



CASE STUDY · AI VOICE PLATFORM

THE CALL ROOM

AN AI AGENT THAT MAKES THE CALLS, ON A PLATFORM A BUSINESS CAN RUN.

I designed and built a full-stack AI calling platform on my own in early 2025, a couple of months after voice agents were first released.

RESULT

Launched for two businesses by October 2025, with more than 500 test calls behind it.

MY PART

All of it. The idea and design, the architecture, full-stack development (backend, databases, frontend and integrations), the voice agent and its tools, testing, deployment and ongoing updates.

WHEN

Built from early 2025. Launched October 2025, and developed further into January 2026.

BUILT WITH

ElevenLabs Conversational AI, Twilio, my own Node.js server on Render, MongoDB, Supabase, Cal.com, Microsoft Outlook, and custom MCP servers for Render, Twilio and MongoDB.

01

WHERE IT STARTED

The idea was mine. Over Christmas 2024 I saw ElevenLabs announce voice agents that could hold a real phone call, and it immediately popped into my head that this could be something huge. Cre8tive AI had only ever made video, and I was sure we could offer more than that. So I took it to the directors: I could build an AI calling system for Investor Signals, which had a lot of leads and former customers that nobody had the time to ring. I convinced them, and I started building.

02

THE JOB

Investor Signals is a share-market advisory service with a list of former customers it wanted back. The plan was a calling campaign: ring each of them, present all the new features, upgrades and changes since they were last a customer, and finish by offering a 14-day free trial. If they accepted, take down their name, number and email and send them the link that activates the trial.

With a person on the phone, that's days of repeating the same call. My answer was a fully automated AI agent to make every one of them.

03

WHEN I BUILT IT

This was early 2025. ElevenLabs had released the first conversational voice agents a couple of months earlier and was the only company offering them. The technology was completely novel, and there were basically no good resources on how to make a voice agent a good one and, most importantly, a reliable one. I jumped on it as soon as it came out and worked that out myself.

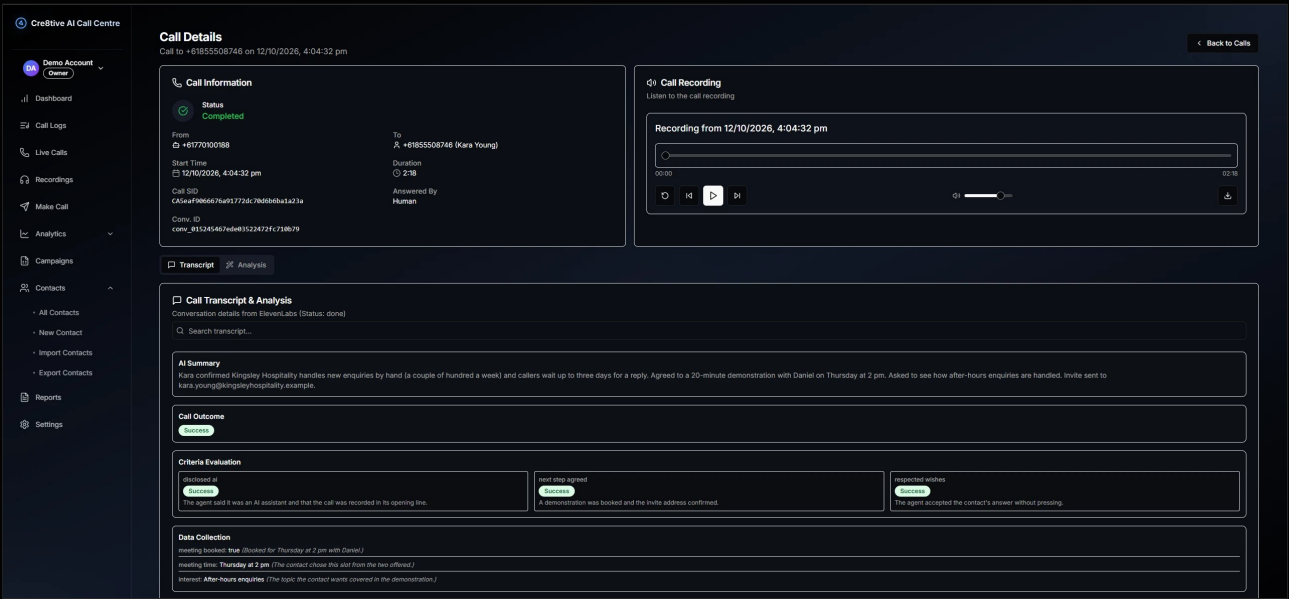
04

WHAT I BUILT

I designed and built the whole platform on my own, full stack: the backend, the databases, the frontend, the integrations, the testing and the deployment.

To the people using it, it works like a CRM. I did that deliberately, so the team at Investor Signals could operate it without much of a learning curve. You import your contacts from your existing CRM as a CSV or Excel file. If the data comes in messy or badly formatted, scripts running in the background sort it into the right columns and structure, so nothing goes wrong when a campaign starts. You set up the campaign, give it the agent designed for it, and it calls every person on the list one after another, noting what happened on each call, such as an answering machine or a voice message left. It takes inbound calls as well, and each business gets its own phone number.

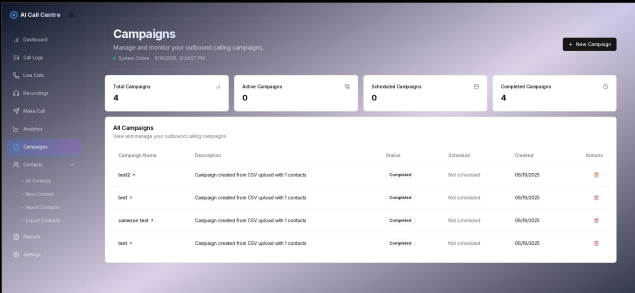
Every call is recorded. After each one, a post-processing step rates the quality of the call from the person's own words, the conversation the agent delivered, the time between responses and any interruptions.



