

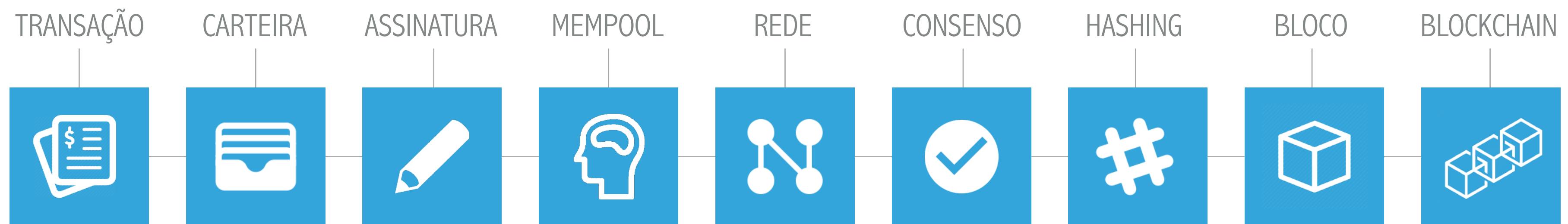
The background of the slide is a dark blue field filled with a complex network of glowing blue lines and dots, resembling a circuit board or a data network. Overlaid on this are several rectangular blocks, each representing a block in a blockchain. Each block has a yellow header with a Bitcoin logo and a hexadecimal hash. The main body of each block is filled with a dense grid of white and blue binary code (0s and 1s). Dashed white lines with small circles at the ends connect the blocks in a chain-like structure, illustrating the sequential and linked nature of a blockchain. The blocks are arranged in a non-linear fashion, with some overlapping and others connected by curved lines.

IMD0293

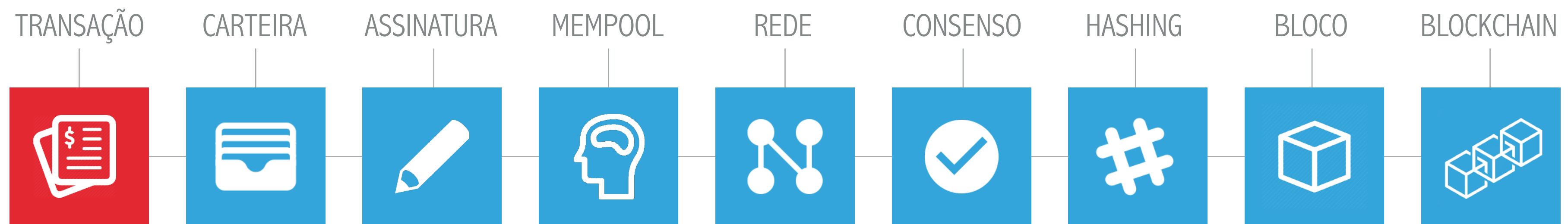
ARQUITETURA DE UM BLOCKCHAIN

SCRIPT BITCOIN

ARQUITETURA DE UM **BLOCKCHAIN**



ARQUITETURA DE UM **BLOCKCHAIN**



SCRIPTS BITCOIN

- ▶ Endereços de saída na verdade são *scripts*

```
"scriptPubKey": "OP_DUP OP_HASH160 7f9b1a7fb68d60c536c2fd8aeaa53a8f3cc025a8 OP_EQUALVERIFY OP_CHECKSIG"
```

- ▶ *Script* ou *Bitcoin Scripting Language*
 - ▶ Baseada em pilha (*push* e *pop*)
 - ▶ Script é avaliado da esquerda para direita
 - ▶ Dois tipos de informação: dados e **OPCODEs**
 - ▶ **Simples**, não é Turing-completo (sem *loops*)

