



Kevin Chuang 莊哲愷

平台 / AI 工程 | 11 年 C# / .NET

十一年 C# 後端出身。擅長快速學習與整合，做的研究常常訂出規則、寫成流程，成為團隊依循的內部工具。

kevin950805@gmail.com 0928-004-925 github.com/chuangkevin

linkedin.com/in/kevin-chuang-5ba83728/ 台北

我希望能長期在一個團隊裡共同成長、往前進，因此僅考慮直接聘僱、不接受客戶端駐點。



kevin.sisihome.org

#自動化

#平台工程

#AI工作流

#攝影

#3D列印

11 年

C# / .NET 後端

40 個

repo 導入工具

3,172

次 AI 自動審查

779

次 GitOps 部署

24 位

開發者採用

工作經歷

資深工程師 · 永慶房產集團

2023/3 - 2026/7

- 帶 4 人團隊 · 釐清需求 → 設計 API → 拆工分派** 主導租賃物件專案、帶 4 人開發團隊：與 PM 需求部門溝通釐清需求，設計資料表與 API 介面，再把各端工作與 API 分派給團隊成員平行開發，產出房屋物件維護系統——B 端物件刊登、C 端一般客戶瀏覽，租屋物件約 10 萬筆。本職 Web 後端（C# / .NET Web API）與系統架構設計，日常使用 Elasticsearch、SQL Server、Redis、RabbitMQ。
- 以告警門檻反推分批節奏** 為公司 AI Agent 專案（檢索服務由外部廠商提供）設計跨系統資料分批結轉的工程判斷：事前評估目標端消化速率，並以自建監控的既有告警門檻為約束，反推批次大小與節奏。結轉標的為社區約 6.8 萬筆 + 買屋物件約 24 萬筆（累計約 30 萬筆）。
- TFS → 自建 CI/CD 交付鏈** 從零自建整條軟體交付鏈：自主研究架設 Kubernetes cluster，於其上自架 Gitea（程式碼託管）、Gitea Actions CI/CD 與自建 DinD runner，再擴建為 Argo CD 的 GitOps 部署平台。原本後端走 TFS、其餘服務無平台可跑；如今前後端與其他自架服務全數遷移至此（38 個 Application、累計 779 次部署，git push 即全自動）。
- 逾 3,000 次觸發** 從零打造並跨部門推動導入內部 AI 工具鏈：其中 AI Code Review 已成研發團隊 push 前的例行檢查——上線至今觸發逾 3,000 次、涵蓋 40 個 repo、24 位開發者的 PR；另打造 AI 原型生成（Design Bridge）與規格驅動自動化測試（Auto Spec Test）等實驗性工具，並負責容器化部署與 CI/CD 管線建置。
- 全部門採用** 建立技術與非技術部門共用的文件協作平台 Note——同一份 repo，非技術人員用無感介面編輯、工程師可在 Note 或 Gitea 操作，共用同一個真相來源；HTML 可一鍵分享成獨立網站，曾有 PM 在 Note 上測試後將程式碼送上正式環境。

軟體工程師 · 全聯實業股份有限公司

2020/9 - 2023/2

- 開發、設計與維護資訊部 ERP 系統（C# WinForms），主導專案完整流程：前置規劃、流程分析、可行性評估與技術選型導入、系統分析與設計、開發測試，以及跨部門協調。
- 數天 → 數分鐘** 資料交換流程自動化：以 Reflection 將資料庫結構自動轉為程式格式，原本數天的工時縮短為數分鐘。
- 門市行銷曲目派發系統（MQTT + REST + Docker）：門市與總部曲目交換全自動化，同時大幅減少長年硬體故障維護問題；另建內部知識庫問答 BOT（Microsoft LUIS 語意識別）。

