Intelligent Applications

With Spring AI

Patrick Baumgartner
42talents GmbH, Zürich, Switzerland

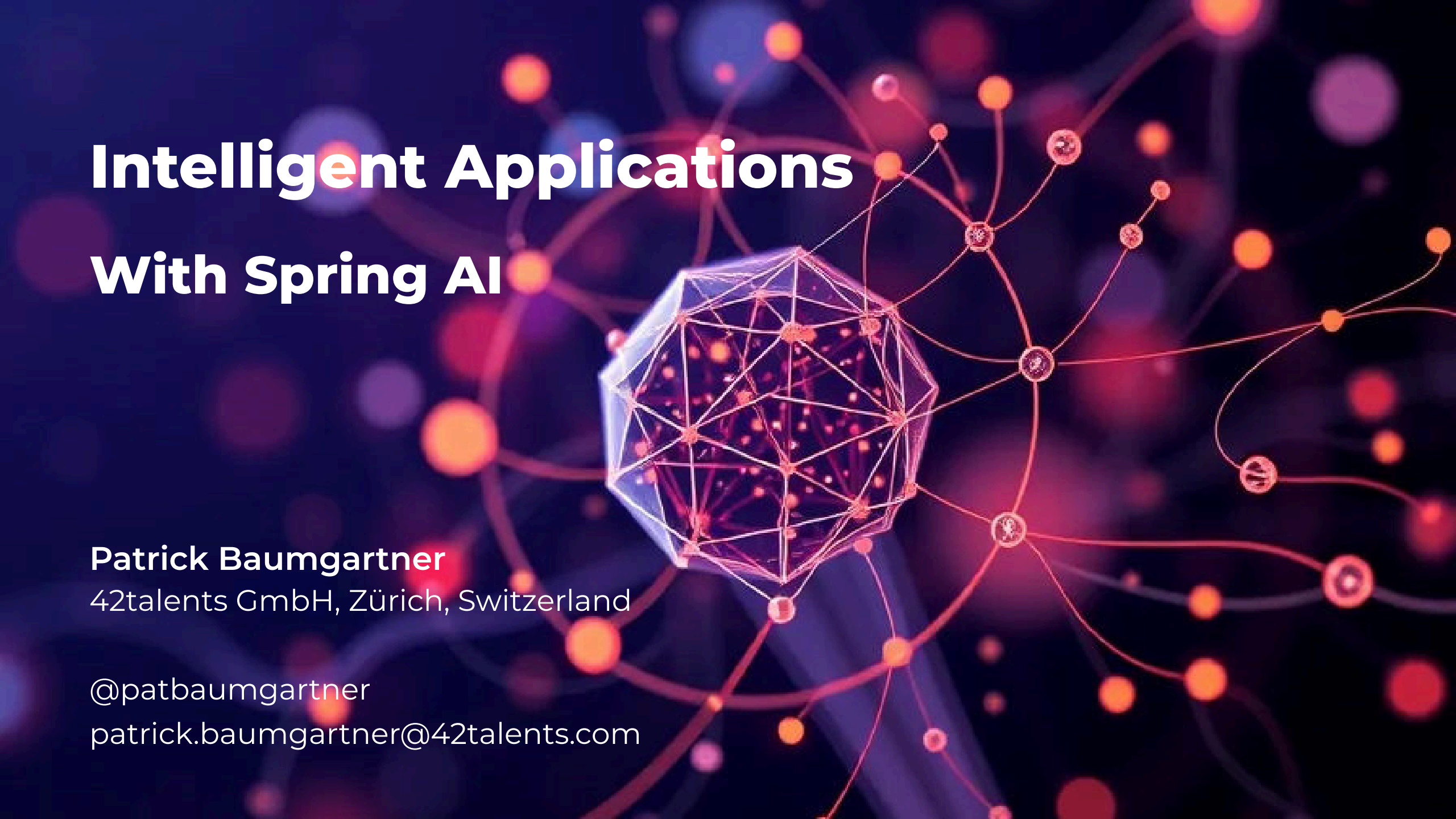
@patbaumgartner
patrick.baumgartner@42talents.com

Abstract

Intelligent Applications with Spring AI

The rise of AI has revolutionised our world and opened up countless new possibilities. While the range of AI solutions is still growing, we can now develop applications with advanced language capabilities much faster.

In this talk, we will look at different language models (LLMs) and explore their application in different scenarios such as chat interaction, image generation and audio transcription. Based on the Spring AI project, we will learn about different AI models and their functionalities and look at their practical integration in the enterprise environment.



Intelligent Applications With Spring AI

Patrick Baumgartner
42talents GmbH, Zürich, Switzerland

@patbaumgartner
patrick.baumgartner@42talents.com

Introduction

Patrick Baumgartner

Technical Agile Coach @ **42talents**

My focus is on the **development of software solutions *with* humans**.

Coaching, Architecture, Development, Reviews, and Training.

Lecturer @ **Zurich University of Applied Sciences ZHAW**

Co-Organizer of **Voxxed Days Zurich**, **JUG Switzerland**, Oracle ACE Pro Java ...

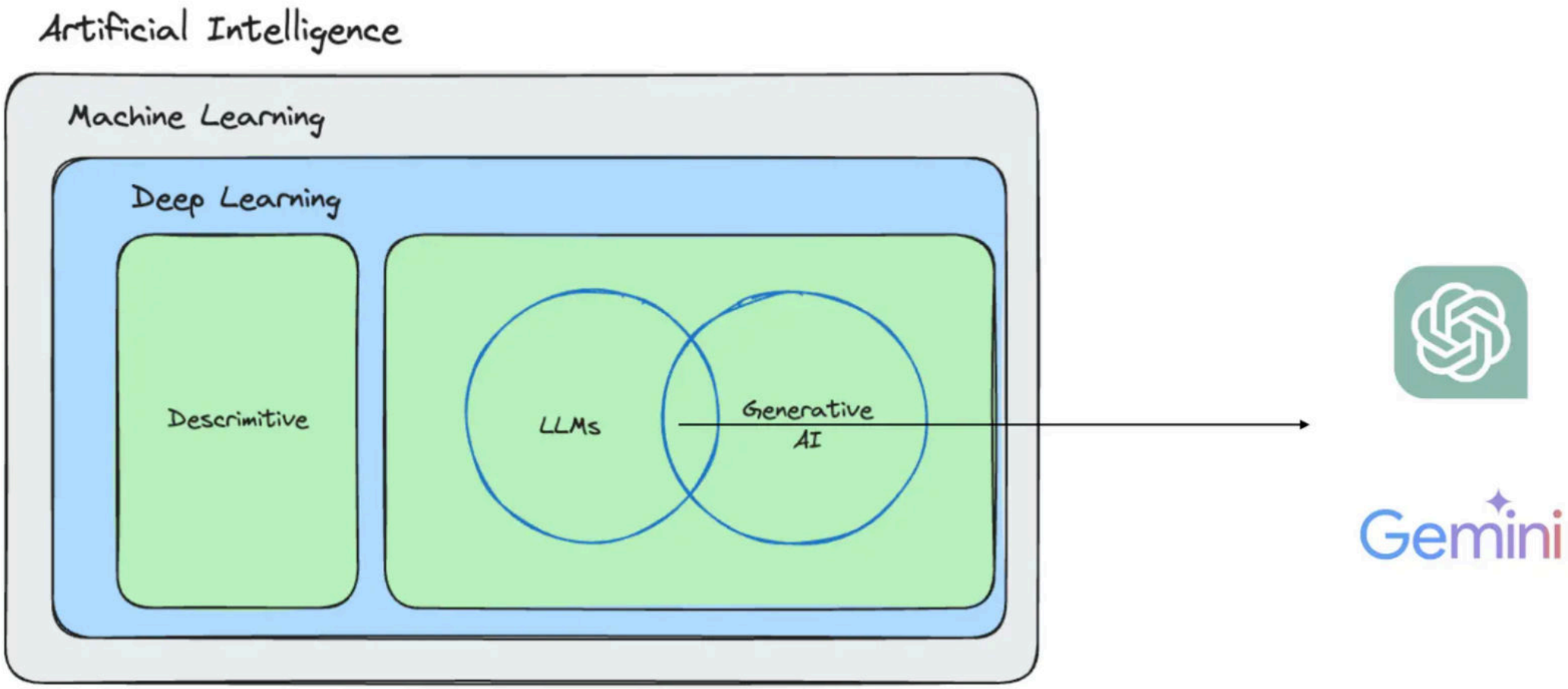
@patbaumgartner

⚠️ WARNING

This talk is focused exclusively on the technical integration of Spring AI with Large Language Models (LLMs). It will not address topics such as prompt engineering, fine-tuning, or model selection. Additionally, important considerations like sustainability, social impact, and the broader sense or nonsense of AI applications are beyond the scope of this presentation.

What is Artificial Intelligence (AI)?

What is Artificial Intelligence (AI)?



Examples ...

Examples ...

- Query data with natural language
  - *"Why did the user ask for their account to be closed?"*
    - Checks emails, complaints, logs, customer support tickets, social media etc.

