Comparing Agentic AI Frameworks

for Java

Sandra Ahlgrimm
Senior Developer Advocate, GitHub

Timo Salm
Master Solutions Architect, Broadcom

April 2026

GitHub & vmware by Broadcom

Who We Are

Sandra Ahlgrimm

Senior Developer Advocate
GitHub

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

Timo Salm

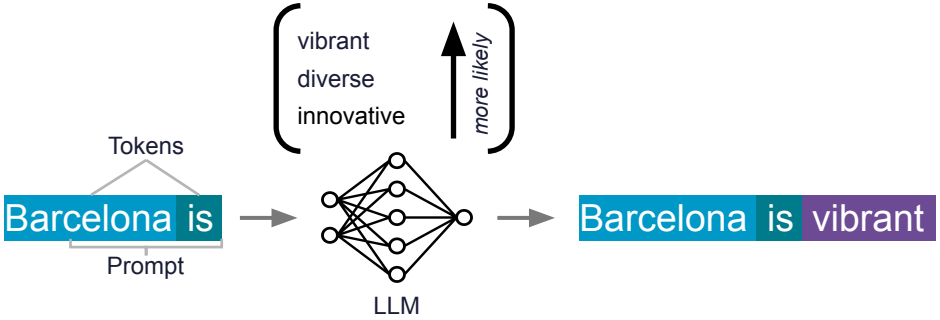
Master Solutions Architect
Broadcom

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

Glossary

A Prompt is the inputs given to a model to guide its response and behavior

Tokens are the text units a model processes, often subwords chosen to balance efficiency and meaning. They are the currency of AI.



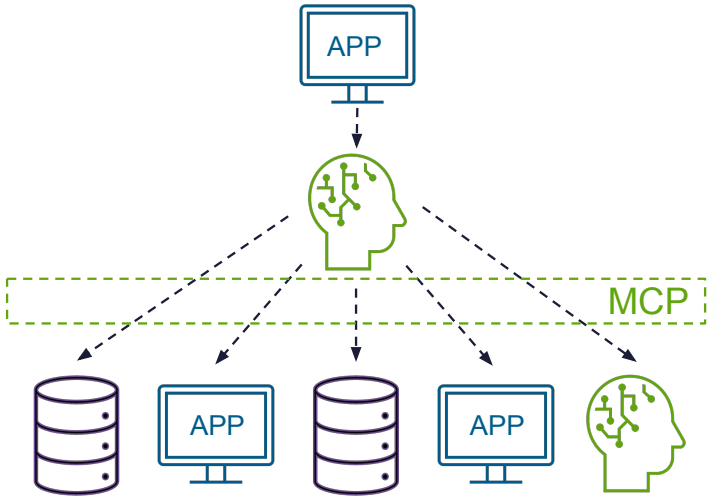
The context length is the maximum amount of information (tokens) a model can handle in one interaction

Retrieval-Augmented Generation (RAG) uses similarity search on a vector store to find the most relevant information and adds it to the context before the prompt is sent to the model, resulting in more accurate and up-to-date responses

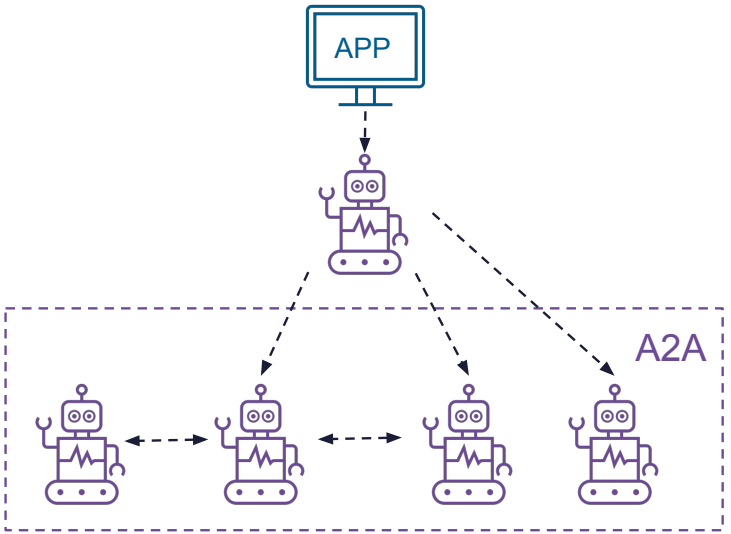
Tools are extra capabilities a model can invoke through Tool Calling, like searching the web, running code, or calling external services

Glossary

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



Agent-to-Agent (A2A) is a protocol that allows multiple AI agents to communicate, share work, and collaborate to complete complex tasks.



