

When Your AI Agents Learn to Speak

Adam Ladachowski | saiden.dev | June 2026

Last week I shipped on-device voice cloning that runs natively on Apple Silicon. No cloud APIs. No GPU rental. Just a Mac and a 5-second audio sample.

It started as a simple question: can an AI coding agent talk back to you — in a voice you designed?

Turns out, the answer required building more than I expected.

The Stack

Three hosts, one mesh. A macOS workstation, two Linux servers, all connected over WireGuard. Each runs an AI coding environment with persistent sessions, semantic memory, and cross-machine task dispatch.

Specialist agents. Instead of one monolithic assistant, I run dedicated agents for different domains — GitHub automation, infrastructure management, communications. They operate in parallel, report to a dispatcher, and share a common memory layer.

Semantic memory. 5,500+ entries in a hybrid retrieval system — full-text keyword search fused with vector cosine similarity via Reciprocal Rank Fusion. The agents remember decisions, procedures, and context across sessions.

Voice synthesis. A GPU TTS daemon supporting multiple engines (Chatterbox, Piper, VoxCPM2, XTTS) for cloned voice output. And now: a native macOS CLI that runs VoxCPM2 via MLX at half the RAM of the server version — with streaming that the server doesn't have yet.

What I Learned

The interesting part isn't any single component. It's what happens when you wire them together.

An agent that can search your codebase, dispatch tasks to other agents, remember what worked last time, and speak the result back to you in a voice you chose — that's not a chatbot. That's a development environment.

The "architects of intent" framing that's appearing in job descriptions now? That's the real shift. You stop writing code line by line and start describing what you want, with constraints, validation criteria, and context. The agents execute. You review.

Open Source

Six repositories are public on my Gitea instance:

- **tensors** — CLI for safetensor inspection and image generation
- **browse** — Browser automation MCP server for AI agents
- **chat** — Web interface with persona calibration and TTS
- **madcat-say** — On-device voice cloning via VoxCPM2/MLX
- **lora** — LoRA fine-tuning pipeline for persona and voice adapters

- **spark** — Fine-tuning recipes for the NVIDIA DGX Spark

All at repos.saiden.dev/aladac

Building in the open because the best way to prove you can build something is to let people read the source.