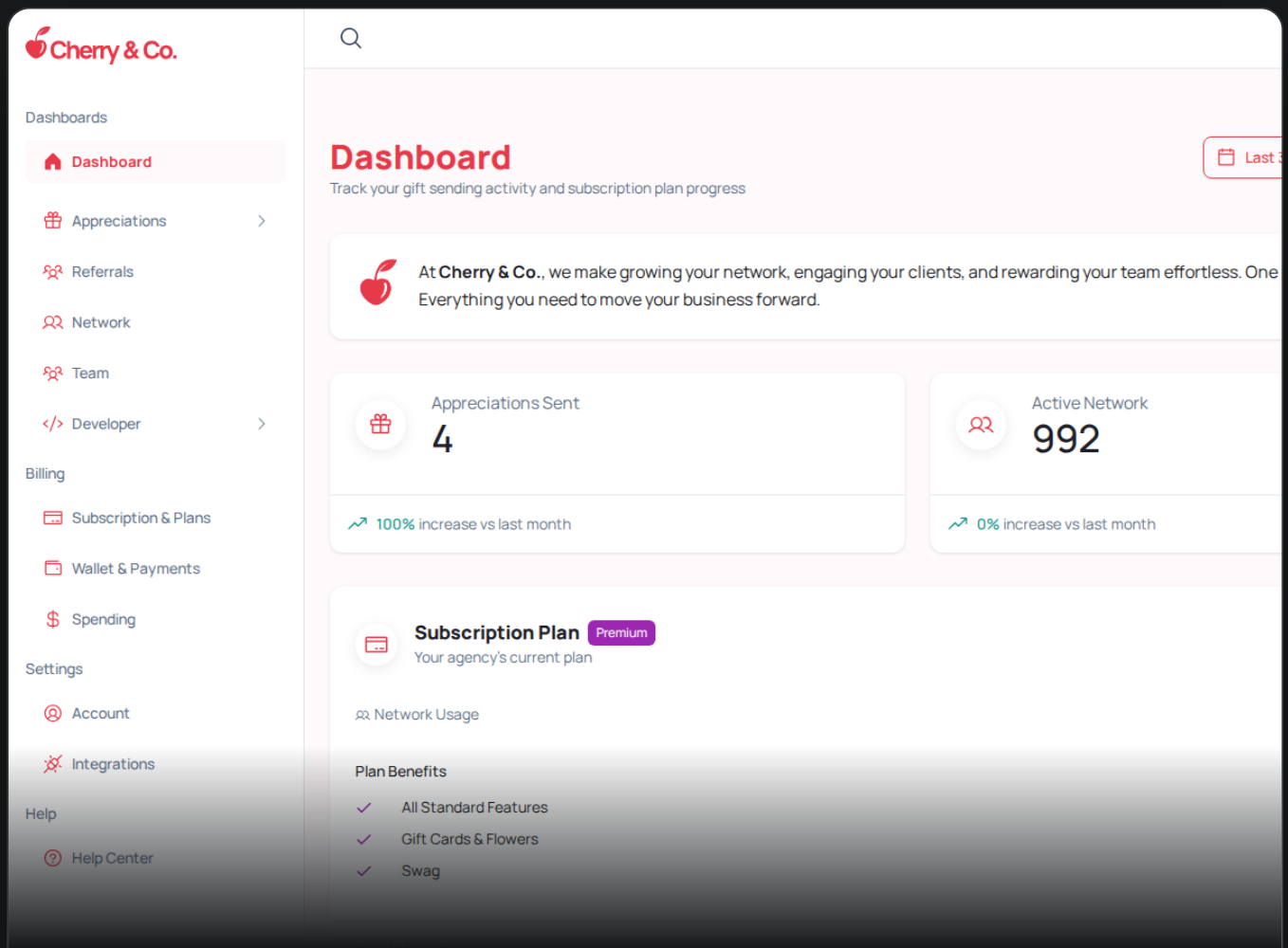


CLIENT APPRECIATION, SIMPLIFIED

# A founder's idea, shipped as a real business.

How Project Assistant took Cherry & Co. — a gift-sending platform for insurance agencies — from a founder-built prototype to a production platform with real payments, real deliveries, and real customer agencies.



