掌握比特币

北京大学 软件与微电子学院
School of Software and Microelectronics, Peking University

Huiping Sun(孙惠平)
sunhp@ss.pku.edu.cn

Course Overview

本次课程内容

1 监管

- 共识
- 社区
- 分叉
- 政府

2 平台

- 平台
- 博彩
- 随机源
- 预测市场

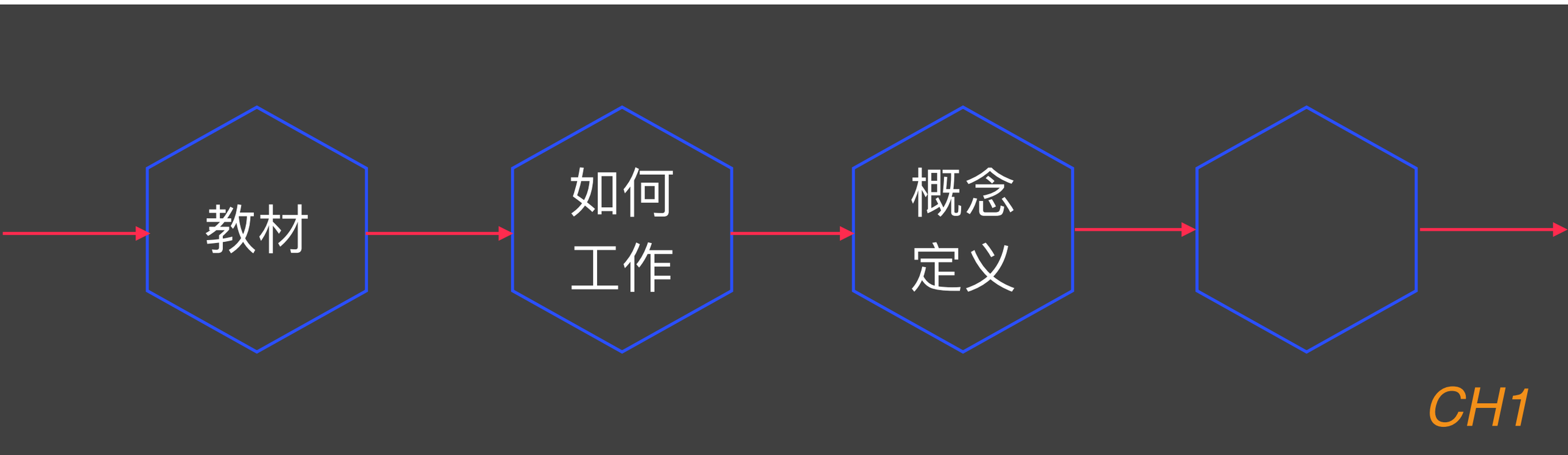
3 代币

- 产生
- 例子
- 共同挖矿
- 侧链

4 应用

- 智能合约
- 面临挑战
- 国际贸易
- 保险

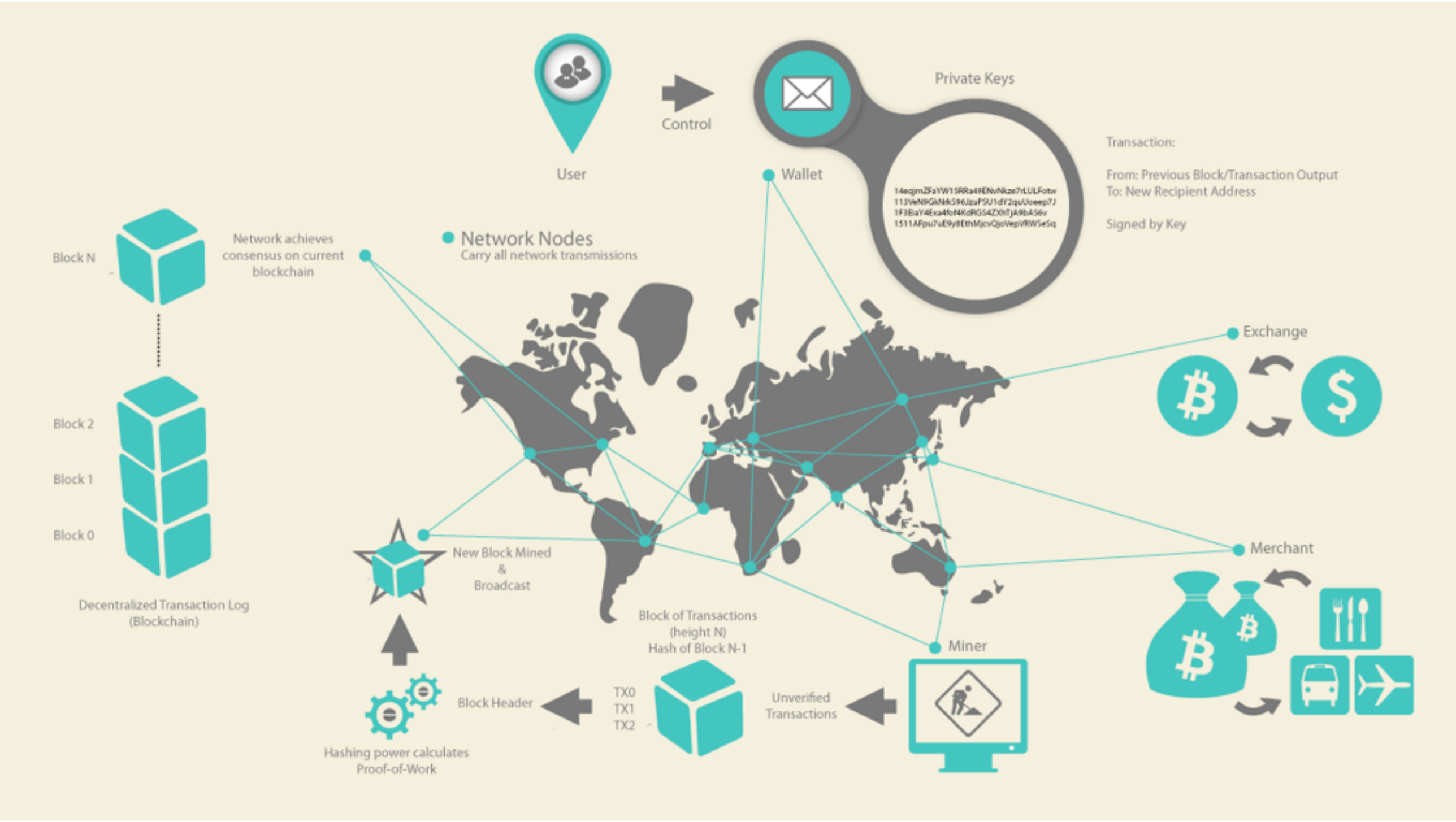
簡介



参考书


O'REILLY
第2版
Mastering
Bitcoin
精通比特币（影印版）
Andreas M. Antonopoulos 著
东南大学出版社
REILLY
原书第2
精通区块链编程
加密货币原理、方法和应用开发
Mastering Bitcoin: Programming the Open Blockchain
[希] Andreas M. Antonopoulos 著
郭理靖 李国鹏 李卓 译
乔延宏 邵周 Higer 审校
机械工业出版社
China Machine Press

Bitcoin如何工作



概念定义

构成数字货币生态系统基础概念和技术的总称

比特币网络中参与者存储和传输的货币单位

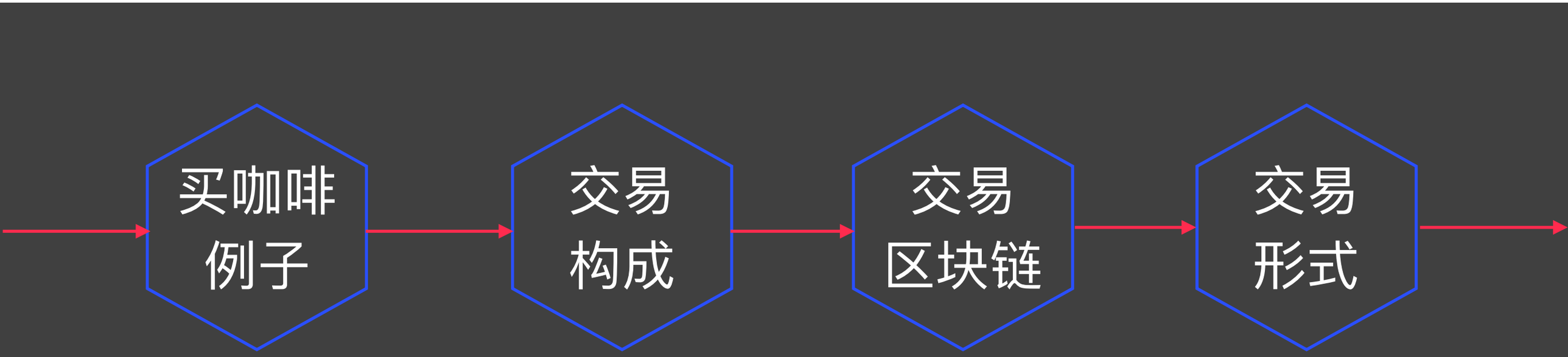
比特币是虚拟的，本身也不是简单数据化的

用户通过网络进行比特币进行转账和可以做到和传统货币一样的事情

比特币隐含在汇款方到收款方的转账交易中，用户用自己私钥来证明

传统银行依靠发行和结算，比特币依靠挖矿

基本原理



交易例子 – 买咖啡

```
bitcoin:1GdK9UzpHBzqzX2A9JFP3Di4weBwqgmoQA?
amount=0.015&
label=Bob%27s%20Cafe&
message=Purchase%20at%20Bob%27s%20Cafe
```

```
A bitcoin address: "1GdK9UzpHBzqzX2A9JFP3Di4weBwqgmoQA"
The payment amount: "0.015"
A label for the recipient address: "Bob's Cafe"
A description for the payment: "Purchase at Bob's Cafe"
```

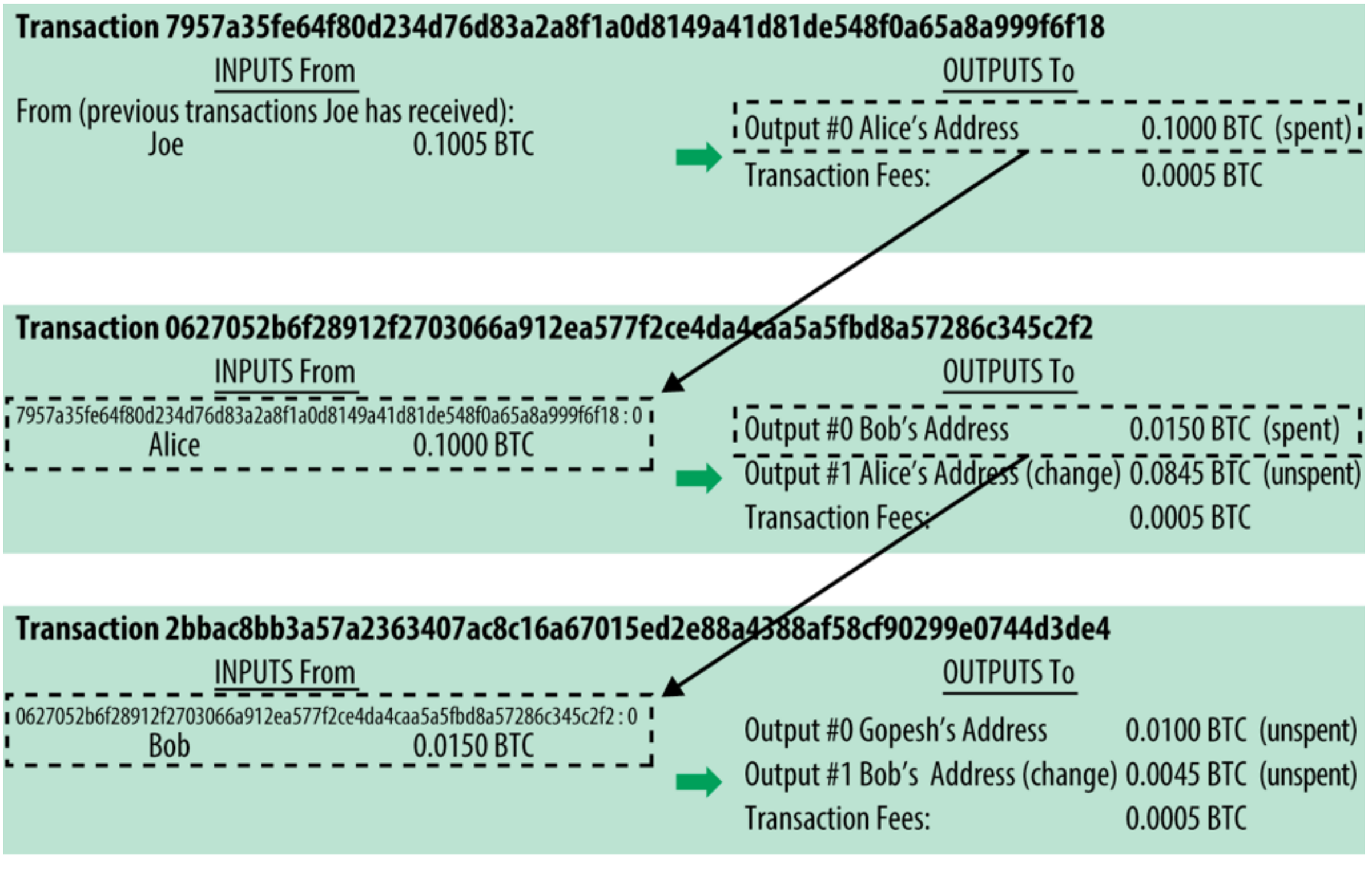
交易构成

Transaction as Double-Entry Bookkeeping

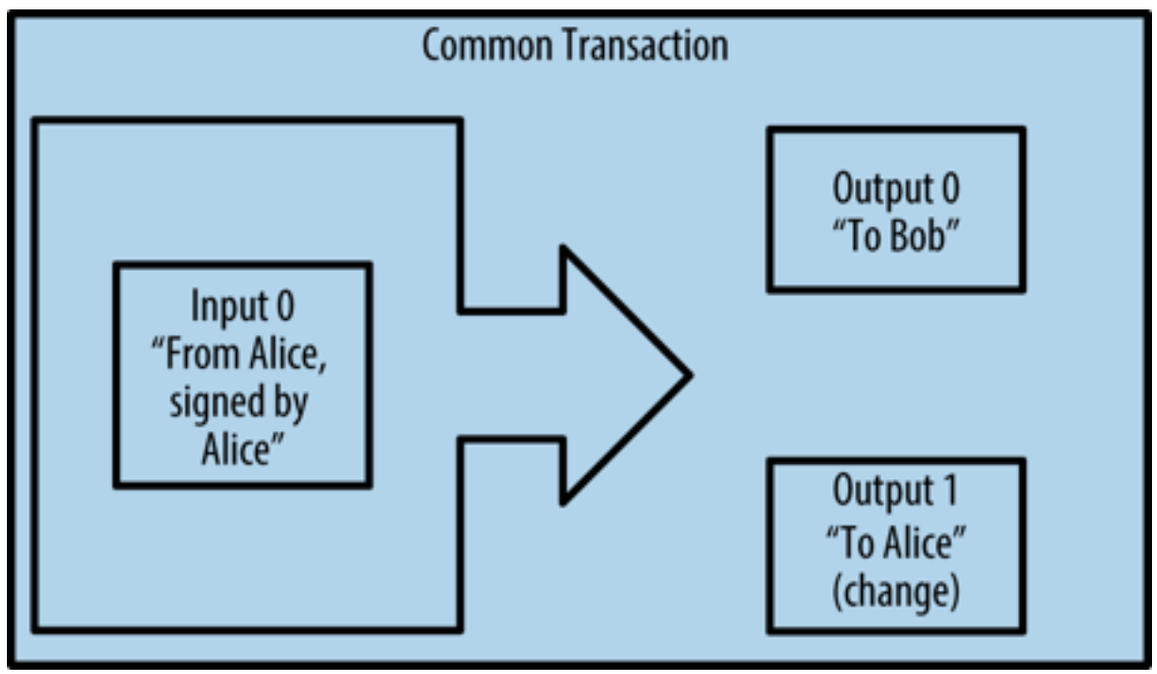
Inputs	Value	Outputs	Value
Input 1	0.10 BTC	Output 1	0.10 BTC
Input 2	0.20 BTC	Output 2	0.20 BTC
Input 3	0.10 BTC	Output 3	0.20 BTC
Input 4	0.15 BTC		
Total Inputs:	0.55 BTC	Total Outputs:	0.50 BTC

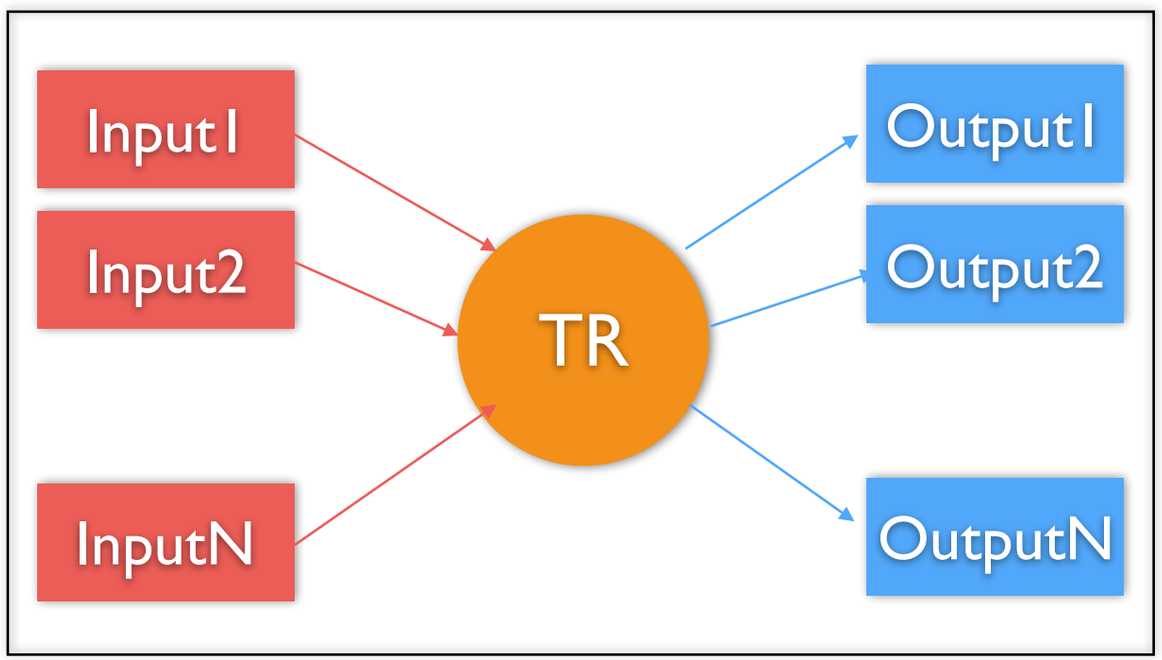
	Inputs	*0.55 BTC*
-	*Outputs*	*0.50 BTC*
	Difference	*0.05 BTC (implied transaction fee)*

