itcoin Script 101

Funktionsweise von Transaktionen im Bitcoin Netzwerk

Michael Schmoock mschmoock@cocus.com

Agenda

1. Einleitung
   - Abgrenzung / Vergleich
   - Transaktionen, Inputs, Outputs, Keys
2. Bitcoin Script
   - OpCodes, Stack
3. Transaktionstypen
   - SegWit, Addressformate, non-standard Tranksaktionen

Einleitung

Netzwerke wie Visa, PayPal und Swift implementieren:

- Salden, Beträge, Dispo
- Sender, Empfänger
- Konto, Account, Passwort

=> Sie modellieren unsere Vorstellung eines Geldsystems.

Einleitung

Bitcoin arbeitet mit flexiblen Bausteinen:

- Inputs, Outputs
- OpCodes, Scripts, Timelocks, Bedingungen
- Adressen und kryptographischen Schlüsseln
- Signaturen und Hashfunktionen

=> Hiermit lässt sich u.a. die Funktion eines Geldsystems abbilden

Einleitung – Internet vs Bitcoin

Das Internet ist ein programmierbares Datenvermittlungsnetzwerk.

→ Versendete Daten liegen Sendern und Empfängern vor.

Bitcoin ist ein programmierbares Wertvermittelungs Netzwerk.

→ Versendete Werte liegen nur den Empfängern vor.

Einleitung – Was ist Geld? Ultra-Kurzfassung

Neben etlichen ökonomischen und sozialen Aspekten, müssen Geldsysteme eines definieren:

„**Wer wie viel hat**“.

Anm.: Trifft auf alle uns bekannten Geldsystem zu.

Einleitung – Transaktion im Blockexplorer

Transaction da0caedb50f4f3661618fa1abd5208964e246d2562cdfe3233f5e108f37270dc

Summary

Size	223 (bytes)
Fee Rate	0.00045403587443946185 BTC per kB
Received Time	Oct 28, 2018 10:51:24 AM
Mined Time	Oct 28, 2018 10:51:24 AM
Included in Block	000000000000000000042bd77cf436f6e07df41922cf9e389ae9c7d5950aacf8

Details

da0caedb50f4f3661618fa1abd5208964e246d2562cdfe3233f5e108f37270dc — mined Oct 28, 2018 10:51:24 AM

1NzKE6CafBJq7z9B5muA4YfKL1XpQ47ewD 0.08358673 BTC

Confirmations:

scriptSig

304402203a5b071973eea976e688966cf5bba89ca23202b43e3337ba...

03a2999a8604de8e1aba5044213f619315fecb121c68ceea3dc0839e0...

3DoTdhVokovd4d58CKzT29aSrGhC43iDtd 0.00774141 BTC (U)

Type scripthash

scriptPubKey

OP_HASH160 84d977691a83cd55553a8b4fe99f77eaa42837de OP_EQUAL

17PNnqEMHvRacMuMYjB1YH9pAgWzHKPGdU 0.07574407 BTC (U)

Type pubkeyhash

scriptPubKey

OP_DUP OP_HASH160 460c186c6ad9edef4fae58df376bdc7df6a49a05 OP_EQUALV...

Einleitung – Addressen, Inputs, Outputs, Keys

1) Transaktionen bestehen aus Inputs und Outputs

2) Trasaktionen „überweisen“ Inputs auf neue Outputs

3) Von Outputs leiten sich Addressen ab (Temp. Kontonr.)

4) Addressen sind Hashes von Public Keys (base58)

5) Private Keys signieren sog. unspent Outputs als neue Inputs

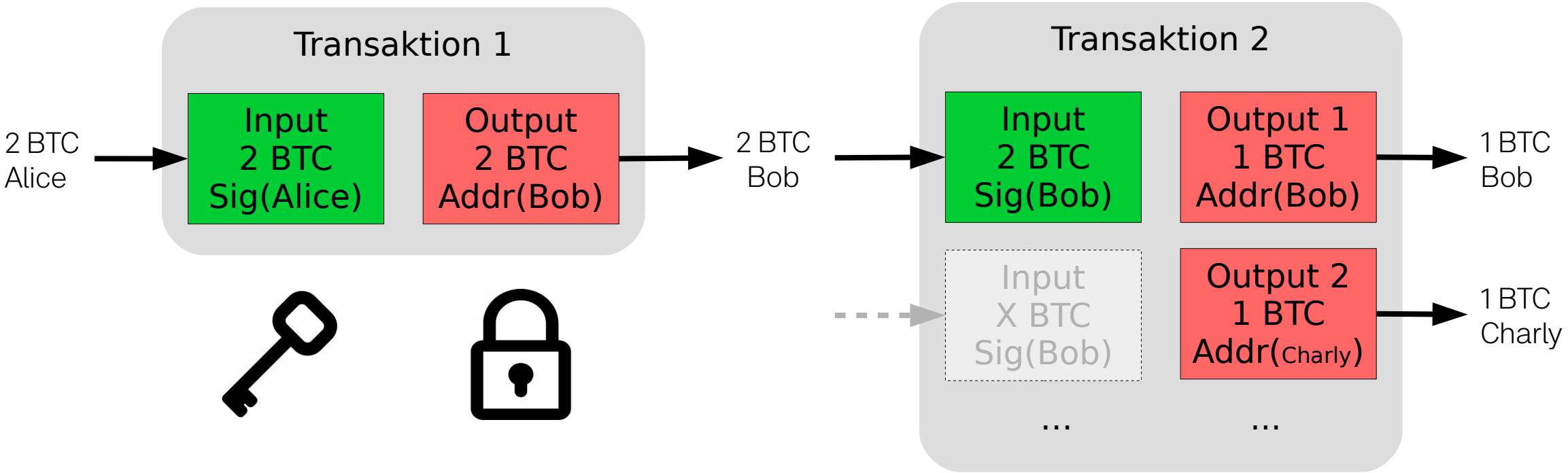
Bitcoin Script – Einsatzgebiete

- Validierung von Bitcoin Transaktionen
- Setzen von Konditionen die in der Zukunft erfüllt werden müssen um bitcoins durch ihren neuen Besitzer zu bewegen: **scriptPubKey** bzw. locking Script.
- Scripts welche diese Konditionen erfüllen: **scriptSig** bzw. unlocking Script.
- Alle Teilnehmer müssen diese Scripts unabhängig voneinander ausführen und zu deterministischen Ergebnissen kommen.

