CyberMiles：商業交易的下一代區塊鏈協議

By 5xlab

技術白皮書

v1.5

[商業情境及代幣銷售條款另文討論]

免責聲明

本文為概念性文件（下稱「**技術白皮書**」），旨在說明我們提出的 CyberMiles 區塊鏈協議及其網路發展方向。我們可隨時修改或替換此技術白皮書，但並無更新或為收受者提供更多資訊的義務。

謹敬告讀者如下：

非向所有人提供：並非所有人都可以使用 CyberMile 平台及 CyberMiles 代幣。參與其中可能需要經過一系列步驟，包括提供某些資訊及文件。

未於任何司法管轄區提供管制產品：CyberMiles 代幣（如本技術白皮書所述）無意在任何管轄區構成證券或其他管制產品。本技術白皮書不構成任何形式的募股公開說明書或報價文件，亦無意在任何管轄區構成證券或其他管制產品的買賣要約或邀請。本技術白皮書並未經過任何司法管轄區的主管機關審核。

未提供建議：本技術白皮書並未建議您參加 CyberMiles 平台或購買 CyberMiles 代幣，亦不應成為制定契約或購買決策的依據。

未聲明或保證：就本文件內或企劃中以其他方式傳播的資訊、陳述、意見或其他事項，我們並未聲明或保證其準確性或完整性，亦不限制、不聲明或保證前瞻性或概念性陳述的完成度或合理性。本文件的內容均非為亦不得作為對未來的承諾或陳述。在相關法律允許的最大範圍內，如果因任何人根據本技術白皮書或其任一部分行事導致發生損失或損害（無論是否可預見），無論為疏忽、過失或缺乏注意，我們概不負責。若責任範圍可予以限制但無法完全免除，則以相關法律允許的最大限度為其限制範圍。

其他公司：除 CyberMiles 基金會（簡稱「基金會」）及 5miles LLC（「**5miles**」）外，使用其他公司及／或平台的名稱、商標並未暗示雙方具從屬關係，或受其背書。本技術白皮書提及具體公司及平台時，僅供說明之用。

您必須聽取所有必要的專業意見，包括稅務及會計處理等。我們希望 CyberMiles 企劃大獲成功，但並不敢保證，而數位資產及平台自有其風險，您必須評估自己的風險及承擔能力。

執行摘要

區塊鏈技術在商業應用中的前景不可限量。然而，目前一代區塊鏈的執行效率過低，開發人員的生產力也不足，因此在一般商業交易中運用不廣。在本文中，我們提出了一項新的區塊鏈網路協議，稱為 CyberMiles 區塊鏈，專為商業契約交易的優化所設計。

我們的解決方案是一項協定的創新方案，可以使區塊鏈上的分散式虛擬機器存取成熟的商業中間軟體技術堆疊。這個新的區塊鏈的效能及擴展性極高，每秒可支援超過 10,000 件交易。企業可藉以編寫智慧商業契約（Smart Business Contracts），也就是可編纂商業規則及流程的分散式中間軟體應用程式。該網路的本地加密貨幣「CyberMiles 代幣」（CMT）可用於結算交易、獎勵網路驗證者（智慧商業契約的執行者），並激勵社群成員相互提供服務。

CyberMiles 區塊鏈有一項獨特的優點：它的部署將可支援 5miles 既有的電子商務網路上超過 1000 萬名美國註冊用戶，以及每年預估超過三十億美元的交易，這等於可以立即建立世界上最大的區塊鏈商業網路。該網路可以提供非集中式用戶身分及信用管理、非集中式結算清算中心、端對端投票及衝突解決方案等各種服務。在網路平台上的應用實例包括非集中式的個人資訊「電子錢包」、端對端小企業貸款，以及同行爭議仲裁。

目錄

1. 介紹

1.1Bitcoin 與 Ethereum

Bitcoin 是區塊鏈技術的第一個殺手級應用。通稱「區塊鏈 1.0」的 Bitcoin 網路主要是一種非集中式帳本系統，擁有內建的非集中式共識機制。雖然透過 UTXO 技術可以寫出程式在 Bitcoin 網路上運行，但低階 UTXO 程式的容量非常有限。這是一種圖靈（Turing）不完備的程式設計環境，非常不易於使用，因此 Bitcoin 網路還是以非集中式的帳本系統為主，以少數社群開發的應用程式記錄 Bitcoin 交易。

Ethereum 企劃旨在建構區塊鏈 2.0。由於加入了圖靈完備（Turing complete）的虛擬機器（稱為「Ethereum 虛擬機器」或「EVM」），可支援名為「智慧契約」的第三方指令碼，在達成特定條件的前提下使代幣／加密貨幣在帳戶之間移動（使用案例之一是以智慧契約為代管帳戶），使得 Ethereum 區塊鏈可望成為「世界電腦」。這類智慧契約是由 Ethereum 節點即時執行，其後再由礦工（或驗證者）進行驗證，儲存至區塊鏈中。

此外，Ethereum 也支援「非集中式應用程式」（Decentralized App，又稱 DApp）。DApp 運作於區塊鏈之外，但可以在區塊鏈中呼叫智慧契約方法。在典型的設定中，DApp 可以是為相應的智慧契約提供 UI 的網路應用程式。

1.2 主要問題及相關工作

不過，目前的區塊鏈技術仍然有效率過低、開發人員生產率過低兩大公認的問題。

作為非集中式系統，區塊鏈網路需要許多獨立、非合作式的節點，才能反覆執行相同的計算任務，接著才能對「true」達成共識。這使得系統效率甚低，難以擴展，因為隨著網路規模的擴大，計算工作量也呈幾何級數增加。正因為擴展性／效能問題，區塊鏈網路上允許的第三方計算任務也必須有所限制，而這又回過頭來成為開發人員缺乏經驗、生產力不足的肇因。其結果是 Ethereum 智慧契約 DApp 目前還無法廣泛應用。

