(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2021-518705
(P2021-518705A)
(43)公表日　令和3年8月2日(2021.8.2)

(51)Int.Cl.			F I			テーマコード（参考）
H04L	9/32	(2006.01)	H04L	9/00	675Z	5B034
G09C	1/00	(2006.01)	G09C	1/00	640D	
G06F	11/20	(2006.01)	G06F	11/20	646	

審査請求 未請求　予備審査請求 未請求　（全 31 頁）

(21)出願番号　特願2020-550639 (P2020-550639)
(86)(22)出願日　平成31年3月8日(2019.3.8)
(85)翻訳文提出日　令和2年9月18日(2020.9.18)
(86)国際出願番号　PCT/EP2019/055894
(87)国際公開番号　WO2019/185329
(87)国際公開日　令和1年10月3日(2019.10.3)
(31)優先権主張番号　15/937,375
(32)優先日　平成30年3月27日(2018.3.27)
(33)優先権主張国・地域又は機関
米国(US)

(71)出願人　390009531
インターナショナル・ビジネス・マシーンズ・コーポレーション
INTERNATIONAL BUSINESS MACHINES CORPORATION
アメリカ合衆国１０５０４　ニューヨーク州　アーモンク　ニュー　オーチャード　ロード
New Orchard Road, Armonk, New York 10504, United States of America
(74)代理人　100108501
弁理士　上野　剛史

最終頁に続く

(54)【発明の名称】ブロックチェーン台帳のためのランタイム自己修正

(57)【要約】
例示の操作は、台帳検証スレッドを介して、分散型台帳のブロックのチェーン内に格納されている破損データ・ブロックを識別すること、分散型台帳へのアクセスを有する複数のブロックチェーン・ピアの中からブロックチェーン・ピアを選択し、選択されたピアから置換データ・ブロックを取得すること、ブロックのチェーンに関連する以前に格納された検証ブロックのうちの１つまたは複数に基づいて、置換データ・ブロックが有効であるかどうかを決定すること、および置換データ・ブロックが有効であると決定したことに応じて、分散型台帳上で破損データ・ブロックを置換データ・ブロックに置き換えることのうちの１つまたは複数を含むことができる。

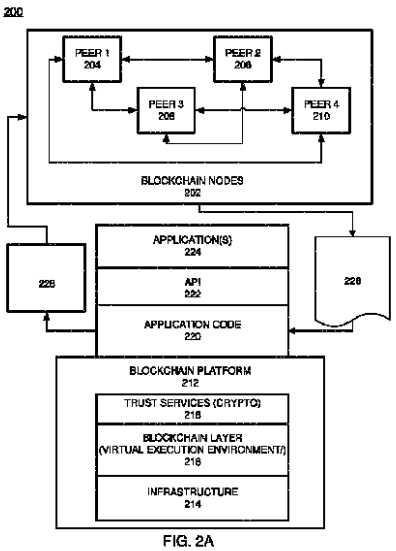


FIG. 2A

【特許請求の範囲】
【請求項１】
破損データ・ブロックを管理するためのコンピューティング・システムであって、前記システムが、
分散型台帳を格納するメモリと、
台帳検証スレッドを介して、前記分散型台帳のブロックのチェーン内に格納されている前記破損データ・ブロックを識別し、前記分散型台帳へのアクセスを有する複数のブロックチェーン・ピアの中からブロックチェーン・ピアを選択し、前記選択されたピアから置換データ・ブロックを取得し、前記ブロックのチェーンに関連する以前に格納された検証ブロックのうちの１つまたは複数に基づいて、前記置換データ・ブロックが有効であるかどうかを決定し、前記置換データ・ブロックが有効であるという前記決定に応じて、前記分散型台帳上で前記破損データ・ブロックを前記置換データ・ブロックに置き換えるように構成されたプロセッサと
を含む、コンピューティング・システム。
【請求項２】
前記台帳検証スレッドは、前記メモリで実行され、前記分散型台帳を構成するファイルとは別に格納されているコンパイル済みコードを有するプログラムを含む、請求項１に記載のコンピューティング・システム。
【請求項３】
前記プロセッサは、前記破損データ・ブロックが元のデータ・ブロックのブロック・サイズを変更したと決定された場合、前記ブロックのチェーンにおける次に続くデータ・ブロックについての置換データ・ブロックを取得するようにさらに構成される、請求項１または２のいずれかに記載のコンピューティング・システム。
【請求項４】
前記以前に格納された検証ブロックが、前記台帳検証スレッドを実行するブロックチェーン・ノードの前記分散型台帳のバックアップ・コピーに格納された正しいブロックを含む、請求項１ないし３のいずれかに記載のコンピューティング・システム。
【請求項５】
前記プロセッサが、前記分散型台帳のデータ・ブロックを順次スキャンし、ランタイムに生成された、前記順次スキャンされたデータ・ブロックのそれぞれのハッシュ値に基づいて、前記順次スキャンされたデータ・ブロックを検証して、前記破損データ・ブロックを識別するように構成される、請求項１ないし４のいずれかに記載のコンピューティング・システム。
【請求項６】
前記プロセッサが、前記分散型台帳のバックアップ・コピーとの前記分散型台帳のビット単位の比較を実行して、前記ビット単位の比較に基づいて前記破損データ・ブロックを識別するように構成される、請求項１ないし５のいずれかに記載のコンピューティング・システム。
【請求項７】
前記プロセッサが、前記分散型台帳から現在選択されているデータ・ブロックに関連するブロック・アクセス・パターンに基づいて前記分散型台帳からデータ・ブロックをプリフェッチし、前記プリフェッチされたデータ・ブロックのそれぞれのハッシュに基づいて前記プリフェッチされたデータ・ブロックを検証して、前記プリフェッチされたデータ・ブロックから前記破損データ・ブロックを識別するように構成される、請求項１ないし６のいずれかに記載のコンピューティング・システム。
【請求項８】
前記プロセッサが、前記分散型台帳のデータ・ブロックを前記それぞれのデータ・ブロックの以前に生成されたチェックサム値に基づいて順次検証して、前記チェックサム値に基づいて前記破損データ・ブロックを識別するように構成される、請求項１ないし７のいずれかに記載のコンピューティング・システム。

【請求項９】
前記プロセッサが、前記分散型台帳からデータ・ブロックをランダムに選択し、前記ランダムに選択されたデータ・ブロックのそれぞれのハッシュに基づいて前記ランダムに選択されたデータ・ブロックを検証して、前記ランダムに選択されたデータ・ブロックから前記破損データ・ブロックを識別するように構成される、請求項１ないし８のいずれかに記載のコンピューティング・システム。
【請求項１０】
破損データ・ブロックを管理するための方法であって、前記方法が、
台帳検証スレッドを介して、分散型台帳のブロックのチェーン内に格納されている前記破損データ・ブロックを識別することと、
前記分散型台帳へのアクセスを有する複数のブロックチェーン・ピアの中からブロックチェーン・ピアを選択し、前記選択されたピアから置換データ・ブロックを取得することと、
前記ブロックのチェーンに関連する以前に格納された検証ブロックのうちの１つまたは複数に基づいて、前記置換データ・ブロックが有効であるかどうかを決定することと、
前記置換データ・ブロックが有効であると決定したことに応じて、前記分散型台帳上で前記破損データ・ブロックを前記置換データ・ブロックに置き換えることと
を含む、方法。
【請求項１１】
前記台帳検証スレッドが、前記分散型台帳を構成するファイルとは別に格納されているコンパイル済みコードを有するメモリで実行されるプログラムを含む、請求項１０に記載の方法。
【請求項１２】
前記取得することは、前記破損データ・ブロックが元のデータ・ブロックのブロック・サイズを変更したと決定された場合、前記ブロックのチェーンにおける次に続くデータ・ブロックについての置換データ・ブロックを取得することをさらに含む、請求項１０または１１のいずれかに記載の方法。
【請求項１３】
前記以前に格納された検証ブロックが、前記台帳検証スレッドを実行するブロックチェーン・ノードの前記分散型台帳のバックアップ・コピーに格納された正しいブロックを含む、請求項１０ないし１２のいずれかに記載の方法。
【請求項１４】
前記台帳検証スレッドを介して前記破損データ・ブロックを前記識別することが、前記分散型台帳のデータ・ブロックを順次スキャンし、ランタイムに生成された、前記順次スキャンされたデータ・ブロックのそれぞれのハッシュ値に基づいて、前記順次スキャンされたデータ・ブロックを検証して、前記破損データ・ブロックを識別することを含む、請求項１０ないし１３のいずれかに記載の方法。
【請求項１５】
前記台帳検証スレッドを介して前記破損データ・ブロックを前記識別することが、前記分散型台帳のバックアップ・コピーとの前記分散型台帳のビット単位の比較を実行して、前記ビット単位の比較に基づいて前記破損データ・ブロックを識別することを含む、請求項１０ないし１４のいずれかに記載の方法。
【請求項１６】
前記台帳検証スレッドを介して前記破損データ・ブロックを前記識別することが、前記分散型台帳から現在選択されているデータ・ブロックに関連するブロック・アクセス・パターンに基づいて前記分散型台帳からデータ・ブロックをプリフェッチし、前記プリフェッチされたデータ・ブロックのそれぞれのハッシュに基づいて前記プリフェッチされたデータ・ブロックを検証して、前記プリフェッチされたデータ・ブロックから前記破損データ・ブロックを識別することを含む、請求項１０ないし１５のいずれかに記載の方法。
【請求項１７】

前記台帳検証スレッドを介して前記破損データ・ブロックを前記識別することが、前記分散型台帳のデータ・ブロックを前記それぞれのデータ・ブロックの以前に生成されたチェックサム値に基づいて順次検証して、前記チェックサム値に基づいて前記破損データ・ブロックを識別することを含む、請求項１０ないし１６のいずれかに記載の方法。

【請求項１８】

前記台帳検証スレッドを介して前記破損データ・ブロックを前記識別することが、前記分散型台帳からデータ・ブロックをランダムに選択し、前記ランダムに選択されたデータ・ブロックのそれぞれのハッシュに基づいて前記ランダムに選択されたデータ・ブロックを検証して、前記ランダムに選択されたデータ・ブロックから前記破損データ・ブロックを識別することを含む、請求項１０ないし１７のいずれかに記載の方法。

【請求項１９】

破損データ・ブロックを管理するためのコンピュータ・プログラム製品であって、前記コンピュータ・プログラム製品が、

処理回路によって可読であり、請求項１０ないし１８のいずれかに記載の方法を実行するために前記処理回路によって実行するための命令を格納するコンピュータ可読記憶媒体を含む、コンピュータ・プログラム製品。

【請求項２０】

コンピュータ可読媒体に格納され、デジタル・コンピュータの内部メモリにロード可能なコンピュータ・プログラムであって、前記プログラムがコンピュータで実行されるとき、請求項１０ないし１８のいずれかに記載の方法を実行するためのソフトウェア・コード部分を含む、コンピュータ・プログラム。

【発明の詳細な説明】

【技術分野】

【０００１】

本出願は、一般に、エラー検出に関し、より詳細には、分散型台帳（ブロックチェーンなどの）の破損データ・ブロックをランタイムに自動検出し、同じ分散型台帳を共有する別のコンピューティング・ノードに格納されているデータに基づいて破損データ・ブロックを自己修正するコンピューティング・ノードに関する。

【背景技術】

【０００２】

台帳は、一般に、トランザクションが記録されるエントリの会計簿として定義される。一方、分散型台帳は、多数のコンピューティング・ノードにわたり全体的にまたは部分的に複製されるデジタル台帳である。分散型台帳は暗号の性質によってセキュリティ保護することができ、暗号分散型台帳（ＣＤＬ）とも呼ばれる。ＣＤＬは、以下の特性、すなわち、不可逆性（例えば、トランザクションは、記録された後、取り消すことができない）、アクセス可能性（例えば、いかなる当事者も、ＣＤＬに全体的にまたは部分的にアクセスすることができる）、時系列およびタイムスタンプ付き（例えば、すべての参加当事者は、トランザクションがいつ台帳に追加されたか、およびどの順序で追加されたかを知っている）、合意ベース（トランザクションは、ネットワークの当事者によって一般に全員一致で是認された場合にのみ追加される）、検証可能性（トランザクションはすべて暗号的に確認することができる）などのうちの少なくともいくつかを有することができる。暗号分散型台帳の１つの非限定の例は、ブロックチェーンである。

