

Ilia Zakharov

Hands-on Engineering Leader

ilja.s.zakharov@gmail.com · github.com/IljaZakharov · [LinkedIn](#) · zakharov.systems
Serbia · Open to relocation (London/EU) · Available for remote (EU/US overlap)

Engineering leader who owns systems end to end — planning and shaping technical direction, engineering delivery and culture, and growing the team. I strengthen the team where it counts: hands-on contribution and technical guidance, mentoring engineers, managing stakeholder expectations, and building a flywheel for predictable delivery. I bring 15+ years in engineering — formal methods, distributed systems, blockchain, operating systems, AI agents, and infrastructure — and 7+ years of technical leadership (owning projects end-to-end, leading audits, and working directly with customers), including 6 years of people management leading engineering teams — a systematic approach grounded in a CS PhD and an executive MBA.

PROFESSIONAL EXPERIENCE

Pi Squared Inc. (Pi2 Labs) — Engineering Manager → Head of Engineering

US-based startup (pre-seed → pre-Series A) · Remote (distributed team: Asia/Europe/US) · 2023-2026

- **Owned engineering culture across the org through the pre-seed → pre-Series A stages** — took mostly-PhD research engineers and helped them adapt to product development with execution practices that made them ship reliably; built the repeatable system for sourcing, hiring, onboarding, and team formation, composing teams on demand from a live inventory of skills, growing the team from 5 to 27 along the way.
- **Built an internal AI-agent operations layer used across all departments** — an API for agents (Hermes/OpenClaw) over vector search on Postgres/pgvector and a LangGraph processing layer, giving leadership live project status and alerts to stay on top of delivery, and helping everyone else automate routine and development tasks.
- **Wrote hands-on Python and TypeScript** across hard problems and ops automation — a translation layer from K-framework semantics to machine-checkable MetaMath proofs (formalizing execution traces, owning the core pipeline), plus automation connecting GitHub workflows, monitoring, reporting, and quality gates — implementing directly, shaping architecture, and reviewing the underlying codebases.
- **Helped bring Fast's payment system for agents to its first mainnet launch** — capable of handling real-value token transactions — working delivery hands-on (engineering practices, DevOps, on-call), shaping the security strategy for mainnet, and making the scaling calls for its long-term data layer under budget and cloud-cost constraints.
- **Stayed hands-on across cloud and bare-metal** as a standing build-vs-buy decision — ran production on bare-metal Hetzner with Docker Swarm at $\sim\frac{1}{3}$ of equivalent AWS/GCP compute cost, using infrastructure-as-code where it earned its keep and direct administration where it didn't, managing users, secrets, and on-call; used Vercel/Railway/Datadog when speed was binding, later swapped Datadog for an OSS stack and self-hosted GitHub runners (\$1,500→\$500/mo), holding infra at \$10-15K/mo.
- **Owned incident response and on-call hands-on across 3 time zones** — wrote the playbooks, carried the rotation personally, ran recovery directly; on a mainnet system handling real-value tokens, successfully defended against multiple perimeter attacks and attempts to compromise the payment system's validators.
- **Cut deployment from hours to 5 minutes** by tightening CI/CD, making release ownership explicit, and automating deployment and dev-to-release handoff.
- **Cut delivery lead time to under a week — from commit to production** — which let the org move from two-week to one-week sprints while holding a 70-90% sprint hit rate, a large gain in flexibility and delivery speed.
- **Shaped the roadmap with founders and ran delivery-risk analysis off a live risk backlog I built and maintained** — sequenced products against real technical risk, dependencies, and release readiness, driving them down before they surfaced as surprises, and pushed back when a plan's execution path or evidence was weak.
- **Built the org's performance and growth process** — ran 360 feedback, then replaced it with lighter bi-weekly check-ins and evaluations for a more dynamic loop — and cut team-lead attrition

from 3/year to 0 by clarifying ownership and decision boundaries and mentoring PMs and team leads.

- **Built two-way reporting** — bottom-up from engineers to leadership, top-down briefings back to teams — so status, priorities, and context moved without founder bottlenecks.

Runtime Verification — Verification Engineer

US-based (Illinois) · Remote · 2021–2023

- **Led audits of Algorand and Ethereum smart contracts** and protocol systems end-to-end — owning each engagement from first customer contact to final report, combining security reasoning, formal methods, and distributed-systems context, and guiding shadow auditors.
- **Built formal models and simulations in Isabelle/HOL** that surfaced high-impact vulnerabilities, rounding/mathematical errors, and correctness risks before release.
- **Reviewed cryptographic and zero-knowledge designs** for correctness (including proof-compression) and led development of in-house analytics/monitoring tooling for protocol behavior and operational risk.

EMURGO Learning — Guest Lecturer, Formal Verification

Los Angeles, CA · Remote, part-time · 2022–2023

- Designed and delivered an online formal-verification course across several cohorts — making formal-verification tools usable by working engineers without a PhD.

Institute for System Programming, Russian Academy of Sciences (ISP RAS) — Verification Engineer / Researcher

2012–2021

- **Built Klever**, a distributed component-wise verification framework for large C codebases, written in Python and scheduled across a Docker Swarm environment — applied to Linux kernel and proprietary filesystem code at Samsung and Huawei, surfacing 300 defects.
- **Wrote automated extraction and generation of environment models** for event-driven C systems, enabling automatic verification runs.
- **Managed and grew junior engineers (master's students) joining the team** — owned their performance, guided them technically, and drove their research projects to completion over multi-year engagements.
- **PhD in automated verification of large C programs** (program analysis, model checking, environment modeling, bug-finding), research cited 180+ times (h-index 7); 3× Google Summer of Code (Linux Foundation); Microsoft PhD Summer School, Cambridge.

TECHNICAL SKILLS

Engineering leadership: Engineering Management · Team Lead · Head of Engineering · hiring · coaching · mentoring · one-on-ones · stakeholder management · cross-functional execution · distributed teams · capacity planning

Infrastructure & platform: IaC · AWS · GCP · Hetzner · Docker Swarm · CI/CD · self-hosted GitHub runners · Datadog · Railway · Vercel · bare-metal & public cloud · build-vs-buy · infrastructure cost management · container service boundaries

Delivery & reliability: delivery planning · roadmaps · release automation · DevOps · DORA (deployment frequency, lead time, MTTR, change failure rate) · quality gates · monitoring · incident response · on-call · developer productivity

Hands-on & AI-agent systems: Python · TypeScript · C · Solidity · code review · architecture · AI-agent orchestration (LangGraph-style) · OpenClaw/Hermes agent harnesses · GitHub workflows · internal tooling

High-trust engineering: distributed systems · smart-contract audits · blockchain security · formal methods & verification · automated software verification · Isabelle/HOL · static analysis · model checking · model-based testing · TDD · fuzzing · program analysis · correctness & security analysis

EDUCATION

- Executive MBA, ESEM Business School, Belgrade — President Scholarship, 2024–2025.
- PhD in Computer Science, Institute for System Programming, Russian Academy of Sciences, 2014–2019.
- MS/BS in Applied Mathematics and Physics, Moscow Institute of Physics and Technology (MIPT), 2008–2014.