

從人治到 AI 協作 醫療資訊系統現代化的轉型路徑



國軍高雄總醫院 資訊室主任 鄭重男

1. MOPCON 2020, 2023–2025 Conference speaker
2. DevOpsDays 2024 Conference speaker
3. COSCUP 2024 Conference speaker
4. RubyWorld 2024 Conference (Japan Shimane Matsue) speaker
5. 台灣醫務管理學會e學院講師
6. 臺灣醫療品質學會講師
7. 經濟部IPAS AI應用規劃師
8. TAISE氣候與健康管理師
9. 國立高雄科技大學資管所碩士

1150203@台中 by Nickle Cheng

ABOUT ME

I'm on Rails. To make me smile:) and on the way toward the master of omakase DevSecOps also To be an evangelist for Hospitals

INTRODUCTION

- 開放原始碼的狂熱份子，喜歡分享及解決痛點。
- 想以程式設計師及醫護人員的快樂為優先。
- 想當醫院的 Ruby on Rails & DevOps 傳教士。
- 想找到第一位 Rails Doctor。

RAILS DOCTRINE

1. Optimize for programmer happiness
2. Convention over Configuration
3. The menu is omakase
4. No one paradigm
5. Exalt beautiful code
6. Provide sharp knives
7. Value integrated systems
8. Progress over stability
9. Push up a big tent



NICKLE 鄭重男 Cheng Chung Nan



CONTACT ME

- 0906728091
- nickle@mail.802.org.tw
- 高雄市苓雅區中正一路2號

REFERENCES



RubyWorld 2024

802 保力賀

Agenda

1. 人治開發
2. XDD(SDD, TDD, BDD, DDD)
3. SMART on FHIR 徵案
4. 擁抱 AI
5. 給轉型期團隊的建議

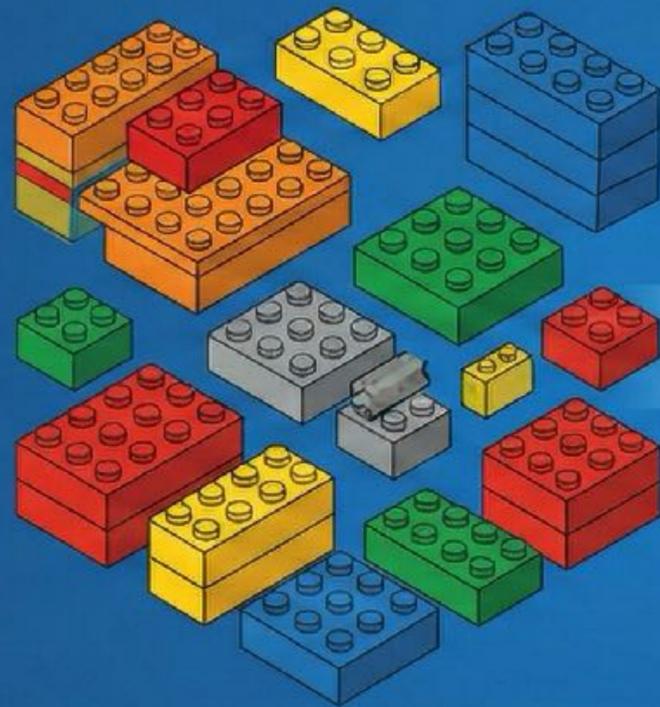


資訊系統碎片化

碎片化與疊床架屋



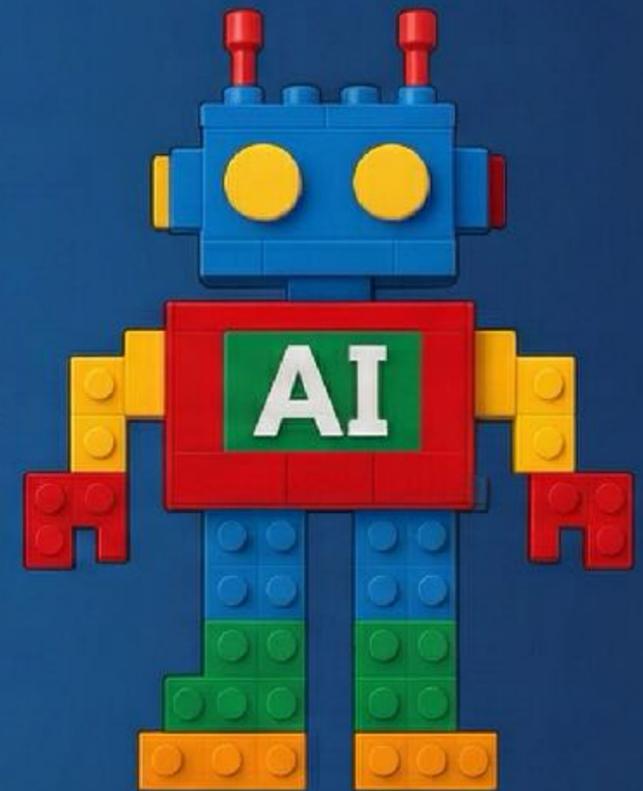
模組化



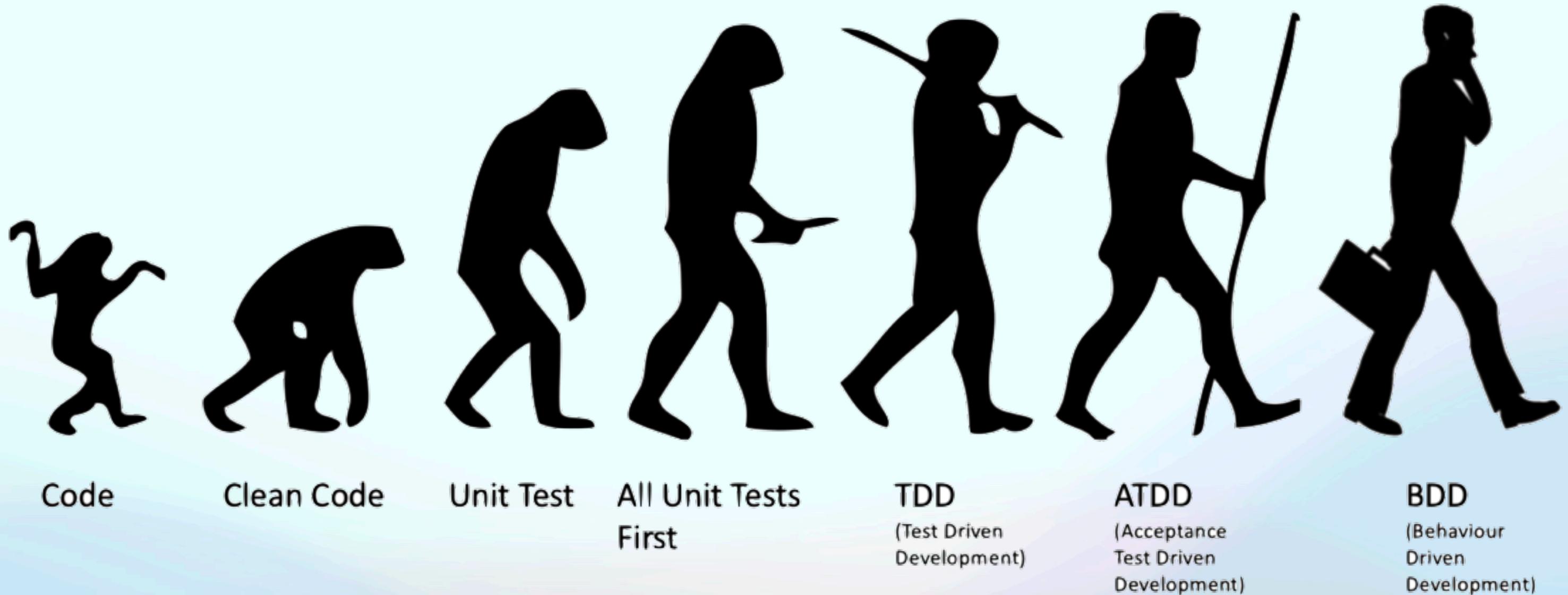
SMART on FHIR



AI 積木



Evolution of Software Development

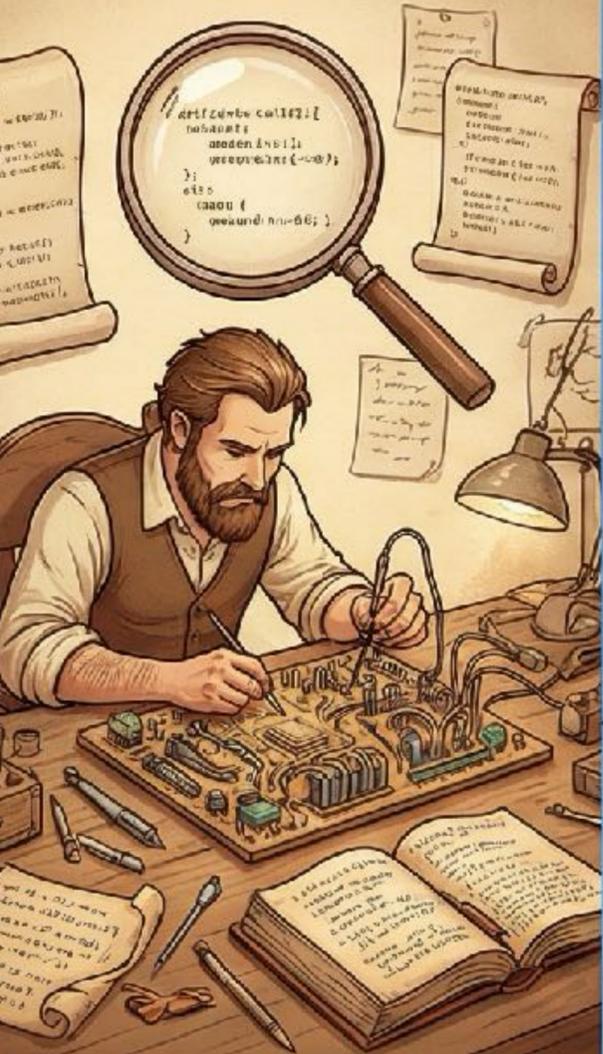


source: <https://www.linkedin.com/pulse/evolution-software-development-balaji-kalyansundaram>



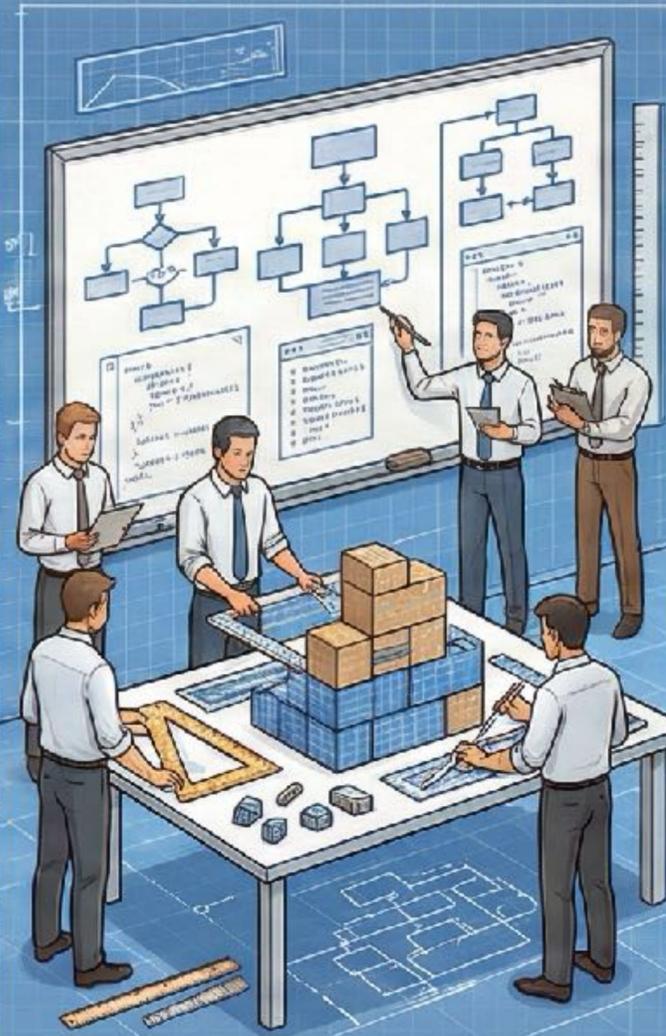
手工

(Handcrafted/
Individual Hero Era)



結構化開發

(Structured Programming)



瀑布式

(Waterfall Model)



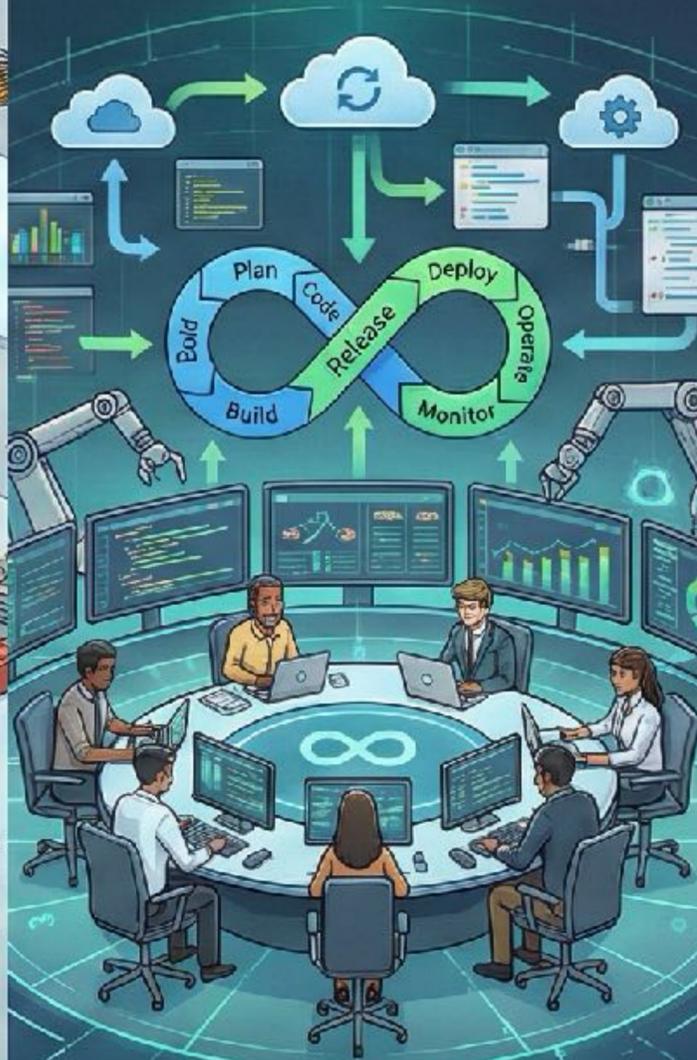
物件導向 & Framework

(OOP/Framework Era)



敏捷 & DevOps

(Agile/Lean &
DevOps/Automation)



人治時期的開發階段

時間演進與自動化程度提升



SDD->TDD->XDD



Spec-Driven Development (SDD)

優點

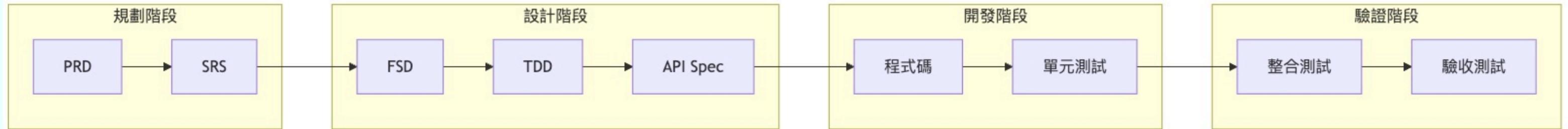
- 開發前有明確共識
- 減少後期變更成本
- 便於團隊分工與溝通
- 可作為驗收依據

缺點

- 前期投入時間長
- 規格可能與實際脫節
- 不適合需求頻繁變動的專案
- 文件維護成本高



Spec-Driven Development (SDD)