【０００３】

ブロックチェーンなどの分散型台帳は、一般に、継続的に増加する記録のリストを格納する。ブロックチェーンは、金融トランザクションでしばしば使用されるが、商品およびサービス（すなわち、製品、パッケージ、ステータスなど）、デジタル通貨、株式、株主権、ソフトウェア・モデル、専有データ、および他の情報に関連する情報などの他のデータを格納することができる。分散化方式は、分散化ネットワークに権限および信頼を提供し、そのノードが、それらのトランザクションを公開のまたは秘密の「ブロック」に連続的におよび順次記録できるようにし、ブロックチェーンと呼ばれる固有の「チェーン」を

作り出す。ハッシュ・コードを介した暗号法を使用して、トランザクション・ソースの認証をセキュリティ保護し、中央の仲介者を排除する。分散型台帳は、その変更不能な性質により、改ざんおよび改訂からセキュリティ保護される。例えば、各ブロックは、タイムスタンプと、前のブロックへのリンクとを含むことができる。ブロックチェーンを使用して、情報を保持、追跡、転送、および確認することができる。ブロックチェーン・ネットワーク内のブロックチェーン・ピアは、１つまたは複数の他のブロックチェーン・ノードとの承認（endorsement）および合意プロトコルを通して、ブロックチェーン台帳へのトランザクションをトリガすることができ、１つのエンティティがそれ自体でブロックチェーン台帳を変更できないことを保証する。

【０００４】

従来、ブロックチェーン・ピア・ノードは、単に、新しいデータ・ブロックをブロックチェーン台帳に付加する。すなわち、ブロックチェーン・ノードは、一般に、以前に格納された既存のデータ・ブロックを修正またはそうでなければ勝手に変更するのではなく、単に新しいデータ・ブロックを追加する。この理由の１つは、分散型台帳の変更不能な性質、および新しいデータが分散型台帳に追加される前に必要とされる承認および合意プロセスのためである。しかしながら、ブロックチェーン台帳内のデータ・ブロックは、悪意のある攻撃、システム故障、ソフトウェア・エラーなどのような様々な理由のために時間とともに破損していくことがある。そのため、ブロックチェーン台帳内の好ましくないブロックを修復するメカニズムが必要とされる。

【０００５】

それゆえに、当技術分野では、前記の問題に対処することが必要である。

【発明の概要】

【０００６】

第１の態様から考察すると、本発明は、破損データ・ブロックを管理するためにコンピューティング・システムを提供し、このシステムは、分散型台帳を格納するメモリと、台帳検証スレッドを介して、分散型台帳のブロックのチェーン内に格納されている破損データ・ブロックを識別し、分散型台帳へのアクセスを有する複数のブロックチェーン・ピアの中からブロックチェーン・ピアを選択し、選択されたピアから置換データ・ブロックを取得し、ブロックのチェーンに関連する以前に格納された検証ブロックのうちの１つまたは複数に基づいて、置換データ・ブロックが有効であるかどうかを決定し、置換データ・ブロックが有効であるという決定に応じて、分散型台帳上で破損データ・ブロックを置換データ・ブロックに置き換えるように構成されたプロセッサとを含む。

【０００７】

第１の態様から考察すると、本発明は、破損データ・ブロックを管理する方法を提供し、この方法は、台帳検証スレッドを介して、分散型台帳のブロックのチェーン内に格納されている破損データ・ブロックを識別することと、分散型台帳へのアクセスを有する複数のブロックチェーン・ピアの中からブロックチェーン・ピアを選択し、選択されたピアから置換データ・ブロックを取得することと、ブロックのチェーンに関連する以前に格納された検証ブロックのうちの１つまたは複数に基づいて、置換データ・ブロックが有効であるかどうかを決定することと、置換データ・ブロックが有効であると決定したことに応じて、分散型台帳上で破損データ・ブロックを置換データ・ブロックに置き換えることとを含む。

【０００８】

さらなる態様から考察すると、本発明は、破損データ・ブロックを管理するためのコンピュータ・プログラム製品を提供し、コンピュータ・プログラム製品は、処理回路によって可読であり、本発明のステップを実行する方法を実行するために処理回路で実行するための命令を格納するコンピュータ可読記憶媒体を含む。

【０００９】

さらなる態様から考察すると、本発明は、コンピュータ可読媒体に格納され、デジタル・コンピュータの内部メモリにロード可能なコンピュータ・プログラムであって、プログ

ラムがコンピュータで実行されるとき、本発明のステップを実行するためのソフトウェア・コード部分を含む、コンピュータ・プログラムを提供する。

【００１０】

１つの例示の実施形態は、台帳検証スレッドを介して、分散型台帳のブロックのチェーン内に格納されている破損データ・ブロックを識別すること、分散型台帳へのアクセスを有する複数のブロックチェーン・ピアの中からブロックチェーン・ピアを選択し、選択されたピアから置換データ・ブロックを取得すること、ブロックのチェーンに関連する以前に格納された検証ブロックのうちの１つまたは複数に基づいて、置換データ・ブロックが有効であるかどうかを決定すること、および置換データ・ブロックが有効であると決定したことに応じて、分散型台帳上で破損データ・ブロックを置換データ・ブロックに置き換えることのうちの少なくとも１つを含む方法を提供することができる。

【００１１】

別の例示の実施形態は、分散型台帳を格納するメモリ、および台帳検証スレッドを介して、分散型台帳のブロックのチェーン内に格納されている破損データ・ブロックを識別し、分散型台帳へのアクセスを有する複数のブロックチェーン・ピアの中からブロックチェーン・ピアを選択し、選択されたピアから置換データ・ブロックを取得し、ブロックのチェーンに関連する以前に格納された検証ブロックのうちの１つまたは複数に基づいて、置換データ・ブロックが有効であるかどうかを決定し、置換データ・ブロックが有効であると決定したことに応じて、分散型台帳上で破損データ・ブロックを置換データ・ブロックに置き換えるように構成されたプロセッサのうちの少なくとも１つを含むシステムを提供することができる。

【００１２】

さらなる例示の実施形態は、プロセッサによって読み取られたとき、台帳検証スレッドを介して、分散型台帳のブロックのチェーン内に格納されている破損データ・ブロックを識別すること、分散型台帳へのアクセスを有する複数のブロックチェーン・ピアの中からブロックチェーン・ピアを選択し、選択されたピアから置換データ・ブロックを取得すること、ブロックのチェーンに関連する以前に格納された検証ブロックのうちの１つまたは複数に基づいて、置換データ・ブロックが有効であるかどうかを決定すること、および、置換データ・ブロックが有効であると決定したことに応じて、分散型台帳上で破損データ・ブロックを置換データ・ブロックに置き換えることのうちの少なくとも１つをプロセッサに実行させる命令を含む非一過性コンピュータ可読媒体を提供することができる。

【００１３】

次に、本発明の実施形態が、単に例として、添付図面を参照して説明される。

【図面の簡単な説明】

【００１４】

【図１】例示の実施形態による、複数のブロックチェーン・ピアを含むブロックチェーン・ネットワークを示す図である。

【図２】例示の実施形態による、分散型台帳と相互作用する台帳検証スレッドを示す図である。

【図３】例示の実施形態による、ピア・ノード・ブロックチェーン・アーキテクチャ構成を示す図である。

【図４】例示の実施形態による、破損ブロックを自己修正するためのブロックチェーン・ノード間のトランザクションの流れを示す図である。

【図５】例示の実施形態による、許可型ブロックチェーン・ネットワークを示す図である。

【図６】例示の実施形態による、台帳修正プロセスの複数の例を示す図である。

【図７】例示の実施形態による、分散型台帳の破損データ・ブロックを自己修正するための方法を示す流れ図である。

【図８】例示の実施形態による、本明細書で説明する１つまたは複数の操作に従ってブロックチェーンで様々な操作を実行するように構成された物理インフラストラクチャを示す

図である。
【図９】例示の実施形態による、ブロックチェーンでスマート・コントラクト条件を実施するように構成された、契約当事者と仲介サーバとの間のスマート・コントラクト構成を示す図である。
【図１０】例示の実施形態の１つまたは複数をサポートするように構成されたコンピュータ・システムを示す図である。
【発明を実施するための形態】
【００１５】
本明細書の図に概して記載および図示されている本構成要素は、多種多様な異なる構成で配列および設計されてもよいことが容易に理解されよう。添付の図に表されているような、方法、装置、非一過性コンピュータ可読媒体、およびシステムのうちの少なくとも１つの実施形態の以下の詳細な説明は、特許請求される本出願の範囲を限定することを意図するものではなく、単に選択された実施形態を表している。
【００１６】
本明細書の全体にわたって説明される本特徴、構造、または特性は、１つまたは複数の実施形態において任意の適切な方法で組み合わせることができる。例えば、本明細書の全体を通して、「例示の実施形態」、「いくつかの実施形態」という句、または他の同様の言語の使用は、実施形態に関連して記載される特定の特徴、構造、または特性が少なくとも１つの実施形態に含まれ得るという事実を指す。本明細書の全体を通して、「例示の実施形態において」、「ある実施形態において」、もしくは「他の実施形態において」という句、または他の同様の言語の出現は、必ずしも、すべてが同じグループの実施形態を指すのではなく、記載される特徴、構造、または特性は、１つまたは複数の実施形態において任意の適切な方法で組み合わされてもよい。
【００１７】
加えて、「メッセージ」という用語が実施形態の説明において使用されていることがあるが、本出願は、パケット、フレーム、データグラムなどのような多くのタイプのネットワーク・データに適用することができる。「メッセージ」という用語は、パケット、フレーム、データグラム、およびそれらの等価物をさらに含む。さらに、特定のタイプのメッセージおよびシグナリングが例示的な実施形態に示されることがあるが、それらは特定のタイプのメッセージに限定されず、本出願は特定のタイプのシグナリングに限定されない。
【００１８】
従来、ブロックチェーン・ピアは、分散型台帳を実装している複数のノードが新しいブロックについて合意に達すると、新しいデータ・ブロックを既存のブロックのチェーンに付加するかまたはそうでなければコミットすることによって、分散型台帳を単に変更する。しかしながら、ブロックチェーン・ピアは、分散型台帳の以前に格納されたブロックを検証することはできない。言い換えれば、ブロックチェーン・ピアは、分散型台帳が正確であることを保証することができない。その結果、破損データ・ブロックが分散型台帳のローカル・コピー内に存在し、ブロックチェーン・ネットワーク全体内で問題を引き起こすことがある。
【００１９】
例示の実施形態は、分散型台帳の以前に格納されたブロックを検証することができる台帳検証スレッドを実装することによってこれらの問題に解決策を与える方法、デバイス、ネットワーク、またはシステム、あるいはその組合せを提供する。台帳検証スレッドは、データ・ブロックが破損状態になったとき、様々な異なる検証方法を使用して検出することができる。破損データ・ブロックの識別に応じて、台帳検証スレッドは、同じチャネルの別のブロックチェーン・ピアから置換データ・ブロックを取得し、置換データ・ブロックを検証し、破損データ・ブロックを、検証された置換データ・ブロックに置き換えることができる。その結果、台帳検証スレッドは、台帳の１つまたは複数のブロックが悪意のあるアクティビティ、ソフトウェア・エラー、ハードウェア障害などにより破損状態にな

った場合に、分散型台帳を自己修復する（すなわち、自己修正する）ことができる。周期的な時間間隔、台帳への変更、管理者からの要求などのような様々なトリガが、ランタイム検証プロセスを開始させることができる。本明細書の説明および図は、ブロックチェーンに関して説明されているが、本出願は、任意のタイプの分散型台帳に等しく適用される。

【００２０】

台帳（デジタル台帳、分散型台帳、変更不能台帳などとも呼ばれる）は、ブロックチェーン・ネットワークによって実行およびコミットされたすべてのトランザクションの記録を含んでいるので、ブロックチェーン・システムの重要な構成要素である。台帳は、さらに、トランザクションの署名およびデータ・ブロックのハッシュ値などの他のメタデータを有する。それゆえに、台帳の無傷性および完全性は、ブロックチェーンが正しく効果的に機能するために重要である。典型的なブロックチェーン・ネットワークでは、各ブロックチェーン・ピア・ノードは、台帳の完全なコピーを維持し、それを記憶媒体に一連のファイルとして格納するが、それは、破損に対して脆弱である。破損は、例えば、悪意のあるユーザがピア・ノードにハッキングし、台帳を格納するファイルを故意に変更することがある、ハードウェア障害が生じることがある、ソフトウェア・バグが台帳ファイルを破損することがある、など、多くの理由のために起こることがある。

