



spring[®] AI

from exploration to production



Sylvain Puchol
Senior Solution Engineer
VMware Tanzu

Nov 2024

Hello world!



Sylvain Puchol

Solution Engineer at Broadcom | Spring & Tanzu

<https://www.linkedin.com/in/sylvain-puchol/>

Agenda

- AI and LLM basics
- Build a LLM-powered App with Spring AI
- Private AI path to production with Tanzu Platform

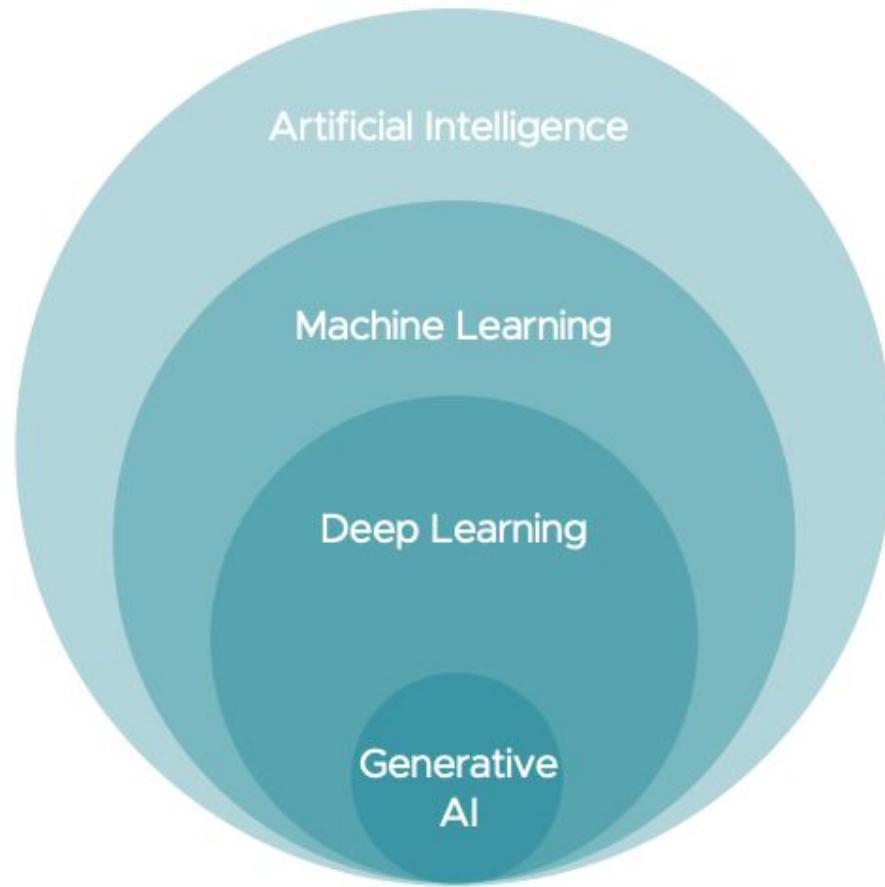


AI and LLM Basics



GenAI and LLMS

Fundamentals



Artificial Intelligence: Machines are capable of performing cognitive functions typically associated with human minds (chatbot, Boston Dynamics)

Machine learning: Algorithms that learn from data to make predictions or decisions without being explicitly programmed (Classification, Data prediction, Finding patterns, Group similar things)

Deep Learning: Algorithms that simulate how the human brain's neurons work. It adds layers (Neural Networks) between input and output data to learn at much depth (Deepfake)

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

GenAI and LLMS

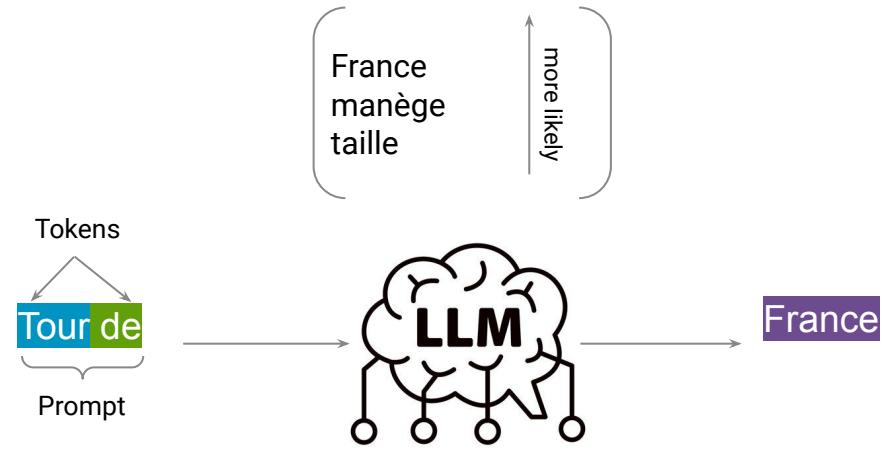
Fundamentals

Machine Learning Model: A mathematical model trained on a specific dataset to make predictions or classifications on new data

Foundation Model: An ML 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

- **Prompts:** Input instructions or data given to the model to guide content generation
- **Tokens:** Basic units of data processed by models, such as words or parts of words in text generation



