Eugene Ash

Kivooeo123@gmail.com | github.com/Kivooeo

Professional Summary

Systems and compiler engineer on the Rust Compiler Team, with 400+ contributions over the past year — reviewing, stabilizing language features, fixing ICEs, improving diagnostics, and reorganizing core test infrastructure. Google Summer of Code 2026 mentor for the Rust Project. Looking for a remote systems or compiler engineering role.

Experience

Core contributor, Rust Compiler Team — Apr 2025 — Present

- **Current focus** – mentoring & review: Selected as a Google Summer of Code 2026 mentor for the Rust Project, guiding a contributor through a compiler project end to end (design, code review, rustc workflow). Day to day, review PRs from new contributors and guide them through the contribution process.
- **Stabilized if let guard feature** – implemented the final changes, wrote comprehensive **tests** and coordinated the **documentation** update.
- **Fixed union edge case** – A low-level memory reference edge case in union fields; **also wrote a follow-up fix to rust-analyzer**.
- Led a major test-suite reorganization initiative; personally migrated and refactored 500+ tests within two weeks, completing the first key milestone in the tracking issue.

Additionally, 150+ merged PRs improving diagnostics, fixing internal compiler errors (ICE) and test infrastructure.

Software Engineer — Mar 2026 — Present

T-Bank — *Remote / Moscow, Russia*

- Brought CO-RE support to our eBPF security agent (Rust), so the same probe runs across kernel versions without per-kernel rebuilds — cut deployment friction across our fleet's mixed kernels.
- Built the CO-RE pipeline as two layers: a C shim exposing BTF-based field accessors, wrapped in safe Rust generated by custom macros — kept the unsafe surface small and auditable.
- Added ancestor-aware scope filtering to the rule engine: policies can now match on a process, its parent, and the full ancestor chain — closed a gap where threats slipped through by spawning from trusted binaries.
- Owned the eBPF/userspace boundary end to end: verifier-constrained kernel code, map design, and rule serialization.

Interests

Systems thinking across domains — from compiler internals and electrodynamics to biological and chemical systems. Off-screen: drawing, strategy games, *stargazing*.

Education

Surgut Polytechnic College — Surgut, Russia

B.Sc. in Instrumentation & Control Engineering — *Sep, 2021 — Jun, 2025*

- Programmed and configured microcontrollers and PLCs for industrial I&C systems — hands-on work with registers, timers, interrupts, and hardware peripherals at the bare-metal level.
- Foundation in real-time constraints, signal processing, and deterministic embedded behavior — the same systems-level thinking I now apply to compiler internals.

Key Skills

- **Languages**: Rust (expert), C, Python
- **Concepts**: Compiler optimizations, diagnostics, AST/IR, memory safety
- **Systems & Low-level**: FFI, unsafe Rust & safe abstractions, memory layout, proc-macros, eBPF/BTF
- **Workflow**: Git, jj, CI systems, test automation, Cargo
- **Soft Skills**: Technical mentoring, self-directed learning
- **OS**: Linux/Unix