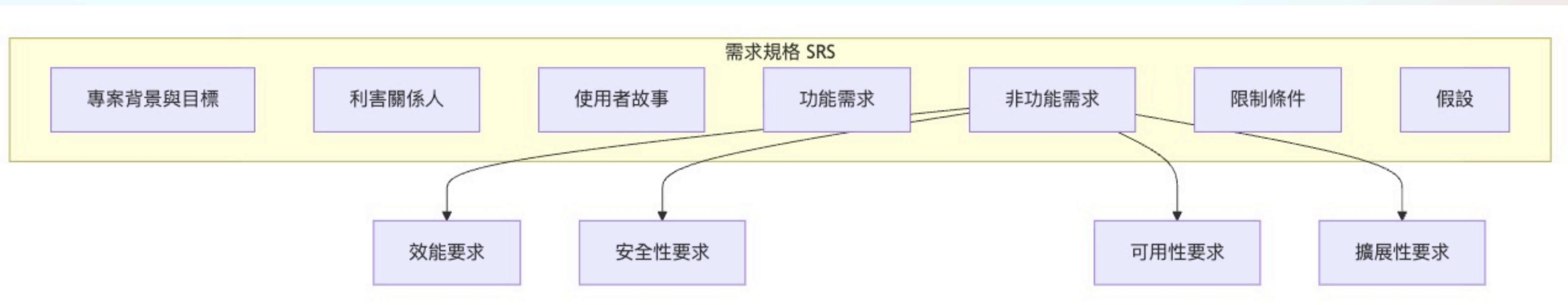


文件類型說明

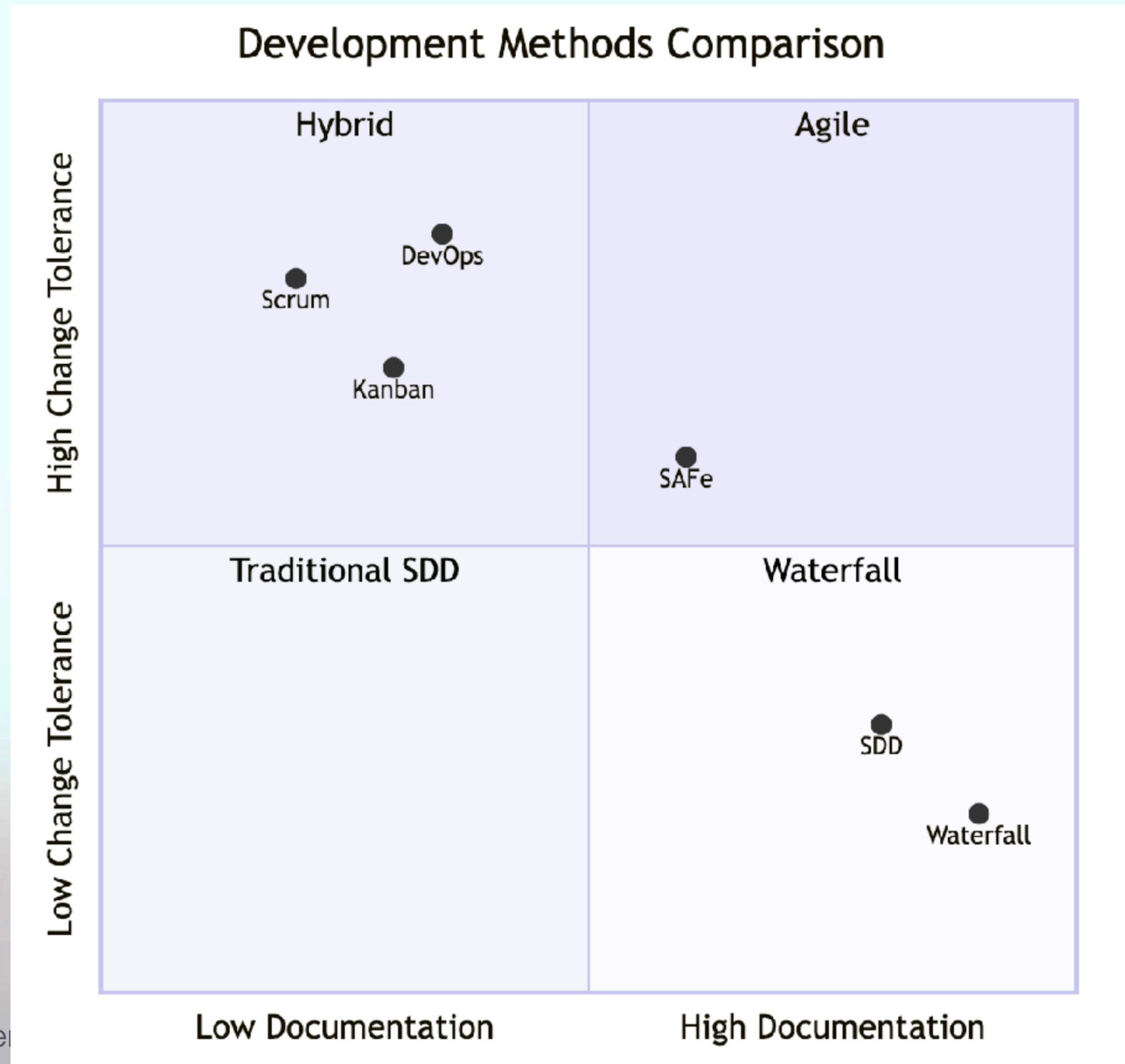
文件	全名	用途
PRD	Product Requirements Document	產品需求文件
SRS	Software Requirements Specification	軟體需求規格
FSD	Functional Specification Document	功能規格文件
TDD	Technical Design Document	技術設計文件
API Spec	API Specification	API 規格文件

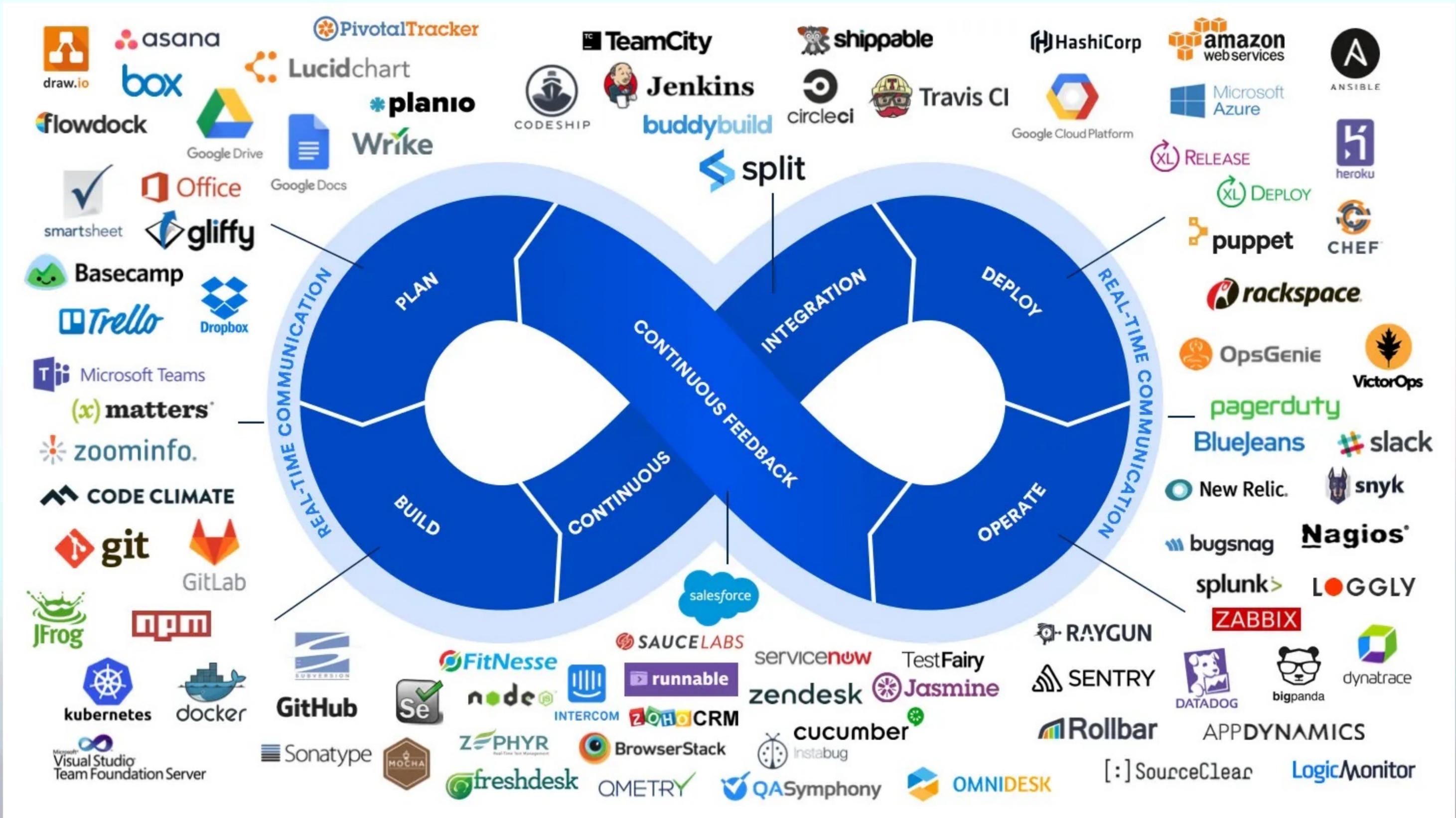
Spec-Driven Development (SDD)

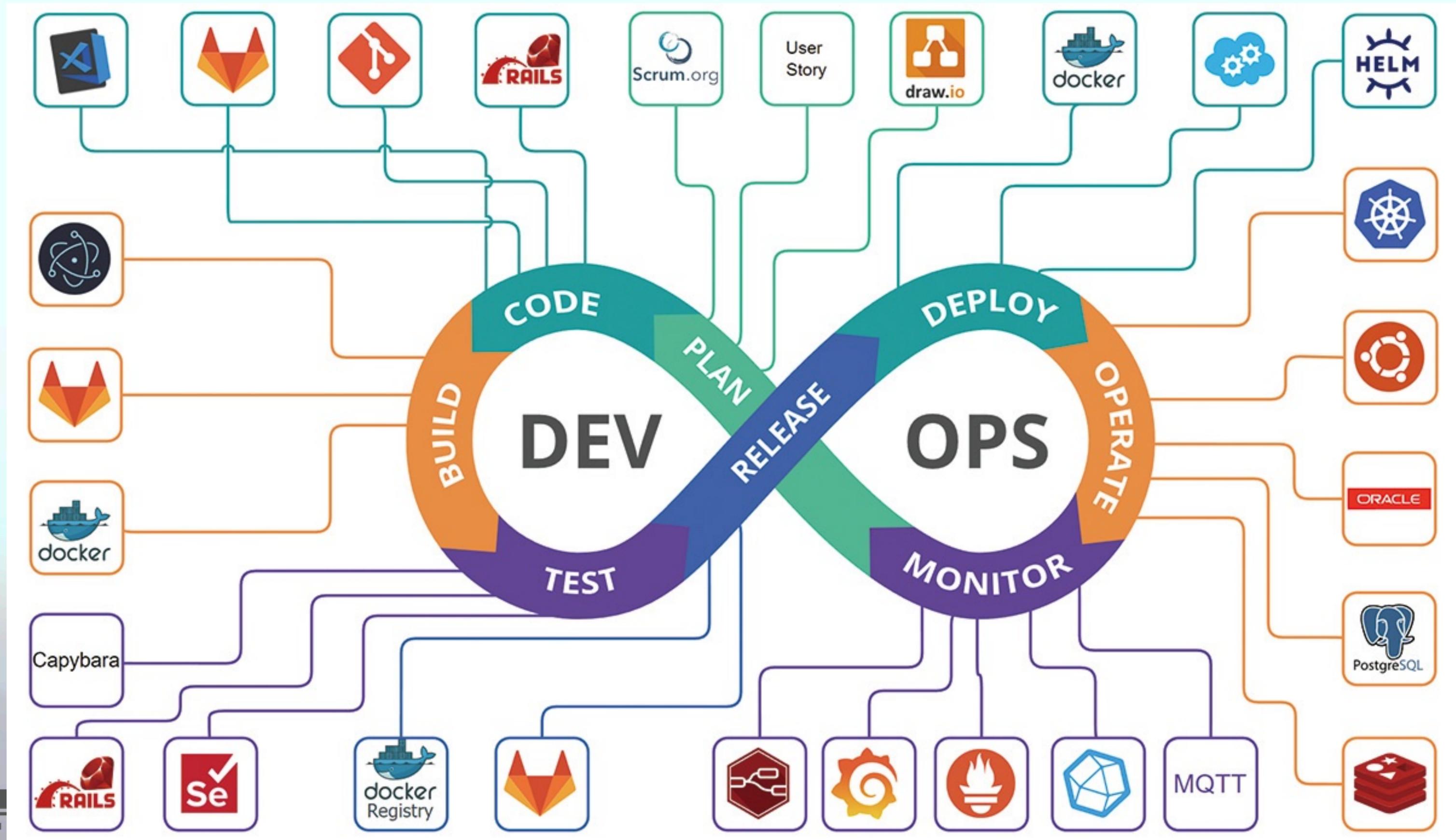
需求規格內容

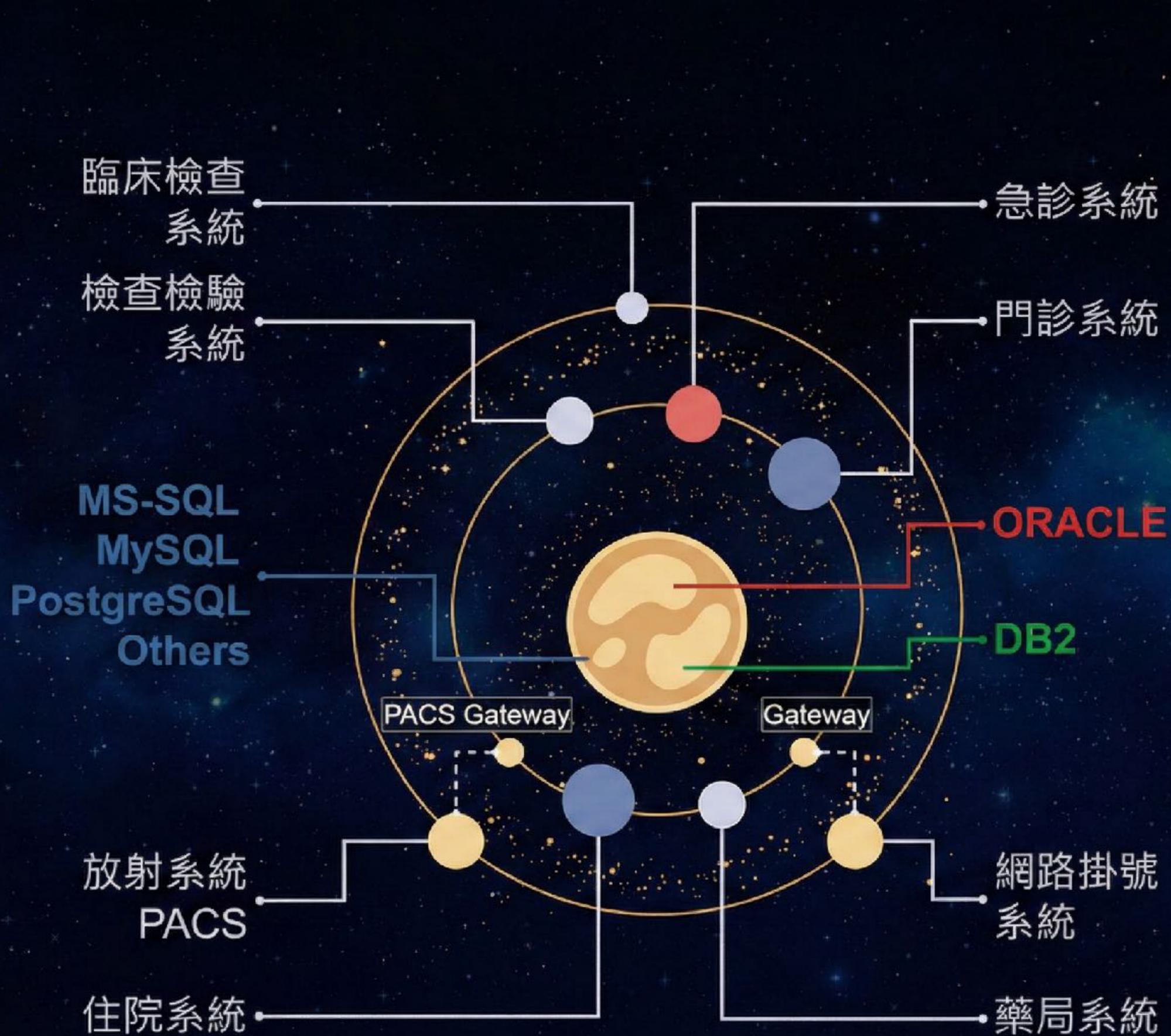


Spec-Driven Development (SDD)









醫療資訊系統1.0

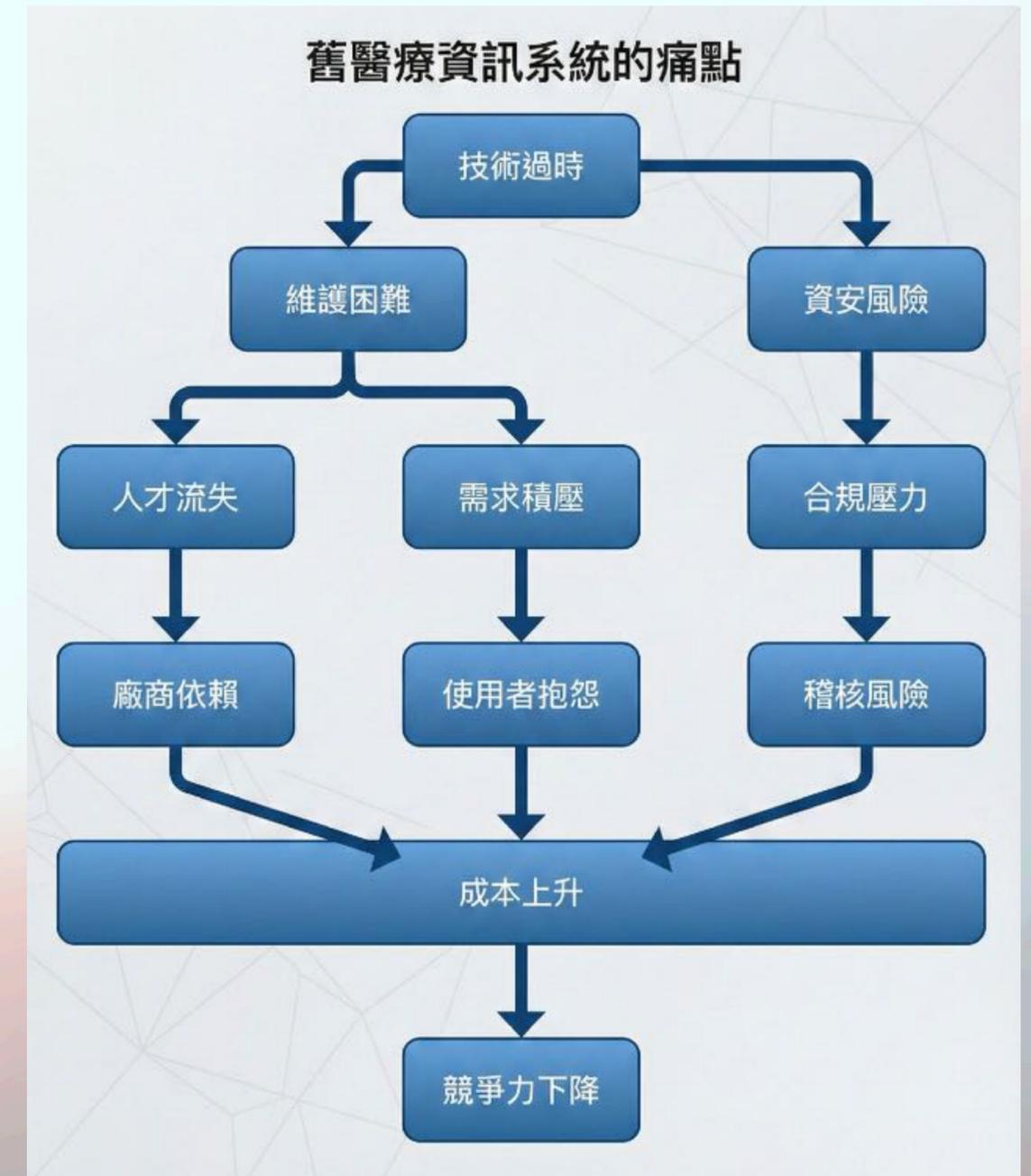
程式語言：Delphi

架構：client-server

資料庫：Oracle

年齡：超過20年

痛點	說明
語言過時	Delphi 開發者難找，社群萎縮，學習資源少
元件停更	第三方元件停止維護，有安全漏洞無法修補
無法升級	升級 Delphi 版本可能導致大量程式碼不相容
緊耦合	商業邏輯散落在 Form 事件中，難以測試與重用
無版控	程式碼用資料夾管理，無法追蹤變更歷史
缺乏測試	沒有自動化測試，改 A 壞 B



