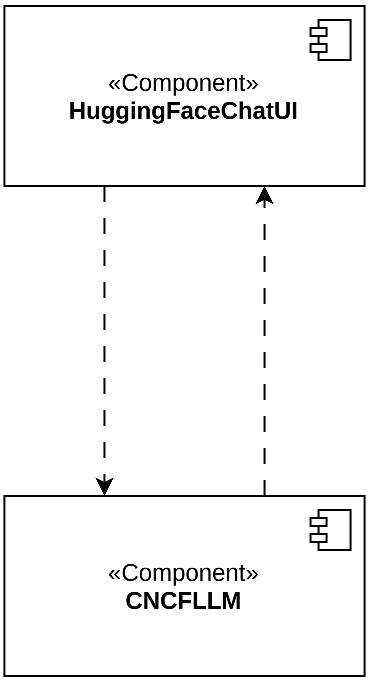
Software Architecture

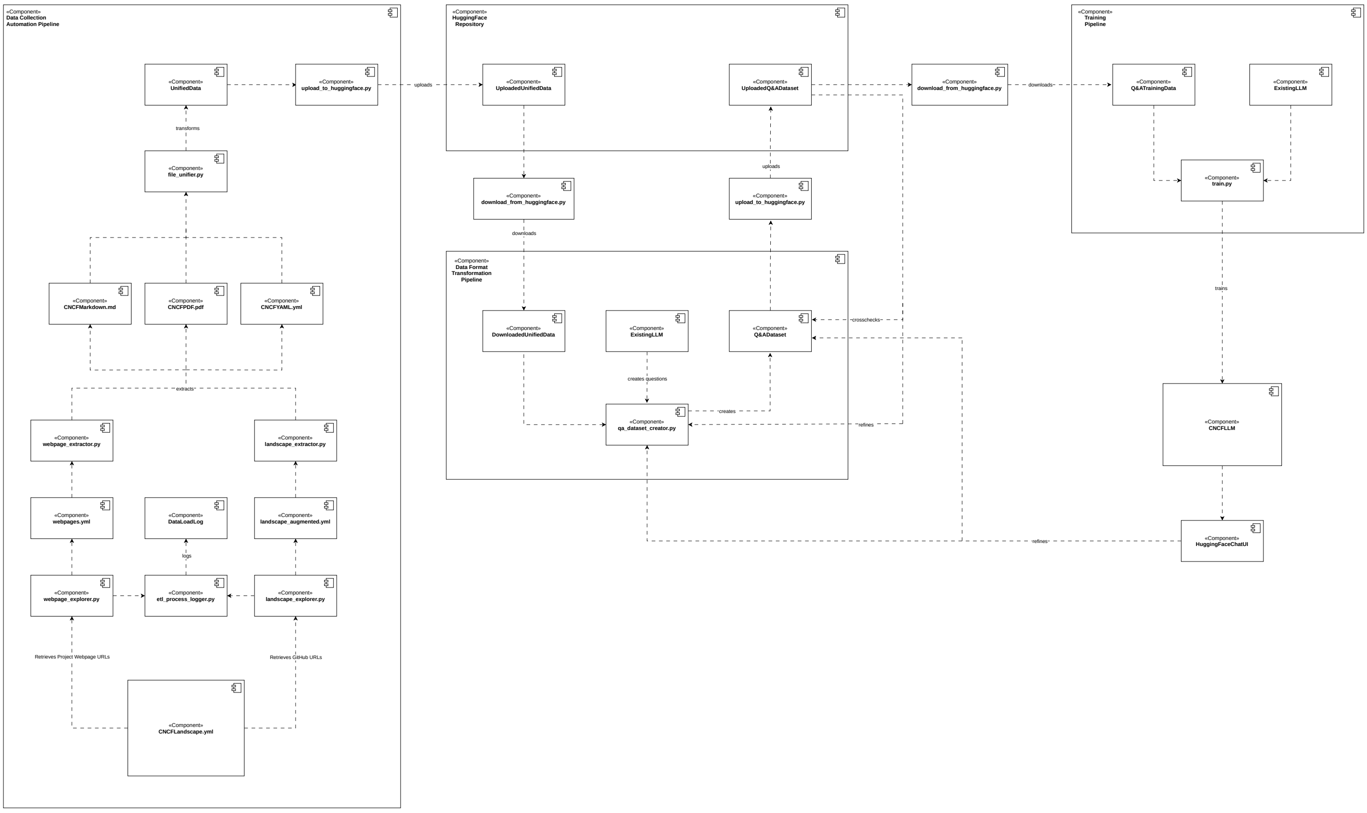
Runtime Components



- HuggingFaceChatUI: Open-Source ChatUI for Large Language Models
- CNCFLLM: Existing Large Language Model tuned with Data from Cloud Native Computing Foundation projects' documentation

The user can interact with the LLM via the ChatUI, asking CNCF related questions. The model then provides the user with elaborated answers.

