Okan Can Özbek

Software Engineer | C++ and Distributed Systems

Amsterdam, Netherlands | +31 6 1441 2521 | devokan@outlook.com | linkedin.com/in/oo-dev | github.com/okan-ozbek

Professional Summary

Software engineer with **5+ years of backend experience**, including Rust gateway development, event-driven systems and C++23 networking projects. Delivered measured query-latency improvements, reusable infrastructure, and production reliability through testing, code review and end-to-end ownership.

Leadership and Entrepreneurship

Pixelware Jun 2024 - Mar 2026

Freelance Software Engineer | Amsterdam, Netherlands

Project-based client engagements, primarily part-time.

- Modernized performance-critical client systems by driving incremental rewrites from legacy PHP to **C++, Rust and TypeScript**, with architecture reviews focused on correctness and throughput.
- Improved maintainability of a property-investment platform by translating financial business rules into application logic, refactoring legacy code and implementing **unit and integration tests**.
- Improved delivery quality across client projects by implementing **unit/integration tests and CI/CD**, reviewing architectural changes, and mentoring engineers through code review and pair programming.

Cyber Dev Aug 2025 - Mar 2026

Technical Co-founder | Cybersecurity SaaS joint venture

- Established the platform's engineering foundations by defining the **TypeScript/Nuxt architecture**, unit/integration tests, CI/CD pipelines and a staged development-to-production release process.
- Enabled intern engineers to contribute through onboarding, pair programming and code reviews, with technical guidance and learning resources tailored to their development.

Professional Experience

Warner Bros. Discovery Mar 2026 - Present

Software Engineer | Amsterdam, Netherlands

- Reduced average search-query latency from **400-500 ms to 5-10 ms** by leading Redis hot-path caching on AWS ElastiCache, with explicit TTL and invalidation behavior.
- Contributed to a Rust gateway tested at approximately **70,000 requests/s** during extended load tests by implementing features, fixing memory leaks and patching vulnerable Cargo dependencies.
- Delivered the team's sitemap ingestion in a **six-team initiative** by implementing Kafka-to-DynamoDB processing and S3 generation, including tombstone handling for source deletions.
- Enabled request-level diagnosis across Kafka, Redis and Apollo GraphQL by maintaining shared Datadog tracing, logging and metrics, supported by unit and integration tests.

FANSTR Aug 2024 - Aug 2025

Software Engineer | Rijswijk, Netherlands

- Reduced failed external transactions by approximately **25-30%** by redesigning API import flows around RabbitMQ, idempotency, deduplication and bounded retries.
- Enabled independent retries and traffic-based worker scaling for email/SMS delivery by separating delivery from request processing and adding backpressure, failure isolation and observability.
- Improved **p95/p99 latency** while preserving write correctness by applying caching, concurrent processing, SQL hot-path indexes, transactions and row-level locking.

TJIP The Platform Engineer Mar 2023 - Aug 2024

Software Engineer | Delft, Netherlands

- Made data changes auditable by leading **event-sourced workflows in Go** that record updates as events and reuse state-transition patterns across platform features.
- Improved testability and developer onboarding by modularizing legacy PHP and supporting incremental migration of performance-critical paths to Go.

Alflex Technologies Apr 2021 - Mar 2023

Software Engineer | Zoetermeer, Netherlands

- Exposed device measurements to customers by building **TCP/IP communication** with proprietary C/Objective-C firmware, parsing and validating binary payloads in PHP, and persisting telemetry in SQL Server.

Projects

Radish | *C++23, Asio, CMake, Catch2*

- Enabled binary-safe key-value operations over TCP by building an **Asio server with RESP parsing and pipelining**; implemented TTL, framed append-only logging, restart replay and explicit compaction.
- Made correctness and performance experiments repeatable through **unit, loopback integration and recovery tests**, plus a configurable benchmark runner reporting throughput, errors and latency distributions to JSON/CSV.

Data Structures | *C++23, CMake, GoogleTest*

- Built **eight containers and ownership types** from first principles, including a vector and smart pointers, with explicit allocation, object lifetimes, move semantics and GoogleTest coverage.

Technical Skills

Languages: C++23, Rust, TypeScript/Node.js, Go, Java, PHP, SQL.

Systems: TCP/IP, binary protocols, data structures, concurrency, Linux, distributed and event-driven systems.

Testing and delivery: Catch2, GoogleTest, CMake, unit/integration testing, CI/CD, GitHub Actions, code review.

Data and operations: Redis, Kafka, RabbitMQ, DynamoDB, MySQL, SQL Server, AWS, Docker, Datadog.

Education

Breda University of Applied Sciences 2020

Coursework in Game Engine & Architecture Development; degree not completed.