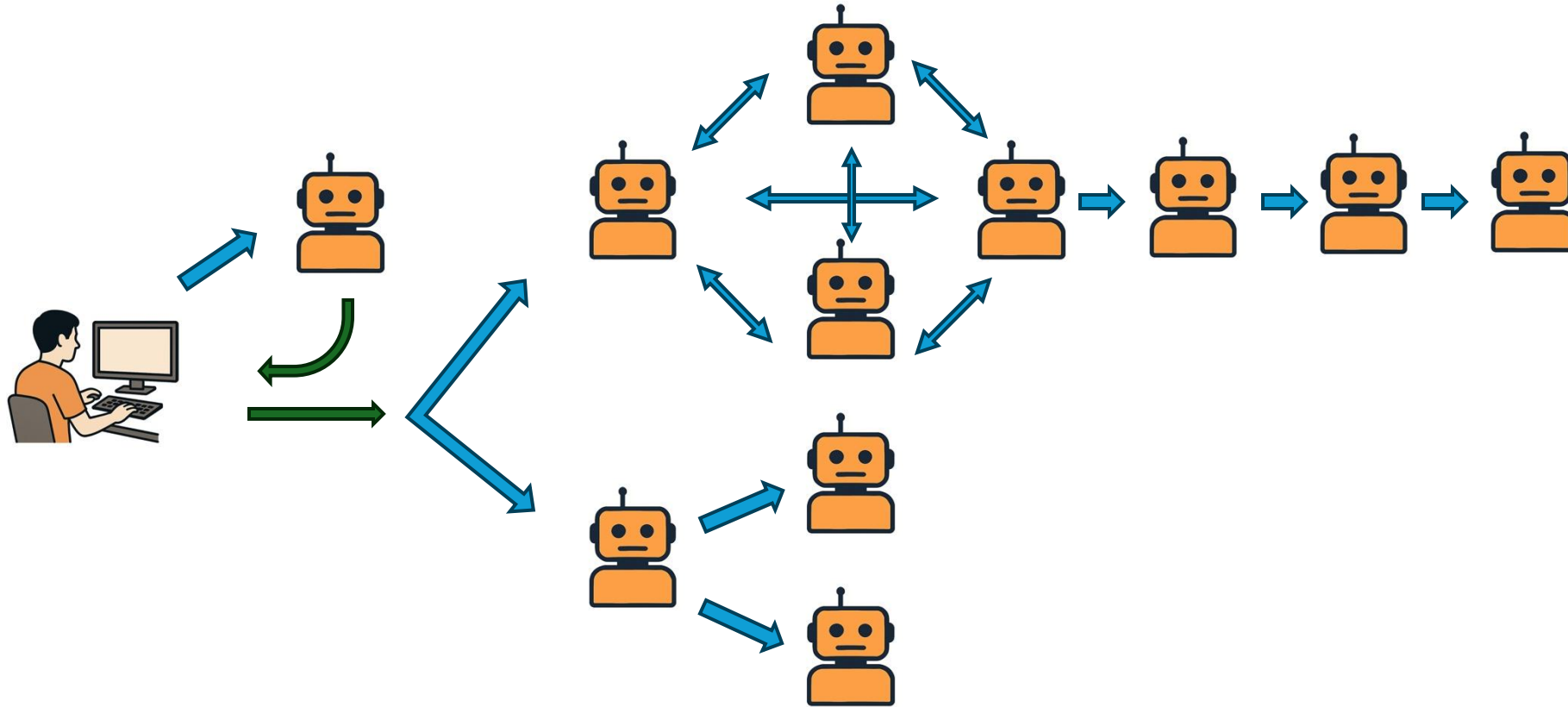
Workflow pattern



Group chat pattern



Chaos pattern?



