

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2020-197873

(P2020-197873A)

(43) 公開日 令和2年12月10日(2020.12.10)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 16/182 (2019.01)	G06F 16/182	
G06F 16/176 (2019.01)	G06F 16/176	
G06F 9/455 (2006.01)	G06F 9/455	150
G06F 9/50 (2006.01)	G06F 9/50	150A

審査請求 未請求 請求項の数 15 O L (全 25 頁)

(21) 出願番号	特願2019-103187 (P2019-103187)	(71) 出願人	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
(22) 出願日	令和1年5月31日(2019.5.31)	(74) 代理人	110000176 一色国際特許業務法人
		(72) 発明者	小澤 洋司 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

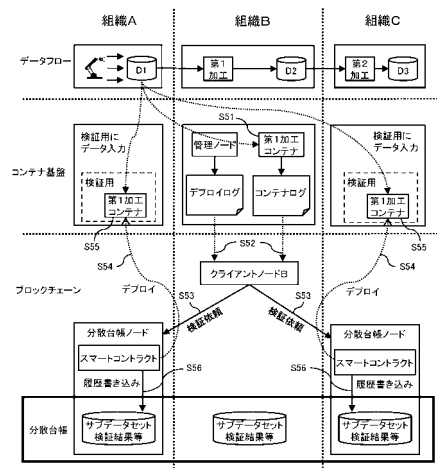
(54) 【発明の名称】 情報処理システム、及び情報処理システムの制御方法

(57) 【要約】

【課題】複数の組織で取り扱われるデータの加工内容を検証する仕組みを効率よく実現してデータの信頼性を向上する。

【解決手段】情報処理システムは、複数の組織の夫々に設けられ、トランザクションの内容を互いに検証し、トランザクションの履歴を夫々が有する分散台帳に保持する複数の分散台帳ノードと、分散台帳ノードにトランザクションを送信するクライアントノードと、複数の組織の夫々に設けられ、検証対象処理を仮想化基盤において実行する処理ノードと、を含む。クライアントノードは、検証を行う複数の分散台帳ノードを選択し、選択した分散台帳ノードの夫々に、検証対象処理の実行要求を含むトランザクションを送信し、分散台帳ノードの夫々は、夫々が属する組織の処理ノードにおいて、検証対象処理を仮想化基盤で実行して検証対象処理の検証を行い、検証の結果を分散台帳に記録する。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

複数の組織の夫々に設けられ、トランザクションの内容を互いに検証し、前記トランザクションの履歴を夫々が有する分散台帳に保持する複数の分散台帳ノードと、

前記分散台帳ノードにトランザクションを送信するクライアントノードと、

前記複数の組織の夫々に設けられ、検証の対象となる処理である検証対象処理を仮想化基盤において実行する処理ノードと、

を含み、

前記クライアントノードは、前記検証を行う複数の前記分散台帳ノードを選択し、選択した前記分散台帳ノードの夫々に、前記検証対象処理の実行要求を含むトランザクションを送信し、

前記分散台帳ノードの夫々は、前記トランザクションを受信すると、夫々が属する前記組織の前記処理ノードにおいて、前記検証対象処理を前記仮想化基盤で実行して前記検証対象処理の検証を行い、前記検証の結果を前記分散台帳に記録する、

情報処理システム。

10

【請求項 2】

請求項 1 に記載の情報処理システムであって、

前記検証対象処理は、複数の前記組織の間でデータフローに従って行われるデータの加工処理である、

情報処理システム。

20

【請求項 3】

請求項 2 に記載の情報処理システムであって、

前記加工処理に用いるデータを提供するデータ格納ノードを更に含み、

前記分散台帳ノードは、前記データ格納ノードから提供される前記データを前記検証対象処理に入力する、

情報処理システム。

【請求項 4】

請求項 3 に記載の情報処理システムであって、

前記データ格納ノードは、提供する前記データについて前記組織ごとにアクセス制限を行い、

前記クライアントノードは、前記検証対象処理の実行要求を含むトランザクションの送信先として、前記検証対象処理に入力する前記データにアクセス可能な前記組織の前記分散台帳ノードを選択する、

情報処理システム。

30

【請求項 5】

請求項 3 に記載の情報処理システムであって、

前記クライアントノードは、前記検証対象処理の実行要求を含むトランザクションの送信先として、前記検証対象処理のデプロイを行う第 1 の前記分散台帳ノードを選択するとともに、前記データ格納ノードから提供される前記データの検証対象処理への入力を行う第 2 の前記分散台帳ノードを選択する、

情報処理システム。

40

【請求項 6】

請求項 2 に記載の情報処理システムであって、

前記クライアントノードは、前記検証対象処理の実行要求を含むトランザクションの送信先として、前記データフローにおいて前記検証対象処理の加工後のデータを利用している他の前記組織の分散台帳ノードを選択する、

情報処理システム。

【請求項 7】

請求項 5 に記載の情報処理システムであって、

前記データ格納ノードは、提供する前記データについて前記組織ごとにアクセス制限を

50

行い、

前記クライアントノードは、前記第2の前記分散台帳ノードとして、前記検証対象処理に入力する前記データにアクセス可能な前記組織の前記分散台帳ノードを選択する、
情報処理システム。

【請求項8】

請求項1に記載の情報処理システムであって、

前記クライアントノードは、前記検証対象処理の実行要求を含むトランザクションの送信先として選択した送信先の数を、検証の質を表す評価指標である検証実施レベルとして前記トランザクションに設定し、

前記分散台帳ノードは、前記検証により前記検証対象処理が正しいと判定した場合、前記検証実施レベルを分散台帳に記録する、
情報処理システム。

【請求項9】

請求項1に記載の情報処理システムであって、

前記分散台帳は、前記組織の夫々が保有しているポイント値の記録を含み、

前記分散台帳ノードは、自身が属する前記組織の前記処理ノードに前記検証対象処理がデプロイされた際、前記分散台帳における当該組織の前記ポイント値を増加させる、
情報処理システム。

【請求項10】

請求項9に記載の情報処理システムであって、

前記クライアントノードは、

前記組織ごとの検証実行のために必要なポイント値を記憶し、

前記検証対象処理の実行要求を含むトランザクションの送信先として、前記必要なポイント値が少ない前記組織の前記分散台帳ノードを優先して選択する、
情報処理システム。

【請求項11】

請求項1に記載の情報処理システムであって、

前記処理ノードは、前記仮想化基盤における前記検証対象処理の再現手順が記録されたイメージに基づきデプロイされた前記検証対象処理を実行する、
情報処理システム。

【請求項12】

複数の組織の夫々に設けられ、トランザクションの内容を互いに検証し、前記トランザクションの履歴を夫々が有する分散台帳に保持する複数の分散台帳ノードと、

前記分散台帳ノードにトランザクションを送信するクライアントノードと、

前記複数の組織の夫々に設けられ、検証の対象となる処理である検証対象処理を仮想化基盤において実行する処理ノードと、

を含んで構成される情報処理システムの制御方法であって、

前記クライアントノードが、前記検証を行う複数の前記分散台帳ノードを選択し、選択した前記分散台帳ノードの夫々に、前記検証対象処理の実行要求を含むトランザクションを送信するステップと、

前記分散台帳ノードの夫々が、前記トランザクションを受信すると、夫々が属する前記組織の前記処理ノードにおいて、前記検証対象処理を前記仮想化基盤で実行して前記検証対象処理の検証を行い、前記検証の結果を前記分散台帳に記録するステップと、

を含む、情報処理システムの制御方法。

【請求項13】

請求項12に記載の情報処理システムの制御方法であって、

前記検証対象処理は、複数の前記組織の間でデータフローに従って行われるデータの加工処理であり、

前記処理ノードが、前記仮想化基盤における前記検証対象処理の再現手順が記録されたイメージに基づきデプロイされた前記検証対象処理を実行するステップを更に含む、

10

20

30

40

50

情報処理システムの制御方法。

【請求項 14】

請求項 13 に記載の情報処理システムの制御方法であって、

前記情報処理システムは、前記加工処理に用いるデータを提供するデータ格納ノードを更に含み、

前記分散台帳ノードが、前記データ格納ノードから提供される前記データを前記検証対象処理に入力するステップを更に含む、

情報処理システムの制御方法。

【請求項 15】

請求項 14 に記載の情報処理システムの制御方法であって、

前記データ格納ノードが、提供する前記データについて前記組織ごとにアクセス制限を行うステップと、

前記クライアントノードが、前記検証対象処理の実行要求を含むトランザクションの送信先として、前記検証対象処理に入力する前記データにアクセス可能な前記組織の前記分散台帳ノードを選択するステップと、

を更に含む、情報処理システムの制御方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、情報処理システム、及び情報処理システムの制御方法に関する。

【背景技術】

【0002】

近年、センサ等から収集される I o T データの活用が加速している。とくに産業分野においては、工場の設備やセンサ等から収集した I o T データについての組織や企業を跨いだ利活用が進んでいる。

【0003】

一般に工場の設備やセンサ等から収集した I o T データは、設備やセンサ等から収集した生のデータがそのまま利用されることは少なく、利用目的に応じて加工した上で用いられる。またある組織で加工されたデータを他の組織で利用する場合、他の組織において更なる加工が行われる。

【0004】

昨今、製造現場におけるデータの改ざんが発生しており、データ自体が改ざんされていないことを保証することが課題となっている。こうした課題に関し、特許文献 1 には、電子ファイルの変更処理（加工）イベントをブロックチェーンに格納して共有することによりリアルタイムに処理ステータスを共有し、変更履歴の信頼性を確保することが記載されている。

【先行技術文献】

【特許文献】

【0005】

【特許文献 1】米国特許出願公開第 2017 / 0295232 号明細書

【発明の概要】

【発明が解決しようとする課題】

【0006】

ところで、特許文献 1 の技術では、データを加工する組織がブロックチェーンに加工履歴の登録を行うため、ブロックチェーンに登録する前に加工履歴を改ざんすることは可能であり、加工履歴の信頼性を必ずしも保証することはできない。