The Evolution of LLM Based Solutions

Standalone LLMs
Prompt Engineering, Fine-Tuning — Automate natural language tasks

Grounded Generation (RAG)
Vector DBs, Embeddings — Business data, less hallucination

Tool-Using Models
Tool Calling, Code Interpreter — External actions

Multimodal Models
Image, Vision, Speech — Richer inputs

Agentic AI Systems
Orchestration, Memory, Planning, MCP — End-to-end automation

Agentic AI

Agentic AI systems are designed to pursue complex goals with a high level of autonomy and predictability.

They are productivity enablers who can effectively incorporate humans in the loop via the use of multi-modality.

Key Characteristics

Autonomy: takes goal-directed actions with minimal human oversight

Reasoning: contextual decision-making, judgment calls & tradeoffs

Adaptable Planning: dynamically adjusts plans based on changing conditions to complete processes efficiently

Context Understanding: comprehends and follows natural language and other modalities

Action Enabled: empowered to take action via access to web services delivering skills

Two Approaches to Orchestrate LLMs and Tools

Workflows

LLMs and tools are orchestrated through predefined code paths

- ✓ Predictable and consistent output
- ✓ Easier to debug and test
- ✓ Better for well-defined, repeatable tasks
- ✓ Lower cost and latency
- ❌ Inflexible — can't adapt to unforeseen subtasks
- ❌ Requires upfront task decomposition

Pure Agents

The LLM dynamically directs its own processes and tool usage

- ✓ Flexible and adaptive for open-ended problems
- ✓ Handles tasks where steps can't be predicted in advance
- ✓ Scales well in trusted environments
- ❌ Higher cost and latency
- ❌ Risk of compounding errors
- ❌ Harder to debug and control
- ❌ Requires extensive testing and guardrails

Use workflows for predictability.

Use agents only when flexibility and model-driven decision-making are genuinely needed.

Workflow Patterns

for Agentic Systems

Prompt Chaining

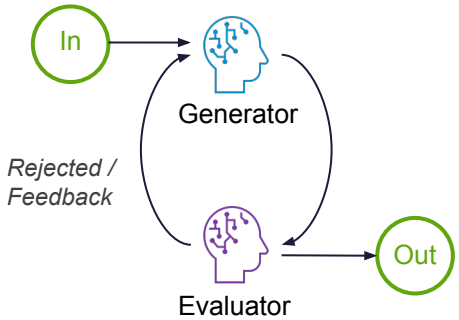
Decomposes tasks into sequential steps. Each LLM call processes the output of the previous one, with optional programmatic gates.

Parallelization

Runs LLM tasks simultaneously. Two variants: Sectioning (parallel subtasks) and Voting (multiple attempts for confidence).

Evaluator-Optimizer

One LLM generates a response while another evaluates and gives feedback in a loop. Continues until quality criteria are met.



Routing

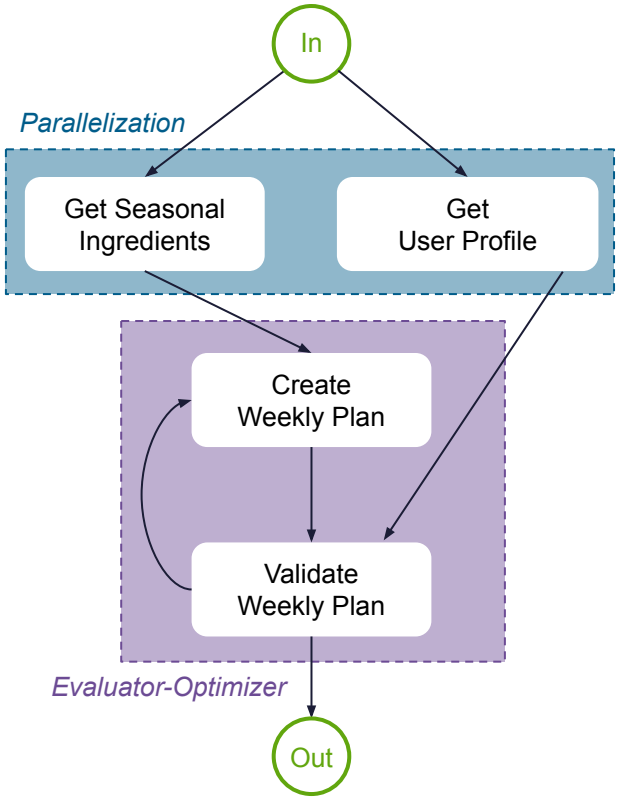
Classifies an input and directs it to a specialized follow-up task. Enables separation of concerns and optimized prompts per category.

