中央銀行デジタル通貨に関する実証実験
「パイロット実験」の進捗報告書（2026 年 6 月）
実験用システムの構築と検証【別冊】

日本銀行決済機構局

2026 年 6 月

目次

はじめに

本稿は、「中央銀行デジタル通貨に関する実証実験「パイロット実験」の進捗報告（2026 年 6 月）」（以下、「本冊」という）における「２．実験用システムの構築と検証」の部分の別冊として、読者の理解を深めていただくために、本冊の各項目について、その詳細を説明することを目的に作成したものである。

まず、実験用システムを用いて実施した高負荷試験において、レコード分割の有効性を評価するための「同一口座集中試験」や、社会実装時の留意点を評価するための「混合業務負荷試験」を実施したため、それらの試験の目的、内容、結果を整理し（1.1）、社会実装を行ううえでの性能設計上のインプリケーションを整理する（1.2）。

次に、実験用システムで実装しない機能・非機能を中心に実施した机上検討の状況を整理する。すなわち、CBDC システムの機能面では、「送金の処理フロー」に関連するテーマに着目し、送金に必要な宛先情報管理の機能、反対取引処理、EC 決済に関する検討（2.1）を行ったほか、送金の処理フローの見直しに関する検討（2.2）、エンドポイントデバイスに関する検討（2.3）も行った。加えて、「CBDC の相互運用性」に関連するテーマに着目し、CBDC と民間マネーとの交換について検討を行った（2.4）。

また、CBDC システムの非機能面では、CBDC を安心・安全に利用するために重要となる「セキュリティ」に関連するテーマに着目し、サイバーセキュリティ対策、不正アクセス・不正取引への対策や、不正作出対策について検討を行った（2.5）。このほか、社会実装時に重要となる「可用性」（システムを停止することなく継続して利用できる能力）に関連するテーマに着目し、計画停止、障害停止、レジリエンス（強靭性）について検討を行った（2.6）。

なお、ここで述べる実験用システムの設計内容や机上での検討内容は、現時点で社会実装時における設計を確定するものではないことを予め明確にしておく。

１．実験用システムを用いた検証の結果

1.1 性能評価

1.1.1 同一口座集中試験

（１）試験の内容

同一口座集中試験では、レコード分割をしない１つの口座（レコード数：１）またはレコード分割した 1 つの口座（レコード数：複数）に対して入金および出金のリクエストを大量投入し、遅滞なく当該リクエストが処理されるか観察した。レコード分割する場合の分割数[1]については、1 つの口座だけで 6,000TPS の処理を行うとの目標を設定[2]し、これを達成しうるレコード分割数を設定した。その際には、「必要なレコード分割数＝処理スループット [TPS[3]]×台帳管理システムにおいて１処理にかかる時間（レイテンシ）［秒]」という関係があることを前提に、台帳管理システムにおいて１処理にかかる時間（レイテンシ）を実測のうえ算出した。その結果、出金処理は 120 分割、入金処理は 60 分割となった[4]ため、その条件で１口座に対して入金および出金のリクエストを大量投入する実験を行った[5]。

（２）試験の結果

出金集中、入金集中いずれの場合においても、投入されたリクエストを遅滞なく処理しており、レコード分割数を増やすことでスループットも向上し、1 口座あたり 6,000TPS の処理が実現可能であることを確認した（図表 1,2）。この間、システムのリソース負荷（各サーバーの CPU 使用率やメモリ使用量の指標等）をみても特段の問

---

[1] ここでは、レコード分割をした結果、1 口座から作られるレコード数のことを分割数と呼称する。ただし、分割数 1 については、レコード分割をしない状態のこととする。例えば、50 分割は、レコード分割をした結果、1 口座が 50 レコードに分割された状態のことをいう。

[2] 実験用システムでは、2 つの台帳管理システム（出金側と入金側）を用いて、10,000TPS の更新系取引（後述）の処理に対応できる設計としていた。すなわち、片側（出金側や入金側）では 5,000TPS の処理に対応するため、余裕率 20%を乗じて 6,000TPS を目標としていたもの。

[3] Transaction Per Second の略で、1 秒あたりの取引件数のこと。

[4] 出金処理と入金処理とでレコード分割数が異なるのは、実装した送金の処理フローにおいて、出金処理は、入金処理に比べて台帳管理システムにおける処理が増える結果、レイテンシが大きくなり、同じスループットを実現するため必要となる多重度（分割数）が入金処理に比べて大きくなるため。

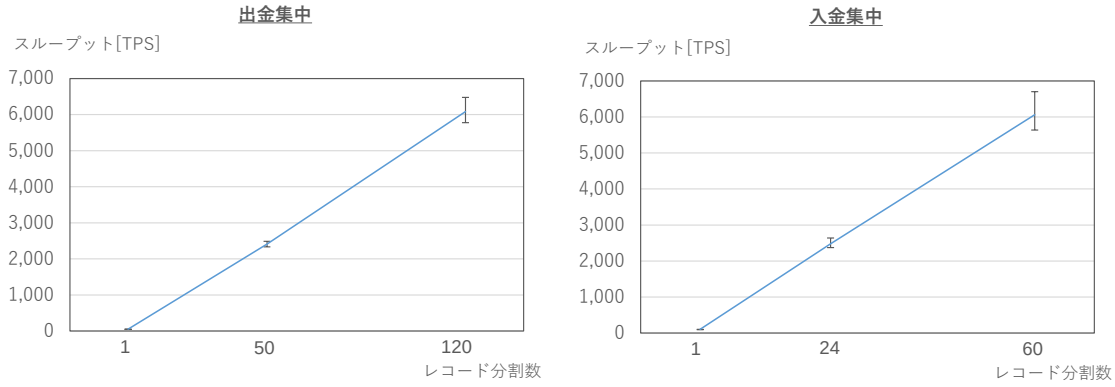
[5] 同一口座集中試験においては、このほかにも分割数 1 の場合（分割無しの場合）やそれ以外の分割数の条件においても試験を実施した。

題は観察されなかったため、同一口座集中の課題に対するレコード分割の有効性が示されたと考えられる。

図表 1　同一口座集中試験の結果

集中の種類	分割数	投入スループット [TPS]（平均値）	実験結果		
			レイテンシ[msec]（平均値）	実績スループット [TPS]（平均値）	実績スループット [TPS]（標準偏差）
出金集中	1	50	156	50	1.1
	50	2,408	151	2,408	52
	120	6,079	189	6,080	439
入金集中	1	100	156	100	2.2
	24	2,475	158	2,475	110
	60	6,065	199	6,060	366

図表 2　レコード分割数と実績スループットの実験結果[6]



もっとも、図表にはないが、レコード分割数を増やし過ぎると、処理能力が低下することも確認した。これは、分割されたレコードを用いて入出金を処理する際、利用可能なレコードの検索が必要となるため、レコードの分割数が増えていくと、この検索に要する処理が性能・システムのリソース上のボトルネックとなることに起因するものである。実際に、レコード分割数を極端に多く（分割数 6,000）したところ、サーバーの CPU 使用率が 100%となり、正常な処理が不可能となる事象が発生した。このため、CBDC の社会実装時にはシステムの設計段階で適切なレコード分割数を見極めることが重要と考えられるほか、事務量の変化にあわせて柔軟にレコード分割数を変更できるような仕組みも考えられる。

[6] グラフ中の各計測点における下線、上線はそれぞれ 5 パーセンタイル、95 パーセンタイル箇所を示す。

1.1.2 混合業務負荷試験

（1）試験の内容

混合業務負荷試験では、更新系取引[7]を 10,000TPS、参照系取引を 40,000TPS、合計 50,000TPS のリクエストを投入した。そのうえで業務種別ごとにスループットやレイテンシ（ターンアラウンドタイム。概ね目標 3 秒以内[8]）を計測し、遅滞なくリクエストが処理されるか観察した[9]。

（2）試験の結果

スループットについては、目標とした 50,000TPS のスループットを実現することができた[10]（図表 3）。

レイテンシについては、参照系取引では大きなボトルネックは観測されなかった一方で、更新系取引において、概ね 3 秒以内で処理を終えたものの、一部で 3 秒を超える業務もあった（図表 4）。これらのうち、オートチャージありの業務は処理が複雑であることに起因するものであることが明らかなため、ここではオートチャージなしの業務を対象にレイテンシが長くなった要因の調査を行った[11]。

---

[7] 更新系取引の対象業務としては、店舗支払い、店舗での反対取引処理（返金）、個人間送金、払出、受入とし、それぞれの事務量については、各業務の比率を 100：1：5：10：1 と仮定したうえで、10,000TPS を案分して設定した。なお、店舗支払いと個人間送金については、オートチャージがある場合とない場合の双方を試験した。

[8] デジタル・ユーロでは、概ね 3 秒以内とすることが仮置きの要件とされている。European Central Bank, "Annex 1: Functional and non-functional requirements linked to the market research for a potential digital euro implementation", 2023 January, https://www.ecb.europa.eu/euro/digital_euro/timeline/profuse/shared/pdf/ecb.dedocs230113_Annex_1_Digital_euro_market_research.en.pdf

[9] その他主な試験の前提としては、仲介機関数を 2 としたうえで、それぞれの仲介機関に個人口座数 600 万口座（2 つの仲介機関合わせて 1,200 万口座）、法人口座数 5 万口座（同 10 万口座）として実験を行った。

[10] 目標以上のスループットを達成しているのは、試験のために利用した負荷投入ツールの投入量の揺らぎにより、設定値（50,000TPS）以上の処理が投入されたことによるもの。

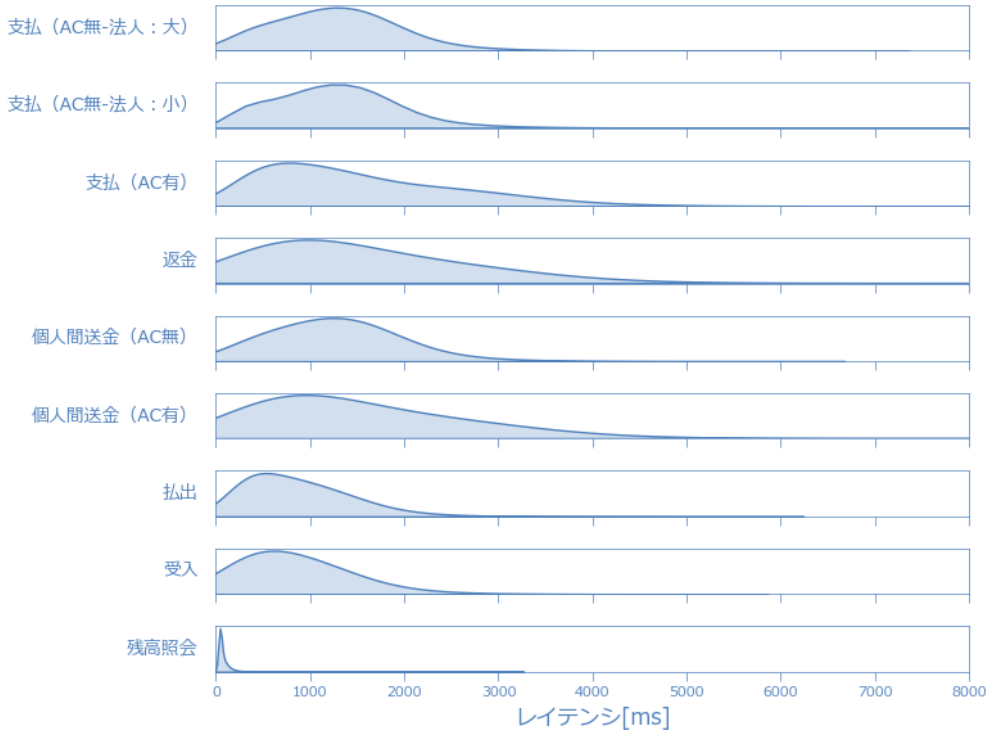
[11] オートチャージありの業務は、チャージ（払出）と送金の 2 種類の処理を行っているため、単純な送金処理よりも複雑。

図表 3　混合業務負荷試験におけるレイテンシ、スループットの結果

業務種別		レイテンシ[ms]				スループット[TPS]			
		平均	最大	標準偏差	99%タイル	目標	平均	最大	標準偏差
	支払（AC無-法人:大）	1,259	6,552	652	3,184	1,000	1,103	1,747	211
	支払（AC無-法人:小）	1,257	7,090	651	3,181	4,982	5,491	8,643	1,043
	支払（AC有）	2,162	8,640	1,027	4,900	2,564	2,825	4,625	731
	返金	2,126	7,450	1,025	4,822	86	91	150	23
	個人間送金（AC無）	1,265	5,631	653	3,190	300	323	492	61
	個人間送金（AC有）	2,131	8,225	1,026	4,907	128	136	250	35
	払出	879	5,522	557	2,621	854	940	1,272	125
	受入	833	4,729	551	2,665	86	91	131	14
	残高照会	83	3,221	81	437	40,000	43,344	50,449	2492
全体						50,000	54,344	64,564	3,598

AC：オートチャージ

図表 4　混合業務負荷試験における業務種別毎のレイテンシ分布[12]

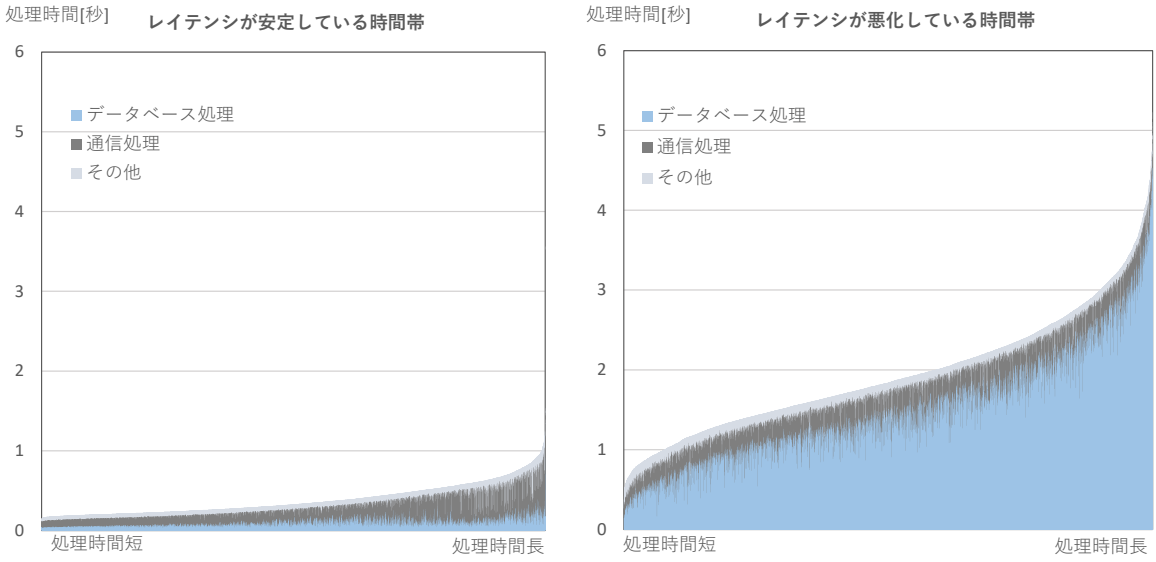


まず、CBDC システムを構成する顧客管理システムと台帳管理システムのいずれでレイテンシの悪化が発生しているかを確認したところ、前者で発生していた。これは、

[12] 図表の縦軸はレイテンシ分布を示しており、その算出にはカーネル確率密度推定量を用いた（Silverman の検定結果を参考にバンド幅を設定して算出）。AC はオートチャージを表す。

処理スパイク[13]に伴い顧客管理システムの負荷が高まり、同システムにおけるデータベース処理に時間を要していることによるものであった（図表 5）。

図表 5　レイテンシ安定時間帯と悪化時間帯における顧客管理システム内の処理時間内訳



次に、データベース処理に時間を要していた要因を調査したところ、実験用システムで利用したデータベース製品に起因していることが確認できた[14]。本試験で発生した事象がそのまま社会実装時に発生するとは限らないものの、高負荷の処理を円滑に行うためには、利用するデータベース製品の性能・特性・限界を詳しく把握し、サイジングを含めたシステム構成の検討[15]、データベースのチューニング[16]、性能試験を適切に行うことが重要と考えられる。

---

[13] 一時的な処理の急増のこと。

[14] 使用したデータベース製品では、更新処理を行う際に更新内容を書き込む一時的な領域（バッファ）が使われており、更新処理によってバッファの内容がストレージに書き込まれるところ、バッファがフルになった場合には書き込み待機時間が発生するため、データベースの更新処理に時間を要することとなった。また、マネージドデータベースサービスの仕様として、更新処理を行う際には、内部的に複数のデータセンターに書き込む仕様となっているため、大量のデータ更新が発生した際にネットワークトラフィックが増加し、結果として処理時間がかかった可能性もある。いずれにせよ、1 台のデータベースサーバーに相当な高負荷がかかったことになる。

[15] DB サーバーの構成については、1.2.4 社会実装時における相当高水準な事務量へのフィージビリティでも述べる。

[16] 本試験で確認された事象についても、データベース製品の個別のチューニング（バッファサイズの調整、ネットワーク帯域の調整等）によりその影響を低減できる可能性はある。