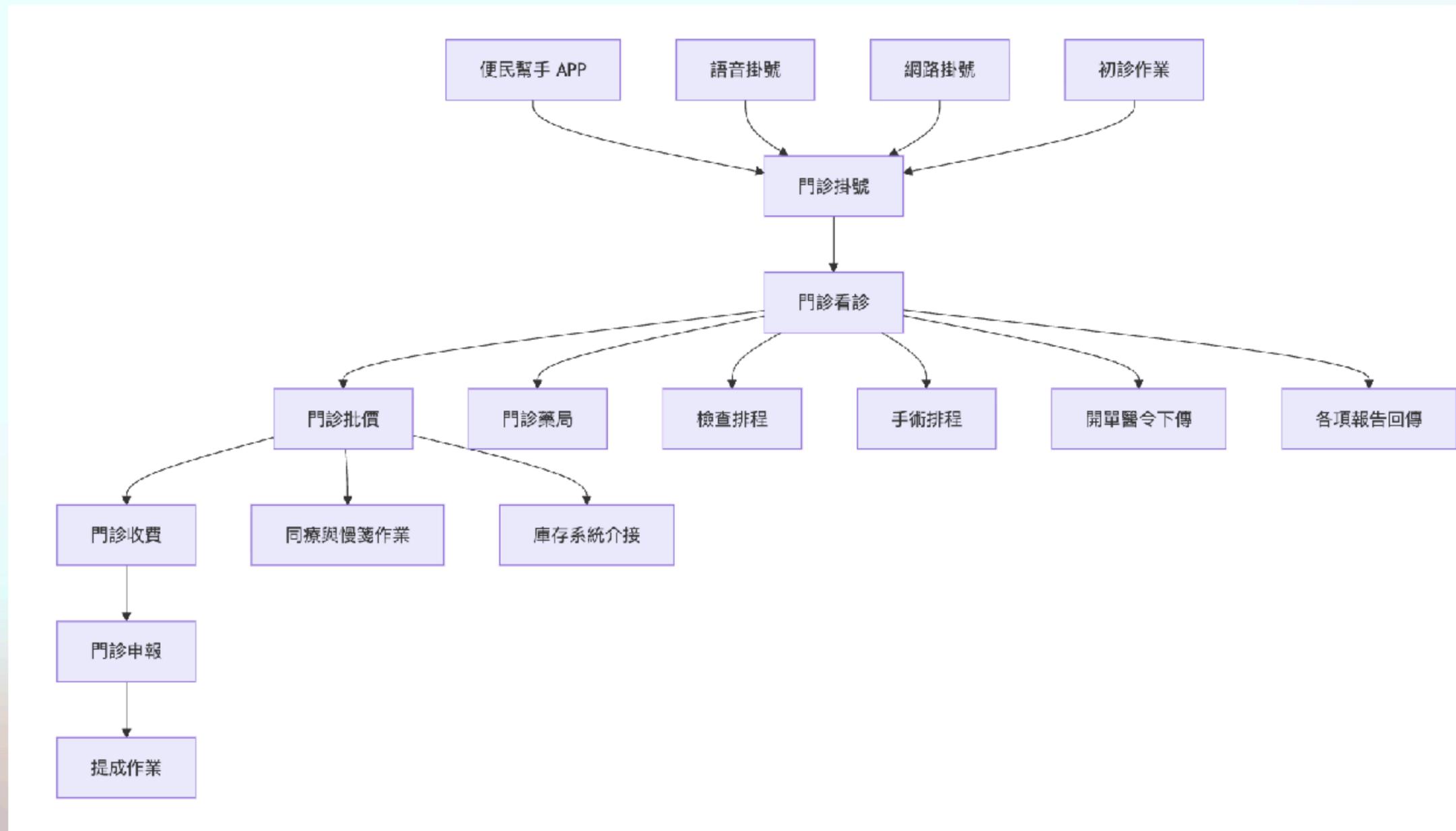
醫療資訊系統2.0

程式語言：C#

架構：.net frame winform

資料庫：Oracle

年齡：6年



醫療資訊系統 1.0 VS 2.0

面向	HIS 1.0 (Delphi C/S)	HIS 2.0 (C# Web)
部署方式	逐台安裝	集中部署
更新速度	緩慢	即時
行動支援	無	有
擴展性	垂直擴展	水平擴展
系統整合	困難	標準 API
開發人才	稀缺	充足
離線使用	支援	需額外設計
硬體整合	直接	需代理程式
資安風險	內網為主	攻擊面較大
學習成本	低 (熟悉)	高 (新技術)





- 1 [Optimize for programmer happiness](#)
- 2 [Convention over Configuration](#)
- 3 [The menu is omakase](#)
- 4 [No one paradigm](#)
- 5 [Exalt beautiful code](#)
- 6 [Provide sharp knives](#)
- 7 [Value integrated systems](#)
- 8 [Progress over stability.](#)
- 9 [Push up a big tent](#)

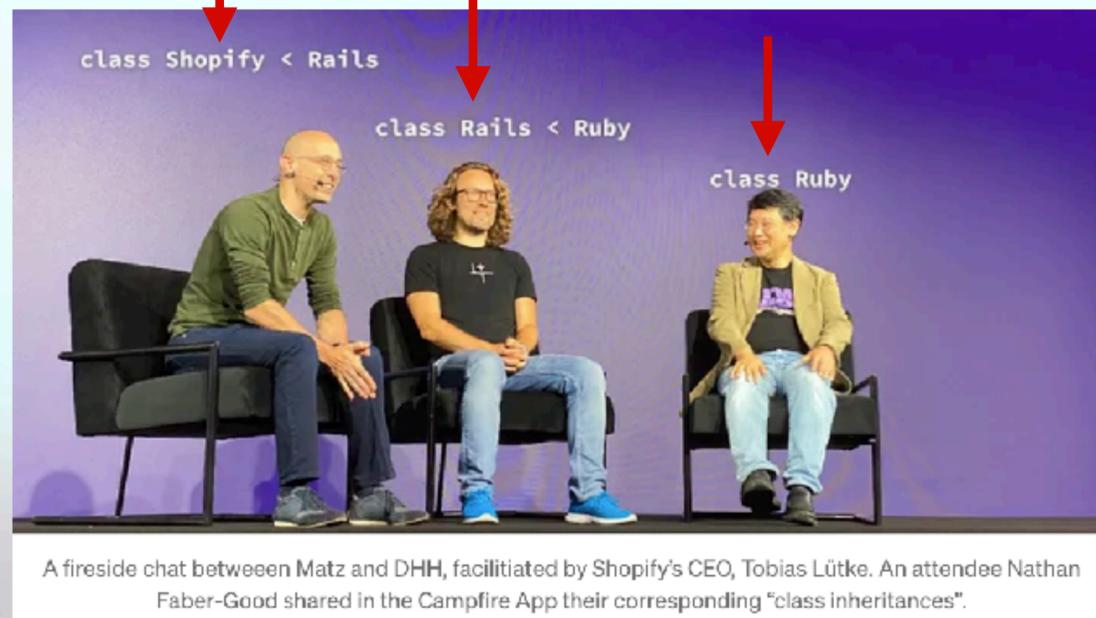
The Rails Doctrine.

David Heinemeier Hansson

德國人

丹麥人

日本人



- 1 程式設計師的快樂優先
- 2 約定優於配置
- 3 主廚精選
- 4 多元範式
- 5 讚揚優美
- 6 提供實用工具
- 7 重視整合系統
- 8 進步比穩定更重要
- 9 包容並重



class Medical < Rails < Ruby

Covid-19 專責病房動態看板

PCR 陽性報告 8 天後可轉出專責病房

49  29/11/9

PCR 陽性住院人數

39  27/4/8

超過 8 天人數

2 

等待轉入專責病房人數

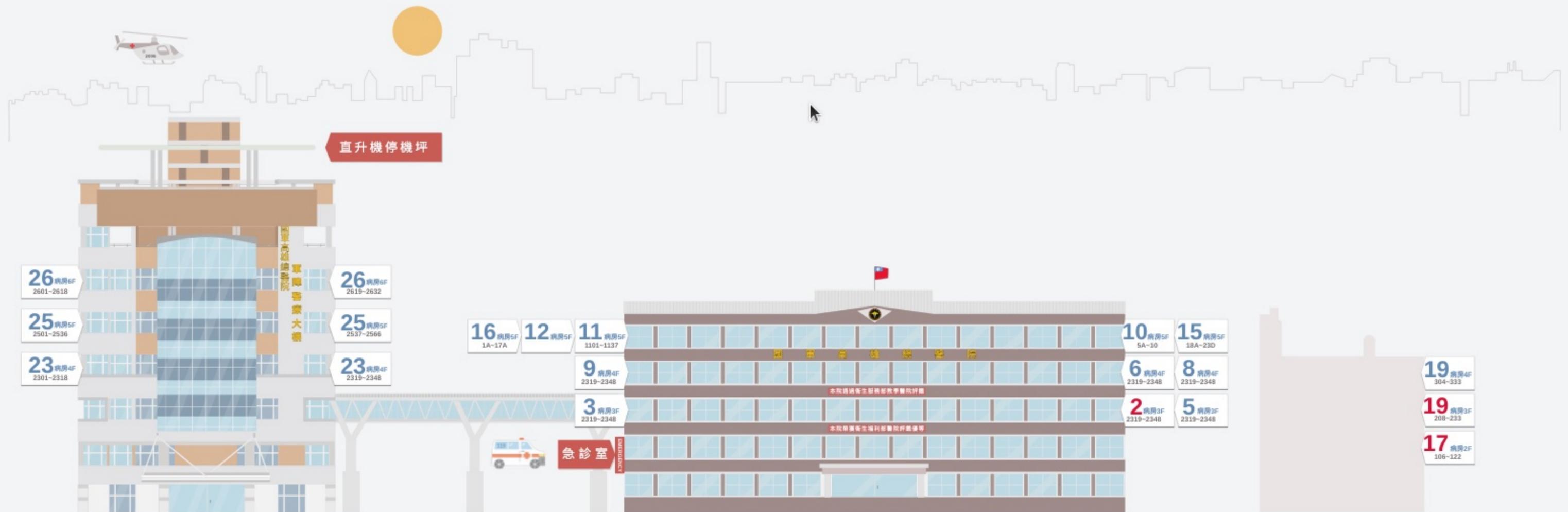
4 

應轉出專責病房人數

建議使用 Chrome 或 Safari 瀏覽器，請勿使用舊版 IE

更新時間：2022-07-22 10:30:03 (每 10 分鐘自動更新)

