Schritt 4

Erkennen des Kernproblems

Wie man eine Herde unabhängiger Computer im Zaum hält

Die vorherigen beiden Schritte haben den Zweck der Blockchain im Allgemeinen aufgezeigt und ihre Bedeutung für rein verteilte Peer-to-Peer-Systeme im Speziellen hervorgehoben. Sie haben gelernt, dass der Hauptzweck der Blockchain in dem Erhalten von Integrität in verteilten Systemen besteht. Aber warum stellt es überhaupt ein Problem dar, die Integrität in verteilten Systemen und in rein verteilten Peer-to-Peer-Systemen zu erhalten? Dieser Schritt beantwortet diese Frage, indem er die subtile Beziehung zwischen Vertrauen und Integrität in rein verteilten Peer-to-Peer-Systemen aufdeckt. So vertiefen Sie Ihr Verständnis für die Bedeutung der Integrität und erkennen das Hauptproblem, das die Blockchain lösen soll. Abschließend beschreibt dieser Schritt die Umgebung, in der der größte Nutzen von der Blockchain zu erwarten ist.

Die Metapher

In vielen Sprachen gibt es Redewendungen, die das Bemühen um Organisation in einer chaotischen Gruppe von Individuen beschreiben. Im Deutschen sprechen wir davon, »einen Sack Flöhe zu hüten«. Das veranschaulicht die Probleme, mit denen man es zu tun bekommt, wenn eine Gruppe eigensinniger und kaum fassbarer Tiere keine zentrale Autorität anerkennt oder überhaupt erkennt. Erinnert Sie die Metapher der Individuen, die als Gruppe keine zentrale Autorität anerkennen oder erkennen, an irgendetwas? Tatsächlich beschreibt sie ganz genau die Lage in einem rein verteilten Peer-to-Peer-System, das aus individuellen und unabhängigen Knoten ohne jede Art zentraler Kontrolle oder Koordinierung besteht. Dieser Schritt beschreibt eine der größten Herausforderungen in rein verteilten Peer-to-Peer-Systemen und verrät, was diese mit der Blockchain zu tun hat.

Vertrauen und Integrität in Peer-to-Peer-Systemen

Vertrauen und Integrität sind zwei Seiten derselben Medaille. Im Kontext von Softwaresystemen ist Integrität ein nichtfunktionaler Aspekt eines Systems, das sicher, vollständig, einheitlich, korrekt und mangel- sowie fehlerfrei sein soll. Vertrauen ist die feste Überzeugung von Menschen in die Zuverlässigkeit, Wahrheit oder Fähigkeit einer Person oder Sache, ohne dass es dafür Beweise, Nachweise

oder Untersuchungen gibt. Es wird im Voraus gewährt und wächst oder nimmt ab – abhängig von den Ergebnissen der laufenden Interaktionen.

Bezogen auf Peer-to-Peer-Systeme bedeutet dies, dass Menschen einem System beitreten und dazu beitragen, sofern sie ihm ihr Vertrauen schenken und die Ergebnisse der Interaktionen mit dem System dieses auch fortlaufend bestätigen oder verstärken. Die Integrität des Systems wird benötigt, um die Erwartungen der Anwender zu erfüllen und ihr Vertrauen in das System zu rechtfertigen. Wird es seitens des Systems infolge eines Mangels an Integrität nicht bestärkt, verlassen die Anwender das System, was letztlich zu seinem Ende führen wird. Da Vertrauen so wichtig für die Existenz von Peer-to-Peer-Systemen ist, stellt sich natürlich die überaus wichtige Frage: Wie erreichen und erhalten wir Integrität in einem rein verteilten Peer-to-Peer-System?

In rein verteilten Systemen ist beides von einer Reihe von Faktoren abhängig. Die wichtigsten davon sind:

- Kenntnis von der Anzahl der Knoten oder Peers
- Kenntnis von der Vertrauenswürdigkeit der Peers

Die Chancen, in einem verteilten Peer-to-Peer-System Integrität zu erzielen, sind höher, wenn die Anzahl der Knoten und deren Vertrauenswürdigkeit bekannt sind. Dies lässt sich mit einem privaten Klub vergleichen, der hohen moralischen Werten folgt und strenge Maßstäbe für die Aufnahme neuer Mitglieder anlegt. Im Gegensatz dazu stellt eine unbekannte Anzahl von Knoten mit unbekannter Vertrauenswürdigkeit eine denkbar schlechte Voraussetzung für das Erreichen von Integrität in einem verteilten Peer-to-Peer-System dar. Das ist der Fall, wenn ein rein verteiltes Peer-to-Peer-System im Internet allen offensteht.

