三、研究計畫內容（以中文或英文撰寫）：

(一) 研究計畫之背景。請詳述本研究計畫所要探討或解決的問題、研究原創性、重要性、預期影響性及國內外有關本計畫之研究情況、重要參考文獻之評述等。如為連續性計畫應說明上年度研究進度。

1.1 研究計畫之背景

衍生性金融商品是金融產業風險管理的重要工具，截至 2021 年底，利率和外匯場外衍生性金融商品的總市值超過 10 萬億美元。在雙邊交易的場外交易中，衍生性金融商品也扮演很重要的角色，但其雙邊交易的性質，讓其本身具有合約違約的風險，像是交易方的信用風險、違約風險和信用評級降低的風險。目前大部分的研究仍無法建構一個有效機制來完全消除這些風險；再者，每個衍生性金融商品的合約定義都不相同，尤其是處理交易流程的程序上，雙方的衍生性金融商品合約都必須嚴謹定義交易流程的程序。更進一步來說，合約需要定義清楚如何確認現階段的價值、抵押程序、保證金和提前贖回終止的程序。若沒有明確的定義，將會導致衍生性金融商品在交易過程中，產生許多不確定性和交易效率低落的問題，進而造成更高的交易風險。

另外，近年來金融業透過集中式管理的方式來降低違約風險，像是歐洲市場基礎設施監管機構 (EMIR) 針對場外衍生性金融商品進行清算程序，清算程序是交易雙方需要與第三方機構(例如：清算所)，簽訂清算合約，合約內會記載現金流量處理、市場價值及違約處置的方式。然而，市場上越來越多人討論清算所這樣的機制是否合理？因為清算所本身也會面臨清算會員破產的風險，尤其是在壓力市場的環境中。清算會員流程的有效性和透明度及清算資金如何分配之程序是具有爭議性的。

1.2 研究計畫之目的

為了解決上述的問題，本計畫提出應用區塊鏈智能合約技術實現衍生性商品契約交易平台。區塊鏈智能合約是以數位化方式產生具法律約束型態的契約形式，由於區塊鏈智能合約的執行具有不可逆的特性，所以需要明確定義合約的事件與合約的狀態，假如明確定義合約的內容，那麼交易雙方的的資訊不對稱風險與違約風險都會大幅降低。區塊鏈智能合約具有精準自動化執行的特性，不會受到任何時間與空間的干擾，加上合約是由機器自動執行，機器會按照原先定義的合約事件與狀態做出適合的處理程序，因此這種自動執行與無需人工參與合約執行的特性，將可以提高交易效率與降低交易過程的風險。再者，區塊鏈智能合約是以分散式帳本技術(DLT)為基礎來打造，更進一步來說，就是沒有中心化的角色參與並協助驗證交易的過程，任何新合約的產生與轉移都需要整個交易網路的參與者共同驗證及確認其交易的正確性。

1.3 重要參考文獻之評述

衍生性金融商品

衍生性金融商品是指價值由利率、匯率、股價、指數或其他利益及其組合等所衍生之交易契約，其契約型態主要有遠期契約(Forward)、期貨(Futures)、交換契約(Swap)及選擇權(Option)等四類，隨著契約型態與連結標的之不同，可以產生各式各樣衍生性金融商品。隨著時間的堆疊，更發展結合固定收益商品與衍生性金融商品之結構型商品，例如信用連結債券和連結衍生性商品之證券化商品。衍生性金融商品之交易場所主要有交易所及 店頭(Over the Counter, OTC) 市場。交易所交易之衍生性金融商品皆為標準化(或準標準化)商品，由交易所按市價評估結算，透過認可會員資格及建立交易規則來規範交易行為及解決交割紛爭等事務，是屬於較安全的交易環境。但因為交易所交易之商品規格皆需標準化，導致金融商品創新速度較慢。目前美國芝加哥商品交易所(Chicago Mercantile Exchange, CME)是全球成交量最大的金融期貨與商品期貨交易所。OTC 市場交易又稱為場外交易，交易商品可 以客製化並較少交易規範，以個別議價方式完成交易，只是這種交易制度易衍生流動性與交易對手風險和缺乏透明度等問題。OTC 市場之監理制度較為寬鬆，提供商品創新之有利條件與參與者可交易的商品也較多，故目前全球衍生性金融商品之交易量，OTC 市場規模遠較交易所市場規模為大。金融市場主要功能，在促使資金進行最有效率之使用與分配，而金融機構扮演資金需求者與供給者間媒介之角色，其業務經營面臨市場、信用、作業及法律等諸多風險。衍生性金融商品之設計，其目的在於幫助市場合理分散風險。然而，近年來衍生性商品不斷創新，交易規模快速成長，金融機構操作衍生性金融商品之目的，除了避險及資產配置需求外，還包括提高資產組合預期收益、進行投機交易來獲取高額利益，並將資產證券化商品包裝出售，賺取豐厚手續費收入等。

區塊鏈

比特幣在推出後因為透明、安全的資產轉移機制備受重視，主要的原因是因為區塊鏈技術和分散式帳本技術的去中心化、開放性、獨立性與安全性構成比特幣得到足夠的信任，作為加密貨幣可以與法定貨幣或者商品進行交易，中本聰於 2008 年發表比特幣白皮書(Nakamoto, 2008)，其中闡述如何利用 P2P 網路技術實現點對點交易、加密技術如何在資料透明的情況下避免偽造竄改，區塊鏈技術吸引人的特性在於，以往的資料或者數位資產均需要機構集中管理，代表機構必須累積足夠的信任，並在大量資料集中管理的作業中背負人員疏失的風險，而區塊鏈技術的去中心化解決了這些問題，去中心化也就是區塊鏈用戶間的「共識機制」，當在分散式帳本技術的基礎上創造的區塊，所有的使用者因為分散式帳本而均擁有這些區塊，當其中一位用戶持有的資訊遭到竄改時，透過共識機制確認這份被竄改的資料與其他使用者的區塊不吻合，而否決被竄改資料的真實性，進而達到防止竄改、去中心化的目的，在傳統的伺服器、資料庫平台中，因為資料集中儲存處理、中心化的關係，當受到資訊安全或者硬體設備出現錯誤等問題，均會受到巨大的財務、信任度上的損害，而去區塊鏈技術因為避免了這些風險而不會遇到這些阻礙。而上述所解釋的比特幣基礎技術，其在