Google



Using a foundational/public LLM for your Apps

How could it help with your business problems

Strengths

Large general knowledge base

Beats the majority of human level scores in exams

Can align responses based on your goals

Ability to create new original content

Multiple modalities

Instant summarization

Weaknesses

Stateless

Not aware of your API's

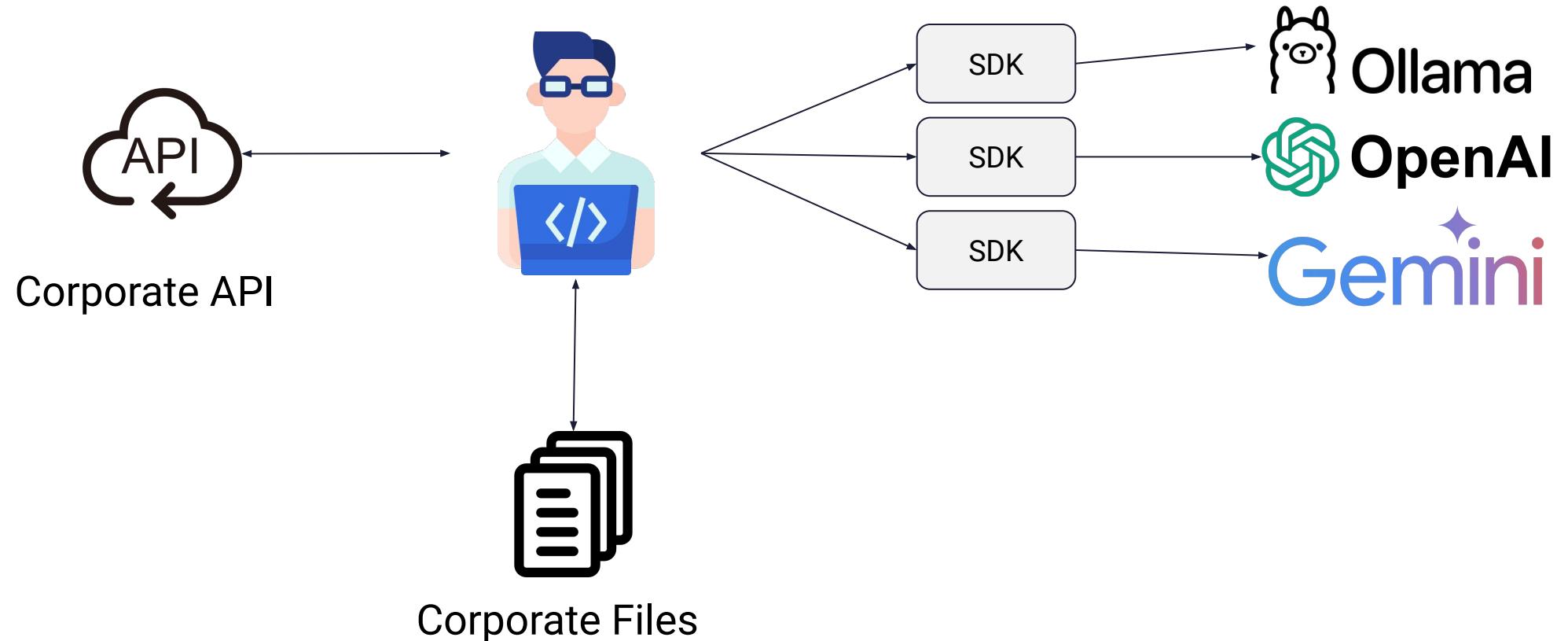
Not trained on your Data

High training costs/time to fill the knowledge gap

No structured output

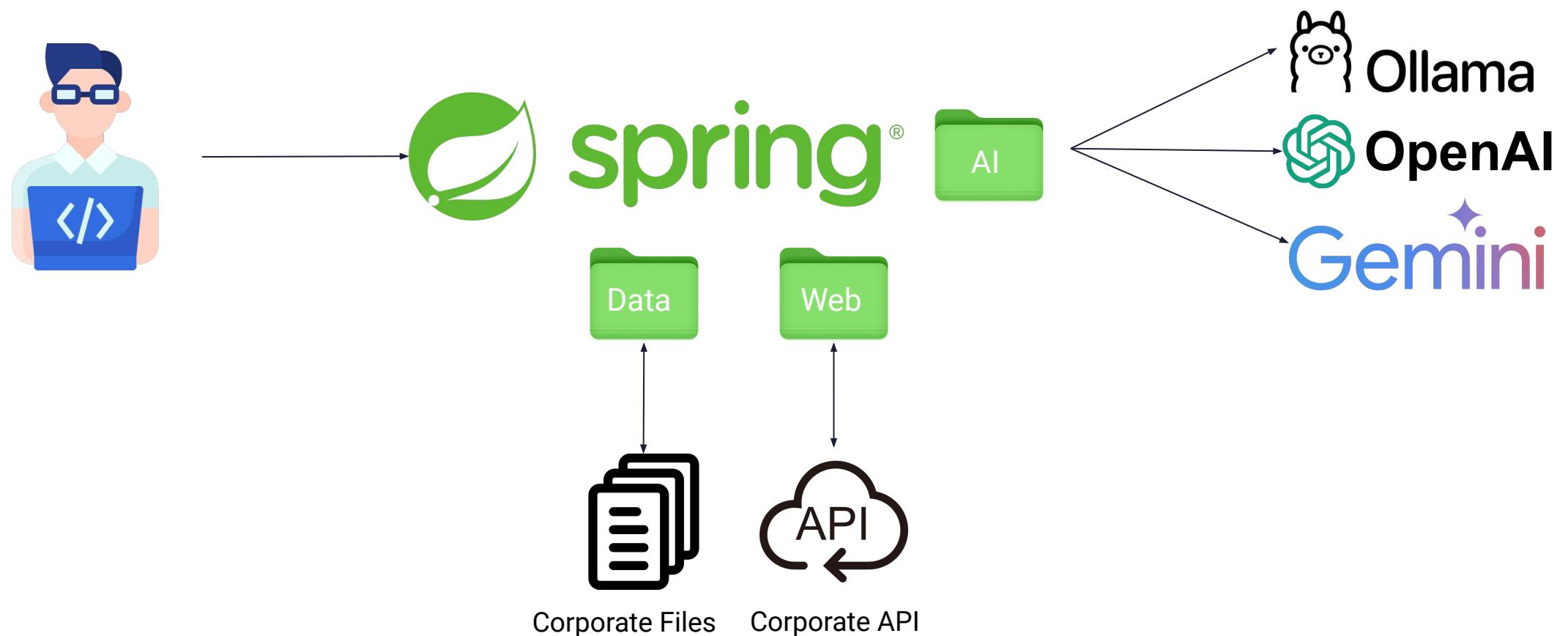
Using a foundational/public LLM on your own

You are the man-in-the-middle between LLM and your Corporate Data



Using Spring AI to cope with LLM challenges

Standardization and simplification with the spring ecosystem



Build a LLM-Powered business application with Spring AI

Recipe Finder powered by AzureOpenAI

Ingredients (comma separated): Prefer available ingredients Prefer own recipes

Roasted Potatoes

Roasted Potatoes with Herbs

Ingredients

- Potatoes
- Olive oil
- Salt
- Black pepper
- Garlic powder
- Dried thyme
- Dried rosemary

Instructions

Preheat the oven to 200°C (400°F).
Wash and scrub the potatoes.
Cut the potatoes into small cubes or wedges.
In a mixing bowl, drizzle the potatoes with olive oil.
Season with salt, black pepper, garlic powder, dried thyme, and dried rosemary. Toss to coat evenly.
Spread the potatoes in a single layer on a baking sheet.
Roast in the preheated oven for about 30-35 minutes or until golden brown and crispy.
Remove from the oven and serve hot.



