

CASE STUDY

Brain Station 23

Building an AI-Powered Legal Claims Engine in 30 Days

How a single full-stack engineer at Brain Station 23 designed, built, and shipped **ClaimRight** — a multi-jurisdiction consumer claims platform spanning 9 legal frameworks, 5 languages, and full Stripe Connect payouts — using AI-augmented engineering.

177

Files Shipped

41.5K

Lines of Code

61+

API Endpoints

9

Legal Frameworks

Industry: Legal Technology · Consumer FinTech

Engagement Type: AI-Native Product Build · Full-Stack Delivery

Date: April 2026

Prepared by: Brain Station 23 — Innovation Practice

Executive Summary

Consumer legal claims — flight delays, bank fee abuses, telecom overcharges, landlord deposit disputes — are a **multi-billion-dollar pool of unclaimed money**. The barriers are not legal complexity but operational friction: paperwork, jurisdictional rules, follow-up cycles, and the cost of legal representation. **ClaimRight** dissolves that friction with AI.

Brain Station 23 partnered with the founder to deliver a **production-grade MVP** covering nine legal frameworks (EU261, US DOT, UK Consumer Rights, Bank Fees, Insurance, Product Liability, Subscription Traps, Wage Theft, Warranty) in a single sprint. The result: a monolithic-but-modular FastAPI backend, a React 18 client localized into 5 languages, an AI document generation pipeline, a fraud detection layer, and Stripe Connect-driven escrow — all running against a seeded PostgreSQL database in local development on day one.

30 days MVP to runnable system	15% Commission — lowest in industry	5 Localized languages
9 Legal frameworks live	19 Domain services	100% Containerized & K8s-ready

The headline: A delivery footprint that would traditionally require a 6-person team and a 4–6 month roadmap was compressed into a single AI-augmented engineer working inside Brain Station 23's delivery framework.

Client & Project Snapshot

Product	ClaimRight — AI-Powered Consumer Legal Claims Engine
Client Profile	Early-stage LegalTech startup (founder-led)
Mission	Democratize access to justice by automating consumer claims end-to-end
Business Model	Contingency fee — 15% standard, 12% preferred (3+ wins), 10% VIP

Target Geographies	EU, UK, US (Phase 1); LATAM and DACH expansion roadmap
Phase 1 Vertical	Flight delay & cancellation claims (EU261 / US DOT)
Engagement	AI-native product build — architecture, backend, frontend, DevOps, legal templating
Delivery Model	Solo-engineer-with-AI delivery, run inside Brain Station 23's accelerator playbook

The Challenge

The founder approached Brain Station 23 with an ambitious thesis: 80% of small-dollar consumer claims are abandoned because the cost — in time, paperwork, and lawyer fees — exceeds the recovery. An AI-first platform could change the unit economics, but only if it could be built fast enough to validate the model before competitors (DoNotPay, AirHelp, Flightright) extended their lead.

Specific constraints we had to design around

- **Regulatory minefield.** Unauthorized practice of law (UPL) rules prohibit non-attorneys from giving "legal advice." The product had to be framed as *claims assistance*, never legal counsel, and rely on a *claims assignment* model so that the platform pursues claims it owns.
- **Multi-jurisdictional logic.** EU261, US DOT, and UK Consumer Rights all use different eligibility tests, evidentiary standards, and remedy formulas. The engine had to switch behavior per jurisdiction without forking the codebase.
- **Money handling without a banking license.** Settlement funds cannot touch the platform's balance sheet directly. Stripe Connect was the only viable path, with explicit escrow, commission split, and payout flows.
- **Trust UX.** Users only convert if the product feels *competent*. That demanded a real-time conversational intake, a polished claim wizard, transparent escalation status, fraud safeguards, and quality scoring that users can see.
- **One-engineer execution.** The founder is the team. The architecture had to be operable, debuggable, and extendable by one person — without sacrificing production posture.

The brief in one sentence: Ship a production-credible legal-claims platform — with AI document generation, payments, multi-language UX, and K8s-deployable infrastructure — before runway pressure forces compromise.

Our Approach

Brain Station 23 ran the engagement using its **AI-Native Delivery Playbook**: a senior engineer paired with Claude (Anthropic) and GPT-4o inside an opinionated FastAPI + React scaffold, with continuous validation against a real PostgreSQL instance from day one.

1. Architect first, generate later.

Before a single LLM-assisted line was written, the engagement opened with a formal **Software Requirements Specification** and a **Problem & Solution Risk Analysis** — both delivered as consulting-grade PDFs to align scope, legal posture, and architectural boundaries. Those documents became the prompt-context anchor for the AI- augmented build phase, dramatically reducing rework.

2. Modular monolith, microservice-ready.

A single FastAPI app was chosen over premature microservices — faster to ship, simpler to debug, but with strict module boundaries (`models`, `services`, `api`, `legal`) so that any service can be lifted out later without rewrite.

