云里雾里的区块链

——通过西夏普山寨币探究区块链基础技术

PRESENTED BY DAVID FAN

0

西夏普山寨币

- 为什么要创建Key，Key有什么用，干嘛不创建一个账户？
- 算24点是不是就是算hash，算对了就算挖到矿了，太儿戏了吧，矿什么时候挖完，我为什么要去傻傻的算24点？
- balance账户余额，不是么有账户吗，哪来的账户余额， 咋还有有个Un-balance？
- 数据在哪存储的，服务器在哪部署？
- 我挖到矿了，奖励了24，放哪去了，拿出来让我看看？
- 创建交易我输入了金额，然后钱转给别人了，交易咋发出去的，在哪给我扣帐的？
- **总结一句：这是个什么鬼！**

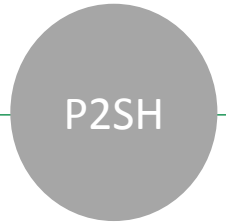
所以，我们先来演示一下西夏普山寨币

0

今天会讲到哪

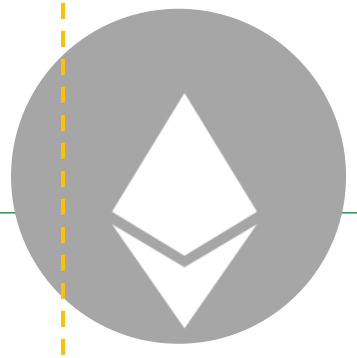
区块链1.0

—2008，以比特币为代表，**分布式数据存储、点对点传输、共识机制、加密算法等**计算机技术的新型应用模式。



区块链1.5

—2012，还是比特币，但增加了**P2SH(Pay to script hash)**



区块链2.0

—2015，以太坊为代表，支持**智能合约**

区块链3.0

—2018，EOS上线，号称支持跨链、TPS达到百万级

CONTENTS

1

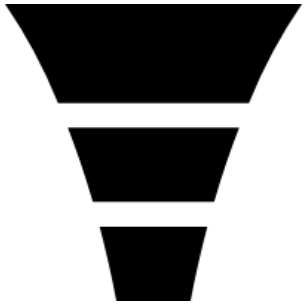
区块链中的加密算法

Encryption algorithm used in block-chain

1 区块链中的加密算法

非对称加密算法

- 如：RSA 、ECC（椭圆曲线算法）
- 特点：一对密钥公钥和私钥，用一个加密，用另一个可以解开。
- 用法：用别人的公钥加密，用自己的私钥签名
- 在区块链里：公钥就是你的身份，私钥就是用来证明你的身份的。



