

啟動第二品牌前要做的 10 個基礎設施決策 — Manyi Life 漫翊

啟動第二品牌前要做的 10 個基礎設施決策

給準備從一個品牌長到多品牌的小團隊一份實戰指南。

by Manyi Life 漫翊團隊 — 我們做過了：三個品牌、四個域名、一套會員系統。以下是我們希望當初就有人告訴我們的事。

為什麼有這份指南

我們看過一個模式：團隊做出一個產品，跑得起來。靈感來了，要做第二個——不同受眾、不同調性、不同網址。他們用建第一個品牌的方法建第二個。六個月後，發現第一個品牌的地基讓第二個品牌默默變成 3 倍難。

這不是「該做這些事」的 checklist。是 10 個你不管有沒有意識到、都會做的決策——不如刻意做。

每個決策有：– 問題是什麼。你實際在決定什麼。– 現在便宜、之後昂貴的陷阱。今天看起來沒事、第二年咬你的預設。– 我們做了什麼、會不會再做一次。

決策 1 — 身份資料住在哪？

問題：用戶在品牌一註冊，明年來到品牌二，什麼東西讓他「是同一個人」？

陷阱：每個產品自己一張 user table。第二年：你無法回答「同時用 ≥ 2 個產品的活躍人數」，除非匯出 CSV 然後用 email join——而 email 會變。

我們做了什麼：Logto 做跨四個域名的 OIDC 身份權威。每個品牌的 cookie 是品牌本地的；身份在 IdP 那一層共享。跨品牌問題用一條 SQL 就能答。

決策 2 — Email 一個 audience 還是多個？

問題：品牌二上線時，它的 email 名單是進新 audience 還是和品牌一合併？

陷阱：「每品牌一個 audience 看起來乾淨」。結果你永遠分不出「同時訂閱 ≥ 2 品牌的活躍讀者」，除非建立手動匯出 pipeline。而且付 2-3 倍 audience tier 費用。

我們做了什麼：單一 Resend audience，用 brand tag 區分。跨品牌報表一個 query，不是一個專案。

決策 3 — Cookie 域名策略

問題：品牌之間共用一個 top-level domain (subdomain 形式) 還是各自獨立 TLD？

陷阱：走獨立 TLD 但沒規劃跨品牌登入的 UX。然後事後補 token-in-URL 暗渡——別這樣，這是安全味道很重的訊號。

我們做了什麼：各自獨立 TLD (品牌獨立性值得) + 每品牌本地 session cookie + OIDC redirect 處理跨品牌登入。第一次跨品牌跳轉約 300ms 延遲。值得。

決策 4 — Hosting：一張帳單還是多張？

問題：所有品牌站跑同一個 hosting 平台，還是各自獨立選？

陷阱：「獨立」聽起來不錯，直到第三個品牌跑在另一個平台、第四個用不同 framework，ops debug 要登入四個地方。

我們做了什麼：四個 property 全部在 Cloudflare Workers (OpenNext adapter)。一個供應商管 compute / storage / DB / queues。一張帳單、一個 dashboard。

決策 5 — Analytics：第一天就跨品牌，還是之後再說？

問題：用戶事件流到哪、schema 是不是設計成可以回答跨品牌問題？

陷阱：每個品牌打事件到自己的 GA4 property、沒有跨品牌識別 ID。第二年：每個跨品牌問題都要手動對齊 user ID (通常做不到)。

我們做了什麼：集中一張 D1 表，四個 property 都餵到這裡，每個事件都帶同一個 external_user_id (Logto 的 sub)。跨品牌歸因 = SQL JOIN。

決策 6 — Repo 結構：monorepo 還是 per-brand ?

問題： 所有 code 在一個 repo，還是每個品牌自己一個？

陷阱： 預設 monorepo 因為「best practice」。然後 CI 變慢、一個品牌壞掉擋住其他人、每個 PR 同時碰到三個品牌的關注點。

我們做了什麼： 每個品牌獨立 repo，加一個集團 repo 放跨品牌 admin 和公開集團站。Code 重複的成本目前低於獨立性帶來的好處。等團隊到 2+ 人再評估。

決策 7 — 用戶資料的單一真實來源是什麼？

問題： 用戶在品牌一更新 email，這份更新什麼時候、怎麼傳到品牌二？

陷阱： 多張可寫的 user table，會 drift。三個月後「品牌一還是品牌二的 email 是對的」變成真實的客服問題。

我們做了什麼： Logto 持有 canonical record。品牌 DB 只存 external_user_id (Logto sub) ——從不存原始 PII。每個品牌在每次 session 重新抓 profile fields。

決策 8 — Cron 和背景任務

問題： 排程任務住在哪、品牌單獨部署時怎麼存活？

陷阱： 每個品牌自己一套 cron (Vercel / Render / GitHub Actions)，每個設定都不太一樣，失敗模式各異。一個 job 默默停了，是你發現「該收到的報表沒收到」才知道。

我們做了什麼： Cloudflare Workers Cron Triggers，宣告在每個品牌的 wrangler.toml。集中觀測 dashboard。每個 job 都打一筆 heartbeat 到 D1——心跳消失就 email 告警。

決策 9 — 集團公開站要不要做？

問題： 母品牌要有公開網站，還是當不可見的水管？

陷阱： 跳過集團站因為「反正是內部」。然後合作夥伴 / 媒體 / 投資人 / 求職者沒地方落地，臨時被迫補上、做得很差。

我們做了什麼： 早期就建集團站（你正在看的 manyilife.com）。承載策略敘事、品牌組合、人才招聘、投資人關係。每個月大約一個 page-week 維護成本。在合作對話中早就回本。

決策 10 — 跨品牌的權限分配

問題： 團隊內部，誰可以對哪個品牌做什麼？

陷阱： 大家都有 root 全部。給很容易、收很難。一個失誤在品牌二會影響全部三個。

我們做了什麼： Admin 層走 role-based (manager / superadmin)。每個品牌的 Cloudflare 帳號 / DNS / production secrets 只有指名個人能接觸。所有 admin 操作有 audit log。

接下來怎麼做

如果這 10 條有 5 條打到你，你已經過了「單一產品」階段，進入「多品牌作業系統」階段。接下來 90 天的決策會塑造未來 3 年的形狀。

這個月可以做的三件事：

1. **記錄你目前的狀態**——10 個決策每個都寫下來。即使「我們還沒決定」也是一個決策。
 2. **挑 3 個 12 個月後爆炸範圍最大的**（通常是：身份、audience、hosting）。刻意做。
 3. **其他 7 個延後**。不要對還沒擋你的東西做 pre-optimization。
-

關於 Manyi Life Advisory

我們是 1 人、3 品牌、4 域名的營運。這份指南裡的每件事都是我們真的在做、在跑、在修的。如果你想找人對你的具體 stack 給第二意見，我們提供 **Stack Audit** 服務（4–6 週，依團隊範圍報價）。

→ 寫信給我們：manyilife.com/contact

這份指南刻意寫得通用。你的具體狀況會有細節改變建議。不要把任何一行當成定論——當作該大聲問出來的問題就好。

