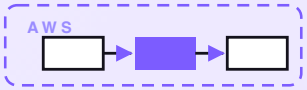
7-PART SERIES · FREE COMPANION



Voice agent

A serverless voice agent on AWS that answers your business phone, replies from your own knowledge in real time, and politely passes the rest to a human. Seven posts on the same system — one diagram at a time — with an engineering reference at the end.

CONTENTS

Voice agent

PART 1 OF 7

APRIL 28, 2026 PART 1 OF 7 · VOICE AGENT SERIES ~5 MIN READ

A voice agent on AWS for the price of a phone plan

Your business has a phone. Most of the time it rings outside hours, or while you're with another customer, or about something simple ("what time do you close?") that doesn't really need you. Here's how to build a small voice agent that picks up, answers from your own knowledge, and politely passes the rest to a human.

KEY TAKEAWAYS

- Three outside surfaces, three inside AWS. The caller, your knowledge file, and your team are wired to a listener, a brain, and a speaker.
- The brain has exactly four tools per turn: answer from the knowledge file, book an appointment, transfer to a human, or end the call gracefully.
- The agent answers only from your knowledge file. It never invents prices, hours, or promises.
- Listening, deciding, and speaking together fit in under a second so the conversation feels natural — with a graceful “one moment” stall when something runs long.
- Phone-bill territory cost: a flat fee for the number, then a few cents per minute the line is in use.

The whole system on one page

Before any code, here’s the shape of what we’re building.

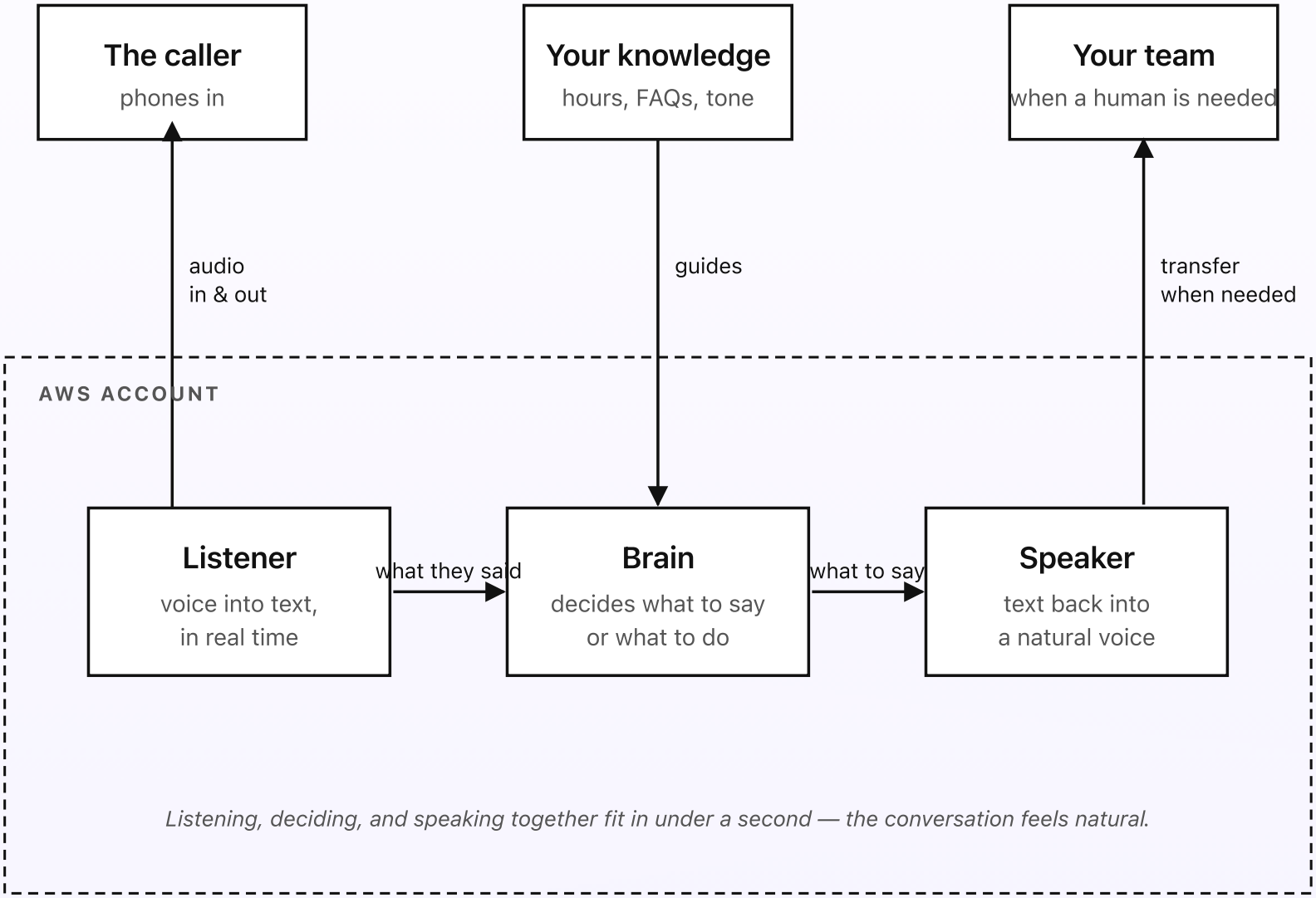

The caller
phones in
Your knowledge
hours, FAQs, tone
Your team
when a human is needed
audio
in & out
guides
transfer
when needed
AWS ACCOUNT
Listener
voice into text,
in real time
what they said
Brain
decides what to say
or what to do
what to say
Speaker
text back into
a natural voice
Listening, deciding, and speaking together fit in under a second — the conversation feels natural.

Fig 1. Three outside surfaces, three pieces inside AWS. Audio in, audio back, with a brain in the middle.

What you set up once (the outside)

- **A business phone number** — a real number callers can dial. Existing numbers can be ported in if you already have one.
- **A short knowledge file** — your opening hours, your most common questions and answers, and the tone you want the agent to use. Lives in a Google Doc you can edit anytime.
- **A way to reach a human** — a number or queue the agent transfers to when a call needs you. Your mobile, your shop's landline, a queue at your service desk — whichever fits.

What runs on every call (the inside)

- **The listener** — turns the caller's voice into text as they speak. Locks in the final version when they pause.
- **The brain** — reads the caller's words and decides one of four things: answer from the knowledge file, book the appointment, transfer to a human, or end the call gracefully.
- **The speaker** — turns the brain's reply back into a natural voice and plays it to the caller in real time.

