# **評估是否需要導入區塊鏈？**

供應鏈導入區塊鏈，主要評估在於是否需要**區塊鏈"永久性記錄"的特性**。以下擷取自 [1] 第58~60頁。



* 問題 A：

要使區塊鏈成為合適的解決方案，需要了解業務環境。這些問題詢問是否需要取消中介機構。在產品合規性和保證的情況下，使用區塊鏈並不試圖消除任何中間團體。最終，產品供應商需要提供適合其用途的證明。需要這些證據的提供，以便營建產業和監管機構人員放心該產品適合於目標用途。

* 問題 B ：

要成功應用區塊鏈，它需要與數位化的本地資產一起進行，這意味著可以成功以數位格式表示的資產。對於具有 GTIN(全球貿易物品編號)的建築產品是可能的。這是一個全球唯一的識別碼，可為實體商品提供數位形式。

* 問題 C：

是否可以為數位資產建立永久記錄。這是關鍵問題之一，因為區塊鏈的優勢之一就是它是信任之源的概念(同上)。從這個意義上講，正是產品供應商需要確保產品符合特定用途。這裡的問題是，隨著產品的不斷發展和供應商提供最新證據，是否需要永久記錄。

* 問題 D：

與業務流程所需的速度有關。目前的區塊鏈無法高速處理交易。區塊鏈的可擴展性是當前技術狀態中最多被引用的問題之一。

* 問題 E：

目前不建議在區塊鏈上儲存非交易型數據。非交易型數據具有定義條件或數據值何時保持不變的開始日期和結束日期。當涉及產品保證和合規性問題時，這一點至關重要。如果對產品的性能或規格進行了改寫或變更，則需要新的證據證明該產品適用於特定目的。

* 問題 F：

是否需要依賴可信任的一方？例如出於合規性或責任性原因(同上)。世界經濟論壇指出："在監管發揮重要作用的使用案例中，可能有必要將主管機關納入計畫，並提供主管機關可以確保如何調適法律的手段。而如果沒主管機關參與，部署區塊鏈可能會非常困難"。因為信任的需要是重要關鍵，並且產品供應商需要提供證明其產品適合用途的證據。主管機關可以在建築標準架構中發揮作用，而在這種決策樹模型中，區塊鏈可能會有其作用，但是需要進一步的研究和開發。以目前情況而言，區塊鏈還沒有足夠成熟的狀態來解決這個問題。