手動更新 



SMART on FHIR 1

火線超人 FHIRLineBot



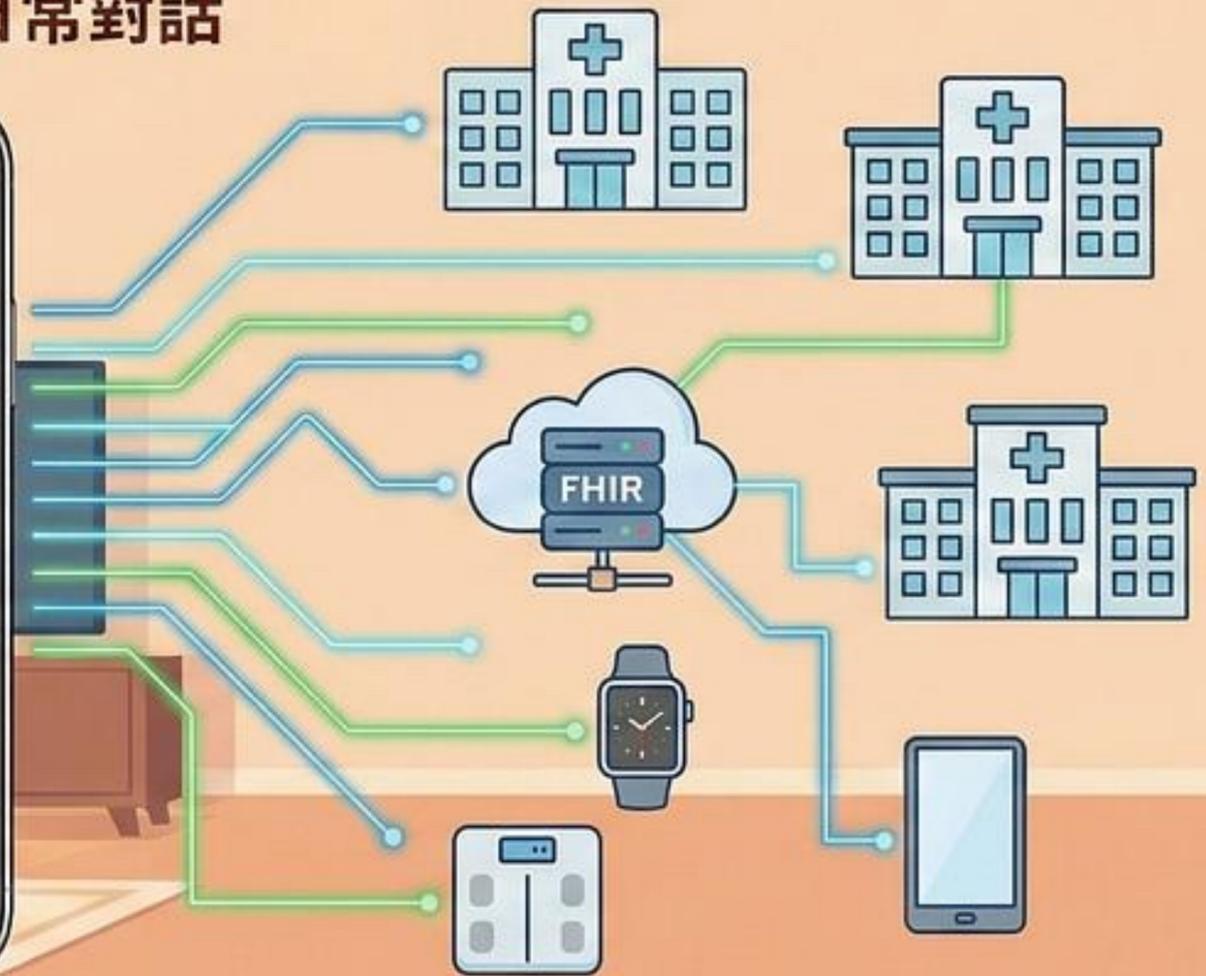
火線超人 FHIR LINE Bot

讓健康資料走進你的日常對話



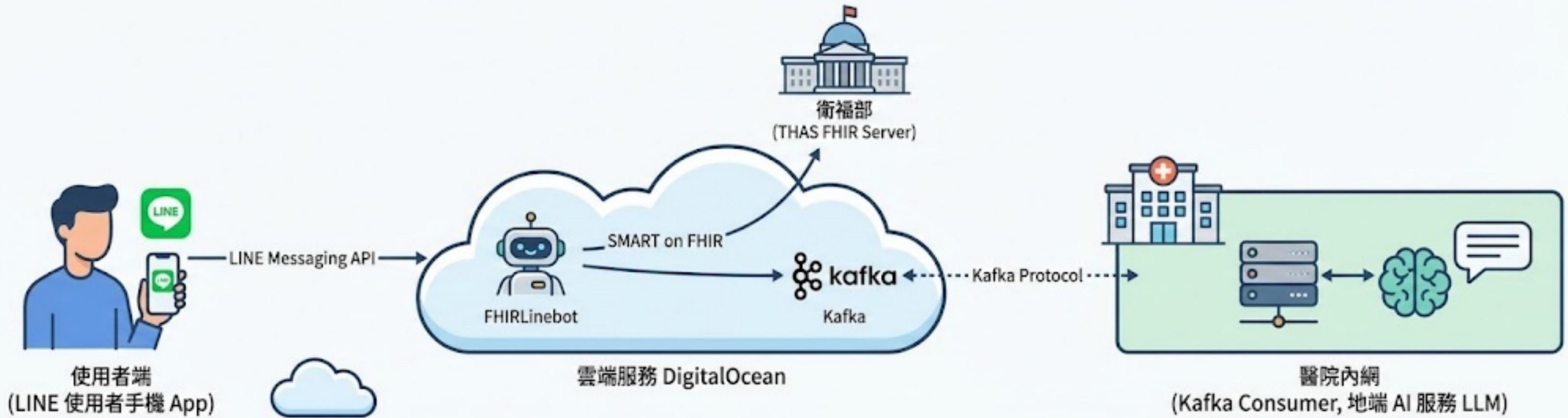
加好友

FHIRLineBot



<https://nicklecheng.turbos.tw/>

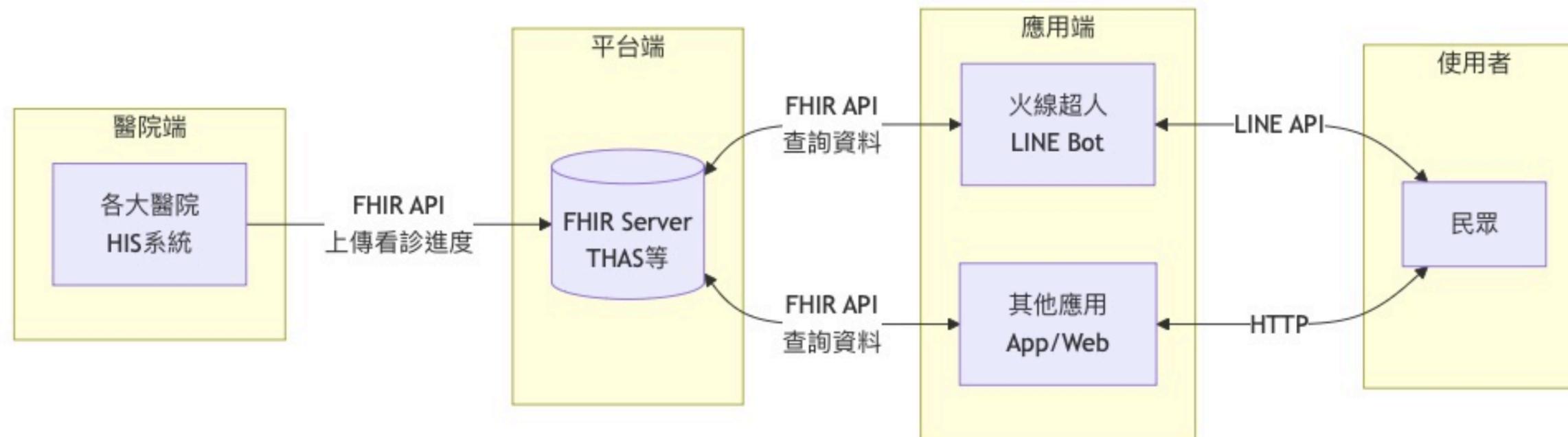
SMART on FHIR 1 – 火線超人



<https://nicklecheng.turbos.tw/series/smart-on-fhir/fhir-line-bot>

SMART on FHIR 1 – 火線超人

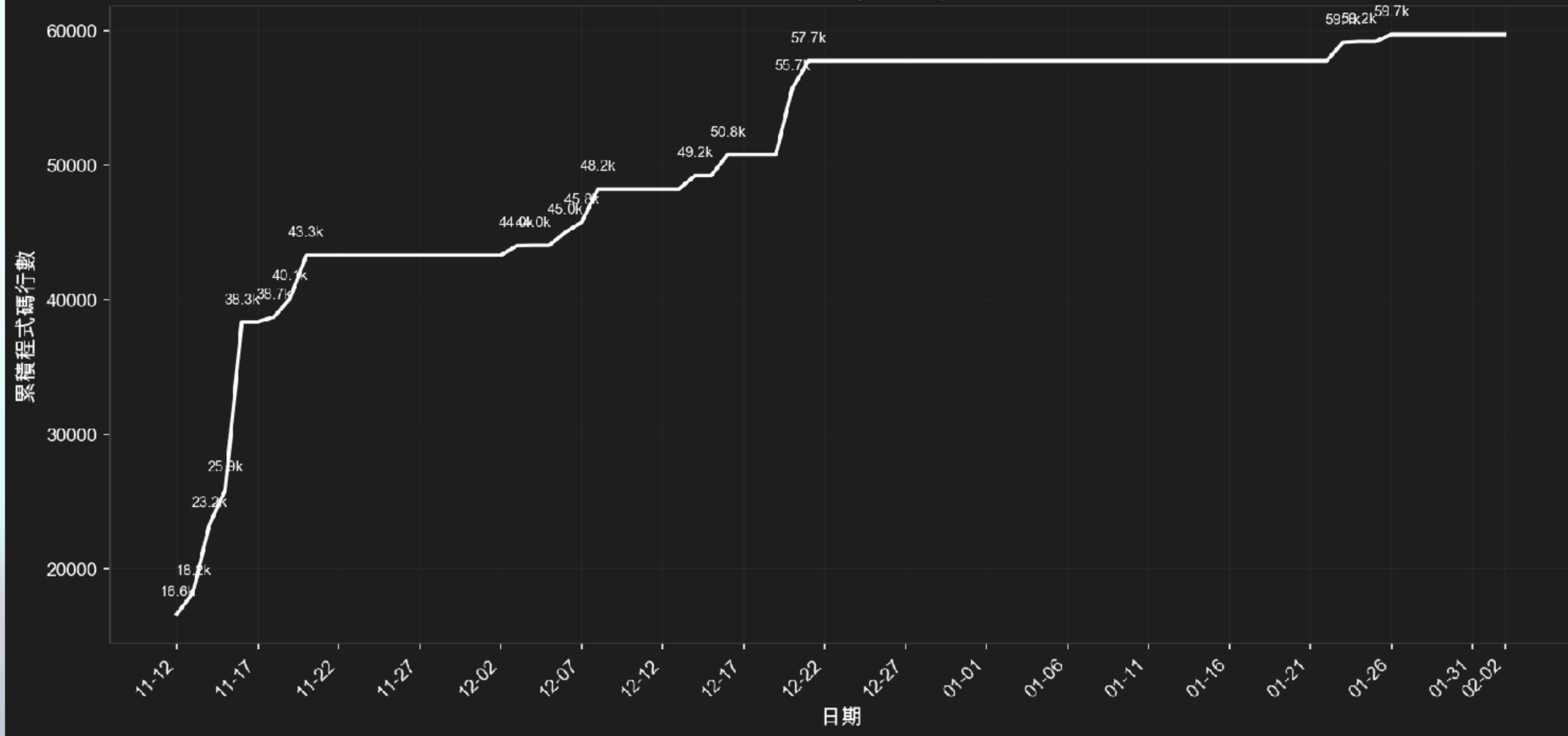
建立醫院看診進度開放生態系



目標：

1. **醫院端** - 鼓勵各醫院透過 FHIR 標準上傳即時看診進度至 FHIR Server
2. **平台端** - 建立統一的看診進度查詢平台，整合各院資料
3. **民眾端** - 透過 LINE 高滲透率（台灣 2,100 萬用戶）觸及更多民眾

fhirlinebot - 程式碼成長趨勢 (日統計) - 累積程式碼行數



SMART on FHIR 2

衛福部救護車資訊一站通



SMART on FHIR 3

癌症用藥事前審查



SMART on FHIR 3 癌藥事審



TWPAS

藥品事前審查申請系統



如何啟動此應用程式

此系統需要透過 SMART on FHIR Launch 流程啟動。

衛福部 THAS Sandbox :

EHR Launch (登入, 選病人)

Provider Standalone Launch (登入, 選病人)

Provider Standalone Launch (登入, 不選病人)

TWPAS 直接啟動 :

TWPAS Launch (不登入, 不選病人)

SMART Health IT Sandbox :

EHR Launch (登入, 選病人)

Provider Standalone Launch (登入, 選病人)

Provider Standalone Launch (登入, 不選病人)

Taiwan Prior Authorization System (TWPAS)

使用 SMART on FHIR 標準整合醫療資訊系統

藥品事前審查申請表單

FHIR Server 資源 (已自動載入)

1 Patient

1. 院所資訊

醫事機構	申請類別
輸入代碼或名稱搜尋(例如: 0502080015、國軍高雄總醫院附設民眾診療服務處)	請選擇
就醫科別	申請醫師身分證號
輸入代碼或名稱搜尋(例如: 03、外科)	例如: A123456789
申請日期	緊急報備日期
2026/02/02	年 / 月 / 日
申請案件類別	原受理編號
請選擇	例如: 1130001234

2. 病人資訊

姓名	身分證號
門診病患	A123456789
病歷號	生日
123456	1997/01/01
性別	體重 (kg)
男性	請輸入體重
身高 (cm)	是否懷孕或哺乳
請輸入身高	請選擇

3. 疾病診斷

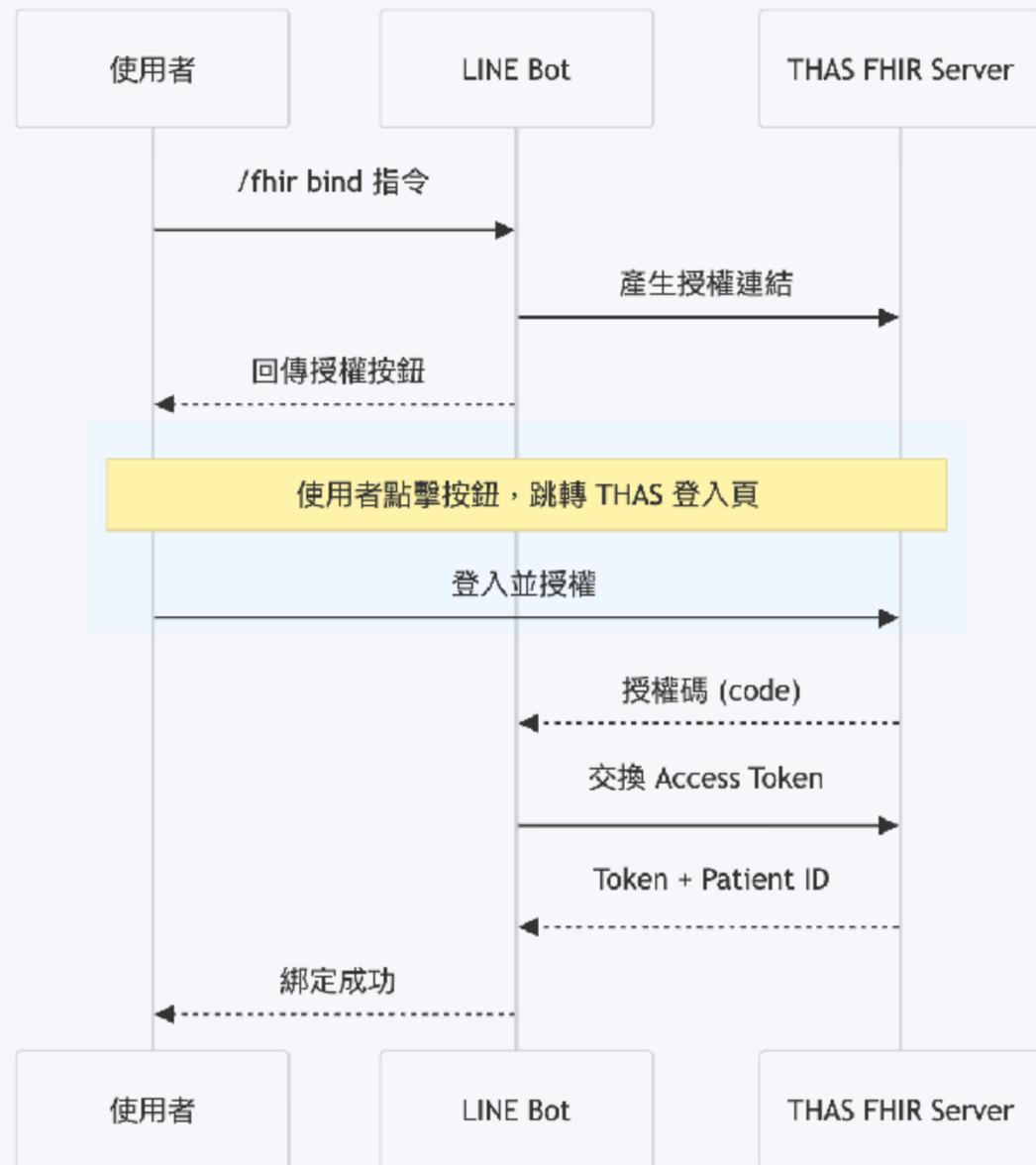
國際疾病分類
輸入代碼或名稱搜尋(例如: C00.9、唇惡性腫瘤、Malignant neoplasm of lip, unspecified)
診斷日期
年 / 月 / 日
簡要病摘(申請原因)
請簡述病史、診斷及申請此藥品的原因

藥品事前審查申請列表

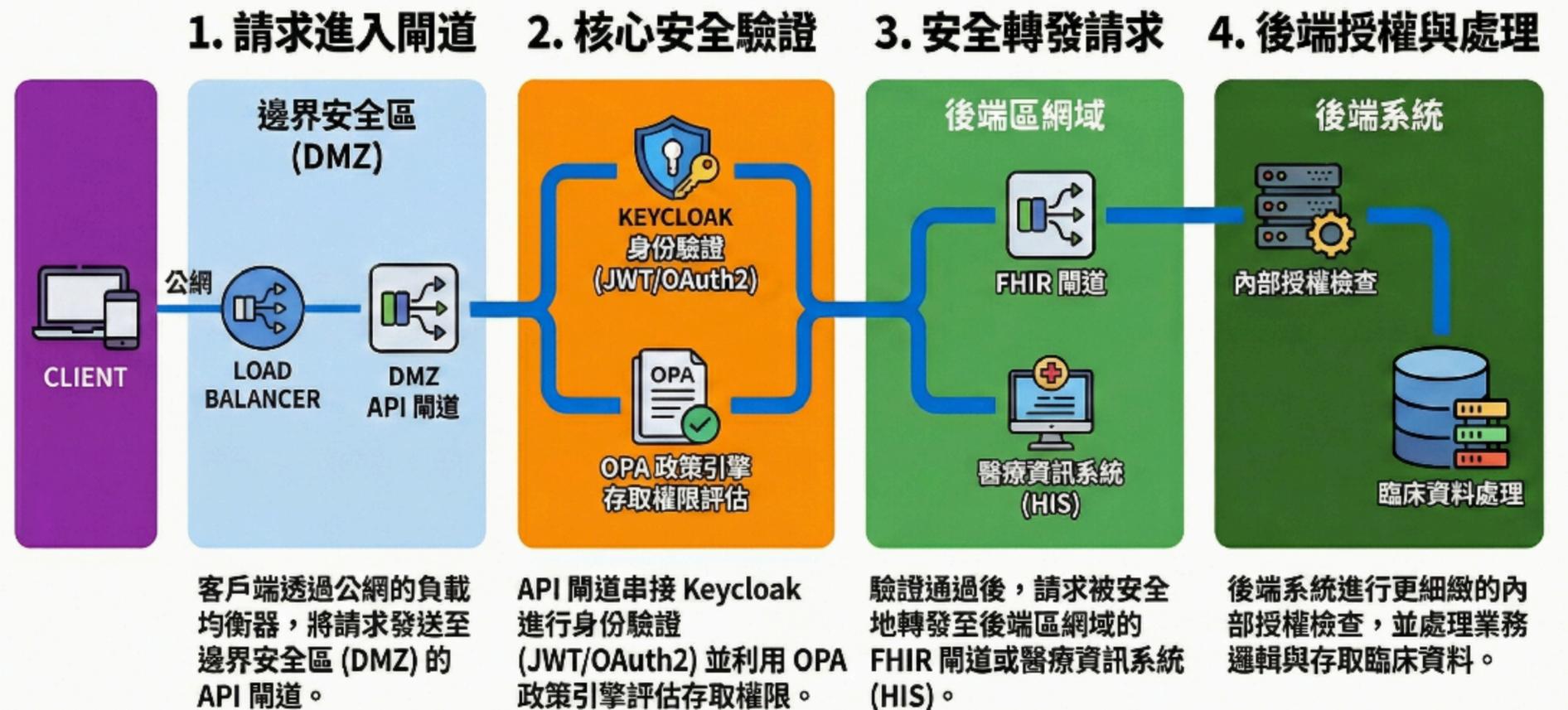
建立新申請

全部	草稿	待審核	審核中	通過	未通過	
編號	病人姓名	狀態	申請日期	藥品數	操作	
#12	門診病患	草稿	2026-02-02	1項	查看 編輯 刪除	
#11	王病人	審核中	2026-02-02	1項	查看	
#10	王病人	待審核	2026-02-02	1項	查看 編輯 產生 FHIR 送審 刪除	
#9	王病人	待審核	2026-01-30	1項	查看 編輯 產生 FHIR 送審 刪除	
#8	王病人	審核中	2026-01-30	1項	查看	
#7	王病人	審核中	2026-01-30	1項	查看	
#6	王病人	審核中	2026-01-30	1項	查看	
#5	王病人	審核中	2026-01-30	1項	查看	
#4	陳大明	草稿	2026-01-29	1項	查看 編輯 刪除	
#3	ICU	草稿	2026-01-29	1項	查看 編輯 刪除	
#2	王大明	待審核	2026-01-29	1項	查看 編輯 產生 FHIR 送審 刪除	
#1	王大明	草稿	2026-01-29	1項	查看 編輯 刪除	

