

Interactive Graphene Documentation using LLMs and LangChain

Swetha Lakshmana Murthy

Focus of Today's Discussion



Introduction

Motivation behind the
Proof of Concept (POC)



Proposed Workflow

Interplay between
Components in an LLM
Pipeline



Future Scope and Conclusion

Potential Enhancements



Demo of the Working Pipeline

AI-Builder

Motivation behind the POC- Enhancing Platform Utility

Source-code tutorials for Graphene Pipelines

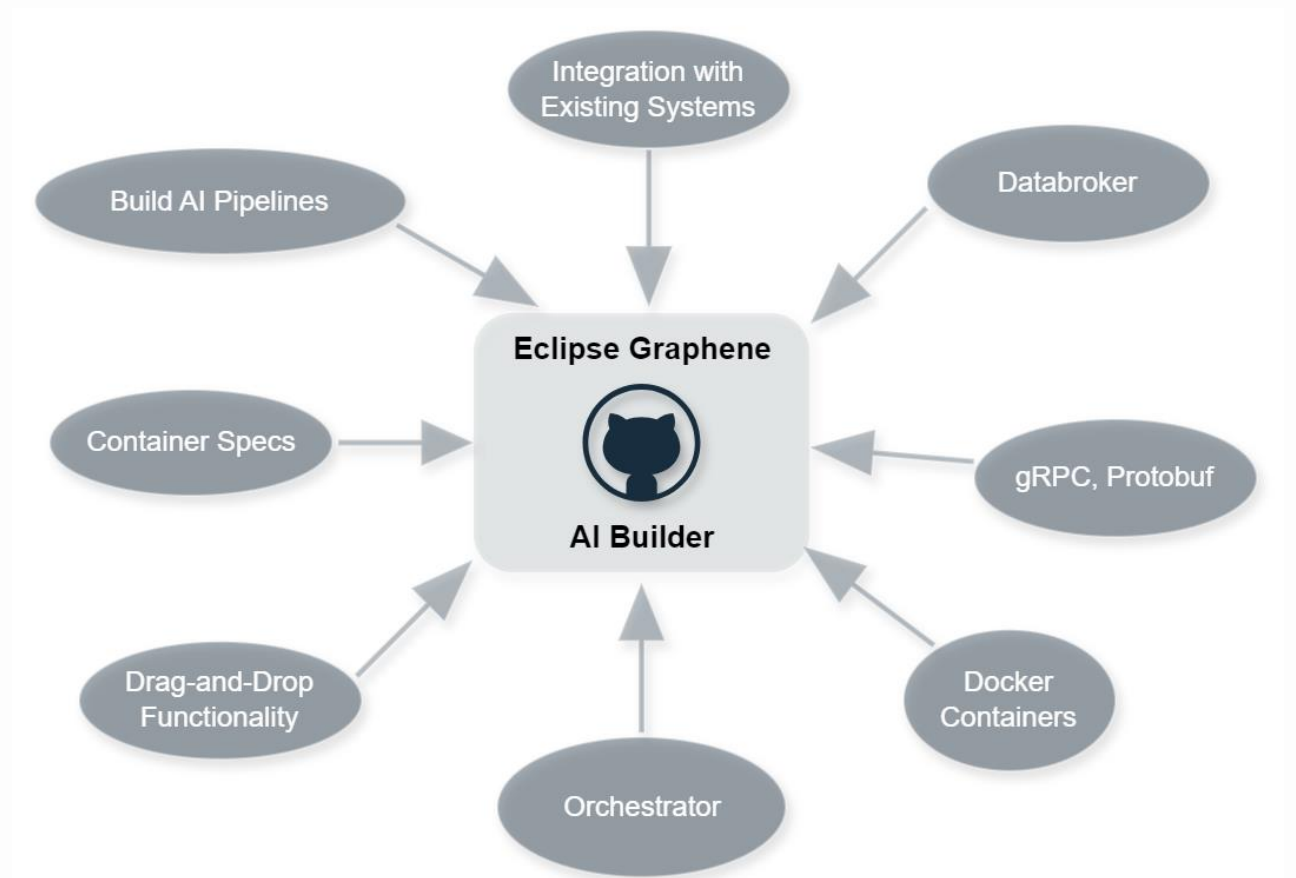
- Frequent documentation updates

Documentation Challenges

- Often, inadequacy, inaccessibility, or cumbersome processes
- Need to automate this process

