(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7319461号
(P7319461)

(45)発行日　令和5年8月1日(2023.8.1)

(24)登録日　令和5年7月24日(2023.7.24)

(51)国際特許分類
G06F 21/60 (2013.01)

FI
G06F 21/60

請求項の数　15　（全31頁）

(21)出願番号　特願2022-514455(P2022-514455)
(86)(22)出願日　令和2年3月30日(2020.3.30)
(65)公表番号　特表2022-547853(P2022-547853A)
(43)公表日　令和4年11月16日(2022.11.16)
(86)国際出願番号　PCT/US2020/025780
(87)国際公開番号　WO2021/201827
(87)国際公開日　令和3年10月7日(2021.10.7)
審査請求日　令和4年3月2日(2022.3.2)

(73)特許権者　000005108
株式会社日立製作所
東京都千代田区丸の内一丁目６番６号
(74)代理人　110001689
青稜弁理士法人
(72)発明者　下沢 拓
アメリカ合衆国 カリフォルニア州、サンタクララ市 オースティン ドライブ ２５３５ 日立アメリカリミテッド内
(72)発明者　大島 訓
東京都千代田区丸の内一丁目６番６号
株式会社日立製作所内
審査官　局 成矢

最終頁に続く

(54)【発明の名称】　コンソーシアムブロックチェーンを用いてプライベートデータを保持する方法および装置

(57)【特許請求の範囲】

【請求項１】

スマートコントラクトトランザクションを容易にするように構成された、複数のノードを含むブロックチェーンシステムのための方法であって、

前記複数のノードのうちの第１のノードにおいて、前記複数のノードのうちの第２のノードから前記スマートコントラクトトランザクションの提案を受信することを契機に、

前記第１のノードにおいて、前記第２のノードに関連付けられたブロックチェーンシステム内の組織を特定することと、

前記第１のノードにおいて、前記組織が前記提案によって示される前記スマートコントラクトトランザクションを実行できるか否かを判定することと、

前記組織が前記提案によって示された前記スマートコントラクトトランザクションを実行する権限を有していないことを示す判定に対して、前記第２のノードにエラーを返すことと、

前記組織が前記提案によって示された前記スマートコントラクトトランザクションを実行する権限を有していることを示す判定に対して、前記スマートコントラクトトランザクションを実行することと、

を含み、

前記複数のノードのうちの、前記スマートコントラクトトランザクションを実行する権限を有しておらず、且つ、他の組織のプライベートデータにアクセスする資格を持つように構成された読み取り専用組織に属する特定のノードにおいて、前記プライベートデータを

含む前記スマートコントラクトトランザクションのエンドースを行う、
方法。
【請求項２】
請求項１に記載の方法において、
前記第１のノードにおいて、前記組織が前記提案によって示される前記スマートコントラクトトランザクションを実行できるか否かを判定することにおいて、
前記組織が前記読み取り専用組織であるために前記組織が前記提案によって示される前記スマートコントラクトトランザクションを実行する権限を有していないことを示す判定を行うことと、
前記組織が前記読み取り専用組織ではないために前記組織が前記提案によって示される前記スマートコントラクトトランザクションを実行する権限を有することを示す判定を行うことと、
を含む、
方法。
【請求項３】
請求項１に記載の方法において、
前記第１のノードにおいて、前記第２のノードから前記スマートコントラクトトランザクションのコミットおよびバリデーティングの要求を受信したことを契機に、
前記第１のノードにおいて、前記第２のノードに関連付けられたブロックチェーンシステム内の組織を特定することと、
前記第１のノードにおいて、前記組織が前記提案によって示される前記スマートコントラクトトランザクションを実行できるか否かを判定することと、
前記組織が前記提案によって示された前記スマートコントラクトトランザクションを実行する権限を有していないことを示す判定に対して、前記第２のノードにエラーを返すことと、
前記組織が前記提案によって示された前記スマートコントラクトトランザクションを実行する権限を有していることを示す判定に対して、オーダリングサービスによって提供されるトランザクションの順序を実行することと、
を含む、
方法。
【請求項４】
請求項３に記載の方法において、
前記ブロックチェーンシステムに発行されたトランザクションのためのオーダリングサービスを実行することを更に含み、
前記オーダリングサービスは、
前記第２のノードからの前記スマートコントラクトトランザクションをコミットするための要求を実行するための前記トランザクションの順序を決定することと、
前記トランザクションの順序を前記第１のノードに提供することと、
を含む、
方法。
【請求項５】
請求項１に記載の方法であって、
前記複数のノードのうちの第３のノードで監査プロセスを実行することをさらに含み、
前記監査プロセスは、
前記第３のノードで、前記スマートコントラクトトランザクションおよび前記スマートコントラクトトランザクションに関連するプライベートデータを受信することと、
前記スマートコントラクトトランザクション及び前記プライベートデータの整合性チェックを実行することと、
を含む、
方法。

【請求項６】
スマートコントラクトトランザクションを容易にするように構成された、複数のノードを含むブロックチェーンシステムのための方法であって、
前記複数のノードのうちの第１のノードにおいて、前記複数のノードのうちの第２のノードから前記スマートコントラクトトランザクションの提案を受信することを契機に、
前記第１のノードにおいて、前記第２のノードに関連付けられたブロックチェーンシステム内の組織を特定することと、
前記第１のノードにおいて、前記組織が前記提案によって示される前記スマートコントラクトトランザクションを実行できるか否かを判定することと、
前記組織が前記提案によって示された前記スマートコントラクトトランザクションを実行する権限を有していないことを示す判定に対して、前記第２のノードにエラーを返すことと、
前記組織が前記提案によって示された前記スマートコントラクトトランザクションを実行する権限を有していることを示す判定に対して、前記スマートコントラクトトランザクションを実行することと、
を含み、
テスト環境の開始のために、
前記スマートコントラクトトランザクションの提案を開始した前記ブロックチェーンシステム内の組織のアイデンティティ及びエンドースメントを、前記テスト環境を管理するテスト環境マネージャが発行したアイデンティティ及びエンドースメントに置き換えることと、
前記ブロックチェーンシステムの構成の変更を示す前記スマートコントラクトトランザクションのために、前記スマートコントラクトトランザクションの組織情報によって示される前記ブロックチェーンシステムの前記組織のトラストルートを変換することと、
を含む、
方法。
【請求項７】
請求項６に記載の方法において、
前記テスト環境上で新しいブロックチェーンネットワークを起動することをさらに含み、
前記起動は、テストネットワークランチャーによって容易に行われ、
前記起動は、
前記テスト環境マネージャによって変換された台帳と、前記テスト環境マネージャによって変換されたステートデータベースとを、新しいブロックチェーンネットワークの各ブロックチェーンノードにコピーすることと、
前記スマートコントラクトトランザクションに関連付けられた前記ブロックチェーンシステムの前記組織に対応する前記テスト環境マネージャによって変換されたプライベートデータを、前記組織に対応する前記新しいブロックチェーンネットワークの各ブロックチェーンノードにコピーすることと、
を含む、
方法。
【請求項８】
スマートコントラクトトランザクションを容易にするように構成されたブロックチェーンシステムであって、
前記ブロックチェーンシステムは、複数のノードを含み、
前記複数のノードのうちの第１のノードは、プロセッサを含み、
前記第１のノードの前記プロセッサは、前記第１のノードが、前記複数のノードの第２のノードから前記スマートコントラクトトランザクションの提案を受信することを契機に、
前記第２のノードに関連するブロックチェーンシステム内の組織を特定し、
前記組織が前記提案で示された前記スマートコントラクトトランザクションを実行できるか否かを判定し、

前記組織が前記提案で示された前記スマートコントラクトトランザクションを実行する権限を有していないことを示す判定に対して、前記第２のノードにエラーを返し、
前記組織が前記提案によって示された前記スマートコントラクトトランザクションを実行する権限を有していることを示す判定に対して、前記スマートコントラクトトランザクションを実行する
ように構成され、
前記複数のノードのうちの、前記スマートコントラクトトランザクションを実行する権限を有しておらず、且つ、他の組織のプライベートデータにアクセスする資格を持つように構成された読み取り専用組織に属する特定のノードは、プロセッサを含み、
前記特定のノードの前記プロセッサは、前記プライベートデータを含む前記スマートコントラクトトランザクションのエンドースを行う、
ように構成された、
ブロックチェーンシステム。

【請求項９】
請求項８に記載のブロックチェーンシステムにおいて、
前記第１のノードの前記プロセッサは、
前記組織が前記読み取り専用組織であるために前記組織が前記提案によって示される前記スマートコントラクトトランザクションを実行する権限を有していないことを示すと判定することと、前記組織が読み取り専用組織ではないために前記組織が前記提案によって示される前記スマートコントラクトトランザクションを実行する権限を有することを示すと判定することにより、
前記組織が前記提案によって示される前記スマートコントラクトトランザクションを実行できるか否かを判定する、
ように構成された、
ブロックチェーンシステム。

【請求項１０】
請求項８に記載のブロックチェーンシステムにおいて、
前記第１のノードの前記プロセッサは、前記第２のノードから前記スマートコントラクトトランザクションのコミットおよびバリデーティングの要求を受信することを契機に、
前記第２のノードに関連付けられたブロックチェーンシステム内の前記組織を特定し、
前記組織が前記提案によって示された前記スマートコントラクトトランザクションを実行できるか否かを判定し、
前記組織が前記提案によって示された前記スマートコントラクトトランザクションを実行する権限を有していないことを示す判定に対して、前記第２のノードにエラーを返し、
前記組織が、前記提案によって示された前記スマートコントラクトトランザクションを実行する権限を有していると判定した場合、オーダリングサービスによって提供されるトランザクションの順序を実行する、
ように構成された、
ブロックチェーンシステム。

【請求項１１】
請求項１０に記載に記載のブロックチェーンシステムにおいて、
他のプロセッサを含む前記オーダリングサービスを更に備え、
前記他のプロセッサは、前記第２のノードからの前記スマートコントラクトトランザクションをコミットするための要求を実行するための前記トランザクションの順序を決定し、決定した前記トランザクションの順序を前記第１のノードに提供するように構成された、
ブロックチェーンシステム。

【請求項１２】
請求項８に記載のブロックチェーンシステムにおいて、
前記複数のノードのうちの第３のノードは、他のプロセッサを含み、
前記他のプロセッサは、

前記スマートコントラクトトランザクションと、前記スマートコントラクトトランザクションに関連するプライベートデータを受信することと、
前記スマートコントラクトトランザクションと前記プライベートデータの整合性チェックを実行することと、を含む監査プロセスを、
実行するように構成された、
ブロックチェーンシステム。
【請求項１３】
スマートコントラクトトランザクションを容易にするように構成されたブロックチェーンシステムであって、
前記ブロックチェーンシステムは、複数のノードを含み、
前記複数のノードのうちの第１のノードは、プロセッサを含み、
前記プロセッサは、前記第１のノードが、前記複数のノードの第２のノードから前記スマートコントラクトトランザクションの提案を受信することを契機に、
前記第２のノードに関連するブロックチェーンシステム内の組織を特定し、
前記組織が前記提案で示された前記スマートコントラクトトランザクションを実行できるか否かを判定し、
前記組織が前記提案で示された前記スマートコントラクトトランザクションを実行する権限を有していないことを示す判定に対して、前記第２のノードにエラーを返し、
前記組織が前記提案によって示された前記スマートコントラクトトランザクションを実行する権限を有していることを示す判定に対して、前記スマートコントラクトトランザクションを実行する
ように構成され、
読み取り専用組織に属する装置は、他のプロセッサを備え、
前記他のプロセッサは、
テスト環境の開始を受信するために、
前記スマートコントラクトトランザクションの前記提案を開始したブロックチェーンシステム内の組織のアイデンティティ及びエンドースメントを、前記テスト環境を管理するテスト環境マネージャが発行したアイデンティティ及びエンドースメントに置き換え、
システムの構成の変更を示す前記スマートコントラクトトランザクションに対して、前記スマートコントラクトトランザクションの組織情報によって示されるブロックチェーンシステム内の組織のトラストルートを変換する、
ように構成された、
ブロックチェーンシステム。
【請求項１４】
請求項１３に記載のブロックチェーンシステムにおいて、
前記他のプロセッサは、
テスト環境上で新しいブロックチェーンネットワークを起動するように構成され、前記起動はテストネットワークランチャーによって容易にされ
前記起動は、
前記テスト環境マネージャによって変換された台帳と、前記テスト環境マネージャによって変換されたステートデータベースを、新しいブロックチェーンネットワークの各ブロックチェーンノードにコピーすることと、
前記スマートコントラクトトランザクションに関連付けられた前記ブロックチェーンシステムの前記組織に対応する前記テスト環境マネージャによって変換されたプライベートデータを、前記組織に対応する新しいブロックチェーンネットワークの各ブロックチェーンノードにコピーすることと、
を含む、
ブロックチェーンシステム。
【請求項１５】
スマートコントラクトトランザクションを容易にするように構成されたブロックチェー

