1. Introduction

Introduce the CS 410 product and the approach to demonstrating its characteristics through prototyping (essentially an abstract). This section should

- be approximately one page in length.
- summarize the societal problem with some reference/date support.
- outline problems and needed solution characteristics.
- introduce your product by name as the solution.

2. CueCode Product Description

Provide a top-level description of CS 410 product for the average reader. Provide a summary of the solution –- and its goals and objectives. This section should be one paragraph minimum.

- With the rapidly evolving landscape of artificial intelligence and software development, enterprises and Software-as-a-Service (Saas) providers encounter significant challenges when integrating large language models (LLMs) into their operation. Even though LLMs are remarkably good at producing language that seems human, the existing solutions does not have the means to integrate consistent business logic and human judgment into the API message production process. For businesses that depend on precise and secure API call execution for data entry and action triggering, this gap creates a high-risk environment.
- Further, getting an LLM to produce Web API payloads and validating them is a specialized task requiring skills that many Web and fullstack developers do not possess. Since these developers are those most often building business applications, a solution for turning natural language into REST API payloads should take into consideration how easy it will be for developers with other skillsets to use the tool. CueCode gives a good foundation for Web and fullstack developers to turn natural language into REST API payloads.

2.1. Key Product Features and Capabilities

What does it do? What is significant/unique/innovative about it? What does it accomplish? Describe how this solves the problem.

What does it do

- CueCode will implement Natural Language Processing (NLP) capabilities to enable and understand natural language.
- Will offer a user friendly interface (API client libraries) that developers can use.
- Will provide a developer portal web application, where developers can upload OpenAPI specifications for their APIs and configure their CueCode service.

Significance, Uniqueness and Innovation

- CueCode is a full framework and service to turn natural language into Web API payloads; nothing like it exists for arbitrary REST APIs. CueCode will work with any REST API defined with an OpenAPI specification.
- CueCode will offer non-technical staff and customers the ability to interact with complex APIs using simple language. This improves access to technology. CueCode's ability to work with existing API documentation allows it to adapt to various data structures without requiring significant reconfiguration.
- It will allow versatility for enterprises that utilize multiple systems and services.

What It Accomplishes

- It will increase efficiency, allowing quick and accurate REST API interactions based on simple language inputs. Organizations can respond to customer requests more quickly, enhancing service levels.
- With the enhanced user experience, client service representatives can focus on customer engagement rather than getting overwhelmed with technical details. This will lead to better customer satisfaction

Problem Solution

- Cuecode makes NLP and LLM generation capabilities accessible for non-specialist developers, while applying it to the problem of turning natural language into REST API payloads. By abstracting the technical complexity of generating the API payloads, CueCode effectively bridges the gap between human input and technical execution of their requests via REST API calls.

2.2. Major Components (Hardware/Software)

Provide an overview of the hardware needed to support the solution. Describe how it is structured based on CS 410 MFCD. Define and describe the software to be developed.

CueCode will require hardware capable of running the following systems separately:

- Ollama 3.1 (minimum) running the 70 billion parameter model (minimum)
    - The CS Systems group already runs Llama models.
- Spacy.io running on either CPU or GPU.

3. Identification of Case Study

For whom is this product being developed? Why? Who else might use this in the future?

- Case study of Steve, fullstack developer who needs to integrate text-to-API-payload features.
- Case study of Patricia, who needs to make an appointment at a hospital.

4. X Product Prototype Description

Provide a top-level description of the CS 411W prototype as it relates to the end product from CS 410 (i.e., the goal). Are capabilities reduced or eliminated? Simulated – modeled?> Include a table of comparison between RWP and Prototype either in section 4, 4.1 or 4.2

4.1. Prototype Architecture (Hardware/Software)

How will the prototype be structured to demonstrate key features of the CS 410 product. Provide describe the Prototype MFCD.

4.2. Prototype Features and Capabilities

What does the prototype demonstrate? Why is that significant in showing how the problem is solved? How you have demonstrated success? How does the prototype address the CS 410 project risk mitigation? Describe the functional goals and objectives.

4.3. Prototype Development Challenges

Describe the expected challenges to be encountered while completing the prototype – e.g., knowledge missing, capability missing, supporting technology issues.

5. Glossary

API Payload (informal): Information that is sent together with an API request or response. This data, which can be organized in JSON or XML forms, usually includes the details needed by the client to comprehend the answer or by the server to carry out an action.

CueCode Developer Portal: A web-based platform that allows easy API creation with NLP-generated requests and gives developers access to CueCode's tools, API configuration, and integration workflow management.

HTTP Header: Additional metadata, such as the content type, authentication information, or caching instructions, are transmitted with HTTP requests and answers. Headers give context, which improves communication.