Large Language Models (LLMs) as Documentation Augmenters

- Leveraging the use of LLMs enhances the Eclipse Graphene platform's [1][2] capability to handle and maintain documentation effectively



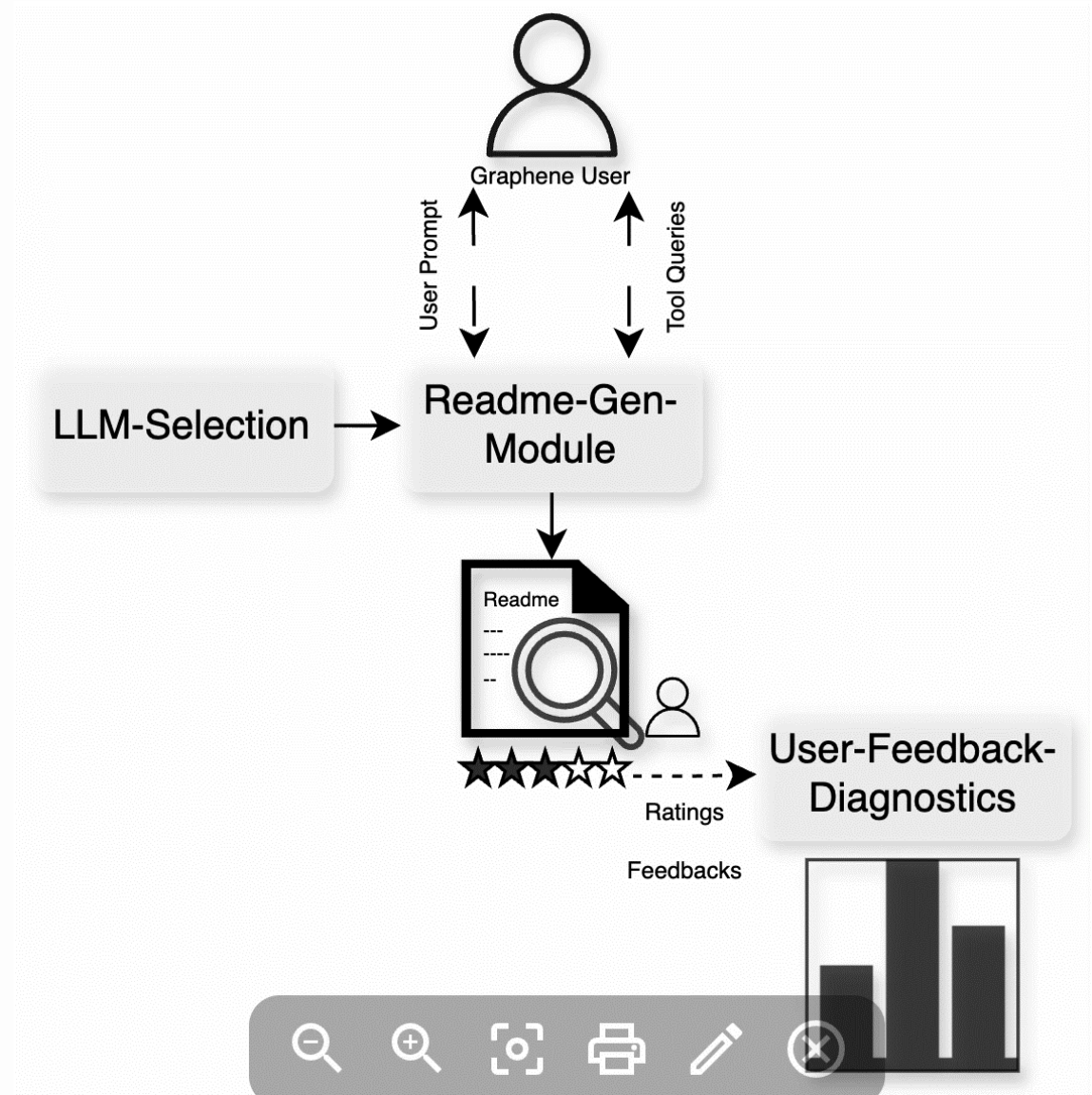
Proposed Workflow

User Input:

- Model Selection
- Intended Graphene Tutorial Repo
