**Voice Chat Use Case**

* **Speech-to-Text:** This component takes audio input from the user's microphone and converts it into text.
* **Large Language Model:** This is the brain of the system, responsible for understanding the user's text input, generating intelligent responses, and maintaining conversation context.
* **Text-to-Speech:** This component takes the LLM's text responses and converts them into natural-sounding speech to be played back to the user.

1. **Audio Input & Speech To Text:** Capture audio and convert it to text using your Speech to text engine.
2. **Embedding Generation:**

* Use the embedding model to create an embedding for the user's transcribed text.

1. **Contextual Chat Interaction:**

* Combine the user's text input with the generated embedding.
* Pass this combined input to the chat model.
* The chat model leverages the embedding to better understand the user's intent and context. It can also compare the user's embedding with embeddings of retrieved information to find relevant responses.

1. **Response Generation & Text to Speech:**

* The chat model generates a text response.
* Convert this response into speech using the Text to Speech engine and play it back to the user.

**Example: Understanding User Intent**

Imagine the user says, "Tell me about the weather in New York City."

1. **Speech To Text:** The speech is transcribed into the text: "Tell me about the weather in New York City."
2. **Embedding:** An embedding is generated for this text, capturing the semantic meaning of the phrase (e.g., it's a question about the weather in a specific location).
3. **Intent Recognition:** The embedding is compared to pre-defined intent embeddings, and it's determined that the user's intent is to ask a question about the weather.
4. **LLM Response:** The LLM uses both the text and the intent information to generate a response like: "The weather in New York City is currently sunny and 75 degrees Fahrenheit."
5. **Text To Speech:** The response is converted to speech and played back to the user.