(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) **EP 3 617 978 A1**

(12) EUROPÄISCHE PATENTANMELDUNG

(43) Veröffentlichungstag:
04.03.2020 Patentblatt 2020/10

(51) Int Cl.:
G06Q 30/00 *(2012.01)*

(21) Anmeldenummer: **18191983.8**

(22) Anmeldetag: **31.08.2018**

(84) Benannte Vertragsstaaten:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR
Benannte Erstreckungsstaaten:
BA ME
Benannte Validierungsstaaten:
KH MA MD TN

(71) Anmelder: **Siemens Aktiengesellschaft**
80333 München (DE)

(72) Erfinder: **Falk, Rainer**
85586 Poing (DE)

Bemerkungen:
Geänderte Patentansprüche gemäss Regel 137(2) EPÜ.

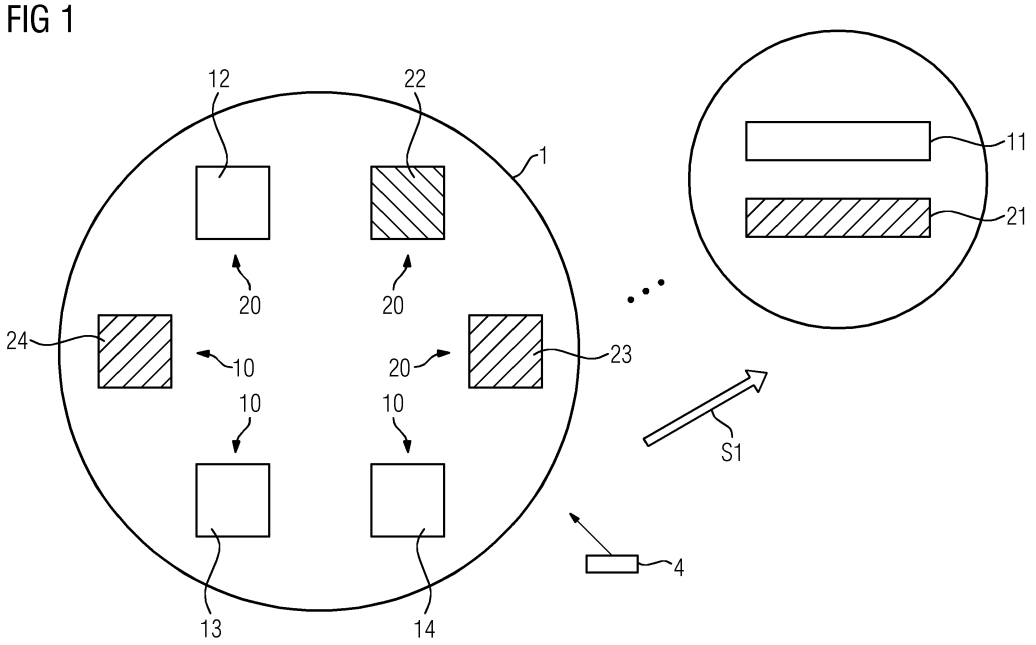
(54) VERTEILTES DATENBANKSYSTEM MIT MEHREREN DATENBANKINSTANZEN UND VERFAHREN ZUM BETREIBEN DESSELBEN

(57) Vorgeschlagen wird ein verteiltes Datenbanksystem (1) mit mehreren Datenbankinstanzen (10, 20). Eine jeweilige der mehreren Datenbankinstanzen (10, 20) ist durch eine Anzahl Knoteneinrichtungen (12, 13, 14; 22, 23, 24) gebildet ist, die dazu eingerichtet sind, gemeinsam konsensbasiert eines von mehreren Transaktionsbüchern (11, 21) des verteilten Datenbanksystems (1) zu verwalten. Das verteilte Datenbanksystem (1) ist eingerichtet, eine zu bestätigende Transaktion (4) durch Aufnehmen in das Transaktionsbuch (11, 21) einer oder mehrerer der mehreren Datenbankinstanzen (10, 20) zu bestätigen. Hierbei ist das verteilte Datenbanksystem (1) dazu eingerichtet ist, zu entscheiden, durch welche der mehreren Datenbankinstanzen (10, 20) die zu bestätigende Transaktion (4) bestätigt wird.

Das Anwendungsdesign kann vereinfacht werden, und es kann eine verbesserte automatisierte Verteilung einer Transaktionslast des verteilten Datenbanksystems über die mehreren Datenbankinstanzen erfolgen.

Weiterhin werden ein Verfahren und ein Computerprogrammprodukt zum Betreiben eines verteilten Datenbanksystems mit mehreren Datenbankinstanzen vorgeschlagen.

FIG 1



EP 3 617 978 A1

Beschreibung

[0001] Die vorliegende Erfindung betrifft das Gebiet der verteilten Datenbanksysteme und spezieller ein verteiltes Datenbanksystem mit mehreren Datenbankinstanzen.

[0002] In einem, etwa mit Blockketten-Technologie implementierten, verteilten Datenbanksystem können Transaktionen ohne Clearing-Stelle oder besonderes Vertrauensverhältnis zwischen den Transaktionspartnern oder einer Clearing-Stelle basierend auf einem Konsens zwischen den Transaktionspartnern transparent und manipulationsgeschützt abgewickelt werden. Ein Transaktionsdatensatz kann Programmcode umfassen oder referenzieren, der beim Bestätigen der Transaktion in dem verteilten Datenbanksystem ausgeführt wird (sog. "Smart Contract"). Ein derartiges verteiltes Datenbanksystem eignet sich als transparente, manipulationsgeschützte IT-Infrastrukturplattform zur Steuerung eines industriellen Automatisierungssystems.

[0003] Bei der Konfiguration einer Konsensregel eines solchen verteilten Datenbanksystems besteht eine Tradeoff-Situation zwischen Parametern wie Transaktionsdurchsatz und Stärke des Manipulationsschutzes. Bekannt sind daher verteilte Datenbanksysteme mit mehreren unterschiedlich konfigurierten Datenbankinstanzen, die auch Haupt- und Side-Chains genannt werden. Hierbei obliegt es einem Anwender bzw. einer zu dem Datenbanksystem externen Entität, zu entscheiden, welcher der Datenbankinstanzen eine jeweilige zu bestätigende Transaktion zum Bestätigen bereitgestellt werden soll. Diese Entscheidung fällt daher für eine jeweilige Art von Transaktion in der Regel bereits in der Designphase der jeweiligen Anwendung, wie etwa des industriellen Automatisierungssystems.

[0004] Vor diesem Hintergrund besteht eine Aufgabe der vorliegenden Erfindung darin, ein verbessertes verteiltes Datenbanksystem mit mehreren Datenbankinstanzen bereitzustellen.

[0005] Demgemäß wird ein verteiltes Datenbanksystem mit mehreren Datenbankinstanzen vorgeschlagen. Eine jeweilige der mehreren Datenbankinstanzen ist durch eine Anzahl Knoteneinrichtungen gebildet, die dazu eingerichtet sind, gemeinsam konsensbasiert eines von mehreren Transaktionsbüchern des verteilten Datenbanksystems zu verwalten. Das verteilte Datenbanksystem ist eingerichtet, eine zu bestätigende Transaktion durch Aufnehmen in das Transaktionsbuch einer oder mehrerer der mehreren Datenbankinstanzen zu bestätigen. Hierbei ist das verteilte Datenbanksystem dazu eingerichtet, zu entscheiden, durch welche der mehreren Datenbankinstanzen die zu bestätigende Transaktion bestätigt wird.

[0006] Insbesondere kann die Entscheidung, durch welche der mehreren Datenbankinstanzen die zu bestätigende Transaktion bestätigt wird, allein dem verteilten Datenbanksystem überlassen sein.

[0007] Somit kann vorteilhaft das Anwendungsdesign vereinfacht werden und eine verbesserte und automatisierte Verteilung einer von dem verteilten Datenbanksystem handzuhabenden Transaktionslast über die mehreren Datenbankinstanzen erfolgen.

[0008] Das Transaktionsbuch (engl. "ledger") einer jeweiligen Datenbankinstanz des verteilten Datenbanksystems kann als Kette oder Pfad von bestätigten Blöcken repräsentiert werden. Insbesondere umfasst ein jeweiliger Block eine Anzahl in dem Transaktionsbuch bestätigter Transaktionen. Die Verkettung kann mittels Verkettungsprüfsummen gebildet sein. Eine jeweilige der Anzahl Knoteneinrichtungen der Datenbankinstanz kann eine Kette von bestätigten Blöcken speichern, die eine Konsensversion des Transaktionsbuchs der Datenbankinstanz repräsentiert. Insbesondere kann die jeweilige Datenbankinstanz eine Blockkette bzw. eine Blockchain, wie etwa eine Haupt-Chain oder eine Side-Chain sein.

[0009] Eine Anzahl bedeutet vorliegend eine Anzahl von eins oder mehr.

[0010] Eine jeweilige Datenbankinstanz ist insbesondere eine verteilte Datenbankinstanz.

[0011] Unter "das Datenbanksystem ist dazu eingerichtet" ist insbesondere zu verstehen, dass das Datenbanksystem ein Mittel aufweisen kann, das dazu eingerichtet ist, die jeweilige Funktionalität zu implementieren. Ein jeweiliges Mittel kann ein separat bzw. zentral bereitgestelltes Mittel wie eine separate Einrichtung oder Einheit sein. Alternativ hierzu kann das jeweilige Mittel dezentral bzw. verteilt implementiert sein. Insbesondere kann das jeweilige Mittel durch Funktionalität einer oder mehrerer der mehreren Datenbankinstanzen implementiert sein. Spezieller kann das jeweilige Mittel durch Funktionalität einer jeweiligen Knoteneinrichtung der mehreren Anzahlen Knoteneinrichtungen der mehreren Datenbankinstanzen implementiert sein.

[0012] Eine bevorzugte Funktionsweise einer jeweiligen Datenbankinstanz wird kurz erläutert. Die Anzahl Knoteneinrichtungen einer jeweiligen Datenbankinstanz können das Transaktionsbuch der Datenbankinstanz insbesondere auf folgende Weise gemeinsam und konsensbasiert verwalten:

[0013] In der jeweiligen Anzahl Knoteneinrichtungen kann eine Kopie bzw. Repräsentation des Transaktionsbuchs der jeweiligen Datenbankinstanz gespeichert sein. Das Transaktionsbuch umfasst eine Kette von bestätigten Transaktionen. Insbesondere umfasst das Transaktionsbuch eine Kette von bestätigten Blöcken, wobei jeder bestätigte Block eine Anzahl von bestätigten Transaktionen umfasst.

[0014] Wird der Datenbankinstanz eine zu bestätigende Transaktion bereitgestellt, kann das Bestätigen der zu bestätigenden Transaktion in dem Transaktionsbuch der Datenbankinstanz insbesondere wie folgt erfolgen:

[0015] Eine blockbildende Knoteneinrichtung der Anzahl Knoteneinrichtungen der Datenbankinstanz kann die zu bestätigende Transaktion in einen von ihr gebildeten Block aufnehmen. Dabei kann die blockbildende Knoteneinrichtung die zu bestätigende Transaktion prü-

fen. Das Prüfen kann insbesondere das Ausführen eines in der Transaktion umfassten Programmcodes oder Smart Contracts umfassen, welcher die Transaktion beschreibt. Bei erfolgreicher Prüfung kann die blockbildende Knoteneinrichtung den gebildeten Block: mit einer Datenblockprüfsumme gegen Manipulationen absichern; mit einer Verkettungsprüfsumme mit dem letzten bestätigten Block der in der blockbildenden Knoteneinrichtung gespeicherten Repräsentation des Transaktionsbuchs verketten; mit einem der Konsensregel entsprechenden Nachweiswert, wie einem Proof-of-Work, Proof-of-Stake versehen; und den gebildeten Block als unbestätigten Block in der Datenbankinstanz bereitstellen.

[0016] Zum Erstellen des Nachweiswerts kann es erforderlich sein, eine vorgegebene Menge an Ressourcen, wie Rechenzeit, Speicherplatz oder Kryptotoken, aufzuwenden oder vorzuhalten. Dies kann dem Schutz gegen nachträgliche Manipulationen dienen.

[0017] Daraufhin können eine Mehrzahl und bevorzugt alle der Anzahl Knoteneinrichtungen den in der Datenbank bereitgestellten unbestätigten Block prüfen. Insbesondere kann geprüft werden, ob der Nachweiswert der Konsensregel entspricht, ob der unbestätigte Block mit der in der prüfenden Knoteneinrichtung gespeicherten Repräsentation des Transaktionsbuchs verkettbar ist, ob die Datenblockprüfsumme korrekt ist und ob die in dem unbestätigten Block enthaltenen zu bestätigenden Transaktionen gültig sind, was auf gleiche Weise wie durch die blockbildende Einrichtung beim Bilden des unbestätigten Blocks geprüft werden kann. Bei erfolgreicher Prüfung kann die jeweilige prüfende Knoteneinrichtung den unbestätigten Block an die in ihr gespeicherte Repräsentation des Transaktionsbuchs anfügen.

[0018] Die von den Knoteneinrichtungen der Datenbankinstanz implementierte und bei dem jeweiligen Prüfen berücksichtigte Konsensregel kann derart eingerichtet sein, dass trotz der Abwesenheit einer Clearing-Stelle und trotz dessen, dass nicht notwendigerweise alle der Knoteneinrichtungen des verteilten Datenbanksystems direkt miteinander kommunizieren und/oder einander vertrauen, sich ein Mehrheitskonsens dergestalt ausbilden kann, dass in der Mehrzahl und bevorzugt allen der Knoteneinrichtungen die gleiche Repräsentation des Transaktionsbuchs der Datenbankinstanz gespeichert ist, die im Vorliegenden als "Konsensversion des Transaktionsbuchs" oder der Einfachheit halber als "das Transaktionsbuch" bezeichnet wird.

[0019] Unter "Bereitstellen in der Datenbankinstanz" im Hinblick auf eine unbestätigte bzw. zu bestätigende Transaktion und/oder einen unbestätigten Block ist insbesondere zu verstehen, dass die unbestätigte Transaktion bzw. der unbestätigte Block an mindestens eine der Knoteneinrichtungen der Datenbankinstanz übermittelt wird. An die übrigen der Knoteneinrichtungen derselben Datenbankinstanz kann die bereitgestellte zu bestätigende Transaktion bzw. der bereitgestellte unbestätigte Block direkt, indirekt oder auf Peer-to-Peer-Weise übermittelt werden.

[0020] Analog dazu ist unter "Bereitstellen an das verteilte Datenbanksystem" bzw. "dem verteilten Datenbanksystem Bereitstellen" im Hinblick auf eine unbestätigte bzw. zu bestätigende Transaktion insbesondere zu verstehen, dass die unbestätigte bzw. zu bestätigende Transaktion an mindestens eine der Knoteneinrichtungen einer der Datenbankinstanzen des verteilten Datenbanksystems übermittelt wird, von wo aus sie direkt, indirekt oder auf Peer-to-Peer-Weise an die weitere der Knoteneinrichtungen weiterer der Datenbankinstanzen übermittelt werden kann und vorzugsweise alle der Knoteneinrichtungen aller der Datenbankinstanzen übermittelt werden kann.

[0021] Die zu bestätigende Transaktion kann insbesondere eine dem verteilten Datenbanksystem bereitgestellte unbestätigte Transaktion sein.

[0022] Vorschlagsgemäß trifft das verteilte Datenbanksystem die Entscheidung, durch welche der mehreren Datenbankinstanzen die zu bestätigende Transaktion bestätigt wird. Anders ausgedrückt entscheidet das verteilte Datenbanksystem, in welches eine und/oder in welche mehrere der mehreren Transaktionsbücher des verteilten Datenbanksystems die zu bestätigende Transaktion als bestätigte Transaktion aufgenommen wird.

[0023] Die Entscheidung kann zentral oder verteilt bzw. konsensbasiert erfolgen. Die Entscheidung kann explizit erfolgen, bevor die zu bestätigende Transaktion basierend auf der Entscheidung an die eine oder die mehreren der mehreren Datenbankinstanzen zum Bestätigen bereitgestellt wird. Die Entscheidung kann implizit erfolgen. Hierbei kann die zu bestätigende Transaktion einer Vielzahl und bevorzugt allen der mehreren Datenbankinstanzen bereitgestellt werden, und die Entscheidung kann implizit mit dem Bestätigen bzw. als Folge des Bestätigens der zu bestätigenden Transaktion durch die eine oder die mehreren der Datenbankinstanzen erfolgen, wie nachstehend noch näher erläutert wird.

[0024] Gemäß einer Ausführungsform ist das verteilte Datenbanksystem dazu eingerichtet, in Abhängigkeit von einem Zustand der jeweiligen Datenbankinstanz zu entscheiden, durch welche der mehreren Datenbankinstanzen die zu bestätigende Transaktion bestätigt wird.

[0025] Unter "Zustand der jeweiligen Datenbankinstanz" ist insbesondere ein automatisiert messbarer oder anderweitig bestimmbarer Zustand zu verstehen, der sich auf den Betrieb des verteilten Datenbanksystems bzw. einer oder mehrerer der mehreren Datenbankinstanzen auswirkt.

[0026] Das verteilte Datenbanksystem kann beispielsweise eingerichtet sein, den Zustand abhängig von einer Anzahl von Parametern zu ermitteln wie: Anzahl der in einem vorgegebenen vergangenen Zeitraum pro Zeiteinheit in einer jeweiligen Datenbankinstanz bestätigten Transaktionen; aktuelle Speichergröße des Transaktionsbuchs einer jeweiligen Datenbankinstanz; aktuelle oder zeitlich gemittelte Transaktions- und/oder Rechenlast der jeweiligen Datenbankinstanz und dergleichen.

[0027] Die Entscheidung anhand des Zustands kann

dabei derart getroffen werden, dass ein möglichst schnelles Bestätigen, eine möglichst günstige Lastverteilung über die Datenbankinstanzen, eine möglichst günstige Verteilung des Speicherbedarfs der jeweiligen Datenbankinstanz und dergleichen erzielt werden können. Auch diese Entscheidung kann explizit vorab oder implizit im Laufe oder als Ergebnis des Bestätigens getroffen werden.

[0028] Auf diese Weise kann vorteilhaft eine automatisierte Verteilung der Transaktionslast in dem verteilten Datenbanksystem realisiert werden.

[0029] Gemäß einer weiteren Ausführungsform ist das verteilte Datenbanksystem dazu eingerichtet, für jede der Datenbankinstanzen eine Rechenlast zu ermitteln und zu entscheiden, dass die zu bestätigende Transaktion durch die eine oder die mehreren der Datenbankinstanzen mit der geringsten Rechenlast bestätigt wird.