Hash算法

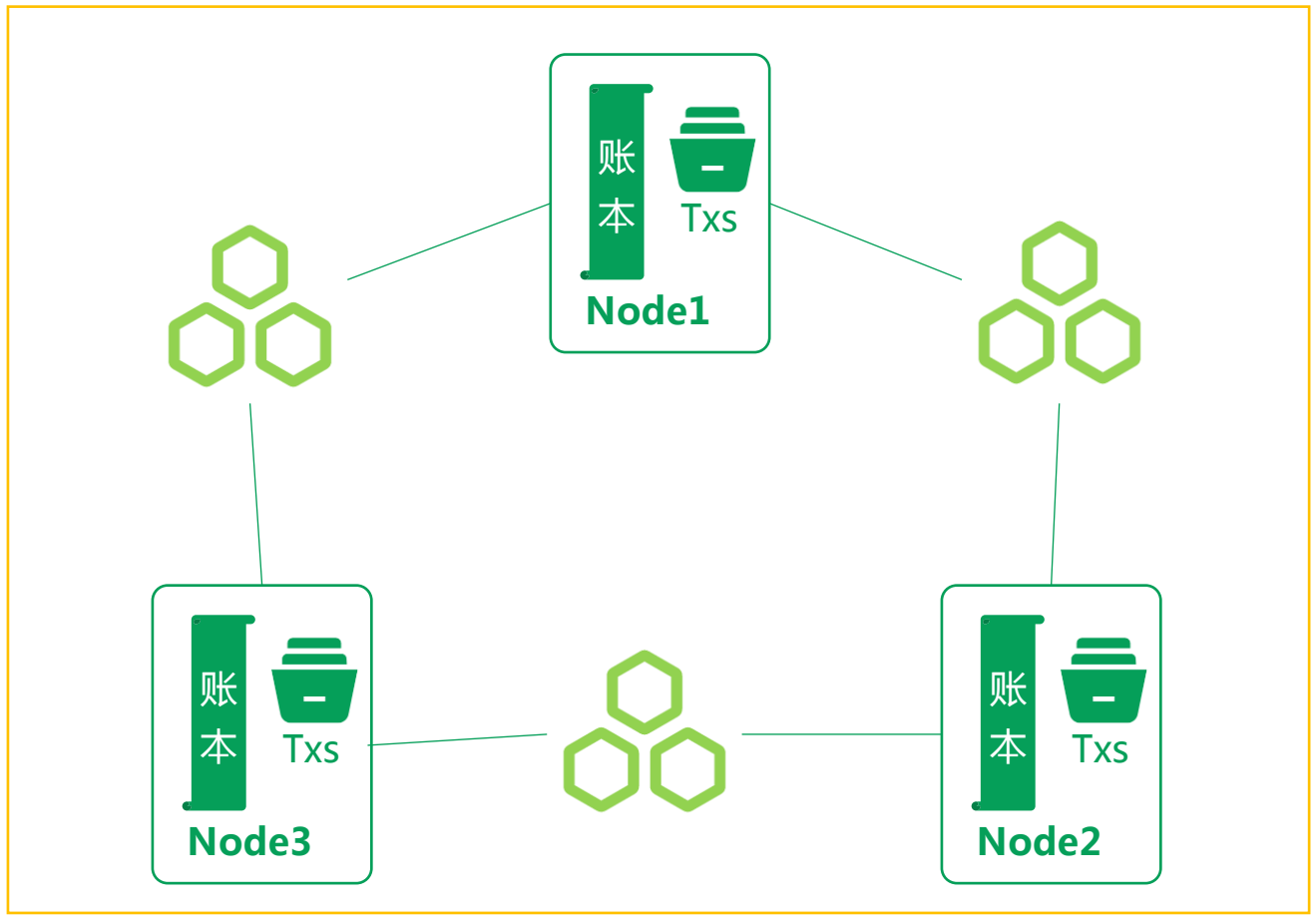
- 如：SHA256 、MD5
- 特点：不管输入多长，都能给你返回一个定长的数据，并且不可逆推。不同的输入返回不同的输出。
- 在区块链里：一堆交易通过Merkel 树算hash，对block header算hash（改nonce，不停的算就是挖矿），防篡改。

2

怎么实现去中心化的

How to achieves decentralization

2 怎么实现去中心化的



分布式共识机制

所有节点都是平等的，每个节点都有一套完整的数据。

1. 新的交易被广播到所有节点上。
2. 每个节点将新的交易放进内存中的临时集合中。
3. 在每个回合，一个**随机**(另一个共识机制，挖矿，谁有权记账)的节点可以广播它创建的区块。
4. 其他节点可以选择接受这个block，前提是如果block里的交易都是有效的（ Unspent，有真有效签名，输入==输出）。
5. 节点如果接受了这个block就会把这个block写到自己本地的账本，然后在这个block下面再继续建新的block。

