(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2019-525685
(P2019-525685A)
(43)公表日　令和1年9月5日(2019.9.5)

(51)Int.Cl.		F I		テーマコード（参考）
H04L 9/32	***(2006.01)***	H04L	9/00	675Z
G06F 21/60	***(2013.01)***	H04L	9/00	675C
G06F 21/64	***(2013.01)***	G06F	21/60	320
		G06F	21/64	ZIT

審査請求 未請求　予備審査請求 未請求　（全 117 頁）

(21)出願番号　特願2019-521195 (P2019-521195)
(86)(22)出願日　平成29年7月7日 (2017.7.7)
(85)翻訳文提出日　平成31年3月8日 (2019.3.8)
(86)国際出願番号　PCT/GB2017/052004
(87)国際公開番号　WO2018/007828
(87)国際公開日　平成30年1月11日 (2018.1.11)
(31)優先権主張番号　1611948.9
(32)優先日　平成28年7月8日 (2016.7.8)
(33)優先権主張国・地域又は機関
英国 (GB)

(71)出願人　519007651
カリプトン　インターナショナル　リミテッド
KALYPTON INTERNATIONAL LIMITED
英国領ジブラルタル　ジブラルタル　ジロズ　パッセージ　ジブロ　ハウス　4
(74)代理人　100105957
弁理士　恩田　誠
(74)代理人　100068755
弁理士　恩田　博宣
(74)代理人　100142907
弁理士　本田　淳

最終頁に続く

(54)【発明の名称】分散トランザクション処理及び認証システム

(57)【要約】
第１エンティティに関連するデバイスでデータトランザクションレコーディング方法は、第１シードデータを決定するステップと、第１エンティティと第２エンティティとの間の第１データトランザクションのレコードを生成するステップと、少なくとも第１シードデータ及び第１データトランザクションのレコードを結合して第２シードデータを決定するステップと、第２シードデータをハッシュして第１ハッシュを生成するステップであって、第１ハッシュは、第１エンティティを含むデータトランザクションのヒストリーを含む、第１ハッシュを生成するステップと、第１データトランザクションのレコードに対する第１ハッシュをメモリに格納するステップとを含む。

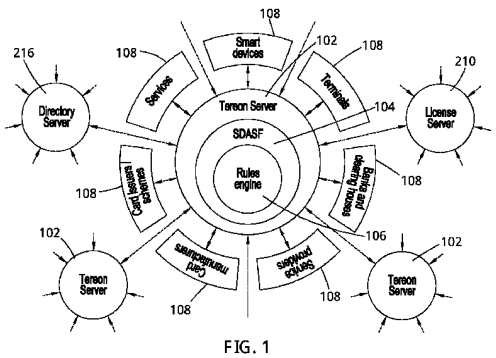


FIG. 1

【特許請求の範囲】
【請求項１】
第１エンティティに関連するデバイスでデータトランザクションレコーディング方法において、
第１シードデータを決定するステップと、
前記第１エンティティと第２エンティティとの間の第１データトランザクションのレコードを生成するステップと、
少なくとも前記第１シードデータ及び前記第１データトランザクションのレコードを結合して第２シードデータを決定するステップと、
前記第２シードデータをハッシュして第１ハッシュを生成するステップであって、前記第１ハッシュは、前記第１エンティティを含むデータトランザクションのヒストリーを含む、前記第１ハッシュを生成するステップと、
前記第１データトランザクションの前記レコードに対する前記第１ハッシュをメモリに格納するステップと、
を含むデータトランザクションレコーディング方法。
【請求項２】
前記第１シードデータは開始ハッシュを含む、請求項１に記載のデータトランザクションレコーディング方法。
【請求項３】
前記開始ハッシュは、前記第１エンティティを含む以前のデータトランザクションのレコードをハッシュした結果である、請求項２に記載のデータトランザクションレコーディング方法。
【請求項４】
前記開始ハッシュはランダムハッシュを含む、請求項２に記載のデータトランザクションレコーディング方法。
【請求項５】
前記ランダムハッシュは、前記デバイスからの署名、前記ランダムハッシュが生成された日付、及び前記ランダムハッシュが生成された時間のうちの少なくとも１つを含む、請求項４に記載のデータトランザクションレコーディング方法。
【請求項６】
第２シードデータを提供するステップは、前記第１シードデータ及び前記第１データトランザクションの前記レコードと第１ゼロ知識証明及び第２ゼロ知識証明を結合するステップをさらに含み、
前記第１ゼロ知識証明は、前記開始ハッシュが前記第１エンティティを含む前記以前のデータトランザクションの真のハッシュを含むという証明を含み、
前記第２ゼロ知識証明は、第２ハッシュが前記第２エンティティを含む以前のデータトランザクションの前記真のハッシュを含むという証明を含む、請求項１乃至請求項５のいずれか一項に記載のデータトランザクションレコーディング方法。
【請求項７】
第２シードデータを提供するステップは、前記第１シードデータ、前記第１データトランザクションの前記レコード、前記第１ゼロ知識証明及び前記第２ゼロ知識証明と第３ゼロ知識証明を結合するステップをさらに含む、請求項６に記載のデータトランザクションレコーディング方法。
【請求項８】
前記第３ゼロ知識証明は、ランダムデータから生成される、請求項７に記載のデータトランザクションレコーディング方法。
【請求項９】
前記第３ゼロ知識証明は、前記第１ゼロ知識証明又は前記第２ゼロ知識証明の繰り返しである、請求項７に記載のデータトランザクションレコーディング方法。
【請求項１０】

前記第３ゼロ知識証明は、前記第２ゼロ知識証明に対応する前記第１データトランザクションの第２レコードを用いて構成される、請求項７に記載のデータトランザクションレコーディング方法。

【請求項１１】

前記第１データトランザクションは少なくとも２つのステージを含み、

第２シードデータを提供するステップは、

前記第１データトランザクションの前記第１ステージのレコードと前記第１ゼロ知識証明を結合するステップと、

前記第１データトランザクションの前記第２ステージのレコードと前記第２ゼロ知識証明を結合するステップと、

を含む、請求項６に記載のデータトランザクションレコーディング方法。

【請求項１２】

第２シードデータを提供するステップは、

前記第１データトランザクションの前記第２ステージのレコードから第３ゼロ知識証明を構成するステップと、

前記第１データトランザクションの前記第２ステージのレコードと前記第２ゼロ知識証明及び前記第３ゼロ知識証明を結合するステップと、

を含む、請求項１１に記載のデータトランザクションレコーディング方法。

【請求項１３】

前記第１データトランザクションは少なくとも３つのステージを含み、

第２シードデータを提供するステップは、

前記第１データトランザクションの前記第３ステージのレコードと前記第１ゼロ知識証明を結合するステップと、

前記第１データトランザクションの前記第３ステージのレコードと前記第３ゼロ知識証明を結合するステップと、

をさらに含む、請求項１１に記載のデータトランザクションレコーディング方法。

【請求項１４】

前記第１データトランザクションは少なくとも３つのステージを含み、

第２シードデータを提供するステップは、

前記第１データトランザクションの前記第３ステージのレコードと前記第１ゼロ知識証明を結合するステップと、

ランダムデータと前記第３ゼロ知識証明を結合するステップと、

をさらに含む、請求項１１に記載のデータトランザクションレコーディング方法。

【請求項１５】

前記第１データトランザクションは少なくとも３つのステージを含み、

第２シードデータを提供するステップは、

前記第１データトランザクションの前記第３ステージのレコードと前記第１ゼロ知識証明を結合するステップと、

前記第１データトランザクションの第４ステージのレコードと前記第２ゼロ知識証明を結合するステップと、

を含み、

前記第１データトランザクションの前記第４ステージは、前記第１データトランザクションの前記第３ステージの繰り返しである、請求項１１に記載のデータトランザクションレコーディング方法。

【請求項１６】

前記第１データトランザクションは少なくとも３つのステージを含み、

第２シードデータを提供するステップは、前記第１データトランザクションの前記第３ステージのレコードと第３ゼロ知識証明を結合するステップをさらに含む、請求項１１に記載のデータトランザクションレコーディング方法。

【請求項１７】

前記第１ゼロ知識証明は、前記第１エンティティに関連する前記デバイスによって構成され、
前記第２ゼロ知識証明は、前記第２エンティティに関連するデバイスによって構成される、請求項６乃至請求項１６のいずれか一項に記載のデータトランザクションレコーディング方法。
【請求項１８】
前記第１ゼロ知識証明及び前記第２ゼロ知識証明を構成するステップは、キー交換アルゴリズムを使用するステップを含む、請求項１７に記載のデータトランザクションレコーディング方法。
【請求項１９】
前記キー交換アルゴリズムは、ＰＡＫＥアルゴリズムを含む、請求項１８に記載のデータトランザクションレコーディング方法。
【請求項２０】
前記第２エンティティに関連するデバイスに前記第１ハッシュを送信するステップと、
前記第２エンティティに関連するデバイスから第２ハッシュを受信するステップであって、前記第２ハッシュは、前記第２エンティティを含む以前のデータトランザクションのハッシュを含む、前記第２ハッシュを受信するステップと、
前記第１パーティー及び前記第２パーティー間の第２データトランザクションのレコードを生成するステップと、
前記第１ハッシュ及び前記第２ハッシュと前記第２データトランザクションの前記レコードを結合して第３シードデータを決定するステップと、
前記第３シードデータをハッシュし、第３ハッシュを生成するステップであって、前記第３ハッシュは、前記第１エンティティを含むデータトランザクションのヒストリー及び前記第２エンティティを含むデータトランザクションのヒストリーを含む、前記第３ハッシュを生成するステップと、
前記第２データトランザクションの前記レコードに対する前記第３ハッシュを前記メモリに格納するステップと、
をさらに含む、請求項１乃至請求項１９のいずれか一項に記載のデータトランザクションレコーディング方法。
【請求項２１】
第３シードデータを提供するステップは、
前記第２データトランザクションの前記レコード、前記第１ハッシュ及び前記第２ハッシュと、第３ゼロ知識証明及び第４ゼロ知識証明とを結合するステップをさらに含み、
前記第３ゼロ知識証明は、前記第１ハッシュが前記第１データトランザクションの真のハッシュを含むという証明を含み、
前記第４ゼロ知識証明は、前記第２ハッシュが前記第２エンティティを含む前記以前のデータトランザクションの前記真のハッシュを含むという証明を含む、請求項２０に記載のデータトランザクションレコーディング方法。
【請求項２２】
前記第２エンティティを含む前記以前のデータトランザクションは、前記第１データトランザクションである、請求項２０又は請求項２１に記載のデータトランザクションレコーディング方法。
【請求項２３】
前記第１エンティティ及び前記第２エンティティの少なくとも一方の識別子と前記ハッシュのそれぞれを関連づけるステップをさらに含む、請求項１乃至請求項２２のいずれか一項に記載のデータトランザクションレコーディング方法。
【請求項２４】
前記第１ハッシュを再算出するステップと、
マッチング（ｍａｔｃｈ）を決定するために前記生成された第１ハッシュを前記再算出された第２ハッシュと比較するステップと、

をさらに含む、請求項１乃至請求項２３のいずれか一項に記載のデータトランザクションレコーディング方法。
【請求項２５】
前記比較が不成功である場合、追加データトランザクションを取り消すステップをさらに含む、請求項２４に記載のデータトランザクションレコーディング方法。
【請求項２６】
前記第１データトランザクションに対応するシステムハッシュをシステムデバイスに生成するステップをさらに含む、請求項１乃至請求項２５のいずれか一項に記載のデータトランザクションレコーディング方法。
【請求項２７】
第２シードデータを提供するステップは、前記第１シードデータ及び前記第１データトランザクションの前記レコードと前記システムハッシュを結合するステップをさらに含む、請求項２６に記載のデータトランザクションレコーディング方法。
【請求項２８】
前記システムハッシュは、前記システムデバイス上の以前のデータトランザクションのレコードをハッシュした結果である、請求項２６又は請求項２７に記載のデータトランザクションレコーディング方法。
【請求項２９】
第２シードデータを提供するステップは、
ライセンスデバイスからライセンスハッシュを受信するステップと、
前記第２シードデータを提供するために前記第１シードデータ及び前記第１データトランザクションの前記レコードと前記ライセンスハッシュを結合するステップと、
をさらに含む、請求項１乃至請求項２８のいずれか一項に記載のデータトランザクションレコーディング方法。
【請求項３０】
前記ライセンスデバイスにおいて、
前記第１ハッシュを受信するステップと、
ライセンス入力を提供するために前記ライセンスハッシュと前記第１ハッシュを結合するステップと、
前記ライセンス入力をハッシュして第２ライセンスハッシュを生成するステップと、
をさらに含む、請求項２９に記載のデータトランザクションレコーディング方法。
【請求項３１】
第２シードデータを提供するステップは、
ディレクトリデバイスからディレクトリハッシュを受信するステップと、
前記第２シードデータを提供するために、前記第１シードデータ及び前記第１データトランザクションの前記レコードと前記ディレクトリハッシュを結合するステップと、
をさらに含む、請求項１乃至請求項３０のいずれか一項に記載のデータトランザクションレコーディング方法。
【請求項３２】
前記ディレクトリサーバにおいて、
前記第１ハッシュを受信するステップと、
ディレクトリ入力を提供するために前記ディレクトリハッシュと前記第１ハッシュを結合するステップと、
前記ディレクトリ入力をハッシュして第２ディレクトリハッシュを生成するステップと、
をさらに含む、請求項３１に記載のデータトランザクションレコーディング方法。
【請求項３３】
第２シードデータを提供するステップは、
前記第１データトランザクションに対する暗号化キーからキーハッシュを生成するステップと、

前記第２シードデータを提供するために前記第１シードデータ及び前記第１データトランザクションの前記レコードと前記キーハッシュを結合するステップと、
をさらに含む、請求項１乃至請求項３２のいずれか一項に記載のデータトランザクションレコーディング方法。
【請求項３４】
前記暗号化キーは、公開キー又は個人キーを含む、請求項３３に記載のデータトランザクションレコーディング方法。
【請求項３５】
前記第１シードデータ及び前記第１データトランザクションの前記レコードを結合するステップは、前記第１データトランザクションが完了するとすぐに実行される、請求項１乃至請求項３４のいずれか一項に記載のデータトランザクションレコーディング方法。
【請求項３６】
前記メモリは遠隔デバイスに位置する、請求項１乃至請求項３５のいずれか一項に記載のデータトランザクションレコーディング方法。
【請求項３７】
他のデバイスから受信されたハッシュに対応する前記第１ハッシュを前記遠隔デバイスで比較するステップをさらに含む、請求項３６に記載のデータトランザクションレコーディング方法。
【請求項３８】
前記デバイスに接続された他のデバイスに前記第１ハッシュを受信することを予想するよう通知するステップをさらに含む、請求項３６又は請求項３７に記載のデータトランザクションレコーディング方法。
【請求項３９】
前記メモリにハッシュのチェーンを格納するステップをさらに含む、請求項１乃至請求項３８のいずれか一項に記載のデータトランザクションレコーディング方法。
【請求項４０】
送信された前記ハッシュのチェーンに対するアクセスを制限するように構成されたデバイス上に位置する第２メモリに前記ハッシュのチェーンを送信するステップをさらに含む、請求項３９に記載のデータトランザクションレコーディング方法。
【請求項４１】
前記ハッシュのチェーンでハッシュを修正又は削除するステップをさらに含み、
前記ハッシュのチェーンでハッシュを修正又は削除するステップは、
前記ハッシュのチェーンで対象のハッシュを再生成するステップと、
前記レコードが修正されていないかの有無を確認するステップと、
前記再生成されたハッシュをレコーディングするステップと、
前記レコードを修正又は削除するステップと、
前記対象のハッシュの結合及び前記修正および削除されたレコードをハッシュして前記レコードに対する新しいハッシュを生成するステップと、
前記新しいハッシュをレコーディングするステップと、
を含む、請求項３９は請求項４０に記載のデータトランザクションレコーディング方法。
【請求項４２】
前記新しいハッシュを用いてシステムハッシュを生成するステップをさらに含む、請求項４１に記載のデータトランザクションレコーディング方法。
【請求項４３】
第１エンティティに関連するデバイスにおいて、前記デバイスは、請求項１乃至請求項４２のいずれか一項に記載の方法を実行するデバイス。
【請求項４４】
前記デバイスはサーバを含む、請求項４３に記載のデバイス。
【請求項４５】

前記デバイスはユーザデバイスを含む、請求項４３に記載のデバイス。
【請求項４６】
前記ユーザデバイスは、パーソナルコンピュータ、スマートフォン、スマートタブレット又はモノのインターネット（ＩｏＴ）可能デバイスのうち少なくとも１つを含む、請求項４５に記載のデバイス。
【請求項４７】
前記ユーザデバイスは、前記デバイス上のメモリで前記第１ハッシュを格納する、請求項４６に記載のデバイス。
【請求項４８】
前記ユーザデバイスは、該当サーバからオフラインである場合にのみ、前記デバイス上のメモリで前記第１ハッシュを格納する、請求項４７に記載のデバイス。
【請求項４９】
前記デバイスは、前記第２エンティティに関連するデバイスに前記第１ハッシュを送信する、請求項４３乃至請求項４８のいずれか一項に記載のデバイス。
【請求項５０】
前記デバイスは、前記第１データトランザクションの前記レコードの署名及び暗号化されたコピーを前記第２エンティティに関連する前記デバイスに送信し、
前記署名は、前記第１データトランザクションの前記レコードに対する配信先サーバの指示（ｉｎｄｉｃａｔｉｏｎ）を含む、請求項４９に記載のデバイス。
【請求項５１】
前記デバイスは、特定のオフライン公開キーで前記レコードにサインする、請求項５０に記載のデバイス。
【請求項５２】
前記デバイスは、前記デバイスに属するキーで前記レコードにサインする、請求項５０に記載のデバイス。
【請求項５３】
前記配信先サーバのみが前記第１データトランザクションの前記レコードの前記暗号化されたコピーを解読できる、請求項５０乃至請求項５２のいずれか一項に記載のデバイス。
【請求項５４】
前記デバイスが、対応するサーバへの接続を回復するとき、前記デバイスは、前記関連するハッシュ及びそのオフラインデータトランザクションの前記暗号化されたレコードを対応するサーバに送信する、請求項４８乃至請求項５３のいずれか一項に記載のデバイス。
【請求項５５】
前記デバイスは、自身が保有する他のエンティティを含むデータトランザクションのレコードのコピーを前記他のエンティティに対応するサーバへの送信のために自身に対応するサーバに送信する、請求項５４に記載のデバイス。
【請求項５６】
前記送信することは、前記レコードが適用される全てのサーバに前記レコードを受信することを期待して通知することを含む、請求項５５に記載のデバイス。
【請求項５７】
前記デバイスは、前記第１データトランザクションでこの部分を識別するために固有の内部トランザクション番号を生成する、請求項４３乃至請求項５６のいずれか一項に記載のデバイス。
【請求項５８】
ライセンスデバイスであって、
第１エンティティに関連するデバイスから第１ハッシュを受信することであって、前記第１ハッシュは、前記第１エンティティを含むデータトランザクションのヒストリーを含む、前記第１ハッシュを受信すること、

ライセンス入力を提供するためにライセンスハッシュと前記第１ハッシュを結合すること、
前記ライセンス入力をハッシュして第２ライセンスハッシュを生成すること、
メモリに前記第２ライセンスハッシュを格納すること、を行うように構成されたライセンスデバイス。
【請求項５９】
ディレクトリデバイスであって、
第１エンティティに関連するデバイスから第１ハッシュを受信することであって、前記第１ハッシュは、前記第１エンティティを含むデータトランザクションのヒストリーを含む、前記第１ハッシュを受信すること、
ディレクトリ入力を提供するためにディレクトリハッシュと前記第１ハッシュを結合すること、
前記ライセンス入力をハッシュして第２ディレクトリハッシュを生成すること、
メモリに前記第２ディレクトリハッシュを格納すること、を行うように構成されたディレクトリデバイス。
【請求項６０】
実行されるときコンピューティングデバイスが請求項１乃至請求項４２のいずれか一項に記載の方法を実行させる複数のコード部分を含むコンピュータ可読記録媒体。
【請求項６１】
デバイスから第１サービスにアクセスする方法において、
要求サーバに前記デバイスの識別子を提供するステップと、
前記識別子に基づいて前記デバイスが前記第１サービスに対するアクセスを要求することを許可するステップと、
前記デバイスが前記第１サービスが位置する第１ホストサーバから前記第１サービスにアクセスさせるステップであって、前記アクセスは、前記要求サーバを介して行われる、前記第１サービスにアクセスさせるステップと、
を含む方法。
【請求項６２】
前記許可するステップは、前記識別子に基づいて前記ユーザデバイスが前記第１サービスにアクセスするように許可されるかを確認するステップを含む、請求項６１に記載の方法。
【請求項６３】
確認するステップは、前記識別子に基づいて前記ユーザが少なくとも１つの基準（ｃｒｉｔｅｒｉａ）を満足するかを確認するステップを含む、請求項６２に記載の方法。
【請求項６４】
第１基準が前記第１ホストサーバ又は前記要求サーバに格納され、第２基準が他のサーバに位置する、請求項６３に記載の方法。
【請求項６５】
前記許可するステップは、前記要求サーバ及び前記第１ホストサーバ間の通信に対する署名を検証するステップを含む、請求項６１乃至請求項６４のいずれか一項に記載の方法。
【請求項６６】
前記許可するステップは前記要求サーバで実行される、請求項６１乃至請求項６５のいずれか一項に記載の方法。
【請求項６７】
前記許可するステップは、前記要求サーバで前記デバイスが前記第１サービスにアクセスするように以前に許可されたかを決定するステップを含む、請求項６６に記載の方法。
【請求項６８】
前記許可するステップはディレクトリサーバで実行される、請求項６１乃至請求項６５のいずれか一項に記載の方法。

【請求項６９】
前記許可するステップは、前記要求サーバが前記ディレクトリサーバから前記デバイスに対する許可を要求するステップを含む、請求項６８に記載の方法。
【請求項７０】
前記アクセスさせるステップは、前記ディレクトリサーバが前記第１ホストサーバに対する識別子を前記要求サーバに送信するステップを含む、請求項６８又は請求項６９に記載の方法。
【請求項７１】
前記識別子を許可するデータは、前記ディレクトリサーバに格納される、請求項６８乃至請求項７０のいずれか一項に記載の方法。
【請求項７２】
第２サービスに対するアクセスを要求するステップと、
前記識別子に基づいて前記デバイスが前記第２サービスにアクセスすることを許可するステップと、
前記デバイスが前記要求サーバを介して前記第２サービスにアクセスさせるステップと、
をさらに含む、請求項６１乃至請求項７１のいずれか一項に記載の方法。
【請求項７３】
前記第２サービスは前記第１ホストサーバに位置する、請求項７２に記載の方法。
【請求項７４】
前記第２サービスは第２ホストサーバに位置する、請求項７２に記載の方法。
【請求項７５】
前記デバイスが前記第１サービスにアクセスすることを許可するステップは、第１ディレクトリサーバで実行され、
前記ユーザデバイスが前記第２サービスにアクセスすることを許可するステップは、第２ディレクトリサーバで実行される、請求項７２乃至請求項７４のいずれか一項に記載の方法。
【請求項７６】
第３サービスに対するアクセスを要求するステップと、
前記識別子に基づいて前記デバイスが前記第３サービスにアクセスすることを許可するステップと、
前記デバイスが前記第３サービスにアクセスさせるステップと、
をさらに含む、請求項７２乃至請求項７５のいずれか一項に記載の方法。
【請求項７７】
前記第２サービスは、前記第１ホストサーバ、前記第２ホストサーバ又は第３ホストサーバに位置する、請求項７６に記載の方法。
【請求項７８】
前記デバイスが前記第３サービスにアクセスすることを許可するステップは、第３ディレクトリサーバで実行される、請求項７６又は請求項７７に記載の方法。
【請求項７９】
識別子を提供するステップは、前記デバイスが暗号化されたトンネルを介して前記要求サーバと通信するステップを含む、請求項６１乃至請求項７８のいずれか一項に記載の方法。
【請求項８０】
それぞれの個別サーバで受信されるデータをキャッシュするステップをさらに含む、請求項６１乃至請求項７９のいずれか一項に記載の方法。
【請求項８１】
それぞれのホストサーバは、二以上のサービスを提供する、請求項６１乃至請求項８０のいずれか一項に記載の方法。
【請求項８２】

請求項６１乃至請求項８１のいずれか一項に記載の方法を実行するデバイス。
【請求項８３】
前記デバイスは、パーソナルコンピュータ、スマートフォン、スマートタブレット又はモノのインターネットが可能なデバイスのうち少なくとも１つを含む、請求項８２に記載のデバイス。
【請求項８４】
実行されるとき、コンピュータデバイスが請求項６１乃至請求項８１のいずれか一項に記載の方法を実行させる複数のコード部分（ｃｏｄｅ　ｐｏｒｔｉｏｎｓ）を含むコンピュータ可読記録媒体。
【請求項８５】
第１データストアーから第２データストアーに第１データをスイッチングするための要求を提供するステップと、
前記要求に含まれた識別子に基づいて前記第１データストアーの識別子をディレクトリサーバから決定するステップと、
前記第１データストアーから前記第２データストアーに前記第１データをマイグレーションするステップと、
を含むデータマイグレーション方法。
【請求項８６】
前記マイグレーションするステップは、前記ディレクトリサーバにおいて、
前記第２データストアーで前記データに対する開始タイムスタンプを割り当てるステップと、
前記第１データストアーで前記データに対する終了タイムスタンプを割り当てるステップと、
を含む、請求項８５に記載のデータマイグレーション方法。
【請求項８７】
前記終了タイムスタンプの後に、前記第１データストアーを介して前記データにアクセスしようと試みる要求サーバに、前記ディレクトリサーバを介して前記第２データストアーで前記ユーザを検索するように指示するステップをさらに含む、請求項８６に記載のデータマイグレーション方法。
【請求項８８】
前記第１データストアーにおける前記データは、第１アカウント提供者との第１アカウント登録を含み、
前記第２データストアーにおける前記データは、新しいアカウント提供者との第２アカウント登録を含む、請求項８５乃至請求項８７のいずれか一項に記載のデータマイグレーション方法。
【請求項８９】
前記マイグレーションするステップは、前記現在のアカウント提供者から前記新しいアカウント提供者に前記第１アカウント登録に関する情報を送信するステップを含む、請求項８８に記載のデータマイグレーション方法。
【請求項９０】
前記情報は、登録（ｒｅｇｉｓｔｒａｔｉｏｎｓ）、残額（ｂａｌａｎｃｅｓ）、コンフィギュレーション（ｃｏｎｆｉｇｕｒａｔｉｏｎｓ）及び支払い指示（ｐａｙｍｅｎｔ　ｉｎｓｔｒｕｃｔｉｏｎｓ）のうち少なくとも１つを含む、請求項８９に記載のデータマイグレーション方法。
【請求項９１】
マイグレーションするステップは、前記第１登録が前記現在のアカウント提供者から前記新しいアカウント提供者にスイッチされなければならないことを示す認証コード（ａｕｔｈｅｎｔｉｃａｔｉｏｎ　ｃｏｄｅ）を確認するステップを含む、請求項８８乃至請求項９０のいずれか一項に記載のデータマイグレーション方法。
【請求項９２】

前記第１アカウント登録は第１ユーザ・クリデンシャルを含み、
前記第２アカウント登録は第２ユーザ・クリデンシャルを含む、請求項８８乃至請求項９１のいずれか一項に記載のデータマイグレーション方法。
【請求項９３】
前記第１ユーザ・クリデンシャルは第１サーバに登録され、
前記第２ユーザ・クリデンシャルは第２サーバに登録される、請求項９２に記載のデータマイグレーション方法。
【請求項９４】
前記第１アカウント提供者によって前記第１ユーザ・クリデンシャルを用いてユーザに伝えられる通信を受信するステップと、
前記第２ユーザ・クリデンシャルを用いて前記通信を前記第２アカウント提供者にルーティングするステップと、
をさらに含む、請求項９３に記載のデータマイグレーション方法。
【請求項９５】
前記第１クリデンシャルを使用する前記第１登録提供者で作られたデータトランザクションを、前記第２ユーザ・クリデンシャルを使用する前記第２登録提供者に反転させるステップをさらに含む、請求項９３又は請求項９４に記載のデータマイグレーション方法。
【請求項９６】
前記トランザクション時に前記ユーザが前記第１ユーザ・クリデンシャルを使用したことを決定するステップを含む、請求項９５に記載のデータマイグレーション方法。
【請求項９７】
前記通信を送信するサーバは、前記第２ユーザ・クリデンシャルにアクセスするように承認されなければならない、請求項９４乃至請求項９６のいずれか一項に記載のデータマイグレーション方法。
【請求項９８】
前記第１ユーザ・クリデンシャル及び前記第２ユーザ・クリデンシャルは同一である、請求項９２乃至請求項９７のいずれか一項に記載のデータマイグレーション方法。
【請求項９９】
請求項８５乃至請求項９８のいずれか一項に記載の方法を実行するデバイス。
【請求項１００】
前記デバイスは、パーソナルコンピュータ、スマートフォン、スマートタブレット又はモノのインターネットが可能なデバイスのうち少なくとも１つを含む、請求項９９に記載のデバイス。
【請求項１０１】
実行されるとき、コンピュータデバイスが請求項８５乃至請求項９８のいずれか一項に記載の方法を実行させる複数のコード部分（ｃｏｄｅ　ｐｏｒｔｉｏｎｓ）を含むコンピュータ可読記録媒体。
【請求項１０２】
第１エンティティから第２エンティティに第１通信を送信するステップであって、前記第１通信は２つ以上のデータフィールドを含み、それぞれのフィールドは個別ラベルを含む、前記第１通信を送信するステップと、
前記第１エンティティから前記第２エンティティに第２通信を送信するステップであって、前記第２通信は前記２つ以上のデータフィールドを含み、前記第２通信における前記２つ以上のデータフィールドの順は、前記第１通信における前記２つ以上のデータフィールドの順と異なる、前記第２通信を送信するステップと、
を含む通信方法。
【請求項１０３】
ランダムフィールドを前記第２通信に追加するステップをさらに含む、請求項１０２に記載の通信方法。
【請求項１０４】

それぞれのフィールドは２つ以上の特徴を含み、
少なくとも１つのフィールドで２つ以上の特徴のケースをミキシングするステップをさらに含む、請求項１０２又は請求項１０３に記載の通信方法。
【請求項１０５】
前記第２通信を処理する前に、前記第２エンティティによって前記第２通信で前記フィールドを解読及び順序化するステップをさらに含む、請求項１０２乃至請求項１０４のいずれか一項に記載の通信方法。
【請求項１０６】
前記第２エンティティによって処理できないフィールドを廃棄するステップをさらに含む、請求項１０５に記載の通信方法。
【請求項１０７】
前記１エンティティ及び前記第２エンティティのうち少なくとも１つはサーバを含む、請求項１０２乃至請求項１０６のいずれか一項に記載の装置。
【請求項１０８】
前記１エンティティ及び前記第２エンティティのうち少なくとも１つは、パーソナルコンピュータ、スマートフォン、スマートタブレット又はモノのインターネットが可能なデバイスを含む、請求項１０２乃至請求項１０６のいずれか一項に記載の装置。
【請求項１０９】
請求項１０２乃至請求項１０８のいずれか一項に記載の方法を実行するデバイス。
【請求項１１０】
前記デバイスは、パーソナルコンピュータ、スマートフォン、スマートタブレット又はモノのインターネットが可能なデバイスのうち少なくとも１つを含む、請求項１０９に記載のデバイス。
【請求項１１１】
実行されるとき、コンピュータデバイスが請求項１０２乃至請求項１０８のいずれか一項に記載の方法を実行させる複数のコード部分（ｃｏｄｅ　ｐｏｒｔｉｏｎｓ）を含むコンピュータ可読記録媒体。
【請求項１１２】
ＵＳＳＤ（ｕｎｓｔｒｕｃｔｕｒｅｄ　ｓｕｐｐｌｅｍｅｎｔａｒｙ　ｓｅｒｖｉｃｅ　ｄａｔａ）を介した通信方法において、
第１デバイスと第２デバイスとの間のＵＳＳＤセッションを開放するステップと、
前記第１デバイスにおいて前記セッションで通信に対するサイファーテキスト（ｃｙｐｈｅｒ　ｔｅｘｔ）を生成するステップと、
前記第１デバイスで前記サイファーテキストを符号化するステップと、
前記第２デバイスで解読のために前記第１デバイスから前記第２デバイスに前記符号化されたサイファーテキストを送信するステップと、
を含む通信方法。
【請求項１１３】
前記符号化するステップは、前記サイファーテキストを７ビット又は８ビットの文字ストリングに符号化するステップを含む、請求項１１２に記載の通信方法。
【請求項１１４】
前記サイファーテキストの長さが前記ＵＳＳＤセッションで前記許容されたスペースよりも長い場合、
前記サイファーテキストを２つ又は２つ以上の部分に分割するステップと、
前記２つ又は２つ以上の部分を個別的に送信するステップと、
をさらに含む、請求項１１２又は請求項１１３に記載の通信方法。
【請求項１１５】
前記第２デバイスにおける解読のために、前記第２デバイスから前記サイファーテキストの全体に前記パートをリアセンブルするステップを含む、請求項１１４に記載の通信方法。

【請求項１１６】
前記１及び第２デバイスを認証するステップをさらに含む、請求項１１２乃至請求項１１５のいずれか一項に記載の通信方法。
【請求項１１７】
認証するステップは、２つの通信コンピュータアプリケーション間のプライバシー及びデータ無欠性を提供するアルゴリズムを使用するステップを含む、請求項１１６に記載の通信方法。
【請求項１１８】
認証するステップは、ＴＬＳ（ｔｒａｎｓｐｏｒｔ　ｌａｙｅｒｓｅｃｕｒｉｔｙ）を使用するステップを含む、請求項１１７に記載の通信方法。
【請求項１１９】
ＴＬＳを使用するステップは、第１セッションキーを生成するステップを含む、請求項１１８に記載の通信方法。
【請求項１２０】
第２セッションキーを生成するためにＰＡＫＥプロトコルネゴシエーション（ＰＡＫＥ　ｐｒｏｔｏｃｏｌ　ｎｅｇｏｔｉａｔｉｏｎ）を暗号化する前記第１セッションキーを使用するステップと、
前記第２セッションキーを用いて前記第１デバイスと前記第２デバイスとの間の前記セッションで追加通信を暗号化するステップと、
をさらに含む、請求項１１９に記載の通信方法。
【請求項１２１】
請求項１１２乃至請求項１２０のいずれか一項に記載の方法を実行するデバイス。
【請求項１２２】
実行されるとき、コンピュータデバイスが請求項１１２乃至請求項１２０のいずれか一項に記載の方法を実行させる複数のコード部分（ｃｏｄｅ　ｐｏｒｔｉｏｎｓ）を含むコンピュータ可読記録媒体。
【請求項１２３】
第１エンティティに関連する第１デバイスと第２エンティティに関連する第２デバイスとの間の通信方法において、前記第１デバイスで、
第１共有秘密を用いて前記第１デバイス及び前記第２デバイス間の第１ＰＡＫＥセッションを生成するステップと、
前記第２デバイスから登録キー及び第２共有秘密を受信するステップと、
第２ＰＡＫＥセッションを生成するための第３共有秘密を提供するために、前記第１共有秘密、前記登録キー、及び前記第２共有秘密をハッシュするステップと、
を含む通信方法。
【請求項１２４】
前記１エンティティ及び前記第２エンティティを認証するステップをさらに含む、請求項１２３に記載の通信方法。
【請求項１２５】
認証するステップは、２つの通信コンピュータアプリケーション間のプライバシー及びデータ無欠性を提供するアルゴリズムを使用するステップを含む、請求項１２４に記載の通信方法。
【請求項１２６】
前記認証するステップはＴＬＳを使用するステップを含む、請求項１２５に記載の通信方法。
【請求項１２７】
第４共有秘密を用いて前記第１デバイス及び第３デバイス間の第２ＰＡＫＥセッションを生成するステップをさらに含む、請求項１２３乃至請求項１２６のいずれか一項に記載の通信方法。
【請求項１２８】

前記第４共有秘密は、前記第１デバイスのために前記第３デバイスによって生成された認証コードを含む、請求項１２７に記載の通信方法。
【請求項１２９】
前記第１共有秘密は、前記第１デバイスのために前記第２デバイスによって生成された認証コードを含む、請求項１２３乃至請求項１２８のいずれか一項に記載の通信方法。
【請求項１３０】
前記認証コードは、前記第１デバイスのために識別子と共に前記第１デバイスに送信される、請求項１２９に記載の通信方法。
【請求項１３１】
前記識別子は、前記第１デバイスの電話番号又はシリアル番号を含む、請求項１３０に記載の通信方法。
【請求項１３２】
前記第１共有内緒は、前記１エンティティに関連する銀行カードのＰＡＮ（ｐｅｒｓｏｎａｌ　ａｃｃｏｕｎｔ　ｎｕｍｂｅｒ）を含む、請求項１２３乃至請求項１３１のいずれか一項に記載の通信方法。
【請求項１３３】
前記第１共有秘密は、前記１エンティティに関連する銀行カードの符号化されたシリアル番号を含む、請求項１２３乃至請求項１３１のいずれか一項に記載の通信方法。
【請求項１３４】
請求項１２３乃至請求項１３３のいずれか一項に記載の方法を実行するデバイス。
【請求項１３５】
前記デバイスは、パーソナルコンピュータ、スマートフォン、スマートタブレット又はモノのインターネットが可能なデバイスのうち少なくとも１つを含む、請求項１３４に記載のデバイス。
【請求項１３６】
実行されるとき、コンピュータデバイスが請求項１２３乃至請求項１３３のいずれか一項に記載の方法を実行させる複数のコード部分（ｃｏｄｅ　ｐｏｒｔｉｏｎｓ）を含むコンピュータ可読記録媒体。
【請求項１３７】
サービスにアクセスする方法において、
クリデンシャル及び前記クリデンシャルに対するコンテキストを提供するステップと、
前記クリデンシャル及び前記コンテキストに基づいて前記サービスに対するアクセスを認証するステップと、
を含む方法。
【請求項１３８】
前記サービスに対するアクセスを認証するステップは、
前記クリデンシャル及び前記コンテキストのうちの少なくとも一方に基づいてサービスの一部に対するアクセスを認証するステップを含む、請求項１３７に記載の方法。
【請求項１３９】
前記クリデンシャルは、デバイス及び前記デバイスのプライマリユーザ（ｐｒｉｍａｒｙ　ｕｓｅｒ）に関連する第１クリデンシャルを含む、請求項１３７又は請求項１３８に記載の方法。
【請求項１４０】
前記クリデンシャルは、デバイス及び前記デバイスのセカンダリーユーザに関連する第２クリデンシャルをさらに含む、請求項１３９に記載の方法。
【請求項１４１】
前記クリデンシャルに基づいて前記サービスに対するアクセスを認証するステップは、前記第１クリデンシャル及び前記第２クリデンシャルのそれぞれに基づいて前記プライマリユーザ及び前記セカンダリーユーザに対する異なるサービスに対するアクセスを認証するステップを含む、請求項１４０に記載の方法。

【請求項１４２】
前記デバイスは、前記プライマリユーザ及び前記セカンダリーユーザに対する異なる支出限度である前記異なるサービス及び銀行カードを含む、請求項１４１に記載の方法。
【請求項１４３】
前記クリデンシャルは、前記コンテキストに基づいて選択される、請求項１３７乃至請求項１４２のいずれか一項に記載の方法。
【請求項１４４】
前記サービスは、前記コンテキストに基づいて選択された複数のサービスを含む、請求項１３７乃至請求項１４３のいずれか一項に記載の方法。
【請求項１４５】
管理者又はユーザは、前記コンテキスト又はクリデンシャルを修正、追加又は取り消しできる、請求項１３７乃至請求項１４４のいずれか一項に記載の方法。
【請求項１４６】
前記クリデンシャルは、パスワード、ＰＩＮ、及び他の直接認証クリデンシャル（ｄｉｒｅｃｔ　ａｕｔｈｅｎｔｉｃａｔｉｏｎ　ｃｒｅｄｅｎｔｉａｌ）のうち少なくとも１つを含む、請求項１３７乃至請求項１４５のいずれか一項に記載の方法。
【請求項１４７】
前記コンテキストは、前記クリデンシャルを提供するデバイス、前記デバイス上のアプリケーション、前記デバイスが接続されたネットワーク、前記デバイスの地理的位置、及びアクセスされる前記サービスのうち少なくとも１つを含む、請求項１３７乃至請求項１４６のいずれか一項に記載の方法。
【請求項１４８】
請求項１３７乃至請求項１４７のいずれか一項に記載の方法を実行するデバイス。
【請求項１４９】
前記デバイスは、パーソナルコンピュータ、スマートフォン、スマートタブレット又はモノのインターネットが可能なデバイスのうち少なくとも１つを含む、請求項１４８に記載のデバイス。
【請求項１５０】
実行されるとき、コンピュータデバイスが請求項１３７乃至請求項１４７のいずれか一項に記載の方法を実行させる複数のコード部分（ｃｏｄｅ　ｐｏｒｔｉｏｎｓ）を含むコンピュータ可読記録媒体。
【請求項１５１】
コンピュータシステム内の複数のモジュール間の通信方法において、
第１モジュールからプロキシに共有メモリチャネルを伝達するステップと、
前記プロキシから第２モジュールに前記共有メモリチャネルを伝達するステップであって、前記プロキシは、前記コンピュータシステムの前記カーネルをバイパスして前記第１モジュールと前記第２モジュールとの間のデータを送信するハンドオフモジュールを含む、前記共有メモリチャネルを伝達するステップと、
前記第１モジュールから前記第２モジュールにデータを送信するステップと、
を含む通信方法。
【請求項１５２】
複数の要求を前記第１モジュールのバッファメモリでバッチされたメッセージ（ｂａｔｃｈｅｄ　ｍｅｓｓａｇｅ）にバッチするステップと、
前記第２モジュールに送信される前記バッチされたメッセージをキューイングするステップと、
システム機能を許可する少なくとも１つのシステムフラグをセッティングするステップと、
前記第２モジュールで前記少なくとも１つのシステムフラグをチェックするステップと、
前記第２モジュールで前記バッチされたメッセージを処理するステップと、

をさらに含む、請求項１５１に記載の通信方法。
【請求項１５３】
前記第１モジュールと前記第２モジュールとの間の少なくとも１つの共有メモリチャネルを設定するステップをさらに含む、請求項１５１又は請求項１５２に記載の通信方法。
【請求項１５４】
前記少なくとも１つの共有メモリチャネルを介して前記第１モジュールに応答する前記第２モジュールを含む、請求項１５３に記載の通信方法。
【請求項１５５】
前記少なくとも１つの共有メモリチャネルは、前記バッチされたメッセージを受信及びアセンブルし、前記第２モジュールに前記メモリの所有権を渡す、請求項１５３又は請求項１５４に記載の通信方法。
【請求項１５６】
前記少なくとも１つの共有メモリチャネルは、前記コンピュータシステムのネットワークスタック（ｎｅｔｗｏｒｋ　ｓｔａｃｋ）を介してバッチされたメッセージを受信する、請求項１５５に記載の通信方法。
【請求項１５７】
前記少なくとも１つの共有メモリチャネルは、ＨＴＴＰゲートウェイを含む、請求項１５３乃至請求項１５６のいずれか一項に記載の通信方法。
【請求項１５８】
ＨＴＴＰゲートウェイはウェブサービスとして用いられる、請求項１５１乃至請求項１５７のいずれか一項に記載の通信方法。
【請求項１５９】
通信は、パスワード認証されたキー交換プロトコルを使用する、請求項１５１乃至請求項１５８のいずれか一項に記載の通信方法。
【請求項１６０】
前記コンピュータシステムのネットワークスタックでゼロコピーネットワーキング（ｚｅｒｏ－ｃｏｐｙ　ｎｅｔｗｏｒｋｉｎｇ）を使用するステップをさらに含む、請求項１５１乃至請求項１５９のいずれか一項に記載の通信方法。
【請求項１６１】
前記コンピュータシステムのネットワークスタックでユーザモードネットワーキングを使用するステップをさらに含む、請求項１５１乃至請求項１６０のいずれか一項に記載の通信方法。
【請求項１６２】
前記第１モジュールから前記データ送信の前記コンポーネントが単一データストリームに結合され、前記第１モジュールで前記コンポーネントに分離されるようにデータを直列化するステップをさらに含む、請求項１５１乃至請求項１６１のいずれか一項に記載の通信方法。
【請求項１６３】
前記直列化は、各モジュールのエッジで抽象化される、請求項１６２に記載の通信方法。
【請求項１６４】
各モジュールのバッファメモリは、構成可能なバッファリング閾値を有する、請求項１５１乃至請求項１６３のいずれか一項に記載の通信方法。
【請求項１６５】
前記第１モジュール及び前記第２モジュールは、同じコンピューティングデバイス上に位置する、請求項１５１乃至請求項１６４のいずれか一項に記載の通信方法。
【請求項１６６】
前記第１モジュール及び前記第２モジュールは、異なるコンピューティングデバイス上に位置する、請求項１５１乃至請求項１６４のいずれか一項に記載の通信方法。
【請求項１６７】

前記第１モジュールから前記第２モジュールに送信されたデータはバージョンＩＤを運ぶ、請求項１５１乃至請求項１６６のいずれか一項に記載の通信方法。
【請求項１６８】
前記バージョンＩＤが前記第１モジュールから前記第２モジュールに送信された前記データに対して、最新であるかを検証するステップをさらに含む、請求項１６７に記載の通信方法。
【請求項１６９】
前記データのうち任意のデータがアップデートされる場合、前記バージョンＩＤを現在のバージョンに再検証するステップをさらに含む、請求項１６８に記載の通信方法。
【請求項１７０】
前記バージョンＩＤが検証されない場合、前記データ送信は失敗する、請求項１６９に記載の通信方法。
【請求項１７１】
前記第１モジュール及び前記第２モジュールのうち少なくとも１つは少なくとも１つのデータサービスモジュールを含み、
前記コンピュータシステム内の各データ処理は、前記少なくとも１つのデータサービスモジュールを介して実行される、請求項１５１乃至請求項１７０のいずれか一項に記載の通信方法。
【請求項１７２】
前記少なくとも１つのデータサービスモジュールは、コアデータベースストアーによって実現されるデータストアーと通信する、請求項１７１に記載の通信方法。
【請求項１７３】
前記少なくとも１つのデータサービスモジュールは、前記データストアーに直接アクセスする前記コンピュータシステムのコンポーネントである、請求項１７２に記載の通信方法。
【請求項１７４】
前記コアデータベースストアーは、少なくとも１つの分散データベースを含む、請求項１７３に記載の通信方法。
【請求項１７５】
前記少なくとも１つの分散データベースは、別途の読み出し及び記録アクセスチャネルを有する、請求項１７４に記載の通信方法。
【請求項１７６】
前記データストアーは、少なくとも１つの異種データベースにインタフェースを提供する、請求項１７３乃至請求項１７５のいずれか一項に記載の通信方法。
【請求項１７７】
前記データストアーは、複数のインタフェースタイプを提供する、請求項１７３乃至請求項１７６のいずれか一項に記載の通信方法。
【請求項１７８】
前記複数のインタフェースタイプは、少なくとも１つのＳＱＬ（Ｓｔｒｕｃｔｕｒｅｄ Ｑｕｅｒｙ Ｌａｎｇｕａｇｅ）インタフェース、セル及びコラムインタフェース（ｃｅｌｌ ａｎｄ ｃｏｌｕｍｎ ｉｎｔｅｒｆａｃｅ）、文書インタフェース（ｄｏｃｕｍｅｎｔ ｉｎｔｅｒｆａｃｅ）、及び前記コアデータベースストアー上にあるグラフィックインタフェース（ｇｒａｐｈ ｉｎｔｅｒｆａｃｅ）のうち少なくとも１つを含む、請求項１７７に記載の通信方法。
【請求項１７９】
前記データストアーレイヤに対する全ての記録は、１つ又は１つ以上のデータトランザクションの全て又は一部を制御する単一共有モジュールによって管理される、請求項１７６乃至請求項１７８のいずれか一項に記載の通信方法。
【請求項１８０】
少なくとも１つの前記共有モジュールのリダンダント・バックアップ（ｒｅｄｕｎｄａ

ｎｔ　ｂａｃｋｕｐ）を作動させるステップをさらに含む、請求項１７９に記載の通信方法。
【請求項１８１】
全てのデータ変更は、シリアルの速いシーケンス（ｓｅｒｉａｌ　ｒａｐｉｄ　ｓｅｑｕｅｎｃｅ）で前記単一共有モジュールを介して行われる、請求項１７９又は請求項１８０に記載の通信方法。
【請求項１８２】
前記単一共有モジュールは、その自体をデータ・トランザクタ・クラスタ（ｄａｔａ　ｔｒａｎｓａｃｔｏｒ　ｃｌｕｓｔｅｒ）に示すホットバックアップ・リダンダンシー・モデル（ｈｏｔ　ｂａｃｋｕｐ　ｒｅｄｕｎｄａｎｃｙ　ｍｏｄｅｌ）を使用し、
前記データ・トランザクタ・クラスタは、ハイアラーキー（ｈｉｅｒａｒｃｈｙ）で１組のモジュールであり、各モジュールは、マスタモジュールが失敗する場合にデータトランザクションを制御する、請求項１７９乃至請求項１８１のいずれか一項に記載の通信方法。
【請求項１８３】
ドメインによって構成される規則に基づいて、複数のモジュール又は複数のデータストアーにわたってデータを分割するステップをさらに含む、請求項１７１乃至請求項１８２のいずれか一項に記載の通信方法。
【請求項１８４】
データトランザクションのレコード又は親データトランザクション（ｐａｒｅｎｔ　ｄａｔａ　ｔｒａｎｓａｃｔｉｏｎ）のレコードのターゲットデータをハッシュするステップをさらに含む、請求項１８３に記載の通信方法。
【請求項１８５】
前記ハッシュするステップは、データパーティションの数と同じカーディナリティ（ｃａｒｄｉｎａｌｉｔｙ）を有する、請求項１８４に記載の通信方法。
【請求項１８６】
挙げられた地理的領域、名字、及び通貨のうち少なくとも１つによってターゲットデータをハッシュするステップをさらに含む、請求項１８４又は請求項１８５に記載の通信方法。
【請求項１８７】
複数のデータパーティションの全てに前記少なくとも１つのデータサービスモジュールを介して少なくとも１つのデータ送信を行うステップをさらに含む、請求項１７１乃至請求項１８６のいずれか一項に記載の通信方法。
【請求項１８８】
複数のモジュールによって前記少なくとも１つのデータサービスモジュールを介して少なくとも１つのデータ送信を完了するステップをさらに含む、請求項１７１乃至請求項１８７のいずれか一項に記載の通信方法。
【請求項１８９】
前記少なくとも１つのデータサービスモジュール上の少なくとも１つのデータ送信を前記データストアーで複数のデータストレージノード上に保持するステップをさらに含む、請求項１７１乃至請求項１８８のいずれか一項に記載の通信方法。
【請求項１９０】
前記コンピュータシステムは、複数のデータサービスモジュールを含み、
それぞれのデータサービスモジュールは、該当インスタンスに対する全ての前記ホットデータのキャッシュされた表現を含み、イン－メモリ（ｉｎ－ｍｅｍｏｒｙ）またはイン－プロセス（ｉｎ－ｐｒｏｃｅｓｓ）データベースエンジンをホストする、請求項１７１乃至請求項１８９のいずれか一項に記載の通信方法。
【請求項１９１】
前記コンピュータシステムは、複数のデータサービスモジュールを含み、
それぞれのデータサービスモジュールは、複数の異種又は同種データベースエンジンを

