(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)公表番号
特表2024-540057
(P2024-540057A)
(43)公表日　令和6年10月31日(2024.10.31)

(51)国際特許分類　FI
H04L 9/32 (2006.01)　H04L 9/32 200Z

審査請求　未請求　予備審査請求　未請求　（全54頁）

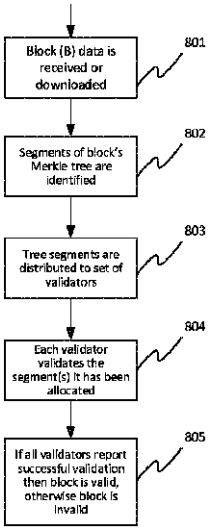
(21)出願番号　特願2024-525218(P2024-525218)
(86)(22)出願日　令和4年10月25日(2022.10.25)
(85)翻訳文提出日　令和6年4月26日(2024.4.26)
(86)国際出願番号　PCT/EP2022/079825
(87)国際公開番号　WO2023/072955
(87)国際公開日　令和5年5月4日(2023.5.4)
(31)優先権主張番号　2115511.4
(32)優先日　令和3年10月28日(2021.10.28)
(33)優先権主張国・地域又は機関
英国(GB)
(81)指定国・地域　AP(BW,GH,GM,KE,LR,LS,MW,MZ,NA,RW,SD,SL,ST,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,RU,TJ,TM),EP(AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,
最終頁に続く

(71)出願人　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　100107766
弁理士　伊東 忠重
(74)代理人　100229448
弁理士　中槇 利明
(72)発明者　ライト，クレイグ スティーヴン
イギリス ダブリュー１ダブリュー ８エーピー ロンドン マーケット プレイス ３０ エヌチェーン ライセンシング アーゲー 内

(54)【発明の名称】　コンピュータ実装システムおよび方法

(57)【要約】
本開示は、データ記録の分散および／または並列処理のための方法およびシステムを提供し、特に、ブロックチェーンブロックにおけるブロックチェーントランザクションの妥当性確認を提供する。好ましい実施形態では、１つまたは複数のトランザクションが複数の妥当性確認リソースのうちの１つの妥当性確認リソースに割り当てられる分散型妥当性確認ノードが開示される。１つまたは複数のトランザクションは、ブロックのマークルツリーの一部に関連し、その結果、各妥当性確認リソースは、ブロックのトランザクションのサブセットの検証時に独立して動作することができ、各サブセットは、マークルツリーのセグメントに基づく。本開示は、少なくとも、異なる妥当性確認リソースへのツリーセグメントの割り当て、ロードバランシング、妥当性確認されるべきトランザクションのダウンロード、分散ＵＴＸＯプール、インデキシング方式、および二重使用イベントの防止のための有利な技法を含む。

Figure 8



【特許請求の範囲】
【請求項１】
複数のブロックチェーントランザクションと前記ブロックのためのマークルツリーのルートとを含むブロックチェーンブロックの少なくとも一部を妥当性確認するコンピュータ実装方法であって、
前記方法は、
前記ブロックチェーントランザクションのそれぞれのサブセットを複数の妥当性確認リソースに割り当てるステップであって、各それぞれのサブセットは、前記マークルツリーのそれぞれの部分を提供し、前記マークルツリーのそれぞれの内部ノードによって表される、ステップと、
前記複数の妥当性確認リソースを使用して、ブロックチェーントランザクションのそれぞれのサブセットを妥当性確認するステップ
を含む方法。
【請求項２】
前記ブロックチェーンブロックおよび／またはブロックチェーントランザクションのサブセットを妥当性確認するステップは、
ｉ）少なくとも１つのブロックチェーントランザクションを妥当性確認および／もしくは検証するステップ、ならびに／または
ｉｉ）簡易支払い検証（ＳＰＶ）プロセスの少なくとも一部を実行するステップ、ならびに／または
ｉｉｉ）所与のブロックチェーントランザクション（Ｔｘ）が前記ブロックチェーンブロック内に含まれているかどうかを確認するステップ、ならびに／または
ｉｉｉ）前記ブロックチェーントランザクションのうちの少なくとも１つのブロックチェーントランザクションのハッシュを生成し、前記ハッシュを使用してマークルパスを構築し、および／もしくは前記ハッシュが前記ブロックチェーンブロックのヘッダ内のトランザクション識別子（ＴｘＩＤ）と一致するかどうかチェックするステップ
を含む、請求項１に記載の方法。
【請求項３】
前記ブロックチェーントランザクションのサブセットのうちの少なくとも１つは、前記サブセットに関連付けられている、前記サブセットを識別する、および／または前記サブセットを表す識別子を含む、
請求項１または２に記載の方法。
【請求項４】
前記識別子は、前記マークルツリー内での前記少なくとも１つのサブセットの前記位置の計算を容易にする、
請求項３に記載の方法。
【請求項５】
前記識別子は、前記ブロックチェーントランザクションの少なくとも１つのサブセット内のブロックチェーントランザクションのハッシュの一部を含む
請求項３に記載の方法。
【請求項６】
前記ブロックチェーントランザクションのそれぞれのサブセットを前記複数の妥当性確認リソースに割り当てる前記ステップは、前記トランザクションのサブセットに関連付けられたそれぞれの識別子に基づいて、前記それぞれのサブセットをそれぞれの妥当性確認リソースにマッチングさせることを含む、
請求項１に記載の方法。
【請求項７】
ｉ）前記複数の妥当性確認リソースのうちの少なくとも１つにブロックチェーントランザクションの少なくとも１つのサブセットをダウンロードするステップ、および／または
ｉｉ）前記複数の妥当性確認リソースのうちの少なくとも１つにブロックチェーントラ

ンザクションの少なくとも１つのサブセットを送信するステップ
をさらに含む、請求項１に記載の方法。
【請求項８】
前記マークルツリーは、前記複数のブロックチェーントランザクションのハッシュの二分木またはメッシュを含む、
請求項１に記載の方法。
【請求項９】
前記複数のブロックチェーントランザクション内の前記ブロックチェーントランザクションのサブセットを識別および／または決定するステップ
をさらに含む、請求項１に記載の方法。
【請求項１０】
前記複数の妥当性確認リソースのうちの少なくとも１つは、仮想マシン、サーバ、ＧＰＵベースのコンピューティングリソース、処理スレッド、および／またはマルチプロセッサシステムのうちの少なくとも１つであるか、またはそれを含む、
請求項１に記載の方法。
【請求項１１】
ｉ）前記複数のブロックチェーントランザクション内の少なくとも２つのトランザクションは前記マークルツリーにおける兄弟であり、および／または
ｉｉ）前記それぞれの内部ノードは、前記ブロックチェーントランザクションのそれぞれのサブセットの親または祖先である、
請求項１に記載の方法。
【請求項１２】
複数のブロックチェーントランザクションと前記ブロックのためのマークルツリーのルートとを含むブロックチェーンブロックの少なくとも一部を妥当性確認するように動作するシステムであって、
前記システムは、複数の妥当性確認リソースを備え、各々の妥当性確認リソースは、
プロセッサと、
実行可能命令を含むメモリと
を備え、前記実行可能命令は、前記プロセッサによる実行の結果として、前記システムに、請求項１に記載のコンピュータ実装方法を実行させる、
システム。
【請求項１３】
コンピュータシステムのプロセッサによって実行された結果として、前記コンピュータシステムに、請求項１に記載のコンピュータ実装方法を実行させる実行可能な命令を記憶した非一時的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、概して、関連するまたは関連付けられたデータ記録を処理するための改善された方法およびシステムに関する。本開示は、ブロックチェーントランザクションのマイニング前および／または後の妥当性確認、ＳＰＶチェックなど、ブロックチェーンネットワークを介してまたはブロックチェーンネットワークを使用して達成される転送に関する使用に特に適しているが、これに限定されない。利点には、セキュリティおよび回復力の向上、効率の向上、または速度およびリソース要件の低減、ならびに従来技術の構成では不可能であった妥当性確認に対する新規の手法が含まれ、これにより、これまで不可能であったブロックチェーン実装構成をもたらすが、これらに限定されない。
【背景技術】
【０００２】
ビットコインプロトコルおよびネットワークは、実装のための例示的な文脈を提供する目的のために本明細書で言及され得るが、本開示は、ビットコインブロックチェーンでの

使用に限定されるものではなく、代替のプロトコルおよび実装がその範囲内に含まれる。

【０００３】

ブロックチェーンは、ブロックから構成されるコンピュータベースの非集中型分散システムとして実装されるピアツーピアの電子台帳であり、ブロックはトランザクションから構成される。各トランザクションは、ブロックチェーンシステムの参加者間でのデジタル資産の制御の転送を符号化するデータ構造であり、少なくとも１つの入力および少なくとも１つの出力を含む。各ブロックは、前のブロックのハッシュを含み、それらのブロックが一緒に連鎖されて、開始以来ブロックチェーンに書き込まれてきたすべてのトランザクションの永久的で変更不可能な記録を作成する。

【０００４】

トランザクション（Ｔｘ）がブロックチェーンに書き込まれるためには、「妥当性確認」されなければならない。ネットワークノード（マイナー）は、各トランザクションが有効であることを確実にする作業を実行し、無効なトランザクションはネットワークから拒否される。ノードにインストールされたソフトウェアクライアントは、そのロックスクリプトおよびロック解除スクリプトを実行することによって、未使用トランザクション（ＵＴＸＯ）に対してこの妥当性確認作業を実行する。ロックスクリプトおよびロック解除スクリプトの実行がＴＲＵＥと評価された場合、トランザクションは有効であり、トランザクションはブロックチェーンに書き込まれる。したがって、トランザクションがブロックチェーンに書き込まれるためには、トランザクションは、ｉ）トランザクションを受信する第１のノードによって妥当性確認され、トランザクションが妥当性確認された場合、ノードはそれをネットワーク内の他のノードに中継すること、すなわち伝搬されること、ｉｉ）マイナーによって構築された新しいブロックに追加されること、およびｉｉｉ）マイニング、すなわち過去のトランザクションの公開台帳に追加されること、が行わなければならない。トランザクションがＵＴＸＯとしてブロックチェーンに記憶されると、ユーザは、関連付けられた暗号通貨の制御を、後にブロックチェーンに書き込まれる別のトランザクションにおける入力に関連付けられた別のアドレスに移すことができる。これは、ユーザの暗号通貨に関連付けられた公開鍵と私有鍵のペアを記憶するデジタルウォレットを使用して行われることが多い。ＳＰＶ（Simplified Payment Verification）ウォレットを含む様々な形態の既知の暗号通貨ウォレットが存在する。ＳＰＶ技法により、ユーザおよびマーチャントノードは、特定の転送に関連する部分的な情報のみに基づいてローカル検証を実行することができる。ＳＰＶについては、以下でより詳細に説明する。

【０００５】

しかしながら、セキュリティ、所与のブロックチェーンのための関連プロトコルとの適合性、および二重使用エクスプロイトに対する保護を保証するためには妥当性確認が不可欠であることが知られているが、そのような妥当性確認タスクでは、ブロックをダウンロードして記憶し、大規模なＵＴＸＯプールを維持し、検証に必要な処理タスクを実行する必要性により、かなりのリソースおよび時間が必要となる可能性があることが認識されている。多くのユーザは、そのような要件を満たすことができないか、場合によっては満たす必要がないので、満たさない方を好むかのいずれかである。したがって、セキュリティを損なったり既存のプロトコルの適合を必要としたりすることなく、少なくともこれらの課題（および他の課題）に対処する、より高速でより効率的な検証モデルが必要とされる。このような改善されたソリューションが考案された。

【発明の概要】

【０００６】

本開示の実施形態は、改善されたブロックチェーン関連の方法、デバイス、およびシステムを提供する。１つの表現形式によれば、そのような実施形態は、ブロックチェーントランザクションおよび／またはブロックチェーンブロックの一部もしくは全体を妥当性確認するためのソリューションを提供する。追加または代替の表現形式によれば、それらは、ブロックチェーントランザクションの処理に対する既知の手法の効率、リソース要件、速度および／または回復力を制御、管理および／または強化するためのセキュアなソリュ

ーションを提供する。実施形態はまた、ブロックチェーン実装ソリューションのスケーラビリティを可能にし、デジタルリソースの電子転送のための改善された方法および技術的アーキテクチャを提供する。

【０００７】

本開示の実施形態は、様々な装置によって部分的または全体的に実装され得る。これらは、１つまたは複数の仮想マシン、サーバ、ＧＰＵベースのコンピューティングリソース、またはマルチプロセッサシステムを含む（がこれらに限定されない）、ハードウェアおよび／またはソフトウェアベースの装置であり得る。追加的または代替的に、実施形態は、１つまたは複数のデジタルウォレットを含み得る。しかしながら、重要なことに、実施形態は、ブロックチェーン関連の妥当性確認タスクの分散処理のためのメカニズムを提供する。分散プロセスの調整、管理、および制御は、関与するハードウェア構成要素とソフトウェア構成要素との間の対話の全体的な理解を必要とするので、本質的に技術的な性質であることが知られており、そのような分散型ソリューションの実装は、技術的に些細なものを超えて拡張する。

【０００８】

実施形態は、複数の処理リソースにわたる妥当性確認タスクの分散を可能または容易にするソリューションを含み得、便宜上、これを「バリデータ」と呼ぶ。バリデータは、単一の処理リソースを含んでもよいし、集合的に妥当性確認リソースとみなされることができる複数の関連する処理リソースを含んでもよい。

【０００９】

一例では、トランザクションのブロックが妥当性確認および／またはダウンロードされる必要があるとき、そのマークルツリーは、１つまたは複数のより小さいセグメントに分解され得、各セグメントは、それ自体のルートを有し、ブロック内のトランザクションのサブセットを表すツリー構造を含む。次いで、これらのセグメントを異なるバリデータに割り当てることができる。各バリデータは、それに割り当てられたトランザクションのサブセットに対して必要な処理タスクを実行するように動作する。

【００１０】

バリデータへのツリーセグメントの割り当ては、様々な方法で実行され得るが、１つの有利な実施形態によれば、ランダムに生成された、ダブルハッシュされたマークルルートの先頭数字（leading digit）を使用して、一致するバイナリ識別子を有するバリデータ（またはバリデータのグループ／クラスタ）に所与のセグメントを割り当てるバイナリインデキシングシステム（binary indexing system）が使用され得る。これにより、複数のバリデータにわたるロードバランシングのための単純で効率的かつ迅速なメカニズムを提供される。

【００１１】

各ツリーセグメントは、バリデータがその動作を終了した後にトランザクションのマークルツリー全体の再構成を可能にする小さなバイナリマーカを含み得る。セグメントマーカにより、コントローラ構成要素は、セグメントをそれらの元の形式に再構築することができ、マーカは元のツリー内の位置を示す。これによって、複数のツリーセグメントは、潜在的には地球全体のどこにでもある、多くの異なるバリデータにわたって分散されるが、それらを迅速かつ容易に再構築して、ブロックの完全なマークルツリーを提供することができるという利点を提供する。

【００１２】

１つまたは複数の実施形態では、バリデータは、それが実行した作業に関するデータおよび／またはそれが処理したデータを記録するリポジトリを含むかまたはリポジトリへのアクセスを有し得る。一実施形態では、これは、処理のために所与のバリデータに割り当てられた未使用トランザクション出力（ＵＴＸＯ）を含むデータベースを含み得る。従来のモデルでは、ブロックチェーン上のすべてのＵＴＸＯは、ＵＴＸＯプールと呼ばれるデータベース内のノードによって追跡されることを想起されたい。各フルノードは、ブロックチェーン用のＵＴＸＯプールの自身の完全なコピーを有する。しかしながら、本開示に

よれば、各バリデータは、妥当性確認のためにそれに割り当てられたトランザクションのＵＴＸＯを追跡する自身のＵＴＸＯプールを有する異なる手法が利用され得る。そのような分散型ＵＴＸＯプールの利点には、データ完全性の保証、速度および効率の向上、ならびにＳＰＶのような様々な妥当性確認技法の組み込みおよびサポートが含まれるが、これらに限定されない。

【図面の簡単な説明】

【００１３】

本開示の実施形態の理解を助け、そのような実施形態がどのように実施され得るかを示すために、単なる例として添付の図面を参照する。

【図１】ブロックチェーンを実装するためのシステムの概略ブロック図である。

【図２】ブロックチェーンに記録され得るトランザクションのいくつかの例を概略的に示す。

【図３】当技術分野で知られている一般的なマークルツリー構造の図を提供する。

【図４】当技術分野で知られているように、マークルルートをブロックチェーントランザクションのセットからどのように導出することができるかを示す。

【図５】本開示の一実施形態による、マークルツリーをサブセット（または「セグメント」）にどのように分割することができるかについての例を提供し、サブセットは次いで、それぞれの妥当性確認リソースに割り当てられ得る。

【図６】マークルツリーを論理セグメントにどのように分割し得るかについての図５の代替的な例を示す。

【図７】本開示の例示的な実施形態による、分散型妥当性確認ノードをシステムレベルで示す。

【図８】本開示の例示的な方法に関与するステップを大まかに示したフローチャートである。

【図９】図７の例示的なシステムをより詳細に示す図である。

【発明を実施するための形態】

【００１４】

ここで、限定ではなく例示を目的として、添付の図面を参照して、本開示の例示的な実施形態を説明する。

【００１５】

従来、ブロックチェーンネットワーク内のノードは、ブロックチェーン上のすべてのトランザクションのグローバル台帳を維持する。グローバル台帳は、分散型台帳であり、各ノードは、グローバル台帳の完全または部分的なコピーを記憶し得る。グローバル台帳に影響を及ぼすノードによるトランザクションは、グローバル台帳の有効性および完全性が維持されるように、他のノードによって検証される。ビットコインプロトコルを使用するものなど、ブロックチェーンネットワークの実装および運用の詳細は、当業者であれば理解するであろう。

【００１６】

各トランザクションは、典型的には、１つまたは複数の入力および１つまたは複数の出力を有する。入力および出力に埋め込まれたスクリプトは、トランザクションの出力に誰がどのようにアクセスすることができるかを指定する。トランザクションの出力は、トランザクションの結果として値の制御が転送されるアドレスであり得る。次いで、その値は、未使用トランザクション出力（ＵＴＸＯ）としてその出力アドレスに関連付けられる。次いで、後続のトランザクションは、その値の制御または所有権を取得するために、そのアドレスを入力として参照し得る。

【００１７】

上述のように、ビットコインネットワークおよびプロトコルを例として使用すると、マイニングノードは、ブロックチェーン内の次のブロックを作成しようと競う。ブロックを組み立てるために、マイナーは、未確認トランザクションのプール（「ｍｅｍｐｏｏｌ」）からのトランザクションのセットとしてブロックを構築する。次いで、マイナーは、それが

組み立てたブロックに関してプルーフオブワーク（ＰｏＷ）パズルを完成させようとする。マイナーは、他のマイナーが自身のブロックの生成およびそのＰｏＷの完成に成功したという通知を受信する前に、何とかＰｏＷを完成させた場合、自身のブロックをネットワーク上のピアノードに送信することによって伝搬する。それらのノードは、ブロックを妥当性確認し、次いで、それをネットワーク内で他のノードにさらに送信する。マイナーが自身のＰｏＷを終える前に、別のブロックが完成したという通知を受信した場合、その労力を放棄し、次のブロックを構築しようとし始める。

【００１８】

したがって、ブロックの高速伝搬は、マイナーおよび妥当性確認ノードに代わって無駄な労力（および関連するエネルギー）を回避するのに役立つ。より高速な妥当性確認、ひいてはブロックの伝搬を可能にするソリューションを提供することによって、本発明は、ネットワーク性能の強化を提供する。これにより、必要とされる計算時間および労力の量が削減され、ネットワークによって必要とされるエネルギーの量も削減される。リソースおよび時間に関してより効率的なネットワークが提供される。最終的に、改善された（ブロックチェーン）ネットワークが提供される。

