(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7676425号
(P7676425)

(45)発行日　令和7年5月14日(2025.5.14)

(24)登録日　令和7年5月2日(2025.5.2)

(51)国際特許分類
H04L 9/32 (2006.01)

FI
H04L 9/32 200Z

請求項の数　29　（全72頁）

(21)出願番号　特願2022-549740(P2022-549740)
(86)(22)出願日　令和3年2月19日(2021.2.19)
(65)公表番号　特表2023-513849(P2023-513849A)
(43)公表日　令和5年4月3日(2023.4.3)
(86)国際出願番号　PCT/IB2021/051428
(87)国際公開番号　WO2021/165907
(87)国際公開日　令和3年8月26日(2021.8.26)
審査請求日　令和6年2月16日(2024.2.16)
(31)優先権主張番号　2002285.1
(32)優先日　令和2年2月19日(2020.2.19)
(33)優先権主張国・地域又は機関
英国(GB)
(31)優先権主張番号　2020279.2
(32)優先日　令和2年12月21日(2020.12.21)
最終頁に続く

(73)特許権者　318001991
エヌチェーン ライセンシング アーゲー
スイス・６３００・ツーク・グラーフェナウヴェーク・６
(74)代理人　100108453
弁理士　村山 靖彦
(74)代理人　100110364
弁理士　実広 信哉
(74)代理人　100133400
弁理士　阿部 達彦
(72)発明者　アンドリュー・ジェームズ・ミー
イギリス・カーディフ・ＣＦ１０・２ＨＨ・チャーチル・ウェイ・チャーチル・ハウス・セブンス・フロア・アーカート－ダイクス・アンド・ロード・エルエル
最終頁に続く

(54)【発明の名称】　順序付けられたデータロギングのための方法、およびシステム

(57)【特許請求の範囲】
【請求項１】
ブロックチェーンシステムの中のトランザクションのセットに関連するコンピュータで実施される方法であって、
第２のトランザクションを作成するステップと、
トランザクションの前記セットの中の最新のトランザクションからの出力に関連する少なくとも1つの入力、および前記第2のトランザクションへの第1の参照を備える第1のトランザクションを作成するステップと、
前記第２のトランザクションを前記ブロックチェーンに出すステップと、
前記第１のトランザクションを前記ブロックチェーンに出すステップとを備え、
前記第2のトランザクションがトランザクションの前記セットの中のトランザクションへの第2の参照を備える、コンピュータで実施される方法。
【請求項２】
トランザクションの前記セットのサブセットの中のトランザクションの総数を決定するステップと、
トランザクションの前記サブセットの中のトランザクションの前記総数が閾値以上であるかどうかを決定するステップとをさらに備える、請求項1に記載のコンピュータで実施される方法。
【請求項３】
トランザクションの前記サブセットのメンバーシップが、前記トランザクションが前記

ブロックチェーン上で確認されたかどうか、および、トランザクションの前記セットの中の任意のトランザクションとの消費関係、のうちの少なくとも1つによって定義される、請求項2に記載のコンピュータで実施される方法。

【請求項４】

トランザクションの前記サブセットのメンバーシップが追加で前記閾値によって定義される、請求項3に記載のコンピュータで実施される方法。

【請求項５】

トランザクションの前記サブセットがトランザクションの第1のチェーンを備える、請求項2に記載のコンピュータで実施される方法。

【請求項６】

トランザクションの前記サブセットがトランザクションの前記第1のチェーンである、請求項5に記載のコンピュータで実施される方法。

【請求項７】

トランザクションの前記第1のチェーンが、前記サブセットの中の最初のトランザクションを除く各トランザクションが前記チェーンの中の前のトランザクションへの参照を備えるように構築される、請求項5または6に記載のコンピュータで実施される方法。

【請求項８】

前記前のトランザクションへの前記参照が、前記前のトランザクションからのトランザクション出力に関連する入力である、請求項7に記載のコンピュータで実施される方法。

【請求項９】

トランザクションの前記セットがトランザクションの複数のサブセットを備える、請求項6に記載のコンピュータで実施される方法。

【請求項１０】

トランザクションの前記セットがトランザクションのさらなるチェーンを備える、請求項9に記載のコンピュータで実施される方法。

【請求項１１】

前記閾値が祖先制限に基づく、請求項2に記載のコンピュータで実施される方法。

【請求項１２】

前記閾値が前記祖先制限より1少ない、請求項11に記載のコンピュータで実施される方法。

【請求項１３】

前記第2のトランザクションおよび前記第1のトランザクションを作成して出すステップが、トランザクションの前記サブセットの中のトランザクションの前記総数と前記閾値との比較に基づいて行われる、請求項2に記載のコンピュータで実施される方法。

【請求項１４】

前記第2の前記トランザクションおよび前記第1のトランザクションを作成して出すステップが、トランザクションの前記サブセットの中のトランザクションの前記総数が前記閾値以上であるかどうかに基づいて行われる、請求項13に記載のコンピュータで実施される方法。

【請求項１５】

前記第1の参照が不変の参照である、請求項1に記載のコンピュータで実施される方法。

【請求項１６】

前記第1の参照が前記第2の前記トランザクションの不変の特徴に基づく、請求項15に記載のコンピュータで実施される方法。

【請求項１７】

前記第1の参照が前記第2の前記トランザクションのトランザクションidを備える、請求項1に記載のコンピュータで実施される方法。

【請求項１８】

前記第2の前記トランザクションが、前記第1の前記トランザクションを出す前に前記ブロックチェーン上で確認される、請求項1に記載のコンピュータで実施される方法。

【請求項１９】
前記第2の前記トランザクションが少なくとも1つの入力を備え、第1の参照が、前記第2の前記トランザクションの前記少なくとも1つの入力の少なくとも1つに基づく、請求項1に記載のコンピュータで実施される方法。
【請求項２０】
前記第2の参照がトランザクションの前記セットの中の初期トランザクションへの参照である、請求項1に記載のコンピュータで実施される方法。
【請求項２１】
前記初期トランザクションへの前記参照が前記初期トランザクションのトランザクションidを備える、請求項20に記載のコンピュータで実施される方法。
【請求項２２】
前記第2の参照が不変の参照である、請求項1に記載のコンピュータで実施される方法。
【請求項２３】
前記第2の参照が前記第1のタイプの前記トランザクションの不変の特徴に基づく、請求項22に記載のコンピュータで実施される方法。
【請求項２４】
前記第2の参照が前記第1の前記トランザクションのトランザクションidを備える、請求項1に記載のコンピュータで実施される方法。
【請求項２５】
前記第2の参照が前記第1の前記トランザクションの前記少なくとも1つの入力の少なくとも1つに基づく、請求項22または23に記載のコンピュータで実施される方法。
【請求項２６】
前記少なくとも1つの入力の少なくとも1つに基づく前記参照が、トランザクション出力点の形態をとる、請求項19または25に記載のコンピュータで実施される方法。
【請求項２７】
プロセッサおよびメモリを備えるコンピューティングデバイスであって、前記メモリが、前記プロセッサによる実行の結果として、前記コンピューティングデバイスに請求項1に記載のコンピュータで実施される方法を実行させる実行可能命令を含む、コンピューティングデバイス。
【請求項２８】
請求項27によるデータ書き込みデバイスと、
データを備える要求を前記データ書き込みデバイスに出すように構成されるコンピューティングデバイスとを備える、コンピュータシステム。
【請求項２９】
コンピュータのプロセッサによって実行される結果として、前記コンピュータに請求項1に記載の方法を実行させる実行可能命令が記憶されている、コンピュータ可読記憶媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本開示は、分散型台帳、すなわちブロックチェーンを使用してデータを記憶するための、方法、デバイス、システム、およびデータ構造に関する。より具体的には、本開示は、順序付けられた追加専用のデータ項目のためのストレージを提供する。
【背景技術】
【０００２】
ブロックチェーンとは、ある形式の分散型データ構造を指し、ブロックチェーンの複製は、分散型ピアツーピア(P2P)ネットワーク(以下では「ブロックチェーンネットワーク」と呼ばれる)の中の複数のノードの各々において維持され、広く公開される。ブロックチェーンは、データのブロックのチェーンを備え、各ブロックは、1つまたは複数のトランザクションを備える。いわゆる「コインベーストランザクション」以外の各トランザクションは、1つまたは複数のコインベーストランザクションまでの1つまたは複数のブロックに

またがり得るシーケンスの中の先行するトランザクションを指し示す。コインベーストランザクションは以下で論じられる。ブロックチェーンネットワークに出されるトランザクションは、新しいブロックに含まれる。新しいブロックは「マイニング」と呼ばれることが多い処理により作成され、これは、複数のノードの各々が競争して「プルーフオブワーク」を実行すること、すなわち、ブロックチェーンの新しいブロックに含められることを待機している、順序付けられ妥当性確認された未処理のトランザクションの定められたセットの表現に基づいて、暗号パズルを解くことを伴う。ブロックチェーンはノードにおいて枝刈りされてもよく、ブロックの公開はブロックヘッダだけの公開により達成され得ることに留意されたい。

【０００３】

ブロックチェーンにおけるトランザクションは、デジタル資産(すなわち、ある数のデジタルトークン)を運ぶこと、仮想化された台帳もしくは登録簿の仕訳のセットを順序付けること、タイムスタンプエントリを受信して処理すること、および/またはインデックスポインタを時間的に順序付けることのうちの、1つまたは複数を実行するために使用される。ブロックチェーンは、ブロックチェーンに追加の機能を重ねるためにも利用され得る。ブロックチェーンプロトコルは、トランザクションにおける追加のユーザデータまたはデータに対するインデックスの記憶を可能にし得る。単一のトランザクションに記憶され得る最大のデータ容量にはあらかじめ指定された限界はないので、ますます複雑になるデータを組み込むことができる。たとえば、これは、ブロックチェーンの中の電子文書、またはオーディオデータもしくはビデオデータを記憶するために使用され得る。

【０００４】

ブロックチェーンネットワークのノード(「マイナー」と呼ばれることが多い)は、以下で詳しく説明される、分散型のトランザクションの登録および検証のプロセスを実行する。要約すると、この処理の間に、ノードはトランザクションを妥当性確認し、それらをブロックテンプレートに挿入し、ノードはそのブロックテンプレートについて有効なプルーフオブワークの解を特定することを試みる。有効な解が見つかると、新しいブロックがネットワークの他のノードに広められるので、各ノードがブロックチェーンに新しいブロックを記録することを可能にする。トランザクションがブロックチェーンに記録されるようにするために、ユーザ(たとえば、ブロックチェーンクライアントアプリケーション)は、トランザクションが広められるように、それをネットワークのノードのうちの1つに送信する。トランザクションを受信するノードは競って、妥当性確認されたトランザクションを新しいブロックへ組み込むプルーフオブワークの解を見つけることができる。各ノードは同じノードプロトコルを実施するように構成され、これは、トランザクションが有効になるための1つまたは複数の条件を含む。無効なトランザクションは、広められることも、ブロックに組み込まれることもない。トランザクションが妥当性確認され、それによりブロックチェーン上で受け入れられると仮定すると、トランザクション(あらゆるユーザデータを含む)は、イミュータブルな公開記録としてブロックチェーンネットワークの中のノードの各々において登録されインデクシングされたままになる。

【０００５】

最新のブロックを作成するためにプルーフオブワークパズルを解くことに成功したノードは通常、ある額のデジタル資産、すなわちある数のトークンを分配する「コインベーストランザクション」と呼ばれる新しいトランザクションにより報酬を受ける。無効なトランザクションの検出および拒絶は、ネットワークのエージェントとして活動し不正を報告して阻止する動機のある、競合するノードの活動によって実施される。情報を広く公開することで、ユーザはノードの実績を継続的に監査することが可能になる。ブロックヘッダのみの公開により、参加者はブロックチェーンの完全性が継続中であることを確実にすることが可能になる。

【０００６】

「出力ベース」モデル(UTXOベースのモデルと呼ばれることがある)では、所与のトランザクションのデータ構造は、1つまたは複数の入力および1つまたは複数の出力を備える

。あらゆる消費可能な出力は、トランザクションの先行するシーケンスから導出可能であるデジタル資産の額を指定する要素を備える。消費可能な出力は、UTXO(「未消費トランザクション出力」)と呼ばれることがある。出力はさらに、出力のさらなる引き換えのための条件を指定するロッキングスクリプトを備え得る。ロッキングスクリプトは、デジタルトークンまたは資産を妥当性確認して移すために必要な条件を定義する述部である。トランザクション(コインベーストランザクション以外)の各入力は、先行するトランザクションにおけるそのような出力へのポインタ(すなわち、参照)を備え、指し示された出力のロッキングスクリプトをアンロックするためのアンロッキングスクリプトをさらに備え得る。よって、トランザクションのペアを考え、それらを第1のトランザクションおよび第2のトランザクション(または「標的」トランザクション)と呼ぶ。第1のトランザクションは、デジタル資産の額を指定し、出力をアンロッキンングする1つまたは複数の条件を定義するロッキングスクリプトを備える、少なくとも1つの出力を備える。第2の標的トランザクションは、第1のトランザクションの出力へのポインタと、第1のトランザクションの出力をアンロックするためのアンロッキングスクリプトとを備える、少なくとも1つの入力を備える。

【０００７】

そのようなモデルでは、第2の標的トランザクションが、ブロックチェーンにおいて広められて記録されるようにブロックチェーンネットワークに送信されるとき、各ノードにおいて適用される有効性の基準の1つは、アンロッキングスクリプトが第1のトランザクションのロッキングスクリプトにおいて定義される1つまたは複数の条件のすべてを満たすというものである。別の基準は、第1のトランザクションの出力が別のより前の有効なトランザクションによってまだ引き換えられていないということである。これらの条件のいずれかに従って標的トランザクションが無効であることを見出したいずれのノードも、トランザクションを広めず(場合によっては無効なトランザクションを登録するために有効なトランザクションとして広めない)、またブロックチェーンに記録されるべき新しいブロックにトランザクションを含めない。

【０００８】

代替的なタイプのトランザクションモデルは、アカウントベースモデルである。この場合、各トランザクションは、過去のトランザクションのシーケンスの中の先行するトランザクションのUTXOを参照することによってではなく、絶対的なアカウント残高を参照することによって、移されるべき額を定義する。すべてのアカウントの現在の状態が、ブロックチェーンとは別のノードによって記憶され、定期的に更新される。

【０００９】

ブロックチェーン技術は、暗号通貨の実装という用途において最も広く知られているが、デジタル企業家は、ビットコインが依拠する暗号セキュリティシステムと、新しいシステムを実装するためにブロックチェーンに記憶され得るデータとの両方を使用することを探求している。暗号通貨の世界に限定されない自動化されたタスクおよび処理のためにブロックチェーンを使用できれば、極めて有利であろう。そのようなソリューションは、ブロックチェーンの利点(たとえば、永続的で耐改竄性のあるイベントの記録、分散型処理など)を利用しながら、その用途をより多様にすることが可能である。

【００１０】

現在の研究の一分野は、「スマートコントラクト」の実装のためにブロックチェーンを使用することである。これらは、機械可読の契約または合意の条件の実行を自動化するように設計されるコンピュータプログラムである。自然言語で書かれる従来の契約とは異なり、スマートコントラクトは機械実行可能プログラムであり、これは、結果を生み出すために入力を処理することができるルールを備え、そしてこのルールにより、それらの結果に依存して行為が実行されることが可能になる。ブロックチェーン関連の別の関心分野は、ブロックチェーンを介して現実世界のエンティティを表現して移すために「トークン」(または「カラードコイン」)を使用することである。取り扱いに注意を要する可能性のある、または秘密である可能性のある項目をトークンによって表現することができ、トークン

は認識可能な意味または値を有しない。したがって、トークンは、ブロックチェーンから現実世界の項目が参照されることを可能にする識別子として機能する。

【先行技術文献】

【特許文献】

【００１１】

【文献】英国特許出願第2002285.1号

【文献】英国特許出願第2007597.4号

【文献】英国特許出願第2020279.2号

【発明の概要】

【発明が解決しようとする課題】

【００１２】

上で言及された例またはシナリオは、永続的で耐改竄性のあるイベントの記録を提供するためにブロックチェーンの利点を利用しながら、クライアントが、デジタル資産を管理するための機能である、楕円曲線デジタル署名アルゴリズム(ECDSA)の暗号鍵を実装するためのソフトウェアおよび/もしくはハードウェアを含み、または実装することと、ブロックチェーントランザクションの構築を実施できることと、ブロックチェーンライブラリへのアクセス権を有することとを必要とする。英国特許出願第2002285.1号(2020年2月19日にnChain Holdings Limited名義で出願された)は、ブロックチェーンに関連する1つまたは複数のサービスのためのプラットフォームを説明しており、それによれば、クライアントに関連するデータまたは情報は、そのようなクライアントがブロックチェーンを使用するためのどのような処理または機能も実施する必要なく、それでもブロックチェーンに関連するすべての利点を利用することが可能でありながら、ブロックチェーンに関連する1つまたは複数のサービスのためのアプリケーションプログラミングインターフェース(API)を提供する方法、デバイス、およびシステムによって、簡単に、セキュアに、かつ瞬時にブロックチェーンに書き込まれ、またはブロックチェーンから取得され得る。そのようなプラットフォームのためには、データがプラットフォームを使用してまたはそれを介してブロックチェーンに入力されると、データが改竄または改変されず、独立に監査され得ることを確実にする必要もある。したがって、本開示は、プラットフォームに関連するもしくはそれを介して提供されるデータに基づくイベントのシーケンス、および/または任意のクライアントから提供されるデータ項目の任意のシーケンスが、耐改竄性があり、独立に調査され検証され得ることを確実にするための1つまたは複数の技法を説明する。

【課題を解決するための手段】

【００１３】

第1の態様において、本開示は、順序付けられた追加専用のデータストレージを実装するための方法、デバイス、データ構造、およびシステムを提案する。第1のトランザクションおよび第2のトランザクションを備える、本態様によるデータ構造。第1の出力および第1のデータ項目の表現を備える、第1のトランザクション。第1のデータ項目のさらなる表現、第2のデータ項目の表現、第1の出力に関連する第1の入力、および第2の入力を備える、第2のトランザクション。

【００１４】

要求を受信するステップであって、要求が、データ項目の表現がブロックチェーンに記憶されることの誘因となる、ステップと、トランザクションのセットの中の最新のトランザクションを取得するステップと、最新のトランザクションからの出力に関連する入力、出力、ブロックチェーンに記憶されるべきデータ項目の表現、および最新のトランザクションへの参照を備える、新しいブロックチェーントランザクションを作成するステップと、トランザクションをブロックチェーンに出すステップとを備える、第1の態様によるブロックチェーンシステムにおけるトランザクションのセットに関連する方法。

【００１５】

第2の態様において、本開示は、順序付けられた追加専用のデータストレージを実装するための方法、デバイス、データ構造、およびシステムを提案する。トランザクションの

セットの中の最新のトランザクションからのトランザクション出力に関連する入力を備える第1のタイプのトランザクションを作成するステップと、第2のタイプのトランザクションを作成するステップと、第2のタイプのトランザクションをブロックチェーンに出すステップと、第1のタイプのトランザクションをブロックチェーンに出すステップとを備える、第2の態様による方法。

【００１６】

第2のタイプのトランザクションへの第1の参照を備える出力を備える第1のタイプのトランザクションと、第2のタイプのトランザクションとを備える、第2の態様によるデータ構造。

【００１７】

(a)トランザクションのチェーンにおける現在のトランザクションを取得するステップと、(b)現在のトランザクションが第1のタイプのトランザクションであると決定し、この決定に基づいて、(i)第1のタイプのトランザクションに基づいて第2のタイプのトランザクションへの参照を取得するステップ、(ii)第2のタイプのトランザクションへの参照に基づいて第2のタイプのトランザクションを取得するステップ、および(iii)第2のタイプのトランザクションを現在のトランザクションとしてステップ(c)に続くステップを行うステップと、(c)現在のトランザクション識別子を取得するステップと、(d)現在のトランザクション識別子を参照するさらなるトランザクションを取得するステップと、(e)そのさらなるトランザクションを現在のトランザクションとして開始してステップ(b)、(c)、(d)、および(e)を行い、ループを作成するステップとを備える、トランザクションのセットを前方に調査するための第2の態様による方法。

【００１８】

任意選択で第1の態様の特徴を含む第2の態様。

【００１９】

第3の態様において、本開示は、順序付けられた追加専用のデータストレージを実装するための方法、デバイス、データ構造、およびシステムを提案する。第3の態様による方法は、第2のタイプのトランザクションを作成するステップと、第2のタイプのトランザクションをブロックチェーンに出すステップと、トランザクションのセットの中の最新のトランザクションからのトランザクション出力に関連する入力および第2のタイプのトランザクションへの参照を備える第1のタイプのトランザクションを作成するステップと、第2のタイプのトランザクションがブロックチェーン上で確認された後に第1のタイプのトランザクションをブロックチェーンに出すステップとを備える。

【００２０】

第3の態様は任意選択で、第1の態様および第2の態様の特徴を含む。

【００２１】

第4の態様において、本開示は、順序付けられた追加専用のデータストレージを実装するための方法、デバイス、データ構造、およびシステムを提案する。第1のタイプのトランザクションおよび第2のタイプのトランザクションを備えるデータ構造、第2のタイプのトランザクションは少なくとも1つの入力を備える。第2のタイプのトランザクションへの参照を備える第1のタイプのトランザクション、この参照は少なくとも1つの入力のうちの少なくとも1つに基づく。

【００２２】

第4の態様は任意選択で、第1の態様、第2の態様、および第3の態様の特徴を含む。

【００２３】

第5の態様において、本開示は、順序付けられた追加専用のデータストレージを実装するための方法、デバイス、データ構造、およびシステムを提案する。第2のタイプのトランザクションへの第1の参照を備える第1のタイプのトランザクションと、第1のタイプのトランザクションへの第2の参照と少なくとも1つの入力とを備える第2のタイプのトランザクションとを備える、第5の態様によるデータ構造。好ましくは、第1の参照は、第2のタイプのトランザクションへの少なくとも1つの入力に基づく。好ましくは、第2の参照は

、第1のタイプのトランザクションのトランザクションidである。
【００２４】
(a)トランザクションのチェーンにおいて現在のトランザクションを取得するステップと、(b)現在のトランザクションが第2のタイプのトランザクションであると決定し、この決定に基づいて、(i)第2のタイプのトランザクションに基づいて第1のタイプのトランザクションへの参照を取得するステップ、(ii)第1のタイプのトランザクションへの参照に基づいて第1のタイプのトランザクションを取得するステップ、および(iii)第1のタイプのトランザクションを現在のトランザクションとしてステップ(c)に続くステップを行うステップと、(c)現在のトランザクションから先行するトランザクションのトランザクション識別子を取得するステップと、(d)先行するトランザクションのトランザクション識別子に基づいて先行するトランザクションを取得するステップと、(e)その先行するトランザクションを現在のトランザクションとして開始してステップ(b)、(c)、(d)、および(e)を行い、それによりループを作成するステップとを備える、第5の態様に従ってデータ構造を後方に調査するための第5の態様による方法。
【００２５】
第5の態様は任意選択で、第1の態様、第2の態様、第3の態様、および第4の態様の特徴を含む。
【００２６】
第6の態様において、本開示は、順序付けられた追加専用のデータストレージを実装するための方法、デバイス、データ構造、およびシステムを提案する。チェンジイントランザクションへの第1の参照を備える第1のタイプのトランザクションと、第1のタイプのトランザクションへの第2の参照を備える第2のタイプのトランザクションとを備える、第6の態様によるデータ構造。第1の参照は、第2のタイプのトランザクションの少なくとも1つの入力のうちの少なくとも1つに基づく。第2の参照は、第1のタイプのトランザクションの少なくとも1つの入力のうちの少なくとも1つに基づく。
【００２７】
第6の態様は任意選択で、第1の態様、第2の態様、第3の態様、第4の態様、および/または第5の態様の特徴を含む。
【００２８】
第7の態様において、本開示は、第3のタイプのトランザクションを備える、順序付けられた追加専用のデータストレージを実装するための方法を提案する。任意選択で、ランデブートランザクションである第3のタイプ。
【００２９】
(a)トランザクションのセットにおいて現在のトランザクションを取得するステップと、(b)現在のトランザクションが第3のタイプのトランザクションであると決定し、この決定に基づいて、(i)先行するトランザクションへの参照を備える入力に対するインデックスを取得するステップ、(ii)先行するトランザクションへの参照を備える入力に関連する出力を取得するステップ、(iii)取得された出力に基づいて次のトランザクションを取得するステップ、および(iv)次のトランザクションを現在のトランザクションとしてステップ(c)に続くステップを行うステップと、(c)現在のトランザクション識別子を取得するステップと、(d)現在のトランザクション識別子を参照するさらなるトランザクションを取得するステップと、(e)そのさらなるトランザクションを現在のトランザクションとして開始してステップ(b)、(c)、(d)、および(e)を行い、ループを作成するステップとを備える、ブロックチェーントランザクションのセットを前方に調査する方法。
【００３０】
(a)トランザクションのセットにおける現在のトランザクションを取得するステップと、(b)現在のトランザクションが第3のタイプのトランザクションであると決定し、この決定に基づいて、(i)現在のトランザクションの入力に対するインデックスを取得するステップ、(ii)現在のトランザクションの取得された入力に基づいて先行するトランザクションへの参照を取得するステップ、(iii)参照に基づいて先行するトランザクションを取得するステ

ップ、および(iv)その先行するトランザクションを現在のトランザクションとしてステップ(c)に続くステップを行うステップと、(c)現在のトランザクションから先行するトランザクションのトランザクション識別子を取得するステップと、(d)先行するトランザクションのトランザクション識別子に基づいて先行するトランザクションを取得するステップと、(e)その先行するトランザクションを現在のトランザクションとして開始してステップ(b)、(c)、(d)、および(e)を行い、ループを作成するステップとを備える、ブロックチェーントランザクションのセットを後方に調査するための方法。

【００３１】

第1の態様、第2の態様、第3の態様、第4の態様、第5の態様、および第6の態様の特徴を任意に選択で含む、第7の態様。好ましくは、第7の態様はまた、態様2、3、4、5、および6に関連する前方および後方の調査方法に関する実施形態を備える。

【００３２】

本開示の態様および実施形態が、ここで単なる例として、添付の図面を参照して説明される。

【図面の簡単な説明】

【００３３】

【図１】ブロックチェーンを実装するための例示的なシステムを示す概略図である。

【図２】ブロックシェーンシステムにおいて使用される例示的なトランザクションを示す概略図である。

【図３Ａ】例示的なウォレットシステムおよびユーザインターフェースを示す図である。

【図３Ｂ】例示的なウォレットシステムおよびユーザインターフェースを示す図である。

【図４】ソフトウェアモジュールを備える例示的なブロックチェーンノードを示す概略図である。

【図５】第1の態様による、ログエントリを記憶するトランザクションのチェーンおよび対応するログエントリの概要を示す概略図である。

【図６Ａ】第1の態様によるデータ構造を示す概略図である。

【図６Ｂ】第1の態様によるデータ構造を示す概略図である。

【図６Ｃ】第1の態様によるデータ構造を示す概略図である。

【図６Ｄ】第1の態様による、順序付けられた追加専用のデータストレージシステムを実装するための方法を示す流れ図である。

【図７Ａ】第2の態様によるデータ構造を示す概略図である。

【図７Ｂ】第2の態様によるデータ構造を示す概略図である。

【図８】第2の態様による、順序付けられた追加専用のデータストレージシステムを実装するための方法を示す流れ図である。

【図９】第2の態様による、順序付けられた追加専用のデータストレージ構造を調査するための流れ図方法である。

【図１０】第3の態様による、順序付けられた追加専用のデータストレージのためのデータ構造を示す概略図である。

【図１１】第4の態様による、順序付けられた追加専用のデータストレージのためのデータ構造を示す概略図である。

【図１２】第5の態様による、順序付けられた追加専用のデータストレージのためのデータ構造を示す概略図である。

【図１３】第5の態様による、順序付けられた追加専用のデータストレージ構造を調査するための流れ図方法である。

【図１４】第6の態様による、順序付けられた追加専用のデータストレージのためのデータ構造を示す概略図である。

【図１５】ある態様による、ブロックチェーンに関連する複数のサービスのためのプラットフォームの概要を示す概略図である。

【図１６】ある態様による、ブロックチェーンに関連する複数のサービスのプラットフォームの構成要素を示す概略図である。

【図１７】本開示の様々な態様および実施形態が実装され得るコンピューティング環境を示す概略図である。
【図１８】第8の態様による、順序付けられた追加専用のデータストレージのためのデータ構造を示す概略図である。
【発明を実施するための形態】
【００３４】
ここで、開示される方法のいくつかの特定の構成要素および実施形態が、添付の図面を参照して例示として説明され、同様の参照番号は同様の特徴を指す。
【００３５】
例示的なシステムの概要
図1は、ブロックチェーン150を実装するための例示的なシステム100を示す。システム100は、通常はインターネットなどのワイドエリアインターネットワークである、パケット交換ネットワーク101からなり得る。パケット交換ネットワーク101は、パケット交換ネットワーク101内でピアツーピア(P2P)ネットワーク106を形成するように並べられ得る複数のブロックチェーンノード104を備える。示されていないが、ブロックチェーンノード104は準完全グラフとして並べられ得る。したがって、各ブロックチェーンノード104は、他のブロックチェーンノード104に高度に接続される。
【００３６】
各ブロックチェーンノード104は、ピアのコンピュータ機器を備え、異なるノード104は異なるピアに属する。各ブロックチェーンノード104は、1つまたは複数のプロセッサ、たとえば1つまたは複数の中央処理装置(CPU)、アクセラレータプロセッサ、特定用途向けプロセッサおよび/またはフィールドプログラマブルゲートアレイ(FPGA)、ならびに特定用途向け集積回路(ASIC)などの他の機器を備える、処理装置を備える。各ノードはまた、メモリ、すなわち、非一時的コンピュータ可読媒体の形態のコンピュータ可読ストレージを備える。メモリは、1つまたは複数のメモリ媒体、たとえば、ハードディスクなどの磁気媒体、ソリッドステートドライブ(SSD)、フラッシュメモリ、もしくはEEPROMなどの電子媒体、および/または高額ディスクドライブなどの光学媒体を利用する、1つまたは複数のメモリユニットを備え得る。
【００３７】
ブロックチェーン150はデータのブロック151のチェーンを備え、ブロックチェーン150のそれぞれのコピーが、分散ネットワークまたはブロックチェーンネットワーク101の中の複数のブロックチェーンノード104の各々において維持される。上で言及されたように、ブロックチェーン150のコピーを維持することは、ブロックチェーン150を完全に記憶することを必ずしも意味しない。代わりに、ブロックチェーン150は、各ブロックチェーンノード150が各ブロック151のブロックヘッダ(以下で論じられる)を記憶する限り、データを枝刈りされ得る。チェーンの中の各ブロック151は1つまたは複数のトランザクション152を備え、この文脈においてトランザクションはある種のデータ構造を指す。データ構造の性質は、トランザクションモデルまたはスキームの一部として使用されるトランザクションプロトコルのタイプに依存する。所与のブロックチェーンは、1つの特定のトランザクションプロトコルを全体で使用する。ある一般的なタイプのトランザクションプロトコルにおいて、各トランザクション152のデータ構造は、少なくとも1つの入力および少なくとも1つの出力を備える。各出力は、ある数量のデジタル資産を表す額を財産として指定し、その例は、出力が暗号によりにロックされる対象であるユーザ103である(アンロック、および引き換えまたは消費のために、そのユーザの署名または他のソリューションを必要とする)。各入力は、先行するトランザクション152の出力を指し示し、それによりそれらのトランザクションをつなぐ。
【００３８】
各ブロック151はまた、ブロック151に対する逐次的な順序を定義するために、チェーンの中の以前に作成されたブロック151を指し示すブロックポインタ155を備える。各トランザクション152(コインベーストランザクション以外)は、トランザクションのシーケ