分散式帳本技術下產生的理念-去中心化，結合加密技術、P2P 網路技術、時間戳等等，形成區塊鏈技術為基礎的數位貨幣，而區塊鏈技術大致被分為三個發展過程：區塊鏈 1.0，即是做為數位貨幣的基礎技術，以比特幣作為例子來說，在分散式帳本技術的基礎下，賦予數位貨幣「去中心化」、「難以偽造」、「可溯源」、「匿名性」的特性，定義其作為貨幣的特性，並且擁有可不透過第三方轉移資產的協議，完成點對點的資產交易。區塊鏈 2.0 主要談論智能合約，也就是根據程式碼編寫出來的程式，作為合約，在滿足智能合約程式設定的條件後，會自動執行而不透過第三方人員的作業流程，這個方式可以用來保存記錄某項資訊以供認證，或者利用智能合約篩選進行點對點的契約媒合，最具代表的平台為以太坊。區塊鏈 3.0 被認為是更貼近生活的概念，加強智能合約的應用範圍，如：金融、醫療、政府、藝文創作領域等，也因此區塊鏈 3.0 必須克服區塊鏈 1.0 或 2.0 當用戶使用量增多，所需運算量不足而導致降低交易速度，服務品質不良的問題。

以太坊與智能合約

以太坊是於 2014 年推出，使用類似於比特幣節點的工作量證明(Proof-of-Work，PoW)與共識機制，進行數據儲存、執行智能合約並維護其網路的區塊鏈分散式系統(Tikhomirov, Voskresenskaya, Ivanitskiy, Takhaviev, Marchenko, & Alexandrov, 2018)，以太坊中的節點一般是參與其網路的電腦或伺服器，並且透過工作量證明與共識機制，在節點試圖協助驗證交易並建立新的區塊時，將會有一定的以太坊加密貨幣(ETH)生成並給予該節點，該種獎勵機制使得以太坊系統有穩定的參與者節點進行維護，開發者能夠使用合約語言(Solidity)編寫智能合約腳本，將其編寫為以太坊虛擬機(Ethereum Virtual Machine，EVM)支持的合約腳本，並於部署合約時根據合約之複雜度給予一定量的 Gas 作為費用，並由節點透過 EVM 執行合約。智能合約可以根據開發者的設計邏輯，可以實現利用合約儲存數據、交付虛擬資產、供其他使用者執行合約簽署等功能，並且可於編寫實現至合約訪問與簽署的條件，限定特定身份或條件參與。以太坊開放式的區塊鏈網路與利用數位機制產生的數據信任，使得智能合約衍生出數據儲存、合約簽署、金融契約等等商業模式應用，開創以智能合約為重要角色的數位經濟。

(二) 研究方法、進行步驟及執行進度。請分年列述：1.本計畫採用之研究方法與原因。2.預計可能遭遇之困難及解決途徑。3.重要儀器之配合使用情形。4.如為須赴國外或大陸地區研究，請詳述其必要性以及預期效益等。

2.1 開發工具與執行環境

本計畫預計以 Metamask 區塊鏈錢包管理應用程式、Ethereum-Remix 智能合約編譯器、以太坊 Rinkeby 測試網路及以太坊區塊瀏覽器 Etherscan，實現衍生性商品契約交易平台。本計畫所使用的開發工具如下所示：

- MetaMask 區塊鏈錢包管理工具

MetaMask 是基於 Google Chrome 瀏覽器的瀏覽器擴充套件，也提供使用者用各種驗證方式(例如：硬體設備、助記詞、金鑰檔案等)管理區塊鏈錢包之私鑰和數位資產，並提供與 Dapp 進行合約互動等功能之應用程式。透過 MetaMask 用於智能合約的開發將有助於減少開發者基礎建設的構建成本，並能在智能合約開發過程中，快速與合約進行互動測試，即時監看智能合約與區塊鏈網路的互動情形，包括發起交易時是否按照智能合約條件執行，查看以建立的合約是否變數正確、連結 Etherscan 查看合約是否正常生產，數位資產有無正確自動發放等。本計畫預計透過 MetaMask 產生四名合約參與者之區塊鏈錢包帳戶，1. 衍生性商品契約交易平台：負責認定發售人與購買人身份並為其註冊地址身份之可信任第三方。2. 發售人：由管理員註冊並於該平台發起衍生性商品契約者。3. 購買人：符合購買人身份並由管理員註冊，可參與合約簽署者。4. 銀行方：負責確認資金到帳狀況並於完成後簽署已確認資金到漲者。

- 以太坊測試鏈網路(Rinkeby)

以太坊區塊鏈網路是專門給開發者測試智能合約之可行性的測試環境，因為若使用以泰坊主網路進行合約的部署及互動，都需要給予手續費(Gas)。開發者為了減少開發成本，在將合約部署至主網路之前，都會將其部署至測試網路進行測試，現存運作的測試網路有 Ropsten、Rinkeby、Kovan 和 Görli，而測試過程所消耗的手續費(Gas)都可以按照測試鏈所提供的管道進行領取。以本計畫的 Rinkeby 測試環境為例，需先創建一個區塊鏈錢包並取得錢包地址，先於社群軟體中撰寫貼文，貼文內容為先前創建的錢包地址，並複製該貼文的連結網頁位址後，再於 Rinkeby Authenticated Faucet 網站，貼上該社群貼文網址送出並等待驗證，驗證通過後，再將區塊鏈錢包切換到 Rinkeby 測試網路後，就會收到轉入的測試幣，如圖 1 所示。此後，只要於 Rinkeby 測試網路進行合約的開發及部署就可以使用測試幣作為手續費(Gas)支付。

