

Chapter 13

元件和相依性 管理 Part II

Winnie

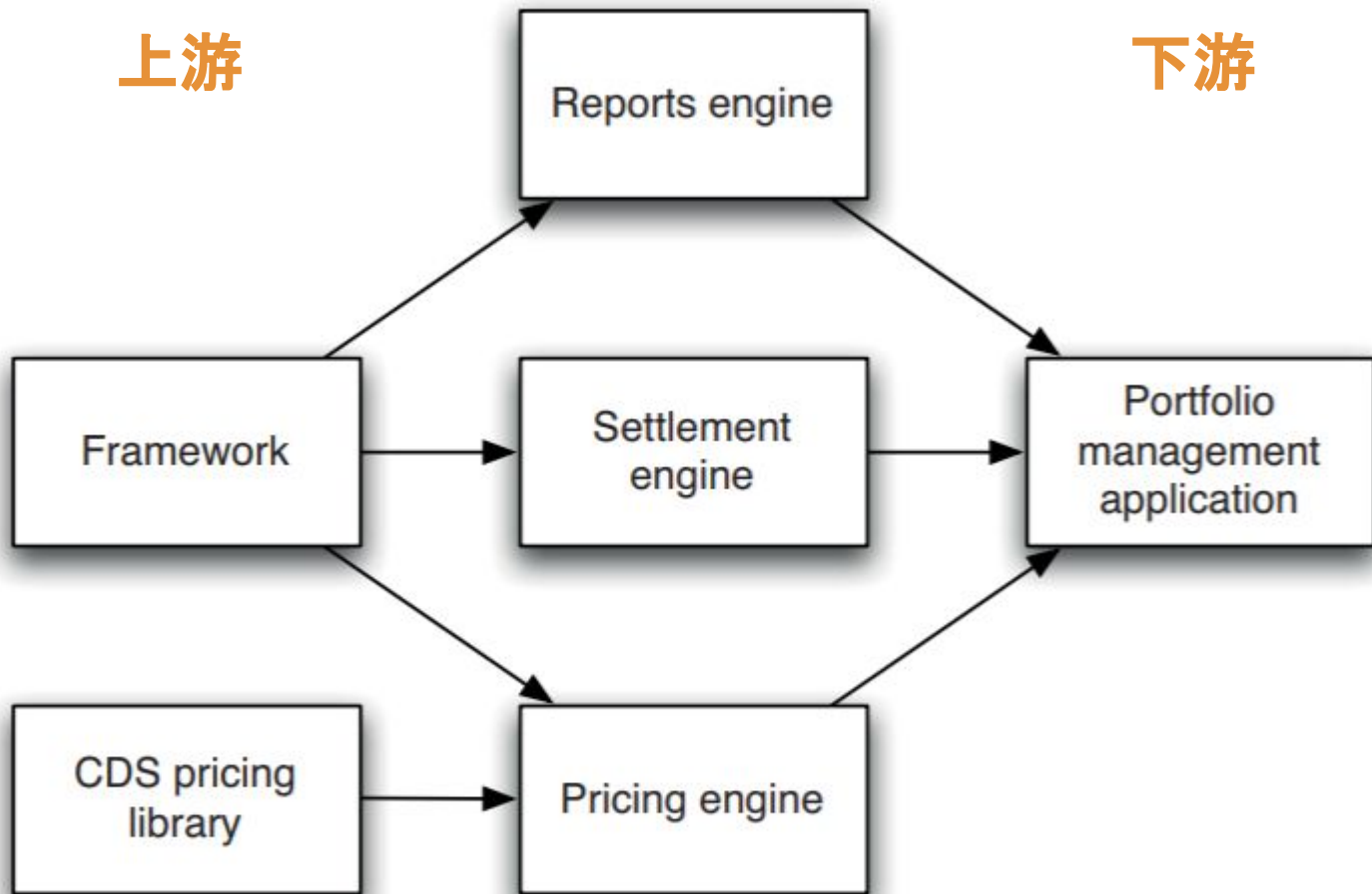
管理相依關係圖

- 相依性的版本管理很重要
 - 包含函式庫和元件的相依性
- 建立相依關係圖
 - DAG (Directed Acyclic Graph)