【0007】

また例えば、コンソーシアム型ブロックチェーンの考え方にに基づき、加工履歴をコンソーシアム内の組織で互いに検証しあうようにした場合、検証のために消費するコンピューティングリソースが多いことが課題となる。単純にコンソーシアム内で互いに検証する構

10

20

30

40

50

成とした場合、全ての組織で同じ加工処理を実施して処理結果を検証することとなり、膨大なIoTデータを処理するために多大なコンピューティングリソースが必要になる。またブロックチェーンのスマートコントラクトで実行できる処理は限られている上、既存の様々な加工ツールで行っている加工処理をスマートコントラクトとして改めて実装し直すことは必ずしも現実的でない。

【0008】

本発明はこのような背景に基づきなされたものであり、複数の組織で取り扱われるデータの加工内容を検証する仕組みを効率よく実現してデータの信頼性を向上することが可能な、情報処理システム、及び情報処理システムの制御方法を提供することを目的とする。

【課題を解決するための手段】

【0009】

上記の目的を達成するための本発明の一つは、情報処理システムであって、複数の組織の夫々に設けられ、トランザクションの内容を互いに検証し、前記トランザクションの履歴を夫々が有する分散台帳に保持する複数の分散台帳ノードと、前記分散台帳ノードにトランザクションを送信するクライアントノードと、前記複数の組織の夫々に設けられ、検証の対象となる処理である検証対象処理を仮想化基盤において実行する処理ノードと、を含み、前記クライアントノードは、前記検証を行う複数の前記分散台帳ノードを選択し、選択した前記分散台帳ノードの夫々に、前記検証対象処理の実行要求を含むトランザクションを送信し、前記分散台帳ノードの夫々は、前記トランザクションを受信すると、夫々が属する前記組織の前記処理ノードにおいて、前記検証対象処理を前記仮想化基盤で実行して前記検証対象処理の検証を行い、前記検証の結果を前記分散台帳に記録する。

【0010】

その他、本願が開示する課題、及びその解決方法は、発明を実施するための形態の欄、及び図面により明らかにされる。

【発明の効果】

【0011】

本発明によれば、複数の組織で取り扱われるデータの加工内容を検証する仕組みを効率よく実現してデータの信頼性を向上することができる。

【図面の簡単な説明】

【0012】

【図1】データ加工履歴管理システムの仕組みを説明する図である。

【図2】データ加工履歴管理システムの概略的な構成を説明するブロック図である。

【図3】データ加工履歴管理システムを構成する情報処理装置の一例を示すブロック図である。

【図4】データフロー管理ノードが備える機能を説明するブロック図である。

【図5】管理ノードが備える機能を説明するブロック図である。

【図6】処理ノードの構成を説明するブロック図である。

【図7】データ格納ノードの構成を説明するブロック図である。

【図8】クライアントノードの構成を説明するブロック図である。

【図9】分散台帳ノードの構成を説明するブロック図である。

【図10】コンテナレジストリの構成を説明するブロック図である。

【図11】データフロー情報の一例である。

【図12】デプロイログの一例である。

【図13】コンテナ情報の一例である。

【図14】コンテナログの一例である。

【図15】データセットの一例である。

【図16】データセットアクセス権の一例である。

【図17】検証先決定用処理ノード情報の一例である。

【図18】検証先決定用パラメータ情報の一例である。

【図19】組織情報の一例である。

10

20

30

40

50

- 【図 2 0】サブデータセット検証結果の一例である。
- 【図 2 1】組織保持ポイント情報の一例である。
- 【図 2 2】コンテナログの一例である。
- 【図 2 3 A】ブロックチェーンの一例である。
- 【図 2 3 B】ブロックチェーンの一例（図 2 3 A の続き）である。
- 【図 2 4】スマートコントラクト情報の一例である。
- 【図 2 5】出力データの一例である。
- 【図 2 6】コンテナイメージの一例である。
- 【図 2 7】ユーザがデータフローを追加する際にデータ加工履歴管理システムにおいて行われる処理を説明するシーケンス図である。
- 【図 2 8】データフローに従ってある組織でデータの加工処理が行われた際にデータ加工履歴管理システムにおいて行われる加工内容の検証処理に関する準備処理を説明するシーケンス図である。
- 【図 2 9】検証先ノード選択処理の詳細を説明するフローチャートである。
- 【図 3 0】ある組織でデータの加工処理が行われた際にデータ加工履歴管理システムにおいて行われる、上記のある組織におけるデータの加工内容の検証処理を説明するシーケンス図である。

10

20

30

40

50

【発明を実施するための形態】

【0013】

以下、実施形態について図面を参照しつつ説明する。以下の説明において、同一のまたは類似する構成に同一の符号を付して重複した説明を省略することがある。また以下の説明において、同種の構成を区別する必要がある場合、構成を総称する符号の後に括弧書きで識別子（数字、アルファベット等）を表記することがある。また以下の説明において、「ブロックチェーン」のことを「BC」と略記することがある。

【0014】

図 1 は、実施形態として示す情報処理システムであるデータ加工履歴管理システム 1 の仕組みを説明する図である。データ加工履歴管理システム 1 は、複数の組織の間でデータフローに従ってデータの提供や加工が行われる場合に、各組織において用いられる上記データの検証をコンソーシアム型のブロックチェーンの仕組みを用いて行い、上記データフローで用いられるデータに対して行われる不正や改ざんを防ぐ。

【0015】

同図には、上記データを利用する 3 つの組織（以下、「組織 A」、「組織 B」、「組織 C」とする。）においてデータフローに従ってデータの提供や加工が行われる場合を例示している。同図において、点線で区切った各段のうち上段にはデータフローを、中段にはデータフローを実行する仮想化基盤であるコンテナ基盤を、下段にはデータの加工履歴を管理するブロックチェーンの構成を、夫々示している。

【0016】

上段に示すように、例示するデータフローでは、組織 A においてデータ源 50 から取得されたデータ D 1 が組織 B に提供され、組織 B においてデータ D 1 について第 1 加工処理が行われてデータ D 2 が生成され、生成されたデータ D 2 が組織 C に提供され、組織 C においてデータ D 2 について第 2 加工処理が行われてデータ D 3 が生成される。

【0017】

中段に示すように、第 1 加工処理は組織 B のコンテナ基盤にデプロイされたコンテナ（以下、「第 1 加工コンテナ」と称する）で実行され、第 2 加工処理は組織 C のコンテナ基盤にデプロイされたコンテナ（以下、「第 2 加工コンテナ」と称する。）で実行される（S51）。

【0018】

ここで組織 B において実行される第 1 加工処理に着目すると、第 1 加工コンテナは、データ D 1 を加工した際、当該第 1 加工コンテナのデプロイログと当該第 1 加工コンテナを実行することにより生成されるコンテナログを組織 B のクライアントノード B に送信する

(S 5 2)。

【 0 0 1 9 】

組織 B のクライアントノードは、第 1 加工コンテナからデプロイログとコンテナログを受信すると、他の組織である組織 A 及び組織 C の夫々の分散台帳ノードに第 1 加工処理の検証依頼を送信する (S 5 3)。

【 0 0 2 0 】

組織 A 及び組織 C の夫々の分散台帳ノードは、上記検証依頼を受信すると、夫々のスマートコントラクトにより夫々が属する組織のコンテナ基盤に検証用の第 1 加工コンテナをデプロイし (S 5 4)、デプロイした第 1 加工コンテナの夫々にデータ D 1 を入力し、夫々の第 1 加工コンテナがデータ D 1 について行った処理 (再現処理) の結果に基づき、データ D 1 の加工内容を検証する (S 5 5)。

10

【 0 0 2 1 】

上記検証に成功すると、組織 A 及び組織 C の夫々の分散台帳ノードは、夫々の分散台帳のサブデータセット検証結果に検証結果を書き込む (S 5 6)。

【 0 0 2 2 】

図 2 は、データ加工履歴管理システム 1 の概略的な構成を説明するブロック図である。同図に示すように、データ加工履歴管理システム 1 は、ユーザ端末 3 0、データフロー管理ノード 4 0、データ源 5 0、コンテナレジストリ 6 0、及び複数の組織の夫々に存在するノード群 (管理ノード 1 1、処理ノード 1 2、データ格納ノード 1 3、クライアントノード 2 1、分散台帳ノード 2 2) を含む。これらはいずれも情報処理装置 (コンピュータ) を用いて構成され、またこれらは通信ネットワーク 5 を介して通信可能に接続されている。尚、通信ネットワーク 5 は、複数のネットワークで構成されていてもよい。例えば、コンテナ基盤 1 0 とデータ源 5 0 を接続する通信ネットワークとクライアントノード 2 1 と分散台帳ノード 2 2 を接続する通信ネットワークは別であってもよい。

20

【 0 0 2 3 】

コンテナ基盤 1 0 は、管理ノード 1 1、処理ノード 1 2、及びデータ格納ノード 1 3 を実現する。コンテナ基盤 1 0 は、例えば、D o c k e r (登録商標) や K u b e r n e t e s (登録商標) 等により構成される。尚、コンテナ基盤はノードの実現基盤 (処理の実行環境) の一例に過ぎず、例えば、コンテナ基盤 1 0 に代えて仮想マシン (Virtual Machine) 等を実行する基盤をノードの実現基盤として用いてもよい。尚、その場合、コンテナレジストリ 6 0 は、仮想マシンのイメージのレジストリとなる。

30

【 0 0 2 4 】

尚、コンテナ基盤 1 0 を K u b e r n e t e s (登録商標) により構成した場合には、加工処理の内容をコンテナイメージとして固定することができる。またコンテナイメージのハッシュ値をブロックチェーンの分散台帳に管理した場合、加工内容が改ざんされているか否かの検証が可能になる。また K u b e r n e t e s (登録商標) はポータビリティに優れるため、例えば、検証用のコンテナ (以下、「検証用コンテナ」と称する。) を各組織のコンテナ基盤 1 0 にデプロイする際、加工処理の実行環境を各組織に容易に構築することができる。

