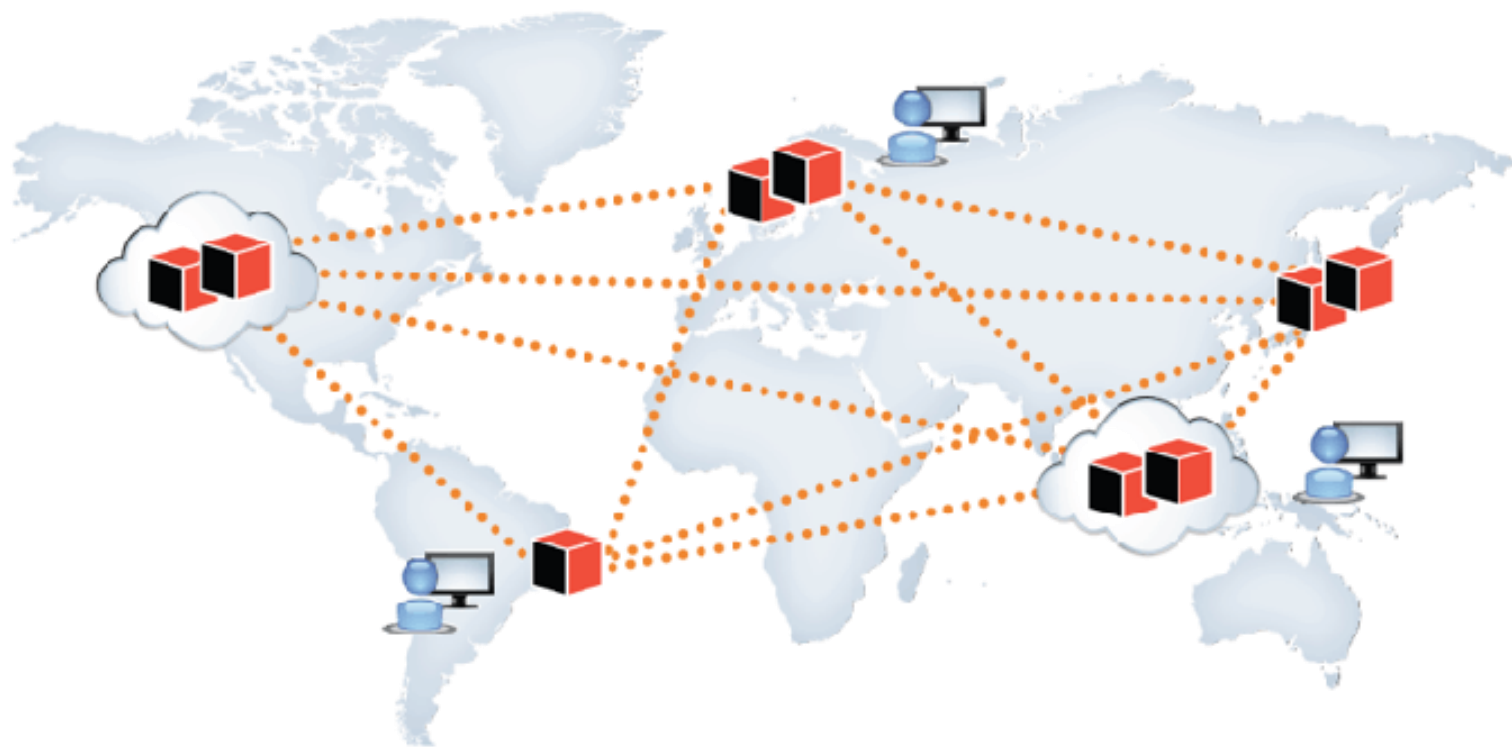
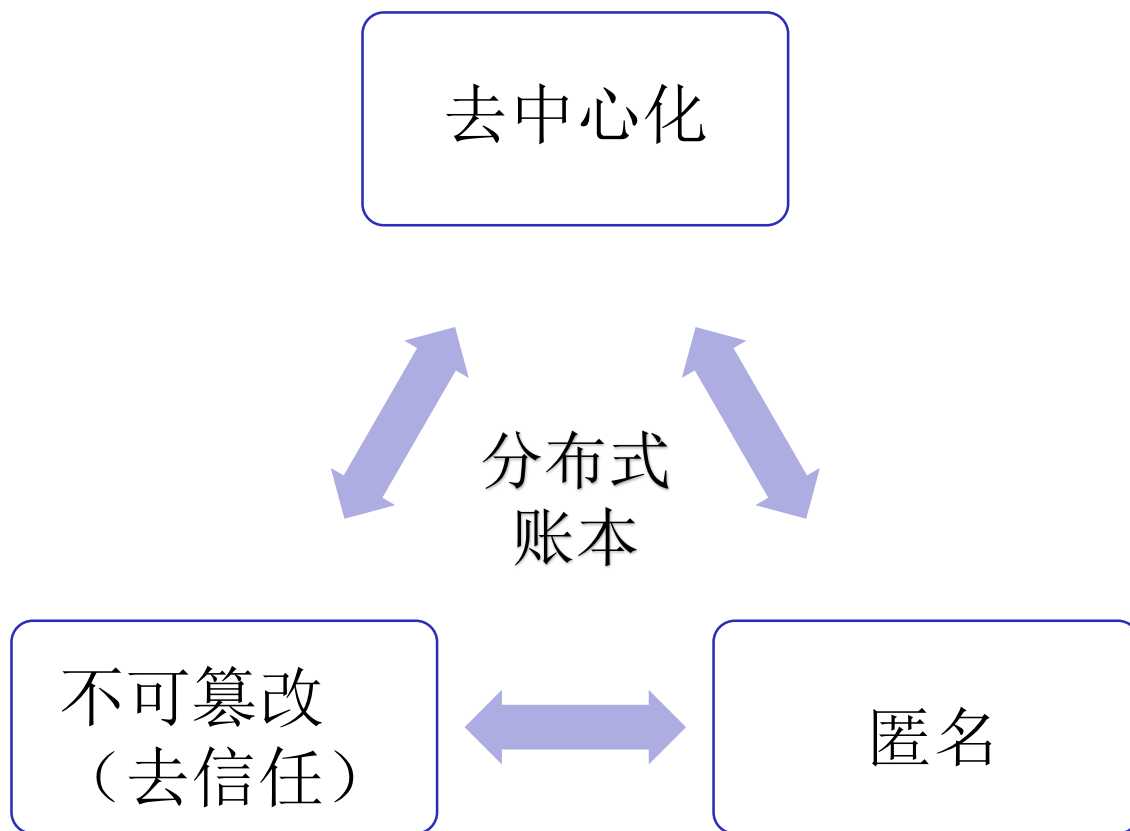