Bitcoin Script – Eigenschaften

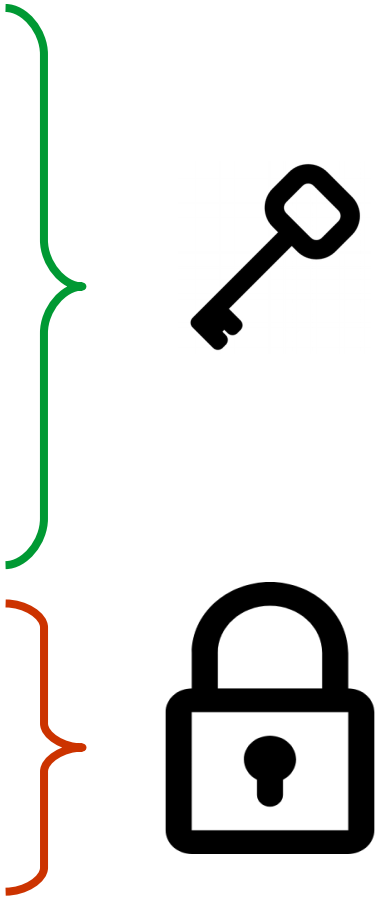
- Stackbasierte Programmiersprache (Forth)
- Geringe Komplexität und Systemanforderungen: CPU und Memory
- Spezielle Operatoren für kryptographische Funktionen
  - Hashing, Signatur Checks, Multisig Checks
- Touring incomplete (by design)
  - Halteproblem, Scripts müssen Terminieren
  - Keine Loops, klare Abbruchbedingungen
  - Jeder OpCode wird maximal einmal ausgeführt
  - Keine Angriffsvektoren durch komplexe Scripts

Einleitung – Transaktions Inputs und Outputs

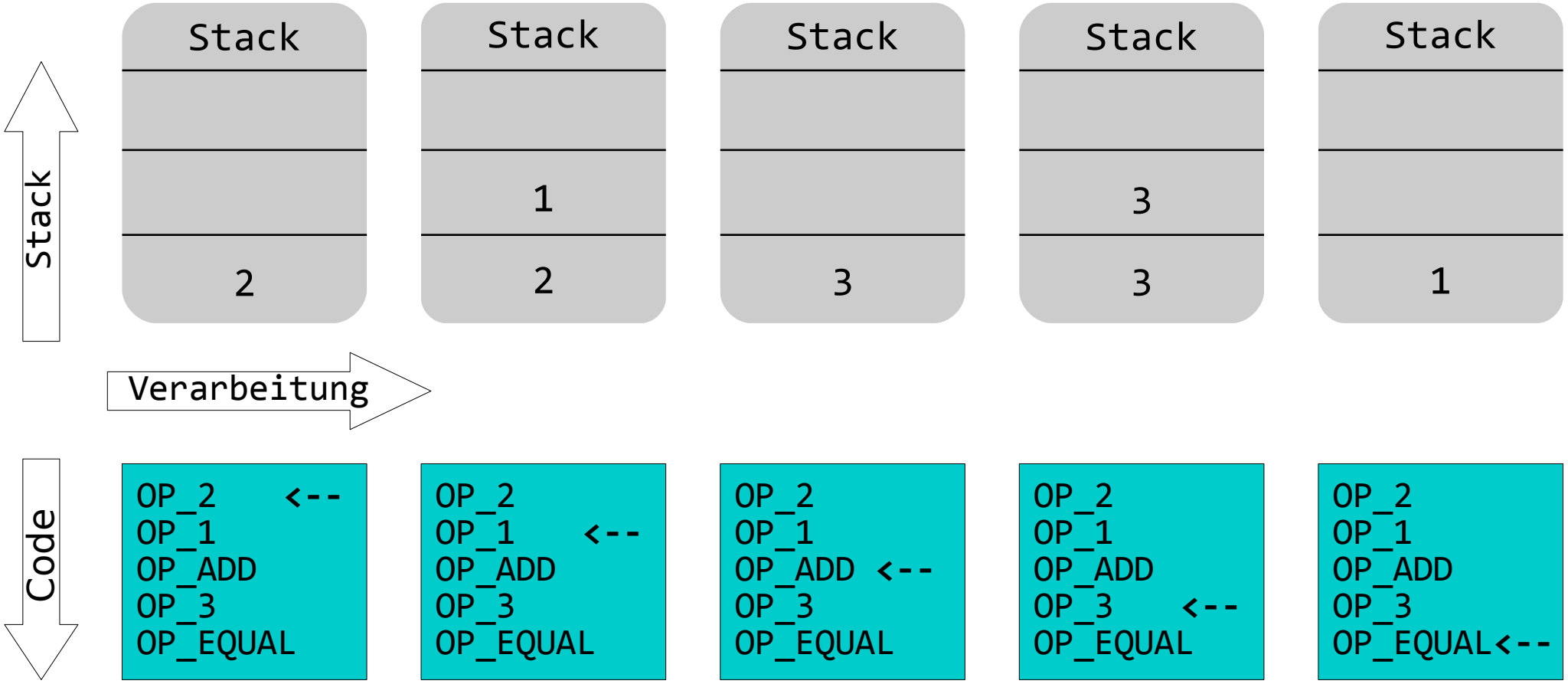


Einleitung – Serialisierte Transaktion

Schlüssel	Wert
Version	01 00 00 00
Anzahl Inputs	01
Hash Input TX (rev. Byteorder)	f2 59 bd f7 0e a6 0a e5 34 5e 46 57 0a a2 c1 89 0f 5f aa 67 a1 28 3b f4 8f 33 e9 b3 e9 70 4b 01
Index Input TX	00 00 00 00
Länge Input Script	6a
Input Script	47 30 44 02 20 3a 5b 07 19 73 ee a9 76 e6 88 96 6c f5 bb a8 9c a2 32 02 b4 3e 33 37 ba bb 19 f0 6b 07 81 af ac 02 20 79 71 1a 2d 4f 44 64 29 79 36 a8 64 ae a4 79 c2 e1 a3 c4 b8 b9 7f d1 5c 9c 25 db a7 6c 68 76 87 01 21 03 a2 99 9a 86 04 de 8e 1a ba 50 44 21 3f 61 93 15 fe cb 12 1c 68 ce ea 3d c0 83 9e 03 3f 53 31 05
Sequence	ff ff ff ff
Anzahl Outputs	02
Wert Output 1	fd cf 0b 00 00 00 00 00
Länge Output Script	17
Output Script	A9 14 84 d9 77 69 1a 83 cd 55 55 3a 8b 4f e9 9f 77 ea a4 28 37 de 87
<OUTPUT #2 Wert, Länge, Script>	...
Locktime	00 00 00 00



Script und Stack 1 + 2 == 3



Bitcoin Script – Eigenschaften

Alle OpCodes sind im Bitcoin Wiki erklärt: https://en.bitcoin.it/wiki/Script

Constants

When talking about scripts, these value-pushing words are usually omitted.

Word	Opcode	Hex	Input	Output	Description
OP_0, OP_FALSE	0	0x00	Nothing.	(empty value)	An empty array of bytes is pushed onto the stack. (This is not a no-op: an item is added to the stack.)
N/A	1-75	0x01-0x4b	(special)	data	The next *opcode* bytes is data to be pushed onto the stack
OP_PUSHDATA1	76	0x4c	(special)	data	The next byte contains the number of bytes to be pushed onto the stack.
OP_PUSHDATA2	77	0x4d	(special)	data	The next two bytes contain the number of bytes to be pushed onto the stack in little endian order.
OP_PUSHDATA4	78	0x4e	(special)	data	The next four bytes contain the number of bytes to be pushed onto the stack in little endian order.
OP_1NEGATE	79	0x4f	Nothing.	-1	The number -1 is pushed onto the stack.
OP_1, OP_TRUE	81	0x51	Nothing.	1	The number 1 is pushed onto the stack.
OP_2-OP_16	82-96	0x52-0x60	Nothing.	2-16	The number in the word name (2-16) is pushed onto the stack.

Flow control

Word	Opcode	Hex	Input	Output	Description
OP_NOP	97	0x61	Nothing	Nothing	Does nothing.
OP_IF	99	0x63	<expression> if [statements] [else [statements]]* endif		If the top stack value is not False, the statements are executed. The top stack value is removed.