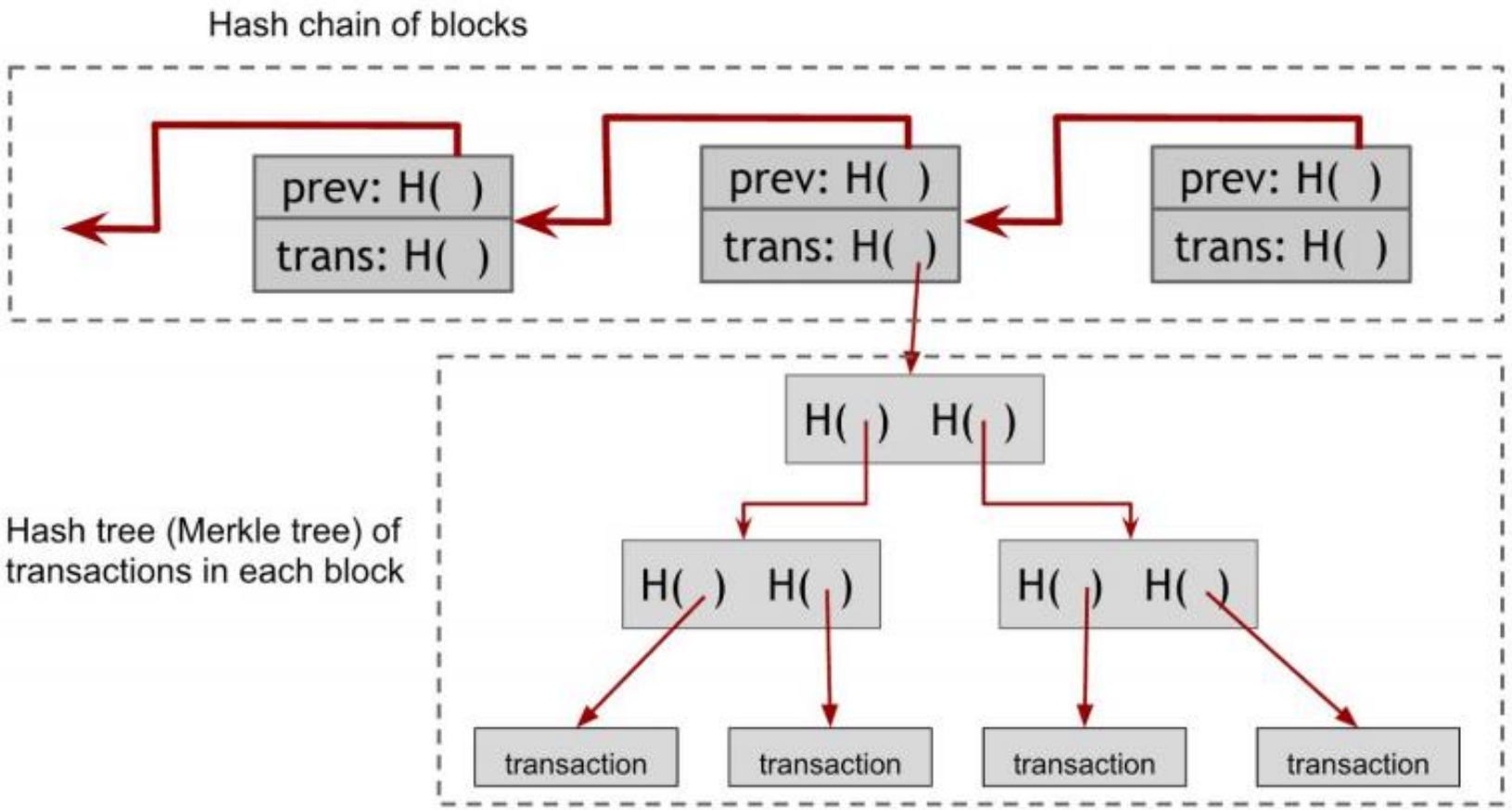
3

区块链1.0运行机制

The mechanics of block chain 1.0

3

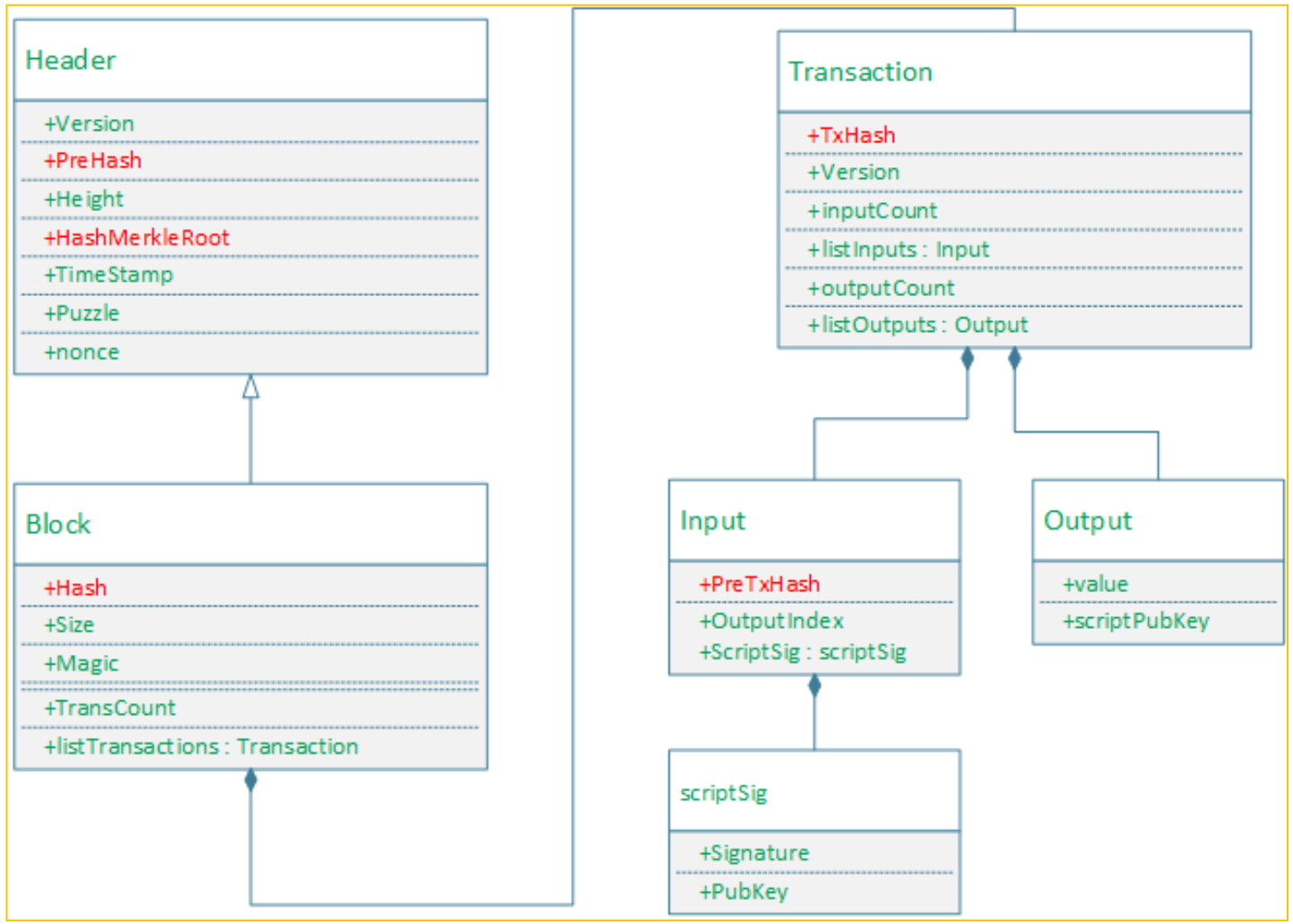
区块链1.0运行机制——Block的hash链结构



Block的hash链结构

- Block中包含上一个block的hash，将整个账本串成一个单链。
- Block中通过merkle tree 确保交易不被篡改。

