SPRING INTO AI

Building intelligent applications with Spring AI

Dan Vega - Spring Developer Advocate @Broadcom

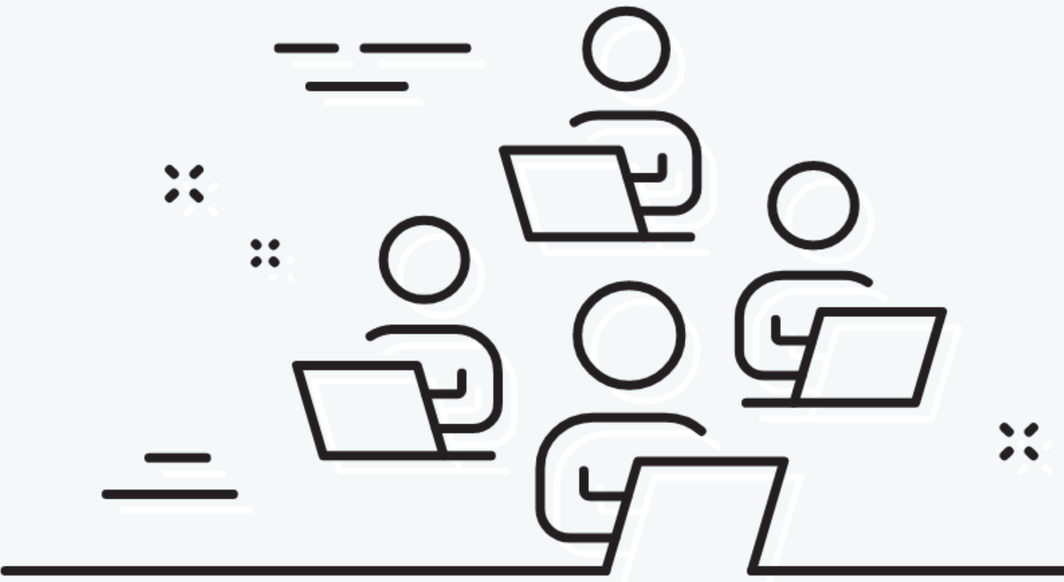
ABOUT ME

Learn more at danvega.dev

- 👨‍👩‍👧‍👦 Husband & Father
- 🏠 Cleveland
- ☕ Java Champion
- 👨‍💻 Software Development 23 Years
- 🌿 Spring Developer Advocate
- 📝 Content Creator
- 🐦 @therealdanvega
- 🏃‍♂️ Running
- 🏌️‍♂️ Golf

spring

OFFICE HOURS



https://www.springofficehours.io

AGENDA

What we will cover today

What is AI?

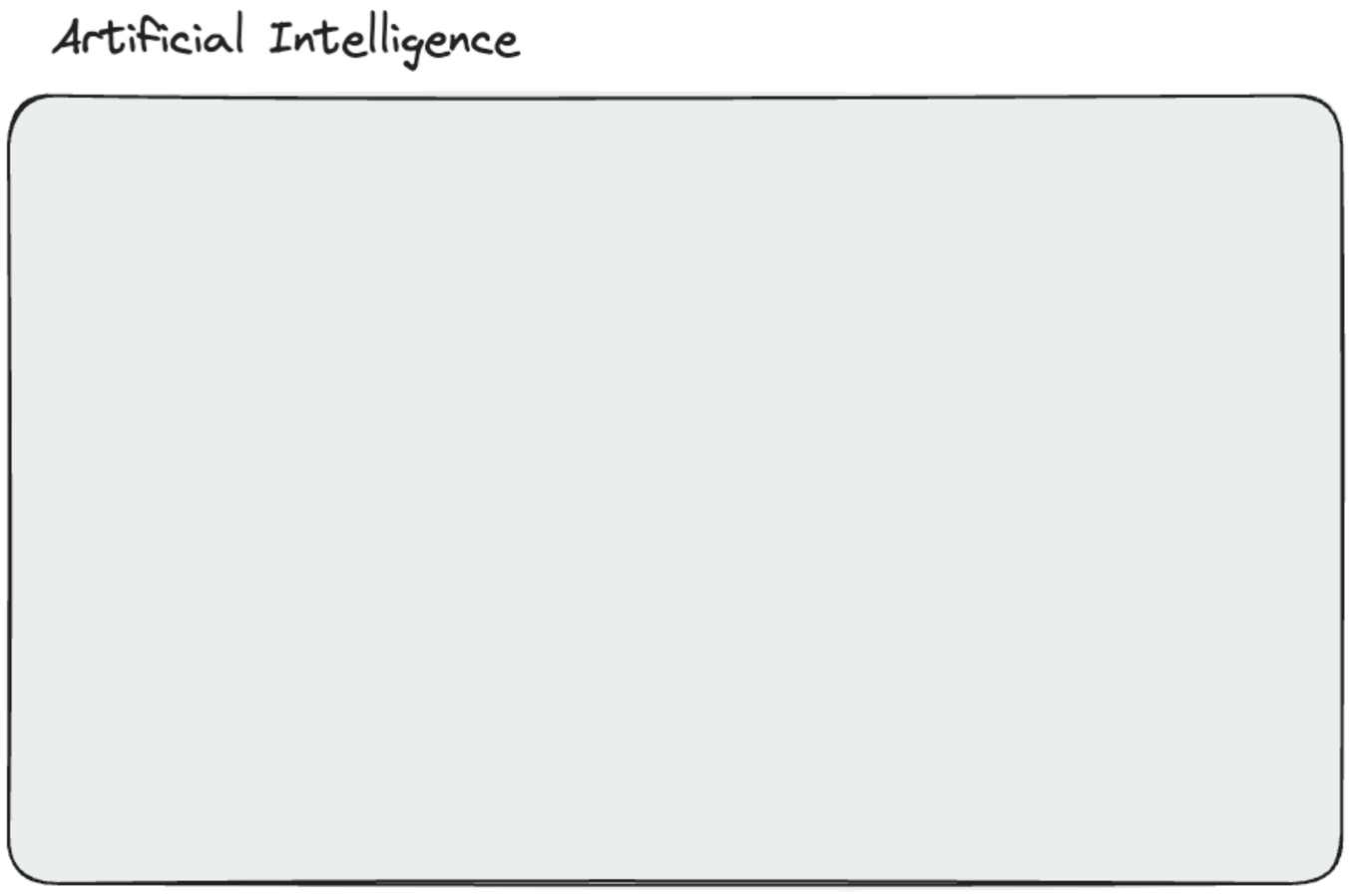
Java & AI

Spring AI

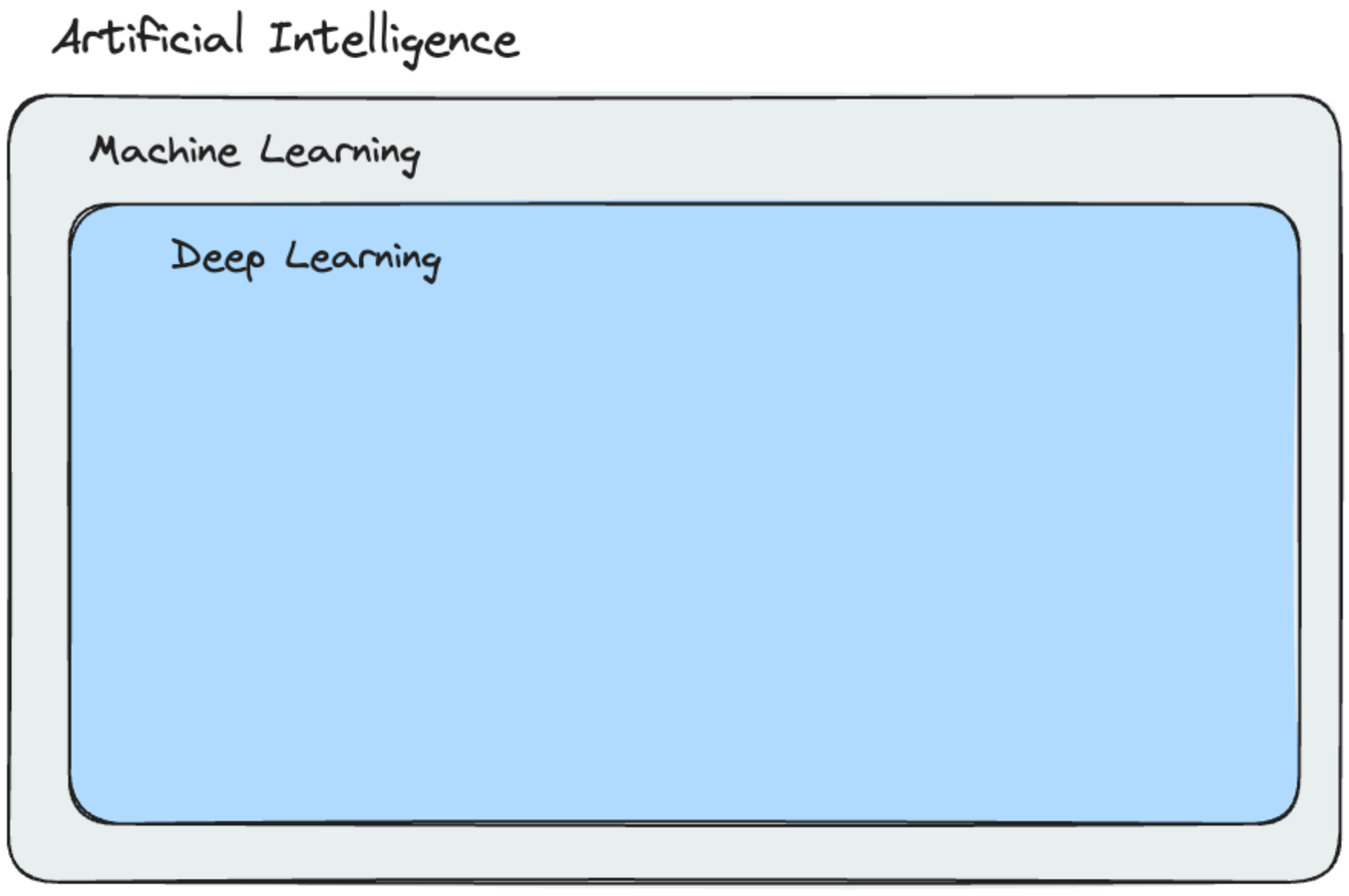
- How to get started
- Models
- Prompts
- Output Parsers
- Bring your own data

WHAT IS ARTIFICIAL INTELLIGENCE (AI)?