- 無線電產品軟硬體整合開發（C# WinForms）：負責派遣台系統的開發規劃，設計 Service 核心與 SIP/MQ 通訊、Command 傳輸介面。
- Line + Email 多通道告警系統：選 Line 是因它為台灣使用率最高的通訊軟體、同時保留 Email 照顧不同使用習慣，透過 Messaging API 即時推送設備異常。
- 設備環境監控（SNMP Polling/Trap 為主、爬蟲 + Regex 補足無介面設備）；派遣台 PTT 麥克風雞型（自學 Onshape CAD/3D 列印/Arduino，於公司預定 Demo 前完成並交付工程部）；兼任 MIS 網路與伺服器維運。

代表作品

Note — 技術與非技術部門共用的文件協作平台 平台工程

讓非技術人員用熟悉的 Google Docs 體驗寫文件、規格書，底層直接存進 Git；工程師也在同一套系統看得到這些文件。文件可以直接播成投影片，HTML 也能一鍵分享成獨立網站。

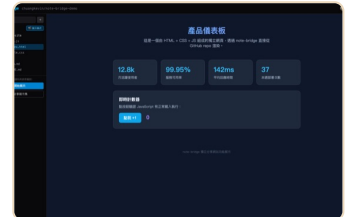
► 全部門採用 · 204 份文件 · 22 位使用者

一份文件、兩種介面，同一個真相來源。非技術人員在 Note 上用所見即所得的方式編輯，涵蓋大多數 Google Docs 的能力，不需要理解 Git；工程師則可以選擇在 Note 上讀寫，或直接用 Gitea 操作同一份 repo。兩邊看到的永遠是同一份內容，版本控制是底層 Git 免費附贈的。

解法 建立技術與非技術部門共用的文件協作平台：同一份 repo，非技術人員用無感的編輯介面，工程師可在 Note 或直接用 Gitea 操作，兩邊共用同一個真相來源；文件可播成投影片，HTML 可一鍵分享成獨立網站。

成果 目前累積 204 份文件（Markdown 165 份、HTML 39 份），分布在 14 個 repo，22 位使用者，近四週每週有 5-10 位不同作者實際編輯。除了文件協作，HTML 分享成獨立網站的功能也讓非工程人員有了自己的發布管道——曾經有 PM 在 Note 上完成測試後，把程式碼送上正式環境。

React 19 · TypeScript · Node.js / Express · SQLite · marked / DOMPurify · GitHub REST API



AI Code Review Action — Multi-Agent PR Review 平台工程

真實上線的 human-in-the-loop 審查系統：AI 產出審查建議，24 位工程師每天在 40 個 repo 上接受或駁回，我再依回饋迭代——AI 建議與人的判斷形成閉環，而不只是一個丟出去的 AI 工具。

► 3,172 次觸發 · 40 repo · 24 位開發者 · 6,240 則 comment

Gitea CI Action：PR 建立/更新時自動觸發 5 個不同人設的 AI reviewer（前端 / 後端 / 資安 / Domain / 架構）平行審查，以 inline comment + 團隊討論總結形式提交 PR Review。關鍵是人機閉環——工程師在實際 PR 上接受或駁回這些建議，這些回饋再回頭修正 reviewer 的人設與準則，而不是一次性生成就結束。

解法 5 個帶專業人設的 reviewer（前端/後端/資安/Domain/架構）依內部準則平行審查 PR，彙整 inline comments 與總結；審查主體演進至帶長期記憶的 Hermes Agent，並檢索 codebase 既有解法。

成果 已是研發團隊 push 前的例行檢查：自 2026-04-02 上線至今觸發 3,172 次（PR 事件 2,087 + 留言回覆 1,085），涵蓋 40 個 repo（正式專案 37 個）、24 位開發者的 PR，累計 review 974 個 PR、產出 6,240 則 inline comment（平均每次 review 3.77 則）。

Go · Gitea Actions · opencode (self-hosted AI provider) · Hermes Agent · Multi-Agent Orchestration · CI/CD

GitOps Platform 平台工程

從零建一條軟體交付鏈，讓工程師送出程式碼就自動上線；目前承載 38 個服務、累計 779 次部署。

► 779 次部署 · 38 Application · 2 座 k3s

從零自建整條軟體交付鏈，並把公司既有的前後端與其他服務全面遷移上來：自架 Gitea（程式碼託管）、Gitea Actions CI/CD 與自建 DinD runner，於自研架設的 Kubernetes cluster 上以 Argo CD 做 GitOps 部署（連 Gitea 本身也跑在這個 cluster 上）。