【００１９】

ビットコインネットワークなどのブロックチェーンの現在の実装では、ブロックを受信する各ノードは、最初にブロックを妥当性確認してから他のノードにそれを送信する。ブロックの妥当性確認に時間がかかると、ネットワークを介したブロックの伝搬が遅くなる。既存のプロトコルの進化を含む、ブロックチェーンの一部の実装は、ネットワーク内の各ノードではなくノードのサブセットのみによるブロック妥当性確認を提供し得るが、無効なブロックがネットワークを介して伝搬することを防止するために、ほとんどのノードにおけるブロック妥当性確認は、依然として任意のブロックチェーン実装の特徴である可能性が高いことに留意されたい。

【００２０】

ブロックを妥当性確認することは、ブロックが適用可能なブロックチェーンプロトコルによって設定された所定の基準を満たすことを確認することを含む。ビットコインプロトコルに適用可能な例示的な基準は、CheckBlockおよびCheckBlockHeaderなどの関数を含み得る。ブロック自体が所定の基準に適合することを確認することに加えて、ブロック内の各トランザクションは、トランザクションレベル基準との適合性について評価され得る。一例として、ビットコインプロトコルにおいて適用されるトランザクションレベル基準は、関数AcceptToMemoryPool、CheckTransaction、およびCheckInputsを含み得る。

【００２１】

ビットコインプロトコルに基づくブロックレベル基準の具体例には以下が含まれ得る：

・　ブロックデータ構造は構文的に有効である。

・　ブロックヘッダのハッシュは、（プルーフオブワークを実施する）目標難易度未満である。

・　ブロックタイムスタンプは、（時間誤差を許容して）２時間未満の未来である。

・　ブロックサイズは許容限度内である。

・　第１のトランザクション（第１のトランザクションのみ）は、コインベース生成トランザクションである。

・　ブロック内のすべてのトランザクションが有効である。

【００２２】

ビットコインプロトコルに基づくトランザクションレベル基準の具体例には以下が含まれ得る：

・　トランザクションのシンタックスとデータ構造は正しくなければならない。

・　入力のリストも出力のリストも空であってはならない。

・　各出力値ｘならびにすべての出力の合計は、$0 < x < 21 \cdot 10^6$の範囲内になければならない。

・　どの入力もヌルハッシュを有しない。
・　nLockTimeは、INT_MAX以下である。
・　バイト単位のトランザクションサイズは、最小値以上最大値未満である。
・　署名動作の数は、署名動作限度未満である。
・　ロック解除スクリプトscriptSigは、スタック上に数をプッシュすることのみができ、ロックスクリプトscriptPubkeyは、isStandard形式に一致しなければならない。
・　各入力について、参照された出力がプール内の任意の他のトランザクション内に存在する場合、そのトランザクションは拒否されなければならない。
・　各入力について、参照された出力トランザクションがコインベース出力である場合、少なくともＣＯＩＮＢＡＳＥ＿ＭＡＴＵＲＩＴＹ（１００）承認（confirmation）が必要である。
・　各入力について、参照された出力は存在していなければならず、使用済みであってはならない。
・　参照された出力トランザクションを使用して入力値を取得し、各入力値および合計が値ｘの許容範囲、すなわち０＜ｘ＜２１・１０^{6}内にあることをチェックする。
・　プール内または主分岐のブロック内に一致するトランザクションが存在しなければならない。
・　入力値の和は、出力値の和以上でなければならない。
・　トランザクション手数料は、空きブロックへのエントリを得るのに十分でなければならない。
・　各入力に対するロック解除スクリプトは、対応する出力ロックスクリプトに照らして妥当性確認されなければならない。

【００２３】

これらの例示的な基準は、例示的なものであり、所定の基準は、プロトコルによって異なり得、プロトコルに変更が行われる場合、所与のプロトコルについて経時的に変化し得るので、すべての実施形態に対して十分または必要であると解釈されるべきではない。一般に、トランザクションレベル妥当性確認基準は、適用可能なブロックチェーンプロトコルの下で有効であるとみなされるためにトランザクションが有さなければならない所定の特性である。同様に、ブロックレベル妥当性確認基準は、適用可能なブロックチェーンプロトコルの下で有効であるとみなされるためにブロックが有さなければならない所定の特性である。

【００２４】

本出願にしたがって、ネットワークにおけるブロックのより高速な伝搬を容易にするためにブロック妥当性確認を高速化する方法およびデバイスが説明される。

【００２５】

一態様では、本出願は、個々のトランザクションの少なくともトランザクションレベル妥当性確認を並行しておよび／または分散方式で実行することによって、ブロックを妥当性確認するように構造化されたノードを説明する。しかしながら、特定のトランザクションレベル基準は、並行して評価されなくてもよい。例えば、ＵＴＸＯの一意性は、シリアルベースで評価され得る。そのような場合、本開示の分散型妥当性確認ノードは、残りのトランザクションレベル基準の妥当性確認のために２つ以上の並列プロセッサのセットの間でトランザクションのセットを割り当てる前に、トランザクションの被参照入力（ＵＴＸＯ）の一意性を確認するように構造化または構成され得る。

【００２６】

特に、本開示の実施形態は、ツリー構造に記憶された関連するまたは関連付けられたデータ記録を処理するための改善された検証およびセキュリティソリューションを提供する。ツリーは、二分木またはメッシュ構造とすることができる。当技術分野で知られているように、ツリー構造は、より小さいツリー（本明細書では、ツリー「セグメント」、「サブセット」、または「部分」と呼ばれることがある）に分解することができ、各セグメントは、ツリー全体におけるデータ記録のサブセットを含み、それ自体のルートを有する。

有利なことに、本開示の実施形態は、この特徴を利用して、複数の処理リソースにわたる関連データ記録の処理の分散および並列化のための方法およびシステムを提供する。

【００２７】

本願の例示的な実施形態では、複数のデータ記録は、それらがマークルツリー内のノードを形成するので、関連するブロックチェーントランザクションを含む。マークルツリーは、ブロックチェーンプロトコルにしたがってトランザクションのブロックのヘッダに含まれた、または含まれることができるルートを有し、これにより、ルートは、ツリー内のすべてのリーフ（すなわち、トランザクションＩＤ（ＴｘＩＤ））まで辿ることができるパスを提供する。本願の例では、ブロックチェーンプロトコルはビットコインプロトコルであるか、またはビットコインプロトコルから導出されるが、他のプロトコルも本開示の範囲内に入る。

【００２８】

本発明の例では、複数のトランザクションを処理することは、複数のブロックチェーントランザクションとブロックのためのマークルツリーのルートとを含むブロックチェーンブロックの少なくとも一部を妥当性確認することを含む。これらの例は非限定的であり、本明細書に開示される技法は、非ブロックチェーン関連データに関して、および／または妥当性確認以外の他のプロセスに関して利用され得る。例えば、実施形態は、マークルツリーで表すことができるあらゆるタイプのデータ記録を記憶、構造化、検索、および／または維持するために使用され得る。ブロックチェーン台帳の代わりに、またはそれに加えて、データベースおよび他の既知の記憶リソースが利用されてもよい。

【００２９】

別の例示的な実施形態では、複数のトランザクションを処理することは、複数のブロックチェーントランザクションとブロックのためのマークルツリーのルートとを含むブロックチェーンブロックの少なくとも一部をダウンロードすることを含む。

【００３０】

完全を期すために、図３および図４を参照して、マークルツリーと、ブロックチェーントランザクションのブロックを表す際のマークルツリーの使用とについての議論を提供する。

【００３１】

マークルツリー

図３を参照すると、マークルツリーは、データの集合体のセキュアな検証を可能にする階層データ構造である。マークルツリーでは、ツリー内の各ノードは、インデックスペア（ｉ，ｊ）が与えられており、Ｎ（ｉ，ｊ）と表される。インデックスｉ、ｊは、ツリー内の特定の位置に関連する単なる数値ラベルである。

【００３２】

マークルツリーの特徴は、そのノードの各々のノードの構成が以下の式によって支配されることである：

【数１】

$$N(i,j)=\begin{cases} H(D_i) & i=j \\ H\big(N(i,k)\parallel N(k+1,j)\big) & i\neq j \end{cases}$$

ここで、Ｈは暗号ハッシュ関数である。

【００３３】

これらの式にしたがって構築されたバイナリマークルツリーの例を図３に示す。図示のように、ｉ＝ｊの場合は、単に対応するｉ番目のデータパケットＤ ｉのハッシュであるリーフノードに対応することが分かる。ｉ≠ｊの場合は、１つの親（マークルルート）が見つかるまで再帰的にハッシュし、子ノードを連結することによって生成される内部ノードまたは親ノードに対応する。

【００３４】

例えば、ノードＮ（０，３）は、次のように、４つのデータパケットＤ０，...，Ｄ３から構築される：

【数２】

$$N(0,3) = H\big(N(0,1) \parallel N(2,3)\big)$$
$$= \big[H\big(N(0,0) \parallel N(1,1)\big) \parallel H\big(N(2,2) \parallel N(3,3)\big)\big]$$
$$= \big[H\big(H(D_0) \parallel H(D_1)\big) \parallel H\big(H(D_2) \parallel H(D_3)\big)\big]$$

【００３５】

ツリー深さＭは、ツリー内のノードの最下位レベルとして定義され、ノードの深さｍは、ノードが存在するレベルである。例えば、m_{root}＝０およびm_{leaf}＝Ｍであり、図３ではＭ＝３である。

【００３６】

ビットコインおよびいくつかの他のブロックチェーンにおけるマークルツリーの場合、ハッシュ関数はダブルＳＨＡ２５６であり、これは、標準ハッシュ関数ＳＨＡ－２５６を２回適用するものである：Ｈ（ｘ）＝ＳＨＡ２５６（ＳＨＡ２５６（ｘ））。

【００３７】

マークルツリーの主な機能は、あるデータパケットD_iがＮ個のデータパケット$D \in \{D_0, ..., D_{N-1}\}$のリストまたはセットのメンバであることを検証することである。検証のためのメカニズムは、マークル証明として知られており、所与のデータパケットD_iおよびマークルルートＲについてマークルパスとして知られているハッシュのセットを取得することを含む。データパケットのマークル証明は、単に、繰り返されるハッシュおよび連結によってルートＲを再構築するのに必要とされるハッシュの最小リストであり、多くの場合「認証証明（authentication proof）」と呼ばれる。

【００３８】

存在証明（proof of existence）は、すべてのパケット$D_0, ..., D_{N-1}$およびそれらの順序が証明者（prover）に知られている場合、自明に実行され得る。しかしながら、これは、マークル証明よりもはるかに大きなストレージオーバーヘッドを必要とするとともに、データセット全体が証明者に利用可能であることを必要とする。

【００３９】

マークル証明を使用することとリスト全体を使用することと比較が、以下の表に示されており、ここでは、バイナリマークルツリーを使用し、データブロックの数Ｎが２の整数乗に正確に等しいと仮定する。

【００４０】

以下の表は、マークルツリーにおけるリーフノードの数とマークル証明（またはマークル証明）に必要とされるハッシュの数との間の関係を示す。

【表１】

					マークルツリー
データパケットの数	32	256	1024	1048576	$N = 2^M$
存在証明に必要とされるハッシュの数	5	8	10	20	$M = \log_2 N$

【００４１】

データパケットの数がリーフノードの数に等しいこの簡略化されたシナリオでは、マークル証明を計算するのに必要とされるハッシュ値の数が対数的にスケーリングすることが分かる。Ｎ個のデータハッシュを記憶して明示的な証明を計算するよりも、$\log_2 N$個のハッシュを含むマークル証明を計算する方がはるかに効率的かつ実用的であることは明らかである。

【００４２】

マークルルートＲが与えられた場合に、データブロックD_0がＲによって表される順序付きリストＤ∈｛D_0，...，D_{N-1}｝に属することを証明したい場合、次のようにマークル証明を実行することができる：

ｉ．信頼できるソースからマークルルートＲを取得する。

ｉｉ．ソースからマークル証明Γを取得する。この場合、Γはハッシュの集合である：Γ＝｛Ｎ（１，１），Ｎ（２，３），Ｎ（４，７）｝。

ｉｉｉ．次のように、D_1およびΓを使用してマークル証明を計算する：

ａ．データブロックをハッシュして、以下を得る：Ｎ（０，０）＝Ｈ（D_0）。

ｂ．Ｎ（１，１）と連結し、ハッシュして、以下を得る：Ｎ（０，１）＝Ｈ（Ｎ（０，０）｜｜Ｎ（１，１））。

ｃ．Ｎ（２，３）と連結し、ハッシュして、以下を得る：Ｎ（０，３）＝Ｈ（Ｎ（０，１）｜｜Ｎ（２，３））。

ｄ．Ｎ（４，７）と連結し、ハッシュして、ルートを得る：Ｎ（０，７）＝Ｈ（Ｎ（０，３）｜｜Ｎ（４，７））、Ｒ´＝Ｎ（０，７）。

ｅ．計算されたルートＲ´を（ｉ）で得られたルートＲと比較する：

１．Ｒ´＝Ｒである場合、ツリー内のD_0の存在、したがってデータセットＤが確認される。

２．Ｒ´≠Ｒである場合、証明は失敗に終わり、D_0がＤのメンバであることは確認されない。

【００４３】

これは、マークルツリーおよびそのルートによって表されるデータセットの一部としていくつかのデータの存在証明を提供するための効率的なメカニズムである。例えば、データD_0がブロックチェーントランザクションに対応し、ルートＲがブロックヘッダの一部として公的に利用可能である場合、トランザクションがそのブロックに含まれたことを迅速に証明することができる。

【００４４】

ＳＰＶ

簡易支払い検証（ＳＰＶ）は、Satoshi Nakamotoの２００８年のwhitepaper「Bitcoin: A Peer-to-Peer Electronic Cash System」の第８節に初めて記載された、マークルツリーのこれらの特徴を利用する。アリスとボブとの間のＳＰＶベースの暗号通貨交換では、両方の当事者が同じタイプのＳＰＶウォレットを使用する。ＳＰＶウォレットは、ユーザの私有鍵および公開鍵と、未使用トランザクションと、ブロックチェーン上でブロックが位置特定されることができるようにブロックを一意に識別するブロックヘッダとを記憶する。説明したように、ブロックヘッダは、ブロック全体のコンテンツの一意の要約または指紋を提供するデータのフィールドと、そのブロックのマークルルートを提供するフィールドとを含む。マークルルートは、単一のハッシュに最終的に到達するまで、ブロックからのトランザクションＩＤ（ＴｘＩＤ）のペアを一緒に繰り返しハッシュすることによって生成される。マークルルートは、ウォレットおよびマーチャントノードなどのユーザが、ブロックチェーン全体をダウンロードすることなく、特定のトランザクションをローカルに検証することを可能にするので、トランザクションがブロックの一部であることを検証するための効率的でセキュアなメカニズムを提供する。これは、フルノードを実行する必要がない、または実行することを望まないが、単に、特定のトラン

ザクションが特定のブロックにあるというローカルなチェックを実行する必要があるユーザにとって、例えば、相互間での転送を実行することを望むマーチャントおよび顧客などの当事者にとって有利である。要約すると、ＳＰＶは、そのようなユーザが、ブロックチェーン全体をダウンロードおよび記憶する必要なく、特定のトランザクションが特定のブロックチェーンブロックに含まれているかどうかをチェック（すなわち、検証）するために、所与のルートを有するマークルツリーを検索することを可能にする。

【００４５】

したがって、ＳＰＶウォレットは、他の形態のウォレットのようにブロックチェーンの完全なチェックを実行するのではなく、トランザクションが検証されたことを確認するだけでよいので（したがって、「簡易支払い検証」という名前である）、電話およびラップトップなどの、電力およびストレージが制約されたデバイスがビットコインエコシステム内で動作することができるという利点を少なくとも提供する。ＳＰＶウォレットは、トランザクションのいずれも含めることなくブロックヘッダのみをダウンロードするので、検証に必要とされるストレージ空間、エネルギー、および処理リソースを大幅に削減する。ＳＰＶウォレットは、以下に説明される理由のために、本開示の実施形態での使用に特に適しており、本明細書では、ＳＰＶチェックを含むために「検証」という用語を使用する。

【００４６】

<u>トランザクションのブロック</u>

図４は、ブロックチェーンブロックの一例を概略的に示す。各ブロックは、ブロックヘッダとトランザクションのセットとを含む。ブロックヘッダは、とりわけ、前のブロックヘッダのハッシュ、すなわち、現在のブロックが構築されたブロックのブロックヘッダのハッシュを含む。ブロックヘッダはまた、トランザクションのセットを使用して構築されたマークルツリーのマークルルートを含む。各トランザクションは最初にハッシュ（例えば、ダブルハッシュ）されて、そのトランザクションのトランザクション識別子（ＴｘＩＤ）が生成される。次いで、トランザクション識別子は、マークルツリーのリーフノードとして使用される。次いで、トランザクション識別子のペアを連結してハッシュし、マークルツリーの第１の内部レベルのそれぞれの内部ノードを形成する。次いで、第１の内部レベルの内部ノードのペアを連結してハッシュし、マークルツリーの第２の内部レベルのそれぞれの内部ノードを形成する。内部ノードのペアを連結してハッシュするプロセスは、単一のハッシュ、すなわちマークルルートのみが残るまで繰り返される。このマークルルートは、ブロックマークルルートと呼ばれることもある。

【００４７】

次に、特に図５、図６および図７を参照して、本開示の実施形態を説明する。

【００４８】

<u>ブロックのマークルツリーのセグメントの識別</u>

特定の当事者、例えばアリスがいくつかのトランザクションの妥当性確認を望むと仮定する。本開示の一実施形態によれば、トランザクションの少なくとも１つのサブセットが識別され、サブセットは、ブロックのマークルツリー全体のセグメントを形成し、かつ／またはブロックのマークルツリー全体のセグメントによって表される。したがって、トランザクションのブロックは、ブロックのマークルツリーに基づいて複数のセグメントに論理的にセグメント化することができ、各セグメントは、ブロックのトランザクションのサブセットを含み、各セグメントは、それ自体のルートノード（または「ルートハッシュ」）を有する。この共通ルートハッシュは、ブロック全体のルートハッシュと区別するために、以下では「セグメントハッシュ」と呼ばれることがある。ツリーセグメント内の同じレベル（すなわち、「リーフレベル」または「リーフ層」と呼ばれることもある最下位レベル）上のトランザクションは、兄弟である。所与のセグメント内のすべてのトランザクションは、そのセグメントの共通ルートノードを共有する。共通ルートノードは、マークルツリーの隣接するレベル、すなわち、最下位レベルのすぐ上のレベルに属し得る。代替的に、共通ルートノードは、上位レベルに属してもよい。一般に、共通ルートノードは、

最下位レベルとマークルルートとの間のマークルツリーの任意のレベルに属し得る。

【００４９】

マークルツリーに基づいてブロックをより小さい部分に分割することは、複数のバリデータにわたってトランザクションを迅速かつ効率的に割り当てる能力を含む、有意な技術的利点を提供する。例えば、ビットコインプロトコルはバイナリツリーを使用するので、複数のマシンにわたってバイナリ割り当てを実装することが可能である。セグメントのインデキシングシステムとして小さなバイナリマーカを使用することによって、マークルツリー全体における各セグメントの位置を迅速に計算することができ、妥当性確認が完了した後にセグメントを元の状態に戻すことが可能になり、ブロックの完全なマークルツリーが再構築される。このバイナリインデキシング手法については、以下でより詳細に説明する。

【００５０】

セグメントの識別には様々な技法を使用することができるが、１つの手法によれば、セグメントの数は、システム内の利用可能なバリデータの数によって決定され得る。例えば、４つのバリデータを有するシステムでは、マークルツリーを４つのセグメントに分割することができ、８つのバリデータがある場合には、マークルツリーを８つのセグメントに分割することができ、以下同様である。所与のマークルツリーのセグメントの識別は、図７のコントローラ７０２によって示される制御エンティティによって実行されるか、または影響を受けることができる。

