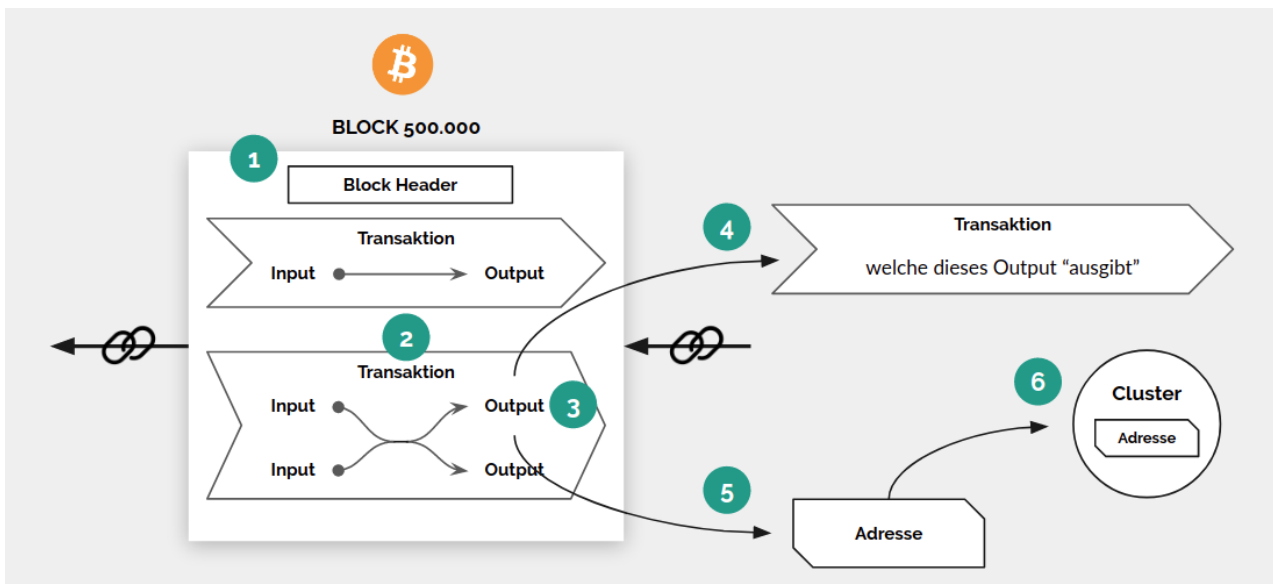


VIRTCRIME Training: BlockSci Demo Part I (Live)

Notwendige Python-Module:

```
In [1]: import blocksci
import matplotlib.pyplot as plt
import matplotlib.ticker
import collections
import pandas as pd
import numpy as np
from datetime import datetime
%matplotlib notebook
```

Warning: You only have 6GB of free disk space left. Running out of disk space may crash the parser and corrupt the BlockSci data files.



- 1 BTC = 100.000.000 sat

Blockchain laden

```
In [2]: # btc = [block 1, block 2, block 3, ...]
btc = blocksci.Blockchain("/mnt/data/blocksci/bitcoin/595303-root-v0.6-0e6e863/config.js
n")
```

1) Block abfragen

```
In [3]: block = btc[500000]
```

```
In [4]: block.tx_count
```

```
Out[4]: 2701
```

```
In [5]: block.fee / 1e8
```

```
Out[5]: 3.39351625
```

```
In [6]: block.hash
```

```
Out[6]: 0000000000000000024fb37364cbf81fd49cc2d51c09c75c35433c3a1945d04
```

2) Transaktion abfragen

```
In [7]: tx = block.txes[66]
```

```
In [8]: tx.hash
```

```
Out[8]: 836e2427557ccea12a7ea2355a3a2005ecbe87bc69228e2efc8decdb05b8df5c
```

```
In [9]: tx.input_value / 1e8
```

```
Out[9]: 47.47421684
```

```
In [10]: tx.output_value / 1e8
```

```
Out[10]: 47.47331684
```

```
In [11]: (tx.input_value - tx.output_value) / 1e8
```

```
Out[11]: 0.0009
```

```
In [12]: tx.fee / 1e8
```

```
Out[12]: 0.0009
```

3) Output abfragen

```
In [13]: output = tx.outputs[1]
```

```
In [14]: output.value / 1e8
```

```
Out[14]: 47.42631123
```

4) Ausgebende Transaktion abfragen

```
In [15]: output.is_spent
```

```
Out[15]: True
```

```
In [16]: spending_tx = output.spending_tx
```

```
In [17]: spending_tx.block_height
```

```
Out[17]: 500009
```

5) Adresse des Outputs aus 3) abfragen

```
In [18]: address = output.address
```

```
In [19]: address.address_string
```

```
Out[19]: '1P9RQEr2XeE3PEb44ZE35sfZRRW1JHU8qx'
```

```
In [20]: address.balance() / 1e8
```

```
Out[20]: 0.02374505
```

6) Cluster abfragen

```
In [21]: clustering = blocksci.cluster.ClusterManager("/mnt/data/blocksci/c_python_btc2__", btc)
```

```
In [22]: cluster = clustering.cluster_with_address(address)
```

```
In [23]: len(cluster)
```

```
Out[23]: 6
```

```
In [24]: cluster.addresses.to_list()
```

```
Out[24]: [PubkeyHashAddress(1F9mradkMQZnBD3XNf9A582ePtagR2GLwT),  
PubkeyHashAddress(1P9RQEr2XeE3PEb44ZE35sfZRRW1JHU8qx),  
PubkeyHashAddress(18g7ZUaxZZKgn05Jcw1pTpWjgnpiHm5vF4),  
PubkeyHashAddress(1LV5y3NkVkmdWnF6xRCEXrAnUkRgge4KSq),  
PubkeyHashAddress(1E1cRscSSvmg68rKTLQKKjmKJwR6rbFguu),  
PubkeyHashAddress(1HzoL2XBL87Zeszk9Ban1MJBpdiU3Q7Dcn)]
```

```
In [25]: cluster.balance() / 1e8
```

```
Out[25]: 0.20316434
```