

...
>_ ./init_panpanmao.sh

Building a fortune-telling app with zero domain knowledge

Day 1 Build Diary: 21 commits, monorepo
architecture, and AI-driven development.

[2026-01-21]

The desire for full-stack product ownership

Big Tech Engineer



- **Scope:** Deep but narrow. Building things others spec out.
- **Focus:** Shipping infrastructure, scaling systems, architectural decisions for millions.
- **Missing Elements:** Never touches GTM strategy, pricing page copy, or user feedback loops (e.g., why is conversion 2% instead of 4%).

Solo Builder



- **Scope:** Broad and holistic. Full lifecycle from Figma to GTM.
- **Focus:** Absolute ownership, direct-to-user feedback, navigating high ambiguity.
- **Present Elements:** Total control. No PMs, no designers, no growth teams.

Testing the limits of AI-driven development

When choosing a side project, there was one strict constraint: it had to be in a domain the author knew absolutely nothing about.

/*

Can AI completely bridge a domain knowledge gap?

Not just 'can AI help an expert work faster' — **can someone with zero domain expertise build a credible product using AI as their entire knowledge base?**

*/

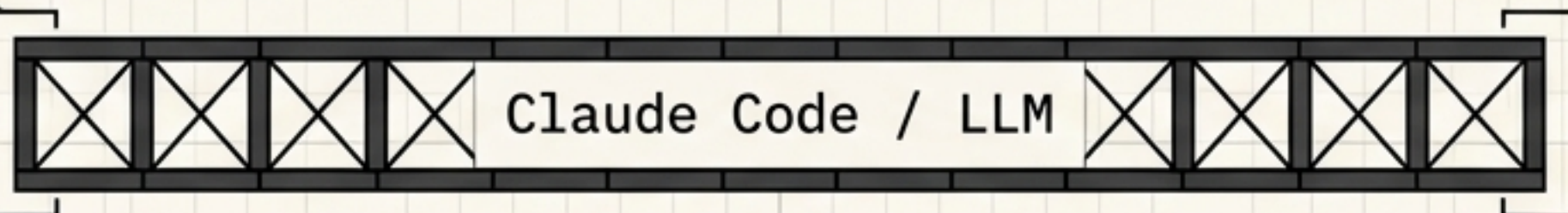
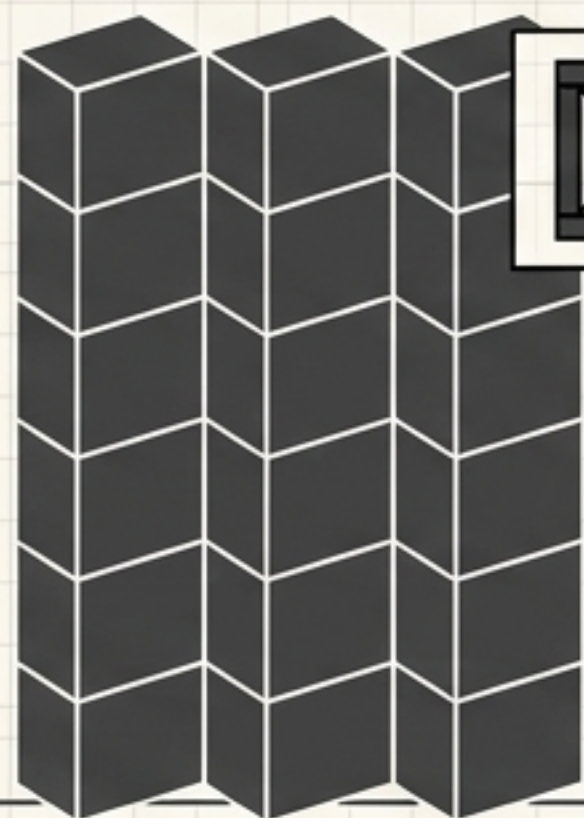
Deliberately choosing an opaque domain

The CT0-type Engineer

- Solid Ground: React, Supabase, Next.js
- Deep technical expertise

The Target Domain

- Chinese Metaphysics: BaZi, astrology, tarot, Liuren
- Completely unknown (cannot define a day pillar or why birth hours matter)



Takeaway: The lack of domain knowledge isn't a bug; it's the required condition for the experiment.