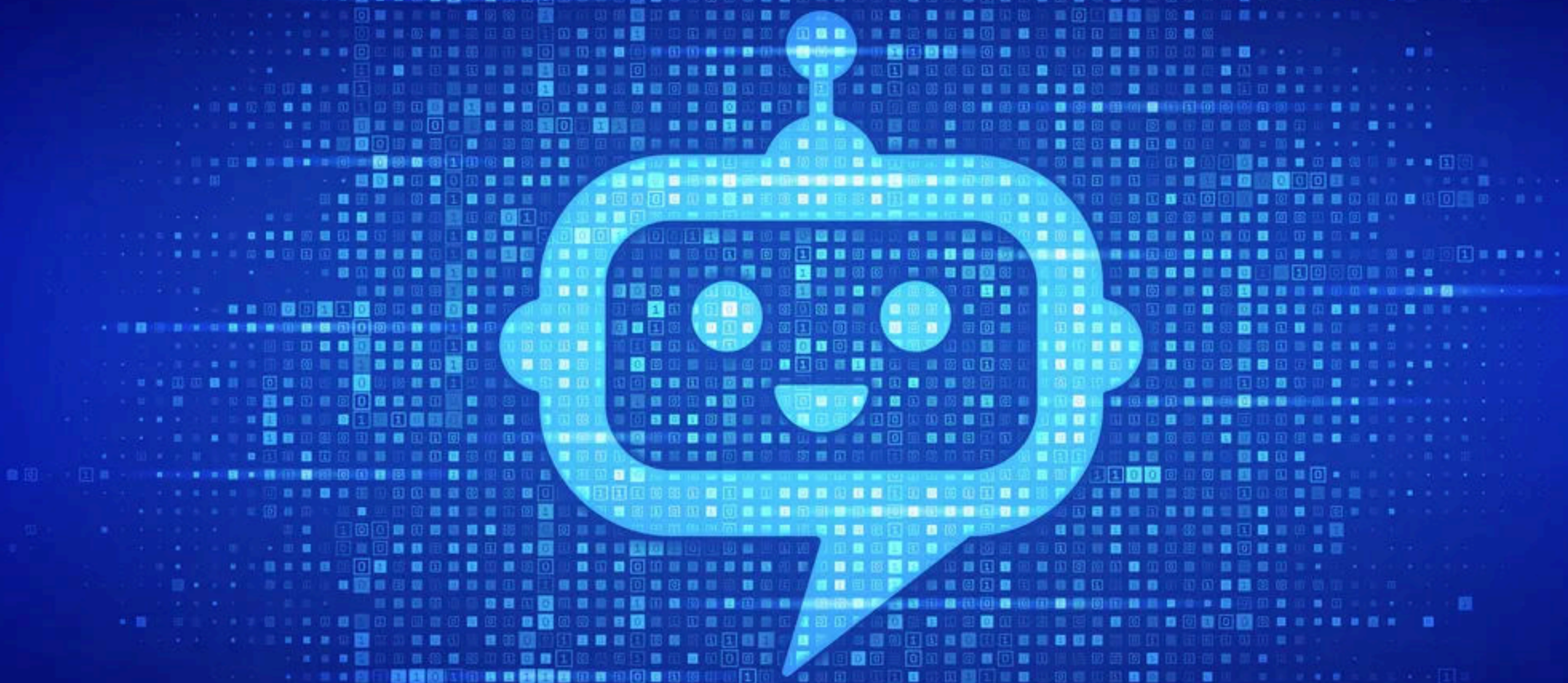
Examples ...

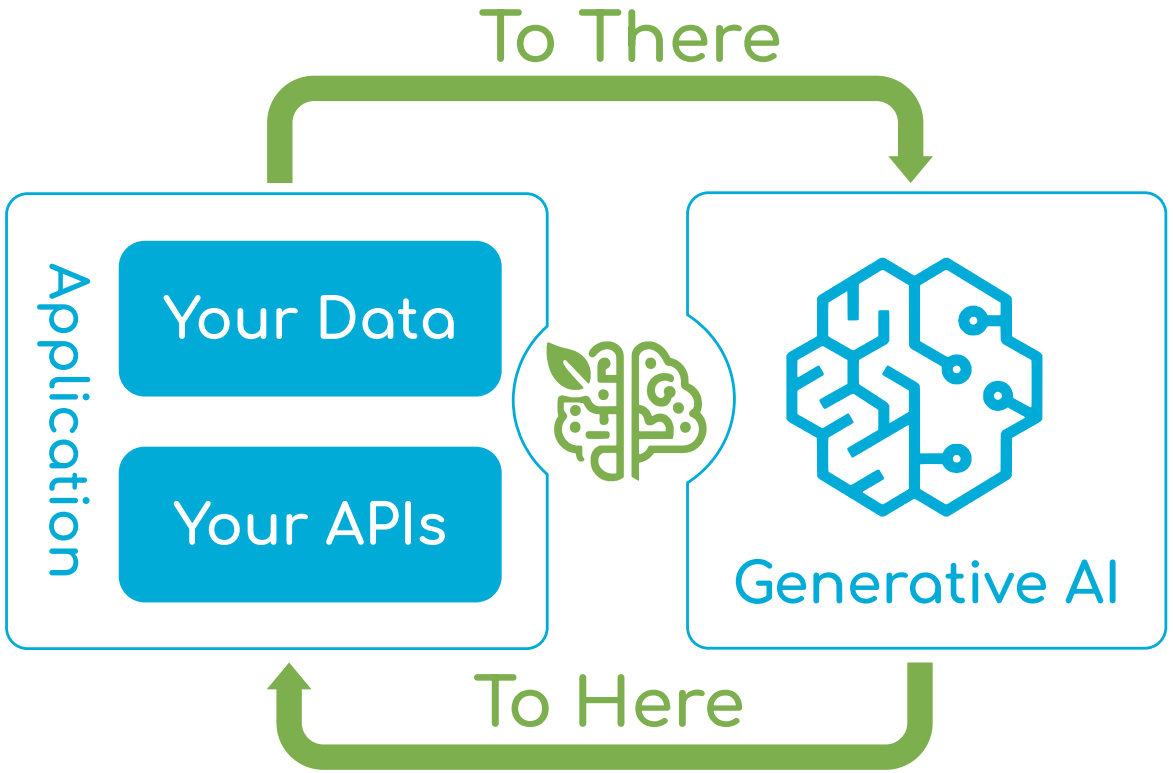
- Perform aggregation and transformation
  - Summarize text
  - Extract action items
  - Censor/identify sensitive data or abuse
  - Translate to different languages

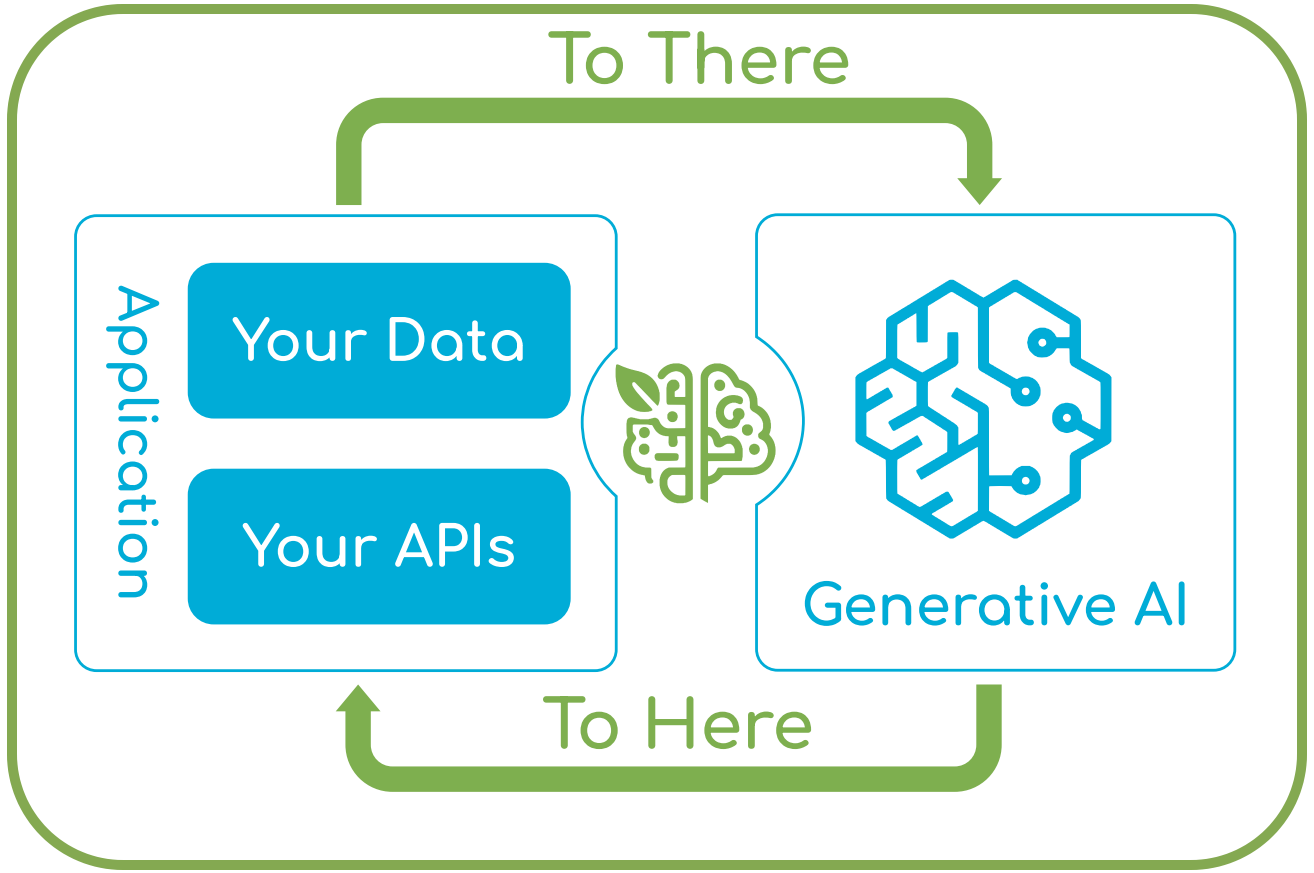
Examples ...

- Manipulate data
  - *"Reactivate customer account and notify them by email."*
- Define and execute business rules
  - *"If customer has not logged in for 6 months, send a reminder."*
- Generate code
  - *"Create a CSV exporter for all orders and write tests for it."*
- ...

Not all prompts are/need user input!