【００２１】

Ｈｙｐｅｒｌｅｄｇｅｒ　Ｆａｂｒｉｃなどの多くのブロックチェーン・システムでは、ピア・ノードは、単に台帳の末尾に新しいブロックを付加し、台帳の既存のブロックを定期的に検証しない方法で機能する。その結果、台帳の破損ブロックを時間内に検出することができず、それは、多くの理由でブロックチェーン・システムにとって有害となることがある。例えば、ピア開始時に、ピアは、破損した台帳に起因して起動し損なうことがある。別の例として、ピア・ランタイム中に、ブロックチェーン分析ワークロードが、ピア・ノードで実行されており、破損ブロックのトランザクションにアクセスする必要がある場合、トランザクションに正常にアクセスすることができない。したがって、分析ワークロードは、この破損ブロックの存在のために失敗することがある。

【００２２】

ブロックチェーン・システムの機能への破損ブロックの影響を低減するために、例示の実施形態は、ブロックチェーン台帳などのデジタル台帳のランタイム自己修正メカニズムに関する。様々な態様によれば、別個のスレッド（すなわち、台帳検証スレッド）が、定期的にピア・ノードで実行されており、このピアの台帳のブロックを検証する。破損ブロックが検出された場合、台帳検証スレッドは、破損ブロックの正しいバージョンを有し得る別のピア（すなわちピアＰ２）を選択することができ、他のピアに置換ブロックを要求する。Ｐ２は、要求を受け取ると、要求しているピアを検証し、要求しているピアが正当である場合、対応するブロックを送ることができる。次いで、Ｐ１は、置換ブロックを受け取った後、ブロックを検証し、受け取ったブロックが有効である場合には破損したものを置き換える。そうでなければ、Ｐ１は、Ｐ２から受け取ったブロックが有効でない場合、異なるピア（すなわちＰ３）に同じブロックを依頼することができる。システムは、ユーザ（例えば、管理者など）が、どのような頻度でスレッドを実行するかおよび台帳のどの部分を検証する必要があるかなどの多くのやり方で台帳検証スレッドを構成できるように設けられる。

【００２３】

記憶媒体上のファイルのセットである台帳と異なり、台帳検証スレッドは、ブロックチェーン・ピアのメモリで実行されるかまたはそうでなければブロックチェーン・ピアとつながりのあるプログラムである。プログラムのコンパイル済みコードが、台帳とは別に格納されてもよい。詳細は実装に依存することができる。例えば、台帳検証スレッドがピアの一部として実装される場合、台帳検証スレッド・プログラムは、ピアのプログラムと同じように保護する（すなわち、安全なコンテナ内の隔離された場所に格納する）ことができる。別の例として、台帳検証スレッドが独立したプログラムとして実装される場合、台

帳検証スレッドは、ピアのプログラムと違う独立したセキュリティ構成を有することができ、それによって、管理者は、より高いセキュリティ設定を構成することができる。どちらの場合も、台帳検証スレッド・プログラムは、台帳から完全に切り離すことができ、破損の原因または不正な行為者は、プログラムにアクセスできないことになる。さらに、台帳検証スレッドは、ブロックチェーンで取引するユーザに対して完全に隠されてもよい。すなわち、ブロックチェーン・ユーザは、スレッドと対話することができないが、しかしながら、ブロックチェーン管理者は、メカニズムを構成し、それを開始および停止することができる。

【００２４】

台帳検証スレッドがピアの一部として実装される場合、台帳検証スレッド・プログラムは、ピア・コードの一部としてインストールされ（追加され）、ピアと一緒に開始および停止することができる。別の例として、台帳検証スレッドが独立したプログラムとして実装される場合、管理者は、スレッドを独立したソフトウェアとして使用して、台帳に接続し、オンデマンドでアルゴリズムを実行することができる。さらに、台帳検証スレッドがピアの一部として実装される場合、検証は、バックグラウンド・プロセス／スレッドとすることができ、様々な可能な方法でトリガすることができ、その方法には、限定はしないが、以下が含まれる。（１）スレッドは、周期的に台帳を検証し、そこで、前のスキャンが終了した後、特定の期間にトリガされる、（２）スレッドは、ピア・ノードのリソース（すなわち、ＣＰＵ、メモリなど）の利用が閾値よりも低いときにトリガされ得る、（３）スレッドは、管理者によって明確にトリガされ得る。台帳検証スレッドが独立したプログラムとして実装される場合、管理者は、オンデマンドで台帳検証スレッドを実行することができる。

【００２５】

図１は、例示の実施形態による、複数のブロックチェーン・ピアを含むブロックチェーン・ネットワーク１００を示す。図１の例を参照すると、ネットワーク１００は、ブロックチェーン・ノード１１０、１１２、１１４、１１６、１１８、および１２０を含み、それらは、インターネット、プライベート・ネットワークなどのようなネットワーク１３０を介して接続される。この例では、ノード（１１０～１２０）のどれでも、ピア、エンドーサ（endorser）、オーダラ（orderer）などのような１つまたは複数の役割を果たすことができる。様々な態様によれば、ノード１１０～１２０のどれでも、それぞれのノードによって格納された分散型台帳のデータ・ブロックを確認／検証するために、そこにインストールされたおよびそこで実行される台帳検証スレッドを有することができる。

【００２６】

例えば、ブロックチェーン・ノード１１０は、ブロックチェーン・ノード１１０が格納した分散型台帳のコピーを確認するために、そこで実行される台帳検証スレッドを有することができる。破損ブロックが存在していると台帳検証スレッドが決定すると、台帳検証スレッドは、ブロックチェーン・ノード１１０に、他のノード１１２、１１４、１１６、１１８、および１２０のいずれかに置換ブロックの要求を送らせることができる。それに応じて、置換ブロックが取得され、ブロックチェーン・ノード１１０は、置換ブロックを確認することができる。成功裏に確認されると、ブロックチェーン・ノード１１０は、破損ブロックを置換ブロックに置き換えることができる。しかしながら、検証が成功しない場合、ブロックチェーン・ノード１１０は、別のノードから置換ブロックを要求することができる。

【００２７】

図２は、例示の実施形態による、分散型台帳１８０と相互作用する台帳検証スレッド１６０を示す。この例では、台帳検証スレッド１６０（すなわち、プログラムおよびコンパイル済みコード）と分散型台帳１８０のファイルとの両方は、台帳検証スレッド１６０と分散型台帳１８０とが互いに分離されるように、メモリ１５０において格納される。様々な実施形態によれば、台帳検証スレッド１６０は、このピアの分散型台帳１８０の完全性をチェックする、ブロックチェーン・ピア・ノードで実行されるプログラムを含むことが

できる。台帳検証スレッド１６０は、台帳上の破損された１つまたは複数のデータ・ブロックを検出するための破損検出器モジュール１６２と、台帳上のどのブロックが影響されているかを検出するための被影響ブロック・モジュール１６４と、破損ブロックに対する正しいブロックを１つまたは複数の隣接するピア・ノードから取得するためのリクエスタ・モジュール１６６と、少なくとも破損データ・ブロックを置換データ・ブロックと置き換える（および場合によっては他の後続のデータ・ブロックを置き換える）ための置換器モジュール１６８とを含むことができる。

【００２８】

破損ブロック検出モジュール１６２は、様々な破損検出プロセスを実施することができる。ピアは多くの台帳を含むことができ、各台帳はリンクされたリストに論理的に編成される。リンクされたリストの各項目はブロックを表し、同じ台帳の異なるブロックはハッシュ・ポインタによって接続される。各ブロックは、ヘッダ・セクションおよびデータ・セクションを含む。ブロック・ヘッダは、前のブロックのハッシュ値を含む。基本的に、ブロックＢは、以下の条件ｈａｓｈ（Ｂ）＝＝ｎｅｘｔＢｌｏｃｋ．Ｈｅａｄｅｒ．ｐｒｅｖＢｌｏｃｋＨａｓｈが満たされる場合、有効であると見なされる。そうでなければ、ブロックＢは破損ブロックと見なされる。

【００２９】

様々な実施形態によれば、多数の手法の中からの１つまたは複数の方法を使用して、台帳内の破損したブロックを検出することができる。例えば、破損検出は、１７１において、台帳を順次スキャンし、ハッシュ値と比較することによって各ブロックを検証することを含むことができるが、それは、ＣＰＵに負担をかけることがある。

【００３０】

別の例として、破損検出は、１７２において、以前にメモリ１５０に格納されたバックアップとの台帳のビット単位（bit-by-bit）の比較を実行することを含むことができる。この手法は、ハッシュ値を計算するよりも速いストリーミング方法で、台帳をバックアップと比較することができる。しかし、台帳バックアップは、いつも有効であるとは限らず、攻撃に対して脆弱である可能性がある。

【００３１】

別の例として、１７３において、破損検出は、現在要求されているブロックなどに基づいた分析ワークロードからのアクセス・パターンに基づいてデータ・ブロックをプリフェッチすることができる。この例では、いくつかのブロックをプリフェッチおよび検証することができる。プリフェッチ操作は、ブロック間の関係について通知するヒューリスティックに基づくことができる。このヒューリスティックに基づき、ブロックがアクセスされると、関連するブロックがプリフェッチおよび検証される。例えば、ヒューリスティックは、ブロック・アクセスの局所性原理に基づくことができ、ブロックがアクセスされると、近い将来に近くのブロックがアクセスされる確率が高いことが認められる。

【００３２】

別の例として、１７４において、破損検出器は、ブロックを検証するためにチェックサム・アルゴリズム（例えば、巡回冗長検査（ＣＲＣ）など）を実装することができる。この手法は、ブロックごとにチェックサムを作成し、このチェックサムを使用してブロックを検証することができる。ＣＲＣなどの高速チェックサム生成アルゴリズムは、ブロックのハッシュを計算するよりも速い。しかし、チェックサム・データも、攻撃に対して脆弱である可能性がある。ＣＲＣは、ハッシュ・ベース手法とは多少違うように動作することができる。この場合、ＣＲＣアルゴリズムを使用して、ブロック和の並列リスト（parallel list）を作成することができる。これらの和が、ハッシュの代わりにチェックされる。そうすることの利点は、この手法の速度が速いことである。データは、チェック・アルゴリズムとプロセスのメモリに保持されるので、これらは、台帳が脆弱である攻撃に対して脆弱ではない。

【００３３】

別の例として、１７５において、破損検出は、すべてではなくいくつかの検証すべきブ

ロックをランダムに選ぶ場合がある。この手法は、ＣＰＵの消費が台帳全体を検証することよりも少ないが、台帳のすべてのブロックの正当性を保証できるとは限らない確率的手法である。

【００３４】

ピアは、置換ブロックを受け取ると、既定のブロックチェーン・アクションであり得る、次に続くブロックに格納されたブロックのハッシュを使用することを含む様々な方法でブロックの有効性を決定することができる。別の例として、ピアは、ピアのメモリに格納されたブロックのハッシュを使用することができる。この例では、ピアは、起動時にピアを横断して検証していたときにこのハッシュを得ることができる。

【００３５】

ピアは様々な方法で選択できることも理解されるべきである。例えば、ブロックが破損ブロックとして検出された場合、ピア（すなわち、ピア１）は、正しいブロックを得るために別のピアに連絡をとる必要がある。台帳修正プロセスのより良好な性能を達成するために、ピアは、ネットワーク接続がより良好な別のピアを選ぶことができる。ピアは様々な場所にわたることがあり、様々なピアの間のネットワーク品質は時々変化する。より良好なネットワーク接続はブロック送信を促進し、それにより、台帳修正プロセスの性能を改善することができる。一例として、ネットワーク接続の品質は、「ｐｉｎｇ」の待ち時間によって測定することができる。別の例として、ピアは、それほどビジーでない別のピアを選ぶことができる。様々なピアで実行されているワークロードは同じではない。例えば、Ｈｙｐｅｒｌｅｄｇｅｒ　Ｆａｂｒｉｃでは、いくつかのピア・ノードは単にブロックをコミットするが、いくつかの他のピア・ノードは、さらに、トランザクションをシミュレートする必要がある。それゆえに、それほど集中的でないワークロードを実行しているピア・ノードを選ぶほうが良いことがあり、その結果、そのピアは、要求に対してより迅速に対応することができ、それは、さらに、台帳修正プロセスの性能を改善する。

