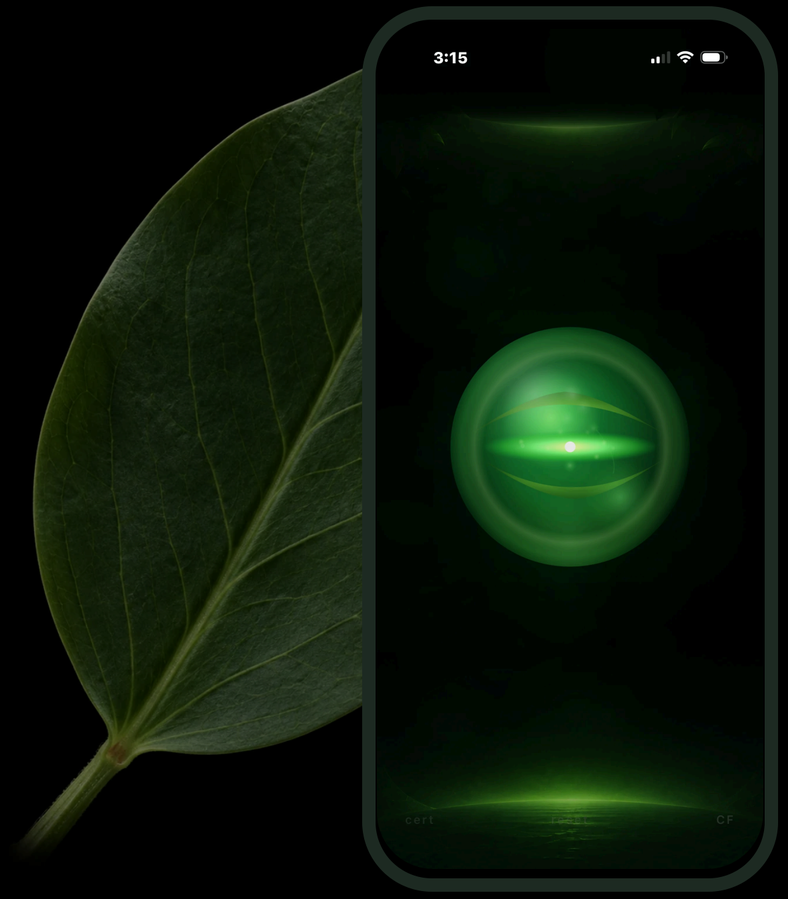
PORTFOLIO · NOVEMBER 2026

Jason Hyde

I build LLM products with AI coding agents,
and I evaluate whether they work.



Taken from the STOMEL app

STOMEL Founder, TerraSTOMEL · AI product builder · Model evaluator

SUMMARY

In five months I took an LLM product from first commit to TestFlight by directing AI coding agents, and measured it in use

1	**Shipped**	Build 38 on TestFlight: React Native client, Node server, two production models. 900+ commits and 199 regression suites.
2	**Measured**	p50 reply 1.7 s, p95 3.8 s. About $0.0013 in model cost per exchange, tracked per call in Founder Lab, my internal ops console.
3	**Evaluated**	Model and prompt changes are decided on small evals. The vision model swap went from 0 of 3 to 3 of 3 faithful answers.
4	**Still open**	Cross-session recall is not reliable yet. There is no CI pipeline or prompt-injection suite.

Source: STOMEL repositories, server logs and Founder Lab telemetry, April to November 2026

BACKGROUND

Twenty-five years in applied biology came first, then model evaluation, then building

Period	Role	Organization
2026 to now	**Founder and product builder**	TerraSTOMEL (STOMEL)
2026 to now	AI trainer and domain expert	Handshake AI
2024 to 2026	Biology expert and model evaluator	Outlier AI
2025	R&D lead, product development	Carolina Natural Solutions
2015 to 2025	Founder and R&D director	Natural Nomad Farms
2011 to 2015	Research and site manager, alternative proteins	Organic Nutrition Industries
1998 to 2011	Aquaculture research, hatchery and life support systems	Full Circle Aquatics, Dyer Aqua, Walt Disney World

Source: résumé, November 2026

THE PRODUCT

STOMEL pairs an on-device plant-state model with an LLM that reasons over the grower's own records

Plant state

Local weather feeds a rules-based state machine with 11 plant states, computed on the phone. The UI renders the state as the stoma's breath, glow and halo.

Conversation

DeepSeek V4 Flash streams replies grounded in the grower's journal, a ledger weighted toward past failures, and the grower's standing requests.

Design goal

No settings screens. The grower talks and the runtime adapts. Success means the grower needs the app less as their own observation improves.