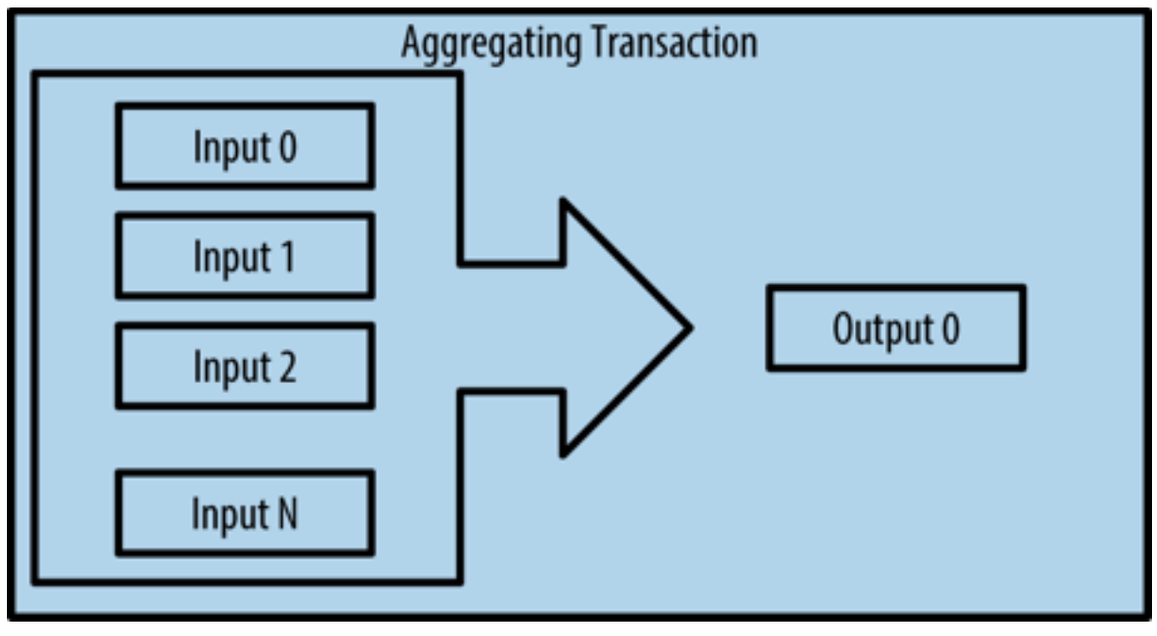
交易链

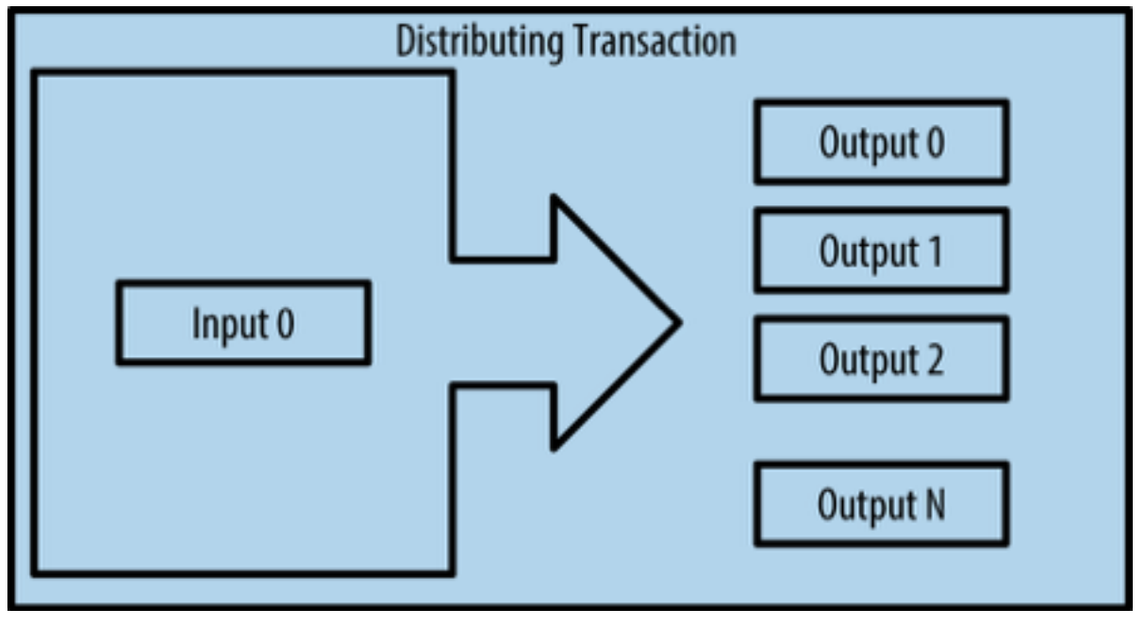

Transaction 7957a35fe64f80d234d76d83a2a8f1a0d8149a41d81de548f0a65a8a999f6f18
INPUTS From
From (previous transactions Joe has received):
Joe 0.1005 BTC
OUTPUTS To
Output #0 Alice's Address 0.1000 BTC (spent)
Transaction Fees: 0.0005 BTC
Transaction 0627052b6f28912f2703066a912ea577f2ce4da4caa5a5fbd8a57286c345c2f2
INPUTS From
7957a35fe64f80d234d76d83a2a8f1a0d8149a41d81de548f0a65a8a999f6f18 : 0
Alice 0.1000 BTC
OUTPUTS To
Output #0 Bob's Address 0.0150 BTC (spent)
Output #1 Alice's Address (change) 0.0845 BTC (unspent)
Transaction Fees: 0.0005 BTC
Transaction 2bbac8bb3a57a2363407ac8c16a67015ed2e88a4388af58cf90299e0744d3de4
INPUTS From
0627052b6f28912f2703066a912ea577f2ce4da4caa5a5fbd8a57286c345c2f2 : 0
Bob 0.0150 BTC
OUTPUTS To
Output #0 Gopesh's Address 0.0100 BTC (unspent)
Output #1 Bob's Address (change) 0.0045 BTC (unspent)
Transaction Fees: 0.0005 BTC

交易形式


Common Transaction
Input 0
"From Alice, signed by Alice"
Output 0
"To Bob"
Output 1
"To Alice" (change)


Input1
Input2
InputN
TR
Output1
Output2
OutputN


Aggregating Transaction
Input 0
Input 1
Input 2
Input N
Output 0


Distributing Transaction
Input 0
Output 0
Output 1
Output 2
Output N

交易描述

Transaction View information about a bitcoin transaction

0627052b6f28912f2703066a912ea577f2ce4da4caa5a5fbd8a57286c345c2f2

1Cdid9KFAaatwczBwBttQcwXYCpvK8h7FK (0.1 BTC - Output)

1GdK9UzpHBzqzX2A9JFP3Di4weBwqgmoQA - (Unspent) 0.015 BTC

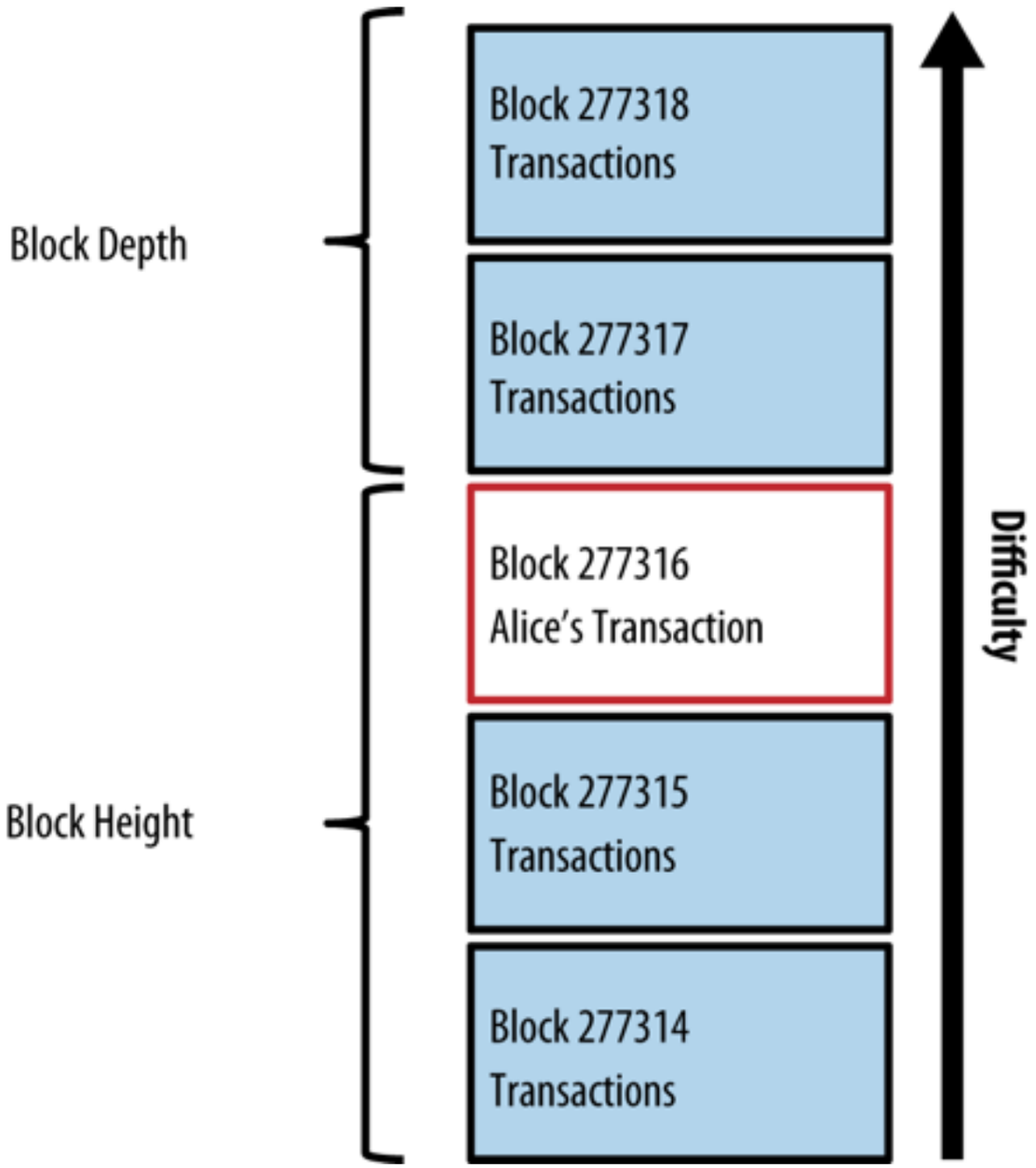
1Cdid9KFAaatwczBwBttQcwXYCpvK8h7FK - (Unspent) 0.0845 BTC

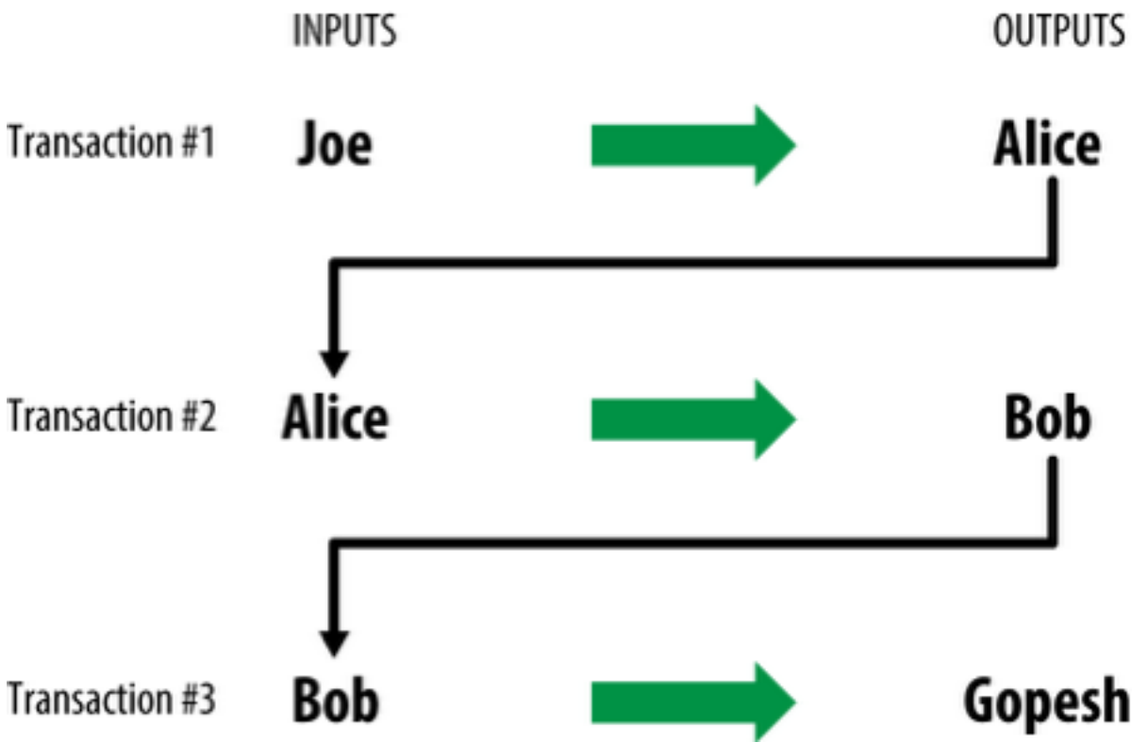
97 Confirmations | 0.0995 BTC

Summary	
Size	258 (bytes)
Received Time	2013-12-27 23:03:05
Included In Blocks	277316 (2013-12-27 23:11:54 +9 minutes)

Inputs and Outputs	
Total Input	0.1 BTC
Total Output	0.0995 BTC
Fees	0.0005 BTC
Estimated BTC Transacted	0.015 BTC

区块和链条

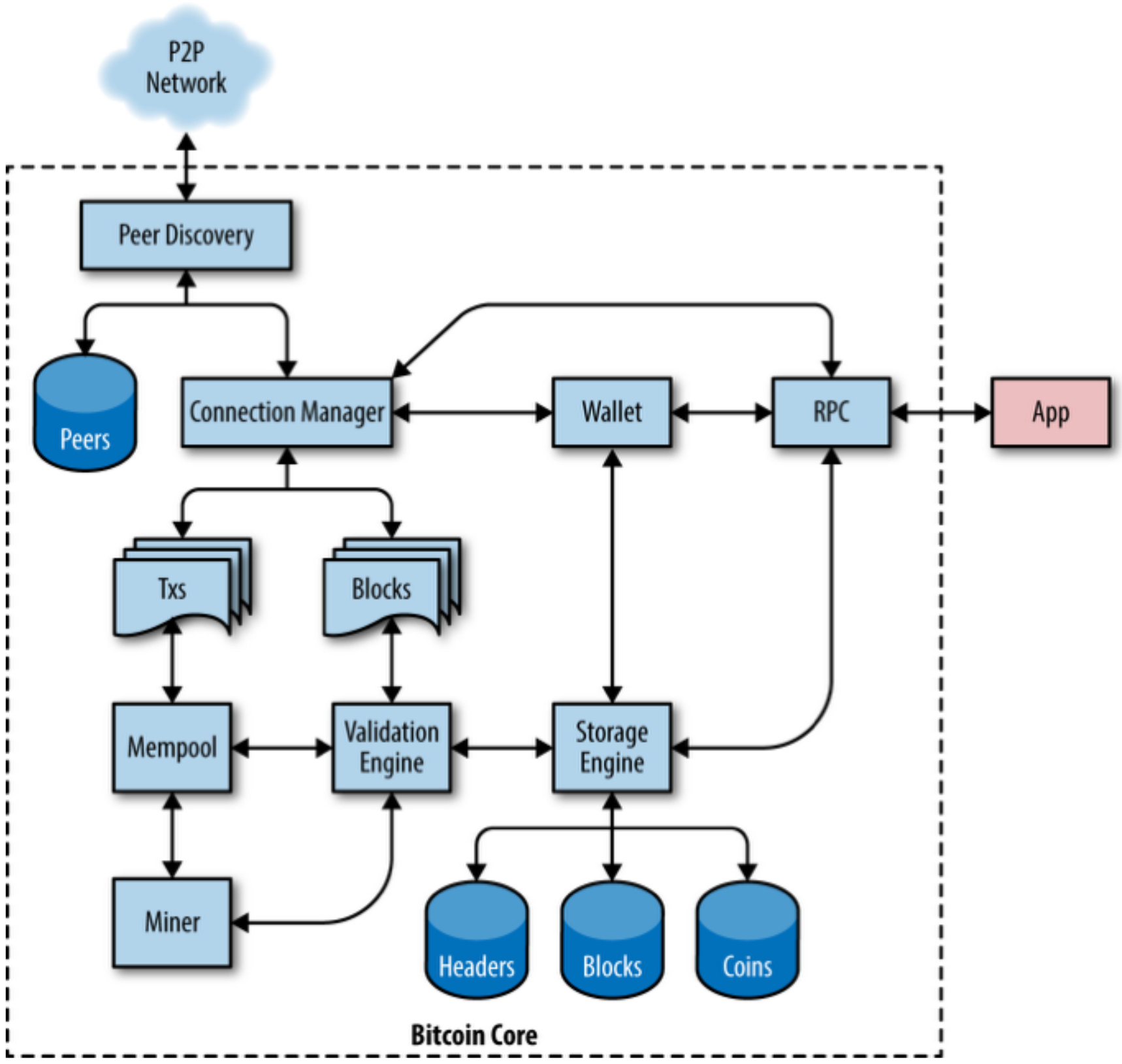

Block Depth
Block Height
Block 277318
Transactions
Block 277317
Transactions
Block 277316
Alice's Transaction
Block 277315
Transactions
Block 277314
Transactions
Difficulty


