Ivan Fedyunin

Senior software engineer · LLM systems, data pipelines, agent-first development

Santiago, Chile · Remote · US time zone overlap, UTC-3/-4 · Fluent English

ivan.fedyunin@gmail.com · github.com/unmanbearpig · ivanfedyunin.com · linkedin.com/in/ifedyunin · hostelpunk.com

ABOUT

I've been building backends for 15 years. For clients, my AI work has mostly been data: scrapers and LLM pipelines that turn messy documents into searchable data. HostelPunk is where I went deeper, with labeled evals, small models replacing LLM calls, and local inference. Now I want AI to be the main thing I work on.

Since Claude Opus 4.5 came out in November 2025, agents write pretty much all of my code. I learned to make that work on HostelPunk and at Oversize.io. Both needed far more than I could build by hand, so the agents had to work without messing up the codebase and without me checking everything after them. I keep learning which models can do what, which workflows hold up, and how to turn a hard, vague problem into something agents can solve on their own. I hate boring work and agents are great at it, so these days I pretty much only do the fun parts.

I'm a geek and I love computers. I know how operating systems, databases and networks behave, and I use that to pick tools. That's why HostelPunk's 74M reviews fit on a $20/month bare-metal server, and why a client's hosting bill went from hundreds of dollars a month to about $50 when I moved them off Heroku. I've written code in 20-odd languages and shipped production work in eight of them: Ruby, Python, Go, Rust, Java, C#, Tcl and Bash.

HOW I WORK WITH AGENTS

- I run several Codex and Claude Code sessions at once in my own fork of T3 Code, an app for driving coding agents. Agents made about 50 changes to it in three weeks, from CLI-style keyboard control to a benchmark of WebSocket against WebTransport on bad networks. I also built claudex-limits to keep an eye on usage limits, and got a PR merged into Herdr, a coding-agent runtime.
- Before a big feature gets built, agents on different models review the spec and question each other until it's clear. The plan and my decisions go into Git with the reasons.
- Tests come from a separate session, so they don't share the implementation's blind spots. Fuzz targets are cheap now, so agents write those too.
- Reviewing agents check the work. One noticed the implementing agent had skipped a database regression test, and ran it.
- The more a mistake would cost, the more models check the change. For UI work, agents write browser tests that record video.

WHAT I'M BUILDING

HostelPunk · solo founder · 2025-present

A hostel search engine. It ranks hostels by what guests write about meeting people and having fun, and AI does all the reading.

- 74M reviews from four sites, normalized, and 22,600 hostels matched across their catalogs.
- Wherever a small classifier matches the LLM's F1 on labeled data, the classifier replaces the LLM. One local model handles the easy quarter of a step's calls and agrees with the LLM 99.6% of the time.
- Go and HTMX, Python workers, PostgreSQL/PostGIS, and a Rust process that owns a DuckDB/Parquet store. Moving the analysis data there freed 70 GB in Postgres. Everything runs on a $20/month shared bare-metal server, with inference on my own machines.

Research platform · personal · 2026

Crawlers with stealth browsers, a PostgreSQL graph of 1.7M companies, and agents that collect, label and rank what the crawlers bring in. Agents wrote all of it, and they deploy it too.

Photo pipeline for Vestuario Antiguo · part-time · 2025-present

Vestuario Antiguo is a fashion atelier in Santiago. DINOv2 matches studio photos to garments, Grounding DINO crops them, and an LLM writes the listing. A shoot turns into draft listings in under an hour. A Lightroom plugin and

a small Go tool move photos between the site and Lightroom for retouching. The models run on a workstation, and I picked them by benchmark.

PAID WORK

Oversize.io · senior engineer, contract · 2022-2026

The founder first hired me to move his apps off Heroku onto bare metal, which also made deploys about 10x faster. After that I worked across five of his products.

- Built an LLM pipeline that turns PDFs and HTML from hundreds of government sites into searchable PostgreSQL sections, with full text and vectors.
- Used AI to add scrapers for 2 to 20 new websites a day.
- Built an EV recommendation site for Chile. It matches each car across local and international sources and ranks them by five-year cost.
- Built speech-data pipelines for a pronunciation app, 3.7M+ labeled records, and hooked up speech-to-IPA models.

Transcribun · founder · 2023

A Rust service that turned recorded interviews into draft articles. Journalists paid for it.

Infrastructure consulting · 2025-2026

Bare-metal Linux servers for clients, with backups that alert when they go stale and deploys the clients run themselves.

ExamSuccess · main developer · 2016-2024

Eight years as the main developer of an Australian education platform. I built its practice-question system in 2019, and learners submitted 2.2M answers by mid-2022.

Jualo · 2018

Made high-traffic pages on an Indonesian marketplace with 4M monthly users about 10x faster.

Earlier · 2009-2016

Scraping infrastructure at Rookee that ran through thousands of rotating IPs. A Tcl test framework and test cases at RSoft/Brocade that ran on real Fibre Channel networks of Brocade switches. Freelance web apps.

HACKS

- **dwl.** My fork of a Wayland compositor written in C. Tiled windows share one border instead of each drawing its own. Agents wrote it and tested it with sanitizers against real Wayland and XWayland clients.
- **Blinky from scratch.** Bare-metal Rust on an STM32, starting from the reset handler. I wrote it by hand in 2021, before AI.
- **unmenu.** A macOS launcher. The fuzzy matcher is Rust, called from Swift over FFI.
- **Mouse overclocking.** A small out-of-tree kernel patch for my own mouse's polling rate, and a write-up on how I measured it.

TOOLS

AI: LLM pipelines, evals on labeled data, classifiers, embeddings, model routing, local inference, vision models

Systems: Linux, kernel, networking, OpenBSD, ZFS, bare metal, systemd, nginx, Docker

Data: PostgreSQL/PostGIS, DuckDB/Parquet, SQLite, scraping, document extraction, entity resolution

Languages: Go, Python, Rust, Ruby/Rails, Java, C#, Tcl, Bash, SQL