WHAT IS ARTIFICIAL INTELLIGENCE

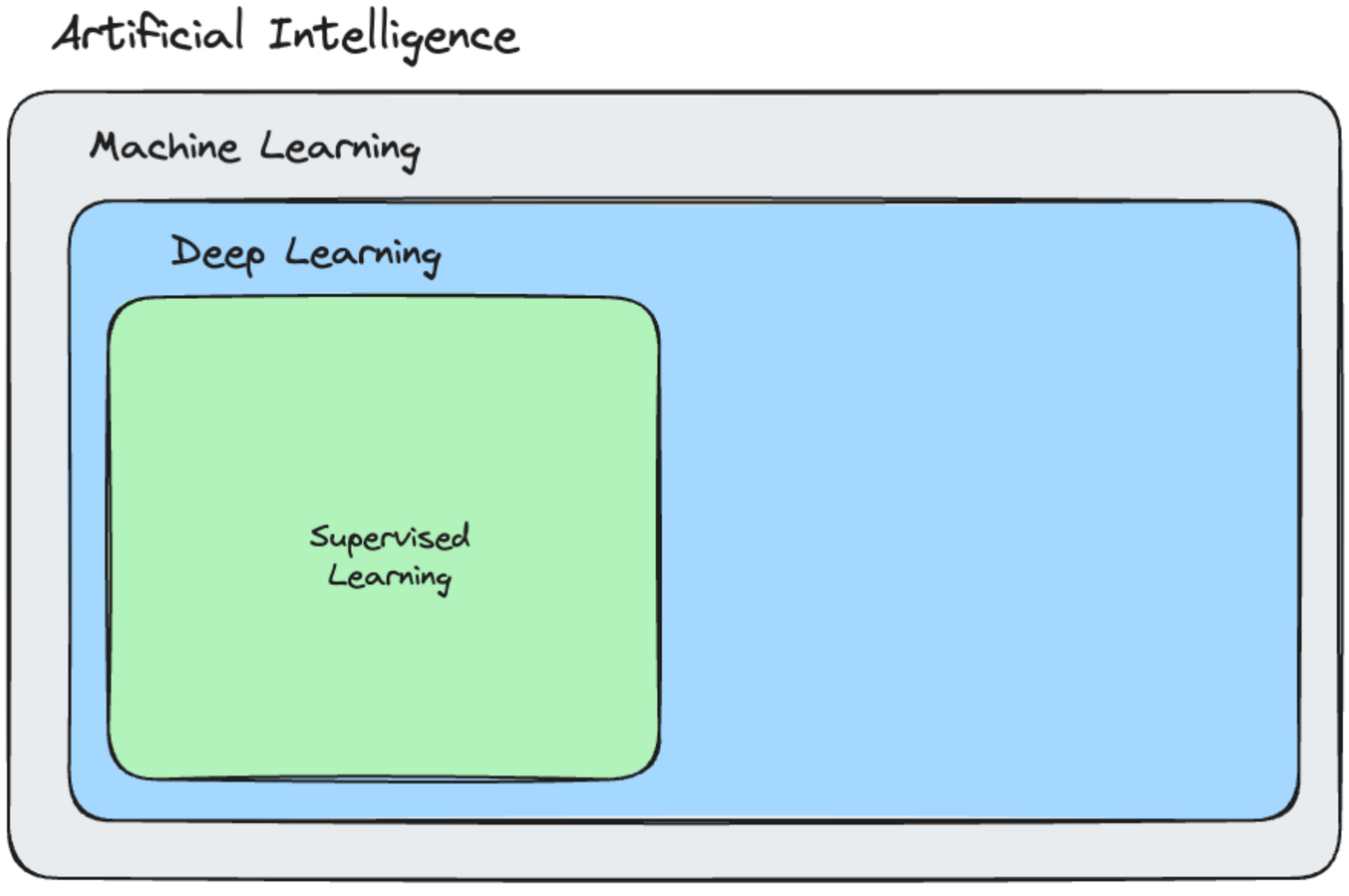


MACHINE LEARNING



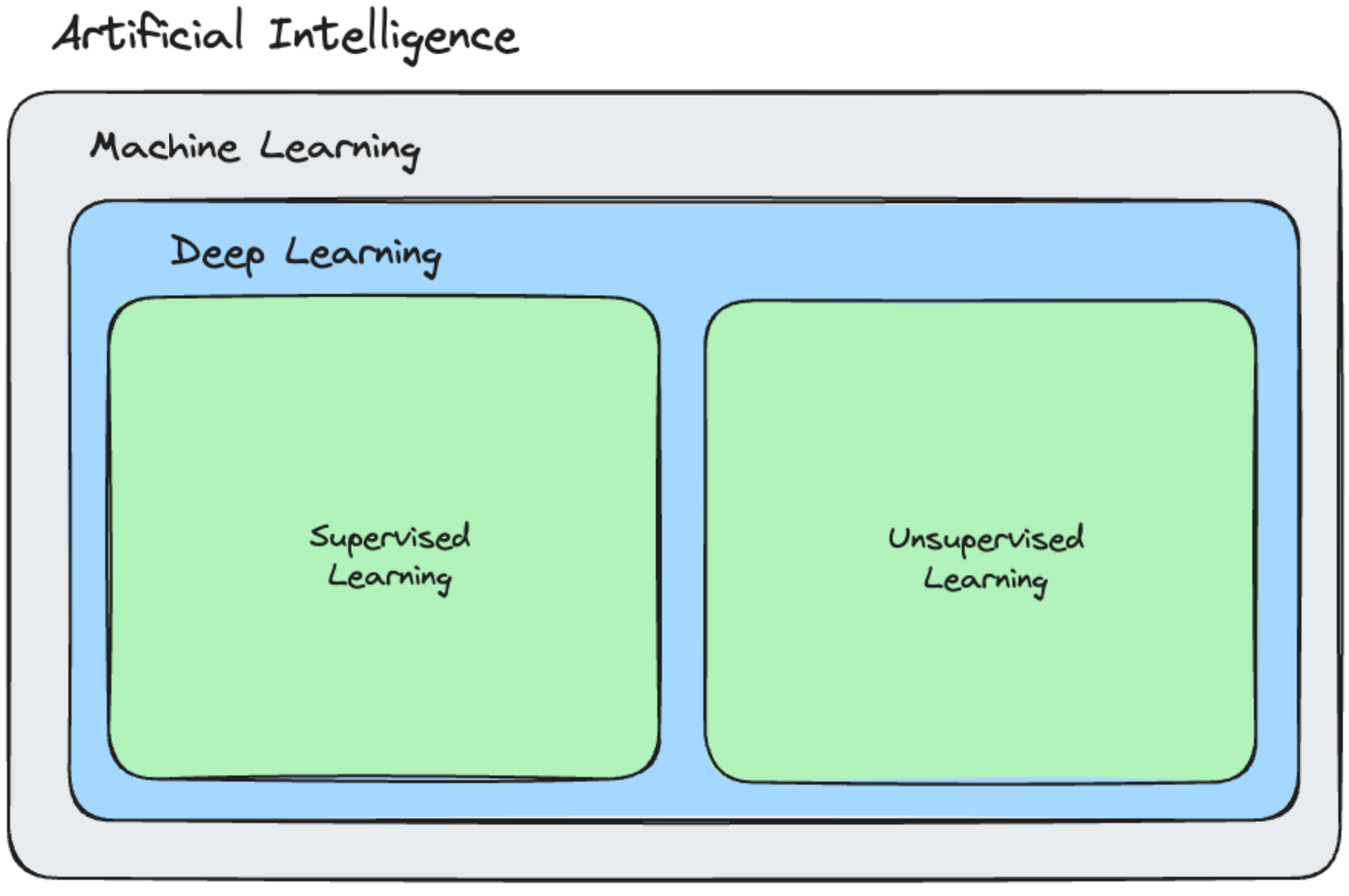
MACHINE LEARNING

Supervised Learning

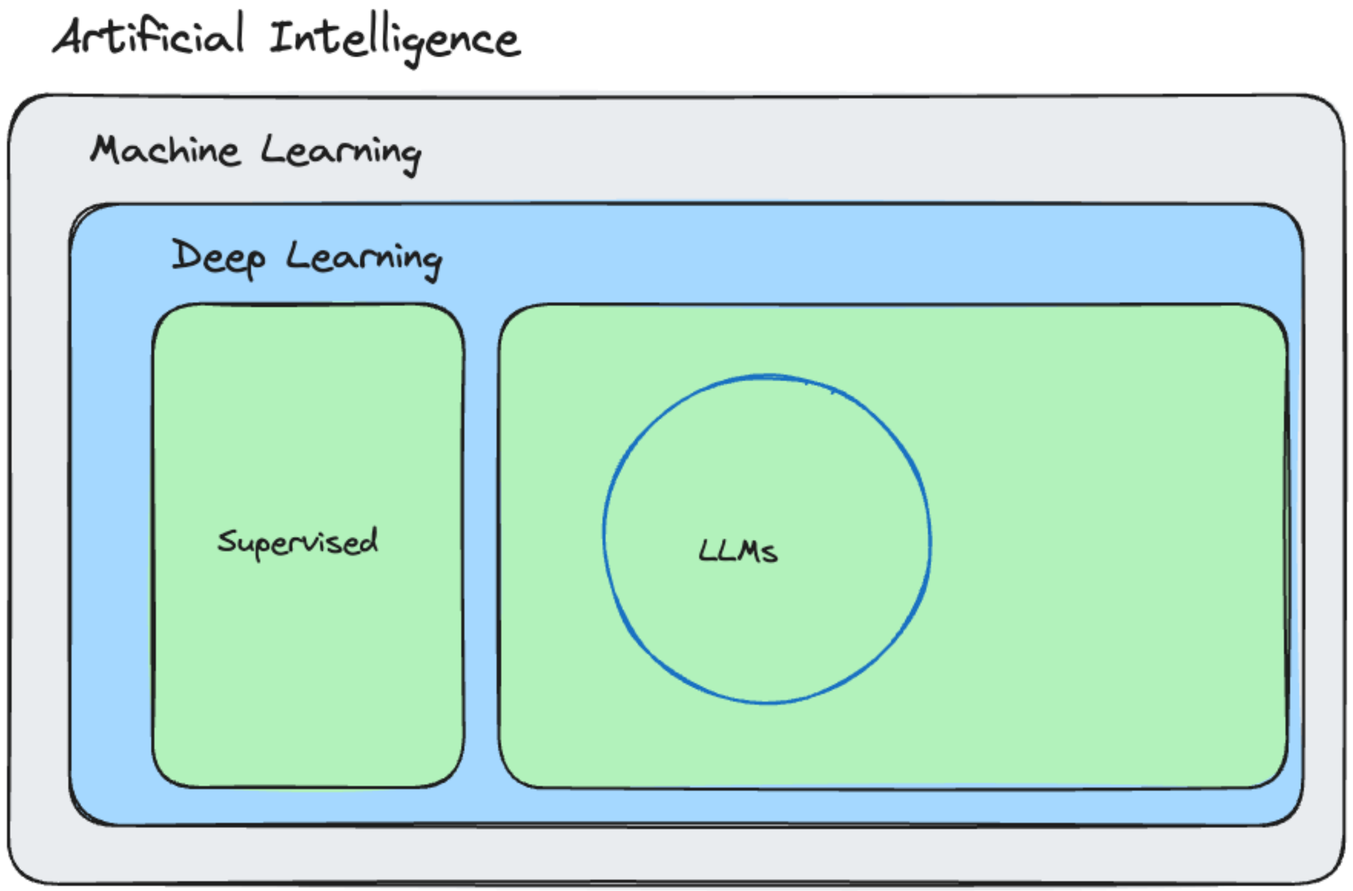


MACHINE LEARNING

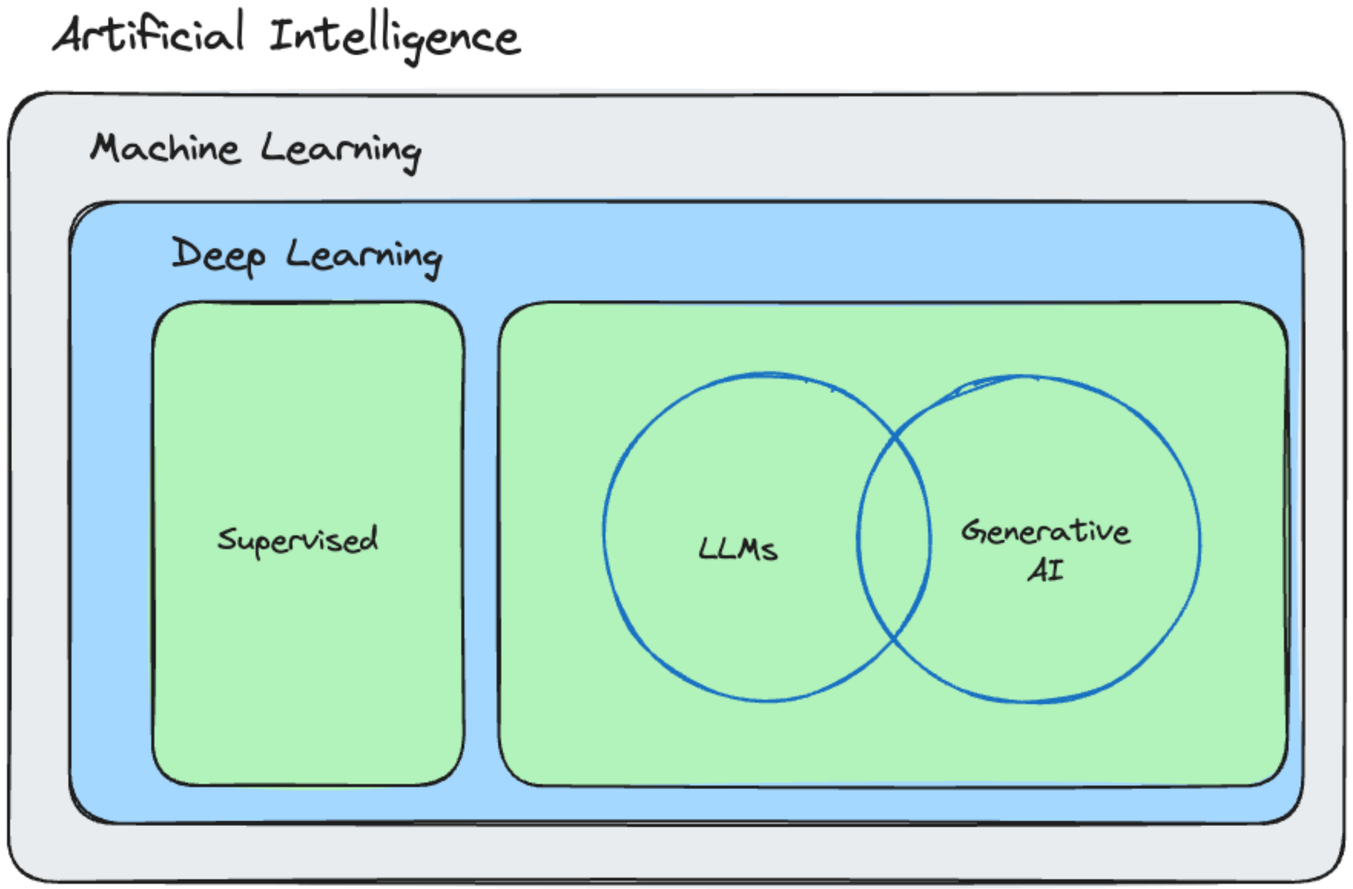
