(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6931999号
(P6931999)
(45)発行日　令和3年9月8日(2021.9.8)　(24)登録日　令和3年8月19日(2021.8.19)

(51)Int.Cl.　F I
G06Q 50/10 (2012.01)　G06Q 50/10

請求項の数 14　(全 25 頁)

(21)出願番号　特願2017-19531(P2017-19531)
(22)出願日　平成29年2月6日(2017.2.6)
(65)公開番号　特開2018-128723(P2018-128723A)
(43)公開日　平成30年8月16日(2018.8.16)
審査請求日　令和1年12月27日(2019.12.27)

(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　110000176
一色国際特許業務法人
(72)発明者　佐藤　竜也
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　山崎　誠也

最終頁に続く

(54)【発明の名称】信用度管理システムおよび信用度管理方法

(57)【特許請求の範囲】
【請求項１】
分散台帳をそれぞれ保持する複数の検証ノードと前記検証ノードへトランザクションを発行する複数のトランザクション発行ノードから構成されるシステムであって、
前記検証ノードそれぞれが、それぞれで管理するブロックチェーンにおいて、スマートコントラクトおよび前記スマートコントラクトの本番の実行トランザクションの他に、前記スマートコントラクトに対する評価実行トランザクションを管理し、
前記検証ノードそれぞれが、所定の前記トランザクション発行ノードが指定した評価用の所定値と、前記トランザクション発行ノードから受け取ったトランザクションが評価実行トランザクションであった際に、前記所定値を前記スマートコントラクトに入力して得た出力値とを含む、前記スマートコントラクトの検証結果を、前記ブロックチェーンにおいて前記評価実行トランザクションとあわせて管理する、または前記出力値に基づき前記スマートコントラクトに関するステート情報を更新するものである、
ことを特徴とする信用度管理システム。
【請求項２】
前記複数の検証ノードのそれぞれは、
ブロックチェーンに含まれる所定の評価実行トランザクションを基準として、前記スマートコントラクトの実行トランザクションおよびその他の評価実行トランザクションの少なくともいずれかに関する信用度を所定アルゴリズムで算定する処理を更に実行するものである、

ことを特徴とする請求項１に記載の信用度管理システム。
【請求項３】
前記複数の検証ノードのそれぞれは、
前記算定した信用度の高さに応じて、前記スマートコントラクトの実行可否を制御する処理を更に実行するものである、
ことを特徴とする請求項２に記載の信用度管理システム。
【請求項４】
前記複数の検証ノードのそれぞれは、
前記スマートコントラクトの提供者および利用者の少なくともいずれかによる、ブロックチェーンへの前記評価実行トランザクションの登録有無に応じて、前記スマートコントラクトの実行可否を制御する処理を更に実行するものである、
ことを特徴とする請求項１に記載の信用度管理システム。
【請求項５】
前記複数の検証ノードのそれぞれは、
分散台帳において、実行トランザクションおよび評価実行トランザクションのそれぞれに関する所定情報を、異なるデータ領域で管理するものである、
ことを特徴とする請求項１に記載の信用度管理システム。
【請求項６】
前記複数の検証ノードのそれぞれは、
前記評価実行トランザクションを所定タイミングで自動発行して、前記スマートコントラクトの検証を実行するものである、
ことを特徴とする請求項１に記載の信用度管理システム。
【請求項７】
前記複数の検証ノードのそれぞれは、
所定期間内の前記実行トランザクションと前記評価実行トランザクションのうち、前記実行トランザクションを優先的に処理するものである、ことを特徴とする請求項１に記載の信用度管理システム。
【請求項８】
分散台帳をそれぞれ保持する複数の検証ノードと前記検証ノードへトランザクションを発行する複数のトランザクション発行ノードから構成されるシステムにおいて、
前記検証ノードそれぞれが、それぞれで管理するブロックチェーンにおいて、スマートコントラクトおよび前記スマートコントラクトの本番の実行トランザクションの他に、前記スマートコントラクトに対する評価実行トランザクションを管理し、
前記検証ノードそれぞれが、所定の前記トランザクション発行ノードが指定した評価用の所定値と、前記トランザクション発行ノードから受け取ったトランザクションが評価実行トランザクションであった際に、前記所定値を前記スマートコントラクトに入力して得た出力値とを含む、前記スマートコントラクトの検証結果を、前記ブロックチェーンにおいて前記評価実行トランザクションとあわせて管理する、または前記出力値に基づき前記スマートコントラクトに関するステート情報を更新する、
ことを特徴とする信用度管理方法。
【請求項９】
前記複数の検証ノードのそれぞれが、
ブロックチェーンに含まれる所定の評価実行トランザクションを基準として、前記スマートコントラクトの実行トランザクションおよびその他の評価実行トランザクションの少なくともいずれかに関する信用度を所定アルゴリズムで算定する処理を更に実行する、
ことを特徴とする請求項８に記載の信用度管理方法。
【請求項１０】
前記複数の検証ノードのそれぞれが、
前記算定した信用度の高さに応じて、前記スマートコントラクトの実行可否を制御する処理を更に実行する、

ことを特徴とする請求項９に記載の信用度管理方法。
【請求項１１】
前記複数の検証ノードのそれぞれが、
前記スマートコントラクトの提供者および利用者の少なくともいずれかによる、ブロックチェーンへの前記評価実行トランザクションの登録有無に応じて、前記スマートコントラクトの実行可否を制御する処理を更に実行する、
ことを特徴とする請求項８に記載の信用度管理方法。
【請求項１２】
前記複数の検証ノードのそれぞれが、
分散台帳において、実行トランザクションおよび評価実行トランザクションのそれぞれに関する所定情報を、異なるデータ領域で管理する、
ことを特徴とする請求項８に記載の信用度管理方法。
【請求項１３】
前記複数の検証ノードのそれぞれが、
前記評価実行トランザクションを所定タイミングで自動発行して、前記スマートコントラクトの検証を実行する、
ことを特徴とする請求項８に記載の信用度管理方法。
【請求項１４】
前記複数の検証ノードのそれぞれが、
所定期間内の前記実行トランザクションと前記評価実行トランザクションのうち、前記実行トランザクションを優先的に処理する、
ことを特徴とする請求項８に記載の信用度管理方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、信用度管理システムおよび信用度管理方法に関するものである。
【背景技術】
【０００２】
従来、金融機関や政府などの信頼できる中央集権機関を経由して実施されてきた取引を、利用者間のＰ２Ｐ（Ｐｅｅｒ　ｔｏ　Ｐｅｅｒ）によって直接的な取引に代替する技術として、分散台帳技術が登場している。
【０００３】
こうした分散台帳技術の例としては、ビットコインと呼ばれる仮想通貨を用いて、銀行などの中央集権機関を必要とせずに決済取引を行う技術（非特許文献１）が存在する。このビットコインによる決済取引では、Ｐ２Ｐネットワーク上において、取引データ（以下、トランザクションとも称する）に関する正当性の判定を、マイナーと呼ばれるノードが実行し、プルーフオブワークと呼ばれる特定のハッシュ値を算出する作業で確定処理を行っている。こうして確定されたトランザクションは、１つのブロックにまとめられ、ブロックチェーン（以下、ＢＣとも称する）と呼ばれる分散台帳に記載される。
【０００４】
また最近では、上記のビットコインで実装されたＢＣをベースにして、ＢＣおよび分散台帳に関する様々な派生技術が提案され、進化を続けている。現状のＢＣの主な特徴としては、（１）ＢＣネットワーク上の参加者間の取引において、中央集権機関ではなく（任意ないしは特定の）参加者による合意形成や承認によって取引を確定させること、（２）複数のトランザクションをブロックとしてまとめ、数珠つなぎに分散台帳に記録し、連続するブロックにハッシュ計算を施すことにより、改ざんを実質不可能にすること、（３）参加者全員が同一の台帳データを共有することにより、参加者全員での取引の確認を可能とすることが挙げられる。
【０００５】
以上の特徴から、ＢＣは、信頼できるデータの管理／共有や、契約に基づく取引の執行

／管理を行う仕組みとして、金融分野やＩｏＴ（Ｉｎｔｅｒｎｅｔ　ｏｆ　Ｔｈｉｎｇ）等、幅広い分野での応用が検討されている。ＢＣを提供する基盤（以下、ＢＣ基盤）を用いることで、中央集権機関による管理がなくとも複数の主体間で情報共有や取引を行うことができる（例えば、特定業界のコンソーシアムやサプライチェーンに関係する複数企業等）。

【０００６】

またＢＣは、ビットコインのような単純な仮想通貨の取引だけでなく、複雑な取引条件や多様なアプリケーションにも適用可能とする仕組みが生まれており、ＢＣの中で（取引）データだけでなくロジックも管理できるようになってきている。このロジックはスマートコントラクト（以下、ＳＣとも称する）と呼ばれる。

【０００７】

上述のＳＣの実行機能を有するＢＣ基盤（非特許文献２、非特許文献３）では、ＳＣそのものとＳＣに対する入力データを管理する。わかりやすく説明すると、ＳＣそのものは（複数の）関数のようなものである。そして入力データは呼び出すＳＣおよび関数名、関数に与える引数のようなものである。ＳＣの実行機能を用いれば、予め定義したコントラクトに従って取引を実行可能となる。

【０００８】

ここでＳＣとその入力データは、ＢＣの中で署名を付けて数珠つなぎにして管理される。そのため、ＳＣ実行機能を有するＢＣ基盤を有することで、データとロジックの登録者が明確となり、さらに登録内容に変更がないことを常に確認可能である。

【０００９】

しかしながら、ＳＣの利用者には、ＳＣの提供者の正しさや一度登録したＳＣに変更がないことまでは信用できても、提供されるＳＣ自体の品質の善しあしまではわからないという問題がある。したがって、利用者（提供者以外の第三者）によってＳＣの品質を評価／信用するための手段が必要である。

【００１０】

ＳＣはバイナリ化された状態あるいは暗号化された状態で分散台帳上に保持されることがあり、利用者にとって、ブラックボックスの場合があるため、ＳＣの品質を評価／信用する手段は非常に重要となる。

【００１１】

この問題に対して、ＳＣ提供者が開発環境など（ＢＣの外側）で事前にＳＣのテストや検証を実施する方法（非特許文献４）が提案されている。また、特定の承認機関がプログラムを審査する技術（非特許文献５）も提案されており、これをＳＣの審査に適用することも可能である。