圖 1：操作 MetaMask 確定測試幣轉入目標錢包帳戶

- Remix Ethereum-IDE 智能合約語言編寫工具

智能合約的撰寫需以合約程式語言(Solidity)來開發合約腳本，並將其部署至 Rinkeby 測試網路，提供參與者對合約進行簽署或觸發合約撰寫的功能。而撰寫合約程式語言(Solidity)的工具就是 Remix Ethereum-IDE 線上編譯器。開發者可以在 MetaMask 與 Remix Ethereum-IDE 線上編譯器的互動之下，透過 MetaMask 所連結的帳戶，撰寫合約語言並部署至區塊鏈網路進行功能測試，該編譯器提供給開發者合約部署、呼叫合約 Function、呼叫變數與合約代碼檢查等諸多功能，有助於開發者減少測試合約功能時所需額外使用其他方法呼叫合約的開發成本，如圖 2 所示。圖 3 為測試與執行合約功能介面，於此介面左側為設定部署環境及設定目標部署合約的功能，此部分可與 MetaMask 連結，使用當前登入的區塊鏈帳號進行部署。圖 4 為部署合約至測試網路後，可於左側功能區對已部署合約進行函式與變數的呼叫，對合約內容的變數格式、函式是否正確執行進行測試，以及是否按照撰寫時設定的合約條件運作。