- Prompts
- Queries
- Ratings and Feedback

Pipeline Output:

- README.md
- Metrics - Metadata

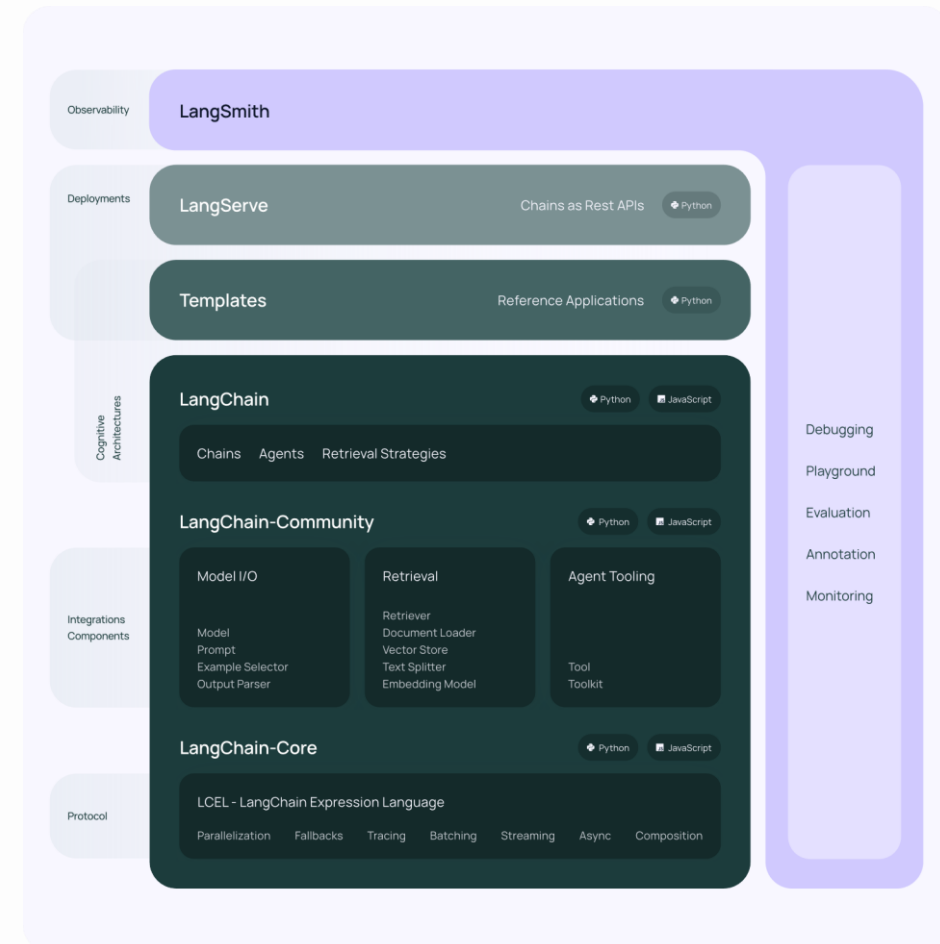


LLM-Selection

- Pipeline Tested with Four LLMs (Ongoing)
 1. gpt-3.5-turbo-instruct
 2. Mistral-7B-Instruct-v0.1_v2
 3. llama-2-13b-chat_v3
 4. OpenGPT-X-24EU-Bactrian-X-ENDEFRTES