Bedrohungen der Integrität in Peer-to-Peer-Systemen

Der Einfachheit halber kann man zwischen zwei großen Bedrohungen der Integrität in Peer-to-Peer-Systemen unterscheiden:

- Technisches Versagen
- Böswillige Peers

Technisches Versagen

Peer-to-Peer-Systeme bestehen aus den Einzelcomputern der Anwender, die über ein Netzwerk kommunizieren. Alle Hardware- und Softwarekomponenten eines Computersystems sowie sämtliche Komponenten des Netzwerks unterliegen dem innewohnenden Risiko eines Ausfalls oder von Fehlern. Daher besteht bei einem verteilten System immer die Gefahr, dass seine Komponenten versagen oder zufällig falsche Ergebnisse liefern.

Böswillige Peers

Böswillige Mitglieder stellen die zweite Bedrohung der Integrität in Peer-to-Peer-Systemen dar. Diese Quelle der Unglaubwürdigkeit ist kein technisches Problem, sondern ein Problem, das durch die Ziele der Individuen verursacht wird, die das System für ihre eigenen Zwecke ausnutzen möchten. Man könnte sagen, dass diese Bedrohung eher mit der Soziologie und Gruppendynamik zu tun hat und weniger mit der Technologie. Unehrliche und böswillige Peers stellen die größte Bedrohung für das Peer-to-Peer-System dar, denn sie greifen das Fundament an, auf dem jedes Peer-to-Peer-System aufgebaut ist: Vertrauen. Sobald die Anwender ihren Peers nicht mehr vertrauen können, wenden sie sich ab und tragen nicht mehr zu den Berechnungsressourcen des Systems bei. Somit nimmt die Anzahl der Mitglieder ab und das Gesamtsystem verliert für die verbleibenden Mitglieder an Attraktivität, was wiederum den Niedergang des Systems beschleunigt, sodass es letztendlich brachliegt.

Das Kernproblem, das die Blockchain lösen soll

Bei besten Bedingungen ist das Erreichen von Integrität und Vertrauen einfach. Die wahre Herausforderung liegt jedoch darin, unter den schlechtesten denkbaren Bedingungen in einem verteilten System Integrität und Vertrauen zu erzielen. Und genau diese Problemstellung soll die Blockchain lösen. Das Kernproblem, das die Blockchain löst, ist das Erreichen und Erhalten von Integrität in einem rein verteilten Peer-to-Peer-System, das aus einer unbekannten Anzahl von Peers mit unbekannter Zuverlässigkeit und Vertrauenswürdigkeit besteht. Dieses Thema ist übrigens nicht neu. Tatsächlich ist es ein allgemein bekanntes und viel diskutiertes Problem in der Informatik. Es wird unter Bezugnahme auf eine militärische Legende als »Problem der byzantinischen Generäle« bezeichnet.[1]

Hinweis

Die durch die Blockchain zu lösende Problemstellung betrifft das Erreichen und Erhalten von Integrität in einem rein verteilten Peer-to-Peer-System, das aus einer unbekannten Anzahl von Peers mit unbekannter Zuverlässigkeit und Vertrauenswürdigkeit besteht.

Ausblick

In diesem Schritt wurde die Wichtigkeit von Integrität und Vertrauen in Peer-to-Peer-Systemen aufgezeigt. Außerdem wurde das Kernproblem, das durch die

1 Lamport, Leslie; Shostak, Robert und Pease, Marshall. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)* 4.3 (1982):382–401.

Blockchain gelöst werden soll, unterstrichen und herausgestellt, welche Bedeutung dem Erreichen von Integrität und Vertrauen in Peer-to-Peer-Systemen zukommt. Allerdings haben wir den Begriff *Blockchain* noch immer nicht definiert. Dies ist Thema des nächsten Schritts.

