Blockchain Engineering

Buchhaltungsmodelle für Kryptowährungen

Dr. Lars Brünjes

GEFÖRDERT VOM
Bundesministerium
für Bildung
und Forschung

MODULARES INNOVATIVES
NETZWERK FÜR DURCHLÄSSIGKEIT

AUFSTIEG DURCH
BILDUNG
OFFENE HOCHSCHULEN

27. September 2019

- Bei Kryptowährungen sind zwei verschiedene Buchhaltungsmodelle üblich, das Konto-Modell (account based model) und das UTxO-Modell (UTxO based model).
- *Ethereum* ist prominentestes Beispiel für eine Kryptowährung, die ein Konto-Modell verwendet.
- *Bitcoin* und *Cardano* benutzen das UTxO-Modell.

- Das Konto-Modell sollte all jenen vertraut sein, die ein Bankkonto besitzen.
- Jeder Addresse (Kontonummer) ist ein Konto zugeordnet,
- Die Adresse eines Benutzers ist sein öffentlicher Schlüssel (oder dessen Hash).
- Eine Transaktion (Überweisung) erniedrigt die Bilanz des Senders und erhöht die Bilanz des Empfängers entsprechend (eventuell abzüglich einer — hoffentlich geringen — Transaktionsgebühr).
- Transaktionen werden mittels digitaler Unterschrift des Senders autorisiert.
- Transaktionen mit mehreren Sendern oder Emfängern sind (zumindest bei Ethereum) nicht möglich.
- Der Status (Zustand) des Systems wird durch die Bilanzen aller Kontos bestimmt.

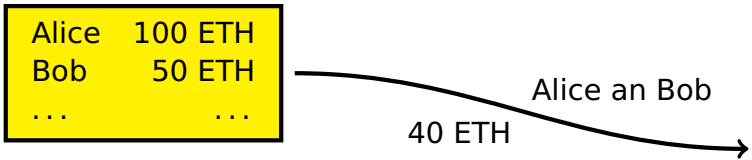
Eine Transaktion im Konto-Modell ist gültig wenn die folgenden drei Bedingungen erfüllt sind:

- Die Transaktion ist mit der digitalen Unterschrift des Senders versehen.
- Die Bilanz des Senders ist mindestens gleich dem überwiesenden Betrag (plus eventuelle Transaktionsgebühren).
- Der überwiesende Betrag ist nicht negativ.

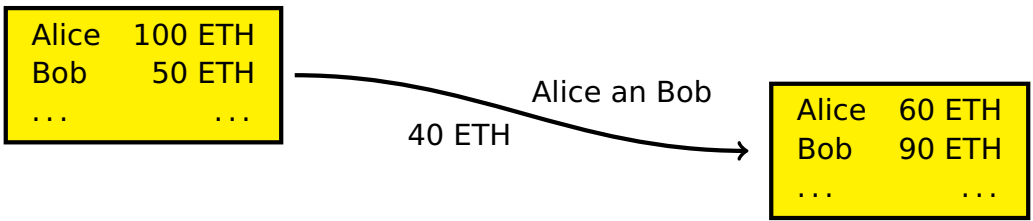
Alice hat 100 ETH auf ihrem Konto und möchte 40 ETH an Bob überweisen, der im Moment 50 ETH auf seinem Konto hat:

Alice	100 ETH
Bob	50 ETH
...	...

Alice hat 100 ETH auf ihrem Konto und möchte 40 ETH an Bob überweisen, der im Moment 50 ETH auf seinem Konto hat:



Alice hat 100 ETH auf ihrem Konto und möchte 40 ETH an Bob überweisen, der im Moment 50 ETH auf seinem Konto hat:



- UTxO steht für unspent transaction output (unausgegebener Output einer Transaktion).
- Wie beim Konto-Modell wird die Adressen eines Benutzers durch seinen öffentlicher Schlüssel gegeben.
- Eine Transaktion hat UTxOs als Inputs und (eventuell mehrere) Outputs.
- Transaktionen werden mittels digitale Unterschrift der Besitzer der Inputs autorisiert.
- Transaktionen mit mehreren Sendern und/oder Emfängern sind möglich.
- Der Status (Zustand) des Systems wird durch die Gesamtheit aller UTxOs bestimmt.