Unsupervised Learning



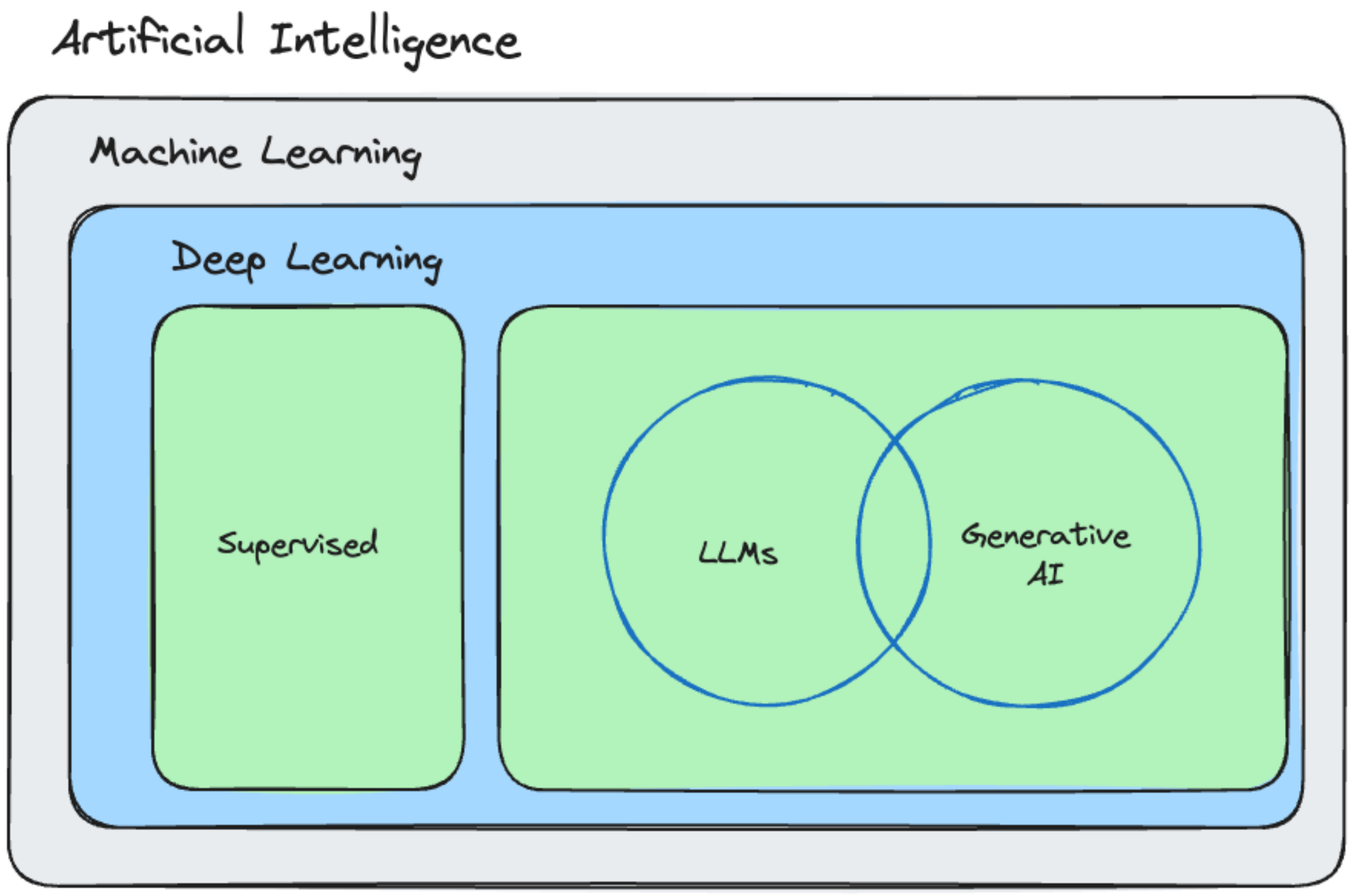
LARGE LANGUAGE MODELS (LLM)



GENERATIVE AI

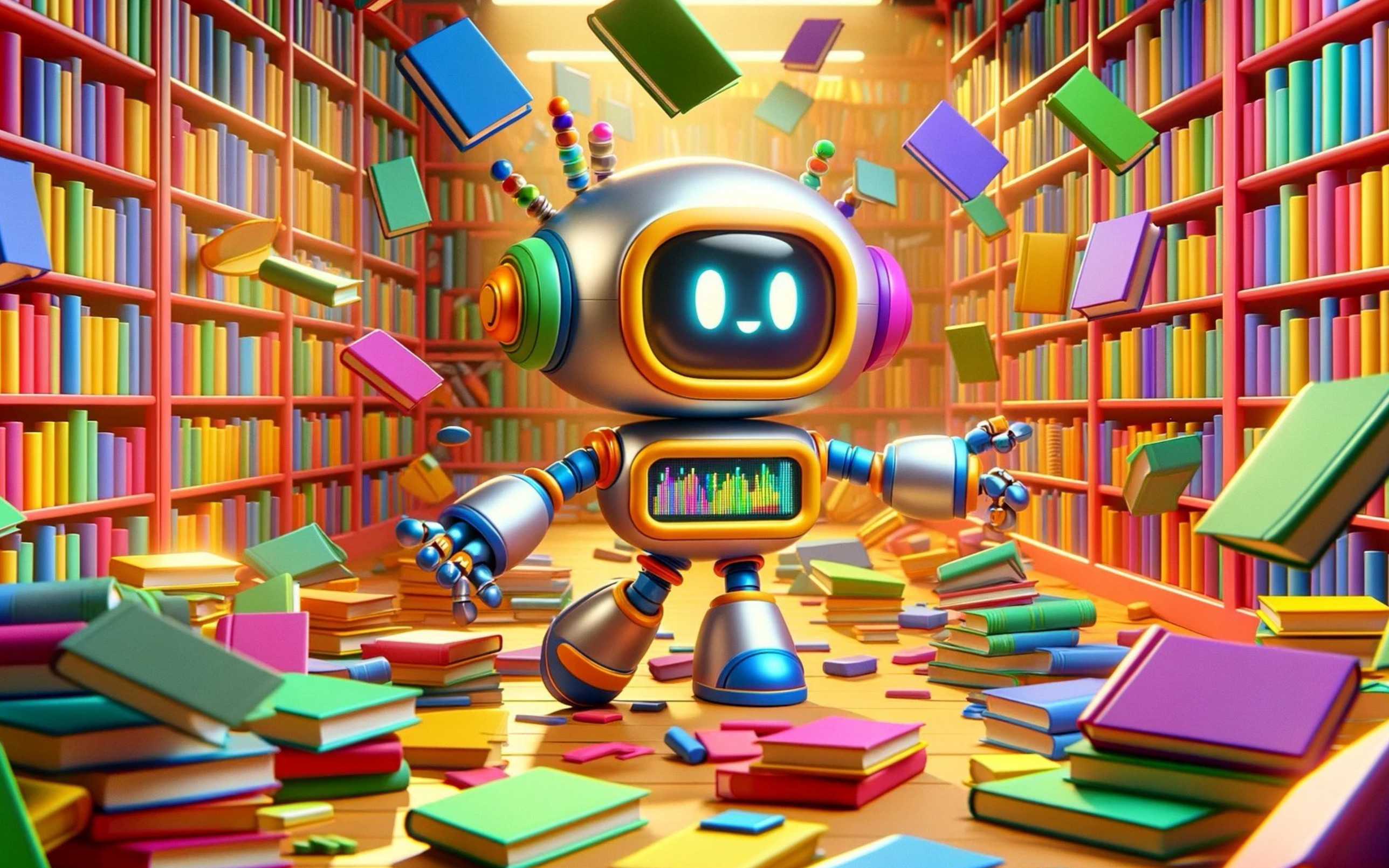


GENERATIVE AI

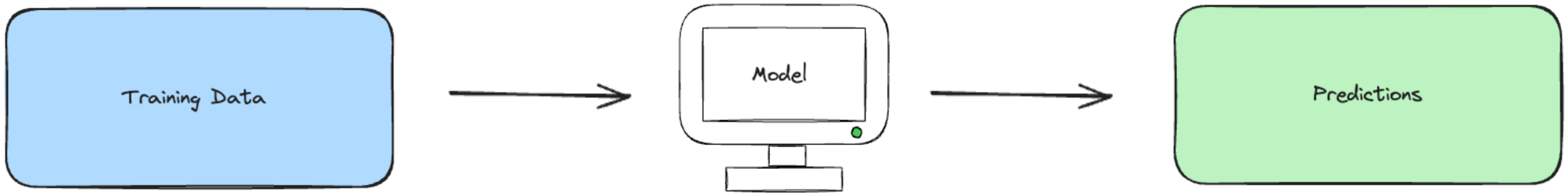


Gemini

MACHINE LEARNING



WHAT IS MACHINE LEARNING?