目前有幾項解決方案可望解決區塊鏈技術的效能及擴展性問題。

- 新的共識機制：無論是目前的 Bitcoin 還是 Ethereum 區塊鏈，使用的都是稱為「工作量驗證」（Proof-of-Work，PoW）的共識機制，目的在於將不受信任的參與者隔絕在網路之外，不過這種共識機制效率非常低。經過無數的努力後，出現了一種更有效的機制，稱為「權益證明」（Proof-of-Stake，PoS），可用以取代 PoW。這個領域主要的競爭者有 Tendermint 的拜占庭容錯演算法（Byzantine fault tolerance，BFT）共識引擎，以及 Ethereum 自己的 CASPER 解決方案。

- 網路的分區：擴大網路規模一種常見的方法是將網路分區，成為若干個子網路。接著只要加入更多子網路，就可以水平地擴展整個網路。不過子網路必須在非集中式區塊鏈網路內相互傳訊，對彼此的狀態達成共識，這項工作遠比一般的資料

庫分區化還要困難。這個領域較佳的解決方案包括 Cosmos 區塊鏈網際網路（Cosmos Internet of Blockchains）以及 Polkadot 網路。

- 鏈外計算：效能問題還有一種更直接的解決方案，就是將大部分的重量級計算任務移出區塊鏈，只使用區塊鏈共識機制來記錄計算結果。目前在這個領域也已經作過許多實驗，從 Lightning Network 的鏈外狀態通道、Plasma 的欺詐證據側鏈，到 TrueBit 的鏈外 Ethereum 智慧契約交易框架。

在本文中，我們並未嘗試解決區塊鏈擴展性的基本問題。我們相信，透過社群共識一定會及時出現理想的解決方案。未來的區塊鏈網路必然會納入這三種方法，達成高效能、高擴展性的目標。

不過，解決這些問題之後，區塊鏈網路仍然需要吸引並支援企業應用程式開發人員，才能進入商業市場。在本企劃中，我們的目標是提出一種架構解決方案，使區塊鏈網路（智慧契約）上的第三方企業應用程式更有力量，同時也更易於開發。

1.3 更理想的智慧契約

作為第一代的技術，同時也基於上述的擴展性／效能因素，Ethereum EVM 及 DApp 非常難以使用。我們的目標是顯著改善 EVM 以及相關的軟體堆疊，讓開發人員更容易上手，也讓企業更容易運用。

- 智慧契約通常需要由區塊鏈外部的事件觸發。就 Ethereum 而言，必須以「oracle」來提供外部世界具代表性及確定性的狀態資訊。oracle 是一種脆弱的解決方案，因為它沒有經過標準化，而且可以在不知會智慧契約的情況下變更。

- 智慧契約只能鬆散地與 DApp 中間軟體偶合。如果不「知道」DApp 裡有什麼，智慧契約就不能呼叫 DApp 內的軟體元件。由於使用圖靈完備的程序性程式語言撰寫複雜的規則極為困難，多數智慧契約只能執行簡單的商業交易規則。

- DApp 中間軟體無法進行封裝及重複使用。DApp 開發人員必須作成架構決策，撰寫一次性的應用程式。

- DApp 中間軟體未與區塊鏈的加密貨幣激勵系統進行整合。DApp 節點必須貢獻鉅量的計算資源，卻得不到加密貨幣作為獎勵。這導致了 DApp 只能由公司以集中的方式運行。

2. 建議的解決方案

為了解決 Ethereum 的缺點，並建立以區塊鏈為基礎、適合商業開發人員建立非集中式應用程式的「虛擬機器」，我們提出了一種新的區塊鏈協定來支援我們所謂的「智慧商業契約」。此協定不僅包含一個虛擬機器，也定義了區塊鏈以外的中間軟體軟體堆疊（目前是由 DApp 以非標準的方式處理）。區塊鏈中各節點不但會運行區塊鏈帳本，也支援標準化的中間軟體。

借鑒於過去成功的企業軟體劇本，關鍵不在於建立全能的虛擬機器或程式設計語言，而在於建立可重複使用的軟體元件大型程式庫，然後使整體軟體堆疊標準化。Linux 操作系統就是一個很好的例子。直到社群以數千種商業友善的軟體套件擴展核心操作系統，接著 Fedora／RedHat 又將堆疊標準化之後，企業用戶才開始廣泛採用。過去的例子還有 Java2 企業版平台、LAMP 堆疊、Ruby on Rails 平台。這些企業平台的優勢在於他們標準化了程式庫及框架。

軟體封裝及重複使用是企業軟體中最重要的最佳實踐方案，現在該把這個實踐方案應用在區塊鏈平台上了。

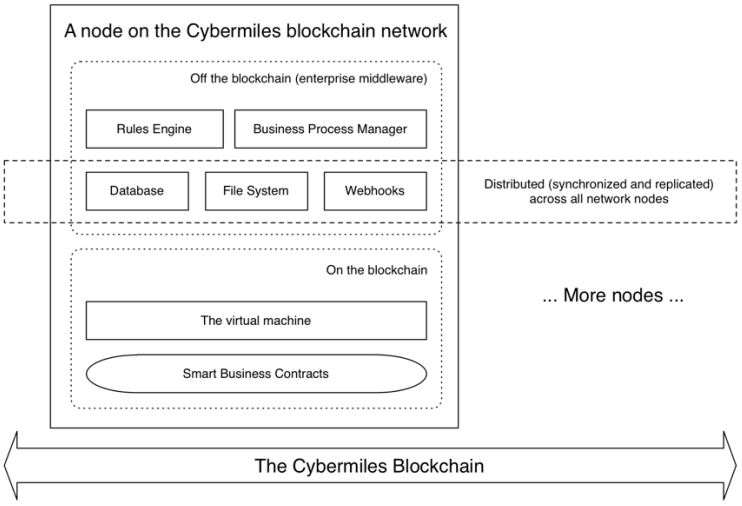

圖 1. 顯示了 CyberMiles 區塊鏈的整體架構。如您所見，區塊鏈之外有大量可重複使用的軟體元件。

2.1 智慧商業契約

CyberMiles 區塊鏈上的「智慧商業契約」類似於 Ethereum 區塊鏈上的「智慧契約」，由區塊鏈節點執行，並在建立新區塊時由礦工進行驗證。智慧商業契約的結果會儲存在新的區塊上。

