C4 Architecture: FAQ Assistant

This document describes the software architecture of the FAQ Assistant.

Level 1: System Context

System: FAQ Assistant

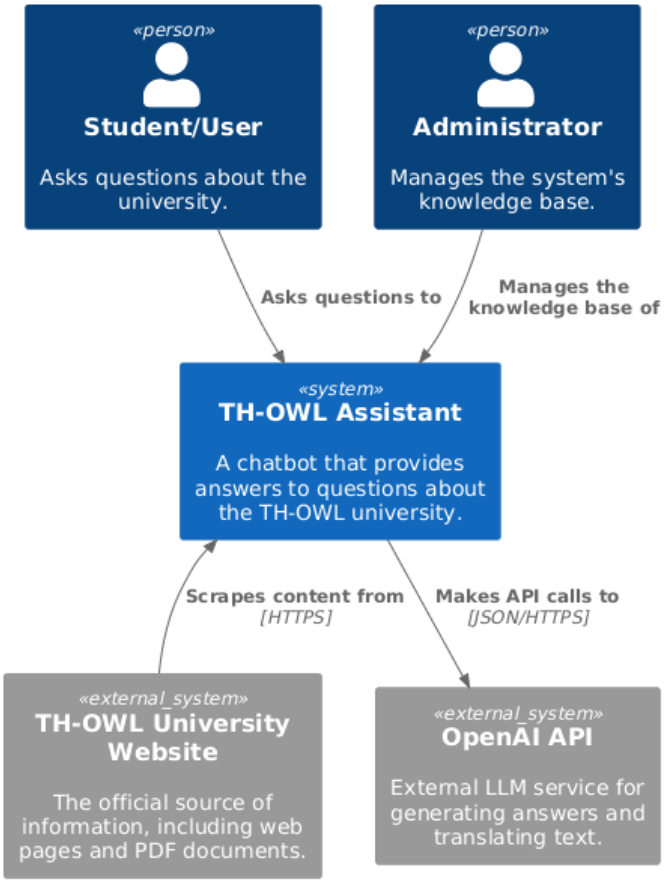
A chatbot application that answers questions about the TH-OWL university using official documentation.

Actors (Users):

- Student/User: The primary user who asks questions in the chat interface.
- Administrator: A technical user who manages the system's knowledge base by rebuilding the search index.

External Systems:

- TH-OWL University Website: The official source of information (th-owl.de), providing the web pages and PDF documents for indexing.
- OpenAI API: An external Large Language Model (LLM) service used for generating human-like answers and translating text.



Level 2: Containers

Technology Stack:

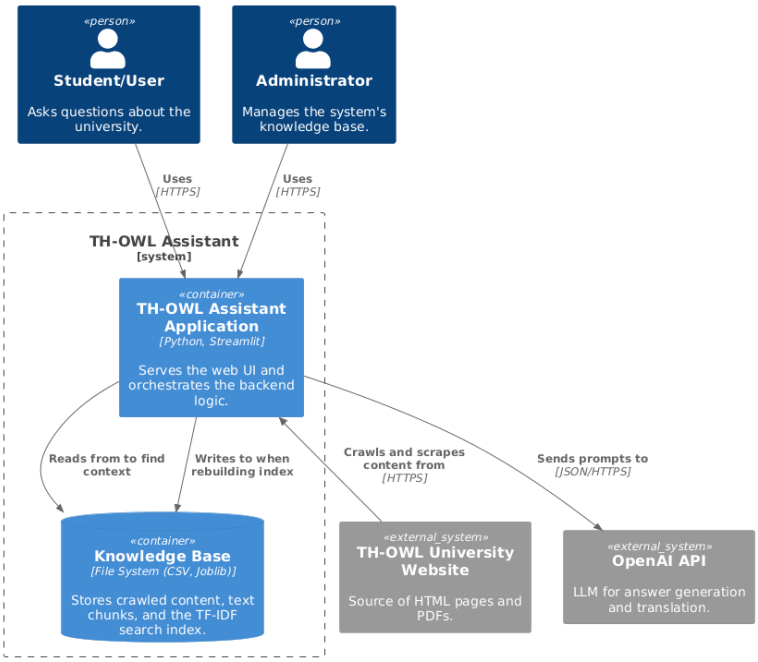
- Frontend: Streamlit (Python web framework)
- Backend: Python 3.10
- AI Service: OpenAI GPT API
- Data Storage: Local filesystem (CSV + Joblib)
- Deployment: Cloud hosting (Streamlit Cloud)

Containers:

- TH-OWL Assistant Application: A monolithic Python application (built with Streamlit) that serves the web UI and contains all the logic for processing user requests. It is the system's single deployable unit.
- Knowledge Base: A set of files on the local file system (CSV for data, Joblib for the model) that stores the indexed knowledge. This acts as the system's long-term memory.

Interactions:

- Users interact with the TH-OWL Assistant Application via their web browser.
- The application reads from and writes to the Knowledge Base to search for information or rebuild the index.
- The application makes secure API calls to the external OpenAI API and sends requests to the TH-OWL University Website during indexing.



Level 3: Components

This diagram zooms into the TH-OWL Assistant Application container to show its internal components, which are logical modules within the codebase.

Components:

- Web UI: Renders the chat interface and admin panel using the Streamlit API.
- Chat Controller: Orchestrates the process of receiving a question and delivering an answer.
- Language Service: Handles language detection (langdetect) and translation tasks.
- Retrieval Service: Implements the search functionality to find relevant text chunks from the Knowledge Base using a TF-IDF model (scikit-learn).
- LLM Service: Acts as a client for the OpenAI API, constructing prompts and handling API calls.
- Indexing Service: An admin-only component that crawls web pages and PDFs to build the search index.