MACHINE LEARNING

Use Cases

- Facial Recognition
- Recognize Tumors on x-ray scans
- Abnormality on ultrasounds
- Self-driving mode (recognize stop sign / pedestrian / etc)
- Fraud detection
- Product Recommendations (YouTube)
- Spam Filtering

SUPERVISED LEARNING

Labeling the training data

Dan Vega

Dan Vega

Dan Vega

TRAINING DATA

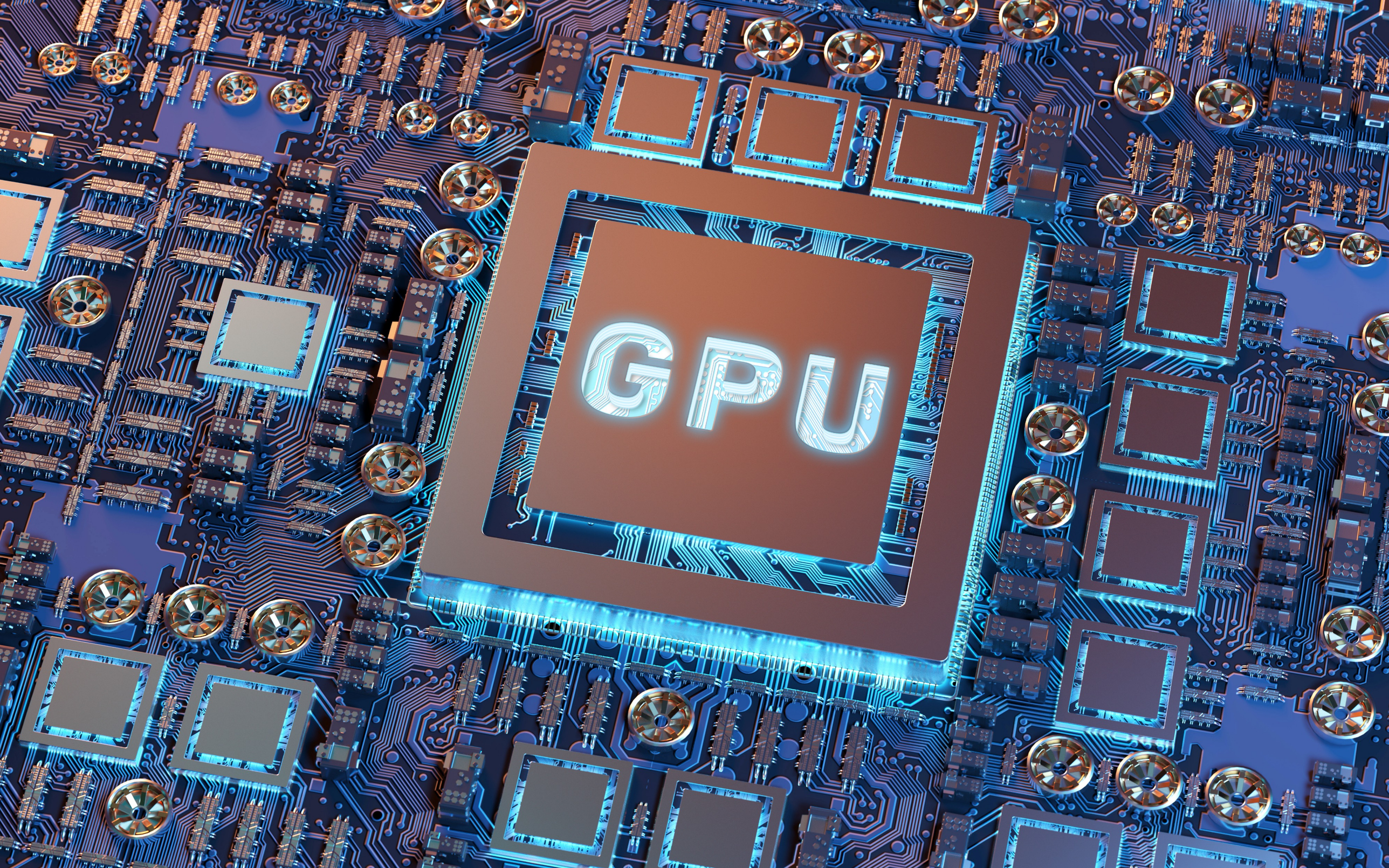
Supervised Learning of Big Data Companies

HOW DOES MACHINE LEARNING WORK?

DEEP LEARNING

ARTIFICIAL NEURAL NETWORK

- Scientific Advances - Deep Learning
- Availability of Big Data (You need data to configure these neural networks)
- Lots of compute power


GPU

LARGE LANGUAGE MODELS (LLM)

ATTENTION IS ALL YOU NEED

Bigger is Better

- Very Large Neural Networks
- Vast Amounts of Training Data
- Huge Compute Power to Train Data
- General Purpose AI

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

1 Introduction

Recurrent neural networks, long short-term memory [12] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation [29, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [31, 21, 13].

*Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

ATTENTION IS ALL YOU NEED

Transformer Architecture

- Specialized architecture for token prediction
- Key Innovation
  - Attention mechanisms
- Not Just a big neural network

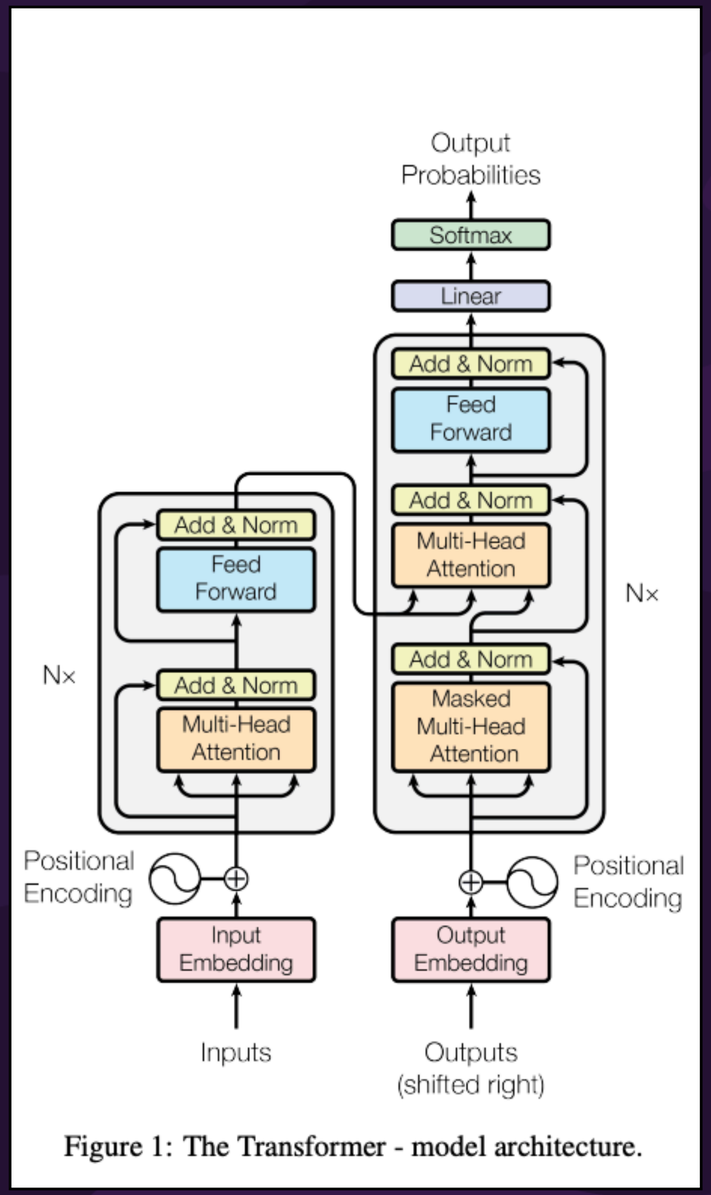


Figure 1: The Transformer - model architecture.

LARGE LANGUAGE MODELS

So what are LLMs

- LLMs are a type of artificial intelligence that can generate text, understand language, answer questions and more.
- They are large because they have a vast number of parameters, which are the parts of the model that are learned from data during training.
- ChatGPT
  - 175 Billion Parameters
  - Training Data - Hundreds of Billions of words

Tasks Libraries Datasets Languages Licenses Other

Filter Tasks by name

Multimodal

- Image-Text-to-Text
- Visual Question Answering
- Document Question Answering

Computer Vision

- Depth Estimation
- Image Classification
- Object Detection
- Image Segmentation
- Text-to-Image
- Image-to-Text
- Image-to-Image
- Image-to-Video
- Unconditional Image Generation
- Video Classification
- Text-to-Video
- Zero-Shot Image Classification
- Mask Generation
- Zero-Shot Object Detection
- Text-to-3D
- Image-to-3D
- Image Feature Extraction

Natural Language Processing

- Text Classification
- Token Classification
- Table Question Answering
- Question Answering
- Zero-Shot Classification
- Translation
- Summarization
- Feature Extraction
- Text Generation
- Text2Text Generation
- Fill-Mask
- Sentence Similarity

Audio

- Text-to-Speech
- Text-to-Audio
- Automatic Speech Recognition
- Audio-to-Audio
- Audio Classification
- Voice Activity Detection

Tabular

- Tabular Classification
- Tabular Regression

Models 570,761

Filter by name

new Full-text search | Sort: Trending

- xai-org/grok-1
  Text Generation • Updated 9 days ago • ♡ 1.74k
- databricks/dbrx-instruct
  Text Generation • Updated about 13 hours ago • ↓ 438 • ♡ 266
- mistralai/Mistral-7B-Instruct-v0.2
  Text Generation • Updated 4 days ago • ↓ 2.02M • ♡ 1.43k
- ByteDance/AnimateDiff-Lightning
  Text-to-Video • Updated 7 days ago • ↓ 57.9k • ♡ 344
- stabilityai/sv3d
  Image-to-Video • Updated 9 days ago • ♡ 395
- databricks/dbrx-base
  Text Generation • Updated about 13 hours ago • ↓ 381 • ♡ 165
- google/gemma-7b
  Text Generation • Updated 29 days ago • ↓ 221k • ♡ 2.63k
- meta-llama/Llama-2-7b-chat-hf
  Text Generation • Updated 9 days ago • ↓ 1.35M • ♡ 3.21k
- Nexusflow/Starling-LM-7B-beta
  Text Generation • Updated about 22 hours ago • ↓ 3.29k • ♡ 129
- alpindale/Mistral-7B-v0.2-hf
  Text Generation • Updated 3 days ago • ↓ 5.99k • ♡ 107
- mistralai/Mixtral-8x7B-Instruct-v0.1
  Text Generation • Updated 28 days ago • ↓ 969k • ♡ 3.48k
- ByteDance/SDXL-Lightning
  Text-to-Image • Updated 14 days ago • ↓ 680k • ♡ 1.46k
- NousResearch/Hermes-2-Pro-Mistral-7B
  Text Generation • Updated 13 days ago • ↓ 37.5k • ♡ 337
- cagliostrolab/animagine-xl-3.1
  Text-to-Image • Updated 10 days ago • ↓ 46.5k • ♡ 242
- stabilityai/stable-diffusion-xl-base-1.0
  Text-to-Image • Updated Oct 30, 2023 • ↓ 3.2M • ♡ 4.82k
- openai/whisper-large-v3
  Automatic Speech Recognition • Updated Feb 8 • ↓ 1.17M • ♡ 2.1k
- stabilityai/stable-code-instruct-3b
  Text Generation • Updated 2 days ago • ↓ 1.2k • ♡ 58
- BAAI/bge-m3
  Sentence Similarity • Updated 2 days ago • ↓ 2.24M • ♡ 607
- hpcai-tech/Open-Sora
  Updated 8 days ago • ♡ 120
- runwayml/stable-diffusion-v1-5
  Text-to-Image • Updated Aug 23, 2023 • ↓ 4M • ♡ 10.5k
- hpcai-tech/grok-1
  Text Generation • Updated 2 days ago • ↓ 1k • ♡ 52
- distil-whisper/distil-large-v3
  Automatic Speech Recognition • Updated about 10 hours ago • ↓ 10.1k • ♡ 51
- CohereForAI/c4ai-command-r-v01
- ostris/ip-composition-adapter

GENERATIVE AI

GENERATIVE PRE-TRAINED TRANSFORMER (GPT)

WHAT IS GENERATIVE AI?

NOT JUST Machine Learning

- Unlike the facial recognition example we saw earlier Generative AI can take the training data and generate something completely new
- NOT Generative Ai
  - Number
  - Classification
  - Probability
- IS Generative AI
  - Natural Language (Text or Speech)
  - Image
  - Audio

JAVA & AI

JAVA & AI

How can we leverage AI?

