掌握框架設計要領

By 高煥堂

框架設計要領不在於技術，而在於心境。
因為框架是個送人的禮物，要表現愛心。

1. 從享受禮物到贈送禮物

茲回顧過去 20 年來的軟件發展經驗了。其中有兩項重要的事蹟：

- 1980 年代後期，CORBA 是一項物件導向的服務標準 API，實現此項標準的系統中，最著名的商業中間層軟件就是 Orbix 系統。然而，在系統架構上，API 是一種約束，不是一種禮物，不能用來嘉惠予 AP 開發者。導致 CORBA 和 Orbix 系統架構無法支撐理想的商業模式，而終告消失匿跡。
- 1990 年代中後期，繼 CORBA 之後的是 Microsoft 公司推出 COM/DCOM 系統架構，雖然提供了當時先進的物件導向(Object-Oriented)的 API，但還是 API，仍然是一種約束，不是一種禮物，不能用來嘉惠予 AP 開發者。與 CORBA 和 Orbix 一樣的系統架構，一樣無法支撐理想的商業模式，也終告消失匿跡。

後來，IT 業界逐漸發現：API 可用來框住應用程序(AP)，如同一把利劍；若要獲得開發者的青睞，利劍必須搭配麵包，就像釣魚勾必須搭配魚餌，才能吸引魚群。於是，Microsoft 改變觀點，把焦點放在麵包上，發現物件導向技術裡的抽象類別(Abstract Class)及其提供的預設函數(Default Function)以及其他具體類別，所整合而成的框架(Framework)正是一項極具誘惑力的魚餌。

此外，由框架所提供的主動型 API，也能發揮巨大的控制力。因之，Microsoft 於 2001 推出.NET 框架來取代 COM/DCOM，由於.NET 框架融合了麵包與利劍，既能嘉惠廣大的開發者，又能有效框住眾多的應用程序。於是，.NET 框架成爲 Microsoft 贈送給廣大的開發者的最佳禮物，表達了 Microsoft 對全球廣大第三方開發者關懷和愛心，讓他們因.NET 而受惠。

到了 2007 年，Google 也依樣畫葫蘆，買來 Android 框架，當成禮物贈送給全球的手機硬件廠商，也贈送給全球廣大的 AP 開發者。由於 Android 框架「禮物」嘉惠予硬件廠商，所以硬件廠商也是受惠者，因而大力支持 Android，也讓 Android 聲勢扶搖直上。

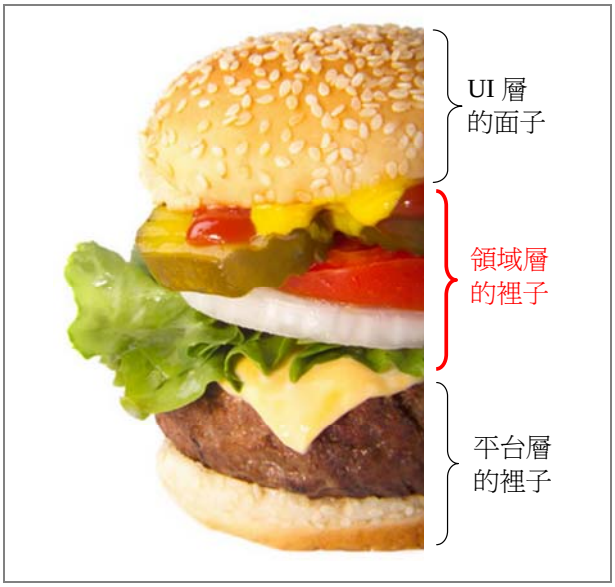
老子說：「聖人無積，既以爲人己愈有，既以予人己愈多。」Google 拿 Android 框架禮物來送人，所以 Google 會是最大的獲利者。例如，Google 公司 CEO 在接受華爾街日報訪問時，他提到 Android 將帶給 Google 公司每年 100 億美金的收益，如下報導：
(http://androinica.com/2010/07/28/android-could-make-google-10-billion-a-year-and-that-is-a-complete-longshot/)：