ンシステムのための命令を格納した非一時的なコンピュータ可読媒体であって、
前記ブロックチェーンシステムは複数のノードを含み、
前記命令は、
前記複数のノードのうちの第１のノードにおいて、前記複数のノードのうちの第２のノードから前記スマートコントラクトトランザクションの提案を受信することを契機に、
前記第１のノードにおいて、前記第２のノードに関連付けられた前記ブロックチェーンシステム内の組織を特定することと、
前記第１のノードにおいて、前記組織が前記提案によって示される前記スマートコントラクトトランザクションを実行できるか否かを判定することと、
前記組織が前記提案によって示された前記スマートコントラクトトランザクションを実行する権限を持っていないことを示す判定に対して、前記第２のノードにエラーを返すステップと、
前記組織が前記提案によって示された前記スマートコントラクトトランザクションを実行する権限を有していることを示す判定に対して、前記スマートコントラクトトランザクションを実行することと、
を含み、
前記複数のノードのうちの、前記スマートコントラクトトランザクションを実行する権限を有しておらず、且つ、他の組織のプライベートデータにアクセスする資格を持つように構成された読み取り専用組織に属する特定のノードにおいて、前記プライベートデータを含む前記スマートコントラクトトランザクションのエンドースを行う、
コンピュータ可読媒体。

【発明の詳細な説明】
【技術分野】
【０００１】
本開示は、一般的にはブロックチェーンシステムに関し、より具体的には、コンソーシアムブロックチェーンを用いてプライベートデータを保持するためのシステムおよび方法に関する。
【背景技術】
【０００２】
米国特許出願公開第２０１９／０２５１５６号明細書に開示されているような関連技術の実装には、コンソーシアムブロックチェーン上でプライベートトランザクションを容易にするためのシステムおよび方法がある。トランザクションのプライベートデータは、特定のクライアントノードに記録され、データのハッシュ値はブロックチェーンネットワークに記録される。クライアントノードは、他のクライアントノードに対してそれぞれプライベートな通信チャネルを持っている。そのため、２つのクライアントノード間のプライベートトランザクションは、他のクライアントノードに公開することなく、その通信チャネルを使って発行することができる。クライアントノードは、ブロックチェーンネットワークに記録されているハッシュを参照することで、他の組織から渡されたプライベートトランザクションのデータを検証し、データが保存されてから改ざんされていないことを確認することができる。
【発明の概要】
【０００３】
しかしながら、関連技術の実装では、プライベートデータのいわゆるSPoT（single point of trust問題に対処する方法は開示されていない。
【０００４】
本明細書に記載された例示的な実装は、コンソーシアムブロックチェーンを用いたシステムおよび方法に関する。コンソーシアムブロックチェーンは、許可型ブロックチェーンまたはエンタープライズブロックチェーンとして知られており、一連の既知の参加者のみにアクセスを制限するように構成され、匿名ではなく、公開されない。コンソーシアムブロックチェーンでは、参加者も組織に分類され、すべての参加者は1つだけの組織に所属

している必要がある。ブロックチェーンは、特定の組織に所属する複数のコンピュータノードが接続され、相互に通信することで成り立っている。システム内で共有・分散されたデータは台帳と呼ばれ、各コンピュータノードはローカルに台帳を持っている。ブロックチェーンは、ブロックチェーンに参加しているすべてのコンピュータノードが、システム内の台帳に対して同じ内容を持つように管理している（つまり、すべてのブロックチェーンノードが台帳のレプリカを持ち、最終的に同じ内容を持つことになる。）。

【０００５】

ブロックチェーンで使用されるデータモデルには、UTXO（unspent transaction output）およびキー・バリュー・ペアが含まれる。台帳は通常、トランザクションのみを含み、各トランザクションは関連するデータ（UTXOではアドレス、key-valueではキーによって識別される）のみの値を含む。すべてのデータの最新の値を得るには、台帳のすべての履歴、つまりトランザクションをトラバースすることで可能ですが、ブロックチェーンのノードは、オプションとして、効率化のために最新データのスナップショットを持ち、新しいトランザクションごとにそれを更新することもできる。

【０００６】

ブロックチェーンシステムのコンピュータノードで同じ台帳を保持することを容易にするために、様々なコンセンサスアルゴリズムが使用される。一例として、Hyperledger Fabricの3フェーズのコンセンサスがあり、これにはエンドースメント、オーダリングおよびバリデーションが含まれる。エンドースメントフェーズでは、クライアントがトランザクション提案を送信することで、ブロックチェーンノードにトランザクションを要求する。ブロックチェーンノードは、トランザクション提案に従ってスマートコントラクトと呼ばれるプログラムを実行し、その結果をエンドースメントと呼ばれるノードの署名付きでクライアントに送り返す。オーダリングフェーズでは、クライアントは提案とエンドースメントを含むトランザクションを、トランザクションの順序を決定するために特化されたブロックチェーンノードに送信する。オーダリングサービスを提供するノードは、受信したトランザクションの順序を決定し、１つまたは複数のトランザクションを集約してブロックを作成する。バリデーションフェーズでは、オーダリングサービスがブロックチェーンノードにブロックを配信する。ブロックチェーンノードは、各ブロックの各トランザクションを検査し、エンドースメントが有効であるか、結果が他の「前」のトランザクションとコンフリクトしていないかをチェックする。すべてのチェックを通過したトランザクションは有効とみなされ、ブロックチェーンノードにスナップショットデータがある場合、トランザクションの結果に応じてスナップショットデータを更新する。それ以外の場合は、無効とみなされ、効果がない。以下の説明では、この３フェーズコンセンサスをベースにしているが、本実装例は特定のアルゴリズムに限定されるものではない。

【０００７】

本開示の態様は、スマートコントラクトトランザクションを容易にするように構成されたブロックチェーンシステムのための方法を含むことができ、ブロックチェーンシステムは、複数のノードを有し、本方法は、前記複数のノードからの第１のノードにおいて、前記複数のノードの第２のノードからスマートコントラクトトランザクションの提案を受信することを契機に、前記第１のノードにおいて、前記第２のノードに関連付けられたブロックチェーンシステム内の組織を特定することと、前記第１のノードにおいて、前記組織が前記提案によって示されるスマートコントラクトトランザクションを実行できるか否かを判定することと、前記組織が前記提案が示すスマートコントラクトトランザクションを実行する権限を有していないことを示す判定に対して、記第２のノードにエラーを返すことと、前記組織が前記プロポーザルが示すスマートコントラクトトランザクションを実行する権限を有していることを示す判定に対して、前記スマートコントラクトトランザクションを実行することと、を含む。

【０００８】

本開示の態様は、スマートコントラクトトランザクションを容易にするように構成されたブロックチェーンシステムのための命令を格納する、非一時的なコンピュータ可読媒体

を含むことができ、ブロックチェーンシステムは複数のノードを含み、命令は、前記複数のノードのうちの第１のノードにおいて、前記複数のノードのうちの第２のノードからスマートコントラクトトランザクションのプロポーザルを受信したことを契機に、前記第１のノードにおいて、前記第２のノードに関連付けられたブロックチェーンシステム内の組織を特定することと、前記第１のノードにおいて、前記組織が前記プロポーザルによって示されるスマートコントラクトトランザクションを実行できるか否かを判定することと前記組織が前記プロポーザルが示すスマートコントラクトトランザクションを実行する権限を有していないことを示す判定に対して、前記第２のノードにエラーを返すことと、前記組織が前記プロポーザルが示すスマートコントラクトトランザクションを実行する権限を有していることを示す判定に対して、前記スマートコントラクトトランザクションを実行することと、を含む。

【０００９】

本開示の態様は、スマートコントラクトトランザクションを容易にするように構成されたブロックチェーンシステムを含むことができ、ブロックチェーンシステムは、複数のノードを有し、ブロックチェーンシステムは、前記複数のノードのうちの第１のノードにおいて、前記複数のノードのうちの第２のノードからスマートコントラクトトランザクションの提案を受信したことを契機に、前記第１のノードにおいて、前記第２のノードに関連付けられたブロックチェーンシステム内の組織を特定する手段と、前記第１のノードにおいて、前記組織が前記提案によって示されるスマートコントラクトトランザクションを実行できるか否かを決定する手段と、前記組織が、前記プロポーザルが示すスマートコントラクトトランザクションを実行する権限を有していないことを示す判定に対して、前記第２のノードにエラーを返す手段と、前記組織が、前記プロポーザルが示すスマートコントラクトトランザクションを実行する権限を有していることを示す判定に対して、前記スマートコントラクトトランザクションを実行する手段と、を含む。

【００１０】

本開示の態様は、スマートコントラクトトランザクションを容易にするように構成されたブロックチェーンシステムを含むことができ、ブロックチェーンシステムは、複数のノードを含み、複数のノードの第１のノードは、プロセッサを含み、プロセッサは、複数のノードの第２のノードからスマートコントラクトトランザクションの提案を受信することを契機に、前記第２のノードに関連するブロックチェーンシステム内の組織を特定することと、前記組織が前記提案によって示されるスマートコントラクトトランザクションを実行する権限を有するか否かを判定することと、前記組織が前記提案によって示されるスマートコントラクトトランザクションを実行する権限を有していないことを示す判定に対して、前記第２のノードにエラーを返すことと、前記組織が前記提案によって示されるスマートコントラクトトランザクションを実行する権限を有することを示す判定に対して、前記スマートコントラクトトランザクションを実行することと、を含むように構成される。

