

2024年4月8日
日本銀行決済機構局

CBDCフォーラム WG2
「追加サービスとCBDCエコシステム」
第5回会合の議事概要

1. 開催要領

(日時) 2024年1月18日(木) 14時00分～16時30分
(形式) 対面形式及びWeb会議形式
(参加者) 別紙のとおり

2. プレゼンテーション

- 事務局から、「BISによるProject Rosalindのご紹介」の資料¹に基づいて、説明を行った。
- 事務局から、WG2の今後の展開イメージについて、説明を行った。
- 株式会社メルペイより、CBDCの実装に向けた考慮点について、プレゼンテーションが行われた。プレゼンテーションにおいては、決済事業の立ち上げの経験に根差して、CBDCを検討する際に考慮すべき点として、①認証・認可の安全性確保と標準化、②APIの技術選択・標準化、③インターオペラビリティの確保、④トランザクション集中下における処理性能の維持などが挙げられた。

3. ディスカッション

- プレゼンテーションを受けて、参加者によるディスカッションを行った。モデレータは、日本銀行が担当した。議論の概要は、以下のとおり。

【WG2の今後の展開に関する議論】

(日本銀行) Project Rosalindは、民間事業者とBIS(国際決済銀行)、BOE(イングランド銀行)が協力して、33のAPIエンドポイントを作成しユースケースの検討を行ったものであり、本WGにおける検討と関係が

¹ https://www.boj.or.jp/paym/digital/d_forum/dfo240408a.pdf 参照。

深いと考えたため、共有させていただいた。

たとえば、同プロジェクトのAPI設計の原則（資料12ページ）において、採用された業界標準として、REST API、JSON、FAPI 認証、ISO20022などが挙げられているが、こうした標準の「パッケージ」は参考になる。また、プロジェクトから得られた示唆（資料23ページ）として挙げられている「シンプルかつ中核的な（core）API機能のセットは、多様なユースケースをサポートしうる」などの点も、シンプルなAPI機能によってエコシステム全体に高い機能拡張性をもたらすための設計を考えるうえで、参考になると思われる。この点、同プロジェクトで構築したAPIの一覧には、プログラマビリティを実現するものとして口座内の資金のロック機能が挙げられている（資料15ページ）。プログラマビリティという用語は多義的ではあるが、APIで表現する場合は資金のロック機能によってかなりの程度実現できる可能性が示されており、今後の検討にとって示唆的である。

今後も、本WGでは、これまでと同様に、各社のご関心に応じたテーマについて、プレゼンテーションを通じて知見を共有いただきたいと考えている。それに加えて、Project Rosalindも参考にしながら、サンドボックスを構築し、APIを活用したCBDCの機能拡張性やユースケースについて、皆様と一緒に「手を動かしながら」検討する場を設けるといったことも考えている。なお、これは、パイロット実験の枠組みの中でCBDCフォーラムと並行して日本銀行が実施している、「実験用システムの構築」とは異なるものである点、ご留意いただきたい。

（参加者）APIサンドボックスプロジェクトで構築される環境では、台帳にDLTを用いることも想定しているのか。

（日本銀行）今回のAPIサンドボックスは、台帳そのものではなく、台帳の外部連携をどのようにするかをフォーカスして取り組むことを考えており、台帳の技術選定は中心的なテーマではないこともあり、台帳は通常データベースで構築することを考えている。

（参加者）本WGの今後の展開として、サンドボックス上でAPIの研究やユースケースの検討を行っていく際の順序としては、APIの研究を行うことから、ユースケースの検討を行うことを想定されているか。逆の順序、つまり、どのようなユースケースが必要であるかを検討したうえで、そのユースケースを実現するためのAPIの研究を行うことも考えられるのでは

ないか。

(日本銀行) サンドボックス上でのAPIの研究とユースケースの検討は並走して行い、相互にフィードバックを行うことを想定している。最初に作成するいくつかのAPIを用いてどのようなことを実現できるかを考え、実現できないものを実現するために新たなAPIの構築にトライする、といった具合に、フィードバックの中で柔軟に進めていきたい。

(参加者) 紹介のあったCBDCの機能拡張性に関する研究は、とても興味深いと感じている。Project Rosalindをなぞるのではなく、CBDCフォーラムでの議論も踏まえて、日本の独自性が出てくると、非常に有意義なものになるのではないか。

(日本銀行) 日本独自の実験の意義についてはご指摘のとおりと考える。もっとも、Project Rosalindの結論はかなり抽象度の高いものであり、本WGで行うサンドボックスプロジェクトで得られる結果が似たようなものになる可能性はある。いずれにせよ、結論のみならず、結論に至るまでの議論の過程で得られる示唆、実際に構築してはじめて分かる知見など、副産物的に得られるものも重要な成果になると考えている。

