

Hauke Schnau

Software Engineer, TypeScript, React & Architecture

hauke@schnau.dev schnau.dev Bremen region, Germany [LinkedIn](#) [GitHub](#) German: Native · English: Fluent

01 PROFILE

I build production software in TypeScript and React, and I'm happiest when a type error catches a mistake before a user does. I've been shipping since 2021, including in a mature insurance-comparison team at CHECK24 and now with architecture responsibility at BuildlinX, where I turn ambiguous requirements into decisions, briefs, and working software across web, backend, and edge systems. Alongside that I build and maintain my own products, most of all Studienbuch, a privacy-sensitive school app. I finished my B.Sc. in Computer Science at the University of Hamburg with a final grade of 1.47 and was recognized for the best degree result of my cohort.

02 EVIDENCE AT A GLANCE

Fullstack TypeScript

BuildlinX and Studienbuch: interfaces, APIs, domain models, persistence, delivery, and runtime behavior.

React and TanStack

Production work with React, Query, and Router; Form and Table in Studienbuch and technical investigations.

Architecture through operation

Current responsibility spans project briefs, domain modeling, backend boundaries, NixOS edge systems, and review.

Quality in the delivery loop

Types, Playwright, previews, CI/CD, reproducible environments, code review, and verification on running systems.

03 EXPERIENCE

BuildlinX GmbH

Oct 2024 – present

Software Engineer with architectural ownership · Apr 2026 – present

Technical ownership for selected building-management product areas across web, backend, and edge systems.

- Specified a new NixOS-based edge platform and operating system for two developers, then reviewed architecture and implementation with the CEO/CTO.
- Work on domain modeling and backend architecture for complex product workflows.
- Contribute to reproducible NixOS-based edge configuration for controlled provisioning, updates, and operation.
- Explore AI-assisted engineering workflows with tools such as Codex while keeping reviews, tests, maintainability, and code quality explicit.

Software Engineer · Oct 2024 – Mar 2026

Product engineering in a small team across frontend, backend boundaries, delivery, and edge operations.

- Helped shape the frontend stack and architecture around TypeScript, React, Vite, pnpm, TanStack Query and Router, Storybook, Playwright, Tailwind CSS, and shadcn/ui.
- Built interactive building-management workflows, analytics, and operational interfaces.
- Implemented type-safe API integration and CI/CD workflows, including branch-based preview deployments.
- Substantially reduced cellular data traffic on operational edge systems, eliminating recurring monthly overage charges.

TypeScript · React · TanStack Query · TanStack Router · Node.js · Go · OpenAPI · PostgreSQL · Playwright · Nix · Vite · Storybook · Tailwind CSS

STACK SIGNAL

TypeScript · React · TanStack Query · TanStack Router · TanStack Form · Node.js · Effect · PostgreSQL · Vitest · Playwright · NixOS

03 EXPERIENCE

CHECK24 Vergleichsportal für Versicherungsprodukte GmbH

Aug 2023 – Oct 2024

Software Engineer · Aug 2023 – Oct 2024

Production feature development and modernization in a mature insurance-comparison product team.

- Built product features in a mature environment using React, TypeScript, PHP, and Symfony.
- Contributed to the incremental JavaScript-to-TypeScript migration, resolving accumulated type issues and refactoring affected areas so subsequent changes could be made with greater confidence.
- Built mobile sorting and filtering interactions for an insurance-comparison flow that felt and performed like native bottom sheets while remaining part of a web application.
- Supported the Webpack-to-Vite migration and modernized complex React and PHP/Symfony legacy code toward clearer target structures.
- Worked in the regular product flow with product managers, Jira, code reviews, technical discussions, and established delivery processes.
- Left with a positive performance review and an invitation to return in a full-time role after graduation.

TypeScript · React · Vite · PHP · Symfony

| *Positive performance review and invitation to return after graduation.*

homevoice GmbH

Oct 2021 – Jul 2022

Front-End Developer · Oct 2021 – Jul 2022

Product development for a property-management web application in a small software team.

- Contributed to product features across React/TypeScript frontend and Java/Spring Boot backend areas.

TypeScript · React · Java · Spring Boot

Freelance

Aug 2022 – Dec 2023

Web Developer · Aug 2022 – Dec 2023

Websites for artists and small businesses, from direct client communication through implementation, CMS work, and SEO.

- Translated content, branding, and business needs into maintainable web presences.

JavaScript · PHP · Kirby CMS · Tailwind CSS

04 WORKING PRINCIPLES

From unclear requirements to a plan

Structure unclear requirements, expose the decisions hidden inside them, and connect product goals to a technical plan a team can implement and review.

Make safe changes easy

Invest in types, tests, previews, automation, and reproducible environments so quality is part of the fast feedback loop rather than a late gate.

Follow features into production

Follow product behavior through interface, API, domain model, persistence, delivery, and runtime failures rather than stopping at one layer.

Verify AI-assisted work

Let agents accelerate exploration and implementation, but require their output to survive the same contracts, reviews, automated checks, and real-system verification as any other change.

05 EDUCATION

Universität Hamburg

Bachelor of Science in Computer Science

Oct 2022 – Mar 2026

1.47 on the German scale (1.0–4.0, 1.0 best)

Recognized for the best degree result of the cohort in the B.Sc. Computer Science program

06 SELECTED PRODUCT CASE

May 2023 – present

Studienbuch

Studienbuch explores how a privacy-sensitive school app can preserve user control while gaining reliable backup, synchronization, and connected features.

