區塊鏈:
IOTA Qubic簡介

For Smart Contracts & DAPP

Reyer Chu (reyer.chu@apsjoin.com)

20181001

什麼是 Qubic?

- Qubic = QBC
  = Quorum-based computation
  = 基於法定人數 (仲裁) 的計算
- Qubic is a protocol: 制定 IOTA 基於仲裁計算的解決方案
  - Oracle機器
  - 外包計算
  - 智能合約
- 從長遠來看，Qubic將允許人們利用全球未使用的計算能力來滿足各種計算需求，同時幫助確保IOTA Tangle(基於IOTA的世界超級計算機)

Qubic: Oracle機器

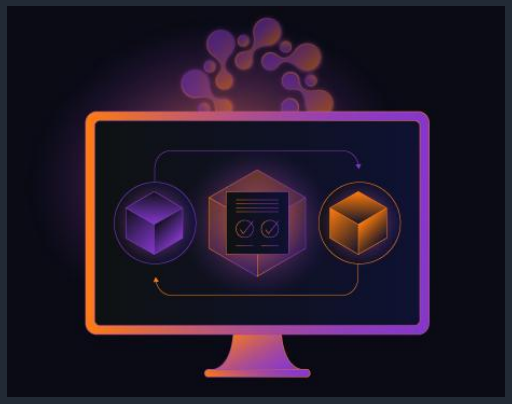
- 背景
  - [類似 Ethereum] Qubics通過oracle機器訪問外部數據，oracle機器充當了qubic和外部世界之間的鏡頭
- 問題
  - Oracle提供不正確資料(Oracle本身不屬於協議範圍)
- Qubic可通過法定人數 (仲裁) 確認有問題的數據

Qubic: 外包計算

- 問題
  - 對於任何計算設備，總存在對於設備來說計算資源過於密集的任務，或需超出本地可用數據的任務。對於IoT設備尤其如此
- Qubic可實現外包計算，為消費者和生產者提供安全的參與。低功耗設備可簡單地將密集型計算外包給外部功能更強的機器

Qubic: 智能合約

- 背景
  - [類似 Ethereum] 智能合約通過將合同義務封裝在軟體中來免除對第三方執行的需要，以便自動驗證和執行，有權訪問合同的任何人都可驗證特定事件是否始終會產生特定結果
- 免費交易 + Qubic (基於通用仲裁的計算)
  →打開了全新的可能性
  - 例如，智能合約可用於將來自不同oracle的溫度數據匯總到平均溫度，該溫度定期發佈到Tangle → 智能合約現在已成為一個oracle (合同本身已成為外部數據的來源，可供oracle機器接收並發送回其他部分)



例: 外匯交易 = Oracle + 外包計算 + 智能合約

- **匯率Qubic**
  - 作為一個原始的oracle機器，定期發布Tangle的匯率
  - 數據發起者可以是預定義的或不是預定義的

  → Oracles從Tangle環境之外獲取數據，這是通過其他方式無法獲得的
- **外匯預測 Qubic**
  - 獲取**匯率Qubic**提供的數據並預測近期的利率

  → Oracles執行密集的外包計算，這對於低級設備來說太難或太昂貴
- **投資組合管理Qubic**
  - 獲取上述兩個提供的數據並出售或購買掛鉤的虛擬美元用於掛鉤的虛擬歐元

  → Oracles執行智能合約，允許所有者不必手動處理所有交易

Qubic: 基於法定人數 (仲裁) 的計算 (1/2)

