Building AI-Driven Applications

With Spring AI

Sandra Ahlgrimm
Senior Cloud Advocate, Microsoft

Timo Salm
Master Solutions Architect, Broadcom

September 2025

Microsoft & vmware by Broadcom

Who We Are

Sandra Ahlgrimm

Senior Cloud Advocate
Microsoft

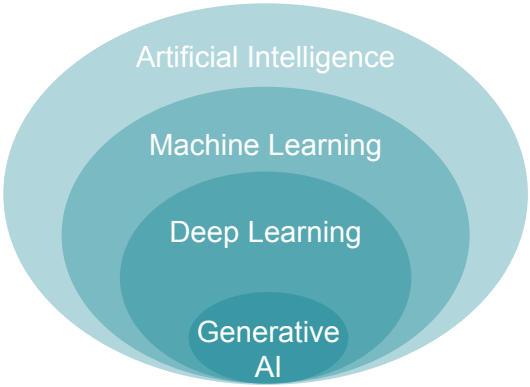
X (Twitter): @sKriemhild
Bluesky: @sandraahlgrimm.bsky.social
LinkedIn: https://linkedin.com/in/sandraahlgrimm

Timo Salm

Master Solutions Architect
Broadcom

X (Twitter): @salmto
Bluesky: @timosalm.bsky.social
LinkedIn: https://linkedin.com/in/timosalm

(Generative) AI Fundamentals

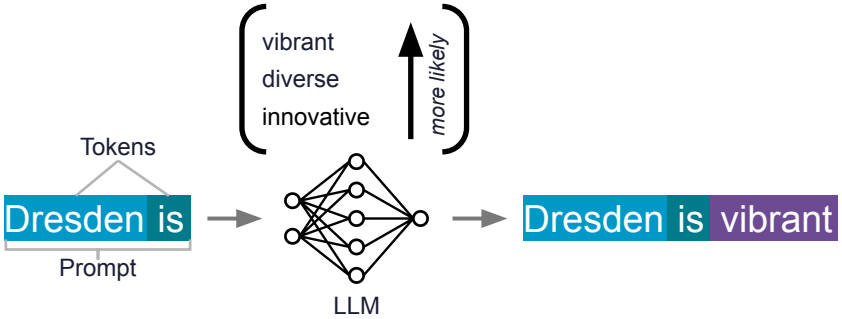


Artificial Intelligence: Machines are capable of performing cognitive functions typically associated with human minds

Generative AI is capable of generating text, images, or other data by utilizing models that learn patterns and structure of their training data

Foundation Model: A mathematical model trained on a huge amount of generic data that serves as the base for various generative tasks

Large Language Models (LLMs): AI models specifically designed to understand and generate human language



Why (Java) AI Frameworks?

```
var request = HttpRequest.newBuilder()
       .uri(URI.create ("https://api.openai.com/v1/chat/completions"))
       .header("Authorization", "Bearer YOUR_API_KEY")
       .header("Content-Type", "application/json")
       .POST(HttpRequest.BodyPublishers.ofString ("""
          {"model":"gpt-4","messages":[
               {"role":"user","content":"Provide a recipe that includes Cheese"}
          ]}
          """))
       .build();
var response = HttpClient.newHttpClient().send(request, HttpResponse.BodyHandlers.ofString ());
var content = new ObjectMapper().readTree(response.body())
       .path("choices").get(0).path("message").path("content").asText();
```

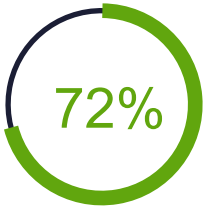
Example: Raw OpenAI API Call with Java using HttpClient

Why (Java) AI Frameworks?