1.2 社会実装を行ううえでの性能設計上のインプリケーション

上記２つの試験を行う過程では、いくつかのシステム上の問題に直面しており、それらを解決しながら試験を進めた。ここでは、直面した問題も踏まえつつ、社会実装を見据えた性能設計上のインプリケーションを考察する。

1.2.1 処理スパイクへの配慮

同一口座集中試験において、台帳管理システムが顧客管理システムから一時的に大量のリクエストを受信し、処理スパイクが発生した[17]ことにより、台帳管理システムにおけるレコード分割した口座の全レコードがロックされ、レコードロックの解除待ちが大量に発生（レイテンシやスループットが悪化）するという事象が発生した。一般論として処理スパイクの原因は主に、(a)リクエストの急増、(b)処理性能が高いシステムから低いシステムへの大量のリクエスト投入（システム間の処理性能の偏り）が考えられる。

(a)は、顧客管理システムや台帳管理システムへのリクエストが短時間に想定以上に集中する状況であり、これにより処理スパイクが発生し、処理負荷が急激に増大してしまうことが想定される。(b)は、顧客管理システム間、台帳管理システム間、および顧客管理システムと台帳管理システム間で、システムをまたいだ処理が行われる[18]ことから、システムごとに処理能力に偏りが存在している状況であり、特定のシステムがその処理能力を超えたリクエストを受信した場合、処理が滞ることが想定される。CBDC のエコシステムの特徴を考えると、処理能力の異なる様々な主体がシステムを運営する可能性があり、このリスクに対処する必要がある。

これら(a)(b)の事象の解決策としては、顧客管理システムや台帳管理システムにおけるリクエストの受付口（フロントサーバー等）において、単位時間あたりのリクエスト受付件数を制御することが考えられる。また、単位時間あたりのリクエスト件数の制御を行う際には、１秒あたりのリクエスト件数のみならず、業務種別による負荷の違い（参照系取引よりも更新系取引の方が高負荷）を踏まえた制御が必要となる。

---

[17] 顧客管理システムにおいてデータベース処理の瞬断が発生（原因は利用したソフトウェアの仕様によるもの）し、その後一気に顧客管理システムで処理が進んだ結果、後続の台帳管理システムに対して大量リクエストが投入されたことによるもの。

[18] 実験用システムにおいて実装した送金の処理フローは、日本銀行決済機構局、「中央銀行デジタル通貨に関する実証実験『パイロット実験』の進捗状況（2025 年 5 月）」を参照。

1.2.2 累計取引金額・回数制限チェックの性能面への影響

同一口座集中試験では、顧客管理システムにおける各種チェックによる性能面の影響を検証するため、顧客管理システムにおいて累計制限判定[19]のチェックを実装した状態での試験を行ったところ、取引件数の累積とともに累計制限判定のための処理時間が増加し、CPU 使用率が上昇した[20]。より仔細に分析したところ、取引履歴が累計で 10 万件に到達した場合は累計制限判定のチェック処理に 1 件あたり約 0.6 秒かかっており、取引件数が累計で数万件以上になると累計制限判定のチェック処理の性能への影響が無視できなくなる可能性があることを確認した。

実験用システムでは、累計制限判定を行う際に取引履歴を毎回全件集計[21]するように実装しており、事務量が多い場合や取引履歴が多い場合にこの全件集計に時間を要していることが上述のような事象が発生する主因であった。

そこで、本事象への解決策として、並列処理性向上策のメリットを活かしつつも、集計を行うデータ量を減少させるという考えのもと、以下の 2 案の机上検討を行った。

1 つ目は、取引履歴を毎回全件集計するのではなく、定期的に（例えば、夜間などシステム負荷が低い時間帯にバッチ処理により）集計しておき、当該定期集計値と、集計以降の全取引履歴を合算して累計制限判定を行う方法である。もっとも、この方法を採用した場合には、定期集計を行う頻度により、本事象と同じような性能劣化を引き起こす可能性がある点には留意が必要である。

2 つ目は、取引履歴を集計して累計制限判定を行うのではなく、上限回数・金額までの残回数・残金額を記録・更新したうえで、当該回数・金額をレコード分割したテーブルで管理する方法である。もっとも、この方法を採用した場合には、台帳管理シス

---

[19] 実験用システムでは、1 口座あたりの累計取引金額制限や累計取引回数制限（それぞれ月単位、日単位）の機能を実装している。なお、制度設計との関係でこのような制限の要否については決まっていない。

[20] 具体的には、2,400TPS の送金リクエストを行った際、取引件数の累積とともに DB サーバーの CPU 使用率が増加し、送金リクエストを送って約 15 秒経過した後に CPU 使用率が 100%に上昇し、正常な処理実行が不可能となる事象が発生した。

[21] なお、実験用システムの設計時には、「取引履歴を集計した累計値自体を保持しておき、取引ごとに当該累計値を修正したうえで、それを累計制限判定に使用する案」の検討も行った（この場合、ここで述べた性能面の劣化事象は発生しづらいと考えられる）。ただし当該案を採用した場合には、各取引の際に、累計値レコードをロックして累計値を修正する必要がある。実験用システムにおいては、並列処理性向上策（残高へのレコード分割の仕組み）を実施している中、累計値のレコードロックにより並列処理性向上策の効果を減殺する可能性があったため、当該案は採用しなかった。

テムにおけるレコード分割の場合と同様に、限界的状況において過検知[22]が発生する可能性がある点には留意が必要である。

以上のように、取引が集中する口座に対して、レコード分割による並列処理性を保ったまま累計制限判定を行う際には、事務量に応じて、性能面への影響、過検知が発生しうる。このため、累計制限判定を行う場合には、ここで述べた解決策も含め、事務量に応じて、累計制限判定の厳密性と性能のバランスを考慮した対応を検討する必要があると考えられる。

1.2.3 レコード分割に関する社会実装時のインプリケーション

同一口座集中試験において、レコード分割により 1 つの口座で 6,000TPS の処理が実現可能であることが確認できたものの、社会実装時には、より高水準の事務量を処理する必要がある可能性もある。そこで、1.1.1（２）試験の結果において言及したとおり、レコード分割数を増やし過ぎると性能劣化する可能性がある点にも留意しつつ、社会実装時の相当高水準な事務量における同一口座集中の課題に対して、レコード分割の仕組みを活用しつつ、技術的に対処することが可能か、検討を行った。

具体的には、１口座あたりの最大リクエスト件数を約 18,500TPS と仮置き[23]し、当該リクエスト件数を 1 つの口座で処理するためのレコード分割数を算出したところ（算出方法は 1.1.1（１）試験の内容を参照）、計算上は 600 分割で足りることを確認した[24]。これは、あくまで机上での評価ではあるものの、同一口座集中に対する処理性能を更に高めるうえでも、レコード分割の有効性が確認されたものと考えられる。

---

[22] 本来エラーが発生しない取引をエラーとして判定すること。

[23] 今次実験の 10 倍の事務量を 2 つの仲介機関のみで処理すると仮定し、その場合に 1 つの口座が処理する必要があるリクエスト件数を試算した。

[24] 当該分割数とした場合に必要となるシステム内サーバーの CPU 使用率を試算した。具体的には、パフォーマンスに影響を与える 3 つの要素（スループット、CPU 使用率、レコード分割数）の関係性を実験用システムにおける実験結果から 1 次近似で導出し、レコード分割数、スループットを当該近似式に代入して、CPU 利用率の想定を算出した。その結果、CPU 使用率の試算値は 56%と、特段支障のない水準となった。

1.2.4 社会実装時における相当高水準な事務量へのフィージビリティ

混合業務負荷試験では、社会実装時の想定事務量の 1/10 程度である、合計 50,000TPS（更新系取引 10,000TPS、参照系取引 40,000TPS）のリクエストが処理可能であることを実験用システムで確認した。このもとで、社会実装時における想定事務量を合計 500,000TPS（更新系取引 100,000TPS、参照系取引 400,000TPS）と一旦想定したときの、（1.2.1～1.2.3 のような個別の技術要素の観点ではなく）システムインフラ全体の観点から、技術的なノックアウトファクター（致命的で解決できない要因）の有無について検討を行った。具体的には、システムインフラを構成する主要コンポーネントである(a)AP サーバー[25]、(b)DB サーバー[26]、(c)ネットワーク・通信、のそれぞれについて、処理能力拡張のフィージビリティの検討を行った。

なお、処理能力拡張の方法としては、機器を水平に増設して処理能力を拡充する方法である「スケールアウト」と、機器自体の能力（CPU コア数等）を増強して機器単体での処理能力を拡充する方法である「スケールアップ」が存在するところ、スケールアウトが可能であれば機器を増設すればよい一方で、何らかの制約によりスケールアウトが難しい場合には、スケールアップによる能力増強が限界に到達しないかを確認する必要がある。

(a)AP サーバーについては、一般的にはスケールアウトすることで、処理能力を拡張することが技術的には可能と考えられる。AP サーバーの処理がステートレス（処理の状態を原則持たない）であることから、AP サーバーの台数を増やしても、データの全体的な整合性に影響がないためである。

(b)DB サーバーについては、データベースのシャーディング（水平分割）の考えに基づきスケールアウトすることで、処理能力を拡張することが技術的には可能と考えられる。一般的に、（実験用システムでも利用している）データベースの場合、スケールアップすることにより処理能力を拡張することが基本的な対応となるが、社会実装時における相当高水準な事務量を想定するとスケールアップが限界に到達する可能性がある。この場合には、スケールアウトを行う必要性が生じるものの、一般的なデータベースにおいては、データベースをスケールアウトして複数の DB サーバーに情報

[25] アプリケーションサーバーの略で、ビジネスロジックの制御を行うための業務アプリケーションが稼働するサーバー。

[26] データベースサーバーの略で、AP サーバーからのリクエストを受け付けて、データを格納・管理するためのデータベースが稼働するサーバー。

が分散してしまうと、それらの間のデータの整合性（一貫性）を確保することが難しくなる。この点、実験用システムは、もともと台帳管理システムを分散して構成するアーキテクチャーとしており、処理フローやその中でのメッセージのやり取りを通じて、複数DBサーバー間のデータの整合性を確保する仕組みとしている。このため、データベースのシャーディング（水平分割）の考えに基づいたスケールアウトを実質的に実装しており、データベースのスケールアウトのフィージビリティについて実験用システムを使って確認できたこととなる[27]。

(c)ネットワーク・通信については、通信制御を行うネットワーク機器や通信回線等[28]が含まれ、これらに対してスケールアウトとスケールアップを組み合わせることが基本になると考えられる。なお、社会実装時における想定事務量に対してネットワーク機器や通信回線等の構成を試算すると、スケールアウトとスケールアップで対応できることを確認した。

[27] この考え方は、台帳管理システムを（分散せずに）集中して構成するアーキテクチャーにおいて処理能力を拡張する場合にも適用可能である。もっとも、机上検討の結果における「2.5 セキュリティ」および「2.6 可用性」において整理したとおり、台帳間の不整合が生じた場合に備えた施策が必要と考えられる点には注意が必要である。

[28] 非同期通信を行うためのキューイングサービスも含まれる。顧客管理システムや台帳管理システムを、APサーバー、DBサーバー、キューイングサービスをセットとして構成することを考えた場合、キューイングサービスについては、(a)(b)同様にスケールアウトで対応することとなる。

２．机上検討の結果

2.1 送金に関する各種機能・ユースケース

ここでは、送金に関する各種機能・ユースケースについて整理する。具体的には、送金に際して重要となる宛先に関する情報を管理する方法（以下 2.1.1）について整理するほか、店舗支払い後の返金といった反対取引処理など、法人から個人への送金について整理する（以下 2.1.2）。さらに、インターネットを通じて物品やサービスを販売するような、いわゆる e コマースサイト等での利用など、非対面取引についても整理する（以下 2.1.3）。

2.1.1 宛先情報管理

CBDC の送金において、例えば、送金元は送金先の口座 ID（CBDC 口座を一意に識別するための ID）を取得し、ウォレットアプリ等に入力するなどして送金を実行することが想定される。ここでは、送金先の口座 ID に関連する機能として、海外における CBDC の検討でも取り上げられているエイリアス機能（以下（１））や、口座 ID のトークン化（トークナイゼーション）（以下（２））、送金先の顧客管理機関を特定するために必要となる宛先確認機能（以下（３））について整理する。そのうえで、これらの整理を通じて、宛先情報（送金先の顧客管理機関を一意に特定し、かつ送金先の CBDC 口座を一意に特定するために必要な情報）について中央集権的に管理するのか、分散的に管理するかについて整理する（以下（４））。

（１）エイリアス機能

個人間送金を想定すると、電話番号等の代替情報（エイリアス）を事前に口座 ID に紐づけておくことによって、送金先の電話番号等を入力するだけで送金先の口座 ID を特定し、送金を実行可能とする仕組みが考えられる。こうした仕組みは、既存の民間の送金サービスでも存在し、「エイリアス機能（またはエイリアス送金）」などと呼称される。

エイリアス機能の実現にあたってのポイントは、送金先のエイリアス（電話番号等）から宛先情報をどのように特定するか、特定するための情報をどのような形で管理する設計とするのか、ということである。エイリアスから送金先の宛先情報を特定・管理する方法として、（a）中央集権的なシステムが全ユーザーのエイリアスと口座 ID の紐づけを管理する、（b）各顧客管理システムが全ユーザーのエイリアスと口座 ID

の紐づけを管理する、が考えられる（口座 ID から顧客管理機関 ID[29]を特定する方法については別途、（３）宛先確認機能において述べる）。これらに加えて、（c）各顧客管理システムが自身のユーザーのエイリアスと口座 ID の紐づけを管理したうえで、送金時は送金先のエイリアスに加えて当該顧客管理機関 ID の情報を付加して送金する（例えば、送金時に送金元は、送金先の電話番号等および送金先の顧客管理機関 ID を入力して送金実行を指示する）、という方法もあり、全部で大きく３つの方法が考えられる。

方法（a）は、最もシンプルな解決策であり、CBDC と民間マネーとの間の相互運用性を確保する際にも有益となる可能性がある。一方で、電話番号や口座 ID 等のユーザー情報を中央集権的に管理することが適切か、どのような主体が管理すべきか、中央集権型の当該システムに障害が発生した場合の影響範囲が大きくなることにどのように対応するか、といった論点が存在する。

方法（b）は、各顧客管理機関が、自身の顧客以外のユーザーも含めた全ユーザーのユーザー情報を管理することが適切か、という論点が存在するほか、顧客管理機関の情報管理負担も大きくなる。

方法（c）は、各顧客管理機関が自身の顧客のユーザー情報のみを取り扱うという点でプライバシー面での懸念は小さい。一方で送金元は送金時に、送金先のエイリアスに加えて送金先の顧客管理機関 ID（または顧客管理機関名）の情報についても必要となる[30]。

（２）口座 ID のトークン化（トークナイゼーション）機能

個人間送金や店舗支払いを想定すると、送金元が送金先に対して、口座 ID を秘匿するニーズが生じうる。具体的には、店舗支払いを例にとると、送金元（個人）は送金先（店舗）に対して自身の口座 ID を秘匿するニーズがあると考えられる[31]。これは、プライバシーに配慮した設計（プライバシー・バイ・デザイン）を行う際の重要な要

---

[29] 顧客管理機関を一意に特定するための ID のこと。

[30] 方法(c)の派生形として、送金時には、送金先のエイリアスのみを指定し、送金先の顧客管理機関 ID の情報については、送金元の顧客管理機関が、全顧客管理機関に対してエイリアスを元に問い合わせる、という方法も考えられる。もっとも、この方法は、エイリアスを使った送金をする都度、送金元顧客管理機関から全顧客管理機関に対しての問い合わせが必要となり、システムのリソース面や性能面への影響が大きくなる可能性がある。

[31] 個人間送金でも、例えばフリーマーケット等、面識のない個人間での送金を行う局面では、送金元・送金先がお互いの口座 ID を秘匿するニーズが想定される。

素の一つとなりうる。口座 ID を秘匿する方法としては、(i)送金先（店舗）の顧客管理システムは送金元（個人）の口座 ID を把握しているものの、それを単に送金先（店舗）の端末に表示させないという簡易な方法や、(ii)そもそも、送金先（店舗）の顧客管理システムに対して送金元（個人）の口座 ID を把握させないという方法が考えられる。ここでは難易度が高いと想定される(ii)について検討を行った。なお、ここでは AML 関連の考慮は一旦捨象している。

送金先（店舗）の顧客管理システムに対して送金元（個人）の口座 ID を把握させない場合には、口座 ID を秘匿化（以下、「トークン化[32]」という）しても個人間送金や店舗支払いを成立させる仕組みを検討する必要がある。

口座 ID のトークン化の実現にあたってのポイントは、送金元（個人）と送金先（店舗）の情報連携時において、口座 ID から一意に生成されたトークン（以下、「トークン化 ID」という）について、トークン化 ID から宛先情報をどのように特定するか、特定するための情報をどのような形で管理する設計とするのか、ということである。その方法としては、トークン化 ID から送金先の宛先情報を特定するために、（a）中央集権的なシステムが全ユーザーのトークン化 ID と口座 ID の紐づけを管理する、（b）各顧客管理システムが自身のユーザーのトークン化 ID と口座 ID の紐づけを管理したうえで、情報連携時においてトークン化 ID に加えて当該顧客管理機関 ID を付加する、の大きく 2 つが考えられる[33]。