3 区块链1.0运行机制——Block 数据结构

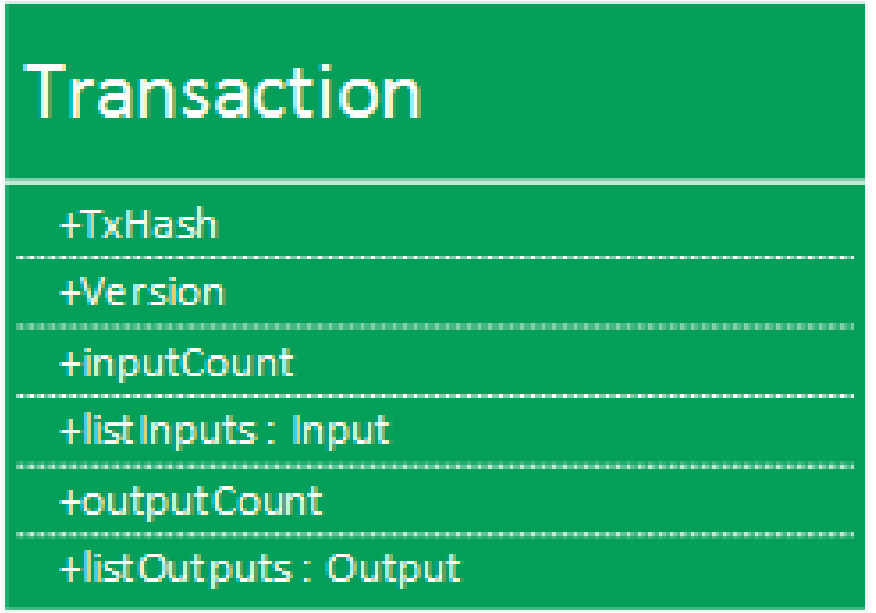


Block中的hash值

- PreHash，Header引入上一个block的hash值，这样形成一个链式结构。
- HashmerkleRoot，所有建议计算默克尔tree得到一个跟节点hash值，用来保证block中的交易不能被篡改。
- Hash，block的hash值，
- TxHash，交易的hash值。
- PreTxHash，已生成的交易hash值