- Für jede Transaktion muss die Summe der Inputs (plus Transaktionsgebühr) gleich der Summe der Outputs sein.
- Eine Transaktion gibt jeden Input vollständig aus. Ist der Input zu groß, muss ein entsprechender Output mit dem “Wechselgeld” erzeugt werden.
- Eine Transaktion besteht aus:
  - Einer Menge von Inputs (UTxOs).
  - Einer geordneten Liste von Outputs, wobei jeder Output eine Adresse und einen Betrag hat.
  - Einer digitalen Unterschrift der Transaktion für jeden Input.

Eine Transaktion im UTxO-Modell ist gültig wenn die folgenden drei Bedingungen erfüllt sind:

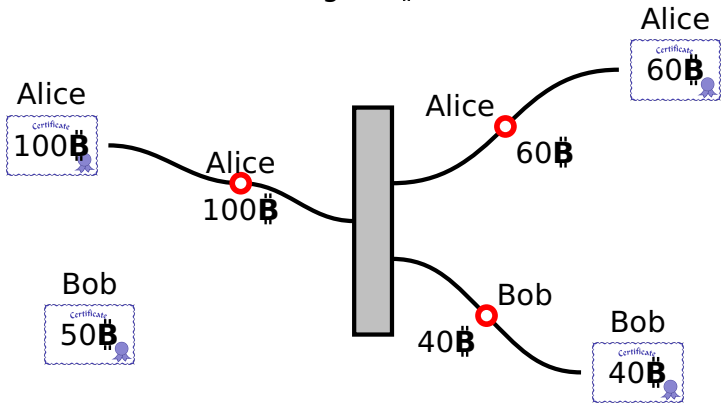
- Die Transaktion ist mit den digitalen Unterschriften der Besitzer aller Inputs versehen.
- Die Summe der Inputs (plus eventuelle Transaktionsgebühren) ist gleich der Summe der Outputs.
- Kein Output-Betrag ist negativ.

Alice hat am Anfang 100 ฿, von denen sie 40 ฿ an Bob überweisen möchte, der am Anfang 50 ฿ hat.

Alice

Bob

Alice hat am Anfang 100 ฿, von denen sie 40 ฿ an Bob überweisen möchte, der am Anfang 50 ฿ hat.



- Jede *Transaktion* hat eine eindeutige Tranaktions-ID (in der Praxis meist der Hash der *serialisierten* Transaktion).
- Ein *Input* ist ein Paar, bestehend aus einer Transaktions-ID und einem Index (der den Transaktions-Output in der Liste aller Outputs identifiziert).
- Ein *Output* ist ein Paar, bestehend aus einer Adresse und einem Betrag.
- Eine Menge von *UTxOs* ist eine Finite Map (Dictionary), deren Schlüssel Inputs und deren Werte Outputs sind.