Script

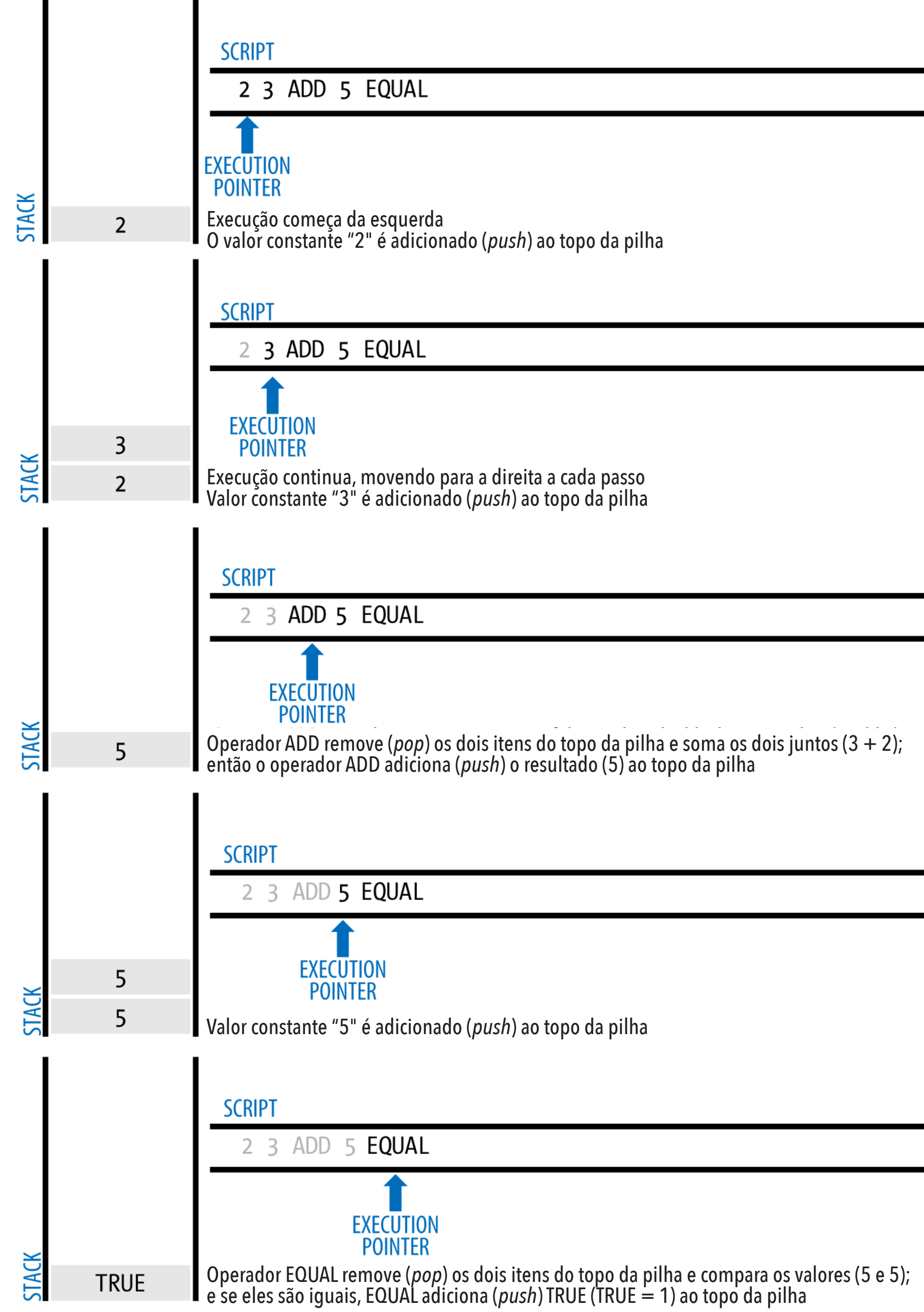
Lista de instruções armazenada em cada transação que quando executada determina se uma transação é válida e os bitcoins podem ser gastos.

Script

○ nome da linguagem de *scripting* do Bitcoin.

SCRIPTS BITCOIN

- ▶ Dados sempre são inseridos (*push*) na pilha
- ▶ OPCODEs podem retirar (*pop*) elementos da pilha, fazer algo com eles, e opcionalmente inserir (*push*) novos elementos na pilha



SCRIPTS BITCOIN

2 7 OP_ADD 3 OP_SUB 1 OP_ADD 7 OP_EQUAL

POR QUE USAMOS SCRIPTS?

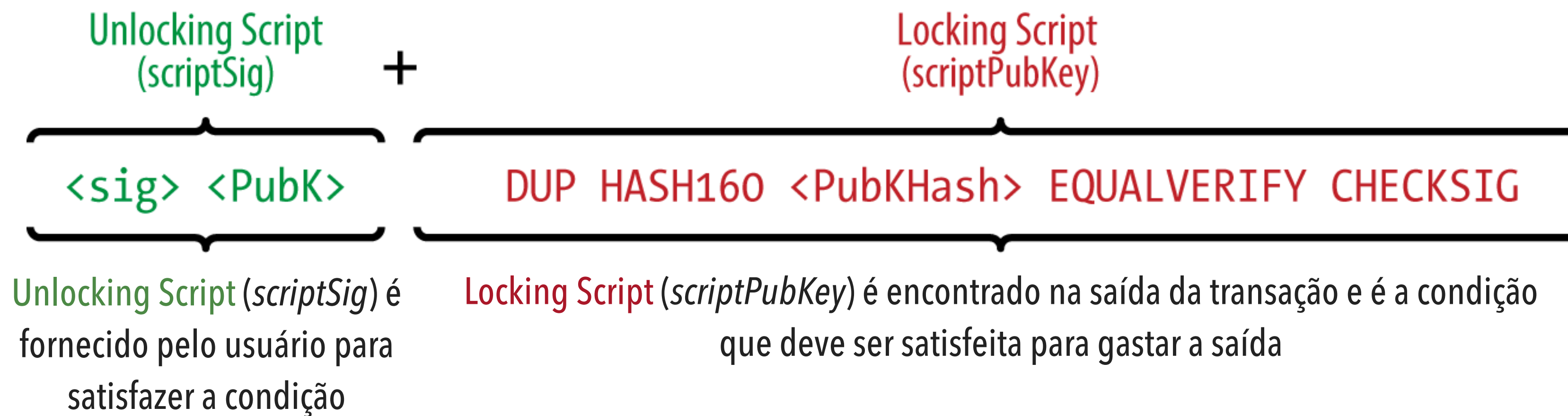
Pergunta: Por que não fazemos uma comparação simples entre chave pública e assinatura ao invés de usar OPCODEs e operações em pilha?!?!?