INPUTS
OUTPUTS
Transaction #1
Joe
Alice
Transaction #2
Alice
Bob
Transaction #3
Bob
Gopesh

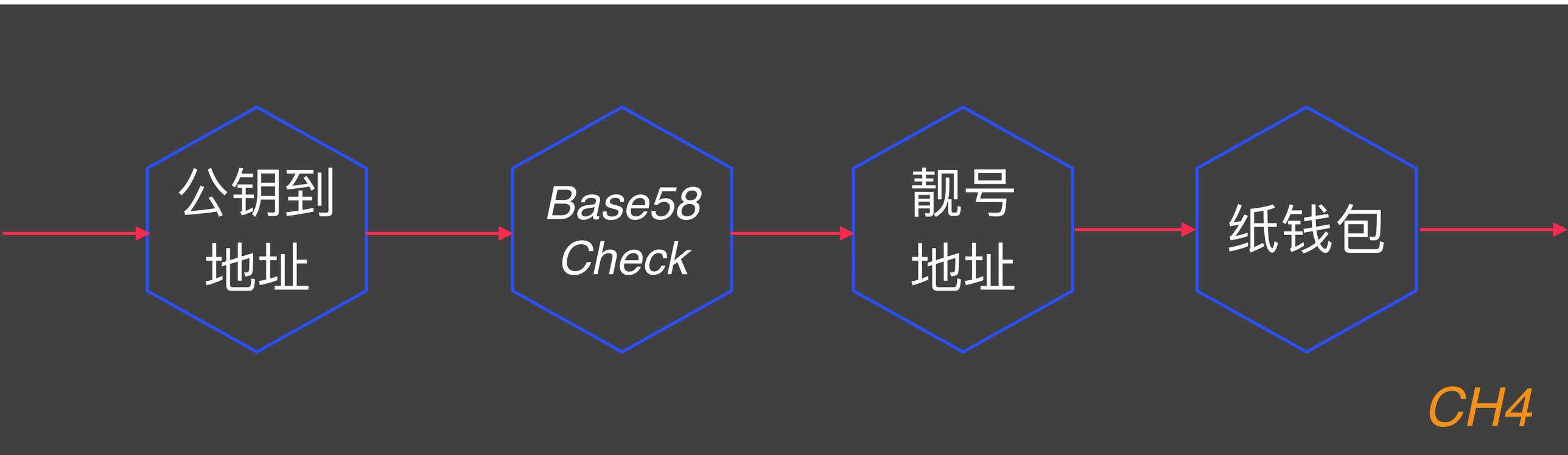
核心客户端

CH3

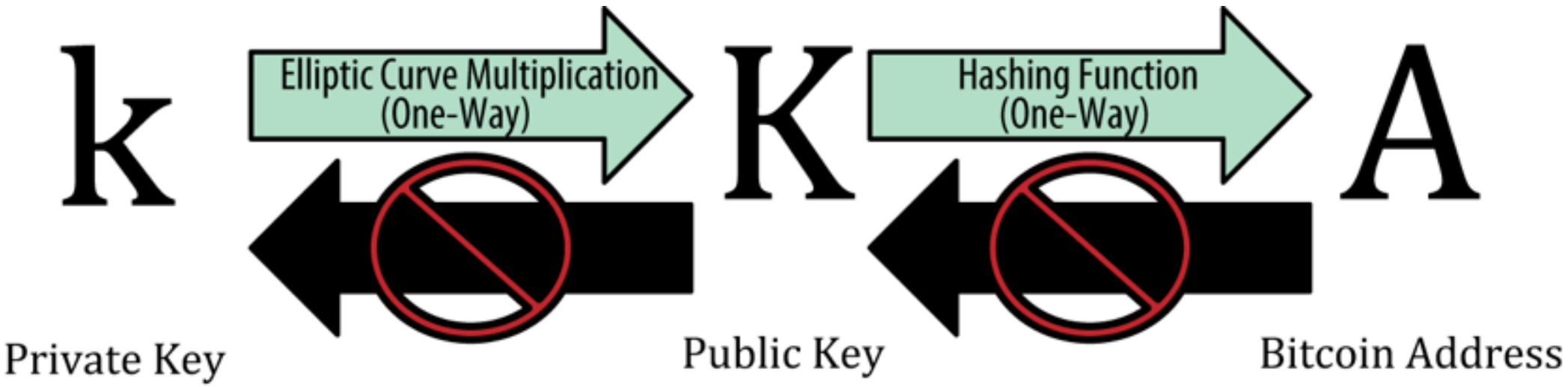
Bitcoin核心架构



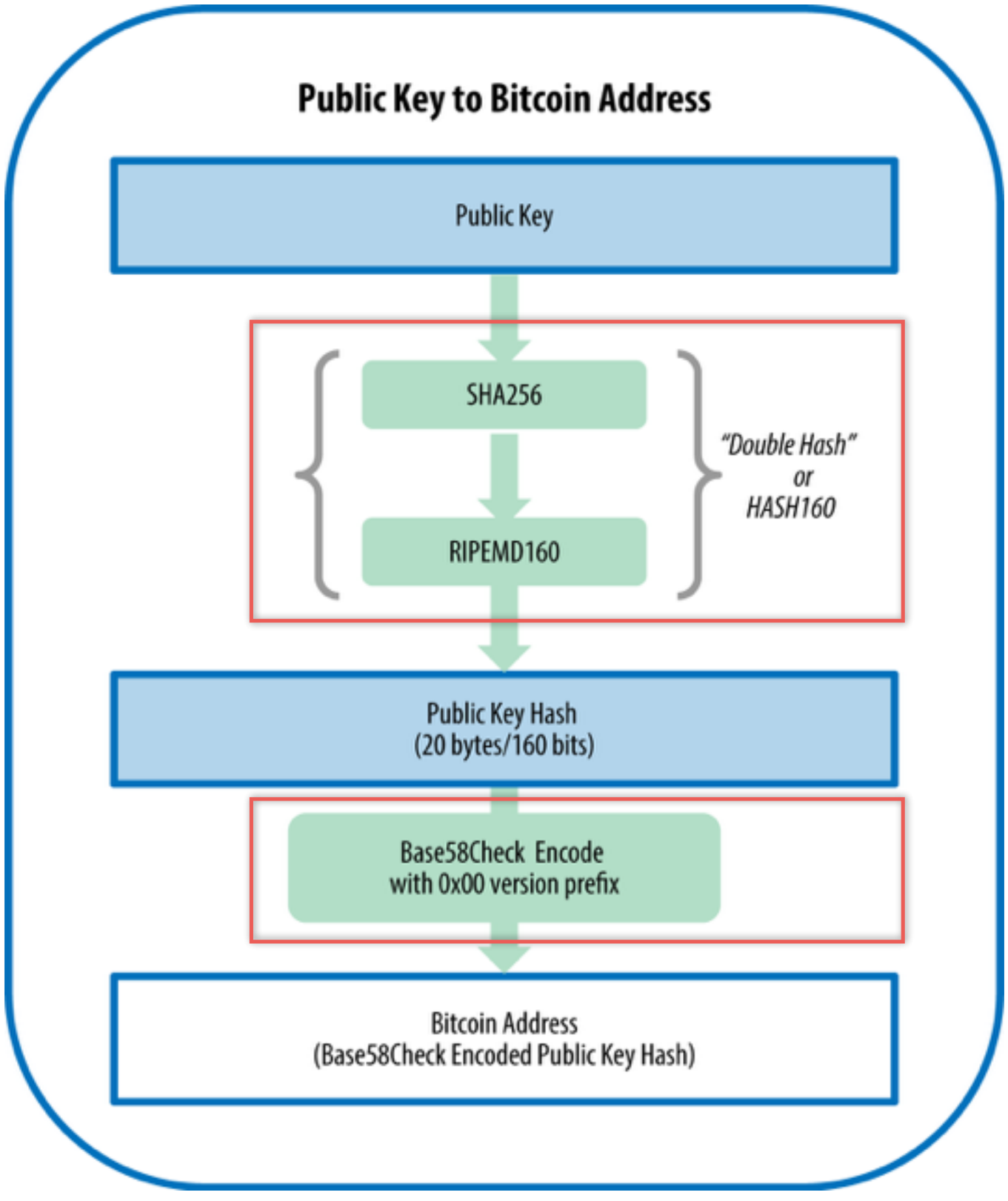
密钥和地址



私钥、公钥、地址


k
Elliptic Curve Multiplication
(One-Way)
K
Hashing Function
(One-Way)
A
Private Key
Public Key
Bitcoin Address

公钥到地址


Public Key to Bitcoin Address
Public Key
SHA256
RIPEMD160
"Double Hash"
or
HASH160
Public Key Hash
(20 bytes/160 bits)
Base58Check Encode
with 0x00 version prefix
Bitcoin Address
(Base58Check Encoded Public Key Hash)

Base58Check编码

Base64

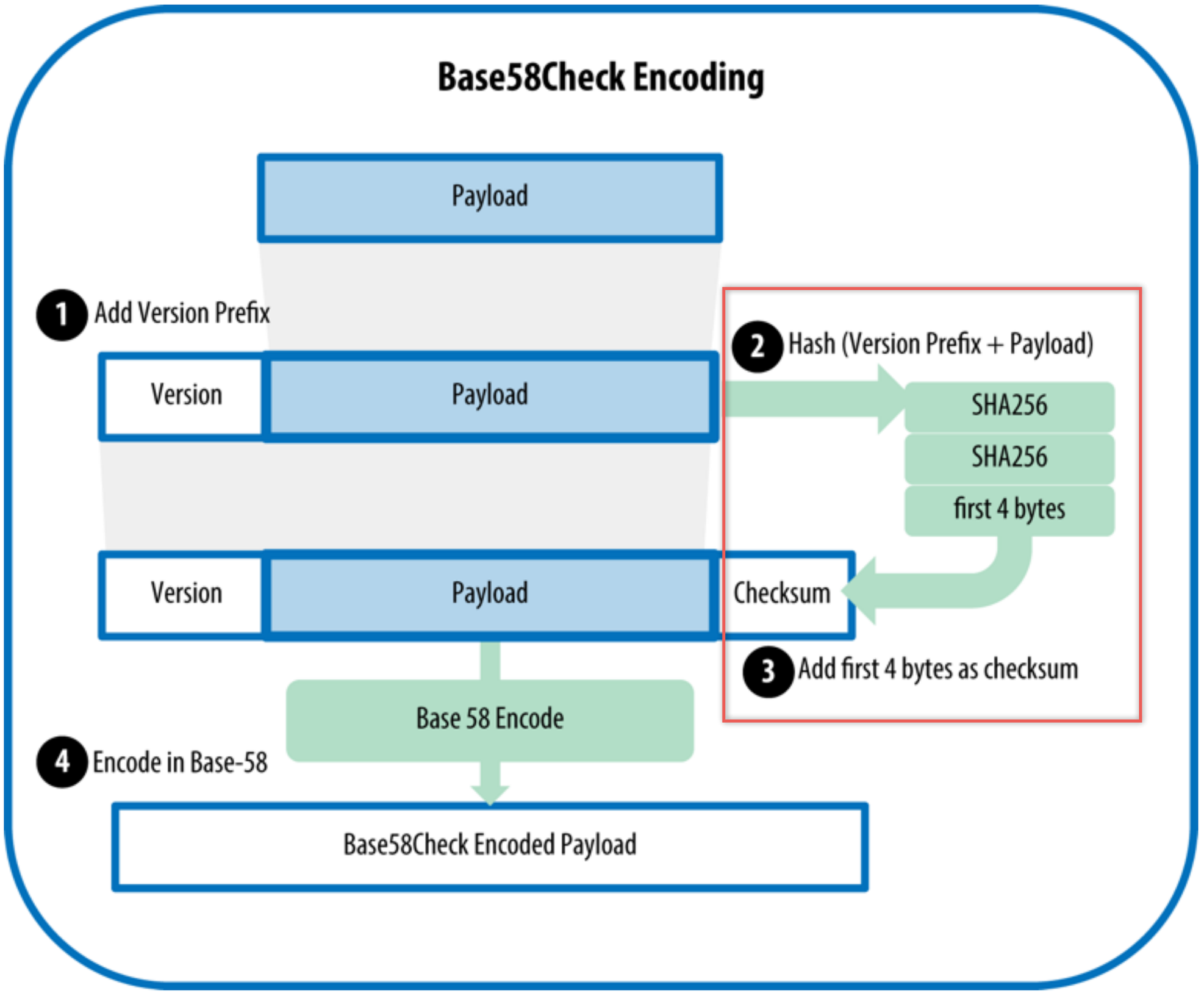
- 大写字母
- 小写字母
- 数字
- +、/

Base58

- 0和o
- I和l

Base58Check

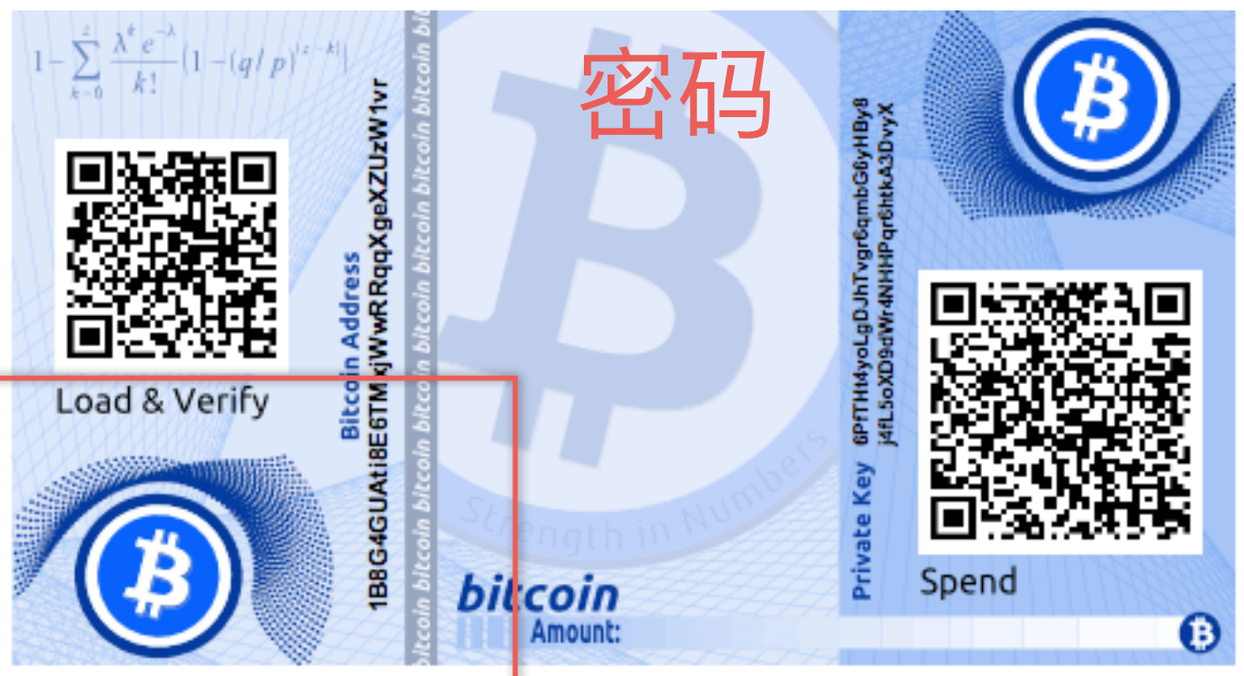
- 检验和



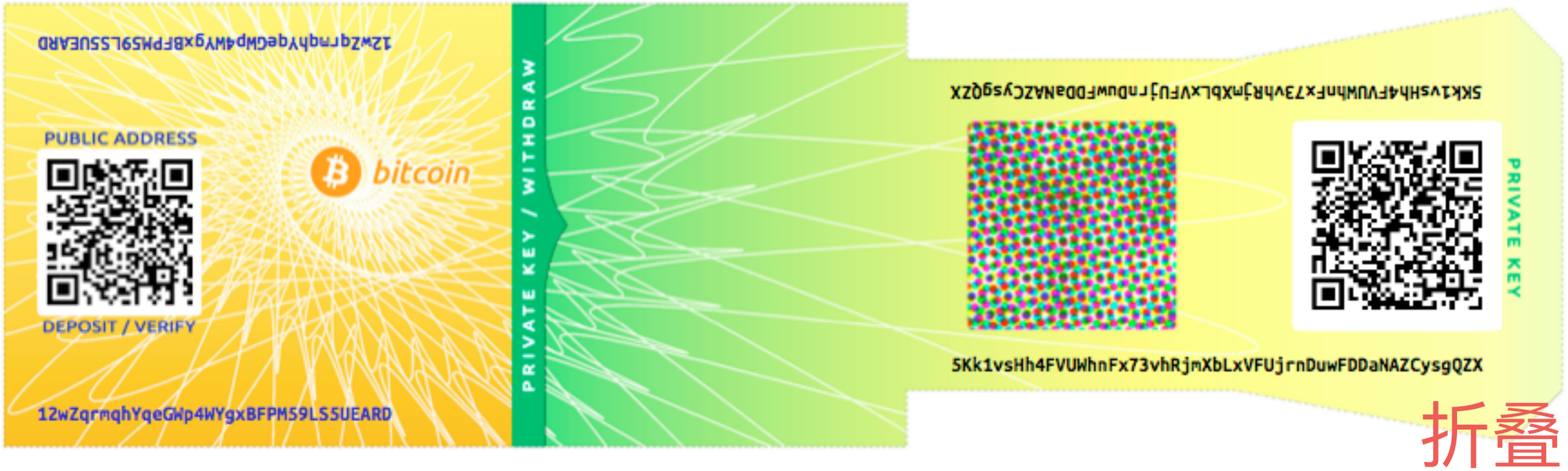
靓号地址

Length	Pattern	Frequency	Average search time
1	1K	1 in 58 keys	< 1 milliseconds
2	1Ki	1 in 3,364	50 milliseconds
3	1Kid	1 in 195,000	< 2 seconds
4	1Kids	1 in 11 million	1 minute
5	1KidsC	1 in 656 million	1 hour
6	1KidsCh	1 in 38 billion	2 days
7	1KidsCha	1 in 2.2 trillion	3–4 months
8	1KidsChar	1 in 128 trillion	13–18 years
9	1KidsChari	1 in 7 quadrillion	800 years
10	1KidsCharit	1 in 400 quadrillion	46,000 years
11	1KidsCharity	1 in 23 quintillion	2.5 million years