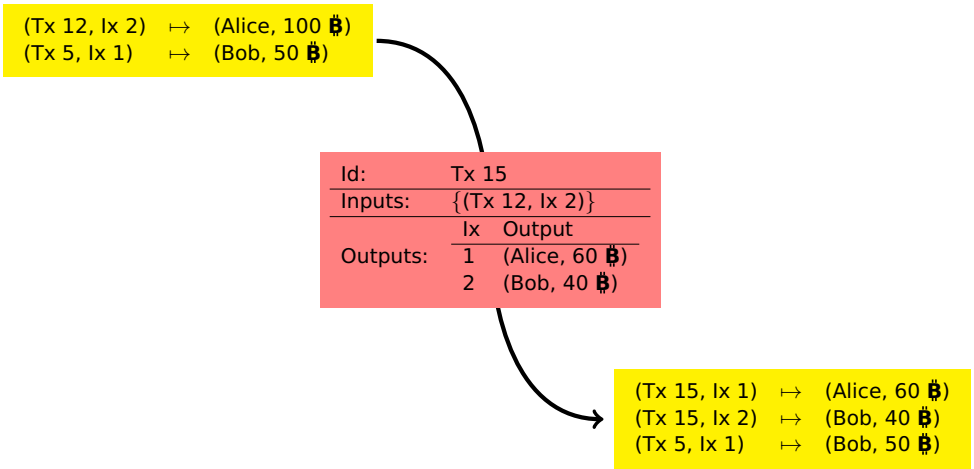
Dieselbe einfache Transaktion — Formal

$$
\begin{aligned}
(\text{Tx } 12, \text{Ix } 2) &\mapsto (\text{Alice}, 100\ ฿) \\
(\text{Tx } 5, \text{Ix } 1) &\mapsto (\text{Bob}, 50\ ฿)
\end{aligned}
$$

Dieselbe einfache Transaktion — Formal

(Tx 12, Ix 2) $\mapsto$ (Alice, 100 ฿)
(Tx 5, Ix 1) $\mapsto$ (Bob, 50 ฿)

Id:	Tx 15	
Inputs:	{(Tx 12, Ix 2)}	
Outputs:	Ix	Output
	1	(Alice, 60 ฿)
	2	(Bob, 40 ฿)


(Tx 12, Ix 2) ↦ (Alice, 100 ฿)
(Tx 5, Ix 1) ↦ (Bob, 50 ฿)
Id: Tx 15
Inputs: {(Tx 12, Ix 2)}
Outputs:
Ix Output
1 (Alice, 60 ฿)
2 (Bob, 40 ฿)
(Tx 15, Ix 1) ↦ (Alice, 60 ฿)
(Tx 15, Ix 2) ↦ (Bob, 40 ฿)
(Tx 5, Ix 1) ↦ (Bob, 50 ฿)

Primitive types

$$
\begin{array}{rcll}
txid & \in & \mathsf{TxId} & \text{transaction id} \\
ix & \in & \mathsf{Ix} & \text{index} \\
addr & \in & \mathsf{Addr} & \text{address} \\
c & \in & \mathsf{Coin} & \text{currency value}
\end{array}
$$

Derived types

$$
\begin{array}{rcll}
tx \in \mathsf{Tx} & = & (inputs, outputs) \in \mathbb{P}(\mathsf{TxIn}) \times (\mathsf{Ix} \mapsto \mathsf{TxOut}) & \text{transaction} \\
txin \in \mathsf{TxIn} & = & (txid, ix) \in \mathsf{TxId} \times \mathsf{Ix} & \text{transaction input} \\
txout \in \mathsf{TxOut} & = & (addr, c) \in \mathsf{Addr} \times \mathsf{Coin} & \text{transaction output} \\
utxo \in \mathsf{UTxO} & = & txin \mapsto txout \in \mathsf{TxIn} \mapsto \mathsf{TxOut} & \text{unspent transaction outputs} \\
b \in \mathsf{Block} & = & tx \in \mathbb{P}(\mathsf{Tx}) & \text{block} \\
pending \in \mathsf{Pending} & = & tx \in \mathbb{P}(\mathsf{Tx}) & \text{pending transactions}
\end{array}
$$

Functions

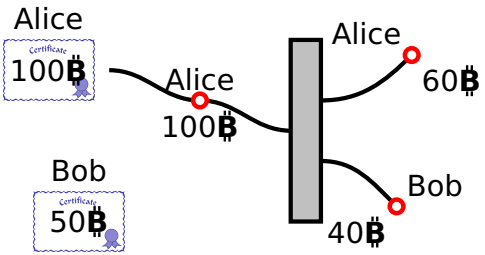
$$
\begin{array}{rcll}
\mathsf{txid} & \in & \mathsf{Tx} \to \mathsf{TxId} & \text{compute transaction id} \\
\mathsf{ours} & \in & \mathsf{Addr} \to \mathbb{B} & \text{addresses that belong to the wallet}
\end{array}
$$