In plain words

Someone calls. The cloud picks up. A small AI listens, decides, and replies in their voice — or hands the call to you. The whole loop takes under a second, so the caller doesn't feel like they're talking to a phone tree.

Total cost runs in phone-bill territory — a flat fee for the number, then a few cents per minute the line is in use.

DESIGN RULES THAT SHAPED EVERY DECISION

- Stay inside the AWS always-free quotas where possible. Voice has unavoidable per-minute costs, but the rest of the system stays free.
- The agent answers from your knowledge file only — never invents prices, hours, or promises.
- If the caller asks something the agent isn't sure about, it transfers. It never bluffs.
- The conversation has to feel real-time. If the agent can't reply in under a second, it stalls naturally ("let me check that for you") instead of going silent.
- Configuration lives in a Drive doc you can edit. Updating tone or hours never needs a deploy.

Why this shape

Most "voice AI" tools collapse under one of three weights: a server bill that climbs every month, replies that confidently invent prices and hours, or a robot voice that

makes callers hang up.

The architecture above is the smallest set of moving parts I could find that solves all three at once. One way in (your phone number), one way out (your team), three small pieces in the middle that listen, decide, and speak fast enough to feel natural. Everything else is plumbing.

The next five posts walk through each piece in turn — how a call connects, how the listener hears in real time, how the brain decides what to say, how the speaker stays natural, and what the whole thing actually costs. One diagram per post. A final engineering reference at the end gives engineers the dense version with precise service names and model IDs.

PART 2 OF 7

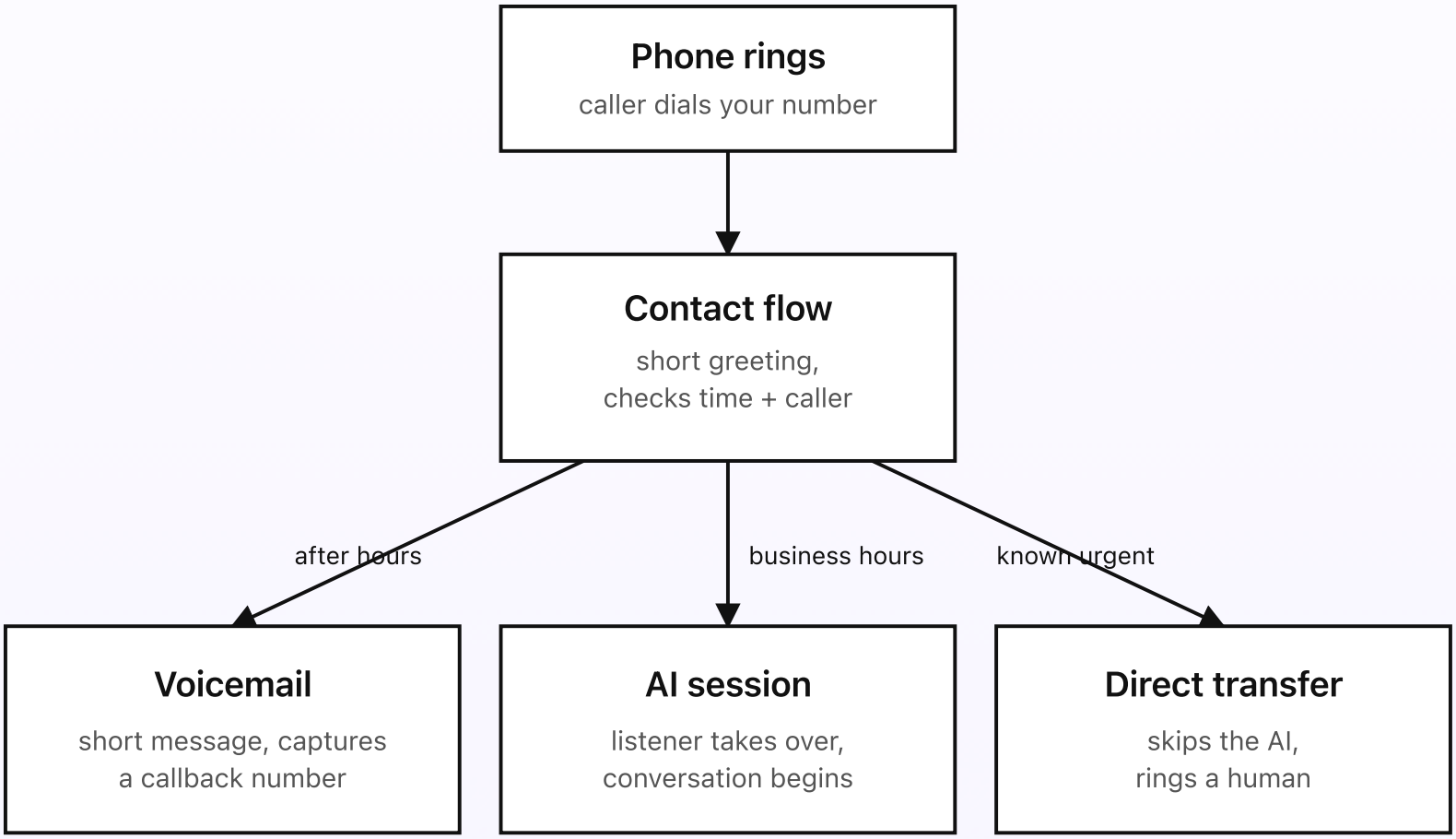
APRIL 28, 2026 PART 2 OF 7 · VOICE AGENT SERIES ~4 MIN READ

How a call connects

Before any AI hears a word, the call has to find your phone number, ring through, and decide whether you're open. The moment the caller starts talking is when the AI takes over.

KEY TAKEAWAYS

- The phone rings on a real number you claim or port in — a flat monthly fee plus pay-as-you-go minutes.
- A short greeting plays while two checks run silently: the time-of-day against your hours and the caller's number against a small VIP list.
- Three branches: after-hours goes to voicemail, business-hours starts the AI session, known-urgent numbers transfer straight to a human.
- Screening happens before any AI compute spins up — voicemail is essentially free, and VIPs never talk to a robot.
- Only the middle branch hands off to the listener. The other two run on simple rules.



By the time the AI hears anything, the call has already been screened for time-of-day and direct-transfer rules.

