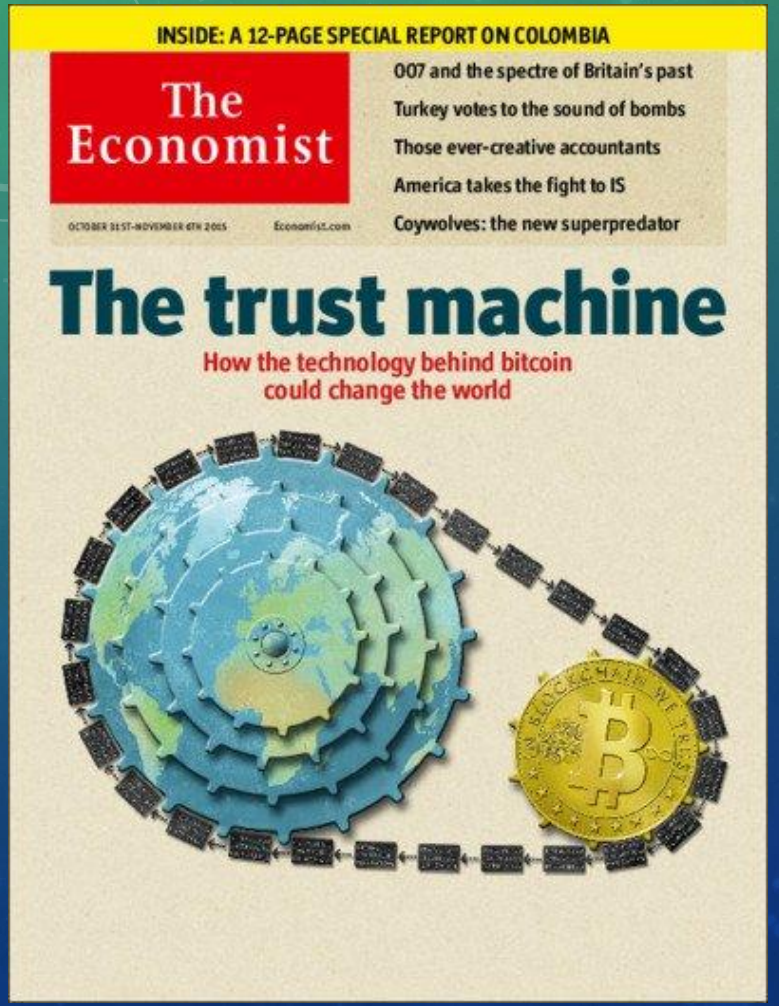
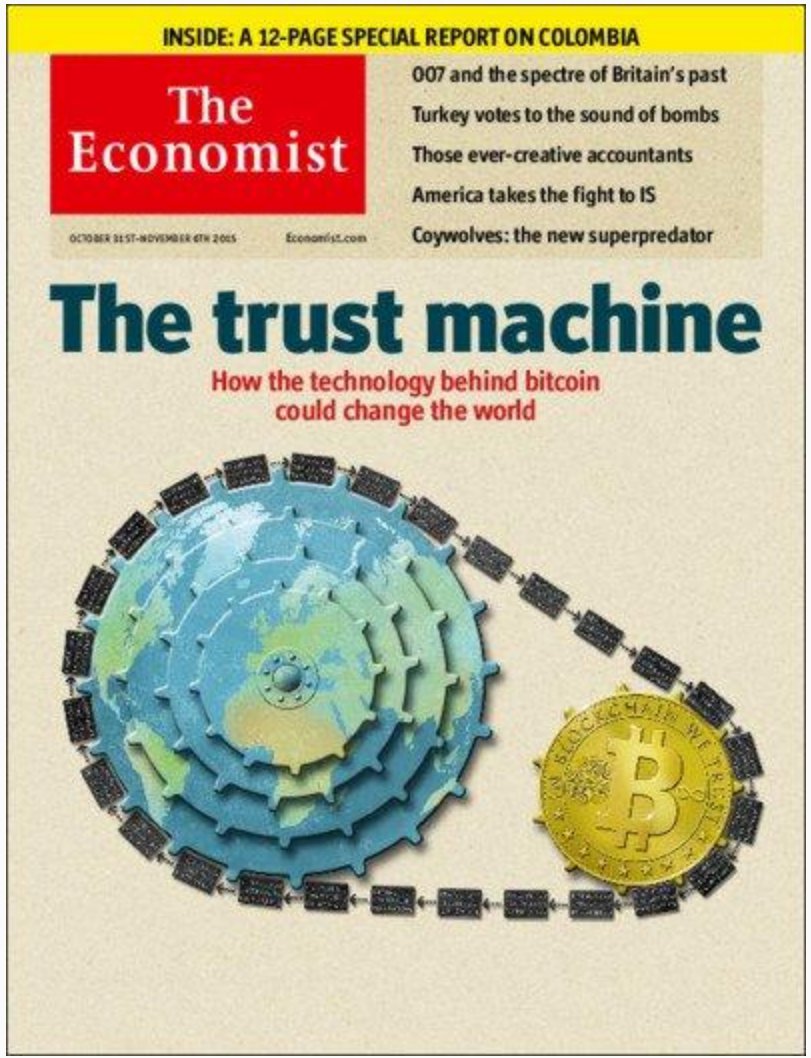
區塊鏈之時代意涵 應用與未來

Professor Liao, 11/7/2016



廖世偉博士

- 廖世偉博士的志業是通過創新，產品，Open Source 來教育服務，改變世界。
- 廖博士 (PhD Stanford) 在斯坦福大學、英特爾、谷歌公司工作了22年，獲得谷歌內部頒發的最高榮譽獎：創始人獎（Founder's 獎）。
- 2013年廖博士從谷歌退休回到斯坦福大學教授程序分析和優化，也在台灣大學教授 最新的安卓系統和金融科技大數據課程。
- 天龍八部（such as 臺大幫幫忙) on Gcoin 區塊鏈。
- 台大黑客松：在天龍八部之前的創新機制，與時俱進。



經濟學人:

狂熱是永恆的源泉 -- 未來將屬於分散式帳本技術, 但普及為時尚早, 在找到應用前, 過高的期望往往會帶來失望

區塊鏈本質:技術:

“證明你不是中心” is key.

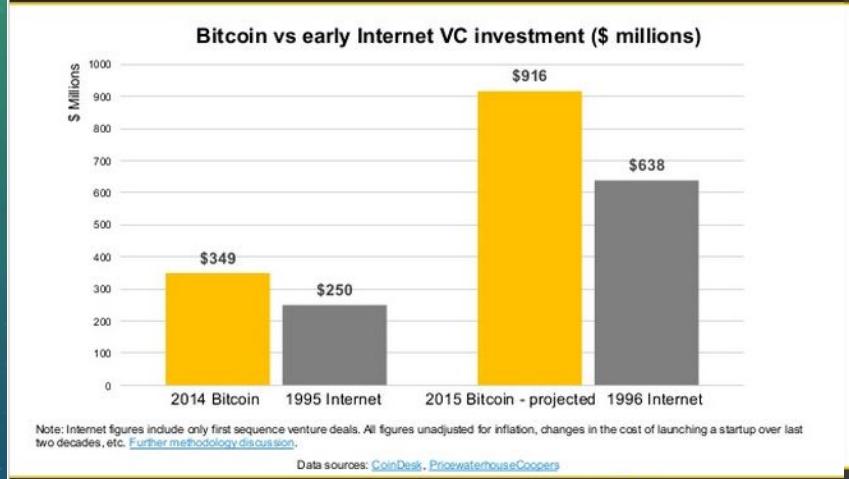
- “你對我不重要，但沒有你，對我很重要” -- 讓子彈飛
- 今天不是在拼誰是中心
  - 所以 Gcoin (bitcoin 3.0) 做到嫁接 both bitcoin and Ethereum 智能合約 on bitcoin 區塊鏈
- 今天若光說 “多中心”，還是本位主義，將不是區塊鏈
  - 不是不技術，就急著綁一家 ...

國際情勢

先4張投影片:4盲人摸象
先感覺一下

風險投資蜂擁至數字金融科技 (區塊鏈) 產業

Bitcoin VC Investment Projected to Continue Outpacing Early-Stage Internet Investment



Note: Internet figures include only first sequence venture deals. All figures unadjusted for inflation, changes in the cost of launching a startup over last two decades, etc. Further methodology discussion.

