CONTACT INFO

- **Name**: Eli Rum
- **Title**: Co-founder & Operating CTO
- **Location**: Moscow / Yerevan / Tbilisi (Remote-first)
- **Phone**: +7 999 821-54-53
- **Email**: explosivebit@gmail.com
- **LinkedIn**: linkedin.com/in/elirum
- **Languages**: Russian (native), English (B1)

My main phrase for 2024-2026

From Idea to MVP with AI and MCP

WHO I AM

Serial technical founder. Over 15+ years I went from a system administrator in a small town to a CTO who has launched multiple products on the international market. I don't just write code — I build companies and the engineering systems that make those companies possible.

My superpower is **turning chaos into a working product**. I can take a napkin idea and in 6 weeks show investors a working MVP with thoughtful architecture, security, and CI/CD. I can take an existing team and bring order: standards, processes, predictable delivery.

Equal parts founder and craftsman. I care not only about *what* we build but *how*: I study 60+ open-source projects before writing the first line, I write formal architecture documents — and still deliver fast.

WHAT I DO

I build products from zero to production. Not prototypes — systems that live for years, scale, and don't break.

I make small teams dangerously effective. With a team of 6, we built a product that earned Product of the Day/Week/Month on Product Hunt. Solo, with AI tools, I work with the output of a 5-7 engineer team.

I help investors avoid losing money. I conduct technical audits of startups — assessing codebase health, architectural risks, real team velocity. The investor gets an honest report and can make decisions based on facts, not founder promises.

I take on select outsource projects. When the challenge is technically interesting — architecture, optimization, AI integration — I'm happy to help.

MY STORY

The Beginning: Hardware (2000-2011)

From childhood I was fascinated by electronics. I built devices from radio components. I was active in radio clubs and moto/auto workshops.

First Steps in IT: From Hardware to Code (2011-2016)

I started as a system administrator — servers, networks, hardware. Then CIO at an advertising agency: responsible for everything from server procurement to implementing 1C and CCTV systems. This taught me the most important lesson: **technology exists not for technology's sake, but for business**. When you're the only technical person in a company, you quickly learn to think about outcomes, not stacks.

My Own Business: 100+ Projects (2012-2017)

I founded a web studio with a partner — the same partner I still work with today at Orchestra (orch.so). Over 5 years we delivered 100+ client projects. I learned the most critical CTO skill: **estimate, promise, and deliver**. Managing client expectations, running multiple projects in parallel, building processes so the team works without micromanagement. This isn't about code — it's about delivery.

First Startup: Multiroad (2017-2022)

A digital freight forwarder — multimodal routes (sea + rail + road + air), cost calculation, booking. **We shipped the MVP fast and secured pre-seed investment from the very first pitch.** The product didn't achieve the expected growth — the market turned out to be harder than it seemed. But this experience gave me two lessons worth more than any success:

1. **An MVP is not "bad code done quickly"** — it's a minimum product with the right foundation
2. **Technology won't save bad unit economics** — but good architecture gives you a chance to pivot
3. Today this project is still running and generating revenue for the co-founder we started with, and there's a deal underway with an oil company. The project's architecture is still the one I built.

Orchestra: 5.5 Years, Product Hunt, 5 Platforms (2020-Present)

Co-founder and Operating CTO. Together with 3 co-founders and a team of ~6 people, we built a B2B SaaS for task management and team communications. The result:

- **Product of the Day, Week, and Month** on Product Hunt in the Productivity category — a rare triple recognition
- A single codebase running on **5 platforms**: Web, iOS, Android, macOS, Windows
- A massive number of production releases over 5.5 years — stable, predictable delivery
- AI assistant inside the product: 8+ LLM providers, autonomous agents, meeting transcription
- A huge amount of software and various engineering solutions built from scratch

The main lesson from Orchestra: **Being an Operating CTO isn't about code. It's about the delivery system.** I built processes, quality standards, release discipline — and that's what allowed a small team to compete with companies 10x larger.

gerts.ai: Solo Architect with AI (2024-Present)

From Orchestra's pain came an idea: an enterprise platform for working with knowledge and reasoning. Graph RAG, ontologies, entity extraction, access control, audit, and much more.

What makes it unique: **I built it alone**, using AI as a force multiplier. 20+ custom AI agents, 10+ MCP servers, a framework I developed for getting fast results in short timeframes — and as a single engineer I worked with the output of a full team. This isn't hype — it's an engineering approach to AI tools: formalized processes, quality gates, research-first pipeline.

The result: 50+ packages, 20+ microservices, 4,246+ tests, 76 architecture documents — one person, production-grade quality. And over 1 million lines of code in 3 months. The biggest time investment was studying the industry and technologies.

Consulting: Technical Audits for Investors (2023-Present)

Private investors ask me to look at startups from the technical side. I dive into the code, architecture, team processes — and deliver a structured report: what's real, what the risks are, where the technical debt lies. The investor takes this information and makes their decision. Objectivity and depth are what matter most in this work. Now with AI it's become easier, faster, and more comprehensive.

HOW I THINK

Research-first, delivery-focused

Before writing code, I study. For gerts.ai I analyzed over 100 open-source projects. Wrote over 100 formal architecture documents. And yet — over 1 million lines of code in 3 months. Research doesn't slow you down — it speeds you up, because you don't have to redo things.

Order from chaos

My role in every project is to turn "everything is on fire" into "predictable delivery." Code standards, automated quality checks, transparent processes, documentation. It sounds boring — but it's what separates a product that lives for 5 years from one that dies in six months.

