2023 智能合約實戰演練

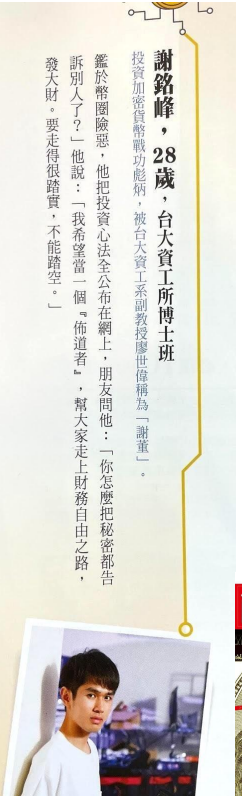
2023

Speaker: Cupid Sie 謝銘峰

Speaker Intro

謝銘峰 Sie Ming Fong

Creditor



經歷

- SuDo Research Labs Researcher
- Smart Contract Research Forum Research Fellow
- 台灣區塊鏈之父 NTU SUIFT 共識實驗室
- 哥，我塊步行了、區塊勢 Podcaster 靈界偵探
- 區塊鏈大學聯盟理事
- Google University Relations 計劃研究員
- Polonex 交易所 Market Maker 月交易量達 500 萬美金
- 台大資工博士候選人、師大資訊教育研究所碩士

研究領域

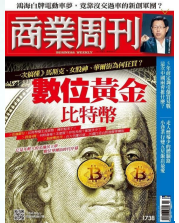
- DeFi、公鏈、合約審計、反洗錢、人工智慧

近期項目

- 2022 區塊鏈愛好者協會平方募資法海洋區塊鏈 第三名
- 2020 總統盃黑客松 海洋區塊鏈 垃圾變黃金 入圍前六強
- 2020 金管會監理松 打虎抓賊也著親兄弟, 希望的種子 AI 企業健檢醫生 入圍全國複賽

媒體報導

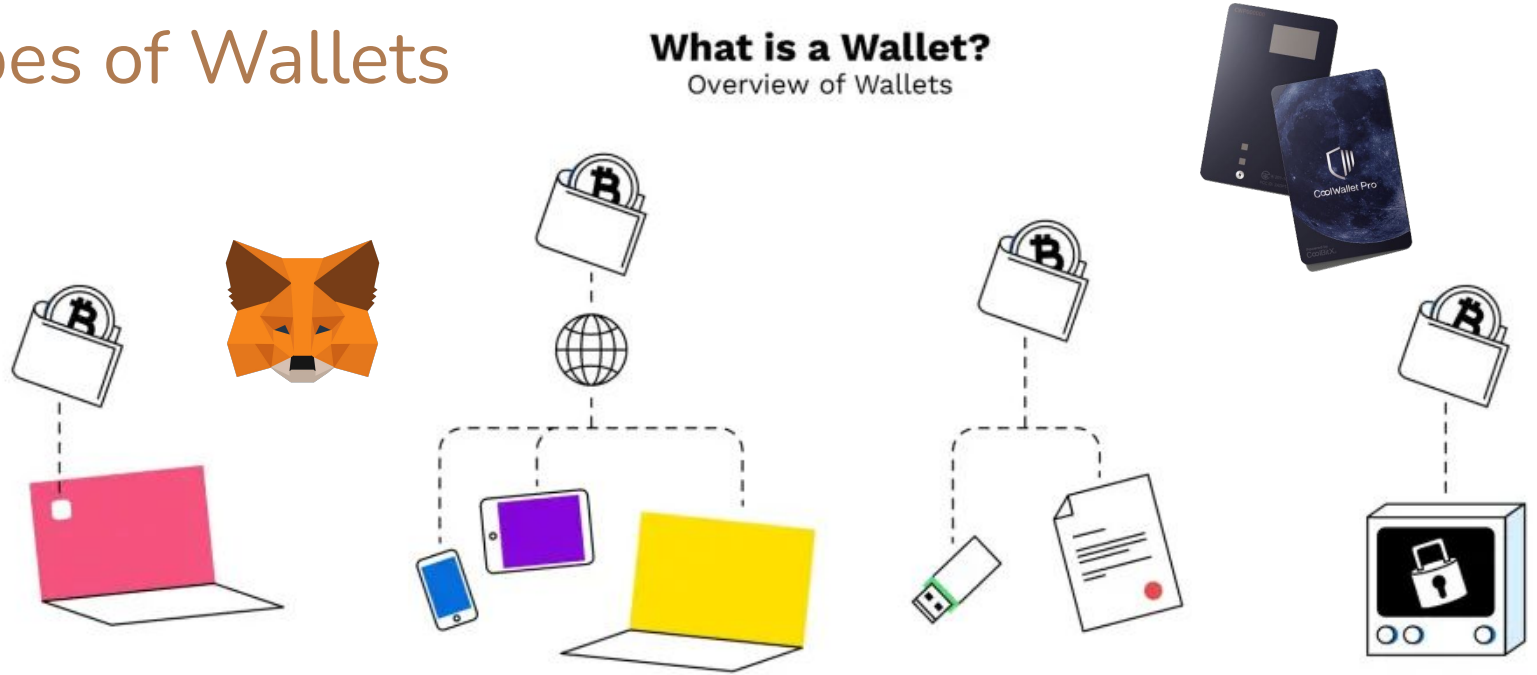
- 2021/07 遠見雜誌 台大資工幣圈高手！百萬虛幣秒蒸發，悟出分散投資之道
- 2021/03 商業週刊 1738 期 數位黃金比特幣
- 2021/04 三立新聞 成員個個身價百萬.千萬! 揭密比特幣百萬富翁
- 2018/10 數位時代 韭菜還是玩家？看發幣方和買幣人的歡笑與淚水
- 2017/08 非凡新聞 從挖礦到ICO 台大生半年賺本金30倍

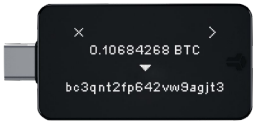


區塊勢

Smart Contract Research Forum

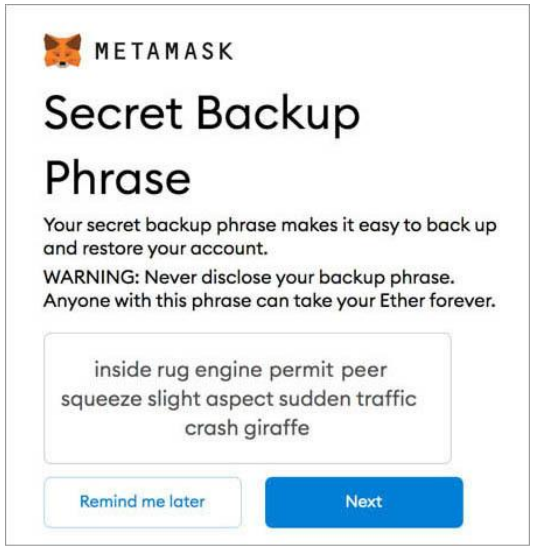
Types of Wallets

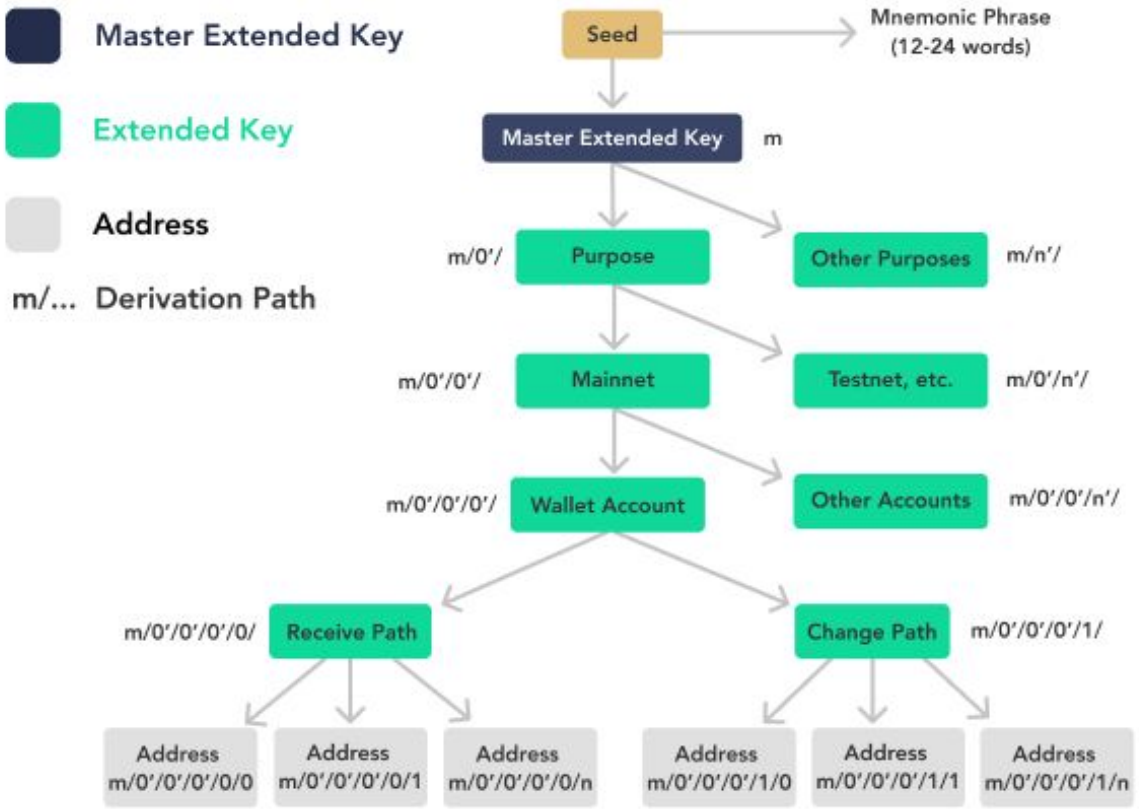




Source: [Safe Storage](), [dchained](), [shiftcrypto]()

How Wallet Work

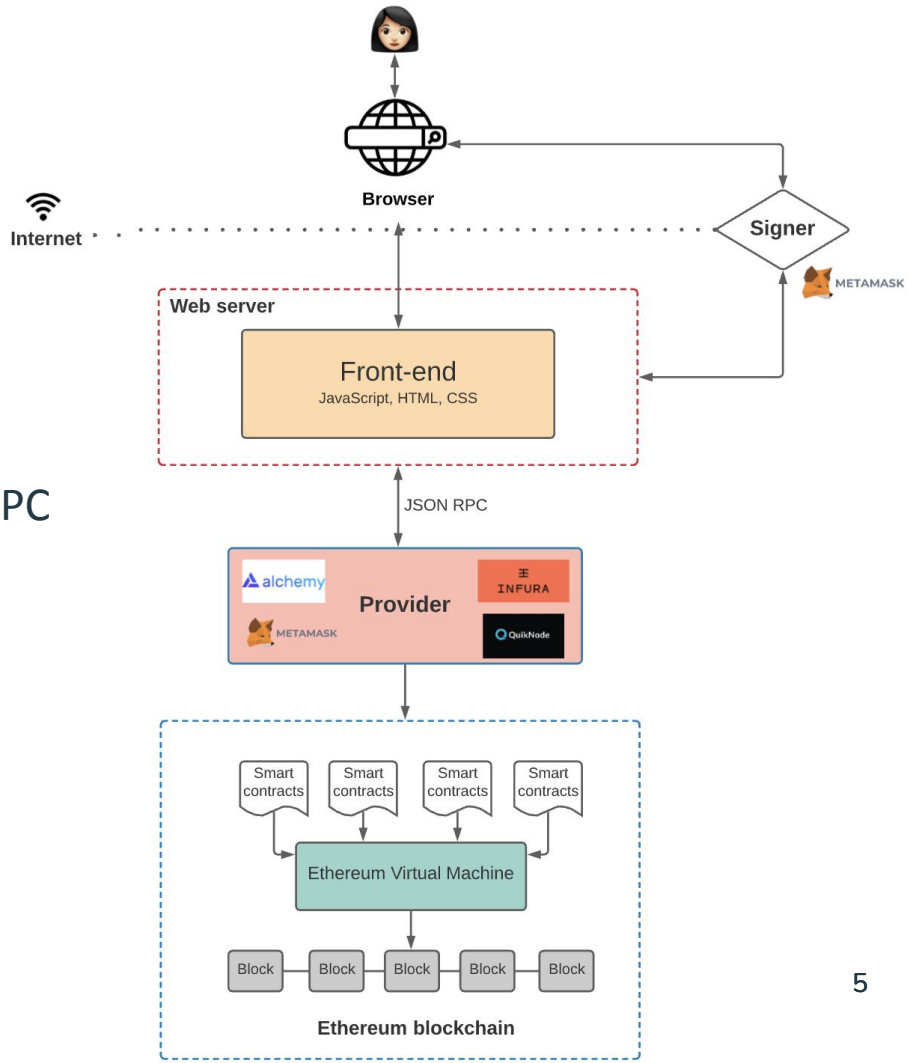




Source: River Finance

Wallet mechanism

- Frontend provide user interface
  - Website
  - App
- Backend - ethereum client provide JSON RPC
  - Infura
  - Alchemy
  - Metamask
  - Quicknode



Source: [Preethi Kasireddy]

Cold Wallet - Paper Wallet

- Website Generate
  - Public Key
  - Private Key

Source: dchained, BitAddress

How to survived - You Own Your Assets

- Cold Wallet Generate
  - Public Key
  - Private Key
- Different Cold Wallet Companies
  - Trezor founded in 2013
  - Ledger founded in 2014
  - Coolbitx founded in 2014

Cold Wallet Support Chains

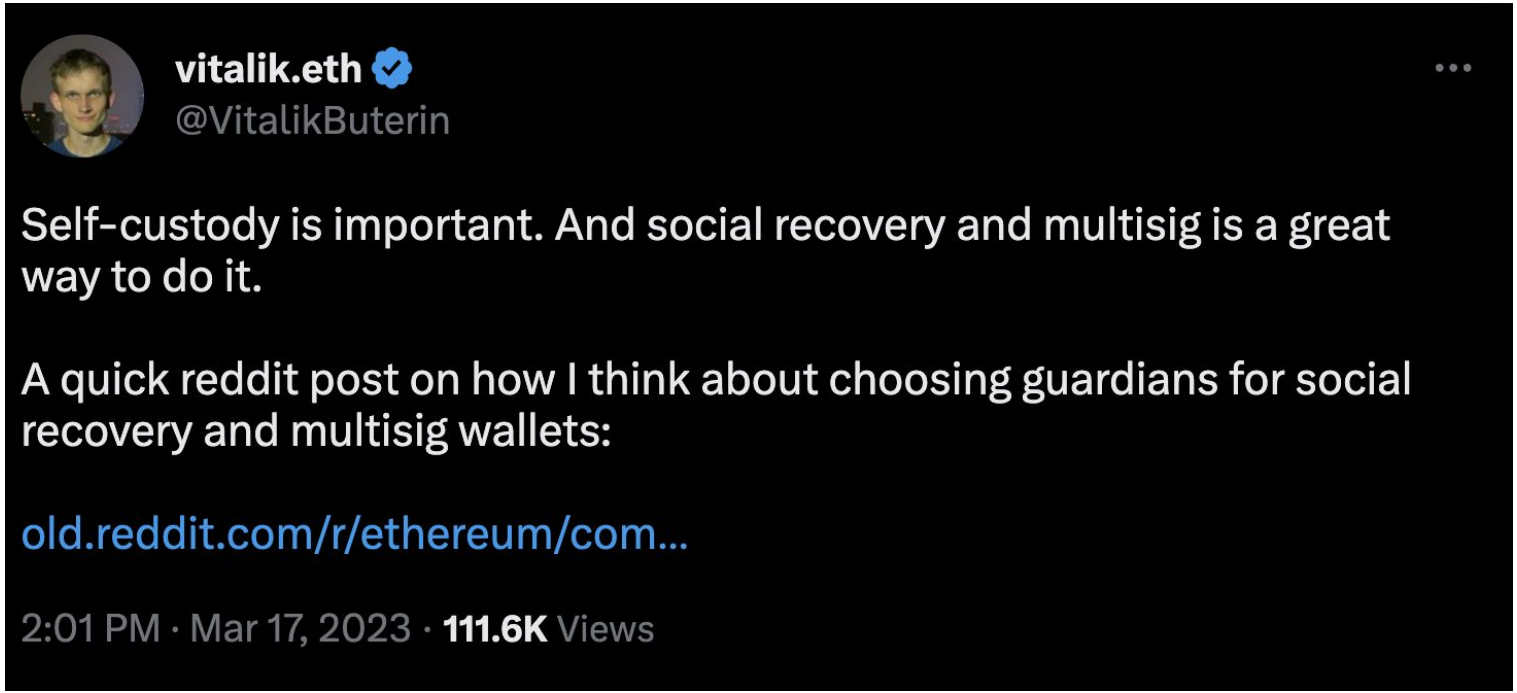
Types of Cold Wallets

The most popular cold wallets brands, include:

Name	Purchase Cost	Supported Coins	NFT Support?
Ledger Nano X	$175	BTC, ETH, XRP, BCH, DOT, LTC, TRX, EOS, XLM, ADA, etc.	Yes
Trezor Model T	$195	BTC, ETH,USDT,ADA, XRP, DOGE, LTC, XMR, etc.	Yes
Ledger Nano S	$59	BTC, ETH, XRP, BCH, DOT, LTC, TRX, EOS, XLM, ADA, etc.	Yes
ELLIPAL Titan	$139	BTC, ETH, LTC, XRP, XLM, BNB, USDT, TRX, DOT, etc.	Yes
CoolWallet Pro	$149	BTC, ETH, LTC, XRP, XLM, BNB, USDT, TRX, Tezos, Cardano, ETH 2.0, and ERC20 tokens.	Yes
Safepal S1	$49.99	BTC, ETH, XRP, LTC, XLM, TRX, DOGE, DOT, POLYGON, ADA, etc.	Yes
Keystone Pro	$169	BTC, ETH, Tether, XRP, Bitcoin Cash, Polkot, LTC, Kucoin, etc.	Yes
Keepkey	$190	Bitcoin, Bitcoin Cash, Bitcoin Gold, DASH, Dogecoin, Namecoin, Ethereum, Litecoin, etc.	Yes

Source: Horizen.io, consensys

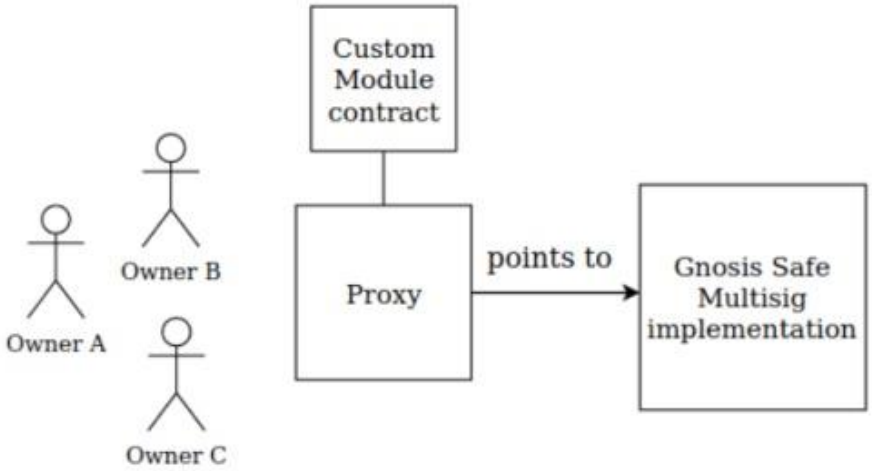
Vitalik: Self-custody is important.