相依圖

上游

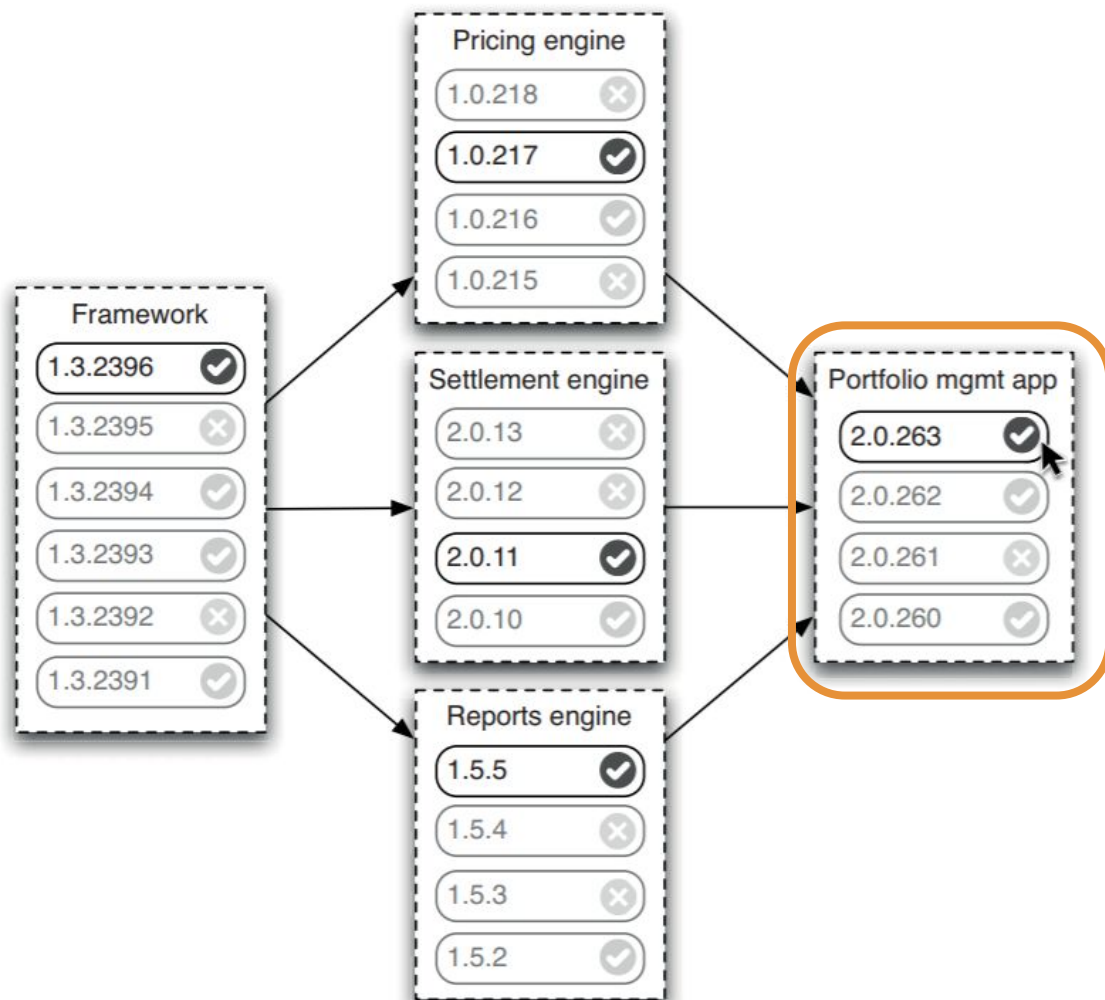
下游



相依場景

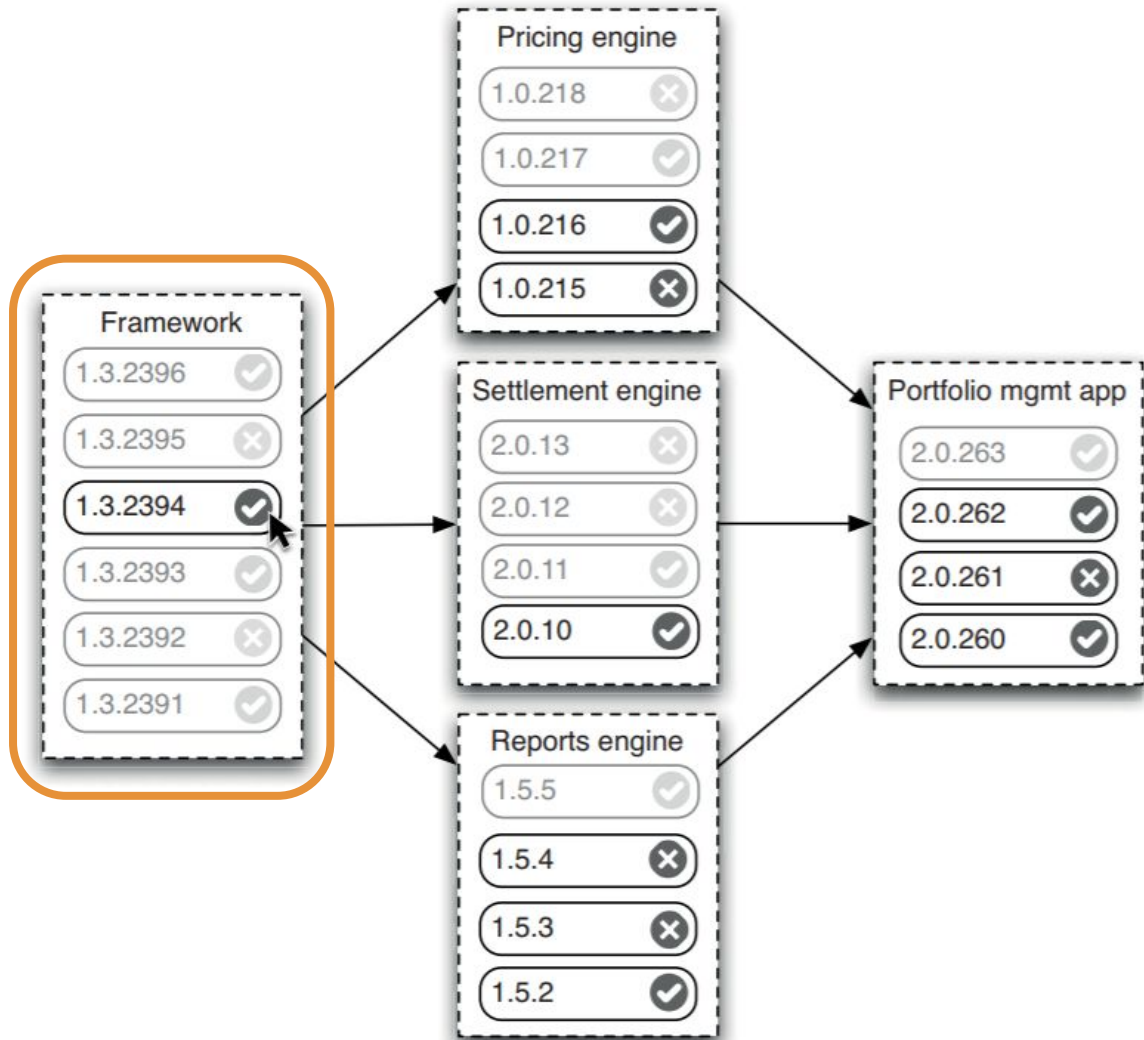
- 最重要的限制是建置 投資組合應用程式 (Portfolio management application) 時應該只依存於架構 (Framework) 的版本**之一**
- 菱形相依性
 - **建置時的相依性地獄!**

