Lab 2 - CUECODE PRODUCT SPECIFICATION

Freddie Boateng

Old Dominion University

CS411W

Dr. Sumaya Sanober

March 9, 2025

Table of Contents

1. Introduction

In today's fast-paced digital world, users expect seamless, intuitive experiences with applications that communicate in natural language. However, there is a gap between how users interact with applications and how these applications operate in reality. Developers face significant challenges when attempting to integrate natural language processing (NLP) and large language models (LLMs) into their applications (Lorica, 2024). These challenges arise due to a lack of mature, risk-aware tooling to facilitate the integration of natural language with application interfaces. CueCode is designed to bridge this gap by providing developers and businesses with a scalable, secure, and efficient solution for leveraging NLP and LLMs.

The software will enable non-technical staff to interact with APIs using simple, natural language commands, simplifying the process of integrating complex systems and services. CueCode's goal is to create a platform that removes the barriers typically faced by developers when incorporating AI-driven applications, ensuring both safety and scalability. By offering tooling that translates natural language into API payloads, CueCode will help businesses confidently integrate NLP capabilities into their systems without compromising the integrity of their business logic.

1.1 Purpose

The purpose of CueCode is to simplify and streamline the integration of natural language processing (NLP) and large language models (LLMs) into applications, providing a solution that both developers and non-technical staff can use effectively. By allowing users to interact with APIs through natural language commands, CueCode eliminates the need for extensive technical knowledge, enabling non-developers to leverage powerful AI capabilities (Lorica*, 2024*). This functionality bridges the gap between the technical complexity of API interactions and the user-friendly, intuitive experiences expected by modern users. With CueCode, businesses can incorporate AI-driven features into their applications with minimal friction, making AI more accessible and easier to implement across various teams and departments.

In addition to simplifying integration, CueCode is designed to ensure security and scalability, helping businesses harness the full potential of NLP and LLMs without compromising the integrity of their systems or exposing them to unnecessary risks. The software provides a risk-aware development framework, ensuring that AI-driven integrations do not interfere with or corrupt critical business logic. CueCode empowers organizations to scale their use of AI applications confidently, offering seamless compatibility with existing enterprise systems while maintaining robust security and control. Ultimately, the purpose of CueCode is to provide businesses with a tool that makes it easy to implement advanced AI capabilities safely.

1.2 Scope

CueCode is designed to simplify the integration of natural language processing (NLP) and large language models (LLMs) into applications, targeting both technical and non-technical users. The platform enables non-technical staff to interact with APIs using simple, natural language instructions, eliminating the need for complex technical expertise. It aims to provide businesses with a secure and scalable solution for incorporating AI-driven applications, ensuring the integrity of business logic while fostering seamless integration across multiple systems and services. By translating natural language into structured API payloads, CueCode allows enterprises to confidently harness the power of NLP without the risk of compromising system stability or introducing unnecessary complexity. The solution is built to support a wide range of systems, enabling businesses of all sizes to scale their use of AI technologies efficiently and safely.

1.3 Definitions, Acronyms, and Abbreviations

- **API (Application Programming Interface)**: A set of rules and protocols that allow different software applications to communicate with each other.
- **LLM (Large Language Model)**: A type of artificial intelligence model trained on vast amounts of text data to understand and generate human language.
- **NLP (Natural Language Processing)**: A field of artificial intelligence that focuses on the interaction between computers and human language, enabling machines to understand, interpret, and generate natural language.
- **CueCode**: The software platform designed to simplify and secure the integration of NLP and LLMs into applications by allowing non-technical staff to interact with APIs using natural language.
- **SaaS (Software as a Service)**: A cloud-based service model where applications are hosted and made available to users over the internet, often on a subscription basis.
- **REST (Representational State Transfer)**: An architectural style for designing networked applications, often used for building web APIs that allow different systems to communicate over HTTP.
- **GraphQL**: A query language for APIs and a runtime for executing those queries with existing data, offering more flexibility compared to traditional REST APIs.