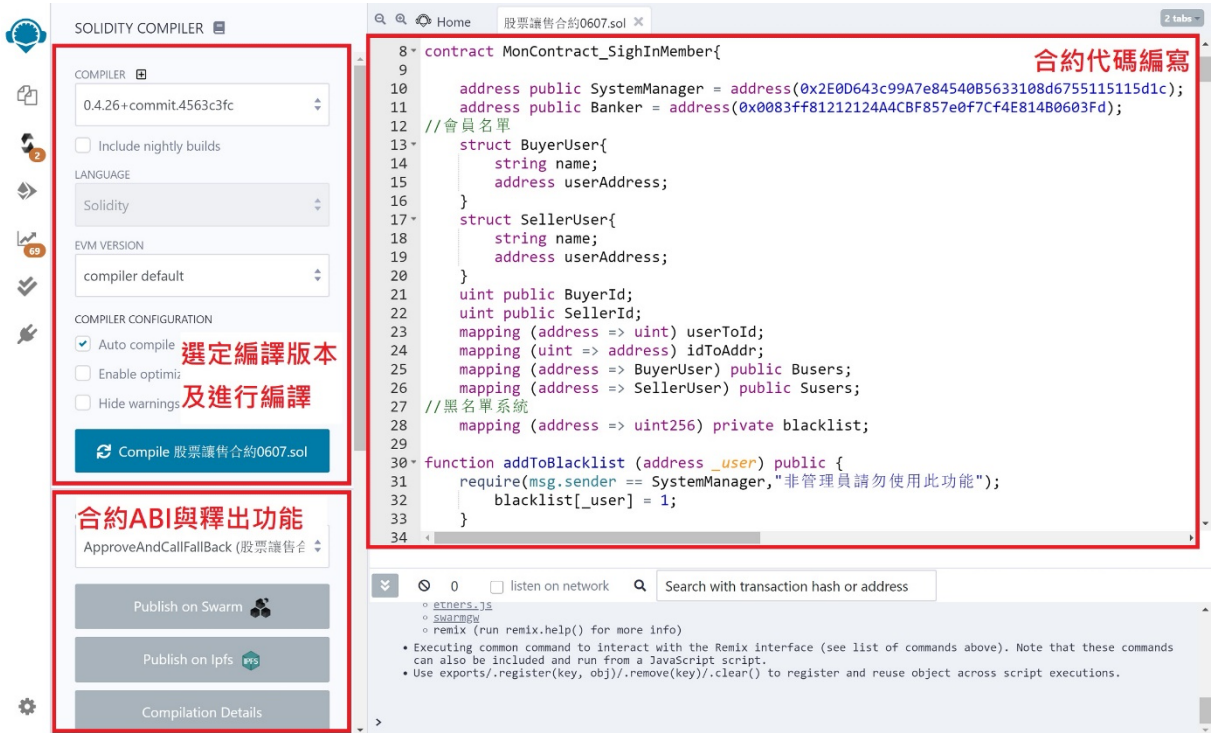

圖 2：Remix Ethereum-IDE 的 Solidity 語言編譯介面

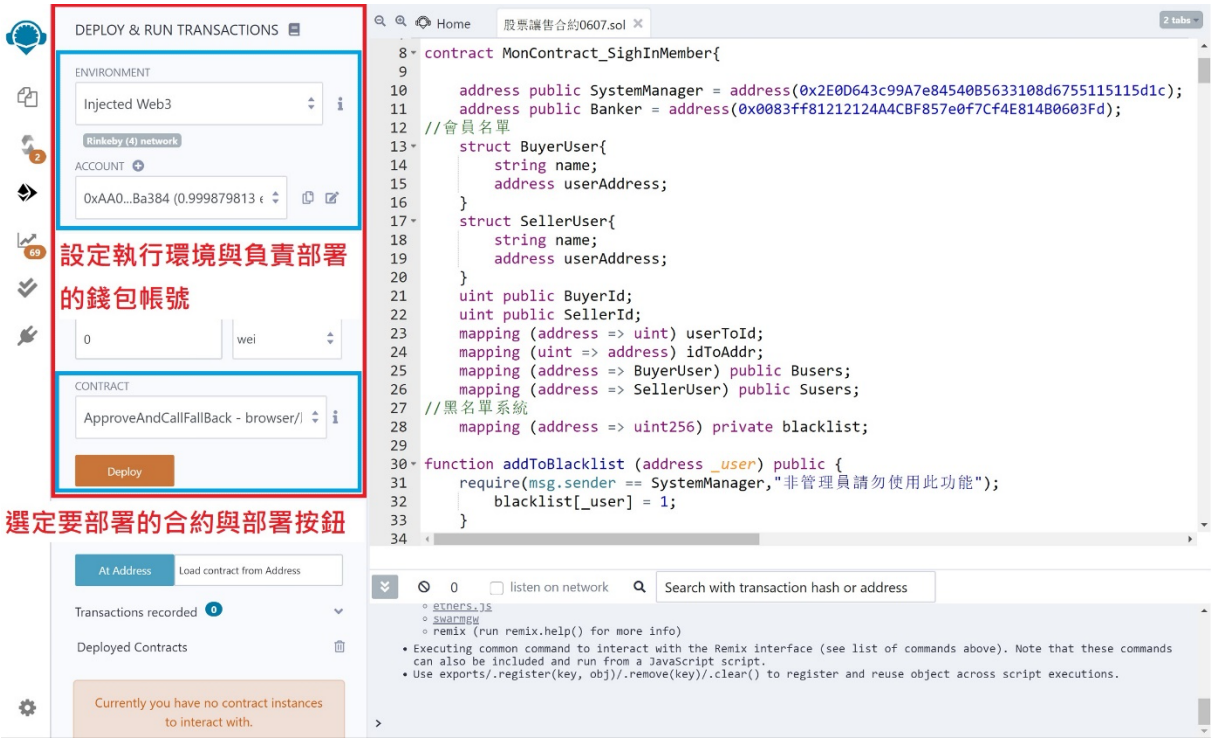

圖 3：智能合約部署功能介面

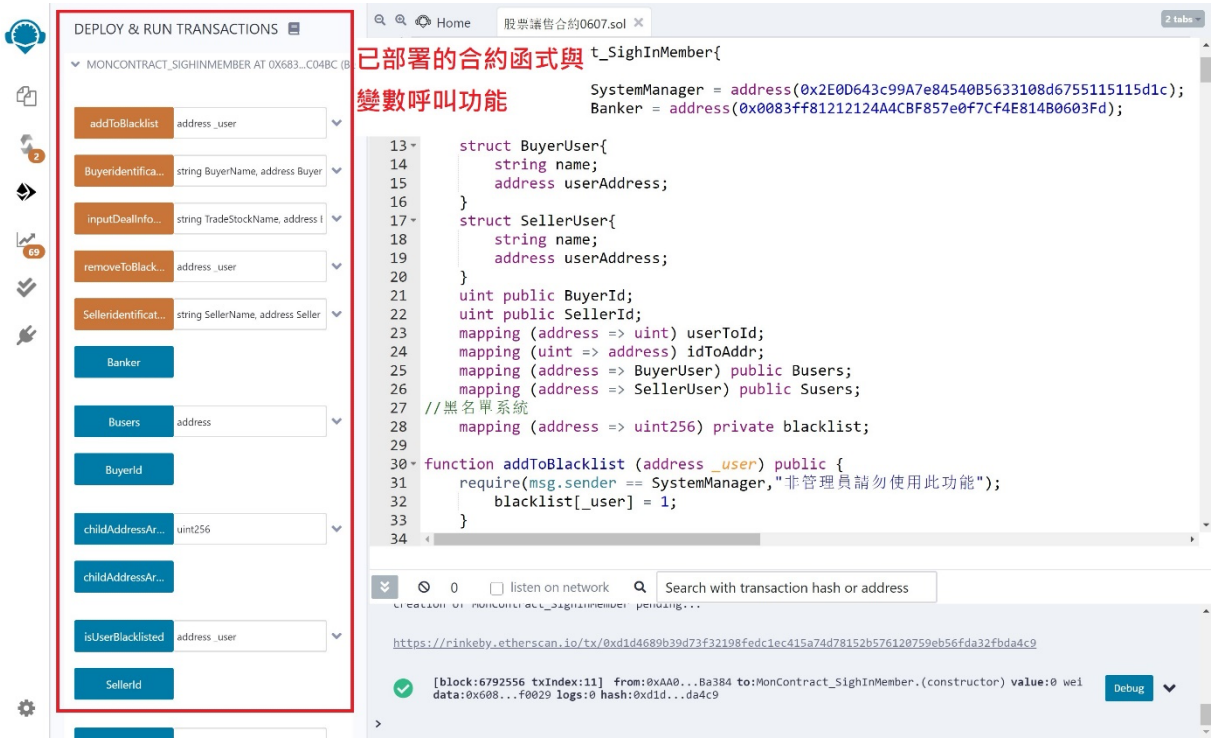

圖 4：函式與變數呼叫功能介面

2.2 衍生性商品契約交易平台運作架構

由發行人與購買人，持錢包帳戶地址分別向衍生性商品契約交易平台申請平台使用資格，發行人應申請賣方資格，購買人應申請買方資格，經核可註冊後，發行人即可於衍生性商品契約交易平台發起衍生性商品契約交易機制，如圖 5 所示。發行人發起之衍生性商品契約，內容必須包括如下：

1. 發行人所發行之衍生性商品的名稱
2. 合約發行人之地址(亦即發行人之錢包地址)
3. 合約購買人之地址(亦即購買人之錢包地址)
4. 合約交易商品數量
5. 交易之商品單位價格
6. 購買人繳納股款之期限