Resposta: Porque com Script e OPCODEs podemos criar diferentes tipos de “problemas” para “destravar” os bitcoins

SCRIPTS BITCOIN

A maioria das transações processadas pela rede Bitcoin tem o formato “*Pagamento para o endereço bitcoin de Bob*”, baseados no script **P2PKH** (*pay-to-public-key-hash*)

SCRIPTS BITCOIN



SCRIPTS BITCOIN: EXEMPLO

- ▶ Usando um exemplo aritmético como *locking script*:

```
3 OP_ADD 5 OP_EQUAL
```

- ▶ Que pode ser satisfeita por uma transação contendo uma entrada com o seguinte *unlocking script*:

```
2
```

- ▶ O *software* que for validar combina os *scripts*:

```
2 3 OP_ADD 5 OP_EQUAL
```

resultado: OP_TRUE



TRANSAÇÃO VÁLIDA!

SCRIPTS BITCOIN: P2PKH

- ▶ *Locking script (scriptPubKey)*: encontrado na saída da transação anterior, especifica os requisitos para resgatar a transação
- ▶ *Unlocking script (scriptSig)*: encontrado na entrada da transação, resgata a saída da transação anterior
- ▶ Um nó bitcoin vai validar a transação executando os *scripts* de *unlocking* e *locking* sequencialmente

Como exemplo, a transação de Alice para Bob:

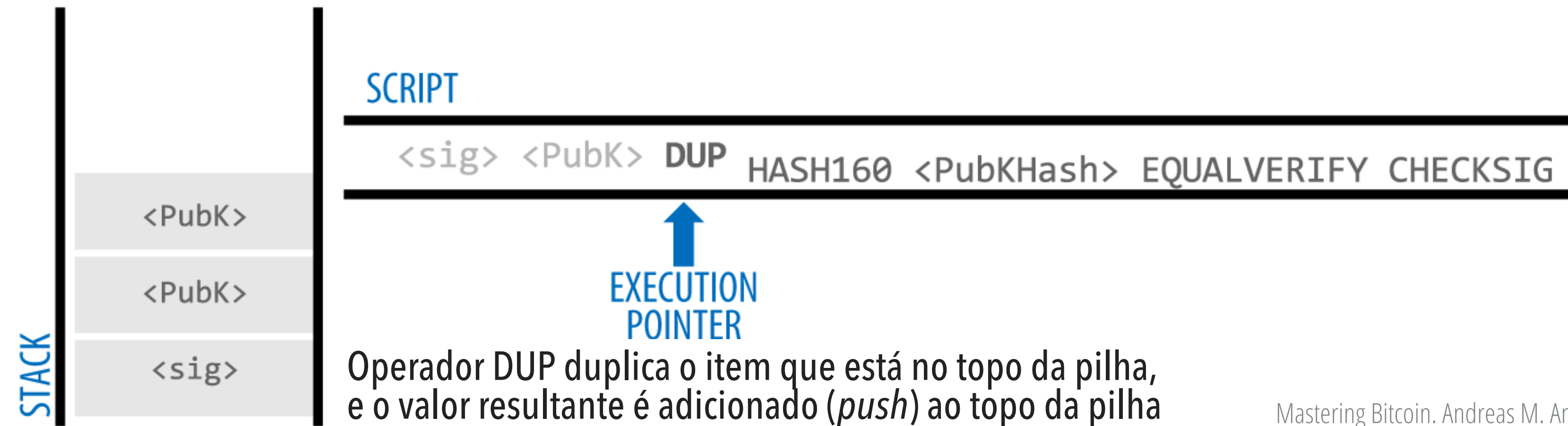
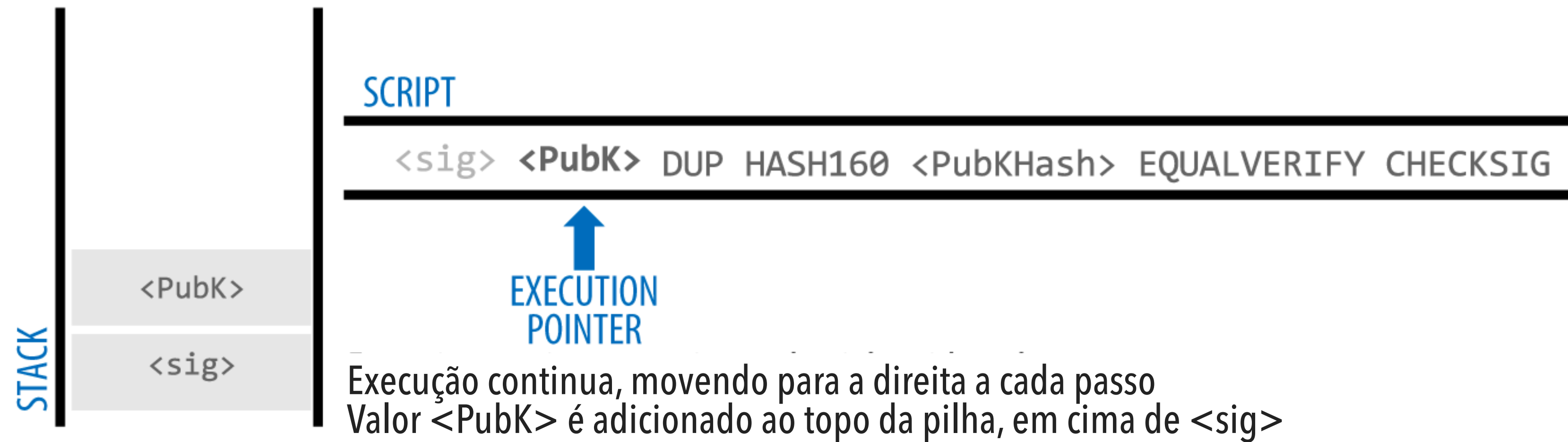
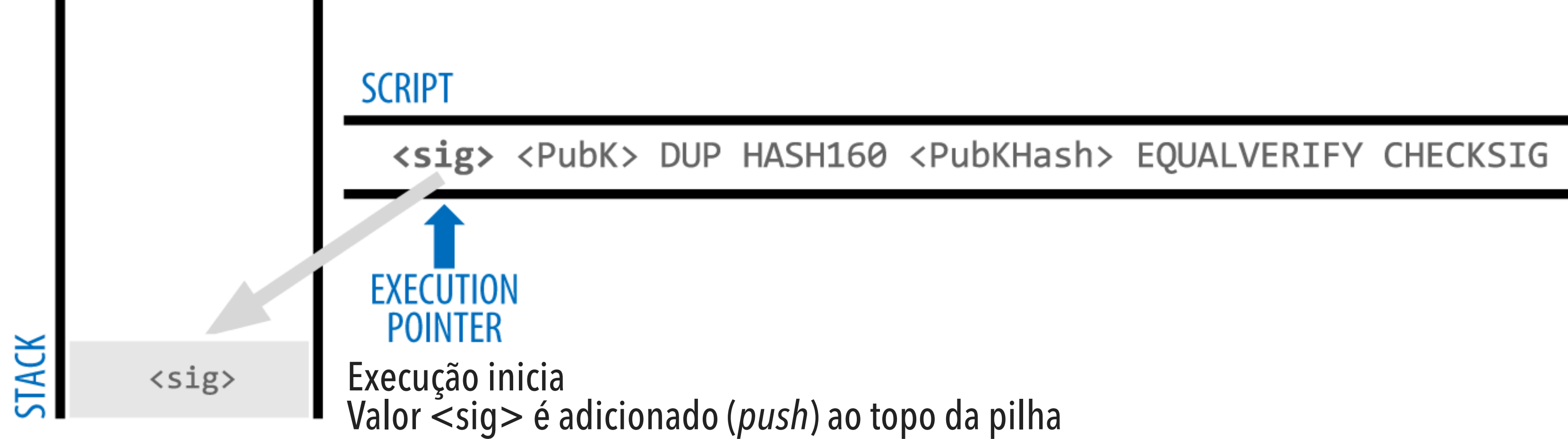
"scriptSig" : <Cafe Signature> <Cafe Public Key>

