



日銀CBDCフォーラム WG4資料

「口座型」 「UTXO」 データモデルの比較

2024年 9 月
ソラミツ株式会社

1. 今回は、台帳の管理体制が「分担管理」（複数の仲介機関が台帳管理）のパターンを想定して「口座型」と「UTXO」のデータモデルの比較を行います。
2. ユーザの保有額制限、送金の1日あたり取引額制限などを設ける事を想定します。
3. 上記の判定のため、ユーザの残高合計、1日あたり取引額などを記録するDB（以下「ユーザ口座DB」）を設け、残高が更新されないと次の送金ができないなどの機能を搭載します。
4. 性能要件は「通常時スループット数万件/秒、ピーク時スループット10万件以上/秒、レイテンシ数秒以内」と想定します。

5. さらに、「口座型」の「中央システム」における送金などの「取引履歴」を格納する

DBについて、以下の2つのパターンを想定します。

A、「取引履歴」を通常のDBシステムに蓄積する場合 (w/o DLT)

B、「取引履歴」をDLTなどに蓄積する場合 (with DLT)

6. また、「口座型」の「仲介機関システム」においては、取引時の処理について、以下の2つのパターンを想定します。

A、取引時の処理を通常のDBシステムで行う場合 (w/o DLT)

B、取引時の処理にDLTを活用する場合 (with DLT)

- ・ハイブリッドシステムにより、「状態を更新」と「状態の変化の蓄積」を両立
- ・DLTに基づく情報の改ざん防止と中央集権DBによるスケールメリットの両者を享受
- ・デジタル人民元は、二重使用の防止・改ざん防止に一部DLT技術を使用と報道

1. 残高管理を中央集権DBで実施

- ・高スループット/低レイテンシを実現
- ・スケーラブルキューなどによるスケールアウト

2. 「取引履歴」をDLTに蓄積

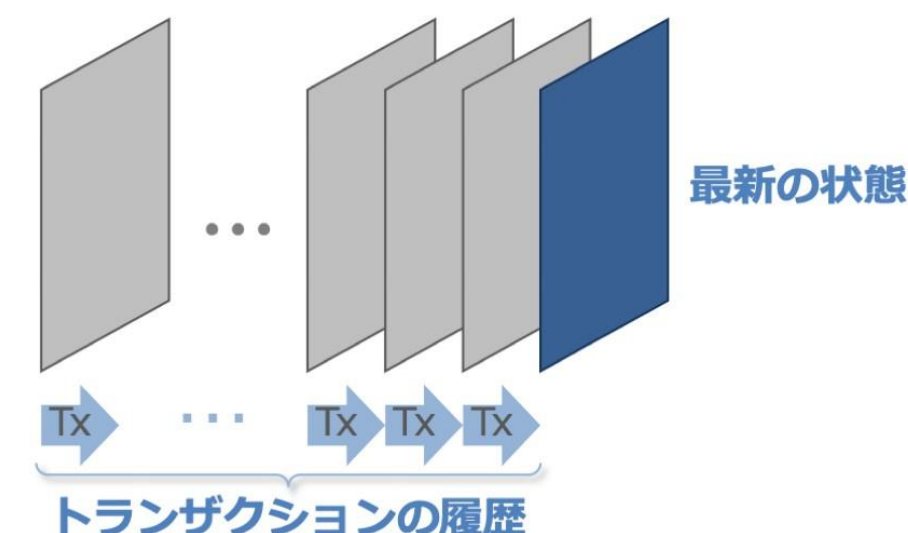
- ・「取引履歴」を「状態の変化」として蓄積
- ・「取引履歴」の改ざん防止・真正性の担保
- ・不正な送金時に「取引履歴」を活用して検証

3 両方の利点を活用したハイブリッド・システム

- ・分散化された「残高台帳」で、即座に「状態を更新」
- ・「取引履歴」をキューに書き込む
- ・その後、「取引履歴」を不変な「状態の変化の蓄積」としてキューからDLTに記録