- Why AI + Java
  - AI is becoming ubiquitous across the IT landscape
  - Java is the language of enterprise, creating Java AI apps is a new requirement
- Spring AI
  - Provide the necessary API access and components for developing AI apps
- Use Cases
  - Q&A over docs
  - Documentation summarization
  - Text, Code, Image, Audio and Video Generation

```
#!/bin/bash
echo "Calling Open AI..."
MY_OPENAI_KEY="YOUR_API_KEY_HERE"
PROMPT="When was the first version of Java released?"

curl https://api.openai.com/v1/chat/completions \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $MY_OPENAI_KEY" \
  -d '{"model": "gpt-3.5-turbo", "messages": [{"role":"user", "content": "'"${PROMPT}"'"}] }'
```


vega@Dans-MacBook-Pro-M1-MAX:~/dev/spring-ai/scripts
⌥⌘1
dev/spring-ai/scripts 🚀 ./hello-open-ai.sh
Calling Open AI...
{
"id": "chatcmpl-96qnh95F1pliuO9tTk5rSvpj7pZCR",
"object": "chat.completion",
"created": 1711419961,
"model": "gpt-3.5-turbo-0125",
"choices": [
{
"index": 0,
"message": {
"role": "assistant",
"content": "The first version of Java was released on January 23, 1996."
},
"logprobs": null,
"finish_reason": "stop"
}
],
"usage": {
"prompt_tokens": 16,
"completion_tokens": 16,
"total_tokens": 32
},
"system_fingerprint": "fp_3bc1b5746c"
}
dev/spring-ai/scripts 🚀 _

```java
public static void main(String[] args) throws IOException, InterruptedException {
    var apiKey = "YOUR_API_KEY_HERE";
    var body = """
            {
                "model": "gpt-4",
                "messages": [
                    {
                        "role": "user",
                        "content": "What is Spring Boot?"
                    }
                ]
            }""";

    HttpRequest request = HttpRequest.newBuilder()
            .uri(URI.create("https://api.openai.com/v1/chat/completions"))
            .header("Content-Type", "application/json")
            .header("Authorization", "Bearer " + apiKey)
            .POST(HttpRequest.BodyPublishers.ofString(body))
            .build();

    var client = HttpClient.newHttpClient();
    var response = client.send(request, HttpResponse.BodyHandlers.ofString());
    System.out.println(response.body());
}
```

SPRING AI PROVIDES US SO MUCH MORE THAN A FACILITY FOR MAKING REST API CALLS

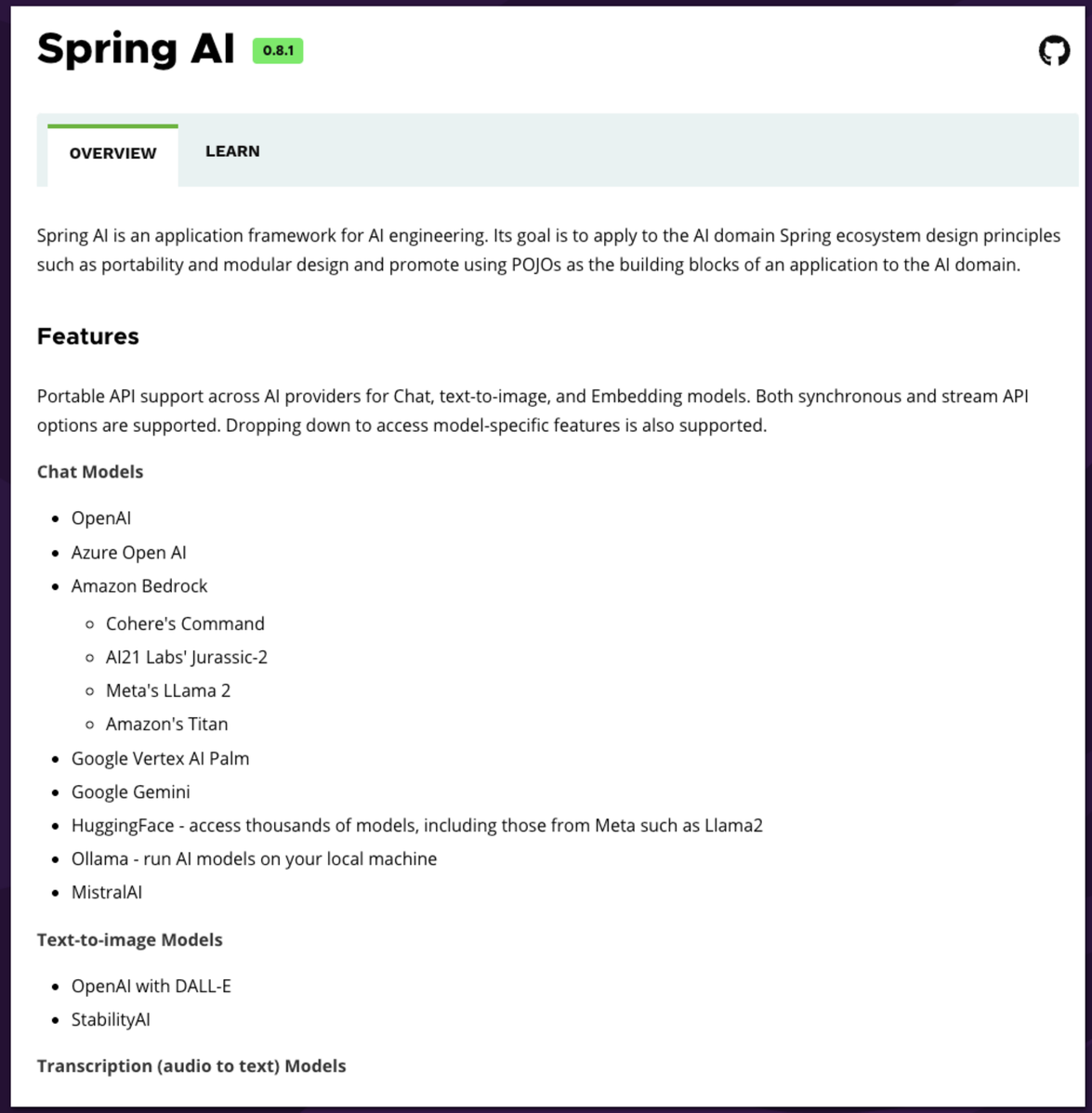
SPRING AI

WHAT IS SPRING AI?

Spring AI

AI for Spring Developers

- A new Spring Project
  - Mark Pollack
  - Current Version 1.0.0-M1
  - https://spring.io/projects/spring-ai
- Inspired by Python projects
  - LangChain
  - LlamaIndex



Spring AI 0.8.1

OVERVIEW LEARN

Spring AI is an application framework for AI engineering. Its goal is to apply to the AI domain Spring ecosystem design principles such as portability and modular design and promote using POJOs as the building blocks of an application to the AI domain.

Features

Portable API support across AI providers for Chat, text-to-image, and Embedding models. Both synchronous and stream API options are supported. Dropping down to access model-specific features is also supported.

Chat Models

- OpenAI
- Azure Open AI
- Amazon Bedrock
  - Cohere's Command
  - AI21 Labs' Jurassic-2
  - Meta's LLama 2
  - Amazon's Titan
- Google Vertex AI Palm
- Google Gemini
- HuggingFace - access thousands of models, including those from Meta such as Llama2
- Ollama - run AI models on your local machine
- MistralAI

Text-to-image Models

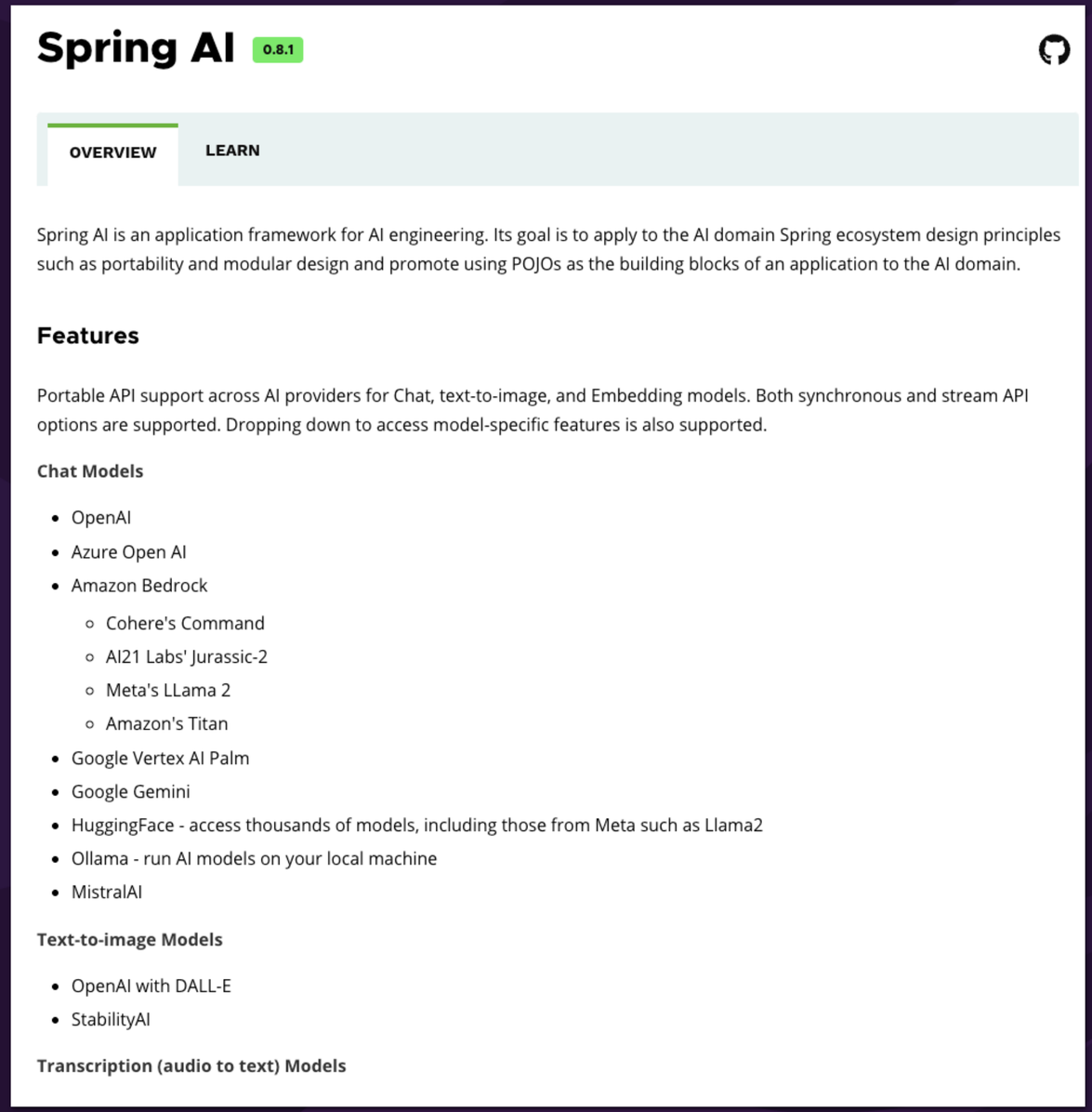
- OpenAI with DALL-E
- StabilityAI

Transcription (audio to text) Models

Spring AI

AI for Spring Developers

- Aligns with Spring project design values
  - Component Abstractions & Default Implementations
  - Portable Chat Completion and EmbeddingClient
  - Multimodality Support
  - Portable Vector Store API & Query Language
  - Function Calling
- Key Components
  - Models
  - Data
  - Chain
  - Evaluation



Spring AI 0.8.1

OVERVIEW LEARN

Spring AI is an application framework for AI engineering. Its goal is to apply to the AI domain Spring ecosystem design principles such as portability and modular design and promote using POJOs as the building blocks of an application to the AI domain.

