



# Mining threats in Namecoin

Alexey Goncharov,  
*Positive Technologies*



**BREAKING**  
**THE CONSTANT**

A large, stylized dollar sign graphic is centered in the background, composed of several overlapping semi-circles in shades of orange and yellow. To the right of this graphic, the word 'BREAKING' is written vertically in bold red capital letters. To the right of 'BREAKING', the words 'THE CONSTANT' are written vertically in large, white, all-caps capital letters.

# Namecoin



Alternate DNS registrar based on blockchain

- censor-proof, domains cannot be seized or blackholed

Maintains unofficial [.bit](#) TLD

Resolvers:

- nodes of OpenNIC project
- plugins for Chrome, Firefox and Opera
- [ncdns](#) - opensource project for full-featured authoritative server

Used by:

- Chthonics
- Dimnie
- [RTM](#)
- GandCrab
- Smoke Loader
- Neutrino

«[Bitcoin frees money – Namecoin frees DNS](#)»

# Agenda

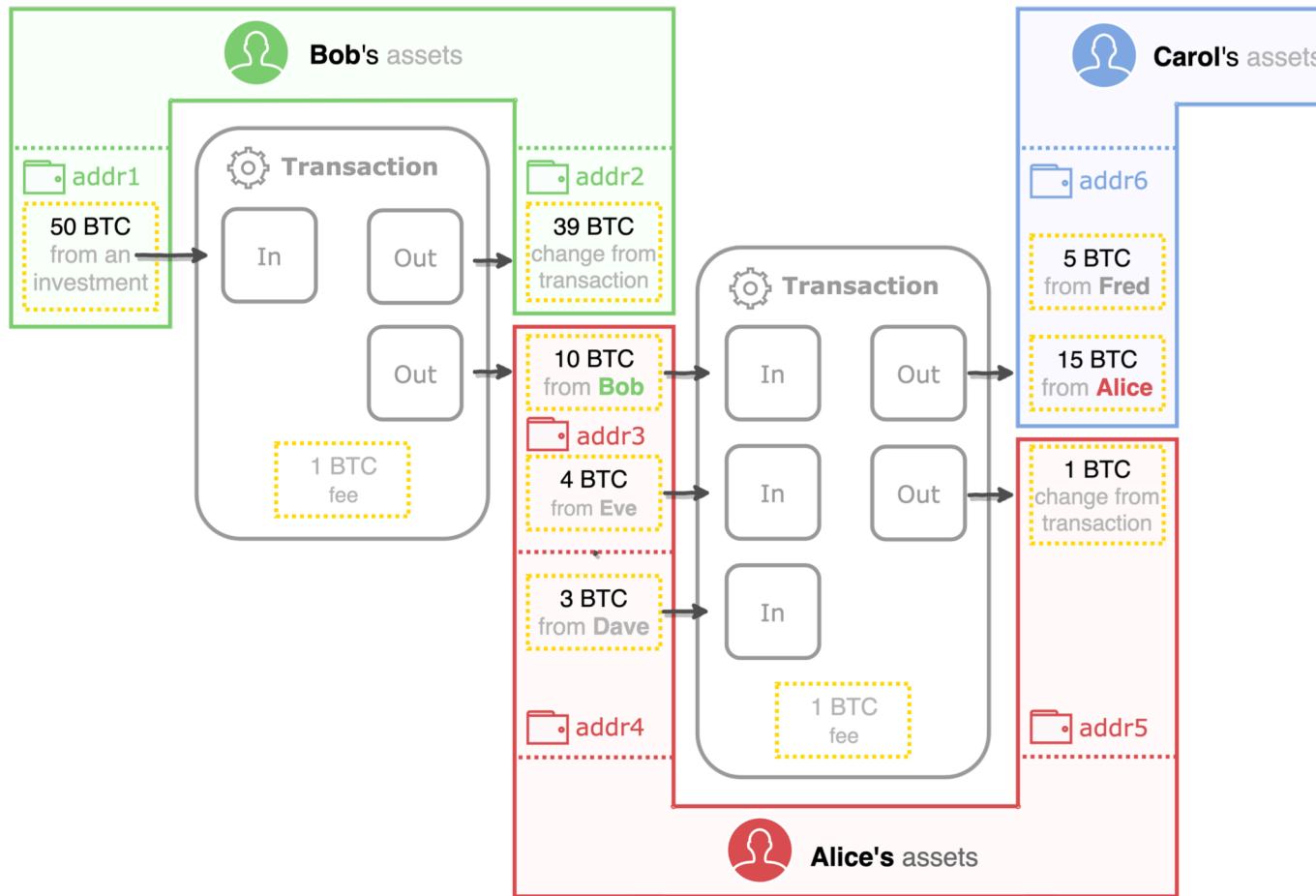


- Bitcoin 201
  - How Namecoin works
  - Mapping assets in Namecoin
  - Takeaways
- # science ends here
- Bonus track #1
  - Bonus track #2



# Bitcoin 201

# Bitcoin flow



# Transaction structure



```
{ "hash": "90b18aa54288ec610d83ff1abe90f10d8ca87fb6411a72b2e56a169fdc9b0219",
  "ver": 1,
  "lock_time": 0,
  "size": 226,
  "vin_sz": 1,
  "vout_sz": 2, }
```

"in": [

```
  { "prev_out": {
      "hash": "18798f8795ded46c3086f48d5bdabe10e1755524b43912320b81ef547b2f939a",
      "n": 0 },
    "scriptSig": "3045022100clefcad5cdcc0dcf7c2a79d9e1566523af9c7229c78..." } ],

```

"out": [

```
  { "value": 4000000000,
    "scriptPubKey": "OP_DUP OP_HASH160 4b358739fc... OP_EQUALVERIFY OP_CHECKSIG" },
  { "value": 1000000000,
    "scriptPubkey": "OP_DUP OP_HASH160 55368b388c... OP_EQUALVERIFY OP_CHECKSIG" }
]
```