Data sources: CoinDesk, PricewaterhouseCoopers

State of Bitcoin Q1 2015 CoinDesk Sponsored by Gem 30

All-Time Bitcoin Venture Capital Investment Increased 51% from End of 2014 to $676m

Q1 2015 bitcoin VC investment:

$229m

Total VC investment in cryptocurrency startups to date:

$676m

Q4 2014 bitcoin VC investment:

$133m

Increase in total to-date VC investment from end of Q4:

+51%

Data sources: CoinDesk (www.coindesk.com/bitcoin-venture-capital/), CrunchBase

State of Bitcoin Q1 2015 CoinDesk Sponsored by Gem 29

source: http://www.coindesk.com/

各大銀行

投資

區塊鏈技術

www.ft.com/intl/cms/s/0/f358ed6c-5ae0-11e5-9846-de406ccb37f2.html#axzz3nrDBpf2m

SwapClear Compression
Delivering Record Efficiencies to You
FIND OUT MORE →
LCH.CLEARNET

September 15, 2015 12:42 pm

Blockchain initiative backed by 9 large investment banks

Philip Stafford

LATEST FROM fastFT

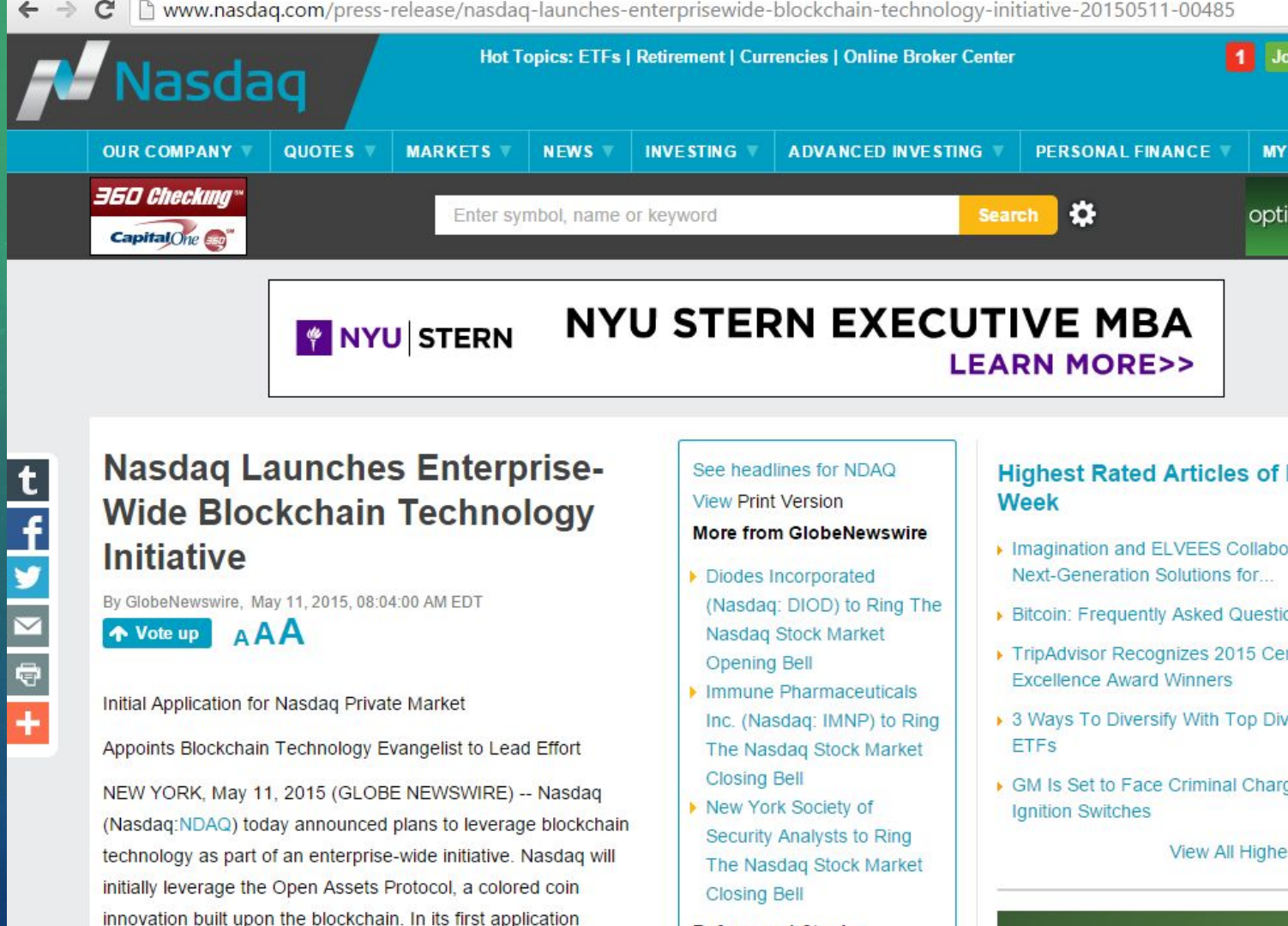
Will gold regain some lustre?
Macau casinos up 20% since Sept 30 nadir
Fast Europe Open: Tesco and Tories
Asia's top performing index this month: Indonesia
Mini oil bounce dents airline shares

SwapClear Compression
Delivering Record Efficiencies to You
FIND OUT MORE →
LCH.CLEARNET

Nasdaq

也使用

區塊鏈技術



新加坡总理呼吁区块链技术

新加坡总理李显龙呼吁银行和监管机构关注区块链技术

2015-11-14 11:25:42 浏览量：7700 关键词：新加坡 区块链 总理

新加坡总理呼吁该国银行和监管机构要时刻关注最新科技的发展，比如区块链技术。

昨天在新加坡举行的联合海外银行八十周年晚宴上，李显龙总理指出目前金融行业正面临着各种挑战，强调金融行业应该与技术发展保持同步，以保证自身的竞争力，不被市场淘汰。

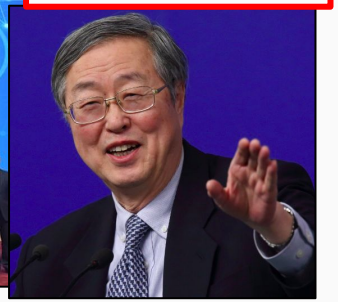
source:
http://www.btc38.com/btc/altgeneral/8681.html

2016國際情勢：合縱連橫

全球約50家銀行組成R3聯盟
共同研發區塊鏈技術標準

中國政府牽頭成立：
阿爾山
中國區塊鏈研究聯盟
中關村區塊鏈產業聯盟

多國央行已展開
區塊鏈技術研究



Nasdaq

London Stock Exchange

TOKYO STOCK EXCHANGE

多國證交所已展開區塊鏈交易之測試與研究


E BOOSTER
BLOCKCHAIN MAKES POSSIBLE: INVESTMENT, DEVELOPMENT & OPERATIONS.
BUILDING A DIGITAL SILK ROAD FOR WIN-WIN COOPERATION
Information Infrastructure Partnership
数字丝路·合作共赢论坛
信息基础设施共建
WORLD INTERNET CONFERENCE
5 INFRASTRUCTURE BOOST

國際技術情勢

先4張投影片: recent facts

640 Currencies / 71 Assets / 2386 Markets Market Cap: $14,105,385,240 / 24h Vol: $119,162,111 / BTC Dominance: 84.2%

Crypto-Currency Market Capitalizations

Market Cap ▾ | Trade Volume ▾ | Trending ▾ | Tools ▾ | Search Currencies

All ▾ | Currencies ▾ | Assets ▾ | USD ▾ | Next 100 → | View All

#	Name	Market Cap	Price	Available Supply	Volume (24h)	% Change (24h)	Price Graph (7d)
1	Bitcoin	$11,847,218,803	$742.52	15,955,356 BTC	$85,868,900	2.18%	
2	Ethereum	$925,528,397	$10.81	85,584,546 ETH	$11,794,000	-1.30%	
3	Ripple	$292,514,964	$0.008233	35,531,082,209 XRP *	$1,916,390	1.63%	
4	Litecoin	$200,186,471	$4.15	48,267,129 LTC	$3,106,240	0.82%	
5	Ethereum Classic	$75,433,994	$0.882403	85,487,010 ETC	$690,392	0.07%	

coinmarketcap.com

SWIFT report on Blockchain

Trust Machine landscape these 10 days

- Fabric: 0.6 preview
- Ethereum yesterday: 0.4.4
  https://blog.ethereum.org/2016/11/01/security-alert-solidity-variables-can-overwritten-storage/
- Ethereum last week: Another fork
- http://www.prweb.com/releases/2016/09/prweb13702075.htm?from=groupmessage&isappinstalled=0

WINGS White Paper released: a cutting edge approach for selecting, backing and managing decentralized autonomous organizations (DAO) on Bitcoin and other blockchains

Today the WINGS team is releasing a white paper, "WINGS: A project backing social platform with incentivized forecasting," describing an user-experience focused system designed specifically for the curation, forecasting, backing, and governance of smart contracts controlled DAOs running on Bitcoin and other smart contracts capable blockchains.

TEL AVIV, ISRAEL (PRWEB) SEPTEMBER 26, 2016

Today's entrepreneurs live in an interconnected global economy yet face many hurdles in establishing a truly global footprint for the next big thing. The challenges of bringing a great idea to reality range from adapting to language and cultural differences, establishing trust, enforcing contracts, navigating between varying regulatory regimes, having an adequate quality workforce, and accessing capital assets globally.