Challenges and constraints for my business application



Provider locking

I want my app to use any LLM

Align responses to goals

I want an engaging chatbot talking like a chef

Not trained on your data

LLM needs to have access to a pdf cookbook

No structured input/output

User inputs has to be used. A recipe java object is defined

Not aware of your Business Logic

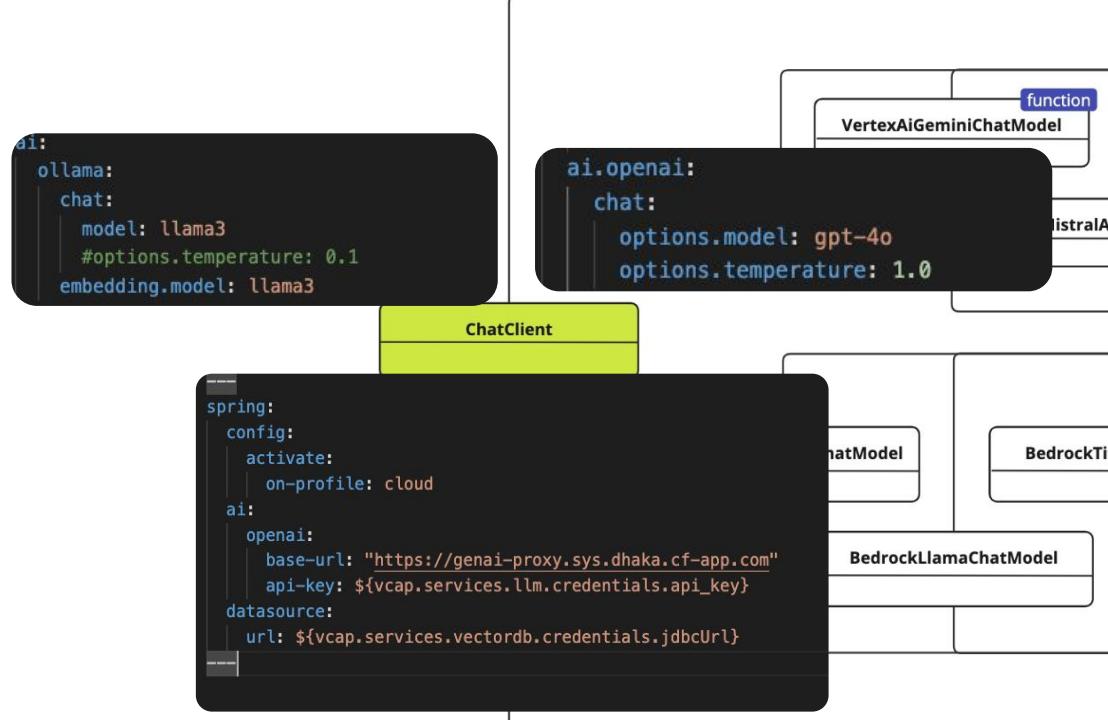
Function giving available ingredients in the kitchen has to be used

Work with more than text

I want the users to be able to scan a dish a get its recipe

Spring AI makes your code portable

No LLM provider locking



Provider	Multimodality	Tools/Functions	Streaming	Retry	Observability	Built-in JSON	Local	OpenAI API Compatibility
Anthropic Claude	text, image	✓	✓	✓	✓	✗	✗	✗
Azure OpenAI	text, image	✓	✓	✓	✓	✓	✗	✓
Google VertexAI Gemini	text, image, audio, video	✓	✓	✓	✓	✓	✗	✓
Google VertexAI PaML2	text	✗	✗	✗	✗	✗	✗	✗
Groq (OpenAI-proxy)	text, image	✓	✓	✓	✓	✗	✗	✓
HuggingFace	text	✗	✗	✗	✗	✗	✗	✗
Mistral AI	text	✓	✓	✓	✓	✓	✗	✓
MiniMax	text	✓	✓	✓	✓	✗	✗	✗
Moonshot AI	text	✗	✓	✓	✓	✗	✗	✗
NVIDIA (OpenAI-proxy)	text, image	✓	✓	✓	✓	✗	✗	✓
Ollama	text, image	✓	✓	✓	✓	✓	✓	✓
OpenAI	text, image	✓	✓	✓	✓	✓	✗	✓
QianFan	text	✗	✓	✓	✓	✗	✗	✗
ZhiPu AI	text	✓	✓	✓	✓	✗	✗	✗
Watsonx.AI	text	✗	✓	✗	✗	✗	✗	✗
Amazon Bedrock/Cohere	text	✗	✓	✗	✗	✗	✗	✗
Amazon Bedrock/Jurassic	text	✗	✗	✗	✗	✗	✗	✗
Amazon Bedrock/Llama	text	✗	✓	✗	✗	✗	✗	✗
Amazon Bedrock/Titan	text	✗	✓	✗	✗	✗	✗	✗
Amazon Bedrock/Anthropic 3	text	✗	✓	✗	✗	✗	✗	✗

Let's solve these LLM challenges with  **spring® AI**

Provider locking

SpringAI API architecture

Align responses to goals

Not trained on your data

No structured input/output

Not aware of your Business Logic

Work with more than text

Easy chatbot through Prompt API

One liner - you don't need to know about the implementation



```
@GetMapping(value = "/hello", produces = MediaType.TEXT_PLAIN_VALUE)
String chat(@RequestParam("q") String query) {
    // A single line API call to connect to your favorite LLM and get a response.
    return chatClient.prompt().user(query).call().content();
}
```

Let's align the model to your goals

System prompt and content streaming



```
public Flux<String> answerCustomerQuestion(String question) {  
    log.info(msg:"Answering question from Chat");  
    var advisorSearchRequest = SearchRequest.query(question).withTopK(topK:1).withSimilarityThreshold(threshold:0.8);  
    var advise = new PromptTemplate(chefCookBook).getTemplate();  
    return chatClient.prompt()  
        .system(chef)  
        .user(question)  
        .advisors(new QuestionAnswerAdvisor(vectorStore, advisorSearchRequest, advise))  
        .stream().content();  
}
```

- **System prompt** gives to the LLM guidances on his persona and behavior
- **Stream** parameter allows us to get tokens on the fly. User feels like he is chatting with somebody that is typing the answer