解法 自建 K8s cluster → 自架 Gitea + Gitea Actions CI/CD + 自建 DinD runner → 以 Argo CD 做 GitOps 部署；前端、後端與其他自架服務的原始碼與 CI/CD 全數遷移至此（含 Gitea 本身也跑在 cluster 上）。

成果 從無到有的團隊軟體交付鏈，目前承載 38 個 Argo CD Application、累計 779 次部署（2025-04-08 起、近四週週均約 41 次）、2 座獨立 k3s（測試機＋正式機）；開發者只需 git push 一步即全自動部署（部署步驟本身僅 3-52 秒）。

Gitea + Actions CI/CD · DinD runner · Argo CD / GitOps · Kubernetes / k3s · Docker · Git

Open Data Service — ASP.NET Core 6 後端 API 後端

把分散的政府公開資料，整合成一套穩定、有單元測試的後端 API；程式碼公開可直接檢視。

► C# / .NET 6 · 五層架構＋單元測試

ASP.NET Core 6 公開資料整合服務：把政府 OpenData 的公車動態 / 路線 / 到站時刻等分散端點，統一成一套受 JWT 保護的 REST API，並以 Hangfire 排程定時抓取與快取。

解法 五專案分層方案（Web API / Service / Repository / Common / Tests）搭配介面化 DI（各層以 ServiceCollection extension 註冊）：Controller → Service（Mapster 做 DTO 對映）→ Repository（SQL Server）。上游整合封裝在具 token 換發（TokenService / JWT）與自訂 HttpClient（處理政府站點憑證）的 outbound service；Hangfire（SqlServer storage + Dashboard）排程定時抓取公車路線寫入 DB；Swagger 附 JWT 安全定義與 XML 文件；整包以 Dockerfile 容器化。

成果 分層清楚、可測試的 .NET 6 後端：Controller / Service / Repository 三層以介面解耦，Hangfire 定時抓取上游、JWT 保護 endpoint，核心 Service 附 AAA 單元測試。作為 C# / .NET 後端架構與測試實務的公開程式碼佐證。

C# / .NET 6 · ASP.NET Core Web API · SQL Server · Hangfire · Mapster · JWT Bearer

Enterprise Observability Platform 監控

建立集中監控平台，把散在各系統的服務狀態收進同一個儀表板；目前納管 20 個監控目標、14 台主機。

► Metrics / Logs / Tracing 三支柱收斂 · OpenTelemetry 自動納管

集中式 Observability 平台，整合 Metrics / Logs / Tracing，提供單一儀表板檢視多服務健康狀態。

解法 建立 Prometheus（Metrics）+ Loki（Logs）+ Tempo（Tracing）整合平台，透過 OpenTelemetry SDK 自動串接服務端，Grafana Dashboard 統一呈現。

成果 Metrics / Logs / Tracing 收斂到單一儀表板，問題定位不再跨系統切換；OpenTelemetry SDK 讓服務端自動接入，新服務零成本納管。目前納管 20 個監控目標（全部 up）、涵蓋 14 台主機（13 Windows + 1 Linux），並設有 8 條 Prometheus 告警規則。

Grafana · Prometheus · Loki · Tempo · OpenTelemetry · Docker

Design Bridge — AI 原型生成平台 AI 應用

做一個工具，用一段文字描述就能生出可點擊的網頁原型，供各部門提需求時直接看到畫面；內部實驗性專案。

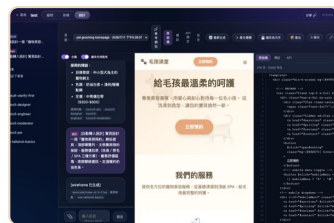
內部實驗性專案

AI 驅動的原型生成工具：用自然語言描述 UI，AI 幾秒內產出可互動的多頁面 HTML 原型。內部實驗性專案——目標同樣是把各部門的原型需求集中到單一平台，但採用程度尚未達到 Note。

