

Evgenii Zakharov

Senior Full Stack Engineer · .NET, React, PostgreSQL

to@risen.dev · +37455433551 · Remote worldwide (contractor or EOR), Yerevan (UTC+4) ·

<https://github.com/risenxxx> · <https://linkedin.com/in/risenx>

PROFESSIONAL SUMMARY

Senior Full Stack Engineer, 8 years building distributed systems end-to-end — ASP.NET Core and .NET 9 on the backend, React and TypeScript on the front, PostgreSQL and EF Core underneath, and the Kubernetes infrastructure they run on. Original author of nestjs-zod, the standard way to use Zod with NestJS, at 5.5M downloads per month.

Ship production software with AI agents doing most of the typing: I write the spec, the constraints and the failing test cases before the agent starts, review every diff against them, and hold edge cases, security review and performance targets as release gates rather than follow-ups. Five production products across five domains were built this way.

Designed, built and operated distributed platforms single-handedly on Kubernetes — ASP.NET Core services with EF Core and PostgreSQL on one, Temporal, ClickHouse and Redis on another — and ship SSR-heavy Next.js products at 7.5M monthly active users against hard Core Web Vitals, latency and error-rate SLOs.

WORK EXPERIENCE

X5 Group · Remote · 04/2024 – Present

One of Europe's largest food retail groups

Senior Software Engineer, Frontend

- Shipped internal React CMS interfaces used daily by over 100 editors and moderators, aligning UI behavior with the content lifecycle and removing manual handoffs from the publishing path.
- Engineered SSR-heavy Next.js flows for a UGC recipe platform serving 7.5M monthly active users, improving Lighthouse Performance on key mobile pages by 15–25 points and stabilizing render and hydration paths.
- Improved API contracts and mapping validation, reducing data-mapping Sentry issues by 70% and raising frontend/backend contract reliability through stricter runtime checks.
- Reduced shipped JS payload by 20–35% through code splitting, dependency optimization and tree-shaking, cutting long main-thread tasks and improving TBT on key pages.
- Embedded unit, integration and contract tests into CI ahead of release, cutting regression escapes before they reached production.
- Implemented A/B testing (Varioqub) into SSR flows, partitioning cache keys by experiment bucket to preserve consistent per-user variant delivery without giving up caching.

Independent — iGaming Platforms · Remote · 2023 – 2025 · <https://github.com/sigma-gaming>

Sole engineer on two products: Sigma Games (.NET 9 + React platform) and Gift Fight (real-time Telegram Mini App, launched to production with around 1,500 players)

Platform & Full Stack Engineer

- Sigma Games — built an iGaming platform backend from scratch as 10 ASP.NET Core (.NET 9) services covering payments, wallets, affiliate, gaming and auth, with NATS pub/sub, Redis caching, distributed locking and Hangfire background jobs.
- Designed and evolved a 27-entity PostgreSQL schema with EF Core 9 across 17 migrations, keeping money movement transactional and idempotent under concurrency.
- Covered domain services with xunit unit tests and Testcontainers-backed integration tests running against real PostgreSQL, Redis and NATS instances, with Respawn resetting state between cases.
- Documented REST surfaces with OpenAPI through Swashbuckle and Scalar, and enforced nullable-reference correctness as a build error across the solution.
- Shipped the player-facing Telegram Mini App and the operator admin panel in React 18 and TypeScript, alongside an Astro authentication app and a grammY Telegram bot.
- Gift Fight — architected and operated a Telegram gaming platform as a distributed system single-handedly, separating real-time WebSocket gameplay from transfer and settlement services behind idempotent handlers and durable workflows for availability and fault tolerance under load.

- Designed a low-latency WebSocket gameplay layer on Socket.IO, holding p95 state-sync latency at 120 ms in load tests with 300 concurrent sessions, improving gameplay latency and availability under load.
- Architected multi-step game rounds using Temporal workflows, reaching a 99.5% workflow completion rate through deterministic retries, compensation and idempotent handlers to reduce settlement failures and simplify recovery.
- Implemented data layers for product and analytics workloads using PostgreSQL (Drizzle), ClickHouse and Redis/Dragonfly caching to optimize cost and query performance across hot and warm workloads.
- Isolated blockchain-based payments behind transfer services and idempotent settlement logic, and hardened the edge with Cloudflare WAF, per-route rate limiting and bot rules.
- Provisioned infrastructure as code with Pulumi across GCP, DigitalOcean, Hetzner and Cloudflare, delivered through FluxCD GitOps onto Talos Kubernetes clusters, with automated releases and rollbacks in under 3 minutes.
- Instrumented both platforms end-to-end — Serilog structured logging, Prometheus metrics and health checks on the .NET services, metrics and tracing across real-time and settlement flows — reducing time-to-detect and informing capacity decisions.