Fig 2. Three ways a call can go. The AI only handles the middle path; the other two run on simple rules.

The phone number itself

You need a real phone number callers can dial. The cloud lets you claim one for a flat monthly fee — the same way you'd rent a phone line from a telco, but pay-as-you-go for the actual minutes used.

If you already have a business number on a regular phone plan, you can usually port it in. Most callers don't notice the change — the number stays the same, the line just answers smarter.

What the contact flow does first

The moment the call connects, a short greeting plays — something like "Thanks for calling [your business]. One moment." While the caller hears that, the cloud quietly checks two things in the background:

- **What time is it?** Compares against your business hours from the knowledge file.
- **Who's calling?** Looks at the caller's number against a small VIP list (existing customers, urgent contacts, your top five accounts).

Both checks finish before the greeting does. The next step happens silently.

Three ways a call can go

Based on those two checks, the contact flow chooses one of three paths.

- **After hours.** The caller hears your closed-hours message and gets a chance to leave a voicemail with a callback number. You see it the next morning — the AI never runs.
- **Business hours.** The greeting ends; the listener takes over. From here, the rest of the series is what happens.
- **Known urgent number.** If the caller is on your VIP list, the AI is skipped entirely — the call is transferred straight to you. Important customers don't talk to a robot.

Why screen before the AI runs

Two reasons.

First, cost. Every minute the AI is running, it's spending money on listening, deciding, and speaking. After-hours voicemail is essentially free; routing a regular customer through the AI is a waste.

Second, trust. Some callers don't want to talk to a robot, full stop. Building a quick way to skip the AI for VIPs (or for a particular dialed extension) means the agent is never in the way of someone who needs you immediately.

In plain words

The AI is one of three possible paths, not the only path. Time-of-day decides whether you're open; the caller's number decides whether they get straight

through to you. Everything else flows into the AI session — which is where the rest of the series picks up.

PART 3 OF 7

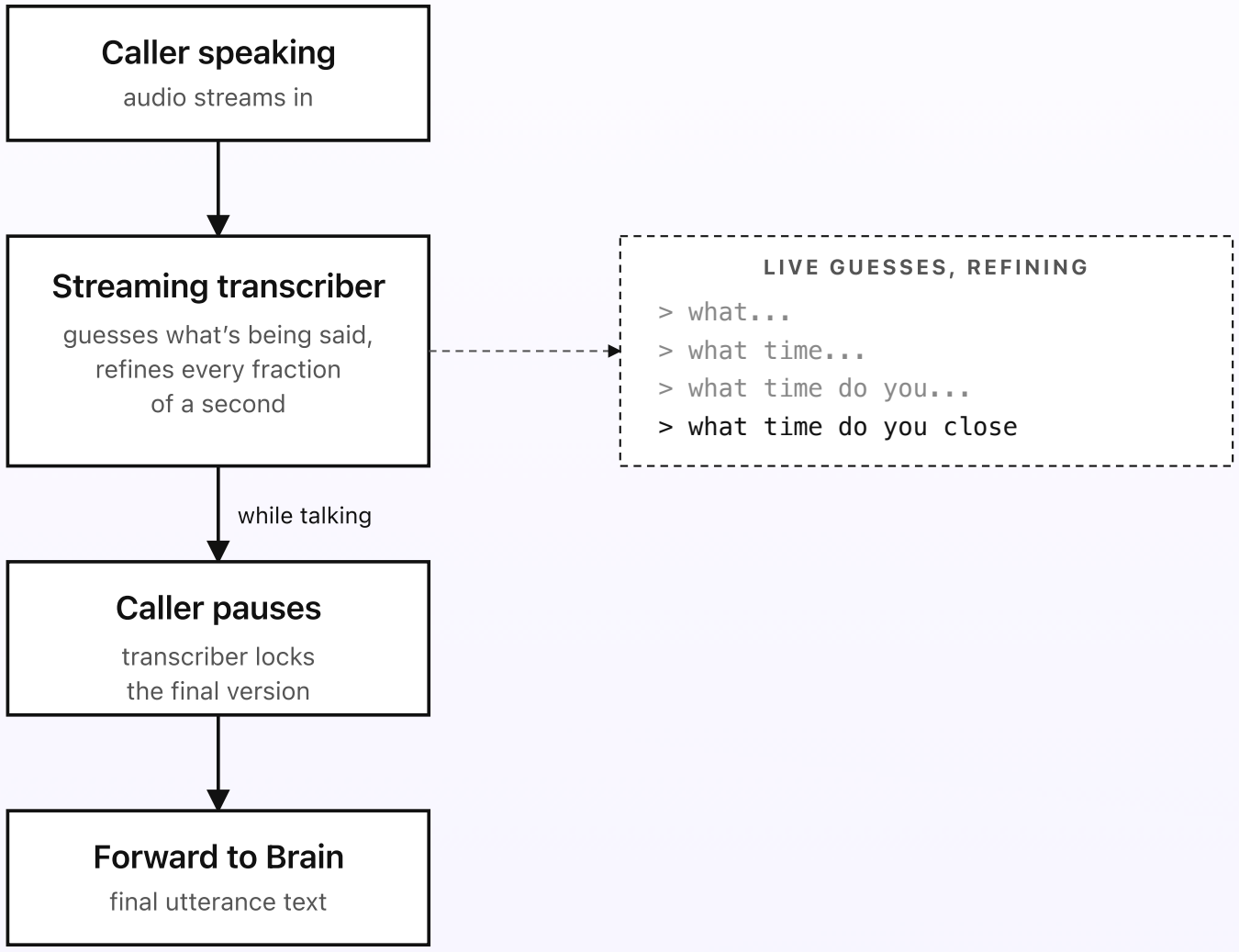
APRIL 28, 2026 PART 3 OF 7 · VOICE AGENT SERIES ~5 MIN READ

How the listener hears

The listener's job is to turn the caller's voice into text the brain can read. It does this not when they finish, but as they're talking — partial guesses get refined every fraction of a second, and when the caller pauses, the listener locks in the final version.

KEY TAKEAWAYS

- Streaming, not batched: audio flows in tiny chunks to a transcription service that produces guesses on the fly.
- Partial guesses refine every fraction of a second — "what…" becomes "what time…" becomes "what time do you close."
- Pause detection waits for a few hundred milliseconds of silence (tunable per caller) before locking the final transcript.
- The brain only ever sees the final, locked version — the partial drafts stay inside the listener.
- Background noise is filtered, accents are handled, and mixed-language callers get understood within reason — the reply still comes back in the speaker's configured voice.



