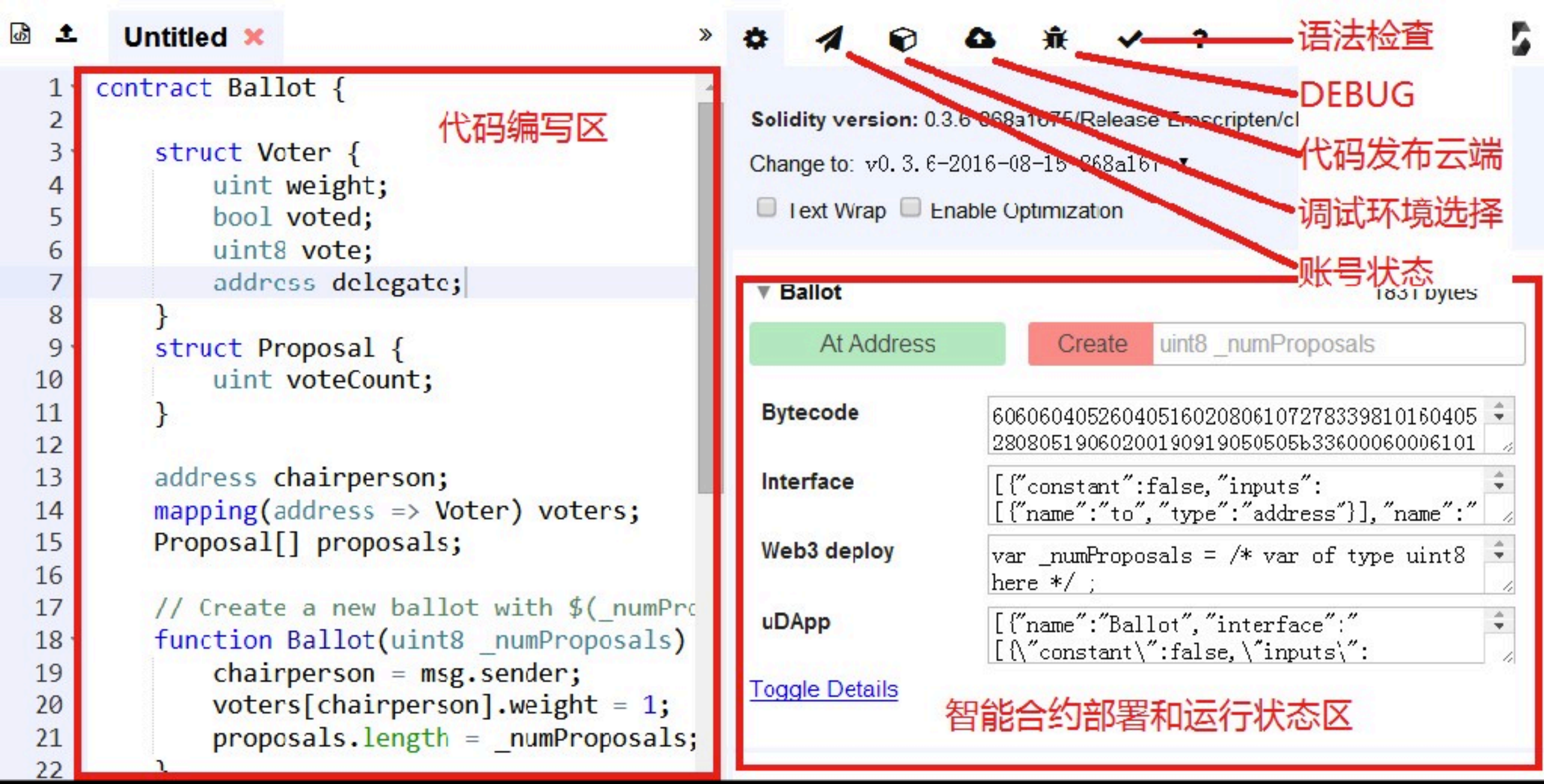
開發以太坊智能合約的方法

盧瑞山 教授

browser-solidity

- 智能合約瀏覽器版本的開發環境,可以支持在瀏覽器中直接開發、調試和編譯
- 對於初學者來說，可以快速上手，不需要安裝，非常方便
- 直接訪問地址使用：https://ethereum.github.io/browser-solidity/
- 或是這個地址https://remix.ethereum.org/
- Ethereum Studio：第三方公司開發的企業版智能合約在線IDE，功能強大，免費使用，可以作為企業級開發的一個工具，訪問地址：https://live.ether.camp/

browser-solidity IDE



```
contract Ballot {

    struct Voter {
        uint weight;
        bool voted;
        uint8 vote;
        address delegate;
    }
    struct Proposal {
        uint voteCount;
    }

    address chairperson;
    mapping(address => Voter) voters;
    Proposal[] proposals;

    // Create a new ballot with $(_numPro
    function Ballot(uint8 _numProposals)
        chairperson = msg.sender;
        voters[chairperson].weight = 1;
        proposals.length = _numProposals;
    }
```

