OLEKSANDR TUREVSKIY

Backend & Platform Architect — Go, Kubernetes & Applied AI

Viersen / Düsseldorf region, Germany · German citizen | me@turevskiy.com | +49 152 58596581 | linkedin.com/in/alexturevski | github.com/oltur

> 25 years in software, as a developer, team lead, CTO and architect, and I still write code. I stay responsible for what I build, including when it goes wrong. Today that means an Ansible-driven edge fleet-management platform and its simulation stack for a supervised-autonomy programme at Mira, and — in parallel, freelance — backend, front ends and AI document intake for a lending platform. Both are Go on Kubernetes. Outside work I ship complete products — three of them live on both app stores.
> **iSAQB CPSA-F · CKAD · AWS Solutions Architect & Developer Associate · HashiCorp Terraform Associate**

CORE SKILLS

Languages Go · Rust · TypeScript/JavaScript · Python · C# · SQL · Swift

AI / LLM Structured extraction against JSON schemas · prompt fallback for constrained-decoding failures · response caching · streaming · document and email intake pipelines · local inference (Ollama)

Infrastructure as Code Ansible — playbooks, roles and inventories driving fleet orchestration at scale · Terraform · Helm · GitOps

Platform Kubernetes · Docker · AWS (EKS) · Google Cloud · Microsoft Azure · gRPC · Prometheus/Grafana · CircleCI

Web & Mobile React · Next.js · Chakra UI · Tailwind · Sanity · SwiftUI · WidgetKit · ActivityKit · Capacitor · fastlane · Three.js / WebGL

Data PostgreSQL · MS SQL Server · MongoDB · Couchbase · Elasticsearch · BigQuery · dbt

EXPERIENCE

Mira GmbH (Rheinmetall Group), Düsseldorf — Senior Backend Developer

07.2024 – now

Supervised-autonomy fleet platform. Go services, Ansible automation, Kubernetes delivery to customer environments.

- Built and own the edge fleet-management platform — a web front end over **Ansible** that turns playbooks, roles and inventories into fleet operations: device registry and discovery, role-based access, maintenance windows, alerting, configuration drift, compliance reporting and parameterised health checks. Runs a **50-device production fleet, validated against 1,000+ simulated devices**. 484 of its 516 commits are mine.
- Own the **Helm** delivery layer — Helmwave-templated charts released to development, production and **four customer-specific environments, each with its own requirements**.
- Built the container and supply-chain security layer — Trivy and Grype against CycloneDX SBOMs, images reached over a read-only Docker socket proxy, upgrade advice ranked by remaining CVE risk — wired into the GitHub and GitLab pipelines alongside Dockerised integration tests, so **security, quality and compliance issues surface at build time rather than as customer-facing incidents**.
- Added an in-app AI assistant that routes free text to navigation, a device action or a question. Actions surface a confirmation step instead of executing directly, answers are grounded strictly in the product's own help content, and every figure is computed server-side from RBAC-scoped endpoints, never by the model.
- Own **common-go**, the shared library every service builds on: type-safe message envelopes using Go generics, OpenTelemetry logging, metrics and tracing, the RabbitMQ client. Moved fleet operations onto a signed-job agent protocol — Ed25519 envelopes with domain separation and replay protection.
- **Proposed, designed and wrote the fleet simulation platform** — my own initiative from the outset. It **proved the messaging, monitoring and control architecture holds at 1,000+ virtual devices**, twenty times the production fleet, and reproduces scenarios that are impractical to stage with real vehicles — crash conditions among them.
- Across the wider platform: an OEM fleet-telemetry API client for real-time vehicle data (sole author), and tag, storage and release-tag APIs with correlation-ID tracing in the Go core.

Twelvegrow — Senior Backend Engineer

03.2025 – now · freelance, parallel engagement · remote

Lending and underwriting platform. Go microservices on AWS EKS, React front ends, Terraform IaC, dbt warehouse.