The transcriber produces guesses continuously. The brain only sees the final version.

Fig 3. The listener watches the audio stream live and refines its guess as it goes; the brain only sees the locked-in final.

Streaming, not batched

The naive way to transcribe a phone call is to wait until the caller finishes speaking, send the whole audio chunk to a transcription service, get the text back, and then start thinking. That works, but the round trip is slow — by the time the caller hears anything, the conversation feels like a phone tree.

The listener works the other way around. As the caller starts talking, the audio is sent in tiny chunks to a transcription service that produces guesses on the fly. The text starts appearing within a fraction of a second of the first word.

Why partial guesses matter

Each partial guess is the listener's best attempt at what's been said so far — subject to revision. Consider a caller saying "what time do you close":

- The first guess might be just "what."
- A moment later, "what time."
- Then "what time do you…"
- Then "what time do you close."

Each new guess is sharper than the last because the listener has more audio to work with. The listener doesn't hand any of these to the brain — they're drafts. The handoff happens later.

Knowing when the caller has stopped

The trickiest part of voice is knowing when it's your turn to speak. People pause mid-sentence to think. They say "um." They take a breath. The listener has to tell the difference between "I'm thinking" and "I'm done."

The way it does this: it watches the audio for a short stretch of silence (a few hundred milliseconds — tunable per caller). When the silence is long enough to count as a real pause, the listener locks in the latest guess as final and hands the text to the brain. From here on, the brain has to be fast.

What about background noise?

The transcription service is trained to ignore everything that isn't a human voice — traffic, fans, music, dogs. It also handles a wide range of accents, microphone qualities, and call quality levels you'd expect from a real-world phone line. It's not perfect, but it's good enough for the kind of short questions a caller actually asks.

What about multiple languages?

You tell the system which language to expect (or list a small set of likely languages, and it picks). For mixed-language regions like the Philippines, callers often switch between languages mid-sentence; the listener can handle that within reason — the bot will understand the question even when half of it is in Tagalog. The reply, however, only comes back in whichever voice you've set for the speaker (typically a Singapore-English voice for the Philippines), so the agent answers in English even when the caller code-switches. If that's a deal-breaker for your

audience, the brain falls back to a polite “let me transfer you to someone who can help.”

In plain words

The listener never makes the brain wait. It transcribes as the caller talks, refines its guess until the caller pauses, then hands the final version off in less than a tenth of a second. From here, the clock is on the brain to reply — and that’s the next post.

PART 4 OF 7

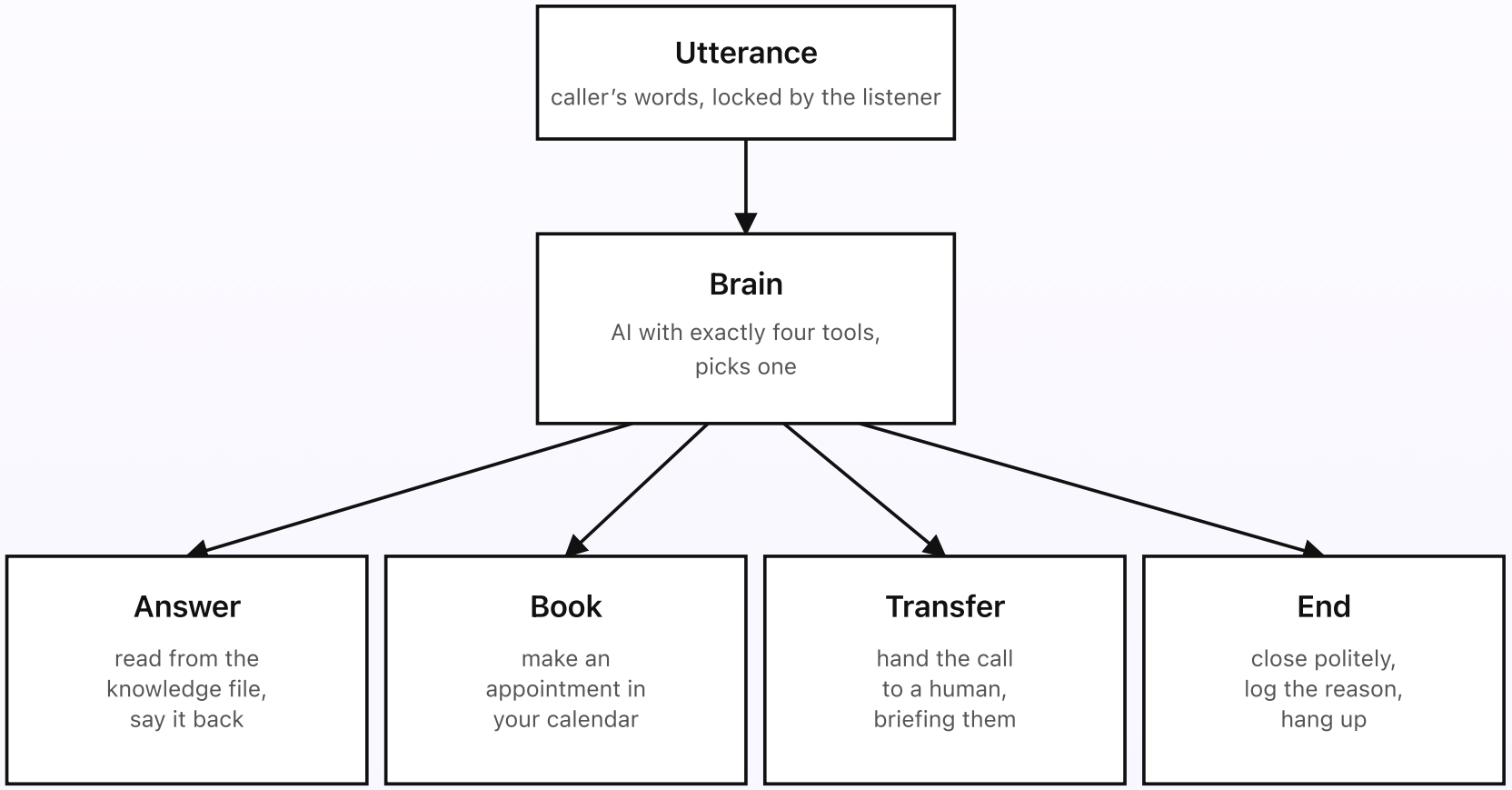
APRIL 28, 2026 PART 4 OF 7 · VOICE AGENT SERIES ~5 MIN READ

How the brain decides what to say

The brain reads the caller's words and decides one of four things: answer from the knowledge file, book the appointment, transfer to a human, or end the call gracefully. It's allowed to be confident or to defer — never to invent.