To There
Application
Your Data
Your APIs
Generative AI
To Here

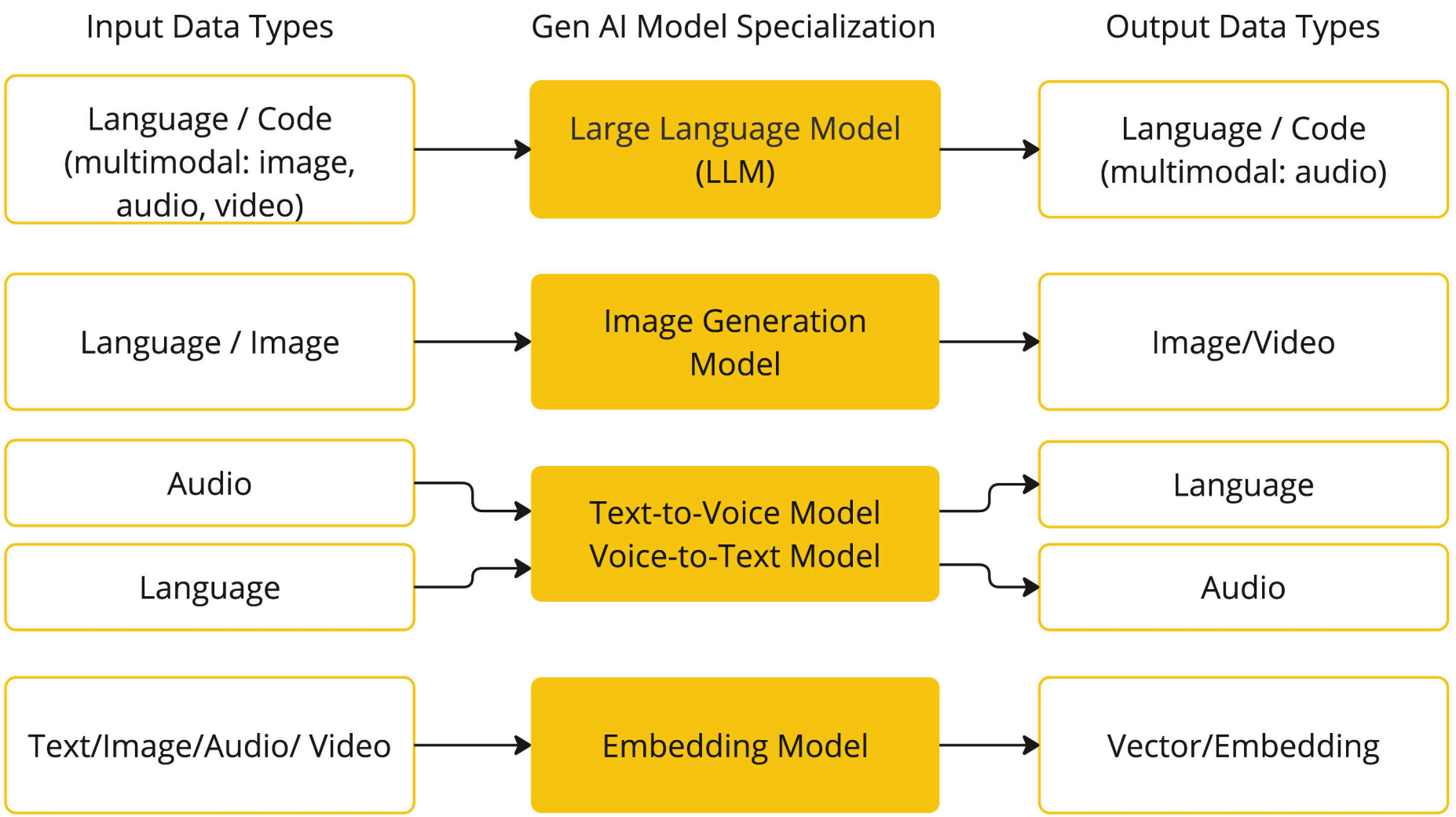


This is an integration problem!

REST APIs Make AI Accessible

```
#!/usr/bin/env bash

echo "Calling OpenAI API ..."

PROMPT="Share a helpful but lesser-known tip for writing better Java code."

curl https://api.openai.com/v1/chat/completions \
    -H "Content-Type: application/json" \
    -H "Authorization: Bearer $OPENAI_API_KEY" \
    --data \
"{
  \"model\": \"gpt-3.5-turbo\",
  \"messages\": [
    {
      \"role\": \"system\",
      \"content\": \"You're an experienced software engineer who gives practical advice to Java developers.\"
    },
    {
      \"role\": \"user\",
      \"content\": \"$PROMPT\"
    }
  ]
}"
```

AI Models



See: Spring AI: Models

https://olshansky.substack.com/p/vibe-checks-are-all-you-need

LLM Leaderboard - Comparison of GPT-4o, Llama 3, Mistral, Gemini and over 30 models

Comparison and ranking the performance of over 30 AI models (LLMs) across key metrics including quality, price, performance and speed (output speed - tokens per second & latency - TTFT), context window & others. For more details including relating to our methodology, see our FAQs.

For comparison of API Providers hosting the models see LLM API Providers Leaderboard

HIGHLIGHTS

Intelligence: o3-mini (high) and o3-mini are the highest quality models, followed by o1 & DeepSeek R1.

Output Speed (tokens/s): DeepSeek R1 Distill Qwen 1.5B (383 t/s) and Nova Micro (329 t/s) are the fastest models, followed by Nova Lite & Gemini 1.5 Flash (May).

Latency (seconds): Aya Expanse 8B (0.11s) and Command-R (Mar '24) (0.15s) are the lowest latency models, followed by Aya Expanse 32B & Command-R.

Price ($ per M tokens): Qwen2.5 Coder 7B ($0.03) and Llama 3.2 1B ($0.04) are the cheapest models, followed by Ministral 3B & DeepSeek R1 Distill Llama 8B.

Context Window: MiniMax-Text-01 (4m) and Gemini 2.0 Pro Experimental (2m) are the largest context window models, followed by Gemini 1.5 Pro (Sep) & Gemini 1.5 Pro (May).

PROMPT OPTIONS

Filter, e.g. GPT, Meta

EXPAND COLUMNS

	FEATURES		INTELLIGENCE	PRICE	OUTPUT TOKENS/S	LATENCY	
MODEL	CREATOR	CONTEXT WINDOW	ARTIFICIAL ANALYSIS INTELLIGENCE INDEX	BLENDED *USD/1M Tokens*	MEDIAN *Tokens/s*	MEDIAN *First Chunk (s)*	FURTHER ANALYSIS
o3-mini (high)	OpenAI	200k	66	$1.93	184.5	42.69	Model Providers
o3-mini	OpenAI	200k	63	$1.93	198.0	14.32	Model Providers
o1	OpenAI	200k	62	$26.25	108.9	27.04	Model Providers
DeepSeek R1	deepseek	128k	60	$0.96	26.2	4.25	Model Providers
QwQ-32B	Alibaba	131k	58	$0.47	95.8	0.54	Model Providers
Claude 3.7 Sonnet Thinking	ANTHROP\C	200k	57	$6.00	77.6	0.73	Model Providers
o1-mini	OpenAI	128k	54	$1.93	226.1	10.30	Model Providers
DeepSeek V3 (Mar' 25)	deepseek	128k	53	$0.48	31.8	4.02	Model Providers
Gemini 2.0 Flash Thinking exp. (Jan '25)	Google	1m	52	$0.00			Model Providers
DeepSeek R1 Distill Qwen 32B	deepseek	128k	52	$0.30	48.7	0.54	Model Providers
GPT-4o (March 2025)	OpenAI	128k	50	$7.50	209.2	0.35	Model Providers
Gemini 2.0 Pro Experimental	Google	2m	49	$0.00			Model Providers
DeepSeek R1 Distill Qwen 14B	deepseek	128k	49	$0.88	45.8	0.65	Model Providers
DeepSeek R1 Distill Llama 70B	deepseek	128k	48	$0.81	100.2	0.49	Model Providers

LLM Leaderboard

SEAL

Leaderboards

Frontier AI Evaluations

Humanity's Last Exam →

Frontier Multimodal Benchmark

Learn More

	Model	Accuracy	95% CI
1	Gemini 2.5 Pro Experimental (Mar...	18.81	+1.47 / -1.47
2	Claude 3.7 Sonnet Thinking (Febr...	8.93	+1.08 / -1.08
2	o1 (December 2024)	8.81	+1.07 / -1.07
2	Gemini 2.0 Flash Thinking (Janua...	7.22	+0.98 / -0.98
2	Gemini 2.0 Pro Experimental (Feb...	7.07	+0.97 / -0.97
4	GPT-4.5 Preview (February 2025)	6.41	+0.92 / -0.92
4	Llama 3.2 90B Vision Instruct	5.52	+0.86 / -0.86
6	Gemini-1.5-Pro-002	5.22	+0.84 / -0.84
6	Gemini 2.0 Flash Experimental (D...	5.19	+0.84 / -0.84
6	Gemini 2.0 Flash	5.07	+0.83 / -0.83

View Full Ranking →

EnigmaEval →

Puzzle Solving

Learn More

	Model	Accuracy (%)	95% CI
1	o1 Pro (March 2025)	6.14	+1.02 / -1.02
1	o1 (December 2024)	5.65	+0.52 / -0.52
3	Claude 3.7 Sonnet Thinking (Febr...	4.23	+0.45 / -0.45
3	Gemini 2.5 Pro Experimental (Mar...	4.14	+0.25 / -0.25
5	GPT-4.5 Preview (February 2025)	3.18	+0.28 / -0.28
6	Claude 3.7 Sonnet (February 2025)	2.26	+0.63 / -0.63
7	Gemini 2.0 Flash Thinking (Janua...	1.10	+0.17 / -0.17
7	Claude 3.5 Sonnet (October 2024)	0.91	+0.16 / -0.16
7	Pixtral Large (November 2024)	0.84	+0.19 / -0.19
7	Gemini 2.0 Pro Experimental (Feb...	0.69	+0.42 / -0.42

View Full Ranking →

MultiChallenge →

Realistic multi-turn conversation

Learn More

	Model	Score	95% CI
1	Gemini 2.5 Pro Experimental (Mar...	51.91	+0.99 / -0.99
1	Claude 3.7 Sonnet Thinking (Febr...	51.58	+1.98 / -1.98
1	o1 Pro (March 2025)	49.82	+1.36 / -1.36
3	o1 (December 2024)	44.93	+3.29 / -3.29
4	GPT-4.5 Preview (February 2025)	43.77	+1.60 / -1.60
4	Claude 3.5 Sonnet (October 2024)	43.20	+3.07 / -3.07

VISTA →

Visual Language Understanding

Learn More

	Model	Score	95% CI
1	Gemini 2.5 Pro Experimental (Mar...	54.65	+1.46 / -1.46
2	Claude 3.7 Sonnet Thinking (Febr...	48.23	+0.70 / -0.70
2	o1 Pro (March 2025)	47.32	+1.78 / -1.78
3	Gemini 2.0 Flash Thinking Experi...	45.50	+1.20 / -1.20
3	o1 (December 2024)	45.25	+0.40 / -0.40
4	Gemini 2.0 Pro Experimental (Feb...	43.25	+1.26 / -1.26

https://scale.com/leaderboard

Arena (battle) | Arena (side-by-side) | Direct Chat | Leaderboard | Prompt-to-Leaderboard | Arena Explorer | About Us

Chatbot Arena LLM Leaderboard: Community-driven Evaluation for Best LLM and AI chatbots

Discord | Twitter | 小红书 | Blog | GitHub | Paper | Dataset | Kaggle Competition

Chatbot Arena is an open platform for crowdsourced AI benchmarking, developed by researchers at UC Berkeley SkyLab and LMArena. With over 1,000,000 user votes, the platform ranks best LLM and AI chatbots using the Bradley-Terry model to generate live leaderboards. For technical details, check out our paper.

