

Daniel Ramirez

Forward-Deployed AI Engineer · Product · Design · CPO-track
Sacramento, CA · open to remote · [/harness](#)

daniel@strategylabs.io
(530) 391-8330
strategylabs.io
[linkedin/daniel-ramirez](https://linkedin.com/in/daniel-ramirez)
github.com/pterodaniel



THE DEPLOYMENT

Forward-deployed is what I've been doing, before the title existed: embedded with one other engineer for **14 months** and shipped **ShimmerWorks** — a production multi-tenant SaaS (60+ pages, 130+ API routes, 130 migrations), later **acquired** — it runs today as BriteBuild. I run the technical practice at **Strategy Labs AI**: agent orchestration, on-prem inference, AI-accelerated delivery. The product judgment comes from nine years as COO of the exact business my software served — I was the customer before I was the engineer.

HOW I BUILD WITH AI

CODING AGENTS	Claude Code (terminal, desktop, IDE plugins, VS Code) · Cursor · Codex . Daily driver across all three; custom agent harnesses for the workflows the off-the-shelf tools don't cover yet.
AGENT ORCHESTRATION	Multi-agent swarms with typed execution contexts: tool-use, file-search, retrieval, code-gen, vision, and computer-use sub-agents that hand off work and converge. Builds custom agent harnesses and 17 authored Claude Code skills.
AGENT-NATIVE APPS	Built an in-app remote agent harness into Breakfast Ball — device registry, command queue, double opt-in — so AI agents can drive the live iOS app to diagnose UX issues. Computer-use for my own product.
INFERENCE	RAG pipelines, structured-output extraction, vision, voice (Whisper + TTS with deterministic tool-action confirmations). On-prem inference on RTX 4090 stacks and Apple Silicon; published the 2025–2026 hardware guide.

EXPERIENCE

Founder & Owner-Operator · Strategy Labs AI, LLC

2024 — Present

AI-native technology consulting · strategylabs.io

- Run the technical practice: architecture, agent systems, on-prem inference, and the builds themselves — deployed into client environments, shipping against their constraints.
- Builds custom agent harnesses and a library of 17 authored Claude Code skills; built the firm's interactive on-prem hardware-sizing tool.

Co-Founder & Engineer · ShimmerWorks

2025 · **acquired**

Multi-tenant CRM & SaaS for solar EPCs · co-built with one other engineer over 14 months · acquired, now operated as BriteBuild

- Designed the **tenant → EPC → contractor data model**: polymorphic state/county territory scoping, **325-permission RBAC across 24 domains and 7 roles**, audit triggers throughout.
- Owned the **bitemporal pricing engine**: quote-version-locked prices via session-pricing snapshot functions, so a quote sent Tuesday still reads Tuesday's prices on Friday without duplicating the catalog.
- Authored the **bitemporal pricing system** (quote-version-locked prices via snapshot functions) and **multi-bucket financing splits** routing line-items to different lenders.
- Shipped the rebates engine (3-tier inheritance), DocuSign contract templates with dynamic merge fields, Slack-like realtime messaging (8 desync bugs fixed, **82 E2E tests**), public REST API v1, and analytics.
- Hardened it like production demands: centralized OAuth hub with signed JWT hand-off to tenant subdomains, real auth CVEs patched, axios removed for supply-chain CVE, `svelte-check` warnings driven **1530 → 0**.
- Scale: **557 of my commits, 174k insertions, 130 migrations, 630 Svelte components, 1,131 TS modules**.

TypeScript

Svelte 5

SvelteKit

Node.js

PostgreSQL

Python / FastAPI

AWS S3 / SES

Stripe Connect

DocuSign

Creator · Breakfast Ball

2025 – Present · **TestFlight beta**

Native iOS + watchOS golf scorecard & side-games app · solo build

- Four Apple targets: SvelteKit core via Capacitor, **SwiftUI watch app** (26+ views), WidgetKit glances, and **ActivityKit Live Activities** with Dynamic Island and 10-second background GPS→yardage lock-screen updates. **11.7k LOC Swift**, plus Kotlin foreground-service equivalents on Android.
- iPhone ↔ Watch sync via a **versioned Codable WatchConnectivity contract**; optimistic updates and offline persistence across web, iOS, watchOS, and Android.
- Wager **settlement engine**: nine simultaneous formats resolved through a pairwise settlement graph with wash detection — pure deterministic TypeScript, **188 tests**, zero DOM dependency.
- Supabase backend: 52 migrations, **100+ RLS policies**, trigger-enforced invite integrity, 5 Deno edge functions running APNS JWT push with quotas.
- Wrote 4 custom Capacitor plugins and 4 custom Vite plugins — including the voice caddie (Whisper + TTS, 10+ tool actions) and the remote agent-drive harness.

Swift / SwiftUI

ActivityKit

WidgetKit

WatchConnectivity

CoreLocation

Capacitor

Kotlin

Supabase

Deno

APNS

OpenAI voice

Co-Founder & COO · Gold Rush Energy Solutions

2015 – 2024 · **exited**

\$30M+/yr solar EPC · where the product judgment comes from

- Ran the company that used the software: nine years of quotes, financing, permits, crews, and contracts — the domain model I later encoded into ShimmerWorks.
- Taught myself to ship here: Salesforce (Apex/LWC/SOQL) tools, Google Apps Script document automation gluing Salesforce ↔ Workspace ↔ DocuSign ↔ QuickBooks.

— **PRODUCT DECISIONS I'LL DEFEND** —

The database is canonical; the UI is a projection.

The financing "rack & cable" diagram shipped with localStorage-only persistence — caught it mid-build and inverted the design so scenarios live in Postgres and the diagram renders from them. Demos forgive that shortcut; production doesn't.

Filter at the data boundary, not the view.

Contractor-scoped users must never see EPC pricing or internal identities — so the filter lives in the API layer, where a forgotten UI hide can't leak it.

Two clean schemas beat one clever one.

Rebates looked like adders, but recurring payouts, eligibility scoping, and certainty tiers made them a different shape. Built a parallel system with 3-tier inheritance instead of forcing one schema to lie.

Meet users in their idiom.

One CRM, four parallel views — Kanban, magazine, table, terminal — because a sales manager, a closer, and an ops coordinator don't read pipeline the same way.

ENGINEERING STACK	
LANGUAGES	TypeScript, JavaScript, Python, Swift, Kotlin, SQL (PostgreSQL), Apex, SOQL, Bash
FRAMEWORKS	SvelteKit, Svelte 5 (runes), React, Next.js, Capacitor, Tauri, Electron, FastAPI, Node.js, Vite (custom plugin author)
NATIVE / MOBILE	SwiftUI, ActivityKit, WidgetKit, WatchConnectivity, CoreLocation, MapKit, Kotlin services, TestFlight / ASC automation, Xcode Cloud
BACKEND & DATA	PostgreSQL (bitemporal versioning, audit triggers, polymorphic FKs, RLS), Supabase, Deno edge functions, Redis, WebSockets (MessagePack), APNS
AI / AGENTS	Claude Code · Cursor · Codex, agent harness design, multi-agent orchestration, RAG, structured extraction, voice pipelines, on-prem inference (CUDA + Apple Silicon), OCR (Tesseract), LLM PDF parsing
QUALITY	Vitest, Playwright, E2E suites, Sentry, svelte-check 1530 → 0, CVE remediation

ALSO	
VP & Secretary, Placerville Food Co-op board — co-led a \$1.39M community building acquisition	2024 – present
California State Contractor (CSLB Lic# 99454SP) — I can also size your transformer	—