# 適用於RWA的區塊鏈設計草稿

本文分爲兩部分：

1. 目標版設計方案。  
   該部分描述的方案可以認爲是最終目標，可用來編寫白皮書。  
   它在設計上是完整的，相應地，要實現它所需的研發資源和時間也較多。
2. 現實版設計方案  
   實際運作中，可先推出滿足所需安全性，性能要求的，剪除非必要功能的版本，以便以最快速度，最低研發投入上綫服務，後續在運營中再逐步迭代，完善版本。

# 設計目標

本方案用於打通RWA資產與DeFi。它是一條區塊鏈，該鏈對應Centrifuge的基於Polkadot的鏈，用來安全，低成本地維護各種RWA池，RWA資產信息，以及讓SPV能夠審核這些RWA，並對其簽名。它需要做到：

* 安全。鏈需要足夠安全，可使用RollUp，利用以太坊提供安全性。需有確定性共識，而不是概率性共識。
* 性能可擴展，能夠滿足將來可能的巨大的業務量需求。
* 提供跨鏈DA能力(RWA數據)，可方便地跨鏈獲取各種所需數據。
* 交易需得到特定用戶（即SPV）驗證，方可被共識。
* 方便跨鏈

## 設計思路

未來我們需要怎樣的區塊鏈？目前區塊鏈與web3緊密結合，後續會滲入到各個領域和場景。因此當前單體區塊鏈（如Ethereum）的運行形式是不合適的，它太龐大了，消耗的資源太多，而承載的數據又太少，如此高成本的運行方式無法支撐下一步的海量應用需求。我們需要考慮怎樣的區塊鏈才能支撐下一步海量的共識需求。

我認爲它需要滿足以下要求：

* 安全性足夠高，這是基本要求。
* 不是一個大的，統一的賬本，而是針對各個領域，場景，甚至應用，可以有自己的賬本，這個賬本是去中心化的。這個賬本可能節點數量不多，用戶群體也不大，但是其安全性與整個區塊鏈相比并沒有降低。各個賬本在各方面擁有足夠的可擴展性（分片擴展，包括TPS，帶寬，世界狀態，賬本存儲等）
* 不同賬本閒可以快速，安全，可信，低成本地跨鏈。
* 有跨越各個賬本的，可信的，可管理權限的DA(data available)服務，這是非常重要的基礎設施。
* 擁有可以跨越各個賬本的統一的身份標識，經濟模型等基礎設施。

本文嘗試設計一個滿足如上要求的區塊鏈底層基礎設施。它可以方便地適用於RWA場景。

區塊鏈的安全性來源於對有限資產的占用，例如POW來源於算力，POS來源於資金的時間價值。在POS共識算法下，在某個時間點（如當下），整個世界中可用於抵押到區塊鏈中，保證安全性的資金是固定的，因此各條鏈之間對資金是競爭關係，這也導致能夠分到具體某條鏈上的資金是有限的。更出名，使用量更大的鏈可能能競爭到更多的資金來保障自己的安全，而比較不出名的鏈則用於保證安全性的資金較少，進而安全性受到挑戰。

然而基於以太坊+RollUp的設計模式給上述安全模型帶來了新的看法。它將安全性、去中心化、性能擴展做了解耦。其中以太坊作為L1，負責安全性和去中心化；RollUp作為L2，負責性能擴展。目前已在運行的RollUp基本都是中心化的，由一台Sequencer完成交易收集、執行，再把執行結果提交到L1上。但從效果上看，業界並不認為RollUp中心化和不安全了，因為其安全性和去中心化由以太坊保證，由以太坊主網絡裁決L2的執行內容和結果。

受到如上設計的啟發，我們設計了本區塊鏈。它採用分層設計，包含共識層和執行層。資金抵押統一在共識層，在共識層保證安全性和去中心化；執行層可以包含多個獨立的區塊鏈，它們完成性能、存儲、網絡的擴展。本區塊鏈可以看作是在執行層進行了分片，它同時將CPU、RAM、NET、DISK （即交易執行、世界狀態存儲、網絡吞吐能力、區塊存儲）進行了分片，因此具有近似於無限擴展的能力，其擴展上限是共識層的吞吐量。

同時它包含一個跨越所有層的IPFS網絡，該網絡提供帶有權限管理的，可信，高校的DA服務。

# 2. 目標版設計方案

## 2.1 概述

我們採用分層架構。分為共識層和執行層。其中共識層保證安全性和去中心化，並達成共識；執行層完成性能擴展，並可以靈活承接各種類型的任務，即如果與Centrifuge對應的話，執行層即各個不同SPV維護的RWA資產池，如房地產資產池，供應鏈金融資產池等。本方案與Centrifuge相比，優勢在於：