【００３６】

本明細書で説明するように、ブロックチェーンは、互いに通信する多数のノードを含む分散型システムである。ブロックチェーンは、チェーンコードと呼ばれるプログラム（例えば、スマート・コントラクトなど）を操作し、状態および台帳データを保持し、トランザクションを実行する。いくつかのトランザクションは、チェーンコードで呼び出された操作である。概して、ブロックチェーン・トランザクションは、一般に、特定のブロックチェーン・メンバによって「承認（endorsed）」されなければならず、承認されたトランザクションのみが、ブロックチェーンにコミットされ、ブロックチェーンの状態に影響を与えることができる。承認されない他のトランザクションは無視される。管理機能およびパラメータのための１つまたは複数の特別なチェーンコードが存在することがあり、システム・チェーンコードと総称される。

【００３７】

ノードは、ブロックチェーン・システムの通信エンティティである。「ノード」は、異なるタイプの多数のノードを同じ物理サーバ上で実行できるという意味で、論理機能を実行することができる。ノードは、信頼ドメインにおいてグループ化され、ノードを様々な方法で制御する論理エンティティに関連付けられる。ノードには、トランザクション呼出し（Transaction-invocation）をエンドーサ（例えば、ピア）にサブミットし、トランザクション提案（Transaction-proposal）を順序付けサービス（例えば、順序付けノード）にブロードキャストするクライアントまたはサブミッティング・クライアント・ノードなどの様々なタイプが含まれてもよい。別のタイプのノードは、クライアントがサブミットしたトランザクションを受け取り、トランザクションをコミットし、ブロックチェーン・トランザクションの台帳の状態およびコピーを維持することができるピア・ノードである。ピアは、必要条件ではないが、エンドーサの役割を有することもできる。順序付けサービス・ノードまたはオーダラは、すべてのノードのための通信サービスを実行するノードであり、トランザクションをコミットし、またブロックチェーンのワールドステートを変更するときのシステムのピア・ノードの各々へのブロードキャストなどの配信保証（deli

very guarantee）を実装し、これは、通常、制御およびセットアップ情報を含む初期のブロックチェーン・トランザクションの別名である。

【００３８】

本明細書で説明するように、台帳は、ブロックチェーンのすべての状態遷移の順序付けられた改ざん防止付き記録である。状態遷移は、参加当事者（例えば、クライアント・ノード、順序付けノード、エンドーサノード、ピア・ノードなど）によってサブミットされたチェーンコード呼出し（すなわち、トランザクション）から生じてもよい。トランザクションは、作成、更新、削除などのような１つまたは複数のオペランドとして台帳にコミットされる１組のアセット・キーと値のペアをもたらすことができる。台帳は、変更不能な順序付け済み記録をブロックに格納するために使用されるブロックチェーン（チェーンとも呼ばれる）を含む。台帳は、ブロックチェーンの現在の状態を維持する状態データベースをさらに含む。一般に、チャネル当たり１つの台帳が存在する。各ピア・ノードは、それらのピア・ノードがメンバであるチャネルごとに台帳のコピーを維持する。

【００３９】

チェーンは、ハッシュ・リンクされたブロックとして構造化されたトランザクション・ログであり、各ブロックは、Ｎ個のトランザクションのシーケンスを含み、ここで、Ｎは１以上である。ブロック・ヘッダは、ブロックのトランザクションのハッシュ、ならびに前のブロックのヘッダのハッシュを含む。このようにして、台帳のすべてのトランザクションは、順序付けられ、暗号によって相互にリンクされ得る。その結果、ハッシュ・リンクを壊さずに、台帳データを改ざんすることはできない。最も最近追加されたブロックチェーン・ブロックのハッシュは、それ以前に生じたチェーンのあらゆるトランザクションを表し、すべてのピア・ノードが一貫した信頼のできる状態であることを保証することを可能にする。チェーンは、ブロックチェーン・ワークロードの追記のみの性質（append-only nature）を効率的にサポートするピア・ノード・ファイル・システム（すなわち、ローカル、付属ストレージ、クラウドなど）に格納されてもよい。

【００４０】

変更不能な台帳の現在の状態は、チェーン・トランザクション・ログに含まれるすべてのキーの最新の値を表す。現在の状態は、チャネルに知られている最新のキー値を表すので、時には、ワールドステートと呼ばれる。チェーンコード呼出しは、台帳の現在の状態のデータに対してトランザクションを実行する。これらのチェーンコードの相互作用を効率的にするために、最新のキーの値を状態データベースに格納することができる。状態データベースは、単に、チェーンのトランザクション・ログへのインデックス付けされたビューとすることができ、それゆえに、それは、チェーンからいつでも再生成することができる。状態データベースは、ピア・ノード開始時に、およびトランザクションが受入れられる前に、自動的に回復されてもよい（または必要ならば生成されてもよい）。

【００４１】

図３は、例示の実施形態による、ブロックチェーン・アーキテクチャ構成２００を示す。図３を参照すると、ブロックチェーン・アーキテクチャ２００は、特定のブロックチェーン要素、例えば、ブロックチェーン・ノード２０２のグループを含むことができる。ブロックチェーン・ノード２０２は、１つまたは複数のノード２０４～２１０を含むことができる。（４つのノードが、単に例として示されている）。これらのノードは、ブロックチェーン・トランザクション追加および検証プロセス（合意）などのいくつかのアクティビティに参加する。ブロックチェーン・ノード２０４～２１０の１つまたは複数は、トランザクションを承認することができ、アーキテクチャ２００のすべてのブロックチェーン・ノードに順序付けサービスを提供することができる。ブロックチェーン・ノードは、ブロックチェーン認証を開始し、ブロックチェーン層２１６に格納されたブロックチェーン変更不能台帳に書き込むことを要求することができ、そのコピーが、さらに、下支えの物理インフラストラクチャ２１４に格納されてもよい。ブロックチェーン構成は、格納されたプログラム／アプリケーション・コード２２０（例えば、チェーンコード、スマート・コントラクトなど）にアクセスおよび実行するためにアプリケーション・プログラム・イ

ンタフェース（ＡＰＩ）２２２にリンクされた１つまたは複数のアプリケーション２２４を含むことができ、それは、参加者によって求められたカスタマイズ化構成に従って作成されてもよく、それ自体の状態を維持し、それ自体のアセットを制御し、外部情報を受け取ることができる。これは、トランザクションとしてデプロイされ、分散型台帳への追記を介して、すべてのブロックチェーン・ノード２０４～２１０にインストールされ得る。

【００４２】

ブロックチェーン・ベースまたはプラットフォーム２１２は、ブロックチェーン・データ、サービス（例えば、暗号トラスト・サービス、仮想実行環境など）、および、下支えの物理コンピュータ・インフラストラクチャの様々な層とを含むことができ、これらは、新しいトランザクションを受け取り格納し、データ・エントリにアクセスしようとしている監査役にアクセスを与えるために使用することができる。ブロックチェーン層２１６は、プログラム・コードを処理し、物理インフラストラクチャ２１４に関与するのに必要な仮想実行環境へのアクセスを提供するインタフェースを公開することができる。暗号トラスト・サービス２１８を使用して、アセット交換トランザクションなどのトランザクションを確認し、情報を秘密に保つことができる。

【００４３】

図３のブロックチェーン・アーキテクチャ構成は、ブロックチェーン・プラットフォーム２１２によって公開された１つまたは複数のインタフェースおよび提供されるサービスを介して、プログラム／アプリケーション・コード２２０を処理および実行することができる。コード２２０は、ブロックチェーン・アセットを制御することができる。例えば、コード２２０は、データを格納および転送することができ、スマート・コントラクトおよび条件またはその実行の対象である他のコード要素に関連付けられたチェーンコードの形態でノード２０４～２１０によって実行されてもよい。非限定の例として、スマート・コントラクトは、リマインダ、更新、または変更、更新などの対象となる他の通知、あるいはその組合せを実行するために作成することができる。スマート・コントラクトは、それ自体を使用して、認定とアクセス要件、および台帳の使用に関連するルールを識別することができる。例えば、情報２２６は、ブロックチェーン層２１６に含まれる１つまたは複数の処理エンティティ（例えば、仮想マシン）で処理することができる。結果２２８は、台帳に追加されるデータのブロックを含むことができる。物理インフラストラクチャ２１４を利用して、本明細書に記載されたデータまたは情報のうちのいずれかを取得することができる。

【００４４】

チェーンコード内で、スマート・コントラクトを、高レベル・アプリケーションおよびプログラミング言語を介して作成し、次いで、ブロックチェーン内のブロックに書き込むことができる。スマート・コントラクトは、ブロックチェーン（例えば、ブロックチェーン・ピアの分散型ネットワーク）により、登録、格納、または複製、あるいはその組合せが行われる実行可能コードを含むことができる。トランザクションは、スマート・コントラクトに関連する条件が満たされたことに応じて実行できるスマート・コントラクト・コードの実行である。スマート・コントラクトの実行は、デジタル・ブロックチェーン台帳の状態への信頼のできる変更をトリガすることができる。スマート・コントラクトの実行によって生じたブロックチェーン台帳への変更は、１つまたは複数の合意プロトコルを通してブロックチェーン・ピアの分散型ネットワークの全体にわたって自動的に複製され得る。

【００４５】

スマート・コントラクトは、キー－値のペアのフォーマットでブロックチェーンにデータを書き込むことができる。さらに、スマート・コントラクト・コードは、ブロックチェーンに格納された値を読み出し、それをアプリケーション操作で使用することができる。スマート・コントラクト・コードは、様々な論理演算の出力をブロックチェーンに書き込むことができる。コードを使用して、仮想マシンまたは他のコンピューティング・プラットフォームで一時的なデータ構造を作成することができる。ブロックチェーンに書き込ま

れたデータは、公開することができ、または暗号化し秘密として維持することができ、あるいはその両方を行うことができる。スマート・コントラクトによって使用／生成された一時データは、提供された実行環境によってメモリに保持され、次いで、ブロックチェーンで必要とされるデータが識別された後、削除される。

【００４６】

チェーンコードは、追加の機能とともに、スマート・コントラクトのコード解釈を含むことができる。本明細書で説明するように、チェーンコードは、合意プロセスの間にチェーン検証ソフトによって一緒に実行および検証される、コンピューティング・ネットワークに配置されたプログラム・コードとすることができる。チェーンコードは、ハッシュを受け取り、以前に格納された特徴抽出器の使用によって作成されたデータ・テンプレートに関連付けられたハッシュをブロックチェーンから取得する。ハッシュ識別子のハッシュと、格納された識別子テンプレート・データから作成されたハッシュとが一致した場合、チェーンコードは、要求されたサービスに認定キーを送る。チェーンコードは、暗号の詳細に関連付けられたブロックチェーン・データに書き込むことができる。

【００４７】

図４は、例示の実施形態による、破損ブロックを自己修正するためのブロックチェーン・ノード間のトランザクションの流れ２５０を示す。この例では、ブロックチェーン・ノード２０４は、ブロックチェーン・ノード２０４によって管理される分散型台帳の破損データ・ブロックを自動検出し自己修復する。ここでは、破損した台帳は、ブロックチェーン・ノード２０４によって格納されているローカル・コピーであり得る。２３１において、ブロックチェーン・ノード２０４で実行される台帳検証スレッドは、図２の１７１～１７５で上述した方法のうちの１つまたは複数を使用して、台帳検証プロセスを実行する。２３２において、ブロックチェーン・ノード２０４の台帳検証スレッドは、破損ブロックを検出する。２３３において、ブロックチェーン・ノード２０４は、置換ブロックを受け取るべきブロックチェーン・ノード２０６を識別および選択し、置換ブロックの要求を送信する。置換ブロックの要求は、破損ブロックのＩＤを含むことができ、１つまたは複数の後続ブロックのＩＤをさらに含むことができる。要求がブロックチェーン・ノード２０６に到着すると、要求は、ブロックチェーン・ノード２０６の台帳検証スレッドによって処理されるのではなく、代わりに、Ｈｙｐｅｒｌｅｄｇｅｒ　Ｆａｂｒｉｃに存在し、そのような要求をリッスンし、それに応じてブロックを戻す組み込みルーチン／機能によって処理され得る。