【００５１】

上記で説明された点は、図５および図６を参照してさらに示され、図５は、どのようにマークルツリーがバリデータに割り当てられる別個の部分５０２に分割され得るかの例を示す。図５の例では、各矢印は、それぞれのトランザクション識別子を形成するためにハッシュされるそれぞれのトランザクションを表し、それぞれのトランザクション識別子は、マークルツリーのそれぞれのリーフノードにおいて使用される。マークルツリーのトップは、ブロックマークルルートである。この例では、マークルツリーによって表されるトランザクションのブロックは、３２個のトランザクションを含む。しかしながら、これは例示的な例にすぎず、一般に、マークルツリーは、ブロック内のトランザクションの数に応じて、任意の数のトランザクションを含み得ることが理解されよう。図示のように、マークルツリーは、破線のボックスによって示される４つの部分５０２ａ～ｄに分割される。各部分５０２は、実線の円によって示されるマークルツリーのそれぞれの共通内部ノード（内部ハッシュ）５０４によってリンクされる。各部分５０２は、８つのトランザクションを表す。この例では、共通内部ノード５０４は、マークルツリーの第４のレベルに属する。本明細書で説明される実施形態によれば、各それぞれの部分５０２（またはむしろ、一部を形成および／または表すトランザクション）は、処理のために、例えば、それぞれの部分５０２に属するトランザクションの妥当性確認のために、それぞれのバリデータに割り当てられる。

【００５２】

図６は、マークルツリーがどのように部分６０２に分割され得るかの別の例を示す。図６のマークルツリーは、図５のマークルツリーと同じである。ここで、この例では、マークルツリーは８つの部分６０２ａ～ｈに分割され、各部分６０２は４つのトランザクションを表す。この例では、共通内部ノード６０４は、マークルツリーの第３のレベルに属する。図５および図６のマークルツリーは、代わりに、より多くの（例えば１６個の）またはより少ない（例えば２つの）部分５０２、６０２に分割されてもよい。一般に、ブロックのトランザクションのセットから形成されるマークルツリーは、任意の数の部分５０２、６０２に分割され得、各部分は、最低２つのトランザクションを含む。

【００５３】

<u>それぞれの妥当性確認リソースへのセグメントの割り当て</u>

それらの識別に続いて、トランザクションのサブセットは、参照を容易にするために「バリデータ」とも呼ばれることがある複数の妥当性確認リソースにわたって分散される。

図７および図９では、複数のバリデータはリソースＡ～Ｄ（７０４ａ～７０４ｄ）として示されている。割り当てプロセスは、限定するものではないが、図９に示されるような構成要素９０４などの専用ユニットによって指示または影響され得る。

【００５４】

各バリデータ（７０４ａ～７０４ｄ）は、１つまたは複数の処理リソースを含むことができる。したがって、複数のバリデータ（７０４ａ～７０４ｄ）内のバリデータのうちの少なくとも１つは、１つまたは複数の仮想マシン、１つまたは複数のサーバ、１つまたは複数のＧＰＵベースのコンピューティングリソース、１つまたは複数のスレッド、および／または１つまたは複数のマルチプロセッサシステムなどのうちの少なくとも１つであり得るか、またはそれらを含み得る。本質的に、複数のバリデータのいずれも、ブロックのマークルツリーのセグメントによって互いに関連付けられた１つまたは複数のトランザクションを各々が妥当性確認することができる、任意のタイプ（複数可）または組合せの処理リソースから構成されることができる。複数のバリデータ（７０４ａ～７０４ｄ）および他のシステム構成要素は、集合的なリソースまたはエンティティ７００を形成し、これを「（分散型）妥当性確認ノード」と呼ぶ。

【００５５】

好ましくは、分散は、セグメントの各々を複数のバリデータ内のそれぞれのバリデータに割り当てることを含む。バリデータは、少なくとも以下を行うように構成され得る：

・　割り当てられたセグメント（複数可）を構成する１つまたは複数のトランザクションに対して動作すること、

・　１つまたは複数のトランザクションを妥当性確認して、それらがブロックチェーンプロトコルに準拠することを検証すること、および／または

・　ブロックチェーン台帳、または既知の、登録された、もしくは使用されたトランザクションのデータベースなどの既存のリポジトリ内でそれらが識別可能であることを妥当性確認すること。

【００５６】

バリデータのアクティビティ、および異なるバリデータへのサブセットの割り当ては、コントローラによって指示され得る。図７は、コントローラ７０２が、それぞれのツリーセグメントに対するトランザクションＡ～Ｄのサブセットをそれぞれバリデータ７０４ａ～７０４ｄに割り当てる様子を示す。システムレベルコントローラ７０２は、分散型妥当性確認ノード内のシステムまたはデバイス７０４ａ～７０４ｄのアクティビティを調整し、ブロックのマークルツリーによるツリーセグメントの識別、識別されたセグメントのそれぞれのバリデータへの割り当て、妥当性確認されたツリーセグメントの、ブロックの完全なマークルツリーへの並べ替え（reordering）、および／または再構築されたブロック内のトランザクションの順序付けなどのタスクを制御するか、またはそれに影響を及ぼし得る。

【００５７】

バリデータのうちの１つまたは複数は、バリデータレベルでコントローラとして働くように構成された少なくとも１つの調整エンティティを含み得る。したがって、バリデータ７０４ａ～７０４ｄのいずれかまたはすべては、それ自体の少なくとも１つのコントローラ構成要素を含み得る。この下位レベルコントローラは、バリデータ内の１つまたは複数の処理リソースへのタスクまたはサブタスクの割り当て、所与のセグメントのためのマークルツリーの再構築、または他のシステム構成要素、例えば他のバリデータもしくは上位レベルコントローラ、ＵＴＸＯプール、ウォレットなどとの対話などの動作に影響を及ぼすかまたはそれを指示し得る。次に、処理リソース自体は、より小さいシステムにさらに分解されてもよく、そのうちの１つまたは複数は、コントローラと、それ自体の１つまたは複数の処理リソースとを含んでもよい。このようにして、システムは、妥当性確認タスクを実行するための１つまたは複数の処理リソースと、プロセッサのアクティビティおよび構成要素間通信の実行の調整のための１つまたは複数のコントローラとを含む妥当性確認エンティティによってセグメント妥当性確認が実行される階層アーキテクチャを含み得

る。
【００５８】
バリデータが複数の処理リソースを含む実施形態では、バリデータは、その割り当てられたセグメントをより小さいセグメントに分割し得る。次いで、バリデータのコントローラは、その制御下にあるプロセッサにわたってサブセグメントを分散させることができる。このようにして、妥当性確認プロセスを階層的かつ分散的に実施することができる。
【００５９】
この階層的分解は、トランザクションレベルに拡張することもでき、その結果、妥当性確認を、ツリーセグメントレベルではなく、トランザクションごとのサブプロセスまたはタスクにさらに分解することができる。この手法では、個々のトランザクション（複数可）の妥当性確認は、異なるマシン、または同じもしくは異なるマシン上で実行される異なるスレッドにわたって分散されるサブタスクに分解される。これらのプロセスは、スレッドが使用可能になると、別のトランザクションまたはタスクがそれに割り当てられるようにキューに入れることができる。
【００６０】
したがって、本開示は、多くのトランザクションが同時に処理されることを可能にし、唯一の制限は、従来の技法のように利用可能な処理速度の量がボトルネックになるのではなく、分散型妥当性確認ノードを形成するために利用可能なハードウェアの量である。これにより、ブロックチェーン処理システムは、ブロックチェーンネットワークの基礎となるプロトコルを変更する必要なく、水平にスケーリングすることができる。
【００６１】
したがって、本開示は、図１および図２を参照して、「本開示の例示的な実施形態を実施するための例示的な技術環境」と題する以下のセクションでより詳細に説明される妥当性確認に対する従来の手法からの著しい逸脱を表す。説明したように、従来の手法は、１つのブロックがエンティティ全体として妥当性確認されることと、妥当性確認ノード（図１の１０４を参照）が単一のコンピューティングユニットであるといの従来の見方とを含む。対照的に、本開示の実施形態は、異なるバリデータ（図７および図９の７０４ａ～７０４ｄ）に与えられる複数のセグメントへとマークルツリーを分割し、セグメントおよびバリデータの各々は、関与する分散の程度を高めるためにさらに分解されることが可能である。
【００６２】
さらに、各ブロックをそのマークルツリーに基づいてセグメントに分割することによって、本開示の実施形態は、バリデータが、ブロック全体ではなくブロックの小さい部分にアクセスし、ダウンロードし、処理することを可能にする。各セグメント内のトランザクションが（ペアで）単一のルート値にハッシュアップすることを想起されたい。これは、ブロック全体が完全にダウンロードされ、記憶され、処理されるのではなく、必要な関連トランザクションのみを使用してセグメントを妥当性確認することができることを意味する。ビットコインＳＶなどのプロトコルは、ブロックサイズをスケーリングすることおよびより大きいブロックを台帳に含めることを可能にするので、ブロック全体をダウンロードする従来のモデルはボトルネックになる。本開示の実施形態は、個々のバリデータが、それらに関連する（より小さい）部分のみを受信および処理することを可能にすることによって、ブロックチェーンスケーラビリティに対するこの課題を克服する。これの結果、全体的な妥当性確認時間が短縮され、ブロックチェーンネットワークが改善し、ブロックチェーン上で実行されるアプリケーションが改善する。
【００６３】
さらに、実施形態は、ＳＰＶプロセスおよびリソースの使用をサポートし、容易にする。なぜなら、そのようなＳＰＶは、所与の当事者にとって関心のあるマークルツリーの部分のみのローカル妥当性確認を含むからである。したがって、ＳＰＶ技術のツリープルーニングの性質は、本開示の実施形態と併用するのに理想的に適している。ＳＰＶの文脈では、バリデータには、それらが必要とするブロックデータの部分、すなわち、ブロックへ

ッダまたはセグメントルートノードおよび関連トランザクションのみが提供され得る。
【００６４】
各バリデータがそのチェックを実行し、それが処理したセグメントの有効性を確認した場合、ツリーを生成するために使用されるハッシュメカニズムにより、ブロックが有効であることを保証することができる。
【００６５】
複数のバリデータにわたるロードバランシング
効率を向上させるために複数のリソースにわたってタスクを均等に分散させることを目的として構成されたロードバランシング技法およびシステムが当技術分野で知られている。その目的は、一部の処理リソースがアイドル状態にある一方で他の処理リソースが過負荷になるリスク、ひいては、性能の劣化さらには故障のリスクを最小限に抑える。したがって、ロードバランシングは、システム全体の回復力ならびにその性能および効率を確実にする際に重要になる。本開示の実施形態は、例えば、静的または動的ロードバランシングなどの任意の既知のロードバランシング技法を利用し得る。追加的または代替的に、本明細書に開示されるロードバランシング手法を有利に使用してもよい。
【００６６】
ツリーセグメントをバリデータに割り当てる必要がある場合、そのダブルハッシュの最初の４桁（すなわち、ツリーセグメントのセグメントハッシュ）を使用して、どのバリデータがそのセグメントを処理するかを決定することができる。マークルルートは、ブロックからのトランザクションＩＤ（ＴｘＩＤ）のペアを一緒にハッシュしてマークルツリーのそれぞれの内部ノード（または内部ハッシュ）を生成し、次いで、単一のハッシュに最終的に到達するまで、隣接する内部ハッシュを繰り返しハッシュすることによって生成されることを想起されたい。このダブルハッシュされたマークルルートは、効率的で迅速かつセキュアな検証メカニズムを提供する。これはまた、本文脈において、ダブルハッシュがランダムな２進数を生成するという利点を提供する。各セグメントハッシュを含む各内部ハッシュは、それ自体がダブルハッシュである。
【００６７】
ツリーセグメントをバリデータに割り当てる必要がある場合、そのダブルハッシュの最初の４桁（すなわち、ツリーセグメントのセグメントハッシュ）を使用して、どのバリデータがそのセグメントを処理するかを決定することができる。マークルルートは、ブロックからのトランザクションＩＤ（ＴｘＩＤ）のペアを一緒にハッシュしてマークルツリーのそれぞれの内部ノード（または内部ハッシュ）を生成し、次いで、単一のハッシュに最終的に到達するまで、隣接する内部ハッシュを繰り返しハッシュすることによって生成されることを想起されたい。このダブルハッシュされたマークルルートは、効率的で迅速かつセキュアな検証メカニズムを提供する。これはまた、本文脈において、ダブルハッシュがランダムな２進数を生成するという利点を提供する。各セグメントハッシュを含む各内部ハッシュは、それ自体がダブルハッシュである。したがって、セグメントハッシュの先頭数字の最初のｘ個を割り当てインデックスとすることができる。４つの先行ゼロを有するハッシュの場合、ＩＤ００００を有するバリデータにツリーセグメントを割り当てることとなり、先頭数字が０００１であるハッシュの場合、ＩＤ０００１を有するバリデータに割り当てることとなり、以下同様である。ダブルハッシュのランダムな生成は、バリデータへのツリーセグメントのランダムな分散を確実にする。
【００６８】
ダブルハッシュは、典型的には、マークルツリーを生成する際に使用されるが、すべての例において必須というわけではなく、代わりに、単一ハッシュのみが使用されてもよい。実際、ハッシュ演算を何回行ってもランダムな２進数が得られる。ロードバランシングタスクは、図９に９０５として示される専用システム構成要素によって行われてもよいし、システム７００内の他の場所に、もしくはシステム７００と関連および通信して提供されてもよい。
【００６９】

ブロックの分散ダウンロード

いくつかの実施形態によれば、異なるバリデータへのブロックマークルツリーのセグメントの割り当てを使用して、トランザクションのブロックの一部または全部をダウンロードするためのより高速でより効率的なプロセスを提供することができる。

【００７０】

各バリデータには、例えば、上記で説明された割り当てインデックスに基づいて、マークルツリーのセグメントが割り当てられる。次いで、任意の所与のバリデータは、割り当てられたツリーセグメントを形成するトランザクションのセットをダウンロードするように動作する。これは、トランザクションのセットをブロックチェーン自体から（例えば、ブロックチェーンノードから）ダウンロードすること、または第三者サービスプロバイダなどの異なるリソースもしくはエンティティからダウンロードすることを含み得る。トランザクションのセットは、バリデータの内部メモリに、またはクラウド内の共有ドライブなどの共有記憶場所にダウンロードされ得る。

【００７１】

分散ノードは、完全なブロック、すなわちブロックを形成するトランザクションのセット全体を必要とし得る。その場合、ツリーセグメントを割り当てられた各バリデータは、そのセグメントを形成するトランザクションのサブセットをダウンロードする。他のシナリオでは、分散ノードは、ブロックの特定の部分のみを必要としてもよい。その場合、所望のトランザクションを取得するために、バリデータのうちの一部のみが、トランザクションのそれぞれのサブセットをダウンロードする必要があり得る。

【００７２】

このようにしてブロック（またはブロックの一部）をダウンロードすると、各バリデータがブロックを形成するトランザクションのセット全体のうちのトランザクションのサブセットのみを処理するので、全体的なダウンロードは速くなる。これは、所与のエンティティ（例えば、フルノード）が、例えば、各トランザクションを、それがブロックに現れる順序でダウンロードすることによって、ブロック全体をダウンロードしなければならない従来のブロックダウンロードとは対照的である。ここで、ブロックは、複数のバリデータによって並列にダウンロードされる。ブロックは、さらに数桁増えないまでも、数万のトランザクションを含み得る。この数のトランザクションをダウンロードする単一のエンティティは、かなりのリソースを消費し、かなりの時間を要する。計算負荷はバリデータの間で分散され、各個々のバリデータが処理リソースの一部を消費するように、される。同様に、ブロックをダウンロードするための全体的な時間が短縮される。

【００７３】

上述したように、各バリデータはトランザクションのサブセットをダウンロードし得る。次いで、サブセットを組み合わせて、単一の記憶位置にブロックを再構築することができる。（「単一の記憶位置」とは、自己完結型のエンティティである記憶リソース、または集合的なエンティティを形成する複数の関連する記憶リソースのいずれかを意味する）。そうするために、個々のバリデータは、トランザクションを正しい順序で配置するように構成された分散ノードの中央コントローラにそれぞれのサブセットを送信し得る。セグメントハッシュ（すなわち、ツリーセグメントをリンクするハッシュ）は、この目的のために利用されてもよい。例えば、セグメントハッシュの、マークルツリー内のその位置へのマッピング、例えば、セグメントハッシュがマークルツリー内に現れるにつれて左から右へのマッピングが維持され得る。次いで、トランザクションのサブセットは、対応するセグメントハッシュに基づいて順番に（例えば、最初から最後まで）配置され得る。

【００７４】

いくつかの実施形態では、個々のバリデータ（または全体として分散ノード）は、マークルツリーを再構築することによって、正しいトランザクションがダウンロードされたこと（またはトランザクションが正しくダウンロードされたこと）を確認し得る。トランザクションのサブセットをダウンロードした後、バリデータは、これらのトランザクションに基づいて候補セグメントハッシュを生成し得る。候補セグメントハッシュは、ＴｘＩＤ

のペアをハッシュしてそれぞれの内部ハッシュを生成し、候補セグメントハッシュが生成されるまで内部ハッシュのペアを繰り返しハッシュすることによって構築される。候補セグメントハッシュが属するマークルツリーのレベルは、マークルツリーが分割されるツリーセグメントの数に依存する。バリデータは、候補セグメントハッシュがマークルツリーのハッシュであることを検証し得る。ハッシュが一致しない場合、ダウンロード中にエラーが発生している。いくつかの例では、各バリデータは、候補セグメントハッシュを生成し、検証を実行するためにそれをコントローラに送信し得る。別の例として、候補ブロックマークルルートは、ダウンロードされたトランザクションのセット全体に基づいて生成されてもよい。この場合も、候補マークルルートは、ブロックが正しくダウンロードされていれば、実際のブロックのマークルルート（すなわち、ブロックに記憶されたマークルルート）と一致するはずである。

【００７５】

場合によっては、バリデータは、上記で説明した技法を使用して、ダウンロードされたトランザクションを妥当性確認し得る。すなわち、各バリデータは、ツリーセグメントを割り当てられ、トランザクションの対応するサブセットをダウンロードし、それらのトランザクションを妥当性確認する。他の場合、バリデータは、必ずしもトランザクションを妥当性確認しなくてもよく、後の使用、例えば、第三者に送信するために、トランザクションを単にダウンロードしてもよい。

【００７６】

<u>分散型ＵＴＸＯプール</u>

好ましくは、分散型妥当性確認ノードの一部を形成する各バリデータ７０４は、未使用トランザクション出力（ＵＴＸＯ）を生成、記憶、および／または維持するためのそれ自体のリポジトリ（プール）を有する。これは、消費されていない、すなわち、ブロックチェーントランザクションに関連付けられた未使用出力の記録を提供するＵＴＸＯプールとして機能する。したがって、各バリデータのＵＴＸＯプールは、マークルツリーセグメントに関してコントローラによってそれに割り当てられるトランザクションに基づき、それから構築される。一実施形態では、これは、処理のために所与のバリデータに割り当てられたトランザクションの未使用ＵＴＸＯに関するデータを含む（グラフ）データベースであってもよい。データベース内の記録は、新しいマークルツリーセグメントがそれに割り当てられるときにバリデータが気付く各ＵＴＸＯについて作成される。したがって、分散型妥当性確認ノードの観点から、ＵＴＸＯプールは、単一のプールではなく、複数の異なるＵＴＸＯプールから構成され、各ＵＴＸＯプールは、異なるバリデータにまたは異なるバリデータ上に提供され、ＵＴＸＯの異なるセットを含む。したがって、ノードのためのＵＴＸＯプールは、データと、それを記憶および／または処理するリソースとの両方に関して分散される。

【００７７】

これは、ネットワーク内の各フルノードが、ブロックチェーン上のすべてのＵＴＸＯを追跡するＵＴＸＯプールのコピーを有する従来のＵＴＸＯモデルから大きく乖離している。対照的に、本開示は、ブロックチェーンのＵＴＸＯセット全体のサブセットであるＵＴＸＯプールを各々が有する複数の妥当性確認リソースにわたってＵＴＸＯプールを分散する。各バリデータのＵＴＸＯプールは、妥当性確認を課されたマークルツリーのサブ部分を構成するトランザクションのＵＴＸＯを含む。

【００７８】

そのような手法によれば、新しいブロックが妥当性確認される必要があるたびに、データベースに関するすべてのコマンド、イベント、および項目がログに記録されるという点で、ＳＱＬトランザクションログと同様の方法で実施されることができる。「データベースログ」という用語は、本明細書では、ブロックチェーンとの関連で知られている「トランザクション」という用語の使用から生じる混乱を回避するために使用されるが、「トランザクションジャーナル」、「トランザクションログ」などの用語を含むために「データベースログ」という用語を使用する。本質的に、データベースログは、データベース管理