【図面の簡単な説明】

【００１１】

【図１】図１は、実装例により実装するシステムの例を示す。

【００１２】

【図２】図２は、ブロックチェーンノードの実装例を示す。

【００１３】

【図３】図３は、ブロックおよびトランザクション提案の実装例を示す。

【００１４】

【図４】図４は、組織情報に関する実装例を示す。

【００１５】

【図５】図５は、プライベートデータ情報の実装例を示す。

【００１６】

【図６】図６は、実装例によるユーザがシステムでトランザクションを実行するためにブロックチェーンクライアントを起動するときの、ブロックチェーンクライアントの処理例

を示す。
【図７】図７は、実装例によるユーザがシステムでトランザクションを実行するためにブロックチェーンクライアントを起動するときの、ブロックチェーンクライアントの処理例を示す。
【００１７】
【図８】図８は、実装例によるブロックチェーンノードにおけるエンドースロジックでの処理例を示す。
【００１８】
【図９】図９は、実装例によるブロックチェーンノード内のコミットロジックにおける例示のプロセスを示す。
【００１９】
【図１０】図１０は、実装例によるブロックチェーンノード内のバリデーションロジックにおける例示的な処理を示す。
【００２０】
【図１１】図１１は、ブロックチェーンベースのシステムと監査者とを含む別の例示的なシステムを示す。
【００２１】
【図１２】図１２は、実装例による監査ノードの処理例を示す。
【００２２】
【図１３】図１３は、テストプラットフォームを備えたブロックチェーンベースのシステムを含む別の例のシステムを示す。
【００２３】
【図１４】図１４は、テスト環境マネージャの実装例を示す。
【００２４】
【図１５】図１５は、実装例による変換ロジックの処理例を示す。
【００２５】
【図１６】図１６は、実施例によるテストネットワークランチャーの処理例を示す。
【００２６】
【図１７】図１７は、実装例で使用するのに適した例示的なコンピュータデバイスを備えた例示的なコンピューティング環境を示す。
【発明を実施するための形態】
【００２７】
以下の詳細な説明では、本願発明の図および実装例の詳細を提供する。参照番号および図間の冗長な要素の記述は、明確にするために省略される。本明細書を通して使用される用語は、例として提供され、限定することを意図していない。例えば、「自動」という用語の使用は、本発明の実装形態を実施する当業者の所望の実装形態に応じて、実装の特定の態様に対するユーザまたは管理者の制御を含む完全自動または半自動の実装形態を含み得る。選択は、ユーザインタフェースまたは他の入力手段を介してユーザが実施することができ、または所望のアルゴリズムを介して実施することができる。本明細書に記載されている実装例は、単数または複数の組み合わせで利用することができ、実装例の機能は、所望の実装例に応じて、任意の手段で実施することができる。
【００２８】
ブロックチェーン技術は、複数の関係者の間で不変的、分散的、信頼できるデータストアを提供する。ブロックチェーンは、暗号資産を実現するために始まったが、企業システム、特に複数の企業や組織が関与するシステムでも採用されている。企業向けのブロックチェーンプラットフォームの多くは、PoW（Proof of Work）、PoS（Proof of Stake）、PoET（Proof of Elapsed Time）などの時間のかかる斬新なアルゴリズムではなく、ブロックチェーンに保存されたデータについて参加者間の合意を得るために、よりシンプルで従来のアルゴリズムを採用している。エンタープライズブロックチェーンのコンセンサスアルゴリズムには、Paxos、RAFT、PBFTなど、CFT（Crash fault tolerance）

やBFT（Byzantine fault tolerance）のバリエーションがある。これらは一般的に、善意の参加者がトランザクションを実行すると、同じ結果になるはずだと想定している。したがって、トランザクションと結果が有効であると考えられるためには、一定数の参加者が同一の結果に到達することを検証する必要がある。

【００２９】

ブロックチェーンは、関係するすべてのデータが参加者の間で共有されている場合、トランザクションに対してうまく機能することができる。しかし、ブロックチェーン技術は、データをブロックチェーンネットワークの参加者と公に共有することで、保存されたデータの不変性と真正性を達成するが、企業のユースケースでは、組織が他の組織に公開できないデータを必要とすることが多い。このような要件を満たすために、多くのエンタープライズブロックチェーンプラットフォームでは、プライベートデータ機能が導入されている。関連技術の実装では、プライベートデータは、データの閲覧を許可された参加者の特定の部分に保存され、データのハッシュ値はブロックチェーンに保存され、すべての参加者が利用できるようになっている。しかし、このような特徴は、依然として、複数のプライベートデータを含むトランザクションで使用される場合には、解決すべき課題がある。

【００３０】

課題の１つを説明するために、３つの組織があると仮定する。Ａ、Ｂ、Ｃの３つの組織があり、それぞれが二者間のプライベートデータに独自の残高を保持している。例えば、Ｂは２つの残高を保持している。１つはＡと共有するためのもの（「残高ＢＡ」）であり、もう１つはＣと共有するためのもの（「残高ＢＣ」）である。ＡとＢが取引を行う際には、残高が相互に確認され、取引のトランザクションはＡとＢの２つの組織によって検証される。次に、Ｂは、残高ＢＡから残高ＢＣにいくらかのお金を送金して、Ｃに支払うトランザクションを開始することができる。Ｂは２つの残高を見ることができる唯一の組織であるため、トランザクションの後の残高は、Ｂにしか確認できず、Ｂは残高を偽ることができる。他の組織は、それらの残高のハッシュ値を知っているだけで、そのハッシュ値から第１の残高からいくら差し引かれ、第２の残高にいくら入金されたかを計算することができないため、トランザクションの正しさを検証することができない。この問題の根本的な原因は、Ｂがこのトランザクションの唯一の信頼点（信頼できる情報源）であるためである。

【００３１】

本明細書に記載されている実装例は、プライベートデータが関与するトランザクションのための信頼点の数を増加させてエンタープライズブロックチェーンに基づくシステムを提供することに関する。一の実装形態は、１つ以上の組織、１つ以上の読み取り専用組織、各組織および各読み取り専用組織の１つ以上のブロックチェーンノード、ノード間のピアツーピア通信を容易にするブロックチェーンノードを接続するネットワーク、および注文サービスによって構成される。読み取り専用組織は、他の組織のプライベートデータにアクセスする資格を持つように構成されているので、特にすべてのプライベートデータにアクセスできる組織の数が１つしかない場合に、複数のプライベートデータを含むトランザクションを検証することができる。

【００３２】

読み取り専用組織を含む組織内の各ブロックチェーンノードは、トランザクション提案に従ってスマートコントラクトを実行し、その結果をエンドースメントと共に返すエンドーシングロジックと、バリデーティングロジックでトランザクションを検証し、そのトランザクションをローカル台帳に永続化するコミットロジックを有する。エンドーシングロジックは、スマートコントラクトの実行に移る前に、トランザクションのアイデンティティに基づいて、トランザクション提案が読み取り専用組織から送られてきたものかどうかのチェックを行う。トランザクション提案が読み取り専用組織から送られてきたものである場合、エンドーシングロジックはそのトランザクション提案を拒否し、読み取り専用組織が提案やトランザクションを行わないようにする。バリデーティングロジックには、読み取り専用組織がトランザクションを送信することに成功した場合でも、トランザクショ

ンの作成者をチェックする機能もある。バリデーティングロジックは、トランザクションの作成者が読み取り専用組織に属していた場合、そのトランザクションに「無効」のマークを付け、トランザクションの結果をブロックチェーンに適用できないようにする。読み取り専用組織が台帳を変更するためのトランザクションを実行できないようにチェックを実施することで、より信頼性が高く、仲介者や監査者にふさわしい組織とすることができる。このような実装形態により、他の組織は、他の組織のプライベートデータに読み取り専用組織がアクセスすることを許容する。

【００３３】

図１は実装例によるシステムの例を示す。図１のシステムは、複数の組織にグループ化された既知の参加者とコンソーシアムブロックチェーンを形成する。組織は、会社、部署、部門であり、それぞれが独自の信頼を確立するのに十分な独立性を持っている。図１の例では、２つの組織１００－１と１００－２、及び、１つの読み取り専用組織１０１を示す。組織の数と読み取り専用組織の数は、１以上の任意の数とすることができる。読み取り専用組織１０１は、特別なタイプの組織であり、他の組織と同じ構造とロジックを持つ。大きな違いは、読み取り専用組織のメンバーは、ブロックチェーンに保存されているデータに変更を加えるトランザクションを起動することができないことである。組織は、１つまたは複数のブロックチェーンノード１１０、１つまたは複数のブロックチェーンクライアント１２０、および、１つまたは複数の認証局（ＣＡ）１５０を持つことができる。

【００３４】

ブロックチェーンノード１１０は、計算能力、メモリ、永続的なストレージ、およびネットワークを介した通信能力を有するコンピュータである。ブロックチェーンノード１１０は、物理的なコンピュータサーバ、仮想マシン、コンテナ、または、所望の実装形態を容易に行うためのコンピュータと同様の能力を有する他の形態であってもよい。本明細書で説明したように、ブロックチェーンノード１１０は、台帳、ステートデータベース、およびプライベートデータを永続化し、受信したトランザクション提案に応じてスマートコントラクト（参加者が合意したプログラム）を実行し、オーダリングサービス１３０によって順序付けされた後、トランザクションを検証して更新する。

【００３５】

オーダリングサービス１３０は、１つまたは複数のコンピュータによって提供されるサービスであり、これらのコンピュータは、物理的なコンピュータ、仮想マシン、コンテナ、任意の他の類似したマシンであることができ、所望の実装に従って、単数または組み合わせで提供される。オーダリングサービス１３０は、本システムで発行されるトランザクションの完全な順序を決定すること、決定された順序で１つまたは複数のトランザクションを含むブロックをそれぞれ作成すること、およびブロックをすべてのブロックチェーンノード１１０に配信することを担当する。ブロックチェーンクライアント１２０およびＣＡ１５０は、コンピュータで実行されるプログラムであり、このコンピュータは、所望の実装形態に応じて、物理的なコンピュータ、仮想マシン、コンテナ、またはその他の同様のマシンとすることができる。また、プログラムは、ブロックチェーンノード１１０のいずれかと同じコンピュータで実行されてもよいし、別のコンピュータで実行されてもよい。ブロックチェーンクライアント１２０は、トランザクション提案をブロックチェーンノード１１０の１つまたは複数に送信し、応答をトランザクションとしてオーダリングサービス１３０に送信することでトランザクションを開始し、オーダリングサービス１３０は、トランザクションを含むブロックをブロックチェーンノード１１０に配信する。読み取り専用組織１０１は、トランザクションを起動することができないため、ブロックチェーンクライアントを持たない。

【００３６】

ＣＡ１５０は、検証可能なアイデンティティを発行するコンピュータプログラムであり、所望の実装形態に応じて、物理コンピュータ、仮想マシン、コンテナなどで実行することができる。ＣＡ１５０は、ＲＳＡやＥＣＤＳＡなどの暗号アルゴリズムを用いて、ＣＡの秘密鍵で署名されたデジタル証明書を発行するなどの暗号方式を用いることができる。

また、ＣＡ１５０は、自身または他の優れたＣＡによって署名されたデジタル証明書のような自身のアイデンティティを持ち、その公開鍵は、自身が発行した証明書の署名に使用された秘密鍵に対応する。ブロックチェーンノード１１０、ブロックチェーンクライアント１２０、オーダリングサービス１３０、および任意にＣＡ１５０は、ネットワーク１４０によって接続されており、ピアツーピア方式で相互に通信することができる。ネットワーク１４０は、ローカルエリアネットワーク（LAN）、ワイドエリアネットワーク（WAN）、または仮想プライベートネットワーク（VPN）とすることができ、所望の実装形態に応じて有線または無線とすることができる。それらの間の通信は、例えば、TLS（Transport Layer Security）を使用することにより、暗号化および／または安全化することができる。

【００３７】

図２は、ブロックチェーンノード１１０に関する実装例を示す。ブロックチェーンノード１１０は、エンドーシングロジック２１０、コミットロジック２２０、バリデーションロジック２３０、およびスマートコントラクト２４０などの実行可能なプログラムを含むことができる。また、ブロックチェーンノード１１０は、台帳２５０、ステートデータベース２７０、プライベートデータストア２８０、テンポラリプライベートデータ２９０を有する。エンドーシングロジック２１０は、コンセンサスのエンドーシングフェーズを実行する。コミットロジック２２０は、コンセンサスのバリデーションフェーズを実行するものであり、トランザクションの検証のコアとなるバリデーションロジック２３０を内部で呼び出す。スマートコントラクト２４０は、トランザクションの動作を定義するプログラムであり、台帳に格納されている現在のデータに基づいて結果を計算する。スマートコントラクト２４０は、任意に１つまたは複数のパラメータを取るいくつかの関数を定義しており、ブロックチェーンクライアント１２０は、もしあれば関数の１つの識別子とパラメータとを指定したトランザクション提案を送信する。ロジックの詳細は、以下の他の図に開示されている。