- **UI (User Interface)**: The graphical and interactive elements of a software system through which users interact with the application, such as buttons, text fields, and menus.
- **Business Logic**: The part of a software application that handles the processing of data according to the rules and requirements of a business or organization.
- **Payload**: The actual data sent in an API request or response, often in a structured format like JSON or XML, containing information to be processed or acted upon by the receiving system.
- **Cloud-based**: Refers to software or services that are hosted and delivered over the internet, typically on cloud platforms such as AWS, Azure, or Google Cloud, rather than being installed on local servers.
- **On-premise**: Refers to software or services that are installed and run on a company's own infrastructure, within its physical premises, rather than being hosted on the cloud.

1.4 References

2 *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/ [12]*

3 *Mark Needham (Director). (2023, July 26). Returning consistent/valid JSON with OpenAI/GPT [Video recording]. https://www.youtube.com/watch?v=lJJkBaO15Po [13]*

4 *Merritt, R. (2023, November 15). What Is Retrieval-Augmented Generation aka RAG? NVIDIA Blog. https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/ [14]*

5 *Microsoft/prompt-engine. (2024). [TypeScript]. Microsoft. https://github.com/microsoft/prompt-engine (Original work published 2022) [15]*

6 *Prabhakaran, S. (2018, October 22). Cosine Similarity - Understanding the math and how it works? (With python). Machine Learning Plus. https://www.machinelearningplus.com/nlp/cosine-similarity/*

7 *Rao, P. (2024, January 24). Turbo-charge your spaCy NLP pipeline. Medium. https://towardsdatascience.com/turbo-charge-your-spacy-nlp-pipeline-551435b664ad*

8 Prabhakaran, S. (2018, October 22). Cosine Similarity - Understanding the math and how it works? (With python). *Machine Learning Plus*. https://www.machinelearningplus.com/nlp/cosine-similarity/

9 Rao, P. (2024, January 24). *Turbo-charge your spaCy NLP pipeline*. Medium. https://towardsdatascience.com/turbo-charge-your-spacy-nlp-pipeline-551435b664ad

10 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

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

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

1.5 Overview

CueCode is a software solution designed to simplify the integration of Natural Language Processing (NLP) and Large Language Models (LLMs) into applications. By enabling non-technical users to interact with APIs using natural language, CueCode bridges the gap between technical complexity and user expectations for intuitive digital experiences. The platform aims to enhance the development and deployment of AI-driven applications, offering businesses a secure, scalable, and risk-aware framework. This allows enterprises to adopt NLP technologies with confidence, avoiding the typical challenges that arise from incorporating AI-driven features while maintaining system integrity and business logic.

2. Overall Description

CueCode is a tool that simplifies the interaction between users and complex systems by allowing them to communicate in plain, natural language. It converts these natural language inputs into API requests, thus enabling non-technical staff to engage with APIs without needing extensive technical knowledge. The software is designed to work with a variety of API types, including RESTful APIs, GraphQL, and more, offering versatility for enterprises using multiple systems. It also provides a secure and scalable solution for integrating AI-driven capabilities, ensuring that business logic remains intact and system stability is maintained. The platform empowers businesses to integrate NLP features and LLM capabilities into their applications while mitigating risks associated with the technical challenges of AI integration.

2.1 Product Perspective

The architecture of CueCode is designed with scalability, security, and ease of use in mind. The system comprises three main components: the user interface (UI), the natural language processing module, and the API integration layer. The UI is a web-based platform that allows users to input natural language commands. This input is sent to the NLP module, where it is processed and converted into a structured API request. The API integration layer then handles the interaction with external services, executing the API requests and returning the appropriate responses. The architecture is cloud-based, enabling seamless scaling and ensuring that businesses of all sizes can leverage the platform. Additionally, the system includes robust security protocols, such as encryption and role-based access controls, to ensure that data and business logic are protected at all times. This architecture ensures that CueCode can be deployed across various environments while maintaining high availability and reliability.

2.2 Product Functions

CueCode's core functionality is centered around translating natural language commands into structured API requests that can be processed by various backend services. The system enables users to specify their desired actions or queries in simple language, which are then interpreted and converted into the appropriate API payloads. The platform’s NLP module uses machine learning models to understand the intent behind the input and map it to the corresponding API call. For example, a user might input a command such as "Get the latest sales data," and

CueCode will translate this into a query that fetches the required information from an API. Additionally, the system allows for customization and configuration, enabling developers to specify certain parameters and fine-tune the interaction logic. The platform also includes error handling and validation processes to ensure that API requests are accurate and that any issues are flagged for resolution, ensuring safe and reliable interactions with external services.

