國泰金控

生成式AI與雲原生技術的融合實現IT現代化

Otto Yen

2025.07

Cloud Strategy Development

Digital Data Technology

顏勝豪 Otto

現在

- 國泰金控 數數發 雲端技術架構師
- 負責執行集團雲端轉型計劃，擬定集團上雲策略

過去

- 10年以上的軟體專案開發經驗
- 歷經紮實的養成訓練，由程式設計師→
  系統設計師→系統分析師→技術經理→協理
- 帶領專案開發部門承接多種領域之
  - ✓ 外商銀行交易系統、商業智慧分析
  - ✓ 警政情資整合分析平台、大數據分析平台架構設計與建置

Agenda

01 國泰雲端轉型旅程

02 Demo: Gemma3 on GCP Cloud Run

03 Demo: 雲端架構生成助手 on Amazon Bedrock

04 Demo: 現代化與重構策略 by GitHub Copilot

05 Demo: 雲原生架構 by OpenAI Codex

06 Live Demo: 雲原生架構 by Claude Code

藉由雲端大會分享每年國泰雲端轉型旅程

2021
Cloud Ready

金融業於混合雲架構下應用系統之上雲策略

01

2022
Cloud Adoption

金融業於雲端轉型過程中應用系統大規模上雲策略

02

2023
Cloud Adoption

金融業雲端轉型三部曲之挑戰與突破

03

2024
Cloud Adoption

金融業雲端轉型後的End Game

04

今年
Cloud Adoption

生成式 AI 與雲原生技術的融合：實現 IT 現代化

05

國泰集團 7 年上雲計畫 — 三部曲

Cloud Ready
啟動集團上雲計畫

- 應用系統上雲評估
- 評估雲地系統相依性
- 進行雲端架構設計
- 建立雲端遷移計畫
- 建立組織治理框架

2020

基礎架構 | 管理治理 | 組織人才 | 應用系統

Cloud Adoption
5年100套系統大規模上雲

- 提供 SaaS 上雲轉型平台服務
- 致力雲端應用及技術發展
- 建立雲端治理組織框架
- 打造高安全、高可用雲端環境

2021

2024 累計 71 套系統上雲

國泰金控 Cathay Financial Holdings	16
國泰世華銀行 Cathay United Bank	31
國泰人壽 Cathay Life Insurance	18
國泰產險 Cathay Century Insurance	6

2025

Cloud Modernization
雲端加速IT現代化

- 解決技術債問題
- 實現永續發展
- 加速業務創新
- 現代化資安與合規

2027

成本節省 Serverless 架構	-39%	時間
員工生產力 容器化技術	+80%	管理效率
營運韌性 托管型資料庫	-13%	時間
商業敏捷性 托管型分析服務	-20%	時間
永續性 實現永續發展目標	-60%	能源消耗

