



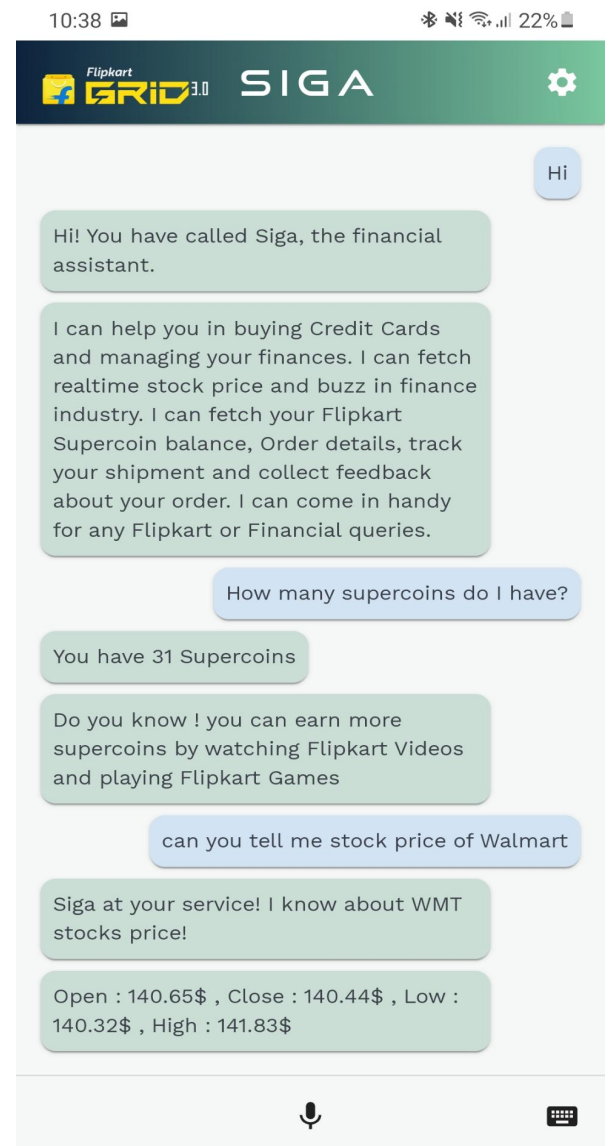
Siga, our voicebot, is live [here](https://nice-island-04efa0100.azurestaticapps.net/).

You may also copy the following URL and paste in the browser-

<https://nice-island-04efa0100.azurestaticapps.net/>

Things you can ask Siga

- Your Flipkart order status and SuperCoin balance.
- Flipkart related queries (warranty, returns, Bajaj Finserv, No Cost EMI etc.)
- Finance related queries (Credit and Debit Card, UPI, Loans, Insurance etc.)
- Recommendation of a new credit card for your needs
- Latest bulletins of Finance world
- Real-Time stock prices of different companies from NYSE
- Your pending credit card bills
- To record your feedback for our product