* 各個SPV可以自己獨占一條鏈，因此其數據是隔離的，而且不用互相競爭鏈上資源。
* 每個SPV鏈在保證安全性的同時，其運行成本是極低的，最低僅需一個節點即可。

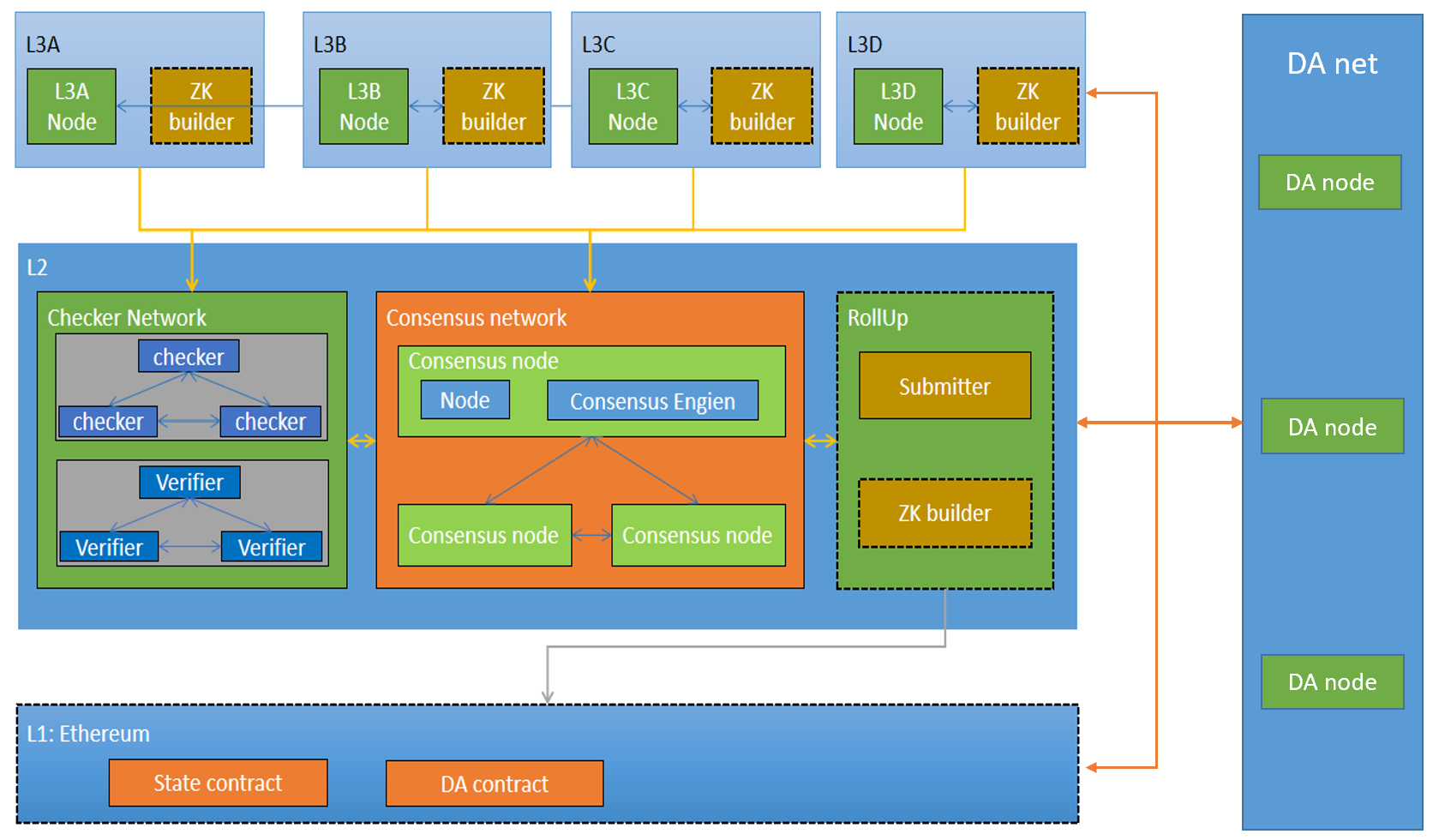
整體結構分為三層，如下圖所示，虛線部分為可選模塊：

● L1層（可選）：即以太坊。在以太坊上部署合約，管理狀態根和對交易的驗證等等，以及數據有效性。這是一個基於Validium的RollUp，因此其成本會比OP RollUp和ZK RollUp都低很多。

● L2層，即共識層。負責完成共識和保證安全性。包含Checkers、共識網絡、RollUp模塊，共識網絡是核心。

● L3層，即運行層。運行交易，並且維護鏈狀態。 L3層體現為多個獨立的鏈，可以提供不同類型的服務。在RWA場景下，每個L3鏈可以是一個SPV提供的服務。

* DA網絡，一個基於IPFS的文件存儲網絡，L1,L2,L3均鏈接到該網絡，可使用CID從該網絡中下載文件。



**共識層(L2)** 使用POS共識，負責：

1. 共識收到的區塊，並廣播到執行層

2. 通過智能合約，管理抵押、執行層交易根等。通過抵押保證整個平台的安全性。

3. 對執行層的執行結果進行檢查，形成安全閉環。

**執行層(L3)**負責：

1. 收集交易

2. 各節點分別將收集到的交易打包成塊發送給共識層

3. 執行共識層共識後的區塊

在運營初期，為了保證鏈的安全性，可以將數據定期RollUp到以太坊，當項目的安全性或需求足夠後，可以考慮移除RollUp部分，以降低成本。

## 2.2 角色

### 2.2.1 L3

L3中可以有多個獨立的網絡，每個網絡維護自己的一條區塊鏈。這條鏈的共識由L2（注意L3的共識不是自己完成的，而是L2完成的，具體參見共識部分）完成，L3只保存鏈的數據、狀態；完成交易的執行等。 L3中可能包含如下角色：

* Node：L3上的節點。這些節點的功能與普通以太坊節點功能幾乎一樣。只不過部分節點會出塊，作為參與共識的一部分，共識流程與以太坊不同。具體共識部分見相關章節。
* ZK Builder（可選）：零知識證明構建者。 ZK-EVM可實用後，考慮實用ZK Builder代替L2中的Verifier角色。

### L2

L2是該網絡的核心，其由三個主要模塊組成：

* Checkers網絡。  
  包括checker和Verifier。  
  在RWA場景下，Checker代表SPV完成對鏈上交易的認證，而這些交易即是要放到鏈上的RWA相關數據。主要執行對L3提供的區塊等數據的檢查。 Checker對外體現為一個發送交易到L2鏈的賬戶。其內部結構可以是區塊鏈節點，也可以是其他結構。  
  每個L3可以指定不同的Checkers集合，所以這個網絡的成員不是固定的。具體工作原理可見共識流程章節。  
  驗證者(Verifier)。完成對L3區塊執行結果（狀態根）的驗證，詳情參考共識流程。
* 共識網絡。這是海峽鏈八爪魚網絡的核心。整個網絡的共識均由該網絡確定。該網絡是一條基於POS的區塊鏈。
  + 節點（Node）。是普通的區塊鏈節點，可以執行區塊交易、保存區塊鏈狀態和歷史、通過p2p網絡傳輸數據等。
  + 共識引擎(Consensus Engien)。與L3節點、Checkers、Verifier共同完成共識。它對外表現為一組智能合約，部署於L2區塊鏈上。
* RollUp。與普通的RollUp類似，完成從八爪魚到以太坊的數據錨定。為了降低gas成本，考慮實用Validium模式，即交易數據不會提交到以太坊主網，而是將交易證明提交到主網，在DA net上提供交易數據可用性。 RollUp模塊在初期採用Optimistic模式，等ZK EVM成熟後，再切換到ZK EVM。
  + Submitter。完成區塊數據收集、提交到以太坊、欺詐證明等任務。
  + ZK Builder。切換到ZK RollUp後，負責構建EVM證明。