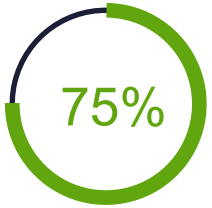
- Abstractions over working directly with LLM APIs
- Productivity features like structured output parsing, chat memory management, and vector store support
- Support for advanced AI patterns like RAG (Retrieval-Augmented Generation), Tool Calling, and AI Agents
- Stay within familiar programming languages and frameworks, like Java and Spring, rather than switching to, for example, Python and LangChain

spring® by VMware Tanzu

The most popular application development framework on the JVM



of Java developers utilize Spring Boot



like Spring because it's very stable, scalable, and secure

History of Spring

Created in 2003 as a lightweight alternative to address the complexity of the early J2EE specifications. Spring and Java/Jakarta EE are not in competition, they are complementary.

Why Spring?

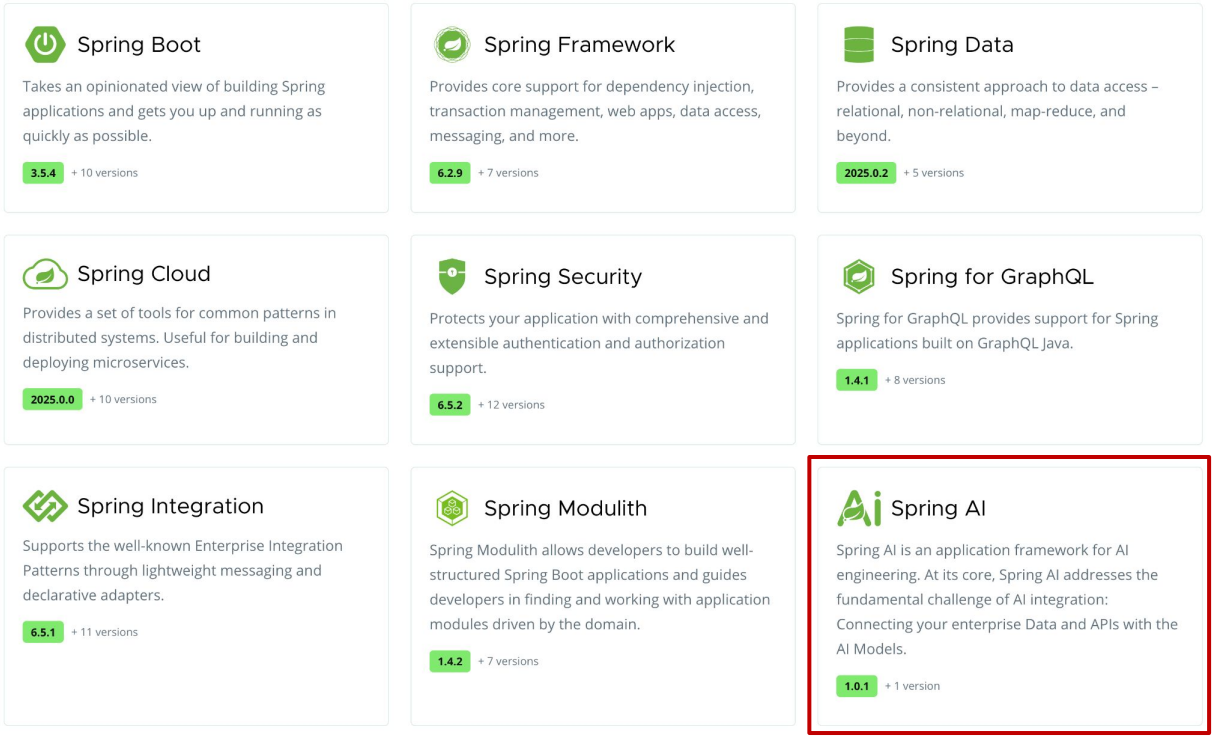
- Focused on speed, simplicity, and productivity
- Enterprise and production ready
- Extensive ecosystem
- Large, active developer community

Microsoft & vmware by Broadcom

Sources: https://www.jetbrains.com/lp/devecosystem-2023/java/
https://tanzu.vmware.com/content/ebooks/the-state-of-spring-2022

Official Spring Projects

spring.io/projects



Spring AI

Provides the key ingredients for creating (Generative) AI applications!

