

ABDULLAH CHAUDARY

Senior Full-Stack Engineer | Systems & DevSecOps Architect | Tech Lead | AI/ML Engineering

> **I architected the AI product that became the main selling point of a $600M+ acquisition.**

For twelve-plus years I have built and secured production platforms end to end: writing the proposal that wins the client, designing the system, choosing and configuring the hardware, building the product, securing it, deploying it and keeping it up. The work spans web and mobile SaaS for US enterprise clients on AWS, GCP, Azure and self-hosted infrastructure, from Next.js, Nuxt.js and React Native front ends through Django, FastAPI and Express.js services to Kubernetes and Terraform. I have led teams of up to 20, run hybrid infrastructure at 99.95% uptime, and shipped LLM, RAG and agentic automation into production.

Lahore, Pakistan abchaudary@outlook.com linkedin.com/in/abchaudary abchaudary.me
github.com/abdullahchaudary

AT A GLANCE

$600M+
Acquisition where the AI product I architected was the main selling point

99.95%
Uptime on hybrid on-prem and cloud infrastructure

~70%
Lower infrastructure cost after an AWS to self-hosted migration, zero downtime

45%
Lower monthly cloud spend across production environments

80%
Fewer critical vulnerabilities within the first two quarters of hardening

Up to 10x
Client revenue growth on key initiatives

20
Largest cross-functional team led, across 5 concurrent products

12+
Years building and securing production platforms

INDUSTRIES SERVED

pharma, marketing research, healthcare, EdTech, FinTech, logistics, real estate, government, business tools, and related.

CONTENTS

1 Part 1. Products and platforms I built

2 Part 2. Automations

3 Part 3. Client engagements

4 Part 4. Earlier professional work

5 Part 5. Creative and open coding projects

6 Part 6. Experience, education and skills

Technical leadership across teams, clients and delivery.

Cross-functional team leadership.	Led development, design and marketing teams of up to 20, owning hiring, mentorship and performance; chaired management meetings and kept stakeholders aligned. Still leading engineering teams today.
Pre-sales and proposals.	Won clients by writing the technical proposals that closed the deals, including the one behind the $600M+ acquisition, which is still attached to my NDA as Exhibit A. Owned client onboarding end to end and led client meetings throughout delivery.
Delivery and engineering management.	Sprint and task management in Jira; full GitLab and GitHub governance (repos, access, branches, code reviews, merges); team-wide engineering standards through an AGENTS.md rulebook and AI-coding tool configurations.
Creative direction.	Drove UI/UX and creative direction hand in hand with designers.
Ownership and reliability.	Single-handed ownership of servers and infrastructure, with zero downtime.
Business impact.	Cut infrastructure costs about 70% and helped grow client revenue streams up to 10x on key initiatives.



PRODUCTS AND PLATFORMS I BUILT

Live products and platforms, each designed, built and deployed by me, from architecture to production infrastructure. Screenshots come from the production sites.

1.1	DrieVerse AI	a cross-platform AI assistant on seven model providers	ai.drieverse.com
1.2	Accountaire FinTech	a multi-currency money app with a confirm-before-write AI assistant	app.accountaire.com
1.3	SkillsVerx	an online academy with its own website, CMS and learning platform	skillsverx.com
1.4	SEO Toolkit SaaS	white-label Local SEO audits for agencies, generated live	seotools.abchaudary.me
1.5	Accountaire	64 static pages migrated to a 74-page, five-market platform	accountaire.com
1.6	DrieVerse	a content platform with publish-time rule enforcement	drieverse.com
1.7	LeadSyft	a Next.js and Payload platform with publish-to-live in seconds	leadsyft.com
1.8	DevOdin	LCP 2.19 s to 1.03 s on an agency rebuild	devodin.com
1.9	Exact Hardware	a server-hardware store stocked by four distributor scrapers	exacthardware.com
1.10	Self-hosted platform	47 containers hosting 10 sites, run by one engineer	
1.11	abchaudary.me	a portfolio, blog and content platform built as a production system	abchaudary.me

DrieVerse AI

A cross-platform AI assistant on seven model providers

INDUSTRIES AI productivity · business tools · meetings and knowledge work

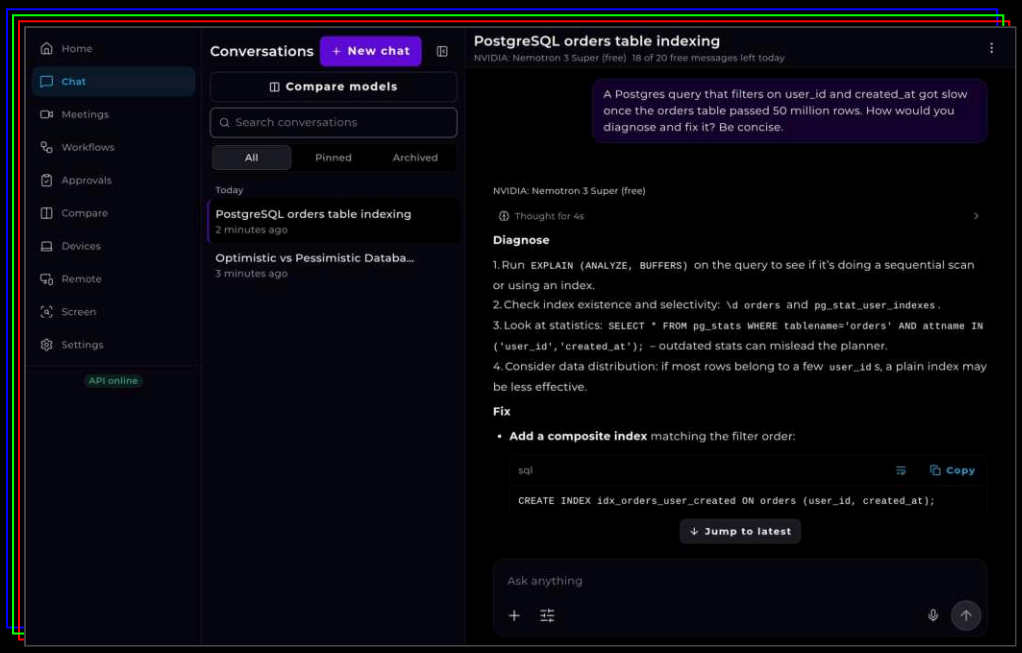
https://ai.drieverse.com ↗

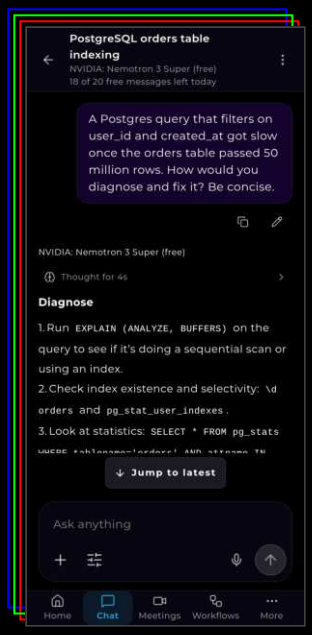
WHAT IT IS

A production-grade AI assistant that runs as one application on web, desktop (Windows, macOS, Linux) and phone (iOS, Android), and connects to the frontier model of your choice. It includes a meetings assistant, screen context, dictation, remote control and agentic workflows.

MY ROLE

Architecture, full stack, infrastructure, security and the AI engineering process: one engineer directing AI coding agents through documented phases, written decisions and independent security reviews.





SCALE

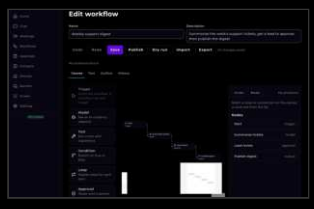
318,000 about 318,000 lines of application code	**399,000** lines of tests	**4 + 13** 4 apps and 13 shared packages	**43** architecture decision records
11,715 API unit tests	**1,488** integration tests	**1,482** Playwright tests passing at the last phase merge	**5** written security reviews
90.6% coverage (API)	**84.5%** coverage (web)	**98%+** coverage (platform, desktop, providers)	**170 KB** every route under 170 KB of JavaScript, enforced in CI



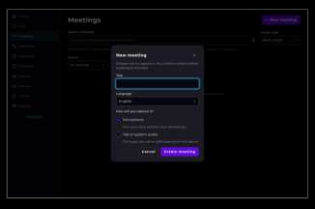
MODEL COMPARISON



CANVAS

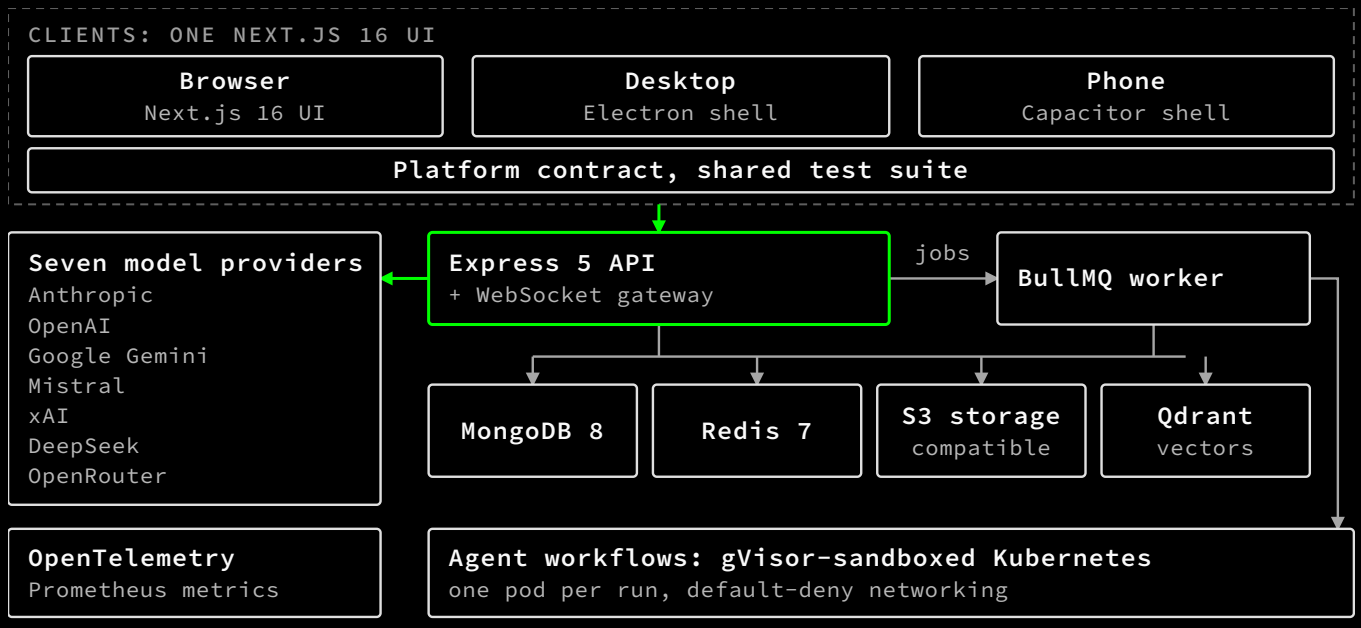


WORKFLOW EDITOR



MEETINGS

ARCHITECTURE



REQUEST PATH: CLIENT TO API TO MODEL PROVIDER

- One Next.js 16 UI that runs in the browser, inside an Electron desktop shell and inside a Capacitor mobile shell; native capabilities sit behind one platform contract with a shared test suite.
- Express 5 API with a WebSocket gateway, plus a BullMQ worker; MongoDB 8 (JSON Schema validation, numbered migrations, no ODM), Redis 7, S3-compatible storage, Qdrant for vector search.
- Seven model providers through official SDKs and one OpenAI-compatible adapter: Anthropic, OpenAI, Google Gemini, Mistral, xAI, DeepSeek and OpenRouter, with free-model fallback chains and per-user, global and per-feature allowances.
- Agent workflows run in one gVisor-sandboxed Kubernetes pod per run with default-deny networking.
- Distroless, read-only containers; OpenTelemetry and Prometheus metrics.

HIGHLIGHTS

- Assistant chat with streaming reasoning, side-by-side model comparison, resumable chunked uploads, a document and code canvas with versions, memory, and cost tracking.
- Meetings: live transcription with speakers, consent before capture, summaries, action items and semantic search.
- Screen context with on-device OCR redaction and a mandatory preview; a CI test proves zero network connections before Send and exactly one upload after it.
- Remote control from phone to desktop over end-to-end encrypted WebRTC, with every assistant-driven click approved on a trusted device.
- Visual workflow editor with schedule, webhook and event triggers, an approvals inbox and an allow-listed MCP client.
- Passkeys, OAuth, MFA, workspaces, audit log, account export and deletion.

STACK Next.js 16 · React 19 · Tailwind 4 · Electron · Capacitor · Express 5 · BullMQ · MongoDB 8 · Redis 7 · Qdrant · MinIO/S3 · Kubernetes + gVisor · Anthropic, OpenAI, Gemini, Mistral, xAI, DeepSeek, OpenRouter · MCP · Playwright · Vitest

Accountaire FinTech

A multi-currency money app with a confirm-before-write AI assistant

INDUSTRIES FinTech · personal finance · small-business accounting

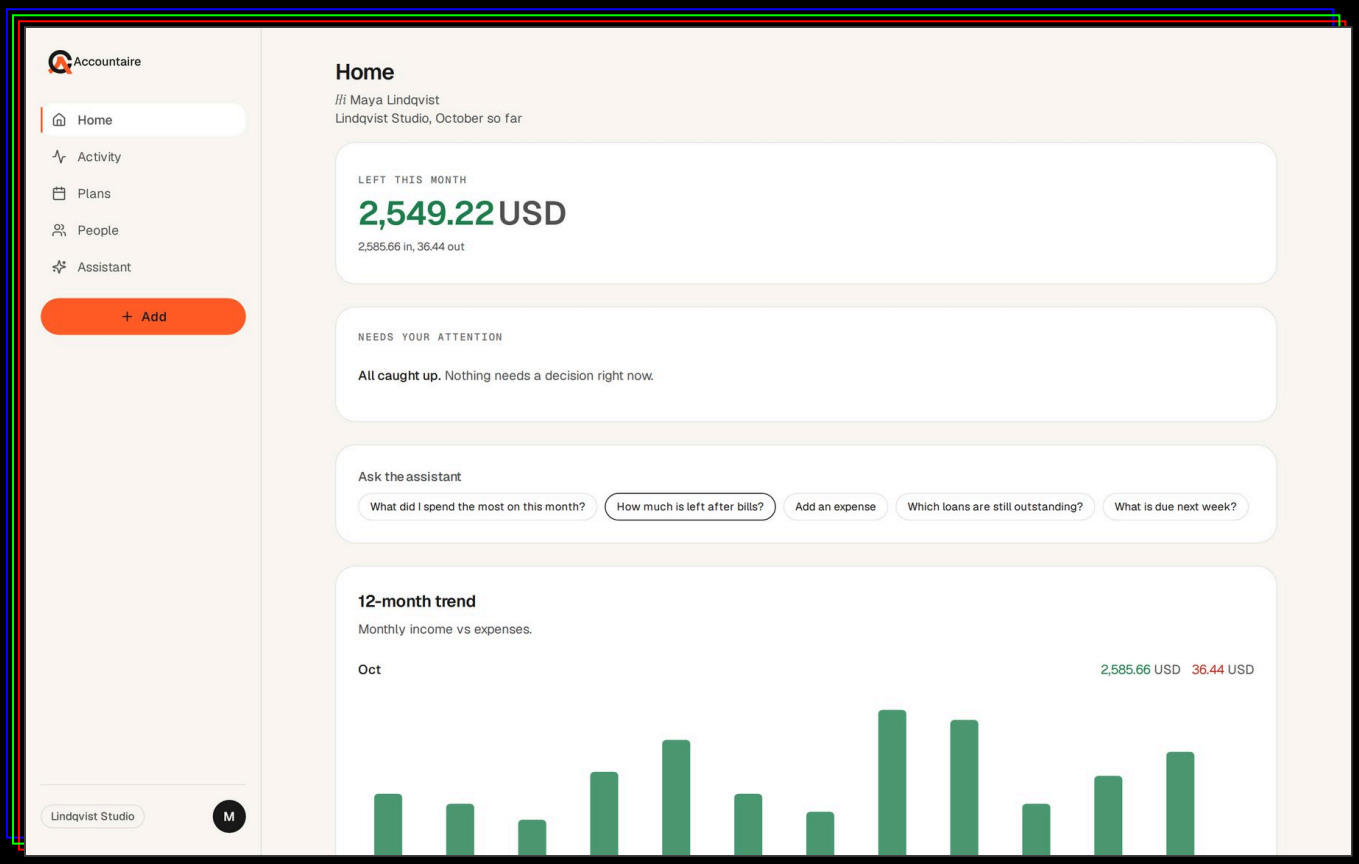
https://app.accountaire.com ↗

WHAT IT IS

One money app for individuals and small businesses. Every feature writes into a single multi-currency ledger: income and expenses, receipt scanning, bills, installment plans, lending and borrowing by person, bank activity imported from forwarded emails, cash-flow forecasts, and an AI assistant that reads the ledger and proposes changes the user confirms.

MY ROLE

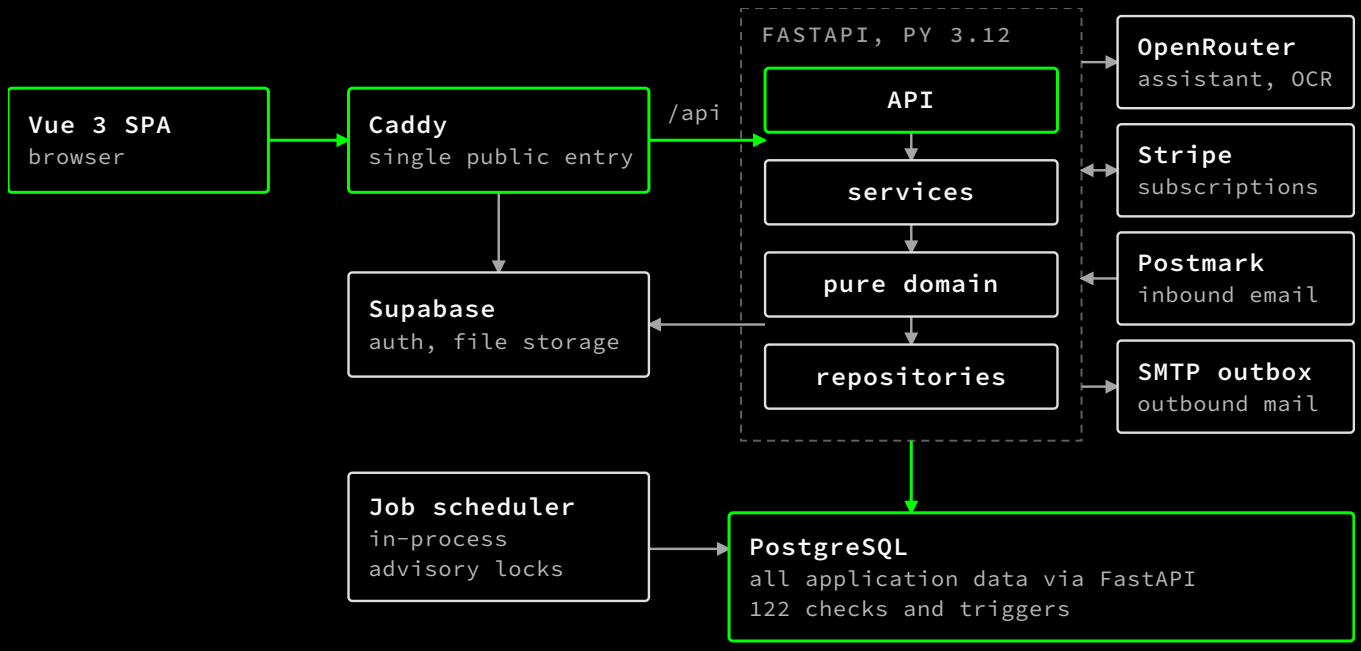
Product architecture, full stack, data model, AI design, infrastructure and CI.



HIGHLIGHTS

- Money is stored as integer minor units with ISO-4217 codes, never floats; FX rates are frozen at the transaction date.
- Key money rules are enforced three times: in pure domain code, in the service transaction and as 122 database checks and triggers.
- AI assistant with 31 tools: 16 read tools and 15 propose-then-confirm write tools; the model never does arithmetic.
- Every AI call is metered for cost; admin access to user data goes to an append-only audit log.
- CI regenerates the API contract and fails on any drift between backend and generated front-end types.

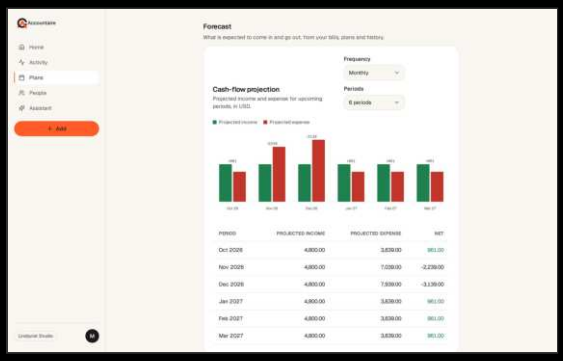
ARCHITECTURE



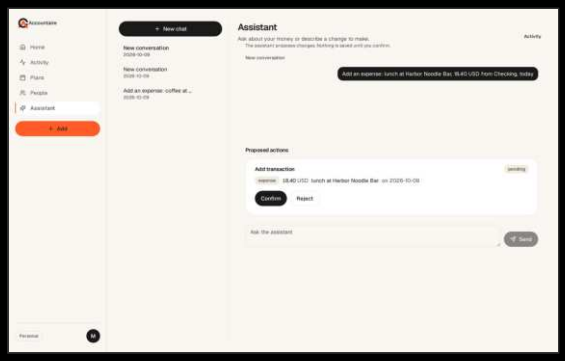
Background jobs: FX rates, recurring bills, reminders, email delivery
each runs exactly once

REQUEST PATH: BROWSER TO CADDY TO FASTAPI TO POSTGRESQL

- Vue 3 single-page app behind Caddy as the single public entry; FastAPI (Python 3.12, asyncpg, no ORM) in strict layers: API, services, pure domain, repositories.
- Self-hosted Supabase for authentication and file storage; all application data goes through FastAPI.
- OpenRouter for the assistant (tool calling) and receipt OCR (vision); Stripe for trials and subscriptions; Postmark for inbound bank emails; SMTP outbox for outbound mail.
- Background jobs (FX rates, recurring bills, reminders, email delivery) on an in-process scheduler with Postgres advisory locks, so each runs exactly once.



PLANS WITH FORECAST



ASSISTANT CONFIRMATION CARD



HOME, PHONE

SCALE

About 88,700 lines of source; 132 API endpoints; 21 migrations; 1,072 Python tests, 1,111 front-end unit tests and Playwright end-to-end tests across six CI jobs.

STACK Vue 3 · Vite · shadcn-vue · Tailwind · FastAPI · Python 3.12 · asyncpg · PostgreSQL · self-hosted Supabase · Caddy · OpenRouter · Stripe · Postmark · Turborepo · GitHub Actions

SkillsVerx

An online academy with its own website, CMS and learning platform

INDUSTRIES EdTech · professional training · certification

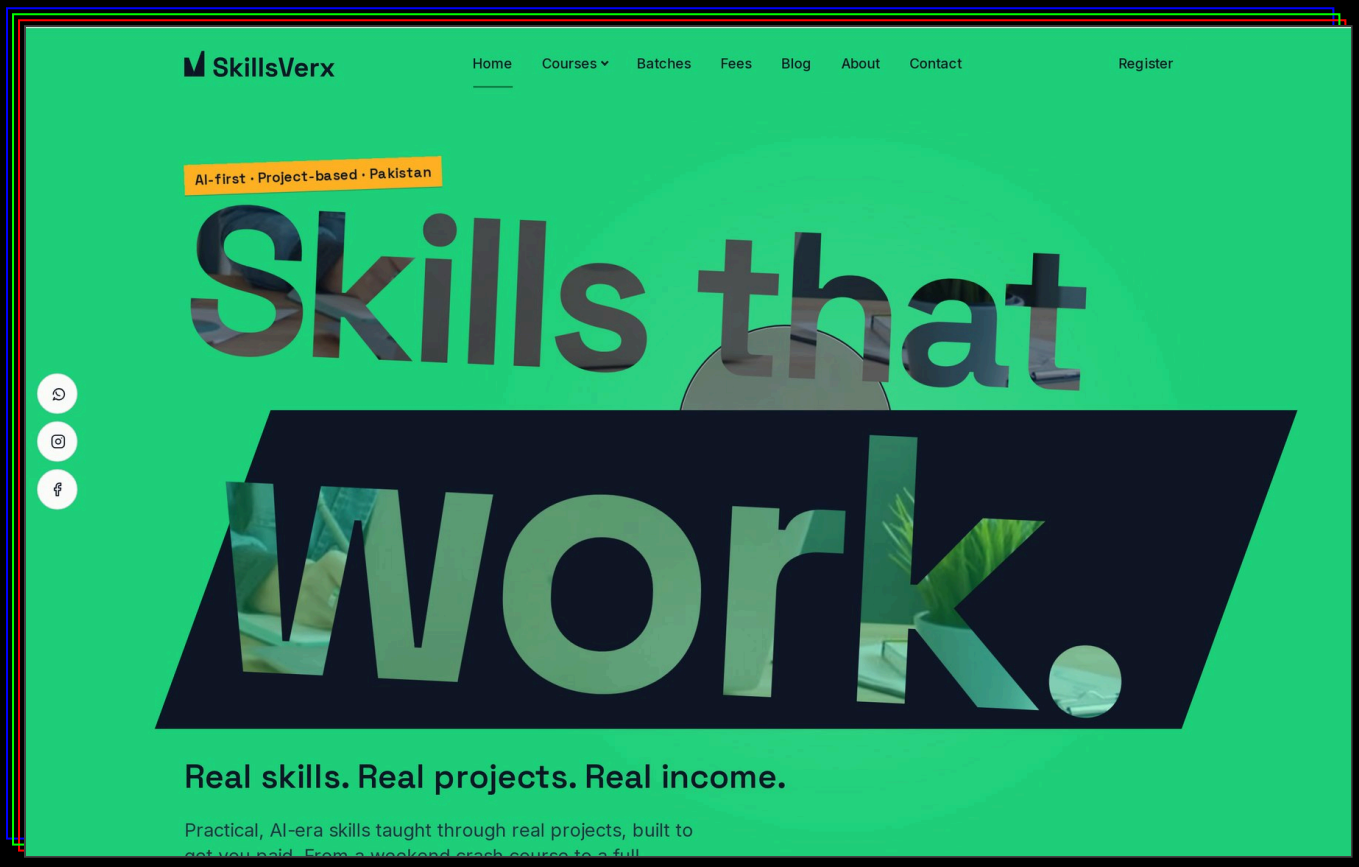
https://skillsverx.com ↗ https://lms.skillsverx.com ↗

WHAT IT IS

An online academy teaching practical, AI-first digital skills: 8 courses in 5 tiers, from crash courses to bootcamps, with live classes and certificates.