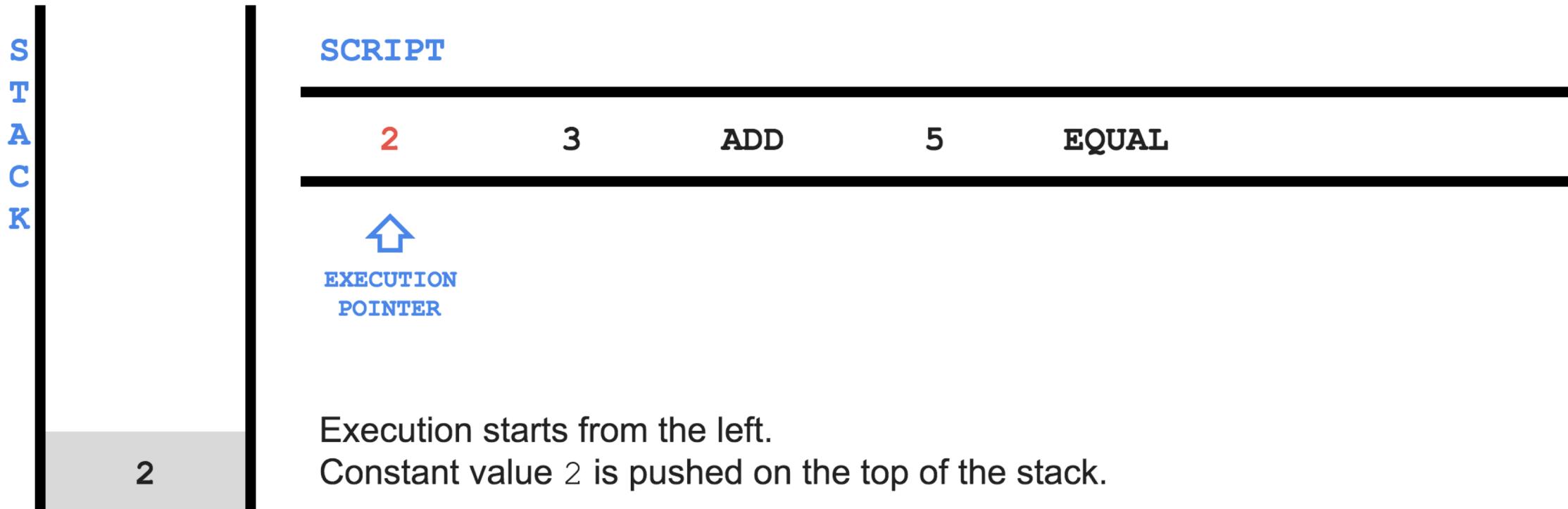
} Header

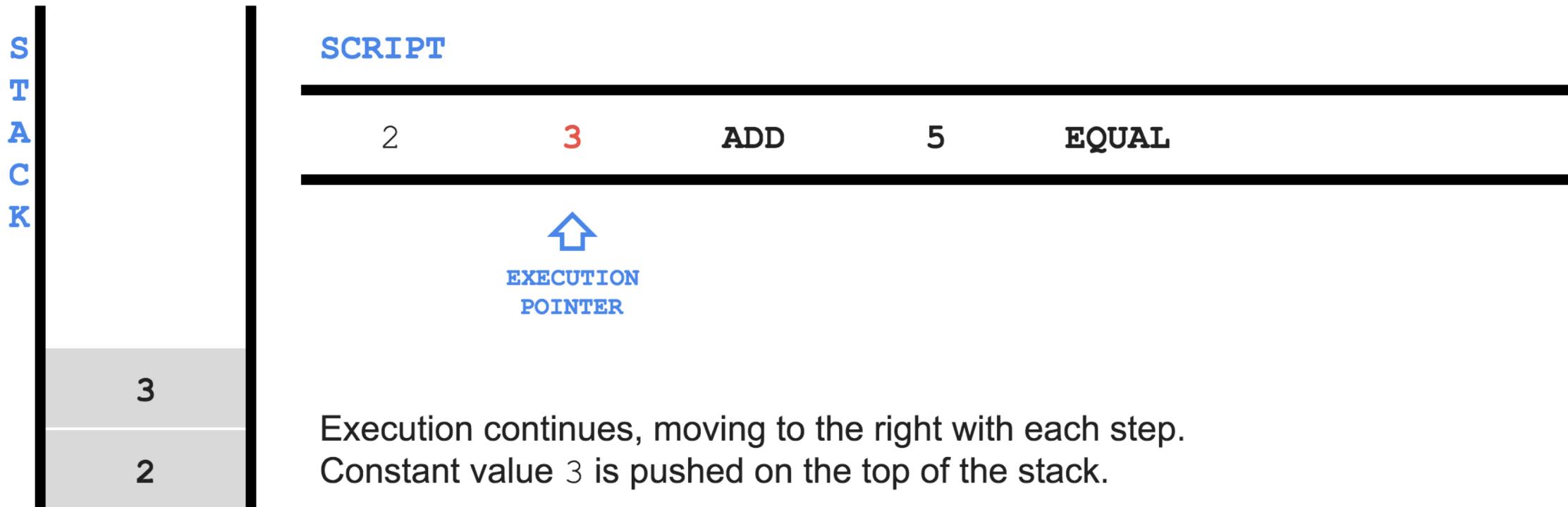
} Inputs

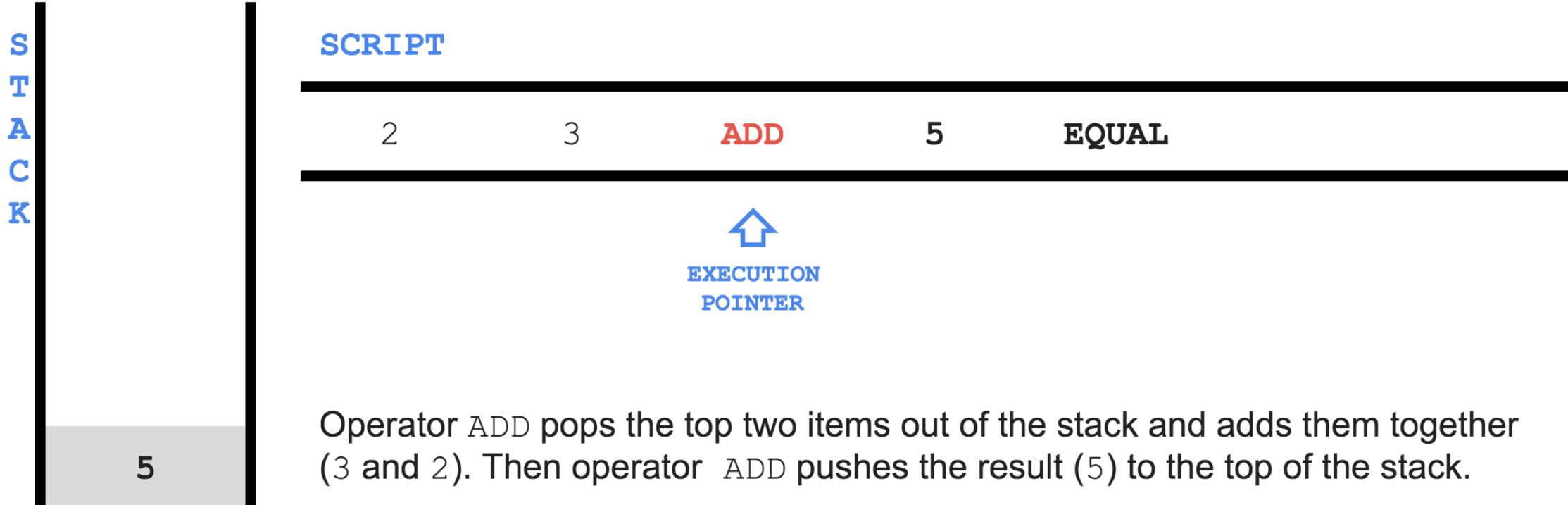
} Outputs

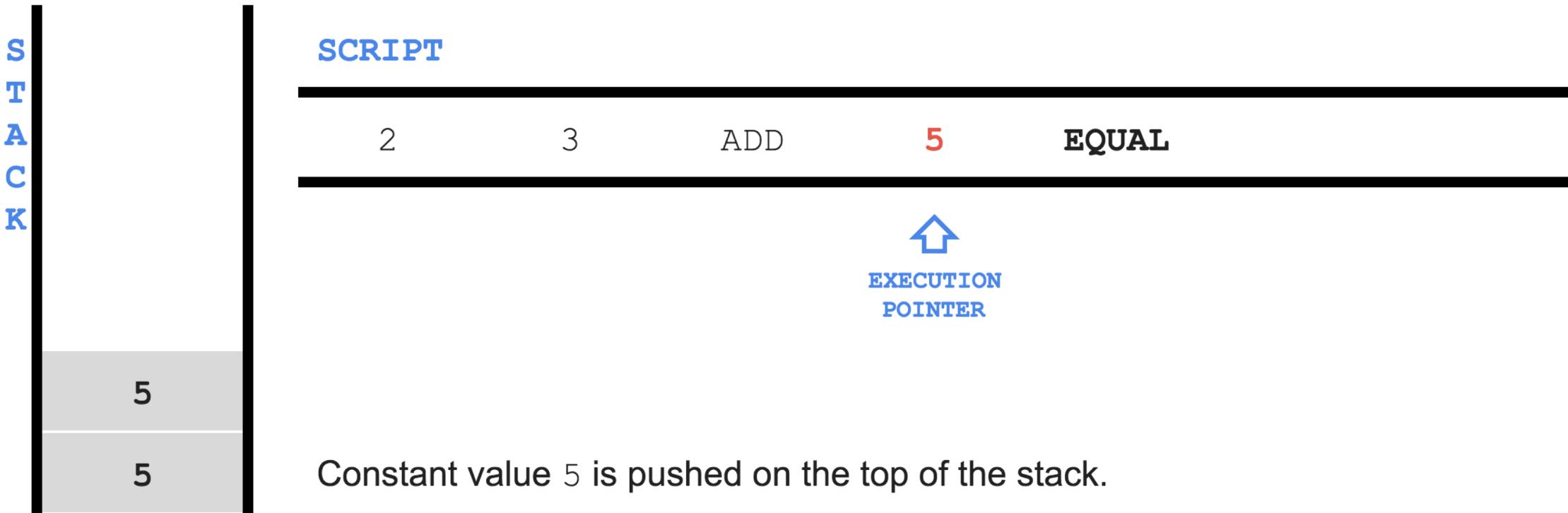
2 3 OP\_ADD 5 OP\_EQUAL

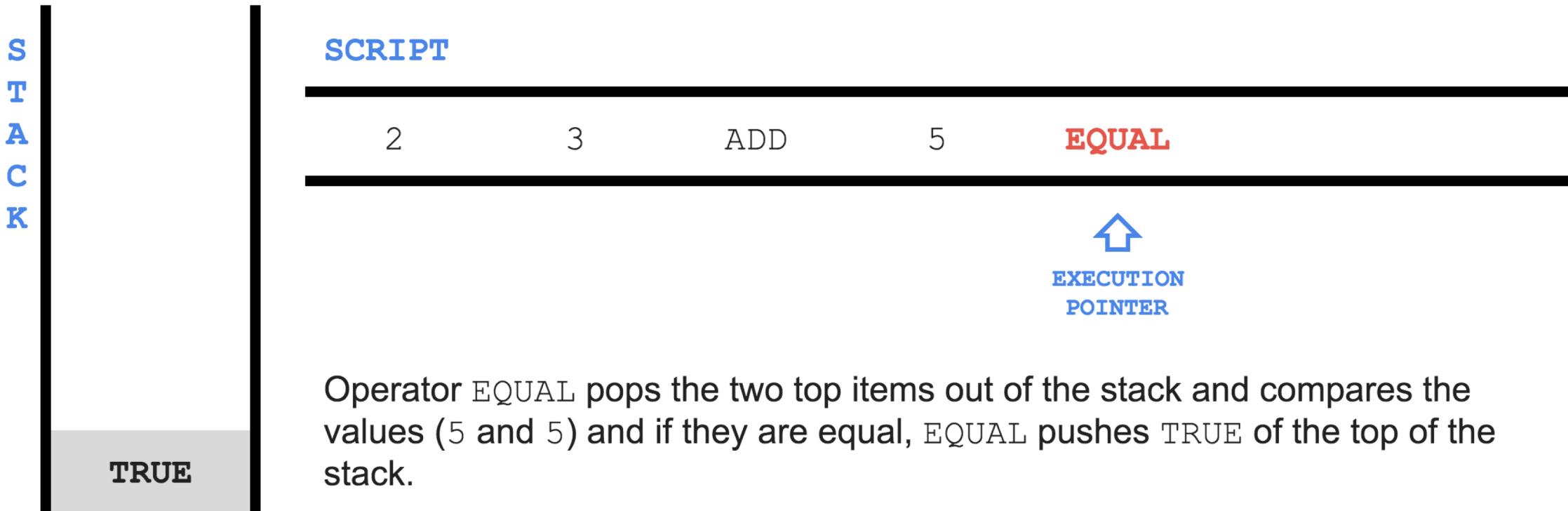
*How Script works*











## 2 3 OP\_ADD 5 OP\_EQUAL



```
{ "hash": "90b18aa54288ec610d83ff1abe90f10d8ca87fb6411a72b2e56a169fdc9b0219",
  "ver": 1,
  "lock_time": 0,
  "size": 226,
  "vin_sz": 1,
  "vout_sz": 2,

  "in": [
    { "prev_out": {
        "hash": "18798f8795ded46c3086f48d5bdabe10e1755524b43912320b81ef547b2f939a",
        "n": 0 },
      "scriptSig": "3045022100c1efcad5cdcc0dcf7c2a79d9e1566523af9c7229c78..." }
  ],

  "out": [
    { "value": 4000000000,
      "scriptPubKey": "OP_DUP OP_HASH160 4b358739fc... OP_EQUALVERIFY OP_CHECKSIG" },
    { "value": 1000000000,
      "scriptPubkey": "OP_DUP OP_HASH160 55368b388c... OP_EQUALVERIFY OP_CHECKSIG" }
  ]
}
```