[0030] Die Last einer jeweiligen Datenbankinstanz kann eine mittlere Last oder eine Spitzenlast der Anzahl Knoteneinrichtungen der Datenbankinstanz sein. Die Last kann insbesondere eine Rechenlast, eine Transaktionslast oder dergleichen sein.

[0031] Die Knoteneinrichtungen der Datenbankinstanz können untereinander regelmäßig Lastinformationen austauschen. Analog dazu können die mehreren Datenbankinstanzen untereinander regelmäßig Lastinformationen austauschen. Dergestalt können eine jeweilige Knoteneinrichtung und eine jeweilige Datenbankinstanz über Informationen über eine Last der eigenen sowie der anderen der mehreren Datenbankinstanzen verfügen.

[0032] Insbesondere kann eine jeweilige der mehreren Datenbankinstanzen derart eingerichtet sein, dass sie die zu bestätigende Transaktion nur bestätigt, wenn die bestätigende Datenbankinstanz die Datenbankinstanz mit der niedrigsten oder einer der niedrigsten Lasten unter den mehreren Datenbankinstanzen ist.

[0033] Demgemäß kann vorteilhaft eine explizite Entscheidung getroffen werden, die einen Lastausgleich in dem verteilten Datenbanksystem bewirkt.

[0034] Gemäß einer weiteren Ausführungsform ist das verteilte Datenbanksystem dazu eingerichtet, in Abhängigkeit von einem oder mehreren Rückbezügen der zu bestätigenden Transaktion zu entscheiden, durch welche der mehreren Datenbankinstanzen die zu bestätigende Transaktion bestätigt wird.

[0035] Eine zu bestätigende Transaktion kann auf vergangene Transaktionen (auch als "rückbezogene Transaktion" bezeichnet) Bezug nehmen. Beispielsweise kann eine Kryptotoken-Transaktion, die ein Kryptotoken aus einem Sender-Wallet an ein Empfänger-Wallet transaktioniert, Bezug auf eine vergangene Kryptotoken-Transaktion nehmen, in welcher das Kryptotoken in das Sender-Wallet transaktioniert wurde, um ihre Berechtigung zum Transaktionieren des Kryptotokens zu dokumentieren.

[0036] Insbesondere kann es zum Prüfen der zu bestätigenden Transaktion somit erforderlich sein, vergangene Transaktionen abzurufen bzw. zu überprüfen. In bestimmten Fällen, etwa wenn keine Inter-Chain- bzw. datenbankinstanzübergreifende Kommunikation vorgesehen ist, kann es daher erforderlich sein, dass die zu bestätigende Transaktion in einem bestimmten Transaktionsbuch bestätigt wird, in dem ihre Rückbezüge auflösbar sind.

[0037] Die Entscheidung gemäß der vorliegenden Ausführungsform kann implizit beim Bestätigen dadurch erfolgen, dass eine Datenbankinstanz, in welcher der Rückbezug der zu bestätigenden Transaktion nicht auflösbar ist (die rückbezogene Transaktion nicht in dem Transaktionsbuch der Datenbankinstanz vorliegt), die zu bestätigende Transaktion nicht erfolgreich prüfen kann, während diejenige Datenbankinstanz, in welcher der Rückbezug der zu bestätigenden Transaktion auflösbar ist, das Prüfen der zu bestätigenden Transaktion erfolgreich verläuft.

[0038] Demgemäß können vorteilhaft auch bei einer automatisierten Verteilung von zu bestätigenden Transaktionen über mehrere Datenbankinstanzen diejenigen Transaktionen, die in einer bestimmten der mehreren Datenbankinstanzen bestätigt werden müssen, in der korrekten Datenbankinstanz bestätigt werden.

[0039] Gemäß einer weiteren Ausführungsform ist das verteilte Datenbanksystem dazu eingerichtet, bei der Entscheidung, durch welche der mehreren Datenbankinstanzen die zu bestätigende Transaktion bestätigt wird, eine jeweilige Datenbankinstanz nur zu berücksichtigen, wenn die Datenbankinstanz von der zu bestätigenden Transaktion spezifiziert ist.

[0040] Demgemäß kann eine dem verteilten Datenbanksystem bereitgestellte, zu bestätigende Transaktion spezifizieren, von welcher oder welchen der Datenbankinstanzen sie zu bestätigen ist.

[0041] Beispielsweise kann eine zu bestätigende kritische Transaktion nur solche Datenbankinstanzen spezifizieren, die über ein vordefiniertes Mindest-Manipulationsschutz-Niveau aufweisen, um vorteilhaft zu vermeiden, im Rahmen des automatisierten Lastausgleichs in einer Datenbankinstanz mit zu geringem Manipulationsschutz bestätigt zu werden.

[0042] Gemäß einer weiteren Ausführungsform ist das verteilte Datenbanksystem dazu eingerichtet, durch eine automatische dezentrale Konsensbildung zwischen den mehreren Datenbankinstanzen zu entscheiden, durch welche der mehreren Datenbankinstanzen die zu bestätigende Transaktion bestätigt wird.

[0043] Anders ausgedrückt kann auf eine explizite Entscheidung, welche der Datenbankinstanzen die zu bestätigenden Transaktion bestätigt wird, die vor dem Bestätigen getroffen wird, verzichtet werden. Vielmehr kann eine Konkurrenzsituation zwischen den Datenbankinstanzen geschaffen werden, in der sich ein Konsens, welche der Datenbankinstanzen die zu bestätigende Transaktion bestätigt, automatisch herausbildet.

[0044] Beispielsweise kann eine im Rahmen der Konsensregel einer jeweiligen Datenbankinstanz zu prüfende Bedingung für die Gültigkeit einer Transaktion oder

eines Blocks lauten, dass die Transaktion, um als gültig bestätigt zu werden, noch in keiner anderen der Datenbankinstanzen bestätigt sein darf. Dies kann beispielsweise durch Inter-Chain-Kommunikation verifiziert werden.

[0045] Somit kann vorteilhaft eine selbstregulierende Verteilung der Transaktionslast über mehrere Datenbankinstanzen realisiert werden.

[0046] Gemäß einer weiteren Ausführungsform ist das verteilte Datenbanksystem dazu eingerichtet, eine Anzahl unbestätigter Transaktionen vorzuhalten. Eine jeweilige der mehreren Datenbankinstanzen ist dazu eingerichtet, die zu bestätigende Transaktion unter der vorgehaltenen Anzahl unbestätigter Transaktionen auszuwählen und zu bestätigen. Hierbei ist das verteilte Datenbanksystem dazu eingerichtet, die zu bestätigende Transaktion aus der vorgehaltenen Anzahl unbestätigter Transaktionen zu entfernen, sobald eine definierte Anzahl der Datenbankinstanzen die zu bestätigende Transaktion bestätigt hat.

[0047] Die unbestätigte Transaktion kann eine Transaktion sein, die dem Datenbanksystem bereitgestellt wurde und von dem Datenbanksystem zu bestätigen ist. Die Bezeichnung "unbestätigte Transaktion" wird insbesondere verwendet, um eine Transaktion zu bezeichnen, für die noch nicht entschieden ist, welche der Datenbankinstanzen sie bestätigt, während die Bezeichnung "zu bestätigende Transaktion" insbesondere verwendet wird, um eine Transaktion bzw. eine Instanz oder Kopie einer Transaktion zu bezeichnen, die von einer Datenbankinstanz zum Bestätigen ausgewählt wurde.

[0048] Unter "vorhalten" ist hierbei insbesondere zu verstehen, dass ein Pool (eine vorgehaltene Anzahl) der unbestätigten Transaktionen in einem zentralen oder dezentralen Speichermittel derart vorgehalten wird, dass eine jeder der Datenbankinstanzen (eine jeder der Knoteneinrichtungen jeder der Datenbankinstanzen) darauf Zugriff hat. Unter einem dezentralen Speichermittel kann dabei beispielsweise verstanden werden, dass eine Instanz des Pools der unbestätigten Transaktionen mindestens einer, bevorzugt, jeder, der Knoteneinrichtungen jeder der Datenbankinstanzen gespeichert ist und Informationen über das Hinzufügen oder Entferner einer Transaktion aus dem Pool auf Peer-to-Peer-Weise oder dergleichen zwischen den Knoteneinrichtungen der Datenbankinstanzen ausgetauscht werden.

[0049] Die definierte Anzahl der Datenbankinstanzen, die die zu bestätigende Transaktion bestätigt hat, ab derer die zu bestätigende Transaktion aus dem Pool der unbestätigten Transaktion entfernt wird, kann eine vordefinierte Anzahl sein und kann eins oder mehr sein. Die definierte Anzahl kann abhängig von der Art, wie etwa einem gebotenen Manipulationsschutzniveau oder dergleichen, der bestätigenden Datenbankinstanz definiert sein. Beispielsweise kann die zu bestätigende Transaktion entweder aus der vorgehaltenen Anzahl unbestätigter Transaktion entfernt werden, sobald die zu bestätigende Transaktion von einer Haupt-Chain bestätigt wird, oder, sobald die zu bestätigende Transaktion von mehreren Seiten-Chains bestätigt wird.

[0050] Durch das Vorhalten des Pools von unbestätigten Transaktionen, aus dem die zu bestätigende Transaktion entfernt wird, wenn sie genügend oft bestätigt wurde, kann besonders vorteilhaft eine automatische dezentrale Konsensbildung zwischen den mehreren Datenbankinstanzen darüber erfolgen, durch welche der mehreren Datenbankinstanzen die zu bestätigende Transaktion bestätigt wird.

[0051] Gemäß einer weiteren Ausführungsform ist die definierte Anzahl der Datenbankinstanzen durch die zu bestätigende Transaktion definierbar.

[0052] Demgemäß kann die definierte Anzahl der Datenbanksysteme, durch welche die zu bestätigende Transaktion bestätigt wird, bevor sie aus dem Pool der vorgehaltenen unbestätigten Transaktionen entfernt wird, durch die Transaktion definiert bzw. spezifiziert werden.

[0053] Somit kann diese Entscheidung vorteilhaft einem Anwendungsdesigner je nach Art der bereitzustellenden Transaktion überlassen bleiben und muss nicht in dem verteilten Datenbanksystem vordefiniert sein.

[0054] Gemäß einer weiteren Ausführungsform ist eine Konsensregel einer jeweiligen der Datenbankinstanzen derart eingerichtet, dass eine Vergütung für das Bestätigen der zu bestätigenden Transaktion nur eine von der zu bestätigenden Transaktion spezifizierte Anzahl Male vergeben wird.

[0055] Demgemäß kann ein Anreiz für das Bestätigen einer Transaktion wegfallen, sobald die Transaktion die spezifizierte Anzahl von Malen vergeben worden ist. Somit kann ebenfalls vorteilhaft durch einen Anwendungsdesigner durch geeignetes Wählen der Vergütung und der spezifizierten Anzahl von Malen implizit vorgegeben werden, durch wie viele Datenbankinstanzen die zu bestätigende Transaktion zu bestätigen ist.

[0056] Gemäß einer weiteren Ausführungsform ist das Datenbanksystem dazu eingerichtet, die Transaktionsbücher der mehreren Datenbankinstanzen in vordefinierten Zeitabständen miteinander zu synchronisieren.

[0057] Beispielsweise kann in vordefinierten Zeitabständen ein jeweiliger Zustand, der durch die Abfolge von Transaktionen des Transaktionsbuchs der jeweiligen Datenbankinstanz definiert ist, zwischen den Datenbankinstanzen synchronisiert werden.

[0058] Der vordefinierte Zeitabstand kann beispielsweise durch den Zeitabstand vordefiniert sein, der gemäß der Konsensregel derjenigen der Datenbankinstanzen mit der geringsten Blockbildungsrate im Mittel zwischen der Bildung eines oder einer Anzahl von Blöcken verstreicht. Es sei angemerkt, dass der vordefinierte Zeitabstand ein Mittelwert sein kann, der sich im statistischen Mittel ergibt, und die Zeitabstände zwischen zwei jeweiligen Synchronisierung jeweils variieren können.

[0059] Durch eine solche regelmäßige Synchronisation zwischen den mehreren Datenbankinstanzen kann vorteilhaft erreicht werden, dass eine neue zu bestäti-

gende Transaktion, die nach einer der Synchronisationen bereitgestellt wird und einen Rückbezug enthält, der sich auf eine Transaktion vor der Synchronisation bezieht, unter Erhalt des Rückbezugs in einer beliebigen der mehreren Datenbankinstanzen bestätigbar ist. In dieser Ausführungsform wird eine zu bestätigende Transaktionen mit Rückbezug beispielsweise genau dann in einer bestimmten Datenbankinstanz bestätigt, wenn ein Rückbezug der zu bestätigenden Transaktion eine rückbezogene Transaktion betrifft, die nach der zurückliegenden letzten Synchronisation bestätigt wurde; andernfalls kann auch eine zu bestätigende Transaktion mit Rückbezug in einer beliebigen der Datenbankinstanzen bestätigt werden.

[0060] Gemäß einer weiteren Ausführungsform sind die mehreren Datenbankinstanzen dazu eingerichtet, beim Bestätigen der zu bestätigenden Transaktion durch eine der mehreren Datenbankinstanzen einen Rückbezug der zu bestätigenden Transaktion datenbankinstanzübergreifend zu verifizieren.

[0061] Dies kann beispielsweise mittels datenbankinstanzübergreifender Kommunikation bzw. Inter-Chain-Kommunikation realisiert werden. Anders ausgedrückt kann eine jeweilige Knoteneinrichtung einer jeder der Datenbankinstanzen mindestens über Lesezugriff auf die Transaktionsbücher auch aller übrigen der Datenbankinstanzen verfügen.

[0062] Demgemäß kann vorteilhaft jede zu bestätigende Transaktion in jedem der Transaktionsbücher bestätigt werden, da gegebenenfalls vorhandene Rückbezüge der zu bestätigenden Transaktion datenbankinstanzübergreifend auflösbar sein können. Die balancierte Verteilung der Transaktionslast kann dadurch weiter verbessert werden.

[0063] Gemäß einer weiteren Ausführungsform ist das Transaktionsbuch einer der mehreren Datenbankinstanzen ein Hauptbuch und das Transaktionsbuch einer jeweiligen weiteren der mehreren Datenbankinstanzen ist ein jeweiliges mit dem Hauptbuch kryptographisch verknüpftes Seitenbuch.

[0064] Die kryptographische Verknüpfung kann beispielsweise dadurch erzielt werden, dass in dem Hauptbuch in vorgegebenen Zeitabständen ein Hashwert eines jeweils aktuellen Zustands des Seitenbuchs bestätigt wird.

[0065] Somit kann vorteilhaft selbst, wenn ein jeweiliges Seitenbuch basierend auf der Konsensregel der zugehörigen Datenbankinstanz einen schwächeren Manipulationsschutz aufweist als das Hauptbuch, das Seitenbuch dennoch langfristig von dem höheren Manipulationsschutz des Hauptbuchs profitieren.

[0066] Gemäß einer weiteren Ausführungsform ist das verteilte Datenbanksystem dazu eingerichtet, die zu bestätigende Transaktion entweder in dem Hauptbuch oder in einer Anzahl der Seitenbücher zu bestätigen.

[0067] Anders ausgedrückt kann eine einzelne Bestätigung im Hauptbuch ausreichend sein, wohingegen im Falle einer Bestätigung im Nebenbuch mehrere Bestätigungen erforderlich sein können.

[0068] Hierdurch kann vorteilhaft ein Manipulationsschutz der bestätigten Transaktion durch mehrfaches Bestätigen der zu bestätigenden Transaktion in mehreren Seitenbüchern auch dann verbessert werden, wenn die zu bestätigende Transaktion nicht im Hauptbuch bestätigt wird. Wenn die zu bestätigende Transaktion dagegen im Hauptbuch bestätigt wird, kann dies als ausreichend betrachtet werden und Ressourcen für ein mehrfaches Bestätigen in weiteren Transaktionsbüchern können in diesem Fall eingespart werden.

[0069] Die jeweilige Einheit, zum Beispiel Knoteneinrichtung, Datenbanksystem und/oder verteiltes Datenbanksystem, kann hardwaretechnisch und/oder auch softwaretechnisch implementiert sein. Bei einer hardwaretechnischen Implementierung kann die jeweilige Einheit als Vorrichtung oder als Teil einer Vorrichtung, zum Beispiel als Computer oder als Mikroprozessor oder als Steuerrechner ausgebildet sein. Bei einer softwaretechnischen Implementierung kann die jeweilige Einheit als Computerprogrammprodukt, als eine Funktion, als eine Routine, als Teil eines Programmcodes oder als ausführbares Objekt ausgebildet sein.

[0070] Weiterhin wird ein Computerprogrammprodukt vorgeschlagen, welches auf einer programmgesteuerten Einrichtung die Durchführung des wie oben erläuterten Verfahrens veranlasst.

[0071] Ein Computerprogrammprodukt, wie z.B. ein Computerprogramm-Mittel, kann beispielsweise als Speichermedium, wie z.B. Speicherkarte, USB-Stick, CD-ROM, DVD, oder auch in Form einer herunterladbaren Datei von einem Server in einem Netzwerk bereitgestellt oder geliefert werden. Dies kann zum Beispiel in einem drahtlosen Kommunikationsnetzwerk durch die Übertragung einer entsprechenden Datei mit dem Computerprogrammprodukt oder dem Computerprogramm-Mittel erfolgen.

[0072] Gemäß einem weiteren Aspekt wird ein Verfahren zum Betreiben eines verteilten Datenbanksystems mit mehreren Datenbankinstanzen vorgeschlagen. Eine jeweilige der mehreren Datenbankinstanzen ist durch eine Anzahl Knoteneinrichtungen gebilde, die dazu eingerichtet sind, gemeinsam konsensbasiert eines von mehreren Transaktionsbüchern des verteilten Datenbanksystems zu verwalten. Das Verfahren umfasst ein Bestätigen einer zu bestätigenden Transaktion durch Aufnehmen in das Transaktionsbuch einer oder mehrerer der mehreren Datenbankinstanzen, wobei das verteilte Datenbanksystem entscheidet, durch welche der mehreren Datenbankinstanzen die zu bestätigende Transaktion bestätigt wird.

[0073] Die für das vorgeschlagene verteilte Datenbanksystem beschriebenen Ausführungsformen und Merkmale gelten für das vorgeschlagene Verfahren entsprechend.