Let's solve these LLM challenges with spring® AI

Provider locking

SpringAI API architecture

Align responses to goals

System prompt, streaming

Not trained on your data

No structured input/output

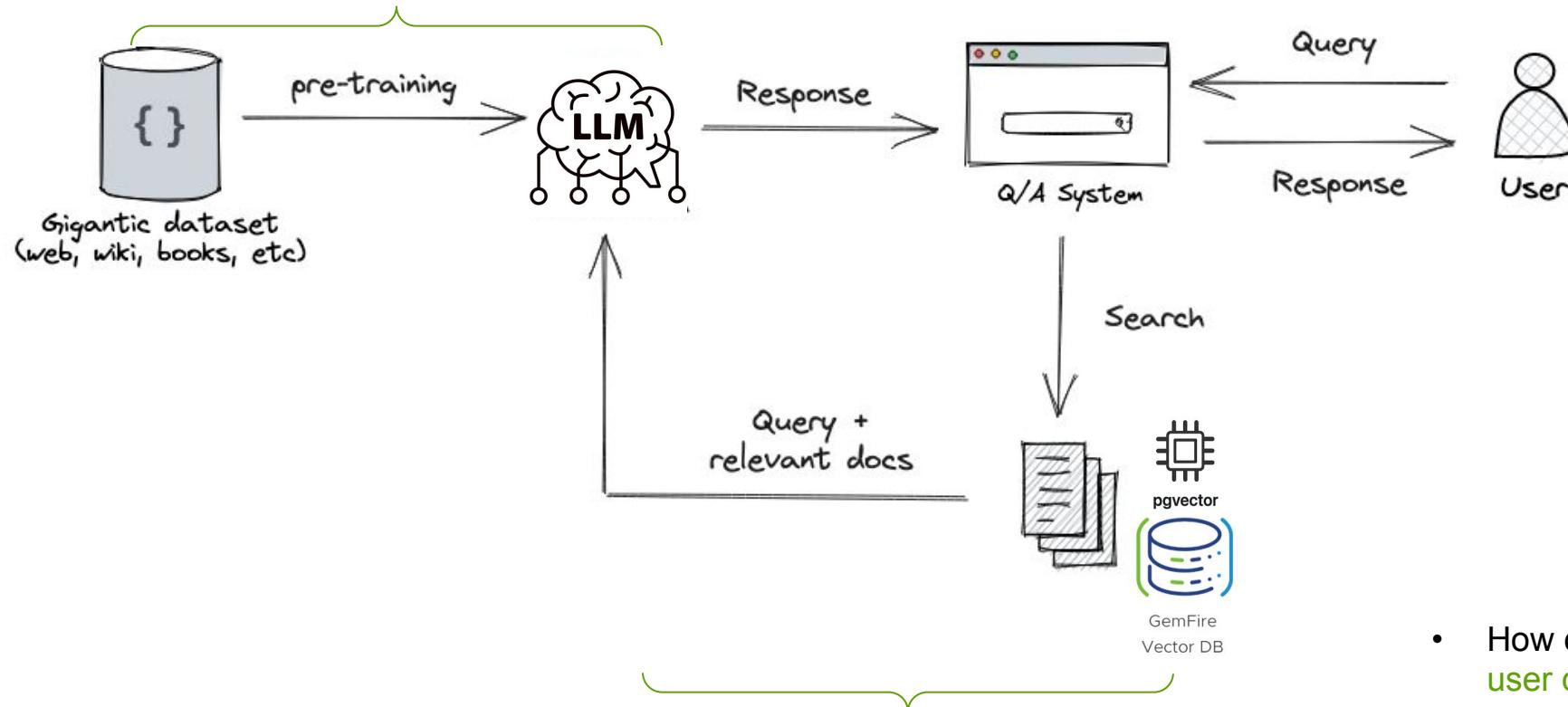
Not aware of your Business Logic

Work with more than text

Give LLM access to your Data

Retrieval augmented Generation: LLM knowledge without additional training

1. Let the LLM provider train a model with generic skills



2. Provide a meaningful context to this model when you call it

- How can I ensure to give data related to the user query?
- How can I avoid to slow down the overall process if my dataset is gigantic and unevenly structured?



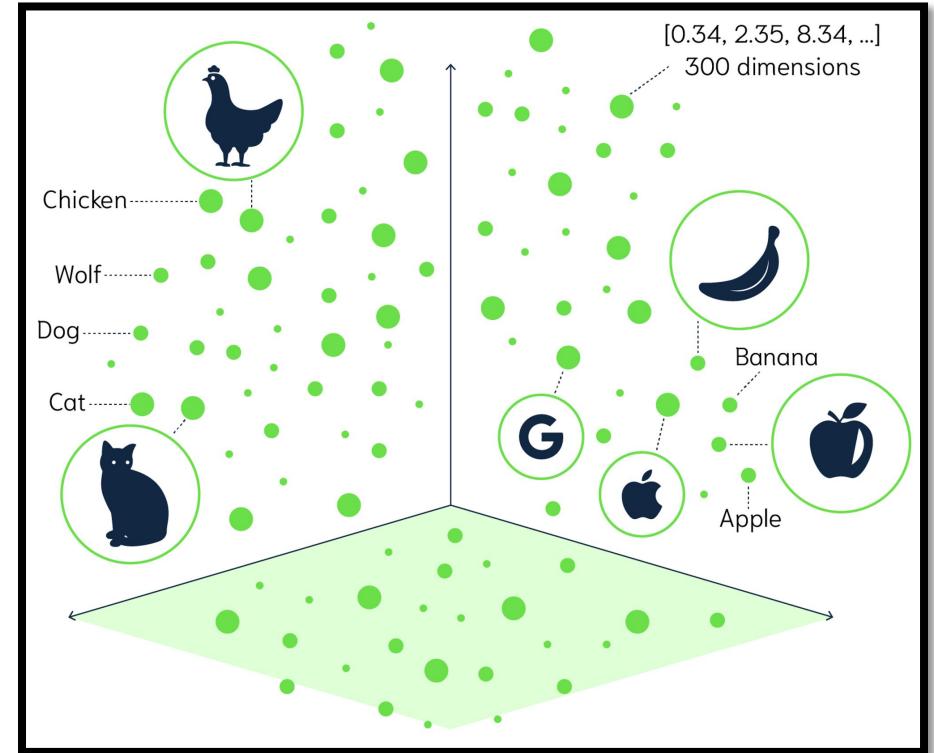
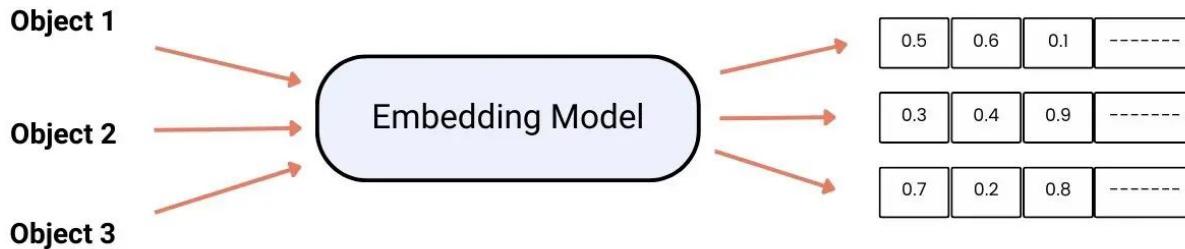
Give LLM access to your Data

A zoom on vector Databases and quantization



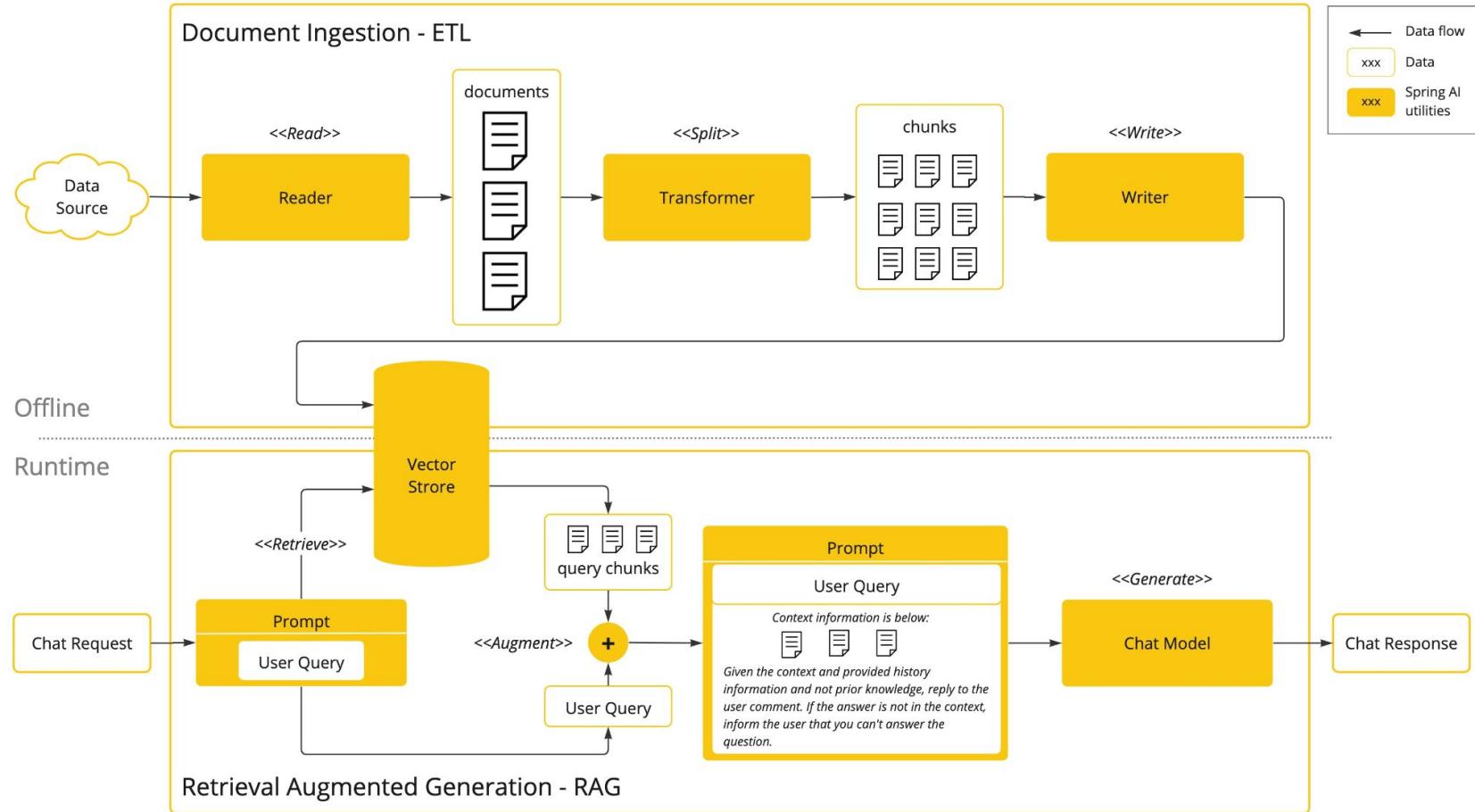
Vector Embeddings enable an LLM to understand the meaning and context of data.