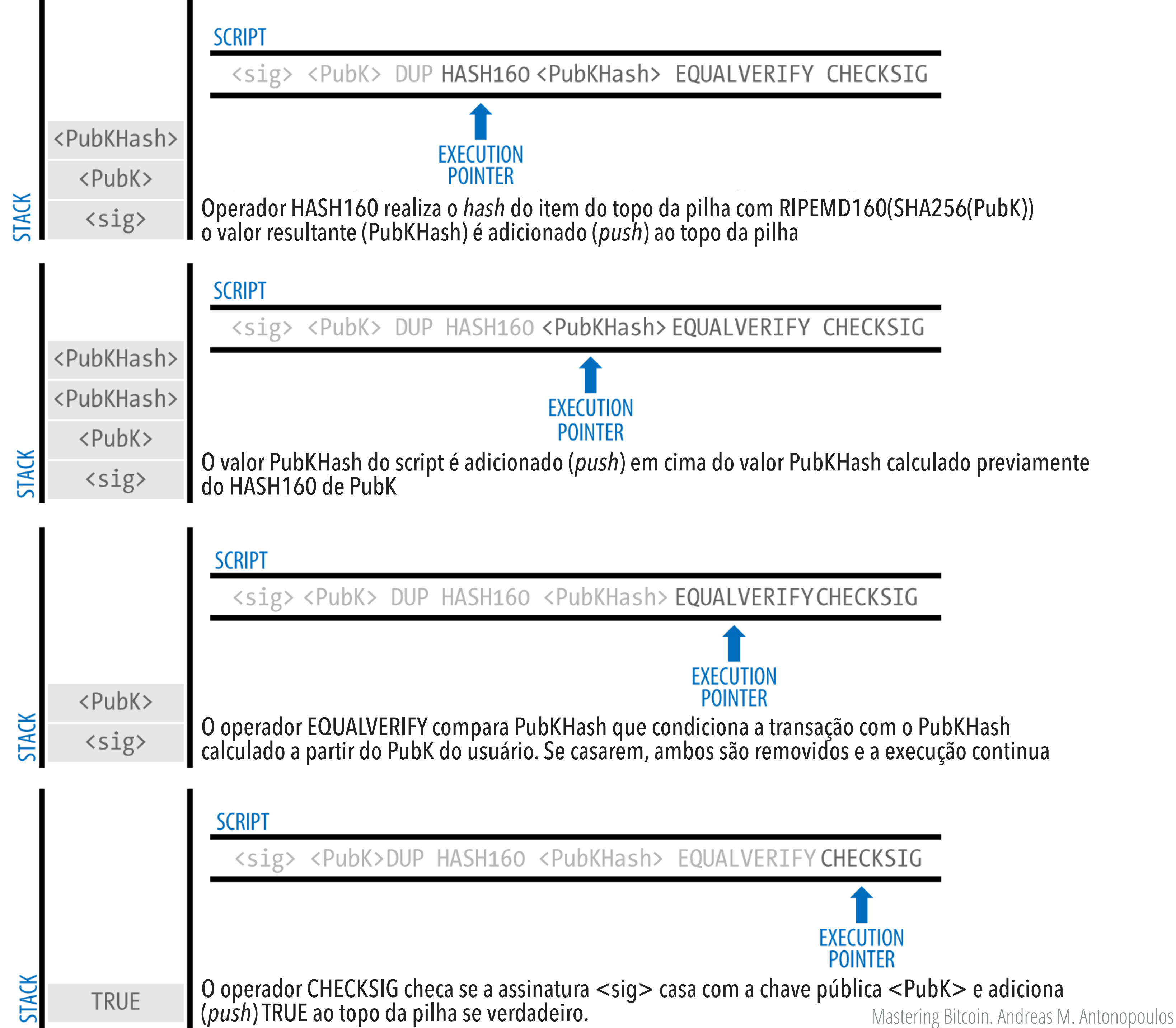
"scriptPubKey": OP_DUP OP_HASH160 <Cafe Public Key Hash> OP_EQUALVERIFY OP_CHECKSIG

<Cafe Signature> <Cafe Public Key> OP_DUP OP_HASH160 <Cafe Public Key Hash> OP_EQUALVERIFY OP_CHECKSIG

"scriptSig" : "3045022100884d142d86652a3f47ba4746ec719bbfbd040a570b1deccbb6498c75c4ae24cb02204b9f039..."

"scriptPubKey": "OP_DUP OP_HASH160 ab68025513c3dbd2f7b92a94e0581f5d50f654e7 OP_EQUALVERIFY OP_CHECKSIG"







TRANSAÇÃO

[Home](#) / [Block - 277316](#) / Transaction 0627052b6f28912f2703066a912ea577f2ce4da4caa5a5fbd8a57286c345c2f2

Summary

Height	277316	Input	0.10000000 BTC
Confirmations	395817	Output	0.09950000 BTC
Timestamp	2013-12-27 20:11:54	Sigops	8
Size (rawtx)	258 Bytes	Fees	0.00050000 BTC
Virtual Size	258 Bytes	Fees Rate (BTC / kVB)	0.00193798 BTC
Weight 	1,032	Other Explorers	 BLOCKCHAIR

Input (1)	0.10000000 BTC	Output (2)	0.09950000 BTC
-----------	----------------	------------	----------------