LangChain Framework

- Designed to simplify the creation of applications using LLMs
- Addresses use-cases:
 - Chatbots, retrieval-augmented generation, document summarization, etc.
- Main Features used in POC
 - Models I/O
 - Chains
 - Agent Tooling
 - Lang Chain Expression Language (LCEL)

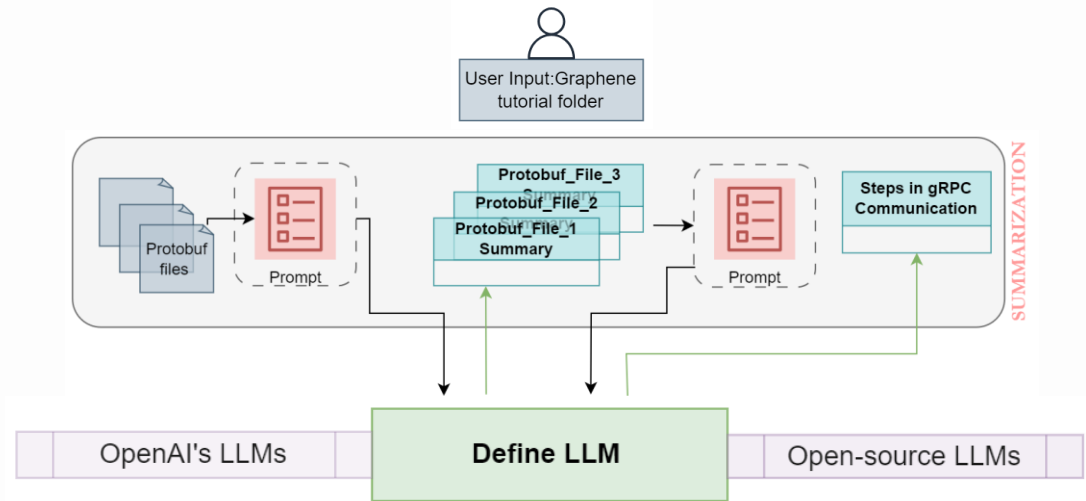


LangChain Stack [4]

Readme-Gen-Module - Core Logic

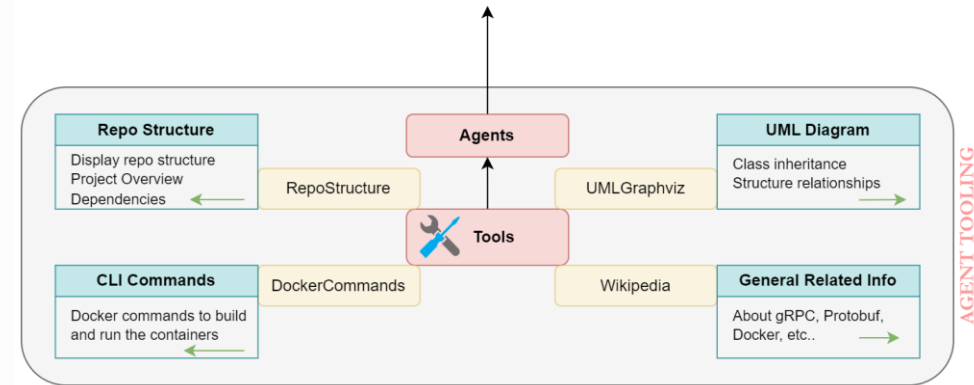
MapReduce Chain

- Summarize each document on its own in a "map" step and then "reduce" the summaries into a final summary
- Iterative approach
- Parallel aggregation of results