Chatbot Arena thrives on community engagement — cast your vote to help improve AI evaluation!

New Launch! Chat with online LLMs in the Search Arena tab!

Language | Overview | Price Analysis | WebDev Arena | Vision | Text-to-Image | Copilot Arena | Arena-Hard-Auto

Total #models: **220**. Total #votes: **2,816,680**. Last updated: 2025-03-25.

Code to recreate leaderboard tables and plots in this notebook. You can contribute your vote at lmarena.ai!

Category: Overall

Apply filter: Style Control | Show Deprecated

Overall Questions

#models: 220 (100%) #votes: 2,816,680 (100%)

Rank* (UB)	Rank (StyleCtrl)	Model	Arena Score	95% CI	Votes	Organization	License
1	1	Gemini-2.5-Pro-Exp-03-25	1443	+11/-8	3474	Google	Proprietary
2	2	ChatGPT-4o-latest (2025-03-26)	1408	+11/-12	2676	OpenAI	Proprietary
2	4	Grok-3-Preview-02-24	1404	+6/-6	10397	xAI	Proprietary
2	2	GPT-4.5-Preview	1398	+6/-7	10907	OpenAI	Proprietary
5	7	Gemini-2.0-Flash-Thinking-Exp-01-21	1381	+4/-5	22987	Google	Proprietary
5	4	Gemini-2.0-Pro-Exp-02-05	1380	+5/-4	20289	Google	Proprietary
7	5	DeepSeek-R1	1360	+5/-4	13074	DeepSeek	MIT
7	12	Gemini-2.0-Flash-001	1355	+6/-4	18650	Google	Proprietary
7	4	o1-2024-12-17	1351	+5/-4	25363	OpenAI	Proprietary
10	12	Qwen2.5-Max	1340	+5/-5	17452	Alibaba	Proprietary
10	12	Gemma-3-27B-it	1339	+7/-5	7238	Google	Gemma
10	9	o1-preview	1335	+4/-3	33188	OpenAI	Proprietary

Advanced AI Concepts

Integrating Custom Data & APIs

How can you equip the AI model with information on which it has not been trained? GPT-4o only has knowledge up until November 2024*, limiting its ability to answer questions about more recent data.

See more on OpenAI's Pricing Docs.

Bring Your Own Data

Limitations of AI Models

AI models:

- Are trained on public knowledge up to a certain date
- They lack access to your private or corporate data
- Don't have access to realtime data
- Can't learn from your data

Bring Your Own Data

Ways of using your data in AI applications:

- Train new models (❌)
- Fine tune existing models (❌)
- "Stuff the prompt" (✅)
- Retrieval Augmented Generation (✅)
- Tool calling (✅)

Java and AI

Java and AI

- Why Java & AI?
  - AI becomes ubiquitous across the IT landscape
  - Java is the language of enterprise
  - Creating Java AI apps is a new requirement
- **Spring AI**, **LangChain4j**, Semantic Kernel
  - Simplify AI integration
  - Provide a unified API
  - Support multiple AI providers

Spring AI vs LangChain4j

r/SpringBoot • 3 hr. ago
Different_Rafal

Spring AI vs LangChain4J

Which of these projects do you think will dominate Spring Boot AI applications in the future?

Spring AI seems to be the obvious choice here since it is from Spring Projects. On the other hand, at conferences I saw that more companies use LangChain4J than Spring AI.

I tested both of these libraries and for my requirements they provided similar functionality. Spring AI has the advantage of being more in the spirit of Spring Boot, as are my applications.

Langchain4J seems more developed now, but I expect Spring AI to catch up. The only question is whether, if most of companies do decide to go with LangChain4J, Spring AI won't be phased out in a few years as unpopular.

What experience do you have with your companies and the materials you see? Do you foresee Spring AI dominance or not? I am wondering what would be a better choice to go with.

1 2 Share

WuhmTux • 47m ago

Does it really matter? Both are just API wrappers. The choice of LLM is more important than the API wrapper.

Vote Reply Award Share ...

See Spring AI on GitHub or on spring.io.

What is Spring AI?

- **AI for Java developers!**
- Simplifies interactions with LLMs with a Spring abstraction
  - Change model by changing one line of configuration!
- Simplifies interactions with Vector databases
- Observability
- ChatMemory
- Tool Calling
- And more ...

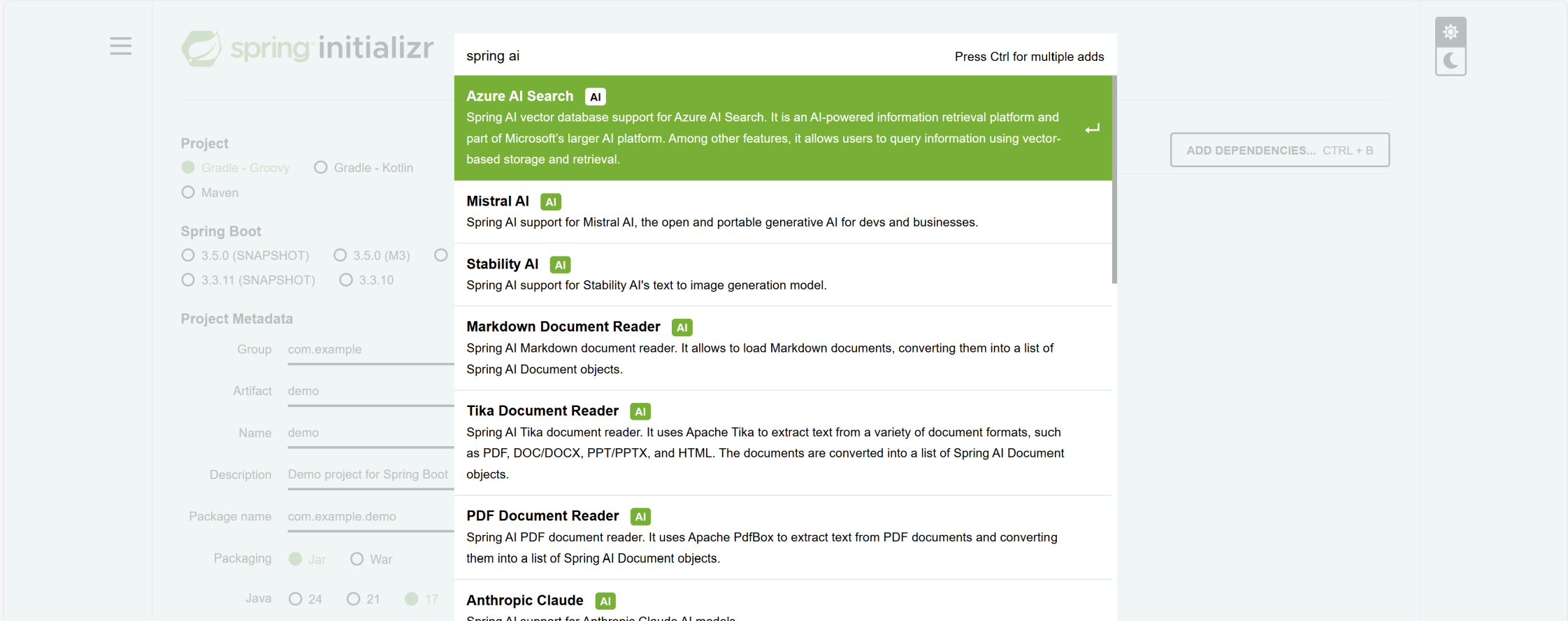
Spring AI Ecosystem

- Created in 2023 by Mark Pollack and Christian Tzolov
- Inspired by LangChain and LlamaIndex
- Current version: Spring AI 1.0.0-M8 / 1.0.0-SNAPSHOT
  - Not yet in a final release version
- Currently based on Spring Framework 6.2 and Spring Boot 3.4

Getting Started

Spring Initializr

Head on over to start.spring.io and select the AI Models and Vector Stores that you want to use in your new applications.



