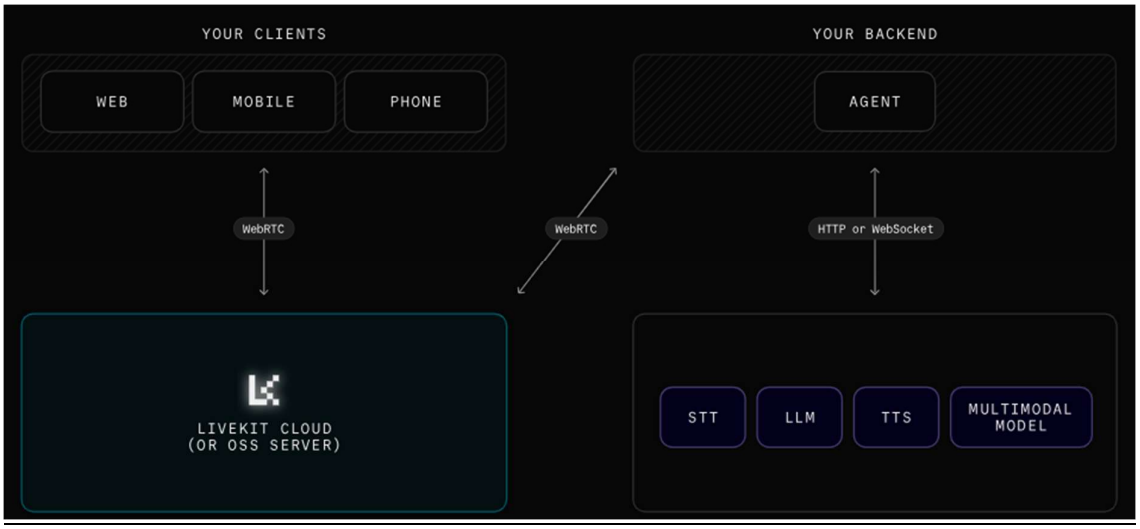
SitaraAI using LiveKit



LiveKit:

LiveKit is a comprehensive, open-source platform for building real-time audio and video applications, leveraging a modern, end-to-end WebRTC stack. It provides developers with tools for creating high-quality, low-latency, and scalable real-time communication experiences, including features like live streaming, video calls, and in-game communication. LiveKit offers both server-side and client-side SDKs, making it easy to integrate real-time media into various applications.

Core Pipeline Components

- STT (Speech-to-Text): Google Speech Recognition with voice activity detection
- LLM: OpenAI GPT integration with streaming responses
- TTS (Text-to-Speech): pyttsx3 with optimized settings for low latency

Comprehensive Metrics Logging

- EOU Delay: End-of-utterance detection latency
- TTFT: Time to first token from LLM
- TTFB: Time to first byte from LLM
- Total Latency: End-to-end pipeline timing
- Usage Summary: Token consumption, interruptions, success rates

Bottlenecks faced during development:

1. Libraries not being compatible with my laptop.
2. Livekit classes or packages being changed or deprecated causing unresolvable problems
3. Error handling
4. Undesired performance of the model
5. Late responses
6. Network delays
7. Insufficient resources

Mechanism:

- Audio Capture & Streaming: The system captures audio from the user's microphone and streams it in real-time over WebRTC connections. LiveKit handles the low-latency audio transport and synchronization.
- Speech-to-Text (STT): The incoming audio stream is processed by a speech recognition service that converts spoken words into text. This typically uses models like Whisper or cloud-based STT APIs that can handle streaming audio with minimal latency.
- Natural Language Processing: The transcribed text is sent to a language model (like GPT, Claude, or other LLMs) that understands the user's intent and generates an appropriate text response.
- Text-to-Speech (TTS): The AI's text response is converted back to natural-sounding speech using TTS engines like ElevenLabs, OpenAI's TTS, or other voice synthesis services.
- Real-time Audio Playback: The generated audio is streamed back to the user through LiveKit's audio infrastructure, maintaining low latency for natural conversation flow.
- Session Management: LiveKit manages the persistent connection, handles interruptions, manages turn-taking, and maintains conversation state throughout the interaction.

Changes to be made:

- Ensure you have all the required libraries installed mentioned in requirements.txt.
- Create a separate virtual environment.
- Put in your secret keys in .env file
- Run it from the terminal
- Use command [python api.py console] to run the file.
- Use ctrl+c to end the conversation and see the metrics.
- The metrics are saved in the excel file in the same directory as your voice assistant.

Conclusion:

LiveKit's voice assistant architecture represents a sophisticated orchestration of modern AI technologies within a real-time communication framework. By seamlessly integrating speech recognition, natural language processing, and speech synthesis through a low-latency audio pipeline, it creates truly conversational AI experiences.

The technology essentially bridges the gap between powerful AI language models and natural human conversation, making AI assistance as intuitive as talking to another person.

Thank You!!