### 2.2.3 L1

即以太坊网络。以太坊网络上会部署智能合约，完成RollUp，不再赘述。

## 2.3. 共識

### 2.3.1 共識合約

L2上部署了一個共識合約工廠，每當添加一條L3區塊鏈時，就會通過該工廠合約創建一個新的合約，負責該L3鏈的共識工作。合約中包括如下內容：

**存儲變量：**

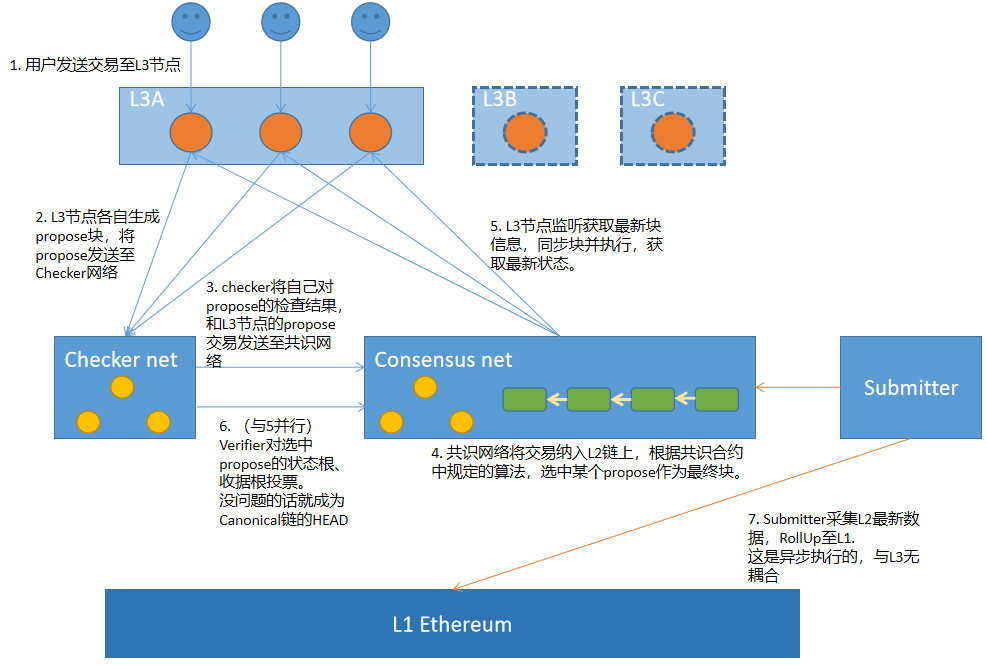
* 被允許提交propose的L3節點列表
* 負責驗證propose的Checker列表（該列表默認是L2共識節點列表）
* propose列表，包括每個L3提交者的propose，每個propose包括塊高，塊hash，交易根、狀態根，收據根
* 區塊驗證結果，是每個L2共識節點對當前塊的狀態根和收據根的驗證結果投票
* canonical鏈信息，包括最近的canonical HEAD塊信息。 （注意只包含最高塊的信息）.  
  包括塊高，交易根、狀態根、收據根。  
  一個新propose被充分驗證後，會被添加到canonical，成為HEAD。
* VMType。虛擬機信息，默認使用EVM。
* TxMode。 propose中交易表達方式，默認是交易全文；當系統吞吐量太大，L2網絡擁塞，無法接收全部propose數據時，可以調整為propose中只包含交易hash。
* stateGap。共識中，塊共識和狀態根共識的最大距離。最小為0。可在合約初始化時設置。

**接口：**

* 共識算法。由Checker的驗證交易觸發。被觸發後，判斷當前是否滿足共識條件，若滿足則執行共識算法，選中某個propose作為新塊。
* 驗證算法。由Verifier的驗證交易觸發。被觸發後，判斷當前新塊的狀態根、收據根是否被充分驗證，如果已被充分驗證，則新塊被添加到canonical鏈。

### 2.3.2 共識流程

原理圖如下：



1. 用戶將交易發送至L3A鏈上的各個節點，如L3A-1，L3A-2, L3A-3。

2. L3A 各個出塊節點各自收集交易並分別打包成塊，生成propose，其內容包括：

a. 新區塊，包含塊頭、塊體（交易集合）

b. propose交易，該交易包括新塊的hash，交易根，狀態根，收據根

c. 狀態轉換證明，即該鏈的狀態，是如何從當前canonical HEAD的狀態根，通過執行propose的交易列表，變為propose中的新狀態根的。該證明同時也可以完成對收據根的校驗。

L3節點將propose被發送至共識合約中規定的Checker列表中的Checker。一般情況下，Checker列表與L2共識列表是一樣的。注意我們有多個L3節點（如上圖L3A-1, L3A-2,L3A-3），會分別生成propose並發送至Checker。

3. 各個Checker（即SPV）收到來自各個L3節點的propose後，會對區塊裡的交易信息進行基本驗證，除了檢查每筆交易是否有正確簽名，以及交易根是否正確外，還會對交易正確性進行檢驗。這裏L3A代表一個SPV經營的網絡，如供應鏈金融服務網絡，因此塊内的數據都是相關的發票，供應鏈三流數據等，SPV需要對這些數據完成檢驗方可放行。

Checker會生成一筆確認propose驗證結果的交易，連同原始的propose交易一起發送至共識網絡。

4. 區塊共識：

a. propose交易和對其確認的交易都調用L2鏈的共識合約，L2鏈會將其納入鏈上。當收到足夠的確認交易時，一個propose就會進入checked狀態。

b. 當checked狀態的propose數量滿足合約要求時，就會根據選擇算法選擇某個propose成為selected狀態。此時對塊（即交易列表）的共識完成。

5. 各個L3節點監聽L2鏈上的針對本鏈的共識合約。當合約中選出提案塊後，各個節點便會執行該塊，得到最新的狀態根。 （不等待Verifier的確認，因為L3節點自己便可確認，誠實的L3節點可以獲得正確的狀態根，且相信Verifier可以得到與自己一致的驗證結果）。

