



Gabriel Suter

Full-Stack-Entwickler · Informatiker EFZ Applikationsentwicklung

kontakt@gabrielsuter.dev

+41 78 257 32 95

Däniken SO, Schweiz

gabrielsuter.dev

linkedin.com/in/gabriel-suter-18973329a

Verfügbar: ab sofort, 100 %

Arbeitsort: 100 % remote, Schweiz

Abrechnung: Stunden- oder
Meilensteinbasis

Sprachen: Deutsch (Muttersprache),
Englisch B2

Profil

Full-Stack-Entwickler mit vier Jahren Berufspraxis in Angular, TypeScript, C# und Microsoft SQL Server bei einem Microsoft-Partner, dazu eigene Produkte in React, Node und Python. Schwerpunkt auf Schnittstellenarbeit, Datenpipelines und SaaS-Produkten: belastbare Integrationen gegen fremde Dienste, Stripe-Abomodelle mit Freemium-Logik, Postgres-Architekturen mit Row Level Security. Die unten aufgeführten Projekte sind Solo-Umsetzungen vom Konzept über Backend und Frontend bis zum Betrieb, inklusive Sicherheitsaudits und Übergabedokumentation.

180'000

Zeilen eigener Code über fünf Projekte,
ohne Abhängigkeiten und Übersetzungen

47

serverseitige Funktionen in Produktion
(Deno)

15

fremde APIs in Produktion, von Stripe bis
swisstopo

4 Jahre

Berufspraxis mit Angular, TypeScript, C#
und SQL Server

Projekte

Renditeradar · Aggregator für Renditeobjekte

Persönliches Projekt · September 2026 · rund 3'500 Zeilen · öffentliches Repository unter MIT, Demo online

Next.js-Anwendung mit statisch erzeugten Detailseiten und Umkreissuche über räumlichen Index; Arbeitsprobe mit generierten Objektdaten.

- 400 Detailseiten statisch erzeugt mit Erneuerung im Hintergrund, Liste dynamisch über Suchparameter; Lighthouse 100 in allen vier Kategorien auf der Detailseite.
- Python-Pipeline mit Normalisierung unsauberer Rohdaten, Duplikaterkennung über Hash und idempotentem Upsert; der zweite Lauf ändert keine Zeile.
- PostgreSQL mit zusammengesetzten Indizes, Row Level Security als Schreibschutz und PostGIS: Umkreis- und Nachbarsuche über GiST-Index, Indexnutzung im Abfrageplan belegt statt Laufzeit geraten.
- Nur zwei Client Components; Filter, Chips und Blättern laufen über Adresse und Formular und funktionieren ohne JavaScript.
- 51 Tests, drei CI-Workflows, sechs Meilensteine mit Git-Tag und ausgeführtem Nachweis.

Stack: Next.js 16, React 19, TypeScript, Tailwind v4, Python 3.13, PostgreSQL mit PostGIS, Supabase, Vercel

ExtensionReach · Lead- und Outreach-Plattform

Solo-Entwicklung · Januar bis Mai 2026 · rund 47'000 Zeilen · privates Repository, Einsicht auf Anfrage

Internes Vertriebssystem mit Datenpipeline, Anreicherung und Deal-Tracking.

- Lead-Discovery über elf Quellen: LinkedIn (RapidAPI, Apify), Reddit, Hacker News, Substack, Medium sowie Community-Verzeichnisse und Foren, jeweils über die dafür vorgesehenen Schnittstellen und Anbieter.
- Betrieb gegen fremde Dienste unter realen Bedingungen: Rate-Limit-Behandlung, Tageslimits, gestaffelte Verzögerungen, Erkennung von Layout-Änderungen und Ausfalltoleranz; GitHub Actions als verteilte Cron-Läufer.
- Zehnstufige Pipeline mit Statusführung, Duplikatprüfung, Anreicherung über Apollo und Antwort-Klassifikation in sieben Kategorien.
- Drei KI-Agenten mit 31 Werkzeugen für Recherche, Anreicherung und Textentwurf, angebunden an OpenAI, Claude über AWS Bedrock und Perplexity.
- 15 serverseitige Funktionen mit 8'658 Zeilen, 38 Datenbankmigrationen, Telegram-Benachrichtigungen, eigene Supabase-Instanz.

Stack: React, TypeScript, Supabase/Postgres, Apify, RapidAPI, Chrome Extension MV3, GitHub Actions, OpenAI, AWS Bedrock, Perplexity, Telegram-API

Entrio · SaaS-Plattform mit Abomodel

Alleiniger Entwickler · Januar 2026 bis heute · rund 79'000 Zeilen im Hauptprodukt, vier Repositories · app.entrio.cloud

Web-App mit KI-Auswertung, Freemium-Abo und vier nativen Client-Anbindungen.