解法 4-Agent 結構化討論（PM → UX → QA → Tech）依準則自動產出可互動網頁與流程圖，QA Validator 自動檢查空頁／死角並 retry，元件級微調讓 AI 只改點選的元件、不必整頁重生。

成果 內部實驗性專案：以 4-Agent 合議＋準則束縛，把自然語言需求變成可互動的多頁面原型（含設計準則、API 清單與對應程式碼）。定位是把各部門的原型需求集中到單一平台，但採用程度尚未達到 Note。

Node.js · TypeScript · Express · React · Multi-Agent Orchestration · Google Gemini API



Auto Spec Test — AI 自動化測試平台 AI 應用

做一個工具，上傳規格書就能自動測試網站，並產出含截圖的測試報告；內部工具，正式環境尚未實際採用。

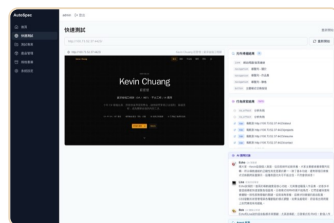
► 44 張 bug issue（測試專案）

規格書驅動的 AI 測試工具：上傳規格書或貼上 URL，AI 自動產出測試腳本、透過內嵌瀏覽器逐條執行並產出含截圖的測試報告。

解法 雙模式：上傳規格書 → AI 產出並逐條執行測試腳本；或貼 URL → 多 Agent 掃描元件、討論策略並深度探索建立整站地圖，三法官（嚴格／寬鬆／仲裁）評判 PASS/FAIL。

成果 在測試專案上實際跑出成果：一次測試回合（2026-03-26 ~ 03-31）自動產出 44 張可追蹤的 Gitea bug issue，每張含重現步驟與截圖。網頁無邊際探索能力（最大廣度 × 最大深度擷取頁面）為自行訓練的核心技術。此為內部工具，正式環境尚未實際採用。

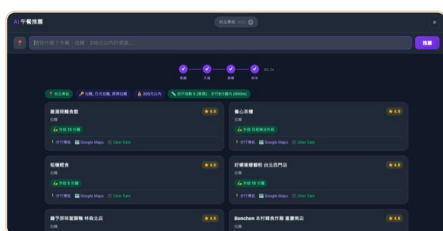
TypeScript · Node.js · Browser Automation · Multi-Agent Orchestration · Google Gemini API · Docker



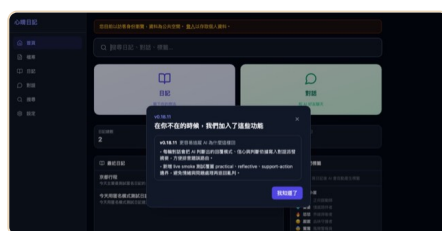
實際系統畫面（節錄，更多見線上版）



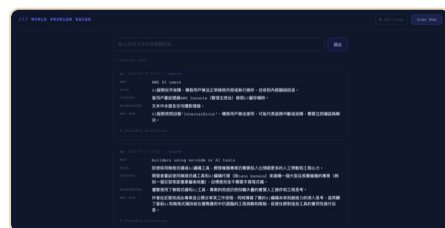
SkyNet



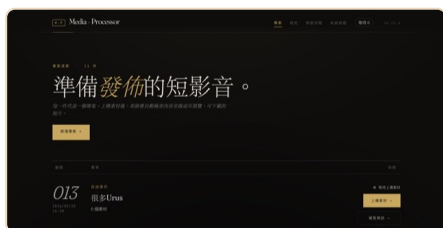
Context-aware Recommendation Platform



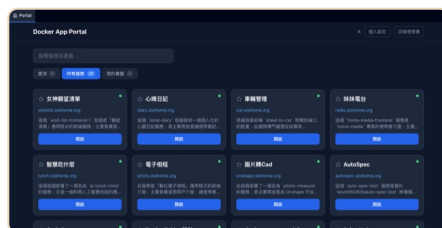
Mind Diary



Personal AI Automation Platform



Media Processor Video Pipeline



HomeLab Self-hosted Platform

其他作品

Skynet — 貼上任何 URL，系統自動推斷服務類型並開始監控——不用手動設定探針。概念延續我在公司做的監控平台，這是個人自架的簡化版。

Context-aware Recommendation Platform — 情境感知推薦系統。整合天氣、地點、預算、時間等外部資料，提供個人化的餐飲推薦。

Claude Code Remote — Web 介面讓任何裝置存取 Claude Code CLI：即時 WebSocket 串流對話、圖片上傳、session 管理，並支援最多 20 個並行 AI agent 的自動派工。

