当区块链遇上智能合约

庄重

2016-01-16

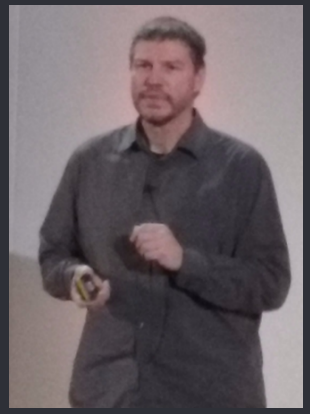
什么是智能合约

Conclusion of smart contract discussion: no-one has a clue what a smart contract actually is, and if we did it'd need oracles.

-- Peter Todd

智能合约概念的由来

- Nick Szabo， 1994年:
  - 以自动售货机为例子
  - 扩展到所有可以数字化的资产
  - 自动执行通常意义上的合约

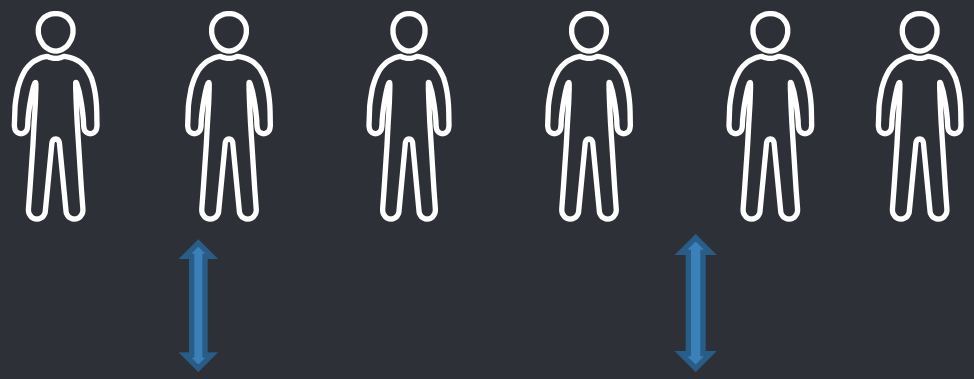


智能合约通常的执行过程是：

1. 创建合约，锁定数字资产
2. 调用合约提供的接口执行程序
3. 由合约来完成资产的转移

区块链上的智能合约

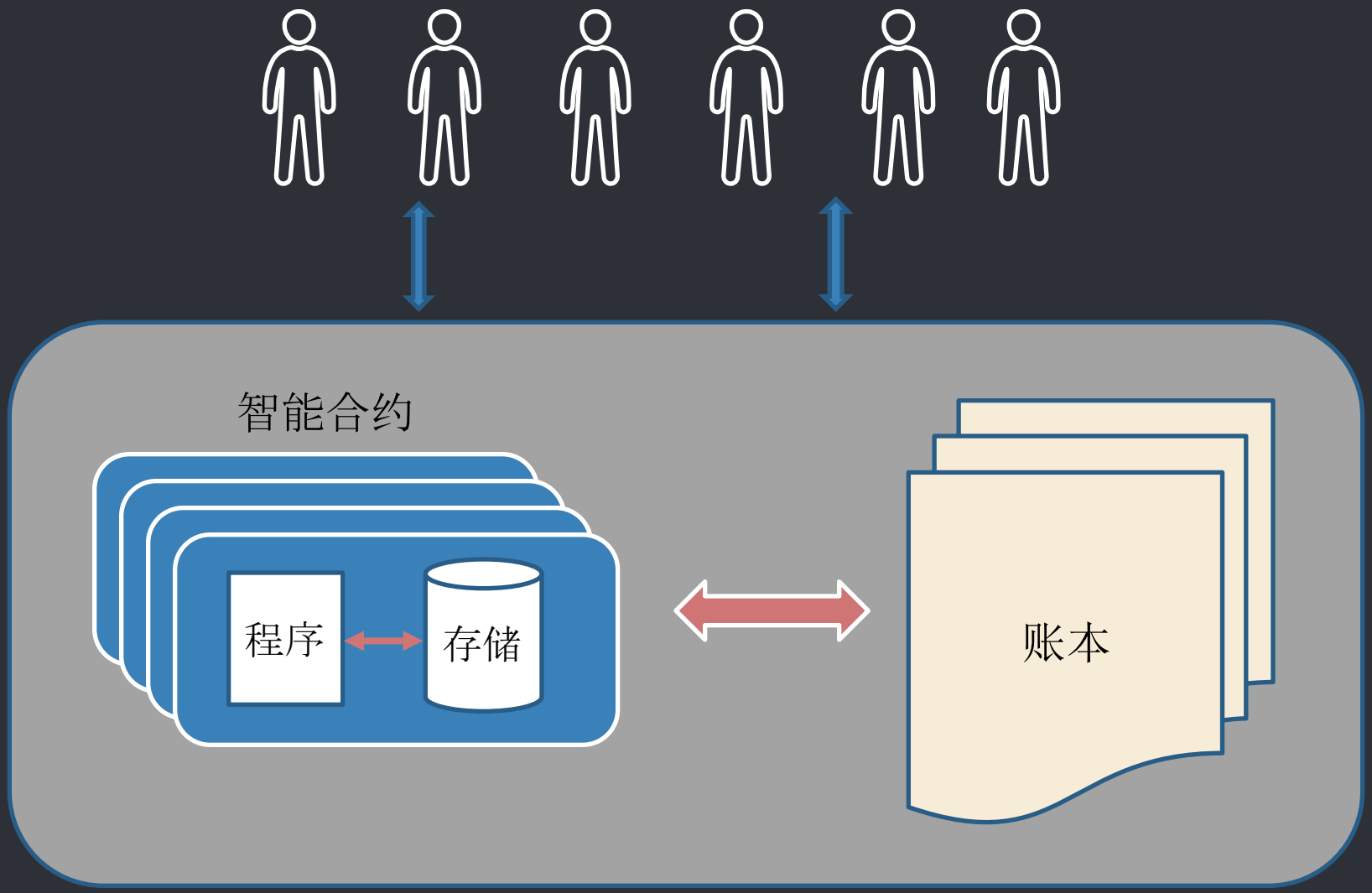
区块链



区块链的功能

- 存储数据（全局的账本）
- 进行计算（校验交易）
- 对数据和计算达成共识
- 内部状态对所有人可见

以太坊 ---- 可编程的智能合约平台



以太坊 ---- 可编程的智能合约平台

智能合约
用Solidty等语言编写，图灵完备，在自己定义的虚拟机EVM上执行。

区块链
基于余额的全局账本，地址可能带有智能合约，每个智能合约会保存独立的状态

合约的执行方式是给对应的地址发送交易，并说明调用的方法和参数。一个合约的执行可能会触发多个其他合约，产生数个交易。

以太坊的一些细节

- 两种Account
  - 普通Account，私钥由用户掌握
  - 合约Account，自动创建，用户没有私钥
  - 两种Account都有余额
- Storage
  - 每个合约256bit到256bit的Key-Value存储
- 交易
  - 包含Sender, Receiver, Gas, Data
  - Receiver为空代表创建新的合约

以太坊的一些细节

- EVM
  - 设计了一套指令集
  - 基于栈的虚拟机
  - 访存空间无限
  - 嵌套深度最大为1024
  - 通过Log将EVM中的状态发送给外部
  - 合约执行过程中会消耗Gas，限制程序的复杂度

在以太坊上开发DApp

- 以太坊
  - 用Solidity编写合约
  - 在以太坊上发布
- Web
  - 前端开发
  - 使用以太坊的JS API查看合约执行中产生的Event

```
contract Coin {
  .........
    event Send(address from, address to, uint value);
    function send(address receiver, uint amount) {
        if (balances[msg.sender] < amount) return;
        balances[msg.sender] -= amount;
        balances[receiver] += amount;
        Send(msg.sender, receiver, amount);
    }
  .........
}
```

```
var event = coin.Send({}, '', function(error, result){
  if (!error)
      console.log("Coin sent: " +
                  result.args.value +
                  " coins were sent from " +
                  result.args.from + " to " +
                  result.args.to
                 );
});
```