ンスに対する順序を定義するために、以前のトランザクションへのポインタを備える(トランザクション152のシーケンスは分岐することが許容されることに留意されたい)。ブロック151のチェーンは、チェーンにおいて最初のブロックであったジェネシスブロック(Gb)153まで戻る。チェーン150の初期の1つまたは複数の元のトランザクション152は、先行するトランザクションではなくジェネシスブロック153を指し示していた。

【００３９】

ブロックチェーンノード104の各々は、トランザクション152を他のブロックチェーンノード104に転送し、それにより、トランザクション152がネットワーク106全体に広められるようにするように構成される。各ブロックチェーンノード104は、ブロック151を作成し、同じブロックチェーン150のそれぞれのコピーをそれぞれのメモリに記憶するように構成される。各ブロックチェーンノード104はまた、ブロック151へと組み込まれるのを待機しているトランザクション152の順序付けられたセット154を維持する。順序付けられたセット154は、「メモリプール」と呼ばれることが多い。本明細書におけるこの用語は、任意の特定のブロックチェーン、プロトコル、またはモデルに限定することを意図しない。それは、ノード104が有効であるものとして受け入れた、かつ同じ出力を消費することを試みる他のトランザクションをノード104が受け入れることが義務付けられない、トランザクションの順序付けられたセットを指す。

【００４０】

所与の現在のトランザクション152jにおいて、入力(または各入力)は、トランザクションのシーケンスの中の先行するトランザクション152iの出力を参照するポインタを備え、これは、この出力が現在のトランザクション152jにおいて引き換えられる、または「消費される」ことになることを指定する。一般に、先行するトランザクションは、順序付けられたセット154または任意のブロック151における任意のトランザクションであり得る。先行するトランザクション152iは、現在のトランザクション152iが作成される時点で、またはネットワーク106に送信される時点ですら、必ずしも存在する必要はないが、現在のトランザクションが有効になるためには、先行するトランザクション152iが存在して妥当性確認される必要がある。したがって、本明細書における「先行する」は、ポインタにより連結される論理シーケンスにおいて先行するものを指し、時間的な順序における作成または送信の時間を必ずしも指さず、したがって、トランザクション152i、152jが順不同で作成または送信されることを必ずしも排除しない(オーファントランザクションについての以下の議論を参照)。先行するトランザクション152iは同様に、祖先トランザクションまたは先行者トランザクションと呼ばれ得る。

【００４１】

現在のトランザクション152jの入力はまた、入力承認、たとえば、先行するトランザクション152iの出力がロックされる対象であるユーザ103aの署名を備える。そして、現在のトランザクション152jの出力は、新しいユーザまたはエンティティ103bに暗号によりロックされ得る。したがって、現在のトランザクション152jは、現在のトランザクション152jの出力において定義されるような新しいユーザまたはエンティティ103bに、先行するトランザクション152iの入力において定義される額を移すことができる。いくつかの場合、トランザクション152は、複数のユーザまたはエンティティ(そのうちの1つは、残金を与えるために元のユーザまたはエンティティ103aであり得る)の間で入力の額を分割するために、複数の出力を有し得る。いくつかの場合、トランザクションはまた、1つまたは複数の先行するトランザクションの複数の出力からの額を一緒に集めて、現在のトランザクションの1つまたは複数の出力を再分配するために、複数の入力を有し得る。

【００４２】

ビットコインなどの出力ベースのトランザクションプロトコルによれば、ユーザまたは機械などのエンティティ103が新しいトランザクション152jを実施することを望むとき、エンティティはそのコンピュータ端末102から受信者に新しいトランザクションを送信する。エンティティまたは受信者は最終的に、このトランザクションをネットワーク106のブロックチェーンノード104(これは今日では通常はサーバまたはデータセンターであるが

、原理的には他のユーザ端末であってもよい)のうちの1つまたは複数に送信する。新しいトランザクション152jを実施するエンティティ103がトランザクションをブロックチェーンノード104のうちの1つまたは複数に送信でき、いくつかの例では受信者に送信できないことも、排除されない。トランザクションを受信するブロックチェーンノード104は、ブロックチェーンノード104の各々において適用されるブロックチェーンノードプロトコルに従って、トランザクションが有効であるかどうかを確認する。ブロックチェーンノードプロトコルは通常、新しいトランザクション152jの中の暗号署名が予想される署名と一致することをブロックチェーンノード104が確かめることを必要とし、予想される署名は、トランザクション152の順序付けられたシーケンスの中の以前のトランザクション152iに依存する。そのような出力ベースのトランザクションプロトコルでは、これは、新しいトランザクション152jの入力に含まれるエンティティ103の暗号署名または他の承認が、新しいトランザクションが割り当てる先行するトランザクション152iの出力において定義される条件と一致することを確かめることを備えることがあり、この条件は通常、新しいトランザクション152jの入力の中の暗号署名または他の承認が、新しいトランザクションの入力がつなげられる以前のトランザクション152iの出力をアンロックすることを、少なくとも確かめることを備える。この条件は、先行するトランザクション152iの出力に含まれるスクリプトによって少なくとも部分的に定義され得る。

【００４３】

代替として、それは単純にブロックチェーンノードプロトコルだけによって固定されてもよく、または、それはこれらの組合せによるものであってもよい。いずれにしても、新しいトランザクション152jが有効である場合、ブロックチェーンノード104は、それをブロックチェーンネットワーク106の中の1つまたは複数の他のブロックチェーンノード104に転送する。これらの他のブロックチェーンノード104は、同じブロックチェーンノードプロトコルに従って同じ試験を適用し、新しいトランザクション152jを1つまたは複数のさらなるノード104に転送するなどする。このようにして、新しいトランザクションが、ブロックチェーンノード104のネットワーク全体に広められる。

【００４４】

出力ベースのモデルにおいて、所与の出力(たとえば、UXTO)が割り当てられるかどうかの定義は、それがブロックチェーンノードプロトコルに従って別の前方のトランザクション152jの入力によりすでに有効に引き換えられているかどうかである。トランザクションが有効になるための別の条件は、そのトランザクションが割り当てることまたは引き換えることを試みる先行するトランザクション152iの出力が、別のトランザクションによってまだ割り当てられていない/引き換えられていないことである。やはり、有効ではない場合、トランザクション152jは、ブロックチェーン150において広められず(無効であるものとしてフラグを立てられて警告のために広められない限り)、または記録されない。これは、取引者が同じトランザクションの出力を一度より多く割り当てることを試みるような、二重消費から守る。一方、アカウントベースのモデルは、アカウント残高を維持することによって二重消費から守る。やはり、トランザクションの定められた順序があるので、アカウント残高は任意のある時間において単一の定められた状態を有する。

【００４５】

トランザクションを検証することに加えて、ブロックチェーンノード104はまた、マイニングと一般に呼ばれるプロセスにおいて、トランザクションのブロックを最初に作成するのを競い、これは「プルーフオブワーク」により支援される。ブロックチェーンノード104において、新しいトランザクションは、ブロックチェーン150に記録されているブロック151にまだ表れていない有効なトランザクションの順序付けられたセット154に追加される。そして、ブロックチェーンノードは、暗号パズルを解こうとすることによって、トランザクションの順序付けられたセット154からトランザクション152の新しい有効なブロック151を競って組み立てる。通常、これは、「ノンス」がトランザクションの順序付けられたセット154の表現と連結されてハッシュされると、ハッシュの出力が所定の条件を満たすような、ノンス値を探すことを備える。たとえば、所定の条件は、ハッシュの

出力がある定められた数の先頭の0を有するということであり得る。これは、プルーフオブワークパズルの1つの具体的なタイプにすぎず、他のタイプが排除されないことに留意されたい。ハッシュ関数の性質は、それがその入力に関して予測不可能な出力を有するというものである。したがって、この探索は、ブルートフォースによってのみ実行することができるので、パズルを解こうとしている各ブロックチェーンノード104において大量の処理リソースを消費する。

【００４６】

パズルを解こうとする第1のブロックチェーンノード104は、これをネットワーク106に告知し、ネットワークの中の他のブロックチェーンノード104によって容易に確かめられ得る証明として解を提供する(ハッシュへの解が与えられると、それによりハッシュの出力が条件を満たすようになることを確かめるのは単純である)。第1のブロックチェーンノード104は、ブロックを受け入れしたがってプロトコルルールを実施する、他のノードの閾値コンセンサスにブロックを広める。トランザクションの順序付けられたセット154は次いで、ブロックチェーンノード104の各々によってブロックチェーン150の中の新しいブロック151として記録されるようになる。ブロックポインタ155はまた、チェーンの中の以前に作成されたブロック151n-1を指し示す新しいブロック151nに割り当てられる。プルーフオブワークの解を作成するために必要とされる、たとえばハッシュの形式の大量の労力は、ブロックチェーンプロトコルのルールに従うという第1のノードの104の意図を示すものである。そのようなルールは、以前に妥当性確認されたトランザクションと同じ出力を割り当てる場合(これは別様に二重消費として知られている)、有効であるものとしてトランザクションを受け入れないことを含む。作成されると、ブロック151を改変することはできず、それは、ブロック151が、ブロックチェーンネットワーク106の中のブロックチェーンノード104の各々において認識され維持されるからである。ブロックポインタ155はまた、逐次的な順序をブロック151に課す。トランザクション152は、ネットワーク106の中の各ブロックチェーンノード104において順序付けられるブロックに記録されるので、これはトランザクションのイミュータブルな公開台帳を提供する。

【００４７】

任意の所与の時間において競ってパズルを解く異なるブロックチェーンノード104は、それらのブロックチェーンノードがいつ解の探索を始めたか、またはトランザクションが受信された順序に応じて、任意の所与の時間におけるまだ公開されていないトランザクション154の順序付けられたセットの異なるスナップショットに基づいて、競ってパズルを解いていることがあることに留意されたい。それぞれのパズルを最初に解いた者が、どのトランザクション152が次の新しいブロック151nに含まれるか、およびどの順序で含まれるかを定義し、公開されていないトランザクションの現在のセット154は更新される。ブロックチェーンノード104は次いで、公開されていないトランザクション154の新しく定義された傑出した順序付けられたセットからブロックを競って作成し続け、以下同様である。生じ得るあらゆる「フォーク」を解決するためのプロトコルも存在し、これは、2つのブロックチェーンノード104が互いに非常に短い時間内にパズルを解き、その結果、ブロックチェーンの矛盾する景色がノード104間で広められるようになる状況である。つまり、フォークの先端がより長く成長した方が、最終的なブロックチェーン150になる。同じトランザクションが両方のフォークに現れるので、これはネットワークのユーザまたはエージェントに影響しないはずであることに留意されたい。

【００４８】

ビットコインブロックチェーン(および大半の他のブロックチェーン)によれば、新しいブロック104を構築することに成功するノードは、(あるエージェントまたはユーザから別のエージェントまたはユーザにある額のデジタル資産を移す、エージェント間またはユーザ間のトランザクションとは対照的に)定められた数量のデジタル資産を分配する新しい特別な種類のトランザクションにおいて、許容される額のデジタル資産を割り当てる能力を与えられる。この特別なタイプのトランザクションは普通、「コインベーストランザクション」と呼ばれるが、「開始トランザクション」とも呼ばれ得る。それは通常、新しいブ

ロック151nの最初のトランザクションを形成する。プルーフオブワークは、この特別なトランザクションが後で引き換えられることを可能にするプロトコルルールに従うという、新しいブロックを構築するノードの意図を示すものである。ブロックチェーンプロトコルルールは、この特別なトランザクションを引き換えられるようになるまで、成熟期間、たとえば100ブロックを必要とし得る。しばしば、通常の(非生成)トランザクション152はまた、そのトランザクションが公開されたブロック151nを作成したブロックチェーンノード104にさらに報酬を与えるために、その出力の1つにおいて追加のトランザクションフィーを指定する。この料金は普通は「トランザクションフィー」と呼ばれ、以下で論じられる。

【００４９】

トランザクションの妥当性確認および公開に関与するリソースにより、典型的にはブロックチェーンノード104の少なくとも各々が、1つまたは複数の物理サーバユニットを備えるサーバという形態をとり、またはデータセンター全体という形態すらとる。しかしながら、原理的には、あらゆる所与のブロックチェーンノード104は、一緒にネットワーク接続されたユーザ端末またはユーザ端末のグループという形態をとり得る。

【００５０】

各ブロックチェーンノード104のメモリは、それぞれの役割を実行し、ブロックチェーンノードプロトコルに従ってトランザクション152を扱うように、ブロックチェーンノード104の処理装置上で実行するように構成される、ソフトウェアを記憶する。ブロックチェーンノード104に対する本明細書に起因するあらゆる活動が、それぞれのコンピュータ機器の処理装置で実行されるソフトウェアによって実施され得ることが理解されるだろう。ノードソフトウェアは、アプリケーション層における1つまたは複数のアプリケーションで、またはオペレーティングシステム層もしくはプロトコル層などのより低次の層で、またはこれらの任意の組合せで実装され得る。

【００５１】

消費するユーザの役割を果たす複数の関係者103の各々のコンピュータ機器102も、ネットワーク101に接続される。これらのユーザは、ブロックチェーンネットワークと対話し得るが、トランザクションおよびブロックの検証、構築、または伝播には参加しない。これらのユーザまたはエージェント103の一部は、トランザクションにおいて送信者または受信者として活動し得る。他のユーザは、必ずしも送信者または受信者として活動することなく、ブロックチェーン150と対話し得る。たとえば、一部の関係者は、ブロックチェーン150のコピーを記憶する(たとえば、ブロックチェーンノード104からブロックチェーンのコピーを取得した)ストレージエンティティとして活動し得る。

【００５２】

関係者103の一部またはすべてが、異なるネットワーク、たとえばブロックチェーンネットワーク106に重畳されるネットワークの一部として接続され得る。ブロックチェーンネットワークのユーザ(「クライアント」と呼ばれることが多い)は、ブロックチェーンネットワークを含むシステムの一部であると言われることがある。しかしながら、これらのユーザはブロックチェーンノード104ではなく、それは、ブロックチェーンノードに必要とされる役割を実行しないからである。代わりに、各関係者103は、ブロックチェーンネットワーク106と対話し、それにより、ブロックチェーンノード106に接続する(すなわち、それと通信する)ことによって、ブロックチェーン150を利用し得る。第1の関係者103aおよびそのそれぞれのコンピュータ機器102a、ならびに第2の関係者103bおよびそのそれぞれのコンピュータ機器102bという、2名の関係者103および彼らのそれぞれの機器102が例示を目的に示されている。より多くのそのような関係者103およびそれぞれのコンピュータ機器102が、システム100において存在して参加していてもよいが、便宜的にそれらは示されていないことが理解されるだろう。各関係者103は、個人または組織であり得る。純粋に例示として、第1の関係者103aはAliceと本明細書では呼ばれ、第2の関係者103bはBobと呼ばれるが、これは限定するものではなく、本明細書でのAliceまたはBobへのあらゆる言及は、それぞれ「第1の関係者」および「第2の関係者」で置き換えられ得

ることが理解されるだろう。

【００５３】

各関係者103のコンピュータ機器102は、1つまたは複数のプロセッサ、たとえば1つまたは複数のCPU、GPU、他のアクセラレータプロセッサ、特定用途向けプロセッサ、および/またはFPGAを備える、それぞれの処理装置を備える。各関係者103のコンピュータ機器102はさらに、非一時的コンピュータ可読媒体の形式のメモリ、すなわちコンピュータ可読ストレージを備える。このメモリは、1つまたは複数のメモリ媒体、たとえばハードディスクなどの磁気媒体、SSD、フラッシュメモリ、もしくはEEPROMなどの電子媒体、および/または光学ディスクドライブなどの光学媒体を利用する、1つまたは複数のメモリユニットを備え得る。各関係者103のコンピュータ機器102のメモリは、処理装置上で実行するようになされる少なくとも1つのクライアントアプリケーション105のそれぞれのインスタンスを備えるソフトウェアを記憶する。所与の関係者103に対する本明細書に起因するあらゆる活動は、それぞれのコンピュータ機器102の処理装置上で実行されるソフトウェアを使用して実行され得ることが理解されるだろう。各関係者103のコンピュータ機器102は、少なくとも1つのユーザ端末、たとえばデスクトップもしくはラップトップコンピュータ、タブレット、スマートフォン、またはスマートウォッチなどのウェアラブルデバイスを備える。所与の関係者103のコンピュータ機器102はまた、ユーザ端末を介してアクセスされるクラウドコンピューティングリソースなどの、1つまたは複数の他のネットワーク接続されたリソースを備え得る。

【００５４】

クライアントアプリケーション105は最初に、たとえばサーバからダウンロードされる、あるいは、リムーバブルSSD、フラッシュメモリキー、リムーバブルEEPROM、リムーバブル磁気ディスクドライブ、磁気フロッピーディスクもしくはテープ、CDもしくはDVD ROMなどの光学ディスク、またはリムーバブル光学ドライブなどの、リムーバブルストレージデバイス上で提供される、適切なコンピュータ可読記憶媒体上の任意の所与の関係者103のコンピュータ機器102に提供され得る。

【００５５】

クライアントアプリケーション105は、少なくとも「ウォレット」機能を備える。これには2つの主要な機能がある。これらのうちの1つは、それぞれの関係者103がトランザクション152を作成し、承認(たとえば署名)し、1つまたは複数のビットコインノード104に送信して、トランザクション152がブロックチェーンノード104のネットワーク全体に広められてブロックチェーン150に含まれるようにすることを可能にすることである。もう1つは、それぞれの関係者が現在所有するデジタル資産の額をそれぞれの関係者に報告することである。出力ベースのシステムでは、この第2の機能は、対象の関係者に属するブロックチェーン150全体に散在する様々な152トランザクションの出力において定義される額を照合することを備える。

【００５６】

注意:様々なクライアント機能は所与のクライアントアプリケーション105へと統合されるものとして説明されることがあるが、これは必ずしも限定するものではなく、代わりに、本明細書において説明されるあらゆるクライアント機能は、一連の2つ以上の別個の適用例、たとえばAPIを介してインターフェースすること、または一方が他方へのプラグインであることにおいて実装され得る。より一般的には、クライアント機能は、アプリケーション層、またはオペレーティングシステムなどのより低次の層、またはこれらの任意の組合せにおいて実装され得る。以下は、クライアントアプリケーション105に関して説明されるが、それは限定するものではないことが理解されるだろう。

【００５７】

各コンピュータ機器102上のクライアントアプリケーションまたはソフトウェア105のインスタンスは、ネットワーク106のブロックチェーンノード104のうちの少なくとも1つに動作可能に結合される。これは、クライアント105のウォレット機能がトランザクション152をネットワーク106に送信することを可能にする。クライアント105はまた、そ

れぞれの関係者103が受信者であるあらゆるトランザクションについてブロックチェーン150にクエリするために、ブロックチェーンノード104に連絡することも可能である(または、実施形態では、ブロックチェーン150が、公的な存在であることにより一部トランザクションに信用をもたらす公的機関であるので、実際にブロックチェーン150における他の関係者のトランザクションを調査する)。各コンピュータ機器102のウォレット機能は、トランザクションプロトコルに従ってトランザクション152を編成して送信するように構成される。上で述べられたように、各ブロックチェーンノード104は、ブロックチェーンノードプロトコルに従ってトランザクション152を妥当性確認し、ブロックチェーンネットワーク106全体にトランザクション152を広めるためにそれらを転送するように構成される、ソフトウェアを実行する。トランザクションプロトコルおよびノードプロトコルは互いに対応し、所与のトランザクションプロトコルは所与のノードプロトコルを伴い、一緒に所与のトランザクションモデルを実装する。ブロックチェーン150の中のすべてのトランザクション152に対して、同じトランザクションプロトコルが使用される。同じノードプロトコルが、ネットワーク106の中のすべてのノード104によって使用される。

【００５８】

所与の関係者103、たとえばAliceが、新しいトランザクション152jをブロックチェーン150に含まれるように送信することを望むとき、彼女は関連するトランザクションプロトコルに従って(彼女のクライアントアプリケーション105のウォレット機能を使用して)新しいトランザクションを編成する。彼女は次いで、クライアントアプリケーション105から、彼女が接続されている1つまたは複数のブロックチェーンノード104に、トランザクション152を送信する。たとえば、これは、Aliceのコンピュータ102に最善に接続されるブロックチェーンノード104であり得る。任意の所与のブロックチェーンノード104が新しいトランザクション152jを受信するとき、ブロックチェーンノード104は、ブロックチェーンノードプロトコルおよびそのそれぞれの役割に従って、新しいトランザクション152jを扱う。これは、新しく受信されたトランザクション152jが「有効」であるための何らかの条件を満たすかどうかをまず確かめることを備え、その例がまもなくより詳しく論じられる。一部のトランザクションプロトコルでは、妥当性確認のための条件は、トランザクション152に含まれるスクリプトによってトランザクションごとに構成可能であり得る。代替として、この条件は単に、ノードプロトコルの内蔵機能であってもよく、またはスクリプトとノードプロトコルの組合せによって定義されてもよい。

【００５９】

新しく受信されるトランザクション152jが有効であるものとして見なされるように試験に合格する条件(すなわち、それが「妥当性確認される」条件)のもとで、トランザクション152jを受信する任意のブロックチェーンノード104が、新しい妥当性確認されたトランザクション152をそのブロックチェーンノード104に維持されているトランザクションの順序付けられたセット154に追加する。さらに、トランザクション152jを受信するあらゆるブロックチェーンノード104は、妥当性確認されたトランザクション152以降をネットワーク106の中の1つまたは複数の他のブロックチェーンノード104に広める。各ブロックチェーンノード104は同じプロトコルを適用するので、トランザクション152jが有効であると仮定すると、これは、それがまもなくネットワーク106全体に広められることを意味する。

【００６０】

所与のブロックチェーンノード104において維持されるトランザクションの順序付けられたセット154の利用を認められると、そのブロックチェーンノード104は、新しいトランザクション152を含むトランザクションのそれぞれの順序付けられたセット154の最新のバージョンについてのプルーフオブワークパズルを競って解き始める(他のブロックチェーンノード104が、トランザクションの異なる順序付けられたセット154に基づいてパズルを解こうとしていることがあるが、最初にたどり着いた者が最新のブロック1511に含まれるトランザクションの順序付けられたセットを定義することを思い出されたい。最終的に、ブロックチェーンノード104は、Aliceのトランザクション152jを含む順序付けら

れたセット154の一部のためのパズルを解く)。プルーフオブワークが、新しいトランザクション152jを含む順序付けられたセット154に対して行われると、それはイミュータブルに、ブロックチェーン150の中のブロック151のうちの1つの一部になる。各トランザクション152は、より前のトランザクションへのポインタを備えるので、トランザクションの順序もイミュータブルに記録される。

【００６１】

異なるブロックチェーンノード104は、所与のトランザクションの異なるインスタンスをまず受信するので、あるインスタンスが新しいブロック151において公開される前は、どのインスタンスが「有効」であるかについて矛盾した見方を有することがあり、それが公開される時点では、公開されるインスタンスが唯一の有効なインスタンスであることにすべてのブロックチェーンノード104が合意している。ブロックチェーンノード104があるインスタンスを有効であるものとして受け入れ、第2のインスタンスがブロックチェーン150に記録されていることを発見する場合、そのブロックチェーンノード104は、これを受け入れ、最初に受け入れたインスタンス(すなわち、ブロック151において公開されていないインスタンス)を廃棄する(すなわち、無効であるものとして扱う)。

【００６２】

一部のブロックチェーンネットワークによって運用される代替のタイプのトランザクションプロトコルは、アカウントベースのトランザクションモデルの一部として、「アカウントベース」プロトコルと呼ばれることがある。アカウントベースの場合、各トランザクションは、過去のトランザクションのシーケンスの中の先行するトランザクションのUTXOを参照することによってではなく、絶対的なアカウント残高を参照することによって、移されるべき額を定義する。すべてのアカウントの現在の状態が、ブロックチェーンとは別に、そのネットワークのノードによって記憶され、定期的に更新される。そのようなシステムでは、トランザクションは、アカウントの実行中のトランザクションタリー(「ポジション」とも呼ばれる)を使用して順序付けられる。この値は、暗号署名の一部として送信者により署名され、トランザクション参照計算の一部としてハッシュされる。加えて、任意選択のデータフィールドはまた、署名されたトランザクションであってもよい。このデータフィールドは、たとえば以前のトランザクションIDがデータフィールドに含まれる場合、以前のトランザクションを指し示し得る。

【００６３】

UTXOベースのモデル

図2は、例示的なトランザクションプロトコルを示す。これは、UTXOベースのプロトコルの例である。トランザクション152(「Tx」と省略される)は、ブロックチェーン150の基本データ構造である(各ブロック151は1つまたは複数のトランザクション152を備える)。以下は、出力ベースまたは「UTXO」ベースのプロトコルに言及して説明される。しかしながら、これはすべての可能な実施形態への限定ではない。例示的なUTXOベースのプロトコルはビットコインに言及して説明されるが、それは他の例示的なブロックチェーンネットワーク上で等しく実装され得ることに留意されたい。

【００６４】

UTXOベースのモデルでは、各トランザクション(「Tx」)152は、1つまたは複数の入力202および1つまたは複数の出力203を備えるデータ構造を備える。各出力203は、未消費のトランザクション出力(UTXO)を備えてもよく、これは、別の新しいトランザクションの入力202のソースとして使用され得る(UTXOがまだ引き換えられていない場合)。UTXOは、デジタル資産の額を指定する値を含む。これは、分散型台帳上のある設定された数のトークンを表す。UTXOはまた、情報の中でもとりわけ、UTXOの由来であるトランザクションのトランザクションIDを含み得る。トランザクションデータ構造はヘッダ201も備えることがあり、これは入力フィールド202および出力フィールド203のサイズのインジケータを備えることがある。ヘッダ201はまた、トランザクションのIDを含むことがある。実施形態では、トランザクションIDは、トランザクションデータ(トランザクションID自体を除く)のハッシュであり、ノード104に出される生のトランザクション152のヘッ

ダ201に記憶される。

【００６５】

Alice 103aが、対象のある額のデジタル資産をBob 103bに移すトランザクション152jを作成することを望んでいるとする。図2において、Aliceの新しいトランザクション152jは「Tx1」とラベリングされる。Tx1は、シーケンスの中の先行するトランザクション152iの出力203においてAliceにロックされるデジタル資産の額をとり、その少なくとも一部をBobに移す。先行するトランザクション152iは、図2では「Tx0」とラベリングされる。Tx0およびTx1は任意のラベルにすぎない。それらは、Tx0がブロックチェーン151の最初のトランザクションであることを必ずしも意味せず、Tx1がプール154の中のすぐ次のトランザクションであることも意味しない。Tx1は、Aliceにロックされている未消費の出力203をまだ有するあらゆる先行する(すなわち、祖先)トランザクションを指し示し得る。