Agent Tooling

- Simple, basic functions as tools
- Reduce LLM Hallucinations
- Controlled text generation
- Aids in building structured prompts



User-Feedback – Human-in-loop

- Manages README file's user ratings and feedback
- Helps gauge the pipeline's effectiveness

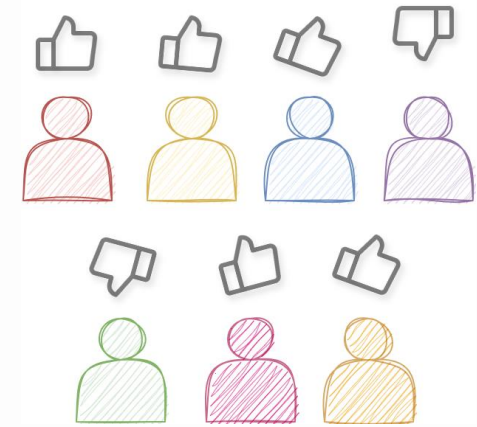
overall_avg_star_rating:

- Scale: 1 to 5 for user satisfaction.
- Mean rating received for READMEs in Graphene Tutorials.

overall_avg_feedback_sentiment_score:

Scale: -1 to 1 for sentiment polarity.

- Average sentiment polarity derived from README feedback

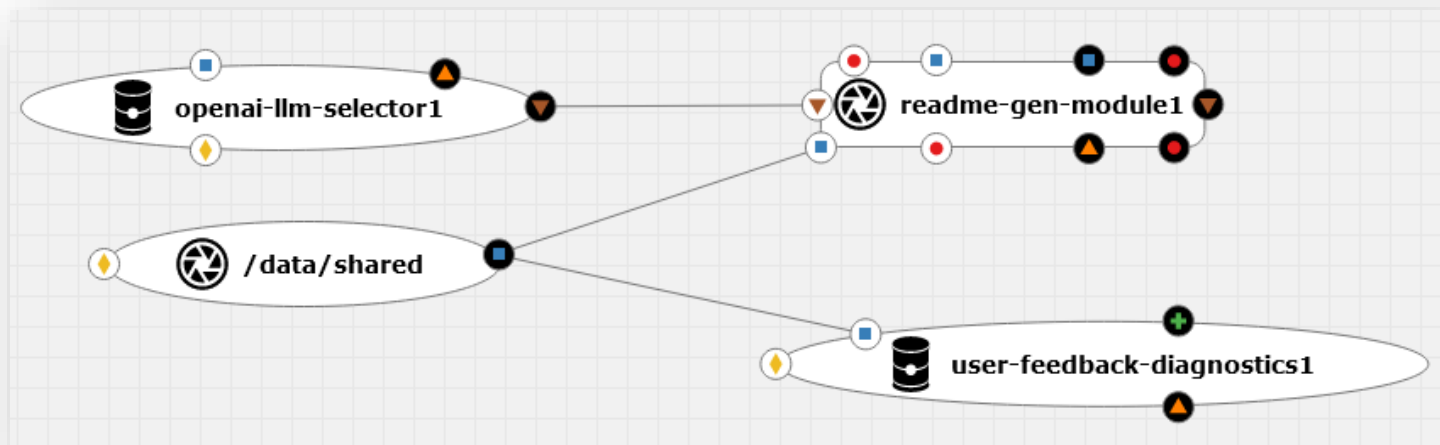


Feedback Text	Corresponding Sentiment Score
Good explanation	0.7
The generated docker commands were not accurate.	-0.2
Very good!	1.0
LLM can generate a better response!	0.5
Good generation	0.7