纸钱包



Public address	Private key (WIF)
1424C2F4bC9JidNjjTUZCbUxv6Sa1Mt62x	5J3mBbAH58CpQ3Y5RNJpUKPE62SQ5tfcvU2JpbnkeyhfsYB1Jcn



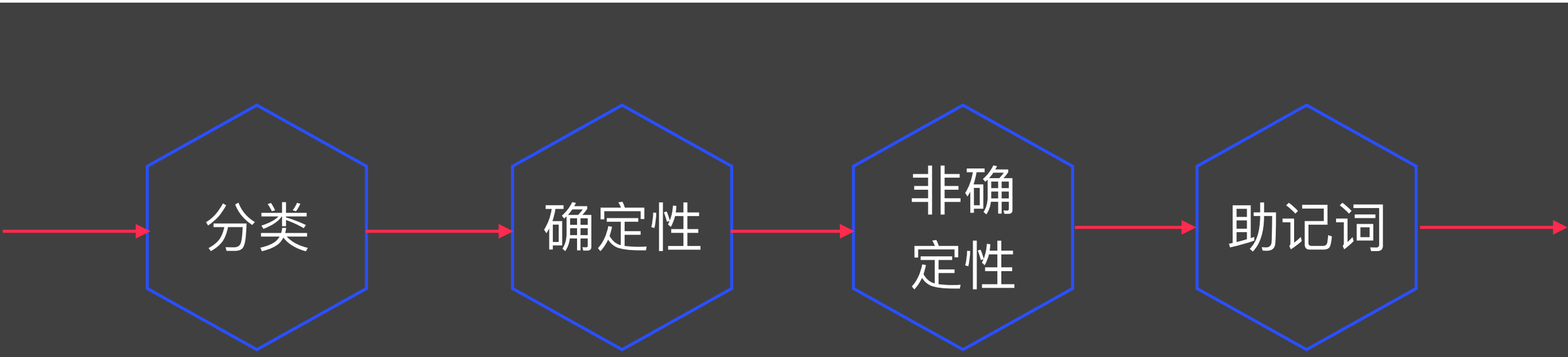
折叠

纸钱包

私钥
密封

多个副本

钱包



CH5

钱包

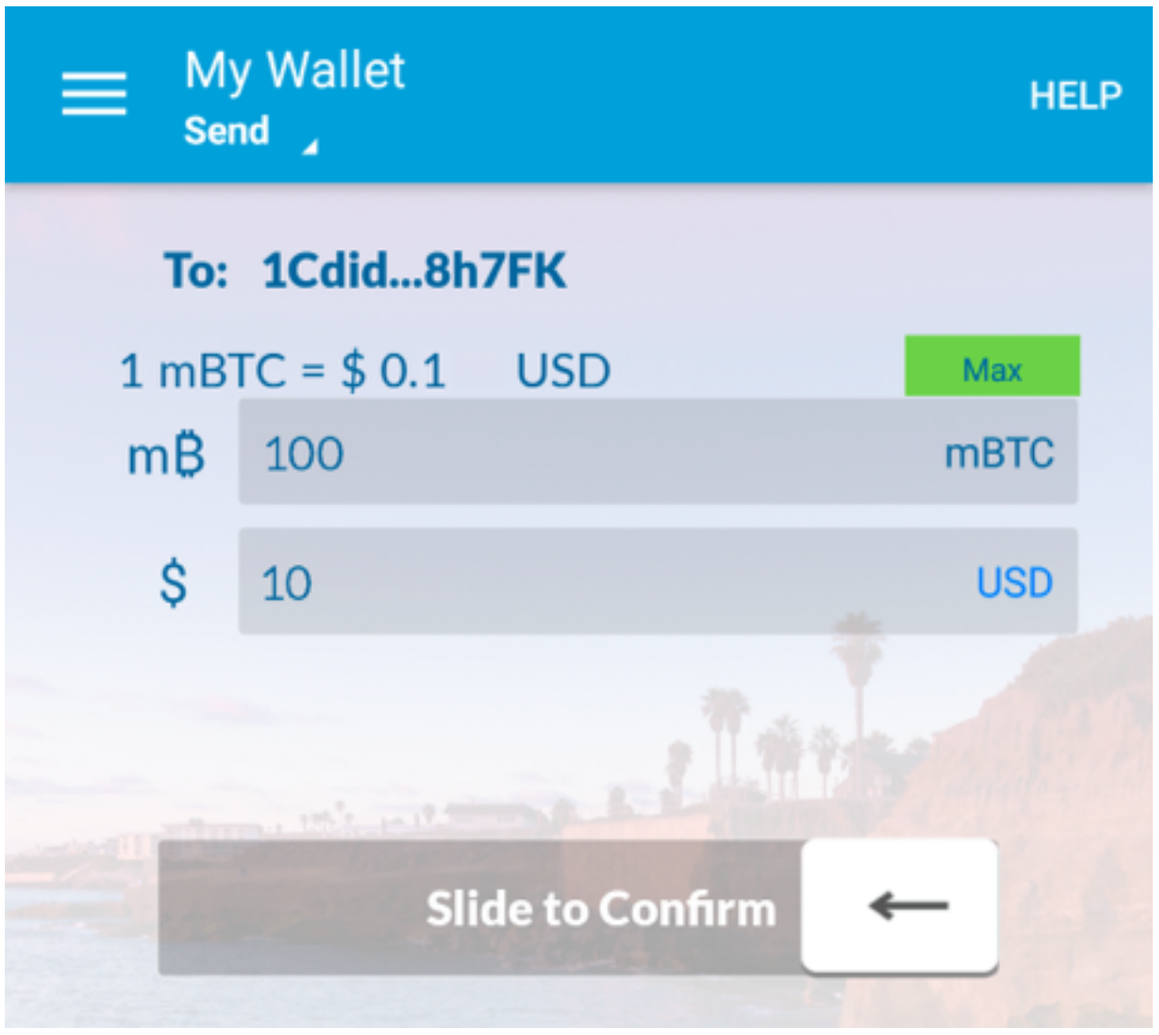
桌面钱包

手机钱包

网络钱包

硬件钱包

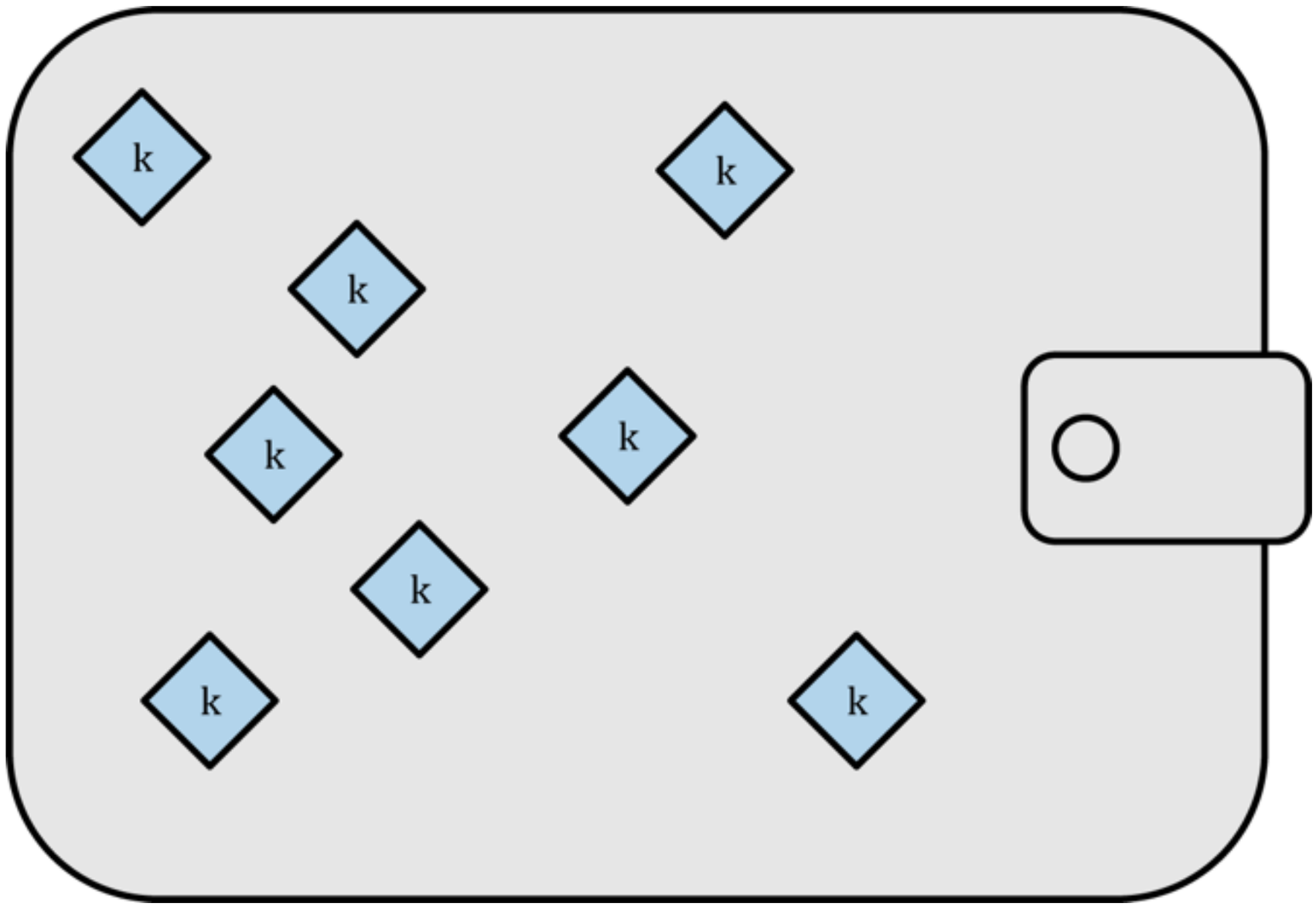
纸钱包



非确定性钱包

随机钱包

JBOK
Just a Bunch of Keys

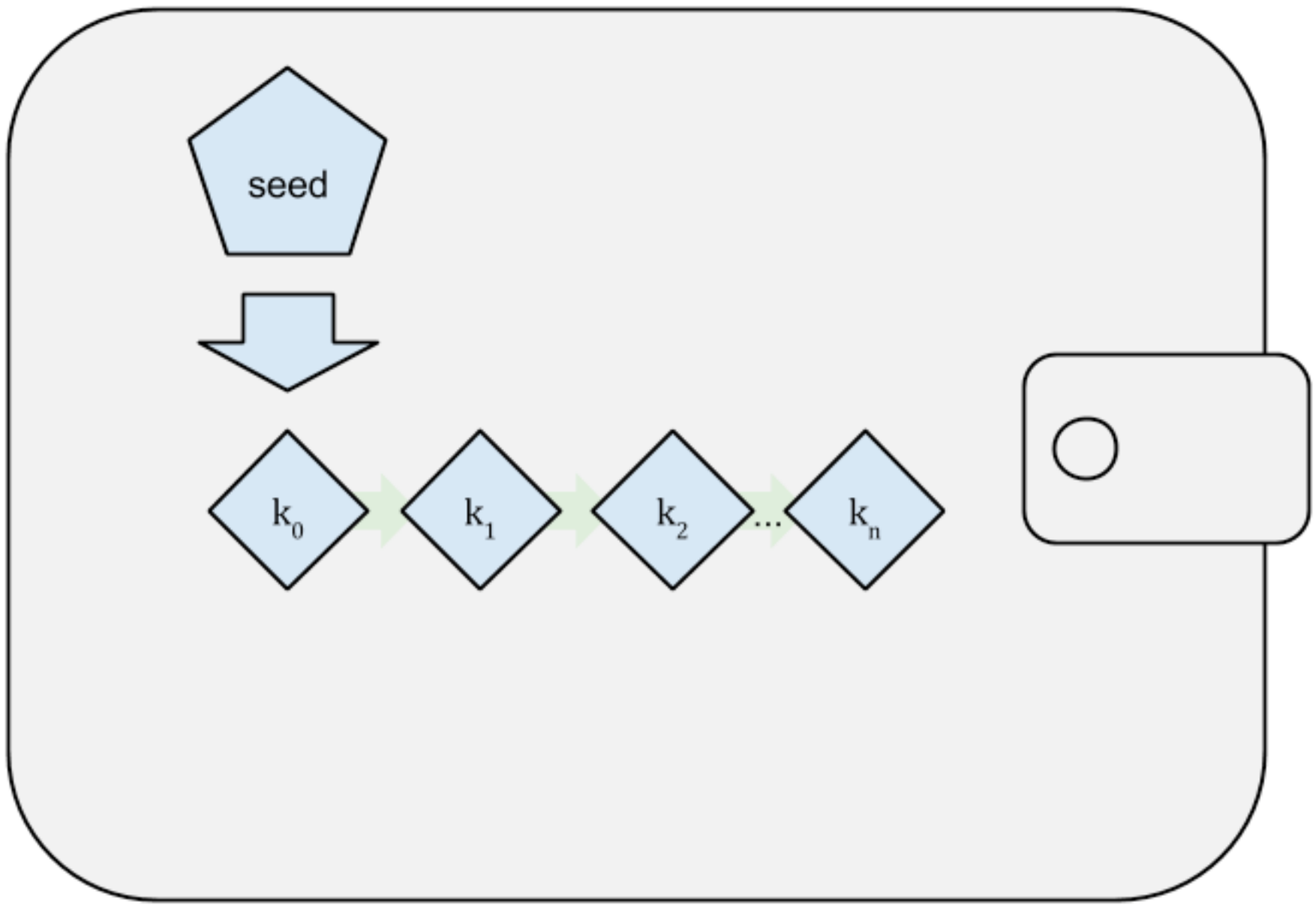


难于管理、备份和导入

确定性钱包

种子钱包

种子
—串随机生
成的数字

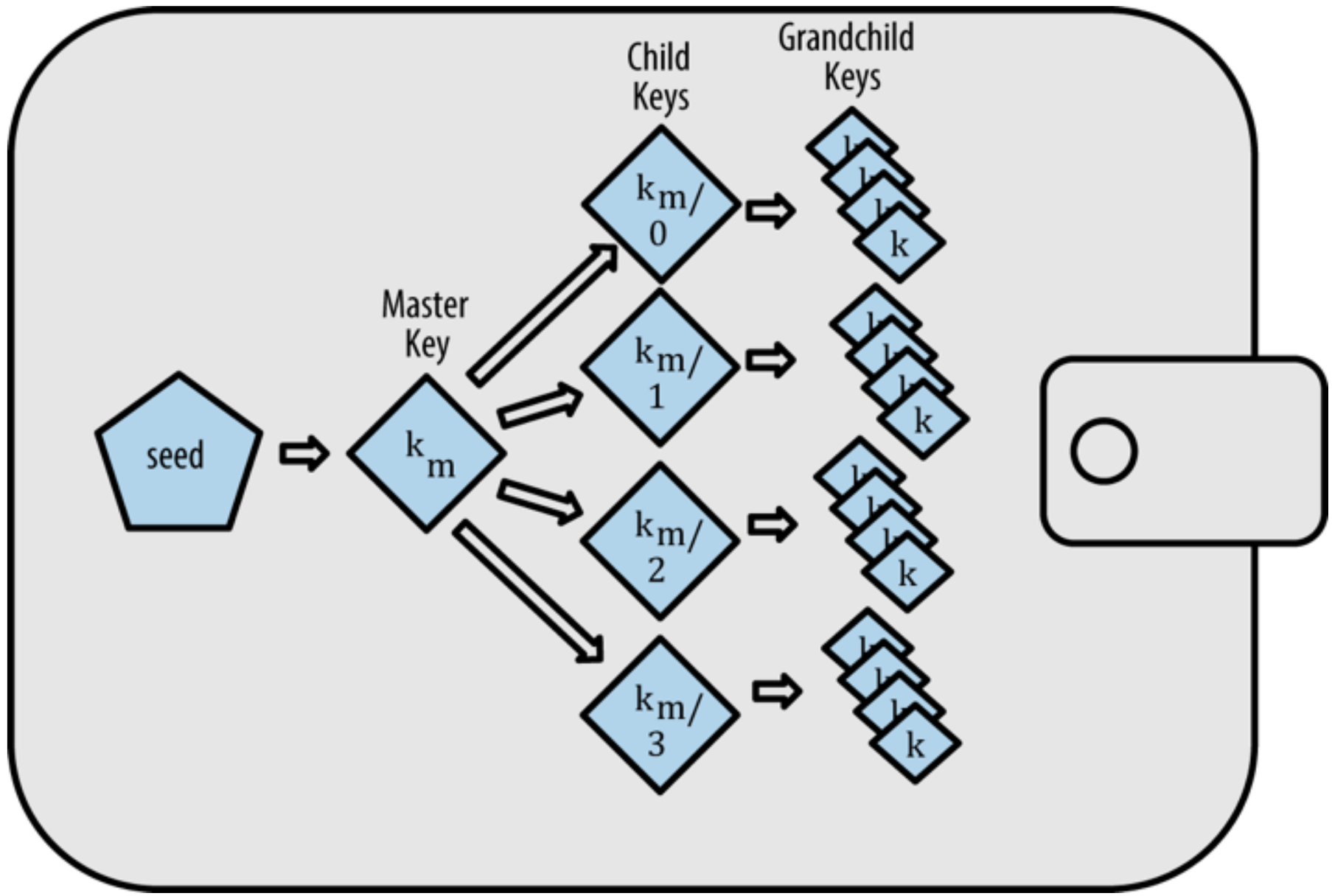


分层确定性钱包

HD钱包

BIP-32
BIP-43
BIP-44

BIP-39

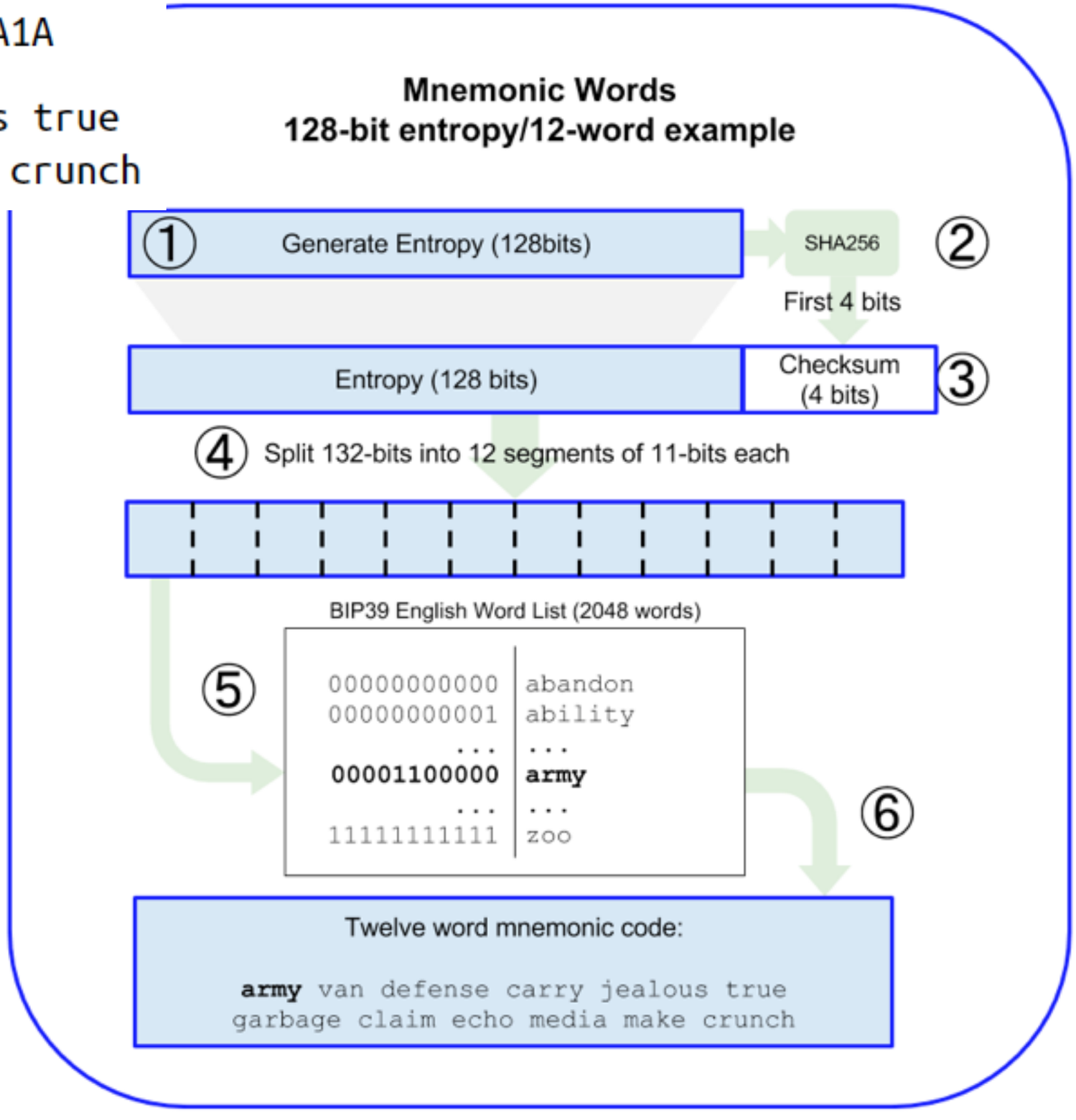


助记词

```
0C1E24E5917779D297E14D45F14E1A1A

army van defense carry jealous true
garbage claim echo media make crunch
```

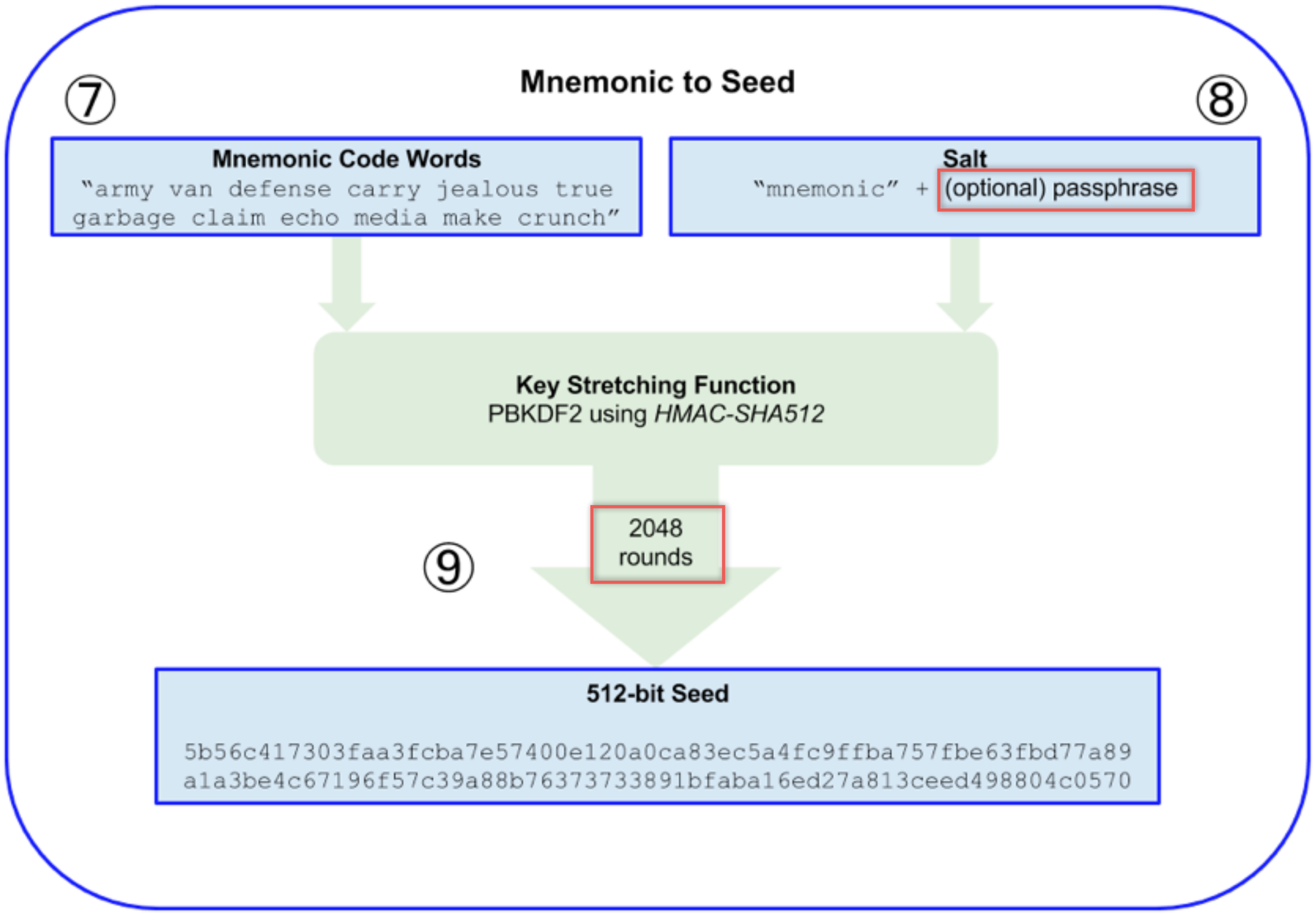
1. *army*
2. *van*
3. *defense*
4. *carry*
5. *jealous*
6. *true*
7. *garbage*
8. *claim*
9. *echo*
10. *media*
11. *make*
12. *crunch*



从助记词产生种子

密码延伸函数

PBKDF2



助记词

Mnemonic

You can enter an existing BIP39 mnemonic, or generate a new random one. Typing your own twelve words will probably not work how you expect, since the words require a particular structure (the last word is a checksum)

For more info see the BIP39 spec