MY ROLE

Designed, built and hosted the website, CMS and the complete learning platform.

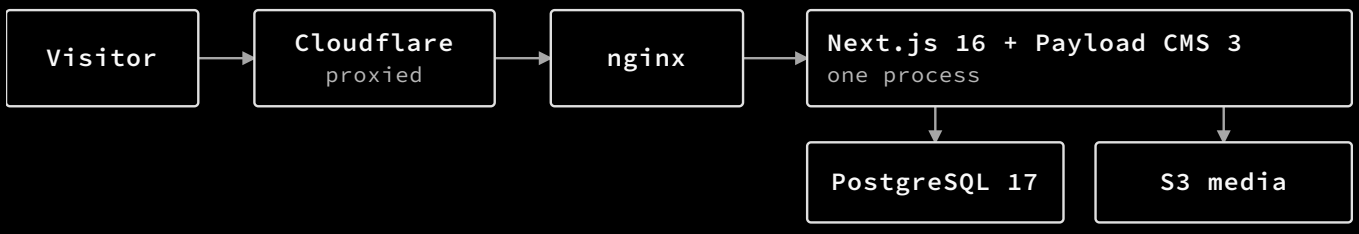


HIGHLIGHTS

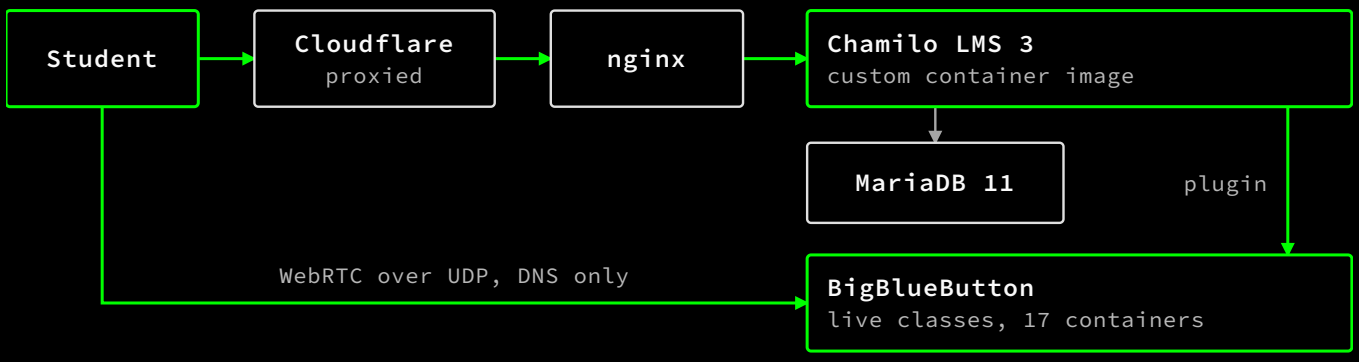
- 40 priced course offerings from one source of truth; registration with a live server-checked price card.
- Discount coupons with caps, scopes and limits, redeemed race-safely with a Postgres row lock.
- Certificate verification with collision-safe IDs, rate limiting and no public data exposure.
- Glossary, blog with 8 category hubs, llms.txt and a structured-data audit score of 97.
- Motion system with smooth scroll, reveals and a pinned horizontal process track, with reduced-motion fallbacks.

ARCHITECTURE

WEBSITE AND CMS



LEARNING PLATFORM

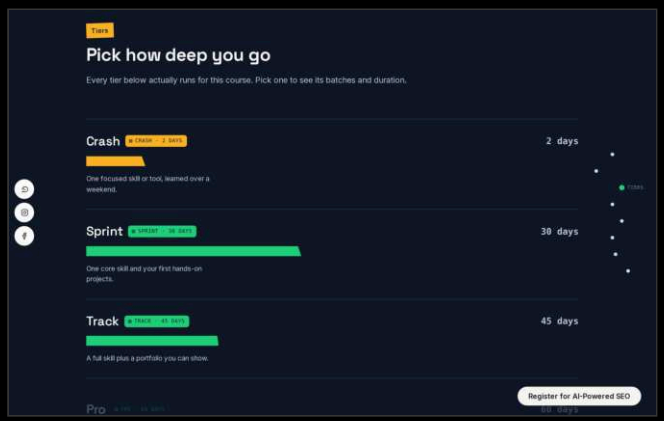


LEARNING PATH: STUDENT TO CHAMILO LMS TO BIGBLUEBUTTON LIVE CLASSES

- Website and CMS: Next.js 16 with Payload CMS 3 in the same process, PostgreSQL 17, S3-compatible media.
- Learning platform: Chamilo LMS 3 in a custom container image (no official Chamilo 3 image exists), MariaDB 11.
- Live classes: a 17-container BigBlueButton stack, integrated through Chamilo's conferencing plugin. BigBlueButton was chosen over Jitsi because it is the engine that plugin actually drives.
- Cloudflare in front of the site and LMS; BigBlueButton DNS-only because WebRTC needs UDP.



COURSE PAGE



COURSE PAGE WITH TIERS

SCALE

About 54,400 lines of source; 52 test files with 329 test cases; 19 CMS collections; 23 migrations; 149 URLs.

STACK Next.js 16 · Payload CMS 3 · PostgreSQL 17 · Tailwind 4 · Chamilo LMS 3 · PHP 8.3 · MariaDB 11 · BigBlueButton 3 · Docker · nginx · Cloudflare

SEO Toolkit SaaS

White-label Local SEO audits for agencies, generated live

INDUSTRIES marketing agencies · local SEO · SaaS

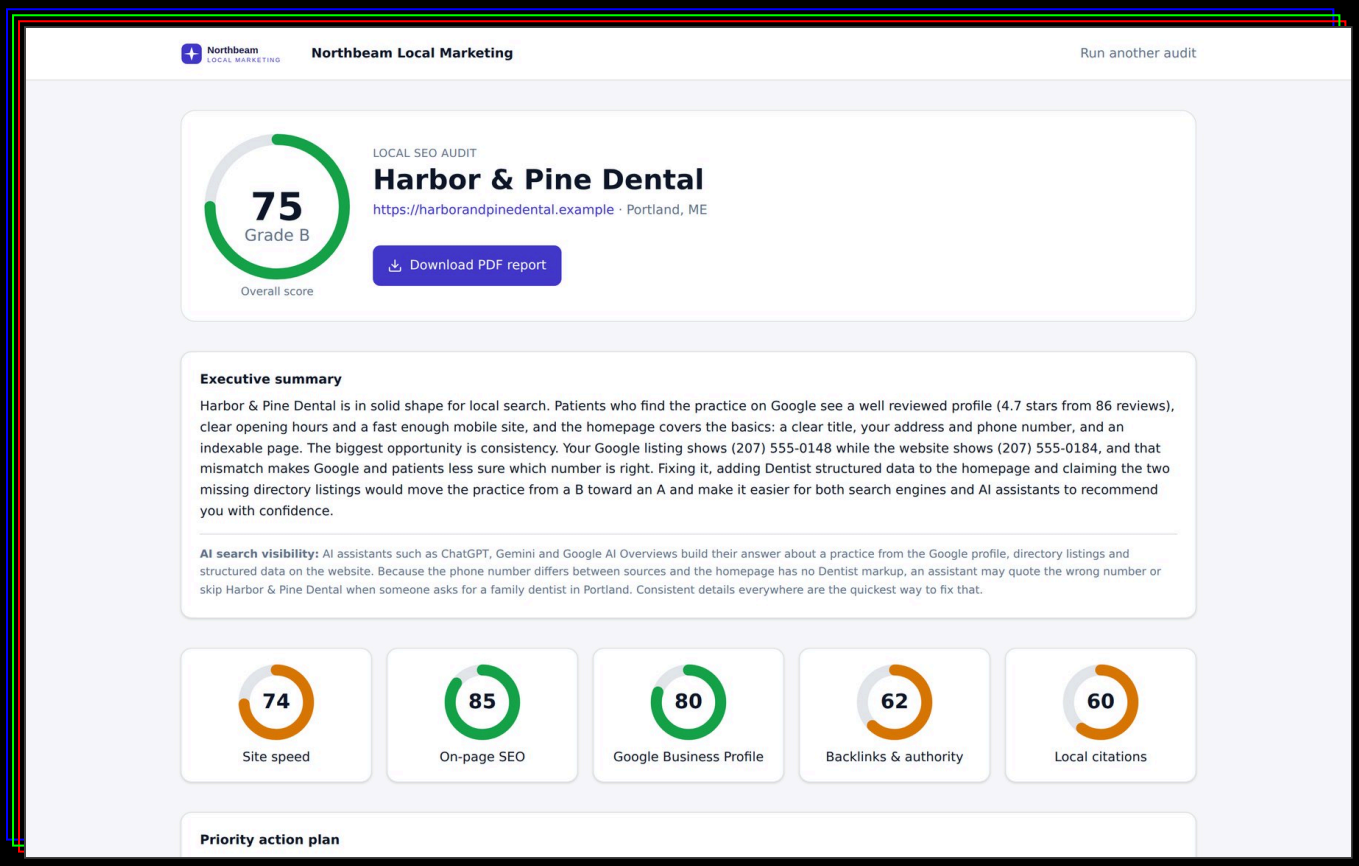
https://seotools.abchaudary.me ↗

WHAT IT IS

A multi-tenant SaaS for agencies. A prospect enters a business name and website on the agency's white-labelled page and gets a scored Local SEO report and a branded PDF; the agency captures the lead.

MY ROLE

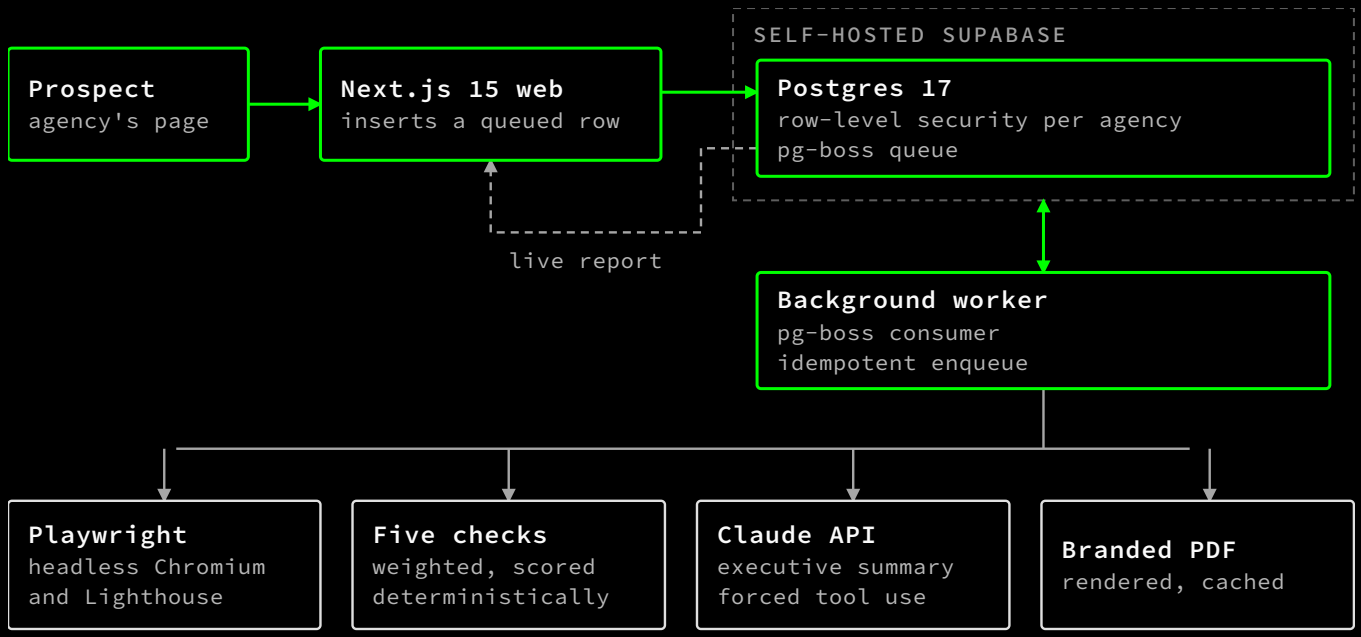
Product design, architecture and full build.



HIGHLIGHTS

- Five weighted checks: Google Business Profile, site speed, on-page, backlinks and citations; a failed check is excluded and the weights renormalise, so an outage degrades a report instead of tanking it.
- Reports fill in live as each check finishes; NAP consistency cross-checked between the listing and the site.
- Owner admin for agencies, users, audits, leads, feature flags and events; CSV lead export; per-IP and per-agency limits.

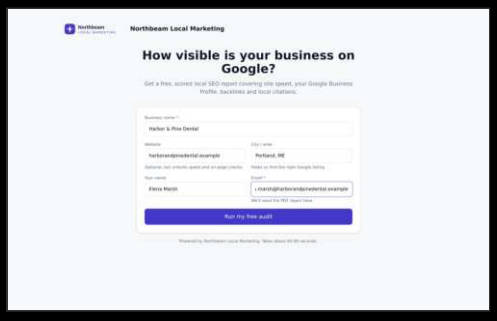
ARCHITECTURE



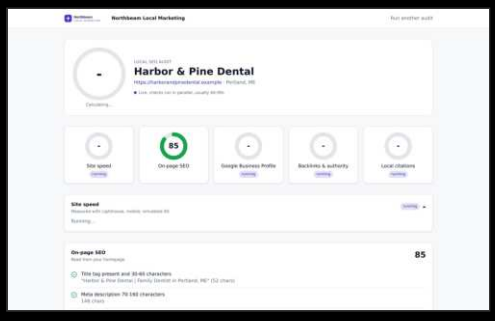
each audit runs locally in the worker, against the prospect's site

AUDIT PATH: PROSPECT TO WEB APP, QUEUE, WORKER

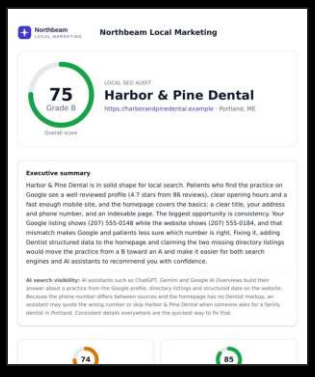
- Turborepo with a Next.js 15 web app and a background worker; self-hosted Supabase on Postgres 17 with row-level security per agency.
- pg-boss queue on the same Postgres instead of Redis; the web app only inserts a queued row and the worker enqueues idempotently.
- The worker runs headless Chromium (Playwright) and Lighthouse locally for each audit, then renders and caches the PDF.
- Claude writes the executive summary through forced tool use with schema validation; all scoring stays deterministic.



PUBLIC AUDIT PAGE



LIVE REPORT



BRANDED PDF

SCALE

2 apps and 3 packages; 8 tables with 20 row-level security policies; 48 tracked bugs, all closed.

STACK Next.js 15 · React 19 · Tailwind 4 · Node worker · pg-boss · Playwright · Lighthouse 12 · Supabase (Postgres 17) · Claude API · Turborepo

Accountaire

64 static pages migrated to a 74-page, five-market platform

INDUSTRIES accounting and bookkeeping · professional services · multi-market SEO

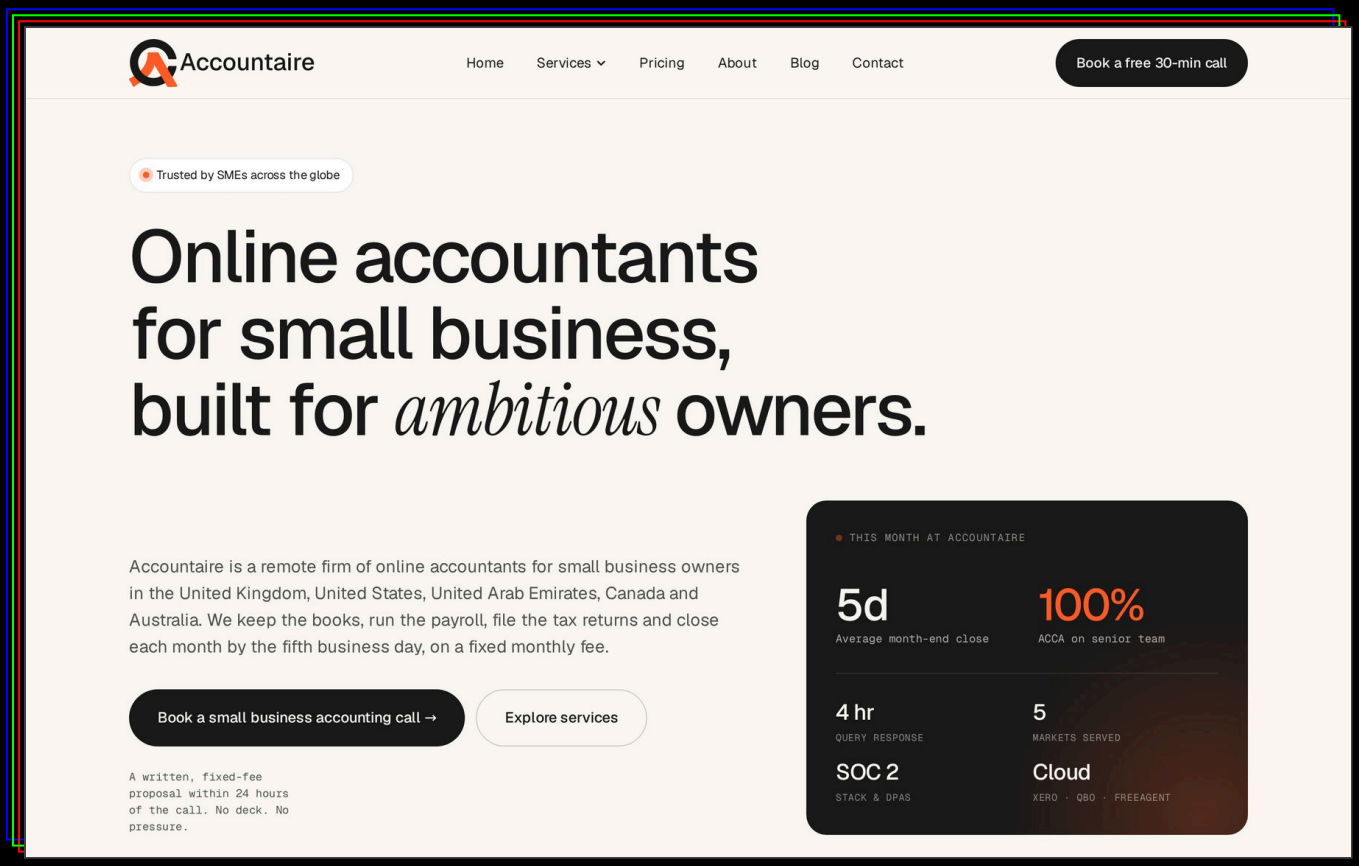
https://accountaire.com ↗

WHAT IT IS

The website of an online accounting and bookkeeping firm serving the UK, US, Canada, Australia and UAE, migrated from 64 static HTML pages to a CMS-driven platform with 74 pages.

MY ROLE

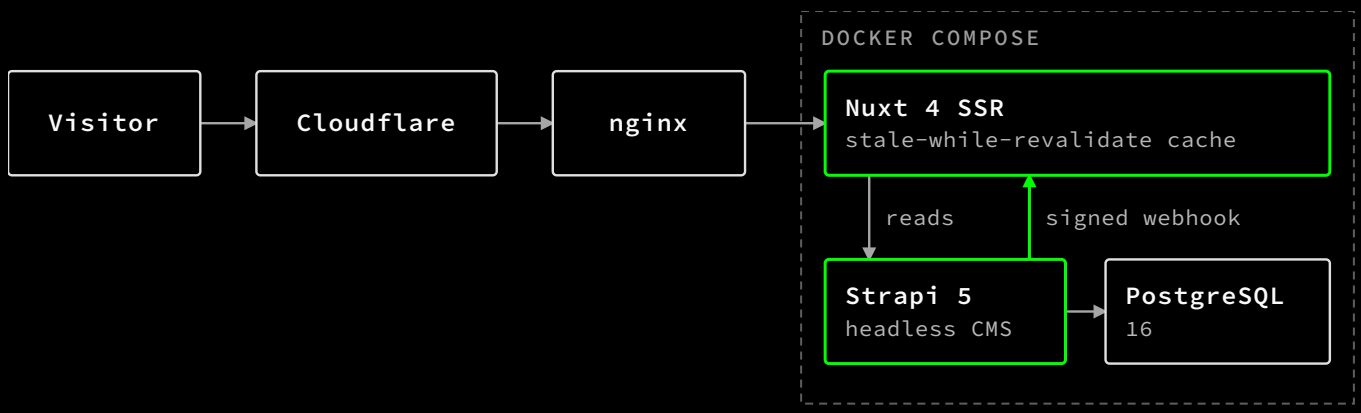
Migration, architecture, build, SEO and GEO programme, and production infrastructure.



HIGHLIGHTS

- Every page editable in the CMS through one block renderer with 22 block types.
- Currency switcher (USD, GBP, AED, CAD, AUD) applied on the client so cached pages stay correct.
- Markdown version of every page plus llms.txt for AI answer engines; per-page generated preview images; 64 legacy URLs redirected.
- SEO and GEO scorecard across 20 signals: composite 49.6 to 88.1 out of 100, with 17 of 20 at 90 or above.

ARCHITECTURE



DEPLOY ON PUSH

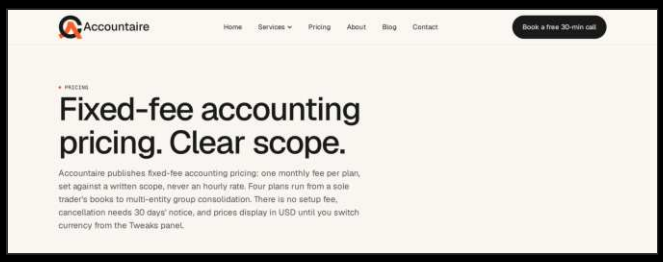


GUARDED CONTENT MIGRATION

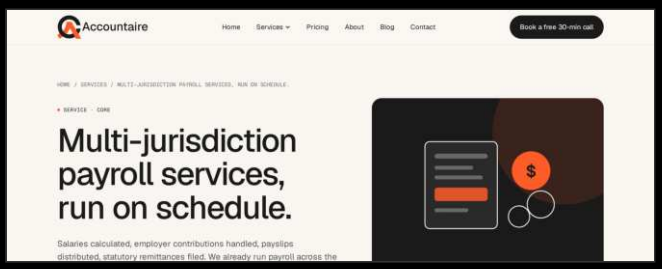


PUBLISH PATH: STRAPI WEBHOOK CLEARS THE PAGE CACHE

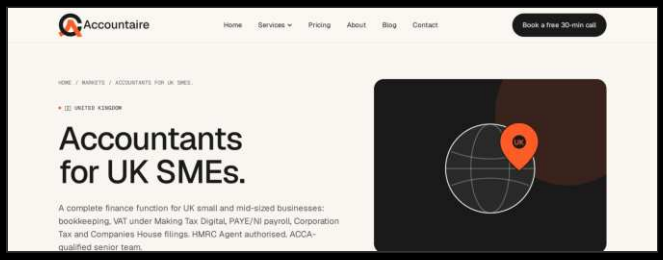
- Nuxt 4 server rendering with stale-while-revalidate caching, Strapi 5 headless CMS, PostgreSQL 16; Docker Compose behind nginx and Cloudflare.
- A Strapi publish fires a signed webhook that clears the page cache, so edits are live on the next request.
- GitHub Actions deploys on push with health gates; a separate guarded content-migration workflow backs up the database first and purges Cloudflare after.