What the platform keeps for every call: the recording, a summary, the outcome, the checks it passed and the full conversation. Shown on a demo account, because customers' calls stay private.

JUNE 2025

The campaign screen, June 2025. Each campaign has its own contact list and its own agent.



04

WHAT I BUILT

CONTINUED

It also handles what goes wrong in a real campaign. You can pause a campaign partway through and resume it later. The same contact can never be called twice. And if the phone account runs low on credit, the campaign pauses itself.

Underneath, ElevenLabs runs the voice agent and Twilio gives it a real phone line. My own server sits between the two, carrying the live audio in both directions and handing the agent what it needs to know about the person it's calling. Everything gathered is stored in its own database, with separate authentication and security in front of it.



The live audio path on every call: the person on the phone, Twilio, my server, the ElevenLabs agent, and back again.

05

THE HARDEST PART: A VOICE AGENT THAT'S RELIABLE

Building and designing the voice agent itself was by far the most difficult part of the whole build.

I was customising its system prompt, the instructions that tell the agent how to run the call, non-stop: running test calls, recording the results, tweaking the prompt and calling again.

I built the custom tools the agent needed to act during a call. It could book a meeting straight into a calendar system while the person was still on the line. It could send them a personalised email from Outlook afterwards, with more information on the areas they'd shown interest in during the call. Behind those sat small scripts I wrote for things like converting times to UTC, so that what the agent said on the phone matched what the calendar service needed.

ElevenLabs didn't give you full developer integration with your own code back then. You had to use the custom tool options on their platform, and many of those rely on a prompt telling the model when to use the tool, which is unreliable by nature. It took a lot of fine tuning and testing to get the agent using the right tool at the right moment reliably.

I also chose the AI model that powers the agent, balancing speed, the quality of the conversation, and how reliably the model follows the call process and calls its tools.



How the agent got reliable: design the behaviour, choose the model, build the tools, call, review, and go round again.

06 FIXING THE DELAY

On a phone call, a pause that runs slightly long is what makes an AI feel uncanny. Early calls had inconsistent latency, and my leading explanation was the number of server hops across a wide geographical range: the people being called were in Australia and the services were in America.

I didn't guess. I cloned the service into a sandbox, tested several different server configurations, measured the latency of each one against the baseline, and moved the part that made the biggest difference. That was the database, which matched my original hypothesis.

I DIDN'T GUESS.

07 HOW I BUILT IT

The coding agents people take for granted now weren't there yet. MCP, the standard for connecting tools to AI models, had only been announced in November 2024, and Claude Desktop had just made it possible to plug those tools in. So I connected file access, a terminal, search and a memory database, set up handoffs so each new session picked up where the last one stopped, and turned Claude Desktop, a chat app, into my own agentic coding environment: an AI that could read the project, write the code and run it. That's what I built the platform in.

Once the platform was in production, I built custom MCP servers so my agents could handle bugs and issues themselves. Back then Supabase was the only part of my stack with an official MCP. Nothing else in my backend had one, so I built my own: one for Render, to read the backend server logs, one for Twilio, to read the call logs, and one for MongoDB. They increased the speed of development exponentially, because my agents could build, test, monitor the live system and fix problems autonomously. That was a big deal in early to mid 2025.

08

TESTED ON REAL
CALLS

I ran more than 500 test calls before launch, to real customers as well as to myself and the team. Across its life, from March 2025 to May 2026, it made and took 894 calls through the phone network.

Because the technology was so new, the first thing I wanted data on wasn't speed. It was how people react to speaking with an AI agent at all. Almost nobody had heard one, and a business that gets this wrong loses its customers' trust. So one rule never moved: the agent always introduced itself as an AI.

The team and I couldn't judge that ourselves. We knew it was an AI and were tuned to look for any sign of it. The customers were the real test, and even with that introduction, the majority of them didn't realise they were speaking to an AI. Nobody reacted negatively to it.

THE CUSTOMERS WERE THE REAL TEST,
AND EVEN WITH THAT INTRODUCTION,
THE MAJORITY OF THEM DIDN'T REALISE
THEY WERE SPEAKING TO AN AI.

09

TWO BUSINESSES

Cre8tive AI then commissioned the platform as a product of its own. I rebuilt it so that any number of businesses could use it side by side, each with its own login, its own contacts, campaigns and call records, and its own agents. By October 2025 I had launched the platform for two businesses.

10

IF I BUILT IT TODAY

Voice AI has moved a long way since early 2025. If I built this again today I'd make it significantly more reliable, with ultra-low latency. I'd give the agents more useful and helpful tools to work with during a live call. I'd make them more adaptable to the person on the other end, with a high standard for how satisfied that person is when the call ends. And I'd integrate it into a larger agentic ecosystem, so a call becomes one step in a more robust pipeline.

A voice demo is the easy part. A business needs everything around it: a tool its staff can run on day one, campaigns that cope when something goes wrong, a record of every call, and an agent it can trust with its customers. I built all of that on my own, on technology that was a couple of months old.

Watch the film →



CAMERON TANG · AI SYSTEMS ARCHITECT AND AI FILM DIRECTOR

I BUILD AI SYSTEMS THAT BUSINESSES RUN ON.
I DIRECT AI FILMS.