Generate a random 12 word mnemonic, or enter your own below.

BIP39 Mnemonic

army van defense carry jealous true garbage claim echo media make crunch

BIP39 Passphrase (optional)

BIP39 Seed

5b56c417303faa3fcba7e57400e120a0ca83ec5a4fc9ffba757fbe63fbd77a89a1a3be4c6719
6f57c39a88b76373733891bfaba16ed27a813ceed498804c0570

Coin

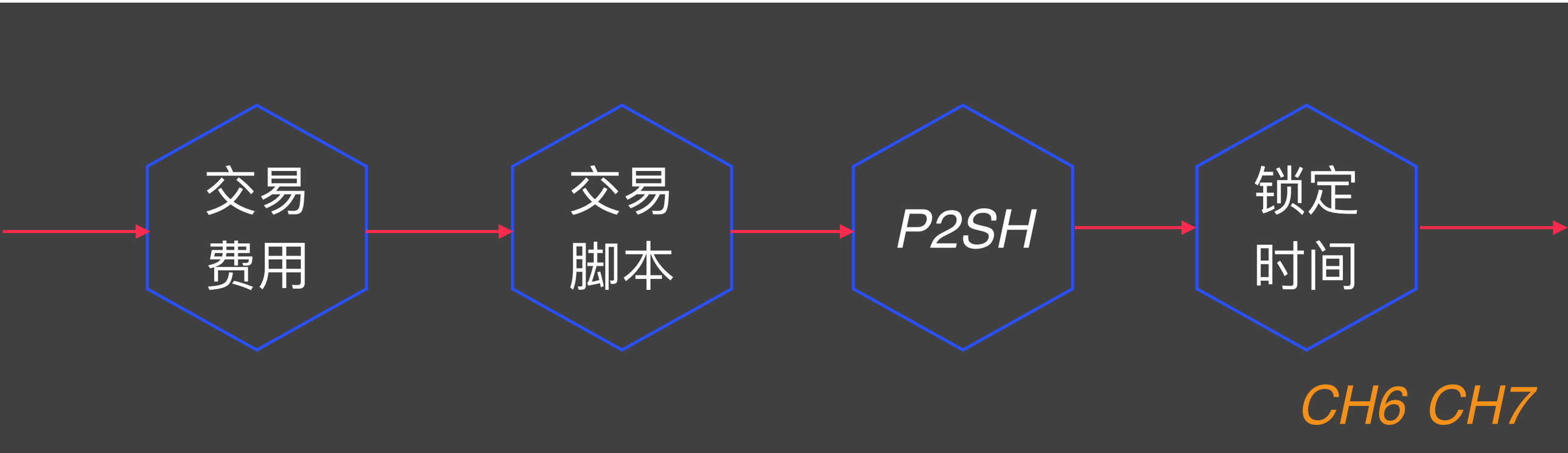
Bitcoin

BIP32 Root Key

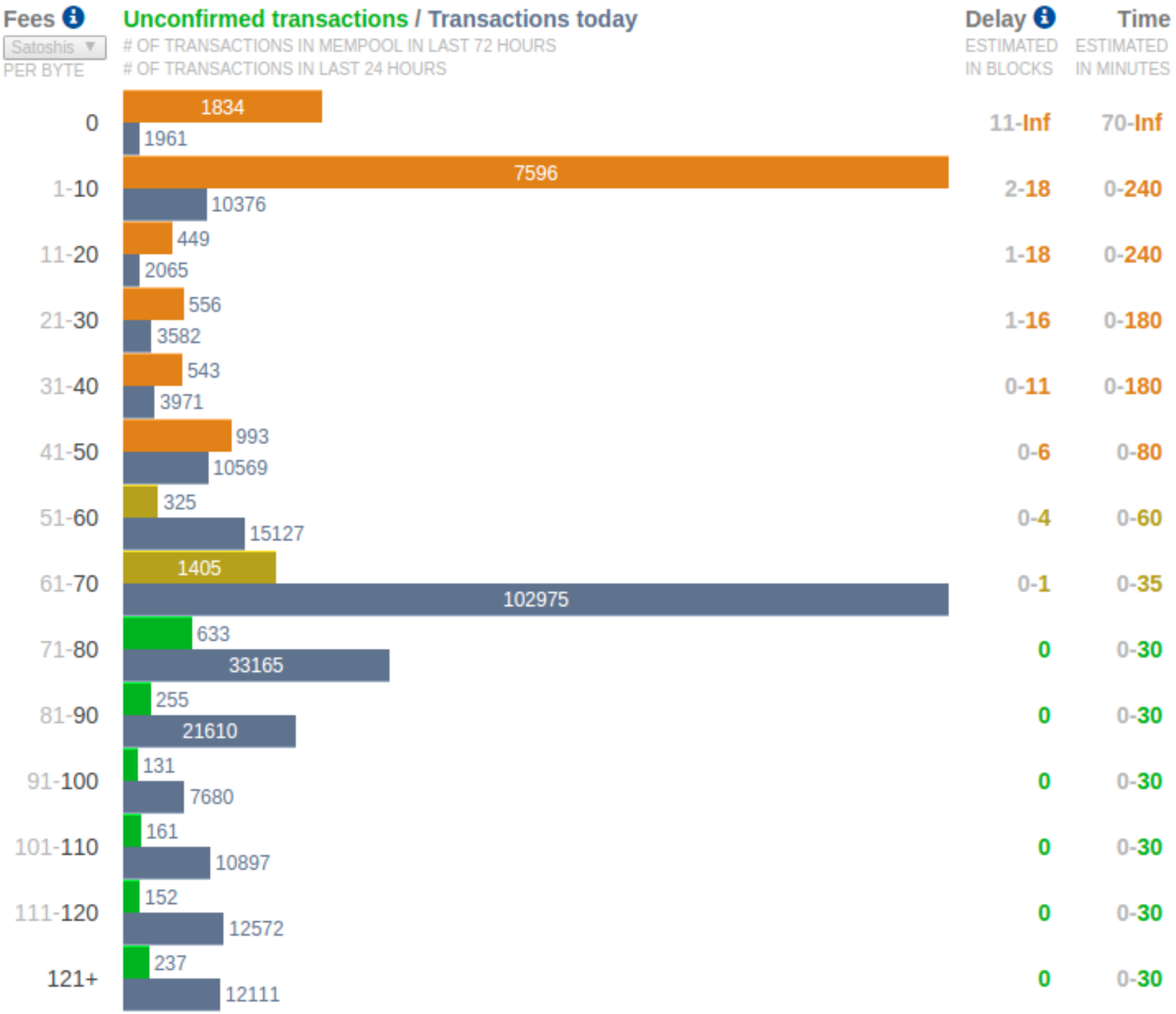
xprv9s21ZrQH143K3t4UZrNgeA3w861fwjYLaGwmPtQyPMmzshV2owVpfBSd2Q7YsHZ9j6
i6ddYjb5PLtUdMZn8LhvuCVhGcQntq5rn7JVMqnie

https://iancoleman.io/bip39/

交易



交易费用


Fees
Satoshis
PER BYTE
Unconfirmed transactions / Transactions today
OF TRANSACTIONS IN MEMPOOL IN LAST 72 HOURS
OF TRANSACTIONS IN LAST 24 HOURS
Delay
ESTIMATED IN BLOCKS
Time
ESTIMATED IN MINUTES
0 1834 1961 11-Inf 70-Inf
1-10 7596 10376 2-18 0-240
11-20 449 2065 1-18 0-240
21-30 556 3582 1-16 0-180
31-40 543 3971 0-11 0-180
41-50 993 10569 0-6 0-80
51-60 325 15127 0-4 0-60
61-70 1405 102975 0-1 0-35
71-80 633 33165 0 0-30
81-90 255 21610 0 0-30
91-100 131 7680 0 0-30
101-110 161 10897 0 0-30
111-120 152 12572 0 0-30
121+ 237 12111 0 0-30