Oauth2 & Security



FHIR 資訊安全流程



模組化積木



AI 夥伴程式設計：積木化開發架構

開發者介面與客戶端



開發者工具

開發者透過瀏覽器或整合開發環境 (IDE) 與 AI 進行互動。

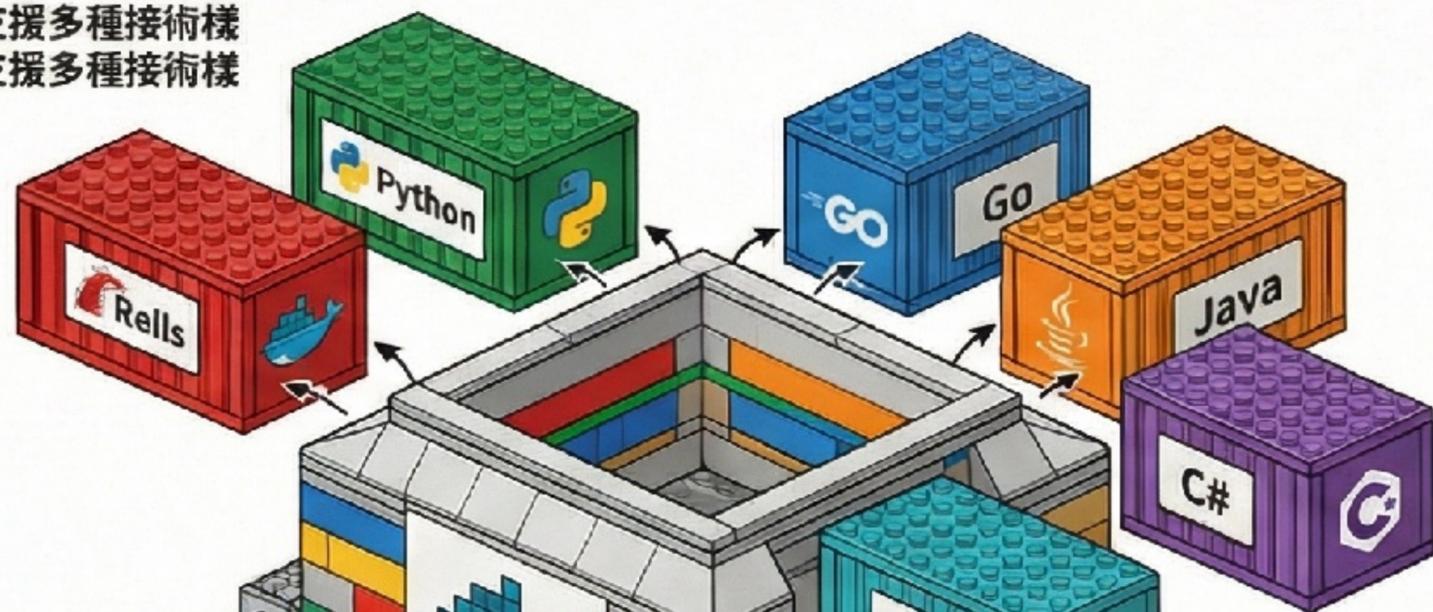


客戶端應用

終端使用者透過手機 App 或網頁與最終產品互動。

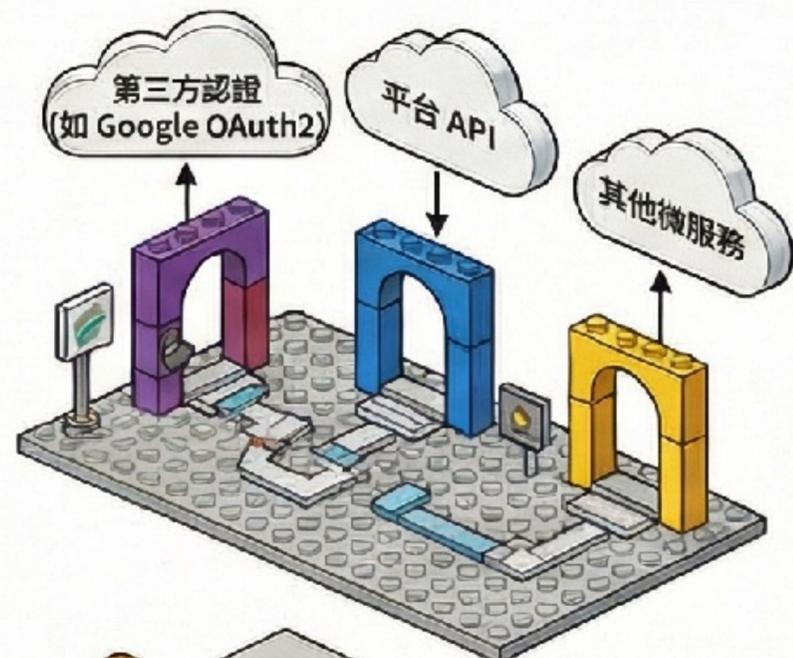
可抽換的語言容器

支援多種接術樣
支援多種接術樣



AI 程式設計引擎

基礎架構與支援服務



外部服務整合

串接第三方認證 (如 Google OAuth2)、平台 API 與其他微服務。

網頁伺服器與應用程式

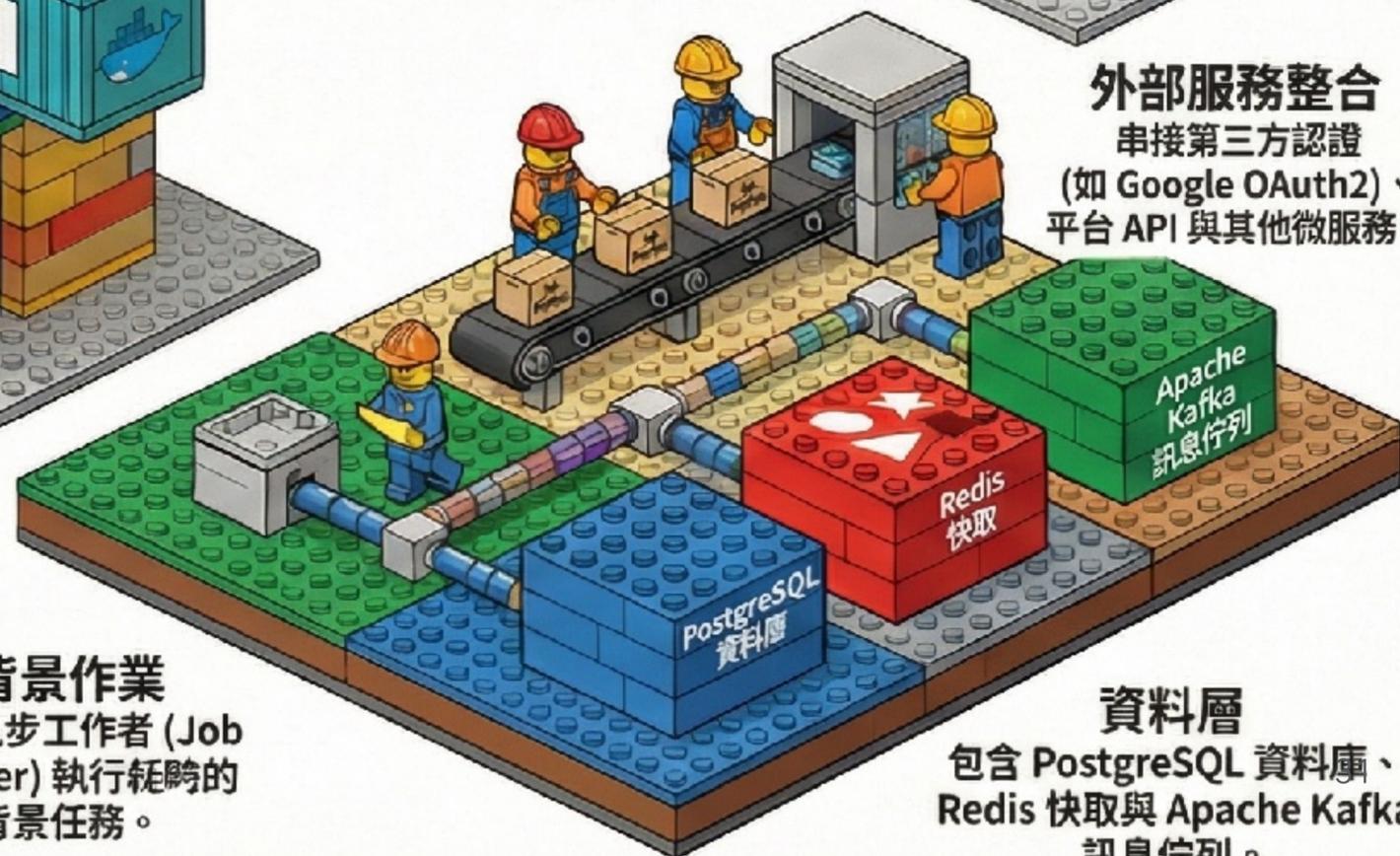
處理請求、執行業務邏輯，並與後端服務溝通

背景作業

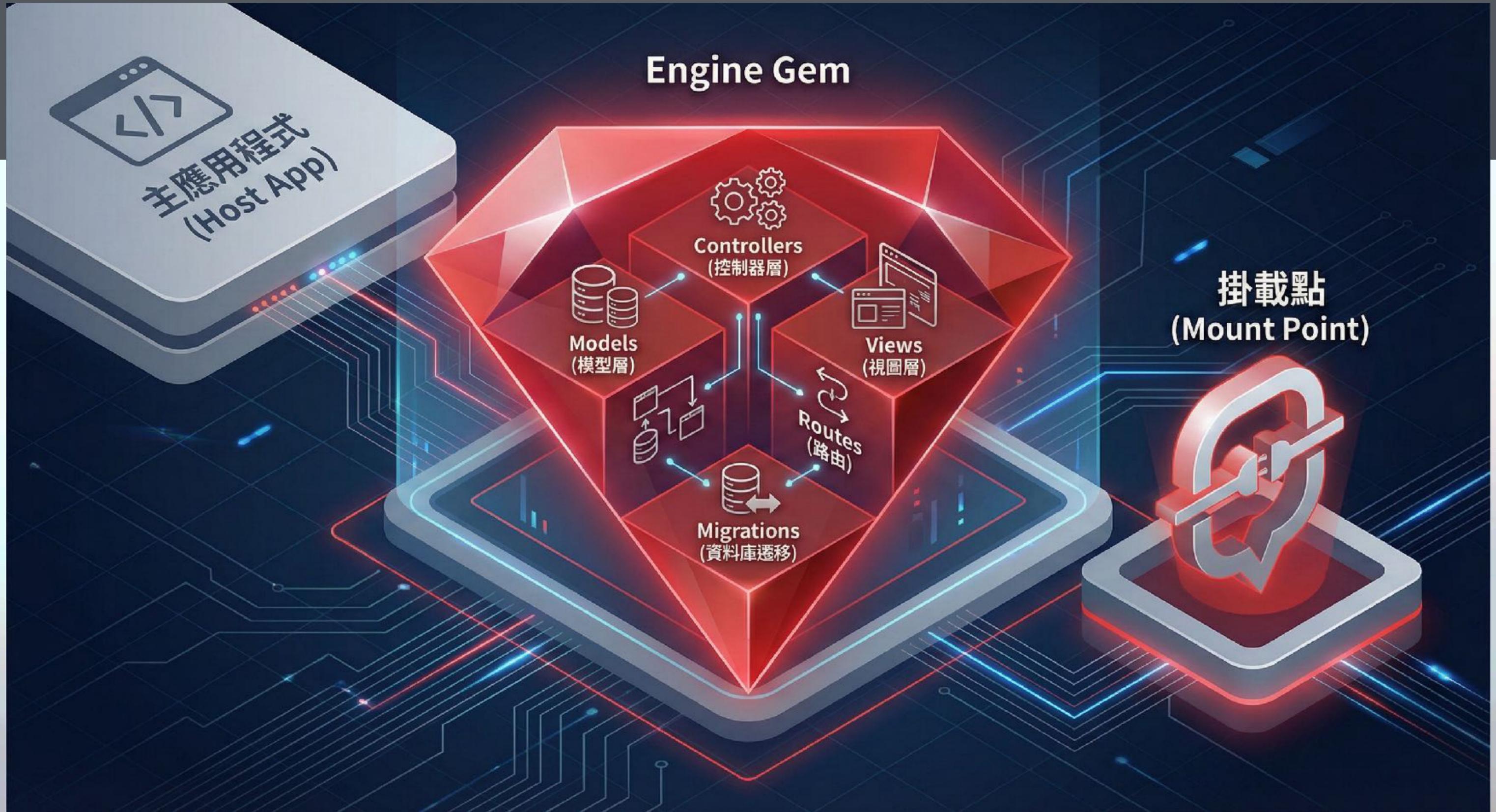
透過見步工作者 (Job Worker) 執行耗時的背景任務。

資料層

包含 PostgreSQL 資料庫、Redis 快取與 Apache Kafka 訊息佇列。



Engine Gem



擁抱AI



Vibe Coding

● 靠感覺寫程式
● 不照規劃
● 不跑測試
● 不查文件
● 只要我感覺對，它就是對的

– ChatGPT



Beyond Vibe Coding

1. 團隊合作專案：容易產生**技術債**，難以維護
2. 需要穩定安全的系統(**醫療**、**金融**)：錯誤風險高
3. 需要交付正式產品或長期維護：沒有規劃的程式碼

會變成**未爆彈**



Gemini CLI

```
> npm install -g @google/gemini-cli
```

```
added 583 packages in 10s
```

```
148 packages are looking for funding  
run `npm fund` for details
```



✓ 10s ⌘ 13:28:59 ⌚

You are running Gemini CLI in your home directory. It is recommended to run in a project-specific directory.

Get started

How would you like to authenticate for this project?

- 1. Login with Google
- 2. Use Gemini API Key
- 3. Vertex AI

No authentication method selected.

(Use Enter to select)

Terms of Services and Privacy Notice for Gemini CLI

<https://github.com/google-gemini/gemini-cli/blob/main/docs/tos-privacy.md>

GEMINI

Tips for getting started:

1. Ask questions, edit files, or run commands.
2. Be specific for the best results.
3. Create **GEMINI.md** files to customize your interactions with Gemini.

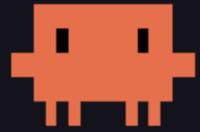
You are running Gemini CLI in your home directory. It is recommended to run in a project-specific directory.

> Type your message or @path/to/file



Claude Code

>> claude



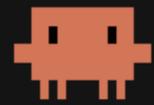
Claude Code v2.0.31
Sonnet 4.5 · Claude Max
/Users/nickle/projects/medicores/medicore

> /

```
/bmad          BMad Master (project)
/add-dir       Add a new working directory
/agents        Manage agent configurations
/bashes        List and manage background tasks
/bmad          BMad Master (project, gitignored)
/clear (reset, new) Clear conversation history and free up context
/compact       Clear conversation history but keep a summary in context. Optional: /compact [instructions for
summarization]
/config (theme) Open config panel
/context        Visualize current context usage as a colored grid
/cost          Show the total cost and duration of the current session
```



Claude Code /BMad



Claude Code v2.0.31
Sonnet 4.5 · Claude Max
/Users/nickle/projects/medicores/medicore

> /bmad is running..

- 🤖 BMad Master Task Executor activated and ready.

I'm your universal executor for all BMad-Method capabilities. I can run any task, generate documents from templates, execute checklists, and more.

Type `*help` to see available commands, or just tell me what you need help with.

> *help

- Available Commands

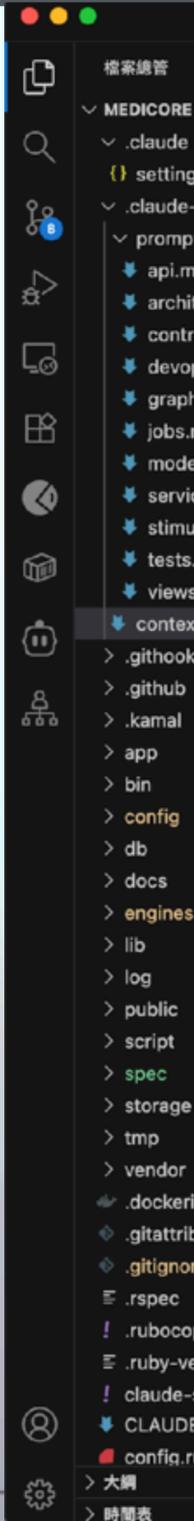
1. `help` - Show these listed commands in a numbered list
2. `kb` - Toggle KB mode off (default) or on, when on will load and reference the knowledge base and converse with you
3. `task {task}` - Execute task, if not found or none specified, list available tasks
4. `create-doc {template}` - Execute task create-doc (no template = show available templates)
5. `doc-out` - Output full document to current destination file
6. `document-project` - Execute the document-project.md task
7. `execute-checklist {checklist}` - Run task execute-checklist (no checklist = show available checklists)
8. `shard-doc {document} {destination}` - Run the task shard-doc against the provided document to the specified destination
9. `yolo` - Toggle Yolo Mode
10. `exit` - Exit (confirm)

—
Note: All commands require the `*` prefix when used (e.g., `*task`, `*create-doc`, `*kb`)

What would you like to do?



ClaudeOnRails



Rails Models Specialist

You are an ActiveF expertise covers:

Core Res

1. Model Design
2. Associations
3. Migrations: V
4. Query Optim
5. Database De

Rails Mod

Validations

- Use built-in v
- Create custor
- Consider data

Associator

- Use appropri
- Consider :dep
- Implement co
- Use :inverse_

Scopes and

- Create named
- Avoid N+1 qu
- Use database

Rails API Specialist

You are a Rails API spec RESTful API design, ser

Core Respon

Rails GraphQL Specialist

You are a Rails GraphQL:

Rails Models Specialist

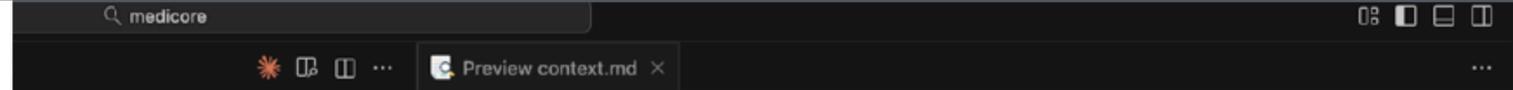
Test-Driven Development Example

Building features test-first:

```
> Create a user registration system with email verification - write tests first
```

The swarm will:

1. Tests agent writes comprehensive specs
2. Models agent creates User model to pass tests
3. Controllers agent implements registration flow
4. Services agent handles email verification logic
5. Tests agent ensures 100% coverage



ClaudeOnRails Context

ails with a swarm of specialized agents for Rails development.

```
render json: {
```

```
end  
end  
end
```

Scopes and Que

- Create named scope

```
describe '#fi  
let(:user)  
  
it 'return:  
expect(u
```

Layouts and Partials

```
<!-- app/views/layouts/application.html.erb -->
```

ClaudeOnRails

```
o > claude-swarm
Starting Claude Swarm from /Users/nickle/projects/medicores/medicore/claude-swarm.yml...
🚀 Starting Claude Swarm: Medicore Rails Development Team

📁 Session files will be saved to: /Users/nickle/.claude-swarm/sessions/Users+nickle+projects+medicores+medicore/6d0f9449-61e5-4fd0-9025-d4266bb433c3/

✓ Generated MCP configurations in session directory

✓ Generated settings files with hooks

🚀 Launching main instance: architect
Model: opus
Directory: /Users/nickle/projects/medicores/medicore
Connections: models, controllers, views, stimulus, jobs, tests, devops
😎 Vibe mode ON for this instance

 Claude Code v2.0.31
Opus 4.1 · Claude Max
/Users/nickle/projects/medicores/medicore
```

🚩 Claude Swarm Summary

```
=====  
Runtime: 22h 23m 50s  
Total Cost: $0.0000 (excluding main instance)  
Session: d373fec5-a10d-4077-9fb8-84e90be5351a  
=====  
→ MCP server mcp_models (PID: 80897) already terminated  
→ MCP server mcp_controllers (PID: 80898) already terminated  
→ MCP server mcp_views (PID: 80978) already terminated  
→ MCP server mcp_stimulus (PID: 80979) already terminated  
→ MCP server mcp_jobs (PID: 80980) already terminated  
→ MCP server mcp_tests (PID: 81004) already terminated  
→ MCP server mcp_devops (PID: 81005) already terminated  
✓ Cleanup complete
```

Model Context Protocol(MCP)

> 用 mcp 幫我找國軍高雄總醫院網路掛號許競文醫師有哪些門診時刻？

- 我來幫你使用 MCP Chrome DevTools 在國軍高雄總醫院網路掛號系統中查詢許競文醫師的門診時刻。
- 我現在就在網路掛號系統中，讓我選擇許競文醫師來查詢他的門診時刻：

● 很好！已經選擇了許競文醫師。現在讓我點擊搜尋按鈕來查詢他的門診時刻：

```
● chrome-devtools - click (MCP)(uid: "4_235")
  └─ # click response
     Successfully clicked on the element
     ## Page content
     ... +100 lines (ctrl+r to expand)
```

● 好的！我找到了許競文醫師的門診時刻。讓我檢查其他週的門診時間，先看下一週：

```
● chrome-devtools - click (MCP)(uid: "5_36")
  └─ # click response
     Successfully clicked on the element
     ## Page content
     ... +113 lines (ctrl+r to expand)
```

● 太好了！讓我繼續檢查其他週的門診時間，看看下二週：

```
● chrome-devtools - click (MCP)(uid: "6_38")
  └─ # click response
     Successfully clicked on the element
     ## Page content
     ... +113 lines (ctrl+r to expand)
```

● 成功使用 MCP Chrome DevTools 查詢到許競文醫師的門診時刻！

許競文醫師門診時刻表：

科別： 外科
診間： 外科 1 診室

門診時間：

每週固定門診：
- 週一上午 (例：10/27, 11/03)
- 週三上午 (例：10/29, 11/05)

門診時段： 僅上午時段，下午和夜診無門診

目前狀況：
- 本週 10/22(三) - 已額滿
- 下週 10/27(一) - 可預約 ✓
- 下週 10/29(三) - 可預約 ✓
- 下二週 11/03(一) - 可預約 ✓
- 下二週 11/05(三) - 可預約 ✓

許競文醫師屬於外科，固定每週一和週三上午有門診，在外科 1 診室看診。目前除了本週三已額滿外，其他時段都還可以預約掛號。

Chrome DevTools

RSpec

```
Claude Code v2.0.31
Opus 4.1 · Claude Max
/Users/nickle/projects/medicores/medicore

> /mcp

task (tests) [read-only]

Tool name: task
Full name: mcp__tests__task

Description:
Execute a task using Agent tests. RSpec testing, factories, and test coverage specialist Thinking budget levels: "think"
< "think hard" < "think harder" < "ultrathink".

Parameters:
• prompt (required): string - The task or question for the agent
• new_session: boolean - Start a new session (default: false)
• system_prompt: string - Override the system prompt for this request
• description: string - A description for the request
• thinking_budget: string - Thinking budget: "think" < "think hard" < "think harder" < "ultrathink". Each level
increases Claude's thinking allocation. Auto-select based on task complexity.
```