システムによって実行されたアクションの履歴として解釈することができ、コンピュータベースのデータベースの分野で知られているように、データベースの状態に関して発生したすべての変更の記録を提供する（https://en.wikipedia.org/wiki/Transaction_log参照）。

【００７９】

順序付けられた履歴データベースログの使用は、ログの履歴をその元の順序で実行することによってＵＴＸＯプール全体が構築可能であることを意味する。有利なことに、これは、必要なときにデータベースのコピーを常に（再）生成することができ、データの別個のコピーを記憶する必要がないことを確実にする。データ完全性が確保され、必要な記憶リソースが少なくなる。各ＵＴＸＯプールは、別々に記憶され、維持され、処理されることができる。また、有利なことに、ＳＰＶ技法は、ＳＰＶ技法がマークルツリーのプルーニングされた部分に対して動作すると仮定すると、各バリデータに対する別個のＵＴＸＯデータベースの作成を容易にする。

【００８０】

データベース内のトランザクション（ＴＸ）は、様々な方法で構造化することができるが、特に有利な手法は、ブロックＩＤとトランザクションＩＤとの連結（block_ID｜｜ＴｘＩＤ）を含む識別子にしたがってトランザクションを構造化することである。ブロックＩＤおよびトランザクションＩＤは両方とも２５６ビットのハッシュであり、その結果、セキュアで衝突のない５１２ビットの連結フィールド構造が得られる。

【００８１】

このようにトランザクションを構造化することで、高速で効率的なルックアップメカニズムが提供される。同じblock_IDを有するすべてのトランザクションがデータベース内に一緒に位置するように、block_IDによってトランザクションをソートすることができる。したがって、（例えば、トランザクションのＵＴＸＯが使用されたかどうかをチェックするために）バリデータがトランザクションを必要とするとき、バリデータは、まず対応するblock_IDを検索し、次いで対応するＴｘＩＤを検索することによって、データベース内のトランザクションを位置特定することができる。これは、検索がデータベースの関連セクションに限定されるという効果を有する。この効率により、検索動作に必要な時間、処理リソースおよびエネルギーが削減され、従来技術に対して大幅な改善をもたらす。

【００８２】

フラグまたはマーカは、バリデータのプール内の各ＵＴＸＯに関連付けられ、ＵＴＸＯがロックされているかまたはロック解除されているかを示す。便宜上、このフラグまたはマーカを「ロックフラグ」と呼ぶことがある。ＵＴＸＯが「ロック（locked）」とマークされるとき、これは、このＵＴＸＯが使用不可能であることをグループ内の（すなわち、分散型妥当性確認ノード内の他の場所の）バリデータに示すインジケータとして機能する。逆に、ＵＴＸＯが「ロック解除（unlocked）」とマークされるとき、これは、ＵＴＸＯが使用可能であることをバリデータに示すインジケータとして機能する。したがって、それは、ＵＴＸＯを使用するトランザクションを検証するために割り当てられたバリデータが、トランザクションが有効であることを証明すると仮定して、それが償還済みであり、したがって使用可能ではなくなっていることをそのピアにシグナリングすることを可能にする方法として機能する。「ロック」状態は、使用が許可されることを意味し、「ロック解除」状態は、使用が禁止されることを意味する。

【００８３】

このロック／ロック解除フラグは、「ロック」に対しては０、ロック解除に対しては「１」など、単純な小さいバイナリマーカとすることができる。マーカメカニズムは、分散ノードシステム内のバリデータによって内部的に使用され、マーカは、トランザクションがプロトコル規則に準拠するように、ブロックチェーンと対話する前にトランザクションから除去される。

【００８４】

使用時には、バリデータは、コントローラによってそれに割り当てられた新しいトランザクションのそれぞれにおける出力を検査する。任意の未使用出力（ＵＴＸＯ）は、バリデータのＵＴＸＯプールに追加され、すなわち、ＵＴＸＯデータベースのエントリとして記録される。各新しいＵＴＸＯのための関連するデータベース記録では、ロックフラグは「ロック解除」に設定される。

【００８５】

ＵＴＸＯが新たに割り当てられたトランザクションによって使用されていることをバリデータが確認すると、バリデータは、複数のバリデータのうちの他のすべてのバリデータにメッセージを送信して、このＵＴＸＯもそれぞれのプールにロックされるべきであることを通知する。本質的に、バリデータは、特定の時間に特定のハッシュＩＤを有するトランザクションを含む使用を確認したことを示す通信をそのピアに送信する。トランザクションハッシュおよびそれが使用するＵＴＸＯのリストは、当該トランザクションを識別し、それ自身のデータベースにロックされているとマークするのに十分であるので、他のバリデータは、トランザクション全体に対する完全なデータを受信する必要はない。メッセージを受信すると、各受信バリデータは、当該ＵＴＸＯがそれらのＵＴＸＯプール内にあるかどうかをチェックする。もしそうであれば、ロックフラグの状態は「ロック」に変更される。したがって、ロックは、バリデータが、後続のトランザクションにおいて同じＵＴＸＯが使用されることを許可するのを防ぐ。新しいトランザクションが同じＵＴＸＯを使用しようとした場合、ロックフラグのチェックは、第２の使用の試みが無視されるべきであることを示す。メッセージを送信したバリデータが、妥当性確認が失敗に終わっており、したがってＵＴＸＯが使用されていないと決定した場合、ＵＴＸＯのロックフラグが「ロック解除」状態に変更されるべきであることを示すさらなるメッセージが、この趣旨でバリデータピアに送出され得る。有効な使用が完了すると、この趣旨でメッセージを送信することができ、ロックされたＵＴＸＯを関連するＵＴＸＯプールから削除することができる。

【００８６】

上記で説明した実施形態では、各バリデータは、それに割り当てられたツリーセグメントのすべてにおけるすべてのトランザクションのＵＴＸＯを含む単一のＵＴＸＯプールを有する。しかしながら、代替的な手法では、各バリデータによって維持されるＵＴＸＯプールは、ブロック毎に１つずつ、複数のサブプールに分割（divided）／分割（split）／区画化／形成されてもよい。このようにして、単一のＵＴＸＯプールを論理階層に編成することができる。さらに別の手法では、１つまたは複数のバリデータは、それぞれの複数のＵＴＸＯプールに関連付けて構成されてもよく、各複数のＵＴＸＯプールは、１つまたは複数のツリーセグメントのセットについてのＵＴＸＯに関連する。したがって、いくつかの実施形態では、バリデータ（複数可）は、ＵＴＸＯを、異なる個々のツリーセグメントのための別個のＵＴＸＯプールに、またはツリーセグメントのタイプもしくは所与の範囲内に入るツリーセグメントなどの何らかの予め定義された基準にしたがって編成し得る。そのような実施形態では、識別子は、検索を関連するＵＴＸＯプールに絞り込むために使用することができるブロックＩＤを含み得、その後、検索がそのプール内で進行して、そのＴｘＩＤにより関連トランザクションを識別する（ことを試みる）ことができる。当業者であれば、いくつかの実施形態において、これらの手法の混合が使用され得ること、すなわち、分散ノード内の１つまたは複数のバリデータが単一のＵＴＸＯプール手法を採用し得る一方で、他のもの（複数可）は、複数の別個のＵＴＸＯプール、および／またはサブプールに編成されるＵＴＸＯプール、またはそれらの任意の組合せを使用するよう構成されることを理解するであろう。

【００８７】

これは、当事者が同じＵＴＸＯを２回使用しようとする「二重使用（double spend）」状況に対する保護を提供する。これは、システム内のバリデータの数または場所にかかわらず、効率的かつ迅速に動作し、ブロックチェーンを介して実装される転送のセキュリティおよび完全性を保存する、単純かつセキュアなロックメカニズムを提供する。

【００８８】

可能な実施形態の例示的なシステム

図７および図９は、説明される実施形態のうちの少なくともいくつかを実装するための例示的なシステム７００を示す。図８は、本開示の方法（の大まかな概要（high-level view））において取られ得る例示的なステップのフローチャートを示す。

【００８９】

システム７００は、それが組織に関連付けられ、より大きな専有システムの一部を形成するという意味で、クローズドシステムであってもよい。そのような場合、そのデータ、例えば、トランザクションは、組織のより広いシステム内の他の構成要素から受信され得、その結果および出力は、内部の宛先に送信され得る。追加的または代替的に、システム７００は、様々なエンティティとインターフェースするように構成されてもよく、その一部または全部は、組織の外部に位置してもよい。そのような場合、システム７００は、サービスとして妥当性確認機能を提供するように構成され得る。例えば、システム７００は、ブロックチェーンネットワークと対話して、それが必要とするデータを取得するように構成され得る。追加的または代替的に、それは、その妥当性確認サービスを使用することを望むエンティティと対話することができる。したがって、システム７００のアクティビティは、特定の組織またはエンティティに関して単に内部的であるか、または他の当事者に妥当性確認サービスを提供するために外部エンティティとの対話に対してオープンであるか、またはその２つの組合せである可能性がある。他の内部または外部エンティティ間の通信は、図９に９０２として示される、１つまたは複数のインターフェースまたは通信構成要素によって協調され得る。

【００９０】

図７および図９に示すように、システム７００は、制御エンティティ７０２（または単に「コントローラ」）と、本明細書では単に「バリデータ」とも呼ばれる複数の妥当性確認リソース７０４とを含む。４つのバリデータ７０１ａ～ｄのみが図７に示されているが、一般に、システム７００は、任意の数のバリデータを含み得る。さらに、コントローラ７０２は、バリデータ７０４とは異なるものとして図７および図９に示されているが、コントローラ７０２がバリデータ７０４のうちの１つを含むか、またはそれに含まれ得ることを排除するものではない。上記で説明したように、各バリデータは、１つまたは複数の処理リソースを含み得、それ自体の内部アクティビティの調整のためのそれ自体のコントローラを含み得る。このようにして実装することができる階層レベルには技術的または論理的な制限はない。しかしながら、図７は、簡潔さおよび理解の容易さのために、そのような階層の１つのレベル（最上位）のみを示す。

【００９１】

図７に示すように、コントローラ７０２は、トランザクションのセットを取得する。トランザクションは、送信リソースから電子チャネルまたはネットワークを介して受信することができる。送信者は、上で説明したように、システムの組織の内部または外部の何らかの種類の妥当性確認チェックを実行することを望む任意のエンティティであり得る。例えば、これは、図１のノード１０４などのブロックチェーンネットワーク上のフルノード、またはデジタルウォレット、または当事者間で行われるブロックチェーン実装転送に関するローカルチェックを実行することを望むマーチャント／ＳＰＶノードとすることができる。インターフェース（複数可）９０２は、システム７００とシステムの外部のソースとの間のデータの送信を容易にし得る。

【００９２】

トランザクションは、トランザクションのブロックを形成するか、または形成し得る。トランザクションは、単一のリソースから（例えば、ブロックチェーンのブロックから）または異なるリソース（例えば、１人以上のユーザ、１つまたは複数のブロックチェーンノードなど）から取得され得る。トランザクションは、それらがブロックチェーン上で発行される前に、すなわちブロックに記録される前に取得され得る。代替的に、トランザクションは、ブロックチェーンに記録された後に取得されてもよい。

【００９３】
コントローラ７０２は、本明細書で説明されるように、トランザクションのそれぞれのサブセットを各バリデータ７０４に割り当てる。トランザクションの各サブセットは、トランザクションのフルセットに基づいて生成されたマークルツリーのそれぞれの部分の少なくとも一部を形成し、マークルツリーのそれぞれの共通内部ノードによってリンクされる。図７の例では、トランザクションサブセットＡはバリデータＡに割り当てられ、トランザクションサブセットＢはバリデータＢに割り当てられ、トランザクションサブセットＣはバリデータＣに割り当てられ、トランザクションサブセットＤはバリデータＤに割り当てられる。トランザクションのサブセットが割り当てられると、バリデータ７０４は、それぞれのサブセットを処理する。いくつかの実施形態では、これは、各バリデータ７０４がトランザクションのそれぞれのサブセットを妥当性確認することを含む。そうするために、コントローラ７０２は、関連トランザクションをそれぞれのバリデータ７０４に送信し得る。バリデータ７０４は、トランザクションのそれぞれのサブセットの各々が有効であること、または少なくとも１つのトランザクションが有効でないことを示すために、コントローラ７０４に返信してもよい。

【００９４】
バリデータ７０４ａ～７０４ｄのうちの少なくとも１つ、好ましくはいくつかまたはすべては、図９に９０１ａ～９０１ｄとして示されるそれら自体のＵＴＸＯプールへのアクセスを有する。このプールは、上記で説明したデータベースのようなストレージ設備を含み、潜在的に、ブロックＩＤとトランザクションＩＤとの連結を含む有利なインデキシング構造を有する。図９では、プールは、それぞれのバリデータ内に含まれるものとして示されているが、当業者であれば、プールが、同様に／代替的に、バリデータの外部にあり、それと通信状態にあるものとして提供されてもよいことを容易に理解するであろう。

【００９５】
１つまたは複数の実施形態では、開示されるプロセスは、ブロックレベル妥当性確認段階を含み得、その間に、入ってくる新しいブロックがブロックレベル基準に照らしてテストされる。例示的なブロックレベル基準は、上記で説明されており、一般に、ブロック内のトランザクションとは対照的に、ブロック自体に適用可能な所定のフォーマッティング要件および特性または制限に関する。例には、ブロックサイズ、ブロックヘッダ構造またはコンテンツ、および同様の基準が含まれる。そのような動作は、コントローラ、またはコントローラの構成要素、または別のシステム構成要素によって行われ得る。

【００９６】
いくつかの実施形態では、方法は、新しいブロック内のトランザクションへの入力の各々、すなわち各ＵＴＸＯが一意であるかどうかを評価するように動作するＵＴＸＯ一意性確認モジュールをさらに含み得る。同じＵＴＸＯが新しいブロックにおける入力として２回以上現れる場合、それは、潜在的な二重使用問題を示し、ＵＴＸＯ一意性基準に違反する。ＵＴＸＯ一意性確認モジュールが、新たなブロックにおけるトランザクション入力の間で２回以上参照されるＵＴＸＯを識別すると、そのブロックが拒否されるべきであることを示すためにエラー信号または他の割り込みを出力し得る。

【００９７】
新しいブロックが拒否されない、すなわち、すべてのＵＴＸＯ入力が一意であると仮定した場合、マークルツリーセグメントが識別され、それらの関連トランザクションがバリデータのセットの間で割り当てられ得る。識別プロセスは、図９に示されるセグメント識別ユニット９０３などの構成要素によって実行され得る。割り当てプロセスは、図９において９０４として示されるセグメント割り当てユニットによって実行され得る。割り当てユニット９０４は、個々のバリデータの間でブロックセグメントを分散させるためのいくつかの可能な割り当て方式のいずれか１つを採用し得るが、有利な手法では、割り当て方式は、上記で説明したようにロードバランシングを目的とすることができる。割り当てユニット９０４は、ロードバランシングユニット９０５を含む（またはそれと通信している）ことができる。これは、図９においてシステムの別個の関連する構成要素として示され

ているが、他の実施形態では、ロードバランシングユニットは、割り当てユニット９０４の一部であってもよいし、コントローラ７０２に対して別個であってもよい。構成要素の任意の組合せは容易に採用され得る。

【００９８】

個々のバリデータは、それらが受け取ったセグメント（複数可）に関連付けられたトランザクションを、トランザクションレベル妥当性確認基準に照らして妥当性確認する。バリデータは、割り当てられたトランザクションが有効であることを検証する際にそれぞれ独立して動作するので、バリデータ間の同期パラダイムを必要としない。各バリデータは、その割り当てられたトランザクションの有効性を確認する結果を出力する。結果は、セグメント内のすべてのトランザクションが有効であることを確認するために追加または累積される。バリデータのうちの１つが非準拠トランザクション、すなわち無効なトランザクションを識別した場合、バリデータは、無効なトランザクションが存在することを示すために、割り込みまたは他の信号などの出力を発行し得る。その割り込みまたは信号は、他のバリデータに、またはコントローラもしくは別のシステム構成要素に送信され得、それらは、それぞれのトランザクションのテストを直ちに中止し、拒否されるべきブロック内のトランザクションの妥当性確認にこれ以上リソースを浪費しないことができる。

【００９９】

いくつかの例では、システムは、ブロックレベル基準をチェックするように構成され得る。これは、バリデータへのセグメントの割り当ての前に実行され得るが、ブロックレベル妥当性確認段階は、バリデータによるトランザクションレベル妥当性確認試験の後に行われてもよいし、いくつかの事例では、トランザクションレベル妥当性確認試験と並行して行われてもよいことが理解されよう。

【０１００】

ここで、ブロックを妥当性確認する方法の一例をフローチャート形式で示す図８を参照する。ブロックは、複数のトランザクションを含み、各トランザクションは１つまたは複数の入力を参照し、各入力はＵＸＴＯである（コインベース生成トランザクションの場合を除く）。方法は、ブロックチェーンネットワーク上のノード内の適切なハードウェアおよびプロセッサ実行可能命令を使用して実装される。

【０１０１】

動作中、分散型妥当性確認ノード７００は、ステップＳ８０１において新しいブロックデータを受信する。これは、ブロック全体であってもよく、またはＳＰＶ関連妥当性確認の場合には、ＳＰＶチェックを実行するのに必要な部分的なデータのみを含んでもよい。便宜上、このデータを「ブロック」と呼ぶ。妥当性確認されるべき新しいブロックは、新しいブロックを生成してプルーフオブワークを完了したブロックチェーンネットワーク上のマイニングノードから受信されてもよいし、（ＳＰＶ）チェックを実行することを望むマーチャントノードから受信されてもよいし、ＳＰＶウォレットなどのウォレットから受信されてもよい。新しいブロックは、ネットワーク内の別の（非マイニング）ノードから受信され得る。いくつかの例では、分散型妥当性確認ノード７００は、ブロックをネットワーク内の任意の他のノードに転送する前に、ブロックを妥当性確認する。上述したように、新しいブロックの妥当性確認は、ブロックが特定のプロトコルベースの基準、および／または所与の実装内で指定され、必要とされ得る他の基準を満たすことを確認することを含み得る。

【０１０２】

ステップＳ８０２において、システム７００は、ブロックのマークルツリーのチャンクを識別する。Ｓ８０３において、セグメントは複数のバリデータに分散され、Ｓ８０４において、バリデータは、トランザクションのそれぞれのサブセットを実質的に並行して互いに独立して処理する。Ｓ８０５において、バリデータは、妥当性確認が成功したか失敗したかをコントローラにシグナリングする。

【０１０３】

「プロセッサ」という用語は、本明細書で並列プロセッサの説明に関連して使用される

とき、必ずしも物理的に別個のマイクロプロセッサを意味するとは限らず、プロセッサ機能を独立して並列に実行することができる並列処理リソースを可能にする任意のハードウェアまたはソフトウェア実装形態を含み得ることに留意されたい。並列プロセッサは、複数のコアを有する１つのプロセッサを含み得る。いくつかの事例では、並列プロセッサは、複数の別個の処理ユニットを含み得る。並列プロセッサは、物理メモリを共有してもしなくてもよい。各並列プロセッサは、どのように実装されても、無効なトランザクションを識別することに応答して信号を出力するような、シグナリングのためのソフトウェアまたはハードウェアメカニズムを有する。並列プロセッサの実装はまた、ソフトウェアおよび／またはハードウェアにおいて、割り当てられたトランザクションデータをローカル処理のためにそれぞれのプロセッサにルーティングするために必要なデータ転送メカニズムを提供することを含む。

【０１０４】

付記（clause）の列挙

本開示の実施形態は、例示の目的で、限定することなく、以下の列挙された付記において提供される。

【０１０５】

本開示の列挙された付記または態様の１つのセットに関して以下で言及される特徴は、そのような点において限定されることが意図されるものではなく、付記の１つのセットに関して言及される任意の特徴（複数可）は、付記の他のセットのうちの１つまたは複数に組み込まれ得る。

【０１０６】

付記セット１：

＜付記１．１＞複数のブロックチェーントランザクションとブロックのためのマークルツリーのルートとを含むブロックチェーンブロックの少なくとも一部を処理する（例えば妥当性確認する）コンピュータ実装方法。