方法（a）は、（１）の方法（a）と同様、最もシンプルな解決策であり、CBDC と民間マネーとの間の相互運用性を確保する際にも有益となる可能性がある。一方で、口座 ID 等のユーザー情報を中央集権的に管理することが適切かどうか、どのような主

---

[32] クレジットカードによる決済などで利用されているデータ流出時の被害を低減するため、クレジットカード番号（PAN）を別の番号列（トークン）に変換（トークン化）する仕組みを参考に考えたもの。クレジットカードにおいては、店舗や EC サイト事業者においては本来のカード番号は保持せず、トークンのみを保持する（非保持化）。これと、トークンは特定条件でなければ利用できない仕組みとすることにより、データ流出時の不正利用被害を低減している。なお、クレジットカードによるトークンの生成は TSP（Token Service Provider）等が行う中、CBDC における口座 ID に対応するトークンの生成主体は様々な主体（中央集権的な主体や各顧客管理機関）が考えられる。

なお、「アセットのトークン化」（アセットトークナイゼーション）という言葉も存在するが、そこでのトークン化の定義は様々あるものの、例えば BIS では「金融資産や実物資産に関する権利を、プログラマブルなプラットフォームにおいて、デジタルに表章するプロセス」とされており、ここで述べるトークン化とは異なる意味で使用している。

[33] 送金先（店舗）の顧客管理システムに対して送金元（個人）の口座 ID を把握させない前提を置いて検討しているため、各顧客管理システムが全ユーザーのトークン化 ID と口座 ID の紐づけを管理する方法は捨象している。

体が管理すべきか、また、中央集権型の当該システムに障害が発生した場合の影響範囲が大きくなることにどのように対応するか、といった論点が存在する。

方法（b）は、情報連携時において顧客管理機関 ID を伝達するために、トークン化 ID と当該顧客管理機関 ID を併用する（（１）の方法（c）と同様）必要がある。

（３）宛先確認機能

CBDC の送金において、送金元は送金先の顧客管理システムに対して CBDC の決済電文を送信するために、送金先の顧客管理機関 ID を何らかの方法で特定する必要がある。

このための方法として、(i) 送金時等にユーザーが顧客管理機関 ID や顧客管理機関名の情報を付加することで、顧客管理機関 ID を特定するといった単純な方法が考えられる（（１）の方法（c）や、（２）の方法（b）で述べた内容）。また、(ii) ユーザーの口座 ID 等の中に当該ユーザーの顧客管理機関 ID を含める（例えば、口座 ID の上位桁が顧客管理機関 ID となるように口座 ID を付番する）こととすれば、口座 ID の情報のみで顧客管理機関 ID が特定可能となる。このほかの方法としては、(iii) 口座 ID と顧客管理機関 ID を紐付けたうえで、送金の際に口座 ID から顧客管理機関 ID を自動で検索するような仕組み（以下、「宛先確認機能」という）を構築する方法が考えられる。

(ii)の方法については、仮に「アカウントポータビリティ（ユーザーが顧客管理機関を変更する際に、口座 ID を不変のまま CBDC 口座を別の顧客管理機関に移設すること）」を利用することを考慮した場合[34]には不都合が生じうる。これは、ユーザーが顧客管理機関を変更した場合に、口座 ID（の中の顧客管理機関 ID）に変更が生じるためである[35]。

(iii)の方法（宛先確認機能）の実現にあたってのポイントとしては、口座 ID から顧客管理機関 ID をどのように特定するか、特定するための情報をどのような形で管理す

---

[34] 例えば、ユーザーが公共料金の引落し用口座として CBDC 口座を利用していた場合等において、ユーザーが何らかの理由で顧客管理機関を変更した際、口座 ID が不変であれば、引落し用に設定した口座 ID は変更手続きを行う必要がなくなる。欧州（デジタルユーロ）では、仲介機関選択の自由を保障しレジリエンスを高めるものとして、仲介機関の間での CBDC 口座の切り替え（switching）を確保することが想定されている。

[35] なお、各顧客管理機関が自身のユーザーの口座移設にかかる履歴の情報（移設先の顧客管理機関 ID）を管理することにより、ユーザーが CBDC 口座を別の顧客管理機関に移設しても口座 ID を不変とすることができると考えられる。もっとも、各顧客管理機関は、ユーザーの口座移設情報を長期間保持し続ける必要があるほか、口座移設の履歴分だけ顧客管理機関に問い合わせが発生し、通信回数が増加するため、システム全体への負荷が高まる懸念がある。

る設計とするのか、が想定される。その方法としては、口座 ID から顧客管理機関 ID を特定・管理するために、（a）中央集権的なシステムが全ユーザーの口座 ID と顧客管理機関 ID の紐づけを管理する、（b）各顧客管理機関が全ユーザーの口座 ID と顧客管理機関 ID の紐づけを管理する、の 2 つが考えられる。

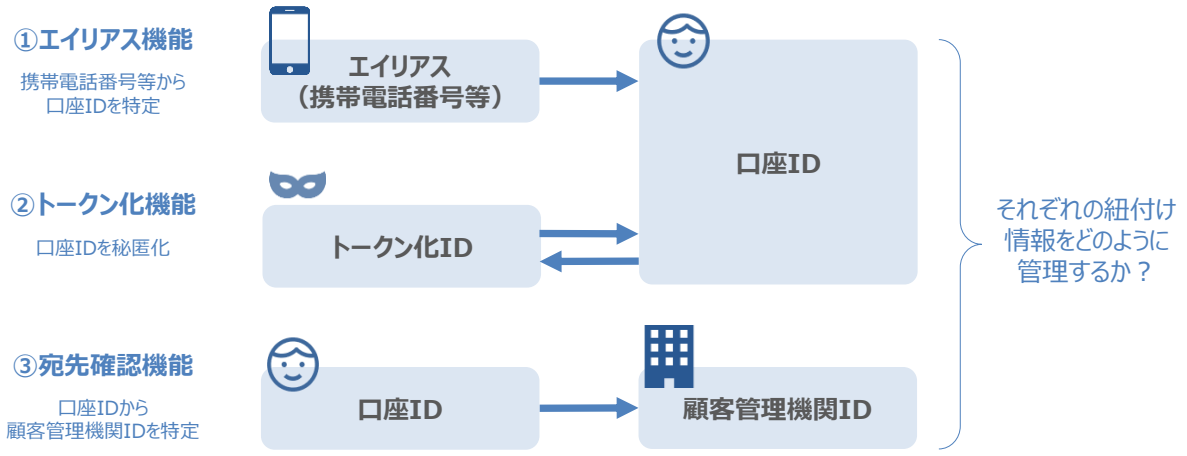
方法（a）は、最もシンプルな解決策であり、CBDC と民間マネーとの間の相互運用性を確保する際にも有益となる可能性がある。一方で、口座 ID 等のユーザー情報を中央集権的に管理することが適切か、中央集権型の当該システムに障害が発生した場合の影響範囲が大きくなることにどのように対応するか、といった論点が生じる。

方法（b）は、各顧客管理機関が、自身の顧客以外のユーザーも含めた全ユーザーのユーザー情報を管理することが適切か、という論点が存在するほか、顧客管理機関の情報管理負担も大きくなる。

（4）まとめ

以上のように、（1）エイリアス機能、（2）トークン化機能、（3）宛先確認機能、といった各機能は、「送金実行時に、送金元／送金先の宛先情報を特定するうえで必要となる、口座 ID に紐付けられた情報」という共通点がある（図表 6 を参照）。本節では、これらの紐付け情報をどのように管理することが適当か（中央集権的に管理するか、中央集権によらない形で管理するか等）、といった点を中心に改めて整理を行った。

図表 6　口座 ID と各紐付け情報との関係



（1）エイリアス機能、（2）トークン化機能、（3）宛先確認機能は、アカウントポータビリティを行うか否かで実現方法が異なる。ここで、アカウントポータビリティが必要という立場に立てば、（A）中央集権的なシステムに紐づけ情報を保有させ

る方式、（B）中央集権でない形で紐づけ情報を保有させる方式（例：各顧客管理機関において全紐づけ情報を保有）に分類でき、アカウントポータビリティは不要という立場に立てば、（C）各顧客管理機関で自身の顧客に対するエイリアスやトークン化にかかる紐づけ情報のみを保有させる方式（ただしこの場合、エイリアス、トークン化 ID、口座 ID の中に顧客管理機関 ID を含める等の対応が必要）、という大きく３通りの対応案が考えられる（図表 7）。

図表７　３機能の各対応案にかかる論点まとめ

	アカウントポータビリティを実現		アカウントポータビリティを実現しない
	（A）中央集権的なシステムに紐づけ情報を保有	（B）中央集権でない形で紐づけ情報を保有	（C）各顧客管理機関で自身の顧客に対するエイリアスやトークン化にかかる紐づけ情報のみを保有（宛先情報に顧客管理機関IDを付加）
(1)エイリアス機能	• ユーザー情報（エイリアス、口座ID等）を中央集権的に管理することの適切性 • 中央集権のシステムに障害が発生した場合の影響範囲が拡大	• 各顧客管理機関が、自身の顧客以外のユーザー情報を管理することの適切性 • 各顧客管理機関の負担	• 送金元のユーザーは、エイリアスに加えて顧客管理機関IDを指定する必要
(2)トークン化機能	• ユーザー情報（口座ID等）を中央集権的に管理することの適切性 • 中央集権のシステムに障害が発生した場合の影響範囲が拡大	―― （送金先（店舗）の顧客管理システムに対して送金元（個人顧客）の口座IDを把握させない前提を置いて検討しているため）	• 口座IDおよびトークン化IDに、顧客管理機関IDを含める等の対応が必要
(3)宛先確認機能	• ユーザー情報（口座ID等）を中央集権的に管理することの適切性 • 中央集権のシステムに障害が発生した場合の影響範囲が拡大	• 各顧客管理機関が、自身の顧客以外のユーザー情報を管理することの適切性	―― （宛先情報に顧客管理機関IDが含まれるため、宛先確認が不要）

2.1.2 反対取引処理

2025 年進捗報告書[36]では、個人間送金や、個人から店舗（法人）への支払いの処理フローを中心に整理を行った。これに加え、CBDC の社会実装時を想定すると、店舗支払い後、何らかの事情により返金を行う場合などを想定した反対取引処理として、法人から個人へ送金する仕組みについても整備することが必要となりうる[37]。ここで反対取引処理とは、(i)個人から店舗（法人）への支払いが完了した後、(ii)店舗（法人）が起動する形で(i)とは別の取引として店舗（法人）から個人に支払いを行うこと、を想定した場合の(ii)のことを指す。

[36] 日本銀行決済機構局、「中央銀行デジタル通貨に関する実証実験『パイロット実験』の進捗状況（2025 年 5 月）」、2025 年 5 月 https://www.boj.or.jp/paym/digital/dig250523b.pdf

[37] なお、法人から個人への送金のユースケースの一つと考えられる給与支払いについては、一定の保有額上限が想定される CBDC において、（仮に保有額上限を超えた金額は預金口座等にオートスウィングされることを前提とした場合でも）どこまでユースケースとして想定すべきか、といった論点がありうる。

ここでは、プライバシー保護の観点（送金先である個人の情報を必要以上に送金元である店舗（法人）に連携しないニーズが存在）も含めた実現方法について、基本形（以下（１））とその派生形（以下（２））を説明する。

（１）店舗支払いにおける反対取引処理（店舗（法人）から個人への送金）

店舗支払いにおける反対取引処理については、基本形として、上記(i)(ii)という流れが考えられる中、(ii)に関しては、(i)の決済内容（個人から店舗（法人）への CBDC の送金）を何らかの方法で特定し、(i)と紐づけた形で反対取引処理（店舗（法人）から個人への CBDC の送金）を行うことが想定される。ここで、(i)の決済内容の特定については、取引 ID[38]の活用が想定される。

上述の基本形における反対取引処理は、例えば、(ii-1)個人は反対取引処理に際して、(i)の取引で利用した取引 ID を店舗（法人）に伝達したうえで、(ii-2)店舗（法人）は店舗用のアプリ等で自身の CBDC の取引履歴を表示し、反対取引処理をしたい取引の取引 ID で検索を行って(i)の取引を特定のうえ、(i)の取引に対する反対取引処理を起動する（店舗（法人）が個人に CBDC を送金する）[39]、という流れで行うことが想定される。この際には、店舗（法人）に対して送金元の個人の口座 ID を開示せずに、CBDC システム内において取引 ID から送金元、送金先を特定したうえで、送金を行う仕組みが必要となりうる。

なお、当初の取引である(i)の取引 ID に紐づけて反対取引処理を行うことで、個人および店舗（法人）がウォレットアプリ等で取引履歴を確認する際に、当初取引(i)と反対取引処理(ii)とを紐づけて表示することが可能となる。

（２）店舗支払いにおける反対取引処理（店舗（法人）から個人への送金）の派生形

次に、反対取引処理の派生形として、(i)個人は店舗（法人）支払いを CBDC 以外の方法で行い、その後に(ii)店舗（法人）が当該支払いへの反対取引処理を CBDC で実施する、といったケースについて考える。この場合、CBDC システム内で当初の取引(i)と反対取引処理(ii)との紐付けを行うことはできないことから、単に店舗（法人）から

---

[38] CBDC システム内の取引を一意に特定するための ID のこと。

[39] 反対取引処理の実行時には、当初取引の全額を反対取引処理する「全額反対取引処理」のほか、当初取引の一部金額のみを反対取引処理する「一部反対取引処理」が考えられるが、いずれも同様のオペレーションで処理が可能である（一部反対取引処理の場合は、店舗側で反対取引額を何らか入力するような形となる）。

個人への CBDC の送金について考えればよいことになる。もっとも、個人の情報を必要以上に店舗（法人）に連携しないというプライバシー保護の観点からは、個人の口座 ID を秘匿したまま店舗（法人）が個人に送金を行う方法を検討する必要がある。ここでは実現方法の一例として、2.1.1（２）口座 ID のトークン化（トークナイゼーション）機能でも述べた、口座 ID のトークン化を用いた方法について紹介する。

口座 ID のトークン化については、中央集権的なシステムまたは各個人の顧客管理機関がトークン化を行うことが考えられるが、ここでは（より複雑な事例である）各個人の顧客管理機関がトークン化を行う方法（2.1.1（２）口座 ID のトークン化（トークナイゼーション）機能の方法（b））を例にとって説明を行う。まず店舗（法人）は、何らかの方法で「トークン化 ID」および「個人の顧客管理機関 ID」を取得する必要がある。この点、例えば個人がウォレットアプリを起動し、個人の顧客管理システムにおいてトークン化 ID の生成を行い、同トークン化 ID および個人の顧客管理機関 ID を、個人のウォレットアプリ上で表示（顧客管理機関 ID はトークン化されない想定）し、店舗（法人）がそれら情報を読み取るという流れが考えられる[40]。これにより、店舗（法人）は個人のトークン化 ID および個人の顧客管理機関 ID を用いて CBDC の送金を実施し、電文を受け取った個人の顧客管理システムがトークン化 ID を個人の口座 ID に変換することで、当該個人の CBDC 口座に着金することとなる[41]。

2.1.3 EC 決済（e コマース）

2025 年進捗報告書では、個人から法人への CBDC の送金として、店舗支払い等の対面取引を中心に整理を行った。これに加えて、CBDC の社会実装時を想定すると、インターネットを通じて商品やサービスを販売するような、いわゆる e コマースサイト（以下、「EC サイト」という）等での利用など、非対面取引についての仕組みの整理が必要となりうる。

---

[40] 非対面であれば、トークン化 ID 等を Web サイト上で入力する方法や、電話や電子メール等で伝達するといった方法が考えられる。

[41] 前述（１）店舗支払いにおける反対取引処理（店舗（法人）から個人への送金）における、反対取引処理の基本形（当初取引として CBDC の送金が行われ、それへの反対取引処理を実施するケース）においても、当初取引である個人から店舗（法人）への CBDC の送金時にトークン化 ID（および顧客管理機関 ID）が使われていれば、反対取引処理では同じトークン化 ID を宛先として反対取引処理を実施することとなる。

ここでは、EC 決済の特徴を整理したうえで（以下（１））、CBDC を使った EC サイトにおける EC 決済の処理フローについて、現行の EC サイトにおける決済の仕組みを参考にしつつ各種方式について検討を行ったほか、不正利用対策の方法についても考察を行った（以下（２））。

（１）EC 決済の近年の特徴

既存のキャッシュレス決済を念頭に、EC 決済における近年の特徴を整理する。この点、「クレジットカード決済システムのセキュリティ対策強化検討会報告書[42]」によると「クレジットカードの不正利用被害総額は近年増加傾向にあり、このうち、クレジットカード番号等の盗用の割合が 94%を占めており、主に非対面取引でのクレジットカード番号等のなりすましによる不正利用が主要な要因である」とされている。

