
SUMMARY

Sixteen years in web development, the last three leading frontend — now owning the entire frontend of a payments fintech on my own. Fintech, marketplaces, education, sport and media; products with millions of users, and products grown from an empty repository to release. I set up how software ships — architecture, review, CI, the conversation with design and product — and at work every change that ships goes through my review.

How I work in 2026. Coding agents (Claude Code) write the code; I work with them the way a lead works with a team — and how closely I read depends on what is at stake.

- **At work** — production payments software — I review everything: each intermediate change, and then the final merge request once more, as if seeing it for the first time.
- In review I watch the balance of lines added and removed, and how much of a change goes into self-documentation and auto-filled context (CLAUDE.md, specs, plans), so that a lost session does not mean the agent parsing the whole project again.
- Spec first: requirements, constraints and the plan are written down before the agent starts, and the product and infrastructure calls stay mine.
- **On personal projects** (below) I am the product owner: I write the spec, make the product and infrastructure decisions and test the result as a user would — but I don't read the code line by line, and the technical design inside is the agent's.

PERSONAL PROJECTS, 2026 · BUILT WITH CLAUDE CODE FROM MY SPEC

Messenger archive and a daily triage board

Personal system on a home server · Node.js, SQLite, Docker, MCP, whisper.cpp

- Two WhatsApp accounts and Telegram archived into SQLite; voice notes transcribed locally on the integrated GPU (Vulkan), no audio leaves the house.
- MCP servers give an AI assistant agentic retrieval (RAG through tool calls over SQLite, no vector store) across years of chats and voice transcripts. Sending is off by default, limited to saved contacts, rate-limited and logged: every message an agent reads is untrusted input, so prompt injection is designed for.
- An agent harness fills a triage board: headless Claude Code runs only when private chats go quiet after new messages, with an allowlist of tools, read-only sources and its own MCP server as the only way to write. It never retells what I wrote myself; past events clear away without spending a token.

My part: the idea and the requirements — transcription stays at home, nothing is sent without my confirmation, the board shows only what I haven't seen — and living with it daily to find what is wrong. **Claude Code's:** the code and the technical design.

Ирпотeka — a board-game library for a group of friends

boardgames.olegsuv.dev · Netlify Functions and Blobs, Telegram bot, BoardGameGeek

- Moved from a spreadsheet rebuilt by an AI every hour to a static page with one function over a key-value store; concurrent edits are safe through ETag-conditional writes.
- Sign-in through a Telegram bot: a one-time code, the bot's webhook, a signed session — friends own their entries without passwords.
- An LLM enrichment pipeline identifies new games and writes their descriptions: structured JSON output, validation, retries with a cap, covers and ratings from the BoardGameGeek API. It asks the site for free and runs a model only when there is something new, once per batch.

My part: moving off the spreadsheet, sign-in with a Telegram account, previews and editing for friends, flat box art, spending tokens only on new games; testing on desktop and phone. **Claude Code's:** the code and the technical design — storage, concurrency, sessions.

Home infrastructure

Ubuntu, Docker, Cloudflare Tunnel and Access, Tailscale, restic, Immich

- Home services published through Cloudflare Tunnel with no open ports; private pages behind Cloudflare Access, with the app verifying Access's signed token itself rather than trusting a header.
 - Family photo library (Immich) with face recognition offloaded to a desktop GPU during the first import.
 - Backups to relatives' PCs through append-only repositories, so a compromised server cannot delete them.
- My part:** the choices — a domain of my own, no open ports, a login wall instead of shared keys, backups that survive a stolen laptop, photos at home rather than in the cloud. **Claude Code's:** the setup and configuration.

EXPERIENCE

Frontend Lead · Concrypt (payments fintech), remote 03/2024 – present

The company's whole frontend, solo: the web interfaces of a payments platform for businesses — card acquiring, alternative payment methods, bank transfers and payouts. React SPA on Vite, Ant Design, Styled Components.

Frontend Lead · 1000Farmacie, remote 02/2023 – 02/2024

Joined a fast-growing marketplace to lead the frontend: built the team's development process, fixed the build at its foundations, upgraded the stack and SEO — with a measurable lift in revenue — and integrated Google Ads, Meta Pixel, OptiMonk and Clerk personalisation.

Senior Frontend Developer · Svitla Systems, remote 01/2019 – 12/2022

- **CreateMe** — a kiosk application for ordering in real time; kiosk and standalone modes for the US and UK rollout.
- **Noodle** — new features for an education platform, and a development process that made releases predictable.
- **Clearstream / Deutsche Börse Group** — a client-critical portal: automated upload and validation, security and performance work.

Senior Frontend Developer · Infront X (formerly Omnigon), Kyiv / remote 02/2015 – 12/2018

- **PGA Tour** — moved the site's components from jQuery to React; 99.8% unit-test coverage.
- **Corebine CMS** — led a small team building a CMS for sports websites from scratch.

Frontend Web Developer · NOOSPHERE, Ukraine 09/2014 – 01/2015

An internal single-page application for a support team; migrated it from Ext JS 4 to Ext JS 5. Before that, freelance web development.

AI-NATIVE DEVELOPMENT

- Claude Code
- Reviewing agent-written code
- Spec-driven development
- Context engineering
- Agent context files (CLAUDE.md)
- MCP
- Agent harnesses
- Agentic RAG
- LLM cost control

SKILLS

- TypeScript
- React
- Next.js
- Node.js
- Vite
- Webpack
- Redux
- Ant Design
- Material UI
- Styled Components
- SCSS
- Jest
- CI/CD
- Docker
- Cloudflare
- Netlify
- SQLite
- GA4 / GTM

EDUCATION & LANGUAGES

Master's degree, Computer Science — National Metallurgical Academy of Ukraine.

Ukrainian — native · English — professional · Spanish — conversational.