Source: Vitalik Twitter

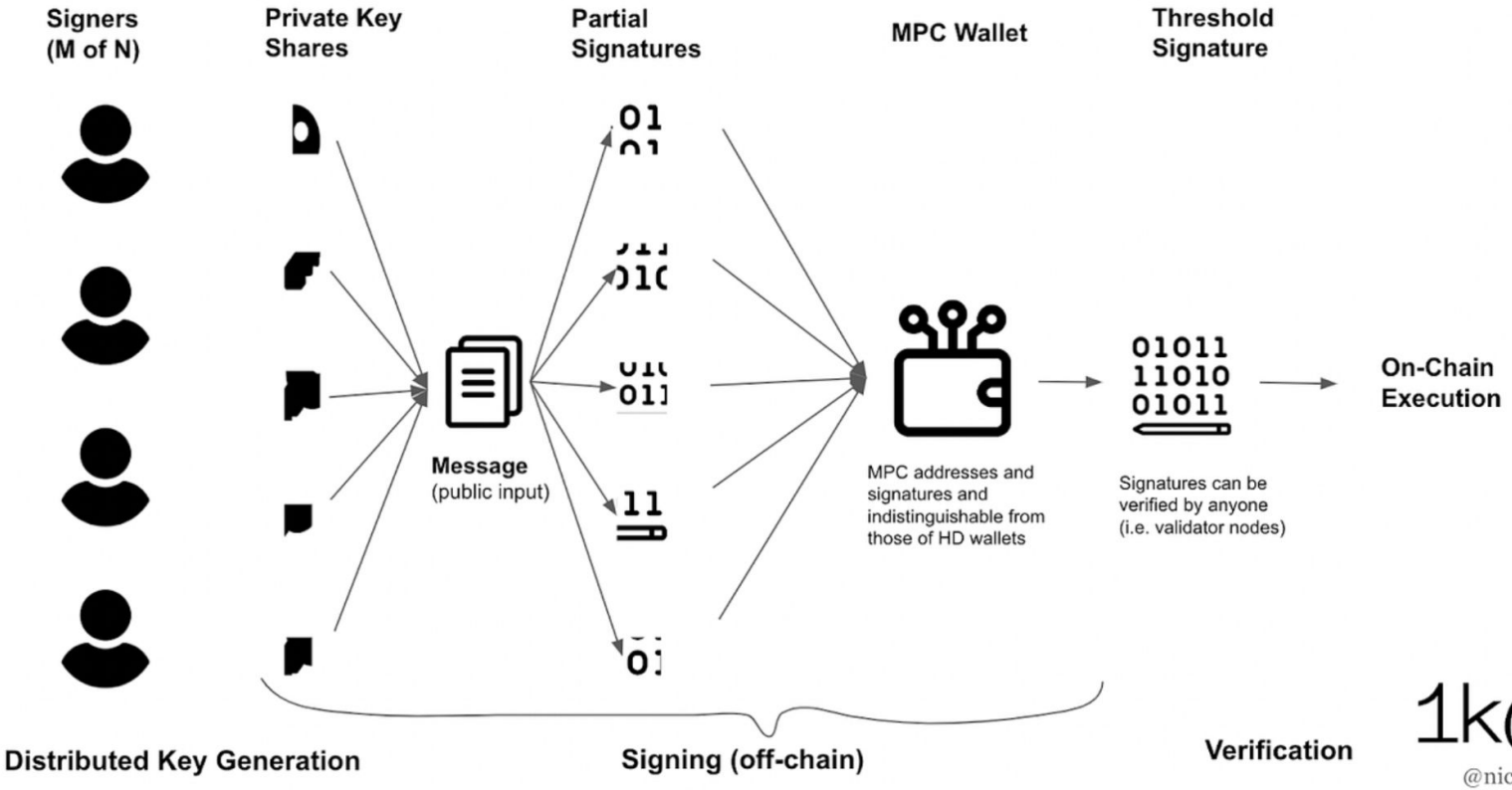
Gnosis Safe

- Smart Contract Account
  - An account manage by at least two private key
  - Access controlled by many private keys enabling co-ownership and stronger security for individuals



Source: OpenZapplin, Gnosis Safe

Multi-Party Computation(MPC) Wallet

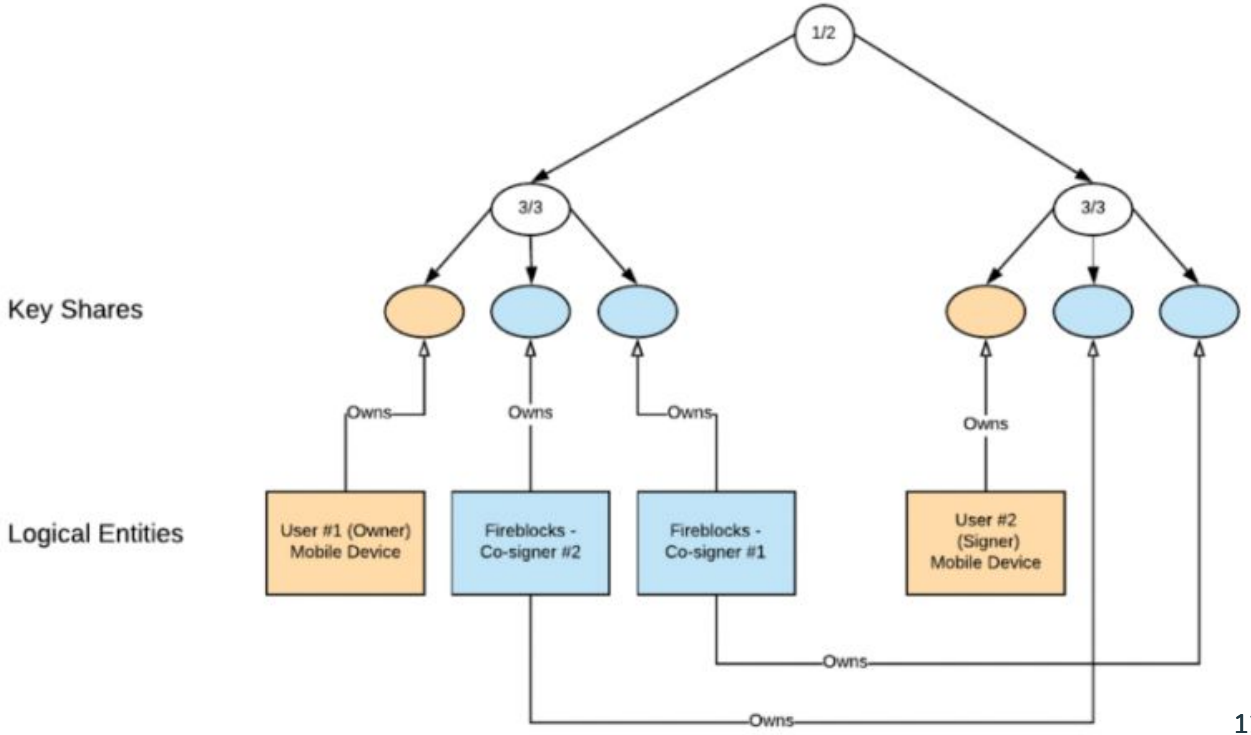


1k(x)
@nichanank

Source: [Seedless Self-Custody: On MPC and Smart Contract Wallets]

Fireblocks Multi-Party Computation(MPC) Wallet

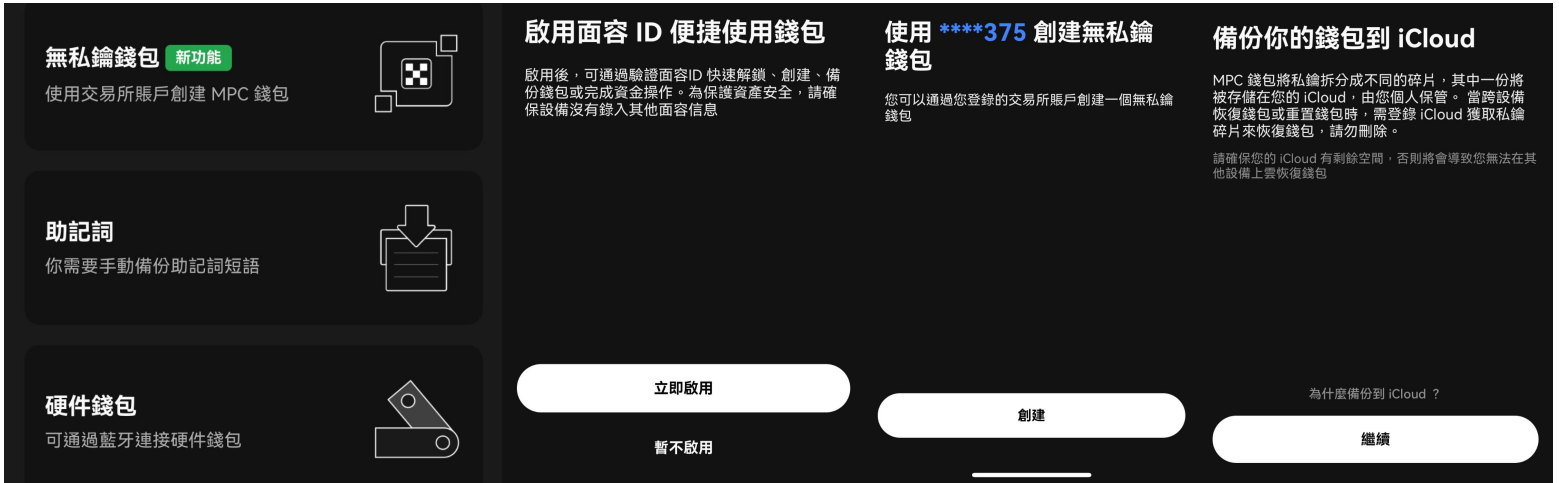
- 3 Key Shares
  - A piece on Client Device
  - Two pieces on the platform



Source: [MPC 钱包调研报告：下一代去中心化钱包解决方案]

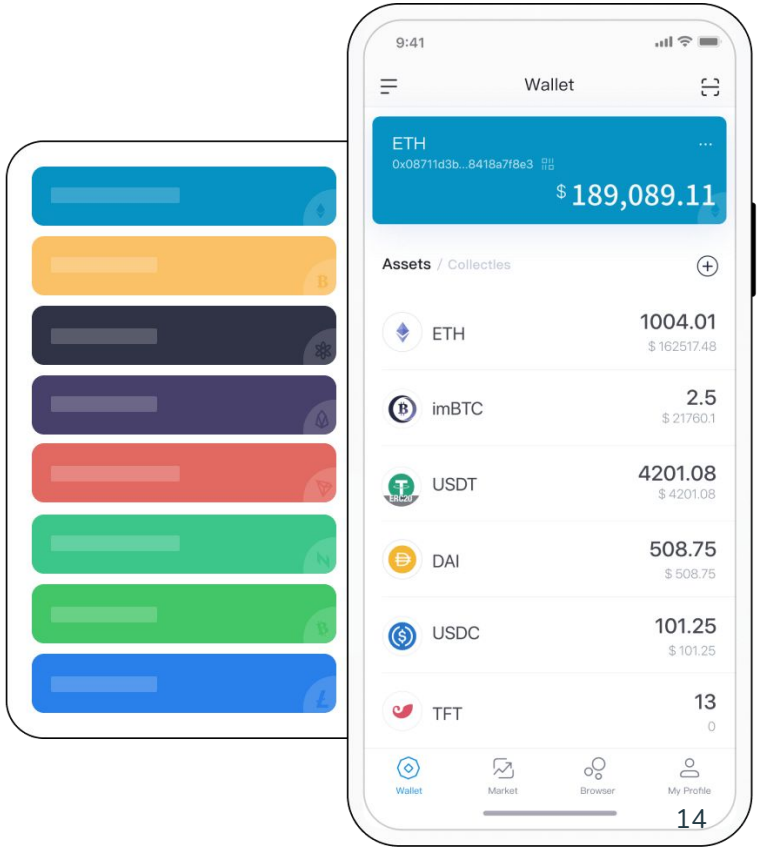
OKX MPC Wallet

- No Private Key
- Phone Device, iCloud/Google, OKX Server backup (2 of 3 recover)
- Support 37 chains
- Do not support some token Airdrop



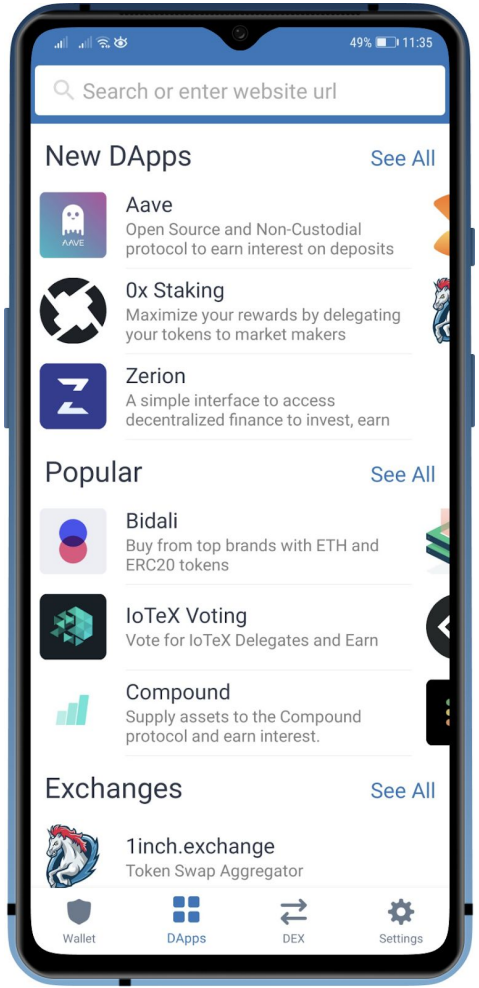
How to survive - You Own Your Assets

- Hot Wallet Generate
  - Public Key
  - Private Key
- Different Hot Wallet Service:
  - MetaMask founded in 2016
  - imToken founded in 2016
  - 麥子錢包 founded in 2017
  - Trust Wallet founded in 2017
  - Coinbase Wallet founded in 2018
  - ZenGo multiplichain MPC Wallet founded in 2018
  - Blocto Wallet founded in 2019
  - OKX multichian MPC Wallet lunched in 2023



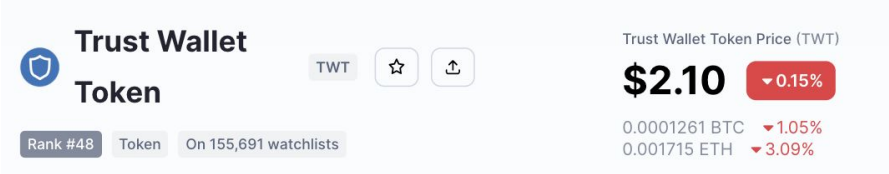
DeFi application with your wallet

- Exchange
- Lending / Borrowing
- Staking
- Farming
- Bridge
- Buy coin with your credit card / apple pay / google pay

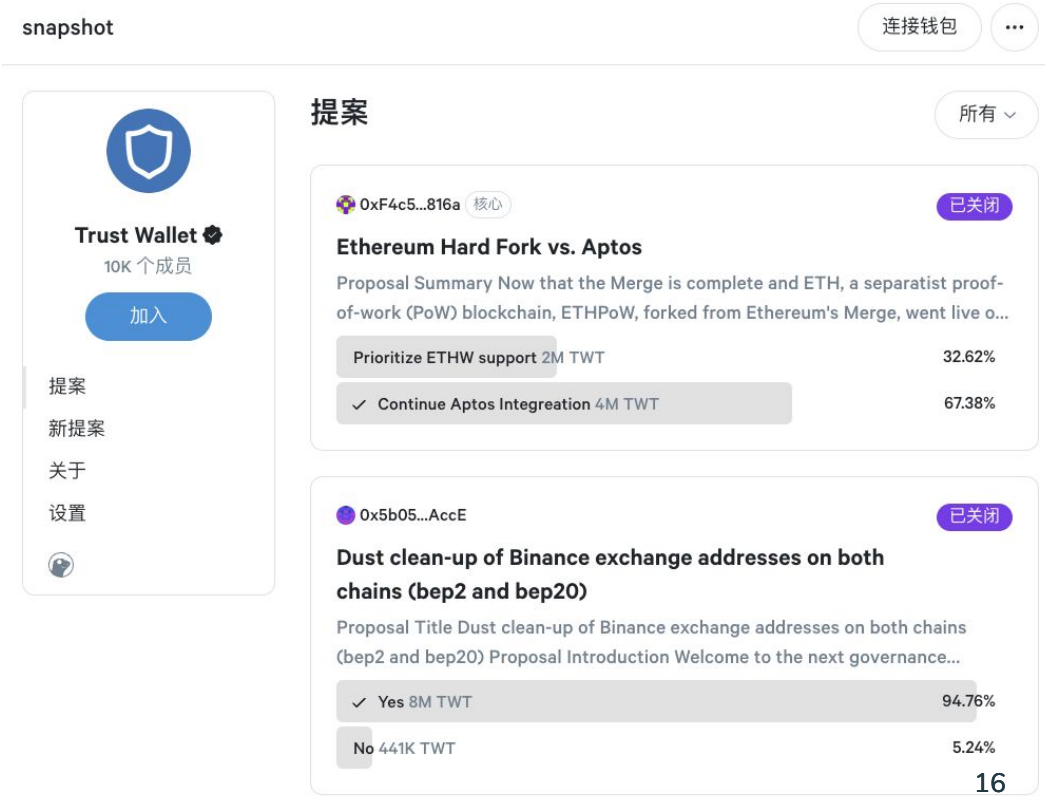


Source: [Trust Wallet]

Wallet Governancn Token



- Total Supply: 1,000,000,000 TWT
- BEP 20 Token
- Participate in the voting processes
- Discounts on DEX services and cryptocurrency purchases
- Trust Wallet Token (TWT) doesn't hold any real value outside its ecosystem



Source: Coinmarketcap, Trust Governance

Smart Contract Implementation

Blockchain Wallet

- Wallet 支援 CLI(Commend Line Interface) 或 GUI(Graphical User Interface)方式
- 以太坊有兩種不同類型的帳戶 : 外部擁有帳戶 (EOAs) 和合約帳戶 (Smart Contract Account)
- EOAs 由以太坊以外的軟體 (如錢包應用程式) 控制
- 合約帳戶由在以太坊虛擬機 (EVM) 內運行的軟體控制

Source: Mastering Ethereum

Ethereum Smart Contract

- Smart Contract 是在區塊鏈上的程式, 合約與執行結果都會儲存在區塊鏈上
- 以太坊的 Smart Contract 運行於 EVM(Ethereum Virtual Machine)上, 程式語言為 Solidity
- Smart Contract 部署後便不可更改
- 合約運行時需要耗費 gas 若沒有足夠數量的以太幣則無法運行合約
  - e.g. Gas Price:0.001 ether, Total Gas:100