* 問題 G：

區塊鏈是否被用於契約關係管理或價值交換。因為就營建產業來看，契約關係並未得被管理。這個想法是，是否可以使用區塊鏈來提供一種產品符合法規的機制。

* 問題 H：

網路中的所有成員是否都需要能夠在區塊鏈上寫入交易。有許多製造商將產品供應紐西蘭市場。要求這些供應商都能向區塊鏈登錄資訊。

* 問題 I ：

合作者是否彼此了解和信任。如果已經建立信任，那麼可能就不需要區塊鏈了。在建築業中，已經知道有許多供應商，但是並沒有要求他們彼此真正認識。

* 問題 J：

是否需要功能控制。如果需要功能控制，那麼就很有必要使用許可制的區塊鏈。

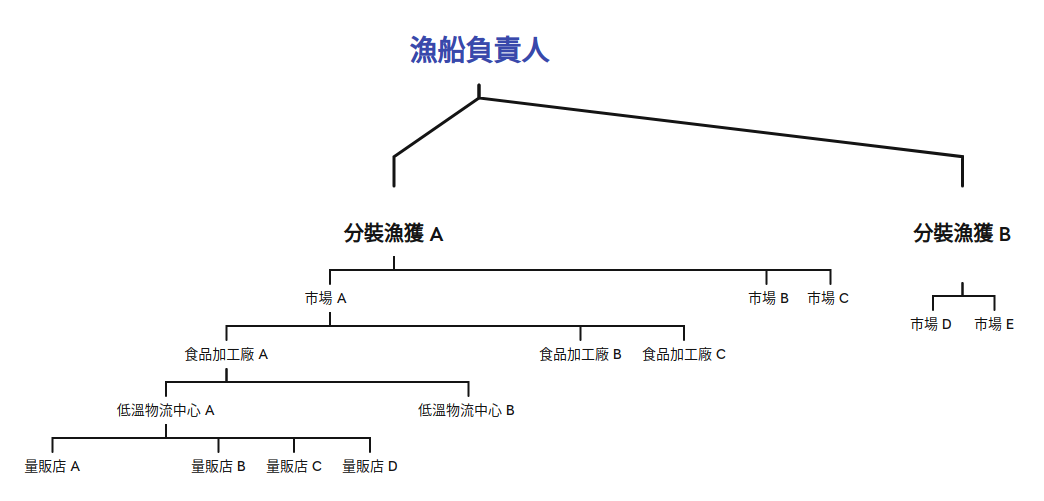
* 問題 K：

交易是否應該公開。如果可以公開進行交易，那麼就有充分的理由採用公有制區塊鏈。

# **區塊鏈導入範例：供應鏈**

內容說明：

* 實際上，供應鏈可能無法像hyperledger fabric的聯盟鏈或私有鏈一樣，成員知道彼此存在。
* 現實中的供應鏈可能最上游廠商不認識最下游廠商，需要後台系統將各公司間進行關係串連，最終達到確認所有分裝貨物的狀態。
* 以下圖為例，最終一群顧客在量販店 A 購買的產品，溯源時，背後會是同一條公司關係鏈：分裝漁獲 A 的漁船→市場 A →食品加工廠 A →低溫物流中心 A →量販店 A
* 區塊鏈在每一個環節針對重要資訊做保存



工作事項：

* 建立後台系統
* 區塊鏈平台的建立，保存重要資訊
* 建立 UI 介面(Web 或 行動裝置)

資料設計：

* 後台系統【供應鏈中的公司鏈結table】

| **公司**  **(unique)** | **序號**  **(unique)** | **父序號** |
| --- | --- | --- |
| 漁船負責人 | 001 | 000 |
| 市場 A | 002 | 001 |
| 食品加工廠 A | 003 | 002 |
| 低溫物流中心 A | 004 | 003 |
| 量販店 A | 005 | 004 |
| 市場 B | 006 | 001 |
| 市場 C | 007 | 001 |
| 食品加工廠 B | 008 | 002 |

* 後台系統【供應鏈中的商品溯源鏈結table】

| **公司** | **貨品名稱** | **貨品序號**  **(unique)** | **產生貨品**  **日期** | **貨品簽收**  **(簽名及日期)** | **貨品狀態**  **(被拒絕**  **被簽收**  **其他)** |
| --- | --- | --- | --- | --- | --- |
| 漁船負責人 | 分裝漁獲 A | 001 | 8/26 | 市場 A  簽收 | 被簽收 |
| 市場 A | 批發鮭魚 A | 002 | 8/27 | 食品加工廠 A  簽收 | 被簽收 |
| 食品加工廠 A | 鮭魚罐頭 A | 003 | 8/28 | 低溫物流中心 A  簽收 | 被簽收 |
| 低溫物流中心 A | 一箱鮭魚罐頭 A | 004 | 8/29 | 量販店 A  簽收 | 被簽收 |
| 量販店 A | 一箱鮭魚罐頭 A | 005 | 8/30 | - | - |
| 市場 B | 批發飛魚 B | 006 | 8/27 | 食品加工廠 B  簽收 | 被簽收 |
| 市場 C | 批發魟魚 C | 007 | 8/27 | 食品加工廠 C  簽收 | 被簽收 |
| 食品加工廠 B | 鯊魚罐頭 B | 008 | 8/28 | 低溫物流中心 B  簽收 | 被簽收 |

* 區塊鏈儲存

key值：貨品序號

value值：貨品名稱、產生貨品日期、貨品簽收、貨品狀態

問題說明：

* 由誰登入所有上下游公司的資訊？由誰修改、刪除、更新？
* 資訊若有修改、刪除、更新，要用什麼方式儲存到區塊鏈中？
* 是否所有環節都已經/ 可以數位化？
* 需要儲存哪些資料到區塊鏈中？(資料設計為參考範例)