}

**Header**

}

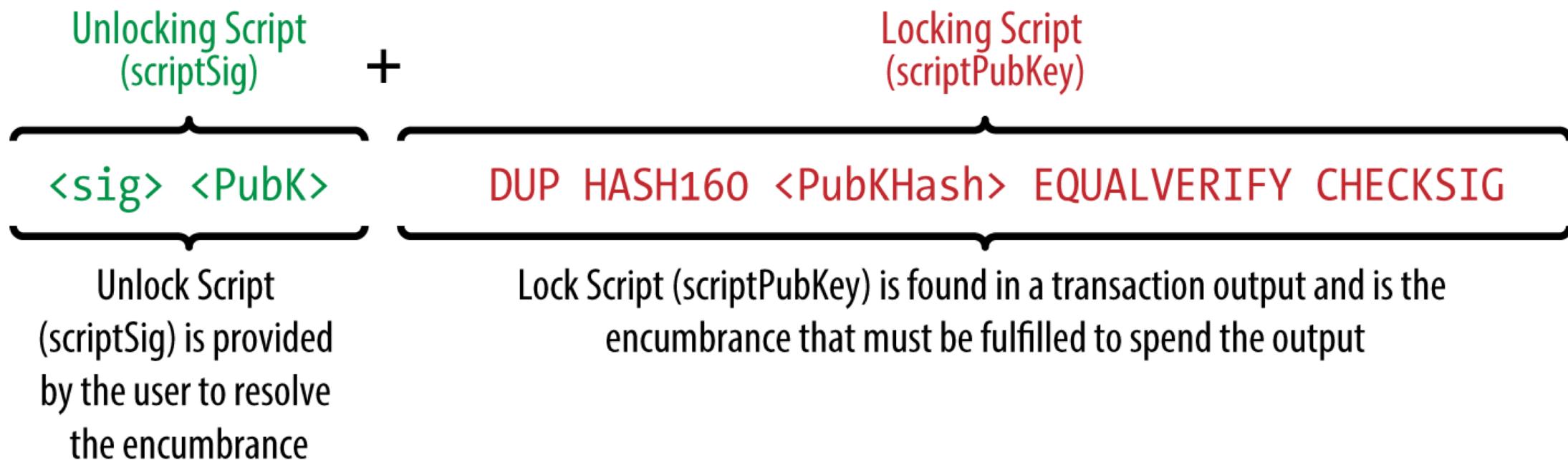
**Inputs**

}

**Outputs**



# P2PKH



S  
T  
A  
C  
K

<sig>

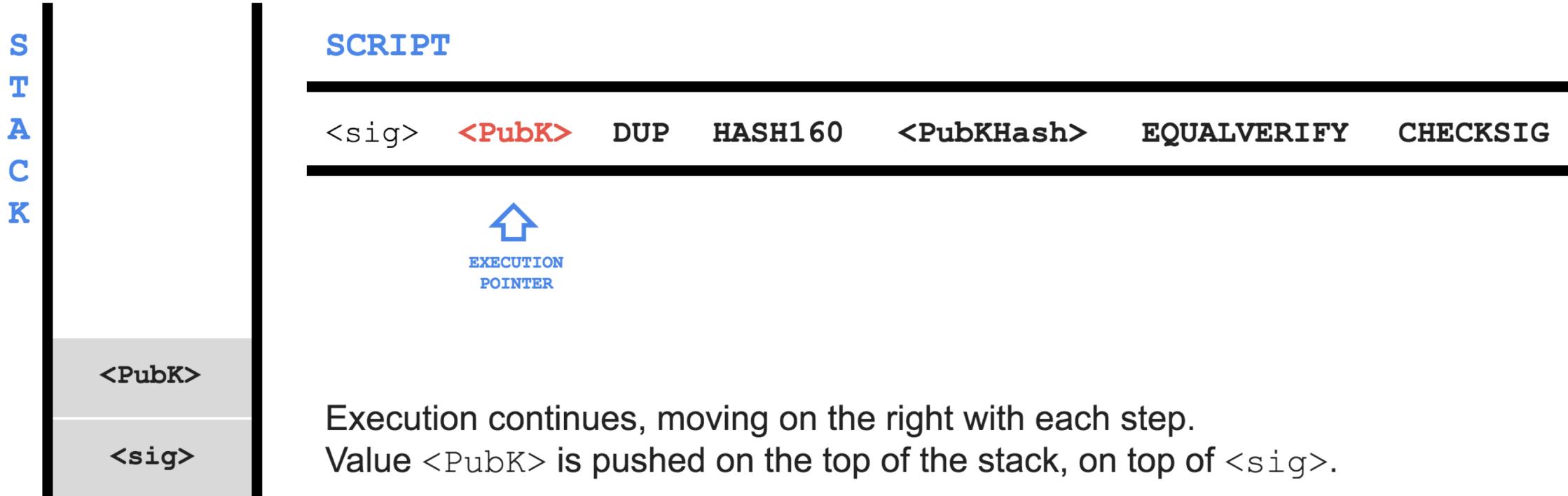
## SCRIPT

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



EXECUTION  
POINTER

Execution starts.  
Value <sig> is pushed to the top of the stack.



S  
T  
A  
C  
K

<PubK>  
<PubK>  
<sig>

## SCRIPT

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



DUP operator duplicates the top item in the stack.  
The resulting value is pushed on the top of the stack.

S  
T  
A  
C  
K

<PubKHash>

<PubK>

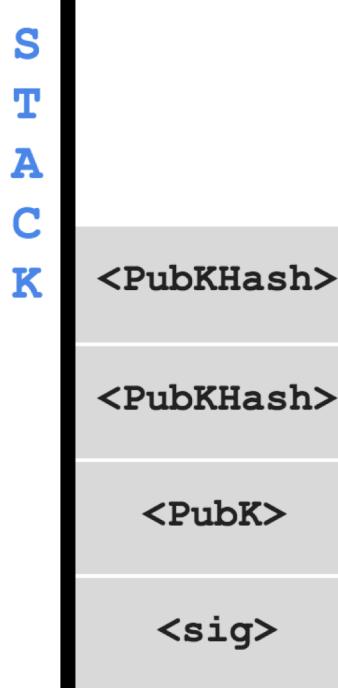
<sig>

## SCRIPT

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



HASH160 operator hashes the top item in the stack with RIPEMD160(SHA256(<PubK>)). The resulting value <PubKHash> is pushed on the top of the stack.



## SCRIPT

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



The value <PubKHash> from the script is pushed on the top of the value <PubKHash> calculated previously from the HASH160 of the value <PubK>.

S  
T  
A  
C  
K

## SCRIPT

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



<PubK>  
<sig>

The EQUALVERIFY operator compares <PubKHash> encumbering the transaction with <PubKHash> calculated from the user's <PubK>. If they match, both are removed and execution continues.