Zusammenfassung

- Integrität und Vertrauen sind in Peer-to-Peer-Systemen von hoher Wichtigkeit.
- Menschen treten einem Peer-to-Peer-System bei und tragen dazu bei, wenn sie ihm vertrauen und die Ergebnisse der Interaktionen mit dem System dieses Vertrauen fortlaufend bestätigen oder verstärken.
- Sobald Menschen das Vertrauen in ein Peer-to-Peer-System verlieren, verlassen sie es, was letztendlich das Ende für das System bedeutet.
- Die großen Bedrohungen der Integrität in Peer-to-Peer-Systemen sind:
  - Technisches Versagen
  - Böswillige Peers
- Das Erreichen von Integrität in einem Peer-to-Peer-System ist abhängig von:
  - der Kenntnis von der Anzahl der Peers
  - der Kenntnis von der Vertrauenswürdigkeit der Peers
- Das Kernproblem, das die Blockchain löst, ist das Erreichen und Erhalten von Integrität in einem rein verteilten Peer-to-Peer-System, das aus einer unbekannten Anzahl von Peers mit unbekannter Zuverlässigkeit und Vertrauenswürdigkeit besteht.

Schritt 14

Speichern von Transaktionsdaten

Erstellen und Erhalten einer Transaktionsdatenhistorie

Die fünf vorherigen Schritte haben gezeigt, wie Sie Eigentum anhand der gesamten Transaktionsdatenhistorie zurückverfolgen und einzelne Eigentumsübertragungen auf sichere Weise durch das Autorisieren von Transaktionen mithilfe von digitalen Signaturen und das eindeutige Identifizieren von Anwenderkonten beschreiben können. Allerdings wurde bislang noch nicht darauf eingegangen, wie all die Transaktionsdaten, aus denen die Transaktionshistorie besteht, auf sichere Art gespeichert werden. An diesem Punkt kommt die Blockchain-Datenstruktur ins Spiel, deren Aufbau, Beschaffenheit und Funktion Sie auf den folgenden Seiten kennenlernen werden.

Die Metapher

Erinnern Sie sich noch an Ihren letzten Büchereibesuch? Gab es zu diesem Zeitpunkt noch die klassischen Buchkataloge in Karteikartenform in großen Aktenschränken? Büchereikataloge sind Verzeichnisse aller Bücher, die die jeweilige Einrichtung besitzt. In einigen älteren Büchereien gibt es auch heute noch solche Kartenverzeichnisse für das gesamte Inventar. Jede Karte in einem dieser Kataloge steht für ein Buch. Auf der Karte sind die wichtigsten Angaben zu dem betreffenden Buch notiert, zum Beispiel der Name des Autors, der Titel, das Veröffentlichungsdatum und sein Standort in der Bücherei, meist als Stockwerk, Raumnummer, Regalnummer usw. angegeben. Zur Identifizierung der Bücher enthalten die Katalogkarten häufig eindeutige Referenznummern, die auch auf den Buchrücken angegeben sind. Die meisten Büchereien besitzen mehrere Kataloge für unterschiedliche Sortiermerkmale. In einem Verfasserkatalog sind die Karten zum Beispiel alphabetisch nach den Namen der Autoren sortiert, in einem Titelkatalog dagegen alphabetisch nach den Buchtiteln. Man könnte auch einen Sortierkatalog erstellen, in dem die Karten nach dem Datum sortiert sind, zu dem die Bücher in die Bücherei aufgenommen worden sind. Dieser Schritt erläutert, wie die Blockchain Transaktionsdaten auf ähnliche Weise wie in einem Sortierkatalog speichert.

Das Ziel

Das Ziel der Blockchain besteht darin, die gesamte Transaktionsdatenhistorie in sortierter Weise zu speichern.

Die Herausforderung

Die Herausforderung besteht in diesem Fall darin, sämtliche Transaktionsdaten, die jemals angefallen sind, so zu speichern, dass die Ausführungsreihenfolge der Transaktionen erhalten bleibt, während Modifikationen an den Daten schnell und problemlos erkannt werden können. Das schnelle Erkennen von Änderungen ist wichtig, da es die Basis zum Verhindern von Manipulationen und Fälschungen der Transaktionshistorie legt.

Die Idee

Die Idee besteht darin, eine Bibliothek der Transaktionsdaten aufzubauen und einen Sortierkatalog zu führen, der die Reihenfolge ausweist, in der die Transaktionen zur Bibliothek hinzugefügt wurden. Um jegliche Änderungen am Sortierkatalog oder den einzelnen Transaktionsdaten erkennen zu können, müssen die Daten mithilfe von Hashreferenzen veränderungssensitiv gespeichert werden.

