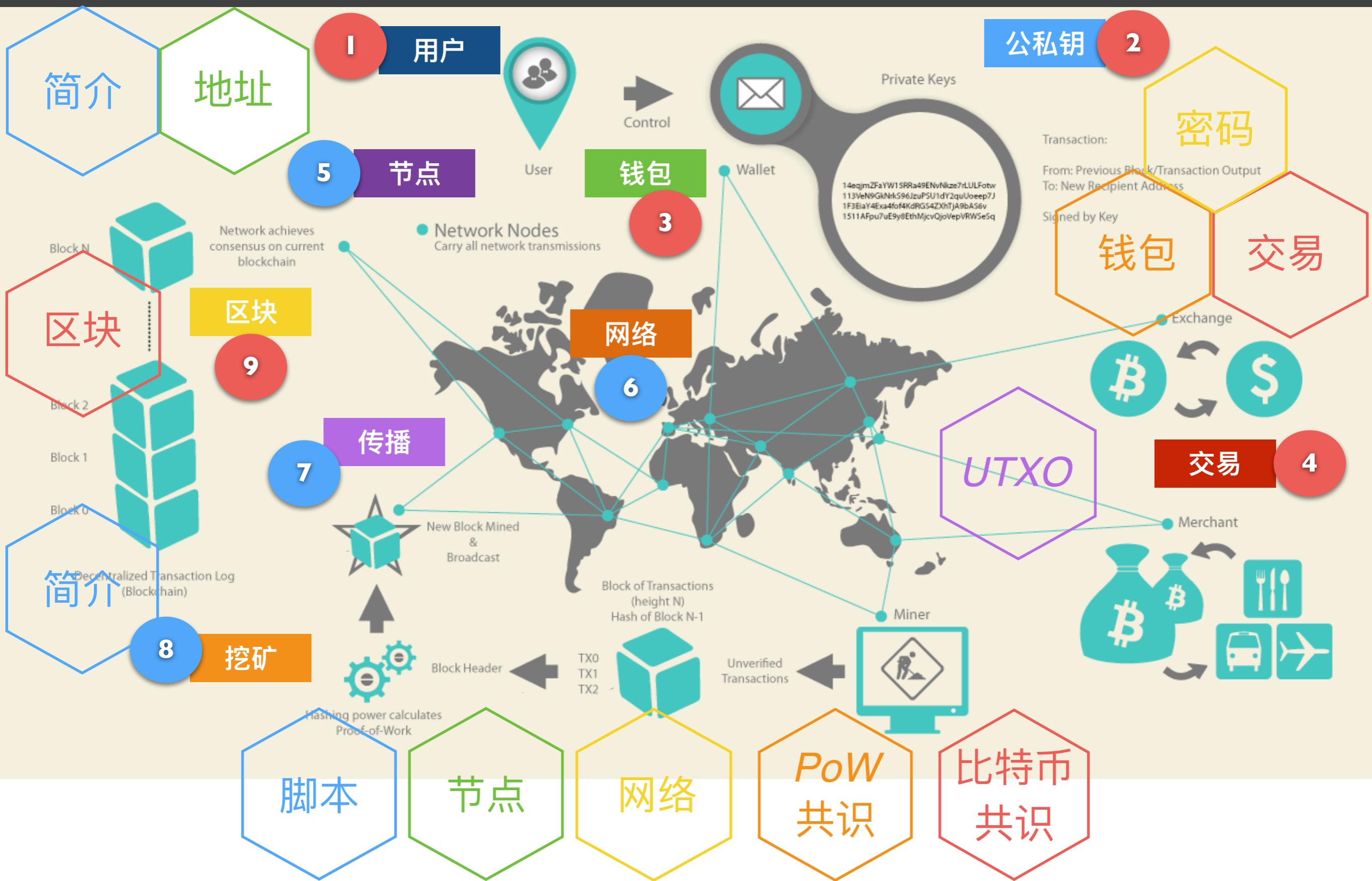




2023.03.14

# 比特币 -03

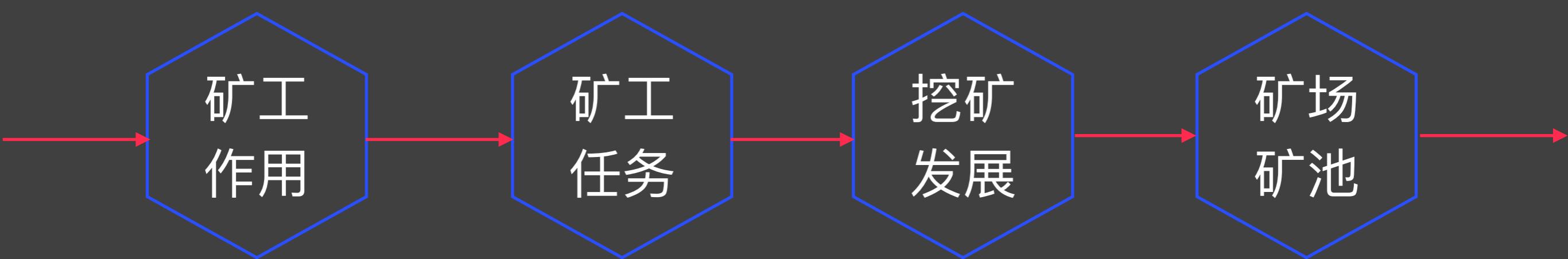




# 本次课程内容



# 比特币挖矿



- 比特币需要矿工
  - \* 存储和广播区块
  - \* 验证交易有效性
  - \* 对区块进行共识投票



但为什么成为一个矿工！

- 监听交易广播
- 维护区块链网络和监听新的区块
- 组装一个备选区块
- 找到一个让你的区块有效的随机数
- 希望你的区块被全网接受
- 利润

验证交易和区块 vs. 和其余矿工竞争

# 挖矿发展



CPU



GPU



FPGA



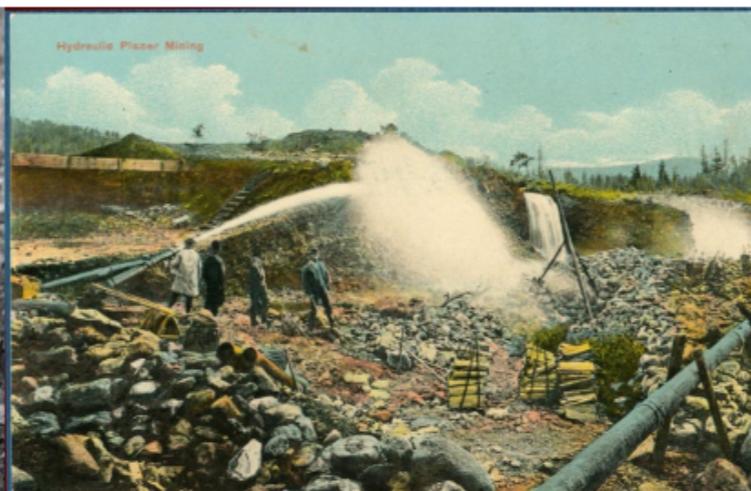
ASIC



gold pan



sluice box

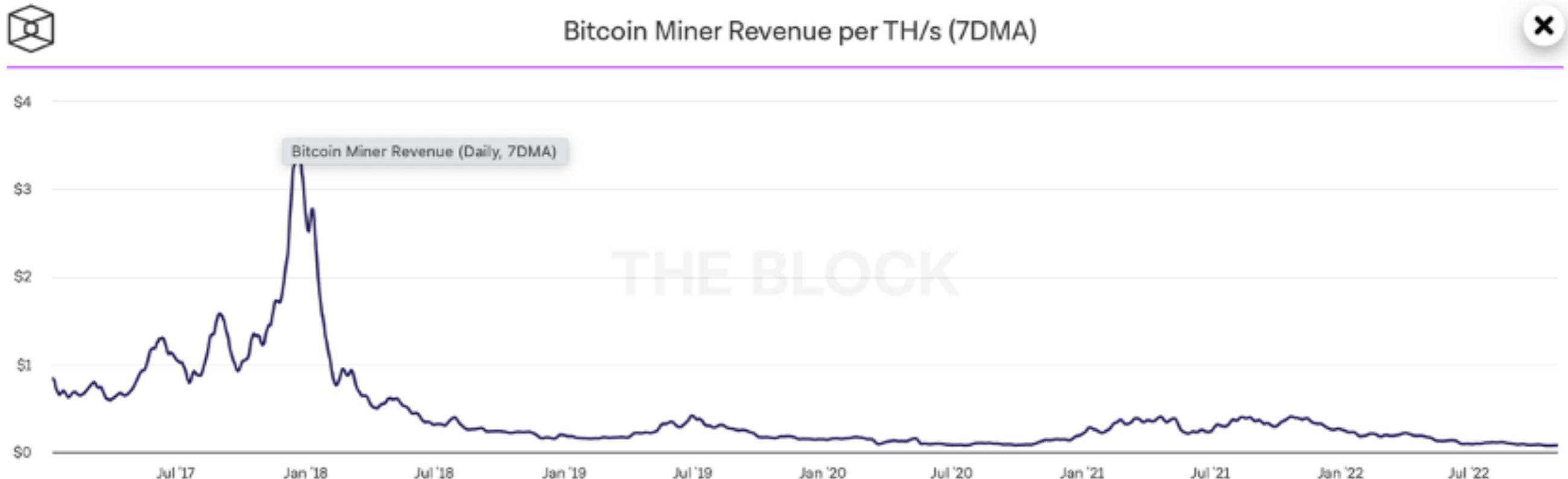


placer mining



pit mining

## 挖矿激励



为了保证所有节点按照比特币协议进行，比特币系统设计了如下的经济激励手段。当矿工挖到一个区块时，将得到相当可观的奖励（当前为30万美元一个区块）。矿工正常生成区块可以获得区块奖励加上该区块中所有交易的手续费，区块奖励每4年减半，保障了比特币总数2100万个，避免通货膨胀。

Bitcoin-02

## 专业矿场



温度

电费

网速

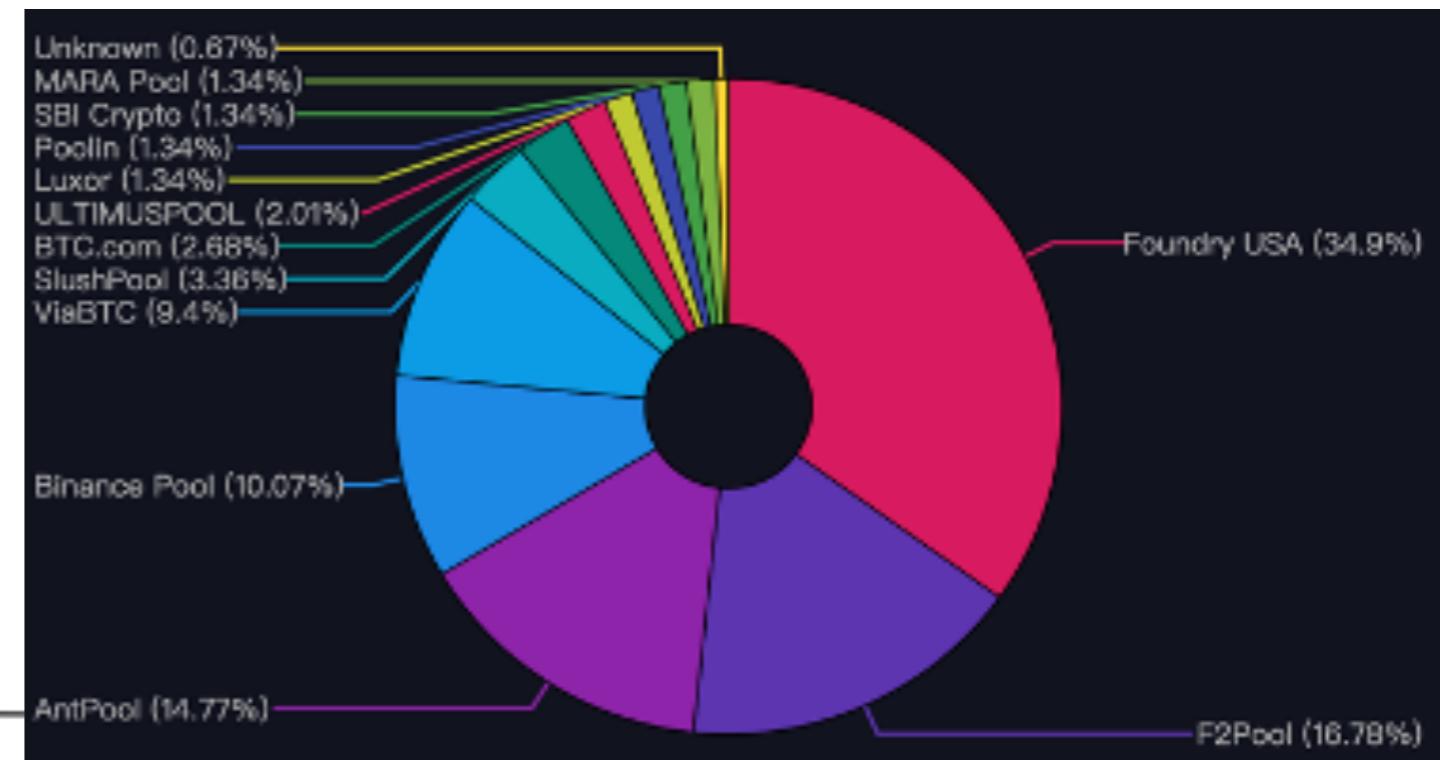
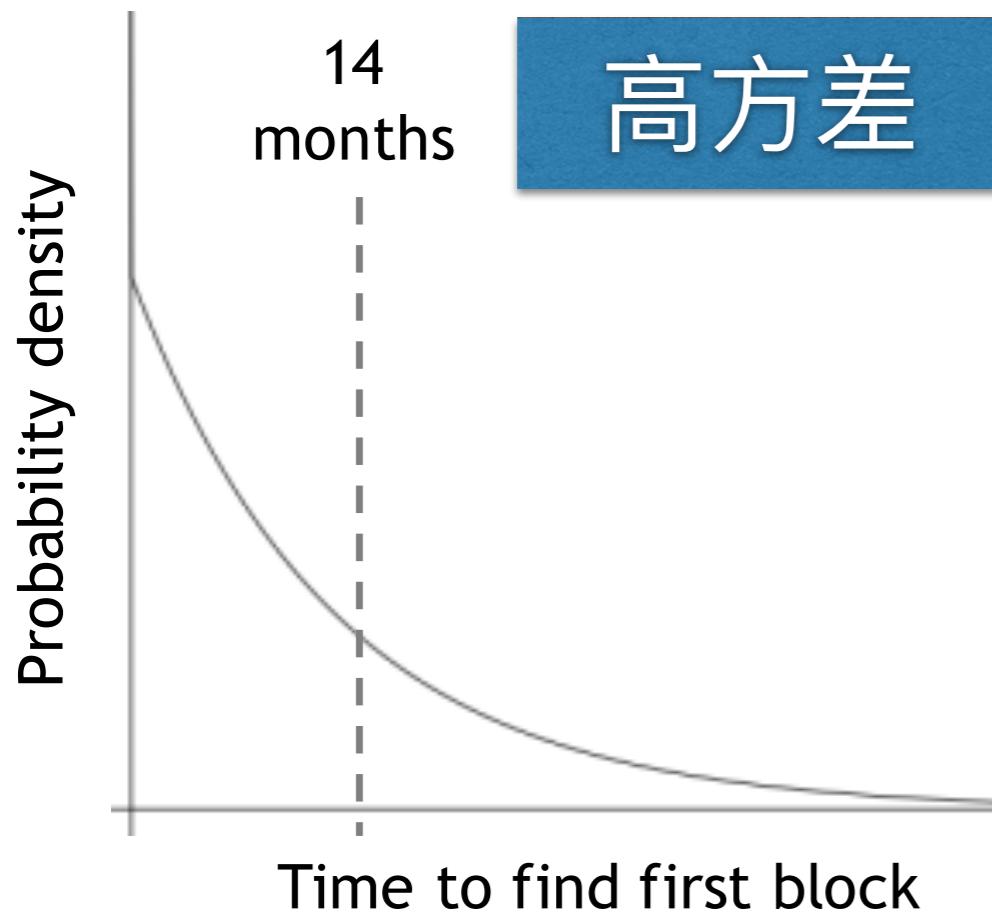
中国



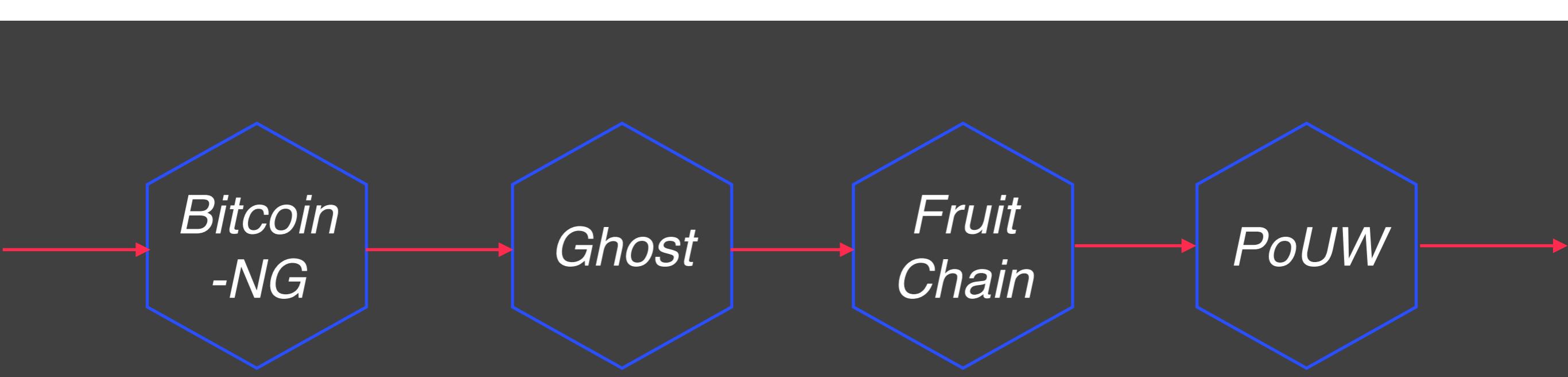
TerraMiner IV

Cost: ≈US\$6,000  
 Expected time to find a block: ≈14 months  
 Expected revenue: ≈\$1,000/month

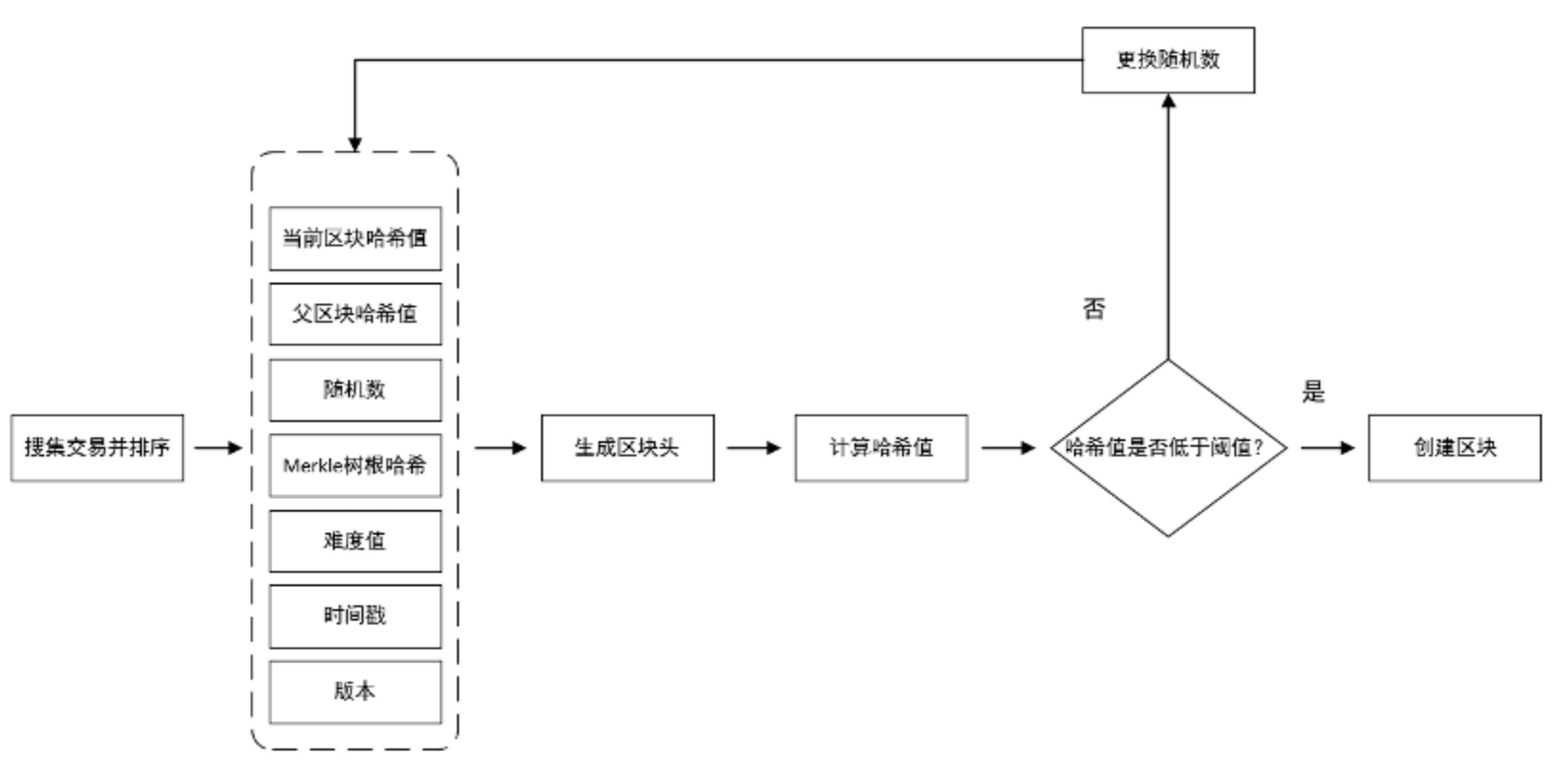
# blocks found in one year	probability (Poisson dist.)
0	42.4%
1	36.4%
2	15.6%
3+	5.6%