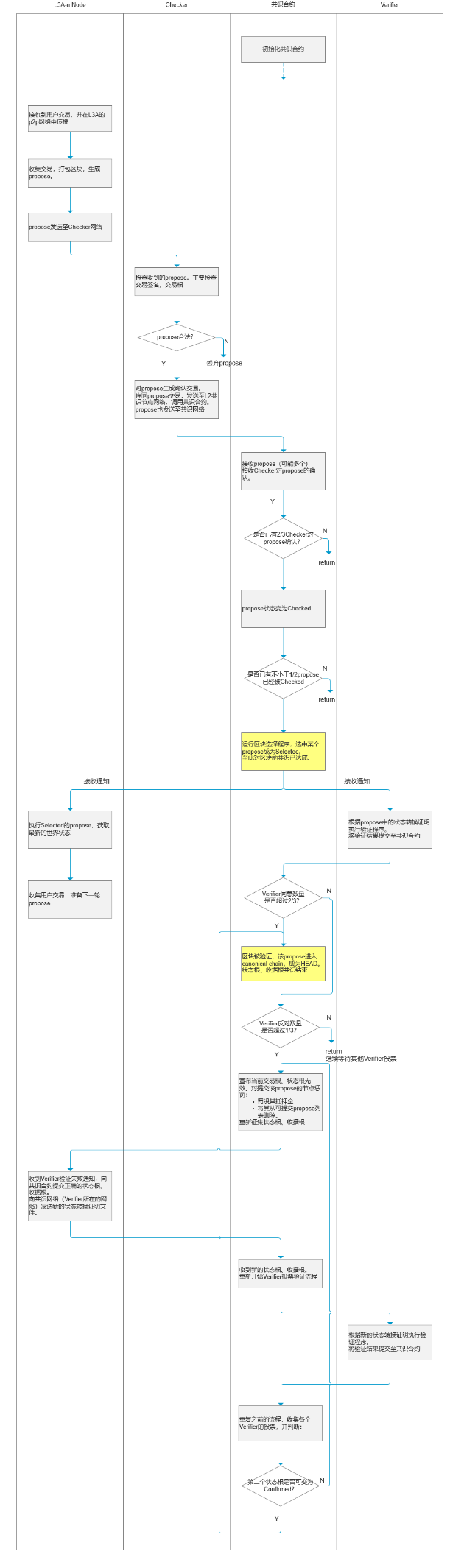
至此在L2,L3上的處理便結束了。

6. （與5並行）針對selected的propose，Verifier就會根據該propose中的狀態轉換證明，驗證其新狀態根是否正確，將驗證結果作為交易發送至共識合約。當不少於2/3的Verifier對propose轉換證明投票驗證後，該塊會變為confirmed狀態，此時會將其更新到canonical chain，作為HEAD。

7. Submitter收集L2鏈上的新區塊信息，將L2鏈內容RollUp至L1以太坊。該步驟與1-5是異步的，相互之間沒有耦合。

注意上述流程中，4和6 完成了兩個共識：對區塊的共識(即某個propose從checked->selected)，和對新的狀態根的共識(即該propose從selected->confirmed)。當對區塊的共識完成時（selected），對L3鏈上的節點來說，該塊的共識已經完成，他們可以繼續準備打包下一個區塊了。對輕節點和跨鏈應用來說，需等待狀態根和交易根共識完成（confirmed），方可認為該塊完結。所以請注意：對區塊的共識和對狀態根的共識是分開的，區塊狀態根的共識可以落後於區塊的共識。即區塊共識完成後，即可進行下一個區塊的打包和新的共識，而不是等待對狀態根也達到共識後，方可進行下一個區塊的打包。例如區塊共識塊高可以是105，而狀態根的共識只到100。但是對區塊的共識和對狀態根的共識之間的共識高度差不是可以無限擴大的，當該差值擴大到一定程度時，說明該L3鏈的節點大概率運行於不正確的狀態，這對L3鏈的安全和可信度是極不利的，因此我們提供一種機制來控制可接受的共識高度差。可在共識合約中通過stateGap設置。該值在合約初始化時設置。最小值可為0，此時對塊的共識需等對狀態根的投票結束，新塊進入canonical chain後方可進行下一個塊的共識。各個L3之間對狀態根確認的要求不同。例如用於DeFi的鏈，其經常需要跨鏈操作，有很多資金操作，因此可將stateGap設置較小，讓Verifier對狀態根充分驗證後再進行下一個塊的共識；而以存證為主，或用於遊戲記錄等場景的鏈，其對TPS的要求較高，跨鏈要求沒那麼頻繁，跨鏈要求實時性要求也沒那麼高，就可以允許塊共識和狀態根共識之間間隔大一點。默認情況下stateGap設置為20，即塊共識最多可以提前交易根共識20個區塊，當交易根共識落後達到20個區塊時，塊共識需等待狀態根共識跟上後，方可繼續前進。

**1-5步驟的詳細流程圖如下：**



### 2.3.3 Verify原理

Verifier 對狀態轉換證明進行驗證，完成Verify動作。

狀態轉換證明文件包括兩部分內容：

* 狀態轉換的舊狀態。該舊狀態是L3鏈當前世界狀態的子集，證明中需包含該子集到上一個塊狀態根（即當前canonical HEAD）的merkle證明  
  Verifier會驗證該merkle證明，與共識合約中的當前canonical HEAD狀態根做比較，完成驗證。
* 被調用合約的字節碼。該字節碼屬於某個合約賬戶，可以通過合約賬戶的狀態到上一個塊狀態根的merkle證明完成驗證。

如上兩部分被分別證明正確後，根據合約中該L3使用的虛擬機類型，Verifier會將如上a、b加載進去，執行propose中的交易列表，交易執行後，生成新的狀態根、收據根。比較新的狀態根、收據根與共識合約中提交的狀態根、收據根是否一致。一致則投同意票，否則投反對票。

