VIRTCRIME

universität innsbruck

AIT AUSTRIAN INSTITUTE OF TECHNOLOGY

Blockchain-Daten analysieren mit

BlockSci

"A high-performance tool for blockchain science and exploration"

Martin Plattner, Universität Innsbruck
martin@plattner.me

Agenda

- Warum Blockchain-Analyse?
- Was ist BlockSci?
- Demos
- Fazit

Warum Blockchain-Analyse?

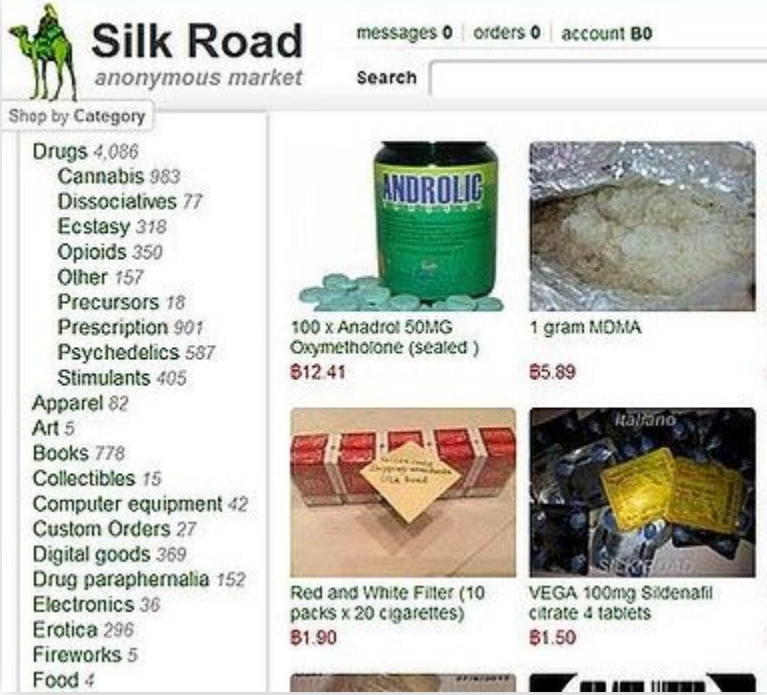
- Vermehrte Nutzung von Blockchain-Systemen
- Relevant für Forensik, Strafverfolgung, Forschung u.a.
- Öffentliche Transaktionshistorie ermöglicht Analysen
  - Bitcoin: über 200 Gigabyte

Warum Blockchain-Analyse?

- Vermehrte Nutzung von Blockchain-Systemen
- Relevant für Forensik, Strafverfolgung, Forschung u.a.
- Öffentliche Transaktionshistorie ermöglicht Analysen
  - Bitcoin: über 200 Gigabyte

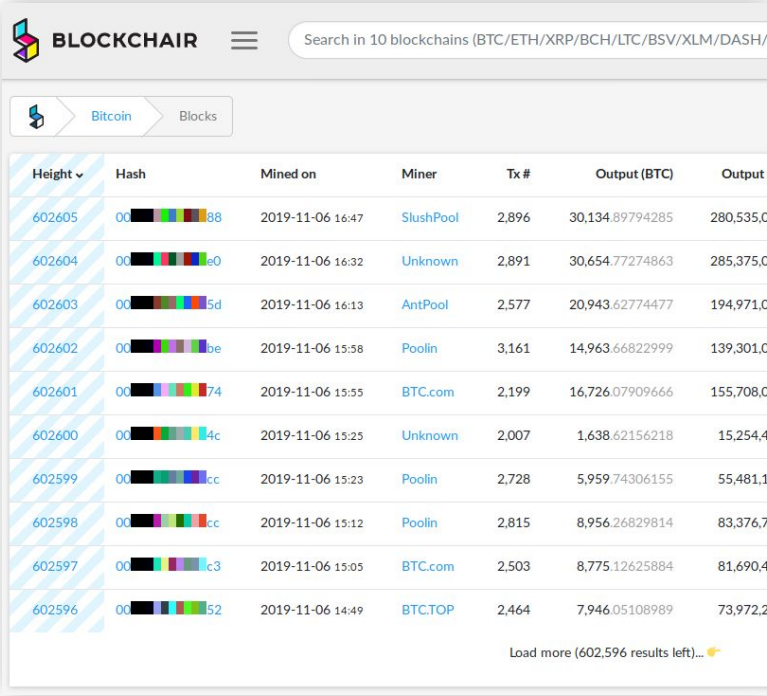
Warum Blockchain-Analyse?

- Vermehrte Nutzung von Blockchain-Systemen
- Relevant für Forensik, Strafverfolgung, Forschung u.a.
- Öffentliche Transaktionshistorie ermöglicht Analysen
  - Bitcoin: über 200 Gigabyte



Warum Blockchain-Analyse?