【００３８】

システムで共有されるデータは、台帳２５０に格納される。データは、ブロック２６０のチェーンとして格納されている。ステートデータベース２７０は、台帳２５０内のデータの最新バージョンのスナップショットである。ブロック２６０の各トランザクションは、通常、そのトランザクションに関与する値の情報しか持たないため、ブロックチェーンノード１１０は、有効なトランザクションがすべてコミットされて最新の状態が得られた後に、最新の状態のスナップショットを維持・更新する。プライベートデータストア２８０は、特定の組織以外では共有されないプライベートデータを保持する。プライベートデータは、プライベートデータストア２８０に格納され、一般に、非プライベートデータ（例えば、台帳２５０およびステートデータベース２７０に格納されたパブリックデータ）と同じデータモデルを有する。トランザクションには、非プライベートデータの値と、プライベートデータの値のハッシュとが含まれる。ブロックチェーンノード１１０が異なるデータストアを持つことができるように、複数のプライベートデータストア２８０が存在することができ、それぞれのデータストアは、アクセスを許可された組織の異なるセットを含む。

【００３９】

テンポラリプライベートデータストア２９０は、まだコミットされていないプライベートデータを保持している。プライベートデータは、そのデータ（プライベートデータ）を含むトランザクションがコミットされると、プライベートデータストア２８０に書き込まれる。エンドースメントフェーズにおいて、エンドースメントロジック２１０が、ブロックチェーンクライアント１２０からプライベートデータを含むトランザクション提案３７０を受信したときにスマートコントラクト２４０を呼び出すと、スマートコントラクト２４０は、プライベートデータの値だけでなく、非プライベートデータの値を含む結果を返す。その性質上、プライベートデータの値は、すべての参加者と共有することができず、エンドースメントロジック２１０は、プライベートデータの値を返さず、値のハッシュを

返す。バリデーションフェーズでコミットされるまで、プライベートデータの新しい値を保持する必要性があるため、エンドースメントロジック２１０は、テンポラリプライベートデータストア２９０にプライベートデータを格納する。トランザクションがバリデーションフェーズで検証されると、コミットロジック２２０は、非プライベートデータのためにステートデータベース２７０を更新し、プライベートデータのためにプライベートデータストア２８０を更新する。

【００４０】

図３は、ブロック２６０およびトランザクション提案３７０に関する実施例を示す。ブロック２６０は、ハッシュ値で互いに直線的に接続されている。ハッシュ値は、SHA-1、SHA-256、MD5などの一方向性関数によって得られる値である。前ハッシュ値３０１（Previous hash value ３０１）は、前のブロックのハッシュ値である。ブロックハッシュ値３０２は、ブロック２６０自体のハッシュ値である。両方のハッシュ値は、所望の実装に応じて、対象ブロックの全バイト、またはブロックのヘッダ部分などの対象ブロックの一部のバイトに対して計算されてもよい。ブロック２６０には、複数のタイプのトランザクションを含むことができる１つまたは複数のトランザクションが含まれる。図３の例では、ノーマルトランザクション３１０とコンフィグトランザクション３２０の２種類のトランザクションを示している。ノーマルトランザクション３１０は、スマートコントラクト２４０を起動した結果、データを変更するトランザクションである。コンフィグトランザクション３２０は、システムの構成を変更するトランザクションである。ブロックチェーンノード１１０は、現在の構成及び構成履歴を、別の台帳とステートデータベースに格納することができる。

【００４１】

ノーマルトランザクション３１０は、プロポーザアイデンティティ３１１、関数およびパラメータ３１２、結果３１３、および１つまたは複数のエンドースメント３１５を有する。プロポーザアイデンティティ３１１は、トランザクションを要求したエンティティ、またはトランザクション提案３７０を送信したエンティティを識別するための情報を含む。プロポーザアイデンティティ３１１は、ブロックチェーンクライアント１２０に対して発行された証明書と、トランザクションの内容に対するデジタル署名とを含むことができる。

【００４２】

関数およびパラメータ３１２は、スマートコントラクト２４０を実行するために使用される関数およびパラメータの名前または識別子を表す。結果３１３は、関数およびパラメータ３１２を用いてスマートコントラクト２４０を実行して得られた結果を含む。結果３１３は、台帳２５０のデータのアドレスや鍵などの識別子を、それぞれの新しい値とともに含むことができる。結果３１３は、古い値またはスマートコントラクトの実行中に読み取られた値を有し、バリデーションフェーズにおいてその値と最新の値とを比較することにより、他のトランザクションとのコンフリクトを検出し、RAW（Read-after-write）データの危険性を防止することができる。結果３１３は、それらの古い値または読み取られた値の実際の値の代わりに、古い値または読み取られた値のバージョンを有してもよい。プライベートデータが使用される場合、結果３１３は、プライベートデータの値のハッシュだけでなく、非プライベートデータの値を含むことができる。エンドースメント３１５は、ブロックチェーンノード１１０が、関数およびパラメータ３１２を用いてスマートコントラクト２４０を実行することにより、結果３１３を得ることに同意したという検証可能な証拠を含む。例えば、エンドースメントは、ブロックチェーンノード１１０の秘密鍵を用いた結果３１３のデジタル署名と、エンティティのアイデンティティ、具体的には、デジタル署名を作成したブロックチェーンノード１１０のアイデンティティとを含むことができる。

【００４３】

コンフィグトランザクション３２０は、組織情報３３０、プライベートデータ情報３４０、エンドースメントポリシー３５０、および１つまたは複数のエンドースメント３１５

を含む。組織情報３３０は、ブロックチェーンシステムに参加している組織、すなわちコンソーシアムに参加している組織の情報を含む。プライベートデータ情報３４０は、もしあればプライベートデータの情報を含む。エンドースメントポリシー３５０は、あるトランザクションに対するエンドースメントが遵守すべき条件を定義する。あるトランザクションにおけるエンドースメントが条件を満たす場合にのみ、そのトランザクションは有効であるとみなされる。条件の一例としては、エンドースメントを与えた異なる組織の最小の数が挙げられる。条件の他の例としては、「組織Ａ ＡＮＤ （組織Ｂ ＯＲ 組織Ｃ）」のようなエンドースメントを与えた組織のＡＮＤ－ＯＲツリーがある。また、通常のトランザクション３１０と、コンフィグトランザクション３２０とで、異なる条件を定義してもよい。コンフィグトランザクション３２０は、上述したすべての情報を有していてもよいし、所望の実装形態に応じて、トランザクションが適用される情報の違いを有していてもよい。

【００４４】

エンドースメントフェーズ中にスマートコントラクト２４０の実行を要求するために、ブロックチェーンクライアント１２０からブロックチェーンノード１１０にトランザクション提案３７０が送信される。それは、プロポーザアイデンティティ３１１と、トランザクションで起動されるスマートコントラクトのための関数およびパラメータ３１２とを含む。図６および図７で説明したように、トランザクション提案のコンテンツは、ノーマルトランザクション３１０にコピーされ、オーダリングフェーズ中にオーダリングサービス１３０に送信される。コンフィグトランザクション３２０のためのトランザクション提案は、異なる構造を有していてもよく、所望の実装に応じて、組織情報３３０、プライベートデータ情報３４０およびエンドースメントポリシー３５０の全てまたは一部を有していてもよい。

【００４５】

図４は、組織情報３３０の実装例を示す。この情報は、各組織の組織名３３１、トラストルート３３２、許可３３３を含む。各組織の組織名３３１は、各組織の識別子であり、例えば、「Ａ社」、「Ｂ社」などとすることができる。プロポーザアイデンティティ３１１やエンドースメント３１５には、アイデンティティの一部として組織名３３１が含まれていてもよい。各組織のトラストルート３３２は、その組織に属することを主張するエンティティのアイデンティティ３１１を検証するために用いられる情報である。トラストルート３３２の一例は、組織内のＣＡ１５０の自己署名証明書である。組織内のブロックチェーンクライアント１２０が、ＣＡ１５０が発行し、自己署名証明書に対応する鍵ペアで署名された証明書を、トランザクション提案３７０におけるプロポーザアイデンティティ３１１として使用する場合、エンドーシングロジック２１０およびバリデーションロジック２３０は、証明書の署名が自己署名証明書の公開鍵で正しいかどうかを確認することで、証明書を検証することができる。別の例としては、別のＣＡがＣＡ１５０のために発行した証明書があり、これは公に知られた信頼できるものであってもよい。この場合も、署名には証明書に対応する鍵ペアを使用する必要がある。各組織の許可３３３は、その組織がシステム内でトランザクションを開始したり提案したりすることを許可されているかどうか、組織の権限を定義する。例えば、組織の許可３３３が“ａｌｌ”の場合、その組織は、トランザクションを開始できる組織１００であり、許可３３３が“ｒｅａｄ－ｏｎｌｙ”の場合、その組織は、トランザクションを開始できない読み取り専用組織１０１である。

【００４６】

図５は、プライベートデータ情報３４０に関する実装例を示す。この情報は、各プライベートデータストアのデータストア名３４１と許可３４２を含む。データストア名３４１は、プライベートデータストアの識別子であり、例えば、「プライベートデータＡ」、「プライベートデータＡＢ」などとすることができる。許可３４２は、プライベートデータストア内のデータに対する組織からの許可されたアクセスを指定する許可リストである。許可３４２の一例は、図に描かれているように、「Readable and Writable by A Company」であり、これは、組織「Ａ社」のブロックチェーンノード１１０がプライベートデ

ータストアを読み書きできることを意味している。エンドースメントフェーズにおいて、スマートコントラクト２４０によってプライベートデータストアがアクセスされると、ブロックチェーンノード１１０に許可が与えられる。

【００４７】

図６および図７は、実装例による、ユーザがシステムでトランザクションを実行するためにブロックチェーンクライアント１２０を起動するときの、ブロックチェーンクライアント１２０のプロセスの例を示す。まず、ブロックチェーンクライアント１２０は、自身のアイデンティティを用いて、トランザクション提案３７０を構成する（ステップ４０１）。トランザクション提案３７０は、プロポーザアイデンティティ、（即ち、ブロックチェーンクライアント１２０と同じ組織のＣＡ１５０が発行したデジタル証明書や、デジタル証明書に対応するキーペアで作成された署名などのブロックチェーンクライアント１２０のアイデンティティ）と、スマートコントラクトを起動するための関数とパラメータを含む。そして、ブロックチェーンクライアント１２０は、以下のように、十分な数のエンドースメントを得るまでループする（ステップ４０２）。４０３において、ブロックチェーンクライアント１２０は、エンドースメントを得るためにトランザクション提案３７０を送信する組織と、その組織に属する特定のブロックチェーンノード１１０とを選択する。ブロックチェーンクライアント１２０は、組織内のいずれかのブロックチェーンノード１１０から既にエンドースメントを取得しているか否か、組織がトランザクション提案３７０に含まれるプライベートデータにアクセスできるか否かなど、複数の基準に従って組織を選択することができる。また、ブロックチェーンノードが受信したブロックの数や、ブロックチェーンノードの負荷など、複数の基準を用いてブロックチェーンノード１１０を選択してもよい。組織とブロックチェーンノード１１０を選択した後、ブロックチェーンクライアント１２０は、トランザクション提案３７０をブロックチェーンノード１１０に送信する（ステップ４０４）。トランザクション提案３７０は、図８で後述するように、ブロックチェーンノード１１０内のエンドーシングロジック２１０によって受信され、処理される。そして、ブロックチェーンクライアント１２０は、ブロックチェーンノード１１０からの応答を待つ（ステップ４０５）。応答は、エンドーシングロジックの処理が正常に終了したか否かを示すものであり、スマートコントラクト２４０からの結果と、成功した場合にはブロックチェーンノード１１０からのエンドースメントとが含まれる。エンドーシングロジックの処理が失敗した場合、ブロックチェーントクライアント１２０はステップ４０３に戻り、同じブロックチェーンノード１１０または別のブロックチェーンノード１１０のいずれかにトランザクション提案３７０の送信を試みることができるようにする。