# 一、区块链技术



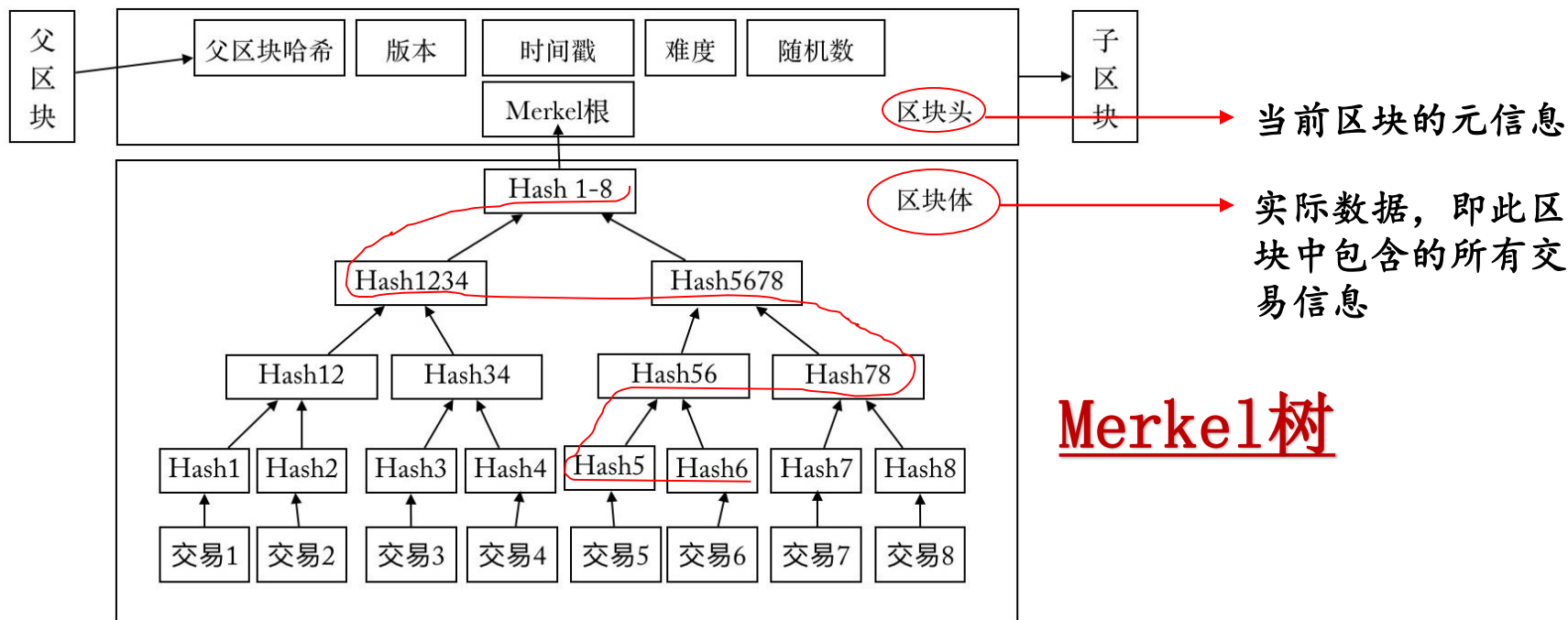
# 什么是区块链

🌈 本质：一种特殊的**分布式数据库**



# 什么是区块

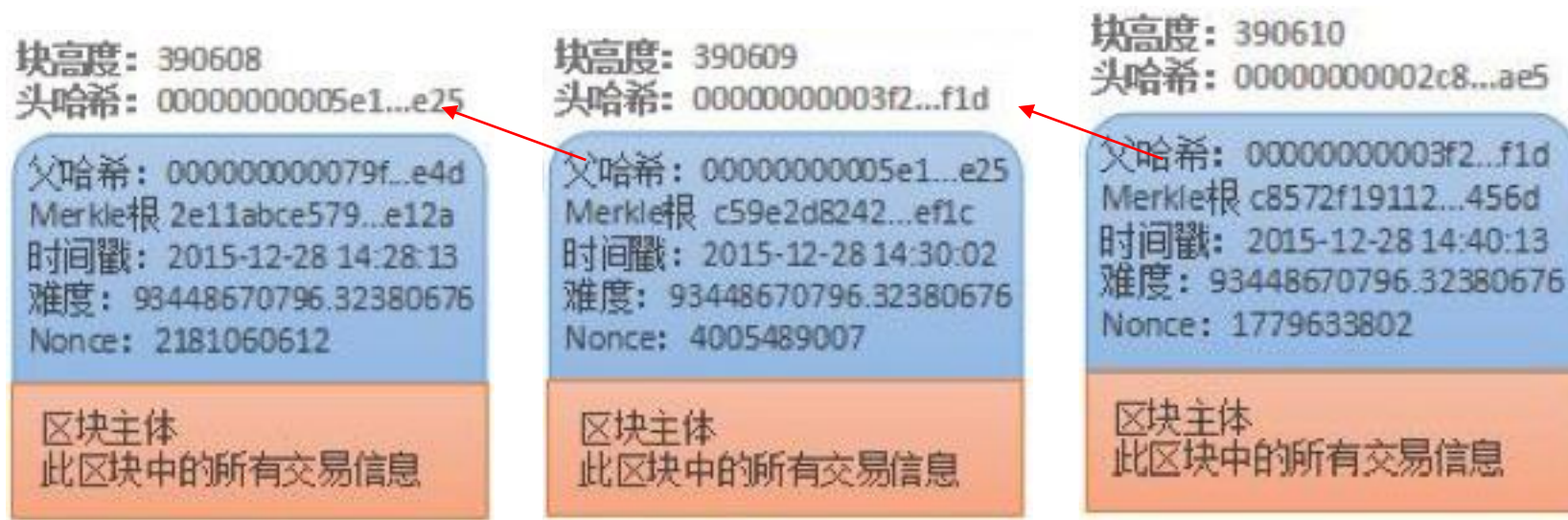
区块链由一个个**区块** (block) 组成，类比数据库的**记录**



如何验证交易6的存在性与正确性？

# 如何不可篡改

- 🌈 区块 Hash = SHA256（区块头）
- 🌈 每个区块的 Hash 都是不一样的，可以通过 Hash 标识区块
- 🌈 如果区块内容变了，或者上一个区块的哈希变了，它的 Hash 一定会变
- 🌈 某个区块被篡改，**必须依次修改后面所有区块**，短时间内修改多个区块几乎不可能发生（51%+）




# 为什么采矿很难

- 🌐 **采矿** (mining) —— 通过海量计算，得到当前区块的有效哈希，将新区块添加到区块链
- 🌐 **难度系数** (difficulty)：决定计算哈希的难度
- 🌐 为了将产出速率恒定在**10min**，难度系数每两周（2016个区块）调整一次
- 🌐 难度的调整是在每个完整节点中独立自动发生的，所有节点都会按统一的公式自动调整难度
- 🌐 9min → 10%↑； 11min → 10%↓

## Block #100000

BlockHash 000000000003ba27aa200b1cecaad478d2b00432346c3f1f3986da1afd33e506 

### Summary

Number Of Transactions	4
Height	100000 (Mainchain)
Block Reward	50 BTC
Timestamp	Dec 29, 2010 7:57:43 PM
Mined by	
Merkle Root	 f3e94742aca4b5ef85488dc37c06c3...
Previous Block	<a href="#">99999</a>

Difficulty	14484.16236122
Bits	1b04864c
Size (bytes)	957
Version	1
Nonce	274148111
Next Block	<a href="#">100001</a>

- 🌐 **目标值**  $\text{target} = \text{targetmax} / \text{difficulty}$ ，只有小于 target 的哈希才有效
- 🌐 **随机值** Nonce: 32位2进制，最大21.47亿，只能通过穷举法试错

# 如何去中心化

- 🌈 **拜占庭将军问题** —— Leslie Lamport 用来为描述分布式系统一致性问题（Distributed Consensus）在**论文**中抽象出来一个著名的例子
- 🌈 **N** 个将军（系统中的多个节点）被分隔在不同的地方，忠诚的将军希望通过某种协议达成某个命令的一致（进攻or后退）。但其中有 **F** 个背叛的将军（系统中节点出错）会努力向不同的将军发送不同的消息，试图干扰一致性的达成。在此情况下，如何让忠诚的将军们能达到行动的一致？

$$N \geq 3F + 1$$

## 🌈 新的解决思路

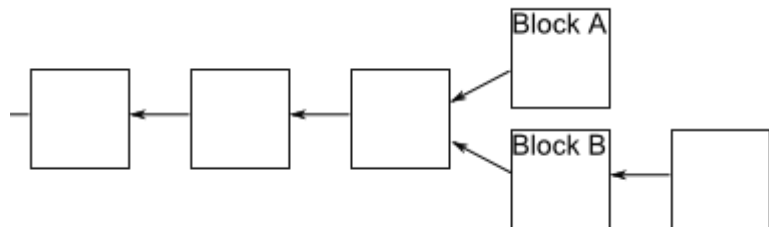
- 🌈 任何时候系统中都可能存在多个提案 → **增加发送信息的成本**，降低节点发送消息速率
- 🌈 完成最终的一致性确认过程十分困难，容易受干扰 → 放宽对最终一致性确认的需求，约定选择**最长链**

例子 3 (计算机C, D = X)

	得到的信息	结果
计算机A	O O X X	?
计算机B	O X X O	?
计算机C	X X O O	?
计算机D	X O O X	?

# 如何保障系统满足不同程度的一致性

- 两个人同时向区块链写入数据，形成分叉，该采纳哪个区块？



- 新节点总是采用最长的那条区块链，哪个分支在分叉点后面，先达到6个新区块，那条链就是有效的——**六次确认**



- 拥有大多数计算能力的那条分支，就是正宗的区块链

## 共识机制

- 工作量证明机制（PoW）
- 权益证明（PoS）

工作量证明机制

Proof of Work, POW

所有节点平等的计算一个数学难题，最先获得答案的节点获得区块的发布权。全网算力构成区块链系统的一道防火墙，抵抗黑客攻击。



# 为什么需要6次确认

- 当某个交易被写进block确认，并且该block后（包括自己）有6个block时，攻击节点想重新伪造block chain概率极低

## 11. Calculations

We consider the scenario of an attacker trying to generate an alternate chain faster than the honest chain. Even if this is accomplished, it does not throw the system open to arbitrary changes, such as creating value out of thin air or taking money that never belonged to the attacker. Nodes are not going to accept an invalid transaction as payment, and honest nodes will never accept a block containing them. An attacker can only try to change one of his own transactions to take back money he recently spent.

The race between the honest chain and an attacker chain can be characterized as a Binomial Random Walk. The success event is the honest chain being extended by one block, increasing its lead by +1, and the failure event is the attacker's chain being extended by one block, reducing the gap by -1.

