

Hyland™

Alfresco Tech Talk Live #154

January 24, 2024

Agenda



- Community news
- Enhancing Document-Centric Features with On-Premise Generative AI for Alfresco Community

Happy New Year, humans



NightCafe



Human made

Meet a Hyland Developer Evangelist at FOSDEM!

FOSDEM'24

[↑](#) [About](#) [News](#) [Schedule](#) [Stands](#) [Volunteer](#) [Practical](#)

Brussels / 3 & 4 February 2024

[schedule](#)

[News](#) [Sponsors](#) [Contact](#)

[FOSDEM 2024](#) / [Schedule](#) / [Events](#) / [Developer rooms](#) / [Collaboration and Content Management](#) / [Using Generative AI and Content Service Platforms together](#)

Using Generative AI and Content Service Platforms together

- [Track: Collaboration and Content Management devroom](#)
- [Room: H.2213](#)
- [Day: Saturday](#)
- [Start: 15:20](#)
- [End: 15:45](#)
- [Video only: h2213](#)
- [Chat: Join the conversation!](#)

Join us to learn from the integration of Generative AI (GenAI Stack) with a Content Service Platform (Alfresco).

Leveraging Open-Source platforms, we will explore different applications of Generative AI within Document Management ecosystem. From enabling conversational searching to generating concise summaries and facilitating intelligent categorization, this session aims to showcase practical approaches that combines the potential of Generative AI and Content Service Platforms.

Live demo and sample source code will be provided!

Join us to discover how this dynamic collaboration enhances document processing, retrieval, and organization, opening new avenues for efficient and intelligent information management.

Speakers



[Angel Borroy](#)

Attachments

[Alfresco GenAI Stack \(slides\)](#)

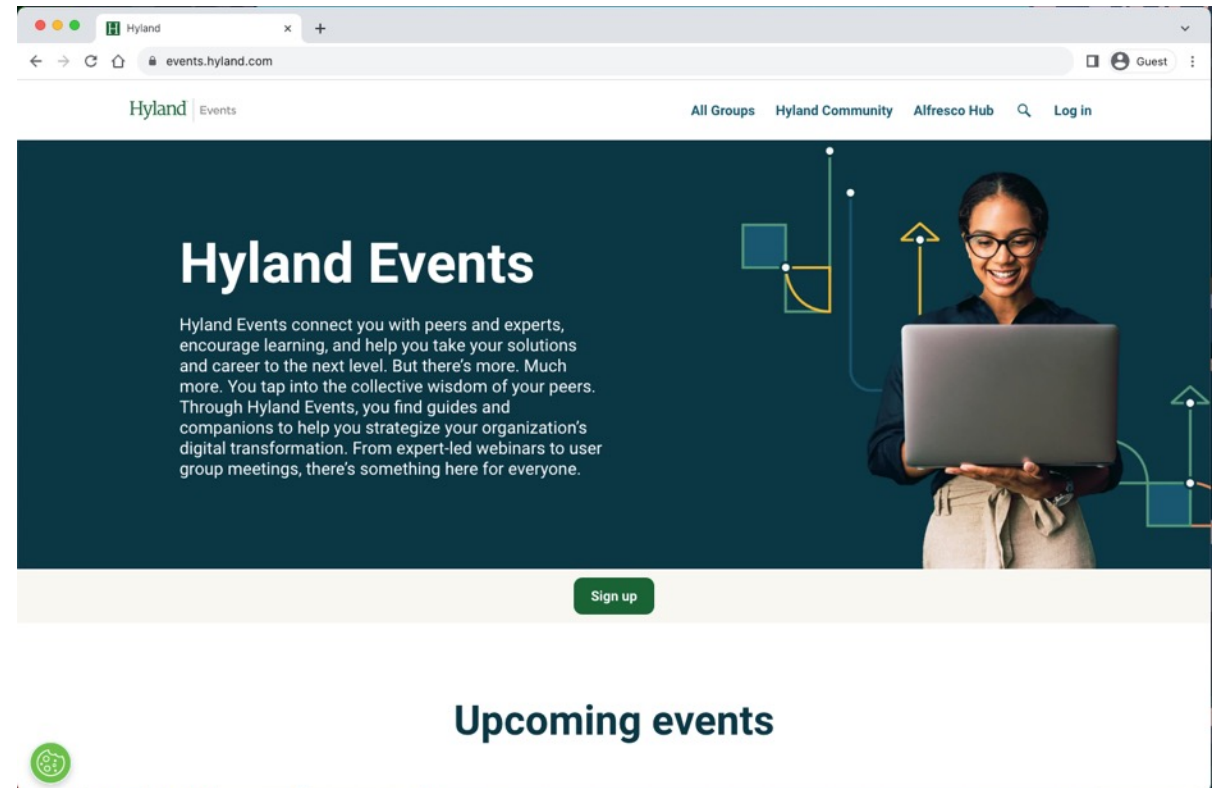
Meetups, meetups, meetups... and a DevCon ?