【００４８】

２３４において、選択されたブロックチェーン・ノード２０６は、要求元のブロックチェーン・ノード２０４を、１つまたは複数の暗号キー、ブロック・データなどを使用して検証し、置換ブロックをブロックチェーン・ノード２０４に送信する。ここで、ブロックチェーン・ノード２０４は、２３５において、置換ブロックを検証し、ブロックチェーン・ノード２０４は、そのブロックが有効である場合、破損ブロックをブロックチェーン・ノード２０６から受け取った置換ブロックと置き換えることができる。しかしながら、この例では、２３６において、ブロックチェーン・ノード２０４の台帳検証スレッドは、置換ブロックが有効でないと決定する。それゆえに、２３７において、台帳検証スレッドは、次のブロックチェーン・ノード２０８を選択し、それにより、２３７において、ブロックチェーン・ノード２０４は、置換ブロックの要求を次のブロックチェーン・ノード２０８に送る。２３８において、ブロックチェーン・ノード２０８は、ブロックチェーン・ノード２０４を検証し、ブロックチェーン・ノード２０４によって要求された置換ブロックのコピーを送信する。２３９において、ブロックチェーン・ノード２０４で実行されている台帳検証スレッドは、置換ブロックの有効性を確認し、２４０において破損ブロックを置き換える。

【００４９】

図５は、分散型の分散されたピア・ツー・ピア・アーキテクチャと、ユーザの役割および許可を管理する認証局３１８とを特徴とする許可型ブロックチェーン・ネットワーク３

００の一例を示す。この例では、ブロックチェーン・ユーザ３０２は、許可型ブロックチェーン・ネットワーク３１０にトランザクションをサブミットすることができる。この例では、そのトランザクションは、デプロイ、呼出し、照会とすることができ、ＳＤＫを活用するクライアント側アプリケーションを通して、ＲＥＳＴ　ＡＰＩなどによって直接発行され得る。信頼できるビジネス・ネットワークは、監査役（例えば、米国の株式市場の証券取引委員会）などの規制当局システム３１４へのアクセスを与えることができる。一方、ノード３１２のブロックチェーン・ネットワーク・オペレータ・システムは、規制当局システム３１４を「監査役」としておよびブロックチェーン・ユーザ３０２を「クライアント」として登録することなどのメンバの許可を管理する。監査役は、台帳の照会にのみ制限されることがあるが、一方、クライアントは、特定のタイプのチェーンコードをデプロイし、呼び出し、照会することを認可されることがある。

【００５０】

ブロックチェーン開発者システム３１６は、チェーンコードおよびクライアント側アプリケーションを書く。ブロックチェーン開発者システム３１６は、ＲＥＳＴインタフェースを通してネットワークにチェーンコードを直接配置することができる。従来のデータ・ソース３３０からの資格情報をチェーンコードに含めるために、開発者システム３１６は、データにアクセスするのにアウトオブバンド接続を使用することができる。この例では、ブロックチェーン・ユーザ３０２は、ピア・ノード３１２を通してネットワークに接続する。トランザクションを続行する前に、ピア・ノード３１２は、認証局３１８からユーザの登録およびトランザクション証明書を取得する。場合によっては、ブロックチェーン・ユーザは、許可型ブロックチェーン・ネットワーク３１０で取引するために、これらのデジタル証明書を所有しなければならない。一方、チェーンコードを駆動しようとするユーザは、従来のデータ・ソース３３０で資格情報を確認するように要求されることがある。ユーザの認定を確認するために、チェーンコードは、従来の処理プラットフォーム３２０を通してこのデータへのアウトオブバンド接続を使用することができる。

【００５１】

図６は、例示の実施形態による、台帳修正プロセスの複数の例を示す。この例では、ブロックチェーン台帳４００は、３つのファイル（すなわち、ファイル１、ファイル２、およびファイル３）を含む複数のファイルから構成される。この例では、台帳検証スレッドは、第２のファイル（ファイル４１０）に格納されている破損データ・ブロック（ブロックＡ）を検出する。Ｈｙｐｅｒｌｅｄｇｅｒ　Ｆａｂｒｉｃなどのブロックチェーン・システムでは、台帳は、複数の固定サイズファイルとして物理的に編成され、ブロックは、これらのファイルに連続した形態で格納される。図６の例に示されるように、第２のファイル４１０のブロックＡは壊れたブロックとして検出される。一方、ブロックＡ’は、別のピアから取得され検証された正しいブロックである。ブロックＡのサイズがブロックＡ’のサイズと等しくない場合、ブロックＡをブロックＡ’と単に置き換えると、ブロックＢに上書きされるか、またはブロックＡとブロックＢとの間にギャップが残される（そしてブロックＢに加えて任意の他の後続ブロックに問題を引き起こす）ことがある。それゆえに、台帳修正プロセスは、最初に、第２のファイル４１０のサイズが変わったかどうかを決定／チェックすることができる。

【００５２】

図６の例に示されるように、台帳検証スレッドによって実行される台帳修正プロセスは、ブロックＡの後に続く第２のファイル４１０のすべてのブロックを置き換えることがある。例えば、プロセス４１１において、台帳修正プロセスは、データ・ブロックＡのサイズが本来あるべきサイズよりも小さいので、ブロックＡ、ブロックＢ、およびブロックＣを置き換える。一方、プロセス４１２の例では、台帳修正プロセスは、データ・ブロックＡのサイズが本来あるべきサイズよりも大きいので、ブロックＡ、ブロックＢ、およびブロックＣを置き換える。その他に、データ・ブロックＡのサイズが変わらない（そして全ファイル・サイズを変更しない）場合、ブロックＡのみを上書きすることができる。

【００５３】

理解されるように、ブロックは、ファイル・システムのファイルに格納される。一般に、１つのファイルは、連続的な一連のブロックを含む。それゆえに、悪意のあるユーザが、ブロックＡを含むファイルを開き、いくつかのランダムなバイトをそのファイルのブロックＡがある場所に追加する場合、ブロックＡは、プロセス４１２のケースのように大きくなることになる。同様に、悪意のあるユーザがそのファイルのいくつかのバイトを削除する場合、ブロックＡはプロセス４１１のケースのように小さくなることになる。データ・ブロックＡをデータ・ブロックＡ’と置き換える場合、ブロックＡおよびブロックＡ’は、ファイルの同じオフセットで開始する必要があるが、しかしながら、ブロックＡ’のサイズはブロックＡのサイズよりも大きく、ブロックＡとブロックＢとが元の台帳ファイルで隣接している。したがって、ブロックＡがブロックＡ’と置き換えられる場合、ブロックＢの一部がブロックＡ’によって上書きされることになり、それは許容できない。

【００５４】

別の例として、ブロックＡが破損して大きくなった場合、ブロックＡをブロックＡ’に単に置き換えると、ブロックＡ’とブロックＢとの間にファイル欠陥（file hole）を残すことになり、それは、やはり、台帳の完全性を壊す。その上、ブロックＡが破損されて、大きくなるかまたは小さくなる場合、これにより、そのファイルのすべての後続のブロックがオフセットされる。それゆえに、この場合、ブロックＡしか破損されていないにもかかわらず、その特定のファイルのすべての後続のブロック（すなわち、この例では、ブロックＢおよびブロックＣ）が、台帳全体の完全性を保証するために、置き換えられるべきである。したがって、ブロックＢおよびＣが破損されなかったとしても、ファイル全体を修正するために、ブロックはすべて、ファイルの連続性を保証するように正しいシーケンスに書き直される必要がある。

【００５５】

一方、プロセス４１３の第３のケースでは、ブロックＡは破損されているが、そのサイズが変わっていない。言い換えれば、ブロックＡ’のサイズはブロックＡのサイズに等しく、このファイルのすべての後続のブロックのオフセットは同じままである。それゆえに、ブロックＡをブロックＡ’に単に置き換えることにより、破損ブロックＡは直されることになる。同時に、ブロックＢは上書きされないことになり、そしてまたブロックＡ’とブロックＢとの間にファイル欠陥は存在しない。

【００５６】

図７は、例示の実施形態による、分散型台帳の破損データ・ブロックを自己修正するための方法５００を示す。例えば、方法５００は、クラウド・プラットフォーム、サーバ、デスクトップ・コンピュータ、ユーザ・デバイスなどで実装されてもよいブロックチェーン・ピア・ノードなどのコンピューティング・ノードによって実行することができる。図７を参照すると、５１０において、この方法は、台帳検証スレッドを介して、分散型台帳のブロックのチェーン内に格納されている破損データ・ブロックを識別することを含むことができる。例えば、台帳検証スレッドは、分散型台帳を構成するファイルとは別に格納されているコンパイル済みコードを有するメモリで実行されるプログラムを含むことができる。分散型台帳ファイルは、分散型台帳と対話しているユーザ／ノードから分離されるか、またはそうでなければ隠されてもよい。

【００５７】

様々な態様によれば、破損した１つまたは複数のデータ・ブロックは、様々な方法のいずれかによって検出することができる。例えば台帳検証スレッドを介して破損データ・ブロックを識別することは、分散型台帳のデータ・ブロックを順次スキャンし、ランタイムに生成された、順次スキャンされたデータ・ブロックのそれぞれのハッシュ値に基づいて、順次スキャンされたデータ・ブロックを検証して、破損データ・ブロックを識別することを含むことができる。別の例として、台帳検証スレッドを介して破損データ・ブロックを識別することは、分散型台帳のバックアップ・コピーとの分散型台帳のビット単位の比較を実行して、ビット単位の比較に基づいて破損データ・ブロックを識別することを含むことができる。別の例として、台帳検証スレッドを介して破損データ・ブロックを識別す

ることは、分散型台帳から現在選択されているデータ・ブロックに関連するブロック・アクセス・パターンに基づいて分散型台帳からデータ・ブロックをプリフェッチし、プリフェッチされたデータ・ブロックのそれぞれのハッシュに基づいてプリフェッチされたデータ・ブロックを検証して、プリフェッチされたデータ・ブロックから破損データ・ブロックを識別することを含むことができる。別の例として、台帳検証スレッドを介して破損データ・ブロックを識別することは、分散型台帳のデータ・ブロックをそれぞれのデータ・ブロックの以前に生成されたチェックサム値に基づいて順次検証して、チェックサム値に基づいて破損データ・ブロックを識別することを含むことができる。別の例として、台帳検証スレッドを介して破損データ・ブロックを識別することは、分散型台帳からデータ・ブロックをランダムに選択し、ランダムに選択されたデータ・ブロックのそれぞれのハッシュに基づいてランダムに選択されたデータ・ブロックを検証して、ランダムに選択されたデータ・ブロックから破損データ・ブロックを識別することを含むことができる。

【００５８】

５２０において、この方法は、分散型台帳へのアクセスを有する複数のブロックチェーン・ピアの中からブロックチェーン・ピアを選択し、選択されたピアから置換データ・ブロックを取得することを含むことができる。ピアは、ランダムに選択されてもよく、または現在のスループット、場所、現在のワークロード、ブロックチェーン・ピアのタイプ（例えば、承認ノード、編成ノード、ピア・ノードなど）などのような１つまたは複数の属性に基づいて選択されてもよい。５３０において、この方法は、ブロックのチェーンに関連する以前に格納された検証ブロックのうちの１つまたは複数に基づいて、置換データ・ブロックが有効であるかどうかを決定することを含むことができ、置換データ・ブロックが有効であると決定したことに応じて、５４０において、この方法は、分散型台帳上で破損データ・ブロックを置換データ・ブロックに置き換えることを含むことができる。いくつかの実施形態では、以前に格納された検証ブロックは、台帳検証スレッドを実行するブロックチェーン・ノードの台帳のバックアップ・コピーに以前に格納されたブロックとすることができる。別の例として、置換データ・ブロックは、ブロックのチェーン内の次に続くデータ・ブロックから得られるブロックのハッシュ値に基づいて検証することができる。

【００５９】

いくつかの実施形態では、破損ブロックは、元のブロックのサイズを変更し、それにより、台帳のサイズの変更を引き起こすことがある。それゆえに、単に破損ブロックを置き換えると、台帳に欠陥が残ることがあり（すなわち、破損ブロックのサイズが小さい場合）、または隣接するブロックの一部が上書きされることがある（すなわち、破損ブロックのサイズが大きい場合）。それゆえに、いくつかの実施形態では、この方法は、破損データ・ブロックが元のデータ・ブロックのブロック・サイズを変更したと決定された場合、分散型台帳のブロックのチェーンにおける１つまたは複数の次に続くデータ・ブロックについての置換データ・ブロックを取得することをさらに含むことができる。