S  
T  
A  
C  
K

TRUE

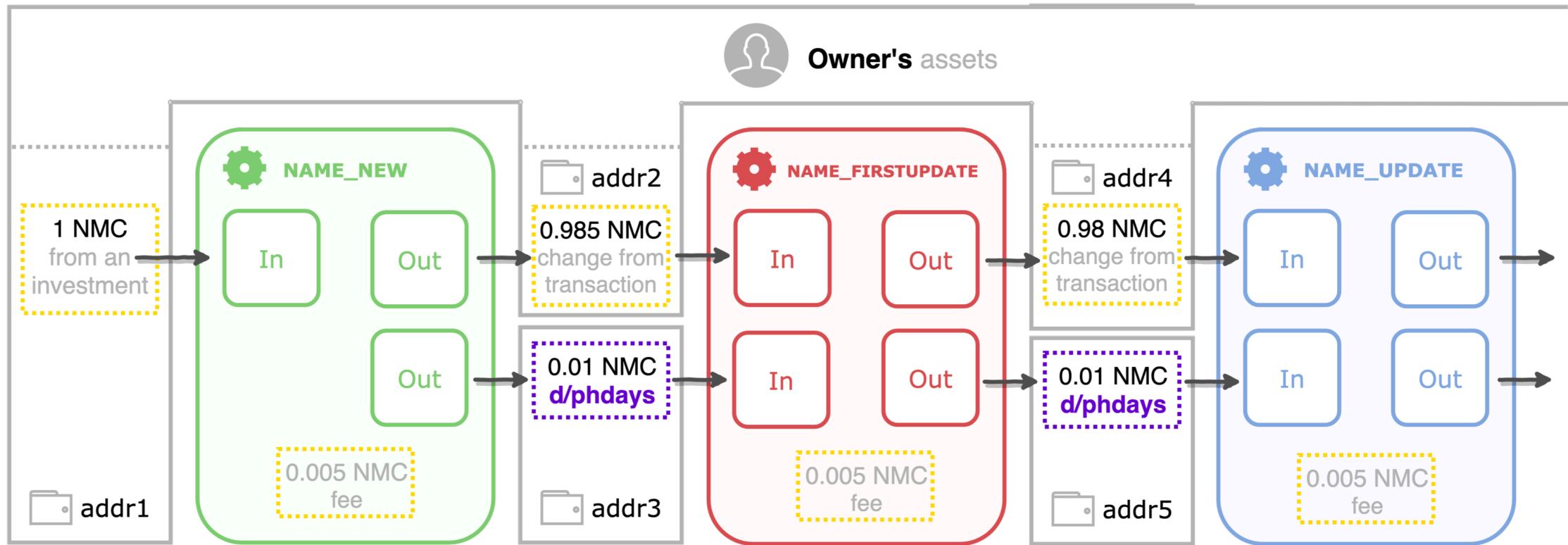
**SCRIPT**<sig> <PubK> DUP HASH160 <PubKHash> EQUALVERIFY **CHECKSIG**EXECUTION  
POINTER

The CHECKSIG operator checks that the signature `<sig>` matches the public key `<PubK>` and pushes TRUE on the top of the stack if true.



# How Namecoin works

# Namecoin flow



# NAME\_NEW



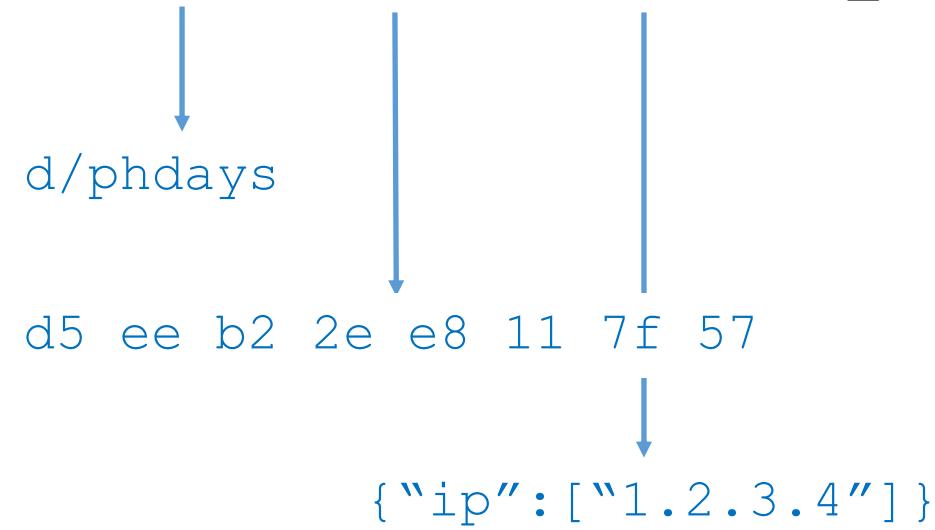
OP\_NAMENEW <20 byte hash> OP\_2DROP <P2PKH>

Pick a random salt	d5 ee b2 2e e8 11 7f 57
Convert d/phdays to ASCII	64 2f 70 68 64 61 79 73
Concatenate the salt with d/phdays in ASCII	d5 ee b2 2e e8 11 7f 57 64 2f 70 68 64 61 79 73
Hash the result with HASH160 (little endian)	75 46 fa a7 ee d7 b4 9f a0 c7 dd 58 1b f2 a8 4f ff a6 74 36

# NAME\_FIRSTUPDATE



OP\_NAME\_FIRSTUPDATE <Name> <Salt> <Value> OP\_2DROP OP\_2DROP <P2PKH>



# NAME\_UPDATE



OP\_NAME\_UPDATE <Name> <Value> OP\_2DROP OP\_DROP <P2PKH>

d/phdays

{"ip": ["1.1.1.1"] }

# Namecoin Economy



- Chhronic C&C: registered in **2016**, still alive

Name d/pationare (pationare.bit)

## Summary

Status	Active
Expires after block	464604 (22089 blocks to go)
Last update	2018-12-07 13:30:11 (block <a href="#">428604</a> )
Registered since	2016-12-25 13:43:35 (block <a href="#">319753</a> )

Current value

```
{"ns": ["a.dnspod.com", "b.dnspod.com"]}
```

## Operations

Date/time	Block	Transaction	Operation	Value
2018-12-07 13:30:11	<a href="#">428604</a>	<a href="#">09381e8d8c...</a>	OP_NAME_UPDATE	{"ns":["a.dnspod.com","b.dnspod.com"]}
2018-05-02 21:18:06	<a href="#">396669</a>	<a href="#">fbe8fedb09...</a>	OP_NAME_UPDATE	{"ns":["ANIRBAN.NS.CLOUDFLARE.COM","REZA.NS.CLOUDFLARE.COM"]}
2018-01-11 20:08:35	<a href="#">379573</a>	<a href="#">6067d7a10e...</a>	OP_NAME_UPDATE	{"ns":["ns1.dnscontrolfff.to","ns2.dnscontrolfff.to"]}

- from **0.0109 NMC** per year
  - 0.00763 USD
  - ~0.5 RUR
- **0.08 USD** per year for daily updates
- compare with **1 USD/year** for **.info**
- or **10 USD/year** for **.com**
- **OPEX instead of CAPEX**



# Threat mining

# Blockchain crawler



# Upstream movement



Collects:

- domain names
- IP addresses
- Namecoin addresses managed by threat actor
- unspent coins (UTXO)

Downstream movement looks similar

Heuristics will be discussed later

```
def upstream_movement(tx):  
    global names  
    global IPs  
    global utxo  
    global known_addresses  
  
    heuristic_result = upstream_heuristic_test(tx)  
  
    if heuristic_result and heuristic_result.guiding_outs:  
        if tx.has_name_op():  
            names.add(tx.name_op.name)  
            for ip_address in tx.name_op.get_ip():  
                IPs.add(ip_address)  
        for guiding_out in heuristic_result.guiding_outs:  
            known_addresses.add(guiding_out.address)  
            tx = namecoin.transactions.find_one({"in.id": guiding_out.id})  
            if tx:  
                upstream_movement(tx)  
            else:  
                utxo.add(guiding_out)
```

# «Common Change»

«If the output of the transaction is only one ordinary coin, then this coin belongs to the person who owns the input coins.

