

# BitGin

- BitGin
  - Problems
  - Solutions
    - Solution 1
    - Solution 2
    - Solution 3

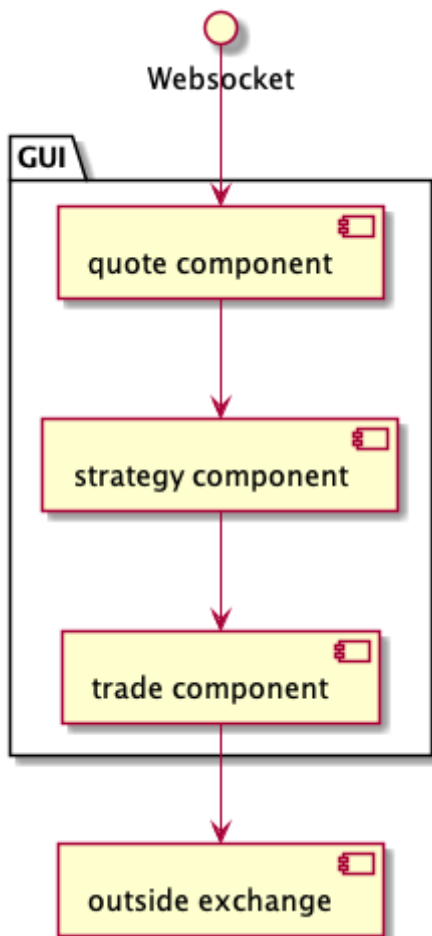
## Problems

1. 請問如果現在你要設計一個量化策略的程式，程式會去擷取第三方的價格資訊，做出買入賣出的決策，請問如果你現在要做一個單機版的自動化交易你會怎麼設計你的架構。為什麼？
2. 承上題，如果要增加一個回測系統，測試設計的策略年化報酬率，你會如何設計。為什麼？
3. 承上題，如果要將上述的系統從單機版變成一個公開的服務，你會怎麼設計及建構整個服務。為什麼

## Solutions

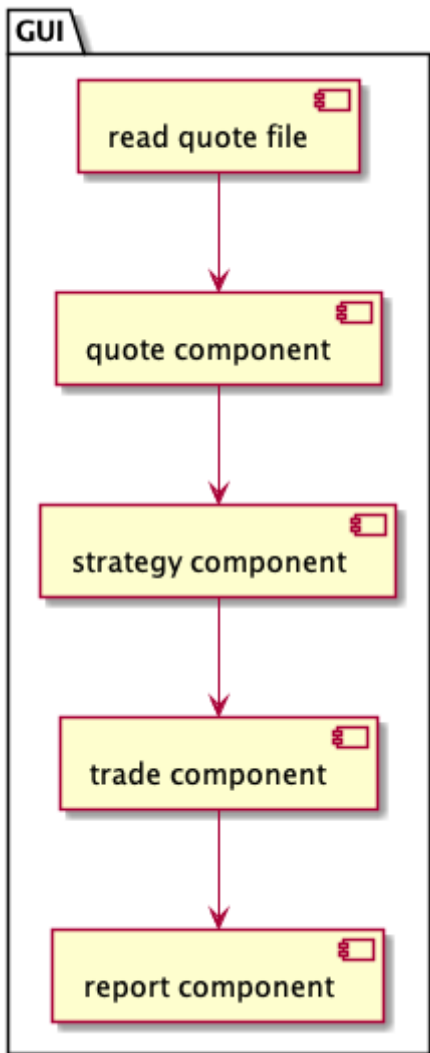
以下設計都基於 micro-service 為基礎來設計。我會先使用 Domain-Driven Design(DDD) 的方式規劃整個系統的架構。