- bin/dev
- claude
- codex
- claude-on-rails



https://kaochenlong.com/openspec



首頁 > 文章列表 > OpenSpec 讓 SDD 變簡單的三個指令

OpenSpec 讓 SDD 變簡單的三個指令

2026.01.15 約 9,190 字

在[前一篇文章](#)我們介紹了 SDD (Spec-Driven Development) 的方法論，簡單的說，SDD 的重點就是在寫程式之前先把「規格」定義清楚，讓我們跟 AI 對「什麼叫做完成」有共同的認知，這樣到時候做出來的東西才不會雞同鴨講。

雖然方法論介紹完了，但從理論到實務的 Gap 其實不小，如果要自己手寫規格可能有點辛苦，對新手來說一開始可能也不知道怎麼下手。還好有很多厲害的大大幫我們做了方便的工具，讓我們可以透過幾個簡單的指令就能建立規格並套用在專案裡面。

Active Changes 1

outpatient-claim-cql-validation
28 mins ago · 0/50 tasks

Tasks 0/50 Proposal Design Specs +35

Why

現有門診申報系統的檢核邏輯散落在 Delphi 程式碼中（主要在 `FUNCTIONopi.pas`），難以維護、測試和更新。健保規則頻繁變動，需要一個宣告式、可版本化的規則引擎來管理檢核邏輯。採用 CQL (Clinical Quality Language) 作為規則定義語言，可實現規則與程式碼分離，提升可維護性和可測試性。

What Changes

- 建立 CQL 規則引擎，支援解析和執行 CQL 表達式
- 設計 FHIR 資料模型映射層，將 TOPOPB/TOPDET/DTLFA/ORDFA 等資料表轉換為 FHIR 資源
- 實作三階段檢核機制：
 - 即時檢核：醫令輸入時驗證健保給付規則
 - 批次檢核：資料審核階段的完整性檢查
 - 申報前檢核：產生 XML 前的最終驗證
- 建立規則管理介面，支援規則的新增、修改、啟用/停用
- 提供檢核結果 API，整合至現有申報流程



測試開發

TDD

Test-Driven Development



TDD

- **步驟 1: Red** – 運行測試，識別失敗

- **Bash**(RAILS_ENV=test bundle exec rspec --format documentation --dry-run 2>&1 | grep -A 1 "FAILED\|pending" | head -100)
└ 565 examples, 0 failures, 1 pending

- 讓我實際運行測試來看失敗情況：

- **Bash**(RAILS_ENV=test bundle exec rspec --format progress --fail-fast 2>&1 | tail -100) timeout: 5m 0s
└F

```
Failures:
```

```
... +18 lines (ctrl+o to expand)
```

- 找到第一個失敗！讓我檢查這個問題：

- **Read**(app/controllers/admin/users_controller.rb)
└ Read **30** lines

- **Read**(spec/controllers/admin/users_controller_spec.rb)
└ Read **20** lines

- 問題找到了！測試使用 `medicore_auth_user` 作為參數鍵，但控制器在第 24 行直接訪問 `params[:user]`。讓我查看完整的控制器：

- **Read**(app/controllers/admin/users_controller.rb)
└ Read **99** lines

- 完美！找到問題了。控制器中有不一致：

- `user_params` 使用 `:medicore_auth_user` (正確)
- 但 `update` 方法中直接訪問 `params[:user]` (錯誤)

- **步驟 2: Green** – 修復代碼使測試通過

- **Update**(app/controllers/admin/users_controller.rb)

TDD

```
Page content: Routing Error
No route matches [GET] "/auth/google_oauth2"
Rails.root: /Users/nickle/projects/medicores/medicore
Application Trace | Framework Trace | Full Trace
Routes
Routes match in priority from top to bottom
Helper (Path / Url) HTTP Verb Path Controller#Action Sour
... +107 lines (ctrl+o to see all)

測試運行完成。結果摘要：

測試結果：
- 總共：378 個測試
- 失敗：120 個 ❌
- 成功：258 個 ✅

主要失敗原因：

1. OAuth 路由缺失 (約 10+ 失敗)
- 錯誤：No route matches [GET] "/auth/google_oauth2"
- 影響：Google OAuth 認證相關測試

2. JavaScript/Stimulus Controller 測試 (約 20+ 失敗)
- Sidebar Controller 測試失敗
- Theme Controller 測試失敗
- 可能與 localStorage 或控制器邏輯有關

3. System 測試 (約 40+ 失敗)
- 國際化 (i18n) 測試失敗
- 語言選擇器 (Language Selector) 測試失敗
- Dashboard 測試失敗

4. WebPush Engine 測試 (約 30+ 失敗)
- 訂閱控制器測試失敗
- 模型驗證測試失敗
- View 相關測試失敗

5. Remember Me 功能測試 (約 10+ 失敗)

需要我幫您修復這些失敗的測試嗎？

> try "fix typecheck errors"
```

```
claude (Python) 361
+ /google_oauth2|keycloak/ }
24
25 # API authentication
26 namespace :api do

claude 362
ith/* 的請求。我需要在主應用的 routes 中添加明確的路由來處理

從原本的 120 個失敗減少到 85 個失敗，已經修復了約 35 個測試！

主要完成的修復：
1. ✅ 創建了測試資料庫
2. ✅ 修復了 OAuth 路由配置
3. ✅ 配置了測試環境的 OAuth providers
4. ✅ 修復了 OAuth callback 路由
5. ✅ 創建了 OAuth 測試輔助方法

剩餘的主要問題：
1. OAuth feature 測試 (需要調整測試方法以使用 callback)
2. JavaScript/Stimulus Controller 測試
3. System 測試 (i18n, Language Selector, Dashboard)
4. WebPush Engine 相關測試
5. Remember Me 功能測試

現在測試套件處於更好的狀態。需要我繼續修復其他失敗的測試嗎？

ie
OAuth request
[get, :post]
cks#google_oauth2'

cks#google_oauth2'

'show', format: :js
instead
```

太好了！測試結果有顯著改善：

測試結果摘要：
- 總共：565 個測試
- 成功：476 個 ✅ (84%)
- 失敗：85 個 ❌ (15%)
- 待定：4 個 ⏸

從原本的 120 個失敗減少到 85 個失敗，已經修復了約 35 個測試！

- 主要完成的修復：
- ✅ 創建了測試資料庫
 - ✅ 修復了 OAuth 路由配置
 - ✅ 配置了測試環境的 OAuth providers
 - ✅ 修復了 OAuth callback 路由
 - ✅ 創建了 OAuth 測試輔助方法

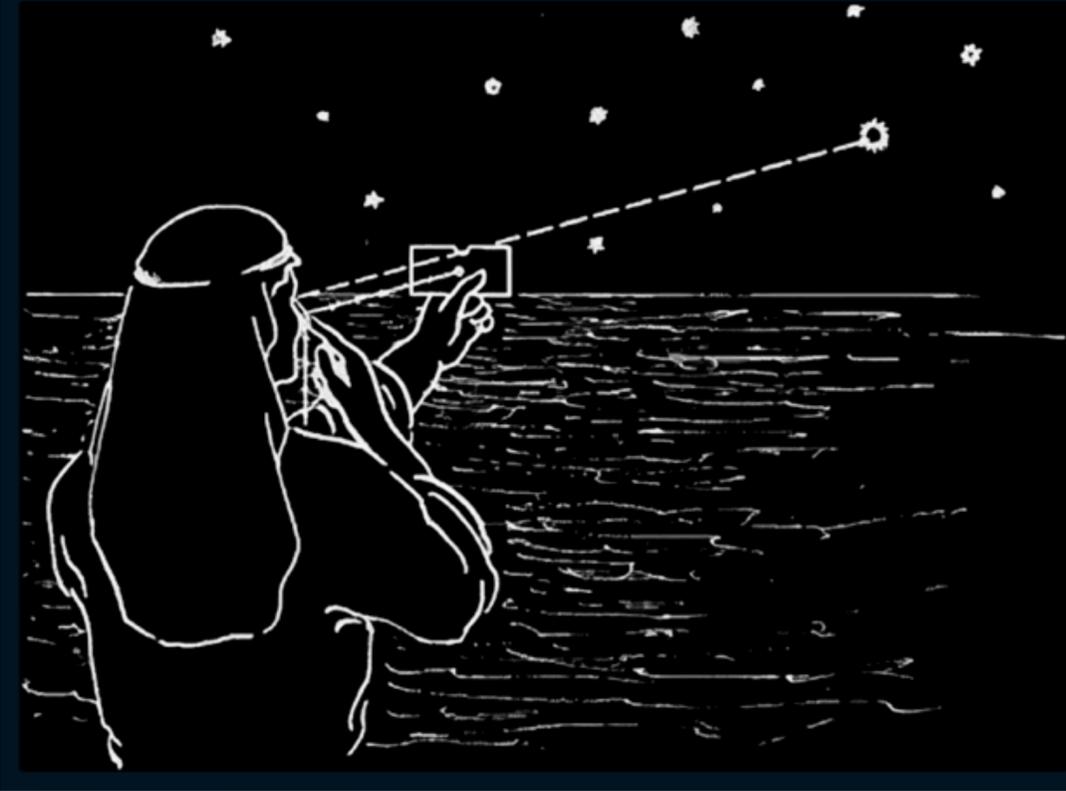
- 剩餘的主要問題：
- OAuth feature 測試 (需要調整測試方法以使用 callback)
 - JavaScript/Stimulus Controller 測試
 - System 測試 (i18n, Language Selector, Dashboard)
 - WebPush Engine 相關測試
 - Remember Me 功能測試

現在測試套件處於更好的狀態。需要我繼續修復其他失敗的測試嗎？



Kamal

Kamal is named after [the ancient Arab navigational tool](#) used by sailors to keep course by determining their latitude via the Pole Star. (Kamal was formerly known as MRSK).



Kamal is named after [the ancient Arab navigational tool](#) used by sailors to keep course by determining their latitude via the Pole Star

不用 Kubernetes 的複雜，就能在任意 VPS 上做到零停機部署



Kamal deploy

部署前檢查清單

- `.env` 檔案已設定所有必要變數
- `KAMAL_REGISTRY_PASSWORD` 已更新為實際密碼
- PostgreSQL 資料庫已建立
- SSH 可以無密碼登入 Synology
- Docker Registry 可以登入
- 必要的 port 未被占用

1. 測試連線

```
# 測試 SSH 連線
ssh ncdr@192.168.2.28 "echo 'SSH connection successful'"

# 測試 Docker Registry 登入
docker login dx.802.org.tw -u ncdr
```



Kamal deploy

2. 執行部署

```
# 載入環境變數
```

```
> claude
```



Claude Code v2.0.31
Sonnet 4.5 · Claude Max
/Users/nickle/projects/portal

```
> 用 kamal deploy
```

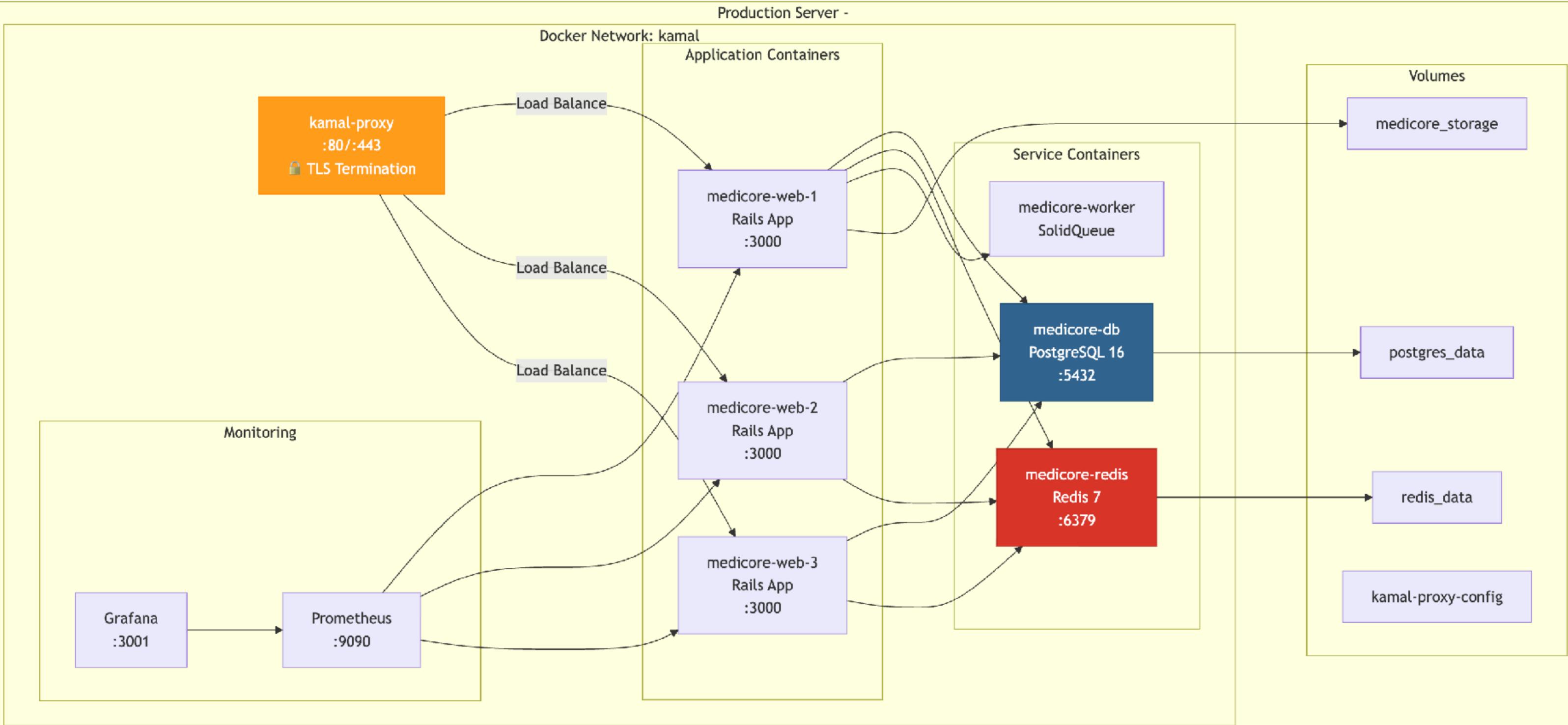
- I'll help you deploy using Kamal. Let me first check if Kamal is configured in this project and then run the deployment.
- **Search**(pattern: "**/config/deploy.yml")
 - └ Found **1** file (ctrl+o to expand)
- **Bash**(which kamal)
 - └ /Users/nickle/.rbenv/shims/kamal
- Great! Kamal is configured and installed. I'll now run the deployment:
 - └ **Interrupted** · What should Claude do instead?

```
> █
```

自動修正錯誤

```
# 建立初始管理員 (如需要)  
kamal app exec "cd test/dummy && rails console"
```

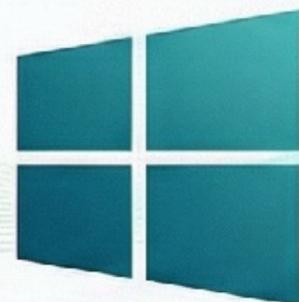




Windows OS free?



打造跨平台的 C# 開發環境：告別 Windows



Windows OS Free

讓 C# 專案的開發、建置與部署不再依賴 Windows 作業系統。



擁抱 DevOps 文化

實現跨平台的自動化建置、測試與部署，提升開發效率。

關鍵技術組合



語言：C#

使用現代、強大且物件導向的程式語言進行開發。



框架：.NET

透過開源的 .NET 框架（取代傳統 .NET Framework）實現跨平台能力。



工具：VS Code & Docker

採用輕量級編輯器與容器化技術，簡化開發與部署流程。

File Explorer (C_SHARP):

- .claude
 - CLAUDE.md
 - settings.local.json
- .kamal
 - hooks
 - secrets
 - ClinicBooking.Api
 - ClinicBooking.Domain
 - ClinicBooking.Tests
- config
 - deploy.yml
- HelloCSharp
 - bin
 - obj
 - Properties
 - appsettings.Development.json
 - appsettings.json
 - HelloCSharp.csproj
 - Program.cs
- published
- scripts
 - init-db.sql
- .dockerignore
- .env.example
- .gitignore
- AI_TIME_TECH_CHOICES.md
- c_sharp.sln
- ClinicBooking.slnx
- DEVELOPMENT_PLAN.md (selected)
- DEVELOPMENT_PLAN.pdf
- DOCKER_COMPOSE_SETUP.md
- docker-compose.yml
- Dockerfile
- KAMAL_SETUP.md
- OPERATIONS_MANUAL.md
- OPERATIONS_MANUAL.pdf
- QUICK_START.md
- QUICK_START.pdf

Bottom navigation:

- 大綱
- 時間表
- 方案總管

```

DEVELOPMENT_PLAN.md > abc # 網掛掛號系統 - 開發文件 (Oracle 混合架構)
1 # 網掛掛號系統 - 開發文件 (Oracle 混合架構)
2
3 ## 項目概述
4
5 **項目名稱:** 醫院線上掛號系統
6 **目標機構:** 中型醫院
7 **核心功能:** 線上掛號、患者管理
8 **技術棧:** ASP.NET Core + Oracle HIS + PostgreSQL 本地庫 + React
9
10 ### 系統定位
11 - 獨立於 HIS 的掛號前端系統
12 - 讀取 Oracle HIS 的科室、醫生、患者資訊 (VIEW)
13 - 通過 API 寫入掛號資訊和號源更新到 Oracle
14 - 本地 PostgreSQL 存儲掛號記錄備份和統計分析
15
16 ---
17
18 ## 1. 系統架構圖 (混合模式)
19
20 ``mermaid
21 graph TD
22   A[React前端] --> B[AppointmentController]
23   A --> C[DepartmentController]
24
25   B --> D[AppointmentService]
26   C --> E[DepartmentService]
27
28   D --> F[OracleHisAdapter]
29   E --> F
30
31   F --> G[Oracle HIS]
32   F --> H[PostgreSQL]
33
34   G --> I[讀VIEW]
35   I --> J[V_SCHEDULES]
  
```

網掛掛號系統 - 開發文件 (Oracle 混合架構)

項目概述

項目名稱: 醫院線上掛號系統
 目標機構: 中型醫院
 核心功能: 線上掛號、患者管理
 技術棧: ASP.NET Core + Oracle HIS + PostgreSQL 本地庫 + React

