Daniel Sobol

Email: sobol.daniil@gmail.com[1]
Site: mynameisdaniil.org[2]
Telegram: t.me/mynameisdaniil

TL;DR:

Software Engineer with an Electronics Engineering background. Experienced in designing complex systems, software and network architecture, low level network programming, distributed systems. My ex-colleagues called me "The best Scrum master in their entire life".

My knowledge and experience allows me to build computer from basic components (including PCB design), create an operating system for it and build distributed application with SLA in mind on top of it; all the while hiring people and setting up processes.

Education:

- 2010-2012 **Master's degree in Radio engineering** Southern Federal University (Taganrog, Russia)
- 2006-2010 **Bachelor's degree in Radio engineering** Southern Federal University (Taganrog, Russia)

Work experience:

- 2024-now Working on my own project: eportald.org
- 2023-2024 Status. Was working on Fluffy – Ethereum portal network client.
- 2022-2023 Revea — custom cosmetics company. Elixir, Ecto, Phoenix, terraform. Plus hardware design and prototyping, Rust & C++.
- 2021-2022 Netronix Group. Under NDA. Elixir, Ecto, Phoenix, k8s, AWS, terraform.
- 2020-2021 AEON — still under NDA. Basically Elixir/Erlang programming, little C++ here and there, little bit DevOping; Scrum-master.
- 2019-2020 Curogram — Was helping out my friends with growing the team and also doing some TypeScript/NodeJS.
- 2017-2019 United Traders — Developed stock exchange and surrounding services in Erlang. And then participated in stock exchange development in Rust. It was a really good case for microservices and event sourcing. Team lead.
- 2012-2017 Lodoss Team — Returned after my graduation and started a long-lasting project. Designed and developed a lot of things from ground up. Starting from general software architecture all the way to physical network architecture. Distributed containers supervisor based on LXC (like Kubernetes, but before it was created), continuous delivery, configuration management in Ansible (back when they hosted their site on 3rd level domain). RPC, microservices, custom high-throughput message exchange, graph database based on Google's Pregel whitepaper [3].
- 2010-2011 Lodoss Team — My first official job. NodeJS, Riak.
- 2006-2010 Freelance — Created various websites for local businesses using PHP/HTML/Flash/Flex.

Toolbelt:

- **Elixir/Erlang** — weapon of choice.
- **Rust** — new weapon of choice for low level, performance-critical and embedded applications.
- **C** — old weapon of choice for low level, performance-critical and embedded applications.
- **Haskell** — for competitive programming, teaching and domain-specific tasks.
- **C++**[4] — Let's just say I've never failed to complete a task in C++.
- **NodeJS** — I think I've done my share of NodeJS for this lifetime.
- **PHP** — Yep, I've done my share of it too.

Interests:

- **Low level network programming.** When in doubt launch Wireshark. I also like reverse-engineering, and code optimisation.
- **DSL, computer language design and linguistics in general.** On each and every new project since 2012 I've been dreaming about language to describe and verify complex systems. This is going to be the basis of my PhD thesis when I will finally come around.
- **Embedded programming.** Since I'm an electronics engineer by training this is another thing I really love to do.
- **Team management and processes optimisation** Working with United Traders changed my perspective on team management. Since then, I'm a believer in that you can build good and reliable processes and still be friendly and unobtrusive.
- **Speaking at conferences** I have knowledge to share, stories to tell and I love doing it.
- **Learning new things** That's what I've been doing my whole life, almost every day of it.

[1] Not a misspell. This is my actual email since before I really cared about how my name pronounced in English-speaking countries.
[2] See [1]
[3] See: https://www.cs.cmu.edu/~pavlo/courses/fall2013/static/papers/p135-malewicz.pdf
[4] I prefer to avoid C++ where I can. May Stroustrup be my witness — there is no single human who can completely understand the whole of modern C++.