“Here is howa CEO talks to the Wall Street Journal when he wants his company’s stock pumpedup. Today, Eric Schmidt was quoted saying, “If we have a billion people using Android,you think we can’t make money from that? ”This was inresponse to his statement and belief that Android could potentially make $10 billion dollars ayear in advertising revenue for Google.”

http://km.funddj.com/kmdj/News/NewsViewer.aspx?a=e2d6a048-85b9-4c97-8ce0-2f7faf25b283
(據華爾街日報報導，谷歌(Google)執行長 Eric Schmidt 預計，經由手機廣告與應用服務的下載的服務，Android 作業平台每年可帶來 100 億美元以上的營收。)

2. 我們能貢獻些什麼樣的禮物呢?

那麼，我們又如何找到自己貢獻的空間呢? 如果拿一顆漢堡(Hamburger)來做比喻，則『UI層』面子就像漢寶上層麵包和芝麻，而『平台層』裡子就像漢堡下層的麵包和牛肉。如下圖：



(此圖摘自: http://www.caloriesloss.com/pics/HamburgerCalories.jpg)

如上圖所示，一顆完整美味的漢堡，除了麵包、牛肉和芝麻之外，還需要新鮮的蔬菜、起司和醬料等。也就是目前全球廣大AP開發者最需要的是各行各業(即各領域)的內容(即禮物本身)。如果我們把注意力放在領域內容(Domain-Specific Contents)上，並且設計精緻的禮盒(即軟件框架)，將之包裝起來，成爲形形色色的特殊領域框架(Domain-Specific Framework)，並且附隨在Android手機或Android電視機裡，大量贈送給Android Market上的全球數以萬計AP開發者，表達了我們對全球廣大第三方開發者關懷和愛心，讓他們都能享受美味可口的禮物。

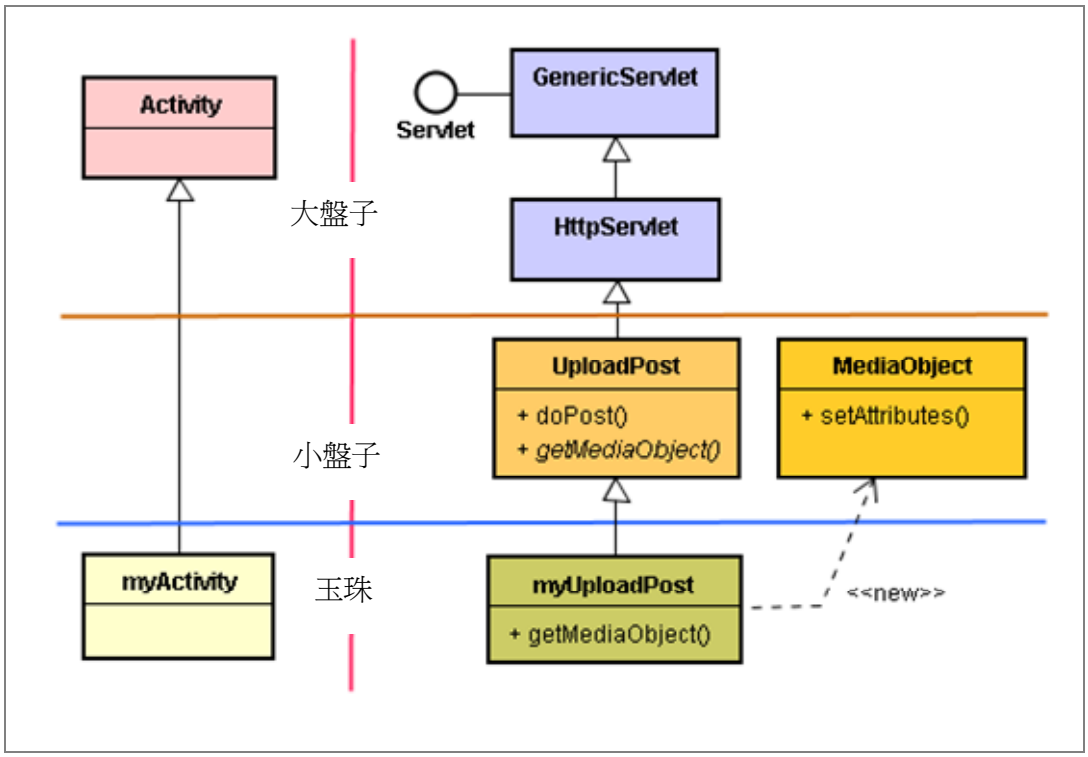
於是，手機或電視機廠商獲得『平台層』裡子；我們則獲得『領域層』裡子，卻給予Google極大的面子，則裡子、面子兼具地嘉惠予全球廣大的開發者，基於這利己利人的作爲，大家都是受惠者。