Solidity version: 0.3.6-068a1075/Release Emscripten/cl

Change to: v0.3.6-2016-08-15-068a161 ▼

□ Text Wrap □ Enable Optimization

▼ Ballot 1831 bytes

uint8 _numProposals

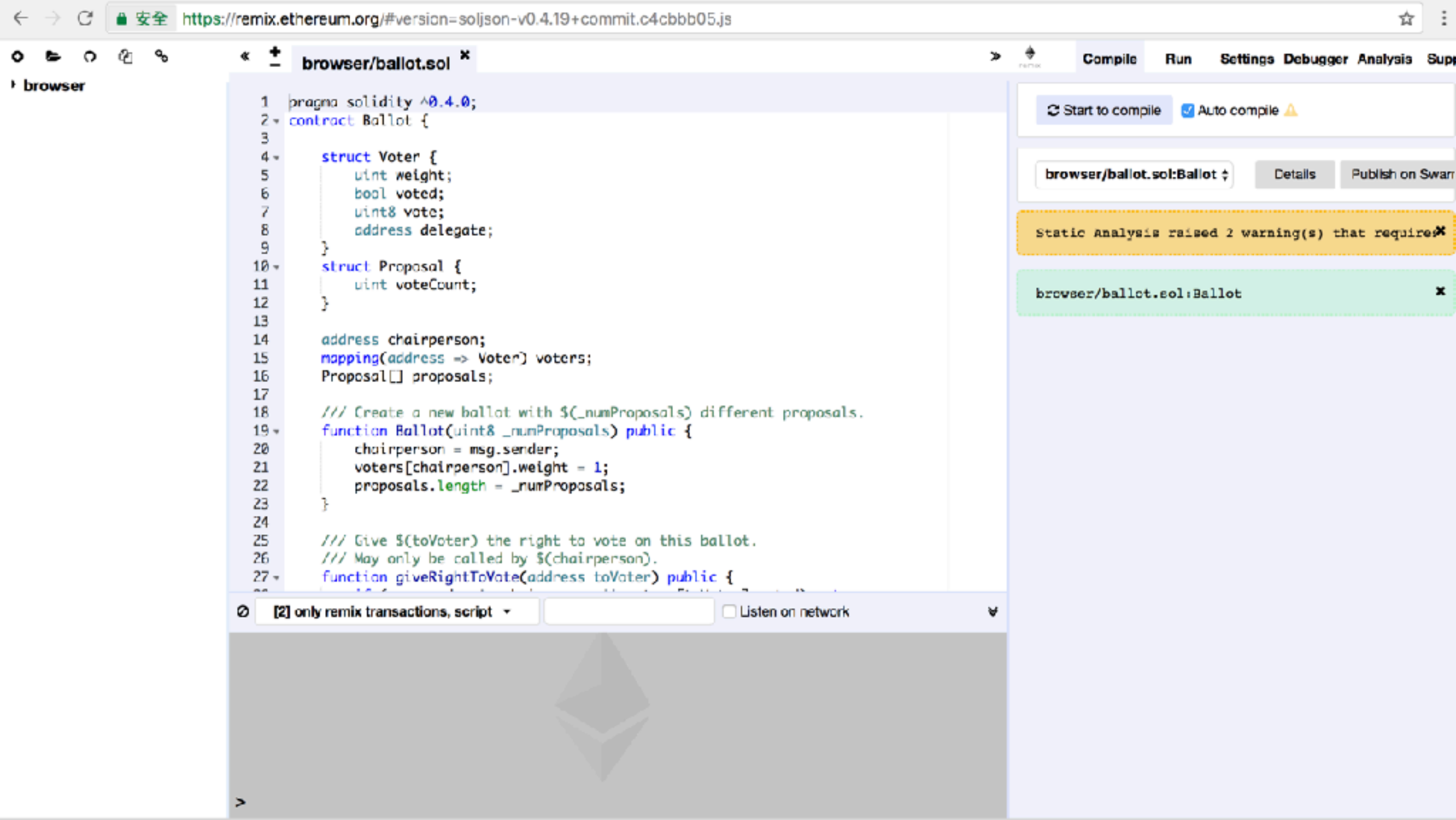
Bytecode
```
60606040526040516020806107278339810160405
2808051906020019091905050b33600060006101
```

Interface
```
[{"constant":false,"inputs":
[{"name":"to","type":"address"}],"name":"
```