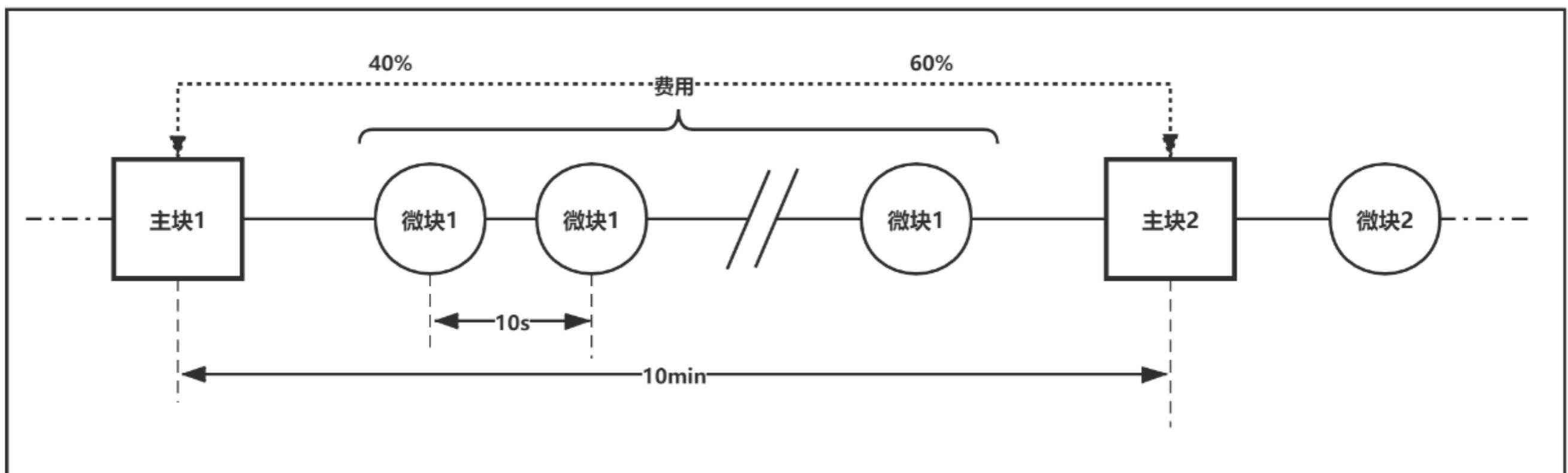


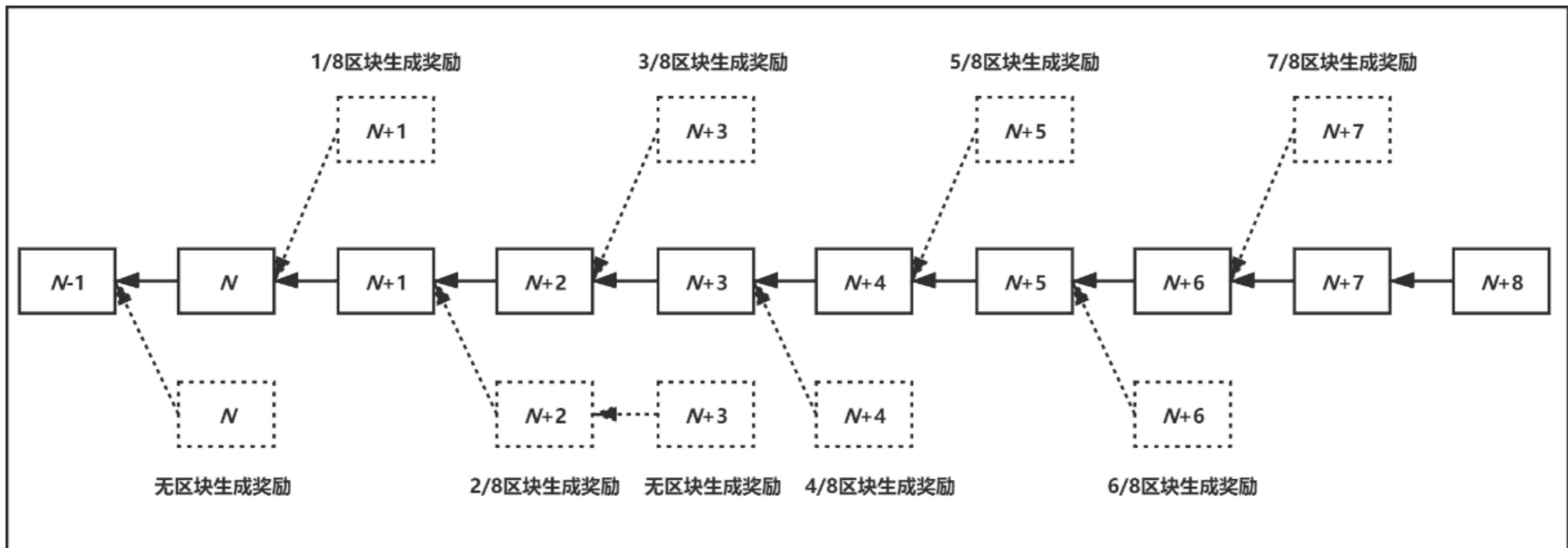
# PoW 改进



## PoW算法







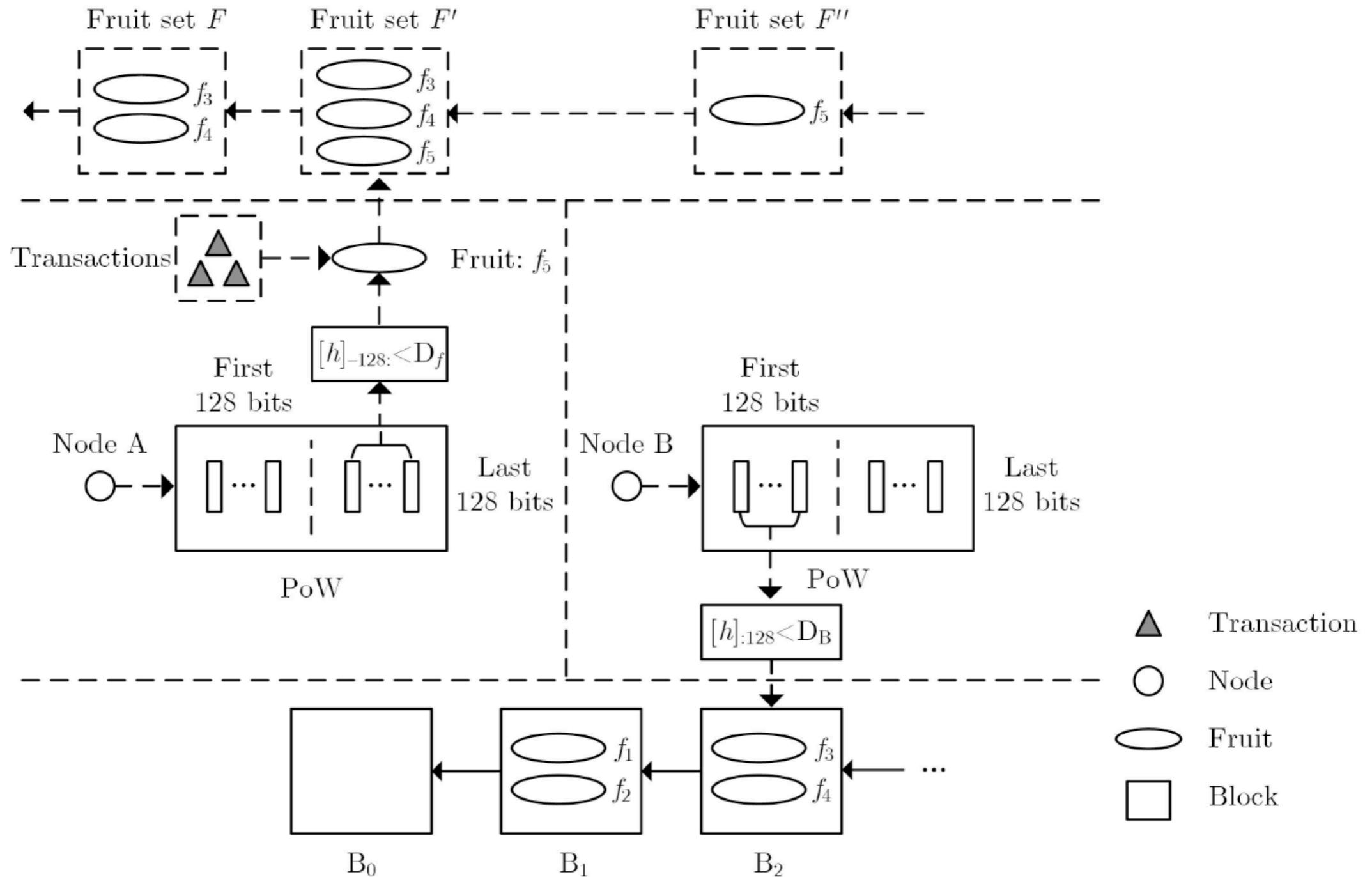


图 6 FruitChains 共识机制

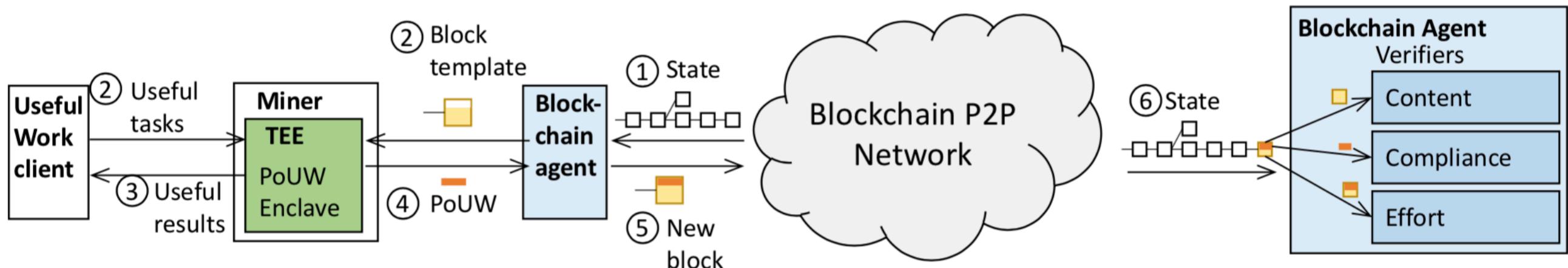
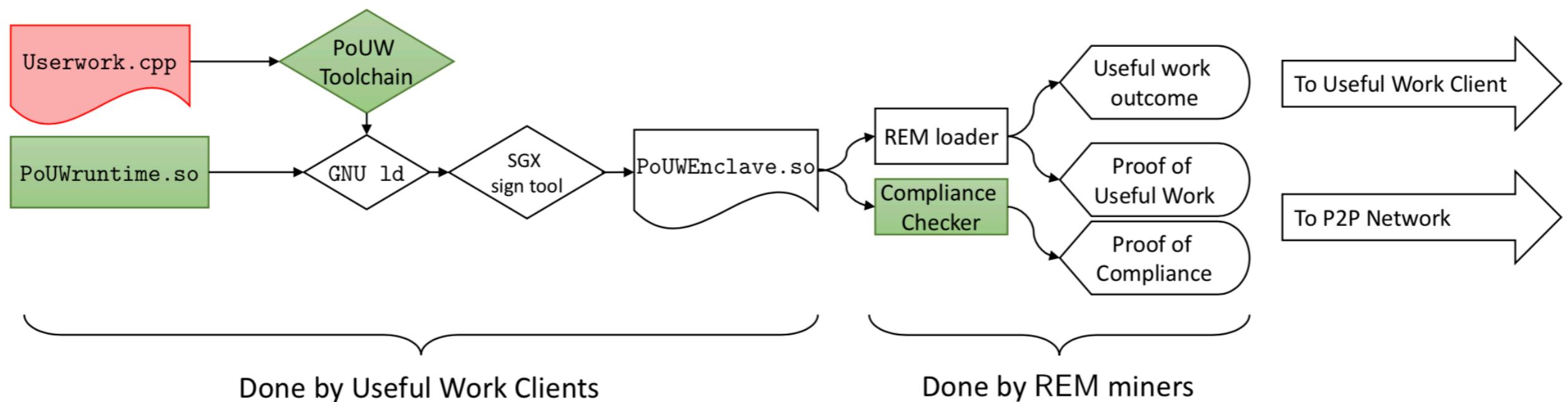
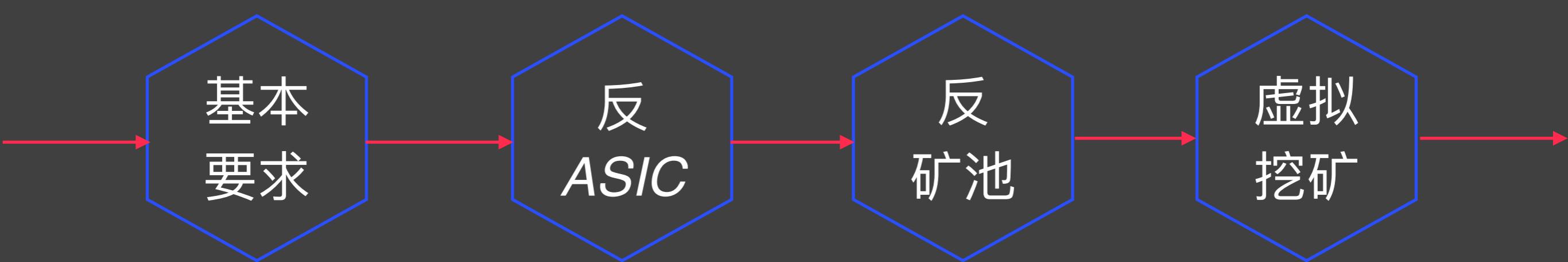