The probability of an attacker catching up from a given deficit is analogous to a Gambler's Ruin problem. Suppose a gambler with unlimited credit starts at a deficit and plays potentially an infinite number of trials to try to reach breakeven. We can calculate the probability he ever reaches breakeven, or that an attacker ever catches up with the honest chain, as follows [8]:

$p$  = probability an honest node finds the next block

$q$  = probability the attacker finds the next block

$q_z$  = probability the attacker will ever catch up from  $z$  blocks behind

$$q_z = \begin{cases} 1 & \text{if } p \leq q \\ (q/p)^z & \text{if } p > q \end{cases}$$

$q=0.1$

$z=0$	$P=1.0000000$
$z=1$	$P=0.2045873$
$z=2$	$P=0.0509779$
$z=3$	$P=0.0131722$
$z=4$	$P=0.0034552$
$z=5$	$P=0.0009137$
$z=6$	$P=0.0002428$
$z=7$	$P=0.0000647$
$z=8$	$P=0.0000173$
$z=9$	$P=0.0000046$
$z=10$	$P=0.0000012$

$q=0.3$

$z=0$	$P=1.0000000$
$z=5$	$P=0.1773523$
$z=10$	$P=0.0416605$
$z=15$	$P=0.0101008$
$z=20$	$P=0.0024804$
$z=25$	$P=0.0006132$
$z=30$	$P=0.0001522$
$z=35$	$P=0.0000379$
$z=40$	$P=0.0000095$
$z=45$	$P=0.0000024$
$z=50$	$P=0.0000006$

Solving for  $P$  less than 0.1%...

$P < 0.001$

$q=0.10 \quad z=5$

$q=0.15 \quad z=8$



# 区块链的分叉—统一状态

🌐 网络中有一个统一的区块链视角，以蓝色区块为主链的“顶点”



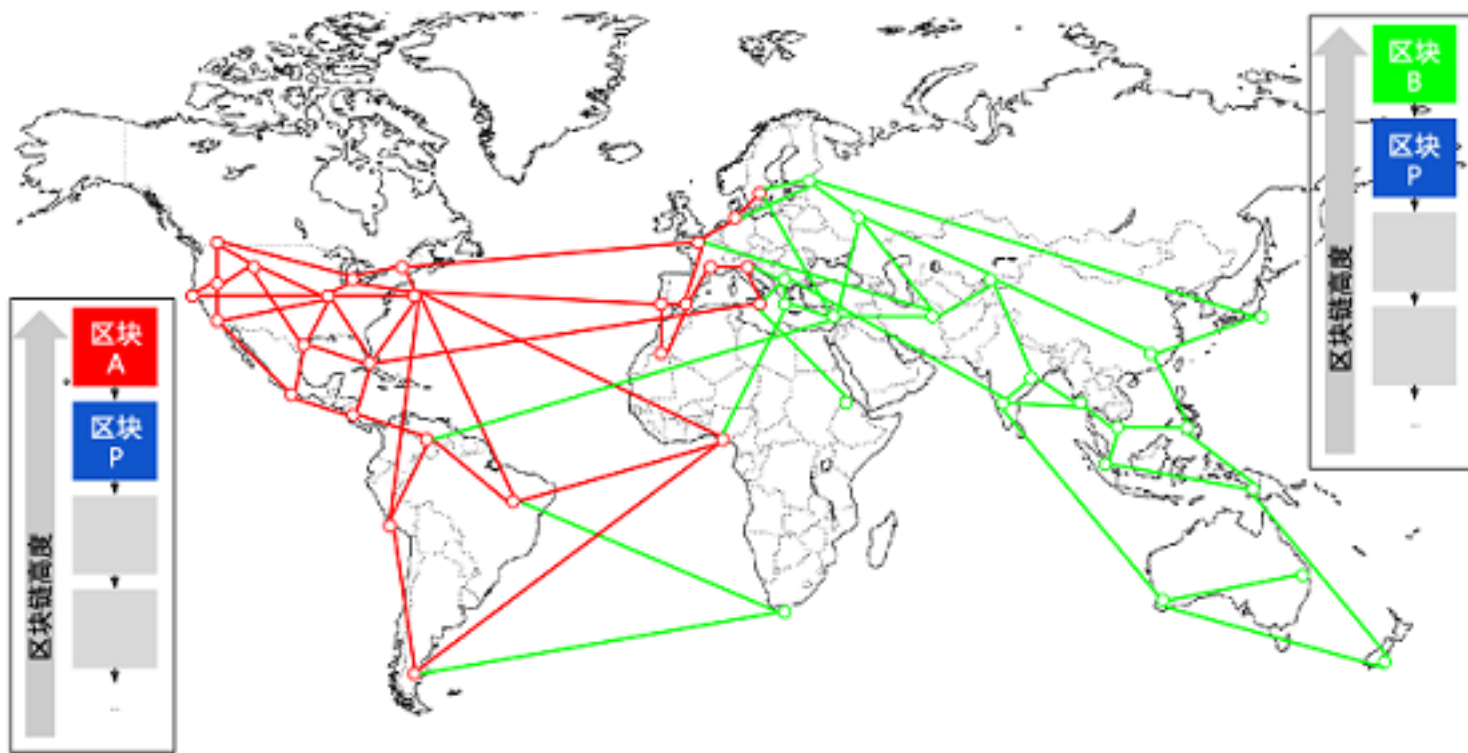
# 区块链的分叉—分叉前

- 两个矿工几乎**同时**挖到了两个不同的区块。这两个区块是顶点区块——蓝色区块的**子区块**，可以延长这个区块链



# 区块链的分叉—顶点分歧

- 当这两个区块传播时，一些节点首先收到“红色”区块，一些节点收到“绿色”区块
- 于是对于区块链的顶点产生了分歧，一派以红色区块为顶点，而另一派以绿色区块为顶点



# 区块链的分叉—分叉并切换主链

- 网络中的一部分算力专注于“红色”区块为父区块，在其之上建立新的区块；另一部分算力则专注在“绿色”区块上
- 工作在“绿色”区块上的矿工找到了一个“粉色”区块延长了区块链(蓝色-绿色-粉色)，他们会立刻传播这个新区块，整个网络都会认为这个区块是有效的