Total Cost = 0.001 * 100 = 0.1ether

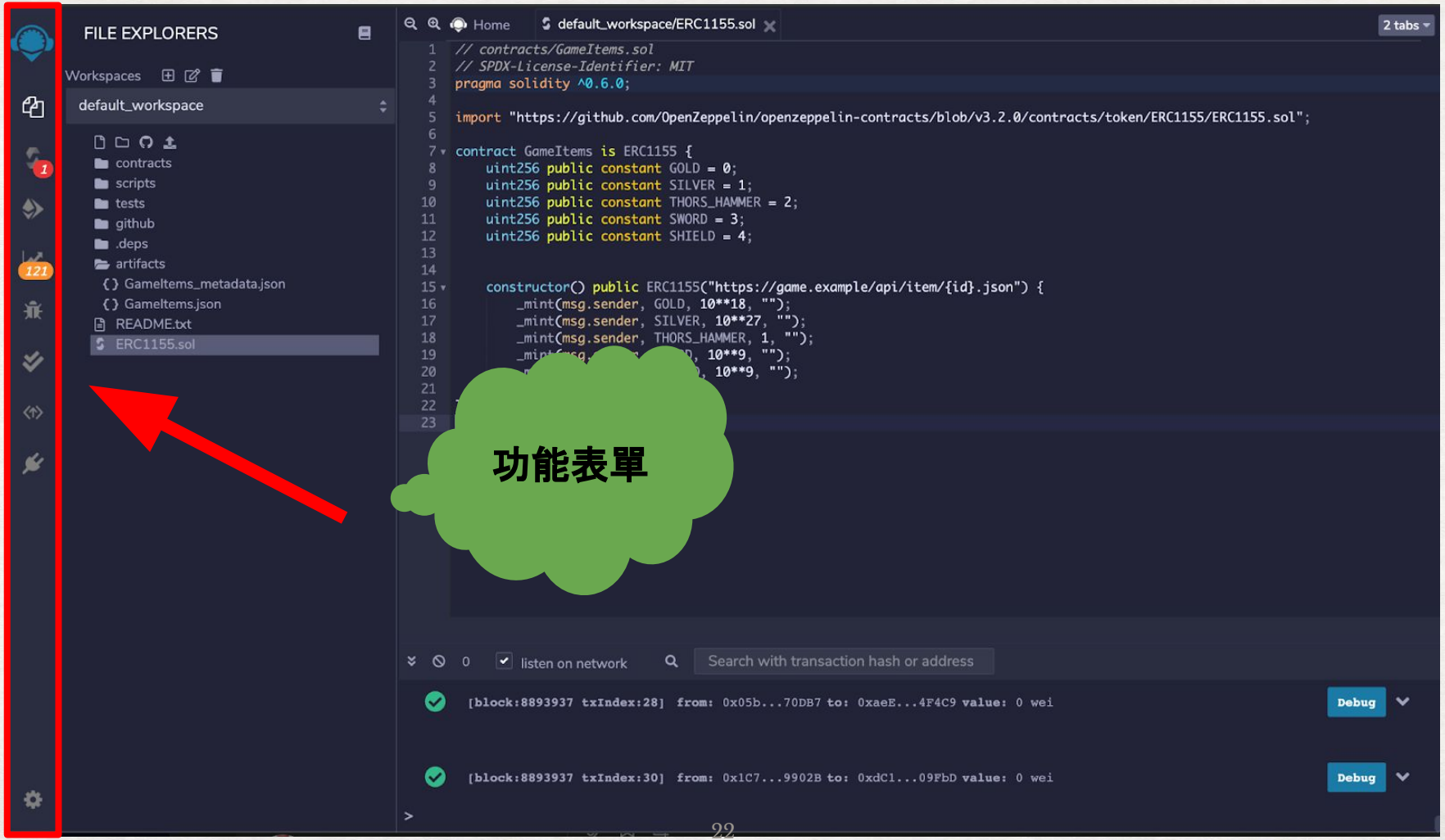
Solidity 

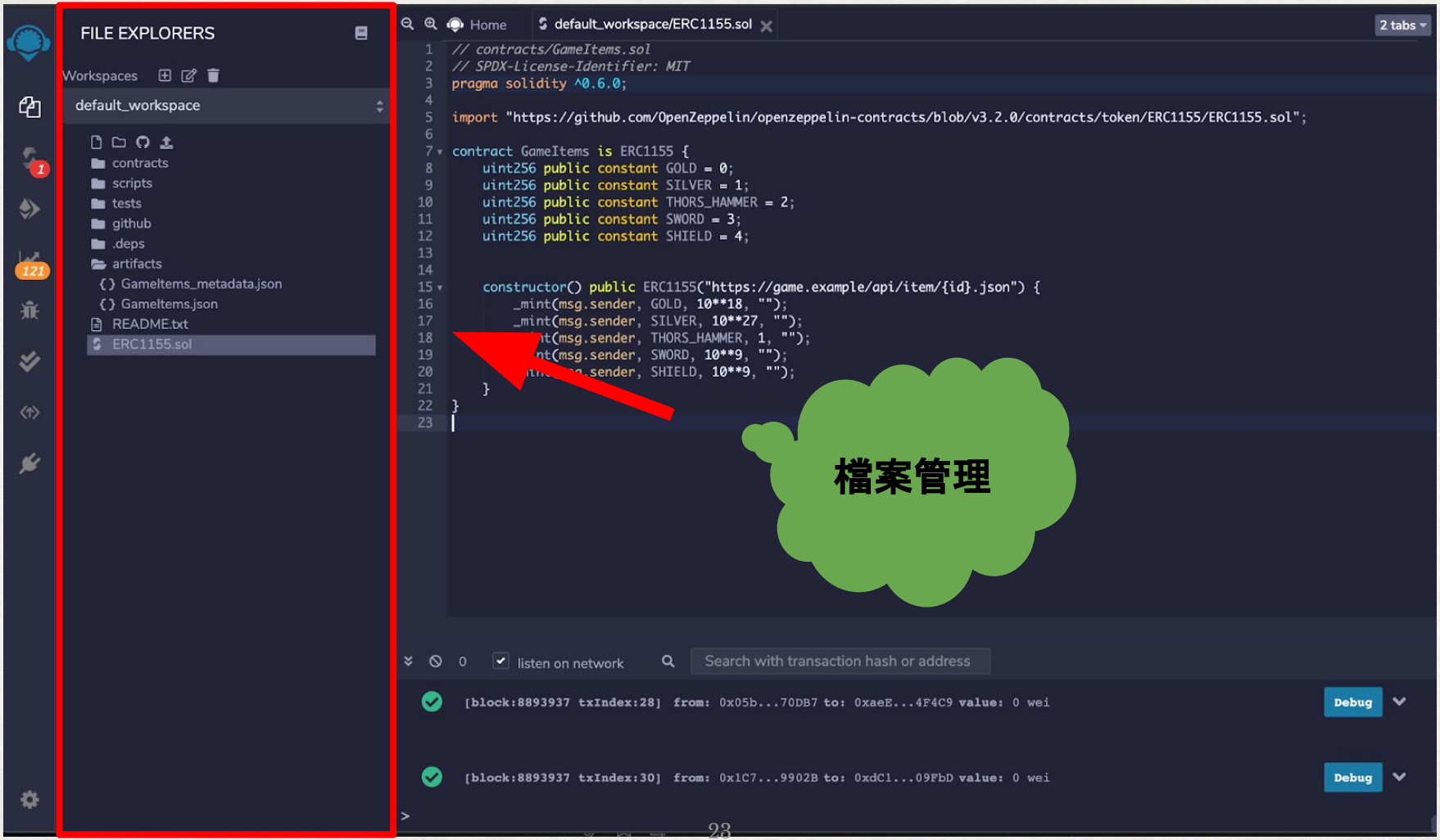
- Solidity 是一種物件(合約)導向的程式語言、為實現智能合約而創建的高級程式語言。
- Solidity 受到 C++, Python 和 Javascript 影響, 設計的目的是能在以太坊虛擬機(EVM)上運行。
- 支援複雜的使用自定義函式, libraries 和繼承。
-
- Solidity 可以建立智能合約如投票、審核、群眾募資等合約應用。

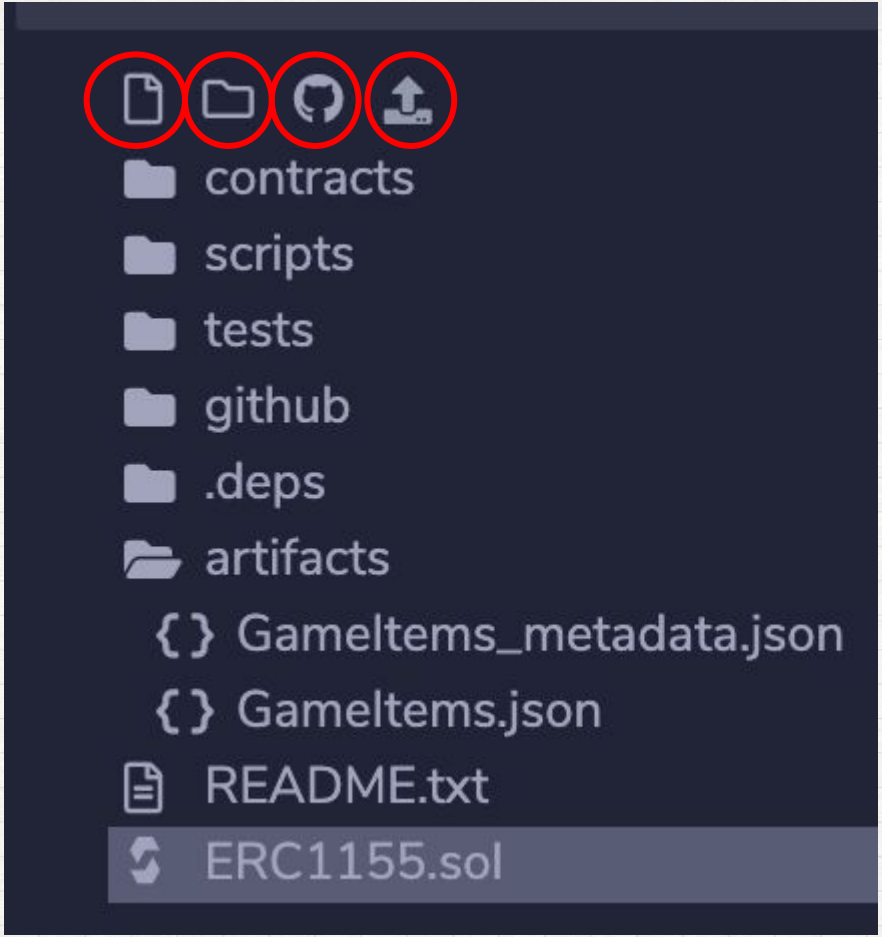
Ethereum smart contract

Remix IDE

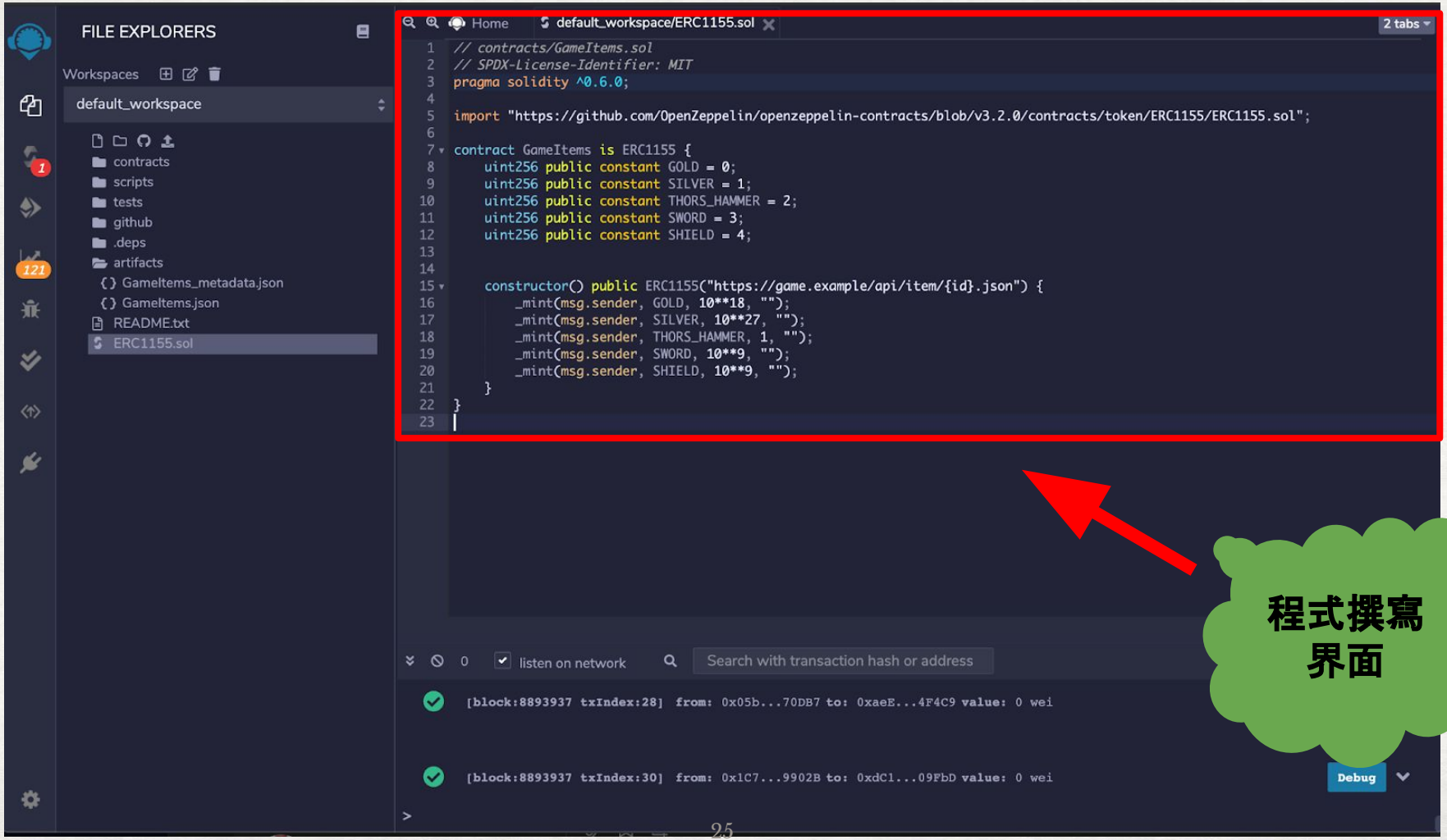
- https://remix.ethereum.org
- 一種網頁式IDE(Intergrated Development Environment), 用於開發 Solidity 所撰寫的智能合約。
- 其內建功能包括:
  - 編譯
  - 除錯
  - 合約模擬 (本身提供5個虛擬帳號, 各帳號含100個ether)
  - 合約部署

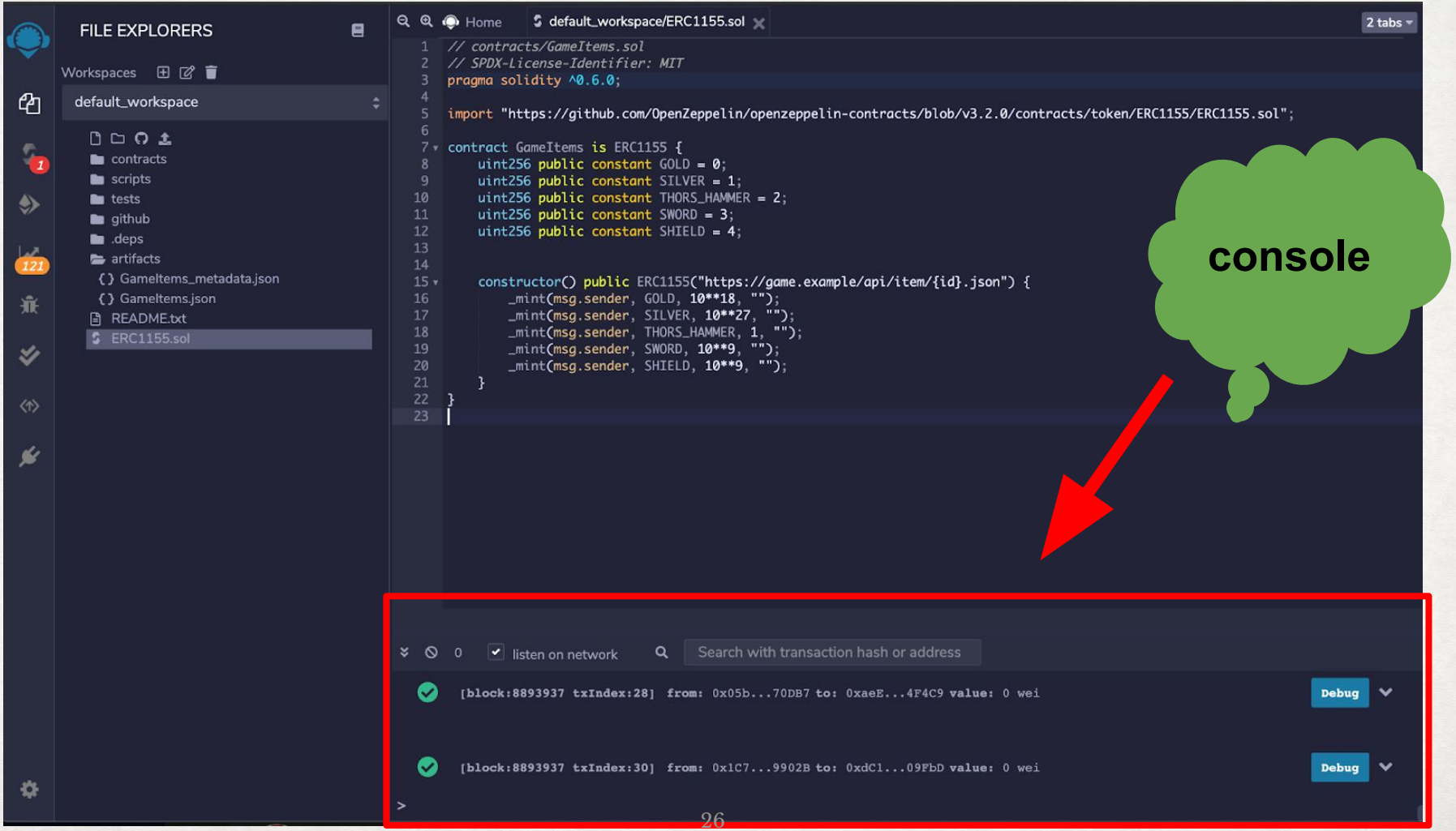

FILE EXPLORERS
Workspaces
default_workspace
contracts
scripts
tests
github
.deps
artifacts
GameItems_metadata.json
GameItems.json
README.txt
ERC1155.sol
Home
default_workspace/ERC1155.sol
2 tabs
// contracts/GameItems.sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.6.0;
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v3.2.0/contracts/token/ERC1155/ERC1155.sol";
contract GameItems is ERC1155 {
uint256 public constant GOLD = 0;
uint256 public constant SILVER = 1;
uint256 public constant THORS_HAMMER = 2;
uint256 public constant SWORD = 3;
uint256 public constant SHIELD = 4;
constructor() public ERC1155("https://game.example/api/item/{id}.json") {
_mint(msg.sender, GOLD, 10**18, "");
_mint(msg.sender, SILVER, 10**27, "");
_mint(msg.sender, THORS_HAMMER, 1, "");
10**9, "");
10**9, "");
功能表單
listen on network
Search with transaction hash or address
[block:8893937 txIndex:28] from: 0x05b...70DB7 to: 0xaeE...4F4C9 value: 0 wei
Debug
[block:8893937 txIndex:30] from: 0x1C7...9902B to: 0xdC1...09FbD value: 0 wei
Debug


FILE EXPLORERS
Workspaces
default_workspace
contracts
scripts
tests
github
.deps
artifacts
GameItems_metadata.json
GameItems.json
README.txt
ERC1155.sol
Home
default_workspace/ERC1155.sol
2 tabs
// contracts/GameItems.sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.6.0;
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v3.2.0/contracts/token/ERC1155/ERC1155.sol";
contract GameItems is ERC1155 {
uint256 public constant GOLD = 0;
uint256 public constant SILVER = 1;
uint256 public constant THORS_HAMMER = 2;
uint256 public constant SWORD = 3;
uint256 public constant SHIELD = 4;
constructor() public ERC1155("https://game.example/api/item/{id}.json") {
_mint(msg.sender, GOLD, 10**18, "");
_mint(msg.sender, SILVER, 10**27, "");
_mint(msg.sender, THORS_HAMMER, 1, "");
_mint(msg.sender, SWORD, 10**9, "");
_mint(msg.sender, SHIELD, 10**9, "");
}
}
檔案管理
0
listen on network
Search with transaction hash or address
[block:8893937 txIndex:28] from: 0x05b...70DB7 to: 0xaeE...4F4C9 value: 0 wei
Debug
[block:8893937 txIndex:30] from: 0x1C7...9902B to: 0xdC1...09FbD value: 0 wei
Debug

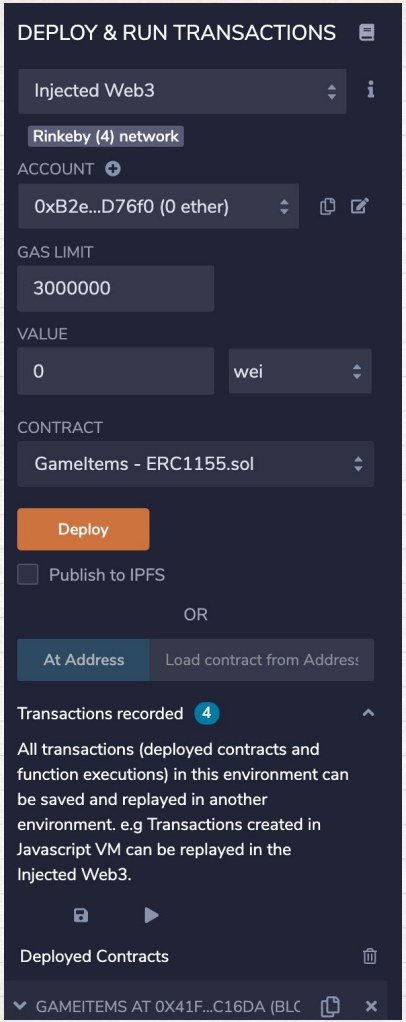


1. 新增檔案
2. 新增檔案夾
3. 連結 gist
4. 上傳本機檔案(local host)


FILE EXPLORERS
Workspaces
default_workspace
contracts
scripts
tests
github
.deps
artifacts
GameItems_metadata.json
GameItems.json
README.txt
ERC1155.sol
Home
default_workspace/ERC1155.sol
2 tabs
// contracts/GameItems.sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.6.0;
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v3.2.0/contracts/token/ERC1155/ERC1155.sol";
contract GameItems is ERC1155 {
uint256 public constant GOLD = 0;
uint256 public constant SILVER = 1;
uint256 public constant THORS_HAMMER = 2;
uint256 public constant SWORD = 3;
uint256 public constant SHIELD = 4;
constructor() public ERC1155("https://game.example/api/item/{id}.json") {
_mint(msg.sender, GOLD, 10**18, "");
_mint(msg.sender, SILVER, 10**27, "");
_mint(msg.sender, THORS_HAMMER, 1, "");
_mint(msg.sender, SWORD, 10**9, "");
_mint(msg.sender, SHIELD, 10**9, "");
}
}
listen on network
Search with transaction hash or address
[block:8893937 txIndex:28] from: 0x05b...70DB7 to: 0xaeE...4F4C9 value: 0 wei
[block:8893937 txIndex:30] from: 0x1C7...9902B to: 0xdC1...09FbD value: 0 wei
Debug
程式撰寫
界面


FILE EXPLORERS
Workspaces
default_workspace
contracts
scripts
tests
github
.deps
artifacts
GameItems_metadata.json
GameItems.json
README.txt
ERC1155.sol
Home
default_workspace/ERC1155.sol
2 tabs
// contracts/GameItems.sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.6.0;
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v3.2.0/contracts/token/ERC1155/ERC1155.sol";
contract GameItems is ERC1155 {
uint256 public constant GOLD = 0;
uint256 public constant SILVER = 1;
uint256 public constant THORS_HAMMER = 2;
uint256 public constant SWORD = 3;
uint256 public constant SHIELD = 4;
constructor() public ERC1155("https://game.example/api/item/{id}.json") {
_mint(msg.sender, GOLD, 10**18, "");
_mint(msg.sender, SILVER, 10**27, "");
_mint(msg.sender, THORS_HAMMER, 1, "");
_mint(msg.sender, SWORD, 10**9, "");
_mint(msg.sender, SHIELD, 10**9, "");
}
}
console
listen on network
Search with transaction hash or address
[block:8893937 txIndex:28] from: 0x05b...70DB7 to: 0xaeE...4F4C9 value: 0 wei
Debug
[block:8893937 txIndex:30] from: 0x1C7...9902B to: 0xdC1...09FbD value: 0 wei
Debug