【先行技術文献】

【非特許文献】

【００１２】

【非特許文献１】”Ａ　Ｐｅｅｒ－ｔｏ－Ｐｅｅｒ　Ｅｌｅｃｔｒｏｎｉｃ　Ｃａｓｈ　Ｓｙｓｔｅｍ”、［ｏｎｌｉｎｅ］、［平成２８年９月１日検索］、インターネット＜ＵＲＬ：ｈｔｔｐｓ：／／ｂｉｔｃｏｉｎ．ｏｒｇ／ｂｉｔｃｏｉｎ．ｐｄｆ＞

【非特許文献２】”Ｅｔｈｅｒｅｕｍ　Ｗｈｉｔｅ　Ｐａｐｅｒ”、［ｏｎｌｉｎｅ］、［平成２８年９月１日検索］、インターネット＜ＵＲＬ：ｈｔｔｐｓ：／／ｇｉｔｈｕｂ．ｃｏｍ／ｅｔｈｅｒｅｕｍ／ｗｉｋｉ／ｗｉｋｉ／［Ｅｎｇｌｉｓｈ］－Ｗｈｉｔｅ－Ｐａｐｅｒ＞

【非特許文献３】”Ｈｙｐｅｒｌｅｄｇｅｒ　Ｆａｂｒｉｃ”、［ｏｎｌｉｎｅ］、［平成２８年９月１日検索］、インターネット＜ＵＲＬ：ｈｔｔｐ：／／ｈｙｐｅｒｌｅｄｇｅｒ－ｆａｂｒｉｃ．ｒｅａｄｔｈｅｄｏｃｓ．ｉｏ／ｅｎ／ｌａｔｅｓｔ／＞

【非特許文献４】”ｃｈａｉｎｔｏｏｌ”、［ｏｎｌｉｎｅ］、［平成２８年９月１日検索］、インターネット＜ＵＲＬ：ｈｔｔｐｓ：／／ｇｉｔｈｕｂ．ｃｏｍ／ｈｙｐｅｒｌｅｄｇｅｒ／ｆａｂｒｉｃ－ｃｈａｉｎｔｏｏｌ＞

【非特許文献５】”コードサイニング証明書”、［ｏｎｌｉｎｅ］、［平成２８年９月１日検索］、インターネット＜ＵＲＬ： ｈｔｔｐｓ：／／ｊｐ．ｇｌｏｂａｌｓｉｇｎ．ｃｏｍ／ｓｅｒｖｉｃｅ／ｃｏｄｅｓｉｇｎ／＞

【発明の概要】

【発明が解決しようとする課題】

【００１３】

上述したように、ＳＣ提供者や特定の承認機関、すなわち中央集権機関によってＳＣ自体の品質の信用性を評価する手段は存在する。しかしながら、これらの手段では中央集権機関による管理が必要となり、ＢＣの持つ非中央集権という特性が損なわれてしまう。つまり、中央集権機関による管理が無ければ、ＳＣ自体の品質の信用性を確認／評価できないという課題がある。

【００１４】

そこで本発明の目的は、中央集権機関による管理が無くとも、スマートコントラクト提供者以外の複数の第三者によってスマートコントラクト自体の品質の信用性を確認、評価する技術を提供することにある。

【課題を解決するための手段】

【００１５】

上記課題を解決する本発明の信用度管理システムは、分散台帳をそれぞれ保持する複数の検証ノードと前記検証ノードへトランザクションを発行する複数のトランザクション発行ノードから構成されるシステムであって、前記検証ノードそれぞれが、それぞれで管理するブロックチェーンにおいて、スマートコントラクトおよび前記スマートコントラクトの本番の実行トランザクションの他に、前記スマートコントラクトに対する評価実行トランザクションを管理し、前記検証ノードそれぞれが、所定の前記トランザクション発行ノードが指定した評価用の所定値と、前記トランザクション発行ノードから受け取ったトランザクションが評価実行トランザクションであった際に、前記所定値を前記スマートコントラクトに入力して得た出力値とを含む、前記スマートコントラクトの検証結果を、前記ブロックチェーンにおいて前記評価実行トランザクションとあわせて管理する、または前記出力値に基づき前記スマートコントラクトに関するステート情報を更新するものである、ことを特徴とする。

【００１６】

また、本発明の信用度管理方法は、分散台帳をそれぞれ保持する複数の検証ノードと前記検証ノードへトランザクションを発行する複数のトランザクション発行ノードから構成されるシステムにおいて、前記検証ノードそれぞれが、それぞれで管理するブロックチェーンにおいて、スマートコントラクトおよび前記スマートコントラクトの本番の実行トランザクションの他に、前記スマートコントラクトに対する評価実行トランザクションを管理し、前記検証ノードそれぞれが、所定の前記トランザクション発行ノードが指定した評価用の所定値と、前記トランザクション発行ノードから受け取ったトランザクションが評価実行トランザクションであった際に、前記所定値を前記スマートコントラクトに入力して得た出力値とを含む、前記スマートコントラクトの検証結果を、前記ブロックチェーンにおいて前記評価実行トランザクションとあわせて管理する、または前記出力値に基づき前記スマートコントラクトに関するステート情報を更新する、ことを特徴とする。

【発明の効果】

【００１７】

本発明によれば、中央集権機関による管理が無くとも、スマートコントラクト提供者以外の複数の第三者によってスマートコントラクト自体の品質の信用性を確認、評価することが可能となる。

【図面の簡単な説明】

【００１８】

【図１】本実施形態におけるコンピュータシステムを模式的に示す図である。

【図２】本実施形態における検証ノードの物理的な構成を示すブロック図である。

【図３】本実施形態の分散台帳上のブロックチェーンのデータ構造例を示す図である。
【図４】本実施形態の分散台帳上のステート情報のデータ構造例を示す図である。
【図５】本実施形態の信用度管理方法のフロー例１を示す図である。
【図６】本実施形態信用度管理方法のフロー例２を示す図である。
【図７】本実施形態の信用度管理方法のフロー例３を示す図である。
【図８】本実施形態の信用度管理方法のフロー例４を示す図である。
【図９】本実施形態のスマートコントラクト信用度計算結果のデータ構造例を示す図である。
【図１０】本実施形態の信用度管理方法のフロー例５を示す図である。
【図１１】本実施形態の信用度管理方法のフロー例６を示す図である。
【図１２】本実施形態の信用度管理方法のフロー例７を示す図である。
【図１３】本実施形態における各種機能を複数のノード上に配置したコンピュータシステムの構成例である。
【発明を実施するための形態】
【００１９】
－－－実施例１－－－
本実施例の信用度管理システムたる分散台帳システム１０は、図１で例示するように、検証ノード３およびクライアントノード４といった複数のノードから構成される情報処理システムである。
【００２０】
こうした分散台帳システム１０における各ノードは、ＳＣに対する評価用の実行トランザクション（以下、評価実行トランザクション）を、いずれかの他ノードより受け付けると、その評価実行トランザクションを用いて、ＳＣに対する本番の実行トランザクション（以下、本番実行トランザクション。請求項における「実行トランザクション」に該当）と同様にＳＣを実行し、分散台帳上に、本番実行トランザクションおよび評価実行トランザクションの各履歴とその実行結果とを含めて管理、保持する。
【００２１】
また各ノードは、上述の評価実行トランザクションの実行履歴を、ＢＣネットワークの参加者内で共有可能とする。本実施例におけるＢＣネットワークとは、所定のスマートコントラクトを共に利用する、検証ノード３およびクライアントノード４を含むネットワークである。
【００２２】
こうした分散台帳システム１０は、図１に関して説明したように、１台以上の検証ノード３、および、１台以上のクライアントノード４によって構成されている。これらの機器は、物理的な通信回線２を通してネットワーク１に接続される。
【００２３】
本実施例では、複数の主体（例えば複数の事業者）によって上述の検証ノード３がそれぞれ管理されていることを想定する。また、１名以上のＳＣ提供者と、複数のＳＣ利用者とが、それぞれ別のクライアントノード４を利用することを想定する。
【００２４】
上述のノードのうち検証ノード３は、トランザクション管理部３１、スマートコントラクト実行部３２（以下、ＳＣ実行部３２とも称する）、メンバー管理部３３、参照ＡＰＩ３４、スマートコントラクト評価管理部３５、およびスマートコントラクト監視部３６、の機能部と、分散台帳Ｄ１、参加メンバー管理情報Ｄ２、および、スマートコントラクト信用度計算結果Ｄ３のデータ群とによって構成される。
【００２５】
このうちトランザクション管理部３１は、合意形成処理部３１１を備える。検証ノード３は、トランザクション管理部３１の機能によって、例えばクライアントノード４からトランザクションを受け付けて、合意形成処理部３１１の機能によって、他の検証ノードとの間でトランザクションを受け入れてよいかの合意形成を行う。また、検証ノード３は、

この合意形成がなされたら、ＳＣ実行部３２の機能を介して、ＳＣのデプロイ、デプロイ済みのＳＣに対する本番実行、デプロイ済みのＳＣに対する評価実行処理を行い、トランザクションの履歴とその実行結果を分散台帳Ｄ１に記録する。また検証ノード３の参照ＡＰＩ３４は、他ノードからの所定要求に対して評価実行トランザクションの履歴情報を取得・閲覧する機能を提供する。

【００２６】

また、検証ノード３のメンバー管理部３３は、ＢＣネットワークに参加するメンバー（ノード）に関する所定の登録処理や、トランザクション処理に用いる鍵類の新規発行、認証機能等を提供する。本実施例におけるメンバー管理部３３では、各メンバーに発行した秘密鍵と公開鍵のペアを用いて、参加メンバーの認証やトランザクションへの署名、ＳＣ実行権限の制御等を行うことを想定する。なお、上述のメンバー管理に関して用いる公開鍵の情報Ｄ２２は、参加メンバー管理情報Ｄ２上に格納・管理される。

【００２７】

また、トランザクション管理部３１は、所定のノードからトランザクションを受け付けた際に、適宜、メンバー管理部３３の機能を介して、当該トランザクションの発行者が権限を持った正しい参加者かどうかを確認する。この機能自体は公知技術であるため説明は省略する。

【００２８】

また、分散台帳Ｄ１では、ブロックチェーンＤ１１とステート情報Ｄ１２を格納・管理している。本実施例では、ＢＣ上で評価実行トランザクションも共有可能とするために、ブロックチェーンＤ１１およびステート情報Ｄ１２には、それぞれ本番実行トランザクションに関する情報（Ｄ１１０、Ｄ１２０）と評価トランザクションに関する情報（Ｄ１１１、Ｄ１２１）の両方を保持している。

【００２９】

一方、クライアントノード４は、トランザクション発行部４１を含む機能部と、参加メンバー管理情報Ｄ２を含むデータ群とによって構成される。ＳＣの利用者もしくは提供者は、クライアントノード４のトランザクション発行部４１を介して、各種トランザクションを発行し、これを検証ノード３に対して送信することとなる。