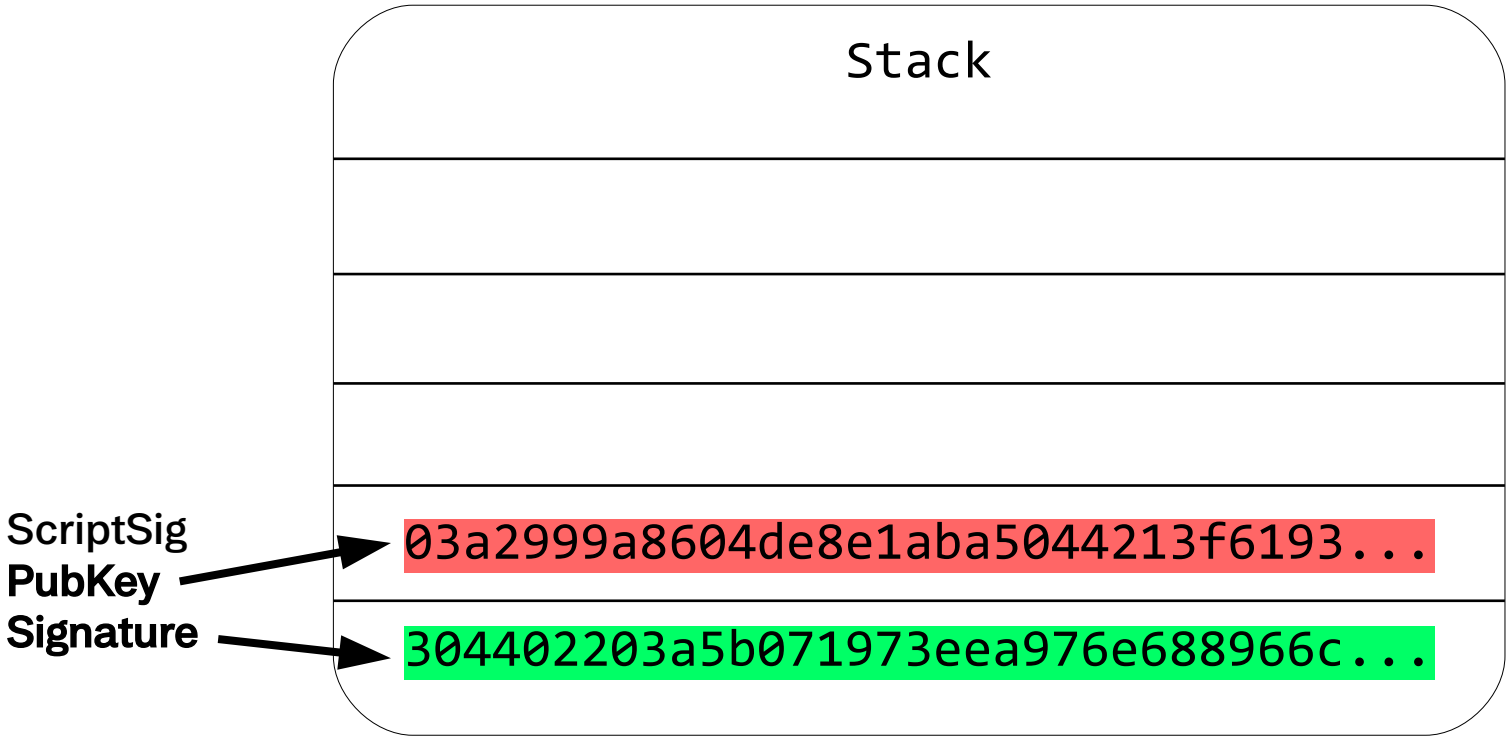
Pay to Public Key Hash - P2PKH

- Alle normale Addressen (Prefix 1) sind P2PKH Scripts.
- Trivialer Smart Contract
- Public Key wird erst sichtbar wenn der Betrag abgerufen wird

scriptPubKey: OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG
scriptSig: <sig> <pubKey>

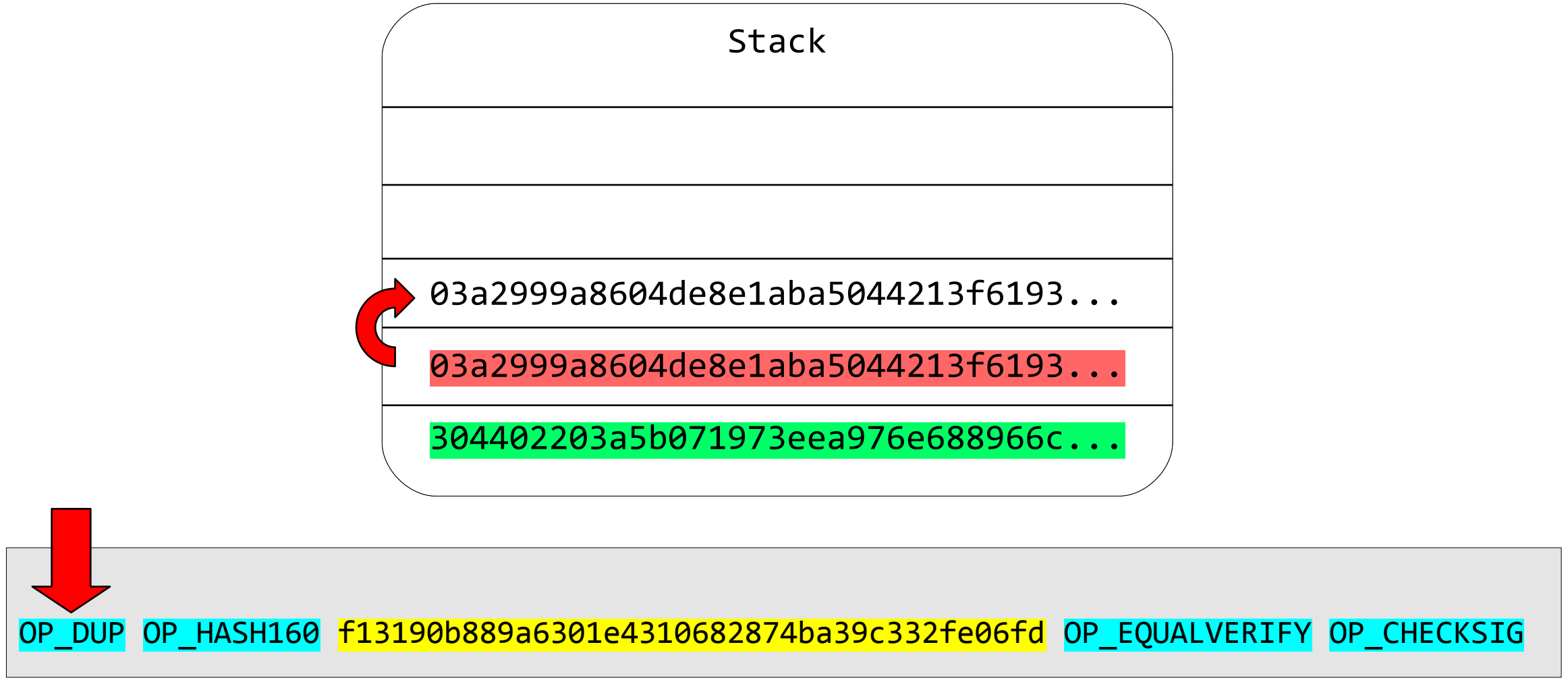
```
304402203a5b071973eea976e688966cf5bba89ca23202b43e3337babb19f06b0781afac022079711a2d
4f4464297936a864aea479c2e1a3c4b8b97fd15c9c25dba76c687687
03a2999a8604de8e1aba5044213f619315fecb121c68ceea3dc0839e033f533105 OP_DUP OP_HASH160
f13190b889a6301e4310682874ba39c332fe06fd OP_EQUALVERIFY OP_CHECKSIG
```