等待驗證中的交易
清除畫面
顯示所有交易
交易訊息
0
listen on network
Search with transaction hash or address
[block:8894005 txIndex:35] from: 0x3DC...9D5f3 to: 0xc39...74774 value: 0 wei
Debug
[block:8894005 txIndex:32] from: 0x068...13fF3 to: 0xD40...ED5c1 value: 0 wei
Debug
[block:8894005 txIndex:24] from: 0xA76...7334f to: 0x665...23ab0 value: 0 wei
Debug
[block:8894005 txIndex:34] from: 0xf40...650E8 to: 0x7a2...2488D value: 439423999733169455 wei
Debug



合約運行環境
e.g. testrpc, testnet, mainnet

交易的 gas 限制

欲部署的合約

已部署的合約

Ethereum smart contract

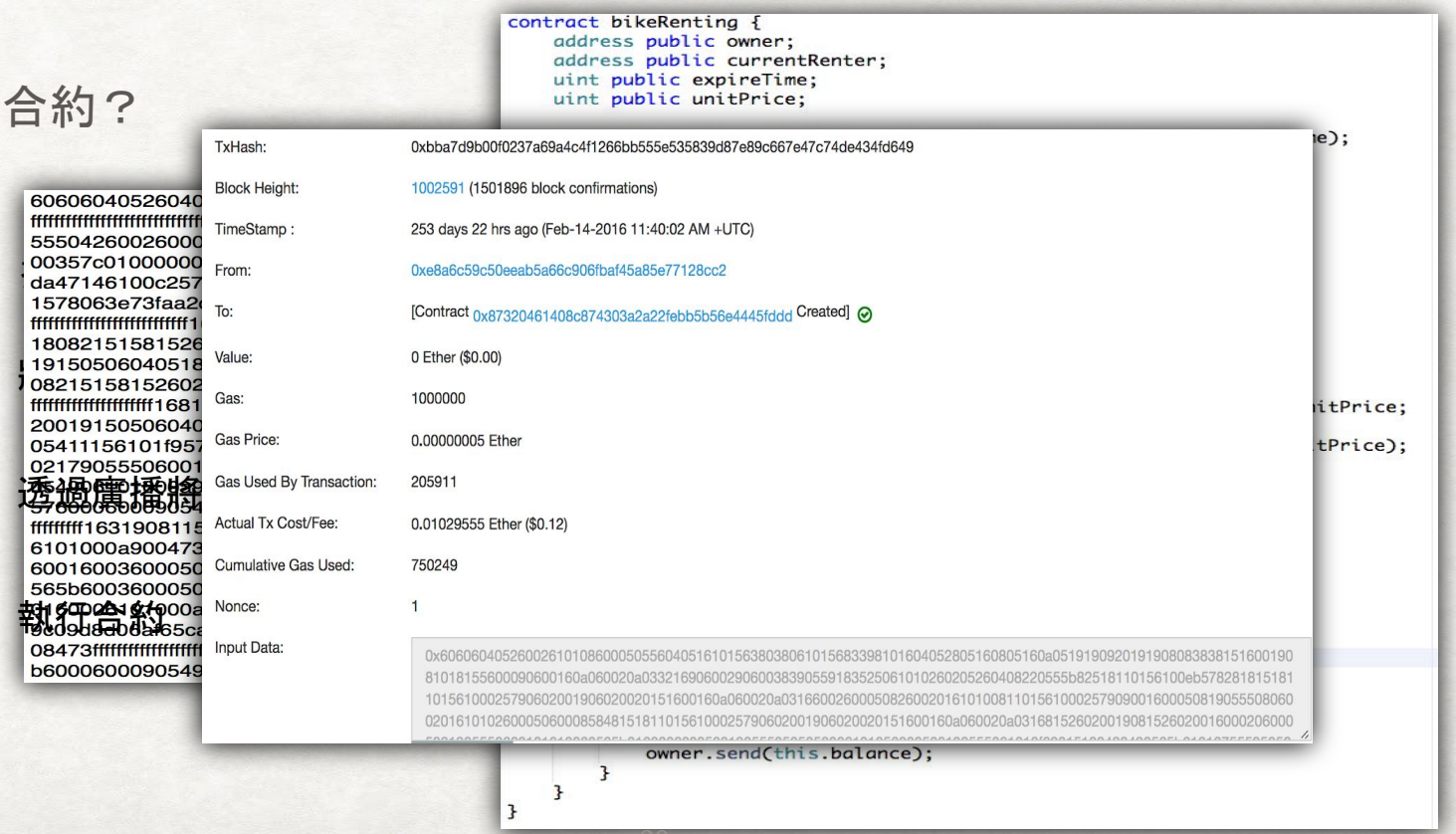
如何建立合約？

1.

2.

3.

4.



快速介紹

- 貨幣單位：ether
- address：代表一個account，可提錢存錢。分為兩種：
  - 使用者
  - 合約：
    - 會有一份code和這個address綁在一起。
    - 同一份code在不同address上就是不同合約。
- transaction：
  - 可為單純的送錢
  - 或用來執行合約（同時也可附錢）

快速介紹

- 合約：分為狀態（state）和動作（function）
  - state：可永久保留，使用者用來記錄合約的相關資訊
  - function：對這合約狀態 產生影響的動作
  - 註：Ethereum的設計機制是不鼓勵儲存 state的，用到state耗費的gas特別高。
    - 因為每個節點都要儲存一份一樣的資料，如果大家都用到大量的儲存，會造成所維護的鏈非常笨重。
- gas
  - 每一個函式都是由許多單元運算所組成，每個運算都有固定的算力消耗，因此每執行一次函式都要付出相對應該付的費用，而這個費用是用 ether來付。
  - 當函式寫好，gas耗費就已經固定，唯一可變的是每單位 gas所價值的ether。你每單位gas給越多ether，礦工更願意收入你的 transaction，也就更快被放進鏈裡。

快速介紹

- 如果是公鏈，每分錢每分貨，每個狀態都要花成本，所以會斤斤計較
- 如果使用的是私鏈呢？
    - gas好像就沒有這麼重要了？
        - 因為要有多少ether有多少ether
    - 但是狀態的儲存就可以想存多少就存多少嗎？
        - 沒錯，但同樣的，最後儲存的成本還是會轉嫁到每個節點身上。

實戰演練 Solidity

ERC20 Token (Fungible)

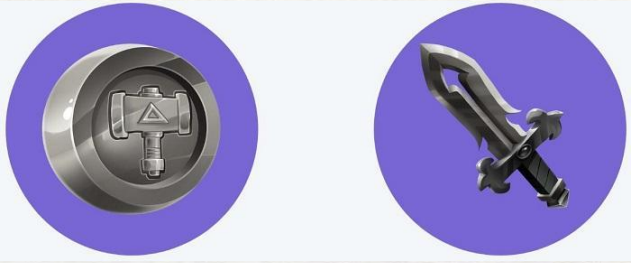
- Ethereum Request for Comments 20
- ERC20 是目前在以太坊上的 Token 最主流的規格標準, 相對安全性較高
- 所有 ERC20 代幣可以利用 Ethereum 生態內的工具做交易、追蹤, 代幣發行商不必額外開發工具

ERC721 Token (Non-Fungible)

- ERC721 是 2017-2019 在以太坊上的 Token 最主流的規格標準
- 所有 ERC721 代幣可以利用 Ethereum 生態內的工具做交易、追蹤，代幣發行商不必額外開發工具
- 可以將影像加入 ERC721 的合約當中，當作卡牌的圖片
- 每個 ERC721 Token 就等於一個限量卡牌包

ERC1155 Token (Non-Fungible)

- ERC1155 是 2019 以來在以太坊上的 Token 最主流的 NFT 規格標準
- 智能合約可以在一次傳輸多種類型的代幣, 節省了80-90%的GAS費用。
- 多種類型間的代幣交易, 也可以直接基於此標準交易, 不需要先核准 (approve) 不同的獨立合約才能互動。
- 人們在單一智能合約內, 進行同質性代幣與非同質性代幣間的互動(Batch Operation)



安裝錢包

安裝 MetaMask 瀏覽器插件

1. Chrome 線上商店搜尋 MetaMask
2. 加到 Chrome

創建帳戶

選擇創建錢包

1. 輸入自定義密碼
2. 備份助憶詞，放在一個沒有人看得見的地方
3. 千萬不要像這樣分享助憶詞

monkey galaxy catalog age across provide

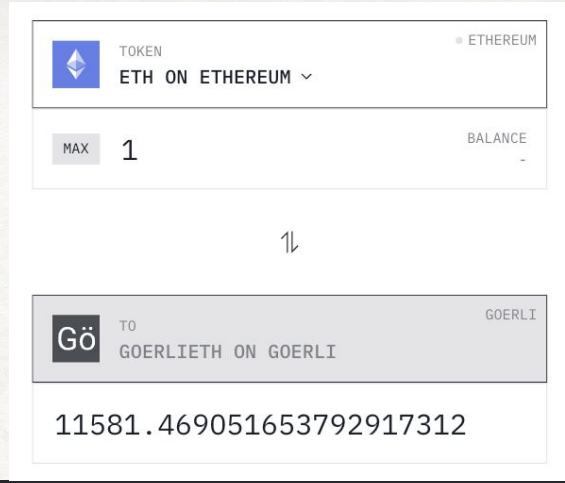
burden sweet track unable wheat expect

開啟 Metamask 測試網路顯示

Metamsak 設定

1. 點選以太坊主網路按鈕
2. 進階
3. Show test networks 開啟

選擇測試網路

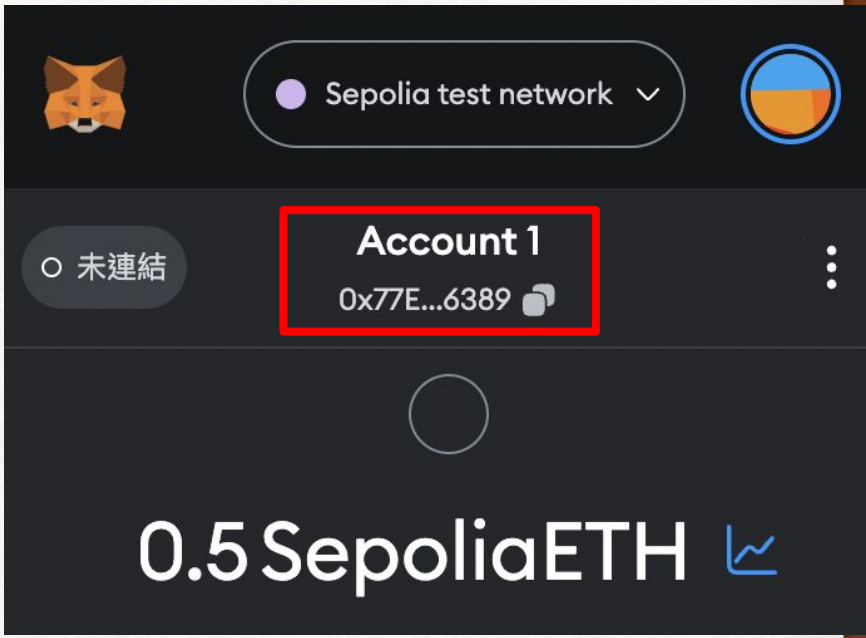


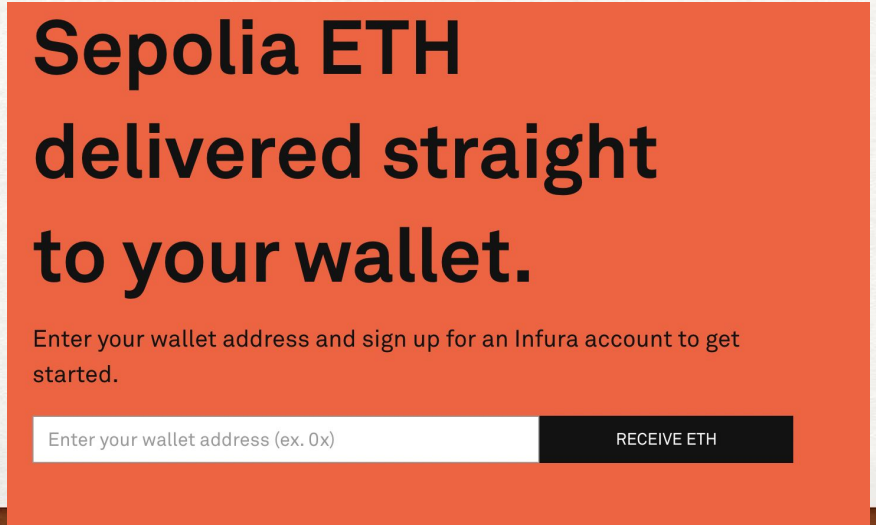
- Goerli (LayerZero 把測試幣變真錢了)
  - ~~囤積測試幣的項目開發者已爆富~~
  - PoS (Proof of Stake)
  - deprecated
- Sepolia
  - PoS (Proof of Stake)
  - Infura 水龍頭可以領幣
- Linea Goerli
  - PoS
  - zkEVM

從水龍頭索取測試以太坊

https://www.infura.io/faucet/sepolia

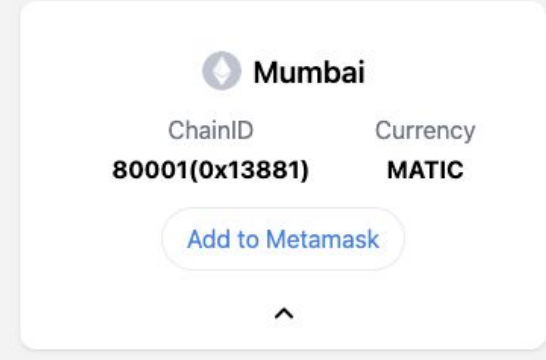
1. 註冊 infura
2. 輸入 sepolia ethereum 地址

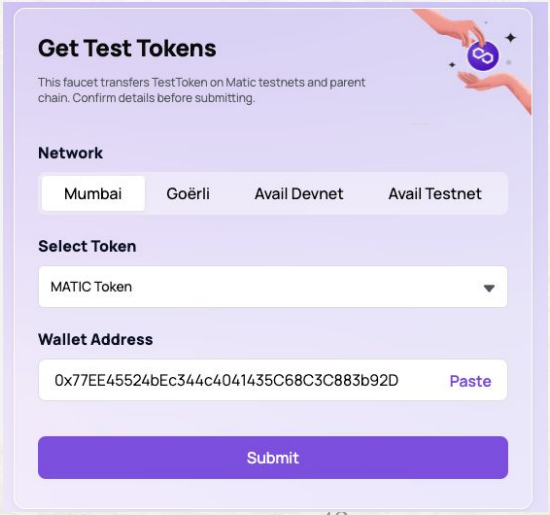


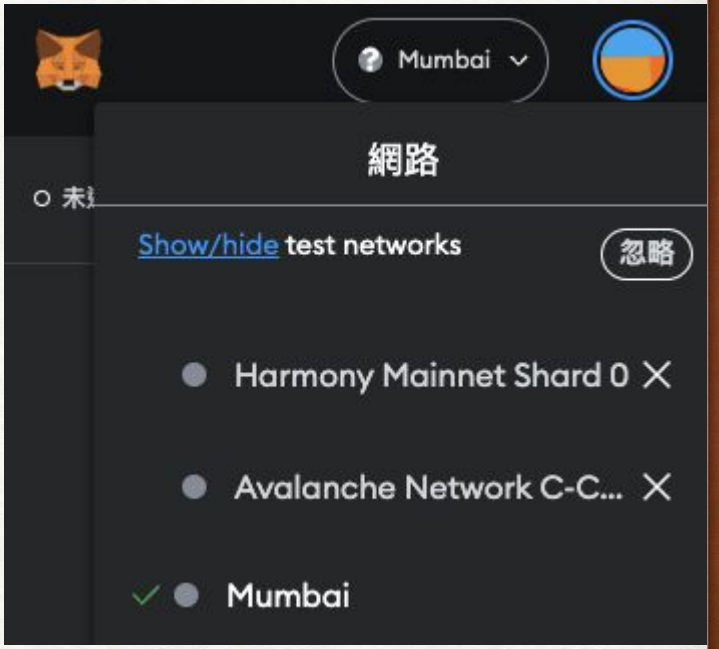


選擇測試網路

- Polygon Munbai
  - PoS (Proof of Stake)
  - Chainlist 搜尋 Polygon 加入 Munbai RPC
  - 官網水龍頭可以領幣







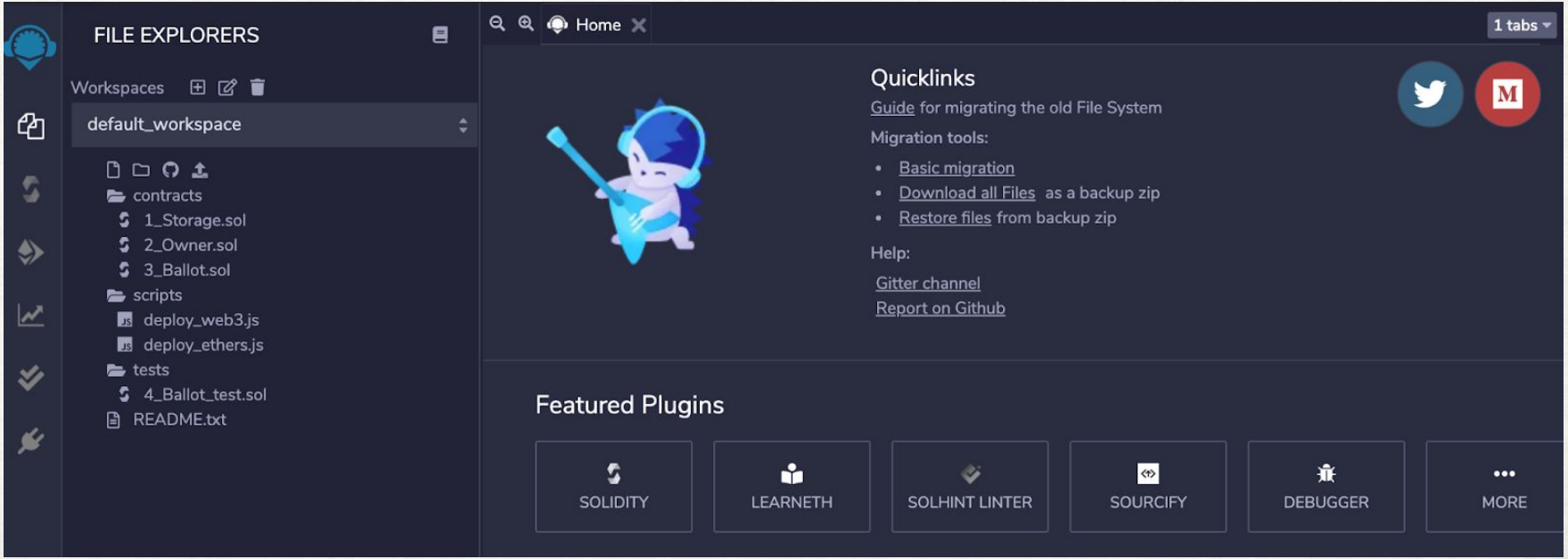
Polygon Testnet

- Network: Polygon Mumbai Testnet
  - RPC URL (public endpoint): https://rpc-mumbai.maticvigil.com
  - RPC URL (dedicated endpoint): https://polygon-mumbai.g.alchemy.com/v2/your-api-key. You'll need a free API key
  - Chain ID: 80001
  - Currency Symbol: MATIC
  - Block Explorer URL: https://mumbai.polygonscan.com/