HTTP (Hypertext Transfer Protocol): The protocol that specifies the format and transmission of messages between web clients and servers. The type of request is determined by the HTTP methods (GET, POST, etc.).

Representational State Transfer (REST): A set of design guidelines for networked apps that use stateless, cacheable, and consistent HTTP processes to facilitate interaction. Through the use of common HTTP techniques, REST allows clients to communicate with servers by modifying resources that match an expected structure.

URL (Uniform Resource Locator): A web address that indicates where a resource is located on the internet. Protocol (such as HTTP/HTTPS), domain, and resource path are all included in URLs. They are necessary in order to access and consult internet resources.

6. List of tables

TODO: risk matrix/matrices

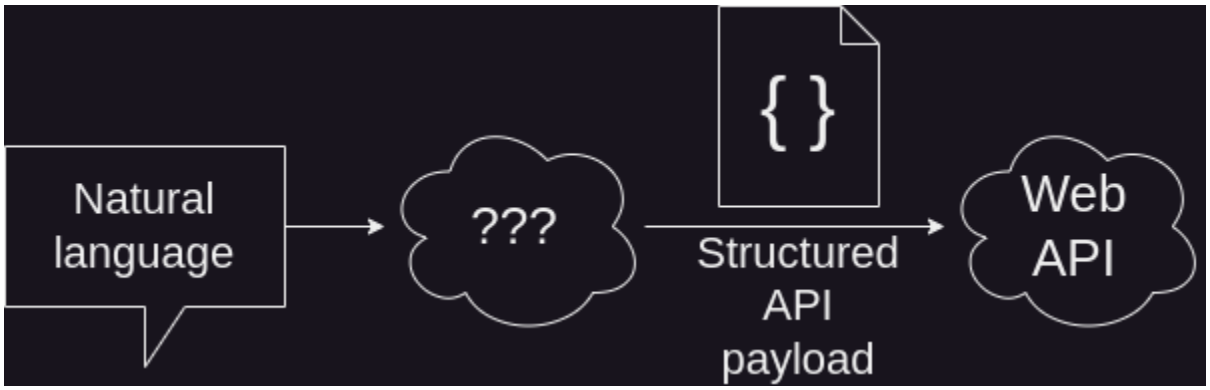
Feature	CueCode	OpenAI Functions	Google Natural Language API	Spacy.io	LangChain	GenKit	Phone AI Alexa, Siri,...
Entity recognition	✔		✔	✔		**P**	✔
Plug and Play	✔				**P**	**P**	**P**

Retrieval Augmented Generation	✔	✔			✔	✔	
API call generation as a service	✔	**P**	**P**		**P**		**P**

Competition matrix, showing features partially or fully implemented in CueCode's Competitors. Demonstrates CueCodes complete feature set as a framework for turning natural language into REST API payloads.