Orchestrator-Workers

A central LLM dynamically breaks down tasks, delegates to worker LLMs, and synthesizes results. Subtasks aren't pre-defined.

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

Sample Application



AI Nutrition Planner *powered by OpenAI (Gpt-5-nano)*

Days & Meals

Monday All	Tuesday All	Wednesday All	Thursday All	Friday All	Saturday All	Sunday None
☑ Breakfast	☑ Breakfast	☑ Breakfast	☑ Breakfast	☐ Breakfast	☑ Breakfast	☑ Breakfast
☑ Lunch	☐ Lunch	☑ Lunch	☐ Lunch	☑ Lunch	☐ Lunch	☑ Lunch
☑ Dinner	☐ Dinner	☑ Dinner	☐ Dinner	☐ Dinner	☐ Dinner	☑ Dinner

Country (ISO code): DE

Additional instructions: Prefer quick recipes with preparation time under 30 minutes

Generate Plan

Your Weekly Plan

MONDAY

BREAKFAST

Wild Garlic Spinach Omelette 10 min

Ingredients

- 3 pieces Eggs
- 60 g Spinach
- 1/2 bunch Wild garlic
- 1 tbsp Chives
- 1 tsp Olive oil
- to taste Salt
- to taste Pepper

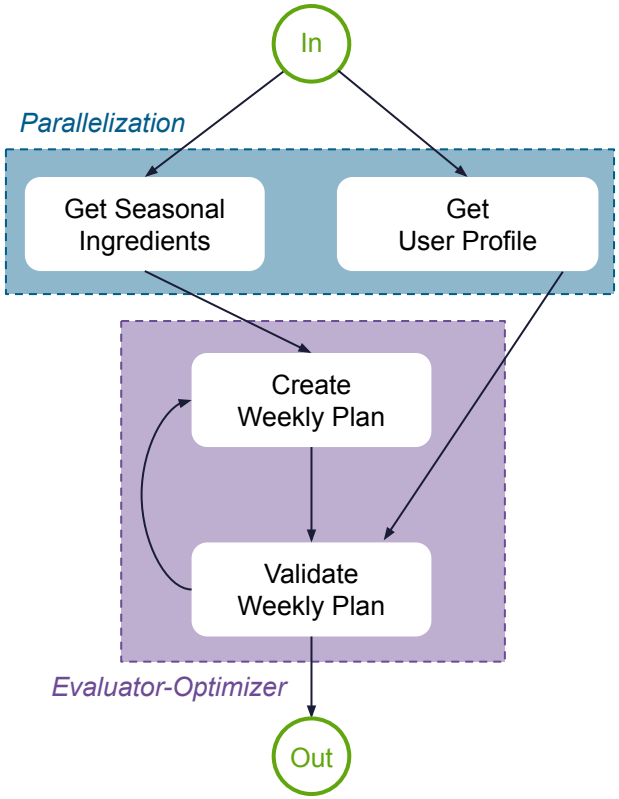
Instructions

Beat eggs with salt and pepper. Sauté wild garlic in oil, add spinach until just wilted. Pour in eggs and cook gently, folding until set. Finish with chopped chives.