【００３０】

なお、クライアントノード４は、トランザクションに付与する発行者情報として、参加メンバー管理情報Ｄ２に格納された当該メンバーの認証情報（秘密鍵Ｄ２１）を用いる。なお、各クライアントノード４と検証ノード３との間では、参加メンバー管理情報Ｄ２の公開鍵Ｄ２２が相互に交換されていることとする。

【００３１】

また、図１における検証ノード３が保持する、スマートコントラクト評価管理部３５、スマートコントラクト監視部３６、および、スマートコントラクト信用度計算結果Ｄ３、については、実施例２以降で説明するものとする。

【００３２】

ここで、上述の分散台帳システム１０を構成する各ノードのうち、一例として検証ノード３のハードウェア構成例について説明する。図２は、実施例１における検証ノード３の物理的な構成を示すブロック図である。

【００３３】

本実施例における検証ノード３は、インターフェイス１００、プロセッサ１０１、およびメモリ１０２を備える計算機である。インターフェイス１００、プロセッサ１０１、メモリ１０２はデータバス１０３によって接続される。

【００３４】

こうした構成の検証ノード３は、インターフェイス１００を介して、ネットワーク１と通信する。また、プロセッサ１０１は、ＣＰＵ等の演算装置である。メモリ１０２は、プログラムおよびデータを保持するための記憶領域である。プロセッサ１０１は、メモリ１０２からデータバス３４を介してプログラムを読み出し、実行する。このプログラムの実

行により、図１で例示した各機能部を実装することとなる。

【００３５】

続いて、分散台帳Ｄ１に保持するブロックチェーンの例について説明する。図３は、分散台帳Ｄ１上で管理するブロックチェーンＤ１１である。ＢＣ型の分散台帳管理では、所定時間帯における複数のトランザクションをブロックとしてまとめて、各ブロックが前の時間帯のブロックのハッシュ値を持つことでデータを数珠つなぎにして管理する。

【００３６】

このブロックチェーンにおいて、前段のブロックの値が１ビットでも変わると後続の全ブロックのハッシュ値も変化するため、改ざんが行われても容易に検知される。なお、本実施例では説明をシンプルにするために、１トランザクションにつき１ブロックに格納し、当該ブロックを連ねてブロックチェーンを生成する例を想定する。但し、複数トランザクションをまとめて１ブロックに格納し、こうしたブロックを連ねてブロックチェーンを生成する形態も想定可能である。

【００３７】

図３で示すブロックチェーンＤ１１は、ブロックＤ１１２～Ｄ１１５が連なったブロックチェーンである。ここで示すブロックＤ１１２～Ｄ１１５の各ブロックは、ブロックＤ１１２がデプロイトランザクション、ブロックＤ１１３が本番実行トランザクション、および、Ｄ１１４が評価実行トランザクション、の各情報を含むものである。

【００３８】

また、各ブロックＤ１１２～Ｄ１１５は、そのブロック生成時のタイムスタンプ情報を含む。さらに各ブロックＤ１１２～Ｄ１１５は、ブロックチェーンＤ１１の連なりにおける前ブロックのハッシュ値を含み、後述のステート情報Ｄ１２から生成したハッシュ値を含む。

【００３９】

上記のようなデータ構造により、デプロイ、本番実行、および評価実行、の各トランザクションが、ブロックチェーンＤ１１の中でデータの連鎖として管理される。

【００４０】

上述のブロックチェーンＤ１１を構成するブロックのうち、ブロックＤ１１２は、デプロイトランザクションを格納したブロックの一例である。本実施例のデプロイトランザクションは、ＳＣを一意に識別するコントラクトＩＤ、ＳＣの実体（例えば実行可能なバイナリ）を含む。また、デプロイトランザクションは、ＳＣが持つ関数名やその引数を利用者が把握するためのコントラクト入力仕様を含む。また、デプロイトランザクションは、このデプロイトランザクションの発行元、すなわち、提供者を識別するための発行者ＩＤを含む。また、デプロイトランザクションは、発行元およびデータに改ざんが無いことを検証するために用いる電子署名を含む。この電子署名は、メンバー管理部３３が発行した各ＢＣネットワーク参加メンバー（すなわちＳＣ提供者や利用者）の秘密鍵を用いて生成され、そのペアとなる公開鍵によって検証をすることが可能である。

【００４１】

また、ブロックＤ１１３は、本番実行トランザクションを格納したブロックの一例である。本実施例の本番実行トランザクションは、呼び出し対象となるスマートコントラクトのコントラクトＩＤ、および、その関数名と入力する引数の情報を含む。また、ブロックＤ１１３は、この本番実行トランザクションの発行元、すなわち、利用者を識別するための発行者ＩＤを含む。また、ブロックＤ１１３は、発行元とデータに改ざんがないことを検証するために用いる電子署名を含む。

【００４２】

また、ブロックＤ１１４は、評価実行トランザクションを格納したブロックの一例である。本実施例の評価実行トランザクションは、本番実行トランザクションと同様に、スマートコントラクトを一意に識別するコントラクトＩＤ、および、その関数名および入力する引数の情報を含む。また、ブロックＤ１１４は、この評価実行トランザクションの発行元、すなわち、利用者を識別するための発行者ＩＤを含む。また、ブロックＤ１１４は、

発行元とデータに改ざんがないことを検証するために用いる電子署名を含む。

【００４３】

このブロックＤ１１４は、さらに、スマートコントラクトの評価、すなわち検証結果に関する情報として、評価ＩＤと期待値を含む。評価のシナリオとして、複数のトランザクションに対する実行結果を確かめたい場合や、初期状態からトランザクションを実行したい場合がある。そこで本実施例では、一連の評価シナリオを一意に特定する識別子として評価ＩＤを導入している。さらに、評価ＩＤ毎に、ステート情報Ｄ１２で利用するデータ領域を独立させることにより、評価間の相互影響を排除できる。

【００４４】

ここで、上述の期待値は、この評価実行トランザクションの実行結果に期待する値もしくはその許容範囲である。なお、本実施例では、評価実行トランザクションのブロックＤ１１４において、評価実行トランザクションの実行結果も格納・保持することとする。通常、トランザクションの実行結果はＢＣを辿って各トランザクションを再実行すれば取得することができる。しかし、評価実行トランザクションの実行結果は、利用者によって参照されることが想定されるため、その度に実行し直すのは処理効率が悪い。そのため、評価実行トランザクションにて実行結果も格納することとしている。さらに評価実行トランザクションのブロックＤ１１４は、評価実行トランザクションの実行結果が期待値を満たすかどうかを判定した結果である評価判定情報（例えば、期待値を満たせば「ＯＫ」、満たさなければ「ＮＧ」）を格納・保持する。

【００４５】

続いて、分散台帳Ｄ１上で管理するステート情報Ｄ１２について説明する。図４は、分散台帳Ｄ１上で管理するステート情報Ｄ１２のデータ構成例を示す図である。

【００４６】

ＢＣ型の分散台帳管理では、通常、（最新の）ステート（例えば、仮想通貨の場合にはアカウントの残高）を取得するためには、ＢＣを辿らなければならない。これでは処理効率が悪いため、ＢＣとは別に、最新のステート情報をキャッシュしておく方法が存在する（非特許文献３等）。本実施例でも、ノードが最新のステート情報を保持することを想定している。本実施例では、スマートコントラクト毎にステートのデータ領域が用意されることとする。よって、ステート情報Ｄ１２は、スマートコントラクトを一意に識別するコントラクトＩＤとそのコントラクトの実体を保持する。これにより、ＳＣ実行部３２は、コントラクトＩＤをキーにして、スマートコントラクトの実体を取得して実行することができる。また、ステート情報Ｄ１２は、ＳＣの実行結果を保持するための内部テーブルを備える。各ノードは、トランザクションが実行される度にこの内部テーブルの内容を更新する。この内部テーブルは、本番実行、評価実行の各トランザクション毎にデータ領域（Ｄ１２０、Ｄ１２１）を有している。これにより、評価実行のトランザクションが本番実行のトランザクションに影響を与えることを防ぎ、各評価が互いに影響することも防ぐことができる。

【００４７】

なお、図３および図４に示した分散台帳Ｄ１２中では、ＩｏＴを活用した貨物輸送に関するビジネスコントラクトにブロックチェーン基盤を適用する具体例を示している。この具体例では、ある貨物について、工場から小売店までの出荷をする際に、複数の輸送者が貨物の輸送を中継する場合のサプライチェーンを想定する。また、ビジネスコントラクトとして、輸送中に貨物の最大湿度が８０％を超えたら検査が必要となって出荷停止となるため、湿度基準を違反した業者が全損失の責任を負う、という契約を事業者間で締結することとする。また、各貨物はＩｏＴデバイスとなっており、湿度センサーが搭載されていて貨物の湿度情報（ある種のＩｏＴデータ）を定期的に取得・記録する。各輸送者のノードは、当該輸送者による輸送が完了する度に記録された湿度情報を、ＢＣ上に登録し、ビジネスコントラクトに対応したスマートコントラクトを自動実行することで、複数の業者間で同一の信頼されたデータを共有でき、契約に基づく処理を自動執行できる。

【００４８】

図３にて示したブロックＤ１１２でデプロイされているＳＣは、上記のビジネスコントラクトを実現するＳＣとなっている。本ＳＣは、コントラクトＩＤ「貨物輸送コントラクト」として、発行者たる「製造者Ａ」によって提供され、以下の関数が定義されている。

【００４９】

・関数名：輸送（）、入力引数：貨物ＩＤ、輸送者、湿度情報、戻り値：違約金

・関数名：検収（）、入力引数：貨物ＩＤ、検収者、戻り値：なし

・関数名：輸送履歴参照（）、入力引数：貨物ＩＤ、戻り値：輸送履歴

上述の関数のうち、「輸送（）」が、このＳＣのコア処理となる。輸送（）は、各輸送者による輸送が完了したタイミングで輸送者のノードによって呼び出され、入力引数として、対象となる貨物を識別する貨物ＩＤ、輸送者の情報、湿度センサーから得られた湿度情報を登録されると、ＳＣの内部で、最大湿度が８０％を超えたかどうかで契約違反の有無を判定し、違反をした場合には超過した湿度に応じて違約金を計算する（湿度に応じた従量計算）。そしてその結果をステート情報Ｄ１２にて格納し、戻り値として違約金の額を返す。