含む、請求項１７１乃至請求項１８９のいずれか一項に記載の通信方法。
【請求項１９２】
全てのデータ読み出しが一貫し、対応するデータ記録を反映するように、前記データストアーに対するアクセスの同時性を管理するＭＶＣＣ（Ｍｕｌｔｉｖｅｒｓｉｏｎ Ｃｏｎｃｕｒｒｅｎｃｙ Ｃｏｎｔｒｏｌ）バージョンシステムを使用するステップをさらに含む、請求項１７２乃至請求項１９１のいずれか一項に記載の通信方法。
【請求項１９３】
データレコードが前記データストアーに記録され、任意の後続データトランザクションが前記データレコードにアクセスする前に記録されたことが確認されるように、前記データストアーに対するアクセスの同時性を管理する悲観的一貫性（ｐｅｓｓｉｍｉｓｔｉｃ ｃｏｎｓｉｓｔｅｎｃｙ）を使用するステップをさらに含む、請求項１７２乃至請求項１９１のいずれか一項に記載の通信方法。
【請求項１９４】
前記コンピュータシステムは、アプリケーションレイヤをさらに含み、
前記少なくとも１つのデータサービスモジュールが前記レコードを記録し、前記データ送信を完了することを確認するまで、前記アプリケーションレイヤは、データトランザクションを進めることができない、請求項１７１乃至請求項１９３のいずれか一項に記載の通信方法。
【請求項１９５】
請求項１５１乃至請求項１９４のいずれか一項に記載の方法を実行するコンピューティングデバイス。
【請求項１９６】
実行されるときコンピュータデバイスが請求項１５１乃至請求項１９４のいずれか一項に記載の方法を実行させる複数のコード部分（ｃｏｄｅ ｐｏｒｔｉｏｎｓ）を含むコンピュータ可読記録媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、単一の実施において全てのタイプのトランザクションを大規模かつ安全にリアルタイムで行う方法及びシステムに関する。
【背景技術】
【０００２】
トランザクション処理には、広範囲な分散コンピュータ基盤システム及び、特に、支払いに関するトランザクションを行う多重ランザクション（ｔｒａｎｓａｃｔｏｒｓ）を含むだけではなく、他の金融資産及び取引、物理的アクセス制御、データに対する論理的アクセス、ＩｏＴ（Ｉｎｔｅｒｎｅｔ ｏｆ Ｔｈｉｎｇｓ）を構成する管理、及びモニタリングデバイスなどにおけるトレード（ｔｒａｄｅ）に関する。
【０００３】
最近、トランザクション処理システムを開発するとき、エンジニアは難しいトレードオフ（ｔｒａｄｅ－ｏｆｆｓ）を行わなければならない。これは速度及び回復力、処理量と一貫性、セキュリティーと性能、一貫性と拡張性などの間の選択が含まれる。このようなトレードオフは、常にシステム全体に影響を及ぼす侵害（ｃｏｍｐｒｏｍｉｓｅｓ）を発生させる。支払い処理システムは、このようなトレードオフの効果を示す。それは１秒当たり６００から数万のトランザクションを処理しなければならないこともあるが、ひたすらシステムの業務量において、しばらくの間に追加的な処理のためにそれを部分処理し、詳細を格納するだけであった。これはたびたび紛失したレコードを調整し、トランザクションを重複し、トランザクション時間からトランザクション処理時間までにアカウントが超過して引き落されるという信用問題の露出などといった問題を発生させる。しかし、問題は支払いに制限されない。
【０００４】

全体的なトランザクションがロールバックされて（原子性）、データベースを一貫性のない状態にしておくことができず（一貫性）、互いに干渉できず（分離性）、さらにサーバが再び開始する時にも持続される（耐久性）場合、ＡＣＩＤ（原子性、一貫性、分離性、及び耐久性）は、各データベーストランザクションが成功しなければならないというデータベースに対する一貫性モデルである。

【０００５】

このモデルは、一般的に、既存の銀行支払いネットワーク及びその他の「ビッグデータ」の取引システムのような大規模システムの可用性及び性能要求事項と互換されないものと見なされる。代わりに、このようなシステムは、ＢＡＳＥ一貫性（基本可用性）、ソフト状態、及び最終てきな一貫性に依存する。このモデルは、データベースが窮極的に一貫性のある状態に達することで充分であると主張する。銀行システムは、一貫性のある状態に達するために頻繁に調整チェックし、トランザクションの処理を一時中止しなければならないことから、このモードで作動する。トレードオフは大容量トランザクション処理で行われなければならないという概念は、基本的な形態で分散コンピュータシステムが一貫性、可用性、及びパーティション耐性（ｐａｒｔｉｔｉｏｎ　ｔｏｌｅｒａｎｃｅ）といった３つの全てを同時に提供できない点を示すＣＡＰ整理に明示されている。現在のベスト思慮ソリューション（ｂｅｓｔ　ｐｒａｃｔｉｃｅ　ｓｏｌｕｔｉｏｎｓ）は、新らに出現する現在の要件を満たすには多すぎる制限及びトレードオフを含んでいる。

【０００６】

ＩｏＴによって生成されたデータを調整する方法に対する問題は、エンジニアがネットワーク及びトランザクション処理システムを構築するとき、それらを行う必要があると考えるトレードオフの効果によって発生する。その効果のうちの１つは、共にモノのインターネットを構成しているデバイスとサーバ間の通信に対するセキュリティーの欠如である。他の１つは、デバイスによって収集されたデータが実際に該当のデバイスにより検出された特定のイベントに関わっていることを保障できないことにある。

【０００７】

また、クラウド基盤の情報格納システムは、このようなトレードオフの効果を示すが、多くの場合、究極的な一貫性のみを保障できる数多いサーバ及びシステムが膨大になる。

したがって、既知のシステムにおいて、ＢＡＳＥ一貫性でのみ利益を取得できる大規模システムにＡＣＩＤ一貫性を提供することが求められる。

【発明の概要】

【発明が解決しようとする課題】

【０００８】

上述したように本発明は、現在のトレードオフによって考慮又は制限する必要のないトランザクションを処理する新しい方法に関する。本発明は、既存のシステムよりも数倍も大きい比率でトランザクションをリアルタイムに認証及び処理し、このようなトランザクションをリアルタイムに支払い又は処理及び完了する方法を提供する。

【０００９】

リアルタイム支払いは、金融トランザクションにのみ適用されるものではない。これは即刻的な認証、許可、処理及び完了の一部又は全体から利益を取得できるか、求められる任意のトランザクションに適用される。これはアクセス制御からレコード有効性検査（ｒｅｃｏｒｄｓ　ｖａｌｉｄａｔｉｏｎ）、レコード及び文書交換（ｒｅｃｏｒｄｓ　ａｎｄ　ｄｏｃｕｍｅｎｔ　ｅｘｃｈａｎｇｅ）、命令及び制御指示（ｃｏｍｍａｎｄ　ａｎｄ　ｃｏｎｔｒｏｌ　ｉｎｓｔｒｕｃｔｉｏｎｓ）などに至るまで様々である。

【００１０】

この方法は７種類の主な領域に構成される。

・任意のデータベース製品に極めて高いスケールでＡＣＩＤ準拠のトランザクションを記録するための方法

・単一リアルタイムセッションの範囲内で極めて高いスケールで完全な数学的証明によりマルチプライベート元帳（ｍｕｌｔｉｐｌｅ　ｐｒｉｖａｔｅ　ｌｅｄｇｅｒｓ）にわ

たってレコード認証を伝達するハッシュチェーンの実現
・拡張性問題を引き起こす「ハブ・アンド・スポーク（ｈｕｂ　ａｎｄ　ｓｐｏｋｅ）」アーキテクチャーを実現するものではなく、トランザクションサービス提供者などのメッシュネットワークを支援するディレクトリサービス
・販売者又はユーザデバイスが無線及び１つのトランザクションから次にトランザクションを処理するために使用するアプリケーション（又は、アプリ）をアップデートさせる拡張可能なフレームワーク
・多様な相違なトランザクションタイプ及び共通データベース構造を支援するアプリ間の移動行列として機能するデータサービスレイヤ
・サービス又はデバイスがサービス又は機能のセットにアクセスさせるクリデンシャルのアッドホクセットをアセンブル及び提案する方法
・ＮＦＣ（Ｎｅａｒ　Ｆｉｅｌｄ　Ｃｏｍｍｕｎｉｃａｔｉｏｎｓ）及びＵＳＳＤ（Ｕｎｓｔｒｕｃｔｕｒｅｄ　Ｓｕｐｐｌｅｍｅｎｔａｒｙ　Ｓｅｒｖｉｃｅ　Ｄａｔａ）を含む任意のプロトコルで安全なリアルタイム通信を生成する方法
本発明のシステムは、処理方法のうち固有にトランザクション数が増加することによりゼロ増分コスト（ｚｅｒｏ　ｉｎｃｒｅｍｅｎｔａｌ　ｃｏｓｔ）でリアルタイムトランザクション処理及び完了を達成する方法を提供する。

【課題を解決するための手段】

【００１１】
一実施形態に係る第１エンティティに関連するデバイスでデータトランザクションレコーディング方法は、第１シードデータを決定するステップと、前記第１エンティティと第２エンティティとの間の第１データトランザクションの前記レコードを生成するステップと、少なくとも前記第１シードデータ及び前記第１データトランザクションのレコードを結合して第２シードデータを決定するステップと、前記第２シードデータをハッシュして第１ハッシュを生成するステップ（前記第１ハッシュは、前記第１エンティティを含むデータトランザクションのヒストリーを含む）と、前記第１データトランザクションの前記レコードに対する前記第１ハッシュをメモリに格納するステップとを含むデータトランザクションレコーディング方法が提供される。他の実施形態によれば、前記方法は実行し、第１エンティティに関連するデバイスが提供される。他の実施形態によれば、実行されるときコンピューティングデバイスが前記方法を実行させる複数のコード部分を含むコンピュータで可読記録媒体が提供される。

【００１２】
他の実施形態によれば、第１エンティティに関連するデバイスから第１ハッシュを受信し（前記第１ハッシュは、前記第１エンティティを含むデータトランザクションのヒストリーを含む）、ライセンス入力を提供するためにライセンスハッシュと前記第１ハッシュを結合し、前記ライセンス入力をハッシュして第２ライセンスハッシュを生成し、及びメモリに前記第２ライセンスハッシュを格納するライセンスデバイスを提供する。

【００１３】
他の実施形態によれば、第１エンティティに関連するデバイスから第１ハッシュを受信し（前記第１ハッシュは、前記第１エンティティを含むデータトランザクションのヒストリーを含む）、ディレクトリ入力を提供するためにディレクトリハッシュと前記第１ハッシュを結合し、前記ライセンス入力をハッシュして第２ディレクトリハッシュを生成し、及びメモリに前記第２ディレクトリハッシュを格納するディレクトリデバイスを提供する。

【００１４】
他の実施形態によれば、デバイスから第１サービスにアクセスする方法は、要求サーバに前記デバイスの識別子を提供するステップと、前記識別子に基づいて前記デバイスが前記第１サービスに対するアクセスを要求することを許可するステップと、前記デバイスが前記第１サービスが位置する第１ホストサーバから前記第１サービスにアクセスさせるステップ（前記アクセスは、前記要求サーバを介して行われる）を含むアクセス方法が提供

される。他の実施形態によれば、前記方法を行うデバイスが提供される。他の実施形態によれば、実行されるときコンピュータデバイスが前記方法を実行させる複数のコード部分（ｃｏｄｅ　ｐｏｒｔｉｏｎｓ）を含むコンピュータで可読記録媒体が提供される。

【００１５】

他の実施形態によれば、第１データストアーから第２データストアーに第１データをスイッチングするための要求を提供するステップと、前記要求に含まれた識別子に基づいて前記第１データストアーの識別子をディレクトリサーバから決定するステップと、前記第１データストアーから前記第２データストアーに前記第１データを移行するステップを含むデータ移行方法が提供される。他の実施形態によれば、前記方法を行うデバイスが提供される。他の実施形態によれば、実行されるときコンピュータデバイスが前記方法を実行させる複数のコード部分（ｃｏｄｅ　ｐｏｒｔｉｏｎｓ）を含むコンピュータで可読記録媒体が提供される。

【００１６】

他の実施形態によれば、第１エンティティから第２エンティティに第１通信（前記第１通信は２つ以上のデータフィールドを含み、それぞれのフィールドは、個別ラベルを含む）を送信するステップと、前記第１エンティティから前記第２エンティティに第２通信（前記第２通信は前記２つ以上のデータフィールドを含み、前記第２通信における前記フィールドの順は、前記第１通信における前記フィールドの順と異なる）を送信するステップを含む通信方法が提供される。他の実施形態によれば、前記方法を行うデバイスが提供される。他の実施形態によれば、実行されるときコンピュータデバイスが前記方法を実行させる複数のコード部分（ｃｏｄｅ　ｐｏｒｔｉｏｎｓ）を含むコンピュータで可読記録媒体が提供される。

【００１７】

他の実施形態によれば、ＵＳＳＤ（ｕｎｓｔｒｕｃｔｕｒｅｄ　ｓｕｐｐｌｅｍｅｎｔａｒｙ　ｓｅｒｖｉｃｅ　ｄａｔａ）を通した通信方法において、第１デバイスと第２デバイス間のＵＳＳＤセッションを開放するステップと、前記第１デバイスにおいて前記セッションで通信に対するサイファーテキスト（ｃｙｐｈｅｒ　ｔｅｘｔ）を生成するステップと、前記第１デバイスで前記サイファーテキストを符号化するステップと、前記第２デバイスで解読のために前記第１デバイスから前記第２デバイスに前記符号化されたサイファーテキストを送信するステップとを含む通信方法が提供される。他の実施形態によれば、前記方法を行うデバイスが提供される。他の実施形態によれば、実行されるときコンピュータデバイスが前記方法を実行させる複数のコード部分（ｃｏｄｅ　ｐｏｒｔｉｏｎｓ）を含むコンピュータで可読記録媒体が提供される。

【００１８】

他の実施形態によれば、第１エンティティに関連する第１デバイスと第２エンティティに関連する第２デバイス間の通信方法において、前記第１デバイスで、第１共有秘密を用いて前記第１デバイス及び前記第２デバイス間の第１ＰＡＫＥセッションを生成するステップと、前記第２デバイスから登録キー及び第２共有秘密を受信するステップと、第２ＰＡＫＥセッションを生成するための第３共有秘密を提供するために前記第１共有秘密、前記登録キー、及び前記第２共有秘密をハッシュするステップとを含む通信方法が提供される。他の実施形態によれば、前記方法を行うデバイスが提供される。他の実施形態によれば、実行されるときコンピュータデバイスが前記方法を実行させる複数のコード部分（ｃｏｄｅ　ｐｏｒｔｉｏｎｓ）を含むコンピュータで可読記録媒体が提供される。

【００１９】

他の実施形態によれば、サービスにアクセスする方法において、クリデンシャル及び前記クリデンシャルに対するコンテキストを提供するステップと、前記クリデンシャル及び前記コンテキストに基づいて前記サービスに対するアクセスを認証するステップとを含むアクセス方法が提供される。他の実施形態によれば、前記方法を行うデバイスが提供される。他の実施形態によれば、実行されるときコンピュータデバイスが前記方法を実行させる複数のコード部分（ｃｏｄｅ　ｐｏｒｔｉｏｎｓ）を含むコンピュータで可読記録媒体

が提供される。
【００２０】
他の実施形態によれば、コンピュータシステム内のモジュール間の通信方法において、第１モジュールからプロキシに共有メモリチャネルを伝達するステップと、前記プロキシから第２モジュールに前記共有メモリチャネルを伝達するステップ（前記プロキシは、前記コンピュータシステムの前記カーネルをバイパスして前記第１モジュールと前記第２モジュールとの間のデータを送信するハンドオフモジュールを含む）と、前記第１モジュールから前記第２モジュールにデータを送信するステップとを含む通信方法が提供される。他の実施形態によれば、前記方法を行うコンピューティングデバイスが提供される。他の実施形態によれば、実行されるときコンピュータデバイスが前記方法を実行させる複数のコード部分（ｃｏｄｅ　ｐｏｒｔｉｏｎｓ）を含むコンピュータで可読記録媒体が提供される。
【００２１】
前記第１シードデータは、開始ハッシュを含んでもよい。前記開始ハッシュは、前記第１エンティティを含む以前のデータトランザクションのレコードをハッシュした結果であり得る。前記開始ハッシュは、ランダムハッシュを含んでもよい。前記ランダムハッシュは、前記デバイスからの署名、前記ランダムハッシュが生成された日付及び／又は時間のうちの少なくとも１つを含んでもよい。
【００２２】
第２シードデータを提供するステップは、前記第１シードデータ及び前記第１データトランザクションの前記レコードと第１ゼロ知識証明及び第２ゼロ知識証明を結合するステップをさらに含んでもよい。ここで、前記第１ゼロ知識証明は、前記開始ハッシュが前記第１エンティティに関連する前記以前のデータトランザクションの前記真のハッシュを含むという証明を含んでもよい。前記第２ゼロ知識証明は、第２ハッシュが前記第２エンティティに関連する以前のデータトランザクションの前記真のハッシュを含むという証明を含んでもよい。第２シードデータを提供するステップは、前記第１シードデータ、前記第１データトランザクションの前記レコード、前記第１ゼロ知識証明及び前記第２ゼロ知識証明と第３ゼロ知識証明を結合するステップをさらに含んでもよい。前記第３ゼロ知識証明は、ランダムデータから生成されてもよい。前記第３ゼロ知識証明は、前記第１ゼロ知識証明又は前記第２ゼロ知識証明の繰り返しであってもよい。前記第３ゼロ知識証明は、前記第２ゼロ知識証明に対応する前記第１データトランザクションの第２レコードを用いて構成してもよい。
【００２３】
前記第１データトランザクションは少なくとも２つのステージを含み、第２シードデータを提供するステップは、前記第１データトランザクションの前記第１ステージのレコードと前記第１ゼロ知識証明を結合するステップと、前記第１データトランザクションの前記第２ステージのレコードと前記第２ゼロ知識証明を結合するステップを含んでもよい。第２シードデータを提供するステップは、前記第１データトランザクションの前記第２ステージのレコードから第３ゼロ知識証明を構成するステップと、前記第１データトランザクションの前記第２ステージのレコードと前記第２ゼロ知識証明及び前記第３ゼロ知識証明を結合するステップを含んでもよい。前記第１データトランザクションは少なくとも３つのステージを含み、第２シードデータを提供するステップは、前記第１データトランザクションの前記第３ステージのレコードと前記第１ゼロ知識証明を結合するステップと、前記第１データトランザクションの前記第３ステージのレコードと前記第３ゼロ知識証明を結合するステップをさらに含んでもよい。
【００２４】
前記第１データトランザクションは、少なくとも３つのステージを含んでもよく、第２シードデータを提供するステップは、前記第１データトランザクションの前記第３ステージのレコードと前記第１ゼロ知識証明を結合するステップと、ランダムデータと前記第３ゼロ知識証明を結合するステップをさらに含んでもよい。前記第１データトランザクショ

ンは、少なくとも３つのステージを含んでもよく、第２シードデータを提供するステップは、前記第１データトランザクションの前記第３ステージのレコードと前記第１ゼロ知識証明を結合するステップと、前記第１データトランザクションの第４ステージのレコードと前記第２ゼロ知識証明を結合するステップを含んでもよく、前記第１データトランザクションの前記第４ステージは、前記第１データトランザクションの前記第３ステージの繰り返しであってもよい。

【００２５】

前記第１データトランザクションは、少なくとも３つのステージを含んでもよくて、第２シードデータを提供するステップは、前記第１データトランザクションの前記第３ステージのレコードと第３ゼロ知識証明を結合するステップをさらに含んでもよい。

【００２６】

前記第１ゼロ知識証明は、前記第１エンティティに関連する前記デバイスによって構成され、前記第２ゼロ知識証明は、前記第２エンティティに関連するデバイスによって構成され得る。

【００２７】

前記第１ゼロ知識証明及び前記第２ゼロ知識証明を構成するステップは、キー交換アルゴリズムを使用するステップを含んでもよい。前記キー交換アルゴリズムはＰＡＫＥアルゴリズムを含んでもよい。

【００２８】

前記方法は、前記第２エンティティに関連するデバイスに前記第１ハッシュを送信するステップと、前記第２エンティティに関連するデバイスから第２ハッシュを受信するステップ（前記第２ハッシュは、前記第２エンティティに関連する以前のデータトランザクションのハッシュを含む）と、前記第１パーティー及び前記第２パーティー間の第２データトランザクションのレコードを生成するステップと、前記第１ハッシュ及び前記第２ハッシュと前記第２データトランザクションの前記レコードを結合して第３シードデータを決定するステップと、前記第３シードデータをハッシュして第３ハッシュを生成するステップ（前記第３ハッシュは、前記第１エンティティに関連するデータトランザクションのヒストリー及び前記第２エンティティを含むデータトランザクションのヒストリーを含む）と、前記第２データトランザクションの前記レコードに対する前記第３ハッシュを前記メモリに格納するステップをさらに含んでもよい。

【００２９】

第３シードデータを提供するステップは、前記第２データトランザクションの前記レコード、前記第１ハッシュ及び前記第２ハッシュと第３ゼロ知識証明及び第４ゼロ知識証明を結合するステップをさらに含み、前記第３ゼロ知識証明は、前記第１ハッシュが前記第１データトランザクションの真のハッシュを含むという証明を含み、前記第４ゼロ知識証明は、前記第２ハッシュが前記第２エンティティに関連する前記以前のデータトランザクションの前記真のハッシュを含むという証明を含んでもよい。前記第２エンティティに関連する前記以前のデータトランザクションは前記第１データトランザクションであってもよい。

【００３０】

前記方法は、前記第１エンティティ及び／又は前記第２エンティティの識別子と前記ハッシュのそれぞれを関連づけるステップをさらに含んでもよい。前記方法は、前記第１ハッシュを再算出するステップと、マッチング（ｍａｔｃｈ）を決定するために前記生成された第１ハッシュを前記再算出された第２ハッシュと比較するステップをさらに含んでもよい。前記方法は、前記比較が不成功である場合、追加データトランザクションを取り消すステップをさらに含んでもよい。前記方法は、前記第１データトランザクションに対応するシステムハッシュをシステムデバイスに生成するステップをさらに含んでもよい。

【００３１】

第２シードデータを提供するステップは、前記第１シードデータ及び前記第１データトランザクションの前記レコードと前記システムハッシュを結合するステップをさらに含ん

でもよい。前記システムハッシュは、前記システムデバイス上の以前のデータトランザクションのレコードをハッシュした結果であってもよい。

【００３２】

第２シードデータを提供するステップは、ライセンスデバイスからライセンスハッシュを受信するステップと、前記第２シードデータを提供するために前記第１シードデータ及び前記第１データトランザクションの前記レコードと前記ライセンスハッシュを結合するステップをさらに含んでもよい。

【００３３】

前記方法は、前記ライセンスデバイスで、前記第１ハッシュを受信するステップと、ライセンス入力を提供するために前記ライセンスハッシュと前記第１ハッシュを結合するステップと、前記ライセンス入力をハッシュして第２ライセンスハッシュを生成するステップをさらに含んでもよい。

【００３４】

第２シードデータを提供するステップは、ディレクトリデバイスからディレクトリハッシュを受信するステップと、前記第２シードデータを提供するために前記第１シードデータ及び前記第１データトランザクションの前記レコードと前記ディレクトリハッシュを結合するステップをさらに含んでもよい。

【００３５】

前記方法は、前記ディレクトリサーバで、前記第１ハッシュを受信するステップと、ディレクトリ入力を提供するために前記ディレクトリハッシュと前記第１ハッシュを結合するステップと、前記ディレクトリ入力をハッシュして第２ディレクトリハッシュを生成するステップをさらに含んでもよい。

【００３６】

第２シードデータを提供するステップは、前記第１データトランザクションに対する暗号化キーからキーハッシュを生成するステップと、前記第２シードデータを提供するために前記第１シードデータ及び前記第１データトランザクションの前記レコードと前記キーハッシュを結合するステップをさらに含んでもよい。前記暗号化つける公開キー又は個人キーを含んでもよい。

【００３７】

前記第１シードデータ及び前記第１データトランザクションの前記レコードを結合するステップは、前記第１データトランザクションが完了するとすぐに実行されてもよい。前記メモリは遠隔デバイスに位置してもよい。前記方法は、他のデバイスから受信されたハッシュに対応する前記第１ハッシュを前記遠隔デバイスで比較するステップをさらに含んでもよい。前記方法は、前記デバイスに接続された他のデバイスに前記第１ハッシュを受信することを予想するよう通知するステップをさらに含んでもよい。

【００３８】

前記方法は、前記メモリにハッシュチェーンを格納するステップをさらに含んでもよい。前記方法は、送信された前記ハッシュチェーンに対するアクセスを制限するように構成されたデバイスに位置する第２メモリに前記ハッシュチェーンを送信するステップをさらに含んでもよい。前記方法は、前記ハッシュチェーンでハッシュを修正又は削除するステップをさらに含み、前記ハッシュチェーンでハッシュを修正又は削除するステップは、前記ハッシュチェーンで対象のハッシュを再生成するステップと、前記レコードが修正されていないかの有無を確認するステップと、前記再生成されたハッシュをレコーディングするステップと、前記レコードを修正又は削除するステップと、前記対象のハッシュの結合及び前記修正及び削除されたレコードをハッシュして前記レコードに対する新しいハッシュを生成するステップと、前記新しいハッシュをレコーディングするステップを含んでもよい。前記方法は、前記新しいハッシュを用いてシステムハッシュを生成するステップをさらに含んでもよい。

【００３９】

前記デバイスはサーバを含んでもよい。前記デバイスはユーザデバイスを含んでもよい

。前記ユーザデバイスは、パーソナルコンピュータ、スマートフォン、スマートタブレット又はモノのインターネット（ＩｏＴ）可能デバイスのうちの少なくとも１つを含んでもよい。前記ユーザデバイスは前記デバイス上のメモリで前記第１ハッシュを格納してもよい。前記ユーザデバイスは、該当サーバからオフラインである場合にのみ、前記デバイス上のメモリで前記第１ハッシュを格納してもよい。前記デバイスは、前記第２エンティティに関連するデバイスに前記第１ハッシュを送信してもよい。前記デバイスは、前記第１データトランザクションの前記レコードに署名し、暗号化されたコピーを前記第２エンティティに関連する前記デバイスに送信し、前記署名は、前記第１データトランザクションの前記レコードに対する配信先サーバの表示を含んでもよい。前記デバイスは、特定のオフライン公開キーで前記レコードに署名してもよい。前記デバイスは、前記デバイスに属するキーで前記レコードに署名してもよい。前記配信先サーバのみが前記第１データトランザクションの前記レコードの前記暗号化されたコピーを解読してもよい。前記デバイスが対応するサーバと接続を回復するとき、前記デバイスは、前記関連するハッシュ及びそのオフラインデータトランザクションの前記暗号化されたレコードを対応するサーバに送信してもよい。前記デバイスは、自身が保有する他のエンティティを含むデータトランザクションのレコードのコピーを前記他のエンティティに対応するサーバへの送信のために自身に対応するサーバに送信してもよい。前記送信は、前記レコードが適用される全てのサーバに前記レコードを受信することを期待するよう通知することを含んでもよい。前記デバイスは、前記第１データトランザクションでこれの部分を識別するために固有の内部トランザクション番号を生成してもよい。

【００４０】

前記許可するステップは、前記識別子に基づいて前記ユーザデバイスが前記第１サービスにアクセスするように許可されるかを確認するステップを含んでもよい。前記確認するステップは、前記識別子に基づいて前記ユーザが少なくとも１つの基準（ｃｒｉｔｅｒｉａ）を満足するかを確認するステップを含んでもよい。第１基準が前記第１ホストサーバ又は前記要求サーバに格納され、第２基準が他のサーバに位置してもよい。前記許可するステップは、前記要求サーバ及び前記第１ホストサーバ間の通信に対する署名を検証するステップを含んでもよい。

【００４１】

前記許可するステップは、前記要求サーバで実行されてもよい。前記許可するステップは、前記要求サーバで前記デバイスが前記第１サービスにアクセスするように以前に許可されたかを決定するステップを含んでもよい。

【００４２】

前記許可するステップは、ディレクトリサーバで実行されてもよい。前記許可するステップは、前記要求サーバが前記ディレクトリサーバから前記デバイスに対する許可を要求するステップを含んでもよい。前記アクセスさせるステップは、前記ディレクトリサーバが前記第１ホストサーバに対する識別子を前記要求サーバに送信するステップを含んでもよい。前記識別子を許可するデータは、前記ディレクトリサーバに格納されてもよい。

【００４３】

前記方法は、第２サービスに対するアクセスを要求するステップと、前記識別子に基づいて前記デバイスが前記第２サービスにアクセスすることを許可するステップと、前記デバイスが前記要求サーバを介して前記第２サービスにアクセスさせるステップをさらに含んでもよい。前記第２サービスは、前記第１ホストサーバに位置してもよい。前記第２サービスは、第２ホストサーバに位置してもよい。

【００４４】

前記デバイスが前記第１サービスにアクセスすることを許可するステップは、第１ディレクトリサーバで実行され、前記ユーザデバイスが前記第２サービスにアクセスすることを許可するステップは、第２ディレクトリサーバで実行されてもよい。

【００４５】

前記方法は、第３サービスに対するアクセスを要求するステップと、前記識別子に基づ

いて前記デバイスが前記第３サービスにアクセスすることを許可するステップと、前記デバイスが前記第３サービスにアクセスさせるステップをさらに含んでもよい。

【００４６】

前記第２サービスは、前記第１ホストサーバ、前記第２ホストサーバ又は第３ホストサーバに位置してもよい。前記デバイスが前記第３サービスにアクセスすることを許可するステップは、第３ディレクトリサーバで実行されてもよい。

【００４７】

識別子を提供するステップは、前記デバイスが暗号化されたトンネルを介して前記要求サーバと通信するステップを含んでもよい。前記方法は、それぞれの個別サーバで受信されるデータをキャッシュするステップをさらに含んでもよい。それぞれのホストサーバは二以上のサービスを提供してもよい。

【００４８】

前記デバイスは、パーソナルコンピュータ、スマートフォン、スマートタブレット又はモノのインターネットが可能なデバイスのうちの少なくとも１つを含んでもよい。

前記移行するステップは、前記ディレクトリサーバで、前記第２データストアーで前記データに対する開始タイムスタンプを割り当てるステップと、前記第１データストアーで前記データに対する終了タイムスタンプを割り当てるステップを含んでもよい。

【００４９】

前記方法は、前記終了タイムスタンプの後に前記第１データストアーを介して前記データにアクセスしようと試みる要求サーバに前記ディレクトリサーバを介して前記第２データストアーで前記ユーザを検索するように指示するステップをさらに含んでもよい。前記第１データストアーにおける前記データは第１アカウント提供者との第１アカウント登録を含んでもよく、前記第２データストアーにおける前記データは新しいアカウント提供者との第２アカウント登録を含んでもよい。前記移行するステップは、前記現在のアカウント提供者から前記新しいアカウント提供者に前記第１アカウント登録に関する情報を送信するステップを含んでもよい。前記情報は、登録（ｒｅｇｉｓｔｒａｔｉｏｎｓ）、残高ｂａｌａｎｃｅｓ）、コンフィギュレーション（ｃｏｎｆｉｇｕｒａｔｉｏｎｓ）及び／又は支払い指示（ｐａｙｍｅｎｔ　ｉｎｓｔｒｕｃｔｉｏｎｓ）のうちの少なくとも１つを含んでもよい。前記移行するステップは、前記第１登録が前記現在のアカウント提供者から前記新しいアカウント提供者にスイッチされることを示す認証コード（ａｕｔｈｅｎｔｉｃａｔｉｏｎ　ｃｏｄｅ）を確認するステップを含んでもよい。前記第１アカウント登録は第１ユーザ・クリデンシャルを含んでもよく、前記第２アカウント登録は第２ユーザ・クリデンシャルを含んでもよい。前記第１ユーザ・クリデンシャルは第１サーバに登録されてもよく、前記第２ユーザ・クリデンシャルは第２サーバに登録されてもよい。前記方法は、前記第１アカウント提供者によって前記第１ユーザ・クリデンシャルを用いてユーザに伝えられる通信を受信するステップと、前記第２ユーザ・クリデンシャルを用いて前記通信を前記第２アカウント提供者にルーティングするステップをさらに含んでもよい。前記方法は、前記第１クリデンシャルを使用する前記第１登録提供者で作られたデータトランザクションを前記第２ユーザ・クリデンシャルを使用する前記第２登録提供者に反転させるステップをさらに含んでもよい。前記方法は、前記データトランザクション時に前記ユーザが前記第１ユーザ・クリデンシャルを使用したことを決定するステップをさらに含んでもよい。前記通信を送信するサーバは、前記第２ユーザ・クリデンシャルにアクセスするように承認されなければならない。

【００５０】

前記デバイスは、パーソナルコンピュータ、スマートフォン、スマートタブレット又はモノのインターネットが可能なデバイスのうちの少なくとも１つを含んでもよい。

前記方法は、ランダムフィールドを前記第２通信に追加するステップをさらに含んでもよい。それぞれのフィールドは２つ以上の特徴を含み、前記方法は、少なくとも１つのフィールドで特徴のケースをミキシングするステップをさらに含んでもよい。

【００５１】

前記方法は、前記第２通信を処理する前に、前記第２エンティティによって前記第２通信で前記フィールドを解読及び順序化するステップをさらに含んでもよい。前記方法は、前記第２エンティティによって処理できないフィールドを廃棄するステップをさらに含んでもよい。前記第１エンティティ及び前記第２エンティティのうちの少なくとも１つはサーバを含んでもよい。前記第１エンティティ及び前記第２エンティティのうちの少なくとも１つは、パーソナルコンピュータ、スマートフォン、スマートタブレット又はモノのインターネットが可能なデバイスを含んでもよい。前記デバイスは、パーソナルコンピュータ、スマートフォン、スマートタブレット又はモノのインターネットが可能なデバイスのうちの少なくとも１つを含んでもよい。

【００５２】

前記符号化するステップは、前記サイファーテキストを７ビット又は８ビット文字ストリングに符号化するステップを含んでもよい。前記方法は、前記サイファーテキストの前記長さが前記ＵＳＳＤセッションで前記許されたスペースよりも長い場合、前記サイファーテキストを２つ又は２つ以上の部分に分割ステップと、前記２つ又は２つ以上の部分を個別的に送信するステップをさらに含んでもよい。前記解読は、前記第２デバイスから前記全体サイファーテキストに前記２つ又は２つ以上の部分をリアセンブルするステップをさらに含んでもよい。

【００５３】

前記方法は、前記第１及び第２デバイスを認証するステップをさらに含んでもよい。前記認証するステップは、２つの通信コンピュータアプリケーション間のプライバシー及びデータ無欠性を提供するアルゴリズムを使用するステップを含んでもよい。前記認証するステップは、ＴＬＳ（ｔｒａｎｓｐｏｒｔ　ｌａｙｅｒｓｅｃｕｒｉｔｙ）を使用するステップを含んでもよい。ＴＬＳを使用するステップは第１セッションキーを生成するステップを含んでもよい。

【００５４】

前記方法は、第２セッションキーを生成するためにＰＡＫＥプロトコルネゴシエーション（ＰＡＫＥ　ｐｒｏｔｏｃｏｌ　ｎｅｇｏｔｉａｔｉｏｎ）を暗号化する前記第１セッションキーを使用するステップと、前記第２セッションキーを用いて前記第１デバイスと前記第２デバイス間の前記セッションで追加通信を暗号化するステップをさらに含んでもよい。

【００５５】

前記方法は、前記第１エンティティ及び前記第２エンティティを認証するステップをさらに含んでもよい。前記認証するステップは、２つの通信コンピュータアプリケーション間のプライバシー及びデータ無欠性を提供するアルゴリズムを使用するステップを含んでもよい。前記認証するステップは、ＴＬＳを使用するステップを含んでもよい。前記方法は、第４共有秘密を用いて前記第１デバイス及び第３デバイス間の第２ＰＡＫＥセッションを生成するステップをさらに含んでもよい。前記第４共有秘密は、前記第１デバイスのために前記第３デバイスによって生成された認証コードを含んでもよい。

【００５６】

前記第１共有秘密は、前記第１デバイスのために前記第２デバイスによって生成された認証コードを含んでもよい。前記認証コードは、前記第１デバイスのために識別子と共に前記第１デバイスに送信されてもよい。前記識別子は、前記第１デバイスのモバイル番号又はシリアル番号を含んでもよい。前記第１共有秘密は、前記第１エンティティに関連する銀行カードのＰＡＮ（ｐｅｒｓｏｎａｌ　ａｃｃｏｕｎｔ　ｎｕｍｂｅｒ）を含んでもよい。前記第１共有秘密は、前記第１エンティティに関連する銀行カードの符号化されたシリアル番号を含んでもよい。

【００５７】

前記デバイスは、パーソナルコンピュータ、スマートフォン、スマートタブレット又はモノのインターネットが可能なデバイスのうちの少なくとも１つを含んでもよい。

前記サービスに対するアクセスを認証するステップは、前記クリデンシャル及び／又は

前記コンテキストに基づいてサービスの一部に対するアクセスを認証するステップを含んでもよい。前記クリデンシャルは、デバイス及び前記デバイスのプライマリユーザ（ｐｒｉｍａｒｙ　ｕｓｅｒ）に関する第１クリデンシャルを含んでもよい。前記クリデンシャルは、デバイス及び前記デバイスのセカンダリーユーザに関する第２クリデンシャルをさらに含んでもよい。前記クリデンシャルに基づいて前記サービスに対するアクセスを認証するステップは、前記第１クリデンシャル及び前記第２クリデンシャルのそれぞれに基づいて前記プライマリユーザ及び前記セカンダリーユーザに対する異なるサービスに対するアクセスを認証するステップを含んでもよい。前記デバイスは、前記プライマリユーザ及び前記セカンダリーユーザに対する異なる支出限度である前記異なるサービス及び銀行カードを含んでもよい。前記クリデンシャルは、前記コンテキストに基づいて選択されてもよい。前記サービスは、前記コンテキストに基づいて選択された複数のサービスを含んでもよい。管理者又はユーザは、前記コンテキスト又はクリデンシャルを修正、追加又は取り消してもよい。前記クリデンシャルは、パスワード、ＰＩＮ、及び／又は他の直接認証クリデンシャル（ｄｉｒｅｃｔ　ａｕｔｈｅｎｔｉｃａｔｉｏｎ　ｃｒｅｄｅｎｔｉａｌ）のうちの少なくとも１つを含んでもよい。前記コンテキストは、前記クリデンシャルを提供するデバイス、前記デバイス上のアプリケーション、前記デバイスが接続されたネットワーク、前記デバイスの地理的位置、及び／又はアクセスされる前記サービスのうちの少なくとも１つを含んでもよい。

【００５８】

前記デバイスは、パーソナルコンピュータ、スマートフォン、スマートタブレット又はモノのインターネットが可能なデバイスのうちの少なくとも１つを含んでもよい。

前記方法は、複数の要求を前記第１モジュールのバッファメモリでバッチされたメッセージにバッチするステップと、前記第２モジュールに送信される前記バッチされたメッセージをキューイングするステップと、システム機能を許可する少なくとも１つのシステムフラグをセッティングするステップと、前記第２モジュールで前記少なくとも１つのシステムフラグをチェックするステップと、前記第２モジュールで前記バッチされたメッセージを処理するステップをさらに含んでもよい。

【００５９】

前記方法は、前記第１モジュールと前記第２モジュールとの間の少なくとも１つの共有メモリチャネルを設定するステップをさらに含んでもよい。前記方法は、前記少なくとも１つの共有メモリチャネルを介して前記第１モジュールに応答する前記第２モジュールを含んでもよい。前記少なくとも１つの共有メモリチャネルは、前記バッチされたメッセージを受信及びアセンブルし、前記第２モジュールに前記メモリの所有権を渡してもよい。前記少なくとも１つの共有メモリチャネルは、前記コンピュータシステムのネットワークスタックを介してバッチされたメッセージを受信してもよい。前記少なくとも１つの共有メモリチャネルは、ＨＴＴＰゲートウェイを含んでもよい。前記ＨＴＴＰゲートウェイは、ウェブサービスとして使用されてもよい。

【００６０】

通信は、パスワード認証されたキー交換プロトコルを使用してもよい。前記方法は、前記コンピュータシステムのネットワークスタックでゼロコピーネットワーキング（ｚｅｒｏ－ｃｏｐｙ　ｎｅｔｗｏｒｋｉｎｇ）を使用するステップをさらに含んでもよい。前記方法は前記コンピュータシステムのネットワークスタックでユーザモードネットワーキングを使用するステップをさらに含んでもよい。

【００６１】

前記方法は、前記第１モジュールから前記データ送信の前記コンポーネントが単一データストリームに結合し、前記第１モジュールで前記コンポーネントに分離されるようにデータを直列化するステップをさらに含んでもよい。前記直列化は、各モジュールのエッジで抽象化されてもよい。

【００６２】

各モジュールのバッファメモリは、構成可能なバッファリング閾値を有してもよい。前

記第１モジュール及び前記第２モジュールは同じコンピューティングデバイス上に位置してもよい。前記第１モジュール及び前記第２モジュールは、異なるコンピューティングデバイス上に位置してもよい。

【００６３】

前記第１モジュールから前記第２モジュールに送信されたデータはバージョンＩＤ（ｖｅｒｓｉｏｎ　ＩＤ）を運んでもよい。前記方法は、前記バージョンＩＤが前記第１モジュールから前記第２モジュールに送信された前記データに対して最新であるかを検証するステップをさらに含んでもよい。前記方法は、前記データのうち任意のデータがアップデートされる場合、前記バージョンＩＤを現在のバージョンに再検証するステップをさらに含んでもよい。前記バージョンＩＤが検証されない場合、前記データ送信は失敗することがある。

【００６４】

前記第１モジュール及び前記第２モジュールのうちの少なくとも１つは少なくとも１つのデータサービスモジュールを含んでもよく、前記コンピュータシステム内のそれぞれのデータ処理は、前記少なくとも１つのデータサービスモジュールを介して実行されてもよい。前記少なくとも１つのデータサービスモジュールは、コアデータベースストアーによって実現されるデータストアーと通信してもよい。前記少なくとも１つのデータサービスモジュールは、前記データストアーに直接アクセスする前記コンピュータシステムのコンポーネントであってもよい。前記コアデータベースストアーは、少なくとも１つの分散データベースを含んでもよい。前記少なくとも１つの分散データベースは、別途の読み出し及びレコードアクセスチャネルを有してもよい。前記データストアーは、少なくとも１つの異種データベースにインタフェースを提供してもよい。前記データストアーは、複数のインタフェースタイプを提供してもよい。前記複数のインタフェースタイプは、少なくとも１つのＳＱＬ（Ｓｔｒｕｃｔｕｒｅｄ　Ｑｕｅｒｙ　Ｌａｎｇｕａｇｅ）インタフェース、セル及びコラムインタフェース（ｃｅｌｌ　ａｎｄ　ｃｏｌｕｍｎ　ｉｎｔｅｒｆａｃｅ）、文書インタフェース（ｄｏｃｕｍｅｎｔ　ｉｎｔｅｒｆａｃｅ）、及び前記コアデータベースストアー上にあるグラフィックインタフェース（ｇｒａｐｈ　ｉｎｔｅｒｆａｃｅ）のうちの少なくとも１つを含んでもよい。前記データストアーレイヤに対する全てのレコードは、１つ又は１つ以上のデータトランザクションの全て又は一部を制御する単一共有モジュールによって管理されてもよい。

【００６５】

前記方法は、少なくとも１つの前記共有モジュールのリダンダント・バックアップを作動させるステップをさらに含んでもよい。全てのデータ変更は、シリアルの高速シーケンス（ｓｅｒｉａｌ　ｒａｐｉｄ　ｓｅｑｕｅｎｃｅ）で前記単一共有モジュールを介して行われてもよい。前記単一共有モジュールは、それ自体をデータトランザクタククラスタ（ｄａｔａ　ｔｒａｎｓａｃｔｏｒ　ｃｌｕｓｔｅｒ）に示すホットバックアップリダンダンシーモデル（ｈｏｔ　ｂａｃｋｕｐ　ｒｅｄｕｎｄａｎｃｙ　ｍｏｄｅｌ）を使用し、前記データトランザクタククラスタは、ハイアラーキー（ｈｉｅｒａｒｃｈｙ）でモジュールのセットであり、各モジュールは、マスタモジュールが失敗する場合にデータトランザクションを制御してもよい。前記方法は、ドメインによって構成される規則に基づいて、モジュール又はデータストアーにわたってデータを分割するステップをさらに含んでもよい。前記方法は、データトランザクションのレコード又は親データトランザクション（ｐａｒｅｎｔ　ｄａｔａ　ｔｒａｎｓａｃｔｉｏｎ）のレコードのターゲットデータをハッシュするステップをさらに含んでもよい。前記ハッシュするステップは、データパーティションの数と同じカーディナリティ（ｃａｒｄｉｎａｌｉｔｙ）を有してもよい。前記方法は、挙げられた地理的領域、名字及び／又は通貨のうちの少なくとも１つによってターゲットデータをハッシュするステップをさらに含んでもよい。

【００６６】

前記方法は、複数のデータパーティションにわたって前記少なくとも１つのデータサービスモジュールを介して少なくとも１つのデータ送信を行うステップをさらに含んでもよ

い。前記方法は、多重モジュールによって前記少なくとも１つのデータサービスモジュールを介して少なくとも１つのデータ送信を完了するステップをさらに含んでもよい。前記方法は、前記少なくとも１つのデータサービスモジュール上の少なくとも１つのデータ送信を前記データストアーで複数のデータストレージノード上に保持するステップをさらに含んでもよい。

【００６７】

前記コンピュータシステムは、複数のデータサービスモジュールを含んでもよく、それぞれのデータサービスモジュールは、該当インスタンスに対する全ての前記ホットデータのキャッシュされた表現を含み、イン－メモリ（ｉｎ－ｍｅｍｏｒｙ）／イン－プロセス（ｉｎ－ｐｒｏｃｅｓｓ）データベースエンジンをホストしてもよい。前記コンピュータシステムは複数のデータサービスモジュールを含んでもよく、それぞれのデータサービスモジュールは、複数の異種又は同種データベースエンジンを含んでもよい。

【００６８】

前記方法は、正確に全てのデータ読み出しが一貫し、対応するデータレコードを反映するように、前記データストアーに対するアクセスの同時性を管理するＭＶＣＣ（Ｍｕｌｔｉｖｅｒｓｉｏｎ Ｃｏｎｃｕｒｒｅｎｃｙ Ｃｏｎｔｒｏｌ）バージョンシステムを使用するステップをさらに含んでもよい。前記方法は、データレコードが前記データストアーにレコードされ、任意の後続データトランザクションが前記データレコードにアクセスする前にレコードされたことが確認されなければならないように、前記データストアーに対するアクセスの同時性を管理する悲観的一貫性（ｐｅｓｓｉｍｉｓｔｉｃ ｃｏｎｓｉｓｔｅｎｃｙ）を使用するステップをさらに含んでもよい。

【００６９】

前記コンピュータシステムは、アプリケーションレイヤをさらに含んでもよく、前記少なくとも１つのデータサービスモジュールが前記レコードをレコードし、前記データ送信を完了することを確認するまで、前記アプリケーションレイヤはデータトランザクションを進めることができない。

【００７０】

第１実施形態ないし第２６実施形態の全ての選択的な特徴は、必要な部分のみを変更し、他の全ての実施形態に関連する。説明された実施形態の変形が想定され、例えば、全ての開示された実施形態の特徴は任意の方式により組み合わせられてもよい。

