Andrey Vlasov

avlasov@cabal.run | Georgia, Batumi

TL;DR

- **Haskell**, **PureScript**, **Rust**, **TypeScript**, **Nix**, **Python**.
- Backend, DevOps, Blockchain, a bit of Frontend.
- English: C1, Russian and Ukrainian: Native.
- I love strong type systems and reproducible builds.

Experience Timeline

Autumn 2025: `C&`

Another experiment with languages, a thin transpiler targeting C.

Summer–Autumn, 2025: Freelance

- Backend in **Rust**. Mostly communicating with APIs.
- Some data analysis of crypto exchanges in **Haskell**.

April, 2025: `Ludum Dare 57`

- Sound effects.
- Minor programming in Unity.

June, 2024 – November, 2024: `Gauss`

In *Gauss* I was building backend in **Haskell**, particularly, integrations with OpenAI and Intercom, Plaid and MethodFI to lesser extent. Worked with Servant, Esqueleto/Persistent/PostgreSQL.

February, 2024 – current time: `https://codeberg.org/aka_dude/lux`

lux is my self-educational project in compiler building. Please see an interactive tutorial `here`.

December, 2023 – February, 2024: `https://scrabble.cabal.run`

`Frontend` in **PureScript**/Halogen for the Scrabble game.

June, 2023 – current time: `Osmos`

- **DevOps/SysAdmin**:
  - Github Actions for automatic testing and deployment.
  - Production and QA servers (**NixOS** at AWS).
  - bot for alerts (checks network connection and HTTP status) in **Haskell**.
  - `Nag CI`, a custom CI in **Haskell** for Github-compatible forges. Makes NixOS deployments much quicker then deployments via Github Actions.
- Substitute **Python**/FastAPI/SQLAlchemy backender.

Spring, 2022 – February, 2023: `MLabs`

- Cardano-Wanchain Bridge
  For CWB we've been writing off-chain code in **PureScript** with Cardano Transaction Library and on-chain code in **Haskell** with PlutusTx.
- DevOps
  As of February, 2023 MLabs uses HerculesCI. I've been writing modules in **Nix** for Hercules.

2018 – 2022: Higher School of Economics, Moscow

- I strongly disliked LabView (a tool for visual programming of sensors), so I found the sensors' firmware and wrote a **Python** library for communicating with sensors. If I needed to do the same thing today, I would write it in Rust.
- Embedded programming with Arduino, C++ and **Python**. I made a demo for a wireless measuring device. It was a box with a gyroscope and a computer program that showed the box's accelerations real-time[1]. There was a plan on making more of such devices, but I guess no one really needed them, so that didn't happen.
- A numerical calculation for `Real Science`.
- A small 2D simulation of bouncy particles on GPU with OpenCL and **Rust**. The code is lost, unfortunately.
- A `Zulip` bot in **TypeScript/NodeJS** for monitoring university's infrastructure.
- A **Gitlab CI/CD** configuration.
- During mandatory internship I wrote backend in **TypeScript/NodeJS**/`light-arrow`.
- For most of my time in the university, I've been studying **Rust** for backend:
  - I've worked on the `teloxide` crate during its early days.
  - I've created a `typed wrapper` for `sled`.
  - I've made a telegram bot for notifying myself of events on my PC: `notify-tg`.
- Together with a professor and another student, we've created a thunderstorm warning system for high-rise workers. I've written two parts of this system in **Rust**: a backend that polls sensors, and a Windows program that communicates with the backend and shows a notification before a thunderstorm.

[1] Demonstration: `/apparatus_demo.mp4`