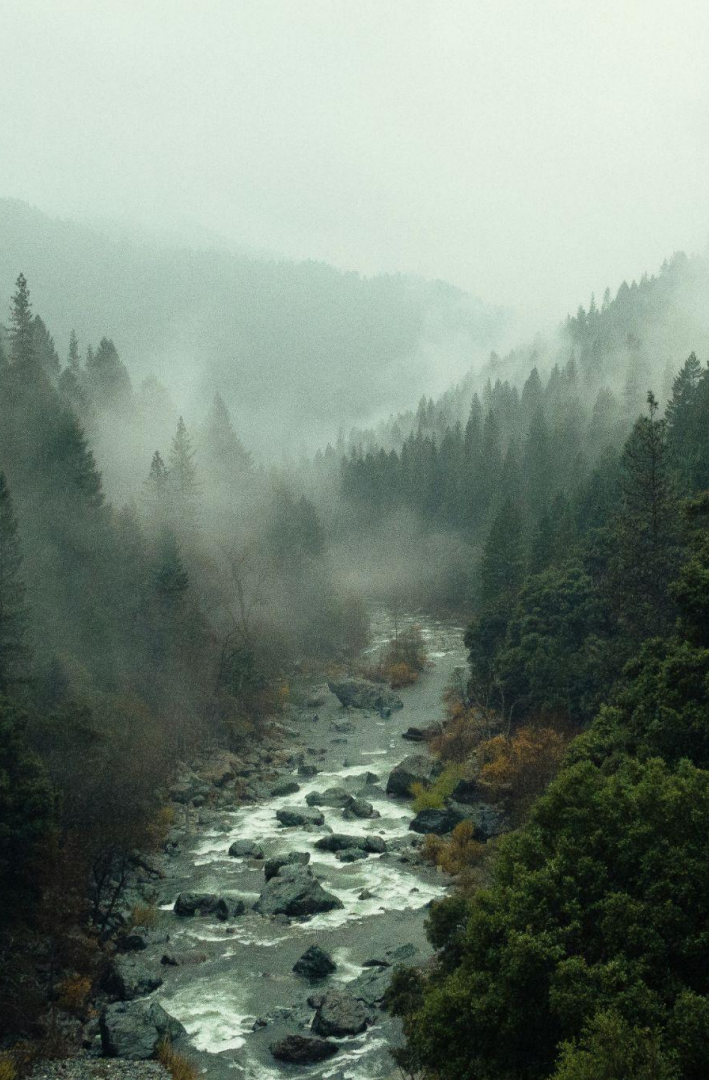
Learning GenAI/LLM Development with langchain4j and Testcontainers

Who’s who of AI

- Foundational AI teams
- AI/ML teams training, fine-tuning models
- Tooling ecosystem vendors
- Application developers



Tooling Ecosystem

- AI products: ChatGPT, Docker AI, Github Copilot, Midjourney,
- Vector databases: ChromaDB, Weaviate, Quadrant
- Runtimes: Ollama, Docker, SageMaker
- Tooling: Ragas, promptfoo
- Libraries: Langchain, SpringAI, Haystack

Hello there!

Oleg Šelajev

Developer advocate, Docker

Brief and incomplete history of LLMs

2017
"Attention is all you need" paper published by Google -> Transformer architecture

2018/2019/2020
GPT-1/2/3 released by OpenAI

2022
ChatGPT is released by OpenAI and triggers LLM hype in the media

Since 2022
OSS and Source-Available models are released: LLaMA, Mistral, etc.