【００４８】

ステップ４０６において、トランザクション提案３７０が成功した場合（Ｎｏ）、ブロックチェーンクライアント１２０は、ステップ４０７において、同じトランザクション提案３７０について他のブロックチェーンノード１１０から得た前の結果と比較する。それらが同一であれば（Ｙｅｓ）、ブロックチェーンクライアント１２０は、結果と新しいエンドースメントを保持して、ステップ４０２に戻る。そうでなければ（Ｎｏ）、ブロックチェーンノード１１０がトランザクション提案３７０の結果についてコンセンサスに達することができないので、ブロックチェーンクライアント１２０は、取得したすべてのエンドースメントを破棄し、ステップ４０２から再び開始する。ステップ４０１～４０８は、ブロックチェーンシステムの３フェーズコンセンサスにおけるエンドースメントフェーズを構成する。取得したエンドースメント（複数可）が十分であれば、ブロックチェーンクライアント１２０は次のステップに切り替える（ステップ４０２：Ｙｅｓ）。

【００４９】

ステップ４１０において、ブロックチェーンクライアント１２０は、提案者アイデンティティ３１１、関数およびパラメータ３１２、結果３１３、並びに、蓄積されたエンドースメント３１５（複数可）によりノーマルトランザクション３１０を構成し、トランザクションをオーダリングサービス１３０に送信する。ステップ４１１では、エラーが発生し

た場合（Ｙｅｓ），そのエラーをユーザに返す（ステップ４１５）。そうでなければ（Ｎｏ）、トランザクションを受信した後、オーダリングサービス１３０は、組織から受信したトランザクションの順序を決定し、それぞれが１つ以上のトランザクションを含むブロックを作成して、ブロックチェーンノード１１０に配信する。これにより、コンセンサスにおけるオーダリングフェーズが形成される。ブロックチェーンクライアント１２０は、ステップ４１２で、ブロックチェーンノード１１０のいずれかからバリデーションフェーズの結果を待つ。結果がエラーを示すか、またはトランザクションが無効であると考えられる場合（Ｙｅｓ）、ブロックチェーントクライアント１２０は、ステップ４１５でユーザにエラーを返す。そうでなければ（Ｎｏすなわち結果が成功）、トランザクションは有効であるとみなされ、ブロックチェーントクライアント１２０は、ステップ４１４でユーザに成功を返す。任意選択で、ブロックチェーントクライアント１２０は、所望の実装形態に応じて、ユーザが新しい値を知るために、スマートコントラクト２４０からの結果を返す。

【００５０】

図８は、実装例による、ブロックチェーンノード１１０内のエンドーシングロジック２１０における例示的なプロセスを示す。エンドーシングロジック２１０は、任意のブロックチェーンクライアント１２０からの新規トランザクション提案３７０を待つ（ステップ５０１）。そして、プロポーザアイデンティティ３１１を用いて、プロポーザが所属する組織を取得する（ステップ５０２）。組織情報３３０を参照して、エンドーシングロジック２１０は、その組織が読み取り専用であるか否かを確認する（ステップ５０３）。読み取り専用であれば（Ｙｅｓ）、エンドーシングロジック２１０は、ブロックチェーンクライアント１２０にエラーを返す（Ｓｔｅｐ　５１１）。そうでない場合（Ｎｏ）、エンドーシングロジック２１０は、アイデンティティが正しいことをさらに確認する（ステップ５０４）。

【００５１】

チェックは、証明書がＣＡ１５０によって発行されているかどうか、およびトランザクション提案３７０の署名が正しいかどうかを含む。チェックが失敗した場合（Ｎｏ）、エンドーシングロジック２１０は、ブロックチェーンクライアント１２０にエラーを返す（ステップ５１１）。そうでなければ（Ｙｅｓ）、エンドーシングロジック２１０は、トランザクション提案３７０の関数及びパラメータに従って、スマートコントラクト２４０を実行する（ステップ５０５）。

【００５２】

ステップ５０６において、スマートコントラクトが失敗した場合（Ｙｅｓ）、エンドーシングロジック２１０は、ブロックチェーンクライアント１２０にエラーを返す（ステップ５１１）。スマートコントラクトが結果を正常に返す場合（Ｎｏ）、エンドーシングロジック２１０は、結果が任意のプライベートデータを含むかどうかをチェックする（ステップ５０７）。プライベートデータが含まれている場合（Ｙｅｓ）、エンドーシングロジック２１０は、プライベートデータをテンポラリプライベートデータストア２９０に格納する（ステップ５０８）。エンドーシングロジック２１０は、結果からプライベートデータを削除し、スマートコントラクトがプライベートデータのハッシュ値を計算しなかった場合に備えて、代わりにプライベートデータのハッシュ値を追加する。最後に、エンドーシングロジック２１０は、結果と、結果のエンドースメントをブロックチェーンクライアント１２０に返す（ステップ５１０）。

【００５３】

図９は、実装例による、ブロックチェーンノード１１０内のコミットロジック２２０における例示的なプロセスを示す。コミットロジック２２０は、オーダリングサービス１３０からの新しいブロックを待機する（ステップ６０１）。そして、コミットロジック２２０は、ブロック内のトランザクションをトラバースし、トランザクションを１つずつコミットする。まず、コミットロジック２２０は、ブロック内にコミットされなかったトランザクションがあるかどうかを確認し（ステップ６０２）、あれば（Ｙｅｓ）、オーダリン

グサービス１３０が決定した、ブロック内で指定された順序でコミットされなかった最初のトランザクションを取得する（ステップ６０３）。次に、コミットロジック２２０は、図１０で後述するトランザクションのバリデーションロジック２３０を実行する（ステップ６０４）。結果が有効であれば（ステップ６０５：Ｙｅｓ）、コミットロジック２２０は、ステートデータベース２７０を更新して、トランザクションの結果を適用する（ステップ６０６）。

【００５４】

プライベートデータがトランザクションに関与し、組織がプライベートデータの許可３４２に従ってプライベートデータを読み取ることが許可されている場合（ステップ６０７：Ｙｅｓ）、コミットロジック２２０は、テンポラリプライベートデータストア２９０から、またはプライベートデータを有する他のブロックチェーンノード１１０から、プライベートデータの実際の値を取得する（ステップ６０８）。後者の実装形態を実現する方法としては、ゴシップなどのピアツーピアの通信プロトコルがある。

【００５５】

コミットロジック２２０は、プライベートデータの値を取得した後、それらに従ってプライベートデータストア２８０を更新する。検証結果が無効である場合（ステップ６０５：Ｎｏ）、コミットロジック２２０は、ステートデータベース２７０およびプライベートデータストア２８０の更新をスキップする。最後に、コミットロジック２２０は、トランザクションの有効性を含むトランザクションに対するコミットの結果を、リスニングしているものがあればブロックチェーンクライアント１２０に発する（ステップ６０９）。ステップ６０９は、通信を最適化するために、各トランザクションに対して、各ブロックに対して、トランザクションに対する結果を集約して実行してもよいし、複数のブロックに対して実行してもよい。ステップ６０２に戻り、すべてのトランザクションが消費されるまで繰り返される。コミットロジックは、各ブロックチェーンノード１１０に分散して実行される。それらは、台帳の中に同じブロックを持ち、同じロジックを持っているので、すべてのブロックチェーンノード１１０のコミットロジック２２０が同じ結果を得ることが保証される。

【００５６】

図１０は、実装例による、ブロックチェーンノード１１０におけるバリデーションロジック２３０における例示的なプロセスを示す。バリデーションロジック２３０は、コミットするトランザクションがあるときにコミットロジック２２０から呼び出される。まず、バリデーションロジック２３０は、トランザクションを取得し（ステップ７０１）、プロポーザアイデンティティ３１１から組織を取得する（ステップ７０２）。次に、バリデーションロジック２３０は、一連のチェックを実行する（ステップ７０３）。バリデーションロジック２３０は、組織が読み取り専用であるかどうかをチェックする。通常の手順では、バリデーションロジック２１０が読み取り専用組織にトランザクションの提案を許可しないため、このようなことは起こりえないが、組織の権限を読み取り専用に変更した直後など、一部のコーナーケースではこのようなことが起こりうる。このチェックにより、読み取り専用組織が提案したトランザクションが、たとえコミットされることになっても、決して有効にならないようにする。次に（ステップ７０４）、バリデーションロジック２３０は、ステップ５０４と同様に、プロポーザアイデンティティ３１１が有効であるか否かをチェックする。次に、バリデーションロジック２３０は、そのトランザクションが、以前にコミットされた他のトランザクションとコンフリクトしていないかどうかをチェックする（ステップ７０５）。例えば、バリデーション時の最新の値と、結果３１３に含まれるエンドースメント時に読み取られた値とを比較することができる。それらが異なる場合、同じデータを含む他のトランザクションが存在し、そのトランザクションがエンドースされた後、そのトランザクションがコミットされようとしている前にコミットされたことを意味し、したがって、そのトランザクションは有効であるとみなされない。次に、バリデーションロジック２３０は、エンドースメントが有効であるか否かをチェックする（ステップ７０６）。このチェックには、各エンドースメントにおける署名が、結果３１

３に対して生成されたものであり、かつ、そのアイデンティティが有効であるか否かをチェックすることが含まれる。最後に、バリデーションロジック２３０は、エンドースメントがエンドースメントポリシー３５０を満たすかどうかをチェックする（ステップ７０７）。トランザクションがすべてのチェックに合格した場合、バリデーションロジック２３０は、コミットロジック２２０に「有効」を返す（ステップ７１０）。そうでなければ、バリデーションロジック２３０はコミットロジック２２０に「無効」を返す（ステップ７１１）。

【００５７】

上述の第１の実装例では、ブロックチェーンシステムは、完全な許可を有する組織と読み取り専用組織との２種類の組織を有することができる。エンドーシングロジックおよびバリデーションロジックは、読み取り専用組織内の任意のエンティティによって提案された任意のトランザクションが拒否されるか、または無効とマークされることを保証する。一方で、読み取り専用組織のブロックチェーンノードは、トランザクションをエンドースすることができる。これらの読み取り専用組織がプライベートデータの読み取りを許可されると、プライベートデータを含むトランザクションに対して複数のエンドースメントを得る機会が増えるため、プライベートデータに関するＳＰｏＴの問題が緩和される。読み取り専用組織がトランザクションを実行できないという制限により、他の組織は中立的な第３者としてそれらを信頼するようになり、エンドースメントを得るためにプライベートデータを公開する機会が増える。

【００５８】

この例は、読み取り専用組織があらゆるプライベートデータを読み取ることを常に許可するように、システムを拡張することができる。これは、すべてのプライベートデータの許可に「読み取り専用組織による読み取り」を暗黙的に追加することに相当する。

【００５９】

第２の実装例において、図１１は、監査者を有するブロックチェーンベースのシステムを含む別の例示的システムを示す。構成要素は、第１の実装例の図１のものと同様に動作する。この例では、１つの読み取り専用組織がシステムの監査者として機能する。したがって、読み取り専用組織１０１は、ネットワーク１４０または読み取り専用組織１０１の内部の他のネットワークによって１つまたは複数のブロックチェーンノード１１０と接続される１つまたは複数の監査ノード１１１０を有する。監査ノード１１１０は、監査目的のために、データに対して様々なチェックを行う。監査ノード１１１０は、システムで共有されているデータの情報、すなわちブロックやトランザクションをブロックチェーンノード１１０から取得する。監査ノード１１１０は、物理的なコンピュータ、仮想マシンまたはコンテナであってもよいし、ブロックチェーンノード１１０の１つと同じマシンであってもよい。監査ノード１１１０を除く図１１の構成要素の実装例および実行フローについては、第１の実装例で開示したものと同一であるため、ここでは説明を省略する。ただし、ブロックチェーンノード１１０には、監査ノード１１１０が台帳やプライベートデータを参照できる機能を追加する必要がある。一つの実装形態は、監査ノード１１１０によってアクセス可能な関数、ＡＰＩ（Application Programming Interface）を公開し、監査ノード１１１０からの要求に応じて、ブロック、トランザクション、プライベートデータを返すことである。他の実装形態としては、ブロックチェーンノード１１０、すなわちコミットロジック２２０が、ブロック、トランザクション、およびプライベートデータをコミットした後に監査ノード１１１０に転送するものがある。他の実装形態としては、監査ノード１１１０が、オーダリングサービス１３０からブロックを直接受け取り、ブロックチェーンノード１１０の１つからプライベートデータを受け取るものがある。