交易脚本

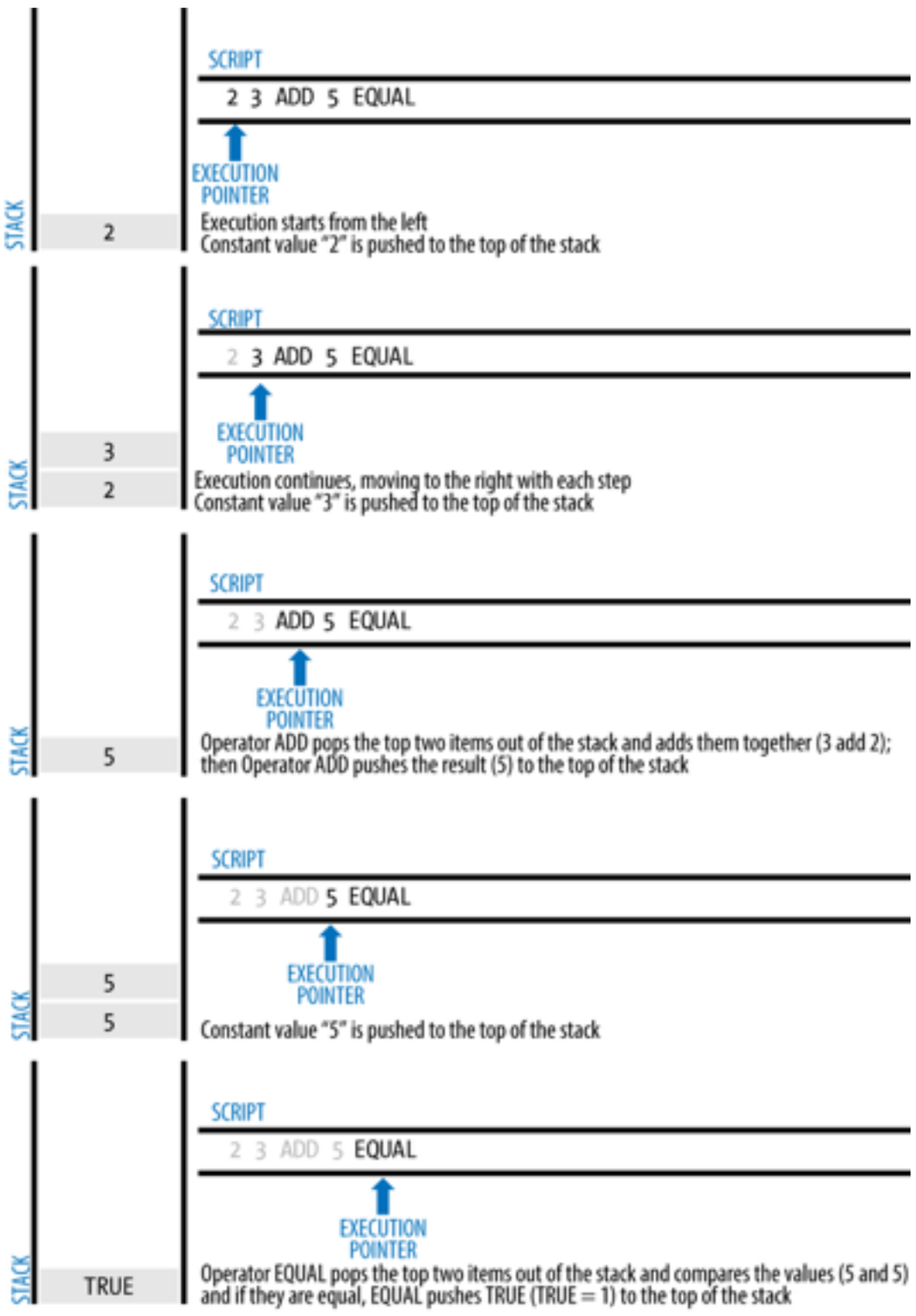
Unlocking Script (scriptSig) + Locking Script (scriptPubKey)

```
<sig> <PubK>   DUP HASH160 <PubKHash> EQUALVERIFY CHECKSIG
```

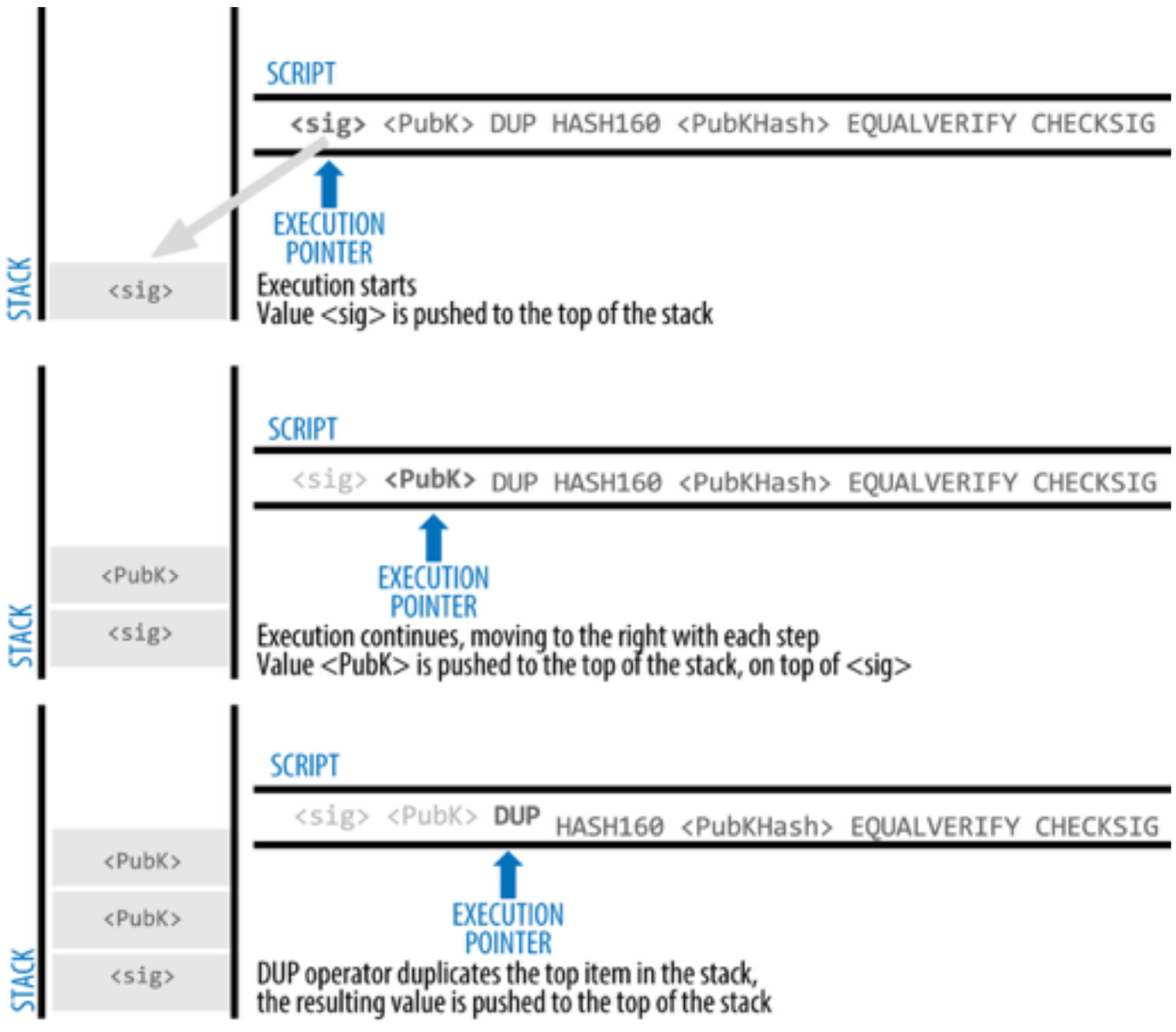
Unlock Script (scriptSig) is provided by the user to resolve the encumbrance

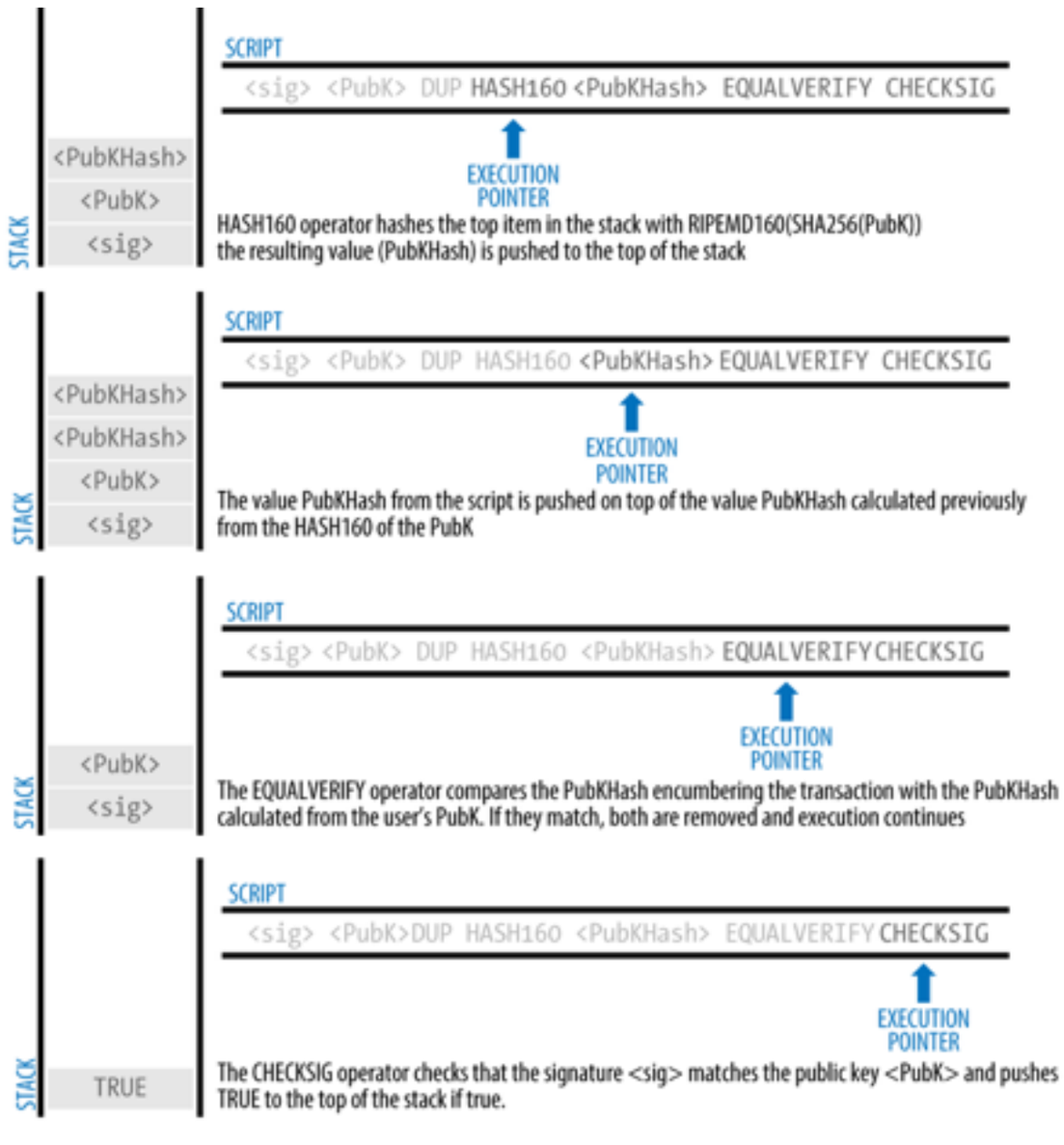
Lock Script (scriptPubKey) is found in a transaction output and is the encumbrance that must be fulfilled to spend the output

交易脚本


SCRIPT
2 3 ADD 5 EQUAL
EXECUTION POINTER
STACK
2
Execution starts from the left
Constant value "2" is pushed to the top of the stack
SCRIPT
2 3 ADD 5 EQUAL
EXECUTION POINTER
STACK
3
2
Execution continues, moving to the right with each step
Constant value "3" is pushed to the top of the stack
SCRIPT
2 3 ADD 5 EQUAL
EXECUTION POINTER
STACK
5
Operator ADD pops the top two items out of the stack and adds them together (3 add 2); then Operator ADD pushes the result (5) to the top of the stack
SCRIPT
2 3 ADD 5 EQUAL
EXECUTION POINTER
STACK
5
5
Constant value "5" is pushed to the top of the stack
SCRIPT
2 3 ADD 5 EQUAL
EXECUTION POINTER
STACK
TRUE
Operator EQUAL pops the top two items out of the stack and compares the values (5 and 5) and if they are equal, EQUAL pushes TRUE (TRUE = 1) to the top of the stack

交易脚本

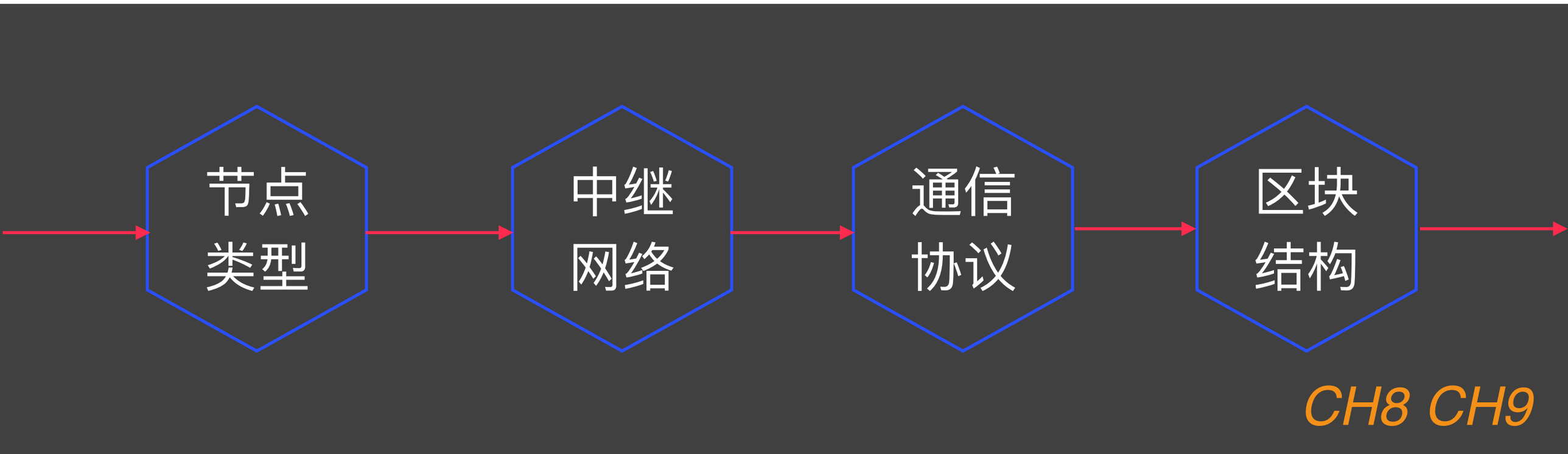

SCRIPT
<sig> <PubK> DUP HASH160 <PubKHash> EQUALVERIFY CHECKSIG
EXECUTION POINTER
STACK
<sig>
Execution starts
Value <sig> is pushed to the top of the stack
SCRIPT
<sig> <PubK> DUP HASH160 <PubKHash> EQUALVERIFY CHECKSIG
EXECUTION POINTER
STACK
<PubK>
<sig>
Execution continues, moving to the right with each step
Value <PubK> is pushed to the top of the stack, on top of <sig>
SCRIPT
<sig> <PubK> DUP HASH160 <PubKHash> EQUALVERIFY CHECKSIG
EXECUTION POINTER
STACK
<PubK>
<PubK>
<sig>
DUP operator duplicates the top item in the stack,
the resulting value is pushed to the top of the stack


SCRIPT
<sig> <PubK> DUP HASH160 <PubKHash> EQUALVERIFY CHECKSIG
EXECUTION POINTER
STACK
<PubKHash>
<PubK>
<sig>
HASH160 operator hashes the top item in the stack with RIPEMD160(SHA256(PubK))
the resulting value (PubKHash) is pushed to the top of the stack
SCRIPT
<sig> <PubK> DUP HASH160 <PubKHash> EQUALVERIFY CHECKSIG
EXECUTION POINTER
STACK
<PubKHash>
<PubKHash>
<PubK>
<sig>
The value PubKHash from the script is pushed on top of the value PubKHash calculated previously
from the HASH160 of the PubK
SCRIPT
<sig> <PubK> DUP HASH160 <PubKHash> EQUALVERIFY CHECKSIG
EXECUTION POINTER
STACK
<PubK>
<sig>
The EQUALVERIFY operator compares the PubKHash encumbering the transaction with the PubKHash
calculated from the user's PubK. If they match, both are removed and execution continues
SCRIPT
<sig> <PubK> DUP HASH160 <PubKHash> EQUALVERIFY CHECKSIG
EXECUTION POINTER
STACK
TRUE
The CHECKSIG operator checks that the signature <sig> matches the public key <PubK> and pushes
TRUE to the top of the stack if true.

多重签名

时间锁

网络、区块链



节点类型

Reference Client (Bitcoin Core)

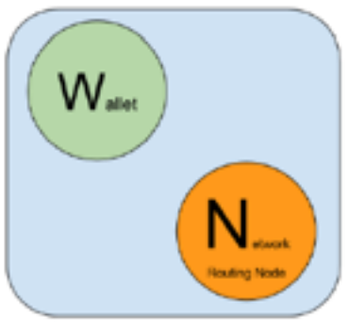
Contains a Wallet, Miner, full Blockchain database, and Network routing node on the bitcoin P2P network.

Full Block Chain Node

Contains a full Blockchain database, and Network routing node on the bitcoin P2P network.