Filtered sets

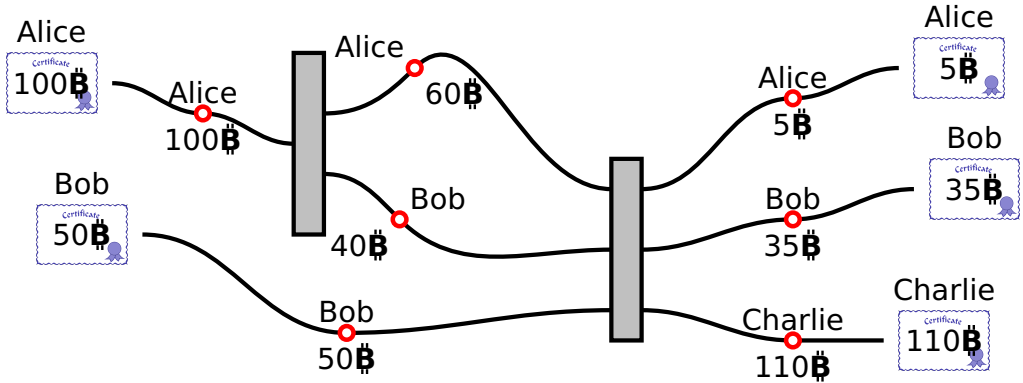
$$
\begin{aligned}
\mathsf{Addr}_{\mathsf{ours}} &= \{a \mid a \in \mathsf{Addr},\ \mathsf{ours}\, a\} \\
\mathsf{TxOut}_{\mathsf{ours}} &= \mathsf{Addr}_{\mathsf{ours}} \times \mathsf{Coin}
\end{aligned}
$$

Abb.: Mathematisches UTxO-Modell

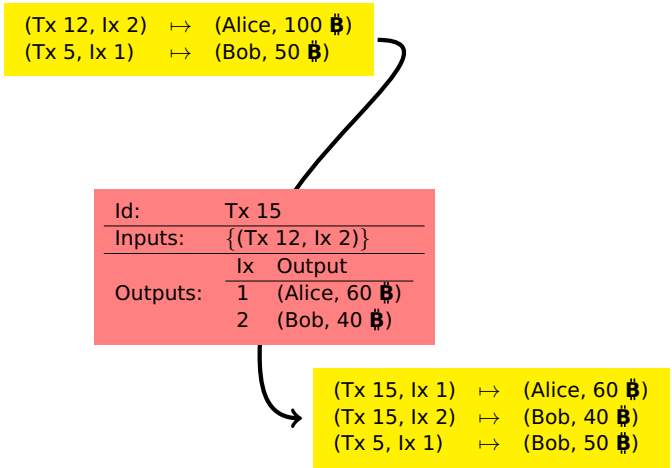
Nachdem Alice Bob 40 ₿ überwiesen hat, möchten sie und Bob Charlie je 55 ₿ überweisen.