Milestone Repository

Because Spring AI is still in Milestone, you need to add the Spring Milestones repository to your `pom.xml` file. From M5 onwards, they are also in Maven Central.

```
<repositories>
  <repository>
    <id>spring-milestones</id>
    <name>Spring Milestones</name>
    <url>https://repo.spring.io/milestone</url>
    <snapshots>
      <enabled>false</enabled>
    </snapshots>
  </repository>
</repositories>
```

Spring AI BOM

Add the Spring AI BOM to your `pom.xml` file and reference the version in the properties section.

```
<properties>
    <spring-ai.version>1.0.0-M8</spring-ai.version>
</properties>

<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.springframework.ai</groupId>
      <artifactId>spring-ai-bom</artifactId>
      <version>${spring-ai.version}</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
```

Spring Boot Starter

Add the specific Spring AI dependency to your `pom.xml` file.

```
<dependency>
    <groupId>org.springframework.ai</groupId>
    <artifactId>spring-ai-starter-model-openai</artifactId>
</dependency>
```

Then, you can configure model parameters in the `application.properties` file as follows:

```
spring.ai.openai.api-key=${OPENAI_API_KEY}
spring.ai.openai.chat.options.model=gpt-4o-mini
...
```

Chat Client

Prompt Templates

Using the StringTemplate Engine

A prompt template allows you to inject parameters into pre-defined text.

```
String message = """
    List 10 of the most popular YouTubers in {genre}, along with their current subscriber numbers.
    If you don't know the answer, just say "I don't know".
    """;
PromptTemplate promptTemplate = new PromptTemplate(message);
Prompt prompt = promptTemplate.create(Map.of("genre", genre));
```

```
String answer = chatClient.prompt(prompt).call().content();
```

```
ChatResponse response = chatClient.prompt(prompt).call().chatResponse();
String answer = response.getResult().getOutput().getContent();
```

See also: https://www.stringtemplate.org/

Chat Client

```
ChatClient chatClient = chatClientBuilder
  .defaultSystem("You're an experienced software engineer who gives practical advice to Java developers.")
  .build();
```

```
String response = chatClient.prompt()
  .user("Share a helpful but lesser-known tip for writing better Java code.")
  .call()
  .content();
```

```
PromptResponsePair pair = chatClient.prompt()
  .system("You're an experienced software engineer who gives practical advice to Java developers.")
  .user("Share a helpful but lesser-known tip for writing better Java code.")
  .call()
  .entity(PromptResponsePair.class);
```

Request / Response

Based on OpenAI chat model:

- Request
  - model
  - list of messages, each with a role and content
- Response
  - model
  - list of choices, each with a message
  - finish reason
  - usage, with the number of tokens used
  - ...

Working with Images

Working with Images

```
@Value("classpath:images/mainz-weather.png")
private Resource imageResourceWeather;
```

```
chatClient.prompt()
  .user(userSpec -> userSpec
    .text("What will be the weather like on Thuesday?")
    .media(MimeTypeUtils.IMAGE_PNG, this.imageResourceWeather))
  .call().content();
```

Working with Audio

Working with Audio / Transcription Model

```
transcriptionModel.call(new AudioTranscriptionPrompt(audioFile)).getResult().getOutput();
```

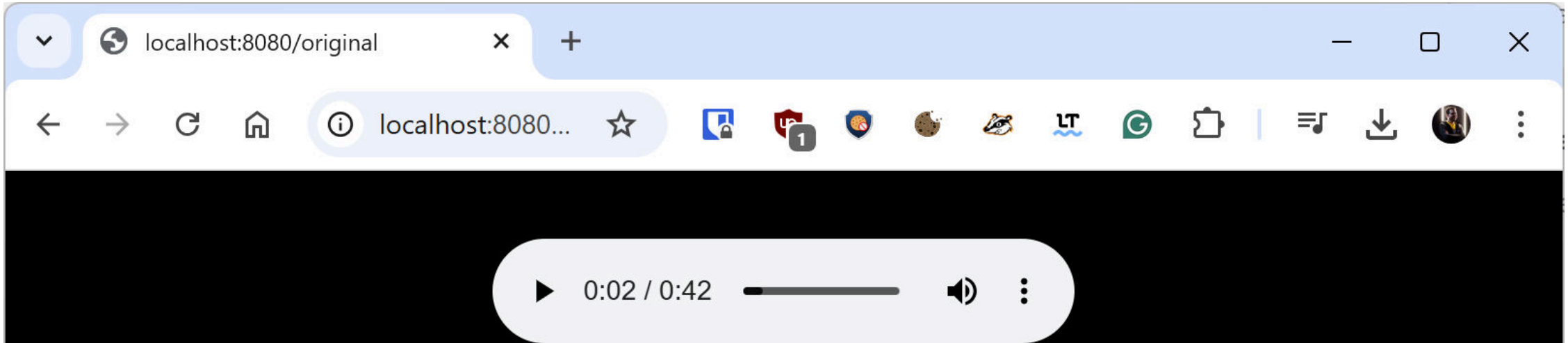
```
var transcriptionResponse = transcriptionModel.call(new AudioTranscriptionPrompt(audioFile,
  OpenAiAudioTranscriptionOptions.builder()
    .language("en")
    .prompt("Transcribe the audio file about Philipp Maloney and the missing bag.")
    .temperature(0f)
    .responseFormat(OpenAiAudioApi.TranscriptResponseFormat.VTT)
    .build()));
return transcriptionResponse.getResult().getOutput();
```

```
WEBVTT 00:00:01.000 --> 00:00:04.000 Philipp Maloney's hair-raising cases
00:00:05.000 --> 00:00:08.000 The hair-raising cases of Philipp Maloney.
00:00:09.000 --> 00:00:11.000 Today, the missing bag.
00:00:24.000 --> 00:00:28.000 The man sounded confused and excited at the same time on the phone.
00:00:28.000 --> 00:00:33.000 His voice sounded familiar, but I couldn't classify it.
00:00:33.000 --> 00:00:36.000 Only when I saw him did I recognize him.
00:00:36.000 --> 00:00:40.000 And then it was too late to run away.
00:00:40.000 --> 00:00:42.000 Do you remember me?
```

Working with Audio / Speech Model

```
speechModel.call(new SpeechPrompt(resource.getContentAsString(Charset.defaultCharset()))).getResult().getOutput();
```

```
var speechResponse = speechModel.call(new SpeechPrompt(resource.getContentAsString(Charset.defaultCharset()),
  OpenAiAudioSpeechOptions.builder()
    .model("tts-1")
    .voice(OpenAiAudioApi.SpeechRequest.Voice.SHIMMER)
    .responseFormat(OpenAiAudioApi.SpeechRequest.AudioResponseFormat.MP3)
    .speed(1.3f)
    .build()));
return speechResponse.getResult().getOutput();
```



Retrieval Augmented Generation (RAG)

Retrieval Augmented Generation

- Bring your own data to the prompt
- Give a lot of context to the prompt
  - Text content, vectors etc.

Prompt Data Takes Precedence (1)

```
You said:

How many sports were there in the Paris Olympics?
```

```
ChatGPT said:

The 2024 Summer Olympics in Paris featured 45 different sports, including 41 core Olympic sports and 4 additional sports.
```

Prompt Data Takes Precedence (2)

```
You said:

Here are the sports for the Paris Olympics: Archery, Athletics, Badminton, Basketball,
Basketball 3x3, Boxing, Canoe Slalom, Canoe Sprint, Road Cycling, Track Cycling,
Mountain Biking, and BMX.

How many sports were there?
```

```
ChatGPT said:

There are 12 sports listed for the Paris Olympics [...]

So, the total is 12.
```

Prompt Stuffing Structure

```
@Value("classpath:/olympics/context.st")
private Resource queryTemplate;
```

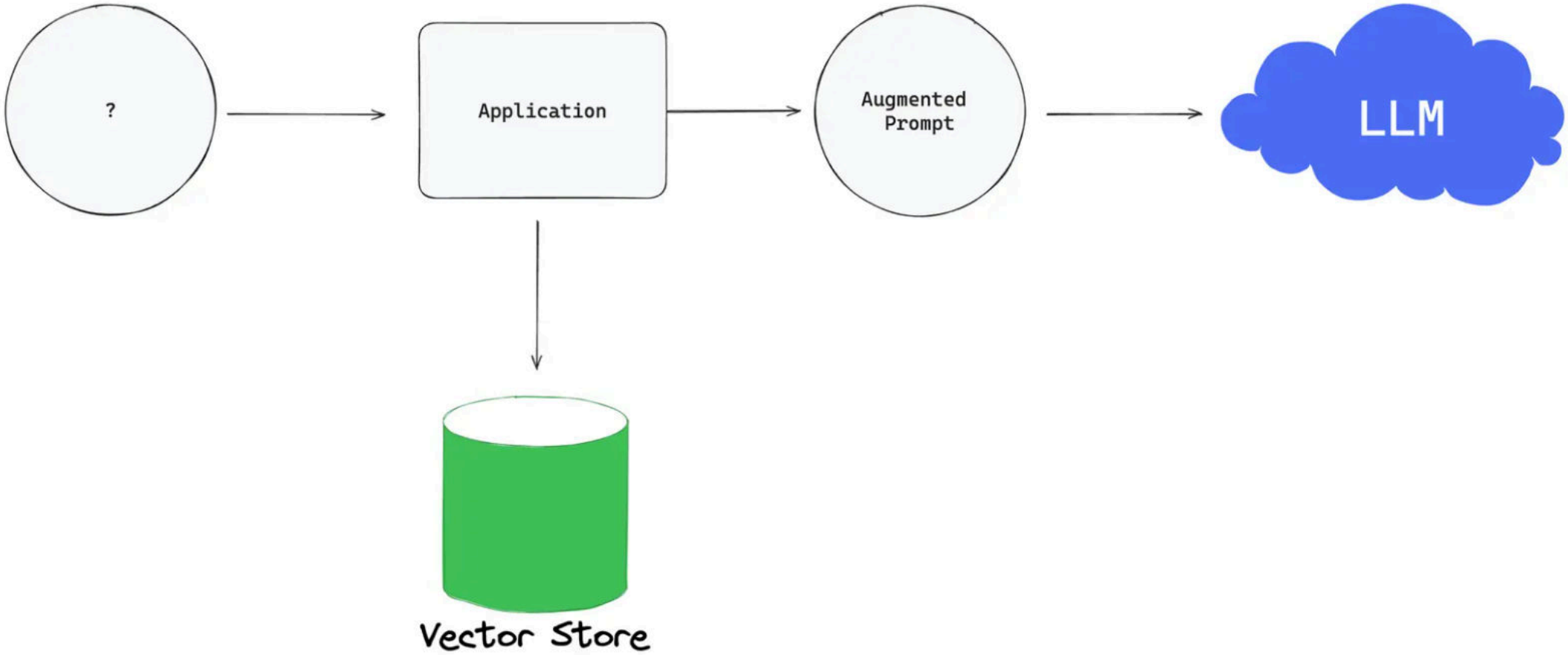
```
Use the following pieces of context to answer the question at the end.

{context}

Question: {question}
```

Using a Stuffed Prompt

```
chatClient.prompt()
  .user( userSpec -> userSpec.text(this.queryTemplate)
    .param( "context", "Archery, athletics, badminton, basketball , boxing")
    .param( "question","How many sports are being included in the 2024 Summer Olympics?")
     )
    .call().content();
```

```
Use the following pieces of context to answer the question at the end.

Archery, athletics, badminton, basketball , boxing

Question: How many sports are being included in the 2024 Summer Olympics
```

Application Architecture - RAG



Vector Stores

Vector Stores

Definition of a Vector

- A Vector is just a type of data
  - Typically an array of floating-point numbers
  - Commonly consists of decimal values
    - Spring AI: *text-embedding-ada-002* (1536) embedding model
    - OpenAI: *text-embedding-3-small* (1536) and *text-embedding-3-small* (3072) embedding model
  - Each value typically ranges between -1 and 1

```
CREATE TABLE paragraph (
  id SERIAL PRIMARY KEY,
  paragraph_text TEXT,
  vector VECTOR(1536)
);
```

Example: `{-0.345, 0.465, 0.856, ..., 0.1543}`

How are Vectors created?