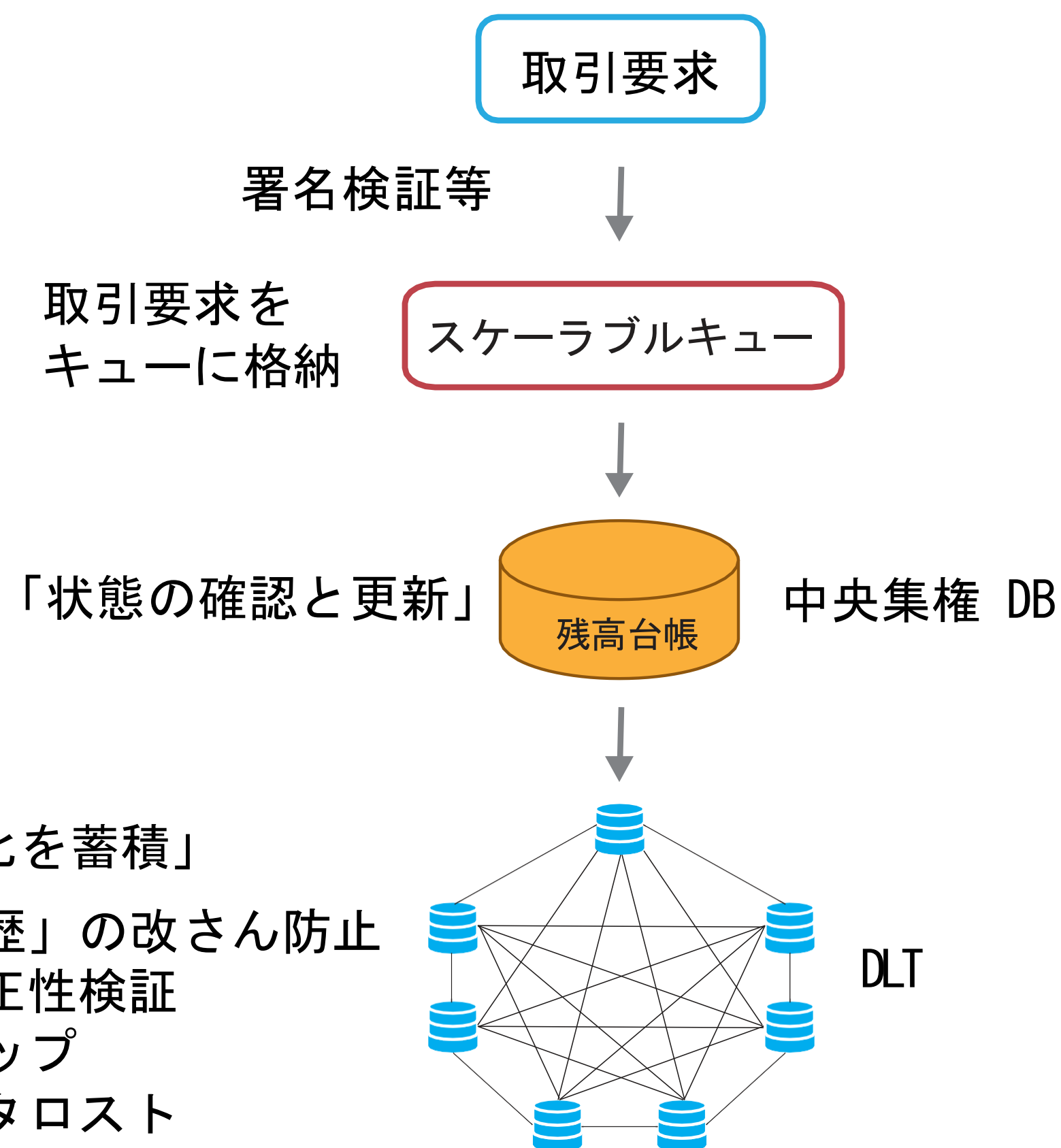
分類の軸②：「状態を更新」するか「状態の変化を蓄積」するか

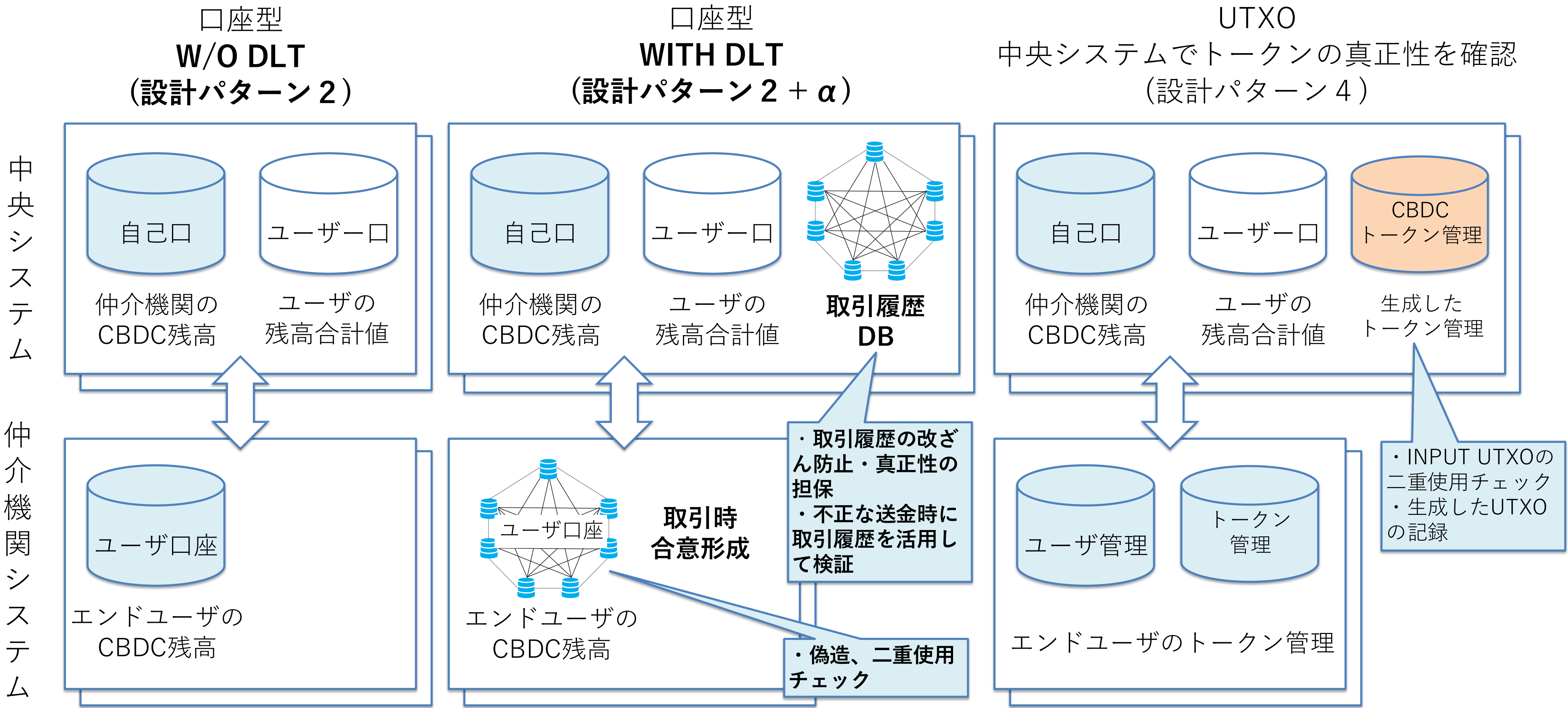
- ・伝統的な金融機関の勘定系システムのデータモデルでは、トランザクションの最新の状態を管理することを重視。
- ・他方でイーサリアムやビットコインのデータモデルは、トランザクションの履歴を、ブロックチェーンで記録することを重視。



出所：日本銀行 CBDCフォーラムWG4資料

1. 「取引要求」の署名検証等を実施し、なりすましや否認を防止
2. 「取引要求」をキューに記録
スケーラブルな仕組みにより10万件以上/秒に対応
3. 「状態の確認と更新」：中央集権DBの「残高台帳」で取引に必要な残高を確認し、残高更新
4. タイムスタンプ等の優先順位に従い、「取引要求」の合意形成を実施
5. 合意形成後の「取引履歴」をDLTに記録：
「状態の変化を蓄積」





出所：20240410 NTTデータ様 CBDCフォーラムWG4第3回会合資料

			データモデル			一言コメント
			口座型 w/o DLT	口座型 with DLT	UTXO	