PRICING PAGE



A SERVICE PAGE



A MARKET PAGE

STACK Nuxt 4 · Vue 3 · Strapi 5 · PostgreSQL 16 · Tailwind · Docker · nginx · Cloudflare · GitHub Actions · Vitest · Playwright

DrieVerse

A content platform with publish-time rule enforcement

INDUSTRIES technology services · custom software · AI automation · growth marketing

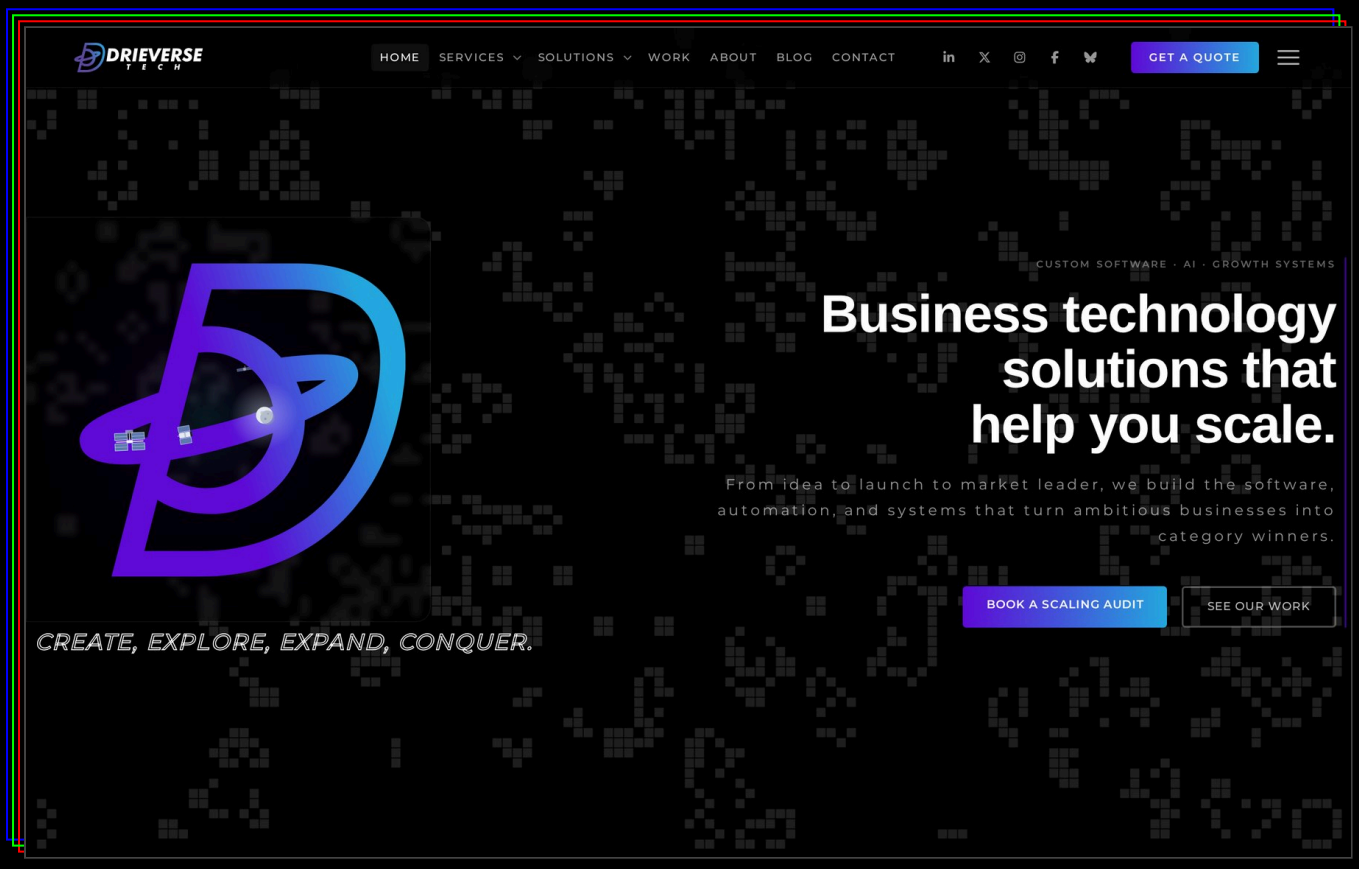
https://drieverse.com ↗

WHAT IT IS

The website and content platform of a business technology company selling custom software, AI automation, infrastructure, branding and growth marketing, rebuilt from a Vue 3 single-page site into a Nuxt 4 SSR platform with 82 pages.

MY ROLE

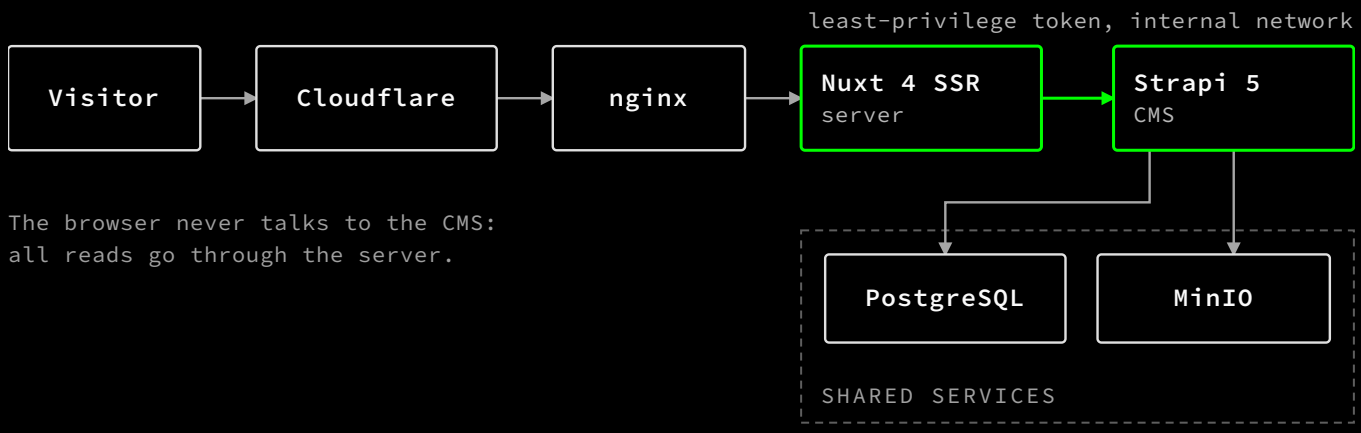
Rebuild, architecture, CMS, lead pipeline and production infrastructure.



HIGHLIGHTS

- The CMS blocks publishing of content that breaks the site rules: no em dashes, no pricing, required relations, sourced claims, fixed slugs.
- Lead pipeline for contact and quote forms: rate limiting, Turnstile, salted IP hashing, and a fallback that emails the lead if the CMS is down.
- Scheduled publishing, draft previews, IndexNow on publish, RSS and llms.txt, strict security headers.
- Motion-heavy front end with GSAP ScrollTrigger, smooth scrolling, an SVG preloader and installable PWA.

ARCHITECTURE

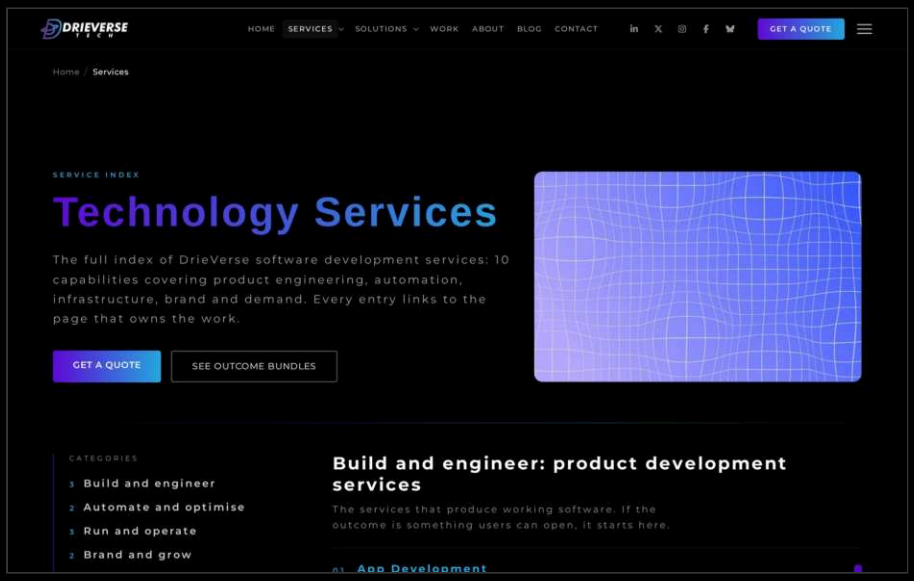


DEPLOY ON PUSH

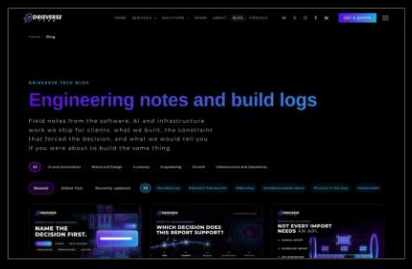


CONTENT PATH: SERVER-SIDE ONLY, BROWSER NEVER REACHES THE CMS

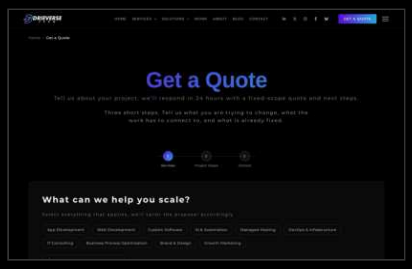
- Nuxt 4 SSR with a Strapi 5 CMS, shared PostgreSQL and MinIO, Cloudflare and nginx in front.
- The browser never talks to the CMS: all reads go through the server with a least-privilege token over the internal network.
- Push-to-deploy on GitHub Actions that builds and swaps two containers and smoke-tests 11 public paths.



SERVICES



BLOG



QUOTE FORM

STACK Nuxt 4 · Vue 3 · Strapi 5 · PostgreSQL · MinIO · Bootstrap 5 · GSAP · Docker · nginx · Cloudflare · GitHub Actions

LeadSyft

A Next.js and Payload platform with publish-to-live in seconds

INDUSTRIES B2B lead generation · brand strategy · marketing services

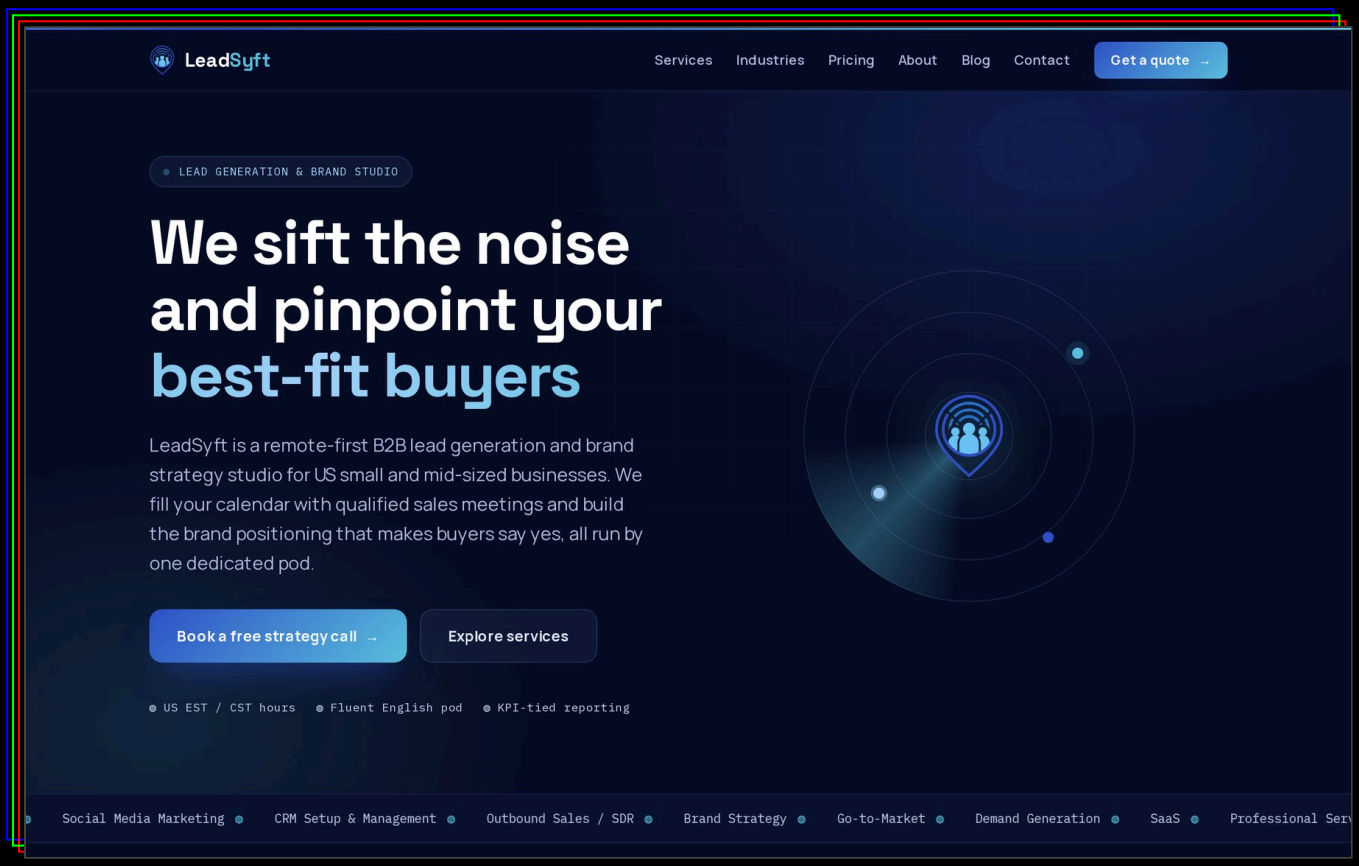
https://leadsyft.com ↗

WHAT IT IS

A rebuild of a B2B lead-generation and brand-strategy studio's website for US small and mid-size businesses, from 28 static pages to a search-optimised Next.js and Payload platform.

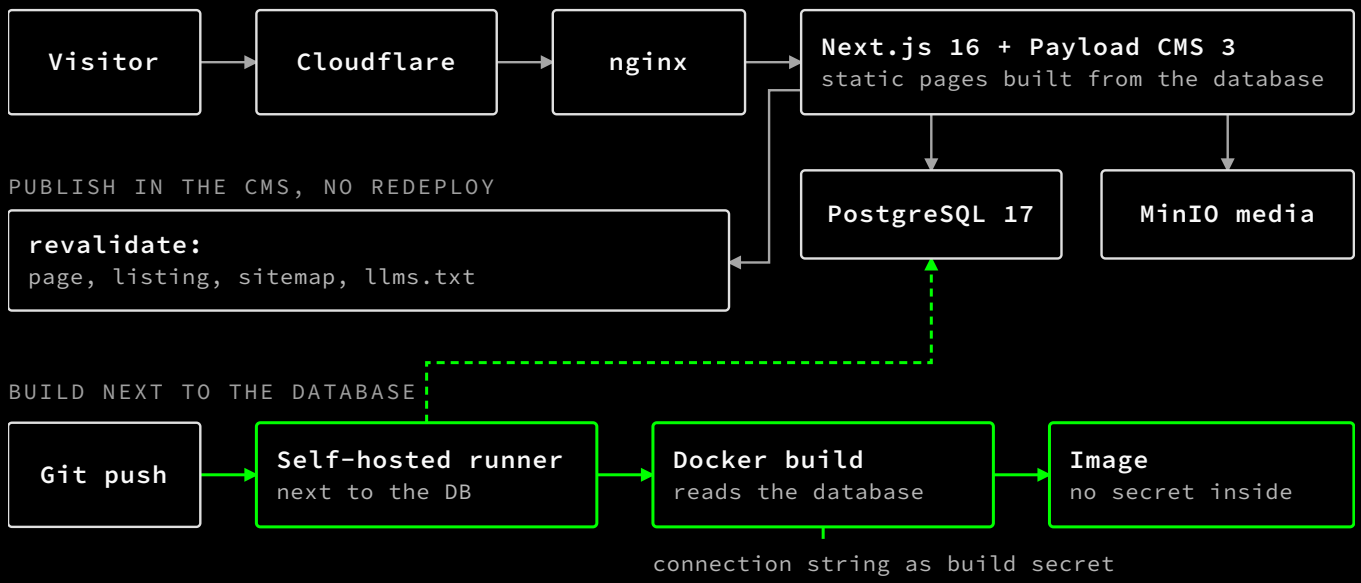
MY ROLE

Architecture, content migration, build, SEO and production deployment.



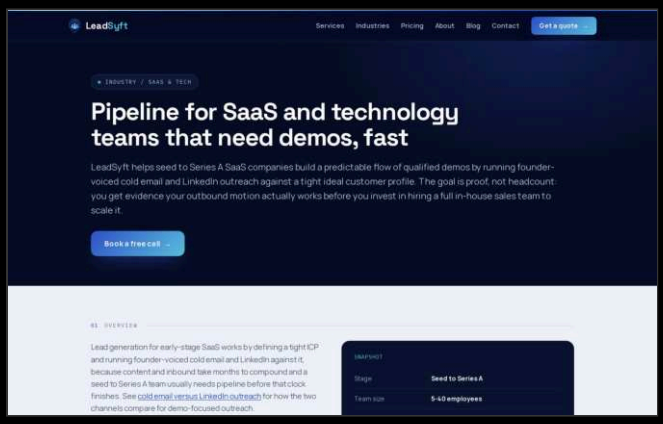
HIGHLIGHTS

- Verbatim import of the legacy site into a normalised content model (158 sections, 1,994 strings verified), with internal links turned into real CMS relationships.
- Renaming a page creates its redirect automatically.
- Full structured-data layer and AI answer-engine surface: comparison pages, glossary, FAQ markup, llms.txt.
- SEO audit composite raised from 47 to 80 over five rounds; production rebuilt from nothing to prove the pipeline.

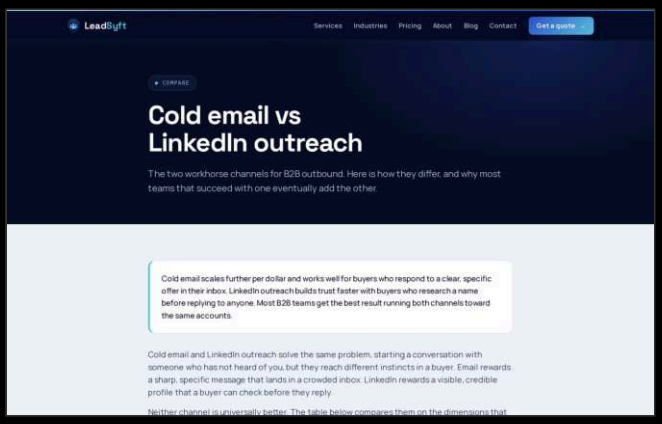


BUILD PATH: THE RUNNER SITS NEXT TO THE DATABASE

- Next.js 16 with Payload CMS 3 embedded, PostgreSQL 17, MinIO media, nginx and Cloudflare.
- Static pages built from the database; publishing in the CMS revalidates the page, its listing, the sitemap and llms.txt within seconds, with no redeploy.
- The build reads the database, so the pipeline runs on a self-hosted runner next to it, with the connection string passed as a build secret that never lands in the image.



AN INDUSTRY PAGE



A COMPARISON PAGE

STACK Next.js 16 · Payload CMS 3 · React 19 · PostgreSQL 17 · MinIO · Tailwind 4 · Docker · nginx · Cloudflare · GitHub Actions

DevOdin

LCP 2.19 s to 1.03 s on an agency rebuild

INDUSTRIES web development · SEO and marketing · IT solutions

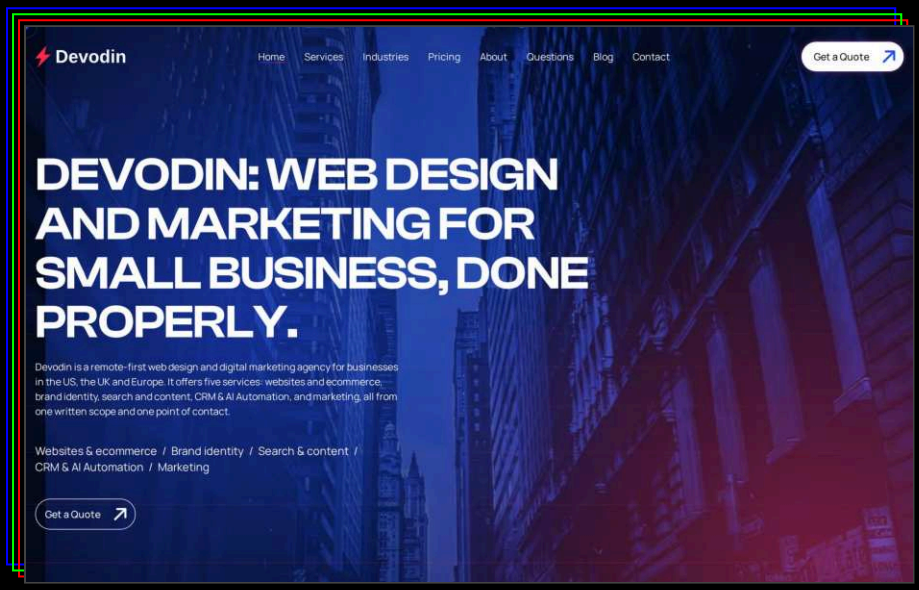
https://devodin.com ↗

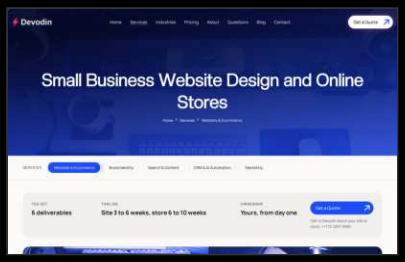
WHAT IT IS

A rebuild of a Dallas web, SEO, marketing and IT solutions agency's website on Next.js and Payload, with a CMS the client edits.

MY ROLE

Research-led service strategy, architecture, build, performance and SEO.





A SERVICE PAGE



QUOTE FORM

BEFORE AND AFTER

HOME-PAGE LCP	PAGE WEIGHT	REQUESTS	SEO AND GEO AUDIT
~~2.19 s~~ to **1.03 s**	~~2,087 KB~~ to **470 KB**	~~102~~ to **50**	~~61~~ to **95**

HIGHLIGHTS

- Three research streams (competitors, search demand, market) cut nine services to a focused set.
- Performance pass: home-page LCP 2.19 s to 1.03 s, page weight 2,087 KB to 470 KB, requests 102 to 50.
- SEO and GEO audit composite raised from 61 to 95 over seven rounds (project audits).
- Three-step quote brief, filterable blog that works without JavaScript, interactive canvas header, strict CSP.

STACK Next.js 16 · Payload CMS 3 · React 19 · PostgreSQL 17 · Tailwind 4 · GSAP · Lenis · Cloudflare Turnstile

Exact Hardware

A server-hardware store stocked by four distributor scrapers

INDUSTRIES e-commerce · enterprise IT hardware · distribution

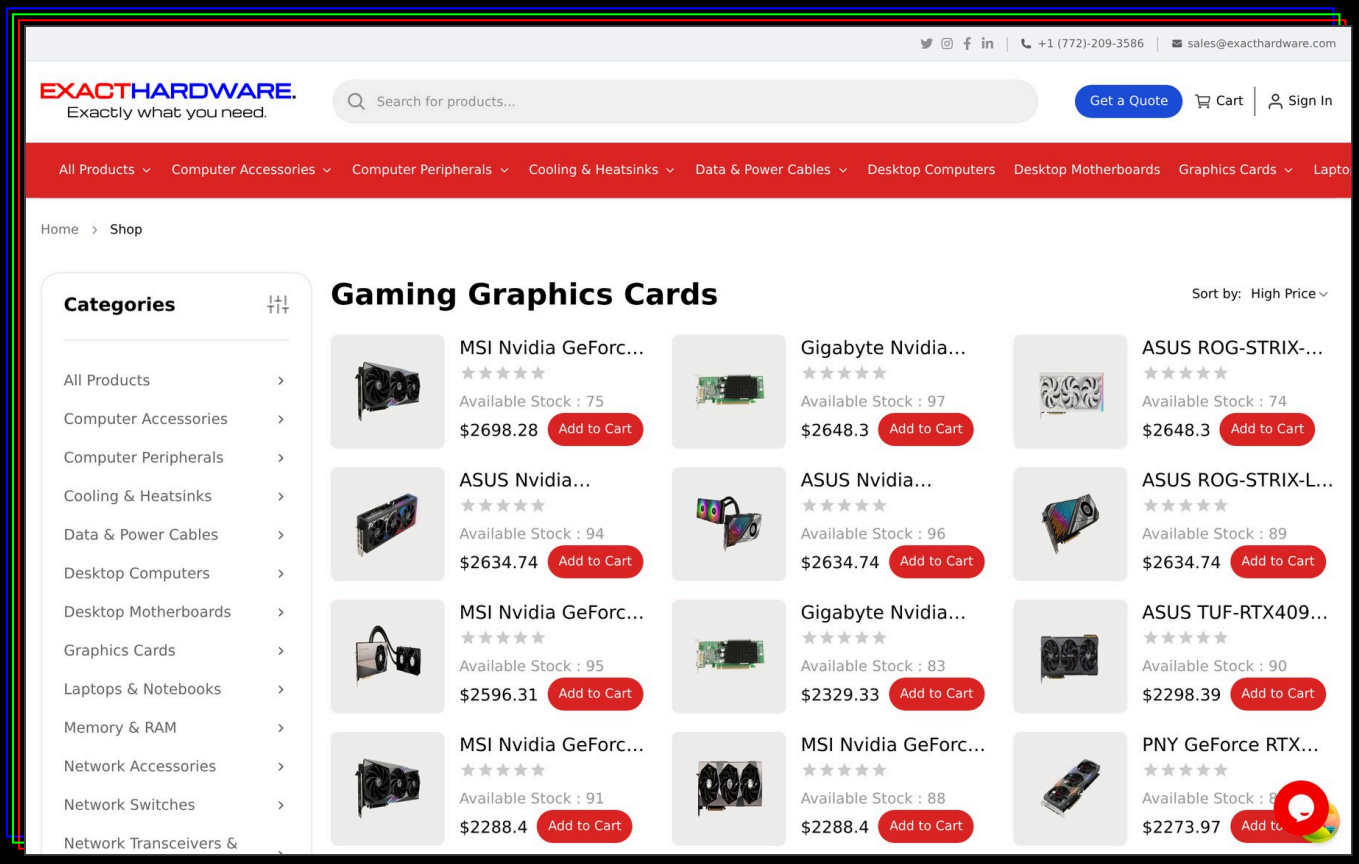
https://exacthardware.com ↗

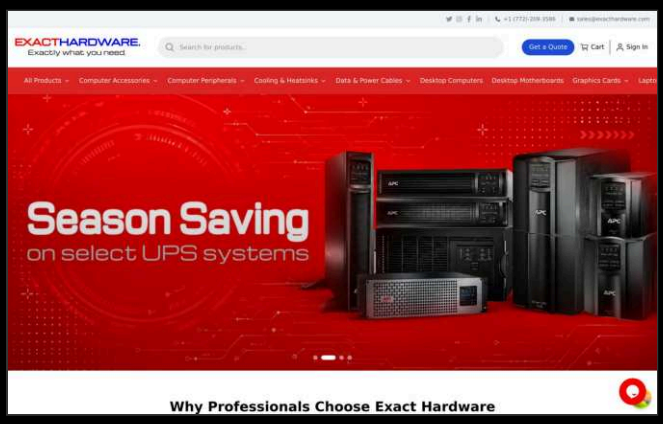
WHAT IT IS

An online store for enterprise and server hardware (GPUs, drives, RAID controllers, power supplies), with an admin panel and a scraping service that pulls catalogues from four distributors.