# **智能合約如何對帳本進行存取？(以超級帳本為例)**

* 智能合約是區塊鏈網路對外溝通的橋樑。
* 在超級帳本中，智能合約可用 GO Java 等開發。
* 超級帳本中提供許多函式，可進行區塊鏈資料存取，此處以 GO SDK 說明一些簡單、常用的函式。
  + <https://pkg.go.dev/github.com/hyperledger/fabric-chaincode-go/shim>
  + <http://www.netkiller.cn/blockchain/hyperledger/chaincode/ch20s03.html#idm429336627504> (參考[8])
* 超級帳本的帳本說明：
  + 以 key-value 方式做儲存
    - value 儲存成 json 格式
  + 將資料儲存到帳本中，意思是會同時儲存在 blockchain 及 world state
    - blockchain 是永久保存的文件檔
    - world state 保存賬本狀態當前值的NoSQL DB

**寫入資料**

PutState(key string, value []byte) error

會同時儲存到 blockchain 及 world state中

**讀取資料**

GetState(key string) ([]byte, error)

讀取 world state，所以若做了 DelState 則會抓不到資料

**刪除資料**

DelState(key string) error

會刪除 world state 中的資料，但不會刪除 blockchain 資料，在 blockchain 中是注記此資料被刪除

**修改資料**

沒有這個函式，但是可以透過先 GetState 再 PutState ，達成修改資料的功能

所以在修改資料時，提供要修改的欄位值與 key 值即可

**富查詢**

GetQueryResult(query string) (StateQueryIteratorInterface, error)

可使用 value 中的 json 之欄位來進行查詢

必需使用 Couch DB，使用 NoSQL 的 selector 功能

**通過 key 值讀取歷史資訊**

GetHistoryForKey(key string) (HistoryQueryIteratorInterface, error)

同一個 key 值對應的 value 可透過上述函式修改、刪除(使用 PutState、DelState)，在歷史資訊中保存所有動作的 txid、時間戳、原始 value 內容、若是做 DelState 則會顯示被刪除

**建立複合鍵來儲存資料**

CreateCompositeKey(objectType string, attributes []string) (string, error)

自行設定index，以字串陣列儲存每個欄位數值，讓每個欄位都是複合鍵 key 值，用來提高查詢效率

eg. CreateCompositeKey("txID~time~hash~device", []string{"txid", "time", "hash", "device"})

**分解複合鍵來查詢資料**

GetStateByPartialCompositeKey(objectType string, keys []string) (StateQueryIteratorInterface, error)

透過給定部份複合鍵來查詢資料

eg. GetStateByPartialCompositeKey("txID~time~hash~device", []string{"txid", "time"})