2.3 External Interfaces

CueCode interacts with a variety of external interfaces to perform its functions. First and foremost, it communicates with external APIs, which can be either public or private services depending on the enterprise's needs. These APIs can be built on various technologies, such as REST, GraphQL, or SOAP, and CueCode supports integration with these diverse systems. Additionally, the platform supports integration with cloud-based systems, offering flexibility for businesses using services from providers like AWS, Google Cloud, or Microsoft Azure. CueCode also includes interfaces for authentication and authorization, ensuring that only authorized users can interact with sensitive business data. The platform can be integrated with third-party NLP models and services, allowing for continuous improvement in language understanding and processing capabilities. Finally, CueCode offers user-facing interfaces that are web-based, providing an intuitive platform for non-technical staff to interact with the system and manage their API interactions through a simple, easy-to-use interface.

2.3.1 Hardware Requirements

- Support for cloud-based or on-premise deployments.
- Sufficient storage capacity to handle API interactions and NLP model storage.
- Scalability to accommodate varying enterprise needs.

2.3.2 Software Requirements

- Integration with popular API standards (e.g., REST, GraphQL).
- Support for NLP tools, libraries, and large language models.
- Secure authentication and authorization mechanisms.

2.3.3 Security Requirements

- Ensure that all user interactions and API payloads are secure and free from vulnerabilities.
- Implement access controls and role-based permissions for different user types (e.g., non-technical staff, developers, administrators).

3 Non-Functional Requirements

3.1 Performance

- CueCode should handle large volumes of API requests without significant latency.
- The system should be capable of scaling to accommodate growing usage within enterprise environments.

3.2 Usability

- The interface must be intuitive for non-technical users to easily understand and use.
- Developers should be able to easily configure and customize API interactions.

3.3 Reliability

- CueCode should have high uptime and fault tolerance to ensure continuous operation in business-critical environments.
- Provide backup and recovery mechanisms to safeguard data.

3.4 Scalability

- The platform must be scalable to meet the needs of both small businesses and large enterprises.
- Ensure compatibility with a wide range of systems and APIs.

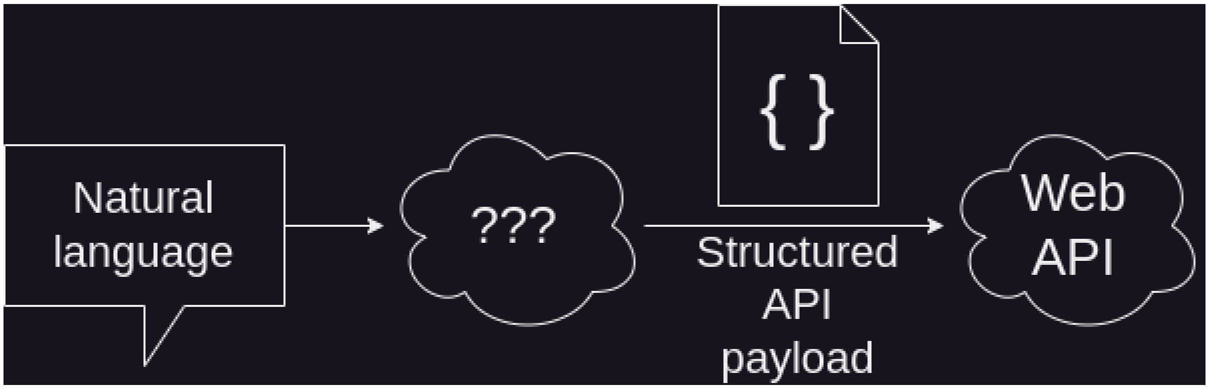
4. Conclusion

CueCode aims to provide a robust, scalable, and secure solution for businesses to leverage the power of NLP and LLMs without the complexities and risks typically associated with these technologies. By simplifying the process of interacting with APIs through natural language and offering a risk-aware development framework, CueCode will enable developers and non-technical staff to work with AI-driven applications confidently and efficiently.