Figure 1: Architecture overview of REM



# 其他挖矿



## 挖矿算法基本要求

挖矿算法是比特币  
系统的核心

需要一个难题  
计算复杂

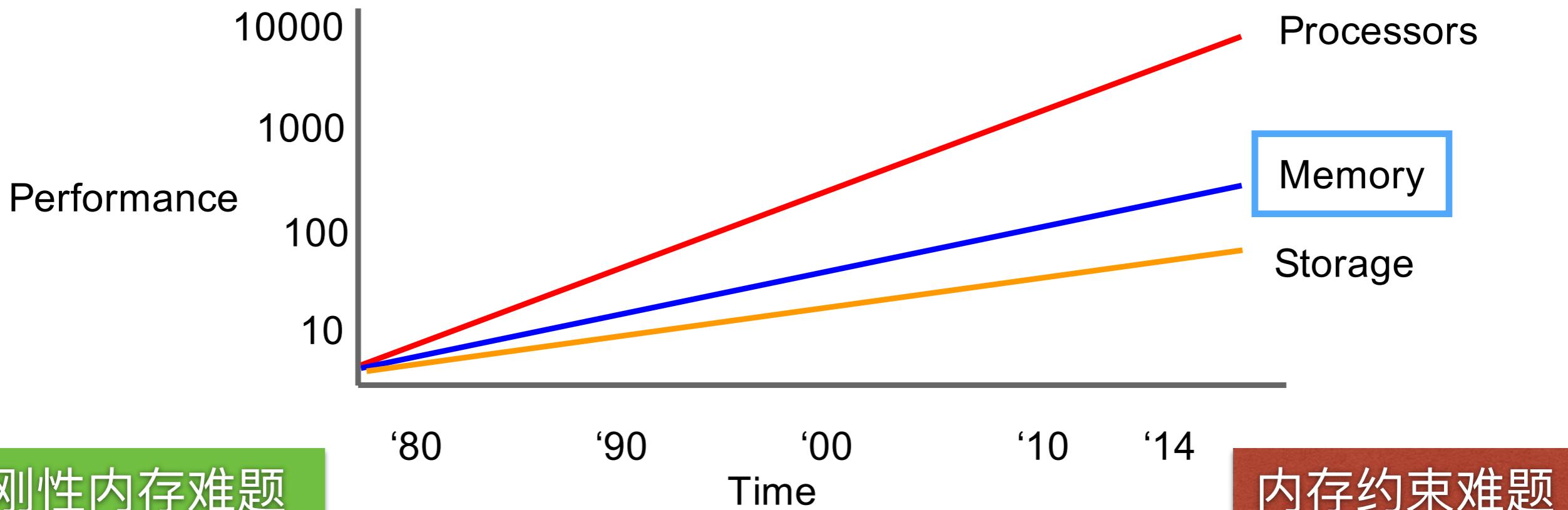
挖矿难题的结果要求验证简单

挖矿难题的难度可调节的特性

成功概率和所贡献的算力成比例



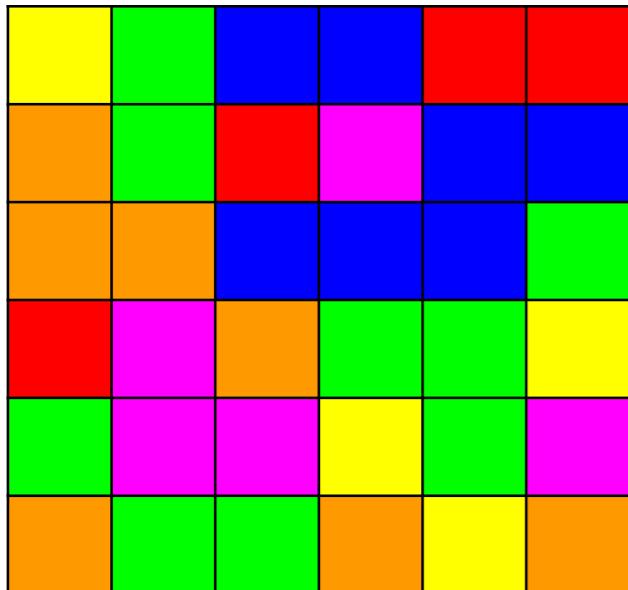
## 反ASIC



比特币前就存在  
加密个人口令

2009

反ASIC



检验成本过高

内存使用参数  
设置过低



DASH

组合多种Hash算法

XII

参数

反ASIC是否可能

SHA256

反ASIC是否有问题

## 有效工作量证明

挖矿能量消耗问题

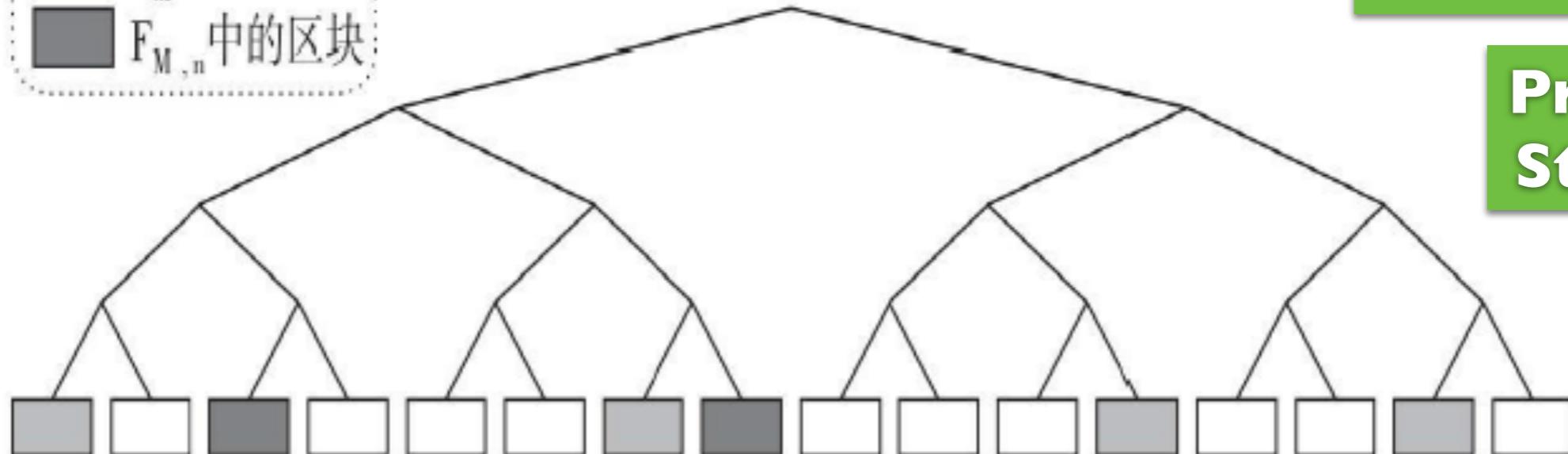
志愿者计算项目



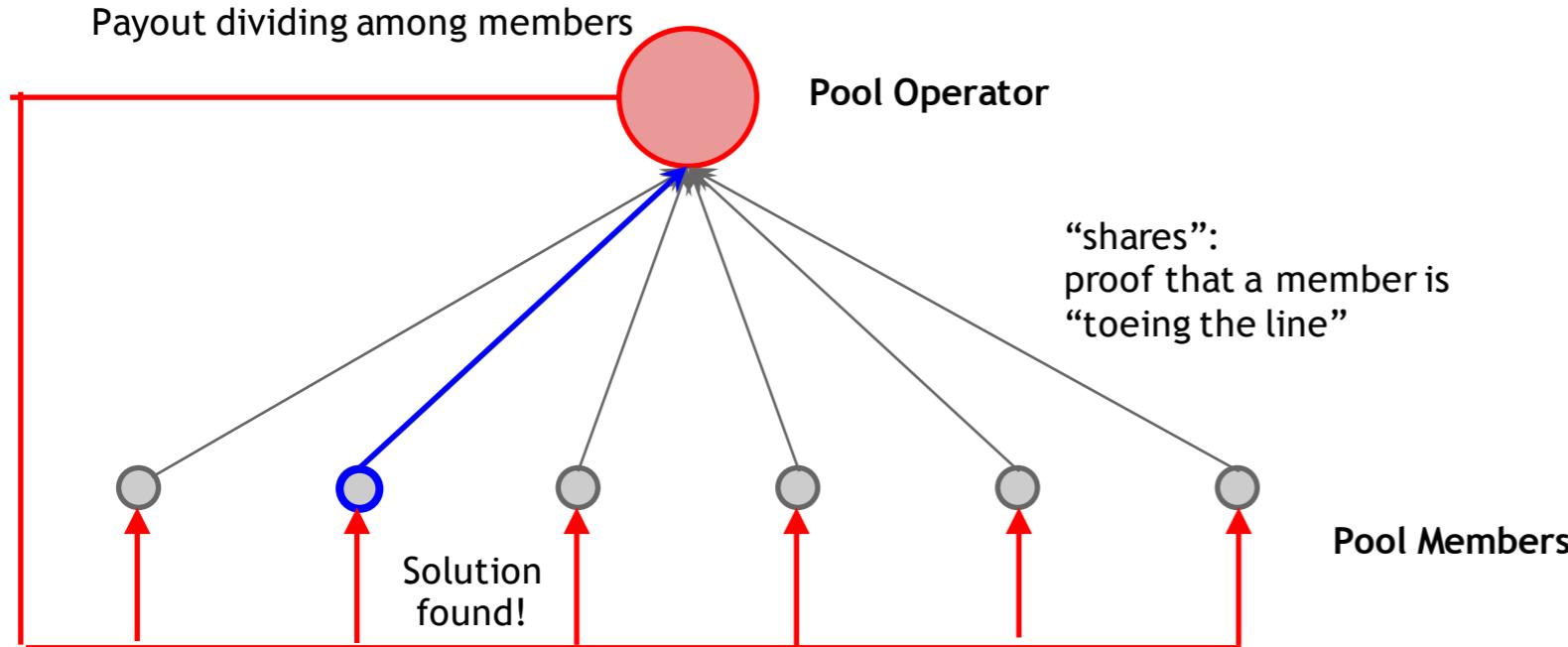
分布式  
存储

- F中的区块
- F<sub>M</sub> 中的区块
- F<sub>M,n</sub> 中的区块

F 的根



## 不可外包的难题



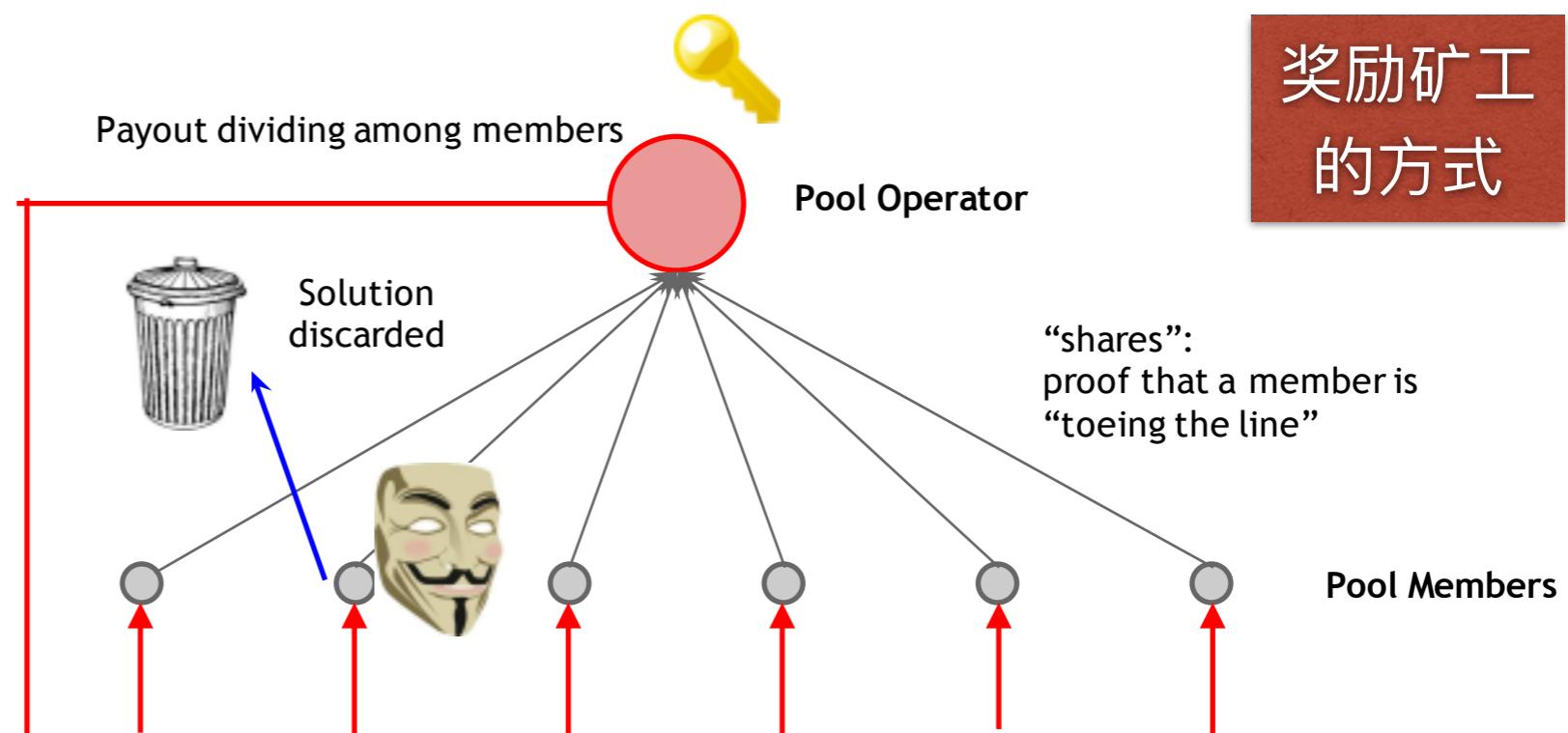
防止矿池的产生

中心化、安全

区块丢弃攻击

奖励破坏

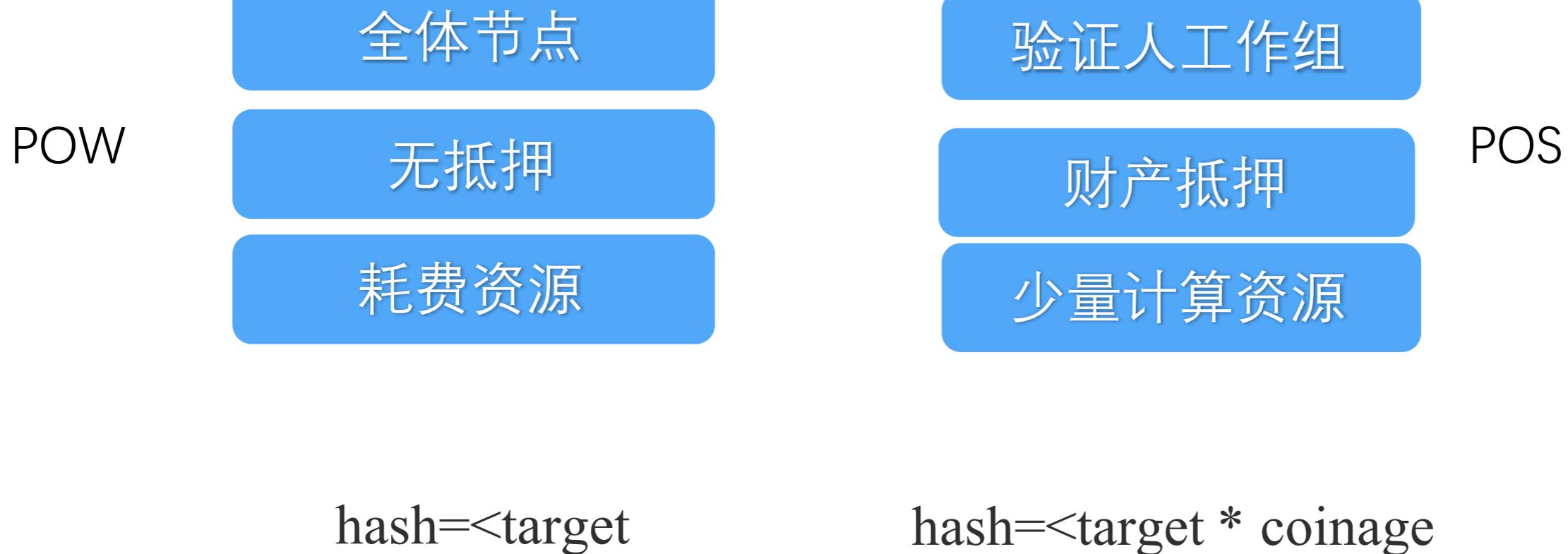
区块数字签名的哈希值  
低于一个特定的目标



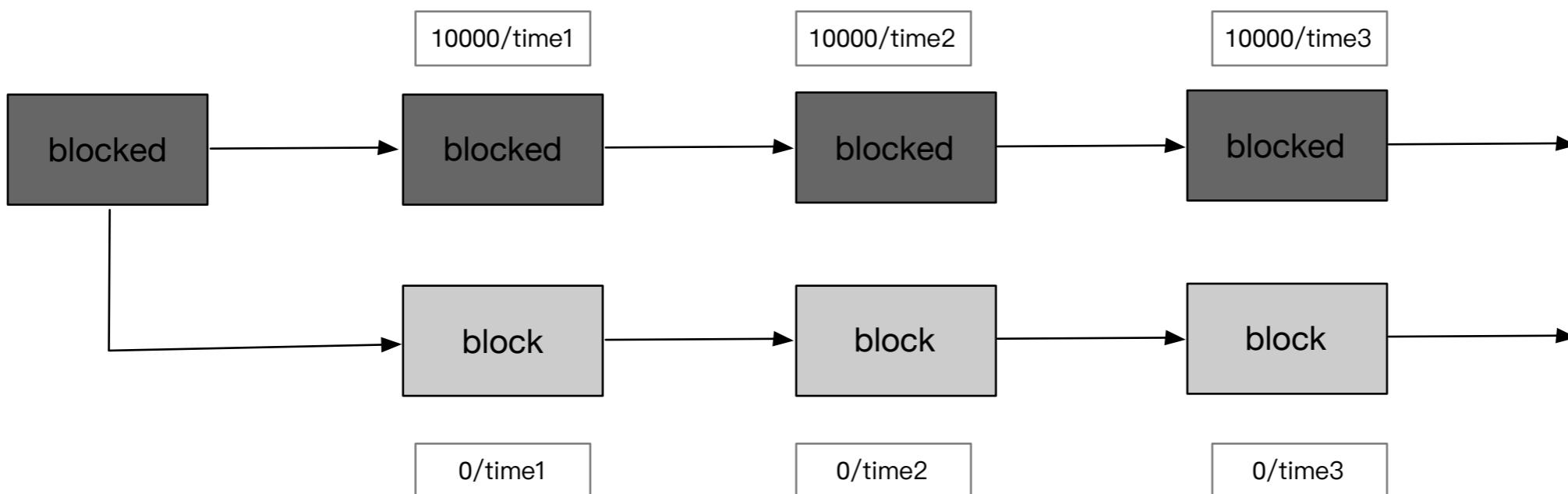
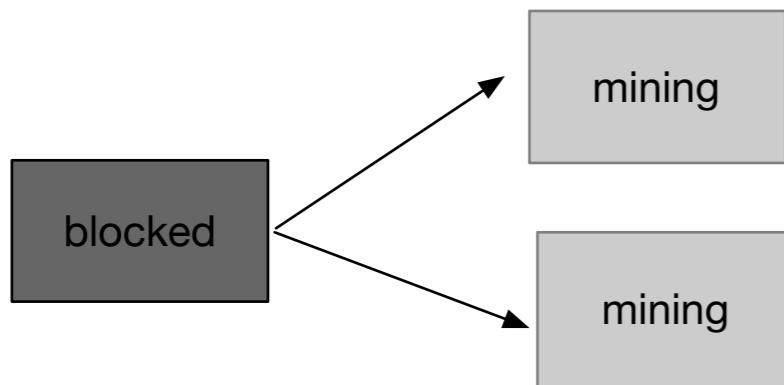
奖励矿工  
的方式

每个领导者被选出的概率与持有股份数量成比例，被选出的领导者可以生成区块。

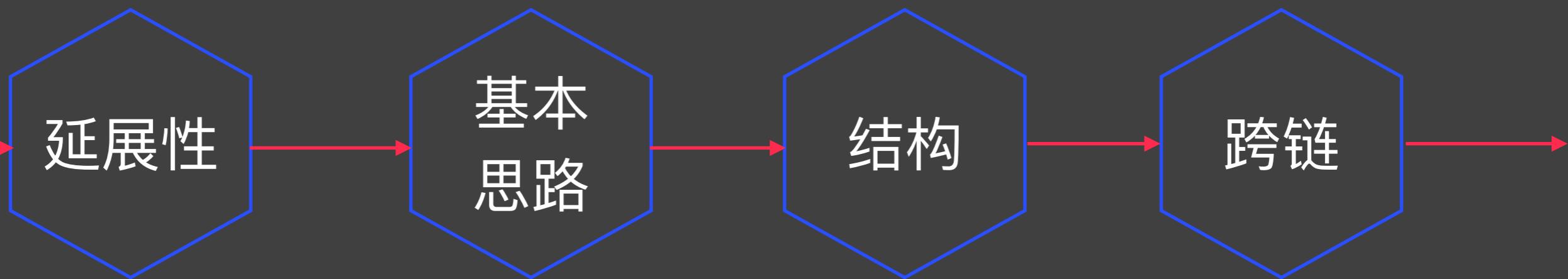
—Janno Siim: Proof-of-Stake Research Seminar in Cryptography



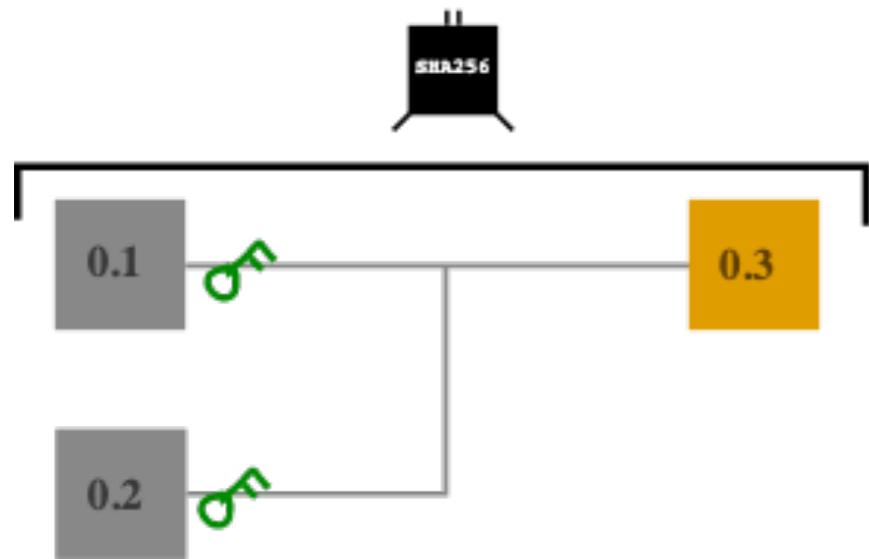
- POS挖矿资源消耗的低
- 矿工在分叉链上挖矿消耗少
- 矿工可以同时挖分叉链
- 没有相应的限制和惩罚
- Deposit-based PoS



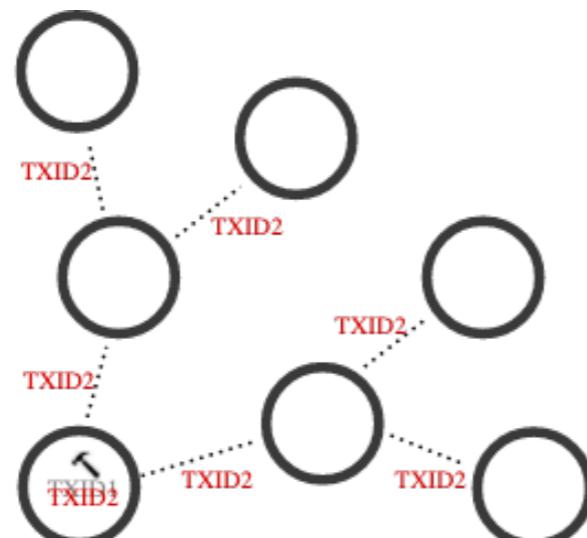
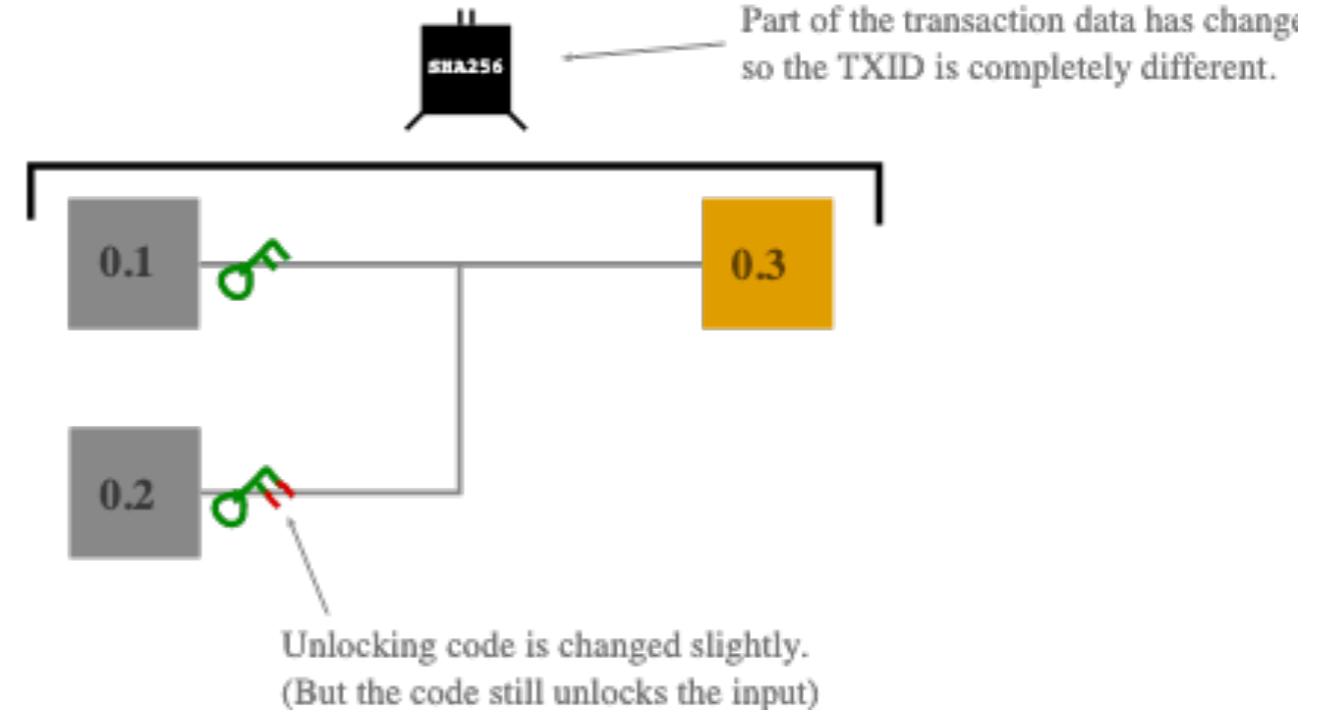
# 隔离见证