Source: Deploy a Smart Contract Using Remix

進入 Remix 線上編譯器

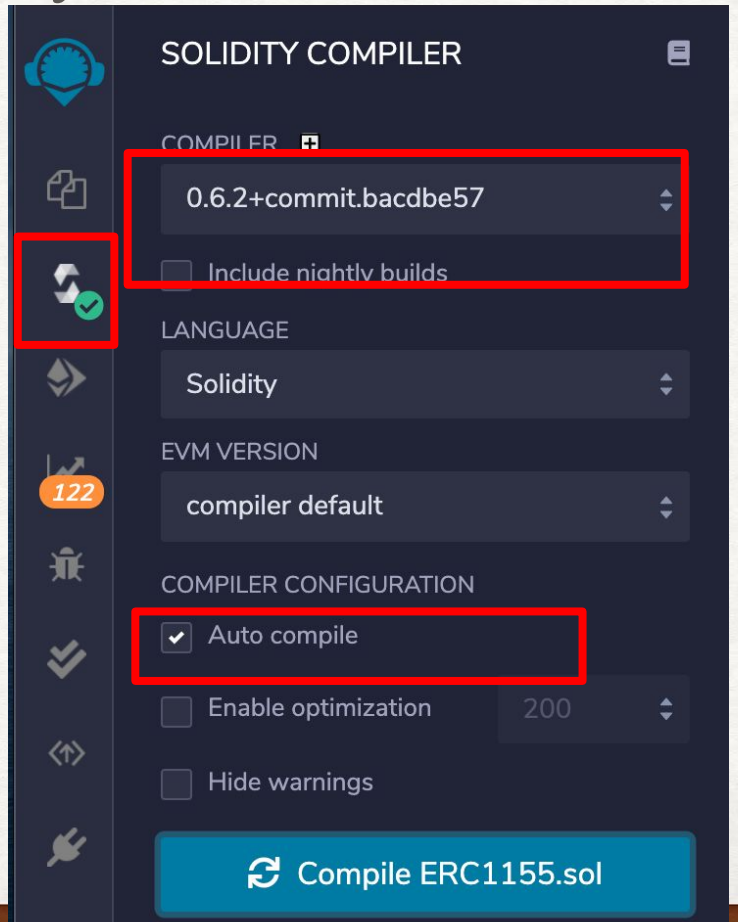
https://remix.ethereum.org/



設置 Remix Solidity 編譯環境

https://remix.ethereum.org/

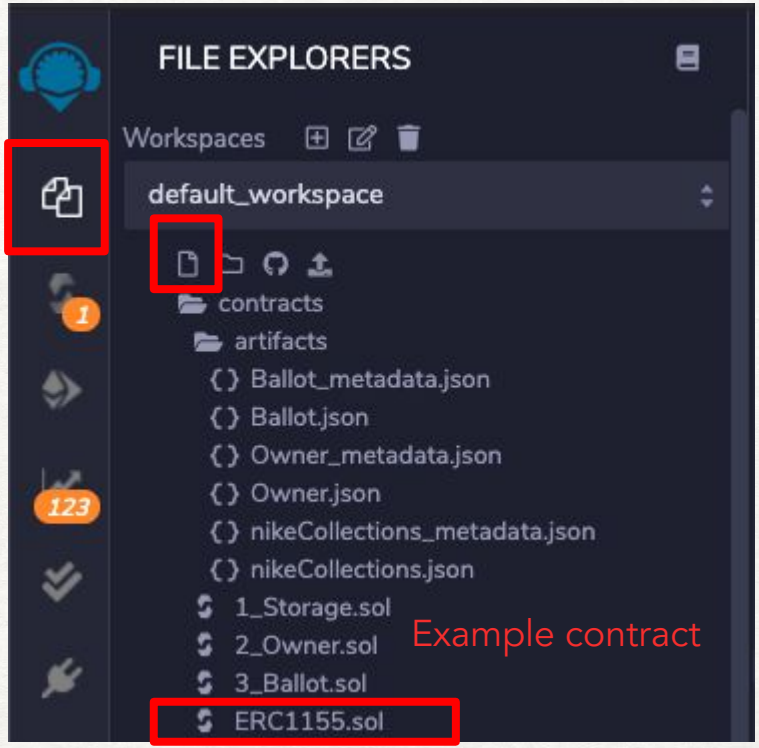
1. Choose Solidity Compiler
2. Choose Compiler version
3. Auto compile



撰寫合約 - 新增 ERC 1155 合約

https://remix.ethereum.org/

1. Select File Explorers
2. Add new file
3. Rename it as ERC1155.sol



撰寫 ERC 1155 合約

```
1.  // contracts/GameItems.sol
2.  // SPDX-License-Identifier: MIT
3.  pragma solidity ^0.6.2;
4.
5.  import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v3.2.0/contracts/token/ERC1155/ERC1155.sol";
6.
7.  contract GameItems is ERC1155 {
8.      uint256 public constant GOLD = 0;
9.      uint256 public constant SILVER = 1;
10.     uint256 public constant THORS_HAMMER = 2;
11.     uint256 public constant SWORD = 3;
12.     uint256 public constant SHIELD = 4;
13.
14.     constructor() public ERC1155("https://game.example/api/item/{id}.json") {
15.         _mint(msg.sender, GOLD, 10**18, "");
16.         _mint(msg.sender, SILVER, 10**27, "");
17.         _mint(msg.sender, THORS_HAMMER, 1, "");
18.         _mint(msg.sender, SWORD, 10**9, "");
19.         _mint(msg.sender, SHIELD, 10**9, "");
20.     }
21. }
```

Gold is fungible token

Thors Hammer is non-fungible token. Only one

https://docs.openzeppelin.com/contracts/3.x/erc1155

撰寫 ERC 1155 合約

```
// contracts/GameItems.sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.6.0;

import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v3.2.0/contracts/token/ERC1155/ERC1155.sol";

contract GameItems is ERC1155 {
    uint256 public constant GOLD = 0;
    uint256 public constant SILVER = 1;
    uint256 public constant THORS_HAMMER = 2;
    uint256 public constant SWORD = 3;
    uint256 public constant SHIELD = 4;

    constructor() public ERC1155("https://abcoathup.github.io/SampleERC1155/api/token/{id}.json") {
        _mint(msg.sender, GOLD, 10**18, "");
        _mint(msg.sender, SILVER, 10**27, "");
        _mint(msg.sender, THORS_HAMMER, 1, "");
        _mint(msg.sender, SWORD, 10**9, "");
        _mint(msg.sender, SHIELD, 10**9, "");
    }
}
```

Gold is fungible token

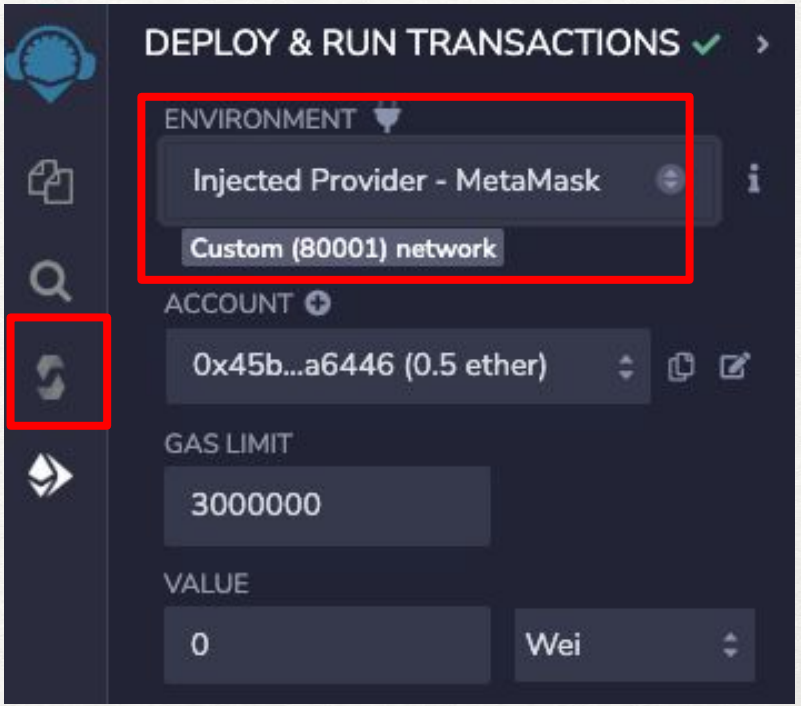
Thors Hammer is non-fungible token. Only one

https://forum.openzeppelin.com/t/create-an-erc1155/4433

連結 MetaMask Polygon Mumbia 錢包

https://remix.ethereum.org/

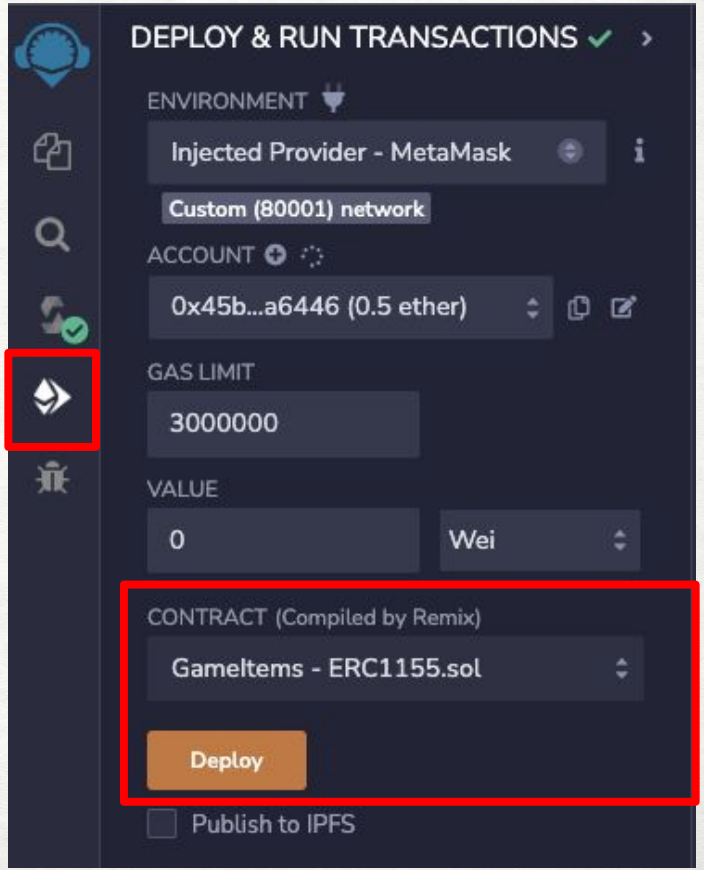
1. Choose Depoly & Run Transaction
2. Metamask choose mumbai testnet
3. Injected Provider
4. Link to your MetaMask Wallet



部署 ERC 1155 合約

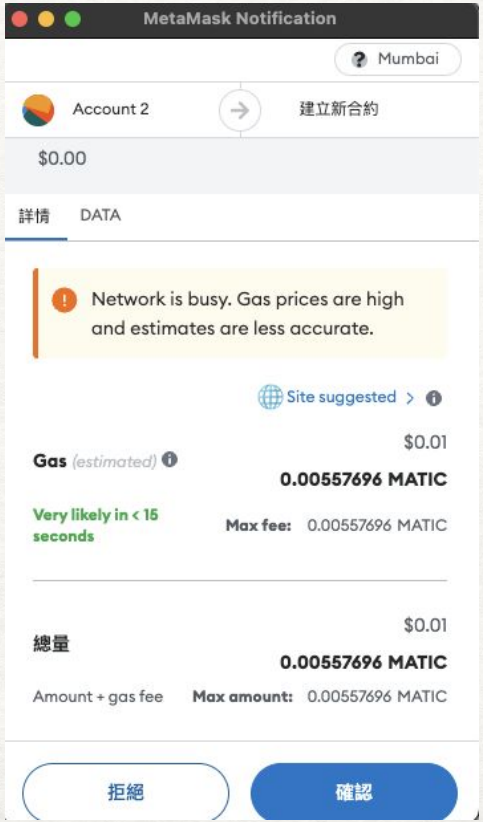
https://remix.ethereum.org/

1. Choose Depoly & Run Transaction
2. Check Contract
3. Deploy your ERC1155.sol

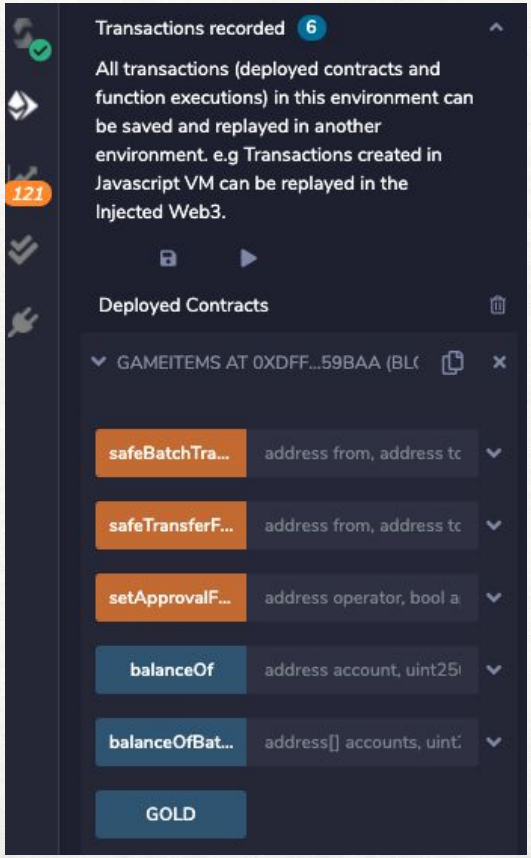


連結 MetaMask 部署合約

1. Confirm MteaMask Network
2. Check yout ETH
3. Confirm Signature



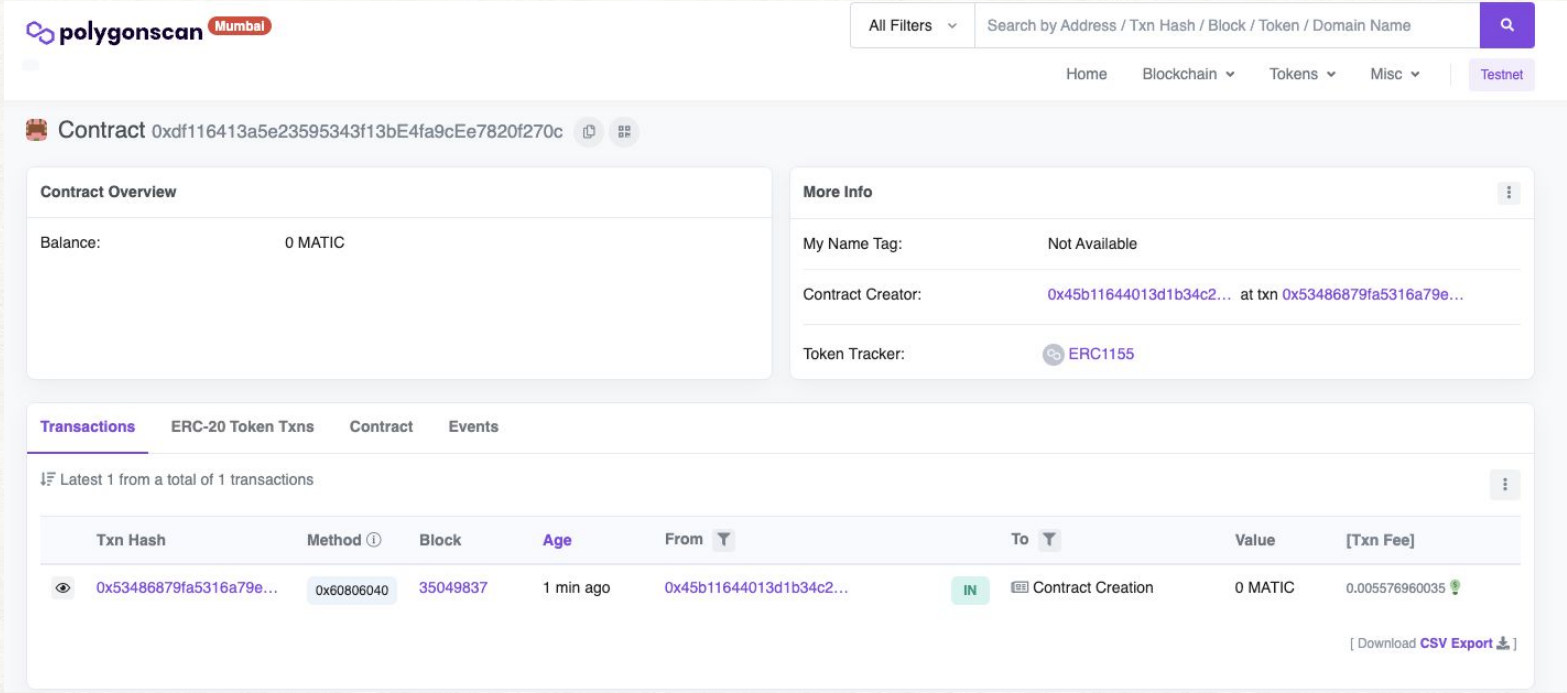
ERC1155 合約部署完成



Console

```
creation of GameItems pending...
```

確認鏈上資訊



https://mumbai.polygonscan.com/address/0xdf116413a5e23595343f13bE4fa9cEe7820f270c

確認 ERC1155 合約寶物數量

balanceOf

account: "0x641dd94Ae964007AB

id: "2"

call

0: uint256: 1

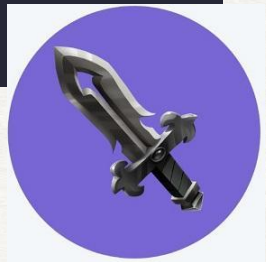
balanceOf

account: "0x641dd94Ae964007AB

id: "3"

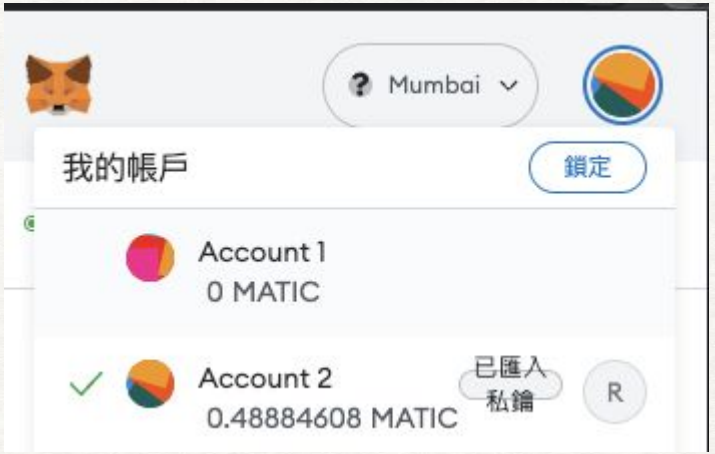
call

0: uint256: 1000000000



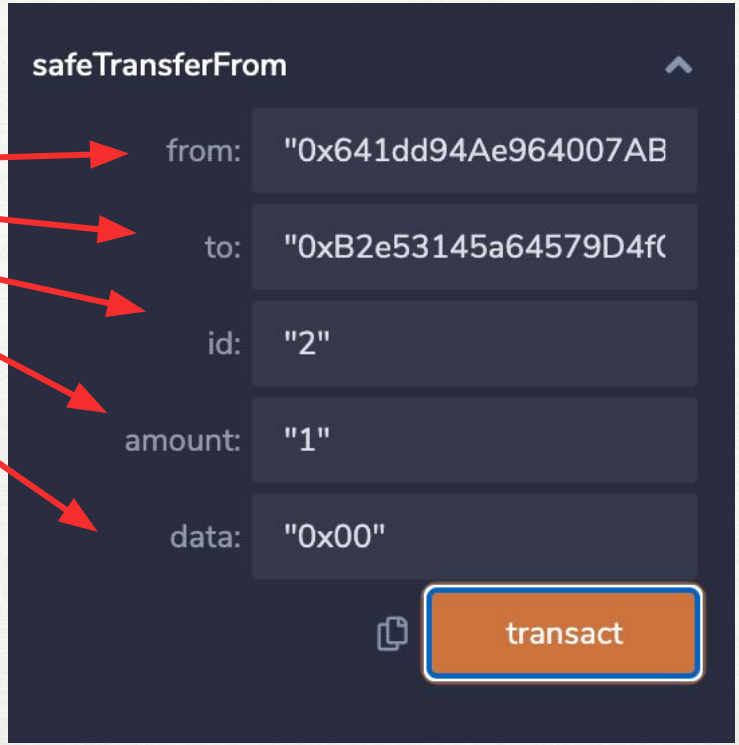
增加 MetaMask 帳戶

1. 建立帳戶
2. 取得帳戶 2
3. 切換回帳戶 1



傳送索爾的錘子

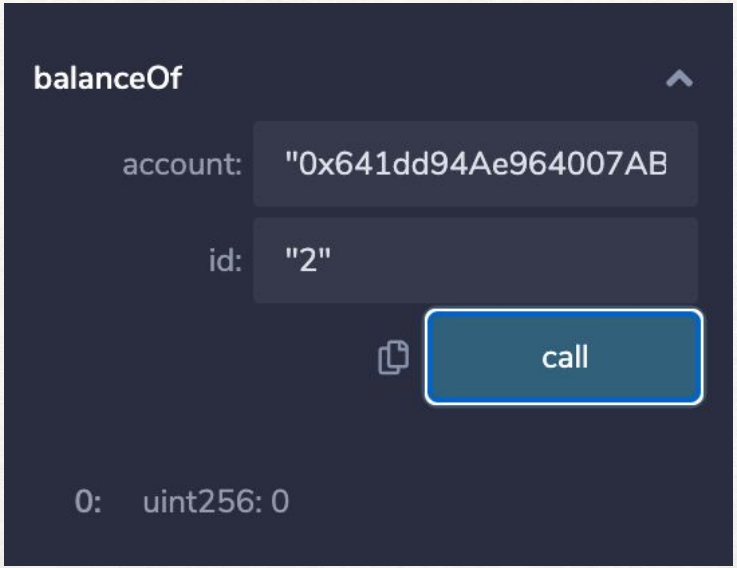
帳戶 1　合約擁有者
帳戶 2　新玩家
ID 2　錘子
數量　1 (只有一個)
Data　“0x00”

