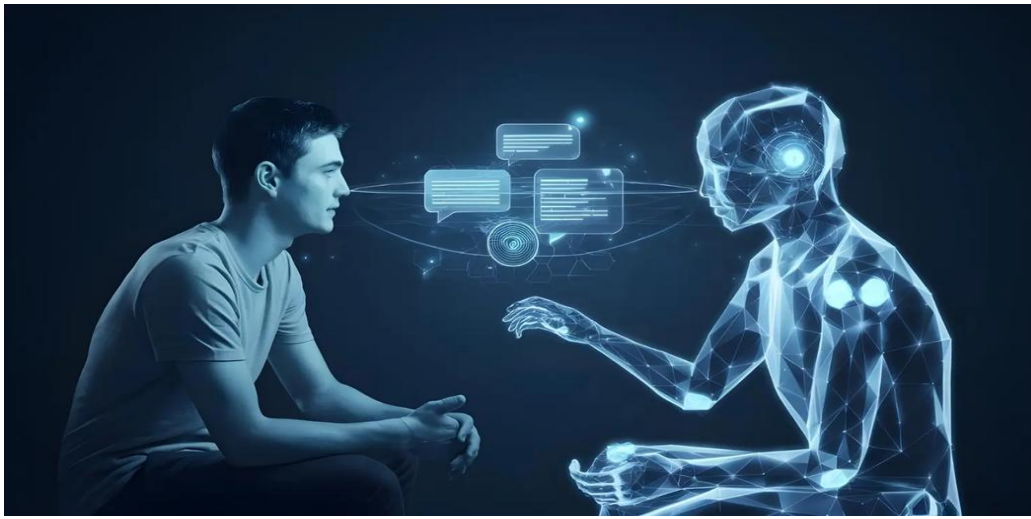


AI Conversational Voice Agent With Avatar

A short report

Author : Ankur Kumar

What is an AI Voice Agent?



Definition:

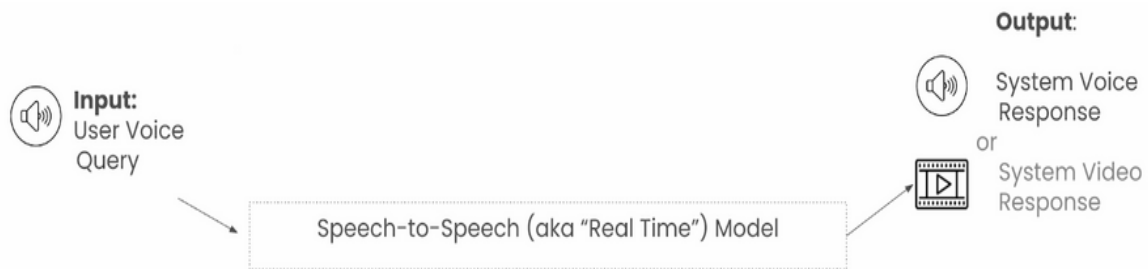
AI Voice Agents are systems that combine advanced speech recognition and reasoning capabilities from large foundation models to deliver real-time, human-like voice interactions.

Use cases:

- **Education and Training:** Provide personalized coaching, skill development guidance, and conduct practice interviews to enhance learning outcomes.
- **Customer Service:** Automate voice calls for tasks like restaurant reservations, sales inquiries, appointment scheduling, and insurance support, improving efficiency and availability.
- **Healthcare and Accessibility:** Enable voice-based medical consultations, therapy sessions, and support for individuals with disabilities to improve access to care and reduce barriers to communication.

There is two way to approach this

Approach 1 : Using Speech to Speech model



This system takes a user's spoken question as input and uses a real-time speech-to-speech model to process it end to end. The model understands the spoken query and immediately generates a spoken (or video) response without requiring intermediate text steps.

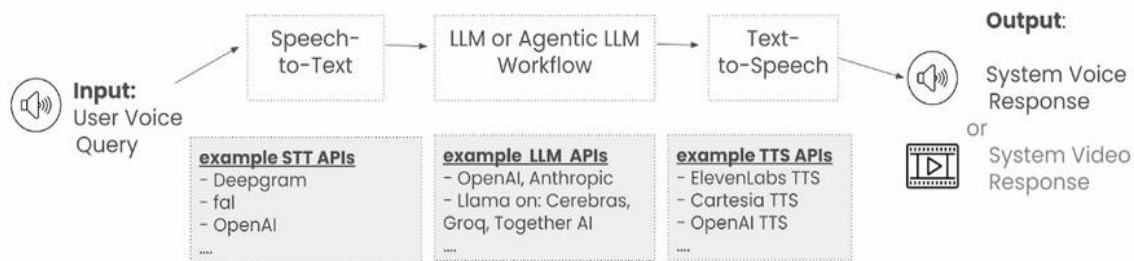
Advantages:

- Feels natural and human-like.
- Enables fast, real-time replies, low latency.
- Improves accessibility.

Disadvantages:

- Less flexibility in handling complex or varied queries.
- Limited context window—struggles to manage long or detailed conversations.
- Not able to retain memory across interactions, making personalization harder.
- Technically demanding to build and maintain.
- Cannot implement agentic workflows (e.g., planning, multi-step reasoning, tool use).

Approach 2 : Using STT, LLM, AI Agent , TTS Workflows



This system takes a user's spoken question and processes it in three modular steps: first, Speech-to-Text (STT) converts the spoken input into text; next, the LLM or Agentic Workflow interprets the text, reasons about it, and generates an appropriate text response; finally, Text-to-Speech (TTS) converts this text answer back into spoken output, enabling a seamless and interactive voice-based conversation.