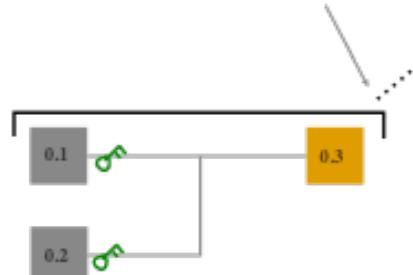
**TXID1**



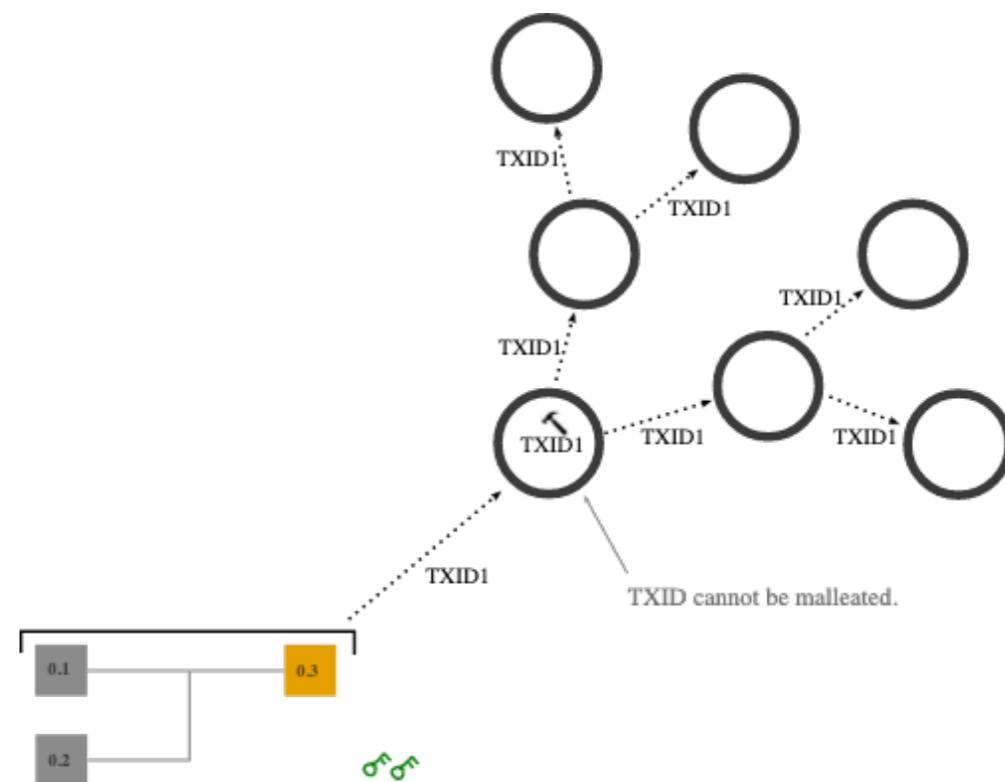
**TXID2**



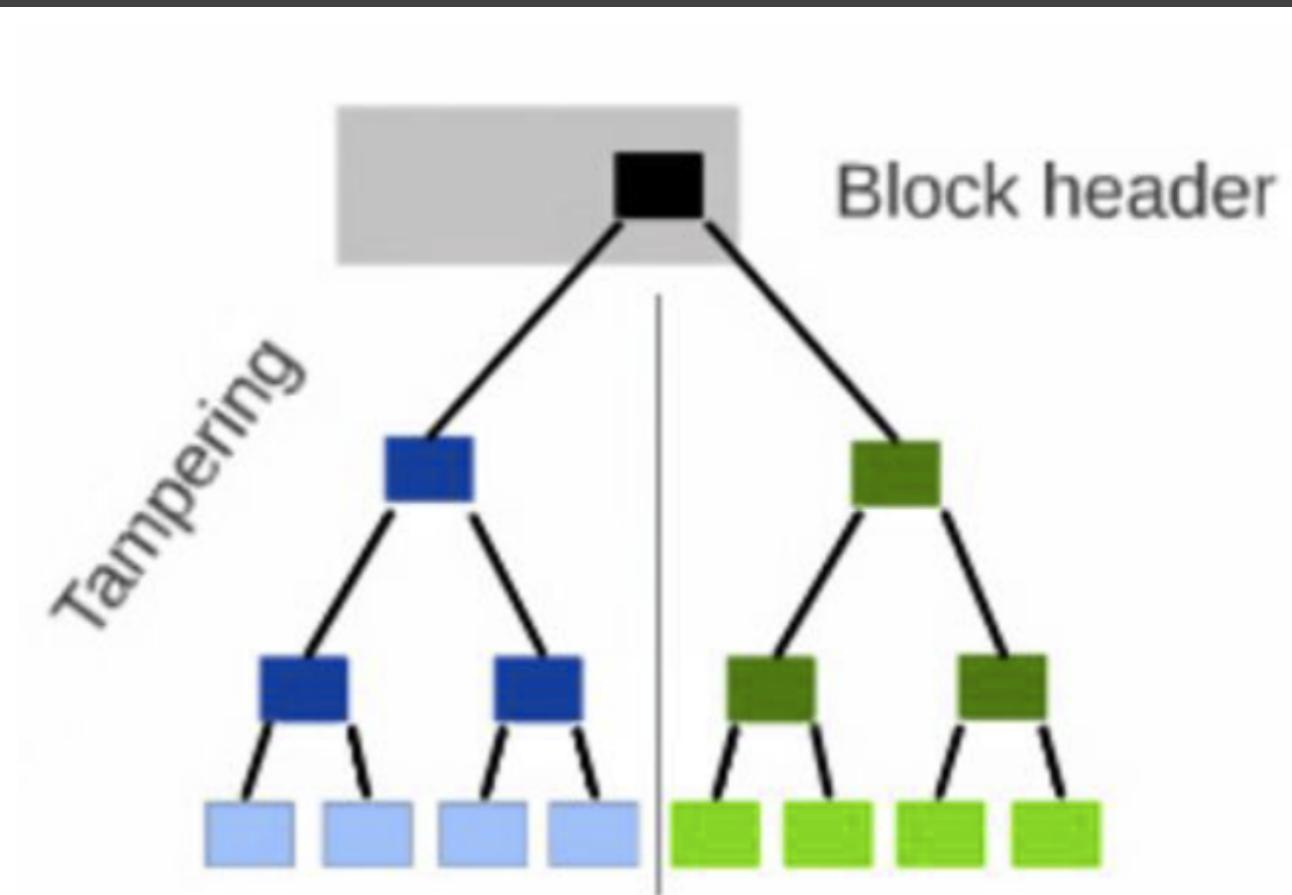
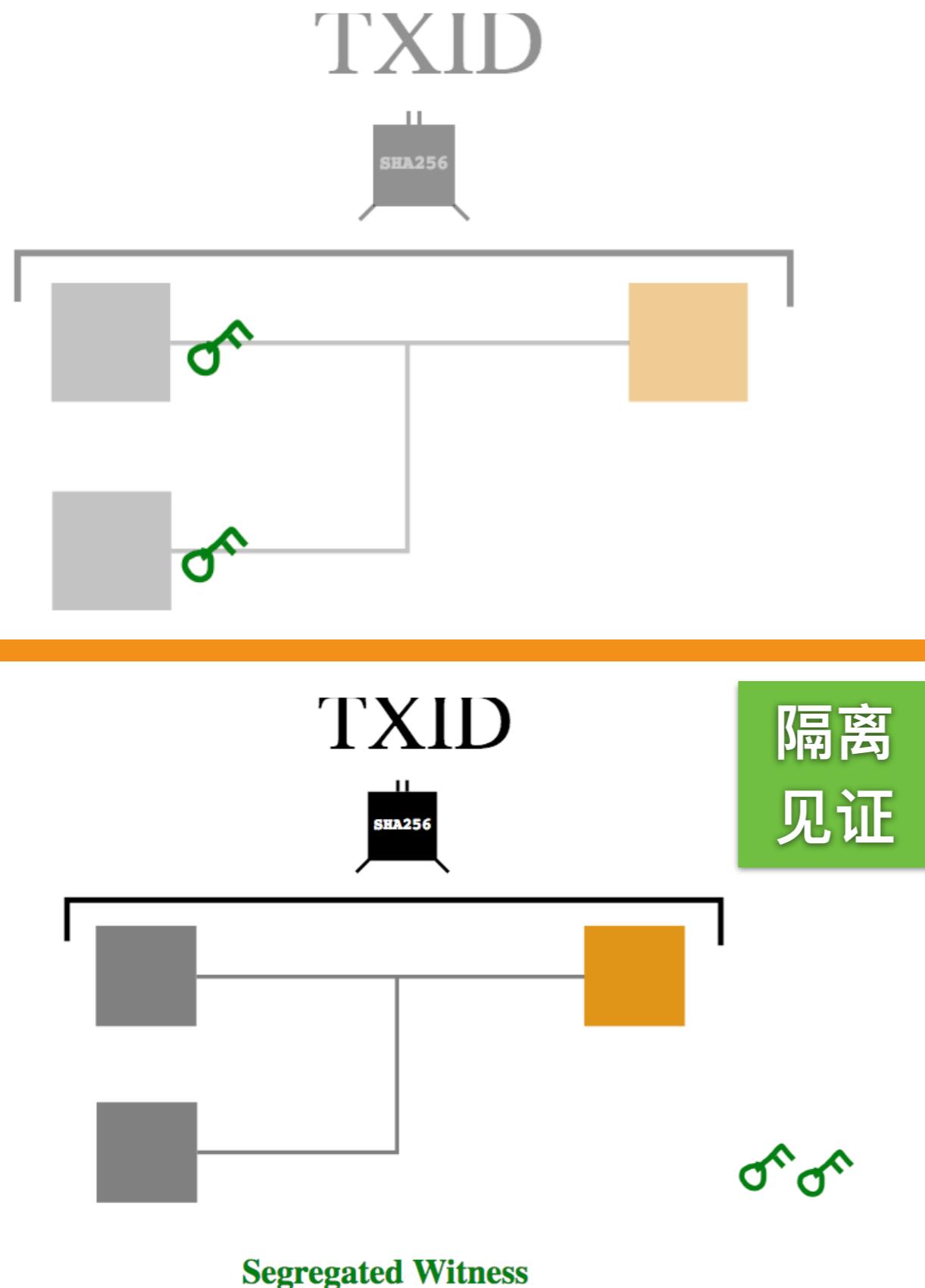
Insert your transaction in to the network.



This node malleates your transaction's TX



## 基本思路

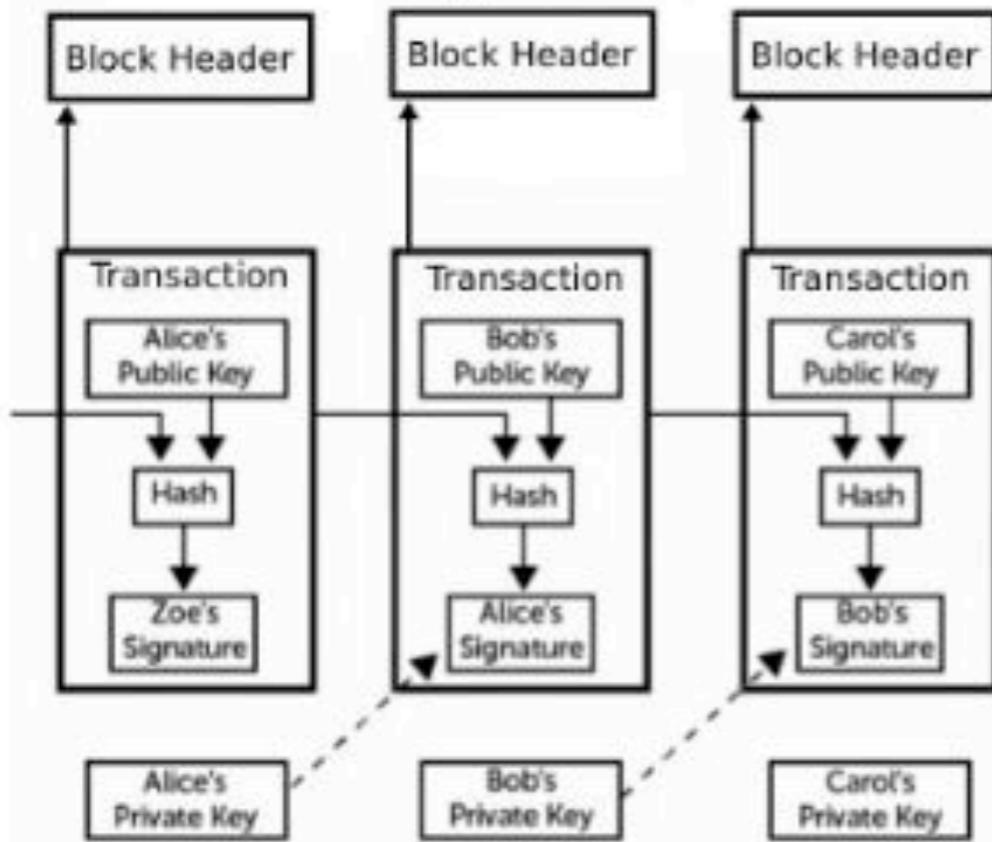


Merkle tree of txn and witness

优点

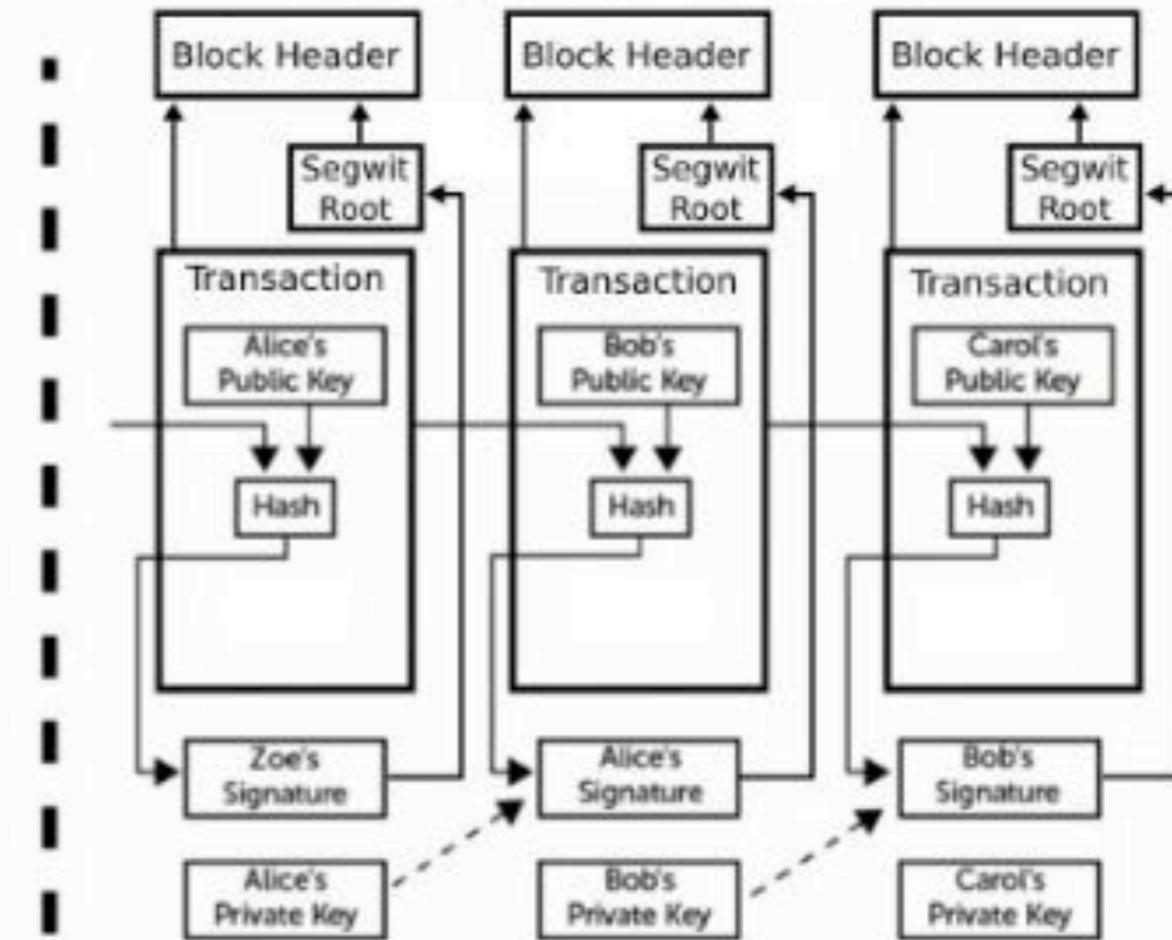
缺点

## Non-segwit blocks



Each block header includes a cryptographically-secured reference to all of the transaction data in that block.

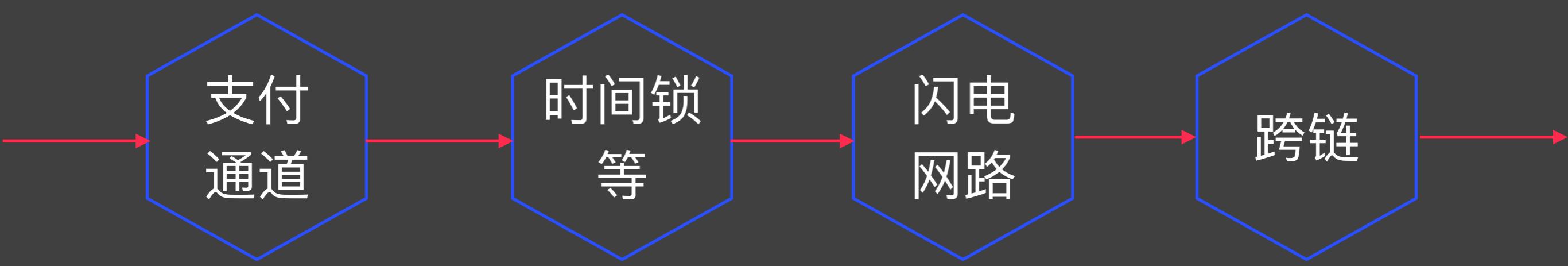
## Segwit blocks



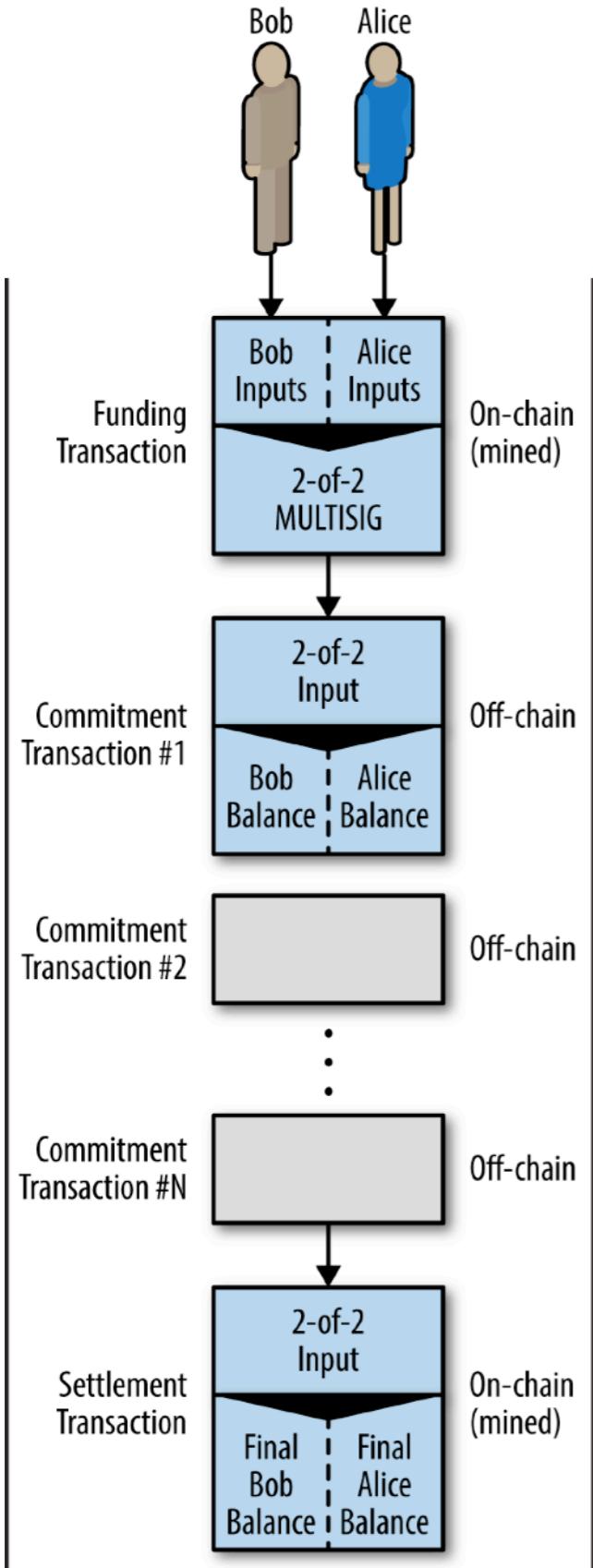
Each block header still includes a cryptographically-secured reference to all of the transaction data in that block, but encumbrances (public keys) are referenced separately from witnesses (signatures) so that software can use each part independently.

