Building Voice Agents

Karim Lalani

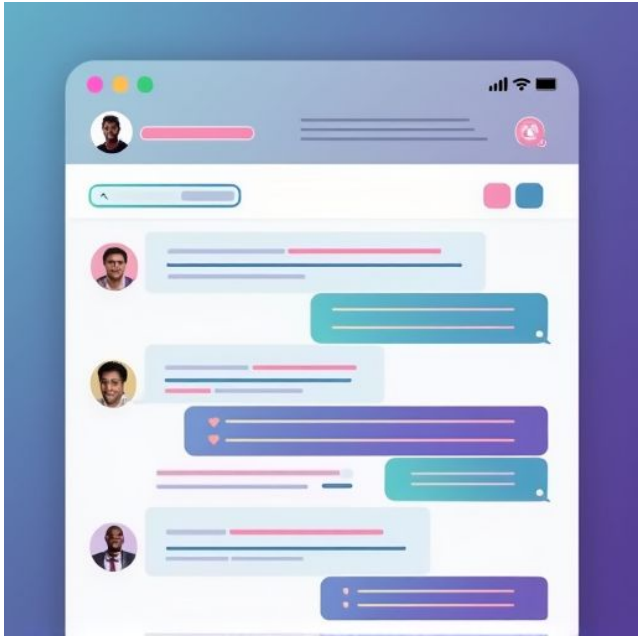
About Me - Karim Lalani

- **Home**: Leander, TX
- **Work**: Software Solutions Architect
- **Background**: Full Stack Engineer, Gen AI
- **FOSS**: LangChain contributor, FastRTC client,
  Tool Calling wrappers for LLMs
- **Using LangChain**:
  Delivered multiple solutions built with Langchain
- **Socials**:
  - Linkedin https://www.linkedin.com/in/-karim-lalani/
  - Github https://github.com/lalanikarim/
  - Medium https://medium.com/@klcoder

Text-Based Chatbots vs. Voice Agents

Text-Based Chatbots

- **Interaction**: Keyboard input, structured dialogue.
- **Accessibility**: Limited for visually impaired users.
- **Speed**: Fast for simple, direct queries.
- **Use Cases**: FAQs, quick answers, transactional tasks.



Text-Based Chatbots vs. Voice Agents

Voice Agents

- **Interaction**: Natural speech, hands-free, real-time.
- **Accessibility**: Enhanced for diverse users (e.g., visually impaired).
- **Speed**: Dynamic, immersive for complex or multitasking scenarios.
- **Use Cases**: Multitasking, accessibility, conversational tasks.

OpenAI Voice Mode

- **Natural Voice Conversations:** Real-time, low-latency interactions for intuitive, human-like dialogue.
- **Accessibility & Convenience:** Enables multitasking and supports users with visual impairments or voice-input preferences.
- **Tech-Driven Integration:** Leverages APIs (e.g., OpenAI Realtime Audio, Deepgram) for speech-to-text, TTS, and sentiment analysis.

Flow of Voice Agents

1. **User Speaks**
    - The user initiates interaction by speaking into a microphone.
2. **Streaming Audio Data**
    - Audio is captured in real-time and sent to the voice agent system.
3. **Voice Activity Detection (VAD)**
    - Detects when the user is speaking (vs. silence or background noise).
4. **Speech-to-Text (STT)**
    - Converts the audio stream into text for processing (e.g., using OpenAI's Realtime API or third-party services).
5. **Processing**
    - The text is analyzed using NLP or AI models (e.g., ChatGPT) to generate a response.
6. **Text-to-Speech (TTS)**
    - The generated response is converted back into audio.
7. **Generated Speech Audio Streamed Back**
    - The audio response is sent to the user in real-time, completing the interaction.

Flow of Voice Agents

Key Notes:

- Real-time streaming ensures low latency for natural conversation.
- VAD and STT are critical for accurate speech recognition and filtering.
- TTS quality impacts the user experience (e.g., naturalness, clarity).

Introducing FastRTC

FastRTC is a platform or framework focused on **real-time communication (RTC)**, built on **WebRTC** (Web Real-Time Communication) and **WebSockets** technology. It enables developers to create **low-latency, audio/video streaming and data transfer** for applications like voice agents, video conferencing, live chats, and collaborative tools.

FastRTC simplifies the complexities of WebRTC, making it easier for developers to build real-time, interactive applications with minimal infrastructure overhead. It’s ideal for projects requiring high-performance, low-latency communication.

FastRTC

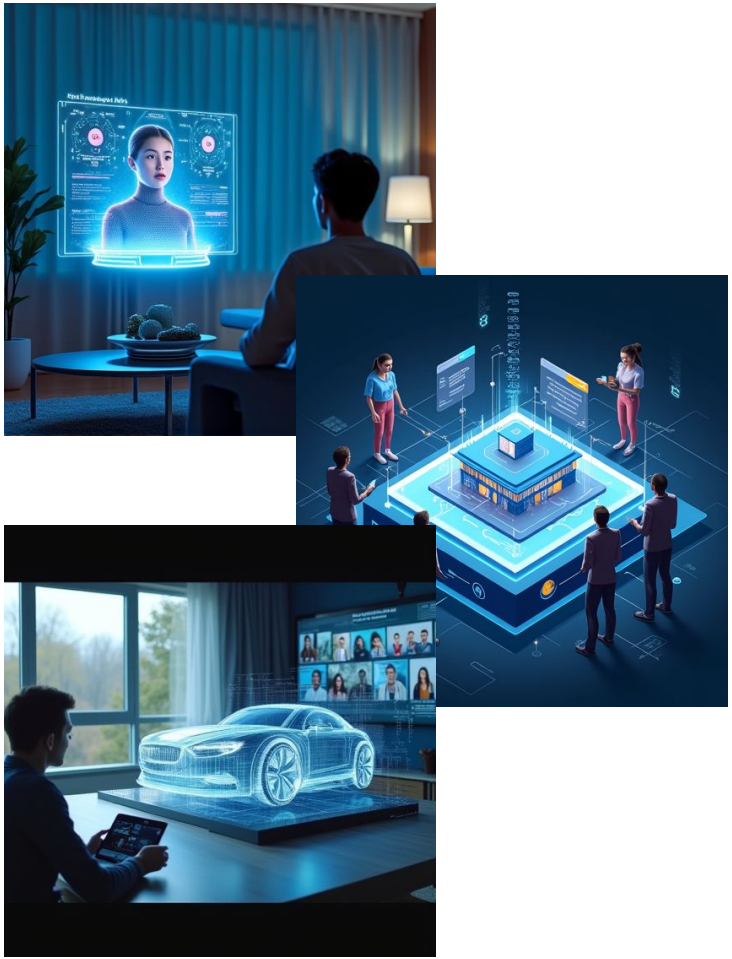
Key Features:

- **Real-time audio/video streaming**: Facilitates seamless, low-latency communication between users.
- **Scalability**: Supports large-scale applications with efficient resource management.
- **Cross-platform compatibility**: Works across browsers, mobile, and desktop environments.

FastRTC

Use Cases:

- Voice agents (e.g., integrating voice chat into AI assistants).
- Live video streaming, virtual meetings, and interactive web apps.
- Real-time data sharing (e.g., collaborative editing, gaming).



Demo

Building an Echo Server with FastRTC

Happy Vibing!