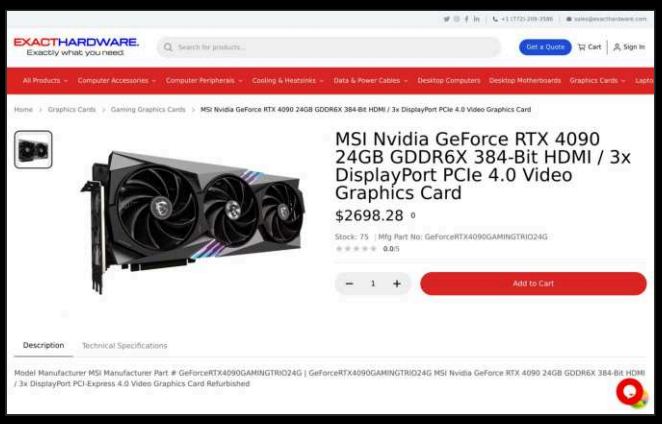
MY ROLE

Containerisation, both deploy pipelines, server provisioning with Ansible, the move from Supabase Cloud to self-hosted Supabase, and new features, working with the team.



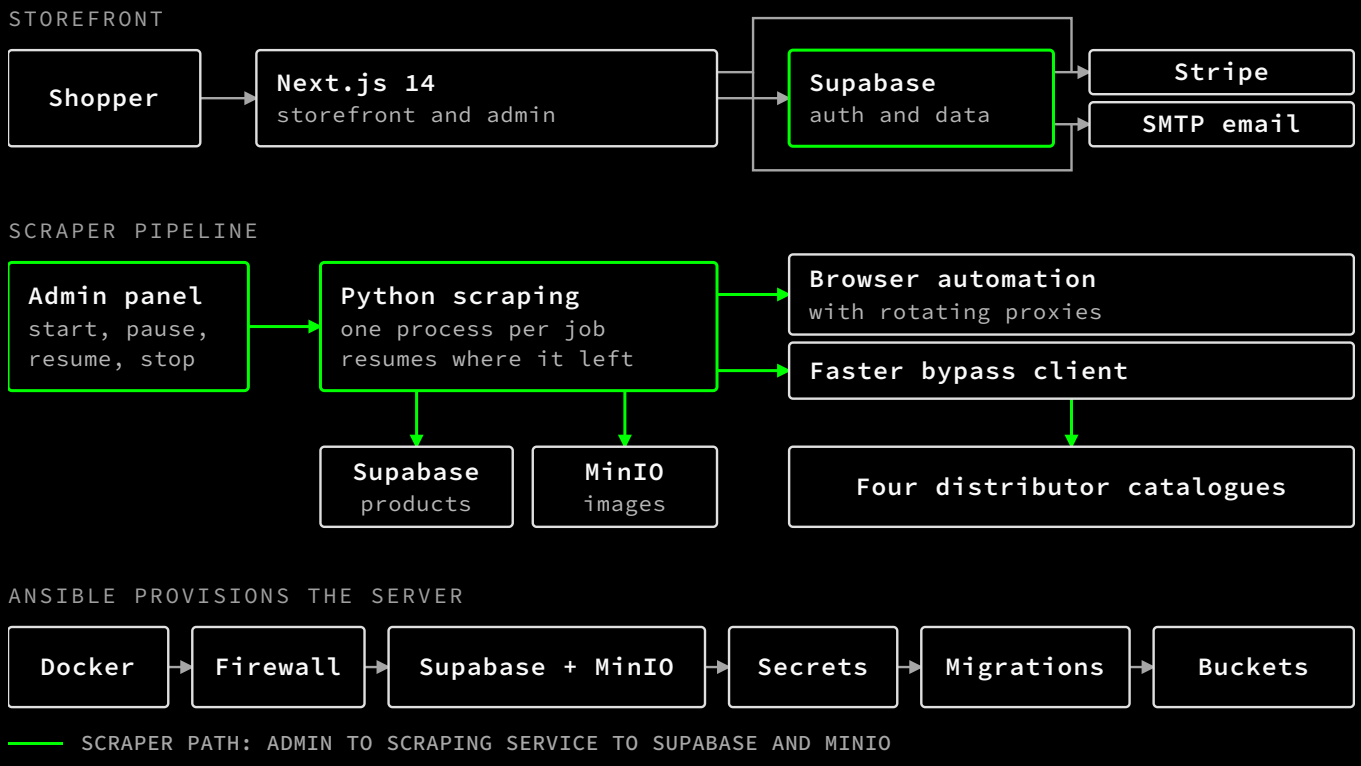


STOREFRONT HOME



A PRODUCT PAGE

ARCHITECTURE AND SCRAPER PIPELINE



- Next.js 14 storefront and admin on Supabase (auth and data), Stripe payments, SMTP email.
- Python scraping service: browser automation with rotating proxies for some distributors and a faster bypass client for others; every job runs as its own process that the admin can pause, resume or stop, and interrupted jobs resume where they left off. Products go to Supabase, images to MinIO.
- Ansible provisions the server: Docker, firewall, self-hosted Supabase with MinIO, generated secrets, database migrations and storage buckets.
- Production and staging environments for the store and the scraper.

STACK Next.js 14 · React 18 · TypeScript · Tailwind · shadcn/ui · Supabase (self-hosted) · Stripe · Python · Playwright · MinIO · Ansible · Docker

The self-hosted platform

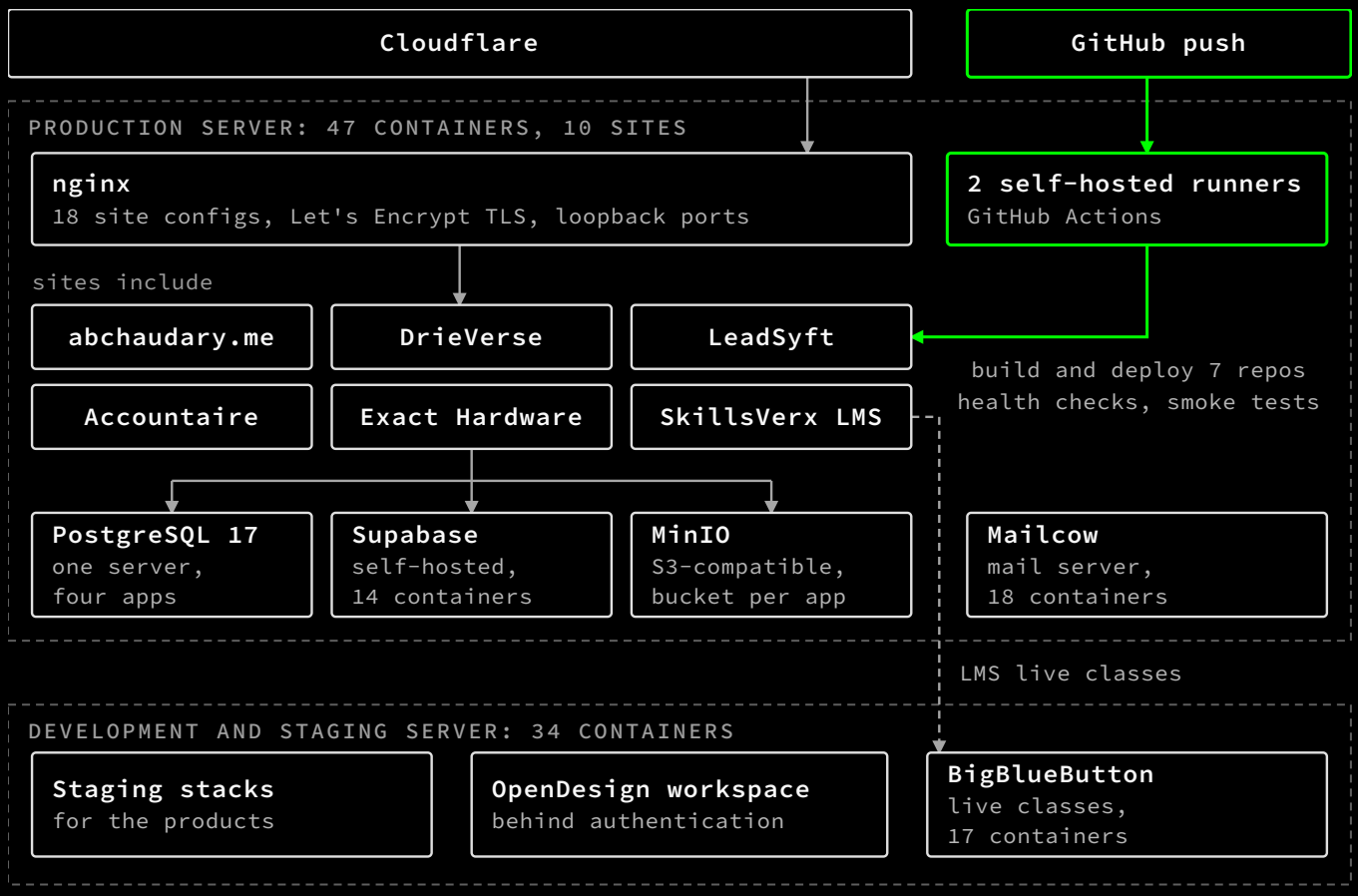
47 containers hosting 10 sites, run by one engineer

WHAT IT IS

The infrastructure I run myself, which hosts every product in this portfolio. Two servers: production and a development and staging server.

MY ROLE

Design, provisioning, security, deployments and operations, single-handed.



- **Production:** 47 containers hosting 10 sites. Cloudflare in front, nginx with 18 site configurations and Let's Encrypt TLS, every app port bound to loopback.
- **Shared data layer:** one PostgreSQL 17 server for four apps, self-hosted Supabase (14 containers) and S3-compatible MinIO storage with one bucket per app.
- **Mail:** a self-hosted Mailcow mail server (18 containers) for the domains.
- **Deploys:** two self-hosted GitHub Actions runners that build and deploy 7 repositories on push, with health checks and route smoke tests.
- **Development and staging server:** 34 containers, including the 17-container BigBlueButton stack for live classes, staging stacks for the products, and the OpenDesign workspace behind authentication.

STACK Linux · Docker · nginx · Cloudflare · Let's Encrypt · PostgreSQL 17 · Supabase · MinIO · Mailcow · BigBlueButton · GitHub Actions self-hosted runners · Ansible

abchaudary.me

A portfolio, blog and content platform built as a production system

INDUSTRIES developer portfolio · technical publishing · SEO and AI-search engineering

https://abchaudary.me ↗

WHAT IT IS

My own site, run as a production system. An animated portfolio with a Three.js scene and a soundtrack, a technical blog, a photography section, and a content engine that researches, gates, illustrates, schedules and distributes an 18-article campaign. Two typefaces (Press Start 2P and Source Code Pro), a red, green and blue colour system and motion timed to one soundtrack.

MY ROLE

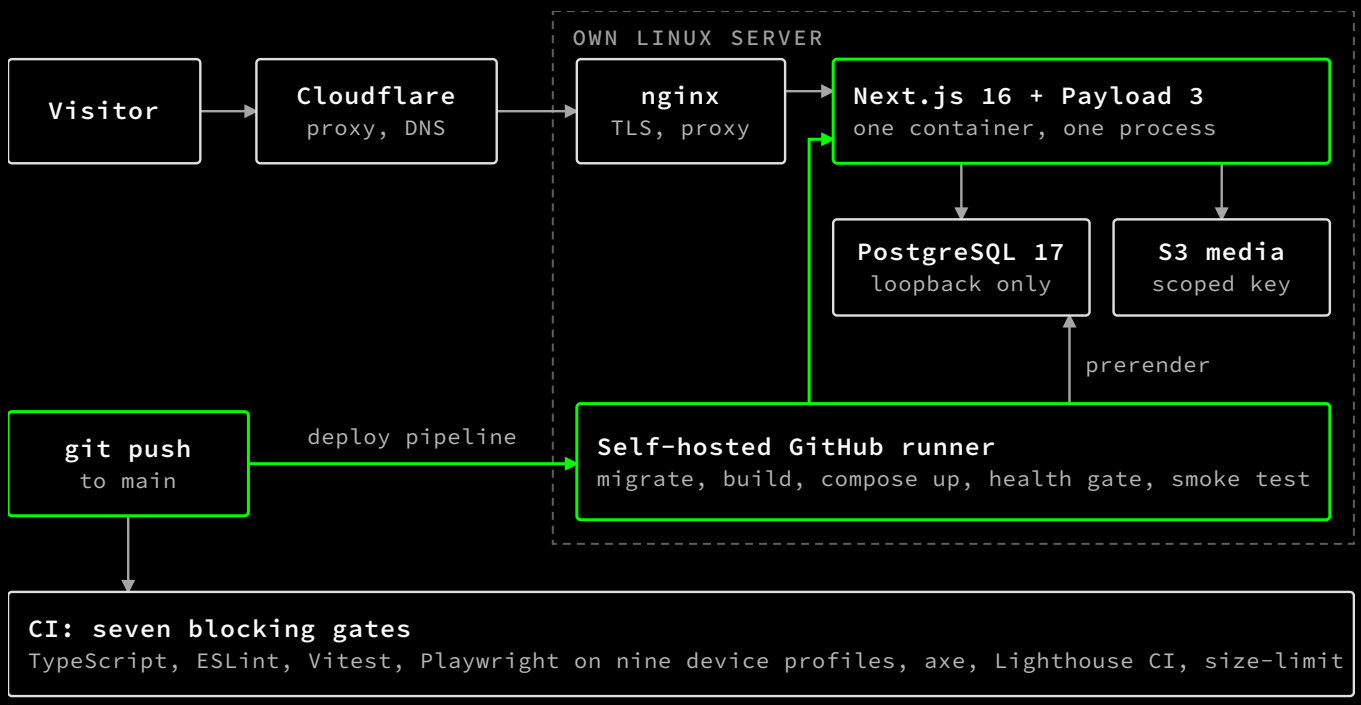
Architecture, full stack, DevSecOps, SEO and GEO engineering, the content pipeline and its automations.



ARCHITECTURE

- Next.js 16 (App Router, React 19, TypeScript) with Payload CMS 3 embedded in the same app and the same process; PostgreSQL 17 in its own loopback-bound container and Docker network; S3-compatible media storage reached with a scoped key; the GraphQL surface switched off; `/admin` is noindex.
- Nine collections (users that double as author profiles, posts, projects, categories, tags, technologies, media, photos, photo categories) and three globals (site settings, navigation, email settings). SMTP settings are editable in the admin and resolved on every send, so mail changes need no redeploy.
- Self-hosted on a Linux server behind Cloudflare and nginx. Heavy libraries (Three.js, GSAP) are route-scoped and dynamically imported, with performance budgets per route.
- Eighteen written architecture decision records.

ARCHITECTURE AND DEPLOY PATH



DEPLOY PATH: PUSH TO THE SELF-HOSTED RUNNER, NEXT TO THE DATABASE

DEPLOYS: A SELF-HOSTED GITHUB RUNNER ON THE PRODUCTION HOST

- Every push to `main` triggers one pipeline: install, database migrations, sync, image build, `docker compose up`, a health gate that polls the container for up to five minutes, an HTTP smoke test, and image pruning.
- The runner lives next to the database because the build prerenders routes from it. The connection string reaches the build as a BuildKit secret that never lands in an image layer; runtime secrets are regenerated from a GitHub Environment on every run and written with owner-only permissions.
- A cache-busting build argument forces a real rebuild on every run, and concurrent pushes queue instead of colliding.

CI: SEVEN BLOCKING GATES, NO `CONTINUE-ON-ERROR` ANYWHERE

01 TypeScript strict

02 ESLint at zero warnings

03 Vitest

04 Playwright e2e, responsive, SEO and visual snapshots across nine device profiles (snapshots are compared, never auto-updated)

05 axe accessibility with zero serious or critical findings on every route and device

06 Lighthouse CI on a mobile preset with hard budgets (LCP, FCP, CLS, INP, server response, accessibility and SEO scores)

07 size-limit on the shared JavaScript chunk

SEO, AEO AND GEO ENGINEERING

- A JSON-LD graph generated from the CMS (Person, WebSite, WebPage, BlogPosting with `speakable`, BreadcrumbList, CollectionPage, ImageObject, Service), a generated sitemap, `llms.txt` and `llms-full.txt` rebuilt on every publish, a Markdown companion of every indexable page, and AI crawlers explicitly allowed in `robots.txt`.
- Publishing in the admin revalidates the page, its hubs, its Markdown companion, the sitemap and `llms.txt` immediately, with no redeploy.

THE CONTENT PIPELINE: ONE CAMPAIGN DAY, END TO END



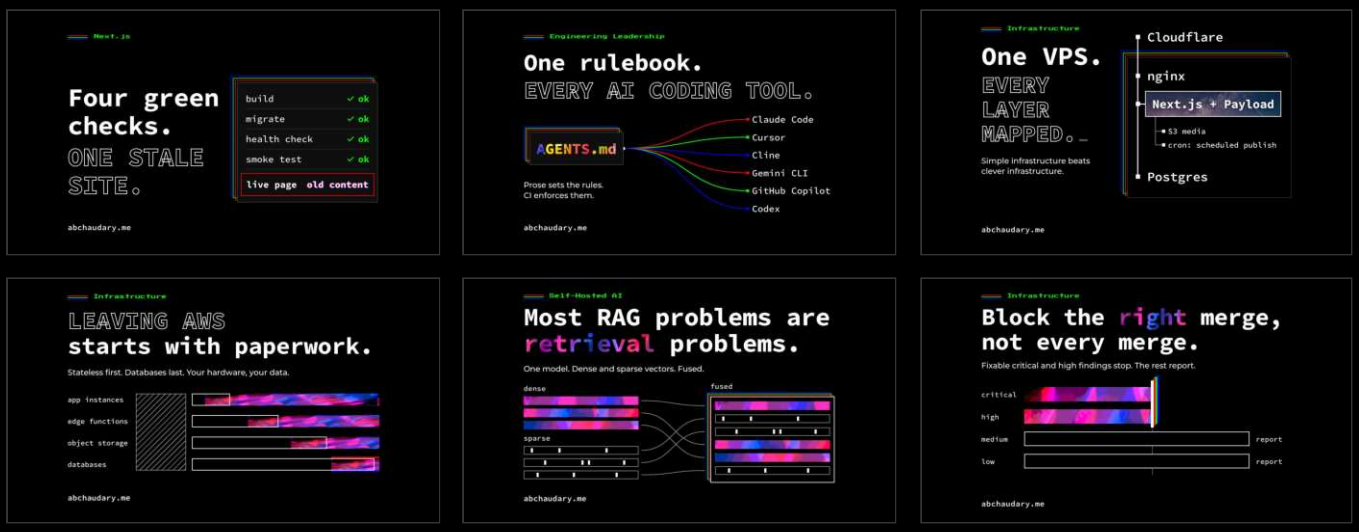
Each day ends with a pass and fail scorecard. 18 articles over four weeks.

THE DAY'S PATH: EVIDENCE TO DRAFT, THROUGH THE GATES, TO SCHEDULED PUBLISHING

- **Campaign plan:** four weeks, 18 articles, 20 native posts per platform on X and Bluesky, plus LinkedIn. Every day's article and native topic is assigned in advance with a brief.
- **"Do day X" workflow:** evidence file from commits, logs and code; keyword brief with a demand check and a cannibalisation check against the sitemap; a short interview with me on client-derived topics; a draft assembled only from evidence and my answers; the draft pushed to Payload as an unpublished post by script.
- **Gates that block the push:** optimisation (SEO, AEO, GEO, E-E-A-T, schema, a rendered-HTML inspection), craft and hook, and image. Each day ends with a scorecard of pass and fail rows with the evidence behind each, plus script checks for dashes, banned phrases and character limits.

- **Imagery with OpenDesign:** a self-hosted design agent produces one image per article, in 16:9 and 1:1, under a fixed rules file and a site-matched design system. Frames are re-rendered at 2x outside the agent, measured for safe-area fit and checked against the copy word for word. Animated GIF and MP4 exports are made for social uploads while the featured image stays static for page speed.
- **Scheduled publishing:** the approved image is attached and the post scheduled with Payload's native scheduled publishing; a host cron calls the jobs endpoint every minute with a bearer secret compared in constant time.
- **Social distribution:** a paste-ready set for LinkedIn, X and Bluesky per article, plus native posts, each with hashtags, verified mentions, UTM tags, alt text and per-platform length counts, written one row per post into a campaign sheet.

CAMPAIGN IMAGERY, MADE WITH OPENDESIGN



FEATURED IMAGES, 16:9

AUTOMATIONS

The push-to-draft, attach-and-schedule, content-seeding and settings scripts; the em-dash, banned-phrase and length checks; the daily trend check that can swap a native topic; the weekly internal-link, `llms.txt` and sitemap audits.

STACK Next.js 16 · React 19 · TypeScript · Payload CMS 3 · PostgreSQL 17 · MinIO/S3 · GSAP · Three.js · Docker · nginx · Cloudflare · GitHub Actions (self-hosted runner) · Playwright · Vitest · axe · Lighthouse CI · OpenDesign · Claude

AUTOMATIONS

Agent and workflow systems that write, schedule and post content for client teams and for my own projects. All of it runs on infrastructure I host and configure myself.

2.1 AUTOMATIONS

Content creation and automated posting, self-hosted end to end

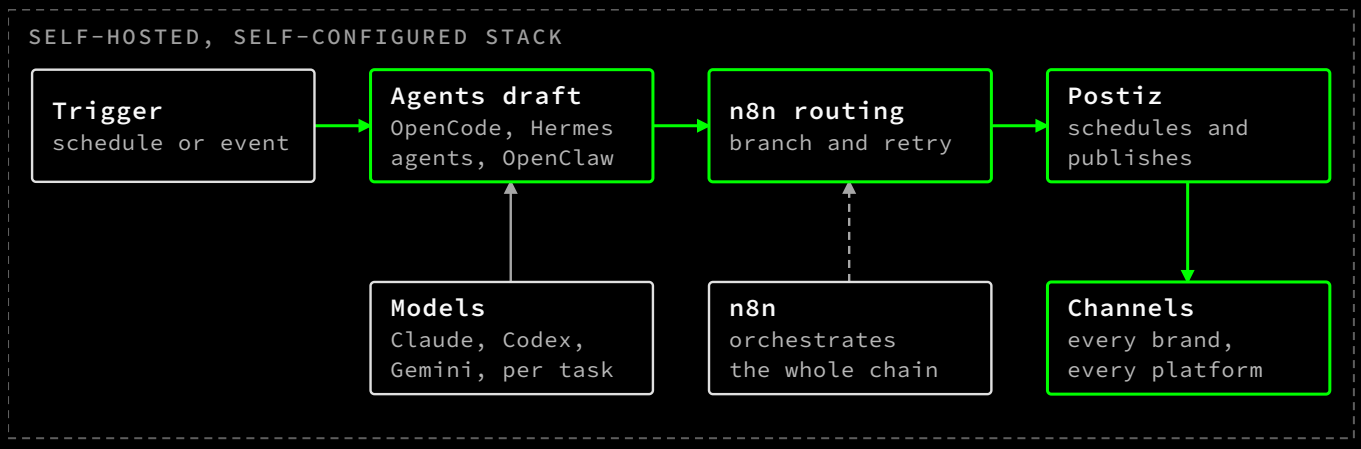
WHAT IT IS

Content-creation and automated-posting workflows I designed, deployed and configured for 10+ clients, and that runs 4+ of my own projects fully automated. The same stack drafts, routes, schedules and publishes across channels, without a vendor platform in the middle.

MY ROLE

Design, self-hosting, configuration, prompt and agent design, integrations and operations.