本方法は、以下を含む：

ブロックチェーントランザクションのそれぞれのサブセットを複数の処理リソース（例えば妥当性確認リソース）に割り当てるステップであって、各それぞれのサブセットは、マークルツリーのそれぞれの部分を提供し、マークルツリーのそれぞれの内部ノードによって表される、ステップ、および

複数の処理（例えば、妥当性確認）リソースを使用して、ブロックチェーントランザクションのそれぞれのサブセットを処理（例えば、妥当性確認）するステップ。

【０１０７】

「バリデータ」という用語は、「妥当性確認リソース」と交換可能に使用され得る。複数の妥当性確認リソースは、分散型妥当性確認ノードを形成してもよい。複数の妥当性確認リソースのうちの少なくとも１つは、１つまたは複数の処理リソースを含み得る。追加的または代替的に、複数の妥当性確認リソースのうちの１つまたは複数は、実質的に本明細書で説明されるようなバリデータコントローラ構成要素を含み得る。

【０１０８】

それぞれの内部ノードは、セグメントルートであり得る。言い換えると、「内部ノード」は、ツリー全体のルートノードでもリーフノードでもないマークルツリー内のノードである。各サブセット内のトランザクションは、それぞれの共通内部ノードを共有し得る。

【０１０９】

別の言い方をすれば、各サブセットは、そのサブセットがマークルツリーの（サブ）部分を提供するおよび／またはそれによって表されるように、マークルツリー内の共通ノードに関連付けられた少なくとも２つのトランザクションを含み得る。共通（内部）ノードは、実質的に本明細書で説明されるようなセグメントノードまたはルートであってもよい。マークルツリーの一部は、実質的に本明細書で説明されるような「セグメント」であってもよい。

【０１１０】

＜付記１．２＞ブロックチェーンブロックおよび／またはブロックチェーントランザクションのサブセットを妥当性確認するステップは、
ｉ）少なくとも１つのブロックチェーントランザクションを妥当性確認および／もしくは検証するステップ、ならびに／または
ｉｉ）簡易支払い検証（ＳＰＶ）プロセスを実行するステップ、ならびに／または
ｉｉｉ）所与のブロックチェーントランザクション（Ｔｘ）がブロックチェーンブロック内に含まれているかどうかを確認するステップ、ならびに／または
ｉｉｉ）ブロックチェーントランザクションのうちの少なくとも１つのブロックチェーントランザクションのハッシュを生成し、ハッシュを使用してマークルパスを構築し、および／もしくはハッシュがブロックチェーンブロックのヘッダ内のトランザクション識別子（ＴｘＩＤ）と一致するかどうかチェックするステップ
を含む、付記１．１に記載の方法。

【０１１１】

＜付記１．３＞ブロックチェーントランザクションのサブセットのうちの少なくとも１つは、サブセットに関連付けられている、サブセットを識別する、および／またはサブセットを表す識別子を含む、
付記１．１または１．２に記載の方法。

【０１１２】

＜付記１．４＞識別子は、マークルツリー内での少なくとも１つのサブセットの位置の計算を容易にする、
付記１．３に記載の方法。

【０１１３】

＜付記１．５＞識別子は、ブロックチェーントランザクションの少なくとも１つのサブセット内のブロックチェーントランザクションのハッシュの一部を含む
付記１．３または１．４に記載の方法。

【０１１４】

＜付記１．６＞ブロックチェーントランザクションのそれぞれのサブセットを複数の妥当性確認リソースに割り当てるステップは、トランザクションのサブセットに関連付けられたそれぞれの識別子に基づいて、それぞれのサブセットをそれぞれの妥当性確認リソースにマッチングさせることを含む、
いずれかの先行する付記に記載の方法。

【０１１５】

＜付記１．７＞ｉ）複数の妥当性確認リソースのうちの少なくとも１つにブロックチェーントランザクションの少なくとも１つのサブセットをダウンロードするステップ、および／または
ｉｉ）複数の妥当性確認リソースのうちの少なくとも１つにブロックチェーントランザクションの少なくとも１つのサブセットを送信するステップ
をさらに含む、いずれかの先行する付記に記載の方法。

【０１１６】

＜付記１．８＞マークルツリーは、複数のブロックチェーントランザクションのハッシュの二分木またはメッシュを含む、
いずれかの先行する付記に記載の方法。

【０１１７】

＜付記１．９＞複数のブロックチェーントランザクション内のブロックチェーントランザクションのサブセットを識別および／または決定するステップ
をさらに含む、いずれかの先行する付記に記載の方法。

【０１１８】

＜付記１．１０＞複数の妥当性確認リソースのうちの少なくとも１つは、
仮想マシン、サーバ、ＧＰＵベースのコンピューティングリソース、処理スレッド、および／またはマルチプロセッサシステム

のうちの１つまたは複数であるか、またはそれを含む、
いずれかの先行する付記に記載の方法。
【０１１９】
＜付記１．１１＞ｉ）少なくとも２つのトランザクションはマークルツリーにおける兄弟であり、および／または
ｉｉ）共通ノードは、少なくとも２つのトランザクションの親または祖先である、
いずれかの先行する付記に記載の方法。
【０１２０】
＜付記１．１２＞複数のブロックチェーントランザクションとブロックのためのマークルツリーのルートとを含むブロックチェーンブロックの少なくとも一部を妥当性確認するように動作するブロックチェーン妥当性確認システムであって、
システムは、複数の妥当性確認リソースを備え、各々の妥当性確認リソースは、
プロセッサと、
実行可能命令を含むメモリと
を備え、実行可能命令は、プロセッサによる実行の結果として、システムに、いずれかの先行する付記に記載のコンピュータ実装方法を実行させる、
ブロックチェーン妥当性確認システム。
【０１２１】
＜付記１．１３＞コンピュータシステムのプロセッサによって実行された結果として、コンピュータシステムに、付記１．１から１．１１のいずれかに記載のコンピュータ実装方法を実行させる実行可能命令を記憶した非一時的コンピュータ可読記憶媒体。
【０１２２】
本開示の別の態様によれば、本明細書で説明または特許請求される任意の方法ステップまたは方法ステップの組合せを実行するように構成されたコンピュータ実装システムが提供される。
【０１２３】
また、複数のコンピュータ実装ノードを含むブロックチェーンシステム（ネットワーク）が提供され、ブロックチェーンネットワーク内の各ノードは、
プロセッサと、
実行可能命令を含むメモリと
を備え、実行可能命令は、プロセッサによる実行の結果として、システムに、本明細書で特許請求または説明されるコンピュータ実装方法のいずれかの変形を実行させる。
【０１２４】
ネットワークは、本明細書で説明されるような記載のブロックチェーンプロトコルを使用して動作するように構成され得る。
【０１２５】
追加的または代替的に、本開示は、ブロックチェーンブロックの少なくとも一部をダウンロードするコンピュータ実装方法を含み得る。ブロックは、複数のブロックチェーントランザクションと、ブロックのためのマークルツリーのルートとを含み得る。本方法は、以下の付記のうちの１つまたは複数に記載されるようなステップを含み得る。
【０１２６】
<u>付記セット２：</u>
＜付記２．１＞複数のブロックチェーントランザクションとブロックのためのマークルツリーのルートとを含むブロックチェーンブロックの少なくとも一部をダウンロードするコンピュータ実装方法であって、
ブロックチェーントランザクションのそれぞれのサブセットを複数の処理リソースに割り当てるステップであって、各それぞれのサブセットは、マークルツリーのそれぞれの部分を提供し、マークルツリーのそれぞれの内部ノードによって表される、ステップと、
複数の処理リソースのうちの１つ、いくつか、またはすべてを使用して、ブロックチェーントランザクションのそれぞれのサブセットをダウンロードするステップと

を含む方法。
【０１２７】
各それぞれのサブセットは、それぞれの内部ノードがそれぞれのサブセットを符号化し得るという意味で、それぞれの内部ノードによって表され得る。すなわち、それぞれの内部ノードは、それぞれのサブセットに基づいて（すなわち、それぞれのサブセットの関数として）生成され得る。それぞれのサブセット内の各トランザクションは、１つまたは複数のハッシュ演算によってそれぞれの内部ノードにリンクされ得る。
【０１２８】
＜付記２．２＞複数の処理リソースのうちの１つ、いくつか、またはすべてが、ブロックチェーントランザクションのそれぞれのサブセットを中央記憶位置に送信するステップを含む、付記２．１に記載の方法。
【０１２９】
＜付記２．３＞マークルツリーのそれぞれの内部ノードは、マークルツリーにおいてそれぞれの位置を有し、本方法は、
マークルツリーのそれぞれの内部ノードのそれぞれの位置に基づいて、ブロックチェーントランザクションのそれぞれのサブセットを配置するステップ
を含む、付記２．２に記載の方法。
【０１３０】
＜付記２．４＞処理リソースのうちの１つ、いくつか、またはすべてが、ブロックチェーントランザクションのそれぞれのダウンロードされたサブセットに基づいて、マークルツリーのそれぞれの候補内部ノードを生成するステップ
を含み、
それぞれの候補内部ノードがマークルツリーのそれぞれの内部ノードと一致することを検証するステップ、および／または
マークルツリーのルートに基づいてマークル証明を実行することによって、それぞれの候補内部ノードがマークルツリーのノードであることを検証するステップ、および／または
マークルツリーのそれぞれの候補内部ノードを１つまたは複数の他の処理リソースに送信するステップ
のうちの少なくとも１つをさらに含む、いずれかの先行する付記に記載の方法。
【０１３１】
＜付記２．５＞複数の処理リソースのうちの１つ、いくつか、またはすべてを使用して、ブロックチェーントランザクションのそれぞれのサブセットを妥当性確認するステップを含む、いずれかの先行する付記に記載の方法。
【０１３２】
＜付記２．６＞ブロックチェーントランザクションのそれぞれのサブセットを妥当性確認するステップは、
ｉ）少なくとも１つのブロックチェーントランザクションを妥当性確認および／もしくは検証するステップ、ならびに／または
ｉｉ）簡易支払い検証プロセスを実行するステップ、ならびに／または
ｉｉｉ）所与のブロックチェーントランザクションがブロックチェーンブロック内に含まれているかどうかを確認するステップ、ならびに／または
ｉｉｉ）ブロックチェーントランザクションのうちの少なくとも１つのブロックチェーントランザクションのハッシュを生成し、ハッシュを使用してマークルパスを構築し、および／もしくはハッシュがブロックチェーンブロックのヘッダ内のトランザクション識別子と一致するかどうかチェックするステップ
付記２．１に記載の方法。
【０１３３】
＜付記２．７＞ブロックチェーントランザクションのそれぞれのサブセットのうちの少なくとも１つは、それぞれのサブセットに関連付けられた、それを識別する、および／ま

たはそれを表すそれぞれの識別子を含む、
いずれかの先行する付記に記載の方法。
【０１３４】
＜付記２．８＞それぞれの識別子は、マークルツリー内の少なくとも１つのそれぞれのサブセットのそれぞれの位置の計算を容易にする
付記２．７に記載の方法。
【０１３５】
＜付記２．９＞それぞれの識別子は、マークルツリーのそれぞれの内部ノードに基づく、
付記２．７または２．８に記載の方法。
【０１３６】
＜１０＞それぞれの識別子は、マークルツリーのそれぞれの内部ノードの一部を含む、
請求項９に記載の方法。
【０１３７】
＜付記２．１０＞ブロックチェーントランザクションのそれぞれのサブセットを複数のそれぞれの処理リソースに割り当てるステップは、トランザクションのそれぞれのサブセットに関連付けられたそれぞれの識別子に基づいて、それぞれのサブセットをそれぞれの処理リソースにマッチングさせるステップを含む、
いずれかの先行する付記に記載の方法。
【０１３８】
＜付記２．１１＞マークルツリーは、複数のブロックチェーントランザクションのハッシュの二分木またはメッシュ構造を含む、
いずれかの先行する付記に記載の方法。
【０１３９】
＜付記２．１２＞複数のブロックチェーントランザクション内のブロックチェーントランザクションのサブセットを識別および／または決定するステップ
を含む、いずれかの先行する付記に記載の方法。
【０１４０】
＜付記２．１３＞複数の処理リソースのうちの少なくとも１つは、仮想マシン、サーバ、ＧＰＵベースのコンピューティングリソース、またはマルチプロセッサシステムであるか、またはそれを含む、
いずれかの先行する付記に記載の方法。
【０１４１】
＜付記２．１４＞複数のブロックチェーントランザクションとブロックのためのマークルツリーのルートとを含むブロックチェーンブロックの少なくとも一部をダウンロードするように動作するブロックチェーン処理システムであって、本システムは、複数の処理リソースを含み、複数の処理リソースの各々は、
プロセッサと、
実行可能命令を含むメモリと
を備え、実行可能命令は、プロセッサによる実行の結果として、システムに、付記２．１から２．１４のいずれか１つに記載のコンピュータ実装方法を実行させるか、または実行することを可能にする、
ブロックチェーン処理システム。
【０１４２】
＜付記２．１５＞コンピュータシステムのプロセッサによって実行された結果として、コンピュータシステムに、付記２．１から２．１４のいずれかに記載のコンピュータ実装方法を実行させるか、または実行することを可能にする実行可能命令を記憶した非一時的コンピュータ可読記憶媒体。
【０１４３】
別の態様によれば、コンピュータシステムのプロセッサによって実行された結果として

、コンピュータシステムに、本明細書で特許請求または説明されるコンピュータ実装方法の任意のバージョンを実行させる実行可能命令を記憶した非一時的コンピュータ可読記憶媒体が提供される。

【０１４４】

付記セット３：

＜付記３．１＞コンピュータ実装方法であって、

ブロックチェーンブロックの複数のブロックチェーントランザクション（ＴＸ）中のトランザクション（Ｔｘ）にそれぞれ関連付けられた複数の未使用トランザクション出力（ＵＴＸＯ）を記録、検索および／または処理するための第１のＵＴＸＯリポジトリを生成、記憶および／または維持するステップ

を含み、

複数のブロックチェーントランザクションは、ブロックチェーンブロックについてのマークルツリーの一部を提供し、および／またはそれによって表される、

方法。

【０１４５】

＜付記３．２＞少なくとも１つのさらなるＵＴＸＯリポジトリを生成、記憶、および／または維持するステップ

を含む、付記３．１に記載の方法。

【０１４６】

＜付記３．３＞ＵＴＸＯリポジトリに関連するアクション、変更、およびイベントの履歴を含むデータベースログを作成および／または維持するステップ

をさらに含む、付記３．１または３．２に記載の方法。

【０１４７】

＜付記３．４＞第１の出力リポジトリおよび／またはＵＴＸＯリポジトリは、

ｉ）未使用トランザクション出力、および／または

ｉｉ）ａ）未使用トランザクション出力（ＵＴＸＯ）、および／またはｂ）複数のブロックチェーントランザクション中のトランザクション（Ｔｘ）に関連付けられた識別子

に関連付けられた少なくとも１つの記録を含む、いずれかの先行する付記に記載の方法。

【０１４８】

＜付記３．５＞少なくとも１つの記録は、

ｉ）ブロックチェーンブロックに関連付けられたブロック識別子（block_ID）、および／または

ｉｉ）複数のブロックチェーントランザクション中のトランザクション（Ｔｘ）に関連付けられたトランザクション識別子（ＴｘＩＤ）

を有する記録識別子を含む、付記３．４に記載の方法。

【０１４９】

＜付記３．６＞ｉ）記録識別子は、ブロック識別子（block_ID）とトランザクション識別子（ＴｘＩＤ）との関数を含み、および／または

ｉｉ）ブロック識別子（block_ID）とトランザクション識別子（ＴｘＩＤ）との連結、および／または

ｉｉｉ）複数のブロックチェーントランザクションのトランザクションは、未使用トランザクション出力（ＵＴＸＯ）に関連付けられる、

付記３．５に記載の方法。

【０１５０】

＜付記３．７＞記録識別子を使用して、ＵＴＸＯリポジトリにおいて少なくとも１つの記録を検索、識別、アクセス、または挿入するステップ

をさらに含む、付記３．５または３．６に記載の方法。

【０１５１】

＜付記３．８＞複数の未使用トランザクション出力（ＵＴＸＯ）における少なくとも１

つのＵＴＸＯは、ＵＴＸＯリポジトリにおいてロックフラグに関連付けられ、ロックフラグは、
ｉ）未使用トランザクション出力（ＵＴＸＯ）が使用可能か使用不可能であるかを示し、および／または
ｉｉ）未使用トランザクション出力の使用が許可されることを示す第１の状態と、未使用トランザクション出力の使用が禁止されることを示す第２の状態との間で構成可能である、
任意の先行する付記のいずれかに記載の方法。
【０１５２】
＜付記３．９＞方法は、
ｉ）未使用トランザクション出力（ＵＴＸＯ）をロックフラグに関連付けるステップ、および／または
ｉｉ）ロックフラグの状態を第１の状態から第２の状態に、または第２の状態から第１の状態に変更するステップ
を含む、付記３．８に記載の方法。
【０１５３】
＜付記３．１０＞第１の処理リソースから少なくとも１つのさらなる処理リソースに通信を送信して、少なくとも１つのさらなる処理リソースに、未使用トランザクション出力に関連付けられたロックフラグの状態を、第１の状態から第２の状態に、または第２の状態から第１の状態に変更させるステップ
をさらに含む、付記３．８または３．９に記載の方法。
【０１５４】
＜付記３．１１＞通信は、
ｉ）トランザクション（ＴＸ）、トランザクション識別子（ＴｘＩＤ）、および／またはトランザクション（Ｔｘ）のハッシュ、ならびに
ｉｉ）１つまたは複数の未使用トランザクション出力（ＵＴＸＯ）のリスト
を含む、付記３．１０に記載の方法。
【０１５５】
＜付記３．１２＞少なくとも１つのさらなる処理リソースにおいて通信を受信するステップと、
ロックフラグの状態を、第１の状態から第２の状態に、または第２の状態から第１の状態に変更するステップ
を含む、付記３．１０または３．１１に記載の方法。
【０１５６】
＜付記３．１３＞ｉ）マークルツリーの一部は、ブロックチェーンブロックのためのマークルツリーのサブ部分またはセグメントである、および／または
ｉｉ）複数のブロックチェーントランザクションは、マークルツリーの内部ノードによって表される、
いずれかの先行する付記に記載の方法。
【０１５７】
＜付記３．１４＞複数の処理リソースを含むブロックチェーン実装システムであって、処理リソースの各々は、
プロセッサと、
実行可能命令を含むメモリと
を備え、実行可能命令は、プロセッサによる実行の結果として、システムに、いずれかの先行する付記に記載のコンピュータ実装方法を実行させるか、または実行することを可能にする、
ブロックチェーン処理システム。
【０１５８】
＜付記３．１５＞コンピュータシステムのプロセッサによって実行された結果として、

コンピュータシステムに、付記３．１から３．１３のいずれかに記載のコンピュータ実装方法を実行させるか、または実行することを可能にする実行可能命令を記憶した非一時的コンピュータ可読記憶媒体。

【０１５９】

付記セット４

付記セット４の任意の付記または付記の組合せにおいて定義された任意の実施形態は、付記セット１から３の任意の付記（複数可）を実施するか、またはそれと組み合わせるように構成され得る。

【０１６０】

＜付記４．１＞複数のブロックチェーントランザクションとブロックのためのマークルツリーのルートとを含むブロックチェーンブロックの少なくとも一部を妥当性確認するように動作するシステムであって、

複数の妥当性確認リソース

を含み、複数の妥当性確認リソースの各々は、

実行可能命令を記憶するメモリの少なくとも一部に関連付けられた少なくとも１つのプロセッサ

を備え、実行可能命令は、少なくとも１つのプロセッサによる実行の結果として、妥当性確認リソースに、

複数のブロックチェーントランザクションの少なくとも１つのサブセットを妥当性確認することであって、少なくとも１つのサブセットは、マークルツリーの一部を提供し、マークルツリーの内部ノードによって表される、処理すること

を実行させるか、または実行することを可能にする、システム。

【０１６１】

＜付記４．２＞ｉ）複数の妥当性確認リソース間での複数のブロックチェーントランザクションの複数のサブセットの分散のバランシングを容易にするように構成されたロードバランシング構成要素、および／または