Source: Business Value of Cloud Modernization, Jan 2023, by known research

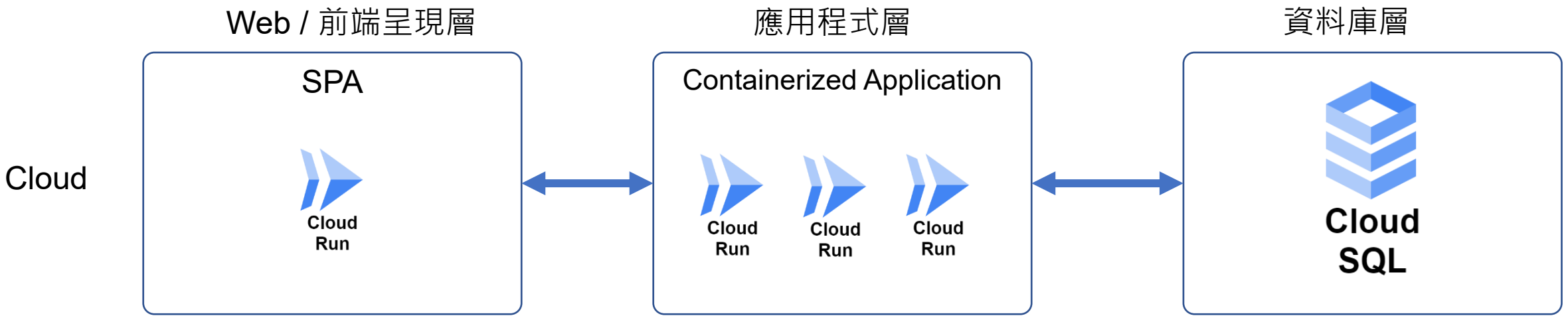
應用系統上雲從容器化到Serverless

比市場快半步

資源彈性伸縮

省錢

天然高可用



除了應用系統上雲GPU算力需求也可以採用雲端PaaS或SaaS服務

- 自行建置GPU機房。
- 採用自有機房或租用外部機房。
- 電力與散熱系統會有額外費用。
- 設備規格變化快。

- 租用外部IDC GPU算力中心。
- 費用稍高，但費用包含電力與空調、空間資源。
- 會需要建置到IDC的線路與通訊設備。

- 直接在雲端上租用有GPU算力之虛擬機。
- 若以長期承諾使用方案有較高的折扣。
- 若採用PAYG方案，資源可按需開關，不使用費用歸0。

H100 x 8張

31 ~ 38 美金/小時

- 使用內含GPU算力容器平台，如GKE。
- 視工作負載的需求動態增減容器數量。
- 容器不用時可自動關閉。

單一Pod/H100

5.893 美金/小時

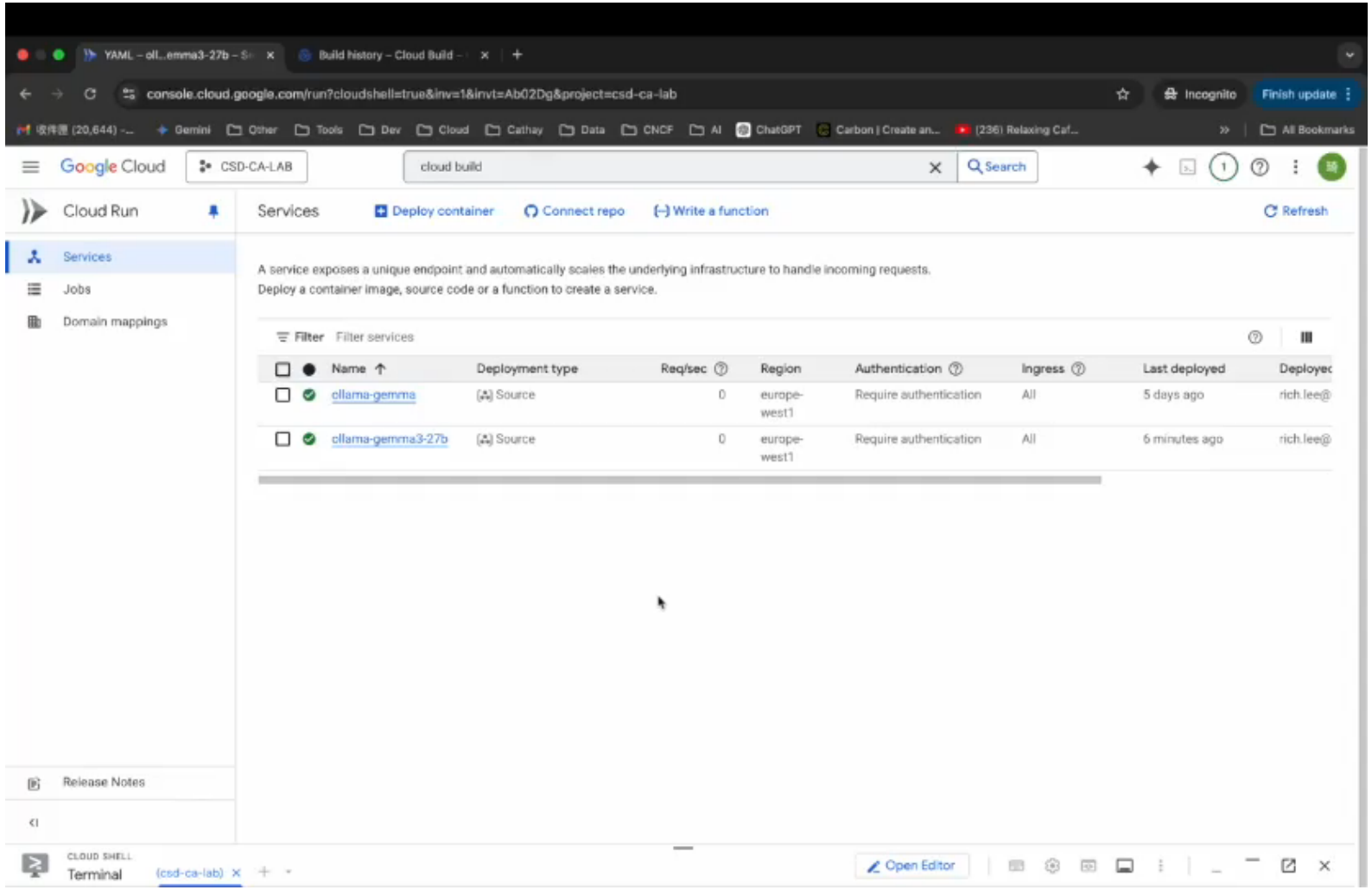
- 以Serverless方式使用GPU算力資源。
- 以Cloud Run，使用內含GPU算力容器運行。
- 使用Model Garden掛載LLM採用API調用模型。

4 CPU, 16G RAM/L4

1.06 美金/小時

Demo: Gemma3 on Cloud Run

4 CPU 16GB NVIDIA L4 1GPU，約 26.5 tokens/sec



1. 透過GCP 主控台 在 Cloud Run 設定使用 Gemma3 27b 的 LLM模型

2. 在Shell模式，詢問 Why sky is blue? (為何天空呈現藍色?)

3. 可以看到 Cloud Run 將問題拋到 Gemma 3 後，Gemma3 開始回答，一行就是一個 Token ，有時是單字、有時是組合的片語或句子，這些回應組合起來就是一般在瀏覽器上會看到的 AI 回答內容

4. Cloud Run調用LLM回應速度為 26.5 Tokens/sec，且Cloud Run + 開源模型，不使用時會自動關閉不產生費用

回到今天的主題 — 生成式AI與雲原生技術的融合

Smart Archie 雲端架構生成助手可以解決三大問題

	大規模上雲的架構圖需求	架構圖規格不一致	過程缺乏完整記錄
問題	國泰面臨大量上雲需求，需繪製大量遵循技術、政策規範的雲端架構圖，但架構師資源相對稀缺。	不同架構師繪製的架構圖格式不一致專案溝通成本增加。	專案跨組織溝通時，架構圖多僅保留最終版本，更新過程缺乏完整記錄，難以追溯需求變更的過程。
解方	**自動生成架構圖內建技術政策** Smart Archie 能自動生成雲端架構圖，同時內建國泰集團的技術政策和標準，確保架構符合公司要求。解決架構師資源稀缺的問題，亦確保每個架構圖都符合規範，減少人工檢查的負擔。	**統一架構圖格式與模板** Smart Archie 提供統一的架構圖模板與繪製工具，無論哪位架構師繪製，都能保持相同的格式和標準，降低格式不一致造成的溝通成本，提升跨部門專案協作效率。	**版本控制與需求變更管理** Smart Archie 提供完整的版本控制功能，架構圖每次更新都會記錄變更歷程，含變動細節。跨組織溝通時，不僅可以查看最終版本，還可以追溯需求變更具體過程。

國泰金控

Smart Archie
架構圖智能生成助手
Product Demo

Cloud Strategy Development

Digital Data Technology

接下來我們要模擬一位架構師

從地端到雲端的遷移法

什麼是Cathay 6R

「Cathay 6R」是國泰金控（Cathay Financial Holdings）為推動集團內部大規模雲端遷移所制定的一套方法論，旨在協助旗下子公司（如銀行、人壽、產險、證券等）評估並選擇最適合的雲端遷移策略。 MoneyDJ +8

Cathay 6R 的六種遷移策略

這套方法論涵蓋以下六種遷移策略： Yahoo奇摩新聞

1. **Rehost（重新託管）**：將現有應用程式直接遷移至雲端基礎設施（IaaS），不進行重大修改。
2. **Re-platform（平台轉換）**：對應用程式進行必要的調整，使其能在雲端平台（PaaS）上運行。
3. **Refactor（重構）**：對應用程式進行部分重構，以充分利用雲端原生功能。
4. **Rewrite（重寫）**：從頭開始重新開發應用程式，以最佳化其在雲端環境中的性能和可擴展性。
5. **Replace（替換）**：用現成的雲端解決方案（如SaaS）替代現有應用程式。
6. **Retain（保留）**：保留應用程式在原有環境中，暫不進行遷移。 Electrum Cloud +6