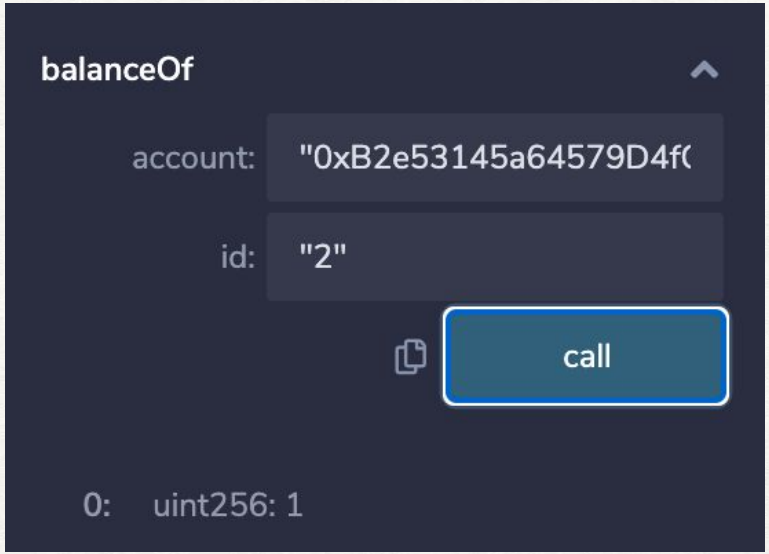


檢查新玩家有沒有收到錘子

帳戶 1　　合約擁有者 (部署帳號)
0x45b11644013D1b34C2ba8da474b605A6C66a6446

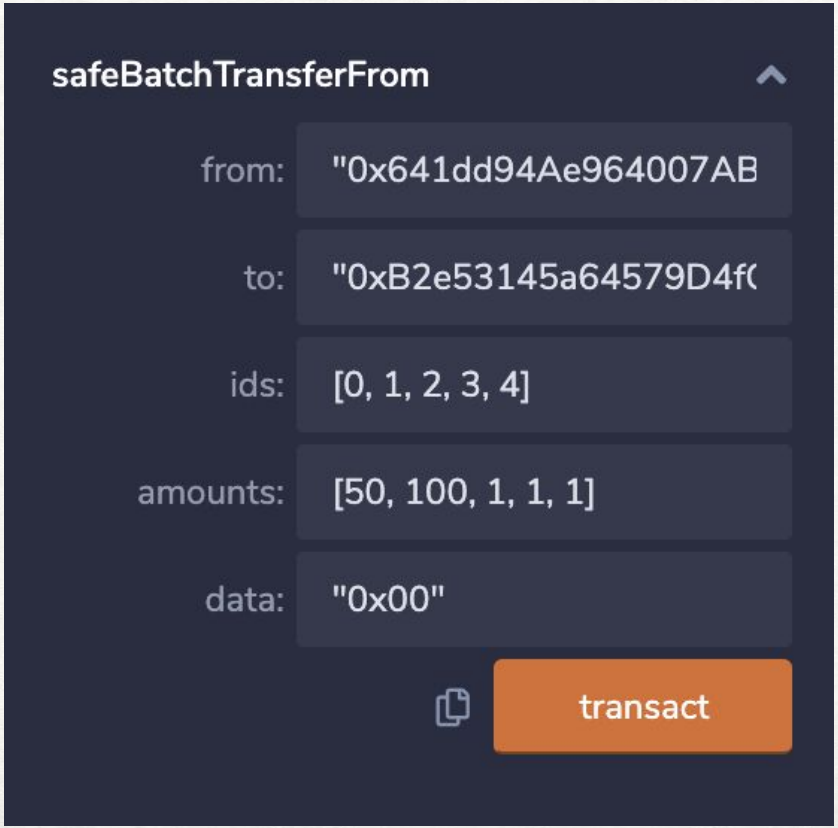
帳戶 2　　新玩家
0x641dd94Ae964007AB86061933D411550b0d0711a

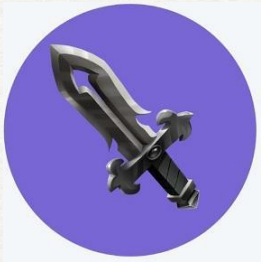




批次傳送寶物

帳戶 1	合約擁有者
帳戶 2	新玩家
多ID	[0, 1, 2, 3, 4]
數量	[50, 100, 1, 1, 1]
Data	“0x00”