Brief and incomplete history of LLMs

2023
GitHub rebrands as AI-driven development platform

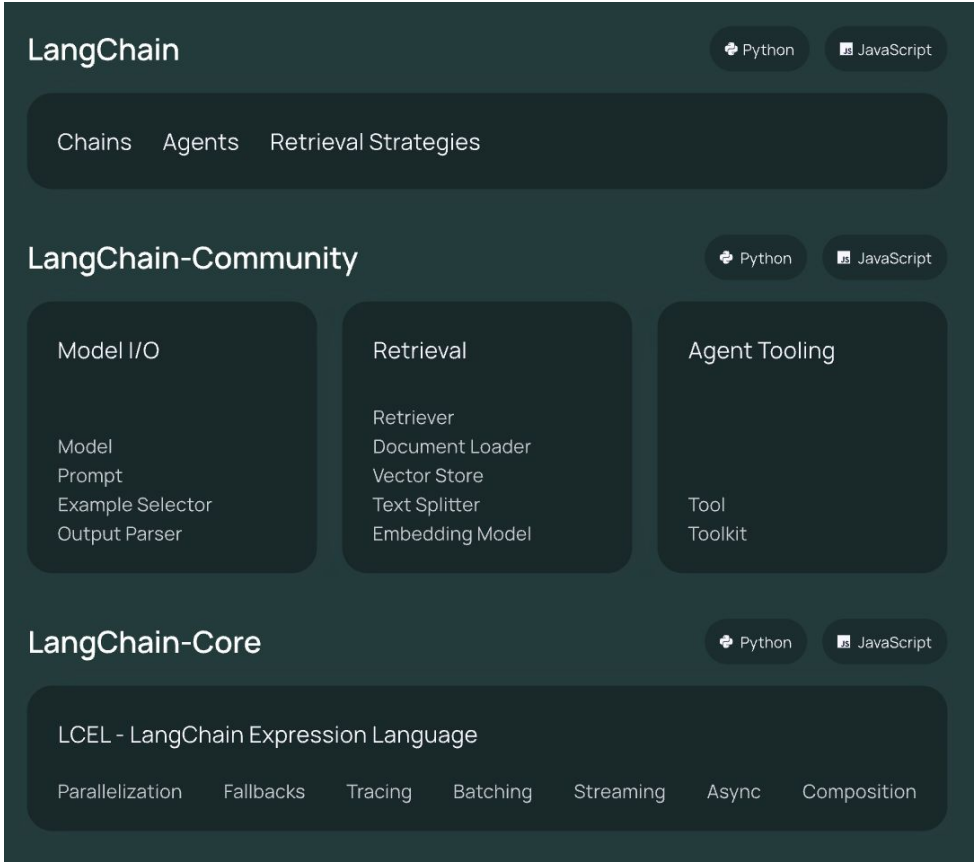
2024
Microsoft announces Copilot PC+Recall

2024
Apple announces Apple Intelligence

2024
Everyone builds AI powered products

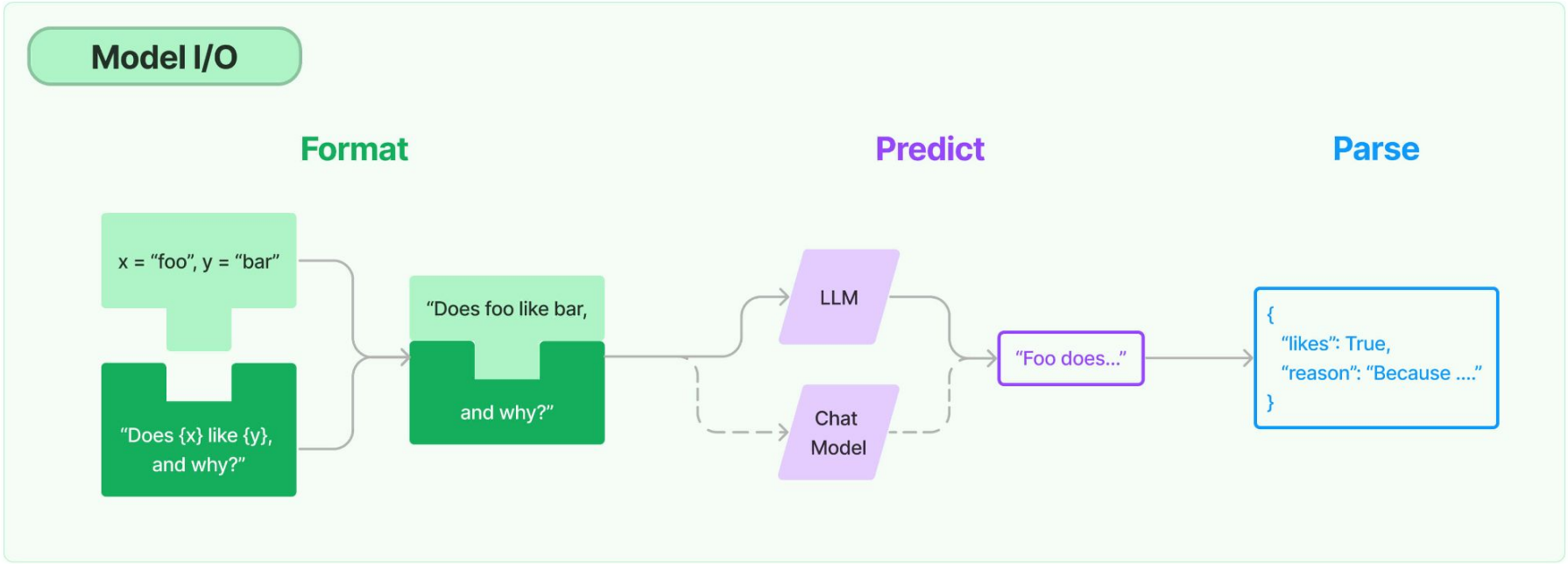
How to develop a LLM app?