性能	1	スループット	○	○	△	・ 口座型の場合は、仲介機関を跨る取引の場合のみ中銀の台帳を更新する。 ・ UTXOの場合、すべての取引において中銀にてトークンごとの二重使用等の防止の検証を集中的に行う必要があるため、ボトルネックになる可能性がある。 ・ 検証のスピードアップするアルゴリズムやハードウェアなどが整備されれば解消される可能性はあるが、膨大なリソースが必要となる可能性がある。
	2	レイテンシー	○	○	△	・ UTXOの場合、中銀にてトークンごとの二重使用等の防止の検証を集中的に行う必要があるため、その処理に時間がかかる可能性がある。 ・ 口座型の場合、仲介機関内の取引であれば処理時間が有利である。 ・ 検証をスピードアップするアルゴリズムやハードウェアなどが整備されれば解消される可能性はあるが、膨大なリソースが必要となる可能性がある。
	3	並行処理性	△	△	○	・ UTXOの方が、並行処理性は高い。 ・ 口座型においても、並行処理のアルゴリズムやハードウェアなどが整備されれば並行処理性を向上できる可能性はある。
	4	残高確認の 容易性	○	○	△	・ UTXOの場合は、都度ユーザ毎のトークンの総和を計算するか、最新のユーザ単位の残高を保存し常にアップデートする必要がある。 ・ ユーザ単位の残高をDBに保存して確定する場合には、並列性の阻害要因となる可能性が高い。 ・ 口座型においては、ユーザ単位の残高や取引回数は常に更新される。

