

Hauke Schnau

Software Engineer, TypeScript, React & Architektur

hauke@schnau.dev

schnau.dev

Region Bremen, Deutschland

LinkedIn

GitHub

Deutsch: Muttersprache · Englisch: Fließend

01 PROFIL

Ich entwickle Produktionssoftware in TypeScript und React und bin am zufriedensten, wenn ein Typfehler einen Bug abfängt, bevor Nutzende ihn finden. Seit 2021 liefere ich Software aus, unter anderem in einem etablierten Team für Versicherungsvergleiche bei CHECK24 und inzwischen mit Architekturverantwortung bei BuildlinX. Dort verwandle ich unklare Anforderungen in Entscheidungen, Briefings und lauffähige Software über Web-, Backend- und Edge-Systeme hinweg. Daneben baue und pflege ich eigene Produkte, vor allem Studienbuch, eine datenschutzorientierte Schul-App. Meinen B.Sc. Informatik an der Universität Hamburg habe ich mit der Note 1,47 abgeschlossen und wurde als Jahrgangsbester ausgezeichnet.

02 WAS ICH MITBRINGE

Fullstack TypeScript

BuildlinX und Studienbuch: Oberflächen, APIs, Domain Models, Persistenz, Delivery und Laufzeitverhalten.

React und TanStack

Produktive Arbeit mit React, Query und Router; Form und Table in Studienbuch und technischen Analysen.

Architektur bis in den Betrieb

Meine aktuelle Verantwortung umfasst Projekt-Briefings, Domain Modeling, Backend-Grenzen, NixOS-Edge-Systeme und Reviews.

Qualität im Delivery-Loop

Typen, Playwright, Previews, CI/CD, reproduzierbare Umgebungen, Code Reviews und Prüfung an laufenden Systemen.

03 BERUFSERFAHRUNG

BuildlinX GmbH

Okt. 2024 – heute

Softwareentwickler mit Architekturverantwortung · Apr. 2026 – heute

Technische Verantwortung für ausgewählte Produktbereiche im Gebäudemanagement, von Web über Backend bis zu den Edge-Systemen.

- Spezifizierte eine neue NixOS-basierte Edge-Plattform samt Betriebssystem für zwei Entwickler:innen und prüfte Architektur sowie Umsetzung gemeinsam mit CEO/CTO.
- Arbeite an Domain Modeling und Backend-Architektur für komplexe Produkt-Workflows.
- Arbeite an reproduzierbarer NixOS-basierter Edge-Konfiguration für kontrollierte Bereitstellung, Updates und Betrieb.
- Erprobe KI-gestützte Engineering-Workflows mit Werkzeugen wie Codex, wobei Reviews, Tests, Wartbarkeit und Codequalität explizite Anforderungen bleiben.

Softwareentwickler · Okt. 2024 – März 2026

Produktentwicklung in einem kleinen Team, quer über Frontend, Backend-Schnittstellen, Delivery und Edge-Betrieb.

- Prägte den Frontend-Stack und dessen Architektur mit: TypeScript, React, Vite, pnpm, TanStack Query und Router, Storybook, Playwright, Tailwind CSS und shadcn/ui.
- Entwickelte interaktive Gebäudemanagement-Workflows, Analysen und operative Oberflächen.
- Implementierte typsichere API-Integration und CI/CD-Workflows einschließlich branchbasierter Preview-Deployments.
- Reduzierte den Mobilfunk-Datenverkehr produktiver Edge-Systeme deutlich und vermied dadurch wiederkehrende monatliche Zusatzkosten.

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

STACK IM ÜBERBLICK

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

03 BERUFSERFAHRUNG

CHECK24 Vergleichsportal für Versicherungsprodukte GmbH

Aug. 2023 – Okt. 2024

Softwareentwickler · Aug. 2023 – Okt. 2024

Produktive Feature-Entwicklung und Modernisierung in einem etablierten Team für Versicherungsvergleiche.

- Entwickelte Produktfeatures in einer etablierten Produktumgebung mit React, TypeScript, PHP und Symfony.
- Wirkte an der schrittweisen Migration von JavaScript zu TypeScript mit, löste angesammelte Typfehler und refaktorierte betroffene Bereiche, sodass spätere Änderungen sicherer umgesetzt werden konnten.
- Entwickelte mobile Sortier- und Filterinteraktionen für einen Versicherungsvergleich, die sich innerhalb einer Webanwendung wie native Bottom Sheets anfühlten und verhielten.
- Unterstützte die Migration von Webpack zu Vite und modernisierte komplexen React- sowie PHP/Symfony-Legacy-Code in Richtung klarerer Zielstrukturen.
- Arbeitete im regulären Produktprozess mit Produktmanagement, Jira, Code Reviews, technischen Diskussionen und etablierten Delivery-Prozessen.
- Schied mit einer positiven Beurteilung und dem Angebot aus, nach dem Studium in Vollzeit zurückzukehren.