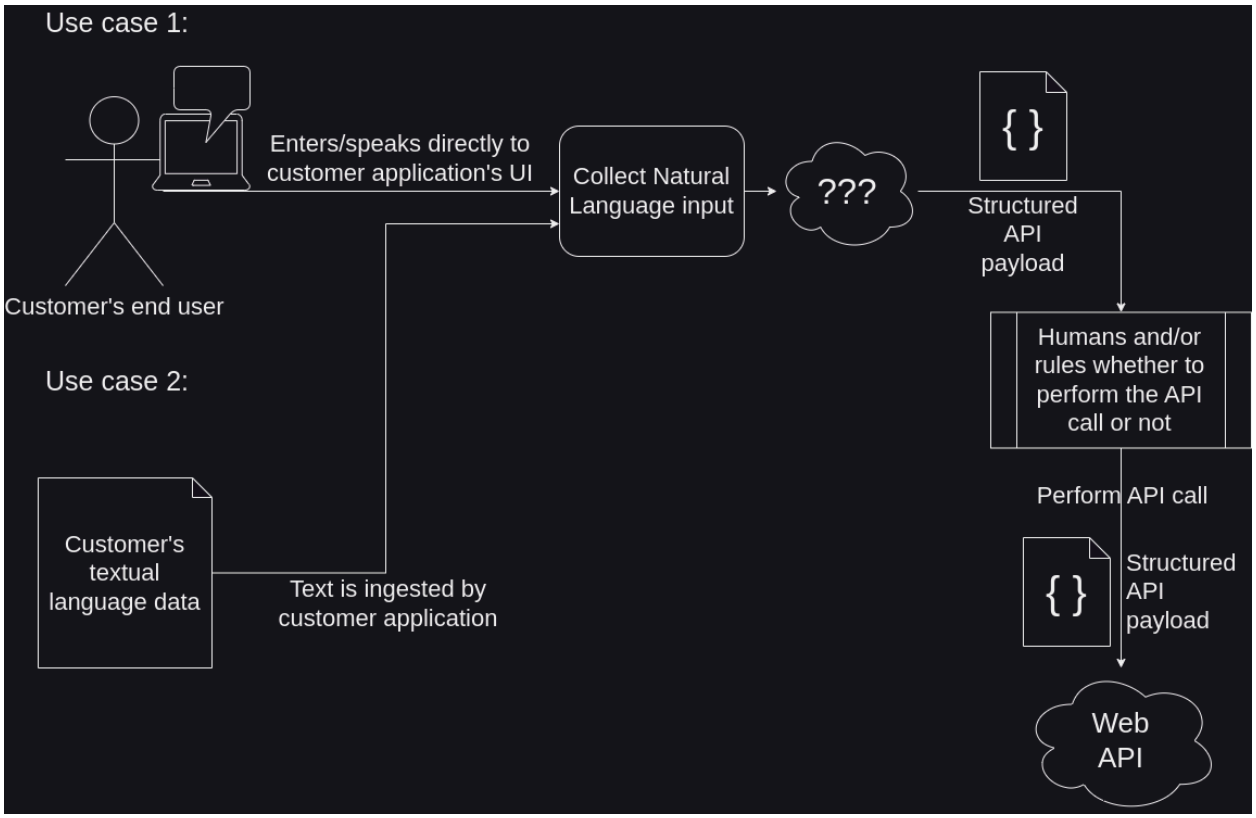
7. List of figures

CueCode logo

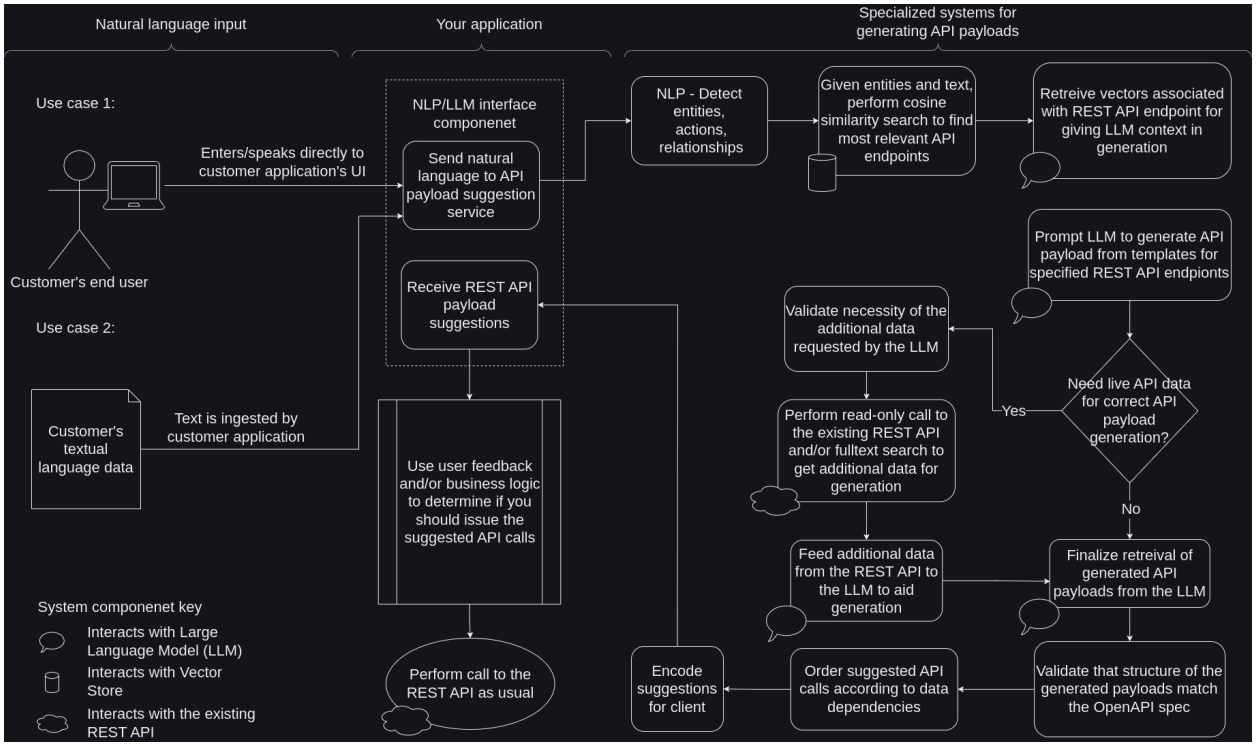


Conceptual diagram of turning natural language into Web API payloads, with the NLP component left a mystery.