Aligns with Spring ecosystem design principles:

- Component abstractions and default implementations
- Portability across AI Provider APIs, Vector databases and more
- Auto Configuration and Starters with Spring Boot

Additional features

- Support for several techniques to adapt models to your needs like “Function Calling” or “Retrieval Augmented Generation”
- Multimodality
- Structured outputs (mapping of AI model output to POJOs)
- Observability
- AI model evaluation testing support
- Builds upon other projects in the Spring ecosystem

Demo

Recipe Finder Sample & Getting Started with Spring AI

Adapting Foundation Models

Popular techniques

Fine-Tuning: Further refining a model with specific data to improve its performance on a particular type of task ← *requires significant computational resources*

Prompt Engineering: Designing effective input prompts to guide a generative model's outputs (e.g. with Few-Shot Prompting, Chain-of-Thought Prompting, or In-Context Learning)

Tool Calling: Allows you to register your own functions to connect the model to the APIs of external systems

Demo

Tool Calling

Microsoft & vmware by Broadcom

Adapting Foundation Models

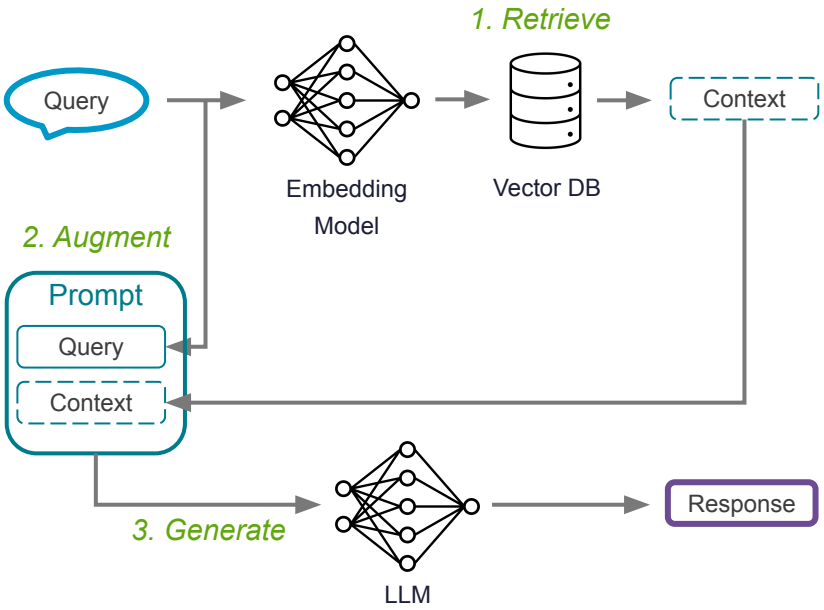
Popular techniques

Fine-Tuning: Further refining a model with specific data to improve its performance on a particular type of task ← *requires significant computational resources*

Prompt Engineering: Designing effective input prompts to guide a generative model's outputs (e.g. with Few-Shot Prompting, Chain-of-Thought Prompting, or In-Context Learning)

Tool Calling: Allows you to register your own functions to connect the model to the APIs of external systems

Retrieval-Augmented Generation (RAG) enhances the output of models by incorporating relevant external information from external data sources



