

Marc Römmelt

Senior Software Engineer · TypeScript / Node.js

Freelance via 0815 Software UG · Fürth (Nürnberg), Germany · remote-first · mail@m-roemmelt.com · [LinkedIn](#)

Software engineer, mostly as the first or lead engineer at early-stage product companies, now freelance. I'm at my best where system design meets product — the territory where architecture, technology choices, and engineering culture all have to be set up from scratch and then defended as the thing grows. My backend work over the last five years has been Node.js/TypeScript in production throughout, with AWS as the cloud platform on the product work at Penzilla and TheMakers, and an increasing focus on distributed, event-driven systems. The work I care about most is reducing complexity through good design.

EXPERIENCE

Senior Software Engineer (freelance) — Web Computing GmbH, Münster

July 2026 – present

Sparkassen-Finanzgruppe — multichannel chatbot platform · remote

Refactoring a TypeScript/Node.js microservice platform behind a multichannel chatbot used across the Sparkassen-Finanzgruppe: decoupling services so they can be deployed independently, defining the API contracts between them, and keeping every change backwards-compatible while the platform stays in production. Infrastructure on Kubernetes/Helm. Around 90% hands-on; the rest is architecture and concept work with the platform team.

- **Agent-assisted from day one.** Claude Code with Atlassian and GitHub MCP integrations: tickets, comments and pull requests are context the agent reads and writes rather than something managed by hand — one workflow from understanding the codebase to implementing the change to keeping the ticket and PR trail current.

Stack: TypeScript, Node.js, Kubernetes, Helm, Claude Code (Atlassian & GitHub MCP).

Independent Software Engineer — Callura (0815 Software UG)

March – August 2026 · paused

Real-time voice and WhatsApp AI platform for German funeral homes — own product, sole engineer

Sole engineer of a multi-tenant platform that answers phone calls and WhatsApp messages for small funeral homes, documents every conversation, and gives the business an AI assistant over its own data. Built end-to-end: telephony, real-time AI orchestration, persistence, web client, deployment, compliance. Currently paused.

- **Real-time voice agent** that takes inbound calls over Telnyx Media Streams, runs the conversation through Gemini Live with synchronous and asynchronous tool-calling, and produces structured outcomes (cases, callbacks, contact updates) that reach the business through whichever channel makes sense — web dashboard, WhatsApp (integrated as Meta Tech Provider), phone.
- **Event-driven by construction.** Each call runs on its own event bus: media frames, model events and tool calls are decoupled, so synchronous lookups and asynchronous work (summaries, notifications) never block the audio path. Post-call pipelines regenerate per-contact memory digests from transcripts. Multi-tenant throughout — phone-number routing, per-business context injection, hybrid keyword + semantic search on PostgreSQL and Typesense.
- **Built with Claude Code in an agentic workflow.** Claude Code is the primary authoring tool: spec-first tasks, agent-generated implementation, then systematic review and re-ownership passes so every module is understood and held to the same standard as hand-written code. This is how I work day to day, on client projects as well.
- **Compliance treated as architecture, not paperwork.** EU data residency (DSGVO/AVV) is a hard constraint; the model endpoint is configuration so Gemini Developer API and Vertex AI are interchangeable as residency demands. EU AI Act Article 50 disclosure is built in; Telnyx contracted as Article 28 sub-processor.
- **Deliberately boring infrastructure** — TypeScript on Bun/ElysiaJS, HTMX for the web surface, Docker Compose with blue/green on a Hetzner VPS, Caddy reverse proxy, full OpenTelemetry to Dash0. The whole system runs on roughly the cost of dinner per month, which is the point.

Stack: TypeScript, Bun, ElysiaJS, HTMX, PostgreSQL, Typesense, Telnyx, Gemini Live API, WhatsApp Business Platform, Docker Compose, Caddy, OpenTelemetry, Dash0, Hetzner, Claude Code.

Lead Software Engineer — Penzilla GmbH, Munich

May 2023 – February 2026

B2B SaaS for corporate pension administration

Hired as the second engineer; became technical lead after the previous lead's departure ~12 months in. Owned platform architecture and the technical roadmap, and grew the engineering team from two to six. By the end of my tenure, the platform managed 3,000–4,000 active employment contracts across ~50 customers, including Pro7 and Personio.

- **Architected the platform** as a TypeScript monolith on NestJS with a hexagonal core, event-driven workflows over SQS/SNS, PostgreSQL on RDS, and React clients for both the admin and customer-facing surfaces — running on AWS (Fargate, Lambda, RDS) with the infrastructure defined in CDK. The choice of a modular monolith over microservices was deliberate and contrarian for the era — it kept per-engineer leverage high on a small team and avoided distribution costs the product didn't yet need. I'd make the same call again.
- **Led the migration from schema-per-tenant to column-based multi-tenancy.** The existing model had hit a wall: 100+ Prisma clients (one per schema, since Prisma doesn't share connection pools across schemas), saturated RDS connections, frequent query timeouts, and cross-tenant features that required brittle multi-client orchestration. Designed a parallel-write migration that adapted only the ORM access layer — no application code changes — with reads cut over per tenant once verified. After cutover: query timeouts dropped to zero, required DB connections went from 200 to 20, RDS cluster size was reduced, and cross-tenant features became trivial (a single Postgres `set_config` per request). Multi-tenancy stopped being a constraint on what we could build.
- **Designed and built the Payroll subsystem** — the part of the product where correctness is non-negotiable. Event-sourced by design: any past month must reproduce exactly, and corrections come in two flavors that have to be handled distinctly — forward-effective adjustments vs. retroactive corrections that rewrite historical records. Integrated asynchronously with DATEV Lohn und Gehalt, DATEV LODAS, SAP, and Personio, each with its own data shape and Stichtag (cutoff-day) semantics. The system has to stay consistent with both the upstream HR data and the downstream payroll provider across corrections from either side.
- **Rebuilt observability around OpenTelemetry on Grafana Cloud** — full traces, logs, and metrics with alerts defined as code and incident response wired through Grafana IRM. Replaced an OpsGenie + AWS CloudWatch setup that had degenerated into noise. The new system makes failures legible rather than just visible, and the alert definitions live next to the code that produces them.
- **Led the migration from Next.js to a separated React + NestJS architecture.** Next.js had been chosen before I joined and was actively making things harder — opaque rendering boundaries, awkward API patterns for a B2B product whose front-end and back-end concerns were genuinely separate. Built the case for the migration, got alignment, and executed it. The codebase became dramatically easier to reason about afterward.
- **Built the engineering team and the engineering culture.** Hired five engineers — initially alongside the previous tech lead, later leading hiring as the team grew. Established code review, static analysis, and conventions as the team's defaults; cared more about raising the bar than being the most senior person in the room. Made decisions collaboratively; my view usually carried, but the team carried it together.