TypeScript · React · Vite · PHP · Symfony

| *Positive Beurteilung und Angebot, nach dem Studium zurückzukehren.*

homevoice GmbH

Okt. 2021 – Juli 2022

Front-End-Entwickler · Okt. 2021 – Juli 2022

Produktentwicklung für eine Hausverwaltungs-Webanwendung in einem kleinen Softwareteam.

- Arbeitete an Produktfeatures, sowohl im React/TypeScript-Frontend als auch im Java/Spring-Boot-Backend.

TypeScript · React · Java · Spring Boot

Freelance

Aug. 2022 – Dez. 2023

Web-Entwickler · Aug. 2022 – Dez. 2023

Websites für Künstler:innen und kleine Unternehmen, von der direkten Kundenkommunikation über Umsetzung und CMS bis SEO.

- Setzte Inhalte, Markenauftritt und geschäftliche Anforderungen in wartbare Webauftritte um.

JavaScript · PHP · Kirby CMS · Tailwind CSS

04 ARBEITSWEISE

Von unklaren Anforderungen zum Plan

Strukturiere unklare Anforderungen, mache die darin verborgenen Entscheidungen sichtbar und verbinde Produktziele mit einem technischen Plan, den ein Team umsetzen und prüfen kann.

Sichere Änderungen einfach machen

Investiere in Typen, Tests, Previews, Automatisierung und reproduzierbare Umgebungen, damit Qualität Teil des schnellen Feedback-Loops statt einer späten Hürde ist.

Features bis in den Betrieb begleiten

Verfolge Produktverhalten von der Oberfläche über API, Domänenmodell, Persistenz und Delivery bis zu Laufzeitfehlern, statt an einer Schicht aufzuhören.

AI-unterstützte Arbeit verifizieren

Lasse Agents Recherche und Umsetzung beschleunigen, verlange aber für ihre Ergebnisse dieselben Typ- und API-Garantien, Reviews, automatisierten Checks und Prüfungen am laufenden System wie für jede andere Änderung.

05 AUSBILDUNG

Universität Hamburg

Bachelor of Science Informatik

Okt. 2022 – März 2026

Abschlussnote 1,47

Ausgezeichnet für den besten Abschluss des Jahrgangs im Studiengang B.Sc. Informatik

06

AUSGEWÄHLTES PRODUKT

Studienbuch

Mai 2023 – heute

Studienbuch erprobt, wie eine datenschutzorientierte Schul-App die Kontrolle der Nutzer:innen über ihre Daten bewahren und zugleich zuverlässiges Backup, Synchronisation und vernetzte Funktionen ermöglichen kann.

MEINE ROLLE

Ich verantworte Produktrichtung und technische Architektur: Mobile, Web, Domänenlogik, APIs, Persistenz, Synchronisation, Tests und Delivery.

PRODUKTFRAGE

Das ursprüngliche Datenschutzmodell war faktisch Local-only geworden. Das schützte Daten, machte Backup, Synchronisation und umfangreichere vernetzte Abläufe jedoch strukturell schwierig. Diese Funktionen nachträglich in die bestehende Form zu pressen erschien weniger sinnvoll, als Datenhoheit und Synchronisationsmodell neu zu entwerfen.

ARCHITEKTURENTSCHEIDUNGEN

Typsicherer Domänenkern

Wiederverwendbare Regeln, Selektoren und Formatierung werden in einen frameworkunabhängigen, typsicheren Domänenkern extrahiert, statt Produktverhalten an eine UI-Generation zu koppeln.

Explizite Sync-Regeln

Groundswell basiert auf expliziten Event- und Adapterverträgen für Offline-Publikation, kanonische Verifikation, Persistenz, Broadcast, Retry und Replay.

Gekapselte Fremdsystem-Integration

Die instabile WebUntis-Integration bleibt hinter strikten Schemas, typsicheren Services, Recovery-Verhalten sowie Contract- und Live-Tests gekapselt.

ERGEBNIS

Eine Schule nutzt weiterhin das bestehende Produkt. Die neue Generation entwickelt sich parallel mit klaren Grenzen zwischen Domain-Logik, Plattform-UI, Fremdsystemen, Persistenz, Synchronisation und Delivery.

WAS ICH DARAUS GELERNT HABE

Anfangs habe ich Event Sourcing zu weit getrieben. Weitgehend statische Referenzdaten wie das Gründungsjahr einer Schule profitierten nicht von Events und erzeugten vermeidbare Komplexität. Heute nutze ich Events dort, wo Historie, Reihenfolge, Replay oder Synchronisation ihren Aufwand rechtfertigen, und einfachere Fetch-and-Cache-Grenzen für Referenzdaten.

UNTERSTÜTZENDE SYSTEME

WebUntis API

Ein typsicherer Effect-Client für moderne WebUntis-REST-Schnittstellen, der von Studienbuch verwendet wird.