この状況を踏まえると、EC 決済といった非対面取引においては、認証情報の不正利用対策や、取引の正当性をしかるべき主体が適切に確認することが重要となり、CBDC における EC 決済においてもこの点について十分な検討が必要となる。

（２）EC 決済の処理フロー

CBDC の EC 決済について検討するにあたり、既存の EC サイトにおける決済方式も参考に、CBDC による EC 決済の処理フローとして考えられる一例を整理する。

処理フローの整理の際には、「誰から（個人の口座 ID）、誰に（EC サイトの口座 ID およびそれに紐づく EC サイト名）、いくら（金額）」支払うという 3 つの情報が必要となり、それら 3 つの情報（取引情報）を送金元（個人）の顧客管理システムにどのように正しく伝えるかという点がポイントとなる。

また、特に EC 決済などの非対面取引においては、送金先（EC サイト）側で、各販売情報（商品・サービスの販売に関する情報で、ここでは「販売管理 ID」により管理されると仮定した）と送金による各決済との突合（消し込み）を行うための仕組みが必要となりうる[43]点もポイントとなる。

---

[42] 経済産業省、「クレジットカード決済システムのセキュリティ対策強化検討会報告書」、2023 年 1 月、https://www.meti.go.jp/shingikai/mono_info_service/credit_card_payment/pdf/20230120_1.pdf

[43] 例えば、①送金先（EC サイト）上で各販売情報に対応する販売管理 ID を生成したうえで、送金先（EC サイト）または送金元（個人）が当該 ID を電文に付加し、送金元（個人）が当該 ID 情報も含めて送金指示を行う、②各販売情報（販売管理 ID）と送金先（EC サイト）の口座 ID の両方に一意に紐づく ID（バーチャル口座 ID と呼称する）を送金先（EC サイト）の顧客管理システムで生成し、送金元（個人）は当該バーチャル口座 ID を疑似的な口座 ID として送金指示を行う（当該バーチャル

(a) 既存の EC サイト決済を参考にした方式（口座 ID 入力＋支払依頼型[44]）

本方式は、送金元（個人）が、送金先（EC サイト）において商品・サービスの購入画面を通じて、自身の情報を当該サイトに入力し、送金先（EC サイト）から還流された情報をもとに決済する方式である（処理フローは、図表 8 を参照）。具体的には、送金元（個人）は、送金先（EC サイト）の決済画面で決済手段（CBDC）を選択し、送金元（個人）の口座 ID を入力（図中❶）した後、送金先（EC サイト）では、当該情報に加えて、送金先（EC サイト）の口座 ID や金額、販売管理 ID といった取引情報を付加する（図中❷）。当該取引情報は、送金先（EC サイト）から同顧客管理システム、送金元（個人）の顧客管理システム、同ウォレットアプリ等へと連携される。この際、送金先（EC サイト）の顧客管理システムは、送金先（EC サイト）の口座 ID をもとに送金先（EC サイト）名（屋号）を検索[45]のうえ、その結果を連携する（図中❸）。その後、送金元（個人）は、ウォレットアプリ等で「追加的な認証行為（取引認証）」を行ったうえで（図中❹）、取引情報[46]をもとに送金を実行する（図中❺）流れである。

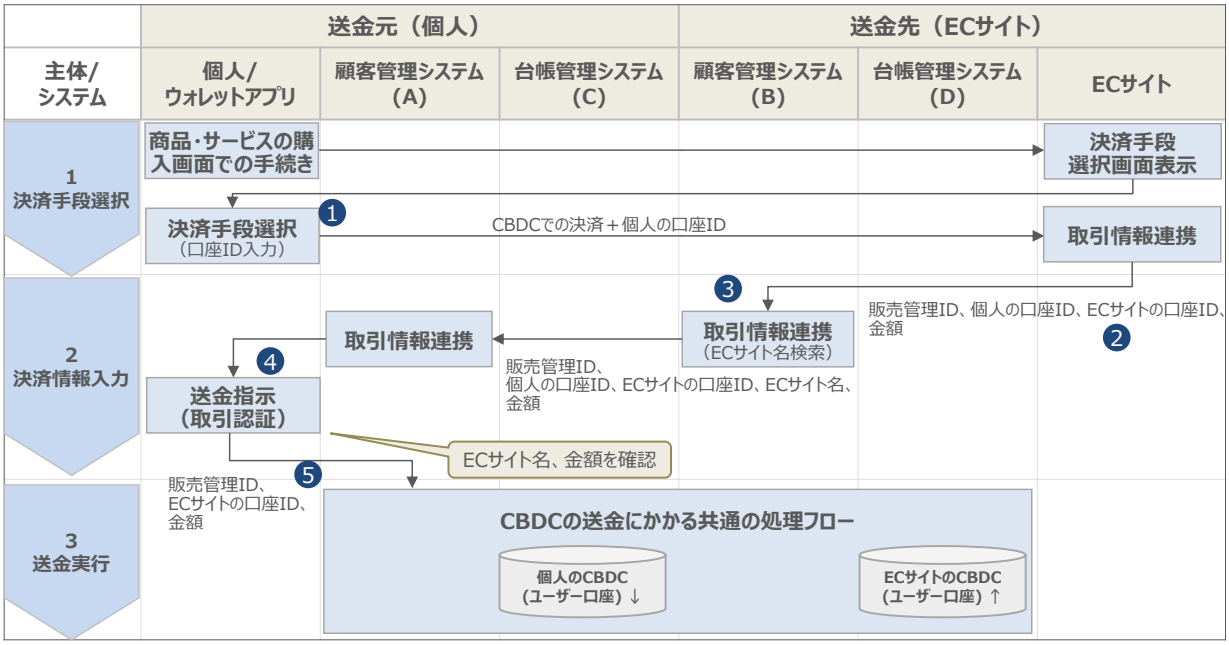
口座 ID は各販売情報と 1 対 1 で紐づくため、EC サイト運営者側での突合が容易になる）、といった方法が考えられる。ここでは、①の方法を説明している。

[44] 既存の EC サイト決済には、口座 ID に相当する情報（クレジットカード番号等）を EC サイト上に入力して決済を行う方式もある中、ここではそれに加えて、後述する取引認証を行うために送金先（EC サイト）から送金元（個人）に対して支払依頼およびその確認を行う処理を追加している。

[45] 送金先（EC サイト）の顧客管理システムは、送金先（EC サイト）の口座開設時等に、送金先（EC サイト）の口座 ID と送金先（EC サイト）名の紐づけの情報を管理しておく。そのうえで、送金先（EC サイト）の顧客管理システムは、当該紐づけの情報をもとに、送金先（EC サイト）の口座 ID から送金先（EC サイト）名を検索する。これにより、送金先（EC サイト）の口座 ID と送金先（EC サイト）名の紐づけに関する正確な情報を、送金元（個人）が把握することができ、他社の EC サイト名を名乗られること（フィッシング）による不正が発生する可能性を低減させる。この考え方は、以降の(b)～(d)の方式についても同様。

[46] 送金先（EC サイト）の口座 ID や金額に加えて、販売管理 ID の情報も送金時に連携することにより、上述したとおり、送金先（EC サイト）側で、各販売情報と送金による各決済との突合（消し込み）が可能となる。この考え方は、以降の(b)～(d)の方式についても同様。

図表 8　既存の EC サイト決済を参考にした方式（口座 ID 入力＋支払依頼型）の処理フロー



図表 8 では、「追加的な認証行為」の例として、送金元（個人）の顧客管理システムが、送金元（個人）に対して送金先（EC サイト）名や金額等を提示し、送金元（個人）が当該情報をウォレットアプリ等で確認したうえで承諾（取引認証[47]）する方法を想定した。取引認証を行うことにより、（ウォレットアプリ等にログインする際の当人認証が正しく行われているという前提において）取引内容が改ざんされておらず、送金元（個人）の意思に基づいた取引情報であるということを確認することができると考えられる[48]。

その際、前提となる「当人認証が正しく行われている」ということをより強固にするための施策として、送金元（個人）の顧客管理システムが、送金元（個人）に対し、追加的にワンタイムパスワード等の情報を求めることも考えられる。これは、現行のクレジットカードによる決済において、なりすましリスクへの対策として、EC サイト

[47] ここでは「誰から、誰に、いくら支払う」という取引情報が、送金元（個人）の意思に基づくものであることを、送金元（個人）の顧客管理機関が確認する行為、として検討する。また、その前提として送金元（個人）がウォレットアプリへのログインする際に、正しく当人認証が行われている前提で検討する。当人認証については、2.5.2 不正アクセス・不正取引への対策でも述べる。

[48] 取引のリスク度合いによらず、口座 ID のみの情報だけで送金を実行することは不適当と考えられるため、ここでは何らかの「追加的な認証行為」が必要になるものと想定した（この点、例えばクレジットカードの場合は、カード番号に加えてセキュリティコードを入力させるため、取引のリスク度合いが低い場合は追加的な当人認証が省略されることがある）。

の運営者等に対して「EMV 3-D セキュア[49]」と呼ばれる当人認証の仕組みの導入を原則必須としており、それと同様に考えたものである。

さらに、送金元（個人）の情報を必要以上に送金先（EC サイト）に連携しないというプライバシー保護の観点からは、2.1.1（2）口座 ID のトークン化（トークナイゼーション）機能でも述べたとおり、送金元（個人）の口座 ID のトークン化が有用となりうる。

(b) 既存の EC サイト決済を参考にした方式（ウォレット遷移型）の処理フロー

本方式は、送金元（個人）が、送金先（EC サイト）において商品・サービスの購入画面を通じて表示されたリンクをクリックすることで、ウォレットアプリが取得した取引情報をもとに決済する方式である（処理フローは、図表 9 を参照）。具体的には、送金元（個人）はスマートフォン上で、送金先（EC サイト）の決済画面で決済手段（CBDC）を選択し、送金元（個人）の顧客管理機関 ID を選択する（図中❶）。送金先（EC サイト）は、その顧客管理システムを経由して、送金元（個人）の顧客管理システムに対して、リンク[50]の生成を依頼するほか、それを受けた送金元（個人）の顧客管理システムは、販売管理 ID をもとにリンクの生成を行い、販売管理 ID、送金先（EC サイト）の口座 ID、送金先（EC サイト）名、金額等の情報を保存しておく（図中❸）。この際、送金先（EC サイト）の顧客管理システムは、送金先（EC サイト）から受領した口座 ID をもとに送金先（EC サイト）名を取得し、その結果を電文に付加して送金元（個人）の顧客管理システムに連携する（図中❷）。その後、送金元（個人）が送金先（EC サイト）の画面に表示されたリンクを押下した後、ウォレットアプリが自動起動し（図中❹）、送金元（個人）の顧客管理システムは、販売管理 ID をもとに、送金先（EC サイト）名、金額等の情報を取得し、それらを送金元（個人）のウォレットアプリへと連携する（図中❺）。その後、送金元（個人）は、(a)で述べ

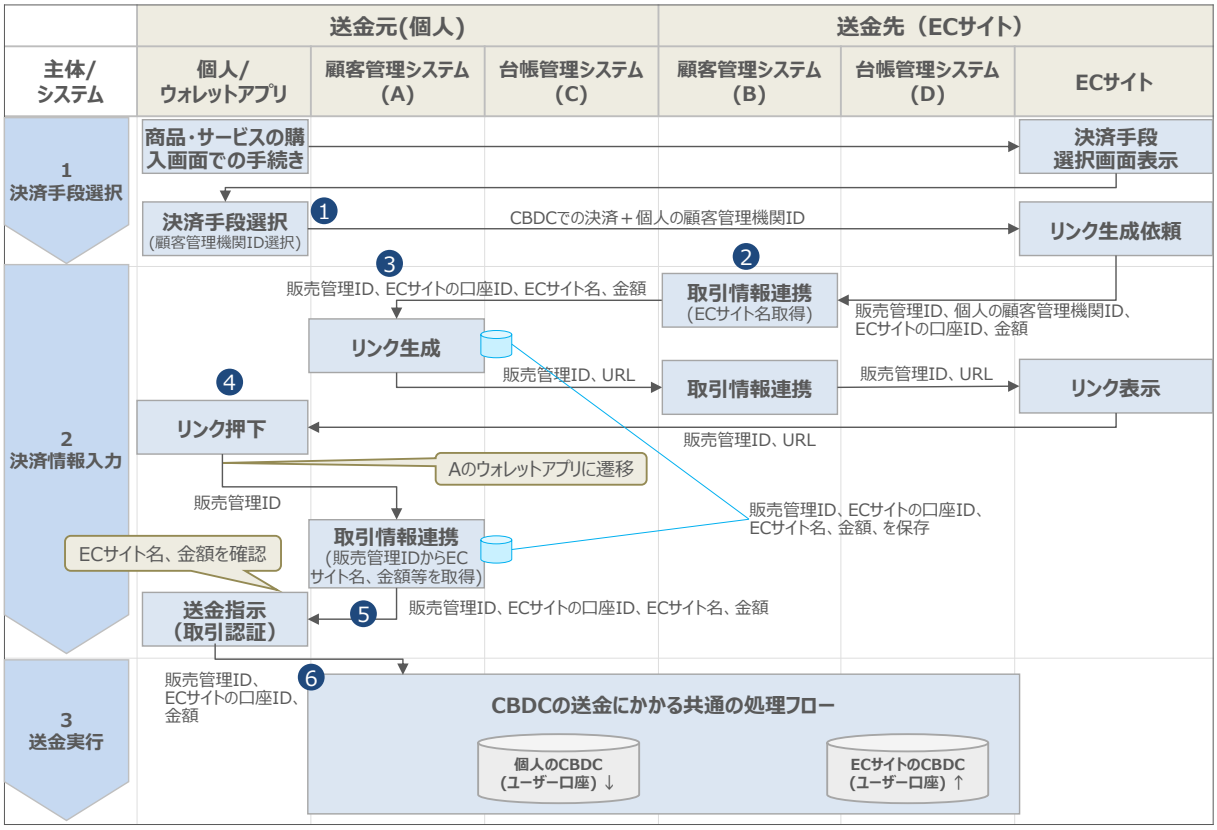
---

[49] クレジットカードを使った取引の実行時に（カード番号やセキュリティコード等のカード情報に加えて）取引のリスク度合いに応じてワンタイムパスワードや生体認証を用いた当人認証を行う仕組みを指す。

[50] 当該リンクには、販売管理 ID と送金元（個人）の顧客管理システムのウォレットアプリを起動するための URL が埋め込まれており、送金元（個人）の顧客管理システムが生成することを想定した処理フローとしている。これは、当該リンクの中に、送金元（個人）の顧客管理システムのウォレットアプリを起動する設定が埋め込まれていることを踏まえ、ウォレットアプリの各種仕様変更に柔軟に対応できるようにする観点から、（送金先（EC サイト）の顧客管理システムがリンクを生成するのではなく）ウォレットアプリの提供者たる送金元（個人）の顧客管理システムがリンクを生成する処理フローが適当と考えたためである。

た「追加的な認証行為（取引認証）」を行ったうえで取引情報をもとに送金を実行する（図中❻）流れである。

図表 9　既存の EC サイト決済を参考にした方式（ウォレット遷移型）の処理フロー

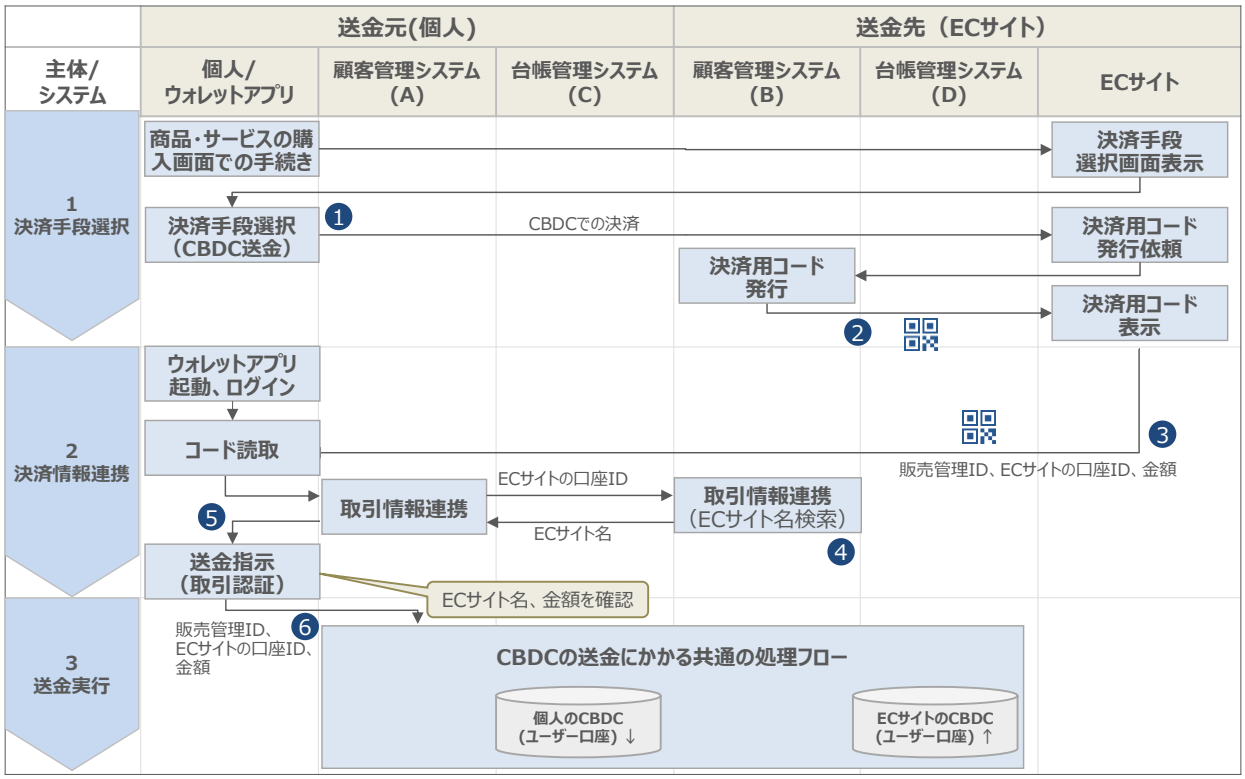


(c) 動的 MPM を活用した方式の処理フロー

本方式は、送金先（EC サイト）上に表示された 2 次元コードを、送金元（個人）がウォレットアプリで読み取って決済を行う方式である（処理フローは、図表 10 を参照）。具体的には、送金元（個人）は、送金先（EC サイト）の決済画面で決済手段（CBDC）を選択（図中❶）した後、送金先（EC サイト）は、その顧客管理システムに 2 次元コードの発行（同コードの中には送金先（EC サイト）の口座 ID、金額、販売管理 ID 等の情報が含まれる）を依頼し、発行された 2 次元コードを同サイト上に表示する（図中❷）。送金元（個人）のウォレットアプリは、搭載スマートフォン等のカメラ機能を用いて当該 2 次元コードを読み取り（図中❸）、同コードに含まれる送金先（EC サイト）の口座 ID をもとに、送金元（個人）の顧客管理システムを経由して、