【 0 0 2 5 】

ユーザ端末 3 0 は、ユーザがデータフローを管理するためのユーザインタフェースを提供する。ユーザ端末 3 0 は、ユーザからデータフローに関する要求を受け付け、その要求をデータフロー管理ノード 4 0 に送信する。データフロー管理ノード 4 0 は、ユーザ端末 3 0 から要求されたデータフローの管理 (新規作成、変更 (編集)、削除等) を管理ノード 1 1 に要求する。尚、データフロー管理ノード 4 0 は、ユーザ端末 3 0 と共通としても (同じ情報処理装置により構成しても) よい。

40

【 0 0 2 6 】

データ源 5 0 は、組織が取り扱うデータの生成源 (各種製造装置、各種機械、各種センサ等) であり、センサデータ (I o T データ) を生成する。

【 0 0 2 7 】

50

コンテナレジストリ 60 は、コンテナイメージを管理し、処理ノード 12 からコンテナイメージの取得要求を受信すると、要求元にコンテナイメージを提供（配信）する。

【0028】

管理ノード 11 は、コンテナ基盤 10 の管理に関する処理を行う。上記処理は、例えば、処理ノード 12 へのコンテナのデプロイに関する処理や、デプロイログの生成に関する処理等である。

【0029】

処理ノード 12 は、コンテナを実行し、コンテナとの間でデータの入出力に関する処理を行う。また処理ノード 12 は、コンテナの実行ログの生成や管理に関する処理を行う。尚、処理ノード 12 は同じ組織に複数存在していてもよい。

10

【0030】

データ格納ノード 13 は、データ源 50 から収集したデータや加工コンテナによって加工されたデータを管理（記憶）する。データ格納ノード 13 は、他のノードに自身が格納しているデータに関する情報を提供する。

【0031】

クライアントノード 21 は、ブロックチェーンに関する各種の処理を行う。クライアントノード 21 は、検証先の分散台帳ノード 22 を選択する。

【0032】

分散台帳ノード 22 は、ブロックチェーンの分散台帳を管理し、データの加工内容の検証に関する処理を行う。分散台帳ノード 22 は、同じ組織に複数存在していてもよい。

20

【0033】

図 3 は、ユーザ端末 30、データフロー管理ノード 40、コンテナレジストリ 60、及びコンテナ基盤 10 を構成する情報処理装置（以下、「情報処理装置 100」と称する。）の一例を示すブロック図である。

【0034】

同図に示すように、例示する情報処理装置 100 は、プロセッサ 101、主記憶装置 102、補助記憶装置 103、入力装置 104、出力装置 105、及び通信装置 106 を備える。尚、情報処理装置 100 は、クラウドにおけるクラウドサーバ等の仮想的な情報処理資源を用いて構成されるものであってもよい。

【0035】

30

プロセッサ 101 は、例えば、演算処理を行う装置であり、CPU (Central Processing Unit)、MPU (Micro Processing Unit)、GPU (Graphics Processing Unit)、AI (Artificial Intelligence) チップ等である。主記憶装置 102 は、プログラムやデータを記憶する装置であり、例えば、ROM (Read Only Memory) (SRAM (Static Random Access Memory)、NVRAM (Non Volatile RAM)、マスク ROM (Mask Read Only Memory)、PROM (Programmable ROM) 等)、RAM (Random Access Memory) (DRAM (Dynamic Random Access Memory) 等) 等である。補助記憶装置 103 は、ハードディスクドライブ (Hard Disk Drive)、フラッシュメモリ (Flash Memory)、SSD (Solid State Drive)、光学式記憶装置 (CD (Compact Disc)、DVD (Digital Versatile Disc) 等) 等である。補助記憶装置 103 に格納されているプログラムやデータは、主記憶装置 102 に随時読み込まれる。

40

【0036】

入力装置 104 は、ユーザから情報を受付けるユーザインタフェースであり、例えば、キーボード、マウス、カードリーダー、タッチパネル等である。出力装置 105 は、各種の情報を出力（表示出力、音声出力、印字出力等）するユーザインタフェースであり、例えば、各種情報を可視化する表示装置 (LCD (Liquid Crystal Display)、グラフィックカード等) や音声出力装置 (スピーカ)、印字装置等である。

【0037】

通信装置 106 は、通信ネットワーク 5 を介して他の装置と通信する通信インタフェースであり、例えば、NIC (Network Interface Card)、無線通信モジュール、USB (

50

Universal Serial Interface) モジュール、シリアル通信モジュール等である。通信装置 106 は、通信可能に接続する他の装置から情報を受信する入力装置として機能することもできる。また通信装置 106 は、通信可能に接続する他の装置に情報を送信する出力装置として機能することもできる。

【0038】

ユーザ端末 30、データフロー管理ノード 40、コンテナレジストリ 60、及びコンテナ基盤 10 により実現される各ノード (管理ノード 11、処理ノード 12、データ格納ノード 13、クライアントノード 21、分散台帳ノード 22) が備える機能は、プロセッサ 101 が、主記憶装置 102 に格納されているプログラムを読み出して実行することにより、もしくは、情報処理装置 100 を構成しているハードウェア (FPGA、ASIC、AI チップ等) により実現される。情報処理装置 100 は、基本的な機能として、例えば、オペレーティングシステム、デバイスドライバ、ファイルシステム、DBMS (Database Management System) 等の機能を備えていてもよい。

10

【0039】

図 4 は、データフロー管理ノード 40 が備える機能を説明するブロック図である。データフロー管理ノード 40 は、記憶部 411 及びデータフロー管理部 412 の各機能を有する。

【0040】

上記機能のうち、記憶部 402 は、データフロー情報 451 を記憶する。データフロー管理部 412 は、データフローの管理や編集 (追加、変更、削除等) に関する処理要求を管理ノード 11 に要求する。データフロー情報 451 は、データフローに関する各種の情報を含む。

20

【0041】

図 5 は、管理ノード 11 が備える機能を説明するブロック図である。管理ノード 11 は、記憶部 111、及びコンテナデプロイ部 112 を備える。記憶部 111 は、デプロイログ 151 を記憶する。デプロイログ 151 は、コンテナのデプロイに関するログ情報を保持する。コンテナデプロイ部 112 は、処理ノード 12 へのコンテナのデプロイに関する処理を行う。尚、管理ノード 11 の全部又は一部の機能を処理ノード 12 により実現してもよい。

【0042】

図 6 は、処理ノード 12 の構成を説明するブロック図である。処理ノード 12 は、記憶部 121、コンテナ実行部 122、及びコンテナログ管理部 123 を備える。

30

【0043】

記憶部 121 は、コンテナ情報 125 及びコンテナログ 126 を記憶する。コンテナ情報 125 は、処理ノード 12 にデプロイされているコンテナに関する情報を含む。コンテナログ 126 は、例えば、コンテナの標準出力の内容やデータの加工処理の実行履歴を含む。

【0044】

コンテナ実行部 122 はコンテナの実行に関する処理を行う。またコンテナ実行部 122 は、コンテナへのデータの入出力の制御に関する処理を行う。

40

【0045】

コンテナログ管理部 123 は、コンテナから出力される実行履歴をコンテナログ 126 として管理 (記憶) する。

【0046】

図 7 は、データ格納ノード 13 の構成を説明するブロック図である。データ格納ノード 13 は、記憶部 131 を備える。記憶部 131 は、データセット 135、データセットアクセス権 136、及びデータ 137 を記憶する。

【0047】

データセット 135 は、データ源 50 から取得したデータや、加工処理により加工されたデータにより構成される。データセットアクセス権 136 は、組織のデータセットに対

50

するアクセス権に関する情報を含む。データ 137 は、データ源 50 から収集したデータや加工処理により加工されたデータである。データの信頼性を高めるために、データ自体やデータのハッシュ値を分散台帳に記録してもよい。

【0048】

図 8 はクライアントノード 21 の構成を説明するブロック図である。クライアントノード 21 は、記憶部 211、トランザクション発行部 212、ノード選択部 213、及び業務アプリ 214 を備える。

【0049】

記憶部 211 は、検証先決定用処理ノード情報 215、検証先決定用パラメータ情報 216、及び組織情報 217 を記憶する。検証先決定用処理ノード情報 215 は、検証先の決定に際して用いられる情報であり、処理ノード 12 に関する情報を含む。検証先決定用パラメータ情報 216 は、検証先の決定に際して用いられる情報であり、各種のパラメータに関する情報を含む。組織情報 217 は、組織と分散台帳ノード 22 の対応を示す情報を含む。

10

【0050】

トランザクション発行部 212 は、ブロックチェーンの分散台帳ノード 22 にトランザクションを発行（送信）する。後述するように、トランザクション発行部 212 は、データの加工内容（加工処理の内容）の検証要求を含むトランザクションを分散台帳ノード 22 に発行する。

【0051】

ノード選択部 213 は、加工処理の検証に際して用いる分散台帳ノード 22 の選択に関する処理を行う。

20

【0052】

業務アプリ 214 は、処理ノード 12 等から加工内容に関する情報を含んだコンテナログを取得し、その内容を検証するようにトランザクション発行部 212 にトランザクションの発行を要求する。

【0053】

図 9 は分散台帳ノード 22 の構成を説明するブロック図である。分散台帳ノード 22 は、記憶部 221、スマートコントラクト実行/管理部 222、検証用コンテナデプロイ部 223、データ入力部 224、及び検証実行部 225 を備える。

30

【0054】

記憶部 221 は、現在（最新）のステート（サブデータセット検証結果 251、組織保持ポイント情報 252、コンテナログ 253）、ブロックチェーン 227、スマートコントラクト情報 228、及び出力データ 229 を記憶する。

【0055】

サブデータセット検証結果 251 は、サブデータセットごとの加工処理の検証結果を含む。サブデータセットは、データセットの構成要素であり、例えば、データセットをデータの所定の取得期間で分割したものである。