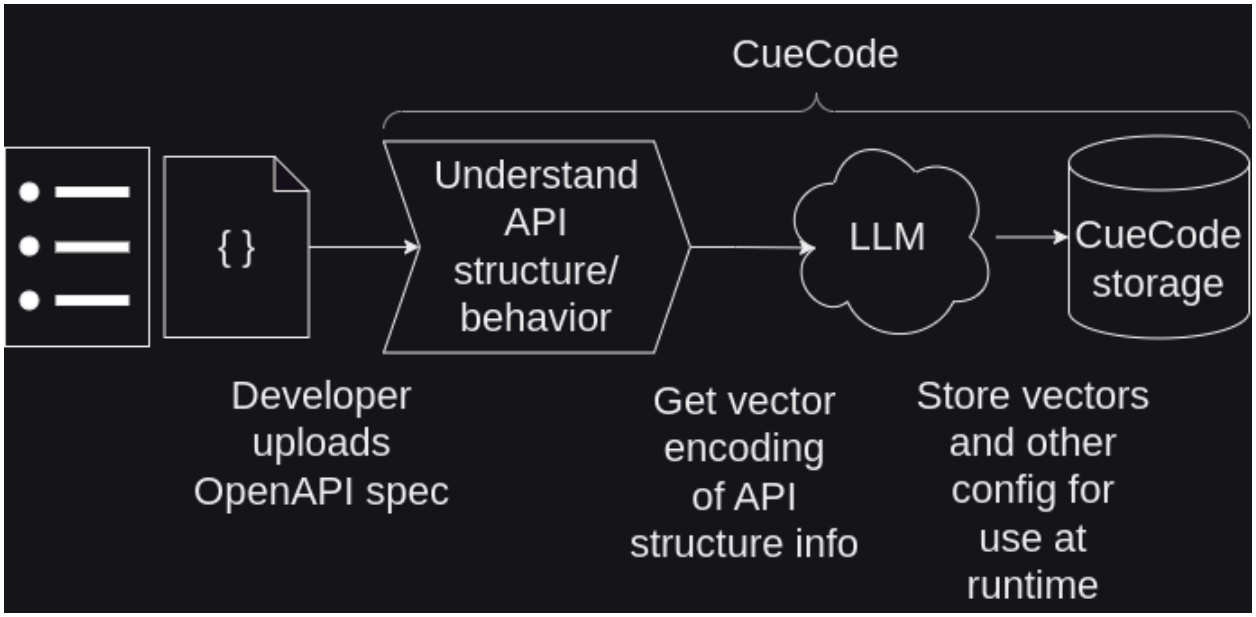
Conceptual diagram of turning natural language into Web API payloads, with CueCode shown as the NLP component.



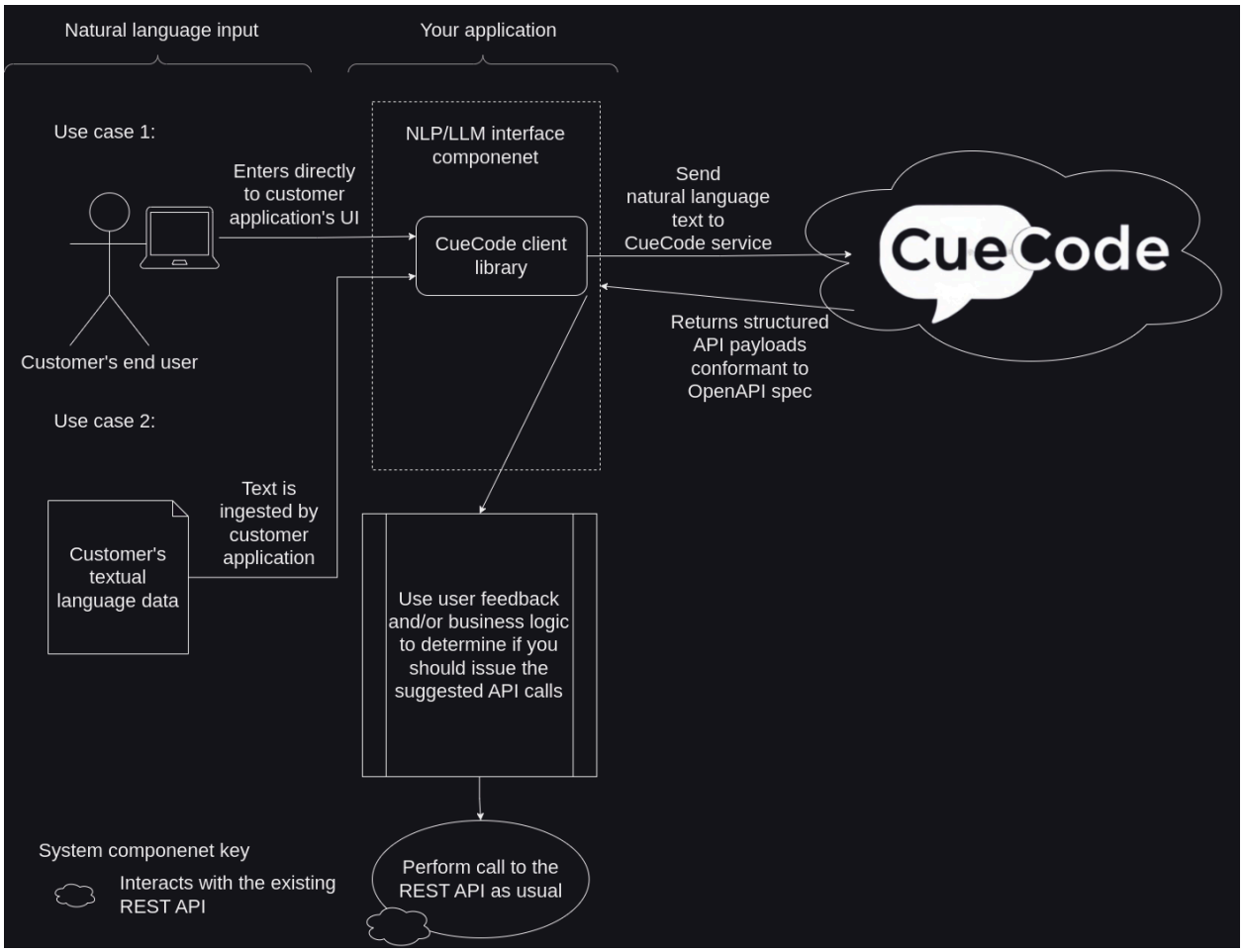
Conceptual diagram of two example use cases where a customer can use CueCode to include validation of generated API payloads.