3. 領域框架與平台框架之結合：

曹操的挟天子(大盤子)以令诸侯(玉珠)。

平台框架像大盤子
領域框架像小盤子
玉珠皆落小盤大盤

本來玉珠直接落到大盤子裡。現在，小盤子疊在大盤子上，而玉珠則落在小盤子上。如下圖所示：



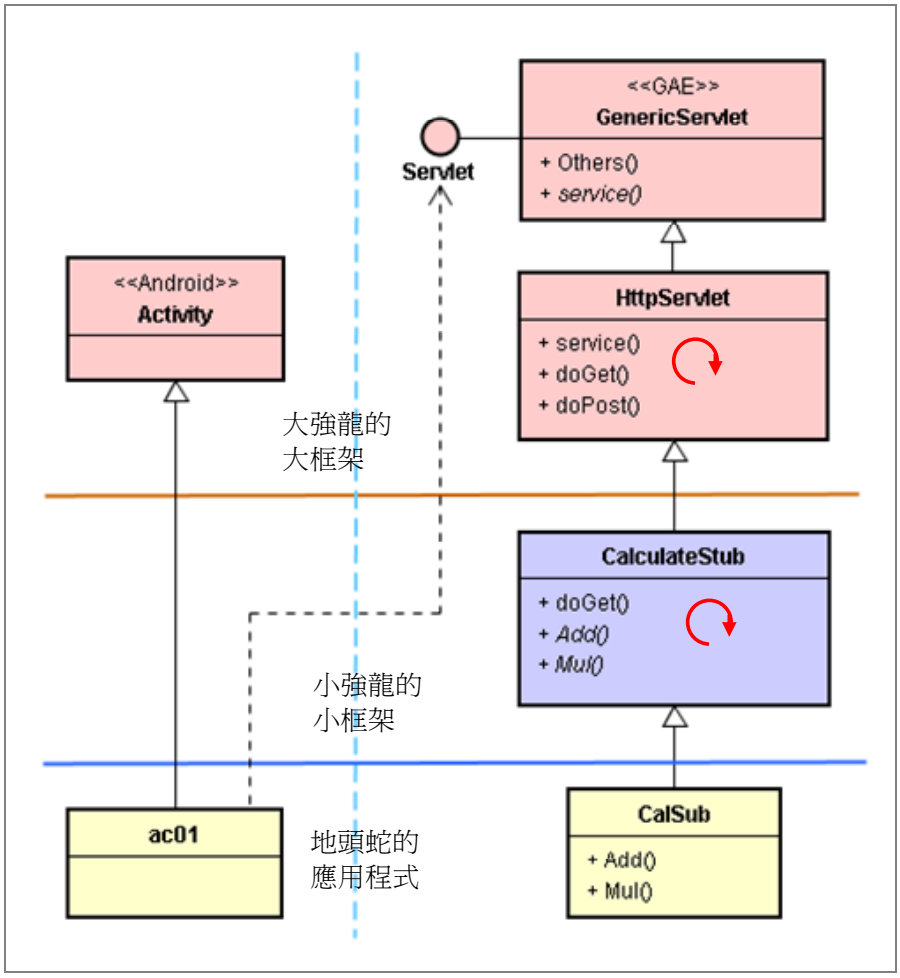
大框架提供的 API 包含 doPost()函數(如同漢憲帝的聖旨)，小框架則將其轉換成爲自己的 API：getMediaObject()函數(曹操的命令)。所以是挾天子以令诸侯。