THE FIRST INSTRUMENT FROM TERRASTOMEL

TerraSTOMEL is the company I founded to carry the same approach to aquaculture, livestock and other living systems. It is quiet for now; STOMEL comes first.

terrastomel.com

THE BUILD

A conventional stack, instrumented for latency, cost and safe rollback

Layer	Implementation
Client	React Native 0.81, Expo SDK 54, Expo Router; SQLite journal on device
Server	Node and Express on Render, deployed from main
Data	Supabase with Sign in with Apple and row-level security
Models	DeepSeek V4 Flash for chat and background jobs; Gemini 3.1 Flash-Lite via OpenRouter for vision
Ops	Founder Lab console for runtime model switching and per-call cost telemetry; 15 kill switches; per-route rate limits

Build 38
on TestFlight

900+
commits since April 26, 2026

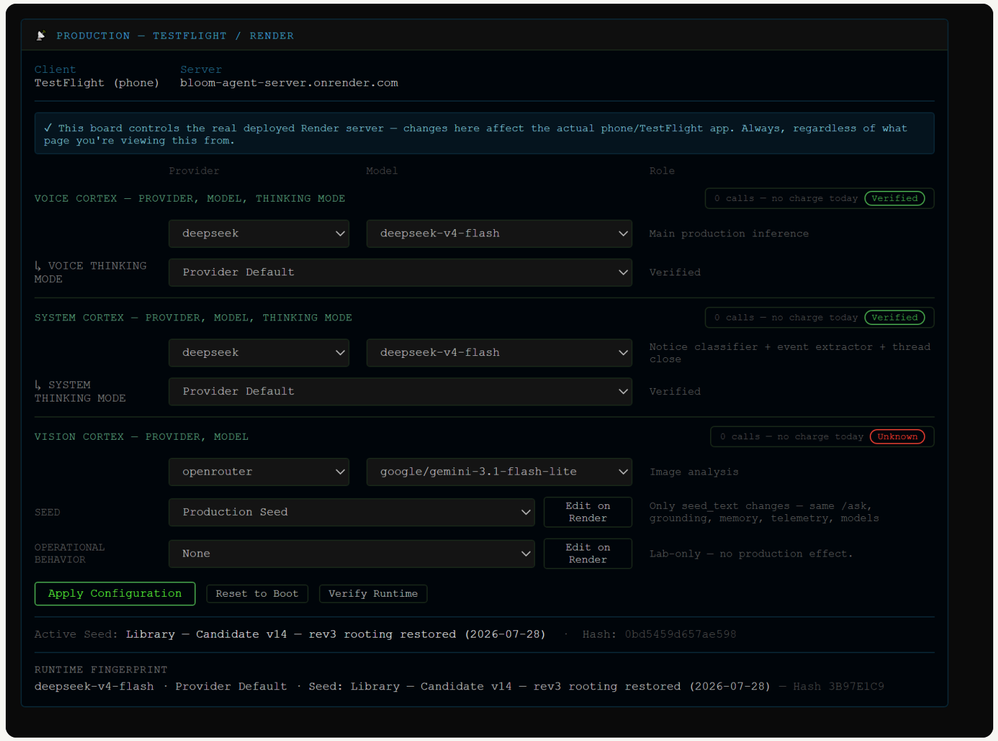
199
regression test suites

Source: STOMEL client and server repositories, November 2026

FOUNDER LAB

Founder Lab, the internal console I built, controls production models at runtime and prices every call

Production switchboard: model and thinking mode per role



Telemetry: one real exchange, five calls, cost per call

RECENT ACTIVITY last 25 calls · always current

TIME	ENDPOINT	MODEL	PROMPT	CMP	CACHE%	LAT	BILLED
05:49:43 PM	/api/steward/agreement	deepseek-v4	1,815	5	91.7%	1.1s	$0.000031
05:49:43 PM	/api/event/extract	deepseek-v4	7,212	14	97.6%	865ms	$0.000055
05:49:43 PM	/api/notice/classify	deepseek-v4	371	8	34.5%	618ms	$0.000042
05:49:42 PM	/ask	deepseek-v4	6,537	3	60.7%	1.7s	$0.00040
05:49:35 PM	/api/thread/close	deepseek-v4	12,416	265	89.7%	2.1s	$0.00038

Prompt versions	50+ system prompts, hashed, promoted to production
Eval harness	67 designed experiments, side-by-side compare
Grounding registry	24 prompt blocks with owner, version and hash

Source: Founder Lab and telemetry console on the production server, captured November 6, 2026

HOW I BUILD

AI coding agents write the code; I own the specification, the tests and the release decision

Change process, applied to every fix and feature

1	2	3	4	5	6
Reproduce the defect in the app	Write and approve a brief	Implement on a branch or lab server	Pass the regression suites	Compare against the baseline	Verify the whole app, then release

Control	What it prevents
Protected paths	Agents changing core files without an approved brief
Status ladder	“Built” being reported as “tested”, “deployed” or “accepted”
Session handoffs	Context lost between agent sessions

Agents used: Claude, GPT, Grok, GLM, Kimi. Source: project rules (STOMEL.md, PROTECTED_PATHS.md)

REQUEST PIPELINE

Each message runs a six-stage pipeline, from on-device state to post-reply extraction

0 · Plant state

On device. Weather drives the 11-state machine before the grower types anything.