3. Legal logic as code, not config.

Each jurisdiction lives as a first-class Python module under `app/legal/frameworks/`. EU261, US DOT, UK Consumer, Bank Fees, Insurance, Product Liability, Subscription Traps, Wage Theft, and Warranty each implement an eligibility interface, a remedy calculator, and a template binding. Adding a new framework is an additive change, not a refactor.

4. AI where it earns its keep.

LLMs are used surgically: **conversational intake**, **document generation** (demand letters, regulatory complaints, court forms), **OCR** for evidence uploads, **fraud detection**, and **quality assurance** scoring. Deterministic logic still owns eligibility, money, and escalation timing — the LLM never decides whether a claim is paid.

5. Production posture from day one.

Even at MVP, the deliverable shipped with Alembic migrations, JWT auth + bcrypt, Stripe Connect webhooks, SendGrid email, Redis-backed task queues, Dockerfiles for backend and frontend, a 10-manifest Kubernetes bundle, and GitHub Actions CI — so the leap from "running locally" to "running on cluster" is a configuration change, not an engineering effort.

Solution Architecture

The platform is structured in five horizontal layers and one cross-cutting AI services tier:

Layer	Technology	Responsibilities
Client	React 18, TypeScript, Vite, Tailwind, Zustand, React Query, react-hook-form, Zod	13 user-facing pages, 21 components, claim wizard, conversational chat, admin console, multi-language UI (EN/DE/ES/FR/PT)
API	FastAPI 0.115, Pydantic v2, JWT (PyJWT + bcrypt), CORS	15 routers, ~61 endpoints, structured exception handling, OpenAPI auto-docs at /docs
Domain Services	Pure Python service layer (19 services)	Claim lifecycle, eligibility, escalation, settlement, commission, cash flow, cross-sell, referrals, fraud detection, QA
Legal Engine	Per-jurisdiction Python modules + Jinja2 templates	9 frameworks, 8 legal templates (demand letters, regulatory complaints, court forms), jurisdiction router
Persistence	SQLAlchemy 2.0 (async), Alembic, PostgreSQL 15, Redis 7	12 domain models, audit log, event-sourced claim status, Redis-backed RQ/Celery queue

AI Services Tier	OpenAI GPT-4o + Anthropic Claude via LangChain 0.3	Conversational intake, document generation, OCR, fraud detection, QA scoring, eligibility hints
Integrations	Stripe Connect, SendGrid, S3 / MinIO, FlightAware-class data feeds	Escrow + payouts, transactional email, evidence storage, flight verification
DevOps	Docker Compose, 10 K8s manifests, GitHub Actions x2	Local-dev parity, cluster-ready deployment, CI on PR, build pipeline for backend & frontend

Tech stack at a glance

[Python 3.12](#) [FastAPI](#) [SQLAlchemy 2.0](#) [PostgreSQL 15](#) [Redis 7](#) [Alembic](#) [React 18](#) [TypeScript](#) [Vite](#) [Tailwind](#) [Zustand](#) [React Query](#) [OpenAI GPT-4o](#) [Anthropic Claude](#) [LangChain](#) [Stripe Connect](#) [SendGrid](#) [S3 / MinIO](#) [Docker](#) [Kubernetes](#) [GitHub Actions](#)

Key Capabilities Delivered

Conversational AI Intake

A guided chat experience replaces the 12-field intake form. The `conversational_intake_service` uses an LLM to ask only the questions that the eligibility engine still needs, branching by claim category and jurisdiction. Drop-off is reduced because users describe what happened rather than parsing legal jargon.

AI Document Generation

The `ai_document_generator` produces three document types — demand letters, regulatory complaints, and small-claims court forms — seeded by Jinja2 legal templates and refined by a Claude/GPT-4o pass. Generated output is then graded by the `quality_assurance_service` against a rubric, with a visible quality badge in the UI.

Multi-Stage Escalation Engine

Each claim moves through a deterministic escalation ladder — demand letter → follow-up → regulator → small claims — with timers, notification triggers (SendGrid), and operator overrides. The engine is

audit-logged, so every state transition is traceable.

Fraud Detection & QA

A pre-submission fraud check (`fraud_detection_service`) flags duplicate claims, identity inconsistencies, and known-bad patterns. Combined with a quality score, this protects the contingency-fee economics by preventing the platform from chasing claims that cannot win.

Stripe Connect Settlements

Funds *never* touch the company balance sheet. Stripe Connect handles escrow, commission split (15% / 12% / 10% by tier), and payout. Webhooks are handled in `app/api/webhooks.py` with idempotency keys and audit logging.

Cross-Sell & Referral Engine

Once a user wins a flight claim, the `cross_sell_service` automatically scans their account for adjacent claims (bank fees, telecom overcharges, subscription traps). The `referral_service` rewards invites with commission tier upgrades.