Langchain Framework



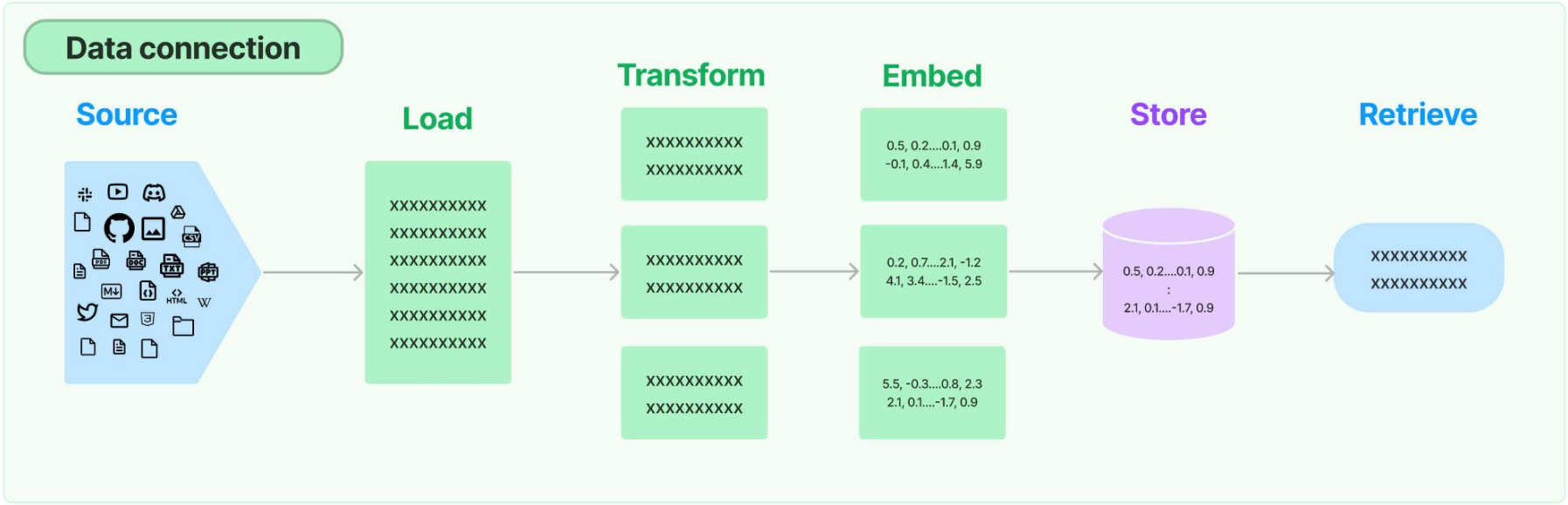
https://python.langchain.com/docs/get_started/introduction

