



Gabriel Suter

Full-Stack Developer · Swiss Federal VET Diploma in IT (EFZ), Application Development

kontakt@gabrielsuter.dev

+41 78 257 32 95

Däniken SO, Switzerland

gabrielsuter.dev/en

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

Available: immediately, 100%

Location: 100% remote, Switzerland

Billing: hourly or milestone-based

Languages: German (native), English B2

Profile

Full-stack developer with four years of professional experience in Angular, TypeScript, C# and Microsoft SQL Server at a Microsoft partner, plus own products built with React, Node and Python. Focus on integration work, data pipelines and SaaS products: resilient integrations with third-party services, Stripe subscription models with freemium logic, Postgres architectures with Row Level Security. The projects listed below were built solo, from concept through backend and frontend to operation, including security audits and handover documentation.

180,000

lines of own code across five projects,
excluding dependencies and translations

47

server-side functions in production (Deno)

15

third-party APIs in production, from Stripe
to swisstopo

4 years

professional experience with Angular,
TypeScript, C# and SQL Server

Projects

Renditeradar · Aggregator for Investment Properties

Personal project · September 2026 · approx. 3,500 lines · public repository under MIT, live demo

Next.js application with statically generated detail pages and radius search via a spatial index; work sample using generated property data.

- 400 detail pages statically generated with background revalidation, listing rendered dynamically from search parameters; Lighthouse 100 in all four categories on the detail page.
- Python pipeline that normalises messy raw data, detects duplicates by hash and performs idempotent upserts; a second run changes no rows.
- PostgreSQL with composite indexes, Row Level Security as write protection and PostGIS: radius and nearest-neighbour search via GiST index, index usage verified in the query plan rather than guessed from timings.
- Only two Client Components; filters, chips and pagination run via URL and form and work without JavaScript.
- 51 tests, three CI workflows, six milestones, each with a Git tag and executed proof.

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

ExtensionReach · Lead and Outreach Platform

Solo development · January to May 2026 · approx. 47,000 lines · private repository, access on request

Internal sales system with data pipeline, enrichment and deal tracking.

- Lead discovery across eleven sources: LinkedIn (RapidAPI, Apify), Reddit, Hacker News, Substack, Medium as well as community directories and forums, each via the interfaces and providers intended for that purpose.
- Operation against third-party services under real-world conditions: rate-limit handling, daily quotas, staggered delays, detection of layout changes and fault tolerance; GitHub Actions as distributed cron runners.
- Ten-stage pipeline with status tracking, duplicate checks, enrichment via Apollo and reply classification into seven categories.
- Three AI agents with 31 tools for research, enrichment and copy drafting, connected to OpenAI, Claude via AWS Bedrock and Perplexity.
- 15 server-side functions with 8,658 lines, 38 database migrations, Telegram notifications, dedicated Supabase instance.

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

Entrio · SaaS Platform with Subscription Model

Sole developer · January 2026 to present · approx. 79,000 lines in the main product, four repositories · app.entrio.cloud

Web app with AI analysis, freemium subscription and four native client integrations.

- Stripe live since March 2026: Checkout with server-side price resolution, webhooks with persistent idempotency, Subscription Schedules for planned downgrades, dunning, promo codes, 14-day trial without a card.
- Authentication with a custom OTP flow, Google OAuth 2.0, password reset and account deletion; six distinct auth patterns depending on the caller.
- Postgres with approx. 40 tables, 102 Row Level Security policies across five role levels, 78 migrations, scheduled jobs via pg_cron.
- Four native client integrations via MQL5, C# / .NET and Electron, each using device authorization per RFC 8628 against the same API.
- Notifications via Resend with lifecycle emails, per-type idempotency and one-click unsubscribe per RFC 8058.
- Quality: 314 unit tests, 89 E2E cases across three viewports, two self-conducted security audits with documented findings.

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

Gado Imkerei · Online Shop with Payments

Built for a beekeeping business · July to September 2026 · approx. 8,000 lines · gadoimkerei.ch

Website and shop with online payment, from positioning to handover in eight weeks.

- Stripe Checkout with card and TWINT, amount recalculated server-side from the catalogue, refunds via signed webhook.
- Row Level Security with a defined initial state: a gap found in the audit, through which orders could be injected as paid, was closed.
- Order notification by email and SMS, guaranteed to be sent only once per order through an atomic database lock.
- Legal texts under Swiss law, food labelling shown before purchase, JSON-LD and sitemap, guides for independent operation by the owner.

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

Baseform · Progressive Web App for iOS

Personal project · since July 2026 · approx. 2,300 lines of React · demo on request

Offline-capable app for nutrition, training and recovery, with no server and no account.

- Four external interfaces, each with a timeout and a defined fallback.
- Google Health API v4 via OAuth 2.0 with read-only scopes, tokens held in memory only, tolerant parsing of poorly documented responses.
- In-browser barcode and QR scanning via getUserMedia with a custom GS1 Digital Link parser.
- Per-day merge storage to prevent data loss across parallel windows, time-zone-correct day keys, migration of legacy formats.