Pay to Public Key Hash - P2PKH

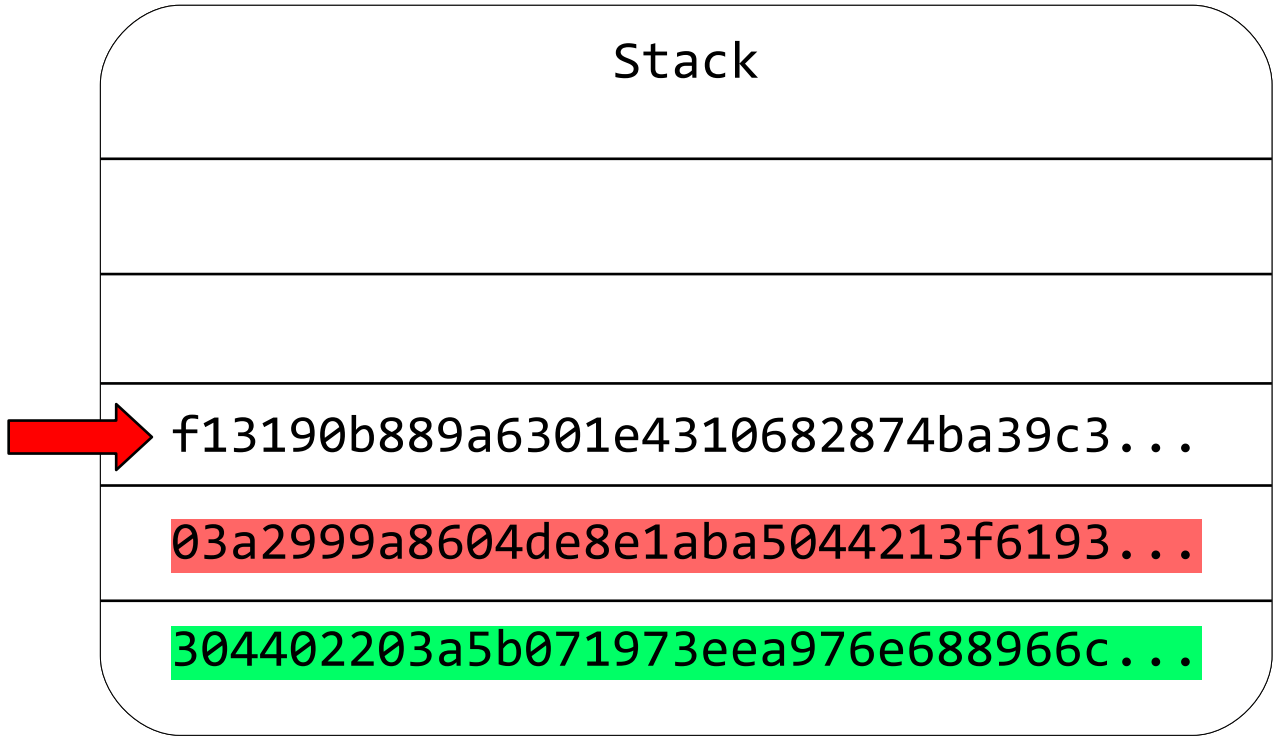



```
Script

OP_DUP OP_HASH160 f13190b889a6301e4310682874ba39c332fe06fd OP_EQUALVERIFY OP_CHECKSIG
```