# 区块链的应用场景

区块链的代价：效率、能耗

适用场景

- 不存在所有成员都信任的管理当局
- 写入的数据不要求实时使用
- 挖矿的收益能够弥补本身的成本

TrustSQL

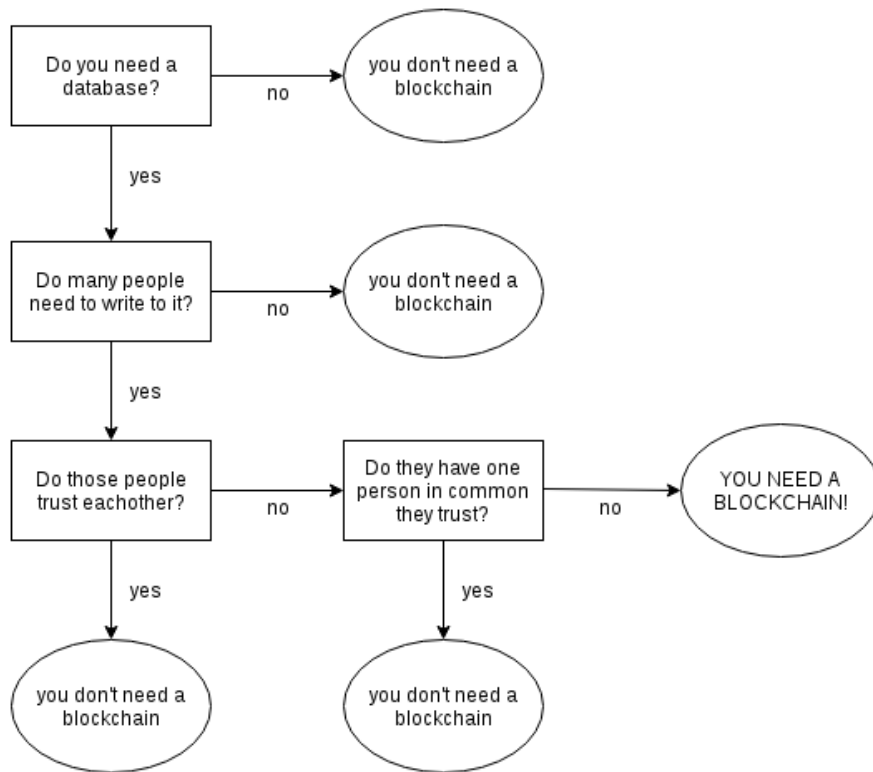


Ant Blockchain

爱心捐赠

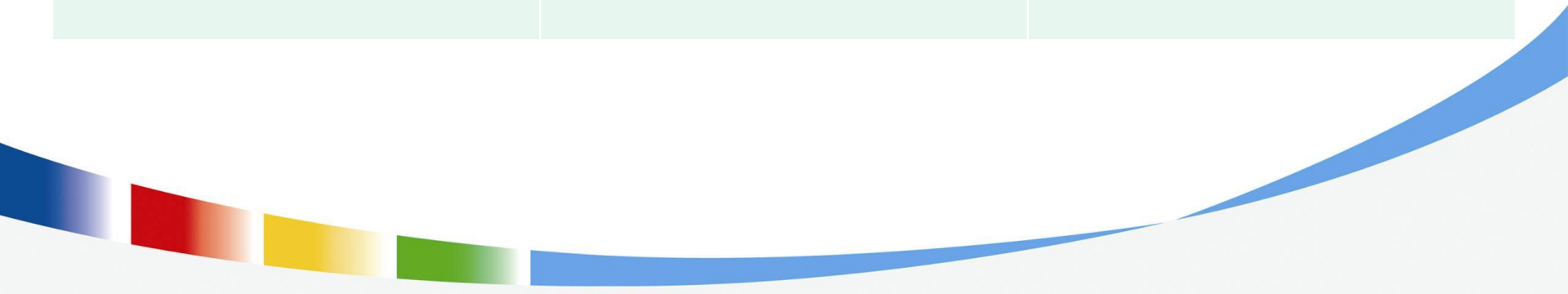
CAI NIAO  
菜鸟联盟

BaaS  
BlockChain as a Service



# 行业应用比较

主要应用领域	应用前	应用后
金融业（银行、支付转账、股票交易等）	流程复杂；中心化数据存储；第三方担保	简化流程；分布式存储，安全性提升；无需第三方，降低成本
网络安全	中心服务器存储数据、转移和传递	信息传播路径改变，不可拦截
身份信息管理	银行、信息卡身份识别过程繁琐；身份信息易被盗用	简化识别过程；加强身份信息
公证	需要政府、公信力第三方提供背书	数字加密作信用背书，自动完成公证；永久保存资料
投票	机票可能存在伪造；选民身份信息保护环节较弱	过程全网公开；选票可追溯；选民身份保密性好
供应链	低效、产品做假、低质量风险高	供应链各环节诚信保证高；产品信息可追溯，质量可保证





## 二、比特币原理



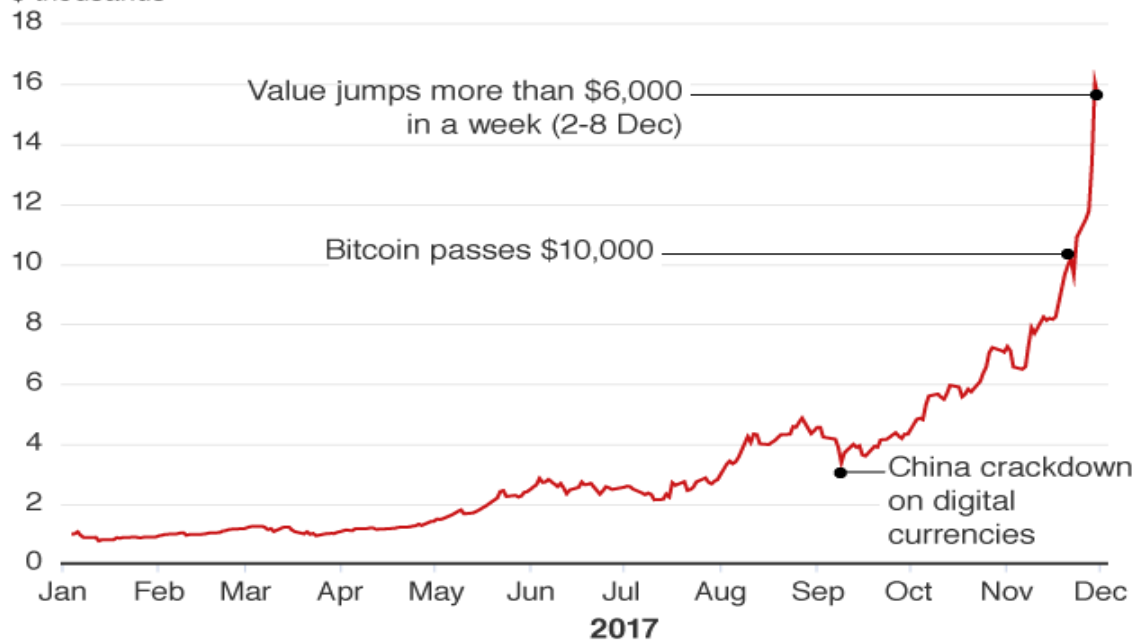
# 起源与发展

- 🌐 诞生于2008年的一篇论文：[\*Bitcoin: A Peer-to-Peer Electronic Cash System\*](#)
- 🌐 中本聪：让我们创造一种不受政府或其他任何人控制的货币！
- 🌐 2009/1 推出第一个比特币客户端，中本聪创建第一个区块，获得首批 50 比特币
- 🌐 暴涨出现在 2017/11 ~ 12，峰值接近 \$2w

## 2017: Bitcoin's unstoppable run

Bitcoin exchange rate with US dollar

\$ thousands

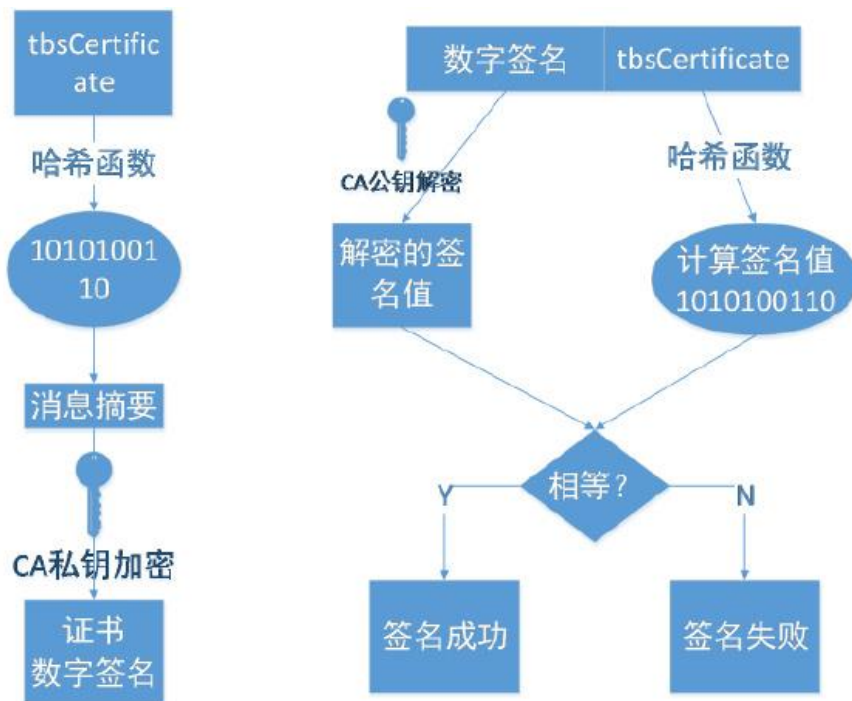
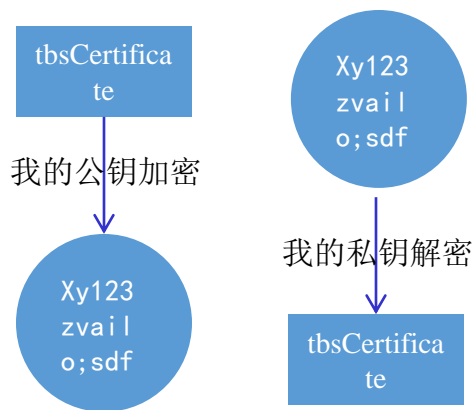


Source: Bloomberg. Data to 8 December, 10:15 GMT

BBC

# 加密学原理

## 非对称加密



非对称加密保证了支付的可靠性

# 比特币钱包




- 🌐 匿名性：钱不是支付给个人，而是给某一把**私钥**
- 🌐 比特币钱包：公钥+私钥
- 🌐 公钥（512位）不易传播，需生成**公钥指纹**（160位2进制，16进制大约26~35个字符）


S SERVICES LIMITED [GB] <https://blockchain.info/wallet/login>

**Blockchain** Home Charts Stats Markets Dev

## My Wallet Be Your Own Bank.

Wallet Home My Transactions Send Money Receive Money Import / Export

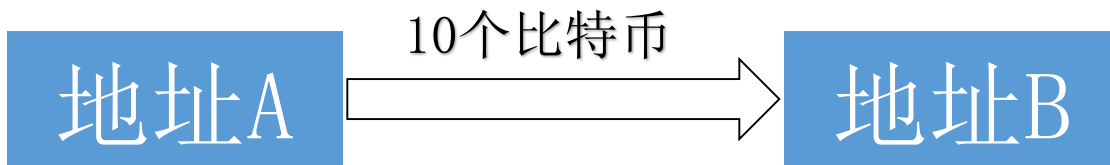
Total Transactions	825	
Total Received	231.44986483 BTC	
Total Sent	231.44852083 BTC	
Final Balance	0.001344 BTC	



This Is Your Bitcoin Address  
**16xTqmGebFBAZZMgyKAsykuefFaAXHMm1H**  
Share this with anyone and they can send you payments.



