

CASE STUDY · 2025-2026

Allmap.io

A collaborative mapping platform where creators build private maps and earn money from them. Designed, built and shipped by one engineer.

Joshua Wong — solo founder-engineer · product, backend, frontend, mobile, infra

1.1M+

trails rendered worldwide

559

commits in one year, solo

3

platforms: web, Android, iOS

[Open allmap.io ↗](#)

[Contact](#)

Maps are where people plan their lives, but creators can't own one.

Travel creators, guides and local experts pour knowledge into Google lists and spreadsheets that they can't brand, gate or charge for. Allmap gives them a map of their own: private or public, collaborative, with paid access built in.

A full product, not a prototype.

`images/map-overview`

Map with layers, clustering and live events

`images/routes`

1.1M trails as vector tiles

`images/paint-grid`

Collaborative paint grid layer

`images/events`

Ticketmaster events, refreshed daily

`images/monetization`

Subscriptions & creator payouts

`images/mobile`

Native Android / iOS app

Django 5 + DRF

PostgreSQL + PostGIS

Next.js 16

MapLibre GL

Capacitor 8

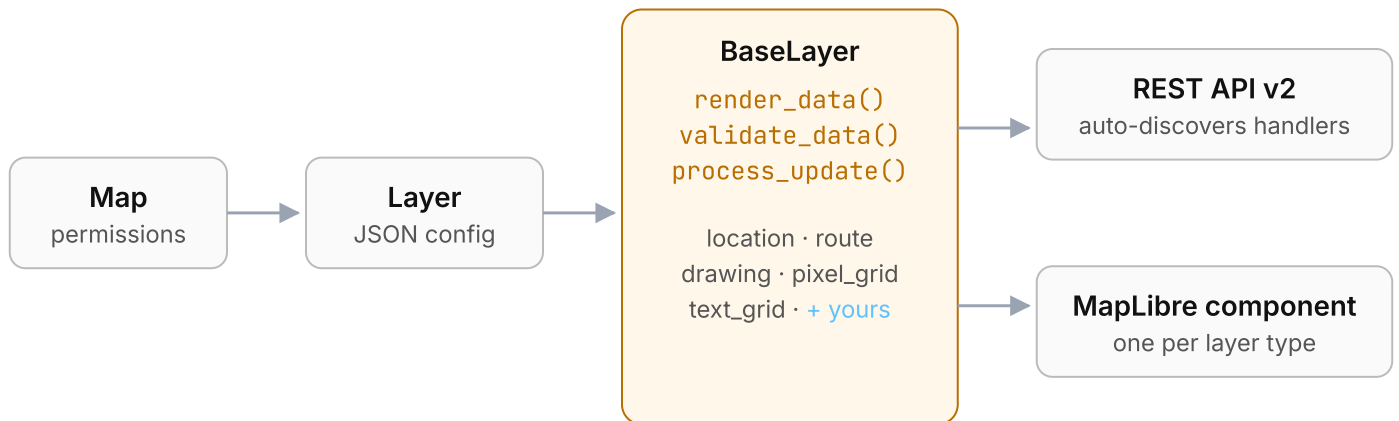
Stripe Connect

Cloudflare R2

Render

New map features plug in. Nothing else changes.

Every map is a stack of layers: pins, routes, drawings, paint grids, text. Each layer type is one self-contained handler that owns its own data format, validation and updates. The API finds handlers on its own, so adding a layer type means writing one backend class and one frontend component.



Why it matters: five layer types shipped without touching the core. The registry stays thin and each handler stays deep.

Designed for millions of places on a phone.

1.1M trails without melting the browser

Problem: streaming route GeoJSON from the API stalled at scale.

Decision: pre-render routes into PMTiles vector tiles with a parallel tippecanoe pipeline (6× faster), hosted on Cloudflare R2.

Result: a 2.1 GB tileset served as byte-range requests from a CDN, with no API server in the path. That's what you're seeing on this map.

150 queries → 11

Problem: the map list fired an N+1 query storm.

Decision: prefetching, dynamic field selection and slim marker types at ingestion.

Result: 93% fewer queries per load. Clustering moved to a web worker, and the cache uses LRU eviction.

Fresh events on a fixed API budget

Problem: Ticketmaster caps calls per day and per second.

Decision: 10 staggered cron jobs across 249 city cells in tiers, ordered so polling always reads fresh counts.

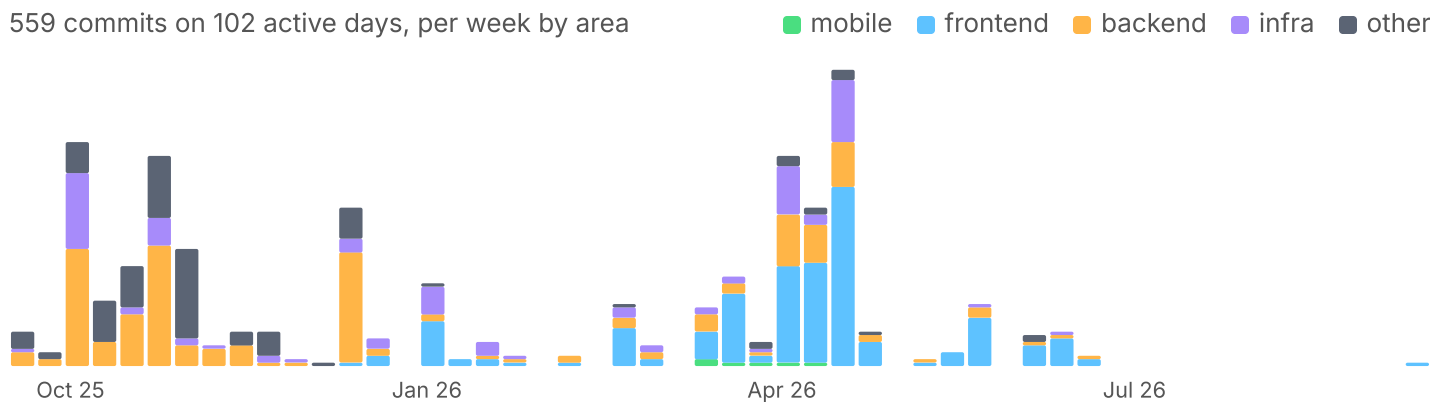
Result: about 4,500 calls a day, always under 5 req/s, with a 500/day on-demand safety valve.

One engineer, the output of a small team.

I build with AI coding agents as the primary implementers and act as architect and reviewer. Speed alone doesn't make that work. Discipline does, so I wrote it down:

- **A written engineering constitution** ([CLAUDE.md](#)): deep modules, information hiding, errors defined out of existence. Agents follow it on every change.
- **Guardrails over trust**: CI on every push (pytest, Playwright, Black, Flake8, Trivy, Bandit) before auto-deploy.
- **Plan → implement → review loops** for anything that touches more than 3 files or the schema.
- **Canonical patterns** named in the docs (for example the anti-drift marker structure), so agents copy the right thing.

559 commits on 102 active days, per week by area



Live, monetizable and still shipping.

- Live at allmap.io on web, Android and iOS.
- Creator economy end to end: subscriptions, one-time purchases, Stripe Connect payouts and contributor revenue sharing.
- Search relevance ranking, onboarding and guided tours, PostHog analytics.

What I learned

AI agents make writing code cheap. They make *design* the bottleneck. The most valuable thing I built isn't a feature. It's the set of rules that keeps the codebase simple while it grows fast.

CONTACT

Let's talk.

I'm looking for roles where product sense, full-stack depth and AI-first velocity matter.

Email

Resume

LinkedIn