

# 手把手邁向HIS 微服務之路 (上)

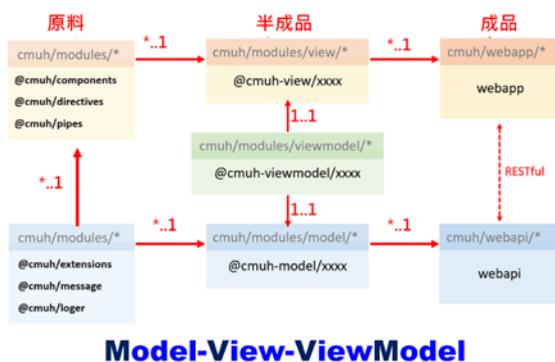
口述／孫培然 彙整／CIO編輯室

上期我們掌握怎麼把單體變成微服務的概念後，就可以進一步來操作，透過系統設計大綱的七個步驟，將單體變成微服務。但要特別注意的是，由於單體與微服務架構的設計思維完全不同，千萬不要為了微服務而微服務，反而可能發生更多的問題。

首先，我們要了解如何把資訊需求轉變成微服務架構的方法。〈圖1〉是 HIS 系統的全端架構概覽，我們把程式分為上半部的前端及下半部的後端程式，再用 MVVM 的概念再去抽離。

其中的原料，是指與企業邏輯沒有關係的設計，比如說西元年轉民國年、或者是一些資料庫連線的元件等。半成品就跟企業邏輯有關係了，比如說病人查詢畫面、檢驗報告查詢畫面，在很多系統都會出現的功能，都要開發成半成品，才能讓程式可以重複使用。比如說成品就是住院醫囑系統、護理照護系統、電子病歷查詢系統，只要嵌入需要的半成品，就可以直接使用半成品的功能，也

全端架構概覽 - 微型化隨插即用



〈圖1〉全端架構概覽

## View 的標準畫面模版



<圖2> View的標準畫面模板

就是所謂的隨插即用（Plug and Play）的概念。

下半部的後端也是一樣的概念，跟企業邏輯有關係的就是一個 Model，也就是沿用 MVC 的概念，組合起來的成品在前端叫做 WebApp，後端就 WebAPI，中間再透過 RESTful API 來溝通，值得一提的是，微服務能夠跨語言、跨平台，就是 RESTful API 的功勞。

為什麼這個架構要叫做 MVVM（Model-View-ViewModel）？這是因為前端跟後端的資料都是透過 ViewModel 來繫結綁定。比如說前端輸入某位病人的病歷號碼，就會要求後端查詢這個病人的基本資料，後端就要把姓名、出生年月日、電話號碼、E-Mail 等資料，裝成一個 ViewModel 傳給前端，這兩者共用的 class 就放在 ViewModel，這就是用 MVVM 的概念去做 HIS

架構的前、後端資料繫結層級。

此外，所有的前端畫面都是力求統一，所以標準畫面就會像<圖2>這樣的框架。這個框架只有裡面的「內容區」可以改變，其他如「功能選單」底下的子功能如字型切換，都是固定規則。

如果有任何關於系統或使用者的相關資訊，就會在「最新消息」中用數字來提示幾條最新消息，「程式集」就是代表有哪些程式可以使用。「資訊列」會顯示現在內容區裡面可識別唯一的相關資訊，比如說我現在開立某病人的醫囑時，其可識別唯一病人的相關基本資料就會顯示在資訊列。

工具列就是這個系統需要使用的相關按鈕，如：新增、刪除、存檔。輔助頁籤會在你需要相關的查詢報告或選取品項的索引。整個 HIS 系統的標準化畫面是剛性的，

才能鎖住架構，使用者介面也才能夠一致。使用者只要學會一個系統操作，以後再使用其他系統，就自然而然可以輕易上手使用。

## 微服務的系統設計大綱

在我們要開發一個系統時，會先跟使用者訪談需求後，再由敏捷軟體開發小組撰寫一份系統設計書，表列產品負責人（Product Owner）、敏捷軟體開發教練（Scrum Master）及敏捷軟體開發小組（Scrum Team）。敏捷軟體開發小組可會有很多人，排第一個的名字代表這個系統設計書的主要撰寫者。

系統設計書裡面的大綱有七大重點如<圖3>。

## 系統設計大綱

1. UI & Event Binding
2. ViewModel
3. View Service
4. WebAPI Controller
5. Model Service
6. Stored Procedure
7. Interactive

<圖3> 系統設計大綱

下一期，我們將就這七個重點逐一解說。