If at the same time there is a special coin at the output, then it also belongs to the person who owns the input coins.

All addresses used in such a transaction are managed by the same person»

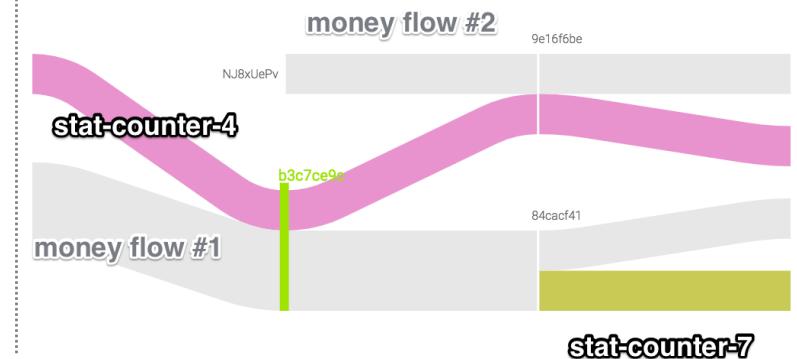
## Case #1: First time used address

- Produced by domain creation and update
- Default behavior of native clients – *namecoind, namecoin-qt*
- The most common transaction pattern in Namecoin



## Case #2: Re-used address

- Corresponds to domain transfer
- In common case doesn't mean that owners are the same, but
- It makes no sense in acquiring a malicious domain, so it is considered as a transfer between the accounts of the same person



# «Common Spending»



«If it is known that at least one of the addresses at the input of the transaction is managed by a certain person, then all other addresses at the inputs of this transaction are managed by the same person.

The coins at these inputs belong to the same person»

- Can be used for *downstream movement* only
- Requires data from *Common Change* heuristic (addresses are managed by threat actor)

```
def common_spending(tx):
    result = { "guiding_ins": [] }

    for input in tx.get_ins():
        if input.address in known_addresses:
            return {"guiding_ins": tx.ins.all}

    return {}
```

# «Known Address»



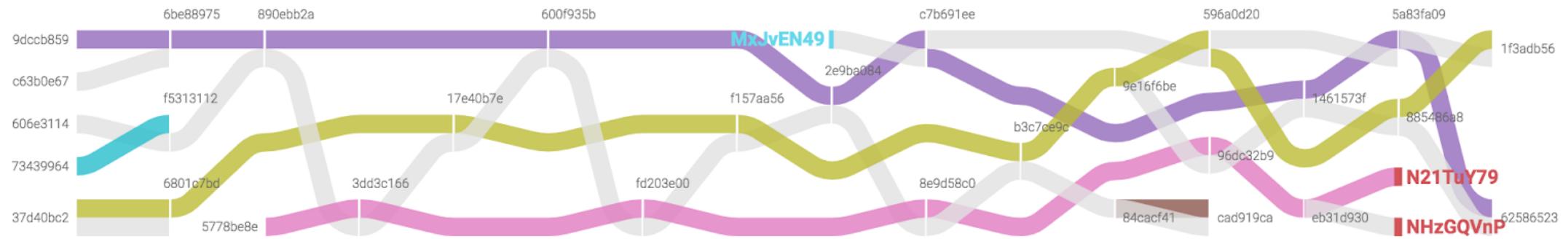
«If it is known that the address at the input (output) of the transaction is managed by a certain person, then the coins that put at this address (spent from this address) belong to the same person»

- Can be used for both upstream and downstream movement
- Also requires addresses from *Common Change*

```
def known_address(tx):
    result = { "guiding_outs": [], "guiding_ins": [] }

    for output in tx.get_outs():
        if output.address in known_addresses:
            result["guiding_outs"].append(output)
    for input in tx.get_ins():
        if input.address in known_addresses:
            result["guiding_ins"].append(input)
    return result
```

# Dangling inputs and outputs





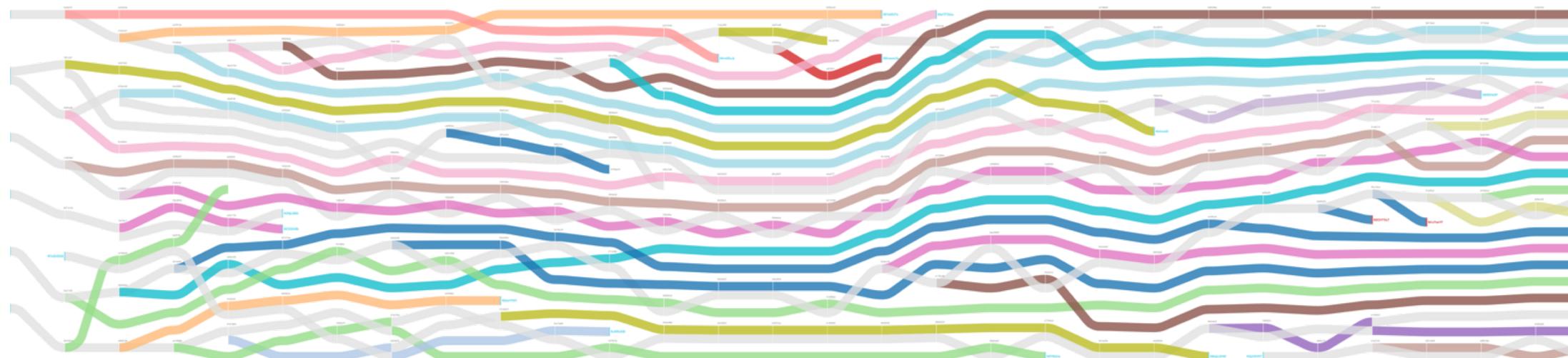
# Takeaways

# Takeaways

⌚ <https://github.com/b4bay/rusty-blockparser>

⌚ [https://github.com/b4bay/threat\\_mining\\_in\\_namecoin](https://github.com/b4bay/threat_mining_in_namecoin)