不過，CyberMiles 智慧商業契約與 Ethereum 智慧契約之間的主要區別在於：智慧商業契約可以存取強大的商業中間軟體的整合性堆疊，而不必從頭開始編寫一個個應用程式，因此智慧商業契約本身很容易開發，而且完全可以重複使用。

由於智慧商業契約是區塊鏈的一環，執行它所需的計算能力（包括運行整個外部企業中間軟體堆疊的工作）可以計為 CyberMiles 系統的虛擬貨幣——CyberMiles 代幣（CyberMiles Token，CMT）。進行交易的網路用戶會以 CMT 向網路驗證者支付小額的交易費用，作為保證資料完整性的報酬。

2.2 中間軟體堆疊

智慧商業契約可以在區塊鏈本身之外存取商業軟體框架。這些軟體框架嵌入於運行區塊鏈的每個節點中，會在每次執行智慧商業契約、區塊鏈礦工驗證結果時運行。Cyber Miles 系統內的企業中間軟體框架堆疊包括以下內容。

- 規則引擎：大多數商業契約必須遵守某些規定。相較於一般的程式設計語言，專屬的規則引擎既易於使用又有效率，已經有許多企業採用。

- 商業流程管理器（BPM）：BPM 系統是一種可模擬多步契約執行狀態的狀態機，由締約方的外部動作驅動，而且通常是以規則引擎來確定下一步。

- 非集中式資料庫：非集中式資料庫是支援複雜的應用程式框架及儲存應用程式資料所必需。這種資料庫會在區塊鏈接上的節點間進行複製與同步，交易結果不會儲存於此，而是會儲存在區塊鏈本身之內。

- 非集中式檔案及資料儲存服務：智慧商業契約及相關的中間軟體服務需要存取檔案服務，以便管理決策所需的大型資料檔案。

- 非集中式 Webhook 服務：作為必須與履行契約義務的外部實體（如電子商務應用程式的 FedEx 送達通知）交流的商業系統，我們會建立可以接收智慧商業契約相關外部事件的非集中式 Webhook 系統。

智慧商業契約包含複雜的規則、流程、資料及 Webhook，不過我們還需要一個程式將所有元件整合在一起並規劃作業。這需要通用且圖靈完備的程式設計語言，而 Cyber Miles 虛擬機器會與區塊鏈軟體一起傳送至各網路節點，使我們可以辦到這一點。

2.3 商業就緒契約範本

智慧商業契約的一個關鍵面向是：契約不僅建構在可重複使用的軟體元件之上，而且它們本身也可以重複使用。由於大多數商業交易情境都已經定義完善（從法律及商業角度而言），所以確實可以建立智慧商業契約的範本，只需要更改關鍵詞作為參數（如契約當事人名稱、日期、金額等）就可以重複使用。範本資料庫可以降低建構與部署商業應用程式的成本，提高網路本身的價值。

2.4 附智慧商業契約的非集中式應用程式

CyberMiles 區塊鏈系統還有一種非集中式應用程式（DApp）的概念。基於隱私或效能因素而不能儲存在區塊鏈內的所有資料與邏輯，可以由 CyberMiles Dapp 管理。商業交易中的相關商業邏輯可以完全卸載至智慧商業契約。

3. 技術

CyberMiles 區塊鏈的中間軟體堆疊技術解決方案相對而言較簡單，因為它們都是已經廣泛應用於商業中間軟體世界的成熟技術。我們正在建立一個工程解決方案，將這些技術納入區塊鏈框架內，並為系統功能設計適當的經濟激勵。

本文討論的具體技術框架僅供參考。社群之後可以自行舉行討論及投票活動，遴選技術管理委員會，然後由該委員會共同決定確切的 CyberMiles 中間軟體堆疊技術。

3.1 規則引擎

CyberMiles 系統會在區塊鏈驗證者軟體納入正向鏈結推理的規則引擎。規則引擎位於區塊鏈本身之外，但驗證者會用以執行智慧商業契約。

規則引擎會執行 Rete 演算法，以比對模式（商業操作的事實）與規則，並解決潛在的衝突。Rete 演算法非常複雜，不在本文的討論範圍之內。但我們要強調的是，目前以 Rete 為基礎的正向鏈結規則引擎已經頗為成熟，成果卓著，廣為企業所採用，Drools 與 Jess 就是很好的例子。

在概念層面上，正向鏈結規則是一組複雜的 IF 及 THEN 語句。規則引擎可以提供一種特殊的「程式設計語言」，讓商業分析師（而不是軟體開發人員）表達商業規則。下方的範例是一組以虛擬規則語言撰寫的規則，說明如何根據買方的資料決定產品的定價。在這種情況下，如果買家的 FICO 得分高於 740，就會獲得兩折的價格優惠。

```
Rule “Pricing”
   dialect “mvel”
   when
      m : Message(status==Message.GET_PRICE)
   then
      when
         m.fico_score > 740
      then
         m.price = m.listed_price * 0.8
End
```

Dapp 叫用智慧商業契約後，智慧商業契約就會執行規則。請注意，「買方的資料」及產品定價所需的資料是擷取自 CyberMiles 平台的非集中式資料庫，我們會在 3.3 節進一步討論。

```
engine = load_rules(“pricing.rl”);

m = new Message ();
m.status = Message.GET_PRICE;
m.fico_score = 741; // Get from profile DB
m.listed_price = 100; // Get from product DB

engine.send(m);
return m.price; // Returns 80 to Dapp caller
```

這個例子很簡單，不過可以非常清楚地看出智慧商業契約是如何處理複雜的規則，並將大部分商業邏輯封裝在系統內。

3.2 商業流程管理器（BPM）

在大多數商業系統中，只有滿足特定條件時才能反應式地應用規則。例如只有在潛在買家詢價（例如載入產品詳細資訊網頁）時才會應用產品定價規則。以這點來說，規則引擎是「按需求」地叫用，系統大部分時間都是「待機」狀態。有限狀態機（FSM）可以模擬此動作。