◀ 1Cdid9KFAaatwczBwBttQcwXYCpvK8h7FK

0.10000000

1GdK9UzpHBzqzX2A9JFP3Di4weBwqgmoQA

0.01500000 ▶

1Cdid9KFAaatwczBwBttQcwXYCpvK8h7FK

0.08450000 ▶

395,817 Confirmations

TRANSAÇÃO

<https://btc.com/1GdK9UzpHBzqzX2A9JFP3Di4weBwqgmoQA>

Home / Address - 1GdK9UzpHBzqzX2A9JFP3Di4weBwqgmoQA

Summary

Address1GdK9UzpHBzqzX2A9JFP3Di4weBwqgmoQA

Balance0.26594525 BTC

Total Received0.26594525 BTC

Tx Count33

Other ExplorersBLOCKCHAIR

Transaction33

Unconfirmed Transaction0

Stats

Mentions0

Export

c70231ff868388d31a525ac2c783ea56bacc2d1138c80ff619fa3bda4be69980

7 Satoshis/vByte0.00001665 BTC567,8512019-03-19 14:50:54

15c6bgm94Sth49ba7TAeVgrcHaXh2xYZ4m

0.00367868

>

1DnZiAXMeB2ba8XGV43yyhKxnKejbeCxW90.00361203

1GdK9UzpHBzqzX2A9JFP3Di4weBwqgmoQA0.00005000

+ 0.00005000

105,302 Confirmations

e6680df8309ee3d74f07d7a347d5aaa53ef3525221977a6a561012cf64b9d273

515 Satoshis/vByte0.00115982 BTC482,8092017-08-31 11:34:48

1Ncih7xWc17bkvUPsMaawBuUnyxxr4FQxG

0.01660349

>

1GdK9UzpHBzqzX2A9JFP3Di4weBwqgmoQA0.01500000

1DcMcwaSzrA1RpMiauD3zMaCYm1XRy4oY0.00044367

OP_RETURN

- ▶ Como escrever dados arbitrários no blockchain?
 - ▶ OP_RETURN <data>
- ▶ Até 80 *bytes*
- ▶ Esse script de saída não pode ser gasto - prova de que você destruiu a moeda (usado como *proof-of-burn* em algumas criptos)
- ▶ Qualquer coisa depois de OP_RETURN não é processado, então dados arbitrários podem ser inseridos

OP_RETURN


- ▶ Casos de uso:
 - ▶ Prova de existência de algo em um instante de tempo específico
 - ▶ <https://proofofexistence.com/>
 - ▶ *Proof-of-burn* para outras criptomoedas *
 - ▶ Counterparty (CXP)
 - ▶ Chaves públicas geradas sem chave privada
 - ▶ <https://btc.com/ICounterpartyXXXXXXXXXXXXXXXXXXXXXXXXXXXXUWLpVr>

OP_RETURN

<https://btc.com/8bae12b5f4c088d940733dcd1455efc6a3a69cf9340e17a981286d3778615684>

[Home](#) / [Block - 308570](#) / Transaction 8bae12b5f4c088d940733dcd1455efc6a3a69cf9340e17a981286d3778615684

Summary

Height	308570	Input	0.00220000 BTC
Confirmations	364585	Output	0.00200000 BTC
Timestamp	2014-06-30 02:45:09	Sigops	4
Size (rawtx)	254 Bytes	Fees	0.00020000 BTC
Virtual Size	254 Bytes	Fees Rate (BTC / kVB)	0.00078740 BTC
Weight ⓘ	1,016	Other Explorers	 BLOCKCHAIR

Input (1)	0.00220000 BTC	Output (2)	0.00200000 BTC
1HnhWpkMHMjgt167kvgcPyrMmsCQ2WPgg	0.00220000	Unable to decode output address	0.00000000 ➤
		1HnhWpkMHMjgt167kvgcPyrMmsCQ2WPgg	0.00200000 ➤

Input Scripts

4830450220446df4e6b875af246800c8c976de7cd6d7d95016c4a8f7bcd8ba81679cbda242022100c1ccfacfeb5e83087894aa8d9e37b11f5c054a75d030d5bfd94d17c5bc953d4a0141045901f6367ea950a5665335065342b952c5d5d60607b3cdc6c69a03df1a6b915aa026b5e07095a2548a98dcdd84d875c6a3e130bafadfd45e694a3474e71405a4

Output Scripts

NULL_DATA OP_RETURN 636861726c6579206c6f766573206865696469

P2PKH OP_DUP OP_HASH160 b8268ce4d481413c4e848ff353cd16104291c45b OP_EQUALVERIFY OP_CHECKSIG

636861726c6579206c6f766573206865696469
=
charley loves heidi

MULTIASSINATURA (MULTISIG)

- ▶ Esquema M -de- N
 - ▶ N : total de chaves
 - ▶ M : *threshold* de chaves necessárias
- ▶ Script P2MS (*pay to multisig*)



```
M <Public Key 1> <Public Key 2> ... <Public Key N> N CHECKMULTISIG
```


MULTIASSINATURA (MULTISIG)

2 <Chave Pública A> <Chave Pública B> <Chave Pública C> 3 CHECKMULTISIG

<Assinatura B> <Assinatura C>

<Assinatura B> <Assinatura C> 2 <Chave Pública A> <Chave Pública B> <Chave Pública C> 3 CHECKMULTISIG



TRANSAÇÃO VÁLIDA!

