

Functora

Hi-end quality software development.
Reliable. Functional. Pure.

About

I am pseudonymous software developer Functora. I do specialize in a software development with [Haskell](#), [Rust](#) and [Nix](#) languages since 2019. I work with passion and create the best software using the best technologies. My areas of expertise and interest are [Bitcoin](#), [Lightning Network](#), sound money, trading, free markets, privacy and sovereignty. My primary technical goals are type-level guarantees of software correctness and safety.

Skills

Haskell is my first programming language. I do have advanced level of Haskell including:

- [GHC.Generics](#), [generic-lens](#) and [SYB](#) for efficient data manipulation.
- [GHCJS](#) compiler, GHC [WASM](#) backend and [JSaddle](#) EDSL.
- [Persistent](#) and [Esqueleto](#) typed SQL libraries.
- [Yesod](#) enterprise web framework.
- [Miso](#) frontend web framework.

Rust is my second programming language, but nowadays I often prefer it over Haskell due to its superior performance and better development experience. I do have good level of Rust including:

- [Chumsky](#) parser combinators.
- [Diesel](#) typed SQL library.
- [Dioxus](#) cross-platform apps framework.

I am also qualified in other useful areas:

- [Nix](#) and [NixOS](#) - Strong medium level. Nix is the best tool for deterministic builds, tests and development environments. I am using NixOS daily.
- [Bitcoin](#) and [Lightning Network](#) - Advanced level as application developer (not as protocol developer). Bitcoin is the greatest achievement of the new millennium, which brings financial freedom and sovereignty back to the people.
- [PostgreSQL](#) and [SQLite](#) - Strong medium level. I am not database expert, but I am using everything what software developer should use to manipulate data storage. Queries, transactions, joins, locks.
- [Docker](#) and [Swarm](#) - Advanced close to expert level. I am using Docker and Swarm for development, builds and production. Docker is a very handy tool for MacOS and Linux compatibility.

Code

Examples of my personal Haskell code:

- [currency-converter](#) - An app for converting currencies, generating financial documents, and sharing them via links or QR codes. It includes optional client-side encryption. Built using Miso and GHCJS, the source code is available on [github](#). Mobile version is also available on [Google Play](#).

- [lightning-verifier](#) - An app for offline verification of [Lightning Network](#) invoices and preimages, and sharing them via links or QR codes. It includes optional client-side encryption. Built using Miso and GHCJS, the source code is available on [github](#).
- [delivery-calculator](#) - A simple app to estimate delivery costs, generate orders in Excel spreadsheet format, and share them with merchants. Built using Miso and GHC WASM backend, the source code is available on [github](#).
- [functora](#) - My own collection of various general-purpose libraries, most of which work with both GHC and GHCJS.
- [lnd-client](#) - Lightning Network Daemon (LND) client library for Haskell.
- [bfex](#) - Bitfinex cryptocurrency exchange client library for Haskell.
- [miso-functora](#) - Reusable Miso widgets, composable through optics.
- [rentier](#) - My first Haskell project which I have used to learn Haskell. The booking system is based on the Yesod web framework. The code is very obsolete.

Examples of my personal Rust code:

- [cryptonote](#) - A serverless, cross-platform app that allows users to create encrypted or plain-text notes directly on their device or browser. It uses strong authenticated encryption (AES-GCM or ChaCha20-Poly1305) with HKDF key derivation, embeds content in shareable URLs or QR codes, and ensures no data leaves the device unless explicitly shared. The source code is available on [github](#).
- [functora-tagged](#) - Lightweight, macro-free newtypes with refinement and derived traits.
- [functora-cfg](#) - A Rust library that merges configuration values from multiple sources into a single typed value.
- [functora](#) - Missing pieces of the Rust standard library.

Examples of other code I was actively working on with other people:

- [btc-lsp](#) - Bitcoin Lightning Service Provider.

Blog

- [Bitcoin Seed Security](#)
September 18, 2025
- [Offline Lightning Wallet](#)
October 23, 2020
- [Sovereign Web Introduction](#)
October 15, 2020

Contact

- functora@proton.me
Email
- [@2lit:matrix.org](https://matrix.to/#/@2lit:matrix.org)
Matrix