智能合约

- 合约的执行由程序控制，降低了执行带来的成本
- 减少了参与方之间的信任？

- 随之带来的问题
  - 可扩展性
  - 合约的安全性
  - 合约的法律效力

编写安全的智能合约

怎样设计安全的协议

智能合约执行中可能的问题

- 合约的参与方可能随时退出，参与方发送给合约的钱需要能找回
- 对智能合约的调用是公开的，合约的参与方可以通过他人的行动谋利
- 合约的执行可能意外中止，如Gas耗尽，嵌套层数太深
- 矿工是否能有选择性得加入交易以及不广播对自己不利的区块

实例

```
contract Bet {
    uint num_teams;
    uint[2] choices;
    address[2] teams;

    function Bet() {
        num_teams = 0;
    }

    function makeChoice (uint choice) returns (uint status) {
        if (num_teams < 2 && msg.value == 1000) {
            choices[num_teams] = choice;
            teams[num_teams] = msg.sender;
            num_teams++;
            return 1;
        } else {
            return 0;
        }
    }

    function reveal() returns (address winner) {
        if (num_teams<2)
            return 0;
        teams[0].send(1000);
        teams[1].send(1000);
        uint A = choices[0];
        uint B = choices[1];
        if ((A+B)%2==0) {
            return teams[0];
        } else {
            return teams[1];
        }
    }
}
```

Bug 1

```
function makeChoice (uint choice) returns (uint status) {
    if (num_teams < 2 && msg.value == 1000) {
```

可能的问题有：

1. 第3个队伍误操作
2. 队伍发送到合约的钱不是1000
3. 第2个队始终不参与投票

发送到智能合约的钱只有合约才能解锁

Bug 2

- 第1个队伍的选择公开可见
  - 改为提交Hash(choice,nonce)
  - 增加open函数，让队伍公布其选择和nonce，与之前的Hash校验
  - 当两个队伍都open后执行reveal确定结果
  - 需要给两个队伍增加open的时限

Bug 3

```
teams[0].send(1000);
teams[1].send(1000);
```

- 在执行send的时候会触发合约的执行，如果此时超过了栈的限制，退款会失败
  - 需要在相应函数入口增加检查

```
function checkStack() returns (uint res) {
    return 1;
}

function reveal() returns (address winner) {
    if (checkStack()!=1)
        return 0;
```

智能合约平台现状

不完整列表

- Smartcontract.com
- Orisi
- Codius
- Symboint
- Hedgy
- BitHalo
- Mirror
- Hyperledger
- Eris Industries
- Ethereum
- 智能坊
- 小蚁
- Colored Coin
- Bitcoin

以太坊

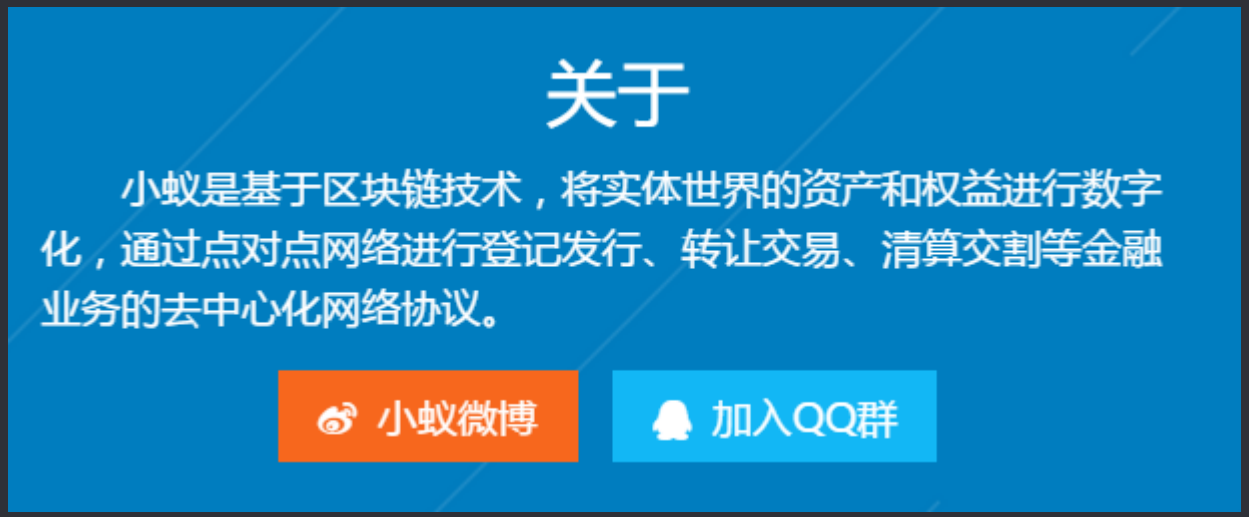
- 当下最为成熟的公开区块链上的智能合约平台
- 以太坊开发组提供了很多工具用于开发合约以及Dapp
- 兼容以太坊的EVM是私有链的好选择