Transformieren eines Buchs in eine Blockchain-Datenstruktur

Dieser Abschnitt erklärt, wie ein Buch in eine kleine Bibliothek mit einem Sortierkatalog transformiert werden kann. Wie Sie sehen werden, entspricht dies einer vereinfachten Version der Blockchain-Datenstruktur.

Ausgangspunkt: Ein Buch

Über viele Jahrhunderte wurden Informationen auf unhandlichen Schriftrollen aus Pergament niedergeschrieben und aufbewahrt. Heutzutage stehen uns schriftliche Informationen in Form gebundener Papierstapeln mit nummerierten Seiten zur Verfügung, den Büchern. Bücher sind so allgegenwärtig, dass wir ihre Erfindung als selbstverständlich hinnehmen. Zu ihren wichtigen Merkmalen gehören diese:

- Aufbewahren von Inhalten: Auf den Buchseiten werden Inhalte aufbewahrt.
- Sortierung: Die Sätze auf den Seiten und die Seiten im Buch befinden sich in einer festen Anordnung bzw. Reihenfolge.
- Verbundene Seiten: Die Seiten sind physisch über den Buchrücken und logisch über ihren Inhalt und die Seitenzahlen miteinander verbunden.

Diese Merkmale erlauben es uns, durch das Umlegen der Seiten in einem Buch zu lesen, zurückzublättern oder direkt mithilfe der Seitenzahlen zu einer bestimmten Seite zu springen. Nachfolgend ändern wir nun einige dieser Merkmale und betrachten das Ergebnis.

Schritt 16

Schützen des Datenspeichers

Die Macht der Unveränderlichkeit

Am Ende von Schritt 15 haben wir festgestellt, dass die Blockchain-Datenstruktur Daten veränderungssensitiv speichert. Jede Änderung der in der Blockchain-Datenstruktur gespeicherten Daten wird offensichtlich und muss über einen aufwendigen Prozess in die bestehende Struktur eingepflegt werden. In diesem Schritt wird erläutert, wie dieses Merkmal genutzt werden kann, um die Transaktionsdatenhistorie für die Freigabe und Verteilung in einer nicht vertrauenswürdigen Umgebung vorzubereiten, ohne dass zu befürchten ist, unehrliche Mitglieder eines Peer-to-Peer-Systems könnten deren Inhalt zu ihrem eigenen Vorteil manipulieren.

Die Metapher

Angenommen, ich möchte mich als Mitglied einer ehrwürdigen Adelsfamilie ausgeben. Wie würde ich das tun? Zum Beispiel durch das Fälschen eines Stammbaums. Ich könnte einen adligen Großvater erfinden und in diesem gefälschten Familienstammbaum eine Verbindung zwischen ihm und mir herstellen. Aber überzeugt das andere von meiner (falschen) adligen Herkunft? Dieser Betrug dürfte recht schnell aufgedeckt werden, denn Stammbäume stehen selten für sich allein – sie sind über Verwandtschaftsbeziehungen mit anderen Stammbäumen verbunden und verwoben. Wenn also keiner der Stammbäume der etablierten Adelsfamilien einen Verweis auf oder eine verwandtschaftliche Beziehung mit meinem erfundenen Großvater enthält, ist meine fiktive Familiengeschichte schnell als Betrug entlarvt. Damit meine erfundene Familie anerkannt wird, muss ich auch die Aufzeichnungen einiger der etablierten Adelshäuser fälschen, indem ich Verweise auf meinen ausgedachten Stammbaum in deren Geschichte einfüge. Doch selbst das ist vielleicht noch nicht genug, schließlich leben echte Menschen echte Leben und hinterlassen dabei Spuren in der Weltgeschichte. Mein erfundener Großvater hat allerdings niemals gelebt. Ich muss also einen Lebenslauf für ihn erfinden, damit die Fälschung real erscheint – und zwar für sein gesamtes Leben, von der Kindheit über seine Ausbildung bis zu seinem späteren beruflichen Werdegang. Auch Dokumente, die diesen Lebenslauf belegen, müssen gefälscht werden: Geburtsurkunde, Einschulungsunterlagen, Zeugnisse, Diplome, Ausbil-

dungsbescheinigungen, Meisterbriefe, Mitgliedschaften usw. Schulen, Universitäten und Arbeitgeber haben Unterlagen über die Schüler, Studenten und Mitarbeiter, veröffentlichen Jahrbücher oder Fotografien von gesellschaftlichen Veranstaltungen. Diese Nachweise müsste ich ebenfalls fälschen, damit mein fiktiver Großvater als ehemaliges Mitglied der entsprechenden Einrichtungen durchgeht. Allerdings ist das Manipulieren all dieser Unterlagen nicht nur kompliziert, sondern auch kostspielig, deshalb denke ich, ich würde letztlich doch bei meiner echten, nicht adligen Familiengeschichte bleiben ...