NUTRITION	
Calories	270 kcal
Protein	18,0 g
Carbs	4,0 g
Fat	19,0 g
Sodium	280 mg

LUNCH

Sample Application



AI Nutrition Planner *powered by OpenAI (Gpt-5-nano)*

Days & Meals

Monday All	Tuesday All	Wednesday All	Thursday All	Friday All	Saturday All	Sunday None
☑ Breakfast	☑ Breakfast	☑ Breakfast	☑ Breakfast	☐ Breakfast	☑ Breakfast	☑ Breakfast
☑ Lunch	☐ Lunch	☑ Lunch	☐ Lunch	☑ Lunch	☐ Lunch	☑ Lunch
☑ Dinner	☐ Dinner	☑ Dinner	☐ Dinner	☐ Dinner	☐ Dinner	☑ Dinner

Country (ISO code): DE

Additional instructions: Prefer quick recipes with preparation time under 30 minutes

Generate Plan

Your Weekly Plan

MONDAY

BREAKFAST

Wild Garlic Spinach Omelette 10 min

Ingredients

- 3 pieces Eggs
- 60 g Spinach
- 1/2 bunch Wild garlic
- 1 tbsp Chives
- 1 tsp Olive oil
- to taste Salt
- to taste Pepper

Instructions

Beat eggs with salt and pepper. Sauté wild garlic in oil, add spinach until just wilted. Pour in eggs and cook gently, folding until set. Finish with chopped chives.

NUTRITION	
Calories	270 kcal
Protein	18,0 g
Carbs	4,0 g
Fat	19,0 g
Sodium	280 mg

LUNCH

Meet the Frameworks

Spring AI

Official Spring project since end of 2023

Aligns with Spring ecosystem design principles

Supports Spring Boot >= 3.4

Enterprise support available

LangChain4j

Initiated in early 2023 due to a lack of Java counterparts to Python libraries like LangChain, LlamaIndex

Framework-agnostic with integrations to Micronaut, Quarkus, Spring Boot, …

Embabel

Announced in 2025 to bring Agentic AI to enterprise JVM applications

Founded by Spring creator Rod Johnson

Built on Spring AI

GitHub & vmware by Broadcom

Koog

ADK for Java

…

Spring AI

Advisors API

Advisors are a key concept to implement Agentic AI capabilities

They allow you to intercept, modify, and enhance AI interactions

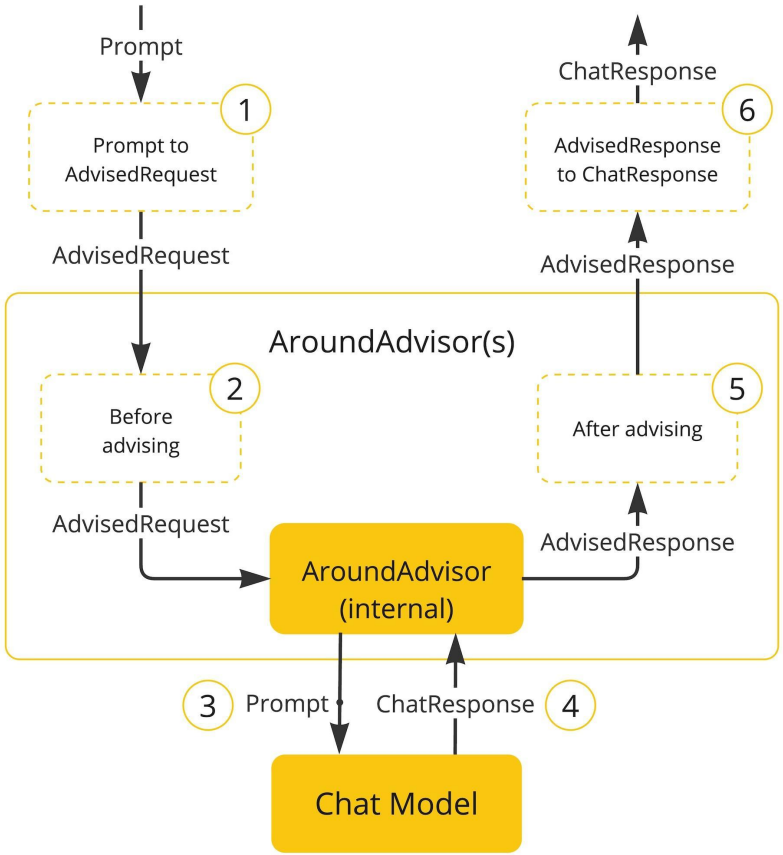
Examples: QuestionAnswerAdvisor, MessageChatMemoryAdvisor, SafeGuardAdvisor