【００５０】

また、図３にて示したブロックＤ１１３は、上述の関数「輸送（）」を呼び出した本番実行トランザクションの例である。この例では、「輸送者Ａ」による貨物ＩＤ「１２３」の輸送が完了し、センサーから得られた湿度情報が「６０％」だった時のトランザクションであることを示す。この実行によって更新されたステート情報Ｄ１２は、Ｄ１２０中の内部テーブルの貨物ＩＤ「１２３」の行である。

【００５１】

この行に示す通り、ＳＣ実行の結果、最大湿度が８０％を超えていないため、契約は「順守」となり、違約金は「０」となる。一方、貨物ＩＤ「２３４」の行では、「輸送者Ｂ」による輸送の湿度が８０％を超えたため、契約は「違反」となり、違約金「２０００」が発生していることを示している。

【００５２】

この例の場合には、ＳＣは製造業者によって提供されるため、利用者である輸送者にとってＳＣの内部処理はブラックボックスとなる。したがって輸送者は、このＳＣの関数の入力引数仕様しかわからない。すなわち、利用者においてはＳＣの内部で処理される計算式が妥当かどうか、認識齟齬がないかどうかを判断することができない。例えば、この例では湿度が８０％ちょうどの場合には違約金が発生するのかどうか、超過した湿度に応じた違約金の計算結果が利用者の想定にマッチしているか等が挙げられる。

【００５３】

ビジネスコントラクトの場合には、別途、契約書面が存在することが予想されるが、契約書の記載はあいまいな表現も多く判断がつかない可能性がある。一方、ＳＣによると契約が自動執行される（さらにしばしば金融決済を伴う）ためリスクが大きい。

【００５４】

そのような場合に、本実施例における評価実行トランザクションの実行機能を活用できる。

【００５５】

また、図３で例示したブロックＤ１１４は、上述のスマートコントラクトの関数「輸送（）」に対する評価実行トランザクションを格納したブロック例である。この例では、評価ＩＤ「１」として「輸送者Ｂ」によって、貨物ＩＤ「３４５」の輸送を「輸送者Ｂ」が行い、湿度情報が「８５％」であった場合の評価を示す。ここでは、評価の結果、「違約金≦１０００」となることを期待値として当該輸送者が設定している。一方、評価実行トランザクションの実行結果は「違約金＝２０００」となり、期待値を満たさない。

【００５６】

これにより「輸送者Ｂ」は、ＳＣの内部処理が自身の期待を満たしていないことを本番実行の前に検知することができる。また、この履歴は他の利用者にも共有されることによって、同様の評価履歴があれば、利用者自ら評価をしなくとも期待を満たしているかどう

かを確認できる。
【００５７】
続いて、本実施例における信用度管理方法のフロー例について説明する。図５は、ＢＣネットワークに参加するメンバーの新規登録処理の例を示すフローチャートである。
【００５８】
この場合、検証ノード３のメンバー管理部３３は、クライアントノード４等の他ノードからメンバー登録要求を受け付ける（Ｓ１０１）。ここで、上述のメンバー登録要求には、要求メンバーを一意に識別するメンバーＩＤが含まれることとする。
【００５９】
次に、メンバー管理部３３は、所定の鍵生成ツール等により、秘密鍵Ｄ２１と公開鍵Ｄ２２の組を生成して、Ｓ１０１で受け取ったメンバーＩＤと紐付ける（Ｓ１０２）。
【００６０】
次に、メンバー管理部３３は、新規登録対象となるメンバーＩＤと、Ｓ１０２で生成した公開鍵Ｄ２２とを、他のノードにブロードキャストする（Ｓ１０３）。ここでブロードキャストされたメンバーＩＤと公開鍵Ｄ２２の各情報は、各ノード上で参加メンバー管理情報Ｄ２として保管される。
【００６１】
さらに、メンバー管理部３３は、上述のメンバー登録要求を行ったノードに対し、Ｓ１０２で生成した秘密鍵Ｄ２１を返す（Ｓ１０４）。この秘密鍵Ｄ２１を受け取ったノードは、参加メンバー管理情報Ｄ２に自身の秘密鍵Ｄ２１として保管することとなる。
【００６２】
本実施例では、以上のようにして生成した秘密鍵と公開鍵のペアを用いて、ＢＣネットワーク参加メンバーの認証やトランザクションへの署名、ＳＣ実行権限の制御等を行うことを想定する。具体的には、例えば、クライアントノード側は、上述のメンバー管理部３３にて発行された秘密鍵で電子署名したトランザクションを発行し、一方、検証ノード側は、このクライアントノードの公開鍵を用いて該当電子署名を検証することで、本人確認を実現できる。なお、公開鍵と秘密鍵の組を生成する手法や署名検証をする手法、鍵とＩＤを紐付ける手法については、公知または周知の技術を適用すれば良い。
【００６３】
続いて、トランザクション実行処理、すなわち、ＳＣデプロイ、本番実行、評価実行処理のフロー例について説明する。図６は、信用度管理方法のフロー例２を示す図である。
【００６４】
この場合、検証ノード３のトランザクション管理部３１は、クライアントノード４等のトランザクション発行元からトランザクションを受け取る（Ｓ２０１）。
【００６５】
また、トランザクション管理部３１は、Ｓ２０１で受け取ったトランザクションの種別を判定し（Ｓ２０２）、この判定で判明した種別、すなわち、ＳＣデプロイ、本番実行、および、評価実行の各処理を行う。
【００６６】
上述の判定の結果、受け取ったトランザクションがデプロイトランザクションであった場合（Ｓ２０２：ＮＯ、Ｓ２０３：ＹＥＳ）、トランザクション管理部３１は、他の検証ノード３との間で、受け取ったトランザクションをブロックとして、ブロックチェーンＤ１１の末尾に追加してよいかの合意形成処理を行う（Ｓ２０４）。具体的な合意形成処理方式としては、公知または周知の技術を適用すれば良い。
【００６７】
具体的には、例えば、Ｐｌａｃｔｉｃａｌ Ｂｙｚａｎｔｉｎｅ Ｆａｕｌｔ Ｔｏｌｅｒａｎｃｅ（ＰＢＦＴ）と呼ばれるアルゴリズム等を採用することが考えられる。ＰＢＦＴは合意形成に参加するすべてのノード（すなわち検証ノード）の間で一定（３分の２）以上のノードによる合意を条件とするアルゴリズムである。
【００６８】

ＰＢＦＴをベースとした合意形成を簡単に説明すると、検証ノード３はまず受け取ったトランザクションをネットワークに参加するすべての検証ノード３に対してブロードキャストし、各検証ノード３でトランザクションに対する署名検証を行って改ざんがされていないことやトランザクションの内容の正当性を確認し、その確認結果を他の検証ノード３に対してブロードキャストする。一定数以上の検証ノード３による確認が取れた場合にトランザクションの承認準備が完了したことを他の検証ノード３に対してブロードキャストする。そして、一定数以上の検証ノード３による承認準備完了が確認できたことをもって合意形成が完了する。

【００６９】

上述の合意形成が完了したら、トランザクション管理部３１は、ＳＣ実行部３２を介して、当該トランザクションに含まれるＳＣを分散台帳Ｄ１に登録する（Ｓ２０５）。具体的には、当該トランザクションの内容に基づき、ステート情報Ｄ１２のコントラクトＩＤとコントラクト実体を登録し、ブロックチェーンＤ１１の末尾にこのデプロイトランザクションを含むブロックを追加する。

【００７０】

次に、トランザクション管理部３１は、上述のデプロイトランザクションの実行結果をトランザクション発行元のノードに返し（Ｓ２０６）、処理を終了する。

【００７１】

一方、受け取ったトランザクションが本番実行トランザクションであった場合（Ｓ２０２：ＮＯ、Ｓ２０３：ＮＯ）、トランザクション管理部３１は、デプロイトランザクションと同様、他の検証ノード３との間で合意形成処理を行う（Ｓ２０７）。この合意形成処理はＳ２０４と同様である。

【００７２】

上述の合意形成が完了後、トランザクション管理部３１は、ＳＣ実行部３２を介して、ＳＣを実行する（Ｓ２０８）。具体的には、本番実行トランザクション内で指定されたコントラクトＩＤを持つＳＣ（登録済みであることを前提とする）に対して、本番実行トランザクション内で指定された呼び出し関数と入力引数を与えて実行する。

【００７３】

トランザクション管理部３１は、その実行結果に基づき、分散台帳Ｄ１の内容を更新する（Ｓ２０９）。また、トランザクション管理部３１は、上述の実行結果に基づいて、このスマートコントラクトに関するステート情報Ｄ１２を更新し、ブロックチェーンＤ１１の末尾のブロックとして本番実行トランザクションを追加する。

【００７４】

最後に、トランザクション管理部３１は、この本番実行トランザクションの実行結果（例えば、関数の戻り値）をトランザクション発行元のノードに返し（Ｓ２１０）、処理を終了する。

【００７５】

他方、受け取ったトランザクションが評価実行トランザクションであった場合（Ｓ２０２：ＹＥＳ）、トランザクション管理部３１は、デプロイトランザクションと同様に合意形成処理を行う（Ｓ２１１）。本合意形成処理はＳ２０４と同様である。

【００７６】

続いて、トランザクション管理部３１は、上述にて合意形成が完了したら、ＳＣ実行部３２を介して、評価実行トランザクションを入力としてＳＣを実行する（Ｓ２１２）。具体的には、評価実行トランザクション内で指定されたコントラクトＩＤを持つＳＣ（本番実行トランザクションと同じバイナリを利用）に対して、評価実行トランザクション内で指定された呼び出し関数と入力引数を与えて実行する。

【００７７】

トランザクション管理部３１は、上述のスマートコントラクト実行の結果に基づき、分散台帳Ｄ１における、本スマートコントラクトに関するステート情報Ｄ１２を更新する（Ｓ２１３）。その際、トランザクション管理部３１は、本番実行トランザクション用とは

別に用意された、評価ＩＤ毎の評価実行トランザクション用のデータ領域に、ステート情報Ｄ１２を格納する。また、トランザクション管理部３１は、ブロックチェーンＤ１１の末尾のブロックとして評価実行トランザクションおよびその実行結果を追加する。

【００７８】

最後に、トランザクション管理部３１は、この評価実行トランザクションの実行結果（例えば、関数の戻り値と評価判定）をトランザクション発行元に返し（Ｓ２１４）、処理を終了する。

【００７９】

以上のフローにより、本番実行トランザクションのデータ領域を汚すことなく、本番と同一のスマートコントラクトの実体を用いて、本番と同一の入力を用いたトランザクション評価実行を実現する。

【００８０】

ここで、上述で説明した評価実行トランザクションに関して、検証ノード３の参照ＡＰＩ３４が、実行する処理について説明する。図７は、参照ＡＰＩ３４による評価実行トランザクション履歴の取得フローを示す図である。