Features

Portable API support across AI providers for Chat, text-to-image, and Embedding models. Both synchronous and stream API options are supported. Dropping down to access model-specific features is also supported.

Chat Models

- OpenAI
- Azure Open AI
- Amazon Bedrock
  - Cohere's Command
  - AI21 Labs' Jurassic-2
  - Meta's LLama 2
  - Amazon's Titan
- Google Vertex AI Palm
- Google Gemini
- HuggingFace - access thousands of models, including those from Meta such as Llama2
- Ollama - run AI models on your local machine
- MistralAI

Text-to-image Models

- OpenAI with DALL-E
- StabilityAI

Transcription (audio to text) Models

SPRING AI API

The Spring AI API covers a wide range of functionalities

Chat Model

Give it some text, get some text back

Text to Image

Give it some text, get an image back

Transcription

Give it some audio, get some text back

Embedding

Portable API across Vector Store providers

Functions

AI model invoke your POJO java.util.Function object

CHAT MODEL

Give it some text, get some text back

- Open AI
- Azure Open AI
- Amazon Bedrock
- Google Vertex AI Palm
- Google Gemeni
- HuggingFace - Access to thousands of models, including those from Meta such as Llama 2
- Ollama - Run AI Models on your local machine
- MistralAI

TEXT-TO-IMAGE MODELS

Give it some text, get an image back

- OpenAI with DALL-E
- StabilityAI

TRANSCRIPTION

Give it some audio, get some text back

- Open AI

Embedding Models

Portable API across Vector Store providers

- Open AI
- Azure Open AI
- Ollama
- ONNX
- PostgresML
- Bedrock Cohere
- Bedrock Titan
- Google VertexAI
- MistalAI

GETTING STARTED

spring initializr

Project

- ○ Gradle - Groovy
- ○ Gradle - Kotlin
- ● Maven

Language

- ● Java
- ○ Kotlin
- ○ Groovy

Spring Boot

- ○ 3.3.0 (SNAPSHOT)
- ○ 3.3.0 (M3)
- ○ 3.2.5 (SNAPSHOT)
- ● 3.2.4
- ○ 3.1.11 (SNAPSHOT)
- ○ 3.1.10

Project Metadata

Group	dev.danvega
Artifact	hello-ai
Name	hello-ai
Description	Demo project for Spring Boot
Package name	dev.danvega
Packaging	● Jar ○ War
Java	● 22 ○ 21 ○ 17

Dependencies ADD DEPENDENCIES... ⌘ + B

Spring Web WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

OpenAI AI

Spring AI support for ChatGPT, the AI language model and DALL-E, the Image generation model from OpenAI.

GENERATE ⌘ + ↵ EXPLORE CTRL + SPACE SHARE...

Introducing Sora: Creating video from text

Learn more about Sora

Pioneering research on the

Transforming work and

Join us in shaping the future of

https://platform.openai.com/api-keys

API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically disable any API key that we've found has leaked publicly.

Enable tracking to see usage per API key on the Usage page.

NAME	SECRET KEY	TRACKING	CREATED	LAST USED	PERMISSIONS	
Secret key	sk-...aWYE	+ Enable	Feb 6, 2023	Never	All	
spring-ai	sk-...NKve	Enabled	Mar 12, 2024	Mar 13, 2024	All	

+ Create new secret key

Default organization

If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the API keys above.

Personal

Note: You can also specify which organization to use for each API request. See Authentication to learn more.

TOKENS

☐ Show prices per 1K tokens

Language models

Multiple models, each with different capabilities and price points. Prices can be viewed in units of either per 1M or 1K tokens. You can think of tokens as pieces of words, where 1,000 tokens is about 750 words. This paragraph is 35 tokens.

GPT-4

With broad general knowledge and domain expertise, GPT-4 can follow complex instructions in natural language and solve difficult problems with accuracy.

Learn about GPT-4

Model	Input	Output
gpt-4	$30.00 / 1M tokens	$60.00 / 1M tokens
gpt-4-32k	$60.00 / 1M tokens	$120.00 / 1M tokens

GPT-3.5 Turbo

GPT-3.5 Turbo models are capable and cost-effective.

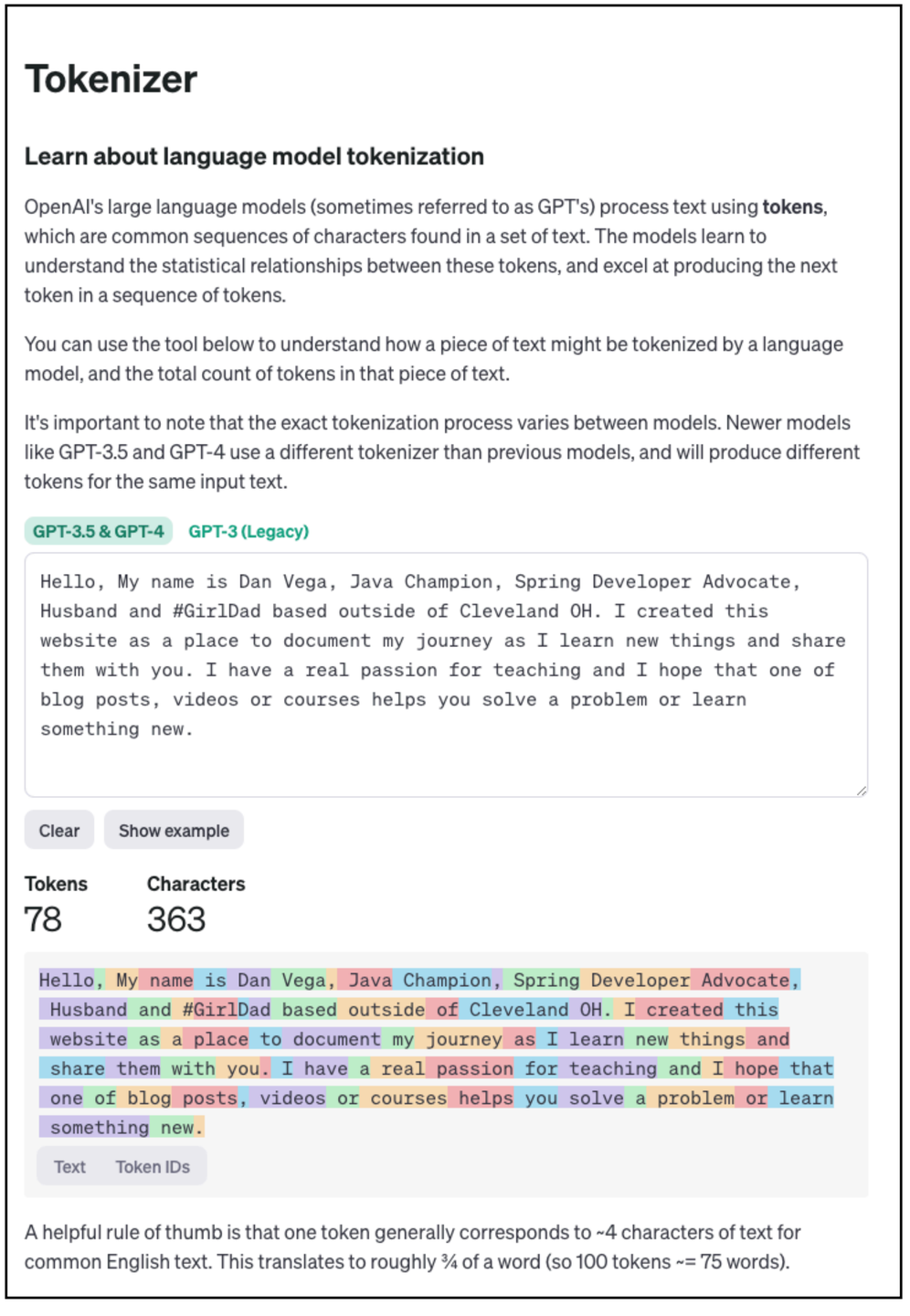
`gpt-3.5-turbo-0125` is the flagship model of this family, supports a 16K context window and is optimized for dialog.

`gpt-3.5-turbo-instruct` is an Instruct model and only supports a 4K context window.

Learn about GPT-3.5 Turbo ↗

Model	Input	Output
gpt-3.5-turbo-0125	$0.50 / 1M tokens	$1.50 / 1M tokens
gpt-3.5-turbo-instruct	$1.50 / 1M tokens	$2.00 / 1M tokens

https://platform.openai.com/tokenizer



Tokenizer

Learn about language model tokenization

OpenAI's large language models (sometimes referred to as GPT's) process text using **tokens**, which are common sequences of characters found in a set of text. The models learn to understand the statistical relationships between these tokens, and excel at producing the next token in a sequence of tokens.

You can use the tool below to understand how a piece of text might be tokenized by a language model, and the total count of tokens in that piece of text.

It's important to note that the exact tokenization process varies between models. Newer models like GPT-3.5 and GPT-4 use a different tokenizer than previous models, and will produce different tokens for the same input text.

GPT-3.5 & GPT-4 | GPT-3 (Legacy)

Hello, My name is Dan Vega, Java Champion, Spring Developer Advocate, Husband and #GirlDad based outside of Cleveland OH. I created this website as a place to document my journey as I learn new things and share them with you. I have a real passion for teaching and I hope that one of blog posts, videos or courses helps you solve a problem or learn something new.

Clear | Show example

Tokens 78 **Characters** 363

Hello, My name is Dan Vega, Java Champion, Spring Developer Advocate, Husband and #GirlDad based outside of Cleveland OH. I created this website as a place to document my journey as I learn new things and share them with you. I have a real passion for teaching and I hope that one of blog posts, videos or courses helps you solve a problem or learn something new.

Text | Token IDs

A helpful rule of thumb is that one token generally corresponds to ~4 characters of text for common English text. This translates to roughly ¾ of a word (so 100 tokens ~= 75 words).

```
spring.application.name=he
spring.ai.openai.api-key=Y
spring.ai.openai.chat.opti
```

Configuration Properties

The prefix `spring.ai.openai.chat` is the property prefix that lets you configure the chat client implementation for OpenAI.

Property	Description	Default
spring.ai.openai.chat.enabled	Enable OpenAI chat client.	true
spring.ai.openai.chat.base-url	Optional overrides the spring.ai.openai.base-url to provide chat specific url	-
spring.ai.openai.chat.api-key	Optional overrides the spring.ai.openai.api-key to provide chat specific api-key	-
spring.ai.openai.chat.options.model	This is the OpenAI Chat model to use	`gpt-3.5-turbo` (the `gpt-3.5-turbo`, `gpt-4`, and `gpt-4-32k` point to the latest model versions)
spring.ai.openai.chat.options.temperature	The sampling temperature to use that controls the apparent creativity of generated completions. Higher values will make output more random while lower values will make results more focused and deterministic. It is not recommended to modify temperature and top_p for the same completions request as the interaction of these two settings is difficult to predict.	0.8
spring.ai.openai.chat.options.frequencyPenalty	Number between -2.0 and 2.0. Positive values penalize new tokens based on their existing frequency in the text so far, decreasing the model's likelihood to repeat the same line verbatim.	0.0f
spring.ai.openai.chat.options.logitBias	Modify the likelihood of specified tokens appearing in the completion.	-
spring.ai.openai.chat.options.maxTokens	The maximum number of tokens to generate in the chat completion. The total length of input tokens and generated tokens is limited by the model's context length.	-
spring.ai.openai.chat.options.n	How many chat completion choices to generate for each input message. Note that you will be charged based on the number of generated tokens across all of the choices. Keep n as 1 to minimize costs.	1
spring.ai.openai.chat.options.presencePenalty	Number between -2.0 and 2.0. Positive values penalize new tokens based on whether they appear in the text so far, increasing the model's likelihood to talk about new topics.	-
spring.ai.openai.chat.options.responseFormat	An object specifying the format that the model must output. Setting to `{ "type": "json_object" }` enables JSON mode, which guarantees the message the model generates is valid JSON.	-

DEMO - GETTING STARTED

PROMPTS

Prompts serve as the foundation for the language-based inputs that guide an AI model to produce specific outputs. For those familiar with ChatGPT, a prompt might seem like merely the text entered into a dialog box that is sent to the API. However, it encompasses much more than that. In many AI Models, the text for the prompt is not just a simple string.

