

Ilya Trapeznikov

Digital Product Maker & AI Systems Builder

Open to remote and relocation

ilyatrapeznikov.com | Telegram @Ilya_S_Tr | info@astrabrain.io | GitHub | LinkedIn

PROFILE

I build digital products solo, from a business hypothesis to working production. I am the person who takes a business problem to a shipped release, not a list of technologies. I pick the method to fit the problem, not the other way around. I work asynchronously and at high density, and I measure success only by what ships.

Before IT, I spent 19 years building and running real businesses: starting companies from zero, owning sales, procurement, projects and IT infrastructure, and writing technical specs for tenders. For several years I developed a corporate CRM on 1C, constantly translating between the language of the business and the language of the developers, and back.

There used to be a middleman between the business's "I want this" and the programmer's "I can't", and the meaning got lost on the way. That link is gone now: I set the task and carry it to code myself, so I see the product from the money side and the code side at once.

HOW I WORK

- I take a business problem and carry it to release solo: requirements, architecture, development, deploy.
- I design what defines the product: the data model, tenant isolation, the math and the business logic, the contracts between layers.
- I act as my own technical customer: I set the task, define the constraints, and accept and check the result.
- I delegate routine code generation to AI, and keep the decisions and the responsibility myself.
- I reverse-engineer closed APIs and stitch disconnected systems into one product.

PRODUCT EXPERIENCE (INDEPENDENT, SHIPPED SOLO)

AstraBrain AI Astrology SaaS astrabrain.io 2026

- Split the system in two: an astronomy core on native Swiss Ephemeris (C) computes the exact chart, and the AI (Gemini) only receives finished numbers and writes the human reading, so it never makes the stars up.
- Isolated the native C ephemeris add-on onto Node functions, working around its incompatibility with Vercel's Edge runtime.
- Built a single surface (web and a Telegram Mini App): content unlocks instantly over Supabase Realtime, with three ways to pay (Telegram Stars, crypto, bonus balance).
- Generated ~3,500 landing pages across 10 languages for organic search.

Built with: Next.js 16 · React 19 · Supabase (Realtime) · Gemini · swisseph (C) · React Three Fiber · grammy

TZ Portal GovTech Procurement SaaS (44-FZ) tzfz.ru 2026

- Isolated each company's data at the PostgreSQL RLS level: in a multi-tenant system, one company's data stays separate from every other's.
- Built a step-by-step builder for 44-FZ procurement specs with hierarchical document sets and export of ready-to-sign files in Word and Excel.
- Deployed an nginx gateway on a Russian VPS so Vercel's cloud can reach state registries past geo-blocks.
- The AI fills values strictly inside a fixed template, each with its justification, and leaves anything it cannot support blank for a person to check.

Built with: Next.js 16 · React 19 · Supabase (Postgres + RLS) · Gemini · docx · exceljs · Vercel

TRPZLabs Quant Radar Quant Strategy Validation 2026

- Built a matrix backtest engine on vectorized computation: thousands of parameter combinations in one pass, about 77 times faster than the obvious loop (roughly 13 minutes down to 10 seconds per instrument).
- Built an overfitting-rejection pipeline: Walk-Forward, Out-of-Sample, Monte Carlo and two more independent checks against luck.
- Modeled leveraged trading through virtual capital with margin rebase, which the underlying library does not support, with no look-ahead.

Built with: Python · Streamlit · VectorBT · pandas · NumPy · ccxt · SQLite · Pine Script

Signage CMS Interactive Installation CMS (white-label) 2025

- Designed a no-code scene builder so staff assemble and change the screens themselves, with no developer and no code.
- Built a master-and-slave architecture: one server keeps physically separate screens in step through WebSocket rooms, and it all runs offline.
- Implemented state recovery: after a power cut a “blind” screen comes back on its own to exactly what it was showing.
- Already live in a museum, running the “Siege of Leningrad” exhibition.

Built with: [Vue 3](#) · [Node.js / Express](#) · [Socket.IO](#) · [PostgreSQL 15](#) · [Electron](#) · [Docker Compose](#)

Aviozo Travel-Social SaaS aviozo.com

2026

- Designed the group-planning model: a “worst-case visa” calculation across the whole group's passports and destination ranking by a composite metric.
- Built a server-side LLM layer with function-calling and streamed answers from an AI guide grounded in the user's own data; all paid service keys stay on the server, never in the browser.
- Tightened the access policies (RLS) so users can never see each other's data.

Built with: [Flutter Web](#) · [Supabase \(Realtime\)](#) · [Vercel Functions](#) · [Gemini](#) · [Telegram Bot API](#)

Deal Intelligence Internal BI Dashboard

2024–2026

- Stitched CRM deals together with a closed procurement-and-warehouse system into one BI layer by reverse-engineering an undocumented API.
- Fixed a systemic error of naively summing quantities that pushed contract fulfillment past 100%.
- Restored ~90% of deals that arrived with no stage or program by reconciling against internal ids: 14 of 16 contracts saved that would otherwise have dropped silently from the report.
- Sped up the build with parallel requests ([ThreadPoolExecutor](#)) and overnight data preparation.

Built with: [Python](#) · [Streamlit](#) · [pandas](#) · [requests](#) · [SQLite](#) · [nginx](#)

CAREER HISTORY

Project Manager · TEK0 (tsml.ru)

since Dec 2024

Ran the full project cycle: client, proposals, specs, delivery, on-site installation. Automated the manual parts for myself: the commercial-proposal generation, and a layer over the CRM that removed manual deal tracking (the Deal Intelligence dashboard above).

Project Lead · own practice, financial-instrument analysis

2020–2024

Built tools and algorithms for analyzing financial instruments and trading at different risk levels (the basis for TRPZLabs Quant Radar).

Technical Director & Company Manager · Gramat, B2B trade

2016–2022

Helped launch the company from zero and built its infrastructure, still running today. Led the technical staff, scoped projects and tender specs, and handled suppliers, clients and public authorities.

IT Specialist · Miratorg, agro-industrial retail holding

2011–2016

Maintained a fleet of about 200 workstations, ran user support (service desk), deployed and supported OS and software, and helped roll out new internal services.

PC repair service engineer and retail sales

2006–2011

Hands-on PC repair and retail sales (top salesperson, 2007).

EDUCATION

- Incomplete higher education: SPbGETU “LETI” (Saint Petersburg Electrotechnical University), Faculty of Computer Science and Technology.
- “Business Incubator” training center, 2002: operator-programmer.
- English: B1 (working proficiency; reading technical docs fluently, spoken improving). Self-taught engineering.

TECHNOLOGIES ACROSS THE PROJECTS

TypeScript / Next.js, React, Python (pandas, NumPy, VectorBT), Vue, Flutter, Kotlin; Supabase / PostgreSQL, Gemini and Claude, Vercel, Docker, nginx.