小框架就扮演著曹操角色，實現了
挾天子(大盤子)以令诸侯(玉珠)。

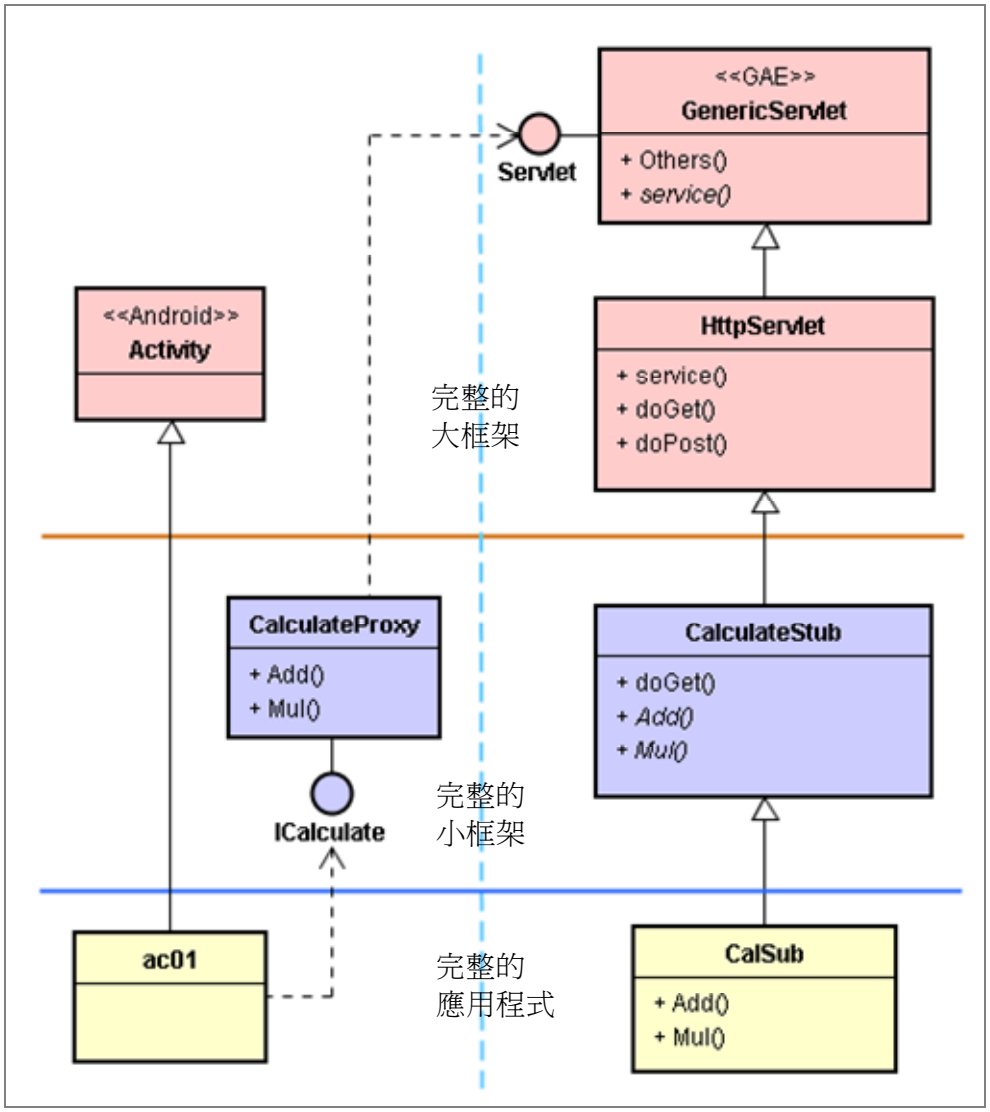
Android的myActivity類別 與myUploadPost類別之溝通，必須經由大框架和小框架。所以整體系統的控制點在於大框架與小框架裡。至於大、小框架何者擁有較大控制權，就得視其框架API和類別內容而定了。