Pay to Public Key Hash - P2PKH



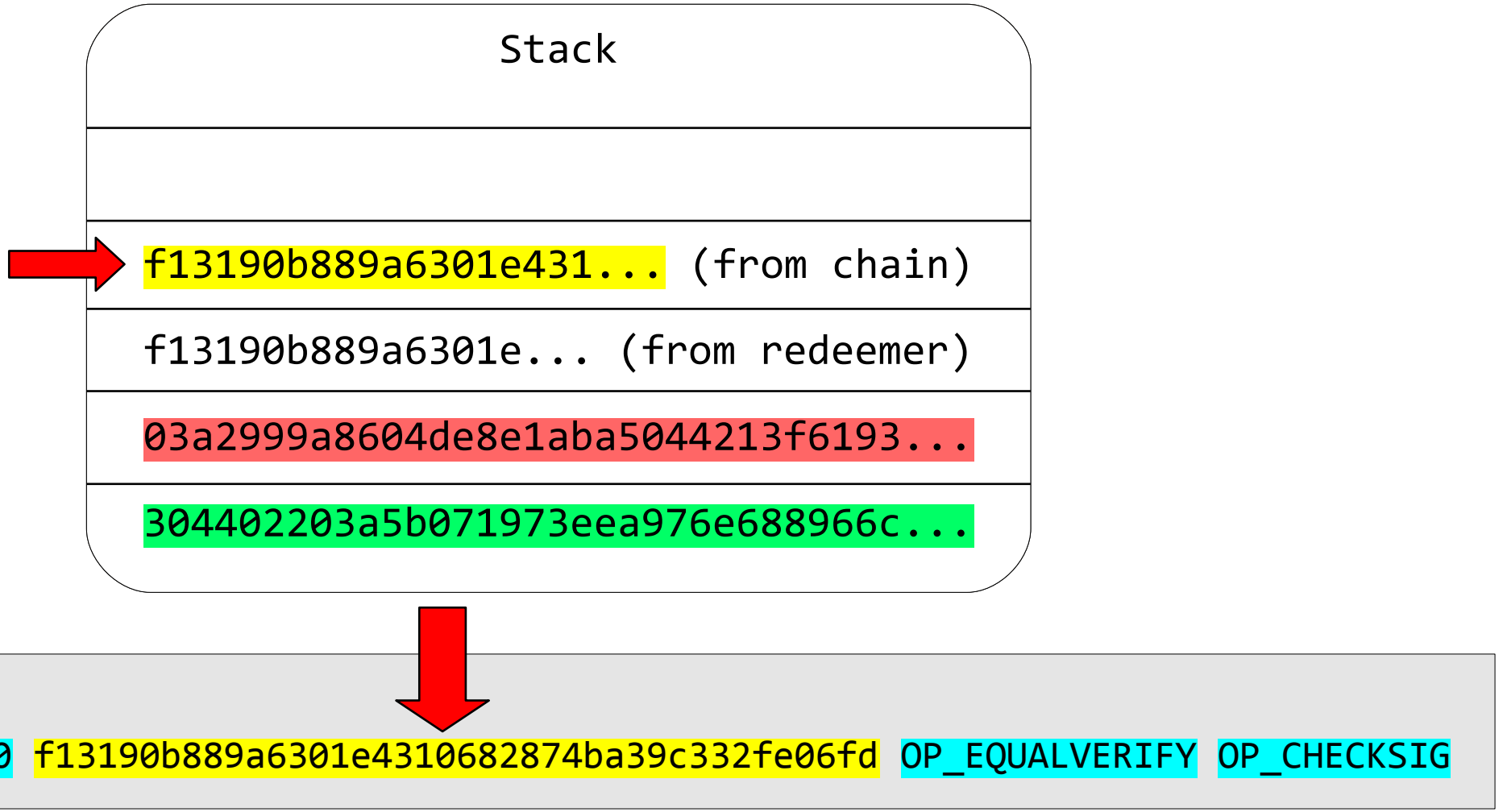
Pay to Public Key Hash - P2PKH



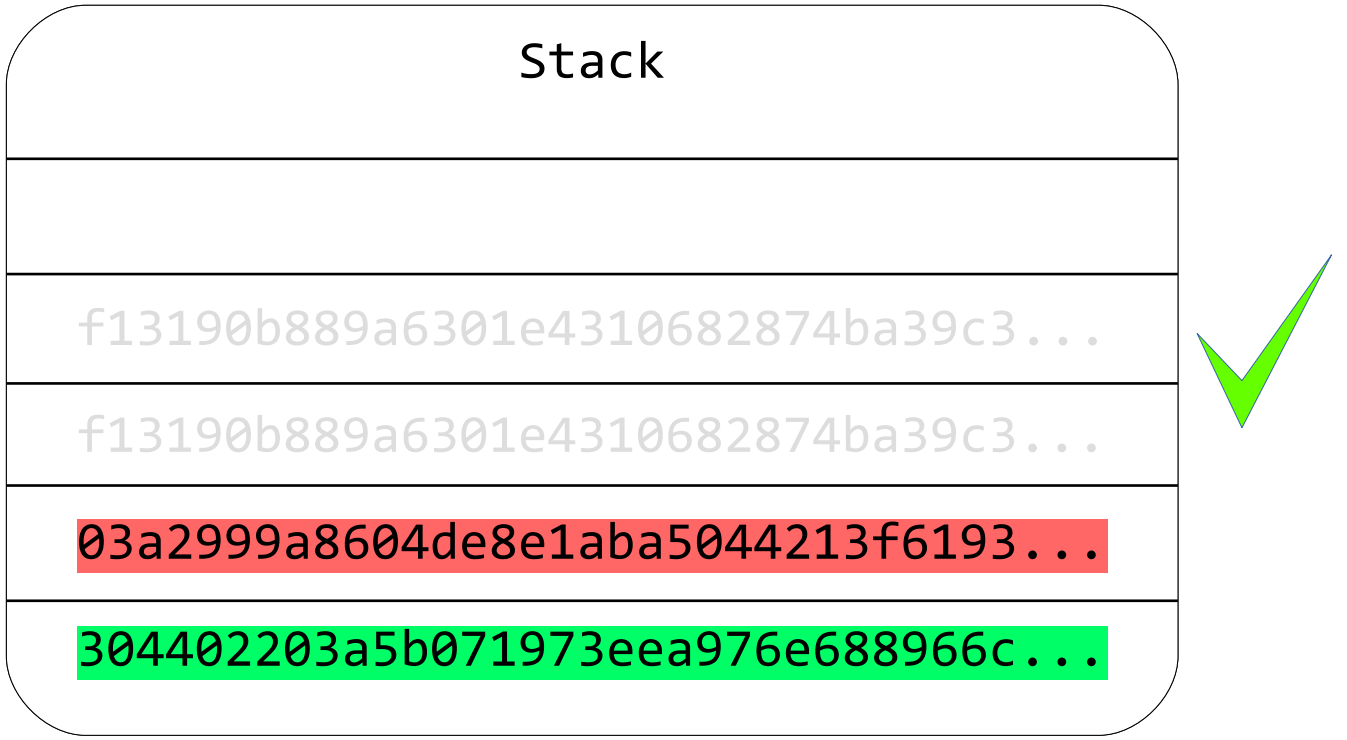
Script

OP_DUP OP_HASH160 f13190b889a6301e4310682874ba39c332fe06fd OP_EQUALVERIFY OP_CHECKSIG

Pay to Public Key Hash - P2PKH



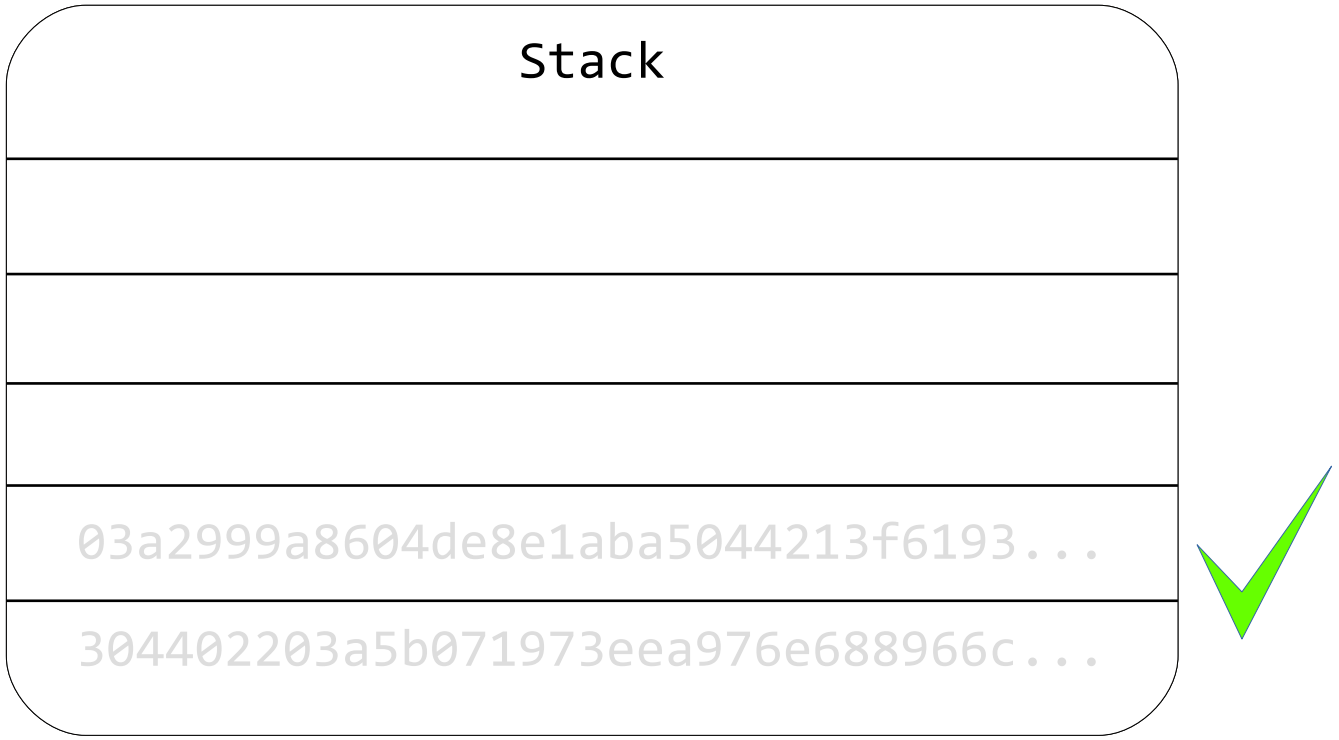
Pay to Public Key Hash - P2PKH



Script

OP_DUP OP_HASH160 f13190b889a6301e4310682874ba39c332fe06fd OP_EQUALVERIFY OP_CHECKSIG