- Quantization process changes data into Vector (list) of floating-point numbers
- Distance between two vectors measures their relatedness (SimilaritySearch)



Spring AI makes RAG easy

SpringAI manages your ETL and vector stores



Spring AI provides:

- A **Vector Store API**, enabling portability across different vector database providers
- **ETL**(Extract Transform and Load) framework to populate the vector database with embeddings
- A **Chat/Embeddings Model API** with support for several models

Spring AI makes RAG easy

Give LLM data to perform RAG, along with behavior guidance

```
public Flux<String> answerCustomerQuestion(String question) {  
    log.info(msg:"Answering question from Chat");  
    var advisorSearchRequest = SearchRequest.query(question).withTopK(topK:1).withSimilarityThreshold(threshold:0.8);  
    var advise = new PromptTemplate(chefCookBook).getTemplate();  
    return chatClient.prompt()  
        .system(chef)  
        .user(question)  
        .advisors(new QuestionAnswerAdvisor(vectorStore, advisorSearchRequest, advise))  
        .stream().content();  
}
```



- **advisorSearchRequest** controls parameters of similar data retrieval on vectorDB
- **Advise** parameter provides guidance on what to do with retrieved data

Let's solve these LLM challenges with spring® AI

Provider locking

SpringAI API architecture

Align responses to goals

System Prompt

Not trained on your data

Spring ETL and Advisors for RAG

No structured input/output

Not aware of your Business Logic

Work with more than text

Prompt templating

Build dynamic prompts based on your users input

Prompt template

```
Provide a recipe that includes in the best case all of the following ingredients.  
  
Ingredients: """"  
{ingredients}  
"""  
  
Add additional ingredients that are necessary for a good flavor or to create a more creative and complex meal.  
  
The recipe should be translated to English, and with quantity in metric system.
```

Prompt rendered from parameters

```
@Value("classpath:/prompts/recipe-for-ingredients")  
private Resource recipeForIngredientsPromptResource;  
  
  
return chatClient.prompt()  
    .user(us -> us  
        .text(recipeForIngredientsPromptResource)  
        .param("ingredients", String.join(", ", ingredients)))  
    .call()  
    .content();
```

Response formatting

Ask your LLM to generate a structured response - and let Spring AI maps it to a Java construct

```
public record Recipe(String name, String description, List<String> ingredients, List<String> instructions, String imageUrl) {  
    public Recipe(Recipe recipe, String imageUrl) {  
        this(recipe.name, recipe.description, recipe.ingredients, recipe.instructions, imageUrl);  
    }  
}
```

```
private Recipe fetchRecipeFor(List<String> ingredients) {  
    log.info(msg:"Fetch recipe without additional information");  
  
    return chatClient.prompt()  
        .user(us -> us  
            .text(recipeForIngredientsPromptResource)  
            .param(k:"ingredients", String.join(delimiter:",", ingredients)))  
        .call()  
        .entity(type:Recipe.class);  
}
```