就商業利益看，小盤子獲利較大。
其實曹操的金銀財寶比漢憲帝多。

如下圖：



在撰寫 ac01 類別時，仍然使用大框架的 Servlet 介面，AP 仍然依賴於大框架的介面。如果小強龍想完全主導地頭蛇的話，就必須想辦法去避免 ac01 類別直接使用大框架的介面。於是改善系統架構如下：



其中，CalSub 類別、CalcuateProxy 類和 ICalculate 介面整合起來成爲小框架，讓 ac01 類別和 CalSub 類別都使用到小框架的介面(如 Add()和 Mul()函數)，而不直接使用大框架的介面(如 Servlet 介面)。就完整實現了「挾天子以令諸侯」的目的。

4. 框架設計範例：雲框架裡的小框架

- 從雲看端的觀點

  茲拿公有雲平台爲例，TM 設計模式提供兩個 API 來支援公有雲，基於 TM 模式的框架讓公有雲成爲系統強龍。如下圖：

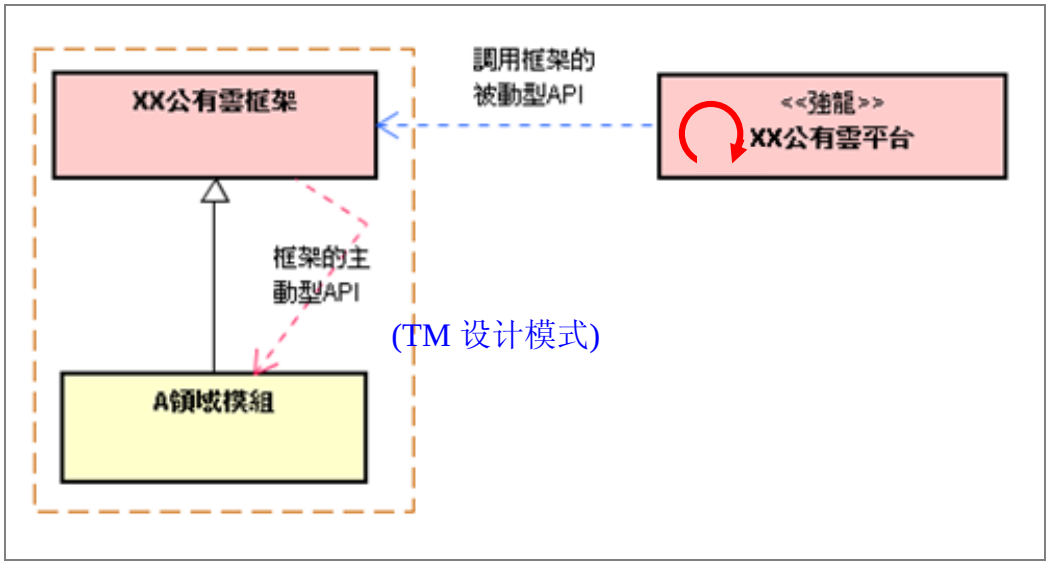


圖 4.1 雲強龍的系統架構(型式一)