AI as an engineering tool, not magic

I don't "use ChatGPT for code." I **engineer AI workflows**: I've created 20+ specialized agents, each with a clear area of responsibility. There's an agent for architecture review, one for deep research, one for API testing. It's a system — with rules, quality gates, persistent memory. The result: one person with the output of a team of 8.

Full-stack doesn't mean "mediocre at everything"

I work at every level: from database design and security systems to UI and UX. But it's not surface-level — at each layer I build production-grade solutions. A full OIDC provider (not a consumer, but a provider). A cross-platform product from a single codebase. Enterprise security with 5 authorization layers. 300+ routes across two frontend products.

WHAT I BRING TO A TEAM

As an Operating CTO / Head of Engineering

- I turn chaotic development into a predictable release cycle
- I build architecture that lives and scales
- I integrate AI into the development process as a system, not as hype
- I hire, train, set standards — making the team self-sufficient

As an Architect

- I design systems that don't need to be rewritten in a year
- I think about security, scaling, and cost from day one
- I document decisions so a new engineer can be productive from week one

As a Founder / Entrepreneur

- I understand that technology is a means, not an end
- I think about product-market fit, unit economics, time-to-market
- I know how to deliver with limited resources — because I've done it my entire life

As an Advisor / Consultant

- An objective view of someone else's code and architecture
- Structured reports for investment decisions
- Practical recommendations, not abstract advice

EXPERIENCE TIMELINE

Period	Company	Role	What happened
2024 — now	**Gerts Ai**	Solo-founder & Architect	Enterprise AI platform, 1m+ LOC
2023 — now	**Consulting**	Independent	Startup tech audits for investors, selective outsource
2020 — now	**Orchestra (orch.so)**	Co-founder & Operating CTO	B2B SaaS. Team ~6. Product Hunt Day/Week/Month. 300+K LOC, 5 platforms
2017 — now	**AWAYO**	CTO & Co-founder	IT solutions. Multi-project curation. Technology adoption
2017 — 2022	**Multiroad**	CTO & Co-founder	Digital freight forwarding. Pre-seed funded. Sea + rail routes
2012 — 2017	**Web-Studio**	Founder	100+ client projects. Full delivery lifecycle
2016 — 2017	**STRIKT**	CIO	Project management, app development
2014 — 2016	**LemonLime**	CIO	Full IT operations for ad agency
2011 — 2014	**SysAdmin**	System Administrator	Servers, networks, infrastructure — Kostroma

---

NUMBERS THAT MATTER

Combined product portfolio

- **1M+ lines of code** across 2 production products
- **100+ packages** in 2 TypeScript monorepos
- **40 microservices** in production
- **5,378+ automated tests** — quality is not optional
- **47+ production releases** — predictable delivery
- **5 platforms from 1 codebase** — Web, iOS, Android, Desktop, PWA

Engineering discipline

- **76 architecture documents** (RFCs) written before coding
- **67 reference implementations** studied before building
- **99 bugs tracked** with severity levels — nothing gets lost

AI as engineering tool

- **20+ custom AI agents** — each with a specific job
- **10+ MCP servers** — AI connected to real systems
- **7-8x productivity multiplier** — one engineer, team output

Business outcomes

- **Product of the Day/Week/Month** — Product Hunt (LAUNCH by Jason Calacanis) (Productivity)
- **Pre-seed investment** — from first investor pitch (Multiroad)
- **100+ client projects** — delivered over 5 years (Web-Studio)
- **3+ companies founded** — serial entrepreneur
- **15+ years** — from sysadmin to CTO to founder to investor advisor

---

EDUCATION & GROWTH

- **Kostroma Machine-Building Technical College** — Radio engineering. Started from hardware — understanding the physical layer shapes how I think about systems
- **Business Design Course** (2013) — When I realized technology alone isn't enough
- **Startup Leadership Program** (2023) — Alumni. Structured approach to building companies
- **MCP Protocol Certification** — Introduction + Advanced. AI tool engineering
- **HighLoad** (2016-2018), **RIT++** (2017) — Russia's top engineering conferences

---

WHAT I'M LOOKING FOR

I'm looking for a role where my responsibility is **not features, but outcomes**:

- **Operating CTO / Head of Engineering** — delivery velocity, architecture, quality, team
- **Principal Architect / Head of Platform** — infrastructure, reliability, developer productivity
- **Co-founder (technical)** — if the product and team inspire me
- **Solo-founder** — Right now, AI is Rock-and-Roll time

Format: office, hybrid, or remote. B2B, AI, SaaS, Platform — my domains. Since I built a remote-first startup, I know how to deliver anything remotely.

ONE LAST THING

People often ask me: "How did you build alone what usually takes a team of 7-9?"

The answer is simple: **I'm not alone.** I have 20+ AI agents, 10+ connected systems, a formalized research-to-build pipeline, and persistent memory across sessions. 15+ years of development and entrepreneurship, and 25+ years of understanding technology and engineering. It's not magic — it's an engineering approach to everything that interests me. And to AI tools especially. And I can build the same system for any team.

The difference between "I use AI" and "I engineer AI processes" is the difference between "I sometimes ask ChatGPT for advice" and "I have 20+ specialized agents with clear areas of responsibility, 10+ automated workflow commands, and persistent memory that remembers context between sessions." Anyone can do the first. The second is a competitive advantage. I also manage all of this from my own project Orchestra, which fits perfectly into management workflows and has its own AI agents built in.