**將復合鍵拆分為其組成的內容部分**

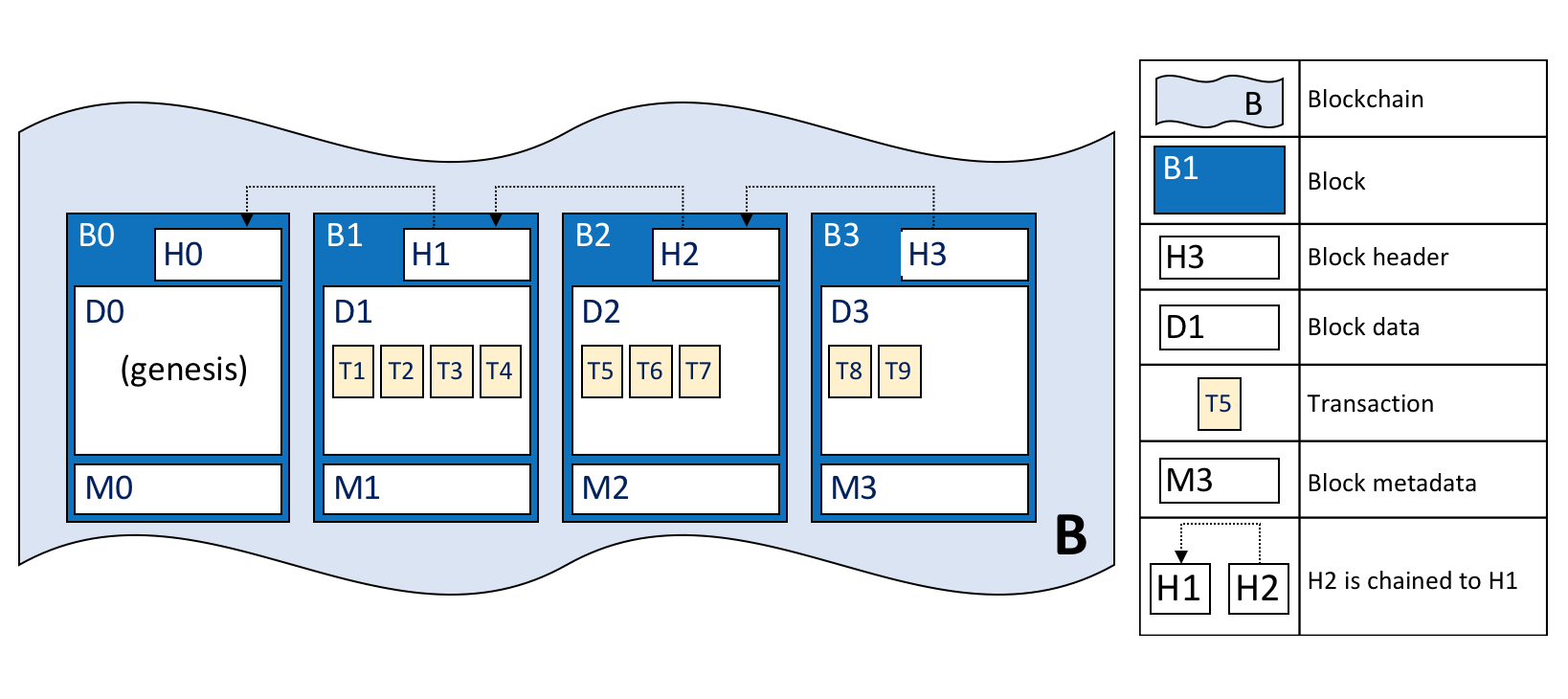
將 GetStateByPartialCompositeKey 抓取的那些複合鍵值做切割，使我們可以透過陣列去獲取每一個複合鍵的數值

SplitCompositeKey(compositeKey string) (string, []string, error)

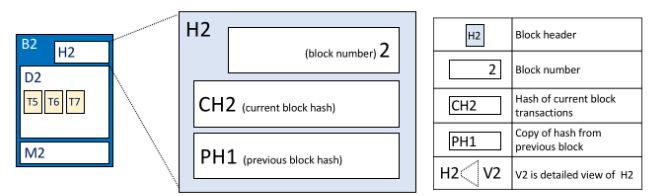
eg. SplitCompositeKey(responseRange.Key)