【００６６】

先行するトランザクションTx0は、Aliceが新しいトランザクションTx1を作成するとき、または少なくとも彼女がそれをネットワーク106に送信するときにはすでに、ブロックチェーン150のブロック151において妥当性確認されそれに含まれていることがある。それは、その時点ですでにブロック151のうちの1つに含まれていることがあり、または、順序付けられたセット154においてまだ待機していることがあり、その場合、それは新しいブロック151にまもなく含められる。代替として、Tx0およびTx1は、一緒に作成されてネットワーク106に送信されてもよく、または、ノードプロトコルが「オーファン」トランザクションのバッファリングを許容する場合、Tx0がTx1の後に送信されることすらあってもよい。トランザクションのシーケンスの文脈で本明細書において使用される「先行する」および「後続の」という用語は、トランザクションにおいて指定されるトランザクションポインタによって定義されるようなシーケンスにおけるトランザクションの順序を指す(どのトランザクションがどの他のトランザクションを指し示すか、など)。それらは、「先行者」および「後継者」、または「祖先」および「子孫」、「親」および「子」などにより等しく置き換えられ得る。これは、それらが作成される順序、ネットワーク106に送信される順序、または任意の所与のブロックチェーンノード104に到達する順序を必ずしも示唆しない。それでも、先行するトランザクション(祖先トランザクションまたは「親」)を指し示す後続のトランザクション(子孫トランザクションまたは「子」)は、親トランザクションが妥当性確認されるまでは、かつ妥当性確認されない限り、妥当性確認されない。親より前にブロックチェーンノード104に到達する子は、オーファンであると見なされる。それは、ノードプロトコルおよび/またはノード挙動に応じて、廃棄され、または親を待機するためにある時間の間バッファリングされ得る。

【００６７】

先行するトランザクションTx0の1つまたは複数の出力203のうちの1つは、ここでUTXO0とラベリングされる特定のUTXOを備える。各UTXOは、UTXOによって表されるデジタル資産の額を指定する値と、後続のトランザクションが妥当性確認されるようにするために、したがってUTXOの引き換えが成功するために、後続のトランザクションの入力202におけるアンロッキングスクリプトによって満たされなければならない条件を定義するロッキンスクリプトとを備える。通常、ロッキングスクリプトは、額を特定の関係者(ロッキングスクリプトが含まれるトランザクションの受益者)にロックする。すなわち、ロッキングスクリプトはアンロッキング条件を定義し、その条件は通常、後続のトランザクションの入力におけるアンロッキングスクリプトが、先行するトランザクションがロックされる対象である関係者の暗号署名を備えるという条件を備える。

【００６８】

ロッキングスクリプト(scriptPubKeyとしても知られている)は、ノードプロトコルによって認識される分野特有の言語で書かれるコードである。そのような言語の具体的な例は、ブロックチェーンネットワークによって使用される「Script」(大文字のS)と呼ばれる。ロッキングスクリプトは、トランザクション出力203を消費するためにどの情報が必要

とされるか、たとえば、Aliceの署名の要件を指定する。アンロッキングスクリプトは、トランザクションの出力に現れる。アンロッキングスクリプト(scriptSigとしても知られている)は、ロッキングスクリプト基準を満たすために必要とされる情報を提供する分野特有の言語で書かれるコードである。たとえば、それはBobの署名を含み得る。アンロッキングスクリプトはトランザクションの入力202に現れる。

【００６９】

よって、示される例では、Tx0の出力203におけるUTXO0は、UXTO0が引き換えられるようにするために(厳密には、UTXO0を引き換えようとする後続のトランザクションが有効になるために)Aliceの署名SIG PAを必要とするロッキングスクリプト[Checksig PA]を備える。[Checksig PA]は、Aliceの公開-秘密鍵のペアからの公開鍵PAの表現(すなわち、ハッシュ)を含む。Tx1の入力202は、Tx1を指し示す(たとえば、そのトランザクションIDであるTxID0によって指し示す、TxID0は実施形態ではトランザクション全体Tx0のハッシュである)ポインタを備える。Tx1の入力202は、Tx0のあらゆる他のあり得る出力の中からUTXO0を特定するために、Tx0内でUTXO0を特定するインデックスを備える。Tx1の入力202はさらに、Aliceが鍵のペアからの自身の秘密鍵をデータのあらかじめ定められた部分(暗号学では「メッセージ」と呼ばれることがある)に適用することによって作成される、Aliceの暗号署名を備えるアンロッキングスクリプト〈Sig PA〉を備える。Aliceにより有効な署名を提供するために署名される必要のあるデータ(または「メッセージ」)は、ロッキングスクリプトによって、またはノードプロトコルによって、またはこれらの組合せによって定義され得る。

【００７０】

新しいトランザクションTx1がブロックチェーンノード104に到達すると、ノードはノードプロトコルを適用する。これは、アンロッキングスクリプトがロッキングスクリプトにおいて定義される条件(この条件は1つまたは複数の基準を備え得る)を満たすかどうかを確かめるために、ロッキングスクリプトおよびアンロッキングスクリプトを一緒に実行することを備える。実施形態では、これは2つのスクリプトを連結することを伴う。

〈Sig PA〉〈PA〉┆┆[Checksig PA]

ここで、「┆┆」は連結を表し、「〈...〉」はスタックにデータを置くことを意味し、「[...]」はロッキングスクリプト(この例では、スタックベース言語)に含まれる関数である。等価的に、スクリプトを連結するのではなく、スクリプトは共通のスタックを用いて次々に実行されてもよい。いずれにしても、一緒に実行されると、スクリプトは、Tx0の出力の中のロッキングスクリプトに含まれるような、Aliceの公開鍵PAを使用して、Tx1の入力の中のアンロッキングスクリプトがデータの予想される部分に署名するAliceの署名を含むことを認証する。データ自体(「メッセージ」)の予想される部分も、この認証を実行するために含まれる必要がある。実施形態では、署名されたデータはTx1の全体を備える(よって、平文でデータの署名された部分を指定する別個の要素が含まれる必要がなく、それは、もともと存在していたからである)。

【００７１】

公開-秘密暗号による認証の詳細は、当業者には馴染みがある。基本的に、Aliceが自身の秘密鍵を使用してメッセージに署名した場合、平文のAliceの公開鍵およびメッセージを与えられると、ノード104などの別のエンティティは、メッセージがAliceによって署名されたに違いないことを認証することが可能である。署名することは通常、メッセージをハッシュし、ハッシュに署名し、これを署名としてメッセージへとタグ付けすることで、公開鍵のあらゆる保有者が署名を認証することを可能にすることを備える。したがって、本明細書における、特定のデータまたはトランザクションの一部に署名することなどへのあらゆる言及は、実施形態では、そのデータまたはトランザクション一部のハッシュに署名することを意味することに留意されたい。

【００７２】

Tx1におけるアンロッキングスクリプトがTx0のロッキングスクリプトにおいて指定される1つまたは複数の条件を満たす場合(よって示される例では、Aliceの署名がTx1におい

て提供されて認証される場合)、ブロックチェーンノード104はTx1を有効であると見なす。これは、ブロックチェーンノード104がTx1をトランザクションの順序付けられたセット154に追加することを意味する。ブロックチェーンノード104はまた、ネットワーク106の中の1つまたは複数の他のブロックチェーンノード104にトランザクションTx1を転送するので、それは、ネットワーク106全体に広められる。Tx1がブロックチェーン150において妥当性確認され含められると、これは消費されるものとしてTx0からのUTXO0を定義する。Tx1は、未消費のトランザクション出力203を消費する場合にのみ、有効であり得ることに留意されたい。別のトランザクション152によってすでに消費されている出力を消費しようとする場合、Tx1は、すべての他の条件が満たされている場合でも無効になる。したがって、ブロックチェーンノード104は、先行するトランザクションTx0の中の参照されるUTXOがすでに消費されているかどうか(すなわち、すでに有効な入力を別の有効なトランザクションへと形成したかどうか)を確かめる必要もある。これは、トランザクション152に定められた順序を課すことがブロックチェーン150にとって重要である1つの理由である。実際には、所与のブロックチェーンノード104は、トランザクション152がその中で消費されたどのUTXO203をマークする別個のデータベースを維持してもよいが、究極的には、UTXOが消費されたかどうかを定義するものは、UTXOが有効な入力をブロックチェーン150の中の別の有効なトランザクションへとすでに形成したかどうかである。

【００７３】

所与のトランザクション152のすべての出力203において指定される総額が、すべてのその入力202によって指し示される総額より大きい場合、これもまた、大半のトランザクションモデルにおいて、無効であることの根拠になる。したがって、そのようなトランザクションは、広められず、ブロック151にも含められない。

【００７４】

UTXOベースのトランザクションモデルにおいて、所与のUTXOは全体として消費される必要があることに留意されたい。それは、消費されるものとしてUTXOにおいて定義される額の一部を、別の一部が消費されながら「置き去りにする」ことができない。しかしながら、UTXOからの額は、次のトランザクションの複数の出力の間で分割され得る。たとえば、Tx0の中のUTXO0において定義される額は、Tx1の中の複数のUTXO間で分割され得る。したがって、AliceがUTXO0において定義される額のすべてをBobに与えることを望まない場合、彼女はリマインダーを使用してTx1の第2の出力の残金を自分に与え、または別の関係者に支払うことができる。

【００７５】

実際には、Aliceは普通は、自分のトランザクション104を公開するビットコインノードに対する料金を含める必要がある。Aliceがそのような料金を含めない場合、Tx0はブロックチェーンノード104によって拒絶されてもよく、したがって、技術的には有効であっても、広められず、ブロックチェーン150に含められなくてもよい(ノードプロトコルは、ブロックチェーンノード104がトランザクション152を受け入れることを望まない場合、それを強いることはない)。一部のプロトコルでは、トランザクションフィーは、固有の別々の出力203を必要としない(すなわち、別個のUTXOを必要としない)。代わりに、入力202によって指し示される総額と所与のトランザクション152の出力203において指定される総額とのあらゆる差が、トランザクションを公開するブロックチェーンノード104に自動的に与えられる。たとえば、UTXO0へのポインタがTx1への唯一の入力であり、Tx1が唯一の出力UTXO1を有するとする。UTXO0において指定されるデジタル資産の額がUTXO1において指定される額より大きい場合、その差は、UTXO1を含むブロックを公開するノード104によって割り当てられ得る。しかしながら、代替または追加として、トランザクションフィーが、トランザクション152のUTXO203のうちの自身固有のUTXOにおいて明示的に指定され得ることは、必ずしも排除されない。

【００７６】

AliceおよびBobのデジタル資産は、ブロックチェーン150のどこかにある任意のトラン

ザクション152において彼らにロックされるUTXOからなる。したがって、通常は、所与の関係者103の資産は、ブロックチェーン150全体の、様々なトランザクション152のUTXO全体に分散している。所与の関係者103の総残高を定義する1つの数字が、ブロックチェーン150のどこかに保管されているということはない。それぞれの関係者にロックされており、別のその先のトランザクションにおいてまだ消費されていないすべての様々なUTXOの値を一緒に照合することが、クライアントアプリケーション150のウォレット機能の役割である。そのウォレット機能は、ビットコインノード104のいずれかに記憶されているようなブロックチェーン150のコピーをクエリすることによって、これを行うことができる。

【００７７】

スクリプトコードはしばしば、概略的(すなわち、厳密な言語を使用せずに)に表現されることに留意されたい。たとえば、特定の関数を表すためにオペレーションコード(オペコード)を使用することがある。「OP_...」は、Script言語の特定のオペコードを指す。例として、OP_RETURNは、ロッキングスクリプトの最初おいてOP_FALSEが前にあるとトランザクション内のデータを記憶できるトランザクションの消費不可能な出力を生み出し、それによりブロックチェーン150にデータをイミュータブルに記録するような、Script言語のオペコードである。たとえば、データは、ブロックチェーンに記憶することが望まれる文書を備え得る。

【００７８】

このようにOP_RETURNを使用することは、ビットコインベースのブロックチェーンシステム上で使用するために、証明可能で消費不可能なスクリプトを使用する具体的な例である。異なるブロックチェーンシステムは異なる機構を有し、スクリプトが消費不可能であることを確実にするための、および/またはトランザクションにデータを記憶するためのデータフォーマットを有することを、当業者は理解するだろう。

【００７９】

通常、トランザクションの入力は、公開鍵PAに対応するデジタル署名を含む。実施形態では、これは、楕円曲線secp256k1を使用するECDSAに基づく。デジタル署名は特定のデータに署名する。いくつかの実施形態では、所与のトランザクションに対して、署名はトランザクション入力の一部、およびトランザクション出力の一部またはすべてに署名する。署名する出力の具体的な部分は、SIGHASHフラグに依存する。SIGHASHフラグは普通は、どの出力が署名されるかを選択するために署名の最後に含まれる(したがって署名の時点で固定される)4バイトのコードである。

【００８０】

ロッキングスクリプトは時々「scriptPubKey」と呼ばれ、それぞれのトランザクションがロックされる対象である関係者の公開鍵をロッキングスクリプトが通常は備えるという事実を指している。アンロッキングスクリプトは時々「scriptSig」と呼ばれ、アンロッキングスクリプトが対応する署名を通常は供給するという事実を指している。しかしながら、より一般的には、UTXOが引き換えられるようにするための条件が署名を認証することを備えることは、ブロックチェーン150のすべての適用例において必須ではない。より一般的には、スクリプト言語は、任意の1つまたは複数の条件を定義するために使用され得る。したがって、より一般的な用語「ロッキングスクリプト」および「アンロッキングスクリプト」が好まれることがある。

【００８１】

図1に示されるように、AliceおよびBobのコンピュータ機器102a、120bの各々のクライアントアプリケーションは、それぞれ、追加の通信機能を備え得る。この追加の機能は、Alice 103aがBob 103bとの別個のサイドチャネル160を確立する(いずれかの関係者または第三者の教唆により)ことを可能にする。サイドチャネル160は、ブロックチェーンネットワークとは別にデータの交換を可能にする。そのような通信は、「オフチェーン」通信と呼ばれることがある。たとえば、これは、AliceおよびBobの一方がトランザクション152をネットワーク106にブロードキャストすることを選ぶまで、トランザクション15

2がブロックチェーンネットワーク106に(まだ)登録されることなく、またはチェーン150に向かって進むことなく、AliceとBobとの間でトランザクション152を交換するために使用され得る。このようにトランザクションを共有することは、「トランザクションテンプレート」の共有と呼ばれることがある。トランザクションテンプレートは、完全なトランザクションを形成するために必要とされる1つまたは複数の入力および/または出力を欠いていることがある。代替または追加として、サイドチャネル160は、鍵、交渉される額または条項、データコンテンツなどの、任意の他のトランザクション関連データを交換するために使用され得る。

【００８２】

サイドチャネル160は、ブロックチェーンネットワーク106と同じパケット交換ネットワーク101を介して確立され得る。代替または追加として、サイドチャネル160は、モバイルセルラーネットワークなどの異なるネットワーク、またはローカルワイヤレスネットワークなどのローカルエリアネットワーク、または、Aliceのデバイス102aとBobのデバイス102bとの間の直接の有線もしくはワイヤレスリンクすらも介して確立され得る。一般に、本明細書の他の箇所において言及されるサイドチャネル160は、「オフチェーン」で、すなわちブロックチェーンネットワーク106とは別にデータを交換するための、1つまたは複数のネットワーキング技術または通信媒体を介した、任意の1つまたは複数のリンクを備え得る。1つより多くのリンクが使用される場合、オフチェーンリンクの束または集合体は全体として、サイドチャネル160と呼ばれ得る。したがって、AliceおよびBobがいくつかの情報またはデータなどを、サイドチャネル160を介して交換すると言われる場合、これは必ずしも、すべてのこれらのデータが厳密に同じリンクで送信されなければならないこと、または同じタイプのネットワークで送信されなければならないことすらも示唆しない。

【００８３】

クライアントソフトウェア

図3Aは、ここで開示される方式の実施形態を実装するためのクライアントアプリケーション105の例示的な実装形態を示す。クライアントアプリケーション105は、トランザクションエンジン301およびユーザインターフェース(UI)層302を備える。トランザクションエンジン301は、上で論じられた方式に従って、かつまもなくさらに詳しく論じられるように、トランザクション152を編成すること、サイドチャネル160を介してトランザクションおよび/もしくは他のデータを受信および/もしくは送信すること、ならびに/または、ブロックチェーンネットワーク106を通じて広められるようにトランザクションを1つまたは複数のノード104に送信することなどの、クライアント105の背後にあるトランザクション関連の機能を実装するように構成される。

【００８４】

UI層302は、機器102のユーザ出力手段を介して情報をそれぞれのユーザ103に出力すること、および機器102のユーザ入力手段を介してそれぞれのユーザ103から入力を受信することを含めて、それぞれのユーザのコンピュータ機器102のユーザ入力/出力(I/O)手段を介してユーザインターフェースをレンダリングするように構成される。たとえば、ユーザ出力手段は、視覚的な出力を提供するための1つもしくは複数の表示画面(タッチスクリーンまたは非タッチスクリーン)、オーディオ出力を提供するための1つもしくは複数のスピーカー、および/または触覚出力を提供するための1つもしくは複数の触覚出力デバイスなどを備え得る。ユーザ入力手段は、たとえば、1つもしくは複数のタッチスクリーン(出力手段のために使用されるものと同じまたは異なる)の入力アレイ、マウス、トラックパッド、もしくはトラックボールなどの1つもしくは複数のカーソルベースのデバイス、発話もしくは音声入力を受けるための1つもしくは複数のマイクロフォンおよび発話もしくは音声認識アルゴリズム、手もしくは体のジェスチャという形態の入力を受けるための1つもしくは複数のジェスチャベースの入力デバイス、または、1つもしくは複数の機械的ボタン、スイッチ、もしくはジョイスティックなどを備え得る。

【００８５】

注意:本明細書において様々な機能は同じクライアントアプリケーション105に統合されるものとして説明されることがあるが、これは必ずしも限定するものではなく、代わりに、それらは一連の2つ以上の別個のアプリケーションにおいて、たとえば一方が他方へのプラグインとなるように、またはAPI(アプリケーションプログラミングインターフェース)を介したインターフェーシングにより実装され得る。たとえば、トランザクションエンジン301の機能は、UI層302とは別のアプリケーションで実装されてもよく、または、トランザクションエンジン301などの所与のモジュールの機能は、1つより多くのアプリケーションの間で分割されてもよい。説明される機能の一部またはすべてが、たとえばオペレーティングシステム層において実装され得ることも、排除されない。本明細書においてどこかで単一のまたは所与のアプリケーション105などへの言及が行われる場合、これは単なる例であり、より一般的には、説明される機能は任意の形態のソフトウェアで実装されてもよいことが理解されるだろう。

【００８６】

図3Bは、Aliceの機器102a上のクライアントアプリケーション105aのUI層302によってレンダリングされ得るユーザインターフェース(UI)600の例のモックアップを与える。同様のUIが、Bobの機器102bクライアント105b、または任意の他の関係者の機器のクライアントによってレンダリングされ得ることが理解されるだろう。

【００８７】

例示として、図3BはAliceの視点からのUI350を示す。UI350は、ユーザ出力手段を介して別個のUI要素としてレンダリングされる1つまたは複数のUI要素351、352、352を備え得る。

【００８８】

たとえば、UI要素は、1つまたは複数のユーザ選択可能要素351を備えてもよく、これは、様々なオンスクリーンボタン、またはメニューの中の様々なオプションなどであってもよい。ユーザ入力手段は、ユーザ103(この場合はAlice 103a)が、画面上のUI要素をクリックもしくはタッチすること、または望ましいオプションの名前を話すことなどによって、オプションのうちの1つを選択し、または別様に操作することを可能にするようになされる(本明細書において使用される「手動の」という用語は、自動であることと対比させることのみを意図しており、手を使用することに必ずしも限定しない)。オプションは、ユーザ(Alice)が...を可能にする。

【００８９】

代替または追加として、UI要素は1つまたは複数のデータエントリフィールド352を備えてもよく、ユーザはそれを通じて...ことができる。これらのデータエントリフィールドは、たとえば画面上の、ユーザ出力手段を介してレンダリングされ、データは、ユーザ入力手段、たとえばキーボードまたはタッチスクリーンを通じて、フィールドに入力され得る。代替として、データは、口頭で、たとえば発話認識に基づいて受信され得る。

【００９０】

代替または追加として、UI要素は、情報をユーザに出力するための、1つまたは複数の情報要素353の出力を備え得る。たとえば、これ/これらは、画面上で、または可聴にレンダリングされ得る。

【００９１】

様々なUI要素をレンダリングし、オプションを選択し、データを入力する特定の手段は、有形ではないことが理解されるだろう。これらのUI要素の機能は、まもなくより詳しく論じられる。図3に示されるUI350は、概略的なモックアップにすぎず、実際には、それは1つまたは複数のさらなるUI要素を備えてもよく、これは簡潔にするために示されていないことも理解されるだろう。

【００９２】

ノードソフトウェア

図4は、UTXOベースまたは出力ベースのモデルの例では、ネットワーク106の各ブロックチェーンノード104で実行されるノードソフトウェア450の例を示す。別のエンティテ

ィは、ネットワーク106のノード104として分類されることなく、すなわち、ノード104に必要とされる活動を実行することなく、ノードソフトウェア450を実行し得ることに留意されたい。ノードソフトウェア450は、限定はされないが、プロトコルエンジン451、スクリプトエンジン452、スタック453、アプリケーションレベル決定エンジン454、および1つまたは複数のブロックチェーン関連機能モジュール455のセットを含み得る。各ノード104は、コンセンサスモジュール455C(たとえば、プルーフオブワーク)、伝播モジュール455P、および記憶モジュール455S(たとえば、データベース)の3つすべてを含むがそれらに限定されない、ノードソフトウェアを実行し得る。プロトコルエンジン401は通常、トランザクション152の様々なフィールドを認識し、ノードプロトコルに従ってそれらを処理するように構成される。別の先行するトランザクション152i(Tx_{m-1})の出力(たとえば、UXTO)を指し示す入力を有するトランザクション152j(Tx_j)が受信されるとき、プロトコルエンジン451は、Tx_jにおいてアンロッキングスクリプトを特定し、それをスクリプトエンジン452に渡す。プロトコルエンジン451はまた、Tx_jの入力の中のポインタに基づいて、Tx_iを特定して取り出す。Tx_iはブロックチェーン150上で公開されてもよく、この場合、プロトコルエンジンは、ノード104に記憶されているブロックチェーン150のブロック151のコピーからTx_iを取り出し得る。代替として、Tx_iはまだブロックチェーン150上で公開されていないことがある。その場合、プロトコルエンジン451は、ノード104によって維持される公開されていないトランザクションの順序付けられたセット154からTx_iを取り出し得る。いずれにしても、スクリプトエンジン451は、Tx_iの参照された出力においてロッキングスクリプトを特定し、これをスクリプトエンジン452に渡す。

【００９３】

したがって、スクリプトエンジン452は、Tx_iのロッキングスクリプトおよびTx_iの対応する入力からのアンロッキングスクリプトを有する。たとえば、Tx_0およびTx_1とラベリングされたトランザクションが図2に示されているが、同じことがトランザクションの任意のペアに当てはまり得る。スクリプトエンジン452は、前に論じられたように2つのスクリプトを一緒に実行し、これは、使用されているスタックベースのスクリプト言語(たとえば、Script)に従って、データをスタック453に置き、スタック453からデータを取り出すことを含む。

【００９４】

スクリプトを一緒に実行することによって、スクリプトエンジン452は、ロッキングスクリプトにおいて定義される1つまたは複数の基準をアンロッキングスクリプトが満たすかどうか、すなわち、ロッキングスクリプトが含まれる出力をアンロッキングスクリプトが「アンロック」するかどうかを決定する。スクリプトエンジン452は、この決定の結果をプロトコルエンジン451に返す。アンロッキングスクリプトが対応するロッキングスクリプトにおいて指定される1つまたは複数の基準を満たすとスクリプトエンジン452が決定する場合、それは「真」という結果を返す。それ以外の場合、それは「偽」という結果を返す。

【００９５】

出力ベースのモデルにおいて、スクリプトエンジン452からの「真」という結果は、トランザクションが有効であるための条件の1つである。通常、やはり満たされなければならないプロトコルエンジン451により評価される1つまたは複数のさらなるプロトコルレベル条件もある。それは、Tx_jの出力において指定されるデジタル資産の総額が入力によって指し示される総額を超えないこと、およびTx_iの指し示される出力が別の有効なトランザクションによってまだ消費されていないことなどである。プロトコルエンジン451は、スクリプトエンジン452からの結果を1つまたは複数のプロトコルレベル条件と一緒に評価して、それらがすべて真である場合にのみ、トランザクションTx_jを妥当性確認する。プロトコルエンジン451は、トランザクションが有効であるかどうかの標示を、アプリケーションレベル決定エンジン454に出力する。Tx_jが実際に妥当性確認されるという条件のもとで、決定エンジン454は、Tx_jに関してそれぞれのブロックチェーン関連機能を実行するようにコンセンサスモジュール455Cと伝播モジュール455Pの両方を制御することを選び得

る。これは、コンセンサスモジュール455Cが、ブロック151に組み込むためにTxjをトランザクションのノードのそれぞれの順序付けられたセット154に追加することと、伝播モジュール455Pが、Txjをネットワーク106の中の別のブロックチェーンノード104に転送することとを備える。任意選択で、実施形態では、アプリケーションレベル決定エンジン454は、これらの機能のいずれかまたは両方を起動する前に、1つまたは複数の追加の条件を適用し得る。たとえば、決定エンジンは、トランザクションが有効でありかつ十分なトランザクションフィーを残すという条件のもとでのみ、トランザクションを公開することを選び得る。

【００９６】

本明細書における「真」および「偽」という用語は、単一の二値の桁(ビット)のみの形式で表される結果を返すことに必ずしも限定しないが、それは当然1つのあり得る実装形態であることにも留意されたい。より一般的には、「真」は成功したまたは肯定的な結果を示す任意の状態を指すことができ、「偽」は不成功のまたは否定的な結果を示す任意の状態を指すことができる。たとえば、アカウントベースのモデルでは、「真」という結果は、署名の暗黙的なプロトコルレベルの妥当性確認と、スマートコントラクトの追加の肯定的な出力との組合せによって示され得る(両方の個々の結果が真であれば、全体の結果が真を示すものと見なされる)。

【００９７】

開示される技法の他の変形または使用事例は、本明細書の開示を与えられれば当業者に明らかになり得る。本開示の範囲は、説明される実施形態ではなく、添付の特許請求の範囲だけによって限定される。

【００９８】

たとえば、上のいくつかの実施形態は、ビットコインネットワーク106、ビットコインブロックチェーン150、およびビットコインノード104に関して説明されている。しかしながら、ビットコインブロックチェーンはブロックチェーン150の1つの特定の例であり、上の説明はあらゆるブロックチェーンに一般に当てはまり得ることが理解されるだろう。すなわち、本発明は、決してビットコインブロックチェーンに限定されない。より一般的には、ビットコインネットワーク106、ビットコインブロックチェーン150、およびビットコインノード104への上記のあらゆる言及は、それぞれ、ブロックチェーンネットワーク106、ブロックチェーン150、およびブロックチェーンノード104に関して置き換えられ得る。ブロックチェーン、ブロックチェーンネットワーク、および/またはブロックチェーンノードは、上で説明されたような、ビットコインブロックチェーン150、ビットコインネットワーク106、およびビットコインノード104の説明された性質の一部またはすべてを共有し得る。

【００９９】

本発明の好ましい実施形態では、ブロックチェーンネットワーク106はビットコインネットワークであり、ビットコインノード104は、ブロックチェーン150のブロック151を作成し、公開し、広め、記憶するという説明された機能の少なくともすべてを実行する。これらの機能のすべてではなく1つまたは一部だけを実行する他のネットワークエンティティ(またはネットワーク要素)があり得ることは排除されない。すなわち、ネットワークエンティティは、ブロックを作成して公開することなく、ブロックを広めるおよび/または記憶する機能を実行し得る(これらのエンティティは好ましいビットコインネットワーク106のノードであるとは考えられないことを思い出されない)。

【０１００】

本発明の好ましくない実施形態では、ブロックチェーンネットワーク106はビットコインネットワークではないことがある。これらの実施形態では、ノードが、ブロックチェーン150のブロック151を作成し、公開し、広め、記憶する機能のすべてではなく、少なくとも1つまたは一部を実行し得ることは排除されない。たとえば、それらの他のブロックチェーンネットワークでは、「ノード」は、ブロック151を作成して公開するが、それらのブロック151を記憶せず、かつ/または他のノードに広めないように構成される、ネット

ワークエンティティを指すために使用されることがある。

【０１０１】

またさらに一般的には、上記の「ビットコインノード」104という用語へのあらゆる言及は、「ネットワークエンティティ」または「ネットワーク要素」という用語で置き換えられてもよく、そのようなエンティティ/要素は、ブロックを作成し、公開し、広め、記憶する役割の一部またはすべてを実行するように構成される。そのようなネットワークエンティティ/要素の機能は、ブロックチェーンノード104に関して上で説明されたのと同じ方法でハードウェアにおいて実装され得る。