MY ROLE

I own the product direction and technical architecture across mobile, web, domain logic, APIs, persistence, synchronization, testing, and delivery.

PRODUCT TENSION

The original privacy model had become effectively local only. That protected data but made backup, synchronization, and richer connected workflows structurally awkward. Retrofitting those capabilities into the existing shape would have meant working around the real limitation, so I redesigned the ownership and synchronization model instead.

ARCHITECTURAL DECISIONS

Typed domain core

Extract reusable policy, selectors, and formatting into a framework-independent typed domain core instead of coupling product behavior to one UI generation.

Explicit sync contracts

Build Groundswell around explicit event and adapter contracts for offline publication, canonical verification, persistence, broadcast, retry, and replay.

Protected external integration

Keep the unstable WebUntis integration behind strict schemas, typed services, recovery behavior, and contract/live tests.

OUTCOME

One school continues to use the current product while the new generation evolves behind explicit boundaries for domain logic, platform UI, external systems, persistence, synchronization, and delivery.

WHAT CHANGED IN MY JUDGMENT

I initially pushed event sourcing too far. Mostly static reference facts such as when a school was founded did not benefit from events and created avoidable complexity. I now reserve events for behavior where history, ordering, replay, or synchronization earns their cost, and use simpler fetch-and-cache boundaries for reference data.

SUPPORTING SYSTEMS

WebUntis API

A typed Effect client for modern WebUntis REST surfaces, consumed by Studienbuch.

TypeScript 7 · Effect 4 · Vitest · Contract/Live testing · Vite+

Groundswell

Explicit event and adapter contracts for offline publication, persistence, broadcast, retry, and replay.

TypeScript · Effect · Hono · Drizzle ORM · PostgreSQL · Vitest

Current rewrite: TypeScript 7 · React Native · Expo · TanStack Start · TanStack Router · Effect · Bun · NativeWind · Nix · Local-first · Across generations: Flutter · Dart · React Native · Expo · TanStack Query · TanStack Router · TanStack Form · TanStack Table · tRPC · Hono · Drizzle ORM · PostgreSQL · SQLite

07 PRODUCT LAB

T3 Code

A personal fork lets me specify, prototype, and dogfood workflow ideas quickly enough to learn which tools earn a place in my engineering work.

A personal product lab for the agent workflows I use every day. I identify friction, write product and technical specifications, direct agent-assisted prototypes, and keep only what survives daily use.

Cross-provider orchestration

One agent can initiate work in another provider’s thread, such as Codex creating a Cursor Composer thread.

OUTCOME

Orchestration, monitoring, pinning, tagging, and usage visibility are now part of my workflow. Cross-machine handoff remains active product work.

One-action handoff

Move a task and its workspace context to a persistent Linux host for background execution and later phone control.

08 SELECTED TECHNICAL WORK

Stack Overflow semantic search

Oct 2025 – Jan 2026

Set the technical direction and implemented most of a four-person university project, from bounded-memory processing of a 93 GB dump through embeddings, retrieval, re-ranking, and evaluation to a local source-grounded chat.

Search pipeline: Python · SentenceTransformers · FAISS · SQLite · Flask · Nix · Semantic search · Retrieval evaluation · StackChat: TypeScript · Next.js · React · Vercel AI SDK · Ollama · Bun · Tailwind CSS

Moodle & STiNE automation

Feb 2024 – 2025

Built and relied on a Moodle crawler that kept changing course material in a predictable local structure. Later explored typed extraction from the public STiNE course catalogue.

Moodle crawler: Go · SAML authentication · Web scraping · STiNE experiment: TypeScript · Bun · Effect · Vitest · Web scraping

IGS Lilienthal website

Sep 2025 – present

A ground-up redesign of a sprawling school website, with clearer information architecture, structured content in Kirby CMS, and a reliable deployment path.

Kirby CMS 5 · PHP 8 · Tailwind CSS · JavaScript · Nix · Gitea Actions · systemd

nix-infra-modules

Jul 2026 – present

Reusable Nix modules extracted from the infrastructure I use to operate macOS and NixOS machines, application deployments, runners, ingress, backups, and development environments.

Nix · NixOS · nix-darwin · systemd · GitHub Actions

09 OPEN SOURCE

TanStack Table	Reduced a Chrome freeze to a deterministic React Suspense and TanStack Table reproduction, traced the library interaction, and proposed a fix with regression coverage.	Upstream issue
T3 Code	Fixed provider-session recovery by retaining and reusing the correct session binding.	Pull request
T3 Code	Normalized custom Codex home-directory handling across application boundaries.	Pull request
Home Manager	Added declarative installation support for Codex skills.	Pull request
Coolify	Added Zitadel as an OAuth provider, including user-facing translations.	Pull request

10 TECHNOLOGY

LANGUAGES

TypeScript · Go

FRONTEND

React · React Native · TanStack Query · TanStack Router · TanStack Form · TanStack Table

BACKEND & RUNTIMES

Node.js · Bun · Hono · tRPC · Effect

DATA & APIS

OpenAPI · PostgreSQL · SQLite · Drizzle ORM

TESTING

Vitest · Playwright · Storybook

BUILD & TOOLING

Vite · pnpm

INFRASTRUCTURE

Nix · NixOS · Docker