透過這六種策略，國泰金控能夠根據各應用系統的特性、業務需求和技術考量，制定最適合的雲端遷移計畫。

~~為主進行遷移規劃。遷移目標是讓新環境能忠實反~~

~~通常是使用CSP提供的工具直接搬VM。~~

發點，重構或重新設計系統，導入新技術與架構

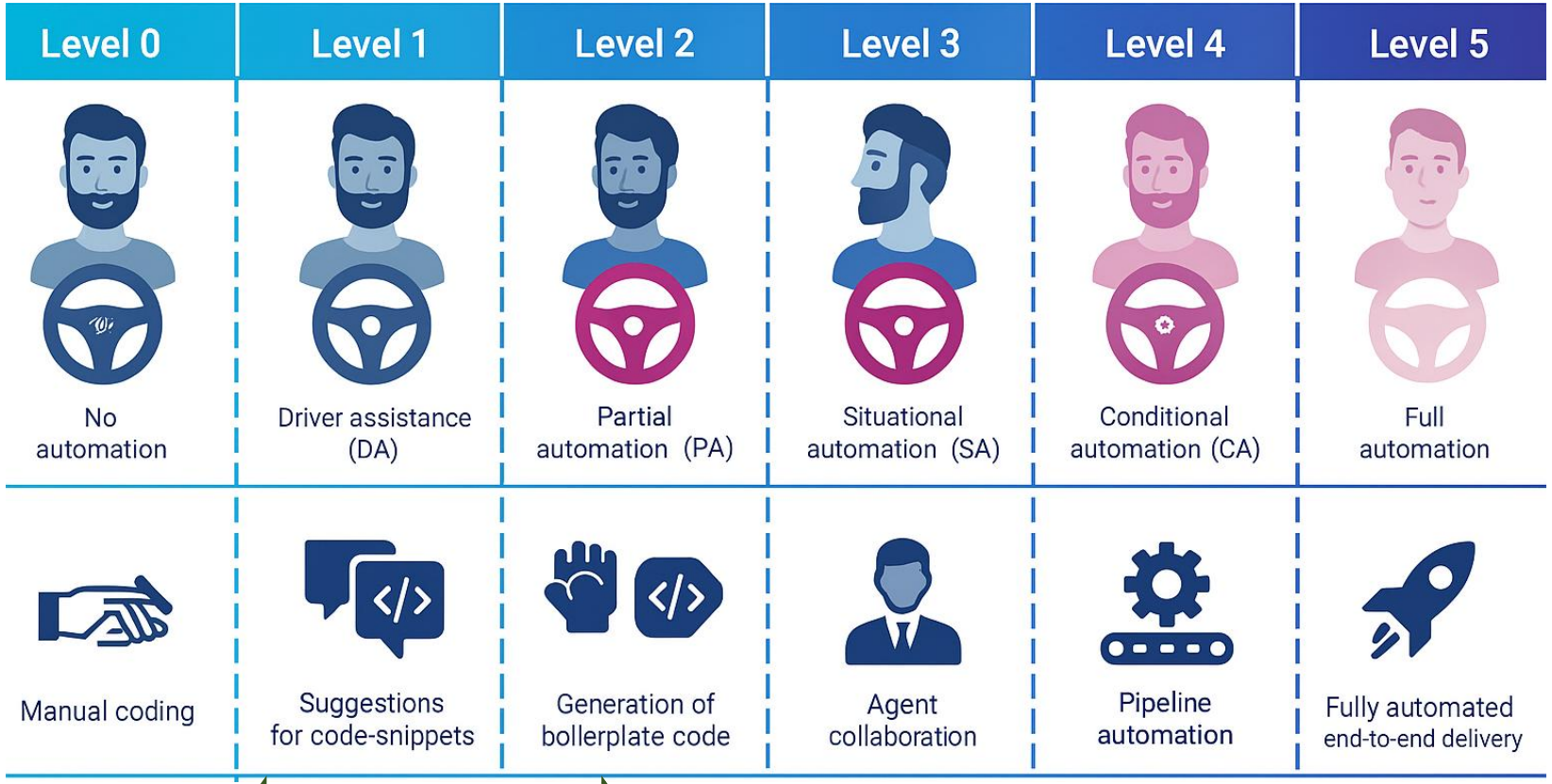
Cathay 6R）。

分析、應用映射、依賴關係分析等）協助進行遷

對象	導入難度	改造程度
上雲、低風險需求	低	低
架構轉型者	高	高
雜遺留系統、大型企業	中至高	中

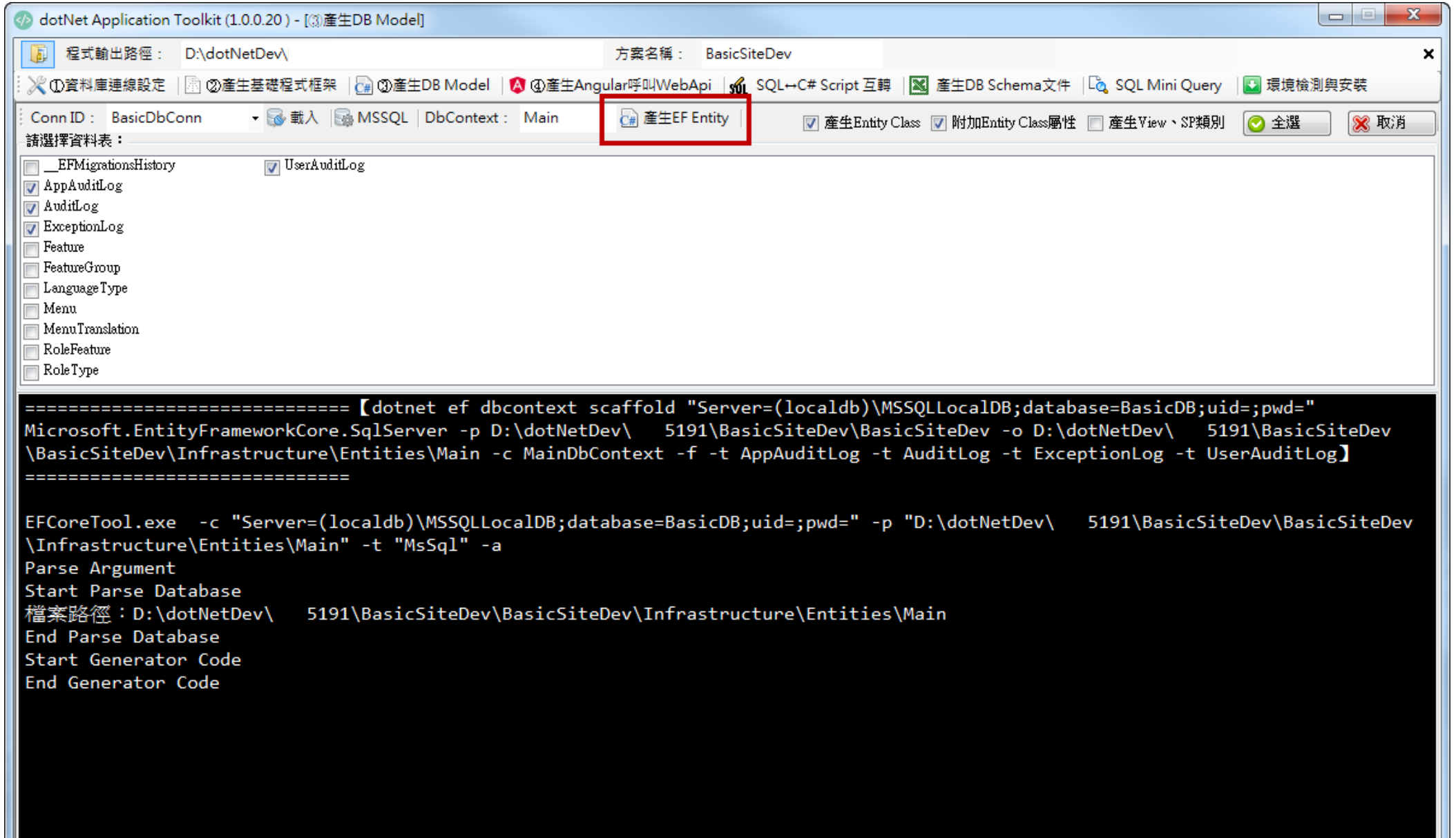
國泰金控

SAE的自動駕駛5個等級 vs AI Coding的5個等級 (人為主 vs AI為主)