AT A GLANCE

01

# One screen instead of six vendor websites.

6

GIFT TYPES

7

LIVE INTEGRATIONS

3<sub>+1</sub>

AGENCY ROLES + ADMIN

6<sub>mo</sub>

IDEA → PRODUCTION

## CLIENT

### Cherry & Co.

Founded by an insurance agency owner — a non-technical founder with a working idea.

## PRODUCT

### Multi-tenant SaaS platform

Owner / Producer / CSR roles with per-role spending limits, plus platform administration.

## ENGAGEMENT

### Full build & launch

Discovery → design → build → production launch → ongoing operation & support.

## STACK

### React · TypeScript · FastAPI · Supabase

MUI component system; deployed on Vercel and DigitalOcean.

#### Postcards

printed & mailed

#### Letters

printed & mailed

#### Gift cards

digital

#### Flowers

delivered fresh

#### Trees

planted in memory

#### Merch

agency-branded

## THE PROBLEM

# Gifting worked. The logistics didn't.

### Too many websites

A postcard, a gift card, and flowers meant three vendor sites, three logins, three checkouts.

### No approval path

Teams couldn't easily send when the owner was out — and owners couldn't easily say yes.

### Untracked spending

Gift spending scattered across accounts — hard to see, hard to budget.

Print & mail site

Gift-card site

Florist site

Merchandise site

Tree-planting site

...and spreadsheets



## One send-gift flow

Six gift types · one checkout · one wallet · per-role spending limits · one dashboard

# The founder had already built a prototype himself.

Before Project Assistant was engaged, the founder — with no engineering background — had built a working prototype on an AI app-builder, including a live florist integration. He arrived with a written scope, a vendor shortlist, and three competing agency proposals. **PA won the work in a competitive bid.**

The hardest question was the first one: *should the prototype become the product?* Everything that followed — the foundation, the payment model, the launch — traces back to how that question was answered.

And it wasn't answered by instinct. It was answered in writing, question by question, before the first line of code.

CLIENT PROTOTYPE SCREENSHOT  
[ AWAITING CLIENT CLEARANCE ]

the founder's self-built prototype,  
december 2025

THE STARTING POINT — A WORKING PROTOTYPE, BUILT BY  
THE FOUNDER HIMSELF

# 38

**Thirty-eight numbered questions across seven domains** — written down and triaged before any code, thirteen flagged must-answer-before-starting: spending limits, account ownership, billing, data privacy, scope. The questions that block a build were found on day one, not mid-sprint.

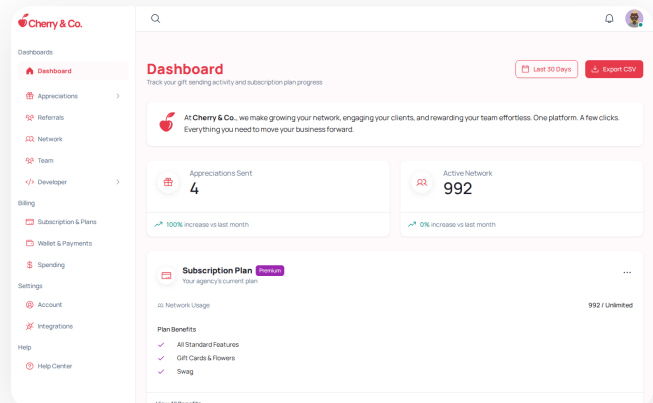
# Restart on a production foundation — without discarding the founder's work.

The easy path was to convert the prototype and absorb the debt. PA made the harder call, in writing, with a side-by-side comparison: rebuild on a licensed production React/TypeScript foundation — and carry forward, by name, the four things the prototype got right.

## BEFORE · THE PROTOTYPE

PROTOTYPE DASHBOARD  
[ AWAITING CLIENT CLEARANCE ]

## AFTER · SHIPPED



THE SHIPPED SPENDING DASHBOARD — IN PRODUCTION

### Florist integration

Working logic kept and productionised.

### Gift-sending flow

A UX already validated with his stakeholders.

### The tier concept

The subscription structure still in use.

### The business rules

Domain knowledge no agency could invent.

# Built around the client's judgment, not just ours.

## The prepaid wallet — the client's call, against our recommendation

02

PA put two payment models to the client in writing and recommended the lighter build: charge the card per gift. The client chose a prepaid wallet — he was reasoning about his own bookkeeping, where dozens of small card transactions a month is an accounting problem. **He had context we didn't. We built what he chose.**

The wallet became load-bearing infrastructure: it is how flowers, merchandise, and gift-card spending are funded — and what makes per-role spending limits enforceable, the founder's original first pain point.