稍微調整一下上圖 4.1，得到下圖 4.2，這是常見的層級(Layered)結構。

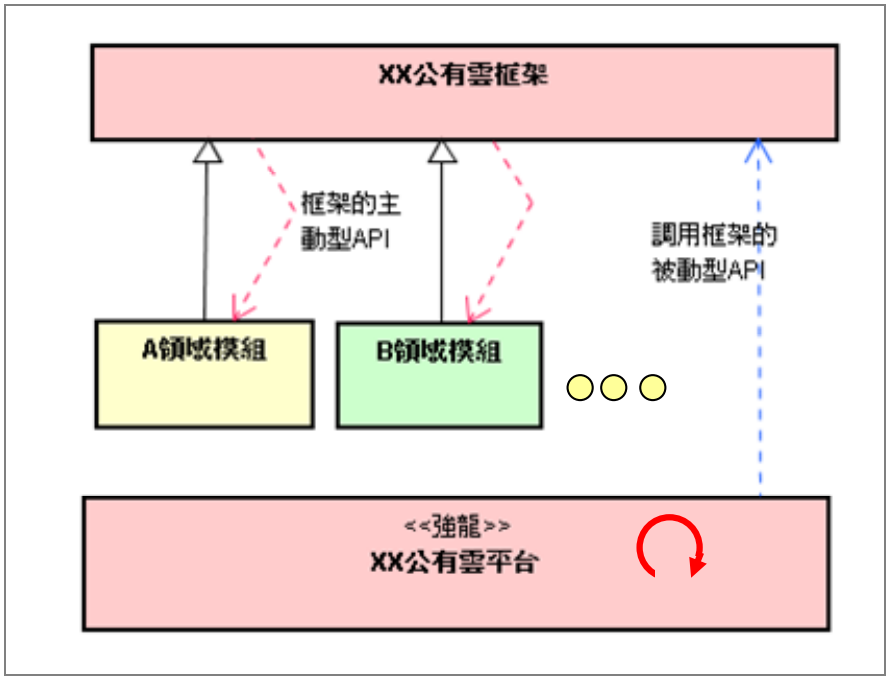


圖 4.2 雲強龍的系統架構(型式二)

這系統架構支撐 XX 公有雲的雲主，讓其穩居強龍地位。由於框架擁有主動型 API，所以雲主敢大膽地讓外人(地頭蛇)進來撰寫領域模組軟件。於是，在公有雲平台(含框架)裡，會有含有眾多外人(例如圖 4.2 中的 A、B、C 等地頭蛇)進來撰寫各自的領域模組。

- 從端看雲的觀點

公有雲像百貨公司，而領域模組像專櫃店面。領域雲際平台就像連鎖專櫃企業。就雲際平台而言，它也希望自己成爲強龍。它除了撰寫雲際平台軟件之外，還在各公有雲裡撰寫領域模組(如同開設專櫃)。如下圖所示：