Introduction of Spring AI

- Abstractions and sane defaults
- Works with multiple LLM providers for:
 - Chat, Embeddings, Text to Image, Text to Speech
- Works with a lot of Vector stores
- Has Observability out of the box
- Execution environment for your tools
- Integration with MCP

<https://docs.spring.io/spring-ai/reference/index.html>

Quick start

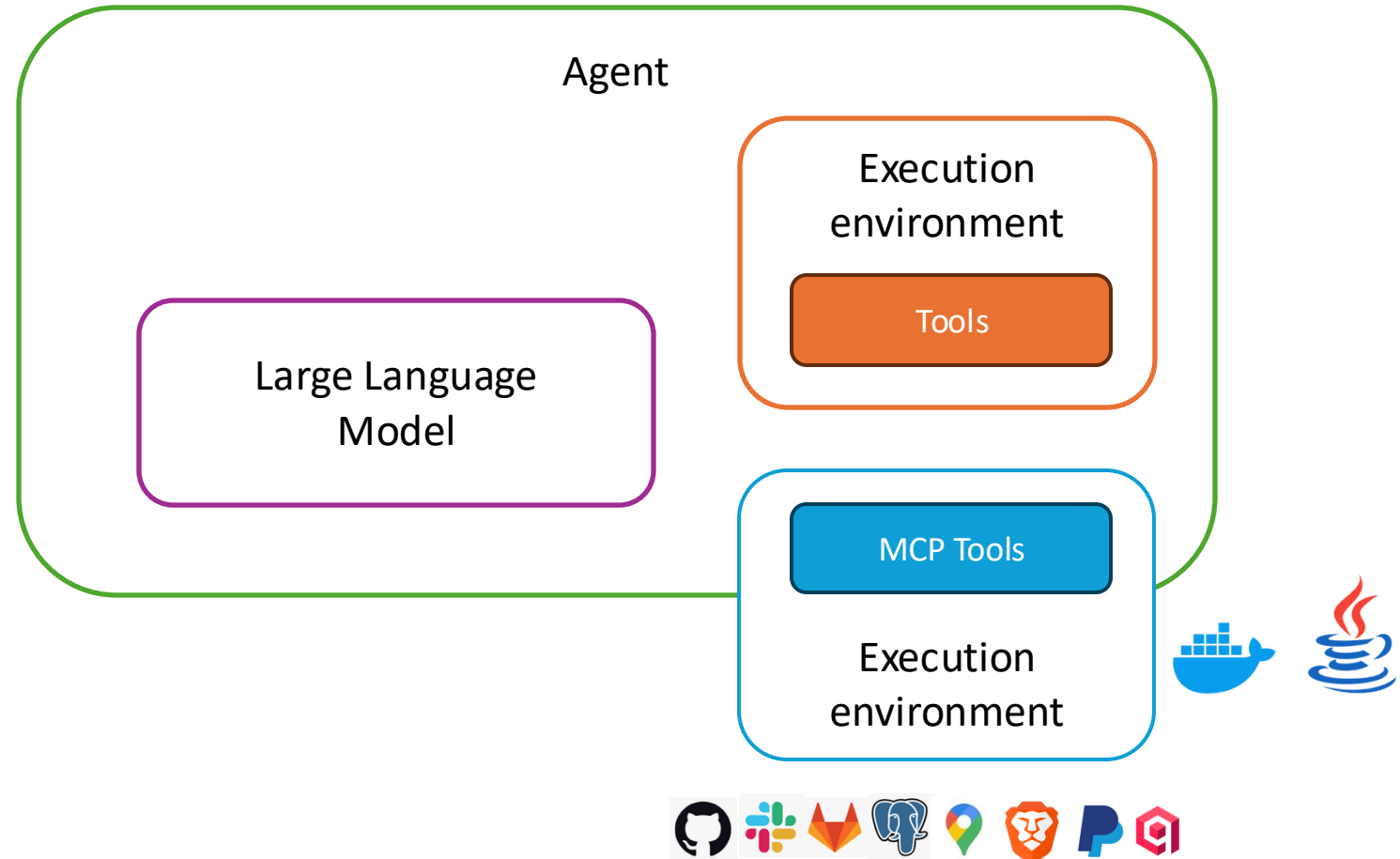
- Add the Bom to Maven for dependency management
- Add dependencies for the model provider
- Add extra dependencies: MCP client
- Create the OpenAiApi instance
- Create the ChatModel instance
- Create the ChatClient instance
- Create the memory, tools
- Use all these components to create the Agent

Model Context Protocol [MCP]