【図面の簡単な説明】

【００７１】

【図１】Ｔｅｒｅｏｎのモジュラー概念を示す。

【図２】Ｔｅｒｅｏｎシステムアーキテクチャーの例を示す。

【図２ａ】Ｔｅｒｅｏｎがサービス及びデバイスを機能領域及びコンテキスト、デバイス、コンポーネント及びプロトコルで抽象化する方法を示す。

【図３】仲介者プロキシを通したＴＬＳ接続を介して開始された通信を示す。

【図４】プロキシメモリで共有メモリ及びメッセージ伝達の使用を示す。

【図４ａ】共有メモリ及びセマフォハンドオーバーモジュールを示す。

【図５】４つのアカウントを含むハッシュチェーンを示す。

【図６】同一システム上の２つのアカウントを含むハッシュチェーンを示す。

【図６ａ】トランザクションステップがインターリビングする同一システム上の３つのアカウントを含むハッシュチェーンを示す。

【図７】ライセンスハッシュの樹枝状特性（ｄｅｎｄｒｉｔｉｃ ｎａｔｕｒｅ）を示す。

【図８】しばらくの間にオフライン状態になる４つのデバイスを含むハッシュチェーンを示す。

【図９】２つのサーバによって実現された逆ルックアップ機能を示す。

【図１０】Ｔｅｒｅｏｎサーバ間の通信設定を示す。

【図１１】ユーザが他のサーバに移動する通信を示す。

【図１２】ディレクトリサービスが要求サーバを２つの他のサーバに接続できる方法を示す。
【図１３】多角的なクリデンシャルを構成するためにサーバが３つのサーバからクリデンシャルを取得しなければならないケースを示す。
【図１４】銀行とユーザとの関係を示す。
【図１５】アカウントが振込されるプロセスを示す。
【図１６】登録されたモバイル番号が変更されるプロセスを示す。
【図１７】２つの貨幣にアクセスするために予め登録されたモバイル番号の保持を示す。
【図１７ａ】それぞれの通貨が別途のサーバ上にある２つの通貨にアクセスするために予め登録されたモバイル番号の保持を示す。
【図１８】ワークフローを示す。
【図１９】代案的なワークフローを示す。
【図２０】代案的なワークフローを示す。
【図２１】例示的なコンピューティングシステムを示す。
【発明を実施するための形態】
【００７２】
本発明の実施形態は同じ部分を示すために同じ参照番号が用いられた添付図面を参照して例として説明される。
Ｔｅｒｅｏｎは、電子トランザクション処理及び認証エンジンである。これはモバイル及び電子支払い処理システムで実現される。また、これはＩｏＴ通信システムの一部として他の実現で使用され得る。
【００７３】
Ｔｅｒｅｏｎは、任意のＩＰ（ｉｎｔｅｒｎｅｔ　ｐｒｏｔｏｃｏｌ）支援デバイス及びこのようなＩＰ支援デバイスと相互作用できる任意のデバイスに対するトランザクション機能を提供する。各デバイスは、固有なＩＤを有する。Ｔｅｒｅｏｎの使用事例は、ＩｏＴデバイスから医療記録アクセス及び管理、モバイル、支払い端末又はＡＴＭ（Ａｕｔｏｍａｔｅｄ　Ｔｅｌｌｅｒ　Ｍａｃｈｉｎ）のような、よく見られる支払いまで様々である。初期の実現例において、Ｔｅｒｅｏｎは、モバイル、カード、ＰＯＳ（ｐｏｉｎｇ－ｏｆ－ｓａｌｅ）端末及び固有参照ＩＤを支援する。Ｔｅｒｅｏｎは、消費者及び販売者が支払い、支払いの受領、資金の送金、資金の受領、返金、返金の受領、資金の引き出し、アカウントデータを確認し、過去のトランザクションのミニ明細書の表示を可能にするために必要な機能を提供する。Ｔｅｒｅｏｎは、通貨間及び国を越えたトランザクションを支援する。したがって、消費者は、１つの通貨でアカウントを保有できるが、例えば、別の通貨で振込みすることができる。
【００７４】
Ｔｅｒｅｏｎの初期の具現において、最終のユーザが特定のトランザクションを実行できるかどうかは、その時点で使用していたアプリケーションによって異なる。販売者又は販売者の端末は、一部のトランザクションを開始することができる一方、消費者デバイスは他のものを開始することができる。
【００７５】
Ｔｅｒｅｏｎが支払いを処理するために用いられる場合、トランザクションは次のようなモードに細分化される。支払い、支払いの受領、モバイル消費者対モバイル販売者、モバイル消費者対オンライン販売者ポータル、顧客のないモバイル消費者対モバイル販売者、アカウントポータル内で消費者アカウント対販売者アカウント、ＮＦＣ－Ｔｅｒｅｏｎカード消費者対カード販売者、ＮＦＣ又は他のカード消費者対カード販売者、資金振込み及び受領、アカウントポータル内で消費者アカウント対消費者アカウント、モバイル消費者対ピアツーピアモバイル消費者、モバイル消費者対ピアツーピアカード消費者、カード消費者対ピアツーピアモバイル消費者、カード消費者対ピアツーピアカード消費者、モバイル消費者対ピアツーピア非ユーザ、カード消費者対ピアツーピア非ユーザ、非ユーザ対ピアツーピア非ユーザ、非ユーザ対ピアツーピアモバイル消費者、及び非ユーザ対ピアツ

ーピアカード消費者、非ユーザは、送金の未受取人のように前に支払いサービスへ登録されていない人を示す。

【００７６】

・システムアーキテクチャー

内部的に、Ｔｅｒｅｏｎサーバは、２つのメインコンポーネントであるＴＲＥ（Ｔｅｒｅｏｎ Ｒｕｌｅｓ Ｅｎｇｉｎｅ）及びＳＤＡＳＦ（Ｓｍａｒｔ Ｄｅｖｉｃｅ Ａｐｐｌｉｃａｔｉｏｎ Ｓｅｒｖｉｃｅｓ Ｆｒａｍｅｗｏｒｋ）を含む。

【００７７】

ＳＤＡＳＦは、Ｔｅｒｅｏｎが様々な相異なるデバイス及びインタフェースを管理することができる。これは、Ｔｅｒｅｏｎが該当のデバイス及びインタフェースが作動し、Ｔｅｒｅｏｎに接続される方式を定義するために、一連の抽象化されたレイヤを使用及びリンク可能にすることによって実現される。

【００７８】

例えば、全ての銀行カードは、基本カード抽象化レイヤを使用する。磁気ストライプ抽象化レイヤは、磁気ストライプのあるカード、ＮＦＣチップのあるカードに対するＮＦＣレイヤ、及びチップコンタクトのあるカードに対するマイクロプロセッサーレイヤに適用されるのであろう。カードが３つの全てを使用する場合、Ｔｅｒｅｏｎは、メインカード抽象化レイヤ及び３つのインタフェースレイヤでそのカードを定義する。ＮＦＣレイヤのそれ自体がカードにのみ適用されるものではない。これは、モバイルを含むＮＦＣを支援できる任意のデバイスにも適用され得る。ＳＤＡＳＦは、デバイス又はインタフェースそれぞれに対するモジュールを生成するために、このような抽象化レイヤを使用する。

【００７９】

外部的には、デバイス又はネットワークに対する各接続及び各サービスはモジュールである。したがって、ピアツーピア支払いサービス、入金サービス、及びミニ明細書のようなサービスは全てモジュールである。カード製造社、銀行、サービス提供者、端末、ＡＴＭなどに対するインタフェースも同様である。Ｔｅｒｅｏｎのアーキテクチャーは、様々なモジュールを支援し得る。

【００８０】

・モジュラー観点（Ｍｏｄｕｌａｒ ｖｉｅｗ）

図１は、Ｔｅｒｅｏｎのモジュラー概念を示す。本質的に、Ｔｅｒｅｏｎは、モジュールの集まりであり、そのほとんどはモジュールを含んでいる。モジュールは、該当のモジュールが動作するコンテキスト及び機能ドメイン、及びそれが実行するために必要な機能を決定するビジネスロジックによって定義される。このような機能は、例えば、ＩｏＴデバイス間の動作及び通信を管理し、電子又はデジタル支払いの管理及び取引、識別又は要求による許可クリデンシャルを管理及び構成したり、任意の他の形式の電子トランザクション又はデバイスを管理及び運営するような任意のタイプの電子トランザクションであり得る。

【００８１】

・Ｔｅｒｅｏｎサーバ

図１に示すように、Ｔｅｒｅｏｎサーバ１０２を構成するモジュールは、ＳＤＡＳＦ１０４及び規則エンジン１０６といった２つのレベルで見ることができる。規則エンジン１０６そのものは、モジュール１０８（その中の一部は図１に示され、これはサービスを定義するモジュール、プロトコル（図示せず）、スマート装置、端末などを含む）それぞれの機能ドメイン及びコンテキストを定義し、次に、このようなモジュール１０８は、ＳＤＡＳＦ１０４の構造を定義する。その次に、ＳＤＡＳＦ１０４及びこれが支援している結果サービス及びインタフェースは、Ｔｅｒｅｏｎが利用できるシステムプロトコルを定義する。その次に、このようなプロトコルは、Ｔｅｒｅｏｎが支援できる規則及びサービス（（例えば、スマートデバイス、端末など）それ自体はＴｅｒｅｏｎが提供する機能ドメイン及びコンテキストを定義する）を定義する。この循環的又は繰り返し的なアプローチは、モジュールの定義及びそれが支援している機能又は要求事項が互いに一致するかを確

認するために用いられる。これにより、システムの動作を制限することなく、元の位置でモジュールがアップデートかつアップグレード、及び交換することができる。

【００８２】

ブロック及びモジュールは、抽象化されたＡＰＩｓ（（ａｐｐｌｉｃａｔｉｏｎ ｐｒｏｇｒａｍｍｉｎｇ ｉｎｔｅｒｆａｃｅｓ）それ自体は、Ｔｅｒｅｏｎが提供する機能ドメイン及びコンテキストを定義する）を用いてインタフェースする。可能であれば、これは共有メモリを使用できるオーダーメード型セマフォハンドオフモジュールを用いて通信し、その一例が図４ａに示されている。これについては後述する。このような方式により、ブロック及びモジュールの内部動作及び機能は、全体システムの動作を損なうことなく、アップデートされたり交換され得る。

【００８３】

・フレームワークインフラストラクチャーコンポーネント（Ｆｒａｍｅｗｏｒｋ ｉｎｆｒａｓｔｒｕｃｔｕｒｅ ｃｏｍｐｏｎｅｎｔｓ）

インフラストラクチャーコンポーネントもモジュラーである。ＳＤＡＳＦの場合、このコンポーネント自体がモジュールを含む。

【００８４】

・マルチインタフェース（Ｍｕｌｔｉｐｌｅ ｉｎｔｅｒｆａｃｅｓ）

各インタフェースは、コアサーバに接続される別途のモジュールとして構成される。したがって、Ｔｅｒｅｏｎのモジュラー構造は、バックオフィス（ｂａｃｋ ｏｆｆｉｃｅｓ）及びコアシステムを含むマルチインタフェース、カード、決済機関、販売者、モバイル電話機、サービス、サービス提供者、ストレージ、端末、ＳＭＳ（ｓｈｏｒｔ ｍｅｓｓａｇａ ｓｅｒｖｉｃｅ）ゲートウェイ、ＨＬＲ（ｈｏｍｅ ｌｏｃａｔｉｏｎ ｒｅｇｉｓｔｅｒ）ゲートウェイなどを支援し得る。

【００８５】

データベースインタフェースは、ＳＱＬ（ｓｔｒｕｃｔｕｒｅｄ ｑｕｅｒｙ ｌａｎｇｕａｇｅ）エントリ及び格納されたデータのグラフ分析の全てを支援する。また、インタフェースは、データベース内にフィールドを区分するためにアクセス制御を支援する。他のユーザ規則及び許可レベルは、定義されたデータセット及びフィールドをアクセスし得る。アクセスは、様々なセキュリティー手段によって制御される。アクセス、認証、及び許可は、ＡＣＬｓ（ａｃｃｅｓｓ ｃｏｎｔｒｏｌ ｌｉｓｔｓ）、ＬＤＡＰ（ｌｉｇｈｔｗｅｉｇｈｔ ｄｉｒｅｃｔｏｒｙ ａｃｃｅｓｓ ｐｒｏｔｏｃｏｌ）、セル及びローセキュリティー（ｃｅｌｌ ａｎｄ ｒｏｗｓｅｃｕｒｉｔｙ）のようなカスタムの役割ベースのアクセス（ｃｕｓｔｏｍ ｒｏｌｅ－ｂａｓｅｄ ａｃｃｅｓｓ）、及び個別の役割に制限されるアクセスインタフェースを含む産業標準の様々なアクセス方法により提供され得る。

【００８６】

・Ｅコマースポータル（Ｅ－ｃｏｍｍｅｒｃｅ ｐｏｒｔａｌｓ）

Ｔｅｒｅｏｎは、ポータルの運営者が該当のポータルに対するプラグインを生成できるように、ＡＰＩを介してＥコマースポータルを支援し得る。

【００８７】

・規則エンジン（Ｒｕｌｅｓ ｅｎｇｉｎｅ）

規則エンジン１０６は、新しいサービスがトランザクションに対して抽象化された様々なコンポーネントを共に編成して構築したり、新しいデバイスを支援する。規則は、配布されたサービスに対するビジネス論理を定義し、サービス提供者などはこのようなサービスを個別ユーザに合わせて調整できる。

【００８８】

規則は、ＵＭＬ（ｕｎｏｆｉｅｄ ｍｏｄｅｌｌｉｎｇ ｌａｎｇｕａｇｅ）又は一般英語（ｐｌａｉｎ ｅｎｇｌｉｓｈ）に類似のコードで定義される。エンジンは、規則を構文分析し、抽象化されたコンポーネントからサービスを生成し得る。

【００８９】

コンポーネントの抽象化された特性は、新しいサービス又はデバイスモジュールが迅速に生成できる。これにより、Ｔｅｒｅｏｎは、必要に応じて、新しいサービス又はデバイスを支援できる。

【００９０】

Ｔｅｒｅｏｎの内部インタフェースは、プロトコルに影響を受けないことから、外部プロトコルモジュールが機能に影響を及ぼすことなく交換することができる。例えば、銀行コアシステムにインタフェースを行うために、カスタムデータ交換プロトコルは、組織の一部とＩＳＯ２００２２プロトコルモジュールを他の部分と共に使用される。

【００９１】

ＳＤＡＳＦ１０４により、Ｔｅｒｅｏｎが多重スマートデバイス及びプロトコルを支援できる。ＳＤＡＳＦ１０４のアイディアは、エンティティをデバイスタイプ及びプロトコルに抽象化することにある。ＳＤＡＳＦ１０４は、各デバイスが特定のサービス又は機能に必要なプロトコルを呼び出すものと多重プロトコルを定義する。

【００９２】

ＳＤＡＳＦ１０４は、インストールの動作に影響を与えることなく、既存のインストールに新しいモジュールを追加して拡張される。どちらの方法を用いて全てのサービスをバックオフィスサーバに定義できる。販売者端末（ｍｅｒｃｈａｎｔ　ｔｅｒｍｉｎａｌｓ）にインストールされれば、Ｔｅｒｅｏｎ端末アプリケーションは、消費者にサービスを提供するためにＳＤＡＳＦと通信する。

【００９３】

図２は、Ｔｅｒｅｏｎシステムアーキテクチャー２００を示す。ここで、ダイヤグラム及び描写が特定のソリューションを介して特定のコンポーネントを示す場合、これが単に実施形態で選択されるコンポーネント又は言語であるためである。オーダーメード型（ｂｅｓｐｏｋｅ）システムは、このようなコンポーネントを置き換えたり、より効率的なものと立証され得る他の言語及びシステムを使用するために構築される。

【００９４】

・Ｔｅｒｅｏｎサーバ（Ｔｈｅ　Ｔｅｒｅｏｎ　ｓｅｒｖｅｒ）

Ｔｅｒｅｏｎサービス２０２は、モノリシックアーチファクトとして識別される論理的構造である。実際に、それは各機能及び範囲により異なる、隔離されたマイクロサービスのセットとして存在し得る。

【００９５】

・通信レイヤ（Ｔｈｅ　ｃｏｍｍｕｎｉｃａｔｉｏｎｓ　ｌａｙｅｒ）

通信レイヤ２０４は、仲介者プロキシを経てＴＬＳ（ｔｒａｎｓｐｏｒｔ　ｌａｙｅｒ　ｓｅｃｕｒｉｔｙ）接続を介して開始される。これについても図３に示される。ＴＬＳは、コンピュータネットワーク、一般的にＴＣＰ／ＩＰ（ｔｒａｎｓｍｉｓｓｉｏｎ　ｃｏｎｔｒｏｌ　ｐｒｏｔｏｃｏｌ／ｉｎｔｅｒｎｅｔ　ｐｒｏｔｏｃｏｌ）ネットワークを介して通信セキュリティーを提供する暗号化プロトコルである。各コンポーネントは、システム、オブジェクト又はサービスに接続されたり、アクセスできるユーザ又はシステムプロセスを明示するＡＣＬ（ａｃｃｅｓｓ　ｃｏｎｔｒｏｌ　ｌｉｓｔ）がある。これにより、仲介者のみがくるオリジナル接続（ｉｎｃｏｍｉｎｇ、ｏｒｉｇｉｎａｌ　ｃｏｎｎｅｃｔｉｏｎ）を設定し、本質的なセキュリティーが強化され、危険プロファイルは減少される。この例では、プロキシは、専門化されたＴｅｒｅｏｎカスタマイズを使用した、従来技術で知られているＨＴＴＰゲートウェイプラットフォームを使用している。

【００９６】

・個人ＤＮＳネットワーク（Ｐｒｉｖａｔｅ　ＤＮＳ　ｎｅｔｗｏｒｋ）

ＤＮＳ２０６は、ディレクトリサービス２１６の基礎として用いられる。ディレクトリサービス２１６は極めて重複し、地理的な位置にわたって複製される。しかし、その構造及び機能は、下記に説明されるように既存のＤＮＳサービスが提供できるものを遥かに超えるものである。

【００９７】

・抽象化（Ａｂｓｔｒａｃｔｉｏｎｓ）
図２ａは、Ｔｅｒｅｏｎがそのサービス及びデバイスを、消費者又は消費者の処理及び規則、販売者の処理及び規則、銀行の処理及び規則、振込みの処理及び規則、デバイス機能及び規則などのような機能ドメイン及びコンテキストに抽象化する方法を示す。図１は、コンポーネント及びシステムのサービスを機能ブロック又はモジュールに抽象化することにより、Ｔｅｒｅｏｎがこのような抽象化にどのように映像を及ぼすかを図示する。
【００９８】
Ｔｅｒｅｏｎモジュールは、このような抽象化から構成される。各デバイス、各インタフェース、及び各トランザクションタイプは、そのドメイン及びコンテキストに抽象化される。このような抽象化は再利用でき、意味がある場合や許可される場合に、他のものにインタフェースし得る。例えば、チャージカード、クレジットカード、デビットカード、及びロイヤルティカードの各モジュールは、一般的な基本抽象概念を使用する。支払い及び資金の振込みモジュールも同様である。
【００９９】
・プロトコル（Ｐｒｏｔｏｃｏｌｓ）
Ｔｅｒｅｏｎが支援するプロトコル２０４及び２１２のそれぞれは、それ自体がモジュールとして実現される。Ｔｅｒｅｏｎは、このようなモジュールを必要とするサービス又はコンポーネントがこのモジュールを利用できるようにする。
【０１００】
レガシーシステムでは、ハードウェアを追加する前に、１００ｓ又は１０００ｓに同時に生じるトランザクションを処理するのに苦労している。システムをアップデートする代わりに、銀行は、調整アカウント及び支払いポイントまで信用をカバーするための高いコストが要求される定期的な支払いシステムに依存してきた。Ｔｅｒｅｏｎは、信用露出及びこのようなアカウントに対する必要性から離隔されている。これは秒当たり１００、０００件のトランザクションを処理するように求められる極めて安価なシステムを提供する。Ｔｅｒｅｏｎは弾力性が構築され、サーバ当たり秒当たり１、０００、０００件のトランザクションを支援し、高価なハードウェアに依存することなく、ハイエンドの商品ハードウェアで動作するように設計されている。また、Ｔｅｒｅｏｎは、ＡＣＩＤ保証やそのリアルタイム性能を損傷することなく、ほぼ水平方式（ｎｅａｒ－ｌｉｎｅａｒ　ｆａｓｈｉｏｎ）で水平及び垂直スケーリングを支援する。
【０１０１】
・ライセンスサブシステム（Ｔｈｅ　ｌｉｃｅｎｓｉｎｇ　ｓｕｂｓｙｓｔｅｍ）
Ｔｅｒｅｏｎライセンスサーバ２１０は、システムのコンポーネントが単一に配布されたインスタンス（単一インスタンスのマイクロサービスはマシーンが、例えば、物理的機械、論理的機械、仮想機械、コンテナや実行可能なコードを含むための一般的に用いられるメカニズム、及び任意の数又は機械のタイプに関わらず単一マシーン上のプロセス間通信に結合される）内で配布インスタンスの全体（例えば、相互通信する個々の消費者プラットフォーム）内で、合法的で、認証されて認可されたピアシステム（ｐｅｅｒ　ｓｙｓｔｅｍｓ）と通信することを保障する。ライセンスプラットフォームは、当技術分野で知られた認証機関の構造を介して実現される。
【０１０２】
コンポーネントがシステムにインストールされるとき、それらは規定されて設定可能な間隔で、安全で認証された接続を介して、ライセンスサーバに認証書署名要求と共にそれらのインストール詳細（組織、コンポーネントタイプ及び詳細、ライセンスキーなど）を伝達する。
【０１０３】
認証書サーバは、その詳細を許可されたコンポーネントディレクトリと比較し、一致すると、インストールの要求を開始するデバイスへ内部の認証機関ハイアラーキーで隔離されたセキュリティー署名キー（一般的に、ハードウェアセキュリティーモジュールを介して）で署名され、一定期間（例えば、１ヶ月）の間に使用可能な新しい認証書を承認する

。接続システムの全てのクロックは同期化される。

【０１０４】

その次に、呼出者（ｃａｌｌｅｒ）は、他のモジュールとの通信を開始するときクライアント認証書として、また、接続の受信者として役割を果たすとき、サーバ認証書として認証書を使用する。個人キーを受信していないライセンスサーバは、たとえ損傷されても、他の第３者がこの認証書を偽装することを可能にするような詳細を有しない。必要に応じて、呼出者は、クライアント認証書及びサーバ認証書といった２つの認証書をライセンスサーバから要求できる。

【０１０５】

各コンポーネントは、サーバ及びクライアント認証書が信頼できる認証された認証機関のエージェントによって署名されていることを検証し、それが中間者攻撃又は監視の対象ではなく、相手側（ｃｏｕｎｔｅｒ－ｐａｒｔｙ）がそれが言う人であるという相当な確信をもって通信できる。各認証書は、各モジュールそのもの（例えば、特定の組織に対するルックアップサーバとして）を表示できる方法を制限する使用コードメタデータによって承認される。組織は、全ての関係者がライセンスを取得した合法的で有効なインスタンスを運営することを保証する。

【０１０６】

多くの認証書は、期限が満了して固定期間の間に更新されることなく、また承認されない。しかし、認証書が損傷されたり、ライセンスが終了又は一時停止される場合に取り消しリストが使用され、必要に応じて、プロキシサービスに非同期的に分配される。アクティブ認証書ディレクトリは常に保持され、定期的な監査に使用できる。

【０１０７】

双方向の有効検査の利点の他に（クライアントは、自分が話者であり、各接続のサーバは報告する者である）、この実現によって、コンポーネントが遠隔ライセンスサーバとの通信に必要な各接続の構築なしに安全に相互通信を可能にし、プラットフォームの全般的な信頼性を低下させることなく安全に通信可能にする。

【０１０８】

・サイト間の通信（Ｓｉｔｅ　ｔｏ　ｓｉｔｅ　ｃｏｍｍｕｎｉｃａｔｉｏｎｓ）

サイト間の通信は、識別されて公開されているＨＴＴＰゲートウェイインスタンス（ＨＴＴＰ　ｇａｔｅｗａｙ　ｉｎｓｔａｎｃｅ）２１２を介して容易になり、カスタムゼロ－コピー及び選択的なユーザモード機能を実行する。これは、モバイルデバイス、端末、及びその他の外部関係者がサイト間の接続は勿論、インスタンスと通信するために用いられるプラットフォームである。これは、産業標準の侵入検知、レート制限、及びＤＤＯＳ（ｄｉｓｔｒｉｂｕｔｅｄ　ｄｅｎｉａｌ－ｏｆ－ｓｅｒｖｉｃｅ）攻撃保護、ハードウェア暗号化オフロードなどに対応する。これは機能的に、論理的インスタンスプロキシメカニズムであり、同じ機能（クライアント／サーバ認証書及び有効性検査を含む）を全て支援すると共に、外部で認められている認証機関を外部の当事者に使用する。

【０１０９】

・Ｔｅｒｅｏｎデータサービス（Ｔｈｅ　Ｔｅｒｅｏｎ　ｄａｔａ　ｓｅｒｖｉｃｅ）

Ｔｅｒｅｏｎシステムの重要な特徴の１つとして、以前のシステムよりも遥かに多いトランザクション（処理量の観点で）を処理できることである。これは、データ及びトランザクションを処理できる高度な同時性、迅速性、及び拡張性に優れる処理ネットワーク、極めて効率的なデータサービスレイヤのみならず、処理オーバーヘッドを最小化するアルゴリズム及びオーダーメード型モジュールを実現する独自の設計によるものである。

【０１１０】

説明された性能特性は、コンピューティングハードウェアの特定の部分で多く実行される規模拡張に主にターゲットされることで、運営コスト及び消費電力が大幅に減少される。ただし、設計は単一システムに限定されず、Ｔｅｒｅｏｎシステムは、複数のデバイス上に同時に実行できる各サービスを用いて垂直及び水平的に膨大な規模でスケールアウト（ｓｃａｌｉｎｇ　ｏｕｔ）し得る。

【０１１１】
単一のシステム又はサーバ上で高いレベルの性能を取得するために、システムは、不要な直列化を避け、不要なストリーム処理を避け、不要なメモリコピーを避け、ユーザからカーネルモードへの不要な切り替えを避け、プロセス間のコンテキストスイッチを避け、ランダム又は不要なＩ／Ｏを避けることで、処理オーバーヘッドを可能であれば最小化する。システムが正常に作動するとき、これは該当システム上に極めて高いレベルのトランザクションパフォーマンス達成を図ることができる。
【０１１２】
既存のモデルでは、サーバＡが要求を受信する。その後、サーバＢへのクエリを構築して直列化し、すぐにサーバＢに該当クエリを送信する。その後、サーバＢは（必要に応じて）該当クエリを解読し、逆直列化して解釈する。その後、応答を生成して直列化し、必要に応じて該当び応答を暗号化してから応答をサーバＡ又は他のサーバに再度送信する。カーネル及びプロセスコンテキストの切り替えは、メッセージごとに数十回行われ、単一のメッセージは様々な形態に何度もキャストされ、メモリは多くの作業バッファ間でコピーされる。このようなカーネル及びプロセスコンテキストの切り替えは、処理されるメッセージごとに大規模の処理オーバーヘッドを課する。
【０１１３】
・通信アーキテクチャー（Ｃｏｍｍｕｎｉｃａｔｉｏｎｓ　ａｒｃｈｉｔｅｃｔｕｒｅ）
Ｔｅｒｅｏｎは、システムによって処理される従来方式のデータ及び通信を再構成し、その処理量を達成する。可能であれば、Ｔｅｒｅｏｎは、カーネルによって課される処理オーバーヘッドを避け、標準データ管理モデルで頻繁に発生するセキュリティー問題を回避するために、運営システムカーネルをバイパスする。
【０１１４】
システムで各データ処理は、データサービスインスタンス２１４を介して実行される。これは、直接データプラットフォームアクセスを有するシステムの唯一のコンポーネントである、規模の縮小されたサービス（指向データサービスレイヤ）である。したがって、システム上の全てのデータ処理は、必ずこれを通過しなければならない。
【０１１５】
データサービスレイヤ２１４は、別途の専用の読み出し及びレコードアクセスチャネル２２６を介してデータストアレイヤ２２０と通信する。データストアレイヤ２２０は、それ自体が少なくとも１つの分散データベースを含むコアデータベースストア２２４を介して実現される。このようなデータベースは、ＡＣＩＤ保証を提供する必要がない。これはデータストアレイヤによって管理される。
【０１１６】
データストアレイヤ２２０に対する全てのレコードは、全てのデータ変更が因果関係を保持するためにシリアルの高速シーケンスで進行されながら、単一共有トランザクタによって管理され、これを通じて全てのデータ変更が因果関係を保持するためにシリアルの高速シーケンスで進行される。トランザクタの設計は、データトランザクタクラスタ２２２として自身を提示するホットバックアップリダンダンシーモデル（ｈｏｔ　ｂａｃｋｕｐ　ｒｅｄｕｎｄａｎｃｙ　ｍｏｄｅｌ）を使用する。１つのトランザクタがいずれかの理由によって失敗したり停止する場合、他のランザクションのいずれか１つはすぐに引き継ぐのであろう。
【０１１７】
データプラットフォームは、全てのデータドメインに対する分割を支援するが、その支援は図示されていない。いずれのケースで単一データストアレイヤ（無制限データノードによってバックアップされる）が禁止されたり、又は規制上の理由がある場合、データは互いに異なるトランザクションを用いて互いに異なるデータクラスタに格納するために、命令的又は宣言的な方法によって分割される。例えば、１つのサイトは４つのデータプラットフォームがあり、プラットフォームは、地理的又は司法的な基準により顧客を分割し

たり、１～５で始まるアカウト、又は６～０で始まる他のアカウントで顧客を分割する。これに対する処理上の問題があるが、これはプラットフォームによって支援される。

【０１１８】

図３は、データサービスレイヤ２１４及びそれから通信をルートする通信レイヤ２０４を通した通信を示す。モジュール３５０が他のモジュール３６０と通信する必要がある場合、まずプロキシ３７０と接続を開始し、ステップ３０２でクライアント認証書を認証し、ステップ３０４でプロキシ認証書が構築するとき有効に信頼されるかをチェックする。モジュール３５０は、ステップ３０６でメッセージをプロキシ３７０に伝達する。プロキシ３７０は、ステップ３０８でターゲットモジュール３６０と関係接続（ｃｏｒｒｅｌａｔｉｎｇ　ｃｏｎｎｅｃｔｉｏｎ）を設定する。まず、ステップ３０８で自身を認証し、ステップ３１０でモジュールの認証書が有効に信頼されるかを検証する。プロキシ３７０は、ステップ３１４でモジュールの応答を受信する前に、ステップ３１２でイニシエーター（モジュール３５０）の確認された詳細を伝達する。プロキシ３７０は、ステップ３１６でターゲット（モジュール３６０）の詳細及びその応答を返す。これにより、プロキシ３７０を介してモジュール３５０及びモジュール３６０間の通信チャネルを確立し、２つのモジュール全てが高い信頼度で互いに認証されて識別され、必要に応じて、全ての通信及びデータが暗号化される。プロキシ３７０は、ステップ３１８でモジュール３５０からのメッセージをステップ３２０においてターゲットモジュール３６０に中継し、ステップ３２２において、ターゲットモジュールの応答をステップ３２４でモジュール３５０に中継する。

【０１１９】

このような接続は、発信者及び受信者の認証書の詳細に基づいてセッション共有及び接続保持（ｋｅｅｐ－ａｌｉｖｅ）を使用する（例えば、モジュール３５０は、プロキシ３７０を介してターゲットモジュール３６０に対する接続を「閉じ（ｃｌｏｓｅ）」、実際に新しいエンドツーエンド接続（ｅｎｄ－ｔｏ－ｅｎｄ　ｃｏｎｎｅｃｔｉｏｎ）を構築することなくリオープン（ｒｅｏｐｅｎ）し、接続は任意の他の回路で絶対共有されない）。通信プロキシ３７０は、ＨＴＴＰゲートウェイ又は他の適切なモジュールもしくはコンポーネントであってもよい。

【０１２０】

このようなアーキテクチャーは、主に大量のメモリの使用によって相当な性能上のコストが発生する。モジュール３５０がターゲットモジュール３６０と通信するために、伝統的に、ターゲットモジュール３６０へ伝達する前にペイロードを直列化し、ペイロードを暗号化し、これをプロキシ３７０にストリームし（ここで、プロキシ３７０はペイロードを解読する）、コンテンツを逆直列化及び解釈し、ペイロードを再直列化し、ターゲットモジュール３６０に対してこれを暗号化する必要がある。ターゲットモジュール３６０は、コンテンツを解読し、逆直列化して解釈し得る。

【０１２１】

Ｔｅｒｅｏｎは、平均及び最大の待ち時間（ｌａｔｅｎｃｙ）を減らし、メモリの負荷を減らし、常用のハードウェア上の単一プラットフォームの性能を向上させるために様々な技術を使用している。これはマイクロサービスの配布利点（ｄｅｐｌｏｙｍｅｎｔ　ｂｅｎｅｆｉｔｓ）、メンテナンス、及びセキュリティーの全てを保持しながら、モノリシックインプロセス性能を達成できる。このようなシステムが提供しなければならない高いレベルのセキュリティー及び制御を損なうことなく実行される。

【０１２２】

Ｔｅｒｅｏｎは、図３に示すように、通信レイヤを介してバッチされたメッセージングモデルを用いることができる。ステップ３０６において、モジュール３５０からプロキシ３７０に伝えられたメッセージのような伝達された各メッセージは、メッセージのバッチであり得る。しかし、Ｔｅｒｅｏｎは、これ以上のことを実行することができる。

【０１２３】

バッチされたメッセージングに加え、図４は、２つのモジュールのサーバが互いに共有

メモリチャネルを交渉するためにプロキシモジュール（オーダーメード型ハンドオーバーモジュール）を介して通信することを示す。ステップ４０２ないし４１２は、図３でステップ３０２ないし３１２に類似し、必要に応じて、サービスの属性をチェックし、それがクライアント要求とマッチするかを確認する。これはステップ３０２ないし３１２で発生し得る。

【０１２４】

モジュール４５０ないしモジュール４６０のインスタンスは、ＴＬＳ、又は伝統的なＴＬＳ　ＨＴＴＰＳだけでなく、呼出者トランザクションに対するＨＴＴＰゲートウェイのユーザモード及びゼロコピーをともに最適した形で使用できる。

【０１２５】

ソースモジュール４５０及び配信先モジュール４６０がローカルである場合、ステップ４０２ないし４１２からのプロキシ４７０を介して接続を設定した後、発信者及び受信者は共有メモリを介して直接接続を選択的に要求し、この選択的な要求により、この方法は図３に設定された方法と異なる。発信者及び受信者が互いに直接接続を要求する場合、ネゴシエーション後で、共有チャネルはステップ４１４でモジュール４６０からプロキシ４７０に、ステップ４１６でプロキシからモジュール４５０に伝えられ、この点から２つのモジュールは、セマフォ及び共有メモリを再び使用する直接処理メカニズムを使用する。これは、ステップ４１８、４２０、４２２などにおけるモジュール４５０及びモジュール４６０間のメッセージによって示される。

【０１２６】

Ｔｅｒｅｏｎモデルでは、サーバ４５０は、タスクに最適した形で基本メモリバッファ（ｎａｔｉｖｅ　ｍｅｍｏｒｙ　ｂｕｆｆｅｒｓ）内の複数の要求をバッチ処理し、サーバ４６０にメッセージをキューイングし、セマフォをトリップする。サーバ４６０は、フラグをチェックし、直接的に共有されたメモリを処理し、共有メモリに応答する。接続には、発信者及び受信者の認証書の詳細と通信のためのセマフォ及び共有メモリに基づいて接続保持及び共有されたメモリを使用する。

【０１２７】

上述した方法を用いて、通信は、単一発信者の配信先、ＡＣＬ－制御、セキュリティーに対する直列化及びストリーミングのオーバーヘッド（機械内に含まれる場合）を回避する。暗号化は不要である。接続は、有効性が検証かつ認証され、セットアップ時に許可され、無効化されず、適切であれば、プロセスは大規模な独自のメモリ構造を共有できる。

【０１２８】

プロキシ４７０及びＴｅｒｅｏｎコードモジュール４５０及び４６０の全ては可能であれば、ゼロコピーネットワーキング及びユーザモードネットワーキングを支援する（必須のＲＣＰ／ＩＰライブラリーでコンパイルされる場合、ＨＴＴＰプロキシはネットワークパケットに対するカーネルコンテキストスイッチの相当なコストを削減できるソリューションを提供する）。これは、プロキシ４７０及びＴｅｒｅｏｎコードモジュールが使用できるネットワークドライバ特定のコードを介して容易になる。これにより、小さなパケットの要求及び応答に対するメモリ使用を最小化できる。これは膨大なＴｅｒｅｏｎ動作（ここで多くの動作は単一ＴＣＰパケットに適する）を構成する。

【０１２９】

図４ａは、ＴｅｒｅｏｎシステムがＴｅｒｅｏｎシステムの任意の２つのコンポーネント（例えば、Ｔｅｒｅｏｎ内の機能を提供するＨＴＴＰゲートウェイ４０６ａ及びマイクロサービス４１０ａ）の間でデータを効率よく交換するために用いられる共有メモリを使用できるオーダーメード型セマフォハンドオフモジュール４０８ａのセットを実現する方法を示す。図４ａにおいて、データサービスレイヤ２１４は、マイクロサービス４１０ａによって実現される。しかし、マイクロサービスは、全ての種類のサービスモジュールを示すことができる。

【０１３０】

ネットワークスタック４０４ａ（ループバック仮想デバイスを含む）は、接続サーバ４

０２ａから要求を受信及びアセンブルし、これをユーザモードターゲットメモリにコピーする代わり、単にメモリ承認の所有権を受信者（この場合にはＨＴＴＰゲートウェイ４０６ａ）に渡す。これはメモリ帯域幅の飽和が発生し始める極めて大きい負荷（例えば、秒当たり数百万の要求）で主に有用である。

【０１３１】

カスタムＴｅｒｅｏｎアップストリームＨＴＴＰゲートウェイモジュール（ｃｕｓｔｏｍ Ｔｅｒｅｏｎ ｕｐｓｔｒｅａｍ ＨＴＴＰ ｇａｔｅｗａｙ ｍｏｄｕｌｅ）４０６ａは、ローカルインスタンス（一般的に各コンテナ又はそれぞれ物理的、論理的、又は仮想機械上にＨＴＴＰゲートウェイインスタンスがある場合にはＨＴＴＰゲートウェイインスタンスに関し）がゲートウェイからモジュールにプロキシメモリに対するメモリ伝達及び共有メモリを使用するためのオプションを許容し、その反対の場合も、該当のアップストリーム接続を許容する。ＨＴＴＰゲートウェイ４０６ａが既存のメカニズムを介して要求を直列化してこれを伝達する代わりに、共有メモリアップストリーム提供者のために構成されるとき、ＨＴＴＰゲートウェイ４０６ａは受信者に伝達する共有メモリを使用する。

【０１３２】

この場合に、共有メモリは、他のＨＴＴＰゲートウェイ、ＨＴＴＰゲートウェイインスタンス、又は他の要素をプロキシに使用して設定される。ＨＴＴＰゲートウェイを使用することは特に効率的である。

【０１３３】

運営システムカーネルによって提供される通信フックを使用する代わりに、各データ交換モジュールは、カーネルをバイパスする。これにより、カーネルオーバーヘッドを回避してシステムの処理量を増加させ、データがカーネルによって提供されるサービスに／から伝達されるとき発生しうる不安定な領域を扱う。例えば、Ｔｅｒｅｏｎ内のモジュールは、システムコンポーネントから直接データサービスレイヤ２１４に、及びデータサービスレイヤ２１４からシステムコンポーネントにデータを効率よく交換するために用いられる。

【０１３４】

このアーキテクチャーがもたらす利点の他の例は、ＨＴＴＰゲートウェイ４０６ａの向上した効率（ＨＴＴＰゲートウェイ４０６ａがデータサービスレイヤ２１４又は他のコンポーネントのようなマイクロサービス４１０ａに対する全ての入力データ、及びマイクロサービス４１０ａ又はデータサービスレイヤ２１４からの全ての出力データをＨＴＴＰゲートウェイ４０６ａにハンドオーバーにするハンドオフモジュール４０８ａを用いて達成される）である。基本ＨＴＴＰゲートウェイのデータ及びメッセージングハンドオフを使用する代わりに、それ自体が効率的で、共有メモリを使用できるセマフォハンドオフモジュールは、データがカーネルを迂回し、データレーサー２１４からＨＴＴＰゲートウェイ４０６ａに、及びデータレイヤ２１４から直接伝えられるようにする。これは、システムの処理量を増加させるのみならず、ＨＴＴＰゲートウェイを使用するシステムにおいて共通の弱点領域のうち１つを保護するという点で追加的な利点がある。

【０１３５】

共有メモリチャネルを提供するモジュール又は共有メモリチャネルと通信するモジュールは、要求をバッチ及び直列化したり逆直列化して分離する。該当のタスクを行うモジュールは、該当モジュールの機能及びモジュールが正常に作動するとき発生する処理オーバーヘッドに達することになる。例えば、ある場合には、それ自体が複数のメッセージ（要求であってもなくてもよい）を受信するモジュールは、受信者モジュールに対してそのメッセージをバッチ及び直列化する共有メモリモジュールにメッセージを伝達するが、バッチ及び直列化のオーバーヘッドが効率的かつロード時にメッセージを処理する他の方法から該当モジュールを妨げるためである。他の場合に、モジュールは、共有メモリチャネルを介して該当受信者にバッチを伝達する前に、そのメッセージを特定の受信者にバッチ及び直列化し得る。

【０１３６】
更なる場合では、メッセージを受信者モジュールに伝達するモジュールは、メッセージをバッチ及び直列化するために共有メモリチャネルを提供するモジュールに依存し得るが、バッチメッセージを受信するモジュールは、それ自体がメッセージを逆直列化及び分離し得る。どのモジュールがバッチ及び直列化、又は、逆直列化及び分離のタスクを行うかに対する問題は、いずれかの選択によりモジュールが行う機能のための最適性能レベルを提供するかにかかっている。バッチ及び直列化の順は、メッセージタイプ及び通信モジュールによって提供される機能に応じて異なる。
【０１３７】
Ｔｅｒｅｏｎは、ウェブサービスになるためＨＴＴＰゲートウェイ４０６ａを使用し、ネットワーク運営者が非標準サービスを遮断する潜在的な問題を回避する。Ｔｅｒｅｏｎは、勿論、必要に応じて、任意の他のサービスのふりをすることが可能であるため、周知のネットワークセキュリティー構成を容易に作業でできる。
【０１３８】
このような設計により、システム（このシステムは、利用可能な資源を使用するように設計されたモジュールを使用）は、全体のアーキテクチャーに通じてこのモジュラーアクセス方式を採用し、可能な場合、オーバーヘッドを回避する。追加的な例として、ネットワークスタック４０４ａでゼロコピーネットワーキング又はユーザモードネットワーキングを支援するモジュールのネットワーキングシステム（可能な場合にＴｅｒｅｏｎを使用）である。これにより、ネットワーキングに対するカーネルを使用する多くのオーバーヘッドを避けることができる。また、モジュラー設計は、Ｔｅｒｅｏｎがシステムの多重タイプ（類似オーダーメード型モジュールが類似機能を提供し、各運営システム又はハードウェア構成に適するようにカスタマイズされる）で動作可能にする。
【０１３９】
図３及び図４に示された方式により、仲介者を使用することは内部機械又は外部機械の全ての通信に対する中央集中制御地点を許容する。これは、速度及びセキュリティー制御、モニタリング及び監査、及び特殊な規則又は再指示に対する単一制御地点である。これにより、ダウンタイムや重大なリスクを招くことなく、システムの運営中にもシステムを柔軟に展開できる。また、クライアントの認識又は複雑性なしに、ロード分散（ｌｏａｄ－ｂａｌａｎｃｉｎｇ）及びリダンダンシー（ｒｅｄｕｎｄａｎｃｉｅｓ）を容易にする。
【０１４０】
図３に示すモジュール３５０がターゲットモジュール３６０に通信することを所望する場合、仲介者の使用は、ターゲットモジュール３６０が「ｎ」機械にわたってロード分散し、仲介者を単に再設定する代わりに、全ての潜在的クライアントを再設定することなく、任意の数又は機械のタイプ間で移動可能にする。
【０１４１】
システムは、２つの通信当事者が互いのキー交換を認証する機能を提供するために生成されたＰＡＫＥ（ｐａｓｓｗｏｒｄ ａｕｔｈｅｎｔｉｃａｔｅｄ ｋｅｙ ｅｘｃｈａｎｇｅ）プロトコルを使用する。これは、異なる周知の共用キー交換プロトコル（例えば、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎキー交換プロトコルのような、中間者攻撃にプロトコルを脆弱させる）に対しては不可能である。正しく使用される場合に、ＰＡＫＥプロトコルは中間者攻撃の影響を受けない。
【０１４２】
Ｔｅｒｅｏｎが外部デバイス又はサーバのような外部システムと通信する場合、通信システムに追加レイヤが追加される。多くのキー交換プロトコルは、中間者攻撃に理論的に脆弱であり、通信が２つの知られたエンティティ間にあることを確認するために、認証書及び署名されたメッセージを用いて接続が設定されれば、システムは、第２セキュリティーセッションキーを設定するためにＰＡＫＥプロトコルを使用することから、通信は中間者攻撃に影響を受けない。したがって、通信は、ＴＬＳセッションキーを用いて、次の全

ての通信を暗号化するためにＰＡＫＥプロトコルのセッションキーを使用する。
【０１４３】
不可避の識別ストリングを有するデバイスと通信する場合、必要に応じて、ＴＬＳは省略され、ＰＡＫＥプロトコルがメインセッションキープロトコルとして用いられる。例えば、デバイスがモノのインターネットのコンポーネントのセットを形成する小型のハードウェアセンサである場合に発生し得る。
【０１４４】
・通信方法（Ｃｏｍｍｕｎｉｃａｔｉｏｎ　ｍｅｔｈｏｄｓ）
Ｔｅｒｅｏｎデータサービス２１４は、調整トランザクタ（１つ以上のトランザクションの全て又は一部を実行、管理又は制御するデバイス又はモジュール）を介して完全なＡＣＩＤ保証を提供し、ｎ＋１又はそれより大きいリダンダンシー及び選択的多重サイト複製を提供するグラフ機能を有するキーバリューストアーに基づく。データサービス２１４は、共有メモリ機能の他に、ゼロコピー機能及び無制限の読み出しスケーリング、インメモリキャッシュ、及び非常に高いレベルのレコード性能を提供するデータ）ドメインサービスにカプセル化される。これは、大量のメモリキャッシュを従う、可変サイズデータクラスタで保持される。非常に独特の状況では、データサービスは、キーバリューストアーを直接使用するために回避され得る。
【０１４５】
データサービス２１４は、高い性能の基本ＳＱＬスタイル機能と共に、貨幣の流れ分析のような機能を支援するグラフ処理機能を提供する。極めて高性能なモジュール通信アーキテクチャー（プラットフォームの効率性及び性能を提供する）と結合されたデータサービス２１４は、汎用サーバハードウェア（結合された１０Ｇｂｐｓネットワーキングを有する）のテストにおいて、秒当たり２８０万トランザクションを超える極めて効率的な設計を提供する。
【０１４６】
以下のアーキテクチャー上の優先順位を実現することで、システムはシステム内及びシステム間で送信されるメッセージを処理するために必要なカーネル及び処理コンテキストスイッチ数を大幅減らすことができる。
【０１４７】
ａ）ゼロコピーネットワーキングは、ネットワークエッジからサービスまでの送信コストを最小化できる。
ｂ）ユーザモードネットワーキングは、ネットワークエッジからサービスまでの送信コストを最小化できる。
【０１４８】
ｃ）直列化が必要な場合（主に、機械又はサーバの境界を越えるとき）、高効率の直列化は、ＳＯＡＰ（Ｓｉｍｐｌｅ　Ｏｂｊｅｃｔ　Ａｃｃｅｓｓ　Ｐｒｏｔｏｃｏｌ）のような高いオーバーヘッドの直列化とは対照的に、プロトコルバッファ又はＡｖｒｏで用いられる。これは、各サーバのエッジで抽象化されているため、性能及び効率性レベルが低くても、特定のサーバがインターネットを介して他の大陸（ｃｏｎｔｉｎｅｎｔ）のピアサーバで容易に通信できる。
【０１４９】
ｄ）サーバは、処理コンテキストスイッチを最小化し、特定のサーバに対するキャッシュ一貫性を最大化にするための要求を、バッチ処理することを試みるバッファリング閾値を設定できる。例えば、サーバＡに１００００個のリクエストが２０ｍｓ期間内に到達し、プラットフォームが２０ｍｓバッファウィンドウを目標とし、該当１００００個のリクエストのためにサーバＢの支援が必要な場合に、１００００個のリクエストを単一のリクエストとしてまとめた後セマフォをフラグ（ｆｌａｇｇｉｎｇ）し、サーバＢに対する非同期メッセージを待機させる。その後、サーバＢは、１００００個のリクエストを迅速に処理し、サーバＡに単一の応答を提供できる。これは、効率と最大応答時間の最適化に基づいて構成できる。

【０１５０】
実際に、カーネル及びプロセスコンテキストスイッチの数を減らすことで、プラットフォームの性能レベルが大幅改善された。メッセージごとに多くのカーネル及びプロセスコンテキストスイッチが発生するのではなく、Ｔｅｒｅｏｎモデルは、通信されるメッセージのバッチによってメッセージのブロックごとに多くのカーネル及びプロセスコンテキストスイッチを発生させる。テストによると、このモデルを用いて既存のモデル及びＴｅｒｅｏｎモデル間の性能差は大きく作業負荷に対して１：１０００以上になる。
【０１５１】
しかし、モジュール及びその利点は単一システムに制限されない。例えば、サーバＡ及びサーバＢが個々の機械にある場合でも、Ｔｅｒｅｏｎシステムは効率的な直列化及びバッチ処理を使用できるのであろう。これは選択的なゼロコピー又はユーザモードネットワーキングと組み合わされるかどうかに関わらず、Ｔｅｒｅｏｎモデルは、ネットワーク及び処理性能を大きく向上させる。
【０１５２】
テストにより、このような設計要素は、超高速ネットワークワイヤー（例えば、１０Ｇｐｓボンディング）を介して毎秒数千万のメッセージ要求及び応答のアラウンドトリップ（バッチ、共有メモリモード）でローカルサーバ間のサーバ運営を実証したことを示した。
【０１５３】
このようなトランザクションは、全てリアルタイムで処理され、すぐに調整されるために、特に銀行、ＩｏＴ、医療、ＩＤ管理、輸送、及び正確なデータ処理が必要な環境では多くの長所がある。特に、このようなシステムは、現在のリアルタイムでトランザクションを調整しない。その代わりに、トランザクションは、一定期間の後に、時にはバッチで調整される。例えば、金融トランザクションは通常、数時間後に実行される別途の照会プロセスを用いてバッチ処理される。Ｔｅｒｅｏｎシステムを使用することで、銀行は今まで不可能であった方式により、全ての金融トランザクションをリアルタイムに調整できるようになる。その結果、銀行は、全てのトランザクションがそれを処理されるとき調整されるため、正確に調整されないか、又はまだ調整されていない金融トランザクションをカバーするために調整アカウントを保有する必要がなくなる。
【０１５４】
トランザクション及びデータ分割
Ｔｅｒｅｏｎシステムの全ての原子処理（ａｔｏｍｉｃ　ａｃｔｉｖｉｔｉｅｓ）はトランザクションである。トランザクションに対するＡＣＩＤ保証を裏付けるシステムの基本的が要求事項として、それらは全体的に失敗したり、全体的に成功する。このセクションは、これがどのように実行されるかに簡単に説明し、Ｔｅｒｅｏｎがトランザクションに対するＡＣＩＤ保証を達成する上での分割の影響を緩和するために、トランザクション及びデータ分割対して行ったアクセス方式の詳細を説明する。
【０１５５】
上述したように、Ｔｅｒｅｏｎプラットフォーム内の各データ処理は、Ｔｅｒｅｏｎデータサービスインスタンス２１４（それ自体がマイクロサービス４１０ａの組みとして動作できる）を介して実行される。これは直接データプラットフォームアクセスのある唯一システムのコンポーネントである拡張されたサービス（指向システム）であるため、全てのデータ処理はこれを通過しなければならない。このようなデータサービスは、インスタンスキャッシュデータＭＶＣＣ（Ｍｕｌｔｉｖｅｒｓｉｏｎ　Ｃｏｎｃｕｒｒｅｎｃｙ　Ｃｏｎｔｒｏｌ）を用いて常に一貫した読み出しデータを有する様々なデータサービスインスタンスを介してシステム内の並列トランザクションを実行するように拡張される。
【０１５６】
データ処理は、データサービスインスタンスへの原子メッセージを介して行われ、全体データジョブ（ｊｏｂ）を含むメッセージと共に発生する。例えば、ジョブには数個のレコード及び属性を読み出したり、従属データ（ｄｅｐｅｎｄｅｎｔ　ｄａｔａ）に基づい