KEY TAKEAWAYS

- Exactly four tools per turn — answer, book, transfer, end — and nothing else. If the request doesn't fit, the brain transfers.
- Answers are grounded in your knowledge file. If a fact isn't there, the brain doesn't guess — it transfers.
- The booking tool follows a strict pattern: read available times, propose, confirm, write to the calendar, read back. No double-booking, no policy invention.
- Transfers come with a short brief for the human — what the caller wanted, what was already said, what to follow up on.
- Refunds, payment details, and policies outside the file are off-menu. The shape of the four tools makes "wing it" impossible.



The brain picks one tool per caller turn. It can be confident or defer — never invent.

Fig 4. Four tools, one pick per turn. The brain can't answer outside this menu.

Four tools, one decision

The most important constraint on the brain is what it's allowed to do. Most voice AIs fail because they're given too much freedom — they invent prices, promise discounts that don't exist, agree to refunds without checking. Here, the brain has exactly four tools available per turn, and nothing else. If it can't fit the caller's request into one of these four shapes, it transfers.

Tool 1 — Answer from the knowledge file

The most common tool. The brain looks up the caller's question in the knowledge file (your hours, your services, your prices, your common FAQs), composes a short reply in your tone, and the speaker says it.

The trick is grounding: the brain answers *only* from the knowledge file. If the file says you close at 6, the brain says "6." If the file doesn't mention closing time at all, the brain doesn't guess — it picks tool 3.

Tool 2 — Book an appointment

Optional. If your business takes appointments and you've connected a calendar, the brain can offer slots, take the caller's name and number, and confirm the booking back.

The booking tool follows a strict pattern: read available times, propose, get confirmation, write to the calendar, read the confirmation back. The brain doesn't book anything outside the offered slots, doesn't double-book, and doesn't make promises about cancellation policies that aren't in the file.

Tool 3 — Transfer to a human

The escape hatch. The brain picks this when:

- The question isn't covered in the knowledge file.
- The caller is upset, asking for a refund, or asking about a sensitive topic.
- The caller explicitly asks to talk to a person.
- The brain isn't confident in its answer.

The transfer is short and warm: "Let me get someone who can help with that — one moment." The brain also writes a short note for the human that picks up: what the caller wanted, what was already said, what to follow up on. The human walks into a brief, not a cold call.

Tool 4 — End the call gracefully

When the conversation has reached a natural close (caller says "thanks, that's all I needed" or similar), the brain wraps up politely and hangs up. The audit log records what was discussed and how it ended.

What it never does

A short list of things the brain refuses to do, by design:

- Quote a price, hour, or policy that isn't in the knowledge file.
- Promise anything — a refund, a callback, a delivery date — outside the booking tool.
- Take payment information.
- Continue answering when it's clearly out of its depth.

If the caller pushes for any of these, the brain transfers. The shape of the four tools makes this the only natural outcome — there's no "wing it" option in the menu.

In plain words

The brain has a small, fixed menu. Every call is a sequence of caller-turns and brain-tool-picks. The AI is fast and flexible inside that menu — it can phrase the same answer ten different ways, depending on the caller's tone — but it never tries to do something that's not on the menu. Boring, safe, correct.

PART 5 OF 7

APRIL 28, 2026 PART 5 OF 7 · VOICE AGENT SERIES ~5 MIN READ

How the speaker stays natural

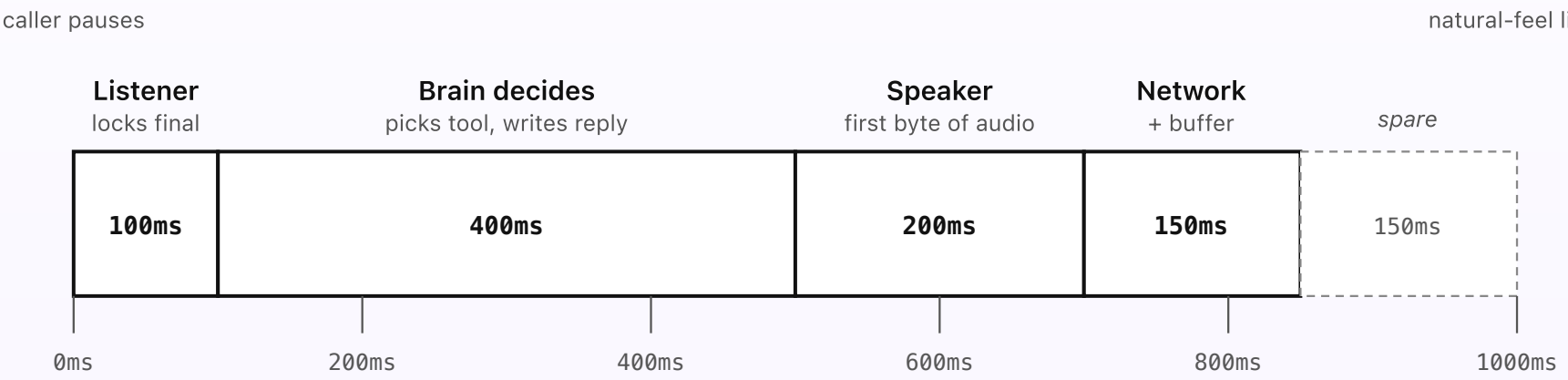
Voice is unforgiving. A two-second pause feels like a lifetime, and a stilted reply makes the caller hang up. The speaker has a budget for every step, and a streaming voice that starts talking before the brain is done.

KEY TAKEAWAYS

- One second total from caller pause to bot reply: ~100ms listener lock, ~400ms brain, ~200ms speaker first byte, ~150ms network, ~150ms spare.
- The speaker streams audio chunk-by-chunk — it starts talking on the first sentence while the brain is still writing the second.
- Modern conversational voices replace robotic text-to-speech — pick one that fits your brand the way a logo does.
- Caller interruptions stop the speaker; short acknowledgements (“mhm”) pass through without breaking the flow.
- If the budget is about to blow, a short filler (“one moment, let me check that for you”) buys two seconds — never silence.