Dieses Gedankenspiel zeigt, dass es zwar möglich ist, die Vergangenheit »zurechtzubiegen«, aber eben auch extrem aufwendig, denn es sind umfassende Änderungen und Fälschungen erforderlich, um die vorzutäuschenden Angaben glaubhaft in die vielen Unterlagen und Querverweise der echten Historie einzubetten. Der hierfür zu betreibende Aufwand ist außerordentlich hoch – im Verhältnis dazu ist es viel einfacher, die Wahrheit zu sagen. In diesem Schritt wird erklärt, inwiefern die Blockchain diese Erkenntnis nutzt, um die Transaktionsdatenhistorie vor Fälschungen zu schützen.

Das Ziel

Es ist wichtig, dass die gesamte Transaktionsdatenhistorie in der Blockchain stets die Wahrheit abbildet, damit sie als vertrauenswürdige Quelle zum Abklären von Eigentumsangelegenheiten dienen kann.

Die Herausforderung

Die Blockchain ist ein für alle zugängliches, rein verteiltes Peer-to-Peer-System. Daher besteht die Gefahr, dass unehrliche Peers die Transaktionsdatenhistorie zu ihrem eigenen Vorteil manipulieren oder fälschen. Die Herausforderung besteht in diesem Fall darin, das System für alle zu öffnen und gleichzeitig die Transaktionsdatenhistorie vor Fälschungen und Manipulationen zu schützen.

Die Idee

In einem offenen verteilten System ist es kaum oder gar nicht möglich, die »ehrlichen Knoten« im Vorfeld von den unehrlichen zu unterscheiden. Um die Transaktionshistorie vor Manipulationen durch »unehrliche Knoten« zu schützen, können wir radikal jede Manipulation der Historie unterbinden, unabhängig davon, wer Änderungen vornehmen will. Wenn niemand die Transaktionsdatenhistorie ändern kann (ob ehrlich oder unehrlich), müssen wir uns um mögliche Manipulationen

keine Gedanken mehr machen. Ist die Transaktionsdatenhistorie also von Anfang an unveränderlich, haben wir das Problem gelöst. Das System kann in der Folge für alle offen bleiben und niemand muss Angst davor haben, dass unehrliche Knoten die Transaktionshistorie manipulieren.

Ein kleiner Abstecher in die Unveränderlichkeit

Unveränderlichkeit bedeutet, dass etwas nicht geändert werden kann. Unveränderliche Daten können nach dem Erzeugen oder Schreiben nicht geändert werden. Daher bezeichnet man diese Daten auch als *Nur-Lese-Daten* oder *schreibgeschützte Daten*. Sie dienen einzig dazu, Informationen ausschließlich zum Lesen oder Darstellen zu liefern – das ist besonders wünschenswert, wenn diese Daten weitergegeben werden und somit keine Kontrolle mehr über ihre weitere Verwendung besteht. Die Weitergabe unveränderlicher Daten ist eine wirkungsvolle Möglichkeit, Änderungen oder Manipulationen an den Dateninhalten zu verhindern. Führerscheine, Ausweispapiere und Zeugnisse sind Beispiele für unveränderliche Dateninhalte im echten Leben. Sie werden von Behörden erstellt, um eine Tatsache zu dokumentieren – und ihr einzig beabsichtigter Zweck besteht darin, vorgezeigt und gelesen zu werden.

