

Damien Meur



Senior Freelance Software Engineer

Full stack · TypeScript · React · Node.js · AI

Brussels, Belgium · +32 475 39 55 16 · damien.meur@tmonier.com

tmonier.com · github.com/get-tmonier · linkedin.com/in/damien-meur

PROFILE

I build web products in TypeScript, from the database to the React UI. Eight years across product, backend and data teams; more recently, LLM features. I care about the screen and the failure paths behind it. Freelance through Tmonier, remotely or in Brussels, including long engagements.

MAIN TOOLS

Product TypeScript, React, Next.js, Node.js, Effect, Fastify, PostgreSQL

Data & platform Python, SQL, DuckDB, Azure, Databricks, Docker, CI/CD

AI features OpenAI SDK, structured outputs, tool calls

Tmonier SRL		May 2026 – Present
Founder · Senior Freelance Software Engineer		Brussels / Remote
Independent practice · Client engagements below		
US HR-tech SaaS		Remote
Senior Software Engineer		May 2026 – Present
• Cut latency by 50–70% on the heaviest data views: analysed SQL plans, replaced N+1 and correlated queries, and halved backend startup work. • Restructured a roughly 150k-line monorepo into enforced layers; added plugins and a Postgres-backed job queue for stateless application instances. • Built integration tests across UI, HTTP, backend and real Postgres, with CI checks that catch tests which assert no useful behaviour. • Worked on scoped, time-bound authorization, slow-request tracing and release startup checks. • Prototyped an LLM analytics assistant: typed query specifications from the model, figures computed by the application from business data.		
TypeScript · Effect · React · PostgreSQL · GraphQL · OpenAI SDK · Playwright		
US product technology company		Remote
Product Engineer		Aug 2026 – Present
• Build authenticated product pages and shared React UI, including page headers and download entry points behind feature flags. • Work on product variants, attributes and creation flows across shared backend models and product-specific rules. • Implement and review changes with a small remote team, in focused pull requests.		
TypeScript · React · Node.js		

PREVIOUS EMPLOYMENT

Aware

Brussels

Senior Full Stack Developer · Employee

Apr 2022 – Apr 2026

- Designed the ETL framework behind 300+ pipelines, with data-quality checks and monitoring.
- Embedded DuckDB in the analytics backend; selected workloads ran up to 100× faster.
- Handled CJS-to-ESM migration, CVE fixes, Azure and Databricks deployments. Mentored developers and helped agree on shared engineering practices.

TypeScript · Effect · React · Next.js · Fastify · Python · DuckDB · Azure

PREVIOUS EMPLOYMENT · CONTINUED

Proxyclick

Brussels

Full Stack Developer · Employee

Oct 2020 – Apr 2022

- Built subscriptions, payments and account-management flows for a visitor-management SaaS used in 100+ countries.
- Designed an event-based automation engine for visitor workflows, worked with Apache Pulsar, and helped migrate the frontend away from AngularJS.

TypeScript · Angular · Node.js · PostgreSQL · Apache Pulsar

Emisys · Full Stack Developer · Louvain-la-Neuve

Apr – Oct 2020

Built ticketing, scheduling, payment and refund flows for festival software used by thousands of people at once.

mPhase · Full Stack Developer · Montréal

2019 – 2020

Sole developer of a water-treatment logistics application, from data model to UI. Built a schema-driven form language for field inspections.

Extia Belgium · Full Stack Developer / Consultant

2018 – 2019

Built IPTV monitoring dashboards and a metrics platform with configurable billing, across media and hospitality projects.

IoTFactory · Full Stack Developer / Intern · Brussels

2018

Built Bluetooth/LoRa gateways that stored data locally and reconciled it when the connection returned, plus device-management and monitoring tools.

OPEN SOURCE

Argot

argot.tmonier.com

I built Argot to catch changes that don't fit a codebase's conventions, especially when coding agents write part of the code. It learns from Git history and checks diffs using statistics and code structure. No LLM; one offline Rust binary.

Rust · Tree-sitter · Git internals · Statistical ML · MCP

Effract

effract.tmonier.com

An experiment in writing React components as Effect programs, with typed errors, services and resources. Exploring how the same component logic can run across browser and server runtimes.

TypeScript · Effect · React · React Server Components

HOW I WORK

I use the business vocabulary in the code and keep dependencies and failure paths explicit with Effect. Integration tests and golden-master SQL protect behaviour and reporting figures. Upgrades and security fixes are regular work.

I use AMP, Claude Code and Codex to read unfamiliar code and implement scoped changes, then review the result. Small PRs and pairing help keep the work understandable.

EDUCATION & COMMUNITY

Institut Paul Lambin Bachelor in Computer Science, AI specialisation

Brussels · 2015–2018 · Cum Laude

Languages French (native) · English (C1) · Dutch (B1)

SoCraTes Belgium community; software craftsmanship workshops and Ziverge's Effect workshop (September 2026).

References and work samples available on request.