- Vectors typically represent the output generated by machine learning models, encapsulating the learned features or embeddings of the input data.
- The example below is based on text AI models. Vectors may also be used with computer vision AI models or audio-based AI model

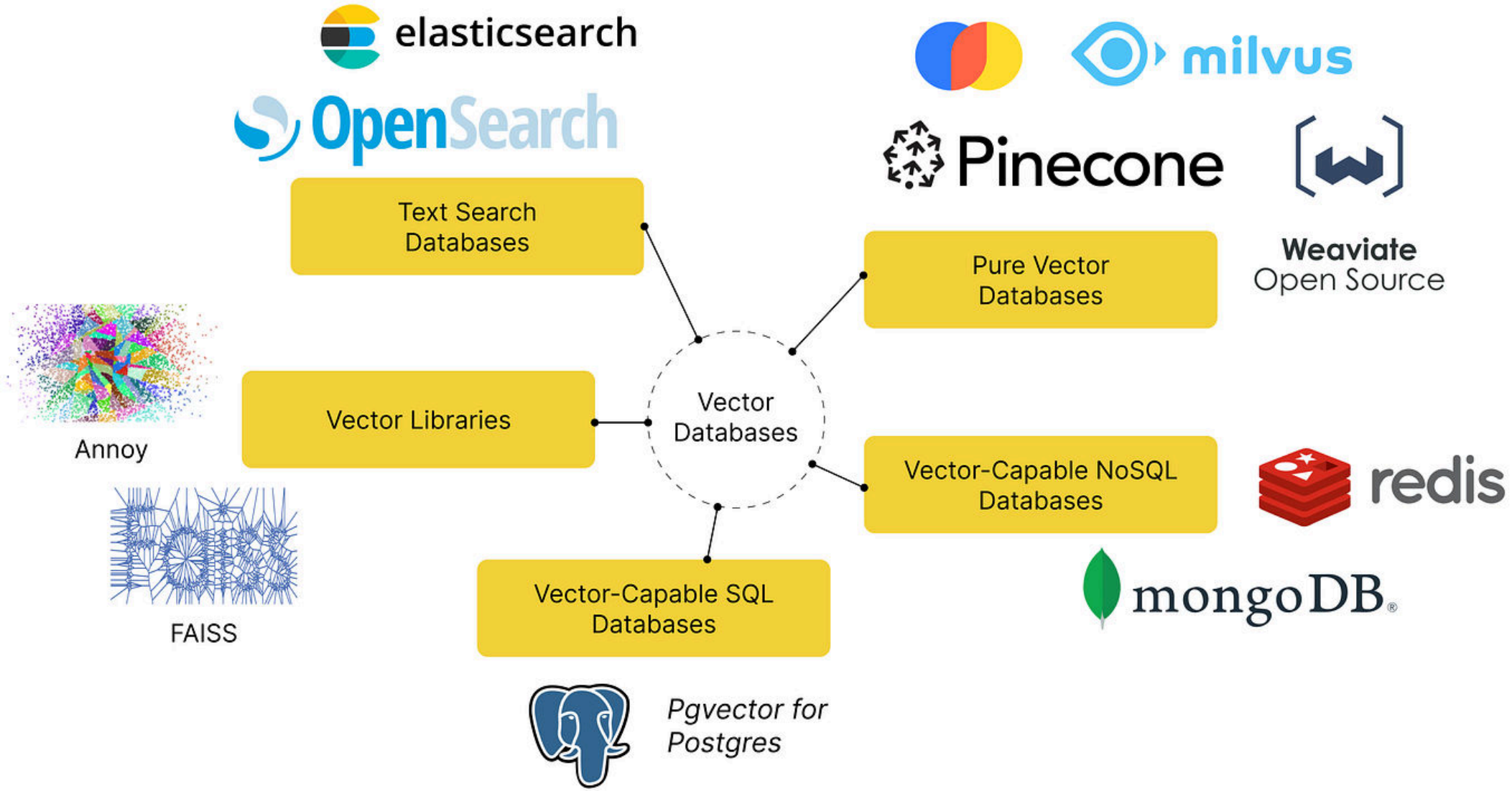
String	Vector
"My house is black"	{0.345, 0.465, 0.856, 0.1543}
"My garden is big"	{0.545, 0.665, 0.056, 0.3543}
"My dog is playful"	{0.645, 0.765, 0.156, 0.4543}

Similarity Search

- Selects the closest vectors to a given vector

  `{-0.436, 0.578, 0.935, 0.2193}`

Vector
{0.345, -0.465, 0.856, 0.1543}
{-0.445, 0.565, 0.956, 0.2543}
{0.545, 0.665, 0.056, 0.3543}
{0.645, 0.765, 0.156, -0.4543}
{0.745, 0.865, 0.256, 0.5543}
{0.845, 0.965, -0.356, 0.6543}


elasticsearch
OpenSearch
milvus
Pinecone
Weaviate
Open Source
Text Search Databases
Pure Vector Databases
Vector Databases
Vector Libraries
Vector-Capable NoSQL Databases
Annoy
FAISS
redis
mongoDB.
Vector-Capable SQL Databases
Pgvector for Postgres

Using a Vector Store

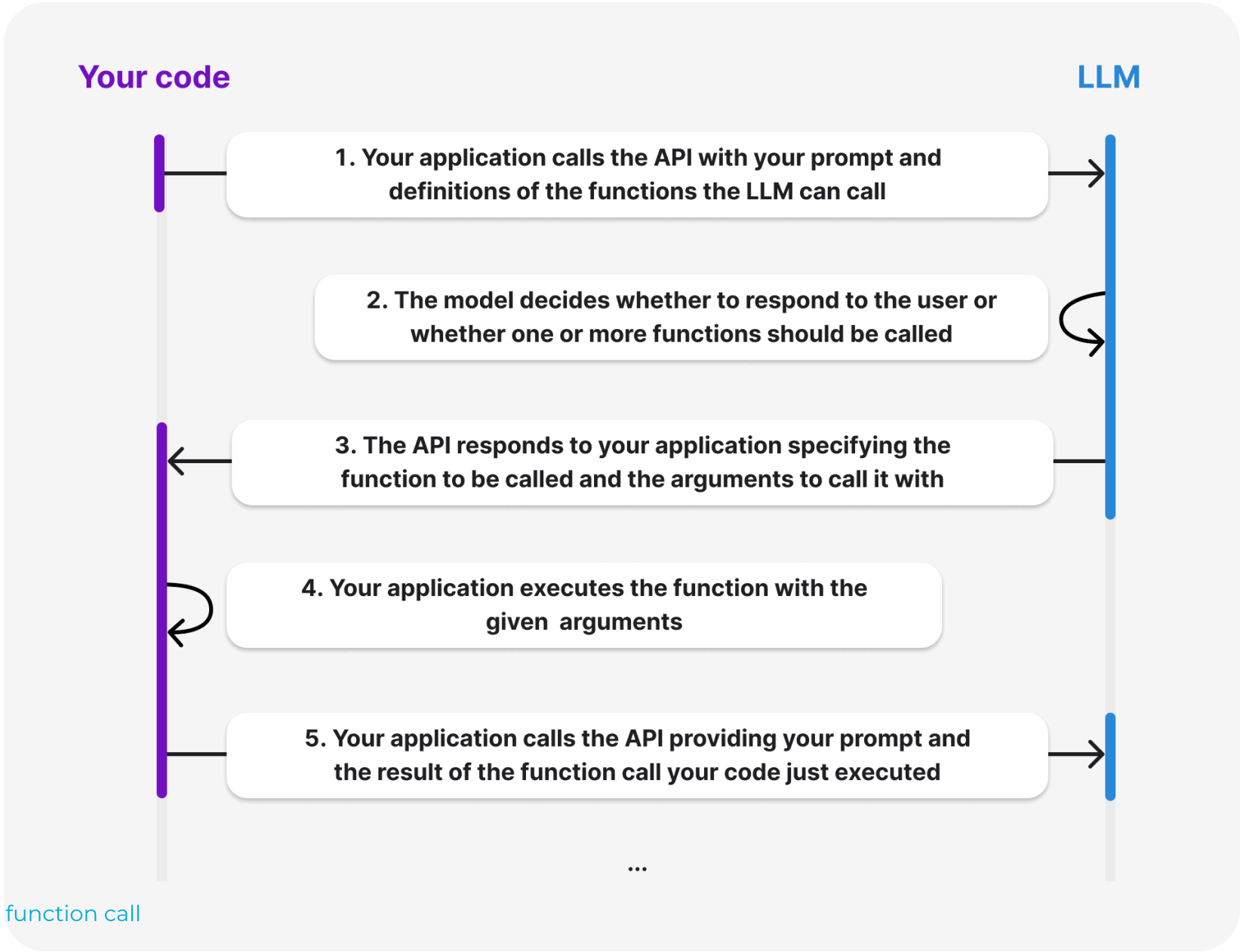
```
@Value("Artificial intelligence - Wikipedia.pdf")
private Resource pdf;
```

```
PagePdfDocumentReader reader = new PagePdfDocumentReader(pdf);
TokenTextSplitter splitter = new TokenTextSplitter();
List<Document> documents = splitter.apply(reader.get());
vectorStore.accept(documents);

String answer = chatClient.prompt()
  .system("""
      You are a virtual assistant and answers questions with the data provided.
      If you are not sure or don't know, honestly say you don't know.
      """)
  .user("Why is the definition of AI difficult?")
  .advisors(new QuestionAnswerAdvisor(vectorStore))
  .call()
  .content();

log.info("ChatGPT answered: {}", answer);
```

Tool calling

Registering Custom Functions

- The model generates a JSON object with arguments for the registered functions.
- The model does not call the function directly.
- Your code executes the function, and the result is returned to the model.
- ...