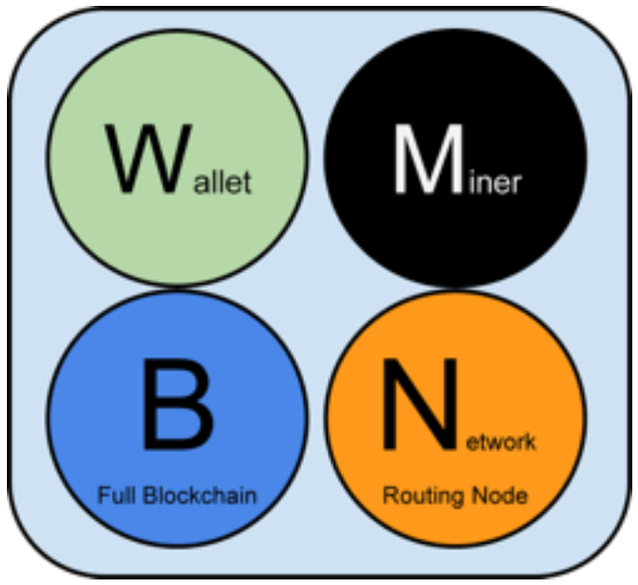
Solo Miner

Contains a mining function with a full copy of the blockchain and a bitcoin P2P network routing node.



Lightweight (SPV) wallet

Contains a Wallet and a Network node on the bitcoin P2P protocol, without a blockchain.

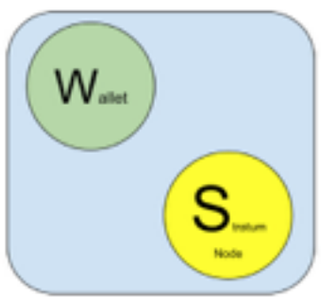


Pool Protocol Servers

Gateway routers connecting the bitcoin P2P network to nodes running other protocols such as pool mining nodes or Stratum nodes.

Mining Nodes

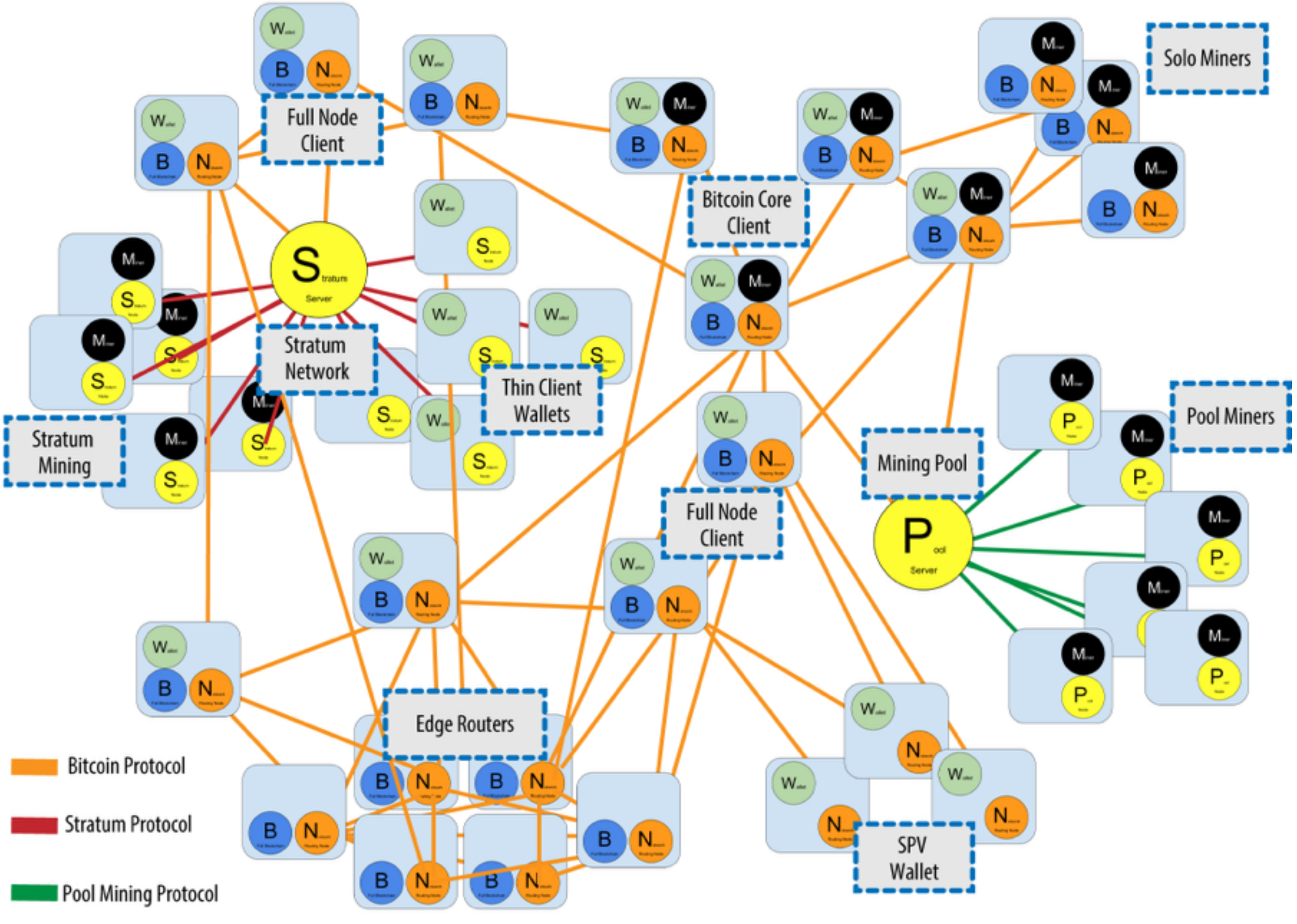
Contain a mining function, without a blockchain, with the Stratum protocol node (S) or other pool (P) mining protocol node.



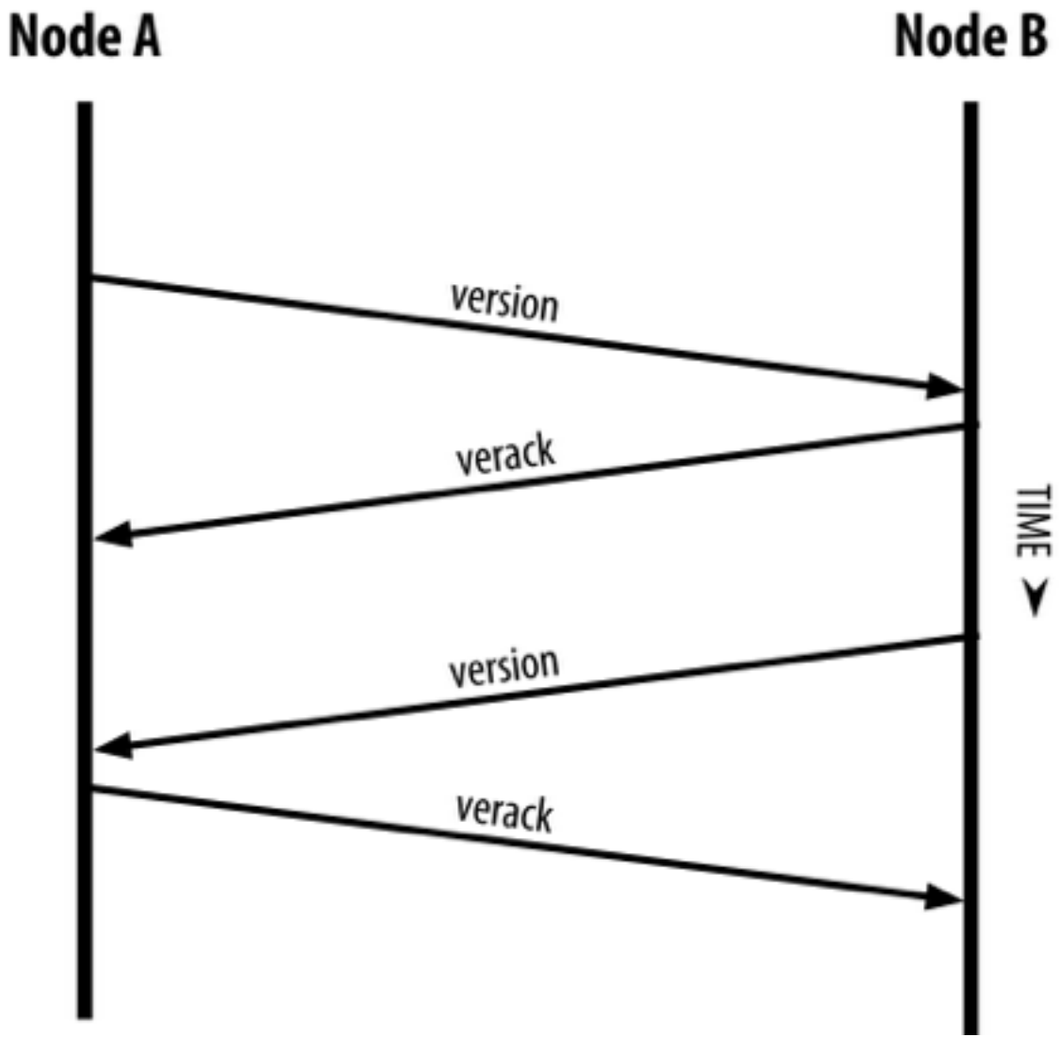
Lightweight (SPV) Stratum wallet

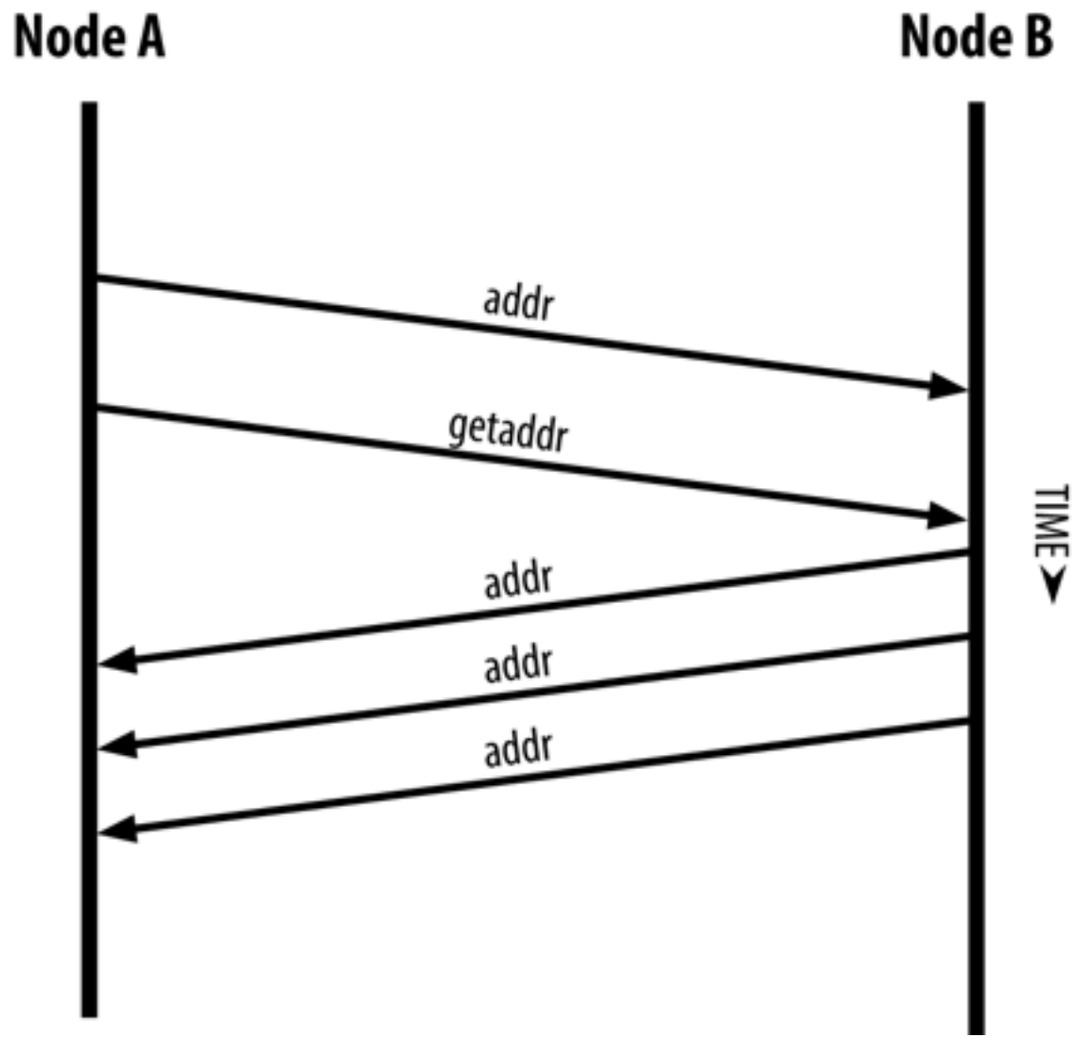
Contains a Wallet and a Network node on the Stratum protocol, without a blockchain.

中继网络

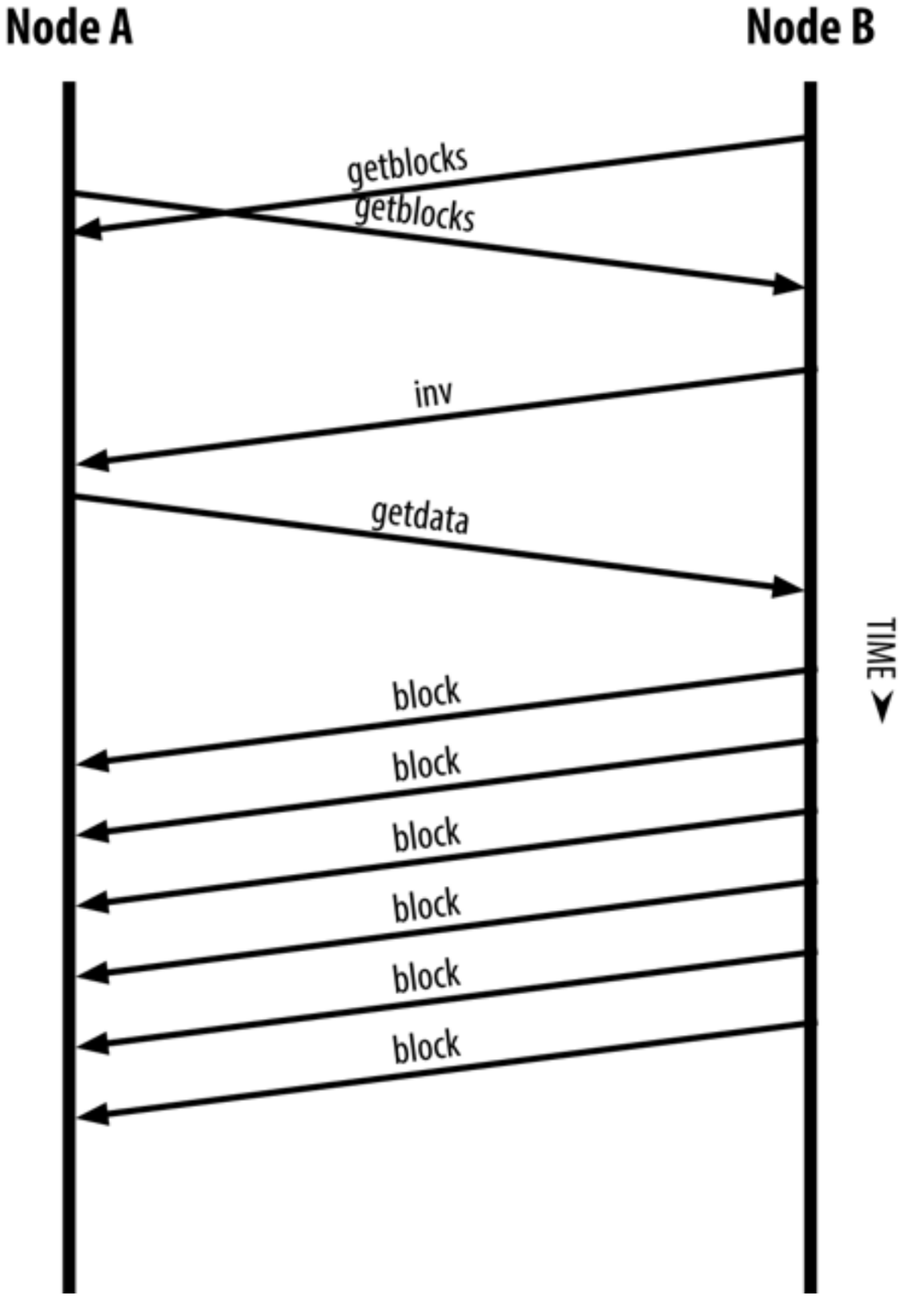

Full Node Client
Solo Miners
Bitcoin Core Client
Stratum Server
Stratum Network
Thin Client Wallets
Stratum Mining
Mining Pool
Pool Miners
Pool Server
Full Node Client
Edge Routers
SPV Wallet
Bitcoin Protocol
Stratum Protocol
Pool Mining Protocol