有一種可執行 FSM 的常見企業軟體產品，叫做「商業流程管理器」（BPM）。BPM 也有自己的宣告式語言，可供商業分析人員指定程序。一個狀態可以對應一個商業規則，以決定如何觸發下一個狀態。BPM 語言通常以 XML 為基礎。從下方的範例可以看到 BPM 在典型的電子商務情境下可能會處理的狀態子集。

```
<process-definition name=”purchase process”>

  <start-state name=”request a purchase”>
    <transition to=”evaluate”/>
  </start-state>

  <state name=”evaluate”>
    <!--...-->
    <transition name=”approve” to=”approved”/>
    <transition name=”disapprove” to=”done”/>
  </state>

  <fork name=”approved”>
    <transition to=”decrement inventory” />
    <transition to=”credit seller” />
    <transition to=”deduct from buyer” />
  </fork>
```

```
    <state name="decrement inventory">
      <!--...-->
      <transition to="done" />
    </state>

    <state name="credit seller">
      <!--...-->
      <transition to="done" />
    </state>

    <state name="deduct from buyer">
      <!--...-->
      <transition to="done" />
    </state>

    <end-state name="done" />
</process-definition>
```

BPM 指令碼可以操縱變量、啟動或結束參數化任務，甚至引用外部規則引擎。

智慧商業契約指令碼中的程式設計現在可以簡化為一系列宣告敘述，用以檢查 FSM 狀態。

```
// pid is the ID of a process
// associated with a shopping session
// It is stored in the distributed DB
if (pid) {
    process = load_process (pid);
} else {
    process = start_process("purchase.bpm");
    pid = process.id;
    // Save pid to the DB
}
```

```
while (process.next()) {
   if (process.state == “credit seller”) {
// Do the transaction …
   }
   if (process.state==”deduct from buyer”) {
// Do the transaction …
   }
   … …
}
```

常用的企業中間軟體 BPM 解決方案有 jBPM、Enhydra Shark、OpenSymphony OSWorkflow。

3.3 非集中式資料庫

規則引擎及 BPM 都需要將內部資料儲存在資料庫內才能有效地運行。隨著商業應用不斷成長，日趨複雜，應用程式本身也需要在區塊鏈交易紀錄外管理資料。這需要的是嵌入於所有區塊鏈節點的資料庫。由於區塊鏈應用程式性質上屬於非集中式，因此節點也必須在所有區塊鏈節點之間進行複製以及同步。

很幸運地，非集中式資料庫技術近年來有了長足的進步，現在已經可以使用現成的開放原始碼軟體建立網際網路規模的非集中式資料庫。不過必須考慮的是，這些資料庫通常是 NoSQL，無法保證系統在任何時點都能維持一致性。相反地，它們的目標是「最終一致」，因為系統會逐漸解決潛在的衝突，這是與區塊鏈本身非常不同但很有效的衝突解決策略。

我們已經初步決定採用流行的 Apache Cassandra 作為 CyberMiles 預設的非集中式資料庫。

3.4 非集中式檔案系統

除了區塊鏈與資料庫紀錄外，智慧商業契約通常也必須檔案或資料 blob。這些檔案必須在區塊鏈的節點上進行複製及存取，因此需要非集中式的檔案及資料儲存系統。

CyberMiles 系統將使用區塊鏈友善的非集中式檔案系統技術（如 Ethereum 蜂群及 IPFS）作為標準檔案儲存設備。

3.5 非集中式 Webhook

如上所述，商業系統會反應外部事件。BPM 會等待締約方的輸入（稱為外部世界狀態的「oracle」）再叫用規則，以確定下一步該做什麼。達到下一個階段後，會等待再次輸入。因為電子商務的基礎設施位於網際網路之上，所以 CyberMiles 系統必須以一個介面來接收來自網際網路的事件。

為了辦到這一點，每個 CyberMiles 區塊鏈節點都會嵌入網站伺服器，可用以接收外部訊息並觸發 BPM 事件。每個智慧商業契約應用程式都可以發布一個或多個 Webhook 網址，以接收外部事件。區塊鏈上的有效節點會自動註冊於 DNS 系統，全都可以接收輸入的 HTTP 請求。

4. 區塊鏈

智慧商業契約可以將所有商業中間軟體元件結合在一起，並連結至保存在區塊鏈上交易帳本。繼 Ethereum 之後，CyberMiles 正在區塊鏈上建造圖靈完備的虛擬機器。虛擬機器可以透過一種類似 JavaScript 的指令碼語言（類似 Ethereum 的 Solidify 程式設計語言）進行程式設計，可以完成的任務包括將 Webhook 事件連結至 BPM 流程、載入商業規則，以及存取共享資料庫（見圖 1）。

針對智慧商業契約，開發人員必須組合應用程式代碼、BPM 配置檔案、Webhook 配置及商業規則檔案，成為單一的檔案庫，然後將封存的檔案提交至區塊鏈進行處理與部署。部署智慧商業契約後，外部系統就可以透過區塊鏈位址存取。例如 DApp 可以為應用程式建立一個 UI，它會利用智慧商業契約處理所有商業邏輯，並將產生的代幣交易記錄於區塊鏈。

4.1 區塊鏈及共識

針對 CyberMiles 系統的區塊鏈層，我們不想疊床架屋，而是要在現有的區塊鏈上建立框架。我們的基礎技術主要標準如下。

- 必須是社群驅動、積極開發的開放原始碼企劃。這可以使 CyberMile 改善基礎架構軟體、影響未來軟體的發展方向，並回饋社會。

- 軟體架構必須可以清楚區隔核心區塊鏈邏輯（即共識邏輯）與應用程式邏輯，以便驗證交易。共識引擎會處理在區塊鏈上提出交付新區塊的程序；自訂應用程式會驗證交易（包括執行智慧商業契約），並決定該在區塊鏈中記錄哪些交易。CyberMiles 虛擬機器以及智慧商業契約的整套軟體堆疊會作為自訂應用程式，寫入區塊鏈之上。

- 區塊鏈共識引擎必須已經證實其效能可以擴展至數百萬用戶的消費級應用。理想情況下，必須是公認的 Ethereum 擴展性頂級候選解決方案之一。換言之，在工程成熟度方面必須是市場領導者。