See The lifecycle of a function call

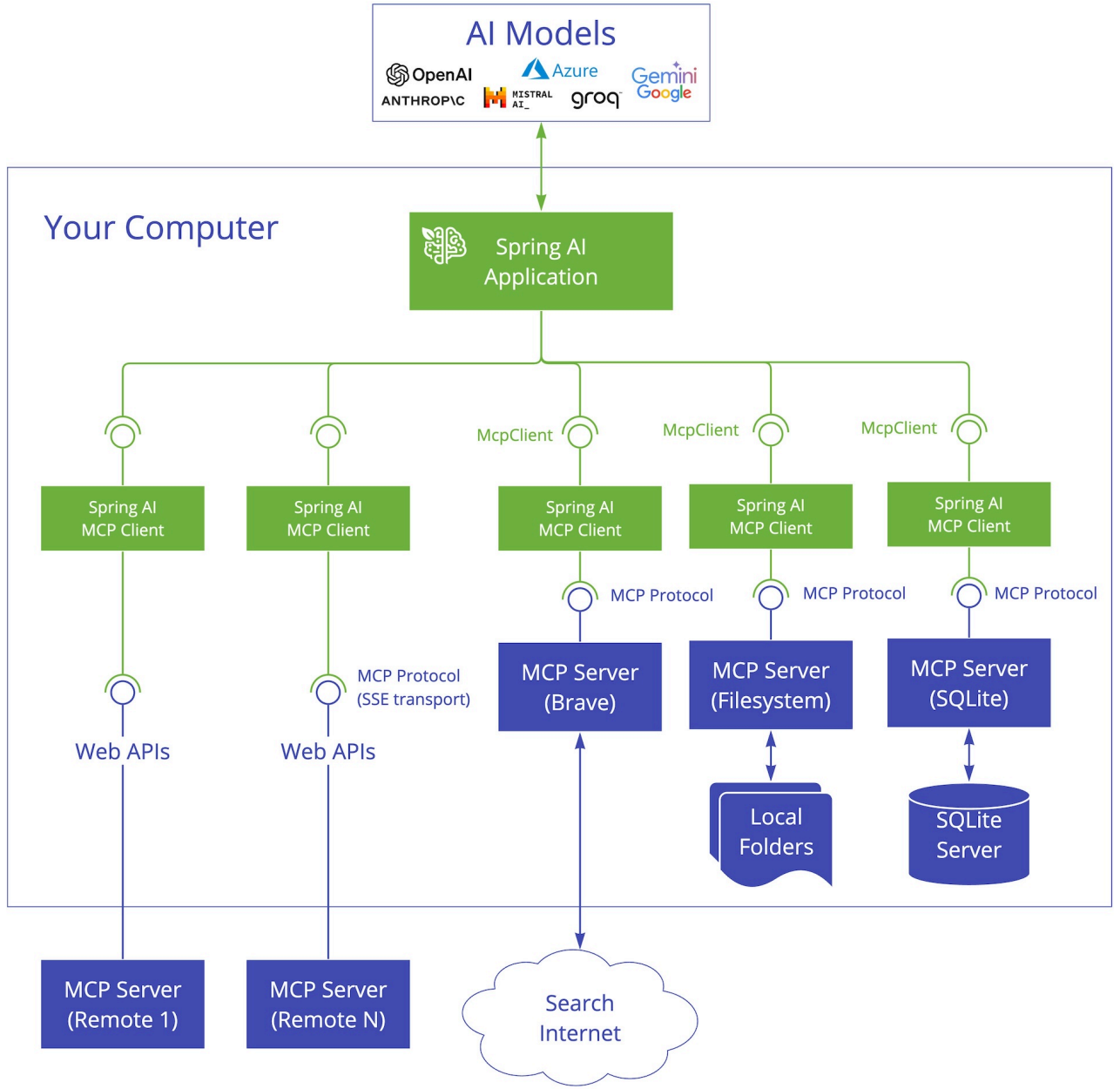
Tool calling

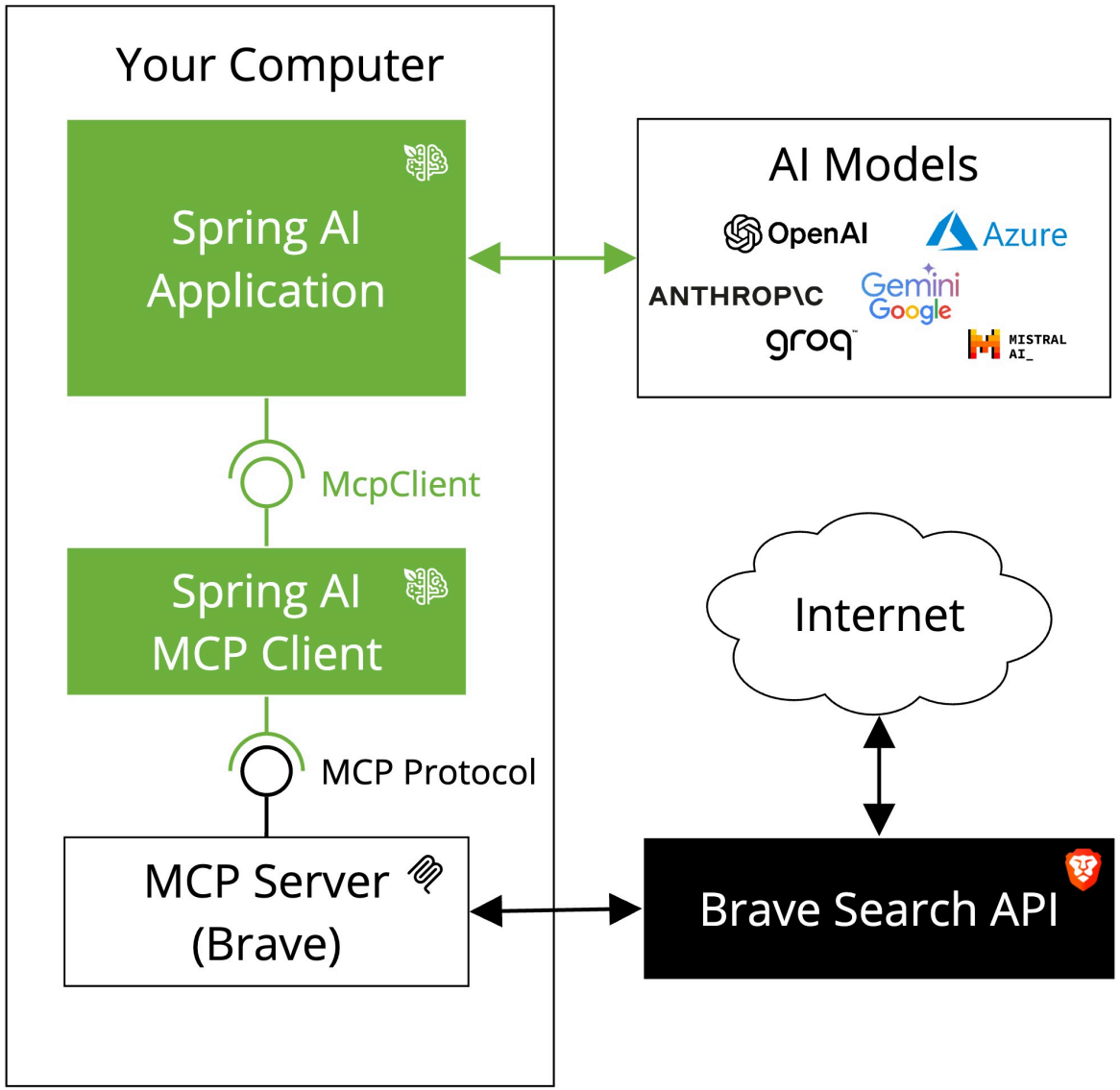
```
@Bean
// Function annotated with @Tool and @ToolParam
// Request annotated with @JsonClassDescription and JsonPropertyDescription
public Function<WeatherService.Request, WeatherService.Response> weatherFunction() {
  return new WeatherService();
}
```

```
String answer = chatClient.prompt()
  .user("What's the weather like in Stockholm, Zurich, and New York?")
  .toolNames("weatherFunction")
  .call()
  .content();

log.info("ChatGPT answered: {}", answer);
```

Model Context Protocol (MCP)


Sell me this pen.
It has an MCP Server.