【００６０】

図８は、例示の実施形態による操作の例示の方法の１つまたは複数に従ってブロックチェーンで様々な操作を実行するように構成された例示の物理インフラストラクチャを示す。図８を参照すると、例示の構成６００は、ブロックチェーン６２０とスマート・コントラクト６４０とを有する物理インフラストラクチャ６１０を含み、物理インフラストラクチャ６１０は、例示の実施形態のいずれかに含まれる操作ステップ６１２のいずれかを実行することができる。ステップ／操作６１２は、１つまたは複数の流れ図または論理図あるいはその両方で説明または示されるステップのうちの１つまたは複数を含むことができる。ステップは、コンピュータ・システム構成の物理インフラストラクチャ６１０に常駐する１つまたは複数のスマート・コントラクト６４０またはブロックチェーン６２０あるいはその両方に書き込まれるかまたはそれから読み出される出力または書き込まれた情報を表すことができる。データは、実行されたスマート・コントラクト６４０またはブロックチェーン６２０あるいはその両方から出力され得る。物理インフラストラクチャ６１０

は、１つまたは複数のコンピュータ、サーバ、プロセッサ、メモリ、または無線通信デバイス、あるいはその組合せを含むことができる。

【００６１】

図９は、例示の実施形態による、ブロックチェーンでスマート・コントラクト条件を実施するように構成された、契約当事者と仲介サーバとの間の例示のスマート・コントラクト構成を示す。図９を参照すると、構成６５０は、１つまたは複数のユーザ・デバイス６５２またはユーザ・デバイス６５６あるいはその両方を明確に識別するスマート・コントラクト６４０によって駆動される通信セッション、アセット転送セッション、またはプロセスもしくは手順を表すことができる。スマート・コントラクト実行の実行、動作、および結果は、サーバ６５４で管理することができる。スマート・コントラクト６４０の内容は、スマート・コントラクト・トランザクションの当事者であるエンティティ６５２および６５６のうちの１つまたは複数によるデジタル署名を必要とすることがある。スマート・コントラクト実行の結果は、ブロックチェーン・トランザクションとしてブロックチェーンに書き込むことができる。

【００６２】

上述の実施形態は、ハードウェアで、プロセッサによって実行されるコンピュータ・プログラムで、ファームウェアで、または上述のものの組合せで実装することができる。コンピュータ・プログラムは、記憶媒体などのコンピュータ可読媒体で具現することができる。例えば、コンピュータ・プログラムは、ランダム・アクセス・メモリ（「ＲＡＭ」）、フラッシュ・メモリ、読出し専用メモリ（「ＲＯＭ」）、消去可能プログラマブル読出し専用メモリ（「ＥＰＲＯＭ」）、電気的消去可能プログラマブル読出し専用メモリ（「ＥＥＰＲＯＭ」）、レジスタ、ハードディスク、リムーバブル・ディスク、コンパクト・ディスク読出し専用メモリ（「ＣＤ－ＲＯＭ」）、または当技術分野で知られている任意の他の形態の記憶媒体に常駐することができる。

【００６３】

例示的な記憶媒体は、プロセッサが記憶媒体から情報を読み出し、記憶媒体に情報を書き込むことができるようにプロセッサに結合され得る。代替では、記憶媒体はプロセッサに一体化されてもよい。プロセッサおよび記憶媒体は、特定用途向け集積回路（「ＡＳＩＣ」）に常駐することができる。代替では、プロセッサおよび記憶媒体は、別々の構成要素として常駐してもよい。例えば、図１０は、上述の構成要素などのいずれかを表すかまたはそれに一体化され得る例示のコンピュータ・システム・アーキテクチャ７００を示す。

【００６４】

図１０は、本明細書に記載された本出願の実施形態の使用または機能の範囲に関していかなる限定も示唆するものではない。それとは関係なく、コンピューティング・ノード７００は、先に記載された機能のいずれかを実装または実行あるいはその両方を行うことができる。

【００６５】

コンピューティング・ノード７００には、多数の他の汎用または専用コンピューティング・システム環境または構成で操作可能なコンピュータ・システム／サーバ７０２がある。コンピュータ・システム／サーバ７０２で使用するのに適する可能性があるよく知られているコンピューティング・システム、環境、または構成、あるいはその組合せの例には、限定はしないが、パーソナル・コンピュータ・システム、サーバ・コンピュータ・システム、シン・クライアント、シック・クライアント、携帯型またはラップトップ・デバイス、マルチプロセッサ・システム、マイクロプロセッサ・ベース・システム、セット・トップ・ボックス、プログラマブル家庭用電化製品、ネットワークＰＣ、ミニコンピュータ・システム、メインフレーム・コンピュータ・システム、および上述のシステムまたはデバイスのいずれかを含む分散型クラウド・コンピューティング環境、などが含まれる。

【００６６】

コンピュータ・システム／サーバ７０２は、コンピュータ・システムによって実行され

るプログラム・モジュールなどのコンピュータ・システム実行可能命令の一般的なコンテキストで説明することができる。一般に、プログラム・モジュールは、特定のタスクを実行するか、または特定の抽象データ型を実装するルーチン、プログラム、オブジェクト、コンポーネント、論理、データ構造などを含むことができる。コンピュータ・システム／サーバ７０２は、通信ネットワークを通してリンクされたリモート処理デバイスによってタスクが実行される分散型クラウド・コンピューティング環境で実践することができる。分散型クラウド・コンピューティング環境では、プログラム・モジュールは、メモリ・ストレージ・デバイスを含むローカルとリモートの両方のコンピュータ・システム記憶媒体に配置することができる。

【００６７】

図１０に示されるように、クラウド・コンピューティング・ノード７００のコンピュータ・システム／サーバ７０２は、汎用コンピューティング・デバイスの形態で示されている。コンピュータ・システム／サーバ７０２の構成要素は、限定はしないが、１つまたは複数のプロセッサまたは処理ユニット７０４と、システム・メモリ７０６と、システム・メモリ７０６を含む様々なシステム構成要素をプロセッサ７０４に結合するバスとを含むことができる。

【００６８】

バスは、メモリ・バスもしくはメモリ・コントローラ、周辺バス、アクセラレーテッド・グラフィック・ポート、および様々なバス・アーキテクチャのいずれかを使用するプロセッサもしくはローカル・バスを含むいくつかのタイプのバス構造のうちのいずれか１つまたは複数を表す。限定ではなく例として、そのようなアーキテクチャは、インダストリ・スタンダード・アーキテクチャ（ＩＳＡ）バス、マイクロ・チャネル・アーキテクチャ（ＭＣＡ）バス、拡張ＩＳＡ（ＥＩＳＡ）バス、ビデオ・エレクトロニクス規格協会（ＶＥＳＡ）ローカル・バス、およびペリフェラル・コンポーネント・インターコネクト（ＰＣＩ）バスを含む。

【００６９】

コンピュータ・システム／サーバ７０２は、一般に、様々なコンピュータ・システム可読媒体を含む。そのような媒体は、コンピュータ・システム／サーバ７０２によってアクセス可能な任意の利用可能な媒体とすることができ、それは、揮発性と不揮発性の両方の媒体、着脱可能と着脱不能の両方の媒体を含む。システム・メモリ７０６は、１つの実施形態では、他の図の流れ図を実装する。システム・メモリ７０６は、ランダム・アクセス・メモリ（ＲＡＭ）７１０またはキャッシュ・メモリ７１２あるいはその両方などの揮発性メモリの形態のコンピュータ・システム可読媒体を含むことができる。コンピュータ・システム／サーバ７０２は、他の着脱可能／着脱不能な揮発性／不揮発性のコンピュータ・システム記憶媒体をさらに含むことができる。単に例として、着脱不能な不揮発性の磁気媒体（図示せず、一般に「ハード・ドライブ」と呼ばれる）からの読出しおよびそれへの書込みのためのストレージ・システム７１４が、備えられてもよい。図示されていないが、着脱可能な不揮発性の磁気ディスク（例えば、「フロッピー（登録商標）ディスク」）からの読出しおよびそれへの書込みのための磁気ディスク・ドライブと、ＣＤ－ＲＯＭ、ＤＶＤ－ＲＯＭ、または他の光学媒体などの着脱可能な不揮発性の光ディスクからの読出しまたはそれへの書込みのための光ディスク・ドライブとが、備えられてもよい。そのような場合に、各々は、１つまたは複数のデータ媒体インタフェースによってバスに接続することができる。以下でさらに図示および説明するように、メモリ７０６は、本出願の様々な実施形態の機能を実行するように構成された１組の（例えば、少なくとも１つの）プログラム・モジュールを有する少なくとも１つのプログラム製品を含むことができる。

【００７０】

１組の（少なくとも１つの）プログラム・モジュール７１８を有するプログラム／ユーティリティ７１６は、限定ではなく例としてメモリ７０６に、ならびにオペレーティング・システム、１つまたは複数のアプリケーション・プログラム、他のプログラム・モジュール、およびプログラム・データに格納することができる。オペレーティング・システム

、１つまたは複数のアプリケーション・プログラム、他のプログラム・モジュール、およびプログラム・データの各々、またはそれらのいくつかの組合せは、ネットワーク環境の実装を含むことができる。プログラム・モジュール７１８は、一般に、本明細書に記載されるような本出願の様々な実施形態の機能または方法あるいはその両方を実行する。例えば、１つまたは複数のプログラム・モジュール７１８は、破損データを検出した際に分散型台帳を検証および自己修正するための、本明細書に記載された台帳検証スレッドを実施することができる。

【００７１】

当業者によって理解されるように、本出願の態様は、システム、方法、またはコンピュータ・プログラム製品として具現され得る。その結果、本出願の態様は、完全にハードウェアの実施形態、完全にソフトウェアの実施形態（ファームウェア、常駐ソフトウェア、マイクロコードなどを含む）、またはすべて一般に本明細書において「回路」、「モジュール」、もしくは「システム」と呼ばれることがあるソフトウェアの態様とハードウェアの態様を組み合わせた実施形態の形態をとることができる。さらに、本出願の態様は、コンピュータ可読プログラム・コードが具現された１つまたは複数のコンピュータ可読媒体で具現されたコンピュータ・プログラム製品の形態をとることができる。

【００７２】

コンピュータ・システム／サーバ７０２は、キーボード、ポインティング・デバイス、ディスプレイ７２２などの１つまたは複数の外部デバイス７２０、ユーザがコンピュータ・システム／サーバ７０２と対話することを可能にする１つまたは複数のデバイス、またはコンピュータ・システム／サーバ７０２が１つまたは複数の他のコンピューティング・デバイスと通信することを可能にする任意のデバイス（例えば、ネットワーク・カード、モデムなど）、あるいはその組合せと通信することもできる。そのような通信は、Ｉ／Ｏインタフェース７２４を介して行うことができる。さらにまた、コンピュータ・システム／サーバ７０２は、ネットワーク・アダプタ７２６を介して、ローカル・エリア・ネットワーク（ＬＡＮ）、一般的なワイド・エリア・ネットワーク（ＷＡＮ）、またはパブリック・ネットワーク（例えばインターネット）、あるいはその組合せなどの１つまたは複数のネットワークと通信することができる。図示のように、ネットワーク・アダプタ７２６は、バスを介して、コンピュータ・システム／サーバ７０２の他の構成要素と通信する。図示されていないが、他のハードウェア構成要素またはソフトウェア構成要素あるいはその両方が、コンピュータ・システム／サーバ７０２とともに使用されてもよいことが理解されるべきである。例には、限定はしないが、マイクロコード、デバイス・ドライバ、冗長処理ユニット、外部ディスク・ドライブ・アレイ、ＲＡＩＤシステム、テープ・ドライブ、およびデータ・アーカイブ・ストレージ・システムなどが含まれる。

【００７３】

様々な実施形態によれば、メモリ７０６は、ブロックチェーン・ピアなどのブロックチェーン・ネットワークの複数のノードにわたって格納および複製される分散型台帳のローカル・コピーを格納することができる。一方、プロセッサ７０４は、台帳検証スレッドを介して、分散型台帳のブロックのチェーン内に格納されている破損データ・ブロックを識別し、分散型台帳へのアクセスを有する複数のブロックチェーン・ピアの中からブロックチェーン・ピアを選択し、選択されたピアから置換データ・ブロックを取得し、ブロックのチェーンに関連する以前に格納された検証ブロックのうちの１つまたは複数に基づいて、置換データ・ブロックが有効であるかどうかを決定し、置換データ・ブロックが有効であるという決定に応じて、分散型台帳上で破損データ・ブロックを置換データ・ブロックに置き換えるように構成することができる。例えば、プロセッサ７０４は、プロセッサ、オペレーティング・システム、ハードウェア（ネットワーク・インタフェース７２６など）を制御する台帳検証スレッドを実行して、破損データ検出および置換プロセスを実行することができる。