Bitcoin provides a novel way of conducting financial transactions in a censorship-free,

議程

- 當今情勢
- 區塊鏈 (Blockchain): 數位經濟基礎建設
- 區塊鏈: Trust Machine = 4D → EFG
- 區塊鏈的本質: Internet of Value as a Trust Machine
- 區塊鏈的信任安全
- Case Study: Blockchain vs. O2O核銷支付的挑戰
    - Blockchain 的解決方案: bitcoin, loyyal.com, blockpoint.io, Gcoin
- 數位經濟美麗新世界的展示

區塊鏈: 數位經濟的基礎建設之一

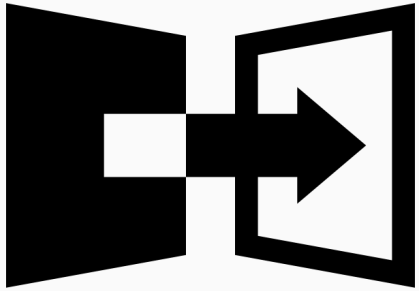
數位經濟這個美麗新世界需要信任機制！

- 世界走向自動化，智能化，anytime on-line 的數位經濟服務網路
- 當美麗新世界到來，每個人的數位經濟供應鏈上將有許多時時刻刻在競爭的服務提供商
  - 銀行將只是數位經濟供應鏈上的一個節點，與其他 FinTech 公司甚至個人激烈競爭
- All-time，all-place, all-provider 的數位經濟服務網路勢必需要一個信任機制：區塊鏈
  - 區塊鏈：數位經濟服務網路的基礎建設之一

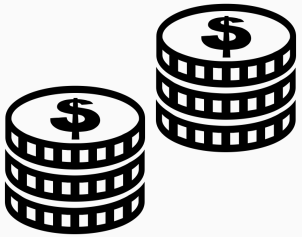
信任機制 (Trust Machine) 活絡經濟,相加相乘打群架

區塊鏈:從4D到EFG美麗新世界

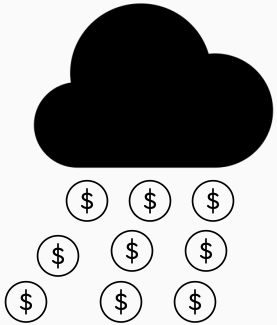
數位經濟的信任問題:4D



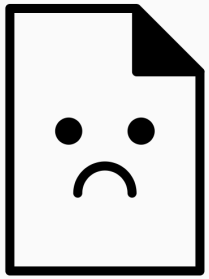
Duplication
偽造



Double
Spending
重複花費



Dilution
濫發



Data Lost
資料庫問題

Digital Finance之前發展不起來的原因:
4D → Blockchain 一次解決！

Blockchain 如何解決 4D 問題

- Duplication:
  - Protected by the Proof-of-work mechanism
  - Every client has a copy of the list of transactions
- Double Spending:
  - The transaction needs to be confirmed by the nodes in the Blockchain network.
  - For example, Bitcoin protects against double spending by verifying each transaction added to the Blockchain to ensure that the inputs for the transaction had not previously been spent
- Dilution:
  - The amount of coin can be limited by proper design via protocol and algorithm.
  - Decentralized system: no central authority controls the right to issuing currency
- Data lost:
  - All transactions are trackable and are recorded in the Blockchain

Trust Machine的發力點：EFG

Efficiency
- 數位化交易
- 快速結算
- 低成本交易

Finality
- 降低違約風險

Inspectability (or Gongping, Gongzheng, Gongkai)
- 公平公正公開
- 透過 API 串接
- 開放 API 符合矽谷精神

Blockchain 技術的優勢：EFG

E

增加交割效率

F

加強安全性

G

可信任性

區塊鏈降低互聯網金融的交易成本，促進資產流動性

HOW DO BITCOINS WORK?

'Miners' create Bitcoins by using computers to solve mathematical functions. The same process also verifies previous transactions

Bitcoin exchanges will trade between conventional currencies and Bitcoin, offering a way into the market for non-miners, as well as a way to cash out

Users download a Bitcoin 'wallet' that works a little like an email address, providing a way to store and receive currency. Bitcoins can be transferred from one wallet to another using a web browser or a phone app

Businesses create a wallet in the same way as an individual user, typically using a website button to enable a Bitcoin payment. For in-the-flesh enterprises, QR codes can be used to let customers pay quickly and easily

What kind of Trust Machine is it?

去中心化	去信任化	安全可靠	不可逆性
共同參與	公開透明	匿名性	無法篡改

信任機制

融合核心技術支持

1. 隱私安全: CoinJoin, CoinShuffle, Zero-knowledge proof, Coin Mix, Mix server, 51% attack
2. 數學: One way function
3. 密碼學: 單向散列演算法/公鑰加密/ECC加密算法/SHA-256演算法
4. 經濟模型: 貨幣的發行設計機制/賽局理論
5. 算法算力: Byzantine Generals problem/解難題/Hashcash/Markov Chain
6. Scalability (Global)
7. Flexibility (Governance structure, smart contract)

Blockchain


Ver.
不可竄改性
交易即清算
安全加密機制
可追蹤性
Immutable
Transaction is Settlement
Security
Traceable

議程

- 當今情勢
- 區塊鏈 (Blockchain): 數位經濟基礎建設
- 區塊鏈: Trust Machine = 4D → EFG
- 區塊鏈的本質: Internet of Value as a Trust Machine
- 區塊鏈的信任安全
- Case Study: Blockchain 如何因應O2O核銷支付的挑戰
    - Blockchain 的解決方案: bitcoin, loyyal.com, blockpoint.io, Gcoin
- 數位經濟美麗新世界的展示

區塊鏈的本質:
Internet of value as a Trust Machine

Blockchain is Internet-of-Value

- 互聯網的本質:
  - Open
  - 制高點
  - Create Value
  - Bottom-up
    - Developer-driven
  - Disruptive

2 Security Paradigms

- Security-through-Obscurity
- Security-through-Internet

2 System Design Layers

- Policy
- Mechanism : Security-through-Internet

我教大二的課 : 超過10億台幣的 lesson :

超過10億台幣被DAO (盜), 並2016/7/20 hard forked blockchain

- Important event: Time will tell eventually
  - What about immutability and other questions that come from this forking?
  - The original chain (Ethereum Classic) vs. The mutated chain?
- Ethereum has created a lot of extra work for exchanges in order to ‘accommodate’ their DAO losses...

DAO

DAO (Decentralized Autonomous Organization) 是一個群眾募資的智能合約

他運行於以太坊 (Ethereum) 的區塊鍊上

ref: https://daohub.org

https://en.wikipedia.org/wiki/The_DAO_(organization)

DAO

在以太坊上的使用者如果持有 DAO 幣，即可參與是否支持 DAO 上的募資計劃

當 DAO 上的某個計畫達到 51% 的 DAO 幣支持，該募資計畫即成功

募資計畫會將所得利潤回饋給投資者

DAO

DAO 的智能合約為了防範 51% 攻擊 (有攻擊者能夠控制一半以上的 DAO 幣)

或是有投資者在經過至少一週的討論以後，依然不想投資該募資計劃

則該投資者可以選擇退出該投資計畫

即是 DAO 對 “多數暴政” 的防範機制

投資者可於任何時刻退出，並獲得該時刻應得的投資利潤

DAO

該防範機制是不同意的投資者可以發起 “splitDAO”

將自己的 “DAO幣” 轉換為另外一個 “新的智能合約”，即是 “childDAO”

