

# RICKARD LINDBOM

Fullstack Developer · Frontend & AI focus • Contracting via Lindforge Digital Studio  
rickard@lindforge.dev • Kalmar, Sweden • lindforge.dev • github.com/rapitzo

## PROFESSIONAL SUMMARY

---

Self-taught fullstack developer, 5+ years shipping production systems. I work at Milou on Vue 3 and C# / .NET customer portals built on EPiServer and Umbraco, used by hundreds of thousands of people. Before that I was the only engineer at Borgholm Återbruk for four years: a Python and Go backend, four frontends, and a Bluetooth bridge to an industrial scale whose RS-232 protocol I had to reverse engineer. I also build MCP servers and agent tooling, and I use Claude Code every day. Through Lindforge Digital Studio I take part-time contracts (10 to 20 h/week) and fixed-scope projects.

## SKILLS

---

**Languages:** TypeScript, Python, Go, C#, Rust, Kotlin, Dart, SQL

**Backend & Infrastructure:** .NET, FastAPI, Node.js, PostgreSQL, Docker, Azure (App Service, DevOps), AWS, Supabase, CI/CD, REST APIs, WebSocket, microservices, Git, Linux

**Frontend & Mobile:** Vue 3, React, Next.js, React Native, Flutter, Expo

**CMS / Platforms:** EPiServer (Optimizely), Umbraco

**AI / Agents:** MCP (Model Context Protocol), Claude Code, agent coordination, LLM pipelines, TensorFlow Lite, on-device inference

**Practices:** TDD, code review, CI/CD, Agile / Scrum, async collaboration, technical writing

## EXPERIENCE

---

### Fullstack Developer, Milou (2026-02 – Present)

Customer portals built on Vue and C# on Azure, serving hundreds of thousands of end users.

- Ship Vue 3 / TypeScript features across production customer portals while incrementally migrating legacy Vue 2 code, without downtime.
- Design CQRS / MediatR backend handlers in C# / .NET, integrated with EPiServer and Umbraco so the frontends can pull content over APIs.
- Own end-to-end delivery through Azure DevOps CI/CD to App Service, including Razor views, REST API controllers, and the tests around them.

### System Developer / Technical Lead, Borgholm Återbruk (2022 – 2026-01)

Sole engineer at a municipal reuse operation. Built and ran the full production stack (backend, four frontends, hardware integration), used daily by 10 staff and the public.

- Designed a Python and Go backend on PostgreSQL that handled roughly 40 scale transactions per day and around 90 product listings per week.
- Shipped four production frontends on top of it: a Flutter staff sales/inventory app, a React TV scheduling display, a Kotlin Android public kiosk, and an internal stats and weighing dashboard.

- Reverse engineered an undocumented RS-232 protocol and bridged the serial-only industrial scale to FastAPI over Bluetooth, removing manual weight entry from ~40 daily transactions.
- Built an AI-powered Facebook listing pipeline that drafted from images and throttled bursts, cutting staff authoring time by ~70% across 90 weekly listings.
- Handled the full lifecycle solo: stakeholder analysis, architecture, build, deploy, and production support for 10 daily users.

#### **IT-Support, Borgholm IT, Borgholms kommun (2020-02 – 2021-06)**

- Automated data workflows across 3 critical business systems, removing ~5 hours per week of manual entry.
- Documented processes and system solutions so operational knowledge stayed inside the support team.

## **PROJECTS**

---

### **livsmedel-mcp: MCP server for the Swedish Food Agency food database (2026)**

Unofficial MCP server for Livsmedelsverket's open food database, which had no MCP server. Five tools for searching foods, comparing them and working out recipe nutrition per serving, with paging, retries, clear error codes and 43 tests. Published on npm with a recorded session against the live API.

*Tech: TypeScript, MCP SDK, Node.js, Vitest • [github.com/Rapitzo/livsmedel-mcp](https://github.com/Rapitzo/livsmedel-mcp)*

### **Skein: Multiplayer agent IDE (Tauri / Rust, 2026)**

A desktop IDE where multiple AI agents share one tiled workspace. Real-time sync runs over Yjs CRDTs, agents coordinate through MCP, and the P2P transport is self-hosted over cloudflared so it has no recurring cost.

*Tech: Tauri 2, Rust, TypeScript, Bun, Yjs, MCP • [portfolio-rick.vercel.app/projects/skein](https://portfolio-rick.vercel.app/projects/skein)*

### **SwedishBirdex: On-device ML bird identification**

React Native mobile app that identifies Swedish bird species from a photo. Inference runs on-device, with no cloud round trip, against a TensorFlow model I trained on a curated Swedish dataset.

*Tech: React Native, TensorFlow Lite, Python (training pipeline)*

### **Lindforge Digital Studio**

My registered software studio, which I use for contract work: Vue / .NET / Umbraco / Optimizely capacity and fixed-price MCP server builds.

*Tech: React Native, Expo, Flutter, Next.js, TensorFlow, Supabase, AWS, Azure*

## **EDUCATION**

---

Self-taught engineer, ~10 years building production systems and side projects.

Currently focused on systems programming, distributed systems, and AI tooling (MCP, agent coordination, on-device ML).