Mikhail Razakov

Software Engineer — Systems & Architecture

razakov.mikhail@outlook.com
t.me/Micodiy
github.com/Misha1302
misha1302.github.io/CV

I design and build software systems, from compilers and program-analysis tools to backend services and offline-first applications. I work on symbolic computation in SharpChecker at ISP RAS and previously implemented LLVM optimization at MCST. In independent projects, I own architecture, implementation, and correctness checks.

Professional experience
ISP RAS: SharpChecker and symbolic computation. MCST: LLVM 22/C++ and optimization validation.

Independent engineering
UniversalToolchain compiler/runtime; botay! web/Android offline sync; VpnMediator payment-state recovery.

External results
Talk delivered at LangDev on 9 Oct 2026; top MCST technical-selection result (organizer correspondence).

EXPERIENCE

October 2026 — present

ISP RAS — SharpChecker development

- Work on symbolic computation in SharpChecker, a C#/.NET static-analysis tool.

July — August 2026

MCST — Compiler Engineering Intern

- Implemented a LICM pass for LLVM 22 in C++; hoisting conditions check loop structure, memory access, side effects, and speculative safety.
- Built a Python harness comparing program behavior before/after optimization and checking transform/no-transform cases.

PROJECTS

UniversalToolchain / Wist2

Resolved component dependencies, conflicts, and backend routes into an immutable LanguagePlan; runtime materializes the checked selection without replanning.

Frontend, IR, optimization passes, interpreter and CIL backend; independent configuration checks and 1,324 tests at the 9 Oct 2026 revision.

botay! / student-execution-os

Built a web/Android planner separating external observations and user state from derived plans; feasible schedules require a validated witness.

Offline operations replay idempotently; the signed Android 0.8.3 release workflow passed on 10 Oct 2026.

VpnMediator

Built backend logic for payments and subscriptions, preserving provider evidence independently of access changes.

Idempotent retries, reconciliation and manual review for conflicts, with backup/restore and rollback safeguards; no scale claim.

SKILLS

Core languages
C++23, C#/.NET, Python; also Rust, Java, JavaScript, SQL

Software / systems
modular architecture, compiler tooling, APIs, Linux, SQLite/PostgreSQL

Product / backend
offline synchronization, Android, payments, state recovery, migrations

Correctness
regression and differential tests, LLVM Verifier, invariants, idempotency

EDUCATION

HSE University — Software Engineering

RECOGNITION

MCST LLVM/C++ selection
Organizer correspondence: only 5.0/5.0 score and only solution passing all 104/104 tests among 49 submissions.

LangDev'26 — Build the Language, Then Make the Abstractions Disappear
Delivered the UniversalToolchain architecture talk at LangDev on 9 October 2026.

Moscow / remote