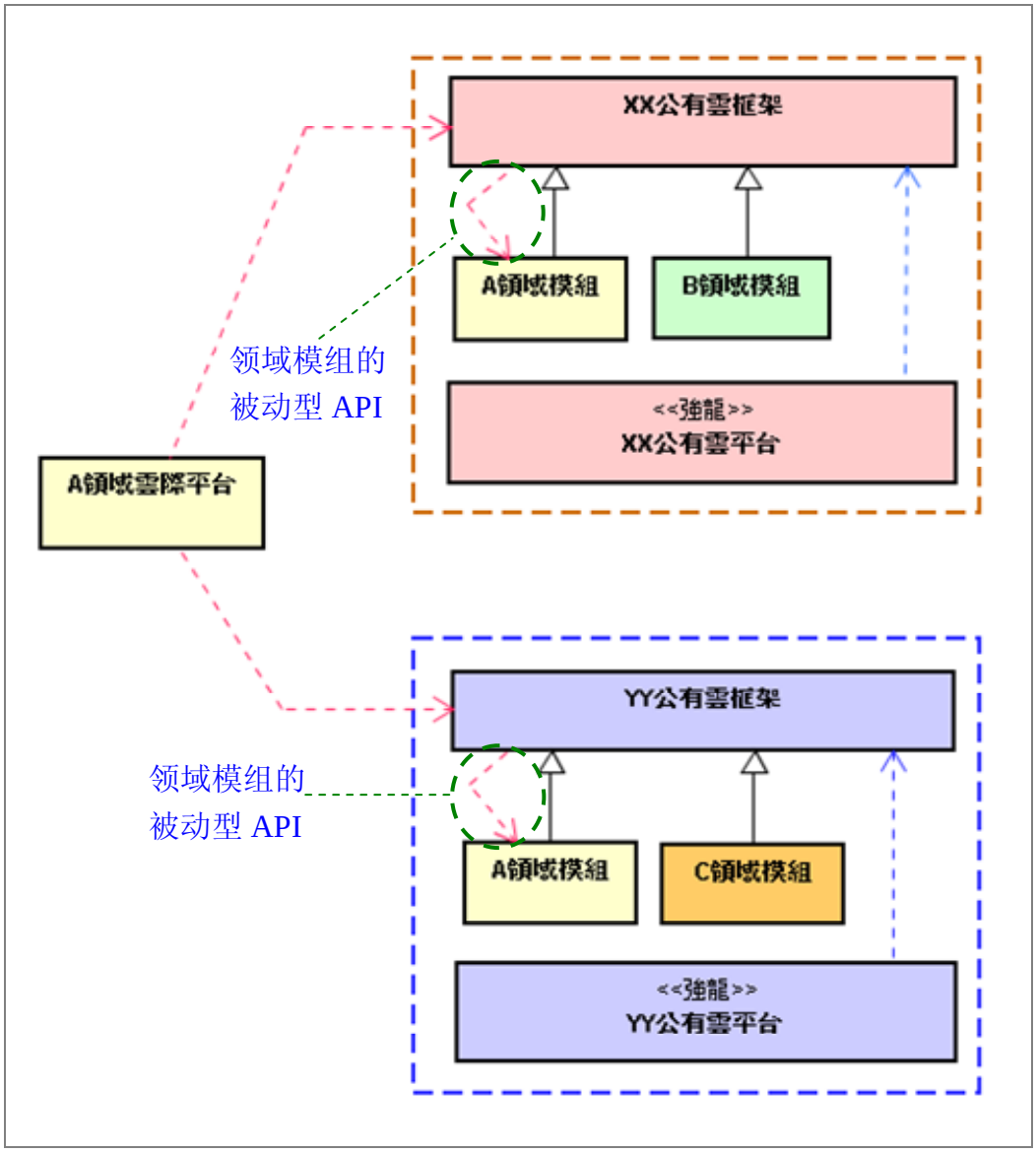


圖 4.3 端(領域雲際平台)期待成爲強龍

雖然 A 領域雲際平台期望自己是強龍，但是從上圖可以看出，其系統架構並不能支撐 A 連鎖專櫃成爲強龍。其原因是：在公有雲裡的「A 領域模組」只提供被動型 API 給公有雲框架來調用(實現了公有雲框架的主動型 API)，因而扮演地頭蛇的角色。那麼，如何才能調整上述的系統架構，才能支撐 A 領域雲際平台的強龍地位呢? 方法是：在公有雲裡創造自己的主動型 API，如下圖：

