

10 Infrastructure Decisions Before Brand #2 — Manyi Life

10 Infrastructure Decisions to Make Before You Launch Brand #2

A field guide for small teams about to split into a multi-brand operation.

By the team at Manyi Life — we did this with three brands, four domains, one membership system. Here's what we'd tell ourselves at the start.

Why this exists

We see a pattern: a team ships one product. It works. They get an idea for a second product — different audience, different tone, different URL. They start building brand #2 the same way they built brand #1. Six months in, they realize that the foundation that worked for one product silently makes the second product 3× harder.

This isn't a "do these things" checklist. It's **10 decisions you'll make whether you mean to or not** — better to make them on purpose.

Each decision has: - **The question.** What you're actually deciding. - **The cheap-now / expensive-later trap.** The default that feels fine today and bites in year 2. - **What we did, and what we'd do again.**

Decision 1 — Where does identity live?

The question: When a user signs up for brand #1 and then visits brand #2 next year, what makes them "the same person"?

The trap: Each product gets its own user table. Year 2: you cannot answer "how many humans use ≥2 products" without exporting CSVs and joining on email — and email changes.

What we did: Logto as the OIDC authority across all properties. Each brand cookie is brand-local; identity is shared at the IdP layer. We can answer cross-brand questions on a single SQL row.

Decision 2 — One audience or many for email?

The question: When brand #2 launches, do its email contacts go into a new ESP audience or merge into the same one?

The trap: "Separate audience per brand for cleanliness." Looks tidy. Then you cannot ever segment "readers active in ≥ 2 brands" without a manual export pipeline. And you're paying 2–3× audience tier costs.

What we did: Single Resend audience tagged by brand. Cross-brand reporting is one query, not a project.

Decision 3 — Cookie domain strategy

The question: Will the brands share a top-level domain (subdomains of one TLD) or live on different TLDs?

The trap: Going to different TLDs without planning for the cross-brand sign-in UX. Then bolting on token-in-URL workarounds. (Don't do this. It's a security smell.)

What we did: Different TLDs (brand-equity is worth it), per-brand session cookies, OIDC redirect for cross-brand sign-in. Costs ~300ms on first cross-brand visit. Worth it.

Decision 4 — Hosting: one bill or many?

The question: Do all brand sites run on the same hosting platform, or are they free to choose independently?

The trap: Independence sounds good until the third brand is on a different platform, the fourth uses a different framework, and ops debugging requires four logins.

What we did: All four properties on Cloudflare Workers (OpenNext adapter). One vendor for compute, storage, DB, queues. One bill. One dashboard for ops.

Decision 5 — Analytics: cross-brand from day one, or bolt on later?

The question: Where do user events go, and is the schema designed to answer cross-brand questions?

The trap: Each brand fires events to its own GA4 property with no cross-brand identifier. Year 2: every cross-brand question requires you to manually align user IDs across properties (often impossible).

What we did: A central D1 table fed by all four properties, every event carrying the same `external_user_id` (Logto's sub). Cross-brand attribution = SQL JOIN.

Decision 6 — Repo structure: monorepo or per-brand?

The question: Does all code live in one repo, or each brand in its own?

The trap: Defaulting to monorepo because "best practices." Then CI gets slow, one brand's broken build blocks the others, and every PR touches concerns from three brands.

What we did: One repo per brand, plus a group repo for cross-brand admin and the public group site. Code duplication has cost less than the autonomy has saved. Revisit when you hire team #2.

Decision 7 — What's the canonical user record's source of truth?

The question: When a user updates their email in brand #1, what propagates to brand #2 and when?

The trap: Multiple writable user tables. They drift. Three months in, "is the email in brand #1 or brand #2 correct?" becomes a real customer-support question.

What we did: Logto holds the canonical record. Brand databases store only `external_user_id` (Logto sub) — never raw PII. Every brand fetches profile fields fresh on each session.

Decision 8 — Cron and background jobs

The question: Where do scheduled tasks live, and how do they survive a brand-level deploy?

The trap: Each brand has its own cron via Vercel / Render / GitHub Actions. Each is configured slightly differently. Failure modes diverge. When a job silently stops, you find out from the missing report.

What we did: Cloudflare Workers Cron Triggers, declared in each brand's wrangler.toml. Centralized observability dashboard. Every job emits a heartbeat to D1 — missing heartbeat alerts via email.

Decision 9 — The “group public site” question

The question: Should the parent brand have a public website, or just be invisible plumbing?

The trap: Skipping the group site because “it’s just internal.” Then partners / press / investors / candidates have nowhere to land. You retrofit it under pressure, badly.

What we did: Built the group site (this one — manyilife.com) early. Hosts the strategy story, the brand portfolio, careers, investor relations. Costs ~1 page-week per month to maintain. Has paid for itself in partnership conversations.

Decision 10 — Who can do what across brands?

The question: Inside your team, who can ship to which brand?

The trap: Everyone has root everywhere. Easy to give access; hard to take away. One mistake on brand #2 affects all three.

What we did: Role-based access at the admin layer (manager / superadmin). Each brand's Cloudflare account / DNS / production secrets are reachable only by named individuals. Audit log on all admin actions.

What you do next

If 5+ of these resonate, you're past the “one-product” stage and into the “multi-brand operating system” stage. The decisions you make in the next 90 days will set the shape of the next 3 years.

Three concrete moves you can do this month:

1. **Document your current state** for each of the 10 decisions above. Even “we haven't decided” is a decision.

2. **Pick the 3 with highest blast radius** at your projected 12-month scale (usually: identity, audience, hosting). Make those on purpose.
 3. **Defer the rest.** Don't pre-optimize the 7 that aren't blocking you yet.
-

About Manyi Life Advisory

We're a 1-person, 3-brand, 4-domain operation. Everything in this guide is something we ship, run, and fix. If you want a second pair of eyes on your specific stack, we offer a **Stack Audit** engagement (4–6 weeks, scoped per team).

→ Talk to us at manyilife.com/contact

This guide is intentionally generic. Your specific situation has details that change the recommendations. Don't treat any line above as gospel — treat them as the questions to ask out loud.