【０１０２】

またさらに一般的には、上記の「ビットコインノード」104という用語へのあらゆる言及は、「ネットワークエンティティ」または「ネットワーク要素」という用語で置き換えられてもよく、そのようなエンティティ/要素は、ブロックを作成し、公開し、広め、記憶する役割の一部またはすべてを実行するように構成される。そのようなネットワークエンティティ/要素の機能は、ブロックチェーンノード104に関して上で説明されたのと同じ方法でハードウェアにおいて実装され得る。

【０１０３】

順序付けられた追加専用のデータストレージ

大容量のデータ指向の用途にブロックチェーンを使用することが、近年顕著に増えている。この増大により、ブロックチェーンに公開されるデータペイロードを構造化し、符号化し、フォーマットするための、ロバストなレイヤ2プロトコルに対する需要も、比例して増えている。ここで、レイヤ2は、既存のブロックチェーンシステムの上に構築される二次的なプロトコル、フレームワーク、データ構造などを意味する。本明細書において説明される態様は、レイヤ2プロトコルであると考えられる。レイヤ1は、ビットコイン、ビットコインSV、または他の背後にあるブロックチェーン技術を指す。

【０１０４】

大量のデータが関わるブロックチェーンベースの適用例では、多くのデータ搬送トランザクションが互いにつながれることを可能にする、データスキームまたは構造化機構を通常は必要とする。これは特に、多くのイベントおよび/またはデータが直線化されたシーケンスにおいて互いにつながれる必要があり得るような適用例(たとえば、サプライチェーンにおける)に関係がある。

【０１０５】

イベントおよび/または順序付けられたデータ項目のシーケンスの維持および追跡を、一意な参照が支援することができ、それにより、1つのデータ搬送トランザクションが明示的に別のトランザクションを参照して、それらの2つのトランザクションがブロックチェーンの監督者によって互いに確実に関連付けられ得るようになる。

【０１０６】

Event Streamおよびダストのチェーン

図5は、本開示の第1の態様に関し、順序付けられた追加専用のデータストレージシステムの基本データ構造およびパラダイムを示す。これは、データロギングシステムとしても説明され得る。図5に示される特定のシステムは、ロギングイベントのためのEvent Streamシステムである。例として、Event Streamが、説明を目的として全体で使用されるが、本明細書において説明される提案されるシステムおよび態様は、一般にデータ項目とともに、および、順序付けられた追加専用のデータ項目ロギングまたはストレージシステムとともに使用され得ることを、当業者は理解するだろう。データ項目は、完全な形式のデータ、たとえばセンサデータまたは文書を指し得る。代替として、データ項目は、実際のデータのハッシュを指す。有利には、データ自体の代わりにハッシュを使用することは、データ(これは大きいことがあり、トランザクションには大きすぎることすらある)がトランザクションに記憶されるのを必要とすることなく、データの存在証明を提供する。

【０１０７】

追加専用ログにおける各イベント502は、ブロックチェーントランザクション504に対

応付けられ、ブロックチェーントランザクションのシーケンスは、「ダストのチェーン」を使用して順序付けられつなげられる(506)。各イベントに関連するデータは、各トランザクションの一部としてペイロード(以下で説明される)に記憶される。データペイロードは、トランザクションの消費不可能なOP_RETURN出力に保持される。これは、ブロックチェーンに任意のデータを書き込むために、かつまた、無効であるものとしてトランザクション出力をマークするために使用され得る、Scriptオペコードである。別の例として、OP_RETURNは、トランザクション内のメタデータなどのデータを記憶し、それによりブロックチェーンにメタデータをイミュータブルに記録できる、トランザクションの消費不可能な出力を作成するためのScript言語のオペコードである。

【０１０８】

ダストのチェーンは、ビットコインの入力および出力の壊れていないチェーンであり、これはここでは、シーケンスの中の各ブロックチェーントランザクションの、その直前のトランザクションに対する消費依存性を課すために使用される。本開示でのブロックチェーントランザクションの文脈における「ダスト」は、出力が低いまたはわずかな値であるデジタル資産または暗号通貨の消費可能なトランザクションであると理解され、すなわち、その値は、ブロックチェーンにおける出力をマイニングするためのはるかに少ないその料金であり得る。

【０１０９】

トランザクションにおいてダスト出力を使用することは、トランザクションが、Event Streamなどの順序付けられた追加専用のデータストレージシステムについて発生するにつれて、すべてのトランザクションのイミュータブルで逐次的な記録を維持するために有利かつ重要である。その理由は、トランザクションをブロックチェーンに投稿することによって、すべてのブロックチェーントランザクションはタイムスタンプを押され、ブロックチェーン上で確認されるとまたはブロックチェーンに追加されると特定の順序にとどまるが、これは、それらの逐次的な順序の保存を保証しないからである。これは、トランザクションが異なる時間にブロックへとマイニングされることがあるから、および/または、同じブロック内であってもトランザクションの順序が異なっているからである。シーケンスの中の次のトランザクションの最初の入力によって消費されるダスト出力を使用することで、有利には、トランザクションの順序が経時的に追跡され、イベント自体とイベントの逐次的な順序の両方の耐改竄性のある記録が作成されることが確実になる。これは、ブロックへとマイニングされると、シーケンスの中の前のトランザクションから次のトランザクションへのダストの支払いが、ビットコインプロトコルルールに従って、ペイロードと呼ばれ以下で論じられるような埋め込まれたデータ搬送要素のシーケンスを並べ替えることができずそして挿入または削除が起こり得ないことを確実にするからであり、そのような並べ替え、挿入または削除は、イベントストリームが危険にさらされていることが即座に明白になることなくシーケンスを変更する可能性がある。いくつかの実施形態では、ビットコインプロトコルに内在する二重消費防止機構が、異なるトランザクション入力と出力との間での暗号通貨(たとえば、ダスト)の移動がトポロジカルな順序にとどまることを確実にする。ダストトランザクションのチェーン形成は、トポロジカルな順序付けを利用して、ブロック間およびブロック内のトランザクション(およびしたがって関連するイベントおよびデータ)の順序の保存をもたらす。したがって、これは、順序付けられた追加専用のデータ項目ストレージの完全性を高める。

【０１１０】

このようにして、ブロックチェーントランザクション504は、トランザクションの有向グラフを形成する。エッジ506によって示されるように、グラフの方向は、シーケンスの中の前のトランザクションから次のトランザクションに向かう、一方向であるものと見なされ得ることに留意されたい。図5のエッジ506上の矢印は、次のトランザクションを指し示すトランザクションを示すが、ビットコイントランザクションにおける消費の関係は実際には、あるトランザクションから先行するトランザクションへの関係である。このグラフは、トランザクション間の消費関係によって作成される。これらの消費関係は、ある

タイプの参照であるものとして見なされ得る。

【０１１１】

ダストのチェーンにおける後方参照

図6Aは本開示の第1の態様に関し、トランザクション600のチェーンが示されている。トランザクションのチェーンは、互いに関連するいくつかのトランザクション602、604a、604b、604cを備える。第1のトランザクション602はTx_0とラベリングされ、チェーンおよびシード番号についてのメタデータを備える。チェーンは、いくつかの追加トランザクション604a、604b、604cも備え、好ましくは、それらはブロックチェーンに含まれるべきデータ項目を備える。好ましくは、データ項目は、以下で説明されるペイロードの一部として記憶される。追加トランザクション604a、604b、604cはまた、それらに先行するトランザクションからの出力に関連する入力を備えるので、消費関係606a、606b、606c、606dを確立する。この入力は、前のトランザクションからのトランザクション出力を消費する。例として、ビットコインプロトコルに関して、上記の「UTXOベースのモデル」という見出しのもとで説明されたように、かつ図2を参照すると、出力は未消費トランザクション出力(UTXO)であり、入力はUTXOへの参照を備える。したがって、各トランザクション(最初を除く)は、消費関係を介したダストのチェーンにおける前のトランザクションへの後方参照606a、606b、606c、606dを備える。初期トランザクションは、図6Cを参照して以下で説明されるように、資金提供UTXOへの後方参照を備える入力を備える。この資金提供UTXOは、ダストのチェーンの一部であると見なされないが、それはダストのチェーンに関するデータまたはメタデータを記憶しないからである。

【０１１２】

任意選択で、2つのさらなる後方参照が、追加トランザクション604a、604b、604cに含まれる。第1のトランザクションへの第2の後方参照608は、左手側の矢印によって図6Aにおいて示される。好ましくは、この参照は、第1のトランザクションのトランザクションidの形態をとる。第3の後方参照610、612a～cは、トランザクションがチェーンの中のどこにあるかに応じて2つの形態をとる。チェーンの中の第2のトランザクション604aに対して、第3の後方参照610は第1のトランザクションに存在するシードである。第2のトランザクションの後のすべての他のトランザクションに対して、参照612a～cは、先行するトランザクション内のデータの原像のハッシュという形態をとる。

【０１１３】

第3の後方参照610、612a～cは、各ペイロードが前のペイロードのセクションに依存することを確実にすることによって、変更からペイロードを保護するためのものである。具体的には、各ペイロードは、先行するペイロードのストリームダイジェストを備える。これにはまた、これらのストリームダイジェスト参照を追うことによってユーザがイベントのストリームを後方に追跡することを可能にする、後方参照を作成するという効果がある。これらの第3の後方参照610、612a～cは、消費参照606a～dと同じ物体(同じ先行するトランザクション)を参照するが、異なる参照物体を使用する。

【０１１４】

追加トランザクション604a、604b、604cのペイロードは任意選択で、$Payload_n$=[$preimage_n$][$streamDigest_n$][...]という形態を有する。ここで、小文字nは現在のトランザクションを表すために使用され、n-1は先行するトランザクションである。

【０１１５】

好ましくは、ペイロードは、OP_FALSE OP_RETURN OP_PUSHDATA1 〈preimage〉 0x20 〈streamDigest〉 [0x20 〈data digest〉 ¦ OP_PUSHDATAN 〈data〉]という形態で、トランザクションの出力のスクリプトに記憶される。

【０１１６】

原像は、現在のトランザクションおよび前のトランザクションのメタデータを備える。原像は任意選択で、以下のフィールドのうちのいずれか1つまたは複数を備える。

・txidcreate: チェーンにおける最初のトランザクションへの参照608、好ましくはチェーンにおける最初のトランザクションのトランザクションid

・index: データまたはイベントのインデックス
・whenRecorded: トランザクションおよび/またはデータ項目の作成に関連する時間
・$dataDigest_n$: データのハッシュ、任意選択でデータはペイロードのイベントデータ表現セクション([...])に記憶され、または任意選択でオフチェーンで記憶される
・$streamDigest_{n-1}$: 先行するトランザクションの原像のハッシュ612a、612b,612c(先行するトランザクションのストリームダイジェスト、またはストリームダイジェスト参照としても説明される)または最初のトランザクションのシード610(最初のトランザクションがstreamDigestを備えないのでトランザクションがチェーンにおける2番目のトランザクションである場合)。上で論じられたように、これは先行するトランザクションへの第3の参照として機能することができる。

【０１１７】
ストリームダイジェスト(streamDigest)は原像(preimage)のハッシュである。

【０１１８】
ペイロードのイベントデータ表現セクション([...])は任意選択で、ブロックチェーンに記憶されるべきデータ項目を備える。データ項目は、
・ブロックチェーンに記憶されるべきデータ自体、
・データのハッシュ、
・ブロックチェーンに記憶されるべきデータのサブセクション、または
・何もなしおよび/もしくは空
のうちの1つであり得る。

【０１１９】
記憶されるべきデータ項目が一部のデータのハッシュであり、トランザクションにデータ自体を記憶する意図がない場合、ペイロードのイベントデータ表現セクションは空であり得る。この場合のデータ項目(データのハッシュ)はすでに、原像のdataDigest部分に記憶されている。

【０１２０】
先行するトランザクションの原像のハッシュ612a、612b、612cは、先行するトランザクションへのさらなる参照であると見なされ得る。このハッシュは、トランザクションのチェーンを調査し、かつ/または先行するトランザクションを妥当性確認するために使用され得る。

【０１２１】
ブロックチェーンに記憶されるべきデータが、ペイロードには大きすぎる場合、および/またはトランザクションをブロックチェーンシステムには大きすぎるものにする場合、データはサブセクションへと分割され得る。したがって、データはいくつかのペイロード(およびしたがってトランザクション)にまたがって記憶される。原像は任意選択で、サブセクションの総数およびイベントデータ表現セクションに存在する現在のサブセクションのインデックスを追跡するためのさらなるフィールドを備える。データを複数のトランザクションにわたって分割するのを管理する他の方法が知られている。第1の代替形態は、データの固有IDを使用し得るので、データのあらゆるチャンクが、それらが関連していることを示すために、すべて同じIDを使用することができる。第2の代替形態は、データ(の残り)が記憶される他の位置へのポインタを使用してもよく、たとえば、データの半分を記憶する1つのTxは、残りの半分を記憶する別のTxのTxIDを含んでおり、このTxIDをその[...]に含んでいてもよく、その逆であってもよい。トランザクションにまたがってデータを分割するさらなる方法が可能であることを、当業者は理解するだろう。

【０１２２】
図6Bおよび図6Cを参照すると、データ追加トランザクション640a、640b、最初のトランザクション660、および最後のトランザクション662の例示的なブロックチェーントランザクションフォーマットが示されている。これらはブロックチェーントランザクションであるので、それらは図2を参照して説明されるトランザクション152i、152jに構造が類似しているが、本発明のこの態様に関連する特定の構成要素を備える。入力および出力

の厳密な順序は具体的ではなく、代替の順序が使用されてもよい。この順序は、好ましくは所与のチェーン上で一貫している。

【０１２３】

図6Bは、2つのデータ追加トランザクション640a、640bを示す。これらの例示的なデータ追加トランザクション640a、640bは、時間的に続けて、かつダストのチェーンにおいて到来する。第1のトランザクション640aのダスト出力644aは、第2のトランザクション640bのダスト入力646bにおいて参照される(すなわち、それによって消費される)。第1のトランザクション640aのダスト出力644aへの第2のトランザクション640bの参照のダスト入力646bは、第1のトランザクションのトランザクションid 648aとUTXOのインデックスの両方を備え、そのインデックスはこの例示的な場合には0である(それがリストの中で最初であり、ゼロインデクシングが使用されるので)。

【０１２４】

トランザクション640a、640b、660、662のすべてが、資金提供入力648a、648b、648c、648dを備える。これらの資金提供入力648a、648b、648c、648dは、コンピューティングデバイスがこれらのトランザクションを管理し、作成し、ブロックチェーンに出すことによって提供される。コンピューティングデバイスは任意選択で、資金提供サービスであり、図16を参照して説明されるようなサービスの一部である。資金提供入力の全体の値は、マイナーが確実にトランザクションを選んでそれをブロックに含めるのを助けるために、トランザクションフィー(マイナーの料金と呼ばれることがある)を含むように選択される。資金提供サービスは、全体の値が入力であるが十分であることを確実にするために、1つまたは複数の入力を提供し得る。トランザクションフィーは可変であり、ネットワークの負荷に依存する。トランザクションフィーは、バイト毎サトシ(またはブロックチェーンシステムが使用する任意のコイン/トークン)を単位とし得る(サトシは1ビットコインの1億分の1である)。したがって、ペイロードが大きい場合、料金も大きくなければならず、資金提供入力はそれに従って調整される。UTXOモデルの結果として、支払われる全体の料金は、入力において参照されるUTXOと、出力のUTXOの両方の値によって左右される。任意選択で、トランザクションフィーを含めることによる残高が、これらのトランザクションを管理し、作成し、ブロックチェーンに出す同じコンピューティングデバイスへ戻される。資金提供入力および前記資金提供入力に起因する残金は、浮動残高として機能し、前記資金提供サービスによって管理される。

【０１２５】

最初のトランザクション660および最後のトランザクション662はまた、ストリームメタデータ664、666を備える。最初のストリームメタデータは、ダストのチェーンの維持に関連する他の値の中でも、図6Bを参照して説明されたようなシードを備える。最後のトランザクション662のメタデータ666は、このトランザクションがチェーンにおいて最後であることを示すための情報を備える。好ましくは、最後のトランザクションのメタデータ666は、最初のトランザクション660のトランザクションid 648cも備える。

【０１２６】

データ追加トランザクション640aと640bの両方がそれぞれ、上で説明されたようなペイロード642a、642bを備える。第2のトランザクション640bのペイロード642bは、先行するトランザクション642aのペイロード642aへの参照を備える。

【０１２７】

図5、図6A、図6B、図6C、および図6Dのブロックチェーントランザクションのチェーン600において、トランザクションがシーケンスの中の次のトランザクションを参照することは明確に必要ではない。これは、トランザクショングラフにおける消費関係が常に、あるトランザクションから次のトランザクションに向かって前方にシーケンスを追跡するだけで十分であるからである。

【０１２８】

代替として、トランザクションは、シーケンスの中の次のトランザクションへの前方参照を備える。任意選択で、これは、シーケンスの中の次のデータ項目および/または追加ト

ランザクションの部分が現在の追加トランザクションを作成する時点で知られていることを要求することによって行われる。たとえば、最初のトランザクションTx_1がシーケンスTx_2の中の次のトランザクションを参照する必要がある場合、典型的な機構は、Tx_1内にハッシュ$H(Tx_2)$を含めることである。これは、Tx_1を作成する時点でTx_2の完全なデータの知識を必要とし、これは常に可能ではないことがある。

【０１２９】

図6Dを参照すると、トランザクションのセットに追加トランザクションを加える方法680が示されている。

【０１３０】

まず、ブロックチェーンに記憶するためのデータ項目が受信される(682)。データは、そのデータがどのチェーンに関連するかなどの、さらなるメタデータを含む要求の一部として受信され得る。代替として、この適用例はすでに、以前の要求および/または構成から関連するチェーンを認識している。データ項目は、イベントデータ表現セクションに記憶するためのデータであり得る。代替として、データ項目はデータのハッシュであり、データ項目はトランザクションの$dataDigest_n$セクションに記憶される。

【０１３１】

トランザクションのセットの中の最新のトランザクションが取得される(684)。任意選択で、最新のトランザクションは、それをメモリから呼び出すことによって取得される。

【０１３２】

新しいトランザクションが作成される(686)。新しいトランザクションは、最新のトランザクションからの出力に関連する入力、ダスト出力、データ項目、および最新のトランザクションへの参照を、少なくとも備える。好ましくは、それは最新のトランザクションからのダスト出力である。好ましくは、最新のトランザクションへの参照は、最新のトランザクションのセクションのハッシュを備える。より好ましくは、参照は最新のトランザクションの原像のハッシュである。

【０１３３】

新しいトランザクションは次いで、ブロックチェーンに出される(688)。

【０１３４】

祖先制限

ビットコインプロトコルの多くの実装形態において、祖先制限として知られている概念が存在する。祖先制限は、単一のブロック間の間隔において処理できる連続する消費依存性とともに、未確認のトランザクションの最大のチェーンに対して最大値を設ける。書き込みの時点で、この制限は、ビットコインでは25個の未確認の祖先に設定され、ビットコインSVでは1000個に設定される(以前は50個であり、その前は25個であった)。本開示の態様および実施形態はこの値に限定されないことが理解されるだろう。

【０１３５】

高頻度で消費依存トランザクションのチェーンを作成しようとするあらゆる適用例において、祖先制限の存在が問題になる。本明細書において説明される順序付けられた追加専用のデータストレージシステムは1つのそのような例であり、それは、ストリームの中のすべてのトランザクションが意図的に消費依存であるので、単一のブロック間の期間において25個より多くのイベントのログをとる必要があるストリームがこの問題により制限されるからである。

【０１３６】

いくつかの例では、(平均で)10分ごとなどに、新しいビットコインブロックが生み出される。したがって、ビットコインネットワーク上で動作する、図5、図6A、図6B、および図6Cを参照して説明される実施形態による、順序付けられた追加専用のデータストレージシステムは、たとえば、祖先制限に遭遇する前に、10分ごとに25個のデータ項目を記憶することができる。

【０１３７】

祖先制限の克服

図5、図6A、図6B、および図6Cを参照して説明されるような本発明の第1の態様では、ダストプロトコルのチェーンは、つなげられたトランザクションのシーケンスの中のオンチェーントランザクションに各データペイロードが含まれることを確実にする。トランザクションは次いで、あるトランザクションから次のトランザクションへのダスト出力を消費し、オンチェーントランザクション消費グラフを作成することによって、互いに逐次つなげられる。

【０１３８】

しかしながら、祖先制限を考慮すると、1つ1つのブロックチェーン作成期間において、祖先制限よりも頻繁にデータがブロックチェーンに追加されることがないか(したがって、ブロックチェーンにデータを記憶することを望むデータ書き込み者および/またはシステムに制約を課す)、または、チェーンの中のトランザクションの完全なセットが効率的に調査されることをそれでも可能にしながら、トランザクション消費グラフにおいて中断が導入されなければならないかのいずれかである。

【０１３９】

課題は、祖先制限がトランザクション消費グラフにおいて中断を導入するにもかかわらず、トランザクションフレームを使用するシーケンスにおいてデータ要素/ペイロードのシーケンスを確実に調査できるようにすることである。

【０１４０】

図7A、図7B、図8、および図9は、本開示の第2の態様に関し、データ構造700、714、716、データ項目をダストのチェーンに追加し、ダストの新しいチェーンを条件的に作成するための方法800、およびダストのチェーンを前方に調査するための方法900を説明する。任意選択で、図5、図6A、図6B、および図6Cを参照して説明される態様に加えて、ここで説明される本態様が使用され得る。本態様は、ある消費関係を伴うトランザクションのシーケンスを構築し、祖先制限を克服するために、他のブロックチェーンシステムとともに使用され得る。

【０１４１】

図7Aを参照すると、第2の態様において使用されるダストデータ構造700のチェーンの概要が示されている。トランザクションの2つの部分集合を備えるトランザクションのセットが示され、部分集合はダスト720、722のチェーンである。いくつかの追加トランザクション704a～dを備える、ダストの各チェーン。ダストの第1のチェーン720はチェンジアウトトランザクション714で終了し、ダストの第2のチェーン722はチェンジイントランザクション716で開始する。チェンジイントランザクション716とチェンジアウトトランザクション714との間には消費関係はないので、祖先制限の問題を克服する。これは、ダストの新しいチェーンへの「ホッピング」として説明され得る。

【０１４２】

チェーンホッピングは、部分集合の中のトランザクションの数が祖先制限に近づくときに行われる。好ましくは、現在のチェーンにおけるトランザクションの総数は祖先制限より1少なく、新しいチェーンが使用されなければならない。そたがって、チェンジアウトトランザクション714とチェンジイントランザクション716のペアが、Tx_{n-2}トランザクション704cとTx_{n-1}トランザクション704dとの間に挿入され、nは祖先制限である。これは、新しいチェーンの中の最初のトランザクション702、716(それが初期トランザクション702であるか、チェンジイントランザクション716であるかにかかわらず)を考慮して、チェンジアウトトランザクション714もダストのチェーンに含まれるようにするための空間を残すことである。

【０１４３】

代替として、チェーンホッピングは、チェーンの中の未確認のトランザクションの数が祖先制限に近づくと行われる。このようにして、チェーンホッピングに関連するトランザクションの部分集合は、各トランザクションが確認されたかどうかによって定義される。

【０１４４】

トランザクションの部分集合は真部分集合ではないので、ダストの1つのチェーンしか

ない場合、部分集合のメンバーはトランザクションの集合と同じである。

【０１４５】

チェンジイントランザクション716はまた、先行するチェーンへのチェーン参照724を備える。具体的には、チェーン参照724は、先行するチェーンの中のトランザクションを参照することによってチェーンを参照する。好ましくは、チェーン参照724は、第1のチェーン720における最初のトランザクション702に対するものである。第1のチェーン720における最初のトランザクション702は、初期トランザクションまたはTx_0とも呼ばれ得る。代替として、チェーン参照724は、先行するチェーンにおける最初のトランザクション、すなわち、先行するチェーンのチェンジイントランザクションに対するものである(先行するチェーンも最初のチェーンであるので、これは図7に示されていない)。

【０１４６】

各追加トランザクション704a～dは、ダストのチェーンにおける先行するトランザクションへの参照706a～fを備える。この参照706a～fは、図5、図6A、および図6Bを参照して説明されるような消費関係606を少なくとも備える。任意選択で、トランザクション604a～dはさらに、図6Aを参照して説明されたような、第3の後方参照($streamDigest_{n-1}$参照としても説明される)610、612a～cを備える。

【０１４７】

チェンジアウトトランザクション714は、チェンジイントランザクション716への前方参照718を備える。この例では、この参照718は、チェンジイントランザクション716のトランザクションidに対するものである。参照718は、トランザクション716のハッシュであり、またはそれを備えるので、チェンジイントランザクション716が最初に作成されなければならない。これは、チェンジイントランザクション716が最初にブロックチェーンに公開されることまたは出されることを必要としない。

【０１４８】

この前方参照718は、このデータストレージシステムに記憶されている情報を抽出することを望む関係者が、チェーンを前方に調査することを可能にする。

【０１４９】

図7Aは、ダスト720、722の2つのチェーンのみを示すが、本明細書において説明されるものと同じ技法を使用して、チェーンが互いに対して追加され得ることを当業者は理解するだろう。

【０１５０】

図7Bを参照すると、チェンジアウトトランザクション714およびチェンジイントランザクション716のための例示的なブロックチェーントランザクションフォーマットが示されている。これらのトランザクションは、図6Bおよび図6Cを参照して説明されるトランザクション604a～dと同じまたは類似する特徴の多くを有する。特徴が同じでありまたは同様である場合、同じまたは同様の名称が使用されている。

【０１５１】

チェンジアウトトランザクション714のダスト入力732は、ダストの先行するチェーンにおける最後の追加トランザクションのダスト出力への参照を備える。チェンジアウトトランザクション714は、その出力の一部として、チェンジイントランザクション716への参照を備える。具体的には、チェンジアウトトランザクション714は、チェンジイントランザクション716のトランザクションid 730bを備える。

【０１５２】

本態様の追加トランザクション704a～dは、ダスト入力646a、646bが以前の追加トランザクションのダスト出力644a、644bを常には参照しないことを除き、図6Bを参照して説明された追加トランザクション640a、640bと非常に類似したフォーマットを有する。チェンジアウトトランザクション714およびチェンジイントランザクション716が図8を参照して説明されるように使用されるとき、次の追加トランザクション704dが、チェンジイントランザクションに続く。すなわち、次の追加トランザクション704dの形態のダスト入力646a/646bは、チェンジイントランザクションのダスト出力734を消費する。

【０１５３】
図8を参照すると、図7を参照して説明されたように、データ項目をダストのチェーンに追加し、構造700に従ってダストの新しいチェーンを条件的に作成するためのコンピュータで実施される方法800。これらのステップは、逐次的なステップとして論じられ示されているが、それらの多くは、任意の順序で、並列に、または同時に行われてもよい。たとえば、最初の3つのステップ802、804、および806は、任意の順序であってもよい。
【０１５４】
まず、ブロックチェーンに記憶するためのデータ項目が受信される(802)。データは、そのデータがどのチェーンに関連するかなどの、さらなるメタデータを含む要求の一部として受信され得る。代替として、この適用例はすでに、以前の要求および/または構成から関連するチェーンを認識している。
【０１５５】
次に、最大の未確認のチェーンの長さが取得される(804)。この値は、あらかじめ構成され、メモリに記憶されている変数を介してアクセスされ、ストレージから呼び出され、または第三者から取得され得る。この最大の未確認のチェーンの長さは、上で論じられたような祖先制限である。好ましくは、このステップは、祖先制限が変化しない限り一度行われ、データを記憶するための要求が受信されるたびに行われることはない。
【０１５６】
現在のチェーンにおける最新のトランザクションのトランザクションidが取得される。トランザクションidは、チェーンの中の次のトランザクションへの入力を最新のトランザクションの出力と関連付けるために使用される。好ましくは、トランザクションidは、最新のトランザクションがブロックチェーンに出されたときから保持される。代替として、最新のトランザクションは、任意選択でチェーンの中の最新のトランザクションまで調査することによって取得される。次いで、トランザクションidが、最新のトランザクションをハッシュすることによって決定される。
【０１５７】
次いで、現在のチェーンの長さが決定される(806)。好ましくは、この値は、迅速な取り出しのためにメモリにおいて変数として記憶される。追加または代替として、現在のチェーンにおける最新のトランザクションが知られている場合、トランザクションのインデックスは、前記トランザクションのペイロードの原像データから読み取られる。さらなる代替として、このステップが実行されるたびに、チェーンの長さ全体が数えられる。任意選択で、現在のチェーンの長さは、現在のチェーンにおける未確認のトランザクションの数を表す。
【０１５８】
現在のチェーンの長さを決定することの代替として、ブロックチェーン上のブロックに含められていない現在のチェーンの中のトランザクションの数が代わりに決定される。方法800は、これを「現在のチェーンの長さ」として使用することによって、平常通り継続する。
【０１５９】
現在のチェーンの長さを決定することのさらなる代替として、ブロックチェーン上で「確認されて」いない現在のチェーンの中のトランザクションの数が代わりに決定される。「確認された」の定義は、対象のブロックチェーンに依存する。例として、多くのビットコイン関連の適用例は、ブロックの後に6つのブロックが追加されると、そのブロックが「確認された」と考えられるものと見なす。
【０１６０】
現在のチェーンの中の未確認のトランザクションの数にわたり現在のチェーンの長さを使用することの利点は、ブロックチェーンのフォーキングに固有の回復力があり、マイニングされたブロックの一部であった、かつ/またはブロックチェーンにおいて「確認された」と見なされるトランザクションが、もはやブロックチェーンの一部ではないということである。これは、必ずしも必要ではないときにチェンジイントランザクションおよびチェ