ｉｉ）複数のブロックチェーントランザクションの少なくとも１つのサブセットの識別を容易にするように構成されたセグメント識別構成要素、および／または

ｉｉｉ）割り当てユニット、および／または

ｉｖ）システムと１つまたは複数のデータソースまたは宛先との間で通信を送信または受信するための１つまたは複数のインターフェース

をさらに備える、付記４．１に記載のシステム。

【０１６２】

＜付記４．３＞システムは、少なくとも１つのコントローラ構成要素を備え、少なくとも１つのコントローラ構成要素は、

少なくとも１つの妥当性確認リソース、

少なくとも１つの妥当性確認リソースの少なくとも１つのプロセッサ、

１つまたは複数のインターフェース、

１つまたは複数のロードバランシング構成要素、および／または

複数のブロックチェーントランザクションの少なくとも１つのサブセットの識別を容易にするように構成された１つまたは複数のセグメント識別構成要素

のうちの少なくとも１つの動作に影響を及ぼし、および／またはそれを制御するように構成される、付記４．１または４．２に記載のシステム。

【０１６３】

＜付記４．４＞ｉ）複数のブロックチェーントランザクション中の少なくとも２つのトランザクションは、マークルツリーにおける兄弟であり、および／または

ｉｉ）内部ノードは、ブロックチェーントランザクションのサブセットの親または祖先である、

いずれかの先行する付記に記載のシステム。

【０１６４】

＜付記４．５＞複数のＵＴＸＯリポジトリであって、複数のリポジトリのうちの各リポジトリは、それぞれの妥当性確認リソースに関連付けられ、複数の未使用トランザクション出力（ＵＴＸＯ）の記録、検索、および／または処理を容易にするように構成される、複数のＵＴＸＯリポジトリ
をさらに備え、
好ましくは、各複数の未使用トランザクション出力は、複数のブロックチェーントランザクション中の少なくとも１つのトランザクション（Ｔｘ）に関連付けられる、
いずれかの先行する付記に記載のシステム。

【０１６５】
＜付記４．６＞複数のＵＴＸＯリポジトリのうちの少なくとも１つに関するアクション、変更、およびイベントの履歴を含むデータベースログを作成および／または維持する
ように動作する、付記４．５に記載のシステム。

【０１６６】
＜付記４．７＞複数のＵＴＸＯリポジトリのうちの少なくとも１つは、
ｉ）未使用トランザクション出力（ＵＴＸＯ）、および／または
ｉｉ）ａ）未使用トランザクション出力および／またはｂ）複数のブロックチェーントランザクション中のトランザクション（Ｔｘ）に関連付けられた識別子
に関連付けられた少なくとも１つの記録を含む、
いずれかの先行する付記に記載のシステム。

【０１６７】
＜付記４．８＞少なくとも１つの記録は、
ｉ）ブロックチェーンブロックに関連付けられたブロック識別子（block_ID）、および／または
ｉｉ）複数のブロックチェーントランザクション中のトランザクション（Ｔｘ）に関連付けられたトランザクション識別子（ＴｘＩＤ）
を有する記録識別子を含む、付記４．７に記載のシステム。

【０１６８】
＜付記４．９＞記録識別子は、
ｉ）ブロック識別子（block_ID）とトランザクション識別子（ＴｘＩＤ）との関数、および／または
ｉｉ）ブロック識別子（block_ID）とトランザクション識別子（ＴｘＩＤ）との連結、
を含む、付記４．７または４．８に記載のシステム。

【０１６９】
＜付記４．１０＞システムは、
記録識別子を使用して、複数のＵＴＸＯリポジトリ中の少なくとも１つのＵＴＸＯリポジトリにおいて少なくとも１つの記録を検索、識別、アクセス、または挿入する
ように動作する、請求項８または９に記載のシステム。

【０１７０】
＜付記４．１１＞複数の未使用トランザクション出力（ＵＴＸＯ）のうちの少なくとも１つのＵＴＸＯは、ロックフラグに関連付けられ、ロックフラグは、
ｉ）未使用トランザクション出力（ＵＴＸＯ）が使用可能か使用不可能であるかを示し、および／または
ｉｉ）未使用トランザクション出力の使用が許可されることを示す第１の状態と、未使用トランザクション出力の使用が禁止されることを示す第２の状態との間で構成可能である、
付記４．５から４．１０のいずれかに記載のシステム。

【０１７１】
＜付記４．１２＞システムは、
ロックフラグの状態を、第１の状態から第２の状態に、または第２の状態から第１の状

態に変更する
　ように動作する、付記４．１１に記載のシステム。
【０１７２】
　＜付記４．１３＞ｉ）ブロックチェーントランザクションのそれぞれのサブセットを複数の妥当性確認リソースに割り当てること、および
　ｉｉ）複数の妥当性確認リソースのうちの１つ、いくつか、またはすべてを使用して、ブロックチェーントランザクションのそれぞれのサブセットをダウンロードおよび／または受信すること
　を行うように動作する、いずれかの先行する付記に記載のシステム。
【０１７３】
　＜付記４．１４＞妥当性確認リソースのうちの１つ、いくつか、またはすべてを使用して、ブロックチェーントランザクションのそれぞれのダウンロードされたサブセットに基づいて、マークルツリーのそれぞれの候補内部ノードを生成すること
　を行うように動作し、
　それぞれの候補内部ノードがマークルツリーのそれぞれの内部ノードと一致することを検証すること、および／または
　マークルツリーのルートに基づいてマークル証明を実行することによって、それぞれの候補内部ノードがマークルツリーのノードであることを検証すること、および／または
　マークルツリーのそれぞれの候補内部ノードを１つまたは複数の他の処理リソースに送信すること
　のうちの少なくとも１つを行うようにさらに動作する、いずれかの先行する付記に記載のシステム。
【０１７４】
　＜付記４．１５＞ｉ）少なくとも１つのブロックチェーントランザクションを妥当性確認および／もしくは検証すること、および／または
　ｉｉ）簡易支払い検証プロセスを実行すること、および／または
　ｉｉｉ）所与のブロックチェーントランザクションがブロックチェーンブロック内に含まれているかどうかを確認すること、および／または
　ｉｉｉ）ブロックチェーントランザクションのうちの少なくとも１つのブロックチェーントランザクションのハッシュを生成し、ハッシュを使用してマークルパスを構築し、および／もしくはハッシュがブロックチェーンブロックのヘッダ内のトランザクション識別子と一致するかどうかチェックすること
　を行うように動作する、いずれかの先行する付記に記載のシステム。
【０１７５】
　＜付記４．１６＞複数の妥当性確認リソースのうちの少なくとも１つは、仮想マシン、サーバ、ＧＰＵベースのコンピューティングリソース、スレッド、および／またはマルチプロセッサシステムのうちの少なくとも１つであるか、またはそれを含む、
　いずれかの先行する付記に記載のシステム。
【０１７６】
本開示の例示的な実施形態を実施するための例示的な技術環境
　ここで、本開示の１つまたは複数の実施形態が実施され得るコンピューティング環境の概要を説明する。しかしながら、上述のように、この文脈は限定を意図するものではなく、実施形態は、ブロックチェーンを介して実装されないデータ記録および構造の処理に対して実施されてもよい。非ブロックチェーンの実施形態は、例えば、分散型台帳ではなくデータベースを使用して考案され得る。
【０１７７】
　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、典型的にはインターネットなどの広域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で構成され得る。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内にピアツーピア（Ｐ２Ｐ）ネットワーク１０６を形成する

ように構成され得る複数のブロックチェーンノード１０４を含む。図示されていないが、ブロックチェーンノード１０４は、ほぼ完全なグラフとして構成され得る。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。

【０１７８】

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含み、ノード１０４の異なるものは、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、例えば、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）などの他の機器を含む処理装置を備える。各ノードはまた、メモリ、すなわち、１つまたは複数の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、例えば、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／または光ディスクドライブなどの光学媒体を採用する１つまたは複数のメモリユニットを備え得る。

【０１７９】

ブロックチェーン１５０は、データブロック１５１のチェーンを含み、ブロックチェーン１５０のそれぞれのコピーは、分散型またはブロックチェーンネットワーク１０６内の複数のブロックチェーンノード１０４の各々で維持される。上述したように、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（後述する）を記憶している限り、データがプルーニングされ得る。チェーン内の各ブロック１５１は、１つまたは複数のトランザクション１５２を含み、この文脈におけるトランザクションは、データ構造の一種を指す。データ構造の性質は、トランザクションモデルまたは方式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、全体を通して１つの特定のトランザクションプロトコルを使用する。１つの一般的なタイプのトランザクションプロトコルでは、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つの出力を含む。各出力は、プロパティとしてデジタル資産の量を表す額を指定し、その例は、出力が暗号的にロックされている（ロック解除され、それによって償還または使用されるためには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）ユーザ１０３である。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、それによってトランザクションをリンクする。

【０１８０】

各ブロック１５１はまた、ブロック１５１への順番を定義するために、チェーン内の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を含む。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への順序を定義するために、前のトランザクションへ戻るポインタを含む（注意：トランザクション１５２のシーケンスは分岐することが可能である）。ブロック１５１のチェーンは、チェーン内の最初のブロックであったジェネシスブロック（Ｇｂ：genesis block）１５３まで戻る。チェーン１５０内の早期にある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。

【０１８１】

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にフォワードし、それによってトランザクション１５２をネットワーク１０６全体に伝搬させ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に組み込まれるのを待っているトランザクション１５２の順序付きセット（またはプール

）１５４を維持する。順序付きプール１５４は、「メムプール」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するものではない。これは、ノード１０４が有効であるとして受け入れたトランザクションの順序付きセットを指し、それに対して、ノード１０４は、同じ出力を使用しようとする他のトランザクションを受け入れないように義務付けられている。

【０１８２】

所与の現在のトランザクション１５２ｊにおいて、その入力（または各入力）は、トランザクションのシーケンスにおける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含み、この出力が現在のトランザクション１５２ｊにおいて償還または「使用」されるべきであることを指定する。一般に、先行するトランザクションは、順序付きセット１５４または任意のブロック１５１内の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が有効となるためには存在および妥当性確認される必要があるが、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１５２ｊが作成されるときまたはネットワーク１０６に送信されるときに必ずしも存在する必要はない。したがって、本明細書における「先行する（preceding）」は、ポインタによってリンクされた論理シーケンスにおける先行するものを指し、必ずしも時間シーケンスにおける作成または送信の時間を指すものではなく、したがって、トランザクション１５２ｉ、１５２ｊが順不同に作成または送信されることを必ずしも除外するものではない（オーファントランザクションに関する以下の説明を参照）。先行するトランザクション１５２ｉは、同様に、先のトランザクション（antecedent transaction）または先行したトランザクション（predecessor transaction）とも呼ばれる。

【０１８３】

現在のトランザクション１５２ｊの入力はまた、入力認可、例えば、先行するトランザクション１５２ｉの出力がロックされている対象のユーザ１０３ａの署名を含む。次に、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に暗号的にロックされ得る。したがって、現在のトランザクション１５２ｊは、先行するトランザクション１５２ｉの入力において定義された額を、現在のトランザクション１５２ｊの出力において定義されたように、新しいユーザまたはエンティティ１０３ｂに転送することができる。いくつかのケースでは、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、残り（change）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）間で入力額を分割するために複数の出力を有し得る。いくつかのケースでは、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まとめ、現在のトランザクションの１つまたは複数の出力に再分配するために複数の入力を有することができる。

【０１８４】

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、個々のユーザまたは組織などの当事者１０３が（手動でまたは当事者によって採用される自動プロセスによって）新しいトランザクション１５２ｊを実施することを望むとき、実施者（enacting party）は、新しいトランザクションをそのコンピュータ端末１０２から受信者に送信する。実施者または受信者は、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の１つまたは複数（これは、現在では、典型的にはサーバまたはデータセンタであるが、原則として他のユーザ端末であってもよい）に送信する。新しいトランザクション１５２ｊを実施する当事者１０３がトランザクションをブロックチェーンノード１０４の１つまたは複数に直接送信し、いくつかの例では、受信者に送信しないことも除外されない。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の各々で適用されるブロックチェーンノードプロトコルにしたがって、トランザクションが有効であるかどうかをチェックする。ブロックチェーンノードプロトコルは、典型的には、新しいトランザクション１５２ｊ内の暗号署名が、トランザクション１５２の順序付きシーケンス内で前のトランザクション１５２ｉ

に依存する予想される署名と一致することをチェックするようにブロックチェーンノード１０４に要求する。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の入力に含まれる当事者１０３の暗号署名または他の認可が、新しいトランザクションが使用する（または「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チェックすることを含み得、この条件は、典型的には、新しいトランザクション１５２ｊの入力における暗号署名または他の認可が、新しいトランザクションの入力がリンクされている前のトランザクション１５２ｉの出力をロック解除することを少なくともチェックすることを含む。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。代替的に、それは、単にブロックチェーンノードプロトコルのみによって固定されてもよいし、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内の１つまたは複数の他のブロックチェーンノード１０４にフォワードする。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したがって同じテストを適用し、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にフォワードし、以下同様である。このようにして、新しいトランザクションはブロックチェーンノード１０４のネットワーク全体に伝搬される。

【０１８５】

出力ベースのモデルでは、所与の出力（例えば、ＵＴＸＯ）が割り当てられる（または、「使用される」）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したがって別の以降のトランザクション１５２ｊの入力によって有効に償還されたかどうかである。トランザクションが有効であるための別の条件は、それが償還しようとす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償還されていないことである。この場合も同様に、有効ではない場合、トランザクション１５２ｊは、（無効としてフラグ付けされ、警告のために伝搬されない限り）伝搬されることも、ブロックチェーン１５０内に記録されることもない。これは、トランザクタ（transactor）が同じトランザクションの出力を複数回割り当てようとする二重使用を防止する。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使用を防止する。ここでも、トランザクション順序が定義されているので、アカウント残高は常に単一の定義された状態にある。

【０１８６】

トランザクションを妥当性確認することに加えて、ブロックチェーンノード１０４はまた、「プルーフオブワーク」によって支持される、一般にマイニングと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しようと競い合う。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が、ブロックチェーン１５０上に記録されたブロック１５１内にまだ現れていない有効なトランザクションの順序付きプール１５４に追加される。次いで、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で、トランザクションの順序付きセット１５４からトランザクション１５２の新しい有効なブロック１５１を組み立てようと競い合う。典型的には、これは、ナンスが保留中のトランザクションの順序付きプール１５４の表現と連結されハッシュされたときにハッシュの出力が所定の条件を満たすような「ナンス」値を検索することを含む。例えば、所定の条件とは、ハッシュの出力が特定の所定の数の先行ゼロを有することであり得る。これは、プルーフオブワークパズルの１つの特定のタイプにすぎず、他のタイプが除外されないことに留意されたい。ハッシュ関数のプロパティは、その入力に対して予測不可能な出力を持つことである。したがって、この検索は、総当たりでしか実行することができないので、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でかなりの量の処理リソースを消費する。

【０１８７】

最初にパズルを解いたブロックチェーンノード１０４は、これをネットワーク１０６に

公表し、後にネットワーク内の他のブロックチェーンノード１０４によって容易にチェックすることができるその解をプルーフとして提供する（ハッシュに対する解が与えられると、ハッシュの出力が条件を満たすことをチェックすることは簡単である）。この最初のブロックチェーンノード１０４は、ブロックを、このブロックを受け入れる他のノードのしきい値コンセンサスに伝搬し、プロトコルルールを強制する。次いで、トランザクションの順序付きセット１５４は、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内に新しいブロック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、チェーン内の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、例えばハッシュの形態の、かなりの量の労力は、ブロックチェーンプロトコルのルールに従うという最初のノード１０４の意図を示す。そのようなルールは、前に妥当性確認されたトランザクションと同じ出力の使用または割当てを行った場合にトランザクションを有効として受け入れること（別名二重使用としても知られている）を行わないことを含む。ブロック１５１は、一旦作成されると、ブロックチェーンネットワーク１０６内のブロックチェーンノード１０４の各々において認識および維持されるので、修正することができない。ブロックポインタ１５５はまた、ブロック１５１に順番を付与する。トランザクション１５２は、ネットワーク１０６内の各ブロックチェーンノード１０４において順序付きブロックに記録されるので、これは、トランザクションの不変の公開台帳を提供する。

【０１８８】

任意の所与の時間にパズルを解こうと競い合う異なるブロックチェーンノード１０４は、それらがいつ解を検索し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、まだ公開されていないトランザクションのプール１５４の異なるスナップショットに基づいてそれを行っていてもよいことに留意されたい。誰がそれぞれのパズルを最初に解いても、どのトランザクション１５２が次の新しいブロック１５１ｎに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プール１５４が更新される。次いで、ブロックチェーンノード１０４は、新たに定義された、公開されていないトランザクションの順序付きプール１５４からブロックを作成しようと競い合い続け、以下同様である。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いて、ブロックチェーンの相反する見解がノード１０４間で伝搬される場合に発生し得る任意の「フォーク」を解決するためのプロトコルも存在する。要するに、フォークのどちらのプロングでも最も長く成長した方が、確定的なブロックチェーン１５０となる。同じトランザクションが両方のフォークに現れるので、これがネットワークのユーザまたはエージェントに影響を与えないことに留意されたい。

【０１８９】

ビットコインブロックチェーン（およびほとんどの他のブロックチェーン）によれば、新しいブロック１０４の構築に成功したノードには、（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転送する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）追加の定義された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、受け入れられた追加の額のデジタル資産を新たに割り当てる能力が与えられる。この特別なタイプのトランザクションは、通常、「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（initiation transaction）」または「生成トランザクション（generation transaction）」と称されることもある。これは典型的に、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、新しいブロックを構築するノードが、この特別なトランザクションが後に償還されることを可能にするプロトコルルールに従うという意図を示す。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が償還され得る前に、満期期間、例えば１００個のブロックを必要とし得る。多くの場合、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

の出力のうちの１つにおいて追加のトランザクション手数料を指定する。この手数料は通常「トランザクション手数料」と呼ばれ、以下で説明される。

【０１９０】

トランザクション妥当性確認および公開に関与するリソースに起因して、典型的には、ブロックチェーンノード１０４の少なくとも各々は、１つまたは複数の物理サーバユニットで構成されるサーバの形態をとるか、さらにはデータセンタ全体の形態をとる。しかしながら、原則として、任意の所与のブロックチェーンノード１０４は、ユーザ端末または互いにネットワーク化されたユーザ端末のグループの形態をとることができる。

【０１９１】

各ブロックチェーンノード１０４のメモリは、そのそれぞれの１つまたは複数の役割を実行し、ブロックチェーンノードプロトコルにしたがってトランザクション１５２を処理するために、ブロックチェーンノード１０４の処理装置上で実行されるように構成されたソフトウェアを記憶する。本明細書においてブロックチェーンノード１０４に帰する任意のアクションは、それぞれのコンピュータ機器の処理装置上で実行されるソフトウェアによって実行され得ることが理解されよう。ノードソフトウェアは、アプリケーション層、またはオペレーティングシステム層もしくはプロトコル層などの下位層、またはこれらの任意の組合せにおける１つまたは複数のアプリケーションにおいて実装され得る。

【０１９２】

消費ユーザの役割を担う複数の当事者１０３の各々のコンピュータ機器１０２もネットワーク１０１に接続されている。これらのユーザは、ブロックチェーンネットワーク１０６と対話し得るが、トランザクションの妥当性確認にもブロックの構築にも参加しない。これらのユーザまたはエージェント１０３のうちのいくつかは、トランザクションの送信者および受信者として動作し得る。他のユーザは、必ずしも送信者または受信者として動作しなくても、ブロックチェーン１５０と対話し得る。例えば、いくつかの当事者は、ブロックチェーン１５０のコピーを記憶する（例えば、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動作し得る。