CyberMiles 團隊比較過現有的區塊鏈基礎設施解決方案，進行了深入的研究。初步的結論是我們會在 Tendermint／Cosmos 平台建立 CyberMiles 區塊鏈的第一個疊代。由圖 2 可以看到我們為 CyberMiles 區塊鏈上的驗證節點所提出的軟體架構。事實上，CyberMiles 已經在 Tendermint／Cosmos 平台有技術上的貢獻。

- Tendermint 企劃建立了「拜占庭容錯演算法」（BFT）共識引擎，這是一個非常活躍且資金充足的開放原始碼工作（繼本身也非常成功的 ICO 之後）。區塊鏈本身可以承受高達 1/3 的節點故障（毀損或破壞）。設定 DPoS（Delegated Proof of

Stake，委任權益證明）可強烈避免個人驗證者破壞網路，使拜占庭容錯演算法極少故障。

- Tendermint 擁有現代且模組化的架構。共識引擎可以獨立地插入其他類型的區塊鏈。例如 Ethermint 企劃就運用了 Tendermint 共識引擎擴展 Ethereum。ABCI（應用區塊鏈介面）是一個簡潔的應用邏輯介面，可使 CyberMiles 開發虛擬機器及應用程式推疊。新交易出現時，區塊鏈會透過 ABCI 將其傳送至 CyberMiles 應用程式；一旦開始執行相關的智慧商業契約，交易經過 CyberMiles 應用程式驗證後，就會傳回區塊鏈共識引擎儲存紀錄。

- Tendermint 是以 DPoS（委任權益證明）為基礎的高效能區塊鏈執行共識機制，獲 Ethereum 官方認可為 Ethereum 擴展性解決方案。在測試過程中，它可以穩定地支援每秒 10,000 件交易，使其成為領先的工程解決方案之一。

基於基礎 Tendermint DPoS 機制， CyberMiles 區塊鏈的區塊產生時間在 10 幾秒以下，區塊中的交易會在交付區塊後立即確認。

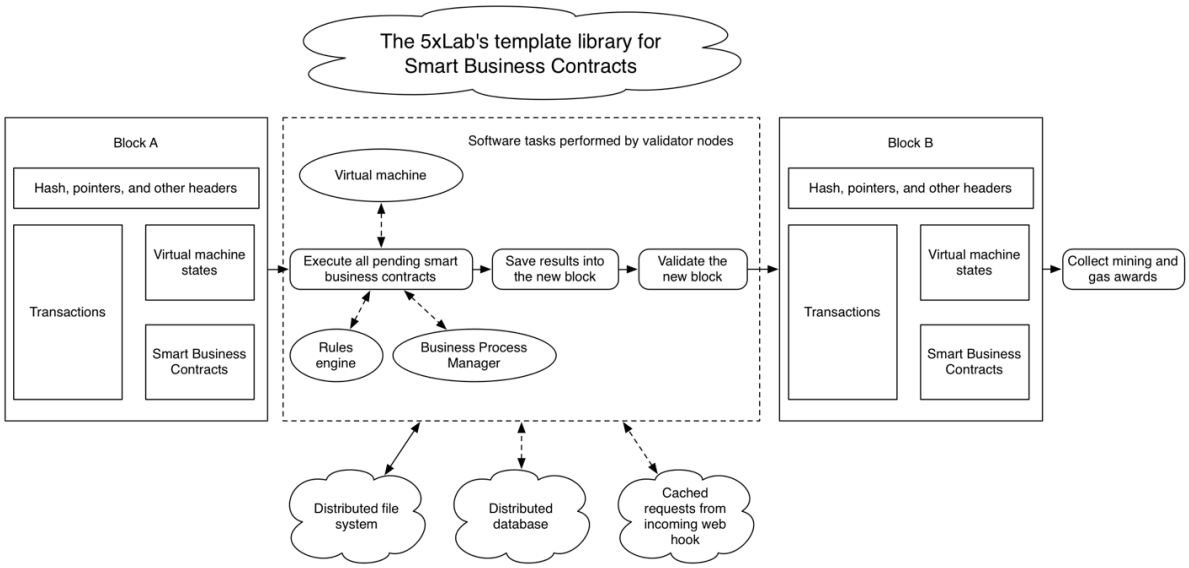

圖 2. 說明 CyberMiles 區塊鏈驗證者執行的工作。

4.2 加密代幣

區塊鏈會建立並記錄本地加密代幣，稱為 CyberMiles 代幣（CMT）。CMT 有兩種用途：一是作為社群成員提供服務的報酬，一是促進網路上的金融交易。這兩種使用其實息息相關，因為從每一筆交易結算收取的「手續費」會用以支付給提供服務、促進交易的人。以下我們會仔細討論這兩種情況。首先，網路參與者可以透過提供服務獲得 CMT。

- 他們可以成為驗證者，在網路上提供服務。具體來說，參加者會執行 DApp（如 5miles 應用程式）的智慧商業契約以獲得 CMT（由 DAapp 支付的「氣體」，見下文）。或者也可以驗證並記錄在區塊鏈上的新交易，並從 DPoS 共識協定賺得 CMT。

- 他們可以在網路上為同行提供服務。消費者及企業在網路上可能使用 CMT 支付對方的服務，如智慧商業契約的爭端解決仲裁服務，甚至開發服務

注意：CMT 會以動態匯率轉換為「氣體」（gas），使轉換後的氣體單位值保持穩定。接著「氣體」會用於支付執行智慧商業契約 CyberMiles 節點的費用。智慧商業契約的「氣體」價格是以智慧商業契約提交至區塊鏈的時間由系統進行估算。

其次，CMT 可作為 CyberMiles 網路應用程式的內部結算貨幣。例如一家小企業的貸款申請（見第 5 節）可以使用 CMT 結算貸款及還貸，不必透過集中結算所，可以確保隱私性、透明度及資金安全；供應鏈管理應用程式可以用 CMT 結算期中交易，只能在一天結束時將餘額轉換為法定貨幣，這可以減少假定與交易成本。不管是哪一種情形，網路都會從各筆結算交易抽取一小筆費用，用以支付給執行交易相關智慧商業契約的驗證者。