Funktionsweise: Das große Ganze

Die Grundidee einer unveränderlichen Transaktionshistorie für die Blockchain besteht darin, Änderungen unverhältnismäßig aufwendig zu machen, sodass jegliche Manipulationsbestrebungen bereits im Keim erstickt werden. Eine unveränderliche Transaktionsdatenhistorie besteht aus drei Komponenten:

1. dem Speichern der Transaktionsdatenhistorie in einer Weise, dass selbst kleinste Manipulationen klar erkennbar und offensichtlich sind,
2. dem Erfordernis, dass das Einbetten einer Manipulation in die Transaktionshistorie ein Neuschreiben großer Teile derselben erfordert,
3. dem Konstruieren einer Historie, in der das Hinzufügen, Schreiben oder Neuschreiben von Daten rechenintensiv ist.

Klar erkennbare Manipulationen

Die Blockchain-Datenstruktur, in der die Daten veränderungssensitiv gespeichert sind, erfüllt die erste Anforderung. Auf diese Weise können Daten, die Teil der Blockchain-Datenstruktur sind, nicht heimlich und in der Hoffnung geändert werden, dass niemand diese Manipulation bemerkt. Jegliche Änderung ist ganz deut-

lich sichtbar, denn sie zerstört die Hashreferenzen, die infolge von Modifikationen in den Verweiszielen ungültig werden.

Erzwungenes Neuschreiben der Historie beim Einbetten von Änderungen

Auch die zweite Anforderung wird von der Blockchain-Datenstruktur erfüllt, denn sie verfolgt bei Änderungen an ihren Daten einen radikalen Alles-oder-nichts-Ansatz: Entweder wird die Datenstruktur ab dem Punkt, der die Änderung verursacht, bis zum Kopf der gesamten Kette geändert, oder man sieht von vornherein besser von Änderungen ab.

Rechenintensives Hinzufügen von Daten

Die dritte Anforderung oder Komponente ist für jene gedacht, die sich nicht scheuen, große Teile der Blockchain-Datenstruktur neu zu schreiben, um eine Manipulation in die Transaktionshistorie einzubetten. Denn sobald das Schreiben oder Neuschreiben der Blockchain-Datenstruktur hohen Rechenaufwand und dadurch hohe (monetäre) Kosten verursacht, überlegen es sich die meisten zweimal, ob das Ändern überhaupt eine gute Idee war.

Das Blockchain-Technologiepaket stellt einen unveränderlichen Inhalt der Blockchain-Datenstruktur sicher, indem das Schreiben, Neu-Schreiben oder Hinzufügen von Blöcken zur Blockchain-Datenstruktur erheblichen Rechenaufwand fordert. Der Rechenaufwand wird durch Hashpuzzles erzeugt, die für jeden Block-Header eindeutig sind und einzeln gelöst werden müssen.[1] Somit kann man entweder den Gesamtaufwand für eine Änderung der Datenstruktur ab dem Punkt, der die Modifikation verursacht hat, bis zum Kopf der Kette (dem Listenkopf) akzeptieren, indem man für jeden beteiligten Block-Header ein Hashpuzzle löst, oder man unterlässt den Manipulationsversuch von vornherein.

Funktionsweise: Die Details

Das Verfahren zum Anfügen eines neuen Blocks an die Blockchain-Datenstruktur selbst (vgl. Schritt 15) ist nicht rechenintensiv, da nur die Hashreferenz, die auf den aktuellen Listenkopf verweist, zum neuen Block-Header hinzugefügt und ein neuer Listenkopf definiert werden muss. Die Herausforderung einer unveränderlichen Blockchain-Datenstruktur besteht darin, das Hinzufügen eines neuen Blocks (künstlich) zu einer rechenintensiven Aufgabe zu machen. Zu diesem Zweck müssen die folgenden Aspekte betrachtet werden:

1 Nakamoto, Satoshi. *Bitcoin: A peer-to-peer electronic cash system*. 2008. `https://bitcoin.org/bitcoin.pdf`.

Schritt 19

Auswählen einer Transaktionshistorie

Wenn Computer mit den Füßen abstimmen

In Schritt 18 wurde gezeigt, wie die Knoten der Blockchain Transaktionsdaten und neue Blöcke verarbeiten. Allerdings kann sich die Transaktionshistorie der einzelnen Knoten des Systems als Folge von Verzögerungen oder Fehlern in der Nachrichtenweitergabe dennoch unterscheiden. In diesem Schritt geht es deshalb darum, wie sich Konflikte zwischen den unterschiedlichen Versionen der Transaktionshistorie einzelner Knoten beheben lassen.