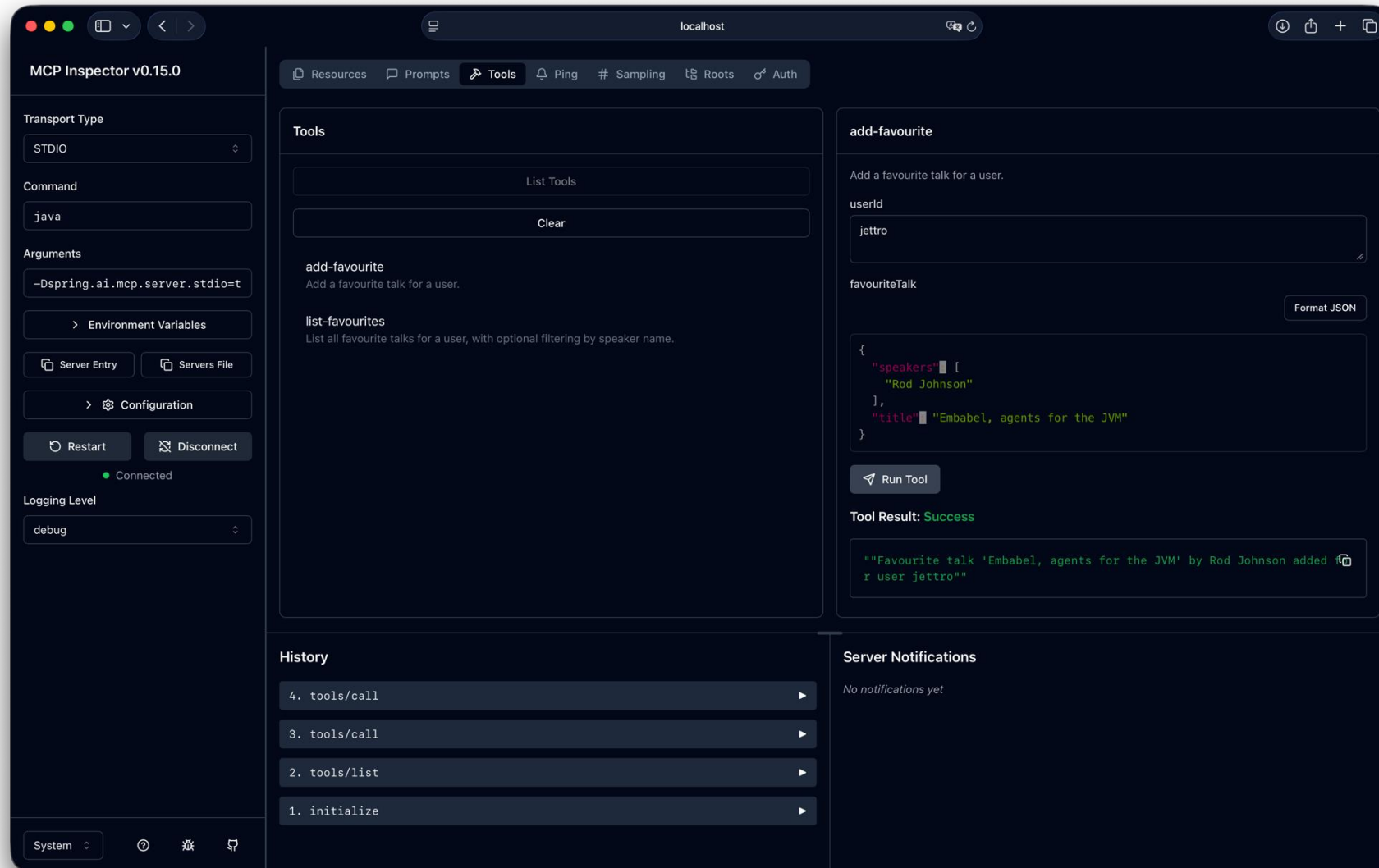
MCP enables you to build agents and complex workflows on top of LLMs and connects your models with the world.

~ <https://modelcontextprotocol.io>

Why do we need MCP?



Use the MCP Inspector



How do you use MCP?