ンジアウトトランザクションを使用する可能性があるという代償を伴う。これらの代替形態は、ブロックチェーンに出されるデータの性質および頻度に応じて選択され切り替えられ得る。

【０１６１】

現在のチェーンの長さが閾値の数以上であるかどうかの決定(808)が行われる。好ましくは、閾値は、少なくとも1つの未確認のトランザクションが未確認のトランザクションのチェーンに残ることが許容され、その結果、チェンジインアウトトランザクションがまだ許容されるように選ばれる。好ましくは、閾値の数は、最大の未確認のチェーンの長さより1少ない。

【０１６２】

現在のチェーンの長さが閾値未満である場合、図6Aおよび図6Bを参照して説明されたようなトランザクションに従って、トランザクションが作成される(810)。すなわち、新しく作成されるトランザクションは、チェーンの中の最新のトランザクションの出力を参照する入力を備える。

【０１６３】

トランザクションはブロックチェーンに出され(812)、その結果、他のコンピューティングデバイスはトランザクションをマイニングし、それをブロックチェーンに追加することができる。

【０１６４】

任意選択で、現在のチェーンの長さが記憶されている場合、現在のチェーンの長さがインクリメントされる(814)。

【０１６５】

現在のチェーンの長さが閾値以上である場合、チェンジアウトトランザクションおよびチェンジイントランザクションが作成される。

【０１６６】

まず、チェンジイントランザクションが、図7Aおよび図7Bを参照して論じられたように作成される(816)。

【０１６７】

次いで、チェンジアウトトランザクションが、図7Aおよび図7Bを参照して論じられたように作成される(818)。チェンジアウトトランザクションは、チェンジイントランザクションへの前方参照を備える。前方参照は、チェンジイントランザクションのトランザクションidを備え、またはそのトランザクションidである。

【０１６８】

ブロックチェーンに出されるべきデータを備える新しいトランザクションが作成される(820)。新しいトランザクションは、チェンジイントランザクションの出力に関連する入力を備える。新しいトランザクションはまた、ブロックチェーンに出されるべきデータを備える。

【０１６９】

任意選択で、現在のチェーンの長さが記憶されている場合、現在のチェーンはチェンジイントランザクションおよび追加トランザクションを備えるので、現在のチェーンの長さは2に設定される。

【０１７０】

任意選択で、新しいトランザクションはまた、現在のチェーンにおける最新のトランザクションへの参照を備える。この参照は、図6Aおよび図6Bを参照して説明されたような第3の後方参照610、612a～cの形態である。

【０１７１】

現在のチェーンの長さが閾値未満であるかどうかを確認すること(808)を含む、チェンジアウトトランザクションおよびチェンジイントランザクションを作成することに関するステップ816、818、820、822、824は、追加トランザクションを出すステップ(812)および/または現在のチェーンの長さをインクリメントするステップ(814)の後でも実行さ

れ得ることを、当業者は理解するだろう。このようにして、新しいダストのチェーンが、次の追加トランザクションのためにすでに確立されている。

【０１７２】

図9を参照すると、図7A、図7B、および図7Cを参照して説明され、図8を参照して説明されたような方法を用いて作成される構造について、ダストのチェーンを前方に調査するための方法900。前方調査は、シーケンスまたはインデックス番号が増大する方向に移動することを意味する。これは、より古いデータ項目からより新しいデータ項目に移動することとしても見ることができる。追加トランザクションにおける後方参照の使用、およびチェンジイントランザクションへのチェンジアウトトランザクションにおける前方参照が、この前方調査を可能にする。

【０１７３】

有利には、調査の方法900は、ブロックチェーン上でのトランザクションのフォーマット以外に、トランザクションを調査するための隠された知識を必要としない。調査において使用される参照はすべて、ブロックチェーン上でデータ項目とともに公開される。したがって、第三者も、私的なサービスまたはプロプライエタリサービスなしで、ダストのチェーンを調査し、ブロックチェーンに記憶されているデータ項目を取得し、記憶し、検証し、および/または別様に使用することが可能である。とりわけ、ブロックチェーンの性質ならびに現在のデータ書き込みシステムおよび方法により、ブロックチェーンに記憶されているデータに対して行われるあらゆる操作が読み取り専用である。ブロックチェーンに記憶されているデータを改変することは現実的ではない。

【０１７４】

最初のトランザクションが取得される(902)。これは、そこから前方に調査すべきトランザクションである。

【０１７５】

任意選択で、シード(それがダストのチェーンの中の最初のトランザクションである場合)、またはストリームダイジェスト(それが追加トランザクションである場合)が取得され、後で検証するために記憶される(904)。

【０１７６】

次に、最初のトランザクションは、そのトランザクションidを取得するためにハッシュされる(906)。

【０１７７】

最初のトランザクションのトランザクションidを使用して、入力においてこのトランザクションidを参照するダストのチェーンの中のトランザクションを見つける(908)。これが、ダストのチェーンにおける次のトランザクションである。次のトランザクションは、次のステップ910のための現在のトランザクションとして割り当てられる。

【０１７８】

トランザクションを見つけるステップ(908)は任意選択で、それを見つけるために、ブロックチェーンの中のブロックのすべてについて繰り返すことを伴う(前方に調査していることを考慮するとそれは先行するブロックには位置しないので、現在のトランザクションがあるブロックから開始する。後方に調査している場合は逆である)。

【０１７９】

代替または追加として、トランザクションを見つけるステップは、ダストのチェーンを複製する、またはダストのチェーンにおける各トランザクションに関連付するメタデータを記憶する、オフチェーンログまたはデータベースを調べることを備える。

【０１８０】

いくつかの実施形態では、オフチェーンログまたはデータベースは、ダストのチェーンの一部である各トランザクションを、トランザクションがその一部であるブロックヘッダまたは少なくともブロックidと関連付ける。このようにして、トランザクション参照(任意選択でトランザクションid、またはいくつかの態様では、PrevOut資金提供入力)を使用して、関心対象のトランザクションを備えるブロックが取得され、次いでトランザクション

を求めて検索される。この実施形態は、適切なトランザクションを見つけるためにブロックチェーンの1つ1つのブロックについて繰り返すことと比較して、ステップを見つけることの少なくとも一部を省略する。

【０１８１】

図15および図16を参照して説明されたようなプラットフォームサービスの一部として動作するプラットフォームプロセッサは任意選択で、このログまたはデータベースを維持する。

【０１８２】

いくつかの実施形態では、英国特許出願第2007597.4号(2020年5月21日にnChain Holdings Limitedの名義で出願された)を参照して説明されるような、チャネルサービスが存在する。このチャネルサービスは、トランザクションがブロックチェーン上で確認されるとき、通知を第三者に提供する。この通知は、トランザクションがその中で確認されたブロックのブロックヘッダおよびトランザクションidなどの詳細を備える。これらの通知を購読する第三者は、情報取得された通知を使用して、正しいブロックに直接向かうことによって、上記と類似する方法でブロックチェーン上のトランザクションを見つけることができる。

【０１８３】

次いで、現在のトランザクション(図9でTx_iとラベリングされる)に対して作用するループに入る。ループにおける最初のステップは、現在のトランザクションが最後のトランザクションであるかどうかを確かめる(908)。現在のトランザクションが最後のトランザクションではない場合、方法は次のステップ912に続く。それが最後のトランザクションである場合、ループおよび方法900は終了する。代替として、最後のトランザクションが最初のトランザクションへの参照を備える場合、方法900は任意選択で、トランザクションのセットの最初に戻り、必要であればチェーンの調査を続けることができる。これは、たとえば、方法がダストのチェーンの最初のトランザクションで開始せず、チェーンを調査するユーザがトランザクションのすべてを調査することを望む場合に必要とされ得る。この場合、調査が開始したトランザクションに再び達する(すなわち、ループが開始位置に戻り、チェーンの中のトランザクションのすべてが調査される)と、ループは終了する。

【０１８４】

トランザクションが最後のトランザクションであるかどうかを決定するステップ(910)は、トランザクションがメタデータを備えるかどうかに基づく。代替として、最後のトランザクションは、それが最後のトランザクションであることを示すためのフラグを備える。最後のトランザクションが最後のトランザクションであるかどうかを決定することのさらなる代替形態は、トランザクションが消費されないダスト出力を備えるかどうか、およびトランザクションがチェンジイントランザクションへの参照を備えない(そうでなければトランザクションはチェンジアウトトランザクションである可能性が高い)かどうかに基づく。

【０１８５】

次に、タイプ現在のトランザクションであるかについての決定(912)が行われる。任意選択で、これは、現在のトランザクションが最後のトランザクションであるかどうかを確かめることの一部として確かめられる。トランザクションがペイロードを備える場合、それは追加トランザクションであると見なされる。トランザクションがペイロードを備えない場合、それはチェンジアウトトランザクションであると見なされる。代替として、トランザクションが次のチェンジイントランザクションへの参照を備える場合、それはチェンジアウトトランザクションである。

【０１８６】

トランザクションが追加トランザクションである場合、ペイロードに対して任意選択の動作が行われる(914)。任意選択で行われる動作は、調査者によって提供される構成に依存する。

【０１８７】

好ましくは、動作は以前のトランザクションを検証することである。この検証は、図6Aおよび図6Bを参照して説明されたような第3の後方参照610、612a～cに基づく。現在のペイロードのストリームダイジェストは、次のループ反復における検証のために記憶される。現在のトランザクションのペイロードに記憶される、以前のトランザクションのストリームダイジェストは、前記値を取得して記憶する以前のループ反復または任意選択のステップ904から記憶されている、以前のストリームダイジェストまたはシードと比較されて使用される。

【０１８８】

追加または代替として、ペイロードのデータ項目は後で使用するために記憶される。追加または代替として、調査者によって提供されるコールバックが呼び出され、それにより、任意の動作が調査者によってペイロードに対して行われることを可能にする。

【０１８９】

現在のトランザクションのトランザクションidを取得するために、現在のトランザクションがハッシュされる(916)。

【０１９０】

次のトランザクションは、現在のトランザクションのトランザクションidを参照する入力を有するトランザクションを探すことによって見つけられる(918)。

【０１９１】

ループは繰り返し、次のトランザクションが現在のトランザクションとして割り当てられる(920)。ループは、現在のトランザクションが最後のトランザクションであるかどうかを決定するステップ(910)で開始する。

【０１９２】

トランザクションのタイプを決定するステップ(912)に戻り、現在のトランザクションがチェンジアウトトランザクションである場合、チェンジイントランザクションへの参照が取得される。参照は、好ましくは、参照が記憶されているチェンジアウトトランザクションから参照を抽出する(922)ことによって取得される。

【０１９３】

チェンジイントランザクションへの参照により、チェンジイントランザクションが発見される(924)。本態様では、チェンジイントランザクションは、チェンジイントランザクションのトランザクションidである。チェンジイントランザクションは、それが確認されている場合はブロックチェーン上で見つかることがあり、または、トランザクションがまだ確認されていない場合はブロックチェーンネットワークノードのメモリプールにおいて見つかることがある。

【０１９４】

チェンジイントランザクションは、チェンジイントランザクションのトランザクションidを取得するためにハッシュされる(926)。ダストのチェーンの中の次のトランザクションは、チェンジイントランザクションidの形態で、チェンジイントランザクションへの後方参照を備える。具体的には、次のトランザクションはチェンジイントランザクションに関連する出力を消費する。

【０１９５】

チェンジイントランザクションのトランザクションidにより、次のトランザクションが取得される(928)。

【０１９６】

ループは繰り返し、次のトランザクションが現在のトランザクションとして割り当てられる(920)。ループは、現在のトランザクションが最後のトランザクションである(これは、上で説明されたように、ループが終了できる点でもある)かどうかを決定するステップ(910)で開始する。

【０１９７】

方法900は、トランザクションのセットの中の最初のトランザクションから開始するものとして説明されている。しかしながら、この方法900は、ループの開始910において開

始することによって、開始点としてあらゆるトランザクションからも、この方法900を使用できることを、当業者は理解するだろう。

【０１９８】

トランザクションタイプを決定するステップ(912)は、好ましくは、現在のトランザクションの内容に基づく。好ましくは、現在のトランザクションの出力のサイズが、それが追加トランザクションであるか、またはチェンジイントランザクションであるかを決定するために使用される。現在のトランザクションがチェンジイントランザクションである場合、出力のうちの1つは、少なくとも1つのトランザクション出力点〈PrevOut(s)〉を備える。代替として、第1の態様を参照して説明されたような、原像、streamDigest、およびイベントデータ表現セクションの存在が、トランザクションが追加トランザクションであるかどうかを決定するために使用される。代替として、本明細書において説明されるような〈PrevOut(s)〉フィールドの存在は、トランザクションがチェンジイン(または前方調査ではチェンジアウト)であるかどうかを決定するために使用される。代替として、現在のトランザクションの出力の中のフィールドの数が、トランザクションのタイプを決定するために使用され、それは、追加トランザクションペイロードの中により多くのフィールドがあるからである。代替として、現在のトランザクションのダスト出力がないことは、現在のトランザクションがチェンジアウトトランザクションであることを示すために使用される。代替として、出力の数がトランザクションのタイプを決定するために使用され、それは、チェンジアウトが追加トランザクションより1少ない出力を有するからである。代替として、現在のトランザクションがとることができるあり得る構造のすべてのテンプレートが、現在のトランザクションとパターンマッチングしてそのタイプを決定するために使用される。

【０１９９】

トランザクション展性

トランザクション展性の概念は、トランザクションを無効にすることなく有効なトランザクション(たとえばビットコイントランザクション)の一部を改変できるような現象に関係する。通常、トランザクションidがトランザクションを参照するために使用され、トランザクションidはトランザクション全体のハッシュである。したがって、トランザクションを参照するときに問題が発生し、それは、トランザクションのいずれの部分へのいずれの改変も、トランザクションのハッシュを変化させ、これが前記トランザクションへのあらゆる参照を無効にすることがあるからである。

【０２００】

一般に、ブロックチェーントランザクションにおいて大まかに2つのタイプの展性があり、それらの両方が、入力において、またはそれを介して提供される署名を無効にすることなく、トランザクションの内容が改変されることを可能にする。

【０２０１】

この例では、両方の場合において、初期トランザクションTxが、1つの入力と、その入力の中の1つの署名と、1つの出力とを有することが仮定される。

【０２０２】

タイプ1:スクリプトレベルの展性

このタイプの展性は、スクリプトオペコードOP_CHECKSIGを用いて確かめられるべきビットコイン署名が、署名を含むトランザクションの中のいずれの入力のscriptSigフィールドも署名しないという事実を利用する。

【０２０３】

この事実により、トランザクションTxへの署名を生成し、トランザクションTx'がTxと同一ではないように入力スクリプトを修正し、それでも、TxとTx'の両方がブロックチェーンコンセンサスルールのもとで同じ署名によって署名される有効なトランザクションメッセージであると見なされるようにすることが可能になる。

【０２０４】

Type2:入力レベルおよび出力レベルの展性

このタイプの属性は、トランザクションにおいて利用されているSIGHAS_ALL以外のSIGHASHフラグの使用に依拠する。

【０２０５】

トランザクションTxが5つの他のSIGHASHフラグの組合せのいずれかを使用する入力署名を有する場合、入力または出力のいずれかが、同一ではないトランザクションTx'を作成するために追加され得るので、署名を変更する必要なく、両方がコンセンサスに従って有効なトランザクションメッセージであると見なされる。

【０２０６】

したがって、本明細書において説明される態様の1つ1つのトランザクションが、好ましくは、この属性を克服するためにSIGHASH_ALLフラグを使用する。

【０２０７】

有利には、以下の第3、第4、第5、および第6の態様は、参照の不変性を確実にするためのさらなるおよび/または代替の技法を提供する、データ構造および関連する方法を備える。

【０２０８】

トランザクションへの「不変の参照」という用語は、トランザクションがビットコインネットワークに公開されると改変することができない(すなわち、不変である)トランザクションの中のデータに基づく、有効なトランザクションへの参照を意味する。

【０２０９】

トランザクションへの「イミュータブルな参照」という用語は通常、確認されたトランザクションのトランザクションIDを意味し、これは、高い確実度で変えることができない。不変の参照はイミュータブルな参照であり得る。

【０２１０】

チェンジイントランザクションの一部が、チェンジアウトトランザクションのペイロードの中のチェンジイントランザクションのトランザクションidの含有と、ブロックチェーン上のチェンジイントランザクションの含有(確認としても知られている)との間で改変される場合、チェンジイントランザクションのトランザクションidは、チェンジアウトトランザクションに記憶されているものと、ブロックチェーンに記憶されているチェンジイントランザクションとの間で異なり、それにより、2つのダストチェーンをつなぐ参照を破壊する。

【０２１１】

言い換えると、チェンジアウトトランザクションの前方参照は、それが改変されたトランザクションidに基づいて新しいトランザクションidを指し示しているはずであるのに、誤ったチェンジイントランザクションidに対するものである可能性がある。そのような改変が起こると、これは参照を含める目的を完全に失わせ、それは、参照がイベントストリームの調査を続けるための正しいオンチェーントランザクションをもはや指し示さないからである。

【０２１２】

トランザクションがビットコインネットワークの大部分に広められる前に悪意のあるマイナーまたは盗聴者がトランザクションをうまく改変するような攻撃が発生するリスクは、0ではない。したがって、この問題に対する堅牢な解決策を見つけることが望ましい。

【０２１３】

追加トランザクションの属性は、「トランザクションの属性」という見出しのもとで上で論じられたように、属性のない消費参照が原因では生じない。チェンジアウトトランザクションおよびチェンジイントランザクションは、そのような消費のつながりを有することができない(そうでなければ、祖先制限を克服するためにそれらを使用することはできない)。

【０２１４】

ブロックチェーン上でのチェンジイントランザクションの確認

図10を参照すると、イミュータブルなチェンジイントランザクション1016およびチェ

ンジイントランザクション1014への不変かつイミュータブルな参照1018を作成するための、第3の態様によるデータ構造1000が示されている。イミュータブルなチェンジイントランザクションにより、チェンジイントランザクションへのイミュータブルな参照を作成することができ、それにより、上で「トランザクションの展性」という見出しのもとで論じられたようなチェンジイントランザクションの展性の問題を克服する。

【０２１５】

図10は、時系列で示されており(より早く出されたトランザクションがページでより上に現れる)、ブロックチェーンの中のどのブロックがどのトランザクションを備えるかを記述しているという点で、前の図とは異なるデータ構造を提示する。どのブロック1050、1052がどのトランザクションを備えるかが、この態様では重要である。

【０２１６】

図10は、第2の態様の特徴と同様の特徴に対して、同様の参照番号および名前を備える。たとえば、各追加トランザクション1004a～dおよび最後のトランザクション1026は、先行するトランザクションへの後方消費参照1006a～fを備える。これは、先行するトランザクションへの後方消費参照706aも備える、図7Aに示されるような追加トランザクション704a～dおよび最後のトランザクション726と同様である。

【０２１７】

本態様は任意選択で、第1の態様の第2の後方参照(608)の特徴および第3の後方参照(610、612a～c)の特徴を含む。

【０２１８】

本態様では、チェンジイントランザクション1016は、チェンジアウトトランザクション1014がそれを参照する前に、ブロックチェーン上でブロック1050において確認される。チェンジイントランザクションがブロックチェーン上で実行されると、悪意のあるマイナーがトランザクションを変えること、したがってトランザクションidを変えることは現実的に不可能である。したがって、参照1018は、参照が有効であり、常に有効であろうという点で、イミュータブルである。参照またはトランザクションの変更は不可能である。

【０２１９】

それでも、チェンジイントランザクションの出力は消費され得るので、次の追加トランザクション1004cは、ブロックにおいて確認されるのを待つ必要なく、チェンジイントランザクション1016を参照することができる。したがって、それでも、追加トランザクションを、チェンジインが確認されるのを待つことなく出すことができる。抽象化のダストレベルのチェーンにおける得られたデータ構造1000は、図7A、図7B、および図9を参照して説明されたのと同じように見え、動作する。

【０２２０】

データ項目をダストのチェーンに加え、第2の態様において説明されるようなダストの新しいチェーンを条件的に作成するための方法は、チェンジイントランザクションがブロックチェーン上のブロックに含められるまで、チェンジアウトトランザクション1014を作成して出すステップ818が遅れることを除き、この態様において実質的に同じように動作する。このステップ818は、方法の残りに対して同期せずに実行されてもよく、それは、この態様における他のトランザクションがチェンジアウトトランザクション1014を参照しないからである。チェンジイントランザクション1016が、その出力を消費することができるブロックチェーンに出されると、方法は平常通りに継続する。

【０２２１】

チェンジイントランザクションは、図7Aを参照して第2の態様において説明されたような、チェーン参照724と同様の先行するチェーンへのチェーン参照1024を備える。

【０２２２】

データ構造1000の調査の方法は、第2の態様において図9を参照して説明されたものと実質的に同じである。

【０２２３】

展性のない前方チェンジイン参照

図11を参照すると、チェンジイントランザクション1116への不変の参照1118を作成するための第4の態様によるデータ構造1100が示されている。

【０２２４】

第4の態様は、第2の態様および第3の態様におけるものと同様の、チェンジアウトトランザクション1114からチェンジイントランザクション1116への前方参照1118を備えるが、前方参照718はチェンジイントランザクション1116のトランザクションidである代わりに、参照はチェンジイントランザクション1116の入力のうちの少なくとも1つを備える。好ましくは、入力のうちの少なくとも1つへの参照は、トランザクション出力点が出力を備えるトランザクションidと前記出力のインデックスとを備えるような、トランザクション出力点の形態である。好ましくは、参照1118は、チェンジイントランザクション1116によって消費される以前のトランザクション出力のセットを備える。以前のトランザクション出力のセットは$PrevOuts_{change\text{-}in}$と呼ばれる。$PrevOuts_{change\text{-}in}$の例示的なフォーマットは、

$PrevOuts_{change\text{-}in}=\{PrevOut_1:(TxID_{Prev,1},vout_1),PrevOut_2:(TxID_{Prev,2},vout_2)\}$

である。

【０２２５】

$TxID_{Prev,n}$が消費されている出力を備えるトランザクションのトランザクションidであり、$vout_n$が前記トランザクションの中の前記出力のインデックスである場合。タプル$(TxID_{Prev,n},vout_n)$はトランザクション出力点である。トランザクションに対するPrevOutsのセットの中の各PrevOutはそれ自体が一意であり、それは、各々がブロックチェーン上のトランザクション出力であり、これらが一意であるからである。したがって、PrevOutsのセットも一意である。

【０２２６】

好ましくは、前方参照は、以下の形式でチェンジアウトトランザクションの出力のスクリプトセクションに記憶される。

OP_FALSE OP_RETURN 0x20 〈$TxID_{Prev1}$〉〈index length〉〈$vout_1$〉

【０２２７】

1つより多くの資金提供入力が、チェンジイントランザクションに資金提供するために使用される場合、さらなる入力がそのように上記のフォーマットに追加される。

〈$TxID_{Prevn}$〉〈index length〉〈$vout_n$〉

【０２２８】

より好ましくは、スクリプトは、OP_RETURNの後に、かつPrevOut(s)参照の前に、0x20 〈$txid_{create}$〉を備える。

【０２２９】

したがって、チェンジイントランザクションへの参照1118は、トランザクションidに基づかず、またはチェンジイントランザクション1116のいずれの他の展性のあるセクションもしくは態様にも基づかない。このようにして、悪意のあるマイナーがトランザクションを改変しても、参照1118は依然として有効であり、それにより、上記の「チェンジイントランザクションの展性」という見出しのもとで論じられたようなチェンジイントランザクションの展性の問題を克服する。

【０２３０】

トランザクションがファイナライズされる(すなわち、有効であり、その署名のすべてが、新しい入力/出力を追加できない適切なフラグを使用する)場合、このことは、本明細書において説明される態様のいずれのトランザクションにおいてもそうであると想定でき(あらゆる入力署名が上で論じられたようなSIGHASH_ALLフラグを使用すべきである)、改変できるトランザクションの一部だけが、アンロッキングスクリプトフィールドである。出力におけるロッキングスクリプトを変えることはできず、それは、出力を譲渡する署名を入力が含んでいなければならないからである。出力のあらゆる変更は、これらの署名を無効にする。$PrevOuts_{change\text{-}in}$参照1118がチェンジアウトトランザクション1114の出力に現れる限り、参照1118を改変することはできず、それは参照1118が出力スクリプトに

位置するからである。トランザクションの入力スクリプトが何者かによって改変される場合であっても、PrevOutの参照は変わらず、ダストのチェーンを調査するときに正しいトランザクションに導く。

【０２３１】

図11は、第2および/または第3の態様におけるものと同様の、特徴の参照番号および名前を備える。たとえば、各追加トランザクション1104ba～dおよび最後のトランザクション1126は、先行するトランザクションへの後方消費参照1106a～fを備える。これは、先行するトランザクションへの後方消費参照706a～fも備える、図7Aに示されるような追加トランザクション704a～dおよび最後のトランザクション726と同様である。

【０２３２】

本態様において説明されるようなデータ構造1100を前方に調査する方法は、以下で概説される差を除き、第2の態様において図9を参照して説明される方法900と実質的に同じである。本態様では、ステップ922において抽出される参照は、チェンジイントランザクションのトランザクションidではなく$PrevOuts_{change\text{-}in}$参照1118である。チェンジイントランザクション1116は、この参照1118に基づいて見つけられる。チェンジイントランザクション1116は、トランザクションを探すことによって見つかるは、$PrevOuts_{change\text{-}in}$出力の少なくともサブセットを入力として備える。チェンジイントランザクション1116は次いで、トランザクションidを取得するためにハッシュされ、方法900は、チェンジイントランザクションidに基づいて、次の追加トランザクションを見つける(928)ことから平常通りに継続する。

【０２３３】

本態様は任意選択で、待機して追加のセキュリティのためにチェンジイントランザクション1116を確認する。本態様は任意選択で、第1の態様の第2の参照(608)の特徴および/または第3の後方参照(610、612a～c)の特徴を含む。

【０２３４】

データ項目をダストのチェーンに追加し、第2の態様において説明されるダストの新しいチェーンを条件的に作成するための方法は、異なる参照が使用されることを除き、本態様において実質的に同じように動作する。

【０２３５】

任意選択で、不変の前方参照1118が、態様2および3において説明されたような前方参照718、1018に加えて使用される。

【０２３６】

チェンジイントランザクションの入力の完全なセットがこの態様では使用されているが、PrevOutsのサブセットも使用され得ることが理解されるだろう。たとえば、チェンジイントランザクションがそのように2つの入力を備える場合、

$$PrevOuts_{change\text{-}in}=\{PrevOut_1:(TxID_{Prev,1},vout_1),PrevOut_2:(TxID_{prev,2},vout_2)\}$$

チェンジイントランザクションへの参照は、それらの2つのうちの1つだけに基づく。それでも、これはチェンジイントランザクションへの一意な参照であり、それは、このトランザクションがビットコインネットワークに公開されると、最初に見られるルールと、出力点がオンチェーンで一度しか消費され得ないという事実とが、この出力が決して代替のトランザクションによって消費され得ないことを確実にするからである。ここで「代替の」は、トランザクションの構造、すなわちその入力、出力、および値の交換を指し、トランザクションが展性を通じて非機能性の変更を受けるのを妨げることに留意されたい。

【０２３７】

そのような展性が発生するかどうかにかかわらず、トランザクションの最終形態は常に、この出力点をどのトランザクションが消費したかを確かめることによって一意に識別され得る。

【０２３８】

1つだけの参照を使用することは、一意な入力ベースの(またはPrevOutベースの)参照を作成するために単一の出力しか必要とされないという点で好ましいことがあり、それは

、参照の全体のサイズを大きく減らすからである。そしてこれは、オンチェーントランザクションに参照を含めるトランザクションフィーのコストを減らす。ビットコインにおけるprevOutベースの参照の最小サイズは、36バイト(32バイトのTxIDおよび4バイトのVout)程度に小さいことがあるので、参照のサイズも、参照されるトランザクションの入力の数とともに1のオーダーでスケーリングする。

【０２３９】

したがって、PrevOut参照は、チェンジイントランザクションへの少なくとも1つの入力のうちの少なくとも1つを備え得る。そして好ましくは、PrevOut参照は、チェンジイントランザクションへの少なくとも1つの入力の1つだけを備える。

【０２４０】

チェンジアウトへの後方参照

2方向の参照は、トランザクションのペアが他方との関係を示すことを可能にする。さらに、これは、2つの間のつながりが、開始点として見なされるトランザクションとは独立に特定されることを可能にし、すなわち、イベントまたはデータ項目のシーケンスが直線的である場合に、前方調査および後方調査を可能にする。

【０２４１】

しかしながら、実際には、そのような2方向の参照を達成することは、一般に各参照が一意性を示さなければならないという事実により難しい。一意性という性質を達成するのは、循環参照が原因で難しいことがある。ハッシュ関数の使用が、一意な識別子をデータに割り当てるための一般的な方法である。しかしながら、2つのブロックチェーントランザクション間で2方向のハッシュベースの参照を作成することは可能ではなく、それは、これが循環参照を作り出すからである。