### Solution 1



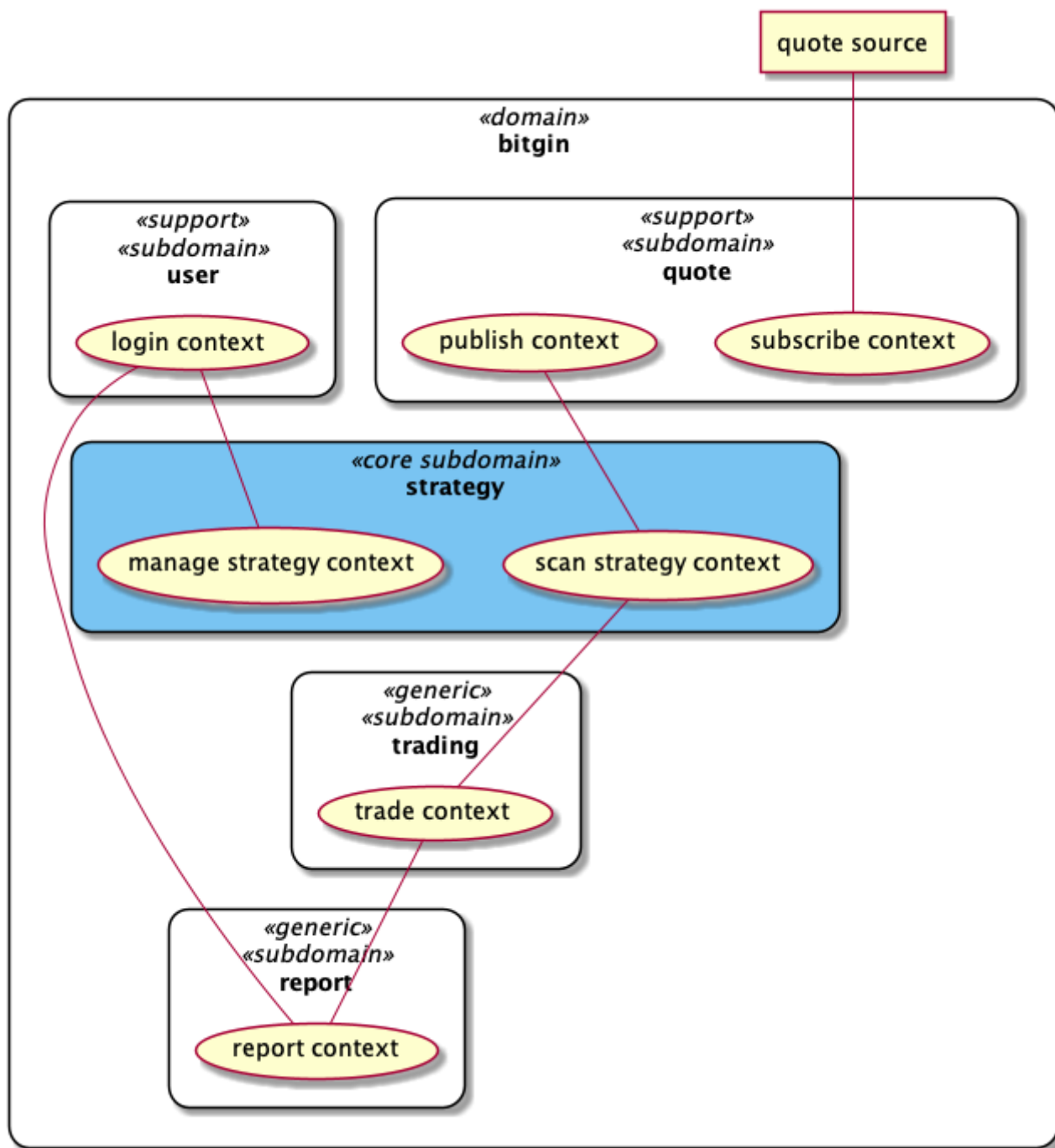
- quote component: 是一個聽取外部來的報價資訊元件。
- strategy component: 聽取到送來的報價資訊進行策略運算後執行交易。
- trade component: 收到 strategy component 的交易訊號後，送出交易到外部系統。
- outside exchange: 外部的交易所。

## Solution 2



- read quote file: 讀取或抓取歷史報價送入 quote component。
- report component: 針對交易紀錄計算年化報酬率產生出相應的報告。

## Solution 3



- quote: subscribe 和 publish 即時報價到 strategy。
- user: 每個 user 都可以登入管理自己的不同的策略和查看整個 report。
- strategy: 我把它定義為整個系統的核心有著各種策略，聽取報價計算策略結果送出給 trading component。
- trade: 主要是跟交易的元件，可能是跟外部系統做溝通。