上游相依圖例

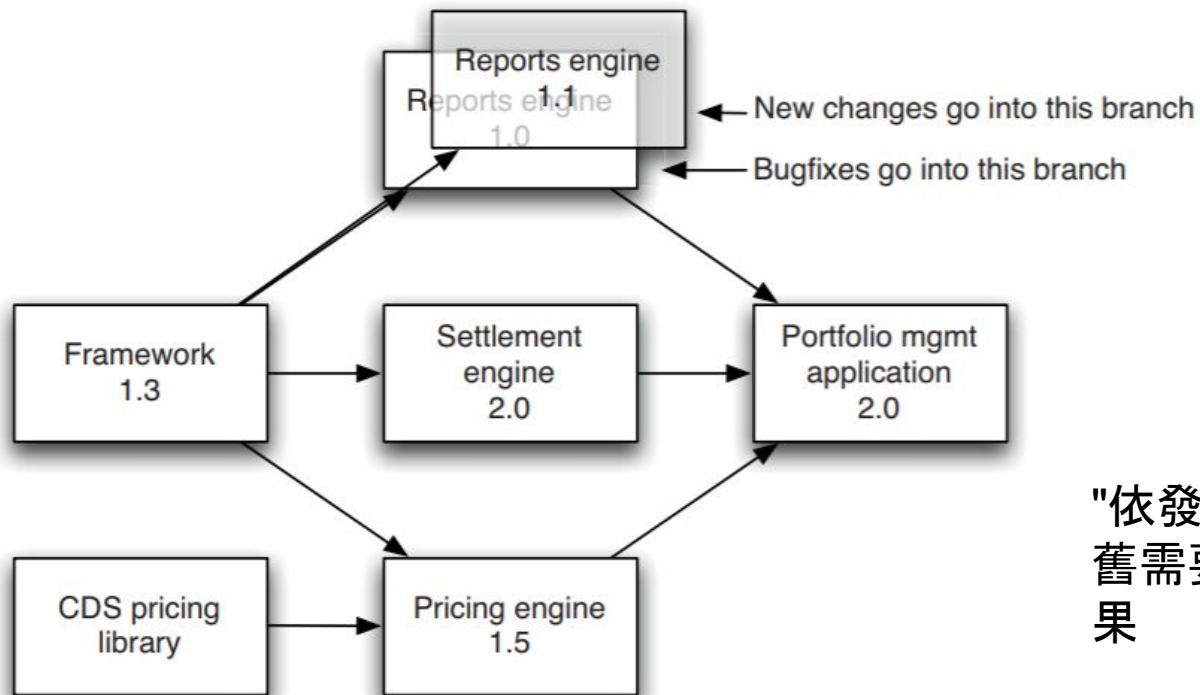


下游相依圖例

哪些下游元件會使用
Framework 1.3.2394
來進行建置



元件的分支



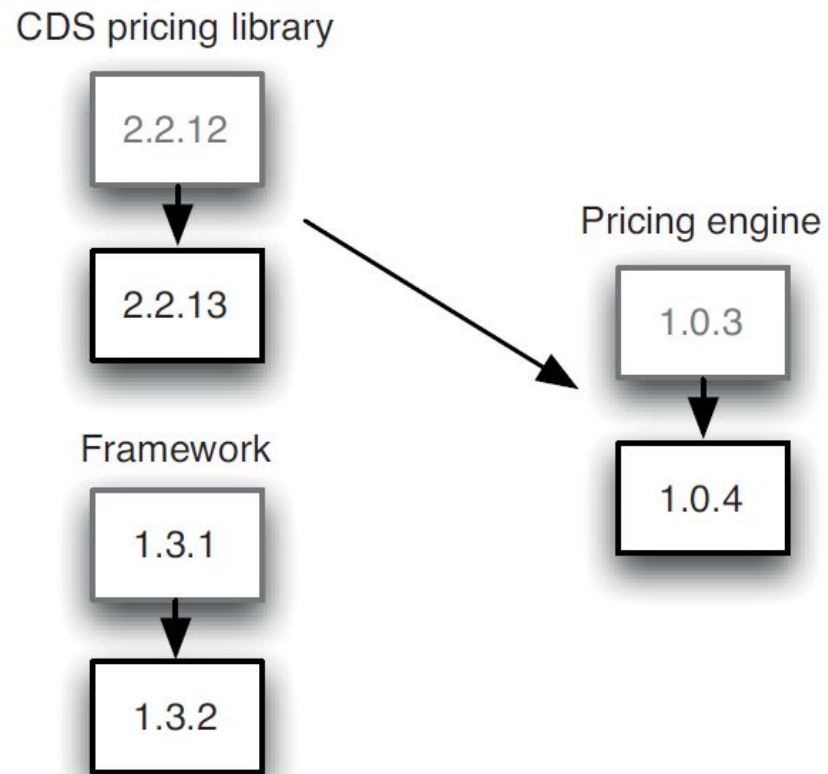
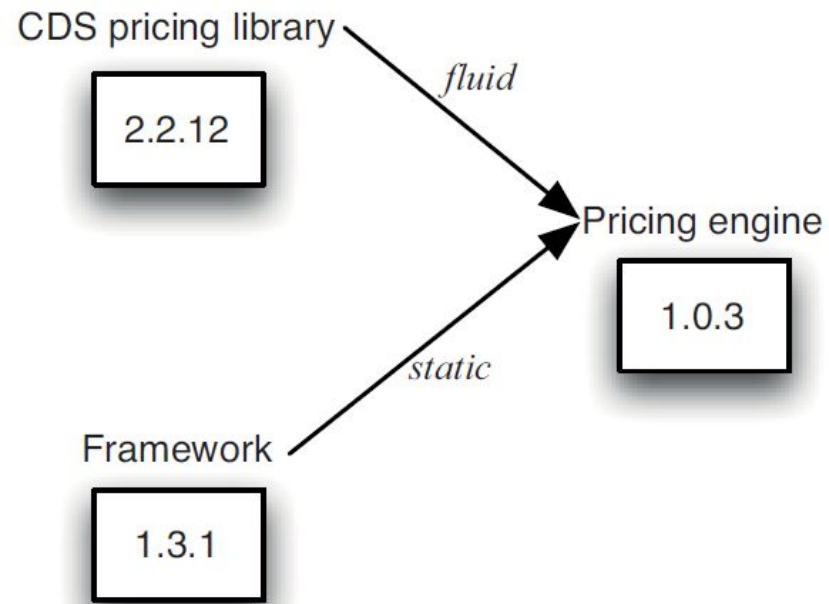
"依發佈建立分支"模式仍舊需要承擔**延後整合**的後果

何時要觸發建置?

- 張力
 - 最好能保持與上游相依性最新的版本一致, 以確保取得最新特性和已修復缺陷
 - 整合最新版本會有一定的成本
- 對相依性之新版本 **信任度** 有多高
 - 對元件變更的掌控性, 可視性和影響力越少, 對它的信任越少