Initiatives:

- Local User Groups with CSMs
- Meetups (Online, In Person)
- Chapters on events.hyland.com

...

We want to check if it makes sense to organize DevCon again.



Resources

Alfresco

[Deploying Alfresco Outlook Transform Engine with Docker Compose](#) @ GitHub

[Script Object for RenditionService2](#) @ GitHub and thanks to @hi-ko

[Alfresco GenAI](#) @ GitHub

Resources to come

- Adapting your logging configuration to log4jv2
- How to set up messages for increased reliability
- Upgrading your addons to Jakarta EE 10 and Spring 6
- Upgrading to Apache Tomcat 10
- Using Control Center App with Community Edition
- Deploying Alfresco with Helm in Kubernetes

Collaboration

Blog posts

- [Summarization of textual content in Alfresco repository with Amazon Bedrock](#) by @abhinavmishra14

Contributions

- <https://github.com/aborroy/alfresco-dockerx-builder/issues/11> by @uvukasinovic
- <https://github.com/Alfresco/alfresco-docker-installer/issues/185> by @luca86r
- <https://github.com/Alfresco/alfresco-docker-installer/issues/186> by @N00BTellaBrot

Conferences

- [FOSDEM 2024](#), 3 & 4 February, Brussels

TTL Speakers wanted!

- Take the opportunity to showcase your work with the community
- About Alfresco, Nuxeo, and associated technologies
- Best practices, integration, scaling, cloud, ...
- In your native language

Today's talk



Enhancing Document-Centric Features with On-Premise Generative AI for Alfresco Community

Angel Borroy
Tech Evangelist at Hyland

Hyland™

Hyland™

Enhancing Document-Centric Features with On-Premise Generative AI for Alfresco Community

Alfresco Tech Talk Live #154

Angel Borroy
Developer Evangelist



docker

January 24, 2024

Agenda

- Docker GenAI Stack
- Alfresco Integration
- Document Features
 - Existing Content
 - New Content
- What Else?



Content credentials
Generated with AI · 19 January 2024 at 8:24 am

Docker GenAI Stack



Components

ollama

- Local management of *open source* LLMs
- Catalog of preconfigured LLMs, such as Llama2 or Mistral



neo4j

- Graph and native vector search capabilities
- Ground LLMs for more precise GenAI predictions and outcomes