系統定位

- 獨立於 HIS 的掛號前端系統
- 讀取 Oracle HIS 的科室、醫生、患者資訊 (VIEW)
- 通過 API 寫入掛號資訊和號源更新到 Oracle
- 本地 PostgreSQL 存儲掛號記錄備份和統計分析

1. 系統架構圖 (混合模式)

```

問題 輸出 偵錯主控台 終端機 連接埠 POSTMAN CONSOLE
{
  "mcpServers": {
    "filesystem": {
      "command": "npx",
      "args": ["-y", "@anthropic/mcp-filesystem", "/path/to/dir"]
    },
    "another-server": {
      "command": "python",
      "args": ["-m", "some_mcp_server"]
    }
  }
}
  
```

這對於需要將 MCP 工具暴露給不支援原生 MCP 協議的系統很有用。例如讓傳統 REST API 客戶端也能使用 MCP 伺服器提供的功能。

你有什麼具體的使用場景想討論嗎？

Terminal window showing:

- zsh
- claude-bmad
- claude
- zsh

Page number: 53

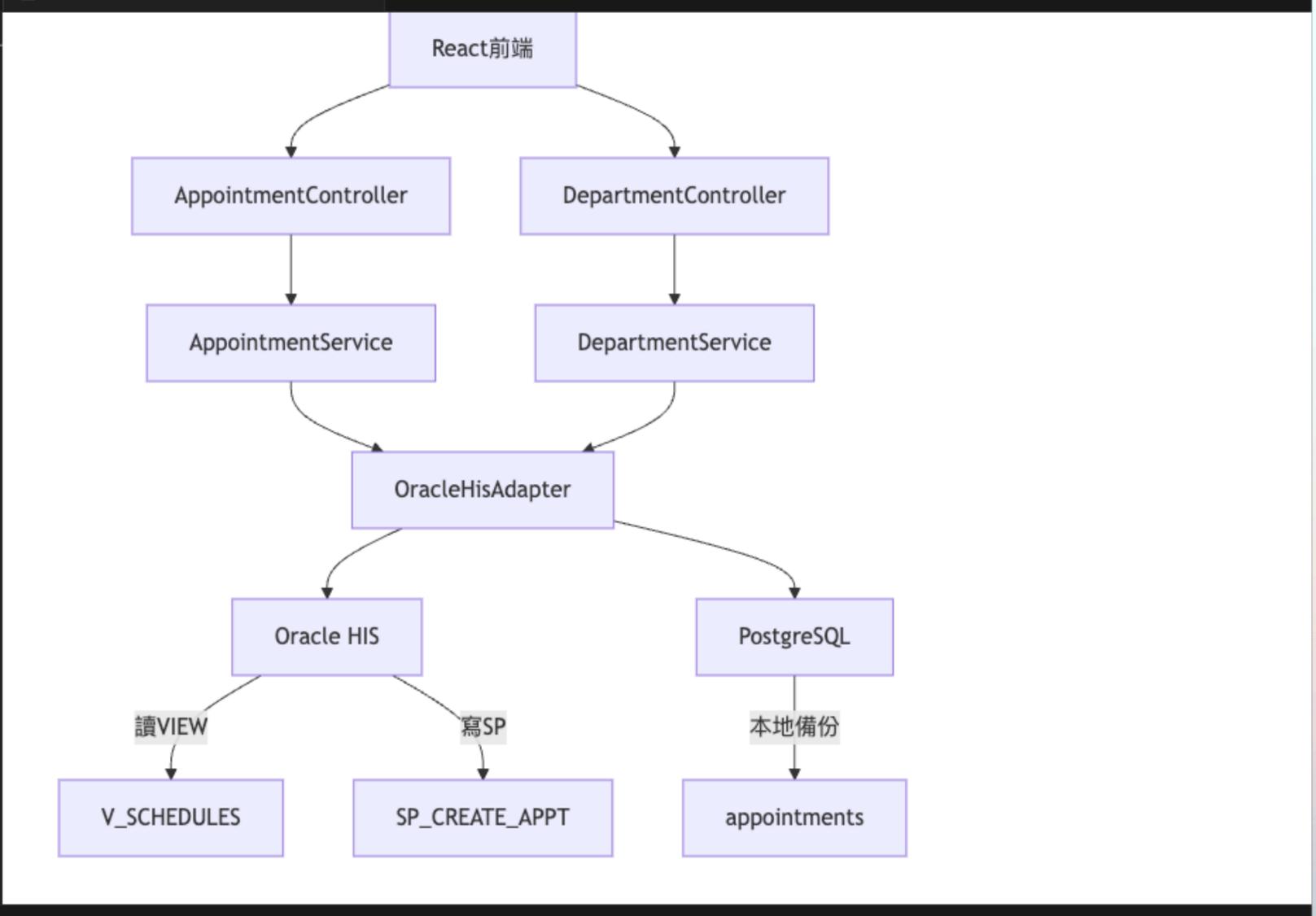
檔案總管

- C_SHARP
 - .claude
 - CLAUDE.md
 - settings.local.json
 - .kamal
 - hooks
 - secrets
 - ClinicBooking.Api
 - ClinicBooking.Domain
 - ClinicBooking.Tests
 - config
 - deploy.yml
 - HelloCSharp
 - bin
 - obj
 - Properties
 - appsettings.Development.json
 - appsettings.json
 - HelloCSharp.csproj
 - Program.cs
 - published
 - scripts
 - init-db.sql
 - .dockerignore
 - .env.example
 - .gitignore
 - AI_TIME_TECH_CHOICES.md
 - c_sharp.sln
 - ClinicBooking.slnx
 - DEVELOPMENT_PLAN.md
 - DEVELOPMENT_PLAN.pdf
 - DOCKER_COMPOSE_SETUP.md
 - docker-compose.yml
 - Dockerfile
 - KAMAL_SETUP.md
 - OPERATIONS_MANUAL.md
 - OPERATIONS_MANUAL.pdf
 - QUICK_START.md
 - QUICK_START.pdf

大綱

- 時間表
- 方案總管

```
DEVELOPMENT_PLAN.md > abc # 網掛掛號系統 - 開發文件 (Oracle 混合架構)
1 # 網掛掛號系統 - 開發文件 (Oracle 混合架構)
3 ## 項目概述
10 ### 系統定位
17
18 ## 1. 系統架構圖 (混合模式)
19
20 ```mermaid
21 graph TD
22   A[React前端] --> B[AppointmentController]
23   A --> C[DepartmentController]
24
25   B --> D[AppointmentService]
26   C --> E[DepartmentService]
27
28   D --> F[OracleHisAdapter]
29   E --> F
30
31   F --> G[Oracle HIS]
32   F --> H[PostgreSQL]
33
34   G --> I[讀VIEW | I[V_SCHEDULES]]
35   G --> J[寫SP | J[SP_CREATE_APPT]]
36   H --> K[本地備份 | K[appointments]]
37 ```
38
39 ---
40
41 ## 2. 資料流架構
42
43 ### 讀取流程 (READ Oracle VIEW)
44
45 ```mermaid
46 graph LR
47   [讀者請求] --> I
48   [讀者請求] --> J
49   [讀者請求] --> K
```



問題 輸出 偵錯主控台 終端機 連接埠 POSTMAN CONSOLE

- Bash(git add -A && git commit -m "Remove build artifacts from version control...")
 - [dev 65b56e2] Remove build artifacts from version control
 - 39 files changed, 3779 deletions(-)
 - delete mode 100644 ClinicBooking.Tests/obj/ClinicBooking.Tests.csproj.nuget.dgspec.json
 - ... +38 lines (ctrl+o to expand)
- Bash(git status && echo "----" && git log --oneline -3)
 - On branch dev
 - nothing to commit, working tree clean
 -
 - ... +3 lines (ctrl+o to expand)

● 完成。已從版本控制移除 39 個 obj/ 目錄下的建置產物。

現在工作目錄乾淨，未來這些檔案不會再被追蹤 (已在 .gitignore 中)。

>

? for shortcuts

zsh

claude-bmad

claude

zsh

Update available! Run: brew upgrade claude-code

54

檔案總管

- C_SHARP
 - .claude
 - CLAUDE.md
 - settings.local.json
 - .kamal
 - hooks
 - secrets
 - ClinicBooking.Api
 - ClinicBooking.Domain
 - ClinicBooking.Tests
 - config
 - deploy.yml
 - HelloCSharp
 - bin
 - obj
 - Properties
 - appsettings.Development.json
 - appsettings.json
 - HelloCSharp.csproj
 - Program.cs
 - published
 - scripts
 - init-db.sql
 - .dockerignore
 - .env.example
 - .gitignore
 - AI_TIME_TECH_CHOICES.md
 - c_sharp.sln
 - ClinicBooking.slnx
 - DEVELOPMENT_PLAN.md
 - DEVELOPMENT_PLAN.pdf
 - DOCKER_COMPOSE_SETUP.md
 - docker-compose.yml
 - Dockerfile
 - KAMAL_SETUP.md
 - OPERATIONS_MANUAL.md
 - OPERATIONS_MANUAL.pdf
 - QUICK_START.md
 - QUICK_START.pdf

DEVELOPMENT_PLAN.md > # 網掛掛號系統 - 開發文件 (Oracle 混合架構)

```

1 # 網掛掛號系統 - 開發文件 (Oracle 混合架構)
172 ## 5. ASP.NET Core Oracle 適配層實現
209 ### 5.3 實現示例
306 ---
307
308 ## 6. TDD 開發策略
309
310 ### 測試框架
311 - **xUnit** - 單元測試
312 - **Moq** - Mock Oracle 適配器
313 - **FluentAssertions** - 斷言
314 - **Testcontainers** - Oracle 測試環境 (可選)
315 ---
316
317
318 ## 7. 開發分階段
319
320 ### Phase 1: Oracle 適配層 (第1週)
321 - [ ] 建立 ASP.NET Core 專案 + xUnit 測試項目
322 - [ ] 寫測試: GetDepartments 返回列表
323 - [ ] 實現 OracleHisAdapter 讀取方法
324 - [ ] 寫測試: CreateAppointment 呼叫存儲過程
325 - [ ] 實現存儲過程調用
326
327 ### Phase 2: 業務邏輯層 (第2週)
328 - [ ] 寫測試: AppointmentService 號源驗證
329 - [ ] 實現 AppointmentService
330 - [ ] 寫測試: 患者驗證邏輯
331 - [ ] 實現 PatientService
332
333 ### Phase 3: API 控制器 (第3週)
334 - [ ] 寫測試: POST /api/appointments
335 - [ ] 實現掛號控制器
  
```

問題 輸出 偵錯主控台 終端機 連接埠 POSTMAN CONSOLE

```

• Bash(git add -A && git commit -m "Remove build artifacts from version control.)
  | [dev 65b56e2] Remove build artifacts from version control
  | 39 files changed, 3779 deletions(-)
  | delete mode 100644 ClinicBooking.Tests/obj/ClinicBooking.Tests.csproj.nuget.cgspec.json
  | ... +38 lines (ctrl+o to expand)
• Bash(git status && echo "---" && git log --oneline -3)
  | On branch dev
  | nothing to commit, working tree clean
  | ---
  | ... +3 lines (ctrl+o to expand)
• 完成。已從版本控制移除 39 個 obj/ 目錄下的建置產物。
  |
  | 現在工作目錄乾淨，未來這些檔案不會再被追蹤 (已在 .gitignore 中) ~
  |
  | >
  | ? for shortcuts
  
```

Update available! Run: brew upgrade claude-code

6. TDD 開發策略

測試框架

- xUnit - 單元測試
- Moq - Mock Oracle 適配器
- FluentAssertions - 斷言
- Testcontainers - Oracle 測試環境 (可選)

7. 開發分階段

Phase 1: Oracle 適配層 (第1週)

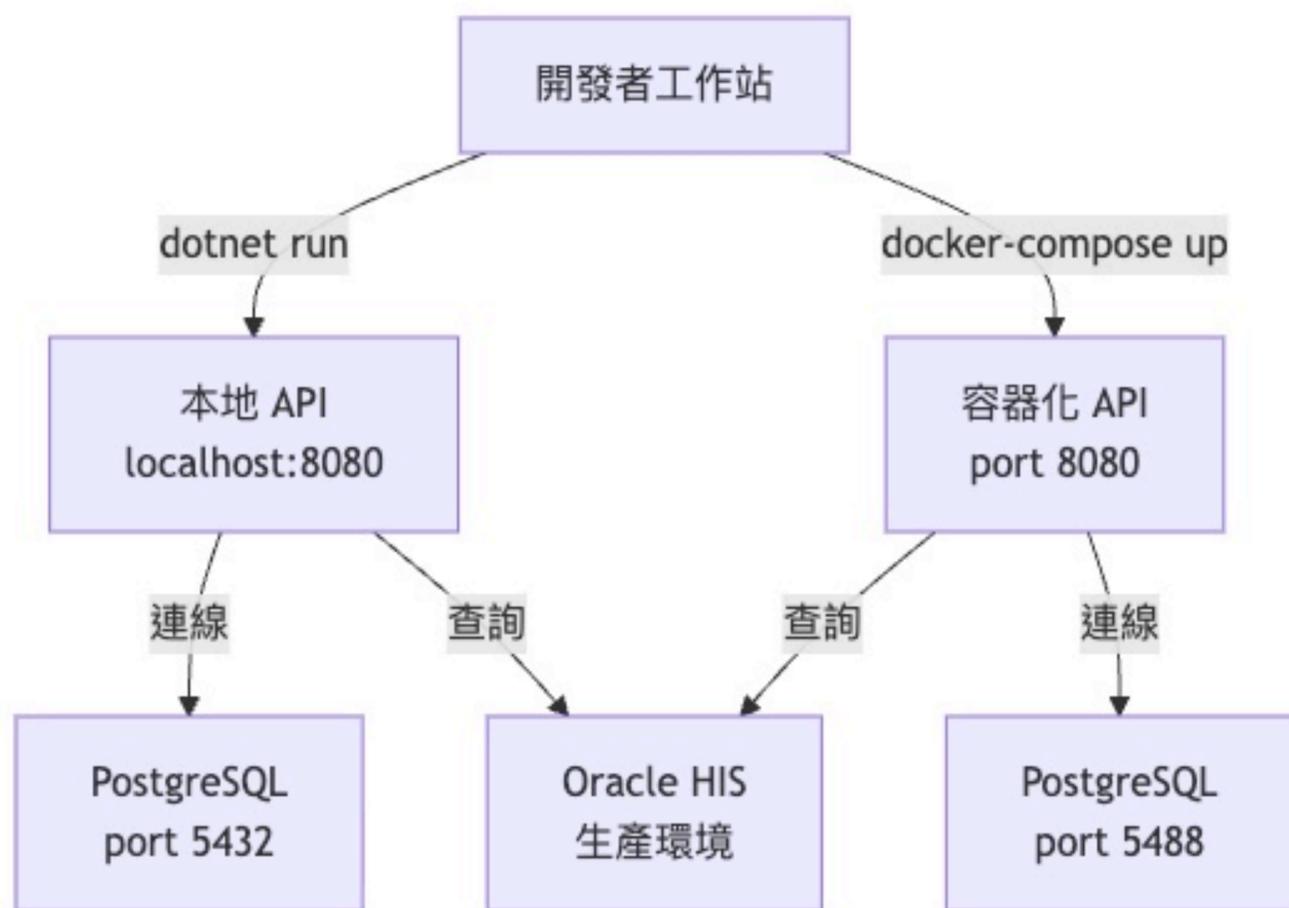
- 建立 ASP.NET Core 專案 + xUnit 測試項目
- 寫測試: GetDepartments 返回列表
- 實現 OracleHisAdapter 讀取方法
- 寫測試: CreateAppointment 呼叫存儲過程
- 實現存儲過程調用

Phase 2: 業務邏輯層 (第2週)

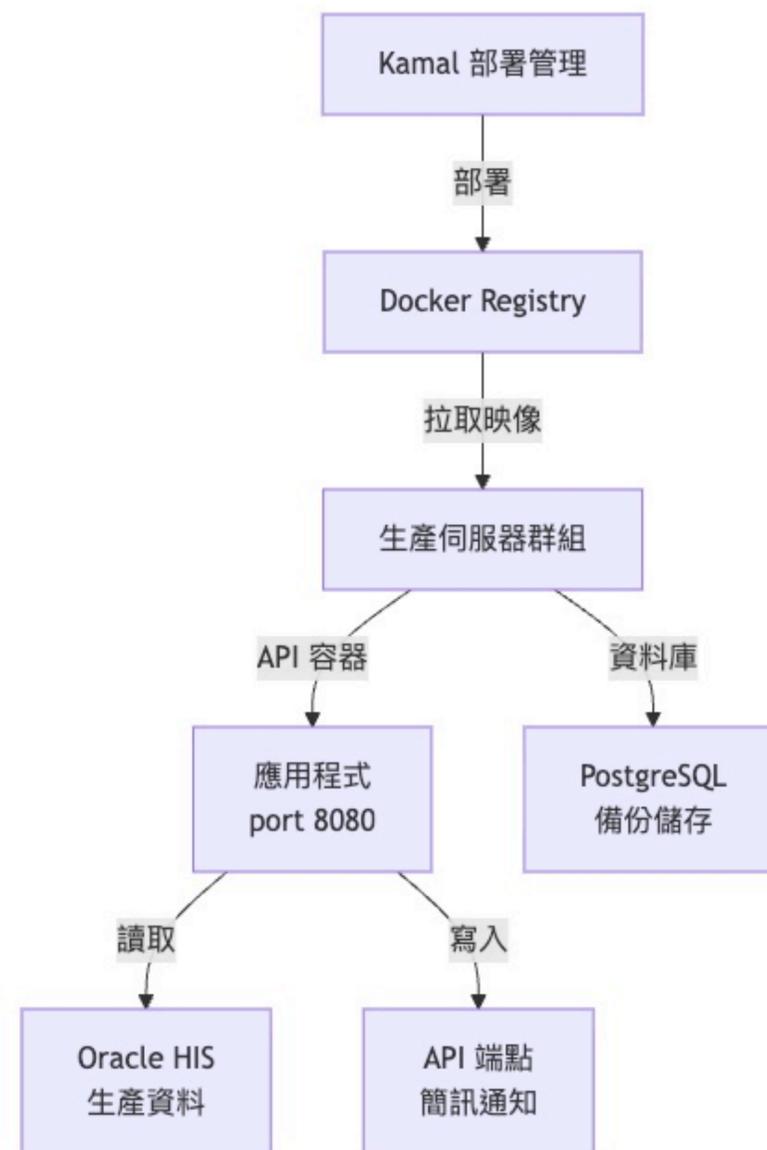
醫院線上掛號系統 - 簡易維運手冊

系統架構

開發環境架構



生產環境架構

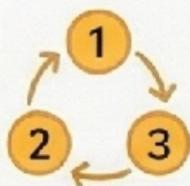


C# 跨平台開發與 Kamal 極速部署指南



核心技術堆疊

-  **.NET SDK:** 跨平台 C# 開發的核心語言與工具集。
-  **Docker & Docker Compose:** 實現開發與生產環境一致性的容器化工具。
-  **Kamal:** 一款實現零停機 (Zero-Downtime) 部署的自動化工具。