比特币

- 尽管脚本不是图灵完备，比特币上仍然有我们熟知的很多智能合约，需要关注社区对新OpCode的态度
  - 多签名地址
  - Micro Payment Channel
  - 闪电网络提出的双向支付通道（需要新OpCode)
  - Counterpart等比特币区块链上的二层协议

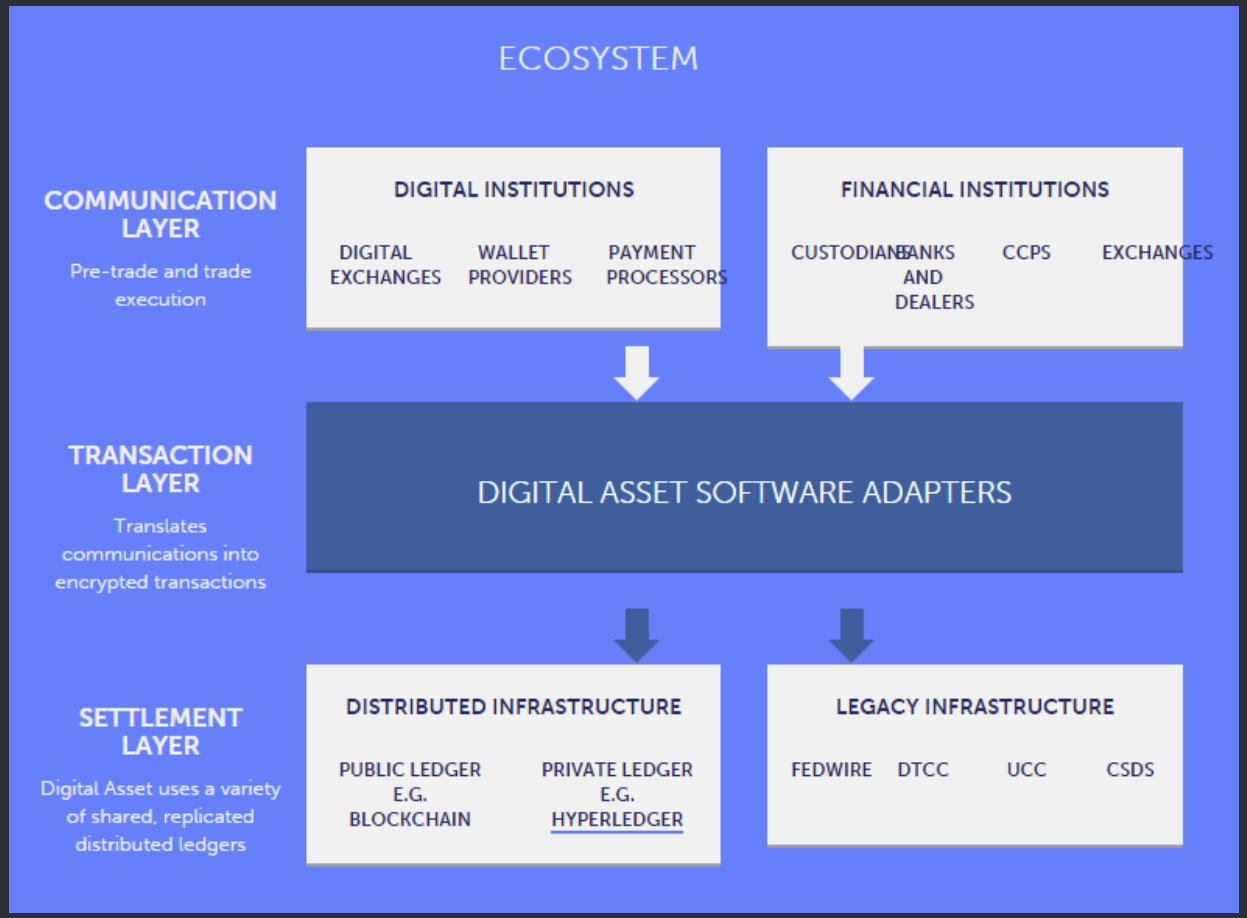
小蚁

- 小蚁扩展了比特币的脚本，实现了图灵完备



Digital Asset Holdings

- 私有区块链，共识算法



Mirror.co

◦Mirror is a financial contracts platform that provides hedging instruments to OTC markets

Hedging and Risk Management Tools

Robust financial contracts platform for OTC markets.

Smart Financial Contracts

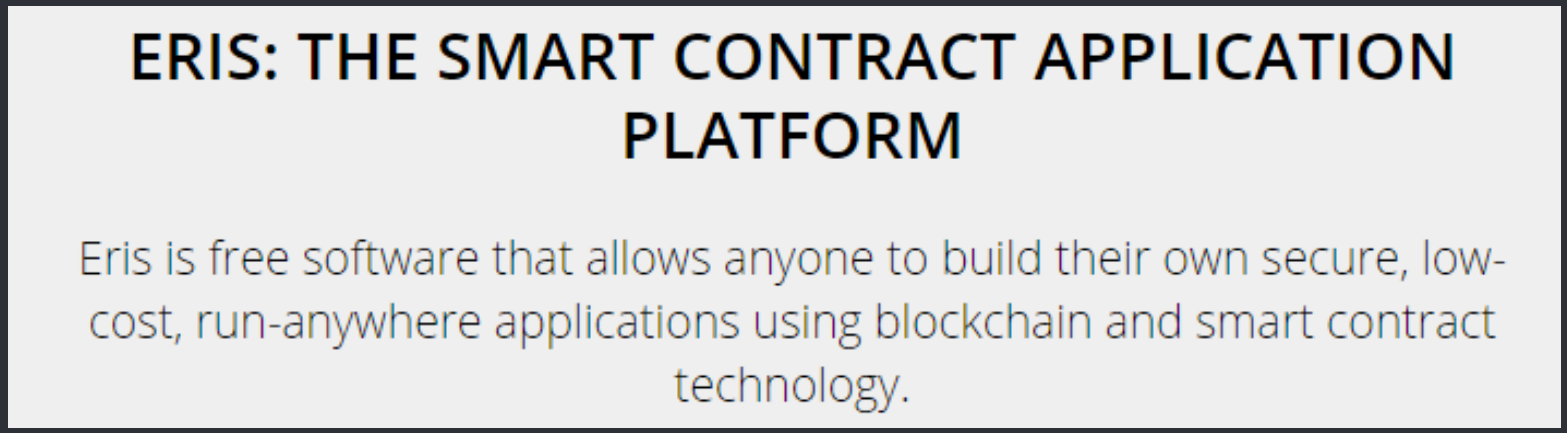
Standardized contract creation and settlement with Mirror's oracle service.

Risk Measurement for OTC Assets

Instruments to price and hedge risk for institutional investors.

Eris Industries

- 私有区块链，兼容以太坊的虚拟机
- 共识算法



搭建自己的智能合约平台

- 账本
  - 公有链？私有链？ 协作链？
  - PoW？PoS？共识算法？
- 合约
  - 兼容以太坊，根据自身业务需求扩展

潜在的研究方向

- 编程语言
  - 设计类型系统更为完备的语言，一定程度上减少不安全的合约
  - 引入形式化验证，证明合约的安全性
- 解决隐私问题
  - 结合零知识证明，在联邦链或者私有链上保证用户执行合约时的隐私（零知识证明通常体积较大）
- 可扩展性
- 改进智能合约平台的模型

参考引用

1. https://erisindustries.com/
2. http://mc2-umd.github.io/ethereumlab/
3. http://digitalasset.com/
4. https://ethereum.gitbooks.io/frontier-guide/
5. https://www.antshares.com

“

◦感谢大家！