[0074] Nähere Einzelheiten und weitere Varianten von auf Blockketten-Technologie basierenden verteilten Datenbanksystemen, auf die die vorgeschlagene Lösung

anwendbar ist, werden im Folgenden erläutert.

[0075] Sofern es in der nachstehenden Beschreibung nicht anders angegeben ist, beziehen sich die Begriffe "durchführen", "berechnen", "rechnergestützt", "rechnen", "feststellen", "generieren", "konfigurieren", "rekonstruieren" und dergleichen vorzugsweise auf Handlungen und/oder Prozesse und/oder Verarbeitungsschritte, die Daten verändern und/oder erzeugen und/oder die Daten in andere Daten überführen, wobei die Daten insbesondere als physikalische Größen dargestellt werden oder vorliegen können, beispielsweise als elektrische Impulse. Insbesondere sollte der Ausdruck "Computer" möglichst breit ausgelegt werden, um insbesondere alle elektronischen Geräte mit Datenverarbeitungseigenschaften abzudecken. Computer können somit beispielsweise Personal Computer, Server, speicherprogrammierbare Steuerungen (SPS), Handheld-Computer-Systeme, Pocket-PC-Geräte, Mobilfunkgeräte und andere Kommunikationsgeräte, die rechnergestützt Daten verarbeiten können, Prozessoren und andere elektronische Geräte zur Datenverarbeitung sein.

[0076] Unter "rechnergestützt" kann im Zusammenhang mit der Erfindung beispielsweise eine Implementierung des Verfahrens verstanden werden, bei dem insbesondere ein Prozessor mindestens einen Verfahrensschritt des Verfahrens ausführt.

[0077] Unter einem Prozessor kann im Zusammenhang mit der Erfindung beispielsweise eine Maschine oder eine elektronische Schaltung verstanden werden. Bei einem Prozessor kann es sich insbesondere um einen Hauptprozessor (engl. Central Processing Unit, CPU), einen Mikroprozessor oder einen Mikrokontroller, beispielsweise eine anwendungsspezifische integrierte Schaltung oder einen digitalen Signalprozessor, möglicherweise in Kombination mit einer Speichereinheit zum Speichern von Programmbefehlen, etc. handeln. Bei einem Prozessor kann es sich beispielsweise auch um einen IC (integrierter Schaltkreis, engl. Integrated Circuit), insbesondere einen FPGA (engl. Field Programmable Gate Array) oder einen ASIC (anwendungsspezifische integrierte Schaltung, engl. Application-Specific Integrated Circuit), oder einen DSP (Digitaler Signalprozessor, engl. Digital Signal Processor) oder einen Grafikprozessor GPU (Graphic Processing Unit) handeln. Auch kann unter einem Prozessor ein virtualisierter Prozessor, eine virtuelle Maschine oder eine Soft-CPU verstanden werden. Es kann sich beispielsweise auch um einen programmierbaren Prozessor handeln, der mit Konfigurationsschritten zur Ausführung des genannten erfindungsgemäßen Verfahrens ausgerüstet wird oder mit Konfigurationsschritten derart konfiguriert ist, dass der programmierbare Prozessor die erfindungsgemäßen Merkmale des Verfahrens, der Komponente, der Module oder anderer Aspekte und/oder Teilaspekte der Erfindung realisiert.

[0078] Unter einer "Speichereinheit", einem "Speichermodul" und dergleichen kann im Zusammenhang mit der Erfindung beispielsweise ein flüchtiger Speicher in Form von Arbeitsspeicher (engl. Random-Access Memory, RAM) oder ein dauerhafter Speicher wie eine Festplatte oder ein Datenträger verstanden werden.

[0079] Unter einem "Modul" kann im Zusammenhang mit der Erfindung beispielsweise ein Prozessor und/oder eine Speichereinheit zum Speichern von Programmbefehlen verstanden werden. Beispielsweise ist der Prozessor speziell dazu eingerichtet, die Programmbefehle derart auszuführen, damit der Prozessor Funktionen ausführt, um das erfindungsgemäße Verfahren oder einen Schritt des erfindungsgemäßen Verfahrens zu implementieren oder realisieren. Ein Modul kann beispielsweise auch ein Knoten des verteilten Datenbanksystems sein, der beispielsweise die spezifischen Funktionen/Merkmale eines entsprechenden Moduls realisiert. Die jeweiligen Module können beispielsweise auch als separate bzw. eigenständige Module ausgebildet sein. Hierzu können die entsprechenden Module beispielsweise weitere Elemente umfassen. Diese Elemente sind beispielsweise eine oder mehrere Schnittstellen (z. B. Datenbankschnittstellen, Kommunikationsschnittstellen - z. B. Netzwerkschnittstelle, WLAN-Schnittstelle) und/oder eine Evaluierungseinheit (z. B. ein Prozessor) und/oder eine Speichereinheit. Mittels der Schnittstellen können beispielsweise Daten ausgetauscht (z. B. empfangen, übermittelt, gesendet oder bereitgestellt werden). Mittels der Evaluierungseinheit können Daten beispielsweise rechnergestützt und/oder automatisiert verglichen, überprüft, verarbeitet, zugeordnet oder berechnet werden. Mittels der Speichereinheit können Daten beispielsweise rechnergestützt und/oder automatisiert gespeichert, abgerufen oder bereitgestellt werden.

[0080] Unter "umfassen", insbesondere in Bezug auf Daten und/oder Informationen, kann im Zusammenhang mit der Erfindung beispielsweise ein (rechnergestütztes) Speichern einer entsprechenden Information bzw. eines entsprechenden Datums in einer Datenstruktur/Datensatz (die z. B. wiederum in einer Speichereinheit gespeichert ist) verstanden werden.

[0081] Unter "zuordnen", insbesondere in Bezug auf Daten und/oder Informationen, kann im Zusammenhang mit der Erfindung beispielsweise eine rechnergestützte Zuordnung von Daten und/oder Informationen verstanden werden. Beispielsweise wird einem ersten Datum hierzu mittels einer Speicheradresse oder eines eindeutigen Identifizierers (engl. unique identifier (UID)) ein zweites Datum zugeordnet, in dem z. B. das erste Datum zusammen mit der Speicheradresse oder des eindeutigen Identifizierers des zweiten Datums zusammen in einem Datensatz gespeichert wird.

[0082] Unter "bereitstellen", insbesondere in Bezug auf Daten und/oder Informationen, kann im Zusammenhang mit der Erfindung beispielsweise ein rechnergestütztes Bereitstellen verstanden werden. Das Bereitstellen erfolgt beispielsweise über eine Schnittstelle (z. B. eine Datenbankschnittstelle, eine Netzwerkschnittstelle, eine Schnittstelle zu einer Speichereinheit). Über diese Schnittstelle können beispielsweise beim Bereitstellen

entsprechende Daten und/oder Informationen übermittelt und/oder gesendet und/oder abgerufen und/oder empfangen werden.

[0083] Unter "bereitstellen" kann im Zusammenhang mit der Erfindung beispielsweise auch ein Laden oder ein Speichern, beispielsweise einer Transaktion mit entsprechenden Daten verstanden werden. Dies kann beispielsweise auf oder von einem Speichermodul erfolgen. Unter "Bereitstellen" kann beispielsweise auch ein Übertragen (oder ein Senden oder ein Übermitteln) von entsprechenden Daten von einem Knoten zu einem anderen Knoten der Blockkette oder des verteilten Datenbanksystems (bzw. deren Infrastruktur) verstanden werden.

[0084] Unter "Smart-Contract-Prozess" kann im Zusammenhang mit der Erfindung insbesondere ein Ausführen eines Programmcodes (z. B. der Steuerbefehle) in einem Prozess durch das verteilte Datenbanksystem bzw. deren Infrastruktur verstanden werden.

[0085] Unter einer "Prüfsumme", beispielsweise eine Datenblockprüfsumme, eine Datenprüfsumme, eine Knotenprüfsumme, eine Transaktionsprüfsumme, eine Verkettungsprüfsumme oder dergleichen, kann im Zusammenhang mit der Erfindung beispielsweise eine kryptographische Prüfsumme oder kryptographischer Hash bzw. Hash-Wert verstanden werden, die insbesondere mittels einer kryptographischen Hashfunktion über einen Datensatz und/oder Daten und/oder eine oder mehrere der Transaktionen und/oder einem Teilbereich eines Datenblocks (z. B. der Block-Header eines Blocks einer Blockkette oder Datenblock-Header eines Datenblocks des verteilten Datenbanksystems oder nur einem Teil der Transaktionen eines Datenblocks) gebildet oder berechnet werden. Bei einer Prüfsumme kann es sich insbesondere um eine Prüfsumme/n oder Hash-Wert/e eines Hash-Baumes (z. B. Merkle-Baum, Patricia-Baum) handeln. Weiterhin kann darunter insbesondere auch eine digitale Signatur oder ein kryptographischer Nachrichtenauthentisierungscode verstanden werden. Mittels der Prüfsummen kann beispielsweise auf unterschiedlichen Ebenen des Datenbanksystems ein kryptographischer Schutz/Manipulationsschutz für die Transaktionen und die darin gespeicherten Daten (sätze) realisiert werden. Ist beispielsweise eine hohe Sicherheit gefordert, werden beispielsweise die Prüfsummen auf Transaktionsebene erzeugt und überprüft. Ist eine weniger hohe Sicherheit gefordert, werden beispielsweise die Prüfsummen auf Blockebene (z. B. über den ganzen Datenblock oder nur über einen Teil des Datenblocks und/oder einen Teil der Transaktionen) erzeugt und überprüft.

[0086] Unter einer "Datenblockprüfsumme" kann im Zusammenhang mit der Erfindung eine Prüfsumme verstanden werden, die beispielsweise über einen Teil oder alle Transaktionen eines Datenblocks berechnet wird. Ein Knoten kann dann beispielsweise die Integrität/Authentizität des entsprechenden Teils eines Datenblocks mittels der Datenblockprüfsumme prüfen/feststellen. Zusätzlich oder alternativ kann die Datenblockprüfsumme insbesondere auch über Transaktionen eines vorhergehenden Datenblocks/Vorgänger-Datenblocks des Datenblocks gebildet worden sein. Die Datenblockprüfsumme kann dabei insbesondere auch mittels eines Hash-Baumes, beispielsweise einem Merkle-Baum [1] oder einem Patricia-Baum, realisiert werden, wobei die Datenblockprüfsumme insbesondere die Wurzel-Prüfsumme des Merkle-Baumes bzw. eines Patricia-Baumes bzw. eines binären Hash-Baumes ist. Insbesondere werden Transaktionen mittels weiterer Prüfsummen aus dem Merkle-Baum bzw. Patricia-Baum abgesichert (z. B. unter Verwendung der Transaktionsprüfsummen), wobei insbesondere die weiteren Prüfsummen Blätter im Merkle-Baum bzw. Patricia-Baum sind. Die Datenblockprüfsumme kann damit beispielsweise die Transaktionen absichern, indem die Wurzel-Prüfsumme aus den weiteren Prüfsummen gebildet wird. Die Datenblockprüfsumme kann insbesondere für Transaktionen eines bestimmten Datenblocks der Datenblöcke berechnet werden. Insbesondere kann eine solche Datenblockprüfsumme in einen nachfolgenden Datenblock des bestimmten Datenblocks eingehen, um diesen nachfolgenden Datenblock beispielsweise mit seinen vorhergehenden Datenblöcken zu verketten und insbesondere damit eine Integrität des verteilten Datenbanksystems prüfbar zu machen. Hierdurch kann die Datenblockprüfsumme beispielsweise die Funktion der Verkettungsprüfsumme übernehmen oder in die Verkettungsprüfsumme eingehen. Der Header eines Datenblocks (z. B. eines neuen Datenblocks oder des Datenblocks für den die Datenblockprüfsumme gebildet wurde) kann beispielsweise die Datenblockprüfsumme umfassen.

[0087] Unter "Transaktionsprüfsumme" kann im Zusammenhang mit der Erfindung eine Prüfsumme verstanden werden, die insbesondere über eine Transaktion eines Datenblocks gebildet wird. Zusätzlich kann beispielsweise eine Berechnung einer Datenblockprüfsumme für einen entsprechenden Datenblock beschleunigt werden, da hierfür beispielsweise bereits berechnete Transaktionsprüfsummen gleich als Blätter z. B. eines Merkle-Baumes verwendet werden können.

[0088] Unter einer "Verkettungsprüfsumme" kann im Zusammenhang mit der Erfindung eine Prüfsumme verstanden werden, die insbesondere einen jeweiligen Datenblock des verteilten Datenbanksystems den vorhergehenden Datenblock des verteilten Datenbanksystems angibt bzw. referenziert (in der Fachliteratur insbesondere häufig als "previous block hash" bezeichnet) [1]. Hierfür wird insbesondere für den entsprechenden vorhergehenden Datenblock eine entsprechende Verkettungsprüfsumme gebildet. Als Verkettungsprüfsumme kann beispielsweise eine Transaktionsprüfsumme oder die Datenblockprüfsumme eines Datenblocks (also ein vorhandener Datenblock des verteilten Datenbanksystems) verwendet werden, um einen neuen Datenblock mit einem (vorhandenen) Datenblock des verteilten Datenbanksystems zu verketten. Es ist beispielsweise aber auch möglich, dass eine Prüfsumme über einen Header des vorhergehenden Datenblocks oder über den gesam-

ten vorhergehenden Datenblock gebildet wird und als Verkettungsprüfsumme verwendet wird. Dies kann beispielsweise auch für mehrere oder alle vorhergehenden Datenblöcke berechnet werden. Es ist beispielsweise auch realisierbar, dass über den Header eines Datenblocks und der Datenblockprüfsumme die Verkettungsprüfsumme gebildet wird. Ein jeweiliger Datenblock des verteilten Datenbanksystems umfasst jedoch vorzugsweise jeweils eine Verkettungsprüfsumme, die für einen vorhergehenden Datenblock, insbesondere noch bevorzugter den direkt vorhergehenden Datenblock, des jeweiligen Datenblockes berechnet wurde bzw. sich auf diesen beziehen. Es ist beispielsweise auch möglich, dass eine entsprechende Verkettungsprüfsumme auch nur über einen Teil des entsprechenden Datenblocks (z. B. vorhergehenden Datenblock) gebildet wird. Hierdurch kann zum Beispiel ein Datenblock realisiert werden, der einen integritätsgeschützten Teil und einen ungeschützten Teil umfasst. Damit ließe sich beispielsweise ein Datenblock realisieren, dessen integritätsgeschützter Teil unveränderlich ist und dessen ungeschützter Teil auch noch später verändert werden kann. Unter integritätsgeschützt ist dabei insbesondere zu verstehen, dass eine Veränderung von integritätsgeschützten Daten mittels einer Prüfsumme feststellbar ist.

[0089] Die Daten, die beispielsweise in einer Transaktion eines Datenblocks gespeichert werden, können insbesondere auf unterschiedliche Weise bereitgestellt werden. Anstelle der Daten, z. B. Nutzerdaten wie Messdaten, Messwerte, Steuerwerte, oder Daten/Eigentumsverhältnisse zu Assets, kann beispielsweise eine Transaktion eines Datenblocks nur die Prüfsumme für diese Daten umfassen. Die entsprechende Prüfsumme kann dabei auf unterschiedliche Weise realisiert werden. Dies kann z. B. eine entsprechende Datenblockprüfsumme eines Datenblocks (mit den entsprechenden Daten) einer anderen Datenbank oder des verteilten Datenbanksystems sein, eine Transaktionsprüfsumme eines Datenblocks mit den entsprechenden Daten (des verteilten Datenbanksystems oder einer anderen Datenbank) oder eine Datenprüfsumme, die über die Daten gebildet wurde.

[0090] Zusätzlich kann die entsprechende Transaktion einen Verweis oder eine Angabe zu einem Speicherort (z. B. eine Adresse eines Fileservers und Angaben, wo die entsprechenden Daten auf dem Fileserver zu finden sind; oder eine Adresse einer anderen verteilten Datenbank, welche die Daten umfasst) umfassen. Die entsprechenden Daten könnten dann beispielsweise auch in einer weiteren Transaktion eines weiteren Datenblocks des verteilten Datenbanksystems bereitgestellt werden (z. B. wenn die entsprechenden Daten und die zugehörigen Prüfsummen in unterschiedlichen Datenblöcken umfasst sind). Es ist beispielsweise aber auch denkbar, dass diese Daten über einen anderen Kommunikationskanal (z. B. über eine andere Datenbank und/oder einen kryptographisch gesicherten Kommunikationskanal) bereitgestellt werden.

[0091] Auch kann beispielsweise zusätzlich zu der Prüfsumme ein Zusatzdatensatz (z. B. ein Verweis oder eine Angabe zu einem Speicherort) in der entsprechenden Transaktion abgelegt sein, der insbesondere einen Speicherort angibt, wo die Daten abgerufen werden können. Das ist insbesondere dahingehend vorteilhaft, um eine Datengröße der Blockkette oder des verteilten Datenbanksystems möglichst gering zu halten.

[0092] Unter "sicherheitsgeschützt" kann im Zusammenhang mit der Erfindung beispielsweise ein Schutz verstanden werden, der insbesondere durch ein kryptographisches Verfahren realisiert wird. Beispielsweise kann dies durch Nutzung des verteilten Datenbanksystems für das Bereitstellen oder Übertragen oder Senden von entsprechenden Daten/Transaktionen realisiert werden. Dies wird vorzugsweise durch eine Kombination der verschiedenen (kryptographischen) Prüfsummen erreicht, indem diese insbesondere synergetisch zusammenwirken, um beispielsweise die Sicherheit bzw. die kryptographische Sicherheit für die Daten der Transaktionen zu verbessern. Anders gesagt kann insbesondere unter "sicherheitsgeschützt" im Zusammenhang mit der Erfindung auch "kryptographisch geschützt" und/oder "manipulationsgeschützt" verstanden werden. Dabei kann "manipulationsgeschützt" auch als "integritätsgeschützt" bezeichnet werden.

[0093] Unter "Verketten der/von Datenblöcken eines verteilten Datenbanksystems" kann im Zusammenhang mit der Erfindung beispielsweise verstanden werden, dass Datenblöcke jeweils eine Information (z. B. Verkettungsprüfsumme) umfassen, die auf einen anderen Datenblock oder mehrere andere Datenblöcke des verteilten Datenbanksystems verweisen bzw. diese referenzieren [1] [4] [5] .