# **為什麼區塊鏈中的資料是永久性記錄？**

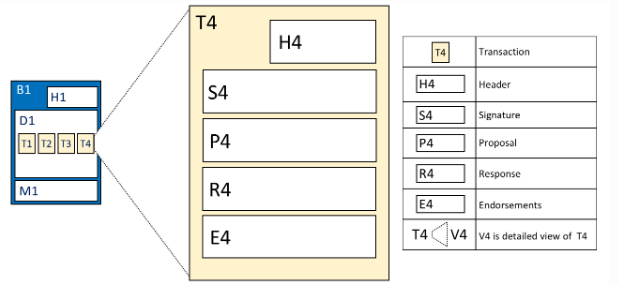
* 以銀行帳戶來說，可以查詢到交易明細、當前餘額
* 超級帳本的帳本類似銀行帳戶，儲存賬本的當前值，以及帳本的交易歷史
* 賬本的當前值可能改變，但有關它的事實歷史是不可變的，可以添加，但不能更改
* 賬本由兩個不同但相關的部分組成：world state 和 blockchain
  + world state：
    - 保存賬本狀態當前值的NoSQL DB
    - 直接訪問狀態的當前值，而不必通過遍歷區塊鏈來計算它
    - 以 key-value 方式做儲存
    - 可以新增、刪除、修改
  + blockchain (區塊鏈)：
    - 區塊鏈就是一個交易記錄，記錄每一筆在 world state 中的變化
    - blockchain 數據結構與 world state 不同，一旦寫入就無法刪除、修改
  + world state 記錄當前值，而 blockchain 帳本的交易歷史
* 超級帳本為分散式帳本
  + 通常會以一個業務場景建立及使用一個帳本
  + 超級帳本網路可建立多節點，每個節點都會有一本帳本副本
  + 每個副本與其他副本內容一致
* blockchain 的數據結構(以下圖說明)
  + Blockchain 區塊鏈：
    - 以文件來實現
    - 區塊鏈 B 包含區塊 B0、B1、B2、B3
    - B0 是區塊鏈中的第一個區塊，叫做創世區塊，記錄了區塊鏈網路配置，因此不包含任何交易，是賬本的起點
    - 區塊 B2 有一個區塊數據 D2，其中包含它的所有交易：T5、T6、T7
    - 區塊 B2 有一個區塊 header H2，其中包含 D2 中所有交易 hash 以及 H1 的 hash，使區塊間密不可分且不可改變地相互關聯



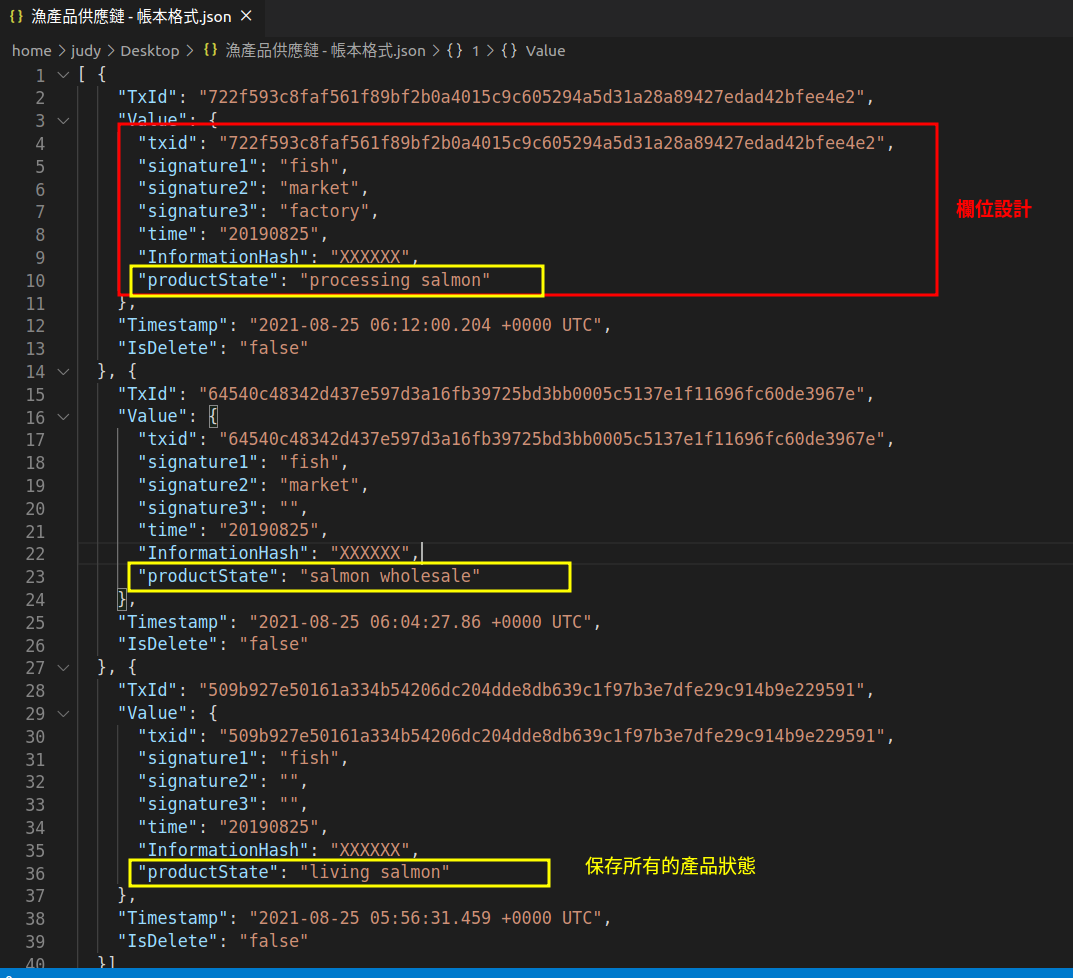
* + Block 區塊：
    - 由 Block Header、Block Data、Block Metadata 組成
    - 在新建立 Block 的時候，Block Header 會寫入以下三個欄位：
      * Block number：一個從 0（創世區塊）開始的整數，每添加一個新區塊就增加 1。
      * Current Block Hash：當前區塊中包含的所有交易的 hash 值。
      * Previous Block Header Hash：來自前一個區塊的 Block Header 的 hash 值。
    - Block Header 中的這三個欄位確保每個區塊都與其鄰居密不可分，形成一個不可變的分類賬。