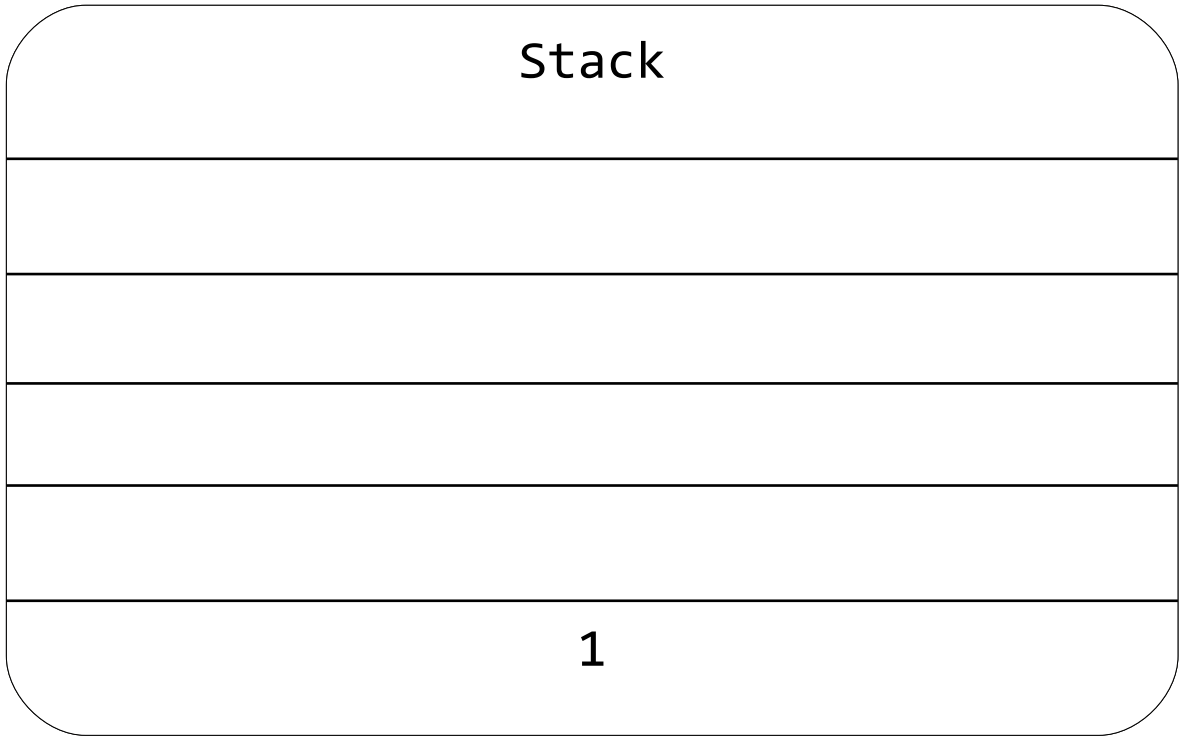
Pay to Public Key Hash - P2PKH



Script

```
OP_DUP OP_HASH160 f13190b889a6301e4310682874ba39c332fe06fd OP_EQUALVERIFY OP_CHECKSIG
```

Pay to Public Key Hash - P2PKH



Script

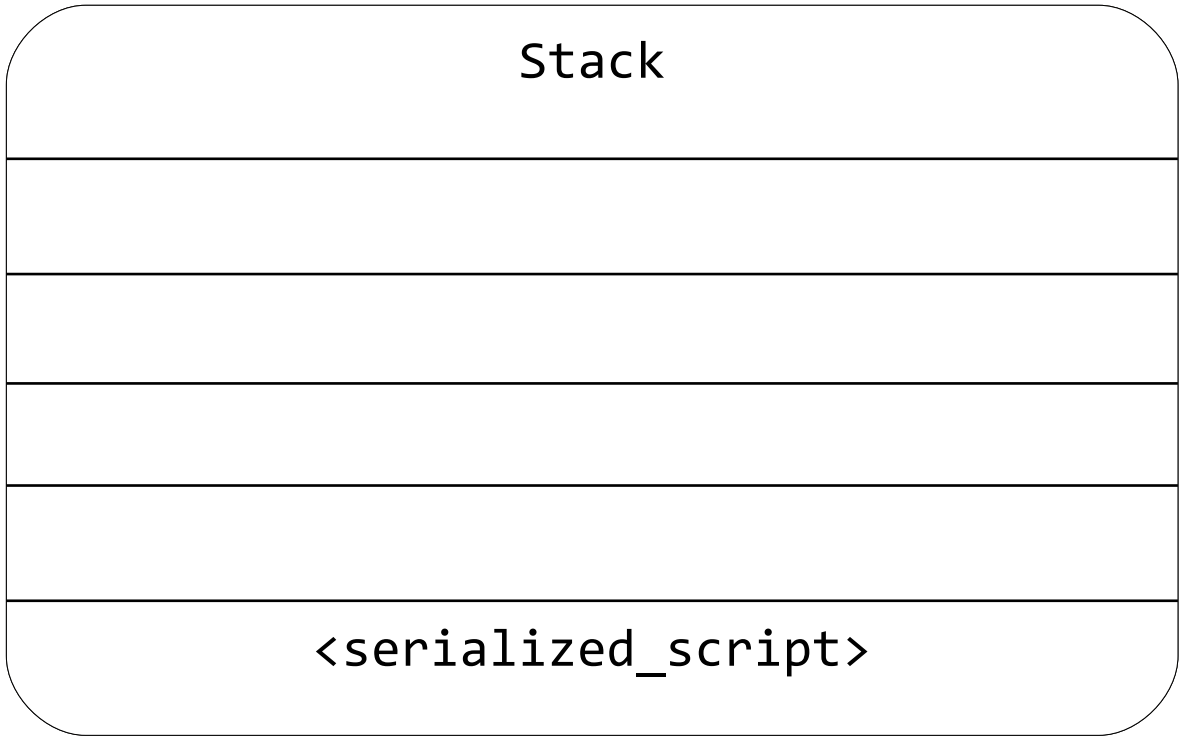
```
OP_DUP OP_HASH160 f13190b889a6301e4310682874ba39c332fe06fd OP_EQUALVERIFY OP_CHECKSIG
```

Pay to Script Hash - P2SH

- Komplexe Smart Contracts deren Addressen den Prefix ‚3' haben.
- Das unlocking Script ist der eigentliche Smart Contract.
- Script wird erst sichbar sobald der Contract ausgeführt werden soll.
- Locking Script stellt sicher, dass ein Script erst ausgeführt wenn sein Hash (Addresse) stimmt.