[0094] Unter "Einfügen in das verteilte Datenbanksystem" und dergleichen kann im Zusammenhang mit der Erfindung beispielsweise verstanden werden, dass insbesondere eine Transaktion bzw. die Transaktionen oder ein Datenblock mit seinen Transaktionen an einen oder mehrere Knoten eines verteilten Datenbanksystems übermittelt wird. Werden diese Transaktionen beispielsweise erfolgreich validiert (z. B. durch den/die Knoten), werden diese Transaktionen insbesondere als neuer Datenblock mit mindestens einem vorhandenen Datenblock des verteilten Datenbanksystems verkettet [1][4][5]. Hierzu werden die entsprechenden Transaktionen beispielsweise in einem neuen Datenblock gespeichert. Insbesondere kann dieses Validieren und/oder Verketten durch einen vertrauenswürdigen Knoten (z. B. einen Mining Node, ein Blockketten-Orakel oder eine Blockketten-Plattform) erfolgen. Insbesondere kann dabei unter einer Blockketten-Plattform eine Blockkette als Dienst (engl. Blockkette als Service) verstanden werden, wie dies insbesondere durch Microsoft oder IBM vorgeschlagen wird. Insbesondere können ein vertrauenswürdiger Knoten und/oder ein Knoten jeweils eine Knoten-Prüfsumme (z. B. eine digitale Signatur) in einem Datenblock hinterlegen (z. B. in denen von ihnen validierten und er-

zeugten Datenblock, der dann verkettet wird), um insbesondere eine Identifizierbarkeit des Erstellers des Datenblockes zu ermöglichen und/oder eine Identifizierbarkeit des Knotens zu ermöglichen. Dabei gibt diese Knoten-Prüfsumme an, welcher Knoten beispielsweise den entsprechenden Datenblock mit mindestens einem anderen Datenblock des verteilten Datenbanksystems verkettet hat.

[0095] Unter "Transaktion" bzw. "Transaktionen" können im Zusammenhang mit der Erfindung beispielsweise ein Smart-Contract [4] [5], eine Datenstruktur oder ein Transaktionsdatensatz verstanden werden, der insbesondere jeweils eine der Transaktionen oder mehrere Transaktionen umfasst. Unter "Transaktion" bzw. "Transaktionen" können im Zusammenhang mit der Erfindung beispielsweise auch die Daten einer Transaktion eines Datenblocks einer Blockkette (engl. Blockchain) verstanden werden. Eine Transaktion kann insbesondere einen Programmcode umfassen, der beispielsweise einen Smart Contract realisiert. Beispielsweise können im Zusammenhang mit der Erfindung unter Transaktion auch eine Steuertransaktion und/oder Bestätigungstransaktion verstanden werden. Alternativ kann eine Transaktion beispielsweise eine Datenstruktur sein, die Daten speichert (z. B. die Steuerbefehle und/oder Vertragsdaten und/oder andere Daten wie Videodaten, Nutzerdaten, Messdaten etc.). Insbesondere ist unter "Speichern von Transaktionen in Datenblöcken", "Speichern von Transaktionen" und dergleichen ein direktes Speichern oder indirektes Speichern zu verstehen. Unter einem direkten Speichern kann dabei beispielsweise verstanden werden, dass der entsprechende Datenblock (des verteilten Datenbanksystems) oder die entsprechende Transaktion des verteilten Datenbanksystems) die jeweiligen Daten umfasst. Unter einem indirekten Speichern kann dabei beispielsweise verstanden werden, dass der entsprechende Datenblock oder die entsprechende Transaktion eine Prüfsumme und optional einen Zusatzdatensatz (z. B. einen Verweis oder eine Angabe zu einem Speicherort) für entsprechende Daten umfasst und die entsprechenden Daten somit nicht direkt in dem Datenblock (oder der Transaktion) gespeichert sind (also stattdessen nur eine Prüfsumme für diese Daten). Insbesondere können beim Speichern von Transaktionen in Datenblöcken diese Prüfsummen beispielsweise validiert werden, so wie dies beispielsweise unter "Einfügen in das verteilte Datenbanksystem" erläutert ist.

[0096] Unter einem "Programmcode" (z. B. ein Smart Contract) kann im Zusammenhang mit der Erfindung beispielsweise ein Programmbefehl oder mehrere Programmbefehle verstanden werden, die insbesondere in einer oder mehreren Transaktionen gespeichert sind. Der Programmcode ist insbesondere ausführbar und wird beispielsweise durch das verteilte Datenbanksystem ausgeführt. Dies kann beispielsweise mittels einer Ausführungsumgebung (z. B. einer virtuellen Maschine) realisiert werden, wobei die Ausführungsumgebung bzw. der Programmcode vorzugsweise Turing-vollständig sind. Der Programmcode wird vorzugsweise durch die Infrastruktur des verteilten Datenbanksystems ausgeführt [4][5]. Dabei wird zum Beispiel eine virtuelle Maschine durch die Infrastruktur des verteilten Datenbanksystems realisiert.

[0097] Unter einem "Smart Contract" kann im Zusammenhang mit der Erfindung beispielsweise ein ausführbarer Programmcode verstanden werden [4][5] (siehe insbesondere Definition "Programmcode"). Der Smart Contract ist vorzugsweise in einer Transaktion eines verteilten Datenbanksystems (z. B. eine Blockkette) gespeichert, beispielsweise in einem Datenblock des verteilten Datenbanksystems. Beispielsweise kann der Smart Contract auf die gleiche Weise ausgeführt werden, wie dies bei der Definition von "Programmcode", insbesondere im Zusammenhang mit der Erfindung, erläutert ist.

[0098] Unter "Proof-of-Work" oder "Proof-of-Work-Nachweis" kann im Zusammenhang mit der Erfindung beispielsweise ein Lösen einer rechenintensiven Aufgabe verstanden werden, die insbesondere abhängig vom Datenblock-Inhalt/Inhalt einer bestimmten Transaktion zu lösen ist [1][4][5]. Eine solche rechenintensive Aufgabe wird beispielsweise auch als kryptographisches Puzzle bezeichnet.

[0099] Unter einem "verteilten Datenbanksystem", das beispielsweise auch als verteilte Datenbank bezeichnet werden kann, kann im Zusammenhang mit der Erfindung beispielsweise eine dezentral verteilte Datenbank, eine Blockkette (engl. Blockchain), ein distributed Ledger, ein verteiltes Speichersystem, ein distributed ledger technology (DLT) based system (DLTS), ein revisionssicheres Datenbanksystem, eine Cloud, ein Cloud-Service, eine Blockkette in einer Cloud oder eine Peer-to-Peer-Datenbank verstanden werden. Auch können beispielsweise unterschiedliche Implementierungen einer Blockkette oder eines DLTS verwendet werden, wie z. B. eine Blockkette oder ein DLTS, die mittels eines Directed Acylic Graph (DAG), eines kryptographischen Puzzles, einem Hashgraph oder eine Kombination aus den genannten Implementierungsvarianten [6][7]. Auch können beispielsweise unterschiedliche Konsensregeln bzw. Konsensusverfahren (engl. consensus algorithms) implementiert werden. Dies kann beispielsweise ein Konsensusverfahren mittels eines kryptographischen Puzzles, Gossip about Gossip, Virtual Voting oder eine Kombination der genannten Verfahren sein (z. B. Gossip about Gossip kombiniert mit Virtual Voting) [6][7]. Wird beispielsweise eine Blockkette verwendet, so kann diese insbesondere mittels einer Bitcoin-basierten Realisierung oder einer Ethereum-basierten Realisierung umgesetzt werden [1][4][5]. Unter einem "verteilten Datenbanksystem" kann beispielsweise auch ein verteiltes Datenbanksystem verstanden werden, von dem zumindest ein Teil seiner Knoten und/oder Geräte und/oder Infrastruktur durch eine Cloud realisiert sind. Beispielsweise sind die entsprechenden Komponenten als Knoten/Geräte in der Cloud (z. B. als virtueller Knoten in einer virtuellen Maschine) realisiert. Dies kann beispielsweise

mittels VM-Ware, Amazon Web Services oder Microsoft Azure erfolgen. Aufgrund der hohen Flexibilität der erläuterten Implementierungsvarianten können insbesondere auch Teilaspekte der genannten Implementierungsvarianten miteinander kombiniert werden, indem z. B. ein Hashgraph als Blockkette verwendet wird, wobei die Blockkette selbst z. B. auch blocklos sein kann.

[0100] Wird beispielsweise ein Directed Acylic Graph (DAG) verwendet (z. B. IOTA oder Tangle), sind insbesondere Transaktionen oder Blöcke oder Knoten des Graphen miteinander über gerichtete Kanten miteinander verbunden. Dies bedeutet insbesondere, dass (alle) Kanten (immer) die gleiche Richtung haben, ähnlich wie dies z. B. bei Zeit ist. Mit anderen Worten ist es insbesondere nicht möglich, rückwärts (also entgegen der gemeinsamen gleichen Richtung) die Transaktionen oder die Blöcke oder die Knoten des Graphen anzulaufen bzw. anzuspringen. Azyklisch bedeutet dabei insbesondere, dass es keine Schleifen bei einem Durchlaufen des Graphen gibt.

[0101] Bei dem verteilten Datenbanksystem kann es sich beispielsweise um ein öffentliches verteiltes Datenbanksystem (z. B. eine öffentliche Blockkette) oder ein geschlossenes (oder privates) verteiltes Datenbanksystem (z. B. eine private Blockkette) handeln.

[0102] Handelt es sich beispielsweise um ein öffentliches verteiltes Datenbanksystem, bedeutet dies, dass neue Knoten und/oder Geräte ohne Berechtigungsnachweise oder ohne Authentifizierung oder ohne Anmeldeinformationen oder ohne Credentials dem verteilten Datenbanksystem beitreten können bzw. von diesem akzeptiert werden. Insbesondere können in einem solchen Fall die Betreiber der Knoten und/oder Geräte anonym bleiben.

[0103] Handelt es sich bei dem verteilten Datenbanksystem beispielsweise um ein geschlossenes verteiltes Datenbanksystem, benötigen neue Knoten und/oder Geräte beispielsweise einen gültigen Berechtigungsnachweis und/oder gültige Authentifizierungsinformationen und/oder gültige Credentials und/oder gültige Anmeldeinformationen, um dem verteilten Datenbanksystem beitreten können bzw. von diesem akzeptiert zu werden.

[0104] Bei einem verteilten Datenbanksystem kann es sich beispielsweise auch um ein verteiltes Kommunikationssystem zum Datenaustausch handeln. Dies kann beispielsweise ein Netzwerk oder ein Peer-2-Peer Netzwerk sein.

[0105] Unter "Datenblock", der insbesondere je nach Kontext und Realisierung auch als "Glied" oder "Block" bezeichnet sein kann, kann im Zusammenhang mit der Erfindung beispielsweise ein Datenblock eines verteilten Datenbanksystems (z. B. eine Blockkette oder eine Peer-to-Peer-Datenbank) verstanden werden, die insbesondere als Datenstruktur realisiert ist und vorzugsweise jeweils eine der Transaktionen oder mehrere der Transaktionen umfasst. Bei einer Implementierung kann beispielsweise die Datenbank (oder das Datenbanksystem) ein DLT basiertes System (DLTS) oder eine Blockkette sein und ein Datenblock ein Block der Blockkette oder des DLTS. Ein Datenblock kann beispielsweise Angaben zur Größe (Datengröße in Byte) des Datenblocks, einen Datenblock-Header (engl. Block-header), einen Transaktionszähler und eine oder mehrere Transaktionen umfassen [1]. Der Datenblock-Header kann beispielsweise eine Version, eine Verkettungsprüfsumme, eine Datenblockprüfsumme, einen Zeitstempel, einen Proof-of-Work-Nachweis und eine Nonce (Einmalwert, Zufallswert oder Zähler, der für den Proof-of-Work-Nachweis verwendet wird) umfassen [1][4][5]. Bei einem Datenblock kann es sich beispielsweise auch nur um einen bestimmten Speicherbereich oder Adressbereich der Gesamtdaten handeln, die in dem verteilten Datenbanksystem gespeichert sind. Damit lassen sich beispielsweise blocklose (engl. blockless) verteilte Datenbanksysteme, wie z. B. die IoT Chain (ITC), IOTA, und Byteball, realisieren. Hierbei werden insbesondere die Funktionalitäten der Blöcke einer Blockkette und der Transaktionen miteinander derart kombiniert, dass z. B. die Transaktionen selbst die Sequenz oder Kette von Transaktionen (des verteilten Datenbanksystems) absichern (also insbesondere sicherheitsgeschützt gespeichert werden). Hierzu können beispielsweise mit einer Verkettungsprüfsumme die Transaktionen selbst miteinander verkettet werden, indem vorzugsweise eine separate Prüfsumme oder die Transaktionsprüfsumme einer oder mehrerer Transaktionen als Verkettungsprüfsumme dient, die beim Speichern einer neuen Transaktion in dem verteilten Datenbanksystem in der entsprechenden neuen Transaktion mit gespeichert wird. In einer solchen Ausführungsform kann ein Datenblock beispielsweise auch eine oder mehrere Transaktionen umfassen, wobei im einfachsten Fall beispielsweise ein Datenblock einer Transaktion entspricht.

[0106] Unter "Nonce" kann im Zusammenhang mit der Erfindung beispielsweise eine kryptographische Nonce verstanden werden (Abkürzung für: "used only once"[2] oder "number used once"[3]). Insbesondere bezeichnet eine Nonce einzelne Zahlen- oder eine Buchstabenkombination, die vorzugsweise ein einziges Mal in dem jeweiligen Kontext (z. B. Transaktion, Datenübertragung) verwendet wird.

[0107] Unter "vorhergehende Datenblöcke eines (bestimmten) Datenblockes des verteilten Datenbanksystems" kann im Zusammenhang mit der Erfindung beispielsweise der Datenblock des verteilten Datenbanksystems verstanden werden, der insbesondere einem (bestimmten) Datenblock direkt vorhergeht. Alternativ können unter "vorhergehende Datenblöcke eines (bestimmten) Datenblockes des verteilten Datenbanksystems" insbesondere auch alle Datenblöcke des verteilten Datenbanksystems verstanden werden, die dem bestimmten Datenblock vorhergehen. Hierdurch kann beispielsweise die Verkettungsprüfsumme oder die Transaktionsprüfsumme insbesondere nur über das dem bestimmten Datenblock direkt vorhergehenden Datenblock (bzw. deren Transaktionen) oder über alle dem ersten

Datenblock vorhergehenden Datenblöcke (bzw. deren Transaktionen) gebildet werden.

[0108] Unter einem "Blockketten-Knoten", "Knoten", "Knoten eines verteilten Datenbanksystems", "Knoteneinrichtung" und dergleichen, können im Zusammenhang mit der Erfindung beispielsweise Geräte (z. B. Feldgeräte, Mobiltelefone), Rechner, Smart-Phones, Clients oder Teilnehmer verstanden werden, die Operationen (mit) dem verteilten Datenbanksystem (z. B. eine Blockkette) durchführen [1][4][5]. Solche Knoten können beispielsweise Transaktionen eines verteilten Datenbanksystems bzw. deren Datenblöcke ausführen oder neue Datenblöcke mit neuen Transaktionen in das verteilte Datenbanksystem mittels neuer Datenblöcke einfügen bzw. verketten. Insbesondere kann dieses Validieren und/oder Verketten durch einen vertrauenswürdigen Knoten (z. B. einem Mining Node) oder ausschließlich durch vertrauenswürdige Knoten erfolgen. Bei einem vertrauenswürdigen Knoten handelt es sich beispielsweise um einen Knoten, der über zusätzliche Sicherheitsmaßnahmen verfügt (z. B. Firewalls, Zugangsbeschränkungen zum Knoten oder Ähnliches), um eine Manipulation des Knotens zu verhindern. Alternativ oder zusätzlich kann beispielsweise ein vertrauenswürdiger Knoten beim Verketten eines neuen Datenblocks mit dem verteilten Datenbanksystem, eine Knotenprüfsumme (z. B. eine digitale Signatur oder ein Zertifikat) in dem neuen Datenblock speichern. Damit kann insbesondere ein Nachweis bereitgestellt werden, der angibt, dass der entsprechende Datenblock von einem bestimmten Knoten eingefügt wurde bzw. seine Herkunft angibt. Bei den Geräten (z. B. dem entsprechenden Gerät) handelt es sich beispielsweise um Geräte eines technischen Systems und/oder industriellen Anlage und/oder eines Automatisierungsnetzes und/oder einer Fertigungsanlage, die insbesondere auch ein Knoten des verteilten Datenbanksystems sind. Dabei können die Geräte beispielsweise Feldgeräte sein oder Geräte im Internet der Dinge sein, die insbesondere auch ein Knoten des verteilten Datenbanksystems sind. Knoten können beispielsweise auch zumindest einen Prozessor umfassen, um z. B. ihre computerimplementierte Funktionalität auszuführen.

[0109] Unter einem "Blockketten-Orakel" und dergleichen können im Zusammenhang mit der Erfindung beispielsweise Knoten, Geräte oder Rechner verstanden werden, die z. B. über ein Sicherheitsmodul verfügen, das beispielsweise mittels Software-Schutzmechanismen (z. B. kryptographische Verfahren), mechanische Schutzeinrichtungen (z. B. ein abschließbares Gehäuse) oder elektrische Schutzeinrichtungen implementiert ist (z. B. Tamper-Schutz oder ein Schutzsystem, das die Daten des Sicherheitsmoduls bei einer unzulässigen Nutzung/Behandlung des Blockketten-Orakels löscht). Das Sicherheitsmodul kann dabei beispielsweise kryptographische Schlüssel umfassen, die für die Berechnung der Prüfsummen (z. B. Transaktionsprüfsummen oder Knotenprüfsummen) notwendig sind.

[0110] Unter einem "Rechner" oder einem "Gerät" kann im Zusammenhang mit der Erfindung beispielsweise ein Computer(system), ein Client, ein Smart-Phone, ein Gerät oder ein Server, die jeweils außerhalb der Blockkette angeordnet sind bzw. kein Teilnehmer des verteilten Datenbanksystems (z. B. der Blockkette) sind (also keine Operationen mit dem verteilten Datenbanksystem durchführen oder diese nur abfragen, ohne jedoch Transaktionen durchzuführen, Datenblöcke einfügen oder Proof-of-Work-Nachweise berechnen), verstanden werden. Alternativ kann insbesondere auch unter einem Rechner ein Knoten des verteilten Datenbanksystems verstanden werden. Mit anderen Worten kann insbesondere unter einem Gerät ein Knoten des verteilten Datenbanksystems verstanden werden oder auch ein Gerät außerhalb der Blockkette bzw. des verteilten Datenbanksystems verstanden werden. Ein Gerät außerhalb des verteilten Datenbanksystems kann beispielsweise auf die Daten (z. B. Transaktionen oder Steuertransaktionen) des verteilten Datenbanksystems zugreift und/oder von Knoten (z. B. mittels Smart-Contracts und/oder Blockketten-Orakel) angesteuert werden. Wird beispielsweise eine Ansteuerung bzw. Steuerung eines Gerätes (z. B. ein als Knoten ausgebildetes Gerät oder ein Gerät außerhalb des verteilten Datenbanksystems) durch einen Knoten realisiert, kann dies z. B. mittels eines Smart Contracts erfolgen, der insbesondere in einer Transaktion des verteilten Datenbanksystems gespeichert ist.