【0056】

組織保持ポイント情報 252 は、各組織が保持しているポイントを保持する。尚、ポイントは、データの加工内容の検証のためにコンピューティングリソース（処理ノード 12 のリソース等）を提供した組織に付与される特典である。またポイントは受益者負担を促し、データセットを利用した組織は利用に必要なポイントを消費する。

40

【0057】

コンテナログ 253 は、例えば、コンテナの標準出力の内容やデータの加工処理の実行履歴を含む。

【0058】

ブロックチェーン 227 は、トランザクションの実行履歴等の情報を含む。

【0059】

スマートコントラクト情報 228 は、分散台帳ノード 22 に登録されているスマートコ

50

ントラクトに関する情報を含む。

【0060】

出力データ229は、検証用コンテナがデータの加工処理を行った際の加工後のデータを含む。

【0061】

スマートコントラクト実行/管理部222は、スマートコントラクトの実行に関する処理（管理）を行う。

【0062】

検証用コンテナデプロイ部223は、スマートコントラクトから呼び出され、トランザクションに指定されている処理ノード12への検証用コンテナのデプロイに関する処理を行う。

【0063】

データ入力部224は、スマートコントラクトから呼び出され、トランザクションに含まれている、加工処理による加工前の入力データを取得し、取得した入力データを検証用コンテナに入力する。

【0064】

検証実行部225は、スマートコントラクトから呼び出され、検証用コンテナの出力データに基づき、データの加工処理が正しいか否かを検証する処理を行う。検証実行部225は、例えば、処理ノード12のコンテナログ126に含まれている値や出力データに含まれている値に基づき上記検証を行う。尚、上記検証の方法は必ずしも限定されない。

【0065】

図10は、コンテナレジストリ60の構成を説明するブロック図である。コンテナレジストリ60は、記憶部601及びコンテナイメージ管理部611を備える。

【0066】

記憶部601は、コンテナイメージ651を記憶する。コンテナイメージ管理部611は、コンテナイメージ651の登録や提供（配布、配信）に関する処理を行う。

【0067】

図11は、図4に示したデータフロー情報451の一例である。データフロー情報451は、データフローID4511、作成組織4512、作成日時4513、削除日時4514、入力データセットID4515、加工コンテナID4516、出力データセットID4517、及び状態4518の各項目を有する一つ以上のレコードで構成される。各レコードは一つのデータフローに対応している。

【0068】

上記項目のうち、データフローID4511には、データフローの識別子（以下、「データフローID」と称する。）が設定される。作成組織4512には、当該データフローの作成を要求した組織の識別子（以下、「組織ID」と称する。）が設定される。作成日時4513には、当該データフローが作成された日時が設定される。削除日時4514には、当該データフローが削除された場合に当該が行われた日時が設定される。削除されていない場合は「N/A」が設定される。

【0069】

入力データセットID4515には、当該データフローに入力されるデータセットの識別子（以下、「データセットID」と称する。）が設定される。同じデータフローに複数のデータセットが入力されることもあり、その場合、入力データセットID4515には、各データセットのデータセットIDが設定される。

【0070】

加工コンテナID4516には、当該データフローの実行に際して利用するコンテナの識別子（以下、「コンテナID」と称する。）が設定される。出力データセットID4517には、当該データフローを実行することにより生成（出力）されるデータセットのデータセットIDが設定される。状態4518には、当該データフローの現在の実行状態を示す情報が設定される。

10

20

30

40

50

【 0 0 7 1 】

図 1 2 は、図 5 に示したデプロイログ 1 5 1 の一例である。デプロイログ 1 5 1 は、デプロイログ ID 1 5 1 1、作成日時 1 5 1 2、作成コンテナ ID 1 5 1 3、及びイメージ ID 1 5 1 4 の各項目を有する一つ以上のレコードで構成される。各レコードは一つのデプロイログに対応している。

【 0 0 7 2 】

上記項目のうち、デプロイログ ID 1 5 1 1 には、デプロイログの識別子（以下、「デプロイログ ID」と称する。）が設定される。作成日時 1 5 1 2 には、当該デプロイログが生成された日時が設定される。

【 0 0 7 3 】

作成コンテナ ID 1 5 1 3 には、当該デプロイログに対応するコンテナをデプロイしたコンテナのコンテナ ID が設定される。イメージ ID 1 5 1 4 には、当該デプロイログに対応するコンテナイメージの識別子（以下、「イメージ ID」と称する。）が設定される。

10

【 0 0 7 4 】

図 1 3 は、図 6 に示したコンテナ情報 1 2 5 の一例である。コンテナ情報 1 2 5 は、コンテナ ID 1 2 5 1、イメージ ID 1 2 5 2、及び状態 1 2 5 3 の各項目を有する一つ以上のレコードで構成される。各レコードは一つのコンテナに対応している。

【 0 0 7 5 】

上記項目のうち、コンテナ ID 1 2 5 1 にはコンテナ ID が設定される。イメージ ID 1 2 5 2 には当該コンテナのコンテナイメージのイメージ ID が設定される。状態 1 2 5 3 には、当該コンテナの現在の実行状態を示す情報が設定される。

20

【 0 0 7 6 】

図 1 4 は、図 6 に示したコンテナログ 1 2 6 の一例である。コンテナログ 1 2 6 は、コンテナログ ID 1 2 6 1、コンテナ ID 1 2 6 2、及びログ情報 1 2 6 3 の各項目を有する一つ以上のレコードで構成される。各レコードは一つのコンテナログに対応している。

【 0 0 7 7 】

上記項目のうち、コンテナログ ID 1 2 6 1 には、コンテナログの識別子（以下、「コンテナログ ID」と称する。）が設定される。コンテナ ID 1 2 5 1 には、当該コンテナログを生成（出力）したコンテナのコンテナ ID が設定される。

30

【 0 0 7 8 】

ログ情報 1 2 6 3 には、コンテナから出力されたログの内容が設定される。例示するログ情報 1 2 6 3 は、入力データの値、出力データの値、出力先のサブデータセットの情報等を含む。

【 0 0 7 9 】

図 1 5 は、図 7 に示したデータセット 1 3 5 の一例である。データセット 1 3 5 は、データセット ID 1 3 5 1、サブデータセット ID 1 3 5 2、及び取得期間 1 3 5 3 の各項目を有する一つ以上のレコードで構成される。各レコードは一つのデータセットに対応している。

【 0 0 8 0 】

上記項目のうち、データセット ID 1 3 5 1 には、データセットの識別子（以下、「データセット ID」と称する。）が設定される。サブデータセット ID 1 3 5 2 には、サブデータセットの識別子（以下、「サブデータセット ID」と称する。）が設定される。取得期間 1 3 5 3 には、当該サブデータセットの取得期間が設定される。

40

【 0 0 8 1 】

図 1 6 は、図 7 に示したデータセットアクセス権 1 3 6 の一例である。データセットアクセス権 1 3 6 は、データセット ID 1 3 6 1 及びアクセス権 1 3 6 2 の各項目を有する一つ以上のレコードで構成される。各レコードは一つのデータセットに対応している。

【 0 0 8 2 】

上記項目のうち、データセット ID 1 3 6 1 には、データセットのデータセット ID が

50

設定される。アクセス権 1362 には、当該データセットについてアクセス権を有するユーザ（本例では組織）とアクセス権の内容が設定される。尚、同図における「R」は読み込み権限であり、「W」は書き込み権限である。

【0083】

図17は、図8に示した検証先決定用処理ノード情報215の一例である。検証先決定用処理ノード情報215は、処理ノード情報ID2151、組織ID2152、アクセス可能データセット2153、現在利用中データセット2154、及び検証実施必要ポイント2155の各項目を有する一つ以上のレコードで構成される。各レコードは一つの検証先決定用の処理ノードの情報に対応している。

【0084】

検証先決定用処理ノード情報215は、クライアントノード21が検証先の分散台帳ノード22を選択する際に用いられる情報である。

【0085】

上記項目のうち、処理ノード情報ID2151には、検証先決定用の処理ノードの情報の識別子（以下、「処理ノード情報ID」と称する。）が設定される。組織ID2152には、組織IDが設定される。

【0086】

アクセス可能データセット2153には、当該組織がアクセス可能な全てのデータセットのデータセットIDが設定される。現在利用中データセット2154には、当該組織が現在利用している全てのデータセットのデータセットIDが設定される。

【0087】

検証実施必要ポイント2155には、当該組織が検証用コンテナを提供するために必要なポイント値が設定される。

【0088】

図18は、図8に示した検証先決定用パラメータ情報216の一例である。検証先決定用パラメータ情報216は、パラメータ情報ID2161、パラメータ2162、及びパラメータ値2163の各項目を有する一つ以上のレコードで構成される。各レコードは一つの検証先決定用のパラメータの情報に対応している。

【0089】

上記項目のうち、パラメータ情報ID2161には、検証先決定用のパラメータ情報の識別子（以下、「パラメータ情報ID」と称する。）が設定される。パラメータ2162には、パラメータを示す情報が設定される。例えば、「検証ノード（デプロイ）最大数」や「検証ノード（データ入力）最大数」が設定される。パラメータ値2163には、当該パラメータの値が設定される。

【0090】

図19は、図8に示した組織情報217の一例である。組織情報217は、組織ID2171、組織名2172、及び分散台帳ノードID2173の各項目を有する一つ以上のレコードで構成される。各レコードは一つの組織に対応している。

【0091】

上記項目のうち、組織ID2171には、組織IDが設定される。組織名2172には、当該組織の名称が設定される。分散台帳ノードID2173には、当該組織の分散台帳ノード22の識別子（以下、「分散台帳ノードID」と称する。）が設定される。一つの組織が複数の分散台帳ノード22を有している場合もあり、その場合、分散台帳ノードID2173には複数の分散台帳ノードIDが設定される。

【0092】

図20は、図9に示したサブデータセット検証結果251の一例である。サブデータセット検証結果251は、サブデータセットID2511、検証実施レベル2112、及び更新日時2113の各項目を有する一つ以上のレコードで構成される。各レコードは一つのサブデータセットに対応している。