Prompt Engineering
Effective Communication

- Input directing an AI model to produce specific outputs
- Model outputs are greatly inference by prompt style and wording
- Prompt Engineering
  - Prompt techniques and effective prompts are share in the community
  - OpenAI Guidelines
  - Mini Course: ChatGPT Prompt Engineering for developers
- Prompts and Spring
  - Prompt management relies on Text Template Engines
  - Analogous to the view in Spring MVC

```java
public class Prompt implements ModelRequest<List<Message>> {
    private final List<Message> messages;
    private ChatOptions modelOptions;

    public Prompt(String contents) {
        this((Message)(new UserMessage(contents)));
    }

    public Prompt(Message message) {
        this(Collections.singletonList(message));
    }

    public Prompt(List<Message> messages) {
        this.messages = messages;
    }

    public Prompt(String contents, ChatOptions modelOptions) {
        this((Message)(new UserMessage(contents)), modelOptions);
    }

    public Prompt(Message message, ChatOptions modelOptions) {
        this(Collections.singletonList(message), modelOptions);
    }

    public Prompt(List<Message> messages, ChatOptions modelOptions) {
        this.messages = messages;
        this.modelOptions = modelOptions;
    }
```

```java
public class Prompt implements ModelRequest<List<Message>> {
    private final List<Message> messages;
    private ChatOptions modelOptions;

    public Prompt(String contents) {
        this((Message)(new UserMessage(contents)));
    }

    public Prompt(Message message) {
        this(Collections.singletonList(message));
    }
```

Choose Implementation of Message (6 found)

AbstractMessage (org.springframework.ai.chat.messages)	Maven: org.springframework.ai:spring-ai-core:0.8.1 (spring-ai-core-0.8.1.jar)
AssistantMessage (org.springframework.ai.chat.messages)	Maven: org.springframework.ai:spring-ai-core:0.8.1 (spring-ai-core-0.8.1.jar)
ChatMessage (org.springframework.ai.chat.messages)	Maven: org.springframework.ai:spring-ai-core:0.8.1 (spring-ai-core-0.8.1.jar)
FunctionMessage (org.springframework.ai.chat.messages)	Maven: org.springframework.ai:spring-ai-core:0.8.1 (spring-ai-core-0.8.1.jar)
SystemMessage (org.springframework.ai.chat.messages)	Maven: org.springframework.ai:spring-ai-core:0.8.1 (spring-ai-core-0.8.1.jar)
UserMessage (org.springframework.ai.chat.messages)	Maven: org.springframework.ai:spring-ai-core:0.8.1 (spring-ai-core-0.8.1.jar)

```java
    }

    public Prompt(Message message, ChatOptions modelOptions) {
        this(Collections.singletonList(message), modelOptions);
    }

    public Prompt(List<Message> messages, ChatOptions modelOptions) {
        this.messages = messages;
        this.modelOptions = modelOptions;
    }
```

ROLES

- **System Role:** Guides the AI's behavior and response style, setting parameters or rules for how the AI interprets and replies to the input. It's akin to providing instructions to the AI before initiating a conversation.
- **User Role:** Represents the user's input – their questions, commands, or statements to the AI. This role is fundamental as it forms the basis of the AI's response.
- **Assistant Role:** The AI's response to the user's input. More than just an answer or reaction, it's crucial for maintaining the flow of the conversation. By tracking the AI's previous responses (its 'Assistant Role' messages), the system ensures coherent and contextually relevant interactions.
- **Function Role:** This role deals with specific tasks or operations during the conversation. While the System Role sets the AI's overall behavior, the Function Role focuses on carrying out certain actions or commands the user asks for. It's like a special feature in the AI, used when needed to perform specific functions such as calculations, fetching data, or other tasks beyond just talking. This role allows the AI to offer practical help in addition to conversational responses.

DEMO - PROMPTS

STRUCTURED OUTPUT

OUTPUT PARSING

Challenges with handling the response

- The Challenge
  - Output of Generative LLM is a java.util.String
  - Even if you ask for JSON, you get a JSON String
  - ChatGPT wants to chat, not reply in JSON
- OpenAI has introduced a new feature to help with this
- Spring AI’s OutputParser uses refined prompts to desired results

STRUCTURED OUTPUT API

Challenges with handling the response

- The Challenge
  - Output of Generative LLM is a java.util.String
  - Even if you ask for JSON, you get a JSON String
  - ChatGPT wants to chat, not reply in JSON
- OpenAI has introduced a new feature to help with this
- Spring AI’s OutputParser uses refined prompts to desired results

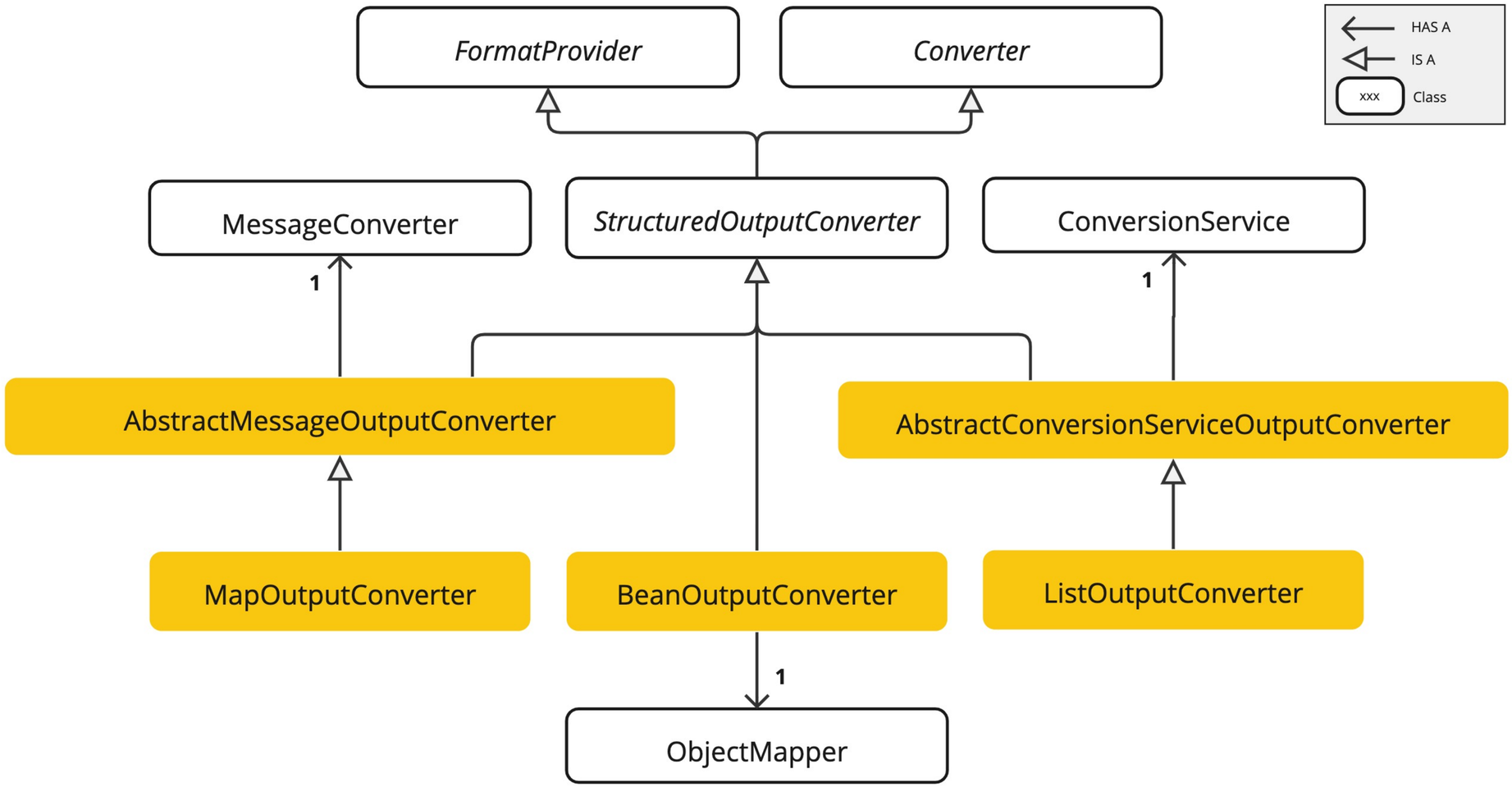
STRUCTURED OUTPUT API

Structured Output Converter Interface

```java
public interface StructuredOutputConverter<T> extends Converter<String, T>, FormatProvider {

   /**
    * @deprecated Use the {@link #convert(Object)} instead.
    */
   default T parse(@NonNull String source) {
      return this.convert(source);
   }

}

public interface FormatProvider {

   /**
    * @return Returns a string containing instructions for how the output of a language
    * generative should be formatted.
    */
   String getFormat();

}
```

STRUCTURED OUTPUT API

Available Converters



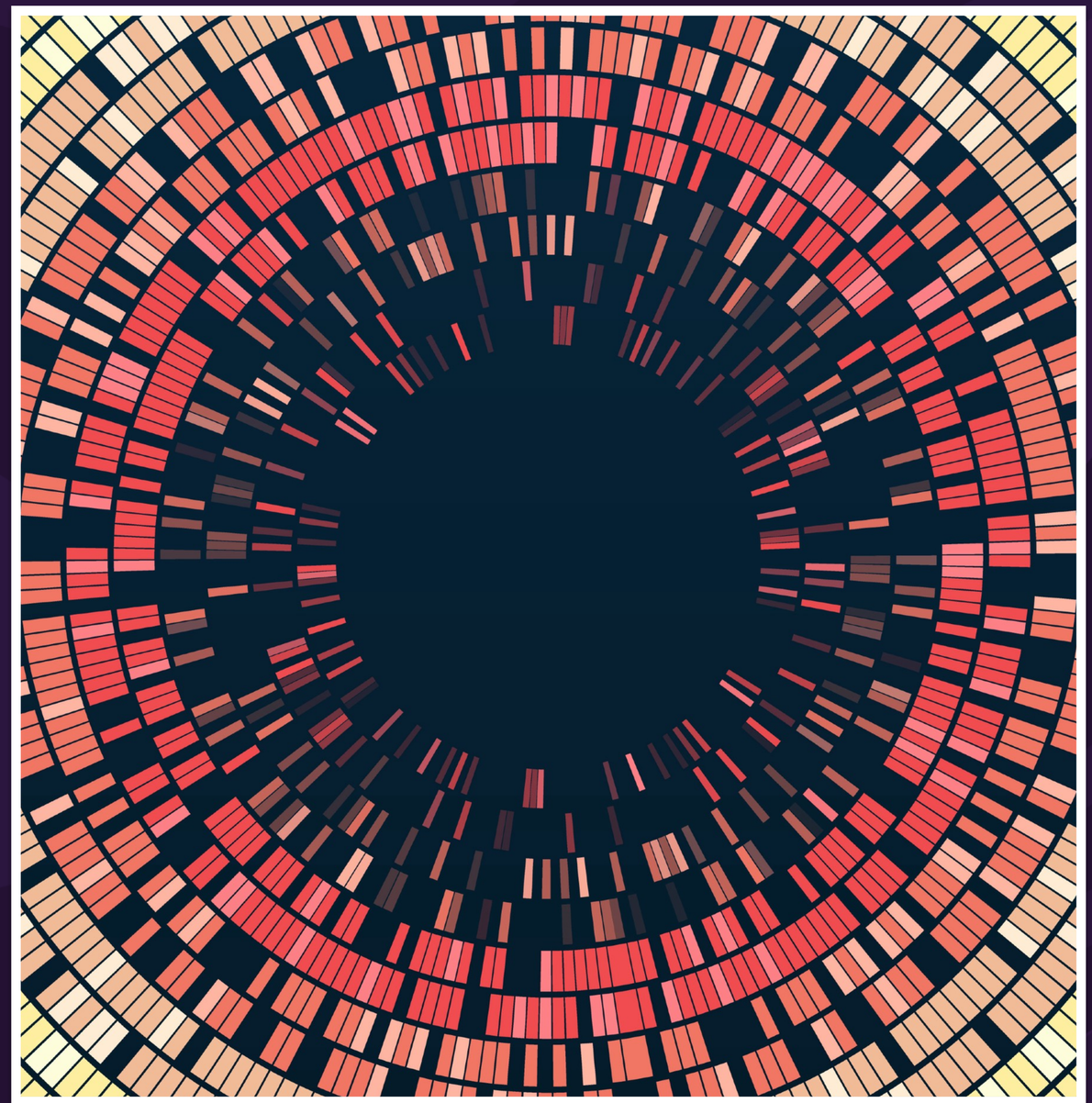
DEMO - OUTPUT PARSER

BRING YOUR OWN DATA

BRING YOUR OWN DATA

How to use your own data in AI Applications

- AI Models have limitations
  - They are trained with public knowledge up to a certain date.
  - They don’t know about your private / corporate data.
- What can we do about this problem?
  - Fine Tune the Model
  - “Stuff the prompt” - add your data into the prompt
  - Function Calling
- Retrieval Augmented Generation (RAG)
  - How to retrieve the relevant data for the user input and add it to your prompt
  - There are many strategies