			データモデル			一言コメント
			口座型 w/o DLT	口座型 with DLT	UTXO	
信頼性	5	セキュリティ リスクへの耐性 (中央システム)	△	○	○	・ 中央システム内部の不正（口座型の残高の改ざん、UTXOのトークン額面の改ざんなど）が考えられる。 ・ UTXOについては、すべての取引で中央システムによるトークンの真正性チェックを行うため、セキュリティ耐性は向上する。 ・ 口座型については、DLTなどに蓄積した利用履歴の活用により、ユーザ残高の改ざんの発見を行うことは可能であり、UTXOと同等レベルにすることは可能と考える。
	6	セキュリティ リスクへの耐性 (仲介機関システム)	△	○	○	・ 仲介機関内部の不正（口座型のユーザ口座の残高の改ざん、UTXOのトークン額面の改ざんなど）が考えられる。 ・ UTXOについては、すべての取引で中央システムによるトークンの真正性チェックを行うため、セキュリティ耐性は向上する。 ・ 口座型については、DLTなどの活用により仲介機関においても、ユーザ残高の改ざん防止を行うことは可能であり、UTXOと同等レベルにすることは可能と考える。
	7	取引のアトミック性（整合性）	○	○	○	・ 口座型では、ファイナリティの観点から中央システムのユーザ口と仲介機関システムのユーザ口座を同時に更新すれば取引の整合性は保証される。 ・ UTXOでは、取引時のトークン検証により整合性が保証される。

			データモデル			一言コメント
			口座型 w/o DLT	口座型 with DLT	UTXO	
拡張性	8	追加サービスとの親和性	△	○	○	・ UTXO型は、中央システムでのトークン検証を通じてデータ整合性を保証するため、仲介機関への信頼性の高いCBDCの提供が可能と考えられる。 ・ 口座型（w/o DLT）は、仲介機関の内部不正によるユーザ口座の残高改ざんなどのセキュリティ耐性が劣っており、追加サービスの促進にあたり必要なセキュリティ対策を講じてゆく必要があり、システム対応およびコンプライアンス上の負担が大きい。 ・ 口座型においても、DLTなどの活用により仲介機関での構造上のセキュリティ耐性を高めることは可能であり、追加サービスとの親和性においてもUTXOと同等レベルにすることは可能と考える。
	9	実装容易性	○	○	△	・ UTXOの場合、残高計算の負荷が大きい。また、最新のユーザー単位の残高をDBに常に保存する場合や送金限度額などを判定する場合には、並列性の阻害要因となる可能性が高く実装が困難になる可能性が高い。
	10	システムアーキテクチャー	○	○	△	・ UTXOでは、ほとんどすべての取引でお釣りを返す必要があるため、通常各取引の出力として2つのUTXOを生成することになる。 ・ したがって多くのUTXOが生成され、それを将来の検証のために追跡する必要があり複雑なアーキテクチャーになると考えられる。 ・ 口座型では、取引を行う際に各アカウントの残高を追跡するだけで済むためシンプルなアーキテクチャーになる。

1. 「UTXO」は、トークンの真正性チェックによるセキュリティ耐性などに優位性があるが、スループット、レイテンシー、ユーザ毎の残高確認の容易性、実装容易性などに課題がある。
2. 「口座型」は、内部不正などによるセキュリティ耐性、追加サービスでのセキュリティ対策などの課題がある。
3. 「口座型」とDLTを組み合わせることにより、セキュリティ耐性を向上し、追加サービスとの親和性を高めることができるのではないか。

照会先

Contact Us

メール : info@soramitsu.co.jp

担当 : 武宮、宮沢、森、米津



ソラミツ株式会社

代表取締役社長 宮沢和正

住所 : 〒151-0051 東京都渋谷区千駄ヶ谷5-27-5リンクスクエア新宿16F