Anatomy of a RAG System

Retrieval-Augmented Generation - Retrieval

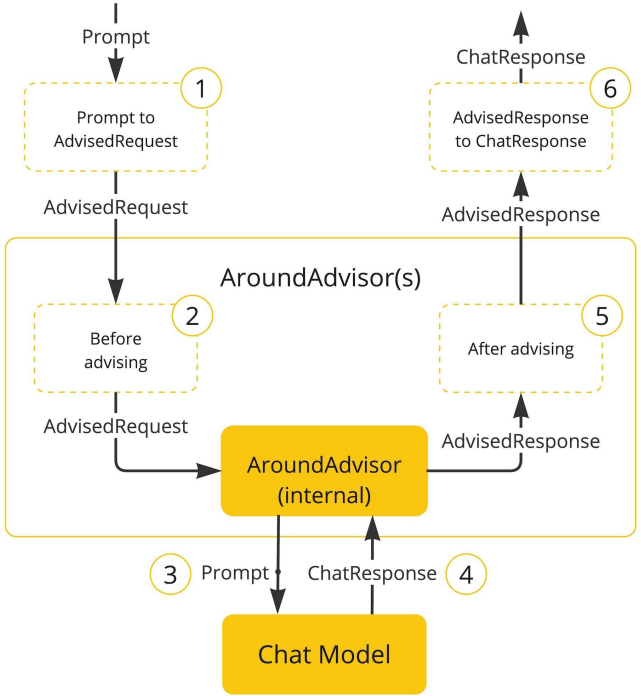
Advisors

`Advisors` allow you to intercept, modify, and enhance AI interactions — for example, by appending contextual data to the prompt

`QuestionAnswerAdvisor` implements naive RAG

`RetrievalAugmentationAdvisor` and additional APIs to implement advanced RAG

More built-in examples are `MessageChatMemoryAdvisor` and `SafeGuardAdvisor`



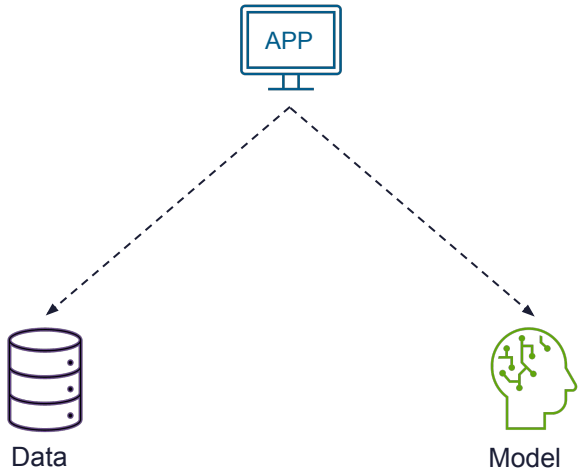
Demo

RAG

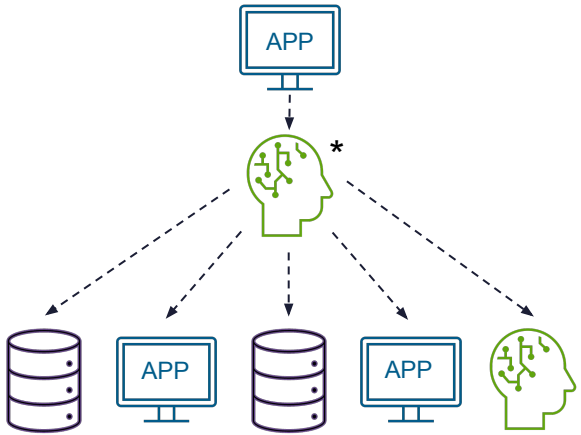
Microsoft & vmware by Broadcom

Agentic AI

Independent decision making and task execution, without constant human intervention



Intelligent Applications

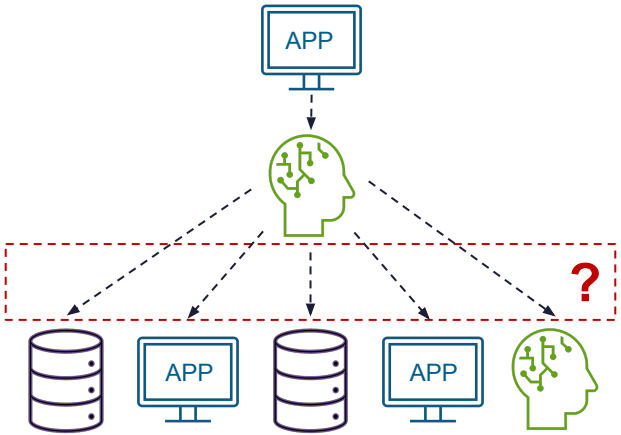


Agentic Applications

Microsoft & vmware by Broadcom

** Visualizes the autonomy and not component interactions*

Model Context Protocol (MCP)