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

geobeet · Geodata Application in Python

Personal project · July to September 2026 · 946 lines of Python · demo on request
FastAPI application that processes official Swiss geodata and generates a to-scale plan.

- 22 typed endpoints with Pydantic schemas and automatic OpenAPI documentation.
- Integration of third-party services via WMTS, WMS, WFS and STAC, including handling of undocumented interface behaviour.
- Coordinate transformation to LV95 with measured accuracy, schema migration with data preservation and foreign key checks.

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

Technologies

Frontend	Angular (versions 5 to 18) with Kendo UI and RxJS, four years professionally, React 18/19, TypeScript, JavaScript, HTML5, CSS3, SASS, Bootstrap, Tailwind CSS, Vite, PWA and offline operation, Electron, mobile-first. Next.js 16 with App Router, static generation and background revalidation.
Backend and API	Node.js and Deno (TypeScript), Python with FastAPI, C# and .NET, REST design, webhooks, rate limiting, idempotency
Data	PostgreSQL, Supabase, Row Level Security, PostGIS with GiST index, reading query plans, Microsoft SQL Server with T-SQL and stored procedures, data modelling up to 3NF, migrations, JSONB, triggers, pg_cron, SQLite
AI Integration	OpenAI, Anthropic Claude (direct and via AWS Bedrock), Perplexity, agents with tool calling, structured prompts, cost and error handling, custom agent tooling and knowledge base for research and decision records
Integration and Automation	Playwright, Apify, RapidAPI, Chrome Extensions (Manifest V3), scheduled pipelines, rate-limit and retry strategies, duplicate detection, fault tolerance with third-party interfaces
Payments and Auth	Stripe (Checkout, Webhooks, Subscription Schedules, Dunning), OAuth 2.0 with Authorization Code and Client Credentials flows, Google Auth, device authorization per RFC 8628, OTP, JWT, HMAC
Operations	Vercel, GitHub Actions, Azure DevOps, Docker with multi-stage build and Nginx, CI/CD, code signing, Git with feature branches, Resend, Telegram API
Quality	Vitest, pytest, Playwright E2E, API contract tests, security audits, SonarQube, SAST and dependency, secret and container scans in the pipeline, Conventional Commits with Husky and commitlint, CSP and HSTS, ESLint and tsc as build gate

Professional Experience

August 2022 to August 2026

Application Developer, Apprenticeship (EFZ) · Kull AG, Switzerland

- Enhancement and new development of internal business applications in Angular with Kendo UI and TypeScript, with ASP.NET Core in the backend.
- Database development and design with Microsoft SQL Server, customisation of SSRS reports, version control with Git and Azure DevOps.
- Maintenance of a production invoicing module, including error analysis, SQL deployments and stage releases.
- Independent ownership of entire modules, from analysis through implementation to bug fixing.
- Graduated in August 2026, since then working full-time on my own product development.

Software Licence Management · Final Apprenticeship Project, March to April 2026

- New module in an internal business application for time tracking and invoicing that imports licence data via the vendor's REST API, maps end customers to internal customer records and feeds the data into the invoicing run.
- API integration with OAuth 2.0, rate-limit handling with retry, and mapping of JSON responses to the internal data model.
- Three-stage customer matching (exact, fuzzy suggestion with score via stored procedure, manual assignment); every manual decision is persisted and applied automatically on the next import.
- Five new tables normalised to 3NF, bulk processing in T-SQL, Angular frontend with virtual scrolling for large data volumes.

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

Apprenticeship Application Platform with CI/CD Pipeline · May to August 2026

- Web application for apprenticeship applications: requirements clarified with the stakeholders, data model, schedule and implementation through to delivery.
- Complete CI/CD pipeline on Azure DevOps: linting, type checking, tests with coverage gate, static code analysis with SonarQube, and dependency, secret and container scans before deployment.
- Docker with multi-stage build and Nginx, automatic versioning, deployment from the main branch only, weekly dependency maintenance.

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

Education

2022 to 2026

Swiss Federal VET Diploma in IT (Informatiker EFZ), Application Development

Berufsfachschule Baden (vocational school) · graduated August 2026

Working Approach

Every one of my projects is documented: architecture, core technical problems, data model, open issues and the reasoning behind decisions, including rejected alternatives. This answers most of the usual questions about my capabilities up front. Project overview and complete documentation as PDF on gabrielsuter.dev, source code of Renditeradar public on GitHub. Work from my employment is subject to confidentiality. I am happy to talk about the approach, architecture and my role in an interview, but I do not share code or documents.

Languages

German: native
English: B2, project documentation and technical literature

Interests

Financial market analysis and algorithmic trading, the domain behind Entrio
Strength and endurance training, tech meetups and networking events

References

Manuel Patrone, Head of Application Development, Kull AG: four years of direct collaboration. Contact details and employment reference (Arbeitszeugnis) on request.