【0093】

10

20

30

40

50

上記項目のうち、サブデータセットID 2511には、サブデータセットIDが設定される。検証実施レベル2112には、検証の質（信頼性）を表す評価指標である検証実施レベルが設定される。本例では検証実施レベルの値が大きい程、検証の質は高くなる。分散台帳に書き込まれている検証実施レベルから行われた検証の質を知ることができる。データ源50から加工することなしにデータセットに格納された場合は検証実施レベル2112には「N/A」が設定される。更新日時2513には、検証実施レベル2112の内容が更新された日時が設定される。

【0094】

図21は、図9に示した組織保持ポイント情報252の一例である。組織保持ポイント情報252は、組織ID2521、組織名2522、残ポイント2523、及び更新日時2524の各項目を有する一つ以上のレコードで構成される。各レコードは一つの組織に対応している。

10

【0095】

上記項目のうち、組織ID2521には組織IDが設定される。組織名2522には当該組織の名称が設定される。残ポイント2523には、当該組織が現在保持しているポイント値が設定される。更新日時2524には、残ポイント2523の直近の更新日時（最終更新日時）が設定される。

【0096】

図22は、図9に示したコンテナログ253の一例である。コンテナログ253は、コンテナログID2531及びログ情報2532の各項目を有する一つ以上のレコードで構成される。各レコードは一つのコンテナログに対応している。

20

【0097】

上記項目のうち、コンテナログID2531には、コンテナログIDが設定される。ログ情報2532には、コンテナから出力されたログの内容が設定される。例示するログ情報2532は、入力データの値、出力データの値、出力先のサブデータセットの情報等を含む。

【0098】

図23A及び図23Bは、図9に示したブロックチェーン227の一例である。ブロックチェーン227は、ブロックID2271、トランザクション情報（タイムスタンプ2272、SCID2273、呼び出し関数2274、TX発行者2275、TX承認者2276、RWセット2277）、及び前ブロックハッシュ値2278の各項目を有する複数のレコードで構成される。各レコードは一つのブロックに対応している。例示するブロックチェーン227では、一つのブロックに一つのトランザクション（TX）を格納しているが、一つのブロックに複数のトランザクション（TX）が格納されていてもよい。

30

【0099】

ブロックID2271には、ブロックチェーンを構成するブロックの識別子（以下、「ブロックID」と称する。）が設定される。トランザクション情報のうち、タイムスタンプ2272には、TXを実行した日時（タイムスタンプ）が設定される。

【0100】

SCID2273には、実行したスマートコントラクト（SC）の識別子（以下、「スマートコントラクトID」と称する。）が設定される。呼び出し関数2274には、呼び出したSC内の関数が設定される（後述する図24の関数2283）。TX発行者2275には、トランザクション（TX）を発行した組織の名称が設定される。TX承認者2276には、トランザクション（TX）を承認した組織の名称が設定される。

40

【0101】

RWセット2277には、Read/Writeセット（ステート（図20のサブデータセット検証結果251、図21の組織保持ポイント情報252、図22のコンテナログ253）に対する読み込みや書き込みの内容が設定される。本例の場合、例えば、ブロックID2271が「1」のレコードのRWセット2277は、データ源50からデータを加工なしにデータセットに格納している場合であり、keyである「検証実施レベル」に

50

ついて Value として「0」が書き込まれている。またブロック ID 2271 が「2」の RW セット 2277 は、検証用コンテナをデプロイするトランザクション (TX) に関するものであり、デプロイの結果、ポイントが付与されるため、key である「組織 A ポイント」、「組織 B ポイント」に夫々、Value として「100」、「400」が書き込まれている。またブロック ID が「5」のレコードの RW セット 2277 は、コンテナログに関するものであり、key である「コンテナログ 1」に Value としてコンテナログ 253 のコンテナログ ID が「1」のログ情報 2302 の内容が設定されている。尚、ステートの情報に加えて、検証用コンテナから出力されたデータの値等の検証結果を RW セットに含めてもよい。

【0102】

前ブロックハッシュ値 2278 には、前ブロックのハッシュ値が格納される。前ブロックのハッシュ値を持つことによりブロックチェーン内のデータの改ざんの困難性が確保される。

【0103】

図 24 は、図 9 に示したスマートコントラクト情報 228 の一例である。スマートコントラクト情報 228 は、スマートコントラクト ID 2281、種類 2282、及び関数 2283 の各項目を有する一つ以上のレコードで構成される。各レコードは一つのスマートコントラクトに対応している。

【0104】

スマートコントラクト情報 228 には、スマートコントラクト ID が設定される。種類 2282 には、スマートコントラクトの種類 (処理内容) を示す情報が設定され、例えば、「検証 (検証用コンテナデプロイ、ポイント更新)」、「検証 (データ入力)」、「検証 (出力データ検証)」等が設定される。関数 2283 には、各スマートコントラクトの関数が設定される。

【0105】

図 25 は、図 9 に示した出力データ 229 の一例である。出力データ 229 は、コンテナ ID 2291、入力サブデータセット ID 2292、及び出力データ 2293 の各項目を有する一つ以上のレコードで構成される。各レコードは一つの出力データに対応している。出力データ 229 は、分散台帳 ノード 22 がデータの加工内容の検証に用いる情報を含む。

【0106】

上記項目のうち、コンテナ ID 2291 には、出力データの識別子 (本例ではコンテナ ID) が設定される。入力サブデータセット ID 2292 は、コンテナに入力するデータを含むサブデータセットのサブデータセット ID が設定される。出力データ 2293 には、出力されたデータの内容が設定される。

【0107】

図 26 は、図 10 に示したコンテナイメージ 651 の一例である。コンテナイメージ 651 は、コンテナ ID 6051 及び内容 6052 の各項目を含む一つ以上のレコードで構成されている。各レコードは一つのコンテナに対応している。

【0108】

上記項目のうち、コンテナ ID 6051 には、コンテナイメージ ID が設定される。内容 6052 には、コンテナイメージの内容 (実体) が設定される。

【0109】

< 処理説明 >

図 27 は、ユーザがデータフローを追加する際にデータ加工履歴管理システム 1 において行われる処理を説明するシーケンス図である。

【0110】

まずユーザ端末 30 が、データフロー管理 ノード 40 に対してデータフローの追加要求を送信する (S2711)。データフロー管理 ノード 40 は、上記追加要求を受信すると、データフローの追加先の組織の管理 ノード 11 にデータフローの追加要求を送信する (

10

20

30

40

50

S 2 7 1 2)。

【 0 1 1 1 】

管理ノード 1 1 (処理ノード 1 2 でもよい) は、上記追加要求を受信すると、自身が属する組織の処理ノード 1 2 にコンテナのデプロイ要求を送信する (S 2 7 1 3)。処理ノード 1 2 は、上記デプロイ要求を受信すると、コンテナレジストリ 6 0 から追加するデータフローのコンテナイメージを取得し (S 2 7 1 4 , S 2 7 1 5)、取得したコンテナイメージを用いて、データの加工処理を行うコンテナをデプロイする (S 2 7 1 6)。

【 0 1 1 2 】

図 2 8 は、データフローに従ってある組織でデータの加工処理が行われた際にデータ加工履歴管理システム 1 において行われる加工内容の検証処理に関する準備処理を説明するシーケンス図である。尚、以下の説明におけるデータは、例えば、時系列データであり、サブデータセットは一定期間ごとに生成されるものとする。

10

【 0 1 1 3 】

まず加工処理の対象となるデータ又はデータセットが、データ源 5 0 又はデータ格納ノード 1 3 から処理ノード 1 2 に送られる (S 2 8 1 1 , S 2 8 1 2)。

【 0 1 1 4 】

処理ノード 1 2 のコンテナは、送られてくるデータを受信し、受信したデータについて加工処理を行い (S 2 8 1 3)、加工後のデータをデータ格納ノード 1 3 のデータセットに格納していく (S 2 8 1 4)。

【 0 1 1 5 】

20

処理ノード 1 2 のコンテナは、サブデータセットの生成が完了すると、自身が属する組織のクライアントノード 2 1 に加工内容の検証を要求する (S 2 8 1 5)。上記要求には、サブデータセットの作成完了の通知と、データを加工した際に生成されたコンテナログが含まれる。尚、本例では、このように処理ノード 1 2 のコンテナが加工内容の検証要求を、サブデータセットの生成が完了したタイミングで送信するが、検証要求を送信するタイミングは必ずしも限定されない。例えば、ユーザが指定したタイミングで検証要求を送信してもよい。また加工したデータの数等に応じて検証要求を送信してもよい。

【 0 1 1 6 】

クライアントノード 2 1 は、検証要求を受信すると、検証先とする他の組織の分散台帳ノード 2 2 を選択する (S 2 8 1 6)。尚、この処理 (以下、「検証先ノード選択処理 S 2 8 1 6 」と称する。) の詳細については後述する。

30

【 0 1 1 7 】

続いて、クライアントノード 2 1 は、選択した他の組織の複数の分散台帳ノード 2 2 の夫々に検証用コンテナのデプロイ要求を送信する (S S 2 8 1 7)。上記デプロイ要求はコンテナイメージのイメージ ID を含む。尚、リソースの節約のため、検証用コンテナに入力されるサブデータセットの検証実施レベル 2 5 1 2 に「 0 」が設定されている場合、クライアントノード 2 1 は上記デプロイ要求の送信を行わない。

【 0 1 1 8 】

各組織の分散台帳ノード 2 2 は、上記デプロイ要求を受信すると、自組織の管理ノード 1 1 に検証用コンテナのデプロイ要求を送信する (S 2 8 1 8)。

40

【 0 1 1 9 】

管理ノード 1 1 は、上記デプロイ要求を受信すると、当該デプロイ要求に指定されているコンテナのデプロイを自組織の処理ノード 1 2 に要求する (S 2 8 1 9)。尚、管理ノード 1 1 は、当該コンテナを検証用としてデプロイする。