# 交易过程



🌐 A 是否拥有足够的比特币？

🌐 所有比特币的交易记录**公开可查**

🌐 如何证明申报人是地址 A 的主人，不是他人冒用？

🌐 申报人需提供以下数据

- 交易金额：10
- 上一笔交易的hash（你从哪里得到这些比特币）
- A地址、B地址
- 支付方公钥
- 支付方**私钥**生成的数字签名

🌐 验证这笔交易属实

找到上一笔交易 →  
确认A的比特币来源



计算支付方公钥指纹，  
确认与支付方地址一致 → 保证公钥属实



使用公钥解开数字签名 → 保证私钥属实

# 区块链的作用——交易确认

---

- ❶ 区块链作为数据库，记载了所有交易，充当**中央记账系统**的角色
- ❷ 交易确认：交易数据必须写入区块链，才算成立，对方才能真正收到钱
  - ❶ 比特币不存放在钱包或其他别的地方，而是只存在于**区块链**上面
  - ❶ 区块链记载了你参与的每一笔交易，由此算出来你拥有多少资产
- ❸ 一个区块最大1M，一笔交易约250B，一个区块最多包含4000多笔交易
- ❹ 矿工：把这2000多笔交易打包在一起，组成一个**区块**，然后计算这个区块的 Hash（采矿）
  - ❶ 收益（**流通比特币的来源**）：每四年减半，目前是12.5个，2140年停止奖励，收益完全靠交易手续费
  - ❶ 矿工们总是优先处理手续费高的交易
  - ❶ 交易一旦写入区块链，任何人都无法再修改



# 区块链的作用——交易确认

## Transactions

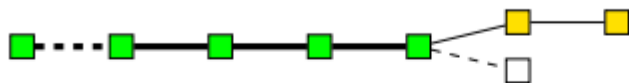
8c14f0db3df150123e6f3dbbf30f8b955a8249b62ac1d1ff16284aef3d06d87 ⓘ		mined Dec 29, 2010 7:57:43 PM	
No Inputs (Newly Generated Coins)		1HWqMzw1jfpXb3xyuUZ4uWXY4tqL2cW47J	50 BTC (5)
		411894 CONFIRMATIONS	50 BTC
fff2525b8931402dd09222c50775608f75787bd2b87e56995a7bdd30f79702c4 ⓘ		mined Dec 29, 2010 7:57:43 PM	
1BNwxHGafBeUBitpjy2AskpJ29Ybxtqvb	50 BTC	1JqDybm2nWTENrHvMyafb5XXtTk5Uv5QAn	5.56 BTC (5)
		1EYTGtG4LnFFiMvJdsU7GMGCQvsR5jYhx	44.44 BTC (5)
FEE: 0 BTC		411894 CONFIRMATIONS	50 BTC
6359f0868171b1d194cbee1af2f16ea598ae8fad666d9b012c8ed2b79a236ec4 ⓘ		mined Dec 29, 2010 7:57:43 PM	
15vScfMHNrXN4QvWe54q5hwfVoYwG79CS1	3 BTC	1H8ANdafjpqYntniT3Ddxh4xPBMCSz33pj	0.01 BTC (5)
		1Am9UTGfdnxabvcywYG2hvzr6qK8T3oUzt	2.99 BTC (5)
FEE: 0 BTC		411894 CONFIRMATIONS	3 BTC
e9a66845e05d5abc0ad04ec80f774a7e585c6e8db975962d069a522137b80c1d ⓘ		mined Dec 29, 2010 7:57:43 PM	
1JxDJCyWNakZ5kECKdCU9Zka6mh34mZ...	0.01 BTC	16FuTPaeRSPVxxCnwQmdyx2PQWxX6HWzhQ	0.01 BTC (5)
FEE: 0 BTC		411894 CONFIRMATIONS	0.01 BTC

# 如何解决重复支付问题

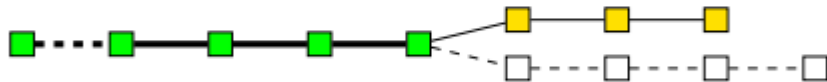
- 非对称加密保证交易不可能被伪造，但交易可能被复制：“A向B转移10个比特币”
  - A或C复制这句话写入区块链 → 由于**每笔交易公开可查**，第二次写入的时候查询区块链得知这笔钱已花掉，认定交易不合法，不能写入
  - 同一笔钱付给两个人：“A向B转移10个比特币” / “A向C转移10个比特币”
    - 同一矿工收到这两笔交易 → 选择一笔写入区块链
    - 两个矿工各收到一笔交易，同时分别写入区块链 → 分叉问题



(a) Initial state of the blockchain in which all transactions are considered as valid.



(b) Honest nodes continue extending the valid chain by putting yellow blocks, while the attacker secretly starts mining a fraudulent branch.



(c) The attacker succeeds in making the fraudulent branch longer than the honest one.

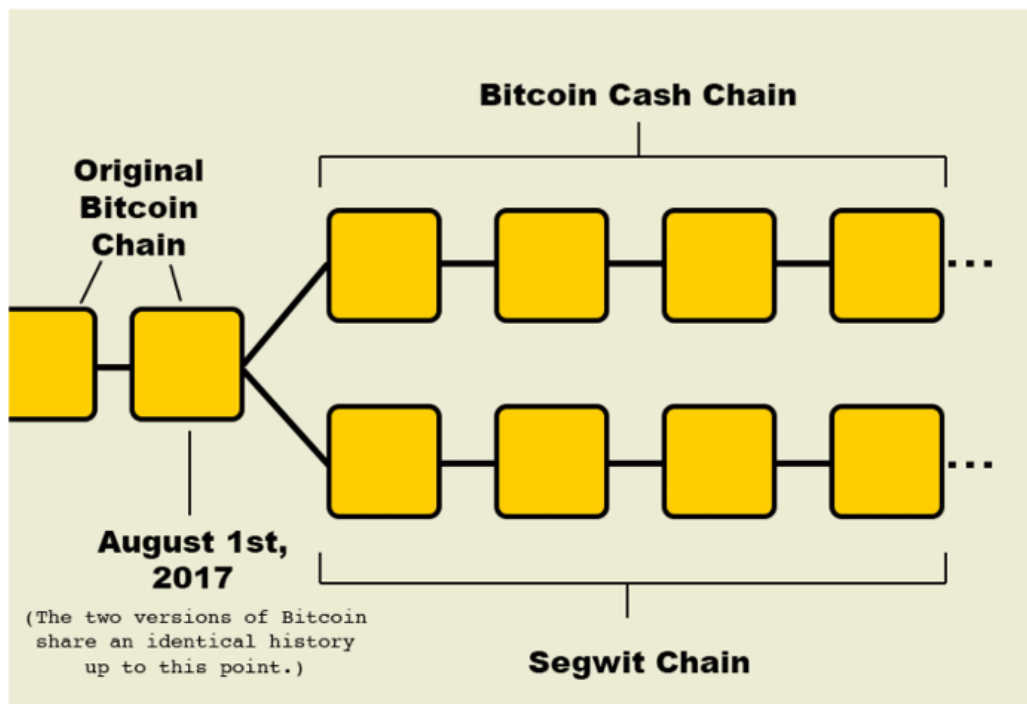


(d) The attacker's branch is published and is now considered the valid one.