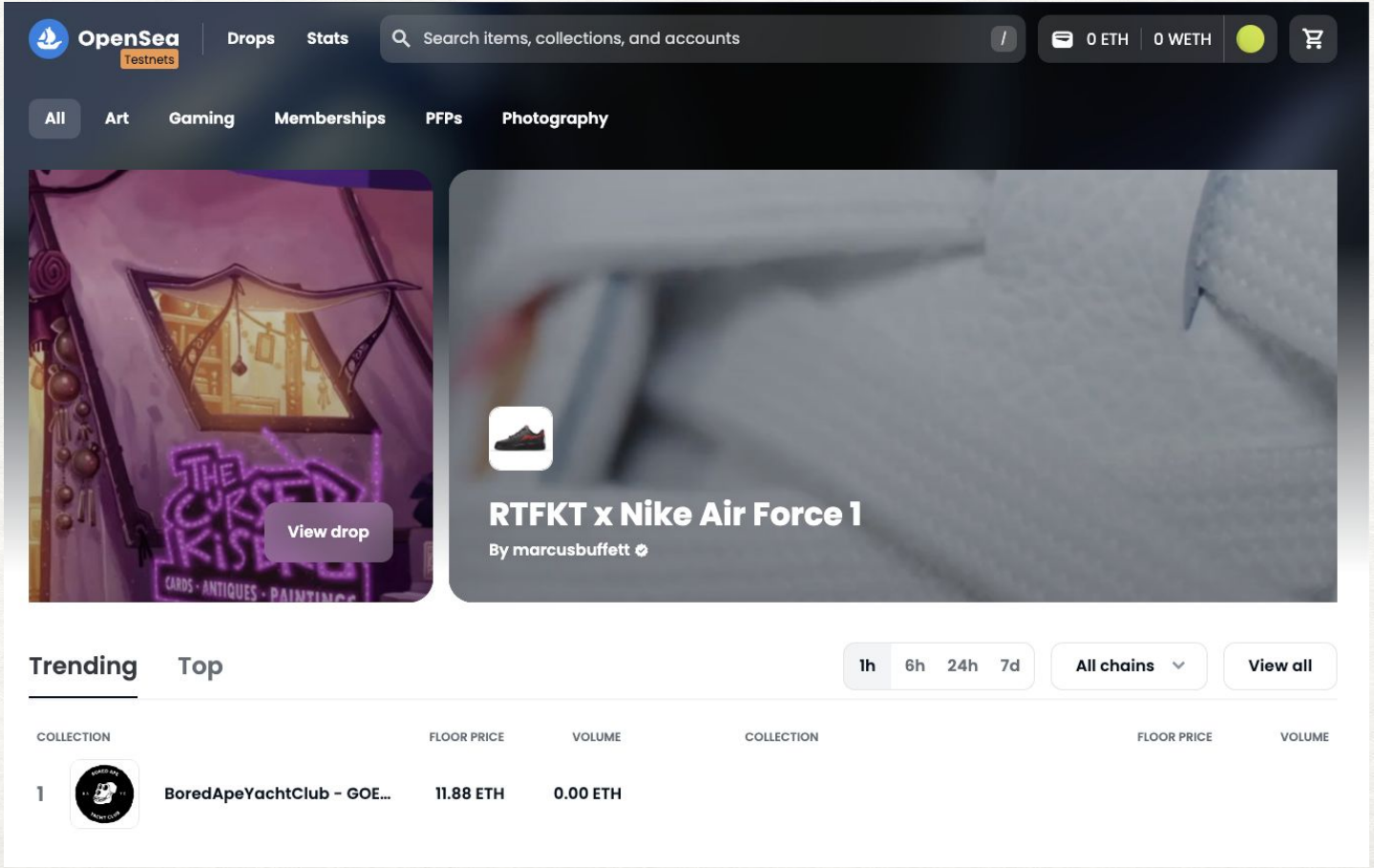




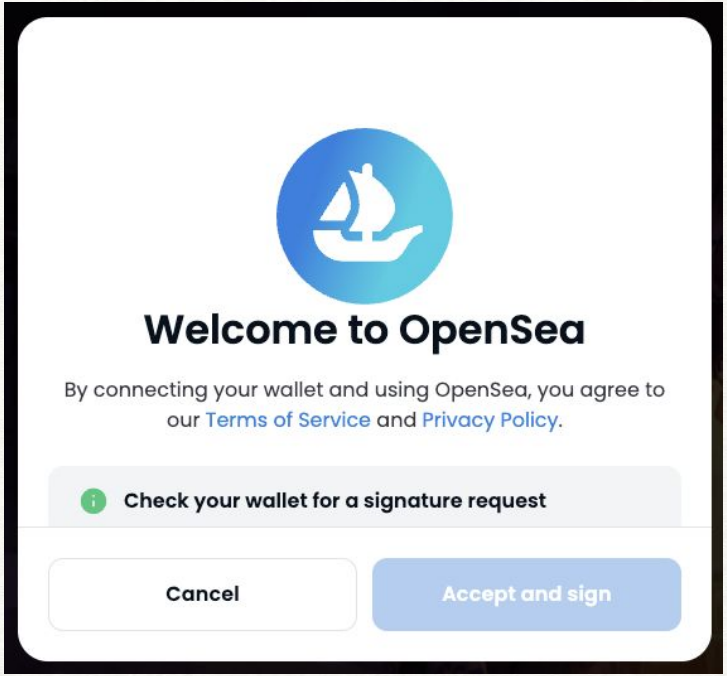
恭喜你已經學會如何製作 NFT 寶物了

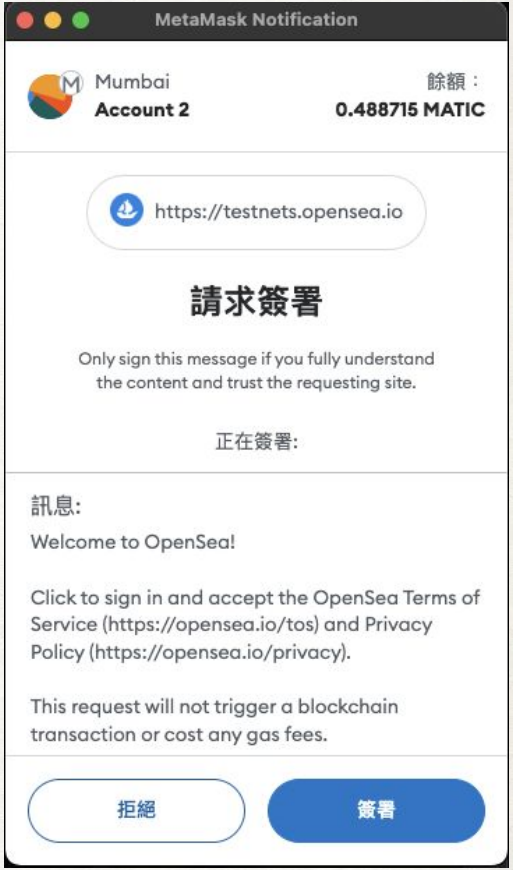
實戰演練 2
上架 OpenSea

OpenSea Testnet



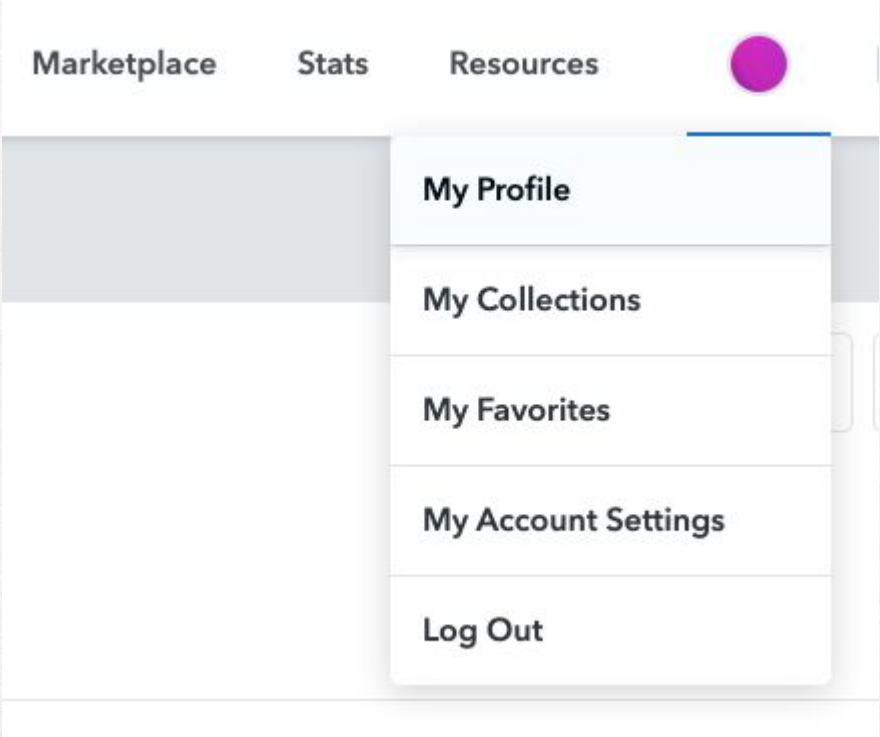
MetaMask 簽署 OpenSea



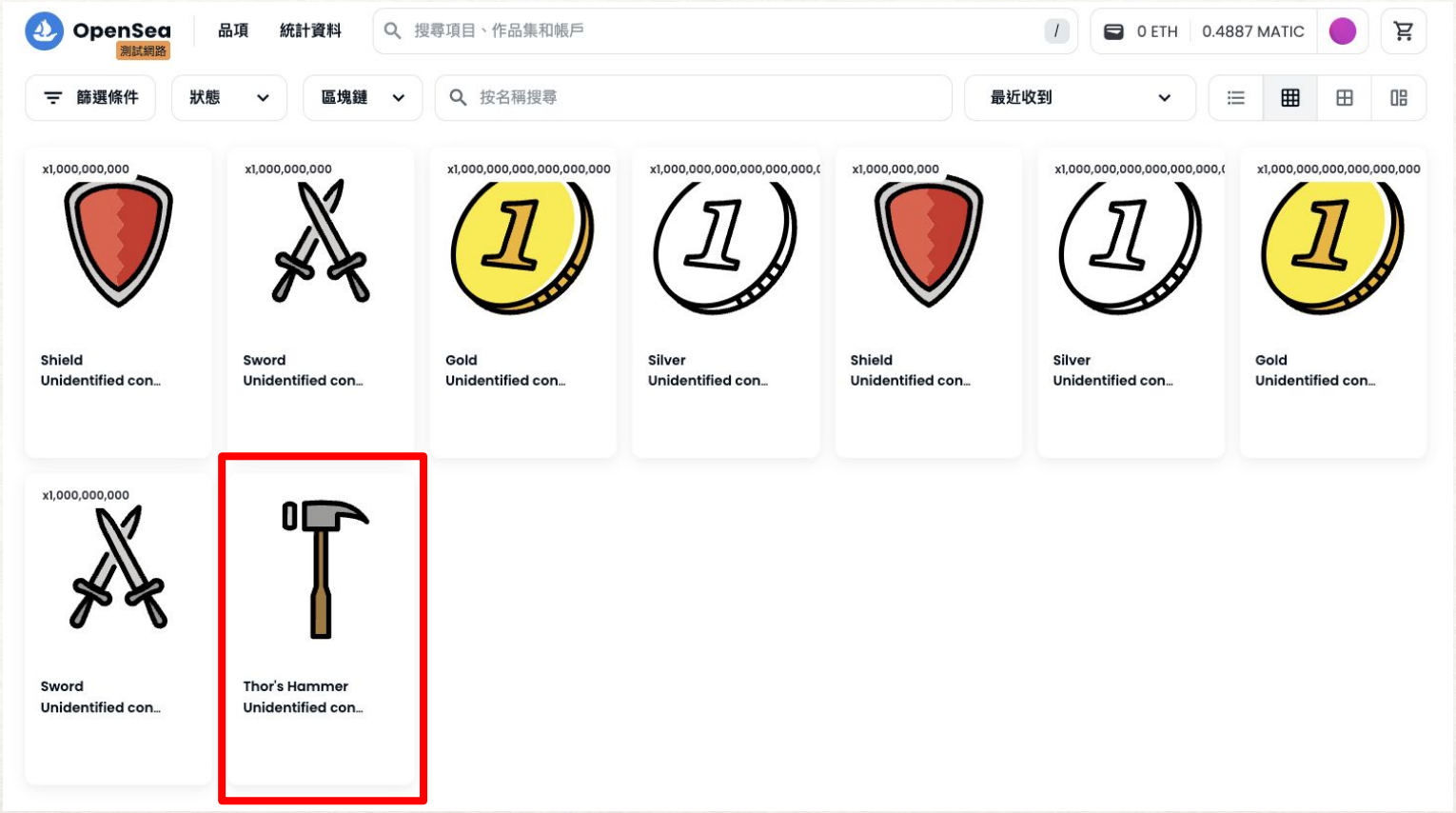


https://testnets.opensea.io/

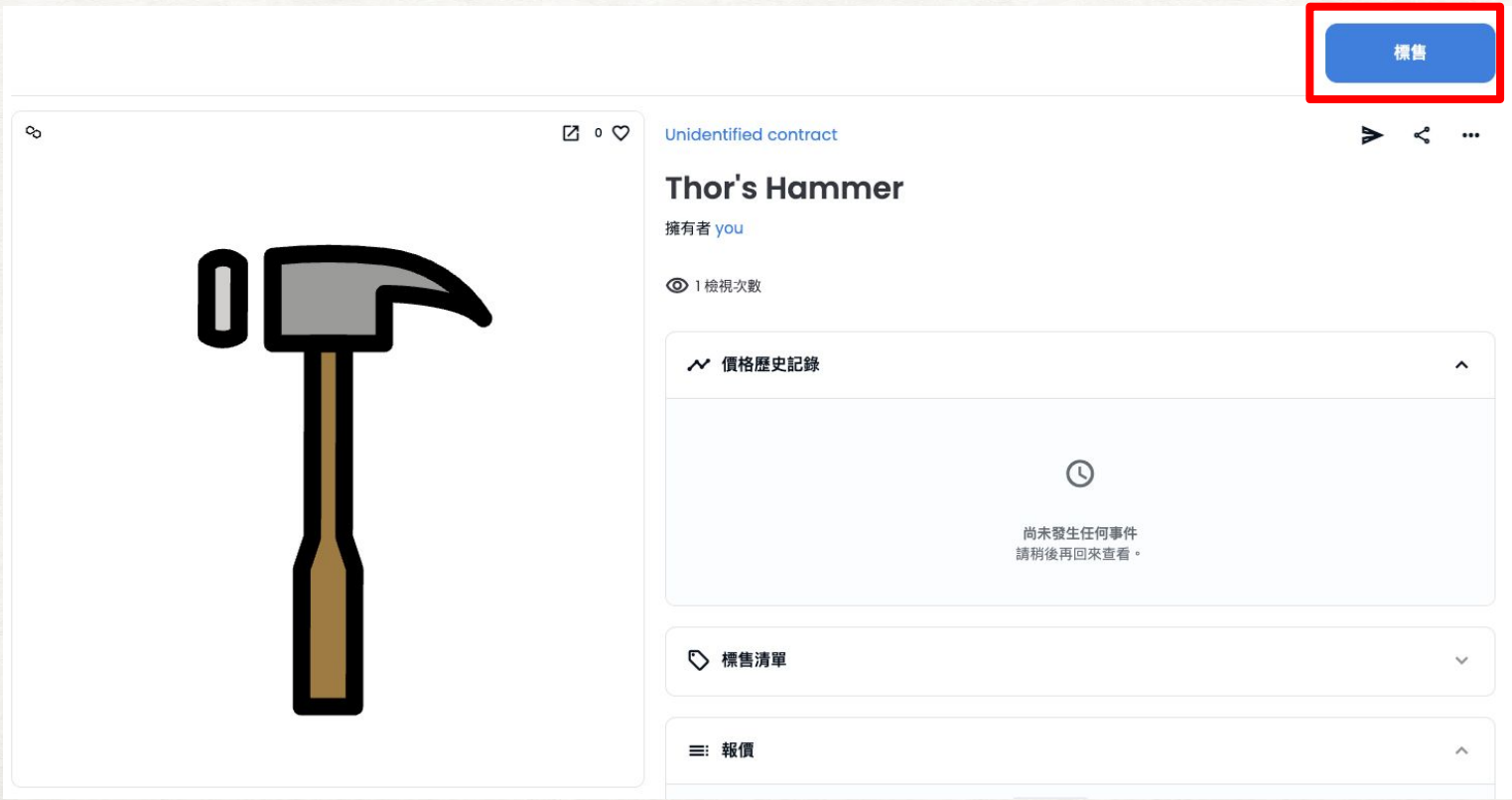
檢查 My Profile



檢查索爾的錘子數量與 ID

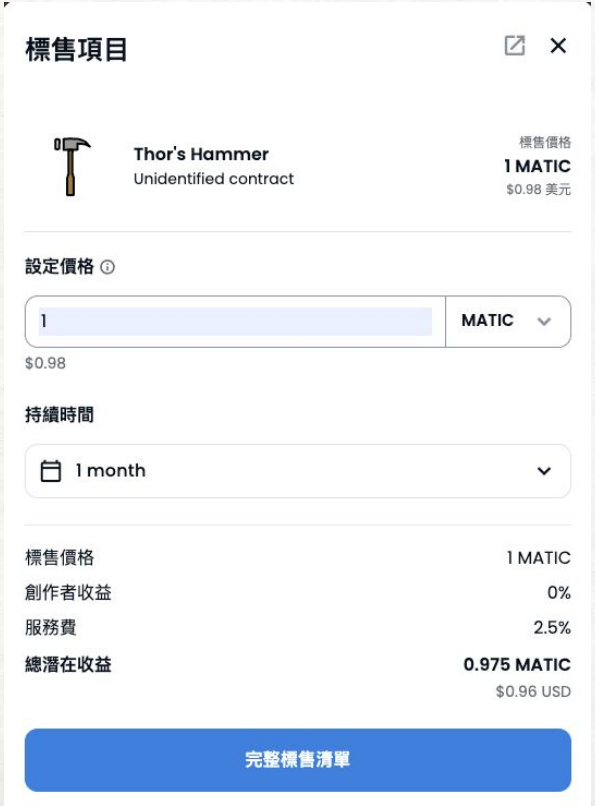


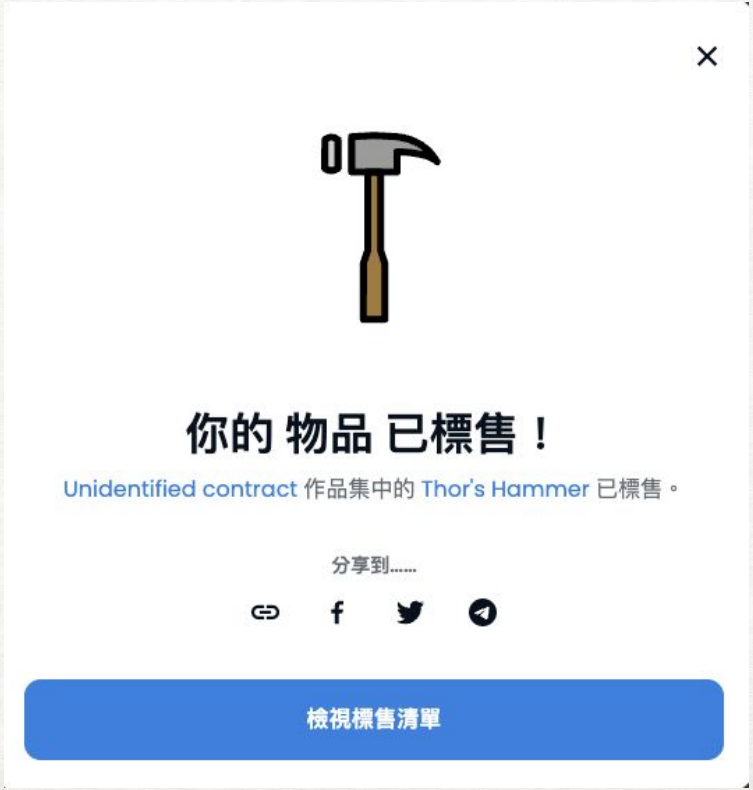
將索爾的錘子上架到 OpenSea



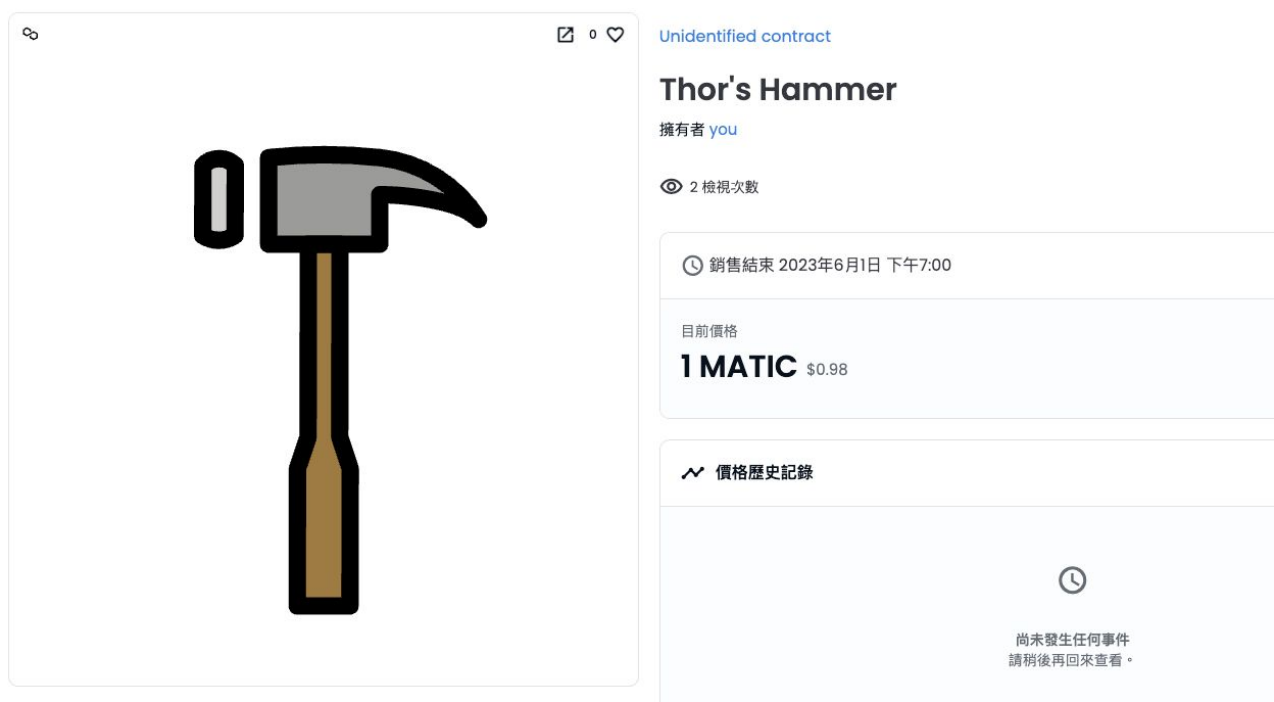
https://testnets.opensea.io/zh-TW/assets/mumbai/0xdf116413a5e23595343f13be4fa9cee7820f270c/2

設定價格後 Post your listing



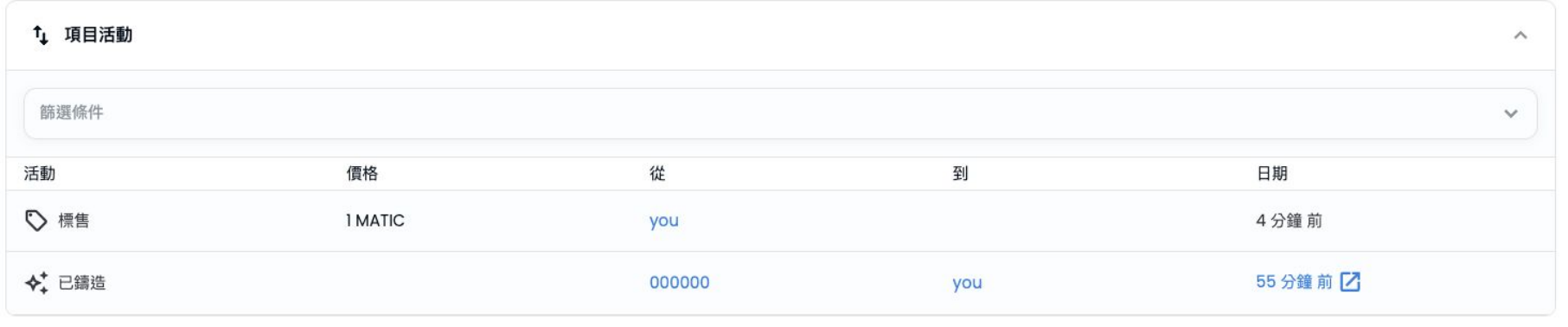


檢查索爾的錘子拍賣頁面



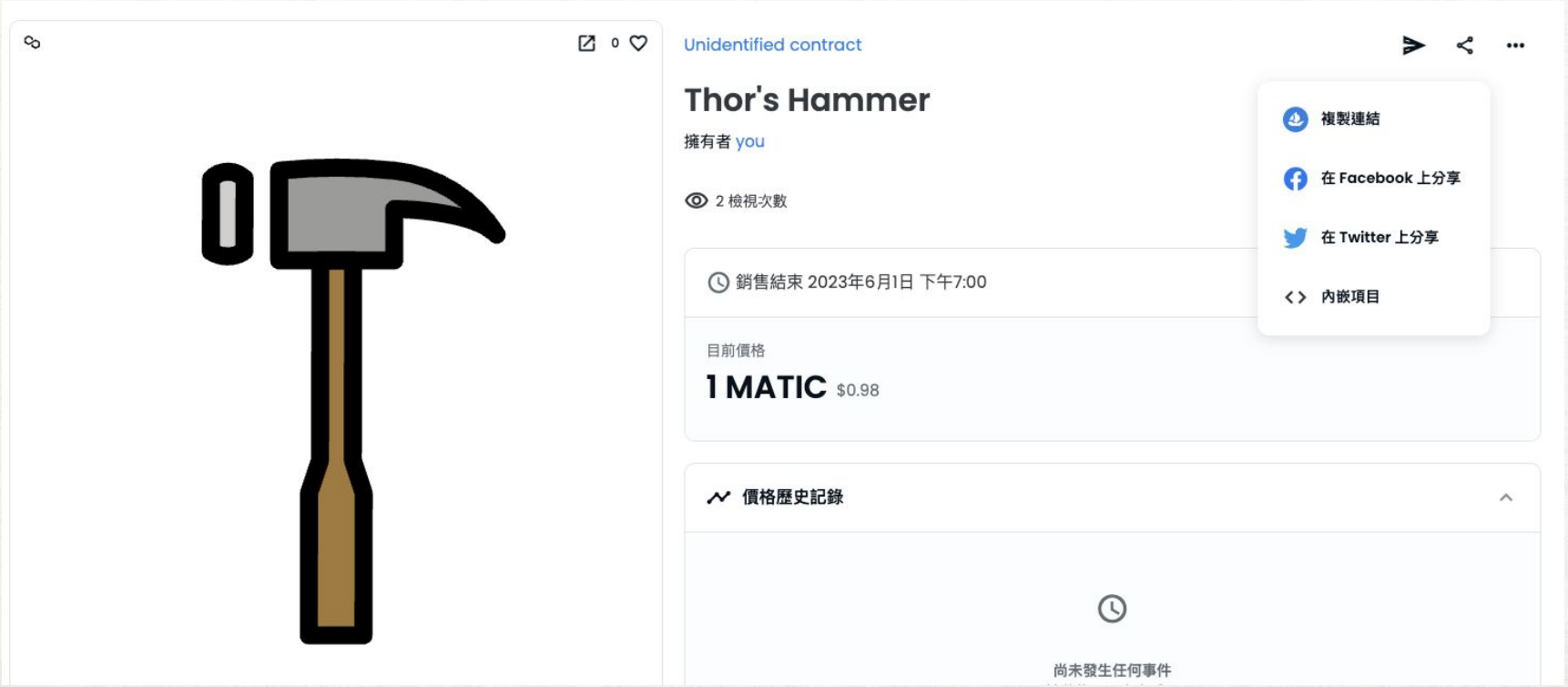
https://testnets.opensea.io/zh-TW/assets/mumbai/0xdf116413a5e23595343f13be4fa9cee7820f270c/2

檢查索爾的錘子拍賣紀錄



https://testnets.opensea.io/zh-TW/assets/mumbai/0xdf116413a5e23595343f13be4fa9cee7820f270c/2

分享給同學與朋友行銷 / ~~舉辦 NFT 活動~~



恭喜你已經完成了
Vitalik 的夢想

請分享寶物拍賣訊息
Twitter
並分享畫面

實戰演練 3
ERC 721 上架 Opensea

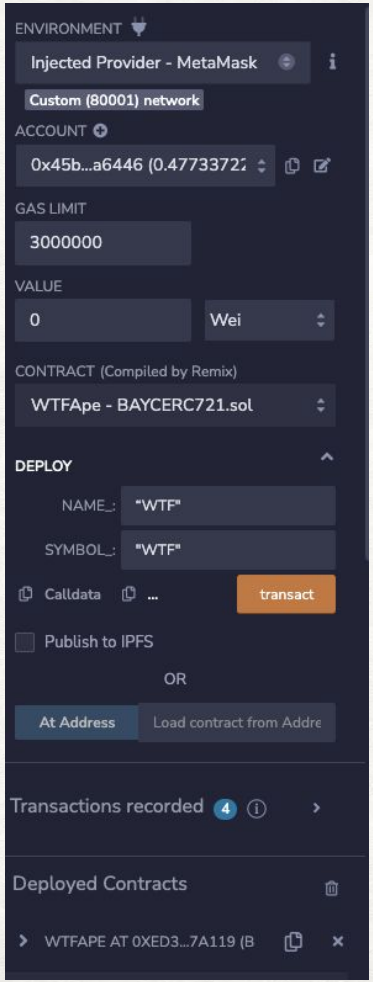
撰寫 ERC 721 合約

```
1.  // SPDX-License-Identifier: MIT
2.  pragma solidity ^0.8.4;
3.  import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
4.  contract WTFApe is ERC721{
5.      uint public MAX_APES = 10000; // 總量
6.
7.      // 構造函數
8.      constructor(string memory name_, string memory symbol_) ERC721(name_, symbol_){
9.      }
10.
11.     //BAYC的baseURI為ipfs://QmeSjSinHpPnmXmspMjwiXyN6zS4E9zccariGR3jxcaWtq/
12.     function _baseURI() internal pure override returns (string memory) {
13.         return "ipfs://QmeSjSinHpPnmXmspMjwiXyN6zS4E9zccariGR3jxcaWtq/";
14.     }
15.
16.     // 鑄造函數
17.     function mint(address to, uint tokenId) external {
18.         require(tokenId >= 0 && tokenId < MAX_APES, "tokenId out of range");
19.         _mint(to, tokenId);
20.     }
21. }
22.
```

BAYC ERC721 is only one NFT

https://forum.openzeppelin.com/t/create-an-erc1155/4433

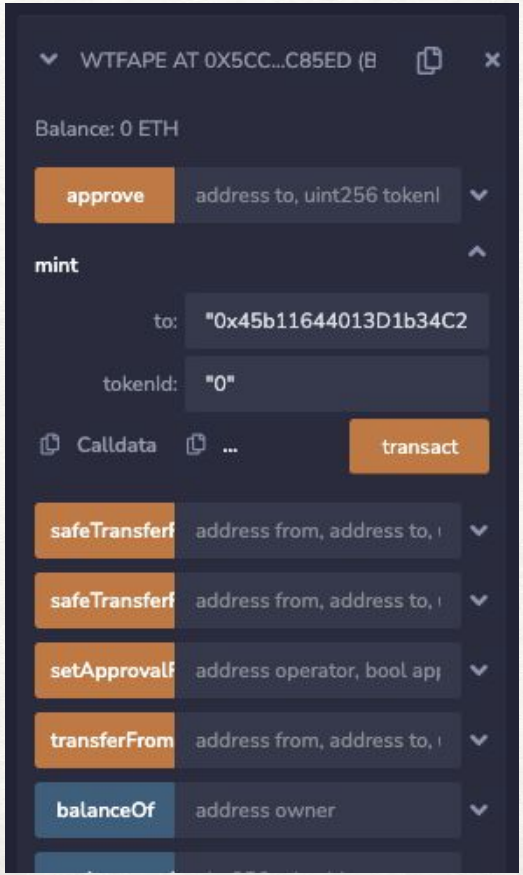
ERC721 合約部署



Console

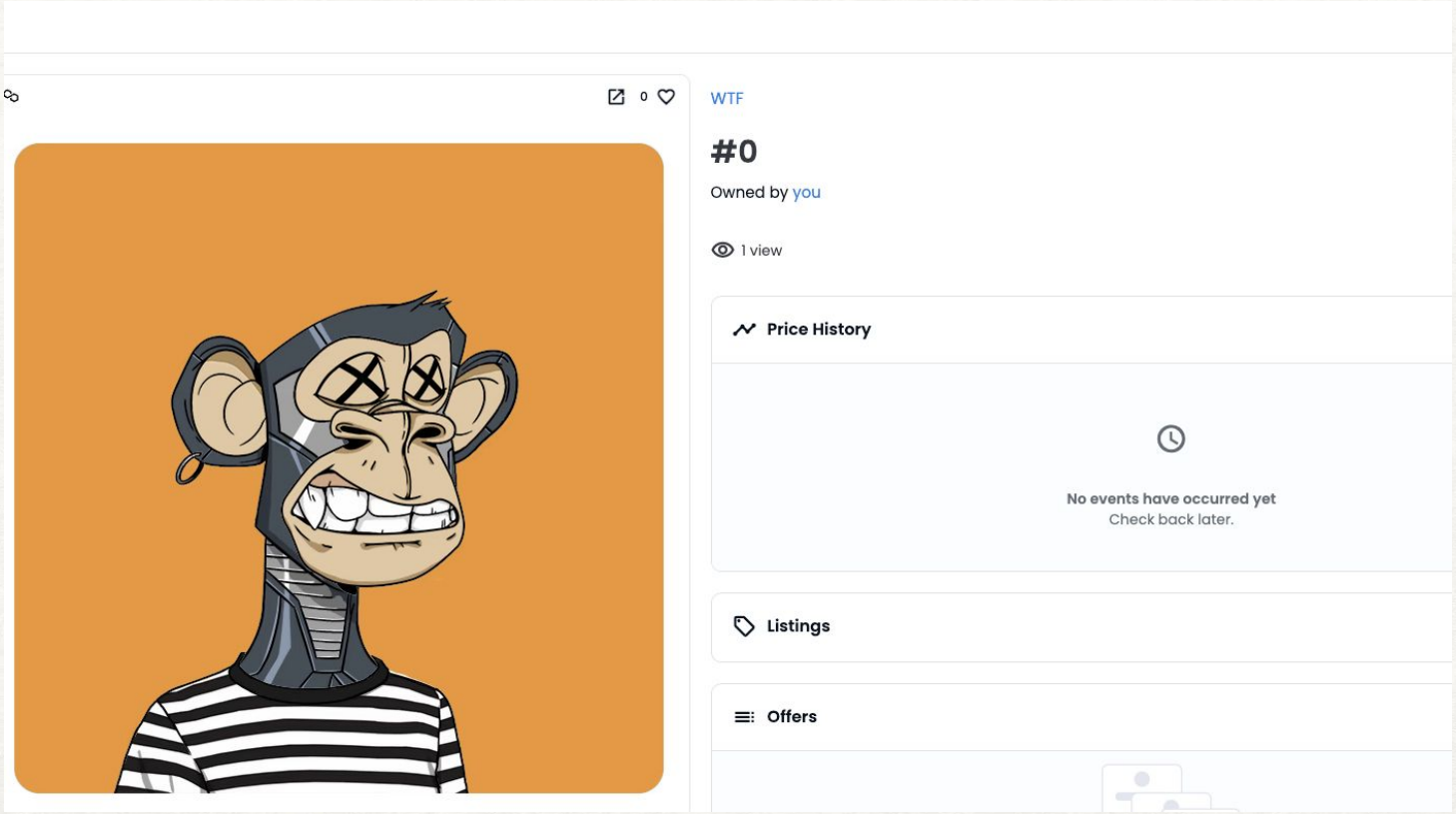
NFT 系列名字
NFT 代幣名字

Mint BAYC NFT



自己錢包地址
tokenid 設為 0

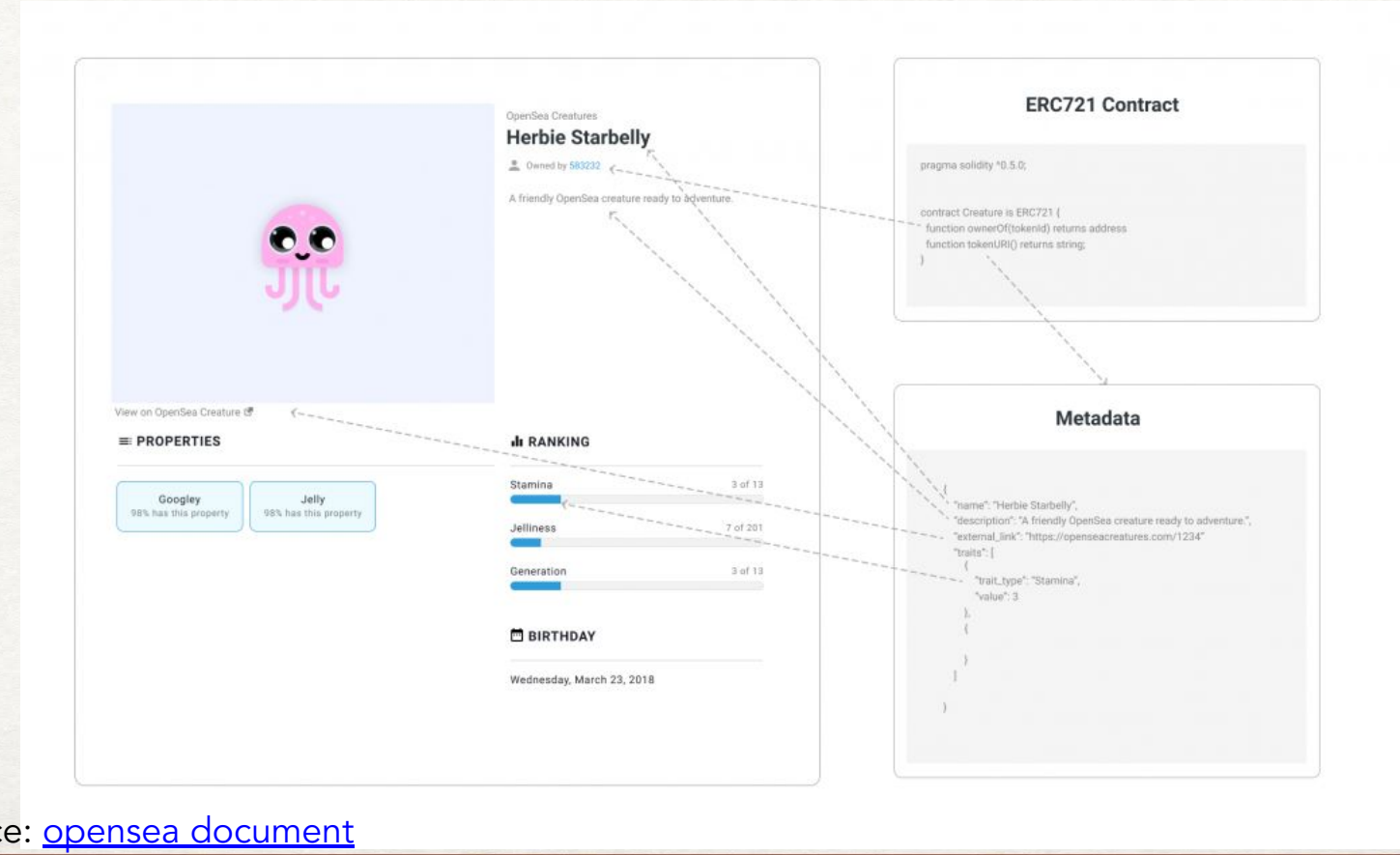
檢查 Opensea Testnet



https://testnets.opensea.io/assets/mumbai/0x5cc92a072796ff2ad14e049e7acf8f9b793c85ed/0