P2PKH VS P2SH

- ▶ No Bitcoin, quem envia especifica um *locking script*, e quem recebe provê um *unlocking script*
- ▶ **Pay-to-Pub-Key-Hash (P2PKH):** Vendedor (recebedor da transação) diz “envie suas moedas para o hash dessa chave pública”.
 - ▶ Caso mais simples e mais comum
- ▶ **Pay-to-Script-Hash (P2SH):** Vendedor diz “envie suas moedas para o hash desse script; Eu vou prover o script e os dados para fazer o script retornar *TRUE* quando eu resgatar as moedas”
- ▶ Vendedor não pode dizer, por exemplo: “para me pagar, escreva um script de saída complicado que me permita gastar usando multi-assinaturas”

POR QUE P2SH?

- ▶ Dispensa a escrita complexa de *script* pelos recebedores
- ▶ Faz mais sentido do ponto de vista de pagador-beneficiário
 - ▶ Mercador (ao invés do cliente) é responsável por escrever o *script* de maneira correta e segura
 - ▶ Cliente não quer saber do *script*
- ▶ Exemplo: **multisig**
 - ▶ M -de- N assinaturas especificadas podem resgatar e gastar a saída da transação

P2SH

Mohammed tem uma empresa de importação/exportação e usa o recurso de multi-assinatura do Bitcoin

Para resgate de recursos é necessário 2-de-5 assinaturas, entre ele, seus três parceiros e seu advogado:

```
2 <Chave Pública de Mohammed> <Chave Pública Parceiro1> <Chave Pública Parceiro2> <Chave Pública Parceiro3> <Chave Pública do Advogado> 5 CHECKMULTISIG
```

Sem P2SH:

1. Mohammed precisaria comunicar esse script a todos os seus clientes antes do pagamento;
2. Todo cliente teria que utilizar um software de carteira especial para suportar multisig, e entender essa funcionalidade;
3. A transação resultante seria aproximadamente 5x maior (devido as longas chaves públicas);
4. O fardo da grande transação recairia sobre o cliente, na forma de taxas de transação (*fees*).

P2SH

- ▶ Com P2SH, *locking scripts* complexos são substituídos pelo seu *fingerprint* digital, ou seja, seu *hash* criptográfico
- ▶ Quando uma nova transação quiser gastar esse UTXO, deve apresentar:
 - ▶ o *script* que casa com a *hash*
 - ▶ o *unlocking script*
- ▶ Em P2SH, o *locking script* que é substituído pela *hash* é chamado de *redeem script*

P2SH

Script complexo sem P2SH:

<i>Locking Script</i>	2 PubKey1 PubKey2 PubKey3 PubKey4 PubKey5 5 CHECKMULTISIG
<i>Unlocking Script</i>	Sig1 Sig2

Script complexo com P2SH:

<i>Redeem Script</i>	2 PubKey1 PubKey2 PubKey3 PubKey4 PubKey5 5 CHECKMULTISIG
<i>Locking Script</i>	HASH160 <20-byte hash do redeem script> EQUAL
<i>Unlocking Script</i>	Sig1 Sig2 <redeem script>

P2SH

Redeem script de Mohammed:

2 <Chave Pública de Mohammed> <Chave Pública Parceiro1> <Chave Pública Parceiro2> <Chave Pública Parceiro3> <Chave Pública do Advogado> 5 CHECKMULTISIG

Redeem script de Mohammed, substituindo por chaves:

2
04C16B8698A9ABF84250A7C3EA7EEDEF9897D1C8C6ADF47F06CF73370D74DCCA01CDCA79DCC5C395D7EEC6984D83F1F50C900A24DD47F569FD4193AF5DE762C587
04A2192968D8655D6A935BEAF2CA23E3FB87A3495E7AF308EDF08DAC3C1FCBFC2C75B4B0F4D0B1B70CD2423657738C0C2B1D5CE65C97D78D0E34224858008E8B49
047E63248B75DB7379BE9CDA8CE5751D16485F431E46117B9D0C1837C9D5737812F393DA7D4420D7E1A9162F0279CFC10F1E8E8F3020DECDBC3C0DD389D9977965
0421D65CBD7149B255382ED7F78E946580657EE6FDA162A187543A9D85BAAA93A4AB3A8F044DADA618D087227440645ABE8A35DA8C5B73997AD343BE5C2AFD94A5
043752580AFA1ECED3C68D446BCAB69AC0BA7DF50D56231BE0AABF1FDEEC78A6A45E394BA29A1EDF518C022DD618DA774D207D137AAB59E0B000EB7ED238F4D800
5 CHECKMULTISIG