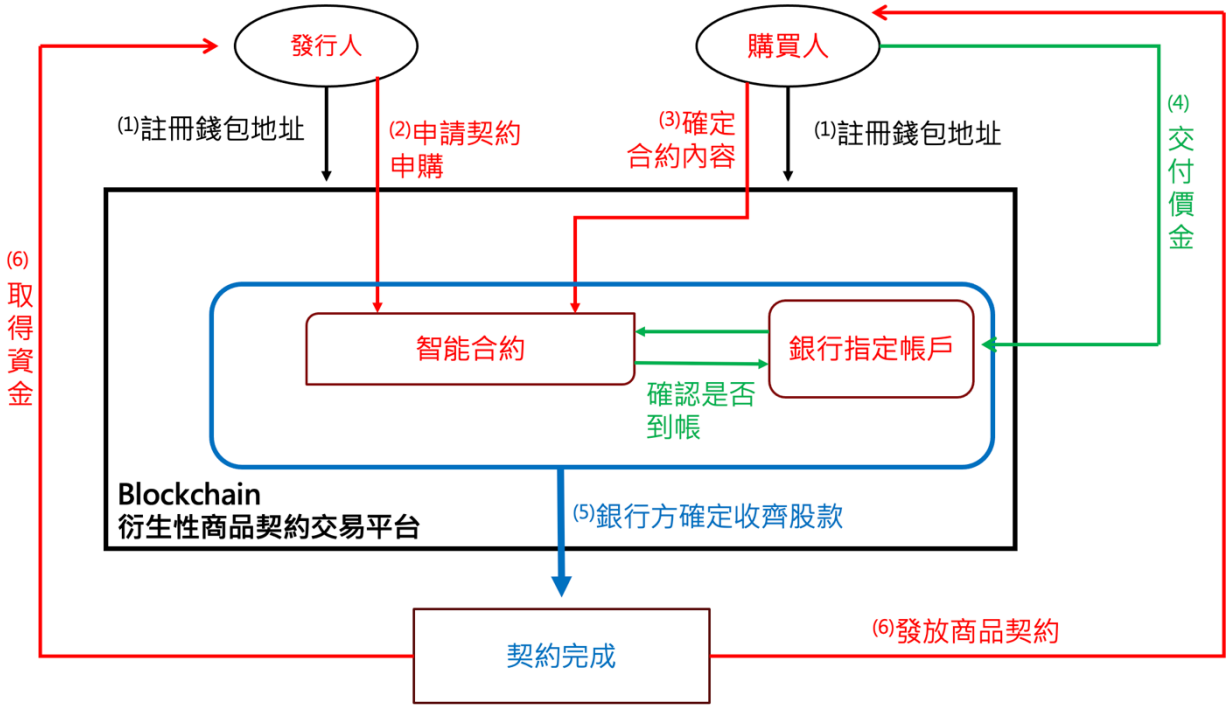

圖 5：衍生性商品契約交易平台運作架構

2.3 智能合約運流程

申請契約經發行人簽署發起後，合約內容中指定之購買人即成為受該合約授權簽署之唯一者，除指定之購買人簽署該合約之外，其他使用者若試圖進行簽署，一律會被合約拒絕，當受指定之購買人簽署後，合約將認定雙方皆同意合約內容之交易標的及交易標的之價格，此時將會記錄雙方同意時間點之時間戳記，購買人須於規定之時限內將款項匯入銀行專戶並由銀行方確認後對該合約簽署，否則合約將失效。當於交付款項時限內，銀行確認完成簽署，即確立該衍生性商品契約已經完成，並且平台會紀錄契約成立之時間戳記，平台立即根據合約標的之名稱和交易股數發放商品契約，商品契約會立即轉入購買人之錢包帳戶。