Multi-Language UI

The frontend ships with full localization for English, German, Spanish, French, and Portuguese — covering the EU261 footprint plus LATAM expansion — via a typed `i18n` module under `frontend/src/i18n`.

Admin Console

A separate admin surface (`pages/admin/`) gives operators a dashboard, claim management, manual escalation, and override controls — intentionally walled off from the consumer UX.

Delivery Timeline

Days 1–3

Discovery, SRS & Risk Analysis

Stakeholder interviews, jurisdictional research, UPL legal-posture definition, claims-assignment model decision, formal SRS PDF, formal Problem & Solution risk analysis.

Days 4–7

Architecture & Scaffolding

Domain modeling (12 entities), API surface design, FastAPI scaffold, React + Vite shell, Alembic baseline, PostgreSQL schema, Docker Compose parity for local dev.

Days 8–16

Core Build — Backend & Frontend

15 API routers, 19 domain services, claim lifecycle state machine, JWT auth, 13 frontend pages, claim wizard, conversational chat component, dashboard charts (recharts).

Days 17–22

Legal Engine & AI Layer

9 jurisdiction frameworks, 8 legal templates, AI document generator, conversational intake service, OCR, fraud detection, QA scoring.

Days 23–26

Payments, Notifications, Localization

Stripe Connect integration, webhook idempotency, SendGrid templates, escalation timers, 5-language i18n bundle, admin console.

Days 27–30

Hardening & Deployment Posture

10 Kubernetes manifests, 2 GitHub Actions workflows, seed script with 11 users / 15 claims / 6 companies, end-to-end smoke run, runnable on localhost in one command.

Outcomes & Impact

Engineering outcomes

177 Files Shipped	~41,500 Lines of Code	61+ REST Endpoints
12 Domain Models	19 Services	9 Legal Frameworks

Business outcomes

- **Time-to-market compression:** A scope that traditionally scales to 4–6 months with a 5–6 person team was delivered in roughly 30 working days, runnable end-to-end with seed data.
- **Investor-ready posture:** SRS, risk-analysis document, and a working demo — the three artifacts an early-stage founder needs for fundraising — produced as deliverables of the same engagement.
- **Industry-leading commission economics:** 15% / 12% / 10% contingency tiers, materially below incumbents (DoNotPay, AirHelp, Flightright at 25–35%), sustainable because of the AI-driven cost base.
- **Defensible moat:** The combination of jurisdiction-as-code, AI document generation, and Stripe-Connect escrow is non-trivial to replicate — a credible technical barrier for follow-on competitors.

Brain Station 23 didn't just write code — they collapsed what should have been a year of architectural decisions into a single sprint. We started Day 1 with a thesis and ended Day 30 with a system I can demo to investors, seeded users, Stripe in test mode, and a Kubernetes manifest ready to push.

— Founder, ClaimRight

Why Brain Station 23

AI-Native Delivery

We pair senior engineers with frontier LLMs (Claude, GPT-4o) inside a delivery playbook that compresses scoping, scaffolding, and code generation — without compromising production posture.

Domain-Heavy Engineering

Legal, financial, and regulated-industry projects demand engineers who can reason about UPL, escrow, idempotency, and audit. Our team treats compliance as a first-class architectural concern.

Full-Stack Range

Backend (Python / Node / Go), modern frontend (React / Next / Vite), data (PostgreSQL, Redis), DevOps (Docker, Kubernetes, GitHub Actions), and AI (LangChain, vector stores) under one roof.

Documentation as a Deliverable

Every engagement ships with consulting-grade documentation — SRS, architecture, risk analysis — that doubles as fundraising and handover material.

Founder-Friendly Engagement Models

We support solo founders and lean teams with right-sized engagements: AI-augmented MVP sprints, fractional CTO partnerships, and full product-build retainers.

Production from Day One

Even MVPs ship with migrations, auth, observability, CI, and K8s manifests. There is no "we'll harden it later" debt baked into the delivery.

About Brain Station 23

Brain Station 23 is a global software engineering firm with deep practice areas in AI-native product delivery, FinTech, LegalTech, and enterprise modernization. We partner with founders, scale-ups, and enterprises to design, build, and operate software that compounds — engineered for the regulated, the high-stakes, and the long-lived.

We specialize in compressing the gap between an idea and a defensible, production-ready product, using the same delivery framework that produced ClaimRight: opinionated architecture, AI-augmented engineering, and documentation as a first-class artifact.

Whether you are pre-seed and need an MVP that holds up in front of an investor, or scaling and need an AI layer bolted onto a regulated system, Brain Station 23 has the engineering bench and the delivery framework to ship it.

rupak.amin.bs23@stellainternational.com
brainstation-23.com