## One merchandise account, not one per agency

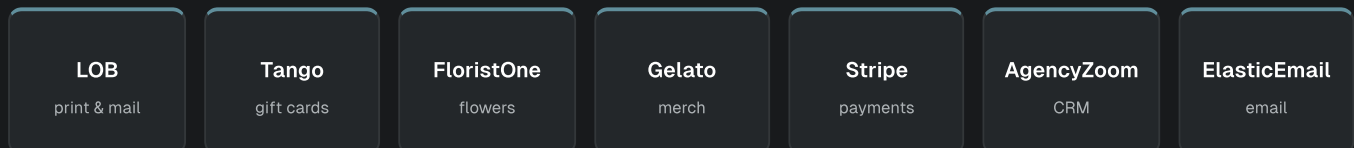
03

The client's team proposed each agency connecting its own fulfilment account. PA scoped it and put the consequences back as an explicit ruling: the vendor exposes ~240,000 products — unusable as an open catalogue — and per-agency accounts would have broken the wallet model. Within a week the platform-owned model was adopted: curated per-agency catalogue, the agency's own logo applied.

One detail worth featuring: agency branding is applied through **deliberate safeguards** — a design decision about trust, not a feature.

# A real platform, run like real infrastructure.

One platform serving many agencies — each with its own team, roles, wallet, and branding. Seven third-party services live in production behind one send-gift flow.



## Engineering that respects the money path

- A prepaid wallet funding six gift types, with full transaction history and per-role limits.
- Card handling designed so the fulfilment vendor never receives raw card data.
- Subscription pricing shipped end-to-end and verified against production — existing subscribers deliberately left untouched pending a separate ruling.

## Operated, not just deployed

- Uptime monitors, error tracking, and scheduled-job monitoring with alerting.
- Staged deploys — staging proven before production, releases in small reviewable steps.
- A self-serve developer portal issuing scoped API keys with signed webhooks, live in production.

# What an agency can do today.

06

## 01 Send a real gift through one flow

Postcard or letter, gift card, fresh flowers, a memorial tree, or branded merchandise — one checkout.

## 02 Manage a client book

Bulk import from Excel, duplicate handling, per-client gift opt-out.

## 03 Control who can spend what

Owner, Producer, and CSR roles with per-role spending limits and a spending dashboard.

## 04 Fund and track spending

A prepaid wallet with transaction history, alongside a subscription tier.

## 05 Brand what goes out

Upload an agency logo — applied to merchandise and print output.

## 06 Connect their own systems

A self-serve developer portal with scoped API keys, signed webhooks, interactive docs.

## 07 Run a referral programme

One referral, one month free — gated on the referred agency completing payment.

**And what didn't ship — on purpose.** A substantial AI feature set was scoped in depth, then removed from the roadmap at the client's direction — effort went to the fulfilment and billing work above instead. Scope that didn't serve the product didn't ship.

# Real product. Real payments. And the client owns all of it.

## 1 The client owns the running system

Repositories, databases, hosting, and DNS all live in the client's own accounts — the client owns the asset; Project Assistant operates and maintains it.

## 2 Real money moves through it

Live payment processing since June 2026. Real mail printed and posted, real gift cards funded, real flowers delivered.

## 3 Real agencies use it

Customer agencies run on production with real client books — and appear as customers on the product's public site.

## 4 The client kept deciding

Nine months of dated, prioritised, written client feedback — engagement that outlasted launch.

## 5 Operated like it matters

PA runs the platform day to day — monitoring, error tracking, staged deploys, incident response. Production discipline, ongoing.



A REAL PRINTED POSTCARD, RENDERED FOR MAIL — TEST DATA SHOWN

CUSTOMER AGENCY LOGOS · [ AWAITING  
REUSE CONSENT ]

# Ideas enter. Software ships.

Project Assistant turns business ambitions into useful software, with one team taking responsibility for seeing the work through — from the first hard question to production, and operating what we build.

---

## Your ambition.

Complex digital products, web applications, AI agents, and automation — built around your business goals.

## Our responsibility.

We sell outcomes, not hours. One team, accountable end to end, delivering against what success looks like.

**project assistant.**