A voice agent can sound polished in a product demo and still fail the first time a caller says, “Wait, that’s not what I meant.” The difference often comes down to settings that are easy to overlook: when the agent starts speaking, whether it notices an interruption, how long it waits for a reply, and what it does when the conversation stalls.

When evaluating a vendor, don’t settle for “we support barge-in” or “timeouts are configurable.” Ask what those terms mean in practice, which parts you can control, and how the behavior changes across your phone lines and website widget. If a vendor can’t demonstrate the settings or explain how they’re tested, treat that as a risk—not a minor configuration detail.

This guide lays out practical vendor questions about timeout settings, interruptions control, and turn-taking configuration. It also covers the voice-agent pipeline, latency, and knowledge quality, because interruption problems are often symptoms of a failure somewhere else in the system.

Start with the full voice-agent pipeline

A voice agent is not one model listening and talking. A typical call passes through several stages: audio capture, speech recognition, turn detection, knowledge retrieval or tool use, response generation, and speech synthesis.

The answer then travels back to the caller. A delay or failure at any point can make the agent seem inattentive—or prompt it to talk over someone.

Ask the vendor to map that pipeline for your deployment. Find out which components are theirs, which come from third parties, and where audio or conversation state is passed between them. For example, if the speech recognizer has finished processing but the turn detector is still waiting for more audio, the agent may appear to pause. If retrieval takes too long, a silence may be mistaken for a timeout problem.

Pipeline stage What can go wrong Question to ask Audio capture and transcription Noise, accents, clipped words, or transcription delay How do you detect low-confidence or incomplete speech? Turn detection The agent decides the caller has finished too early—or waits too long Which end-of-turn settings can we adjust? Retrieval and tools Slow lookup, irrelevant content, or a failed integration What happens while a knowledge or system lookup is pending? Response generation and speech A long answer starts late or keeps playing after an interruption Can speech be stopped immediately and resumed or rewritten?

Ask about latency and realistic turn-taking

Think about it: latency is not just a number on a slide. It is what the caller experiences between finishing a thought and hearing a useful response. A system may report a fast model response while adding delay in transcription, retrieval, integration calls, or speech synthesis. Ask for end-to-end timing from real interactions, not only the time taken by one model component.

Useful vendor questions include:

- How do you measure response latency—from the caller’s last spoken word to the first audible response?
- Can you show median and high-percentile latency, such as the slowest five percent of turns?
- Can you break the timing down by transcription, turn detection, retrieval, tool calls, generation, and speech output?

- What does the agent say or do if a lookup is taking longer than expected?
- Can we configure short acknowledgments while the system works, and prevent them from sounding repetitive?

Turn-taking realism depends on more than speed. A caller may pause to remember an account number, take a breath, or speak in short fragments. Ask whether the vendor lets you tune end-of-turn detection for those patterns. Pretty simple.. A short pause should not automatically trigger a response, but an agent also shouldn't leave callers wondering whether the line has gone dead.

I evaluate this with the same demo line twice: once from a quiet room, then again with background noise. I also use deliberately awkward caller scripts—interrupting mid-sentence, changing my answer, and pausing at unusual points. Studio-quality demos rarely reveal how an agent handles those moments.

Get specific about interruption handling

"Barge-in supported" is not a complete answer. It might mean the agent detects a new voice, stops generating audio, stops playback, or simply waits until its current sentence ends. Those behaviors feel very different to a caller. Ask the vendor to show the entire interruption path, including what happens to the words the caller said while the agent was speaking.

Questions about interruption controls

- What event counts as an interruption: any detected speech, a minimum length of speech, or a confidence threshold?
- Can the agent stop speech immediately, or does it finish the current phrase?
- Can we control sensitivity to background voices, TV audio, and line noise?
- When interrupted, does the agent discard its response, pause it, or regenerate it after listening?
- Does it retain the caller's complete interruption, including speech that begins before playback stops?
- Can it distinguish "yeah, go on" from "no, cancel that" or a new question?
- Can we configure different behavior for confirmations, safety messages, and ordinary answers?
- Are interruption events and resulting actions visible in logs or call recordings?

Good interruptions control should respect intent, not merely detect sound. If a caller says, "I have the order number—actually, never mind," the agent should not keep delivering a prepared answer as if nothing happened. Conversely, a cough or a nearby conversation should not derail every turn. Ask how the vendor balances responsiveness against false interruptions and how you can test that balance.



Make the vendor demonstrate interruptions during a long response, a short confirmation, and a response that depends on a live lookup. Then try speaking over it from a noisy environment. Listen for how quickly playback stops, whether the agent acknowledges the caller naturally, and whether it responds to what was actually said.