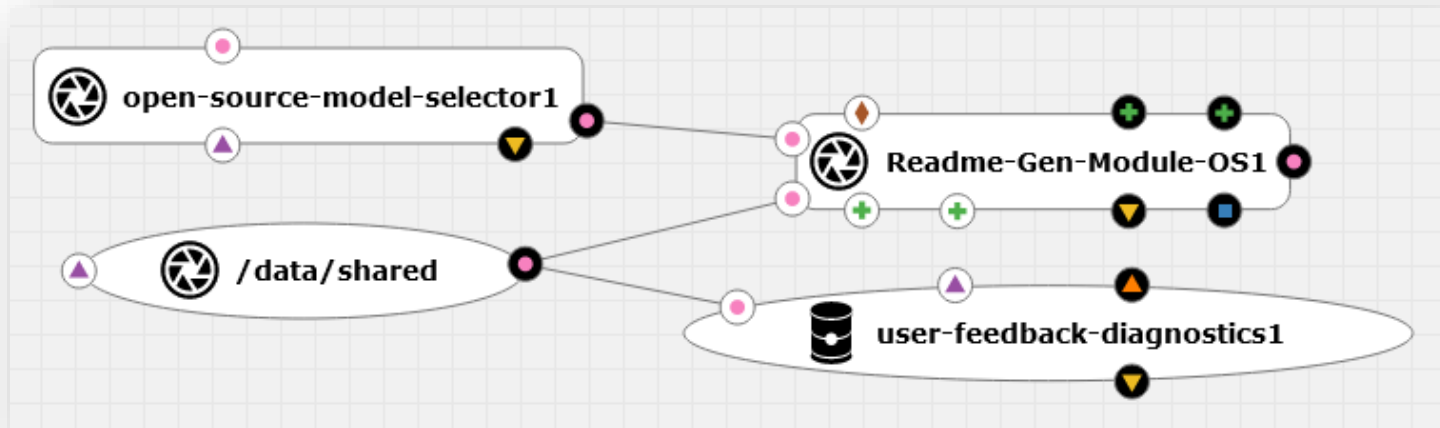
▼ metrics:	
date_time:	"2024-04-18 17:32:01"
type:	"LLM Metrics"
status_text:	"success"
▼ more_is_better:	
overall_avg_star_rating:	2.625
overall_avg_feedback_sentiment_score:	0.375

AI-Builder Pipeline

OpenAI LLMs



Open-source LLMs



Future Scope

1. **Q&A Chatbot** using the generated README
2. Automated tracking of **code updates** and README maintenance
3. Source **code optimization** recommendations
4. Improvise content generation for **Domain-Specific terminologies**
5. Identify issues and areas for enhancement in **large codebases**

References

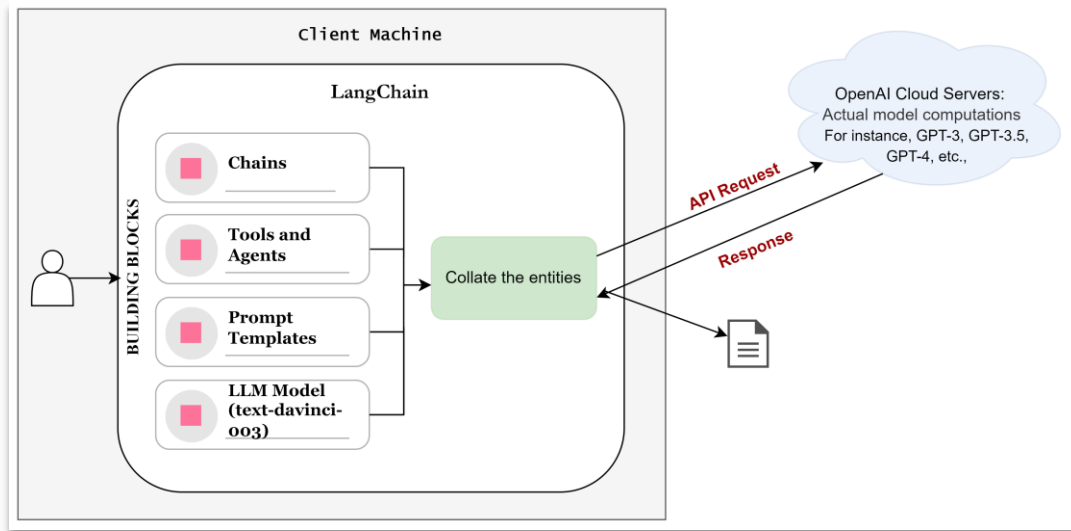
1. <https://gitlab.eclipse.org/eclipse/graphene/tutorials>
2. <https://www.ai4europe.eu/ai-builder>
3. https://gitlab.eclipse.org/eclipse/graphene/tutorials/-/tree/main/graphene_llm_readme_gen?ref_type=heads
4. https://python.langchain.com/docs/get_started/introduction
5. https://python.langchain.com/docs/modules/chains/document/map_reduce
6. <https://python.langchain.com/docs/modules/tools/>
7. <https://arxiv.org/abs/2403.10588>
8. <https://arxiv.org/abs/2404.02183>
9. <https://arxiv.org/abs/2308.03099>
10. <https://arxiv.org/abs/2306.01394>
11. <https://arxiv.org/abs/2201.11903>
12. <https://arxiv.org/abs/2310.12430>
13. <https://www.lakera.ai/blog/large-language-model-evaluation>
14. <https://textblob.readthedocs.io/en/dev/>
15. <https://openai.com/>
16. <https://huggingface.co/mistralai/Mistral-7B-v0.1>