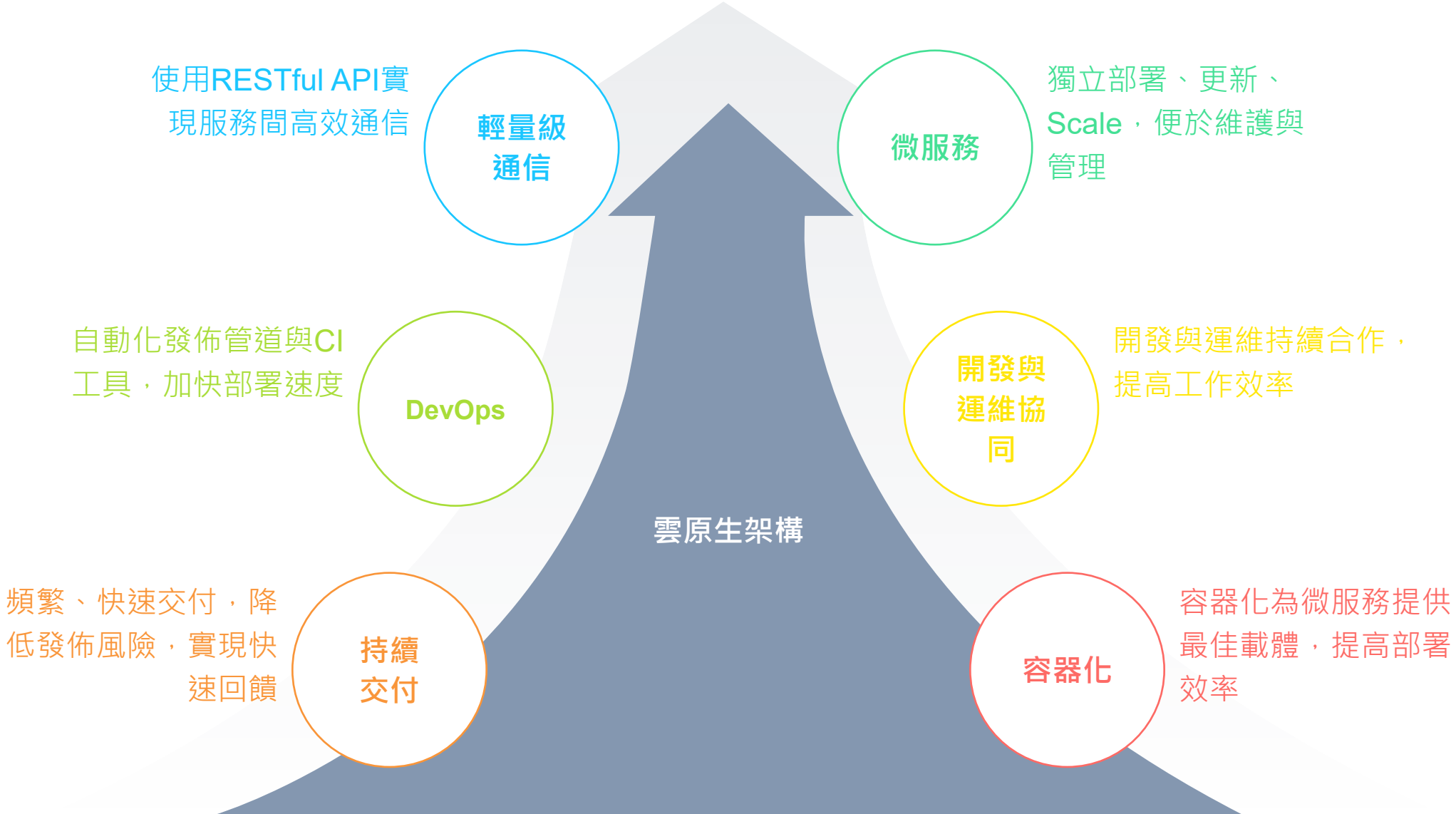
國泰金控

時代的眼淚 – 程式碼產生器



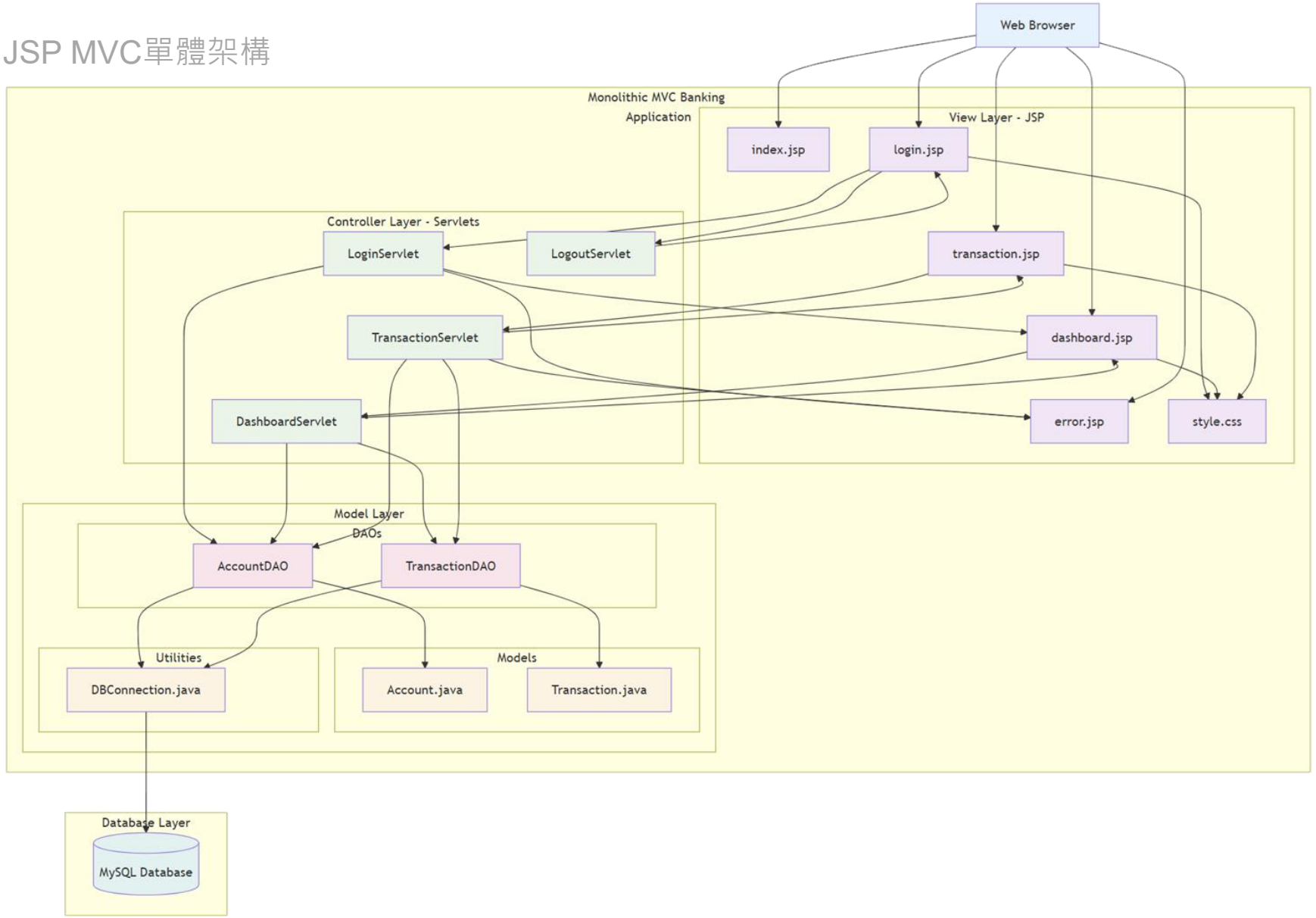
將一個JSP舊型單體系統重構為雲原生架構

CNCF：雲原生代表的技術包含微服務、容器化、不可變基礎設施、聲明式API、服務網格



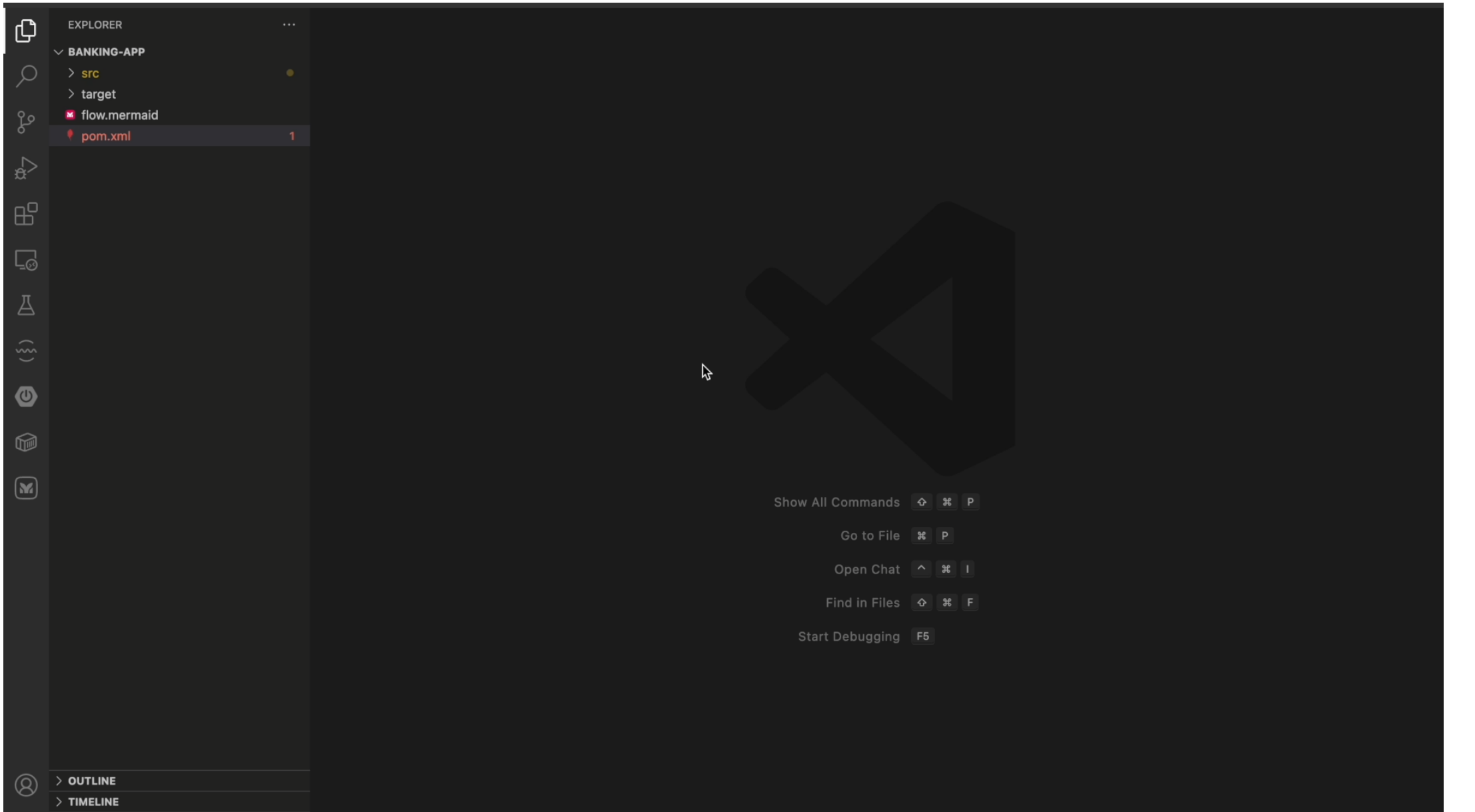
將一個JSP舊型單體系統重構為雲原生架構

使用Servlet & JSP MVC單體架構

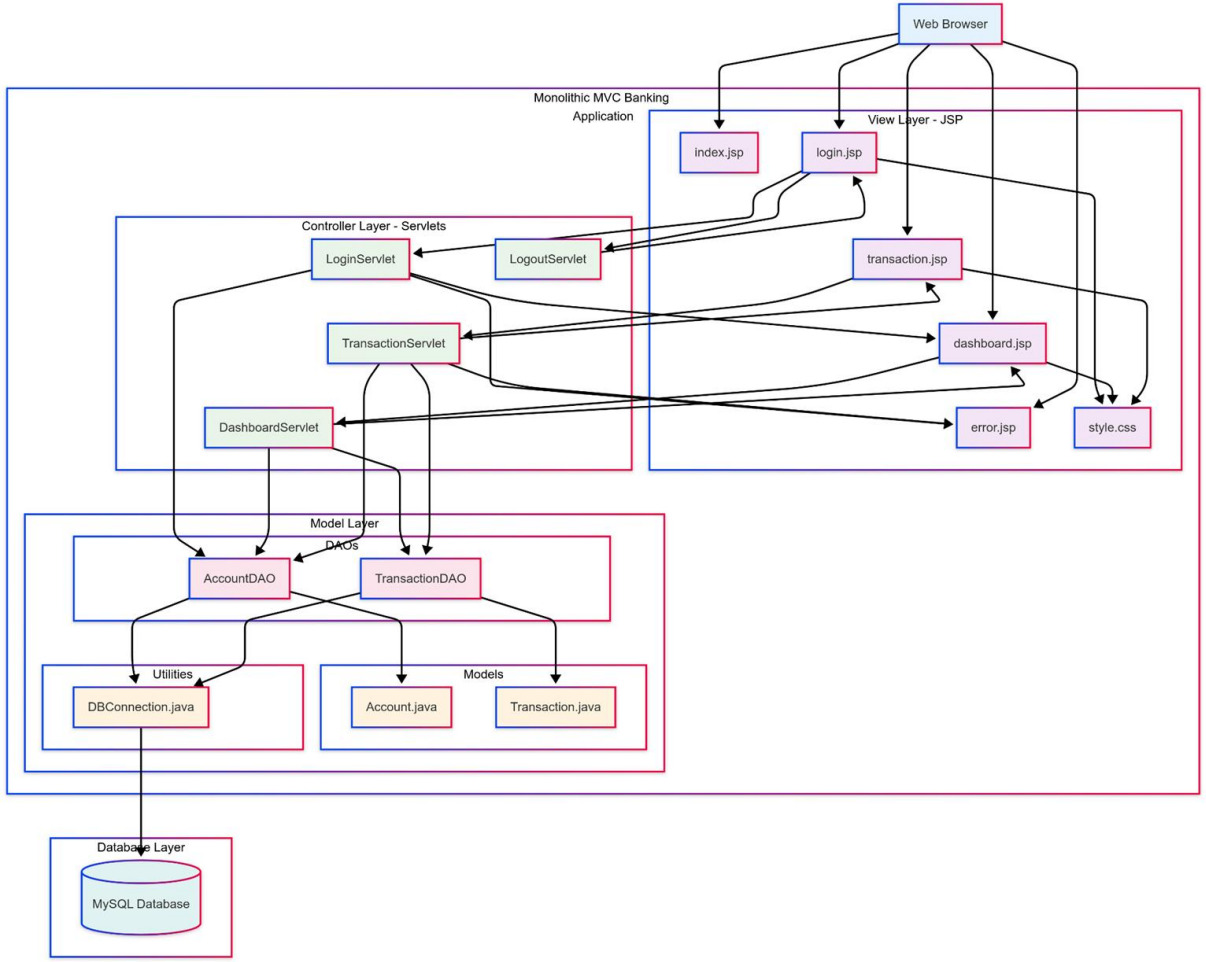