- RTM, Shifu, Dimnie and GandCrab
- 164 domains
- 277 IP addresses
- 39 UTXO

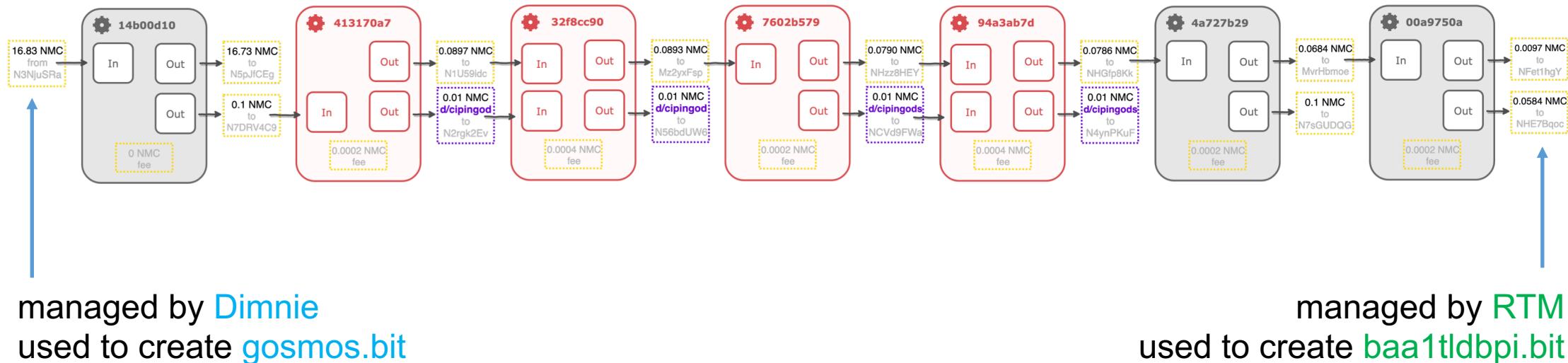




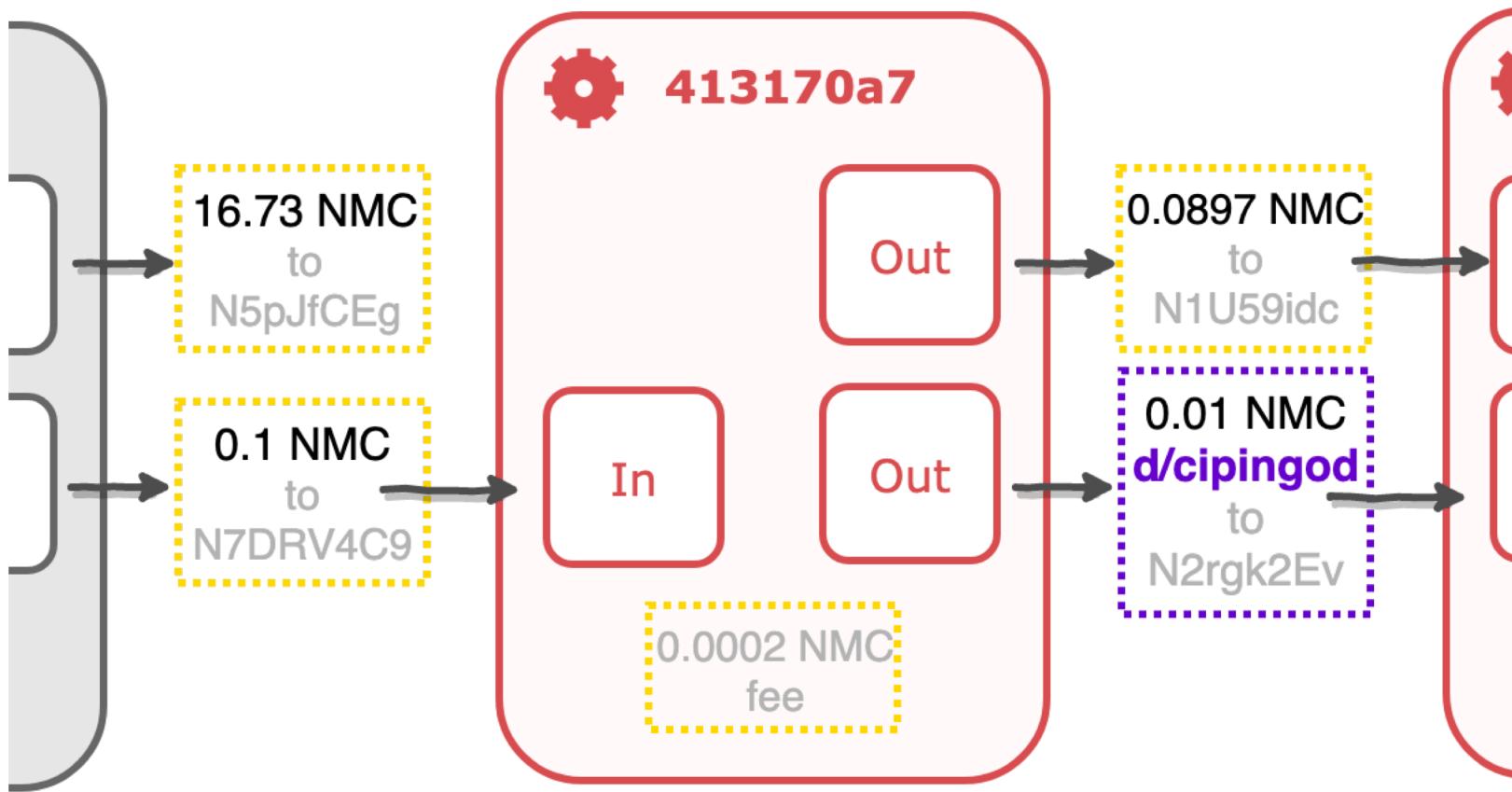
# Bonus track

*May contains assumptions with no strong proof*

# Dimnie ❤️ RTM



# cipingod.bit → 103.208.86.185



# cipingod.bit → 103.208.86.185



RISKIQ

Q 103.208.86.185

Tours Upgrade

First Seen 2017-02-16 ASN Zappie Host LLC  
Last Seen 2019-04-24 Netblock 103.208.86.0/23

Routable Zapple-Host Categorize

HEATMAP

DATA

Resolutions 5 WHOIS 1 Certificate 6 Trackers 0 Components 0 Host Pairs 0 OSINT 4 Hashes 0 Projects 0 Cookies 0

FILTERS DOMAIN (5 / 5)

✓ ✗ mx.orinase.eu	1
✓ ✗ stackoverflow....	1
✓ ✗ uniccc.at	1
✓ ✗ zap.phishfinde...	1
✓ ✗ zap.phishstor...	1

UNIQUE RESOLVE (1 / 5)

✓ ✗ Show Unique ... 5

STATUS

RESOLUTIONS (1 / 5)

Resolve	First	Last	Source	Tags
zap.phishstorm.center	2018-10-03	2019-04-24	pingley, riskiq	
zap.phishfinder.online	2018-05-16	2018-07-19	riskiq	
stackoverflow.party	2017-06-28	2018-02-17	riskiq	
uniccc.at	8-01-01	2018-01-10	pingley, riskiq	
	7-02-16	2017-02-16	riskiq	

Show : 25 ▶ 1-5 of 5 Sort : Last Seen Descending Download Copy

1-5 or 5

uniccc.at

# cipingod.bit → 103.208.86.185



RISKIQ

Q 103.208.86.185

Tours Upgrade

First Seen 2017-02-16 Last Seen 2019-04-24 ASN Zappie Host LLC Netblock 103.208.86.0/23 Routable Zapple-Host Categorize