Web3 deploy
```
var _numProposals = /* var of type uint8
here */ ;
```

uDApp
```
[{"name":"Ballot","interface":"
[{\"constant\":false,\"inputs\":
```

新版的介面



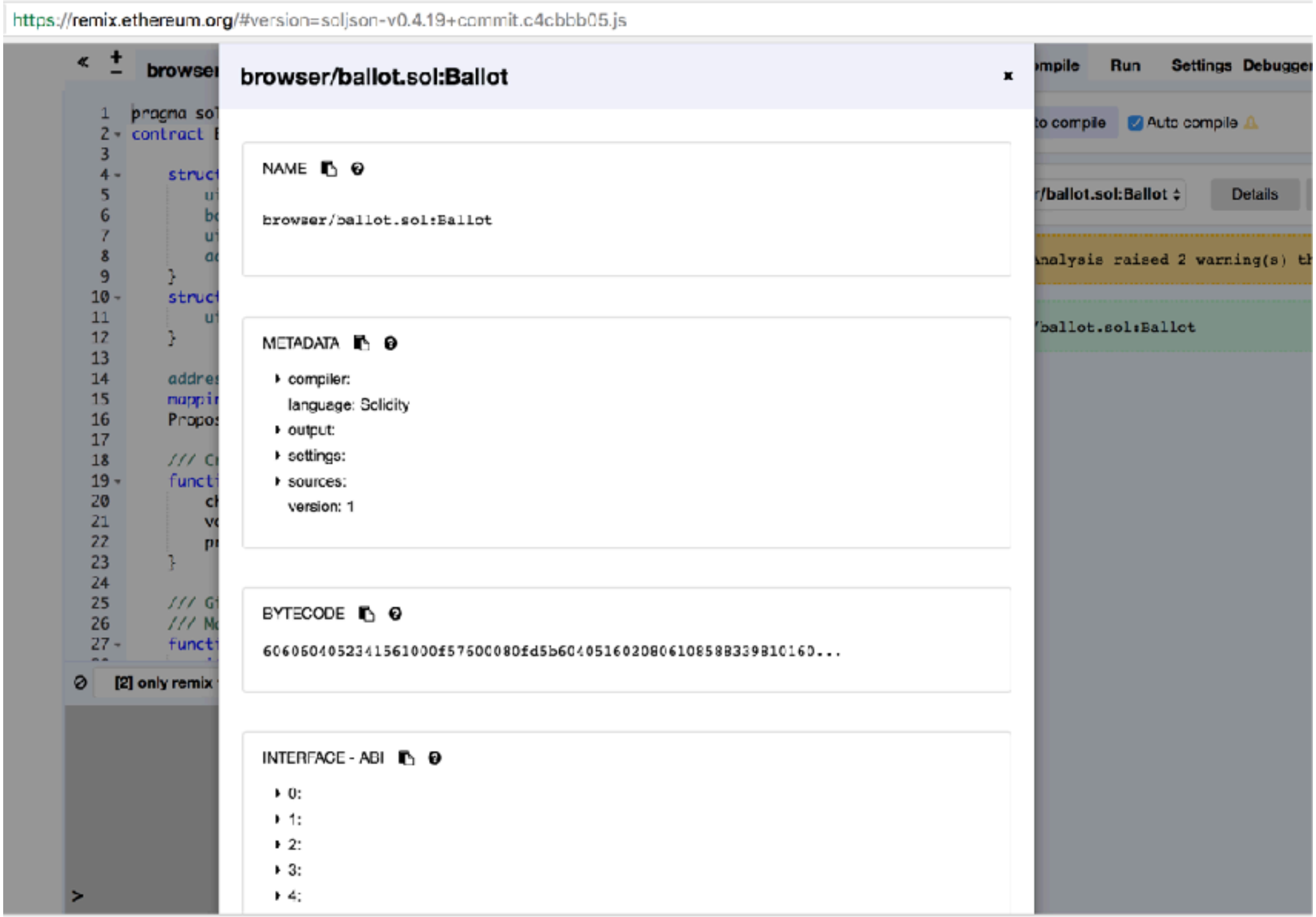
```
pragma solidity ^0.4.0;
contract Ballot {

    struct Voter {
        uint weight;
        bool voted;
        uint8 vote;
        address delegate;
    }
    struct Proposal {
        uint voteCount;
    }

    address chairperson;
    mapping(address => Voter) voters;
    Proposal[] proposals;

    /// Create a new ballot with $(_numProposals) different proposals.
    function Ballot(uint8 _numProposals) public {
        chairperson = msg.sender;
        voters[chairperson].weight = 1;
        proposals.length = _numProposals;
    }

    /// Give $(toVoter) the right to vote on this ballot.
    /// May only be called by $(chairperson).
    function giveRightToVote(address toVoter) public {
```

browser/ballot.sol:Ballot

Static Analysis raised 2 warning(s) that requires

browser/ballot.sol:Ballot

新版的介面



https://remix.ethereum.org/#version=soljson-v0.4.19+commit.c4cbbb05.js

```
pragma sol
contract B
    struct
        ui
        bo
        ui
        ad
    }
    struct
        ui
    }

    addres
    mappin
    Propos

    /// Cr
    functi
        ch
        vo
        pr
    }

    /// Gi
    /// Ma
    functi
```

```
analysis raised 2 warning(s) th
/ballot.sol:Ballot
```

browser/ballot.sol:Ballot ×

NAME

```
browser/ballot.sol:Ballot
```

METADATA

- compiler:
- language: Solidity
- output:
- settings:
- sources:
- version: 1

BYTECODE

```
6060604052341561000f57600080fd5b6040516020806108588339810160...
```