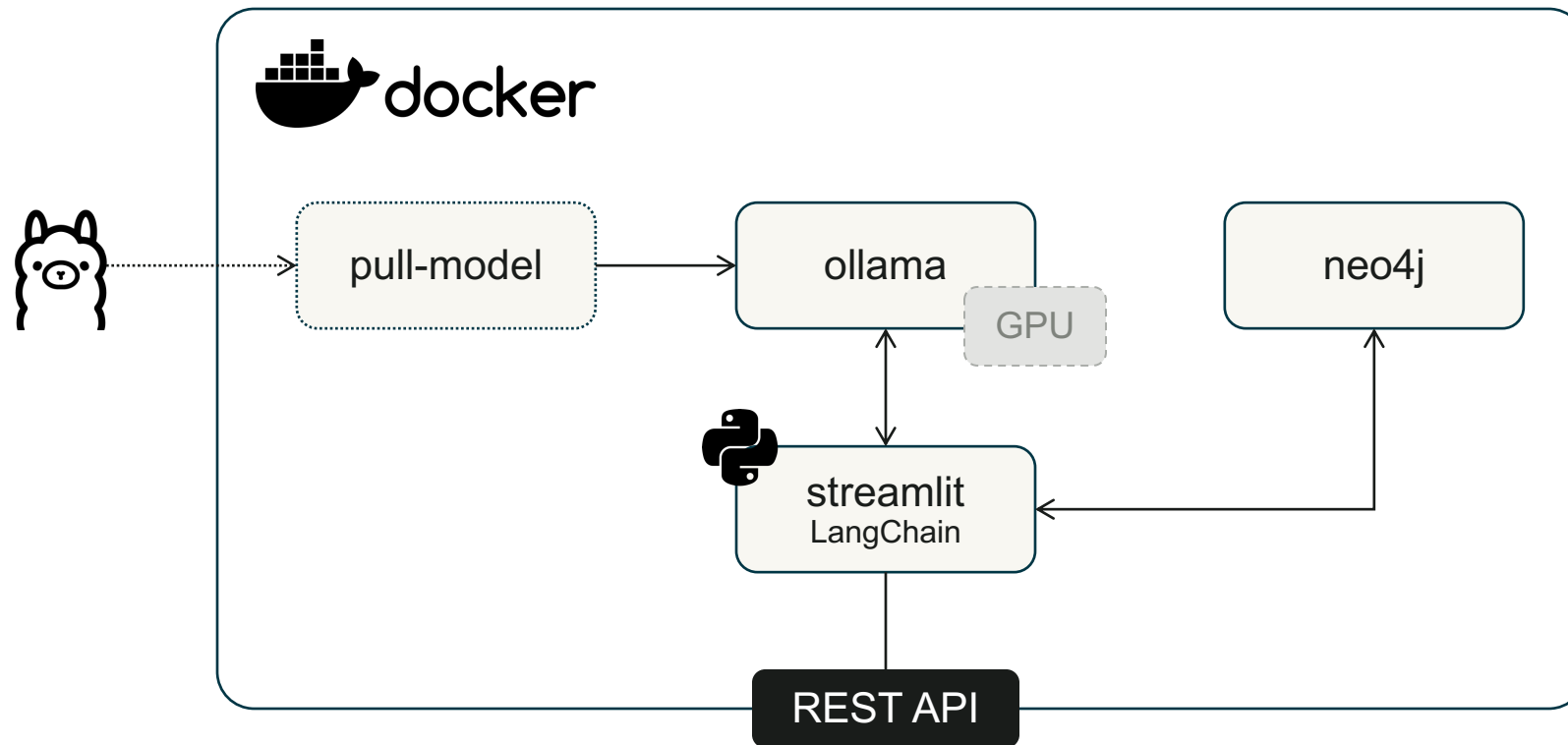


LangChain

- Communication between the LLM, your application, and the database
- Python framework for developing applications powered by LLMs



Deployment





LLM 3 Billion Parameters

phi	MIT License
dolphin-phi	MIT License
orca-mini	cc-by-nc-sa-4.0
deepseek-coder	DEEPSEEK LICENSE AGREEMENT



deepseek coder





LLM 7 Billion Parameters

llama2	<u>LLAMA2 COMMUNITY LICENSE AGREEMENT</u>
codellama	<u>LLAMA2 COMMUNITY LICENSE AGREEMENT</u>
vicuna	<u>LLAMA2 COMMUNITY LICENSE AGREEMENT</u>
mistral	<u>Apache License 2.0</u>
mistral-openorca	<u>Apache License 2.0</u>
llava	<u>Apache License 2.0</u>
orca-mini	<u>cc-by-nc-sa-4.0</u>
deepseek-coder	<u>DEEPSEEK LICENSE AGREEMENT</u>
orca2	<u>MICROSOFT RESEARCH LICENSE</u>
falcon	<u>FALCON 180B TII LICENSE 1.0</u>

8 GB
RAM
1010
1010

4 GB
Size





LLM 13 Billion Parameters

llama2	LLAMA2 COMMUNITY LICENSE AGREEMENT
codellama	LLAMA2 COMMUNITY LICENSE AGREEMENT
vicuna	LLAMA2 COMMUNITY LICENSE AGREEMENT
orca-mini	cc-by-nc-sa-4.0
llava	Apache License 2.0
orca2	MICROSOFT RESEARCH LICENSE