An enormous, digitally underserved market

[The Current State]

Existing apps in the Chinese-speaking fate analysis market are predominantly ad farms.

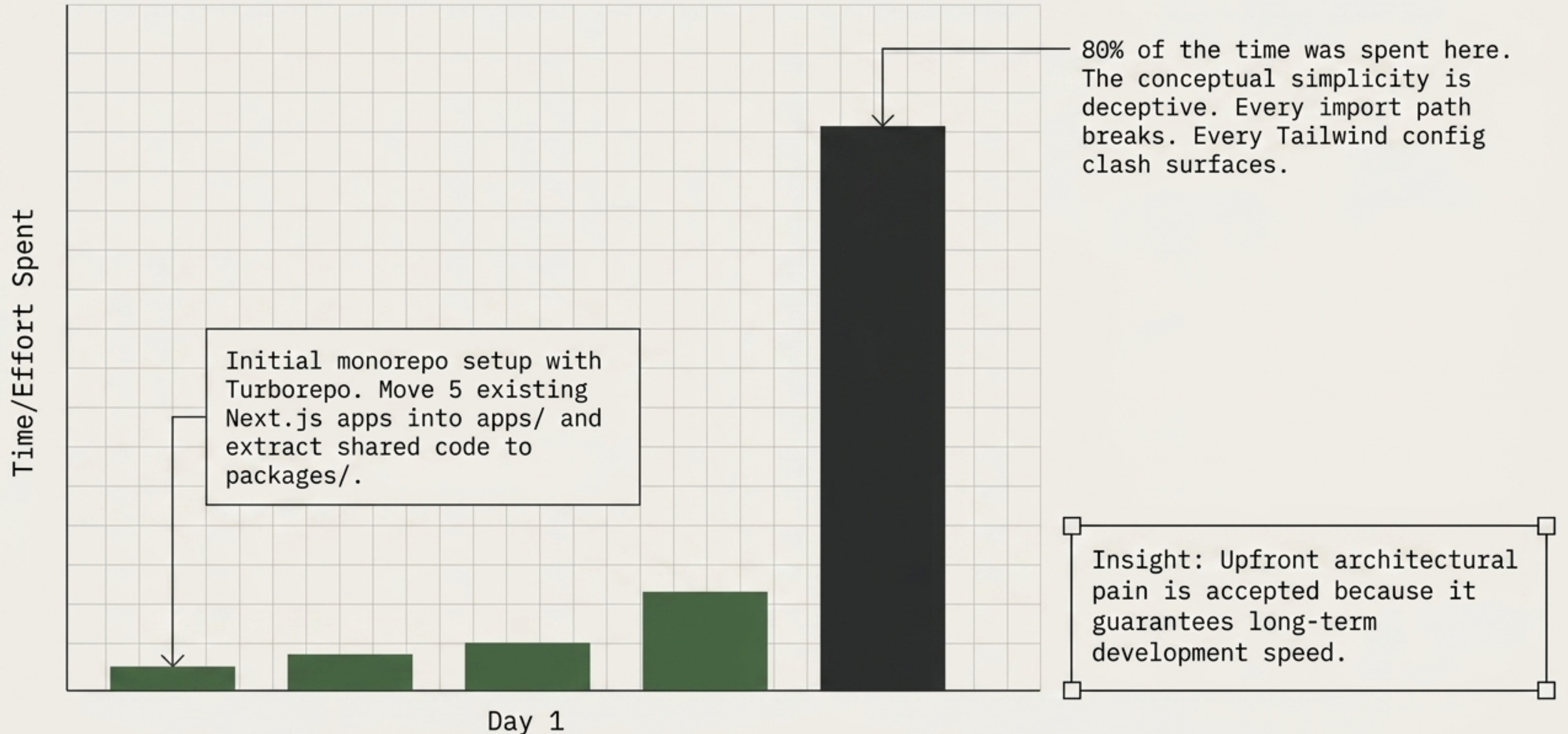
Output is limited to shallow, single-paragraph generators.