An illustration of the old and new transaction serialization posted by [David A. Harding](#).

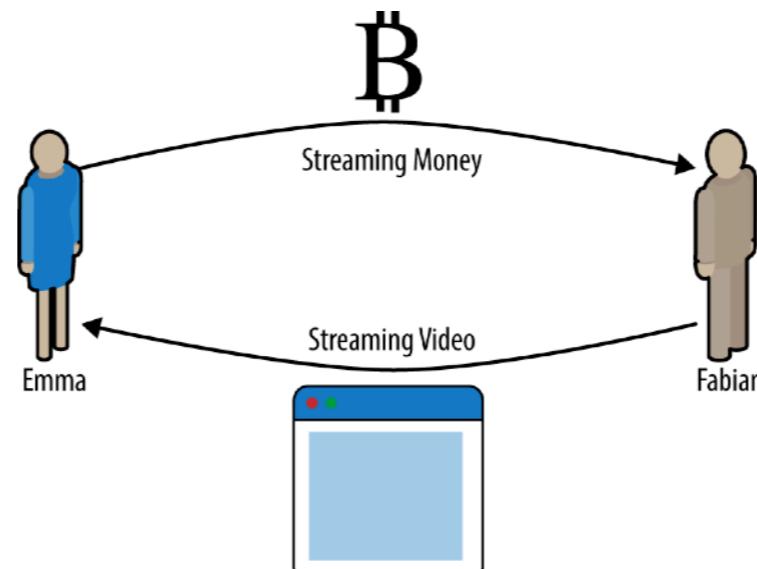
# 链下扩容



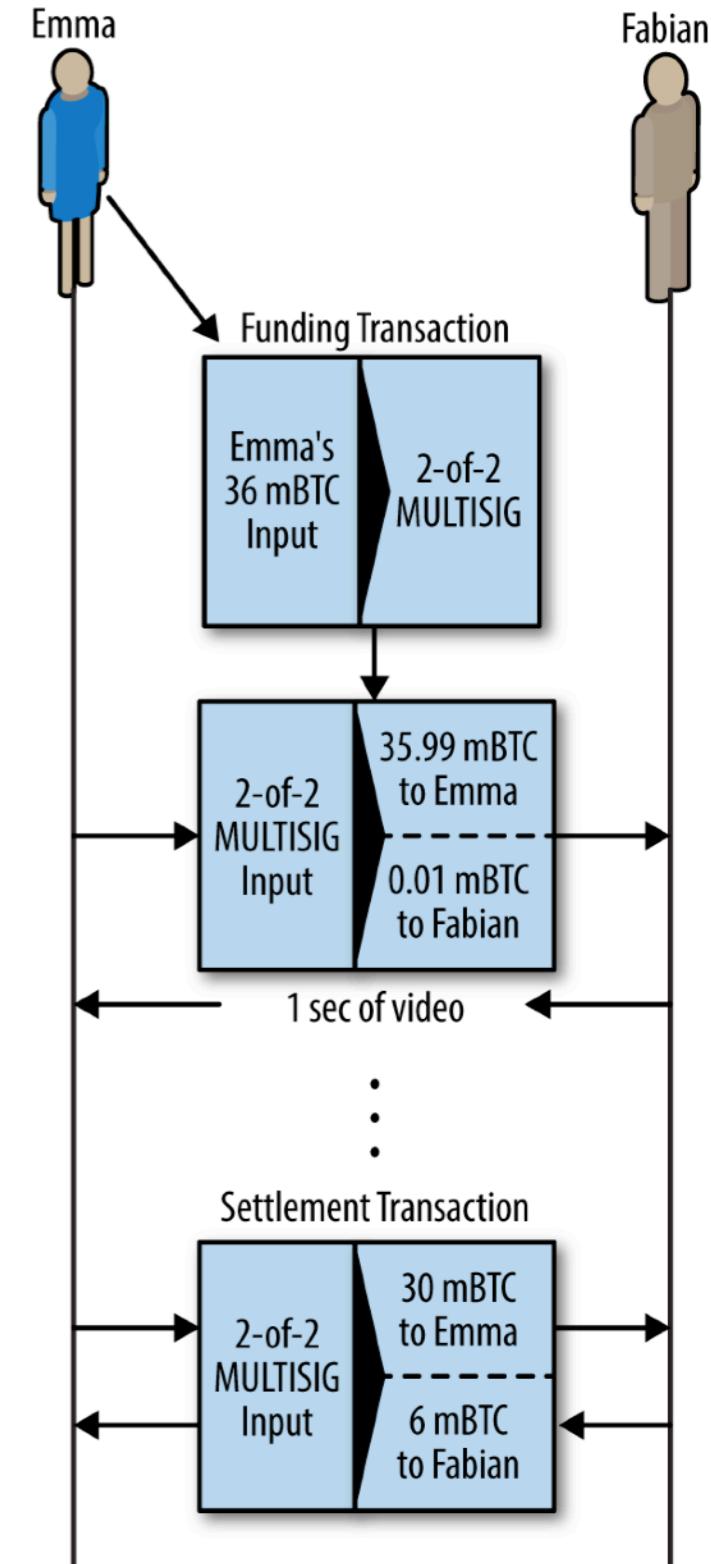
# 支付通道



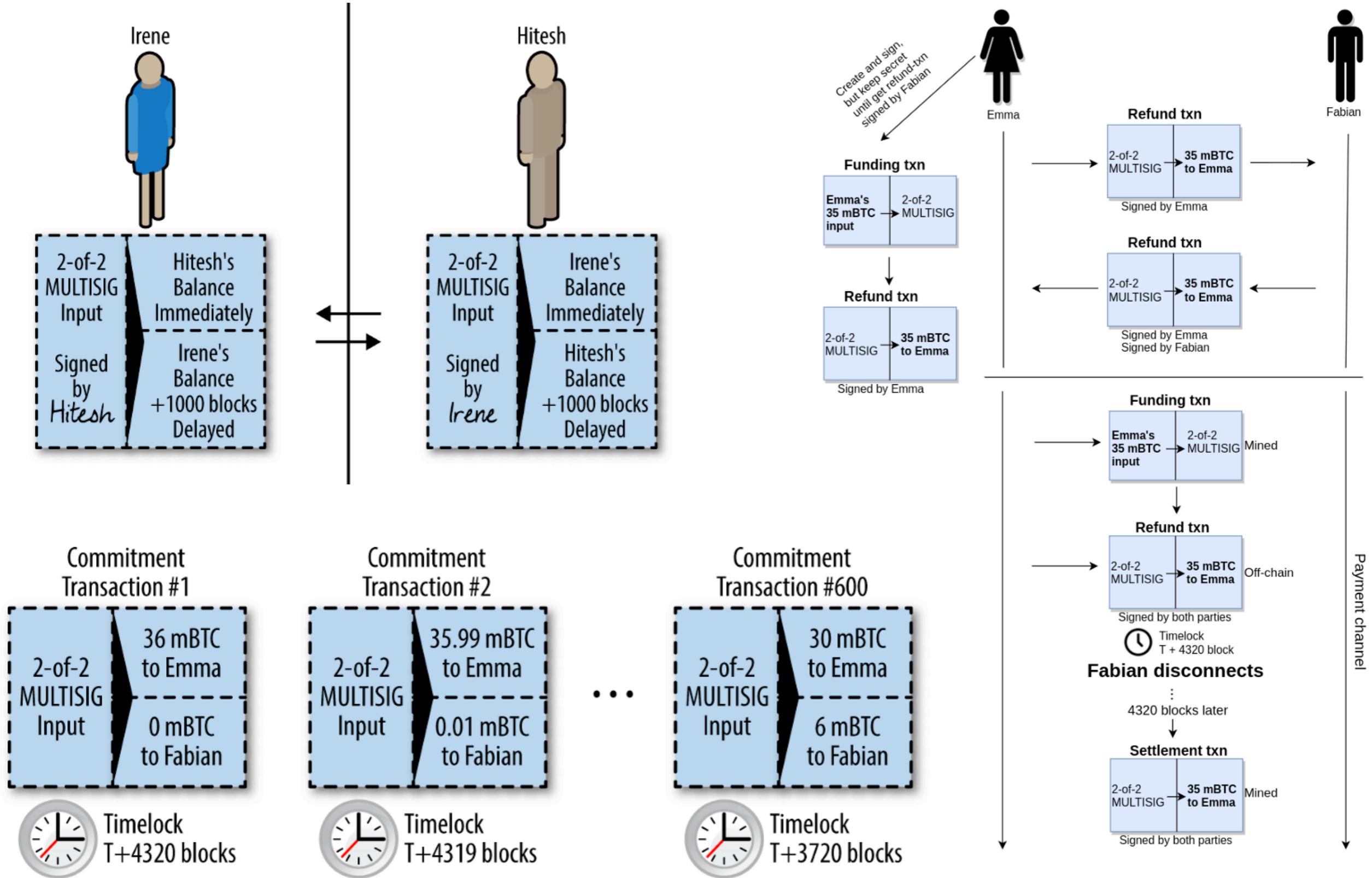
**支付通道**  
是在比特币和区块链之外双方  
之间交换的比特币交易的一种  
无需双方互相信任的机制

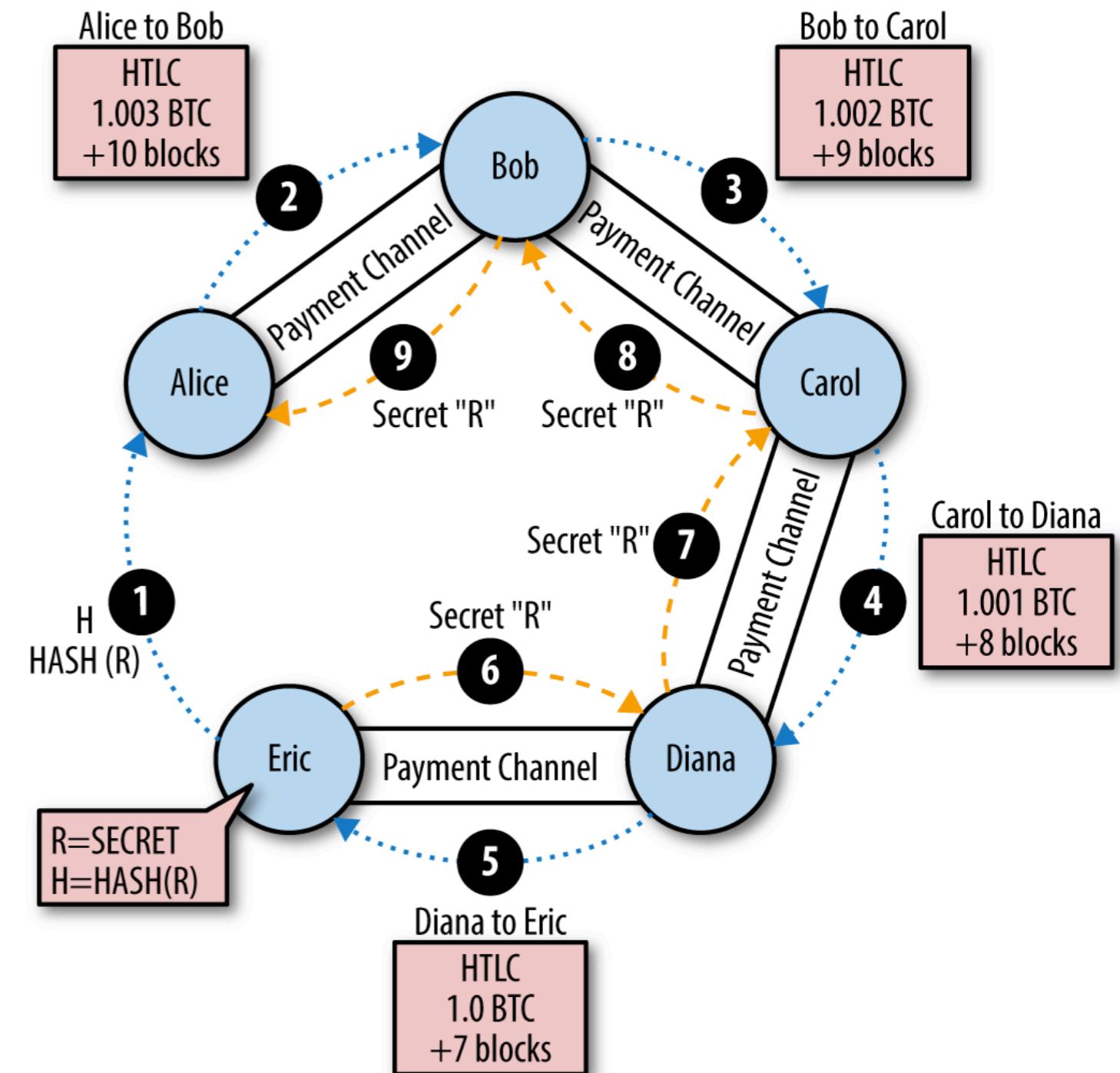
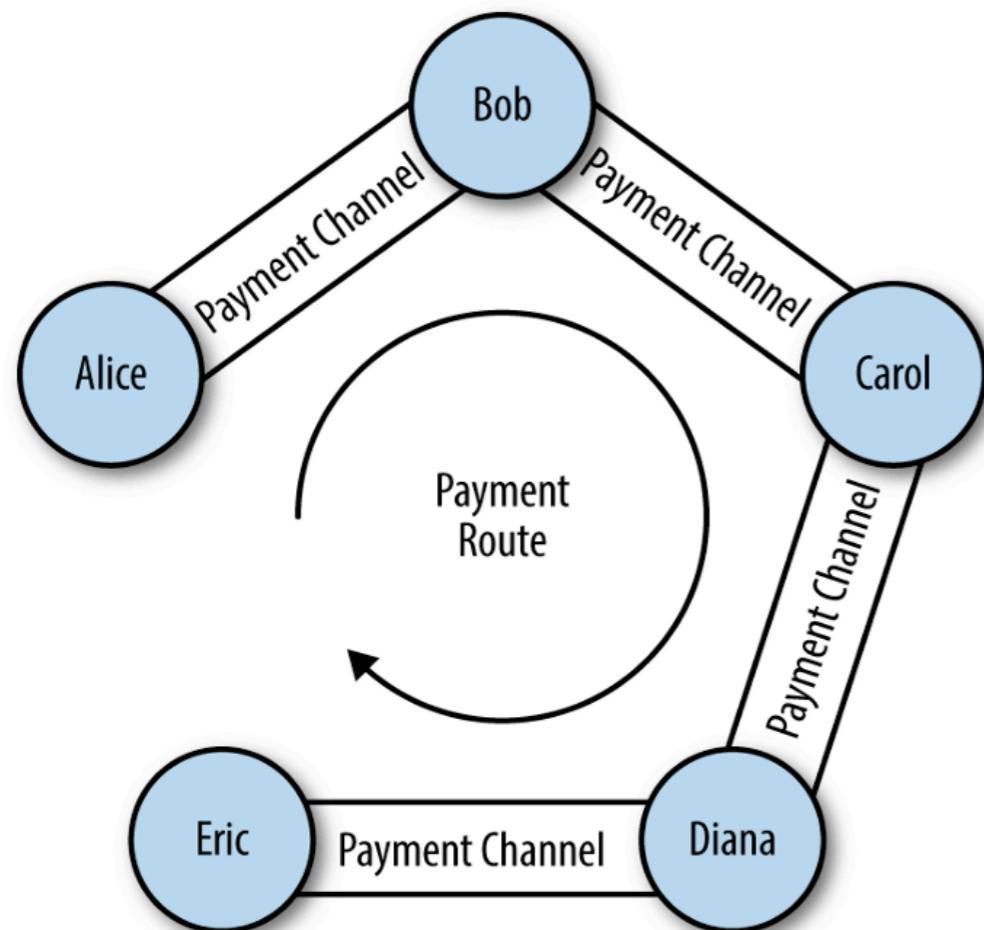


高吞吐量、低延迟、细粒度

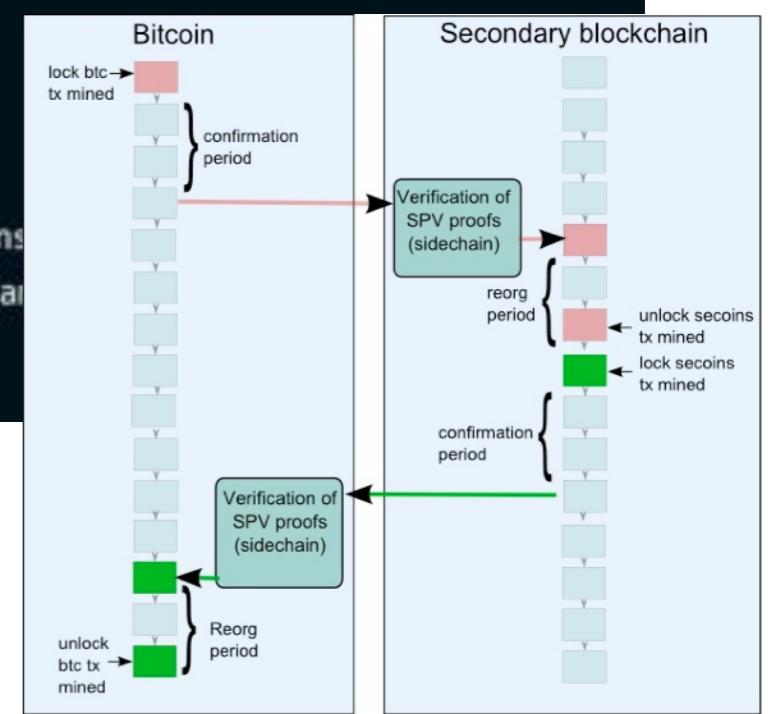
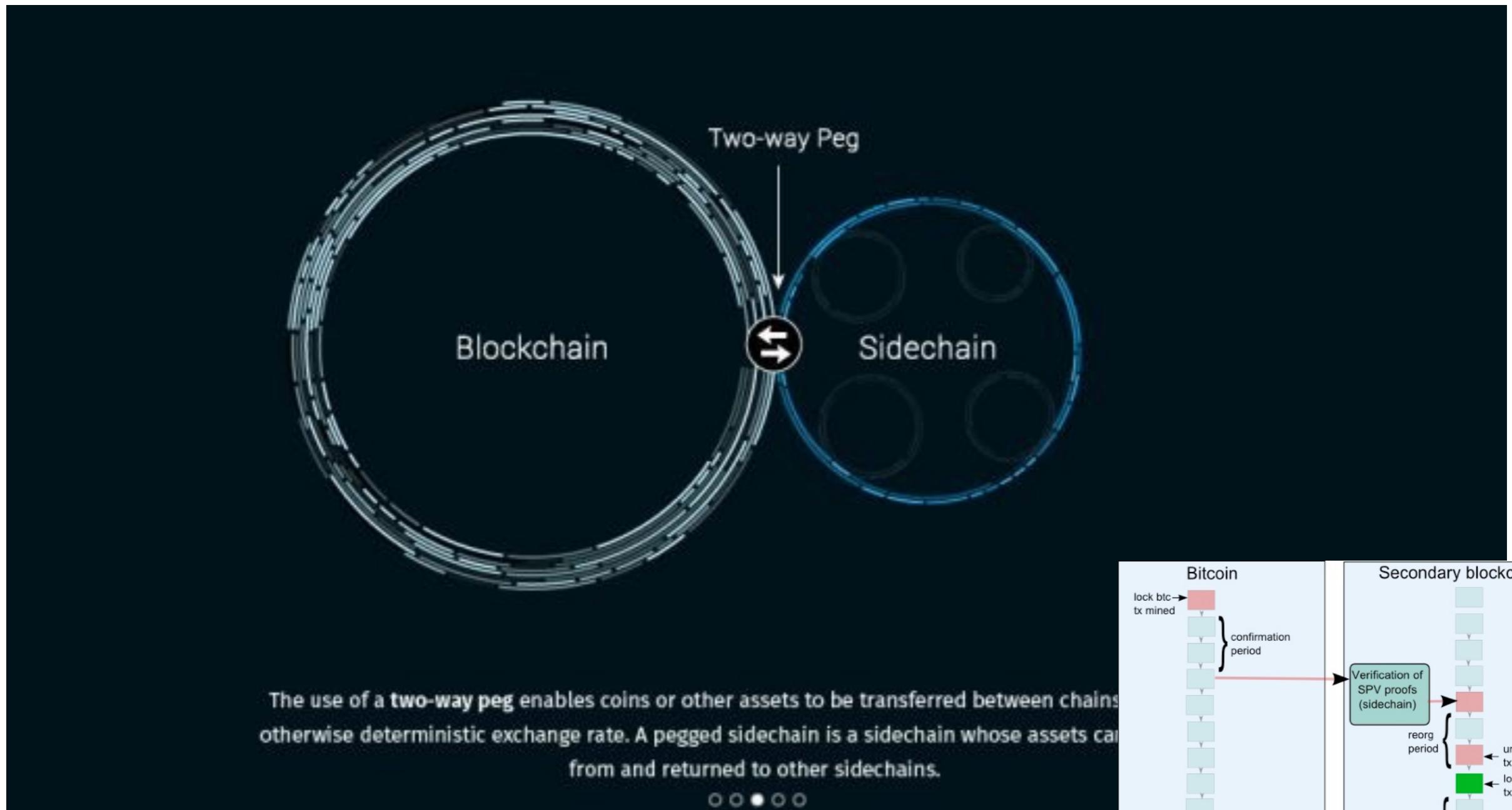


