ElevenLabs is widely known for text-to-speech (TTS): turning written text into spoken audio. ElevenAgents takes that voice capability into a broader job. It is designed for building conversational agents that can listen, respond, and carry out interactions across channels such as phone lines and a website widget.

The distinction matters if you are choosing technology for customer support. A convincing synthetic voice is only one part of a useful voice agent stack. The system also needs to recognize what a caller said, decide what to do next, respond at an appropriate speed, handle interruptions, retrieve accurate information, and know when to hand a conversation to a person.

In short, ElevenLabs TTS helps an application speak. ElevenAgents is aimed at managing more of the conversation around that speech. This post explains the difference, outlines where voice-agent deployments commonly fail, and offers practical ways to evaluate the https://aiagentstore.ai/guides-and-tutorials/how-to-evaluate-a-conversational-ai-platform-for-voice-agents experience before connecting it to customers.

What is ElevenLabs TTS?

Text-to-speech software converts text into audio. An application supplies a sentence or script, selects a voice and configuration, and receives spoken output. Teams use TTS for narrated content, voice interfaces, accessibility features, and the spoken responses inside automated services.

ElevenLabs provides text-to-speech capabilities, including a range of voice options. In a support workflow, TTS might read a response that another part of the system has already written. For example, a help application could retrieve an answer about a return policy, then use TTS to speak it over a phone line.

On its own, TTS does not determine what the customer means, look up an order, or decide whether the issue needs a human. Those responsibilities belong to the surrounding application or to other services in the voice agent stack.

What is ElevenAgents?

ElevenAgents is a conversational-agent offering from ElevenLabs. Rather than focusing only on generating spoken audio, it is intended to support interactive agents that can take in a user's input and produce responses in a conversation. Depending on the deployment and configuration, that can involve speech recognition, an AI model, instructions and tools, knowledge sources, and spoken output.

Think of the difference this way: TTS is a component; an agent is a system that uses components to manage a task. An agent may need to understand a request, decide whether it has enough information, consult a knowledge base or connected system, formulate a response, and deliver it through voice or another supported interface.

That does not mean every conversation is automatically reliable just because it runs on an agent platform. The quality still depends on the configuration, integrations, information available to the agent, and behavior under real-world conditions. A polished demo can show what is possible; it cannot establish how the system will perform with noisy callers, ambiguous requests, or slow back-end services.

ElevenAgents vs. ElevenLabs TTS at a glance

Area ElevenLabs TTS ElevenAgents Primary job Convert supplied text into spoken audio. Support an interactive agent that listens and responds. Conversation decisions Typically made by the application that supplies the text. Part of the agent experience, with behavior shaped by its instructions, model, tools, and configuration. Input Text to speak. Conversation input, such as speech or text, depending on the channel and setup. Typical role in a support stack Spoken output for a response generated elsewhere. A conversational layer that can be connected to channels and relevant business systems. What to evaluate Voice quality, pronunciation, and audio delivery. End-to-end accuracy, latency, turn-taking, interruptions, integrations, escalation, and answer quality.