# 区块的扩容

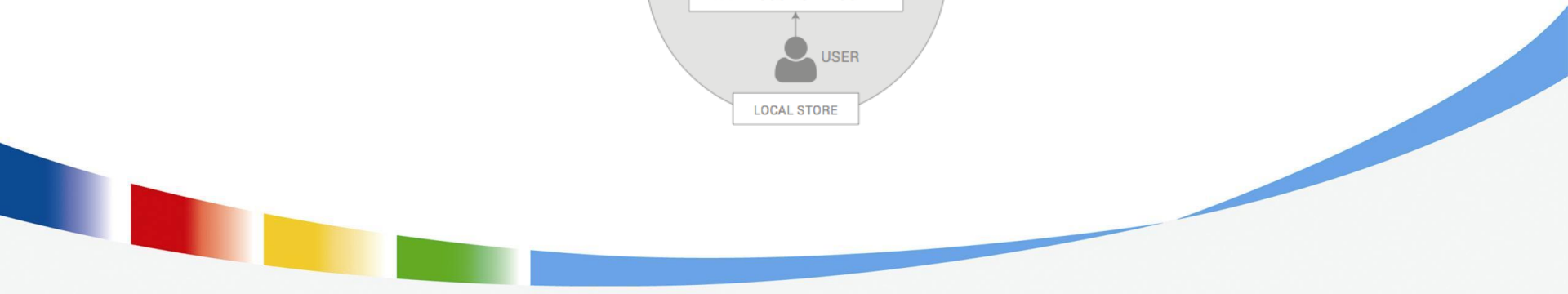
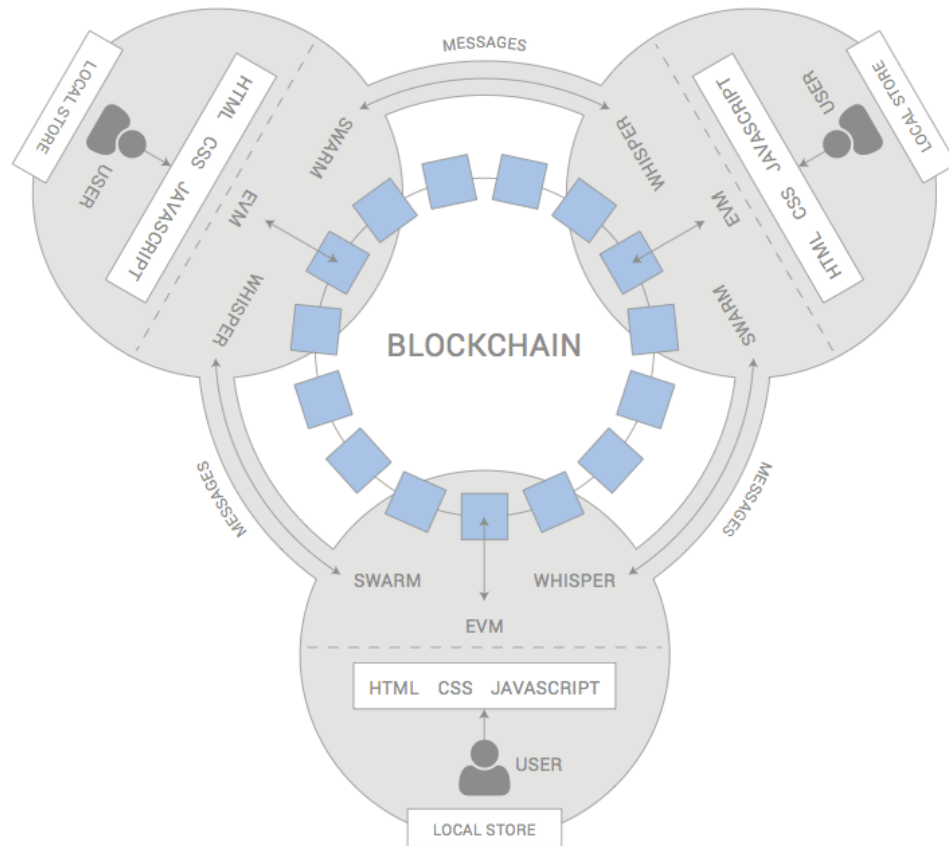
- 交易处理的速度 —— 6~7笔/s
- 2017/8: **Bitcoin Cash (BCH)**，与比特币区别，大小1M → 8M
  - 处理速度提升8倍，手续费降低
  - 对原有区块链的分叉，当时持有比特币的人，等于一人获赠了一份同样数量的 BCH
  - 本质是创造了一种新货币

DIAGRAM OF THE BITCOIN CASH FORK



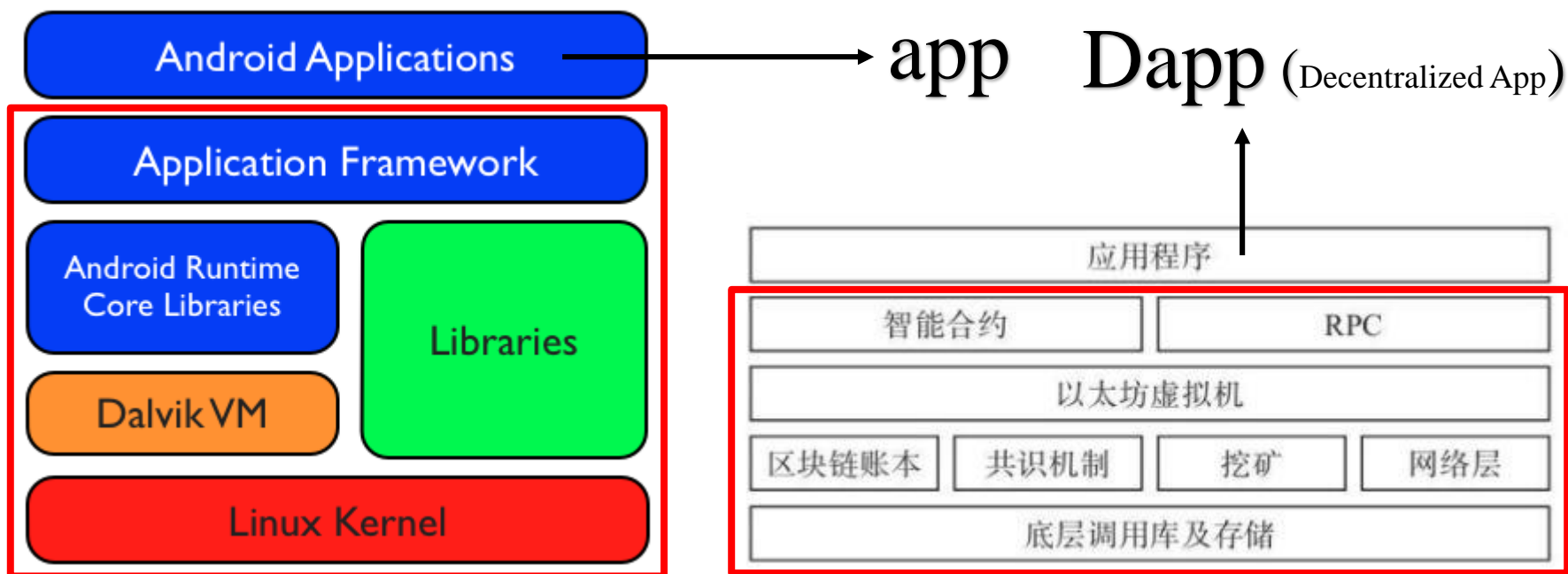
硬分叉

### 三、区块链应用开发



# 什么是以太坊（Ethereum）

- 本质：建立在区块链技术之上的去中心化**应用平台**
- 作用：允许任何人在平台中建立和使用通过区块链技术运行的**去中心化应用**
  - 以太坊平台对底层区块链技术进行了封装，开发者只要专注于应用本身的开发



“古典互联网”

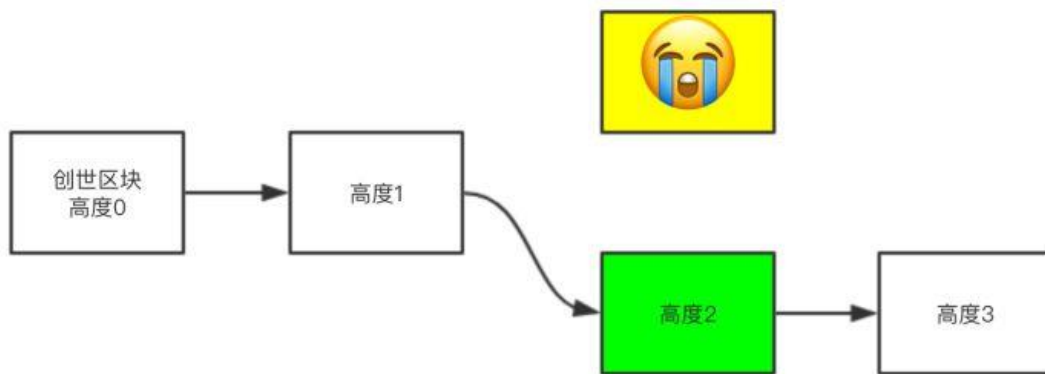
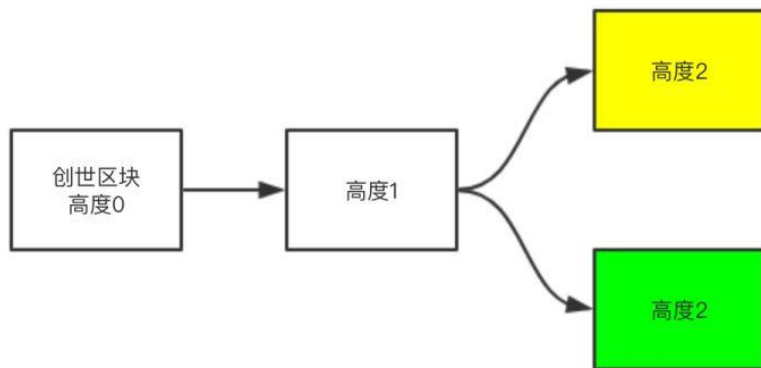


