Transaktionen

Ture ClauSSen, 1531067, ture.claussen@stud.hs-hannover.de und Jannes Neemann, 1530893, jannes.neemann@stud.hs-hannover.de

Fakultät IV, Abteilung Informatik, Hochschule Hannover, Ricklinger Stadtweg 120, 30459 Hannover

Zusammenfassung. Schlüsselwörter:

Inhaltsverzeichnis

1 Einführung 2
2 Struktur und technische Umsetzung einer Transaktion 3
2.1 Komponenten von Transaktionen 3
2.2 Typen von Transaktionen 3
2.3 Serialisierung 3
3 Aufbau einer Transaktion 3
3.1 Nonce 3
3.2 Gas 3
3.3 Value und Data 4
3.4 Signatur 4
4 Transaktionsabwicklung 5
4.1 Propagation 5
4.2 Speicherung 5
5 Ausblick 5

1 Einführung

Das Wort Transaktion stammt von dem lateinischen Wort *transigere* ab, welches im übertragenden Sinne mit 'durchführen', 'vollführen' oder 'abmachen' (Geschäft) übersetzt werden kann. [1] Dieser Wortsinn besteht auch weiterhin im technischen und wirtschaftlichen Bereich, jedoch gibt es noch spezifischere Abgrenzungen. In der Wirtschaft ist es ein Vorgang bei dem Waren und Forderungen ausgetauscht werden. [3, S. 18 f.] In der Informatik ist es im Zusammenhang mit Datenbanken eine unteilbare, *atomare*, Abfolge von Anweisungen, die einen Übergang von einem konsistenten Zustand in einen Anderen beschreibt. [4, S.520]

Ethereum ist ein "transaktionsbasierter Automat"(*transaction-based state machine*). Somit sind Transaktionen ein grundlegender Baustein von Ethereum im Allgemeinen und ihnen kommt eine ähnliche Bedeutung wie ACID Transaktionen bei. Der Automat speichert seinen Zuständ σ_t in der Blockchain, eine Transaktion T ist Argument der Zugstandsübergangsfunktion Υ, die von *externen Akteuren (EA)* angestoSSen wird und diesen gespeicherten Zustand σ_t in einen neuen, gültigen Zustand σ_{t+1} überführen soll: $\sigma_{t+1} = \Upsilon(T, \sigma_t)$. Im Falle eines Konsens des Netzwerkes wird diese Zustandsveränderung durchgeführt beziehungsweise gespeichert.

Im Kontrast zu Kryptowährungen wie Bitcoin ist der Umfang des Automaten bzw. des Protokolls bei Ethereum deutlich geweitet, denn Zweck ist nicht nur die Schöpfung, Speicherung und der Austausch eines digitalen Zahlungsmittels [6], sondern eine allgemeine dezentrale Rechenmaschine, ein "Weltcomputer". Daher bestehen bei Ethereum auch an Transaktionen andere technische und konzeptionelle Anforderungen, die im Folgenden erläutert werden. [7, S. 1-4]

2 Struktur und technische Umsetzung einer Transaktion

Die Komponenten welche eine Transaktion in Ethereum ausmachen, sind vergleichbar mit denen eines Briefes. Sie besitzen einen Empfänger sowie eine Frankierung, welche die Transportkosten zum Empfänger bezahlen. AuSSerdem besitzt eine Transaktion Ether oder Nutzdaten zu verschicken, wie es bei einem Brief ebenfalls möglich ist. Ether wären dabei ein Geldschein und Nutzdaten ein Text oder eine Karte. Im folgenden sollen die einzelnen Felder die eine Transaktion ausmachen vorgestellt werden.

2.1 Komponenten von Transaktionen

Allgemein enthält eine Transaktion, wie sie im offiziell [7, S. 4] definiert ist, folgende Felder:

nonce: Ein Skalar welcher gleich der Anzahl der vom EOA versendeten Transaktionen ist. Der Nutzen wird in 3.1 erläutert.

gasPrice: Ein Skalar der angibt, wie viel Wei man pro Einheit *Gas* bezahlt, die bei der Gesamtheit aller Berechnungen die während der Ausführung der Transaktion anfallen (S. 3.2)

gasLimit: Ein Skalar der die maximal Anzahl an *Gas* angibt, die während der Ausführung der Transaktion verbraucht werden darf. Dieser Betrag muss im Voraus bezahlt werden.

to: Die 160-Bit Adresse des Empfängers.

value: Skalar der die Menge Wei angibt, die der Empfänger erhält.

v,r,s: Komponenten der ECDSA-Signatur (S. 3.4), um den Sender der Transaktion zu bestimmen

init: Contract Creation

data: Ein Byte-Array unbegrenzter Länge, welches die Nutzdaten des Kontrakts enthält