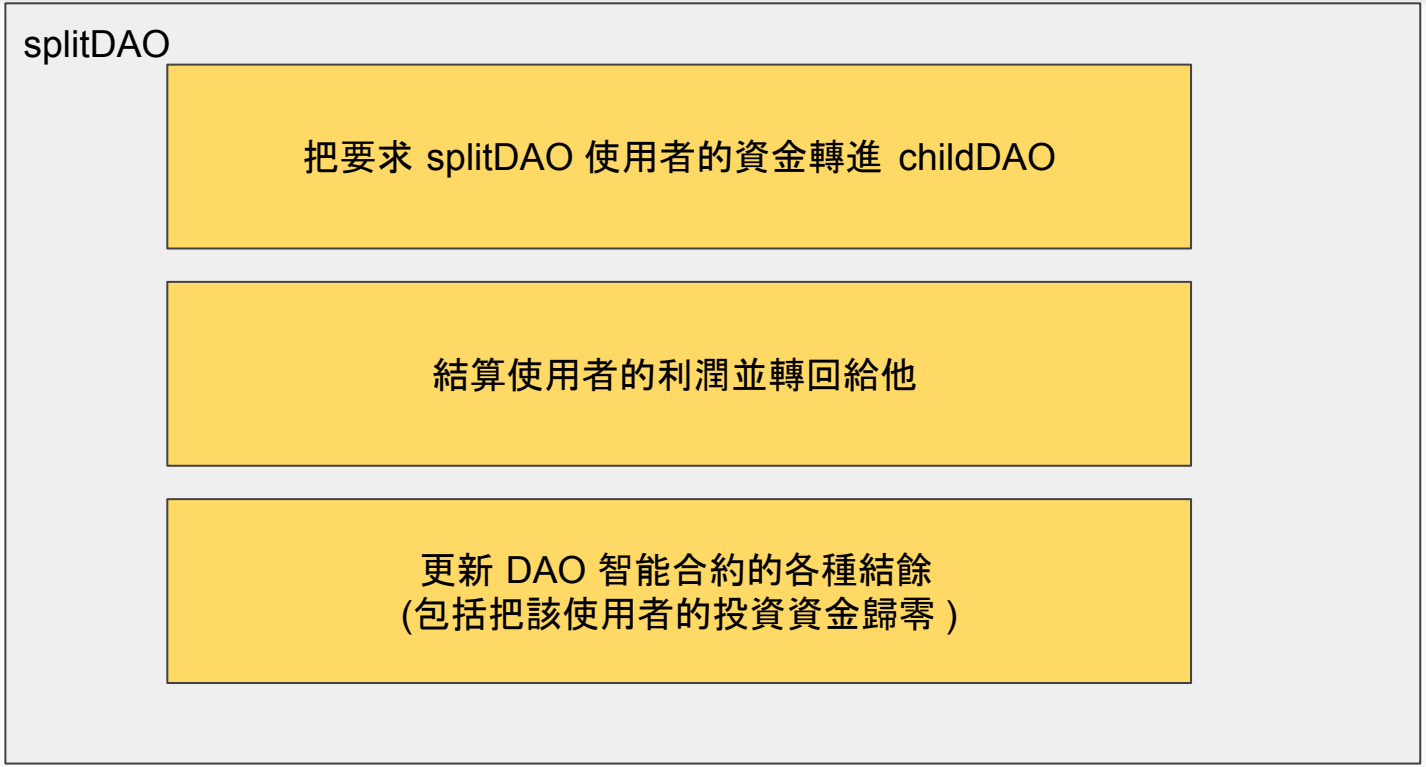
childDAO 並不屬於原來的 DAO幣，因此不會參與那個募資計劃

Race to empty 攻擊

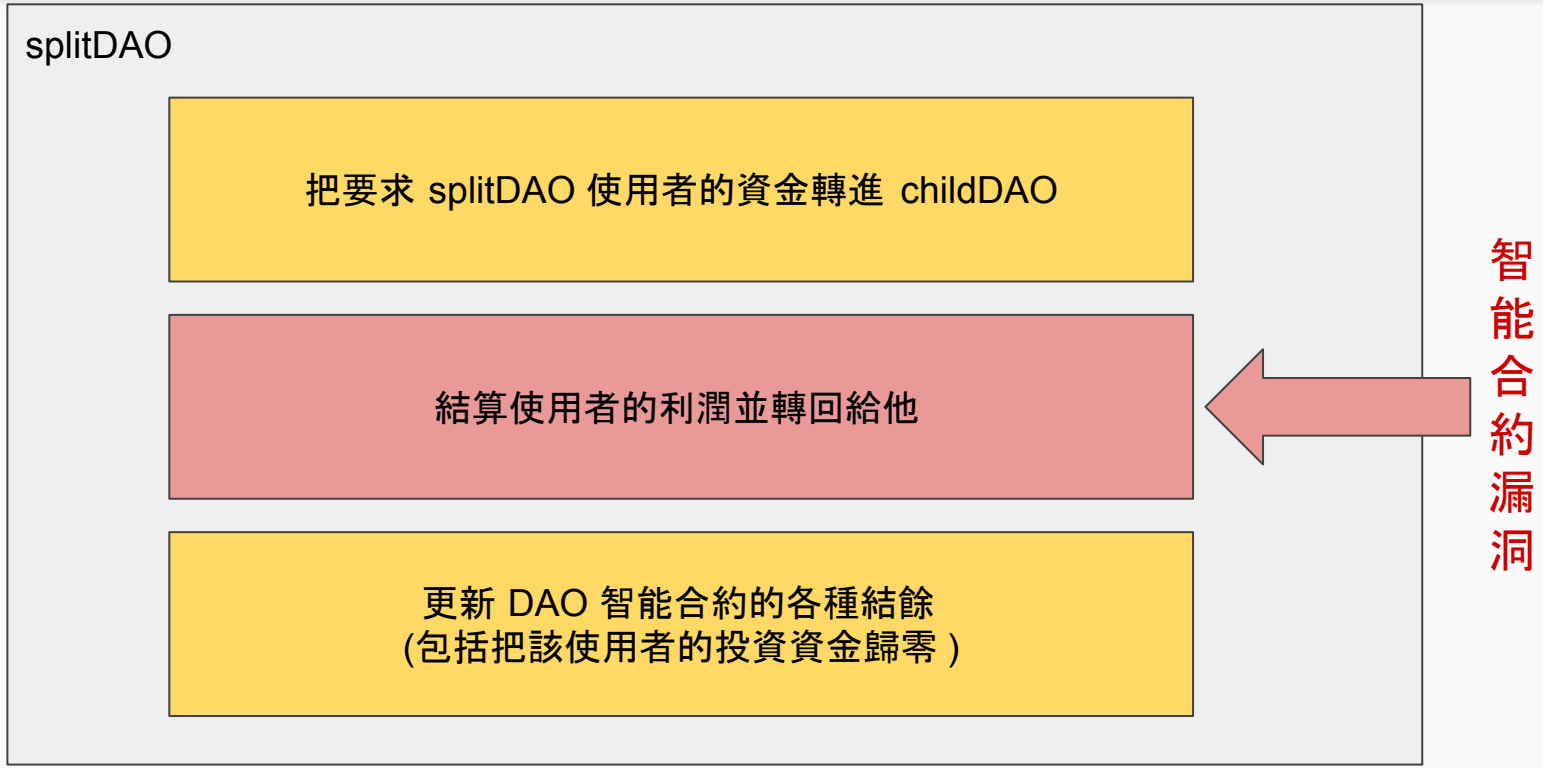
但 DAO 的智能合約在設計上有漏洞，導致攻擊者可以發動一種攻擊

“Race to Empty”

Race to empty 攻擊



Race to empty 攻擊

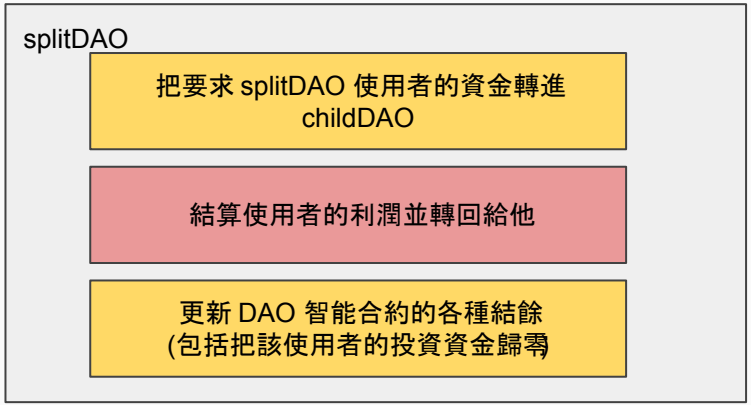


Race to empty 攻擊

攻擊者利用這個漏洞，在程式運作到

紅色區塊的時候，讓程式暫停在這裡，並且

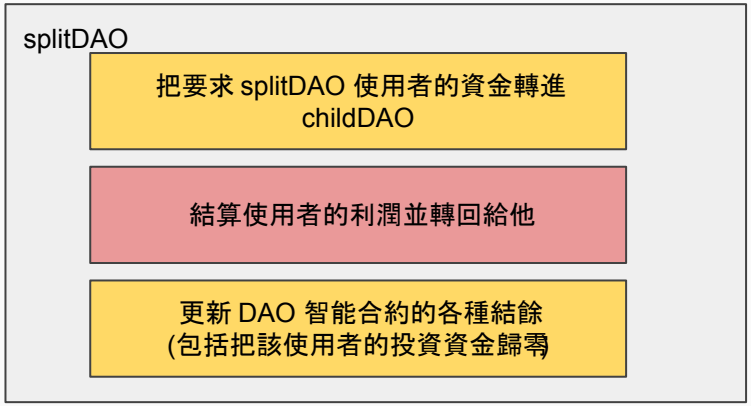
再次呼叫了 splitDAO



Race to empty 攻擊

於是攻擊者就可以一直要求智能合約募資計畫

反覆的退相同數目的 DAO幣到自己的 childDAO



Race to empty 攻擊

打個比方

Alice 眼睛不好

Alice 跟小明借了 100 元，今天小明來跟 Alice討債

Alice 剛從錢包抓出 100 元的時候，小明就把它偷走，並且跟 Alice 說：

“欸你剛剛沒抓到錢，你還沒還我 100 元” (splitDAO還沒運行到清算)