STUFFING THE PROMPT

What sports are being included in the 2024 Summer Olympics?

```
Use the following pieces of context to answer the question at the end. If you
don't know the answer just say "I'm sorry but I don't know the answer to that".

{context}

Question: {question}
```

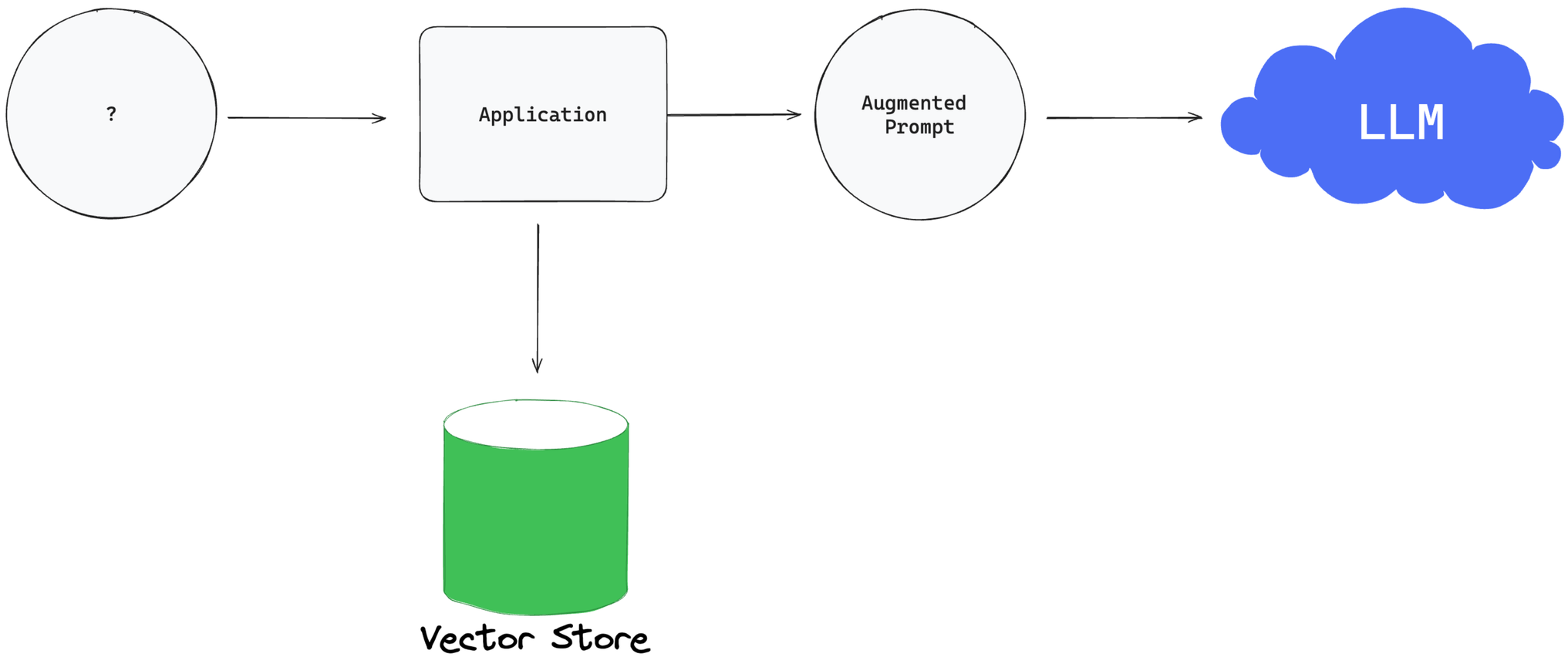
Archery, athletics, badminton, basketball , basketball 3×3, boxing, canoe slalom, canoe sprint, road cycling, cycling track, mountain biking, BMX freestyle, BMX racing, equestrian, fencing, football, golf, artistic gymnastics, rhythmic gymnastics, trampoline, handball, hockey, judo, modern pentathlon, rowing, rugby, sailing, shooting, table tennis, taekwondo, tennis, triathlon, volleyball, beach volleyball, diving, marathon swimming, artistic swimming, swimming, water polo, weightlifting,wrestling,breaking, sport climbing, skateboarding, and surfing.

I don't know the answer to that

Answers Correctly

RAG (RETRIEVAL AUGMENTED GENERATION)

AI APPLICATION ARCHITECTURE - RAG



VECTOR STORES

Not just for text search

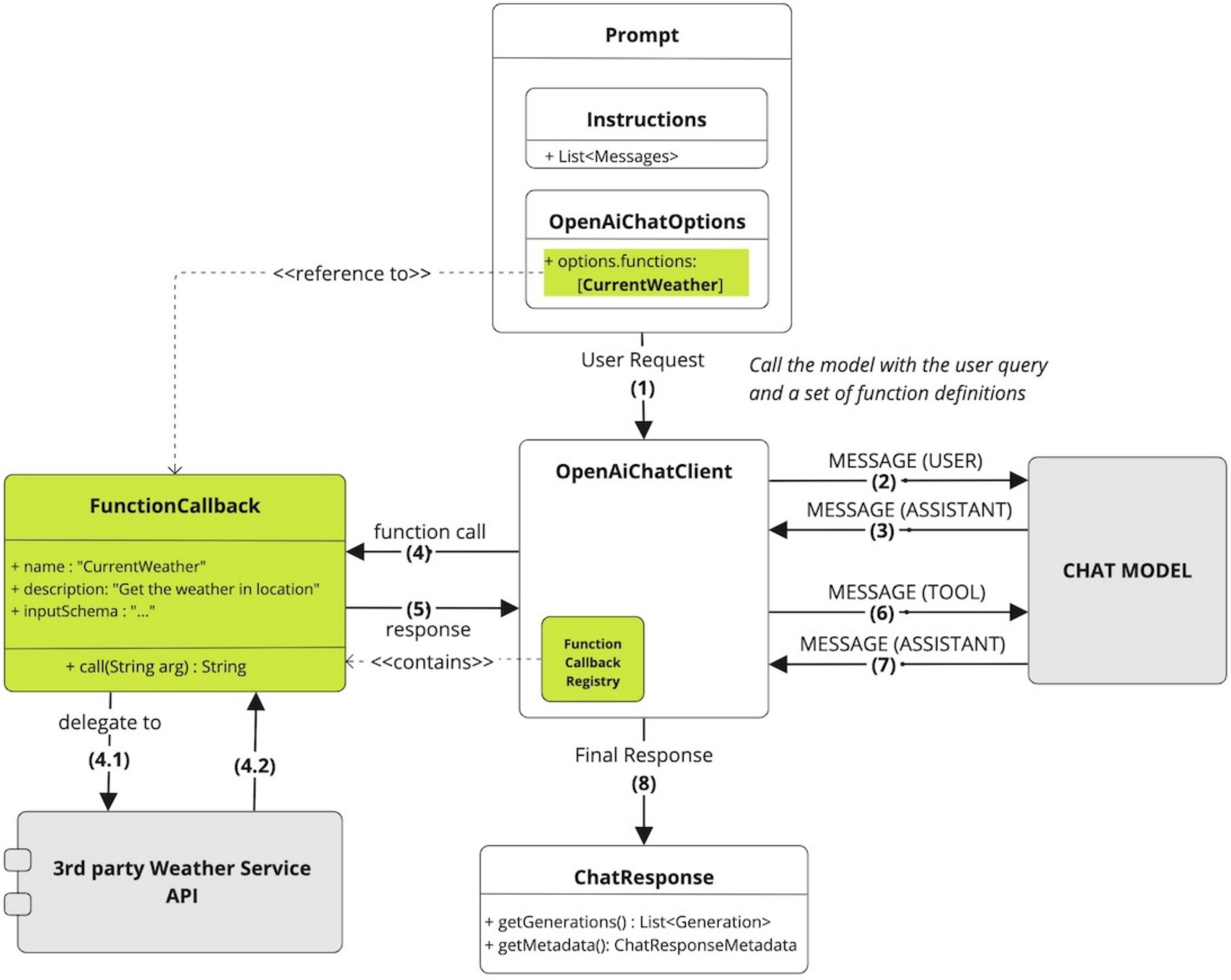
- Azure Vector Search
- ChromaVectorStore
- MilvusVectorStore
- Neo4JVectorStore
- PgVectorStore
- QdrantVectorStore
- RedisVectorStore
- WeaviateVecvtorStore
- SimpleVectorStore

FUNCTION CALLING

FUNCTION CALLING

Registering Custom Functions

- You can register custom Java functions with the **OpenAiChatClient** and have the OpenAI model intelligently choose to output a JSON object containing arguments to call one or many of the registered functions.
- The OpenAI API does not call the function directly; instead, the model generates JSON that you can use to call the function in your code and return the result back to the model to complete the conversation.


Prompt
Instructions
+ List<Messages>
OpenAiChatOptions
+ options.functions:
[CurrentWeather]
<<reference to>>
User Request
(1)
Call the model with the user query
and a set of function definitions
OpenAiChatClient
MESSAGE (USER)
(2)
MESSAGE (ASSISTANT)
(3)
CHAT MODEL
MESSAGE (TOOL)
(6)
MESSAGE (ASSISTANT)
(7)
FunctionCallback
+ name : "CurrentWeather"
+ description: "Get the weather in location"
+ inputSchema : "..."
+ call(String arg) : String
function call
(4)
(5)
response
<<contains>>
Function
Callback
Registry
delegate to
(4.1)
(4.2)
3rd party Weather Service
API
Final Response
(8)
ChatResponse
+ getGenerations() : List<Generation>
+ getMetadata(): ChatResponseMetadata

DEMO - YOUR OWN DATA

CHAT MEMORY

THE WEB IS STATELESS

ChatGPT 4o

Hello, my name is Dan

Memory updated

Hello, Dan! How can I assist you today?

2/2

What is my name?

Your name is Dan. How can I help you today?

Message ChatGPT

ChatGPT can make mistakes. Check important info.

?

DEMO - CHAT MEMORY

REFERENCES

Links & Citations

- Spring AI Reference Documentation
  - https://docs.spring.io/spring-ai/reference/
- Google Course
  - https://www.cloudskillsboost.google/course_templates/536
- Spring Team
  - Mark Pollack
  - Christian Tzolov
  - Craig Walls
  - Josh Long

THANK YOU

dan.vega@broadcom.com

@therealdanvega

https://www.danvega.dev