【００６０】

図１２は、実装例による、監査ノード１１１０の例示的なプロセスを示す。まず、監査ノード１１１０は、最後の段落で説明されるいくつかの例の手段によって、新しいブロックおよびブロック内のトランザクションに関連するプライベートデータを待つ（ステップ１２０１）。次に、監査ノード１１１０は、ブロックの整合性をチェックする（ステップ

１２０２）。このチェックには、前のブロックと現在のブロックの実際のデータに対するハッシュ値３０１と３０２のバリデーションが含まれる。次に、監査ノード１１１０は、ブロック内の各未処理トランザクションに対して（ステップ１２０３：Ｙｅｓ）、トランザクションの保存された順序でチェックを繰り返す（ステップ１２０４）。監査ノード１１１０は、トランザクションの整合性をチェックし（ステップ１２０５）、もしあればプライベートデータをチェックする（ステップ１２０６）。トランザクションの整合性には、プロポーザアイデンティティ３１１およびエンドースメント３１５の署名が含まれ、プライベートデータの整合性には、実際のプライベートデータと一致する結果３１３のハッシュが含まれる。また、監査ノード１１１０は、追加のチェック（ステップ１２０７）を行ってもよく、例えば、ある口座の残高が他の口座と一致しているかどうかをチェックしたり、マネーロンダリングなどの疑わしい行為がないかをチェックしたりなど、所望の実装形態に応じたチェックを行うことができる。チェックの結果、何らかのエラーが見つかった場合（ステップ１２０８）、監査ノード１１１０は、エラーの重大性に応じて警告を発してもよい（ステップ１２０９）。その後、監査ノード１１１０は、結果をレポートに蓄積する（ステップ１２１０）。レポートは、トランザクションごと、ブロックごと、または一定期間ごとに生成されてもよい。

【００６１】

この第２の実装例では、信頼できる第三者機関として、読み取り専用組織は、ブロックチェーンベースのシステムを監査することができる。読み取り専用組織は、ブロックチェーンの整合性を検証することに加えて、検証された不変性に基づいて、規制や法律などで推奨または義務付けられた様々なチェックを行うことができる。これにより、監査をより強固で効率的なものにすることに貢献する。

【００６２】

第３の実装例において、図１３は、テストプラットフォームを備えたブロックチェーンベースのシステムを含む別の例示的システムを示す。構成要素は、第１の実装例における図１のものと同様に動作する。組織１００－１および１００－２、読み取り専用組織１０１およびオーダリングサービス１３０、ならびに各組織のブロックチェーンノード１１０、ブロックチェーンクライアント１２０およびＣＡ１５０を含むブロックチェーンネットワークは、以下では「本番ブロックチェーンネットワーク」または「本番環境」で呼ばれる。

【００６３】

この例では、１つの読み取り専用組織が、システムのテスト環境を規定する。開発中および展開前にブロックチェーンベースのシステムをテストするためには、実際のシステムと同じデータを使用することが望ましい場合がある。既存システムのアップデートにおいては、より実用的で効果的なテストを実現するために、既存システムの台帳がテスト環境で使用するデータの有力な候補となる。しかし、公開鍵や秘密鍵などの暗号材料は、テスト環境での署名が本番環境と同様の効果や影響を与える可能性があるため、テスト環境では利用できないし、利用すべきではない。さらに、プライベートデータも必要となるが、台帳には含まれていない。この例では、読み取り専用組織が、既存のブロックチェーンシステムから抽出したデータに、暗号材料や関連情報に必要な変換を施し、全組織のプライベートデータを加えたものをテスト環境に提供することができる。

【００６４】

テスト環境マネージャ１３１０は、ブロック、トランザクション、およびプライベートデータを適切に変換し、変換されたデータを使用して、テスト目的のブロックチェーンネットワークを生成および破壊する。テスト環境マネージャ１３１０は、物理的なコンピュータ、仮想マシン、コンテナなどであり、そのハードウェアを他のコンポーネントと共有することができる。テスト環境マネージャ１３１０は、ネットワーク１４０または読み取り専用組織１０１の内部の他のネットワークを使用して、ブロックチェーンノード１１０と通信することができる。テスト環境マネージャ１３１０は、第２の実装例で示した方法で、ブロックとプライベートデータを取得する。テストブロックチェーンネットワーク１

３２０は、テスト環境マネージャ１３１０によって生成および構成されたブロックチェーンベースのシステムである。それらは、本番環境にできるだけ近い環境で新しいソフトウェアをテストすることを目的としているため、図１と同じまたは類似した構造を有している。大きな違いは、テストブロックチェーンネットワークのすべての組織は、読み取り専用組織１０１が行うテストのためだけに構築されているため、読み取り専用組織１０１によって管理されていることである。テストブロックチェーンネットワークの各組織におけるノードの数は、テストケースやテストに利用可能なリソースに応じて、図１３の組織におけるノードの数とは異なっていてもよい。テストマネージャ１３３０は、テストを管理するものであり、テストを行う前にテスト環境マネージャ１３１０に新しいテストネットワークの作成を指示し、テストブロックチェーンネットワーク１３２０でテストを行う。テストマネージャ１３３０は、物理的なコンピュータ、仮想マシン、コンテナなどであり、そのハードウェアを他のコンポーネントと共有することができる。テストマネージャ１３３０のソフトウェアは、システムのソースコードのリポジトリに変更が加えられたときにテストをトリガーする、ＣＩ（Continuous Integration）用の自動テストツールの一部とすることができる。

【００６５】

図１４は、テスト環境マネージャ１３１０に関する実装例を示す。テスト環境マネージャ１３１０は、変換ロジック１４１０とテストネットワークランチャー１４２０との２つのロジックを有する。変換ロジック１４１０は、読み取り専用組織１０１のブロックチェーンノード１１０、オーダリングサービス１３０から、いずれかのブロックとプライベートデータを受け取り、ブロックとプライベートデータを、テストブロックチェーンネットワーク１３２０で使用するのに適したものに変換する。変換ロジック１４１０は、変換されたデータを、変換台帳１４３０、変換ステートデータベース１４４０、および変換プライベートデータストア１４５０に蓄積し、それらは後にテストネットワークランチャー１４２０によって使用され、新しいブロックチェーンネットワーク１３２０を生成する。テストネットワークランチャー１４２０は、要求に応じて新しいブロックチェーンネットワーク１３２０を生成する。変換されたデータを使用してテストブロックチェーンネットワーク１３２０を初期化し、本番のブロックチェーンネットワークと同じ状態で開始することで、テストマネージャ１３３０は本番データを使用してより現実的なテストを行うことができる。変換については、本番環境の組織のＣＡ１５０の秘密鍵は、読み取り専用組織１０１では利用できないため、自分のＣＡ１５０を使って証明書や署名を生成する。変換台帳１４３０、変換ステートデータベース１４４０、変換プライベートデータストア１４５０は、それぞれ図２の台帳２５０、ステートデータベース２７０、プライベートデータストア２８０に対応する。これらのコンポーネントの構造は同一であるが、変換されたデータを保存することを意図している。本番環境の組織のモックアップを作成するために、テスト環境マネージャ１３１０は、複数のＣＡ１５０および変換されたプライベートデータストア１４５０を、それぞれがモックアップの組織に対応するように維持する。

【００６６】

図１５は、実装例による、変換ロジック１４１０の例示的なプロセスを示す。最初に、変換ロジック１４１０は新しいブロックおよびブロック内のトランザクションに関連するプライベートデータを待つ（ステップ１５０１）。変換ロジック１４１０は、ブロック内の未処理のトランザクションに対して、以下の変換を繰り返し行う（ステップ１５０３、１５０４）。トランザクションがノーマルトランザクションである場合（ステップ１５０５：Ｎｏ）、変換ロジック１４１０は、プロポーザアイデンティティ３１１とエンドースメント３１５を変換する（ステップ１５０６）。この変換は、アイデンティティおよびエンドースメントの証明書を、テスト環境マネージャ１３１０内のＣＡ１５０が発行したものに置き換え、署名を、置き換えられた証明書に対応する秘密鍵によって生成されたものに置き換えることを含む。任意選択で、変換ロジック１４１０は、テストに同じデータを使用することが懸念される場合に、トランザクション内のデータの匿名化などの追加の変換（ステップ１５０７）を行うことができる。ステップ１５０７で結果３１３または任意

のデータが変換された場合、署名を再度計算することができる。そして、トランザクションが有効であれば、ステップ６０６で行ったように、トランザクションの結果を適用するために、変換ステートデータベース１４４０を更新する（ステップ１５０８）。トランザクションが有効であり、任意のプライベートデータを含む場合には、変換プライベートデータストア１４５０を適切に更新する（ステップ１５０９）。

【００６７】

トランザクションがコンフィグトランザクションである場合（ステップ１５０５：Ｙｅｓ）、変換ロジック１４１０は、トランザクション内の組織情報３３０内の組織に対するトラストルート３３２を変換する（ステップ１５１１）。テストブロックチェーンネットワーク１３２０では異なるＣＡが使用され、トラストルートの証明書は本番環境のトラストルート３３２の証明書とは異なるはずなので、トラストルート３３２はテスト環境マネージャ１３１０のＣＡ１５０の証明書に置き換えておく。次に、変換ロジック１４１０は、ステップ１５０６のように、プロポーザアイデンティティとエンドースメントを変換する（ステップ１５１２）。

【００６８】

ブロック内のすべてのトランザクションが変換されると（ステップ１５０３：Ｎｏ）、変換ロジック１４１０は、ブロックのハッシュ値を計算し、ブロックのハッシュ値３０２を新しい値に置き換える。また、ステップ１５２０では、前のブロックのハッシュ値３０１を変換台帳１４３０の前のブロックのハッシュ値に置き換える。最後に、変換ロジック１４１０は、変換されたブロックを変換台帳１４３０に追加する（ステップ１５２１）。

【００６９】

図１６は、実装例による、テストネットワークランチャー１４２０の例示的なプロセスを示す。テストマネージャ１３３０からの要求に応じて、テストネットワークランチャー１４２０は、すべての組織および読み取り専用組織に対して新しいブロックチェーンノード１１０を作成する（ステップ１６０１）。ノードを空の台帳とデータベースで開始する代わりに、変換台帳１４３０と変換ステートデータベース１４４０を各ブロックチェーンノードにコピーする（ステップ１６０２）。効率化のために、変換台帳１４３０と変換ステートデータベース１４４０が保存されている永続的なデバイスのいくつかの機能、例えば、スナップショットやゼロコピークローニングなどを使用することができる。次に、変換プライベートデータストア１４５０を、適切な組織のブロックチェーンノードにコピーする（ステップ１６０３）。最後に、テストネットワークランチャー１４２０は、ブロックチェーンノード１１０を起動し（ステップ１６０４）、テストマネージャ１３３０に情報を渡して、新しく作成されたブロックチェーンネットワークでテストを実行できるようにする。

【００７０】

図１７は、本明細書に記載されているようなブロックチェーンシステムにおける任意のノード、テスト環境マネージャ、組織、注文サービスなどを容易にする装置など、いくつかの実装例で使用するのに適した例示的なコンピュータデバイスを備えた例示的なコンピューティング環境を示している。コンピューティング環境１７００のコンピュータデバイス１７０５は、１つまたは複数の処理ユニット、コア、またはプロセッサ１７１０、メモリ１７１５（例えば、ＲＡＭ、ＲＯＭ、および／または同様のもの）、内部ストレージ１７２０（例えば、磁気、光学、ソリッドステートストレージ、および／または有機）、および／またはＩＯインタフェース１７２５を含むことができ、これらのうちのいずれかは、情報を通信するための通信機構またはバス１７３０上に結合されるか、またはコンピュータデバイス１７０５に組み込まれることができる。ＩＯインタフェース１７２５はまた、所望の実装に応じて、カメラから画像を受信したり、プロジェクタまたはディスプレイに画像を提供したりするように構成される。