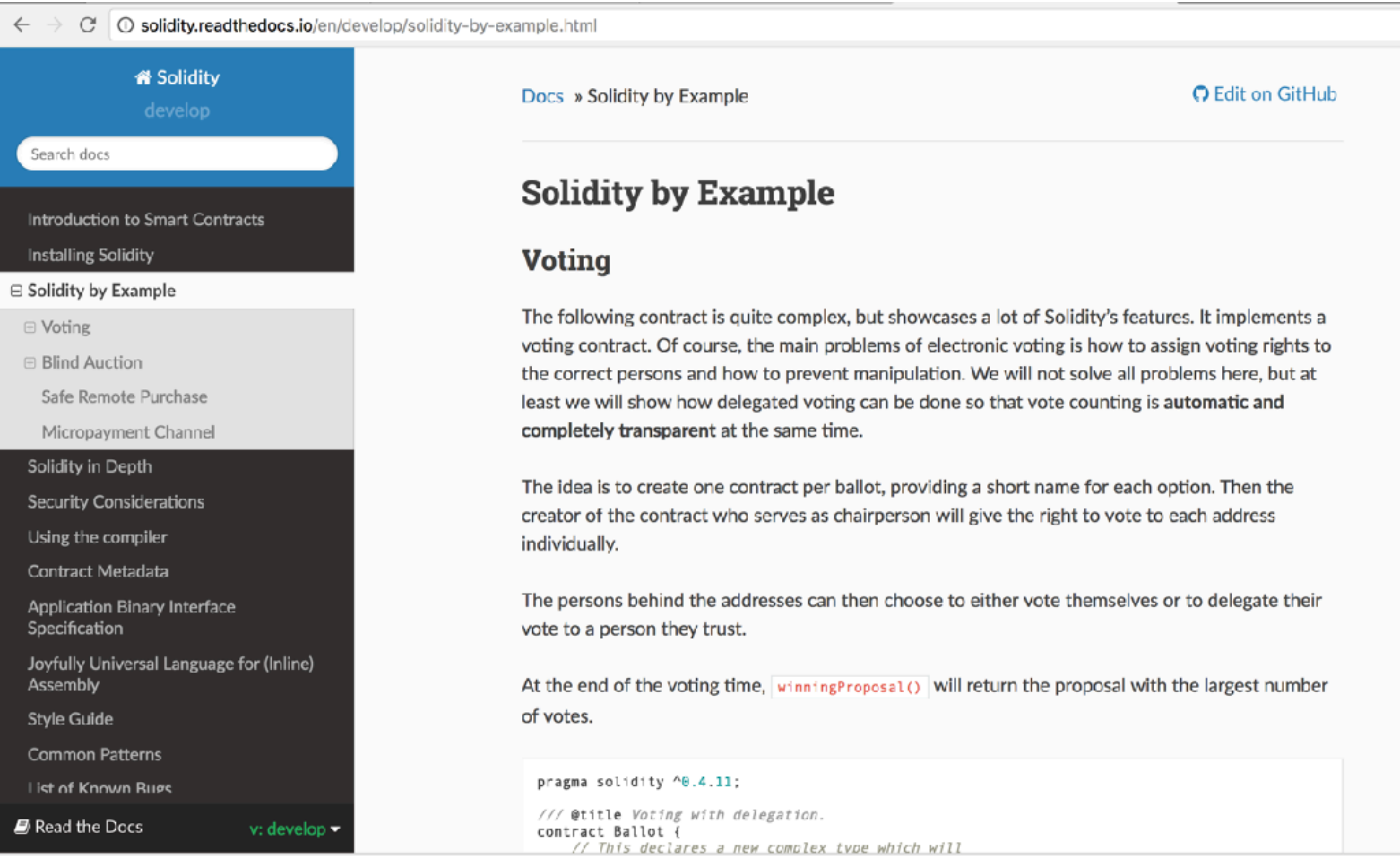
INTERFACE - ABI

- 0:
- 1:
- 2:
- 3:
- 4:

智能合約語法學習方法

智能合約的語法和示例可以在Solidity的文檔網站

http://solidity.readthedocs.io/en/latest/　查看



Solidity by Example

Voting

The following contract is quite complex, but showcases a lot of Solidity's features. It implements a voting contract. Of course, the main problems of electronic voting is how to assign voting rights to the correct persons and how to prevent manipulation. We will not solve all problems here, but at least we will show how delegated voting can be done so that vote counting is **automatic and completely transparent** at the same time.

The idea is to create one contract per ballot, providing a short name for each option. Then the creator of the contract who serves as chairperson will give the right to vote to each address individually.

The persons behind the addresses can then choose to either vote themselves or to delegate their vote to a person they trust.