Let's solve these LLM challenges with spring® AI

Provider locking

SpringAI API architecture

Align responses to goals

System Prompt

Not trained on your data

Spring ETL and advisors for RAG

No structured input/output

Prompt templating / Output Converters

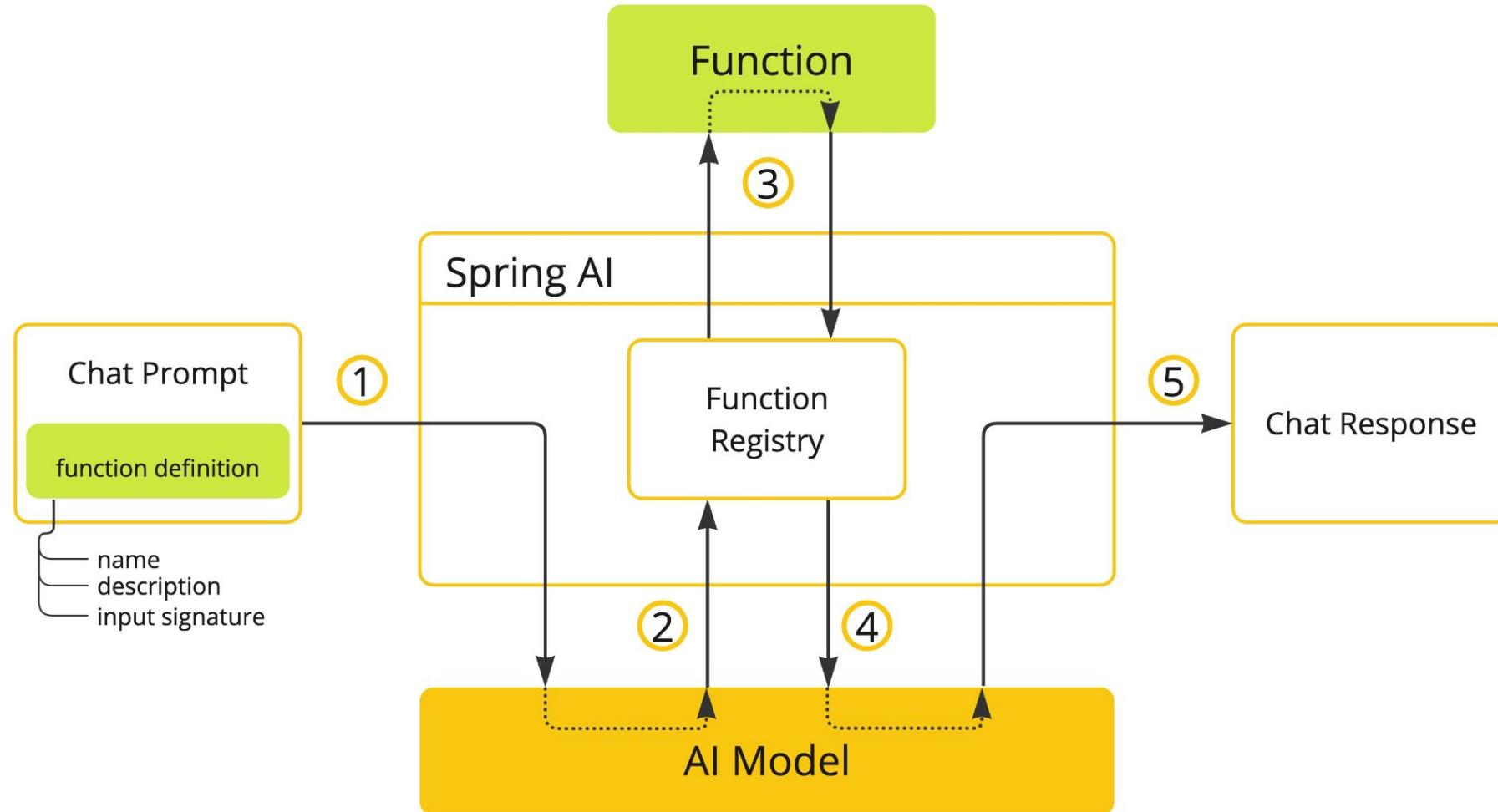
Not aware of your Business Logic

Work with more than text

Function API



Let the LLM call your Java methods as needed



Function API in action

Let the LLM call your Java methods as needed



Describe function for LLM guidance

```
@Description("Fetches ingredients that are available at home")
@Service("fetchingredientsAvailableAtHome")
public class FetchIngredientsAvailableAtHomeFunction implements Function<FetchIngredientsAvailableAtHomeFunction.Request, FetchIngredientsAvailableAtHomeFunction.Response> {

    private static final Logger log = LoggerFactory.getLogger(clazz:FetchIngredientsAvailableAtHomeFunction.class);

    private final List<String> alwaysAvailableIngredients;
    private final List<String> availableIngredientsInFridge;

    public FetchIngredientsAvailableAtHomeFunction(@Value("${app.always-available-ingredients}") List<String> alwaysAvailableIngredients,
                                                   @Value("${app.available-ingredients-in-fridge}") List<String> availableIngredientsInFridge) {
        this.alwaysAvailableIngredients = alwaysAvailableIngredients;
        this.availableIngredientsInFridge = availableIngredientsInFridge;
    }
}
```

Use function bean in the call

Provide a recipe that includes in the best case all of the following ingredients plus the **available ingredients at home** which are ordered by highest priority.

Ingredients: """"
{ingredients}
"""

Add additional ingredients that are necessary for a good flavor or to create a more creative and complex meal.

The recipe should be translated to English, and with quantity in metric system.

```
return chatClient.prompt()
    .user(us -> us)
    .text(recipeForAvailableIngredientsPromptResource)
    .param(k:"ingredients", String.join(delimiter:",", ingredients))
    .functions(...functionBeanNames:"fetchIngredientsAvailableAtHome")
    .call()
    .entity(type:Recipe.class);
```

Let's solve these LLM challenges with **spring**[®] AI

Provider locking

SpringAI API architecture

Align responses to goals

System Prompt

Not trained on your data

Spring ETL and advisors for RAG

No structured input/output

Prompt templating / Output Converters

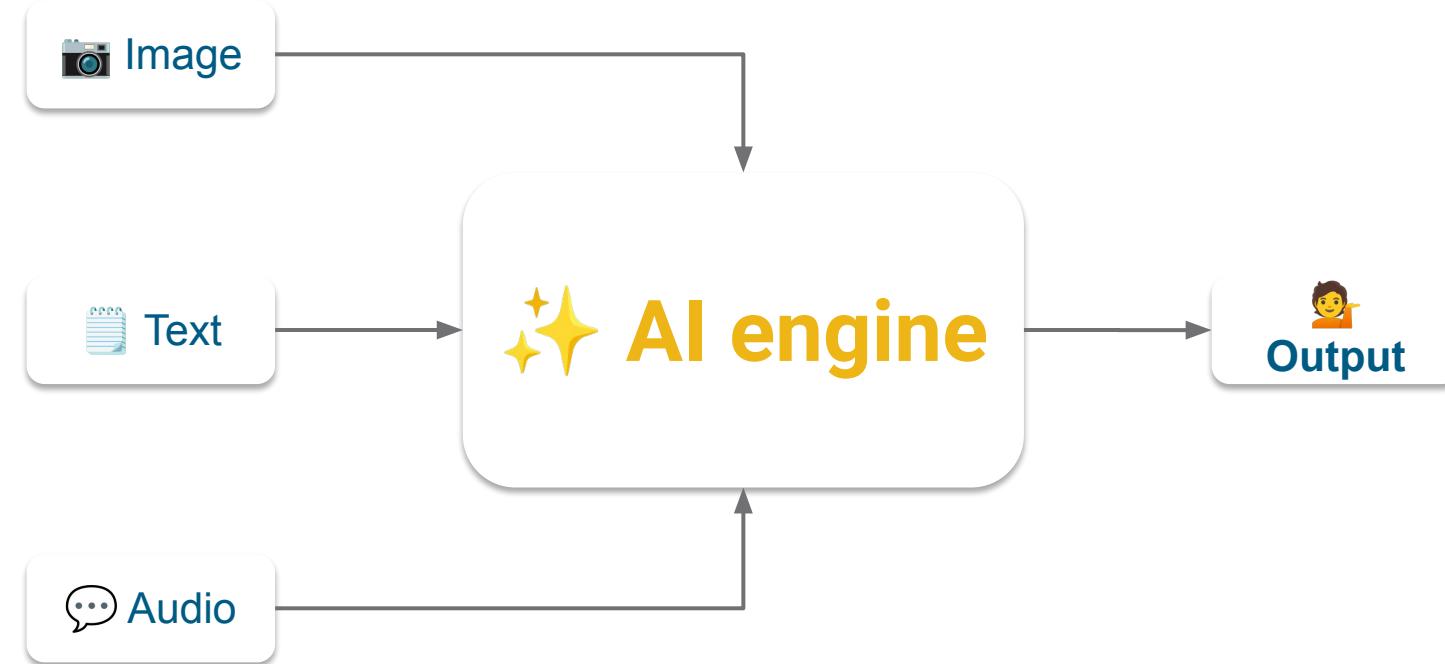
Not aware of your Business Logic

Function Calling

Work with more than text

Prompt is not only text

Process and generate text in conjunction with other modalities such as images, audio, or video



Build a multimodal prompt with Spring AI

Include an image to provide context to your text based prompt



```
return chatClient.prompt()  
    .user(us -> {  
        try {  
            us  
                .text("""  
                    The following image is a dish,  
                    you have to find out what is this dish and then provide the recipe for this dish  
                    """)  
            .media(MediaTypeUtils.IMAGE_JPEG, new URI(ingredients.getFirst().toURL()));  
        } catch (IOException e) {  
            throw new RuntimeException(e);  
        }  
    });
```

Let's solve these LLM challenges with **spring[®] AI**

Provider locking

SpringAI API architecture

Align responses to goals

System Prompt

Not trained on your data

Spring ETL and advisors for RAG

No structured input/output

Prompt templating / Output Converters

Not aware of your Business Logic

Function Calling

Work with more than text

Multimodal prompt

Private AI path to production for my application



What do you need to run LLM powered app in production?