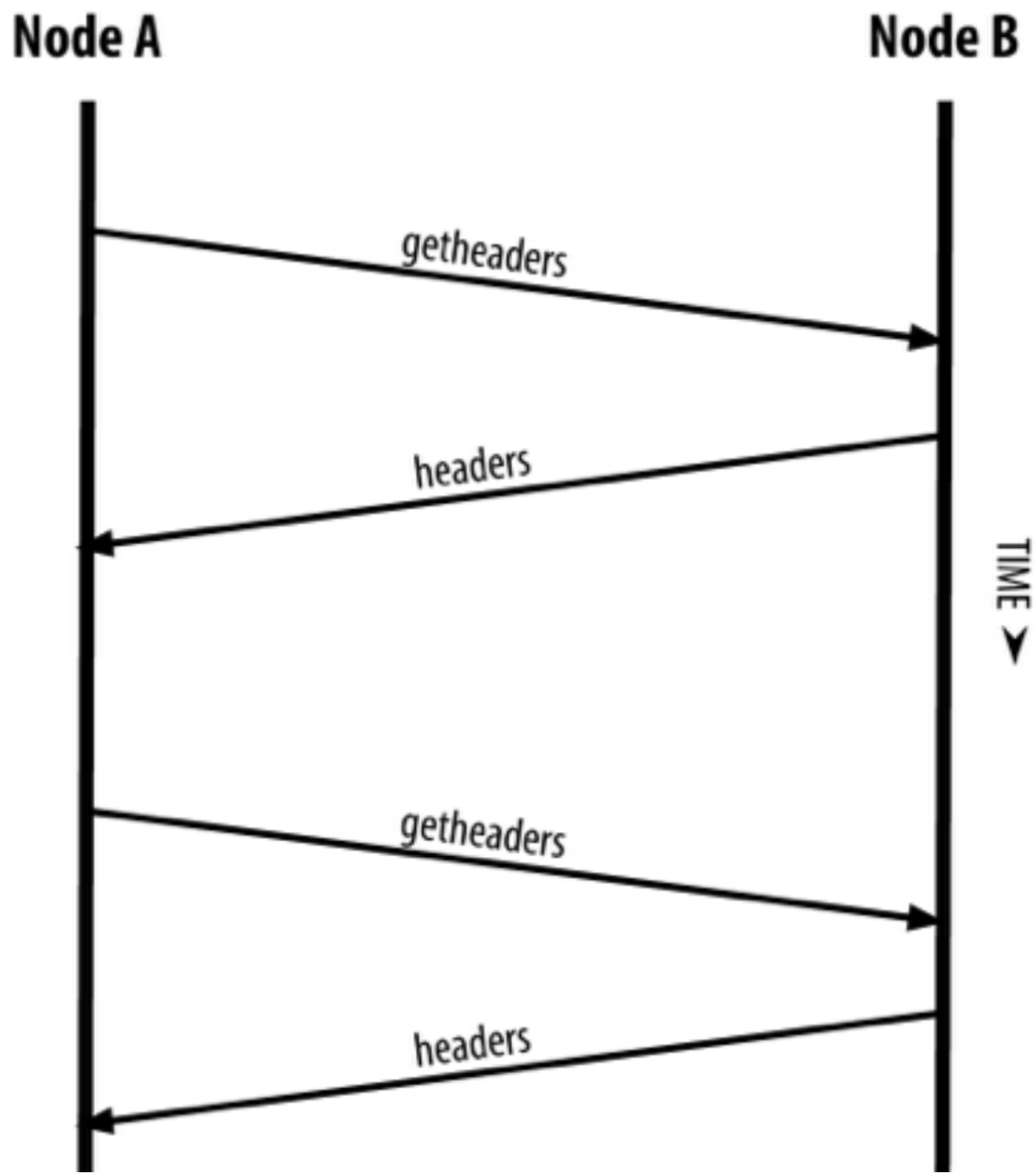
协议


Node A
Node B
version
verack
version
verack
TIME

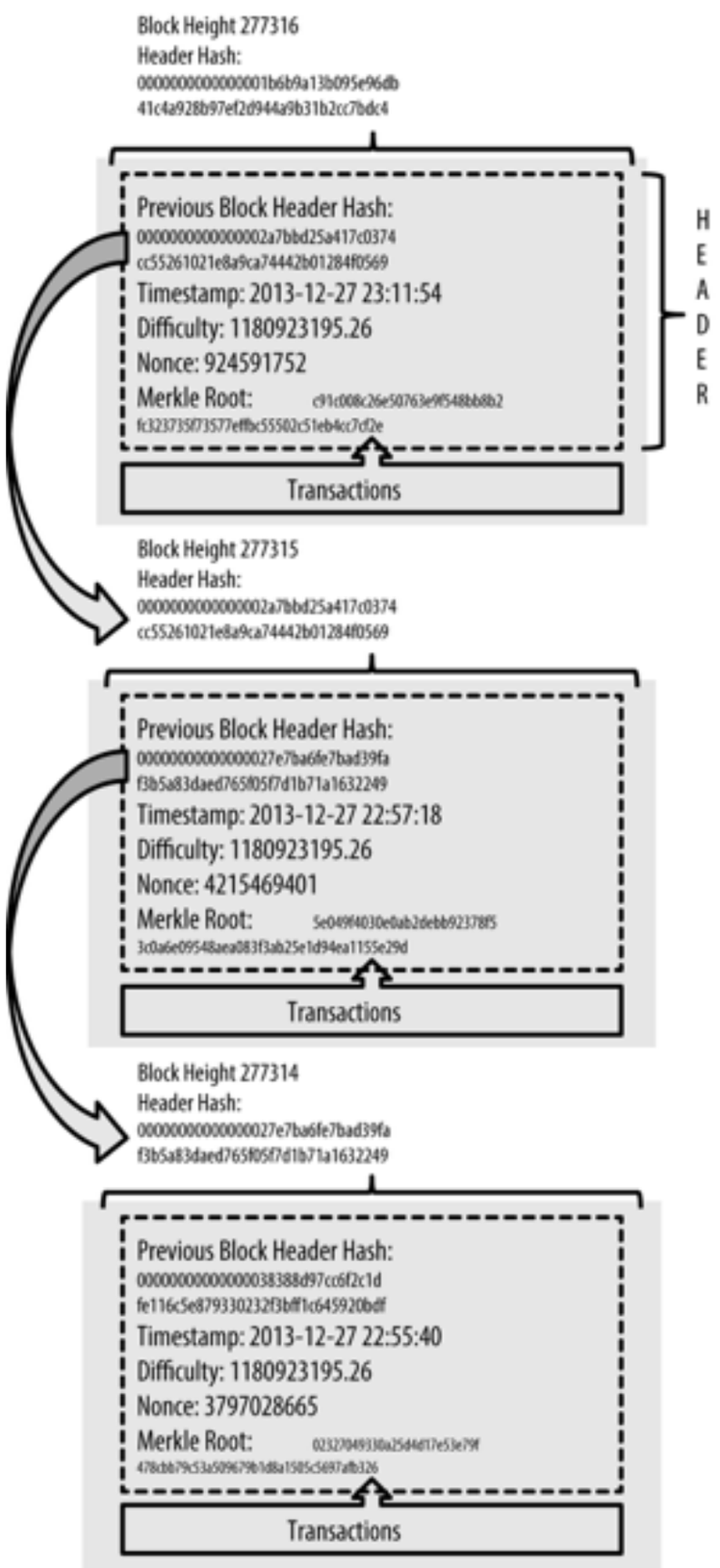

Node A
Node B
addr
getaddr
addr
addr
addr
TIME

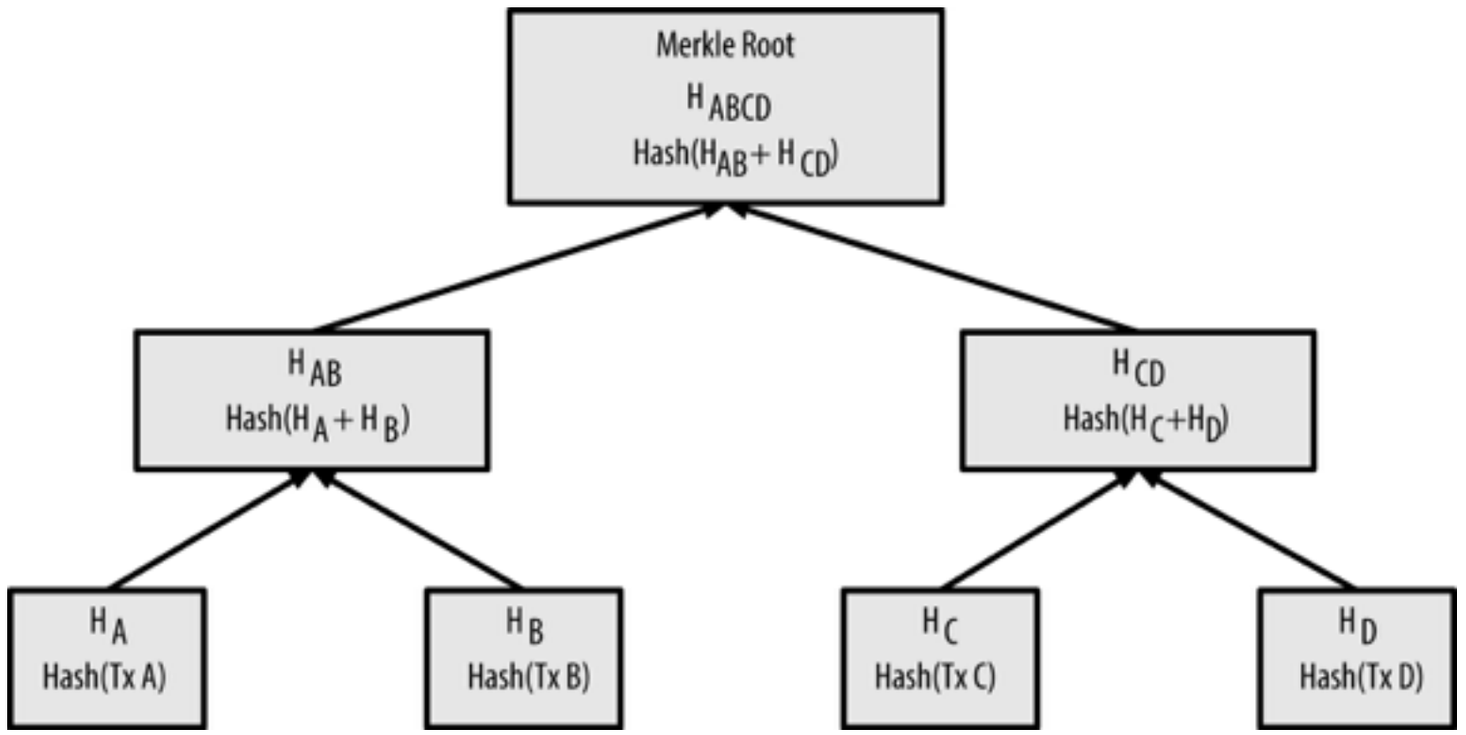
协议

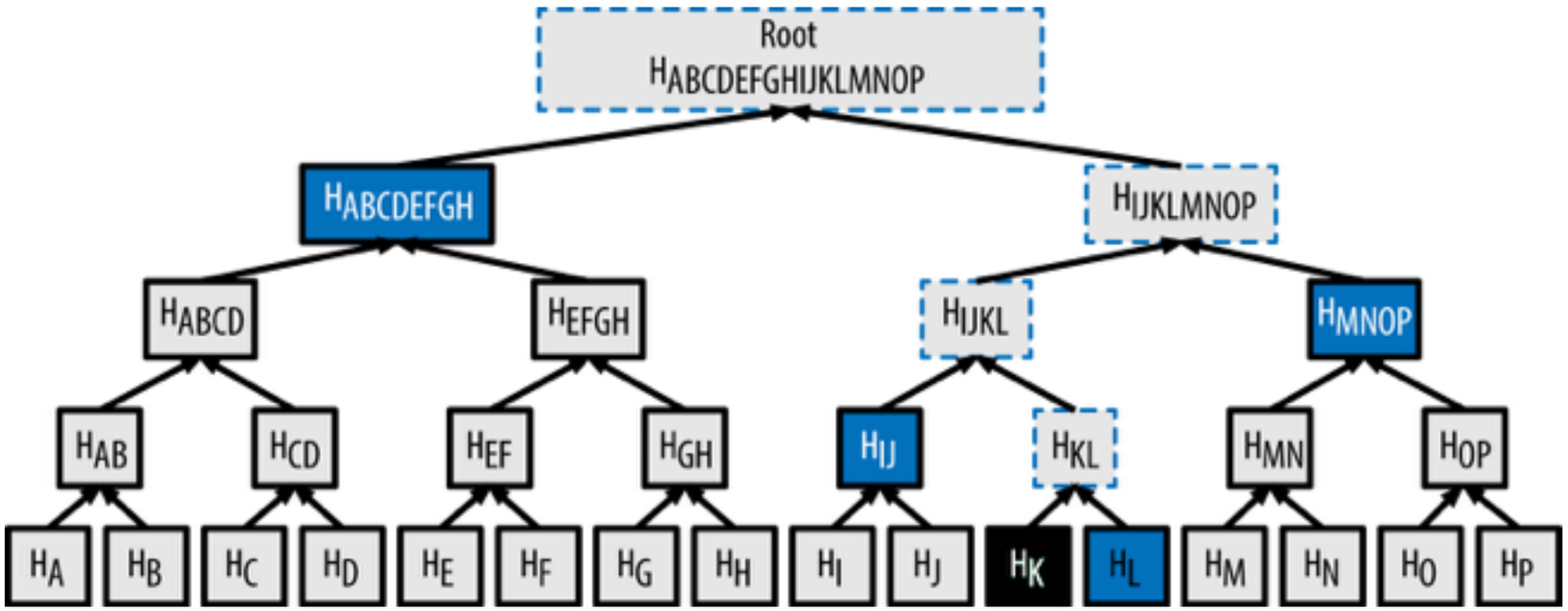

Node A
Node B
getblocks
getblocks
inv
getdata
block
block
block
block
block
block
TIME


Node A
Node B
getheaders
headers
getheaders
headers
TIME

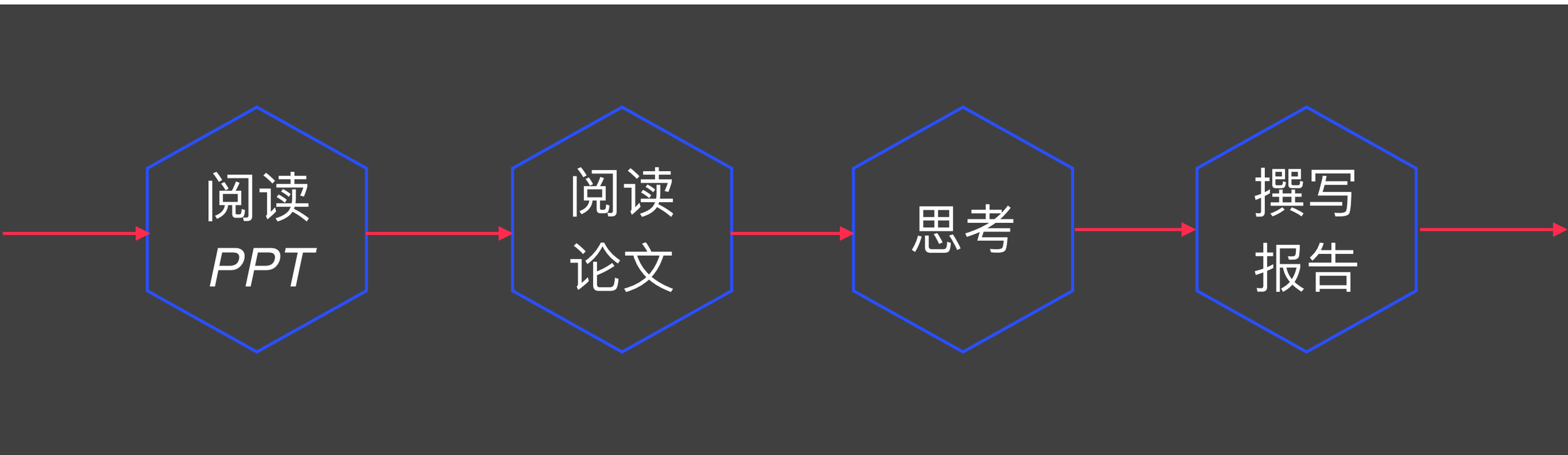
梅克尔树


Block Height 277316
Header Hash:
000000000000001b6b9a13b095e96db
41c4a928b97ef2d944a9b31b2cc7bdc4
Previous Block Header Hash:
000000000000002a7bbd25a417c0374
cc55261021e8a9ca74442b01284f0569
Timestamp: 2013-12-27 23:11:54
Difficulty: 1180923195.26
Nonce: 924591752
Merkle Root: c91c008c26e50763e9f548bb8b2
fc323735f73577effbc55502c51eb4cc7d2e
Transactions
HEADER
Block Height 277315
Header Hash:
000000000000002a7bbd25a417c0374
cc55261021e8a9ca74442b01284f0569
Previous Block Header Hash:
0000000000000027e7ba6fe7bad39fa
f3b5a83daed765f05f7d1b71a1632249
Timestamp: 2013-12-27 22:57:18
Difficulty: 1180923195.26
Nonce: 4215469401
Merkle Root: 5e049f4030e0ab2debb92378f5
3c0a6e09548aea083f3ab25e1d94ea1155e29d
Transactions
Block Height 277314
Header Hash:
0000000000000027e7ba6fe7bad39fa
f3b5a83daed765f05f7d1b71a1632249
Previous Block Header Hash:
00000000000000038388d97cc6f2c1d
fe116c5e879330232f3bff1c645920bdf
Timestamp: 2013-12-27 22:55:40
Difficulty: 1180923195.26
Nonce: 3797028665
Merkle Root: 023270493 30a25d4d17e53e79f
478cbb79c53a5096 79b1d8a1505c5697afb326
Transactions


Merkle Root
H_{ABCD}
$Hash(H_{AB}+H_{CD})$
H_{AB}
$Hash(H_A+H_B)$
H_{CD}
$Hash(H_C+H_D)$
H_A
$Hash(Tx\ A)$
H_B
$Hash(Tx\ B)$
H_C
$Hash(Tx\ C)$
H_D
$Hash(Tx\ D)$


Root
$H_{ABCDEFGHIJKLMNOP}$
$H_{ABCDEFGH}$
$H_{IJKLMNOP}$
H_{ABCD}
H_{EFGH}
H_{IJKL}
H_{MNOP}
H_{AB}
H_{CD}
H_{EF}
H_{GH}
H_{IJ}
H_{KL}
H_{MN}
H_{OP}
H_A
H_B
H_C
H_D
H_E
H_F
H_G
H_H
H_I
H_J
H_K
H_L
H_M
H_N
H_O
H_P

课后作业



Homework

课后作业

基于Bitcoin开发一个应用

1、基于Bitcoin开源库

2、实现一个常见的应用

3、可以运行，可以演示

4、可以改造Bitcoin

12月2日晚上12点前提交给助教

谢谢！

Huiping Sun

sunhp@ss.pku.edu.cn

https://huipingsun.github.io