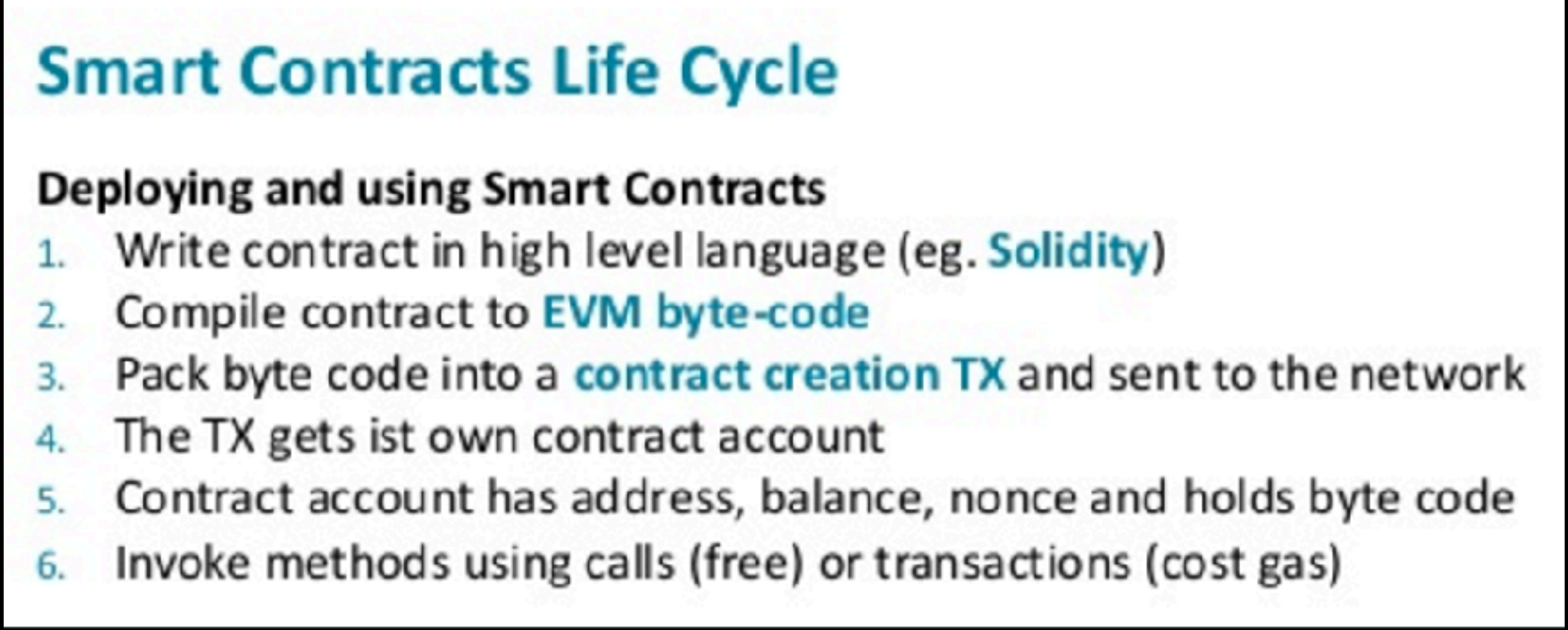
At the end of the voting time, `winningProposal()` will return the proposal with the largest number of votes.

```
pragma solidity ^0.4.11;

/// @title Voting with delegation.
contract Ballot {
    // This declares a new complex type which will
```

