Oleksii Monakhov

Email: goshagriboedskii@gmail.com
LinkedIn: linkedin.com/in/alex-monakhov-5b6116238
GitHub: github.com/AlekseyMonakhov
Portfolio: oleksii-monakhov-portfolio.vercel.app

Skills

- **JavaScript (JS), TypeScript, React, Redux, Next.js, Remix**
- **Vue.js, Vuex, Nuxt.js, Svelte, SvelteKit, CSS, HTML**
- **Micro Frontend, Microservice Architecture**
- **GCP, AWS, Docker, Kubernetes**
- **MongoDB, PostgreSQL, RabbitMQ, Kafka, Redis**
- **Go, GoLang, gRPC, Protobufs**
- **WebSocket, HTTP, REST, GraphQL**
- **Node.js, Express.js, NestJS**
- **React Native**

Professional Experience

Playtech – Full Stack Developer

January 2024 – Present | Remote

- Developed a notification system that greatly enhanced user experience by ensuring timely and efficient message delivery.
- Optimized frontend-backend communication by upgrading the load balancer to support **HTTP/2**, improving request efficiency and reducing latency.
- Implemented a **service mesh** architecture in **Kubernetes** using **Istio**, simplifying logging, enabling mutual TLS (mTLS), and optimizing communication between the Istio gateway and sidecars via **HTTP/2** and **gRPC**.
- Migrated applications from **HTTP 1.1** to **HTTP/2** and introduced synchronous **gRPC-based** communication for microservices, resulting in lower latency and improved reliability.
- Designed and implemented an asynchronous microservices architecture using **Redis Streams** and **Redis Pub/Sub** as an event bus, enhancing system scalability and decoupling.

- **Proposed and led the refactoring of multiple Node.js services to Go**, securing management support for a more efficient concurrency model. The original approach used Node.js threads for parallel tasks, whereas the new solution leverages **Go's goroutines**, significantly boosting throughput and resource efficiency.

- Proposed and delivered **several new features** that received stakeholder approval and **substantially improved user experience** across the platform.

- Introduced **infrastructure-level architectural solutions** by refactoring Node.js apps previously using cluster mode. Transitioned to running **multiple container replicas** in Kubernetes and adopted **Redis** as a message broker, simplifying scaling, restarts, health checks, and load balancing.

- **Reworked a plugin-based application**—originally integrated via iframes—into a **microfrontend** architecture, streamlining development, enabling more flexible business functionality, and greatly improving the developer experience.

- Upgraded the frontend stack by migrating from **React 16.2** to **React 18.3**, refactoring legacy class components into modern functional components with hooks and best practices.

- Revamped **React Router** architecture by upgrading from v5.2 to v6.4, reducing excessive *useEffect* calls tied to data fetching and boosting application performance.

- Integrated **MUI (Material UI)** to modernize the frontend UI, replacing outdated components with optimized MUI-based solutions for a consistent, responsive design.

- Resolved database performance bottlenecks by rewriting inefficient queries, adding strategic indexing, and implementing a **Redis caching** layer, cutting query execution times from several days to just 10–15 minutes and significantly reducing database load.

FusionWorks – Full Stack Developer

May 2023 – December 2023 | Chisinau, Moldova (Hybrid)

- Enhanced microservices architecture by leveraging **NestJS** and **Kafka** as a message broker, utilizing **AWS** infrastructure and cloud functions for improved scalability and reliability.

- Advanced a **microfrontend** architecture using **Vue**, **React**, and **Svelte** to deliver diverse frontend solutions.

- Integrated **Next.js** for SEO-critical application parts and **upgraded from Next.js 12.2 to 13.4**, refactoring to the **App Router** for better performance and maintainability.

- Built a robust **design system** with **Storybook**, streamlining UI development and ensuring a consistent design language.

- Deployed and orchestrated microservices with **Kubernetes**, enabling seamless scaling, monitoring, and management of services.

Tibica – Full Stack Developer

April 2022 – March 2023 | Remote, Odessa, Ukraine

- Built **microfrontend** solutions using **React**, **TypeScript**, **Redux**, and **WebSocket** for real-time interactions and modular UI development.
- Developed **serverless** backend components on **AWS** infrastructure, leveraging **API Gateway**, **AWS Lambda**, **DynamoDB**, **RDS**, **EventBridge**, **SQS**, and **SNS** for event-driven, highly scalable services.
- Containerized and deployed **Express.js** applications on **Amazon ECS** using **AWS Fargate**, streamlining resource management and improving performance for backend workloads.

Iambloom – Full Stack Developer

January 2020 – February 2022 | Hybrid, Odessa, Ukraine

- Built a complete **e-commerce solution** from scratch, choosing **Vue** for the admin panel and **SvelteKit** for the storefront, both implemented entirely by me.
- Selected **Google Cloud** as the project's infrastructure, leveraging **Firebase Hosting** for the admin SPA, **Firebase Authentication** for secure admin access, **Cloud SQL** (Postgres) for data management, and **Cloud Storage** for image hosting.
- Deployed a **Fastify** server on **Google App Engine**, enabling streamlined CI/CD and reducing manual infrastructure overhead.
- Integrated multiple external services — including delivery providers, payment gateways, and **email notification systems** — ensuring a seamless user experience across the entire platform.
- Extended the CRM system by integrating with the **Facebook API**, centralizing chats from Facebook, Instagram, and other channels into one unified communication flow.
- Established robust communication and monitoring strategies, enabling reliable cross-service interactions and maintaining high performance.

Key Achievements

- **Mentored developers** from junior to confident mid-level, enabling them to take ownership of complex tasks and guide newer team members, which increased overall team capacity.
- **Owned full-cycle architectural decisions** spanning frontend, backend, and infrastructure — personally implementing each layer to minimize costs and maximize performance.
- **Drove development of business-critical features** by identifying user pain points, creating proofs of concept, presenting to stakeholders, and delivering fully realized solutions that significantly improved the product.

- **Led performance optimizations and refactoring** across multiple services and applications, reducing codebases by up to 50% while retaining functionality. This cut resource usage, expedited onboarding for new developers, and boosted overall system efficiency.

Education

Bachelor's Degree
V. N. Karazin Kharkiv National University

Last Updated: February 2025