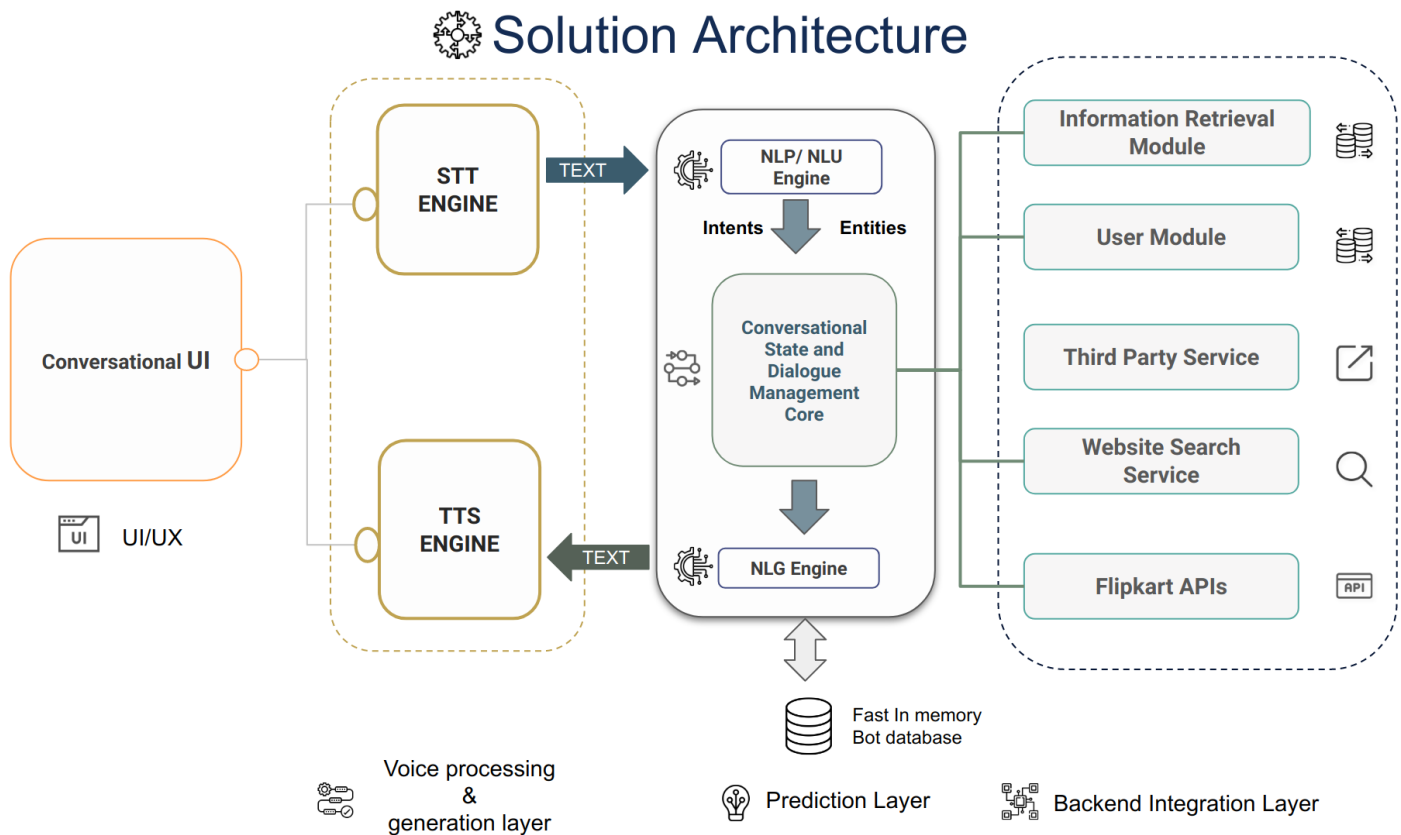


Important Keywords

1. **RASA** - ML based framework to design chatbot
2. **NLU** - Natural Language Understanding
3. **NLP** - Natural Language Processing
4. **NLG** - Natural Language Generation
5. **Entities** - Important information embedded in the user message
6. **STT** - Speech To Text
7. **TTS** - Text To Speech
8. **BERT** - Bidirectional Encoder Representations from Transformers