* + - Block Data
      * 包含了由 orderer service 排序好的交易串
    - Block Metadata
      * 時間戳記、區塊寫入者、公鑰、證書、交易驗證的 Flags
  + Transactions 交易：
    - Header
      * 記錄 Chaincode 名稱、版本等資訊。
    - Signature
      * 客戶端製造的數位簽章
    - Proposal
      * 將客戶傳進來的輸入參數作編碼，以進入 Chaincode 執行，合約在被執行之前呼叫者需要這個步驟擬好自己輸入參數，記錄在交易內容內。
      * 由於在此階段還未代表合約會被執行，所以稱之爲提案。
    - Response
      * 用來更新 world state 狀態。
    - Endorsement
      * 一個經過簽章後的反應清單（List of Signed Transaction Responses），經過 Validator 的驗證已達成簽署的資料。

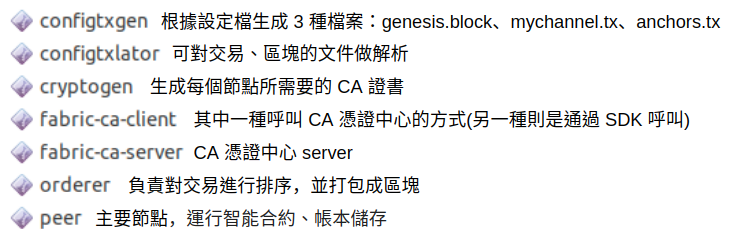


* 下圖為從 blockchain 中存取的資訊範例
  + 同樣一個 key 值，做了三次交易，欄位做了哪些修改都記錄在上面

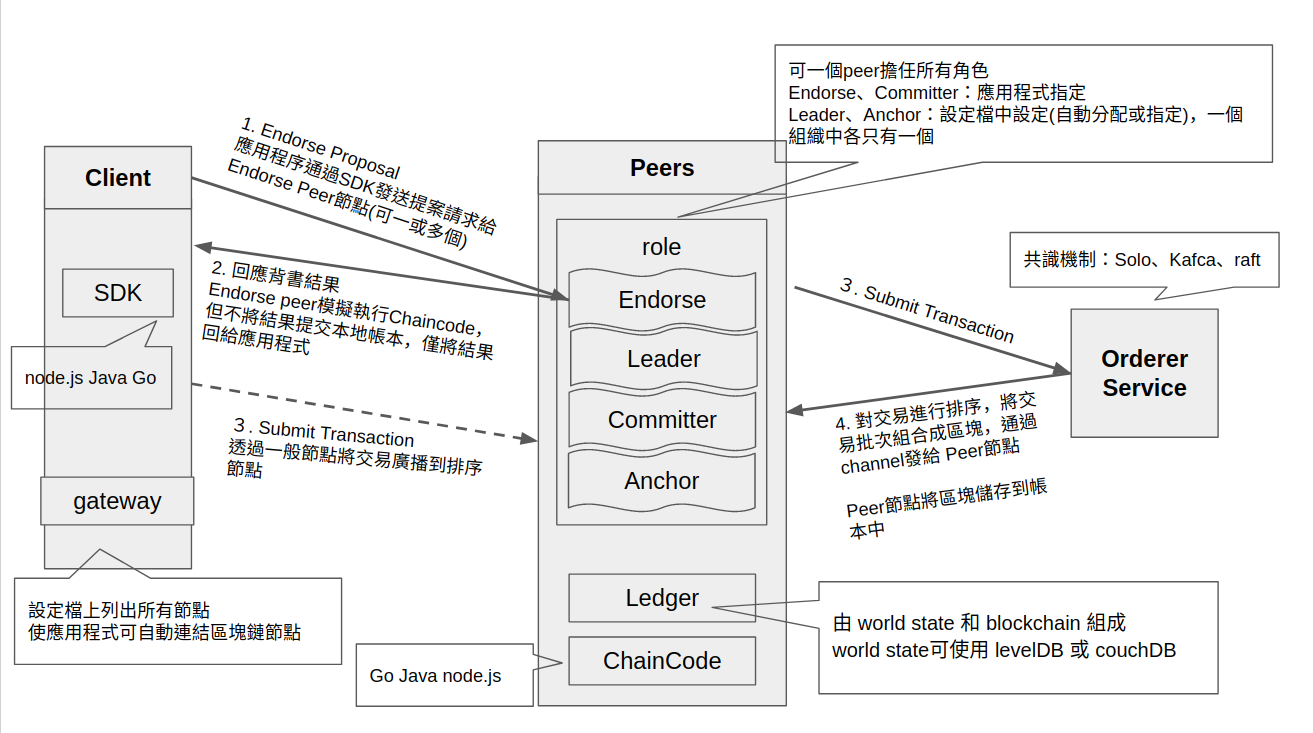