3 区块链1.0运行机制—— Transaction数据结构

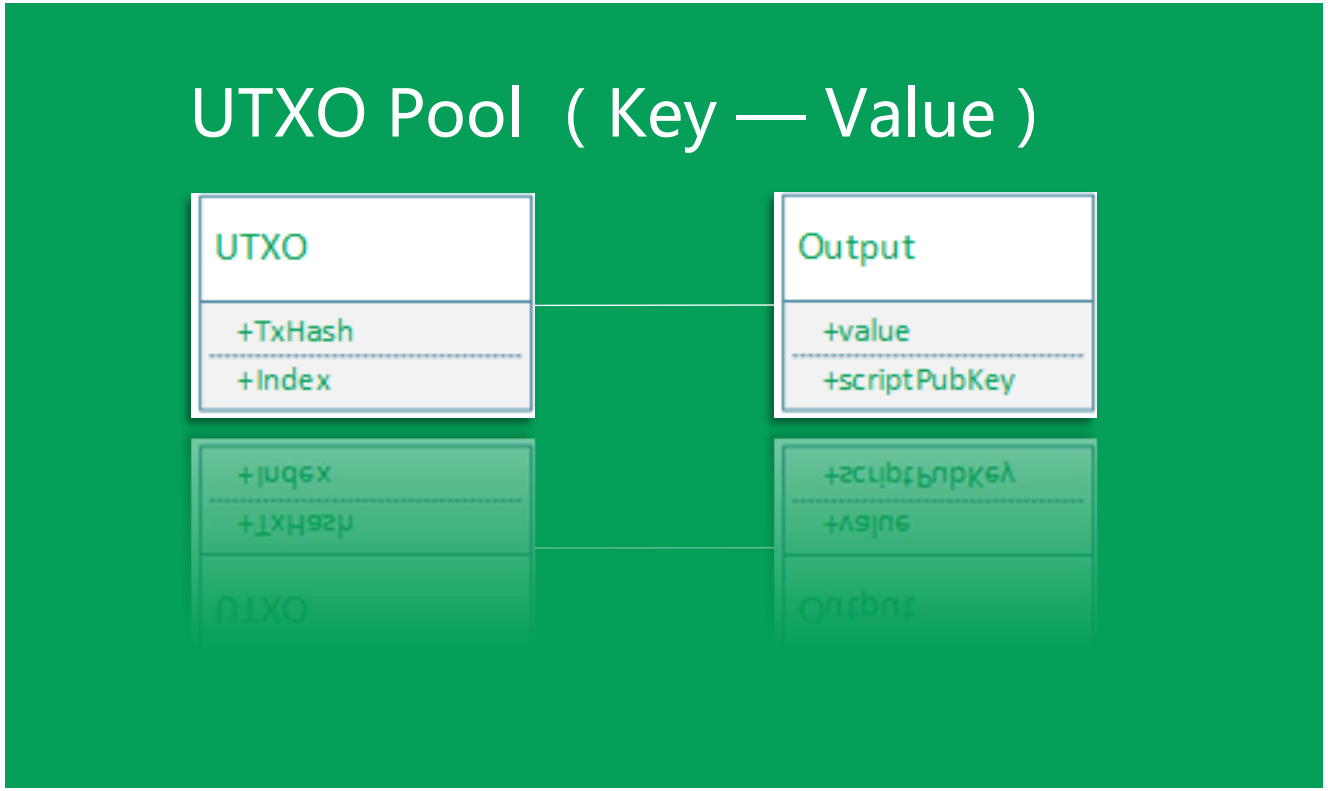

```
{
        {
        "TxHash": "55BF609F7C3CC037F7E91A8F41B4616B6B430F70BF528A14260720F378AEF031",
        "Version": "0.0.0.6",
        "inputCount": 1,
        "listInputs": [{
                "PreTxHash": "A1E933260EBE84F18C811AAC3EF9C3B146566BA1F26F3C114129006D1
                "OutputIndex": 0,
                "ScriptSig": {
                        "Signature": "A473C175C1917F46...",
                        "PubKey": "-----BEGIN RSA PUBLIC KEY...--END RSA PUBLIC KEY-----\r\n"
                }
        }],
        "outputCount": 2,
        "listOutputs": [{
                "value": 1.0,
                "scriptPubKey": "OP_DUP OP_HASH160 4F54AC9D55552C633360FEA08A4322EED29138E8025D582F9F6F58E5905FD63B OP_EQUALVERIFY OP_CHECKSIG"
        }]
}
```