小明重複這個步驟很多次 (重複呼叫 splitDAO)

Race to empty 程式說明

以下將對程式碼進行說明

ref: http://hackingdistributed.com/2016/06/18/analysis-of-the-dao-exploit/

http://vessenes.com/deconstructing-thedao-attack-a-brief-code-tour/

http://ethfans.org/posts/115 (上一篇的簡體翻譯)

http://ethfans.org/topics/419

http://ethfans.org/posts/116

splitDAO

```
function splitDAO(
        uint _proposalID,
        address _newCurator
    ) noEther onlyTokenholders returns (bool _success) {

        ...

        // Move ether and assign new Tokens
        uint fundsToBeMoved = (balances[msg.sender] * p.splitData[0].splitBalance) / p.splitData[0].totalSupply;
        if (p.splitData[0].newDAO.createTokenProxy.value(fundsToBeMoved)(msg.sender) == false)
            throw;

        ...

        // Burn DAO Tokens
        Transfer(msg.sender, 0, balances[msg.sender]);
        withdrawRewardFor(msg.sender); // be nice, and get his rewards
        totalSupply -= balances[msg.sender];
        balances[msg.sender] = 0;
        paidOut[msg.sender] = 0;
        return true;
    }
```

splitDAO

```
function splitDAO(
        uint _proposalID,
        address _newCurator
    ) noEther onlyTokenholders returns (bool _success) {

        ...

        // Move ether and assign new Tokens
        uint fundsToBeMoved = (balances[msg.sender] * p.splitData[0].splitBalance) / p.splitData[0].totalSupply;
        if (p.splitData[0].newDAO.createTokenProxy.value(fundsToBeMoved)(msg.sender) == false)
            throw;

        ...

        // Burn DAO Tokens
        Transfer(msg.sender, 0, balances[msg.sender]);
        withdrawRewardFor(msg.sender); // be nice, and get his rewards
        totalSupply -= balances[msg.sender];
        balances[msg.sender] = 0;
        paidOut[msg.sender] = 0;
        return true;
    }
```

splitDAO

```
function splitDAO(
        uint _proposalID,
        address _newCurator
    ) noEther onlyTokenholders returns (bool _success) {

        ...

        // Move ether and assign new Tokens
        uint fundsToBeMoved = (balances[msg.sender] * p.splitData[0].splitBalance) / p.splitData[0].totalSupply;
        if (p.splitData[0].newDAO.createTokenProxy.value(fundsToBeMoved)(msg.sender) == false)
            throw;

        ...

        // Burn DAO Tokens
        Transfer(msg.sender, 0, balances[msg.sender]);
        withdrawRewardFor(msg.sender); // be nice, and get his rewards
        totalSupply -= balances[msg.sender];
        balances[msg.sender] = 0;
        paidOut[msg.sender] = 0;
        return true;
    }
```

把投資金額轉成 childDAO

splitDAO

```
function splitDAO(
        uint _proposalID,
        address _newCurator
    ) noEther onlyTokenholders returns (bool _success) {

        ...

        // Move ether and assign new Tokens
        uint fundsToBeMoved = (balances[msg.sender] * p.splitData[0].splitBalance) / p.splitData[0].totalSupply;
        if (p.splitData[0].newDAO.createTokenProxy.value(fundsToBeMoved)(msg.sender) == false)
            throw;

        ...

        // Burn DAO Tokens
        Transfer(msg.sender, 0, balances[msg.sender]);
        withdrawRewardFor(msg.sender); // be nice, and get his rewards
        totalSupply -= balances[msg.sender];
        balances[msg.sender] = 0;
        paidOut[msg.sender] = 0;
        return true;
    }
```

把投資金額轉成 childDAO

歸還利潤

splitDAO

```
function splitDAO(
    uint _proposalID,
    address _newCurator
) noEther onlyTokenholders returns (bool _success) {

    ...

    // Move ether and assign new Tokens
    uint fundsToBeMoved = (balances[msg.sender] * p.splitData[0].splitBalance) / p.splitData[0].totalSupply;
    if (p.splitData[0].newDAO.createTokenProxy.value(fundsToBeMoved)(msg.sender) == false)
        throw;

    ...

    // Burn DAO Tokens
    Transfer(msg.sender, 0, balances[msg.sender]);
    withdrawRewardFor(msg.sender); // be nice, and get his rewards
    totalSupply -= balances[msg.sender];
    balances[msg.sender] = 0;
    paidOut[msg.sender] = 0;
    return true;
}
```

把投資金額轉成 childDAO

歸還利潤

清算

有問題的部分：歸還利潤

```
function withdrawRewardFor(address _account) noEther internal returns (bool _success) {
    if ((balanceOf(_account) * rewardAccount.accumulatedInput()) / totalSupply < paidOut[_account])
        throw;

    uint reward = (balanceOf(_account) * rewardAccount.accumulatedInput()) / totalSupply - paidOut[_account];

    reward = rewardAccount.balance < reward ? rewardAccount.balance : reward;

    if (!rewardAccount.payOut(_account, reward))
        throw;
    paidOut[_account] += reward;
    return true;
}
```

有問題的部分：歸還利潤

```
function withdrawRewardFor(address _account) noEther internal returns (bool _success) {
    if ((balanceOf(_account) * rewardAccount.accumulatedInput()) / totalSupply < paidOut[_account])
        throw;

    uint reward = (balanceOf(_account) * rewardAccount.accumulatedInput()) / totalSupply - paidOut[_account];

    reward = rewardAccount.balance < reward ? rewardAccount.balance : reward;

    if (!rewardAccount.payOut(_account, reward))
        throw;
    paidOut[_account] += reward;
    return true;
}
```

有問題的部分：歸還利潤

```
function withdrawRewardFor(address _account) noEther internal returns (bool _success) {
    if ((balanceOf(_account) * rewardAccount.accumulatedInput()) / totalSupply < paidOut[_account])
        throw;

    uint reward = (balanceOf(_account) * rewardAccount.accumulatedInput()) / totalSupply - paidOut[_account];

    reward = rewa                    e < reward ? rewardAccount.balance : reward;

    if (!rewardAccount.payOut(_account, reward))
        throw;
    paidOut[_account] += reward;
    return true;
}
```

出問題的地方

payOut

```
function payOut(address _recipient, uint _amount) returns (bool) {
    if (msg.sender != owner || msg.value > 0 || (payOwnerOnly && _recipient != owner))
        throw;
    if (_recipient.call.value(_amount)()) {
        PayOut(_recipient, _amount);
        return true;
    } else {
        return false;
    }
}
```

出問題的 payOut

```
function payOut(address _recipient, uint _amount) returns (bool) {
    if (msg.sender != owner || msg.value > 0 || (payOwnerOnly && _recipient != owner))
        throw;
    if (_recipient.call.value(_amount)()) {
        PayOut(_recipient, _amount);
        return true;
    } else {
        return false;
    }
}
```

出問題的 payOut

```
function payOut(address _recipient, uint _amount) returns (bool) {
    if (msg.sender != owner || msg.value > 0 || (payOwnerOnly && _recipient != owner))
        throw;
    if (_recipient.call.value(_amount)()) {
        PayOut(_recipient, _amount);
        return true;
    } else {
        return false;
    }
}
```

把利潤退回去

退回利潤

因為每個使用者的錢包可以設定默認函數

所以可以在錢包收到這筆利潤的時候，再次呼叫 DAO 的 splitDAO

此時第一個 splitDAO 還在等錢包的默認函數結束

所以就卡住了

退回利潤

此時第一個 splitDAO 還在等錢包的默認函數結束

所以就卡住了

而且還有新的 splitDAO 呼叫，開始落入迴圈

“Recursive calling vulnerability”

其他問題

- 轉送代幣的部分用 recipient.call.value()() 並不指定最多能用多少 gas，所以這個 execution 能使用所有的 gas

  改進 => 使用 recipient.send，gas 只有預設的 2300 gas

- 即使限制 gas 的數量，攻擊者仍然可以針對這個弱點繼續攻擊

  改進 => 先做清算才做實際轉送

議程

- 當今情勢
- 區塊鏈 (Blockchain): 數位經濟基礎建設
- 區塊鏈: Trust Machine = 4D → EFG
- 區塊鏈的本質: Internet of Value as a Trust Machine
- 區塊鏈的信任安全
- Case Study: Blockchain vs. O2O核銷支付的挑戰
  - Blockchain 的解決方案: bitcoin, loyyal.com, blockpoint.io, Gcoin
- 數位經濟美麗新世界的展示