@kevinsisi/ai-core — Provider-aware 的多 provider AI runtime 套件，供旗下多個服務（mind-diary、Design Bridge、auto-spec-test...）共用：KeyPool、retry、streaming、tools、agent-runtime。

Mind Diary — AI 輔助日記與個人知識整理系統。13 個 AI 角色獨立分析後，Master Agent 整合產出。

Personal AI Automation Platform — Personal AI Agent / Automation Framework。探索 AI Agent、自動化流程與資料來源整合。

Media Processor Video Pipeline — 影音處理自動化流程。Multi-worker GPU/CPU 架構，支援 AI 場景標記、自動運鏡與數位防手震。

HomeLab Self-hosted Platform — Self-hosted infrastructure。把部署、監控、反向代理與 AI 工具鏈整套自架，跑在 Tailscale mesh network 上。

Sheet-to-Car Inventory Platform — 中古車庫存管理平台。將試算表流程轉換為 domain-driven Web 應用，同時作為異質平台 API 整合與 key 調度的壓測場域。

Home Media — 類廣播電台的多人同步聽歌平台：任何人可以開一個「台」當 DJ 放音樂，有興趣的同事點進去就能即時跟聽。DJ 切歌或調整播放進度，所有在台裡的人近乎毫秒級同步跟隨。

Side projects：Key Manager · Onshape CAD AI FeatureScript Generator · 攝影展示網頁 · Greed Island · Ebook Reader · 手搖磨豆機電動化改裝

技術能力

熟悉

C# / .NET (11 年：WinForms → Web API) · ASP.NET Web API 後端 / 架構設計 · SQL Server · Elasticsearch · Redis · RabbitMQ · TypeScript · Node.js 22 · Express · React 19 · Next.js 15 · Vite · Tailwind CSS 4 · SQLite (better-sqlite3, Drizzle) · Docker · Docker Compose · Git / GitHub Actions · Tailscale · Caddy (xcaddy + cloudflare DNS plugin) · Linux (Manjaro / Debian)

實務經驗

Python 3 (FastAPI, OpenCV) · MQTT / ZeroMQ / SIP / SNMP (軟硬整合通訊協定) · Arduino / C (硬體雛型) · Scrum / Kanban · PostgreSQL 16 · Astro / Svelte · Socket.io · Phaser · Pnpm monorepo · Playwright (E2E) · FTS5 (SQLite full-text search) · Fastify + WebSocket / SSE streaming · Gitea (self-hosted Git + Actions CI/CD) · DinD runner (自架 Gitea Actions CI 執行環境) · Kubernetes / k3s (自研架設 cluster，運行 Gitea + Argo CD) · GitOps (Gitea + Argo CD) · Grafana / Prometheus / Loki / Tempo · OpenTelemetry · @kevinsisi/ai-core (Multi-provider client + Gemini key pool) · Multi-Agent orchestration / deliberation · Prompt engineering (persona injection) · OpenAI API + OAuth PKCE · Google Gemini API · RDKit / CAD (Onshape)

曾接觸

Go (Gitea code-review action) · Blazor / Angular / MAUI · MySQL / MariaDB · Jenkins / SVN · MCP (Model Context Protocol) · Whisper / YOLOv8n / MediaPipe (media-processor AI stack) · MusicGen / ffmpeg / vidstab · Onshape (3D modeling) · CUDA (GPU worker scheduling)

學歷

銘傳大學 (Ming Chuan University) · 醫療資訊與管理學系 (學士學位)

2011 - 2015

Hello, I'm Kevin.