[0111] Weitere mögliche Implementierungen der Erfindung umfassen auch nicht explizit genannte Kombinationen von zuvor oder im Folgenden bezüglich der Ausführungsbeispiele beschriebenen Merkmale oder Ausführungsformen. Dabei wird der Fachmann auch Einzelaspekte als Verbesserungen oder Ergänzungen zu der jeweiligen Grundform der Erfindung hinzufügen.

[0112] Weitere vorteilhafte Ausgestaltungen und Aspekte der Erfindung sind Gegenstand der Unteransprüche sowie der im Folgenden beschriebenen Ausführungsbeispiele der Erfindung. Im Weiteren wird die Erfindung anhand von bevorzugten Ausführungsformen unter Bezugnahme auf die beigelegten Figuren näher erläutert.

Fig. 1 veranschaulicht schematisch ein verteiltes Datenbanksystem gemäß einem ersten Ausführungsbeispiel sowie ein Verfahren zum Betreiben desselben;

Fig. 2 zeigt Details einer möglichen Ausgestaltung eines der Transaktionsbücher des verteilten Datenbanksystems;

Fig. 3 veranschaulicht schematisch ein verteiltes Datenbanksystem und Aspekte seines Betriebs gemäß einem zweiten Ausführungsbeispiel.

[0113] In den Figuren sind gleiche oder funktionsgleiche Elemente mit denselben Bezugszeichen versehen

worden, sofern nichts anderes angegeben ist.

[0114] Fig. 1 veranschaulicht schematisch ein verteiltes Datenbanksystem 1 gemäß einem ersten Ausführungsbeispiel sowie ein Verfahren zum Betreiben desselben.

[0115] Das verteilte Datenbanksystem 1 weist eine erste Datenbankinstanz 10 und eine zweite Datenbankinstanz 20 auf. Die erste Datenbankinstanz 10 ist durch die Knoteneinrichtungen 12, 13 und 14 gebildet. Die zweite Datenbankinstanz 20 ist durch die Knoteneinrichtungen 22, 23, 24 gebildet.

[0116] Die Knoteneinrichtungen 12, 13, 14 der ersten Datenbankinstanz 10 verwalten gemeinsam und konsensbasiert das Transaktionsbuch 11 des verteilten Datenbanksystems 1. Die Knoteneinrichtungen 22, 23, 24 der zweiten Datenbankinstanz 20 verwalten gemeinsam und konsensbasiert das Transaktionsbuch 21 des verteilten Datenbanksystems 1.

[0117] Wenn dem verteilten Datenbanksystem 1 eine zu bestätigende Transaktion 4 bereitgestellt wird, bestätigt in Schritt S1 des Verfahrens zum Betreiben des verteilten Datenbanksystems 1 das verteilte Datenbanksystem 1 die zu bestätigende Transaktion 4, indem sie die zu bestätigende Transaktion 4 als bestätigte Transaktion in eines oder beide der Transaktionsbücher 11, 21 aufnimmt.

[0118] Hierbei entscheidet das verteilte Datenbanksystem 1, durch welche der mehreren Datenbankinstanzen 10, 20 die zu bestätigende Transaktion 4 bestätigt wird. Anders ausgedrückt entscheidet das verteilte Datenbanksystems 1, ob die zu bestätigende Transaktion 4 von der ersten Datenbankinstanz 10 in das Transaktionsbuch 11 und/oder von der zweiten Datenbankinstanz 20 in das Transaktionsbuch 21 aufgenommen wird.

[0119] Insbesondere versteht sich, dass die Knoteneinrichtungen 12, 13, 14, 22, 23, 24 miteinander kommunikativ vernetzt sind, um die vorstehend beschriebene Funktionalität gemeinsam zu realisieren.

[0120] Insbesondere versteht sich weiterhin, dass die schematisch dargestellten Transaktionsbücher 11, 21 Repräsentationen eines auf verteilte Weise in dem verteilten Datenbanksystem 1 gespeicherten jeweiligen Transaktionsbuchs 11, 21 sind. Insbesondere kann auf jeder der Knoteneinrichtungen 12, 13, 14 eine Repräsentation des Transaktionsbuchs 11 gespeichert sein, wobei eine Konsensregel der Datenbankinstanz 10 dafür sorgt, dass die jeweiligen Repräsentationen ganz oder im Wesentlichen miteinander abgeglichen sind, und auf jeder der Knoteneinrichtungen 22, 23, 24 kann eine Repräsentation des Transaktionsbuchs 21 gespeichert sein, wobei eine Konsensregel der Datenbankinstanz 20 dafür sorgt, dass die jeweiligen Repräsentationen miteinander abgeglichen sind.

[0121] Fig. 2 zeigt Details einer möglichen Ausgestaltung eines der Transaktionsbücher 11, 21 des verteilten Datenbanksystems 1. Es wird weiter auch auf Fig. 1 Bezug genommen. Beispielhaft wird das Transaktionsbuch 11 der Datenbankinstanz 10 beschrieben, die Beschreibung gilt jedoch analog auch für das Transaktionsbuch 21 der Datenbankinstanz 20.

[0122] Der in Fig. 2 gezeigte Ausschnitt des Transaktionsbuchs 11 gemäß der möglichen Ausgestaltung umfasst drei Blöcke 101, 102, 103. Ein jeweiliger Block 101, 102, 103 umfasst jeweils einen Kopfdatenabschnitt K und einen Nutzdatenabschnitt N.

[0123] Der Kopfdatenabschnitt K des Blocks 101 umfasst eine Datenblockprüfsumme 1014, eine Verkettungsprüfsumme 1012 und einen Nachweiswert 1011. Analog dazu umfassen die Kopfdatenabschnitte K des Blocks 102 bzw. 103 jeweils eine Datenblockprüfsumme 1024, 1034, eine Verkettungsprüfsumme 1022, 1032 und einen Nachweiswert 1021, 1031.

[0124] Der Nutzdatenabschnitt N des jeweiligen Blocks 101, 102, 103 umfasst jeweils eine Anzahl bestätigte Transaktionen, die als abgerundete Rechtecke dargestellt sind. Eine der bestätigten Transaktionen trägt beispielhaft das Bezugszeichen 5.

[0125] Die Transaktionen 5 umfassen jeweils Nutzdaten des verteilten Datenbanksystems 1. Im Speziellen kann eine jeweilige Transaktion 5 Daten und/oder Programmcode (sog. Smart Contracts) umfassen, welche einen Übergang von einem Zustand, den das Transaktionsbuch 11 des verteilten Datenbanksystems 1 vor dem Bestätigen der Transaktion 5 beschreibt, in einen Zustand beschreibt, den das Transaktionsbuch 11 des verteilten Datenbanksystem 11 nach dem Bestätigen der Transaktion 5 beschreibt. Durch schrittweises Nachverfolgen der Blöcke 101, 102, 103 und der Transaktionen 5 darin kann der von dem Transaktionsbuch 11 beschriebene Zustand zu jedem aktuellen und vergangenen Zeitpunkt transparent nachvollzogen werden. Unter Zustand kann hierbei jede Art von Daten verstanden werden, die sich aus einer Folge von Transaktionen rekonstruieren lässt. Rein beispielhaft sei die Gesamtheit aller Kontostände in einer Anzahl von in der Datenbankinstanz 10 definierten Adressen bzw. Kryptowallets genannt. Denkbar sind jedoch beispielsweise auch Zustände wie Steuer- oder Schaltzustände von Aktoren eines industriellen Automatisierungssystems und dergleichen, wobei eine Transaktion einen jeweiligen Schaltvorgang repräsentiert.

[0126] Die jeweilige Datenblockprüfsumme 1014, 1024, 1034 ist insbesondere ein kryptographischer Hashwert, der die Transaktionen 5 des jeweiligen Blocks 101, 102, 103 gegen Manipulationen schützt. Insbesondere kann die Datenblockprüfsumme 1014, 1024, 1034 ein Wurzelwert eines Merkle- oder Patricia-Hashbaums sein.

[0127] Die jeweilige Verkettungsprüfsumme 1012, 1022, 1032 ist ein kryptographischer Hashwert des jeweils vorangehenden Blocks 101, 102. Insbesondere ist die Verkettungsprüfsumme 1022 ein kryptographischer Hashwert des gesamten Blocks 101. Der Verkettungs-Hashwert 1032 ist ein kryptographischer Hashwert des gesamten Blocks 62. Der jeweilige Verkettungs-Hashwert 1012, 1022, 1032 kann die Reihenfolge der Verket-

tung der Blöcke 101, 102, 103 definieren sowie das Transaktionsbuch 11 gegen Manipulationen sichern. Würde etwa die mit 5 bezeichnete der Transaktionen 5 nachträglich manipuliert, würde dadurch nicht nur die Datenblockprüfsumme 1014 invalidiert, sondern auch die Verkettungsprüfsumme 1022 und jede nachfolgende Verkettungsprüfsumme 1032.

[0128] Der jeweilige Nachweiswert 1011, 1021, 1031 ist ein Wert, der so aufgefasst werden kann, dass er dazu dient, ein berechtigtes Interesse derjenigen der Konteneinrichtungen 12-14, die den jeweiligen Block 101, 102, 103 gebildet hat (im Weiteren "blockbildende Knoteneinrichtung"), an der Aufnahme des Blocks 101, 102, 103 in das Transaktionsbuch 11 zu dokumentieren. Der Nachweiswert 1011, 1021, 1031 ist insbesondere derart eingerichtet, dass er auf nachprüfbare Weise eine Menge von durch die blockbildende Knoteneinrichtung 12-14 aufgewandter Rechenleistung (sog. Proof-of-Work), eine für eine bestimmte Dauer vorgehaltene Menge an Kryptotoken (sog. Proof-of-Stake), eine Menge anderweitig eingesetzter Ressourcen und/oder eine Berechtigung wie etwa eine Signatur eines Privileged Ledger dokumentiert.

[0129] Das Bestätigen einer zu bestätigenden Transaktion 4 durch die Datenbankinstanz 10 kann insbesondere wie nachstehend beschrieben ablaufen. Hierbei wird davon ausgegangen, dass der in Fig. 2 gezeigte Block 103 zu Beginn des Vorgangs noch nicht Teil des Transaktionsbuchs 11 ist, das Transaktionsbuch 11 also zu Beginn des nun beschriebenen Vorgangs nur aus den bestätigten Blöcken 101 und 102 besteht.

[0130] Die blockbildende Knoteneinrichtung, beispielsweise die Knoteneinrichtung 12, bildet den, zu diesem Zeitpunkt noch unbestätigten, Block 103, prüft die zu bestätigende Transaktion 4 auf Gültigkeit, nimmt die zu bestätigende Transaktion 4 bzw. eine Kopie davon als bestätigte Transaktion in den unbestätigten Block 103 auf, bestimmt die Datenblockprüfsumme 1034 des unbestätigten Blocks 13, verkettet den unbestätigten Block 103 mit dem letzten bestätigten Block 102 des Transaktionsbuchs 11, wozu sie die Verkettungsprüfsumme 1032 des unbestätigten Blocks auf den kryptographischen Hashwert des letzten bestätigten Blocks 102 des Transaktionsbuchs 11 setzt, und bestimmt den Nachweiswert 1031 des unbestätigten Blocks 103. Der unbestätigte Block 103 wird dadurch in der auf der blockbildenden Knoteneinrichtung 12 gespeicherten Repräsentation des Transaktionsbuchs 11 zum neuen letzten Block 103 des Transaktionsbuchs 11. Wenn auf diese Weise ein unbestätigter Block erfolgreich gebildet ist, stellt die blockbildende Knoteneinrichtung 12 den unbestätigten Block 103 den anderen Knoteneinrichtungen 13, 14 derselben Datenbankinstanz 10 bereit. Diese prüfen den unbestätigten Block 103 und fügen ihn, sofern die Prüfung erfolgreich ist, ebenso als neuen letzten Block 103 des Transaktionsbuchs 11 an die in ihnen jeweils gespeicherte Repräsentation des Transaktionsbuchs 11 an.

[0131] Stellt sich in dieser Weise ein Mehrheitskonsens ein, gelten insbesondere der Block 103 und alle darin gespeicherten Transaktionen als von der Datenbankinstanz 10 bestätigt.

[0132] Die erwähnten Prüfungsvorgänge können dabei insbesondere eine Prüfung umfassen, ob die zu bestätigende Transaktion 4 des unbestätigten Blocks 103 einen gültigen Zustandsübergang beschreibt. Hierbei kann Programmcode eines von der zu bestätigenden Transaktion 4 umfassten oder referenzierten Smart Contracts zur Ausführung gelangen. Ferner können die erwähnten Prüfungsvorgänge eine Prüfung der Datenblockprüfsumme 1034 auf Richtigkeit, der Verkettungsprüfsumme 1032 auf Richtigkeit, und eine Prüfung des Nachweiswerts 1031 auf Übereinstimmung mit den Anforderungen der Konsensregel der Datenbankinstanz 10 umfassen.

[0133] Insbesondere kann die Anforderung der Konsensregel der Datenbankinstanz 10, dass ein jeweiliger Block 101, 102, 103 einen solchen Nachweiswert 1011, 1021, 1031 enthalten soll, das Bilden eines korrekten, der Konsensregel entsprechenden Blocks 101-103 erschweren bzw. verteuern. Dies kann dem Manipulationsschutz dienen, da ein nachträgliches Verändern des Transaktionsbuchs 11 auch ein erneutes ressourcenaufwändiges Bestimmen veränderter Nachweiswerte 1011, 1021, 1031 erforderlich machen kann.

[0134] Die Beschreibung der möglichen Ausgestaltung erfolgte anhand der Datenbankinstanz 10 und des Transaktionsbuchs 11, gilt jedoch für die Datenbankinstanz 20 und das Transaktionsbuch 21 analog.

[0135] Angesichts des Vorstehenden wird deutlich, dass das Bestätigen der zu bestätigenden Transaktion 4 ein rechenaufwändiger Vorgang sein kann. Insbesondere kann die genaue Menge des Rechenaufwands für das Bestätigen einer jeweiligen zu bestätigenden Transaktion 4 schwer planbar sein, da verschiedene zu bestätigende Transaktionen 4 Smart Contracts von unterschiedlicher Komplexität umfassen können.

[0136] Somit kann es in einer der Datenbankinstanzen 11, 12 zu einer vorübergehenden Überlastsituation mit einer zu hohen Transaktionslast kommen.

[0137] In dem verteilten Datenbanksystem 1 gemäß der möglichen Ausgestaltung wird dem damit begegnet, dass das verteilte Datenbanksystem 1 entscheiden kann, die zu bestätigende Transaktion 4 entweder in dem Transaktionsbuch 11 der Datenbankinstanz 10 oder in dem Transaktionsbuch 12 der Datenbankinstanz 20 zu bestätigen.

[0138] Im Speziellen tauschen gemäß einer Weiterbildung des ersten Ausführungsbeispiels die Knoteneinrichtungen 12-14, 22-24 untereinander Zustandsinformationen über den Zustand der jeweiligen Datenbankinstanzen 10, 20 aus. Die Zustandsinformationen können jede Art von geeigneten Zustandsinformationen umfassen, beispielsweise eine Rechenleistung einer jeweiligen der Knoteneinrichtungen 12-14, 22-24 der jeweiligen Datenbankinstanz 10, 20, eine Größe des jeweiligen Trans-

aktionsbuchs 11, 21, ein mittlerer Transaktionsdurchsatz der jeweiligen Datenbankinstanz 10, 20 und dergleichen.

[0139] Demgemäß kann eine jede der Knoteneinrichtungen 12-14, 22-24 über den gegenwärtigen Zustand jeder der Datenbankinstanzen 11, 21 informiert sein. Die Knoteneinrichtungen 12-14, 22-24 der jeweiligen Datenbankinstanz 10, 20 können eingerichtet sein, die zu bestätigende Transaktion 4 dann zu bestätigen, wenn die Zustandsinformationen über den Zustand der Datenbankinstanz 10, 20 im Vergleich zu den Zustandsinformationen über den Zustand der übrigen Datenbankinstanzen 10, 20 eine bestimmte Bedingung erfüllt.

[0140] Beispielsweise können nur die Knoteneinrichtungen 12-14, 22-24 derjenigen Datenbankinstanz 10, 20 die zu bestätigende Transaktion 4 bestätigen, die unter den Datenbankinstanzen 10, 20 die geringste Last, die geringste Transaktionsbuchgröße, den geringsten Transaktionsdurchsatz oder dergleichen aufweist.

[0141] Somit kann in den Datenbankinstanzen 10, 20 gemäß dem ersten Ausführungsbeispiel vorteilhaft ein automatischer Lastausgleich realisiert werden.

[0142] Insbesondere kann erreicht werden, dass Transaktionen zeitnah bestätigt werden, unabhängig davon, ob einzelne Datenbankinstanzen 10, 20 temporär überlastet sind. Weiterhin kann ein Datenbanksystem 1 mit hoher Resilienz realisiert werden. Es können Transaktionen durch das Datenbanksystem 1 bestätigt werden, selbst wenn eine der Datenbankinstanzen 10, 20 ausgefallen ist.

[0143] Fig. 3 veranschaulicht schematisch ein verteiltes Datenbanksystem 1 und Aspekte seines Betriebs gemäß einem zweiten Ausführungsbeispiel.

[0144] Das zweite Ausführungsbeispiel ist eine Weiterbildung des ersten Ausführungsbeispiels, so dass nachstehend vornehmlich auf Unterschiede und zusätzliche Merkmale und Ausgestaltungen eingegangen wird, um redundante Beschreibungen zu vermeiden. Das verteilte Datenbanksystem 1 gemäß dem zweiten Ausführungsbeispiel umfasst drei Datenbankinstanzen 10, 20, 30. Die jeweils zugehörigen Knoteneinrichtungen (vgl. Knoteneinrichtungen 12-14, 22-24 in Fig. 1) sind in Fig. 3 nicht dargestellt. Insofern nachstehend Funktionalität einer der Datenbankinstanzen 10, 20, 30 beschrieben ist, ist dies insbesondere so zu verstehen, dass diese Funktionalität durch die jeweiligen Knoteneinrichtungen (nicht gezeigt) der jeweiligen Datenbankinstanz 10, 20, 30 implementiert sein kann.