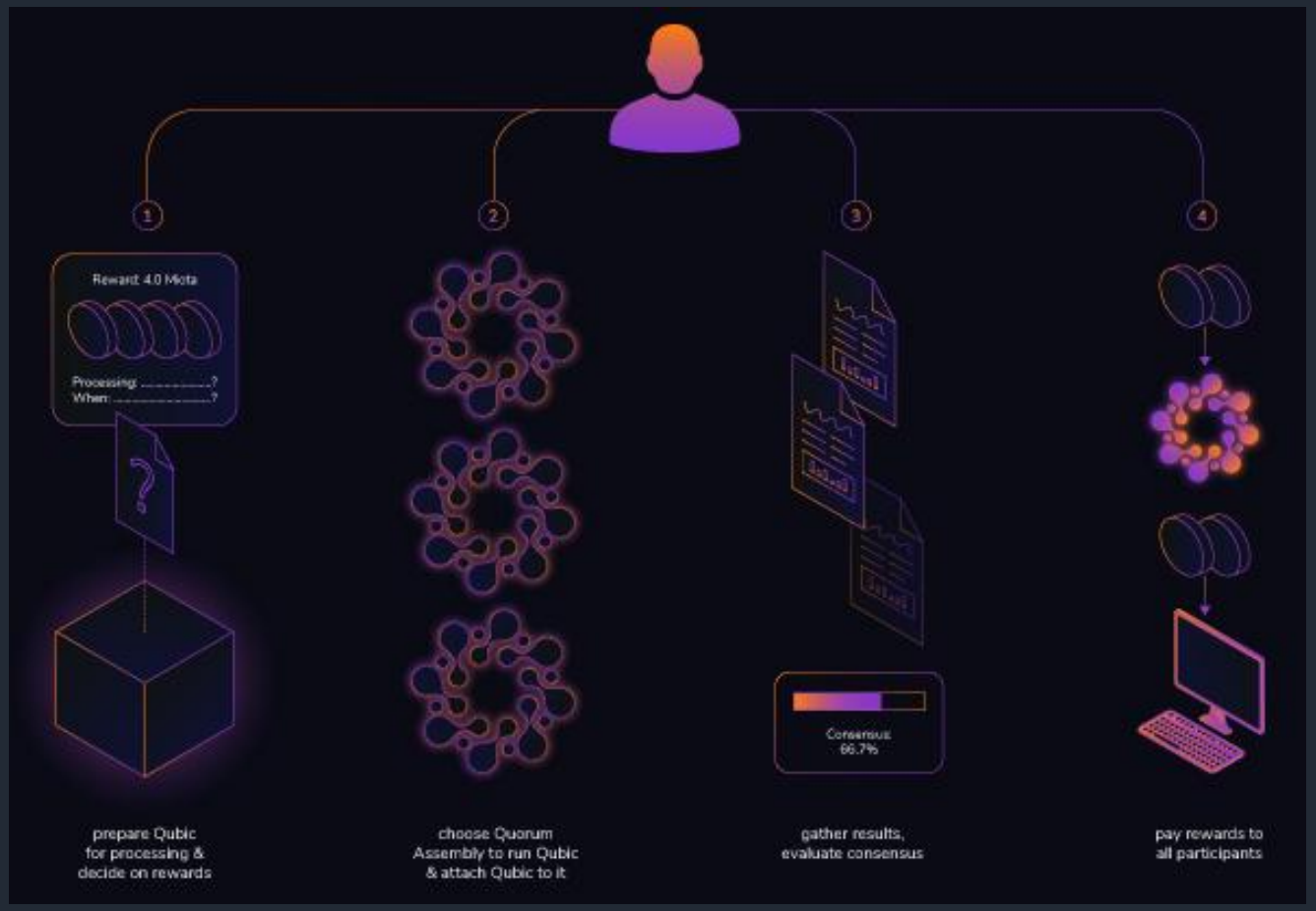
- 問題: Oracle提供不正確資料
- Qubic可通過法定人數 (仲裁) 確認有問題的數據
  - **強制達到法定人數共識**
  - 接受 or 拒絕
    - 在Qubic系統中，一組oracles將組合成一個程序集，所有成員將處理同一組Qubics，且每個oracle將在Tangle上發布每個Qubic的處理結果
    - 這將允許Qubic所有者確定大會的法定人數共識。如果他們不能形成法定人數（集會中至少有2/3的oracles對結果達成一致）那麼結果將不會被Qubic所有者接受。
  - 激勵機制
    - 由Qubic的所有者提供獎勵，將分發給集會中的oracles，這些oracles產生了該Qubic的法定人數結果
    - 如果你想出一個篡改或錯誤的反對結果，或決定不參與處理Qubic ，你將錯過獎勵
  - Qubic所有者可通過選擇更大的程序集來增加對處理結果的信心，或者甚至可以通過多個隨機程序集處理Qubic。這一切都取決於數據的重要性和他願意花的獎勵金額

Qubic: 基於法定人數 (仲裁) 的計算 (2/2)

- 這類似於立法機構
  - 法定人數: 開展該業務所必需通過的審議大會（如立法機構）的最低人數
- 就Qubic而言
  - 審議大會: 一組具體的oracles
  - 法定人數: 該小組投票權的最小百分比，須達共識才能使結果被認為是有效的
- 雖然讓多個設備處理相同的計算似乎很浪費，但這是保持數據未在無信任的分散系統中被篡改的唯一方法
- Qubic只希望特定程序集中的oracles參與給定的計算
  - Qubic甚至允許單oracle程序集 ➔ 可在運營商受信任的環境中使用相同的協議，讓結果的有效性易於驗證
  - 不同於 Ethereum: 期望每個網絡參與者都參與每個計算

Qubic 的生命週期

1. 準備一個Qubic進行處理
   - 決定所提供的獎勵
   - 決定運行Qubic的程序集(assembly)
2. 程序集開始處理Qubic
3. 收集提交的處理結果 & 評估是否達法定人數共識
4. 收集顯示的處理結果
   - 當結果到達時，將承諾的獎勵支付給參與者



Qubic 使用的程式語言: Abra

- Abra is *an intermediate trinary-based functional programming language*.
- *Intermediate*
  - 為了能在不同硬件平台上運行相同的代碼，Qubics以這種中間語言打包，這實質上意味著該語言可以輕鬆地轉換或解釋特定的硬件。這使得Qubic在很大程度上與硬件無關
- *Trinary-based*
  - 三元系統可以提供顯著的節能，這是物聯網設備的關鍵考慮因素。一個三位數，一個trit，可以代表1.58位。因此，三元系統所需的佈線量可以減少到等效二元系統的約64%，從而導致相應的能量減少
- *Functional*
  - 一個函數編程語言可以更容易的分析來證明代碼的正確性。
  - 可自主分析 (for formal verification)
  - 可平行化