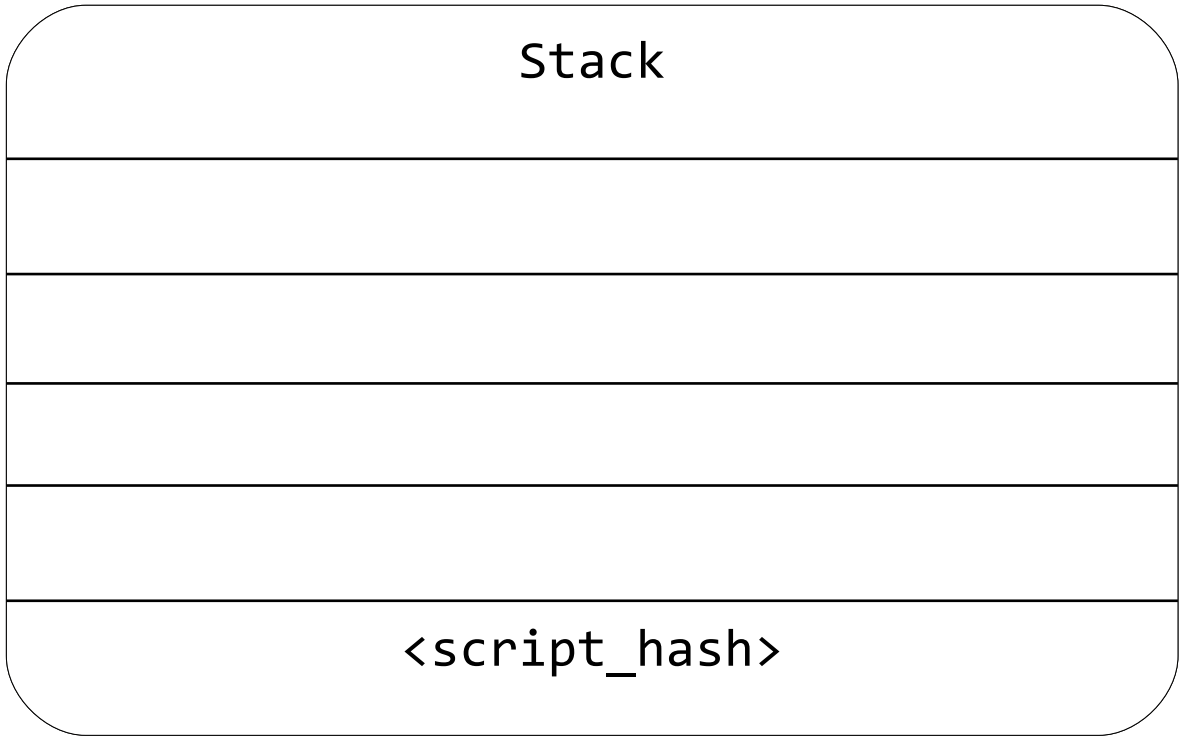
```
scriptPubKey:    OP_HASH160 <scriptHash> OP_EQUAL
scriptSig:       <script>
```

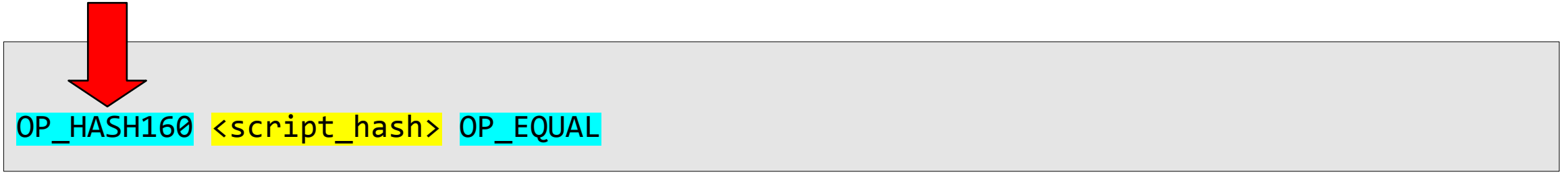
Pay to Public Key Hash - P2PKH



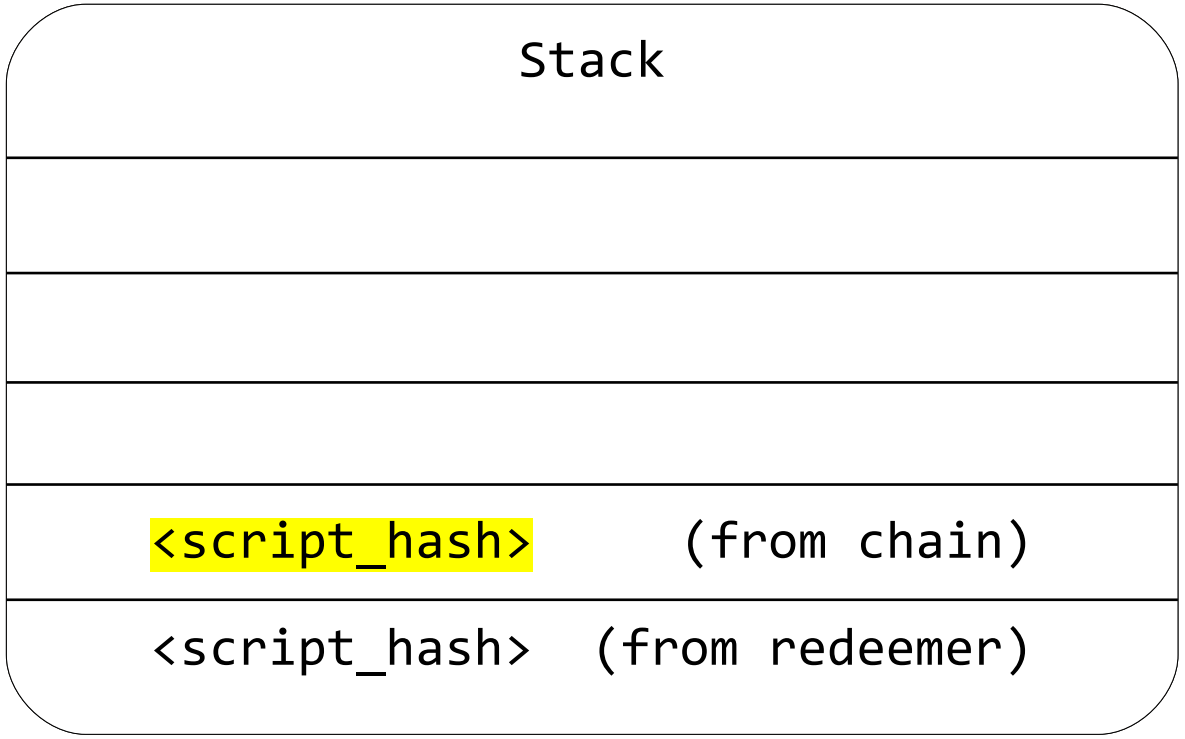

```
Script

OP_HASH160 <script_hash> OP_EQUAL
```

