

Vyacheslav Dzhagaev

Backend Engineer (NestJS / TypeScript) · Growing in Go

Volgograd, Russia · Open to relocation (EU) · slava.dzhagaev@proton.me · vdzhagaev.tech · github.com/vdzhagaev

SUMMARY

Backend engineer with **3+ years of commercial experience**, **sole architect** of a production NestJS/MongoDB platform built on **clean architecture** - domain entities free of persistence concerns, narrow purpose-built read/write ports instead of a monolithic repository layer, and a hybrid event-dispatch system for eventual consistency between aggregates. Comfortable diagnosing production incidents down to root cause and shipping the fix under real constraints (legacy schema quirks, rate-limited third-party APIs, single-writer storage engines). Actively specialising in **Go** for self-hosted, API-first services, and going deeper into **Rust** through hobby projects. Looking for a backend role in the **EU** with relocation support.

SKILLS

CORE

TypeScript, NestJS, Node.js, Go, SQL, Rust, Python

BACKEND

REST APIs, Clean architecture, DDD, Domain events, Event sourcing / outbox, CQRS-style read models, BullMQ / Redis, Playwright/CDP automation, WebSockets, FastAPI

FRONTEND

Angular, React, RxJS, Svelte

DATA

MongoDB, PostgreSQL, SQLite, Migrations

INFRA

Docker, docker-compose, Bitbucket Pipelines, Monorepo, Linux, Git

EXPERIENCE

Full-Stack Engineer, **Comrade Digital Marketing Agency**

Mar 2023 - Present

Remote

Link-building / backlink-monitoring platform (NestJS + MongoDB, pnpm monorepo) - **sole architect and primary author**

- Designed the **clean architecture** from scratch: domain entities with no persistence concerns, unit-tested without a database, plus narrow read/write ports per domain instead of one fat repository - dashboards and reports get denormalized, query-optimized data

without touching write-side invariants.

- Built a **hybrid event-dispatch system** (change-stream watch + polling fallback with lease-based claiming) for eventual consistency between aggregates, storing every event as a replayable outbox.
- Diagnosed a bulk-persist regression (40s for ~1,150 rows) as a **missing index**, not an architecture problem, before a multi-day queue-split was greenlit - the real fix cut write time to under 1 second.
- Redesigning a 30K-row scraping bulk-write that silently hung 7-21 minutes with no retries into **chunked, idempotent upsert batches** with per-chunk timeouts and queue-level retry.
- Implemented a **FIFO semaphore** gating a shared pool of browser contexts across 14 concurrent scraping workers, paired with randomized interaction pacing so repeat requests don't read as a bot to anti-automation systems.
- Run three separate persistent-session browser-automation services rather than one generic scraper, each behind its own **headful Chrome** sidecar (Xvfb + noVNC) that an operator authenticates once, kept alive via an adaptive auth-health prober and continuous ambient mouse/scroll simulation - defending against a silently logged-out or flagged-idle session, not request volume.
- Engineered an **AI-augmented extraction pipeline**: per-platform field rules live in a data-driven catalog, not hardcoded, and feed a composable, cache-friendly prompt that discovers a stable CSS selector once per page template - reused with zero further AI calls, proven at 5/5 fields from cache on a second account.

Editorial workflow tool (NestJS + MongoDB) - **lead contributor, founding author**

- Root-caused two production incidents to fire-and-forget webhook delivery (a data race and a silent message loss), then designed a **transactional outbox** - atomic status flag, lease-based polling fallback, backoff with jitter, dead-lettering.
- Found a bulk-approval path that bypassed the new outbox entirely, silently skipping notifications - fixed with transactional snapshotting on the same query path instead of duplicating outbox logic.
- Delivered the matching Angular feature (server-side row model grids) end-to-end on top of this backend.

Shared frontend platform (Angular) - **contributor to specific modules**

- Built the real-time **WebSocket layer** for a set of collaborative-editing features, through a later migration and event-schema alignment with the backend.
- Adapted an existing **undo/redo engine** to a second data domain via a generic adapter interface without touching the engine's core, including a TTL soft-delete edge case that could otherwise silently corrupt restores.

Monitoring infrastructure (Python + FastAPI)

- Built a sync layer that diffed the internal client database against Uptime Kuma's monitor set and created, updated, or deleted monitors to match - each one provisioned with the right proxy, notification channels, and checks per client, replacing manual monitor upkeep.

Also maintained a legacy Laravel (PHP) tool, migrated the operator dashboard from server-rendered HTML to a Vite + React SPA, dockerised services, and ran Bitbucket Pipelines CI/CD across the monorepo.

Software Developer (IT), Volgograd State University of Architecture and Civil Engineering

Aug 2020 - Jan 2024
Volgograd, Russia

- Maintained the university's remote-learning platform used by students and staff.
- Built internal tools to streamline administrative workflows.
- Supported the online admissions campaign infrastructure over multiple admission cycles.

SELECTED PROJECTS

watchlight GO · WIP github.com/vdzhagaev/watchlight

Solo-built uptime-monitoring service. Split the scheduler into three channels with distinct backpressure semantics (bounded job/result buffers, unbuffered signaling) instead of one shared queue, sized a fixed worker pool, and serialized all SQLite writes through a single consumer to respect its one-writer-even-in-WAL constraint. Graceful shutdown drains in-flight work through a staged cancellation sequence before a hard timeout escalates it. Migrations, metrics, and load testing in progress.

pokeroll TAURI · RUST · BETA github.com/vdzhagaev/pokeroll

Cross-platform desktop widget (Tauri 2, SvelteKit + Rust) that rolls a random Pokémon from PokéAPI and renders it as a card with persisted local progress. Shipped as a public beta with builds for Windows/macOS/Linux - most complete shipped project so far.

EDUCATION

M.Sc. - Information Systems & Technologies (09.04.02)
Volgograd State University of Architecture and Civil Engineering · 2023-2026

B.Sc. - Information Systems & Technologies (09.03.02)
Volgograd State University of Architecture and Civil Engineering · 2019-2023

LANGUAGES

Russian	Native
English	A2 · Elementary