送金先（EC サイト）顧客管理システムに対して、送金先（EC サイト）名を検索する[51]（図中❹）。その後、送金元（個人）は、ウォレットアプリで「追加的な認証行為（取引認証）」を行ったうえで（図中❺）取引情報をもとに送金を実行する（図中❻）流れである。

図表 10　動的 MPM を活用した方式の処理フロー



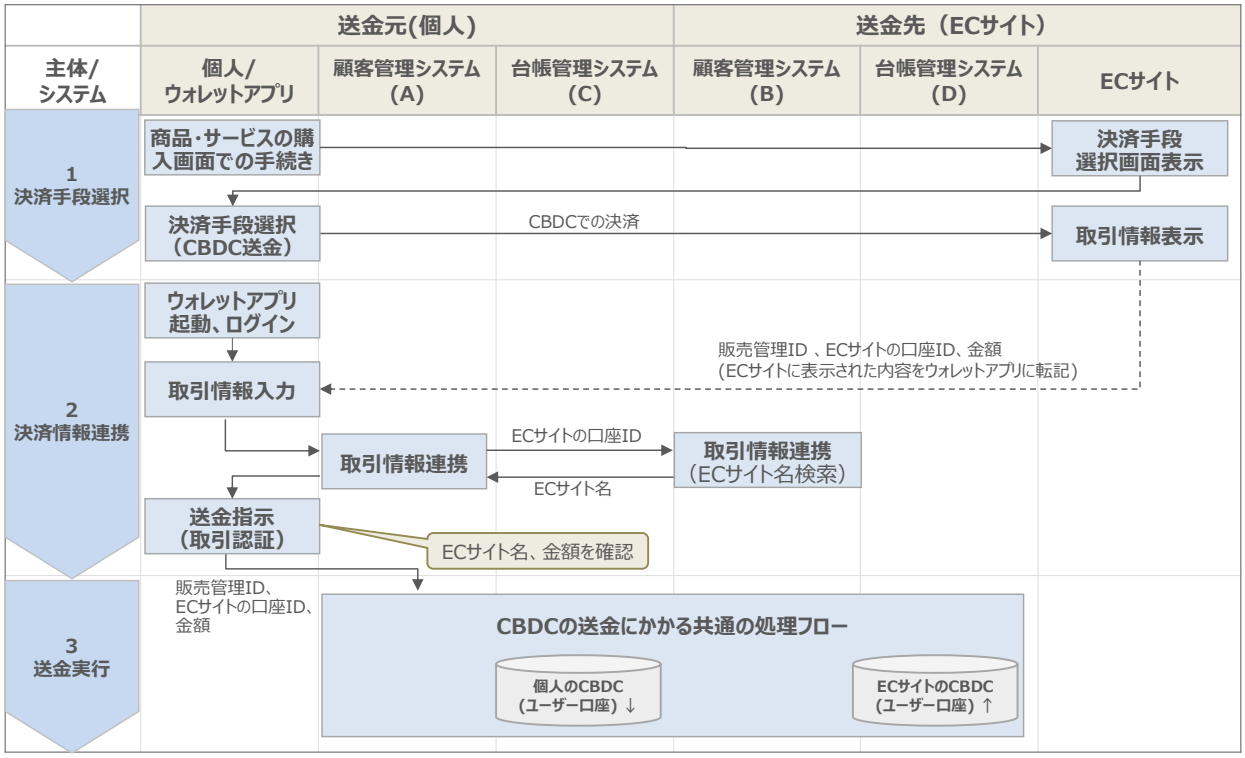
(d) 銀行振込を参考にした方式

本方式は、既存の銀行振込と同様、送金元（個人）は、EC サイトの口座 ID と金額等を EC サイト上の記載等から取得のうえ、ウォレットアプリ等を通じて送金を実行する方式である（処理フローは、図表 11 を参照）。

この場合は、送金元（個人）が送金先（EC サイト）の口座 ID と金額を手入力することになる。

[51] フィッシングサイトにおいて偽の 2 次元コードを表示する可能性までを想定すると、EC サイト上の 2 次元コードから店舗名を取得すべきではなく、2 次元コードの口座 ID 情報から別途の仕組みにより店舗名を検索して表示する方法が望ましい。

図表 11　銀行振込を参考にした方式の処理フロー



2.2 送金の処理フローの見直し

ここでは、発生したエラーに対処するための機能と適切な処理フローについて整理する。具体的には、送金処理の過程で複数の主体（送金元の顧客管理システム、台帳管理システム、送金先の顧客管理システムおよび台帳管理システム）が関与することを想定したうえで、何らかのエラーが発生し送金処理が停止した場合においても、ユーザーが不必要に待たされない仕組み（タイムアウト管理機能[52]）や、エラーが発生して送金処理が中途半端になった場合においても、適切に元の状態に戻す仕組み（留保機能）について整理を行う（以下 2.2.1）。そのうえで、実験用システムで構築した送金の処理フローに組み込むことでどのような改善が期待できるか、同処理フローのバリエーション、各処理フローの特徴をもとにした比較評価を机上で実施した内容を紹介する（以下 2.2.2）。

2.2.1 タイムアウト管理機能と留保機能の整理

タイムアウト管理は、個々のアプリケーションや機器、サブシステムなど、様々なコンポーネントを通じ一体的に設計・実装されることにより、システム全体としての適切な管理が実現する。もっとも、ここでは台帳管理システムにおいて求められるタイムアウト管理に着目し、検討を行った。

タイムアウト管理の目的は、図表 12 のとおり、主に(a)台帳間の整合性確保、(b)ユーザビリティ確保、の 2 つが考えられる。なお、ここでいう「台帳間の整合性」とは、例えば太郎から花子へ送金を行う際に、太郎の CBDC 台帳の減額と花子の CBDC 台帳の増額が一体であることを指す。一方、太郎の CBDC 台帳の減額のみが実施され、花子の CBDC 台帳の増額が行われない（その逆の状態も含む）という状態が一定時間継続した場合、「台帳間が不整合である」という。

[52] 設定時間を超えるとタイムアウトエラーとし、送金処理を中断・キャンセルする機能のこと。

図表 12　タイムアウト管理の目的・用途および想定される設定時間

目的	用途	設定時間
(a)台帳間の整合性	決済処理中に何等かのトラブルにより送金先台帳管理システムの処理状況が不明になった際などに、送金元台帳管理システムが処理中の決済を取り消すことで、台帳間の整合性を確保する。	～数秒
(b)ユーザビリティ	店舗支払いなど即時性が求められるユースケースにおいて、何等かのトラブルによって決済処理に一定以上の時間を要した場合に、処理の失敗をユーザーに通知することで、利用者が処理を待ち続けないようにする。	～数秒、数十秒
	即時性よりも確実な着金が求められるユースケースにおいて、時間的な間隔を空けたリトライによって、短時間のトラブルが生じても決済処理が失敗しないようにする。	～数時間

設定するタイムアウトの時間については、「(a)台帳間の整合性」の観点では台帳間が不整合である時間を極力短くする観点からタイムアウト時間を短く設定することが考えられる一方、「(b)ユーザビリティ」において即時性よりも確実な着金が求められるユースケースの場合、タイムアウト時間を長く設定することも考えられる。このように、タイムアウト管理を行うにあたっては、目的にあわせてタイムアウト時間を柔軟に設定できる設計とすることが望ましいと考えられる。

なお、以下の送金の処理フローの説明では、送金元を太郎、送金先を花子（太郎から花子への送金の事例）としたうえで、太郎の顧客管理システムを A、花子の顧客管理システムを B、太郎の台帳管理システムを C、花子の台帳管理システムを D と呼称して説明する。

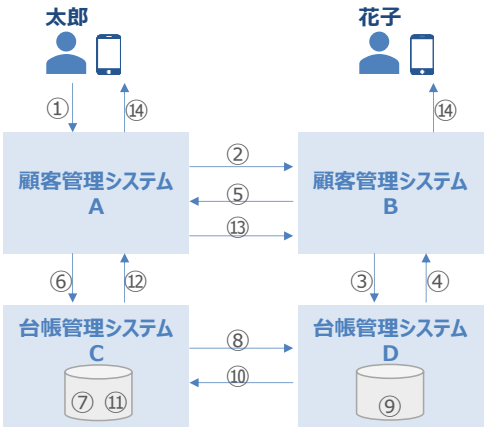
実験用システムで実装した送金の処理フロー（図表 13、タイムアウト管理機能は実装していない）では、ファイナリティ時点[53]を花子側の増額が完了した時点とした（図中⑨）。この点、本来は、「太郎の CBDC 台帳の減額と花子の CBDC 台帳の増額」をアトミック[54]に実施することで決済が完了することが望ましい。しかしながら、実験用システムは、太郎の CBDC 台帳と花子の CBDC 台帳がそれぞれ異なる主体によって管理されうる前提で構築（独立したシステムとして構築）したため、太郎の減額と花子

[53] ここでは、システム面でこれ以降は巻き戻しを行わない時点のことをいう。ファイナリティ時点より前では、処理を巻き戻すことで処理のキャンセル（処理がなかったことにする）ができる一方、ファイナリティ時点より後では、処理を巻き戻すことはできない。

[54] 2 つ以上の処理を一括して全て実行するか、仮に実行できなかった場合に、全く実行されない（片方だけ実行されるということが起こらない）か、のいずれかとなる一連の処理のことを指す。

の増額のタイミングが異なる（太郎の CBDC 台帳の減額と花子の CBDC 台帳の増額をアトミックに実施する難易度が高い）ことを踏まえた設計を行う必要が生じた。こうしたことから、図中⑦における太郎の CBDC 台帳の減額は留保付きで実施し、花子の CBDC 台帳の増額が成功すれば留保を解除し（図中⑪）、一方で花子の CBDC 台帳の増額が失敗すれば留保前に戻す設計とした。

図表 13　実験用システムで実装した太郎から花子への送金の処理フロー



#	行為者	処理内容
①	太郎	顧客管理Aに送金指示
②	A	送金先（花子）を特定し顧客管理Bに送金を通知
③	B	台帳管理Dに台帳更新許可トークン*を発行依頼
④	D	顧客管理Bに同トークン*を発行
⑤	B	顧客管理Aに了解を返送
⑥	A	台帳管理Cに送金指示
⑦	C	太郎の台帳を留保付きで減額
⑧	C	台帳管理Dに増額指示
⑨	D	③と⑧のトークン*を突合し、花子の台帳を増額 **（ファイナリティ時点）**
⑩	D	台帳管理Cに完了を通知
⑪	C	太郎の台帳の留保を取り減額を確定
⑫	C	顧客管理Aに決済完了通知を送信
⑬	A	顧客管理Bに決済完了通知を送信
⑭	A, B	太郎宛、花子宛それぞれに決済完了通知を送信

*台帳更新許可トークン：ここでは台帳管理システムが自らの台帳の更新を許可していることを示すための証明書の意味としてトークンという言葉を用いている
当該トークンは台帳管理Dにて生成され、④⑤⑥⑧の電文に付加される。
※各通信は、非同期による通信を行うことで、サーバーのリソースを効率的に活用し処理の効率性を高めている。

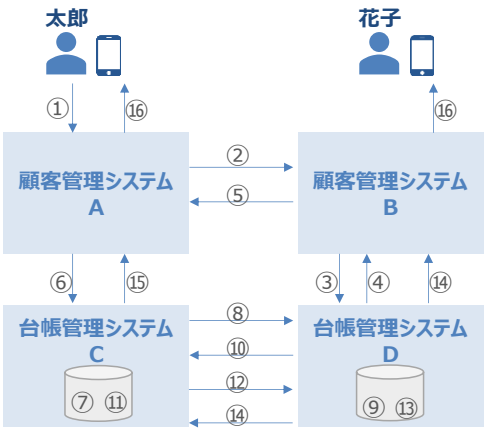
仮に、タイムアウト管理機能をこの送金の処理フローへ組み込む場合、C がタイムアウト管理の主体となり、A からの送金指示（図中⑥）を受け取ってから、完了通知（図表⑫）または失敗した旨の通知を A に送信するまでの時間を管理（成否を含めた処理時間に対してタイムアウトを管理）することが考えられる。

しかしながら C は、図中⑧の後、D との通信等にトラブルが発生したことにより図中⑩の電文を受け取れない場合、D の増額が成功したか不明な状態となる。その状態で、タイムアウト時間が到来した場合、C の減額留保状態を元に戻す（減額を行わないこととする）のか、それとも留保を解除して処理を先に進める（減額を確定させる）のか判断ができない状態となり、台帳間が不整合となる可能性が生じる[55]。

[55] 仮に、タイムアウト時間が到来したことをもって送金に失敗したと判断し減額を留保前に戻した場合、実は D の CBDC 台帳の増額が成功していると、台帳間の不整合（C の CBDC 台帳は減額が行われず、D の CBDC 台帳のみが増額した状態）が発生することになる。

このような台帳間の不整合を防止する方法の一つとして、台帳の留保機能を活用しながら、C において送金の成否を判断できるようにする、即ち、C でファイナリティ時点が生じるようにしたうえで、図表 14 に示す処理フローのように、D の CBDC 台帳の増額（図中⑨）を留保付きで行うことが考えられる。

図表 14　増額を留保付きで行う場合の処理フロー



#	行為者	処理内容
①	太郎	顧客管理Aに送金指示
②	A	送金先（花子）を特定し顧客管理Bに送金を通知
③	B	台帳管理Dに台帳更新許可トークンを発行依頼
④	D	顧客管理Bに同トークンを発行
⑤	B	顧客管理Aに了解を返送
⑥	A	台帳管理Cに送金指示
⑦	C	タイムアウト管理を開始し、太郎の台帳を留保付きで減額
⑧	C	台帳管理Dに増額指示
⑨	D	④と⑧のトークンを突合し、花子の台帳を留保付きで増額
⑩	D	台帳管理Cに完了を通知
⑪	C	タイムアウト管理を終了し、太郎の台帳の留保を取り減額を確定 **（ファイナリティ時点）**
⑫	C	台帳管理Dに完了を通知
⑬	D	花子の台帳の留保を取り増額を確定
⑭	D	顧客管理B、台帳管理Cに決済完了通知を送信
⑮	C	顧客管理Aに決済完了通知を送信
⑯	A, B	太郎宛、花子宛それぞれに決済完了通知を送信

具体的な処理フローとしては、C は、A からの送金指示（図中⑥）を受け取った時点からタイムアウト管理を開始し、C では留保付きで太郎の CBDC 台帳を減額する（図中⑦）。D では、C から増額指示（図中⑧）を受け取り、留保付きで花子の CBDC 台帳を増額する（図中⑨）。C は、タイムアウト時間内に D から通知（図中⑩）を受け取った場合、太郎の CBDC 台帳の減額の留保を解除して減額を確定（ファイナリティ時点：図中⑪）し、D に完了を通知する（図中⑫）。D は、花子の CBDC 台帳における増額の留保を解除して増額を確定させる（図中⑬）。

一方で、何らかのエラーが発生した場合における処理フローは次のとおりとなる。

図中⑩の部分において、C が、D から通知を受け取る前に、何らかのトラブルによりタイムアウト時間が到来したと判断した場合、C は、太郎の CBDC 台帳を留保前の状態に戻した（太郎の CBDC 台帳で減額をキャンセルした）うえで、D に対して花子の CBDC 台帳を留保前の状態に戻す（花子の CBDC 台帳で増額をキャンセルする）ための通知を行う。