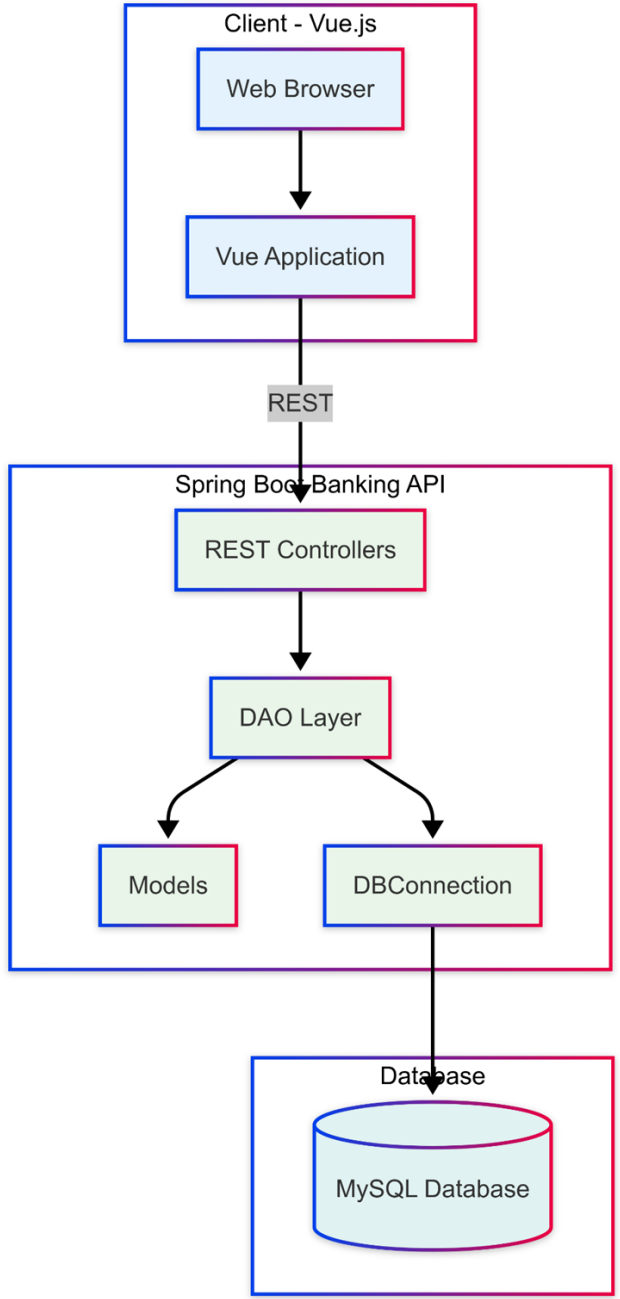
國泰金控

Demo: 現代化與重構策略 by GitHub Copilot



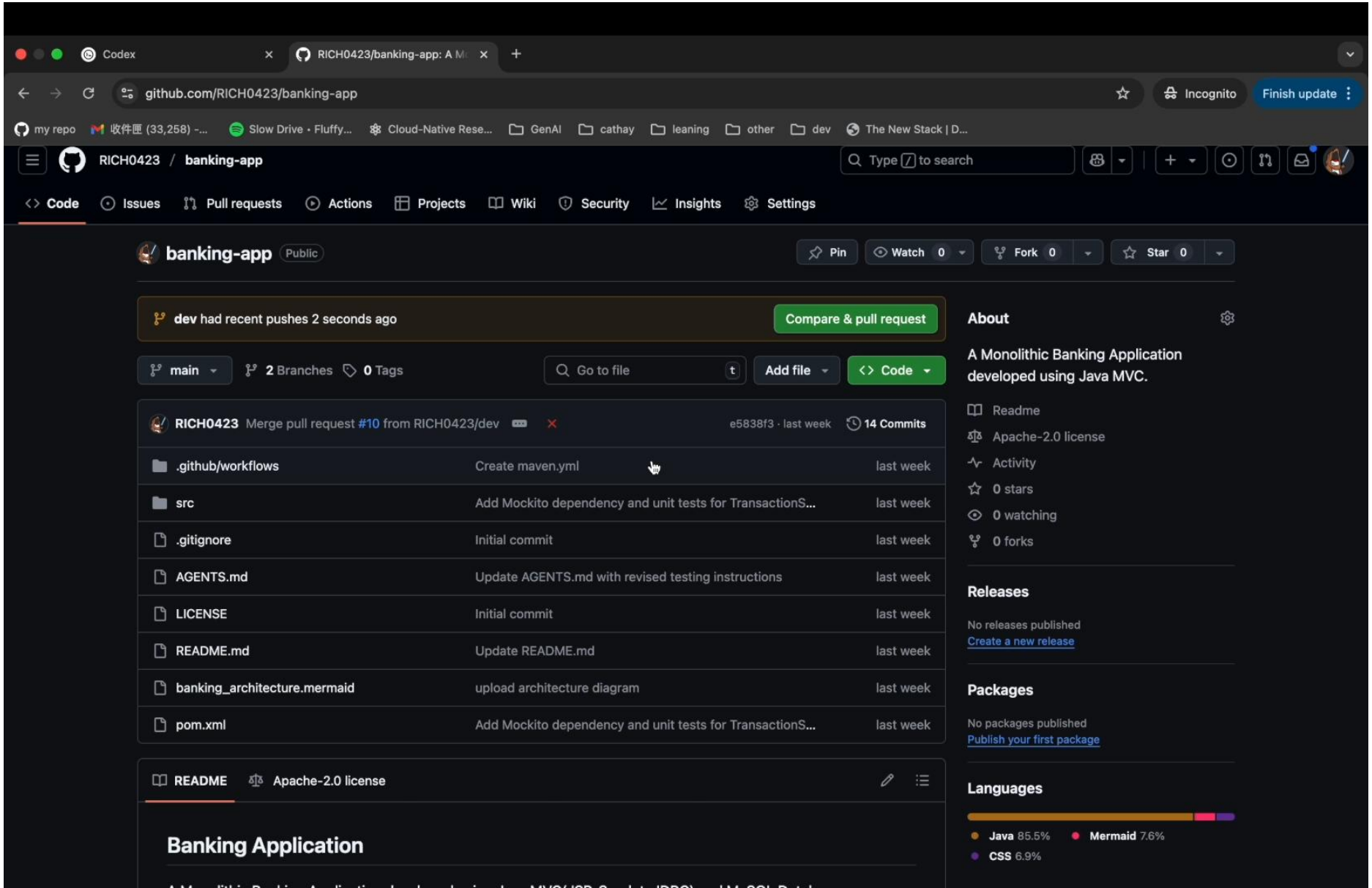
重構為前後端分離架構



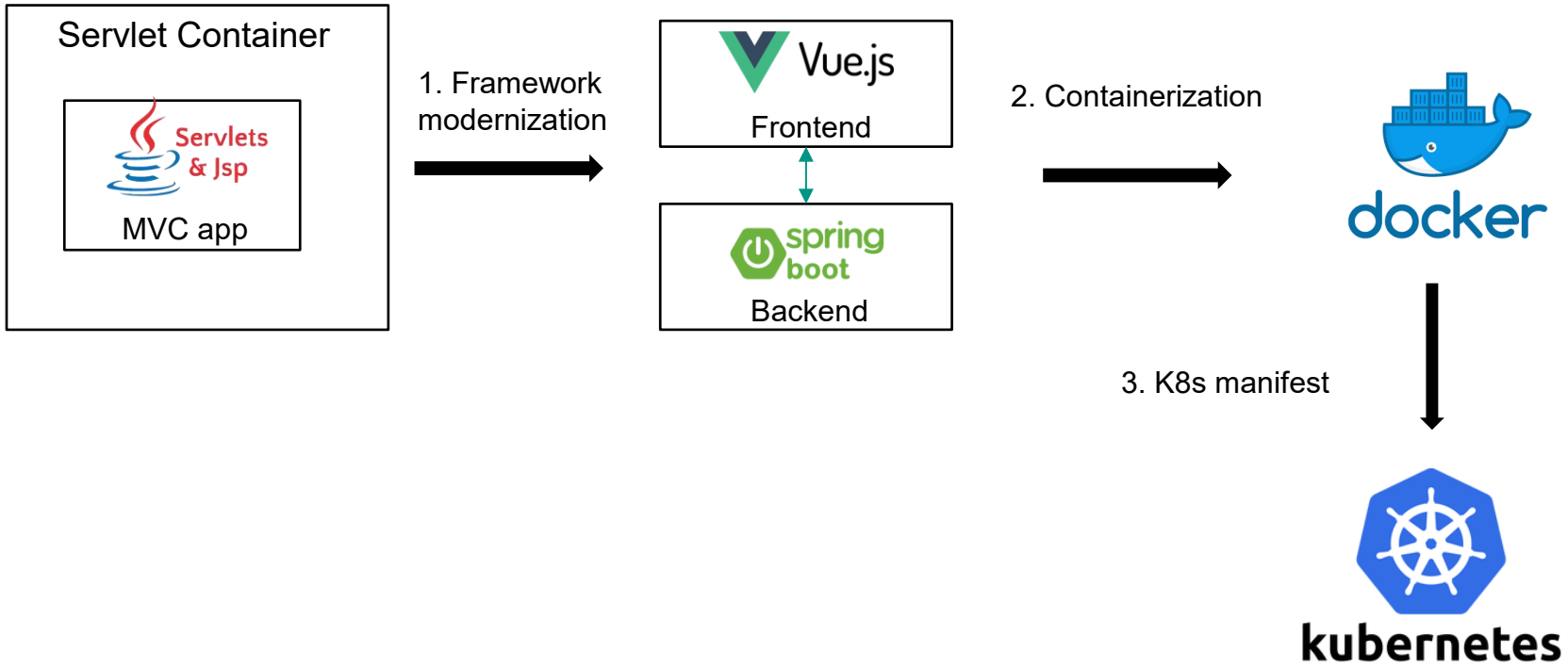


國泰金控

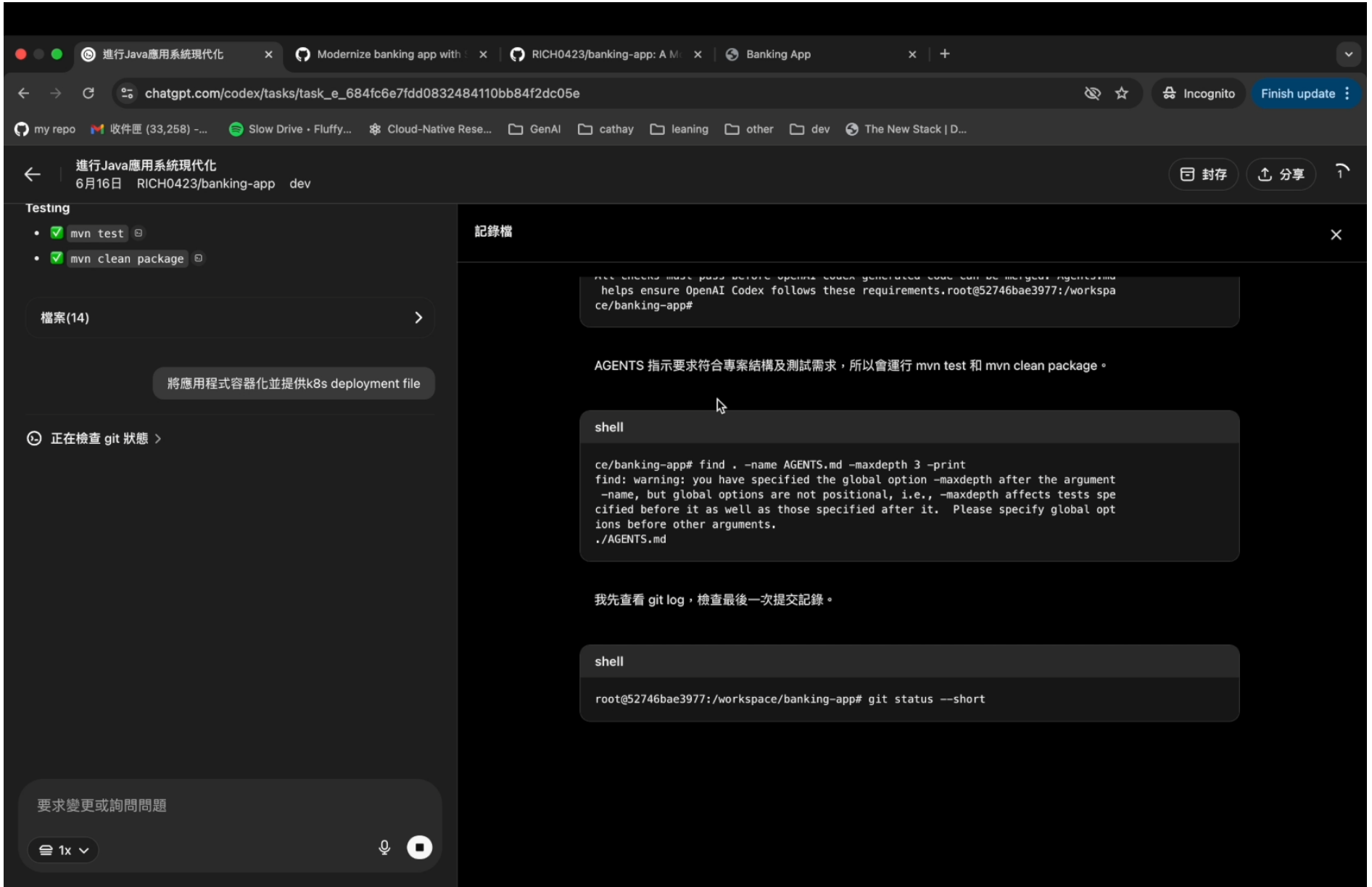
Demo: 重構 by OpenAI Codex



重構為容器化


Servlet Container
Servlets & Jsp
MVC app