Visual Studio 2015

Visual Studio 2015：没错，就是微软的VS 2015，微软已经把以太坊的智能合约编写功能整合了，可以看出微软对以太坊的重视。

Summary

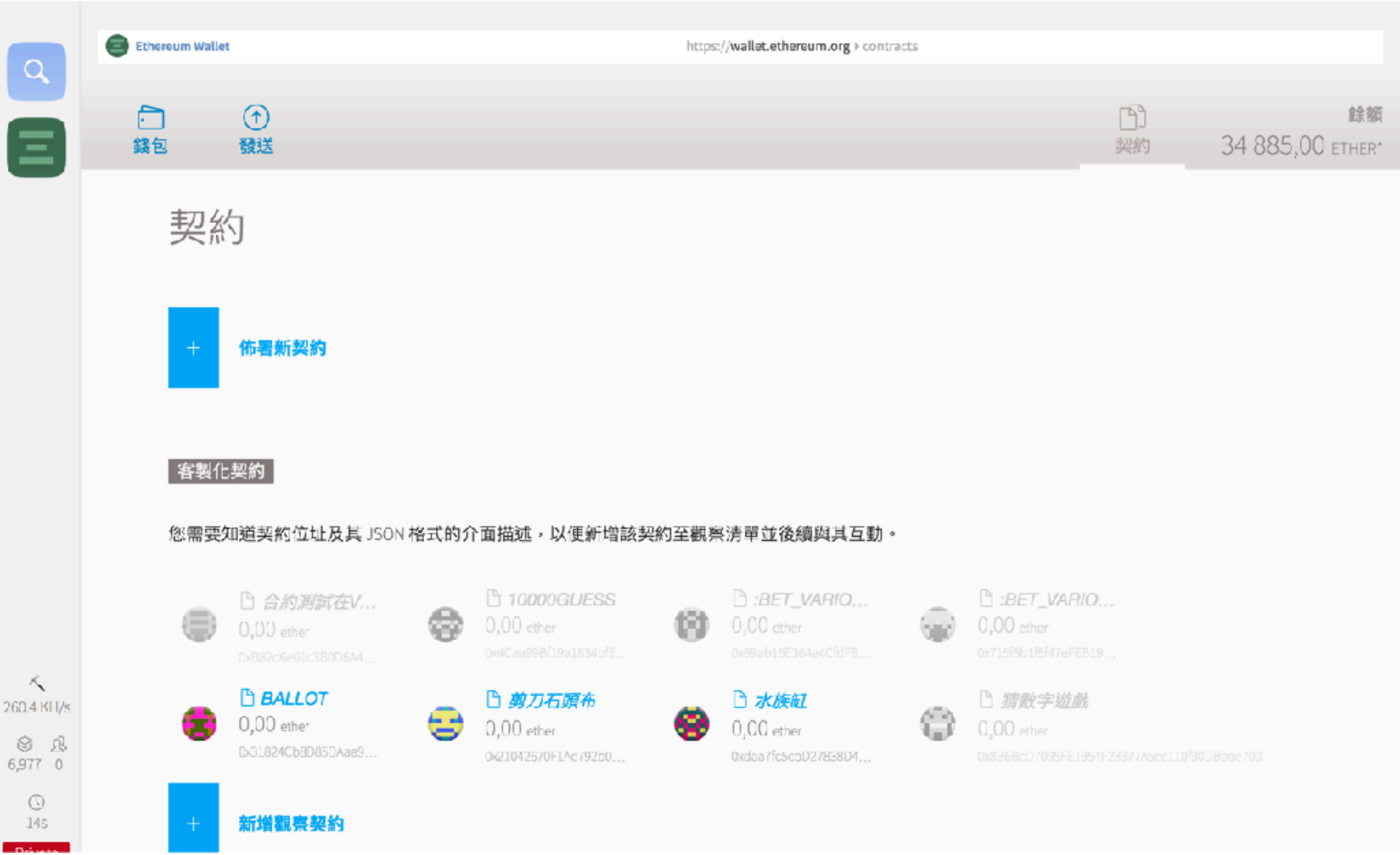


Smart Contracts Life Cycle

Deploying and using Smart Contracts

1. Write contract in high level language (eg. **Solidity**)
2. Compile contract to **EVM byte-code**
3. Pack byte code into a **contract creation TX** and sent to the network
4. The TX gets ist own contract account
5. Contract account has address, balance, nonce and holds byte code
6. Invoke methods using calls (free) or transactions (cost gas)

合約寫好與編譯之後



Ethereum Wallet　https://wallet.ethereum.org › contracts

錢包　發送　契約　餘額 34 885,00 ETHER*

契約

＋ 佈署新契約

客製化契約

您需要知道契約位址及其 JSON 格式的介面描述，以便新增該契約至觀察清單並後續與其互動。

合約測試在V...
0,00 ether
0xB82d6e01c3B0D6A4...

10000GUESS
0,00 ether
0x4Caa99Bf19a1834bfE...

:BET_VARIO...
0,00 ether
0x69ab15E364a4CfdF6...

:BET_VARIO...
0,00 ether
0x715f5b1f5f47eFEB19...

BALLOT
0,00 ether
0x31824CbBD85DAae9...

剪刀石頭布
0,00 ether
0x21042570F1Ac792b0...

水族缸
0,00 ether
0xdaa7fc5cbD27838D4...