この通知に関して重要なことは、C と D との間で通信障害等といった何らかのトラブルが発生している場合は、C が D に対して通知を行ったとしても、それが即座に D

に到達しない可能性があるという点である。その場合は、通信障害等が復旧してから、C からの通知に基づき、D が花子の CBDC 台帳を留保前に戻すことになる。通信障害が継続している間、D は花子の CBDC 台帳の増額を留保し続けることになるものの、留保している資金は他の取引に使用されない。このため、「通信障害により太郎の CBDC 台帳の減額はキャンセルされたが、花子の CBDC 台帳の増額は実行され、花子の増額した資金が他の取引に使用された結果、元の取引（太郎から花子への送金）のキャンセルが難しくなる」といった事態は発生しない。

2.2.2 送金の処理フロー別案

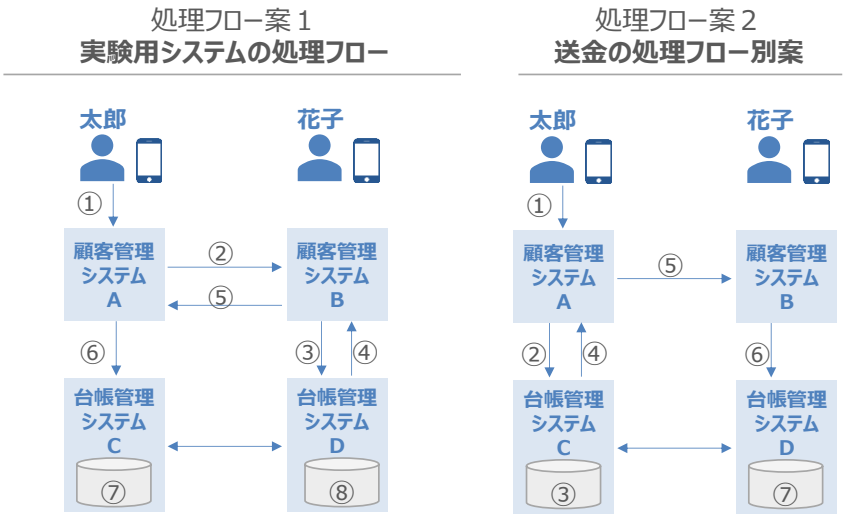
実験用システムにおいて導入した送金の処理フローに関し、より効率的なものになりうる別案の検討を行った。

すなわち、実験用システムは、送金の大きな流れとして、まずは CBDC の送金元・送金先双方の顧客管理システム（A、B）で処理をした後、双方の台帳管理システム（C、D）で処理をするものであったところ、別案では、CBDC の送金元の顧客管理システム（A）と台帳管理システム（C）で処理をした後、送金先の顧客管理システム（B）と台帳管理システム（D）が処理をするものについて検討を行った。

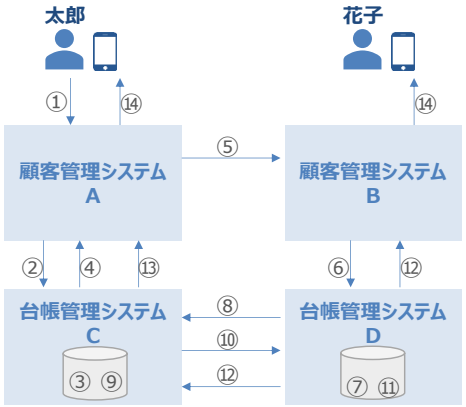
検討にあたっては、実験用システムの処理フローに関し、実験用システムには実装していないタイムアウト管理機能と増額留保[56]も実装したと仮定した処理フロー（ここでは「処理フロー案１」という）としたうえで、別案（ここでは「処理フロー案２」という）との比較を行った。

[56] それぞれの機能の詳細については、2.2.1 タイムアウト管理機能と留保機能の整理において議論。

図表 15　送金の処理フロー別案



図表 16　処理フロー案 2 における送金の処理フロー



#	行為者	処理内容
①	太郎	顧客管理Aに送金指示
②	A	送金先（花子）を特定し台帳管理Cに送金指示
③	C	太郎の台帳を留保付きで減額し、台帳更新許可トークンを発行
④	C	顧客管理Aに送金準備の完了を通知、同トークンもあわせて通知
⑤	A	顧客管理Bに送金指示
⑥	B	台帳管理Dに送金指示
⑦	D	花子の台帳を留保付きで増額
⑧	D	台帳管理Cに減額の確定を指示
⑨	C	③と⑧のトークンを突合し、太郎の台帳の留保を取り減額を確定 **（ファイナリティ時点）**
⑩	C	台帳管理Dに完了を通知
⑪	D	花子の台帳の留保を取り増額を確定
⑫	D	顧客管理B、台帳管理Cに決済完了通知を送信
⑬	C	顧客管理Aに決済完了通知を送信
⑭	A, B	太郎宛、花子宛それぞれに決済完了通知を送信

処理フロー案 2 を、処理フロー案 1 と比較すると次の 3 つの特徴がある。

特徴の 1 点目は、送金に必要な CBDC 台帳の資金を予め確保する点である。処理フロー案 1 は、顧客管理システムで取引制限判定を行い、同システムとしては取引を実行できることを確認したうえで、台帳管理システムで取引を実行する（その際に残高不足の判定等を行う）というものであった。一方、処理フロー案 2 は、送金元である A で取引制限判定を行うとともに、C で残高不足の判定や送金に必要な CBDC 台帳の資金を確保した後、送金先で必要な処理を行う（その際に、B で取引制限判定を行うほか、D で保有額制限判定を行う）。

特徴の 2 点目は、台帳管理システムにおける処理が 2 段階になる点である。処理フロー案 1 では、特徴の 1 点目で示したとおり、台帳管理システムにおいて指図を受け

た後は同システム内の処理（C の CBDC 台帳の減額と D の CBDC 台帳の増額）を留保機能も使用しつつ一体的に行う。一方で、処理フロー案 2 では、A において各種確認を行った後、台帳管理システムの処理の 1 段階目として C の CBDC 台帳を留保付きで減額（図表 16 図中③）し、B において各種確認を行った後、2 段階目として D の CBDC 台帳における留保付きの増額から留保解除までの一連の処理（図表 16 図中⑦～⑪）を行っている。このため、処理フロー案 2 は、C の留保付きの減額から、D の増額が確定するまでの処理に関わる主体数・処理のステップ数ともに処理フロー案 1 よりも多くなる（C において減額留保している時間が処理フロー案 1 に比較して長くなる）ため、エラーハンドリングが相対的に複雑になることが考えられる。

特徴の 3 点目は、主体間の通信回数が相対的に少ない点である。実験用システムの送金の処理フローは、台帳管理システムが CBDC 台帳を更新する場合には、相対する顧客管理システムによる指示が必要である、という考え方に基づいており、その結果、主体間の通信については、顧客管理システムと台帳管理システム間は少なくとも 1 往復（指示とその結果の通知）が必要となる。また、太郎から花子への送金は、単純化すると送金元（太郎）から送金先（花子）に増額を指示するものであるといえる。この 2 つを掛け合わせると、太郎から花子への送金においては、送金元での処理（A および C）を先に行った後に、送金先での処理（B および D）を行う処理フローが効率的であると考えられる。これを処理フロー案 1 および 2 に当てはめると、処理フロー案 2 は効率的な処理フローに近い一方、処理フロー案 1 については送金元と送金先との間で取引の確認（A および B）にかかる通信の往復と台帳管理間の送金指示にかかる通信の往復（A および C、B および D）が必要である点が通信回数の違いとなって表れていると考えられる。

2.3 エンドポイントデバイス

ここでは、ユニバーサルアクセス（誰でも使える）やセキュリティ（安心して使える）といった点を踏まえつつ、CBDC のエンドポイントデバイスに関する論点について整理する。具体的には CBDC のエンドポイントデバイスとして必要となりうる要素や、スマートフォンおよびカード型デバイスに関するエンドポイントデバイスとしての親和性について個人が店舗支払いを行う場面を想定し考察を行った（以下 2.3.1, 2.3.2）。

また、個人のエンドポイントデバイスとして、仮にカード型デバイスを想定する場合は、店舗決済端末に当該カードを挿入（接触型）またはタッチ（非接触型）することによって決済を行うこととなり、店舗決済端末の存在が必須となる。店舗決済端末については、CBDC の決済に対応する新規の端末を用意するという考え方も存在する一方、コスト面や店舗のオペレーション面といった導入の容易性等を踏まえれば、既存の端末を活用することも考えられる。また、店舗決済端末から CBDC システムに取引に関する情報を連携するための経路についても、既存のインフラの活用が考えられる。ここでは、そうした店舗決済端末等の既存インフラの活用可能性やシステム改修が必要となりうるポイントも簡単に整理した（以下 2.3.3）。

2.3.1 エンドポイントデバイスにおけるユニバーサルアクセス

エンドポイントデバイスにおけるユニバーサルアクセスについては、CBDC のユーザーの属性や環境等によって排除されることなく、誰でも使えることが重要と考えられる。具体的には、図表 17 のような考慮事項および対応策が想定され、これらのうち、システム面で留意しておくべき内容をまとめると、大きくみれば、(A)デバイスに表示される情報等が誰にとっても分かりやすいこと、(B)デバイスの取得や既存デバイスの活用が行いやすいこと、の 2 つに集約される。

図表 17　ユニバーサルアクセスにかかる考慮事項と主な対応策

ユニバーサルアクセスの観点での考慮事項			主な対応策	分類*
年齢	高齢層		文字等の大きさ、見やすさ	(A)
	若年層		わかりやすい画面	(A)
障がい	障がいがある	視覚	視覚面の配慮（色覚多様性への配慮、音声読み上げ対応等）	(A)
		身体	身体面の配慮（手や足が不自由な方による決済など）	(A)
言語	日本語が堪能でない（国内居住者）		平易な日本語や英語等での表記	(A)
経済面等	保有済みのデバイスの活用	利用者	既存のデバイスなどでCBDCが使える	(B)
		店舗	既存のデバイスを活用可能	(B)
			既存のデバイスの改修費用負担が大きくない	(B)
	新たなデバイス等が必要な場合への配慮	利用者	新たなデバイスを手頃な価格で取得可能	(B)
		店舗	新たなデバイスを手頃な価格で取得可能	(B)

＊ (A)デバイスに表示される情報等が誰にとっても分かりやすい、(B)デバイスの取得や既存デバイスの活用が行いやすい

ここで、個人ユーザーのエンドポイントデバイスとして、主に想定される「(a)スマートフォン」と「(b)カード型デバイス[57]」のほか、参考として、個人側はデバイスを保有せず店舗側のデバイスを使った生体認証のみで CBDC の決済を行う「(c)デバイスレス」の 3 つを比較すると、図表 18 のようになる。

図表 18　ユニバーサルアクセスの観点でのエンドポイントデバイスの評価

<table>
<tr><th rowspan="2">主な対応策</th><th colspan="2">(a)スマートフォン</th><th colspan="2">(b)カード型デバイス</th><th>（参考）(c)デバイスレス</th></tr>
<tr><th>コード決済</th><th colspan="3">IC決済</th><th>デバイスレス決済</th></tr>
<tr><td>文字等の大きさ、見やすさ</td><td colspan="2" rowspan="2">画面で対応可能</td><td colspan="3" rowspan="3">店舗決済端末に依存</td></tr>
<tr><td>わかりやすい画面</td></tr>
<tr><td>視覚面の配慮（色覚多様性への配慮、音声読み上げ対応）</td><td colspan="2">画面やサポート機能等で
対応可能</td></tr>
<tr><td>身体面の配慮（手や足が不自由な方による決済など）</td><td colspan="2">デバイス操作が必要</td><td colspan="2">カードの出し入れ等が必要</td><td>生体認証の方式に依存
（ユーザー側が対応困難な場合あり）</td></tr>
<tr><td>平易な日本語や英語等での表記</td><td colspan="2">画面で対応可能</td><td colspan="3">店舗決済端末に依存</td></tr>
<tr><td>既存のデバイスなどでCBDCが使える（利用者）</td><td colspan="2">ウォレットアプリのインストールで
対応可能</td><td colspan="2">カード型デバイスの
実現方法に依存</td><td>—</td></tr>
<tr><td>既存のデバイスを活用可能（店舗）</td><td colspan="5" rowspan="2">店舗決済端末や決済インフラの改修が必要</td></tr>
<tr><td>既存のデバイスの改修費用負担が大きくない（店舗）</td></tr>
<tr><td>新たなデバイスを手頃な価格で取得可能（利用者）</td><td colspan="2">スマートフォン非保有者は
端末購入等が必要</td><td colspan="2">カード型デバイスの
実現方法に依存</td><td>—</td></tr>
<tr><td>新たなデバイスを手頃な価格で取得可能（店舗）</td><td colspan="2">新規導入の場合でも
静的MPMは低コスト</td><td colspan="3">店舗決済端末が存在しない場合は端末購入等が必要</td></tr>
</table>

[57] ここでは、IC チップが搭載されている一般的な IC カードを想定し、指紋認証機能、ディスプレイ、PIN パッド等は具備していないような物理カードを仮定する。

まず(a)スマートフォンについては、既にスマートフォンを保有している個人からみれば、システム的には新たなデバイスを準備する必要なくウォレットアプリをインストールすることで CBDC が利用可能となるほか、搭載するウォレットアプリの工夫等によって、画面や表記、機能等をユーザー個別の事情に合わせて一定程度はカスタマイズ可能という特徴がある。またスマートフォンは、決済の方式としてコード決済（MPM 方式、CPM 方式）や対応機種では IC 決済（非接触方式[58]）など様々な方式が選択可能となる。特に 2 次元コードを設置した静的 MPM 方式によるコード決済については、現状の支払い手段が現金のみ等で店舗決済端末が存在しない小規模店舗が新たに CBDC を利用可能とする場合において、店舗からみればコスト面や運用面等で親和的と想定される。

一方で、スマートフォンを保有していない個人（特に低年齢層および高年齢層のユーザー[59]）も一定程度存在すること等を踏まえると、何らかの(b)カード型デバイスを用意することも考えられる。この場合、CBDC の決済方式は、当該カード内の IC チップを利用した IC 決済方式に限定され、店舗決済端末による IC チップの読み取りが必須となる。このため、仮にカード型デバイスを導入する場合は、店舗決済端末が存在しない小規模店舗等の取り扱いを検討する必要が生じうる[60]。

なお、(c)デバイスレス決済については、カード型デバイスと同様に店舗決済端末が必須となる。例えば、店舗決済端末側で何らかの生体認証を行う場合は、生体情報を読み取れる端末を店舗に設置したうえで、店舗側の顧客管理システムと個人側の顧客管理システムとの間での認証情報の連携を行う必要がある。

2.3.2 セキュリティ面でのスマートフォンやカード型デバイスの特徴

CBDC の利用において、セキュリティ面で特に注意を要する点は、第 3 者による不正アクセス・不正利用、不正作出と想定される（詳しくは、2.5.2 不正アクセス・不正取引への対策、2.5.3 不正作出への対策を参照）。CBDC のエンドポイントデバイスと

---

[58] NFC（Near Field Communication）による近距離無線通信技術を利用する方法。

[59] 総務省「令和 6 年通信利用動向調査（世帯構成員編）」をみると、20 歳代～60 歳代のスマートフォン保有率は 87%～95%程度と比較的高い。一方で、13 歳未満は 50%程度、70 歳台は 68%程度、80 歳以上は 31%程度と、低年齢層および高年齢層の保有率は顕著に低くなっている。

[60] 例えば、店舗決済端末が存在しない小規模店舗等ではカード型デバイスは利用不可とすることも考えられるが、その場合、カード型デバイスしか保有しない個人は当該店舗では CBDC を利用できない、といった課題が生じる。

しては、これらのリスクへの対策として、（１）「当人認証」や（２）「取引認証」等の認証行為を適切に実施可能であること、（３）デバイス内の重要な情報等が漏洩・改ざんされないための耐タンパ性[61]を有すること、が特に重要となる。

（１）当人認証

「当人認証」については、まずカード型デバイスによる IC 決済をみると、カードを保有しているという状態（所持認証）をもって当人認証済みとみなす考え方もありうるものの、カードの盗難等のリスクを踏まえれば、（リスクに応じて）決済時に店舗決済端末等において PIN 入力（知識認証）と組み合わせる方法が想定されるほか、状況によっては顔認証等（生体認証）と組み合わせる方法も想定される。この場合、店舗決済端末等ではそうした当人認証を適切に行うための機能（PIN 入力・照合機能、生体情報入力等）と、カード型デバイスとの通信機能（IC 読み取り等）の具備が必要となる。

次に、スマートフォンによる IC 決済（非接触方式）をみると、例えばスマートフォンのウォレットアプリ等を使って決済を行う際に、スマートフォン側で顔認証（生体認証）や PIN 入力（知識認証）を行うことが可能であれば、当人認証の選択肢が増える点はメリットと考えられる。この点、スマートフォンによるコード決済においても、例えばウォレットアプリの起動時やウォレットアプリによる決済実行時に、生体認証等による当人認証を行うことが考えられる。