【００７１】

コンピュータデバイス１７０５は、入力／ユーザインタフェース１７３５および出力デバイス／インタフェース１７４０に通信可能に結合され得る。入力／ユーザインタフェー

ス１７３５および出力デバイス／インタフェース１７４０のいずれか一方または両方は、有線または無線のインタフェースあってもよく、着脱可能であってもよい。入力／ユーザインタフェース１７３５は、入力を提供するために使用することができる、物理的または仮想的な任意のデバイス、コンポーネント、センサ、またはインタフェース（例えば、ボタン、タッチスクリーンインタフェース、キーボード、ポインティング／カーソルコントロール、マイク、カメラ、点字、モーションセンサ、光学リーダ、および／または同様のもの）を含むことができる。出力デバイス／インタフェース１７４０は、ディスプレイ、テレビ、モニタ、プリンタ、スピーカ、点字などを含んでもよい。いくつかの実装例では、入力／ユーザインタフェース１７３５および出力デバイス／インタフェース１７４０は、コンピュータデバイス１７０５と共に埋め込まれるか、またはコンピュータデバイス１７０５に物理的に結合され得る。他の実装例では、他のコンピュータデバイスは、コンピュータデバイス１７０５のための入力／ユーザインタフェース１７３５および出力デバイス／インタフェース１７４０の機能として機能するか、またはその機能を提供することができる。

【００７２】

コンピュータデバイス１７０５の例は、高度なモバイルデバイス（例えば、スマートフォン、車両および他の機械に搭載されたデバイス、人間および動物によって運ばれたデバイスなど）、モバイルデバイス（例えば、タブレット、ノートブック、ラップトップ、パーソナルコンピュータ、ポータブルテレビ、ラジオなど）、およびモバイル用に設計されていないデバイス（例えば、デスクトップコンピュータ、他のコンピュータ、情報キオスク、１つまたは複数のプロセッサがそこに埋め込まれたおよび／またはそれに結合されたテレビ、ラジオなど）を含み得るが、これらに限定されない。

【００７３】

コンピュータデバイス１７０５は、同じ構成または異なる構成の１つまたは複数のコンピュータデバイスを含む、任意の数のネットワーク化されたコンポーネント、デバイス、およびシステムと通信するために、外部ストレージ１７４５およびネットワーク１７５０に（例えば、ＩＯインタフェース１７２５を介して）通信可能に結合することができる。コンピュータデバイス１７０５または接続された任意のコンピュータデバイスは、サーバ、クライアント、シンサーバ、一般機械、特別目的機械、または別のラベルとして機能し、サービスを提供し、または参照することができる。

【００７４】

ＩＯインタフェース１７２５は、コンピューティング環境１７００の少なくともすべての接続されたコンポーネント、デバイス、およびネットワークとの間で情報を通信するための、任意の通信またはＩＯプロトコルまたは標準（例えば、イーサネット、８０２．１１ｘ、ユニバーサルシステムバス、ＷｉＭａｘ、モデム、セルラーネットワークプロトコルなど）を使用した有線および／または無線インタフェースを含むことができるが、これに限定されない。ネットワーク１７５０は、任意のネットワークまたはネットワークの組み合わせ（例えば、インターネット、ローカルエリアネットワーク、ワイドエリアネットワーク、電話ネットワーク、セルラーネットワーク、衛星ネットワークなど）とすることができる。

【００７５】

コンピュータデバイス１７０５は、一過性の媒体および非一過性の媒体を含む、コンピュータで使用可能な媒体またはコンピュータで読み取り可能な媒体を使用および／または通信することができる。一過性の媒体は、伝送媒体（例えば、金属ケーブル、光ファイバー）、信号、搬送波などを含む。非一過性の媒体には、磁気媒体（ディスク、テープなど）、光学媒体（CD ROM、デジタルビデオディスク、ブルーレイディスクなど）、固体媒体（RAM、ROM、フラッシュメモリ、ソリッドステートストレージなど）、その他の不揮発性のストレージまたはメモリが含まれる。

【００７６】

コンピュータデバイス１７０５は、いくつかの例示的なコンピューティング環境におい

て、技術、方法、アプリケーション、プロセス、またはコンピュータ実行可能な命令を実装するために使用することができる。コンピュータ実行可能な命令は、一過性の媒体から取得することができ、非一過性の媒体に格納され、そこから取得することができる。実行可能な命令は、任意のプログラミング言語、スクリプト言語、および機械言語（C、C++、C#、Java、Visual Basic、Python、Perl、JavaScriptなど）の１つまたは複数に由来することができる。

【００７７】

プロセッサ（複数可）１７１０は、任意のオペレーティングシステム（ＯＳ）（図示せず）の下で、ネイティブ環境または仮想環境で実行することができる。論理ユニット１７６０、アプリケーション・プログラミング・インタフェース（ＡＰＩ）ユニット１７６５、入力ユニット１７７０、出力ユニット１７７５、および異なるユニットが相互に、ＯＳと、および他のアプリケーション（図示せず）と通信するためのユニット間通信機構１７９５を含む、１つまたは複数のアプリケーションを展開することができる。説明されたユニットおよび要素は、デザイン、機能、構成、または実装において変化することができ、提供される説明に限定されない。プロセッサ（複数可）１７１０は、中央処理装置（ＣＰＵ）などのハードウェアプロセッサの形態であっても、ハードウェアユニットとソフトウェアユニットの組み合わせであってもよい。

【００７８】

いくつかの実装例では、情報または実行命令がＡＰＩユニット１７６５によって受信されると、それは１つまたは複数の他のユニット（例えば、論理ユニット１７６０、入力ユニット１７７０、出力ユニット１７７５）に伝達されてもよい。いくつかの例では、論理ユニット１７６０は、上述のいくつかの例示的な実装例において、ユニット間の情報の流れを制御し、ＡＰＩユニット１７６５、入力ユニット１７７０、出力ユニット１７７５によって提供されるサービスを指示するように構成されてもよい。例えば、１つまたは複数のプロセスまたは実装の流れは、論理ユニット１７６０単独で、またはＡＰＩユニット１７６５と連携して制御されてもよい。入力ユニット１７７０は、例示的な実装で説明された計算のための入力を得るように構成されてもよく、出力ユニット１７７５は、例示的な実装で説明された計算に基づいて出力を提供するように構成されてもよい。

【００７９】

ブロックチェーンシステムは、複数のノードを介してスマートコントラクトトランザクションを容易にするように構成される。第１のノードに関する例では、（例えば、ブロックチェーン内の複数のノードの任意のノード）のプロセッサ（複数可）１７１０は、複数のノードの第２のノードからスマートコントラクトトランザクションの提案を受信することを契機に、第２のノードに関連付けられたブロックチェーンシステム内の組織を特定し、組織が提案によって示されるスマートコントラクトトランザクションを実行できるかどうかを判定し、組織が提案によって示されるスマートコントラクトトランザクションを実行する権限を有していないことを示す判定に対して、第２のノードにエラーを返し、組織がプロポーザルによって示されるスマートコントラクトトランザクションを実行する権限を有していることを示す判定に対して、図８に示されるように、スマートコントラクトトランザクションを実行する。

【００８０】

第１のノードの一例では、プロセッサ（複数可）１７１０は、図４および提案者アイデンティティ３１１に例示されているように、組織が、組織が読み取り専用組織であるために提案が示すスマートコントラクトトランザクションを実行する権限を有していないことを示していると判定することと、組織が、組織が読み取り専用組織でないために提案が示すスマートコントラクトトランザクションを実行する権限を有していることを示していると判定することによって、組織が、提案が示すスマートコントラクトトランザクションを実行できるかどうかを判定するように構成されている。

【００８１】

第１のノードの例では、プロセッサ（複数可）１７１０は、第２のノード（例えば、第

１のノードに結合された別のノード）からスマートコントラクトトランザクションをコミットおよび検証するための要求を受信することに起因して、第２のノードに関連付けられたブロックチェーンシステム内の組織を特定し、組織が提案によって示されたスマートコントラクトトランザクションを実行できるかどうかを決定し、組織が提案によって示されたスマートコントラクトトランザクションを実行する権限を有していないことを示す判定に対して、第２のノードにエラーを返し、組織が提案によって示されたスマートコントラクトトランザクションを実行する権限を有することを示す判定に対して、図１０に例示されているように、オーダリングサービスによって提供されるトランザクションの順序を実行するように構成される。

【００８２】

オーダリングサービス１３０のための例では、プロセッサ（複数可）１７１０は、スマートコントラクトトランザクションをコミットするための他の要求とともに第２のノードからスマートコントラクトトランザクションをコミットするための要求を実行するための順序を決定し、例として図１に示されるように、第１のノードに順序を提供するように構成され得る。

【００８３】

監査ノード１１１０などの第３のノードの一例では、監査プロセスを実行するように構成されたプロセッサ（複数可）１７１０を備え、監査プロセスは、図１２に示されるように、スマートコントラクトトランザクションおよびスマートコントラクトトランザクションに関連付けられたプライベートデータを受信することと、スマートコントラクトトランザクションおよびプライベートデータに対して整合性チェックを実行することとを含む。

【００８４】

読み取り専用組織１０１に属する装置の一例では、プロセッサ１７１０（複数可）は、テスト環境の開始を受信するために、スマートコントラクトトランザクション内の提案を開始したブロックチェーンシステム内の組織のアイデンティティおよびエンドースメントを、テスト環境を管理するテスト環境マネージャによって発行されたアイデンティティおよびエンドースメントに置き換え、システムの構成の変更を示すスマートコントラクトトランザクションのために、図１５に示されるように、スマートコントラクトトランザクション内の組織情報によって示されるブロックチェーンシステム内の組織のトラストルートを変換するように構成されている。

【００８５】

読み取り専用組織１０１に属するそのような装置の一例では、プロセッサ（複数可）１７１０は、テスト環境上で新しいブロックチェーンネットワークを起動するように構成され、この起動は、テストネットワークランチャーによって容易にされ、テスト環境マネージャによって変換された台帳およびテスト環境マネージャによって変換されたステートデータベースを、新しいブロックチェーンネットワークの各ブロックチェーンノードにコピーすることと、図１６に示されるように、スマートコントラクトトランザクションに関連するブロックチェーンシステム内の組織に対応するテスト環境マネージャによって変換されたプライベートデータを、組織に対応する新しいブロックチェーンネットワーク内の各ブロックチェーンノードにコピーすることと、を含む。

【００８６】

詳細な説明の一部は、コンピュータ内の操作のアルゴリズムおよび記号的表現の観点から提示されている。これらのアルゴリズムの記述および記号的な表現は、データ処理技術の当業者が、その技術革新の本質を当業者に伝えるために使用する手段である。アルゴリズムとは、所望の最終状態や結果に至るまでの一連の定義されたステップのことである。実装例では、実行されるステップは、有形の結果を得るために有形の量を物理的に操作する必要がある。

【００８７】

特に明記しない限り、議論から明らかなように、本明細書を通じて、「処理」、「コンピューティング」、「計算」、「決定」、「表示」などの用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的（電子的）量として表されるデータを、コンピュータシステムのメモリまたはレジスタまたは他の情報記憶、送信、または表示デバイス内の物理的量として同様に表される他のデータに操作および変換する、コンピュータシステムまたは他の情報処理デバイスの動作およびプロセスを含むことができることが理解される。

【００８８】

実装例は、本明細書の操作を実行するための装置に関するものでもある。この装置は、必要な目的のために特別に構築されていてもよいし、１つまたは複数のコンピュータプログラムによって選択的に起動または再構成される１つまたは複数の汎用コンピュータを含んでいてもよい。このようなコンピュータプログラムは、コンピュータ読取可能な記憶媒体やコンピュータ読取可能な信号媒体などのコンピュータ読取可能な媒体に格納されていてもよい。コンピュータ読み取り可能な記憶媒体は、光ディスク、磁気ディスク、リードオンリーメモリ、ランダムアクセスメモリ、ソリッドステートデバイスおよびドライブなどの有形媒体、または電子情報を格納するのに適した他のタイプの有形または非一時的な媒体を含むことができるが、これらに限定されるものではない。コンピュータ読み取り可能な信号媒体は、搬送波などの媒体を含んでもよい。本明細書に示されたアルゴリズムおよび表示は、特定のコンピュータまたは他の装置に本質的に関係しない。コンピュータプログラムは、所望の実装の動作を実行する命令を含む純粋なソフトウェアの実装を含むことができる。