Advantages:

- **Supports Agentic Workflows:** Can perform multi-step reasoning, planning, and external tool/API calls.
- **Highly Flexible:** Each component (STT, LLM, TTS) can be independently chosen or replaced to optimize cost, quality, or language support.
- **Better Context Handling:** Can maintain memory across turns for more coherent, personalized conversations.
- **Improved Observability:** Easier to monitor, debug, and optimize individual stages.
- **Scalable Personalization:** Supports user profiles, long-term memory, and advanced dialogue management.

Disadvantages:

- **Higher Latency:** Multiple processing steps increase total response time compared to direct speech-to-speech models.
- **Integration Complexity:** Needs careful orchestration of separate APIs and services.




- **Higher Costs:** Each step may involve paid APIs, increasing total operational expense.
- **Privacy Concerns:** User data passes through multiple services, requiring strong safeguards.
- **Dependency on Network Stability:** Any failure or delay in a single module can break the entire conversation flow.





Developers often prefer the modular STT → LLM → TTS approach.





Developers generally prefer the modular Speech-to-Text → LLM → Text-to-Speech approach because it offers greater flexibility and control. By breaking the workflow into clear stages, they can independently choose, upgrade, or fine-tune each component to suit specific needs. This design also supports more advanced agentic workflows, enabling complex reasoning, planning, and external tool use that aren't possible in simpler real-time speech-to-speech models. Additionally, it allows for better context management and memory across conversations, making interactions more personalized and coherent. While it can introduce slightly higher latency and integration complexity, developers value the transparency, debuggability, and customization this approach provides.

How to improve latency read this article : [How to Optimize Latency for Conversational AI](#)

Overview of Infrastructure & Components Needed to Build a Conversational AI Avatar

Component	Purpose (Detailed Description)	Example Tools / APIs	Advantages / Notes
 Voice Activity Detection (VAD)	<p>Detects when the user begins and ends speaking, allowing the system to activate processing only when speech is present. It filters out silence or background noise, enabling efficient and accurate capturing of user input.</p>	<ul style="list-style-type: none"> - WebRTC VAD - Silero VAD - AssemblyAI VAD - Deepgram built-in VAD 	<p>Reduces false triggers, saves compute, improves UX</p>
 End of Turn Utterance (EOU)	<p>Determines the natural pause or semantic completion in the user's speech, signaling when they are done speaking. This enables smoother, human-like turn-taking by preventing interruptions and reducing latency between listening and responding.</p>	<ul style="list-style-type: none"> - Deepgram EOU - AssemblyAI EOU - Whisper (via pauses + punctuation) - LLM heuristics 	<p>Critical for real-time, natural turn-taking; enhances responsiveness and fluency</p>
 Speech-to-Text (STT)	<p>Converts the user's spoken words into written text in real-time. This is the first transformation step from voice to machine-readable input, essential for passing accurate transcriptions to the language model for understanding and response generation.</p>	<ul style="list-style-type: none"> - OpenAI Whisper - Deepgram - Google Speech-to-Text - AssemblyAI 	<p>Real-time transcription with low latency; Deepgram & AssemblyAI support VAD & EOU</p>

Component	Purpose (Detailed Description)	Example Tools / APIs	Advantages / Notes
 LLM / Conversational Reasoning	Takes the transcribed text and interprets the meaning, intent, and context. The LLM generates coherent, helpful, and engaging responses while optionally performing tool use, memory recall, or workflow execution to enable advanced assistant or agent behaviors.	- OpenAI GPT-4/GPT-4o - Anthropic Claude - Google Gemini - LLaMA (Groq, Together AI) - LangChain	Advanced reasoning, multi-turn memory, tool use, integration with agentic frameworks
 Text-to-Speech (TTS)	Converts the LLM-generated text back into expressive, natural-sounding speech. A high-quality TTS engine enhances the emotional tone and realism of the AI avatar, improving user engagement and accessibility.	- ElevenLabs ( priority) - OpenAI TTS - Amazon Polly - Microsoft Azure TTS - Google TTS - Coqui TTS	ElevenLabs offers high emotion, multilingual, real-time streaming; excellent for avatars
 Avatar Animation	Brings the AI assistant to life by synchronizing audio with lip movements and gestures on a 2D or 3D avatar. This visual embodiment increases user trust, engagement, and immersion in the interaction.	- D-ID - HeyGen - NVIDIA Audio2Face - Ready Player Me - Unreal Engine / Unity	Brings personality and engagement through 2D/3D visual rendering

Component	Purpose (Detailed Description)	Example Tools / APIs	Advantages / Notes
 Orchestration Layer	Coordinates how the components work together in a smooth pipeline. It handles input/output flow, state management, UI elements, API calls, and error handling, ensuring modularity and scalability in the system.	<ul style="list-style-type: none">- LangChain- Gradio- Streamlit- Node.js / React- Unity / Unreal	Manages logic flow, memory, API calls, and UI/UX
 Real-Time Communication / Streaming	Manages the real-time transmission of audio and/or video between the user and avatar. Ensures low-latency, bi-directional streaming with synchronization for natural conversation dynamics.	<ul style="list-style-type: none">- LiveKit ( priority)- WebRTC- Agora.io- Twilio Video	LiveKit enables scalable real-time audio/video; open source; perfect for avatars
 Deployment / Hosting	Provides the infrastructure to host the pipeline and scale with usage. It supports GPUs for inference, API deployment, and global delivery, making the system production-ready and accessible from anywhere.	<ul style="list-style-type: none">- AWS- Google Cloud- Azure- Render.com- Replicate- Hugging Face Inference Endpoints	Flexible compute, GPU support, global delivery