【プログラマブル・ペイメントとプログラマブル・マネーに関する議論】

(参加者) Project Rosalindでは、プログラマビリティに関するAPIエンドポイントによって、様々なことが実現できるといったことを、手触り感を持って確認できたことが分かった。一方で、ECBのように、デジタルユーロはプログラマブル・マネーを志向しない中央銀行もある。参考までに、プログラマビリティに対する世界的な潮流があれば共有いただきたい。

(日本銀行) Project Rosalindにおけるプログラマビリティは、支払いの手続きをプログラムすることで条件付きの支払いを可能とするような、プログラマブル・ペイメントのことを指すと認識している。プログラマブル・ペイメントにおいては、マネー自体をプログラム可能なものとするのではなく、外側のシステムを用いてマネーの動きを制御するようなイメージである。

一方、おっしゃるようなプログラマブル・マネーは、マネーそのものに

プログラムが組み込まれ制御されるイメージであり、国際的な議論の中でも、プログラマブル・ペイメントとは別のものと整理する考え方が多いように見受けられる。このように、国際的な潮流としては、プログラマブル・マネーとプログラマブル・ペイメントを丁寧を使い分ける傾向があると感じている。欧州においても、プログラマブル・マネーについては否定的であるが、条件付き支払いのようなプログラマブル・ペイメントについては排除せず検討がなされていると考えられる。

(参加者) プログラマブル・ペイメントとプログラマブル・マネーの話が出たが、これは重要な整理だと感じている。Project Rosalindにおいては、プログラマビリティとして各種のロック機能が中心に扱われていたが、これはプログラマブル・ペイメントのことを指す。例えば、ロック機能を用いてエスクローサービスを設計する際には、プログラマブル・ペイメントの枠組みの中で考えていくことになる。今後こうした議論においては、プログラマブル・ペイメントとプログラマブル・マネーを切り分けて議論することが重要だろう。

【CBDCの実装に向けた考慮点① 認証・認可の安全性確保と標準化】

(参加者) CBDCに関して考えるべきポイントは、第一に認証・認可である。標準化された認証・認可の仕組みがとにかく重要だ。決済とは、結局のところ、誰であるかを確認して、どこの店で、いくら払うかを認可するということである。あえて、「決済は認証・認可である」と断言したい。

(参加者) 過去を振り返ると、決済事業者は口座振替依拠の銀行KYCの仕組みを採用した。このことが日本のフィンテックの拡大に寄与した一方で、これまで起こってきた決済関係のインシデントは、結局は認証・認可の問題となっている。

(参加者) 「決済は認証・認可」という言葉に非常に共感した。現在のいわゆるペイ系の事業者が円滑に立ち上がった背景には、リアルタイム口座振替やウェブ口座振替の存在があると考えている。たしかに、今とは異なる認証・認可の仕組みも考えられたと思うが、これらの口座振替があるからこそ円滑な立ち上げができたことを踏まえると、メリット・デメリットがあるということだろう。

(参加者) C B D Cでは標準化された認証・認可を採用してほしい。OAuthでもFIDOでもよいが、そういう標準化されたものを使わないと脆弱性があったときに個別に対応しなければならず、コストも時間もかかる。標準を意識した技術選択の重要性を強調したい。

今回のAPIサンドボックスプロジェクトでも、民間事業者と日銀が、あるべき認証・認可について、現状にこだわらずに検討した上で実際に作る練習をすることが重要ではないか。

(参加者) 標準化やインターオペラビリティが重要ということは、これまでも関係者に共有されていたのだが、にもかかわらず結果的に実現しなかったということが多くあった。この背景には、標準化やインターオペラビリティは民間の競争の中で決まっていくものだといった考えがあったり、関係者の認識が既存の仕組みに引っ張られて収まるところに収まってしまったこともあったと感じる。

今回せっかくC B D Cをゼロから作るのだったら、一番優れたものをみんなで作ろうというコンセプトで、例えば仲介機関はこの方式に切り替えてもいいのではないかと、といった将来を見越した標準化の動きが求められるのだと考える。

今後、このWGでのディスカッションや、APIサンドボックスプロジェクトを通じて、「本当の」インターオペラビリティをみんなで作りにいくことは、とても大切なことではないか。

(参加者) C B D Cの統一的な認証・認可の仕組みに乗らないとC B D Cを扱えない、といったことになれば、将来を見越した標準化を促すのではないかと、いちエンジニアとしては思う。