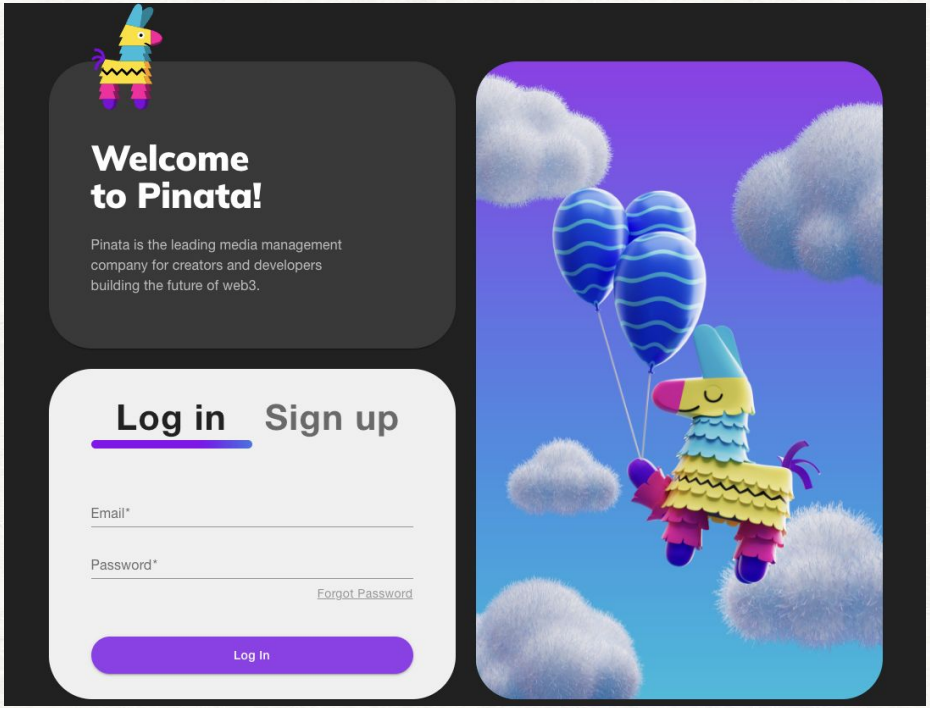
實戰演練 4
ERC 721 接上 Opensea Metadata、IPFS上傳圖片

Opensea Metadata



Source: [opensea document]

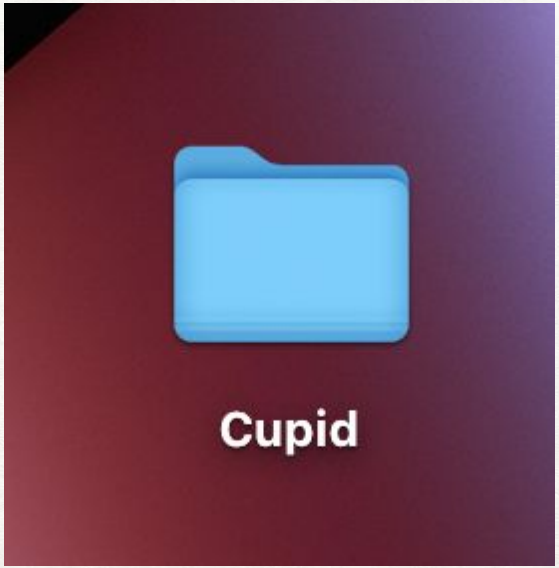
上傳圖片與 JSON 到 IPFS 取得 BaseURI



[Pinata](#) | Effortless IPFS File Management 註冊獲得 1G 容量

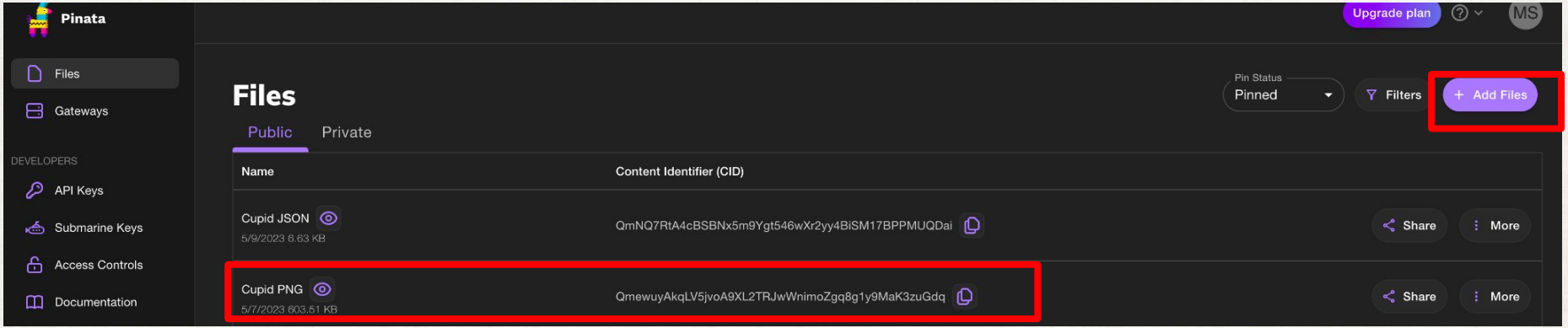
製作圖片

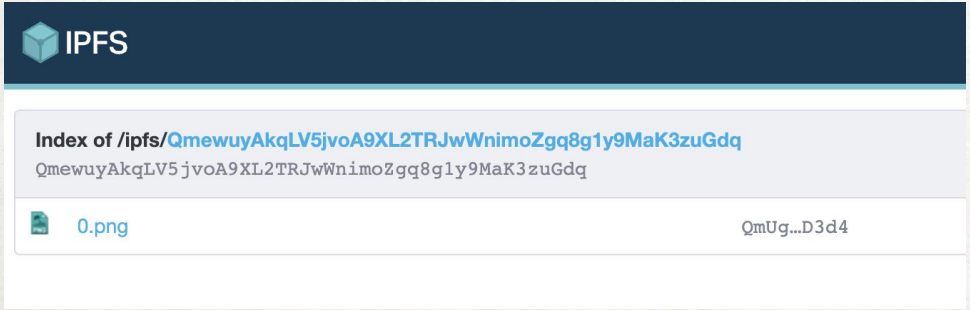
1. 新增資料夾
2. 命名圖片為 0.png



上傳圖片資料夾到 IPFS 取得 BaseURI

上傳資料夾





圖片資料夾 CID (Content IDentifier)

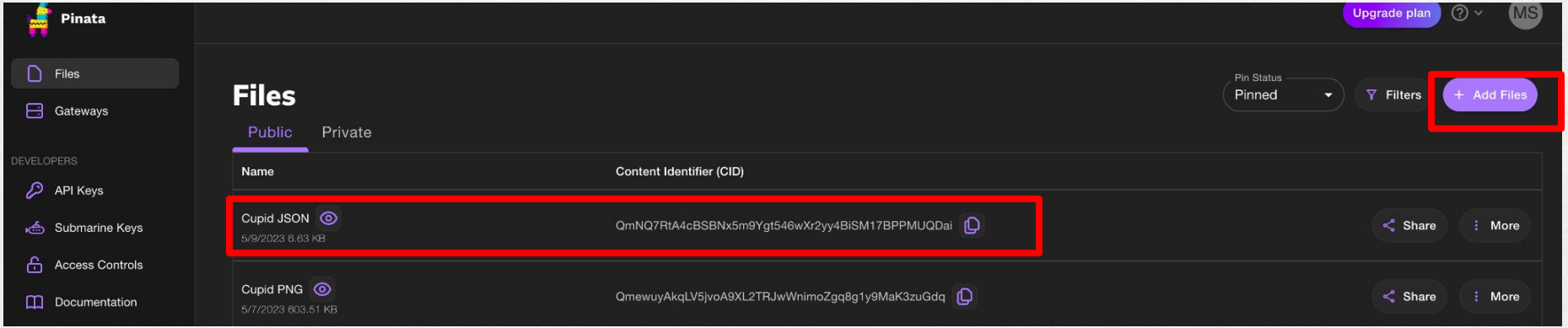
製作 Metadata “0.json” 檔案

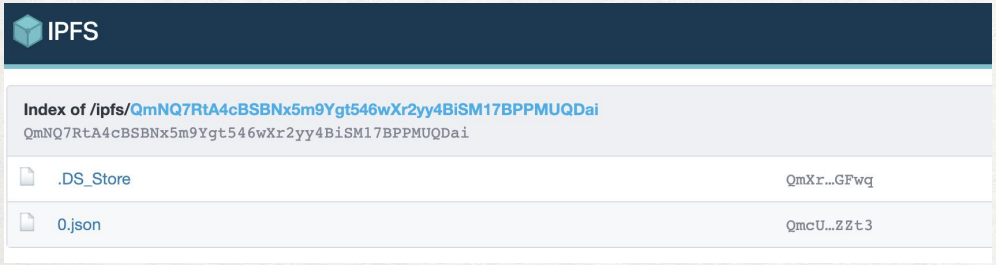
```
{
  "name": "Cupid",
  "image": "ipfs://QmewuyAkqLV5jvoA9XL2TRJwWnimoZgq8g1y9MaK3zuGdq/0.png",
  "description": "This NFT collection is created by Cupid",
  "external_url": "https://linktr.ee/siemingfong",
  "attributes": [
    {
      "trait_type": "Name",
      "value": "Cupid"
    },
    {
      "trait_type": "creator",
      "value": "Cupid"
    }
  ]
}
```

https://forum.openzeppelin.com/t/create-an-erc1155/4433

上傳 JSON 資料夾到 IPFS 取得 BaseURI

上傳資料夾



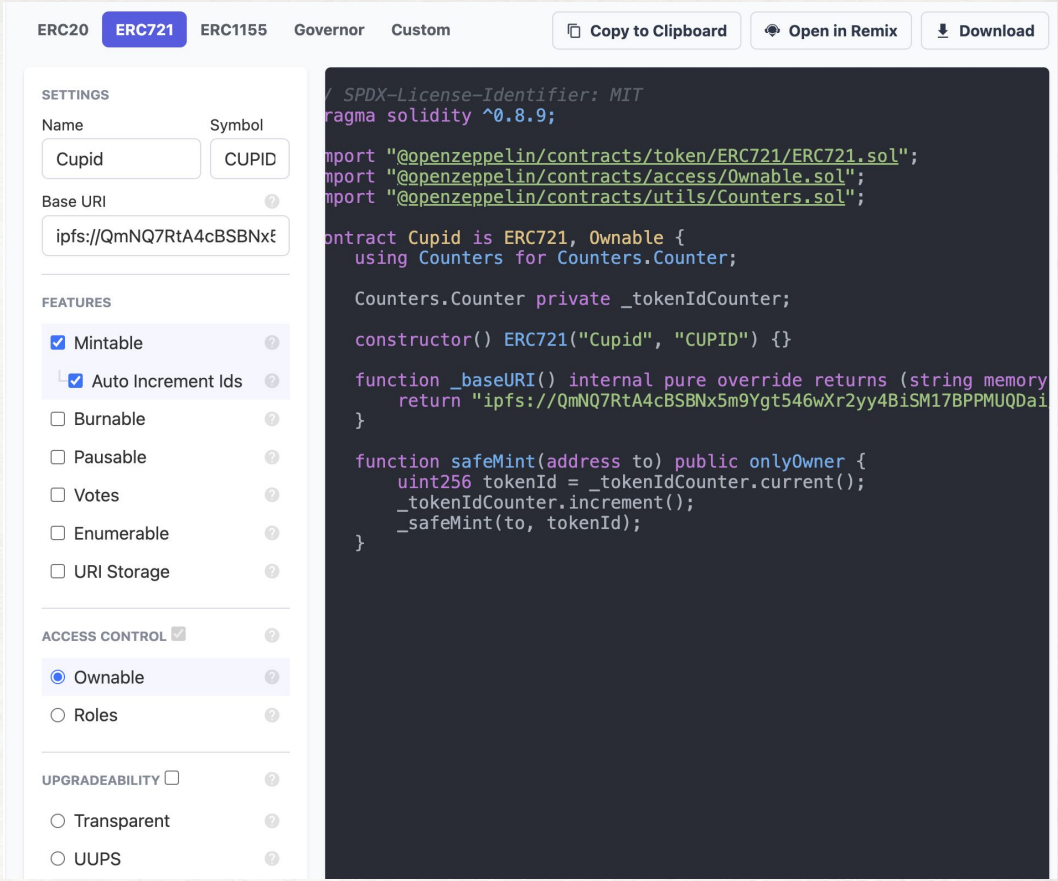


JSON資料夾 CID (Content IDentifier)

OpenZippelin Contract Wizard

Base URI = IPFS Json CID

ipfs://QmNQ7RtA4cBSBNx5m9Ygt546wXr2yy4BiSM17BPPMUQDai



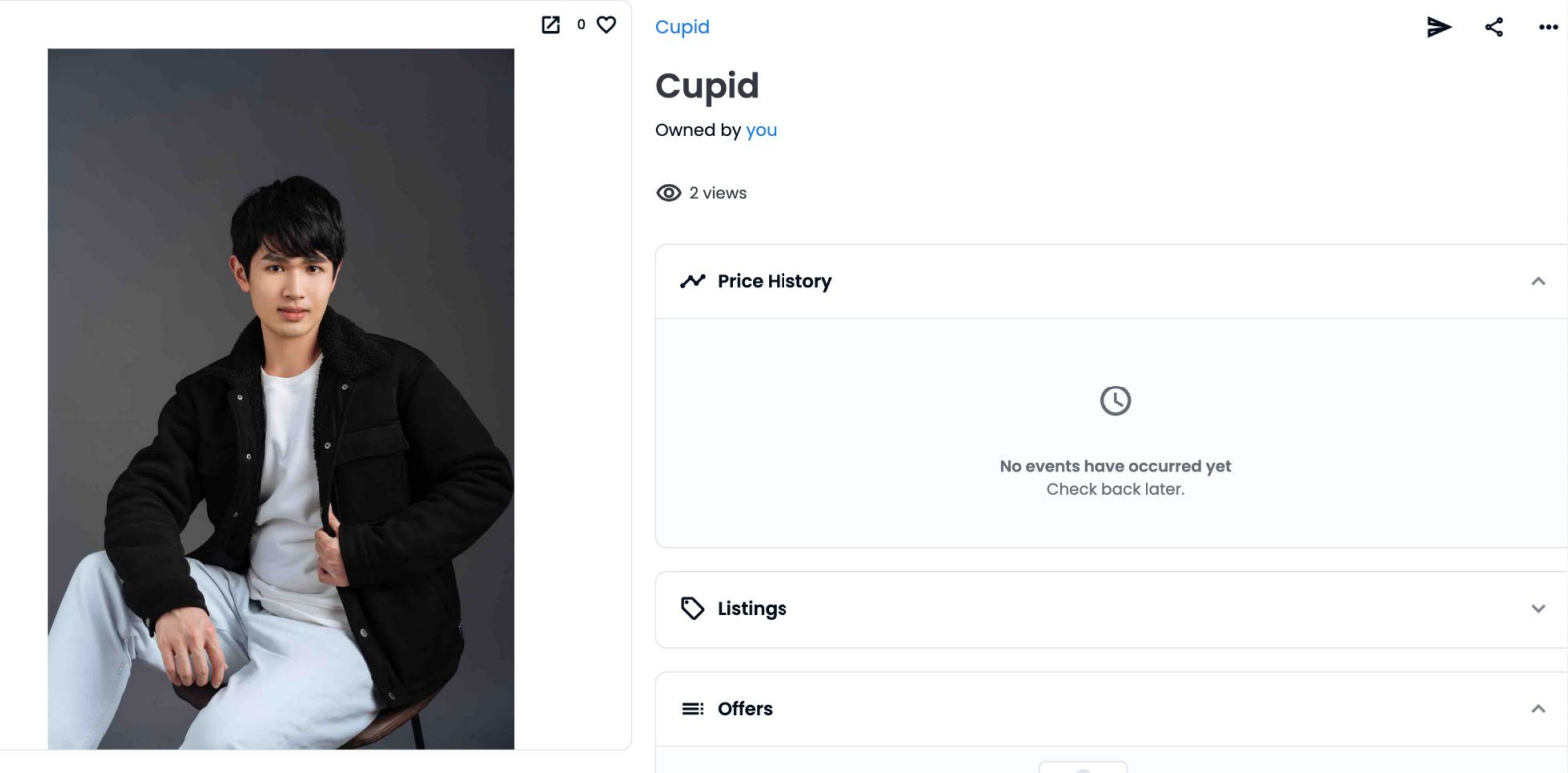
撰寫 Cupid ERC 721 合約

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.9;

import "@openzeppelin/contracts@4.8.3/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts@4.8.3/access/Ownable.sol";
import "@openzeppelin/contracts@4.8.3/utils/Counters.sol";

contract Cupid is ERC721, Ownable {
    using Counters for Counters.Counter;

    Counters.Counter private _tokenIdCounter;

    uint256 MAX_SUPPLY = 30;
    string baseURI = "ipfs://QmNQ7RtA4cBSBNx5m9Ygt546wXr2yy4BiSM17BPPMUQDai/";
    string baseExtension = ".json";

    mapping (address => bool) public mintedWallets;

    constructor() ERC721("Cupid", "CUPID") {}

    function tokenURI(uint256 tokenId) public view virtual override returns (string memory){
        require(
            _exists(tokenId),
            "ERC721Metadata: URI query for nonexistent token"
        );
    return string(abi.encodePacked(baseURI, Strings.toString(tokenId), baseExtension));
    }
    function safeMint(address to) public onlyOwner {
        uint256 tokenId = _tokenIdCounter.current();
        _tokenIdCounter.increment();
        _safeMint(to, tokenId);
    }
}
```

JSON CID

把整個合併變成代幣的 URI
baseURI/ + 代幣標號 +.json

檢查 Opensea Testnet



https://testnets.opensea.io/assets/mumbai/0x1ba18df26a694289f85bd466f58ceca7b33c57c4/0

Q&A

推薦學習

- Solidity 極簡入門 WTF Solidity
- 建造鐵人賽專屬NFT！(五)設定圖片/名稱Metadata並上傳至IPFS！