Model I/O



https://python.langchain.com/docs/modules/model_io/

Retrieval Augmented Generation (RAG)



https://python.langchain.com/docs/modules/data_connection/

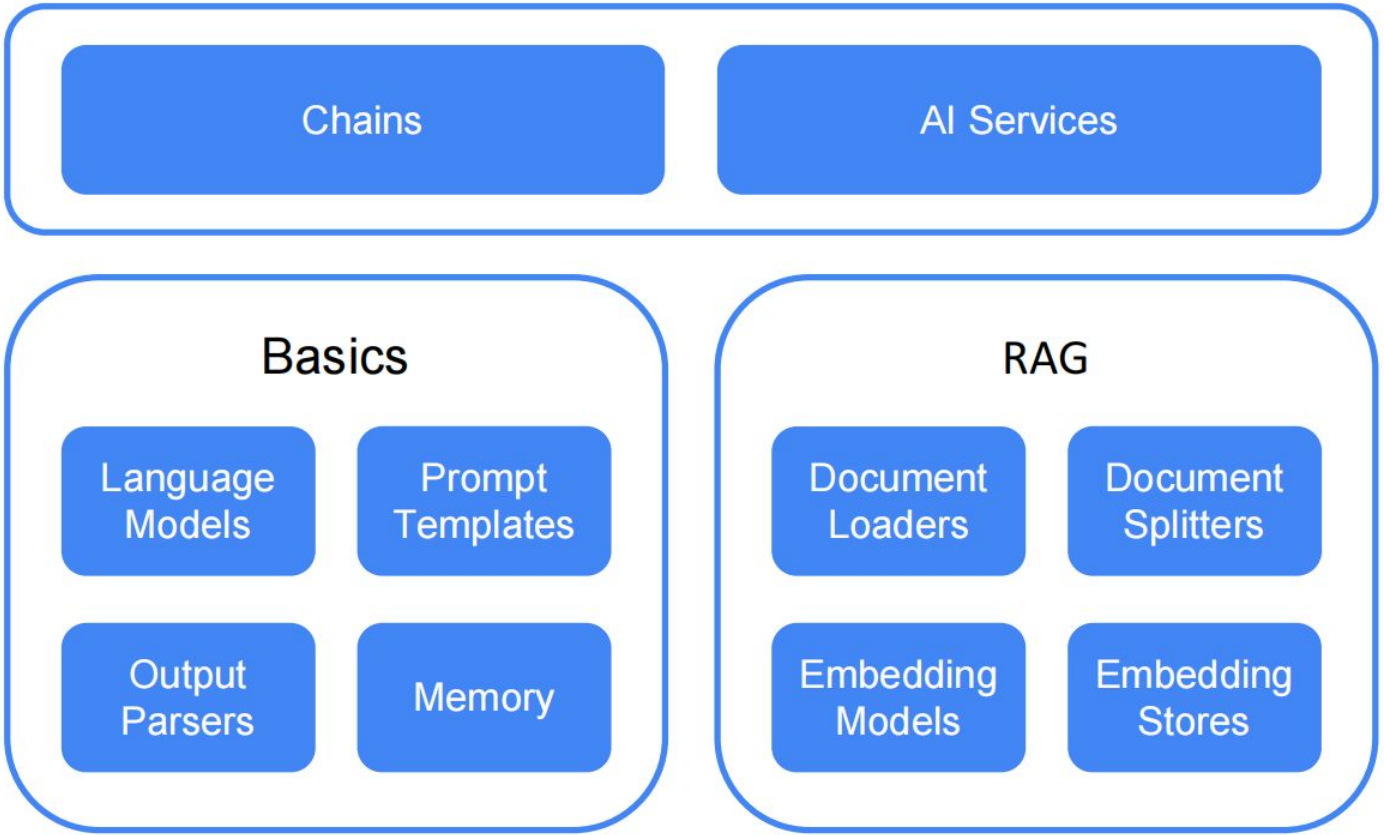
But I am an enterprise Java developer :(

langchain4j

“LangChain4j began development in early 2023 amid the ChatGPT hype. We noticed a lack of Java counterparts to the numerous Python and JavaScript LLM libraries and frameworks, and we had to fix that! Although "LangChain" is in our name, the project is a fusion of ideas and concepts from LangChain, Haystack, LlamaIndex, and the broader community, spiced up with a touch of our own innovation.”

