

Roman Miakotin

Technical Leader & Solution Architect — Turning Uncertainty Into Scalable Products

Founding-CTO Profile • Founding Engineer • AI-Native Delivery

Location: Kyiv, Ukraine · Open to Remote

GitHub: github.com/adverax · LinkedIn: linkedin.com/in/roman-miakotin-904831230



Metric	Value
Experience	15+ years (Go, distributed & high-load systems)
Domains	Healthcare · Telecom · Crypto
Peak designed throughput	~1M events/sec (streaming platform)
Solo end-to-end delivery	4 business-critical production services
Team building	Hired 4 senior engineers, led 6
Delivery model	Documented AI-native SDLC pipeline (full cycle)

EXECUTIVE SUMMARY

Hands-on technical leader who turns ambiguous ideas into production systems. 15+ years across Healthcare, Telecom and Cryptocurrency — from Backend Engineer to Team Lead, Solution Architect and Staff Engineer. I operate best at 0→1: setting the technical foundation, hiring the first engineers, and shaping product direction through architectural decisions.

My differentiator is a documented **AI-native delivery pipeline** covering the entire SDLC — it lets one senior engineer ship the scope of a small team, with engineering quality gates (domain modeling, TDD, ADRs, mandatory review) that keep the speed honest.

AI-NATIVE DELIVERY PIPELINE — KEY DIFFERENTIATOR

Most engineers apply AI at the coding stage. I run it across the **entire SDLC as a supervised participant**, under explicit engineering control:

- 1. Discovery & Domain.** Problem framing → EventStorming → generated domain model + dependency graph → contradiction analysis of requirements before a single line of code.
- 2. Planning.** C4 architecture views → ADRs → prioritized backlog → task decomposition into cards; kanban board and Gantt generated as by-products, not overhead.
- 3. Implementation.** Parallel execution of cards by supervised AI agents, TDD-first.
- 4. Verification.** Automated + human code review, fix loops, documentation generated alongside code.

Quality gates: domain model as the source of truth · TDD · ADR discipline · mandatory review. *AI accelerates; the engineer owns correctness.*

Evidence:

- **OpenCRM (MVP)** — built end-to-end with this pipeline, solo, in 2 weeks: a security layer, a metadata & custom-objects layer, and SOQL as the access language. [github.com/adverax/crm]
- **Nabuto** — 4 business-critical production services delivered solo (see Experience).
- Measured **3–5x cycle-time reduction** vs comparable scope delivered traditionally.
- **Published write-up** — *The bottleneck in AI-assisted development isn't generation. It's verification.* — the methodology behind the pipeline's machine-checkable quality gates. [adverax.io/blog/verification-bottleneck]

In founder terms: an investor-ready MVP in weeks, not months — with tests and documentation included, not a demo held together by tape. A seed-stage company gets multi-team throughput from a founding CTO of one.

WHAT I BRING

Technical strategy and foundation-first architecture · Building and leading small, high-leverage teams · AI-native engineering as an organization-level multiplier · High-load distributed systems delivered under uncertainty · Product-oriented systems thinking and 2–3 year system evolution.

ENGINEERING LEADERSHIP

Built an engineering team from the ground up: ran ~20 technical interviews, hired 4 senior engineers, led a team of 6. Defined engineering standards and architectural direction; established ADR-driven decision-making and a structured review culture; mentored senior engineers and aligned technical direction with product goals.

TECHNOLOGY STRATEGY & DECISION-MAKING

Foundation-first: invest in the right base instead of stacking workarounds — applied across Nabuto, Proxima, EMS and OpenCRM. Owned core technology choices and build-vs-buy decisions; architected for 2–3 year evolution, not just immediate delivery.

SIGNATURE ARCHITECTURE CASES

Nabuto Streaming Platform — streaming architecture processing ~1M events/sec, delivered as the sole engineer; four production-ready services.

Proxima Virtualization Layer — gradual modernization of a large healthcare platform with extended Row-Level Security, without a disruptive rewrite; integrated a legacy CRM with the platform's security system.

EMS Objective Function — decision-making platform constructing coherent objectives from competing directives and constraints.

AI-Native Delivery Pipeline — end-to-end AI-native SDLC, from vision to documentation (see dedicated section above).

PROFESSIONAL EXPERIENCE

Nabuto | Staff Backend Engineer / Solution Architect | 2026

- Owned end-to-end technical direction and delivery of 4 business-critical services as the sole Golang engineer — from architecture to production.
- Defined architecture and implementation strategy for a streaming platform processing ~1M events/sec.
- Designed wallet-state synchronization and multi-level risk evaluation services (individual, group, cross-exchange) for cryptocurrency platforms.
- Designed Kubernetes workload balancing architecture for horizontal scalability and minimal downtime.
- Applied the AI-native delivery pipeline across the lifecycle.

Proxima Research | Backend Engineer → Team Lead → Solution Architect | 2024-2026

- Key achievement: designed a virtualization layer enabling gradual migration of a large healthcare platform while extending Row-Level Security — without a disruptive rewrite.
- Integrated a legacy CRM with the platform's security system.
- Led 6 senior engineers; defined engineering standards and architectural direction.
- Ran ~20 technical interviews; hired 4 engineers.
- Designed pricing, discount and business-rule systems in close collaboration with product stakeholders.

Spring Radius | Architect / Lead Engineer

- Built a flexible billing system designed to operate under high load (~10,000 users, ~5,000 RPS workloads).
- Implemented sharding and scalability improvements; led PHP-to-Go modernization.
- Built an embedded RADIUS server.

CORE TECHNOLOGIES

Architecture & Patterns: DDD · Hexagonal · Microservices · Event-Driven · Event Sourcing · CQRS · API Gateway · Outbox · Saga · Distributed & High-Load Systems

Languages: Go (Golang)

Data: PostgreSQL (Row-Level Security) · Redis · MongoDB

Streaming & Messaging: Kafka (~1M events/sec) · RabbitMQ · NATS · NSQ

Infra & DevOps: Kubernetes · Docker · Terraform · GitLab CI/CD

APIs: gRPC · REST · WebSocket · OAuth2

AI-Native Engineering: Claude Code · Cursor · supervised multi-agent delivery

OPEN SOURCE & TECHNICAL ASSETS

- **OpenCRM (MVP)** — flagship proof of the AI-native pipeline: security layer, metadata & custom objects, SOQL access language; built solo in 2 weeks. [github.com/adverax/crm]
- **Dependency Injection Framework** [github.com/adverax/di]
- **Policy Framework** [github.com/adverax/policy]