1 · Input

Text, a photo or both. Photos go to /api/vision/analyze and Gemini Flash-Lite.

2 · Context assembly

weave() writes the turn to SQLite first, then gathers thread, place and season.

3 · Prompt build

POST /ask adds retrieved records, the ledger and standing requests. ~19.4k chars.

4 · Streaming

DeepSeek V4 Flash, thinking off. 15 s timeouts on stream open and between chunks.

5 · Post-reply jobs

Five async calls classify the turn, extract agreements and events, and close the thread.

Source: client and server code on main, November 2026. Full detail in “STOMEL, Start to Finish”

MODEL SELECTION

A three-photo eval moved vision to a model that was faithful, nearly 4× faster and about 14× cheaper

3/3

faithful answers, from 0/3 on the previous model

1.8 s

per eval reply, from about 6.8 s

~14×

lower cost per photo

5.1 s

time to first token on live photos, from 12.3 s

Method. The same photos went to each candidate model, and each answer was scored against what the photo actually shows. Live time to first token was then compared for the week before and after the switch.

Source: Founder Lab eval, July 2026 (n=3); production logs, July 27 and August 3, 2026

LATENCY

Median reply time is 1.7 s, and stalls over 15 s fell from 33 in July to one a month

September, /ask	Value
p50 total reply time	**1.7 s**
p90	**3.1 s**
p95	**3.8 s**
Input tokens served from cache	**64%**
Stalls over 15 s, per month	**33 → 1**

Direct API for chat

Through OpenRouter, DeepSeek ran 3.7× slower than the direct API, with time to first token ranging from 1.7 to 14.1 s on identical calls. Chat now calls DeepSeek directly; remaining OpenRouter calls pin a provider.

Stable prompt prefix

The fixed system prompt comes first, so the provider caches it across calls.

Timeout contract

No first token in 15 s: the app says it can’t reach the server. Once streaming starts, the reply always completes.

Source: server logs, mostly the founder’s own use, July to September 2026

INCIDENTS

Two July incidents became engineering rules that still apply

	Memory writes failing	Prompt config lost
Symptom	Thread-close calls returned HTTP 500, and those conversations saved no memory	A version of the core system prompt could not be recovered
Root cause	A caching refactor moved the JSON instruction out of last position. The model prefixed prose, and the parser required a leading “{”	Prompt versions lived as untracked JSON on one machine
Fix	Provider JSON mode, a balanced-object parser, and the instruction re-anchored last. Caching kept: close cost fell from ~$0.0012 to ~$0.00025	Prompt versions committed to the repository; a protected-paths list added
Rule	**Moving an instruction is a behavior change and gets tested like one**	**Anything the runtime depends on is in version control**

Source: project handoff records, July 29, 2026; server commits 36dbe19 and ac9bbfa

OPEN PROBLEM

Retrieval finds the right records, but the model still denies having them

Observed in production

With 11 prior-session messages in its context window, the model replied: “I don’t carry the content of past threads.”

Likely causes

1. Retrieved records sit mid-prompt under a “use invisibly, never recite” header
2. Competing instructions, including the grower’s own request to “say less”
3. The model’s earlier denials, replayed in conversation history
4. Large days truncated without notice, but still labeled as found

Lab result

0/3 → 3/3

faithful answers after moving the records next to the user’s question

Not shipped. Next: a larger eval set, then release behind a flag.

Source: production logs, August 23, 2026; Founder Lab placement test (n=3)

CURRENT GAPS

Four gaps remain, and each has a defined next step

Gap	Current state	Next step
Cross-session recall	Placement fix passes 3 of 3 in the lab	Larger eval set, then release behind a flag
Continuous integration	199 suites run by hand before each release	Run the suites on every push
Adversarial testing	Honesty and harm probes pass; no prompt-injection set	Injection suite for retrieved and web content
My code fluency	I verify behavior and tests, not every line	Reading the codebase module by module

Source: project state and investigation records, November 2026

WHAT I WOULD BRING

A builder who can ship with AI coding agents and measure the result

Agent-driven delivery

Specs, guardrails and release gates that keep agents productive on one codebase

Evaluation

Targeted evals for faithfulness, latency and cost, run before decisions

Domain depth

25+ years in biology and cultivation to catch confident wrong answers

Production telemetry

Founder Lab: runtime model control, per-call cost and latency, kill switches

Jason Hyde Jasonhyde68@yahoo.com stomel.org terrastomel.com