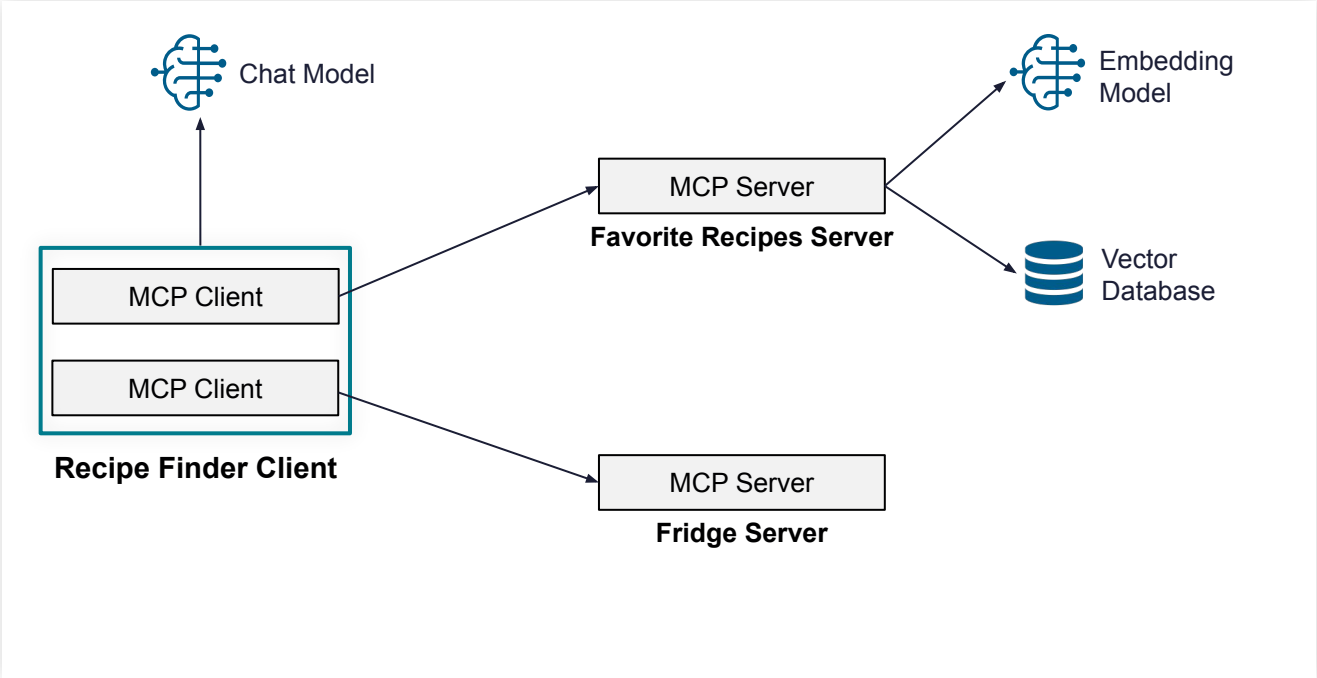


Agentic Applications

The Model Context Protocol provides a standardized way to connect AI models to different data sources and tools

A MCP Client is responsible for establishing and managing connections with MCP servers, which provide tools and resources

Recipe Finder MCP Architecture



Demo

MCP

Microsoft & vmware by Broadcom

Follow us
@sKriemhild
@salmto

Thank You

https://github.com/timosalm/spring-ai-recipe-finder