自動化智慧化的智能合約

智能合約的自動產生 →

1. 各商家智能發行點數
2. 用戶智能合約式的互相贈與交換點數
3. 用戶智能合約式的使用點數
   商家核銷點數

核銷支付的挑戰

當今挑戰: Too many O2O points

- 太多各自中心的序列號
- 商家頭痛如此多號, 多支付方式
  - Nightmare for application providers
  - Nightmare for SI 廠商
  - Nightmare for Hardware 廠商
  - Nightmare for Software 廠商

  → 商家使用O2O支付成本高昂
- 只能透過定型化契約保證
  - 信託
  - 履約保證

現行方法 -- 以大黑屋為例

- 線下交易例子:
  - 金卷平台
  - 大黑屋

- Cons
  - 實體化店面，展店成本
  - 無法統整不同店面的資訊

現行方法 -- 以8591為例

- 線上交易例子:
  - 8591平台
- Cons
  - 平台牽涉金錢交易
  - 金管會？

https://buzzorange.com/techorange/2014/08/28/the-greatest-sin-in-taiwan-is-to-embrace-innovation-and-entrepreneurial-dreams/

O2O核銷支付
結合區塊鏈技術

O2O核銷支付結合區塊鏈

如果結合區塊鏈技術：

- Pros:
  - 建置成本：增加新商家、新點數的邊際成本低
  - 廠商：分享用戶，以區塊鏈差異化，仍可追查金流
  - 用戶：輕鬆核銷/管理/取得點數
- Cons:
  - 建置成本：新技術、一開始的平台建置成本
  - 廠商：與其他廠商有競爭關係，共用客群，共用條約
  - 用戶：乏使用誘因，不想改變習慣

Case 1: 使用Bitcoin技術作為blockchain平台

Pros

目前最穩定的區塊鏈產品

數位化，台灣不認為是“貨幣”

交易紀錄透明可追蹤

Cons

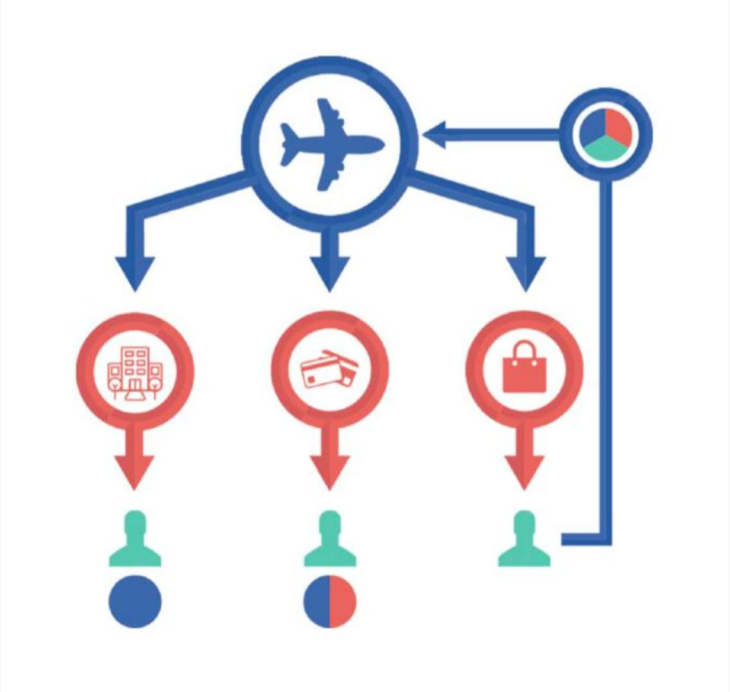
只有一種currency，串接各商家的點數票卷服務會有難度

交易頻率受限

Case 2: Blockchain 點數管理平台實際案例

美國今年5月17日推出的 loyyal.com

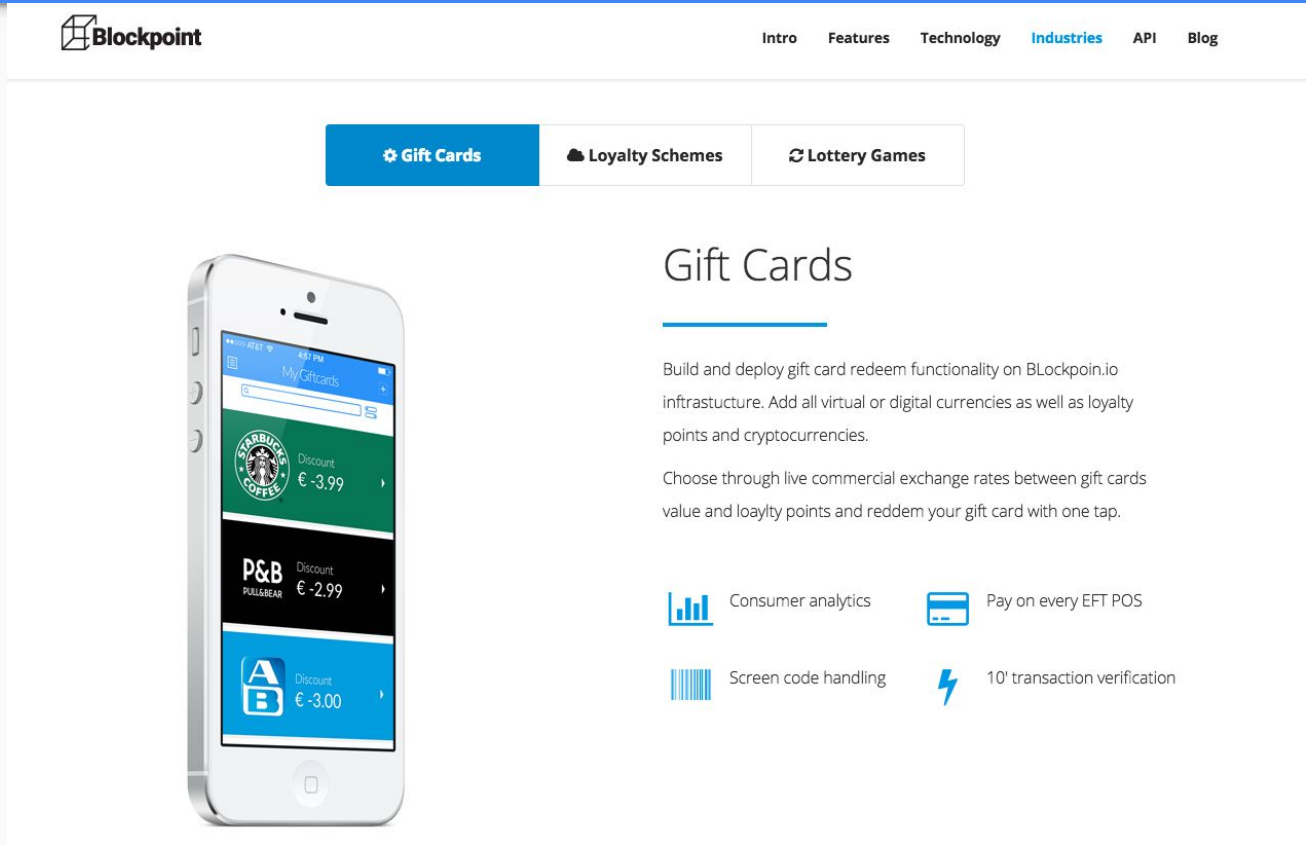
- 整合回饋金
- co-branded reward
- multi-branded reward