- Stripe seit März 2026 im Live-Modus: Checkout mit serverseitiger Preisauflösung, Webhooks mit persistenter Idempotenz, Subscription Schedules für geplante Downgrades, Dunning, Promo-Codes, 14-Tage-Trial ohne Karte.
- Authentifizierung mit eigenem OTP-Flow, Google OAuth 2.0, Passwort-Reset und Kontolöschung; sechs unterschiedliche Auth-Muster je nach Aufrufer.
- Postgres mit rund 40 Tabellen, 102 Row-Level-Security-Policies über fünf Rollenebenen, 78 Migrationen, zeitgesteuerte Jobs über pg_cron.
- Vier native Client-Anbindungen über MQL5, C# / .NET und Electron, jeweils mit Device-Auth nach RFC 8628 gegen dieselbe API.
- Benachrichtigungen über Resend mit Lifecycle-Mails, Idempotenz pro Typ und One-Click-Unsubscribe nach RFC 8058.
- Qualität: 314 Unit-Tests, 89 E2E-Fälle über drei Viewports, zwei selbst durchgeführte Sicherheitsaudits mit dokumentierten Fundstellen.

Stack: React, TypeScript, Vite, Tailwind, Supabase/Postgres, Deno Edge Functions, Stripe, OpenAI, Resend, Vercel, Electron, MQL5, C# / .NET

Gado Imkerei · Webshop mit Online-Zahlung

Umsetzung für eine Imkerei · Juli bis September 2026 · rund 8'000 Zeilen · gadoimkerei.ch

Website und Shop mit Online-Zahlung, von der Positionierung bis zur Übergabe in acht Wochen.

- Stripe Checkout mit Karte und TWINT, Betrag serverseitig aus dem Katalog nachgerechnet, Erstattungen über signierten Webhook.
- Row Level Security mit definiertem Anfangszustand: eine im Audit gefundene Lücke, über die sich Bestellungen als bezahlt einschleusen liessen, wurde geschlossen.
- Bestellmeldung per E-Mail und SMS, garantiert nur einmal pro Bestellung über eine atomare Datenbanksperre.
- Rechtstexte nach Schweizer Recht, Lebensmitteldeklaration vor dem Kauf, JSON-LD und Sitemap, Anleitungen für den Eigenbetrieb durch die Betreiberin.

Stack: React 19, TypeScript, Tailwind v4, Supabase, Stripe, Resend, Vercel

Baseform · Progressive Web App für iOS

Persönliches Projekt · seit Juli 2026 · rund 2'300 Zeilen React · Vorführung auf Anfrage

Offline-fähige App für Ernährung, Training und Erholung, ohne Server und ohne Konto.

- Vier externe Schnittstellen, jede mit Timeout und definierter Rückfallebene.
- Google Health API v4 über OAuth 2.0 mit Nur-Lese-Scopes, Token nur im Arbeitsspeicher, tolerantes Parsen schwach dokumentierter Antworten.
- Barcode- und QR-Scan im Browser über getUserMedia mit eigenem GS1-Digital-Link-Parser.
- Merge-Speicherung pro Tag gegen Datenverlust bei parallelen Fenstern, zeitzonenkorrekte Tagesschlüssel, Migration alter Formate.

Stack: React 18, Vite, PWA, ZXing, Google Health API, Anthropic API, Supabase mit Postgres, SVG, Vercel

geobeet · Geodaten-Anwendung in Python

Persönliches Projekt · Juli bis September 2026 · 946 Zeilen Python · Vorführung auf Anfrage
FastAPI-Anwendung, die amtliche Schweizer Geodaten verarbeitet und einen massstäblichen Plan erzeugt.

- 22 typisierte Endpunkte mit Pydantic-Schemata und automatischer OpenAPI-Dokumentation.
- Anbindung fremder Dienste über WMTS, WMS, WFS und STAC, inklusive Umgang mit undokumentiertem Verhalten der Schnittstellen.
- Koordinatentransformation nach LV95 mit gemessener Genauigkeit, Schema-Migration mit Datenerhalt und Fremdschlüsselprüfung.

Stack: Python 3.13, FastAPI, Pydantic, Shapely, pyproj, SQLite, Leaflet, SVG

Technologien

Frontend	Angular (Version 5 bis 18) mit Kendo UI und RxJS, vier Jahre beruflich, React 18/19, TypeScript, JavaScript, HTML5, CSS3, SASS, Bootstrap, Tailwind CSS, Vite, PWA und Offline-Betrieb, Electron, mobile-first. Next.js 16 mit App Router, statischer Erzeugung und Erneuerung im Hintergrund.
Backend und API	Node.js und Deno (TypeScript), Python mit FastAPI, C# und .NET, REST-Entwurf, Webhooks, Rate Limiting, Idempotenz
Daten	PostgreSQL, Supabase, Row Level Security, PostGIS mit GiST-Index, Abfragepläne lesen, Microsoft SQL Server mit T-SQL und Stored Procedures, Datenmodellierung bis 3NF, Migrationen, JSONB, Trigger, pg_cron, SQLite
KI-Integration	OpenAI, Anthropic Claude (direkt und über AWS Bedrock), Perplexity, Agenten mit Werkzeugaufrufen, strukturierte Prompts, Kosten- und Fehlerbehandlung, eigene Agenten-Werkzeuge und Wissensbasis für Recherche und Entscheidungsdokumentation
Integration und Automation	Playwright, Apify, RapidAPI, Chrome Extensions (Manifest V3), zeitgesteuerte Pipelines, Rate-Limit- und Retry-Strategien, Duplikaterkennung, Ausfalltoleranz bei fremden Schnittstellen
Zahlung und Auth	Stripe (Checkout, Webhooks, Subscription Schedules, Dunning), OAuth 2.0 mit Authorization-Code- und Client-Credentials-Flow, Google Auth, Device-Auth nach RFC 8628, OTP, JWT, HMAC
Betrieb	Vercel, GitHub Actions, Azure DevOps, Docker mit Multi-Stage-Build und Nginx, CI/CD, Code Signing, Git mit Feature-Branches, Resend, Telegram-API
Qualität	Vitest, pytest, Playwright E2E, API-Vertragstests, Sicherheitsaudits, SonarQube, SAST und Dependency-, Secret- und Container-Scans in der Pipeline, Conventional Commits mit Husky und commitlint, CSP und HSTS, ESLint und tsc als Build-Gate