[The Pragmatic Opportunity]

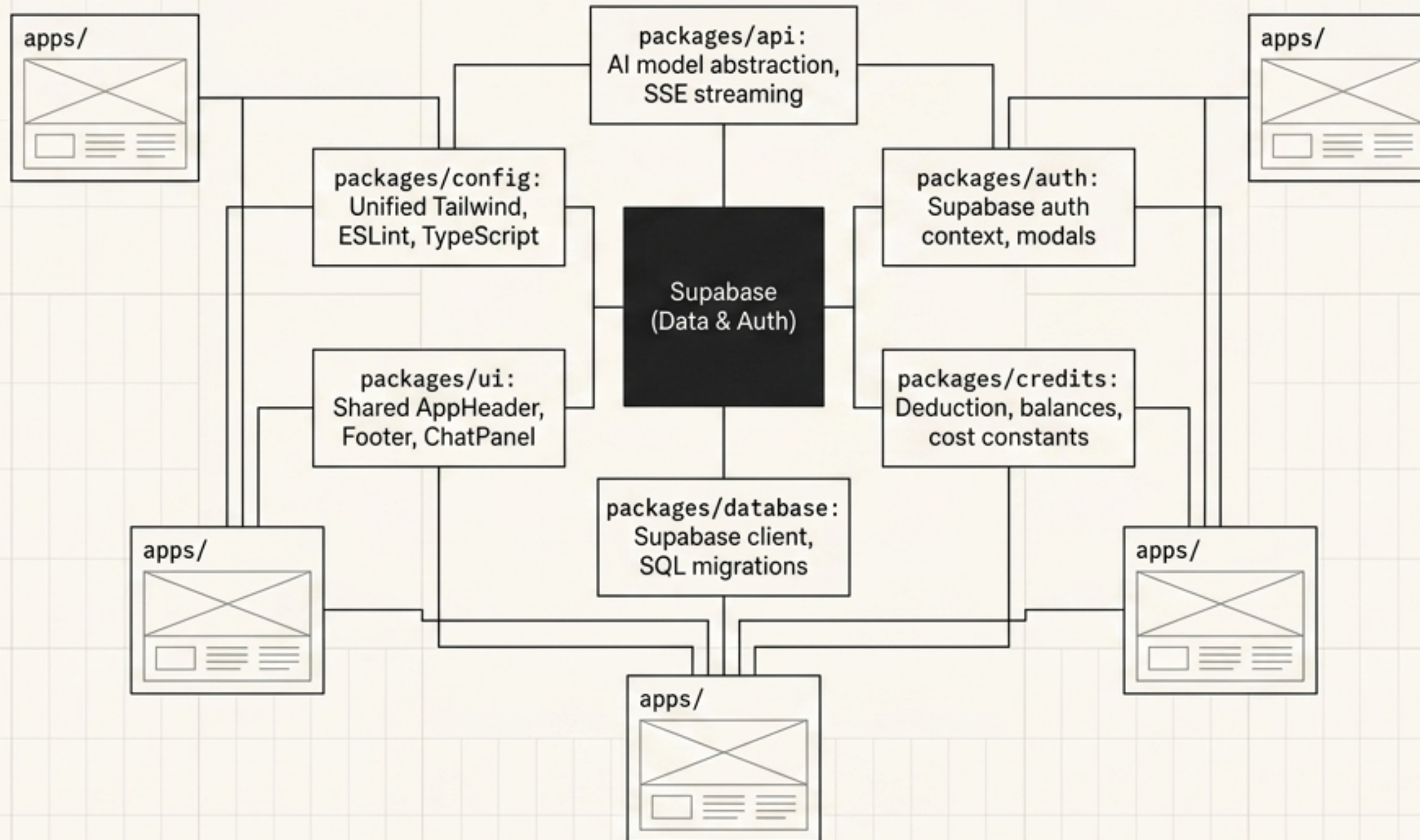
A massive audience actively seeking fate analysis.

An opportunity to use AI to build a highly credible, multi-faceted platform that replaces shallow generators with deep, interactive analysis.