Local access [StdIO]: Single user

```
spring:
  ai:
    mcp:
      client:
        enabled: true
        type: SYNC
      stdio:
        connections:
          favourites:
            command: java
            args:
              - -jar
              - "${FAVOURITES_MCP_JAR:${user.dir}/favourites-mcp/target/favourites-mcp-0.0.1-SNAPSHOT.jar}"
```

How do you use MCP?

Remote running [SSE]: Multi-user

```
spring:
  ai:
    mcp:
      client:
        enabled: true
        type: SYNC
      sse:
        connections:
          favourites:
            url: http://localhost:8081
            sse-endpoint: /sse
```

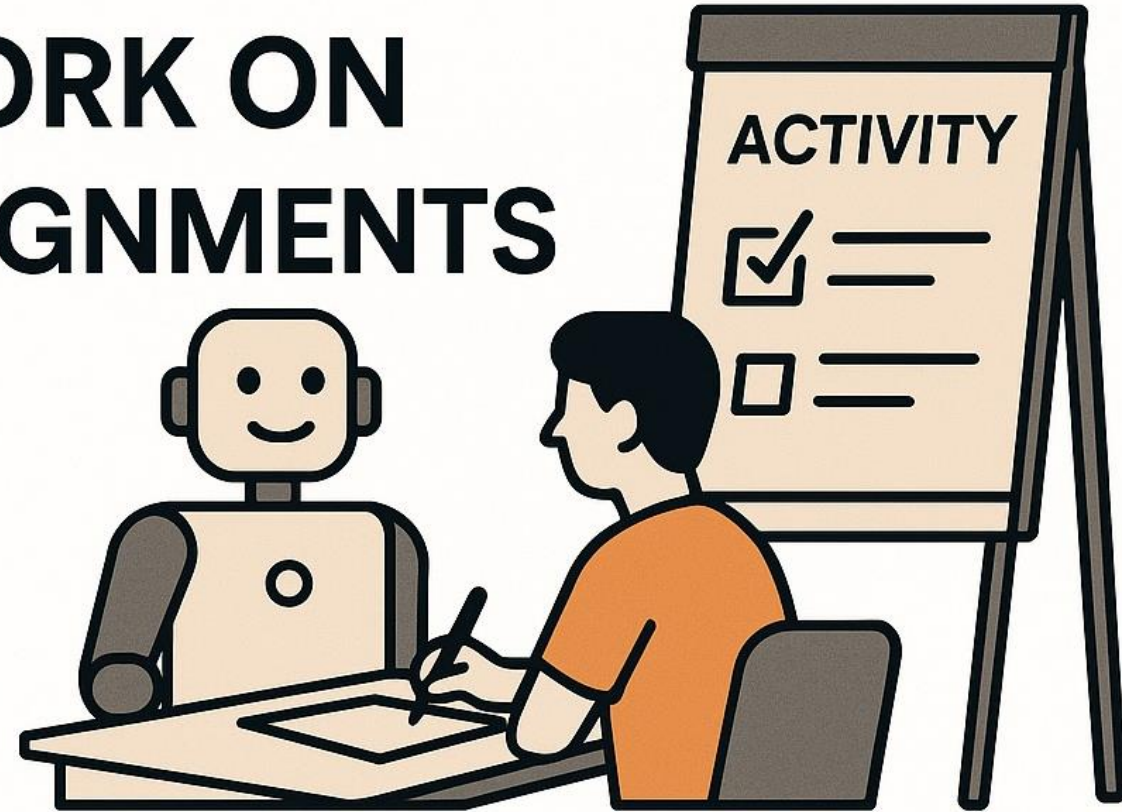
Is MCP Secure?

Not by default!

- Connection to the MCP
- Connection from the MCP to remote services
- Poorly programmed MCP servers (Without User consent)
- The MCP server provides tools with too many rights
- Malicious MCP servers that trick the agents

TIME TO WORK ON ASSIGNMENTS

Spring AI +
MCP



Guardrails

Guardrails are important for agents because they ensure safety, reliability, and alignment with human goals by preventing unintended, harmful, or biased behaviours.