SberEducation • Remote • 09/2022 – 03/2024

EdTech division of Sber, a major European tech company

Senior Software Engineer, Frontend

- Built BFF endpoints aggregating multiple upstream services into stable client contracts, adding versioning and runtime validation to prevent breaking changes from reaching clients.
- Introduced schema-driven client generation (Hasura, GraphQL) in CI, generating TypeScript types and enforcing schema checks to shift integration failures left.
- Migrated an accessibility-first payment widget and reuse-first UI primitives, reaching Lighthouse Accessibility 95+ and standardizing keyboard and focus behavior across monetization paths.
- Delivered Telegram-integrated web apps under strict navigation and session constraints, improving Lighthouse Performance on core mobile WebView flows by 10–20 points.
- Integrated SSO and delivered core platform modules for a multi-tenant B2B marketplace launch, focusing on security boundaries and staged rollout.
- Built backend services with NestJS during a refactor of the core product — S3-backed storage and BullMQ queues for asynchronous work — and delivered a Telegram bot connecting mentors and mentees.
- Mentored mid-level engineers through code review and pairing, and drove team-wide frontend standards for API contracts, accessibility and testing to raise overall engineering quality.

Haiku Dev • Saint Petersburg • 10/2021 – 09/2022

Senior Software Engineer, Frontend

- Set and documented a shared CSS-in-JS UI kit and design primitives adopted by 4 product teams, running code reviews that cut duplicate implementations and visual drift.
- Implemented URL-synced catalog filters and an Effector GraphQL wrapper, standardizing query and state flow and removing duplicate fetch logic across pages.
- Reworked a cross-browser modal library including Safari, with focus trap, scroll lock and stacking rules to prevent UI breakage across environments.
- Prototyped a micro-frontend with Webpack Module Federation, validating remote ownership, shared-dependency strategy and independent release workflows.

Freelance • Remote • 03/2018 – 09/2021 • <https://www.upwork.com>

Full Stack Engineer

- Delivered projects end-to-end with direct client ownership over scope, deadlines, deployment and post-launch support.
- Automated a hotel-management CRM with a room-by-date booking grid and drag-to-create and drag-to-move reservations, kept responsive through virtualized rendering and a normalized client-side cache.
- Implemented server-validated conflict detection for overlapping reservations, partial-day turnovers and per-booking services, so two administrators editing the same day could not corrupt a booking.
- Integrated fiscal receipt printing end-to-end across payment, receipt issuance and printer hardware, with retries and reconciliation so no payment was recorded without a receipt and no receipt without a payment.

EDUCATION

Studies in Computer Engineering

South Ural State University (SUSU)

PROJECTS

Frame Player

07/2026 - Present • <https://github.com/risenxxx/frame-player>

Video player for Windows and macOS in Rust, Tauri 2 and libmpv, casting 4K HDR to televisions without re-encoding and streaming torrents during download, written largely by AI agents against requirements and acceptance cases I defined and verified.

nestjs-zod

05/2022 - 10/2024 • <https://github.com/BenLorantfy/nestjs-zod>

The standard way to use Zod with NestJS, at 5.5M downloads per month, where one schema serves validation, TypeScript types and generated OpenAPI docs. Transferred to another maintainer in 2024.

@cardbox/editor

05/2021 - 10/2021 • <https://github.com/cardbox/editor>

Slate-based WYSIWYG editor, shipped as a published library (ESM/CJS/UMD with type declarations).

Registry-driven architecture for actions, keybindings, decorations and per-element settings; feature modules for lists, links, headings and code with a Prism-backed decoration pass. Sole engineer on it.

Tuned

2026 - Present

AI resume tailoring: every variant is a patch over one base CV, and a provenance check refuses any claim that does not trace back to a source node. Anthropic and OpenAI-compatible adapters behind a single port, deterministic page-fitting against a real PDF renderer, Bun and TypeScript monorepo.

SKILLS

Languages: English (full professional proficiency), Russian (native)

Programming Languages: C# (.NET), JavaScript, Rust, SQL, TypeScript

Backend: MVC, API design, BFF, Fastify, GraphQL, Hasura, Hono, Microservices, NestJS, Node.js, REST, Socket.IO, WebSockets, Bun, JWT, OAuth, ASP.NET Core, Hangfire, OpenAPI/Swagger, Anthropic API, OpenAI API, RabbitMQ, WebSocket

Frontend: Bundle and Runtime Profiling, Core Web Vitals, Next.js, React, SSR, User Experience, Tailwind, ESLint

Data: ClickHouse, Dragonfly, Drizzle, NATS, PostgreSQL, Prisma, Redis, Temporal, Metabase, EF Core, Npgsql, BullMQ, Kafka, JSON

Platform: AWS, Cloudflare, Cloud Infrastructure, Docker, FluxCD, GCP, GitOps, Hetzner, IaC, Kubernetes, Pulumi, Prometheus, Talos, S3, Rybbit

Practices: AI-assisted development (Claude Code, Copilot), CI/CD, Code Review, Distributed Systems, Fault Tolerance, High Availability, Mentoring, Multi-tenant SaaS, Observability, Scalability, SLOs, System Architecture, Technical Design, Technical Documentation, unit/integration/contract testing, A/B testing, TDD, Testcontainers, Caching

Tooling: Git, Yarn, Vite, Vitest, Prettier, Nx, xunit, Serilog, Moq, TurboRepo