16 GB
RAM
1010
1010

8 GB
Size
1010
1010





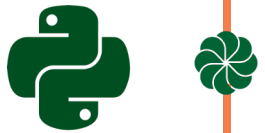
LLM 13+ Billion Parameters

llama2	LLAMA2 COMMUNITY LICENSE AGREEMENT	70B	
codellama	LLAMA2 COMMUNITY LICENSE AGREEMENT	34B	
mixtral	Apache License 2.0	56B	
deepseek-coder	DEEPSEEK LICENSE AGREEMENT	33B	
vicuna	LLAMA2 COMMUNITY LICENSE AGREEMENT	33B	
falcon	FALCON 180B TII LICENSE 1.0	40B / 180B	



deepseek coder





Creating a new REST API

```
import streamlit as st
from langchain.chains import RetrievalQA
from langchain.vectorstores.neo4j_vector import Neo4jVector
from fastapi import FastAPI, UploadFile
```

```
app = FastAPI()
```

```
@app.post("/document")
```

```
async def document(file: UploadFile, termList: str):
```

```
    vector_store = Neo4jVector.from_texts(...)
```

```
    qa = RetrievalQA.from_chain_type(
```

```
        llm=llm, chain_type="stuff", retriever=vector_store.as_retriever()
```

```
    )
```

```
    stream_handler = StreamHandler(st.empty())
```

```
    term = qa.run("Hello", callbacks=[stream_handler])
```



Streamlit



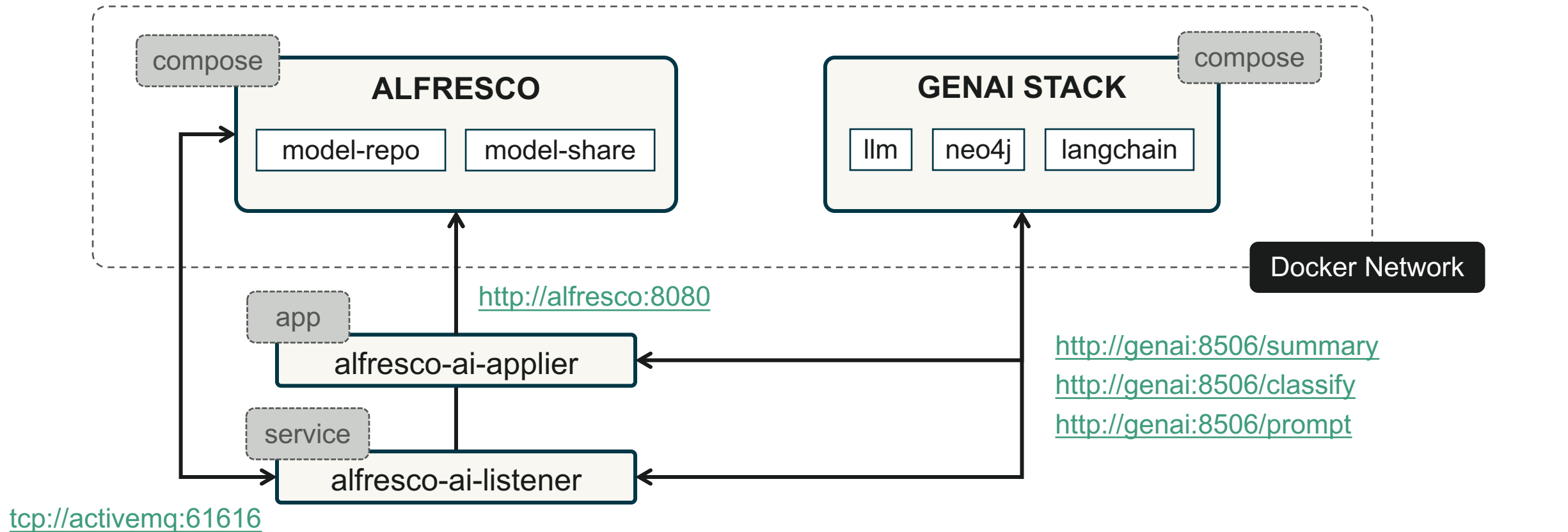
LangChain



FastAPI


Alfresco Integration


Deployment








Alfresco Aspects

 summarizable
summary text
tags text
llmSummary text

 promptable
question text
answer text
llmPrompt text

 classifiable
terms text, multiple

 classified
term text
llmClassify text



tags are identified by LLM without restrictions
term is selected by LLM from a list of **terms**
llm* includes the LLM name used