CONTENT AUTOMATION PIPELINE



10+ clients, 4+ of my own projects fully automated

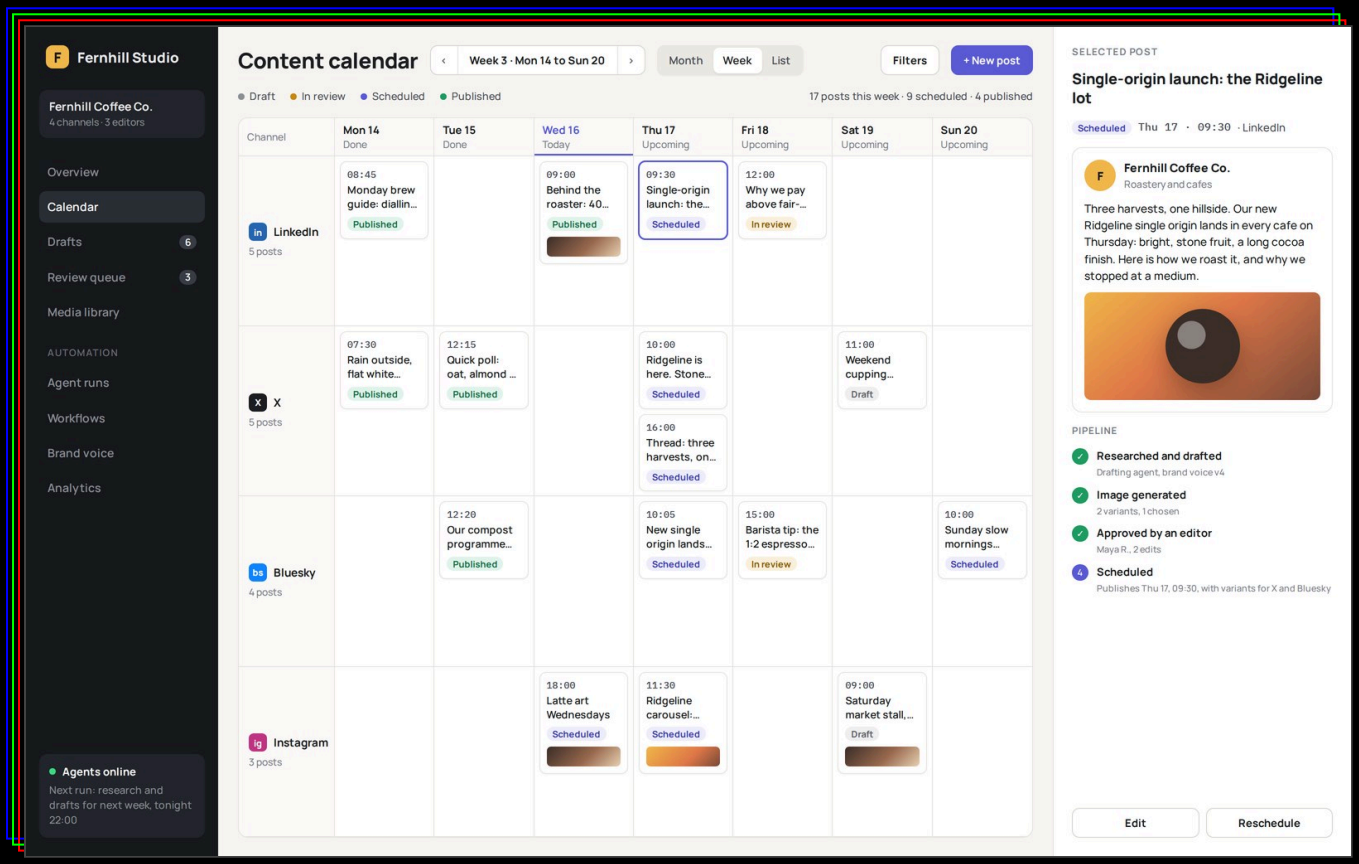
CONTENT PATH: TRIGGER TO AGENTS, N8N ROUTING, POSTIZ, EVERY CHANNEL

THE STACK, AND WHAT EACH PIECE DOES

- **n8n (self-hosted, self-configured):** the orchestrator. Schedules, triggers, API calls, branching, retries and delivery.
- **OpenCode, Hermes agents and OpenClaw:** the agent runtimes that do the multi-step work: research, drafting, tool use and handoffs.
- **Claude, Codex and Gemini:** the models behind the agents, chosen per task rather than per brand.
- **Postiz (self-hosted):** the scheduler and publisher for social channels.

WHY IT MATTERS

- One owner of the whole chain means no hand-offs between a content tool, a scheduling tool and an AI tool. I tune all three together.
- Built once, adapted per project and per client: the same pattern serves different brands, voices and channels.
- Self-hosting keeps credentials, drafts and schedules inside infrastructure I control.



CONTENT CALENDAR: POSTS SCHEDULED ACROSS CHANNELS

Representative interface · fictional brand and copy

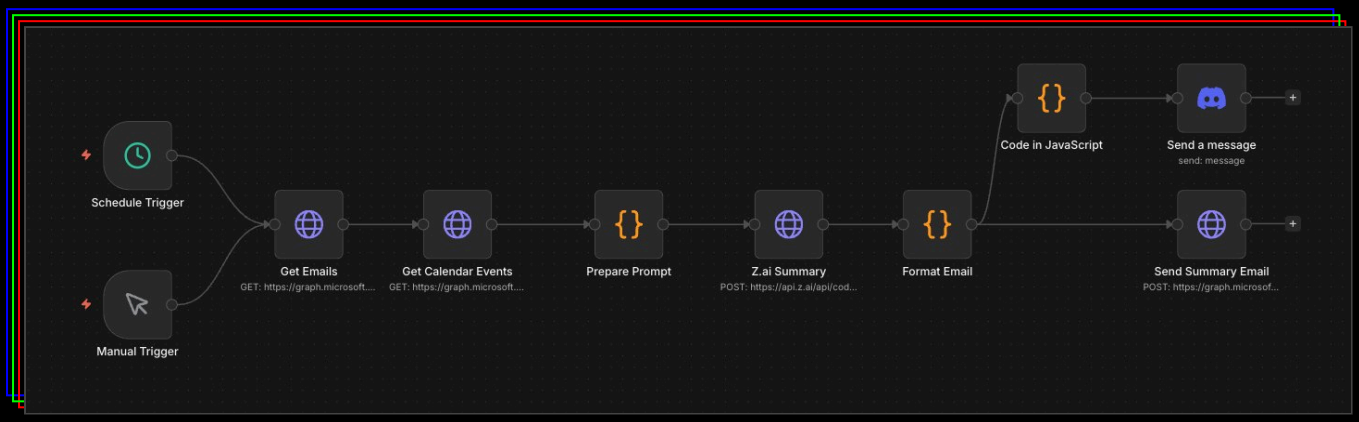
10+ clients served by the stack

4+ of my own projects fully automated

Personal n8n workflows: briefings and news digests

Self-hosted n8n workflows, all built around AI.

Daily work briefing: email and calendar, summarised by AI

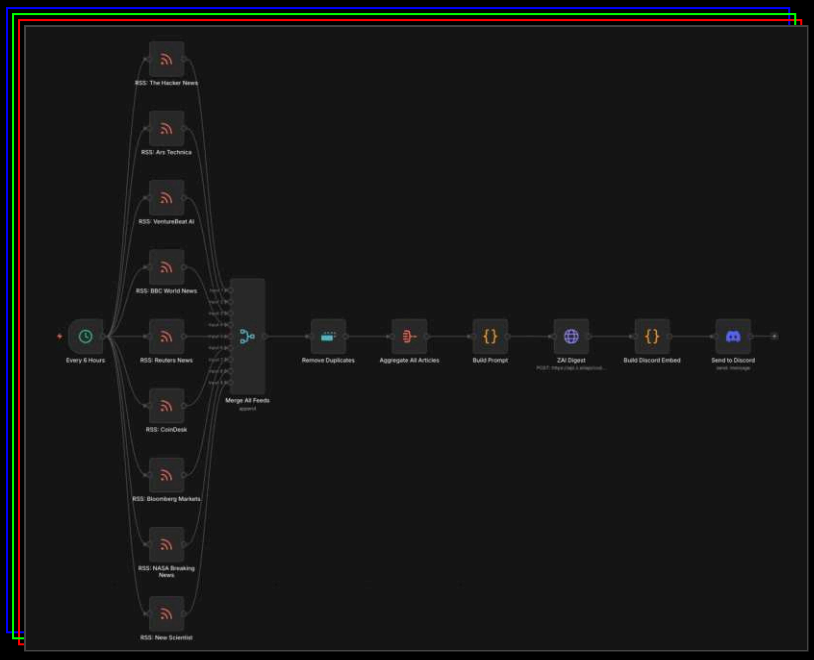


TRIGGER	On a schedule, or by hand.
FLOW	Reads my work inbox and today's calendar through the Microsoft Graph API, builds a prompt, has an LLM (Z.ai) write the briefing, formats it, then delivers it twice: as an email to a separate address through Microsoft Graph, and as a message in my Discord channel.
WHY	One read at the start of the day replaces scanning two inboxes and a calendar.

Personal email briefing

The same pattern for my personal inbox: fetch, summarise with AI, deliver to email and Discord.

News digest: nine RSS sources summarised by AI every six hours



TRIGGER

Every 6 hours.

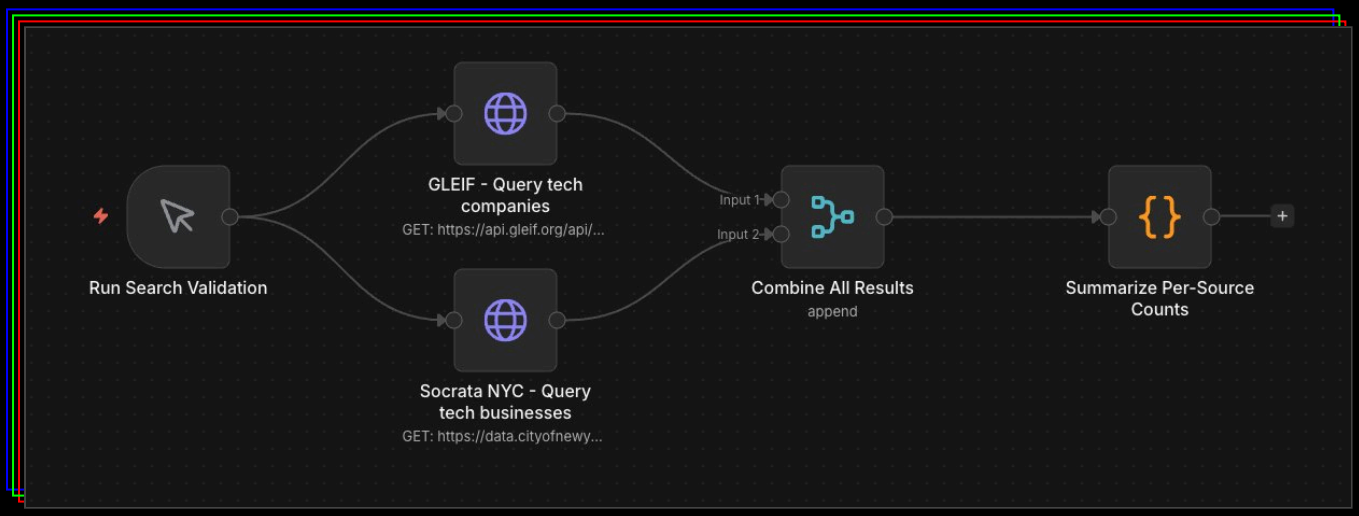
SOURCES

Nine RSS feeds in parallel: The Hacker News, Ars Technica, VentureBeat AI, BBC World News, Reuters, CoinDesk, Bloomberg Markets, NASA Breaking News and New Scientist.

FLOW

Merges all feeds, removes duplicates, aggregates the articles, builds a prompt, has an LLM (Z.ai) write the digest, formats it as a Discord embed and posts it to a dedicated news channel.

ACRM search source validation



Queries two open company-data sources in parallel, the GLEIF legal-entity API and New York City's Socrata open data, combines the results and summarises the counts per source, to validate data sources for the ACRM lead-intelligence platform (Part 4).

STACK n8n (self-hosted) · Microsoft Graph API · Z.ai LLM API · RSS · Discord · JavaScript code nodes

CLIENT ENGAGEMENTS

Delivered under NDA and presented by industry. Product screens are representative mock-ups with fictional data.

Scope across these engagements: GPU clusters, colocation across providers, a private 10 Gb link, in-app billing across stores, and the AI platform behind a $600M+ acquisition.

3.1	**AI platform**	pharmaceuticals · marketing research · healthcare analytics
3.2	**Math app for schools**	EdTech · education partnerships · subscriptions and billing
3.3	**Card automation**	healthcare · business tools · marketing and customer engagement
3.4	**Award-letter API platform**	EdTech · higher education · FinTech and financial aid · document AI
3.5	**AI coach**	coaching · personal training · consumer AI
3.6	**Platform migrations**	across client verticals
3.7	**Workforce tools**	business tools · workforce management · HR operations
3.8	**Fitness-market analytics**	fitness · market research · data analytics

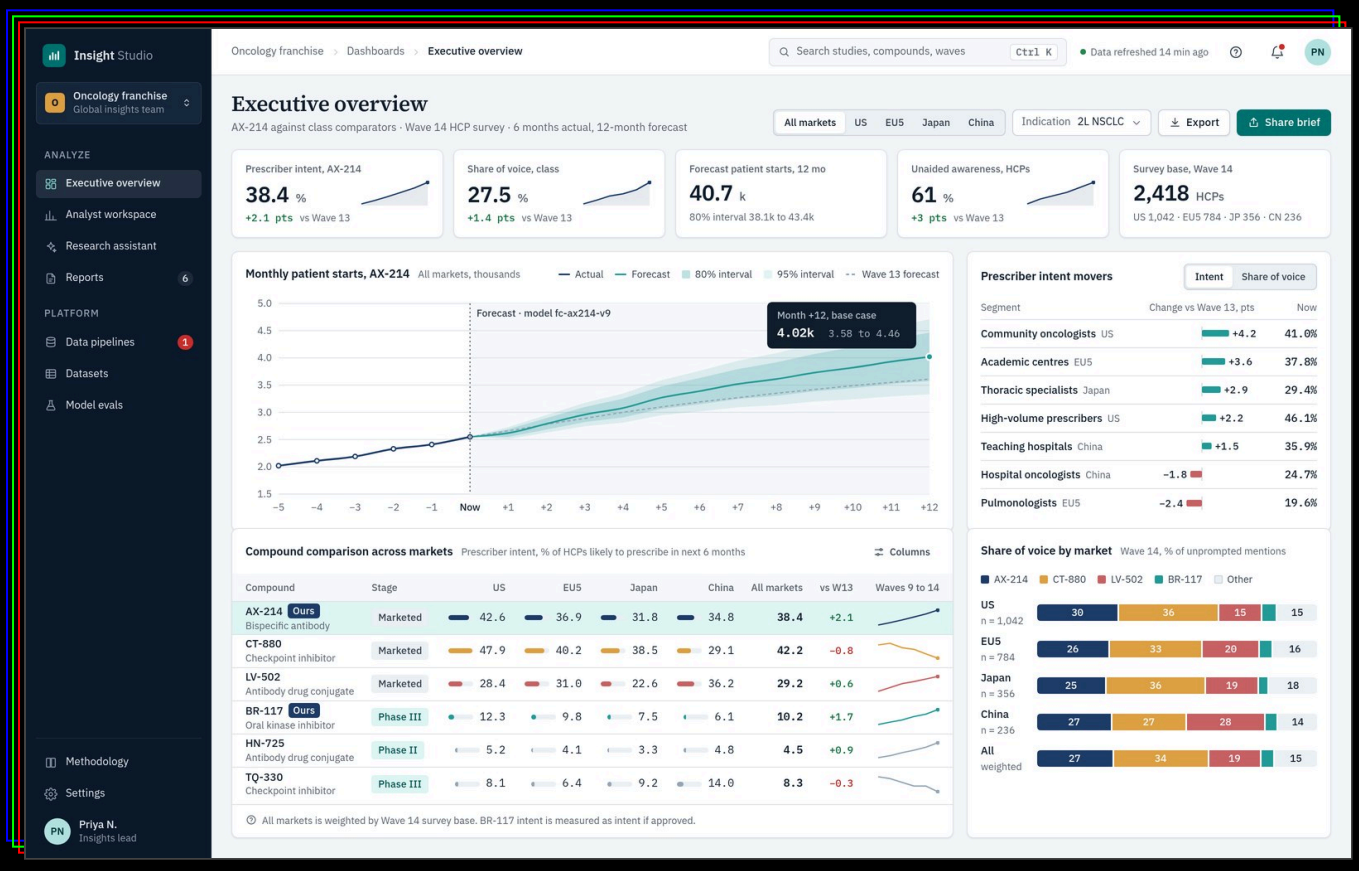
The AI platform that became the main selling point of a $600M+ acquisition

INDUSTRIES pharmaceuticals · marketing research · healthcare analytics

ROLE Lead Engineer, DevSecOps Architect and Team Lead. About nine months, proposal to production, leading a team of four.

$600M+	Nine months	4	H100 then H200
acquisition	proposal to production	a team of four, led	GPU tier

The problem. A pharmaceutical marketing-research company held survey and research data in huge volumes, refreshed frequently. Scientists, analysts, sales teams and executives each needed answers rather than exports: scores, forecasts and comparisons across trials, drugs and markets, at a scale no spreadsheet reaches.



EXECUTIVE OVERVIEW: FORECAST AND MARKET COMPARISON Representative interface · client work under NDA

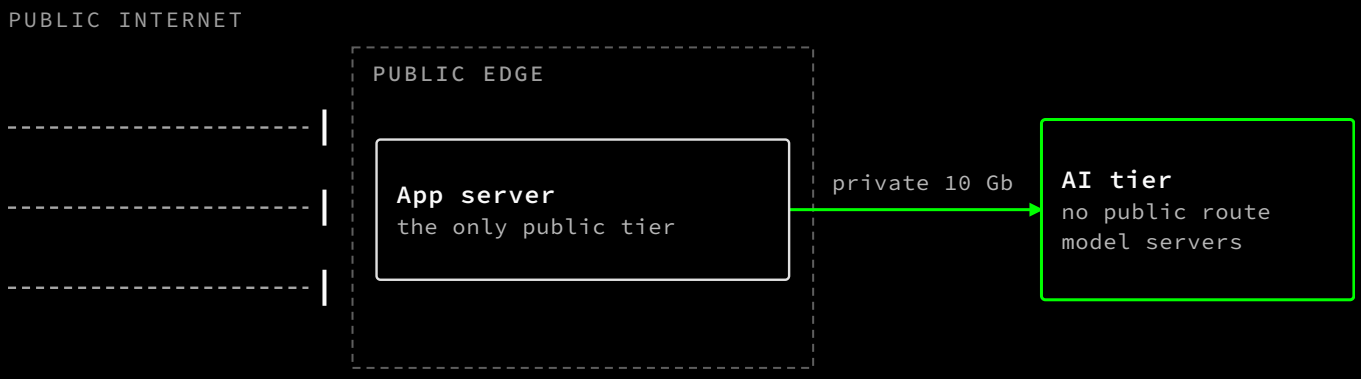
What I built. A self-hosted AI data-intelligence platform. It ingests research big data from surveys and other sources, then runs evaluations and complex math over it: scoring, forecasting, trial-to-trial, drug-to-drug and market-to-market comparison, and more. LLMs and RAG pipelines sit on top, behind a custom chat and evaluation engine with tool calling, so a scientist, an analyst, a salesperson and an executive can each ask in their own terms. The models are fine-tuned, self-hosted open-weight models (GPT-OSS, Llama, Qwen).

Security that the biggest names in tech did not have.

A WAVE OF AI SUPPLY-CHAIN ATTACKS REACHED

~~OpenAI~~ ~~Anthropic~~ ~~Meta~~ ~~Google~~ ~~AWS~~

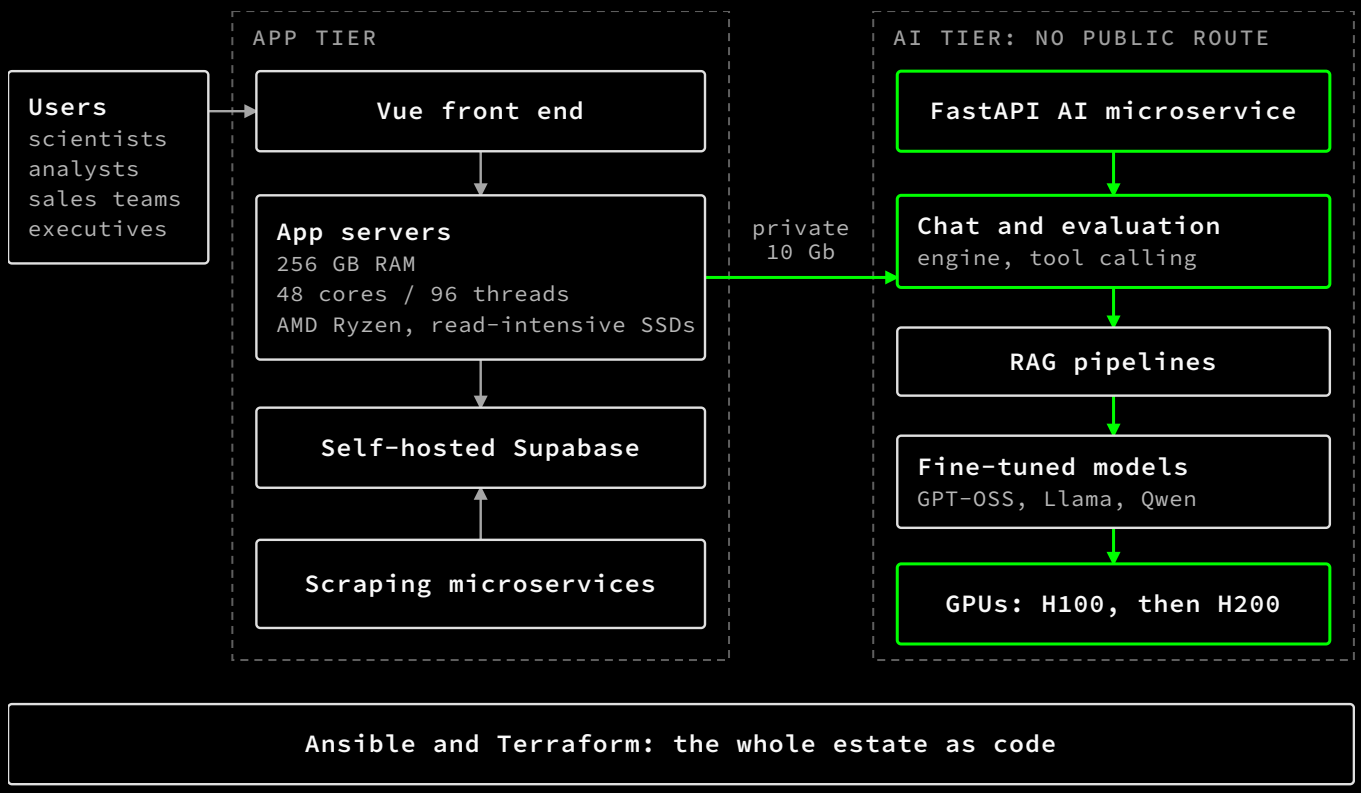
This platform's model servers stayed out of reach.



THE ONLY ROUTE TO THE AI TIER: A PRIVATE LOCAL 10 GB NETWORK

A wave of AI supply-chain attacks reached OpenAI, Anthropic, Meta, Google and AWS. This platform's model servers stayed out of reach, because I designed them that way from day one: the AI tier had no public route at all and talked to the application server only over a private local 10 Gb network that I architected and configured. Security designed in from the first diagram, not bolted on after the breach.

ARCHITECTURE: HYBRID CLOUD AND COLOCATION

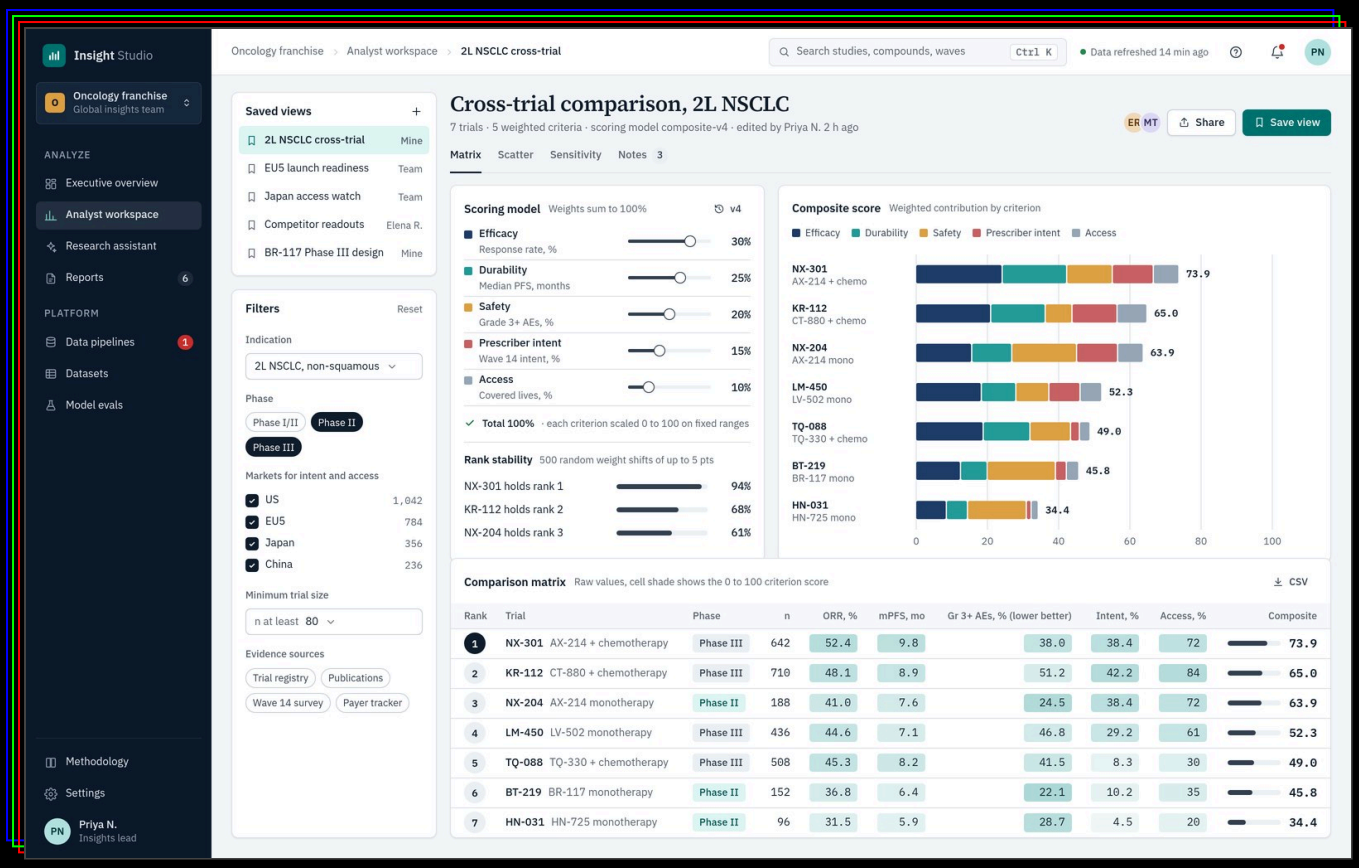


Managed cloud and colocation, across different providers

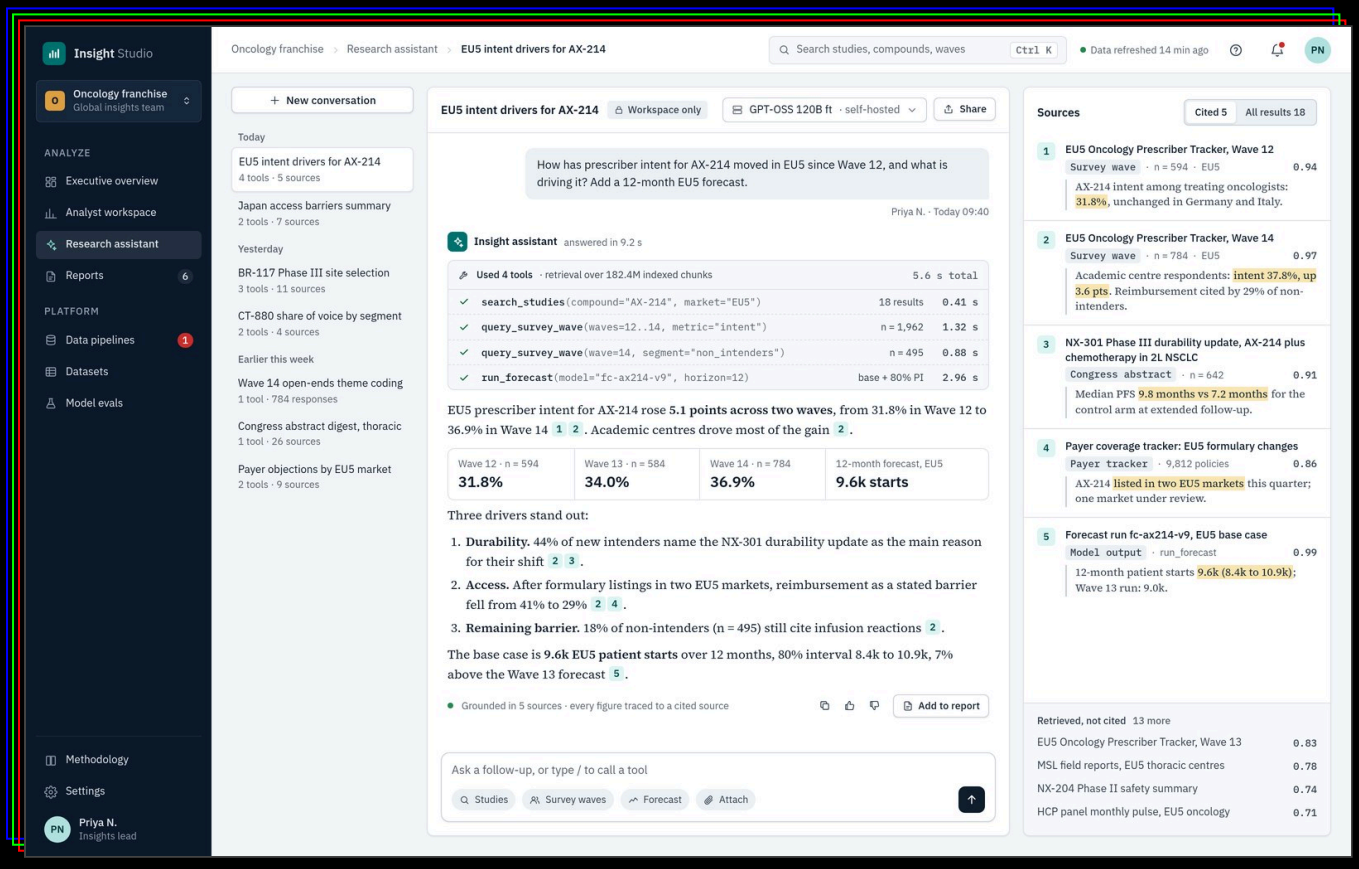
KEY PATH: APP SERVERS TO THE AI TIER OVER THE PRIVATE 10 GB LINK, DOWN TO THE GPUS