### 2.3.4 異常處理流程

2. 3.2為順利情況下的共識流程。當L3區塊作惡時，其提交的propose可能是無效的，這包括兩種可能：

a. 交易列表無效。交易列表可能包括未正確簽名的交易，或交易根計算錯誤。

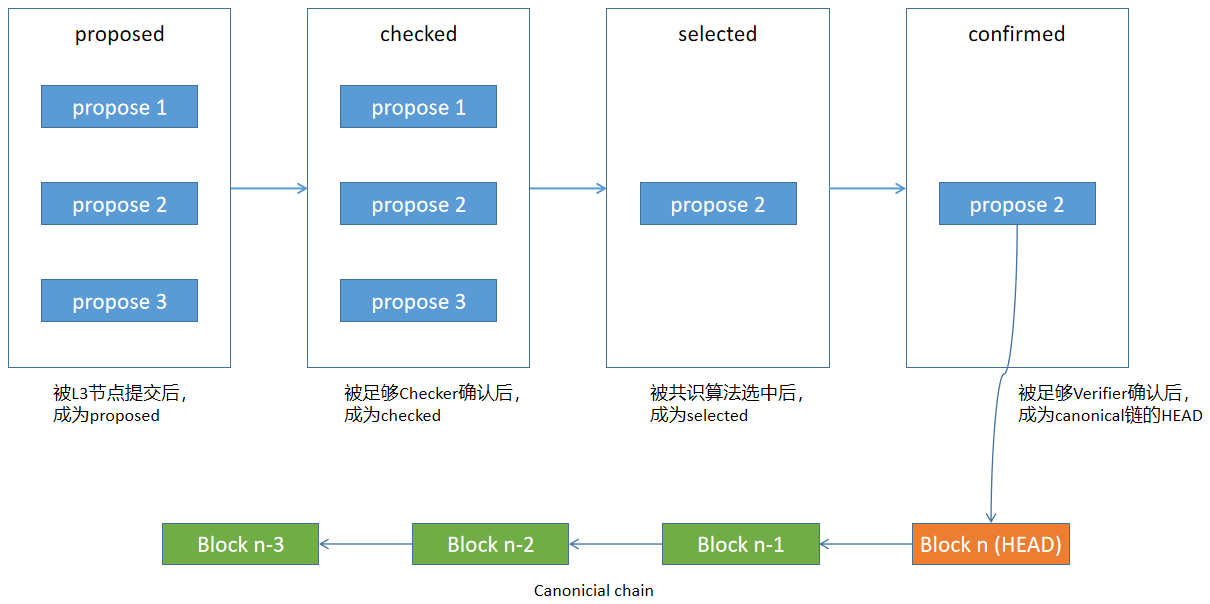
b. propose對應的狀態根、收據根錯誤。

針對a異常，Checker可以發現這兩種情況，直接丟棄這個propose，因此該propose無法出現在共識合約中。交易列表中可能有些交易會執行失敗，如nonce錯誤、value不夠、gas不夠等。這些交易仍是有效的，只是其執行結果會失敗。包含這種交易的propose仍是有效propose。

b異常需在Verify過程中發現。當Verifier對propose的投票中，反對票超過1/3時，該propose被認為是無效的。此時我們可以認為交易列表本身是正確的（因為已經經過Checker驗証投票了），但交易根、收據根是錯誤的，原因可能是作惡的L3節點直接修改了賬戶狀態，或者執行的交易列表與提交的propose中的交易列表不一致。此時合約中會發出狀態根無效的通知。 L3節點收到該通知後，會重新計算狀態根、收據根，並提交新的、被認為是正確的狀態根、收據根到共識合約中，重新觸發Verify流程。如果這次Verify正確完成，則該塊共識成功，進入下一個塊的共識。如果仍然共識失敗，則重複該流程，只要L3節點中有一個誠實的節點，則該流程就應該能完結。如果L3所有節點都是不誠實的，則該L3的狀態根共識將卡在此塊高。同時該L3的交易共識還將繼續進行，即該L3雖然全都是作惡節點，但他們內部的共識還可以通過L2完成，該鏈還將繼續運行下去，但是輕節點和跨鏈機制將無法信任後續新出的塊，後續該L3鏈無法進行跨鏈。當塊共識超過交易根共識的塊數達到共識合約中設定的stateGap時，塊共識會暫停，以等待交易根共識追上。

### 2.3.5 共識細節

#### 2.3.5.1 propose的生命周期



* 從proposed變為checked，需要至少2/3的Checker提供確認
* 何時運行共識的選擇算法，選中某個塊成為selected：
  + 當L3鏈上可提交propose的節點的數量為1，2時，只要有至少一個propose變為checked，就運行選擇算法，將某個propose變為selected。
  + 當L3鏈上可提交propose的節點數量大於等於3個時，需等待不小於一半的節點提供的propose成為checked後，方可運行選擇算法
* 從selected變為confirmed，需要至少2/3的Verifier提供確認

#### 2.3.5.2 區塊選擇算法

將待選propose的hash（區塊hash）排序，

a. 當待選區塊數量為奇數時，選擇最中間的。

b. 當待選區塊數量為偶數時，選擇最中間的中數值較小的。

#### 2.3.5.3 gas补偿

為了鼓勵Checker和L2共識節點參與共識投票，考慮在共識達成後，補償它們在共識投票中消耗的gas，即在Verify完成後，需要向參與共識的節點轉賬，金額為其各自花費的gas。這個轉賬可以只記錄一個可提取數字，後續由各個節點一次性自己提取出來，減少交易量。之所以要做這一步，是因為L2節點的收入是L2的出塊獎勵和塊內gas收入，而投票本身是要消耗gas的，如果僅僅等別人投票，自己不投票，對L2節點來說是最佳選擇，但這樣會導致L3共識無法完成。因此為了鼓勵L2節點參與投票，要把他們消耗的gas補償回去。

### 2.3.5.4 額外出塊激勵（可選）

為了激勵L2節點更積極地對L3塊進行verify，考慮針對每個達成共識的L3塊，有個額外激勵。

**誰能獲得激勵**：

積極的Checker、Verifier可以獲得該激勵。

○ 如果在某個L2塊中，一個propose成為checked，則此塊中及之前塊中針對該propose進行check的節點是有效的，可獲得一份激勵。

○ 如果在某個L2塊中，一個propose從selected變為confirmed，則此塊中及之前塊中對該propose進行Verify的Verifier，可獲得一份激勵。