3

区块链1.0运行机制—— UTXO

Unspent Transaction Output



UTXO不一样的账户模型

- UTXO 就是未消费的交易的输出，通过交易的hash值和output 索引就可以唯一确定。
- UTXO Pool 未花费交易输出表，一个Key-Value型的数据结构，遍历所有block**加入**所有交易的**输出**，**移除**交易**输**入，这样剩下的就是没有消费的交易输出。
- 在验证一笔交易是否有效时首先验证交易的input是否在UTXO Pool中，规避双花问题。
- 本无账户的概念，但是通过遍历UTXO Pool可以找出有你公钥的output，从而算出你的balance。

3 区块链1.0运行机制—— P2SH

Pay to script hash

● **OP_DUP OP_HASH160**

4F54AC9D55552C633360FEA08A4322EED29138E8025D582F9F6F58E5905FD63

OP_EQUALVERIFY OP_CHECKSIG

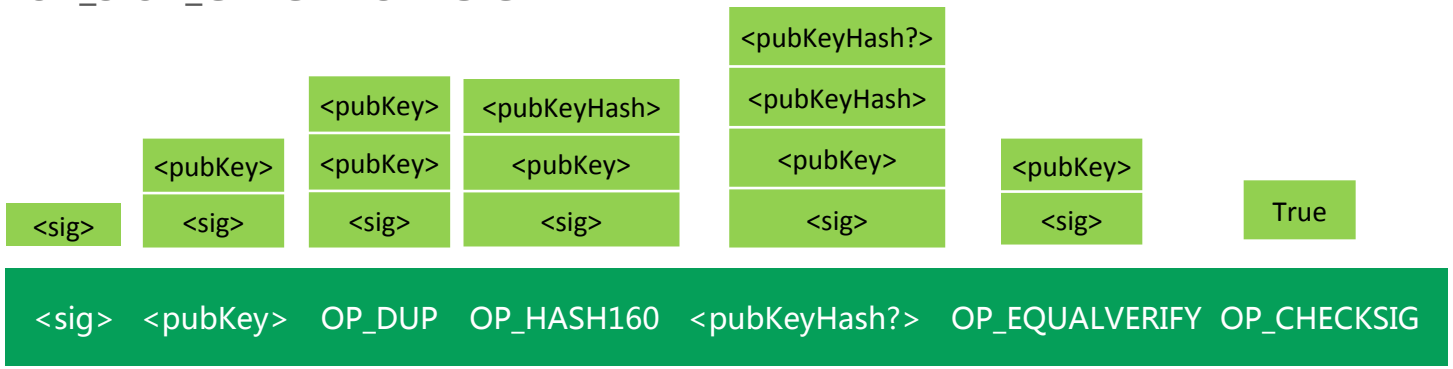
● **OP_2**

01314092DF4A3BDBA944E458CED99ED34A88ED4F0B9F179A5AEFC62017BB99

69C79662330838155EF3474300908FAC9F14D912AF52F5863E5143B551F5D869

4F54AC9D55552C633360FEA08A4322EED29138E8025D582F9F6F58E5905FD63

OP_3 OP_CHECKMULTISIG



Pay to script hash

● Pay to public key:“凭借公钥X的所有者的签名，才可以获得这笔资金”.

● Pay to script hash:” 凭借哈希值为X的公钥，以及这个公钥所有者的签名，才可以获得这笔资金。”