# **hyperledger fabric可以有哪些模組做組合？**

**超級帳本有哪些模塊**

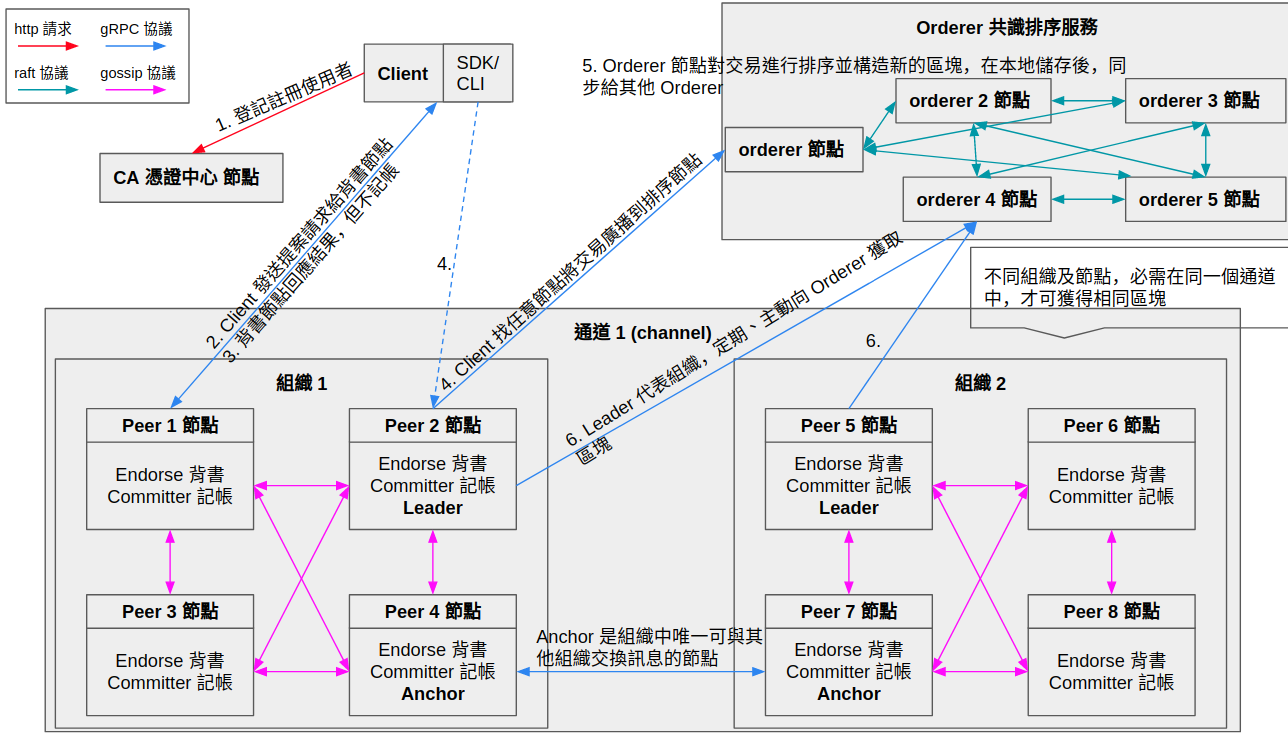


**交易流程 與 可替換插件**



**多組織的組成**

* 至少一個 CA 憑證中心 節點
* 組織中有一個或以上的 Peer 節點，兩個節點表示有帳本副本，可避免單一節點故障資料遺失的問題
* Orderer 節點分別放在不同機器上可避免單一機器故障的問題。由於使用 raft 排序，一定要有5個 Orderer 節點



# **參考資料**

[ 1 ] 林谷陶（無日期）。PR10912-0123 。區塊鏈技術及營建產業應用案例探討 成果報告 。109 年 12 月，取自：https://ws.moi.gov.tw/Download.ashx?u=LzAwMS9VcGxvYWQvNDA0L3JlbGZpbGUvOTQ4OS8yMTQzMzYvNDY1OTM1YmQtM2I2ZC00NzIzLWIyYjItM2RjMzZlZmZkNTVjLnBkZg%3D%3D&n=MTA5MTIyMV8xMDnljYDloYrpj4jmioDooZPlj4rnh5%2Flu7rnlKLmpa3mh4nnlKjmoYjkvovmjqLoqI4o5oiQ5p6c5aCx5ZGKKS5wZGY%3D