たデータのアップデート又は挿入、あるいはタスクの組合せを含んでもよい。データサービスインスタンスは、全てのバッキング（ｂａｃｋｉｎ）、トランザクションデータストアーにわたって２フェーズコミットトランザクション（ｔｗｏ－ｐｈａｓｅｄ　ｃｏｍｍｉｔ　ｔｒａｎｓａｃｔｉｏｎ）として実行する。

【０１５７】

Ｔｅｒｅｏｎモデルは次の技術によってデータの一貫性を保証する。

ａ）読み出しデータのどのセットにもバージョンＩＤがある。

全てのレコード（アップデート及び従属挿入）は、このバージョンＩＤが楽観的トランザクションとして全ての関連データに対して最新であるかを検証する。これは、様々なアカウント属性（例えば、許可（ｐｅｒｍｉｓｓｉｏｎｓ）、残高（ｂａｌａｎｃｅ）、及び通貨データ（ｃｕｒｒｅｎｃｙ　ｄａｔａ））を取得するために３つのソースがレコードを読み出す場合、このデータのクラスタが一貫したバージョンＩＤが存在することを意味する。これらの値のいずれかがアップデートされるか、従属データがレコードされれば（例えば、金融の振込み）、バージョンＩＤが最新バージョンであるかが再び確認され、レコードが異なる場合（通貨の仮定が変更されたり、為替レートが変更されるなど）、レコードは全体として完全に失敗する。ダウンストリームサービスは、必要に応じて、重要な方式によりトランザクションを変更するか否かを再度読み出し、評価する。そうでなければ、トランザクションは再び提出される。再びトランザクションが失敗した場合、これは設定可能な再試行回数を超えて深刻なエラーが発生するまで繰り返す。深刻なエラーは、通常の状況ではきわめて珍しい。

【０１５８】

大多数の実際のシナリオにおいて、大量のトランザクション容量及びアカウントの多様性があっても、失敗した楽観的なトランザクションは発生しない。まれに、データは損傷することなく、処理のオーバーヘッドも最小限に抑えられる。このＭＶＣＣ／楽観的なモデルは、使用されているプラットフォームが（例外的な状況で要求される規制上の削除を除く）永久履歴データベースの場合、削除されたレコードに対しても完全に保護する。

【０１５９】

ｂ）与えられたデータパーティション（これはデータサービスの水平拡張とは別途の概念である）のためにプラットフォームへのレコード

多くのデータサービスインスタンスは、１つのデータパーティションから読み出し及び１つのデータパーティションでレコードされ、単一のデータサービスインスタンスは、複数のデータパーティションから読み出し及び複数のデータパーティションで全て格納する。全ての読み出し及びレコードは、必要に応じて、１つ以上の冗長動作バックアップ（ｒｅｄｕｎｄａｎｔ　ｏｐｅｒａｔｉｎｇ　ｂａｃｋｕｐ）と共に、単一のマスタトランザクタクインスタンス（ｓｉｎｇｌｅ　ｍａｓｔｅｒ　ｔｒａｎｓａｃｔｏｒ　ｉｎｓｔａｎｃｅ）２２２を介して発生する。しかし、単一のインスタンスのみ常に活性化する。これは、トランザクション及び因果的な妥当性が全ての状況（例えば、ネットワーク分割中、又は短時間の通信遅延中にスキューが発生しない）下で保持されることを保障する。このトランザクタは、全ての楽観的なトランザクションが有効であるか否かを確認し、該当のインスタンスに対して文脈上重要なものとしてアップデートされた最新情報でデータサービスインスタンス内のキャッシュ管理者を持続的にリフレッシュする。

【０１６０】

ｃ）選択的なデータ分割

単一トランザクタに制約されれば、極めて大きいＴｅｒｅｏｎインスタンスの拡張性を潜在的に制限される可能性がある（単一の組織が地域ごとに多重Ｔｅｒｅｏｎインスタンスを管理できるということを理解）。データ分割は、Ｔｅｒｅｏｎデータサービスクラスタがドメインごとに構成されたＴｅｒｅｏｎ規則に基づいて、トランザクタドゥル２２２又はデータストアー２２４にわたってデータを分割できるという概念である。Ｔｅｒｅｏｎプラットフォームは、現在、異種、多重コンポーネントハッシュ戦略として、次のパーティショニング規則を支援する。

【０１６１】
ｉ）与えられた要素又は上位要素の対象データにハッシュ（例えば、親レコードにより詳細ハッシュ）。高性能ハッシュは、パーティション数と同じカーディナリティ（ｃａｒｄｉｎａｌｉｔｙ）を有する。
【０１６２】
システムは、再調整（ｒｅｂａｌａｎｃｉｎｇ）を現在提供しないため、将来の実現で再調整が提供されても、現在の実現でハッシュは前もって行う必要がある（起源の日時によるハッシュを含む多重部分規則（ｍｕｌｔｉ－ｐａｒｔ　ｒｕｌｅ）を用いてパーティションが現在追加されることができる）。
【０１６３】
ｉｉ）与えられた要素又は任意の上位要素（例えば、挙げられた地理的な地域ごと、Ａ～Ｋ又はＬ～Ｚなど姓別、通貨別、その他）のターゲットされたデータのハッシュが構成されたデータ。
【０１６４】
データターゲットハッシュは、アルファニューメリック（ａｌｐｈａｎｕｍｅｒｉｃ）、ユニコード（ｕｎｉｃｏｄｅ）、及び他の文字コード範囲、整数範囲、浮動小数点範囲、及び挙げられたセットを支援する。
【０１６５】
ｉｉｉ）上記の組合せ
実現例において、二文字Ａ及びＢは、地理的な地域全体にわたって該当地域の２つの部分を示す数字１及び２で共通の２つの個別データセットを示す。例えば、単一パーティション規則は、地理的な地域のような最上位レベルパーティション１ＡＢ及び２ＡＢ間の分割を支援し、次に、アカウント番号ハッシュを介してＡ及びＢサブパーティション間の分割をさらに支援する。
【０１６６】
ｄ）単一のデータサービスインスタンスを介して実行される単一ジョブは、複数のデータパーティションを横断し、多重トランザクタドゥルにより完了され、複数のデータストレージノードに存続する。
【０１６７】
これは明白なデータ無欠性複雑さを示す。しかし、データの無欠性は、トランザクションの全ての構成要素が単一の２フェーズコミットラッパー（ｓｉｎｇｌｅ　ｔｗｏ－ｐｈａｓｅｄ　ｃｏｍｍｉｔ　ｗｒａｐｐｅｒ）にバインドされていることにより保障される。全ての永久的なノード及びアクターに対するトランザクションの全体が、全体として完了又は失敗し、全ての同じバージョンの保証を提供する。
【０１６８】
アーキテクチャーの設計の合流の結果、システムは完全にトランザクション的に安全で、冗長性が高く、水平的かつ垂直的に拡張性が高くなる。レコードトランザクション（多くのシナリオで小さいパーセントの処理を含む）は、パーティションごとに単一トランザクタのトランザクション上の必要によって制限されるが、規則ベース分割の付加（特に、優れたデータ要素）は、分岐インスタンスを検討する前に、概念的に無制限のレベルでシステムを拡張できる柔軟性を提供する。
【０１６９】
Ｔｅｒｅｏｎデータストアーの実現
Ｔｅｒｅｏｎインフラストラクチャーは、１秒当たり１０００００００を超えるＡＣＩＳ保障トランザクションを処理する。これは、個別の読み出し及びレコードアクセスチャネル２２６と共に、ストレージレイヤに対する高性能キー／値分散データベースを用いて分散したデータベース又はデータベース２２４上にデータストアーレイヤ２２０を抽象化又は他の方法により実現することによって達成される（これは、Ｔｅｒｅｏｎデータサービスによって抽象化からストレージレイヤに対する直接データベース使用に至るまで全ての深層レベルであり得る）。Ｔｅｒｅｏｎのデータストアーの使用及び構成は独特である。

【０１７０】
データサービスレイヤは、そのオーダーメード化データ交換モジュールを介してデータストアーレイヤと通信する。データベース自体は、データストアーレイヤ２２０によって処理されるＡＣＩＤ保証を提供する必要が全くない。これがレコードプロセスを著しく遅延させるため、それらはグラフ機能を提供する必要もない。データストアーレイヤ２２０は、異種データレイヤに対するインタフェースを提供し、システムの他の部分が必要とするインタフェース機能を提供する。したがって、書き込み効能は高速のセル及び列構造を提供する一方、読み出しインタフェースは、グラフィックインタフェースを提供し、マイクロ秒で分散データストアーをトラバースできるようにする。

【０１７１】
データストアーレイヤは、コアデータストアーデータベースの上にＳＱＬインタフェース及びグラフィックインタフェースレイヤを提供し、Ｔｅｒｅｏｎを区分する多くの重要なアーキテクチャー長所を提供する。各クライアントインスタンス（Ｔｅｒｅｏｎデータサービスインスタンス）２１４は、該当インスタンスに対する全てのホットデータのキャッシュされた表現を含むインメモリ／インプロセスデータベースエンジンをホストする。その結果、インスタンスはデータベースエンジン及び全ての現在のトランザクションデータのキャッシュされた表現、各現在のトランザクションの状態、及び該当インスタンスが動作中の機械又は機械の異なる高速メモリ又はそのＲＡＭの部分内のインスタンスの現在状態に関する全ての異なる情報をホストする。

【０１７２】
これにより、Ｔｅｒｅｏｎデータサービスが極めて高速レートで多くの読み出し指向のタスクを容易にし（１秒当たり、インスタンスごとに数百万の分離されたクエリ、ここで、ホット関連データはローカルでキャッシュされる）、達成される性能レベル以上の重要度（ｍａｇｎｉｔｕｄｅｓ）は、外部データベースシステムへの外部又は機械以外の要求を直列化して行う。データがインプロセスキャッシュにない場合は、キーバリューストアーから検索される。

【０１７３】
ＭＶＣＣバージョンシステムは、同時性を管理するために使用され、データレイヤの属性はデータが決して削除されないこと（規定準拠のための強制削除の他）、システムは、データシステムの存続期間中全てのレコード変更の全ヒストリーを保有する。これによって、「ａｓ　ｏｆ」クエリ、及び全てのプラットフォーム変更の監査のような簡単な作動を可能にする。

【０１７４】
データレイヤの書き込み実現は、単一の共有トランザクタを使用し、全てのデータ変更は一連の高速シーケンスで処理される。これはトランザクションの有効性、一貫性、多くのデータベースプラットフォームで負担となる加重値である変更同時性オーバーヘッドを最小化する。トランザクタクの設計は、ホットバックアップリダンダンシーモデルを使用する。トランザクタクプロセスが変更されれば、これは全ての活性クエリエンジン（この場合、Ｔｅｒｅｏｎデータサービスに存在）を通知し、必要に応じて、インメモリキャッシュをアップデートする。

【０１７５】
この設計は、データストアーのサイズに関係なく、読み出し、書き込み、及び検索に対するマイクロ秒の待ち時間を提供する。また、動作に影響を与えることなく、コンポーネントのアップデート及び交換を可能にするモジュラー構成を提供する。このデータストアーは、既存となる実現から抽象化され、Ｔｅｒｅｏｎデータサービスの他のストアーに代替されてもよい。

【０１７６】
データストアーレイヤが悲観的なＡＣＩＤ保証２２６で動作するように設定された場合、すなわち、次のトランザクションに進む前にレコードを書き込んだことを確認するための追加ステップを入れる場合、これは短い遅延を追加するが、ＡＣＩＤ一貫性及びデータ

無欠性の絶対的な保証を提供する。
【０１７７】
この設計の利点は、データレイヤがレコードを書き込んでトランザクションを完了したことをデータレイヤが確認するまで進行できないため、ＡＣＩＤ保証を提供するのである。
【０１７８】
これは、例えば、銀行、支払い、及び因果関係を維持しなければならない他のトランザクションタイプでは、最終的な一貫性により発生した問題が除去されることを意味する。また、ＡＣＩＤ保証で設計することで、銀行システムが不一致なプロセスを発見するとき、不足分を補うための調整アカウントに対する必要性もなくなる。リアルタイム処理は、最終的な一貫性システム上で調整プロセスが発生する時間遅延も除去されることを意味する。
【０１７９】
このプラットフォームの設計は、汎用ハードウェア上の極めて高いレベルのリダンダンシーと安定性、及び優れた拡張性（垂直及び水平的の両方）を提供する。トランザクタクシステムの可能性のある限界に対する理論的な懸念は、それらの限界を克服するためにデータサービスで分割プラットフォームを構築したが、多くのシナリオの下では該当のプラットフォームを使用する必要がない。
【０１８０】
ルックアップ／ディレクトリサービス
Ｔｅｒｅｏｎシステムは、ユーザ又はデバイス２１８がどのようなサーバに登録されるか、又は、特定の機能、リソース、設備、トランザクションタイプ、又は、他のタイプのサービスを提供しているサーバを識別するシステムにおいてクリデンシャル及び情報のディレクトリであるディレクトリサービス２１６を有する。ディレクトリサービスは、特定ユーザに関する様々な異なるタイプのクリデンシャルを格納するため、ユーザ２１８の複数の認証方法を可能にする、例えば、ユーザ２１８は、自分のモバイル番号、メールアドレス、地理的位置、ＰＡＮｓ（ｐｒｉｍａｒｙ　ａｃｃｏｕｎｔ　ｎｕｍｂｅｒｓ）などを用いて認証され、毎回認証する必要がないようにデータをキャッシュする。
【０１８１】
ディレクトリサービス２１６は、基本サービス、サーバ、及び実際のユーザアカウントからユーザの認証ＩＤを分離する抽象化レイヤを提供する。これは、ユーザ２１８又はマーチャントがサービスにアクセスするために使用できるクリデンシャル、及びＴｅｒｅｏｎがサービスそのものを行うために必要な情報間の抽象化を提供する。例えば、支払いサービスでは、ディレクトリサービス２１６は、単にモバイル番号のような認証ＩＤ、サーバアドレスと共に恐らく通貨コードをサーバアドレスとリンクさせるのであろう。ユーザ２１８が銀行アカウントを保有しているか否か、ユーザ２１８がどの銀行を使用しているかを決定する方法は全くない。
【０１８２】
システムアーキテクチャーにより、Ｔｅｒｅｏｎが既存システムの範囲を簡単に越える様々な新しいサービス又は機能を提供する。
Ｔｅｒｅｏｎシステムアーキテクチャーは、拡張可能でリダンダントシステムを許容するため有効である。銀行コアシステムは、例えば、カード管理、Ｅコマース、モバイルの支払いなど、個別チャネル専用のモジュールを提供する傾向がある。これはサイロ（ｓｉｌｏｓ）を強化され、そのＩＴシステムの複雑性を増加させる。このような複雑性は、銀行が自身のサービスやシステムを定期的にアップデートできない理由の１つである。
【０１８３】
Ｔｅｒｅｏｎは、高度な構成及びオーダーメード可能なモジュラーアーキテクチャーで全てのデバイス及び全てのユースケースを支援されるように設計されている。この核心は、上述したＳＤＡＳＦ１０４、高レベルの抽象化、ビジネス規則エンジン１０６である。これは、Ｔｅｒｅｏｎの柔軟性を可能にする拡張可能なフレームワークの組合せである。

【０１８４】
Ｔｅｒｅｏｎは、運営者が標準キャリアグレードシステム（ｓｔａｎｄａｒｄ ｃａｒｒｉｅｒ－ｇｒａｄｅ ｓｙｓｔｅｍｓ）を使用し、様々なトランザクションタイプを提供及び支援する。Ｔｅｒｅｏｎは、トランザクションが認証を必要とするか否かに関係なく、全てのトランザクションを支援するのであろう。
【０１８５】
スペシャルプロセス
スペシャルプロセス２０８は、データサービスの機能を理想的に活用する。しかし、独特の要求事項がコアデータサービスを変更又は拡張する正当化しないインスタンスがあるため、データライブラリーはデータから直接もってくるためにスペシャルプロセス内に活用される。例えば、これはＡＭＬ（ａｎｔｉ－ｍｏｎｅｙ ｌａｕｎｄｅｒｉｎｇ）、ＣＲＭ（ｃｕｓｔｏｍｅｒ ｒｅｌａｔｉｏｎｓｈｉｐ ｍａｎａｇｅｍｅｎｔ）、又はＥＲＰ（ｅｎｔｅｒｐｒｉｓｅ ｒｅｓｏｕｒｃｅ ｐｌａｎｎｉｎｇ）機能のようなグラフィック機能プロセスを含んでもよい。
【０１８６】
複数のサービス
各サービスがモジュールであるため、Ｔｅｒｅｏｎのモジュラー構造は様々なタイプのサービス及びデバイスを支援し得る。例えば、支払いにおいて、この構造はＴｅｒｅｏｎが銀行、請求カード、クレジットサービス、クレジットユニオン、デビットサービス、従業員制度、ｅＰｕｒｓｅ、ロイヤリティー制度、メンバーシップ制度、小額金融、前払い、学生サービス、発券、ＳＭＳの知らせ、ＨＬＲ検索などを含んで複数の支払いタイプ及びデバイスを支援することを可能にする。
【０１８７】
マルチエンドポイントデバイス（Ｍｕｌｔｉｐｌｅ ｅｎｄ－ｐｏｉｎｔ ｄｅｖｉｃｅｓ）
Ｔｅｒｅｏｎモジュラー構造により、磁気ストライプカード、スマートカード、フィーチャーフォン、スマートフォン、タブレット、カード端末、ＰＯＳ（ｐｏｉｎｔ ｏｆ ｓａｌｅｓ）端末、ＡＴＭ、ＰＣ、表示画面、電子アクセス制御、Ｅコマースポータル、リストバンド及びその他のウェアラブルなどを含み、直接または間接的に通信可能な全てのエンドポイントデバイスを支援する。
【０１８８】
多重データベース（Ｍｕｌｔｉｐｌｅ ｄａｔａｂａｓｅｓ）
モジュラー構造は、システムが１つのデータベースに制限されない点で異なる利点がある。代わりに、様々なデータベースは、問題のデータベースに固有モジュールでそれぞれ接続され得るため、特定の目的に特定データベースを使用したり、複数の異種データベースにわたるデータレコードの組合せを使用する。
【０１８９】
ライセンスサブシステム２１０の実現は、これが提供する許可及び認証の利点に加えて、ライセンス目的のための認証書機関のその使用において新規である。各モジュールは、相互の主張を信頼する代わりに共有データベースとの簡単な認証を使用するか、各接の続構築時に個別ライセンスサーバへの委任（性能及び安定性のオーバーヘッドを従う）がモジュール基幹システムに対する最も一般的な実現パターンである。Ｔｅｒｅｏｎで、ライセンスサブシステムは、モジュール間の接続が本質的に安全で、最小限の性能及び安定性オーバーヘッドでアクターに関する信頼できる検証済みのメタデータを有することを保障できる。
【０１９０】
また、この実現は、ライセンスサーバの侵害（compromise）のインスタンス内潜在的な脆弱性の範囲も制限する。従来の展開では、このような侵害は全てのコンポーネントの大規模な再構築に値するのであろう。Ｔｅｒｅｏｎモデルでは、新しい仲介者の署名認証書を要求する（ハードウェアセキュリティーモジュールによって保護されない場合）時間べ

ースの露出がある。事前感染が付与された既存の全ての認証書は古く、正常なスケジュールに更新され得る。新しい認証書は新しい権限の下で付与され、その他の不正な認証書は侵害されているとして拒否される。この露出ウィンドウ制御は、最悪のシナリオに役に立つ。ライセンスサーバが保有しているデータは、ハードウェアセキュリティーモジュールに理想的に保管される署名認証書の個人キーの外部にある、完全に権限のない情報である。

【０１９１】

また、Ｔｅｒｅｏｎの設計は、モバイル又はＩｏＴデバイスのようなエンドポイントデバイスを、そのようなサーバネットワーク一部として他のＴｅｒｅｏｎサーバと通信する小型のＴｅｒｅｏｎサーバと組み合わせることができる。それらはデータを収集し、処理を調整するためにＴｅｒｅｏｎライセンスサーバ２１０、及び恐らく１つ以上の運営者が運営するＴｅｒｅｏｎサーバと通信するのであろう。それにもかかわらず、エンドポイントデバイス及びＴｅｒｅｏｎサーバの間の区別（全ての区別はデバイス及びサーバが置かれているユースケースにのみ基づく）は抽象的なものである。

【０１９２】

ハッシュチェーン

ブロックチェーンの大きい短所の１つは、ブロックチェーンが以前の全てのトランザクションに対する監査を格納することである（すなわち、認証のために用いられるブロックチェーン内のトランザクションヒストリーを判断できる）。これはブロックチェーンのサイズが最終的に大きくなって現実的な時間フレームで管理できないことから、ブロックチェーンアクセス方式が無限に拡張できないことを意味し、一方、各ブロックのサイズがブロックチェーンが登録できる秒当たり最大トランザクションを制限することを意味する。

【０１９３】

第２の短所は、トランザクションヒストリーがブロックチェーンにアクセスできる人であれば誰でも使用できることである。そのため、トランザクション当事者が誰であるかを確認することができる。そのため、プライバシー及び／又は機密性の最も重要な要求事項である意味のある全ての処理において、ブロックチェーンを使用することに対する重大なプライバシー及び規制上の問題を提示する。

【０１９４】

更なる短所は、ブロックチェーンがトランザクションの結果又は最終レコードをハッシュすることができず、実際のプロセス又はトランザクション自体のステップを検証できないことにある。

【０１９５】

ここに開示されたハッシュチェーンは、トランザクション当事者間のレコードを非公開にし、Ｔｅｒｅｏｎ全てのユーザ（それが公開または非公開のネットワーク上で動作するに関係なく）を含む分散した認証ネットワークを提供するため、特定ハッシュアクセス方法を使用して上記のような問題を克服しようとする。

【０１９６】

これは、第３者に基盤となる通信のコンテンツを公開せず、公開及び非公開ネットワークにわたってリアルタイム動作する分散チェーンの持続的な構成によって達成される。これは、分散したハッシュ又は元帳の標準モデルと直接的に対照される（ここで、全ての当事者は、全ての通信コンテンツをそれが該当通信に対する当事者であるか否かに関係なく報告、受け入れる）。

【０１９７】

ハッシュチェーンがゼロ知識証明を含むプロトコルを使用すると、トランザクションの各ステップ及び情報又はトランザクションのステップによって生成された結果を認証できる。

【０１９８】

実現は、同一の中間ハッシュを生成する通信に対する当事者、又は、同一の通信に対して固有な中間ハッシュを生成する、それが発生する可能性がある。また、この構造により

、ハッシュチェーンの無欠性に影響を与えることなく、既存のアルゴリズムが廃止されることまら、当事者は新しいハッシュアルゴリズムに移行できる。これは、ブロックチェーンのような既存のライブソリューションで用いられるアルゴリズムをアップデート又はアップグレードする困難さと直接的に対照される。

【０１９９】

Ｔｅｒｅｏｎは、トランザクションの各側面（アカウント）に対してハッシュ監査チェーンを生成する。ここで、

・Ｔｅｒｅｏｎはレコードに関するハッシュを生成し、該当レコードに対するハッシュを格納する。Ｔｅｒｅｏｎは、レコードを生成するアクションが完了すると、該当ハッシュを生成するが、レコードを生成するステップと該当ステップから発生する情報又は結果を使用するためである。

【０２００】

・Ｔｅｒｅｏｎは、現在のレコードに対するデータの一部として前のレコードに対するハッシュを使用する、そして、

・全てのレコードチェーンの最初のハッシュは、サーバの署名、Ｔｅｒｅｏｎが該当ハッシュを生成する日時、必要に応じてランダム番号を有するランダムハッシュであり得る。

【０２０１】

このレコードが２以上の当事者が関与するアクションのレコードであり、各当事者がこのアクションの側面のレコードを保有しなければならない場合、Ｔｅｒｅｏｎはそれぞれのアクションについて次のことを行う。

【０２０２】

・他の当事者又は当事者と各当事者のレコードのハッシュを共有する。

・そのハッシュを使用して、レコードハッシュを生成するＴｅｒｅｏｎに対する受信当事者のレコードの一部を形成する。

【０２０３】

・他の当事者又は当事者からのハッシュを含むレコードの中間ハッシュを生成する。

・各当事者が各アクションで他の当事者の一部をカプセル化したハッシュを有するように、該当中間ハッシュを他の当事者と共有する。

【０２０４】

・アクションのレコードに中間ハッシュを含む。

・アクションに対して格納し、次のレコードの一部として使用する最終ハッシュを生成する。そして、

・送信された各ハッシュ、又は、ゼロ知識証明を用いてプロトコルで生成された中間ハッシュを送信者のＩＤ又はＴｅｒｅｏｎ番号と関連づける。

【０２０５】

Ｔｅｒｅｏｎは、以下に説明するように、これに必要なＡＣＩＤ保証及びリアルタイムセッショントランザクション及び処理速度を提供できる。また、ブロックチェーンの普及により、この分野における開発は考慮されていないことを意味する。

【０２０６】

トランザクションが完了すると、ブロックチェーンは、トランザクションのレコードをハッシュする。ブロックチェーンに伝えられたレコードが実際にトランザクションそれ自体の本物レコードであり保証はない。基礎となるハッシュ構造は、動的及びリアルタイムトランザクションではない静的データの収集用として設計されているため、ブロックチェーンは、正直に行動するその運営者の大多数に依存するため、この方式に制限される。ブロックチェーンそれ自体は、最終的な一貫性しか提供できない点でさらに限界がある。ＡＣＩＤの一貫性は、トランザクションの時系列順ではなく、それらのトランザクションがブロックに組み込まれる順序、およびわずかに異なるトランザクションセットを含２つ以上のブロックが検出された場合のブロックチェーン内のフォーク管理のコンセンサスモデルによって決定される。

【０２０７】
図５は、４つのアカウント５０２、５０４、５０６及び５０８を含むハッシュチェーンの樹枝状特性（ｄｅｎｄｒｉｔｉｃ　ｎａｔｕｒｅ）を示す。アカウントは、同じサーバ上にあったり、個別サーバにあってもよい。各システムは、１つ以上のサーバを支援し、各サーバは１つ以上のアカウントを支援する。アカウントのある位置は関係ない。また、図５は、対のアカウント間に発生する５つのトランザクションを示す。アカウント５０２及び５０４との間に発生する２つのトランザクション、アカウント５０２及び５０６との間に発生する２つのトランザクション、及びアカウント５０６及び５０８との間に発生する１つのトランザクションがある。図面において、各ボックスは、列が最上位にあるアカウントに関するステップである。各ステップは、該当アカウント内における検索、又は、該当アカウント及び他に見えないアカウント又はシステムのような見えないアクション又はトランザクションを含む。これらのトランザクション又はアクションが何であるかは関係ない。重要なことは、それらが、この監査にＴｅｒｅｏｎシステムレコードを何かを含んでいることである。

【０２０８】
ステップ５１０において、Ｔｅｒｅｏｎシステムは、このアカウントに対する以前のハッシュｈ５０２を取得する。上述したように、最初のハッシュはサーバの署名、Ｔｅｒｅｏｎがハッシュを生成した日時、必要に応じて、ランダム番号のあるランダムハッシュである。Ｔｅｒｅｏｎは、ハッシュをステップ５１０で発生するトランザクション又はアクションに対するレコードに追加し、その次に、このトランザクションに対するハッシュｈ５１２を算出するシードとして使用する。このステップでのレコードは、ｈ５０２及びｈ５１２を含む。

【０２０９】
ステップ５１２において、システムは、アカウント５０４を保有するサーバとハッシュｈ５１０を交換する。これは、アカウント５０４に対するこのトランザクションに対するハッシュｈ５０４をレコードに追加し、中間ハッシュｈ５１２ｉを生成し、このレコードに追加し、その次に、アカウント５０４から中間ハッシュｈ５１４ｉ（後述するように、ステップ５１４で生成される）で交換する。次に、このハッシュをそのレコードに追加してハッシュｈ５１２を生成する。

【０２１０】
ハッシュｈ５１２は、アカウント５０２からステップ５１２まで、及びアカウント５０４からステップ５１４の中間段階までのハッシュチェーンを有効にした情報をさらに含む。レコードは、ｈ５１０、ｈ５１２ｉ、ｈ５１４ｉ、ｈ５０４、及びｈ５１２を含む。

【０２１１】
ステップ５１４において、システムは、アカウント５０２を保有するサーバとハッシュｈ５０４を交換する。これは、アカウント５０２からのハッシュｈ５１０をレコードに追加し、中間ハッシュｈ５１４ｉを生成し、これをレコードに追加し、次に、これをアカウント５０２からの中間ハッシュｈ５１２ｉと交換する。次に、それからこのハッシュをレコードに追加してハッシュｈ５１４を生成する。

【０２１２】
このチェーンは、アカウント５０２からステップ５１２まで、及びアカウント５０４からステップ５１４までハッシュチェーンを有効にした情報をさらに含む。
この順は、上述したように、まったく同じ方式に基づいてトランザクションに対するハッシュを生成するためにアカウント５０２、５０４、５０６及び５０８間の追加トランザクションのために実行される。例えば、ステップ５３４において、システムは、ステップ５２８で生成されたアカウント５０２に対する以前のハッシュｈ５２８を取得し、監査レコードを発生させるトランザクション又はアクション（見えない）に対するレコードにこれを追加し、このトランザクションに対するハッシュｈ５３４を生成する。このチェーンは、ステップ５３４までのアカウント５０２、ステップ５２６までのアカウント５０４、ステップ５３０までのアカウント５０６、ステップ５３０において、ｈ５３０を生成する

ために使用されたアカウント５０８からの中間ハッシュまでのアカウント５０８に対するハッシュチェーンを有効にした情報が含まれる。レコードｈ５３４及びｈ５２８を含む。Ｔｅｒｅｏｎは、ステップ５３０において、ｈ５２４から生成されたｈ５３０ｉを含むレコードからステップ５２８でハッシュｈ５２８を生成する。ハッシュｈ５２４には、ステップ５２４において、ｈ５２４を生成するために使用されたアカウント５０８から中間ハッシュまでのアカウント５０８を有効にした情報が含まれる。

【０２１３】

調整（Ｒｅｃｏｎｃｉｌｉａｔｉｏｎ）

詐欺師が以前のトランザクションのレコードを変更した場合、トランザクションが実行されないようにするため、最初の最後の「Ｎ」トランザクションについて調整できる。したがって、例えば、Ｔｅｒｅｏｎがステップ５２２で提示されるトランザクションを行う前に、これはアカウント５０２に対する以前「Ｎ」トランザクションまでステップ５１６、及び、恐らくステップ５１２などに対するハッシュをまず再算出することができる。監査追跡は、トランザクションのために最終ハッシュ値を再算出するための十分な情報を有する。同様に、アカウント５０４を保有しているシステムは、ステップ５２６、ステップ５２０などのためにハッシュを再算出してもよい。Ｔｅｒｅｏｎは、ステップ５２２のトランザクションのためにアカウント５０６に対する全てのハッシュを再算出する必要がない。

【０２１４】

ハッシュチェーンでは、レコードされたハッシュのいずれか１つが再算出されたハッシュと一致しない場合、これはレコードが認証なしに変更されたことを意味し、運営者はすぐに問題を調査したり追加トランザクションを遮断する。

【０２１５】

システムハッシュチェーン

また、システムハッシュは、各レコードに追加されてもよい。これはアクションがハッシュされたレコードが属するアカウントと関連があるかに関係なく、レコードのハッシュであり、ここで、シードはシステム上に以前のアクションのハッシュであろう。次に、システムハッシュが各アカウント内のハッシュチェーンに追加される場合、システム全体のハッシュチェーンが提供される。

【０２１６】

図６は、同一のシステム上の２つのアカウント６０２及び６０４を含むハッシュチェーンの樹枝状特性を図示して、全てのシステムイベントを記録する‘システムアカウント（ｓｙｓｔｅｍ　ａｃｃｏｕｎｔ）’は６０６である。システムはレコードが常駐する位置に関係なく、レコードを発生させる全てのアクションに対するレコードの新しいハッシュを生成する。ｈ６０６、ｈ６０８、ｈ６１２等システムハッシュがある。

【０２１７】

図６は、同じシステム上の２つのアカウント６０２及び６０４を含むハッシュチェーンの樹枝状特性を示し、全てのシステムイベントを記録する「システムアカウント」は６０６である。システムは、レコードが存在する場所に関係なく、レコードを発生させる全てのアクションに対するレコードの新しいハッシュを生成する。ｈ６０６、ｈ６０８、ｈ６１２などがシステムハッシュである。

【０２１８】

ステップ６０８において、Ｔｅｒｅｏｎはシステムの監査レコードでエントリをトリガーするアカウント６０２で見えないアクション又はトランザクションのレコードのハッシュを生成し（アカウント６０２に対するレコードは、該当アカウントに対する以前のレコードハッシュ、ハッシュｈ６０２を含む）、新しいシステムハッシュｈ６０２に対するｈ６０６を使用する。そして、システムは、トランザクションに対するレコードに対してこのハッシュをレコードし、ステップ６１０でアカウント６０２に対するハッシュｈ６１０を算出する。

【０２１９】

システムのコンピューティング性能が許容すれば、これはアカウントハッシュの動作を反映するシステムハッシュのために強力な変形を使用できる。

ステップ６１０において、Ｔｅｒｅｏｎは、システムアカウント６０６とハッシュｈ６０２をハッシュｈ６０６で交換する。これは、システムアカウント６０６からのハッシュｈ６０６をそのレコードに追加し、中間ハッシュｈ６１０ｉを生成する。システムの監査レコードへのエントリをトリガーし、レコードにハッシュを追加するアカウント６０２での見えないアクション又はトランザクションを完了した後、これを生成する。その次に、Ｔｅｒｅｏｎは、中間システムハッシュｈ６０８ｉでこの中間ハッシュを交換する。その後、これとｈ６０８をそのレコードに追加して新しいアカウントハッシュｈ６１０を生成する。

【０２２０】

ステップ６１２において、Ｔｅｒｅｏｎは、ステップ６０８で生成されたハッシュｈ６０８をアカウント６０２及び６０４と交換する。ステップ６１０で生成されたｈ６１０、及びｈ６０４をそのレコードに追加して中間ハッシュｈ６１２ｉを生成する。アカウント６０２及び６０４とそれらの中間アカウントシステムハッシュｈ６１４ｓｉ及びｈ６１６ｓｉ、及びアカウント６０２に対応する中間ハッシュｈ６１４ｉ及びアカウント６０４に対応するｈ６１６ｉで交換する。その後、新しいシステムハッシュｈ６１２を生成する。その後、システムはこのハッシュを記録する。

【０２２１】

ステップ６１４で、Ｔｅｒｅｏｎは、ステップ６１０で生成されたハッシュｈ６１０をシステムアカウント６０６と交換する。これはシステムアカウント６０６からのハッシュｈ６０８（ステップ６０８で生成）をレコードに追加し、中間アカウントシステムハッシュｈ６１４ｓｉを生成する。これは、アカウント６０４を用いてトランザクションを完了した後にハッシュを生成し（中間トランザクションハッシュｈ６１４ｉ及びｈ６１６ｉを交換）、これをレコードに追加し、それからこれを中間システムハッシュｈ６１２ｉで交換する。その後、これとｈ６１８をそのレコードに追加してアカウントハッシュｈ６１４を生成する。

【０２２２】

ステップ６１６において、Ｔｅｒｅｏｎは、システムアカウント６０６とハッシュｈ６０４を交換する。システムアカウントからのハッシュｈ６０８をそのレコードに追加し、中間アカウントシステムハッシュｈ６１６ｓｉを生成する。アカウント６０２を使用してトランザクションを完了した後これを生成し（そして、中間トランザクションハッシュｈ６１４ｉ及びｈ６１６ｉを交換）、ハッシュをそのレコードに追加し、それからこれを中間システムハッシュｈ６１２ｉで交換する。そのあと、これとｈ６０８をそのレコードに追加してアカウントハッシュｈ６１６を生成する。

【０２２３】

ステップ６１２において、システムがアカウント６０４に中間システムハッシュｈ６１４ｓｉを送信し、アカウント６０２に中間システムハッシュｈ６１６ｓｉを送信することが１つのオプションである。これは、これらのアカウントに対する最終レコードハッシュを意味し、ｈ６１４及びｈ６１６は３つの中間システムハッシュｈ６１４ｓｉ、ｈ６１４ｓｉ、及びｈ６１２ｉのレコードを含んでいるため、確実性の追加レイヤを提供する。

【０２２４】

システムハッシュチェーンは、個々のトランザクションの両側のハッシュのみならず、トランザクション全体的にハッシュチェーンを含み、ハッシュチェーンが大幅に強化する。

【０２２５】

Ｔｅｒｅｏｎが異なるシステム上にアカウント間のトランザクションを管理する場合、プロセスは、それらの各システムに対するステップ６０８及び６１０の場合と同様である。

【０２２６】

ライセンスサーバハッシュ（Ｌｉｃｅｎｃｅ　ｓｅｒｖｅｒ　ｈａｓｈｅｓ）
ハッシュは、個別Ｔｅｒｅｏｎシステム間に生成されたハッシュに関連がある。このようなシステムが相互作用することから、それらのシステムの全てのトランザクションを検証された情報を含んだハッシュツリーに参加できる。しかし、このようなシステムが相互作用する速度でしか増加しない。システムは一層進んで、各サーバがすぐにグローバルハッシュツリーに参加できることを保証する別のレイヤを構築することができる。これはブロックチェーンとハッシュチェーンを完全に区別する。

【０２２７】
ブロックチェーン運営者がプライベートブロックチェーンを設定すると、該当ブロックチェーンは他の全てと別に作動する。全般的な処理速度が向上するのは別の方法で提供できる任意のセキュリティーで失われるが、ユーザがトランザクションの有効性を検査するために大規模ブロックチェーンのネットワークに依存できないためである。セキュリティーに対するブロックチェーンの主張のうちの１つは、攻撃者がセキュリティーを侵害させるために多くのブロックチェーンネットワークのノードを侵害させなければならない（ノードの２５－３３％程度の侵害はブロックチェーンを侵害させるのに充分である）。単一プライベートブロックチェーンは定義によりその数を１つに減らす。

【０２２８】
ハッシュチェーンを使用すると、プライベートＴｅｒｅｏｎサーバ又はネットワークはさらにパブリックＴｅｒｅｏｎサーバ及びネットワークによって生成されたハッシュチェーンからの利点を取得できる。プライベートＴｅｒｅｏｎサーバ又はネットワークを運営するのは、運営者がＴｅｒｅｏｎシステムの認証強度に対して妥協しなければならないことを意味しないが、該当システムが依然としてグローバルハッシュチェーンのメンバーであるためである。これは単に、トランザクション（ライセンスサーバに関連したその以外）が該当システムに対して完全にプライベートに保持されるのであろう。

【０２２９】
これを達成するために、全てのサーバは他のＴｅｒｅｏｎサーバと相互作用するか否かに関わらず、ライセンスサーバと相互作用しなければならない。Ｔｅｒｅｏｎサーバが閉ループサーバとして動作し、該当のループが２つ以上のサーバから構成される場合に該当ループ内で他のＴｅｒｅｏｎサーバとのみ相互作用するのであろう。

【０２３０】
ライセンスサーバハッシュを追加することで、全てのサーバはライセンスサーバと相互作用すると同時に（基本的に毎日行わなければならない）グローバルサーバハッシュチェーンに参加する。ライセンスサーバハッシュは、基本的にＴｅｒｅｏｎサーバ及びライセンスサーバ間の２つのパーティートランザクションによって生成される。ライセンスサーバトランザクションは、Ｔｅｒｅｏｎサーバ間の基本データトランザクションに影響を及ぼさないということ以外にも、各サーバのためのシステムハッシュがライセンスサーバハッシュから派生した情報をさらに含むものであり、その反対の場合も同様である。

【０２３１】
図７は、ライセンスハッシュの樹枝状特性を示す。簡単な例で、システムサーバ７０２は、システムサーバ７０４及び７０６が相互接続する閉ループシステムである。３つのす全てはライセンスサーバ７０８と周期的に相互作用しなければならない。

【０２３２】
ライセンスサーバ７０８との最初質問において、各サーバはその公開キー、サーバが初めてライセンスが付与された日時、及びデータのランダムセットから最初のハッシュを生成する。

【０２３３】
ステップ７１０において、Ｔｅｒｅｏｎは、中間ライセンスハッシュｈ７１０ｉを生成するためにハッシュｈ７０８を生成し、それをレコードに追加し、これをサーバ７０２からの中間システムハッシュｈ７１２ｉと交換する。その後、これはこのハッシュをそのレコードに追加し、それからそのレコードに追加するライセンスハッシュｈ７１０を生成す

る。
【０２３４】
ステップ７１２において、Ｔｅｒｅｏｎは、中間システムハッシュｈ７１２ｉを生成するためにハッシュｈ７０２を生成し、それをレコードに追加し、これをライセンスサーバ７０８からの中間ライセンスハッシュｈ７１０ｉと交換する。その後、これはこのハッシュをそのレコードに追加し、そのレコードに追加するシステムハッシュｈ７１２を生成する。
【０２３５】
ステップ７１２において、Ｔｅｒｅｏｎは、中間システムハッシュｈ７１２ｉを生成するためにハッシュｈ７０２を生成し、それをそのレコードに追加し、これをライセンスサーバ７０８からの中間ライセンスハッシュｈ７１０ｉと交換する。その後、これはこのハッシュをそのレコードに追加し、そのレコードに追加するシステムハッシュｈ７１２を生成する。
【０２３６】
ステップ７１６において、Ｔｅｒｅｏｎは、中間システムハッシュｈ７１６ｉを生成するためにハッシュｈ７０４を生成し、それをライセンスサーバ７０８からの中間ライセンスハッシュｈ７１４ｉと交換する。その後、これはこのハッシュをそのレコードに追加し、そのレコードに追加するシステムハッシュｈ７１６を生成する。
【０２３７】
ステップ７１８において、Ｔｅｒｅｏｎは、中間ライセンスハッシュｈ７１８ｉを生成して、これをそのレコードに追加し、これをサーバ７０６からの中間システムハッシュｈ７２０ｉと交換する。その後、これはこのハッシュをそのレコードに追加し、そのレコードに追加するライセンスハッシュｈ７１８を生成する。
【０２３８】
ステップ７２０において、Ｔｅｒｅｏｎは、中間システムハッシュｈ７２０ｉを生成するためにハッシュｈ７０６を使用し、これをそのレコードに追加し、これをライセンスサーバ７０８からの中間ライセンスハッシュｈ７１８ｉと交換する。その後、このハッシュをそのレコードに追加し、そのレコードに追加するシステムハッシュｈ７２０を生成する。
【０２３９】
Ｔｅｒｅｏｎサーバトランザクションに対する３つのライセンスサーバは、次のような結果を取得される。
・ステップ７１２で生成されたハッシュｈ７１２は、以下の状態を検証する情報を含む。
【０２４０】
・中間ハッシュｈ７１０ｉまでライセンスサーバ７０８のハッシュチェーン
・ハッシュｈ７１２までサーバ７０２のハッシュチェーン
・ステップ７１６で生成されたハッシュｈ７１６は、以下の状態を検証する情報を含む。
【０２４１】
・中間ハッシュｈ７１４ｉまでライセンスサーバ７０８のハッシュチェーン
・中間ハッシュｈ７０２ｉｉまでサーバ７０２のハッシュチェーン
・ハッシュｈ７１６までサーバ７０４のハッシュチェーン
・ステップ７２０で生成されたハッシュｈ７２０は、以下の状態を検証する情報を含む。
【０２４２】
・中間ハッシュｈ７１８ｉまでライセンスサーバ７０８のハッシュチェーン
・中間ハッシュｈｋ７０２ｉｉまでサーバ７０２のハッシュチェーン
・中間ハッシュｈ７１６ｉまでサーバ７０４のハッシュチェーン
・ハッシュｈ７２０までサーバ７０のハッシュチェーン

・ステップ７１８で生成されたハッシュｈ７１８は、以下の状態を検証する情報を含む。

【０２４３】

・ハッシュｈ７１８までライセンスサーバ７０８のハッシュチェーン
・中間ハッシュｈｋ７０２ｉｉまでサーバ７０２のハッシュチェーン
・ハッシュｈｋ７０４ｉまでサーバ７０４のハッシュチェーン
・ハッシュｈ７２０までサーバ７０６のハッシュチェーン

したがって、ライセンス及びシステムハッシュは、それらのサーバが相互接続されたか、閉ループで動作するか否かに関係なく、ネットワーク内の全てのサーバ上にトランザクションを検証するようにする情報が含まれる。

【０２４４】

Ｔｅｒｅｏｎは、ライセンスサービスによって作成されたハッシュチェーンと同様の方法で動作するルックアップディレクトリサービスを用いて、同様のレイヤを実現できる。

オフライントランザクション（Ｏｆｆ－ｌｉｎｅ　ｔｒａｎｓａｃｔｉｏｎｓ）

このアプローチを用いて、オフライントランザクションはデバイスとそれらのサーバ間の持続的な通信リンクを除去されなければならないため、オンライントランザクションと同じ有効性をもう有し得る。したがって、センサ、モバイル支払い端末、及びその他のようなデバイスは、互いに通信でき、データをダウンロード及びアップロードするために一定の間隔でそれらのとサーバとを接続する。システムは接続環境と非接続環境との間で円滑に動作することができる。

【０２４５】

ハッシュチェーンは、デバイスが各サーバと通信できない間にあっても、それがオフライントランザクションに関与するか否かを決定するためにビジネス規則を使用し、その間のトランザクションを有効にして監査することを可能にする。デバイスは、それらが再度それらのサーバと接続されるとき、それらのサーバとそれらの監査及びトランザクションレコードを単に調整する。

【０２４６】

図８は、４つのデバイスのＴｅｒｅｏｎサーバから一時的にオフラインされる４つのデバイスを含むハッシュチェーンの例を示す。そのうち３個８０２、８０４及び８０６は視角化される（ステップ８２８で４番目デバイス８０８はチェーンと相互作用する）。

【０２４７】

デバイス間のオフライントランザクションを支援するために、デバイス自身が参加する各トランザクションのハッシュを生成する。デバイスがオンラインに戻ってサーバと通信するとき、該当デバイスはそのトランザクションに対するハッシュをサーバに送信する。

【０２４８】

トランザクションを開始するデバイスがオフラインであれば、トランザクションに対するハッシュを生成して該当ハッシュを格納する。また、それは、相手側のデバイス（これがトランザクション中であるデバイス）に該当ハッシュを送信することで、相手側デバイスは第１デバイスにこのハッシュを送信する。これは、上述したハッシュチェーンと同じ方式により達成される。デバイスは、ブルートゥース、ＮＦＣ、ローカルＷｉ－Ｆｉなどのような双方向チャネルを介して通信し得る。それらは、さらに異なる者が読み出すようにし、各トランザクションステージに対するこのコードをスクリーン上に掲示する。また、各デバイスは、自分のトランザクションレコードの署名済みの暗号化コピーを他のデバイスにも送信する。ここで、署名にはそのレコードの配信先サーバも含まれる。送信先サーバのみが該当レコードを解読できる。

【０２４９】

デバイスがそのＴｅｒｅｏｎサーバと通信を回復すると、該当デバイスはこのオフライントランザクション及びそれらに関するハッシュの暗号化されたレコードをサーバに送信する。また、それはまた、相手側からのレコードなど、それが保有する他のトランザクションのコピーを該当サーバに送信し、その後、サーバはそれらのレコード及びそれらに関

するハッシュを相手側デバイスが登録されているサーバに送信する。各デバイスは、トランザクションでこの部分を識別する自身の固有内部トランザクション番号（例えば、モノトニクカウンタによって生成されるもののような）を生成する。また、トランザクションがオンラインの場合、デバイスに接続されたサーバは、デバイスとサーバの両方が使用する固有のトランザクション番号を生成する。

【０２５０】

デバイスは、各トランザクションの因果関係を保持するために内部トランザクション番号と日時スタンプ、デバイスクロックスキュー（ｄｅｖｉｃｅｓ　ｃｌｏｃｋ　ｓｋｅｗ）に関する情報、及び他の情報を組み合わせる。それらの各サーバがトランザクション情報を受信すると、それらはトランザクションの順を再構成することができ、全てのデバイスに対するオンライン及びオフライントランザクションの両方の因果関係を保持する。

【０２５１】

図８を参照すると、ステップ８１２において、デバイス８０２はｈ８１２を生成するために、サーバ８１０からのハッシュｈ８０２、以前のレコードハッシュ、及びハッシュｈ８１０を含むトランザクションのレコードをハッシュする。その次に、このハッシュをサーバ８１０に伝達するが、ここで、該当ハッシュはステップ８１４でｈ８１４を算出するために用いられるレコードの一部を形成する。デバイス８０２は、このＴｅｒｅｏｎサーバ８１０に接続されることを意味する。ここで、オンラインである。ステップ８１４において、Ｔｅｒｅｏｎは、サーバ８１０に対する以前のハッシュｈ８１０を使用し、これとｈ８１２をレコードに追加し、その次にｈ８１４を算出する。レコードはｈ８１０、ｈ８１２、及びｈ８１４を含む。

【０２５２】

運営者がシステムハッシュを含むためにＴｅｒｅｏｎを構成した場合、上述したように、ハッシュｈ８１４を算出する前にこれをレコードに追加する。その後、レコードは、ｈ８１２、ｈ８１０、関連する場合は中間システムハッシュ、及びｈ８１４を含んでもよい。

【０２５３】

ステップ８１６において、デバイス８０２はオフラインであるため、サーバ８１０に接続されることができない。これはデバイス８０４とトランザクションする。デバイス８０４も、この各Ｔｅｒｅｏｎサーバからオフラインである。デバイス８０２及び８０４は、ステップ８１８でデバイス８０２からの中間ハッシュｈ８１６、デバイス８０４からの中間ハッシュｈ８１８、デバイス８０２からのハッシュｈ８１６、及びデバイス８０４からのハッシュｈ８１８を生成するために、上述したハッシュの手続きに従う。デバイス８０２及び８０４は、自分のオフライン公開キーで自分のハッシュに署名し、他のデバイスにこれを該当トランザクションに対するレコードの暗号化されたコピーと共に伝達する。これはデバイス８０２の最初のオフライントランザクションであり（これはサーバ８１０と接触がなくなったため）、デバイス８０４の最初のオフライントランザクションである（このサーバと接触がなくなったため）。管理者は、アプリケーションがオフラインでトランザクションする固有のデバイスに、最後のｎ個までのトランザクションを送信できるようにシステムを設定する。