【００７４】

システム、方法、および非一過性コンピュータ可読媒体のうちの少なくとも１つの例示

的な実施形態が、添付の図面に示され、前述の詳細な説明に記載されたが、本出願は、開示された実施形態に限定されるのではなく、以下の特許請求の範囲に記載および定義されるように、多数の再配置、変更、および置換が可能であることが理解されよう。例えば、様々な図のシステムの機能は、本明細書に記載されたモジュールまたは構成要素の１つまたは複数によって、あるいは分散型アーキテクチャで実行することができ、送信機、受信機、または両方のペアを含むことができる。例えば、個々のモジュールによって実行される機能のすべてまたは一部は、これらのモジュールの１つまたは複数によって実行されてもよい。さらに、本明細書に記載された機能は、様々な時点に、様々なイベントに関連して、モジュールまたは構成要素の内部または外部で実行されてもよい。さらに、様々なモジュール間で送られる情報は、データ・ネットワーク、インターネット、音声ネットワーク、インターネット・プロトコル・ネットワーク、無線デバイス、有線デバイスのうちの少なくとも１つを介して、または複数のプロトコルを介して、あるいはその両方を介して、モジュール間で送られ得る。さらに、モジュールのいずれかによって送られまたは受け取られるメッセージは、直接、または他のモジュールの１つまたは複数を介して、あるいはその両方で、送られるかまたは受け取られてもよい。

【００７５】

当業者は、「システム」が、パーソナル・コンピュータ、サーバ、コンソール、携帯情報端末（ＰＤＡ）、携帯電話、タブレット・コンピューティング・デバイス、スマートフォン、または任意の他の好適なコンピューティング・デバイス、あるいはデバイスの組合せとして具現され得ることを理解するであろう。上述の機能を「システム」によって実行されるとして提示することは、決して本出願の範囲を限定するものではなく、多くの実施形態のうちの１つの例を提供するものである。実際、本明細書で開示される方法、システム、および装置は、コンピューティング技術と両立する局所形態および分散形態で実施することができる。

【００７６】

本明細書に記載されたシステムの機能のいくつかは、それらの実施態様の独立性をさらに特に強調するために、モジュールとして提示されていることに留意されたい。例えば、モジュールは、カスタム超大規模集積（ＶＬＳＩ）回路もしくはゲート・アレイ、論理チップなどの既製の半導体を用いた素子、トランジスタ、または他の個別の構成要素を含むハードウェア回路として実装することができる。モジュールはまた、フィールド・プログラマブル・ゲート・アレイ、プログラマブル・アレイ・ロジック、プログラマブル・ロジック・デバイス、グラフィック処理ユニットなどのようなプログラマブル・ハードウェア・デバイスで実装することができる。

【００７７】

モジュールはまた、様々なタイプのプロセッサによる実行のために少なくとも部分的にソフトウェアで実装することができる。例えば、実行可能コードの識別されたユニットは、例えば、オブジェクト、プロシージャ、または機能として編成することができるコンピュータ命令の１つまたは複数の物理的または論理的ブロックを含むことができる。それにもかかわらず、識別されたモジュールの実行ファイルは、物理的に一緒に配置する必要はなく、異なる場所に格納された異種の命令を含むことができ、異種の命令は、論理的に一緒に結合されたとき、モジュールを構成し、モジュールの規定された目的を達成する。さらに、モジュールは、例えば、ハードディスク・ドライブ、フラッシュ・デバイス、ランダム・アクセス・メモリ（ＲＡＭ）、テープ、またはデータを格納するために使用される任意の他のそのような媒体とすることができるコンピュータ可読媒体に格納されてもよい。

【００７８】

実際、実行可能コードのモジュールは、単一の命令または多くの命令とすることができ、さらに、いくつかの異なるコード・セグメントにわたって、異なるプログラムの間で、およびいくつかのメモリ・デバイスにわたって分散されてもよい。同様に、操作データは、本明細書では、モジュール内で識別され示されてもよく、任意の適切な形態で具現され

任意の適切なタイプのデータ構造内で編成されてもよい。操作データは、単一のデータセットとして収集されてもよく、または異なるストレージ・デバイスにわたることを含む異なる場所にわたって分散されてもよく、少なくとも部分的に、単にシステムまたはネットワークに電子信号として存在してもよい。

【００７９】

本明細書の図に概して記載および図示されている本出願の構成要素は、多種多様な異なる構成で配列および設計することができることが容易に理解されよう。したがって、実施形態の詳細な説明は、特許請求される本出願の範囲を限定するものではなく、単に本出願の選択された実施形態を表している。

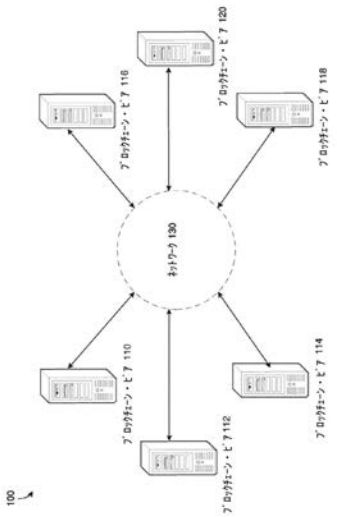
【００８０】

当業者は、上述が、異なる順序のステップ、または開示されているものと異なる構成のハードウェア要素、あるいはその両方で実践できることを容易に理解されよう。それゆえに、本出願が、これらの好ましい実施形態に基づいて記載されているが、特定の変更、変形、および代替の構造が明白であることが当業者には明らかであろう。

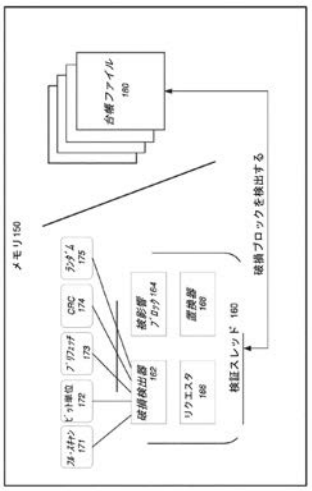
【００８１】

本出願の好ましい実施形態が記載されているが、記載された実施形態は単に例示であり、本出願の範囲は、その等価物および変更の全範囲（例えば、プロトコル、ハードウェア・デバイス、ソフトウェア・プラットフォームなど）で考えられるとき、添付の特許請求の範囲によってのみ定義されるべきであることが理解されるべきである。

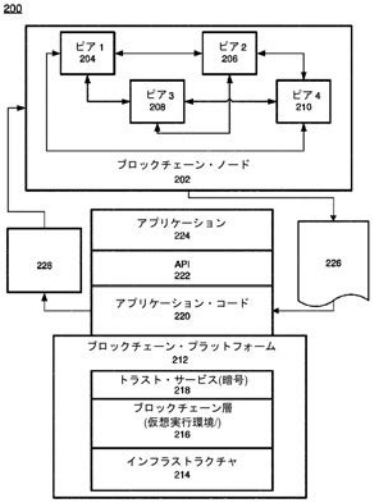
【図１】



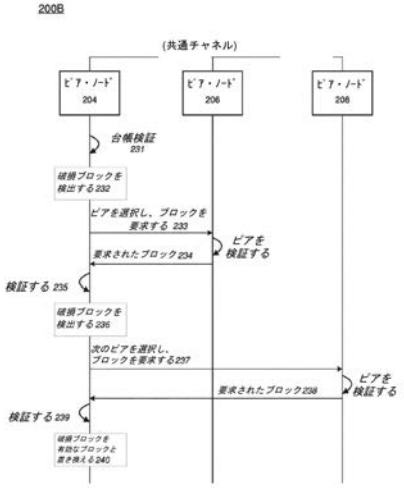
【図２】



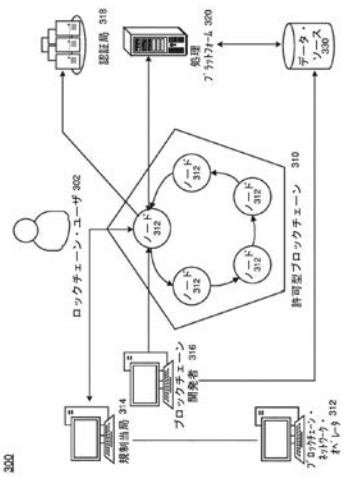
【図３】


200
ピア 1
204
ピア 2
206
ピア 3
208
ピア 4
210
ブロックチェーン・ノード
202
アプリケーション
224
API
222
アプリケーション・コード
220
228
226
ブロックチェーン・プラットフォーム
212
トラスト・サービス(暗号)
218
ブロックチェーン層
(仮想実行環境)
216
インフラストラクチャ
214

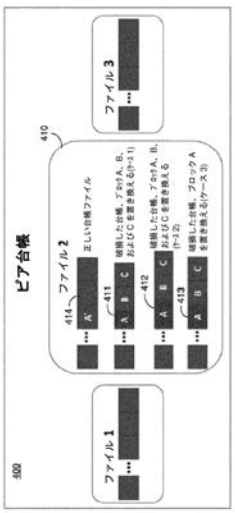
【図４】


200B
(共通チャネル)
ピア・ノード
204
ピア・ノード
206
ピア・ノード
208
台帳検証
231
破損ブロックを
検出する232
ピアを選択し、ブロックを
要求する 233
ピアを
検証する
要求されたブロック234
検証する 235
破損ブロックを
検出する 236
次のピアを選択し、
ブロックを要求する237
ピアを
検証する
要求されたブロック238
検証する 239
破損ブロックを
有効なブロックと
置き換える240

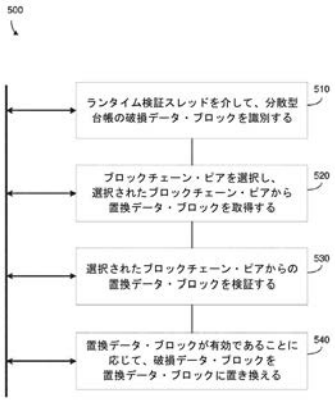
【図５】


300
認証局 318
処理
プラットフォーム 320
データ・
ソース
330
ノード
312
許可型ブロックチェーン 310
ロックチェーン・ユーザ 302
ブロックチェーン
開発者 316
規制当局 314
ブロックチェーン・
ネットワーク・
オペレータ 312

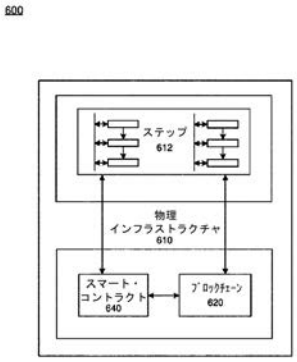
【図６】


ピア台帳
400
410
ファイル 3
ファイル 2
正しい台帳ファイル
414
411
412
413
破損した台帳、ブロックA、B、
およびCを置き換える(ケース1)
破損した台帳、ブロックA、B、
およびCを置き換える
(ケース2)
破損した台帳、ブロックA
を置き換える(ケース3)
ファイル 1

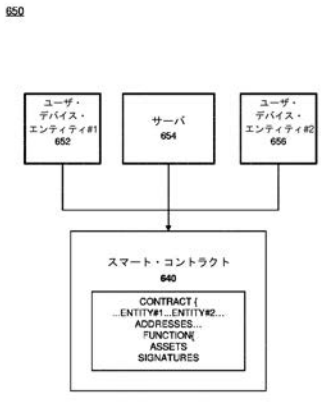
【図７】


500
510
ランタイム検証スレッドを介して、分散型台帳の破損データ・ブロックを識別する
520
ブロックチェーン・ピアを選択し、選択されたブロックチェーン・ピアから置換データ・ブロックを取得する
530
選択されたブロックチェーン・ピアからの置換データ・ブロックを検証する
540
置換データ・ブロックが有効であることに応じて、破損データ・ブロックを置換データ・ブロックに置き換える

