

CBDCフォーラムWG4 台帳データモデルに関する考察

株式会社JPX総研

2024年7月



今回は、台帳の管理体制が「**中央管理**」（中央銀行が単一の台帳を管理）のパターンを前提として、「**口座残高型データモデル**」と「**口座残高型以外のデータモデル**」の比較を行います。

「口座残高型以外のデータモデル」として、「**固定額面トークン型**」、「**UTXO型**」の2つを想定します。

※ 「固定額面トークン型」の詳細は日本銀行様[「概念実証フェーズ1」結果報告書](#)をご参照ください。
「UTXO型」の用語については、2024年1月30日SBI R3 Japan様プレゼンテーション[「UTXOモデルの特徴理解」](#)にお示しいただいた用法を踏襲しています。

対象機能は「送金」とし、保有額、取引額および取引回数にいずれも制限を設けるものとします。

（送金の都度、各種制限に違反しないことを必ず確認してからCBDCを移転するものとします）

➤ 送金の都度、取引額・回数制限判定時に参照できるように、ユーザごとの累計取引額や取引回数を記録するDB（以下「取引履歴DB」）を設けます。

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。

データモデル①（中央管理×口座残高型）

中央銀行システムのDB構成案

CBDC台帳DB

口座ID	残高
A001	10000
B002	5000
:	:

取引履歴DB

口座ID	取引履歴
A001	XXX
B002	XXX
:	:

上限値DB

口座ID	上限値
A001	YYY
B002	YYY
:	:

送金指図

「A001からB002に¥500送金」

中央銀行システムでは、**必要な排他制御を行ったうえで**（同じ口座IDを対象とする他の処理を待機させたうえで）、**取引額・回数制限判定、保有額制限判定**を行い、問題なければ残高と取引履歴を更新。

CBDC台帳DB

口座ID	残高
A001	10000 9500
B002	5000 5500
:	:

取引履歴DB

口座ID	取引履歴
A001	XXX XXXX
B002	XXX XXXX
:	:

上限値DB

口座ID	上限値
A001	YYY
B002	YYY
:	:

1ユーザ1口座の前提で記載。1ユーザ複数口座の場合は多少変わる。

データモデル②（中央管理×固定額面トークン型）

中央銀行システムのDB構成案

CBDC台帳DB

トークンID	額面	ユーザID
100-1	100	A001
500-1	500	A001
1000-1	1000	B002

取引履歴DB

ユーザID	取引履歴
A001	XXX
B002	XXX
:	:

上限値DB

ユーザID	上限値
A001	YYY
B002	YYY
:	:

送金指図

「トークン500-1をA001からB002に送金」

トークンの保有者ユーザIDを
更新することでCBDC移転

考慮すべき事項①

保有額制限判定を行うために、**ユーザ単位の残高計算が必要***。

(*) ユーザ単位の残高DBを併設することで、都度の残高計算を回避する設計も可能だが、その場合、口座残高型対比での優位性は何か？（追加サービスとの親和性？）

考慮すべき事項②

意図した金額分のトークンがないときに**両替かお釣りが必要**。

CBDC台帳DB

トークンID	額面	ユーザID
100-1	100	A001
500-1	500	A001 B002
1000-1	1000	B002

取引履歴DB

ユーザID	取引履歴
A001	XXX XXXX
B002	XXX XXXX
:	:

上限値DB

ユーザID	上限値
A001	YYY
B002	YYY
:	:

データモデル③（中央管理×UTXO型）

中央銀行システムのDB構成案

CBDC台帳DB

ユーザID	金額
A001	3000
A001	100
B002	500
:	:

取引履歴DB

ユーザID	取引履歴
A001	XXX
B002	XXX
:	:

上限値DB

ユーザID	上限値
A001	YYY
B002	YYY
:	:

送金指図

「A001からB002
に¥500送金」

INPUT 「A001 : ¥3000」
OUTPUT 「B002 : ¥500」
OUTPUT 「A001 : ¥2500」

考慮すべき事項

保有額制限判定を行うために、**ユーザ単位の残高計算が必要***。

(*) ユーザ単位の残高DBを併設することで、都度の残高計算を回避する設計も可能だが、その場合、口座残高型対比での優位性は何か？（追加サービスとの親和性？）

お釣り授受や両替を別途行わずに、意図した金額を移転できる

CBDC台帳DB

ユーザID	金額	
A001	3000	消滅
A001	100	
B002	500	
A001	2500	お釣り
B002	500	送金額
:	:	

取引履歴DB

ユーザID	取引履歴
A001	XXX XXXX
B002	XXX XXXX
:	:

上限値DB

ユーザID	上限値
A001	YYY
B002	YYY
:	:

性能に影響する観点	中央管理 口座残高型	中央管理 固定額面トークン型	中央管理 UTXO型
CBDC台帳の更新対象レコード数	2レコード	1レコード以上 (非常に多い場合も)	2レコード以上
ユーザ単位の排他制御	必要 各種制限判定のため	必要 各種制限判定のため	必要 各種制限判定のため
ユーザ単位の残高計算（送金都度）	不要 (1ユーザ1口座の前提) ※ 1ユーザ複数口座の場合 は必要（ただしユーザ 単位の残高DBを併設す れば不要）	必要 ユーザ単位の保有額制限判 定のため ※ ユーザ単位の残高DBを 併設すれば不要だが、 それなら口座残高型で よい？	必要 ユーザ単位の保有額制限判 定のため ※ ユーザ単位の残高DBを 併設すれば不要だが、 それなら口座残高型で よい？
ダブルスペントチェック (トークンやUTXOは使用可能か)	—	不要 中央の管理者が管理するDB に記録されていることを もって使用可能とみなせる	不要 中央の管理者が管理するDB に記録されていることを もって使用可能とみなせる
両替	不要	いずれかが 必要な場合あり	UTXOの性質により、 両替やお釣りの効果を実現
お釣り	不要		

○＞△＞▲（相対評価、▲もその一点をもって選択肢から除外するような致命的な欠点とまでは言えない）

		中央管理 口座残高型	中央管理 固定額面 トークン型	中央管理 UTXO型	コメント
性能	スループット	○	▲	△	送金時に更新対象となるレコード数はデータモデルごとに異なる（大まかな傾向としては、口座残高型が最も少なく、次がUTXO型、固定額面トークン型は最も多くなりそう）。このため、スループット要件を達成する際に使用するマシンパワーに違いが生じると考えられる。
	レイテンシ	○	▲	△	固定額面トークン型・UTXO型ではユーザ単位の残高計算が必要な分、レイテンシが悪化。 「両替」「お釣り」が必要な固定額面トークン型は相対的にレイテンシが悪いと考察。
	並列処理性	○	○	○	固定額面トークン型やUTXO型は複数の相手と同時に取引できそうに一見思えるが、各種制限判定のためにユーザ単位で排他制御を行うのであれば、モデル間の差異とならないと考察。

比較（つづき）

○＞△＞▲（相対評価、▲もその一点をもって選択肢から除外するような致命的な欠点とまでは言えない）

		中央管理 口座残高型	中央管理 固定額面 トークン型	中央管理 UTXO型	コメント
信頼性	セキュリティリスクへの耐性	○	○	○	いずれのモデルでも、中央銀行や仲介機関におけるセキュリティリスク管理は必要。モデル間に差異は見当たらない。
	取引のアトミック性（整合性）	○	○	○	いずれのモデルでも、取引ごとにアトミック性を確保することが可能。モデル間に差異は見当たらない。
拡張性	追加サービスとの親和性	▲？	○？	△？	「お金に色を付ける」ようなユースケースがあれば、最も親和性が高いのは固定額面トークン型、次点でUTXO型か（口座残高型でも工夫次第かかもしれず「？」つき）。
	実装容易性	○	△～▲	△	構成がシンプルでマシンパワーの消費も少ないと思われる口座残高型が最も実装容易と推察。 固定額面トークン型は、送金時に指定するトークンを選択するアルゴリズム次第で複雑さが増す（実装難度が上がる）可能性。

送金時にCBDC台帳において更新するレコードの数

口座残高型

口座ID	残高
A001	10000 9500
B002	5000 5500
:	:

口座残高型の場合、送金側エンドユーザー・受領側エンドユーザーの**2レコード**を更新

固定額面トークン型

トークンID	額面	ユーザID
100-1	100	A001
500-1	500	A001 B002
1000-1	1000	B002

固定額面トークン型の場合は、送金時に指定するトークンの数だけレコードを更新

UTXO型

ユーザID	金額	
A001	3000	消滅
A001	100	
B002	500	
A001	2500	お釣り
B002	500	送金額
:	:	

UTXO型の場合は、Input・Outputのレコード数だけ更新

固定額面トークン型の場合、送金額に加えて、**送金対象とするトークンの指定方法**や**固定額面の設定内容**も更新対象レコード数に影響を及ぼす。これらが設計上の論点となると予想。

- 送金対象とするトークンの指定方法は、複数考えられる。
例えば、9980円支払いたいとき、送金者エンドユーザの保有トークンをかき集めてちょうどの金額で送金するか、あるいは、10000円分のトークン送金＆お釣り20円分のトークンで処理するか？
後者の方が更新対象レコードは少ない（が、お釣りトークンが必ず準備されているとは限らないほか、そもそも受領側エンドユーザの保有トークンをお釣りとして勝手に移転してもよいのか？）。
- 固定額面の設定としては、現在の紙幣や貨幣と同様（1円、・・・,5000円,10000円）とするか、それとも、異なる設定（例えば、2のべき乗（まさに“デジタル”）で、1円,2円,4円、・・・,1024円、・・・）とするか？
（現金のように利用者にとって「わかりやすい」金額設定である必要はあるか？）