(参加者) 既存の仕組みに合わせた方が早いというのはその通りだが、既存のものに引きずられ過ぎると、セキュリティホールが残ってしまったり、開発しにくい、使いにくい、といったことになりうる。C B D Cを作るのであれば、やっぱりベストというか、理想的なものを想定して、みんなで作りに行くというコンセプトが良いと考えている。

(参加者) 既存の電文に合わせるのは、おそらくホストコンピューター側がそういう電文じゃないと受け付けないといった点がやはりあるのだろう。受入可能な電文に変換する機能を、残高を持っているDBの前に置かなけれ

ばいけないのだろうと考える。

(日本銀行) C B D Cは存在していないものなので、他のシステムと結合して
いて現にサービスを提供している既存のシステムのリプレイスに比べると、
多くのことをゼロから考えやすいものと思う。他方で、既存のシステム
ともつながっていくことになるので、C B D Cを一つのきっかけとして
既存の仕組みについても考えることが重要と思う。

積極的な発言に感謝する。サービス提供の基盤としてのC B D Cについ
て、目指すべき方向性についてご意見をたくさん頂いた。

【C B D Cの実装に向けた考慮点② A P Iの技術選択・標準化】

(参加者) 次に、A P Iの技術選択・標準化は重要だ。具体的には、型
(type)がある言語でA P Iを構築した方が良いと考えており、例として
GraphQLを挙げたい。最近REST APIではなくGraphQLで構築されている
A P Iも多い。そこでは、スキーマの定義ファイルを用意しておいて、そ
れから自動生成でコードを作っている。こうすることで、A P Iを提供す
る側と利用する側で同じ型になるため、バグも入りやすく品質に対して大
きな効用がある。こうした仕組みを、銀行も決済事業者も関係するエンテ
ィティが共有すると開発の容易性も大いに高まると考える。

(参加者) なお、標準化について言えば、たとえば単一のブロックチェーンに
皆がプログラムをデプロイするというのも一つの解だが、ブロックチェー
ンを採用せず基盤はバラバラであっても、仕様が共有されることで容易な
相互接続が実現するというあり方で、同じことが実現できると思われる。

(参加者) 型は非常に重要である。大規模なシステム開発時にはPythonがよ
く用いられるが、型がないため、集団での開発は困難となりうる。

(参加者) 型がない状態で集団で開発することは、困難に直面することにな
る。避けたほうがよいと思われる。

(参加者) A P Iの議論となると、REST APIが第一のオプションとなりがちな
ことに疑問を感じている。GraphQLは、データの型が存在することに加え、
例えばREST APIであればA P Iを3回たたかねばならないような処理が、
GraphQLであれば1回で済ますこともできるといった特性があり、

また柔軟性もあるものと感じている。とはいえ、世界的に見ても GraphQL が第一に取り上げられていることはない印象である。GraphQL には何か課題が存在しているのか、ご存じの方がいれば教えていただきたい。

(参加者) おっしゃった GraphQL の利点が、短所にもなり得るためであると考えている。GraphQL は自由なクエリを実行できる点が優れている。一方、自由なクエリの裏側で、どのような処理がなされるのかが分かりづらく、処理時間を把握しづらい。より自由で複雑なクエリを実行するほど、返答が返ってくるまでの時間や裏側の処理もより複雑化するため、コストが読めなくなってしまう。GraphQL を使って複雑なクエリを文法上作成することは可能だが、個人的には、複雑なクエリの実行は許容すべきではないと考えており、事前に登録したクエリだけを受け付けるような仕組みを活用することがポイントであると考えている。

(参加者) Project Rosalind の紹介において、「効率性とパフォーマンスを向上させるためには、非同期的な API モデルが探求される」といったように非同期化に触れられていた。GraphQL は同期的な処理を必要とするため、この点にうまく対処できないのではないかと考える。

(参加者) 開発に関連した既存のアセットを活用できるという利点があって GraphQL を推しているが、「型を持つ」ことのメリットは、REST API でも享受できる。すなわち、REST API にも型を追加的に付与することで (アノテーション)、GraphQL と同様の効果が得られる。

(参加者) API は作って終わりではない。常にメンテナンスやアップデートをしていかなければならないし、それに伴って、ライブラリ、各種のドキュメント、それらを含む SDK (software development kit、ソフトウェア開発キット) を更新していかなければならない。

過去に自分が関わっていた事業では、API を DSL (domain-specific language、ドメイン固有言語) で記述し、ライブラリ、ドキュメント、SDKなどを、Java や Ruby など多くの言語に対応する形で自動生成していた。今回の API サンドボックスの段階ではこうした点まで意識しなくてもよいと思うが、実装段階ではディベロッパーの開発の容易性を維持・向上することは、エコシステムの発展にとって死活的に重要であるため常に意識すべき 이슈 だ。