## 时间锁等

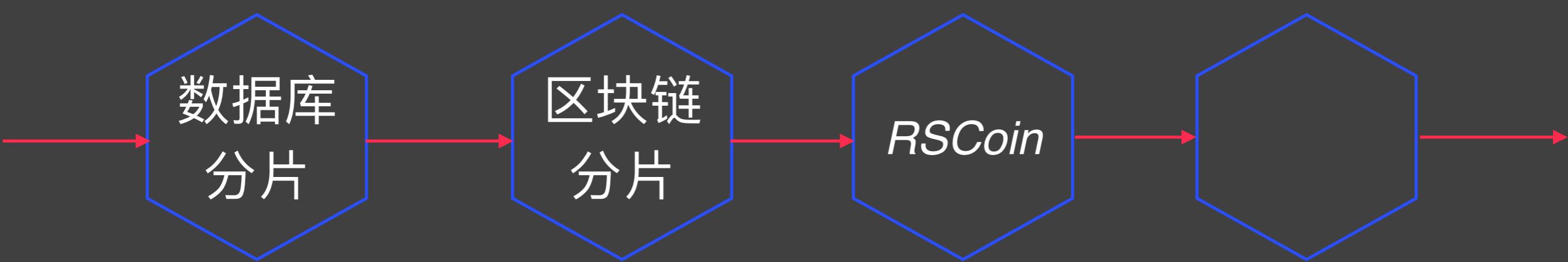




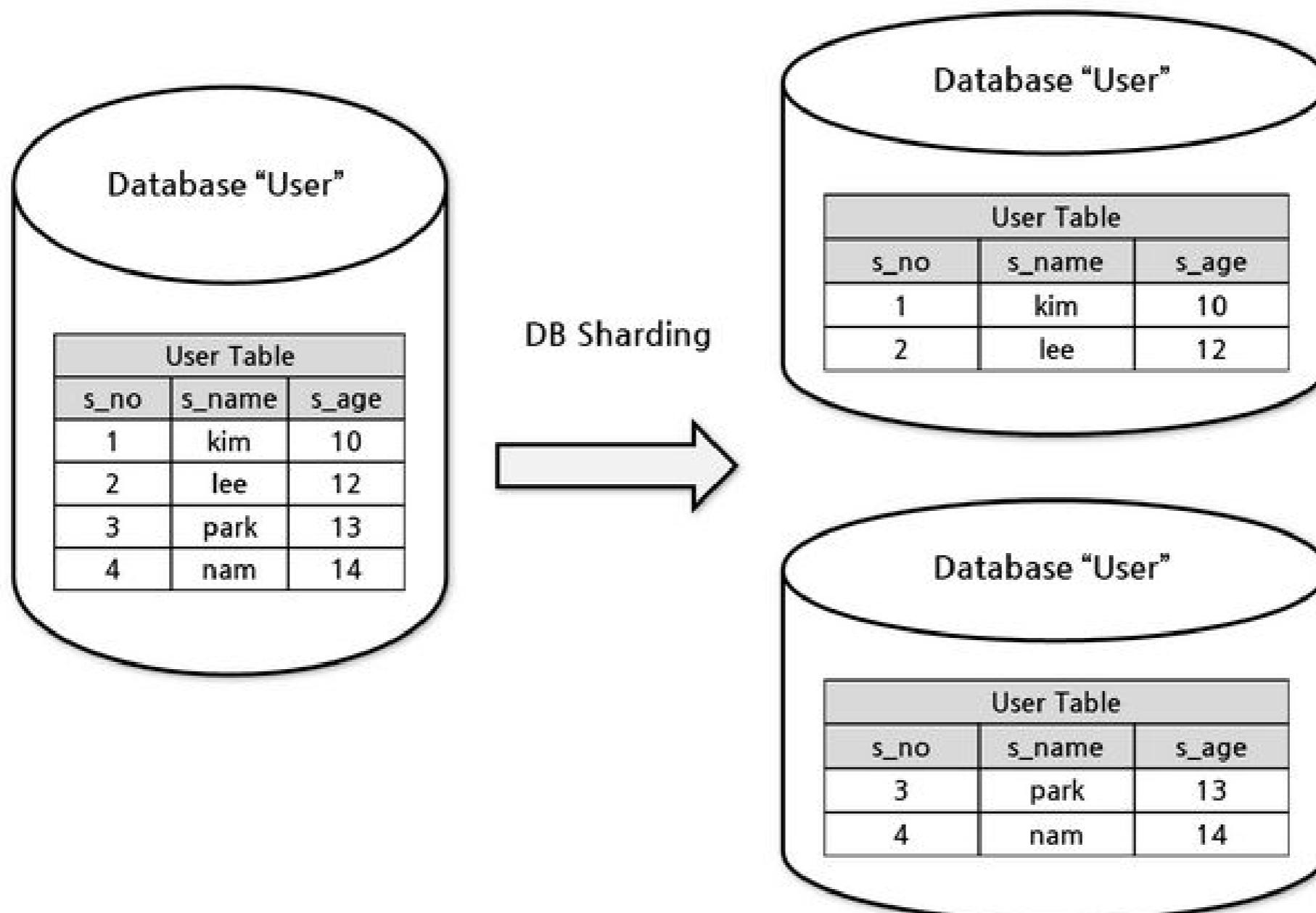
# 侧链



# 分片



## 数据库分片



# 区块链分片

1

## 节点划分

- 分片技术的核心
- 将系统的矿工分为不同的验证分片(shard)或者验证委员会(committee).

2

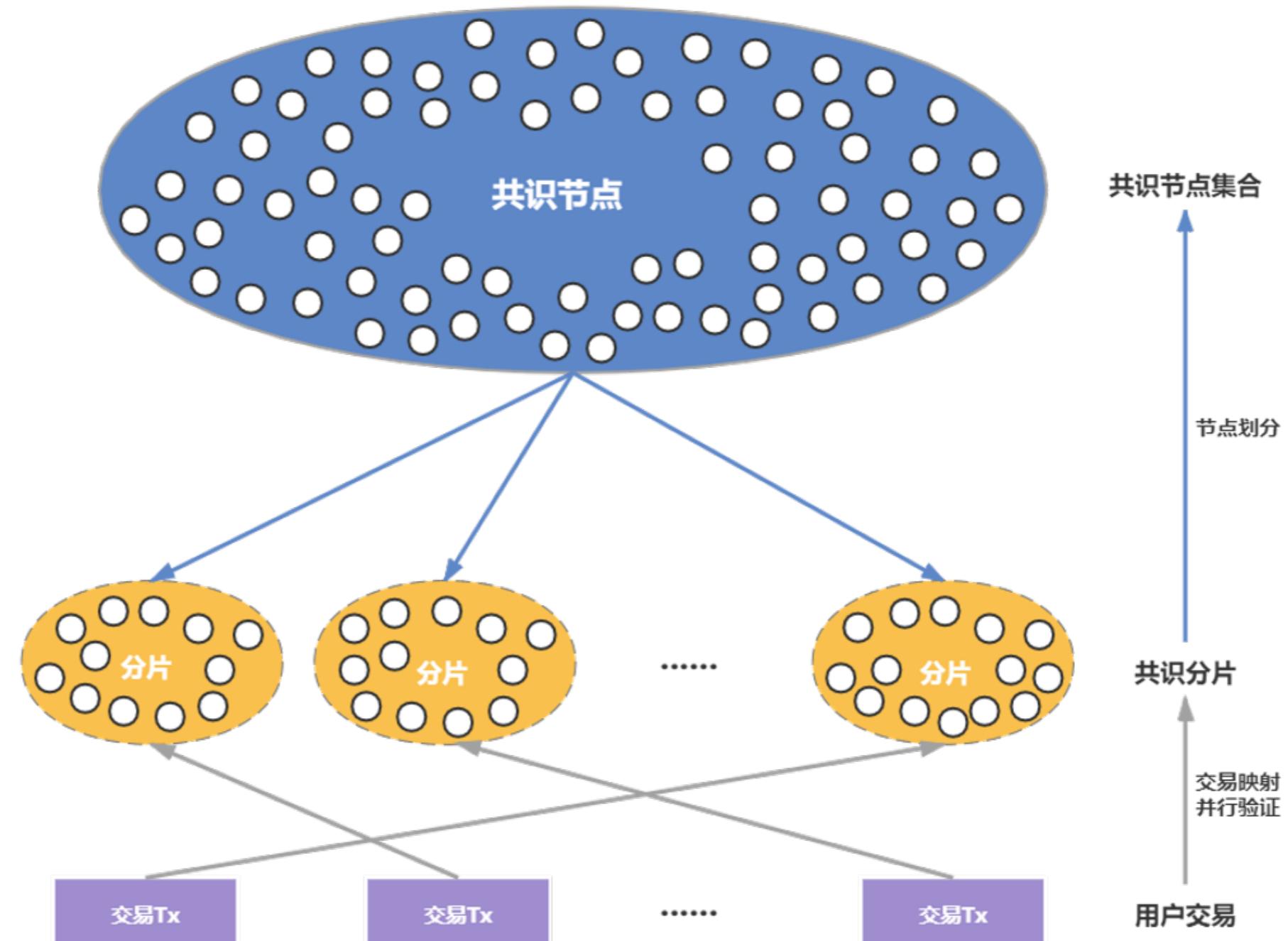
## 交易映射 并行验证

- 每个交易被映射到确定的分片进行处理
- 每个委员会只负责处理系统全部交易的一部分
- 各个委员会可以**并发地**处理交易

3

## 分片方案的优点

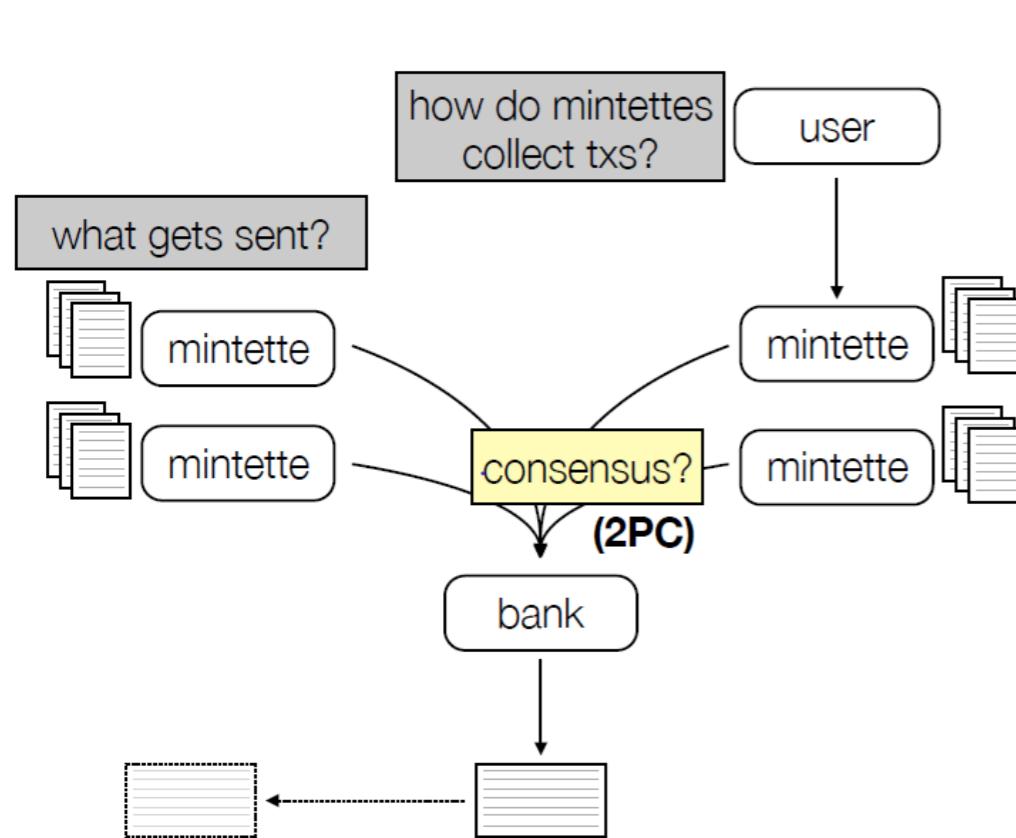
- 系统整体的**交易处理吞吐量**因而得到提升
- 使用**分布式一致性算法**在节点较少时效果比使用非确定性共识更好
- 降低每个节点的**存储开销**
- 使得系统具有良好的**可扩展性**



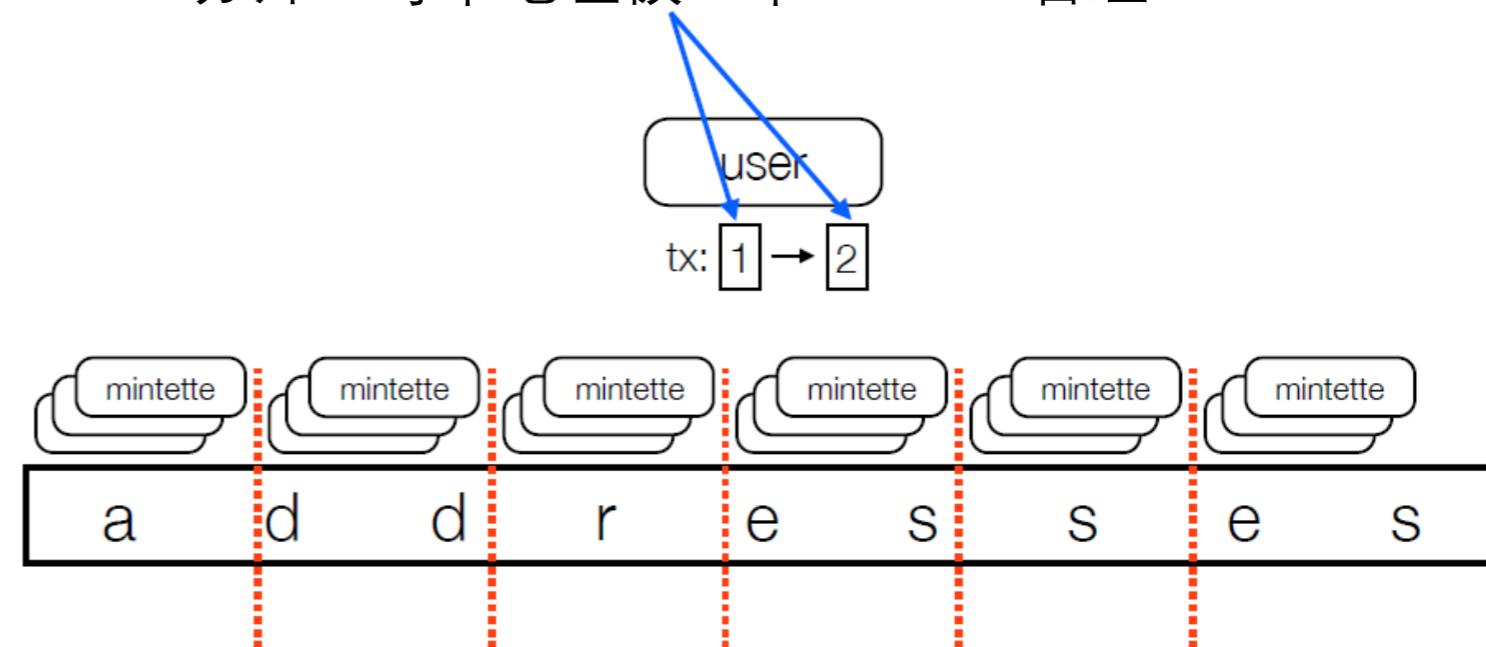
4

## 问题

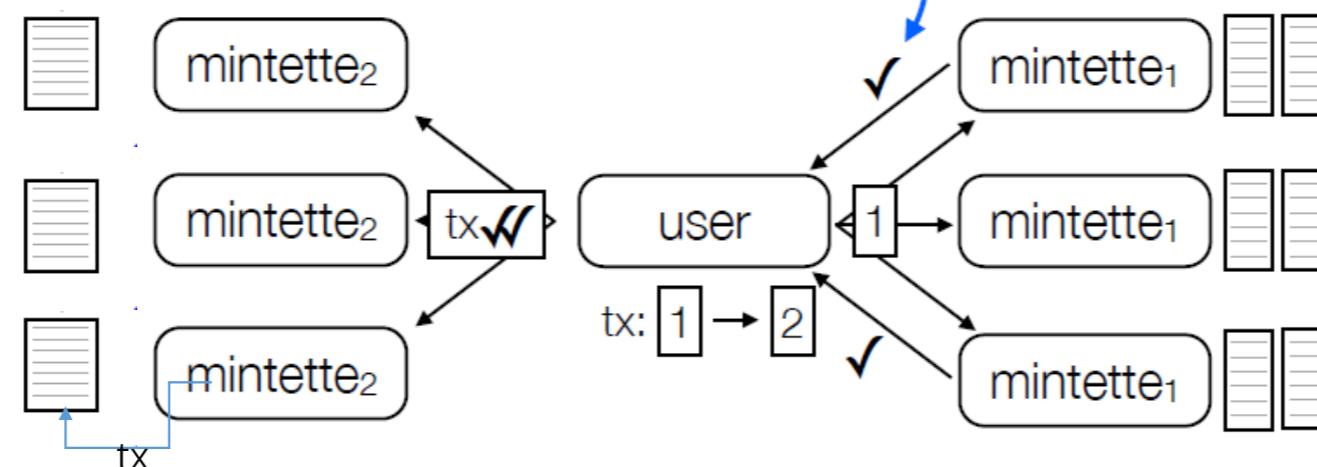
- 跨片交易的存在
- RapidChain中分析得出了96.3%的交易为跨片交易
- 交易处理能力不能随着系统节点数的增加**线性增长**



分片：每个地址被一个mintette管理



mintettes通过检查授权的mintette的签名来决定交易的有效性

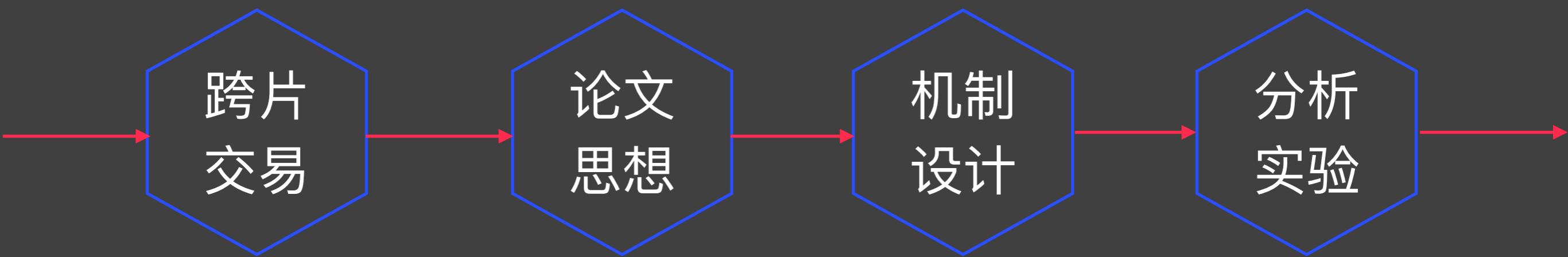
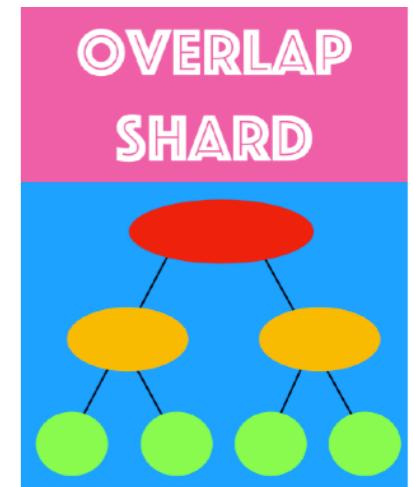