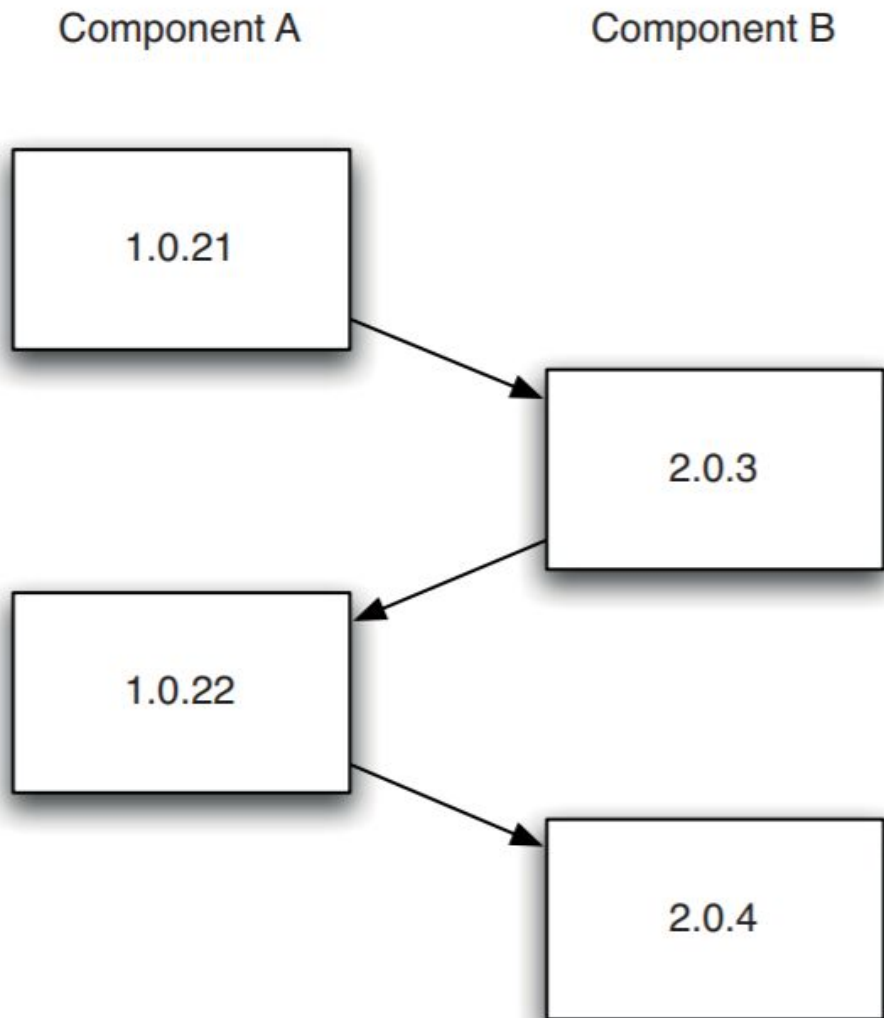
謹慎樂觀主義

- Alex Chaffee
- 觸發類型
 - 靜止 (static)
 - 不會觸發新的建置
 - 活躍 (fluid)
 - 一定會觸發新的建置
 - 慎用 (guarded)



迴圈相依性

- 建置階梯



管理 Binary

- 產出物儲存庫不應該包含無法重現的產物
- 不建議將產出物簽入版本控制
- 保存每個產出物的雜湊值, 以驗證正確性

產出物儲存庫

- 最簡單的是目錄結構
 - 放在 RAID 或 SAN
 - 應該能將 Binary 關聯到產生該檔案的原始程式碼版本
 - 為每個流水線建一個目錄
 - 為每個建置編號建立一個子目錄
- Web service
 - 建議使用市面上的產品或open source

結合 部署流水線 與 產出物儲存庫

1. 將建置期間的產出物放到產出物儲存庫
2. 之後需要時能取出它
 - a. 取出做單元測試, 驗收測試
 - b. 將產生的測試報告放入產出物儲存庫
 - c. 取出部署到UAT環境進行手動測試
 - d. 發佈階段：取出部署到生產環境中
- 如何存取
 - 共用檔案系統
 - 管理工具：Nexus 或 Artifactory

小結

- 確保團隊能儘早得到程式碼修改後產生的影響
 - 將每次修改都分解成小型且增量式的步驟, 並分成許多小步驟簽入主線
 - 將應用程式分解成多個元件
- 直到應用程式夠大時, 才需要分別建置元件