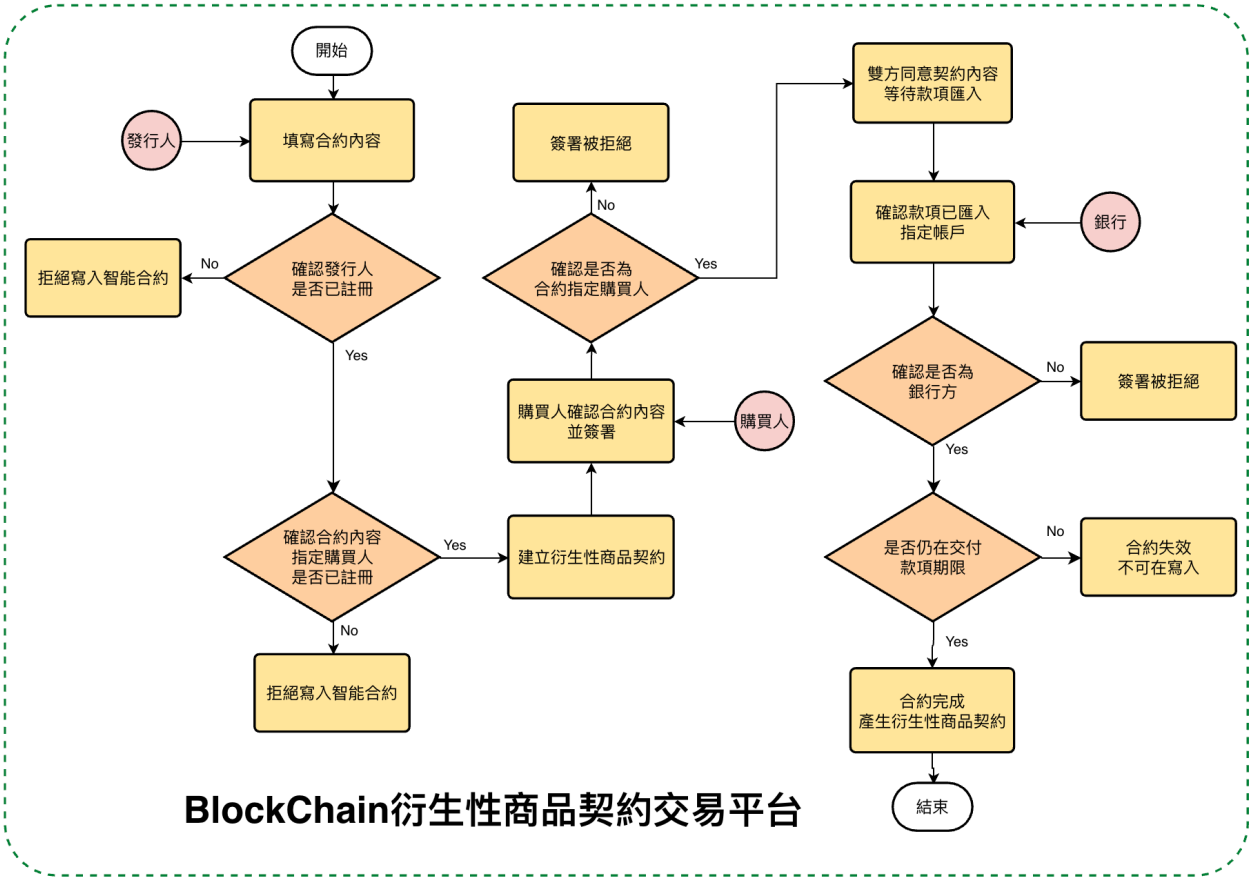

圖 6：智能合約運流程

2.4 智能合約之設計

本計畫撰寫智能合約所用語言是 Solidity 語言，智能合約語言可以控制使用者對合約進行內容簽署與寫入時，限定寫入的變數形式(如指定只得輸入數字或帳戶地址等)，以及限定某些功能只得特定使用者進行操作，因此 Solidity 語言得以確保合約流程不被有不法意圖的第三方干擾，以本計畫開發之智能合約為例，當平台管理員對發行人之區塊鏈錢包帳戶註冊至交易平台後，經智能合約檢視發行人帳戶已註冊，才得以發起衍生性商品契約；再者，發起之契約內容，須指定購買人之區塊鏈錢包帳號，並且經發起的衍生性商品契約，只有契約指定的購買人錢包帳號才可以進行簽署。綜合上述，智能合約可以成立不受第三方介入的數位契約，不僅提升契約簽署內容的透明度，也降低非必要之風險，而且智能合約所帶來的資訊保障與架構透明度，也提供產生契約糾紛時，具有足夠的舉證效力。下列為部分的合約內容之函式功能。

- 平台主合約：MonContract

主合約為平台儲存每個發行人與購買人的使用資格及建立並儲存子合約的重要基礎，契約的運作是必須先連結到主合約，才得以操作及功能並連接到子合約來查看契約內容，因此 MonContract 代表整個 Blockchain 衍生性商品交易平台的後端基礎架構，並以此延伸至子合約及其他對應之功能。

Function： Buyeridentification () 註冊購買人使用資格

此功能為註冊購買人使用資格至平台主合約，以 require 方法限定平台管理員使用該功能，註冊所需輸入資料為購買人名稱(資料類型指定字串類型)、購買人錢包帳戶地址(資料類型指定地址 Address 類型)，以此功能註冊後，會將購買人名稱與對應地址以 Struct 結構寫入儲存於已部署之主合約，並且於呼叫 Busers()時得以輸入地址查詢，再以 mapping 方法查詢對應的購買人名稱。

```
function Buyeridentification(string memory BuyerName, address Buyer)
    public{
    require(msg.sender == SystemManager,"非管理員請勿使用此功能");
    BuyerId++;
    idToAddr[BuyerId] = Buyer;
    userToId[Buyer] = BuyerId;
    Busers[Buyer] =  BuyerUser({
        name: BuyerName,
        userAddress: Buyer
        });
    }
```

圖 7：註冊購買人使用資格

Function： Selleridentification () 註冊發行人使用資格

此功能與 Buyeridentification ()方法雷同，為註冊發行人使用資格至平台主合約，以 require 方法限定平台管理員使用該功能，註冊所需輸入資料為發行人名稱(資料類型指定字串類型)、發行人錢包帳戶地址(資料類型指定地址 Address 類型)，以此功能註冊後，會將發行人名稱與對應地址以 Struct 結構寫入儲存於已部署之主合約，並且於呼叫 Susers()時得以輸入地址查詢，再以 mapping 方法查詢對應的發行人名稱。

```
function Selleridentification(string memory SellerName, address Seller)
    public{
    require(msg.sender == SystemManager,"非管理員請勿使用此功能");
    SellerId++;
    idToAddr[SellerId] = Seller;
    userToId[Seller] = SellerId;
    Susers[Seller] =  SellerUser({
        name: SellerName,
        userAddress: Seller
        });
    }
```

圖 8：註冊發行人使用資格

Function：inputDealInfomation () 建立衍生性商品契約

此功能為建立衍生性商品契約至平台主合約中，以 require 方法限制合約發起人必須為已註冊使用資格之發行人帳號，使用此功能必須輸入衍生性商品的名稱(資料類型指定字串類型)、指定的購買人的錢包地址(資料類型指定地址 Address 類型，且必須為已註冊使用資格之購買人帳號)、欲售出衍生性商品數量(資料類型指定數字類型)、欲售出衍生性商品價格(資料類型指定數字類型)、購買人繳納款項期限(資料類型指定數字類型)，發起後會於平台