【０２４２】

図12および図13を参照すると、第5の態様による、データ構造1200およびデータ構造を調査する方法1300が示されている。データ構造1200は、チェンジアウトトランザクション1214からチェンジイントランザクション1216への不変の前方参照1218と、チェンジイントランザクション1216からチェンジアウトトランザクション1214への後方参照1224とを備える。

【０２４３】

図12は、第2、第3、および/または第4の態様の特徴と同様の特徴に対して、同様の参照番号および名前を備える。たとえば、各追加トランザクション1204ba～dおよび最後のトランザクション1226は、先行するトランザクションへの後方消費参照1206a～fを備える。これは、先行するトランザクションへの後方消費参照706a～fを同様に備える、図7Aに示されるような追加トランザクション704a～dおよび最後のトランザクション726と同様である。不変の前方参照1218は、図11を参照して説明されるような不変の前方参照1118と同じように、または似たように構築され、機能する。

【０２４４】

後方参照1224は、チェンジアウトトランザクション1224のトランザクションidを備える。これは、後方参照1218がチェンジイントランザクション1216のトランザクションidにもはや依存せず、それにより、上で説明されたような循環ハッシュ参照の問題を回避するので可能である。

【０２４５】

後方参照1224を用いると、任意のチェンジアウト/チェンジイントランザクションにわたるダストのチェーンの後方への調査が可能である。図13に示される、調査のコンピュータで実施される方法1300は、ダストのチェーンの中の最後のトランザクション1226において開始する。方法は任意選択で、ループの最初のステップ1308において開始することによって、ダストのチェーンの任意の位置で開始することができることを、当業者は理解するだろう。

【０２４６】

後方調査の方法900のように、有利には、この調査の方法1300は、ブロックチェーン

上のトランザクションのフォーマット以外のトランザクションを調査するために隠れた知識を必要としない。調査において使用される参照はすべて、ブロックチェーン上のデータ項目とともに公開される。したがって、第三者は、私的なサービスまたはプロプライエタリサービスなしで、ダストのチェーンを調査し、ブロックチェーンに記憶されているデータ項目を取得し、記憶し、検証し、かつ/または別様に使用することも可能である。とりわけ、ブロックチェーンの性質ならびに現在のデータ書き込みシステムおよび方法により、ブロックチェーンに記憶されているデータに対して行われるあらゆる操作は、読み取り専用である。ブロックチェーンに記憶されているデータを改変することは現実的ではない。

【０２４７】

第1のステップ1302は最後のトランザクションを取得することであり、ここから方法はダストのチェーンを後方に調査する。

【０２４８】

任意選択で、ダストのチェーンにおける最初のトランザクションのトランザクションidは、最後のトランザクションに記憶されている。初期のトランザクションのトランザクションidは、後で参照するために記憶される。

【０２４９】

最後のトランザクションのトランザクションidは、最後のトランザクションをハッシュすることによって抽出される(1306)。前記トランザクションidは、現在のトランザクションidとして割り当てられる。方法は次いで、現在のトランザクションに対して作用するループに入る。

【０２５０】

ループの最初のステップは、現在のトランザクションidが初期トランザクションを指すかどうかを決定することである(1308)。任意選択で、これは、現在のトランザクションのトランザクションidを、第2のステップ1304において記憶された初期トランザクションidのそれと比較することによって行われる。代替として、この決定1308は、トランザクションが取得された(1310)後で行われる。決定1308は次いで、トランザクションに記憶されているデータおよび/またはメタデータに基づく。たとえば、図6Cを参照して第2の態様において説明されたように、トランザクションがメタデータの中にシードを備える場合、そのトランザクションが初期トランザクションであると決定することが可能であり得る。

【０２５１】

現在のトランザクションが初期トランザクションである場合、ループは終了する。任意選択で、供給されるトランザクションidに基づいて、ループは別のトランザクションで終了する。

【０２５２】

現在のトランザクションは、現在のトランザクションidに基づいて取得される(1310)(現在のトランザクションが初期トランザクションであったかどうかを決定することの一部として、トランザクションがすでに取得されていない限り)。

【０２５３】

現在のトランザクションが追加トランザクションである場合、第1の態様において説明されるようなペイロードの存在に基づいて決定される(1312)ように、動作1314は任意選択で、トランザクションのペイロードに対して行われる。

【０２５４】

好ましくは、動作は現在のトランザクションを検証することである。検証は、図6Aおよび図6Bを参照して説明されるような第3の後方参照610、612a～cに基づく。現在のペイロードのストリームダイジェストは、ループにおける次のペイロードのためのストリームダイジェストとともに、次のループ反復における検証のために記憶される。ループの中の前の反復で記憶されたストリームダイジェストがすでにある場合、現在のトランザクションのストリームダイジェストは、ストリームダイジェストが正しいことを検証するために前記記憶されている前のストリームダイジェストと比較される。

【０２５５】

追加または代替として、ペイロードのデータ項目は、後で使用するために記憶される。追加または代替として、調査者によって提供されるコールバックが呼び出されて、それにより、任意の動作が調査者によってペイロード上で行われることを可能にする。

【０２５６】

チェーンにおける次のトランザクションのトランザクションidは、現在のトランザクションから抽出される(1316)。チェーンにおける次のトランザクションのトランザクションidは、現在のトランザクションへの入力として記憶される。

【０２５７】

次のトランザクションidは現在のトランザクションidとして記憶され(1318)、ループはループの最初1318において再開する。

【０２５８】

トランザクションのタイプを決定する(1312)ことに戻ると、トランザクションがチェンジイントランザクションである場合、チェンジアウトトランザクションへの参照が取得される。好ましくは、参照は、参照が記憶されているチェンジイントランザクションから参照を抽出する(1320)ことによって取得される。本態様では、チェンジアウトトランザクションへの参照は、チェンジアウトトランザクションのトランザクションidである。

【０２５９】

チェンジアウトトランザクションは次いで、トランザクションidを使用して取得される(1322)。

【０２６０】

チェーンにおける次のトランザクションのトランザクションidは、現在のトランザクションから抽出される(1324)。チェーンにおける次のトランザクションのトランザクションidは、チェンジアウトトランザクションへの入力として記憶される。

【０２６１】

次のトランザクションidは現在のトランザクションidとして記憶され(1326)、ループはループの最初1318において再開する。

【０２６２】

データ項目をダストのチェーンに加え、本態様のダストの新しいチェーンを条件的に作成するための方法は、チェンジインとチェンジアウトとの間で異なる参照が使用されることを除き、第2の態様と実質的に同じ方法で動作する。

【０２６３】

本態様のデータ構造1200を前方に調査するための方法は、チェンジアウトからチェンジインへの同じ前方参照を備えるので、第4の態様と実質的に同じ方法で動作する。

【０２６４】

チェンジアウトトランザクションがチェンジイントランザクションのトランザクションidを備え、チェンジイントランザクションがチェンジアウトトランザクションの入力への参照を備えるように、参照が逆に使用されてもよいことを当業者は理解するだろう。

【０２６５】

トランザクションタイプ(追加であるかまたはチェンジインであるか)を決定するステップ1312は、チェンジアウトトランザクションではなくチェンジイントランザクションが決定されることを除き(後方調査はチェンジイン調査に最初に遭遇するので)、前方調査方法を参照して説明されるものと同じように、または似たように行われる。代替として、現在のトランザクションにダスト入力がないことは、現在のトランザクションがチェンジイントランザクションであることを示すために使用される。代替として、チェンジインは追加トランザクションより1少ない出力を有するので、入力の数がトランザクションのタイプを決定するために使用される。

【０２６６】

展性のない後方チェンジアウト参照

図14を参照すると、チェンジアウトトランザクション1414からチェンジイントランザクション1416への不変の前方参照1418と、チェンジイントランザクション1416からチ

ェンジアウトトランザクション1414への不変の後方参照1424とを作成するための第6の態様に従った、データ構造1400。

【０２６７】

図14は、第2、第3、第4、および/または第5の態様におけるものと同様の、特徴に対する参照番号および名前を備える。たとえば、各追加トランザクション1404ba～dおよび最後のトランザクション1426は、先行するトランザクションへの後方消費参照1406a～fを備える。これは、先行のトランザクションへの後方消費参照706a～fを同様に備える、図7Aに示されるような追加トランザクション704a～dおよび最後のトランザクション726と同様である。

【０２６８】

チェンジアウトトランザクション1414からチェンジイントランザクション1416への前方参照1418は、第5の態様において説明される前方参照1218と実質的に同じ方法で形成され動作する。

【０２６９】

チェンジイントランザクション1416からチェンジアウトトランザクション1414への後方参照1424は、$PrevOuts_{change\text{-}out}$セットに基づく。$PrevOuts_{change\text{-}out}$セットは、$PrevOuts_{change\text{-}in}$セットが第4の態様において説明されたように構築されるのと同じ方法で構築されるが、それは、チェンジアウトトランザクションによって消費されるトランザクション出力を使用する。第4の態様における参照と同様の方法を使用することで、特に不変性に関して同じ利益が得られる。$PrevOuts_{change\text{-}out}$セットは任意選択で、少なくとも1つの入力のうちの1つのみを備える。

【０２７０】

後方参照1424を追うこともほとんど同じ方法で行われる。$PrevOuts_{change\text{-}out}$セットに存在するトランザクションおよびインデックスを消費するトランザクションを見つける。このトランザクションがチェンジアウトトランザクションである。

【０２７１】

したがって、前方または後方に調査する方法は、後方に調査するときに、チェンジアウトトランザクションのトランザクションidが使用されないことを除き、第5の態様において説明されたのと実質的に同じ方法で動作し、$PrevOuts_{change\text{-}out}$参照は、上で説明されたように使用される。

【０２７２】

データ項目をダストのチェーンに追加し、第2の態様において説明されたようにダストの新しいチェーンを条件的に作成するための方法は、異なる参照が使用されることを除き、この態様と実質的に同じように動作する。

【０２７３】

ランデブートランザクション

第7の態様によれば、ダストのチェーンはさらにランデブートランザクションを備える。ランデブートランザクションのさらなる詳細は、英国特許出願第2020279.2号(2020年12月21日にnChain Holdings Limitedの名義で出願された)において説明され、参照によって本明細書に組み込まれる。

【０２７４】

ダストのチェーンは、ランデブートランザクションの存在がが祖先制限に関して考慮される必要があることを除き、(図8を参照して説明されたように)新しいデータ項目を追加することと実質的に同じように動作する。

【０２７５】

ダストのチェーンは、トランザクションタイプ確認ステップ912、1312における追加の確認が行われることを除き、(図9および図13を参照して説明されたように)ダストのチェーンを調査することに関して実質的に同じように動作する。ここで、現在のトランザクションがランデブートランザクションであると決定される場合、それは、追加トランザクションと同じダスト参照を備えるので、追加トランザクションに対して同様に反復される

。現在のトランザクションは、第1の態様を参照して説明されたように、複数のペイロードの存在に基づいてランデブートランザクションとして特定される。ランデブートランザクションは、異なるチェーンに関連する複数のダスト入力および出力を備え、正しいダスト出力が調査を継続するために選択される。ランデブートランザクション1802を備えるダストのチェーンのデータレイアウト1800が、図18において提供される。複数のペイロードがあるので、それらは「r」によってランデブートランザクションにおいてインデクシングされ、関連する出力は2rおよび2r+1において提供される。

【０２７６】

ダストのチェーンおよびランデブートランザクションを前方に調査することが見つかる場合、図9を参照して説明されたような方法900は、追加で以下のステップを備える。

【０２７７】

これらのステップは、現在のトランザクションがランデブートランザクションであると決定した(912)結果として発生する。ランデブートランザクションの関連する出力が、チェーンの調査を継続するために必要とされる。

【０２７８】

まず、インデックス「r」は、先行するトランザクションのダスト出力点を消費する入力を見つけることによって決定される。データを記憶する出力は次いで2r+1に位置し、ダスト出力のチェーンは2rに位置する。

【０２７９】

次に、ダスト出力のチェーンが見つかると、次のトランザクションが見つかる。次のトランザクションは、出力点($TxID_{rendezvous}$,2r)を消費するトランザクションを取得することによって見つかり、TxIDはランデブートランザクションのトランザクションidであり、ランデブートランザクションをハッシュすることによって取得される。

【０２８０】

この次のトランザクションは、方法900のループ910の最初において現在のトランザクションとして使用され、方法は継続する。

【０２８１】

任意選択で、2r+1に記憶されているデータは、図9の方法900の動作ステップ914を行うことに関して説明されたように、そのデータに対して行われる動作を有する。

【０２８２】

例として、図18のチェーンLが前方に調査されている場合、Lチェーンに関連する出力は、位置4および5に位置し、データは5に記憶され、ダストのチェーンは4に記憶され、すなわちr=2である。rは、L+2トランザクション(ランデブートランザクションに先行するトランザクション)のダスト出力(第0の出力)を見て、前記出力をどの入力が消費するか見つけることによって決定される。この場合、入力はランデブートランザクションのインデックス2にある。r=2であることが決定されると、ランデブートランザクションの関連する出力は、2r=4および2r+1=5にある。L+4トランザクションへの調査を続けるために、ランデブートランザクションがハッシュされ、出力($TxID_{rendezvous}$,4)を消費するトランザクションはL+4である。

【０２８３】

ダストのチェーンおよびランデブートランザクションを後方に調査することが見つかる場合、図13を参照して説明されたような方法1300は追加で、以下のステップを備える。これらの追加のステップは、逆であることを除いて前方調査のステップと同様である。

【０２８４】

$TxID_i=TxID_{i-1}$と割り当てる最後のステップに加えて、$k_i=k_{i-1}$を追加で割り当てるように、ダスト出力点のインデックス部分(ここでは「k」と呼ばれる)も各ループ反復(および初期ステップ)において記憶される。

【０２８５】

これらのステップは、現在のトランザクションがランデブートランザクションであると決定した(1312)結果として発生する。ランデブートランザクションの関連する入力が、チ

ェーンの調査を継続するために必要とされる。

【０２８６】

まず、「r」が$k_i/2$に設定される。「r」インデックスにより、ランデブートランザクションのインデックス「r」における入力が取得される。前記入力は$TxID_{i-1}$を備える。

【０２８７】

$TxID_i=TxID_{i-1}$を割り当てた後、ループは開始ステップ1308から継続する。

【０２８８】

任意選択で、2r+1(または代替として説明されるように、k_{i+1})において記憶されるデータは、図13の方法1300の動作ステップ1314を行うことに関して説明されたように、そのデータに対して行われる動作を有する。

【０２８９】

例として、やはり図18のチェーンLが逆に調査される場合、現在のトランザクションがランデブートランザクション1802であると決定する。以前の反復から、インデックスk_iはすでに知られており、(先行する反復の消費される出力点から)記憶される。k_iは4であり、2で割るとr=2が得られる。したがって、ランデブートランザクションの関連する入力は、インデックス2にある。インデックス2における入力によって消費される出力点のトランザクションidは、$TxID_{i-1}$である。調査を継続するために、トランザクションL+2は、トランザクションid $TxID_{i-1}$を使用して見つけられる。

【０２９０】

任意選択で、ランデブートランザクションに関連するあらゆるデータが取得され、調査者による使用のために記憶される。

【０２９１】

PrevOutトランザクションの展性

第8の態様によれば、PrevOut入力の展性を克服する方法が説明される。態様4および6において説明されるようなチェンジイントランザクションおよびチェンジアウトトランザクションに資金提供する(および、参照として使用され、この態様では「資金提供トランザクション」として参照される)トランザクションは、この説明全体で説明されるものと同じ展性の問題を抱える。参照として使用されるときと参照として確認されるときの間に、資金提供トランザクションが改変される場合、チェーンは、少なくとも1つの方向において不可逆に壊れることがあり、ダストのチェーンの調査を妨げる。

【０２９２】

さらに、これらの資金提供トランザクションがブロックチェーン上で確認されない場合、それらは未確認の祖先のカウントに寄与する。

【０２９３】

この展性の問題を克服するために、態様3を参照して説明されたものと同様の解決策が実施される。祖先制限に近づく前に(およびしたがって、チェンジイントランザクションおよびチェンジアウトトランザクションにおける資金提供トランザクションを使用することが必要になる前に)、資金提供トランザクションがブロックチェーンに出され、確認される。資金提供トランザクションをブロックチェーン上ですでに確認されている状態にすることによって、それらはイミュータブルになる。

【０２９４】

好ましくは、チェンジアウトトランザクションは、チェンジイントランザクションのための資金提供トランザクションが確認されるまで、ブロックチェーンに出されない。好ましくは、チェンジアウトトランザクションのための資金提供トランザクションが確認されるまで、チェンジイントランザクションがブロックチェーンに出されない。

【０２９５】

これには、資金提供トランザクションが未確認の祖先制限に寄与しないというさらなる利点がある。代替または追加として、この資金提供トランザクションを考慮するために、閾値の数は最大の未確認の祖先制限より2少ない。

【０２９６】

任意選択で、チェンジイン資金提供トランザクションが確認されてブロックチェーン上でイミュータブルとなるようにチェンジイン資金提供トランザクションを出すことは、図16を参照して説明されるようなサービスに関連する資金提供サービスによって行われる。資金提供サービスは任意選択で、ブロックチェーン上ですでに確認されているいくつかの資金提供トランザクションを維持し、データ書き込みサービスはそれらの資金提供トランザクションを使用する準備ができている。

【０２９７】

データ書き込みサービス

さらなる態様によれば、先行する態様のデータ構造および方法の任意の1つまたは複数が、上で説明されたような順序付けられた追加専用のデータストレージを少なくとも提供するための、以下で説明されるようなプラットフォームプロセッサとともに使用され得る。このさらなる態様は、BSVブロックチェーンなどのブロックチェーンネットワークを使用して、ソフトウェアで制御される技術システムまたはスマートコントラクトの管理などの、現実世界での有用なビジネスおよび技術への応用を迅速に行うことを有利に可能にする、Platform as a Service (Paas)およびSoftware as a Service (Saas)の提供であり得る。

【０２９８】

システムの高水準の概略図を示す、プラットフォームサービスの概要を図15において見ることができる。プラットフォームサービスは、API1508を提供するプラットフォームプロセッサ1500を有し、サービスはAPI1508を介して1つまたは複数のクライアントによってアクセスされ得る。

【０２９９】

この図に示されるようなプラットフォームサービス1500は、3つの群のサービスからなり、ユーザおよび組織が、クライアント側でブロックチェーンベースのソフトウェア、知識、またはライブラリを実際に実装することなく、ブロックチェーンの固有の性質によりもたらされる利点を簡単にかつセキュアに利用することを可能にすることを目的とする。

- チェーンの用途を商品データ台帳として簡略化することを目的とするデータサービス1502。好ましくは、データサービスは、ブロックチェーンへのデータの書き込みおよびブロックチェーンからのデータの読み取りを実施するために、本明細書において提供されるデータ構造および方法を使用する。
- ビットコインSVなどのデジタル資産によって裏付けられる一般化された計算ネットワークを提供することを目的とする計算サービス1504。
- ビットコインSVなどのデジタル資産を使用してトランザクションを行うための企業クラスの能力を提供するコマースサービス1506。

APIはウェブサービスとして実装されるので、APIにおいてクライアントからHTTPSプロトコルを介して、またはそれを使用して、要求が受信され得る。要求されるサービスは次いで、背後のソフトウェア1510を使用して1つまたは複数のサービスモジュールまたは処理リソース1502～1506によって実装され、そのような背後のソフトウェア1510は、ブロックチェーンに関連し、すなわち、ブロックチェーンに関連するトランザクションを作成し、処理し、出すための、リソース、ライブラリ、および/またはキー管理ウォレットの実装形態を実装するためのものである。処理されると、トランザクションは、(任意のそのような機能またはトランザクションライブラリを実装するクライアントの代わりに)ブロックチェーンネットワーク1512に出され得る。最大で、クライアントは、暗号通貨もしくは何らかの他のデジタル資産に関連するデジタルウォレットなどを実装することがあり、または実装することができるが、これは必須ではなく、それは、プラットフォームサービス1500は、クライアントのためのデジタル資産を提供して管理することも可能であり得るからである。

【０３００】

図16は、ブロックチェーンに関連する複数のサービスのより粗い概略図を提供し、これは、提供されるサービスの1つまたは複数にそれを介してアクセスすることができるAPIに

関連するプラットフォーム1600によって実装され得る。この図において見られるように、データサービス1602は、データ書き込みサービス1602aおよびデータ読み取りサービス1602bを含み得る。データ書き込みサービスおよびデータ読み取りサービスは、好ましくは、第6の態様において説明されるようなデータ構造を使用する。代替として、ここで説明される他の態様の任意の1つまたは複数が使用される。データ書き込みサービス1602aの例示的な使用法は、上で簡単に論じられたようなイベントストリームである。イベントストリームのさらなる詳細は、英国特許出願第2002285.1号(2020年2月19日にnChain Holdings Limitedの名義で出願された)の図4から図8を参照して論じられ、参照によって本明細書に組み込まれる。データ書き込みサービス1602aは、簡単で、セキュアで、最適化された方式で、クライアントがブロックチェーンへとデータを書き込むことを可能にする。データ読み取りサービス302bは、クライアントがクエリを送信することを可能にし、これはブロックチェーンに記憶されているデータを返す。これは、クライアントがアドホックにもしくは定期的に、すなわちある時間枠内にブロックチェーンから読み取ることを望むデータのタイプを、または、ブロックチェーン1610において処理される関係するもしくは関係しないイベントもしくは文書に関連するデータのタイプをあらかじめ定義できるような、フィルタリングされたストリームを使用することであり得る。データアーカイブ機能が、指定されたイベントまたは契約のための以前のトランザクションのログへのアクセスを可能にする。

【０３０１】

プラットフォーム1600の計算サービス1606は、スマートコントラクトに関連するアプリケーション1606aおよびフレームワーク1606bを含み、これらは、いくつかの実施形態では、ブロックチェーン1610においてステートマシンとして表され得る。計算サービス1606はデータサービス1602と相互作用し、それは、データが入力される必要があり、結果があらゆるそのような計算のためにクライアントに提供される必要があるからである。

【０３０２】

コマースサービス1604は、ベストインクラスのセキュリティの実践および技術に基づく、ブロックチェーン1610を調査するための企業ウォレット1604aを介した企業クラスの能力の提供を担う。たとえば、いくつかの実施形態では、企業ウォレットは、1より多くの人物またはユーザまたはアカウントが、定められた基準、すなわちあらかじめ定められた限界を超える大きい値の暗号通貨に関連する基準を満たすトランザクションを承認する必要があり得るとき、ブロックチェーントランザクション処理を可能にするための機能を実装し得る。企業ウォレットはまた、別のリソースを表す暗号通貨またはトークンなどの大量のデジタル資産を移動するための、署名の閾値の数および/または署名のタイプを実装するための機能を含み得る。次いで、これらの資産の移動は、そのような企業ウォレットの実装形態によって適用される基準に基づく処理に従って、ブロックチェーン上で表現され得る。

【０３０３】

SPVサービス1608(簡略化された支払検証)は、マイナーノードを実行しないのでブロックチェーンからの情報を必要とするがそれへの直接のリンクを含まない、適用例である。そのようなSPVサービス1608は、軽量のクライアントが、ブロックチェーン1610全体をダウンロードすることなく、トランザクションがブロックチェーンに含まれることを検証することを可能にする。

【０３０４】

データ書き込みデバイス

ここで図17を見ると、本開示の少なくとも一実施形態を実践するために使用され得るコンピューティングデバイス2600の例示的な簡略化されたブロック図が提供される。様々な実施形態において、コンピューティングデバイス2600は、上で示され説明されたシステムのいずれかを実装するために使用され得る。たとえば、コンピューティングデバイス2600は、図のDBMSの1つまたは複数の構成要素として使用されるように構成されてもよく、または、コンピューティングデバイス2600は、所与のユーザに関連するクライアン

トエンティティであるように構成されてもよく、クライアントエンティティは、図9のDBMSによって管理されるデータベースに対してデータベース要求を行う。したがって、コンピューティングデバイス2600は、ポータブルコンピューティングデバイス、パーソナルコンピュータ、または任意の電子コンピューティングデバイスであり得る。図17に示されるように、コンピューティングデバイス2600は、メインメモリ2608および永続的ストレージ2610を含むストレージサブシステム2606と通信するように構成され得る、1つまたは複数のレベルのキャッシュメモリおよびメモリコントローラ(集合的に2602とラベリングされる)を伴う1つまたは複数のプロセッサを含み得る。メインメモリ2608は、示されるようなダイナミックランダムアクセスメモリ(DRAM)2618および読み取り専用メモリ(ROM)2620を含み得る。ストレージサブシステム2606およびキャッシュメモリ2602は、本開示において説明されるようなトランザクションおよびブロックに関連する詳細などの、情報の記憶のために使用され得る。プロセッサ2602は、本開示において説明されるような任意の実施形態のステップまたは機能を提供するために利用され得る。

【０３０５】

プロセッサ2602はまた、1つまたは複数のユーザインターフェース入力デバイス2612、1つまたは複数のユーザインターフェース出力デバイス2614、およびネットワークインターフェースサブシステム2616と通信することができる。

【０３０６】

バスサブシステム2604は、コンピューティングデバイス2600の様々な構成要素およびサブシステムが意図されたように互いに通信することを可能にするための機構を提供し得る。バスサブシステム2604は単一のバスとして概略的に示されているが、バスサブシステムの代替の実施形態は複数のバスを利用してもよい。

【０３０７】

ネットワークインターフェースサブシステム2616は、他のコンピューティングデバイスおよびネットワークへのインターフェースを提供し得る。ネットワークインターフェースサブシステム2616は、コンピューティングデバイス2600からの他のシステムからデータを受信し、それにデータを送信するための、インターフェースとして働き得る。たとえば、ネットワークインターフェースサブシステム2616は、データ技術者がデータセンターなどの遠隔の位置にいる間にデータをデバイスに送信してデータをデバイスから受信することが可能になり得るように、データ技術者がデバイスをネットワークに接続することを可能にし得る。

【０３０８】

ユーザインターフェース入力デバイス2612は、キーボード、統合されたマウス、トラックボール、タッチパッド、またはグラフィクスタブレットなどのポインティングデバイス、スキャナ、バーコードスキャナ、ディスプレイに組み込まれたタッチスクリーン、音声認識システム、マイクロフォンなどのオーディオ入力デバイス、および他のタイプの入力デバイスなどの1つまたは複数のユーザ入力デバイスを含み得る。一般に、「入力デバイス」という用語の使用は、情報をコンピューティングデバイス2600に入力するためのすべてのあり得るタイプのデバイスおよび機構を含むことが意図されている。

【０３０９】

1つまたは複数のユーザインターフェース出力デバイス2614は、ディスプレイサブシステム、プリンタ、またはオーディオ出力デバイスなどの非視覚的ディスプレイなどを含み得る。ディスプレイサブシステムは、陰極線管(CRT)、液晶ディスプレイ(LCD)、発光ダイオード(LED)ディスプレイ、もしくはプロジェクションなどのフラットパネルデバイス、または他のディスプレイデバイスであり得る。一般に、「出力デバイス」という用語の使用は、コンピューティングデバイス2600から情報を出力するための任意の可能なタイプのデバイスおよび機構を含むことが意図される。1つまたは複数のユーザインターフェース出力デバイス2614は、たとえば、説明されるプロセスおよびその変形を実行するアプリケーションとのユーザ対話を、そのような対話が適切であり得るときに支援するための、ユーザインターフェースを提示するために使用され得る。

【０３１０】
ストレージサブシステム2606は、本開示の少なくとも1つの実施形態の機能を提供し得る基本的なプログラミングおよびデータ構築物を記憶するための、コンピュータ可読記憶媒体を提供し得る。アプリケーション(プログラム、コードモジュール、命令)は、1つまたは複数のプロセッサによって実行されると、本開示の1つまたは複数の実施形態の機能を提供してもよく、ストレージサブシステム2606に記憶されてもよい。これらのアプリケーションモジュールまたは命令は、1つまたは複数のプロセッサ2602によって実行され得る。ストレージサブシステム2606は、本開示に従って使用されるデータを記憶するためのリポジトリを追加で提供し得る。たとえば、メインメモリ2608およびキャッシュメモリ2602は、プログラムおよびデータのための揮発性ストレージを提供し得る。永続性ストレージ2610は、プログラムおよびデータのための永続的(不揮発性)ストレージを提供することができ、フラッシュメモリ、1つまたは複数のソリッドステートドライブ、1つまたは複数の磁気ハードディスクドライブ、関連するリムーバブルメディアを伴う1つまたは複数のフロッピーディスクドライブ、関連するリムーバブルメディアを伴う1つまたは複数の光学ドライブ(たとえば、CD-ROMまたはDVDまたはBlue-Ray)、および他の同様の記憶媒体を含み得る。そのようなプログラムおよびデータは、本開示において説明されるような1つまたは複数の実施形態のステップを実行するためのプログラム、ならびに本開示において説明されるようなトランザクションおよびブロックに関連するデータを含み得る。