【０２５４】

この順はデバイス８０２及びデバイス８０４の間、及びデバイス８０４及びデバイス８０６の間のチェーン内の更なるトランザクションのために繰り返す。このようなトランザクションで、デバイス８０２及び８０４は、それぞれ既にコピーを保有しているため、以前のトランザクションについてハッシュ及びレコードを交換する必要がない。

【０２５５】

デバイス８０２は、ステップ８３０において、そのサーバ８１０と接触を再設定するまでこの方式で続けて動作する。デバイス８０２は、オフライントランザクション及びそれに関するハッシュ（一例でとして、ステップ８１６、８２２、及び８２６でそれぞれ生成されたｈ８１６、ｈ８２２、及びｈ８２６）の暗号化されたレコードの全てをアップロー

ドする。また、これはデバイス８０４、８０６、及び８０８に対して、保有する暗号化されたトランザクションレコード及びハッシュをアップロードする。サーバはこれらを格納し、各デバイス８０４、８０６及び８０８に対応するサーバにアップロードする。サーバ８１０は、デバイス８０４、８０６、及び８０８からのハッシュのレコードとこのアップロードをトランザクションで登録して、ステップ８３２でハッシュｈ８３２を生成する。デバイス８０２は、デバイス８０４、８０６、及び８０８からのハッシュのレコードとそれぞれのトランザクションレコードとをクリアし、ステップ８３０でハッシュｈ８３０を生成する。

【０２５６】

デバイス８０２は、ステップ８２０において、ハッシュｈ８２０及びｈ８０８を発生させるデバイス８０６及び８０８間のトランザクションに対するハッシュ及び暗号化されたレコードを保有する。この例では、オフライントランザクションがいくつ発生したか分からないため、ｈ８０８を用いて該当トランザクションのために生成されたデバイス８０８に対するハッシュを参照する。

【０２５７】

サーバ８１０は、それがデバイス８０２から受信したオフラインレコードをデバイス８０４、８０６、及び８０８、ならびにそれらのトランザクションを含む任意の他のサーバから受信するものと調整する。サーバ８１０は、どのようなサーバからレコードを受信したか分かるのであろう。それは、これらはデバイス８０２に関するトランザクションのレコードが送信されたサーバに対応するためである。デバイス８０２は、デバイス８０８からのレコードを受信することを期待せず、デバイス８０２がデバイス８０８とトランザクションしていないためである。デバイス８０４又は８０６が他のサーバに接続されたオフラインデバイスとトランザクションした場合、サーバ８１０は、これらの他のサーバから追加レコードを受信することができる。

【０２５８】

サーバ８１０は、該当トランザクションの注文及び番号付けのためにトランザクションレコード及び署名に対する日時スタンプを使用して、それらをオフライントランザクションで表示する。

【０２５９】

オフラインモードは、様々な変形を提示する。最初は、中間オフラインハッシュを使用せず、各デバイスの以前トランザクションに対するハッシュを簡単に使用することにある。これは階層の確実性を失うが、よく作動する。第二に、オフライントランザクション専用のデバイスハッシュを生成することにある。これはオンライントランザクションを僅かに単純化するが、やはり階層の確実性は失う。第３に、変形は、特定のオフライン公開キーでオフライントランザクションのレコードに署名するのではなく、単にデバイスキーを用いて全てのレコードに簡単に署名することにある。アカウント監査追跡にレコードされるため、サーバ及びデバイスの両方は、どのトランザクションがどのオンライン及びオフラインであるかを確認できる。しかし、デバイスに対して個別キー及び一連のトランザクション番号を実行すると、オフライントランザクションとオンライントランザクションを示すことは簡単になる。

【０２６０】

第４の変形は、接続されたデバイスからオフライントランザクションのレコードを受信するとき、各サーバがそれらのレコードが適用される全てのサーバがそのサーバからのレコードを予想できるよう通知することにある。例えば、図８に示されたオフラインダイヤグラムにおいて、デバイス８０４が後でサーバに接続し、デバイス８０６が他のデバイス（図示せず）とトランザクションすることを仮定する。デバイス８０４がサーバに接続すると、該当サーバは、デバイス８０２に関するレコードをサーバ８１０に送信する。デバイス８０４は、他のデバイスとオフラインでトランザクションせず、他のデバイスについてのオフラインレコードを保有しない。一方、サーバ８１０は、デバイス８０４に対するレコードをデバイス８０４に対応するサーバに送信し、該当サーバにデバイス８０６から

同じレコードのコピーを受信することを予想できるように通知する（ステップ８２６及び８２８においてトランザクションの間にデバイス８０２はこれをデバイス８０６へ伝達）。同様に、デバイス８０６がそのサーバに接続すると、該当サーバは、デバイス８０２に関するレコードをサーバ８１０へ、デバイス８０４に関するものをデバイス８０４に対応するサーバへ、デバイス８０８に対するものをデバイス８０８に対応するサーバへ、他のデバイスに対するものを各サーバに送信するのであろう。また、これはデバイス８０２（サーバ８１０）及び８０４に対応する両方のサーバに他のデバイスに対応するサーバからのレコードを予想するよう通知する。

【０２６１】

ハッシュチェーンを使用しても、Ｔｅｒｅｏｎにこれ以上の負荷がかかることはない。１つのアクションは２以上の当事者が関与することはほとんどなく、そのアクションは、一般的に、一対多（ｏｎｅ－ｔｏ－ｍａｎｙ）の送信になり、それ自体は一対一（ｏｎｅ－ｔｏ－ｏｎｅ）の送信の集合となる。また、多対一（ｍａｎｙ－ｔｏ－ｏｎｅ）の送信は、一般的に一連の一対一の送信であり、これは単に２者間のアクション集合である。

【０２６２】

レコードの修正（Ａｍｅｎｄｉｎｇ　ｒｅｃｏｒｄ）

ユーザがレコードを修正する場合、Ｔｅｒｅｏｎはオリジナルレコードを上書きしない。代わりに、Ｔｅｒｅｏｎは、単に修正されたレコードを含んでいる新しいレコードを生成し、これは、該当レコードが再び修正されるまでＴｅｒｅｏｎが示すバージョンである。修正はアクションである。これは、前のトランザクション結果を効率よく修正する全ての金融及びトランザクションレコード（支払いなどのようなトランザクションの結果）で発生する。また、これは、運営者がＥメール、医療記録などのような他のレコードタイプを管理するためにＴｅｒｅｏｎのサブセットを使用する場合にも発生する。これにより、Ｔｅｒｅｏｎはレコードの各バージョンのコピーを保持する。

【０２６３】

裁判所、又は、一般的な法の運営において、運営者がレコードを完全に消したり、オリジナルレコードを修正することを要求する状況が発生しえる。このような状況で、Ｔｅｒｅｏｎは、オリジナルレコードの内容、及び恐らく関連するレコードの内容を削除又は修正する。Ｔｅｒｅｏｎは、後続ハッシュを無効にすることなくこれを達成できる。

【０２６４】

Ｔｅｒｅｏｎが履歴レコードを削除又は修正する場合、次のようになる。

・Ｔｅｒｅｏｎがレコードを削除又は修正する前に、該当レコードを修正又は変更されていないことを確認し、該当レコードのハッシュを再生性し、再生成されたハッシュを記録する。

【０２６５】

・削除又は修正されたレコードの内容、及び削除又は修正の理由をオリジナルレコード新しいフィールドに記録する。

・レコード内の関連フィールドの削除又は修正、及び該当削除又は修正の日時を追加する。

【０２６６】

・該当レコードにその新しいハッシュ生成する。

・新しいハッシュを記録する。

この順に従うことで、Ｔｅｒｅｏｎはどのような方式でもハッシュチェーンを修正する必要がない。削除又は修正されたレコードのオリジナルハッシュから生成された有効なレコードの全てのハッシュは有効である。システムハッシュは、削除又は修正がアクションであることから、削除又は修正されたレコードの新しいハッシュを含む。このような方式により、詐欺行為は、再算出されたハッシュと一致しない全てのレコードされたハッシュを探し出して容易に認識することができる。

【０２６７】

ゼロ知識証明を有するハッシュチェーン（Ｈａｓｈ　ｃｈａｉｎ　ｗｉｔｈ　ｚｅｒｏ

ｋｎｏｗｌｅｄｇｅ　ｐｒｏｏｆｓ）
ハッシュチェーンは、トランザクションの両側でハッシュらに関するレコードをハッシュしたことを相手に証明できるようにする追加レイヤを提供する。これはレコードのハッシュが該当レコードの実際のハッシュであることを１方の当業者が他方の当業者（検証者）に証明するようにするハッシュチェーン内にキー交換アルゴリズムを含んで行われる。
【０２６８】
二人の当事者が共通のキーを交渉することを可能にする任意のアルゴリズムがここで使用されることができ、ゼロ知識証明を使用する必要がない。しかし、ゼロ知識証明を使用するＰＡＫＥ（ｐａｓｓｗｏｒｄ　ａｕｔｈｅｎｔｉｃａｔｅｄ　ｋｅｙ　ｅｘｃｈａｎｇｅ）アルゴリズムは、ここで使用することが最も効率的である。中間段階で正しいＰＡＫＥプロトコル及びゼロ知識証明を使用することは、当事者が同じ中間ハッシュを生成することになるので、ハッシュを交換する必要がない。
【０２６９】
両側がゼロ知識証明を用いて同じハッシュを生成することを可能にするＰＡＫＥアルゴリズムのようなアルゴリズムを使用すると、当事者はさら進むことができる。トランザクションを構成する情報を含んで使用して「証明」を生成することのできるゼロ知識照明を使用することにより、当事者は両方とも同じ中間ハッシュを生成することができる。これによって、それらの中間ハッシュを互いに交換する必要がない。また、これは、レコード及び情報を生成するステップとそれらのステップから発生する結果がハッシュチェーンプロセスのコンポーネントになることを意味する。２以上の当事者が参加する場合、Ｔｅｒｅｏｎはプロトコル及びゼロ知識証明のグループ変形を使用して、全ての当事者が共用ハッシュを生成できるようにする。
【０２７０】
当事者が同じハッシュが生成されるようにするＰＡＫＥアルゴリズムは、通常、当事者間で中間ハッシュを生成できるようになるまでに２回又は３回の情報伝達を行う。トランザクションが完了するのに２つの段階しが必要としない場合（例えば、要求及び受諾／検証）、当事者は、１つの中間ハッシュのみを生成する。トランザクションが３つの段階を必要とし、アルゴリズムが２つのパスによりハッシュを生成する場合、当事者は３つの段階を２回繰り返して４セットの情報を交換し、２つのハッシュ（トランザクションで最初の２つの段階後に最初のハッシュ、その次に３つの段階を繰り返してから２番目のハッシュ）を生成する。
【０２７１】
このようなゼロ知識証明の例がＳｃｈｎｏｒｒ　ＮＩＺＫ証明である。このゼロ知識証明は、Ｓｃｈｎｏｒｒ　ＮＩＺＫ証明に対する明細書に開示されたように、証明の一部に送信される情報及び証明の一部であるハッシュを生成するために用いられる情報の全てに追加情報を追加することで簡単に拡張できる。
【０２７２】
また、ＳＰＥＫＥ（ｓｉｍｐｌｅ　ｐａｓｓｗｏｒｄ　ｅｘｐｏｎｅｎｔｉａｌ　ｋｅｙ　ｅｘｃｈａｎｇｅ）プロトコルにおける共用キーを生成する方法を採択することのような他の方法も使用でき、その方式は上記のことから明らかである。
【０２７３】
また、当事者がトランザクションデータに基づいて共用キーを生成するようキー交換プロトコルを拡張できるこよも簡単な方法である。言い換えれば、ここでは簡潔さが目的として単純に図示されていない。
【０２７４】
共用キーを生成するために、当事者は共用キーのハッシュを簡単に生成する。ハッシュは、トランザクション情報を有効にできる情報を含むが、該当の情報が共用キー、及びハッシュを生成するプロセッサで使用されたためである。
【０２７５】
２段階のトランザクション（Ｔｒａｎｓａｃｔｉｏｎ　ｉｎ　ｔｗｏ　ｓｔａｇｅ）

この動作方法を示す例は、４つのアカウント５０２、５０４、５０６及び５０８を含むハッシュチェーンの樹枝状特性を示す図５に示されている。アカウントは、同じシステム上に存在してもよく、個別システム上に存在してもよい。アカウントが存在する位置は関係ない。ステップ５１２及び５１４でのトランザクションは２つの段階を必要とする。

【０２７６】

２パスＰＡＫＥ（Ｔｗｏ　ｐａｓｓ　ＰＡＫＥ）

ステップ５１２の最初のパスで、アカウント５０２は、ステップ５１０で生成されたアカウントの前のハッシュｈ５１０を取り、最初の段階のトランザクション情報に加え、最初のゼロ知識証明を構成し、これをアカウント５０４に伝達する。ゼロ知識証明は、最初の段階のトランザクション情報及びハッシュｈを構成する情報を伴う。

【０２７７】

第２パスで、アカウント５０４は、該当アカウントの前のハッシュｈ５０４を取って、これを第２段階のトランザクション情報に加え、第２ゼロ知識証明を構成し、これをアカウント５０２に伝達する。第２ゼロ知識証明は、第２段階のトランザクション情報及びハッシュｈ５０４を構成する情報を伴う。

【０２７８】

アカウント５０２及び５０４は、両方のアカウントの中間ハッシュであるハッシュｈ５１２ｉ～５１４ｉを独立的に構成する。２つのアカウント５０２及び５０４は、このハッシュをそのレコードに追加する。アカウント５０２は、ステップ５１２でトランザクションのレコードのハッシュｈ５１２を生成し、アカウント５０４は、ステップ５１４でトランザクションのレコードのハッシュｈ５１４を生成する。

【０２７９】

３パスＰＡＫＥ（Ｔｈｒｅｅ　ｐａｓｓ　ＰＡＫＥ）

この例で、ステップ５１２及び５１４において、トランザクションは当事者が３つのパスの後で共通ハッシュを構成するようにするＰＡＫＥアルゴリズムを用いて２段階を取る。

【０２８０】

第１パス及び第２パスは上記のように実行される。第３パスで、アカウント５０２は、アカウント５０４が第２パスで送信した情報を取って、該当情報で第３ゼロ知識証明を構成し、これをアカウント５０４に送信する。また、第３ゼロ知識証明は、第２段階のトランザクション情報及びハッシュｈ５０４を構成する情報を伴う。

【０２８１】

アカウント５０２及び５０４は、ハッシュｈ５１２ｉ～５１４ｉを独立的に構成する。２つのアカウント５０２及び５０４は、ハッシュをそれらのレコードに追加する。２パスＰＡＫＥアクセス方式と同様に、アカウント５０２は、ステップ５１２においてトランザクションのレコードのハッシュｈ５１２を生成し、アカウント５０４は、ステップ５１４においてトランザクションのレコードのハッシュｈ５１４を生成する。

【０２８２】

どちらの場合も、チェーンには、アカウント５０２からステップ５１２まで、およびアカウント５０４からステップ５１４までのハッシュのチェーンを検証する情報が含まれている。アカウント５０２と５０４の両方に、それらの記録のために中間ハッシュｈ５１２ｉ、５１４ｉとそのハッシュが含まれる。ただし、ここでの中間ハッシュは、ゼロ知識証明を使用しない前述した例のシステム間で交換された中間ハッシュのものとは微妙に異なる。ここでは、中間ハッシュは、アカウント５０２と５０４の間のトランザクションのハッシュであるため、アカウント５０２と５０４の両方に共有である。ハッシュは、トランザクションのハッシュであり、トランザクションの一部として生成される。トランザクションと同時に発生する。ハッシュｈ５１２は、アカウント５０２のそのトランザクションのレコードのハッシュであり、それに対して私的な情報を含むが、アカウント５０４のハッシュｈ５１４は、トランザクションのそのレコードのそのハッシュである。したがって、アカウント５０２とアカウント５０４は、それらの間のトランザクションにおける実際

のステップと、そのトランザクションのレコードとの両方を証明できる。

【０２８３】

３段階のトランザクション（Ｔｒａｎｓａｃｔｉｏｎ ｉｎ ｔｈｒｅｅ ｓｔａｇｅｓ）

図５を用いて他の例として、ステップ５２８及び５３０において、トランザクションが２つではなく３つの個別段階を含むと仮定する。

【０２８４】

２パスＰＡＫＥ（Ｔｗｏ ｐａｓｓ ＰＡＫＥ）

最初のパスで、アカウント５０２は、ステップ５２２で生成されたこのアカウントの前のハッシュｈ５２２を取り、これを最初の段階のトランザクション情報に追加して第１ゼロ知識証明を構成し、これをアカウント５０６に送信する。ゼロ知識証明は、第１段階のトランザクション情報及びハッシュｈ５２２を構成する情報を伴う。

【０２８５】

第２パスでは、アカウント５０６は、ステップ５２４で生成された該当アカウントの前のハッシュｈ５２４を取り、これを第２段階のトランザクション情報に追加して第２ゼロ知識証明を構成し、これをアカウント５０２に送信する。第２ゼロ知識証明は、第２段階のトランザクション情報及びハッシュｈ５２４を構成する情報を伴う。

【０２８６】

アカウント５０２及び５０６は、共用ハッシュをハッシュｈ５２８ｉ５３０ｉを独立的に構成できるが、ＰＡＫＥアルゴリズムが第２パス後で当事者が共用ハッシュを構成するためである。しかし、トランザクションには実行すべき第３段階がまだある。

【０２８７】

この例では、システムは単に、ＰＡＫＥアルゴリズムを用いて第２パスセットを介して実行する（トランザクションの第３段階で開始す）。第２パスセットの第２パスは、単にランダムデータを使用してもよい。また、これは、２段階トランザクションで３パスＰＡＫＥを使用するのと同様に、最後の段階を繰り返すこともできる。

【０２８８】

後者の場合で、第３パス（新しいＰＡＫＥアルゴリズムの第１パス）は実行され、アカウント５０２が署名したｈ５２８ｉ５３０ｉを取り、これを第３段階のトランザクション情報に追加し、第３ゼロ知識証明を構成し、これをアカウント５０６に送信する。第４パス（新しいＰＡＫＥアルゴリズムの第２パス）は実行され、アカウント５０６が署名したｈ５２８ｉ５３０ｉを取り、これをアカウント５０２が送信した第３段階のトランザクション情報に追加し、その情報を用いて第４ゼロ知識証明を構成し、これをアカウント５０２に送信する。アカウント５０２及び５０６は、ハッシュｈ５２８ｉ２５３０ｉ２を独立的に構成してもよい。これは、このトランザクションで生成された第２共通ハッシュであり、これがトランザクションの３つの段階の全てを含んでいるため、アカウント５０２及び５０６間のトランザクションのハッシュである。アカウント５０２及び５０６の両方が、このハッシュをレコードに追加する。アカウント５０２は、ステップ５２８において、このトランザクションのレコードのハッシュｈ５２８を生成し、アカウント５０６は、ステップ５３０において、このトランザクションのレコードのハッシュｈ５３０を生成する。

【０２８９】

この順は、上述したものと全く同じ方式で各トランザクションのハッシュを生成するために、アカウント５０２、５０４、５０６及び５０８の間の追加トランザクションについて実行される。

【０２９０】

３パスＰＡＫＥ（Ｔｈｒｅｅ ｐａｓｓ ＰＡＫＥ）

第１パス及び第２パスは、上記のように実行される。第３パスでは、アカウント５０２は、第３段階のトランザクション情報を構成する情報を用いて第３ゼロ知識証明を構成し、これをアカウント５０６に送信する。ゼロ知識証明は、第３段階のトランザクション情

報を構成する情報を伴う。

【０２９１】

アカウント５０２及び５０６は、ハッシュｈ５２８ｉ～５３０ｉ独立的に構成する。アカウント５０２及び５０６の両方は、このハッシュをレコードに追加する。アカウント５０２は、ステップ５２８でこのトランザクションのレコードのハッシュｈ５２８を生成し、アカウント５０６は、ステップ５３０でこのトランザクションのレコードのハッシュｈ５３０を生成する。

【０２９２】

図５に関する例示として、システムは、中間ハッシュ又はトランザクションハッシュを生成するためにゼロ知識証明を使用し、ハッシュｈ５３０は、アカウント５０２のハッシュの全てをｈ５２８ｉ、アカウント５０４のハッシュの全てをｈ５２６ｉ、アカウント５０８のハッシュの全てをアカウント５０６がｈ５２４を生成したときに生成されるアカウント５０８の中間又はトランザクションハッシュまで、及びアカウント５０６のハッシュの全てをｈ５３０で検証された情報を含む。しかし、それがトランザクションネットワーク内の全てのハッシュを検証するが、アカウント５０６は、他のアカウント、システム、又は、サーバに入力されたトランザクションに対するトランザクションレコードのみを保有する。たとえ、そのハッシュがアカウント５０２又はアカウント５０４がそれらのトランザクションのハッシュを検証するために使用できる情報を含んでいても、アカウント５０２及び５０４間のトランザクションに対するトランザクションレコードの内容について何も知らない。

【０２９３】

重要なことは、当事者の全てが同じ中間ハッシュを独立的に生成するために使用されるアルゴリズムは、当事者がトランザクションに影響を与えるために交換するステップを使用することにある。したがって、レコードを生成するトランザクションは、ハッシュチェーンプロセスのコンポーネントになり、ハッシュチェーンエントリ（ｈａｓｈ　ｃｈａｉｎ　ｅｎｔｒｙ）を生成するプロセスは、トランザクションに影響を与えるプロセスと同一である。もう１つの見方は、トランザクションがトランザクションの一部としてハッシュを生成し、該当ハッシュ及びこれに伴う情報がトランザクションの監査になるのであろう。それらは１つになって同一である。ブロックチェーンを使用すると、トランザクションの開始者はトランザクションを完了し、後で監査のためにそのレコードをブロックチェーンに送信する。これにより、トランザクションに統合されることなく、他のステップがプロセスに追加される。

【０２９４】

トランザクション自体がハッシュチェーンが提供する監査追跡の同時コンポーネントになると、監査追跡によって詳細がキャプチャー又は検証されないトランザクションを有するということは不可能である。トランザクションの完了後に、ほとんど完了されたトランザクションレコードが監査システムに伝えられる点で、多くの監査追跡は「イベント後（ａｆｔｅｒ　ｔｈｅ　ｅｖｅｎｔ）である。このような場合、監査によって受信されたレコードがトランザクションにより生成されたレコードと同一でない可能性がある。したがって、コンピュータの記録は通常小文字（ｈｅａｒｓａｙ）と見なされる。正しいＰＡＫＥ又は類似のプロトコルを使用してゼロ知識証明を統合することは、監査追跡がトランザクションによって生成され、トランザクション及びそのレコードが監査追跡の一部になることを意味する。これはリアルタイムで報告されるため、これはリアルタイムトランザクションに対して重大な影響を与える。

【０２９５】

ゼロ知識証明を用いてハッシュを構成するプロセスは、ハッシュチェーンでハッシュを生成するいずれのシナリオに適用されてもよい。これは図８によって示されたシステムハッシュ、ライセンスサーバハッシュ、及びオフラインハッシュに使用されてもよい。重要なことは、ハッシュが２つ又はそれ以上のエンティティ間のトランザクションを含むことにより、そのエンティティが当事者、デバイス、又は、システムであるか否かに関係ない

。プロセスは、標準ハッシュの使用も排除されない。したがって、１つのシステムは、デバイスがオンライン又はオフラインであるか否かに関係なく、アカウント間のトランザクションのゼロ知識証明を用いて生成されたハッシュを使用できるが、システムハッシュ及びライセンスハッシュのために標準ハッシュを使用してもよい。第２システムは全てのハッシュに対してゼロ知識証明を使用できるが、第３システムは標準ハッシュのみを使用する。

【０２９６】

多重トランザクション段階を含むマルチパスＰＡＫＥ（Ｍｕｌｔｉｐｌｅ ｐａｓｓ ＰＡＫＥｓ ｗｉｔｈ ｍｕｌｔｉｐｌｅ ｔｒａｎｓａｃｔｉｏｎ ｓｔａｇｅｓ）

前記の例は、２つ又は３つの段階を含むＰＡＫＥで２つ又は３つの段階を含むトランザクションを用いてトランザクションの両側で共用キーを生成できるようにする方法であり、システムは、該当の例に限定されない。実際には、異なる複数のパスが必要なＰＡＫＥを使用する複数の段階を含むトランザクションを支援するシステムに対して同じ方法が効果を有する点である。しかし、システムは、トランザクション全ての段階をカバーするために必要な多くのＰＡＫＥ実行を簡単に使用する。これは、最終的な共通キーを生成するために要求されるＰＡＫＥパスを生成するために最終段階を何度も繰り返してトランザクションハッシュを生成する。

【０２９７】

ゼロ知識証明を有するシステムハッシュチェーン（Ｓｙｓｔｅｍ ｈａｓｈ ｃｈａｉｎ ｗｉｔｈ ｚｅｒｏ ｋｎｏｗｌｅｄｇｅ ｐｒｏｏｆｓ）

図６に戻って、ゼロ知識証明及び古典的なハッシュを用いて生成されたハッシュの全てを使用できるハッシュチェーンが示されている。図面には、システムハッシュｈ６０６、ｈ６０８、ｈ６１２、などと共に、同じシステム６０６上の２つのアカウント６０２及び６０４を示す。システムは、レコードが存在する位置に関係なく、レコードを発生させる全てのアクションに対するレコードの新しいハッシュを生成する。アカウント間のトランザクションは、上述したように、アカウントそれぞれに対する中間ハッシュ又はトランザクションハッシュを生成するためにゼロ知識証明を使用してもよい。システムハッシュは、該当レコードを生成するとき各レコードのシステムハッシュを含む。

【０２９８】

ステップ６１４及び６１６において、アカウント６０２及び６０４間のトランザクションが３パス後も当事者が共通ハッシュを構成することを可能にするＰＡＫＥアルゴリズムを用いて３つの個別段階を含むと仮定する。

【０２９９】

トランザクションの第１ステップで、アカウント６０２は、以前のレコードのハッシュであるハッシュｈ６１０を、ステップ６０８で生成されたシステムハッシュｈ６０８のシステムアカウント６０６と交換する。それは、このシステムハッシュ及びこのハッシュｈ６１０をステップ６１０で生成された第１段階のトランザクション情報に追加し、第１ゼロ知識証明を構成し、これをアカウント６０４に送信する。ゼロ知識証明は、第１段階のトランザクション情報、ハッシュｈ６１０、及びハッシュｈ６０８を構成する情報を伴う。

【０３００】

トランザクションの第２ステップで、アカウント６０４は、ステップ６０８で生成されたシステムハッシュｈ６０８に対するシステムアカウントとハッシュｈ６０４を交換する。それは、このシステムハッシュ及び第１段階のトランザクション情報に対するこの以前のレコードのハッシュであるハッシュｈ６０４第１段階のトランザクション情報に追加し、第２ゼロ知識証明を構成し、これを６０２に送信する。ゼロ知識証明は、第２段階のトランザクション情報、ハッシュｈ６０４、及びハッシュｈ６０８を構成する情報を伴う。

【０３０１】

トランザクションの第３ステップで、システムアカウント６０６は、ｈ６１０及びｈ６０４をレコードに追加し、中間システムハッシュｈ６１２ｉを生成する。

第４ステップで、アカウント６０２は、第３段階のトランザクション情報を構成する情報を用いて第３ゼロ知識証明を構成し、これをアカウント６０４に送信する。第３ゼロ知識証明は、第３段階のトランザクション情報を構成する情報を伴う。

【０３０２】

第５ステップで、アカウント６０２及び６０４は、ハッシュｈ６１４ｉ６１６ｉを独立的に構成する。アカウント６０２及び６０４両方はこのハッシュをそれらのレコードに追加する。ハッシュｈ６１４ｉ６１６ｉは、トランザクションのハッシュである。

【０３０３】

第６ステップで、アカウント６０２は、システムアカウント６０６とｈ６１４ｉ６１６ｉをｈ６１２ｉに交換し、ｈ６１２ｉをそのレコードに追加し、ステップ６１３で、トランザクションのレコードのハッシュｈ６１４を生成する。アカウント６０４は、システムアカウント６０６とｈ６１４ｉ６１６ｉをｈ６１２ｉに交換し、ｈ６１２ｉをそのレコードに追加し、ステップ６１６で、トランザクションのレコードのハッシュｈ６１６を生成し、システムアカウント６０６は、ｈ６１４ｉ６１６ｉの２つのコピーをレコードに追加して、ステップ６１２で新しいシステムハッシュｈ６１２を生成する。

【０３０４】

ステップ６１４で、トランザクションに対するアカウント６０２のレコードは、ハッシュｈ６１０、ハッシュｈ６０４、システムハッシュｈ６０８、トランザクションハッシュｈ６１４ｉ６１６ｉ、中間システムハッシュｈ６１２ｉ、３段階のトランザクション情報、トランザクションのレコード、アカウントＩＤ、及びハッシュｈ６１４を含む。

【０３０５】

ステップ６１６で、トランザクションに対するアカウント６０４のレコードは、ハッシュｈ６１０、ハッシュｈ６０４、システムハッシュｈ６０８、トランザクションハッシュｈ６１４ｉ６１６ｉ、中間システムハッシュｈ６１２ｉ、３つの段階のトランザクション情報、これのトランザクションのレコード、アカウントＩＤ、及びハッシュｈ６１６を含む。

【０３０６】

（アカウント６０２のトランザクションのレコードは、それぞれ異なる状態でトランザクションを開始及び終了したため、アカウント６０４のレコードとは異なり、それぞれ異なるアカウントの詳細及びＩＤを有する異なるアカウントである。）

システムハッシュｈ６１２は、個々のトランザクションだけではなく、全体トランザクションといった両側のハッシュを含むため、ハッシュチェーンを相当強化する。

【０３０７】

Ｔｅｒｅｏｎが異なるシステム上のアカウント間のトランザクションを管理する場合、プロセスは若干ことなる。ここでは、各システムが管理するアカウントとシステムハッシュ及び中間システムハッシュを交換する。そうでなければ、図６も関連して上述した方法は、アカウント６０２及び６０４及びシステム６０６を有する代わりに、関連するアカウント６０２に関するシステム６０６、及びアカウント６０４に関する第２システム６０５を示すこと以外は同一である。ステップ６１４及び６１６で発生したトランザクションと共に、発生するシステムハッシュはステップ６１２で行われるトランザクションでは、結果として生じるハッシュシステムは、ステップ６１２でのシステムトランザクションと、アカウント６０４に対応する第２システム６０５上の同等のトランザクションと表す。同時にトランザクションできるアカウントでは、システムはレコードを生成する各対話に対してハッシュを生成する。

【０３０８】

図６は、順次ハッシュ及び中間ハッシュを示すが、現実は異なる。図６ａは３つのアカウント６０２ａ、６０４ａ、及び６０６ａ図示し、全てシステムアカウント６０８ａと共に外部サーバ上のアカウントと相互作用している。トランザクションの段階はトランザクションがシステム上で同時に行われるとき発生の可能性を説明するためにインターリビングする。便宜のために、これらは全て同じサーバ上に示されている。

【０３０９】
前記の例で、ステップ６１２ａで、アカウント６０２ａは、ｈ６１２ａを取得するためにシステム６０８ａとハッシュｈ６０２ａを交換するだろう。システム６０８ａは、前記例が中間ハッシュｈ６１６ａｉとして示すものを生成する。この添字「ｉ」は、各トランザクションが３つのシステムハッシュ、トランザクション前のオリジナルハッシュ、トランザクションの特定段階でのシステムハッシュ（中間ハッシュ）、及びトランザクションの終わりでシステムハッシュを含むことを明確にするために用いられる。添字「ｉ」は、中間ハッシュを示す。前記の推理により、最終システムハッシュはｈ６１６ａであり得る。複数の同時に発生する又はインターリーブされたトランザクションがある場合、ラベリングによりこれ以上の進行状況が分からない。代わりに、各システムハッシュは、トランザクションのうち又はその後で生成の有無に関係なく、以前のハッシュに対する増分（ｉｎｃｒｅｍｅｎｔ）ではあるが、システムハッシュである。アカウント６０２ａが開始し、次にアカウント６０４ａが開始し、アカウント６０６ａが開始し、アカウント６０２ａが終了し、アカウント６０４ａが終了する前にアカウント６０６ａが終了するために３つのトランザクションが発生する場合、他のトランザクション又はアクションがサーバ上のこれ（又はアカウント）又は任意の他のアカウントで発生していない場合、ハッシュの順は次のようになり、結果的にダイヤグラムは以前の図面と微妙に異なる。

【０３１０】
アカウント６０２ａは、ｈ６１２ａを取得するためにシステムとハッシュｈ６１０ａを交換する。システムは、該当ハッシュｈ６１０ａを使用して、次のシステムハッシュｈ６１６ａを生成する（これは、Ｈ６２８ａｉがそのトランザクションの最終的なシステムハッシュであるため、アカウント６０２ａのトランザクションが完了すると、ｈ６２８ａｉは元々のラベルされているのである）。

【０３１１】
アカウント６０４ａは、ｈ６１６ａを取得するためにシステムとハッシュｈ６１４ａを交換する。システムは、次のシステムハッシュｈ６２０ａを生成するために該当ハッシュｈ６１４ａを使用する。

【０３１２】
アカウント６０６ａは、ｈ６２０ａを取得するためにシステムとハッシュｈ７１８ａを交換する。システムは、次のシステムハッシュｈ６２４ａを生成するために該当ハッシュｈ６１８ａを使用する。

【０３１３】
アカウント６０２ａがその中間又はトランザクションハッシュを生成すると、そのハッシュｈ６２２ａをシステムハッシュｈ６２４ａと交換する。システムは、次のシステムハッシュｈ６２８ａを生成するために該当ハッシュｈ６２２ａを使用する。

【０３１４】
アカウント６０６ａがその中間又はトランザクションハッシュを生成すると、そのハッシュｈ６２６ａをシステムハッシュｈ６２８ａと交換する。システムは、次のシステムハッシュｈ６３２ａを生成するために該当ハッシュｈ６２６ａを使用する。

【０３１５】
アカウント６０４ａがその中間又はトランザクションハッシュを生成すると、そのハッシュｈ６３０ａをシステムハッシュｈ６３２ａと交換する。システムは、次のシステムハッシュｈ６３６ａ（図示せず）を生成するために該当ハッシュｈ６３０ａを使用する。

【０３１６】
ハッシュチェーンは、システムがトランザクションを処理し、該当トランザクションを監査し、同時に、該当トランザクションによって送信されたり生成されたデータを認証するようにする。このようなステップは同時に発生する。デバイスがトランザクションを監査システムで正直に報告すると仮定する必要がない。トランザクションは監査を生成し、監査はトランザクションを生成する。

【０３１７】

これにより、プログラムされたデバイスによって実行されるトランザクションの特性を全て変更する。ＩｏＴデバイスを含む任意のプログラムされたデバイスは、トランザクションと、その監査及び認証が同時に行われるため、他のデバイスとの間のトランザクション及びデータを有効にして信頼し得る。

【０３１８】

該当トランザクション及び監査が同じプロセスの一部として生成されるため、デバイスがトランザクションの正確なレコードを監査システムに送信すると仮定する必要がない。この同時発生する特性は、監査追跡の証拠値（ｅｖｉｄｅｎｔｉａｌ　ｖａｌｕｅ）の品質を変更する。各デバイスは他のデバイスによって送信される情報に依存できるが、他のデバイスの信頼性に関して仮定することはない。送信及び受信されるデータは、処理されるデータ及び認証及び監査されるデータである。

【０３１９】

ルックアップサービスと組み合わせるとき、以前に相互作用していないデバイスは互いに認証し、それぞれが行うサービス又は機能を決定し、その次に相互間に通信し、その通信に依存して任意の人が介入する必要なくプログラムされた通りに作業を行うことができる。

【０３２０】

ハッシュチェーンは、ＩｏＴデバイスを含むプログラムされたデバイスがオンライン及びオフラインの全てに動作できる。デバイスがオフラインのとき、タイムスタンプ、該当デバイスのクロックスクリューに関する情報、デバイス固有のトランザクションＩＤ（内部モノトニックカウンタなどによって生成されたもの）、及びトランザクション情報にその他同期化情報を含む場合、それらのサーバがデバイス又は第３パーティーサーバからオフライントランザクションのレコードを最終的に受信するとき、それらのサーバが各トランザクションに対する因果関係を格納する正確なタイムラインを再構成するようにする。ハッシュチェーンは、オンライン及びオフラインモード両方で、サーバがトランザクションレコードの内容に依存することを可能にする。

【０３２１】

デバイス間の通信を保護する通信セキュリティーモデルと組み合わせると、デバイス及びサーバは、中間者攻撃に影響を受けない方式で通信し得る。Ｔｅｒｅｏｎは、ＩｏＴ及び他のプログラムされたデバイスが安全に通信し、該当デバイス間の送信されたデータに依存するようにする。

【０３２２】

その１つの例として、産業用センサ及び制御装置のセットで動作するＩｏＴ及び他のプログラムされたデバイスのネットワークであり得る。セキュリティーモデルは、ルックアップディレクトリサービスを使用し、このようなデバイス間で安全に通信するようにし、オリジナルコレクションに追加されるとき該当デバイスが新しいデバイスと相互作用するようにする。Ｔｅｒｅｏｎは、新しいデバイスを認識し、新しいデバイスを信頼できるようにするために再構成する必要がない。ハッシュチェーンは、デバイスがそれらの間の通信コンテンツ及びタイミングを信頼できるようにし、送信されたデータの真実性に対する人の評価を必要とせずに運営者が生成及び送信されたデータに依存可能にする。第３者が、データとインタフェースすることができない。その監査及び認証チェーンがその送信と同時に発生する。

【０３２３】

ルックアップサービスは、セキュリティーモデル及びルックアップサービスと結合されるとき人の介入を必要とせず、デバイスが信頼及び認証できるアドホック相互接続を生成できるようにする。デバイスが認証され、これの詳細がルックアップサービスに追加されれば、必要に応じて、他のデバイスは該当デバイスに接続できる。該当デバイスが任意の方式により損傷される場合、これに対する全てのアクセスは、同一のルックアップサービスによって不活性化される。

【０３２４】

システムは、ハッシュチェーン及びルックアップサービスから発生する追加利点を提供する。全てのデバイスが個別的に認証され、監査されるため、システムは必要に応じて、特定のデバイスがそれらのデバイスのソフトウェアに対するアップデートをダウンロードするよう指示し、デバイスは安全で、信頼されるソースのみを行う。ルックアップサービスは、特定のデバイスが提供及び使用する、例えば、サービス、インタフェース、及びデータフォーマットを詳細に説明する。したがって、デバイスが特定のデバイス（ｓｕｒｖｉｖｅ）にアクセスするため、他のデバイスに接続を試みる場合、要求されるインタフェース又はフォーマットを支援するために必要なソフトウェアがないとき、接続中であるデバイスのいずれか１つ、又は必要に応じて、デバイスの両方のデバイスが互いに通信できるようにする必要なソフトウェア又は構成をダウンロードするためにシステムサーバと通信し得る。デバイス間通信が完了した後、デバイスがソフトウェアを保持するか否かは、デバイス又はデバイスが行うサービス、及び該当デバイスの容量により決定される。ハッシュチェーンは、たとえ、それらが該当ソフトウェアを除去したとしても（それがダッシュ通信するとき、それを再びインストールしてもよい）、２つのデバイスは必要に応じて、後ほど他のデバイス又はサーバにアップロードできるデバイス間通信の完全な監査及びレコードを保持することを意味する。この機能は、完全に自動化されたＩｏＴデバイスから支払いデバイスのようなプログラムされた他のデバイスに至るまで、全タイプのデバイスに拡張される。

【０３２５】

ハッシュチェーンの分散レコード（Ｄｉｓｔｒｉｂｕｔｅｄ　ｒｅｃｏｒｄｓ　ｏｆ　ｔｈｅ　ｈａｓｈ　ｃｈａｉｎ）

全体のハッシュチェーンの分散複製を提供するために、Ｔｅｒｅｏｎシステムは、該当サーバの現在の接続と最後の接続間に発生した全てのトランザクションに対してハッシュチェーンをライセンスサーバ、ルックアップサーバ、又は、他のサーバセットのような中央サーバ集合にアップロードする。同じＴｅｒｅｏｎシステムが異なるＴｅｒｅｏｎシステムに対応するハッシュチェーンをダウンロードし得る。これにより、全てのＴｅｒｅｏｎシステムの全てのトランザクションに対してハッシュチェーンの分散元帳が提供するが、トランザクションごとに各ハッシュチェーンを再び算出する必要がない。しかし、Ｔｅｒｅｏｎシステムに追加のストレージ負担がかかる。中央サーバは、ライセンス及び検索サーバのようなグローバルサーバにする産業、地域又はその他の制約条件に限定され得る。ハッシュチェーンのコピーの到達範囲を制限することで、この変形の算出及び格納上の負担を減らし得る。

【０３２６】

中央サーバの範囲を制限する代わりに、他のシステムでアップロードしたハッシュチェーンをダウンロードできるシステムを制限することができる。したがって、ある銀行のハッシュチェーンは他の銀行によってダウンロードされることができるが、該当銀行がアップロード銀行と同じ地域にあるか、又は他の銀行と取引したか否かに応じて制約を受ける。同様に、病院システムは、同じ地域の病院によってアップロードされたハッシュチェーンのみをダウンロードできる。柔軟性には制約されない。

【０３２７】

Ｔｅｒｅｏｎで用いられるハッシュチェーンには極めて有用な属性を有する。それはローカル元帳のを提供するが、分散認証を提供する。トランザクション情報をトランザクションに関するユーザ及びサービスに非公開として保持するが、ハッシュによって提供された認証を全てのサーバ、サービス及びデバイスに分散する。ゼロ知識証明で生成されたハッシュはこれを示す。特定トランザクションに関するシステムのみがトランザクション情報を保有する。しかし、システムと相互作用する全てのシステムとデバイスは、該当システムの初期ハッシュに関する情報を含むハッシュを生成する。

【０３２８】

分散認証は、変調されたレコード（ｔａｍｐｅｒｅｄ　ｒｅｃｏｒｄ）を隠そうとする潜在的な詐欺師に対して算出不可能な障壁を提供するため重要である。

ブロックチェーンを使用すると、詐欺師は、変調されたレコードを隠してブロックチェーンを変更し、誤ったレコードを有効なレコードとして記録するために２５～３３％のサーバだけを制御すれば良い。一回行われると、プロセスを元に戻すことは不可能である。

【０３２９】

Ｔｅｒｅｏｎハッシュチェーンを使えば、詐欺師は全てのＴｅｒｅｏｎサーバ、全てのＴｅｒｅｏｎサービス及び全てのＴｅｒｅｏｎデバイスを制御して該当サーバ及びデバイス皆でチェーンの全てのハッシュを再び算出しなければならない。これは算出上で実行不可能である。

【０３３０】

ハッシュチェーンは、ブロックチェーンの提案者が後者に対して予測するのと少なくとも同じレベルの経済的節減及び経済的効率性を提供する。差異点は、Ｔｅｒｅｏｎハッシュチェーンが実際にそうすることができることである。ブロックチェーンは、その設計とその設計に固有の限界のために、そうすることができない。

【０３３１】

このシステムの長所は、詐欺師が全てのハッシュ及び該当レコードと接続されたハッシュを再び算出しないと、データベースでレコードを削除したり修正できないことである。Ｔｅｒｅｏｎがシステムハッシュやライセンスサーバへの接続なしで単一のサーバ上で作動する場合、これは理論的に可能であるが、リンクされたチェーンのうちの１つが他のサーバ又はデバイスのパーティーとのトランザクションを含む場合、詐欺師は、他のサーバ又はデバイスの全てのハッシュを再び算出する必要がある。そうすることの困難さは、オリジナルレコードの日時の後にハッシュチェーンと相互作用する追加のサーバ又はデバイスごとに急激に増加する。

【０３３２】

ハッシュチェーンにより、組織は全てのデバイスによって収集、生成、又は管理されるデータの正確を保障し、レコードのオリジナルコンテンツと無欠性を保障して、以前のレコードを基盤としたトランザクションのコンテンツと無欠性を保障できるようにする。これは、支払いデバイスから医療デバイス、交通センサ、気象センサ、水流検出器などに至る、あらゆる全てのデバイス又はトランザクションに適用される。

【０３３３】

各地域の元帳は個々の組織の責任であるため、これは明確なガバナンス上の利点を有するが、それらは強度を共有しながら明確な責任と説明の責任を提供する方式により、他の組織の元帳のから学び、頼ることができる。ハッシュチェーンは、情報及びトランザクション管理を施行して支援する技術ツールを作成する。

【０３３４】

また、ハッシュチェーンが支払いシステムの構成要素として使用されるとき、Ｔｅｒｅｏｎは、支払い金額を処理し、アーキテクチャーは支払いが今日行われている方式と整合し、Ｂｉｔｃｏｉｎのような暗号化通貨と同等又はそれ以上の利点を提供する。それは、確立された支払いサービスの提供者と中央銀行に「Ｂｉｔｃｏｉｎ　ｂｅａｔｅｒ」を提供する。

【０３３５】

ハッシュチェーンは、極めて安定した迅速な認証を可能にするためＴｅｒｅｏｎシステムで特に魅力的な部分である。

Ｔｅｒｅｏｎのユニークな機能の１つは、包括的なリアルタイムログ及び監査証跡を作成する機能である。Ｔｅｒｅｏｎトランザクションレコードには、トランザクションに必要なすべてのキーストローク（実際の認証資格情報（ＰＩＮやパスワードなど）は除く）であるが、そのトランザクションに関するすべてのデータおよびメタデータと共に、規制および業務上の要件を満たすために求められる。必要条件の複数のサービスプロバイダにまたがって格納されている場合、そのレコードを改ざんされやすいものにし、問題となっているトランザクション以降の一連のトランザクションを改ざんされないようにすることが重要である。

【０３３６】
ブロックチェーンはこれができない。そのレコードが生成されてから権限が付与される前にトランザクションレコードのみ承認できる。ブロックチェーンは、様々なレコードに接続され、ブロックを生成した後これをブロックチェーンに追加する。それは、ブロックチェーンが以前の全てのトランザクションに関する情報が含まれたブロックを含む事実に依存する。ブロックチェーンが追加ブロックを追加するにつれて、このようなブロックの存在の有無に応じてブロックチェーン内のレコードと全ての以前のレコードの有効性を検査する。これによって、ファイルの大きさが増大するにつれてスケーリングの問題が発生し、もし不一致が発生すると、ブランチ全体が認証を失う。
【０３３７】
ブロックチェーン又はその派生物を使用するのではなく、Ｔｅｒｅｏｎのハッシュチェーンは後続トランザクションの認証を損傷させることなく、調査のために疑わしいレコードを隔離するハッシュ戦略を使用する。それは静的レコードでもリアルタイムトランザクションでも関係なく、あらゆるレコードタイプに合わせて設計されているため、スケーリングの問題を回避できる。
【０３３８】
中間ハッシュを含むハッシュは、管理者がハッシュチェーンを迅速に探索してハッシュ及び該当レコードを確認し、その確認に必要な情報を提供できる。レコード自体も同じである。
【０３３９】
トランザクション又はアクションが発生すると、以前のハッシュが調整されるため、ユーザとシステムが新しいトランザクションの出力を信頼する可能性があることを意味する。したがって、Ｔｅｒｅｏｎは、トランザクションを行う前に各アカウントの累積合計（ｒｕｎｎｉｎｇ　ｔｏｔａｌｓ）を信頼し得る。ハッシュチェーンの有効性は累積合計が正しいかを確認する。
【０３４０】
ハッシュチェーンをブロックチェーン及びその派生物から分離する改正されたレコード、削除されたレコード、又は変調されたレコードの効果を隔離することがこの機能である。定義上、ブロックチェーンが確実に隠されている修正又は変調されたレコードは、該当ブロックチェーンの全体再算出に影響を及ぼす。全てのブロックチェーンを修正しなければならないため、全体ブロックチェーンコミュニティの民主的な決定以外に、偽造されたレコード又は偽りレコードを検索して修正する方法がない。セキュリティー研究者がブロックチェーン設計の主な欠陥として確認したのはこの機能である。その設計は変更できない。
【０３４１】
ハッシュチェーンでは、攻撃者が後続のハッシュを全て再計算できない限り、改ざんされたレコードがハッシュチェーンの残りの部分に影響を与えることができない。改ざん前のハッシュは有効であり、有効なままであるため、それらのハッシュに基づくトランザクション及びそれらのハッシュに関する値は有効なままである。
【０３４２】
オフライントランザクションに対する樹枝状ハッシュチェーンは、オフラインデバイスがサーバに再び接続できる前に該当デバイスが失われたり侵害されても、オフラインデバイスによって実行されたオフライントランザクションを登録する可能性があることを意味する。
【０３４３】
ハッシュチェーンは、ブロックチェーン及びその派生物だけでは達成できないオフライントランザクションの有効性を完ぺきに支援する。ブロックチェーンのコピーを運営するノードは、ブロックを確認するためにオンライン状態でなければならない。ビットコインウォレットは、オフラインでトランザクションを生成できるが、オンライン状態になって該当トランザクションのレコードをノードにプッシュするまで該当トランザクションの有

効性を検査することができない。ノードのうの１つがブロックチェーンで次のブロックを生成する競争で勝ってブロックにレコードを追加するまでトランザクションの有効性が検査されない。

【０３４４】

ディレクトリサービス（Ｄｉｒｅｃｔｏｒｙ　Ｓｅｒｖｉｃｅ）

輸送システム、ＥＭＶ（Ｅｕｒｏｐａｙ、ＭａｓｔｅｒＣａｒｄ、Ｖｉｓａ）のような支払いネットワーク、及びその他のレガシーシステムのような既存システムは、ハブ・アンドスポーク・アーキテクチャー（ｈｕｂ　ａｎｄ　ｓｐｏｋｅ　ａｒｃｈｉｔｅｃｔｕｒｅ）を使用する。ここで、全てのトランザクションは、障害又は脆弱性の潜在的な単一地点を示して拡張のために高いコストの中央ユーティリティを通過する。

【０３４５】

Ｔｅｒｅｏｎシステムはピアツーピアで、あるサーバが他のサーバーと直接通信するため、ハッシュチェーンの検証はピアツーピアネットワークの全ての要素で行われるため、セキュリティ上のハッシュチェーンが極めて重要である。