Establishing the monorepo laboratory

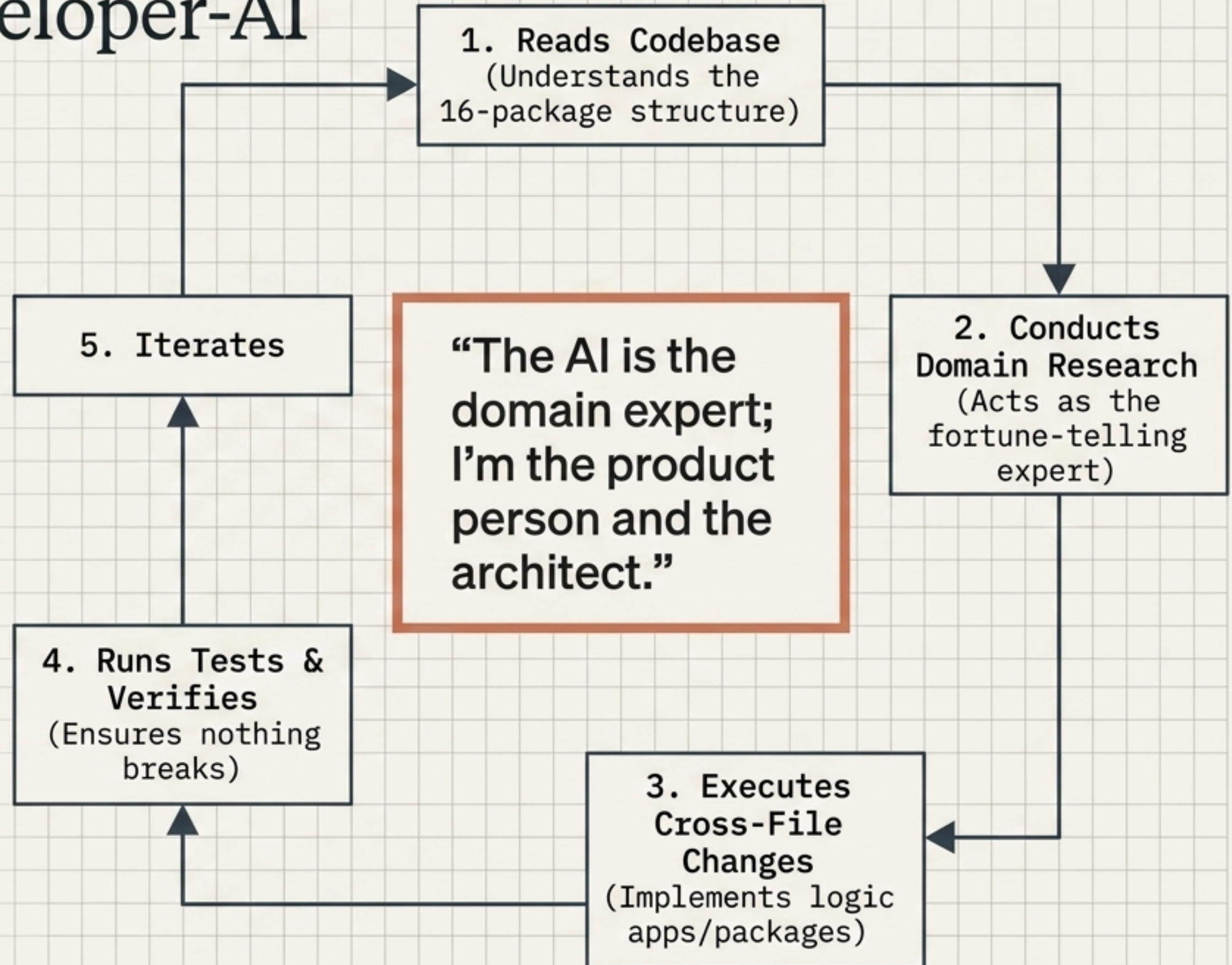


Extracting shared infrastructure to scale across five apps

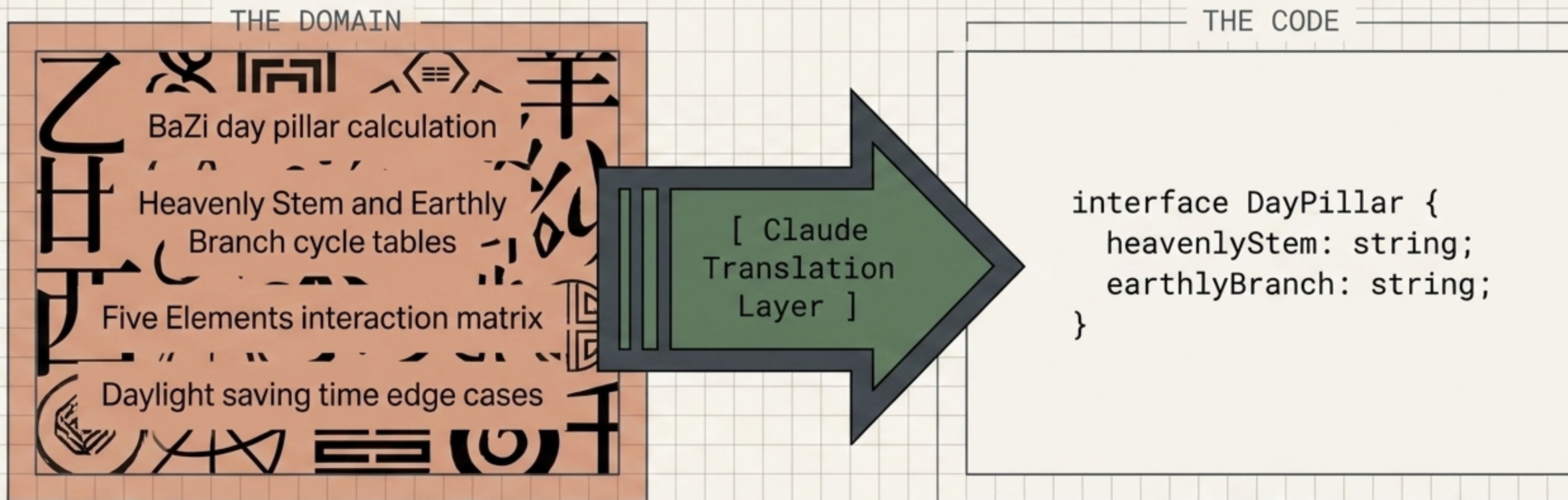


Redefining the developer-AI relationship

Claude Code is used as a primary development tool — not a chatbot to paste code into, but an autonomous coding agent.



Translating opaque metaphysics into strict TypeScript



The AI researches the domain logic and instantly outputs working TypeScript matrices.

Reversing the traditional flow of domain logic

	Traditional Development	The PanPanMao Model
Role of Human	Coder + Domain Learner	Product Architect
Role of AI	Syntax Helper (Copilot)	Sole Domain Expert + Implementer
Source of Domain Logic	Human Research & Specs	Native to the LLM's training data
Speed to MVP	Slow (bottlenecked by human learning)	Hyper-accelerated

End of Day 1: 21 commits to a unified ecosystem

[OK] Build Status

The Turborepo builds successfully. All 5 apps load.

[OK] Authentication

Shared auth works across the ecosystem. One single login.

[OK] Database

The Supabase client is perfectly unified.

[OK] Economics

One unified credit balance spanning all five products.

**21 commits. One monorepo.
Zero personal knowledge of fortune-telling.**

The foundational stack powering the platform

{ Architecture }

Monorepo

Powered by Turborepo.

{ Framework }

Next.js

Utilizing the App Router.

{ Backend/Auth }

Supabase

Postgres + Row Level Security (RLS) + real-time subscriptions.

{ AI Engine }

Claude + Gemini

Multi-model setup toggled via PostHog feature flags per endpoint.

{ Language }

TypeScript

Strict mode enforced, shared types across 16 packages.

{ Styling & Deploy }

Tailwind CSS / Vercel

Per-app design token themes deployed seamlessly.

The architecture of ignorance



Insight: Extreme technical rigor at the foundation is precisely what enables extreme flexibility at the domain level.

By over-engineering the shared infrastructure into a robust monorepo, the author created a blank, indestructible canvas.

The AI isn't just an assistant; it is the content engine instantly 'painting' unknown domain expertise onto a meticulously built human chassis. Upfront architectural pain makes AI-driven product discovery possible.