Code Components


«Component» Data Collection Automation Pipeline
«Component» UnifiedData
«Component» upload_to_huggingface.py
uploads
transforms
«Component» file_unifier.py
«Component» CNCFMarkdown.md
«Component» CNCFPDF.pdf
«Component» CNCFYAML.yml
extracts
«Component» webpage_extractor.py
«Component» landscape_extractor.py
«Component» webpages.yml
«Component» DataLoadLog
«Component» landscape_augmented.yml
logs
«Component» webpage_explorer.py
«Component» etl_process_logger.py
«Component» landscape_explorer.py
Retrieves Project Webpage URLs
Retrieves GitHub URLs
«Component» CNCFLandscape.yml
«Component» HuggingFace Repository
«Component» UploadedUnifiedData
«Component» UploadedQ&ADataset
«Component» download_from_huggingface.py
downloads
«Component» upload_to_huggingface.py
uploads
«Component» Data Format Transformation Pipeline
«Component» DownloadedUnifiedData
«Component» ExistingLLM
«Component» Q&ADataset
crosschecks
creates questions
«Component» qa_dataset_creator.py
creates
refines
«Component» download_from_huggingface.py
downloads
«Component» Training Pipeline
«Component» Q&ATrainingData
«Component» ExistingLLM
«Component» train.py
trains
«Component» CNCFLLM
«Component» HuggingFaceChatUI
refines

Summary

- Data Collection Automation Pipeline: pipeline that is used to automatically extract, transform and load the CNCF data
    - CNCFLandscape.yml: regularly updated YAML File containing a list of URLs to Markdown, pdf and YAML files of all the projects in the CNCF landscape
    - landscape_explorer.py: script to retrieve GitHub URLs to Markdown, pdf and YAML files of the projects listed in the CNCF landscape and store them in a YAML file
    - webpage_explorer.py: script to retrieve Webpage URLs of the projects listed in the CNCF landscape and store them in a YAML file
    - etl_process_logger.py: script to log the amount of loaded data and save the log in a DataLoadLog file
    - DataLoadLog: file containing the log of the amount of loaded data
    - webpages.yml: YAML file containing the retrieved URLs to the projects' webpages
    - landscape_augmented.yml: YAML file containing the retrieved URLs to Markdown, pdf and YAML files of the projects
    - webpage_extractor.py: script to extract information listed on the projects' webpages and save it in an appropriate file format (e.g. pdf)
    - landscape_extractor.py: script to extract Markdown, pdf and YAML files of the projects and save them in an appropriate file format (pdf, yml, md)
    - CNCFMarkdown.md: markdown file containing the content of the markdown files of the projects
    - CNCFPDF.pdf: pdf file containing the content of the pdf files of the projects
    - CNCFYAML.yml: YAML file containing the content of the YAML files of the projects
    - file_transformer.py: script to transform the extracted data and save it in an appropriate file format
    - TransformedData: file containing the transformed extracted data
    - ner_knowledge_extractor.py: script performing NER knowledge extraction

- Training Pipeline: pipeline that is used to train an existing LLM model with the extracted and transformed CNCF data
    - TrainingData: file consisting of the ner knowledge extracted data that is used for training an existing model
    - ExistingLLM: LLM that is used as a base model for the CNCF model (e.g. Google GEMMA)
    - train.py: script to train the exisiting llm model with the ner extracted data
- CNCFLLM: Model trained with extracted CNCF data that is able to answer CNCF related prompts
- HuggingFaceChatUI: User Interface that uses the CNCF LLM and allows for user interaction with the model

So far, our project's code structure consists of two main pipelines, the data collection automation pipeline which combines all procedures needed to automatically extract the CNCF data from the internet, and the training pipeline which tunes an exisiting LLM with the extracted CNCF data so that it's able to elaborately answer queries related to CNCF projects. The resulting model is then loaded into the open-source HuggingFaceChatUI in order to provide a user interface that allows for user interaction with the model.

Architecture Choices

Programming language

We use python for gathering and extracting the CNCF landscape data as it's a modern language that provides us with numerous packages we can use for a fast and easy ETL process

Machine learning library

We use Pytorch for training the existing model as it provides us with an easy to use interface and is well-documented

AI Model

GEMMA LLM is our first choice for an existing LLM to build upon, as it's a performant model, however we'll might try out other alternatives if they fit better later on

User Interface

We decided on the HuggingFaceChatUI as our Chat User Interface as it's open source, easy-to-use, and popular. As our main work consists of creating a well-functioning model, we decided to use an exisiting AI chat interface. As it's open source, we can add functionalities if we'll need them in the future.

Technology Stack

Frontend

- HuggingFaceChatUI: TypeScript, Svelte

Backend

- Python
- Pytorch

- Python Mock
- Python Logging