■ Voice-to-Voice Assistant Models: STT + TTS + Integration

This report provides a structured overview of voice-to-voice assistant models, combining Speech-to-Text (STT), Text-to-Speech (TTS), and integration workflows. It includes details about model types, online/offline capabilities, languages, hardware requirements, and architectural recommendations.

■ Speech-to-Text (STT) / ASR (Automatic Speech Recognition)

odel / API	Offline / Online	Languages	Notes / Strengths	GPU / CPU Suitab
hisper (OpenAI)	Offline & API	Many	Robust, supports translation & segmentation	GPU / CPU
eepSpeech	Offline	English, few others	Open source, light	CPU / light GPU
av2Vec 2.0 / XLS-R	Offline	Multilingual	Strong representation-based ASR	GPU preferred
aldi	Offline / Embedded	Many	Highly customizable, research standard	CPU / GPU
beechBrain	Offline	Multilingual	Flexible, supports training/inference	GPU
ssemblyAI API	Online	English + others	Streaming, enterprise-grade	Cloud GPU
oogle Cloud STT	Online	Many	Reliable, streaming support	Cloud
icrosoft Azure Speech	Online	70+	Enterprise customization	Cloud
mazon Transcribe	Online	Many	AWS ecosystem	Cloud
M Watson STT	Online	Multilingual	Enterprise-grade	Cloud

■ Text-to-Speech (TTS) Models / Engines

Model / Engine	Offline / Online	Languages / Voices	Notes / Strengths	GPU / CPU Suitable
Coqui TTS	Offline / Online	Many	Expressive, cloning, multilingual	GPU preferred
Tacotron 2 + vocoder	Offline	English, others	Classic neural TTS	GPU
FastSpeech 2	Offline	Many	Fast non-autoregressive	CPU / GPU
Glow-TTS	Offline	Multilingual	Good prosody control	GPU recommended
VITS	Offline	Multi-speaker	End-to-end neural TTS	GPU ideal
YourTTS	Offline / Online	Multilingual	Expressive & cloning	GPU
Bark (Suno AI)	Offline / Online	Multilingual	Expressive & realistic	GPU
XTTS v2	Offline / Online	Many	Powerful style & cloning	GPU
NaturalSpeech 2	Offline / Research	Many	Diffusion-based, zero-shot	GPU
Festival / eSpeak	Offline	Many	Lightweight, classic	CPU

■ Integration Workflow (STT → LLM → TTS + Subtitles)

1. Capture audio via microphone (e.g., PyAudio or sounddevice).
2. Perform Speech-to-Text (STT) transcription (e.g., Whisper, Wav2Vec).

3. Display live subtitles while recording.
4. Process recognized text using an LLM (GPT, Claude, etc.) for response.
5. Convert response text to speech (TTS) using Coqui, XTTS, etc.
6. Play synthesized speech and show synced subtitles.

Considerations: Minimize latency, support streaming, handle interruptions, and optimize for hardware.

■ GPU vs CPU / Compatibility Notes

- Large TTS/STT models perform best on GPU.
- Smaller models (e.g., Whisper small, Coqui small) can work on CPU.
- Use server offloading for heavy models in production.
- Windows builds for Whisper and Coqui are widely available.
- Hybrid setups (online + offline) improve resilience and flexibility.