Alfresco Integration

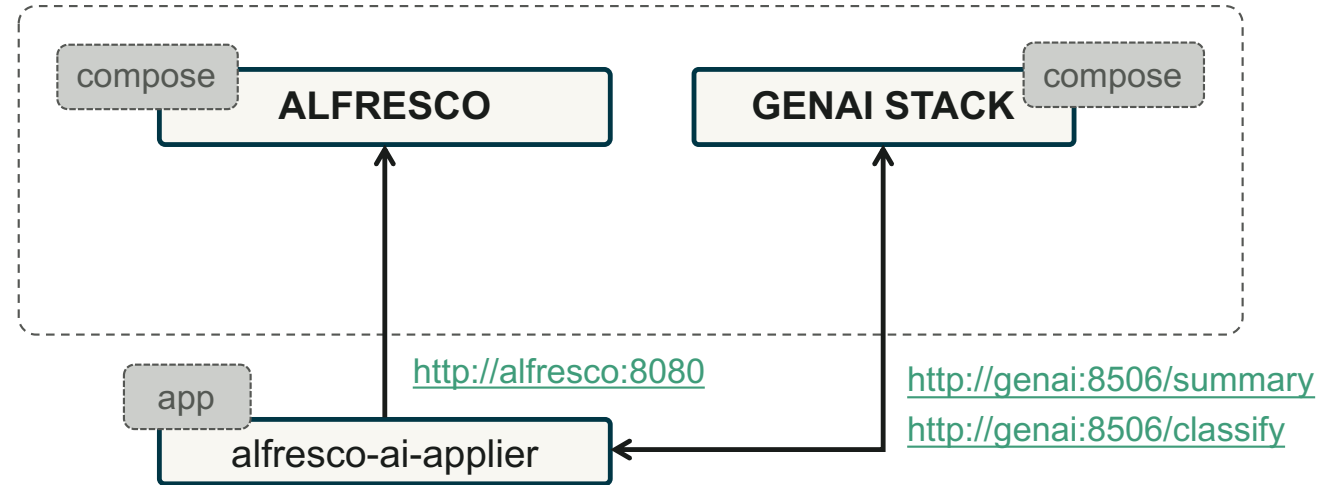
alfresco-ai-applier



AI Applier

Run action on a Repository Folder

- Summary
- Classify
 - List of terms required as parameter



```
$ java -jar alfresco-ai-applier-0.8.0.jar \  
--applier.root.folder=/app:company_home/app:shared \  
--applier.action=SUMMARY
```

```
$ java -jar alfresco-ai-applier-0.8.0.jar \  
--applier.root.folder=/app:company_home/app:shared \  
--applier.action.classify.term.list=English,Spanish \  
--applier.action=CLASSIFY
```