1. Framework modernization
Vue.js
Frontend
spring boot
Backend
2. Containerization
docker
3. K8s manifest
kubernetes

Demo: 容器化 by OpenAI Codex

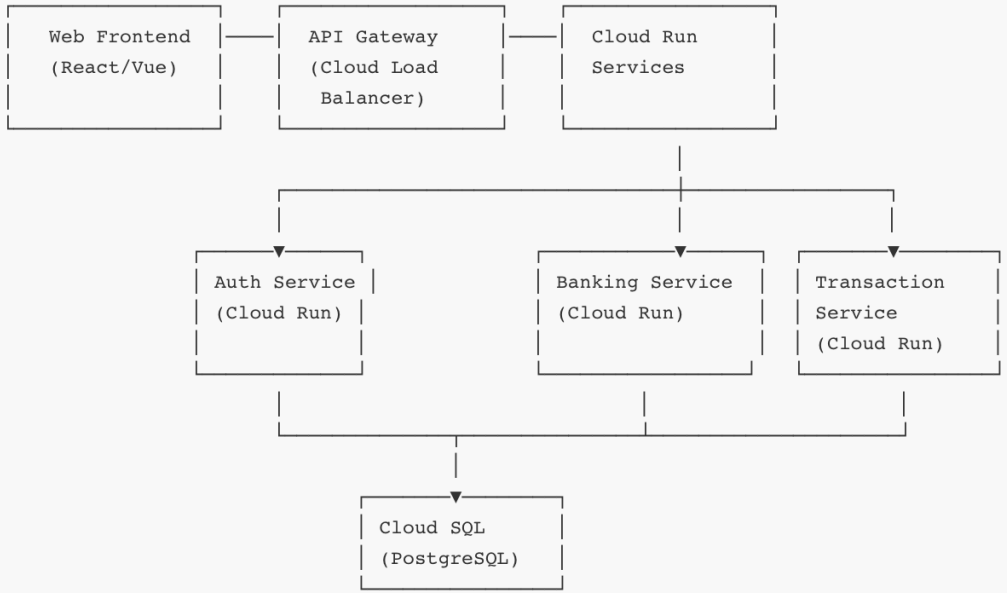


Live Demo:

雲原生架構 by Claude Code

🏗️ 雲原生架構設計

建議架構模式



微服務拆分建議

1️⃣ Authentication Service

- 功能: 用戶認證、JWT Token 管理
- 技術: Spring Boot + Spring Security
- 部署: Cloud Run (最小資源配置)

2️⃣ Account Service

- 功能: 帳戶管理、餘額查詢
- 技術: Spring Boot + JPA
- 部署: Cloud Run (自動擴展)

Vide Coding工具實為重構、應用系統現代化神器

回答了應用系統現代化 How 的問題

完成項目	LLM處理時間	使用工具
產生現代化與重構策略	4 mins	GitHub Copilot
Framework現代化與前後端分離	9 mins	OpenAI Codex
生成Unit tests與comments	3 mins	OpenAI Codex
產生系統架構圖	< 1 min	OpenAI Codex
容器化(產生Dockerfile與 K8s manifests)	4 mins	OpenAI Codex

Rakuten 以 Claude Code 將 1,250 萬行程式碼重構　七小時完成 準確率達 99.9%

流動日報
更新於 11小時前 • 發布於 2天前 • NewMobileLife

追蹤

不少人認為 AI 生成的 Code 無法處理大型項目，但日本科技巨擘 Rakuten 正以 Claude Code 重塑軟體開發流程，工程團隊藉由這項 AI 工具自動完成程式設計任務，大幅縮短上線時間。

國泰金控

Thank You !