Requirements checklist

Container



Ship your application anywhere

CI/CD



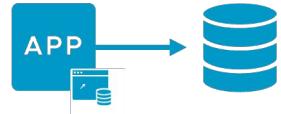
Being to deploy and update your app in minutes

Fault Tolerance



Replicas an http routing

RAG DB



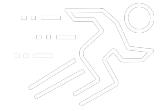
Provide Context to your LLM

GPU powered LLM



Tanzu Platform: SpringAI App lifecycle made easy

A CF push like experience on top of a GPUready cloud



Build



Containerize apps,
Scalable on VMs & K8s
Runtimes

Deploy



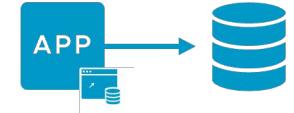
Apps instrumented with
Platform Capabilities

Scale



API Based Automation
for Apps Operations

Bind



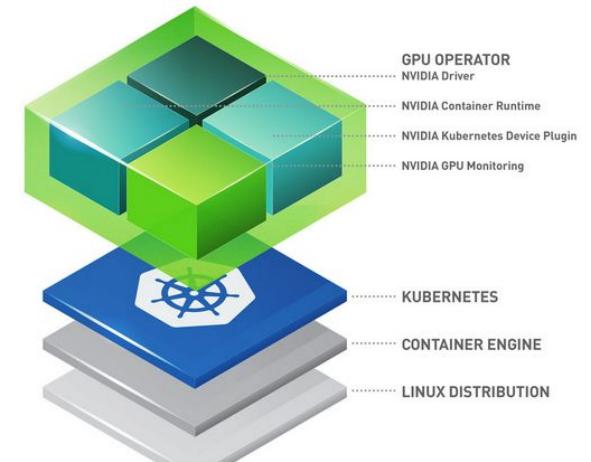
Bindings to DBaaS,
OSS/Cloud Catalogs



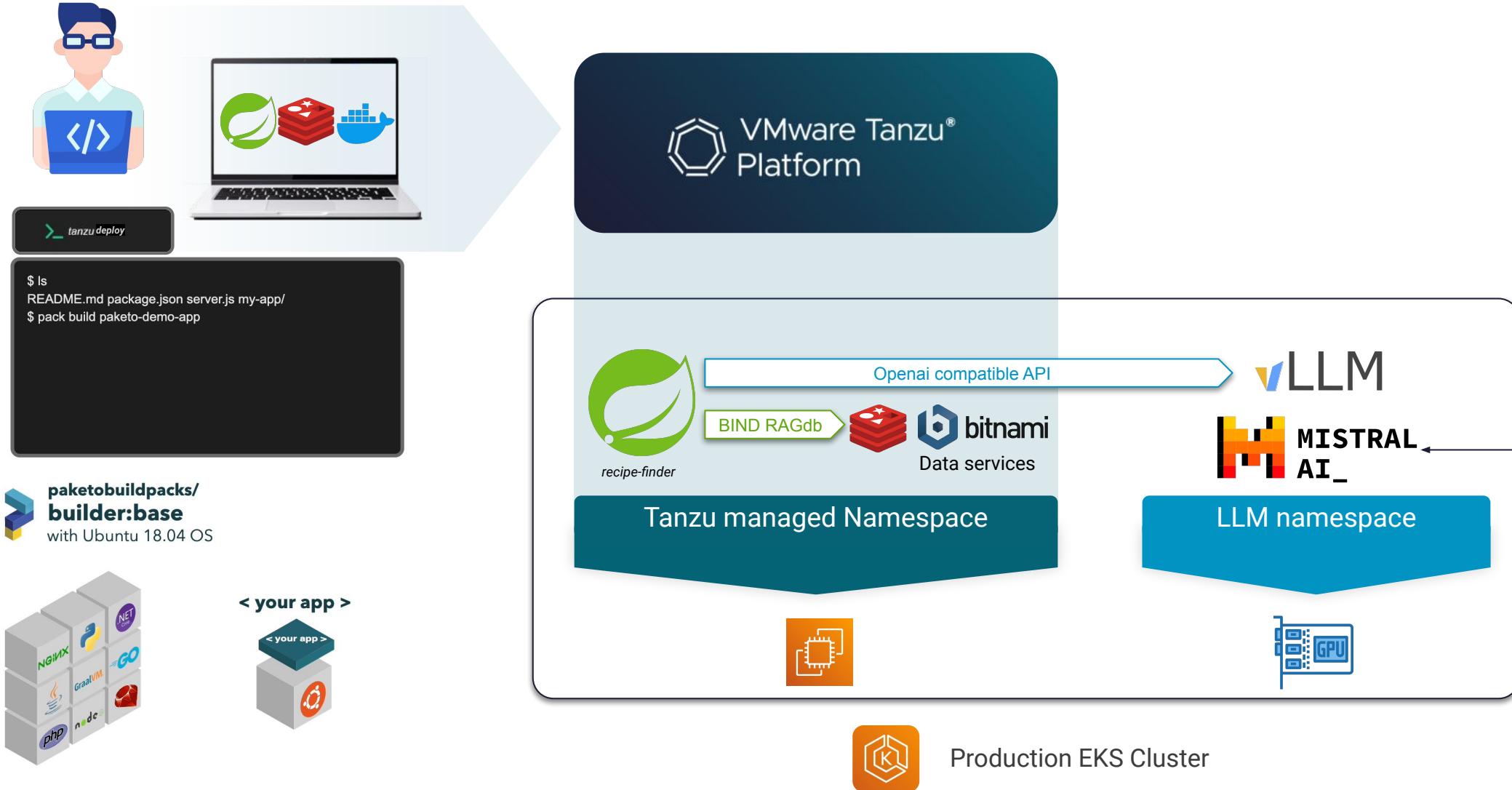
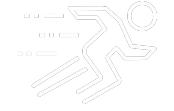
Hugging Face

vmware®
by Broadcom

Broadcom Proprietary and Confidential. Copyright © 2024 Broadc...
All Rights Reserved. The term "Broadcom" refers to Broadcom Inc. and



Private AI path to production Demo: details



VMware Private AI reference architecture

Deepdive on training and inference on any k8s

A screenshot of a laptop displaying a website article. The header of the article reads "Deploying Enterprise-Ready Generative AI on VMware Private AI". Below the title are sections for "Introduction", "Executive Summary", and a detailed description of the platform's benefits. On the left sidebar, there is a navigation menu with links like "Introduction", "Architecture Design", "Deployment and Configuration", "Running LLM Tasks on vSphere with Tanzu Kubernetes", "Conclusion", "References", "About the Authors", "Feedback", and "Appendix".