每個節點可獲得多少激勵：

針對一個L3塊，其額外激勵為一個普通L2塊的1/10。該激勵被分為n份，n為參考上面“誰能獲得激勵”部分的總激勵份數。每個節點可獲得自己的份數。

**罰沒激勵**

如果Verifier在驗證中發現某個L3節點提交的狀態根、交易根是錯誤的，則將罰沒該L3節點抵押的質押幣，這些質押幣中的一半會被平分給所有對該塊投反對票的Verifier，另一半會被燒毀。

### 2.3.6 区块数据结构

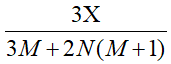
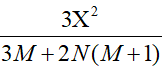
除以太坊區塊內容外，L2塊的塊頭中需要包含：

○ L2共識節點列表。可以是一個列表，也可以是對列表的增刪。總之需要讓L3節點只獲取塊頭，就能獲取L2的共識列表。這是為了防止有節點偽造L2塊欺騙L3節點。同時在跨鏈中也提供信任。

○ 共識節點對塊頭的簽名。這也是為了防止偽造塊頭的。

### 2.3.7 TPS

假设单条区块链独自运行的TPS为X。

* + 对某条L3来说，它的一个区块最快可在1个L2块确定，因此其TPS上限为X。
  + 假设每个L3上有M个节点提交propose，L2上有N个Checker节点，则一个L3块至少需要​笔L2交易才能被确认。因此L2上最多可以容纳条L3链，因此整个网络理论最高TPS=

网络的单个L3 TPS和整体TPS都与单链的TPS相关，如下表格是不同单链TPS情况下，网络可以达到的理论TPS（假设L2和L3单链TPS相同，M=3, N=6）。

|  |  |  |  |
| --- | --- | --- | --- |
| L2 TPS | L3单链最高TPS | 可容纳L3链数量 | 全网最高TPS |
| 1000 | 1000 | 52 | 52000 |
| 2000 | 2000 | 105 | 210000 |
| 5000 | 5000 | 263 | 1315000 |

## 2.4. 再看Checker

觀察共識過程，可以看到我們引入了"Checker"這個角色。它的職責在上文中已經清楚了，它能帶來什麼價值呢？

在L2中，我們只對區塊數據進行共識，並未限制數據是什麼樣子的，因此我們並未規定L3是什麼樣子的。這就為L3帶來極大的可擴展性。 L3可能是針對公共的區塊鏈、針對特定行業的區塊鏈(如處理RWA)、IPFS數據服務網絡、預言機網絡，或者其他任何東西。

如果L3是針對公共的區塊鏈，一般情況下不需要額外的參與方對L3鏈的內容做出檢查，所以此時Checker的身份可以與L2共識節點相同。

如果L3是針對特定行業的區塊鏈，例如針對能源、金融等行業的特定目的的區塊鏈，則可能需要監管者、審批者等對交易進行監管、審批。此時Checker集合可以與L2共識節點是不同的集合，Checker可以在區塊鏈自身的規則之外，進行額外的規則檢查。符合規則的塊才能變為Checked，參與區塊選擇競爭。這為區塊鏈的合法合規帶來了新的設計思路，讓合法合規檢查的成本更小，監管更容易。

L3還可能是提供數據服務的IPFS網絡、預言機網絡等。此時的Checker可以針對各自應用領域，提供特定的檢查服務。

Checker角色的引入，讓L2從一個單純的區塊鏈系統，擴展成可針對各種類型數據和服務的”去中心化驗證器“，大大擴展了海峽鏈的應用邊界，同時降低了監管合規成本。

在RWA的場景下，L3可以是針對某個具體業務的網絡，如供應鏈金融業務，而對應的Checkers就對應Centrifuge裏的SPV，這樣SPV可以方便地review每筆交易是否合法，合法的話才允許其進入共識流程。

# ~~2.5. 经济学设计~~（暫不考慮）

~~我们主要考虑L2的经济学设计，发行L2币。~~

~~L3的经济学可由各个L3自己设计，它们可以发行自己的代币。~~

## ~~2.5.1 L2币的用途如下：~~

* ~~L2节点需抵押L2币后才能成为L2层共识节点。~~
* ~~L3节点如果希望提交propose，需在L2节点抵押L2币。~~
* ~~作为向L2提交交易的gas费。~~

~~为了保证安全性，可以在初始时直接给内定节点大量币，如4个节点，每个1kw的币。后续其他节点想参与进来时，可以商量购买币的价格（一级市场），如1token = 1 USD，这样我们就直接在L1获得了至少4kw USD质押的安全性。这样在原理上是安全的。通过币获取途径的中心化来控制进入L1共识的节点数量和质量，在发展早期保证系统的安全性。随着影响力的发展，可以逐步放开L2币的销售，由市场定价，同样可以获得足够的安全性。~~

## ~~2.5.2 费用市场~~

~~各个L3节点提交待选区块、状态根等数据到L2时，需要使用L2币作为gas费才能够被打包进块，而对一个块共识的过程中，还需要L2有大量交易提交。~~

~~假设有10个L3链，每个链上有5个builder参与竞争；每个L3链指定的共识节点数量为6个。则一条链要达成共识，需要的最少交易数量为：~~

* + ~~3（超过一半节点提交propose，最少数量为3个）~~
  + ~~4\*3=12 （每个propose至少4个checker确认）~~
  + ~~4(selected变为confirmed，需要至少4个Verifier确认)~~

~~所以单链需要最少19个交易，10条L3需要19\*10=190笔交易。假设所有运行效率都是最高的，则平均一个L3块在2个L2块内达成共识，假设这些交易平均分布，则需要至少190/2=95 TPS方可达成运行目标。现实中不可能达到完美的平均，而且L2上也可能会有其他类型的交易，因此对L2的tps要求更高。因此需要竞争才能让自己的交易被打包至L2上。~~

## ~~2.5.3 RollUp 成本~~

~~RollUp成本由L2 Validator层承担，是向L1以太坊提交交易的成本，预计一次提交会修改好几个插槽数据（EVM的slot），同时包含部分calldata。该成本使用ETH支付。~~

