

Nikita Doronin

📍 Vienna, Austria | 📞 +43 677 63973045 | ✉ nikdorkir@gmail.com | 🌐 nikdor.xyz | 🐙 github.com/duckysmacky

Education

University of Vienna

BSc in Computer Science, Software Engineering

Vienna, Austria

Sep. 2026 – 2030 (expected)

Moscow Polytechnic University

BSc in Information Technology, Software Engineering

Moscow, Russia

Sep. 2025 – Jul. 2026 (transferred)

Experience

Junior Firmware Engineer (C / UEFI)

Mar. 2026 – Aug. 2026

Aquarius

Moscow, Russia

- Developed and maintained UEFI/BIOS firmware in C on AMI Aptio 5 and EDK2 across four platform trees for dual-socket Intel servers (Ice Lake-SP, Raptor Lake-S; 50+ commits); received a full-time return offer.
- Designed a topology-aware PCI bus redistribution policy in the PEI phase that enabled full enumeration of a 96-drive cascaded NVMe/JBOF array, preserving fail-safe fallback for unaffected IIO stacks.
- Built bounded PCIe endpoint recovery ahead of driver binding (link-state gating, identity/BAR retry, Secondary Bus Reset, invalid-device quarantine), and extended server inventory reporting via SMBIOS, IPMI/FRU and HII Setup.

Freelance Rust Developer

Jul. 2025 – Mar. 2026, Aug. 2026 – Present

Self-employed

Remote

- Delivered backend services, automation tooling and bots in Rust for international clients: REST and gRPC APIs (axum, tonic), Telegram and Discord bots, browser automation and trading systems on Tokio, backed by PostgreSQL, Redis and Docker.
- Owned the full delivery cycle: scoping loose requirements with clients, designing the architecture and data model, writing tests and deploying to hardened VPS infrastructure with GitHub Actions CI/CD.
- Built an async crypto trading bot automating Bybit P2P order processing and payment execution, including a fully reverse-engineered payment provider API (sessions, CSRF tokens, request signing).

Projects

Umbra VPN | Rust, Tokio, axum, tonic (gRPC), PostgreSQL, mTLS

2026 – Present

- Self-operated SaaS in production with a paying subscriber base: a Rust control plane orchestrates independently deployed VPN nodes over mTLS-secured gRPC, exposing an HTTP API (axum) and persisting state in PostgreSQL.
- Automated provisioning, subscriptions and usage tracking behind a Telegram bot gateway; run production infra and support solo.

Synco | Rust, futures, procedural macros

2026

- Asynchronous runtime built from scratch to explore what sits beneath Tokio: task queue, spawner, executor, custom wakers and delayed futures on the standard library alone, plus a procedural-macro entry-point crate.

Simple Rust Kernel | no_std Rust, x86-64, QEMU

2026

- Bare-metal x86-64 OS kernel written entirely in no_std Rust and booted under QEMU: kernel startup, boot process, hardware abstraction and low-level memory handling - with a custom test runner executing integration tests inside QEMU.

Qt Chat | C++20, Qt6, raw TCP sockets

2026

- Client-server chat platform built from zero on raw TCP: a custom wire protocol with typed packets, its own serialization layer and connection abstractions over the socket API instead of an off-the-shelf messaging library.
- Supports concurrent clients, encrypted transport and the full user lifecycle (registration, auth, sessions) behind a Qt6 GUI.

Cogniflex | Java, Spring Boot, gRPC, Python, React

2026

- Team-built platform detecting AI-generated text, images and video: a Spring Boot API coordinates a Python inference service over a shared Protocol Buffers/gRPC contract, serving a React site and browser extension.
- Pairs learned classifiers with a custom Java rule engine whose rules emit weighted evidence combined into a probabilistic score.

Also: Win-Tools - Win32 UNIX-utility reimplementations (C++/C, 2024); CSPC - smart pointers and memory management tools for C (2026); Copper - C/C++ build system (Rust, 2025); Databoxer - encryption CLI with custom file format and encoding (Rust, 2025)

Technical Skills

Programming Languages: Rust, C, C++, Java, x86-64 ASM, SQL

Systems & Low-level: UEFI / EDK2, bare-metal no_std, PCIe, SMBIOS, IPMI/FRU, Win32 API, libc, RAIL, templates, memory management, concurrency, async internals, FFI

Frameworks & Tools: Tokio, axum, tonic, serde, sqlx, Qt6, Spring Boot, Linux (Arch), Git, Docker, Cargo, CMake/Make, QEMU, GDB, PostgreSQL, GitHub Actions

AI & Agentic Tooling: Claude Code, Codex, OpenClaw, MCP servers, published agent skills, agentic workflows

Certifications & Achievements

- **1st place (2025) and 2nd place (2024)** – “Professionals” national championship (all-Russian final, IoT track); **Prize winner (2025)** – All-Russian Olympiad in Cybersecurity
- **Perfect score (100/100)** - Russian Unified State Exam in Computer Science, 2025
- **Codecademy:** Learn C++, Learn Java, Learn Intermediate Java, Learn Java Spring, Introduction to Cybersecurity (2022-2024)
- **Spoken languages:** Russian (native), English (C1), German (B1-B2)