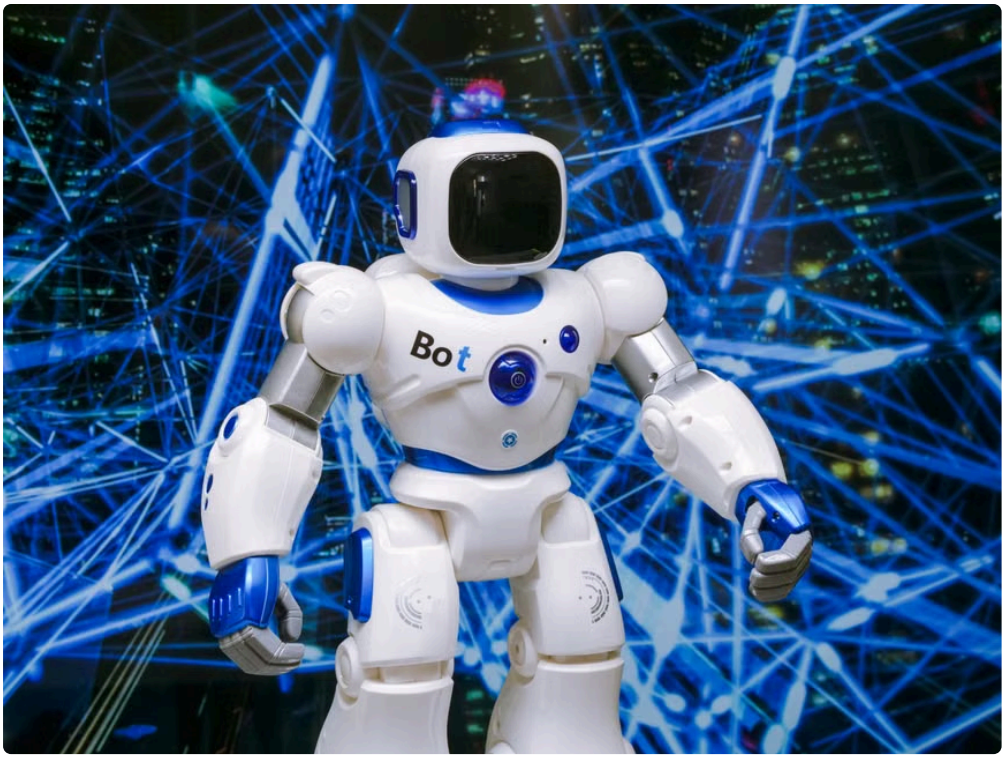
The voice agent stack: where things can break

A customer-facing voice agent is a pipeline, not a single voice. A simplified interaction might pass through these stages:

1. **Audio capture:** A caller speaks through a phone line, or a visitor speaks through a website widget.
2. **Speech recognition:** The system converts audio into text or another representation it can work with.
3. **Conversation logic:** An agent interprets the request and decides what information or action is needed.
4. **Retrieval or tool use:** The system may search approved information, check a connected service, or request missing details.
5. **Response generation:** The agent composes a reply or selects an appropriate action.
6. **Speech generation and delivery:** TTS turns the reply into audio, which is sent back to the customer.
7. **Handoff and recordkeeping:** If needed, the conversation is transferred or summarized for a support team.

Every stage introduces potential failure points. Recognition can miss a product name. An instruction can be unclear. A knowledge source can be stale. An integration can time out. A response can be correct but too slow, or an escalation can reach a person without the context they need.

This is why comparing ElevenAgents with TTS should not stop at “Does the voice sound natural?” For a support use case, test the entire voice agent stack. If the agent uses Zendesk, for example, check what happens when it needs a human: Does it route the interaction correctly? Does the agent pass a useful summary, customer details where appropriate, and the steps already attempted? The precise integration capabilities depend on the configured product and workflow, so verify them in your own environment.



Latency and turn-taking: realism is about timing, too

People judge a phone conversation by its rhythm as much as its sound. A response that arrives after a long pause can feel broken, even if the words are accurate. A response that begins while the caller is still speaking can feel rude or confusing.

End-to-end latency is the accumulated delay across the pipeline: receiving audio, recognizing speech, reasoning, retrieving information, generating a reply, and delivering audio. TTS may contribute to that delay, but it is only one piece. A slow lookup or a model that waits for the caller to finish a long utterance can matter just as much.

Turn-taking is also more than choosing a short response time. A good interaction distinguishes between a natural pause and the end of a thought. If a caller says, “I need to change the address on…” and pauses to find an order number, an agent that jumps in too early may force the caller to repeat themselves. If it waits too long after a complete question, the silence can sound like a dropped call.

When testing ElevenAgents or any alternative, measure the time from the end of a caller’s utterance to the beginning of a useful response. Note the longest pauses, not just the average. Listen to the recordings. In evaluations, I count dead-air seconds out loud because a dashboard’s average can conceal the moment when a customer decides the system is not responding.

Interruption handling and timeouts

Real callers interrupt. They correct themselves, add a detail, change their mind, or say “wait” when the agent starts down the wrong path. An agent that cannot stop speaking, register the correction, and adjust its response may sound scripted rather than conversational.