~~我们使用Validium进行rollup（即交易数据不上L1，只提供获取方法和证明），这样能减少成本。~~

~~RollUp成本需要尽可能压缩，可以考虑：~~

1. ~~在需要时才执行rollup，如设定两个值：time limit和tx limit。即当到达一定时间（如1h）或L2/L3交易数量达到一定量时才执行rollup（可选两者到达任意一个都rollup）~~
2. ~~将rollup成本分摊到L3用户层，即动态调整L2层gas的基础费用，让基础费用能cover rollup的成本，所以L3的Builder可能就得多付gas费，进一步的，L3 builder可以提高L3层的基础gas费，让用户来最终cover这部分成本。这也是用户应该付的。~~

## 2.6. 跨鏈(僅考慮外部跨鏈，所以前半部分沒用)

~~跨链是个复杂的工程，根据不同维度来看有不同区分方法。我们需要先找到自己的定位。~~

~~从针对的目标来看，可以分为：~~

* 1. ~~针对特定资产的跨链。它将特定类型的币通过去中心化网络验证，跨到目标链。例如将BTC跨到以太坊的BitGo网络。~~
  2. ~~针对特定DApp的跨链。它为单个DApp提供跨链服务，DApp一般是做借贷、交易类的DeFi程序。这种相当于是有项目方的跨链。例如ThorChain，Chainflip。~~
  3. ~~专门针对某条链的跨链桥。这类跨链桥专门服务于一条链的跨链需求，一般是链的项目方推出的。其他链想要与其跨链时，可以实现相关服务。例如Avalanche Bridge、Polygon Bridge 和 Rainbow Bridge 等~~
  4. ~~通用跨链基础设施。这类项目可能是协议，也可能是网络；可以传输除了代币之外的其他内容，如数据、跨链合约调用、流动性整合等。例如Layerzero。~~

~~我们的目标是c：针对某条链的跨链桥。~~

~~从跨链资产类型来看，可以分为：~~

* 1. ~~原生资产跨链。这种的作用跟交易所一样，就是A链上的原生资产置换成B链上的原生资产。~~
  2. ~~替代资产跨链。这种是指A链上的某种代币，跨链到B链上时，不是原生代币（因为没办法是原生的），而是创建一个可以代表A上代币的替代代币，该替代代币与A链上跨出来的代币可以1：1兑换。例如将BTC跨到以太坊上时，是在以太坊上新发出一种WBTC代币，该代币可以按1：1转回比特币网络。~~

~~我们的目标是b。~~

~~从实现技术来看，可以分为：~~

* 1. ~~原子交换，即哈希锁。~~
  2. ~~外部验证。即单个或多个独立的第三方对跨链交易进行验证。~~
  3. ~~本地验证。即由跨链参与方直接验证，一般针对具体DApp的验证会使用这种方案。~~
  4. ~~原生验证。即跨链双方在对方链上通过部署合约、节点等方式，使对方链成为自己的轻客户端，在出块时完成对自己链的验证。Cosmos是这种方案的升级。cosmos把轻客户端从1对1升级成1对N，通过一条专门的HUB链来完成对所有对接链的轻客户端验证，降低了跨链成本。~~
  5. ~~原生验证+外部验证。原生验证的成本太高，因为它要在一条链上建立另一条链的轻节点，这会带来非常多提交，而且很多提交可能是无用但必要的。因此可以考虑一些综合方案，如Layerzero使用第三方预言机+超轻节点。~~

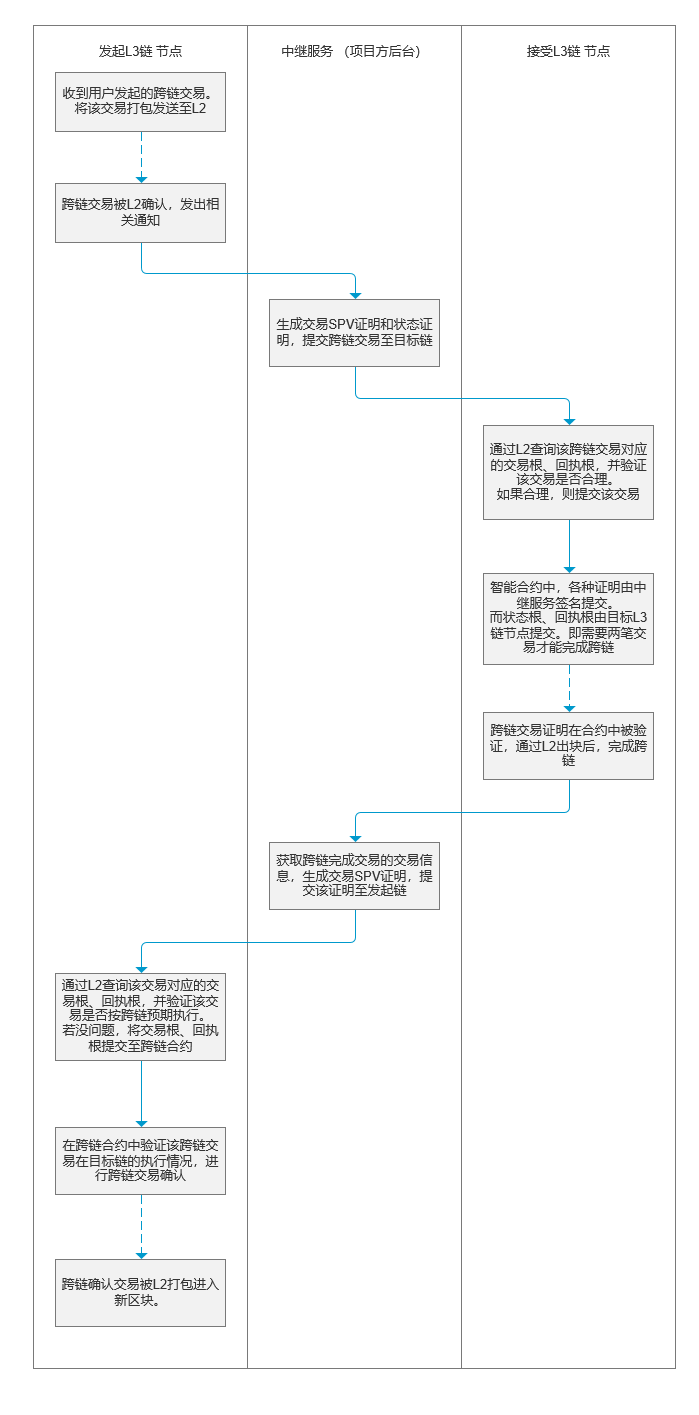
~~海峡链内部两个L3之间的跨链，使用原生验证方案，L2块和节点充当验证者，作为中介完成安全验证。（即代替Cosmos HUB地位）~~