Alfresco Integration

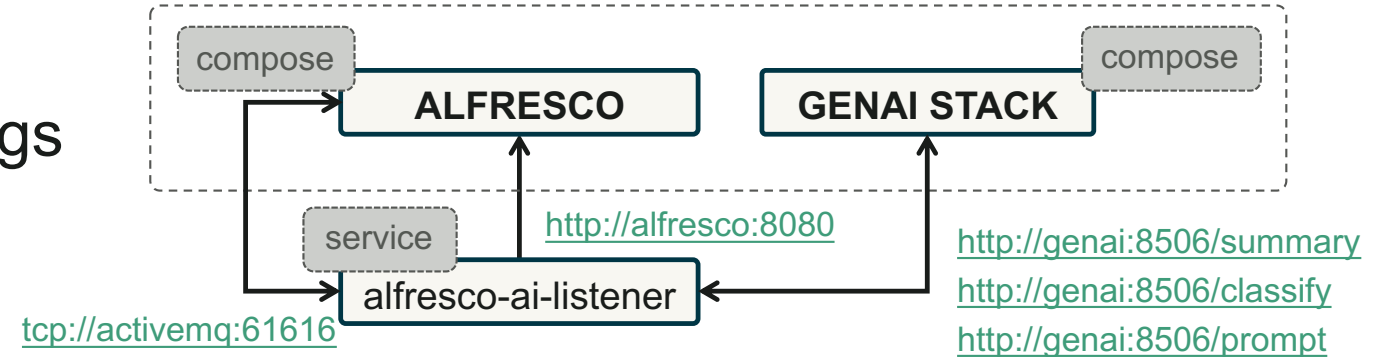
alfresco-ai-listener



AI Listener

Listen to ActiveMQ for aspect settings

- Summary
- Classification
- Prompting



```
alfresco-ai-listener:  
  image: alfresco-ai-listener  
  environment:  
    CONTENT_SERVICE_URL: "http://alfresco:8080"  
    SPRING_ACTIVEMQ_BROKERURL: "tcp://activemq:61616"  
    GENAI_URL: "http://genai:8506"
```

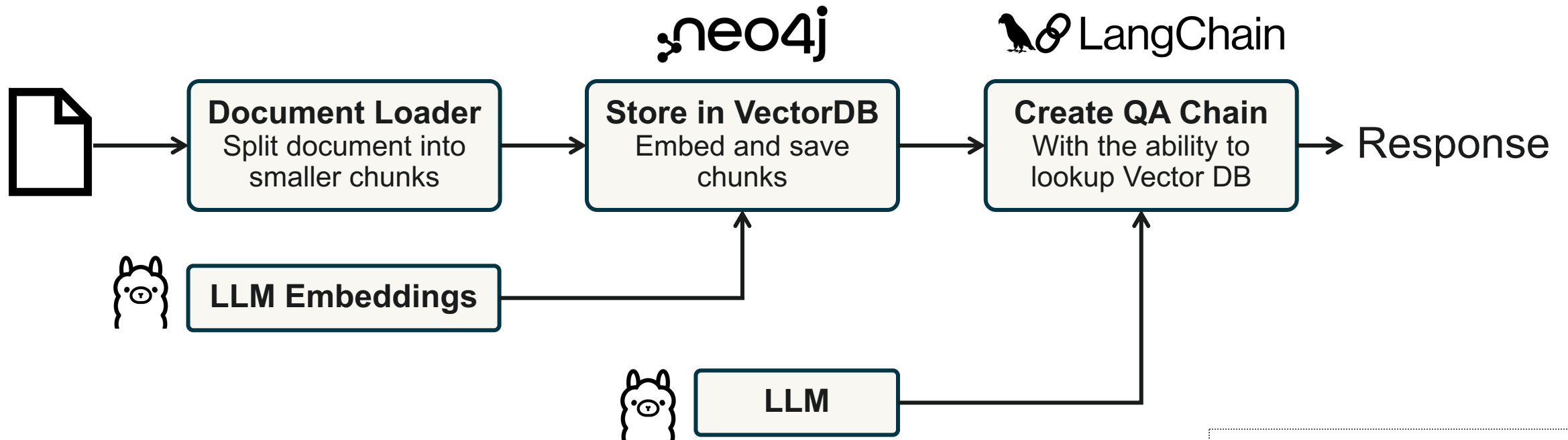


Spring Boot



GenAI App

Python application that provides REST API endpoints



```
LLM=mistral
SUMMARY_LANGUAGE=English
SUMMARY_SIZE=120
TAGS_NUMBER=3
```



GenAI App



```
curl --location 'http://localhost:8506/summary' --form 'file=@"/file.pdf" '{
  "summary": " The text discusses...",
  "tags": " Golang, Merkle, Difficulty",
  "model": "mistral"
}
```

SUMMARY Write a short summary of the text in 120 words only in ENGLISH

TAGS Provide 3 words to categorize the document in language ENGLISH in a single line.
Use only language ENGLISH for these 3 words in the answer.
Don't add any explanation for the words in the answer.
Don't add any note after the list of words in the answer.
Don't use bullets or numbers to list the words in the answer.
Don't add in the answer the translation of the words in a different language after the list of words.
Give the answer exactly as a list of 3 words in language ENGLISH separated with comma and without ending dot.



GenAI App

```
curl --location \
'http://localhost:8506/classify?termList="Japanese,Spanish,Vietnamese"' \
--form 'file=./file.pdf'
{
  "term": " English",
  "model": "mistral"
}
```

- TERM** Pick one of the following list of categories: Japanese, Spanish, Vietnamese.
Write the answer only in ENGLISH language.
Don't add any explanation for the choice in the answer.
Don't add any note after the word in the answer.
Don't add any space before the word in the answer.
Don't add in the answer the translation of the word in a different language after chosen word.
Give the answer exactly as a single word from the list.