CMT 的兩個用途都廣受區塊鏈技術社群接受。CMT 是不可或缺的，因為我們正在建設一個新的 CyberMiles 區塊鏈基礎設施，因此不能簡單地用 ETH 或 BTC 支付新區塊鏈的原有功能。我們可以比較 CMT 與目前流行的代幣。

<u>與 XRP 比較</u>

正如 Ripple 網，CyberMiles 網路也會以本地加密代幣促進交易的非集中式結算。

不過，每件交易中 Ripple 網都會「燒掉」一小筆 XRP，而相對地，CyberMiles 會收取交易費，支付給執行交易相關智慧商業契約的驗證者。此外，Ripple 網是以權限為基礎的區塊鏈，所有節點都是大型金融機構；相對地，CyberMiles 網路是為小企業服務的公共區塊鏈。

與 ETH 比較

正如 Ethereum 網，CyberMiles 網路既會獎勵創造新區塊的驗證者，也會獎勵在區塊內執行智慧契約的驗證者（透過「氣體」費）。

然而，目前這一代的 Ethereum 非常緩慢，因此運行複雜的智慧契約會非常昂貴。CyberMiles 在設計上追求高效能及高擴展性，足以運行複雜的智慧商業契約。此外，Ethereum 的目標是成為通用的計算網路，而 CyberMiles 智慧商業契約是專為優化電子商務交易而設。

4.3 啟動網路效應

為了使 CyberMiles 網路起飛，達成網路效應、為企業及礦工提供足夠參加價值至關重要。ICO 的目的之一是提供啟動網路的資源。我們的目標是透過 CyberMilesICO 達成下列成果。

首先，開發團隊會運用 5miles 在美國經營最大電子商務網站的經驗，建立智慧商業契約範本。5miles 有成千上萬的契約範本，可以分為 20 大類，全都經過現實應用程式的測試，而且很容易重新使用。此外，系統還會經營一家本身就屬於區塊鏈的「商店」，

以販售由第三方用戶開發的智慧商業契約範本。這些範本可以用 CMT 或「氣體」為定價單位。

其次，5miles 會建立一個新的應用程式，以支援其 1 千萬名美國用戶與小企業間的小企業貸款。這個應用程式會支援非集中式的個人身分及信用管理，以及非集中式貸款／還款結算（不必透過集中結算所）。這項工作可能會將 1 千萬名美國用戶的身分及信用紀錄移至 CyberMiles 區塊鏈。這個應用程式可以提供一個藍圖，讓其他開發人員能夠運用 CyberMiles 上的用戶身分及信用記錄，並建立自己的消費者端應用程式。以 CMT 作為結算貨幣，使 CyberMiles 網路可以提取一小部分費用，支付給執行貸款條款相關智慧契約的驗證者。越來越多應用程式建構於 CyberMiles 的同時，CMT 氣體的消費量也會增加。

最後，5miles 會將其旗艦 C2C（消費者對消費者）電子商務應用程式遷移至 CyberMiles 區塊鏈。這將是一個由 5miles 管理、透過 CyberMiles 上的智慧商業契約支援的 DApp，可能會使 30 億美元的交易量移動至 CyberMiles 平台，其結果是 5miles 自己就會購買及消費大量 CMT，以支付運行智慧商業契約的「氣體」費用。

5. 應用

CyberMiles 區塊鏈平台主要支援商業交易應用程式的中間軟體商業操作，因此主要是在用戶端應用程式的 UI 之下。不過基於非集中式區塊鏈的特點，其智慧商業契約可以實現集中式電子商務世界中絕對無法實現的潛在新功能及應用。

5.1 非集中式身分管理平台

正如 Equifax 遭駭事件（2017 年有超過 1 億名美國人的個人身分及信用記錄遭到竊取）所見的，集中式的個人身分管理對消費者而言風險極高，公司持有這類資料也必須背負很大的責任。為了解決這個問題，我們必須重新考慮身分管理的整體模式，顯而易見的解決方案是讓用戶完全掌控自己的個人資訊。用戶應該有權利逐案決定誰可以存取其資料。存取的時點、期間、資料使用的同意，都應該經由用戶核准。如此一來，根本就沒有個人資訊的中央儲存庫可供攻擊。但若不是以區塊鏈為基礎的智慧型商務契約，很難實現這樣的系統。

區塊鏈網路透過加密金鑰管理身分。用戶在 Bitcoin 或 Ethereum 區塊鏈的「電子錢包」是非集中式的，可以透過私人金鑰的用戶完全掌控。我們可以透過智慧商業契約擴展「電子錢包」的概念，保護的對象不只有加密代幣存款，也包含任何個人資訊。就像加密貨幣錢包，網路上也可以有很多「個資錢包」。根據用戶的請求（以用戶的私人金鑰簽署的交易），錢包可以授權第三方應用程式暫時透過 OAUTH 協定存取資料。一名用戶可以使用多個錢包，分別用於不同用途，就像現在的加密代幣錢包一樣。

下方的工作流程及圖 3 說明了個資「線上錢包」的運作方式。這種特殊的「錢包」可以儲存用戶的個人銀行資訊，因此用戶可以授權 CyberMiles 網路上的金融應用程式去使用它。第 5.2 節所述的端對端小企業貸款申請就是一個例子。