Agentic AI Risk Management

OWASP and IBM Whitepapers

[Agentic AI - OWASP Lists Threats and Mitigations](#)

[IBM Whitepaper: Accountability and Risk Matter in Agentic AI](#)

- Intent Breaking & Goal Manipulation.
- Tool Misuse.
- Memory Poisoning.
- Identity Spoofing & Impersonation.
-

Guardrail Policies

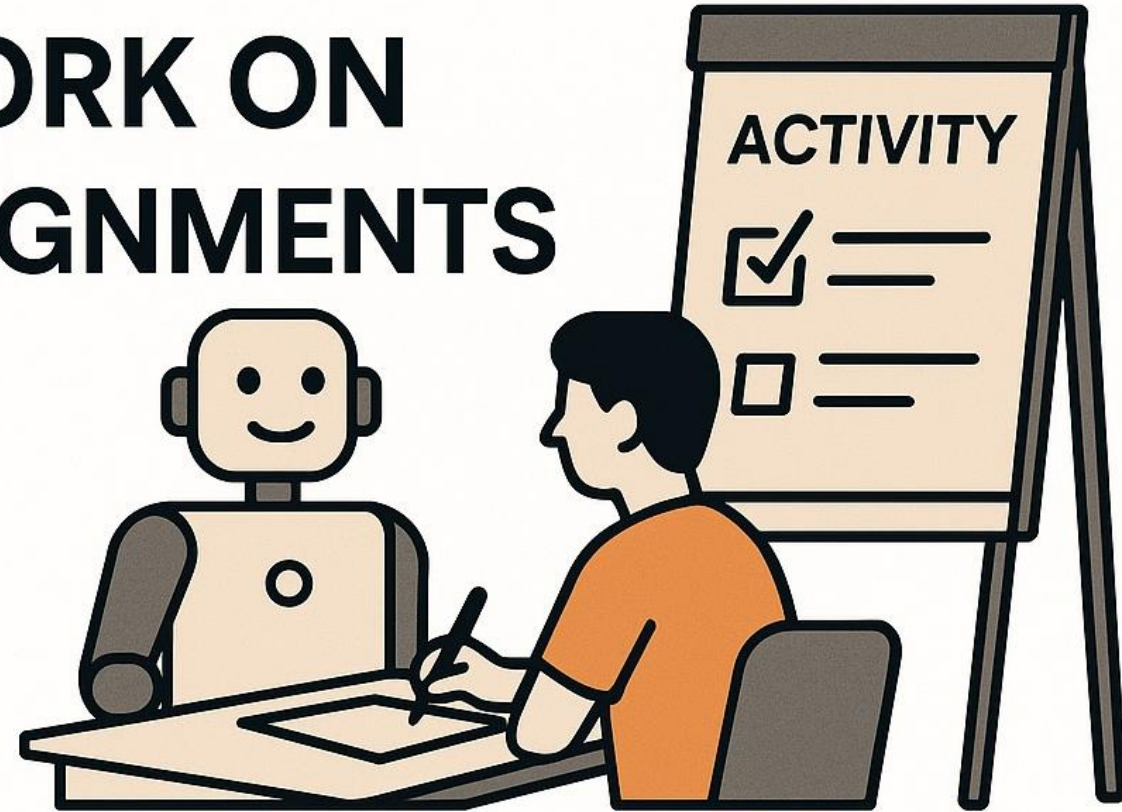
- Policy-first.
- Start simple.
- Make policies readable (non-technical).
- Use layered defence.
- Balance security with user experience.
- Fail closed, not open.
- Align guardrails with business and brand policies.
- Incorporate human oversight strategically.
- Log, monitor and evaluate.

Guardrail Types

- Input filters.
- Output validation.
- Agent and/or Tool restriction.
- Human-in-the-Loop.