【００８１】

この場合、参照ＡＰＩ３４は、クライアントノード４等から発行された評価実行トランザクション履歴の取得要求を受け取る（Ｓ３０１）。この要求では、取得対象となるＳＣのコントラクトＩＤが必ず指定され、さらに必要に応じて、呼び出し関数名、利用者、評価ＩＤ、期待値が指定されていてもよい。

【００８２】

参照ＡＰＩ３４は、上述の要求を受け取ると、分散台帳Ｄ１のブロックチェーンＤ１１上から、指定されたＳＣに関する　評価実行トランザクションの履歴を格納したブロックをすべて取得する（Ｓ３０２）。

【００８３】

また、参照ＡＰＩ３４は、Ｓ３０２で取得した評価実行トランザクションの履歴を、評価ＩＤ毎にグループ化する（Ｓ３０３）。

【００８４】

また、参照ＡＰＩ３４は、Ｓ３０３でグループ化した評価実行トランザクションの履歴を加工して（例えば、ＪＳＯＮ（ＪａｖａＳｃｒｉｐｔ　Ｏｂｊｅｃｔ　Ｎｏｔａｔｉｏｎ）形式の配列として）、要求元のノードに返し（Ｓ３０４）、処理を終了する。

【００８５】

以上で示したとおり、提供されたＳＣに対して、提供者および利用者がそのトランザクションの評価を行うことが可能である。また、その評価結果は、ＢＣのデータの連なりの中で、本番実行トランザクションと共に管理され、他の参加者と共有することができる。

この仕組みにより、ＳＣの利用者は、中央集権機関による管理なしにＳＣが信用できるかどうか確認／評価できることになる。また、評価実行トランザクションをＢＣのデータの連なりの中で管理することで、改ざんを困難なものとし、かつ、ＢＣネットワークへの参加者に公開可能となる。さらに、各ノードのユーザらは、ＳＣの信用性を随時確認／評価できるため、悪意ある提供者による品質の悪いＳＣの提供を抑止する効果も奏する。スマートコントラクト提供者と複数の第三者による試行／評価により、従来の中央集権的な評価方法と比べて、スマートコントラクトの信用性を高めることができる。また、その信用性に関する情報の共有により、第三者が個々にスマートコントラクトを検証する場合に比べて手間を削減できる効果がある。

【００８６】

また、評価実行トランザクションを識別して、本番実行トランザクションとは異なるデータ領域上で、本番と同一のスマートコントラクトの実体を用いて、本番と同一の入力を用いたトランザクションの評価実行を可能とする。このことで、本番実行トランザクションのデータを汚染すること無く、スマートコントラクトの評価を行うことができるという効果がある。

【００８７】
－－－実施例２－－－
続いて、ＢＣ上に共有された評価実行トランザクションを活用する形態のバリエーションとして、評価実行トランザクションの履歴を利用してＳＣに対する信用度を計算する例を示す。
【００８８】
本実施例においては、図１で例示した分散台帳システム１０たるコンピュータシステムのうち、検証ノード３におけるスマートコントラクト評価管理部３５およびスマートコントラクト信用度計算結果Ｄ３に基づく機能等について説明する。以降では、実施例１と異なる部分についてのみ説明し、実施例１と同様の部分についての説明は省略する。
【００８９】
この場合のスマートコントラクト評価管理部３５（以下、ＳＣ評価管理部３５とも称する）は、ブロックチェーンＤ１１上の評価実行トランザクションの履歴Ｄ１１１を用いて、ＳＣや提供者がどの程度信用できるかの度合いを定量的に示す指標であるスマートコントラクト信用度（以下、ＳＣ信用度とも称する）を計算し、その結果をスマートコントラクト信用度計算結果Ｄ３（以下、ＳＣ信用度計算結果Ｄ３とも称する）上に格納・保持する。ここで、上述のＳＣ信用度は、０．０～１．０の間でスコアリングされ、値が大きいほど信用性が高いことを示すものとする。
【００９０】
図８は、評価実行トランザクションの履歴に基づくスマートコントラクト信用度計算処理の例を示すフローチャートである。この場合のＳＣ評価管理部３５は、参照ＡＰＩ３４を介して、分散台帳Ｄ１のブロックチェーンＤ１１からＳＣ毎に評価実行トランザクションの履歴をすべて取得する（Ｓ４０１）。図３を例にとると、ブロックＤ１１４等を取得することとなる。
【００９１】
次に、ＳＣ評価管理部３５は、Ｓ４０１で取得した各評価実行トランザクションに含まれる利用者（発行者）、評価判定、の各情報に基づき、関数毎およびこのＳＣ全体のＳＣ信用度を計算する（Ｓ４０２）。
【００９２】
例えば、本実施例では、各ＳＣの関数毎のＳＣ信用度を、評価実行トランザクションの全件のうち、評価判定情報が「ＮＧ」ではなく「ＯＫ」だった件数の割合によって求めることとする。また、提供者よりも利用者が行った評価のほうが、より客観性が高く信用できる評価と見なして重み付けを行うこととする。例えば、評価実行トランザクションの利用者が、スマートコントラクトの提供者でない場合には２倍の重み付けをすることとする。
【００９３】
ここで、図３で用いたブロックチェーンＤ１１の例をベースとして、スマートコントラクトにおける関数「輸送（）」の評価実行トランザクションの履歴が以下のとおりだった場合を想定する。
【００９４】
・「製造者Ａ」によるＳＣ評価実行回数が１０件、うち１０件が評価判定「ＯＫ」
・「輸送者Ａ」によるＳＣ評価実行回数が５件、うち４件が評価判定「ＯＫ」
・「輸送者Ｂ」によるＳＣ評価実行回数が１５件、うち１３件が評価判定「ＯＫ」
この場合には、ＳＣ評価実行回数が、１０件×１倍＋５件×２倍＋１５件×２倍＝５０件、となる。また、評価判定「ＯＫ」だった件数が、１０件×１倍＋４件×２倍＋１３件×２倍＝４４件、となる。したがって、この関数のＳＣ信用度は４４／５０＝０．８８となる。さらに本実施例では、各ＳＣのＳＣ信用度を、上記のように算出した各関数のＳＣ信用度の平均値によって求めることとする。
【００９５】
次に、ＳＣ評価管理部３５は、ＳＣ提供者毎のＳＣ信用度計算結果に基づき、提供者の

ＳＣ信用度を計算する（Ｓ４０３）。本実施例では、ある提供者によるすべてのＳＣに対するＳＣ信用度の平均値によって計算することとする。

【００９６】

最後に、ＳＣ評価管理部３５は、以上によって求めたＳＣ信用度の計算結果を、ＳＣ信用度計算結果Ｄ３に格納し（Ｓ４０４）、処理を終了する。

【００９７】

図９は、ＳＣ信用度計算結果Ｄ３のデータ構造の例である。本実施例では、ＳＣ信用度の計算結果を、このようにテーブル上に保持することとする。当該テーブルでは、ＳＣの提供者Ｄ３０１、ＳＣのコントラクトＩＤ（Ｄ３０２）、および関数名Ｄ３０３に対し、信用度Ｄ３０４が対応付けて管理される。また、評価者Ｄ３０５で示すように、スマートコントラクトの評価、すなわち信用度計算を行ったＢＣネットワークメンバーの情報を保存しておくとしても良い。スマートコントラクト信用度Ｄ３における行Ｄ３１１は、ある関数のＳＣ信頼度、行Ｄ３１２はあるＳＣのＳＣ信頼度、行Ｄ３１３はある提供者のＳＣ信頼度をそれぞれ示している。

【００９８】

なお、本実施例に示したＳＣ信用度では、シンプルな例として「ＯＫ」だった割合をベースに計算しているが、実行された評価回数を掛けあわせて計算を行うとしても良い。これにより、例えば、試行やテストの回数が少ないため評価が不十分な場合には信用度がより低くなるように算出することができる。

【００９９】

こうしてスマートコントラクトの信用度を定量化することによって、ＳＣ、その関数、あるいは提供者が信用できるかどうかを一目で把握することができる。このように複数の第三者によるテスト／評価結果を活用することで、ＳＣの信用度評価をより客観的に行うことができる効果がある。

【０１００】

－－－実施例３－－－

続いて、評価実行トランザクションの履歴を活用してトランザクションの実行を制御する例について説明する。なお、本実施例における分散台帳システム１０たるコンピュータシステムの構成は、上述の実施例１あるいは実施例２と同様である。また、トランザクションの実行制御の判定基準として、ＳＣ信頼度を用いる場合には実施例２の構成を、他方、ＳＣ信頼度を用いない場合には実施例１の構成をとるものとする。

【０１０１】

本実施例では、上述の実行制御の一例として、評価の状況に応じてトランザクションの本番実行を不可とする場合について示す。

【０１０２】

図１０は、スマートコントラクトの評価状況に基づく本番実行判定処理を含むフローチャートの例である。このフローチャートは図６と概ね一緒であるため、ここでは差分のみを説明する。

【０１０３】

この場合、トランザクション管理部３１は、ノードから受け取ったトランザクションが本番実行トランザクションだった場合（Ｓ２０３：ＮＯ）、該当ＳＣが関係する評価状況に基づいて本番実行可否判定を行う（Ｓ２１６）。

【０１０４】

例えば、判定基準としては、ブロックチェーンＤ１２の評価実行トランザクション履歴を参照して、「本ＳＣに対してＳＣ提供者によって少なくとも１件以上の評価実行トランザクションが実行されていなければ実行不可」、「複数名の利用者によって評価実行トランザクションが実行されていなければ実行不可」、ＳＣ信用度計算結果Ｄ３を参照して「ＳＣ信用度が一定のしきい値を超えていなければ実行不可」、等のバリエーションが挙げられる。

【０１０５】

上述の判定の結果、本番実行が「実行可」であった場合（Ｓ２１５：実行可）、トランザクション管理部３１は、Ｓ２０７～Ｓ２１０と同様に本番実行トランザクションを実行する。一方、上述の判定の結果、本番実行が「実行不可」であった場合（Ｓ２１５：実行不可）、トランザクション管理部３１は、本番実行を行わず、トランザクション発行元のノードに実行不可（例えばエラーメッセージ）を返し（Ｓ２１６）、処理を終了する。

【０１０６】

このように、スマートコントラクトの評価結果（ＳＣ信用度計算結果）や、あるいは評価実行トランザクションが該当ブロックチェーンに格納済みか否かは、当該スマートコントラクトの信用に基づいたトランザクションの実行制御に活用することができる。

