

Yura Dupyn

yura@dupyn.com | dupyn.com | [GitHub](#) | [Forgejo](#) | EU Citizen | Remote (CET/GMT+1)

Mathematically trained Software Engineer with over a decade of programming experience. I view Informatics theory and practical software engineering as fundamentally intertwined:

- I have practical enterprise experience developing in large-scale codebases
- When thrown into the middle of a project, I try to make progress even when I don't understand everything up-front, slowly building a mental-model in the background for the larger system at play
- My workflow is to first design and implement the most direct solution that works. Afterwards progressively clean-up, generalize, and draw abstraction-boundaries that are motivated by the direct solution
- I am most creative when working against real-world constraints and not being in a vacuum
- I take Type Theory as a guide, whether I am dealing with pure, immutable, functional systems or stateful, always-changing, imperative systems
- Love to help people understand things I understand 1-on-1, considering where they are at, figuring out the underlying assumptions. I don't like lecturing!

Experience

Independent Developer

Jan 2026 - Present

Implementing live, programmable knowledge environment (inspired by TiddlyWiki/Smalltalk) and setting up a self-hosted home-lab/web-server:

- Implemented a push-based Signals runtime in TypeScript, using Directed Acyclic Graph (DAG) with topological sorting for glitch-free value propagation
- Created a boundary between reactive-signals and a custom, purely-functional language
- Implemented transactional hot-swapping at runtime of live signal nodes with cycle-detection and automated rollback
- Deployed a self-hosted homelab on bare-metal NixOS, manually configuring Nginx reverse proxies, Wire-Guard VPNs, automated DNS/domain routing, and a personal Forgejo Git instance
- Visually designed and built my personal webpage to force myself to think more like a designer. I gathered as many constraints as possible, narrowing the design space by separating data from layout, and building a consistent design system (especially thinking systematically about color via OKLCH).

Co-Founder | Hammurapi (AI Legal Tech)

Mar 2025 - Jan 2026

Replaced basic semantic-search tool with an LLM-based legal-assistant:

- Built a TypeScript backend that uses the Model Context Protocol (MCP) to expose the legal database of Slovak legislation and external APIs as tools for the model
- Created the frontend chat interface with project and session management (TypeScript/React/Node)
- Implemented Server-Sent Events (SSE) to handle streaming and manage state with multiple LLM providers in a unified API.

Master's Student (Systems & PL) | Masaryk University, Faculty of Informatics

Sep 2023 - Feb 2025

Used the program to impose constraints so that I could learn Rust, basic embedded/FPGA, basic AI/ML, and improve/explore my language design/implementation skills:

- Rust DB server (team): Implemented the SQL interpreter and a persistence engine with custom binary format for tables and a simple GC to manage deletions
- Rust MLP (solo): Built a deep-learning MLP trained on fashion-MNIST. Created my own abstract backpropagation formulation based on composition of smooth maps between Riemannian manifolds and adjoints of differentials. Implemented all the Linear Algebra on CPU without using external libraries
- Polymede Language/Compiler (Rust, WASM, solo)

- Master's Thesis Research (solo): Designed and implemented a hybrid/multiple-fragment language and interpreter in Rust in an attempt to unify Intuitionistic Linear Logic with standard functional programming.

After 3 semesters, paused studies to co-found AI legal-tech startup.

Full-Stack Engineer | Itravel (Remote, Germany)

May 2022 - Aug 2023

Developed and maintained features across a distributed travel-tech platform, from Scala/PostgreSQL/GraphQL backend to Elm/TS Frontend:

- Onboarded senior backend developers into the Elm frontend
- Developed end-to-end features - from PostgreSQL schema design to responsive Elm UI, including the implementation of i18n/localization pipeline, and analytics
- Delivered backend features in Scala in a distributed architecture, managing concurrency requirements and distributed caching mechanisms
- Reduced payload size on the frontend. Reverse-engineered an undocumented third-party API to replace a bloated React embed with a lightweight JS/Elm, and translated UX designs into optimized, responsive frontend components (like lazy-loading image carousel in Elm).

Frontend Developer | GWI (Remote, UK)

Dec 2020 - Mar 2022

- Quickly onboarded into a complex ~100K line Elm enterprise codebase even though I had no previous Elm experience, implementing frontend features and data-management UI components based on high-level architectural specifications
- Genuinely enjoyed guiding colleagues who were new to functional-programming, introducing them to Elm-architecture, state-management, and FP-philosophy to help them contribute to the codebase
- Proactively built custom Elm parsers (including a CSV parser and a dataset query DSL) to overcome the limitations of existing libraries, focusing on clear error messages for end-users
- Maintained strict technical standards during code review (often 3,000+ lines). Argued against anti-patterns, actively engaged in constructive architectural debates to ensure highly performant, maintainable codebase.

Projects

For more projects and demos see my personal webpage.

Polymede Language Design, Compiler & Runtime

2024

| Tech: Rust, WebAssembly, JS

Designed and implemented a purely-functional, statically-typed language and zero-dependency WASM compiler from scratch in 6 weeks:

- Designed the frontend and intermediate representation, with Parametric-Polymorphism, Bidirectional type-checking, Algebraic-Data-Types, Pattern Matching, IO-monad, and a custom Graph Memory Machine (GMM) IR
- Implemented a custom WASM bytecode serializer and a hand-rolled copying 2-space Garbage Collector (Cheney's algorithm) in ~1,000 lines of raw .wat
- Limited language scope and made trade-offs/simplifications under a tight deadline (e.g. ditched object-system for closures, made tail-calls work well instead of having specialized looping constructs)
- Managed runtime implementation complexities by writing high-level pseudocode first, then progressively refining it into raw WASM, surviving WASM's inability to distinguish integers from pointers.

Actor-Model Stack Machine

2022

| Tech: Elm

Created a runtime simulator to showcase incremental execution of Control Flow, Lexical Closures, first-class Runtime Stacks, Delimited Continuations (call/cc), and message-passing Concurrency.

Newtonian Dance (N-Body Physics)

2026

| Tech: TypeScript, SolidJS, Canvas

Created a generic n-body physics simulation in the browser that uses Data-Oriented Design and Structure-of-Arrays memory layouts.

Education

Bc. in Mathematics | Masaryk University, Faculty of Science

2019

Thesis: Algebraic Structures (Categories, Algebraic Theories, Monads)

Keywords/Skills

- Languages: Rust, TypeScript / JavaScript, Elm, Scala, C, WASM, SQL, Ruby / Python
- Frontend & UI/UX: Modern CSS / SCSS, OKLCH, React, SolidJS, Responsive UI/UX
- Backend & Data: PostgreSQL, node, GraphQL, REST
- Systems & Infrastructure: Linux / NixOS, Docker, GCP / AWS, Nginx, WireGuard, CI/CD, Neovim / Tmux
- Theory & Architecture: Functional Programming (Haskell, Scheme), Type Theory / PL Theory, Proof Assistants / Dependent Types (Lean, Rocq), Category Theory, Linear Logic