● 交易的output中的公钥变成了脚本，对于收款方可以通过写脚本实现更多的功能，比如第三方担保交易。

● 基于堆栈，图灵不完备，不能循环。

4

考虑一下怎么用区块链

Consider how to use blockchain

4 区块链作为一个只能添加的记录

著作、专利、你的idea手稿

- 算个hash
- 写到链上，根据区块的创建时间，就可以证明你什么时候就已经有这个想法了

4

运用脚本实现第三方担保交易

打赌

明天下雪，赌100块

Value： 100

Script： OP_2 A-pkHash B-pkHash C-pkHash OP_3 OP_CHECKMULTISIG

5

再往前走半步--智能合约

Go half step further – Smart contract

5 再往前走半步--智能合约

打赌　明天下雪，赌100块

Value： 100

Script： OP_2 A-pkHash B-pkHash C-pkHash

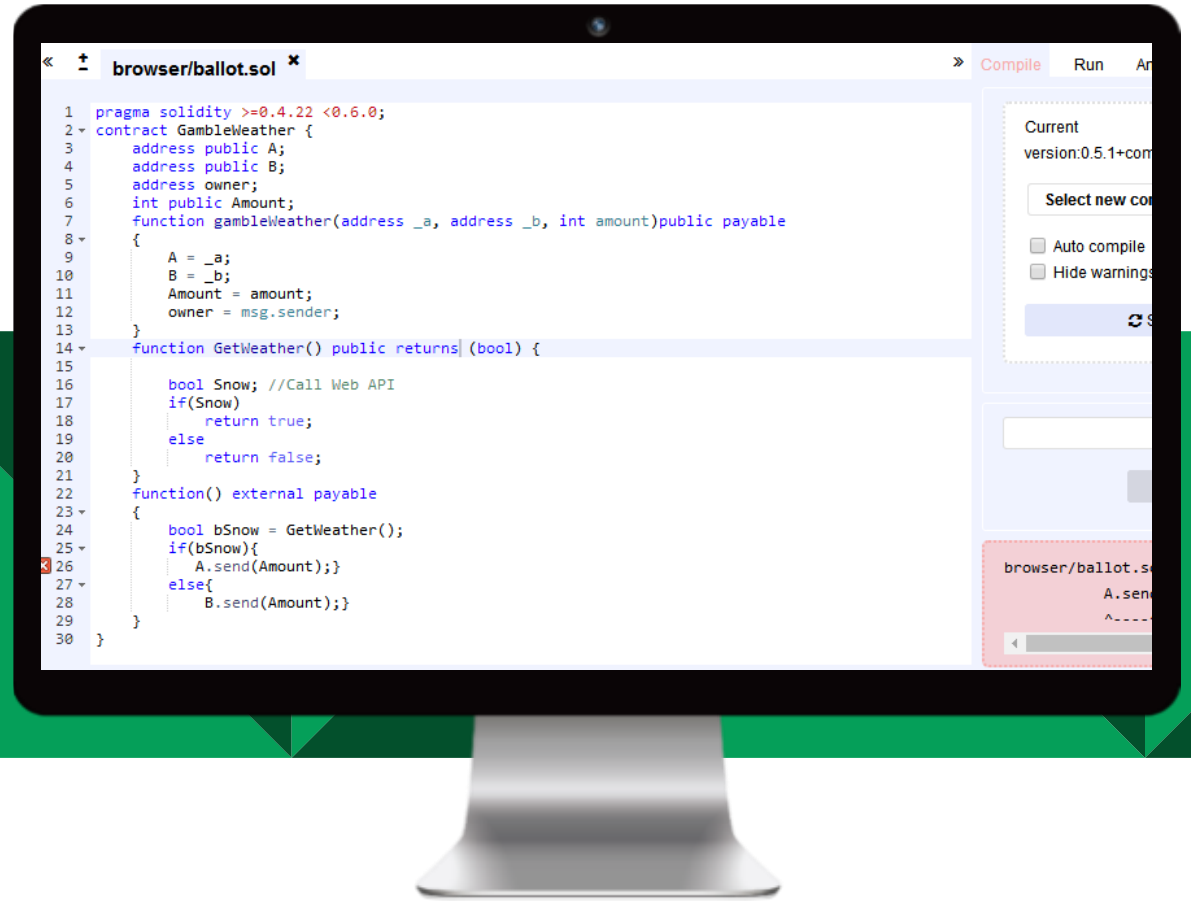
OP_3 OP_CHECKMULTISIG

脚本能不能明天中午12点直接去调气象局web API查询实时天气，

下雪将钱自动转给A，不下雪转给B ？

5

再往前走半步--智能合约



智能合约

- 智能合约可以简单理解为是一段写在区块链上的代码，一旦某个事件触发合约中的条款，代码即自动执行
- 图灵完备，可以循环
- 有循环了，那就可能有死循环，所以就得有机制防止死循环，如：以太坊Gas机制。
- 图灵完备了，理论上能解决任何算法，那原来一个Stack肯定实现不了，那就得有一个运行环境，如：以太坊EVM

5 智能合约是如何演变来的

区块链1.0

区块链1.5

区块链2.0

THANK YOU!

PRESENTED BY DAVID FAN