A Recursive Advisor is a special type of Advisor

Allows to loop through downstream advisor chain multiple times

Enables controlled retries, validation, and multi-step tool execution

Examples: StructuredOutputValidationAdvisor, ToolCallAdvisor



Code Sample

String representation.
- String that represents location, as 2 double values split with coma.
framework.data.solr.core.geo.Point instance
if (locationString.contains(", ")) {
locationString = locationString.replaceAll(regex: ", ", replacement: ",");
String[] location = locationString.split(regex: ",");
double lat = Double.parseDouble(location[0]);
double lon = Double.parseDouble(location[1]);
return new Point(lat, lon);

LangChain4j

langchain4j-agentic module

The agentic module provides abstractions and utilities to define agentic workflows, manage tool usage, and maintain context across interactions with different LLM

Agents can be defined like AI services with the `@Agent` annotation. `outputKey` names the shared variable in the `AgenticScope` that stores the agent result

```
public interface NutritionPlanner {
    @UserMessage("...")
    @Agent(outputKey = "mealplan", description = "...")
    MealPlan createMealPlan(@V("ingredients") Ingredients ingred);
}
```

LangChain4j

langchain4j-agentic module

Instance and workflows can be build using the `AgenticServices.agentBuilder()` method

```
NutritionPlanner nutritionPlanner = AgenticServices
        .agentBuilder(NutritionPlanner.class)
        .chatModel(chatModel)
        .build();
```

```
NutritionPlanner nutritionPlanner = AgenticServices
        .sequenceBuilder(NutritionPlanner.class) // .loopBuilder .parallelBuilder .conditionalBuilder ...
        .subAgents(seasonalIngredientAgent, weeklyPlanCreator, validationLoop)
        .build();
```

LangChain4j

Code Sample

GitHub & vmware by Broadcom

Embabel

Agent Framework

Embabel goes beyond sequential execution with an AI planning step (A* Search)

Plans are dynamic, formulated at runtime, revised after every action (OODA* loop)

Embabel models agentic flows through Actions, Goals, Conditions, and a Domain Model and provides Spring-style annotations for it

```
@Agent(description = "...")
class NutritionPlanner {

 @Action(pre = {"spel:..."})
 Ingredients fetchSeasonalIngredients() {
 ...
 }

 @AchievesGoal(description = "Create meal plan")
 @Action
 MealPlan createMealPlan(Ingredients ingred) {
 ...
 }
```

GitHub & vmware by Broadcom

* Observe-Orient-Decide-Act loop

Embabel

Code Sample

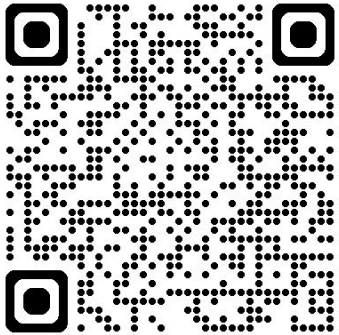
GitHub & vmware by Broadcom

Comparing Agentic AI Frameworks' Capabilities for Spring Devs

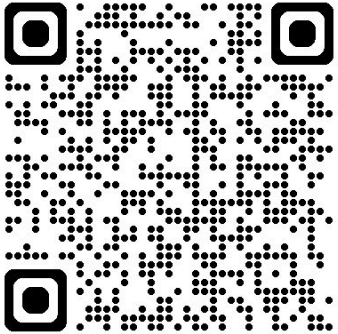
Aspect	Spring AI	LangChain4j	Embabel
Documentation			
Deterministic Workflow			
Pure Agents			
Subagents			
Advanced tool selection			
Human-in-the-loop			
AI skills			
Observability			
Testing			
MCP		No HTTP Server	
A2A			

Modest effort to implement, or already available via a community project

Related Previous Talks



Spring AI Intro
Slides Demo Video



Comparing Acentic AI Frameworks for Java
Slides Demo Video

Follow us
@sKriemhild
@salmto

Thank You

https://github.com/SandraAhlgrimm/ai-nutrition-planner

GitHub & vmware by Broadcom