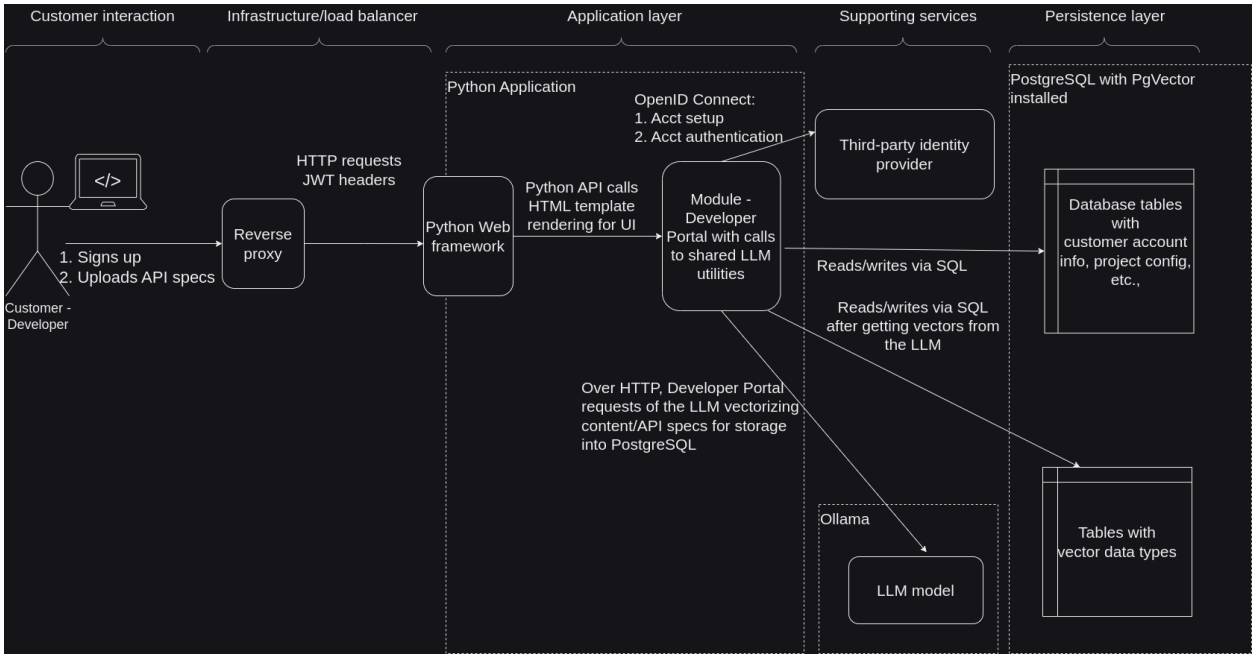
Current process flowchart for engineering REST API generation with OpenAPI specifications, showing the two example customer use cases from other slides. Major system components involved at each process step are labeled with icons.