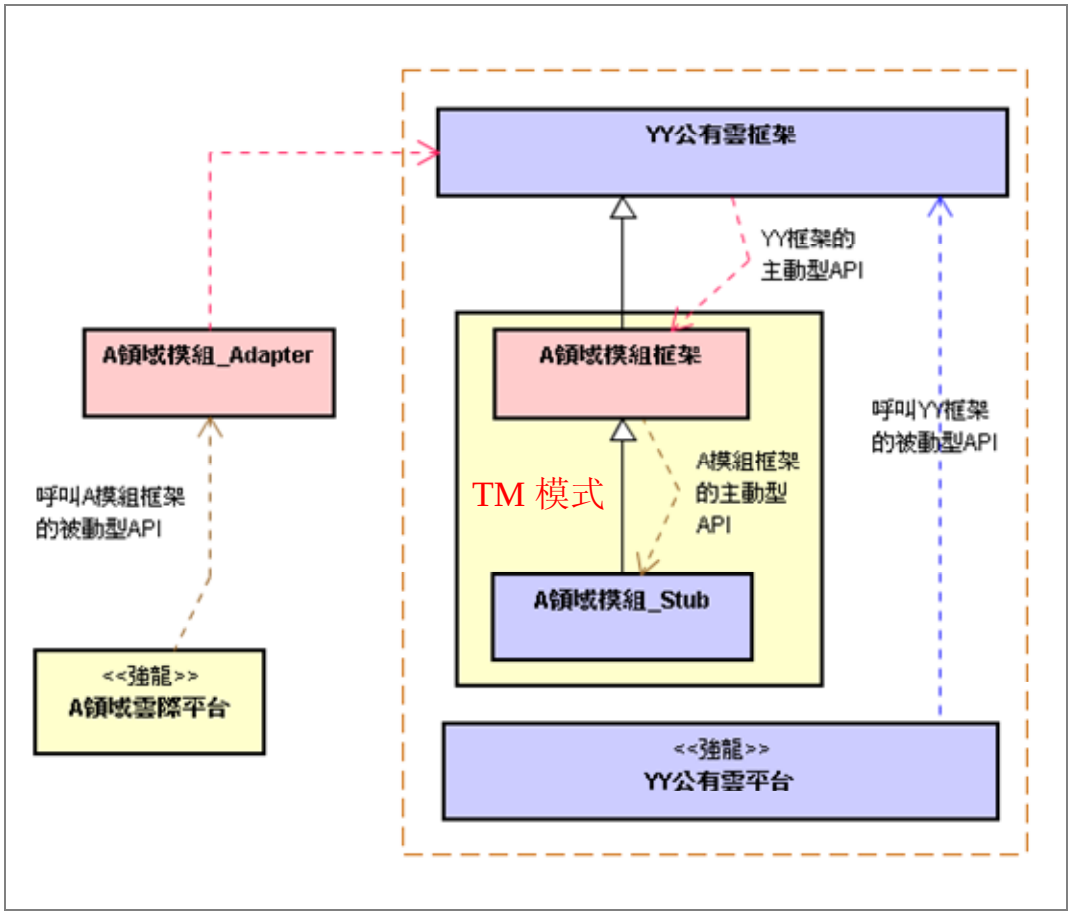


圖 4.4　從端看雲的強龍系統架構

起初，可能必須由 A 領域雲際平台負責撰寫公有雲裡的「A 領域模組_Stub」，然而隨著 A 領域雲平台所支撐的商業強龍地位日益形成，可能轉變爲 A 領域雲際平台與 YY 公有雲平台合作一起開發「A 領域模組_Stub」，或者由 YY 公有雲平台負責撰寫「A 領域模組_Stub」。

爲了讓 A 領域雲際平台不直接使用 YY 框架的被動型 API(因爲這個 API 是由 YY 公有雲平台所定義的，其定義權掌握在 YY 公有雲平台手中，而不是由 A 領域雲際平台所定義的)，就增添一個「A 領域模組_Adapter」。爲了說明上圖的經濟效益，還是拿曹操的「挾天子以令諸侯」來做比喻最爲傳神了。上圖的「A 領域模組_Adapter」和「A 領域模組框架」成爲曹操角色，將 YY 雲平台的 API 封裝起來，達到「挾天子」的意境。由於「A 領域模組框架」提供主動型 API 來調用 A 領域模組_Stub，逐漸形成「令諸侯」的意境。◆