(参加者) APIの検討にあたって、単なるデータ授受だけではなく、APIを利用する側のソフトウェア開発まで考慮して議論することは重要だ。

(参加者) エンジニアファーストを意識しているAPIはよく使われている印象がある。

(参加者) 先ほどから論点になっている型については、プログラマビリティについて検討するときにも考えるべきポイントの一つになるのかなと思う。

(日本銀行) プログラマビリティの話と関係してくる可能性については、トランザクション処理の仕方と同様、型がCBDC全体の機能拡張性に関係があるかもしれないという考え方もあれば、APIなどでうまく繋ぐことで、あまり関係がないものと捉え得るとの考え方もありそうで、追究すべきテーマと考えている。

【CBDCの実装に向けた考慮点③ インターオペラビリティの確保】

(参加者) CBDCと他のシステムとのインターオペラビリティは重要である。ブロックチェーンとのインターオペラビリティを考えておくと、ブロックチェーンでない既存の決済手段とのインターオペラビリティについても知見が蓄積されると考えている。

(参加者) ブロックチェーンとのインターオペラビリティを考えた際に、例えば暗号資産のアカウントからCBDCのアカウントに送金する場合、暗号資産が危険なウォレットから送金されてくる可能性も存在するため、受け取り拒否のような機能も想定しておくことが良いのではないかと感じている。

(参加者) 利用者が、異なるチェーンなりシステムの間でのスワップを第三者の介在なしに行うことは難しいだろう。通常の暗号資産の取引でも、銀行預金と暗号資産の交換を暗号資産交換業者が行っている。CBDCについても、暗号資産が流通するブロックチェーンと、CBDCのシステムの間、なんらかのエンティティが入って媒介するというのではないか。

(日本銀行) インターオペラビリティという言葉は多義的であると感じている。先ほどの話に合ったように、なんらかのエンティティが間に入りブリ

ッジする、もしくはコンバートすることで、ユーザーが円滑に交換できたと感じれば、それはインターオペラブルであると表現できるかもしれない。

すべてをネットワークの深い部分で接続させることだけがインターオペラビリティではないと考えており、法制度や他の仕組みを踏まえたうえで、様々な方法を検討していくのが良いと考えている。

【CBDCの実装に向けた考慮点④ トランザクション集中下における処理性能の維持】

(参加者) トランザクションの集中は処理性能に大きな影響を与えうる。たとえば1つの口座に対して、多くのユーザーが振り込んだり引き落とししたりすると、大量のトランザクションが集中して、データ書き換えのためのレコードロックが積み重なって遅くなってしまう。もちろん振り込み先の口座を分けるなど対処方法はいくつかあるが、いずれにせよきちんと考えるべき問題だ。

(参加者) 決済事業者の中には、トランザクションを追記するモデルを採用しているところもあり、その場合はトランザクション集中が問題になりにくい。出金時にオーバードラフトになってしまう可能性には対応しなければならないが、入金集中には強い仕組みになっている。追記型でなければ起こる2つの口座の同時ロックは回避できるので、これは大きなメリットである。

APIサンドボックスプロジェクトでは、中央システムの処理性能というよりは外部連携をテーマにしているとのことなので、少なくとも当初のテーマではないと思うが。

(日本銀行) この点は、大量のトランザクションを処理するであろうCBDCでは、ポイントの一つになると思う。データを書き換えるときのロックの範囲を小さくする工夫については、CBDCフォーラムではWG4で扱っている。WG2では、データモデルやトランザクション処理の方式について深入りしないながらも、基盤としてのCBDCについて、追加サービスや機能拡張性の観点から、今後も皆様からご意見を頂ければと思う。

4. 次回予定

次回の会合は2月27日（火）に開催。

以 上

CBDCフォーラム WG2
「追加サービスとCBDCエコシステム」
第5回会合参加者

(参加者) ※五十音・アルファベット順

株式会社イオン銀行
株式会社インキュリオン
株式会社ジェーシービー
セコム株式会社
ソニー株式会社
ソフトバンク株式会社
大和証券株式会社
東京海上日動火災保険株式会社
トヨタファイナンシャルサービス株式会社
株式会社トレードワルツ
日本電気株式会社
野村證券株式会社
株式会社野村総合研究所
株式会社ふくおかフィナンシャルグループ
株式会社マネーフォワード
株式会社みずほ銀行
三井住友海上火災保険株式会社
株式会社三井住友銀行
株式会社三菱 UFJ 銀行
株式会社メルペイ
株式会社横浜銀行
BIPROGY 株式会社
株式会社 BOOSTRY
株式会社 NTT データ
株式会社 NTT ドコモ
PayPay 株式会社
株式会社 Startale Labs Japan
TIS 株式会社

(事務局)

日本銀行