【０１９３】

当事者１０３のうちのいくつかまたはすべては、異なるネットワーク、例えば、ブロックチェーンネットワーク１０６の上にオーバーレイされた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１０６を含むシステムの一部であるといえるが、これらのユーザは、ブロックチェーンノードに要求される役割を果たさないので、ブロックチェーンノード１０４ではない。代わりに、各当事者１０３はブロックチェーンネットワーク１０６と対話してもよく、それによって、ブロックチェーンノード１０６に接続する（すなわち通信する）ことでブロックチェーン１５０を利用し得る。２つの当事者１０３およびそれぞれの機器１０２、すなわち、第１の当事者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の当事者１０３ｂおよびそのそれぞれのコンピュータ機器１０２ｂは、例示の目的で示されている。そのような当事者１０３およびそれぞれのコンピュータ機器１０２ははるかに多く存在し、システム１００に参加し得るが、便宜上、それらは図示されていないことが理解されよう。各当事者１０３は、個人または組織であり得る。純粋に例示として、第１の当事者１０３ａは、本明細書ではアリスと呼ばれ、第２の当事者１０３ｂはボブと呼ばれるが、これは限定的なものではなく、本明細書におけるアリスまたはボブへのいかなる言及も、それぞれ「第１の当事者」および「第２の当事者」と置き換えられ得ることが理解されよう。

【０１９４】

各当事者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、例えば、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を含むそれぞれの処理装置を備える。各当事者１０３のコンピュータ機器１０２は、メモリ、すなわち、１つまたは複数の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さらに備える。このメモリは、１つまたは

複数のメモリ媒体、例えば、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／または光ディスクドライブなどの光学媒体を採用する１つまたは複数のメモリユニットを備え得る。各当事者１０３のコンピュータ機器１０２上のメモリは、処理装置上で実行されるように構成された少なくとも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含むソフトウェアを記憶する。本明細書において所与の当事者１０３に帰する任意のアクションは、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よう。各当事者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、例えば、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含む。所与の当事者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化されたリソースを含み得る。

【０１９５】

クライアントアプリケーション１０５は、最初に、１つまたは複数の適切なコンピュータ可読ストレージ上で任意の所与の当事者１０３のコンピュータ機器１０２に提供され得、例えば、サーバからダウンロードされ得るか、またはリムーバブル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記憶デバイス上で提供され得る。

【０１９６】

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。これは２つの主要な機能を有する。これらのうちの１つは、それぞれの当事者１０３が、トランザクション１５２を作成し、認可し（例えば署名し）、１つまたは複数のビットコインノード１０４に送信することを可能にして、トランザクション１５２をブロックチェーンノード１０４のネットワーク全体に伝搬させ、それによってブロックチェーン１５０に含まれるようにすることである。もう１つは、それぞれの当事者に、その当事者が現在所有しているデジタル資産の額を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、当該当事者に属するブロックチェーン１５０全体に散在している様々なトランザクション１５２の出力において定義された額を照合することを含む。

【０１９７】

様々なクライアント機能が、所与の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得るが、必ずしもこれに限定されるものではなく、代わりに、本明細書で説明される任意のクライアント機能は、例えば、ＡＰＩを介してインターフェースする、または一方が他方へのプラグインである２つ以上の別個のアプリケーション一式において実装され得ることに留意されたい。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もしくはオペレーティングシステムなどの下位層、またはこれらの任意の組合せにおいて実装され得る。以下では、クライアントアプリケーション１０５に関して説明するが、これに限定されないことが理解されよう。

【０１９８】

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。これにより、クライアント１０５のウォレット機能はトランザクション１５２をネットワーク１０６に送信することができる。クライアント１０５はまた、それぞれの当事者１０３が受信者である任意のトランザクションについてブロックチェーン１５０にクエリを行うためにブロックチェーンノード１０４にコンタクトすることができる（または、実施形態では、ブロックチェーン１５０は、部分的にその公開性（public visibility）を通じてトランザクションにおける信頼を提供する公共施設であるので、実際にブロックチェーン１５０における他の当事者のトランザクションを検査する）。各コンピュータ機器１０２上のウォレット機能は、トランザクションプロ

トコルにしたがってトランザクション１５２を定式化し、送信するように構成される。上述したように、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したがってトランザクション１５２を妥当性確認し、トランザクション１５２をフォワードして、それらをブロックチェーンネットワーク１０６全体に伝搬するように構成された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に従い（ｇｏ　ｗｉｔｈ）、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内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。ネットワーク１０６内のすべてのノード１０４によって同じノードプロトコルが使用される。

【０１９９】

所与の当事者１０３、例えばアリスが、ブロックチェーン１５０に含まれるべき新しいトランザクション１５２ｊを送信することを望むとき、アリスは、関連トランザクションプロトコルにしたがって（アリスのクライアントアプリケーション１０５内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定式化する。次いで、アリスは、クライアントアプリケーション１０５から、アリスが接続されている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。例えば、これは、アリスのコンピュータ１０２に最良に接続された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は、新しいトランザクション１５２ｊを受信すると、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したがってそれを処理する。これには、新たに受信されたトランザクション１５２ｊが「有効」であるための特定の条件を満たすかを最初にチェックすることが含まれ、その例については、以下でより詳細に説明する。いくつか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。代替的に、条件は、単にノードプロトコルの組込み特徴であってもよいし、スクリプトとノードプロトコルとの組合せによって定義されてもよい。

【０２００】

新たに受信されたトランザクション１５２ｊが、有効であるとみなされるためのテストにパスすることを条件として（すなわち、それが「妥当性確認される」ことを条件として）、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は、そ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きセット１５４に新たな妥当性確認済みトランザクション１５２を追加する。さらに、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は、妥当性確認済みトランザクション１５２をネットワーク１０６内の１つまたは複数の他のブロックチェーンノード１０４に伝搬する。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、ネットワーク１０６全体にわたってすぐに伝搬されることを意味する。

【０２０１】

所与のブロックチェーンノード１０４において維持される保留中のトランザクションの順序付きプール１５４に承認されると、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を含むそれぞれのプール１５４の最新バージョンに対してプルーフオブワークパズルを解こうと競い始める（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プール１５４に基づいてパズルを解こうと試みている可能性があるが、どのノードでも最初に解いたものが、最新のブロック１５１に含まれるトランザクションのセットを定義することを想起されたい。最終的に、ブロックチェーンノード１０４は、アリスのトランザクション１５２ｊを含む順序付きプール１５４の一部についてパズルを解くことになる。）新しいトランザクション１５２ｊを含むプール１５４に対してプルーフオブワークが行われると、それは普遍的にブロックチェーン１５０内のブロック１５１のうちの１つの一部となる。各トランザクション１５２は、先のトランザクションへ戻るポインタを含むので、トランザクションの順序も不変的に記録される。

【０２０２】
異なるブロックチェーンノード１０４は、最初、所与のトランザクションの異なるインスタンスを受信し得るので、１つのインスタンスが新しいブロック１５１において公開される（この時点で、公開された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が同意している）までは、どのインスタンスが「有効」であるかについて相反する見解を有する。ブロックチェーンノード１０４が１つのインスタンスを有効として受け入れ、次いで、別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した場合、そのブロックチェーンノード１０４は、これを受け入れなければならず、最初に受け入れたインスタンス（すなわち、ブロック１５１で公開されていないもの）を破棄する（すなわち、無効として扱う）。

【０２０３】
いくつか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得る。アカウントベースのケースでは、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における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、絶対アカウント残高を参照することによって転送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は、ブロックチェーンとは別個にそのネットワークのノードによって記憶され、絶えず更新される。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、送信者によってその暗号署名の一部として署名され、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、トランザクションにおけるオプションのデータフィールドも署名され得る。このデータフィールドは、例えば、前のトランザクションＩＤがデータフィールドに含まれている場合、前のトランザクションを指し示し得る。

【０２０４】
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一例である。トランザクション１５２（「Ｔｘ」と略記される）は、ブロックチェーン１５０の基本的なデータ構造である（各ブロック１５１は１つまたは複数のトランザクション１５２を含む）。以下では、出力ベースまたは「ＵＴＸＯ」ベースのプロトコルを参照して説明する。しかしながら、これはすべての可能な実施形態に限定されるものではない。例示的なＵＴＸＯベースのプロトコルは、ビットコインを参照して説明されるが、他の例示的なブロックチェーンネットワーク上でも等しく実装され得ることに留意されたい。

【０２０５】
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を含むデータ構造を含む。各出力２０３は、未使用トランザクション出力（ＵＴＸＯ）を含み得、これは、（ＵＴＸＯがまだ償還されていない場合）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、分散型台帳上のトークンの設定数を表す。ＵＴＸＯはまた、他の情報の中でも、元とな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、入力フィールド（複数可）２０２および出力フィールド（複数可）２０３のサイズを示すインジケータを含み得るヘッダ２０１も含み得る。ヘッダ２０１はまた、トランザクションのＩＤを含み得る。実施形態では、トランザクションＩＤは、（トランザクションＩＤ自体を除く）トランザクションデータのハッシュであり、ノード１０４にサブミット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。

【０２０６】
アリス１０３ａが、当該デジタル資産の額をボブ１０３ｂに転送するトランザクション１５２ｊを作成することを望むとする。図２では、アリスの新しいトランザクション１５

２ｊは「Ｔｘ$_1$」とラベル付けされている。これは、シーケンス内の先行するトランザクション１５２ｉの出力２０３においてアリスにロックされたデジタル資産の額をとり、これのうちの少なくとも一部をボブに転送する。先行するトランザクション１５２ｉは、図２では「Ｔｘ$_0$」とラベル付けされている。Ｔｘ$_0$およびＴｘ$_1$は、単なる任意のラベルである。それらは、Ｔｘ$_0$がブロックチェーン１５１内の最初のトランザクションであること、またはＴｘ$_1$がプール１５４内のすぐ次のトランザクションであることを必ずしも意味するものではない。Ｔｘ$_1$は、アリスにロックされた未使用出力２０３を依然として有する任意の先行する（すなわち先の）トランザクションを指し示すことができる。

【０２０７】

先行するトランザクションＴｘ$_0$は、アリスが新しいトランザクションＴｘ$_1$を作成した時点では、または少なくともアリスがそれをネットワーク１０６に送信する時点までには、すでに妥当性確認されブロックチェーン１５０のブロック１５１に含まれている可能性がある。それは、その時点でブロック１５１のうちの１つにすでに含まれていてもよいし、順序付きセット１５４で依然として待機していてもよく、このケースでは、すぐに新しいブロック１５１に含まれることになる。代替的に、Ｔｘ$_0$およびＴｘ$_1$を作成してネットワーク１０６に一緒に送信することもできるし、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可能にする場合には、Ｔｘ$_0$をＴｘ$_1$の後に送信することさえもできる。トランザクションのシーケンスの文脈において本明細書で使用される「先行する」および「後続する」という用語は、トランザクション内で指定されているトランザクションポインタ（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）によって定義されるシーケンス内のトランザクションの順序を指す。それらは、同様に、「先行するもの（predecessor）」および「後続するもの（successor）」、または「先の（antecedent）」および「後の（descendant）」、「親（parent）」および「子（child）」などと置き換えられ得る。これは、それらの作成、ネットワーク１０６への送信、または任意の所与のブロックチェーンノード１０４への到着の順序を必ずしも意味するものではない。それでも、先行するトランザクション（先のトランザクションまたは「親」）を指し示す後続するトランザクション（後の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に到着する子は、オーファンとみなされる。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、親を待つために特定の時間バッファされるかまたは破棄され得る。

【０２０８】

先行するトランザクションＴｘ$_0$の１つまたは複数の出力２０３のうちの１つは、本明細書ではＵＴＸＯ$_0$とラベル付けされた特定のＵＴＸＯを含む。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、ロックスクリプトとを含み、ロックスクリプトは、後続するトランザクションが妥当性確認され、したがってＵＴＸＯが正常に償還されるために、後続するトランザクションの入力２０２内のロック解除スクリプトが満たさなければならない条件を定義する。典型的には、ロックスクリプトは、その額を特定の当事者（それ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、ロックスクリプトは、典型的には、後続するトランザクションの入力内のロック解除スクリプトに、先行するトランザクションがロックされる当事者の暗号署名が含まれるという条件を含むロック解除条件を定義する。

【０２０９】

ロックスクリプト（通称scriptPubKey）は、ノードプロトコルによって認識されるドメイン固有言語で書かれたコードの一部である。そのような言語の特定の例は、ブロックチェーンネットワークによって使用される「スクリプト（Script）」（大文字Ｓ）と呼ばれる。ロックスクリプトは、トランザクション出力２０３を使用するためにどの情報が必要とされるか、例えばアリスの署名の必要性、を指定する。ロック解除スクリプトはトランザクションの出力に現れる。ロック解除スクリプト（通称scriptSig）は、ロックスクリプト基準を満たすのに必要な情報を提供するドメイン固有言語で書かれたコードの

一部である。例えば、それはボブの署名を含み得る。ロック解除スクリプトは、トランザクションの入力２０２に現れる。

【０２１０】

つまり、図示の例では、Tx_0の出力２０３内の$UTXO_0$は、$UTXO_0$が償還されるために（厳密には、$UTXO_0$を償還しようとする後続するトランザクションが有効となるために）アリスの署名Sig P_Aを必要とするロックスクリプト[Checksig P_A]を含む。[Checksig P_A]は、アリスの公開鍵－私有鍵ペアの公開鍵P_Aの表現（すなわち、ハッシュ）を含む。Tx_1の入力２０２は、（例えば、実施形態ではトランザクションTx_0全体のハッシュであるそのトランザクションＩＤ、$TxID_0$によって）Tx_1を指し示すポインタを含む。Tx_1の入力２０２は、Tx_0の任意の他の可能な出力の中から$UTXO_0$を識別するために、Tx_0内の$UTXO_0$を識別するインデックスを含む。Tx_1の入力２０２は、アリスが鍵ペアのアリスの私有鍵をデータの所定の部分（暗号では「メッセージ」と呼ばれることもある）に適用することによって作成された、アリスの暗号署名を含むロック解除スクリプト〈Sig P_A〉をさらに含む。有効な署名を提供するためにアリスによって署名される必要があるデータ（または「メッセージ」）は、ロック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。

【０２１１】

新しいトランザクションTx_1がブロックチェーンノード１０４に到着すると、ノードはノードプロトコルを適用する。これは、ロックスクリプトおよびロック解除スクリプトを一緒に実行して、ロック解除スクリプトがロックスクリプトで定義されている条件（この条件は１つまたは複数の基準を含み得る）を満たすかどうかをチェックすることを含む。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含む：

〈Sig P_A〉 〈P_A〉 ¦¦ [Checksig P_A]

ここで、「¦¦」は連結を表し、「〈…〉」はデータをスタックに置くことを意味し、「[…]」はロックスクリプト（この例ではスタックベースの言語）で構成される関数である。同等に、スクリプトは、スクリプトを連結するのではなく、共通スタックを用いて次々に実行され得る。いずれにしても、一緒に実行されると、スクリプトは、Tx_0の出力内のロックスクリプトに含まれるようなアリスの公開鍵P_Aを使用して、Tx_1の入力内のロック解除スクリプトが、データの予想される部分に署名するアリスの署名を含むことを認証する。この認証を実行するためには、データの予想される部分自体（「メッセージ」）も含まれる必要がある。実施形態では、署名されるデータはTx_1の全体を含む（つまり、平文のデータの署名された部分を指定する別個の要素は、すでに本質的に存在するので、含まれる必要がない）。

【０２１２】

公開－秘密暗号法による認証の詳細は、当業者によく知られている。基本的に、アリスが自身の私有鍵を使用してメッセージに署名した場合、アリスの公開鍵および平文のメッセージが与えられると、ノード１０４などの別のエンティティは、メッセージがアリスによって署名されたものに違いないことを認証することができる。署名は、典型的には、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にタグ付けすることを含み、これにより、公開鍵の任意の保持者が署名を認証することができるようになる。したがって、データの特定の部分またはトランザクションの一部などに署名することへの本明細書におけるいかなる参照も、実施形態では、データのその部分またはトランザクションの一部のハッシュに署名することを意味し得ることに留意されたい。

【０２１３】

Tx_1内のロック解除スクリプトが、Tx_0のロックスクリプト内で指定されている１つまたは複数の条件を満たす場合（つまり、図示の例では、アリスの署名がTx_1内で提供され、認証された場合）、ブロックチェーンノード１０４は、Tx_1が有効であるとみなす。これは、ブロックチェーンノード１０４が、保留中のトランザクションの順序付きプール１５４にTx_1を追加することとなることを意味する。ブロックチェーンノード１

０４はまた、トランザクションTx_1をネットワーク１０６内の１つまたは複数の他のブロックチェーンノード１０４にフォワードして、トランザクションTx_1がネットワーク１０６全体に伝搬されるようにする。Tx_1が妥当性確認されてブロックチェーン１５０に含まれると、これは、Tx_0からの$UTXO_0$を使用済みとして定義する。Tx_1は、未使用トランザクション出力２０３を使用する場合にのみ有効になり得ることに留意されたい。別のトランザクション１５２によってすでに使用された出力を使用しようとする場合、Tx_1は、他のすべての条件が満たされたとしても無効になる。したがって、ブロックチェーンノード１０４はまた、先行するトランザクションTx_0内の参照されたＵＴＸＯがすでに使用済みであるかどうか（すなわち、それが別の有効なトランザクションへの有効な入力をすでに形成したかどうか）をチェックする必要がある。これは、ブロックチェーン１５０がトランザクション１５２に定義された順序を課すことが重要である理由の１つである。実際には、所与のブロックチェーンノード１０４は、どのトランザクション１５２内のどのＵＴＸＯ２０３が使用されたかをマーキングする別個のデータベースを維持し得るが、最終的には、ＵＴＸＯが使用されたかどうかを定義するものは、ブロックチェーン１５０内の別の有効なトランザクションへの有効な入力をすでに形成しているかどうかである。

【０２１４】

所与のトランザクション１５２のすべての出力２０３において指定された総額が、そのすべての入力２０２によって指し示された総額よりも大きい場合、これは、ほとんどのトランザクションモデルにおいて無効性の別の根拠となる。そのため、そのようなトランザクションは、伝搬されることも、ブロック１５１に含まれることもない。

【０２１５】

ＵＴＸＯベースのトランザクションモデルでは、所与のＵＴＸＯが全体として使用される必要があることに留意されたい。ＵＴＸＯにおいて使用済みとして定義された額の一部は、別の一部が使用されている間、「残す」ことはできない。しかしながら、次のトランザクションの複数の出力間でＵＴＸＯからの額を分割することはできる。例えば、Tx_0内の$UTXO_0$において定義された額を、Tx_1内の複数のＵＴＸＯ間で分割することができる。したがって、アリスが、$UTXO_0$において定義された額のすべてをボブに与えたくない場合、アリスは、リマインダを使用して、Tx_1の第２の出力において自分自身に残りを与えるか、または別の当事者に支払うことができる。

【０２１６】

実際には、アリスはまた、通常、アリスのトランザクション１０４をブロック１５１に成功裏に含めるビットコインノード１０４に対する手数料を含める必要がある。アリスがそのような手数料を含めない場合、Tx_0は、ブロックチェーンノード１０４によって拒否され得、したがって、技術的に有効であっても、伝搬されず、ブロックチェーン１５０に含まれない可能性がある（ノードプロトコルは、ブロックチェーンノード１０４が望まない場合にトランザクション１５２を受け入れることを強制しない）。いくつかのプロトコルでは、トランザクション手数料は、それ自体の別個の出力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、所与のトランザクション１５２の入力（複数可）２０２によって指し示される総額と出力（複数可）２０３で指定されている総額との間の任意の差が、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。例えば、$UTXO_0$へのポインタがTx_1への唯一の入力であり、Tx_1は唯一の出力$UTXO_1$を有するとする。$UTXO_0$において指定されたデジタル資産の額が$UTXO_1$において指定された額よりも大きい場合、その差分は、$UTXO_1$を含むブロックを作成するためのプルーフオブワーク競争に勝つノード１０４によって割り当てられ得る（または、使用され得る）。しかしながら、代替的または追加的に、トランザクション手数料がトランザクション１５２のＵＴＸＯ２０３のうちのそれ自体の１つにおいて明示的に指定され得ることは必ずしも除外されない。