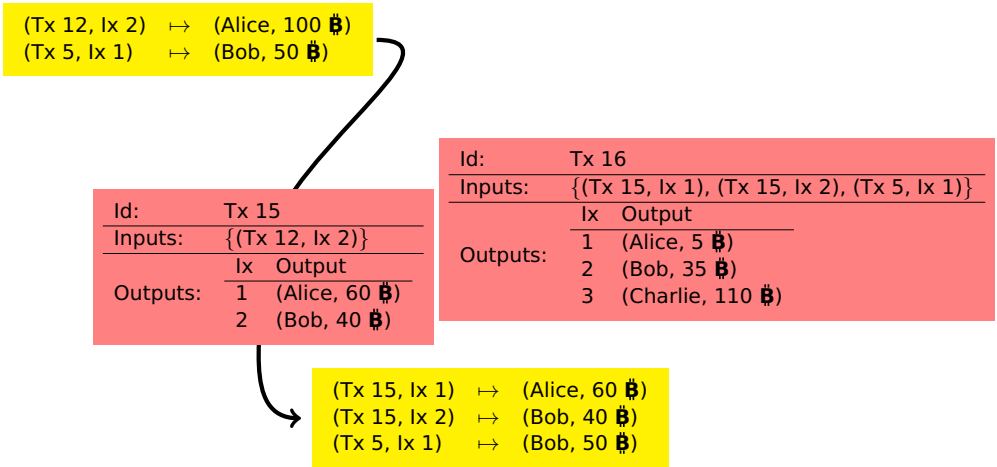


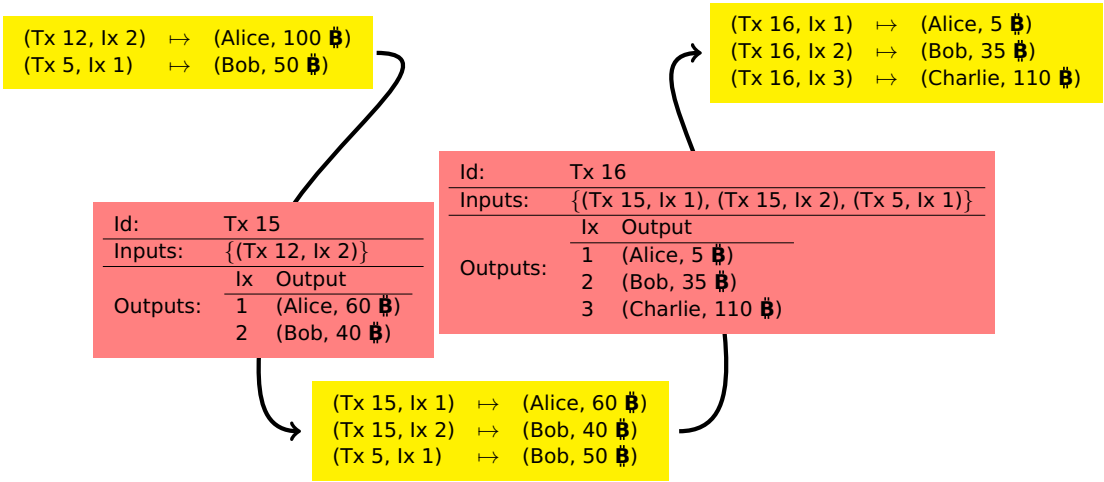
Nachdem Alice Bob 40 ₿ überwiesen hat, möchten sie und Bob Charlie je 55 ₿ überweisen.



Dieselbe komplexe Transaktion — Formal




(Tx 12, Ix 2) ↦ (Alice, 100 ฿)
(Tx 5, Ix 1) ↦ (Bob, 50 ฿)
Id: Tx 15
Inputs: {(Tx 12, Ix 2)}
Outputs:
Ix Output
1 (Alice, 60 ฿)
2 (Bob, 40 ฿)
Id: Tx 16
Inputs: {(Tx 15, Ix 1), (Tx 15, Ix 2), (Tx 5, Ix 1)}
Outputs:
Ix Output
1 (Alice, 5 ฿)
2 (Bob, 35 ฿)
3 (Charlie, 110 ฿)
(Tx 15, Ix 1) ↦ (Alice, 60 ฿)
(Tx 15, Ix 2) ↦ (Bob, 40 ฿)
(Tx 5, Ix 1) ↦ (Bob, 50 ฿)


(Tx 12, Ix 2) ↦ (Alice, 100 ฿)
(Tx 5, Ix 1) ↦ (Bob, 50 ฿)
Id: Tx 15
Inputs: {(Tx 12, Ix 2)}
Outputs:
Ix Output
1 (Alice, 60 ฿)
2 (Bob, 40 ฿)
(Tx 15, Ix 1) ↦ (Alice, 60 ฿)
(Tx 15, Ix 2) ↦ (Bob, 40 ฿)
(Tx 5, Ix 1) ↦ (Bob, 50 ฿)
Id: Tx 16
Inputs: {(Tx 15, Ix 1), (Tx 15, Ix 2), (Tx 5, Ix 1)}
Outputs:
Ix Output
1 (Alice, 5 ฿)
2 (Bob, 35 ฿)
3 (Charlie, 110 ฿)
(Tx 16, Ix 1) ↦ (Alice, 5 ฿)
(Tx 16, Ix 2) ↦ (Bob, 35 ฿)
(Tx 16, Ix 3) ↦ (Charlie, 110 ฿)