[0145] Fig. 3 veranschaulicht weiterhin entlang einer gedachten, von links nach rechts verlaufenden Zeitachse, die Transaktionsbücher 11, 12, 13 der Datenbankinstanzen 10, 20 bzw. 30.

[0146] Die erste Datenbankinstanz 10 ist eine Hauptkette (engl. "main chain"), und bei dem Transaktionsbuch 11 der ersten Datenbankinstanz 10 handelt es sich demgemäß um ein Hauptbuch (engl. "main ledger"). Die zweite und die dritte Datenbankinstanz 20, 30 sind eine jeweilige Seitenkette (engl. "side chain"), und bei den Transaktionsbüchern 21, 31 der jeweiligen Seitenkette 20, 30 handelt es sich demgemäß um ein jeweiliges Seitenbuch (engl. "side ledger").

[0147] Das Hauptbuch 11 weist insbesondere einen höheren Manipulationsschutz, jedoch eine niedrigere Blockbildungsrade als die Seitenbücher 21, 31 auf. Beispielsweise kann in der Datenbankinstanz 10 für das Hauptbuch 11 ein aufwändiger Proof-of-Work im Rahmen der Konsensregel verwendet werden, während bei den Datenbankinstanzen 20, 30 für die Seitenbücher 21, 31 vorzugsweise ein schneller zu generierender, aber weniger sicherer Nachweiswert verwendet wird. Denkbar ist hier beispielsweise ein vertrauensbasierter Privileged-Ledger-Ansatz.

[0148] Wie in Fig. 3 angedeutet, sind die Seitenbücher 21, 31 mehrfach mit dem Hauptbuch 11 kryptographisch verknüpft. Insbesondere entspricht zu dem Zeitpunkt, zu dem im Hauptbuch 11 der Block 101 bestätigt wird, ein von dem Hauptbuch 11 repräsentierter Zustand - wie etwa eine Gesamtheit von Schaltzuständen, Kontoständen und dergleichen - einem von jedem der Seitenbüchern 21, 31 repräsentierten Zustand. Dieser gemeinsame Zustand sei im Folgenden als Ausgangszustand bezeichnet. Die zu einem späteren Zeitpunkt in den Blöcken 201-204 des ersten Seitenbuchs 21 bestätigten Transaktionen schreiben - durch die von den bestätigten Transaktionen beschriebenen Zustandsübergänge - den Ausgangszustand fort. Anders ausgedrückt ist der erste Block 201 des ersten Seitenbuchs 21 mittels einer (in Fig. 3 nicht gezeigten) Verkettungsprüfsumme mit dem Block 101 des Hauptbuchs 11 kryptographisch verkettet bzw. verknüpft. Desgleichen schreiben die in den Blöcken 301-304 des zweiten Seitenbuchs 31 beschriebenen Zustandsübergänge den Ausgangszustand fort. Ab dem Zweitpunkt, wo in dem Seitenbuch 21 und dem Seitenbuch 31 ein jeweiliger erster Block 201, 301 bestätigt worden ist, unterscheidet sich mithin der von dem ersten Seitenbuch 21 beschriebene erste fortgeschriebene Zustand von dem von dem zweiten Seitenbuch 31 beschriebenen zweiten fortgeschriebenen Zustand. Ab diesem Zeitpunkt kann eine zu bestätigende Transaktion 4, die auf den durch die Blockfolge 101, 201 des ersten Seitenbuchs 21 definierten ersten fortgeschriebenen Zustand Rückbezug nimmt, nicht mehr ohne Weiteres in dem zweiten Seitenbuch 31 bestätigt werden. Aus diesem Grund muss in einer herkömmlichen Architektur mit mehreren Transaktionsbüchern eine jede zu bestätigende Transaktion vorab eindeutig spezifizieren, in welchem der Transaktionsbücher sie zu spezifizieren ist. Gemäß der vorgeschlagenen Lösung hat jedoch das Datenbanksystem 1 mindestens eine gewisse Entscheidungsfreiheit darüber, in welchem der Seitenbücher 21, 31 oder dem Hauptbuch 11 eine zu bestätigende Transaktion 4 bestätigt wird. Dies wird nachfolgend näher erläutert.

[0149] In dem verteilten Datenbanksystem 1 kann ein Pool 40 aus mehreren unbestätigten Transaktionen 41-45 vorgehalten werden. Die jeweilige unbestätigte Transaktion 41-45 kann von einer Anwendung, die das zentrale Datenbanksystem 1 nutzt, in eine nicht gezeigte

zentrale Vorhalteeinrichtung des Datenbanksystems 1 geschrieben oder aber an eine beliebige der nicht gezeigten Knoteneinrichtungen des Dantebanksystems 1 übermittelt und von dort auf Peer-to-Peer-Weise an die übrigen (nicht gezeigten) Knoteneinrichtungen des Datenbanksystems 1 weiterübermittelt werden. Anders ausgedrückt kann der Pool 40 zentral mittels der Vorhalteeinrichtung oder dezentral mittels Peer-to-Peer-Kommunikation oder dergleichen zwischen den Knoteneinrichtungen des Datenbanksystems 1 implementiert sein.

[0150] Das verteilte Datenbanksystem 1 wählt nacheinander jeweils eine der unbestätigten Transaktionen 41-45 aus dem Pool 40 als die aktuelle zu bestätigende Transaktion 4 aus und bestätigt die jeweilige zu bestätigende Transaktion 4 in jeweils einem oder mehreren der Transaktionsbücher 11, 21, 31.

[0151] Die Entscheidung, in welchem der Transaktionsbücher 11, 21, 31 die zu bestätigende Transaktion 4 bestätigt wird, kann auf mehrere mögliche Weisen getroffen werden.

[0152] In einer Variante kann jeder der Knoteneinrichtungen jeder der Datenbankinstanzen 10, 20, 30 mit dem Bilden eines unbestätigten Blocks (nicht gezeigt) beginnen, in den die zu bestätigende Transaktion 4 aufgenommen wird. Jedoch kann eine Konsensregel jeder der Datenbankinstanzen 10, 20, 30 unter anderem vorsehen, dass eine zu bestätigende Transaktion 4 nur dann als gültig zu betrachten ist, wenn diese in keinem der Transaktionsbücher 11, 21, 31 des verteilten Datenbanksystems 1 bereits bestätigt ist. Auf diese Weise kann sich diejenige der Datenbankinstanzen 10, 20, 30 durchsetzen, die die zu bestätigende Transaktion 4 am schnellsten bestätigen kann.

[0153] In einer weiteren Variante kann in einer jeweiligen unbestätigten Transaktion 41-45 des Pools 40 eine Anzahl von Datenbankinstanzen 10, 20, 30 spezifiziert sein, in welcher die unbestätigte Transaktion 41-45 zu bestätigen ist. Wenn eine der Datenbankinstanzen 10, 20, 30 die jeweilige unbestätigte Transaktion 41-45 als zu bestätigende Transaktion 4 auswählt und bestätigt, kann sie die spezifizierte Anzahl von Datenbankinstanzen um eins dekrementieren. Wird die Anzahl dabei auf null dekrementiert, kann die unbestätigte Transaktion 41-45, die als zu bestätigende Transaktion 4 ausgewählt worden ist, aus dem Pool 40 entfernt werden. Auf diese Weise wird die ausgewählte unbestätigte Transaktion 41-45 von keiner weiteren der Datenbankinstanzen 10, 20, 30 mehr bestätigt.

[0154] Ein derartiges mehrfaches Bestätigen einer zu bestätigenden Transaktion 4 durch eine Anzahl von Datenbankinstanzen 10, 20, 30 hat den Vorteil, dass ein besonders hoher Manipulationsschutz erreicht wird. Eine Manipulation einer Transaktion in nur einer der Datenbankinstanzen 10, 20, 30 kann durch einen Vergleich der mehreren Datenbankinstanzen 10, 20, 30 erkannt werden. Ein weiterer Vorteil des mehrfachen Bestätigens kann darin bestehen, dass ein Rückbezug auf die in mehreren der Datenbankinstanzen 10, 20, 30 bestätigte Transaktion durch künftige, nachfolgende Transaktionen (nicht dargestellt) in diesen mehreren Datenbankinstanzen 10, 20, 30 möglich ist. Dadurch wird eine höhere Flexibilität erreicht, in welchen der mehreren Datenbankinstanzen 10, 20, 30 die nachfolgende unbestätigte Transaktion bestätigbar ist.

[0155] In einer weiteren Variante kann die Konsensregel der jeweiligen Datenbankinstanz 10, 20, 30 vorsehen, dass eine Vergütung für das Bestätigen der zu bestätigenden Transaktion 4 nur eine Anzahl von Malen gewährbar ist, die in der jeweiligen zu bestätigenden Transaktion 4 spezifiziert ist, bzw. dass die Vergütung mit jedem Bestätigen der zu bestätigenden Transaktion 4 in einem der Transaktionsbücher 11, 21, 31 dekrementiert wird. Sinkt dabei die Vergütung auf null ab, besteht für die Knoteneinrichtungen weiterer der Datenbankinstanzen 10, 20, 30 kein Anreiz mehr, die zu bestätigende Transaktion 4 in weiteren der Transaktionsbücher 11, 21, 31 zu bestätigen, und ein solches weiteres Bestätigen kann unterbleiben.

[0156] In einer dritten Variante können die Datenbankinstanzen 10, 20, 30 Informationen über ihre jeweilige Lastsituation (Verarbeitungslast der einzelnen Knoteneinrichtungen der jeweiligen Datenbankinstanz 10, 20, 30; Transaktionslast der jeweiligen Datenbankinstanz 10, 20, 30 oder Speichergröße des jeweiligen Transaktionsbuchs 11, 21, 31 und dergleichen) austauschen und derart eingerichtet sein, dass von vornherein nur diejenige Datenbankinstanz 10, 20, 30 die zu bestätigende Transaktion 4 zum Bestätigen auswählt, die aktuell die niedrigste Last aufweist.

[0157] In allen geschilderten Varianten kann somit eine automatische dezentrale Konsensbildung zwischen den mehreren Datenbankinstanzen darüber erfolgen, in welcher der Datenbankinstanzen 10, 20, 30 die zu bestätigende Transaktion 4 bestätigt wird.

[0158] Es wird nun im Detail das Bestätigen der unbestätigten Transaktionen 41-45 beschrieben.

[0159] Die erste Transaktion 41 ist eine Orakeltransaktion. Die Orakeltransaktion 41 ist insbesondere frei von Rückbezügen auf vergangene bestätigte Transaktionen, vielmehr enthält die Orakeltransaktion eine Information über die reale Welt, wie beispielsweise einen Messwert, der in dem verteilten Datenbanksystem 1 bekannt gemacht werden soll.

[0160] Die unbestätigte Orakel-Transaktion 41 wird von dem verteilten Datenbanksystem 1 sowohl als bestätigte Orakel-Transaktion 511 in dem ersten Seitenbuch 21 als auch als bestätigte Orakel-Transaktion 512 des zweiten Seitenbuchs 22 bestätigt. Damit wird jedem der Seitenbücher 21, 31 die Information über die reale Welt bekannt gemacht.

[0161] Ein solches Bestätigen einer zu bestätigenden Transaktion 4 in jedem der Seitenbücher 21, 31 kann beispielsweise das standardmäßig vorgegebene Verhalten des verteilten Datenbanknetzwerks 1 bei Orakel-Transaktionen sein. Alternativ kann die Orakel-Transaktion 41 explizit spezifizieren, dass sie in genau zwei oder

in mindestens zwei Transaktionsbüchern 11, 21, 31 zu bestätigen ist.

[0162] Die Transaktion 42 ist eine erste Kryptotoken-Transaktion, die eine Menge von Kryptotoken an eine Adresse (auch "Output" genannt), wie beispielsweise 0x4EAC, eines Kryptowallets transferiert. Die Transaktion 42 spezifiziert mittels darin umfassten Spezifikationsdaten (nicht gezeigt), dass die Transaktion 42 in genau einem der Seitenbücher 21, 31 zu bestätigen ist, um auf diese Weise eine schnellere Abwicklung als in dem Hauptbuch 11 zu erzielen.

[0163] Die unbestätigte erste Kryptotoken-Transaktion 42 wird von dem verteilten Datenbanksystem 1 basierend auf einer aktuellen Lastverteilung der Datenbankinstanzen 21 und 31 beispielsweise, wie in der Fig. 3 gezeigt, in dem zweiten Block 302 des zweiten Seitenbuchs 31 bestätigt. Sie könnte je nach der aktuellen Lasterteilung alternativ auch in einem der Blöcke des ersten Seitenbuchs 21 bestätigt werden. Die unbestätigte erste Kryptotoken-Transaktion 42 wird jedoch nicht in dem Hauptbuch 11 bestätigt. Das heißt, das verteilte Datenbanksystem 1 berücksichtigt bei der Entscheidung, durch welche der mehreren Datenbankinstanzen 10, 20, 30 die zu bestätigende Transaktion 4 (die unbestätigte Kryptotoken-Transaktion 42) bestätigt wird, nur die Datenbankinstanzen 20, 30, die von der zu bestätigenden Transaktion 42 spezifiziert sind.

[0164] Die unbestätigte Transaktion 43 ist eine zweite Kryptotoken-Transaktion, die die Menge von Kryptotoken von der Adresse 0x4EAC an eine zweite Adresse 0x81F2 transferiert. Die Transaktion 43 enthält daher einen Rückbezug auf die bestätigte Transaktion 52, welche als Nachweis dient, dass die Absenderadresse 0x4AEC (auch "unspent output" genannt) die Menge an Kryptotoken tatsächlich enthält und seitdem nicht anderweitig ausgegeben wurde. Auch die unbestätigte zweite Kryptotoken-Transaktion 43 spezifiziert, dass die unbestätigte Transaktion 43 in genau einem der Seitenbücher 21, 31 zu bestätigen ist.

[0165] Dieser Rückbezug der unbestätigten Transaktion 43 auf die bestätigte Transaktion 52 ist nur in dem zweiten Seitenbuch 31 auflösbar, in dem die bestätigte Transaktion 52 in Block 302 umfasst ist. Dieser Rückbezug wird von dem verteilten Datenbanksystem 1 beim Bestätigen der unbestätigten Transaktion 43 berücksichtigt, und die unbestätigte Transaktion 43 wird als die bestätigte Transaktion 53 in dem Block 304 des zweiten Transaktionsbuchs 31 bestätigt.

[0166] Die unbestätigte Transaktion 44 ist eine dritte unbestätigte Kryptotoken-Transaktion, die die Menge von Kryptotoken von der zweiten Adresse 0x81F2 an eine dritte Adresse 0x99EA transferiert und somit einen direkten Rückbezug auf die bestätigte Transaktion 53 enthält, welche das Vorhandensein der Kryptotoken an der Absenderadresse (zweiten Adresse) 0x81F" dokumentiert, sowie einen indirekten Rückbezug auf die bestätigte Transaktion 52, auf welche die bestätigte Transaktion 53 Rückbezug nimmt.

[0167] Grundsätzlich wäre daher die dritte unbestätigte Kryptotoken-Transaktion 44 auch in dem zweiten Transaktionsbuch 31 zu bestätigen, in dem die rückbezogene bestätigte Transaktion 53 in Block 304 bestätigt ist.

[0168] Um jedoch den Vorteil der automatisierten Lastverteilung besser zur Geltung zu bringen, synchronisiert das verteilte Datenbanksystem 1 in vordefinierten Zeitabständen die Transaktionsbücher 11, 21, 31 der mehreren Datenbankinstanzen 10, 20, 30. Die vordefinierten Zeitabstände sind zum Beispiel durch die Zeitabstände zwischen dem Bilden eines jeweiligen Blocks 101, 102 des Hauptbuchs 11 definiert.

[0169] Im Speziellen umfasst der Block 102 des Hauptbuchs 11 eine zusammenfassende Transaktion 56, die den von dem ersten Seitenbuch 21 fortgeschriebenen Zustand (das Ergebnis des Abwickelns aller Transaktionen, die in dem ersten Seitenbuch 21 in der Zeit zwischen dem Bestätigen des ersten Blocks 101 des Hauptbuchs 11 und dem Bestätigen des zweiten Blocks 102 des Hauptbuchs 11 bestätigt worden sind) definiert, und eine zusammenfassende Transaktion 57, die den von dem zweiten Seitenbuch 31 fortgeschriebenen Zustand definiert. Somit wird durch das Bestätigen des zweiten Blocks 102 im Hauptbuch 11, der die zusammenfassenden Transaktionen 56 und 57 umfasst, ein neuer gemeinsamer Ausgangszustand des Hauptbuchs 11, des ersten Seitenbuchs 21 und des zweiten Seitenbuchs 31 geschaffen.

[0170] Der nächste Block 205 des Seitenbuchs 21 wird daher kryptographisch nicht mit dem vorhergehenden Block 204 des Seitenbuchs 21, sondern mit dem zweiten Block 102 des Hauptbuchs 11 verknüpft. Entsprechendes gilt für den nächsten Block 305 des Seitenbuchs 31.

[0171] Demgemäß kann die dritte unbestätigte Kryptotoken-Transaktion 44, deren Bestätigen erst nach dem Bestätigen des Blocks 102 des Hauptbuchs 11 beginnt, trotz ihres Rückbezugs auf die in dem zweiten Transaktionsbuch 31 bestätigte Transaktion 53 in jedem der Seitenbücher 21, 31 bestätigt werden und wird beispielsweise, wie in Fig. 3 gezeigt, nun aufgrund einer veränderten Lastsituation in dem ersten Seitenbuch 21 als die bestätigte Transaktion 54 des sechsten Blocks 206 bestätigt.

[0172] Somit kann vorteilhaft auch bei unbestätigten Transaktionen 43, 44 mit Rückbezügen auf ein konkretes der Transaktionsbücher 11, 21, 31 in vordefinierten Zeitabständen durch das verteilte Datenbanksystem 1 eine automatisierte Lastverteilung durch Bestätigen in einem anderen der Transaktionsbücher 11, 21, 31 erfolgen.