THE HARD PARTS

- **Infrastructure across providers.** Managed cloud and colocation, spread across different providers. GPU compute started on a cluster of NVIDIA H100s and moved to H200 servers. The application servers were 256 GB RAM, 48-core, 96-thread machines on the latest AMD Ryzen with read-intensive SSDs. I configured every one of them myself, and handled hardware procurement.
- **Continuously refreshed data.** Scraping microservices and research-source ingestion keep huge datasets fresh, and the evaluation math has to stay correct as they refresh.
- **Self-hosted models.** Fine-tuned, self-hosted open-weight models, a custom evaluation engine and tool calling, with no third-party model API in the path of the client's data.
- **Four audiences on one platform.** Scientists, analysts, sales teams and executives use the same platform for different questions.
- **DevOps as code.** Ansible and Terraform for the whole estate, so the colocation and cloud environments are reproducible.



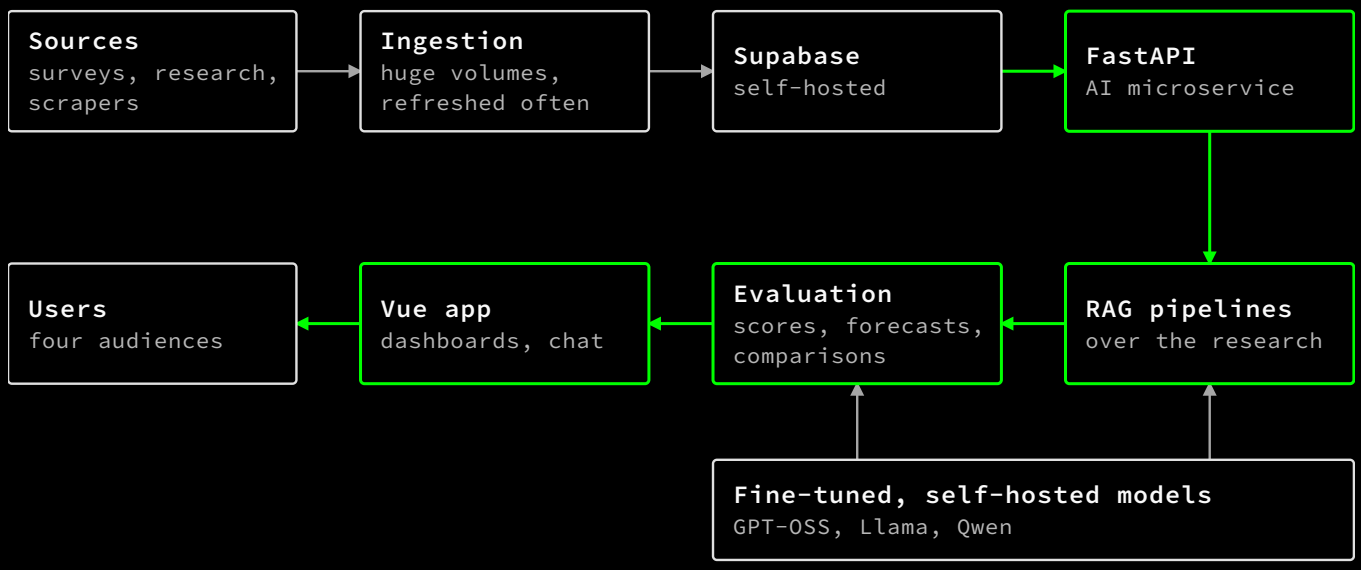
ANALYST WORKSPACE: CROSS-TRIAL SCORING AND COMPARISON Representative interface · client work under NDA



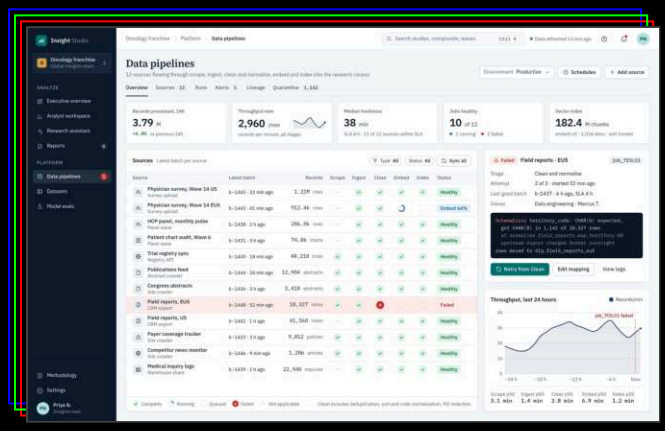
CHAT AND EVALUATION WORKSPACE: TOOL CALLS AND CITATIONS

Representative interface · client work under NDA

DATA FLOW: RESEARCH SOURCES TO ANSWERS

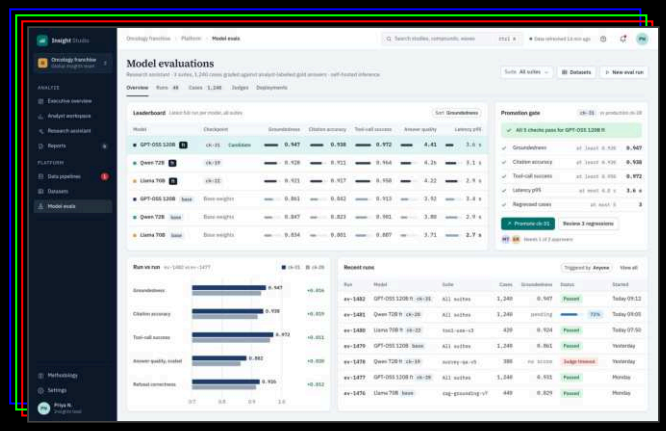


ANSWER PATH: FASTAPI TO RAG, EVALUATION AND THE VUE APP



INGESTION PIPELINES

Representative interface · client work under NDA



MODEL EVALUATION CONSOLE

Representative interface · client work under NDA

- **End-to-end ownership.** I wrote the winning technical proposal (still attached to my NDA as Exhibit A), led client meetings and onboarding end to end, architected the stack and system design alone, from infrastructure and security to hardware procurement, hired two developers, and wrote code alongside the team.

OUTCOME

The product became the main selling point of the client company's $600M+ acquisition.

STACK Vue.js · FastAPI AI microservice · self-hosted Supabase · scraping microservices · GPT-OSS · Llama · Qwen · RAG · Python · NVIDIA H100 and H200 · AMD Ryzen · Ansible · Terraform · GitLab · Docker · Linux

A math learning app for schools on several continents, and a cloud migration that cut infrastructure cost about 70%

INDUSTRIES EdTech · education partnerships · subscriptions and billing

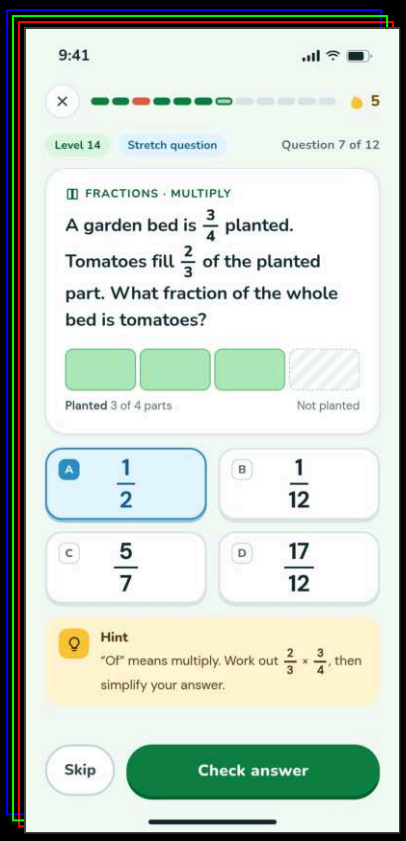
ROLE Principal Engineer, Infrastructure, Full-Stack, DevSecOps and Team Lead

The problem. A math-enrichment organisation partnered with schools in different countries and on different continents, serving tens of thousands of students. It needed one learning app on every device, a billing model that survives two app stores and the web, and infrastructure whose cost fit the product's revenue.

What I built. One React Native app that ships to web, iOS and Android, on a Flask backend. Adaptive levels, competitions, progress tracking, teacher views, white-labelling for some of the client's own customers, and per-seat selling. I led the team and ran the company's engineering operation: infrastructure, pipelines, GitLab, Jira, several WordPress sites, and at times IT operations including Microsoft 365.

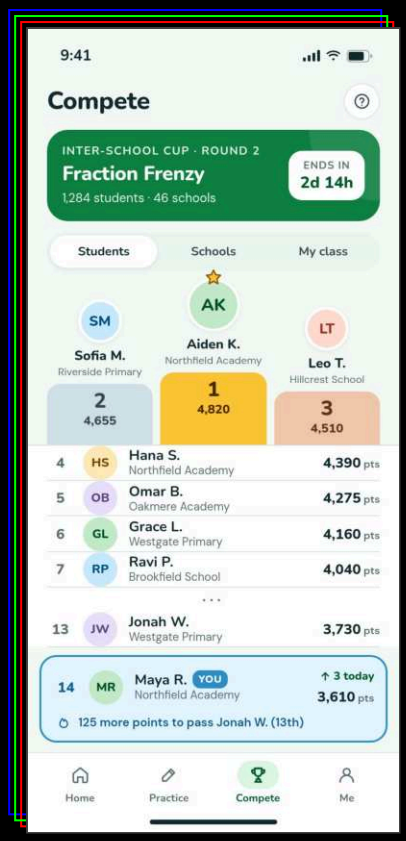
THE HARD PARTS

- **Billing across stores.** Apple in-app billing on iOS and Stripe on the web, with entitlement tracked across devices so a student who pays once is unlocked everywhere.
- **Selling to schools.** Per-seat licensing and white-labelling mean the same codebase presents as different brands with different seat counts.
- **Infrastructure sized to the business.** The company started on a basic AWS setup. I moved it to my self-hosted, secure private cloud architecture: server selection, networking, Linux hardening, and containerised deployments behind an nginx reverse proxy for the apps and the WordPress properties.
- **Staging and testing environments and CI/CD** for every product, with GitLab governance across the team.



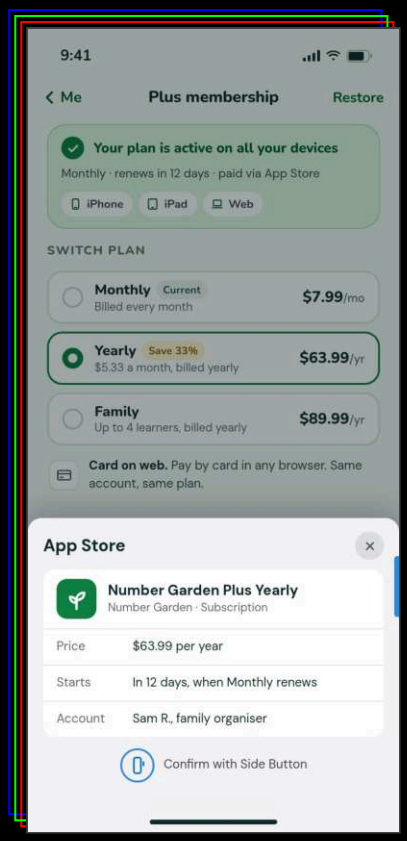
ADAPTIVE PRACTICE

Representative interface · client work under NDA



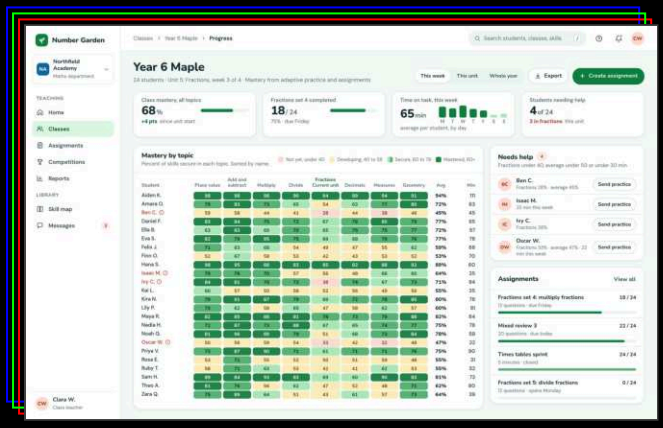
COMPETITION LEADERBOARD

Representative interface · client work under NDA



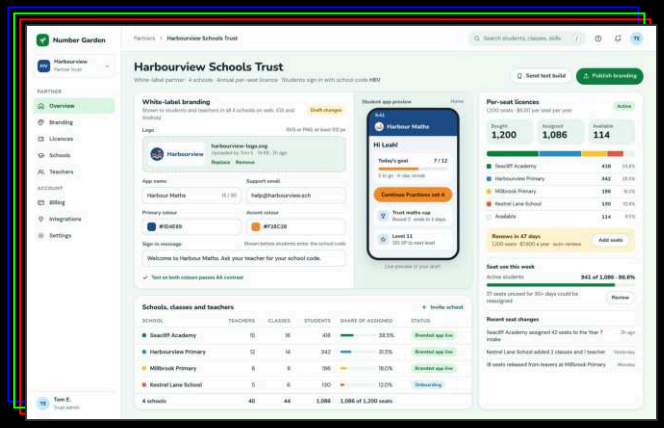
ONE PLAN, EVERY DEVICE

Representative interface · client work under NDA



TEACHER VIEW: CLASS PROGRESS

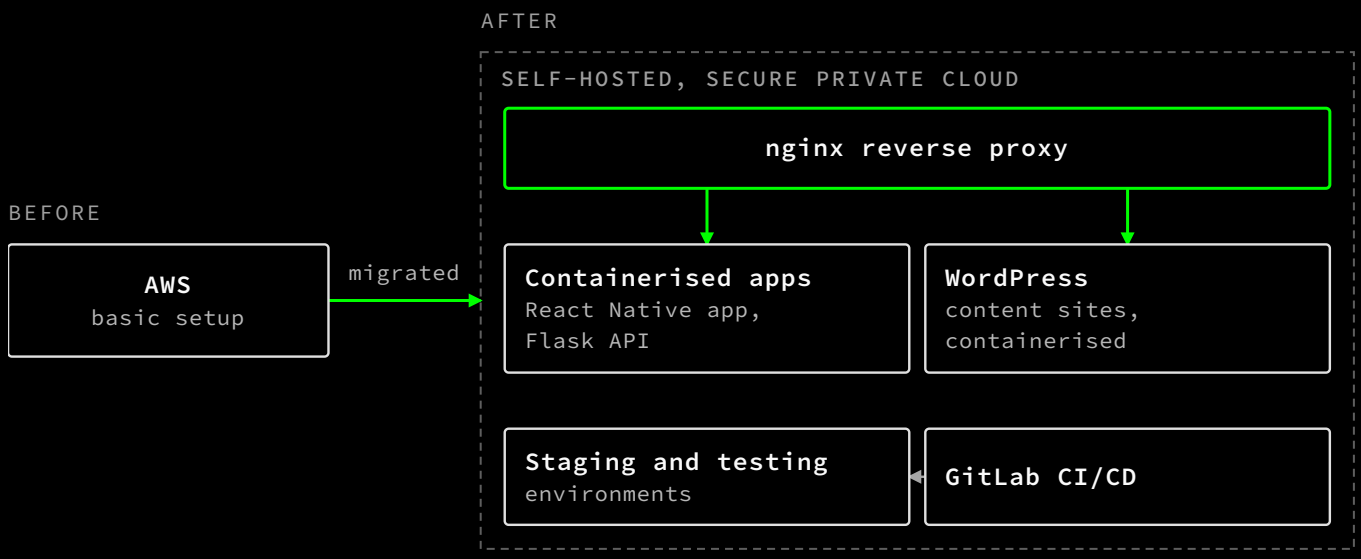
Representative interface · client work under NDA



PARTNER ADMIN: WHITE-LABEL BRANDS AND PER-SEAT LICENCES

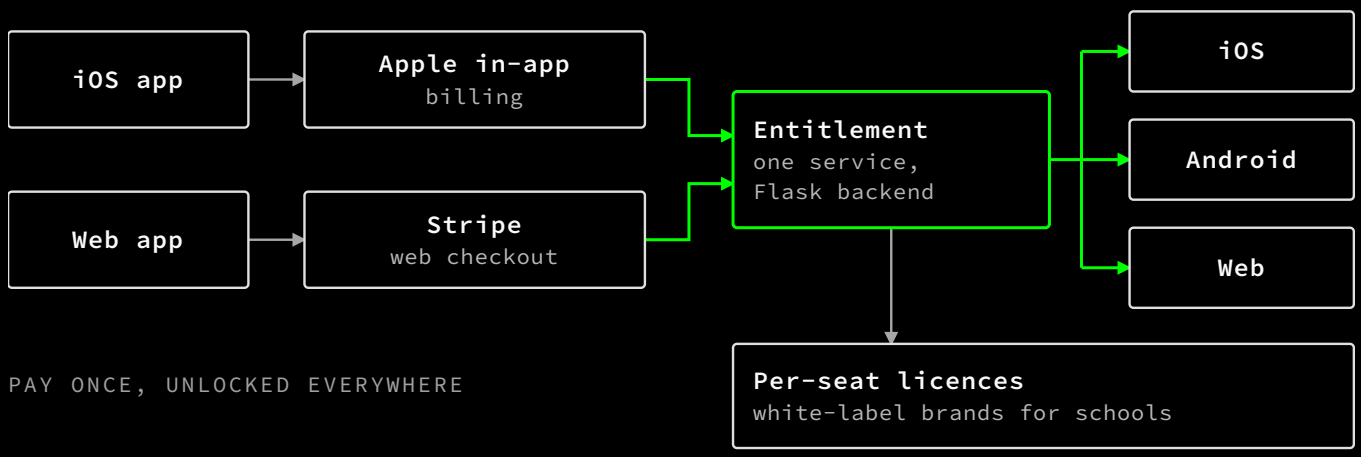
Representative interface · client work under NDA

INFRASTRUCTURE, BEFORE AND AFTER



About 70% lower infrastructure cost, zero downtime

ENTITLEMENT: ONE SUBSCRIPTION ON EVERY DEVICE



PAY ONCE, UNLOCKED EVERYWHERE

PURCHASE PATH: APPLE OR STRIPE INTO ONE ENTITLEMENT SERVICE, OUT TO EVERY DEVICE

OUTCOME

About 70% lower infrastructure cost, with zero downtime through the migration, and single-handed ownership of servers, infrastructure and security afterwards.