https://docs.langchain4j.dev/

Langchain4j Components



https://docs.langchain4j.dev/intro

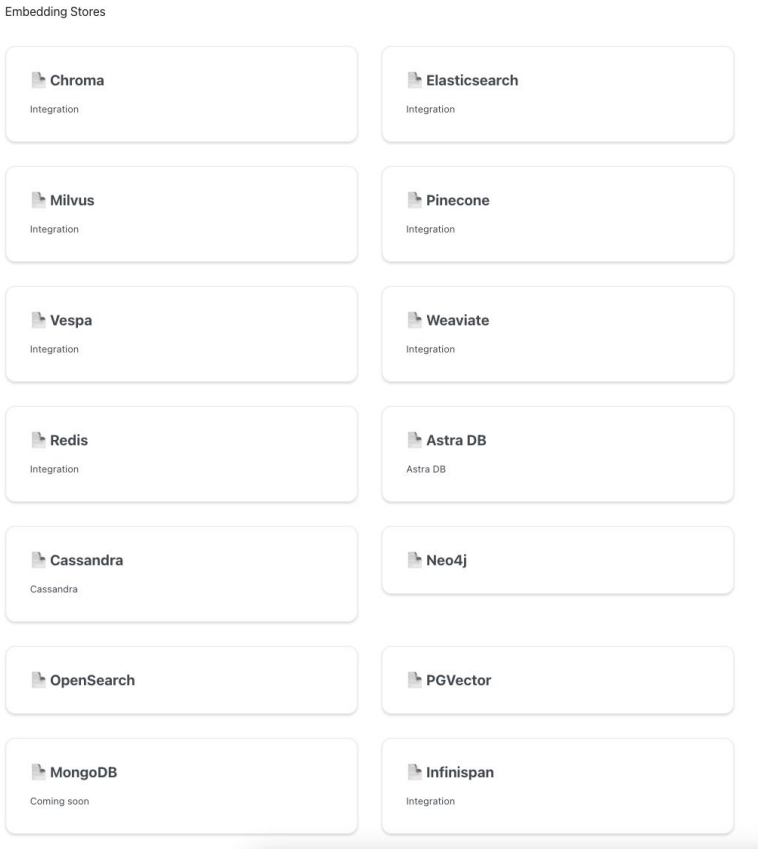
Langchain4j LLM Integrations

Supported LLM Integrations

Provider	Native Image	Completion	Streaming	Async Completion	Async Streaming	Embeddings	Image Generation	ReRanking
OpenAI		✅	✅	✅	✅	✅	✅	
Azure OpenAI		✅	✅			✅	✅	
Hugging Face		✅		✅		✅		
Amazon Bedrock		✅				✅		
Google Vertex AI Gemini		✅	✅	✅	✅			
Google Vertex AI	✅	✅		✅		✅	✅	
Mistral AI		✅	✅	✅	✅	✅		
DashScope		✅	✅		✅	✅		
LocalAI		✅	✅	✅		✅		
Ollama		✅	✅	✅	✅	✅		
Cohere								✅
Qianfan		✅	✅	✅	✅	✅		
ChatGLM		✅						
Nomic						✅		