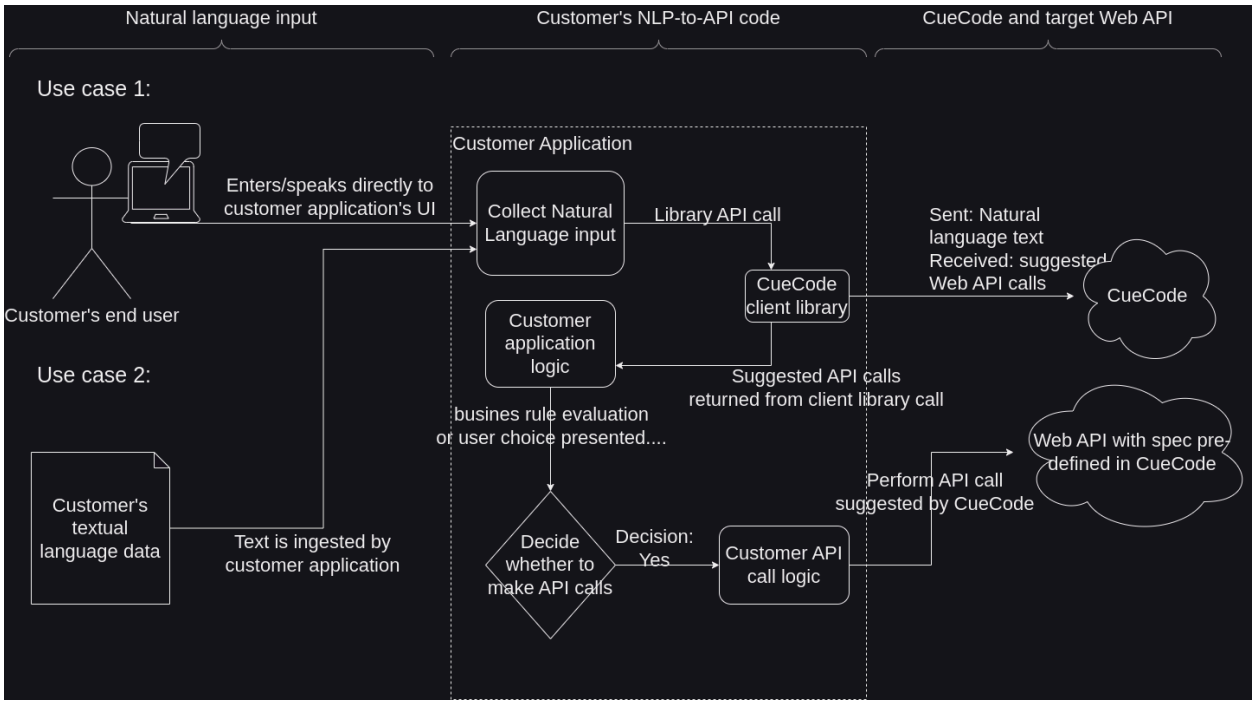
The process flowchart for configuring CueCode with an OpenAPI specification.



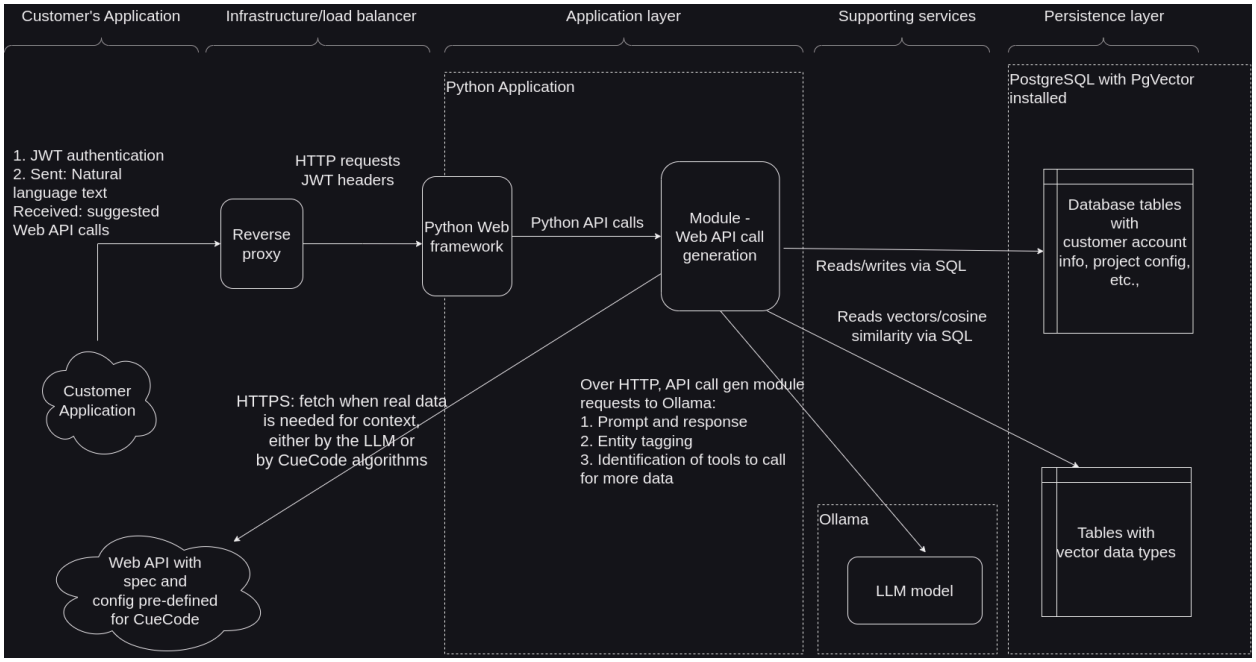
The process of turning natural language to REST API calls when using CueCode. Diagram includes validation and the two example use cases.