1. 用戶選擇自己信任的「錢包」應用程式。

2. 用戶在電子錢包註冊個人資訊及銀行資訊。

3. 錢包進行 AML／KYC 驗證，進行政府規定的反洗錢檢查。

4. 錢包產生公鑰／私鑰組合，公鑰會播送至區塊鏈備份。

5. 錢包授權並測試銀行連線。

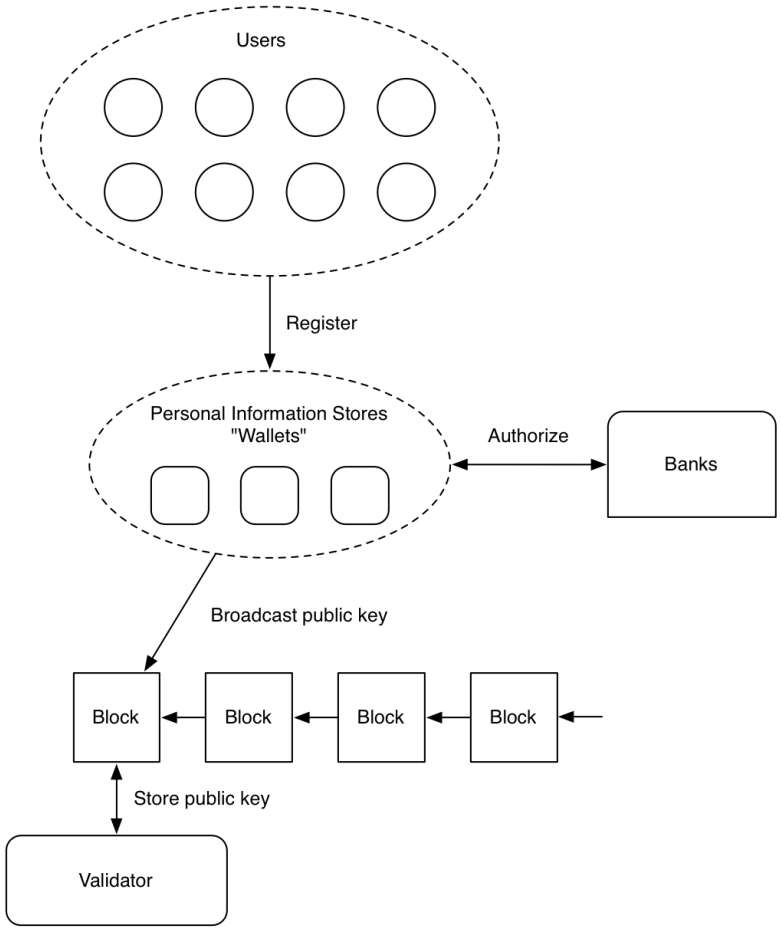

圖3. CyberMiles 的非集中式式身分管理平台

5.2 端對端小企業貸款市集

CyberMiles 區塊鏈另一個潛在的內建應用程式是端對端小企業貸款市集。如 5.1 節所述，我們會在 CyberMiles 建立一個非集中式的身分管理平台，然後區塊鏈可以透過公鑰儲存用戶的信用記錄。

有了身分及信用記錄，我們可以在區塊鏈建立貸款配對引擎（貸款「交換」）。只要貸款條件配對成功，智慧商業契約就會使用 CMT（透過自己的「個人資訊錢包」授權）從當事人的銀行帳戶直接自動結算貸款，不必經過集中結算所。以下的工作流程及圖 4 說明了如何配對及結算貸款。

1. 用戶從自己的錢包透過 OAUTH 登入交換所。交換所會快取但不會儲存個人資訊。
2. 用戶提出希望的貸款條件（借或貸、期限、利率）。
3. 交換所提出配對建議。
4. 交換所提供配對候選人的詳細的信用評分及歷史。
5. 如果用戶選擇一名候選人，雙方都必須表示同意。
6. 該貸款契約由交換所記錄，儲存於區塊鏈。
7. 交換所請求錢包透過雙方的銀行帳戶同時為雙方進行結算。

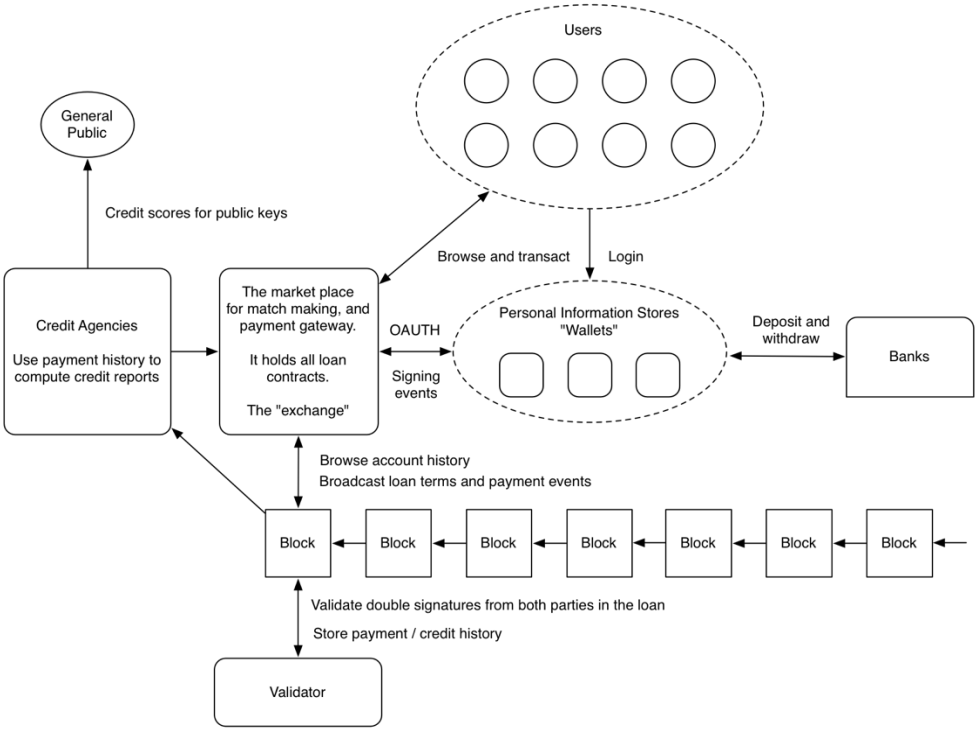

圖 4. 配對並結算非集中式的貸款。

在整個貸款期限，款項到期時，智慧商業契約會自動執行以下動作。

1. 交換所要求雙方當事人的錢包透過他們的銀行帳戶結算款項。

2. 交易結果播送至區塊鏈，並成為信用紀錄的一部分。

5.3 供應鏈現金流

CyberMiles 區塊鏈系統的虛擬代幣（CMT）主要用作存取網路的報酬（即企業付款以執行其智慧商業契約並驗證區塊鏈網路）。不過它也可以作為供應鏈中的各方當事人（包括終端消費者及銷售商）結帳的網路內媒體。