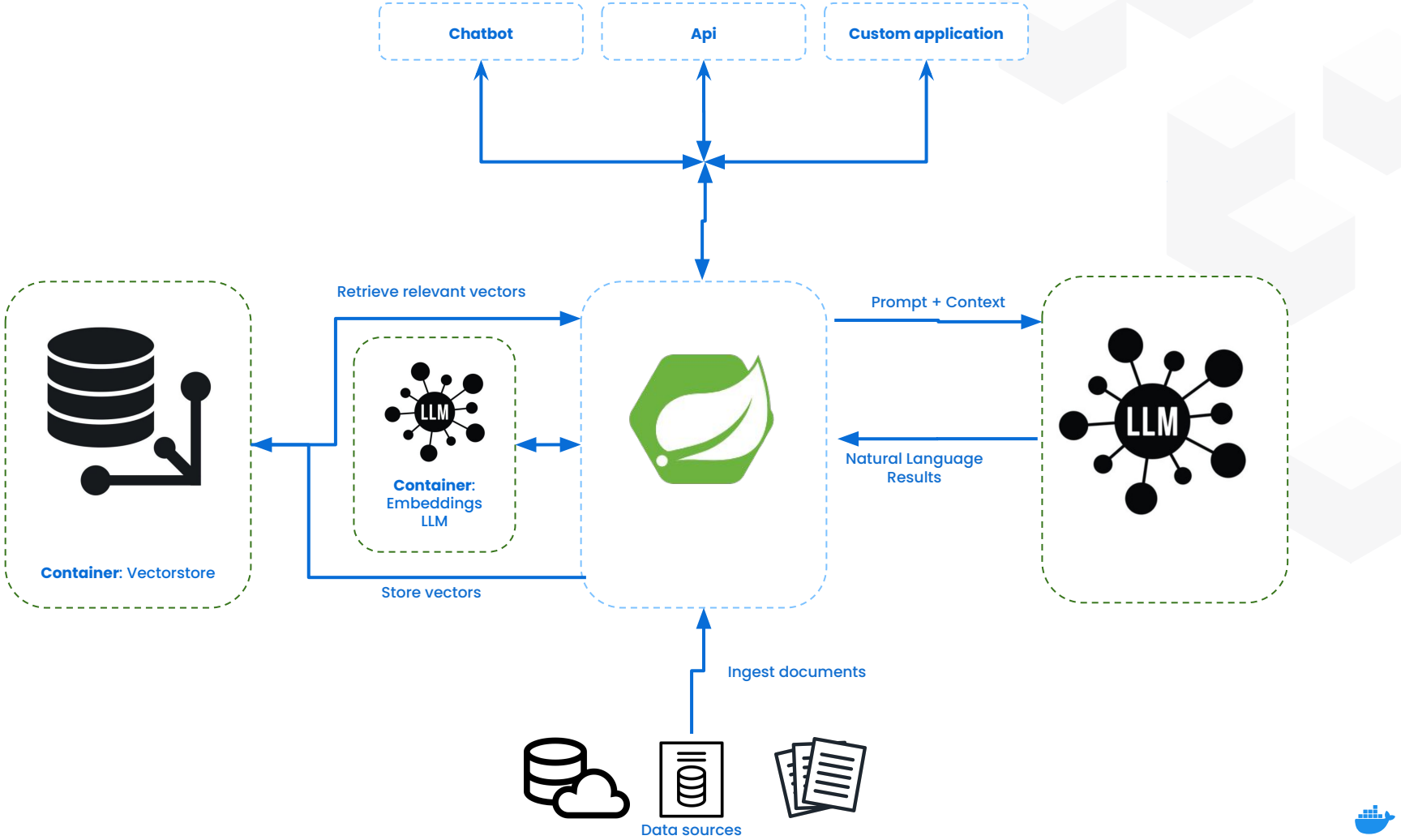
https://docs.langchain4j.dev/intro

Langchain4j Embedding Stores



https://docs.langchain4j.dev/category/embedding-stores


Commoditization of AI


Chatbot
Api
Custom application
Retrieve relevant vectors
Prompt + Context
LLM
Container: Embeddings LLM
Container: Vectorstore
Natural Language Results
LLM
Store vectors
Ingest documents
Data sources

Sounds like a lot of stuff to set up...

Testcontainers

Unit tests with real dependencies

TEST DEPENDENCIES AS CODE

Get lightweight and throwaway containers during your tests

FOR YOUR ENTIRE STACK

Test against any container image: database, message broker, browser, etc.

FOR ANY LANGUAGE

Get started with the open source library for any of 6+ languages

WE ❤ OPEN SOURCE
& EMPLOY MAINTAINERS

Testcontainers

No more need for mocks or complicated environment configurations. Define your test dependencies as code, then simply run your tests and containers will be created and then deleted.

```
var redis = new GenericContainer(
  "redis:5.0.3-alpine"
).withExposedPorts(6379);

redis.start();
```