Stack: TypeScript, Node.js, NestJS, React, Tailwind, PostgreSQL, AWS (Fargate, SQS, SNS, Lambda, RDS), CDK, OpenTelemetry, Grafana Cloud.

Technical Lead — TheMakers.ai GmbH, Berlin

April 2022 – January 2023

B2B marketplace for sustainable food product development

Hired to take a proof-of-concept built outside engineering and turn it into a maintainable product. Hired and led a team of four engineers and one product manager through technical interviews and onboarding.

- **Owned the redesign and rebuild** of the initial proof-of-concept into a production codebase — NestJS backend, Angular front-end, MongoDB persistence, hosted on AWS.
- **Established the engineering foundations** — coding standards, static analysis, automated testing — and shaped the product development process as a pragmatic mix of Kanban and Scrum that fit the team's actual rhythm rather than the framework's.
- **Integrated OptiSign's ML models** into the platform to surface sustainability and nutritional insights at the point where buyers were making product decisions.

Stack: TypeScript, Node.js, NestJS, Angular, MongoDB, AWS.

Freelance Lead Engineer — Fanzone Media GmbH, Berlin

January 2021 – March 2022

Web3 sports collectibles platform — blockchain-backed digital trading cards

Engaged exclusively and full-time as external lead engineer for the platform build. Primary technical contributor and de facto tech lead through the period.

- **Owned the platform architecture and the production stack.** Multi-target front-end (web, iOS App Store, Android Play Store) on Ionic/React with a backend built on MongoDB Atlas — Cloud Functions for typical request paths, GraphQL via MongoDB Realm for direct cluster access, and a separate Express server for blockchain operations whose latency made them a poor fit for cloud functions.
- **Set up the engineering foundations** — monorepo structure, CI/CD pipelines, development tooling, production infrastructure — and pushed test-driven development through regular mob-programming sessions and workshops as the team grew.

Stack: TypeScript, Node.js, Express, React, Ionic, MongoDB Atlas, MongoDB Realm, GraphQL, blockchain integrations.

Software Engineer — ELIGO GmbH, Berlin

June 2019 – June 2020

Software for psychometric assessment of job candidates

Worked across a Vue.js client, a Laravel web server, and a Node.js job processor.

- **Owned the Vue.js client** and led the effort to break a monolithic front-end into smaller, more maintainable components; designed and built multi-tenant support (theming, restructured authentication flow).
- **Pushed for engineering basics** the codebase didn't yet have — unit and integration tests, consistent code style, shared conventions.

Stack: Vue.js, Laravel, Node.js, Express.

SKILLS

Backend: Node.js, TypeScript, NestJS, Bun, ElysiaJS, Express

Messaging & async: AWS SQS/SNS, Redis Streams, event-driven workflows, event sourcing, real-time media streaming

Data: PostgreSQL, MongoDB, Typesense, Prisma, GraphQL

Cloud & infrastructure: AWS (Fargate, Lambda, SQS, SNS, RDS, CDK), Kubernetes/Helm, Docker, Hetzner, Caddy, CI/CD

Observability: OpenTelemetry, Grafana Cloud, Dash0, CloudWatch

AI-assisted development: Claude Code (daily, agentic workflow), MCP integrations (Atlassian, GitHub), LLM integration (Gemini Live, tool-calling), voice/WhatsApp AI agents

Frontend: React, Vue.js, Angular, Ionic, HTMX, Tailwind

Architecture: Modular monoliths, service decoupling & API contracts, hexagonal architecture, CQRS, multi-tenancy

INTERESTS & DIRECTION

Drawn toward **mature engineering at the technical frontier** — the territory where the discipline of doing things well actually matters and compounds. Most concretely this means **distributed systems and reliability engineering** — the current Sparkasse work is exactly that kind of problem, and it is where I'd like to spend the next chapter of my career. More broadly, I'm interested in **software design as complexity reduction** (the underlying craft I care about most) and in **AI systems** — as a daily user of coding agents, as a builder of LLM-based products, and as someone curious about where the field is going. Architecture should follow requirements; the design problems don't get less interesting as the scale grows, only more.

Outside of work: partner, an aging Podenco, walks on the Frankenalb and through the Lorenzer Reichswald, houseplants in unreasonable quantities, books on mindfulness and how to live well.

EDUCATION & LANGUAGES

Bachelor of Arts, Entrepreneurship — University of Erlangen-Nuremberg

September 2017

German (native), English (fluent)