【０２１７】

アリスおよびボブのデジタル資産は、ブロックチェーン１５０内のどこかで任意のトラ

ンザクション１５２においてそれらにロックされたＵＴＸＯから構成される。したがって、典型的には、所与の当事者１０３の資産は、ブロックチェーン１５０全体にわたる様々なトランザクション１５２のＵＴＸＯ全体に散在している。ブロックチェーン１５０内のどこにも、所与の当事者１０３の総残高を定義する数字は記憶されない。クライアントアプリケーション１０５におけるウォレット機能の役割は、それぞれの当事者にロックされ、別の以降のトランザクションでまだ使用されていない様々なＵＴＸＯのすべての値を一緒に照合することである。これは、ビットコインノード１０４のいずれかに記憶されたブロックチェーン１５０のコピーにクエリを行うことによって行うことができる。

【０２１８】

スクリプトコードは、多くの場合、概略的に（すなわち、正確な言語を使用せずに）表されることに留意されたい。例えば、特定の機能を表すためにオペレーションコード（オペコード）が使用され得る。「OP_...」は、スクリプト言語の特定のオペコードを指す。例として、OP_RETURNは、ロックスクリプトの最初にOP_FALSEが先行するときに、トランザクション内にデータを記憶することができ、それによってデータをブロックチェーン１５０内に不変的に記録することができる、トランザクションの使用不可能な出力を作成するスクリプト言語のオペコードである。例えば、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含み得る。

【０２１９】

典型的には、トランザクションの入力は、公開鍵P_Aに対応するデジタル署名を含む。実施形態において、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、データの特定の一部に署名する。いくつかの実施形態では、所与のトランザクションについて、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全部に署名する。署名された出力の特定の部分は、ＳＩＧＨＡＳＨフラグに依存する。ＳＩＧＨＡＳＨフラグは、通常、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４バイトコードである（したがって、署名時に固定される）。

【０２２０】

ロックスクリプトは、典型的には、それぞれのトランザクションがロックされる当事者の公開鍵を含むという事実を指して、「scriptPubKey」と呼ばれることがある。ロック解除スクリプトは、典型的には、それが対応する署名を供給するという事実を指して「scriptSig」と呼ばれることがある。しかしながら、より一般的には、ＵＴＸＯが償還されるための条件が署名を認証することを含むことは、ブロックチェーン１５０のすべてのアプリケーションにおいて必須ではない。より一般的には、スクリプト言語を使用して、任意の１つまたは複数の条件を定義することができる。したがって、より一般的な用語「ロックスクリプト」および「ロック解除スクリプト」が好まれ得る。

【０２２１】

サイドチャネル

図１に示すように、アリスおよびボブのコンピュータ機器１０２ａ、１２０ｂの各々上のクライアントアプリケーションは、それぞれ、追加の通信機能を含み得る。この追加の機能により、（いずれかの当事者または第三者の扇動で）アリス１０３ａは、ボブ１０３ｂと別個のサイドチャネル１０７を確立することができる。サイドチャネル１０７は、ブロックチェーンネットワークとは別での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。例えば、これは、当事者の一方がトランザクションをネットワーク１０６にブロードキャストすることを選択するまで、トランザクションが（まだ）ブロックチェーンネットワーク１０６に登録されることなく、またはチェーン１５０上に進むことなく、アリスとボブ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てもよい。代替的または追加的に、サイドチャネル１０７は、鍵、交渉された額または条件、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。

【０２２２】

サイドチャネル１０７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。代替的または追加的に、サイドチャネル３０１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、またはアリスのデバイス１０２ａとボブのデバイス１０２ｂとの間の直接のワイヤードまたはワイヤレスリンクを介して確立され得る。一般に、本明細書のどこでも、参照されるサイドチャネル１０７は、「オフチェーン」すなわちブロックチェーンネットワーク１０６とは別で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含み得る。２つ以上のリンクが使用される場合、全体としてのオフチェーンリンクの束または集合は、サイドチャネル１０７と呼ばれることがある。したがって、アリスおよびボブがサイドチャネル１０７上で情報またはデータの特定の部分などを交換すると言われている場合、これは、これらのデータの部分のすべてが全く同じリンクまたは同じタイプのネットワーク上で送信されなければならないことを必ずしも意味するものではないことに留意されたい。

【０２２３】

さらなる備考

開示される技法の他の変形形態またはユースケースは、本明細書の開示が与えられると、当業者には明らかになり得る。本開示の範囲は、説明された実施形態によって限定されず、添付の特許請求の範囲によってのみ限定される。例えば、上記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、上記の説明は一般に任意のブロックチェーンに適用されてもよいことが理解されよう。すなわち、本発明はビットコインブロックチェーンに限定されるものではない。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への上記の任意の参照は、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への参照に置き換えられてもよい。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、上記で説明し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の説明された特性の一部またはすべてを共有してもよい。

【０２２４】

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、発行、伝搬、および記憶する説明した機能を少なくともすべて実行する。これらの機能のすべてではなく１つまたはいくつかのみ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）が存在し得ることは除外されない。すなわち、ネットワークエンティティは、ブロックを作成および発行することなしに、ブロックを伝搬および／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が好ましいビットコインネットワーク１０６のノードとみなされないことを想起されたい）。

【０２２５】

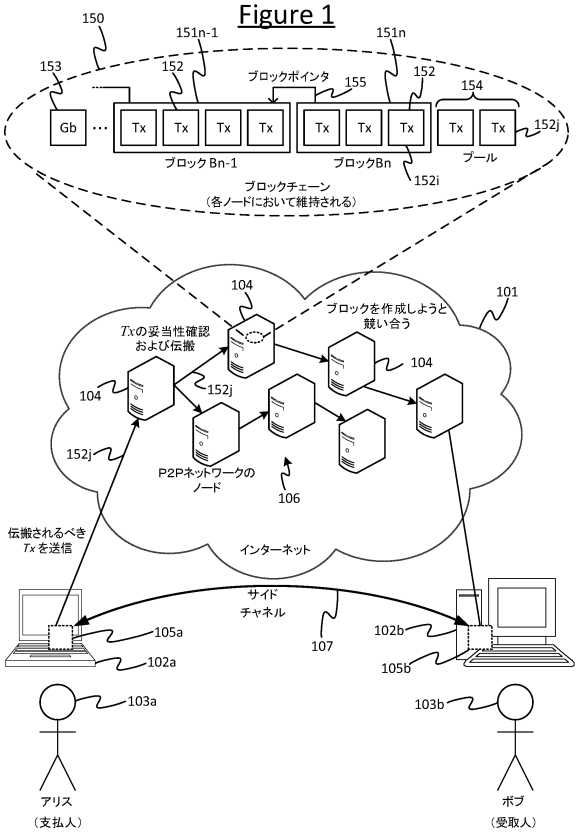
本発明の他の実施形態では、ブロックチェーンネットワーク１０６はビットコインネットワークでなくてもよい。これらの実施形態では、ノードが、ブロックチェーン１５０のブロック１５１を作成、発行、伝搬、および記憶する機能のうちの少なくとも１つまたはすべてではないがいくつかを実行してもよいことは除外されない。例えば、それらの他のブロックチェーンネットワーク上では、「ノード」は、ブロック１５１を作成および発行はするが、それらのブロック１５１を記憶および／または他のノードへの伝搬はしないように構成されたネットワークエンティティを指すために使用され得る。

【０２２６】
さらにより一般的には、上記の「ビットコインノード」１０４という用語へのいかなる言及も、「ネットワークエンティティ」または「ネットワーク要素」という用語と置き換えラレ絵、そのようなエンティティ／要素は、ブロックを作成、発行、伝搬、および記憶する役割の一部または全部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を参照して上記で説明したものと同じ方法でハードウェアに実装されてもよい。
【０２２７】
「ユーザ」という用語は、人間および機械ベースのエンティティを含むように本明細書で使用され得る。
【０２２８】
上述の実施形態は、本開示を限定するのではなく例示するものであり、当業者であれば、添付の特許請求の範囲によって定義される本開示の範囲から逸脱することなく、多くの代替実施形態を設計することができるであろう。特許請求の範囲において、括弧内に置かれた参照符号は、特許請求の範囲を限定するものとして解釈されるべきではない。「comprising」および「comprises」などの用語は、請求項または明細書全体に列挙されたもの以外の要素またはステップの存在を排除するものではない。本明細書において、「comprises（～を備える／含む）」は「includes or consists of（～を含むかまたは～から成る）」を意味し、「comprising（～を含んでいる）」は「including or consisting of（～を含んでいるかまたは～から成っている）」を意味する。本明細書全体を通して、「comprise（～を含む）」という単語、または「includes（～を含む）」、「comprises」もしくは「comprising」などの変形は、述べられた要素、整数もしくはステップ、または要素、整数もしくはステップのグループを包含することを意味し、任意の他の要素、整数もしくはステップ、または要素、整数もしくはステップのグループを除外することを意味するものではないことが理解されよう。要素の単数の参照は、そのような要素の複数の参照を除外するものではなく、逆もまた同様である。本開示は、いくつかの別個の要素を備えるハードウェアによって、および適切にプログラムされたコンピュータによって実装され得る。いくつかの手段を列挙する装置請求項では、これらの手段のいくつかは、ハードウェアの全く同一のアイテムによって具現化され得る。特定の手段が互いに異なる従属請求項に記載されているという事実だけでは、これらの手段の組合せが有利に使用できないことを示さない。

【図面】

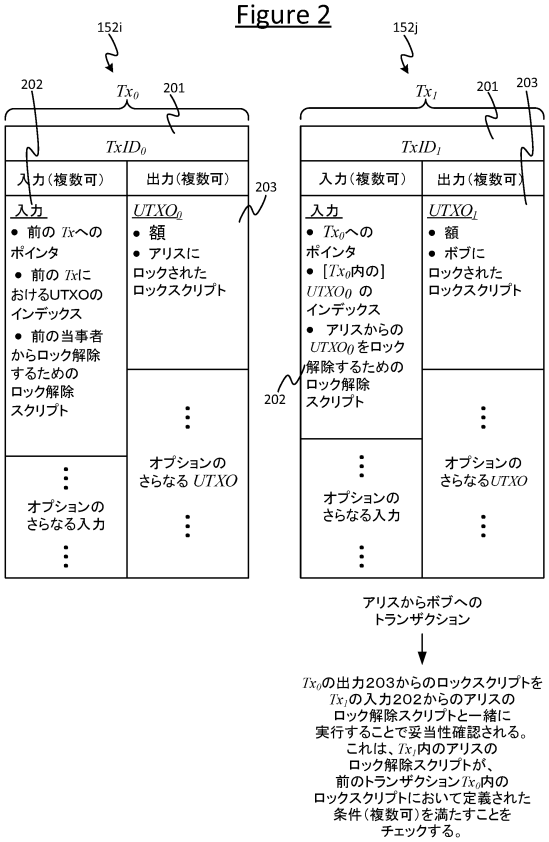
【図１】

Figure 1



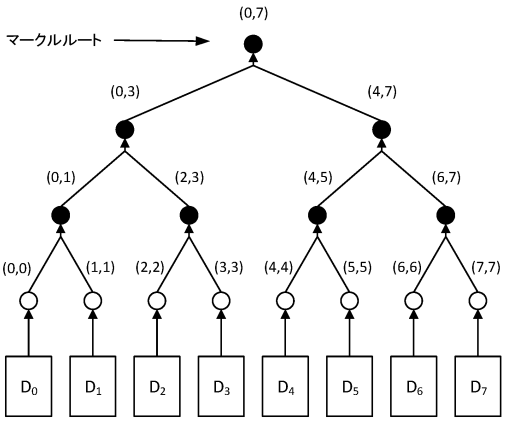
【図２】

Figure 2



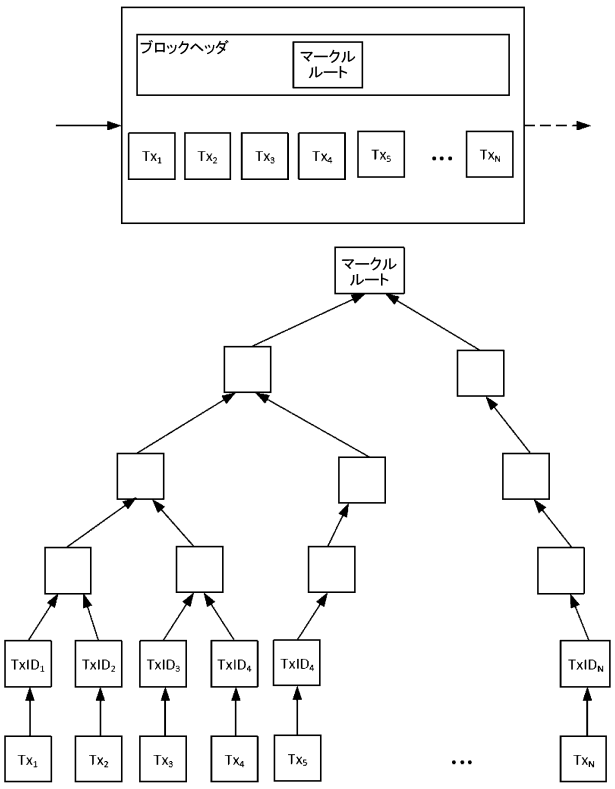
【図３】

Figure 3



【図４】

Figure 4



【図５】

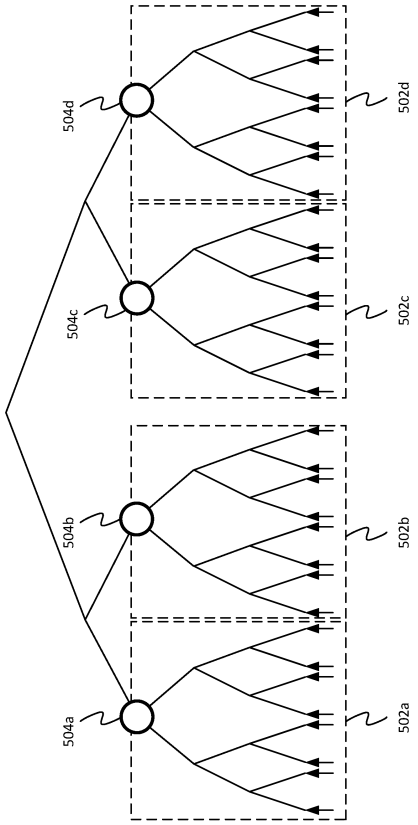


Figure 5

【図６】

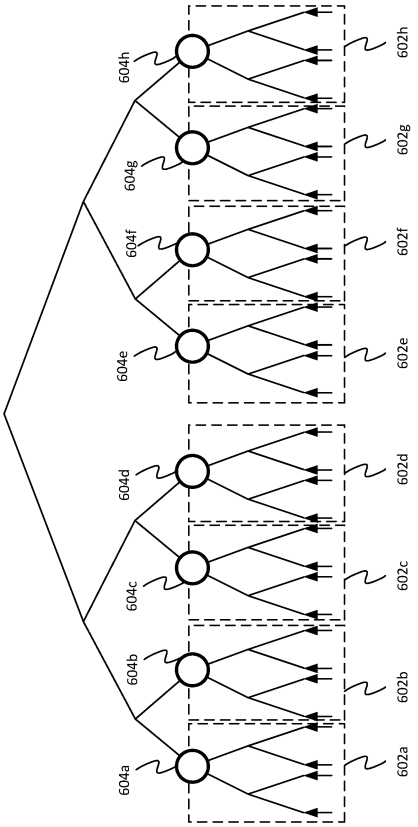
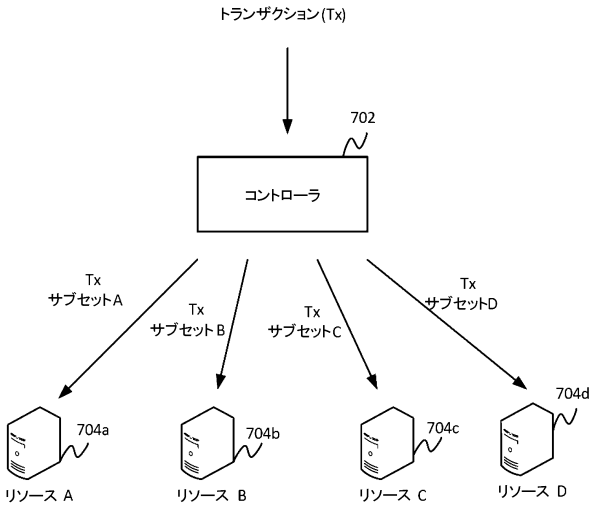


Figure 6

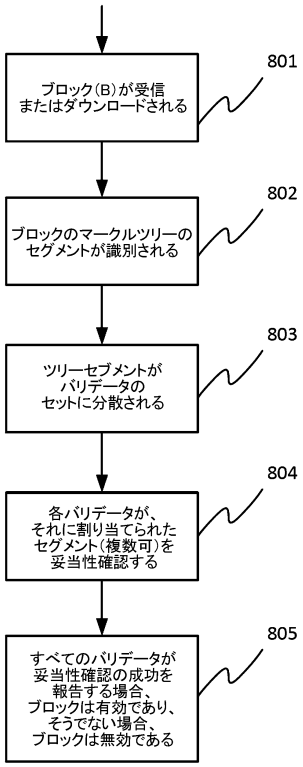
【図７】

Figure 7



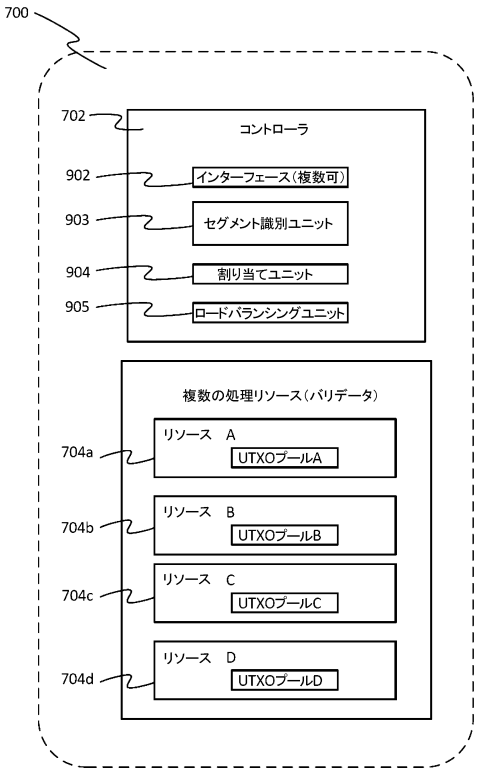
【図８】

Figure 8



【図９】

Figure 9


700
702
コントローラ
902
インターフェース（複数可）
903
セグメント識別ユニット
904
割り当てユニット
905
ロードバランシングユニット
複数の処理リソース（パリデータ）
704a
リソース A
UTXOプールA
704b
リソース B
UTXOプールB
704c
リソース C
UTXOプールC
704d
リソース D
UTXOプールD

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2022/079825

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L9/00 G06F21/64
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 111 581 214 A (CHENGDU HANWEI SCIENCE & TECH CO LTD) 25 August 2020 (2020-08-25) paragraph [0051] - paragraph [0055] paragraph [0018] - paragraph [0024] claim 4 figure 1 ----- -/--	1-13

[x] Further documents are listed in the continuation of Box C. [x] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance;; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance;; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
23 January 2023	31/01/2023

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Dobre, Dan

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2022/079825

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2021/273807 A1 (WERTHEIM ODED [IL] ET AL) 2 September 2021 (2021-09-02) paragraph [0082] paragraph [0086] paragraph [0169] - paragraph [0171] paragraph [0241] paragraph [0240] paragraph [0056] paragraph [0085] paragraph [0166] figures 1B,2,3,8 paragraph [0168] paragraph [0178] - paragraph [0179] paragraph [0084] -----	1-13

Form PCT/ISA/210 (continuation of second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2022/079825

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
CN 111581214 A	25-08-2020	NONE	
US 2021273807 A1	02-09-2021	US 2021273807 A1	02-09-2021
		WO 2020033216 A2	13-02-2020

Form PCT/ISA/210 (patent family annex) (April 2005)

フロントページの続き

MK,MT,NL,NO,PL,PT,RO,RS,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,KM,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BN,BR,BW,BY,BZ,CA,CH,CL,CN,CO,CR,CU,CV,CV,CZ,DE,DJ,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IQ,IR,IS,IT,JM,JO,JP,KE,KG,KH,KN,KP,KR,KW,KZ,LA,LC,LK,LR,LS,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PA,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SA,SC,SD,SE,SG,SK,SL,ST,SV,SY,TH,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,WS,ZA,ZM,ZW