How does it work

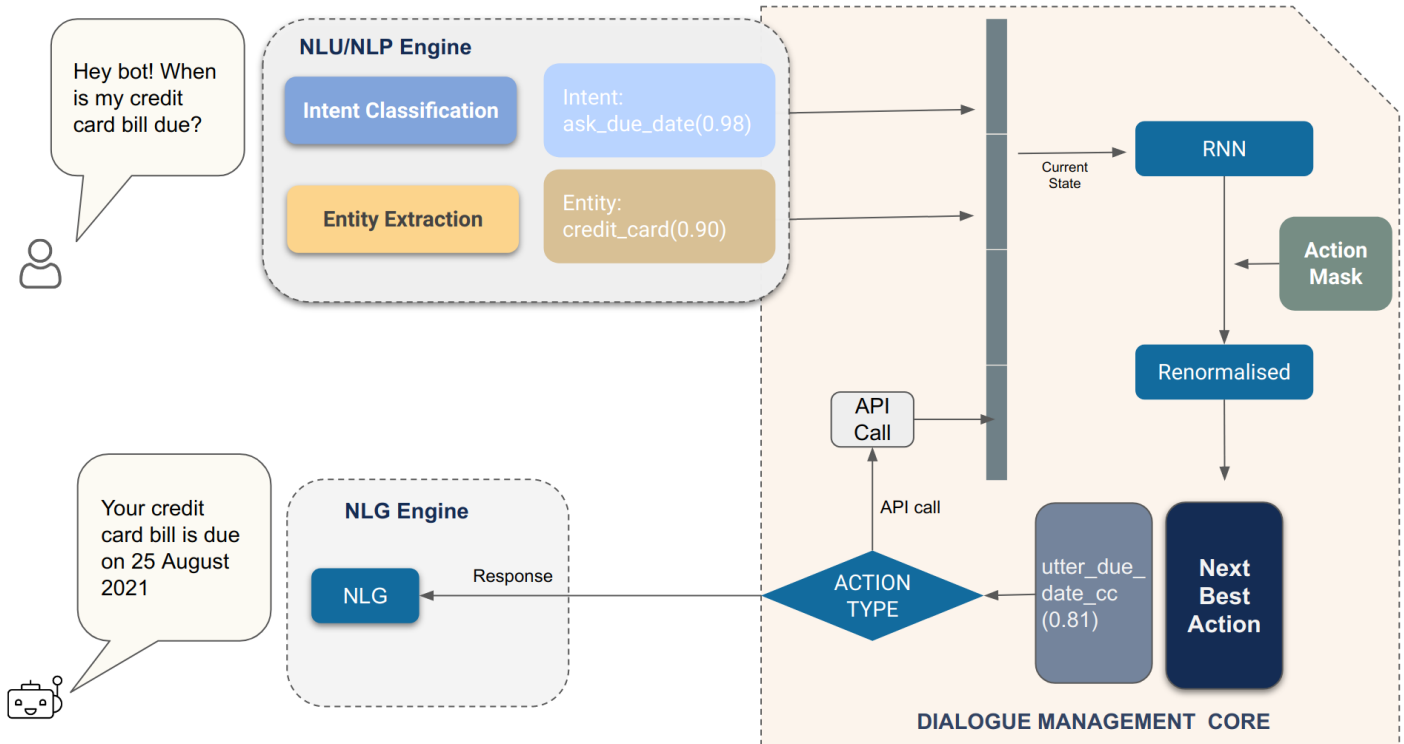


We have used **ReactJS** to develop the UI of the app. The UI records the voice of the user and sends it to the **Django** backend server. It uses **Azure Cognitive Services** to convert the audio to text and to translate the text into English, in the case of Hindi users. The text then goes to our core model.



Core Model of Bot

We have used [RASA](#), an open-source conversational AI tool to design the core of Siga. It is responsible for understanding the intent of the user and extracting entities from it. For eg. see what happens when a user asks about his credit card dues



Components

Rasa comes up with 2 components :

1. *Rasa NLU* — for natural language understanding (NLU) which does the classification of intent and extract the entity from the user input and helps the bot to understand what the user is saying.
2. *Rasa Core* — a chatbot framework with machine learning-based dialogue management that takes the structured input from the NLU and predicts the next best action using a probabilistic model.



Overview of the files

`data/nlu/` - contains NLU training data

`data/nlu/rules.yml` - contains rules training data

`data/stories/` - contains stories training data

`data/bert_encoding/` - contains sentence embeddings created using BERT

`actions/` - contains custom action/API code

`domain.yml` - the domain file, including bot response templates

`config.yml` - training configurations for the NLU pipeline and policy ensemble

Natural Language Understanding (NLU)

For intent classification and entity extraction we used the following pipeline:

❖ Language Models

- The following components load pre-trained models of word vectors in the pipeline.
- [SpacyNLP](#) : [en_core_web_md](#)

❖ Tokenizers

Tokenizers split the received text into tokens. We used the following in our pipeline.

- [SpacyTokenizer](#)

❖ Featurizers

Featurizers transform the words into meaningful numbers (or vectors) that can be fed to the training algorithm.



- [SpacyFeaturizer](#)
- [RegexFeaturizer](#)

❖ Intent Classifiers and Entity Extractors

Intent classifiers assign one of the intents defined in the domain file to incoming user messages.