本地開發三步驟

- 1 初始化專案:** 使用 `dotnet restore` 指令安裝所有專案依賴。
- 2 啟動服務:** 使用 `docker-compose up` 指令在容器中運行所有服務。
- 3 提交代碼:** 完成測試後，使用 `git push` 將代碼提交至版本庫。



Kamal 自動化部署

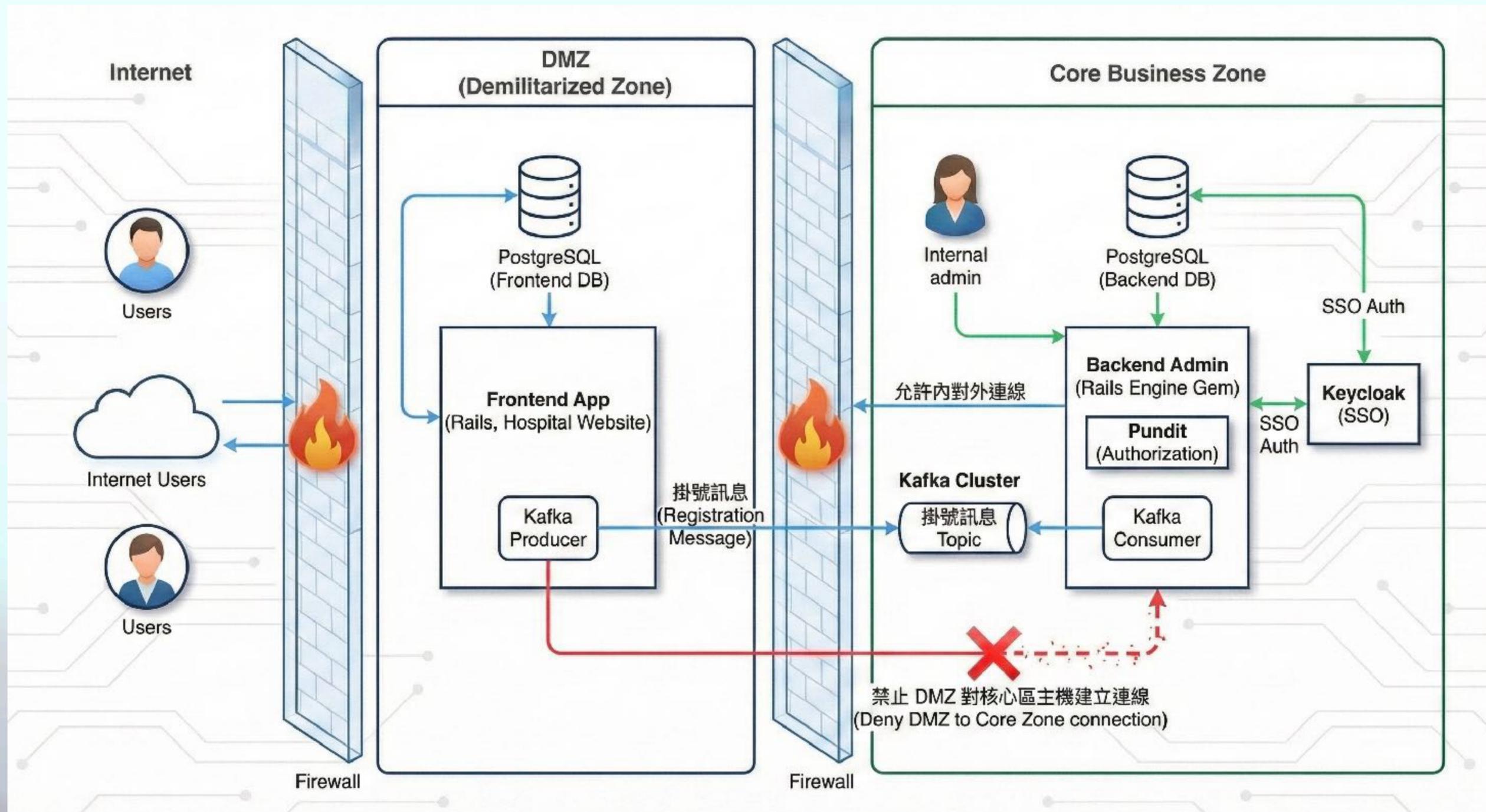
- 1 部署設定:** 在 `config/deploy.yml` 中設定映像檔名稱、伺服器 IP 與網域。
- 2 一鍵部署:** 執行 `kamal deploy` 指令，Kamal 將自動完成建置、推送與啟動。
- 3 零停機更新:** Kamal 會無縫切換新舊容器，確保服務不中斷。



現代化架構成果

-  **容器化一切:** 後端 API、資料庫等所有基礎設施均封裝於 Docker 容器中。
-  **擺脫系統限制:** 整個開發到部署流程，皆可在 macOS、Linux 或 Windows 上無差別運行。

零信任 DMZ 架構



轉型路徑

過去：賣人力、賣工時、賣客製化

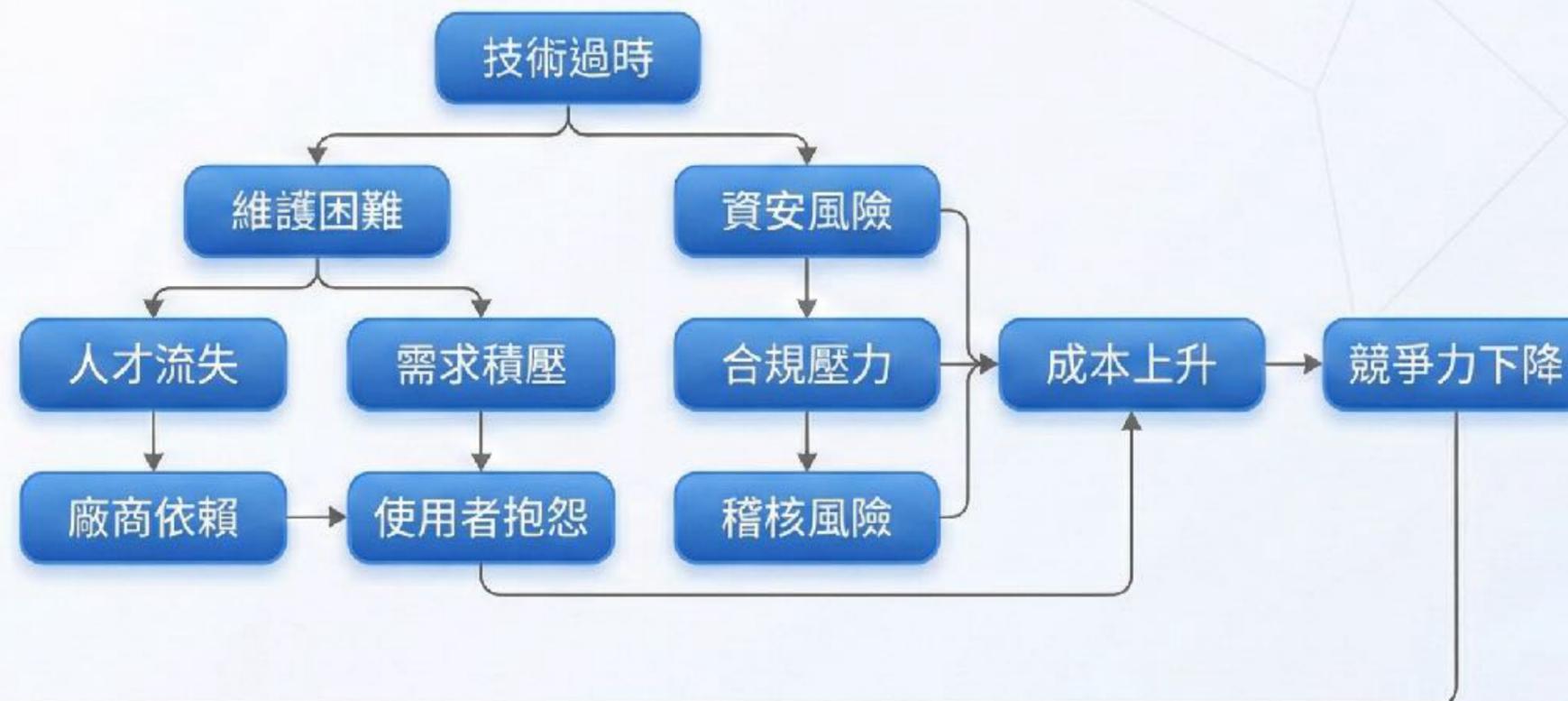
未來：賣解決方案、賣數據價值、賣領域知

識



轉型路徑圖與舊系統痛點解析

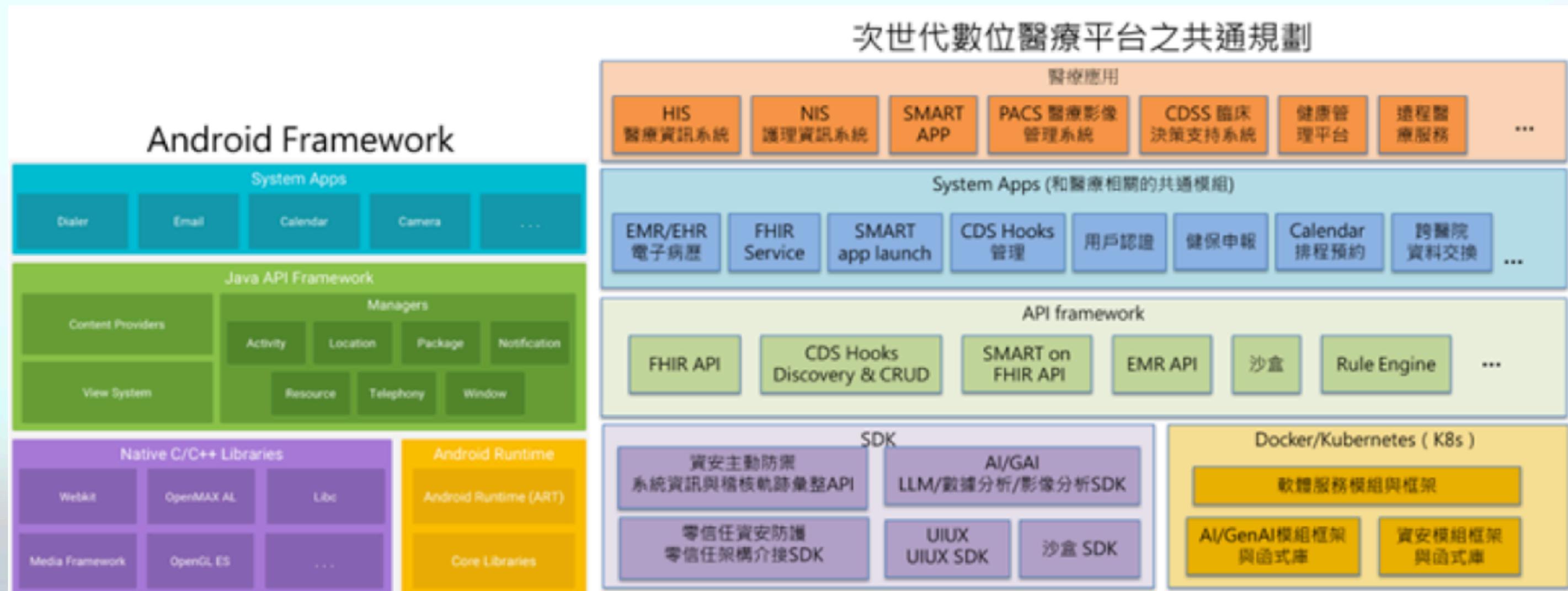
舊系統痛點因果關係



轉型路徑圖



次世代數位醫療平台(1/2)

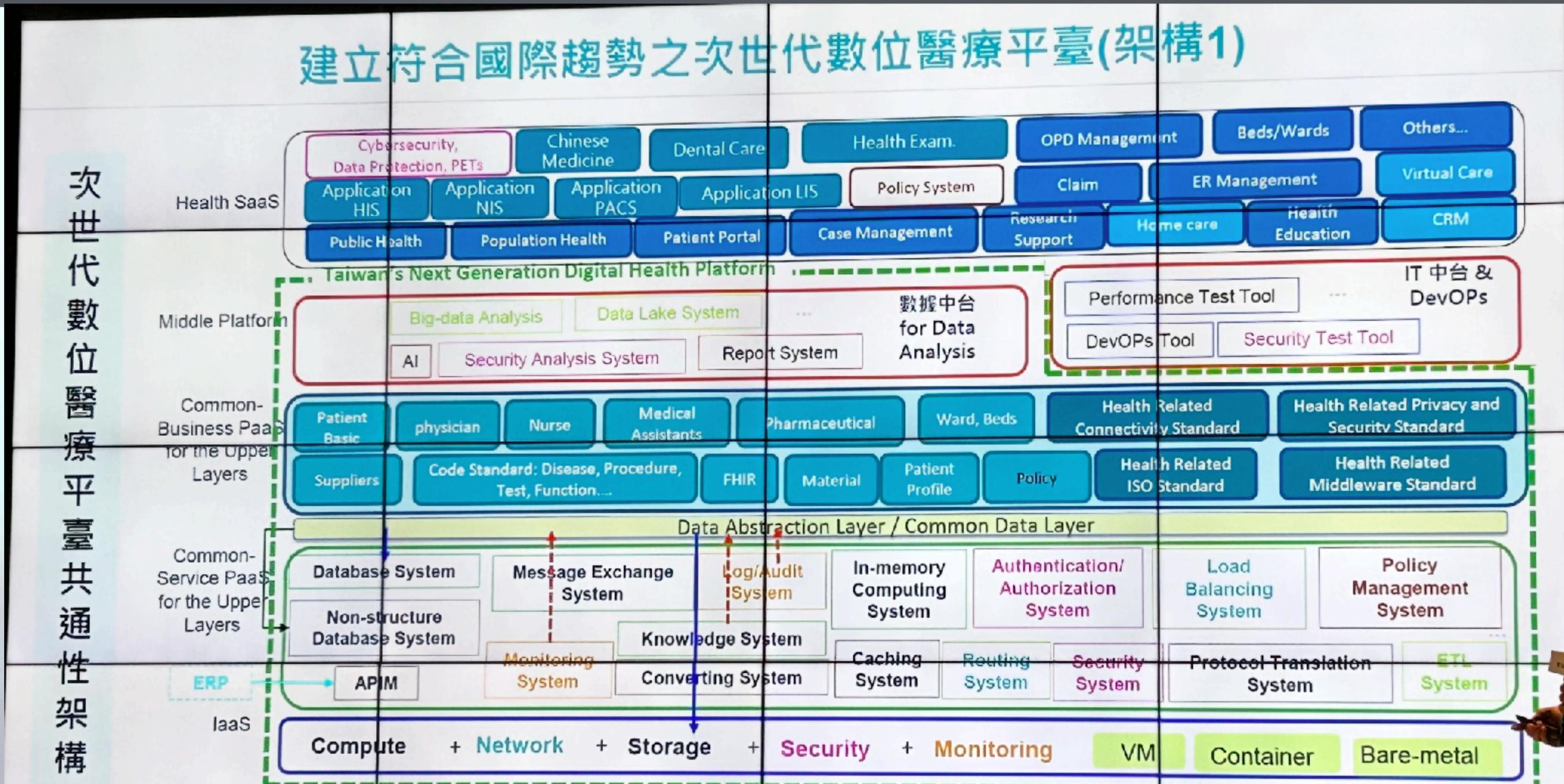


source: 衛福部 blog, <https://blog.mohw.gov.tw/459/>



次世代數位醫療平台(2/2)

建立符合國際趨勢之次世代數位醫療平臺(架構1)



次世代數位醫療平臺共通性架構

健康台灣深耕計畫



健康台灣深耕計畫

🏠 首頁

📖 簡介

📢 公告專區

📄 下載專區

👥 聯絡我們

📄 計畫願景

5年5百億 / 期程：114年-118年

1

優化醫療
工作條件

2

規劃多元
人才培訓

3

導入智慧
科技醫療

4

社會責任
醫療永續

成為亞太地區
最具醫療發展力國家

深耕計畫最新消息

- [公告] 更新健康台灣深耕計畫查核點填報說明會Q&A問答集(115/1/27更新)
- [公告] 健康台灣深耕計畫採購案件通案監督一覽表，供各單位下載使用

活動專區(影片/簡報)

- [公告] 「健康台灣深耕計畫」受補助單位查核點填報說明會簡報及影片(115/1/29更新)
- [公告] 「健康台灣深耕計畫」受補助單位執行注意事項說明會(範疇三一導入智慧科技醫療)簡報及影片(114/11/27更新)

友善連結

- 健康台灣推動委員會
- 政府電子採購網
- 衛福部食藥署核准應用AI/ML技術之醫療器材清單(供受補助單位參考運用)

人才培育

Talent Development



醫療資訊人力的挑戰

- 📌 醫院數位轉型速度加快，但資訊人力供給不足，缺乏懂醫療流程又具備AI、資料分析與系統整合能力的人才。
- 📌 醫療人員訓練集中於臨床專業，資訊應用能力落差大。
- 📌 外部招募成本高、留任率低，難以建立可持續的開發團隊。
- 📌 醫療資訊化 ≠ 單靠購買系統，而是需要「內部懂醫療又懂資訊的人」。



醫療資訊人才培育

📍 跨領域交流

開設「臨床 × 資訊 × AI」研討會及課程，讓醫護人員能理解資料流、演算法與臨床應用之間的關聯，強化數位思維與AI素養，縮短臨床與資訊之間的鴻溝。

📍 尋找內部開發者

系統性培育具備系統開發、資料治理、資安維運能力的關鍵成員，打造具自主研發與維運能力的核心團隊，降低外部依賴。

📍 鼓勵參與技術社群

與學術單位、智慧醫療中心及產業夥伴合作，推動技術交流與創新應用。

📍 鼓勵學習新技術

鼓勵員工持續學習並實際運用新興技術，例如 AI Coding、Vibe Coding、Model Context Protocol (MCP)、agents、skills 等，讓團隊能快速掌握次世代開發工具，保持競爭力與創新動能。

結語：回歸以人為本的設計



理想的系統：符合人的需求

直覺易用

彈性靈活

彈性靈活

滿足需求



不佳的設計：人被迫配合系統

僵化繁瑣

學習曲線高

被迫遷就



「一個好的系統，應該是去適應人，而非讓人去遷就它。」



Any Question?