【０１０７】

他の実行制御バリエーションとしては、例えば、Ｓ２０１とＳ２０２の間にトランザクションの優先度付きのキューを用意して、評価状況に応じて優先的に実行するべきトランザクションを制御する例が考えられる。

【０１０８】

例えば、システム全体の負荷が高い場合には、評価実行よりも本番実行を優先し、さらに評価実行の中でもより評価実行回数が少なかったり、信用性の低かったりするトランザクションから優先する制御方法が考えられる。また、別の例として、評価実行回数が既に多く、信頼度が既に高い評価実行トランザクションについては追加で評価を行う効果が低いため、評価実行トランザクションを受け付けない等の制御方法も考えられる。

【０１０９】

－－－実施例４－－－

続いて、上述の実施例２におけるＳＣ信頼度計算のバリエーションとして、評価実行トランザクションだけでなく本番実行トランザクションの履歴も活用する例について説明する。なお、本実施例における分散台帳システム１０たるコンピュータシステムの構成は、実施例２と同様であり、内部のＳＣ信頼度計算処理のみが異なる。

【０１１０】

図１１に、本番実行および評価実行の各トランザクションの履歴に基づくスマートコントラクト信用度計算処理のフロー例を示す。

【０１１１】

本実施例に記載の計算方法では、評価実行トランザクションの期待値を本番実行トランザクションに当てはめる。また、評価実行トランザクション間の期待値のずれも考慮する。その実現のための前提として、評価実行同士あるいは評価実行と本番実行の各トランザクションを部分一致によってマッチングできるようにする。

【０１１２】

具体的な実現方法はいくつか存在する。例えば、各評価実行トランザクションにおいて、ＳＣ利用者が、入力引数のうち期待値に影響を及ぼす重要な引数を発行時に指定する。例えば図３の例では、入力引数のうち「貨物ＩＤ」は評価において重要ではなく（すなわち任意の値で良い）、「湿度情報」が重要である。別の実現方法としては、入力引数に対して特定の固定値ではなく、入力値の範囲や入力条件を定めても良い。これらの情報を活用することで、項目等が完全一致することを前提条件とせずともトランザクション間のマッチングを取ることができる。

【０１１３】

この場合、ＳＣ評価管理部３５は、参照ＡＰＩ３４を介して、分散台帳Ｄ１のブロックチェーンＤ１１からＳＣ毎に評価実行トランザクションおよび本番実行トランザクションの履歴をすべて取得する（Ｓ５０１）。本番実行トランザクションも対象としている以外はＳ４０１と同様である。

【０１１４】

次に、ＳＣ評価管理部３５は、Ｓ５０１で取得した評価実行トランザクション同士をマッチングして、その期待値が相反するものを抽出し、当該相反するトランザクションを発行した利用者の多数決によって、少数派のトランザクションを、計算対象とするトランザ

クションから除外する（Ｓ５０２）。
【０１１５】
例えば、同じ入力に対して期待値「１０００≦違約金≦２０００」の評価実行トランザクションが、利用者「輸送者Ａ」および「製造者Ａ」によって実行され、期待値「違約金＝０」のトランザクションが、利用者「輸送者Ｂ」によって実行されたとする。その場合、ＳＣ評価管理部３５は、後者のトランザクションを除外する。なお、この方法は期待値が相反する場合に対応するための一例であり、例えば利用者自身の信用度等で重み付けを行っても良い。
【０１１６】
さらに、ＳＣ評価管理部３５は、評価実行トランザクションと本番実行トランザクションをマッチングして、マッチした本番実行トランザクションに対して評価実行の期待値をあてはめて評価する（Ｓ５０３）。
【０１１７】
以降、ＳＣ評価管理部３５は、上述のＳ５０２～Ｓ５０３で除外されずに使われたトランザクションに含まれる利用者（発行者）、評価判定の各情報に基づき、関数毎およびこのＳＣ全体のＳＣ信用度、提供者のＳＣ信用度を計算し（Ｓ５０４～Ｓ５０５）、その結果をＳＣ信用度計算結果Ｄ３に格納する（Ｓ５０６）。この処理は、計算対象として使われるトランザクションが異なる以外はＳ４０２～Ｓ４０４の処理と同様である。
【０１１８】
以上のようにして、評価実行トランザクションだけでなく本番実行トランザクションの履歴も活用してＳＣ信頼度を計算することで、計算に用いるサンプル数を増やすことが出来る。このため、実施例２よりも計算精度が向上する効果がある。また、期待値の相反するトランザクションを考慮することで、実施例２よりも計算精度が向上する効果がある。
【０１１９】
－－－実施例５－－－
続いて、評価実行トランザクションを自動発行することで、ＳＣの健全性や信用性の監視を行う応用例について説明する。本実施例の分散台帳システム１０たるコンピュータシステムの構成は、上述の実施例１～４の構成において、スマートコントラクト監視部３６（以下、ＳＣ監視部３６と称する）の構成が機能する点が異なる。
【０１２０】
図１２は、評価実行トランザクション自動発行によりＳＣの健全性や信用性を監視する処理の例を示すフローチャートである。
【０１２１】
この場合、ＳＣ監視部３６は、所定時間の経過毎に（Ｓ６０１：ＹＥＳ）、評価実行トランザクションを生成する（Ｓ６０２）。この評価実行トランザクション生成において、対象となる評価実行トランザクションが予め定義されているとしてもよいし、或いは、評価実行トランザクションの履歴を参照して流用するとしてもよい。また、実施例４と同様に、評価実行トランザクションに対して入力引数の重要項目を定義、あるいは入力引数に対する範囲や入力条件を定めておき、評価実行トランザクションの一部情報をＳＣ監視部３６が自動生成するとしてもよい。さらに、生成する評価実行トランザクションは１件ずつでも複数件でもよい。
【０１２２】
続いて、ＳＣ監視部３６は、Ｓ６０２で生成した評価実行トランザクションを、トランザクション管理部３１に送信する（Ｓ６０３）。この場合のトランザクション管理部３１は、Ｓ６０３で送信された評価実行トランザクションを受信し、図６等で示したフローに従ってＳＣを実行することになる。
【０１２３】
ＳＣ監視部２６は、上述のトランザクション管理部３１におけるＳＣの実行結果を受け取る（Ｓ６０４）。
【０１２４】

また、ＳＣ監視部３６は、Ｓ６０４で得た実行結果に基づいて、アラートの生成や監視結果の記録等を行う（Ｓ６０５）。例えば、Ｓ６０４で得た実行結果における評価判定が「ＮＧ」だった場合、ＳＣ監視部３６は、例えば外部の監視システムに対するアラート通知や、運用管理者やＳＣ利用者に対するアラートメールの送信を行う。また、監視結果の記録としては、実行日時と、評価判定の「ＯＫ」、「ＮＧ」や処理にかかった時間などからレコードを生成し、これを所定のログ管理用のデータベース等に登録し、後に正常稼働率等の稼働傾向の分析を実行するとしてもよい。

【０１２５】

このようにして評価実行トランザクションを任意のタイミングで自動発行し、抜き打ちでのスマートコントラクト評価を行うことで、スマートコントラクトの改ざん等に対するモニタリングを行うことができる。そのため、結果としてスマートコントラクトの提供者による悪意ある改ざんなどを抑止し、スマートコントラクトの信用性を継続的に維持することができるという効果がある。また、従来のように、ＢＣ基盤においてダミートランザクションを発行して、サービスの健全性を監視する場合、本番と同じようにトランザクションを取り扱うため、本番環境が汚染される可能性があった。ところが、本実施例では、データ領域を分けることにより本番環境の汚染をすることなく監視を行うことができるという効果がある。

【０１２６】

以上で示した各実施例では、単一の検証ノード３上に各種機能部を配置した例を示したが、これらは機能を逸脱しない範囲で、別々のサーバに配置されていてもよい。図１３には、各機能部を別々のサーバに配置した構成例を示す。

【０１２７】

さらに、各実施例においては、クライアントノード４とＢＣネットワーク参加者が一対一で対応する例を示した。しかし、本発明はこのような構成のみに限定されない。クライアントノード４をＢＣネットワーク参加者各々が持つのではなく、ＢＣネットワーク参加者各々が、外部端末からクライアントノード４にアクセスして、クライアントノード４経由で検証ノード３にアクセスしてもよい。その場合、クライアントノード４が複数のＢＣネットワーク参加者の鍵情報を管理していてもよい。

【０１２８】

また各実施例では、評価実行トランザクションと本番実行トランザクションとが別種のトランザクションとして存在する例を示した。しかし、本番実行トランザクションの一オプションとして評価実行トランザクションを実現してもよい。例えば、本番実行トランザクションの拡張領域の情報として評価ＩＤと期待値など、すなわち評価実行かどうかを識別する情報と評価実行の入力情報を持たせることでも本発明を実現可能である。

【０１２９】

また各実施例では、ＳＣ評価実行を行う各種機構をＢＣ基盤側の機能として実現した例を示したが、ＳＣ側の機能（内部機能あるいはＳＣの共通部品）として作りこむこともできる。例えば、特定の識別子を持つトランザクション（例えば利用者情報の先頭が”ｔｅｓｔ＿”で始まる等）を評価実行トランザクションとして扱う等の実現方法が考えられる。しかし、この実現方法では、ＢＣ基盤側の機能によって実現する場合に比べて確実性が損なわれる。具体的には、提供者がこの機能を実装・提供しなければ、利用者が信用性を確認できない。また、提供者がこの識別子に応じて内部ロジックを切り替えている場合には検知できない。さらに、本番用と評価用のデータ空間を分けて管理することが難しくなる。

【０１３０】

以上、本発明を実施するための最良の形態などについて具体的に説明したが、本発明はこれに限定されるものではなく、その要旨を逸脱しない範囲で種々変更可能である。

【０１３１】

こうした本実施形態によれば、ＳＣの利用者（利用予定者含む）は、中央集権機関による管理なしにＳＣの信用性を確認／評価できる。ＳＣの信用性をいつでも誰でも確認／評

価できる仕組みを提供することで、悪意のある提供者による質の悪いＳＣ提供を抑止する効果がある。ＳＣの提供者と複数の第三者による試行／評価により、特定の中央機関が集中的かつ一方的に行っていた従来の評価方法と比べて、第三者視点が含まれるため信用性を高めることができ、その信用性に関する情報の共有により第三者が個々に検証する場合に比べて手間を削減できる。複数の第三者によるテスト／評価結果を用いることで、ＳＣの信用度評価をより客観的に行うことができる。

【０１３２】

すなわち、中央集権機関による管理が無くとも、スマートコントラクト提供者以外の複数の第三者によってスマートコントラクト自体の品質の信用性を確認、評価することが可能となる。