[0173] Die unbestätigte Transaktion 45 schließlich spezifiziert, dass sie, beispielsweise aufgrund erhöhter Sicherheitsanforderungen, nur in dem Hauptbuch 11 bestätigt werden darf. Dies wird von dem verteilten Datenbanksystem 1 berücksichtigt, und die unbestätigte Transaktion 45 wird als bestätigte Transaktion 55 unmittelbar in dem zweiten Block 102 des Hauptbuchs 11 bestätigt. Dies kann auch so verstanden werden, dass die vorgeschlagene Lösung, die Entscheidung über die be-

stätigende Datenbankinstanz 10, 20, 30 dem verteilten Datenbanksystem 1 zu überlassen, für bestimmte unbestätigte Transaktionen 45 außer Kraft gesetzt werden kann, sofern dies im Einzelfalle zweckmäßig erscheint.

[0174] Obwohl die vorliegende Erfindung anhand von Ausführungsbeispielen beschrieben wurde, ist sie vielfältig modifizierbar.

[0175] Anhand des zweiten Ausführungsbeispiels wurde beschrieben, dass die von den jeweiligen Transaktionsbüchern 11, 21, 31 bechriebenen Zustände in vordefinierten Zeitabständen miteinander synchronisiert werden. Es ist jedoch auch denbkar, dass die Datenbankinstanzen 10, 20, 30 zu einer datenbankinstanzübergreifenden Kommunikation beim Bestätigen der jeweiligen zu bestätigenden Transaktion 4 eingerichtet sind. Anders ausgedrückt kann jede der Knoteneinrichtungen (in Fig. 3 nicht gezeigt) jeder der Datenbankinstanzen 10, 20, 30 mindestens über Lesezugriff auf jedes der Transaktionsbücher 11, 21, 31 verfügen. Somit kann zu jedem Zeitpunkt von der Gesamtheit der Transaktionsbücher 11, 21, 31 ein gemeinsamer Zustand repräsentiert werden, der von jeder der Datenbankinstanzen 10, 20, 30 beim Bestätigen der jeweiligen zu bestätigenden Transaktion 4 überprüfbar ist. In diesem Fall kann es möglich sein, jede Transaktion mit Rückbezug, wie die unbestätigten Transaktionen 43, 44 in jedem der Transaktionsbücher 11, 21, 31 zu bestätigen, ohne dass eine Synchronisation der mehreren Datenbankinstanzen 10, 20, 30 in vordefinierten Zeitabständen erforderlich ist.

[0176] Anhand der Ausführungsbeispiele wurde ein stark vereinfachtes verteiltes Datenbanksystem 1 beschrieben, und es wurden lediglich die Grundzüge der Funktionsweise der Blockketten-Technologie angerissen. Es versteht sich, dass der von den Ansprüchen definierte Gedanke der Erfindung auch auf beliebige Weiterbildungen von verteilten Datenbanksystemen und Blockketten-Datenbanken mit mehreren Transaktionsbüchern anwendbar ist, insbesondere auch auf solche, wie sie in den Referenzen [1] bis [8] genannt sind.

[0177] Die Erfindung kann allgemein dahingehend verstanden werden, dass es mindestens teilweise einem verteilten Datenbanksystem 1 überlassen bleibt, in welchem von mehreren Transaktionsbüchern 11, 21, 31 des verteilten Datenbanksystems 1 eine zu bestätigende Transaktion 4 bestätigt wird.

Referenzen

[0178]

[1]
Andreas M. Antonopoulos "Mastering Bitcoin: Unlocking Digital Cryptocurrencies", O'Reilly Media, December 2014

[2]
Roger M. Needham, Michael D. Schroeder "Using encryption for authentication in large networks of computers" ACM: Communications of the ACM. Band 21, Nr. 12 Dezember 1978,

[3]
Ross Anderson "Security Engineering. A Guide to Building Dependable Distributed Systems" Wiley, 2001

[4]
Henning Diedrich "Ethereum: Blockchains, Digital Assets, Smart Contracts, Decentralized Autonomous Organizations", CreateSpace Independent Publishing Platform, 2016

[5]
"The Ethereum Book Project/Mastering Ethereum" https://github.com/ethereumbook/ethereumbook, Stand 5.10.2017

[6]
Leemon Baird "The Swirlds Hashgraph Consensus Algorithm: Fair, Fast, Byzantine Fault Tolerance", Swirlds Tech Report SWIRLDS-TR-2016-01, 31.5.2016

[7]
Leemon Baird "Overview of Swirlds Hashgraph", 31.5.2016

[8]
Blockchain Oracles
https://blockchainhub.net/blockchain-oracles/

Patentansprüche

1. Verteiltes Datenbanksystem (1) mit mehreren Datenbankinstanzen (10, 20),
wobei eine jeweilige der mehreren Datenbankinstanzen (10, 20) durch eine Anzahl Knoteneinrichtungen (12, 13, 14; 22, 23, 24) gebildet ist, die dazu eingerichtet sind, gemeinsam konsensbasiert eines von mehreren Transaktionsbüchern (11, 21) des verteilten Datenbanksystems (1) zu verwalten,
wobei das verteilte Datenbanksystem (1) eingerichtet ist, eine zu bestätigende Transaktion (4) durch Aufnehmen in das Transaktionsbuch (11, 21) einer oder mehrerer der mehreren Datenbankinstanzen (10, 20) zu bestätigen,
wobei das verteilte Datenbanksystem (1) dazu eingerichtet ist, zu entscheiden, durch welche der mehreren Datenbankinstanzen (10, 20) die zu bestätigende Transaktion (4) bestätigt wird.

2. Verteiltes Datenbanksystem nach Anspruch 1,
dadurch gekennzeichnet,
dass das verteilte Datenbanksystem (1) dazu eingerichtet ist, in Abhängigkeit von einem Zustand der

jeweiligen Datenbankinstanz (10, 20) zu entscheiden, durch welche der mehreren Datenbankinstanzen (10, 20) die zu bestätigende Transaktion (4) bestätigt wird.

3. Verteiltes Datenbanksystem nach Anspruch 1 oder 2 **dadurch gekennzeichnet,**
dass das verteilte Datenbanksystem (1) dazu eingerichtet ist, für jede der Datenbankinstanzen (10, 20) eine Last zu ermitteln und zu entscheiden, dass die zu bestätigende Transaktion (4) durch die eine oder die mehreren der Datenbankinstanzen (10, 20) mit der geringsten Last bestätigt wird.

4. Verteiltes Datenbanksystem nach einem der Ansprüche 1 bis 3,
dadurch gekennzeichnet,
dass das verteilte Datenbanksystem (1) dazu eingerichtet ist, in Abhängigkeit von einem oder mehreren Rückbezügen der zu bestätigenden Transaktion (4) zu entscheiden, durch welche der mehreren Datenbankinstanzen (10, 20, 30) die zu bestätigende Transaktion (4) bestätigt wird.

5. Verteiltes Datenbanksystem nach einem der Ansprüche 1 bis 4,
dadurch gekennzeichnet,
dass das verteilte Datenbanksystem (1) dazu eingerichtet ist, bei der Entscheidung, durch welche der mehreren Datenbankinstanzen die zu bestätigende Transaktion (4) bestätigt wird, eine jeweilige Datenbankinstanz (10, 20, 30) nur zu berücksichtigen, wenn die Datenbankinstanz (10, 20, 30) von der zu bestätigenden Transaktion (4) spezifiziert ist.

6. Verteiltes Datenbanksystem nach einem der Ansprüche 1 bis 5,
dadurch gekennzeichnet,
dass das verteilte Datenbanksystem (1) dazu eingerichtet ist, durch eine automatische dezentrale Konsensbildung zwischen den mehreren Datenbankinstanzen (10, 20, 30) zu entscheiden, durch welche der mehreren Datenbankinstanzen (10, 20, 30) die zu bestätigende Transaktion (4) bestätigt wird.

7. Verteiltes Datenbanksystem nach einem der Ansprüche 1 bis 6,
dadurch gekennzeichnet,
dass das verteilte Datenbanksystem (1) dazu eingerichtet ist, eine Anzahl (40) unbestätigter Transaktionen (41-45) vorzuhalten,
wobei eine jeweilige der mehreren Datenbankinstanzen (10, 20, 30) dazu eingerichtet ist, die zu bestätigende Transaktion unter der vorgehaltenen Anzahl (40) unbestätigter Transaktionen (41-45) auszuwählen und zu bestätigen,
wobei das verteilte Datenbanksystem (1) dazu eingerichtet ist, die zu bestätigende Transaktion (4) aus der vorgehaltenen Anzahl (40) unbestätigter Transaktionen (41-45) zu entfernen, sobald eine definierte Anzahl der Datenbankinstanzen (10, 20, 30) die zu bestätigende Transaktion (4) bestätigt hat.

8. Verteiltes Datenbanksystem nach Anspruch 7,
dadurch gekennzeichnet,
dass die definierte Anzahl der Datenbankinstanzen (10, 20, 30) durch die zu bestätigende Transaktion (4) definierbar ist.

9. Verteiltes Datenbanksystem nach einem der Ansprüche 1 bis 8,
dadurch gekennzeichnet,
dass eine Konsensregel einer jeweiligen der Datenbankinstanzen (10, 20, 30) derart eingerichtet ist, dass eine Vergütung für das Bestätigen der zu bestätigenden Transaktion (4) nur eine von der zu bestätigenden Transaktion (4) spezifizierte Anzahl Male vergeben wird.

10. Verteiltes Datenbanksystem nach einem der Ansprüche 1 bis 9,
dadurch gekennzeichnet,
dass das Datenbanksystem (1) dazu eingerichtet ist, die Transaktionsbücher (11, 21, 31) der mehreren Datenbankinstanzen (10, 20, 30) in vordefinierten Zeitabständen miteinander zu synchronisieren.

11. Verteiltes Datenbanksystem nach einem der Ansprüche 1 bis 10,
dadurch gekennzeichnet,
dass die mehreren Datenbankinstanzen (10, 20, 30) dazu eingerichtet sind, beim Bestätigen der zu bestätigenden Transaktion durch eine der mehreren Datenbankinstanzen (10, 20, 30) einen Rückbezug der zu bestätigenden Transaktion (4) datenbankinstanzübergreifend zu verifizieren.

12. Verteiltes Datenbanksystem nach einem der Ansprüche 1 bis 11,
dadurch gekennzeichnet,
dass das Transaktionsbuch (11) einer der mehreren Datenbankinstanzen 10) ein Hauptbuch ist und das Transaktionsbuch (21, 31) einer jeweiligen weiteren der mehreren Datenbankinstanzen (20, 30) ein jeweiliges mit dem Hauptbuch (11) kryptographisch verknüpftes Seitenbuch (21, 31) ist.

13. Verteiltes Datenbanksystem nach Anspruch 12,
dadurch gekennzeichnet,
dass das verteilte Datenbanksystem (1) dazu eingerichtet ist, die zu bestätigende Transaktion (4) entweder in dem Hauptbuch (11) oder in einer Anzahl der Seitenbücher (21, 31) zu bestätigen.

14. Verfahren zum Betreiben eines verteilten Datenbanksystems (1) mit mehreren Datenbankinstanzen

(10, 20),
wobei eine jeweilige der mehreren Datenbankinstanzen (10, 20) durch eine Anzahl Knoteneinrichtungen (12, 13, 14; 22, 23, 24) gebildet ist, die dazu eingerichtet sind, gemeinsam konsensbasiert eines von mehreren Transaktionsbüchern (11, 21) des verteilten Datenbanksystems (1) zu verwalten,
mit den Schritten:

Bestätigen (S1) einer zu bestätigenden Transaktion (4) durch Aufnehmen in das Transaktionsbuch (11, 21) einer oder mehrerer der mehreren Datenbankinstanzen (10, 20),
wobei das verteilte Datenbanksystem (1) entscheidet, durch welche der mehreren Datenbankinstanzen (11, 21) die zu bestätigende Transaktion (4) bestätigt wird.

15. Computerprogrammprodukt, welches auf einer oder mehreren programmgesteuerten Einrichtungen die Durchführung des Verfahrens nach Anspruch 14 veranlasst.

Geänderte Patentansprüche gemäss Regel 137(2) EPÜ.

1. Verteiltes Datenbanksystem (1) mit mehreren Datenbankinstanzen (10, 20),
wobei eine jeweilige der mehreren Datenbankinstanzen (10, 20) durch eine Anzahl Knoteneinrichtungen (12, 13, 14; 22, 23, 24) gebildet ist, die dazu eingerichtet sind, gemeinsam konsensbasiert eines von mehreren Transaktionsbüchern (11, 21) des verteilten Datenbanksystems (1) zu verwalten,
wobei das verteilte Datenbanksystem (1) eingerichtet ist, eine zu bestätigende Transaktion (4) durch Aufnehmen in das Transaktionsbuch (11, 21) einer oder mehrerer der mehreren Datenbankinstanzen (10, 20) zu bestätigen,
dadurch gekennzeichnet, dass
das verteilte Datenbanksystem (1) dazu eingerichtet ist, zu entscheiden, durch welche der mehreren Datenbankinstanzen (10, 20) die zu bestätigende Transaktion (4) bestätigt wird.

2. Verteiltes Datenbanksystem nach Anspruch 1,
dadurch gekennzeichnet,
dass das verteilte Datenbanksystem (1) dazu eingerichtet ist, in Abhängigkeit von einem Zustand der jeweiligen Datenbankinstanz (10, 20) zu entscheiden, durch welche der mehreren Datenbankinstanzen (10, 20) die zu bestätigende Transaktion (4) bestätigt wird.

3. Verteiltes Datenbanksystem nach Anspruch 1 oder 2
dadurch gekennzeichnet,
dass das verteilte Datenbanksystem (1) dazu eingerichtet ist, für jede der Datenbankinstanzen (10, 20) eine Last zu ermitteln und zu entscheiden, dass die zu bestätigende Transaktion (4) durch die eine oder die mehreren der Datenbankinstanzen (10, 20) mit der geringsten Last bestätigt wird.

4. Verteiltes Datenbanksystem nach einem der Ansprüche 1 bis 3,
dadurch gekennzeichnet,
dass das verteilte Datenbanksystem (1) dazu eingerichtet ist, in Abhängigkeit von einem oder mehreren Rückbezügen der zu bestätigenden Transaktion (4) zu entscheiden, durch welche der mehreren Datenbankinstanzen (10, 20, 30) die zu bestätigende Transaktion (4) bestätigt wird.

5. Verteiltes Datenbanksystem nach einem der Ansprüche 1 bis 4,
dadurch gekennzeichnet,
dass das verteilte Datenbanksystem (1) dazu eingerichtet ist, bei der Entscheidung, durch welche der mehreren Datenbankinstanzen die zu bestätigende Transaktion (4) bestätigt wird, eine jeweilige Datenbankinstanz (10, 20, 30) nur zu berücksichtigen, wenn die Datenbankinstanz (10, 20, 30) von der zu bestätigenden Transaktion (4) spezifiziert ist.

6. Verteiltes Datenbanksystem nach einem der Ansprüche 1 bis 5,
dadurch gekennzeichnet,
dass das verteilte Datenbanksystem (1) dazu eingerichtet ist, durch eine automatische dezentrale Konsensbildung zwischen den mehreren Datenbankinstanzen (10, 20, 30) zu entscheiden, durch welche der mehreren Datenbankinstanzen (10, 20, 30) die zu bestätigende Transaktion (4) bestätigt wird.

7. Verteiltes Datenbanksystem nach einem der Ansprüche 1 bis 6,
dadurch gekennzeichnet,
dass das verteilte Datenbanksystem (1) dazu eingerichtet ist, eine Anzahl (40) unbestätigter Transaktionen (41-45) vorzuhalten,
wobei eine jeweilige der mehreren Datenbankinstanzen (10, 20, 30) dazu eingerichtet ist, die zu bestätigende Transaktion unter der vorgehaltenen Anzahl (40) unbestätigter Transaktionen (41-45) auszuwählen und zu bestätigen,
wobei das verteilte Datenbanksystem (1) dazu eingerichtet ist, die zu bestätigende Transaktion (4) aus der vorgehaltenen Anzahl (40) unbestätigter Transaktionen (41-45) zu entfernen, sobald eine definierte Anzahl der Datenbankinstanzen (10, 20, 30) die zu bestätigende Transaktion (4) bestätigt hat.

8. Verteiltes Datenbanksystem nach Anspruch 7,
dadurch gekennzeichnet,
dass die definierte Anzahl der Datenbankinstanzen

(10, 20, 30) durch die zu bestätigende Transaktion (4) definierbar ist.

9. Verteiltes Datenbanksystem nach einem der Ansprüche 1 bis 8,
dadurch gekennzeichnet,
dass eine Konsensregel einer jeweiligen der Datenbankinstanzen (10, 20, 30) derart eingerichtet ist, dass eine Vergütung für das Bestätigen der zu bestätigenden Transaktion (4) nur eine von der zu bestätigenden Transaktion (4) spezifizierte Anzahl Male vergeben wird.

10. Verteiltes Datenbanksystem nach einem der Ansprüche 1 bis 9,
dadurch gekennzeichnet,
dass das Datenbanksystem (1) dazu eingerichtet ist, die Transaktionsbücher (11, 21, 31) der mehreren Datenbankinstanzen (10, 20, 30) in vordefinierten Zeitabständen miteinander zu synchronisieren.

11. Verteiltes Datenbanksystem nach einem der Ansprüche 1 bis 10,
dadurch gekennzeichnet,
dass die mehreren Datenbankinstanzen (10, 20, 30) dazu eingerichtet sind, beim Bestätigen der zu bestätigenden Transaktion durch eine der mehreren Datenbankinstanzen (10, 20, 30) einen Rückbezug der zu bestätigenden Transaktion (4) datenbankinstanzübergreifend zu verifizieren.

12. Verteiltes Datenbanksystem nach einem der Ansprüche 1 bis 11,
dadurch gekennzeichnet,
dass das Transaktionsbuch (11) einer der mehreren Datenbankinstanzen (10) ein Hauptbuch ist und das Transaktionsbuch (21, 31) einer jeweiligen weiteren der mehreren Datenbankinstanzen (20, 30) ein jeweiliges mit dem Hauptbuch (11) kryptographisch verknüpftes Seitenbuch (21, 31) ist.

13. Verteiltes Datenbanksystem nach Anspruch 12,
dadurch gekennzeichnet,
dass das verteilte Datenbanksystem (1) dazu eingerichtet ist, die zu bestätigende Transaktion (4) entweder in dem Hauptbuch (11) oder in einer Anzahl der Seitenbücher (21, 31) zu bestätigen.