【０３４６】

説明したように、Ｔｅｒｅｏｎシステムは、システムのクリデンシャル及び情報のディレクトリであるディレクトリサービス２１６を有する。ディレクトリサービス２１６は、特定のユーザに関する複数のタイプのクリデンシャルを格納するため、ユーザ又はデバイス２１８がどのサーバに登録されているか、又はあるサーバが特定のサービス又は機能を提供するかを識別し、ユーザ２１８の複数認証方法が発生できるようにする。例えば、ユーザ２１８は、自身のモバイル番号、メールアドレス、地理的位置、ＰＡＮ（プライマリアカウント番号）などを用いて認証され、毎回認証する必要がないよう全てのものをキャッシュする。

【０３４７】

ディレクトリサービス２１６は、基本となるサービス、サーバ及び実際のユーザアカウントからユーザの認証ＩＤを分離する抽象化レイヤを提供する。これはユーザ２１８又はマーチャントサービスにアクセスするために使用できるクリデンシャルとＴｅｒｅｏｎがサービス自体を行うために必要な情報間に抽象化を提供する。例えば、支払いサービスとして、ディレクトリサービス２１６は、モバイル番号のような認証ＩＤ及び恐らく通貨コードをサーバアドレスとリンクさせるのであろう。ユーザ２１８が銀行アカウントを有しているか否か又はユーザ２１８がどの銀行を決定するの方法は全くない。

【０３４８】

ディレクトリサービス２１６は、サービス提供者が相互を見ることができず、ユーザデータのセキュリティーが提供されるよう様々なサービス間の仲介者の役割を果たす。各サービスは、当該サービスに特定のフィールド（変数）及び値を定義する。しかし、各サービスは、サービスを識別する特定のフィールドと値を含む。

【０３４９】

トランザクションが知られていない当事者との間でトランザクションが完了すれば、ユーザ２１８に関するＴｅｒｅｏｎサーバは、ディレクトリサービス２１６にＵＲＮ（ｕｎｉｆｏｒｍ　ｒｅｓｏｕｒｃｅ　ｎａｍｅ）を送信し、ディレクトリサービス２１６は、ユーザ２１８によって要求されたサービスに対する支払いサービス提供者のＴｅｒｅｏｎサーバに対するＩＰアドレスを返送する。これは、トランザクションがピアツーピアに基づいてユーザ２１８とサービス提供者との間で直接完了することを可能にする。また、Ｔｅｒｅｏｎサーバは、後続の全てのトランザクションがディレクトリサービス２１６を使用する必要がないように、キャッシュにＩＰアドレスを保持する。

【０３５０】

このような抽象化は、ユーザ及びサービス詳細にセキュリティー及び個人情報を提供し、一般ユーザ・クリデンシャル（ｐｕｂｌｉｃ　ｕｓｅｒ　クリデンシャル）に影響を与えることなく、基本となるサービスを追加及び修正できる柔軟性を提供し、必要に応じて、それぞれ異なるものと隔離されている様々なサービスを分割して支援できる機能を提供

する。データサービスのどのフィールドもトランザクションを開始するために必要なデータを含まず、ユーザの認証ＩＤ以外のユーザデータはディレクトリサービス２１６に格納されない。

【０３５１】

しかし、Ｔｅｒｅｏｎディレクトリサービス２１６は、これ以上のものである。複数のクリデンシャルを支援する。したがって、ユーザ２１８は、任意数のクリデンシャルを支払いＩＤとして使用してもよい。例として、モバイル番号、ＰＡＮ、電子メールアドレスなどを含む。クリデンシャルが一意である限り、Ｔｅｒｅｏｎは支援される。

【０３５２】

ディレクトリサービス２１６は、複数のサービスを支援する。これは多面的クリデンシャル（又は「サイキックペーパー（ｐｓｙｃｈｉｃ　ｐａｐｅｒ）」の概念が生まれた所である。サービス提供者がディレクトリサービス２１６上のクリデンシャルをチェックすると、クリデンシャルが自身のサービスに対して登録されているかどうか、そのクリデンシャルをサービスするＴｅｒｅｏｎサーバのみを見ることができる。サービス提供者は、ユーザ２１８が資格を取得したり登録できる異なるサービスの詳細を見ることができない。

【０３５３】

例えば、モバイル又はカードは、図書館の図書館カードクリデンシャル、バス又は汽車の交通機関のチケット、部屋又は施設にアクセスするためのセキュリティーキー、会社の食堂の社内支払いデバイス、劇場チケット、及びスーパーマーケットの標準的な支払いデバイスとなる。それは、運転免許証、健康管理カード又はＩＤカードとなり、サービスへの資格を証明することができる。サービスが必要に応じて、マーチャントのデバイス上で写真ＩＤを表示されることがある。デバイスが作成できるクリデンシャルタイプに対する制限はほとんどない。

【０３５４】

カードの本来の外観を偽装することは難しいが（カードがＯＬＥＤカバー又はカラー電子ペーパーカバーが組み込まれている場合に可能、例えば、サービスがカードに特定クリデンシャルやサービスに必要な情報を表示するよう指示できる。）、フォンアプリケーションの外観は、クリデンシャル及びサービスの性質を反映するようにＴｅｒｅｏｎによって変更される。

【０３５５】

逆ルックアップ機能は、各サーバに対して実現され得る。この機能は、それと通信するサーバが認可されて認証されているか否かを確認できる。Ｔｅｒｅｏｎデバイス（カード、端末、モバイル又はサーバ）間の全ての通信が署名されなければならないため、この機能は必要ではない。しかし、運営者が逆ルックアップを介して追加セキュリティーを必要としたり所望する状況が存在し得る。ここで、ディレクトリサービス２１６は、サービス、Ｔｅｒｅｏｎサーバドメインアドレス Ｔｅｒｅｏｎサーバ番号、Ｔｅｒｅｏｎサーバ運営者、ＴＴＬ（Ｔｉｍｅ　Ｔｏ　Ｌｉｖｅ）、端末認証ＩＤなどのような複数のフィールドを含む。ここで、サービスタグは、トランザクションサービスではないサーバ逆ルックアップを示す。

【０３５６】

図９は、２つのサーバ、すなわちサーバ２０２ａ及びサーバ２０２ｂを有する例を示している。ユーザ２１８はサーバ２０２ｂに登録され、サーバ２０２ａに接続された端末を介してサービスにアクセスする。

【０３５７】

ステップ９０２で、ユーザ２１８は、自分の装置を使って自身を端末に対して自身を識別し、それは自動的に自身を端末に対して自身を識別する。スマートデバイスを用いる場合、端末はそのＩＤをユーザのデバイスにも渡す。ユーザ２１８がカードを使用する場合、そのデバイスがマイクロプロセッサ・カードである場合には、デバイスはその識別をユーザの装置に渡すだけでよい。この場合、カードは、それが登録されているサーバ２０２

ｂと通信する。端末を通した暗号化されたトンネルを介してデバイスのＩＤをサーバ２０２ｂに渡す。

【０３５８】

ステップ９０４で、サーバ２０２ａは、ユーザのデバイスによって提供されるＩＤを受け取り、それが保持するリストに対してＩＤをチェックする。それはＩＤを保有しないため、以前にユーザ２１８を扱っていない。サーバ２０２ａは、ディレクトリサービス２１６に接続する。ディレクトリサービス２１６は、サーバ２０２ａの通信上の署名を検査し、それが有効なことと見なす。ディレクトリサービス２１６は、要求されたサービス（サーバ２０２ａの署名がサーバがそのサービスに対する要求を行う権限があることを確認する）に対するサービスタグに対してＩＤを検索し、ライブ情報に対するキャッシュタイムと共にサーバ２０２ｃを識別する情報で応答する。

【０３５９】

ステップ９０６で、サーバ２０２ａは、ユーザのデバイスがサーバ２０２ｂに登録されているかを確認するためにサーバ２０２ｂに接続する。サーバ２０２ａは、端末のＩＤをサーバ２０２ｂに伝達する。

【０３６０】

ステップ９０８で、サーバ２０２ｂはそれがまだ行われていない場合、端末が登録されているサーバを検索するようディレクトリサービス２１６に同様の要求を行う。また、それはサーバ２０２ａへ端末が要求されたサービスに登録されたかを確認できる。ディレクトリサービス２１６は、ライブ情報へのキャッシュタイムと共にサーバ２０２ａを識別する情報で応答する。

【０３６１】

ステップ９１０で、サーバ２０２ａとサーバ２０２ｂは、必要なトランザクションを行うために互いに直接通信する。これは支払いをすることからドアが開けることに至るまで多様である。

【０３６２】

Ｔｅｒｅｏｎサーバそのものは、トランザクションを開始するために必要な情報を含み、ライセンスが付与されて認証された他のサーバ又はデバイスとのみ通信する。

まず、サーバがディレクトリサービス２１６及び互いに通信すると、それらはデータ自身のミニディレクトリサービスで期限きれするまでデータをキャッシュする。

【０３６３】

この場合、Ｔｅｒｅｏｎサーバ２０２ａとＴｅｒｅｏｎサーバ２０２ｂ間の接続を確立するための通信は明らかに簡単である。これは図１０に示されている。

ステップ１００２で、ユーザ２１８は、自分のデバイスを用いてサーバ２０２ａに接続された端末に対して自身を識別し、それは自動にそれ自身をデバイスに対して識別する。スマートデバイスを使用している場合、端末はそのＩＤをユーザのデバイスにも伝達する。

【０３６４】

ステップ１００４で、サーバ２０２ａは、ユーザのデバイスによって提供されるＩＤを受け取り、それが保持するリストに対してＩＤをチェックする。それが保有しているデータは有効であるため、サーバ２０２ａはサーバ２０２ｂに接続し、デバイスが要求されたサービスについてデバイスが登録されているかを確認する。また、サーバ２０２ａは、端末のＩＤをサーバ２０２ｂに伝達する。サーバ２０２ｂは、デバイスが登録されていることを確認する。

【０３６５】

サーバ２０２ａのキャッシュは、端末のＩＤに対する有効なデータを含んでいるため、それは端末がそれに登録されているかを確認するためにサーバ２０２ｂへ接続する。サーバ２０２ｂはこれを確認する。

【０３６６】

ステップ１００６で、サーバ２０２ａ及びサーバ２０２ｂは、必要なトランザクション

を行うために互いに直接通信する。
キャッシュされたデータがサーバで期限切れになった場合、該当サーバは、以前のようにディレクトリサービス２１６へ接続する。ユーザ２１８が他のサーバに移動した場合、通信は僅かに異なる。この場合について図１１に示されている。差異点は、現在キャッシュされていない情報に基づいた、サーバ２０２ｂとの１番目の通信は、サーバ２０２ａがディレクトリサービス２１６で新しいデータを検索させることである。
【０３６７】
ステップ１１０２で、ユーザ２１８は、自分のデバイスを用いてサーバ２０２ａに接続された端末に対して自分を識別し、それは自動的に自分を端末に対して識別する。スマートデバイスを使用している場合、端末はそのＩＤをユーザのデバイスにも伝達する。サーバ２０２ａは、ユーザデバイス装置によって提供された識別を受け取り、それが維持するリストに対してそのＩＤをチェックする。それはそのＩＤを保持し、キャッシュされたデータがそのＩＤがサーバ２０２ｂに登録されていることを示すことを見る。
【０３６８】
ステップ１１０４で、サーバ２０２ａは、ユーザデバイスがサーバ２０２ｂに登録されているかを確認するためにサーバ２０２ｂへ接続する。サーバ２０２ａは、端末のＩＤをサーバ２０２ｂに伝達する。サーバ２０２ｂは、ＩＤがこれ以上登録されていないことを応答する。
【０３６９】
ステップ１１０６で、サーバ２０２ａは、ディレクトリサービス２１６に接続する。ディレクトリサービス２１６は、サーバ２０２ａの通信上の署名を検査してそれが有効であることを確認する。ディレクトリサービス２１６は、要求されたサービスに対するサービスタグに対してＩＤを検索し、ライブ情報に対するキャッシュタイムと共にサーバ２０２ｃを識別する情報で応答する。
【０３７０】
ステップ１１０８で、サーバ２０２ａは、ユーザのデバイスが同じサービスに対してサーバ２０２ｃへ登録されているかを確認するためにサーバ２０２ｃへ接続する。また、サーバ２０２ａは、端末のＩＤをサーバ２０２ｃに伝達し、ユーザのデバイスからＩＤに対する新しい詳細でそのキャッシュをアップデートする。
【０３７１】
ステップ１１１０で、サーバ２０２ｃはそれがまだ行われていなければ、この端末が登録されたサーバを検索するようにディレクトリサービス２１６に同様の要求を行う。また、端末がサーバ２０２ａに要求されたサービスに対して登録されたかを確認できる。ディレクトリサービス２１６は、ライブ情報に対するキャッシュタイムと共にサーバ２０２ａを識別する情報で応答する。
【０３７２】
ステップ１１１２で、サーバ２０２ａとサーバ２０２ｃは必要なトランザクションを行うために互いに直接通信する。
ディレクトリサービス２１６は、ユーザ２１８がユーザ２１８に割り当てられた日付と共に、ユーザ２１８が登録したユーザＩＤの新旧両方の完全な証跡を常に保持する。
【０３７３】
サーバ２０２ｃは、ＩＤがそれに登録された日付から、登録されたＩＤに関する情報のみを保持する。サーバ２０２ｂは、そのＩＤをサービスした期間に関するデータを保有する。
【０３７４】
ディレクトリサービス２１６によって提供される抽象化レイヤは、サービスを分割するにつれてさらに進む。したがって、危疑例で、サーバ２０２ａは要求されたサービスに対してユーザのデバイスを登録したサーバを識別する情報のみを要求し得る。
【０３７５】
サーバ２０２ａは、デバイスとの各通信に署名しなければならず、その署名は、通信が

関連するサービスを識別する。サーバが二以上のサービスを提供できる場合、それは当該サービスそれぞれに対して個人キーを有し、該当キーを用いて関連通信に署名する。

【０３７６】

Ｔｅｒｅｏｎサーバ自体は、上記の場合にはサーバ２０２ａ及び２０２ｂで、提供されるタグ又は情報からユーザのアカウントデータを識別するルックアップ情報を含む。したがって、サーバ２０２ｂのみがユーザデバイスのＩＤをユーザのアカウントにマッピングするデータを含む。ディレクトリサービス２１６の情報は、単にサーバ２０２ｂに対するポインタである。ユーザのデバイスは、様々なサービスのために他のサーバに容易に登録され得る。Ｔｅｒｅｏｎサーバが正確なサーバを見つけることができるのは、ユーザのデバイスＩＤとサービスを定義するクリデンシャルの組合せである。

【０３７７】

まず、サーバ２０２ａがサーバ２０２ｂと通信し、サービスタグ、ユーザＩＤ、及び任意の他の関連トランザクションデータ（例えば、年齢、通貨、金額など）を伝達すると、サーバ２０２ｂは関連ユーザのデータを検索し、トランザクションの側（ｓｉｄｅ）を行う。サーバ２０２ａは、ユーザデータを決して見ない。見ることができるのは、ユーザの認証ＩＤとサーバ２０２ｂによって伝えられたトランザクションデータだけである。

【０３７８】

同様に、サーバ２０２ｂは、端末が接続されているアカウントを識別する情報を決して見ることができない。それは単にサーバ２０２ａによって伝えられた端末ＩＤ及びトランザクションデータを見るだけである。

【０３７９】

サイキックペーパー・多面的クリデンシャル（Ｐｓｙｃｈｉｃ　ｐａｐｅｒ？　ｔｈｅ　ｍｕｌｔｉｆａｃｅｔｅｄ　ｃｒｅｄｅｎｔｉａｌ）

ディレクトリサービスのより興味深い効果の１つは、クリデンシャルが必要な時特定のサービスに合わせてカスタマイズされたアドホック多面的クリデンシャルを作成する機能である。ディレクトリサービスがこのようなクリデンシャルを提供できるように、ディレクトリサービスが作成された時点でサービスは想定された必要がない。これは「サイキックペーパー（ｐｓｙｃｈｉｃ　ｐａｐｅｒ）」として知られている。

【０３８０】

アドホック多面的クリデンシャルは、ユーザのデバイスが特定のサービスに必要なクリデンシャルになることを意味する。それはサービス認証、権限付与、又は、サービスから他の恩恵を受けるために必要な情報を正確に提供し、それがサービス提供業者が表示される全てのものである。

【０３８１】

例として、ユーザ２１８は、自分の銀行からの支払いサービス及び地域図書館での図書館借入サービスなど、複数の異なるサービスに登録している。それはＴｅｒｅｏｎに登録するとき誕生日を提供しなければならないため、自動的に年齢確認サービスへのアクセスを有する。

【０３８２】

図１２は、ディレクトリサービス２１６が、ユーザ２１８の要求したサービスに応じて、要求元サーバ（サーバ２０２ａ）を２つの他のサーバ（サーバ（２０２ｂ及び２０２ｃ））に向かうようにする方法を示す。必要に応じて、別途のサービスのための２つ以上の個別ディレクトリサービスは使用されてもよい。重要なことは、トランザクションデータが抽象化の一部であり、基本アカウントデータと分離されていることである。

【０３８３】

ユーザ２１８は、例えば、バー（サービス２）でアルコール飲み物を購入するために年齢を確認する必要がある。ステップ１２０２ないし１２１０は、図９のステップ９０２ないし９１０として実行される。この場合、サーバ２０２ａ及び２０２ｂではなく、サーバ２０２ａ及び２０２ｃの間にある。したがって、ステップ１２１０で、サーバ２０２ａ及びサーバ２０２ｃは互いに直接通信する。この場合、サーバ２０２ａは、ユーザ２１８が

２１歳以上であることを確認したい。サーバ２０２ｃは、単に彼が２１歳以上であることを確認する。

【０３８４】

運営者が法的又は規制上の要求事項により追加確認を要求すると、サーバ２０２ｃは、端末に表示するためにユーザ２１８のパスポートタイプのイメージを送ることができ、これによって運営者は彼又は彼女が実際にユーザ２１８と話していることを見ることができる。ユーザ２１８が自身をサーバ２０２ａにすでに識別したため、そうする必要がほとんどないが、サーバは正確なユーザであることを追加確認するために、ユーザ２１８が答えるための質問を送ってもよい。運営者は、ユーザの実際の年齢又は不要な個人情報を見ることができない。運営者が確認する必要があることは、ユーザ２１８がアルコールの飲み物を買うのに十分に年上であるかだけである。ユーザ２１８が自分の飲み物を支払うためにそのデバイスを使用すると、サーバ２０２ａに接続された端末はサーバ２０２ｃに再び接続するが、今回は支払いサービス（サービス１）についてである。

【０３８５】

ユーザ２１８は、自分の地域図書館で行って本を借りたい（サービス３）。ステップ１２１２で、ユーザ２１８は、自分のデバイスを使って自身をライブラリ内の端末に自分を識別させ、それは自動的にそれを端末に自身を識別する。図書館内の端末はサーバ２０２ｂに接続されている。スマートデバイスを使用している場合、端末はそのIDをユーザのデバイスに伝達する。

【０３８６】

ステップ１２１４で、サーバ２０２ｂは、ユーザのデバイスによって提供されるＩＤを受け取り、それが保持しているリストに対してＩＤをチェックする。それは該当ＩＤを保有するが、キャッシュが古くなっている。サーバ２０２ｂは、ディレクトリサービス２１６と接続する。ディレクトリサービス２１６は、サーバ２０２ｂの通信上の署名をチェックし、それが有効であることを確認する。ディレクトリサービス２１６は、要求されたサービスに対するサービスタグに対してＩＤを検索し、ライブ情報に対するキャッシュタイムと共に、サーバ２０２ｃを識別する情報で応答する。

【０３８７】

ステップ１２１６で、サーバ２０２ｂは、ユーザのデバイスがサーバ２０２ｃに登録されているかを確認するためにサーバ２０２ｃに接続する。また、サーバ２０２ｂは、端末のＩＤをサーバ２０２ｃに伝達し、ユーザのデバイスからのＩＤに対する新しい詳細でキャッシュを更新する。

【０３８８】

ステップ１２１８で、それがまだ行われていなければ、サーバ２０２ｃは、端末が登録されたサーバを検索するようディレクトリサービス２１６に同様の要求を行うことができる。また、端末がサーバ２０２ｂに要求されたサービスに対して登録されたかを確認してもよい。ディレクトリサービス２１６は、サーバ２０２ｂを識別するクリデンシャルで応答する。

【０３８９】

ステップ１２２０で、サーバ２０２ｂとサーバ２０２ｃは必要なトランザクションを行うために互いに直接通信する。サーバ２０２ｂは、ユーザ２１８が本を借りることができるかどうか（サービス３）を知りたく、サーバ２０２ｃは、ユーザが２１８が本を借りることができる図書館サービスに登録されていることを確認する（Ｔｅｒｅｏｎ運営者が図書館に提供するサービスである）。ユーザ２１８が本を借りるために料金を支払うために自分のデバイスを使用する必要があれば、端末は、サーバ２０２ｃに再び接続するが、今回は支払いサービス（サービス１）についてである。

【０３９０】

サーバ２０２ｃは、いかなるサービスを図書館に提供する必要がない。ユーザ２１８は、サーバ２０２ｄ（図示せず）のような他のサーバへ容易に登録することができ、この場合、サーバ２０２ｄは、ユーザ２１８が本を借りる可能性があることをサーバ２０２ｂに

確認する。重要なことは、最初の場合では、サーバ２０２ａはユーザ２１８が２１歳以上であることを確認するだけである。彼が本を借りることができることを知らず、ユーザ２１８がＴｅｒｅｏｎによって支払うことを知らない。同様に、サーバ２０２ｂは、ユーザ２１８が本を借りることができることを知っているが、彼が特定の年齢を越えたり、Ｔｅｒｅｏｎによって支払うことができることを知らない。

【０３９１】

要求サーバは、特定のトランザクションに対するクリデンシャル集合をまとめる必要がある場合、別途のサーバに様々な要求を出すこともできる。例えば、ユーザ２１８が年齢制限のある映画を借りたいと仮定する。この場合、要求サーバは、ユーザの年齢を確認するための１つの要求と、図書館から映画を借りるために登録されていることを確認する２種類の別途の要求を行う。Ｔｅｒｅｏｎは、図書館で要求するクリデンシャル集合を構成するために検証された個別クリデンシャルを集める。

【０３９２】

ディレクトリサービス２１６の構造は、個別クリデンシャルを伝達するサーバが分離することを可能にする。したがって、要求サーバは、特定のサービスをユーザ２１８に伝達できるか否かを確認するために必要なクリデンシャルセットを構成するのに必要な個別クリデンシャルを取得するために任意の数のサーバに問い合わせることができる。

【０３９３】

図１３は、サーバ２０２ａがユーザ２１８にサービスを提供するための多面的なクリデンシャルを構成するために３つのサーバ２０２ｃ、２０２ｄ、及び２０２ｅからクリデンシャルを取得する必要のある場合を示す。例えば、サーバ２０２ｄ上でサービス２はフィルムをレントするサービスで、サーバ２０２ｃから第１クリデンシャルとしての年齢検証、サーバ２０２ｄからメンバーシップクリデンシャル及びサーバ２０２ｅから十分な資金のクリデンシャル（ｓｕｆｆｉｃｉｅｎｔ　ｆｕｎｄｓ　ｃｒｅｄｅｎｔｉａｌ）を必要とする。

【０３９４】

関係は、必ず一対一でなく、３つのサーバそれぞれが１つのクリデンシャルだけを保有する。３つのサーバのうち、任意のサーバは、それぞれ１つ以上のクリデンシャルをサーバ２０２ａに伝達し得る。これはサーバ２０２ａに１つのクリデンシャルのみを伝達してもよい。クリデンシャルの数は関係ない。重要なことは、サーバ２０２ａでユーザ２１８がサービスにアクセス可能にするために必要なクリデンシャルを取得するために１つ以上の外部サーバと接触することにある。

【０３９５】

ユーザ２１８が端末にアクセスするサーバ２０２ａは、一部のサービスをユーザ２１８に伝達するために必要なクリデンシャルをすでに保有していてもよい。しかし、データ保護の目的で、ユーザ２１８は、サーバ２０２ａに特定の詳細（例えば、年齢など）を提供することを所望しない。全てのサーバ２０２ａで使用２１８が特定の年齢を越えたり特定の商品を注文するよう許可されていることを検証する必要がある場合、その問い合わせを確認したり拒否するサーバへ簡単に接続できる。これはＥコマースウェブサイトに極めて有用である。それらは正確な詳細を知らなくても特定の事実やパラメータを確認できる。基本的に、ディレクトリサービス２１６は、ゼロ知識証明提供者又は機密公証人としての役割を果たす。Ｔｅｒｅｏｎは、その事実が何であるかを開示せず、事実又はパラメータをサーバ２０２ａに証明したり反証することができます。

【０３９６】

したがって、特定のサービスに対するクリデンシャルは、２０２ａ、２０２ｃ、２０２ｄ、２０２ｅ及び他のサーバからのクリデンシャルを含んでもよい。クリデンシャルは、１つのサーバ上にあっても、様々なサーバに分散してもよい。

【０３９７】

個人や組織は、開示する必要のない情報を公開する必要がなく、サービスを受ける権利があることを証明できるため、極めて強力である。再び、Ｅコマースウェブサイトの例を

とると、ユーザ２１８は、自分の名前とアドレスをウェブサイトに登録してもよい。しかし、その銀行は支払いクリデンシャルを保有し、政府のサーバは、彼が制限されたアイテムを購入できる権限のあるという事実を登録し、地域鉄道会社は彼の旅行承認を保有し、彼の保健当局のサーバはその年齢を確認できる。

【０３９８】

サービスに対するクリデンシャルのアドホック集合を組み合わせる方法は、ユーザと該当デバイスにしか適用されない。たとえば、異なる時間に異なるサービスへ接続されなければならないＩｏＴデバイスのような自律センサ、デバイス及びサービスにも同様に適用できる。このようなクリデンシャルの集合が要求されたとき、それらのサービスに必要なクリデンシャルを簡単に組み合わせることができる。

【０３９９】

アカウントの切り替え（Ａｃｃｏｕｎｔ　ｓｗｉｔｃｈｉｎｇ）

新しいシステムの採択を遅延させる主な問題は、損失又はサービスの中断なしにデータをレガシーシステムから新しいシステムへ送信することの困難さが認識されていることである。同じ問題がシステムのアップグレードに影響を及ぼし、運営者は、アップグレード又はアップデート時にデータを失う危険性に対する認識により、アップグレード及びアップデートではなく、初期ハードウェア及びソフトウェアの構成をそのまま使用する場合が多い。

【０４００】

ディレクトリサービス２１６は、データ、アカウント及び構成情報を１つのサーバ又はデータストアーから他のサーバ又はデータストアーへ円滑に移動させるメカニズムを提供することで、このような問題を解決する。機関間のリアルタイム口座振込を支援するブロックの１つは、未定の支払い（ｉｎ－ｔｈｅ－ａｉｒ　ｐａｙｍｅｎｔｓ）を把握して処理する方法の問い合わせである。この産業は、現在に合計１８ヶ月かかるアカウント振込みシステムを有している（初期の切り替えの場合、７日後、支払い又は振込みを受けとるまで１８ヶ月）。これは、あるデータストアーから他のデータストアーにデータの集合を切り替えるのにも適用できる。

【０４０１】

ディレクトリサービス２１６は、基礎となるサービス、サーバ及び実際のユーザアカウントからユーザの認証ＩＤを分離する抽象化レイヤを提供する。したがって、ユーザ２１８は、自分のデバイスが登録されたサービス及び基本サーバを変更する間に自身の認証ＩＤを維持できる。

【０４０２】

アカウントの切り替えプロセスは、例で最もよく説明されている。この例では、ユーザ２１８は、銀行Ａと取引する（ｂａｎｋ）。図１４は、銀行Ａ及びＴｅｒｅｏｎサーバ２０２ａとのユーザ関係を示す。ユーザ２１８がまだ顧客でなくても、銀行Ｂはサーバ２０２ｂ上でＴｅｒｅｏｎを支援する。ユーザ２１８は、自分のアカウントを銀行Ａから銀行Ｂへ移動させることを決定する。

【０４０３】

図１５は、ユーザ２１８が自分のアカウントを銀行Ａから銀行Ｂへ振り込むために着手するプロセスを示す。この例で、ユーザ２１８は、銀行Ａから超過に引き落とされず、ローンもない。

【０４０４】

ステップ１５０２で、ユーザ２１８は、銀行Ｂにアカウントを開設し、そのカード及びモバイルをその銀行及びＴｅｒｅｏｎサーバ２０２ｂに登録する。

ステップ１５０４で、銀行ＢのＴｅｒｅｏｎサーバ２０２ｂは、Ｔｅｒｅｏｎディレクトリサービス２１６上でユーザのモバイル及びカードのＰＡＮを検索し、その全てが銀行Ａに登録されているかを検出する。

【０４０５】

ステップ１５０６で、銀行ＢのＴｅｒｅｏｎサーバ２０２ｂは、自分の登録を銀行Ｂに

移動したいことを確認するためにユーザ２１８と接触し、ユーザ２１８は、特に、この目的のためにユーザ２１８に送られた追加認証コードを入力することでこれを確認する。

【０４０６】

ステップ１５０８で、銀行ＢのＴｅｒｅｏｎサーバ２０２ｂは、銀行Ａのサーバ２０２ａに接続し、ユーザ２１８が自分のアカウント及びＩＤに対して銀行Ｂへの移動を要求し、これを確認したことを銀行Ａのサーバ２０２ａに通知する。

【０４０７】

ステップ１５１０で、銀行ＡのＴｅｒｅｏｎサーバ２０２ａは、ユーザ２１８にアカウントへの移動を所望するかを確認する要求を送信し、ユーザ２１８は自分の移動を確認する。

【０４０８】

ステップ１５１２で、銀行ＡのＴｅｒｅｏｎサーバ２０２ａは、これを銀行ＢのＴｅｒｅｏｎサーバ２０２ｂと確認し、ユーザのアカウント登録、残高、構成、支払い指示などをユーザＢのサーバ２０２ｂに通知する。銀行Ｂのサーバ２０２ｂは、このようなアカウントを銀行Ａと同じ方式で設定したり、提供権限の付与されたサービスを提供するためにできる限り近く設定する。

【０４０９】

例えば、ユーザ２１８は、ＧＢＰ、ＵＳＤ及びＥＵＲを保有可能にする銀行Ａに３つの分離した通貨アカウントを有する。残念ながら、銀行Ｂは、ＧＢＰとＵＳＤアカウントのみを提供しているが、それは任意のアカウントとの間でＥＵＲの支払いを受けることができる。銀行Ｂのサーバ２０２ｂは、ユーザがアカウントを開設すればこれをユーザ２１８に通知し、ユーザはＥＵＲをＧＢＰに変換することを決定する。その後、銀行Ｂは、銀行ＡにＧＢＰとしてＥＵＲを送るように指示する。

【０４１０】

ステップ１５１４で、銀行ＢのＴｅｒｅｏｎサーバ２０２ｂは、ユーザのＩＤがサーバ２０２ｂに登録されていることをディレクトリサービス２１６に通知する。

ステップ１５１６で、銀行ＢのＴｅｒｅｏｎサーバ２０２ｂは、ディレクトリサービス２１６にユーザのＩＤを登録したことを銀行Ａのサーバ２０２ａに通知し、銀行Ａに残高を振り込むように指示する。

【０４１１】

ステップ１５１８で、銀行Ａは、これ以上ユーザのＩＤを管理しないことをディレクトリサービス２１６で確認する。ディレクトリサービス２１６は、新しいＩＤ登録に対する開始日時を銀行Ｂに設定し、銀行Ａに対する旧登録に対して終了日時をフィールドに設定する。銀行Ａは、ディレクトリサービスを設定し、これ以上のユーザアカウントを保有していないユーザ２１８に支払うことを試みる任意のサーバへ通知し、該当サーバにユーザの詳細をディレクトリサービス２１６内で検索するように指示する。終了日フィールドに日時を入力しこれを実行する。銀行Ｂは、初めには銀行Ａに接続されたユーザ２１８に支払われた全ての支払いを受信する。

【０４１２】

ディレクトリサービス２１６は、ユーザ２１８が新しいアカウントに切り替えた後ユーザの古いアカウントに行われた支払いである未定の支払い（ｉｎ－ｔｈｅ－ａｉｒ　ｐａｙｍｅｎｔｓ）をキャッチし得る。同じ方式で、Ｔｅｒｅｏｎは、古いアカウントから支給される予定の延期された支払い（ｄｅｆｅｒｒｅｄ　ｐａｙｍｅｎｔｓ）も受けとることができる。残高が送信されること、これらのアカウントは新しいアカウントから出金され、このタスクにはは数日、数週間又は数ヶ月でなく数分かかる。

【０４１３】

ステップ１５２０で、銀行Ａは、残高を銀行Ｂに振り込む。銀行Ｂは、銀行Ａに資金を受信したことを通知する。

ステップ１５２２で、銀行Ａは、ユーザのアカウントをクローズし、そのように行った新しい銀行で残高を振込んだことをユーザ２１８に通知する。

【０４１４】
ステップ１５２４で、銀行Ｂは、銀行Ａから自分の残高を受信したことをユーザ２１８に通知する。
ユーザ２１８が銀行Ａで自分のアカウントの１つ以上に超過に引き落とされ、銀行Ｂが自分の事業を引き受けることに同意した場合、ステップ５１６及び５２０で銀行Ｂは残高を銀行Ａに振込み、銀行Ｂのユーザに対応するアカウントは引き落とされる。ユーザ２１８は、銀行Ｂに自分のアカウントを振り込む前に超過に引き落とされること（ｏｖｅｒｄｒａｆｔ）をクリアするため、銀行Ａのアカウント間で資金を振り込むことを決定してもよい。
【０４１５】
支払いの場合、Ｔｅｒｅｏｎ番号付けシステム（Ｔｅｒｅｏｎ　ｎｕｍｂｅｒｉｎｇ　ｓｙｓｔｅｍ）は、ユーザ、組織、アカウント、サービスのタイプ及びトランザクションを区分する。それらは全て別途の番号付けシステムを有する。このような特性は、ディレクトリサーバは、ユーザ２１８が自分のアカウントを新しいサービス提供者にリアルタイム移動させるプロセスを管理することができる。ディレクトリサービス２１６の構造は、トランザクションをリアルタイムで処理する能力と共に、ユーザが数日ではなく数分でアカウントを変更することを可能にする。
【０４１６】
ディレクトリサービス２１６は、上述したように、全てのトランザクションのリアルタイム処理と共に未定の支払い（ｉｎ－ｔｈｅ－ａｉｒ　ｐａｙｍｅｎｔ）のような未定のトランザクション（ｉｎ－ｔｈｅ－ａｉｒ　ｔｒａｎｓａｃｔｉｏｎ）の問題を除去する。Ｔｅｒｅｏｎでは、トランザクションは単に未定の状態に進入できない。それらは完了したり取り消される。
【０４１７】
Ｔｅｒｅｏｎは、銀行アカウントの移動性（ｂａｎｋ　ａｃｃｏｕｎｔ　ｐｏｒｔａｂｉｌｉｔｙ）、市場での競争を増加させる機能、そして銀行及び規制機関は実現できないものと考えている機能などの、アカウントの移動性（ａｃｃｏｕｎｔ　ｐｏｒｔａｂｉｌｉｔｙ）という概念を支援する。Ｔｅｒｅｏｎは、アカウントの詳細を直接使用せず、各支払人（ｐａｙｅｒ）と受取人（ｐａｙｅｅ）を識別するために別途のクリデンシャルを使用することから、ユーザ２１８とユーザの銀行アカウントの詳細間に抽象化を挿入する。ディレクトリサービス２１６が提供する抽象化は、アカウントスイッチング及び移動性を容易にする。
【０４１８】
クリデンシャルの変更（Ｃｈａｎｇｉｎｇ　ｃｒｅｄｅｎｔｉａｌｓ）
ディレクトリサービス２１６は、運営者及びユーザが既存のＩＤクリデンシャルを新しいクリデンシャルに変更し、ＩＤの以前のユーザとのトランザクションを混乱させることなく、過去のクリデンシャルを再利用することを可能にする。ディレクトリサービス２１６によって提供される抽象化レイヤは、Ｔｅｒｅｏｎがこれを可能にする。
【０４１９】
ユーザ２１８が自分のアカウントを他のサーバに振り込む場合、ユーザ２１８は、ＰＡＮのような特定クリデンシャルを保有することができ、又は、サーバがユーザ２１８に新しいクリデンシャルを発行し得る。後者の場合、本来のサーバはほとんどすぐにクリデンシャルを再利用できる。各クリデンシャルは、それがユーザ２１８に発行されるときを反映する日時スタンプを有するため、特定のクリデンシャルの新しいユーザ２１８はそのクリデンシャルをほとんどすぐに使用できる。
【０４２０】
各クリデンシャルは、特定のサーバの特定のユーザに発行された日時スタンプを有する。各トランザクションも日時スタンプを保持し、各Ｔｅｒｅｏｎサーバも各トランザクションに使用されたクリデンシャルを保有するため、Ｔｅｒｅｏｎはトランザクションを正しい配信先にルーティングするためにこのコンポーネントを使用する。例えば、ユーザ２

１８は、クリデンシャルＡ（例えば、モバイル番号）を有するマーチャントから何かを購買し、他のクリデンシャルＢ（例えば、新しいモバイル番号）を使用する必要があるとき、数日後に他の銀行に移動する。アイテムが欠陥のある場合、後でユーザ２１８はアイテムをマーチャントに送り返す。マーチャントは、トランザクションを探して払い戻せば良い。本来のトランザクションがクリデンシャルＡを使用したが、クリデンシャルＡに対するサーバは、クリデンシャルの変更を示す日時スタンプを報告する。マーチャントのサーバは、クリデンシャルＡを探して、トランザクション当時にクリデンシャルＡを使用したユーザ２１８が現在にクリデンシャルＢを使用していることを発見する。サーバは、クリデンシャルＢに対するサーバ（クリデンシャルＢに対するユーザ２１８がトランザクション当時にクリデンシャルＡを使用したことを確認する）に接続し、サーバは払い戻しのプロセスを開始する。

【０４２１】

Ｔｅｒｅｏｎのセキュリティーモデルでは全ての通信に署名しなければならないため、ユーザＡはユーザＢが不正でないことを確信できる。サーバ２０２ｂは、ライセンスサーバからの有効なライセンスを有する場合にのみその通信に署名し、サーバ２０２ｂがデバイスのライセンスを発行して確認することからユーザＢのデバイスはサーバ２０２ｂが有効な場合、その通信を署名する。ユーザＢがトランザクションを承認するかデバイスのアプリケーションにアクセスするために必要な正確なクリデンシャルを知らない限り、ユーザＢはトランザクションを完了できない。

【０４２２】

更なる例として、ユーザは自分の電話帳に連絡先のモバイル番号を入力し、その連絡先にサプライズＰ２Ｐ振込みを所望することもある。Ｔｅｒｅｏｎは、該当番号に対するレコードを検索し、上記のように連絡先がモバイル番号に変更されたことを検出する（連絡先がＴｅｒｅｏｎユーザである場合）。新しいサーバ番号を使用するユーザが以前のサーバに登録された古い番号を使用したことを正確なサーバで確認する。また、Ｔｅｒｅｏｎは、特定の承認された連絡先が古いクリデンシャルを介してトランザクションを試みるとき、ディレクトリサーバで該当ユーザのモバイル番号又は異なるＴｅｒｅｏｎクリデンシャルをアップデートできるように連絡先が自分のアカウントを設定する機能を支援する。この例で、叔母の姪が家族全員をアップデートするようにアカウントを設定しているため、次に叔母が自分の連絡先リストにアクセスすれば、彼女は自分の姪の新しいモバイル番号を確認できる。

【０４２３】

図１６は、サーバ２０２ａ、サーバ２０２ｂ、及びディレクトリサービス２１６に対する例を示す。ここで、古いユーザは、自分のアカウントをサーバ２０２ａからサーバ２０２ｂで移転した。２０２ａは銀行Ａのサーバ、２０２ｂは銀行Ｂのサーバである。

【０４２４】

古いユーザは、最初に自分のＩＤとしてモバイル番号１を使用する。そのアカウントを移転した後、彼はしばらくの間モバイル番号１を続けて使用する。ユーザ２１８、ディレクトリサービス２１６、及びサーバ２０２ａ及び２０２ｂ間の通信は、図１５に図示されて上述したように行われる。ディレクトリサービスのエントリは、ユーザ２１８が日時１（ｄａｔｅ－ｔｉｍｅ１）から日時３（ｄａｔｅ－ｔｉｍｅ　３）までサーバ２０２ａを使用し、日時２（ｄａｔｅ－ｔｉｍｅ　２）から彼がサーバ２０２ｂを使用したことを示す。僅かな重複は、全ての未定の支払いがキャッチされ、ユーザが自分のＩＤが登録されているサーバを有しないという時間的なギャップがないことを保証することである（アカウントの移行先のサーバがその移行のすべての日時エントリとＩＤエントリを制御可能にすることで、日時エントリが重複しないようにすることができる。これがシステム移行の動作方法である）。

【０４２５】

ある時点で、ユーザ２１８は、モバイル番号を変更することを決定した。新しいモバイル番号２を自分のＩＤとしてサーバ２０２ｂに登録し、モバイル番号１を登録解除する。

サーバ２０２ｂは、ディレクトリサービス２１６に変更を通知する。これはユーザが日時４にモバイル番号２を自分のＩＤとして使い始めたことを示し、モバイル番号１が日時５でサーバ２０２ｂに対してＩＤでの使用が中止されたことを示す。

【０４２６】

後で、新しいユーザはサーバ２０２ａにアカウントを生成し、日時６で自分のＩＤとしてモバイル番号１を登録する。新しいユーザへ古いユーザの以前のモバイルが提供された場合もあり、該当番号がモバイル運営者によって再利用のために解除されることもあり得る。サーバ２０２ａは（ＩＤが利用可能であることを確認した後）ＩＤを登録したことをディレクトリサービス２１６に通知し、ディレクトリサービスは、日時６の時点でモバイル番号１がサーバ２０２ａに登録されていることを示す。

【０４２７】

図１６に示すように、古いユーザが銀行Ａ２０２ａによって発行されたカードを使用する場合、まずユーザ２１８が自分のアカウントを銀行Ｂ２０２ｂに送信すると、銀行はＰＡＮのようなクリデンシャルを用いて新しいカードをユーザ２１８に発行する。ユーザ２１８はカードを受け取ると、カードを活性化し、銀行Ｂのサーバ２０２ｂは、ユーザの本来のクリデンシャルがこれ以上使用されないことを銀行Ａのサーバ２０２ａに通知する。銀行Ｂは、Ｔｅｒｅｏｎディレクトリサービス２１６に新しいクリデンシャルを登録する。ユーザ２１８は、本来のクリデンシャルを保持することを要求してもよく、銀行Ａが要求に同意した場合、そのようにするために彼は銀行Ａから小額が割り与えられる。したがって、Ｔｅｒｅｏｎはカード番号又はＰＡＮの移動性を支援する。

【０４２８】

ユーザは、将来のある時点で、最初に銀行Ａから発行したカードの使用を中断し、該当クリデンシャルを解除してもよい。銀行Ｂがそれを解除した後、又はユーザがアカウントを銀行Ｂへ振込んだ後６ヶ月の間に、銀行Ａは該当ＰＡＮのクリデンシャルを再利用できない。正確な時間は、銀行の規制当局が許容する内容に応じて異なる。ここで、時間が経過すると、ディレクトリサービス２１６がモバイル番号、ＰＡＮ又は他のクリデンシャルを含まないため、それはクリデンシャルを使用することができる。また、それはクリデンシャルの登録された日付、ユーザ基準として期限切れになった日付、又はユーザごとにリリースされた日付のリストを含む。

【０４２９】

アカウントの切り替え方法は、システムが未定の支払いをキャプチャー可能にする。また、以前のトランザクションで使用されたクリデンシャルに基づいて以前のトランザクションから後続のトランザクションを送信できる極めて柔軟で強力な方法を提供する。以前のトランザクションに対する払い戻しは、実際の事例の１つである。元のＩＤが後で再び使用された場合にもディレクトリサービス２１６がサーバに正しいＩＤを支払うように指示するため、古いＩＤに対して返金するマーチャントは、正しいアカウントを返金できる。ＥＭＶ及び現在のモバイルルックアップ技術は、数字が再利用されることは決してないと仮定する。残念ながら、彼らは時にはそうである。

【０４３０】

これについて図１６で示されている。日時１と日時２の間のある時間で、古いユーザがモバイル番号１をＩＤとして有するデバイスを用いてマーチャントからアイテムを購入すると仮定する。後でそのアイテムが誤ったことが分かり、ユーザは払い戻しを所望する。

【０４３１】

ユーザ２１８が払い戻しのために日時１と日時２の間にマーチャントへ行く場合、Ｔｅｒｅｏｎシステムは、マーチャントシステムがシステム２０２ａ上のユーザアカウントに払い戻しの支払いを行うよう指示する（ユーザがまだアカウントを閉鎖していないため）。

【０４３２】

ユーザ２１８が払い戻しのために日時２と日時４の間にマーチャントへ行く場合、アイテムに対する支払いが元のサーバ２０２ａからきたにもかかわらず、Ｔｅｒｅｏｎシステ

ムは、マーチャントシステムがシステム２０２ｂ上のユーザアカウントに払い戻しするように指示する。

【０４３３】

アカウントの切り替え方法は、ユーザの新しいＩＤも考慮されるのであろう。ユーザ２１８が払い戻しのために日時４の後マーチャントに行き、そのモバイル番号２をそのＩＤとして使用した場合、たとえアイテムに対する支払いが元のサーバ２０２ａからきたとしても、ユーザが本来自分の支払いＩＤでモバイル番号１を使用したにもかかわらず、Ｔｅｒｅｏｎシステムは、マーチャントシステムがシステム２０２ｂ上のユーザアカウントに払い戻しするように指示する。

【０４３４】

ＰＡＮ、電子メールアドレス、その他の再利用可能なクリデンシャルのレコードも同一に保持される（明らかな理由で生体クリデンシャル（Ｂｉｏｍｅｔｒｉｃ　クリデンシャル）は再利用できない）。

【０４３５】

このシステムは、クリデンシャルをあらゆるレベルに細分化できる。この支払い方法の１つの例は、通貨又は通貨コードを含む。ここで、ユーザは同じ通貨又は別途のサーバで互いに異なる通貨に対して互いに異なるＩＤを使用してもよい。

【０４３６】

図１７は、サーバ２０２ｂ、サーバ２０２ｃ、及びディレクトリサービス２１６に対する例を示す。図面において、ユーザ２１８は、図１５に示すように管理されるサーバ間の通信を介して図１６に示すものと同じ方式により、サーバ２０２ｂからサーバ２０２ｃへ自分のアカウントをすでに移動させている。

【０４３７】

ユーザ２１８は、初期に自分のＩＤとしてモバイル番号１を使用する。そのアカウントを移動した後、彼は通貨１及び通貨２内のトランザクションに対してしばらくの間にモバイル番号１を続けて使用する。ディレクトリサービス２１６のエントリは、ユーザ２１８が日時１から日時３までサーバ２０２ｂを使用し、日時２から彼がサーバ２０２ｃを使用したことを示す。僅かな重複は、全ての未定の支払いがキャッチャされ、ユーザが自分のＩＤの登録されているサーバを有しない場合に、時間差がないようことを保証することにある。

【０４３８】

ある時点で、ユーザ２１８は、通貨２内のトランザクションに対して新しいモバイルを使用することを決定した。彼は、通貨２内のトランザクションに対して自分の新しいモバイル番号２を自分のＩＤとしてサーバ２０２ｂに登録した。サーバ２０２ｂは、ディレクトリサービス２１６に変更を通知した。これは、ユーザが日時４で通貨２の全てのトランザクションに対してモバイル番号２を自分のＩＤとして使い始め、モバイル番号１が日時５までの全てのトランザクションに対してＩＤでの使用が中止されたことを示す。

【０４３９】

図１７ａは、サーバ２０２ｂ、サーバ２０２ｃ、及びディレクトリサービス２１６に対する異なる例を示す。図面において、ユーザ２１８は、図１５に示すように管理されるサーバ間の通信を介して図１６に示されたものと同じ方式でサーバ２０２ｂからサーバ２０２ｃへ自分の通貨１アカウントをすでに移動させた。

【０４４０】

アカウントを移動した後、ユーザはモバイル番号１を使用する時間の間に、通貨１と通貨２間のトランザクションを続く。ディレクトリサービス２１６内のエントリは、ユーザ２１８が日時１から日時３までサーバ２０２ｂを使用し、日時２から彼が通貨１内のトランザクションに対してモバイル番号１を自分のＩＤとしてサーバ２０２ｃに使用したことを示す。また、ディレクトリサービスエントリは、ユーザが通貨２内のトランザクションに対してモバイル番号１を自分のＩＤとしてサーバ２０２ｂに続けて使用したことを示す。

【０４４１】
ある時点で、ユーザ２１８は、通貨２内のトランザクションに対して新しいモバイルを使用することを決定する。彼は通貨２内のトランザクションに対して自分の新しいモバイル番号２を自分のＩＤとしてサーバ２０２ｂに登録した。サーバ２０２ｂは、ディレクトリサービス２１６に変更を通知する。これはユーザが日時４で通貨２内の全てのトランザクションに対してモバイル番号２を自分のＩＤとして使い始め、モバイル番号１が日時５までの全てのトランザクションに対してＩＤでの使用が中止されたことを示す。
【０４４２】
日時４より前では、ユーザ２１８は、自分のモバイル番号１を全てのトランザクションに対してＩＤとして使用した。ディレクトリサービス２１６は、単にトランザクションが通貨２である場合にトランザクションをサーバ２０２ｂに向かうようにし、トランザクションが通貨１内にあれば、トランザクションをサーバ２０２ｃに向かうようにした。トランザクションの伝えられるサーバを制御するクリデンシャルの完全な集合であるため、ユーザが２つのサーバに同じＩＤを登録したという事実は無関係である。日時２の後に初めて通貨１でユーザとトランザクションするマーチャントのシステムは、ユーザが前に該当の通貨内のトランザクションに対してサーバ２０２ｂを使用したことを知らない。同様に、マーチャントのシステムが通貨２でユーザとトランザクションを開始しない限り、マーチャントのシステムは、ユーザが通貨２内のトランザクションに対して同じＩＤをサーバ２０２ｂを使用したことを知らないのであろう。
【０４４３】
Ｔｅｒｅｏｎは、単にユーザ２１８を１つのネットワークから別のネットワークに切り替えること以上の役割をする。すでに前述したように、ユーザを切り替える一般的な方法は、未定の支払いを処理できない。創始者などによって主張されたように、現在の利用可能な最も進歩したアカウントの切り替えシステムは、ユーザが自分を待つ前に支払い金を受けるために１８ヶ月の手動プロセスを必要とする。１８ヶ月の間に、銀行とユーザは古いアカウントから新しいアカウントへ既存の全ての支払い命令を移行するように努めなければならない。Ｔｅｒｅｏｎはこの要件を完全になくしたのである。
【０４４４】
現在、銀行は支払いクリデンシャルを再利用できない。Ｔｅｒｅｏｎのアカウントの切り替えメカニズムは、このような制限を取り除き、規制機関から許容された場合、特定期間が過ぎた後銀行がＰＡＮ及びアカウント番号を再発行できる。
【０４４５】
この方法は、をアカウントの切り替え機能とと呼ばれるが、実際には、基本アカウントの切り替えに加えて多くのアプリケーションがある。例えば、銀行のコアシステムが障害が生じた場合、バックアップサービス提供者に障害克服（ｆａｉｌｏｖｅｒ）を提供し、情報の損失なく、１つのデータ形式から別のデータ形式に変換することで、あるシステムから他のシステムにデータを移行移行できる方法を提供する。
【０４４６】
更なる例は、モバイルシステムで番号の移動性（ｎｕｍｂｅｒ　ｐｏｒｔａｂｉｌｉｔｙ）を簡素化することにある。現在、ユーザが自分のモバイル番号をある供給者から他の供給者に切り替えた場合、第１供給者は、新しい供給者に対する全ての呼出（ｃａｌｌ）を再びルーティングしなければならない。ユーザが第３供給者に切り替えた場合、第１供給者は、第２供給者に呼出をルーティングしなければならず、第２供給者は、第３供給者に呼出をルーティングしなければならない。これは効率的ではなく、多くのコストがかかるが、運営者は、番号の移動性を支援しなければならない。Ｔｅｒｅｏｎは、呼出を何度もルーティングする必要がなくなる。
【０４４７】
運営者が番号の移動性を支援するためにＴｅｒｅｏｎを使用すれば、彼らは複数のホップを支援する必要がない。ユーザ自分の番号を第１運営者から第２運営者に移すことと決定すれば、第２運営者は、単にディレクトリサーバに現在の該当モバイル番号を支援する