猜數字遊戲
0,00 ether
0x836BcD7095FE1954F23377A5cc110f30DB6ae703

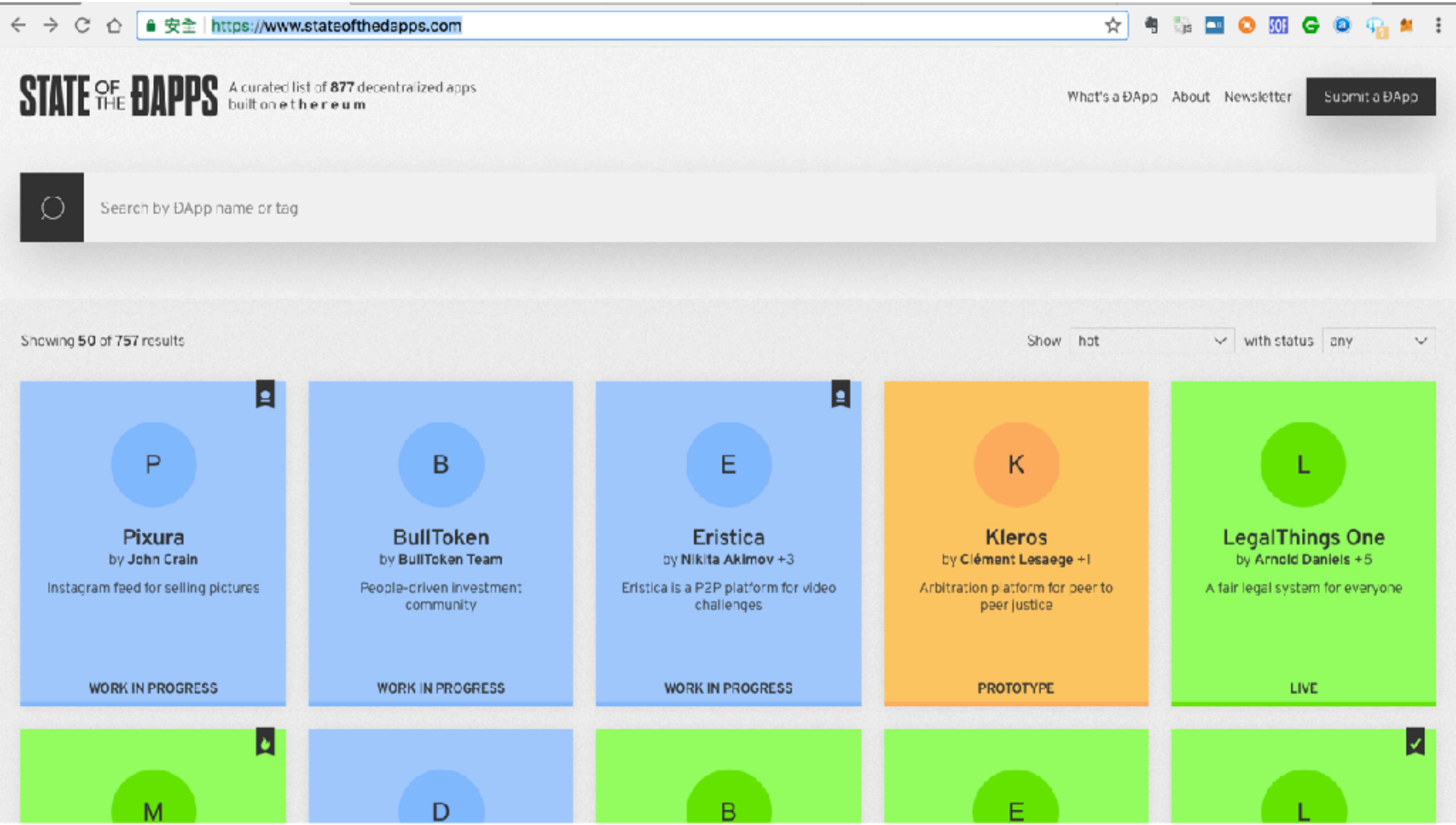
＋ 新增觀察契約

260.4 KH/s
6,977　0
14s
Private

如何自我進步？

多看別人的Dapp範例

https://
www.stateofthedapps.com/



安全 https://www.stateofthedapps.com

STATE OF THE ĐAPPS

A curated list of **877** decentralized apps built on **ethereum**

What's a ĐApp About Newsletter Submit a ĐApp

Search by DApp name or tag

Showing **50** of **757** results

Show hot with status any

Pixura
by **John Crain**
Instagram feed for selling pictures
WORK IN PROGRESS

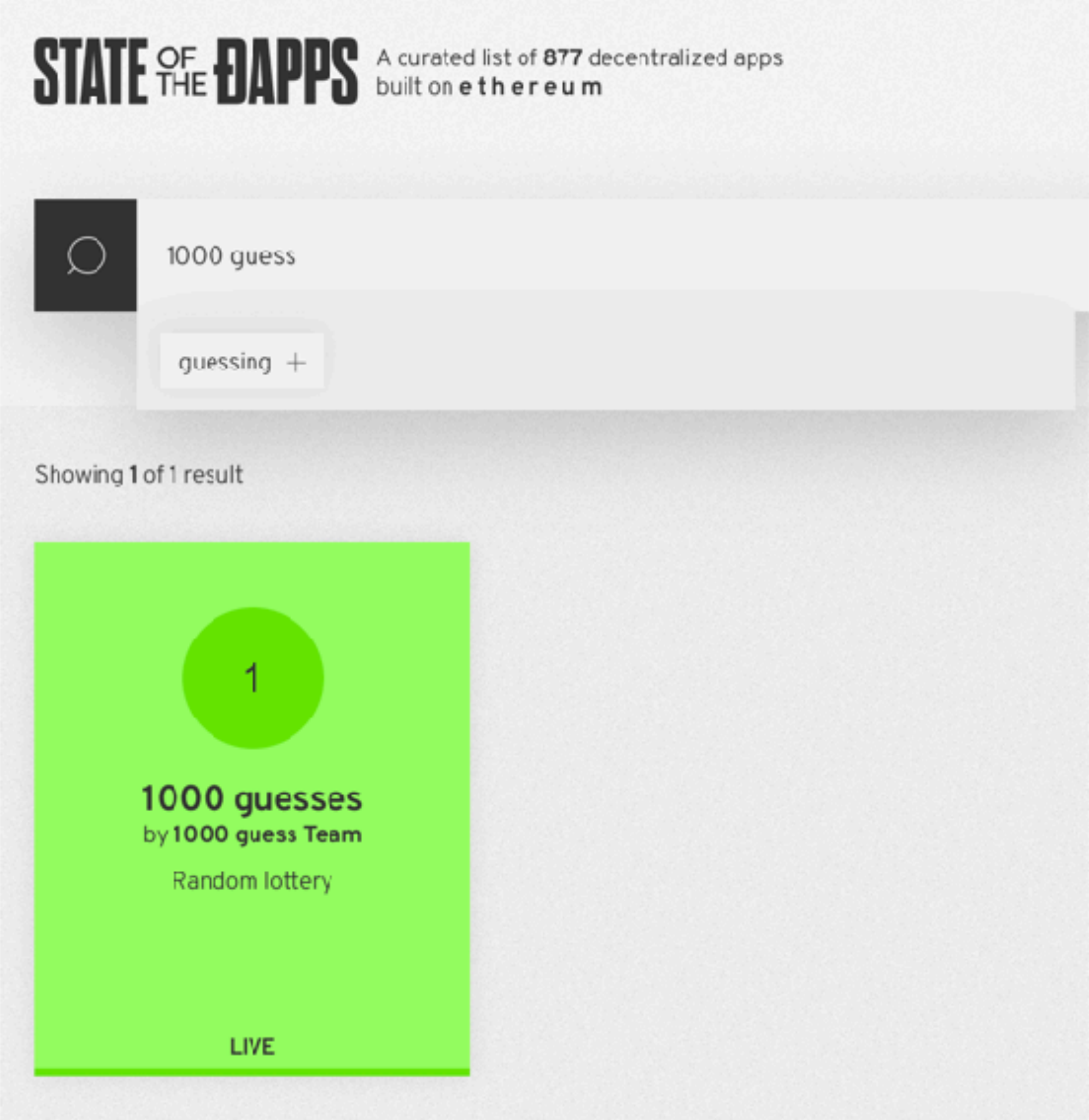
BullToken
by **BullToken Team**
People-driven investment community
WORK IN PROGRESS