GenAI App

```
curl --location 'http://localhost:8506/prompt?prompt="What is this text about?"' \
--form 'file=./file.pdf'

{
  "answer": " Yes, it is difficult to find childcare in Tokyo.",
  "model": "mistral"
}
```

PROMPT

What is this text about?

Write the answer only in ENGLISH language.

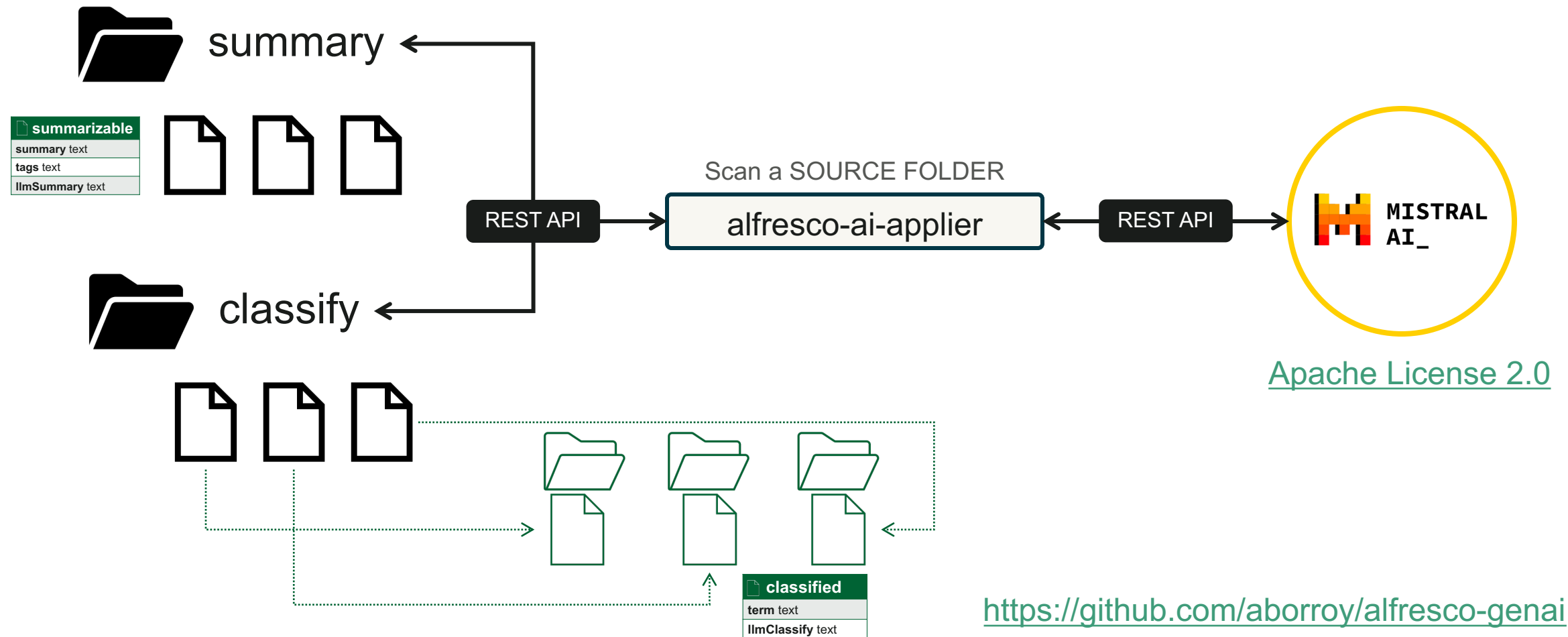
Don't add any translation to the answer.