AI Models
OpenAI
Azure
ANTHROP\C
MISTRAL AI_
groq
Gemini
Google
Your Computer
Spring AI Application
McpClient
McpClient
McpClient
Spring AI MCP Client
Spring AI MCP Client
Spring AI MCP Client
Spring AI MCP Client
Spring AI MCP Client
MCP Protocol
MCP Protocol
MCP Protocol
MCP Protocol (SSE transport)
MCP Server (Brave)
MCP Server (Filesystem)
MCP Server (SQLite)
Web APIs
Web APIs
Local Folders
SQLite Server
MCP Server (Remote 1)
MCP Server (Remote N)
Search Internet

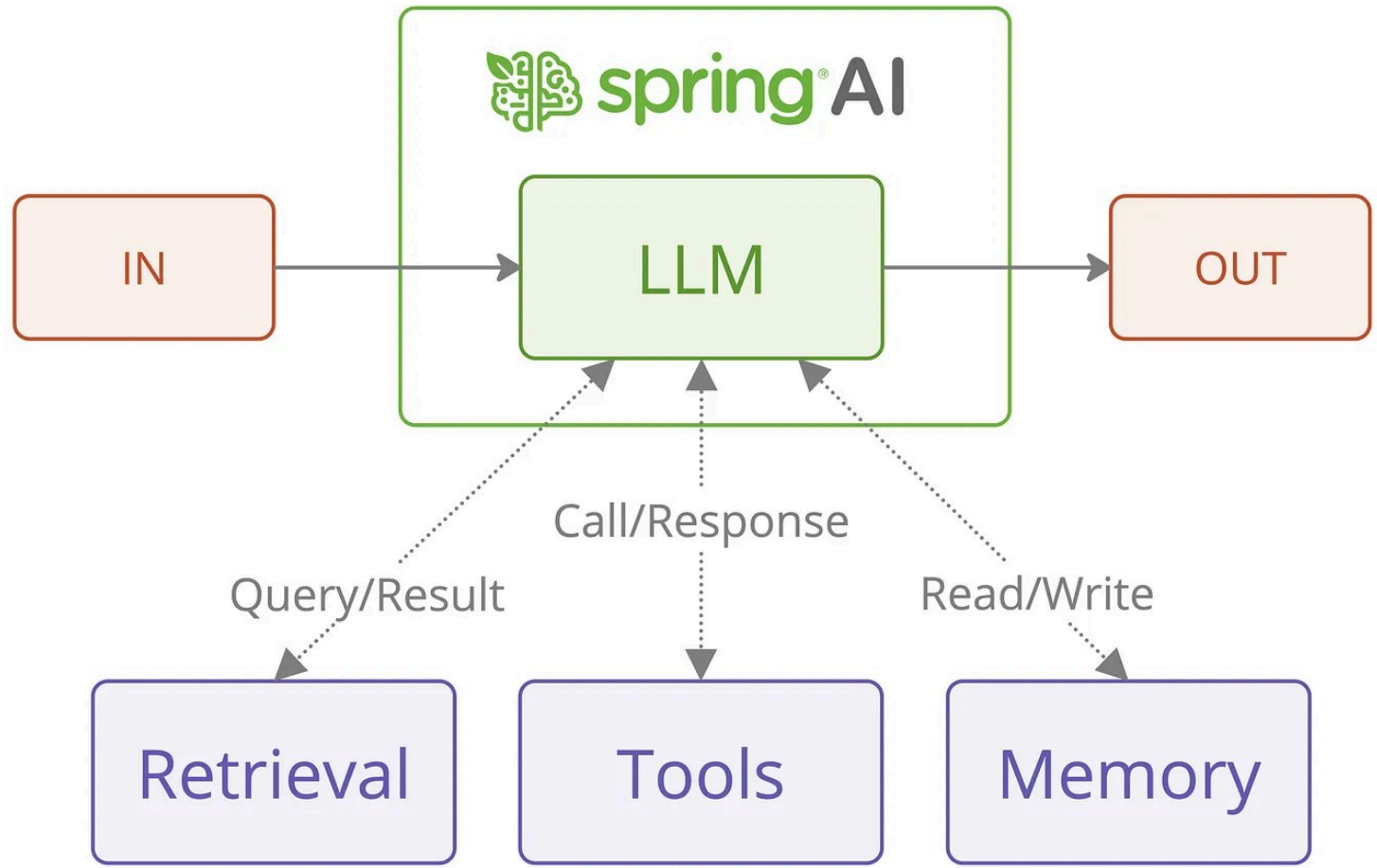

Your Computer
Spring AI Application
AI Models
OpenAI
Azure
ANTHROPIC
Gemini Google
groq
MISTRAL AI_
McpClient
Spring AI MCP Client
Internet
MCP Protocol
MCP Server (Brave)
Brave Search API

Model Context Protocol (MCP)

```
// https://github.com/modelcontextprotocol/servers/tree/main/src/brave-search
var stdioParams = ServerParameters.builder("npx")
  .args("-y", "@modelcontextprotocol/server-brave-search")
  .addEnvVar("BRAVE_API_KEY", System.getenv("BRAVE_API_KEY"))
  .build();

var mcpClient = McpClient.sync(new StdioClientTransport(stdioParams)).build();
var init = mcpClient.initialize();
```

```
var chatClient = chatClientBuilder
  .defaultTools(new SyncMcpToolCallbackProvider(mcpSyncClients))
  .build();
```

Agentic Patterns

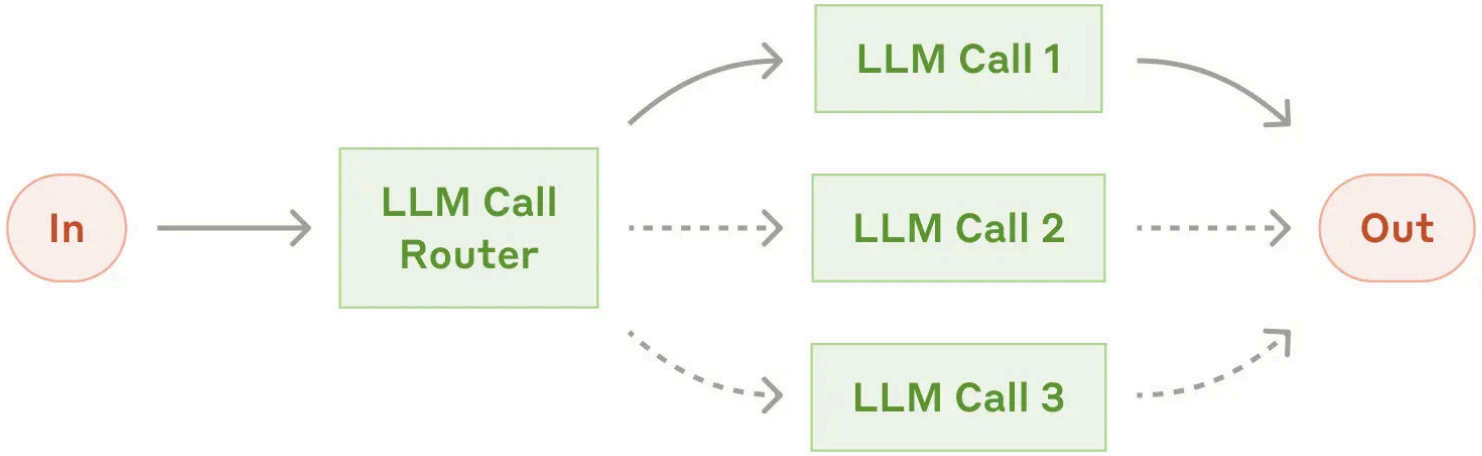

spring AI
IN
LLM
OUT
Call/Response
Query/Result
Read/Write
Retrieval
Tools
Memory

Workflow Patterns

- Prompt chaining
- Routing
- Parallelization
- Orchestrator-workers
- Evaluator-optimizer

See: https://www.anthropic.com/engineering/building-effective-agents and https://github.com/spring-projects/spring-ai-examples/

E.g. Routing Workflow



E.g. Routing Workflow

```
public class RoutingWorkflow {

    private final ChatClient chatClient;

    public RoutingWorkflow(ChatClient chatClient) {
        this.chatClient = chatClient;
    }

    public String route(String input, Map<String, String> routes) {

        String routeKey = determineRoute(input, routes.keySet());

        String selectedPrompt = routes.get(routeKey);

        if (selectedPrompt == null) {
            throw new IllegalArgumentException("Selected route '" + routeKey + "' not found in routes map");
        }

        return chatClient.prompt(selectedPrompt + "\nInput: " + input).call().content();
    }

    ...
}
```

Generative AI Challenges

Generative AI Challenges

Challenges	Patterns
Align responses to goals	System prompts
No structured output	Output converters
Not trained on your data	Prompt stuffing
Limited context size	RAG
Stateless APIs	Chat Memory
Not aware of your APIs	Tool calling
Integration of standard APIs	Model Context Protocol (MCP)
Hallucinations	Evaluators

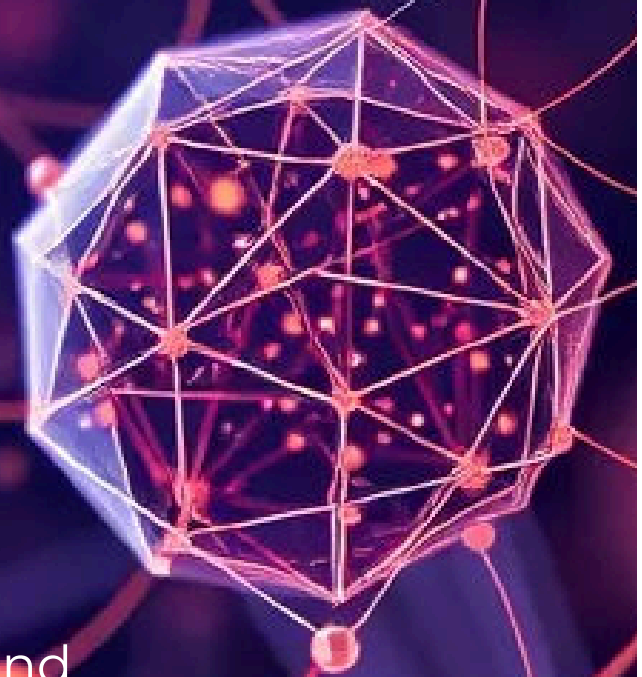
"Artificial Intelligence Will Change the World"

How AI Will Impact the Future

- Improved automation
- Job disruption
- Data privacy issues
- Ethical considerations
- Increased regulation
- Climate change concerns

Famous Last Words ...

Intelligent Applications With Spring AI



Patrick Baumgartner
42talents GmbH, Zürich, Switzerland

@patbaumgartner
patrick.baumgartner@42talents.com