如果签名满足条件，被提交到账本并返回收据

投票

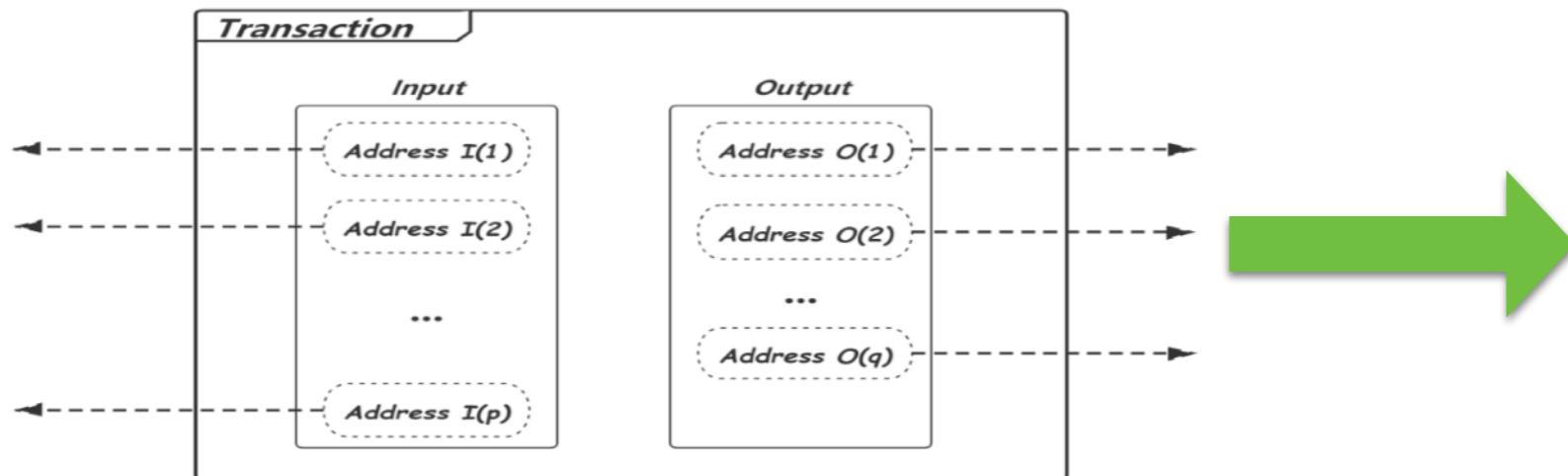
mintette维护一个UTXO列表并负责双重支付的检查

# 重叠分片



# 跨片交易

## 跨片交易



基于UTXO(Unspent transaction outputs)的交易模型

- 论文[1]证明了所有的交易都是跨片的
- RapidChain  $\leftrightarrow 96.3\%$
- Omniledger  $\leftrightarrow 90\% +$

2

### 现有的跨片交易处理

- 解决跨片交易发生时交易的处理和验证问题
- 现有方法：阶段提交方法 拆分交易法
- 没有实际减少大量跨片交易问题对可拓展性造成的影响

1

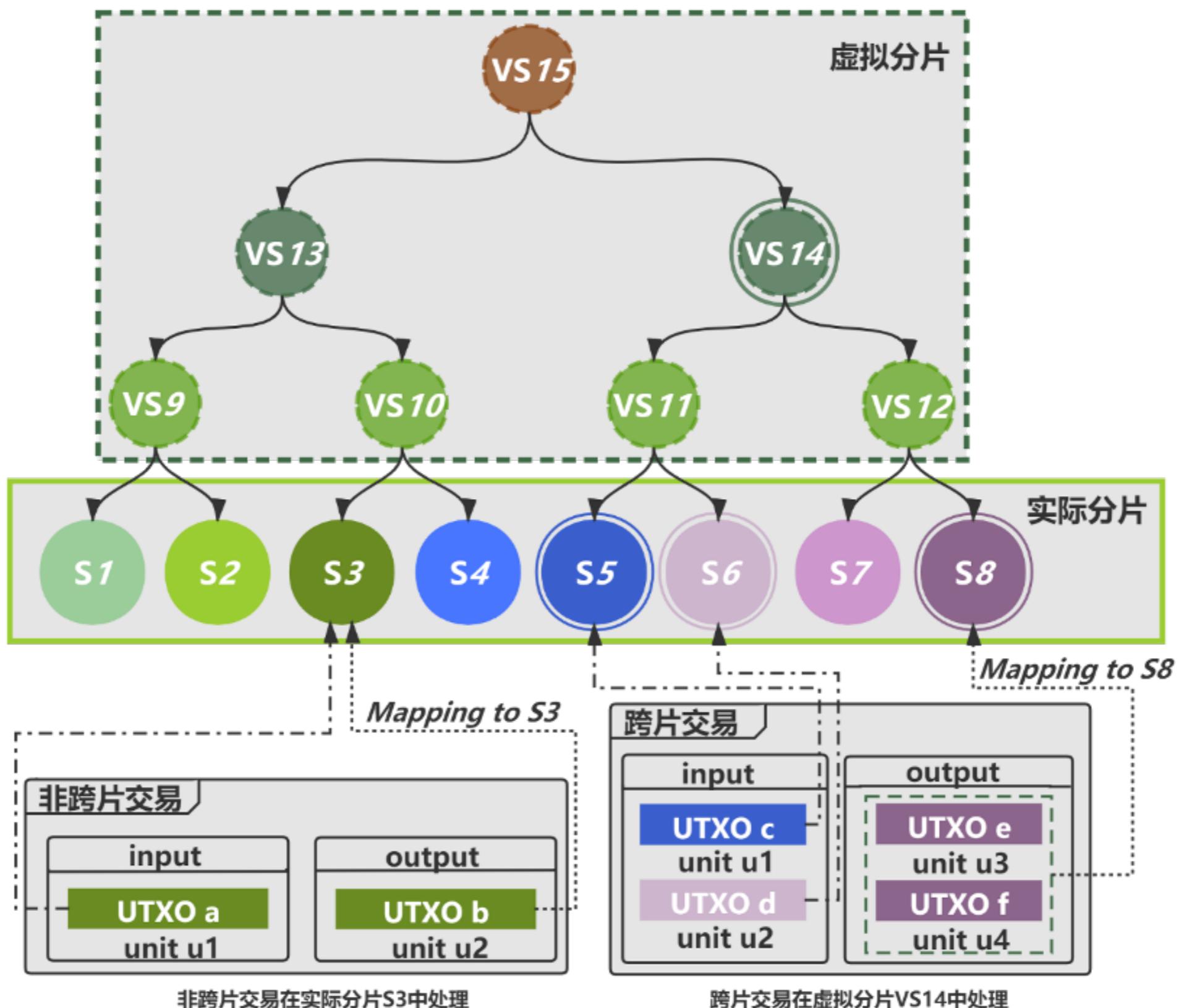
### 跨片交易的影响

- 多分片通信——通信开销
- 多分片参与——并行性
- 大量的必然发生的跨片交易会对系统的可拓展性造成影响，降低系统性能，不仅降低交易吞吐量，也增加了交易处理延迟



减少跨片交易对于分片方案的影响

## 重叠分片



## 重叠分片

### 分片结构

- 由实际分片和虚拟分片构成“存储树”分片结构

### 重叠率

- 基于哈希的分组维持重叠率的稳定和分片的均匀性

### 跨片交易处理方式

- 重叠节点处理交易
- 非重叠节点更新状态

### 路由方式

- 分片内路由
- 分片间路由

主要  
设计



### 分片重新配置

- 账本修剪
- 检查点

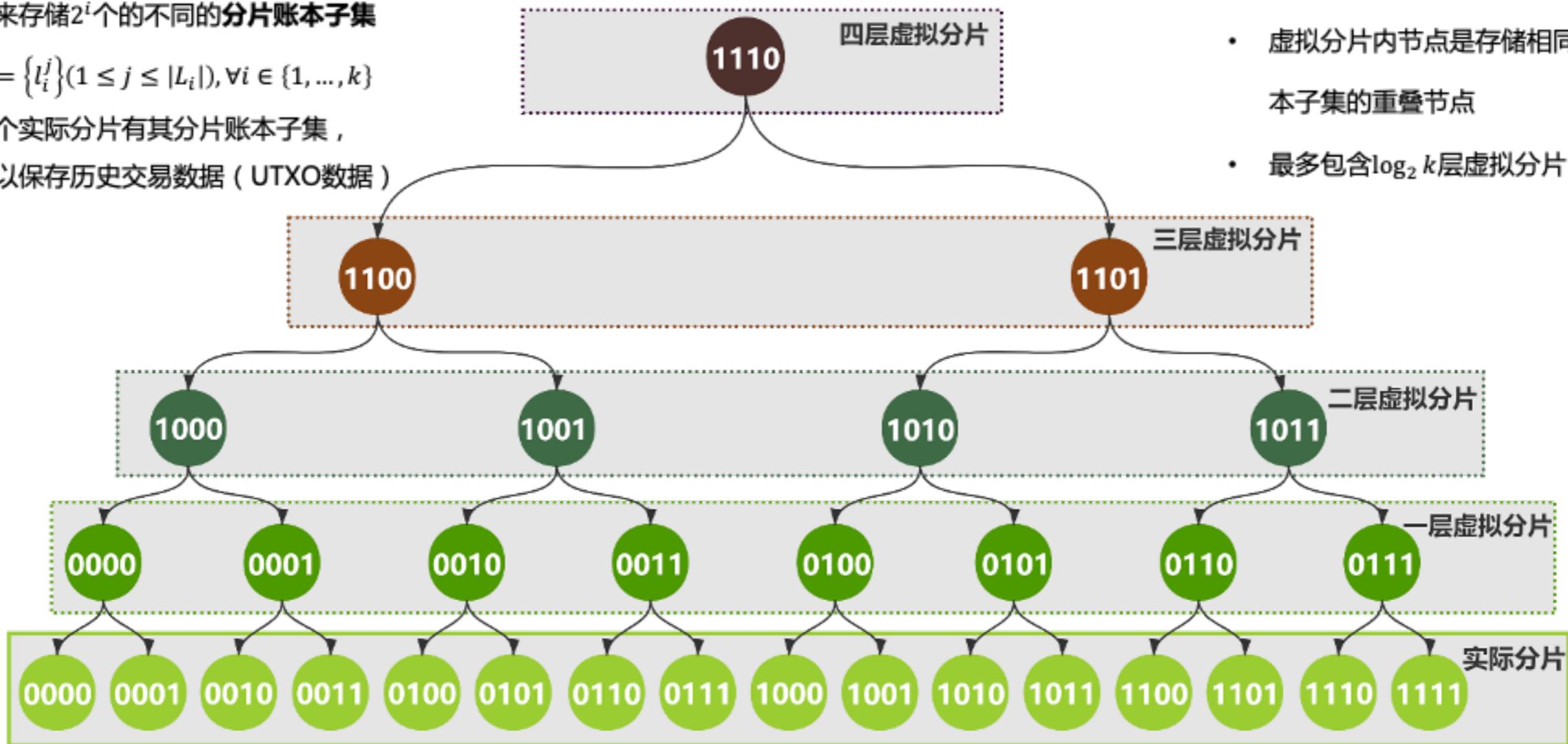
# 分片结构

## 实际分片-- $S_i, \forall i \in \{1, \dots, k\}$

- 由 $1, 2, 4, \dots, 2^i$ 倍存储节点组成，用来存储 $2^i$ 个的不同的分片账本子集  
 $L_i = \{l_i^j\} (1 \leq j \leq |L_i|), \forall i \in \{1, \dots, k\}$
- 每个实际分片有其分片账本子集，用以保存历史交易数据（UTXO数据）

## 虚拟分片-- $VS_j, \forall j \in \{1, \dots, k-1\}$

- 由实际分片间的重叠节点组成
- 虚拟分片内节点是存储相同账本子集的重叠节点
- 最多包含 $\log_2 k$ 层虚拟分片

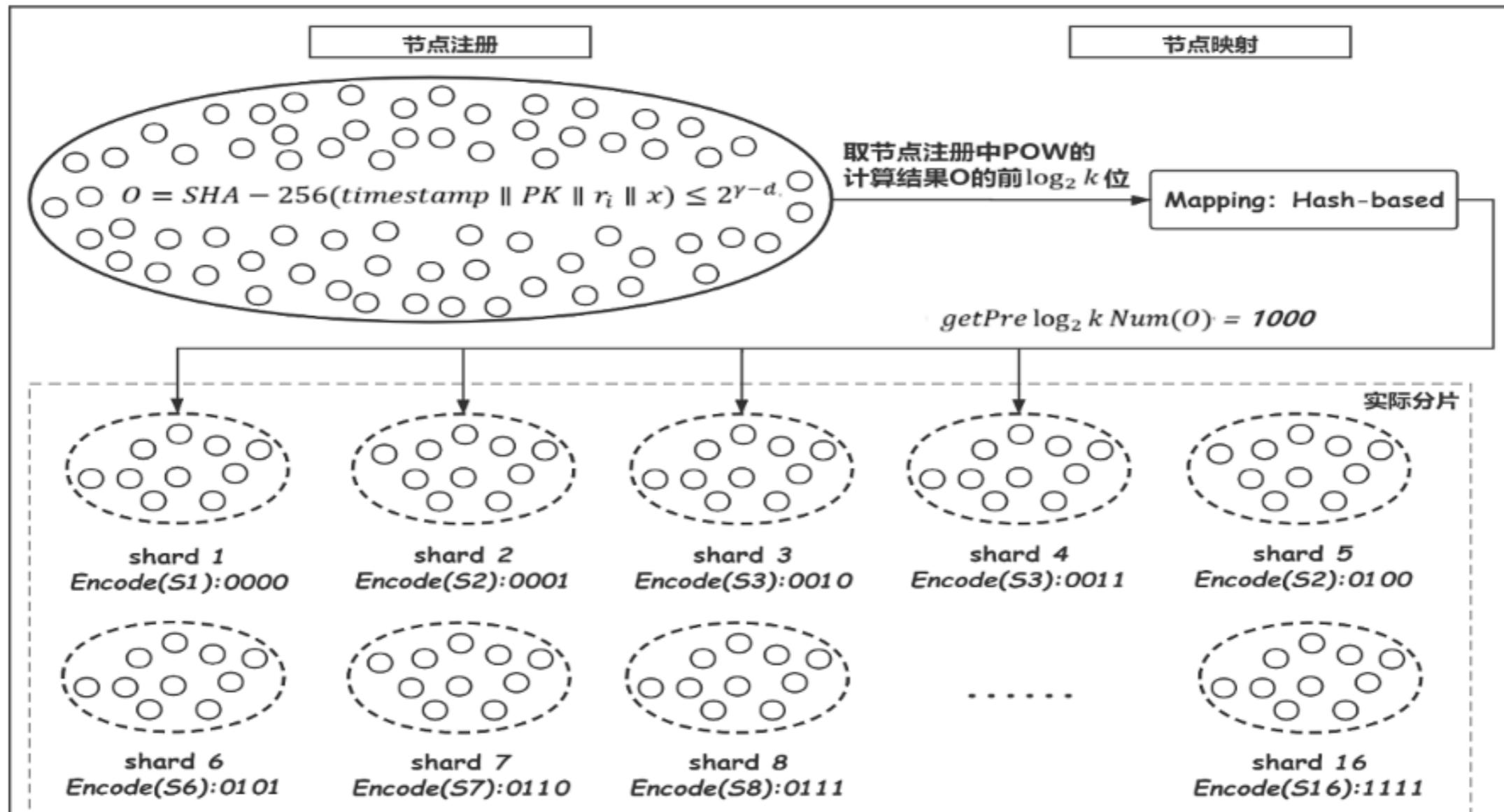


- 系统中的 $n$ 个节点被划分为 $k$ 个并行处理交易的实际分片 $S_i, \forall i \in \{1, \dots, k\}$
- $k$ 个分片并行处理不相交的交易子集 $X_i = \{x_i^j\} (1 \leq j \leq |X_i|)$
- 每个分片保存其分片账本子集 $L_i = \{l_i^j\} (1 \leq j \leq |L_i|)$

- 由实际分片层和虚拟分片层组成 “存储树” 分片结构
- 虚拟分片中的节点是其左右叶子实际分片中的重叠节点
- 实际分片和虚拟分片分别编码构成编码树 $T$

分片模型

# 节点注册与映射



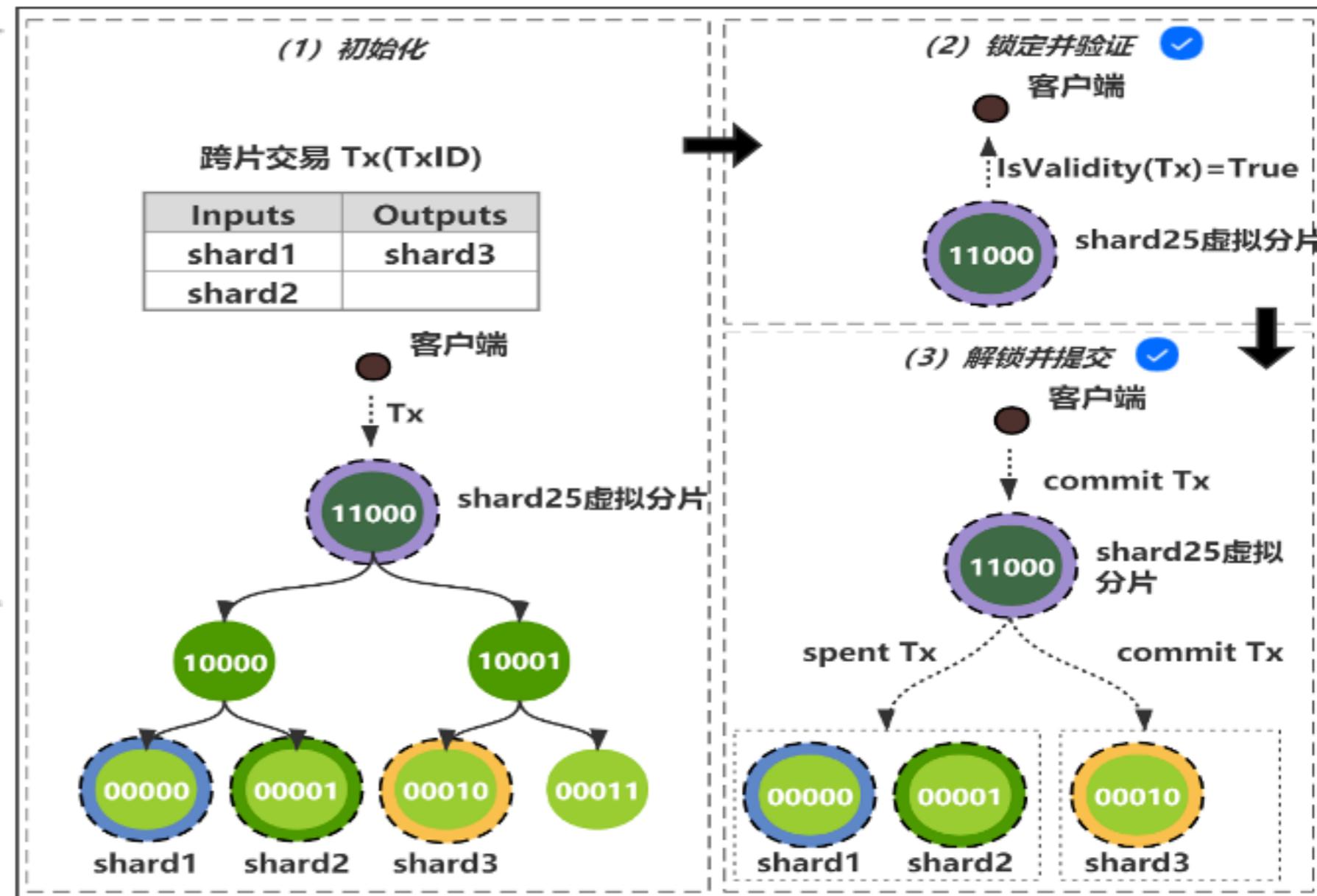
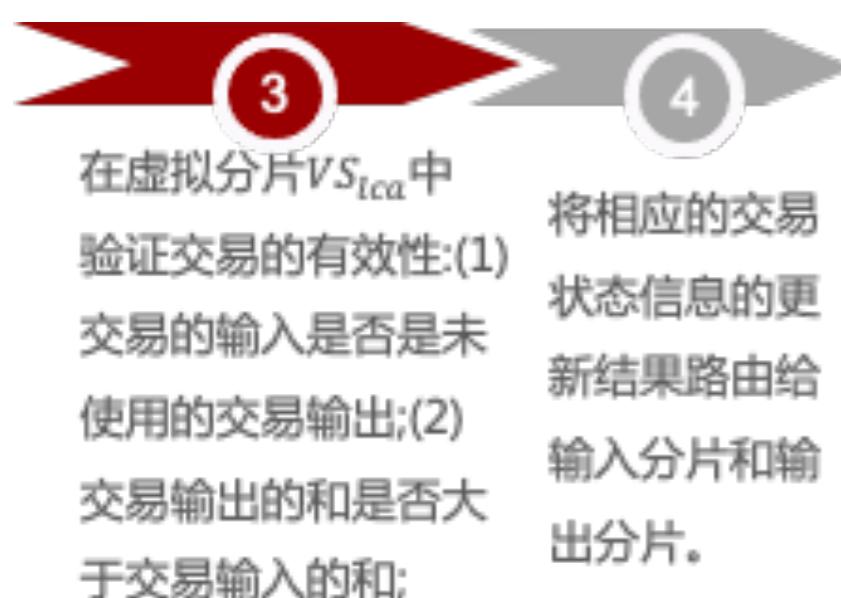
提供工作量证明难题的解决方案 $x$ ，使得 $O \leq 2^{Y-d}$ ；