- In unserer Einführung von Blockchains hatten wir kurz die Fälschungsmöglichkeit diskutiert, eine Transaktion zu duplizieren.
- Damals hatten wir bemerkt, man müsse Transaktionen unwiederholbar machen, um dies zu verhindern.
- Im Konto-Modell kann dies dadurch erreicht werden, dass Absender ihre Transaktionen numerieren; eine Transaktion ist nur gültig, wenn ihre Nummer größer als alle vorherigen Transaktionen desselben Absenders ist.
- Im UtxO-Modell bekommen wir Unwiederholbarkeit geschenkt: UTxo's, die einmal als Input einer Transaktion benutzt wurden, sind nicht als Input für eine zweite Transaktion verfügbar.
- Dies ist ein klarer Vorteil des UTxO-Modells gegenüber dem Konto-Modell.

- Bisher haben wir als Adresse bei beiden Modellen den *öffentlichen Schlüssel* des Besitzers benutzt.
- In der Praxis wird stattdessen oft der Hash dieses öffentlichen Schlüssels verwendet:
  - Er ist kürzer und daher einfacher zu lesen und platzsparender.
  - Er ist sicherer gegen gewisse zu erwartende technologische Durchbrüche (“Quantencomputer”).
- In diesem Fall müssen die digitalen Unterschriften auch den öffentlichen Schlüssel enthalten, und beim Verifizieren muss geprüft werden, dass dieser öffentliche Schlüssel den erwarteten Hash hat.

- Egal ob Konto-Modell oder UTxO-Model — Mit den gegebenen Regels nimmt die Gesamtsumme von Geld im System nie zu.

- Egal ob Konto-Modell oder UTxO-Model — Mit den gegebenen Regels nimmt die Gesamtsumme von Geld im System nie zu.
- Wo kommt das Geld ursprünglich her? Wie wird neues Geld “gedruckt”?

- Egal ob Konto-Modell oder UTxO-Model — Mit den gegebenen Regels nimmt die Gesamtsumme von Geld im System nie zu.
- Wo kommt das Geld ursprünglich her? Wie wird neues Geld “gedruckt”?
- Die Antworten auf diese Fragen hängen von der konkreten Kryptowährung ab, die wir betrachten.
- Allen gemeinsam ist aber, dass der erste Block in der Blockchain, der sogenannte Genesis-Block, das ursprünglich vorhandene Geld gewissen Personen oder Organisationen “fest verdrahtet” zuweist.

- Egal ob Konto-Modell oder UTxO-Model — Mit den gegebenen Regels nimmt die Gesamtsumme von Geld im System nie zu.
- Wo kommt das Geld ursprünglich her? Wie wird neues Geld “gedruckt”?
- Die Antworten auf diese Fragen hängen von der konkreten Kryptowährung ab, die wir betrachten.
- Allen gemeinsam ist aber, dass der erste Block in der Blockchain, der sogenannte Genesis-Block, das ursprünglich vorhandene Geld gewissen Personen oder Organisationen “fest verdrahtet” zuweist.
- Außerdem gibt es meist Mechanismen, neues Geld zu “drucken”, etwa bei Bitcoin als Belohnung für die Erzeugung eines Blocks (Coinbase Transaction).

- Sowohl das Konto-Modell als auch das UTxO-Modell erlauben Addressen, die nicht einfach einem öffentlichen Schlüssel zugeordnet sind.
- Statt des geheimen Schlüssels des Besitzers werden solche Adressen von Skripten, sogenannten Smart Contracts kontrolliert.
- Bitcoin, Ethereum und Cardano haben alle solch einen Smart-Contract-Mechanismus (wobei der von Bitcoin sehr eingeschränkt ist).
- Wir werden im zweiten Teil des Kurses ausführlich über Smart Contracts reden.

Hinweis

Diese Publikation wurde im Rahmen des vom Bundesministerium für Bildung und Forschung (BMBF) geförderten Bund- Länder- Wettbewerbs “Aufstieg durch Bildung: offene Hochschulen” erstellt. Die in dieser Publikation dargelegten Ergebnisse und Interpretationen liegen in der alleinigen Verantwortung der Autor/innen.