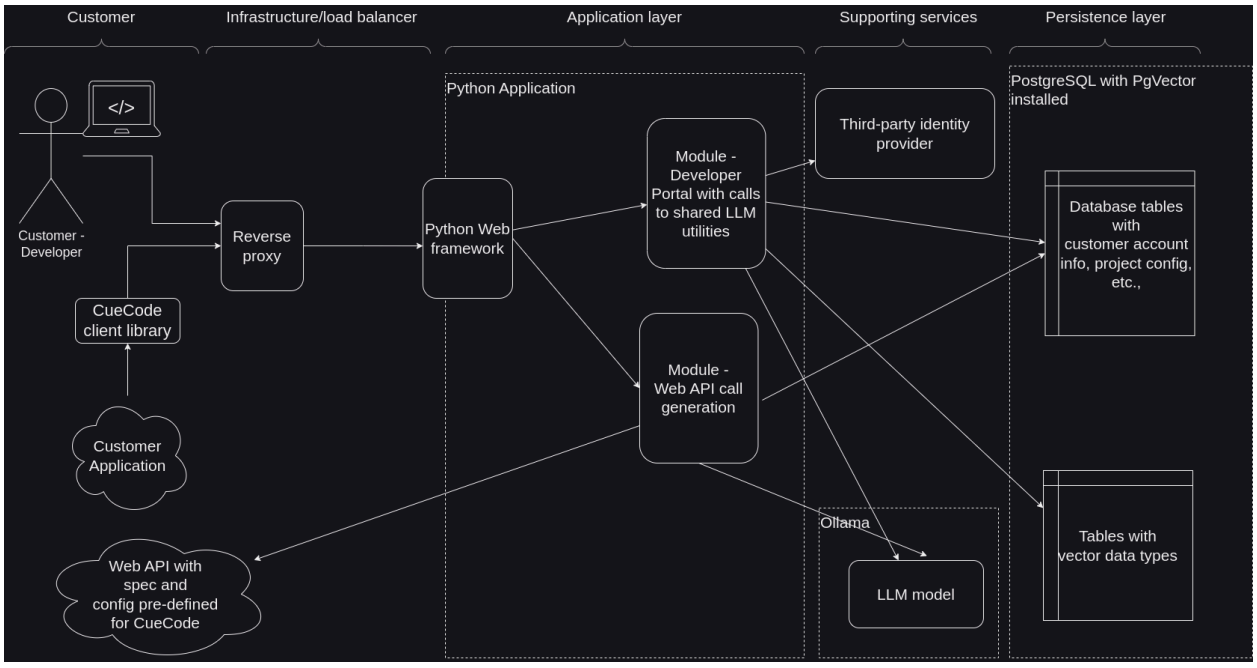
Major functional components diagram, focusing on customer interactions and components involved in the Developer Portal and CueCode configuration process.



Major functional components diagram, showing how the customer's application can use CueCode to turn natural language to REST API payloads, perform validation of the payloads, then issue the suggested REST API calls.



Major functional components diagram, focusing on the customer's application interacting with CueCode and components involved in the API payload generation process.



An overview of all major functional components involved.