MCP and
Guardrails

TIME TO WORK ON ASSIGNMENTS



Evaluations

The Hidden Key to Reliable Agentic AI

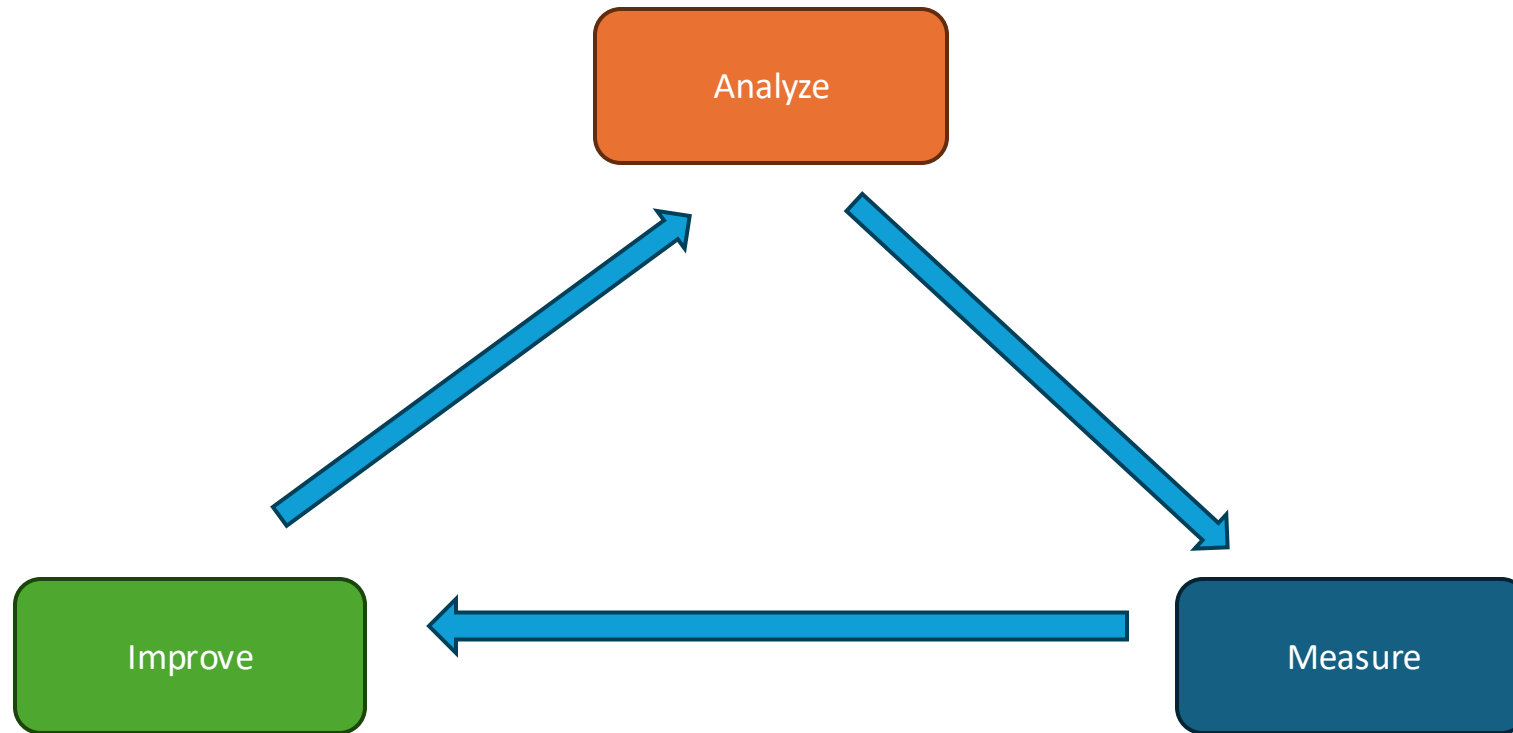
Why?

- LLMs are unpredictable, nondeterministic, and subtle in failure.
- No evals = trial-and-error, user complaints, unreliable systems.
- Evals = confidence in changes, systematic improvement.

Core challenges of Evals

- **Data understanding**
→ real users behave unpredictably.
- **Specification gap**
→ hard to encode “good” answers.
- **Inconsistent behaviour**
→ tiny changes, very different outputs.