ことを通知だけすればよい。第１運営者は、該当番号に対する呼出をディレクトリサーバに送信すれば第２運営者に呼出がルーティングされる。ユーザが自身の番号を再び移転するごとに、新しい運営者は、ディレクトリサーバに変更事項を通知し、ディレクトリサーバは、該当番号をサービスする運営者に呼出をルーティングする（ユーザが全世界に固有なＩＢＡＮのような銀行アカウントを保有している場合、Ｔｅｒｅｏｎは、モバイル番号の移動性を支援するのと同じ方式で銀行アカウントの移動性を支援する）。

【０４４８】

同様の例は、物理的機械、論理機械、仮想機械、コンテナ、又は、実行可能なコードを含む他の一般的に用いられるメカニズムなどの単純移行は充分でないＴｅｒｅｏｎシステムをアップグレードするために、運営者が１つのサーバから別のサーバにＩｏＴサービスとデバイスを移行する例である。

【０４４９】

他の例は、システムが移行ツールとして作動することにある。例えば、これは運営者が、あるバージョンのＴｅｒｅｏｎシステムからジョンでアップグレードされたバージョンでデバイスの登録されたアカウントと共にサービスを移行しようとする場合である。運営者は、デバイス登録、アカウント、及びシステム構成（ｓｙｓｔｅｍ　ｃｏｎｆｉｇｕｒａｔｉｏｎｓ）を新しいサーバに送信するように古いサーバを設定し、システムがその送信を行う。各アカウントは、データ及び監査ログ（ａｕｄｉｔ　ｌｏｇｓ）と共に送信され、送信進行によってサーバはディレクトリサービス２１６をアップデートする。今、その分野のデバイスが、支払いデバイス、交通センサ、ＩｏＴデバイスなどのサーバと通信することを所望する場合、ディレクトリサービス２１６は、それらを古い又は新しいサーバに単にリダイレクトする。アカウントが送信される前または後に、彼らはサーバに接続する。

【０４５０】

上記の例は、Ｔｅｒｅｏｎがクリデンシャル移動性を容易にし、アドホック多面的なクリデンシャルを支援する方法を示す。これは幅広いアプリケーションを保有し、Ｔｅｒｅｏｎをネットワークでクリデンシャルを管理しなければならないことの全てのネットワーク分野に適用する。

【０４５１】

拡張可能なフレームワーク（Ｅｘｔｅｎｓｉｂｌｅ　Ｆｒａｍｅｗｏｒｋ）

既存のトランザクション処理システムのワークフローは、本質的に静的なものである。ひとまず具現されると、それらは変更することが難しく、システムが支援するサービス又は動作は柔軟性がない。

【０４５２】

今まで、支払い提供者がサービスを開始した場合、当該サービスに対する支払いパターンは静的であった。提供者は交換又は修正されたサービスを開始し、当該のサービスを支援するために新しいカード又はアプリケーションを発行することで当該サービスを修正のみを行ってもよい。これがＥＭＶの深刻な弱点に対する普遍的な知識にもかかわらず、既存のすべてのＥＭＶカードをリコールしてＥＭＶ支払いインフラを再プログラミングして実行し、新しいカードを発行する意味を意味し、システムを修正できない理由のうちの１つである。カードこれは何千もの発行者と取得者が協力することを要求する。

【０４５３】

ＴｅｒｅｏｎはＳＤＡＳＦを用いて全ての機能をバックエンドに置き、バックエンド（ｂａｃｋ－ｅｎｄ）は、プロセスを介してリアルタイムにマーチャントデバイスを案内できる。これにより、サービス提供者が個別ユーザだけ細部的な新しいサービスを作成できる。

【０４５４】

拡張可能なフレームワークは、Ｔｅｒｅｏｎシステム内に位置するフレームワークで、Ｔｅｒｅｏｎシステムを再構成する必要がなく、新しいサービスを追加し得る。拡張可能なフレームワークは、Ｔｅｒｅｏｎシステムに様々な利点を提供するためにディレクトリ

サービス２１６と共に作動する。

【０４５５】

柔軟なメッセージ構造（Ｆｌｅｘｉｂｌｅ ｍｅｓｓａｇｅ ｓｔｒｕｃｔｕｒｅ）

拡張可能なフレームワークは、柔軟なメッセージ構造（可変長フィールドのある全てのデータ又はレコードタイプが提供され、Ｔｅｒｅｏｎシステムがレガシー又は互換されないシステムで作動するようにフィールドの長さを修正できる）によって部分的に提供される。

【０４５６】

拡張可能なフレームワークは、標準的なプロセスの順序を変更することによって通信インフラストラクチャーに追加のセキュリティーレイヤを追加できる。多くの産業分野で支払いは単に一例であり、通信は、固定されたメッセージ構造を使用する。通信が暗号化された場合にも、これは犯罪者が悪用できる脆弱さが発生させる。構造化されたメッセージは、徹底的な攻撃に脆弱である。組織及び他の組織は、ＨＭＡＣ（ｈａｓｈ ｍｅｓｓａｇｅ ａｕｔｈｅｎｔｉｃａｔｉｏｎ ｃｏｄｅ）を用いてメッセージの無欠性を保護できるが、ＨＭＡＣはメッセージが引き付けるべき絶対的な機密性を保管しない。

【０４５７】

拡張可能なフレームワークは、全てのトランザクション処理システムに対して静的システムの問題を解決する。それは既存のシステム及びサービスと共に作動できる柔軟性を提供し、提供者が既存のサービスをアップデートし、インフラストラクチャーを再び稼動したり、カードのような新しいエンドポイントデバイス（ｅｎｄ－ｐｏｉｎｔ ｄｅｖｉｃｅ）を発行する必要がなく、新しいサービスを構築できるようにする。これについては下記で説明する。

【０４５８】

乱読化（Ｏｂｆｕｓｃａｔｉｏｎ）

構造化されたメッセージ形式を有するシステムが直面する理論的なリスクの１つは、メッセージ形式を繰り返し使用すると、ハッカーが無差別な攻撃に使用できる十分な資料を提供する。これは、いずれかの形態のランダムシード（ｒａｎｄｏｍ ｓｅｅｄｉｎｇ）を用いて暗号化アルゴリズムを正しく実現していないシステムに該当する。

【０４５９】

拡張可能なフレームワークにより、運営者とユーザはデバイスとサーバとの間に構造化メッセージを送信する必要性をなくす。代わりに、メッセージは乱読化される。

Ｔｅｒｅｏｎの各トランザクション通信は、該当フィールドのレーベルと共に２つ以上のフィールドを含む。通信ごとにフィールドの固定順序をたどる代わりに、順序をランダムに変更することができる。各フィールドには常に識別タグが付いているため、通信の両端にあるデバイスをまず復号化し、次にフィールドを処理する前に順序付けするようにする必要がある。

【０４６０】

例えば、ＪＳＯＮ（ＪａｖａＳｃｒｉｐｔ Ｏｂｊｅｃｔ Ｎｏｔａｔｉｏｎ）の資料で提供される例からの抜粋を利用すれば（もちろん他の形式もシステム内にあり、使用される）、次の３つの表現が同一である。

・{“version”: 1, "firstName": "John", "lastName": "Smith", "isAlive": true, "age": 25}

・{“version”: 1, "firstName": "John", "isAlive": true, "lastName": "Smith", "age": 25}

・{"age": 25, "firstName": "John", "isAlive": true, "lastName": "Smith", “version”: 1}

攻撃者は自分が有するサイファーテキスト（ｃｙｐｈｅｒ ｔｅｘｔｓ）に既存の同じ順のような情報を含んでいるか分からない。乱読化の正確なモードは、使用されている形式及び使用されているシリアル化プロトコルによって異なるが、原則は変わらない。

【０４６１】

乱読化モードは、追加の利点を有する。予め定義された通信のコンテンツは、通信プロトコルを壊すことなく拡張される。デバイスが処理できないフィールドを受信すると、デバイスは、該当フィールドとその値は単に廃棄される。したがって、１つ以上のランダムフィールド及び値の対（ｐａｉｒ）はシステムが廃棄することに含まれ、追加的な不確実性を通信に追加する。

【０４６２】

したがって、次の３つ通信は同一である。

・{“version”: 1, "firstName": "John", “nonce”: 5780534, "lastName": "Smith", "isAlive": true, "age": 25}

・{“whoknows”: “698gtHGF”, “version”: 1, "firstName": "John", "isAlive": true, "lastName": "Smith", "age": 25}

・{"age": 25, "firstName": "John", "isAlive": true, "lastName": "Smith", “whatis this”: “Jor90%hr,” “version”: 1}

上記の通信において、デバイスは未知のフィールド及び値の対を捨てる。

【０４６３】

通信ごとにケースをランダムに混在させることで、フィールド名をさらに難読化できる。デバイスはこれらのフィールドを標準形式に処理する。

したがって、次の３つ通信は同一である。

・{“veRsioN”: 1, "firstName": "John", “nOnce”: 5780534, "laStnAMe": "Smith", "isAlive": true, "Age": 25}

・{“whoknows”: “698gtHGF”, “vErsion”: 1, "fiRStname": "John", "iSaLive": true, "lastName": "Smith", "age": 25}

・{"aGE": 25, "firstname": "John", "isAlive": true, "lasTName": "Smith", “whatis this”: “Jor90%hr,” “versIOn”: 1}

追加フィールドを含む可能性のあるバージョン２メッセージが送信されれば、バージョン１しか理解できないデバイスはメッセージを拒否したり、下位の互換性（ｂａｃｋｗａｒｄｓ　ｃｏｍｐａｔｉｂｉｌｉｔｙ）が保障される場合は、理解したフィールドを処理して残りは捨てる。どのジョンが一部のフィールドと下位互換されるかを示すフィールドを提供することで、これらをより向上させ得る。

【０４６４】

これにより、攻撃に対する脆弱性を徹底に除去する。メッセージの構造も保持さできるが、可変長フィールドを使用する。これもまた、同じ結果を達成する。また、ＨＭＡＣを使用することで、メッセージの無欠性と機密性の両方が保護される。エンド組織のコアシステムが構造化された形式のメッセージを必要とする場合、Ｔｅｒｅｏｎは、メッセージがサーバに達すればメッセージを再構成し、組織のコアシステムから要求される形式にメッセージを再フォーマットする。拡張可能なフレームワークは、レガシーシステムのセキュリティー問題を克服することを可能にし、依然としてこのようなシステムで動作することを可能にする。

【０４６５】

拡張可能なフレームワークは上記と同じレベルのセキュリティー及び柔軟性で、あらゆるデータ又はレコードタイプを支援する。

抽象化されたワークフローコンポーネント（Ａｂｓｔｒａｃｔｅｄ　ｗｏｒｋｆｌｏｗ　ｃｏｍｐｏｎｅｎｔｓ）

既存ソリューションでは、支払いプロセスはソフトウェアで定義され、具現され、テストされ、そして、リリースされる。その支払いトランザクション構造は固定され、デバイス、端末、及びサーバをリコール及び交換したり再プログラムするための著しい努力なしには変えることができない。

【０４６６】

Ｔｅｒｅｏｎはこれを行わない。代わりに、これは個々が接続されたコンポーネントと相互作用する個別コンポーネントから支払いプロセスを構成する。このようなコンポーネ

ントは、本質的にプロセスのワークフローを配置する。各コンポーネントは、アップデートされ、支払いプロセスそのものに影響を与えることなく機能を追加することができる。これは、デバイスからプロセスコンポーネントを抽象化するため、一度定義されたトランザクションがカードやカード端末、モバイル、又は、ウェブポータルのような複数のデバイスに適用されてもよい。

【０４６７】

各コンポーネントは、受信した命令の結果に応じて命令及び情報を次のコンポーネントに伝達する。命令は、トランザクション式（ｔｒａｎｓａｃｔｉｏｎａｌ）でも次のコンポーネントが作動する方法などの制御を含むことができる（例えば、選択事項である場合にＰＩＮの要求、選択の集合を提供、特定メッセージの表示、及び予想される応答又は許可される応答）。

【０４６８】

これにより、既存のエンドポイントを再プログラミングしたり置き換えする必要がなく、既存の支払いサービスを変更して新しいサービスを構築することができる。現在、支払いサービスの提供者が支払いシステムを実現すると、支払いサービス提供者は、エンドポイントを交換せずシステムを容易に変更することができない。既存のシステムは基本的に静的である。これは、それらを動的システムに置き換える。

【０４６９】

拡張可能なフレームワークにより、運営者がこのようなコンポーネントを用いて特定のトランザクションに対するワークフローを計画することができる。それは、意思決定ツリーなどを含むワークフローを実現する。運営者は、既存のコンポーネントを再配列したり、新しい機能を提供する新しいコンポーネントを追加したり、コンポーネントを除去することによって既存のワークフローを修正する。既存のシステムでこれを行うために、サーバ及び端末を再プログラムし、カード自体を交換させる必要がある。

【０４７０】

これの例が図１８～図２０に図示されている。各コンポーネントが何をしているかを視覚化しやすくするために、コンポーネント自体は端末画面によってブロックとして表されています。しかし、これらのコンポーネントは、モバイルトランザクション、ウェブポータルトランザクション、及びカード端末トランザクションに同じく適用される。既存のワークフローを変更するために、コンポーネントの順と接続は簡単に変更される。新しいワークフローを作るためには、必要なコンポーネントを単に所望する順に接続する。

【０４７１】

通常の支払いプロセスは、非接触式、連絡先、及びモバイルの支払いに対して別途の支払いプロセスを生成する。コンポーネント１８０４は、図１８に示すような「定時に完全なトランザクション（ｃｏｍｐｌｅｔｅ　ｔｒａｎｓａｃｔｉｏｎ　ｉｎ　ｔｉｍｅ）」コンポーネント１８０２直後にチェーンの左側に一般的に示される。

【０４７２】

しかし、図１９に示すように、このコンポーネントを右側に沿ってさらに移動させることで、２つの異なる決定コンポーネント１９０２及び１９０４をチェーンに挿入すると、運営者は、単一支払いプロセスで接触、非接触、及びモバイル支払いを管理できる単一支払いプロセスを生成し得る。

【０４７３】

運営者は、さらに進むことができる。システムが顧客を識別した後プロセスに特別な季節の提案（ｓｅａｓｏｎａｌ　ｏｆｆｅｒ）を追加しようとする。図２０に示すように、それはいつでもコンポーネント１８０４をさらに右側に移動させ、マーチャントが金額及びＰＩＮを入力しなければならない前に、自動的に顧客へ提案を提供する新しいコンポーネント２００２を本来の位置に挿入する。運営者は、例えば、そのコンポーネントをクリスマスまでの２４日間で動作するように設定し、それ以降は新年までの間は別のコンポーネントを提供することができる。運営者がリコール、再プログラム、及びデバイスを必要とせず、これはクリスマス及び新年シーズンの支払いプロセスを動的に変更し得る。コン

ポーネントは、顧客に提案を表示するようモバイル又はカード端末であるディスプレイデバイスに指示するだけでよい。運営者は、ＰＩＮ要求事項を不活性化するコンポーネント１８０４を構成することで、ＰＩＮ要求事項を容易に不活性化し得る。同様に、コンポーネントがＰＩＮを要求する機能を有していない場合、運営者は機能を含むように該当コンポーネントをアップデートし得る。

【０４７４】

運営者はさらに進んで、顧客が必要に応じて様々な提案の中から選択できるように全体的な意思決定ツリーを構築し得る。提案シーズンが終了すると、運営者は、新しいコンポーネントを除去するだけで、プロセスは本来の構造を再び開始する。

【０４７５】

重要なことは、運営者がいつでもプロセスを変更するためにデバイスをリコールする昼用がないことにある。バックエンド（ｂａｃｋ　ｅｎｄ）でプロセスを再構成してから、選択した日時に変更事項を実現する。

【０４７６】

Ｔｅｒｅｏｎサーバの内部管理及び運営を提供するフレームワークは、正確に同じ方式で構成できる。ここで、フレームワークコンポーネントは、ユーザと管理者がアクセスできる方法及び情報、実行できるタスクを管理するアクセスコンテキストと相互作用する。

【０４７７】

動的サービス（Ｄｙｎａｍｉｃ　ｓｅｒｖｉｃｅｓ）

拡張可能なフレームワークにより、組織は新しいサービスを迅速に作動及び実現可能にする。運営者は、必要なブロックを互いにリンクし、関連メッセージを定義することで、このようなサービスを簡単に定義する。サービスコードを作成するためにプログラマーを雇用する必要がなく、フレームワークはマーケティング及びＩＴ部門がワークフローを定義する定義ファイルを作成したり、「ワークフローを描く」ためのグラフィックシステム（ｇｒａｐｈｉｃａｌ　ｓｙｓｔｅｍ）を使用したり、又は、プロセスを定義する異なるワークフローによりサービスを実現することができる。ワークフローを確認した後、運営者は定義されたステップ又はブロックをともにリンクすることによってワークフローを簡単に実現し、Ｔｅｒｅｏｎは、全ての資格のあるユーザがサービスを利用できるようにする。

【０４７８】

例えば、運営者は、ブロックを用いて任意の値の支払いを受け取ることができ、その後のブロックでＰＩＮを要求する必要がある。しかし、運営者がアクセス制御システムを提供しようとする場合、同じ運営者は、別のセットのルームにアクセスできるＰＩＮを要求するためのブロックを使用する一方で、ワンセットのルームにＰＩＮなしでアクセスを許容するブロックを生成する。

【０４７９】

これは、既存のシステムとは異なり、組織がトランザクション処理システムを始めた後でもユーザに発行されたデバイスを交換する必要なく、組織が新しいサービスを設計及び実現したり、既存のサービスを修正又は除去できることを意味する。デバイスが定義されたステップを理解してそれを作動できる場合、該当デバイスは、組織が定義した任意のサービスを該当ステップを使用して提供し得る。組織がサービスを定義すると、システムは、当該サービスを対象ユーザ又はユーザに直ちに利用可能にする。

【０４８０】

抽象化されたデバイス（Ａｂｓｔｒａｃｔｅｄ　ｄｅｖｉｃｅｓ）

拡張可能なフレームワークは、抽象化の原理を取り入れ、デバイスそのものを抽象化する。フレームワークは、それらのデバイスの機能に関するデバイスの各クラスのプロセスコンポーネントを定義する。プロセスコンポーネントは、該当機能コンポーネントと相互作用する。使用可能な機能により、プロセスコンポーネントは何を出力し、何を入力するかのような作業を実行するような機能コンポーネントに指示する。

【０４８１】

粒度（Ｇｒａｎｕｌａｒｉｔｙ）
Ｔｅｒｅｏｎは、各デバイス、ユーザ、及びアカウントを個別的に識別し、ユーザがデバイスを用いてサービスにアクセスするコンテキストをアクセスできる。したがって、運営者は、個々のユーザがサービスにアクセスするコンテキストに基づいて動作（ａｃｔｉｏｎ）をトリガーするために、コンポーネント及び該当コンポーネント内のオプションを設定できる。Ｔｅｒｅｏｎは、運営者が効率よく各ユーザ、各ユーザのデバイス、及びユーザが該当デバイスを使用してサービスにアクセスするコンテキストに合わせてサービスを調整できる。
【０４８２】
例えば、あるユーザが１つのトランザクションで３つの提案のうち１つを選択することができ、他のユーザは自動的に受け取る１つの提案のみを選択することができ、３番目のユーザは提案が全く表示されない場合もある。
【０４８３】
プロセスがレコード（例えば、患者レコード）にアクセスすることに関する場合、ユーザは自分のレコードにアクセスし、ユーザが医療施設又はホームドメインのレコードにアクセスすれば、ユーザはアクセス権限を管理することができる。しかし、ユーザ（又は、他のユーザ）が該当ドメイン外部のレコードにアクセスする場合、ユーザは、該当レコードの下位集合しか表示されないか、又は該当レコードにアクセスできない（当該サービスに対するコンテキスト設定に応じて異なる）。
【０４８４】
ユーザがカード端末を用いてサービスにアクセスする場合、コンポーネントは、カード端末に関連情報を表示するように指示する。ユーザがモバイル又は他の画面デバイスを用いて同じサービスにアクセスする場合、コンポーネントは、スクリーンに関連情報を表示するように指示する。このような方式で、拡張可能なフレームワークの抽象化レイヤはデバイスに独立的である。ユーザシステム間の相互作用を制御するために適切なディスプレイ及びアクセスポイントを使用し得る。
【０４８５】
提供されるサービスにも同一に適用される。各ユーザのアカウントには、サービスの提供者デフォルトレベル（ｄｅｆａｕｌｔ　ｌｅｖｅｌ）を有する。運営者が新しいサービスを追加したり、１つ以上のユーザに対する既存サービスを修正する場合、該当ユーザのアカウントには当該サービスが存在する。サービスの核心は、提供者タグ、ユーザのアカウント番号、ユーザのデバイス登録タグの組合せである。これにより、該当ユーザ対するサービスの定義及び規則に樹枝状経路（ｄｅｎｄｒｉｔｉｃ　ｐａｔｈ）を生成する。
【０４８６】
例えば、発信者は、双方向又は自動振込み（ｉｎｔｅｒａｃｔｉｖｅ　ｏｒ　ａｕｔｏｍａｔｉｃ　ｔｒａｎｓｆｅｒ）を許容する規則を設定したモバイルを使用してもよい。受信者は、自動振込みを許容するようにデバイスを設定してもよい。この場合、発信者のデバイスは自動振込みを行うためのステップを実行するだけである。サービスタグは、振込みが双方向であるか否かに対する情報を含まない。これは発信者と受信者のサーバに格納されたサービスに関する情報に残る。
【０４８７】
受信者が双方向又は自動振込みを許容するようデバイスを設定した場合、発信者のデバイスは発信者に使用するモードを要求する。受信者が特定の時間内に自動振込みを許容するようにデバイスを設定し、それ以外の時間には双方向振込みを許容するようにデバイスを設定してもよい。ここで、受信者のＴｅｒｅｏｎサーバは、受信者の時間に応じて発信者のサーバに使用する振込みモードを通知するだけである。
【０４８８】
発信者又は受信者のデバイスが双方向振込みのみを受け入れる場合、受信者及び発信者が同時にオンライン状態であると、彼らは振込みを実行するステップを行う。受信者がカードしか持っていない場合、受信者はトランザクションのその側面（ｈｉｓ　ｓｉｄｅ）

を実行するためにマーチャントの端末に行かなければならない。受信者がオフラインである場合、発信者はそのステップを実行するが、受信者は、Ｔｅｒｅｏｎが振込みを完了する前に振込みを受諾してＰＩＮを入力するなどのようなトランザクション内のステップを行わなければならない。それまで、Ｔｅｒｅｏｎは、Ｔｅｒｅｏｎ以外のユーザに振込みを扱う同じ方式により、エスクロー機能（ｅｓｃｒｏｗ　ｆａｃｉｌｉｔｙ）で振込みを保留する。

【０４８９】

動的インタフェース（Ｄｙｎａｍｉｃ　ｉｎｔｅｒｆａｃｅｓ）

拡張可能なフレームワークは、コンテキスト依存的サービス（例えば、提案、イベントで着席できるようにユーザを助けること、マーチャント特定のプロセスなど）を誘導する。これは組織がユーザがＴｅｒｅｏｎと相互作用するとき、各ユーザが有するサービスと経験、コンテキストによりサービス可能な程度、表示されるボタン、使用可能なオプションなどをユーザが指定可能にする。

【０４９０】

各ユーザ及び各マーチャントが相互作用できるサービス数は、個々のユーザがアクセスできるサービスとマーチャントが提供できるサービス間の重複的な部分に完全に依存する。

【０４９１】

例えば、マーチャントが支払い、入金、及び引き出しを提供できて、ユーザが該当マーチャントを訪問し、該当ユーザはマーチャントの支払いにしかアクセスできない場合に、ユーザとマーチャントは支払いに関する機能、すなわち支払いと返金のみを見ることができる。ユーザが同じマーチャントを訪問した場合、該当ユーザは支払い、入金、及び引き出しにアクセスし、該当ユーザは全ての機能を見ることができる。該当マーチャントが入金及び引き出しを支援する十分な資金がない場合、フルサービスユーザが該当マーチャントを訪問したとき、ユーザは自分のデバイス又はマーチャントの端末で支払い機能のみを見ることができる。該当マーチャントは、マーチャントまで入金又は引き出しを提案するマーチャントに対する検索にもこれ以上表示されなくなる。ユーザが一部のマーチャントの特定サービスにアクセスできないが、他のマーチャントのサービスにアクセスすることはできる。フレームワークはこのような場合を処理する。

【０４９２】

動的インタフェースは、多面的なクリデンシャルの使用を補完し、デバイス及び関連アプリケーションが言及したように「サイキックペーパー（ｐｓｙｃｈｉｃ　ｐａｐｅｒ）」に類似したものにすることを可能にする。この場合、デバイスは、利用可能なサービスのみを提供し、ユーザが登録される複数のサービスに関係なく、インタフェースはそのようなサービスに合わせて調整される。あるサービスへの支払いデバイス、他のサービスに対するトランスポートチケット、他のサービスに対するドアキーなどとのように見えることができる。サービス提供者は、サービスにアクセスするために別途のデバイスを発行する必要がないことから、サービス提供及び当該サービスアップグレードの複雑性とコストを節減できる。

【０４９３】

拡張可能なフレームワークはデバイスがその外観を変えること、デバイスが使用されるコンテキストによって求められるクリデンシャル及びサービスを提供することを可能にする。例えば、ユーザが食料品店にあるような独立的なＡＴＭへアクセスするとき、ユーザの運営者の外観と感じを取るために該当ＡＴＭのスクリーンを調整し、ユーザが加入したサービスのみを提供する。

【０４９４】

他のレイヤとの相互作用（Ｉｎｔｅｒａｃｔｉｏｎ　ｗｉｔｈ　ｏｔｈｅｒ　ｌａｙｅｒｓ）

Ｔｅｒｅｏｎシステム内の他のコンポーネントと相互作用できる拡張可能なフレームワークの能力は、拡張可能なフレームワークの基本的な機能である。さらに広いセキュリテ

ィーモデルを含むコンテキストセキュリティー（ｃｏｎｔｅｘｔｕａｌｓｅｃｕｒｉｔｙ）とは別に、拡張可能なフレームワーク命令は、ハッシュチェーンを介して送信されるトランザクション情報内に埋め込むことができる（ゼロ知識証明を有するハッシュチェーンと関連して開始されたように）。

【０４９５】

オフラインモード（Ｏｆｆ－ｌｉｎｅ　ｍｏｄｅ）

Ｔｅｒｅｏｎは、３つオフラインモードを提供する。ユーザオフライン、マーチャントオフライン、及び両方オフライン。

【０４９６】

最初の２つの場合では、Ｔｅｒｅｏｎは四角の反対方向に移動してリアルタイムトランザクションを完了する。すなわち、ユーザは、マーチャント端末及びマーチャントのＴｅｒｅｏｎサーバを介して自分のＴｅｒｅｏｎサーバと通信する。マーチャントやユーザの全てはサービス低下を経験しない。Ｔｅｒｅｏｎは、関連のデバイスに対する正方形の３辺を通るセキュリティー経路を作るためにＰＡＫＥプロトコル又は類似の機能を有するプロトコルを使用する。

【０４９７】

２つのデバイスのがオフラインである３番目の場合では、即刻的な引き上げは、Ｔｅｒｅｏｎがユーザ又はマーチャントがトランザクションを支援する十分な資金を有するかどうかをリアルタイム確認できないため、Ｔｅｒｅｏｎが克服するために設計された信用リスクの露出が生じる。

【０４９８】

拡張可能なフレームワークの機能及びハッシュチェーンのバージョンを使用することで、Ｔｅｒｅｏｎはシステムが資金を続けて確認できるようにする。ユーザとマーチャントの両方は、自分の全ての機能を実行できる。ユーザはモバイル又はマイクロプロセッサー・カードを使用しなければならないが、ユーザやマーチャントは自分が経験するサービスの低下を見ることはない。マーチャントデバイスとユーザデバイスの両方はそれらの間のトランザクションの暗号化された細部情報と、マーチャントが作った前のオフライントランザクションのランダムサンプルを格納する。マーチャントデバイスは、ユーザのカードや電話に伝達される各トランザクションの最大のコピー数を設定する。

【０４９９】

Ｔｅｒｅｏｎはあるユーザがオフラインデバイスとオンラインデバイスを組合せて使用し、アカウント内の存在する以上の金額を引き出さないようにするため、ビジネスロジックとセキュリティーモデル及びハッシュチェーンの結合せを使用する。アカウントが信用機能を提供する場合、アカウントは、オフラインデバイスのみを支援する。オフラインロジックはクレジットを必要としないが、クレジットを提供するための許可は、サービス提供者の規制機関によって要求され得る。

【０５００】

デバイスがオフラインで作動するように許可されていない場合、オフラインのときには他のデバイスとトランザクションすることができない。デバイスの署名がオンライントランザクションを支援するものとして識別するため、セキュリティー及び認証モデルはそうすることを防止し、デバイスは、登録されたアカウントの価値にも影響を与える、どのトランザクションも処理できない。

【０５０１】

デバイスがオフライントランザクションを支援できる場合、サービス提供者は、これをオフライン許容量（ｏｆｆ－ｌｉｎｅ　ａｌｌｏｗａｎｃｅ）の一定の金額に制限する（デバイスがオンラインであるとき常にアップデートされるクレジットの限度、又はアカウントの残高の一部）。デバイスは、アカウントから合計額又は該当オフライン許容量での資金の振込み又は支払いのみを承認する。もちろん、サービス提供者は、デバイスが振込み又は資金を受容するよう権限を付与することができ、このような受容価値（オフライン受容許容量）を制限し得る。第１デバイスがオフライン間ユーザがアカウントにアクセス

すると、ポータルを介して直接又は他のオンラインデバイスへ、ユーザがアカウントの残高からオフライン許容量を差し引いた金額までのみアカウント振込み又は支払いを承認する。

【０５０２】

Ｔｅｒｅｏｎは、関連レコードの含まれたデバイスの１つがオンラインになると、全てのオフライントランザクションを調整する。当然一部のトランザクションのコピーを受け取ることになるが、これを用いて以前の調整を確認することができる。

【０５０３】

したがって、サーバがオフラインデバイスへの振込み又は支払いに関するオフライントランザクションの第３者サーバ（ｔｈｉｒｄ－ｐａｒｔｙ　ｓｅｒｖｅｒｓ）からレコードを受信すると、それは該当トランザクションのコピーを十分に受信すれば該当トランザクションを処理し、その資金をアカウントの残高に追加する。同様に、サーバがオフラインデバイスからの支払い又は振込みに関するオフライントランザクションの第三者サーバからレコードを受信すると、それは該当トランザクションの十分なコピーを受信すれば該当トランザクションを処理し、アカウントの残高と残りのオフライン許容量からそれらの金額を差し引く。

【０５０４】

上記の図では支払いに関するものが示され、これらは視角化が容易であるため、同じ動作モードは全てのタイプのトランザクションシステムに適用されてもよい。１つ例として、ＩｏＴデバイス又は他の産業コンポーネント間の相互作用である。再配置、挿入、又は、除去可能なモジュールを含むワークフローを生成することによって、運営者などは再プログラム及び再インストールする必要なしに新しい方式で作動するようにデバイスを再構成できる。

【０５０５】

運営者は、現場でデバイスの用途を変更したり、作動方式を変更したり、デバイスが異なるデバイスを制御して該当デバイスが作動する環境で検出した変更事項によってワークフローを修正したりすることができる。

【０５０６】

また、ＩｏＴデバイスは、必要に応じて、ワークフローを構成するモジュールのアセンブリを修正して互いのワークフローを修正してもよい。ルックアップサービスは、デバイスが互いを識別し認証することを可能にする間に、デバイス間通信を管理するセキュリティーモデルは、その通信を中間者攻撃に対して遮断する。

【０５０７】

オフラインモードは、このようなデバイスが自律的又は半自動的に作動して互いに相互運用され、該当デバイス間の全てのトランザクションを確認及び検証し、必要に応じて運営者のシステムと相互作用できるようにする。

【０５０８】

以下説明されたコンテキストセキュリティーモデルは、ＩＯＴデバイスのような全てのタイプのデバイスまで拡張される。デバイスが作動することを許可される限り、該当デバイスのサービスが関連ルックアップサービスにリストされている限り、任意のデバイスや他のデバイスと通信し、それぞれはデバイス間トランザクション及びデータ通信を信頼して有効にするためにハッシュチェーンを使用し、それぞれはデバイスのワークフローを修正したりデバイスのシステムをアップグレードしたり、該当システムの間にデータを単に伝達又は対照するための命令を含む。各デバイスは、トランザクションに対する完全な監査を保持する。

【０５０９】

セキュリティー（Ｓｅｃｕｒｉｔｙ）

Ｔｅｒｅｏｎシステムは、レガシートランザクション処理システムに用いられる現在のセキュリティーモデル及びプロトコルに存在する欠陥及び制限事項を克服する複数の固有セキュリティーモデルを使用する。例えば、セキュリティーモデルは、デバイスにデータ

を格納する必要性がなくなる。これは既存システムの主なイシューである。
【０５１０】
ＵＳＳＤのセキュリティー（Ｓｅｃｕｒｉｎｇ ＵＳＳＤ）
ＵＳＳＤ（ｕｎｓｔｒｕｃｔｕｒｅｄ ｓｕｐｐｌｅｍｅｎｔａｒｙ ｓｅｒｖｉｃｅ ｄａｔａ）は、フィーチャーフォンとの支払いをはじめとする様々なトランザクションタイプに対する通信チャネルとして一般的に用いられる。ＴｅｒｅｏｎはＵＳＳＤを安全に使用可能にする。
【０５１１】
多くの実現では、ユーザがＵＳＳＤコードを入力するか、番号の決められたメニューから動作（ａｃｔｉｏｎ）を選択する必要がある。暗号化されていない一連のメッセージは前後に移動する。これは、コスト、セキュリティーの低下、及びユーザ経験不足のイシューを発生させる。
【０５１２】
セキュリティーの問題が発生する７ビット又は８ビットテキストでメッセージを送信する代わりに、Ｔｅｒｅｏｎは、新しい方式でＵＳＳＤ及び類似の通信チャネルを使用する。Ｔｅｒｅｏｎは、単にそれをセッション基盤の短いバースト通信チャネル（ｓｅｓｓｉｏｎ－ｂａｓｅｄ ｓｈｏｒｔ－ｂｕｒｓｔ ｃｏｍｍｕｎｉｃａｔｉｏｎｓ ｃｈａｎｎｅｌ）と見なす。
【０５１３】
Ｔｅｒｅｏｎは、ＵＳＳＤに合わせてメッセージを調整しない。これんは既存のシステムの動作である。代わりに、トランザクションセッション内の各暗号化された通信に対して、Ｔｅｒｅｏｎは、サイファーテキストを生成するためにＴＣＰ／ＩＰ（ＧＰＲＳ、３Ｇ、４Ｇ、ＷｉＦｉなど）を通した通信と同様に通信を暗号化し、サイファーテキストを基本６４　７ビット文字列に符号化する。その次に、Ｔｅｒｅｏｎは、サイファーテキストの長さを確認する。ＵＳＳＤメッセージの許された空間よりも長い場合、サイファーテキストを２つ以上の部分に分割し、ＵＳＳＤを用いてこれらを個別的に送信する。反対側の端では、Ｔｅｒｅｏｎは、部分を全体の文字列に再調合し、これをサイファーテキストに変換して解読する。
【０５１４】
Ｔｅｒｅｏｎはこの方法を用いて、当事者を識別して認証するためにまずＴＬＳ（ｔｒａｎｓｐｏｒｔ ｌａｙｅｒｓｅｃｕｒｉｔｙ）を使用する。これで第１セッションキーを生成する。その次に、Ｔｅｒｅｏｎは、当事者がセッション内の以降の全ての通信を暗号化するために使用する第２セッションキーを生成するＰＡＫＥプロトコルネゴシエーションを暗号化するために、このセッションキーを用いてもよい。
【０５１５】
一部のフィーチャーフォンは、ＷＡＰ（ｗｉｒｅｌｅｓｓ ａｐｐｌｉｃａｔｉｏｎ ｐｒｏｔｏｃｏｌ）を支援する。このような実現がＵＳＳＤ上でＷＡＰを使用する場合、ＴｅｒｅｏｎはＵＳＳＤを介して通信する方法としてＷＡＰプロトコルスタックを使用する。これは、単に追加認証レベルとして機能するＷＴＬＳ（ｗｉｒｅｌｅｓｓ ｔｒａｎｓｐｏｒｔ ｌａｙｅｒｓｅｃｕｒｉｔｙ）レイヤを提供する（これは、Ｔｅｒｅｏｎが基本的に使用するＴＬＳ及び高級暗号化標準２５６（ＡＥＳ２５６））よりも弱いため、Ｔｅｒｅｏｎは全てのイベントに通信を暗号化するためにＡＥＳ２５６を使用する）。
【０５１６】
これはまた、Ｔｅｒｅｏｎがセキュリティーが不足している他の通信チャネル（例えば、ＮＦＣ、ブルートゥースなど）を確保する方法でもある。メッセージングセッションを慎重に構成することにより、ＵＳＳＤ及びその他の「セキュアでない」チャネルの性質は完全に変更されることができる。
【０５１７】
能動デバイス（及びＩｏＴ）のセキュリティーモデル（Ｓｅｃｕｒｉｔｙ ｍｏｄｅｌ ｆｏｒ ａｃｔｉｖｅ ｄｅｖｉｃｅｓ（ａｎｄ ｔｈｅ Ｉｎｔｅｒｎｅｔ ｏｆ

Ｔｈｉｎｇｓ））
モバイル、カード端末などのような能動デバイスのセキュリティーモデルは、カードに対するセキュリティーモデルと同じ方式で動作する（下記参照）。セキュリティーアルゴリズムがこの前にクラックされたため、ＳＩＭは使用されない。代わりに、デバイスに暗号化されて格納される登録キーは、ネットワークが生成する固有のキーと共に用いられる。モバイルデバイスで、Ｔｅｒｅｏｎは該当キーを用いて検索し、モバイルによって報告されるＩＭＳＩ（ｉｎｔｅｒｎａｔｉｏｎａｌ ｍｏｂｉｌｅ ｓｕｂｓｃｒｉｂｅｒ ｉｄｅｎｔｉｔｙ）が本物であるかを確認できる。

【０５１８】
ユーザが最初にアプリケーションを実行すれば（ユーザが所望する場合、複数のアプリケーションを有してもよい）、アプリケーションはＴｅｒｅｏｎサーバがデバイスのモバイル番号又はシリアル番号と共にユーザのアカウントに対して生成する一回だけの性質認証コード（ｏｎｅ－ｔｉｍｅ ａｕｔｈｅｎｔｉｃａｔｉｏｎ ｃｏｄｅ）を要求する（アプリケーションが該当番号を最初に確認できない場合）。ユーザは、複数のＴｅｒｅｏｎサーバに自分のアプリケーションを登録してもよい。ここで、各サーバは、サーバがユーザに対して作動する各アカウント又はサービスに対して固有な一回だけの性質活性化コード（ｏｎｅ－ｔｉｍｅ ａｃｔｉｖａｔｉｏｎ ｃｏｄｅ）を生成する。

【０５１９】
ユーザが一回だけの活性化コードを入力すると、アプリケーションは、第１ＰＡＫＥセッションを生成するために該当コードをサーバとの間の共有秘密（ｓｈａｒｅｄｓｅｃｒｅｔ）として使用する（必要に応じて、アプリケーションとＴｅｒｅｏｎサーバがＴＬＳ又は類似のプロトコルを用いて互いに有効検査した後）。それが第１ＰＡＫＥセッションを確立すると、Ｔｅｒｅｏｎサーバは、暗号化されて署名された登録キーを新しい共有秘密と共にアプリケーションに送信する。サーバとアプリケーションの両方は、一回だけの活性化コード、登録キー、及び共有秘密のハッシュを生成することにより、新しい共有秘密を生成するために一回だけの活性化コード、登録キー、及び共有秘密を使用する。

【０５２０】
サーバとアプリケーションが通信するたびに、それらは、オンライン通信でそれらの間で通信した以前のメッセージのハッシュで以前の共有秘密をハッシュして共有秘密を生成する。アプリケーションとサーバが互いに通信するたびに、それらは以前の交換のハッシュと交換したトランザクションのコンテンツのハッシュ（トランザクションハッシュ）を生成する。どちらもこのトランザクションハッシュを使用して新しい共有秘密を生成する。

【０５２１】
ユーザがデバイスを紛失したり、アプリケーションを再び登録したり、デバイスを変更しなければならない場合、Ｔｅｒｅｏｎサーバは、新しい一回だけの認証コードと登録キーを生成する。サーバがアプリケーションに伝達する新しい共有秘密は、サーバとアプリケーションの間に交換された以前のメッセージのハッシュから生成される。

【０５２２】
このキー伝達により、アプリケーションとＴｅｒｅｏｎサーバが各ＰＡＫＥセッションに対して新しい共有秘密を有することができる。したがって、攻撃者がＴＬＳセッションを切断できる場合（サーバとアプリケーションがメッセージに署名をする時非常に難しい）、攻撃者は、依然としてＰＡＫＥセッションキーを切断する必要がある。当事者が特徴的な事項（ｆｅａｔ）を管理したのであれば、それは当事者にセッションに対するキーが与えられるのであろう。各通信に対して新しいキーを生成するプロセスは、当事者が各通信に対して特徴的な事項（ｆｅａｔ）を繰り返さなければならないことを意味し、これは事実上算出不可能である。

【０５２３】
アプリケーションは全てのセッションで特定のサービスに対して認証されるため、ユーザのアプリケーションは当該サービスとのみ相互作用する。サーバは、ユーザのアプリケ

ーションが登録された他のサービスについて知らない。事実上、アプリケーションは、ユーザが登録されることができる複数のサービスとは関係なく、「サイキックペーパー」と類似なもの、サービスに要求されるクレデンシャルのみを提供する識別デバイスとなる。それは、あるサービスへの支払いデバイス、他のサービスへの移送チケット、他のサービスへのドアキーなどのように見える。サービス提供者は、サービスにアクセスするために別途のデバイスを発行する必要がないことから、サービス提供及び当該サービスアップグレードの複雑性とコストを節減できる。

【０５２４】

セキュリティーモデルは、追加の利点を有する。ユーザが自分のデバイスを失う場合、ユーザはまったく同じ番号の新しいデバイスを取得し得る。アプリケーションがある古いデバイスは作動しないが、新しいデバイスが一度登録されれば有効な秘密キーと登録コードを有するので作動できる。紛失のデバイスを報告するまでには時間がかかるが、必要なパスワードとＰＩＮ又は他の認証トークンを有しないために、誰もトランザクションを行うことができない。

【０５２５】

ユーザがアプリケーションにアクセスする前に、ユーザ又はＴｅｒｅｏｎシステム管理者は、暗号を要求するようにアプリケーションを構成してもよい。このパスワードは、Ｔｅｒｅｏｎサーバと共に点検される。有効な場合、Ｔｅｒｅｏｎサーバはアプリケーションに（常に署名され暗号化された通信で）作動するよう指示する。パスワードが無効な場合、Ｔｅｒｅｏｎサーバは、アプリケーションに制限された回数の試みで（ｌｉｍｉｔｅｄ ｎｕｍｂｅｒ ｏｆ ａｔｔｅｍｐｔｓ）新しいパスワードを要求するように指示する。その後、Ｔｅｒｅｏｎサーバはユーザのアプリケーションをかけて（ｌｏｃｋ ｏｕｔ）、ユーザは、アプリケーションのロックを解除し、デバイスを再登録するために、管理者とコンタクトする必要がある。

【０５２６】

各クリデンシャルは時間が計られている。すなわち、あるユーザが定義された期間中に特定のクリデンシャルが割り当てられ、その期間中に該当クリデンシャルで発生する全てのトランザクションはそのユーザにリンクされることを意味する。そのユーザがクリデンシャルを変更すると、元のクリデンシャルは他のユーザに割り当てられる。しかし、ルックアップサーバは、クリデンシャルとそのクリデンシャルに登録された期間の組合せに基づいて、トランザクションとクリデンシャルをリンクし続ける。

【０５２７】

同じモデルを「ＩｏＴ」内のデバイス間の通信を保護するために採択することができる。ここで、認証書又はハードウェアに内蔵されたシリアル番号を用いて各デバイスを識別することができる。これは、トランザクション日付又はデバイス間で送信された以前のメッセージとハッシュされるとき、それは各デバイスが最初のコンタクトでスワップ（ｓｗａｐ）する第１共有秘密になる。２つの番号は、デバイスを識別できる公開シリアル番号、ＰＫＩ（ｐｕｂｌｉｃ ｋｅｙ ｉｎｆｒａｓｔｒｕｃｔｕｒｅ）認証書の代わりに機能する役割、及び共有秘密として作用する暗号で保護されたシリアル番号に使用されてもよい。あるいは、単一シリアル番号をＩＤと第１共有秘密として使用し、新しい秘密キーをセキュリティー通信チャネルを介してアップロードしてもよい（システムアーキテクチャーの通信レイヤに対する説明を参照）。

【０５２８】

Ｔｅｒｅｏｎのモバイルセキュリティーモデルは、他の利点を有する。運営者は個々のサービスに対するアクセス権限を設定し、特定の使用がそのサービスを成功させようと試みるデバイス及びネットワークによりアクセスレベルを構成するためにこれを使用する。例えば、提供者は、管理者がモバイルデバイスでなく固定されたデバイスを介してセキュリティー共用ネットワークを介してシステムログを見て、インターネットネットワークを介してシステム管理機能にのみアクセスできるように指定してもよい。

【０５２９】

この機能は、支払いに一部のアプリケーションを有するが（定義されたネットワーク及びデバイスのシステム管理機能に対するアクセスを保障）、機密性の高いコンテンツ又は権限のあるコンテンツに対する制限されたアクセスが必要な他のサービスに対して提供されるため、ユーザは、特定データ、これを見ることができる人、このような第三者が閲覧できるデータ、及び閲覧できる場所を正確に制御できる。

【０５３０】

セキュリティーモデルにより、組織はあらゆるデバイスによって収集、生成、又は送信される全てのデータの個人情報及びセキュリティーを保証できる。これは、全てのデバイスやトランザクション、支払いから医療デバイス、交通センサ、気象センサ、水流検出器などに適用される。

【０５３１】

カードセキュリティーモデル（Ｃａｒｄｓｅｃｕｒｉｔｙ　ｍｏｄｅｌ）

ホストカードエミュレーションを使用するＥＭＶカード及びモバイルは、チップ又はモバイルのセキュリティー要素にＰＩＮを格納する。非接触式カード及びこのカードをエミュレートするモバイルも、カードの詳細の大部分を明確かつ容易に読みやすい形式で格納する。カード端末は、ユーザが入力したＰＩＮをカードに格納されているＰＩＮと照合する。これはＥＭＶシステムの多くの弱点が明らかになるようにし、ＥＭＶプロセスを十分に立証された攻撃にたいしてオープンさせる。

【０５３２】

Ｔｅｒｅｏｎは、認証キーだけをカードに格納し、Ｔｅｒｅｏｎサービスに格納された値（値が実際の値と一致しないことのみを確認する管理者に対して閉鎖されているデータベースのセキュリティー領域）と入力された値を確認する。それは、サービスと特定の機能、リソース、施設、又は、トランザクションタイプ、又は、当該サービスによって提供される他のタイプのサービスを認証する。Ｔｅｒｅｏｎは、２種類のセキュリティーモデルを使用し、その１つは他のモデルの下位集合（ｓｕｂｓｅｔ）である。

【０５３３】

多くのカードは、ＰＡＮ（長い番号）を表示する。Ｔｅｒｅｏｎは、アカウントを識別するためにこの番号を使用しない。むしろ、モバイル番号のような方式でＰＡＮを使用する。これは単にアクセスクリデンシャルである。各カードは。暗号化されたＰＡＮを有する。カードは。モバイルに登録キーが該当デバイスを認証するのと同じ方式で、カードの登録された各サービスに対して有効であることを識別する暗号化された登録キーを有する。Ｔｅｒｅｏｎサービスに登録された暗号化されたＰＡＮ文字列に関するアドレスの詳細がまだ有していない場合、暗号化されたコードは、マーチャントのＴｅｒｅｏｎサービスが要求する必要があるカントリールックアップディレクトリサービス（country look-up directory service）を示すプレフィクス（ｐｒｅｆｉｘ）を有する。

【０５３４】

ユーザがカードを端末に提示するとき、端末は暗号化されたＰＡＮを読み出し、暗号化された登録キーを使用してカードの登録された端末でカードの有効性を検査する。ユーザのＴｅｒｅｏｎサービスがカードとマーチャントのＴｅｒｅｏｎサービスを全て確認及び認証すれば、ユーザサービスは、マーチャントのＴｅｒｅｏｎサービスにＰＡＮを暗号化されない形式で送信し、ここで、それは暗号化された形式でこれをキャッシュに登録してもよい。したがって、ユーザが後ほどＥコマースポータル又はマーチャントの端末を介してＰＡＮを暗号化無しで（in the clear）入力すると、サービスは、連絡する他のサービスを知るようになる。