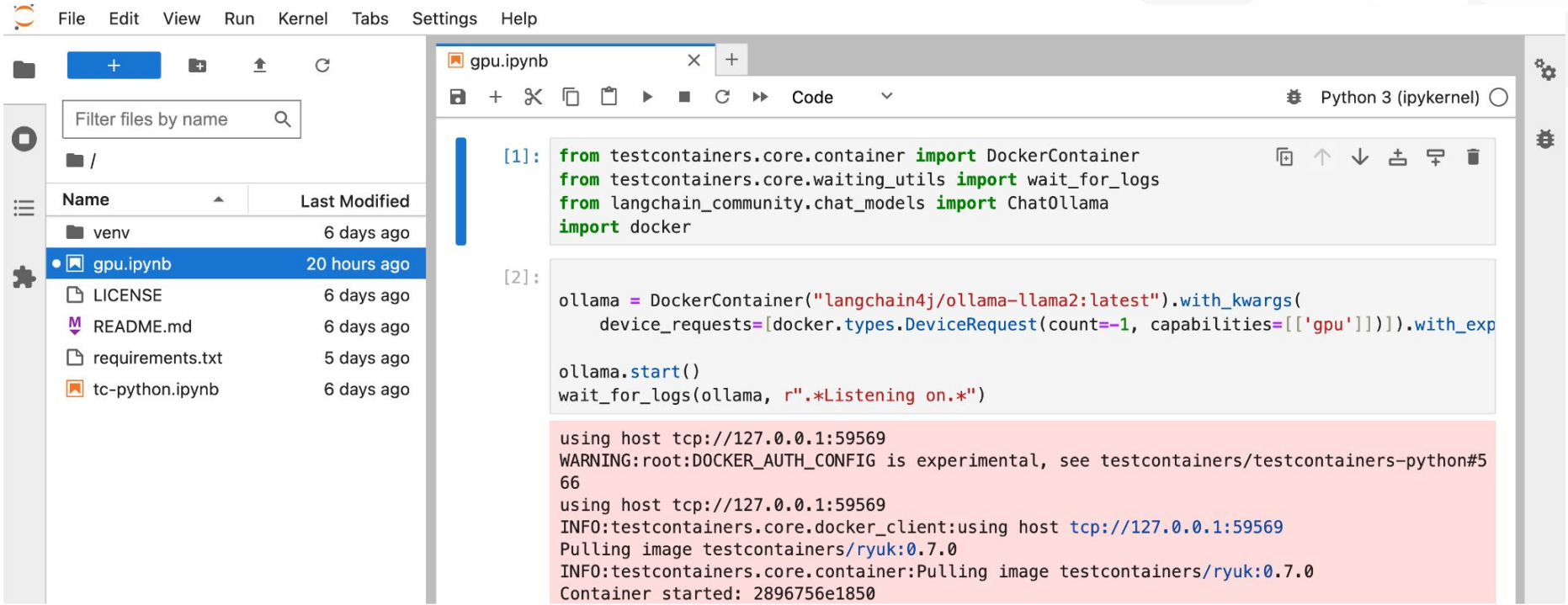
GPU Support

- Use withCreateContainerCmdModifier() and docker-java
- Only supported for NVIDIA GPUs (on Linux or Docker Desktop on Windows)

```
if (runtimes != null) {
    if (runtimes.containsKey("nvidia")) {
        withCreateContainerCmdModifier(cmd -> {
            cmd
                    .getHostConfig()
                    .withDeviceRequests(
                            Collections.singletonList(
                                    new DeviceRequest()
                                            .withCapabilities(
                                                    Collections.singletonList(
                                                            Collections.singletonList("gpu")))
                                            .withCount(-1)
                            )
                    );
        });
    }
}
withExposedPorts(11434);
```

Testcontainers Cloud


DEMO


File Edit View Run Kernel Tabs Settings Help
Filter files by name
/
Name
Last Modified
venv 6 days ago
gpu.ipynb 20 hours ago
LICENSE 6 days ago
README.md 6 days ago
requirements.txt 5 days ago
tc-python.ipynb 6 days ago
gpu.ipynb
Code
Python 3 (ipykernel)
[1]:
from testcontainers.core.container import DockerContainer
from testcontainers.core.waiting_utils import wait_for_logs
from langchain_community.chat_models import ChatOllama
import docker
[2]:
ollama = DockerContainer("langchain4j/ollama-llama2:latest").with_kwargs(
device_requests=[docker.types.DeviceRequest(count=-1, capabilities=[['gpu']])]).with_exp
ollama.start()
wait_for_logs(ollama, r".*Listening on.*")
using host tcp://127.0.0.1:59569
WARNING:root:DOCKER_AUTH_CONFIG is experimental, see testcontainers/testcontainers-python#5
66
using host tcp://127.0.0.1:59569
INFO:testcontainers.core.docker_client:using host tcp://127.0.0.1:59569
Pulling image testcontainers/ryuk:0.7.0
INFO:testcontainers.core.container:Pulling image testcontainers/ryuk:0.7.0
Container started: 2896756e1850