STACK React Native · Flask · MongoDB · Apple in-app billing · Stripe · Docker · nginx · Linux · GitLab CI/CD · AWS · WordPress · Jira · Microsoft 365

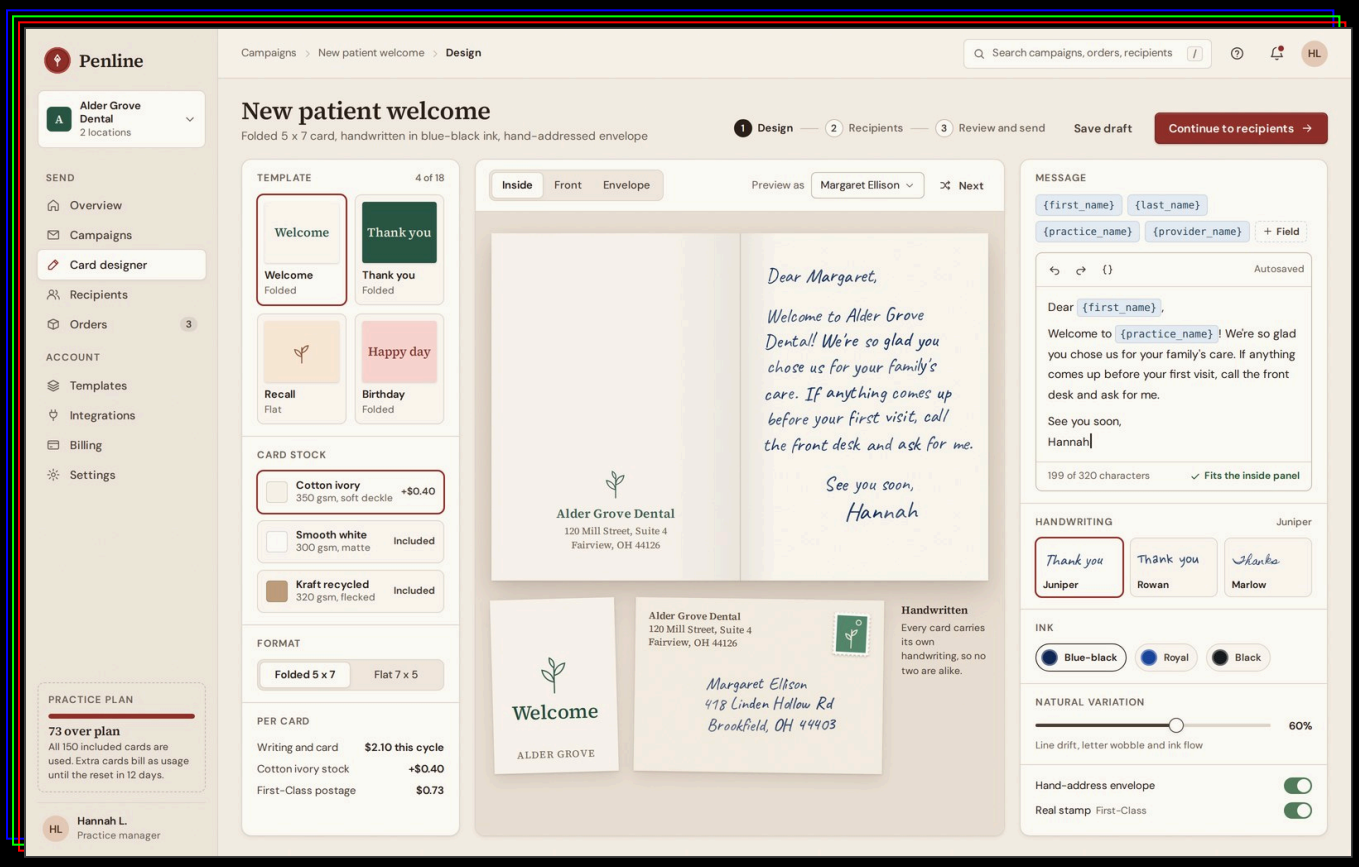
Bulk handwritten and printed card automation, from order to mailbox

INDUSTRIES healthcare (the main client vertical) · business tools · marketing and customer engagement

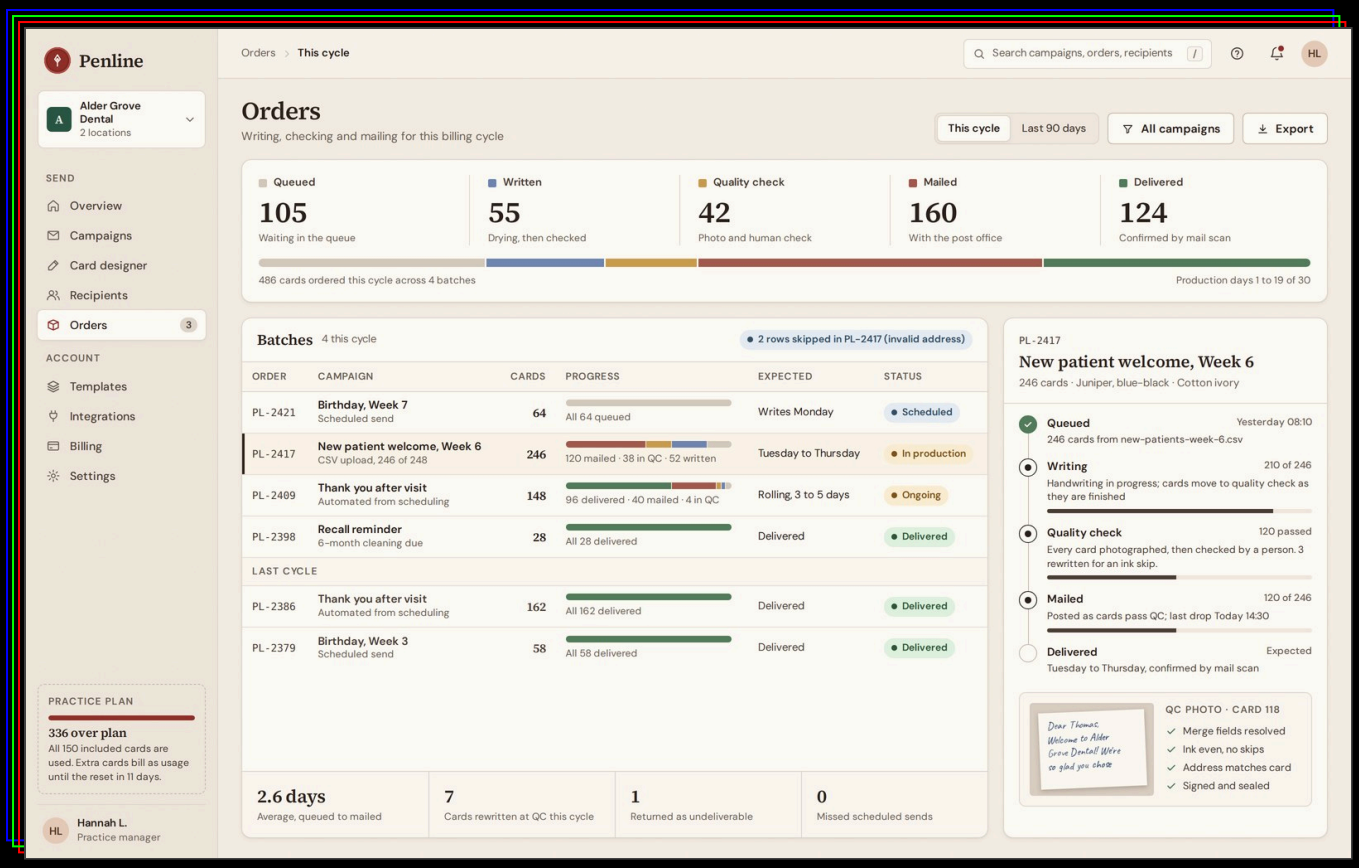
ROLE Full-Stack Engineer and Technical Lead

The problem. Businesses, healthcare practices above all, send physical handwritten or printed letters and cards in bulk to keep patients and customers close. Ordering, making and sending them was manual work, and it does not scale by hand.

What I built. A platform that automates the full path: ordering, making and sending handwritten letters, cards and printed pieces. Customers place bulk orders, and the platform carries each one through to the mailbox. Billing runs on monthly subscriptions together with dynamic billing. Vue front end, Express back end, MongoDB, all hand-coded. I led the whole design team on the front end.



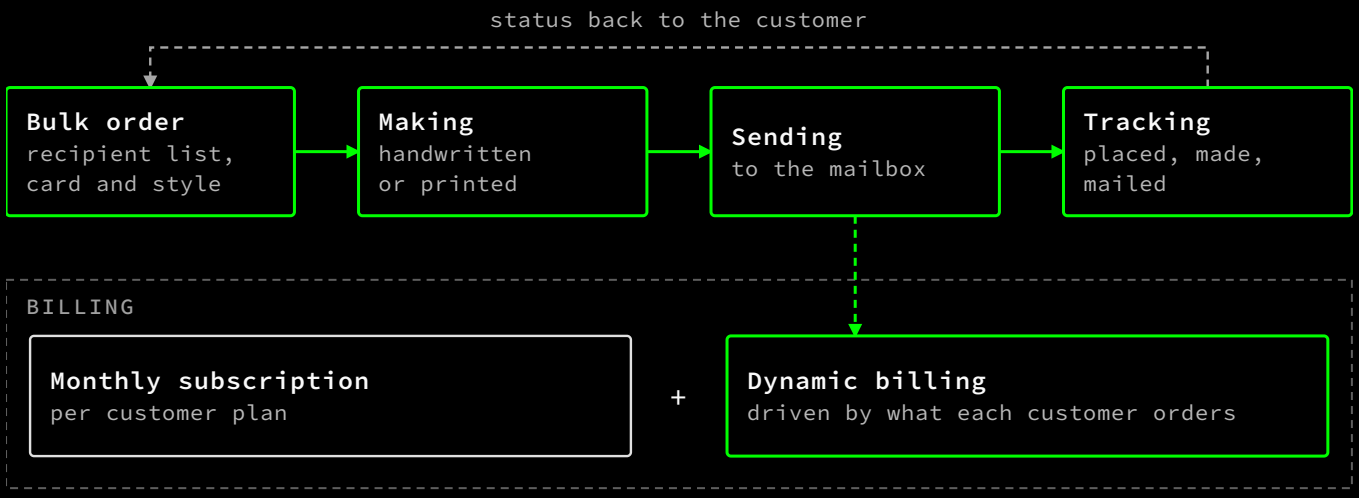
CARD DESIGNER: STYLE, HANDWRITING AND LIVE PREVIEW Representative interface · client work under NDA



ORDER TRACKING: PLACED, MADE, MAILED

Representative interface · client work under NDA

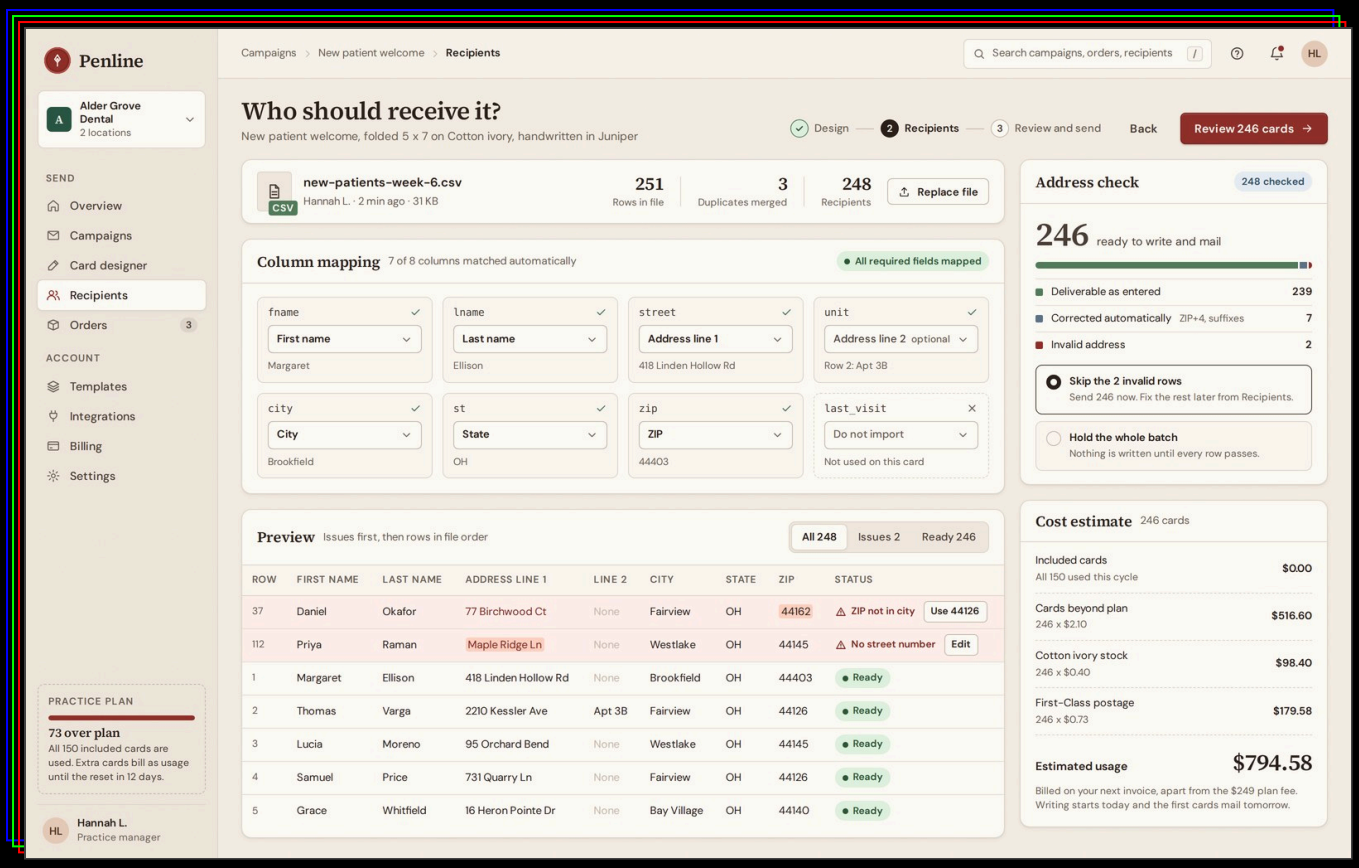
ORDER TO MAILBOX



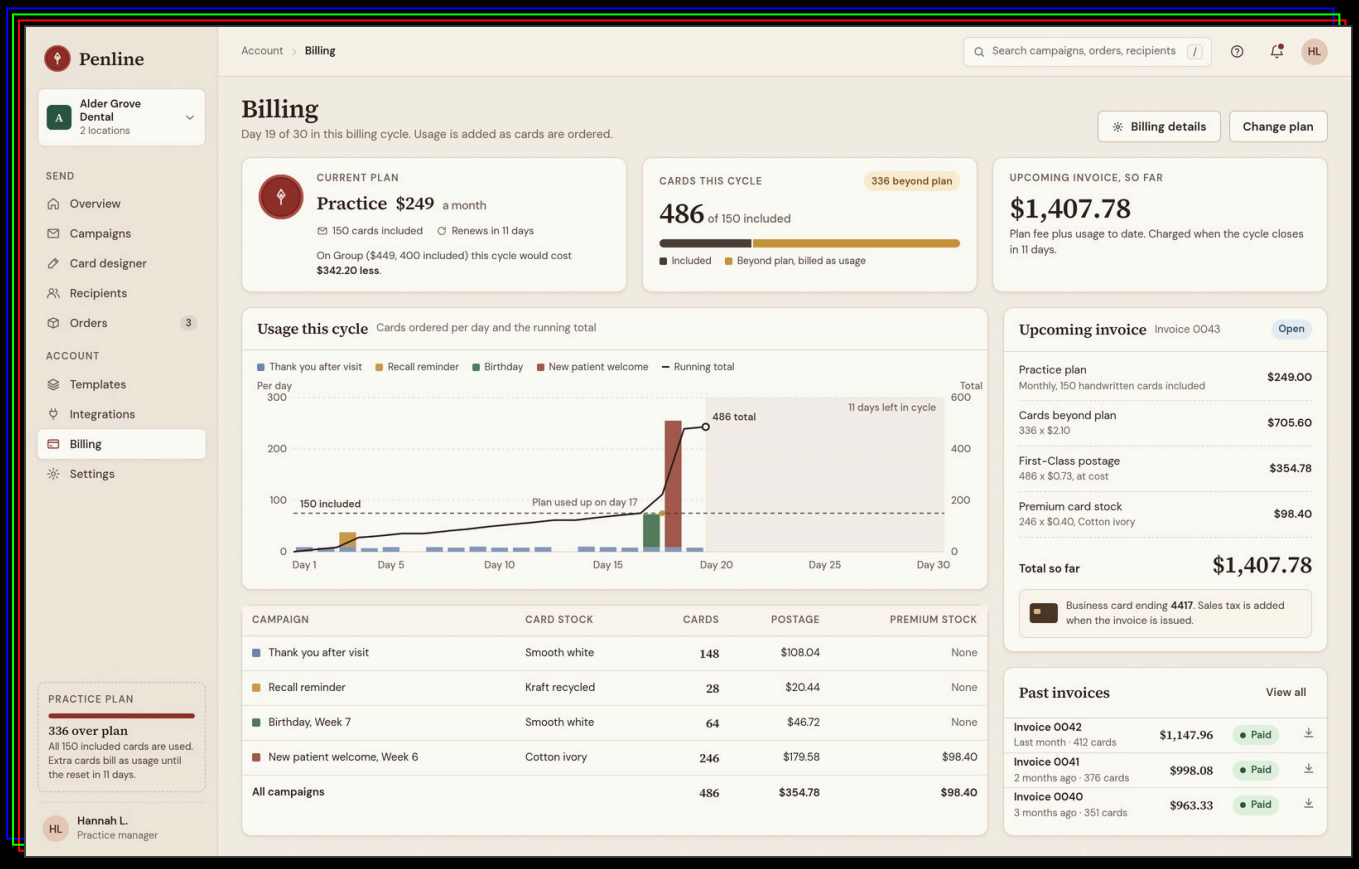
Vue front end, Express back end, MongoDB, GitLab CI/CD to staging and production

THE HARD PARTS

- **A physical product behind a web order.** Software has to coordinate a real-world production and mailing process, in volume, for each order.
- **Two billing models at once.** Monthly subscriptions plus dynamic billing driven by what each customer actually orders.
- **Hand-coded end to end** across the Vue front end, the Express back end and MongoDB.
- **Proposal and delivery.** I wrote the technical proposal that won the client, and ran GitLab staging and production CI/CD.



BULK ORDER: RECIPIENT LIST UPLOAD AND ADDRESS CHECK Representative interface · client work under NDA



BILLING: MONTHLY PLAN PLUS DYNAMIC CHARGES Representative interface · client work under NDA

STACK Vue.js · Express.js · MongoDB · GitLab CI/CD

A multi-role API platform for evaluating college award letters, with a student-facing app

INDUSTRIES EdTech · higher education · FinTech and financial aid · document AI

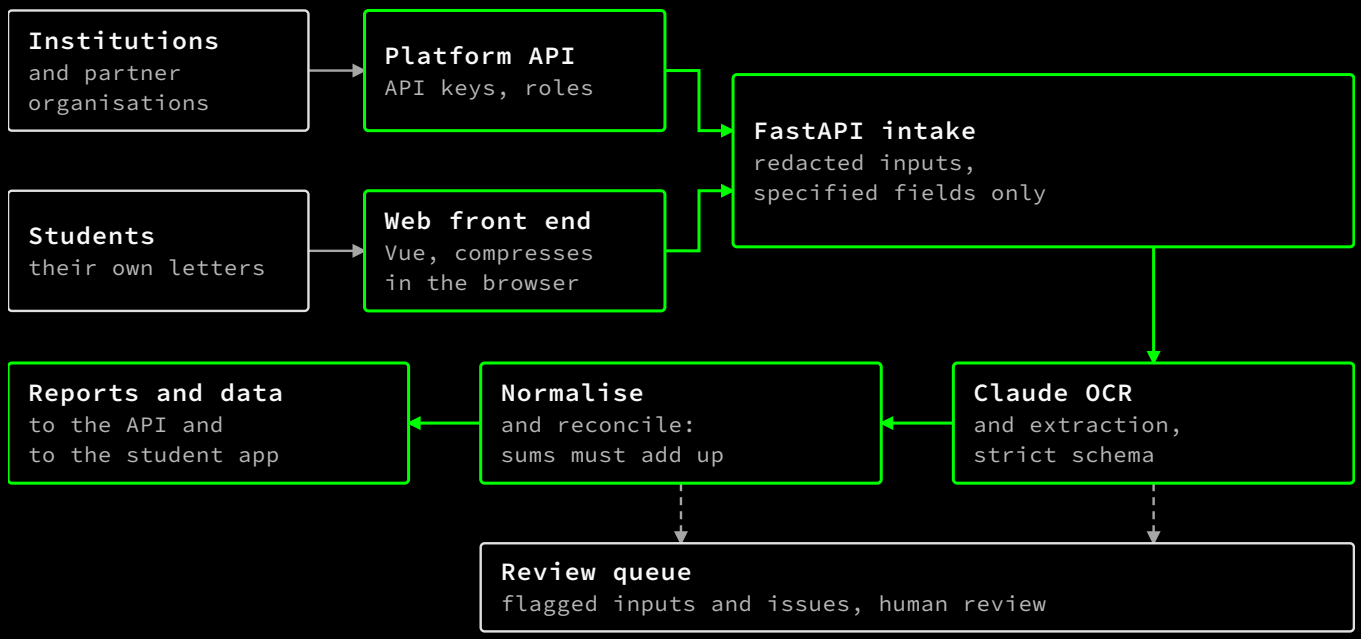
ROLE Full-Stack Engineer. I also owned the proposal and client onboarding.

The problem. Every college writes its financial-aid award letter in its own layout and terms, and letters reach families in any format, including phone screenshots. Students need to compare offers on the same basis. Educational organisations that want to run their own evaluations need structured, comparable data from those letters rather than PDFs and images.

What I built. A multi-role platform that serves an API, with a front-end implementation of that API on top. Two surfaces share one pipeline:

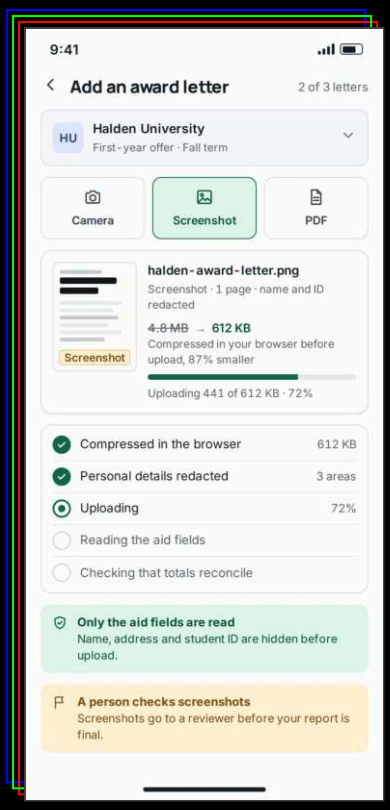
- **The platform and its API.** Other educational organisations use the platform's data through the API to create their own evaluations.
- **The student app.** A front-end implementation of the same API where students evaluate their own college award letters and compare offers themselves.

MULTI-ROLE API PLATFORM



Vue, FastAPI, Claude API (Claude Sonnet)

SHARED PIPELINE: FASTAPI INTAKE, CLAUDE OCR AND EXTRACTION, NORMALISATION AND RECONCILIATION



STUDENT UPLOAD ON A PHONE

Representative interface · client work under NDA

HOW A LETTER MOVES THROUGH IT

01 **Upload.** The Vue front end compresses the image or document in the browser, then uploads it.

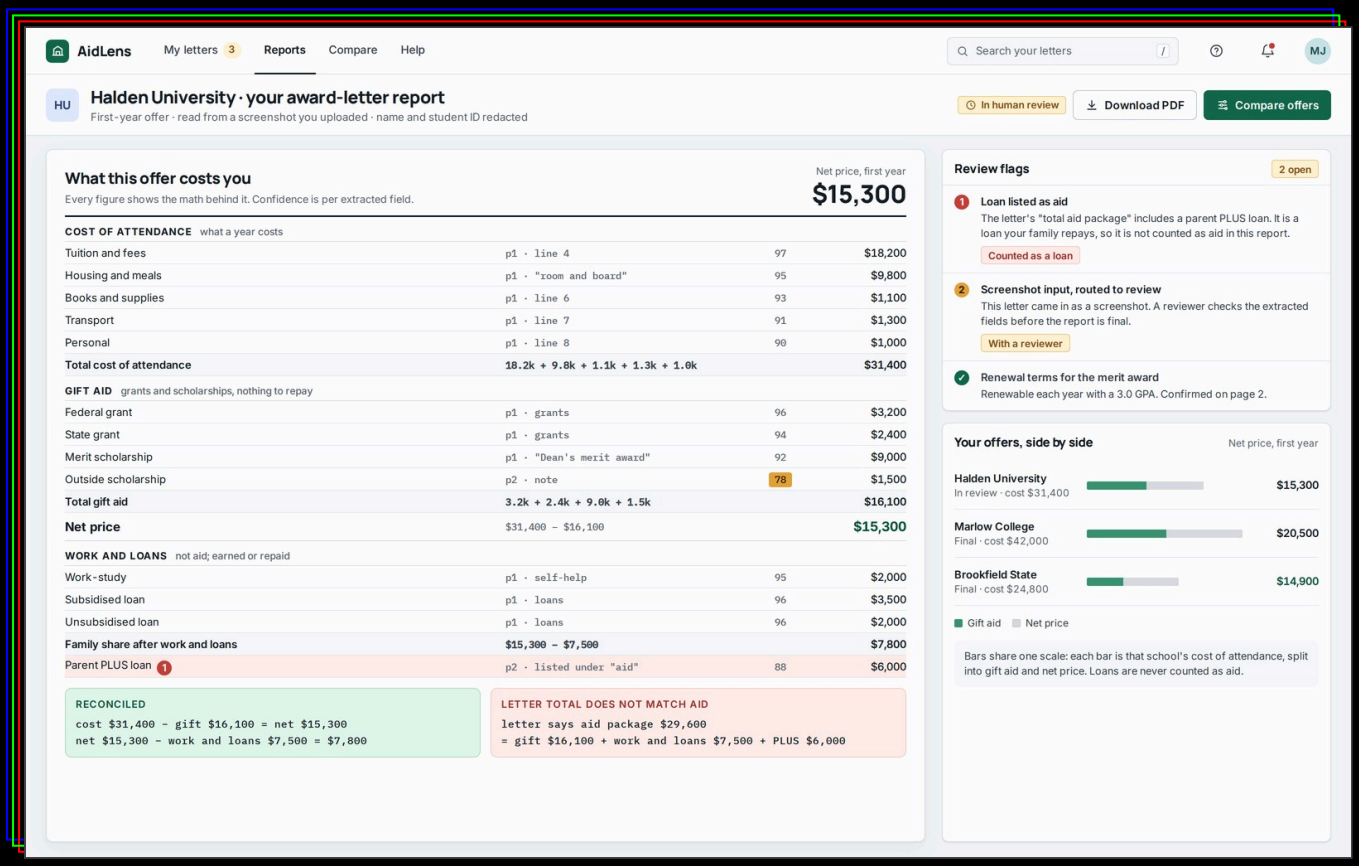
02 **Intake.** FastAPI receives the upload. Inputs arrive redacted, and only the specified fields are processed.

03 **OCR and extraction.** The Claude API reads the letter and fills a strict extraction schema.

04 **Comparison.** Rules normalise each school's terms onto one vocabulary, and sums must reconcile before anything is reported.

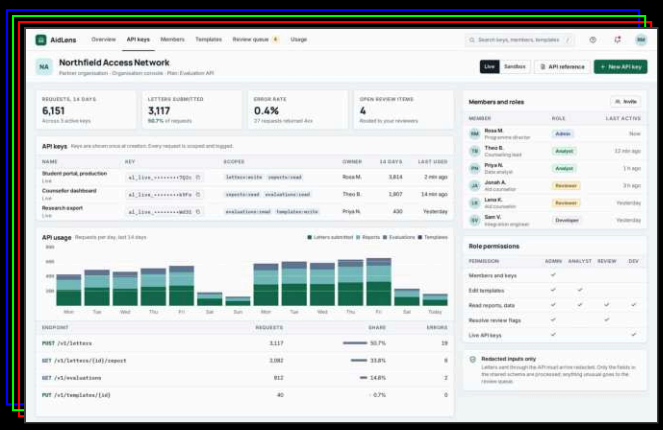
05 **Report.** The report includes the math behind every figure.

06 **Review.** Anything unusual is flagged and routed to a person.