8. References

About continuous integration with GitHub Actions. (n.d.). GitHub Docs. Retrieved October 22, 2024, from https://docs.github.com/en/actions/about-github-actions/about-continuous-integration-with-github-actions

About Git. (n.d.). GitHub Docs. Retrieved October 22, 2024, from https://docs.github.com/en/get-started/using-git/about-git

Against LLM maximalism · Explosion. (2023, May 18). https://explosion.ai/blog/explosion.ai

AppDirect | Developer Portal. (2024). Appdirect.com. https://developer.appdirect.com/

Au-Yeung, J. (2020, March 2). Best practices for REST API design. Stack Overflow Blog. https://stackoverflow.blog/2020/03/02/best-practices-for-rest-api-design/

Baker, S. (2024). *Paragonsean/ChatBotAsync* [Python]. https://github.com/paragonsean/ChatBotAsync (Original work published 2024)

Cloud Natural Language. (n.d.). Google Cloud. Retrieved September 26, 2024, from https://cloud.google.com/natural-language

Evaluation | Genkit. (n.d.). Firebase. Retrieved September 14, 2024, from https://firebase.google.com/docs/genkit/evaluation

Firebase Genkit. (n.d.). Retrieved September 14, 2024, from https://firebase.google.com/docs/genkit

Function Calling. (n.d.). Retrieved September 14, 2024, from https://platform.openai.com/docs/guides/function-calling

HTTP headers - HTTP | MDN. (n.d.). Developer.mozilla.org. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers

Learn Data with Mark (Director). (2023, July 26). *Returning consistent/valid JSON with OpenAI/GPT* [Video recording]. https://www.youtube.com/watch?v=lJJkBaO15Po

Lorica, B. (2024, January 25). *Expanding AI Horizons: The Rise of Function Calling in LLMs*. Gradient Flow. https://gradientflow.com/expanding-ai-horizons-the-rise-of-function-calling-in-llms/

Merritt, R. (2023, November 15). *What Is Retrieval-Augmented Generation aka RAG?* NVIDIA Blog. https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/

Microsoft/prompt-engine. (2024). [TypeScript]. Microsoft. https://github.com/microsoft/prompt-engine (Original work published 2022)

Natural Language Processing [NLP] Market Size | Growth, 2032. (n.d.). Retrieved September 14, 2024, from https://www.fortunebusinessinsights.com/industry-reports/natural-language-processing-nlp-market-101933

OpenAI Platform. (n.d.-a). Retrieved September 10, 2024, from https://platform.openai.com

OpenAI Platform. (n.d.-b). Retrieved October 24, 2024, from https://platform.openai.com

OpenAPI Specification—Version 3.1.0 | Swagger. (n.d.). Retrieved September 10, 2024, from https://swagger.io/specification/

OpenAPITools/openapi-generator. (2024). [Java]. OpenAPI Tools. https://github.com/OpenAPITools/openapi-generator (Original work published 2018)

piembsystech. (2023, October 2). *Dynamic Binding in Python Language*. PiEmbSysTech. https://piembsystech.com/dynamic-binding-in-python-language/

Scarpati, J. (n.d.). What is a URL (Uniform Resource Locator)? SearchNetworking. https://www.techtarget.com/searchnetworking/definition/URL

SpaCy · Industrial-strength Natural Language Processing in Python. (n.d.). Retrieved September 26, 2024, from https://spacy.io/

Stanfordnlp/dspy. (2024). [Python]. Stanford NLP. https://github.com/stanfordnlp/dspy (Original work published 2023)

Su, Y., Awadallah, A. H., Khabsa, M., Pantel, P., Gamon, M., & Encarnacion, M. (2017). Building Natural Language Interfaces to Web APIs. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 177–186. https://doi.org/10.1145/3132847.3133009

Tool/function calling | LangChain. (n.d.). Retrieved September 14, 2024, from https://python.langchain.com/v0.1/docs/modules/model_io/chat/function_calling/

Tutorial: ChatGPT Over Your Data. (2023, February 6). LangChain Blog. https://blog.langchain.dev/tutorial-chatgpt-over-your-data/

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2023). *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models* (arXiv:2201.11903). arXiv. http://arxiv.org/abs/2201.11903

What Is NLP (Natural Language Processing)? | IBM. (2021, September 23). https://www.ibm.com/topics/natural-language-processing

What is Representational State Transfer (Rest) API? - Ampcontrol. (2024). Ampcontrol.io. https://www.ampcontrol.io/ev-terminology/what-is-rest-api

Why Visual Studio Code? (n.d.). Retrieved October 22, 2024, from https://code.visualstudio.com/docs/editor/whyvscode

W3Schools. (n.d.). HTTP Methods GET vs POST. W3schools.com. https://www.w3schools.com/tags/ref_httpmethods.asp

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). *ReAct: Synergizing Reasoning and Acting in Language Models* (arXiv:2210.03629). arXiv. http://arxiv.org/abs/2210.03629

Zafin, E. at. (2023, August 15). Bridging the Gap: Exploring use of Natural Language to interact with Complex Systems. *Engineering at Zafin*.

https://medium.com/engineering-zafin/bridging-the-gap-exploring-using-natural-language-to-interact-with-complex-systems-11c1b056cc19