Test interruptions deliberately. Ask a question, begin a correction while the agent is speaking, and see whether it yields. Try speaking over a confirmation, changing an account detail mid-sentence, or saying “No, that’s not what I meant.” Then check whether the next response reflects the correction, rather than continuing with the previous plan.

Timeout behavior deserves the same attention. Systems may need to decide how long to wait for a caller to answer, what to do when audio is unclear, and when to retry or offer another route. Too-short timeouts can cut off thoughtful or slower-speaking customers. Too-long timeouts create dead air and may leave callers unsure whether the line is still active.

Do not rely on broad claims such as “natural conversation” without testing controls and outcomes. Find out which interruption and timeout behaviors can be configured for your setup, how they are exposed, and how they appear in call logs or recordings. The settings that are available can vary by channel and configuration; confirm them directly instead of assuming that a demo reflects your production controls.

Knowledge bases and RAG quality

When an agent answers questions using company material, the source and retrieval process matter. Retrieval-augmented generation, often shortened to RAG, is a common pattern: the system searches relevant content and uses that information to help generate an answer. Retrieval can make a response more grounded, but it does not guarantee that the right passage will be found or interpreted correctly.

A weak knowledge base can produce weak answers even when the voice is excellent. Common causes include outdated policy pages, duplicate articles with conflicting instructions, missing product names, and documents written for employees rather than customers. Search can also return a passage that is related to a question but does not actually answer it.

Prepare a test set using real customer questions, including awkward versions: incomplete sentences, informal wording, product nicknames, and questions that combine two issues. Include cases where the approved answer is “I don’t know” or where the agent should clarify before responding. Then review whether the retrieved material supports the answer, not simply whether the answer sounds plausible.

For a support team using Zendesk, align the agent’s knowledge sources and escalation path with the way the team actually works. A policy answer should not contradict the current help content, and a transfer should not leave an agent to ask the customer to start over. Keep source material current, define which information is authoritative, and test what happens when the answer is absent or conflicting.

Choosing between TTS and an agent platform

Choose TTS when your application already handles the conversation logic and you mainly need spoken output. For example, a fixed announcement or a response generated by an existing support workflow may need a clear, well-produced voice but no independent conversational decision-making.

Consider ElevenAgents when you want to build an interactive agent and need more than audio generation: a system that can take turns, respond to caller input, and work within a broader conversational setup. The right choice depends on the exact channels, integrations, controls, and operational requirements you need. Treat product capabilities as something to validate in the current offering and your own configuration.

Some teams will use TTS as one part of a larger voice agent stack. Others may evaluate an agent platform that brings more of the conversational workflow together. In either case, map the responsibilities clearly: Which service recognizes speech? Which one retrieves information? Where is business logic enforced? Who controls handoff? Where are recordings and transcripts reviewed?

A practical evaluation checklist

- Test from actual phone lines, not only from a clean browser or studio setup.
- Call once from a quiet room and again with realistic background noise.
- Use an awkward-caller script with pauses, corrections, interruptions, and unclear details.
- Record response latency and identify the slowest moments in each pipeline stage.
- Check whether the agent waits appropriately, stops speaking when interrupted, and recovers from misunderstandings.
- Test silence, missing information, recognition errors, and back-end delays.
- Ask questions that the knowledge base can answer, cannot answer, and answers only after clarification.
- Verify escalation: the human should receive useful context, not just a transfer with no explanation.
- Review the website widget separately from phone calls; the channel can change the interaction and available controls.
- Confirm that your team can inspect the relevant logs, tune the available settings, and repeat tests after changes.

The bottom line

ElevenLabs TTS is about generating spoken audio from text. ElevenAgents is aimed at interactive conversations, where speech is only one part of the system. That is the central difference—and it is why the evaluation should cover the full voice agent stack, not just the sound of the voice.



For customer support, focus on what happens when the call is noisy, the customer interrupts, the answer is missing, or an integration is slow. Test the knowledge base, measure the pauses, and confirm that a handoff preserves context. A voice can make an interaction pleasant, but reliable turn-taking, accurate information, and a graceful recovery are what make the experience useful.