Evaluation-driven workflow



End-to-End vs Component-Level

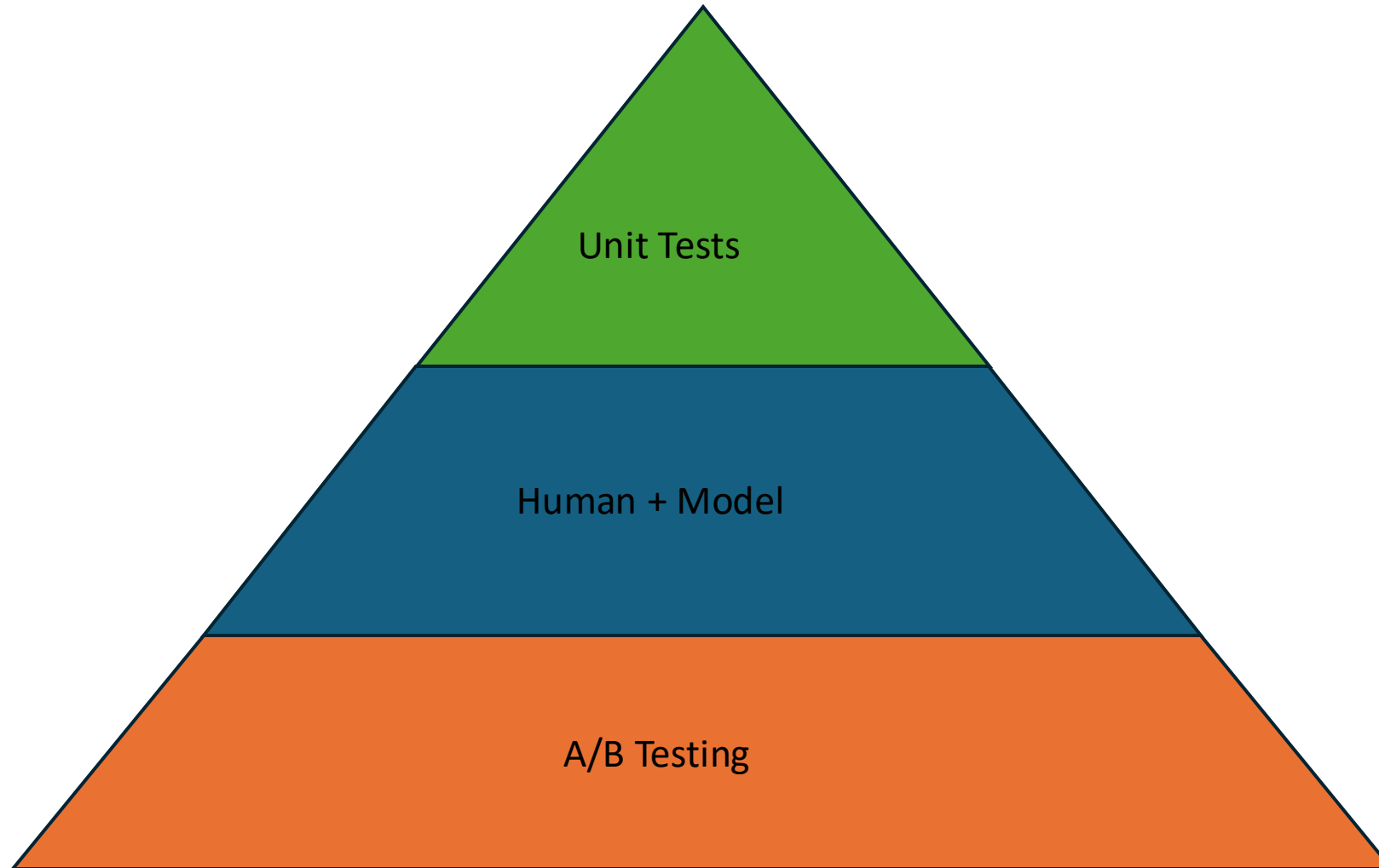
End-to-End:

- It's what users see.
- Correlate with business KPIs/OKRs.

Component-Level:

- More granular.
- For debugging.
- Improvements always impact End-to-End.

Eval Pyramid



Metric Types

Reference-based:

When a “golden answer” is known (exact match, semantic equivalence, task execution correctness, retrieval metrics).

When to use: deterministic tasks, structured outputs, code/SQL generation, RAG quality with labelled relevant docs.

Reference-free:

When multiple outputs can be valid (tone, safety / compliance, agent operational metrics, human-model agreement).