建立子合約 ChildContract 並於主合約新增子合約編號與對應的子合約區塊鏈地址，若使用此函式時未符合 require 之條件(發起合約的發行人或指定購買人無使用資格)，則會跳出錯誤訊息，拒絕寫入數據。

```
    uint public childAddressArray1Id;
    address[] public childAddressArray1;

    function inputDealInfomation(string memory TradeStockName, address BuyerAddress,
        uint ShareAmount, uint SharePrice, uint DateTime)
        public
        returns(address SellerAddress, address BankerAddress){
        require(msg.sender == Susers[msg.sender].userAddress,"你使用的帳號沒註冊過");
        require(blacklist[BuyerAddress] == 0 && blacklist[SellerAddress] == 0);
        if(BuyerAddress == Busers[BuyerAddress].userAddress){
        SellerAddress = msg.sender;
        BankerAddress = Banker;
        address childAddress = (new ChildContract_DealInfomation)(TradeStockName,BuyerAddress,
        SellerAddress,ShareAmount,SharePrice,DateTime,BankerAddress);
        childAddressArray1.push(childAddress);
        childAddressArray1Id++;
        }else{
            require(msg.sender == SystemManager);
        }
    }
}
```

圖 9：建立衍生性商品契約

- 子合約(私募合約)：ChildContract

子合約會儲存發行人使用 InputDealInfomation ()功能所寫入的資料內容，並以 Constructor 方法導入子合約，子合約兼具購買人簽署同意和銀行方核可款項到帳，並且會根據合約簽署狀態的進度變化，記錄其簽署時間、契約完成時間及顯示當前合約狀態，並於合約完成後顯示衍生性商品契約的契約錢包地址。

```
constructor (string _TradeStockName, address _BuyerAddress, address _SellerAddress,
    uint _ShareAmount, uint _SharePrice,  uint _DateTime, address _BankerAddress)
    public{
    TradeStockName = _TradeStockName;
    SellerAddress = _SellerAddress;
    BuyerAddress = _BuyerAddress;
    BankerAddress = _BankerAddress;
    ShareAmount = _ShareAmount;
    SharePrice = _SharePrice;
    TotalValue = _ShareAmount * _SharePrice * 1000;
    DateTime =  _DateTime;
    // LegalSeller = _LegalSeller;
    // LegalBuyer = _LegalBuyer;
    DealState = "合約撰寫中，等待買方確認內容";
}
```

圖 10：子合約相關資訊

Function：DealCreationTime() 購買人簽署同意契約內容

此功能為衍生性商品契約指定之購買人，應操作其合約指定地址的區塊鏈錢包帳戶，呼叫 DealCreationTime()對契約進行同意契約內容的簽署動作，並且該函式會於簽署後，記錄簽署合約所產生的區塊時間戳，以此來記錄購買人同意合約的時間，並且智能合約會以該時

間為計算款項應於何時繳納完成之依據，當此函式呼叫成功後，合約狀態會由「合約撰寫中，等待買方確認內容」改變為「購買人已確認，進入等待匯款」。

Function：DealTimeCheck() 銀行簽署款項到帳

此功能為銀行方確認股款到帳功能，呼叫此函數的使用者必須為銀行專用區塊鏈錢包帳戶(已於智能合約代碼中指定，如同管理員帳戶不得更改)，該函式被成功呼叫後會記錄呼叫時間，並且此功能函式呼叫後會根據發行人當初於合約設定的繳納款項期限，計算自購買人簽署同意合約內容至今，是否超過合約所訂之時間，若時間逾期，合約交易狀態將會顯示「合約失敗，可能是未按時匯款」，並且自始無法再對合約呼叫任何函式，而發行人與購買人只能重新締約，且失敗的合約資訊依然會留存在區塊鏈網路上，只要輸入合約地址，還是可以查看。若銀行方操作呼叫此函數並成功通過，交易狀態則會顯示「款項已確認匯入，合約完成」，合約被系統認定完成的同時，會根據契約資訊中的交易標的、交易量、合約完成時間等變數，將衍生性商品契約全數轉移至購買人的區塊鏈錢包帳戶。

```
    function DealTimeCheck()
        public returns (address){
        require(msg.sender == BankerAddress);
        require(DealCreateTime != 0);
        require(FinishTime == 0);
        FinishTime = now;
        uint a = FinishTime - DealCreateTime;
        uint b = DateTime * 1 days;
        if (a < b) {
            ERC20GreenLight = bool(true);
            DealState = '款項已確認匯入，合約完成';
            FinishInfoTradeStockName = TradeStockName;
            FinishInfoSellerAddress = SellerAddress;
            FinishInfoBuyerAddress = BuyerAddress;
            FinishInfoShareAmount = ShareAmount;
            FinishInfoSharePrice = SharePrice;
            StockDealToken newToken = (new StockDealToken(TradeStockName,ShareAmount,FinishTime));
            childAddressArray2.push(address(newToken));
            newToken.transfer(address(FinishInfoBuyerAddress),uint256(ShareAmount)*1000);
            newToken.DealTime();
            return address(newToken);
            emit getDealState(DealState);
            emit getDealuint(FinishTime);
        }
        else{
            ERC20GreenLight = bool(false);
            DealState = '合約失敗，可能是未按時匯款';
            emit getDealState(DealState);
            // emit getDealuint(FinishTime);
        }
    }
}
```

圖 11：銀行機構確認款項到帳

(三) 預期完成之工作項目、成果及績效。請分年列述：1.預期完成之工作項目。2.對於學術研究、國家發展及其他應用方面預期之貢獻。3.對於參與之工作人員，預期可獲之訓練。4.預期完成之研究成果及績效（如期刊論文、研討會論文、專書、技術報告、專利或技術移轉等質與量之預期績效）5.本計畫如為整合型研究計畫之子計畫，請就以上各點分別說明與其他子計畫之相關性。