【０１３３】

本明細書の記載により、少なくとも次のことが明らかにされる。すなわち、本実施形態の信用度管理方法において、分散台帳システムが、所定ノードが指定した評価用の所定値と、前記所定値を前記スマートコントラクトに入力した場合の出力値とを含む、前記スマートコントラクトの検証結果を、前記評価実行トランザクションに含めて管理する、としてもよい。

【０１３４】

これによれば、スマートコントラクトの評価結果として入力と出力の因果関係も含めて検証することが可能となる。

【０１３５】

また、本実施形態の信用度管理方法において、分散台帳システムが、ブロックチェーンに含まれる所定の評価実行トランザクションを基準として、前記スマートコントラクトの実行トランザクションおよびその他の評価実行トランザクションの少なくともいずれかに関する信用度を所定アルゴリズムで算定する処理を更に実行する、としてもよい。

【０１３６】

これによれば、例えば、スマートコントラクトの適用条件が、ノード間で異なるといった不平等、不誠実な内容であるケース等について、トランザクションを照合し比較することで特定し、スマートコントラクト自体の信用度を明らかなものとできる。

【０１３７】

また、本実施形態の信用度管理方法において、分散台帳システムが、前記算定した信用度の高さに応じて、前記スマートコントラクトの実行可否を制御する処理を更に実行する、としてもよい。

【０１３８】

これによれば、信用度が所定基準より低いスマートコントラクトについては、その実行を許可しないといった制御が可能となる。

【０１３９】

また、本実施形態の信用度管理方法において、分散台帳システムが、前記スマートコントラクトの提供者および利用者の少なくともいずれかによる、ブロックチェーンへの前記評価実行トランザクションの登録有無に応じて、前記スマートコントラクトの実行可否を制御する処理を更に実行する、としてもよい。

【０１４０】

これによれば、評価実行トランザクションの登録が無い、すなわち何ら評価、検証がなされていないスマートコントラクトの実行を制限することが可能となる。

【０１４１】

また、本実施形態の信用度管理方法において、分散台帳システムが、分散台帳において、実行トランザクションおよび評価実行トランザクションのそれぞれに関する所定情報を、異なるデータ領域で管理する、としてもよい。

【０１４２】

これによれば、実行トランザクションおよび評価実行トランザクションが膨大な数にのぼるブロックチェーンであっても、例えば、スマートコントラクトに関する最新の検証結

果を参照する際、評価実行トランザクションのデータ領域のみを参照すればよいことになり、処理効率が向上する。

【０１４３】

また、本実施形態の信用度管理方法において、分散台帳システムが、前記評価実行トランザクションを所定タイミングで自動発行して、前記スマートコントラクトの検証を実行する、としてもよい。

【０１４４】

これによれば、スマートコントラクトの提供者にとって想定外の、いわゆる抜き打ちでの評価実行トランザクションを実行することが可能となり、ひいては、スマートコントラクトの不適切な変更等に対する抑止効果を奏することになる。

【０１４５】

また、本実施形態の信用度管理方法において、分散台帳システムが、所定期間内の前記実行トランザクションと前記評価実行トランザクションのうち、前記実行トランザクションを優先的に処理する、としてもよい。

【０１４６】

これによれば、分散台帳システムに含まれる或るノードが、評価実行トランザクションが不必要に多く発行した場合など、実行トランザクションの処理が評価実行トランザクションの処理に阻害されかねない状況を的確に回避することが出来る。

【０１４７】

本実施形態の信用度管理システムにおいて、前記ノードそれぞれが、所定ノードが指定した評価用の所定値と、前記所定値を前記スマートコントラクトに入力した場合の出力値とを含む、前記スマートコントラクトの検証結果を、前記評価実行トランザクションに含めて管理するものである、としてもよい。

【０１４８】

本実施形態の信用度管理システムにおいて、前記ノードの少なくともいずれかが、ブロックチェーンに含まれる所定の評価実行トランザクションを基準として、前記スマートコントラクトの実行トランザクションおよびその他の評価実行トランザクションの少なくともいずれかに関する信用度を所定アルゴリズムで算定する処理を更に実行するものである、としてもよい。

【０１４９】

本実施形態の信用度管理システムにおいて、前記ノードの少なくともいずれかが、前記算定した信用度の高さに応じて、前記スマートコントラクトの実行可否を制御する処理を更に実行するものである、としてもよい。

【０１５０】

本実施形態の信用度管理システムにおいて、前記ノードの少なくともいずれかが、前記スマートコントラクトの提供者および利用者の少なくともいずれかによる、ブロックチェーンへの前記評価実行トランザクションの登録有無に応じて、前記スマートコントラクトの実行可否を制御する処理を更に実行するものである、としてもよい。

【０１５１】

本実施形態の信用度管理システムにおいて、前記ノードそれぞれが、分散台帳において、実行トランザクションおよび評価実行トランザクションのそれぞれに関する所定情報を、異なるデータ領域で管理するものである、としてもよい。

【０１５２】

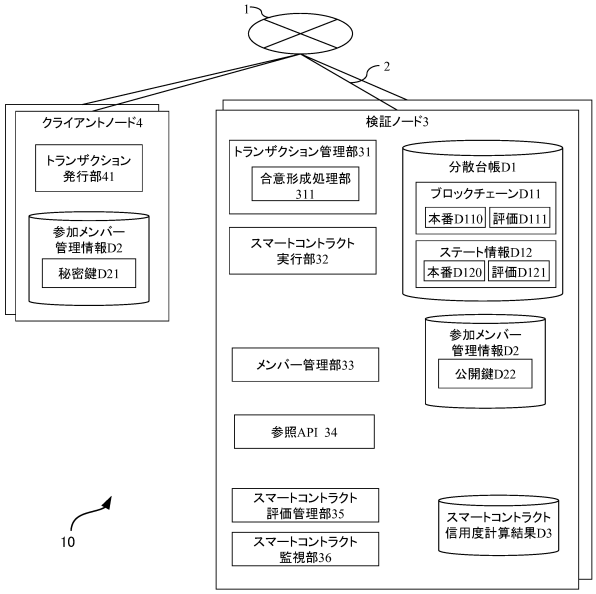
本実施形態の信用度管理システムにおいて、前記ノードの少なくともいずれかが、前記評価実行トランザクションを所定タイミングで自動発行して、前記スマートコントラクトの検証を実行するものである、としてもよい。

【０１５３】

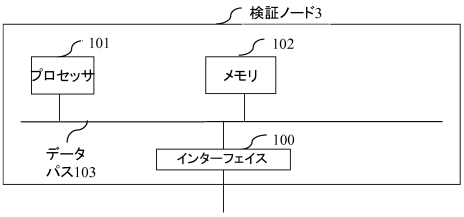
本実施形態の信用度管理システムにおいて、前記ノードの少なくともいずれかが、所定期間内の前記実行トランザクションと前記評価実行トランザクションのうち、前記実行トランザクションを優先的に処理するものである、としてもよい。

【符号の説明】
【０１５４】
１　ネットワーク
２　物理的な通信回線
３　検証ノード
１０　分散台帳システム（信用度管理システム）
１００　インターフェイス
１０１　プロセッサ
１０２　メモリ
１０３　データバス
３１　ランザクション管理部
３１０　合意形成処理部
３２　スマートコントラクト実行部（ＳＣ実行部）
３３　メンバー管理部
３４　参照ＡＰＩ
３５　スマートコントラクト評価管理部（ＳＣ評価管理部）
３６　スマートコントラクト監視部（ＳＣ監視部）
４　クライアントノード
４１　トランザクション発行部
Ｄ１　分散台帳
Ｄ１　ブロックチェーン（ＢＣ）
Ｄ１２　ステート情報
Ｄ２　参加メンバー管理情報
Ｄ２１　秘密鍵
Ｄ２２　公開鍵
Ｄ３　スマートコントラクト信用度計算結果（ＳＣ信用度計算結果）

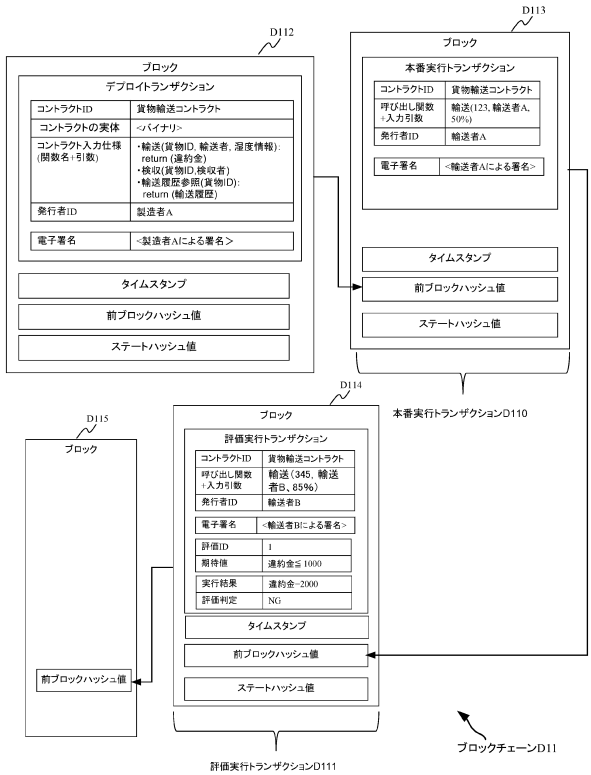
【図１】


1
2
クライアントノード4
トランザクション発行部41
参加メンバー管理情報D2
秘密鍵D21
検証ノード3
トランザクション管理部31
合意形成処理部311
スマートコントラクト実行部32
メンバー管理部33
参照API 34
スマートコントラクト評価管理部35
スマートコントラクト監視部36
分散台帳D1
ブロックチェーンD11
本番D110
評価D111
ステート情報D12
本番D120
評価D121
参加メンバー管理情報D2
公開鍵D22
スマートコントラクト信用度計算結果D3
10