When to use: customer support, research assistants, creative drafting, multi-tool agents.

Principles for Success

- Start small, keep humans in the loop.
- Look at lots of data, automate only after.
- Build domain-specific evals.

→ Evaluation-driven workflow!

TIME TO WORK ON ASSIGNMENTS

Evaluations



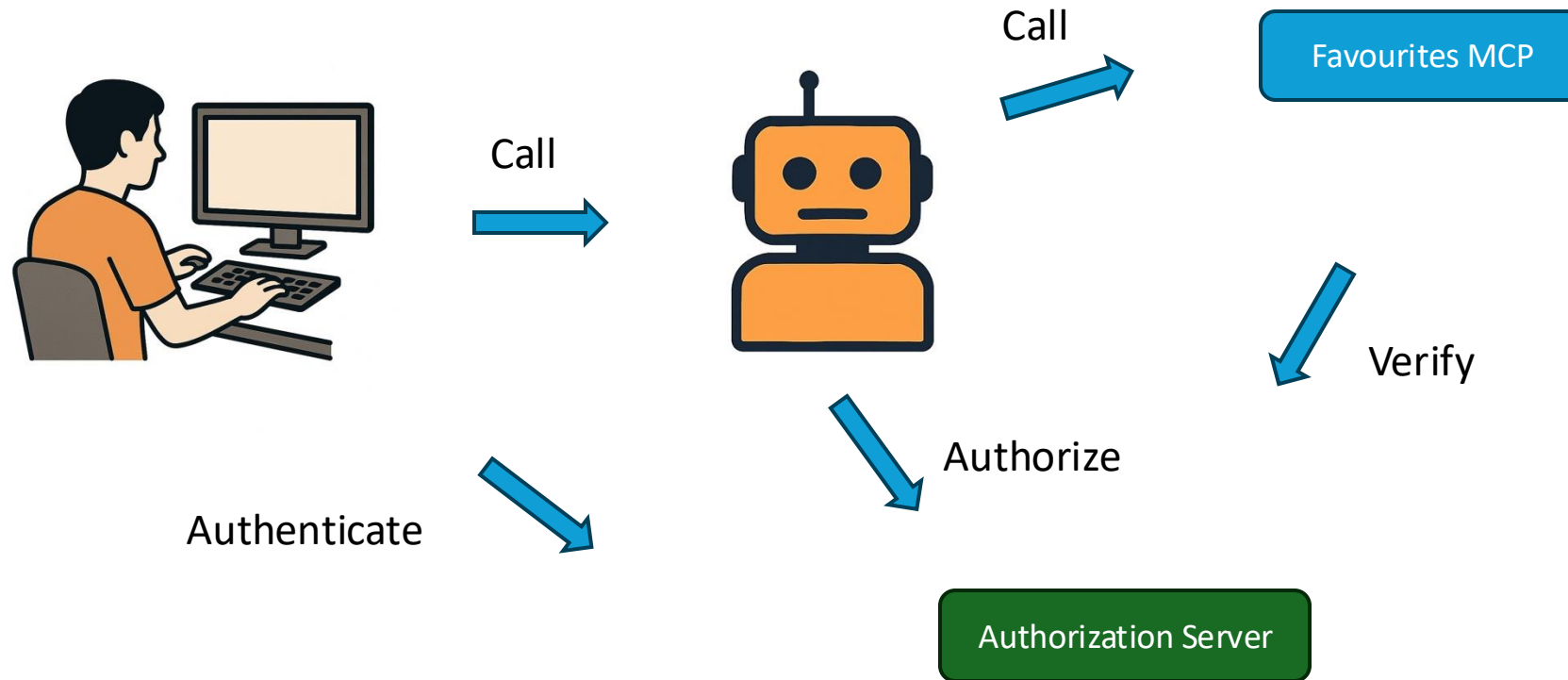
Embabel

High-level introduction

- Builds on top of Spring AI
- GOAP – Goal-Oriented Action Planning.
- Agent – Action – Goal
- Logs are essential for development
- It is new, a lot of developments



OAuth2 Integration



TIME TO WORK ON ASSIGNMENTS

Embabel + OAuth

