

張育誠 Taco

React + TypeScript

Vue.js

Monorepo

Design System

Internal Tooling

CI/CD Automation

AI-Assisted Development



私立大葉大學 / 資訊管理學系 大學畢業

E-MAIL	LINKEDIN	GITHUB
tabacotaco@gmail.com	https://linkedin.com/in/tabacotaco	https://github.com/taco3064

工作經驗 EXPERIENCE

2026/2 – 2026/9
8 個月

Sr. Frontend Engineer

Cypher Lab Sdn. Bhd.

前端工程師 · 馬來西亞

— 建立 Token 化 Design System，將所有顏色集中管理，並透過自訂 Build Plugin 與 Lint 規則避免直接使用硬編碼色彩。

— 將大型單體元件重構為具分層設計、可獨立測試的模組，並將測試與實作集中管理。

— 建立 AI 協作開發流程，包含可重複使用的 Agent Skills、Design-to-Code Validation 流程，以及可重播的資料蒐集流程，讓需求能更有效率地轉換為正式功能。

註：本職務因公司停止營運而終止

#AI Agent Workflow

#React.js

#Vue.js

#GitHub Actions

#TypeScript

2025/5 – 2025/10
6 個月

Sr. Frontend Engineer

BTSE

前端工程師 · 台北市信義區

— 開發一套基於 AST 的 CLI 工具，用於分析 Router 層級依賴關係，並追蹤程式修改所造成的影響範圍。

— 在高度依賴跨部門流程的多白牌系統中，處理修改影響、測試範圍與交付不確定性，累積對責任邊界與流程透明度的實務經驗。

#Vue.js

#AST

#WebSocket

#GitLab Runner

#Jira

2023/6 – 2025/4
1 年 11 個月

Sr. Frontend Engineer

Clinico

前端工程師 · 新北市中和區

— 開發 CRM、BPM、LIFF 等內外部系統，透過可重複使用的架構模式與一致的資料流，降低功能間的維護成本。

— 重構 BPM 架構，將表單層的共用邏輯抽離，提升系統擴充性並減少重複實作。

— 將多套前端服務整併至 Nx Monorepo，統一共用模組與工具，並在各產品間落實 SOLID 設計原則。

— 建立 LIFF 共用架構層，統一所有 LIFF 頁面的登入、路由以及 UI 行為。

— 推動程式品質改善，整理共用模式、建立一致性，並持續透過重構降低後續維護成本。

#ReactJS

#TypeScript

#GraphQL

#Monorepo

2019/10 – 2023/4
3 年 7 個月

Front-end Engineer

Gorilla Technology Group

前端工程師 · 台北市內湖區

- 將既有 React 專案由 v15 升級至 v16，透過重整共用 State 與 Component，降低升級過程中的衝擊。
- 建立 Config-driven 的 No-code Framework，透過結構化設定動態產生 React 頁面，而非依賴大量硬編碼。
- 建置並維護前端 Shared Packages，集中管理共用邏輯，確保不同產品之間的一致性。
- 導入 Nx Monorepo，整合程式碼、型別與共用工具，提升專案擴充性與團隊協作效率。
- 建立 GitHub Actions 與 GitLab Runner 自動化流程，統一 Build、Test 與 Release 作業。
- 建立開發規範，包括 Folder Structure 與 Git Branch 管理流程。

#ReactJS

#TypeScript

#Monorepo

#GitLab Runner

#GitHub Actions

2016/7 – 2019/10
3 年 4 個月

Software Engineer

Walsin Lihwa

軟體工程師 · 台北市信義區

- 開發 ERP／MES 內部系統，建立可重複使用的 UI 模式與一致的資料流。
- 規劃並執行前端教育訓練，將技術概念整理成容易理解與傳承的開發模式。

#AngularJS

#GitLab

#Java / Spring

#MS SQL

#Bootstrap

2010/8 – 2016/6
5 年 11 個月

Software Engineer

Evergreen International

軟體工程師 · 台北市中山區

- 建立可透過設定完成欄位轉換的 EDI 系統，降低客製化開發需求，提升長期維護性。
- 建立 Ext-JS UI／UX 抽象層，統一元件行為與開發模式，讓工程師能更專注於商業邏輯，同時維持整體程式一致性。