Thank you



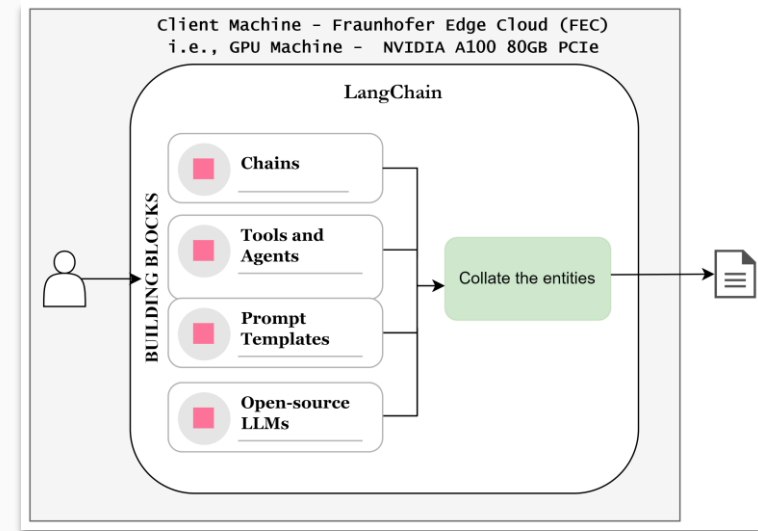
Inside the Pipeline: Key Components

- Open-Source and Closed-Source LLMs



OpenAI LLMs

- Model : text-davinci-003
- GPT-3.5 architecture



Open-source LLMs

- Model : mistralai/Mistral-7B-Instruct-v0.1
- Instruct fine-tuned version of the Mistral-7B-v0.1 generative text model
- Set up in FEC VM with 1 A100 GPU

• Sample README.md

Introduction

This project is based on an existing project. The main purpose of this project is to predict the price of a house. The main purpose of this project is to predict the price of a house. The main purpose of this project is to predict the price of a house.

What is gRPC communication?

gRPC is a remote procedure call (RPC) system that allows programs to communicate with each other over a network or the Internet. It is based on an interface definition language (IDL) that describes the structure of the data and the operations that can be performed on it.

Protobuf files

This code defines a set of protobuf files for the project. The main purpose of this project is to predict the price of a house. The main purpose of this project is to predict the price of a house. The main purpose of this project is to predict the price of a house.