CMT 是數位代幣，所以結算上非常迅速、自由、安全，可實現高效率的供應鏈管理，因為「交易流量」可以在產品移動的同時發生。當事人只需要透過網路上的交換所定期將自己的 CMT 餘額轉換為其他資產即可。

50.4 認證產品

區塊鏈有一個重要特點，就是維持固定、安全的數位紀錄的能力。這個功能可以協助處理全球電子商務最為棘手的問題之一：仿冒品。

智慧商業契約可以設定為為產品製造商／生產商製造的每件產品建立真品認證（例如透過工廠的生產系統與 CyberMiles 智慧商業契約之間的 API 連線）。產品在供應鏈內從賣家移動至買家的過程中，可以透明地追蹤這項認證。

5.5 以社群為基礎的紛爭解決方案

集中式電子商務公司必須聘請客服人員來解決買家及賣家之間的糾紛。建立在 CyberMiles 區塊鏈 DApp 上的電子商務公司顯然也辦得到這件事。不過作為非集中式的平台，CyberMiles 要提供另一項引人注目的解決方案。

CyberMiles 社群用戶可以自願成為仲裁人，以 CMT 為報酬。由於交易的關鍵步驟都記錄在區塊鏈中（包括產品的真品認證及送貨單），智慧商業契約可以開發一種機制，讓仲裁人在買賣雙方的同意下找出這些紀錄。衝突解決之前，智慧商業契約可以代管買

賣雙方的 CMT 作為保證。仲裁人解決衝突後，如果雙方都接受，代管金就會釋出給「得勝」的當事人，而仲裁人也可以分配到一定的比例。

詞彙表

CyberMiles 區塊鏈：一個新的非集中式區塊鏈協議，為進行商業交易而經過優化。

智慧商業契約：可以在 CyberMiles 區塊鏈執行的商業應用程式。

CyberMiles 代幣（CMT）：加密貨幣／代幣，用於獎勵管理 CyberMiles 區塊鏈節點以維持區塊鏈、執行智慧商業契約之人。提出將在網路上執行的智慧商業契約的企業及當事人，必須視契約的複雜度支付 CMT。

CyberMiles 驗證者：貢獻計算能力，以維持 CyberMiles 區塊鏈基礎設施（包括執行智慧商業契約）的個人或實體。此人可能身在世界任何一個角落，也可能任職於 5miles，可接受獎勵（收取 CMT 作為維護網路的副產品）。

CyberMiles 應用程式：任何企業都可以在 CyberMiles 區塊鏈上建構及部署應用程式。該企業必須提出一套智慧商業契約，以便在網路上執行。企業必須購買 CMT 用以交換網路取得存取權限，並且可以使用 CMT 結算或促進內部金融交易。

終端用戶： 5miles 應用程式的買賣雙方不必知道 CyberMiles 的存在。智慧商業契約可在交易前後為他們轉換美元與 CMT。

5miles：由 5Miles LLC 開發的 C2C（消費者對消費者）電子商務市場應用程式。5miles 在美國擁有超過 1000 萬名客戶，每年的運行速度交易價值估計為 30 億美元。

謝詞

5xlab 誠致感謝 Michael Yuan 博士、Lucas Lu 博士對本文的貢獻。

參考資料

[1] Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf 2008.

[2] The Ethereum Team. A Next-Generation Smart Contract and Decentralized Application Platform. https://github.com/ethereum/wiki/wiki/White-Paper 2014

[3] Kwon, J. Tendermint: Consensus without Mining. https://tendermint.com/static/docs/tendermint.pdf 2014.

[4] Popov, S. IOTA: The tangle. https://iota.org/IOTA_Whitepaper.pdf 2016.

[5] Zamfir, V. Introducing Casper “the Friendly Ghost”. https://blog.ethereum.org/2015/08/01/introducing-casper-friendly-ghost/ 2015.

[6] Kwon, J and Buchman, E. Cosmos: A Network of Distributed Ledgers. https://cosmos.network/whitepaper 2016.

[7] Wood, G. Polkadot: Vision for a heterogeneous multi-chain framework. https://github.com/polkadot-io/polkadot-white-paper 2016.

[8] Poon, J and Dryja, T. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. https://lightning.network/lightning-network-paper.pdf 2016.

[9] Poon, J and Buterin, V. Plasma: Scalable Autonomous Smart Contracts. http://plasma.io/plasma.pdf 2017.

[10] Teutsch, J and Reitwiebner, C. A scalable verification solution for blockchains. http://bit.ly/2vIConl 2017.

[11] Forgy, C. Rete: A Fast Algorithm for the Many Pattern / Many Object Pattern Match Problem. Artificial Intelligence. 19: 17–37. 1982.

[12] Oracle. The Java Enterprise Edition Platform. https://www.oracle.com/java/technologies/java-ee.html

[13] Redhat. The Drools Business Rules Management System. http://drools.org/

[14] Sandia National Laboratories. Jess, the Rule Engine for the Java Platform. http://www.jessrules.com/

[15] Redhat. jBPM, a flexible Business Process Management Suite. http://www.jbpm.org/

[16] Lazo, D. OSWorkflow. http://shop.oreilly.com/product/9781847191526.do 2007

[17] DataStax. The Apache Cassandra database. http://cassandra.apache.org/

[18] The Ethereum Team. Swarm, serverless hosting incentivised peer-to-peer storage and content distribution. http://swarm-gateways.net/bzz:/theswarm.eth

[19] Benet, J. IPFS - Content Addressed, Versioned, P2P File System.

[20] Protocol Labs. Filecoin: A Decentralized Storage Network. http://filecoin.io/filecoin.pdf 2017.

[21] The Civic Team. Civic Whitepaper. https://tokensale.civic.com/CivicTokenSaleWhitePaper.pdf 2017

[22] Thomas, S. & Schwartz, E. A Protocol for Interledger Payments. https://interledger.org/interledger.pdf 2015