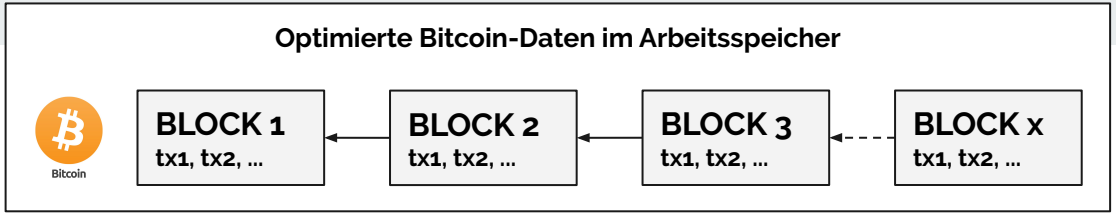
- Vermehrte Nutzung von Blockchain-Systemen
- Relevant für Forensik, Strafverfolgung, Forschung u.a.
- Öffentliche Transaktionshistorie ermöglicht Analysen
  - Bitcoin: über 200 Gigabyte



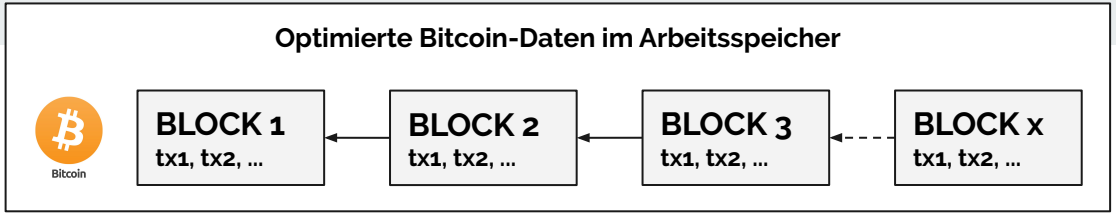
Quelle: blockchair.com

Was ist BlockSci?

- Sehr performantes Blockchain Analysewerkzeug (Open-Source)
  - Geringe Systemvoraussetzungen: 64GB RAM, 500GB SSD
- Erlaubt Zugriff auf Blöcke, Transaktionen, Inputs, Outputs, Adressen etc.
- Unterstützt Bitcoin-Datenstruktur
  - daher auch Bitcoin Cash, Dash, ZCash u.a.
- Abfragen und Analysen über Python-Interface
  - Programmierkenntnisse notwendig



Was ist BlockSci?



Was ist BlockSci?

Training Demo - Jupyter Notebook - Mozilla Firefox

Training Demo - Jupyter

localhost:8888/n 133%

jupyter Training Demo (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
In [1]: import blocksci

In [2]: btc = blocksci.Blockchain("/mnt/data/blocksci/bitcoin/595303-root-v0.6-0

In [3]: address = btc.address_from_string("1BvBMSEYstWetqTFn5Au4m4GFg7xJaNVN2")

In [4]: address.balance() / 1000000000

Out[4]: 0.060332617
```

Demo I

Python: Variablen und Listen

jupyter Python Crashcourse Last Checkpoint: Yesterday at 4:45 PM (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Markdown

Python Crashkurs

In [1]:
```
print("Hallo!")
```