#Ext.js

#Java / Spring

#Oracle

專長

CAPABILITIES

01	前端架構	專注於打造可擴充、可維護且能持續演進的前端架構。 擅長透過清楚的模組邊界、共用抽象層與一致性的設計，降低系統複雜度；具備架構設計、Monorepo、Shared Library、依賴治理與工程規範建立經驗。
02	前端系統設計	擅長將複雜的商業需求整理成清楚、可測試且容易維護的前端系統。 著重狀態管理、元件架構、Design System、效能優化與可持續演進的程式結構。
03	工程生產力	透過工具、自動化流程與工程規範，提升團隊的開發效率與維護品質。 具備內部 CLI、ESLint 規則、Code Generation、Git Hooks、CI 品質管控與可重複使用流程的實作經驗。
04	AI 協作開發	將 AI 視為工程協作的一部分，而不是單純的程式碼生成工具。 建立 Agent Skills、Prompt Workflow、Design-to-Code Validation 與工程 Guardrails，同時保留工程師對品質與架構的最終判斷。
05	技術領導	透過架構設計、工程規範、技術文件與知識分享，推動團隊建立一致的開發方式。 相信良好的工程文化來自清楚的設計原則與可落實的流程，讓團隊持續交付高品質且容易維護的軟體。

我的工程理念一直在追求同一件事情：

一路走來，我發現自己堅持的其實是「唯一真相」。

SINGLE SOURCE OF TRUTH

一份資料、一條規則、一個模組的責任，都應該有明確的來源。當同一件事情散落在不同地方，需要依靠人的記憶維持一致，複雜度就會開始累積。

職涯早期，我先在 Evergreen International 把反覆出現的 EDI 轉換與介面行為整理成可配置、可共用的能力；到了 Walsin Lihwa，則進一步把個人的實作方式整理成團隊可以沿用的 UI Pattern 與訓練內容。當時我還沒有替這件事取名字，只知道重複的規則不應該各自存在。

在 Gorilla Technology Group，問題擴大到多品牌、Shared Packages 與不同交付流程。我開始用 Config-driven Framework、Monorepo、依賴邊界與 CI/CD，讓程式、版本與團隊遵循同一套結構。Folder Structure 也不再只是一份文件，而是能由圖說明、由 ESLint 執行的架構規則。

到了 Clinico，我更清楚看見「共用」不是把所有東西放在一起，而是辨認哪些行為本質相同，讓它們只存在一份；品牌差異則必須保留清楚的覆寫邊界。無論是 BPM、LIFF、多品牌服務，或 GraphQL 與 REST 之間的資料模型，我處理的都是同一件事：避免相同邏輯在產品演進中逐漸分叉。

BTSE 讓這個原則從程式架構延伸到協作。多白牌系統的修改範圍不能靠猜，跨部門的需求、交付與測試責任也不該各自解讀。因此，我嘗試把依賴關係計算出來，把受影響的頁面交給 QA，也開始要求所有角色在同一份事實上對齊。

在 Cypher Lab Sdn. Bhd.，AI Agent 進入開發流程後，這個問題變得更加直接。AI 可以快速產生程式碼，但如果架構規則散落在不同人的記憶、文件與工具中，它只會更快地放大不一致。於是我把分層、邊界、Lint、CI、文件與 Agent Contract 收斂成 Blueprint，讓人與 AI 都從同一個來源理解系統。

01	02	03
減少重複的真相	建立清楚的邊界	讓人與 AI 都能理解系統真正的規則

回頭看，我一路做的並不是彼此無關的架構工具。從可配置系統、共用 Pattern、Monorepo、影響分析，到 Architecture as Code，我始終在追求同一件事：讓系統真正的答案只有一份，而且能被理解、驗證並持續演進。