【 0 1 2 0 】

処理ノード 1 2 は、上記デプロイ要求を受信すると、当該デプロイ要求に指定されているコンテナイメージをコンテナレジストリ 6 0 から取得し (S 2 8 2 0 、 S 2 8 2 1)、取得したコンテナイメージを用いて検証用コンテナをデプロイし (S 2 8 2 2)、管理ノード 1 1 にデプロイの完了通知を送信する (S 2 8 2 3)。尚、上記完了通知は、デプロイした検証用コンテナの所在を特定する情報 (U R L (Uniform Resource Locator)、I

50

Pアドレス等)を含む。

【0121】

管理ノード11は、上記完了通知を受信すると、検証用コンテナのデプロイの完了通知を自組織の分散台帳ノード22に通知する(S2824)。

【0122】

分散台帳ノード22は、上記完了通知を受信すると、自組織のポイントに検証用コンテナをデプロイした分に相当するポイントを反映(図21に示した組織ポイント情報226の残ポイント2523を更新)する(S2825)。尚、ポイントの更新履歴は、図23Aに例示したRWセット2277としてブロックチェーン227に記録される。

【0123】

続いて、分散台帳ノード22は、S2817で検証用コンテナのデプロイ要求を送信してきたクライアントノード21にデプロイの完了通知を送信する(S2826)。

【0124】

図29は、図28に示した検証先ノード選択処理S2816の詳細を説明するフローチャートである。同図に示すように、検証先ノード選択処理S2816は、検証用コンテナのデプロイ先の分散台帳ノード22を選択する処理(S2911~S2914)と、検証用コンテナにデータを入力する分散台帳ノード22を選択する処理(S2915~S2917)とを含む。

【0125】

まずクライアントノード21は、図20に示した検証先決定用処理ノード情報215を参照し、自組織以外の他の組織のうち、加工後のデータセットを現在利用している組織があるか否かを判定する(S2911)。加工後のデータセットを現在利用している組織がなければ(S2911:NO)、クライアントノード21は、加工後のデータセット(サブデータセット)の図20に示したサブデータセット検証結果251の検証実施レベル2512に「0」を設定し、処理を終了する(S2912)。このように加工後のデータセットが他の組織で利用されない場合、分散台帳ノード22は選択されず、検証は行われないので、リソースが無駄に消費されてしまうのを防ぐことができる。

【0126】

一方、加工後のデータセットを現在利用している組織がある場合(S2911:YES)、クライアントノード21は、加工後のデータセットを現在利用している組織の分散台帳ノード22を検証用コンテナのデプロイ先のノード(第1の分散台帳ノード)として選択する(S2913)。このように加工後のデータセットを現在利用している組織の分散台帳ノード22を検証用コンテナのデプロイ先として指定することで、検証により生じる負荷を負担する意義を有する組織を選択することができる。デプロイ先として選択可能な分散台帳ノード22の数が、図18に示した検証先決定用パラメータ情報216の検証ノード(デプロイ)最大数2162のパラメータ値2163を超える場合、クライアントノード21は、例えば、必要なポイントが少ない(安い)組織の分散台帳ノード22を優先して最大数分の分散台帳ノード22を選択する。

【0127】

続いて、クライアントノード21は、加工後のデータセットの図20に示したサブデータセット検証結果251の検証実施レベル2512に、選択した分散台帳ノード22の数を設定する(S2914)。従って、選択した分散台帳ノード22の数が高い程、検証実施レベル(検証の質)は高くなる。これは多くの分散台帳ノード22で検証が行われる程、検証の質が向上するという前提に基づく。

【0128】

続いて、クライアントノード21は、図16に示したデータセットアクセス権136を参照し、加工処理を行った組織以外の他の組織のうち、加工後のデータセットにアクセス可能な組織があるか否かを判定する(S2915)。加工後のデータセットにアクセス可能な組織がない場合(S2915:NO)、クライアントノード21は、加工後のデータセットの図20に示したサブデータセット検証結果251の検証実施レベル2512に「

10

20

30

40

50

0」を設定し、処理を終了する（S 2 9 1 2）。このように加工後のデータセットにアクセス可能な組織がない場合、分散台帳ノード 2 2 は選択されず、検証は行われないので、リソースが無駄に消費されてしまうのを防ぐことができる。

【 0 1 2 9 】

一方、加工後のデータセットにアクセス可能な組織がある場合（S 2 9 1 5 : Y E S）、クライアントノード 2 1 は、加工後データセットにアクセス可能な組織の分散台帳ノード 2 2 を検証用コンテナにデータを入力するノード（第 2 の分散台帳ノード）として選択する（S 2 9 1 6）。デプロイ先の分散台帳ノード 2 2 の数が、図 1 8 に示した検証先決定用パラメータ情報 2 1 6 の検証ノード（データ入力）最大数 2 1 6 2 のパラメータ値 2 1 6 3 を超える場合、クライアントノード 2 1 は、例えば、必要ポイントが少ない（安い）組織の分散台帳ノード 2 2 を優先して最大数分の分散台帳ノード 2 2 を選択する。このように、クライアントノード 2 1 は、分散台帳ノード 2 2 の選択に際し、組織のデータセットへのアクセス権を考慮するので、データのセキュリティを確保しつつ、検証処理を行う分散台帳ノード 2 2 を選択することができる。

10

【 0 1 3 0 】

続いて、クライアントノード 2 1 は、加工後のデータセット（サブデータセット）の図 2 0 に示したサブデータセット検証結果 2 5 1 の検証実施レベル 2 5 1 2 に、選択した分散台帳ノード 2 2 の数を加算する（S 2 9 1 7）。

【 0 1 3 1 】

続いて、クライアントノード 2 1 は、S 2 9 1 3 で選択した検証用コンテナのデプロイ先の分散台帳ノード 2 2 と、S 2 9 1 6 で選択した検証用コンテナにデータを入力する分散台帳ノード 2 2 との一つ以上の組み合わせを生成する（S 2 9 1 8）。この組み合わせの方法は必ずしも限定されない。尚、例えば、選出した検証用コンテナにデータを入力する分散台帳ノード 2 2 の数が、選出した検証用コンテナのデプロイ先の分散台帳ノード 2 2 の数よりも少ない場合、一つの検証用コンテナにデータを入力する分散台帳ノード 2 2 が複数の検証用コンテナのデプロイ先の分散台帳ノード 2 2 と組み合わせられる。

20

【 0 1 3 2 】

図 3 0 は、ある組織でデータの加工処理が行われた際にデータ加工履歴管理システム 1 において行われる、上記のある組織におけるデータの加工内容の検証処理を説明するシーケンス図である。

30

【 0 1 3 3 】

まずクライアントノード 2 1 は、検証先ノード選択処理 S 2 8 1 6 で選択したデータの入力を担当する分散台帳ノード 2 2 に、S 2 9 1 8 において当該分散台帳ノード 2 2 と組み合わせられた検証用コンテナのデプロイ先の分散台帳ノード 2 2 が属する組織の処理ノード 1 2 の検証用コンテナにデータ（加工処理における入力データ）を入力するように要求する（S 3 0 1 1）。尚、上記要求には、入力データセットや入力先のコンテナの所在を示す情報（URL、IP アドレス等）が含まれる。

【 0 1 3 4 】

分散台帳ノード 2 2 は、上記要求を受信すると、当該要求で指定されているデータ（データセット）を保持しているデータ格納ノード 1 3 に、上記検証用コンテナへのデータの入力要求を送信する（S 3 0 1 2）。

40

【 0 1 3 5 】

データ格納ノード 1 3 は、上記要求で指定されているデータ（データセット）を上記処理ノード 1 2 の検証用コンテナに入力する（S 3 0 1 3）。

【 0 1 3 6 】

検証用コンテナは、入力されたデータに対して、コンテナに定義されている加工処理を実行し（S 3 0 1 4）、加工後のデータを、検証用コンテナのデプロイ要求を送信してきた分散台帳ノード 2 2 に送信する（S 3 0 1 5）。尚、このとき、検証用コンテナが加工内容に関する情報を含むコンテナログを上記分散台帳ノード 2 2 に送信してもよい。

【 0 1 3 7 】

50

分散台帳ノード 2 2 は、加工後のデータを受信すると、S 3 0 1 1 でデータの入力を要求してきたクライアントノード 2 1 に完了通知を送信する (S 3 0 1 6) 。

【 0 1 3 8 】

クライアントノード 2 1 は、上記完了通知を受信すると、分散台帳ノード 2 2 に加工後のデータ (出力データ) の検証を要求する (S 3 0 1 7) 。この要求には、図 2 8 の加工処理 S 2 8 1 3 における加工後のデータが含まれる。

【 0 1 3 9 】

分散台帳ノード 2 2 は、上記要求に含まれている上記加工後のデータと、S 3 0 1 5 で受信した (検証環境からの) 加工後のデータとを比較することにより、ある組織におけるデータの加工内容が正しいか否かを検証し (S 3 0 1 8) 、検証結果をクライアントノード 2 1 に通知する (S 3 0 1 9) 。

10

【 0 1 4 0 】

クライアントノード 2 1 は、全ての検証先の分散台帳ノード 2 2 から上記検証結果を受信し、その結果を確認する (S 3 0 2 0) 。上記確認の結果、例えば、検証を依頼した分散台帳ノード 2 2 のうち、既定数以上の分散台帳ノード 2 2 からの結果が同一であった場合、クライアントノード 2 1 は、加工内容 (検証対象の加工処理の内容) は正しいと判定し、全ての分散台帳ノード 2 2 に対して、分散台帳 (図 2 0 に示したサブデータセット検証結果 2 5 1) の更新要求を送信する (S 3 0 2 1) 。尚、加工内容が正しいか否かの判定方法は必ずしも限定されない。