[ 2 ] 李俊宏（無日期）。OAC-NKUST-109-017(研究報告)。朝向藍色經濟的全球化願景-區塊鏈技術導入漁產品供應鏈生產履歷系統之開發研究（成果報告） 。109年 09月，取自：https://www.oac.gov.tw/filedownload?file=subsidy/202101111617230.pdf&filedisplay=%E6%B9%AF%E8%80%80%E7%BF%94\_%E6%88%90%E6%9E%9C%E5%A0%B1%E5%91%8A+%E5%88%AA%E4%BD%9C%E8%80%85.pdf&flag=doc

[ 3 ] hyperledger官方文件。取自：https://hyperledger-fabric.readthedocs.io/

[ 4 ] hackmd筆記。取自：https://hackmd.io/@iknowright/HJCP6P1wI

[ 5 ] Hyperledger fabric 賬號 CA篇 （七）。取自：https://www.796t.com/article.php?id=204820

[ 6 ] Hyperledger Fabric：系統架構。取自：https://ithelp.ithome.com.tw/articles/10205811

[ 7 ] Hyperledger Fabric的基本概念和架構。取自：<https://www.gushiciku.cn/pl/pi6e/zh-tw>

[ 8 ] Mr. Neo Chan, 陈景峯(BG7NYT)。Netkiller Blockchain 手札。取自：<http://www.netkiller.cn/blockchain/index.html>