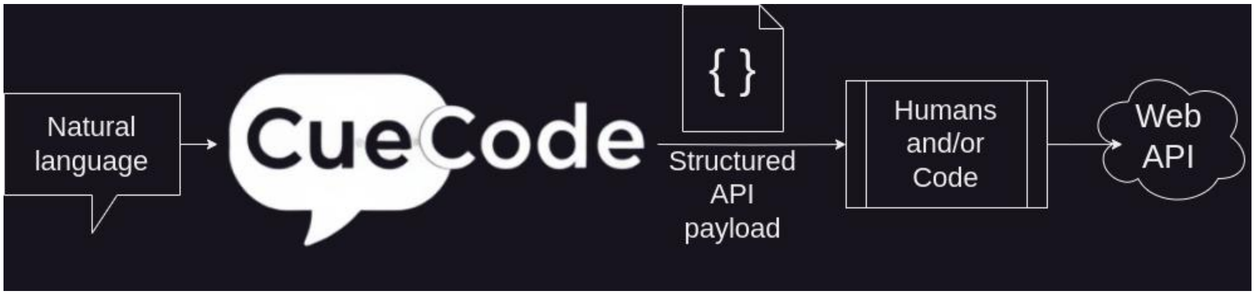
1 List Of Tables

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

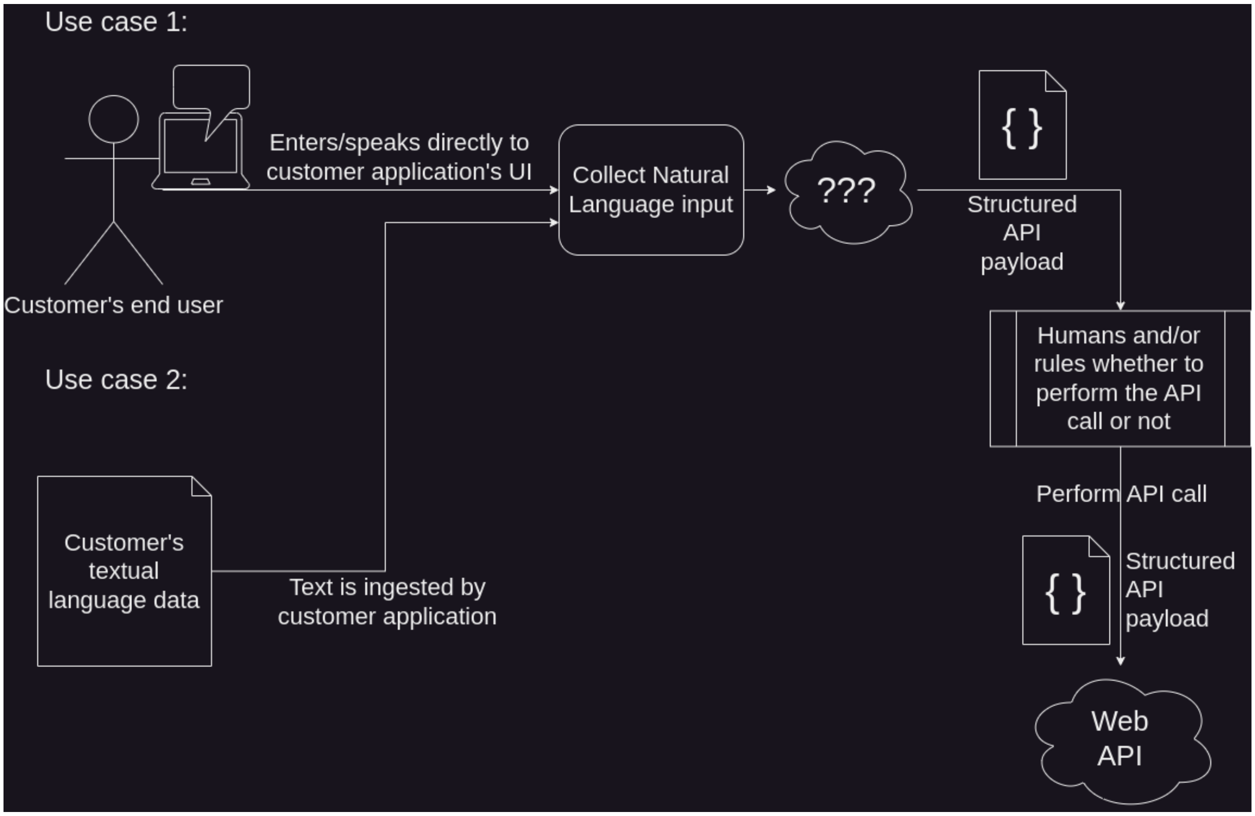