From caller pauses to bot speaks

Total budget: under 1 second · the "feels like a real conversation" threshold



Best case: ~850ms. Real-world traffic typically lands around 1.0–1.2 seconds once cold-start protection and network jitter are included.

If the brain isn't ready in time, the system stalls naturally with "one moment" — never goes silent.

Fig 5. The latency budget. Each step has a target; the spare at the end soaks up the inevitable jitter.

Why latency is the design problem

For text chat, a two-second reply is fine. Email is fine if you reply in an hour. Voice is different: a two-second pause feels like a lifetime to the caller. They'll start to repeat themselves, ask "hello?", or hang up.

The threshold for "feels like a real conversation" is around one second from when the caller stops talking to when they hear something back. Above that, the conversation breaks. The whole pipeline is designed around staying inside that one-second window on every reply.

The budget

One second sounds like plenty until you start carving it up:

- **Listener locks final** — about a tenth of a second after the caller pauses, the listener locks in the transcript and hands it to the brain. (Most of this is the listener double-checking the caller really stopped, not just took a breath.)
- **Brain decides** — the AI reads what the caller said, picks a tool, writes the reply. About four-tenths of a second.
- **Speaker first byte** — the speaker turns the first chunk of text into audio. About two-tenths of a second.
- **Network and audio buffer** — the audio chunk travels to the caller's phone and starts playing. About a tenth and a half.

That's 850 milliseconds, with 150 milliseconds of spare for when something is slower than usual. The budget is tight on purpose — if you let any one step take longer, another step has to make it up.

In practice, real traffic lands around 1 to 1.2 seconds once you account for the cloud waking up, network hiccups, and the occasional long question from the caller. That's still inside the "feels like a real conversation" threshold for most callers — but only just. The diagram above is the target, not the average.

Streaming the voice back

The most important trick: the speaker doesn't wait for the brain to finish writing the entire reply. The moment the brain writes the first sentence, the speaker starts synthesising it — while the brain is still thinking about the second sentence.

From the caller's side, the bot starts talking almost immediately, and the rest of the reply just keeps flowing naturally. The actual synthesis and the brain's thinking overlap in time.

Sounding like a person, not a phone tree

The voice itself matters. Old-school text-to-speech sounds robotic and formal — the kind of voice that makes callers say "agent" in frustration. Modern voices, properly chosen, can sound conversational, with natural pauses and inflection. The speaker uses the modern kind — it costs slightly more per character, but it's the difference between a real conversation and a phone tree.

You also get to pick a voice that matches your business. A clinic might want a calm, warm voice. A restaurant might want a friendly, energetic one. The voice is part of the brand, the same way your logo is.

When the caller interrupts

Sometimes the caller starts talking while the bot is still mid-reply. Real conversations do this all the time. The pipeline detects new caller audio, stops the speaker, and hands the new words to the brain.

This is harder than it sounds — you have to tell the difference between “the caller cut me off” and “the caller said ‘mhm’ in agreement.” The listener watches for substance, not just sound. Short acknowledgements pass through; new questions stop the bot.

When the budget blows

Sometimes the brain takes too long — the question is unusual, the AI service is slow, the network has a hiccup. If the budget is about to blow, the system stalls gracefully with a short filler (“one moment, let me check that for you”) before going silent. The caller hears the system thinking, not the system frozen.

That filler buys another two seconds, which is usually enough. If even that doesn’t finish, the brain transfers to a human.

In plain words

Voice is harder than text because the caller is on the other end with no patience and no second screen. The speaker’s job isn’t just to read a reply — it’s to do it fast enough that the conversation feels natural. Every step has a budget; every step starts before the previous one is fully done. That’s the only way to fit a real conversation in under a second.

PART 6 OF 7

APRIL 28, 2026 PART 6 OF 7 · VOICE AGENT SERIES ~4 MIN READ

What the voice agent costs

Voice is more expensive than the other systems on this blog. There's a real phone number to pay for, and listening and speaking aren't free. But it's still less than a human receptionist by a lot — and the system sleeps when the phone isn't ringing.

KEY TAKEAWAYS

- About $35–$50 per month at typical SMB volume (~200 call minutes). The phone number alone is most of the bill until call time gets serious.
- Fixed monthly: ~$22 for the claimed number, ~40¢ per Secrets Manager secret, cents for S3. The phone number is the floor.
- Per-minute on active calls: ~2¢ for call minutes, ~2¢ for listening, ~3¢ for speaking. A 5-minute call costs about 25–50¢.
- The brain charges pennies per call — 50 calls/month is roughly 50¢ to $1 in AI fees.
- No always-on server, no managed voice-bot platform markup, 7-day log retention — an $80 monthly Budget alarm catches anything weird.

Where the dollars go in a typical month

ALWAYS FREE

$0

Lambda runs	`free`
Queues, alerts	`free`
Small tables	`free`
Webhook URLs	`free`

all under the perpetual free tier

FIXED EACH MONTH

~ $23/mo

Phone number	`~$22`
Password vault	`~$0.40 each`
S3 storage	`cents`

the phone number is the floor

GROWS WITH CALL TIME

~ $10–$25/mo

Call minutes	`~2¢/min`
Listening	`~2¢/min`
Speaking	`~3¢/min`
AI brain	`cents/call`

scales with how often it rings

TOTAL, TYPICAL MONTH

about $35–$50 / month at SMB volume

~200 call minutes/month; budget alarm at $80 catches anything weird

The phone number is the floor. Everything else scales with how often it rings.

Fig 6. Three tiers of cost. The phone number sets the floor; per-minute costs are where heavy use shows up.

The phone number

Voice has one cost the other systems on this blog don't: a real phone number that callers can dial. The cloud rents you one for a flat monthly fee — about a couple of tens of dollars a month, depending on the region and whether it's a regular number or a toll-free one.

This is the floor of your bill. Even if no one ever calls, you pay this every month. There's no way around it — phone numbers cost money to keep alive, and that's true whether the line answers as your AI agent or as a human receptionist.

Per-minute costs

Three small per-minute costs add up while a call is in progress:

- **Call minutes.** The cloud charges roughly a couple of cents per minute the line is in use. Common across all phone services.
- **Listening.** There's a per-minute charge for turning voice into text — another couple of cents.
- **Speaking.** There's a per-character charge for turning text back into voice. For a typical reply, that comes out to a few cents per minute of conversation.