Start ▾ Solutions ▾ Blog Advanced Search ▾

Community Liked Not Rated

Deploying Enterprise-Ready Generative AI on VMware Private AI

Introduction

Executive Summary

Generative artificial intelligence (GenAI), especially in the form of Large Language Models (LLMs), is at the forefront of technological innovation, offering human-like creativity, reasoning, and language understanding. Organizations across the globe recognize its potential, but implementing LLMs, particularly in regulated industries, brings about unique challenges. On-premises deployment in the private cloud offers a strategic solution, allowing organizations to retain complete control over data and comply with industry regulations. This fosters trust and transparency, ensuring that sensitive information and intellectual property are securely protected within enterprise environments.

IT organizations now can use VMware Private AI platform for running GenAI models. This platform for AI services enables privacy and control of corporate data, choice of open source and commercial AI solutions, quick time-to-value, and integrated security and management.

Utilizing VMware Private AI, we can democratize GenAI by igniting business innovation for all enterprises and providing the following advantages:

- Get the flexibility to run a range of AI solutions for your environment: NVIDIA, open-source, and independent software vendors.
- Deploy with confidence, knowing that VMware has partnerships with NVIDIA and other partners, all of whom are respected leaders in the high-tech space
- Achieve great performance in your model with vSphere and VMware Cloud Foundation's GPU integrations.
- Augment productivity by building private chatbots, eliminating redundant tasks, and building intelligent process

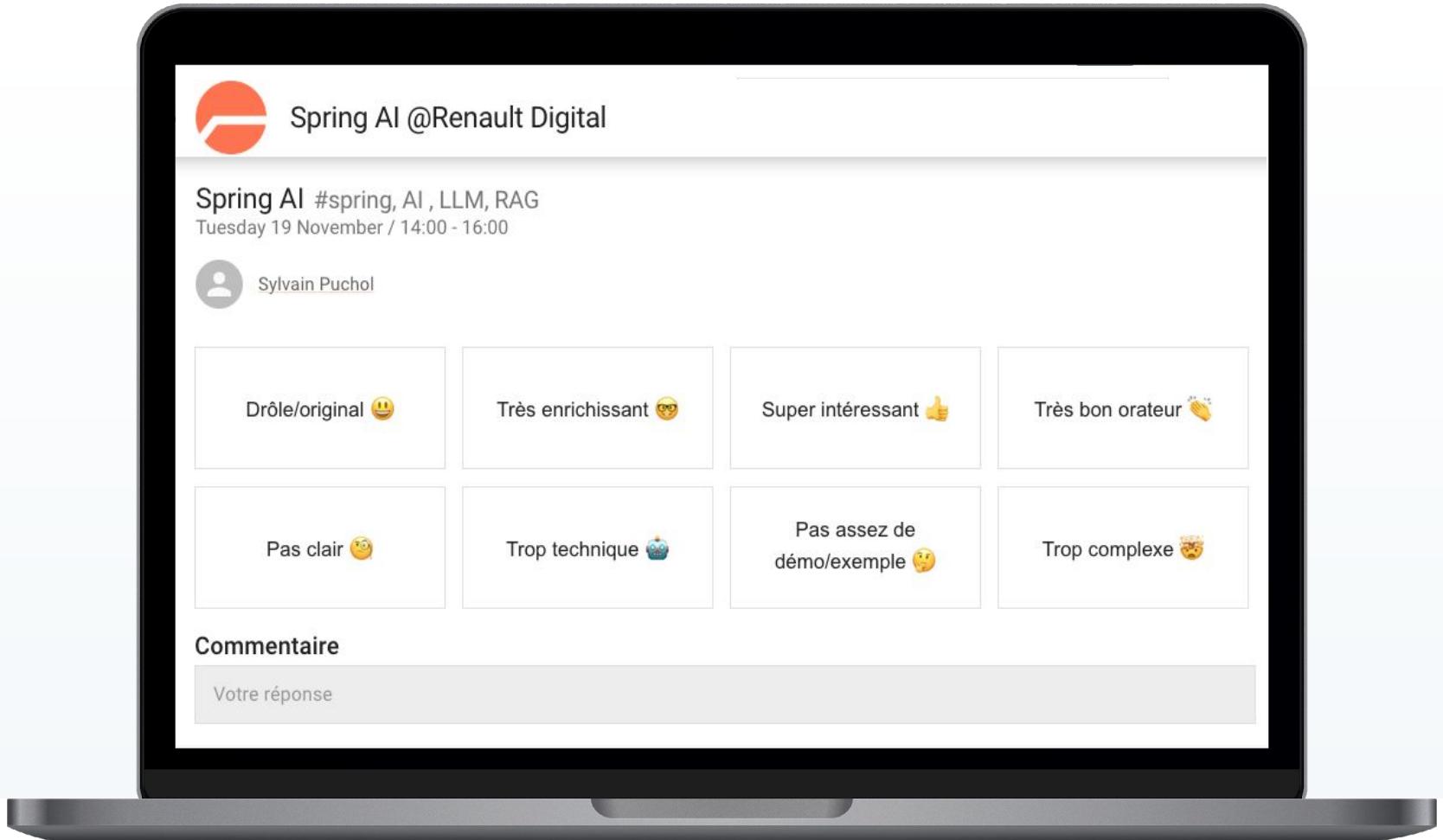
Tanzu platform overview

Build bind an scale your AI apps to any k8s



A screenshot of a laptop displaying the VMware Docs website for the Tanzu Platform. The page title is "VMware Tanzu Platform". On the left, there's a sidebar with a "Expand All" button and several documentation links: "VMware Tanzu Platform for Cloud Foundry Documentation", "Creating and managing applications with Tanzu Platform for Kubernetes", "Using and Managing VMware Tanzu Platform Hub", "VMware Tanzu Application Catalog Documentation", "VMware Tanzu Spring Documentation", and "Using Tanzu Platform cloud services console". Below this is a section titled "VMware Tanzu Data Services" with a link to "VMware Tanzu for MySQL Documentation". At the bottom of the sidebar is a link to "VMware Tanzu for Postgres Documentation". The main content area features a large box titled "VMware Tanzu Platform" with sections for "Tanzu Platform Runtimes" (listing "Tanzu Platform for Cloud Foundry" and "Tanzu Platform for Kubernetes"), "Tanzu Data Services" (listing "Tanzu RabbitMQ", "Tanzu for MySQL", "Postgres", and "Redis"), "Tanzu Application Catalog", "Tanzu Spring", and "Tanzu Salt". A callout at the bottom states: "Tanzu Platform provides an application development framework with Tanzu Spring, a collection of ready-made". On the right side of the slide, there's a vertical bar with a globe icon and "EN", a "Send Feedback" button, and a "In this article" section with a "Tanzu Data Services in Tanzu Platform" link.

#opentofeedback



Thank You

Let's keep in touch:



Sylvain Puchol

<https://www.linkedin.com/in/sylvain-puchol/>