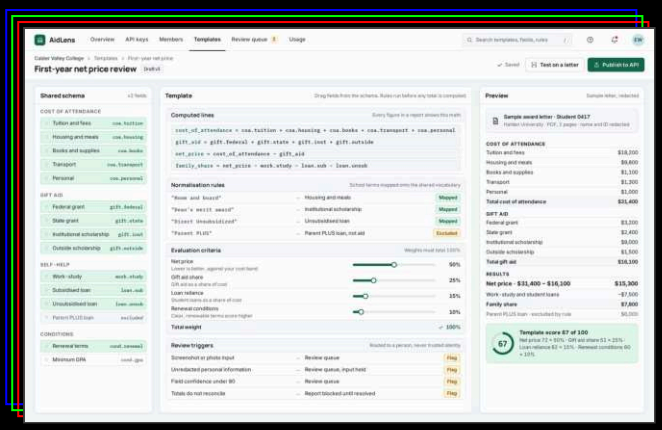
STUDENT AWARD-LETTER REPORT: RECONCILED TOTALS AND REVIEW FLAGS

Representative interface · client work under NDA



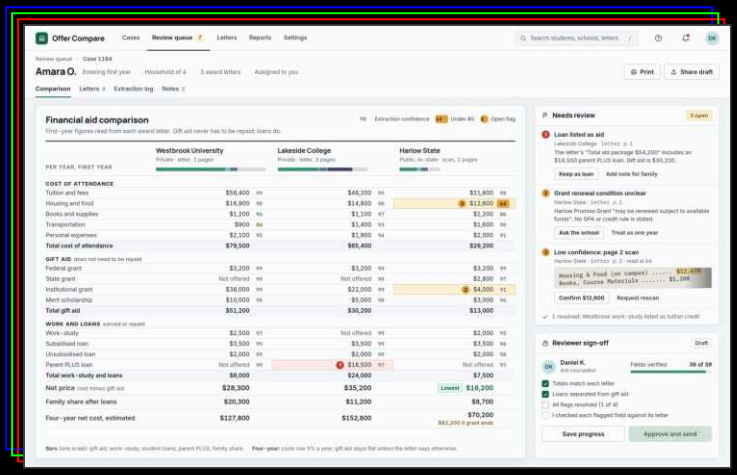
ORGANISATION CONSOLE: API KEYS, ROLES AND USAGE

Representative interface · client work under NDA



INSTITUTION EVALUATION-TEMPLATE BUILDER

Representative interface · client work under NDA



OFFER COMPARISON: RECONCILED TOTALS AND REVIEW FLAGS

Representative interface · client work under NDA

THE HARD PARTS

- **One platform, several roles.** The same data serves educational organisations through the API and students through the web app, so the API is the product and the student app is one client of it.
- **Letters in any format.** PDFs, scans and phone screenshots all have to land in the same schema. Front-end compression keeps uploads light, and screenshots are flagged for review rather than trusted.
- **Privacy by design.** Names and sensitive details arrive redacted, and the model only processes the fields the schema specifies.
- **Every school uses its own terms.** A strict shared extraction schema, with rules that normalise each school's terms, turns different letters into comparable numbers.
- **Numbers that must add up.** Sums reconcile before they reach a report, and the report shows the math behind them, so a student or a reviewer can trace any figure.
- **Human review where it counts.** Screenshots, unredacted personal information and any pipeline issue are flagged and routed to a person.
- **Model and cost.** The Claude API (Claude Sonnet) runs extraction today, with plans to self-host the models.
- **Demo and upsell.** A separate demo and upsell app on Vue, Express and MongoDB, with its own CI/CD pipeline.
- **Owning the chain.** I owned the proposal and client onboarding as well as the build.

STACK Vue.js · FastAPI · Claude API (Claude Sonnet) · Express.js · MongoDB · CI/CD

READ MORE abchaudary.me/blog/claude-api-ocr-financial-aid-letters

An AI coach that keeps a personal trainer's voice

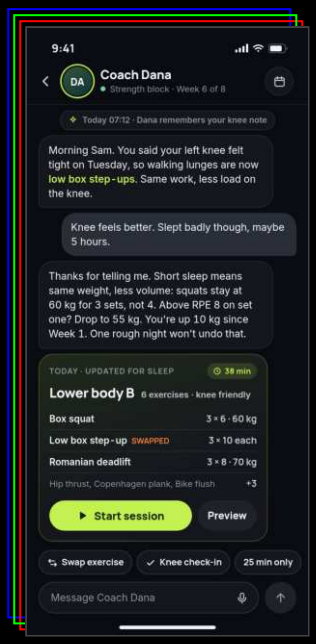
INDUSTRIES coaching · personal training · consumer AI

ROLE Full-Stack Engineer

What I built. An AI chat app that speaks in a personal trainer's voice and coaches users in life and professionally. The trainer's voice is held by prompting, retrieval over the trainer's own content and fine-tuning. The app keeps memory of each user and tracks their goals. Vue, Express and MongoDB, with GitLab CI/CD pipelines for staging and production.

The hard part. Voice consistency. A coach that drifts into generic assistant language loses the trainer's brand, so the voice is enforced in three layers: prompting, retrieval and fine-tuning.

STACK Vue.js · Express.js · MongoDB · GitLab CI/CD



COACH CHAT WITH MEMORY

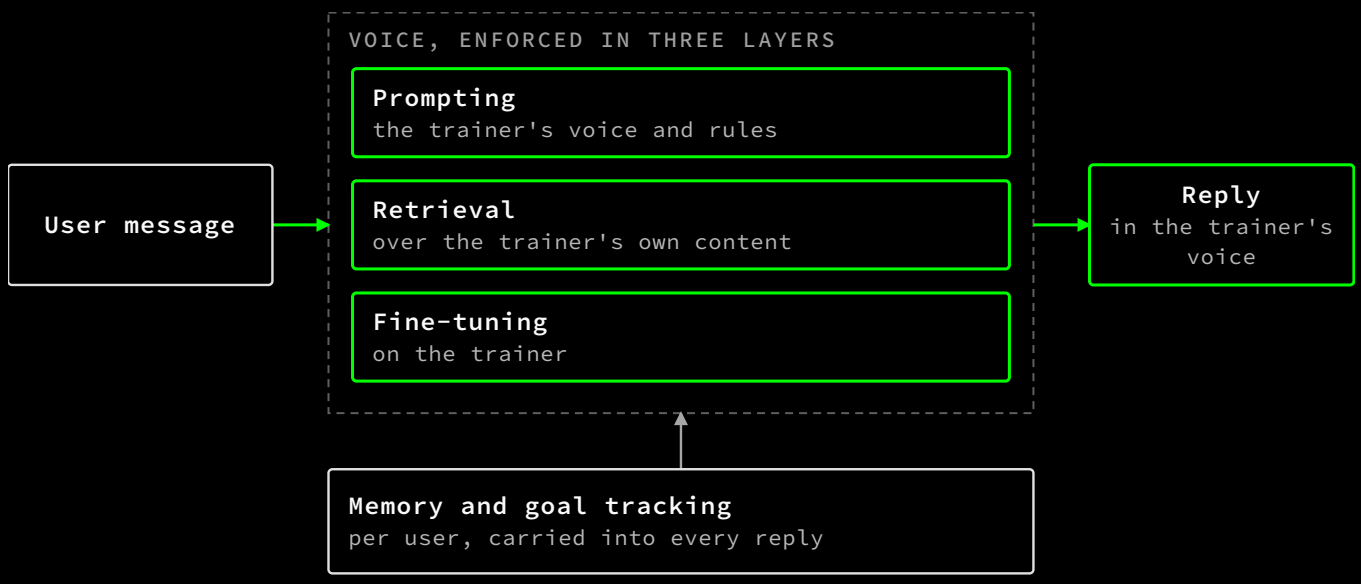
Representative interface · client work under NDA



GOALS AND PROGRESS

Representative interface · client work under NDA

THE VOICE STACK



Vue, Express, MongoDB; GitLab CI/CD to staging and production

3.6 CLIENT WORK UNDER NDA

Platform migrations: client systems moved to new infrastructure

INDUSTRIES across client verticals

ROLE Principal Engineer, Infrastructure and DevSecOps

Client platforms moved onto new infrastructure by exporting their existing servers and data onto new servers. These were migrations, not rebuilds: the work is moving a live system across without disrupting the client's users.



MIGRATIONS, NOT REBUILDS

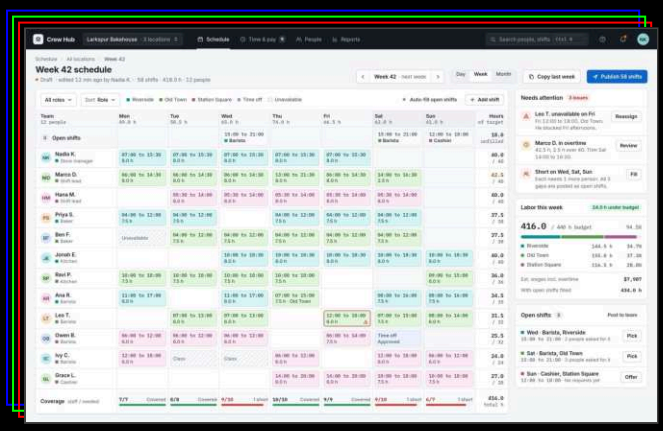
3.7 CLIENT WORK UNDER NDA

Workforce tools: scheduling, time and payroll inputs in one place

INDUSTRIES business tools · workforce management · HR operations

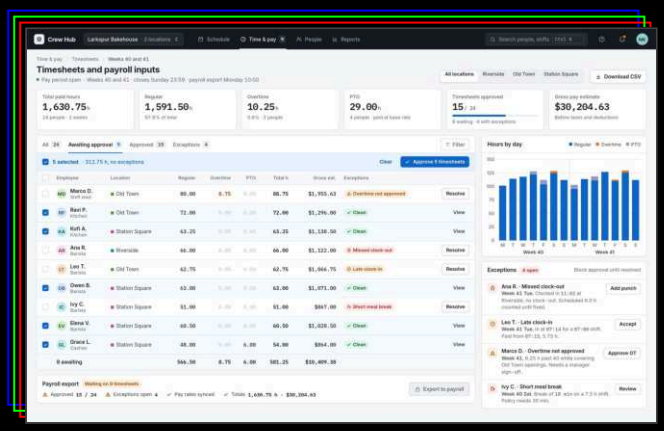
ROLE Full-Stack Engineer

New-build internal tools for workforce management: scheduling, time tracking, payroll inputs and dashboards.



SCHEDULE BOARD

Representative interface · client work under NDA



TIMESHEET APPROVAL AND PAYROLL INPUTS

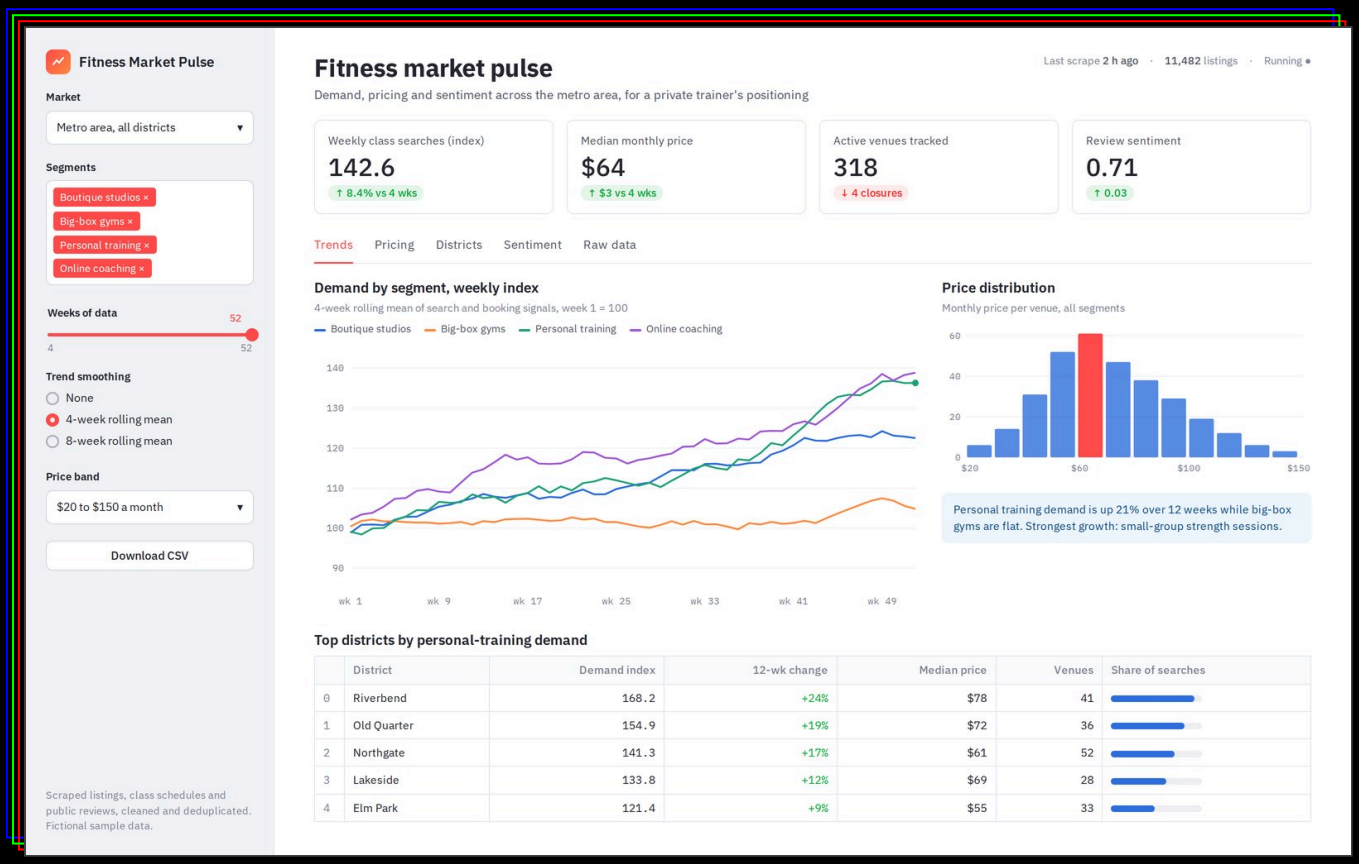
Representative interface · client work under NDA

Fitness-market analytics from scraped data to dashboard

INDUSTRIES fitness · market research · data analytics

ROLE Data Engineer and Analyst

Scraped and aggregated fitness-market data, cleaned and evaluated trends, and shipped interactive Streamlit dashboards for a private trainer.



MARKET DASHBOARD WITH TREND LINES

Representative interface · client work under NDA

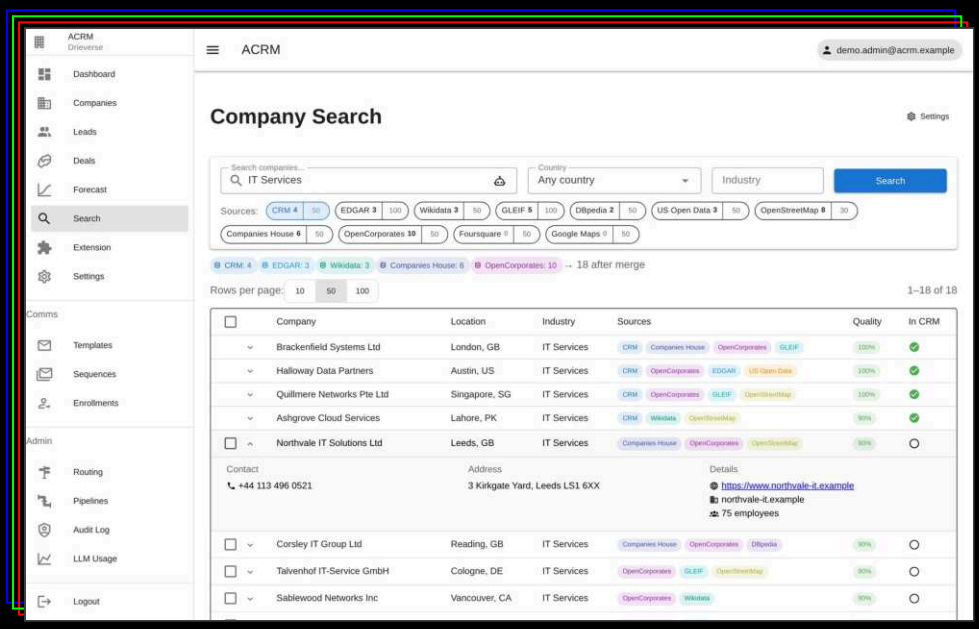
STACK Python · web scraping · Pandas · Streamlit

EARLIER PROFESSIONAL WORK

Earlier platforms, product suites and device work.

ACRM: multi-country CRM and lead-intelligence platform

ROLE Development and Technical Lead, leading a team of 20 across 5 concurrent products



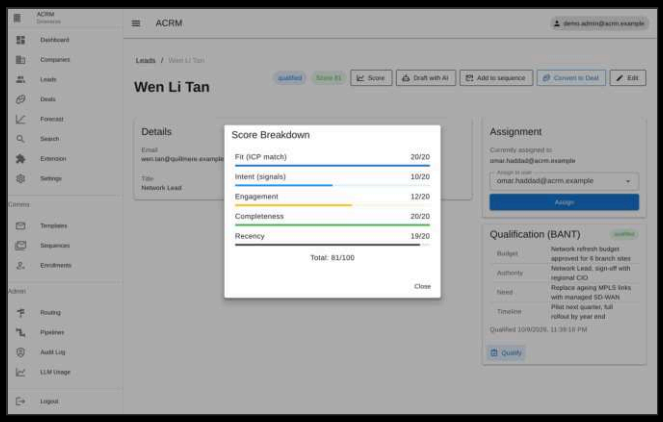
SEARCH RESULTS

STACK

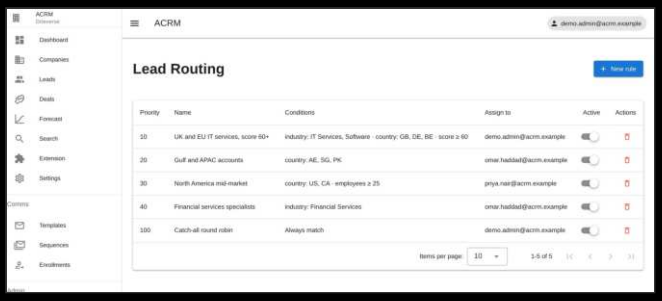
- FastAPI
- SQLAlchemy 2.0 async
- Pydantic v2
- Nuxt 4
- Vuetify 4
- PostgreSQL 16
- Chrome extension
- MCP
- LLM enrichment
- n8n

- An internal B2B lead-intelligence platform and CRM serving seven business lines across 10 target countries.
- **Lead enumeration:** one search fans out in parallel to the internal database and 12 external sources (company registries in the UK and US, OpenStreetMap, maps listings and more), backed by 28 scheduled collectors across registries, directories and job boards. Results are merged and de-duplicated by domain and normalised company name.
- **Enrichment and scoring:** leads are enriched in tiers (internal history, then company registries, then an LLM), scored 0 to 100 on five factors against an editable ideal-customer profile, and routed by rules to the right business line.

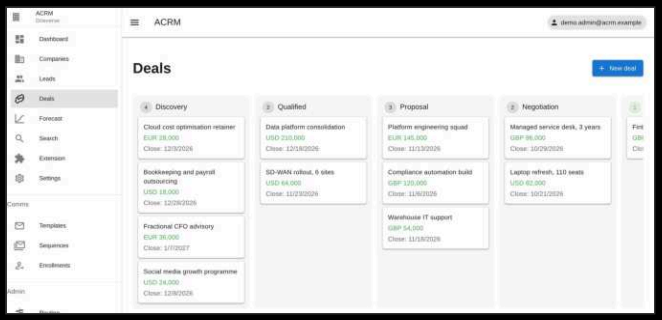
- **CRM:** email sequences, a deals pipeline with forecasting, AI-drafted social replies, and a Chrome extension that captures leads from the user's own browser tabs, driven by the web app.
- Nine domain schemas in PostgreSQL 16 for auth, leads, scraping, audit, enrichment, routing, deals, communications and messaging; an MCP server exposes 50 tools to AI agents.
- **Scale:** about 63,000 lines of code, 149 API routes, 1,065 backend tests and 331 front-end unit and end-to-end cases.



LEAD DETAIL WITH SCORE BREAKDOWN



ROUTING RULES



DEALS PIPELINE

Enterprise SaaS suite: CRM, CMS, accounting and HRMS

- Built a suite of business products alongside bespoke client projects; introduced CI/CD and standardised deployments, tripling release frequency.
- Build, deploy and rollback systems that closed 80% of open critical and high vulnerability findings and shortened mean time to recovery.

IoT smart-farming platform

- End to end, from firmware to dashboard: C/C++ firmware on ESP32 and Raspberry Pi, MQTT and REST telemetry, a Django backend and a Vue.js real-time dashboard.

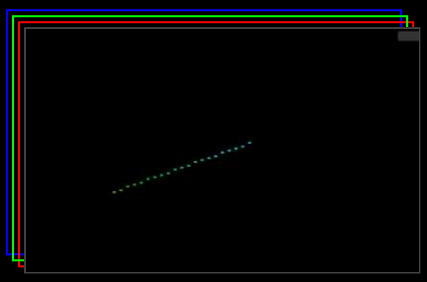
READ MORE abchaudary.me/blog/esp32-wifi-provisioning-mqtt-reconnect

Early product portfolio

33+ web and mobile applications, including Chef Finder, Gift Box, Personal Security, six video-streaming apps and four e-commerce platforms, plus 40+ websites and stores on Laravel, CodeIgniter, Django, Vue.js and Bootstrap, and console automation tools in C/C++ and Python.

CREATIVE AND OPEN CODING PROJECTS

Independent projects built for the browser: real-time graphics, audio, computer vision and on-device ML. All live and open.

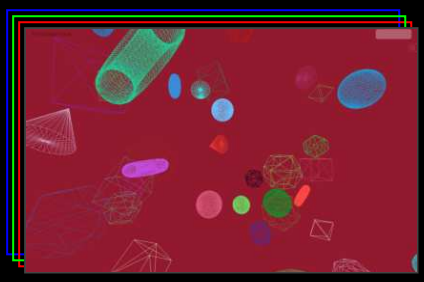


MusicGen

musicgen.abchaudary.me

A browser instrument with a touch theremin, a synth keyboard and camera-driven play; MIDI support; works offline as a PWA.

Vue.js · Three.js · Tone.js · Web MIDI

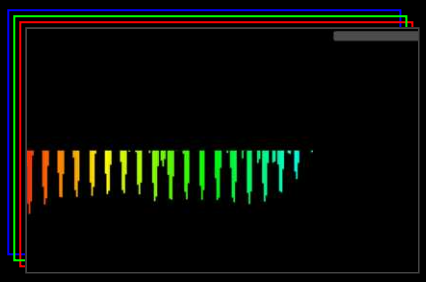


Psychedelica

psychedelica.abchaudary.me

Generative 3D visuals with reactive soundscapes.

Vue.js · Three.js · Web Audio



AudioViz

audioviz.abchaudary.me

Real-time audio analysis as dynamic waveforms and visualisations.

Vue.js · Web Audio API · Canvas

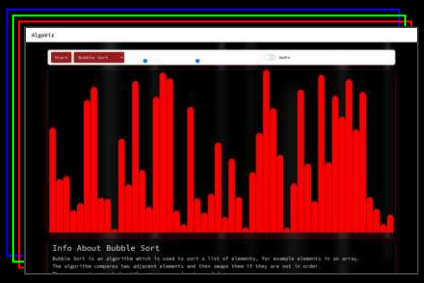


Emotions AI

emotions-ai.abchaudary.me

Real-time facial-emotion analysis from the webcam, entirely on device.

TensorFlow.js · WebRTC



AlgoViz

algoviz.abchaudary.me

Animates algorithms step by step for learning data structures and algorithms.

React



abchaudary.me

abchaudary.me

The animated portfolio and technical blog, described in full in 1.11.

Next.js · Payload CMS · GSAP · Three.js

EXPERIENCE, EDUCATION AND SKILLS

Principal Full-Stack Engineer & DevSecOps Architect

Fuentes Tech, Hillsboro, Oregon (Remote)

The $600M+ acquisition product; 99.95% uptime and 45% lower cloud spend; 80% fewer critical vulnerabilities; 95% on-time delivery and 30% fewer production defects; LLM, RAG and agentic automation; client platform migrations; still leading teams.

Development & Technical Lead

DrieVerse Tech, Lahore

Led 20 people across 5 concurrent products; 15 production releases; the multi-country CRM; the AGENTS.md engineering standard.

Full-Stack Engineer & Systems Expert

Intersoft AI, Lahore

CRM, CMS, accounting and HRMS products; 3x release frequency; 75% revenue increase contributed to.

Freelance Full-Stack Developer

Self-employed, Remote

33+ apps, 40+ websites and stores, the IoT smart-farming platform.

EDUCATION AND CERTIFICATIONS

Master of Science in Computer Science	The Superior University, Lahore
Bachelor of Science in Computer Science	The Superior University, Lahore
CCNP Routing & Switching; CCNP Security	Corvit Systems, Lahore

LANGUAGES	Python, JavaScript, TypeScript, PHP, C/C++, Bash, SQL
FRONTEND	Next.js, React, React Native, Vue.js, Nuxt.js, Tailwind, Vuetify, Three.js, GSAP
BACKEND	Django, FastAPI, Flask, Express, Node.js, Laravel, REST, GraphQL
CLOUD AND DEVOPS	AWS (EKS, Lambda, RDS, S3, CloudFront, IAM), GCP, Azure, Docker, Kubernetes, Helm, ArgoCD, GitOps, Terraform, Ansible, GitHub Actions, GitLab CI/CD, Linux, Nginx, VMware, Proxmox, self-hosted cloud, SRE
AI/ML	LLMs, RAG, vector databases, Qdrant, Ollama, Qwen3, BGE-M3, MCP, agentic workflows, n8n, prompt engineering, TensorFlow, Pandas, scikit-learn
AI CODING TOOLS	Claude Code, Cursor, Cline, Gemini CLI, GitHub Copilot, AGENTS.md and rules-file conventions
DATASTORES	PostgreSQL, MySQL, MongoDB, Redis, Elasticsearch, Supabase
SECURITY AND NETWORKING	IAM, HashiCorp Vault, VPC, VPN, TLS, OWASP, SIEM, WAF, SSO, OAuth2, SAST/DAST, Trivy, Snyk, fail2ban, UFW, Cloudflare
OBSERVABILITY	Prometheus, Grafana, ELK, OpenTelemetry, Loki
EMBEDDED	ESP32/ESP8266, Zephyr RTOS, BLE, MQTT, circuit design



CONTACT

Available for senior full-stack, DevSecOps, AI and technical leadership roles, remote with full working-hours overlap for US, UK and EU teams.

EMAIL	abchaudary@outlook.com
LINKEDIN	linkedin.com/in/abchaudary
SITE	abchaudary.me