取 $O$ 的前 $\log_2 k$ 位，映射到一个虚拟分片 $VS_j$ 中；

根据 $VS_j \rightarrow \{S_{actual}\}$ ，将节点映射至实际分片集合 $\{S_{actual}\}$ 中；

若无对应的虚拟分片编码，根据 $O$ 的后一个 $\log_2 k$ 位，将节点映射到实际分片 $S_i$ 中。

# 交易映射及处理



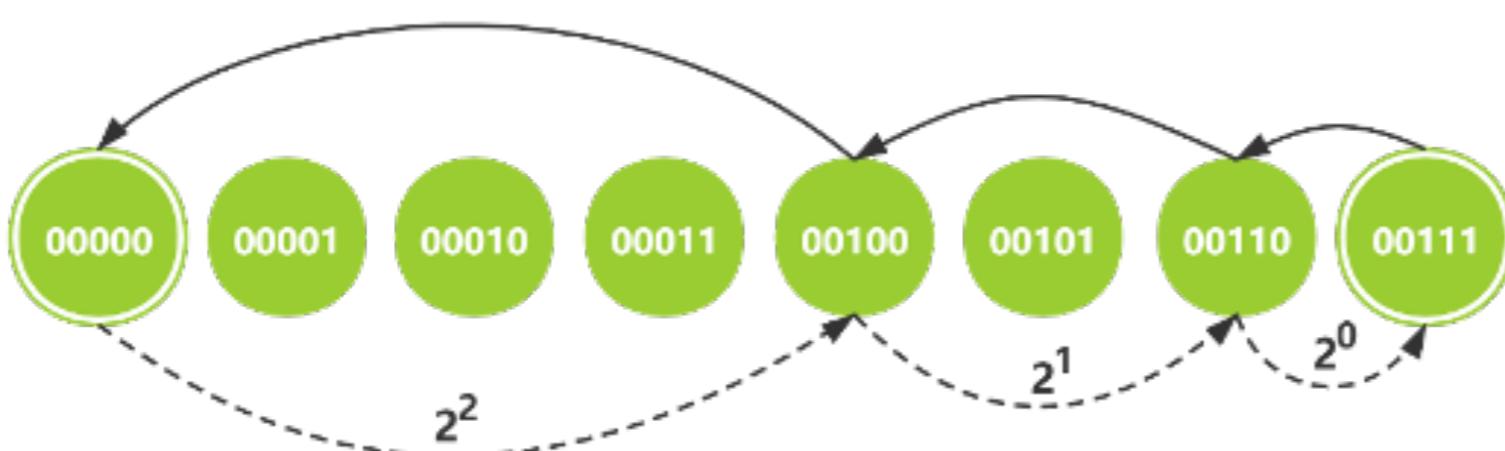
# 片间和片内路由

## 分片间路由

- 每个分片维护一个具有 $\log n$ 个记录的路由表
- 维护最接近的 $\log n$ 个分片中的 $\log \log(n)$ 个节点的网络信息
- 快速地建立连接进行通信，无需存储所有成员的网络信息

(1) 每个分片保存与其距离为 $2^i$ 的分片的网络信息

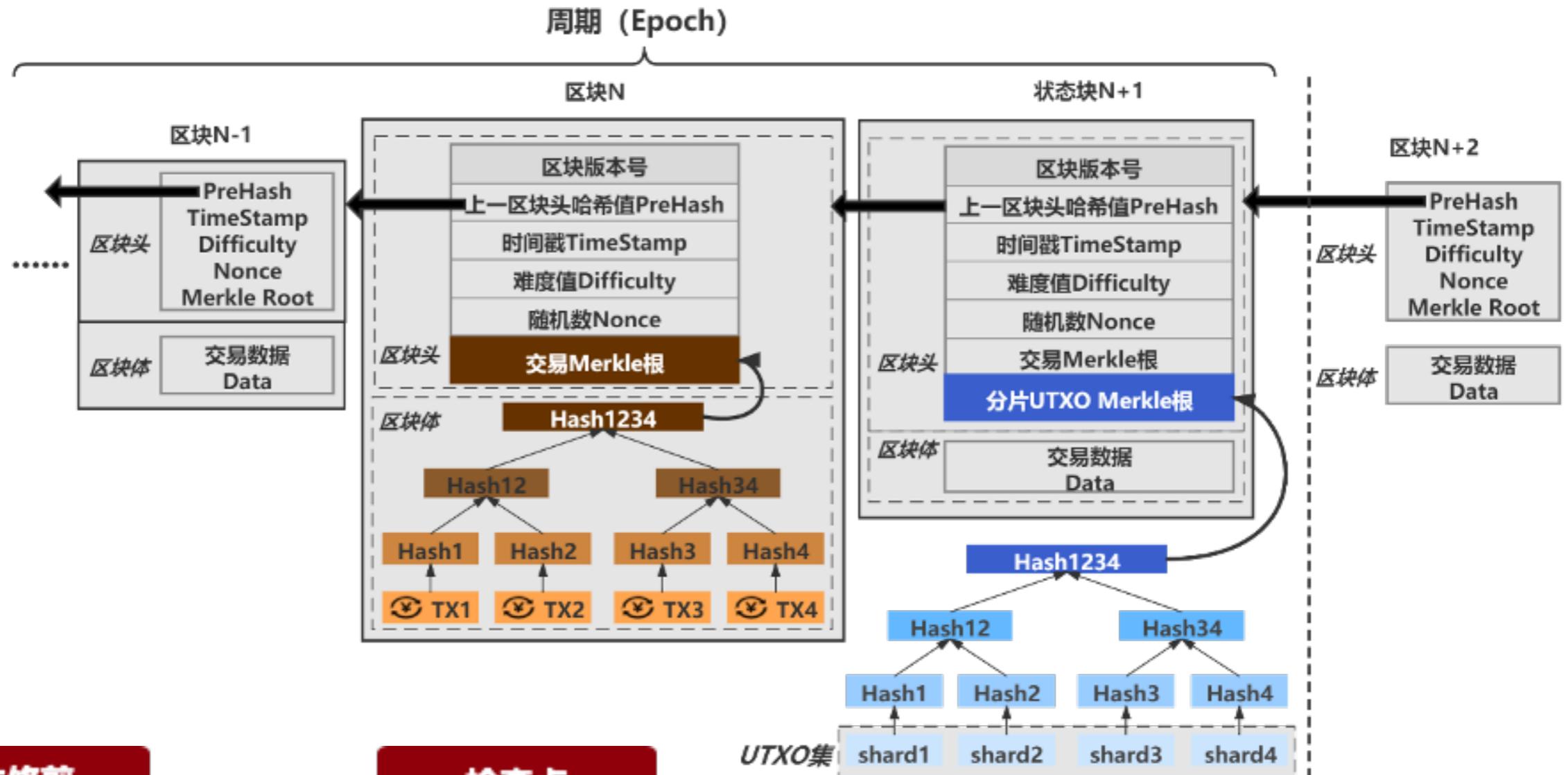
(2) 分片S0与分片S7间路由路径



## 分片内路由

- 维护其所属分片的成员网络信息，快速定位到分片中成员节点的IP地址
- 维护“存储树”的编码树T

# 分片重新配置



## 账本修剪

- 删除不必要的数据（例如已经花费的交易输出）来修剪账本
- 新节点 $new$ 只需从其需要存储的分片中下载未花费的交易集（UTXOs）
- 减少实际的存储开销

## 检查点

- 总结整个区块链状态的状态块
- 将每个分片的账本子集作为叶子节点存储在排序的默克尔树中
- 默克尔树的根哈希值放在检查点的块头中，作为状态块提交到链上

# 机制分析

## 性能分析

### 重叠分片机制

### 非重叠分片机制

存储开销  $\frac{|B|}{k}, \frac{2|B|}{k}, \frac{4|B|}{k}, \dots, 2^t \cdot \frac{|B|}{k}$

$|B|/k$

### 共识开销

$m \times V(tx, l)$

$m \times (p + q) \times V(tx, l)$

### 通信空间开销

$m \cdot |T| \cdot \log_2 k$

$m \cdot |T|(p + 2q)$

### 通信时间开销

2Times

3Times

### 通信空间开销

- 每笔交易所需的通信开销
- $k$ 为16时，当 $p$ 为2， $q$ 为2，通信空间开销比为1:1.5
- 重叠机制的通信空间仅为非重叠机制的66.7%

### 通信时间开销

- 把一个阶段的平均时间开销定义为*Times*
- 重叠分片机制与非重叠分片机制的通信时间开销比为1:1.5

### 存储开销

- $|B|$ 表示区块链的大小
- 全量节点的存储开销为 $|B|$

## 安全分析

### 周期安全性

- 利用二项分布函数来计算每个周期（epoch）的失败概率
- 数量级为 $10^{-11}$ 和 $10^{-12}$ ，安全性并没有太大的影响

### 共识安全性

- “虚拟分片”中参与共识的节点数  
量 $m'' = m$
- 恶意节点比例不变，低于共识安全  
比例

### 共识开销

- 交易的输入数为 $p$ ，输出数为 $q$
- $V(tx, l)$ 为一次验证操作
- 当 $p$ 为2， $q$ 为2时，共识开销比为1:4，重叠机制的共识开销仅为非重叠机制的25%

## 划分分片

- 根据地址ID将节点按照映射规则映射到分片中
- 记录节点所属分片组： $ID\_SHARD = \{ID \rightarrow [shard\_number]\}$
- 记录分片内节点地址ID： $SHARD\_ID = \{shard\_number \rightarrow [ID]\}$

## 构造图

- 无向连通图 $G = \{E_i\}, i \in N^*$ 作为分布式节点网络的图结构
- 每个节点最多共有300个连接，其中200个连接取为片内连接，其余100个连接取为片间连接

## 边划分

- 边的权重 $W_i$ 取值阈值[0.2, 3.1]
- 以中位数为界，为片内连接和片间连接在赋权重，用以表征通信开销
- 权重指标采用权重矩阵 $M_{weight}$ 进行存储

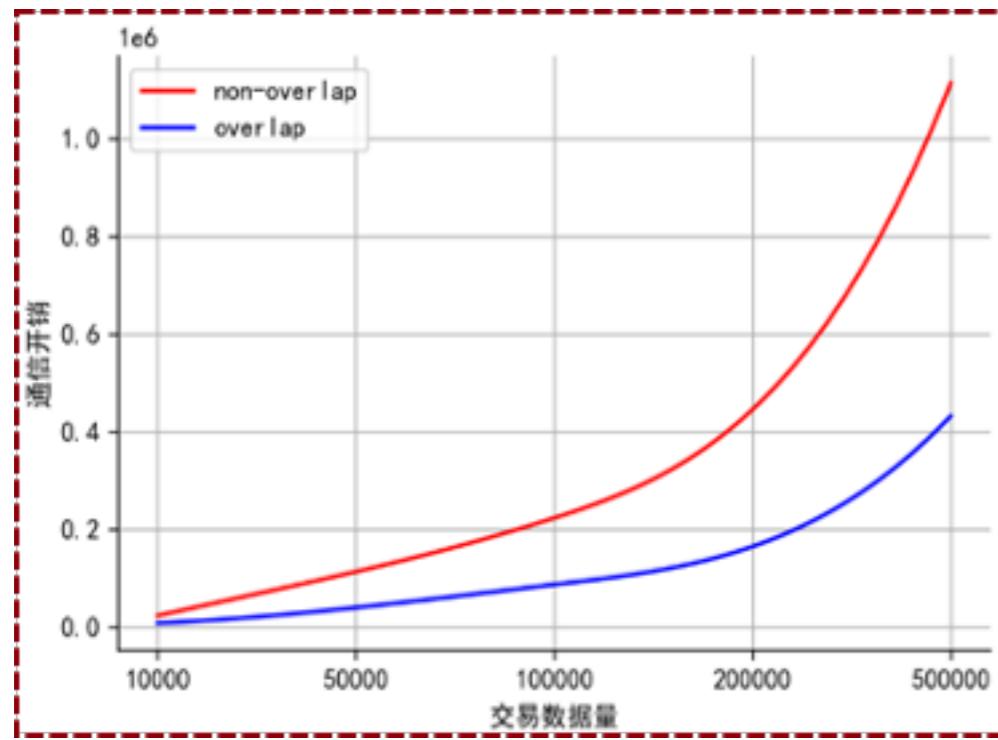
## 交易映射处理

- 数据指标：交易处理过程中的传输路径对应的边权重累加值 $\sum_{i=1}^N W_i$ 和跨片率Q
- 根据交易的字段 $vin\_sz$ 抽取节点 $Input_1, Input_2, \dots, Input_{vin\_sz}$ 作为路由交易的输入节点，根据 $tx\_index$ 抽取节点 $Output$ 作为路由路径传输的输出节点，计算从输入节点到输出节点的最短传输路径对应的边权重累加值 $\sum_{i=1}^N W_i$
- 根据 $txhash$ 字段将交易随机映射到分片 $S_{out}$ 中处理，根据 $vin\_sz$ 选定输入分片 $S_{in}^1, S_{in}^2, \dots, S_{in}^{vin\_sz}$ 。分别在非重叠分片方案和重叠分片方案的实验对照组中统计该笔交易是否为跨片交易，计算跨片率Q = 跨片交易数/交易总数

# 实验分析

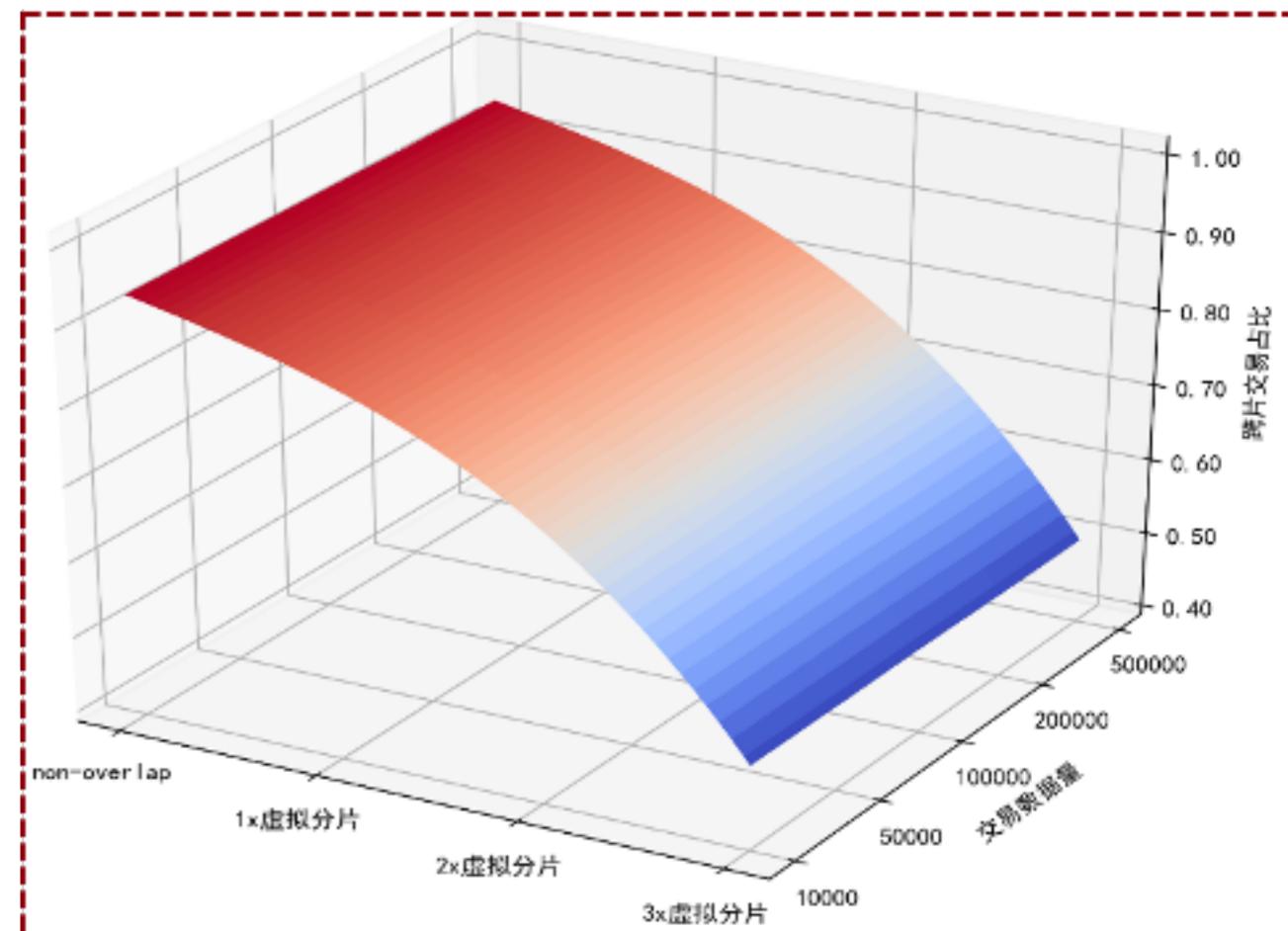
## 实验一结果

- 与非重叠分片机制相比，重叠分片机制的平均通信开销减少了**63.30%**

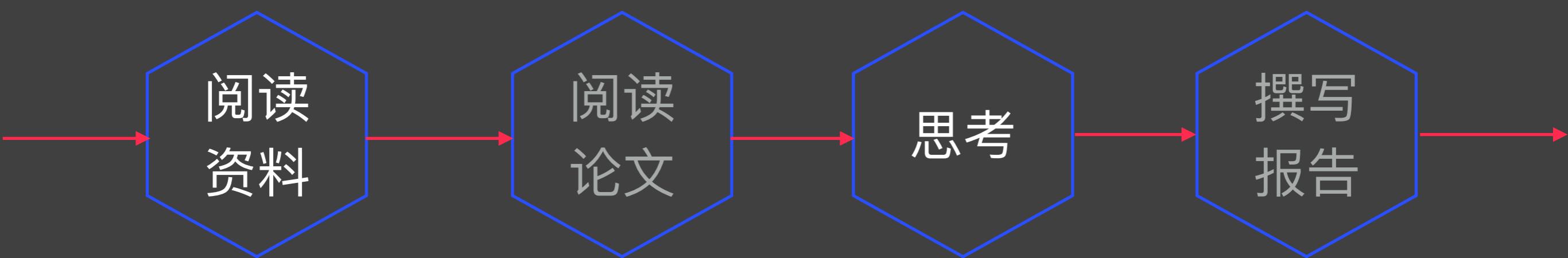


## 实验二结果

- 各种模型中跨片率不会随交易数量的增加发生明显变化
- 非重叠分片模型中跨片率的平均值为**93.77%**
- 一层、两层和三层虚拟分片层的重叠分片模型中Q平均值分别为**87.44%、75.00%、50.01%**



# 课后作业



# Homework

## 课后阅读建议



阅读全部章节

<https://www.8btc.com/book/281955>

<https://github.com/bitcoinbook/bitcoinbook>

<https://www.bitcoin.org/>

A screenshot of the official Bitcoin website at bitcoin.org/en/. The page features a large headline: 'Bitcoin is an innovative payment network and a new kind of money.' Below this are three main calls-to-action: 'Get started with Bitcoin', 'Choose your wallet', and 'Buy Bitcoin'. A large blue box on the right side contains the text: '介绍性内容', 'Introduction', and 'White Paper'.

开发方面内容  
Developer Guides、Reference、  
Examples、Learning Resources

# 谢谢！

孙惠平

[sunhp@ss.pku.edu.cn](mailto:sunhp@ss.pku.edu.cn)