2.2 Typen von Transaktionen

2.3 Serialisierung

3 Aufbau einer Transaktion

3.1 Nonce

3.2 Gas

Gas ist ein zentraler konzeptioneller Lösungsansatz im Rahmen von Ethereum. Da Ethereum turing-vollständig ist [7, S. 1], ergibt sich unter anderem das sogenannte "Halteproblem". Dieses besagt, dass im Voraus nicht vorhergesagt werden kann, ob das Programm einer Turing-Maschine jemals zu einem Ende kommt. [2, S.70] Um die Funktionalität des Netzwerks zu gewährleisten, wird die Laufzeit

einer jeden Zustandsveränderung der Blockchain, sprich Transaktion, durch Gas begrenzt.

Gas ist eine eigenständige Währung innerhalb von Ethereum, dessen Einheit einen Rechenschritt in der EVM bemisst [5, S. 9:3], wobei für jeden Opcode die Kosten in Gas spezifiziert werden. [7, S. 25 ff.] Gas ist also eine Gebühr für Rechnenaufwand. Zusätzlich werden auch Kosten für die Nutzung von persistentem Speicher miteinbezogen. Es gilt sogar das Inverse: Wird durch eine Transaktion persistenter Speicher freigegeben, werden Rabatte gewährt.

Die maximale Gebühr einer Transaktion wird durch die Kombination der Datenfelder *gasPrice* und *gasLimit* angegeben. Die resultierende Gebühr *gasPrice*× *gasLimit* wird bei Erstellung der Transaktion in voller Höhe vom Konto abgezogen. Nach Bestätigung der Transaktion wird nicht genutztes Gas zu dem angegebenen Preis in Ethereum zurückerstattet.

Somit gilt es im Voraus abzuschätzen wie hoch der Rechenaufwand sein wird. Je mehr Ressourcen des Weltcomputers in Anspruch genommen werden, desto höher die Gebühr. Gerade wegen des Halteproblems kann dies aber nur grob vorgenommen werden, eine robuste Programmierung von *Smart Contracts* ist essentiell. Ein erster Anhaltepunkt dafür sind zunächst die intrinsischen Kosten einer Transaktion. Das ist der Overhead der allein durch die Transaktion und deren Inhalt betsteht. Diese intrinsischen Kosten g_0 lassen sich mit auf Basis folgender Grundlage berechnen.

$$g_0 \equiv \sum_{i \in T_i, T_d} \begin{cases} G_{datazero} \text{ if } i = 0 \\ G_{txdatanonzero} \text{ otherwise} \end{cases} + \begin{cases} G_{txcreate} \text{ if } T_t = \emptyset \\ 0 \text{ otherwise} \end{cases} + G_{transaction}$$

Also steigen die Kosten einer Transaktion mit der GröSSe des Feldes *data* an und $G_{transaction}$ bestimmt den Basiswert an Gas für eine Transaktion, welcher sich im Jahr 2020 auf 21000 beläuft. Generell sollte das *gasLimit* tendenziell zu hoch angelegt sein, da Transaktionen mit unzureichendem Gas einfach abgebrochen werden *(out-of-gas Exception)*. In diesem Fall wird keine der begonnenen Veränderungen am Zustand gespeichert.

Preis und Latenz Gas kann bewusst nur mit Ether erworben werden, da die Gas-Preise möglichst unabhändig von den Preisschwankungen (von Ether) sein sollen. Der *gasPrice* kann frei gesetzt werden, auch ein Wert von 0 ist gültig. Ein Richtwert für den Wert lässt sich durch Werkzeuge wie ETH Gas Station ermitteln, welche vergangene Transaktionen im *Ledger* betrachten und daraus Richtwerte ermitteln. Dort wird auch ein Umstand kenntlich, denn die Höhe des Gas-Preises scheint maSSgeblich über die Latenz zu entscheiden. [7, S. 7]

Anreiz und Spieletheorie

3.3 Value und Data

3.4 Signatur

ECDSA

Multisignaturen

4 Transaktionsabwicklung

4.1 Propagation

4.2 Speicherung

5 Ausblick

Literatur

1. Transigere - Translation from Latin into German | PONS. https://en.pons.com/translate/latin-german/transigere
2. Davis, M.: Computability and Unsolvability. Courier Corporation (Apr 2013)
3. Ehrlicher, W.: Kompendium der Volkswirtschaftslehre. Vandenhoeck & Ruprecht (1975)
4. Herold, H., Lurz, B., Wohlrab, J., Hopf, M.: Grundlagen Der Informatik. Pearson, third edn. (2017)
5. M.Spain, M.Foley: OASIcs-Tokeneconomics. Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany (2019)
6. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System
7. Wood, G.: Ethereum/yellowpaper. 2019-10-20 (Oct 2019)