Eristica
by **Nikita Akimov** +3
Eristica is a P2P platform for video challenges
WORK IN PROGRESS

Kleros
by **Clément Lesaege** +1
Arbitration platform for peer to peer justice
PROTOTYPE

LegalThings One
by **Arnold Daniels** +5
A fair legal system for everyone
LIVE

老師課堂教過的範例



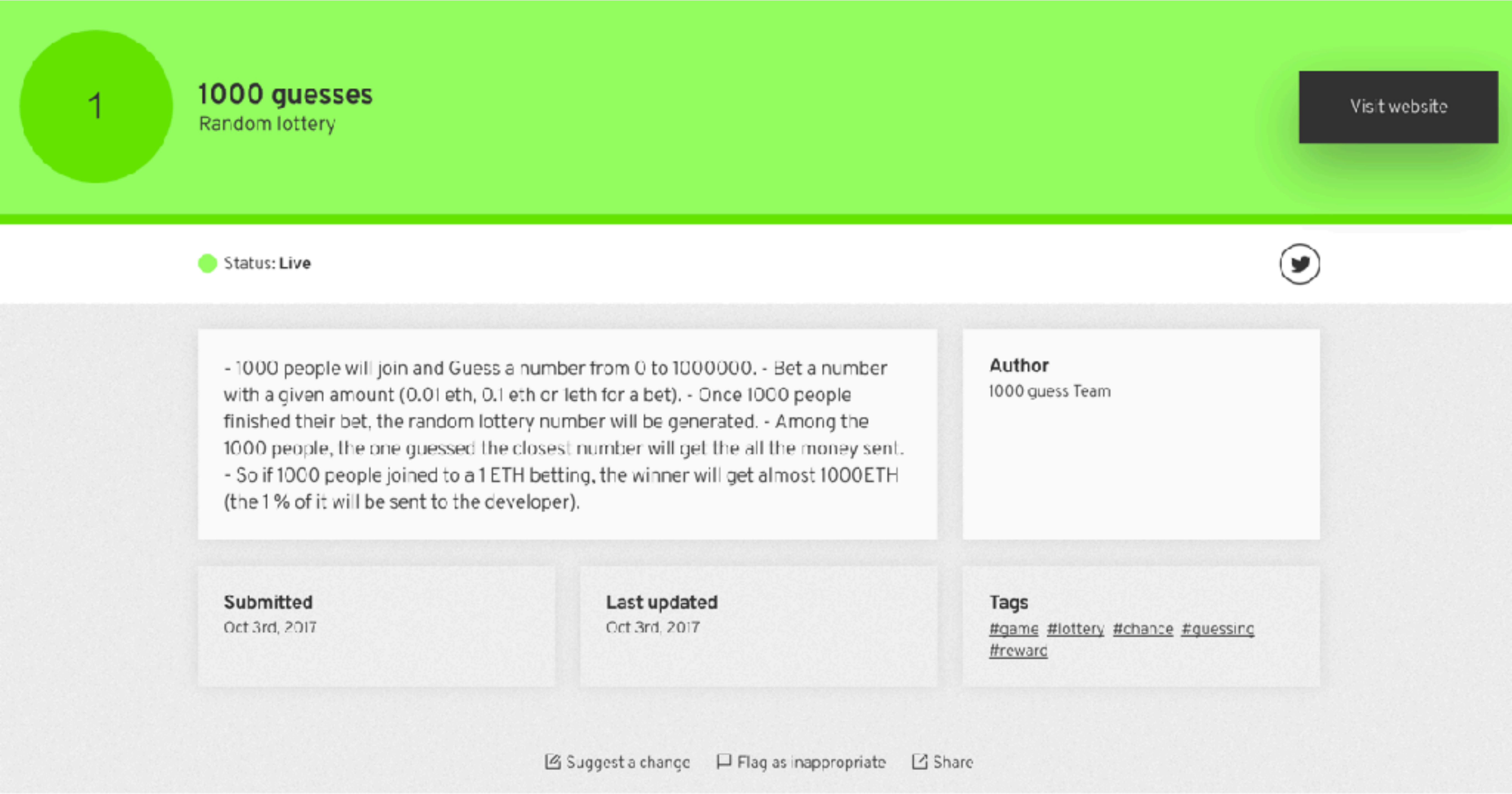
STATE OF THE ĐAPPS A curated list of 877 decentralized apps built on ethereum

1000 guess

Showing 1 of 1 result

1000 guesses
by 1000 guess Team
Random lottery

LIVE



1000 guesses

Random lottery

● Status: **Live**

- 1000 people will join and Guess a number from 0 to 1000000. - Bet a number with a given amount (0.01 eth, 0.1 eth or 1eth for a bet). - Once 1000 people finished their bet, the random lottery number will be generated. - Among the 1000 people, the one guessed the closest number will get the all the money sent. - So if 1000 people joined to a 1 ETH betting, the winner will get almost 1000ETH (the 1 % of it will be sent to the developer).

Author
1000 guess Team

Submitted
Oct 3rd, 2017

Last updated
Oct 3rd, 2017

Tags
#game #lottery #chance #guessing #reward