【０３１１】
コンピューティングデバイス2600は、ポータブルコンピュータデバイス、タブレットコンピュータ、ワークステーション、または以下で説明される任意の他のデバイスを含む、様々なタイプであり得る。加えて、コンピューティングデバイス2600は、1つまたは複数のポート(たとえば、USB、ヘッドフォンジャック、Lightningコネクタなど)を通じてコンピューティングデバイス2600に接続され得る別のデバイスを含み得る。コンピューティングデバイス2600に接続され得るデバイスは、光ファイバコネクタを受け入れるように構成される複数のポートを含み得る。したがって、このデバイスは、処理のために、デバイスをコンピューティングデバイス2600に接続するポートを通じて送信され得る電気信号に、光信号を変換するように構成され得る。コンピュータおよびネットワークの変化し続ける性質により、図16に示されるコンピューティングデバイス2600の説明は、デバイスの好ましい実施形態を示すことが目的の具体的な例として意図されているにすぎない。図16に示されるシステムより多数または少数の構成要素を有する多くの他の構成が可能である。

【０３１２】
上で説明された様々な方法は、コンピュータプログラムによって実装され得る。コンピュータプログラムは、上で説明された様々な方法のうちの1つまたは複数の機能を実行するようにコンピュータに命令するようになされるコンピュータコードを含み得る。そのような方法を実行するためのコンピュータプログラムおよび/またはコードは、1つまたは複数のコンピュータ可読媒体、またはより一般的にはコンピュータプログラム製品で、コンピュータなどの装置に提供され得る。コンピュータ可読媒体は、一時的または非一時的であり得る。1つまたは複数のコンピュータ可読媒体は、データ送信のために、たとえばインターネットを介したコードのダウンロードのために、たとえば、電子的な、磁気的な、光学的な、電磁的な、赤外線の、もしくは半導体のシステム、または伝播媒体であり得る。代替として、1つまたは複数のコンピュータ可読媒体は、半導体またはソリッドステートメモリ、磁気テープ、リムーバブルコンピュータディスケット、ランダムアクセスメモリ(RAM)、読み取り専用メモリ(ROM)、固い磁気ディスク、および、CD-ROM、CD-R/W、またはDVDなどの光学ディスクなどの、1つまたは複数の物理的コンピュータ可読媒体の形式をとり得る。

【０３１３】
特に別様に述べられない限り、以下の議論から明らかなように、説明全体で、「決定す

る」、「提供する」、「算出する」、「計算する」、「特定する」、「組み合わせる」、「確立する」、「送信する」、「受信する」、「記憶する」、「推定する」、「確かめる」、「取得する」などの語を利用した議論は、コンピュータシステムのレジスタおよびメモリ内の物理(電気的な)量として表されるデータを、コンピュータシステムメモリまたはレジスタまたは他のそのような情報の記憶、送信、もしくは表示デバイス内の物理量として同様に表される他のデータへと操作して変換する、コンピュータシステムまたは同様の電子コンピューティングデバイスの活動および処理を指すことが理解されるだろう。

【０３１４】

ここで、本開示は、上記の態様に関連する以下の条項に基づいて論じられ、それらは、態様および実施形態のより良い説明、記述、および理解のために、例示的な実施形態として本明細書において提供される。

【０３１５】

1．ブロックチェーンに関連するコンピュータで実装されるデータ構造であって、

第1の出力と、

第1のデータ項目の表現と

を備える、第1のトランザクションと、

第1のデータ項目のさらなる表現と、

第2のデータ項目の表現と、

第1の出力に関連する第1の入力と、

第2の出力と

を備える、第2のトランザクションとを備える、コンピュータで実装されるデータ構造。

【０３１６】

2．第1のデータ項目がメタデータおよびシードである、条項1によるコンピュータで実装されるデータ構造。

【０３１７】

3．第1の項目の表現が、シードを備えるメタデータを備える、条項2によるコンピュータで実装されるデータ構造。

【０３１８】

4．第1の項目のさらなる表現がシードを備える、条項2または3によるコンピュータで実装されるデータ構造。

【０３１９】

5．第1のデータ項目の第1の表現がデータ項目のハッシュを備える、条項1によるコンピュータで実装されるデータ構造。

【０３２０】

6．第1のデータ項目の第1の表現がデータ項目を備える、条項1によるコンピュータで実装されるデータ構造。

【０３２１】

7．第1のトランザクションが、第1のデータ項目の第1の表現を備える原像を備える、条項5または6によるコンピュータで実装されるデータ構造。

【０３２２】

8．第1のデータ構造のさらなる表現が、第1のトランザクションの原像のハッシュを備える、条項7によるコンピュータで実装されるデータ構造。

【０３２３】

9．第2のトランザクションがさらに、第1のトランザクションへの参照を備える、条項1から8のいずれか1つまたは複数によるコンピュータで実装されるデータ構造。

【０３２４】

10．第2のタイプのトランザクションへの第1の参照を備える第1のタイプのトランザクションと、

第2のタイプのトランザクションと

をさらに備える、条項1から9のいずれか1つまたは複数によるコンピュータで実装される

データ構造。
【０３２５】
11．第1の参照が第1のタイプのトランザクションの出力に記憶される、条項10によるコンピュータで実装されるデータ構造。
【０３２６】
12．第2のタイプのトランザクションへの第1の参照を備える出力を備える第1のタイプのトランザクションと、
第2のタイプのトランザクションと
を備える、ブロックチェーンに関連するコンピュータで実装されるデータ構造。
【０３２７】
13．第1の参照が不変の参照である、条項10から12のいずれか1つまたは複数によるコンピュータで実装されるデータ構造。
【０３２８】
14．第1の参照が第2のタイプのトランザクションの不変の特徴に基づく、条項13によるコンピュータで実装されるデータ構造。
【０３２９】
15．第1の参照が第2のタイプのトランザクションのトランザクションidを備える、条項10から14のいずれか1つまたは複数によるコンピュータで実装されるデータ構造。
【０３３０】
16．第2のタイプのトランザクションが少なくとも1つの入力を備え、第1の参照が第2のタイプのトランザクションへの少なくとも1つの入力の少なくとも1つに基づく、条項10から15のいずれか1つまたは複数によるコンピュータで実装されるデータ構造。
【０３３１】
17．第2のタイプのトランザクションが第2の参照を備える、条項10から16のいずれか1つまたは複数によるコンピュータで実装されるデータ構造。
【０３３２】
18．第2の参照が第2のタイプのトランザクションの出力に記憶される、条項17によるコンピュータで実装されるデータ構造。
【０３３３】
19．第2の参照が第1のタイプのトランザクションのトランザクションidを備える、条項17または18によるコンピュータで実装される。
【０３３４】
20．第2の参照が不変の参照である、条項17または18によるコンピュータで実装されるデータ構造。
【０３３５】
21．第2の参照が第1のタイプのトランザクションの不変の特徴に基づく、条項20によるコンピュータで実装されるデータ構造。
【０３３６】
22．第1のタイプのトランザクションが少なくとも1つの入力を備え、第2の参照が第1のタイプのトランザクションの少なくとも1つの入力の少なくとも1つに基づく、条項17、18、20、または21のいずれか1つまたは複数によるコンピュータで実装されるデータ構造。
【０３３７】
23．少なくとも1つの入力を備える参照が、トランザクション出力点の形式をとる、条項16または22によるコンピュータで実装されるデータ構造。
【０３３８】
24．ブロックチェーンに関連するコンピュータで実装されるデータ構造であって、
第2のタイプのトランザクションへの第1の参照と、
少なくとも1つの入力と
を備える第1のタイプのトランザクションと、

第1のタイプのトランザクションへの第2の参照と、
少なくとも1つの入力と
を備える第2のタイプのトランザクションとを備え、
第1の参照が第2のタイプのトランザクションへの少なくとも1つの入力の少なくとも1つに基づき、第2の参照が第1のタイプのトランザクションへの少なくとも1つの入力の少なくとも1つに基づく、コンピュータで実装されるデータ構造。

【０３３９】

25．ブロックチェーンに関連するコンピュータで実装されるデータ構造であって、
第2のタイプのトランザクションへの第1の参照と、
少なくとも1つの入力と
を備える、第1のタイプのトランザクションと、
第1のタイプのトランザクションへの第2の参照と、
少なくとも1つの入力と
を備える第2のタイプのトランザクションとを備え、
第1の参照が第2のタイプのトランザクションのトランザクションidを備え、第2の参照が第1のタイプのトランザクションへの少なくとも1つの入力の少なくとも1つに基づく、コンピュータで実装されるデータ構造。

【０３４０】

26．ブロックチェーンに関連するコンピュータで実装されるデータ構造であって、
第2のタイプのトランザクションへの第1の参照と、
少なくとも1つの入力と
を備える、第1のタイプのトランザクションと、
第1のタイプのトランザクションへの第2の参照と、
少なくとも1つの入力と
を備える第2のタイプのトランザクションとを備え、
第1の参照が第2のタイプのトランザクションへの少なくとも1つの入力の少なくとも1つに基づき、第2の参照が第1のタイプのトランザクションのトランザクションidを備える、コンピュータで実装されるデータ構造。

【０３４１】

27．ブロックチェーンシステムの中のトランザクションのセットに関連するコンピュータで実施される方法であって、
要求を受信するステップであって、要求が、データ項目の表現がブロックチェーンに記憶されることの誘因となる、ステップと、
トランザクションのセットの中の最新のトランザクションを取得するステップと、
最新のトランザクションからの出力に関連する入力と、
出力と、
ブロックチェーンに記憶されるべきデータ項目の表現と、
最新のトランザクションへの参照と
を備える、新しいブロックチェーントランザクションを作成するステップと、
トランザクションをブロックチェーンに出すステップとを備える、コンピュータで実施される方法。

【０３４２】

28．最新のトランザクションへの参照が最新のトランザクションの不変の特徴のハッシュである、条項27によるコンピュータで実施される方法。

【０３４３】

29．最新のトランザクションが原像を備え、最新のトランザクションへの参照が最新のトランザクションの原像のハッシュである、条項28によるコンピュータで実施される方法。

【０３４４】

30．新しいブロックチェーントランザクションがさらに、トランザクションのセットの

中の初期トランザクションへの参照を備える、条項27から29のいずれか1つまたは複数によるコンピュータで実施される方法。

【０３４５】

31．トランザクションのセットの中の初期トランザクションへの参照が初期トランザクションに基づく、条項30によるコンピュータで実施される方法。

【０３４６】

32．初期トランザクションへの参照が初期トランザクションのハッシュである、条項30または31によるコンピュータで実施される方法。

【０３４７】

33．第2のタイプのトランザクションを作成するステップと、

トランザクションのセットの中の最新のトランザクションからのトランザクション出力に関連する入力を備える第1のタイプのトランザクションを作成するステップと、

第2のタイプのトランザクションをブロックチェーンに出すステップと、

第1のタイプのトランザクションをブロックチェーンに出すステップとをさらに備える、条項27から32のいずれか1つまたは複数によるコンピュータで実施される方法。

【０３４８】

34．ブロックチェーンシステムの中のトランザクションのセットに関連するコンピュータで実施される方法であって、

第2のタイプのトランザクションを作成するステップと、

トランザクションのセットの中の最新のトランザクションからの出力に関連する少なくとも1つの入力を備える第1のタイプのトランザクションを作成するステップと、

第2のタイプのトランザクションをブロックチェーンに出すステップと、

第1のタイプのトランザクションをブロックチェーンに出すステップとを備える、コンピュータで実施される方法。

【０３４９】

35．トランザクションのセットのサブセットの中のトランザクションの総数を決定するステップと、

トランザクションのサブセットの中のトランザクションの総数が閾値以上であるかどうかを決定するステップとをさらに備える、条項33または34によるコンピュータで実施される方法。

【０３５０】

36．トランザクションのサブセットのメンバーシップが、トランザクションがブロックチェーン上で確認されたかどうかによって定義される、条項35によるコンピュータで実施される方法。

【０３５１】

37．トランザクションのサブセットのメンバーシップが、トランザクションのセットの中の任意のトランザクションとの消費関係によって定義される、条項35または36によるコンピュータで実施される方法。

【０３５２】

38．トランザクションのサブセットのメンバーシップが追加で閾値によって定義される、条項35または36によるコンピュータで実施される方法。

【０３５３】

39．トランザクションのサブセットがトランザクションの第1のチェーンを備える、条項35から38のいずれか1つまたは複数によるコンピュータで実施される方法。

【０３５４】

40．トランザクションのサブセットがトランザクションの第1のチェーンである、条項39によるコンピュータで実施される方法。

【０３５５】

41．トランザクションの第1のチェーンが、サブセットの中の最初のトランザクションを除く各トランザクションがチェーンの中の前のトランザクションへの参照を備えるよう

に構築される、条項39または40によるコンピュータで実施される方法。
【０３５６】
42．前のトランザクションへの参照が、前のトランザクションからのトランザクション出力に関連する入力である、条項41によるコンピュータで実施される方法。
【０３５７】
43．トランザクションのセットがトランザクションの複数のサブセットを備える、条項40から42のいずれか1つまたは複数によるコンピュータで実施される方法。
【０３５８】
44．トランザクションのセットがトランザクションのさらなるチェーンを備える、条項43によるコンピュータで実施される方法。
【０３５９】
45．閾値が祖先制限に基づく、条項35から44のいずれか1つまたは複数によるコンピュータで実施される方法。
【０３６０】
46．閾値が祖先制限より1少ない、条項45によるコンピュータで実施される方法。
【０３６１】
47．第2のタイプのトランザクションおよび第1のタイプのトランザクションを作成して出すステップが、トランザクションのサブセットの中のトランザクションの総数と閾値との比較に基づいて行われる、条項35から46のいずれか1つまたは複数によるコンピュータで実施される方法。
【０３６２】
48．第2のタイプのトランザクションおよび第1のタイプのトランザクションを作成して出すステップが、トランザクションのサブセットの中のトランザクションの総数が閾値以上であるかどうかに基づいて行われる、条項47によるコンピュータで実施される方法。
【０３６３】
49．第1のタイプのトランザクションが第2のタイプのトランザクションへの第1の参照を備える、条項33から48のいずれか1つまたは複数によるコンピュータで実施される方法。
【０３６４】
50．第1の参照が不変の参照である、条項49によるコンピュータで実施される方法。
【０３６５】
51．第1の参照が第2のタイプのトランザクションの不変の特徴に基づく、条項50によるコンピュータで実施される方法。
【０３６６】
52．第1の参照が第2のタイプのトランザクションのトランザクションidを備える、条項49から51のいずれか1つまたは複数によるコンピュータで実施される方法。
【０３６７】
53．第2のタイプのトランザクションが、第1のタイプのトランザクションを出す前にブロックチェーン上で確認される、条項33から52のいずれか1つまたは複数によるコンピュータで実施される方法。
【０３６８】
54．第2のタイプのトランザクションが少なくとも1つの入力を備え、第1の参照が、第2のタイプのトランザクションの少なくとも1つの入力の少なくとも1つに基づく、条項49から53のいずれか1つまたは複数によるコンピュータで実施される方法。
【０３６９】
55．第2のタイプのトランザクションがトランザクションのセットの中のトランザクションへの第2の参照を備える、条項33から54のいずれか1つまたは複数によるコンピュータで実施される方法。
【０３７０】
56．第2の参照がトランザクションのセットの中の初期トランザクションへの参照であ

る、条項55によるコンピュータで実施される方法。
【０３７１】
57．初期トランザクションへの参照が初期トランザクションのトランザクションidを備える、条項56によるコンピュータで実施される方法。
【０３７２】
58．第2の参照が不変の参照である、条項55から57のいずれか1つまたは複数によるコンピュータで実施される方法。
【０３７３】
59．第2の参照が第1のタイプのトランザクションの不変の特徴に基づく、条項58によるコンピュータで実施される方法。
【０３７４】
60．第2の参照が第1のタイプのトランザクションのトランザクションidを備える、条項55から59のいずれか1つまたは複数によるコンピュータで実施される方法。
【０３７５】
61．第2の参照が第1のタイプのトランザクションの少なくとも1つの入力の少なくとも1つに基づく、条項58または59によるコンピュータで実施される方法。
【０３７６】
62．少なくとも1つの入力の少なくとも1つに基づく参照が、トランザクション出力点の形態をとる、条項54または61によるコンピュータで実施される方法。
【０３７７】
63．ブロックチェーントランザクションのセットを前方に調査するためのコンピュータで実施される方法であって、
(a)トランザクションのセットにおける現在のトランザクションを取得するステップと、
(b)現在のトランザクションが第1のタイプのトランザクションであると決定し、この決定に基づいて、
i．第1のタイプのトランザクションに基づいて第2のタイプのトランザクションへの参照を取得するステップ、
ii．第2のタイプのトランザクションへの参照に基づいて第2のタイプのトランザクションを取得するステップ、および
iii．第2のタイプのトランザクションを現在のトランザクションとしてステップ(c)に続くステップ
を行うステップと、
(c)現在のトランザクション識別子を取得するステップと、
(d)現在のトランザクション識別子を参照するさらなるトランザクションを取得するステップと、
(e)そのさらなるトランザクションを現在のトランザクションとして開始してステップ(b)、(c)、(d)、および(e)を行い、ループを作成するステップとを備える、コンピュータで実施される方法。
【０３７８】
64．第2のタイプのトランザクションへの参照が第1のタイプのトランザクションに記憶され、参照が第1のタイプのトランザクションから参照を抽出することによって取得される、条項63によるコンピュータで実施される方法。
【０３７９】
65．第2のタイプのトランザクションの参照が第2のタイプのトランザクションのトランザクションidを備える場合、第2のタイプのトランザクションを取得するステップが、
第2のタイプのトランザクションのトランザクションidと同じトランザクションidを伴う、ブロックチェーンの中のまたはブロックチェーンネットワークノード内のトランザクションを見つけるステップを備える、条項63または64によるコンピュータで実施される方法。
【０３８０】

66．第2のタイプのトランザクションへの参照が第2のタイプのトランザクションへの少なくとも1つの入力のセットを備える場合、第2のタイプのトランザクションを取得するステップが、
第2のタイプのトランザクションへの参照に含まれるような入力セットの少なくとも1つの入力を伴う、ブロックチェーンの中のまたはブロックチェーンネットワークノード内のトランザクションを見つけるステップを備える、条項63または64によるコンピュータで実施される方法。
【０３８１】
67．現在のトランザクションが第2のタイプのトランザクションではないと決定し、この決定に基づいて、現在のトランザクションに関連するデータペイロードに対して動作を行い、ステップ(c)に続くステップをさらに備える、条項63から66のいずれか1つまたは複数によるコンピュータで実施される方法。
【０３８２】
68．現在のトランザクションが、現在のトランザクションの内容に基づいて、第1のタイプのトランザクションではないと決定される、条項63から67のいずれか1つまたは複数によるコンピュータで実施される方法。
【０３８３】
69．現在のトランザクションが、現在のトランザクションの出力のサイズに基づいて、第1のタイプのトランザクションではないと決定される、条項68によるコンピュータで実施される方法。
【０３８４】
70．データペイロードに対する動作を行うステップが、
先行するトランザクションのデータペイロードに基づいてハッシュを記憶するステップと、
現在のトランザクションから、先行するトランザクションのデータペイロードへの参照を抽出するステップと、
先行するトランザクションのデータペイロードに基づくハッシュおよび先行するトランザクションのデータペイロードへの参照が有効であることを検証するステップとを備える、条項67によるコンピュータで実施される方法。
【０３８５】
71．先行するトランザクションへの参照が、先行するトランザクションのペイロードのさらなるハッシュである、条項70によるコンピュータで実施される方法。
【０３８６】
72．検証するステップが、先行するトランザクションのデータペイロードのハッシュおよび先行するトランザクションのデータペイロードへの参照が同じであるかどうかを決定するステップを備える、条項70または71によるコンピュータで実施される方法。
【０３８７】
73．ステップ(a)の前に実行されるべき、
トランザクションのセットの中の初期トランザクションを取得するステップと、
初期トランザクションがシード値を備えることを検証するステップと、
初期トランザクション識別子を取得するために初期トランザクションをハッシュするステップと、
第1のトランザクション識別子を参照する第2のトランザクションを取得するステップと、
第2のトランザクションがシード値を備え、それが第1のトランザクションのシード値と同じであることを検証するステップと、
第2のトランザクション識別子を取得するために第2のトランザクションをハッシュするステップと、
第2のトランザクション識別子を参照する第3のトランザクションを取得するステップと、

第3のトランザクションを現在のトランザクションとしてステップ(b)に移るステップとをさらに備える、条項63から72のいずれか1つまたは複数によるコンピュータで実施される方法。

【０３８８】

74．ステップ(a)の後で実行されるべき、

現在のトランザクションが最後のトランザクションであるかどうかに基づいて調査を終了するステップをさらに備え、現在のトランザクションが最後のトランザクションであるかどうかを決定するステップが、現在のトランザクションに含まれるデータに基づく、条項63から73のいずれか1つまたは複数によるコンピュータで実施される方法。

【０３８９】

75．ステップ(a)の後で実行されるべき、

現在のトランザクションが第3のタイプのトランザクションであると決定し、この決定に基づいて、

i．先行するトランザクションへの参照を備える入力に対するインデックスを取得するステップ、

ii．先行するトランザクションへの参照を備える入力に関連する出力を取得するステップ、

iii．取得された出力に基づいて次のトランザクションを取得するステップ、および

iv．次のトランザクションを現在のトランザクションとしてステップ(c)に続くステップ

を行うステップをさらに備える、条項63から74のいずれか1つまたは複数によるコンピュータで実施される方法。

【０３９０】

76．ブロックチェーントランザクションのセットを後方に調査するためのコンピュータで実施される方法であって、

(a)トランザクションのセットにおける現在のトランザクションを取得するステップと、

(b)現在のトランザクションが第2のタイプのトランザクションであると決定し、この決定に基づいて、

i．第2のタイプのトランザクションに基づいて第1のタイプのトランザクションへの参照を取得するステップ、

ii．第1のタイプのトランザクションへの参照に基づいて第1のタイプのトランザクションを取得するステップ、および

iii．第1のタイプのトランザクションを現在のトランザクションとしてステップ(c)に続くステップ

を行うステップと、

(c)現在のトランザクションから先行するトランザクションのトランザクション識別子を取得するステップと、

(d)先行するトランザクションのトランザクション識別子に基づいて先行するトランザクションを取得するステップと、

(e)その先行するトランザクションを現在のトランザクションとして開始してステップ(b)、(c)、(d)、および(e)を行い、ループを作成するステップとを備える、コンピュータで実施される方法。

【０３９１】

77．第1のタイプのトランザクションへの参照が第2のタイプのトランザクションに記憶され、第2のタイプのトランザクションから参照を抽出することによって参照が取得される、条項76によるコンピュータで実施される方法。

【０３９２】

78．第1のタイプのトランザクションへの参照が第1のタイプのトランザクションのトランザクションidを備える場合、第1のタイプのトランザクションを取得するステップが、

第1のタイプのトランザクションのトランザクションidと同じトランザクションidを伴

う、ブロックチェーンの中のまたはブロックチェーンノード内のトランザクションを見つけるステップを備える、条項76または77によるコンピュータで実施される方法。

【０３９３】

79．第1のタイプのトランザクションへの参照が第1のタイプのトランザクションへの少なくとも1つの入力のセットを備える場合、第1のタイプのトランザクションを取得するステップが、

第1のタイプのトランザクションへの参照に含まれるような入力セットの少なくとも1つの入力を伴う、ブロックチェーンの中のまたはブロックチェーンノード内のトランザクションを見つけるステップを備える、条項76または77によるコンピュータで実施される方法。

【０３９４】

80．現在のトランザクションが第1のタイプのトランザクションではないと決定し、この決定に基づいて、現在のトランザクションに関連するデータペイロードに対して動作を行い、ステップ(c)に続くステップをさらに備える、条項76から79のいずれか1つまたは複数によるコンピュータで実施される方法。

【０３９５】

81．現在のトランザクションが、現在のトランザクションの内容に基づいて、第2のタイプのトランザクションではないと決定される、条項76から80のいずれか1つまたは複数によるコンピュータで実施される方法。

【０３９６】

82．現在のトランザクションが原像を備えるデータペイロードを備えるかどうかに基づいて、現在のトランザクションがチェンジイントランザクションではないと決定される、条項81によるコンピュータで実施される方法。

【０３９７】

83．データペイロードに対して動作を行うステップが、

先行するトランザクションのデータペイロードに基づいてハッシュを記憶するステップと、

現在のトランザクションから、先行するトランザクションのデータペイロードへの参照を抽出するステップと、

先行するトランザクションのデータペイロードに基づくハッシュおよび先行するトランザクションのデータペイロードへの参照が有効であることを検証するステップとを備える、条項76から82のいずれか1つによるコンピュータで実施される方法。

【０３９８】

84．先行するトランザクションへの参照が、先行するトランザクションのペイロードのさらなるハッシュである、条項83によるコンピュータで実施される方法。

【０３９９】

85．検証するステップが、先行するトランザクションのデータペイロードのハッシュおよび先行するトランザクションのデータペイロードへの参照が同じであるかどうかを決定するステップを備える、条項83または84によるコンピュータで実施される方法。

【０４００】

86．ステップ(a)の前に実行されるべき、

トランザクションのセットにおける最後のトランザクションを取得するステップと、

最後のトランザクションからトランザクションのセットにおける最初のトランザクションのトランザクションidを取得するステップと、

最後のトランザクション識別子を取得するために最後のトランザクションをハッシュするステップと、

最後のトランザクション識別子を参照する第2のトランザクションを取得するステップと、

第2のトランザクションを現在のトランザクションとしてステップ(b)に移るステップとをさらに備える、条項76から85のいずれか1つまたは複数によるコンピュータで実施され

る方法。
【０４０１】
87．ステップ(a)の後で実行されるべき、
現在のトランザクションのトランザクションidが初期トランザクションidに等しいかどうかに基づいて調査を終了するステップをさらに備える、条項86によるコンピュータで実施される方法。
【０４０２】
88．ステップ(a)の後で実行されるべき、
現在のトランザクションが初期/その初期トランザクションであるかどうかに基づいて調査を終了するステップをさらに備え、現在のトランザクションが初期トランザクションであるかどうかを決定するステップが、現在のトランザクションのデータペイロードに含まれるデータに基づく、条項76から87のいずれか1つまたは複数によるコンピュータで実施される方法。
【０４０３】
89．ステップ(a)の後で実行されるべき、
現在のトランザクションが第3のタイプのトランザクションであると決定し、この決定に基づいて、
i．現在のトランザクションの入力に対するインデックスを取得するステップ、
ii．現在のトランザクションの取得される入力に基づく先行するトランザクションへの参照を取得するステップ、
iii．参照に基づいて先行するトランザクションを取得するステップ、および
iv．先行するトランザクションを現在のトランザクションとしてステップ(c)に続くステップ
を行うステップをさらに備える、条項76から88のいずれか1つまたは複数によるコンピュータで実施される方法。
【０４０４】
90．ブロックチェーントランザクションのセットを前方に調査するためのコンピュータで実施される方法であって、
(a)トランザクションのセットにおける現在のトランザクションを取得するステップと、
(b)現在のトランザクションが第3のタイプのトランザクションであると決定し、この決定に基づいて、
i．先行するトランザクションへの参照を備える入力に対するインデックスを取得するステップと、
ii．先行するトランザクションへの参照を備える入力に関連する出力を取得するステップと、
iii．取得される出力に基づいて次のトランザクションを取得するステップと、
iv．次のトランザクションを現在のトランザクションとしてステップ(c)に続くステップと
を行うステップと、
(c)現在のトランザクション識別子を取得するステップと、
(d)現在のトランザクション識別子を参照するさらなるトランザクションを取得するステップと、
(e)そのさらなるトランザクションを現在のトランザクションとして開始してステップ(b)、(c)、(d)、および(e)を行い、ループを作成するステップとを備える、コンピュータで実施される方法。
【０４０５】
91．ブロックチェーントランザクションのセットを後方に調査するためのコンピュータで実施される方法であって、
(a)トランザクションのセットにおける現在のトランザクションを取得するステップと、
(b)現在のトランザクションが第3のタイプのトランザクションであると決定し、この決

定に基づいて、
i. 現在のトランザクションの入力に対するインデックスを取得するステップと、
ii. 現在のトランザクションの取得される入力に基づいて先行するトランザクションへの参照を取得するステップと、
iii. 参照に基づいて先行するトランザクションを取得するステップと、
iv. 先行するトランザクションを現在のトランザクションとしてステップ(c)に続くステップと
を行うステップと、
(c)現在のトランザクションから先行するトランザクションのトランザクション識別子を取得するステップと、
(d)先行するトランザクションのトランザクション識別子に基づいて先行するトランザクションを取得するステップと、
(e)その先行するトランザクションを現在のトランザクションとして開始してステップ(b)、(c)、(d)、および(e)を行い、ループを作成するステップとを備える、コンピュータで実施される方法。

【０４０６】

92. プロセッサおよびメモリを備えるコンピューティングデバイスであって、メモリが、プロセッサによる実行の結果として、デバイスに条項27から62のいずれか1つまたは複数に記載されるコンピュータで実施される方法を実行させる実行可能命令を含む、コンピューティングデバイス。

【０４０７】

93. 条項92によるデータ書き込みデバイスと、
データを備える要求をデータ書き込みデバイスに出すように構成されるコンピューティングデバイスとを備える、コンピュータシステム。

【０４０８】

94. コンピュータのプロセッサによって実行される結果として、コンピュータに条項27から62のいずれか1つまたは複数の方法を実行させる実行可能命令が記憶されている、コンピュータ可読記憶媒体。