Document Features

Document Features



Existing Content





c5a.8xlarge

32 vCPUs

64 GiB RAM

~~GPU~~



Demo Time

Document Features



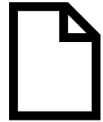
New Content

documentable
summary text
tags text
IlmSummary text



summary

promptable
question text
answer text
IlmPrompt text



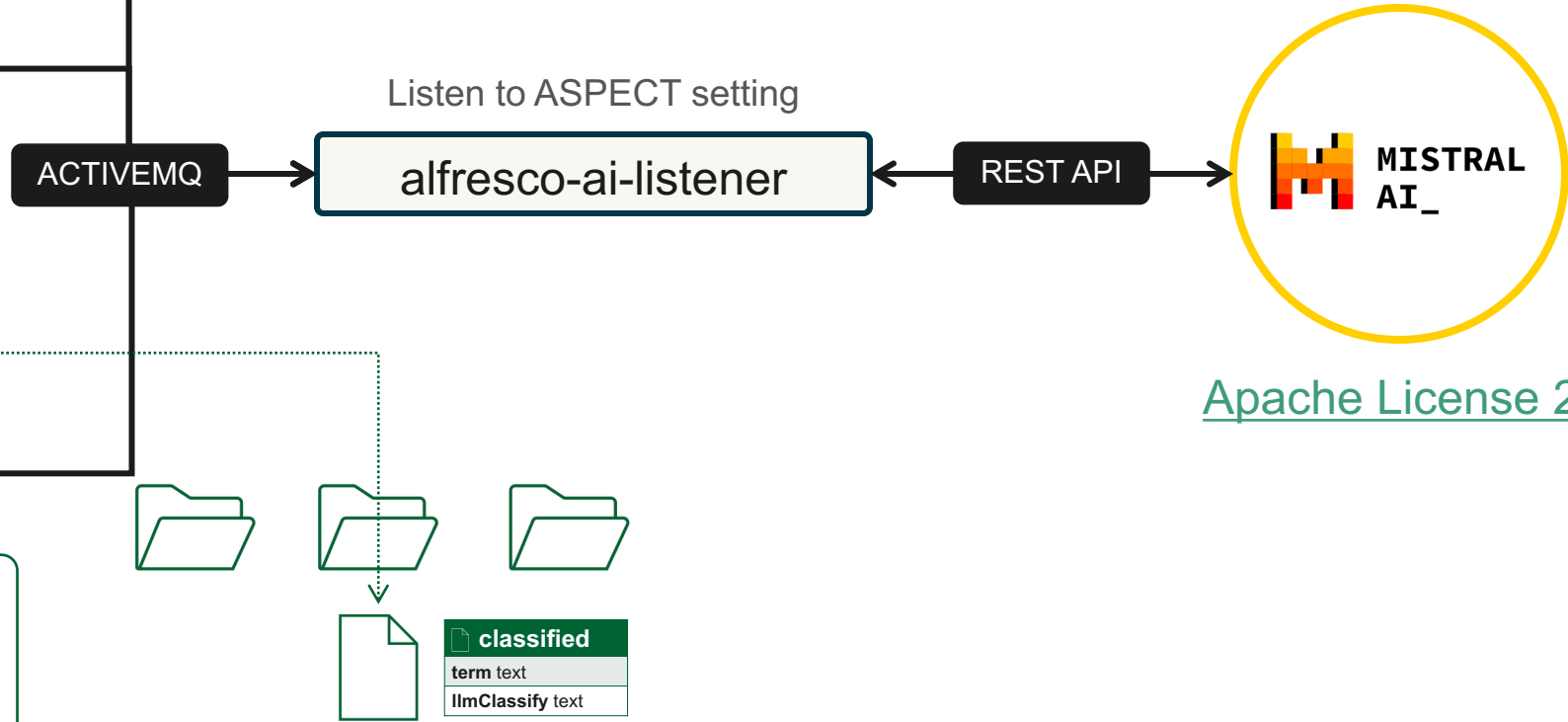
prompt

classifiable
terms text, multiple



classify

```
document.move(  
  document.parent.childByNamePath(  
    document.properties["genai:term"]  
  )  
);
```



Apache License 2.0



c5a.8xlarge

32 vCPUs

64 GiB RAM

~~GPU~~



Demo Time

What Else?

Find your way



Docker AI/ML Hack-a-thon

- [Readme AI](#)
- [Techdocs](#)
- [Docker Image Analyzer](#)
- [Docker Log Sentiment Analyzer](#)
- [GitChats AI](#)

ollama alternatives

- <https://gpt4all.io/index.html>
- <https://www.secondstate.io/run-llm>
- <https://huggingface.co/docs/hub/spaces-sdks-docker-first-demo>



Hyland™

Thanks!