- [Dual Intent Entity Transformer \(DIET\) Classifier](#)
- [NLU Fallback Classifier](#)
- [RegexEntityExtractor](#)
- [EntitySynonymMapper](#)

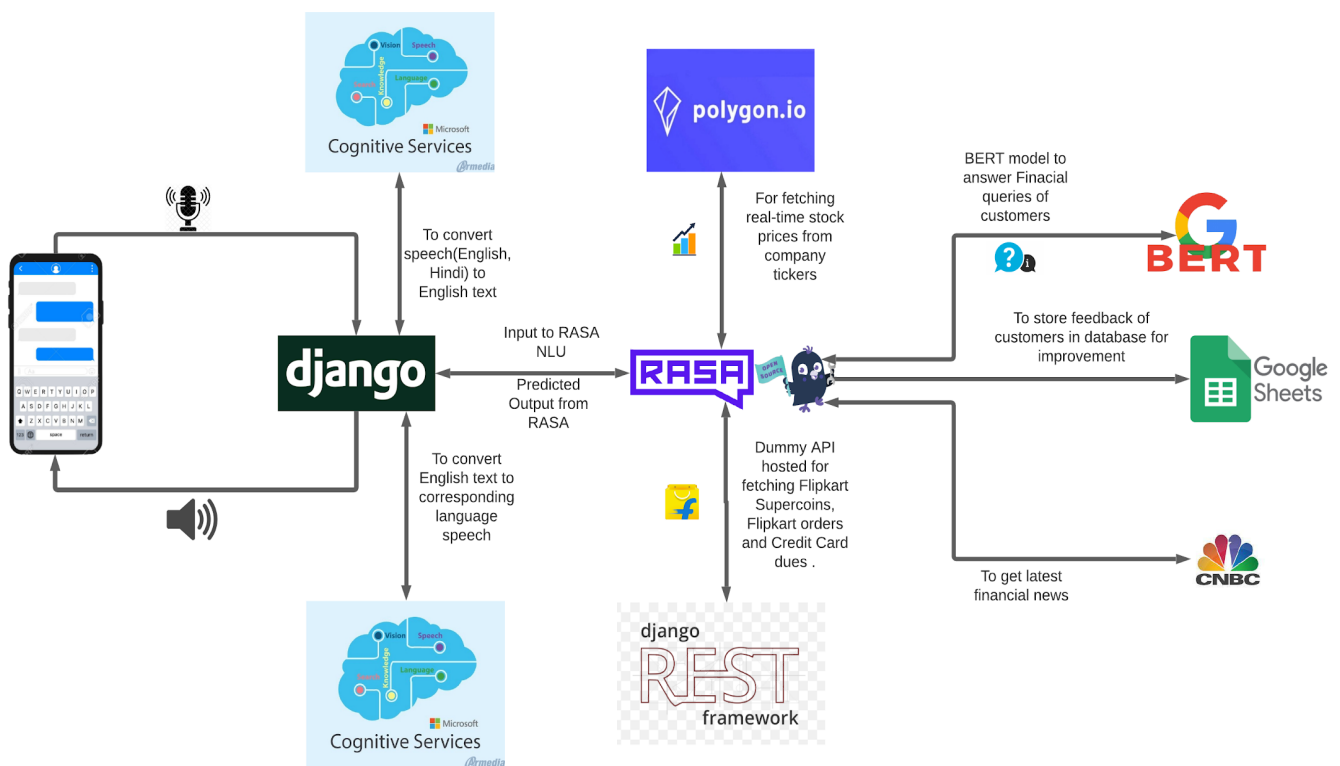
Core Policies

Policies to decide which action to take at each step in a conversation. We used the following configuration with fine tuning of parameters:

1. Machine Learning-Based Policies
 - [Transformer Embedding Dialogue \(TED\) Policy](#)
 - [Memoization Policy](#)
2. Rule-based Policies
 - [Rule Policy](#)



Following is the **low-level architecture** of various API's used by our core model.



Various API's used by RASA are:-

1. [Google Sheets API](#) to propagate the collected feedback from the users to Spreadsheet in real-time.
2. [CNBC News API](#) for fetching the latest news in the Finance domain.
3. [Ticker Search API](#) for getting stock tickers by company name.
4. [Polygon.io - Stock Market Data APIs](#) for retrieval of stock prices of top 100 companies listed on NYSE.
5. Dummy APIs (self-created) for Credit Card, and Flipkart Order.
6. Finance specific domain trained [BERT](#) for searching faqs.



Domain-Specific Training

We have scrapped common questions and answers according to our domain-specific needs from a wide range of websites that make the knowledge base of our bot. Siga will intelligently answer user questions based on this knowledge base using the following model

[Sentence Transformers: Multilingual Sentence, Paragraph, and Image Embeddings using BERT](#)

It provides state-of-the-art pre-trained models which are used to compute sentence embedding. Later, sentence embeddings are used to calculate the similarity between the user's question and the questions in the knowledge base.