区块链互联网

# 以太坊的奖励机制

分叉

比特币（孤儿区块）

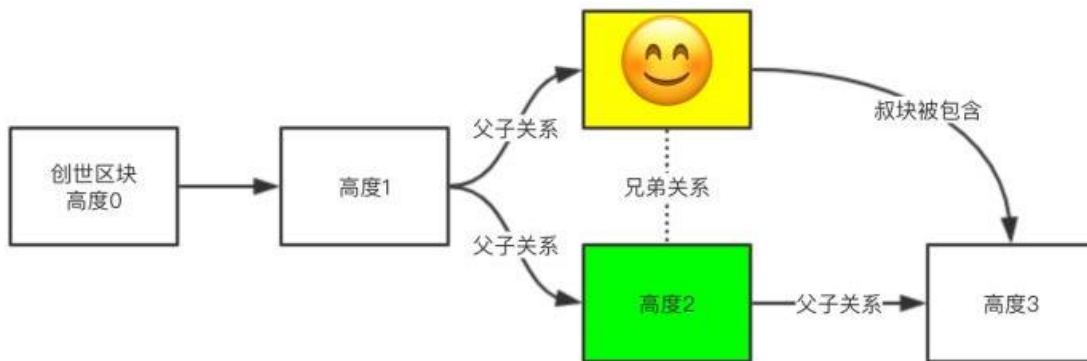
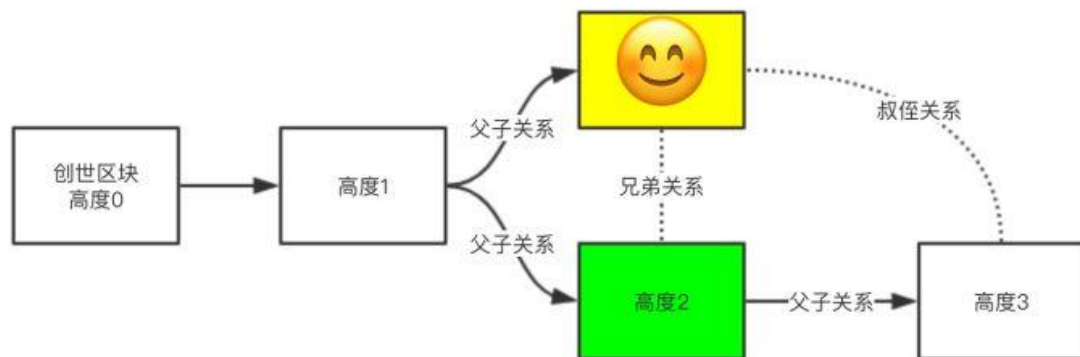




# 以太坊的奖励机制

分叉

以太坊（叔块）



叔块率

# 以太坊的奖励机制

---

## 普通区块奖励

固定奖励 5 ETH，每个普通区块都有

**注意：**随着协议升级，17年底每个区块的奖励由之前的5个 ETH 减少为3个

区块内包含的所有程序的 Gas 花费的总和

如果普通区块包含了叔块，每包含一个叔块可以得到固定奖励 5ETH 的1/32，也就是 0.15625ETH

## 叔块奖励

$$(\text{叔块高度} + 8 - \text{包含叔块的区块的高度}) * \text{普通区块奖励} / 8$$

## 以太坊区块浏览器

# 以太坊的奖励机制

## 以太坊区块浏览器

### Block Information

Height:	<a href="#">&lt; Prev</a> <b>4222300</b> <a href="#">Next &gt;</a>
TimeStamp:	4 mins ago (Aug-31-2017 05:05:31 AM +UTC)
Transactions:	<a href="#">107 transactions</a> and <a href="#">56 contract internal transactions</a> in this block
Hash:	0xfe8c1080bfa54fc8396f739b73e7d47f9f1ad947497ab191a836d0107edfa75e
Parent Hash:	<a href="#">0x479ea5613fdc054e5a98b8da682b4e880c6d73a7a1277645812dd0493f3fd621</a>
Sha3Uncles:	0xc31bbd9e8088f3c7c596d15d3ffd431609875970318af96b6dac3647d2fc65b6
Mined By:	<a href="#">0x829bd824b016326a401d083b33d092293333a830</a> in 75 secs
Difficulty:	2,255,032,776,672,791
Total Difficulty:	813,558,265,078,432,542,096
Size:	26592 bytes
Gas Limit:	6,712,390
Gas Used:	6,697,815
Nonce:	0x883206036c1fabd23b
Block Reward:	5.594337168043699381 Ether (5 + 0.281837168043699381 + 0.3125)
Uncles Reward:	8.75 Ether (2 Uncles at <a href="#">Position 0</a> , <a href="#">Position 1</a> )
Extra Data:	ä_fâ½©çŸžä»™é±¼ (Hex:0xe4b883e5bda9e7a59ee4bb99e9b1bc)

固定奖励

Gas花费(交易费)

将两个叔块包含的奖励

- 固定奖励: 5ETH
- Gas总花费(交易费):  
0.281837168043699381ETH
- 将两个叔块包含进来的奖励:  $5 * (1 / 32) * 2 = 0.3125\text{ETH}$

# 以太坊的奖励机制

## Overview

### Uncle Information

Uncle Height:	4222271 叔块高度
Uncle Position:	0
Block Height:	4222272 包含叔块的区块的高度
Hash:	0x1c2cbba0403f1079dcd70e5971a87ce0fbc03d4572be30e2d17e4e4a0f136d5
Parent Hash:	0x0dfe11b91ccb68294a2b60ed574398b979673fb888c3fe2bc0cbbff1175d3e82
Sha3Uncles:	0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347
Mined By:	0x829bd824b016326a401d083b33d092293333a830 in 19 secs
Difficulty:	2,258,524,587,473,917
Gas Limit:	6,735,996 Wei
Gas Used:	3,846,939 Wei
TimeStamp:	32 mins ago (8/31/2017 4:53:07 AM)
Uncle Reward:	4.375 Ether