【 0 1 4 1 】

上記要求を受信した分散台帳ノード 2 2 は、分散台帳をステートの内容 (サブデータセット検証結果 2 5 1 、組織保持ポイント情報 2 5 2 、コンテナログ 2 5 3) に基づき更新し (S 3 0 2 2) 、管理ノード 1 1 に検証用コンテナの削除を要求する (S 3 0 2 3) 。

20

【 0 1 4 2 】

管理ノード 1 1 は、上記要求を受信すると、処理ノード 1 2 に検証用コンテナの削除要求を送信する (S 3 0 2 4) 。

【 0 1 4 3 】

処理ノード 1 2 は、指定された検証用コンテナを削除し (S 3 0 2 5) 、管理ノード 1 1 に削除完了を通知する (S 3 0 2 6) 。

【 0 1 4 4 】

管理ノード 1 1 は、上記通知を受信すると、分散台帳ノード 2 2 に処理完了を通知する (S 3 0 2 7) 。

30

【 0 1 4 5 】

以上、詳細に説明したように、本実施形態のデータ加工履歴管理システム 1 によれば、ブロックチェーンの仕組みを利用して他の組織で行われたデータの加工処理を検証することができる。また検証対象となる加工処理はブロックチェーンの外部のリソースであるコンテナ基盤のリソースを用いて行われるため、既存のツール等を活用して加工内容の検証を効率よく行うことができる。このように、本実施形態のデータ加工履歴管理システム 1 によれば、複数の組織で取り扱われるデータの加工内容を検証する仕組みを効率よく実現してデータの信頼性を向上することができる。

40

【 0 1 4 6 】

以上、本発明の一実施形態について詳細に説明したが、本発明は上記の実施形態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ことはいうまでもない。例えば、上記の実施形態は本発明を分かりやすく説明するために詳細に説明したものであり、説明した全ての構成を備えるものに必ずしも限定されるものではない。また上記実施形態の構成の一部について、他の構成の追加や削除、置換をすることが可能である。

【 0 1 4 7 】

例えば、検証先の分散台帳ノード 2 2 の決定を、各組織が過去に利用していたデータセットの情報を利用して行ってもよい。

50

【 0 1 4 8 】

また上記の各構成、機能部、処理部、処理手段等は、それらの一部または全部を、例えば、集積回路で設計する等によりハードウェアで実現してもよい。また上記の各構成、機能等は、プロセッサがそれぞれの機能を実現するプログラムを解釈し、実行することによりソフトウェアで実現してもよい。各機能を実現するプログラム、テーブル、ファイル等の情報は、メモリやハードディスク、SSD (Solid State Drive) 等の記録装置、ICカード、SDカード、DVD等の記録媒体に置くことができる。

【 0 1 4 9 】

また上記の各図において、制御線や情報線は説明上必要と考えられるものを示しており、必ずしも実装上の全ての制御線や情報線を示しているとは限らない。例えば、実際には殆ど全ての構成が相互に接続されていると考えてもよい。

10

【 0 1 5 0 】

また以上に説明した各情報処理装置の各種機能部、各種処理部、各種データベースの配置形態は一例に過ぎない。各種機能部、各種処理部、各種データベースの配置形態は、これらの装置が備えるハードウェアやソフトウェアの性能、処理効率、通信効率等の観点から最適な配置形態に変更し得る。

【 0 1 5 1 】

また前述した各種のデータを格納するデータベースの構成 (スキーマ (Schema) 等) は、リソースの効率的な利用、処理効率向上、アクセス効率向上、検索効率向上等の観点から柔軟に変更し得る。

20

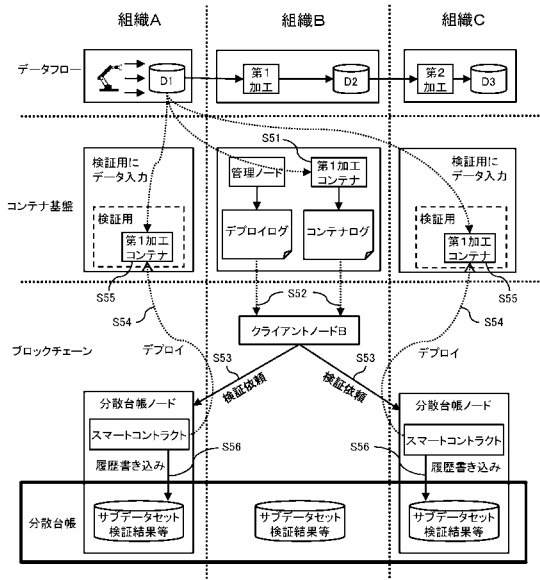
【 符号の説明 】

【 0 1 5 2 】

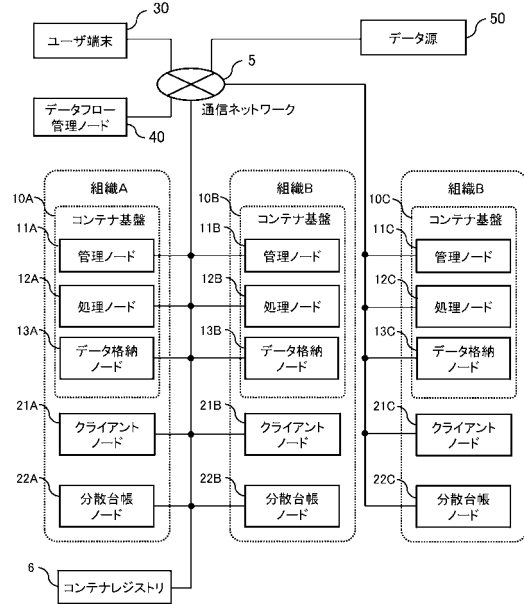
1 データ加工履歴管理システム、5 通信ネットワーク、10 コンテナ基盤、11 管理ノード、111 記憶部、112 コンテナデプロイ部、151 デプロイログ、12 処理ノード、121 記憶部、122 コンテナ実行部、123 コンテナログ管理部、125 コンテナ情報、126 コンテナログ、13 データ格納ノード、131 記憶部、135 データセット、136 データセットアクセス権、137 データ、21 クライアントノード、211 記憶部、212 トランザクション発行部、213 ノード選択部、214 業務アプリ、215 検証先決定用処理ノード情報、216 検証先決定用パラメータ情報、217 組織情報、22 分散台帳ノード、221 記憶部、222 スマートコントラクト実行/管理部、223 検証用コンテナデプロイ部、224 データ入力部、225 検証実行部、227 ブロックチェーン、228 スマートコントラクト情報、229 出力データ、251 サブデータセット検証結果、252 組織保持ポイント情報、253 コンテナログ、30 ユーザ端末、40 データフロー管理ノード、411 記憶部、412 データフロー管理部、451 データフロー情報、50 データ源、60 コンテナレジストリ、601 記憶部、611 コンテナイメージ管理部、651 コンテナイメージ

30

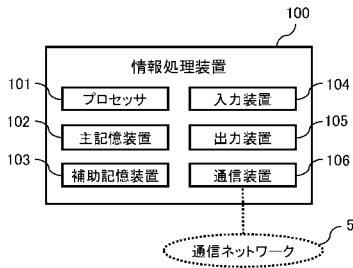
【図1】



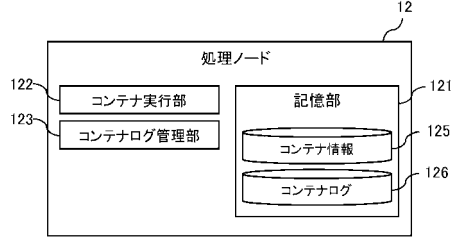
【図2】



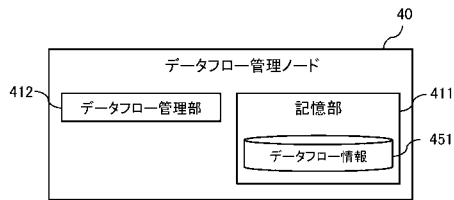
【図3】



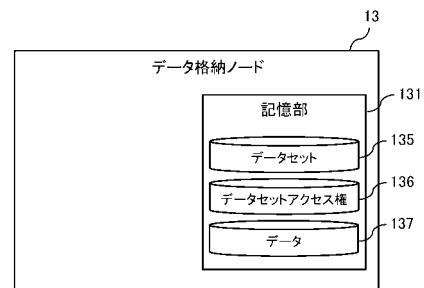
【図6】



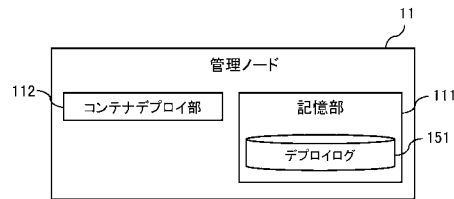
【図4】



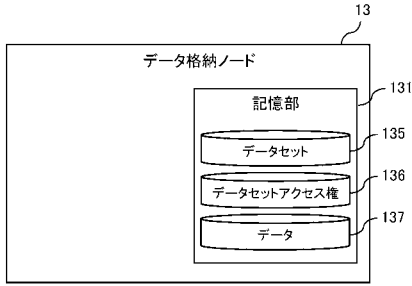
【図7】



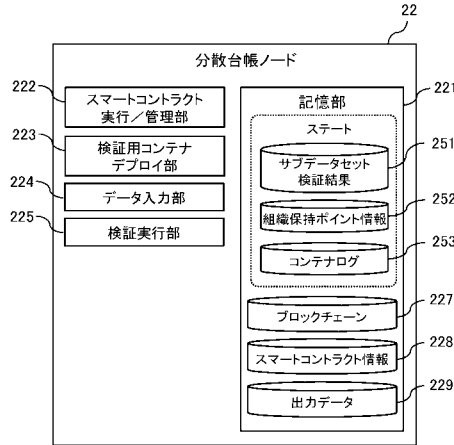
【図5】



【図 8】



【図 9】



【図 13】

コンテナ情報 125

コンテナID	イメージID	状態
1	1	稼働中
2	2	稼働中

【図 14】

コンテナログ 126

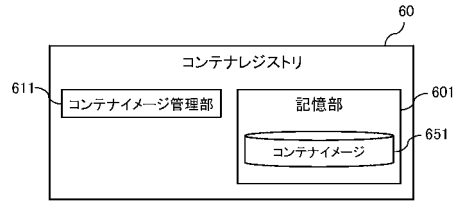
コンテナログID	コンテナID	ログ情報
1	1	2019/3/10 1:24:00. データ11(値:100,300.0)(サブデータセット1)からデータ21(値:{"val":100,"result":"OK"})を生成。サブデータセット21に格納 2019/3/10 1:24:01. データ12(値:20,310.1)(サブデータセット1)からデータ22(値:{"val":20,"result":"NG"})を生成。サブデータセット21に格納 2019/3/11 0:00:01. データ13(値:110,310.0)(サブデータセット12)からデータ23(値:{"val":110,"result":"OK"})を生成。サブデータセット22に格納 2019/3/11 0:01:01. データ14(値:120,310.0)(サブデータセット12)からデータ24(値:{"val":120,"result":"OK"})を生成。サブデータセット22に格納
2	2	2019/3/11 2:35:10. データ23(値:{"val":110,"result":"OK"})~データ26(サブデータセット22)からデータ31(値:{"period":"3/11","result":"All OK"})を生成。サブデータセット31に格納