- Backend across the core Go services — the domain and use-case layer of customer, deal, underwriting and normaliser, their datastores, migrations and the gRPC/protobuf contracts between them
- Cash-flow modelling, installment scheduling and marketing-performance views — the financial heart of the product — across both React front ends, roughly 600 commits between them
- Built the document and email intake pipeline and the Anthropic-backed extraction behind it: structured output against JSON schemas, a prompt-based fallback for when constrained decoding fails, caching, streaming
- **1,100+ commits across 27 of the platform's services** since March 2025 — including BigQuery and dbt models over banking data, and Terraform for the EKS development and production environments

Galeria Karstadt Kaufhof GmbH, Essen — Senior Backend Developer

08.2023 – 07.2024

- Took ownership of the web-shop codebase as the platform was brought in-house from an external vendor
- Stood up the company's own production infrastructure with the DevOps team and established the development processes for the in-housed stack

Nylas Inc. — Senior Backend Developer

05.2022 – 02.2023 · remote

- Built services for synchronising customer communication data; supported the Kubernetes deployment with Docker and Terraform
- Authored solution architecture documents and migrated data from the preceding systems

JoinClap — Backend Architect

01.2021 – 05.2022 · remote

- Designed the load-tolerant microservices architecture for a high-load social platform and built the backend in Go with Couchbase, deployed on GCP/GKE
- Ran performance testing and optimisation under load; contributed to product design

Earlier. ControlExpert, Langenfeld — *Senior Developer*, 01.2020–10.2021: built accident damage assessment software for car insurers from scratch on a current Microsoft stack, with the CI and test automation. **VMRay**, Bochum — *Senior Developer*, 04.2018–12.2019: backend and frontend of a cloud malware analysis and phishing detection platform, in Python and JavaScript. **MySupermarket**, UK/Israel — *Senior Developer*, 08.2016–04.2018: full-stack feature work on mysupermarket.co.uk, the Angular admin panel for site operations and the end-to-end test suite. **SeeYourTravel.com**, Ukraine/Israel — *CTO*, 01.2015–01.2017: conceived a connected travel community product, designed the UX, built the key web and mobile components. **Itera Consulting**, Kyiv — *Technical Lead / Engagement Manager*, 02.2014–02.2016: won and delivered projects, led core component design, mentored engineers. **Infopulse**, Kyiv & Germany — *Onsite Consultant / Development Lead, Bosch Security Systems*, 08.2007–02.2014: led joint onsite and offshore development of video surveillance software.

OWN PROJECTS

Where I get to build the whole thing, end to end. This is the work I enjoy most.

The 5001 series — three apps, each shipped on both the App Store and Google Play.
5001 Words — SwiftUI vocabulary trainer for iPhone, iPad and Apple Watch plus a native Android build; 290+ decks across 84 languages, neural-voice pronunciation, downloadable packs with a range-fetch so the word list is usable seconds into a download. **5001 Tomatoes** — SwiftUI interval timer with Live Activities and widgets, 76+ locales, data never leaving the device. **5001 Axolotls** — 3D multiplayer game in Three.js/WebGL; Node server and browser client import the *same* rendering-free simulation module, so multiplay is server-authoritative with no second implementation to keep in sync, and identity is a SHA-256 proof-of-work with no email, no password and no free text anywhere — therefore no moderation surface.
apps.apple.com/app/id6784747555 · play.google.com/store/apps/details?id=com.turevskiy.a5001word · apps.apple.com/app/id6813990177 · play.google.com/store/apps/details?id=com.turevskiy.axolotls5001 · apps.apple.com/app/id6753728553 · play.google.com/store/apps/details?id=com.turevskiy.fiveminutes

SeeYourTravel — the travel-community product I built as CTO in 2015 and never let go of. Brought back in 2026: containerised behind a Cloudflare Tunnel with no inbound ports open, deployment driven from a Makefile. A ten-year-old codebase kept alive without a rewrite. seeyourtravel.app

LLMChatBot (Go, Ollama) — retrieval chatbot over any website on fully local inference. **Plast Düsseldorf** (Next.js 15, React 19, Sanity) — trilingual site for a Ukrainian scouting organisation. **just-another-vpn-server** (Rust) — an OpenVPN-protocol server written from the wire format up that a stock client connects to unmodified.

CERTIFICATIONS

iSAQB® Certified Professional for Software Architecture, Foundation Level (CPSA-F) — Nov 2025 · **CKAD: Certified Kubernetes Application Developer** — Jun 2024

AWS Certified Developer – Associate — Oct 2023 · **AWS Certified Solutions Architect – Associate** — May 2023 · **HashiCorp Terraform Associate** — Mar 2023

Earlier: PMP (2013) · ISTQB CTFL (2012) · Certified Scrum Master (2008) · MCSD (2001) · credly.com/users/alex-turevski/badges

EDUCATION

MS, Technology Management — Lviv Business School, Ukraine, 2014–2017 · **MS, Computer Science** — Kharkiv National University of Radioelectronics, 1993–1998

LANGUAGES

English (fluent) · German (advanced) · Ukrainian (native) · Spanish, Hebrew (beginner) · German citizen

Updated November 2026