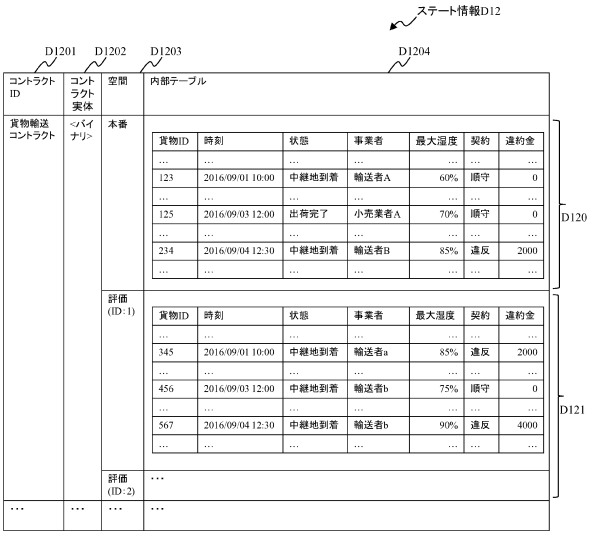
【図２】


検証ノード3
101
プロセッサ
102
メモリ
データバス103
100
インターフェイス

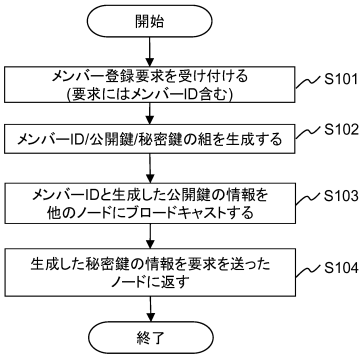
【図３】


D112
ブロック
デプロイトランザクション
コントラクトID
貨物輸送コントラクト
コントラクトの実体
<バイナリ>
コントラクト入力仕様(関数名+引数)
・輸送(貨物ID, 輸送者, 湿度情報): return (違約金)
・検収(貨物ID,検収者)
・輸送履歴参照(貨物ID): return (輸送履歴)
発行者ID
製造者A
電子署名
<製造者Aによる署名>
タイムスタンプ
前ブロックハッシュ値
ステートハッシュ値
D113
ブロック
本番実行トランザクション
コントラクトID
貨物輸送コントラクト
呼び出し関数+入力引数
輸送(123, 輸送者A, 50%)
発行者ID
輸送者A
電子署名
<輸送者Aによる署名>
タイムスタンプ
前ブロックハッシュ値
ステートハッシュ値
本番実行トランザクションD110
D114
ブロック
評価実行トランザクション
コントラクトID
貨物輸送コントラクト
呼び出し関数+入力引数
輸送（345, 輸送者B, 85%）
発行者ID
輸送者B
電子署名
<輸送者Bによる署名>
評価ID
1
期待値
違約金≦ 1000
実行結果
違約金=2000
評価判定
NG
タイムスタンプ
前ブロックハッシュ値
ステートハッシュ値
評価実行トランザクションD111
D115
ブロック
前ブロックハッシュ値
ブロックチェーンD11

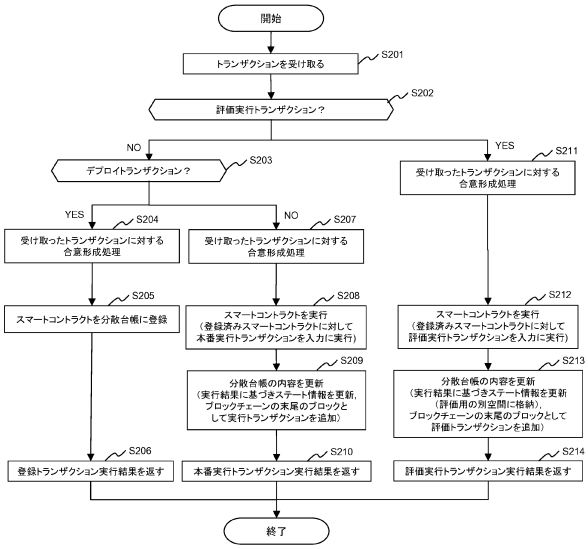
【図４】


ステート情報D12
D1201
D1202
D1203
D1204
コントラクトID
コントラクト実体
空間
内部テーブル
貨物輸送コントラクト
<バイナリ>
本番
貨物ID | 時刻 | 状態 | 事業者 | 最大湿度 | 契約 | 違約金
… | … | … | … | … | … | …
123 | 2016/09/01 10:00 | 中継地到着 | 輸送者A | 60% | 順守 | 0
… | … | … | … | … | … | …
125 | 2016/09/03 12:00 | 出荷完了 | 小売業者A | 70% | 順守 | 0
… | … | … | … | … | … | …
234 | 2016/09/04 12:30 | 中継地到着 | 輸送者B | 85% | 違反 | 2000
… | … | … | … | … | … | …
D120
評価(ID:1)
貨物ID | 時刻 | 状態 | 事業者 | 最大湿度 | 契約 | 違約金
… | … | … | … | … | … | …
345 | 2016/09/01 10:00 | 中継地到着 | 輸送者a | 85% | 違反 | 2000
… | … | … | … | … | … | …
456 | 2016/09/03 12:00 | 中継地到着 | 輸送者b | 75% | 順守 | 0
… | … | … | … | … | … | …
567 | 2016/09/04 12:30 | 中継地到着 | 輸送者b | 90% | 違反 | 4000
… | … | … | … | … | … | …
D121
評価(ID:2)
…
…

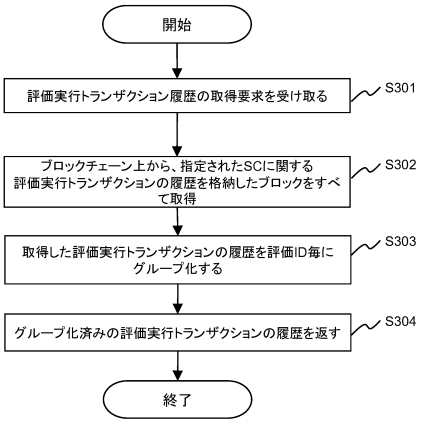
【図５】



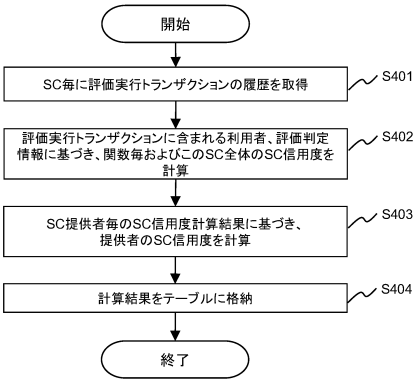
【図６】



【図７】



【図８】

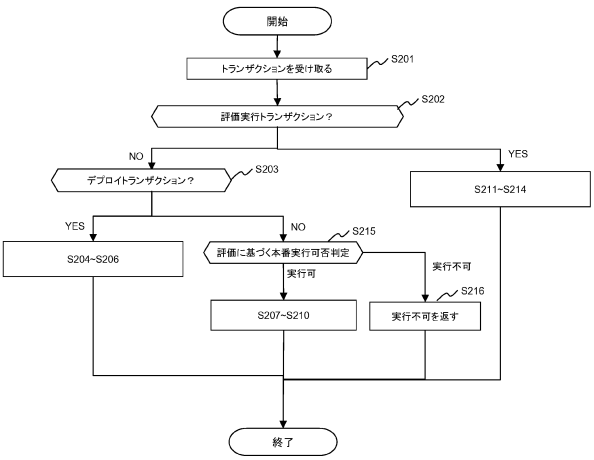


【図９】

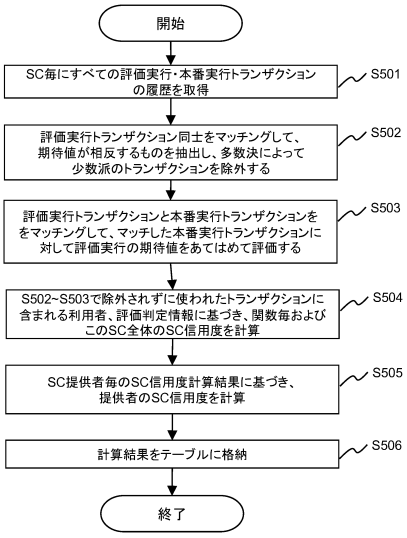
スマートコントラクト信用度 D3

D301 提供者	D302 コントラクトID	D303 関数名	D304 信用度	D305 評価者	
…	…	…	…	…	
製造業A	貨物輸送コントラクト	輸送	0.88	輸送者A, 輸送者B, 製造業A	D311
…	…	…	…	…	
製造業A	貨物輸送コントラクト	-	0.85	(略)	D312
…	…	…	…	…	
製造業A	-	-	0.83	(略)	D313
…	…	…	…	…	

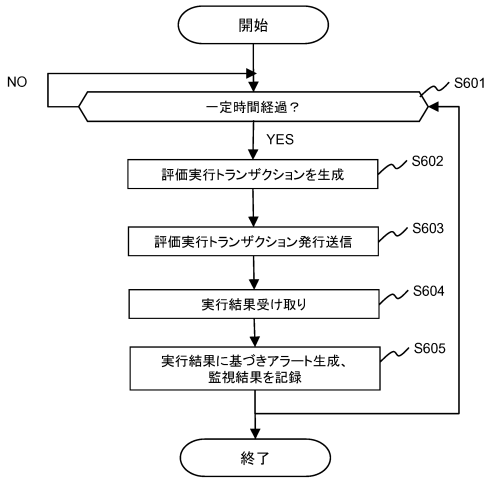
【図１０】



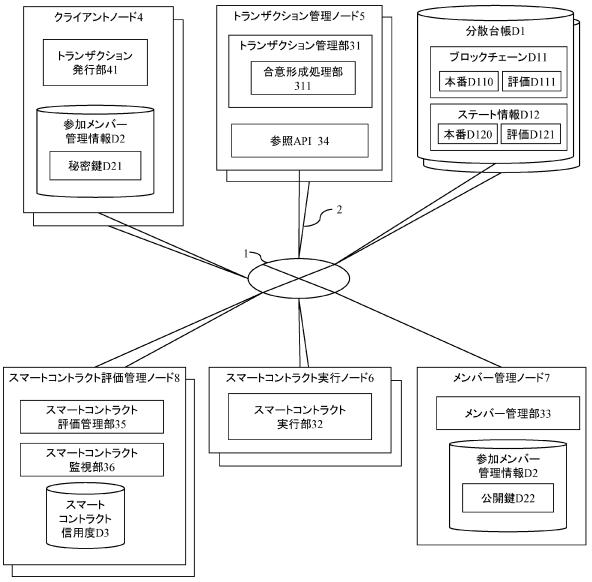
【図１１】



【図１２】



【図１３】



フロントページの続き

(56)参考文献 特開２０１６－１７０５３０（ＪＰ，Ａ）
米国特許出願公開第２０１５／０３３２２８３（ＵＳ，Ａ１）
淵田 康之，イノベーションと金融，野村資本市場クォータリー ２０１５年秋号 第１９巻第２号，日本，株式会社野村資本市場研究所，2015年11月01日，第19巻,第2号，p.11-35，ISSN 2185-4629

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｑ １０／００－９９／００