Together, these run about a nickel to a dime per minute of actual call time. A 5-minute call costs around 25 to 50 cents.

AI per call

The brain charges per call, not per minute — pennies per call at the volumes this kind of system handles. A typical SMB getting 50 calls a month might pay around 50 cents to a dollar in AI fees total.

Three traps you're avoiding

- **No always-on server** — would be $30+ a month before answering anything. The voice agent only spends compute when the phone is actually ringing.
- **No managed voice-AI service** — specialist voice-bot platforms charge $50+ per month with a per-minute markup on top. This is cheaper at the bottom and stays cheaper as you scale.
- **No infinite logs** — 7-day retention. Logs can't pile into a slow-growing surprise bill.

When this stops being cheap

The math turns at high volume. If your line is taking thousands of call minutes a month, the per-minute costs add up — you might be looking at $200 a month or more.

At that point you're competing with the cost of an actual receptionist, which might be the right answer. But for the SMB this design targets — a line that rings a few dozen times a day, mostly with simple questions — the agent stays cheaper than any human option, by a wide margin.

In plain words

Phone-bill territory at typical small-business volume. The phone number alone is most of the bill until call time gets serious. Set a budget alarm that fits your expected volume and the bill can’t surprise you.

PART 7 OF 7

APRIL 28, 2026 PART 7 OF 7 · VOICE AGENT SERIES ~3 MIN READ

Engineering reference: the voice agent architecture

Same system as the rest of the series, drawn purely for engineers. Service names, resource identifiers, region, Bedrock model IDs, and the actual flow operations — everything you'd need to recreate this in your own AWS account.

KEY TAKEAWAYS

- Single AWS account in `ap-southeast-1` (Singapore); Bedrock via Global cross-Region inference.
- Five subsystems: Build & Deploy, Config Sync, Connect (call entry + audio), Listener + Brain (per turn), Speaker.
- Audio bridge: Amazon Connect ↔ Kinesis Video Streams ↔ `fn-call-orchestrator`; Polly Bidirectional Streaming API (March 2026) drives the speaker.
- Model: `global.anthropic.claude-haiku-4-5-20251001-v1:0` with strict tool_use over four tools (`answer_from_knowledge`, `book_appointment`, `transfer_to_human`, `end_call`); knowledge retrieval via `vec-knowledge` on S3 Vectors.
- Alternatives noted but off-path: Connect AI agents (less code, less control) and Amazon Nova 2 Sonic (single-call speech-to-speech, not yet in `ap-southeast-1`).

Posts 1–6 walk through the system in plain language. This page is the dense version — no softening, just the architecture as you'd sketch it on a whiteboard during a design review.