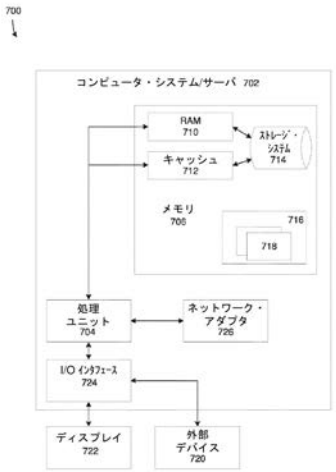
【図８】


600
ステップ
612
物理
インフラストラクチャ
610
スマート・
コントラクト
640
ブロックチェーン
620

【図９】


650
ユーザ・
デバイス・
エンティティ#1
652
サーバ
654
ユーザ・
デバイス・
エンティティ#2
656
スマート・コントラクト
640
CONTRACT {
...ENTITY#1...ENTITY#2...
ADDRESSES...
FUNCTIONS{
ASSETS
SIGNATURES

【図１０】


700
コンピュータ・システム/サーバ 702
RAM
710
キャッシュ
712
ストレージ・
システム
714
メモリ
706
716
718
処理
ユニット
704
ネットワーク・
アダプタ
726
I/O インタフェース
724
ディスプレイ
722
外部
デバイス
720

【手続補正書】
【提出日】令和2年9月30日(2020.9.30)
【手続補正１】
【補正対象書類名】特許請求の範囲
【補正対象項目名】全文
【補正方法】変更
【補正の内容】
【特許請求の範囲】
【請求項１】
破損データ・ブロックを管理するためのコンピューティング・システムであって、前記システムが、
分散型台帳を格納するメモリと、
台帳検証スレッドを介して、前記分散型台帳のブロックのチェーン内に格納されている前記破損データ・ブロックを識別し、前記分散型台帳へのアクセスを有する複数のブロックチェーン・ピアの中からブロックチェーン・ピアを選択し、前記選択されたブロックチェーン・ピアから置換データ・ブロックを取得し、前記ブロックのチェーンに関連する以前に格納された検証ブロックのうちの１つまたは複数に基づいて、前記置換データ・ブロックが有効であるかどうかを決定し、前記置換データ・ブロックが有効であるという前記決定に応じて、前記分散型台帳上で前記破損データ・ブロックを前記置換データ・ブロックに置き換えるように構成されたプロセッサと
を含む、コンピューティング・システム。
【請求項２】
前記台帳検証スレッドは、前記メモリで実行され、前記分散型台帳を構成するファイルとは別に格納されているコンパイル済みコードを有するプログラムを含む、請求項１に記載のコンピューティング・システム。
【請求項３】
前記プロセッサは、前記破損データ・ブロックが元のデータ・ブロックのブロック・サイズを変更したと決定された場合、前記ブロックのチェーンにおける次に続くデータ・ブロックについての置換データ・ブロックを取得するようにさらに構成される、請求項１または２のいずれかに記載のコンピューティング・システム。
【請求項４】
前記以前に格納された検証ブロックが、前記台帳検証スレッドを実行するブロックチェーン・ノードの前記分散型台帳のバックアップ・コピーに格納された正しいブロックを含む、請求項１ないし３のいずれかに記載のコンピューティング・システム。
【請求項５】
前記プロセッサが、前記分散型台帳のデータ・ブロックを順次スキャンし、ランタイムに生成された、前記順次スキャンされたデータ・ブロックのそれぞれのハッシュ値に基づいて、前記順次スキャンされたデータ・ブロックを検証して、前記破損データ・ブロックを識別するように構成される、請求項１ないし４のいずれかに記載のコンピューティング・システム。
【請求項６】
前記プロセッサが、前記分散型台帳のバックアップ・コピーとの前記分散型台帳のビット単位の比較を実行して、前記ビット単位の比較に基づいて前記破損データ・ブロックを識別するように構成される、請求項１ないし５のいずれかに記載のコンピューティング・システム。
【請求項７】
前記プロセッサが、前記分散型台帳から現在選択されているデータ・ブロックに関連するブロック・アクセス・パターンに基づいて前記分散型台帳からデータ・ブロックをプリフェッチし、前記プリフェッチされたデータ・ブロックのそれぞれのハッシュに基づいて前記プリフェッチされたデータ・ブロックを検証して、前記プリフェッチされたデータ・

ブロックから前記破損データ・ブロックを識別するように構成される、請求項１ないし６のいずれかに記載のコンピューティング・システム。
【請求項８】
前記プロセッサが、前記分散型台帳のデータ・ブロックを前記それぞれのデータ・ブロックの以前に生成されたチェックサム値に基づいて順次検証して、前記チェックサム値に基づいて前記破損データ・ブロックを識別するように構成される、請求項１ないし７のいずれかに記載のコンピューティング・システム。
【請求項９】
前記プロセッサが、前記分散型台帳からデータ・ブロックをランダムに選択し、前記ランダムに選択されたデータ・ブロックのそれぞれのハッシュに基づいて前記ランダムに選択されたデータ・ブロックを検証して、前記ランダムに選択されたデータ・ブロックから前記破損データ・ブロックを識別するように構成される、請求項１ないし８のいずれかに記載のコンピューティング・システム。
【請求項１０】
破損データ・ブロックを管理するための方法であって、前記方法が、コンピュータが、
台帳検証スレッドを介して、分散型台帳のブロックのチェーン内に格納されている前記破損データ・ブロックを識別するステップと、
前記分散型台帳へのアクセスを有する複数のブロックチェーン・ピアの中からブロックチェーン・ピアを選択し、前記選択されたブロックチェーン・ピアから置換データ・ブロックを取得するステップと、
前記ブロックのチェーンに関連する以前に格納された検証ブロックのうちの１つまたは複数に基づいて、前記置換データ・ブロックが有効であるかどうかを決定するステップと、
前記置換データ・ブロックが有効であると決定したことに応じて、前記分散型台帳上で前記破損データ・ブロックを前記置換データ・ブロックに置き換えるステップと
を実行する、方法。
【請求項１１】
前記台帳検証スレッドが、前記分散型台帳を構成するファイルとは別に格納されているコンパイル済みコードを有するメモリで実行されるプログラムを含む、請求項１０に記載の方法。
【請求項１２】
前記取得するステップは、前記破損データ・ブロックが元のデータ・ブロックのブロック・サイズを変更したと決定された場合、前記ブロックのチェーンにおける次に続くデータ・ブロックについての置換データ・ブロックを取得するステップをさらに含む、請求項１０または１１のいずれかに記載の方法。
【請求項１３】
前記以前に格納された検証ブロックが、前記台帳検証スレッドを実行するブロックチェーン・ノードの前記分散型台帳のバックアップ・コピーに格納された正しいブロックを含む、請求項１０ないし１２のいずれかに記載の方法。
【請求項１４】
前記台帳検証スレッドを介して前記破損データ・ブロックを前記識別するステップが、前記分散型台帳のデータ・ブロックを順次スキャンし、ランタイムに生成された、前記順次スキャンされたデータ・ブロックのそれぞれのハッシュ値に基づいて、前記順次スキャンされたデータ・ブロックを検証して、前記破損データ・ブロックを識別するステップを含む、請求項１０ないし１３のいずれかに記載の方法。
【請求項１５】
前記台帳検証スレッドを介して前記破損データ・ブロックを前記識別するステップが、前記分散型台帳のバックアップ・コピーとの前記分散型台帳のビット単位の比較を実行して、前記ビット単位の比較に基づいて前記破損データ・ブロックを識別するステップを含む、請求項１０ないし１４のいずれかに記載の方法。

【請求項１６】
前記台帳検証スレッドを介して前記破損データ・ブロックを前記識別するステップが、前記分散型台帳から現在選択されているデータ・ブロックに関連するブロック・アクセス・パターンに基づいて前記分散型台帳からデータ・ブロックをプリフェッチし、前記プリフェッチされたデータ・ブロックのそれぞれのハッシュに基づいて前記プリフェッチされたデータ・ブロックを検証して、前記プリフェッチされたデータ・ブロックから前記破損データ・ブロックを識別するステップを含む、請求項１０ないし１５のいずれかに記載の方法。
【請求項１７】
前記台帳検証スレッドを介して前記破損データ・ブロックを前記識別するステップが、前記分散型台帳のデータ・ブロックを前記それぞれのデータ・ブロックの以前に生成されたチェックサム値に基づいて順次検証して、前記チェックサム値に基づいて前記破損データ・ブロックを識別するステップを含む、請求項１０ないし１６のいずれかに記載の方法。
【請求項１８】
前記台帳検証スレッドを介して前記破損データ・ブロックを前記識別するステップが、前記分散型台帳からデータ・ブロックをランダムに選択し、前記ランダムに選択されたデータ・ブロックのそれぞれのハッシュに基づいて前記ランダムに選択されたデータ・ブロックを検証して、前記ランダムに選択されたデータ・ブロックから前記破損データ・ブロックを識別するステップを含む、請求項１０ないし１７のいずれかに記載の方法。
【請求項１９】
破損データ・ブロックを管理するためのコンピュータ・プログラムを格納した記録媒体であって、前記コンピュータ・プログラムが、
コンピュータによって可読であり、請求項１０ないし１８のいずれかに記載の方法の各ステップを実行するために前記コンピュータによって実行するための命令をを含む、記録媒体。
【請求項２０】
コンピュータに、請求項１０ないし１８のいずれかに記載の方法の各ステップを実行させるためのコンピュータ・プログラム。

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2019/055894

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F11/08 G06F11/14 G06Q20/22 G06Q20/38
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)
G06F G06Q

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	US 2018/082296 A1 (BRASHERS PER W [US]) 22 March 2018 (2018-03-22) page 2, paragraph 27 - page 8, paragraph 124 figures 1-11 -----	1-20
X	RAVI KIRAN RAMAN ET AL: "Distributed Storage Meets Secret Sharing on the Blockchain", 2018 INFORMATION THEORY AND APPLICATIONS WORKSHOP (ITA), 1 February 2018 (2018-02-01), pages 1-6, XP055566040, DOI: 10.1109/ITA.2018.8503089 ISBN: 978-1-7281-0124-8 the whole document ----- -/--	1-20

[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
15 May 2019	22/05/2019

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	Rachkov, Vassil

1

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

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2019/055894

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 2017/295023 A1 (MADHAVAN AJAY [US] ET AL) 12 October 2017 (2017-10-12) page 4, paragraph 32 - page 20, paragraph 179 figures	1-20
A	US 2017/228371 A1 (SEGER II ROBERT ALLAN [US]) 10 August 2017 (2017-08-10) page 2, paragraph 19 - page 6, paragraph 59 figures	1-20
A	ELLI ANDROULAKI ET AL: "Hyperledger fabric : a distributed operating system for permissioned blockchains", PROCEEDINGS OF THE THIRTEENTH EUROSYS CONFERENCE ON , EUROSYS '18, 1 January 2018 (2018-01-01), pages 1-15, XP055554083, New York, New York, USA DOI: 10.1145/3190508.3190538 ISBN: 978-1-4503-5584-1 the whole document	1-20

1

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

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2019/055894

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2018082296 A1	22-03-2018	US 2018082296 A1	22-03-2018
		WO 2018057719 A1	29-03-2018
US 2017295023 A1	12-10-2017	NONE	
US 2017228371 A1	10-08-2017	US 2017228371 A1	10-08-2017
		WO 2017136527 A1	10-08-2017

フロントページの続き

(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,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,CZ,DE,DJ,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IR,IS,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,SM,ST,SV,SY,TH,TJ,TM,TN,TR,TT

(74)代理人　100112690
弁理士　太佐　種一
(74)復代理人　110000420
特許業務法人エム・アイ・ピー
(72)発明者　バセト、サルマン、アブドゥル
アメリカ合衆国１０５９８　ニューヨーク州ヨークタウン・ハイツ　ピーオーボックス２１８　ルート１３４　キッチャワン・ロード１１０１
(72)発明者　ディレンバーガー、ドンナ、エヌ、エング
アメリカ合衆国１０５９８　ニューヨーク州ヨークタウン・ハイツ　キッチャワン・ロード１１０１　ピーオーボックス２１８
(72)発明者　ノヴォトニ、ペトル
アメリカ合衆国１０５９８　ニューヨーク州ヨークタウン・ハイツ　キッチャワン・ロード１１０１　ピーオーボックス２１８
(72)発明者　チャン、チイ
アメリカ合衆国１０５９８　ニューヨーク州ヨークタウン・ハイツ　キッチャワン・ロード１１０１　ピーオーボックス２１８

Ｆターム(参考) 5B034 CC02