2 List Of Figures



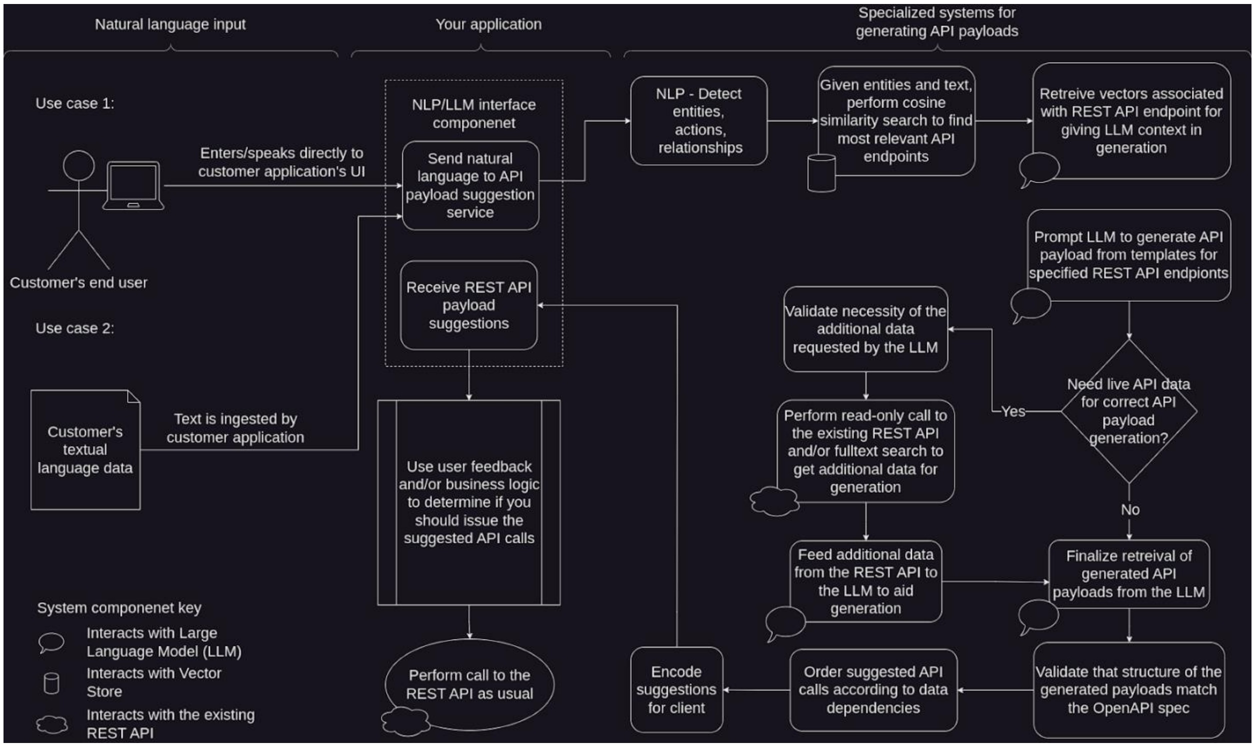
Conceptual diagram of turning natural language into Web API payloads, with the NLP component left a mystery.



