

Nicholas Braun

Freelance Fullstack Engineer · Backend focus — Node.js & TypeScript, Go

✉ hallo@nicholasbraun.de ☎ +49 151 4051 3728 📍 Nürnberg/Fürth, Germany 🌐 nicholasbraun.de 🏠 Remote only

📅 Available: immediately · 5 days/week

Profile

Fullstack engineer with 10+ years building web products end to end — system design, data model and API through to the interface on top — with the centre of gravity on the backend. Distributed Node.js/TypeScript services: event sourcing and CQRS, real-time systems, queues, horizontal scaling, observability and long-term maintainability.

Client work spans event-sourced microservices with end-to-end-encrypted real-time chat, a three-year engagement on a video-commerce platform, three production applications delivered as sole developer, and public-sector delivery for the German state of Baden-Württemberg. Go for personal projects, and the direction I want client work to take.

Occasional work continues for long-standing frontend clients; new engagements are backend or backend-heavy fullstack.

Work Experience

ressourcenmangel (rsm)

Remote

Frontend Developer & Feature Owner (Freelance)

Jan 2020 – Jul 2026

- Multi-tenant component library and static-asset pipeline for the public-sector web presence of the German state of Baden-Württemberg — Landesportal, Beteiligungsportal and ministry variants, 7–10+ tenants from a single source. Ongoing client relationship, project-based alongside other engagements.
- Built the job-search feature end to end: API query layer, postal-code and local search, filter validation, accessible pagination, and an iframe/script-tag embed mode.
- Introduced TypeScript into the codebase and established the testing setup (property-based tests, Playwright visual-regression and axe-core a11y against BITV/WCAG).
- **Tech:** React, TypeScript, SCSS/BEM, Storybook, Vitest, Playwright + axe-core, TanStack Query, GitLab CI.

<https://rysm.com>

qrago

Remote

Fullstack Engineer (Freelance)

Jan 2025 – Jun 2025

- Medical patient-transport (Krankentransport) logistics SaaS connecting hospitals and care facilities with ambulance operators: ride requests, marketplace, vehicle and driver planning.
- De-facto owner of the real-time chat subsystem, delivered end to end as an event-sourced Node.js microservice.
- Built the in-app chat from scratch on the Matrix protocol (self-hosted Synapse homeserver, matrix-js-sdk): rooms, message timeline, read receipts, unread badges.
- Delivered end-to-end encryption: cross-signing, SSSS secret storage, key backup and room-key request handling.
- Implemented the chat backend on a CQRS and event-sourcing foundation, with aggregates emitting domain events and command-rejection handling; moved Matrix room creation into a saga.
- Push notifications (background and foreground) via Firebase/FCM and Signal.
- **Tech:** Node.js, @weave-js microservices framework (Redis Pub/Sub broker), MongoDB, Redis, custom CQRS/event-sourcing package, Zipkin tracing, Stripe. Production monorepo of ~20 backend services and gateways, 3 Vue web apps and 6 shared packages.

<https://qrago.de>

jooli

Remote

Senior Fullstack Engineer (Freelance)

Feb 2021 – Jun 2024

•

Video-commerce / live-shopping platform — vendors publish shoppable videos, customers browse a feed, watch and buy — connected to the German jewelry retailer Juwelo. Small team of ~5 engineers; backend was the centre of gravity, alongside both web frontends.

- Platform rolled out in two markets (Germany and India), with the Indian rollout carrying the bulk of the traffic.
- Owned the migration of the backend from routing-controllers to NestJS: a cross-cutting framework migration touching every controller and module, establishing the modular structure and DI conventions the backend still runs on.
- Designed the video-status pipeline end to end — Bull/Redis queue, processor, WebSocket gateway with Redis socket adapter, migration and cron; fixed cross-environment queue-namespace contamination.
- Migrated real-time video-impression state from an in-process Map to Redis, resolving multi-instance and horizontal-scaling bugs.
- Led the CRA to Next.js SSR migration on the customer watch/share frontend (needed for Open Graph share tags) and owned client-side impression tracking.
- **Tech:** NestJS 9, TypeScript, TypeORM + MySQL, Redis, Bull, WebSockets (socket.io + Redis adapter), gRPC/Protobuf, Google Cloud Pub/Sub, Algolia, Next.js 12. Three repositories; core backend of ~40+ modules and 116 migrations.