Case 3: Blockchain 點數管理平台實際案例

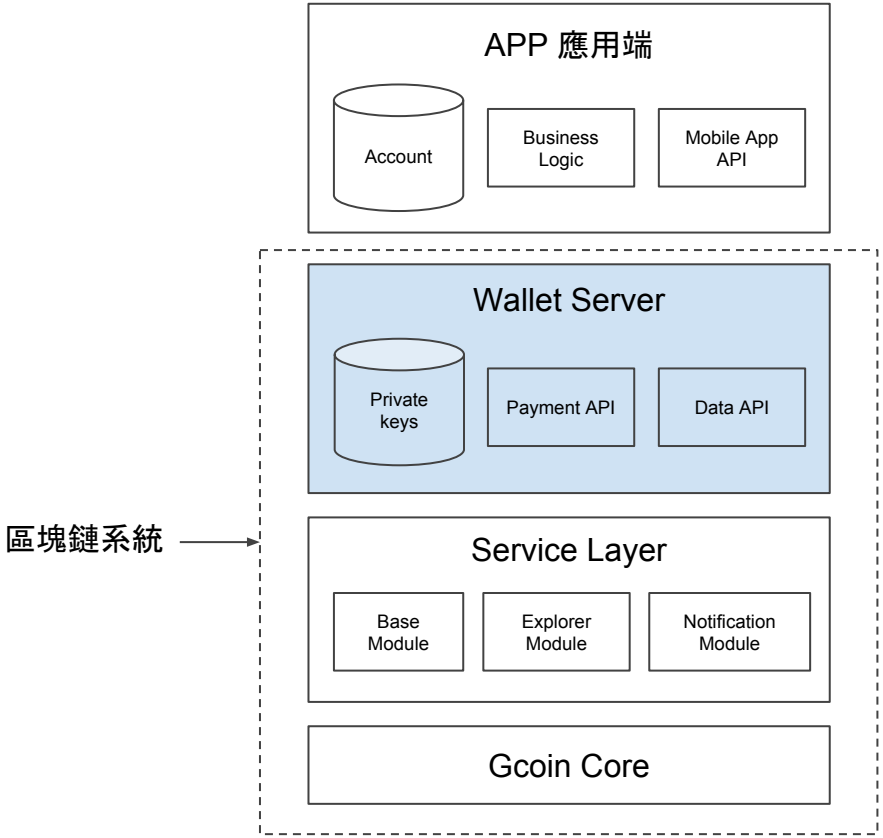
Blockpoint.io

- Gift Cards
- 管理不同折價券

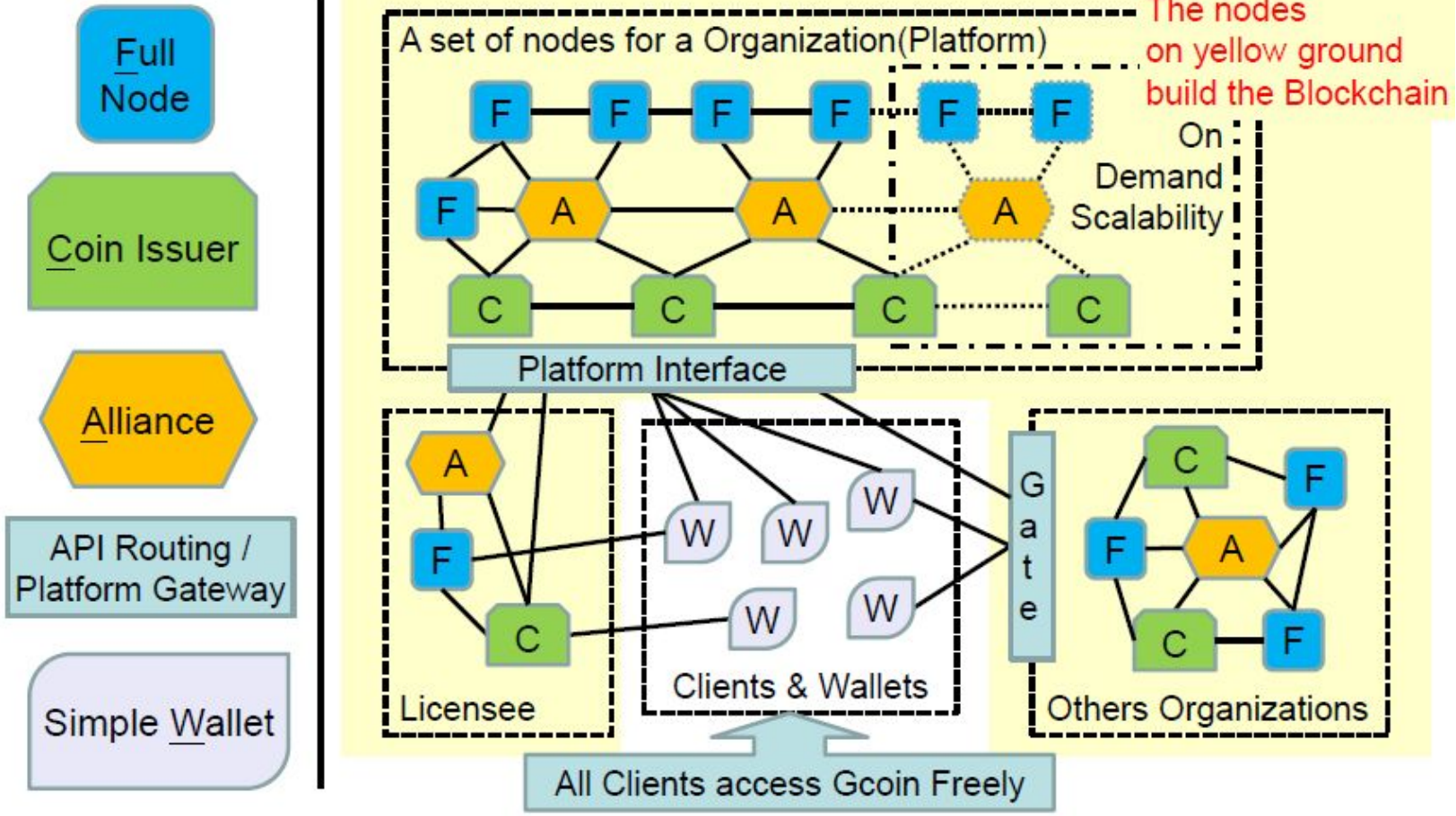


Gcoin API 串接

- 基於最穩定的blockchain開發
- Multi Currency, not bitcoin
- 可分為發行商 & 使用者
- 明確定義的API

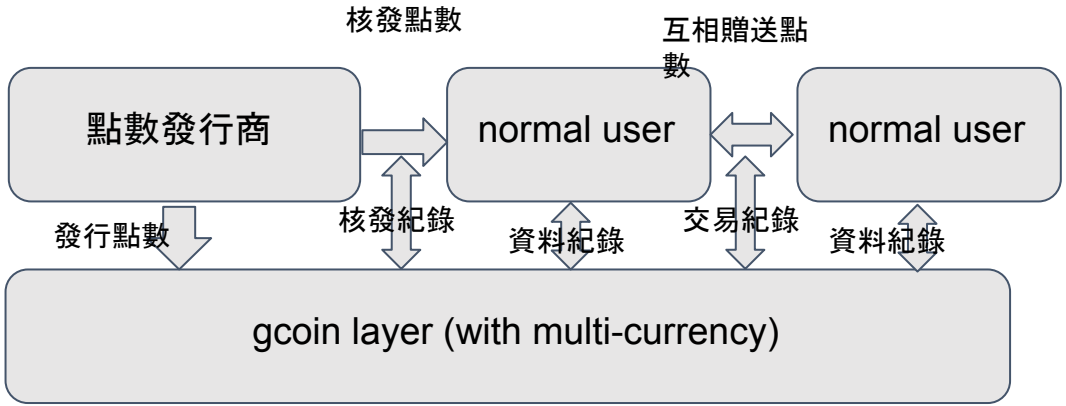


Gcoin Architecture



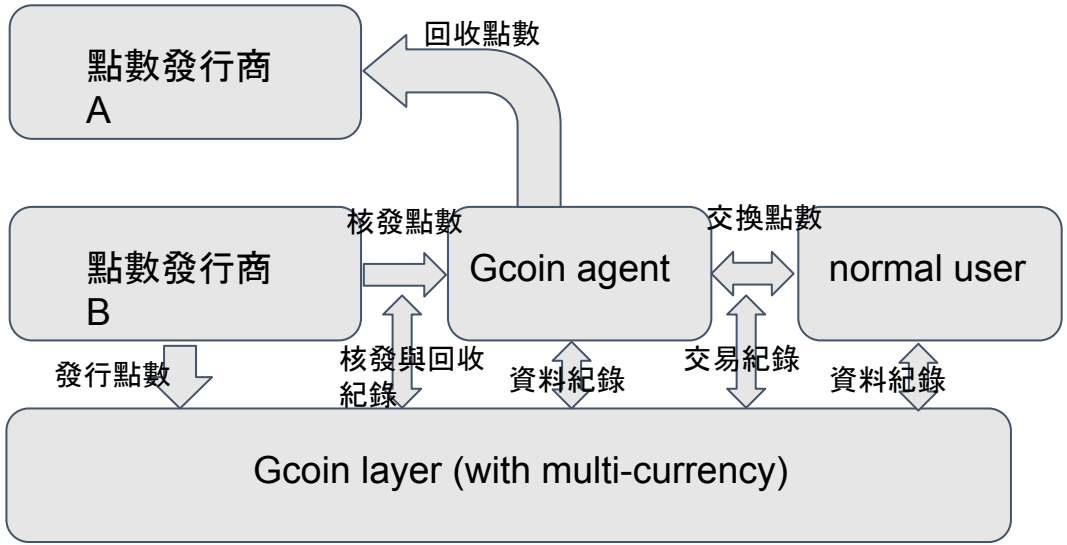
Step 1
點數管理平台 : 發行

- 技術 :
  - Gcoin layer: multi-currency
  - server side: 點數核發平台
  - 由程式提供履約保證,使用期限
- UI 呈現 :
  - 點數管理 app
  - Gcoin tx紀錄查詢