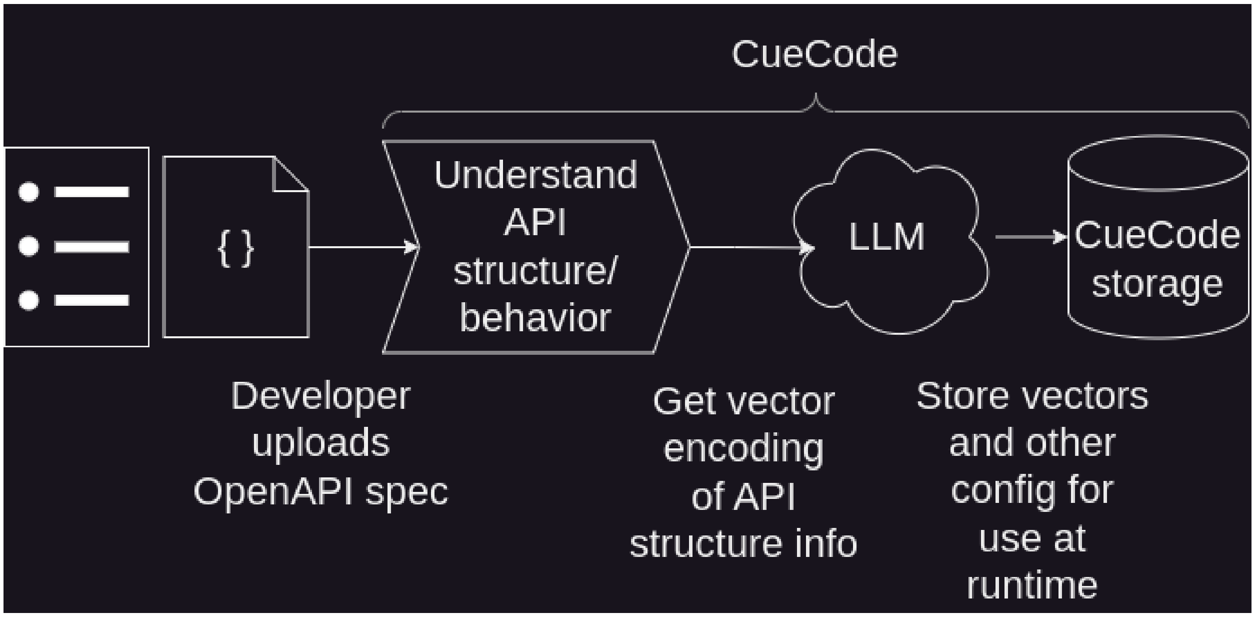
Conceptual diagram of turning natural language into Web API payloads, with CueCode shown as the NLP component.


Use case 1:
Customer's end user
Enters/speaks directly to customer application's UI
Collect Natural Language input
???
{ }
Structured API payload
Humans and/or rules whether to perform the API call or not
Perform API call
{ }
Structured API payload
Web API
Use case 2:
Customer's textual language data
Text is ingested by customer application

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


Natural language input
Your application
Specialized systems for generating API payloads
Use case 1:
Customer's end user
Enters/speaks directly to customer application's UI
Use case 2:
Customer's textual language data
Text is ingested by customer application
NLP/LLM interface componenet
Send natural language to API payload suggestion service
Receive REST API payload suggestions
Use user feedback and/or business logic to determine if you should issue the suggested API calls
Perform call to the REST API as usual
NLP - Detect entities, actions, relationships
Given entities and text, perform cosine similarity search to find most relevant API endpoints
Retreive vectors associated with REST API endpoint for giving LLM context in generation
Prompt LLM to generate API payload from templates for specified REST API endpionts
Need live API data for correct API payload generation?
Yes
No
Validate necessity of the additional data requested by the LLM
Perform read-only call to the existing REST API and/or fulltext search to get additional data for generation
Feed additional data from the REST API to the LLM to aid generation
Finalize retreival of generated API payloads from the LLM
Validate that structure of the generated payloads match the OpenAPI spec
Order suggested API calls according to data dependencies
Encode suggestions for client
System componenet key
Interacts with Large Language Model (LLM)
Interacts with Vector Store
Interacts with the existing REST API

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.


CueCode
{ }
Understand API structure/ behavior
LLM
CueCode storage
Developer uploads OpenAPI spec
Get vector encoding of API structure info
Store vectors and other config for use at runtime