本計畫預計用 1 年的時間，開發 1 套應用區塊鏈智能合約技術實現衍生性商品契約交易平台，以數位化方式產生具法律約束型態的契約形式，由於區塊鏈智能合約的執行具有不可逆的特性，因此在生成區塊鏈智能合約時，需要明確定義合約的事件與合約的狀態。假如明確定義合約的內容，那麼交易雙方的的資訊不對稱風險與違約風險都會大幅降低。區塊鏈智能合約具有精準自動化執行的特性，不會受到任何時間與空間的干擾，加上合約是由機器自動執行，機器會按照原先定義的合約事件與狀態做出適合的處理程序，因此這種自動執行與無需人工參與合約執行的特性，將可以提高交易效率與降低交易過程的風險。本計畫預期完成之平台示意內容及工作事項，如下列所示：

1. **衍生性商品契約交易平台：**首頁為顯示已生成之衍生性商品契約列表，當前空白表示未有衍生性商品契約被發起，而右上方則會顯示當前 MetaMask 所連結的區塊鏈錢包帳戶，以及該帳戶與區塊鏈測試網路之互動紀錄。
2. **註冊發行公司與應募人使用資格頁面：**本頁面上方為為註冊發行人使用資格功能，此前端功能與智能合約函式 Selleridentification ()相互鏈結；本頁面下方為註冊購買人使用資格功能，此前端功能與智能合約函式 Buyeridentification ()相互鏈結。平台成功呼叫函式後，發行人和購買人之使用資格將會以區塊的方式進行打包並登錄於系統中。
3. **衍生性商品交易契約頁面：**此頁面為發行人取得平台使用資格後，得於此頁面建立衍生性商品交易契約，此前端功能與智能合約函式 inputDealInfomation ()相互鏈結，發行人必須於該頁面輸入交易標的名稱、指定之購買人之錢包地址、商品交易數量、商品之價格及應繳納款項之期限，按下功能鍵呼叫函式後，將會將生成之衍生性商品契約顯示於平台首頁。
4. **衍生性商品交易契約細部交易資訊頁面：**進入此頁面後，交易平台會透過合約地址，抓取合約內的資訊變數，呈現合約內容、當前交易狀態、合約簽署時間及完成時間，並且此頁面會隨區塊鏈智能合約簽署狀況進行資訊更新。
5. **等待購買人確認交易內容頁面：**此頁面為等待購買人確認內容，對合約簽署同意其內容。購買人應詳細檢視本頁面後，於頁面右下角按下「同意」功能鍵，此功能與智能合約函式 DealCreationTime()相互鏈結，呼叫此函式成功後，將記錄雙方合意時間、通知應繳納款項之期限，並且合約狀態更新為購買人已確認並等待匯款之狀態。
6. **銀行方簽署確認股款到帳頁面：**此頁面為購買人已確認內容並等待匯款，銀行方於購買人繳納款項後，操作此頁面簽署確認到帳，銀行得檢視本頁面之合約狀態、款項總額及

應繳納款項期限，使用頁面右下角「確認已到帳」功能鍵簽署，此前端功能與智能合約函式 DealTimeCheck()相互鏈結，呼叫此函式成功後，將記錄合約完成時間，並且根據智能合約條件顯示合約成功或失敗的合約狀態描述。

7. **使用 Etherscan 查看契約交易記錄頁面：**點選首頁內「查看合約」按鈕後，即跳轉至 Etherscan 中查看已部署於區塊鏈網路中的衍生性商品契約，Etherscan 的功能主要為讓所有區塊鏈上的購買人可以查看有多少契約已經完成。當合約紀錄可以使用 Etherscan 查詢時，代表該訊息數據已被發送至區塊鏈網路並上鏈成功，讓購買人確定契約是否是否真實存在於區塊鏈網路，增加購買人對於交易平台的信任度及保持交易的透明度，降低交易不存在的信任風險。

參考文獻

Alao, O., & Cuffe, P. (2021, June). Towards a Blockchain Weather Derivative Financial Instrument for Hedging Volumetric Risks of Solar Power Producers. In *2021 IEEE Madrid PowerTech* (pp. 1-6). IEEE.

Arusoaie, A. (2021). Certifying Findel derivatives for blockchain. *Journal of Logical and Algebraic Methods in Programming*, *121*, 100665.

Collins, R. (2016). Blockchain: A new architecture for digital content. *EContent*, *39*(8), 22-23.

Gorkhali, A., & Chowdhury, R. (2021). Blockchain and the Evolving Financial Market: A Literature Review. *Journal of Industrial Integration and Management*, 1-35.

Himeur, Y., Sayed, A., Alsalemi, A., Bensaali, F., Amira, A., Varlamis, I., ... & Dimitrakopoulos, G. (2022). Blockchain-based recommender systems: Applications, challenges and future opportunities. *Computer Science Review*, *43*, 100439.

Liu, J., Xu, Z., Li, R., Zhao, H., Jiang, H., Yao, J., ... & Chen, S. (2021). Applying blockchain for primary financial market: A survey. *IET Blockchain*, *1*(2-4), 65-81.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

Rouhani, S., & Deters, R. (2019). Security, performance, and applications of smart contracts: A systematic survey. *IEEE Access*, *7*, 50759-50779.

Schär, F. (2021). Decentralized finance: On blockchain-and smart contract-based financial markets. *FRB of St. Louis Review*.

Scharfman, J. (2022). Decentralized Finance (DeFi) Compliance and Operations. In *Cryptocurrency Compliance and Operations*(pp. 171-186). Palgrave Macmillan, Cham.

Sunny, J., Undralla, N., & Pillai, V. M. (2020). Supply chain transparency through blockchain-based traceability: An overview with demonstration. *Computers & Industrial Engineering*, 106895.

Tikhomirov, S., Voskresenskaya, E., Ivanitskiy, I., Takhaviev, R., Marchenko, E., & Alexandrov, Y. (2018, May). Smartcheck: Static analysis of ethereum smart contracts. In *Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain* (pp. 9-16).

Zheng, X., Zhu, Y., & Si, X. (2019). A survey on challenges and progresses in blockchain technologies: A performance and security perspective. *Applied Sciences*, *9*(22), 4731.