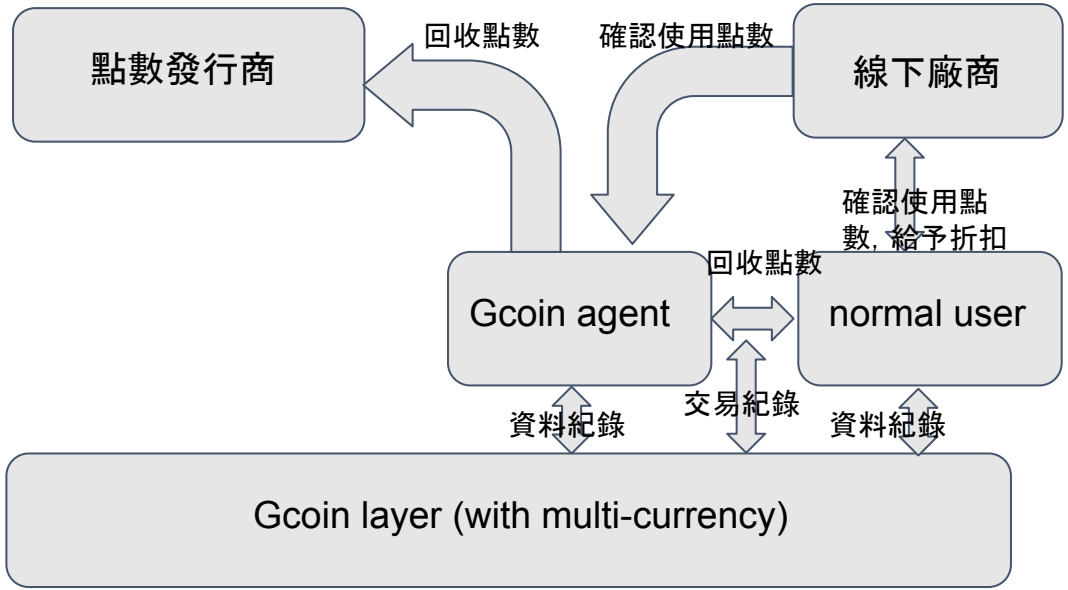
Step 2
點數管理平台 : 交換

- 技術
  - 儲存管理平台之技術
  - 點數交換流程
- 流程(點數A換成點數B)
  - 使用者交給agent點數A
  - agent回收點數A給A商
  - agent得到點數B從B商
  - agent給使用者點數B



Step 3
點數管理平台 : 核銷

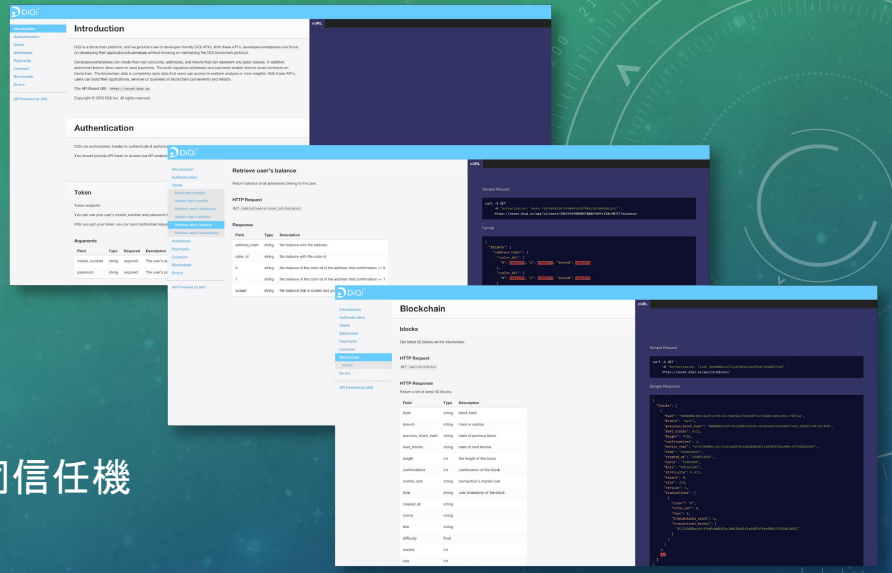
- 技術
  - multi-signature
- 流程
  - 線下廠商與使用者確認購買關係
  - 使用者交給agent點數A
  - 線下廠商簽約確認點數核銷
  - agent回收點數A給A商



數位經濟美麗新世界的展示

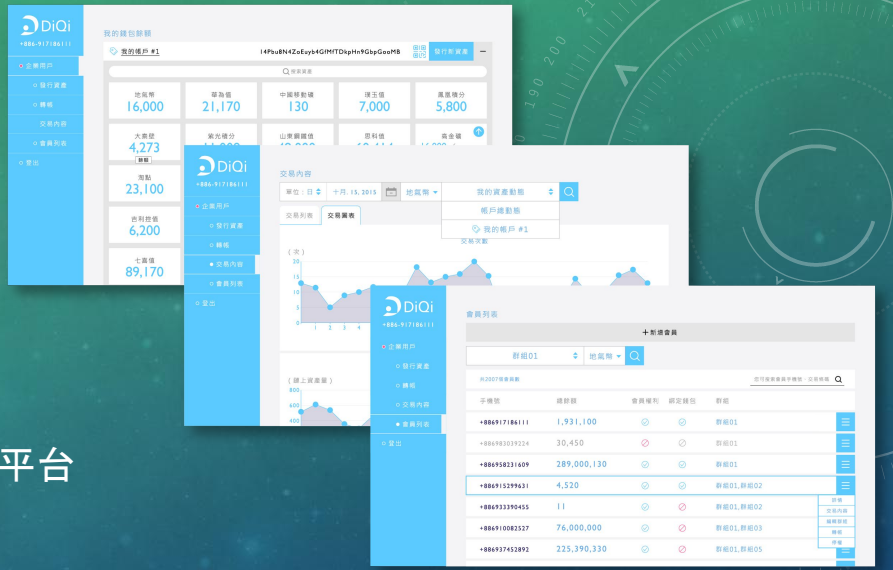
區塊鏈點數平台

- 各機構間使用區塊鏈，作到高效、快速的交易共同信任機制
- 第三方支付帳聯網
  - 一般用戶可在行動/第三方支付系統間互聯互通
  - 銀行/業者可快速連結帳聯網，享有互聯互通帶來的網路效應，將資源專注於提供創新用戶體驗與服務
- 供應鏈金融帳聯網
  - 營運資金和現金流量的實時監控和管理
  - 實現營運資金及現金流的靈活調度



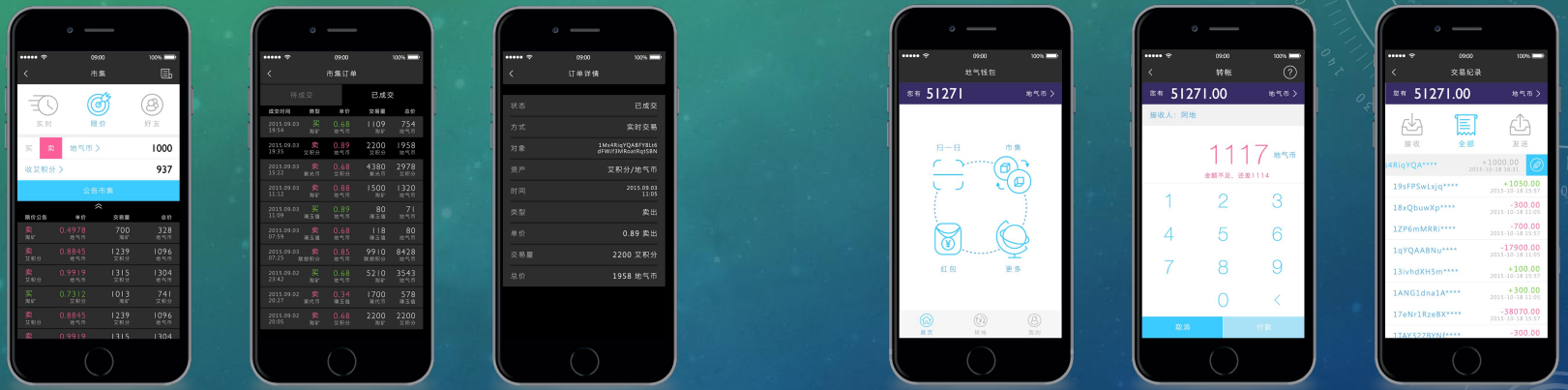
帳聯網 API

數字資產交易平台



- 具有強大可追蹤性與延展性的數字資產商品交易平台
  - 實物商品：電影票、演唱會、藝術品
  - 非實物商品：貸款商品、股權商品、債權商品、智慧財產權
- 平台運營商 (Platform operator)、網路商家 (Merchant) 互聯互通
  - 提供平台運營商互相加盟的環境，例如：票據投融平台。提供網路商家可同時在多個平台運營商之間獲得客戶

區塊鏈:點數管理平台及交易平台



- 區塊鏈數位經濟還有大數據好處
  - 大數據預測分析 – 依據用戶行為做定向廣告
  - 商務情報 – 提供對業務經理人具洞見的指標指數

GCOIN 區塊鏈使用情境

支付結算系統

顧客忠誠計劃

crowdsourcing 平台

私募平台

票據平台

去中心化交易所