（２）取引認証

「取引認証」については、まずカード型デバイスによる IC 決済をみると、カード型デバイス単体でユーザーが取引金額を確認することができない（カード型デバイスにディスプレイが具備されていない）ため、ユーザーは店舗決済端末等のディスプレイ上で決済時の取引金額を確認したうえで取引を継続する方法が考えられる。この場合、店舗決済端末等では金額を表示するための機能の具備が必要となる。

他方で、例えばスマートフォンの MPM 方式の場合には、スマートフォン上で送金先、金額の取引内容を確認することができるため、ユーザー自身のスマートフォンにより取引認証を実施することができる。ただし、取引認証が正しく行われる前提として、スマートフォンのウォレットアプリにおいて（１）の当人認証が正しく行われて

[61] システムやデバイス、ソフトウェアが外部からの不正アクセスや改ざん（タンパリング）等に対する抵抗力を備えていることを指す。

いる必要がある。また、静的 MPM の場合、店舗設置の 2 次元コードが悪意のあるものに貼り替えられるリスクを踏まえた対策が求められる。

（3）耐タンパ性

デバイス内の重要な情報等が漏洩・改ざんされないためには、スマートフォンやカード型デバイスを用いた決済において、スマートフォン内またはカード内の IC チップ自体が適切に耐タンパ性を有する必要がある。また、IC チップに格納されているアプリやスマートフォン等の脆弱性対策等を講じて、それらリスクを低減することが重要となる。なお、個人のデバイスが耐タンパ性を有していたとしても、決済方式によっては店舗決済端末等が介在するため、店舗決済端末側のセキュリティ対策も併せて重要となる[62]。

2.3.3 店舗決済端末等における既存インフラの活用可能性

ここでは、スマートフォンのコード決済における CPM 方式や IC 決済（非接触方式）を前提として、店舗決済端末等の既存インフラの活用可能性について考察する[63]。これは、既に店舗決済端末を利用してキャッシュレス決済を行っている店舗において CBDC に対応する場合、店舗側のコスト面や運用面での導入容易性に鑑みると、既存の店舗決済端末を利用することが一つの案として考えられることによるものである。

まず、既存の CPM 方式のコード決済の場合は、店舗決済端末は決済ゲートウェイ事業者に電文を送信し、当該事業者のシステムが各コード決済事業者のシステムに電文を振り分けるという流れが一般的であり、既存の IC 決済の場合は、決済端末センターや決済ネットワーク事業者が電文の振り分けを行うことが一般的である。また、既存の店舗の売上管理用システム（POS システム[64]）と店舗決済端末が連動している状況も考えられる。

---

[62] 例えば、IC 決済や CPM 方式のコード決済のように店舗決済端末を介して決済電文が送信されるケースでは、当該電文に含まれるプライバシー情報が漏洩しないよう、電文内容を暗号化して送信し、端末内にプライバシー情報が残存しないような仕組みが重要となる。また、コード決済のうち、例えば静的 MPM 方式については、前述のとおり店舗に貼付された 2 次元コード（紙など）の貼り替えリスクがありうる点にも注意が必要である。

[63] 静的 MPM 方式のコード決済については、店舗決済端末等の既存インフラを原則活用しないことから、検討の対象外としている。

[64] 一般に、商品名や金額、個数、販売店舗等の売上情報をリアルタイムで記録・集計するシステムを指す。

これらの既存インフラ（店舗決済端末、決済ゲートウェイ事業者や決済端末センター、決済ネットワーク事業者のシステム、POS システム等）に対して、CBDC 対応を行うためのシステム改修の内容としては、例えば、（イ）CBDC の電文の判別とその後の振り分け（スイッチング）を行うための改修、（ロ）店舗決済端末や POS システム側で CBDC 決済に対応するための操作画面の改修（CBDC 決済の選択画面の追加等）、などが想定される。このうち（イ）に関しては、CBDC システム側で行うべき対応として、CBDC の決済電文に「CBDC の電文であることを判別するための情報」と「送金元（個人）[65]の顧客管理機関 ID の情報」といった電文の通信に必要な情報をそれぞれ含めておく必要があると考えられる。なお、これらのシステム改修を行ううえで、店舗決済端末等といったハードウェア自体に改修を行う可能性もあれば、店舗決済端末等に搭載されるソフトウェアの改修で足りる可能性もある。

また、プライバシーの観点では、既存インフラを利用するに際して、様々な主体を CBDC の電文が経由することになる。このため、2.1.1（２）口座 ID のトークン化（トークナイゼーション）機能で述べたトークン化の仕組みにより口座 ID を様々な主体に対して秘匿することも考えられる。

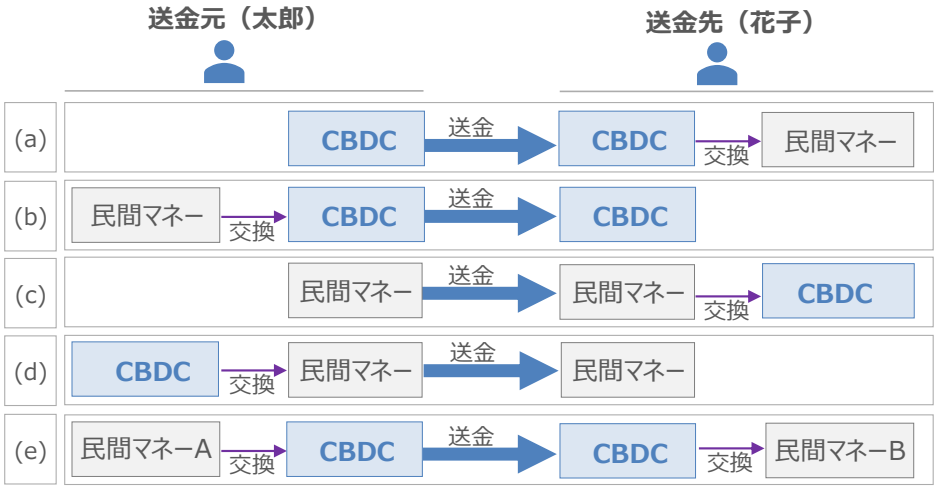
[65] CPM 方式を前提とした議論であるため、店舗決済端末から送金先（店舗）顧客管理システム、その後、送金元（個人）顧客管理システムへと電文を送付する必要があり、当該電文の宛先となる送金元の顧客管理機関 ID の情報が必要となる。

2.4 相互運用性

ここでは、CBDC と民間マネー（ここでは、主に資金移動業者によるコード決済を想定）の相互運用性について、個人間送金や店舗支払いのユースケースを念頭に、実現方法等を整理する。

CBDC と民間マネーとの相互運用性のあり方としては、様々な実現方法が考えられる中、ユーザー視点での大まかなイメージを図表 19 に示す。

図表 19　CBDC と民間マネーとの相互運用パターン

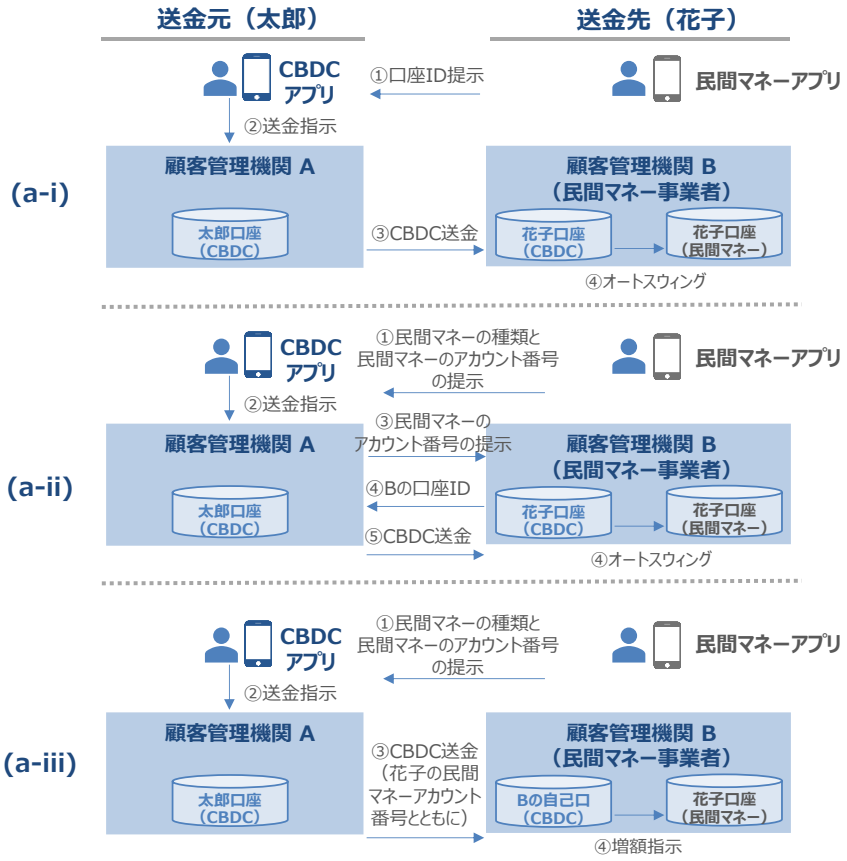


ここでは、CBDC と民間マネーとの相互運用性の実現方式の例として、個人間送金（以下 2.4.1）と店舗支払い（以下 2.4.2）のユースケース別に、主に図表 19 の(a)(b)のケースを取り上げて説明する（ (c)(d)(e)は、(a)(b)における CBDC と民間マネーとの交換の部分を組み合わせることで応用が可能と考えられる）。また、検討の前提として、ユーザーが送金や支払いを行うタイミングで、自動的に民間マネーが CBDC に交換されて決済が行われ、CBDC が支払い手段として意識されずに利用される（CBDC が黒子のように働く）形態であり、民間マネーを利用するユーザーが CBDC 口座を有している場合を例に整理する。

2.4.1 個人間送金

個人間送金について、CBDC と民間マネーを円滑に交換しながら送金を行う処理フロー、すなわち、(a)において送金元（太郎）が CBDC を利用して送金し、送金先（花子）は民間マネーで受け取るケース、(b)において太郎が民間マネーを利用し CBDC の送金を行い、花子は CBDC で受け取るケースについて検討を行った。

図表 20　(a)CBDC から民間マネーに個人間送金を行う際の処理フロー



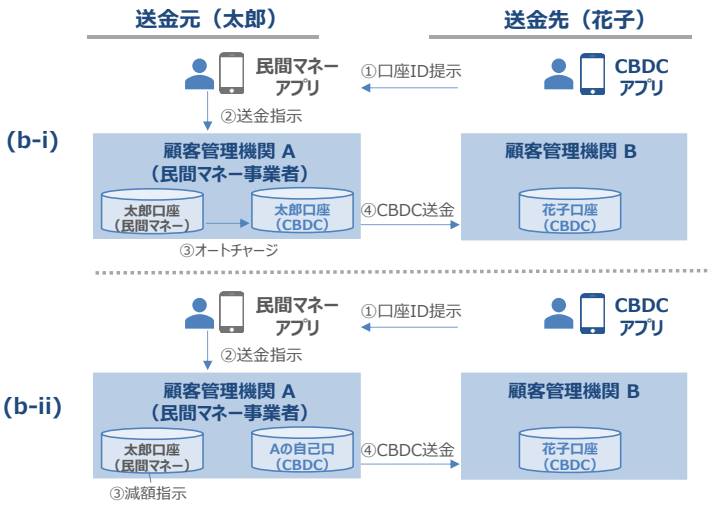
(a)のケースでは、処理フローとして、以下の 3 通りの方式が考えられる。すなわち、(a-i)太郎が花子の CBDC の口座 ID を指定して花子に CBDC の送金を行い、花子の顧客管理機関が即時に花子の民間マネーに交換（オートスウィング）する方式、(a-ii)太郎が花子の民間マネー情報（民間マネーの種類および同アカウント番号）を取得し、（これを顧客管理機関間で連携して）花子の CBDC の口座 ID に変換したうえで花子に CBDC の送金を行い、花子の顧客管理機関がこれを即時に花子の民間マネーに交換（オートスウィング）する方式、(a-iii) 太郎が花子の民間マネー情報を取得し、花子の顧客管理機関（B）の自己口に CBDC の送金を行ったうえで（その際に花子の民間マネーアカウント番号も伝達）、花子の顧客管理機関が即時に花子の民間マネーを増額する方式、が考えられる。

(a-i)の方式は、花子が CBDC のオートスウィング先の口座として民間マネーのアカウント番号を指定のうえ、CBDC 口座に着金した CBDC を全て民間マネー口座にオートスウィングする設定としておけば、システム的には実現可能と考えられる。なお、ここでは太郎が花子の CBDC の口座 ID を何らかの方法により取得することを前提としている。

(a-ii)の方式は、(a-i)の方式と同様の対応（花子の CBDC 口座から民間マネー口座へのオートスウィング設定）に加えて、太郎の CBDC のウォレットアプリにおいて花子の民間マネー情報等を入力する画面の実装が必要となる。また、太郎の顧客管理機関（A）が、入力された花子の民間マネー情報から花子の顧客管理機関（B）を特定したうえで、A は B に対して花子の民間マネーのアカウント番号を送信し、B から花子の CBDC の口座 ID を取得する仕組みが必要であると考えられる[66]。

(a-iii)の方式は、太郎の CBDC ウォレットアプリにおいて花子の民間マネー情報等を入力する画面の実装が必要となる。また、太郎の顧客管理機関（A）が、入力された花子の民間マネー情報から花子の顧客管理機関（B）を特定する仕組みが必要と考えられる。なお、ここでは CBDC の送金は太郎から B の自己口に対して行われるため、太郎や花子の CBDC の取引履歴の表示方法等が論点となりうる。

図表 21　(b) 民間マネーから CBDC に個人間送金を行う際の処理フロー



(b)のケースでは、処理フローとして、以下の 2 通りの方式が考えられる。すなわち、太郎は花子の CBDC の口座 ID を取得のうえ、民間マネーアプリを用いて CBDC の送金を行う際、(b-i)太郎の送金指示に伴い太郎の民間マネーを太郎の CBDC に交換（オートチャージ）し、これを花子に送金する方式、(b-ii)太郎の送金指示に伴い太郎の顧客管理機関（A）が太郎の民間マネーを減額し、この分を A が自己口から花子に送金する方式、が考えられる。

[66] B においても、同様に A から受領した花子の民間マネーのアカウント番号をもとに花子の CBDC の口座 ID を検索し、A に送信する仕組みが必要と考えられる。

(b-i)の方式は、太郎の民間マネーアプリおよび関連システム[67]において、花子にCBDC で送金する旨と CBDC の口座 ID を入力する画面の実装が必要となる[68]。また、太郎が CBDC のオートスウィング先の口座として民間マネー口座を指定のうえ、当該口座からオートチャージして CBDC の送金を行う設定としておく必要がある。

(b-ii)の方式は、(b-i)と同様の対応（太郎の民間マネーアプリおよび関連システムでの画面実装）が必要なほか、太郎の顧客管理機関（A）において、太郎の送金指示および入力された CBDC の情報から CBDC の送金であることを認識し、自動的に太郎の民間マネーを減額のうえ、その分 A の自己口から花子に CBDC の送金を行うような仕組みが必要となる。なお、ここでは CBDC の送金は A の自己口から花子に対して行われるため、太郎や花子の CBDC の取引履歴の表示方法等が論点となりうる。

以上のように、(a)、(b)の各ケースにおいて、実現方式によって必要な仕組みは変わりうるものの、多くの方式では、顧客管理機関や民間マネー事業者が提供する CBDC または民間マネーのウォレットアプリ等に追加的な対応が必要となる。また、(a-ii)では、送金元・送金先の双方の顧客管理機関において、民間マネーのアカウント番号から CBDC の口座 ID を検索するための追加的な対応を要する。

2.4.2 店舗支払い

対面での店舗支払いについて、CBDC と民間マネーを円滑に交換しながら支払いを行う場合の処理フローについて検討する。ここでは、送金元（個人）が民間マネーアプリで支払いを行い、送金先（店舗）が CBDC で受け取るケースを想定する（2.4.1 個人間送金の(b)のケースに類似）。なお、個人が CBDC で支払いを行い、店舗が民間マネーで受け取るケース（2.4.1 個人間送金の(a)のケースに類似）も考えられるが、一般に、店舗側が民間マネーで支払いを受けた場合、そのマネーは（民間マネーではなく）銀行預金として振り込まれることを踏まえ、当該ケースについては説明を省略する。

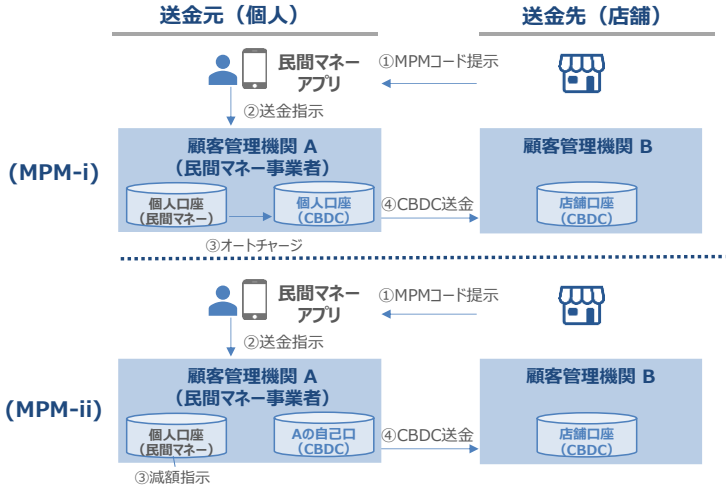
民間マネーによる店舗支払いとして、2 次元コードを用いた決済を例に説明する。まず、MPM 方式を利用する場合には、送金元（個人）の民間マネーアプリで、送金先（店舗）が提示した CBDC 用の MPM コードを読み取り支払いを行う形となる。この