HEATMAP

DATA

Resolutions 1 WHOIS 6 Certificate 0 Trackers 0 Components 0 Host Pairs 0 OSINT 4 Has 0

FILTERS DOMAIN (5 / 5)

- ✓ x mx.orinase.eu 1
- ✓ x stackoverflow.... 1
- ✓ x uniccc.at 1
- ✓ x zap.phishfinde... 1
- ✓ x zap.phishstor... 1

UNIQUE RESOLVE (1 / 5)

✓ x Show Unique ... 5

STATUS

RESOLUTIONS

Resolve	First	Last
zap.phishstorm.center	2018-10-03	2019-04-24
zap.phishfinder.online	2018-05-16	2018-07-19
stackoverflow.party	2017-06-28	2018-02-17
uniccc.at	8-01-01	2018-01-10
	7-02-16	2017-02-16

«U.S. Arrests 13, Charges 36 in ‘Infraud’ Cybercrime Forum Bust»

// Brian Krebs, 08 Feb 2018

The screenshot shows the UNICC website interface. At the top, there are several navigation icons: HOME, BUY, BILLING, ORDERS, SUPPORT, SETTINGS, and EXIT. Below this is a banner with the text "Hello, HeXi! Balance: 0.00 USD". The main area is titled "Search dumps" and contains a form with fields for Price, Basename, Card type, Level, and Ctype. There are dropdown menus for BINS, Bankname, Country, SVC, and Additional, along with checkboxes for "Fresh only (+\$)", "With T1 Orig", and "Discount dumps". A "Search" button is located at the bottom right of the form. Below the form, a table titled "Search result:" displays a single row of data:

BIN	Type	Code	EXP	Country	Bank	Price
408625	PLATINUM CREDIT VISA	201	02/14	TURKEYKEY	DENIZBANK AS	65.00\$

**cipingod.bit → 103.208.86.185**



The screenshot shows the RISKIQ interface for the IP address 103.208.86.185. The top navigation bar includes the RISKIQ logo, search bar, and tours link. Below the search bar, there's a summary card with first/last seen dates, ASN, and Netblock information. A 'Categorize' button is also present. The main content area has a 'HEATMAP' section and a 'DATA' section with various metrics (5, 1, 6, 0, 0, 0, 4, 0) and tabs for Resolutions, WHOIS, Certificate, Trackers, Components, Host Pairs, OSINT, and Hashes. A 'FILTERS' sidebar lists domain filters like mx.orinase.eu, stackoverflow.com, unicc.at, zap.phishfinder.com, and zap.phishstorm.com. The 'RESOLUTIONS' section displays a table of resolved domains with columns for Resolve, First Seen, and Last Seen. One row for 'unicc.at' is highlighted with a red box.

Resolve	First	Last
zap.phishstorm.center	2018-10-03	2019-04-24
zap.phishfinder.online	2018-05-16	2018-07-19
stackoverflow.party	2017-06-28	2018-02-17
unicc.at	8-01-01	2018-01-10
	7-02-16	2017-02-16

## «U.S. Arrests 13, Charges 36 in ‘Infraud’ Cybercrime Forum Bust»

// Brian Krebs, 08 Feb 2018

The screenshot shows the UNICC website's homepage. At the top left is a large logo with a golden dollar sign and the word "UNICC". Below it are two circular buttons: "HOME" with a credit card icon and "BUY" with a shopping bag icon. The main content area features a large blue header with the text "Thirty-six Defendants In Transnational Criminal Organization" and a sub-header "than \$530 Million in Loss". Below the header is a search form titled "Search dumps" with fields for "price" (two input boxes), "BINs" (dropdown: "(+0.0\$)", "Any"), "Bankname" (dropdown: "Any"), "Country" (dropdown: "Any"), "SVC" (dropdown: "Any"), "Additional" (dropdown: "Any"), and "Action" (dropdown: "Any"). There is also a checkbox group for "Fresh only (+1\$)", "With T1 Orig", and "Discount dumps", and a "Search" button. A sidebar on the right lists various names: Svyat, Ampr, Rolan, Mir, Freed, O, B, R, A, V, J, G, Edg, John, Ran, etc. At the bottom, there is a "Search result:" section with a table showing one row of data: BIN 408625, Type PLATINUM CREDIT VISA, Code 201, EXP 02/14, Country TURKEY/KEY, Bank DENIZBANK AS, and Price 65.00\$.

## «Thirty-six Defendants Indicted for Alleged Roles in Transnational Criminal Organization Responsible for More than \$530 Million in Losses from Cybercrimes»

// The U.S Department of Justice, 07 Feb 2018

- Svyatoslav Bondarko of Ukraine;
  - Amjad Ali aka "Amjad Ali Chaudary," aka "ReduMZ," aka "Amjad Chaudary," 35, of Pakistan;
  - Roland Patrick N'Djimbi Tchikaya aka "Darker," aka "darkgr.evr," 37, of France;
  - Miroslav Kovacevic aka "Goldjunge," 32, of Serbia;
  - Frederick Thomas aka "Mosto," aka "istunna," aka "Bestsss," 37, of Alabama;
  - O
  - B
  - R
  - A
  - V
  - J
  - Gennaro Fioretti aka "DannyLogort," aka "Genny Fioretti," 56, of Italy;
  - Edgar Rojas aka "Edgar Andres Viloria Rojas," aka "Guapo," aka "Guapoo88" aka "Onlyshop," 27, of Australia;
  - John Telusma aka "John Westley Telusma," aka "Petterelhot," aka "Pete," aka "Pette," 33, of Brooklyn, New York;
  - Rami Fawaz aka "Rami Amil Fawaz," aka "Validshop," aka "Th3d," aka "Zatcher," aka "Darkeyes," 26, of Ivory Coast;
  - Muhammad Shiraz aka "Moviestar," aka "Leslie" of Pakistan;
  - Jose Gamboa aka "Jose Gamboas-Soto," aka "Rafael Garcia," aka "Rafael101," aka "Memberplex2006" aka "Knowledge," 29, of Los Angeles, California;
  - Alexey Klimenko aka "Grandhost," 34, of Ukraine;
  - Edward Laviole aka "Eddie Laviole," aka "Skizo," aka "Eddy Laviole," 29, of Canada;
  - Anthony Nnandi Okeakpu aka "Aslikei," aka "Aslike," aka "Moneymafia," aka "Shilonng," 29, of the United Kingdom;
  - Pius Sush Wilson aka "FDIC," aka "TheRealGuru," aka "TheRealGuruNYC," aka "RealGuru," aka "Poison," aka "infection," aka "infected," 31, of Flushing, New York;
  - Muhammad Khan aka "Cool2J," aka "CoolJ," aka "Secureroot," aka "Secureroot," aka "Securerootz," aka

## Andrey Sergeevich Novak aka “Unicc,”