4ARGE (client: FIFA)

Remote

Fullstack Engineer (Freelance)

Jul 2020 – Aug 2020

- Fullstack prototype engagement — product and feature details restricted under NDA.
- **Tech:** Node.js, NestJS (backend); TypeScript, React (frontend).

ressourcenmangel (rsm)

Remote

Fullstack Developer (permanent)

Mar 2016 – Dec 2019

Sole developer on three production applications, each owned end to end from data model to interface.

- **Universität Halle — internal analysis tool for programme evaluation:** graph data model in MongoDB with shortest-path analysis (Dijkstra and related graph algorithms) implemented in the Node.js backend, surfaced through an interactive D3 graph frontend.
- **Weisse Liste — health-app directory and hospital search (service discontinued):** two public-facing search products with nationwide hospital coverage. Built a GraphQL layer in front of the client-owned API and database as a caching and aggregation tier, decoupling both frontends from a data source outside our control.
- Designed and built the custom UI component system both search products were assembled from — shared components, theming and search/filter primitives reused across both apps.
- Frontend delivery across the agency's public-sector and corporate portfolio: baden-wuerttemberg.de and associated state portals and ministry sites, Bertelsmann, Agora Energiewende, degewo.
- **Tech:** Node.js, TypeScript, GraphQL, MongoDB, D3.

<https://rysm.com>

Personal Projects

Job-discovery crawler service (Go)

Mar 2026 – present

- Long-running Go service that discovers company career pages across the web and extracts job listings by keyword — REST API and an embedded React dashboard served from a single binary, with crawls managed as background runs.
- Crash-safe and resumable by design: no authoritative state in process memory. Postgres holds the durable catalog, definitions and listings; Redis holds per-run frontier state (per-domain FIFO queues with cooldown, visited set, in-flight leases). Next and AddURL are each a single Lua script, so concurrent workers can neither double-pop nor race the dedup; a restarted process adopts and resumes in-flight runs, and SIGINT drains them. Model calls run off a Redis queue, so a slow LLM never blocks the crawl and a crash redelivers the work rather than dropping it.
- Generic goroutine worker pools and composable generic filter chains; robots.txt (RFC 9309) compliance with per-host caching and singleflight collapsing; HTTP retry decorator with exponential backoff behind a Downloader interface.
- Cost-aware LLM use: rule-based gates decide most pages for free and the model only sees the ambiguous ones. Classification and extraction sit behind small interfaces (OpenRouter or any OpenAI-compatible endpoint, including

local Ollama) with prompt-injection mitigations. Postgres and Redis repositories tested against real instances via testcontainers.

- Reads applicant tracking systems (Greenhouse, Lever, Personio and seven more) through their APIs, and crawls everything else.

[GitHub](#)

leetcode-anki2026

- Terminal UI for practising LeetCode problems with spaced repetition (personal project): browse, scaffold, solve, test and submit; accepted problems enter an SM-2 review rotation.
- Single-binary Go CLI/TUI built with Bubble Tea (4-screen flow), interfaces at every external boundary and constructor DI throughout.
- Built strictly test-first (red-green-refactor). Contract tests make shape-invariant assertions against both a fake and the live LeetCode API under a build tag; GitHub Actions on PRs and a daily cron.

[GitHub](#)

Skills

Languages & Runtimes

Go, TypeScript, JavaScript, Node.js, SQL

Backend Frameworks

NestJS, Express

Architecture

Event Sourcing, CQRS, Event-Driven Architecture, Sagas, Domain-Driven Design, Hexagonal Architecture, Modular Monolith, Microservices

APIs & Protocols

REST, GraphQL, gRPC / Protobuf, OpenAPI

Distributed Systems & Real-time

Redis, Bull, Google Cloud Pub/Sub, WebSockets / Socket.IO, Push notifications (FCM, APNs), Worker pools, Rate limiting

Databases

PostgreSQL, MySQL, MongoDB, SQLite, pgx, TypeORM, goose

Infrastructure & CI

Docker, Google Cloud Platform, Azure, AWS, DigitalOcean, GitHub Actions, GitLab CI

Observability

OpenTelemetry, Prometheus, Grafana, Loki, Zipkin, pprof

Testing & Quality

TDD, Contract testing, Property-based testing, Go testing (table-driven, -race), Jest, Vitest, Playwright, Testcontainers, testify, fast-check

Frontend

React, Next.js, Vue + Pinia, TanStack Query, Tailwind, WCAG/BITV a11y

Education

Technische Universität Berlin (TU Berlin)

Computer Science

B.Sc.
Berlin, Germany • 2012 – 2016

Interests

Rock Climbing, Specialty Coffee