【００８９】

本明細書の実施例に従ったプログラムおよびモジュールを用いて様々な汎用システムを使用することができるが、所望の方法ステップを実行するために、より専門的な装置を構築することが便利であることが分かるかもしれない。さらに、本実施例は、任意の特定のプログラミング言語を参照して説明されていない。様々なプログラミング言語が、本明細書に記載された例示の実装の教示を実施するために使用され得ることが理解されるであろう。プログラミング言語の命令は、１つまたは複数の処理装置、例えば、中央処理装置（ＣＰＵ）、プロセッサ、またはコントローラによって実行されてもよい。

【００９０】

当技術分野で知られているように、上述の操作は、ハードウェア、ソフトウェア、またはソフトウェアとハードウェアの何らかの組み合わせによって実行することができる。実装例の様々な態様は、回路および論理デバイス（ハードウェア）を使用して実装されてもよいが、他の側面は、機械読み取り可能な媒体に格納された命令（ソフトウェア）を使用して実装されてもよく、これらの命令は、プロセッサによって実行された場合、プロセッサに、本願発明の実装形態を実行する方法を実行させるであろう。さらに、本願のいくつかの実装例は、ハードウェアのみで実行されてもよいが、他の実装例は、ソフトウェアのみで実行されてもよい。さらに、説明した様々な機能は、単一のユニットで実行することができ、あるいは、任意の数の方法で多数のコンポーネントに分散させることができる。ソフトウェアで実行する場合、方法は、コンピュータ可読媒体に格納された命令に基づいて、汎用コンピュータなどのプロセッサによって実行することができる。必要に応じて、命令は圧縮および／または暗号化された形式で媒体に保存することができる。

【００９１】

さらに、本願の他の実装例は、本願明細書の検討および本願の教示の実践から当業者に明らかになるであろう。記載された実装例の様々な形態および／またはコンポーネントは、単独でまたは任意の組み合わせで使用することができる。本願の真の範囲と主旨は、以下の特許請求の範囲によって示されており、本明細書および実装例は、例示としてのみ考慮されることが意図されている。

【図面】

【図１】

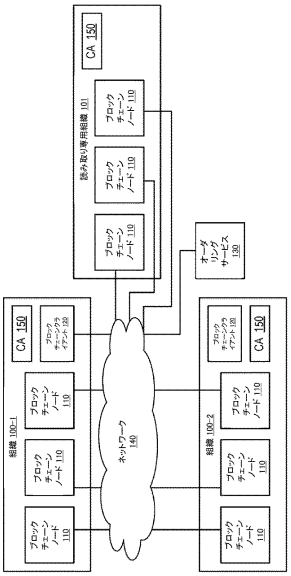

図1

【図２】

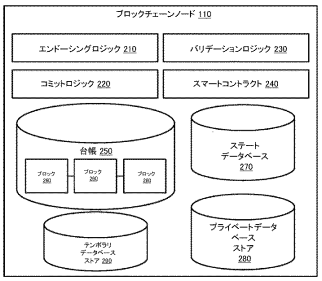

図2

【図３】

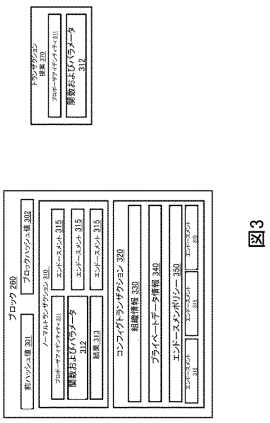

図3

【図４】

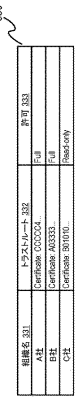

図4

【図５】

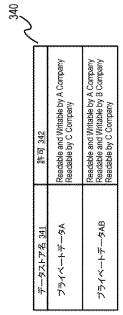

図5

【図６】

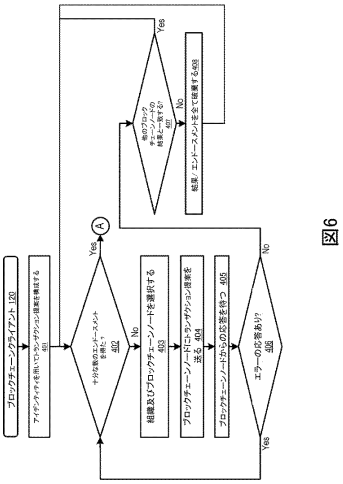

図6

【図７】

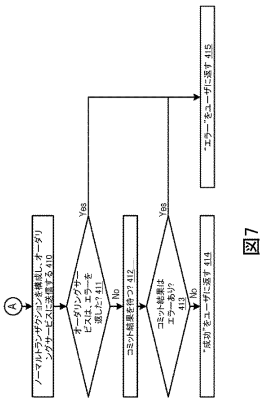

図7

【図８】

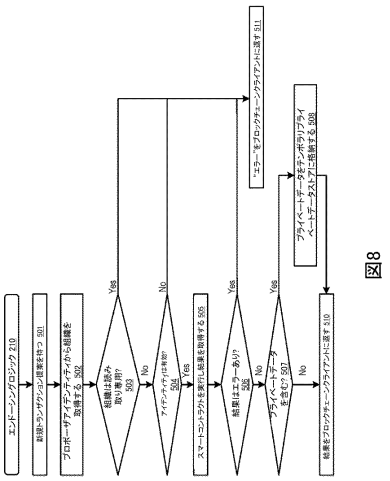

図8

【図９】

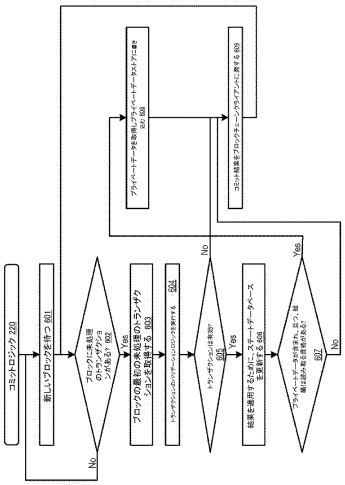

図9

【図１０】

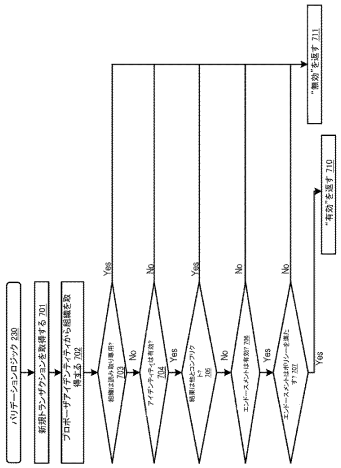

図10

【図１１】

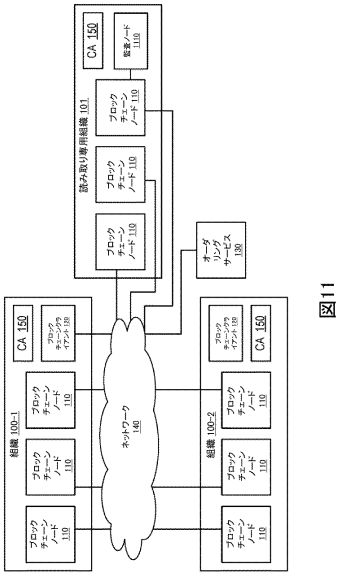

図11

【図１２】

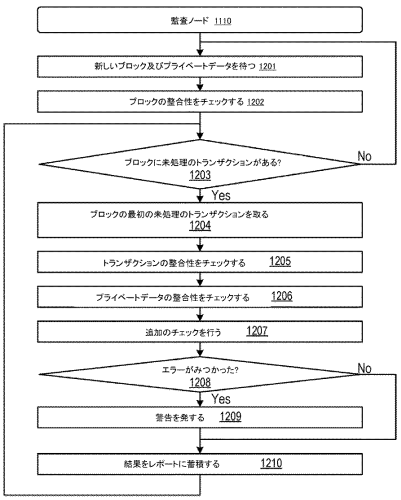

図12

【図１３】

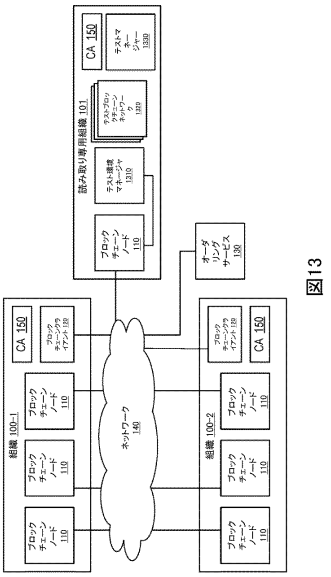

図13

【図１４】

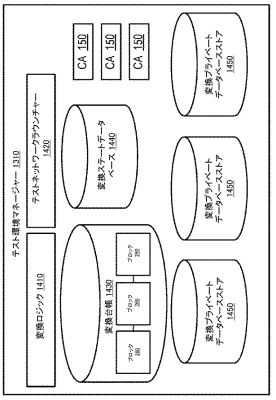

図14

【図１５】

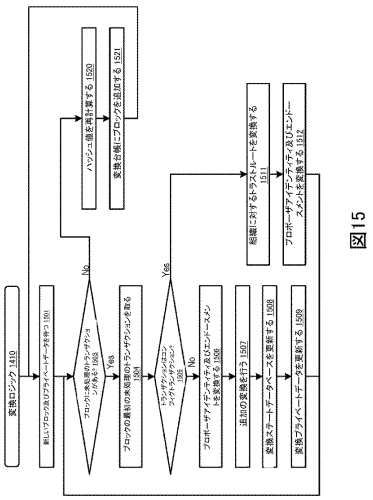

図15

【図１６】

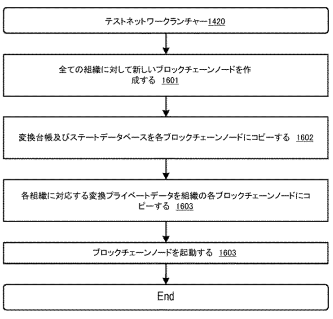

図16

【図１７】

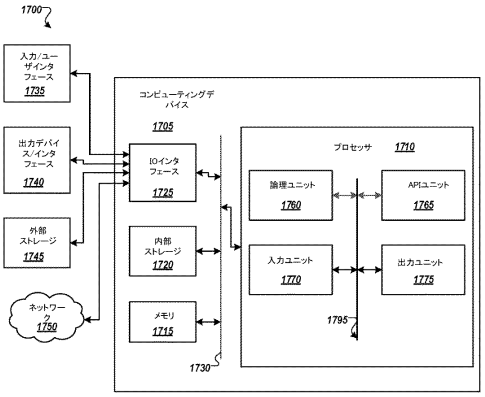
1700
入力/ユーザインタフェース
1735
出力デバイス/インタフェース
1740
外部ストレージ
1745
ネットワーク
1750
コンピューティングデバイス
1705
IOインタフェース
1725
内部ストレージ
1720
メモリ
1715
1730
プロセッサ
1710
論理ユニット
1760
APIユニット
1765
入力ユニット
1770
出力ユニット
1775
1795

図17

フロントページの続き

(56)参考文献　特開２０１９－１６０３１２（ＪＰ，Ａ）
特表２０２０－５０１２２０（ＪＰ，Ａ）
特表２０２０－５０７８２７（ＪＰ，Ａ）
特表２０２０－５０２６２１（ＪＰ，Ａ）

(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　２１／６０