Steps in client-server communication

- 1. Client sends a request to the server for the PredictStatus message.
- 2. Server sends a response to the client for the PredictStatus message.
- 3. Client sends a request to the server for the PredictPrice message.
- 4. Server sends a response to the client for the PredictPrice message.

Dependencies

The external dependencies used in this project are listed below. The main purpose of this project is to predict the price of a house. The main purpose of this project is to predict the price of a house. The main purpose of this project is to predict the price of a house.

UML Diagram

The UML diagram for the project shows the relationships between the different components. The main purpose of this project is to predict the price of a house. The main purpose of this project is to predict the price of a house. The main purpose of this project is to predict the price of a house.

Docker commands

The Docker commands for the project are listed below. The main purpose of this project is to predict the price of a house. The main purpose of this project is to predict the price of a house. The main purpose of this project is to predict the price of a house.

Introduction

This project is based on an existing project. The main purpose of this project is to predict the price of a house. The main purpose of this project is to predict the price of a house. The main purpose of this project is to predict the price of a house.

What is gRPC communication?

gRPC is a remote procedure call (RPC) system that allows programs to communicate with each other over a network or the Internet. It is based on an interface definition language (IDL) that describes the structure of the data and the operations that can be performed on it.

Protobuf files

This code defines a set of protobuf files for the project. The main purpose of this project is to predict the price of a house. The main purpose of this project is to predict the price of a house. The main purpose of this project is to predict the price of a house.

Steps in client-server communication

- 1. Client sends a request to the server for the PredictStatus message.
- 2. Server sends a response to the client for the PredictStatus message.
- 3. Client sends a request to the server for the PredictPrice message.
- 4. Server sends a response to the client for the PredictPrice message.

Dependencies

The external dependencies used in this project are listed below. The main purpose of this project is to predict the price of a house. The main purpose of this project is to predict the price of a house. The main purpose of this project is to predict the price of a house.

UML Diagram

The UML diagram for the project shows the relationships between the different components. The main purpose of this project is to predict the price of a house. The main purpose of this project is to predict the price of a house. The main purpose of this project is to predict the price of a house.

Docker commands

The Docker commands for the project are listed below. The main purpose of this project is to predict the price of a house. The main purpose of this project is to predict the price of a house. The main purpose of this project is to predict the price of a house.

DEV AI-Builder

KI-NRW Playground

dev01.ki-lab.nrw:7080/dashboard

Fraunhofer IAIS Intran...

SAP Self Services

Outlook Web

Dashboard - Fraunhof...

Fraunhofer Identity As...

Fraunhofer IAIS Intran...

SAP Self Services

Outlook Web

Dashboard - Fraunhof...

DEV AI-Builder

Fraunhofer Identity As...

Imported From Firefox

Imported

KiNRW

7377657468612e6c616b73686d616e612e6d757274687940696169732e667261756e68666665722e6465

Readme-Pipeline

ReadmePipelineOS

Deployment: Readme-Pipeline

Status: Ready

Run

Reset

Delete

Logs

Capture Execution Metadata

Status Check	Nodename	Status Details	Logs	WebUI/Folder
✓	openai-llm-selector1	view	view	view
✓	orchestrator	view	view	view
✓	readme-gen-module1	view	view	view
✓	user-feedback-diagnostics1	view	view	view
✓	SharedFolder	view		view

Solution description

```
graph LR; A[openai-llm-selector1] --> B[readme-gen-module1]; C[/data/shared] --> B;
```

The primary goal of the pipeline is to create a user-friendly interface connecting the Large Language Models (LLMs) and generate a comprehensive document (README.md) for various tutorials in the Graphene platform. Additionally, it strives to explore and integrate frameworks such as LangChain to access the seamless pipeline deployment. This sample pipeline offers insights into the interplay between different components within an LLM pipeline.

Three containers are utilized within the Graphene Design Studio framework to achieve this.

1. Openai LLM Selector

dev01.ki-lab.nrw:7080/dashboard#