Unpack timeout settings instead of accepting one number

“Timeout” can refer to several different timers. ...where was I?. There may be a silence timeout while waiting for the caller, an end-of-turn pause threshold, a limit on how long a tool call can run, a maximum response duration, and an overall call or session limit. Ask the vendor to name each timer, explain its default, and show where it can be configured.

Timeout type What it controls Potential failure Caller silence timeout How long the agent waits for speech It prompts too soon or leaves a silent caller stranded End-of-turn pause How much silence signals that a thought is complete It cuts off a pause—or waits through an awkward gap Tool or retrieval timeout How long an integration or knowledge lookup can take The agent hangs, gives a stale answer, or fails silently Session or call limit Maximum time for a conversation or interaction A legitimate support call is ended without a safe handoff

Ask what the agent says when each timeout is reached. Does it repeat the question, offer more time, explain that a lookup is taking longer, or transfer the caller? Can you configure the number of retries and the wording? Does the system preserve the conversation context if it transfers to a person?

Timeout behavior should fit the task. A caller entering a long address may need more silence than someone answering yes or no. A failed account lookup should not be treated like caller inactivity. Ask whether timeout settings can vary by flow or step, and whether changes apply to phone lines, the website widget, or both.

Test knowledge bases and RAG quality

Interruption and timeout settings cannot rescue an agent that has retrieved the wrong information. If a knowledge base or retrieval-augmented generation (RAG) system is involved, ask how content is indexed, selected, and kept current. A slow or irrelevant retrieval can cause long pauses, confident-sounding mistakes, or repeated clarifying questions that callers experience as poor turn-taking.

- Can the vendor show which source passages supported a specific answer?
- How does the agent respond when the knowledge base has no reliable answer?
- Can we set a confidence threshold that triggers clarification or escalation?
- How are duplicate, outdated, and conflicting documents handled?
- Can we measure retrieval time separately from response-generation time?
- What context is passed to a human agent when the bot escalates?

If your team uses Zendesk, ask how the agent accesses the relevant help content and how updates flow into answers. Don't assume that connecting a knowledge source guarantees accurate retrieval. Test real support questions, including near-matches, outdated phrasing, and questions the documentation does not answer. The agent should be able to say it doesn't know rather than fill the silence with a guess.

Run controlled tests before choosing a vendor

Good vendor questions lead to observable tests. Ask the vendor to configure a test environment and let your team change the relevant settings. If interruption or timeout controls are hidden behind support tickets, or if the vendor cannot show what changed between test runs, it will be difficult to tune the experience after launch.



1. **Build realistic scenarios.** Include a caller who pauses mid-sentence, changes their answer, interrupts a long response, speaks with background noise, and says nothing after a prompt.
2. **Test both channels.** Use phone lines and the website widget. Audio behavior, network conditions, and available controls may differ between channels.
3. **Change one setting at a time.** Adjust an end-of-turn threshold or silence timeout, then repeat the same script so you can tell what the change did.
4. **Record outcomes.** Track interruption detection, playback stop time, caller wait time, false interruptions, recovery behavior, and successful handoffs.
5. **Review failures with context.** Confirm that logs show the relevant transcript, timing events, tool results, and what information passed to a human.

Count dead-air seconds out loud when reviewing a test call. That simple habit makes pauses easier to notice and compare. Also listen for the opposite problem: an agent that jumps in before the caller has finished. Both are turn-taking failures, even if only one appears in a aiagentstore.ai latency dashboard.

Ask how the settings are monitored after launch

Settings that work in a pilot may behave differently with real callers, changing knowledge content, or a new integration. Ask what metrics the vendor exposes, how you can segment results by flow or channel, and whether you can inspect recordings and event traces. Look for a way to identify where the delay occurred, not just a report that says the call lasted too long.

If you are evaluating ElevenLabs or ElevenAgents, ask for a live walkthrough of the interruption and timeout controls available for your specific setup. The important question is not whether a platform has a particular feature name; it is whether your team can configure, observe, and test the behavior you need. Apply the same standard to any vendor: show the settings, run the awkward scenarios, and explain what happens when something fails.

A practical vendor checklist

- Can the vendor map the complete voice pipeline and identify failure points?
- Are end-to-end latency and individual stage timings measurable?
- Can we tune turn-taking configuration and interruption sensitivity?
- Are caller silence, end-of-turn, retrieval, and session timeouts separate settings?
- Does the agent recover gracefully from interruptions, slow lookups, and silence?
- Can we test the same scenarios across phone lines and the website widget?
- Can we inspect what the agent heard, retrieved, said, and passed to a human?

The strongest vendor conversations move beyond feature claims and into repeatable behavior. Let me tell you about a situation I encountered learned this lesson the hard way.. Ask what the agent does when a caller pauses, talks over it, changes direction, or gets no useful answer from the knowledge base. Then test those moments yourself. That is how you learn whether the timeout settings and interruptions control are adjustable—and whether the agent can keep a real support conversation on track.