~~对外跨链时，使用LayerZero方案，可以综合保证安全性和成本。~~

~~因此综合来看，海峡链跨链目标为：~~

~~专门针对海峡链自身推出一套跨链机制，无论是特定token，还是DApp，均可使用该机制完成资产跨链。跨链时会在原始链冻结相关资产，同时在目标链上发出等价的替代资产。~~

## ~~5.1 内部跨链~~

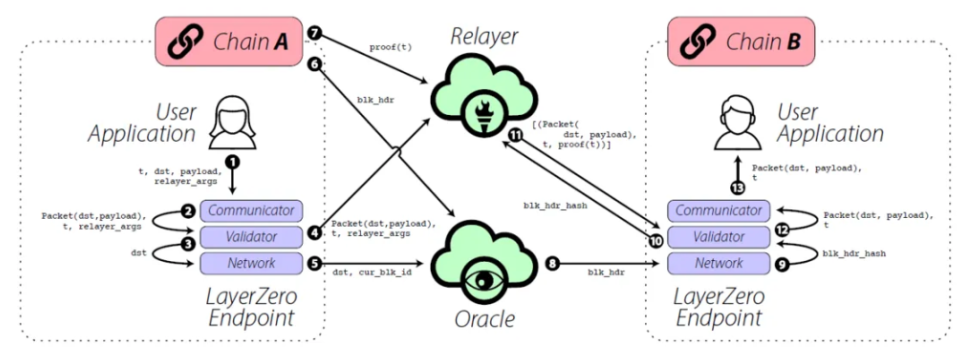
~~~~

~~该方案的关键是：目标链并不是完全信任中继的，但信任自己链上的节点。因此中继只能提供跨链的交易和状态proof。而对应的状态根和回执根由节点自行查找L2节点后提交入智能合约，并在目标L3链上，于合约内完成对proof的验证。因此只有目标链大部分节点作恶，且中继器合作作恶时，才会出现安全问题。~~

## 2.6.2 外部跨鏈

外部跨鏈對與DeFi結合的RWA應用來説是至關重要的，因爲流動性均來源於其他鏈，如Ethereum等。所以必須完成對外跨鏈，才能與流動性池對接。這裏可以參考Centrifuge的方案，其方案是合理且高效的。

~~外部跨链中，直接使用LayerZero的方案，该方案简单安全，且可适配性高。~~

~~~~

~~跨链时，发起链的块头、交易根由第三方预言机（如ChainLink）提供给目标链，而状态转换证明由Relayer提供，在目标链的智能合约中完成对跨链交易有效性的验证。因此只有当Oracle和Relayer合谋作恶时，才会发生安全问题。而这个难度是非常高的。~~

~~使用该方案，无需持续在链上提交另一条链的区块头，只需在需要时通过预言机提交即可，因此成本可控。且两条链间的对接比较容易。成本主要在预言机和合约验证上。~~

## 2.7 DA層

DA層在本區塊鏈中至關重要，因爲區塊數據等很多都是通過DA層傳播和保存。DA層需要滿足如下要求：

* 可信。長久保存在DA的數據需要保障其不被刪除。
* 高性能。儅服務于很多場景和L3時，可能DA節點的數量會非常多，因此我們需要高效的路由算法快速找到正確的DA節點
* 安全和權限管理。鏈上數據不是無條件公開的，需要擁有可信的權限管理方案。
* 同時服務于所有網絡，節點。這對IPFS來説不是問題。

詳細的DA層設計因爲細節較多，因此不在此展開，需要的話可以進一步溝通。

## 2.8. 其他

### 2.8.1 多虛擬機支持

因為L3與L2之間是以propose和L2區塊方式，使用網絡協議溝通，因此L2並未對L3有虛擬機限制。但Verifier需要驗證L3的區塊，需要與L3使用的虛擬機一致。因此考慮在共識合約中記錄虛擬機類型，L3和L2均會按照共識合約中約定的類型使用虛擬機。

當L3使用非EVM虛擬機時，需與L2溝通達成一致，要求L2支持該虛擬機，否則無法完成Verify。

# 現實版設計方案

目標版方案雖然能力較强，但是實現成本稍高，儅我們實際應用時還是需要著眼於現實，以合理的成本服務與自己的業務目標。因此現實版是一個容易落地的實用版本。

針對RWA場景，Checker是必須保留的角色，其他的基本都可以刪減。

網絡不用分層，只需使用L2一條鏈即可，等業務量大不夠用了再擴展到L3。

具體實現上，可考慮在EVM兼容的現有開源鏈（如EVMOS）上做二次開發，添加進Checker（即SPV）,添加方法也很簡單 ，只需對所需交易要求多方簽名即可。

DA層可以直接使用IPFS，對文件頭加密，配合鏈上合約管理密鑰和權限即可。

安全性方面，有兩個措施，可以都用：

1. 因爲使用POS，所以可以通過控制發行的token來控制出塊權。比如初始4個節點，預發行4000w的token，分給這四個節點，讓其抵押出塊。控制流通到市面的token總量，讓其不足以挑戰初始節點的出塊權即可。這樣在一開始是比較中心化的，後續隨著鏈的發展規模變大了，再逐步增加去中心化程度。
2. 作爲RollUp，將數據錨定到Ethereum。使用Validium可以極大地減小成本，成本是可控的。

另外，其實目標版的實現難度也不大，主要都是些智能合約的開發，對節點的改造不會很多。裏面唯一難一些的是Verifier，但Verifier也可以通過直接運行L3節點，或盲目信任的方式規避。