14. Verfahren zum Betreiben eines verteilten Datenbanksystems (1) mit mehreren Datenbankinstanzen (10, 20),
wobei eine jeweilige der mehreren Datenbankinstanzen (10, 20) durch eine Anzahl Knoteneinrichtungen (12, 13, 14; 22, 23, 24) gebildet ist, die dazu eingerichtet sind, gemeinsam konsensbasiert eines von mehreren Transaktionsbüchern (11, 21) des verteilten Datenbanksystems (1) zu verwalten,
mit dem Schritt:
Bestätigen (S1) einer zu bestätigenden Transaktion (4) durch Aufnehmen in das Transaktionsbuch (11, 21) einer oder mehrerer der mehreren Datenbankinstanzen (10, 20),
dadurch gekennzeichnet, dass
das verteilte Datenbanksystem (1) entscheidet, durch welche der mehreren Datenbankinstanzen (11, 21) die zu bestätigende Transaktion (4) bestätigt wird.

15. Computerprogrammprodukt, welches auf einer oder mehreren programmgesteuerten Einrichtungen die Durchführung des Verfahrens nach Anspruch 14 veranlasst.

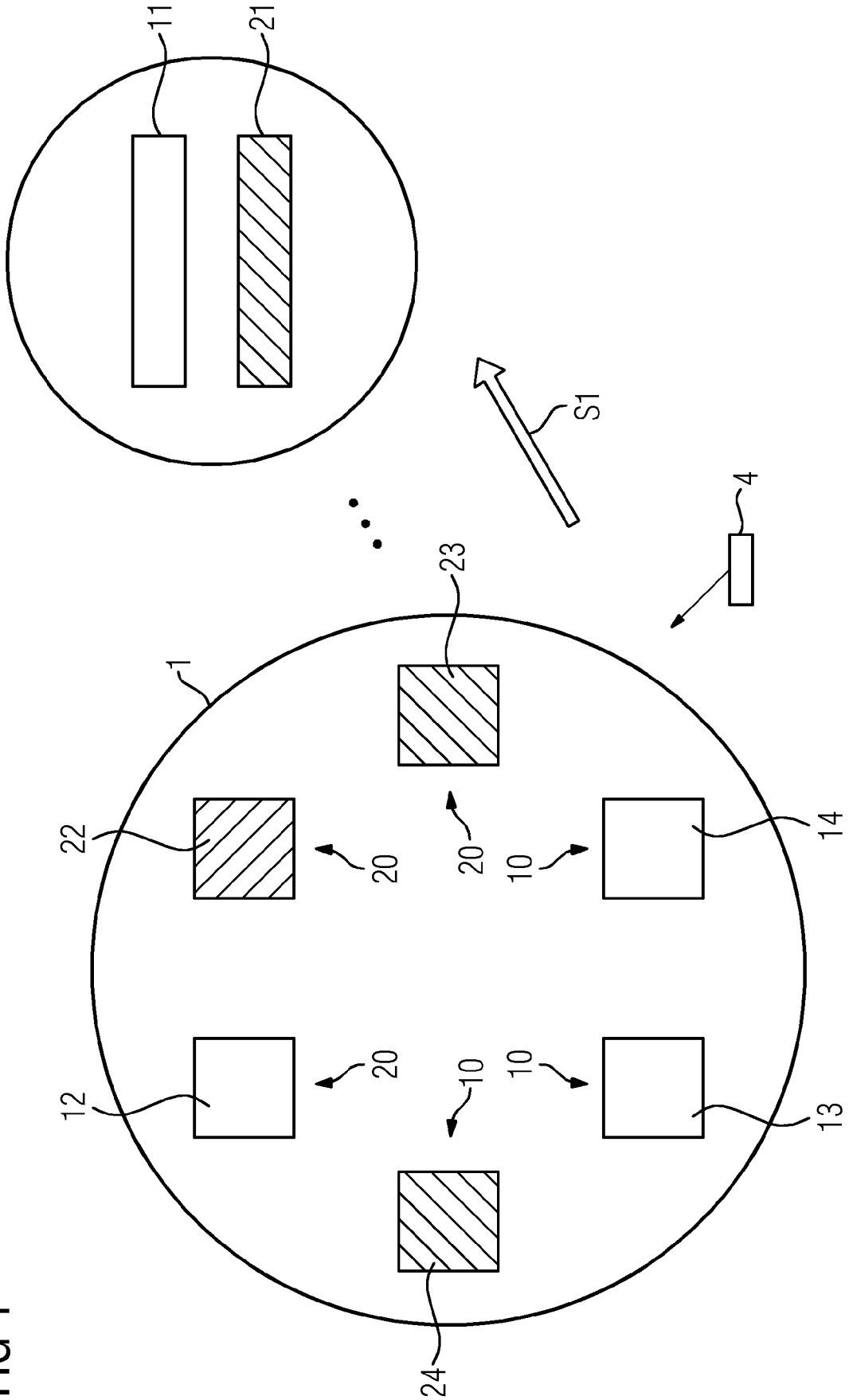


FIG 1

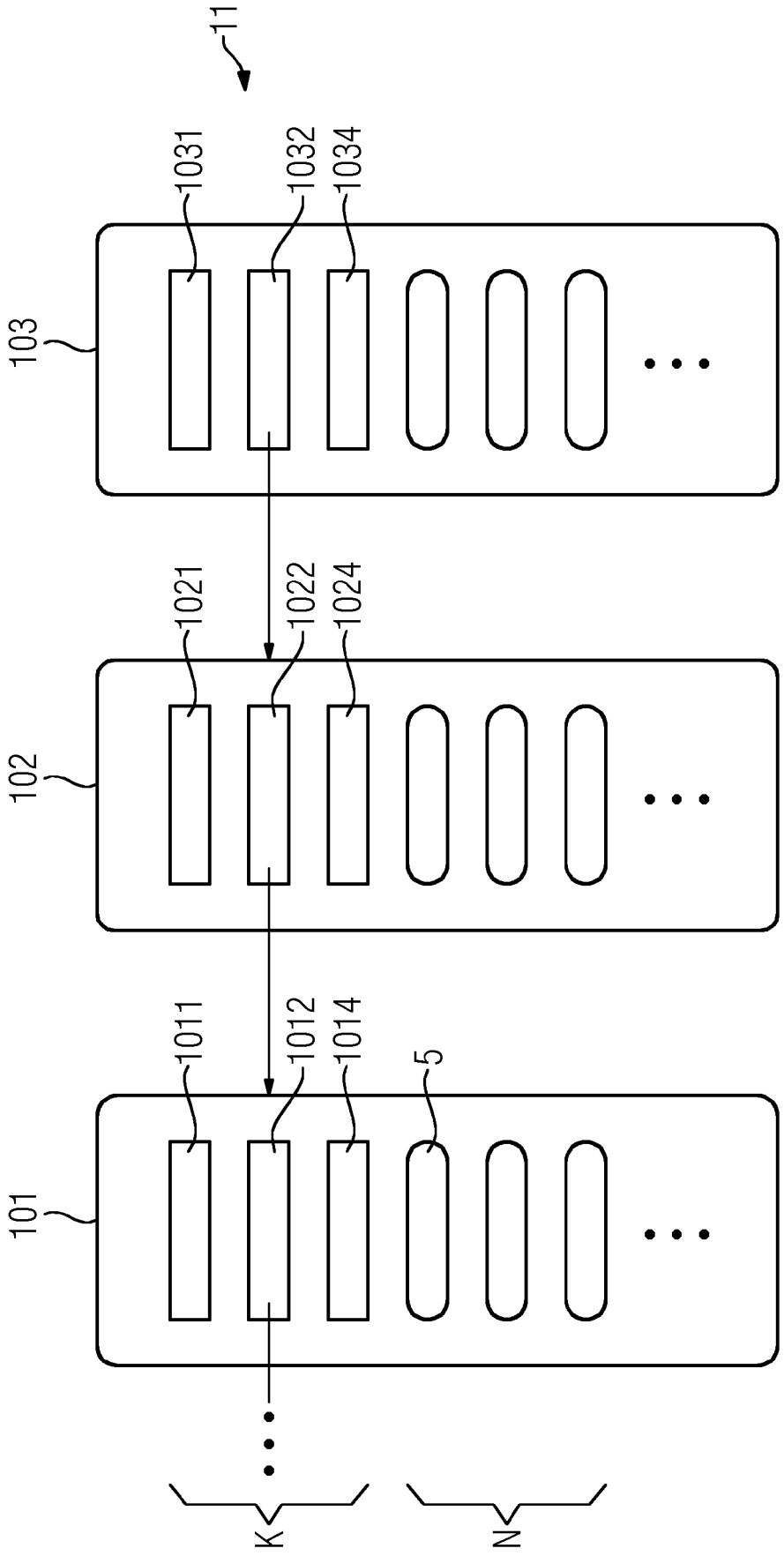
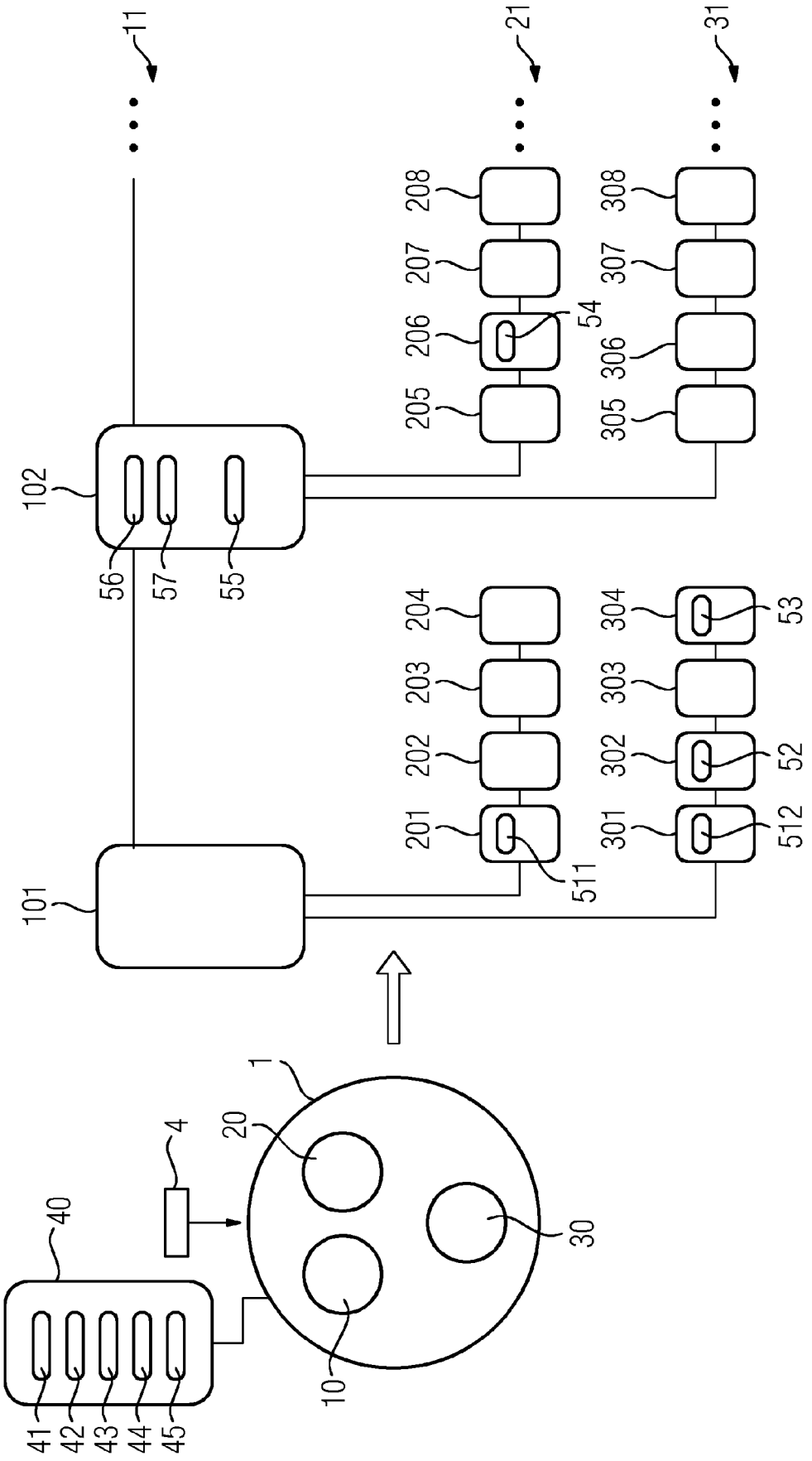

11
1031
1032
1034
103
1021
1022
1024
102
1011
1012
1014
5
101
K
N

FIG 2

FIG 3


11
21
31
102
101
56
57
55
54
511
512
53
52
1
20
4
40
30
10
41
42
43
44
45
201
202
203
204
205
206
207
208
301
302
303
304
305
306
307
308

Europäisches Patentamt
European Patent Office
Office européen des brevets

EUROPÄISCHER RECHERCHENBERICHT

Nummer der Anmeldung
EP 18 19 1983

EINSCHLÄGIGE DOKUMENTE

Kategorie	Kennzeichnung des Dokuments mit Angabe, soweit erforderlich, der maßgeblichen Teile	Betrifft Anspruch	KLASSIFIKATION DER ANMELDUNG (IPC)
X	US 2018/121909 A1 (CHRISTIDIS KONSTANTINOS [US] ET AL) 3. Mai 2018 (2018-05-03) * Absatz [0018] * * Absatz [0021] - Absatz [0023] * * Absatz [0032] * * Abbildungen 3, 4 * -----	1-15	INV. G06Q30/00
X	US 2018/123882 A1 (ANDERSON SHEEHAN [US] ET AL) 3. Mai 2018 (2018-05-03) * Absatz [0024] - Absatz [0026] * * Abbildung 4A * -----	1-15	
X	US 2016/330034 A1 (BACK ADAM [MT] ET AL) 10. November 2016 (2016-11-10) * Absatz [0032] - Absatz [0033] * * Absatz [0079] - Absatz [0084] * -----	1-15	
X	Andreas M. Antonopoulos: "Mastering Bitcoin - Unlocking Digital Cryptocurrencies" In: "Mastering bitcoin : [unlocking digital cryptocurrencies]", 20. Dezember 2014 (2014-12-20), O'Reilly Media, Beijing Cambridge Farnham Köln Sebastopol Tokyo, XP055306939, ISBN: 978-1-4493-7404-4 * Seite 113 * * Seite 123 * * Seite 158 * * Seite 179 - Seite 182 * -----	1-15	RECHERCHIERTE SACHGEBIETE (IPC) G06Q
X	US 9 875 510 B1 (KASPER LANCE [US]) 23. Januar 2018 (2018-01-23) * Spalte 15, Zeile 60 - Spalte 18, Zeile 22 * * Abbildung 2 * -----	1-15	

Der vorliegende Recherchenbericht wurde für alle Patentansprüche erstellt

Recherchenort	Abschlußdatum der Recherche	Prüfer
Den Haag	25. Oktober 2018	Stark, Konrad

KATEGORIE DER GENANNTEN DOKUMENTE

X : von besonderer Bedeutung allein betrachtet
Y : von besonderer Bedeutung in Verbindung mit einer anderen Veröffentlichung derselben Kategorie
A : technologischer Hintergrund
O : nichtschriftliche Offenbarung
P : Zwischenliteratur

T : der Erfindung zugrunde liegende Theorien oder Grundsätze
E : älteres Patentdokument, das jedoch erst am oder nach dem Anmeldedatum veröffentlicht worden ist
D : in der Anmeldung angeführtes Dokument
L : aus anderen Gründen angeführtes Dokument

& : Mitglied der gleichen Patentfamilie, übereinstimmendes Dokument

EPO FORM 1503 03.82 (P04C03)

ANHANG ZUM EUROPÄISCHEN RECHERCHENBERICHT ÜBER DIE EUROPÄISCHE PATENTANMELDUNG NR. EP 18 19 1983

In diesem Anhang sind die Mitglieder der Patentfamilien der im obengenannten europäischen Recherchenbericht angeführten Patentdokumente angegeben.
Die Angaben über die Familienmitglieder entsprechen dem Stand der Datei des Europäischen Patentamts am
Diese Angaben dienen nur zur Unterrichtung und erfolgen ohne Gewähr.

25-10-2018

Im Recherchenbericht angeführtes Patentdokument	Datum der Veröffentlichung	Mitglied(er) der Patentfamilie	Datum der Veröffentlichung
US 2018121909 A1	03-05-2018	KEINE	
US 2018123882 A1	03-05-2018	KEINE	
US 2016330034 A1	10-11-2016	KEINE	
US 9875510 B1	23-01-2018	KEINE	

EPO FORM P0461

Für nähere Einzelheiten zu diesem Anhang : siehe Amtsblatt des Europäischen Patentamts, Nr.12/82

IN DER BESCHREIBUNG AUFGEFÜHRTE DOKUMENTE

Diese Liste der vom Anmelder aufgeführten Dokumente wurde ausschließlich zur Information des Lesers aufgenommen und ist nicht Bestandteil des europäischen Patentdokumentes. Sie wurde mit größter Sorgfalt zusammengestellt; das EPA übernimmt jedoch keinerlei Haftung für etwaige Fehler oder Auslassungen.

In der Beschreibung aufgeführte Nicht-Patentliteratur

- **ANDREAS M. ANTONOPOULOS.** Mastering Bitcoin: Unlocking Digital Cryptocurrencies. *O'Reilly Media,* Dezember 2014 **[0178]**
- **ROGER M. NEEDHAM ; MICHAEL D. SCHROEDER.** Using encryption for authentication in large networks of computers. *ACM: Communications of the ACM,* Dezember 1978, vol. 21 (12 **[0178]**
- **ROSS ANDERSON.** Security Engineering. A Guide to Building Dependable Distributed Systems. Wiley, 2001 **[0178]**
- **HENNING DIEDRICH.** Ethereum: Blockchains, Digital Assets, Smart Contracts, Decentralized Autonomous Organizations. CreateSpace Independent Publishing Platform, 2016 **[0178]**
- The Ethereum Book Project/Mastering Ethereum **[0178]**
- **LEEMON BAIRD.** The Swirlds Hashgraph Consensus Algorithm: Fair, Fast, Byzantine Fault Tolerance. *Swirlds Tech Report SWIRLDS-TR-2016-01,* 31. Mai 2016 **[0178]**
- **LEEMON BAIRD.** *Overview of Swirlds Hashgraph,* 31. Mai 2016 **[0178]**