【０５３５】

カード読み出し機器が何らかの理由でもカードを読み出すことができなければ、ユーザ又はマーチャントはＰＡＮを入力してマーチャントのＴｅｒｅｏｎサービスはユーザのＴｅｒｅｏｎサービスのアドレスを取得するためにこのＰＡＮを使用する。カードのＰＡＮは、ユーザが使用できる多くのクリデンシャルの１つだけである。

【０５３６】

マーチャントのＴｅｒｅｏｎサービスがカードを認証すると、マーチャントの端末は、ハッシュされたキーを用いてＴＬＳを設定し、次にハッシュされたキーを用いてＰＡＫＥセッションをＴｅｒｅｏｎサービスとして設定する（端末がそのサービスと通信するごとに以前のキーを登録キーとしてハッシュしてＰＡＫＥセッションに対する新しい共有秘密を生成する）。マーチャントの端末がＰＩＮを要求するまで、マーチャントプロセスは続く（支払いサービス提供者によって決定され、Ｔｅｒｅｏｎサービスのビジネス規則エンジンに明示されたように、ユーザが該当トランザクションにＰＩＮを必要とする場合）。ユーザのＴｅｒｅｏｎサービスは、マーチャントのサービスとＰＡＫＥセッションを生成し、次に一回だけのキーをマーチャントのサービスに送信し、ＴＬＳを最初に使用して生成された他のＰＡＫＥセッションを介して暗号化されたメッセージを端末に送信する。

【０５３７】

マーチャント端末はキーを受信し、ユーザによって選択されたテキスト（端末がマーチャントのサービスによって許可されることを示す）を表示するためにメッセージを解読する。ユーザは、端末のＰＡＫＥセッションを介してユーザのサービスと通信される自分のＰＩＮを入力する。このプロセスは、ユーザが自分のＰＩＮをマーチャント端末に入力しなければならない場合にのみ発生する。これは、セキュリティーアプリ（マーチャントの端末がユーザのＴｅｒｅｏｎサービスからアクセスし、ユーザのサービスが安定の署名されたキー交換で端末に送信する第２のワンタイムキー（second one-time key）に暗号化される）に入力されるため、マーチャントの端末は、ＰＩＮを明確に見ることができない。全ての通信は一般的にマーチャントのサービスによって行われ、端末とユーザＴｅｒｅｏｎサービス間の直接通信は端末がその機能を支援できる場所で設立される。

【０５３８】

カードがマイクロープロセッサカード（Ｃｈｉｐ＆ＰＩＮ、非接触式、又は２種類両方）の場合、カードは発行時に最初生成された共有秘密を有してもよい。

マイクロープロセッサカードは、登録されたＴｅｒｅｏｎサービス（又はサービス用サービス）とセッションを確立するためにＰＡＫＥを使用する。このセッションは、Ｔｅｒｅｏｎサービスのあるカード端末（モバイルタブレット又はＰｏＳカード端末であり得る）によって確立されたセッションと共に行われる。これは既存の端末及びＣｈｉｐ＆ＰＩＮカードが示す主な脆弱性（複数の「中間者」又は「ウェッジ（ｗｅｄｇｅ）」攻撃を介してＰＩＮ検証プロセスを妨害して破壊する既存のインフラストラクチャーの脆弱性）を即座に除去する。

【０５３９】

カードは、サービスに送信するキー（サービスがＰＩＮを暗号化するためにマーチャント端末に送信）を生成するためにこのチャネルを使用する。カードが最後のオンライントランザクションの残高、オフライントランザクションに対して使用する一連のキーを生成するためのシードとしてで使用するキー、及び第三者オフライントランザクションのレコードを格納するとき、それはオフライントランザクションを容易にするためにこのチャネルを使用する。

【０５４０】

カードの紛失又は盗難された場合、Ｔｅｒｅｏｎのセキュリティーモデルは、発行者が新しいＰＡＮを発行する必要がないことを意味する。

コンテキスト基盤のセキュリティー（Ｃｏｎｔｅｘｔ　ｂａｓｅｄｓｅｃｕｒｉｔｙ）

多くのセキュリティープロトコルは、いくつかのクリデンシャルを使用し、基本的な前提を基盤とする。この仮定が、エラー及びセキュリティの低下を招く可能性がある。Ｔｅｒｅｏｎシステムは、このシステムがないと通信ネットワークが安全ではなく信頼できないという仮定、及びデバイスが動作する環境も安全でないという仮定以外の根本的な仮定には依存しない。

【０５４１】

Ｔｅｒｅｏｎシステムは様々な段階を経て、クリデンシャルセットとこのクリデンシャルが提示されるコンテキストを全て調べる。これは追加的なセキュリティーを提供し、組

織が従業員又は構成員が一部又は全ての状況で自分のデバイス（ＢＹＯＤともいう）を使用可能にする手段１つを確保する。

【０５４２】

Ｔｅｒｅｏｎは、ユーザのパスワード、ＰＩＮ、又は、その他の直接認証クリデンシャルだけではなく、デバイスの詳細、該当デバイスのアプリケーション、該当デバイスがＴｅｒｅｏｎにアクセスするネットワーク、セッション時間にこのデバイスの地理的な位置、及びユーザがこのデバイスにアクセスしているサービス又は情報を使用する。

【０５４３】

Ｔｅｒｅｏｎはクリデンシャルを受け取り、該当クリデンシャルと設定されたコンテキストに基づいて、クリデンシャルに適切なアクセスレベルを付与する情報のアクセスを制御する。

【０５４４】

例えば、Ｔｅｒｅｏｎにより承認されない個人デバイスの深層管理サービスにアクセスしようとする管理者は、この管理者が職場と会社のネットワークにあるかどうかに関わらず、当該サービスから遮断される。しかし、同じ管理者は同じデバイスのシステムログの一部を見る権限がある。

【０５４５】

第２の例は、コンテキストセキュリティーモデルがセカンダリーユーザが見ることのできるサービスを管理する場合である。ユーザは設定限度（信用限度又は使用可能な最大金額まで）なしに入金、引き出し、及び支払いのような様々な機能を提供するモバイル又はカードを保有している。そのユーザは、何回もカフェを訪問し、いつもコーヒーとアーモンドクロワッサンを購入した。現在、ユーザは自分のカードを息子に渡し、合計４０ポンドをそのカードの上限として設定した。ユーザは、コーヒーを買うために同じカフェにカードを持っていく息子の使用のために第２ＰＩＮを設定した。彼は過去６個をすでに購入したため、Ｔｅｒｅｏｎシステムは、一般的に無料アーモンドクロワッサンをユーザに提供し、カフェはその提案を顧客に伝達するためにＴｅｒｅｏｎを使用する。しかし、ユーザの息子がＰＩＮを入力するとき、Ｔｅｒｅｏｎシステムは支払っているのがユーザの息子であること（自分の父のＰＩＮを知らない）を検出し、彼がピーナッツアレルギーがあるため、その父が息子のＰＩＮを息子のプロファイルと接続したことから、今日の提案は遮断される。マーチャントは、無料クロワッサン提供に対する通知を見ることができず、Ｔｅｒｅｏｎはユーザの息子がナッツ類を食べることができないことを知っている。マーチャントが見ることができるのはコーヒーの支払いである。

【０５４６】

ユーザは、その息子が１０ポンドまで現金を引き出すことを許容したが、資金を入金することを許容していない。したがって、ユーザの息子は最大１０ポンドの引き出しを提供できるマーチャントに入ると、彼はマーチャントのオプションを見ることができる。

【０５４７】

コンテキスト基盤のセキュリティーは、アクセス制御よりも優れている。ユーザがデバイスを提示したり使用するコンテキストに応じて、該当デバイスは、該当コンテキストに必要なクリデンシャルのみを提供する。それが「サイキックペーパー」となる。このような方式で、ディレクトリサービス２１６は、コンテキスト基盤のセキュリティーを支援できる機能を提供する。

【０５４８】

コンテキスト基盤のセキュリティーでは、特定のコンテキストに対する別のクリデンシャル及びデバイスが不要になる。これで、単一のデバイスが図書館の図書館カードクリデンシャル、バスや汽車の交通チケット、部屋や施設にアクセスするためのセキュリティーキー、会社の食堂の社内支払いデバイス、劇場のチケット、スーパーマーケット内の標準支払いデバイス、運転免許証、ＮＨＳカード、サービスへの資格を証明するＩＤカード（サービスが必要であればマーチャントのデバイスに写真付きのＩＤを提示できる）などである。

【０５４９】
Ｔｅｒｅｏｎは、動的でリアルタイムトランザクション処理及び支払いを提供するために、管理者又はユーザは許可されたコンテキストやクリデンシャルをリアルタイムで修正、追加、又は取り消すことができる。修正は、サービスを提供するＴｅｒｅｏｎサーバ、又は、ルックアップディレクトリサービス２１６、又は両方に直ちに反映される。現在のシステムがデバイスを不活性化するまで、紛失したデバイスはこれ以上金銭的又はＩＤの露出期限といったリスクをもたらす必要がない。ユーザ又は、管理者がクリデンシャル又はコンテキストを取り消し又は修正すると、変更事項はすぐに活性化される。
【０５５０】
ワンタッチトランザクション（Ｏｎｅ　ｔｏｕｃｈ　ｔｒａｎｓａｃｔｉｏｎ）
Ｔｅｒｅｏｎは、既存システムのセキュリティーの欠陥を除去するワンボタントランザクション権限付与及びアクセス方法（ｏｎｅ－ｂｕｔｔｏｎ　ｔｒａｎｓａｃｔｉｏｎ　ａｕｔｈｏｒｉｓａｔｉｏｎ　ａｎｄ　ａｃｃｅｓｓ　ｍｅｔｈｏｄ）を実現する。例えば、現在のＰＩＮなし、又はＮＰＣの支払いは、支払いに対する認証を提供しないことから極めて危険である。カード発行者が非接触式ＥＭＶシステムでモバイル又はカードクリデンシャルを取り消すまで、ユーザは全ての支払いに対して責任を負う。デバイスが発行者によって取り消されても、消費者は依然として支払いを活性化していないことを証明しなければならない。支払いが認証のためにＰＩＮを必要としない場合、どのようにすればよいか。これは誰かが非接触式カードやモバイルを手に取り、単にタップして支払うことを可能にする大きな穴を残す。デバイスが取り消すまで、デバイスは続けて有効である。
【０５５１】
Ｔｅｒｅｏｎは、３つのモードのうち１つでタップアンドゴー（ｔａｐ－ａｎｄ－ｇｏ）を支援し、それぞれのモードは運営のためにコンテキストにより異なる。これらの１つは、個人を識別するアクセス方式を使用するワンタッチトランザクションを提供する。ユーザとサービス提供者が提供される認証レベルが満足である点に同意すると、システムはワンタッチ認証方法を提供し、デバイスが大きいボタンを表示したりユーザがタッチできるように画面に広い領域を構成する。他のモードは、ユーザがクリデンシャルを入力しない従来の非接触式トランザクションとデバイスが互いに識別した後、ユーザが標準支払いクリデンシャルを入力することのような完全に非接触式モードである。
【０５５２】
ボタン又は領域そのものは、タッチスクリーンを介して認証を提供する。全ての個人は、各自が押す場所と使用する圧力パターンの観点から、全て独特の方式で画面を押す。個人がこの機能を使用しようとする場合、Ｔｅｒｅｏｎは、該当の個人に各自の署名が押された（ｓｉｇｎａｔｕｒｅ　ｐｒｅｓｓ）ことを知るまで、ボタン又は領域を何度も押すように要求する。画面は、論理的には数個の個別セルに分割され、Ｔｅｒｅｏｎはトレーニング期間中にユーザがタッチしたセルの近接とパターンを見て、可能であれば、ユーザが押すときに発生する圧力パターンとデバイスの動きを確認する。ユーザ認証のために使用されるプロファイルを作成するために該当データを用いてモニターする。
【０５５３】
図２１は、上述した任意の１つ以上の方法を実行させるための命令セットが実行できるコンピューティングデバイス２１００の一実施のブロック図を示す。代案的な実現形態で、コンピューティングデバイスは、近距離通信網（ＬＡＮ）、イントラネット、エクストラネット又は、インターネット内の他の機械に接続（例えば、ネットワーク化される）される。コンピューティングデバイスは、クライアントサーバネットワーク環境でサーバ又はクライアント機械の容量で動作したり、ピアツーピア（又は、分散）ネットワーク環境でピアマシン（peer machine）として動作してもよい。コンピューティングデバイスは、ＰＣ、タブレットコンピュータ、セットトップボックス（ＳＴＢ）、ＰＤＡ（Ｐｅｒｓｏｎａｌ　Ｄｉｇｉｔａｌ　Ａｓｓｉｓｔａｎｔ）、セルラー電話（ｃｅｌｌｕｌａｒ　ｔｅｌｅｐｈｏｎｅ）、ウェブ機器、サーバ、ネットワークルータ、スイッチ又はブリッジ、プロセッサ、又は機械によって取られる動作を指定する一連の命令（順次的又は他の方

法）を実行できる任意の機械であり得る。また、１つのコンピューティングデバイスが図示されているが、「コンピューティングデバイス」という用語は、説明された方法のうち任意の１つを行うための命令セット（又は、複数のセット）を個別的又は共通に実行する任意の機械の集合（例えば、コンピュータ）を含むように使用されなければならない。

【０５５４】

例示的なコンピューティングデバイス２１００は、バス２１３０を介して通信する処理デバイス２１０２、メインメモリ２１０４（例えば、ＲＯＭ（ｒｅａｄ－ｏｎｌｙ ｍｅｍｏｒｙ）、フラッシュメモリ、ＳＤＲＡＭ（ｓｙｎｃｈｒｏｎｏｕｓ ＤＲＡＭ）又はＲＤＲＡＭ（Ｒａｍｂｕｓ ＤＲＡＭ）のようなＤＲＡＭ）、静的メモリ２１０６（例えば、フラッシュメモリ、ＳＲＡＭ（ｓｔａｔｉｃ ｒａｎｄｏｍ ａｃｃｅｓｓ ｍｅｍｏｒｙ））、及びセカンダリーメモリ（例えば、データ格納デバイス）２１１８を含む。

【０５５５】

処理デバイス２１０２は、マイクロ・プロセッサ、中央処理デバイスなどのような１つ以上の汎用プロセッサを示す。特に、処理デバイス２１０２は、ＣＩＳＣ（ｃｏｍｐｌｅｘ ｉｎｓｔｒｕｃｔｉｏｎ ｓｅｔ ｃｏｍｐｕｔｉｎｇ）マイクロプロセッサー、ＲＩＳＣ（ｒｅｄｕｃｅｄ ｉｎｓｔｒｕｃｔｉｏｎ ｓｅｔ ｃｏｍｐｕｔｉｎｇ）マイクロプロセッサー、ＶＬＩＷ（ｖｅｒｙ ｌｏｎｇ ｉｎｓｔｒｕｃｔｉｏｎ ｗｏｒｄ）マイクロプロセッサー、他の命令のセットを実現するプロセッサ、又は、命令のセットの組合せを実現するプロセッサであり得る。また、処理デバイス２１０２は、ＡＳＩＣ（ａｐｐｌｉｃａｔｉｏｎ ｓｐｅｃｉｆｉｃ ｉｎｔｅｇｒａｔｅｄ ｃｉｒｃｕｉｔ）、ＦＰＧＡ（ｆｉｅｌｄ ｐｒｏｇｒａｍｍａｂｌｅ ｇａｔｅ ａｒｒａｙ）、ＤＳＰ（ｄｉｇｉｔａｌ ｓｉｇｎａｌ ｐｒｏｃｅｓｓｏｒ）、ネットワークプロセッサなどのような１つ以上の特殊目的の処理デバイスであり得る。処理デバイス２１０２は、本明細書で説明された動作及びステップを行うための処理ロジック（命令２１２２）を実行するように構成される。

【０５５６】

コンピューティングデバイス２１００は、ネットワークインタフェースデバイス２１０８をさらに含んでもよい。また、コンピューティングデバイス２１００は、ビデオディスプレイユニット２１１０（例えば、ＬＣＤ（ｌｉｑｕｉｄ ｃｒｙｓｔａｌ ｄｉｓｐｌａｙ）、ＣＲＴ（ｃａｔｈｏｄｅ ｒａｙ ｔｕｂｅ））、英数字入力デバイス２１１２（例えば、キーボード又はタッチスクリーン）、カーソル制御デバイス２１１４（例えば、マウス又はタッチスクリーン）、及びオーディオデバイス２１１６（例えば、スピーカ）を含んでもよい。

【０５５７】

データ格納デバイス２１１８は、上述した任意の１つ以上の方法又は機能を実現する１つ以上の命令のセット２１２２が格納された１つ以上の機械）可読格納媒体２１２８、又は、さらに具体的には、１つ以上の非一時的にコンピュータ可読格納媒体）を含んでもよい。コンピュータシステム２１００、メインメモリ２１０４、及びコンピュータで可読格納媒体を構成する処理デバイス２１０２によって実行される間に、命令２１２２は、メインメモリ２１０４及び／又は処理デバイス２１０２内に完全又は少なくとも部分的に存在し得る。

【０５５８】

上述した様々方法は、コンピュータプログラムによって実現され得る。コンピュータプログラムは、上述した１つ以上の様々な方法の機能を行うために指示するように構成されたコンピュータコードを含んでもよい。そのような方法を行うためのコンピュータプログラム及び／又はコードはコンピュータのようなデバイス、１つ以上のコンピュータで可読記録媒体、又は、より一般的には、コンピュータプログラム製品上に提供されてもよい。コンピュータで可読記録媒体は、一時的又は非一時的であり得る。例えば、１つ以上のコンピュータで可読記録媒体は、電子、磁気、光学、電磁気、赤外線、又は半導体システム、又は、データ送信（例えば、インターネットを介してコードをダウンロード）のための

電波媒体であり得る。代案的に、１つ以上のコンピュータで可読記録媒体は、半導体又は固体状態メモリ、磁気テープ、着脱式コンピュータディスケット、ＲＡＭ（ｒａｎｄｏｍ　ａｃｃｅｓｓ　ｍｅｍｏｒｙ）、ＲＯＭ（ｒｅａｄ－ｏｎｌｙ　ｍｅｍｏｒｙ）、剛性磁気ディスク、及び光学ディスク－ＣＤ－ＲＯＭ、ＣＤ－Ｒ／Ｗ、又はＤＶＤと同様な１つ以上の物理的コンピュータで可読記録媒体の形態を有する。

【０５５９】

一実施形態で、ここで説明されたモジュール、コンポーネント及びその他の特徴は、個別コンポーネントとして具現されたり、個別化サーバの一部としてＡＳＩＣＳ、ＦＰＧＡ、ＤＳＰ又は類似のデバイスのようなハードウェアコンポーネントの機能に統合され得る。

【０５６０】

「ハードウェアコンポーネント」は、特定の動作を行うことのできるタイプの（例えば、一時的でない（ｎｏｎ－ｔｒａｎｓｉｔｏｒｙ））物理的なコンポーネント（例えば、１つ以上のプロセッサセット）であり、特定の物理的方式で構成されたり配列されてもよい。ハードウェアコンポーネントは、特定の動作を行うよう永久的に構成された専用回路又はロジックを含んでもよい。ハードウェアコンポーネントは、ＦＰＧＡ（ｆｉｅｌｄ　ｐｒｏｇｒａｍｍａｂｌｅ　ｇａｔｅ　ａｒｒａｙ）又はＡＳＩＣのような特殊目的のプロセッサを含んでもい。また、ハードウェアコンポーネントは、特定の動作を行うためにソフトウェアによって一時的に構成されるプログラミング可能なロジック又は回路を含んでもよい。

【０５６１】

したがって、「ハードウェアコンポーネント」という文句は物理的に構成されたり、永久的に構成されたり（例えば、ハードウェアに内蔵された）、又は、特定の方式で動作したり記述された特定の動作を行うように一時的に構成（例えば、プログラミング）されるタイプのエンティティ（ｅｎｔｉｔｙ）を含むものとして理解されなければならない。

【０５６２】

機械（ｍａｃｈｉｎｅ）は、例えば、物理的機械、論理的機械、仮想機械、コンテナ、又は実行可能なコードを含むために一般的に用いられるメカニズムであり得る。機械は単一機械であってもよく、又は、機械が同じタイプであるか、又は複数のタイプであるかに関わらず、複数接続された又は分散された機械を示す。

【０５６３】

モジュール及びコンポーネントは、ハードウェアデバイス内のファームウェア又は機能回路で実現されてもよい。また、モジュール及びコンポーネントは、ハードウェアデバイス及びソフトウェアコンポーネントの任意の組合せ又はソフトウェア（例えば、機械可読媒体又は送信媒体に格納又は具現されたコード）でのみ実現される。

【０５６４】

特に説明しない限り、次の説明から明らかなように、「送信（ｓｅｎｄｉｎｇ）」、「受信（ｒｅｃｅｉｖｉｎｇ）」、「決定（ｄｅｔｅｒｍｉｎｉｎｇ）」、「比較（ｃｏｍｐａｒｉｎｇ）」、「可能（ｅｎａｂｌｉｎｇ）」、「保持（ｍａｉｎｔａｉｎｉｎｇ）」、「識別（ｉｄｅｎｔｉｆｙｉｎｇ）などのような用語は、コンピュータシステム又は類似の電子コンピューティングデバイス（コンピュータシステムのレジスタ及びメモリ内の物理的（電子的）量で表現されたデータをコンピュータシステムメモリ又はレジスタ又は他の情報ストレージ内の物理量と同様に表現される他のデータに操作及び変換）送信又はディスプレイデバイスの動作及びプロセスを指し示す。

【０５６５】

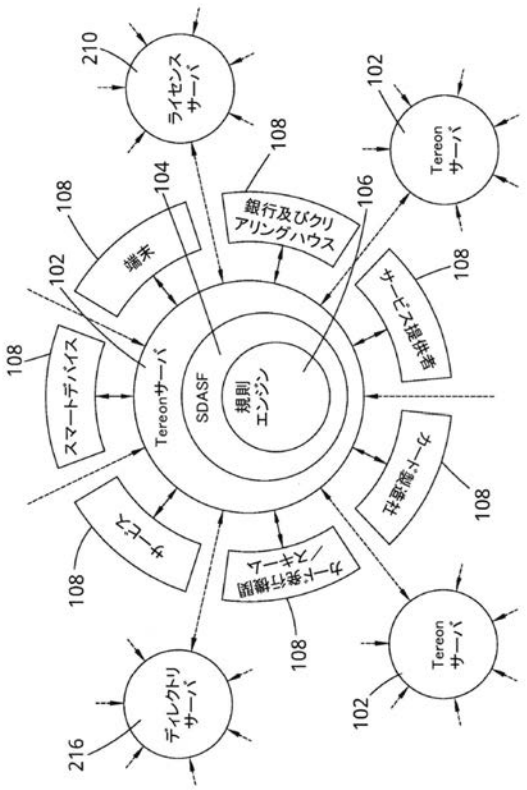
上述した説明は、例示的であり、制限的ではないことを理解しなければならない。上述した説明を読んで理解すれば、多くの異なる実現例が当業者にとって明白になるのであろう。本発明は、特定の例示的な実現例を参照して説明されたが、説明される実施形態に限定されず、添付の請求範囲の思想及び範囲内で変形及び変更して実施できることは理解できるのであろう。したがって、明細書及び図面は制限的な意味であるよりも例示的な意味

として見なす。したがって、請求範囲が属する均等物の全体範囲と共に、添付された請求範囲を参照して本発明の範囲が決定されなければならない。

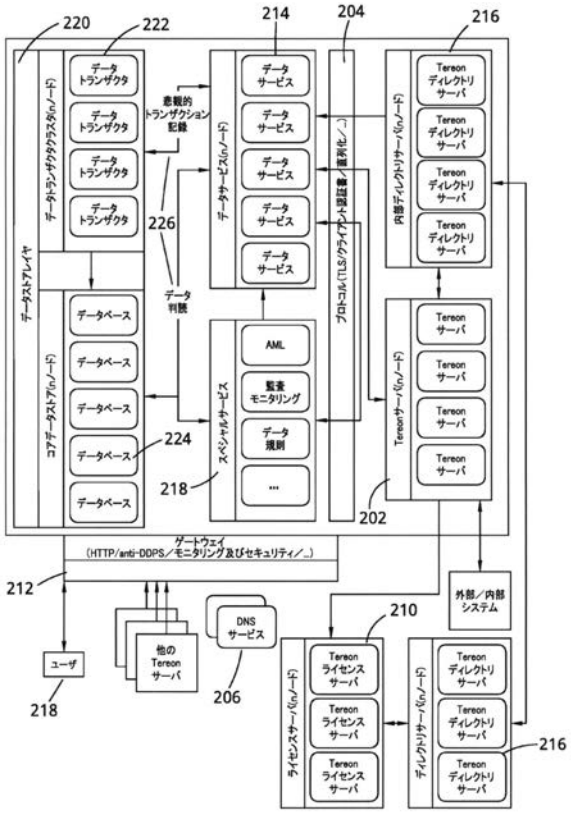
【０５６６】

様々な側面の全ての選択的特徴は、他の全ての側面に関する。記述された実施形態の変形例が考慮され、例えば、開示された全ての実施形態の特徴が任意の方式により組み合わせられてもよい。

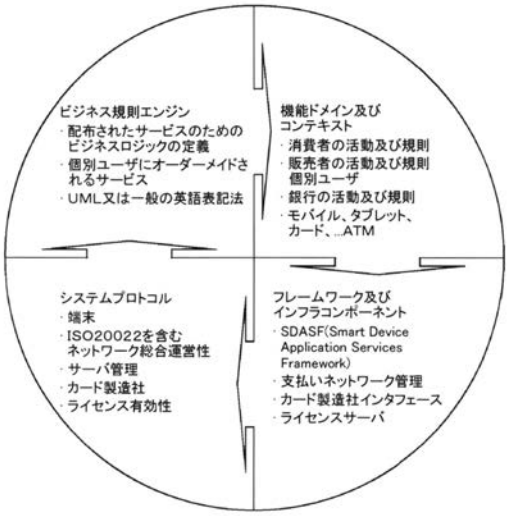
【図１】



【図２】



【図２ａ】



【図３】

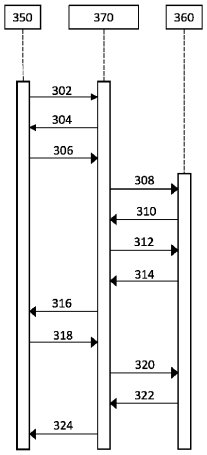


FIG. 3

【図４】

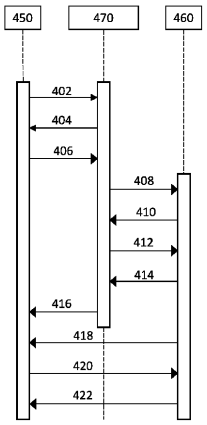
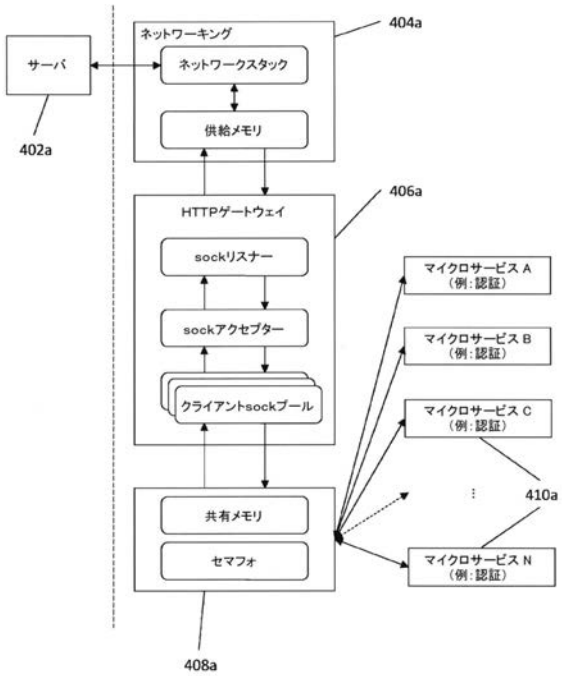


FIG. 4

【図４ａ】



【図５】

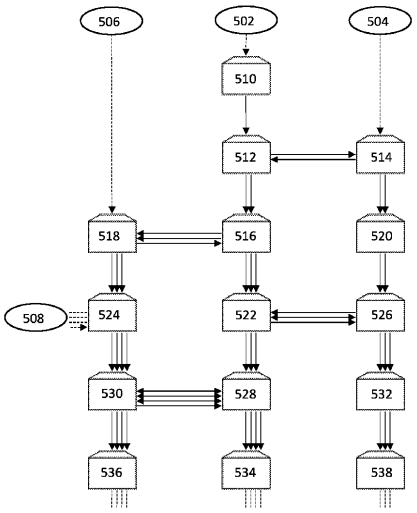


FIG. 5

【図６】

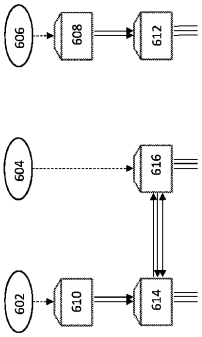


FIG. 6

【図６ａ】

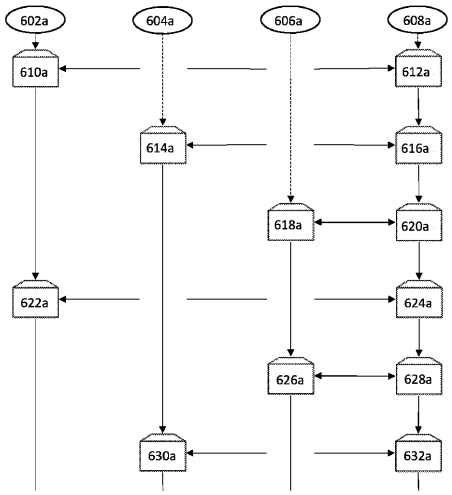


FIG. 6a

【図７】

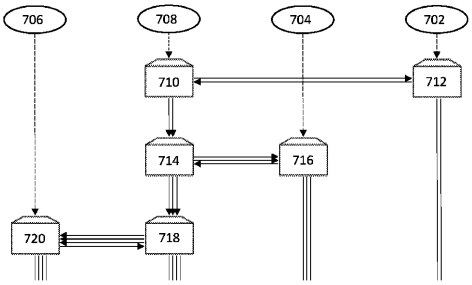


FIG. 7

【図８】

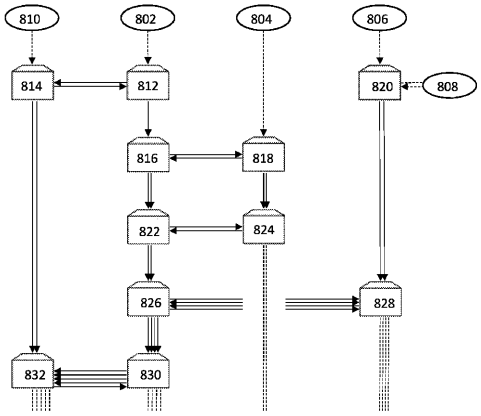
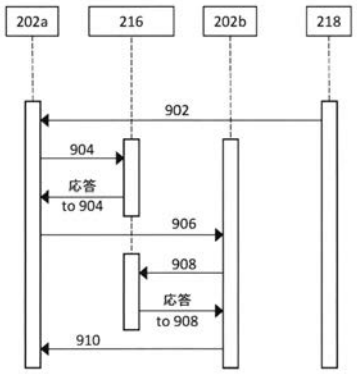


FIG. 8

【図９】



【図１０】

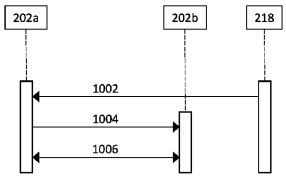
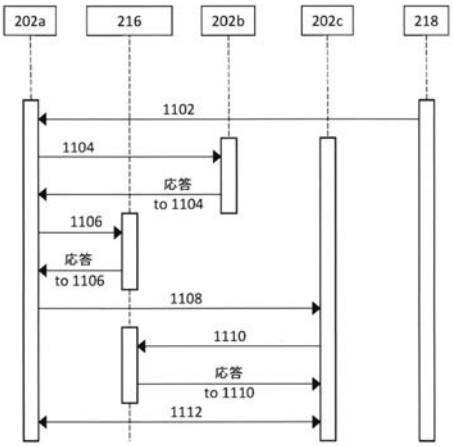
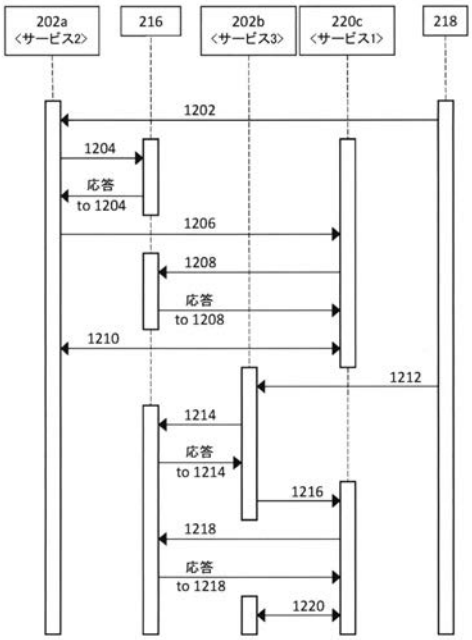


FIG. 10

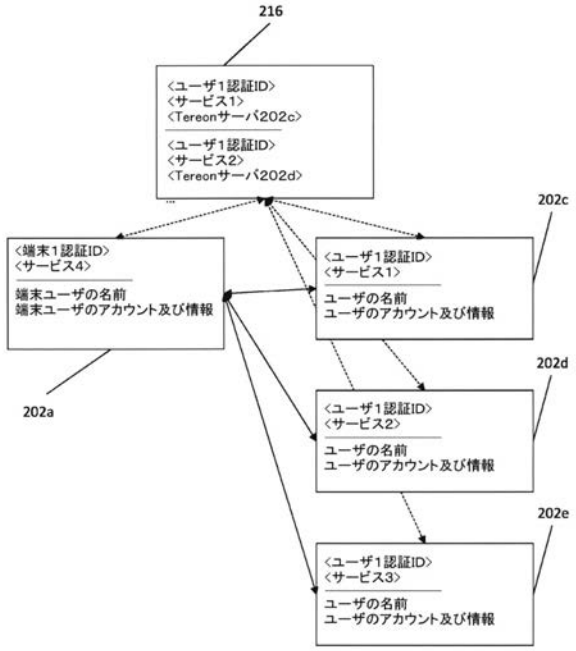
【図１１】



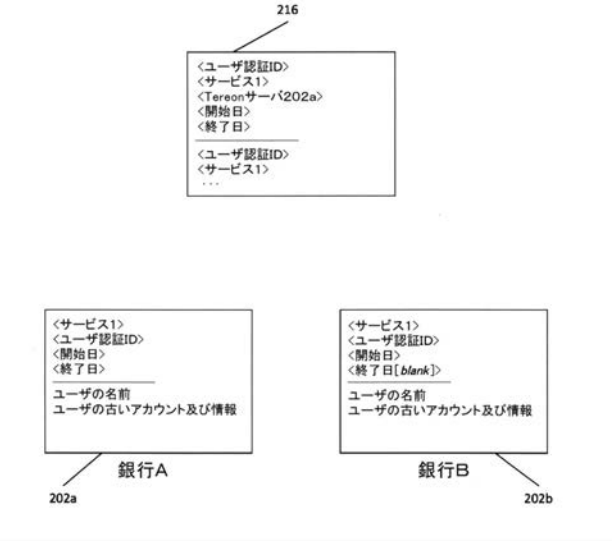
【図１２】



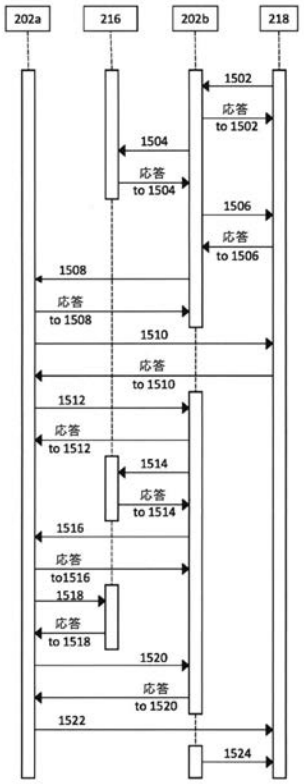
【図１３】



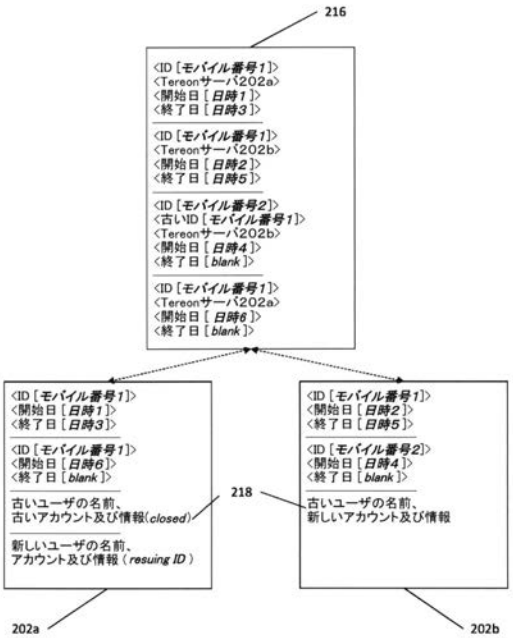
【図１４】



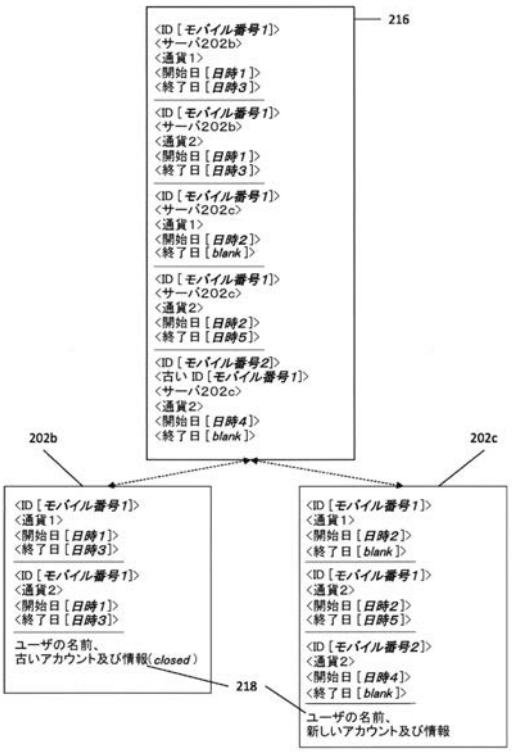
【図１５】



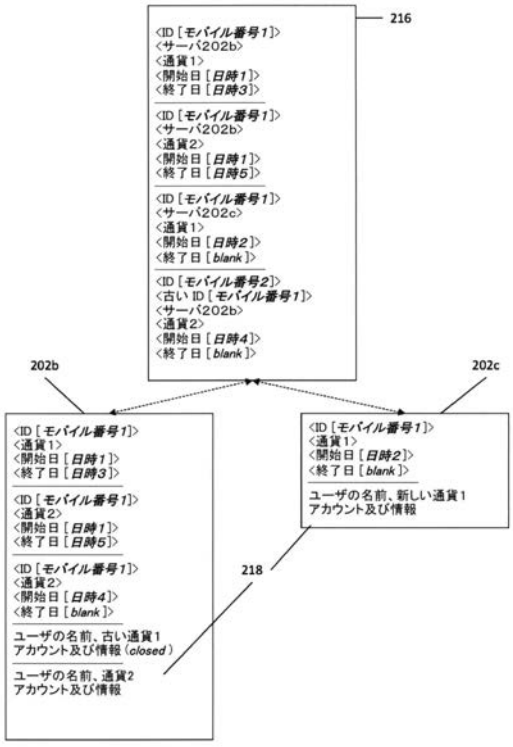
【図１６】



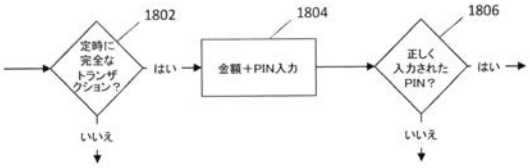
【図１７】



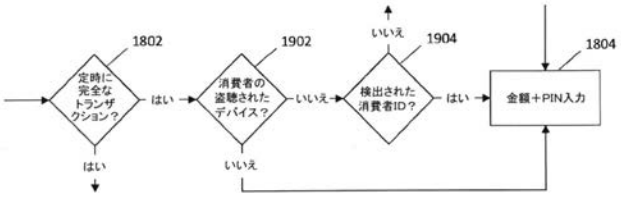
【図１７ａ】



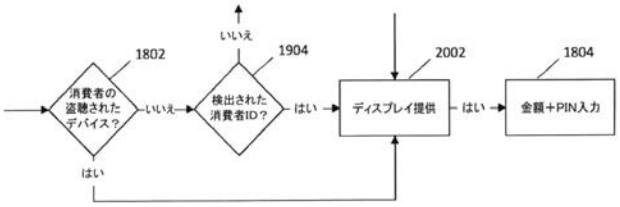
【図１８】



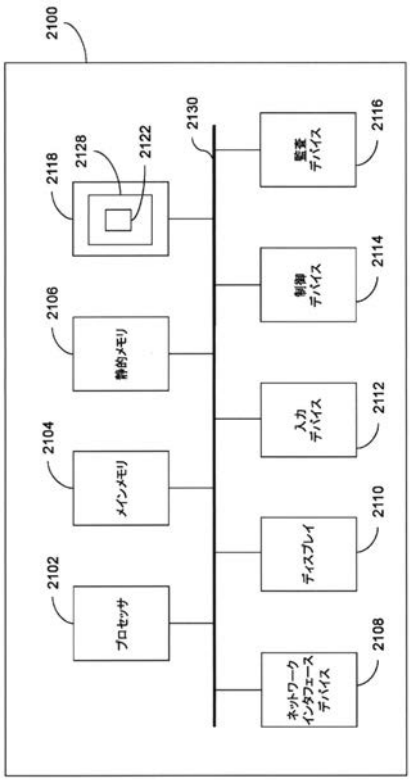
【図１９】



【図２０】



【図２１】



【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2017/052004

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L9/32
ADD.

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

B. FIELDS SEARCHED

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

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	EP 2 897 051 A2 (PALANTIR TECHNOLOGIES INC [US]) 22 July 2015 (2015-07-22) paragraph [0024] paragraph [0049] - paragraph [0061]; figure 2 paragraph [0097] - paragraph [0106] paragraph [0114] -----	1-60
X	US 2016/063100 A1 (ANTON DHRYL [US] ET AL) 3 March 2016 (2016-03-03) paragraph [0068] - paragraph [0069] -----	1-60
X	US 2015/269570 A1 (PHAN CHARLES [CH] ET AL) 24 September 2015 (2015-09-24) paragraph [0017] - paragraph [0020] ----- -/--	1-60

[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
9 January 2018	17/01/2018

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	Tenbieg, Christoph

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

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2017/052004

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 2011/099277 A1 (YAO XUEFENG [US] ET AL) 28 April 2011 (2011-04-28) paragraph [0018] paragraph [0064] - paragraph [0075]; figure 9 paragraph [0023] - paragraph [0032]; figures 2A, 2B	61-84
A	US 2012/150750 A1 (LAW SIMON [CA] ET AL) 14 June 2012 (2012-06-14) paragraph [0051] - paragraph [0077] paragraph [0090] paragraph [0135] - paragraph [0145]	61-85
X	US 2015/142650 A1 (JOHNSTON CALE T [US] ET AL) 21 May 2015 (2015-05-21) paragraph [0034] - paragraph [0040] paragraph [0059] - paragraph [0080]	85-101
A	US 8 250 640 B1 (ZHANG YUAN [CN] ET AL) 21 August 2012 (2012-08-21) column 4, line 10 - column 6, line 21 page 7, line 7 - line 61	85-101
X	US 9 241 004 B1 (APRIL BENJAMIN [US]) 19 January 2016 (2016-01-19) column 2, line 53 - column 4, line 27 column 5, line 24 - column 6, line 22 column 7, line 61 - column 8, line 51	102-111
A	EP 2 028 794 A1 (HOPLING GROUP B V [NL]) 25 February 2009 (2009-02-25) paragraph [0064] - paragraph [0092]	102-111
X	US 2015/371216 A1 (OLAWALE ABIMBOLA OMONIYI [FI] ET AL) 24 December 2015 (2015-12-24)	112-115, 121,122
Y	paragraph [0004] - paragraph [0005] paragraph [0035] - paragraph [0042]; figure 3	116-120
Y	Faculdade De Ciências ET AL: "UNIVERSIDADE DE LISBOA SECURING USSD IN MOBILE FINANCIAL TRANSACTIONS (A PRACTICAL PROPOSAL FOR M-FINANCE)", , 31 December 2011 (2011-12-31), XP055433720, Retrieved from the Internet: URL:http://repositorio.ul.pt/bitstream/10451/8707/1/ulfc104212_tm_Paula_Cravo.pdf [retrieved on 2017-12-11]	116-120
A	paragraph [0005] - paragraph [0009]	112-115, 121,122

-/--

3

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

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2017/052004

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 2008/077979 A1 (JEFFRIES CLARK D [US] ET AL) 27 March 2008 (2008-03-27) paragraph [0010] paragraph [0030] - paragraph [0042] -----	123-136
A	US 2010/049975 A1 (PARNO BRYAN [US] ET AL) 25 February 2010 (2010-02-25) paragraph [0052] - paragraph [0064]; figures 2, 3 -----	123-136
A	US 2005/025091 A1 (PATEL ALPESH [US] ET AL) 3 February 2005 (2005-02-03) paragraph [0046] - paragraph [0047] paragraph [0063] - paragraph [0070] -----	123-136
X	US 2014/379576 A1 (MARX JOSEPH A [US] ET AL) 25 December 2014 (2014-12-25) paragraph [0017] - paragraph [0020] paragraph [0037] - paragraph [0041] -----	137-150
X	US 2011/093939 A1 (BARBOUR MARC R [US] ET AL) 21 April 2011 (2011-04-21) paragraph [0014] paragraph [0020] - paragraph [0024] paragraph [0035] - paragraph [0044] paragraph [0059] - paragraph [0061] -----	137-150
A	US 2013/046690 A1 (CALMAN MATTHEW A [US] ET AL) 21 February 2013 (2013-02-21) paragraph [0026] - paragraph [0031] paragraph [0038] - paragraph [0047] paragraph [0090] - paragraph [0126] -----	137-150
X	US 5 617 537 A (YAMADA SHIGEKI [JP] ET AL) 1 April 1997 (1997-04-01) column 1, line 10 - line 16 column 6, line 42 - column 7, line 45 column 12, line 17 - column 13, line 42 column 24, line 26 - column 44, line 31 -----	151-196
A	US 6 026 474 A (CARTER JOHN B [US] ET AL) 15 February 2000 (2000-02-15) column 4, line 12 - line 22 column 5, line 30 - column 7, line 57 column 11, line 17 - column 17, line 59 -----	151-196
A	US 2013/232217 A1 (KRISTIANSSON JOHAN [SE] ET AL) 5 September 2013 (2013-09-05) paragraph [0032] - paragraph [0066] -----	151-196

3

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

INTERNATIONAL SEARCH REPORT

International application No.
PCT/GB2017/052004

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☒ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☒ No protest accompanied the payment of additional search fees.

International Application No. PCT/GB2017/052004

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-60

   method and devices for recording a data transaction.

   ---

2. claims: 61-84

   method and devices for providing a service to a device.

   ---

3. claims: 85-101

   method and devices for migrating data.

   ---

4. claims: 102-111

   method and devices for exchanging data between entities

   ---

5. claims: 112-122

   method and devices for communicating securely in a USSD session

   ---

6. claims: 123-136

   method and devices for generating and exchanging shared secrets

   ---

7. claims: 137-150

   method and devices for identifying a user.

   ---

8. claims: 151-196

   method and devices for communicating between modules in a computer.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/GB2017/052004

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 2897051 A2	22-07-2015	EP 2897051 A2	22-07-2015
		EP 3255549 A1	13-12-2017
		US 2015188715 A1	02-07-2015
		US 2016254906 A1	01-09-2016
US 2016063100 A1	03-03-2016	NONE	
US 2015269570 A1	24-09-2015	NONE	
US 2011099277 A1	28-04-2011	NONE	
US 2012150750 A1	14-06-2012	NONE	
US 2015142650 A1	21-05-2015	NONE	
US 8250640 B1	21-08-2012	NONE	
US 9241004 B1	19-01-2016	NONE	
EP 2028794 A1	25-02-2009	NONE	
US 2015371216 A1	24-12-2015	US 2015371216 A1	24-12-2015
		WO 2015193538 A1	23-12-2015
US 2008077979 A1	27-03-2008	US 2005132192 A1	16-06-2005
		US 2008077979 A1	27-03-2008
		US 2008229105 A1	18-09-2008
US 2010049975 A1	25-02-2010	NONE	
US 2005025091 A1	03-02-2005	AT 408298 T	15-09-2008
		AU 2003294330 A1	18-06-2004
		CA 2506670 A1	10-06-2004
		CN 1714560 A	28-12-2005
		EP 1563668 A2	17-08-2005
		US 2005025091 A1	03-02-2005
		WO 2004049672 A2	10-06-2004
US 2014379576 A1	25-12-2014	NONE	
US 2011093939 A1	21-04-2011	CN 102576399 A	11-07-2012
		EP 2491515 A2	29-08-2012
		US 2011093939 A1	21-04-2011
		US 2013205382 A1	08-08-2013
		WO 2011049711 A2	28-04-2011
US 2013046690 A1	21-02-2013	NONE	
US 5617537 A	01-04-1997	DE 69424114 D1	31-05-2000
		DE 69424114 T2	09-11-2000
		EP 0646876 A1	05-04-1995
		US 5617537 A	01-04-1997
US 6026474 A	15-02-2000	AU 7303498 A	10-06-1998
		US 6026474 A	15-02-2000
		WO 9822891 A1	28-05-1998
US 2013232217 A1	05-09-2013	EP 2636199 A1	11-09-2013

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/GB2017/052004

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		US 2013232217 A1	05-09-2013
		WO 2012060747 A1	10-05-2012

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

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

（特許庁注：以下のものは登録商標）
１．ブルートゥース
２．ＪＡＶＡＳＣＲＩＰＴ

(72)発明者　デイビス、ラーズ
イギリス国　ＲＨ１９　３ＡＦ　ウエスト　サセックス　イースト　グリンステッド　ハイ　ストリート　３７　クラウン　ハウス