$$(4222271 + 8 - 4222272) * 5 / 8 = 4.375\text{ETH}$$

# 以太坊 VS 比特币

以太坊的区块生成时间更短

平均 14s

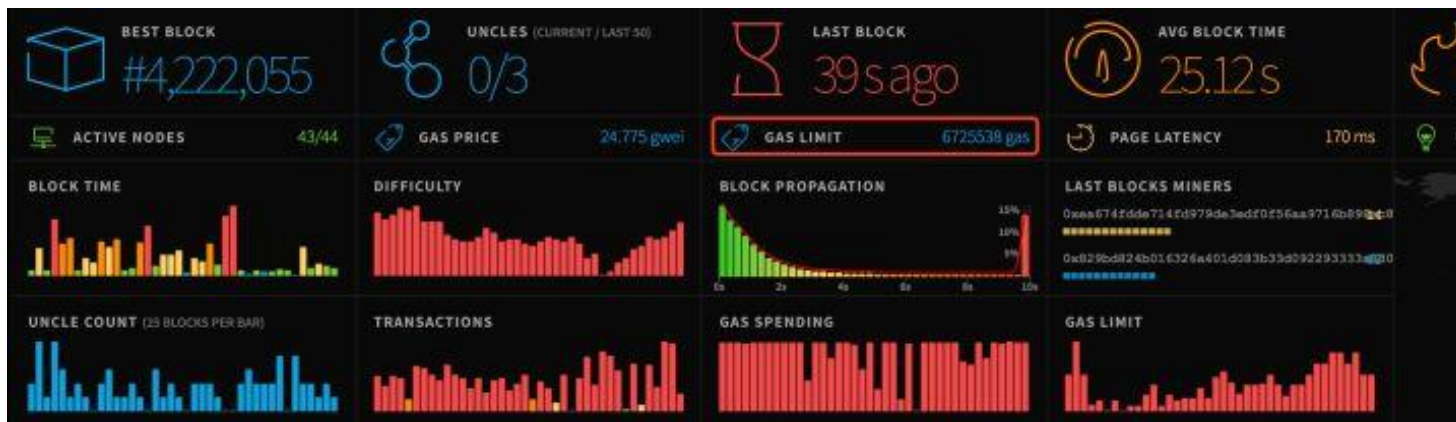
以太坊区块更小

以太坊区块大小根据在上面运行的智能合约的复杂性决定 — Gas限制

**目前**以太坊中最大区块大小大约为 1500000Gas

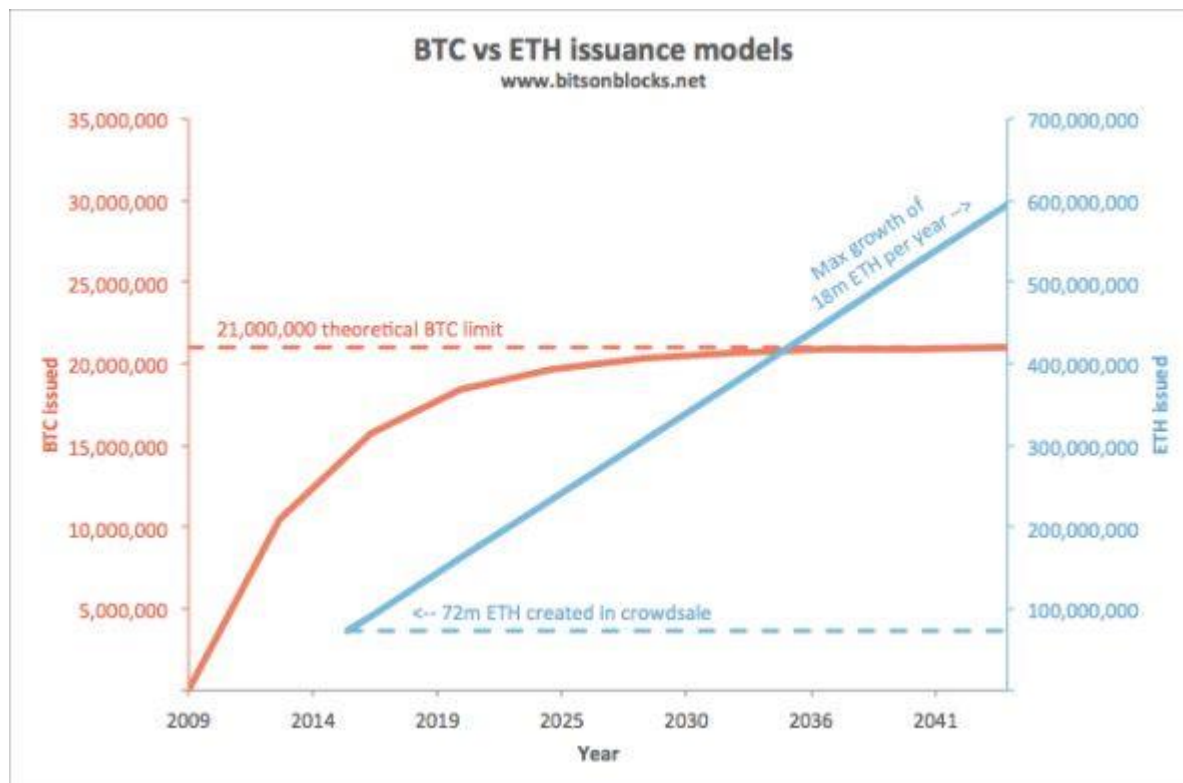
一个交易大约消耗 21000Gas

每个区块中大概可以放进 70 (1500000/21000) 笔交易



# 以太代币的发行

- 比特币：每四年减半
- 以太币：每年固定（最多 1800 万）





# 以太代币的发行

---

- Pre-mine (矿前) + Block rewards (区块奖励) + Uncle rewards (叔块奖励) + Uncle referencing rewards (叔块引用奖励)
- 矿前: 2014.7 - 8月众筹 7200 万个, 之后每年产量为1/4
- 区块奖励:
  - 每年产生  $365 * 24 * 60 * 60 / 14 = 225w$  区块
  - $225w * 5 = 1130w$  以太币
- 叔块奖励:
  - $500$  (目前每天产生叔块)  $* (5 * 7 / 8) * 365 = 70w$
- 叔块引用奖励

# Dapp开发前须知

- 智能合约（Smart Contract）：在区块链上可以自动执行的（由消息驱动的）、以**代码形式编写**的合同（特殊的交易）
  - 适合对**信任**、**安全**和**持久**性要求较高的应用场景
  - 数字货币**、数字资产、投票、保险、金融应用、预测市场、产权所有权管理、物联网、点对点交易



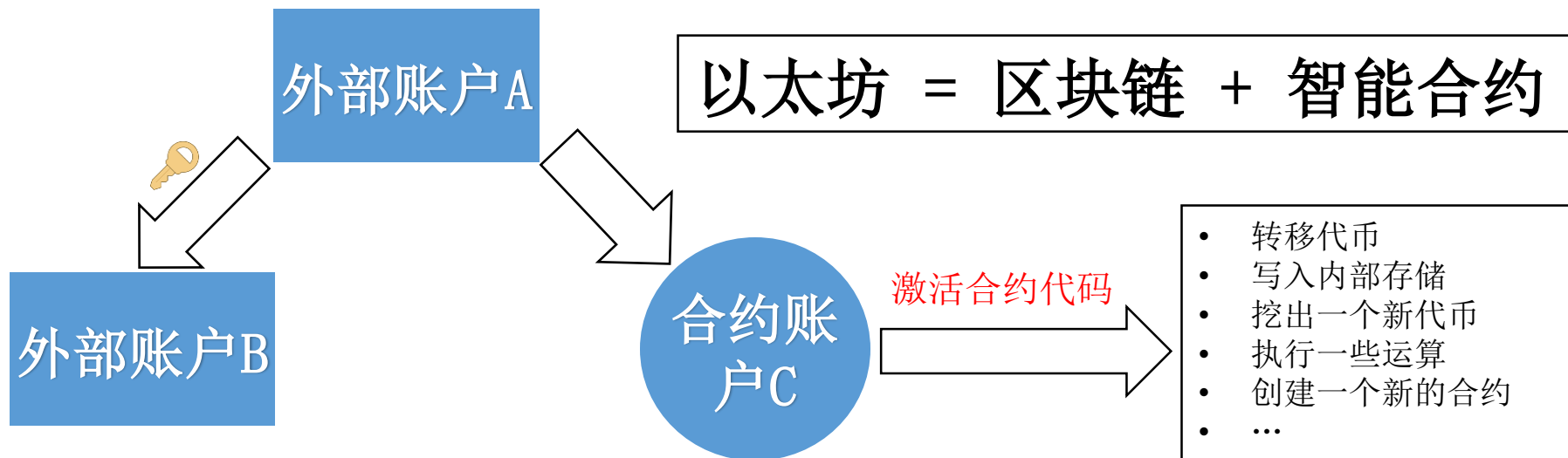
- 编程语言：Solidity，生成 .sol 文件，与 JS 类似，用它开发合约并编译成 EVM 字节代码（**Browser-Solidity**）
- 运行环境：EVM，运行在以太坊节点上，是一个隔离的环境
- 合约编译：**Browser-Solidity Web IDE**、**solc编译器**
- 合约部署：
  - 以太坊客户端（钱包）：开发者工具，提供账户管理、挖矿、转账、智能合约的部署和执行等功能，EVM是由以太坊客户端提供的（**Geth**、**Mist**）
  - 把合约字节码发布到区块链上，并使用一个特定的地址来标示这个合约，这个地址称为**合约账户**

# 以太坊是如何工作的

以太坊的基础单位是**账户**

外部账户（Externally Owned Accounts）：由人（**私钥**）控制，不关联代码

合约账户（Contract Accounts）：由**合约代码**控制，只能由外部账户激活



合约运行：向这个**合约账户**发送消息（交易），通过消息触发后智能合约的代码就会在EVM中执行了。

Gas：计费机制，普通交易、合约的部署和运行需要支付费用（Gas价格（以太币）\*Gas数量）→ 避免有人写出死循环的合约阻塞网络

# 总结

---

- 🌈 Dapp (Decentralized App) : 基于智能合约的去中心化的应用程序
  - 🌈 区块链 —— 不可篡改的数据库, 智能合约 —— 和数据库打交道的程序
  - 🌈 Dapp = 智能合约 + GUI
- 🌈 Dapp 开发
  - 🌈 基于以太坊平台, 方便地使用区块链技术开发去中心化的应用
  - 🌈 使用 **Solidity** 来编写和区块链交互的智能合约
  - 🌈 合约编写好后之后, 需要用以太坊客户端用一个有余额的账户去部署及运行合约
  - 🌈 为了开发方便, 我们可以用 **Geth** 或 **testrpc** 来搭建一个测试网络

# 附录

---

## 官网

 [比特币](#)

 [以太坊](#)

## 资料合集

 [区块链技术指南](#)

 [深入浅出区块链](#)

## 文章

 [区块链入门教程](#)

 [比特币入门教程](#)

 [加密货币的本质](#)

## Dapp开发实践教程

 [萌新带你入坑区块链](#)

 [以太坊Dapp开发初探](#)