【０４０９】

95. プロセッサおよびメモリを備えるコンピューティングデバイスであって、メモリが、プロセッサによる実行の結果として、デバイスに条項63から75のいずれか1つまたは複数に記載されるコンピュータで実施される方法を実行させる実行可能命令を含む、コンピューティングデバイス。

【０４１０】

96. コンピュータのプロセッサによって実行される結果として、コンピュータに条項63から75のいずれか1つまたは複数の方法を実行させる実行可能命令が記憶されている、コンピュータ可読記憶媒体。

【０４１１】

97. プロセッサおよびメモリを備えるコンピューティングデバイスであって、メモリが、プロセッサによる実行の結果として、デバイスに条項76から91のいずれか1つまたは複数に記載されるコンピュータで実施される方法を実行させる実行可能命令を含む、コンピューティングデバイス。

【０４１２】

98. コンピュータのプロセッサによって実行される結果として、コンピュータに条項76から91のいずれか1つまたは複数の方法を実行させる実行可能命令が記憶されている、コンピュータ可読記憶媒体。

【０４１３】

本明細書および特許請求の範囲において使用される「備える(comprising)」という用語は、「少なくとも一部～からなる」を意味する。「備える」という用語を含む本明細書および特許請求の範囲における各々の陳述を解釈するとき、その用語の後にあるもの以外の

特徴も存在していてもよい。「備える(comprise)」および「備える(comprises)」などの関連する用語も、同じように解釈されるべきである。

【０４１４】

本明細書において使用される場合、「および/または」という用語は、「および」もしくは「または」または両方を意味する。本明細書において使用される場合、名詞の後の「(s)」は、名詞の複数形および/または単数形を意味する。単数形での要素の言及は、複数形でそのような要素に言及することを排除せず、その逆も当てはまる。

【０４１５】

上記の説明は、限定的ではなく例示的であることが意図されていることを理解されたい。上記の説明を読んで理解すれば、多くの他の実装形態が当業者に明らかになるだろう。本開示は、特定の例示的な実装形態を参照して説明されたが、本開示は、説明された実装形態に限定されず、添付の特許請求の範囲内の修正および変更とともに実践され得ることが認識されるだろう。したがって、明細書および図面は、限定するものではなく例示するものであると見なされるべきである。したがって、本開示の範囲は、添付の特許請求の範囲に関連して、そのような請求項が権利を与えられる均等物の完全な範囲とともに決定されるべきである。

【符号の説明】

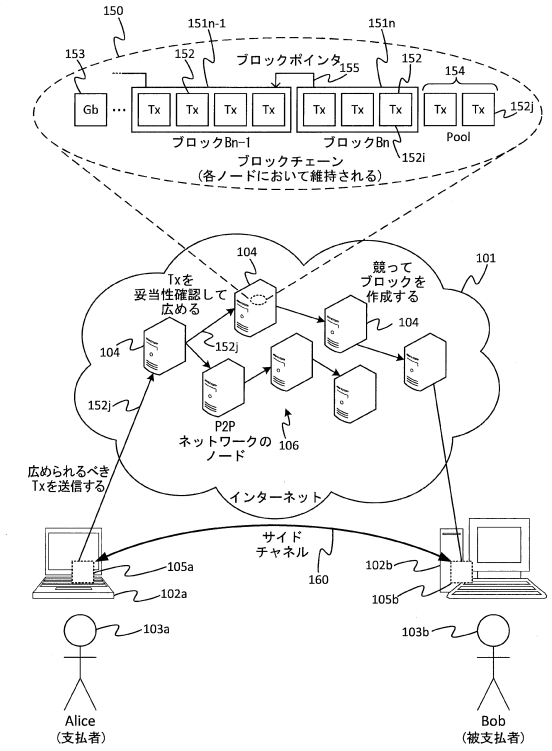
【０４１６】

101 ブロックチェーンネットワーク
103 ユーザ
104 ブロックチェーンノード
105 クライアントアプリケーション
150 ブロックチェーン
151 ブロック
152 トランザクション
153 ジェネシスブロック
160 サイドチャンネル
201 ヘッダ
202 入力フィールド
203 出力フィールド
301 トランザクションエンジン
302 UI層
350 UI
450 ノードソフトウェア
451 プロトコルエンジン
452 スクリプトエンジン
453 スタック
454 アプリケーションレベル決定エンジン
455 ブロックチェーン関連機能モジュール
502 イベント
504 ブロックチェーントランザクション
1500 プラットフォームサービス
1502 データサービス
1504 計算サービス
1506 コマースサービス
1508 プラットフォームAPI
1510 ブロックチェーンノードソフトウェア
1512 ブロックチェーンネットワーク
1600 プラットフォーム
1602 データサービス

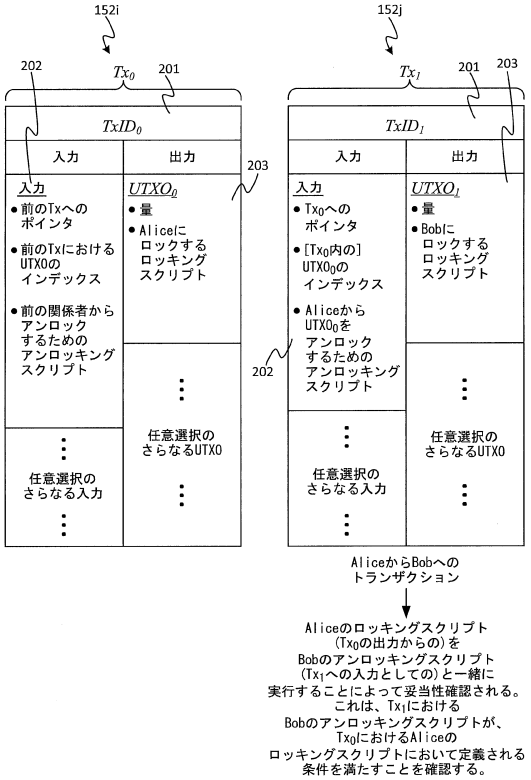
1604 コマースサービス
1606 計算サービス
1608 SPVサービス
1610 ブロックチェーンネットワーク
2602 プロセッサ
2604 バスサブシステム
2606 ストレージサブシステム
2608 メモリサブシステム
2610 ファイルストレージサブシステム
2612 ユーザインターフェース入力デバイス
2614 ユーザインターフェース出力デバイス
2616 ネットワークインターフェース
2618 RAM
2620 ROM
2624 クロック

【図面】

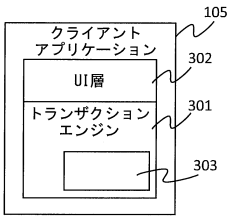
【図１】



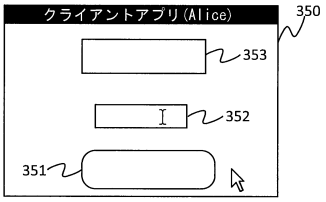
【図２】



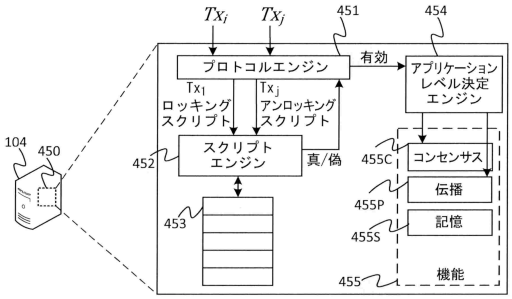
【図３Ａ】



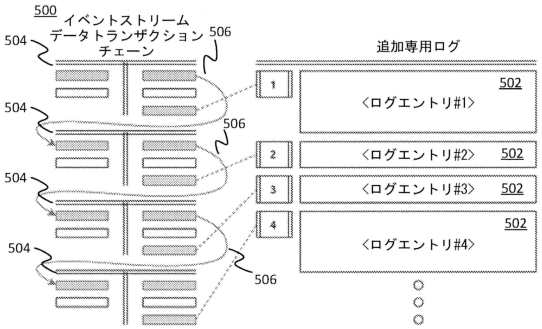
【図３Ｂ】



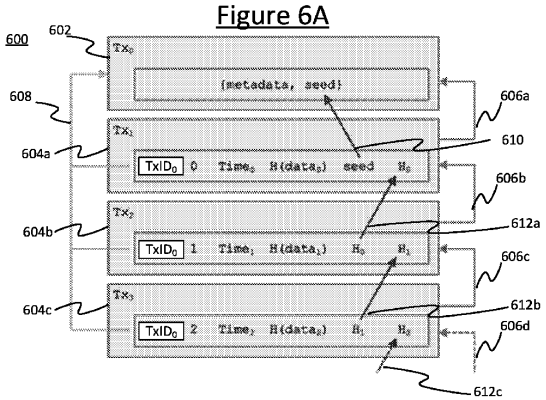
【図４】



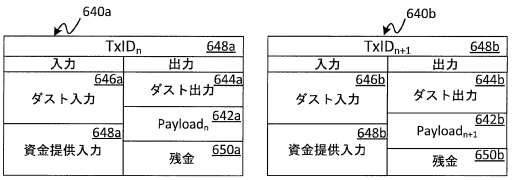
【図５】



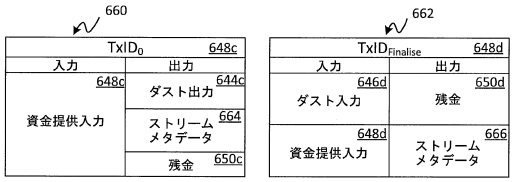
【図６Ａ】



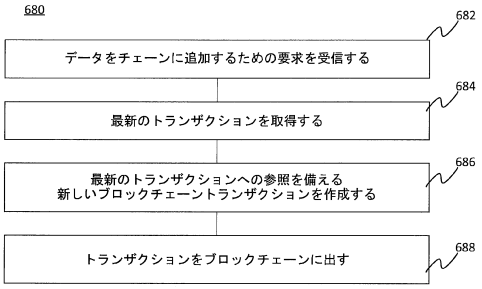
【図６Ｂ】



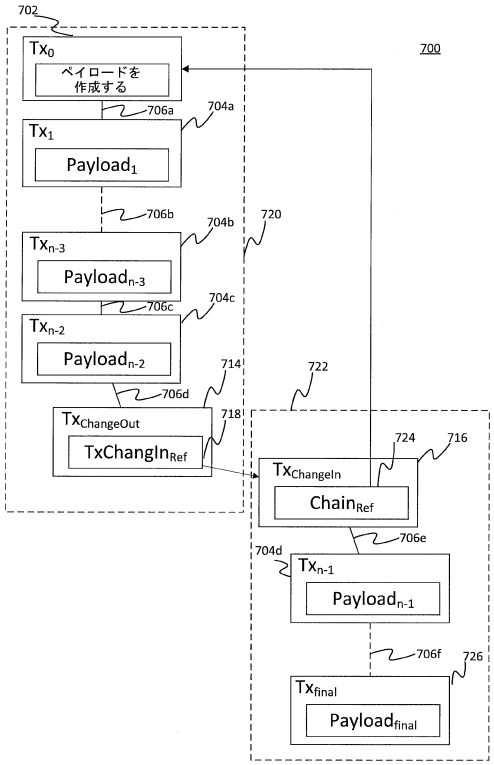
【図６Ｃ】



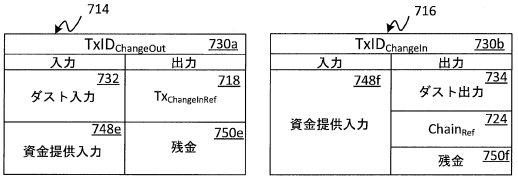
【図６Ｄ】



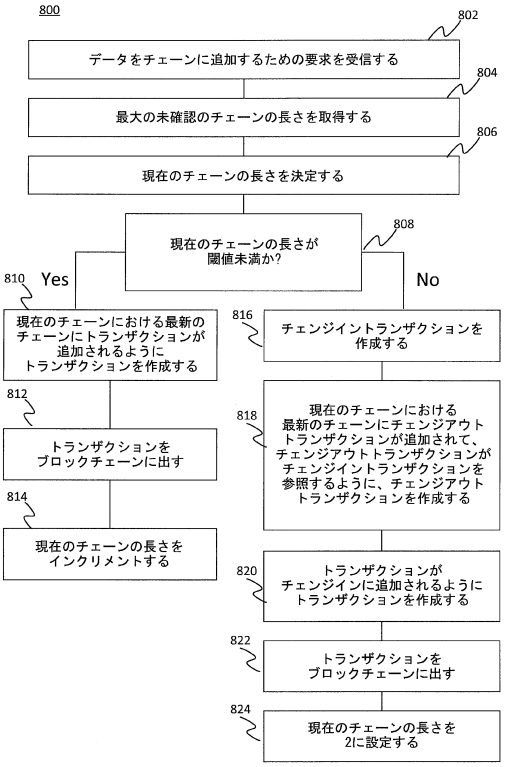
【図７Ａ】



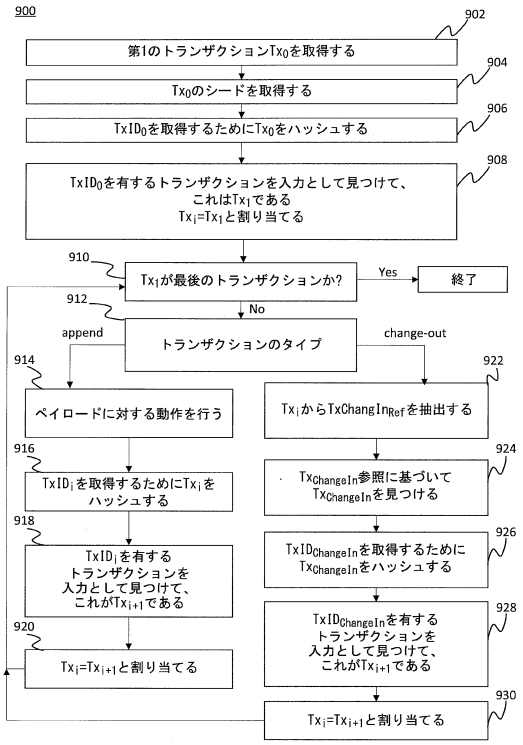
【図７Ｂ】



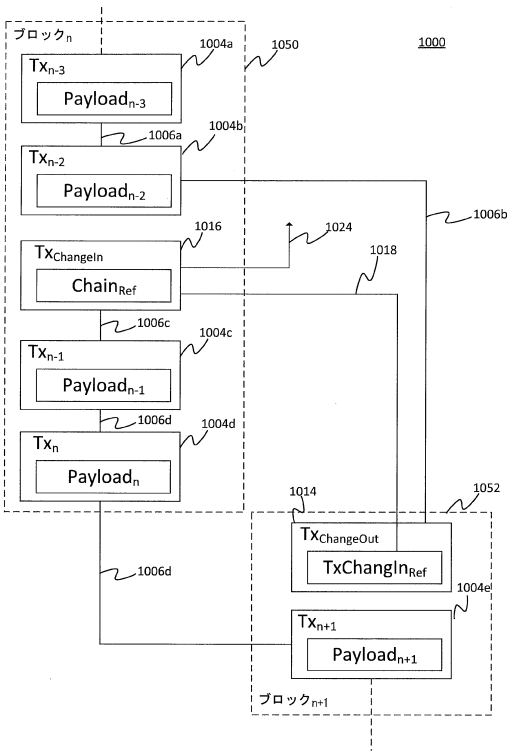
【図８】


800
802
データをチェーンに追加するための要求を受信する
804
最大の未確認のチェーンの長さを取得する
806
現在のチェーンの長さを決定する
808
現在のチェーンの長さが閾値未満か?
Yes
No
810
現在のチェーンにおける最新のチェーンにトランザクションが追加されるようにトランザクションを作成する
812
トランザクションをブロックチェーンに出す
814
現在のチェーンの長さをインクリメントする
816
チェンジイントランザクションを作成する
818
現在のチェーンにおける最新のチェーンにチェンジアウトトランザクションが追加されて、チェンジアウトトランザクションがチェンジイントランザクションを参照するように、チェンジアウトトランザクションを作成する
820
トランザクションがチェンジインに追加されるようにトランザクションを作成する
822
トランザクションをブロックチェーンに出す
824
現在のチェーンの長さを2に設定する

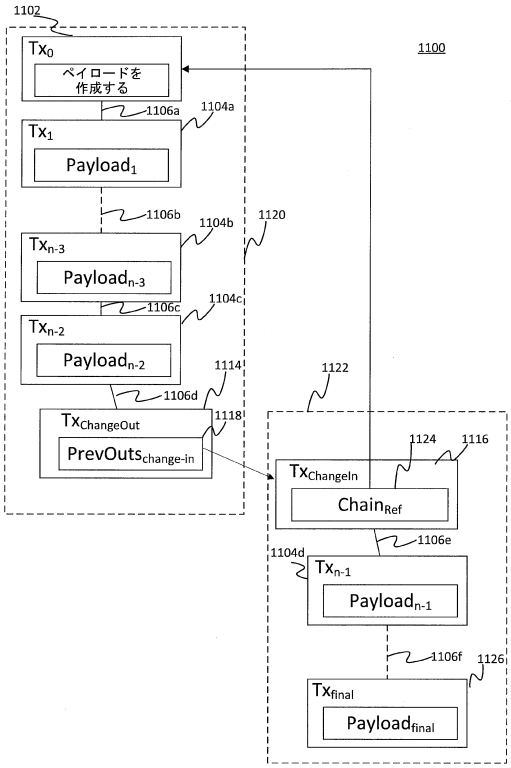
【図９】


900
902
第1のトランザクションTx0を取得する
904
Tx0のシードを取得する
906
TxID0を取得するためにTx0をハッシュする
908
TxID0を有するトランザクションを入力として見つけて、これはTx1である
Txi=Tx1と割り当てる
910
Txiが最後のトランザクションか?
Yes
終了
No
912
トランザクションのタイプ
append
change-out
914
ペイロードに対する動作を行う
916
TxIDiを取得するためにTxiをハッシュする
918
TxIDiを有するトランザクションを入力として見つけて、これがTxi+1である
920
Txi=Txi+1と割り当てる
922
TxiからTxChangInRefを抽出する
924
TxChangeIn参照に基づいてTxChangeInを見つける
926
TxIDChangeInを取得するためにTxChangeInをハッシュする
928
TxIDChangeInを有するトランザクションを入力として見つけて、これがTxi+1である
930
Txi=Txi+1と割り当てる

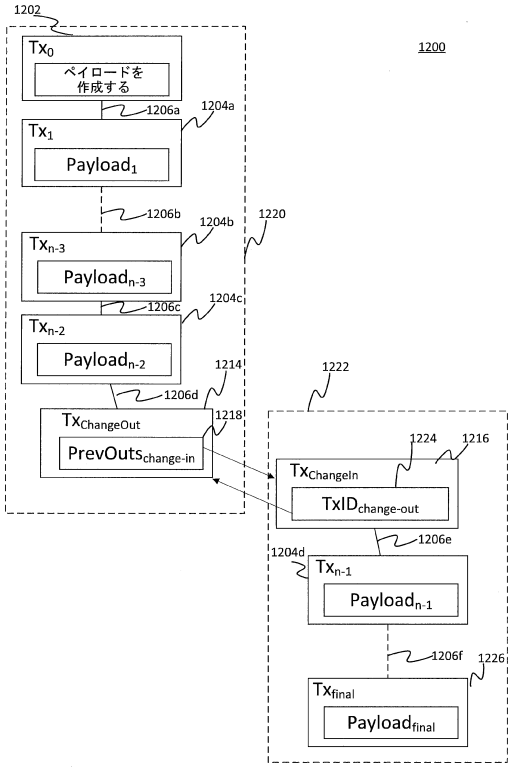
【図１０】


ブロックn
1000
1004a
1050
Txn-3
Payloadn-3
1006a
1004b
Txn-2
Payloadn-2
1006b
1016
1024
1018
TxChangeIn
ChainRef
1006c
1004c
Txn-1
Payloadn-1
1006d
1004d
Txn
Payloadn
1014
1052
TxChangeOut
TxChangInRef
1004e
Txn+1
Payloadn+1
1006d
ブロックn+1

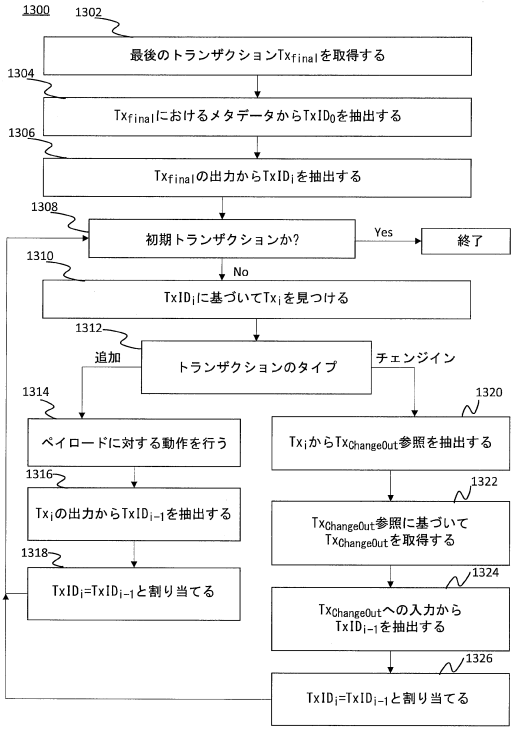
【図１１】


1102
1100
Tx0
ペイロードを作成する
1106a
1104a
Tx1
Payload1
1106b
1104b
1120
Txn-3
Payloadn-3
1106c
1104c
Txn-2
Payloadn-2
1114
1106d
1122
TxChangeOut
PrevOutschange-in
1118
1124
1116
TxChangeIn
ChainRef
1106e
1104d
Txn-1
Payloadn-1
1106f
1126
Txfinal
Payloadfinal

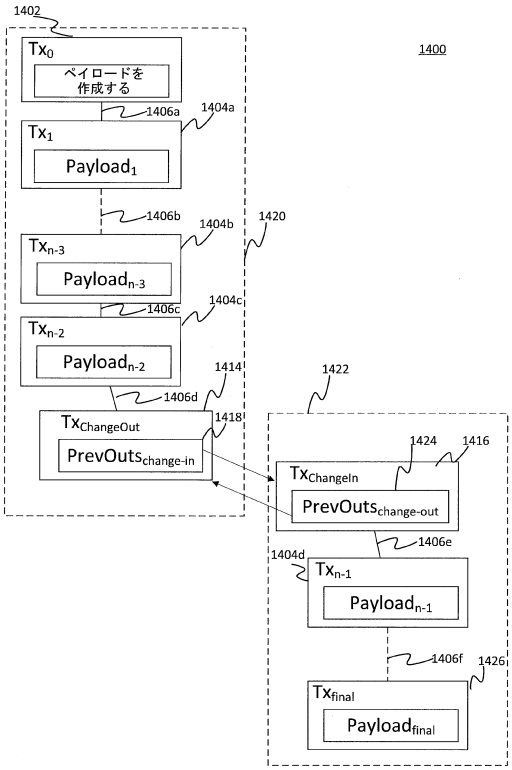
【図１２】



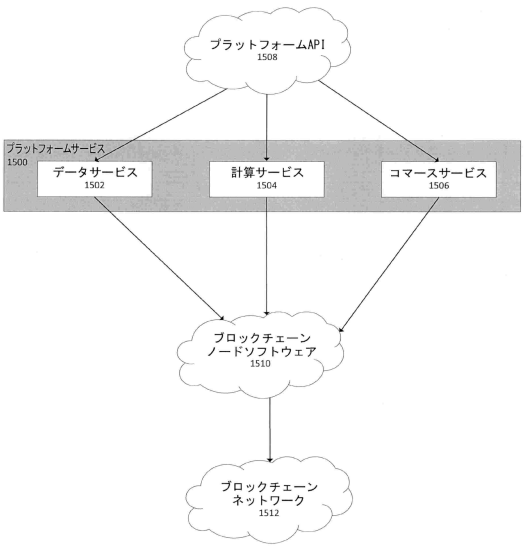
【図１３】



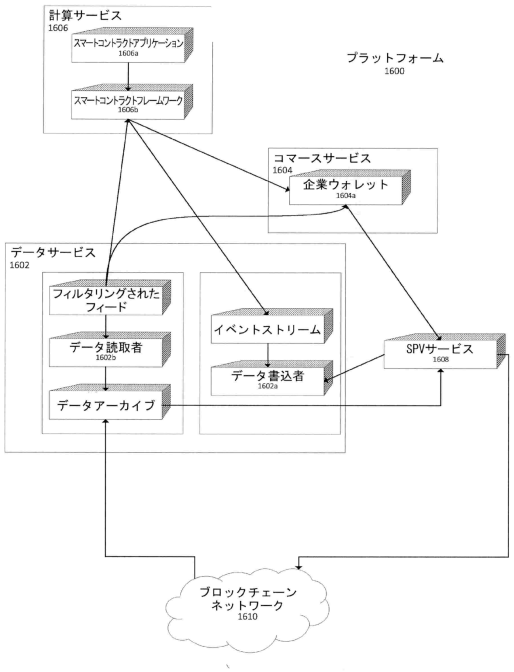
【図１４】



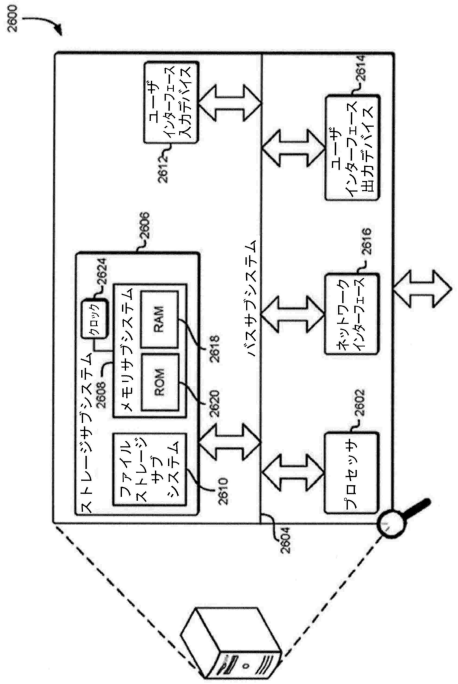
【図１５】



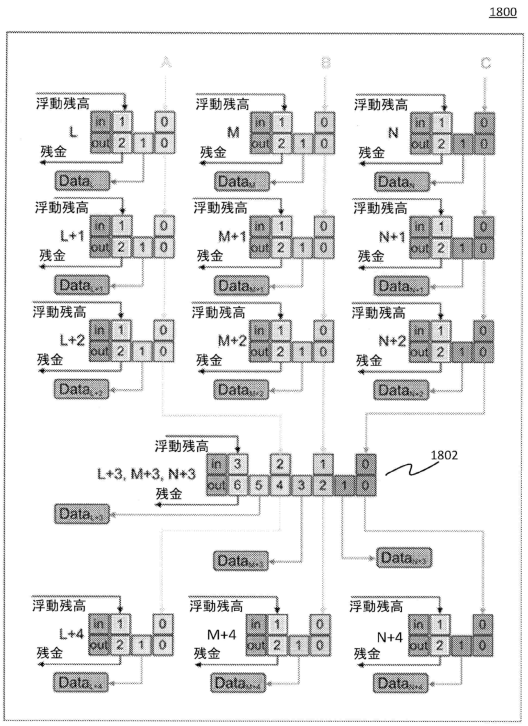
【図１６】


計算サービス
1606
スマートコントラクトアプリケーション
1606a
スマートコントラクトフレームワーク
1606b
プラットフォーム
1600
コマースサービス
1604
企業ウォレット
1604a
データサービス
1602
フィルタリングされたフィード
データ読取者
1602b
イベントストリーム
データ書込者
1602a
データアーカイブ
SPVサービス
1608
ブロックチェーンネットワーク
1610

【図１７】


2600
ユーザインターフェース入力デバイス
2612
ユーザインターフェース出力デバイス
2614
2606
バスサブシステム
2624
クロック
RAM
2618
ROM
2620
メモリサブシステム
2608
ストレージサブシステム
ファイルストレージサブシステム
2610
ネットワークインターフェース
2616
プロセッサ
2602
2604

【図１８】


1800
A
B
C
浮動残高
L
残金
M
N
L+1
M+1
N+1
L+2
M+2
N+2
L+3, M+3, N+3
1802
L+4
M+4
N+4

フロントページの続き

(33)優先権主張国・地域又は機関
英国(GB)
(31)優先権主張番号 2102314.8
(32)優先日 令和3年2月18日(2021.2.18)
(33)優先権主張国・地域又は機関
英国(GB)
ピー内
(72)発明者 リッキー・チャールズ・ランド
イギリス・カーディフ・ＣＦ１０・２ＨＨ・チャーチル・ウェイ・チャーチル・ハウス・セブンス・フロア・アーカート－ダイクス・アンド・ロード・エルエルピー内
(72)発明者 ジャック・オーウェン・デイヴィーズ
イギリス・カーディフ・ＣＦ１０・２ＨＨ・チャーチル・ウェイ・チャーチル・ハウス・セブンス・フロア・アーカート－ダイクス・アンド・ロード・エルエルピー内
審査官 平井 誠
(56)参考文献 国際公開第２０１９／０４３５３８（WO，Ａ１）
特表２０２０－５３２２１７（ＪＰ，Ａ）
(58)調査した分野 (Int.Cl.，ＤＢ名)
Ｇ０９Ｃ １／００－ ５／００
Ｈ０４Ｌ ９／００－４０