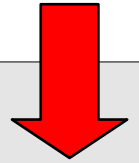
Pay to Public Key Hash - P2PKH





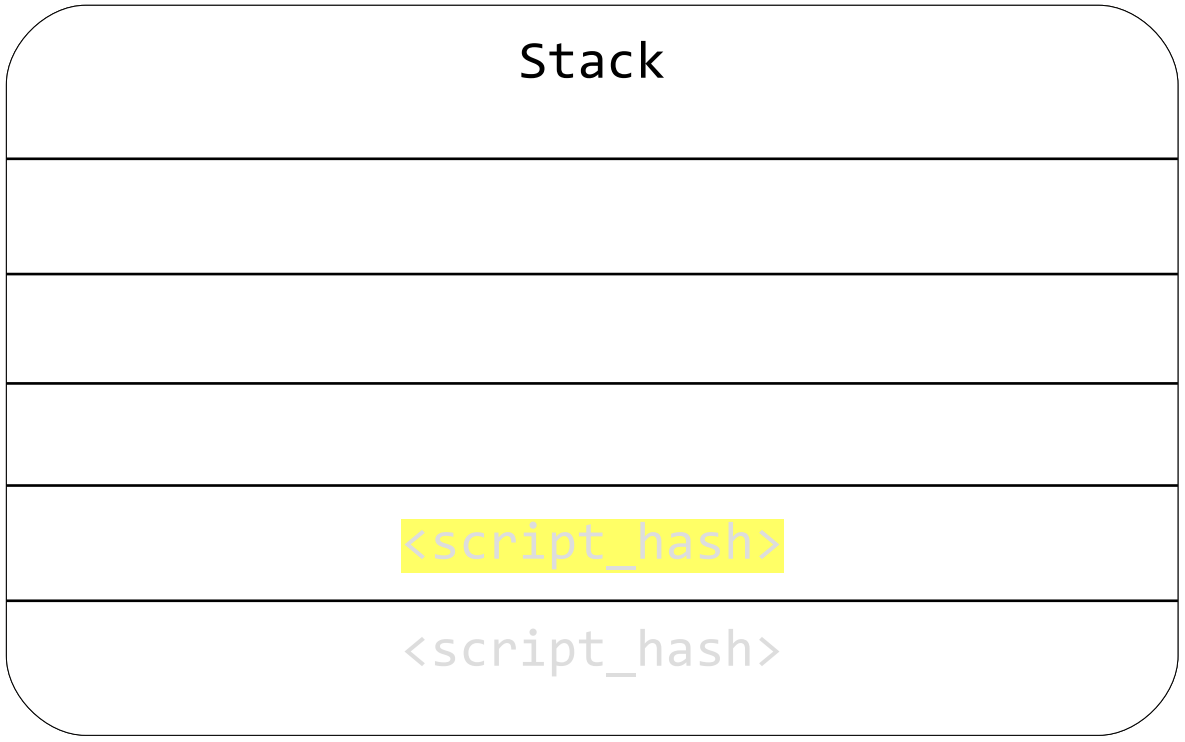
Pay to Public Key Hash - P2PKH

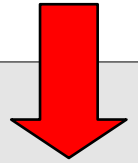




```
Script

OP_HASH160 <script_hash> OP_EQUAL
```

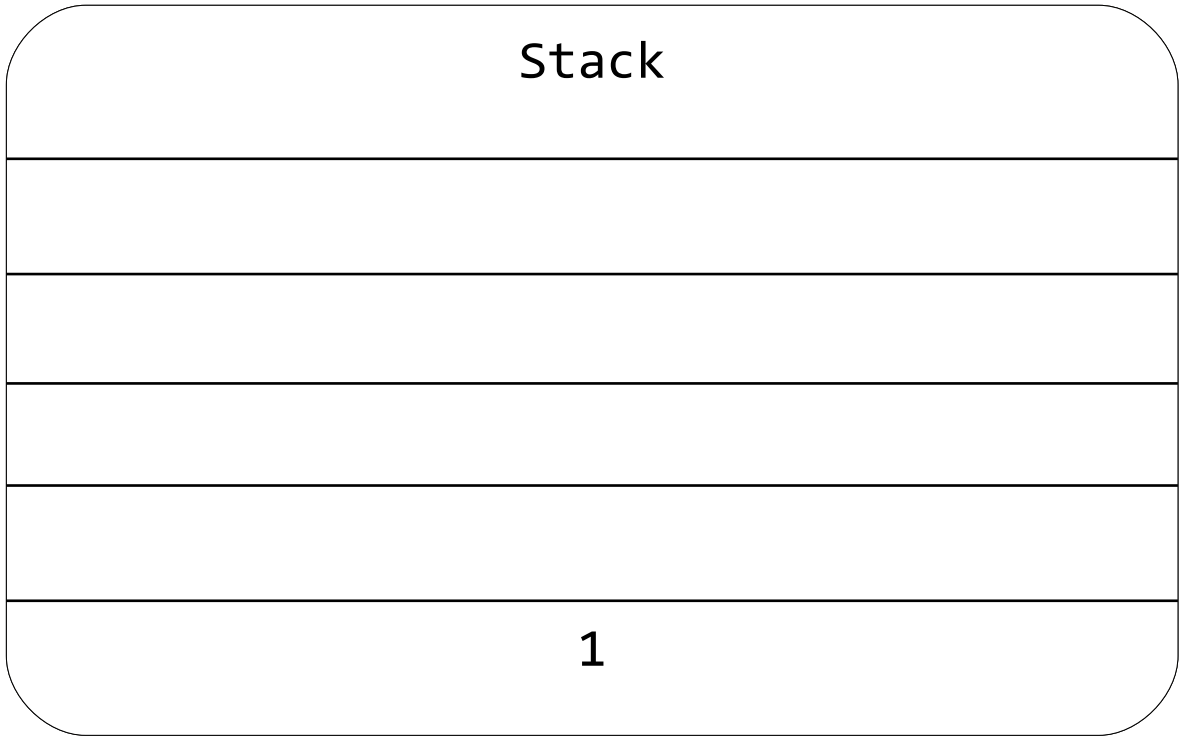
Pay to Public Key Hash - P2PKH

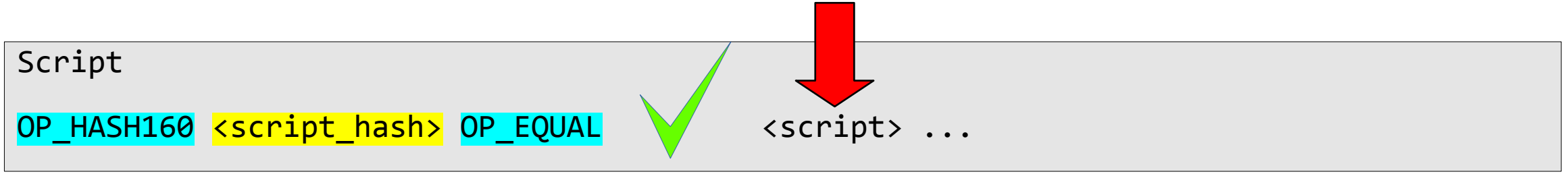




```
Script

OP_HASH160 <script_hash> OP_EQUAL
```

Pay to Public Key Hash - P2PKH





Standard Transaktionstypen

- **P2PK** Pay to Public Key (outdated)
- **P2PKH** Pay to Public Key Hash
- **P2SH** Pay to Script Hash - Multisig, Timelocks, ...
- **P2SH-P2WPKH** Pay to Script Hash SegWit (legacy support)
- **P2WKH** Pay to Witness Public Key Hash
- **P2WSH** Pay to Witness Script Hash

Alle non-standard Transaktionen werden im p2p Netzwerk i.d.r. nicht weitergeleitet, können aber dennoch durch die Miner direkt ausgeführt werden.

Nicht-Standard Transaktionen

- OP_RETURN (40 Bytes)
  - Timestamps, beliebige Daten
  - Proof of Burn
- Hash Collision Incentives

```
scriptPubKey: OP_2DUP OP_EQUAL OP_NOT OP_VERIFY OP_SHA1 OP_SWAP OP_SHA1 OP_EQUAL
scriptSig: <preimage1> <preimage2>
```

- Crypto Puzzles

```
scriptPubKey: OP_HASH256 6fe28c0ab6f1b372c1a6a246ae63f74f931e8365e15a089c68d6190000000000 OP_EQUAL
```

- Bare Multisig, Timelock Contracts

Vielen Dank

Fragen und Antworten

Quellen und Links

- https://bitcoin.org/bitcoin.pdf
- https://blockexplorer.com/tx/da0caedb50f4f3661618fa1abd5208964e246d2562cdfe3233f5e108f37270dc
- https://en.bitcoin.it/wiki/Script
- https://siminchen.github.io/bitcoinIDE/build/editor.html
- http://n.bitcoin.ninja/checkscript