Hash de 20 bytes do redeem script de Mohammed:

54c557e07dde5bb6cb791c7a540e0a4796f5e97e

Locking Script da transação:

HASH160 54c557e07dde5bb6cb791c7a540e0a4796f5e97e EQUAL

P2SH: VALIDANDO OS SCRIPTS

Primeiro o *redeem script* é conferido com o *locking script* para garantir que o *hash* casa:

```
<2 PK1 PK2 PK3 PK4 PK5 5 CHECKMULTISIG> HASH160 <redeem scriptHash> EQUAL
```

Se o *redeem script* casa, o *unlocking script* é executado para destravar o *redeem script*:

```
<Sig1> <Sig2> 2 PK1 PK2 PK3 PK4 PK5 5 CHECKMULTISIG
```

P2SH

https://btc.com/d3adb18d5e118bb856fbea4b1af936602454b44a98fc6c823aedc858b491fc13

Home / Block - 232593 / Transaction d3adb18d5e118bb856fbea4b1af936602454b44a98fc6c823aedc858b491fc13

Summary

Height	232593	Input	0.10000000 BTC
Confirmations	440564	Output	0.09980000 BTC
Timestamp	2013-04-22 11:29:22	Sigops	0
Size (rawtx)	222 Bytes	Fees	0.00020000 BTC
Virtual Size	222 Bytes	Fees Rate (BTC / kVB)	0.00090090 BTC
Weight	888	Other Explorers	BLOCC

Input (1)	0.10000000 BTC	Output (1)	0.09980000 BTC
<div><div><div>1HJnNmse4FuWgcwKs12EnSxgmXY5AqYJvm</div><div>0.10000000</div></div><div><div>3JALUHKvqB7NToPA2jALntCUWmvsgYMyGj</div><div>0.09980000</div></div></div>			
<div>440,564 Confirmations</div>			

Input Scripts

48304502204c3da378d8323a7233892b8050f738da69daf765ddc0d9815d9ad352286b70c2022100dff11c19338daa85ec5b124434de3b4378ebe8c56950348d5f66197af1d04f09014104910ae6c9b41b04d366ea54e920663c691843bb83ef7336cd6a0f79b0ac82ee38d2ca23a24adc2348d82c8ca13f0db885712493e89d88551118a7e80ff66ab23c

Output Scripts

P2SH OP_HASH160 b4acb9d78d6a6256964a60484c95de490eaaae75 OP_EQUAL

https://btc.com/cc11ca9e9dc188663c41eb23b15370f68eded56b7ec54dd5bc4f2d2ae93addb2

Home / Block - 232595 / Transaction cc11ca9e9dc188663c41eb23b15370f68eded56b7ec54dd5bc4f2d2ae93addb2

Summary

Height	232595	Input	0.09980000 BTC
Confirmations	440563	Output	0.09980000 BTC
Timestamp	2013-04-22 11:58:23	Sigops	0
Size (rawtx)	436 Bytes	Fees	0.00020000 BTC
Virtual Size	436 Bytes	Fees Rate (BTC / kVB)	0.00090090 BTC
Weight	1,744	Other Explorers	BLOCC

Input (1)	0.09980000 BTC	Output (1)	0.09980000 BTC
<div><div><div>3JALUHKvqB7NToPA2jALntCUWmvsgYMyGj</div><div>0.09980000</div></div><div><div>3BnZYLFFeGAa4N4aTavSUhez7nAAnjjAJpB</div><div>0.09980000</div></div></div>			
<div>440,563 Confirmations</div>			

Input Scripts

00483044022018b88c05ab571cf31bf317a98ed5909cf43218f472bfbeb82b6857d3b1edf4ee0220686627e66b368e298114a097b5814a76fdaac23f7728a42d38415552ba68c8220101493046022100b7a70f4c3b2b5d24475f9664bb77b6046c0251c89b446aad8a86b584b74ed414022100d53dd27b741801a908fdf7c2ee259f444f549b56d7a6c682599ef67b1edb6d24014cc9524104f3d35132084eb1b99b6506178c20adb42d26296012e452e392689bdb6553db33ba24b900000892805de1646821c7b0fb50b3d879c26e2b493b7041e6215356a04104ab4ecc9e8ea2da0562af25bcaede00c4d5a00db60edc17672376decf0a35a34fdc9f1ffad1fb74fd7b1b198b9231c25df88e0769bec49975649b4b3f40adafb04104f7149f270717c00f6cc09b9ce3c22791c4aab1af40a5107aacca85b6f644cc0d84459e308f998d801b8d9d355f8ec33b0e41866841e2870754cf667a9821703d53ae