```
Hallo!
```

Variablen

In [2]:
```
name = "Hubert"
print(name)
```

```
Hubert
```

In [3]:
```
a=5
```

In [4]:
```
b=10
```

In [5]:
```
a+b
```

Out[5]:
```
15
```

Listen

In [6]:
```
fruits = ["Orange", "Banane", "Kiwi", "Ananas"]
```

In [7]:
```
fruits[0]
```

Out[7]:
```
'Orange'
```

In [8]:
```
fruits[0:2]
```

Out[8]:
```
['Orange', 'Banane']
```

In [9]:
```
fruits[-1]
```

Out[9]:
```
'Ananas'
```

In [10]:
```
blocks = ["block1", "block2", "block3", "..."]
```

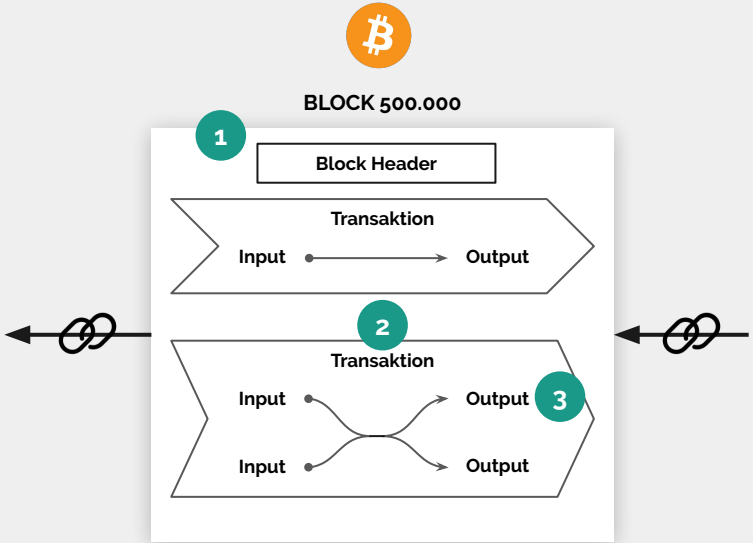
Schleifen

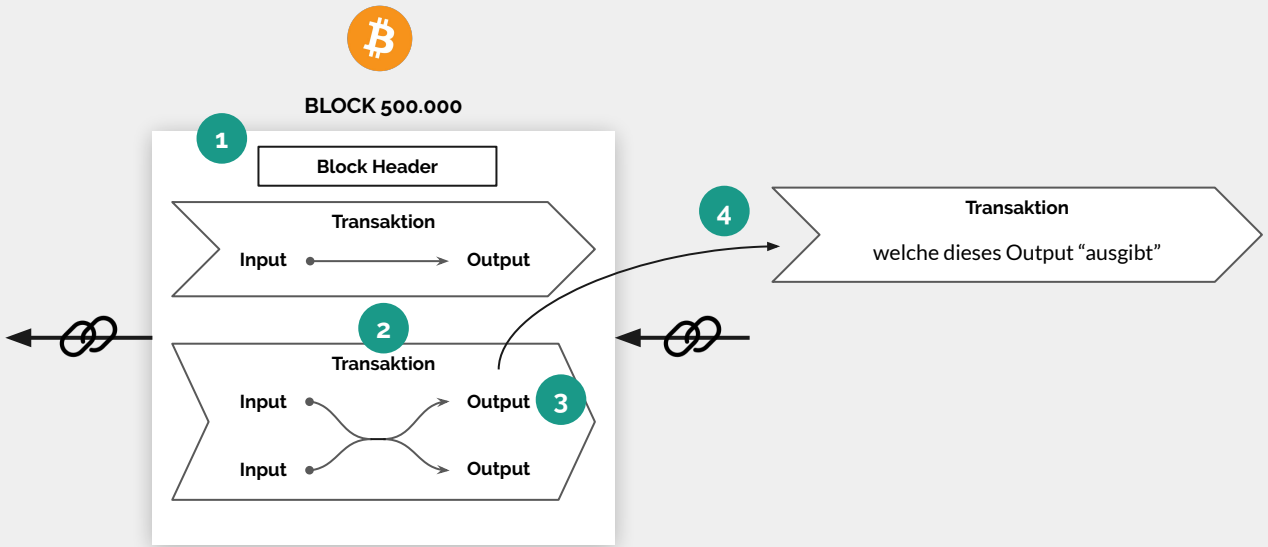
In [11]:
```
for fruit in fruits[-3:]:
    print(fruit)
```

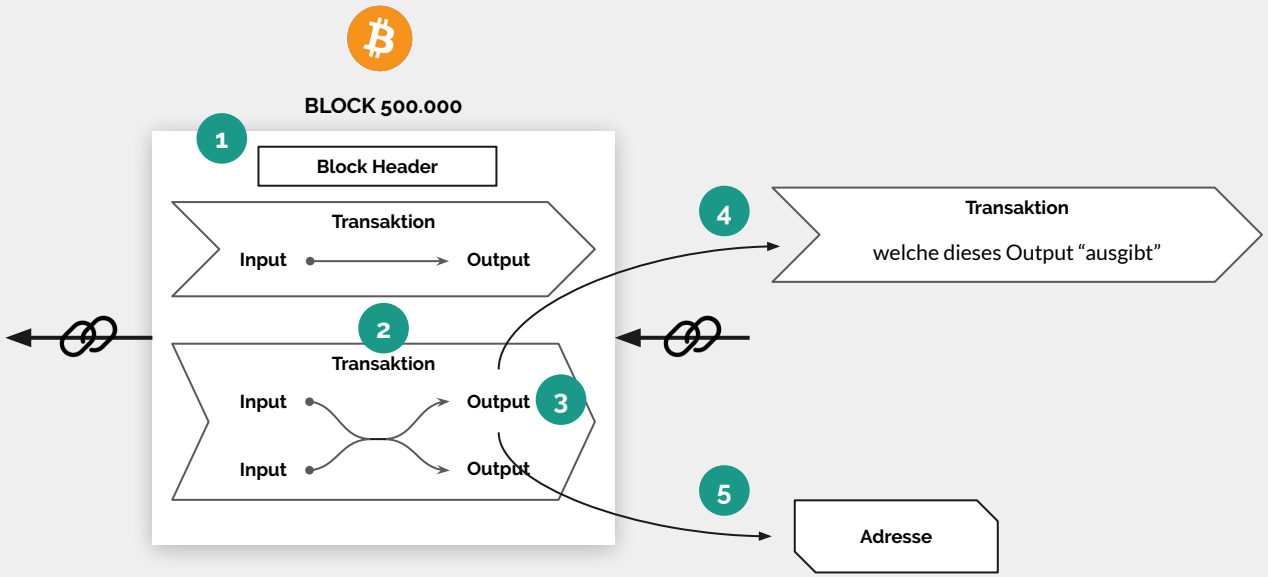
```
Banane
Kiwi
Ananas
```

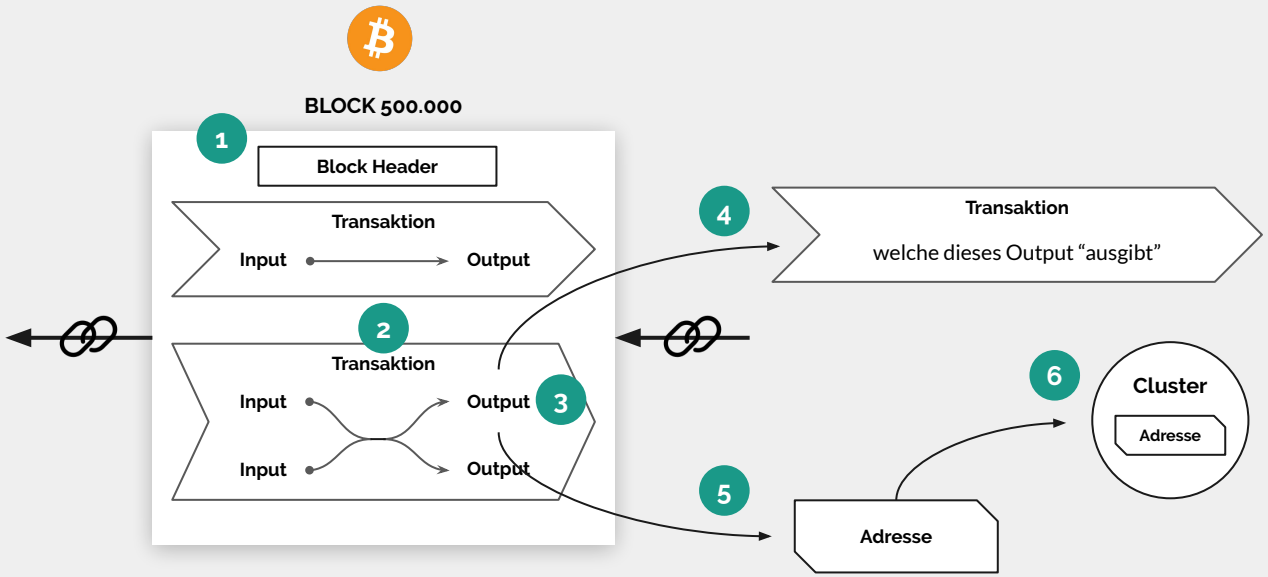
Demo II

Verwendung von BlockSci


BLOCK 500.000
1
Block Header
Transaktion
Input
Output
2
Transaktion
Input
Input
Output
3
Output


BLOCK 500.000
1
Block Header
Transaktion
Input
Output
2
Transaktion
Input
Input
Output
Output
3
4
Transaktion
welche dieses Output “ausgibt”


BLOCK 500.000
1
Block Header
Transaktion
Input
Output
2
Transaktion
Input
Input
Output
Output
3
4
Transaktion
welche dieses Output “ausgibt”
5
Adresse


BLOCK 500.000
1
Block Header
Transaktion
Input
Output
2
Transaktion
Input
Output
3
Input
Output
4
Transaktion
welche dieses Output “ausgibt”
5
Adresse
6
Cluster
Adresse

Fazit

- BlockSci erlaubt hoch-performante Analysen
  - Insbesondere über die gesamte Blockchain, aber auch Einzelfall-Analysen
- Programmierkenntnisse sind notwendig
  - Dafür sehr hohe Flexibilität bei den Analysen
- Ausblick: Erweiterung von BlockSci um Cross-Chain Address Clustering
  - siehe VIRTCRIME Deliverable D.3.2

Vielen Dank

Zeit für Fragen, Anregungen etc.

Martin Plattner
martin@plattner.me

VIRTCRIME

universität innsbruck

AIT AUSTRIAN INSTITUTE OF TECHNOLOGY