【図 15】

データセット 135

データセットID	サブデータセットID	取得期間
1	11	2019/3/10 0:00:00 - 2019/3/10 23:59:59
1	12	2019/3/11 0:00:00 - 2019/3/11 23:59:59
2	21	2019/3/10 0:00:00 - 2019/3/10 23:59:59
2	22	2019/3/10 0:00:00 - 2019/3/10 23:59:59

【図 10】



【図 11】

データフロー情報 451

データフローID	作成組織	作成日時	削除日時	入力データセットID	加工コンテナID	出力データセットID	状態
1	組織B	2019/3/10 01:23:45	N/A	1	1	2	アクティブ
2	組織C	2019/3/11 02:34:56	N/A	2	2	3	アクティブ

【図 12】

デプロイログ 151

デプロイログID	作成日時	作成コンテナID	イメージID
1	2019/3/10 01:23:50	1	1
2	2019/3/11 02:35:00	2	2

【図 16】

データセットアクセス権 136

データセットID	アクセス権(ユーザ/権限)
1	組織A(RW)、組織B(RW)、組織C(R)
2	組織B(RW)、組織C(RW)、組織D(RW)、組織E(RW)
3	組織B(RW)、組織C(RW)、組織E(R)

【図 17】

検証先決定用処理ノード情報 215

処理ノード情報ID	組織ID	アクセス可能データセット	現在利用中データセット	検証実施必要ポイント
1	組織A	1	N/A	10
2	組織B	1, 2, 3	2	20
3	組織C	1, 2, 3	2, 3	15

【図 18】

検証先決定用パラメータ情報 216

パラメータ情報ID	パラメータ	パラメータ値
1	検証ノード(デプロイ)最大数	5
2	検証ノード(データ入力)最大数	4

【図 19】

組織情報

組織ID	組織名	分散台帳ノードID
1	組織A	1
2	組織B	2
3	組織C	3

【図 22】

コンテナログ

コンテナログID	ログ情報
1	2019/3/10 1:24:00. データ11(値:100,300.0) (サブデータセット11)からデータ21(値: ["val":100,"result": "OK"])を生成. サブデータセット21に格納

【図 20】

サブデータセット検証結果

サブデータセットID	検証実施レベル(検証ノード数)	更新日時
11	N/A	2019/3/09 03:33:33
12	N/A	2019/3/10 00:02:09
21	0	2019/3/11 00:02:10
22	2 (コンテナデプロイ:組織C、データ入力:組織A)	2019/3/12 0:00:10

【図 21】

組織保持ポイント情報

組織ID	組織名	残ポイント	更新日時
1	組織A	100	2019/3/11 01:11:11
2	組織B	210	2019/3/11 01:11:12
3	組織C	400	2019/3/11 02:00:01

【図 23 A】

ブロックチェーン

ブロックID	タイムスタンプ	SC ID	呼び出し関数	TX 発行者	TX 承認者	トランザクション情報				前ブロックハッシュ値
						RW区分	Key	Value	バージョン	
1	2019/3/11 00:00:10	3	検証(出力データ検証)	組織B	組織B	W	検証実施レベル(サブデータセット21)	0	4	Xxx
2	2019/3/12 00:00:10	1	デプロイ	組織B	組織A.C	W	組織Aポイント	100	2	Xxxa
						W	組織Cポイント	400	2	
3	2019/3/12 00:01:00	2	データ入力	組織B	組織A.C	N/A				Xxxb
4	2019/3/12 00:01:00	3	検証	組織B	組織A.C	W	検証実施レベル(サブデータセット22)	2	3	Xxxc

【図 23 B】

ブロックチェーン

ブロックID	タイムスタンプ	SC ID	呼び出し関数	TX 発行者	TX 承認者	トランザクション情報				前ブロックハッシュ値
						RW区分	Key	Value	バージョン	
5	2019/3/10 01:30:00	4	ログ蓄積	組織B	組織A.C	W	コンテナログ1	2019/3/10 1:24:00. データ11(値:100,300.0) (サブデータセット11)からデータ21(値: ["val":100,"result": "OK"])を生成. サブデータセット21に格納	1	Xxx

【図 24】

スマートコントラクト情報

スマートコントラクトID	種類	関数
1	検証(コンテナデプロイ、ポイント更新)	・デプロイ
2	検証(データ入力)	・データ入力
3	検証(出力データ検証)	・検証
4	コンテナログ蓄積	・ログ蓄積

【図 25】

出カデータ

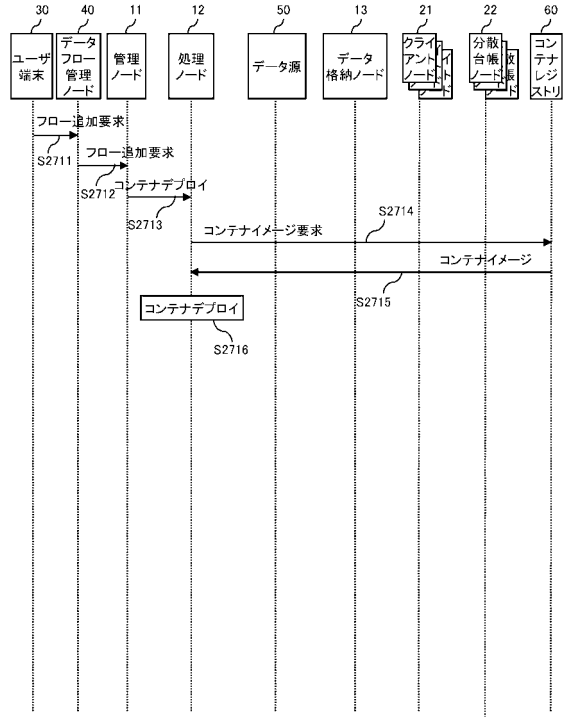
2291	2292	2293
コンテナID	入力サブデータセットID	出カデータ
101	12	[{"val":110,"result": "OK"} {"val":120,"result": "OK"}]
201	22	[{"period": "3/11","result": "All OK"}]

【図 26】

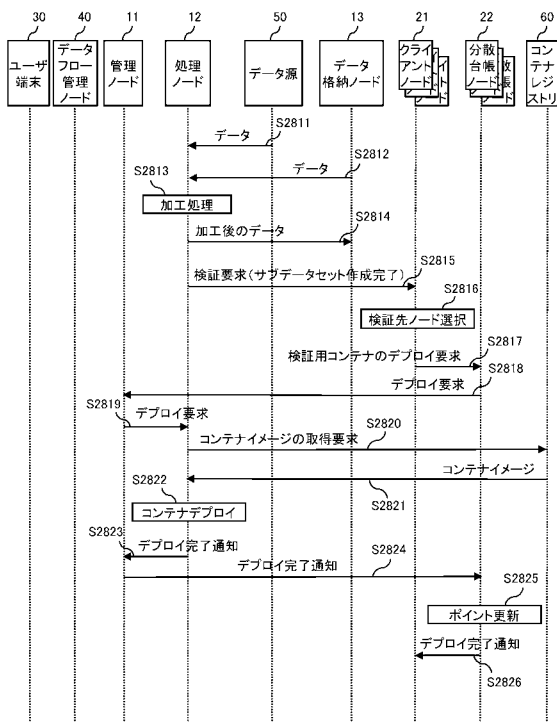
コンテナイメージ

6051	605
コンテナID	内容
1	2FDCE3E13116503B5DF0697.....
2	9407ECBC1F7EDF0B4634D5.....
:	:

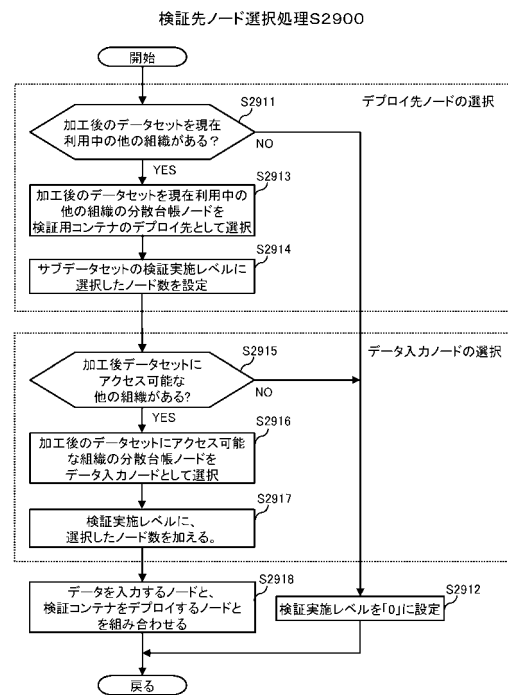
【図 27】



【図 28】



【図 29】



【図 30】