Berufserfahrung

August 2022 bis August 2026
Applikationsentwickler EFZ (Berufslehre) · Kull AG, Schweiz

- Weiter- und Neuentwicklung firmeninterner Applikationen in Angular mit Kendo UI und TypeScript sowie ASP.NET Core im Backend.
- Datenbankentwicklung und -design mit Microsoft SQL Server, Anpassung von SSRS-Reports, Versionskontrolle mit Git und Azure DevOps.
- Wartung eines produktiven Fakturamoduls mit Fehleranalyse, SQL-Deployments und Stage-Releases.
- Eigenverantwortliche Bearbeitung ganzer Module von der Analyse über die Umsetzung bis zur Fehlerbehebung.
- Abschluss im August 2026, seither hauptberuflich in der eigenen Produktentwicklung.

Lizenzverwaltung für Softwarelizenzen · Abschlussarbeit, März bis April 2026

- Neues Modul in einer internen Unternehmenssoftware für Zeiterfassung und Fakturierung, das Lizenzdaten über die REST-API des Lieferanten importiert, Endkunden auf interne Kundenstämme abbildet und die Daten in den Fakturierungslauf überführt.
- API-Anbindung mit OAuth 2.0, Rate-Limit-Behandlung mit Retry und Abbildung der JSON-Antworten auf das interne Datenmodell.
- Dreistufiges Kunden-Matching (exakt, Fuzzy-Vorschlag per Stored Procedure mit Score, manuelle Zuweisung); jede manuelle Entscheidung wird persistiert und beim nächsten Import automatisch angewendet.
- Fünf neue Tabellen bis 3NF normalisiert, Bulk-Verarbeitung in T-SQL, Frontend in Angular mit Virtual Scrolling für grosse Datenmengen.

Stack: Angular, TypeScript, C# / .NET, Microsoft SQL Server, REST mit OAuth 2.0

Bewerbungsplattform mit CI/CD-Pipeline · Mai bis August 2026

- Web-Anwendung für Lehrstellenbewerbungen: Anforderungen mit den Auftraggebern geklärt, Datenmodell, Zeitplan und Umsetzung inklusive Auslieferung.
- Vollständige CI/CD-Pipeline auf Azure DevOps: Linting, Typprüfung, Tests mit Coverage-Gate, statische Codeanalyse mit SonarQube sowie Dependency-, Secret- und Container-Scan vor dem Deploy.
- Docker mit Multi-Stage-Build und Nginx, automatische Versionierung, Deploy nur aus dem Hauptbranch, wöchentliche Abhängigkeitspflege.

Stack: React 19, TypeScript, Tailwind, .NET, Docker, Nginx, Azure DevOps

Ausbildung

2022 bis 2026
Informatiker EFZ, Fachrichtung Applikationsentwicklung
Berufsfachschule Baden · Abschluss August 2026

Arbeitsweise

Jedes eigene Projekt ist dokumentiert: Architektur, technische Kernprobleme, Datenmodell, offene Punkte und die Begründung der Entscheidungen samt verworfener Alternativen. Das ersetzt einen Grossteil der üblichen Rückfragen zum Leistungsstand. Projektübersicht und die vollständigen Dokumentationen als PDF auf gabrielsuter.dev, der Quellcode von Renditeradar öffentlich auf GitHub. Arbeiten aus dem Anstellungsverhältnis unterliegen der Geheimhaltung. Zu Vorgehen, Architektur und meiner Rolle gebe ich dort gerne im Gespräch Auskunft, Code und Dokumente gebe ich nicht weiter.

Sprachen

Deutsch: Muttersprache
Englisch: B2, Projektdokumentation und Fachliteratur

Interessen

Finanzmarkt-Analyse und algorithmisches Trading, fachliche Grundlage von Entrio
Kraft- und Ausdauertraining, Tech-Meetups und Netzwerkanlässe

Referenzen

Manuel Patrone, Head of Application Development, Kull AG: vier Jahre direkte Zusammenarbeit. Kontakt und Arbeitszeugnis auf Anfrage.