Die Metapher

Wann haben Sie zuletzt einen Spaziergang im Park unternommen? Vielleicht ist Ihnen ja ein Phänomen aufgefallen, das sich in nahezu allen Parks weltweit finden lässt: Es gibt befestigte Wege, die den Plänen und Ideen der Landschaftsgestalter entsprechen, und außerdem von den Parkbesuchern hinterlassene Trampelpfade. Häufig handelt es sich bei diesen Pfaden um die kürzeste Verbindung zwischen zwei Sehenswürdigkeiten, zwei Sitzbänken oder anderen beliebten Zielen. Trampelpfade entstehen dort, wo viele Menschen unabhängig voneinander die befestigten Wege verlassen, weil es ihnen vorteilhaft erscheint, einem anderen als dem vorgesehenen Weg zu folgen. So betrachtet ist ein ausgetretener Pfad im Park eine spezielle Grundform der Demokratie: Es gibt keine offiziellen Umfragen oder Wahlen, die entscheiden, wo ein solcher Pfad geschaffen wird, sondern jeder Besucher entscheidet spontan, wo er entlang läuft und seine Fußstapfen hinterlässt. Seltener genutzte Pfade verschwinden nach und nach, wenn die Natur sie zurückerobert. Andere bleiben bestehen, weil sie von vielen Menschen genutzt werden. In diesem Lernschritt wird ein Aspekt der Blockchain erläutert, der an das Erscheinen und Verschwinden von Trampelpfaden in einem Park erinnert.

Das Ziel

Das Ziel besteht darin, eine eindeutige Transaktionsdatenhistorie unter allen Knoten im Netz beizubehalten, damit beim Abklären des Eigentums ungeachtet des befragten oder kontaktierten Knotens dasselbe Ergebnis erreicht wird.

Die Herausforderung

Der in Schritt 18 beschriebene Blockchain-Algorithmus erlegt allen Knoten im System einen zweistufigen Arbeitstakt auf: Zu jedem beliebigen Zeitpunkt ist jeder Knoten im System entweder mit der Untersuchung eines neuen, von einem Peer erzeugten Blocks beschäftigt oder ringt darum, selbst der nächste Knoten zu sein, der einen neuen Block erzeugt, der wiederum von allen anderen Knoten untersucht wird. Allerdings gibt es keinen übergeordneten Taktgeber, der den Knoten vorschreibt, welche der beiden Arbeiten sie gerade erledigen sollen. Das Eintreffen neuer Blöcke in den Posteingängen der einzelnen Knoten löst den Arbeitstakt des Knotens aus – ebendieses Eintreffen hängt jedoch stark von den Fähigkeiten des Netzwerks zur Nachrichtenverbreitung ab: Nachrichten können verloren gehen, verzögert zugestellt werden oder in zufälliger Reihenfolge eintreffen. Somit liegen zu einem bestimmten Zeitpunkt keine identischen Informationen an allen Knoten vor. Außerdem erfolgt der Wechsel zwischen den beiden Arbeitstakten nicht an allen Knoten gleichzeitig. Stattdessen wechselt jeder Knoten abhängig vom Eintreffen der Nachrichten im Posteingang individuell zwischen den Arbeiten – und das führt zu Überlappungen der Arbeitstakte unter den Knoten. Beide Effekte stellen ein gewaltiges Hindernis für das Führen einer eindeutigen Transaktionsdatenhistorie bei allen Peers im Netz dar. Die Herausforderung besteht also darin, eine Möglichkeit zu finden, trotz aller Widrigkeiten bei der Nachrichtenzustellung und ohne Rückgriff auf eine zentralisierte Lösung eine eindeutige Transaktionsdatenhistorie auszuwählen.

Die Idee

Unser Beispiel mit den Trampelpfaden im Park zeigt, dass eine Gruppe von Menschen allein durch unabhängiges Abstimmen »mit den Füßen« eine Einigung oder einen Konsens für kollektive Entscheidungsprobleme erzielen kann. Diese Art Abstimmungsverhalten wird häufig als *verteilte Konsensentscheidung* bezeichnet, da sie das Ergebnis unabhängig voneinander agierender Individuen ohne zentrale Kontrolle oder Koordinierung ist.

Hinweis

Konsens ist laut Duden die Übereinstimmung der Meinungen mehrerer Parteien. Eine *verteilte Konsensentscheidung* ist die Übereinstimmung der Mitglieder in einem rein verteilten Peer-to-Peer-System.