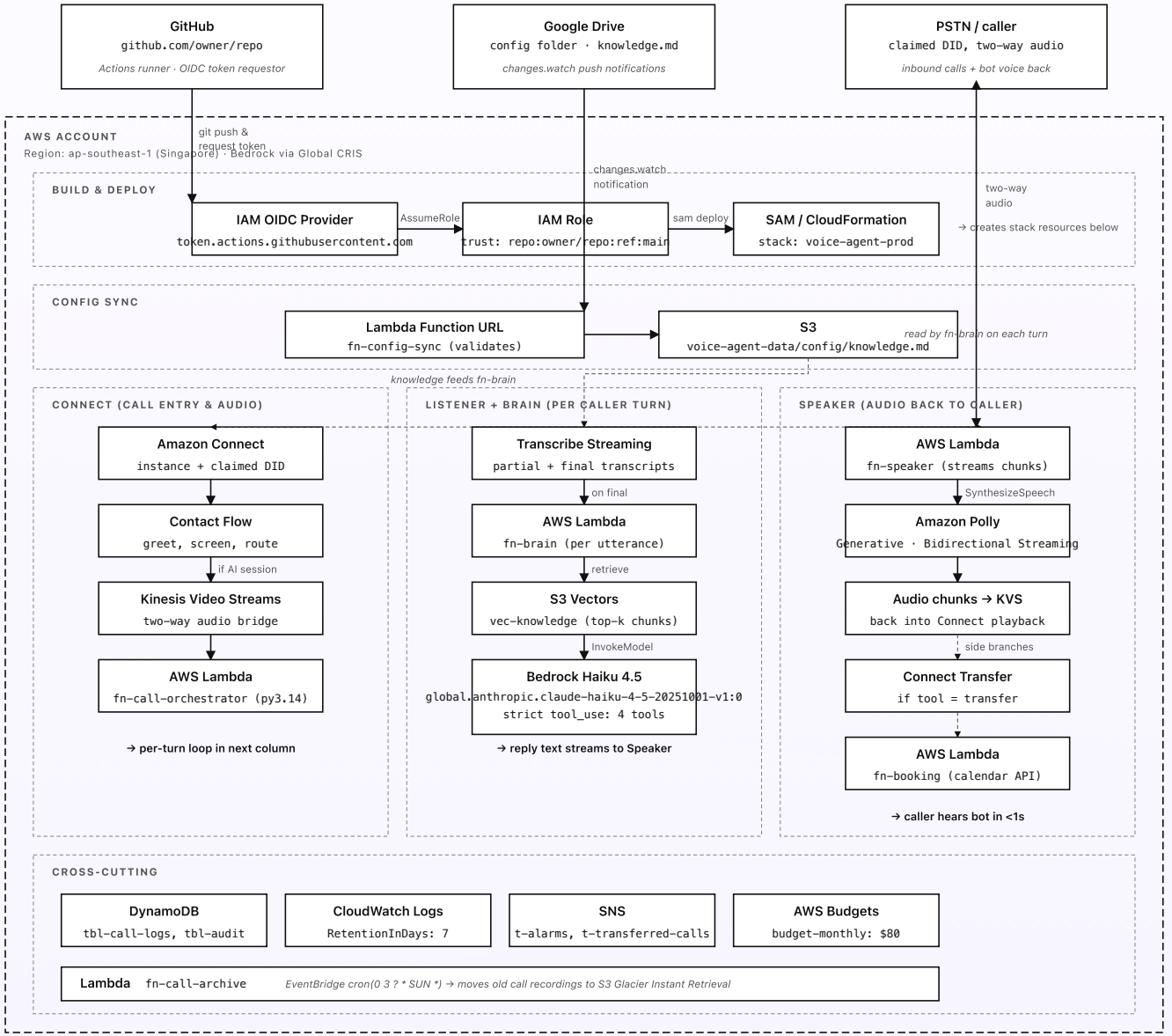

GitHub
github.com/owner/repo
Actions runner · OIDC token requestor
Google Drive
config folder · knowledge.md
changes.watch push notifications
PSTN / caller
claimed DID, two-way audio
inbound calls + bot voice back
AWS ACCOUNT
Region: ap-southeast-1 (Singapore) · Bedrock via Global CRIS
git push & request token
changes.watch notification
two-way audio
BUILD & DEPLOY
IAM OIDC Provider
token.actions.githubusercontent.com
AssumeRole
IAM Role
trust: repo:owner/repo:ref:main
sam deploy
SAM / CloudFormation
stack: voice-agent-prod
→ creates stack resources below
CONFIG SYNC
Lambda Function URL
fn-config-sync (validates)
S3
voice-agent-data/config/knowledge.md
read by fn-brain on each turn
knowledge feeds fn-brain
CONNECT (CALL ENTRY & AUDIO)
LISTENER + BRAIN (PER CALLER TURN)
SPEAKER (AUDIO BACK TO CALLER)
Amazon Connect
instance + claimed DID
Contact Flow
greet, screen, route
if AI session
Kinesis Video Streams
two-way audio bridge
AWS Lambda
fn-call-orchestrator (py3.14)
→ per-turn loop in next column
Transcribe Streaming
partial + final transcripts
on final
AWS Lambda
fn-brain (per utterance)
retrieve
S3 Vectors
vec-knowledge (top-k chunks)
InvokeModel
Bedrock Haiku 4.5
global.anthropic.claude-haiku-4-5-20251001-v1:0
strict tool_use: 4 tools
→ reply text streams to Speaker
AWS Lambda
fn-speaker (streams chunks)
SynthesizeSpeech
Amazon Polly
Generative · Bidirectional Streaming
Audio chunks → KVS
back into Connect playback
side branches
Connect Transfer
if tool = transfer
AWS Lambda
fn-booking (calendar API)
→ caller hears bot in <1s
CROSS-CUTTING
DynamoDB
tbl-call-logs, tbl-audit
CloudWatch Logs
RetentionInDays: 7
SNS
t-alarms, t-transferred-calls
AWS Budgets
budget-monthly: $80
Lambda fn-call-archive
EventBridge cron(0 3 ? * SUN *) → moves old call recordings to S3 Glacier Instant Retrieval

Fig 7. Full architecture, ap-southeast-1. White boxes = AWS resources; dashed AWS container; dashed grey boxes = subsystem groupings; dashed grey arrows = config feed and side branches.

Read this top-down, then column-by-column

Top row is the three external surfaces. Below it, the AWS account contains five subsystems: Build & Deploy across the top, then Config Sync, then three runtime columns (Connect, Listener+Brain, Speaker), with a Cross-cutting strip at the bottom. The bidirectional audio arrow runs from the caller (top right) all the way down to Connect (bottom left) — carrying caller speech in and bot voice back out on the same channel. The dashed grey arrow from Config Sync to `fn-brain` shows the knowledge dependency — the brain reads the latest knowledge file from S3 on every turn.

Naming conventions used in the diagram

- **Lambda functions:** `fn-<purpose>` — `fn-call-orchestrator`, `fn-brain`, `fn-speaker`, `fn-booking`, `fn-config-sync`, `fn-call-archive`.
- **DynamoDB tables:** `tbl-call-logs`, `tbl-audit`.
- **SNS topics:** `t-alarms` for general failures, `t-transferred-calls` for human-handoff notifications.
- **S3 layout:** single bucket `voice-agent-data` with prefixes `config/`, `recordings/{date}/`, `archive/`.
- **S3 Vectors index:** `vec-knowledge` — chunked + embedded knowledge file for top-k retrieval.

Region and Bedrock model access

Everything runs in `ap-southeast-1` (Singapore) for low latency from the Philippines. Bedrock model invocations use the **Global cross-Region inference** profile (model IDs prefixed with `global.`) — data at rest stays in Singapore; inference may route to other regions for capacity. Pricing is the same as on-demand Singapore pricing.

The brain uses **strict tool_use**: four tool definitions (`answer_from_knowledge`, `book_appointment`, `transfer_to_human`, `end_call`) with required parameter schemas, so the model can only emit a structured tool call — not a free-text reply. Free text would let the model invent prices or promises; tool_use makes that mathematically impossible.

What's deliberately not on the diagram

- IAM policy details — per-Lambda execution role inline policies are minimal (one bucket prefix, one table, one Connect instance as appropriate).
- Per-business knowledge schema — the `knowledge.md` file is a single Drive doc with sections for hours, services, FAQs, and tone. Updating sections updates the agent's answers without a deploy.
- X-Ray tracing — on for `fn-brain` and `fn-speaker`, sampling 100% during tuning, 10% in steady state. Latency is the design problem; tracing is non-negotiable here.
- The CloudFormation parameters for the Bedrock model ID and the Polly voice are templated, so swapping voices or models doesn't require code changes.

- **Connect AI agents** — AWS's built-in path with native `ESCALATION` and `HANDOFF` tools and managed Bedrock integration. Less code than the custom KVS+Lambda path here, at the cost of less control over tool schemas and retrieval. Worth picking when the four tools and bring-your-own knowledge file aren't strictly required.
- **Amazon Nova 2 Sonic** — Anthropic-style speech-to-speech via a single Bedrock model, GA December 2025. Collapses the Transcribe + Bedrock + Polly three-stage path into one call when it's available in your region (currently us-east-1, us-west-2, ap-northeast-1; not yet in ap-southeast-1).
- Live agent escalation via **Connect Tasks** — the natural follow-up to a transfer. Tasks now ship with AI-powered overviews and recommended next actions, so the human picking up doesn't walk into a cold call.

IF YOU'RE RECREATING THIS

Start with Build & Deploy alone (a single Lambda, no triggers). Once `git push` reliably updates an empty stack, claim a phone number on Connect and get a static greeting playing. Then a contact flow with the time-of-day check. Then the audio bridge into `fn-call-orchestrator`. Then the Listener + Brain on a single hard-coded tool. Then the Speaker. Then the other three tools. Cross-cutting (audit, logs, alarms, budget, archive) goes in from day one.