TypeScript 7 · Effect 4 · Vitest · Contract-/Live-Testing · Vite+

Groundswell

Explizite Event- und Adapterverträge für Offline-Publikation, Persistenz, Broadcast, Retry und Replay.

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

Aktueller Rewrite: TypeScript 7 · React Native · Expo · TanStack Start · TanStack Router · Effect · Bun · NativeWind · Nix · Local-first · Über alle Generationen: Flutter · Dart · React Native · Expo · TanStack Query · TanStack Router · TanStack Form · TanStack Table · trPC · Hono · Drizzle ORM · PostgreSQL · SQLite

07 PRODUKTLABOR

T3 Code

Ein persönlicher Fork erlaubt mir, Workflow-Ideen schnell genug zu spezifizieren, zu prototypisieren und selbst zu erproben, um herauszufinden, welche Werkzeuge sich einen Platz in meiner Engineering-Arbeit verdienen.

Ein persönliches Produktlabor für die Agenten-Workflows, die ich täglich nutze. Ich identifiziere Reibung, schreibe Produkt- und technische Spezifikationen, steuere agentenunterstützte Prototypen und behalte nur, was sich im Alltag bewährt.

ERGEBNIS

Orchestrierung, Monitoring, Pinning, Tagging und Nutzungsübersicht sind heute Teil meines Workflows. Der rechnerübergreifende Handoff ist weiterhin aktive Produktarbeit.

Providerübergreifende Orchestrierung

Ein Agent kann Arbeit in einem Thread eines anderen Providers starten, etwa Codex in Cursor Composer.

Handoff mit einer Aktion

Aufgabe und Workspace-Kontext wechseln für Hintergrundarbeit und spätere Steuerung vom Smartphone auf einen persistenten Linux-Rechner.

08 AUSGEWÄHLTE TECHNISCHE ARBEITEN

Stack Overflow semantic search

Okt. 2025 – Jan. 2026

Prägte die technische Richtung und implementierte den Großteil eines vierköpfigen Universitätsprojekts: vom 93-GB-Datendump über Embeddings, Retrieval, Re-Ranking und Evaluation bis zu einem lokalen Chat mit Quellen.

Suchpipeline: Python · SentenceTransformers · FAISS · SQLite · Flask · Nix · Semantische Suche · Retrieval-Evaluation · StackChat: TypeScript · Next.js · React · Vercel AI SDK · Ollama · Bun · Tailwind CSS

Moodle & STiNE automation

Feb. 2024 – 2025

Baute und nutzte intensiv einen Moodle-Crawler, der geändertes Kursmaterial lokal verlässlich strukturiert hielt. Später erprobte ich die typisierte Extraktion aus dem öffentlichen STiNE-Vorlesungsverzeichnis.

Moodle-Crawler: Go · SAML-Authentifizierung · Web-Scraping · STiNE-Experiment: TypeScript · Bun · Effect · Vitest · Web-Scraping

IGS Lilienthal website

Sep. 2025 – heute

Ein grundlegendes Redesign einer weit verzweigten Schulwebsite mit klarerer Informationsarchitektur, strukturierten Inhalten in Kirby CMS und einem verlässlichen Deployment-Pfad.

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

nix-infra-modules

Juli 2026 – heute

Wiederverwendbare Nix-Module aus der Infrastruktur, mit der ich macOS- und NixOS-Systeme, Anwendungs-Deployments, Runner, Ingress, Backups und Entwicklungsumgebungen betreibe.

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

09 OPEN SOURCE

TanStack Table	Reduzierte einen Chrome-Freeze auf eine deterministische Reproduktion mit React Suspense und TanStack Table, grenzte die Bibliotheksinteraktion ein und schlug einen Fix mit Regressionstests vor.	Upstream-Issue
T3 Code	Behob die Wiederherstellung von Provider-Sessions, indem die korrekte Session-Bindung erhalten und wiederverwendet wird.	Pull Request
T3 Code	Vereinheitlichte die Behandlung benutzerdefinierter Codex-Home-Verzeichnisse über Anwendungsgrenzen hinweg.	Pull Request
Home Manager	Ergänzte deklarative Installationsunterstützung für Codex Skills.	Pull Request
Coolify	Ergänzte Zitadel als OAuth-Provider einschließlich der Benutzeroberflächen-Übersetzungen.	Pull Request

10 TECHNOLOGIEN

SPRACHEN

TypeScript · Go

FRONTEND

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

BACKEND & LAUFZEITEN

Node.js · Bun · Hono · tRPC · Effect

DATEN & APIS

OpenAPI · PostgreSQL · SQLite · Drizzle ORM

TESTING

Vitest · Playwright · Storybook

BUILD & TOOLING

Vite · pnpm

INFRASTRUKTUR

Nix · NixOS · Docker