[67] 太郎の民間マネーアプリにおける画面の実装に加えて、同アプリから指示を受信して処理を行うシステム（関連システム）に対しても何らかの実装が必要となりうる。

[68] 花子が CBDC のウォレットアプリで提示した個人間送金用の 2 次元コードを太郎の民間マネーアプリにおいて読み取る機能の実装等も考えられる。

点、基本的な処理フローは個人間送金と同様であり、実現方式も 2.4.1 個人間送金の(b)のケースにおける(b-i)と(b-ii)の 2 通りが考えられる。これらの場合、送金元（個人）の民間マネーアプリでは、店舗が提示した CBDC 用の MPM コード（当該コードには「CBDC の支払いであること」や「店舗の口座 ID」等の情報が含まれる想定）を読み取る機能の実装が必要となる。その他の必要な対応についても、基本的には個人間送金と同様である。

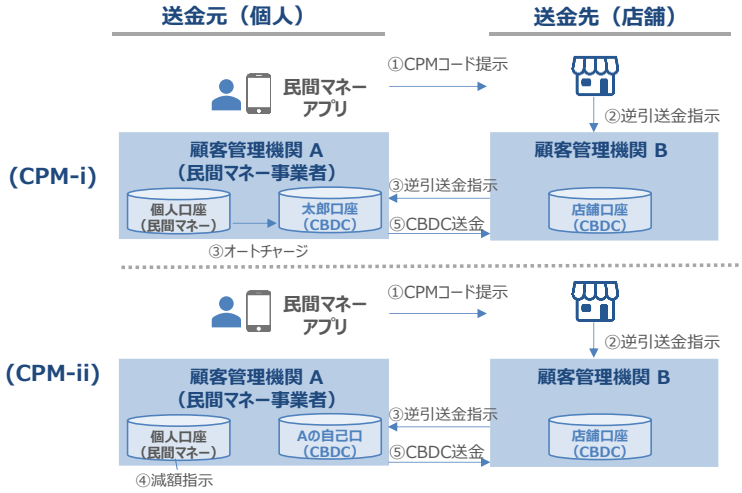
図表 22　民間マネーから CBDC に店舗支払いを行う際の処理フロー（MPM 方式）



次に、CPM 方式を利用する場合について説明する。CPM 方式の場合、送金元（個人）の民間マネーアプリで CPM コードを提示し、それを送金先（店舗）の端末で読み取ったうえで、送金先（店舗）の顧客管理機関（B）を経由して個人の顧客管理機関（A）に電文を中継する形となり、個人間送金とは異なる処理フローとなる[69]。この場合の実現方式は、（1）個人間送金の(b)のケースと同様、(CPM-i)個人の民間マネーから個人の CBDC にオートチャージを行ったうえで、店舗に CBDC の送金を行う方式、(CPM-ii)A が個人の民間マネーを減額したうえで A の自己口から店舗に CBDC の送金を行う方式の 2 通りが考えられる。

[69] 詳細な処理フローは、2025 年進捗報告書【別冊】の「1.7. CPM 方式の事務フロー例」を参照。

図表 23　民間マネーから CBDC に店舗支払いを行う際の処理フロー（CPM 方式）



いずれの方式においても、個人の民間マネーアプリでは、提示する CPM コードに「CBDCの支払いであること」や「個人の口座ID」の情報を含める必要が生じうる[70]。また、CPM 方式の場合、店舗決済端末を介して個人の顧客管理機関（A）に対して逆引送金[71]の指示を行う形となるため、個人の CPM コードに「CBDC 用の逆引トークン（個人の CBDC 口座に対して逆引送金を許可するための認可情報）」を埋め込む必要が生じる点にも留意を要する。

[70] この点、個人の民間マネーアプリ上で「民間マネー用の CPM コード」と「CBDC 用の CPM コード」を使い分ける必要が生じうるが、これに対しては、例えば、民間マネーのアプリ上でユーザーが CBDC 用に切り替えたうえで CPM コードを提示する、といった方法が考えられる。

[71] 詳細は、2025 年進捗報告書【別冊】の「Ⅰ.7.CPM 方式の事務フロー例」を参照。

2.5 セキュリティ

ここでは、システムのセキュリティ確保のための要素である機密性、完全性を中心に議論を行い[72]、特に重要となると考えられる、台帳管理システムや、顧客管理システムにおけるユーザーとの接点の部分を中心に整理する[73]。具体的には、まず、一般的な重要インフラにおけるサイバーセキュリティ対策を概観したうえで（以下 2.5.1）、その後、CBDC で特に注意すべき主なリスクおよびその対策を整理する（以下 2.5.2, 2.5.3）。なお、ここでは、技術的管理策を中心に整理することとし、物理的管理策（データセンターの管理区域の入退室管理等）、組織的管理策（経営層の関与、責任者・権限割り当て等）、人的管理策（人材育成、意識啓発等）などについては本検討の対象外とする。また、ここで挙げるリスクや対策は代表例であり、網羅的に記載しているわけではないことにも留意が必要である。

2.5.1 一般的なサイバーセキュリティ対策の考え方

「セキュリティ・バイ・デザイン」においては、まず「セキュリティリスク分析」を行うことが重要となる。そこでは、保護すべき情報資産を特定したうえで、想定脅威やシステムへの影響度を踏まえたリスクを分析し、それを踏まえたセキュリティ対応方針の検討を行う。

その後、機能面・非機能面に関する「セキュリティ要件定義」を行ったうえで、セキュリティ要件に基づいた調達を行い、その後「セキュリティ設計」、「セキュリティ実装」、「セキュリティ運用」を行うという流れになる。

「セキュリティ設計」や「セキュリティ実装」においては、システムにおけるアプリケーション、ミドルウェアや OS、ネットワーク等の各コンポーネントに対する重層

[72] システムのセキュリティを構成するもう一つの要素となる「可用性」については、2.6 可用性で整理する。

[73] EC 決済におけるセキュリティ対策は 2.1.3 EC 決済（e コマース）において、エンドポイントデバイスにおけるセキュリティ対策は 2.3.2 セキュリティ面でのスマートフォンやカード型デバイスの特徴において述べている。

的なセキュリティ対策[74]が重要となる。また、サイバーレジリエントな設計[75]や、これらセキュリティ対策の品質が確保されているかを確認するためのテストも重要となる。

「セキュリティ運用」においては、セキュリティ運用体制を確立したうえで、平時・有事の運用手順の整備、有事を想定した訓練、システムに影響する脅威・脆弱性情報の定常的な分析、脆弱性対応、速やかなインシデント検知、インシデント発生時の速やかな対応・復旧といったセキュリティ運用プロセスの整備が重要となる。また、システムの変更、脅威の変化等に応じて継続的にリスク分析とリスクへの対応を行うことが重要である。特に、脅威の変化については、近年の AI 技術の進歩により攻撃者の能力が飛躍的に向上する可能性も意識する必要がある。

これらの観点を踏まえつつ、CBDC システムにおいて、特に注意すべきリスクや特徴的なリスク、これらへの対策について 2.5.2 不正アクセス・不正取引への対策、2.5.3 不正作出への対策でまとめる。

なお、これらの対策の中でも、特に機密性対策として重要となる「暗号化」について、将来的に大規模かつ実用的な量子コンピュータが実現されることを想定した、PQC（Post-Quantum Cryptography）の利用や PQC への移行について議論が広く行われている[76]。

この点、CBDC システムにおいても PQC の利用または PQC への移行について意識する必要があり、システム設計面で、例えば、システムのアーキテクチャーの観点で暗号処理部分をモジュール化・疎結合化したうえで、暗号アルゴリズムの移行を容易にする設計としておくといった工夫が重要となる。また、同システムが既存の様々なシステムと接続したうえで、暗号化通信を行う可能性を考慮すると、PQC への移行を容易とするために「ハイブリッド方式[77]」を活用することも考えられる。さらに、PQC の利用により、鍵データ・暗号化データのサイズが増大する可能性や、処理速度が低

---

[74] 例えば、アプリケーションに関するセキュアコーディング、アクセス制御、プログラム改ざん検知、マルウェア対策、ファイアウォール・IDS/IPS による防御、システムログの統合的な監視・分析等。

[75] 例えば、ネットワークの分離、アクセス可能範囲の最小化等。

[76] CRYPTREC（Cryptography Research and Evaluation Committees）によると、2020 年時点で「現在の量子コンピュータの開発状況を踏まえると、暗号解読には規模の拡大だけでなく量子誤り訂正などの実現が必要であるため、CRYPTREC としては、CRYPTREC 暗号リスト記載の暗号技術が近い将来に危殆化する可能性は低い」とされている。もっとも、中長期的な将来としては、大規模かつ実用的な量子コンピュータが実用化されると、現在普及している公開鍵暗号のセキュリティが低下する可能性があり、そのリスクへの対応として、PQC の利用や PQC への移行が考えられる。

[77] 既存の暗号アルゴリズムと PQC のアルゴリズムを組み合わせて実装すること。

下する可能性を想定すると、CBDC システムにおける PQC 利用に伴う処理性能面への影響（それに伴うサイジング面での影響）を見極めることが重要となる。

CBDC システムにおいて採用する暗号アルゴリズムは、安全性や実装性能が確認されたと専門家によって十分に評価されたアルゴリズム（CRYPTREC 暗号リスト）を利用することが重要となる。「政府機関等における耐量子計算機暗号（PQC）への移行について（中間取りまとめ）[78]」においては、「CRYPTREC 暗号リストの更新が可能となるよう、PQC の安全性評価・実装性能評価に関する活動を開始している」や「政府機関等における耐量子計算機暗号（PQC）への移行について、原則として、2035 年までに行うことを目指し（以下略）」とされている。

このように、CRYPTREC や政府から発信される情報等、PQC を含めた暗号技術の最新動向について継続的に情報収集しつつ、CBDC システムの設計・構築・運用にあたって必要となる施策を適切に実施していくことが重要である。

2.5.2 不正アクセス・不正取引への対策

ここでは、不正アクセス・不正取引への対策として、認証と認可について検討を行う。

（1）認証

ユーザーと主に顧客管理システムとの接続において、リスクの度合いに応じて「当人認証」、「取引認証」、「端末認証[79]」等、様々な要素での認証技術を適切に組み合わせることが重要である。また、それらの認証手段と補完的に利用される「リスクベース認証[80]」の活用も有効である。

「当人認証」は、サービスを利用しようとするものが予め登録された本人であることを確認する行為であり、当人になりすまして取引を行うことを防ぐことが目的である。スマートフォンでの取引を例に考えると、ワンタイムパスワードを使ったとして

---

[78] 内閣官房国家サイバー統括室、「政府機関等における耐量子計算機暗号（PQC）への移行について（中間取りまとめ）」、2025 年 11 月
https://www.cas.go.jp/jp/seisaku/pqc/pdf/report_202511.pdf

[79] 端末認証とは、サービスを利用しようとする端末（スマートフォンや IC カード等）が予め登録された端末であることを確認する行為をいう。

[80] リスクベース認証とは、不正取引の疑いや脅威レベルが高いと判定された場合に追加的な認証を要求する行為をいう。

も、リアルタイムフィッシングと呼ばれる高度な手法によりなりすましを許してしまう手口も確認されている[81]ため「フィッシング耐性が高い[82]多要素認証[83]」を利用することにより、当人認証の強度を引き上げることが対策となる。その具体的な技術の一つとして、FIDO2 標準に基づくパスワードレスの認証方法である「パスキー」を利用することが考えられる。なお、同技術はセキュリティと UX の双方に配慮した認証技術である。

「取引認証」は、ここでは「誰から誰にいくら支払う」という取引情報が送金元（個人）の意思に基づくものであることを、送金元（個人）の顧客管理機関が確認する行為と定義して議論を行う。この行為は、取引情報が改ざんされることによる不正を防ぐことを目的としている。店舗支払いでの各種取引を例に考えると、MPM 方式での店舗支払い[84]については、送金元（個人）のスマートフォンのウォレットアプリ等により「誰から誰へいくら支払う」という 3 情報について確認し、それを送金元（個人）の顧客管理システムに直接伝える仕組みとなっている[85]。CPM 方式での店舗支払い[86]については、 3 情報が、送金元（個人）の顧客管理システムに到達する前に、送金先（店舗）の店舗決済端末や顧客管理システムを経由する。すなわち、送金元（個人）は、自身が保持するデバイス（スマートフォン等）のみを用いて 3 情報の確認ができない（そのため送金元（個人）の顧客管理システムも送金元（個人）の意思を直接確認できない）ため、店舗決済端末等の信頼性を高めることが重要となる[87]。なお、市中におけるクレジットカード等の決済においても同様に、送金元（個人）は、自身が保持するデバイス（クレジットカード）のみを用いて 3 情報の確認ができない。このため、（国際ブランドにより認定が行われた）信頼された店舗決済端末を利用し、店舗

---

[81] IPA（独立行政法人情報処理推進機構）、「IPA NEWS Vol.69」、2024 年 11 月 https://www.ipa.go.jp/about/ipanews/ipanews202411.html

[82] ここでは、ログインしようとしているサービスのサイトが本物か偽物かの判定を人の判断に依存せずにできることをいう。

[83] パスワード等による知識認証、生体認証、所有物認証といった複数の認証要素を組み合わせることをいう。

[84] 詳細は、2025 年進捗報告書【別冊】の「Ⅰ.6.静的 MPM 方式の事務フロー例」を参照。

[85] 取引認証を厳密に実施するためには、通信経路での改ざんを検知するための仕組み等が必要となるが、簡単のためここでは通信経路の完全性確保については議論の対象外とする。

[86] 詳細は、2025 年進捗報告書【別冊】の「Ⅰ.7.CPM 方式の事務フロー例」を参照。

[87] もしくは、送金元から見て、店舗決済端末や店舗顧客管理システムの信頼性に依存しない前提に立つのであれば、送金元の顧客管理システムに 3 情報が伝達された後に、送金元に当該内容を確認する処理フローを追加する方法もあるものの、この場合にはユーザビリティが低下することが想定される。

決済端末の金額表示を送金元（個人）が確認したうえで支払い行為を行うことで、3情報の確認を送金元（個人）が行っている。EC 決済などの非対面取引における取引認証については、2.1.3 EC 決済（e コマース）において述べたとおりである。CBDC のエコシステムにおける店舗支払いのあり方についてまだ結論が得られているわけではないものの、市中のリテール決済における状況も参考にしつつ、セキュリティ、利便性、各主体の負担のバランスを取りながら、どのような取引認証を行うべきか見極めていく必要がある。

「リスクベース認証」は、他の認証手段と補完的に利用されることが想定される。当該認証は、不正送金に伴うリスクが高いと判断される場合には利便性を一定程度犠牲にしつつも追加的な認証行為を求める一方、リスクが低いと判断される場合には追加的な認証の省略など簡素化した措置を適用するという方法である。このため、セキュリティと利便性のバランスをとるための一つの方法となると考えられる。ただし、リスクを判定する[88]際により精度を上げるため、他の顧客管理機関等からリスクに関する情報を収集して学習するといった対応も考えられ、その場合にはプライバシー保護の観点で慎重な検討が必要となる。

なお、これら認証はユーザーにおける送金等の各種取引を想定したものであるが、システム管理者による、システム運用のための管理コンソール等への接続に関しては、昨今の情報セキュリティインシデントにおいてそのような経路からの不正アクセスが存在している状況に鑑みると、より厳格な運用（多要素認証の必須化、電子証明書による認証の必須化、アクセス元の限定等）が重要となる。

（2）認可

認証が正しく行われた後は、認可を実施（データ等へのアクセス権限を付与）することが求められる。

認可については、役割・職務（Role）に応じたアクセス権限を付与する方法（RBAC：Role-Based Access Control）、主体の属性・状態（Attribute）に応じたアクセス権限を付与する方法（ABAC：Attribute-Based Access Control）やそれらの組み合わせ等、様々な方法が考えられる。このため、これらを踏まえ認可ポリシーを適切に設計して実装することが重要となる。

---

[88] CBDC の社会実装時に、リスクベース認証におけるリスクの高さ（リスク値）を計算する主体、およびリスクベース認証の機能を提供する主体は未定。

一般に、金融サービスにおけるオープン API にかかる認可処理の実装では OAuth2.0 に準拠するケースが多い中、OAuth2.0 を実装する際には脆弱性を極力排除する必要があり、そのうえで OAuth2.0 のセキュリティ・プロファイルである FAPI2.0 を活用することが有効である[89]。CBDC のエコシステムにおける各主体間の接続において、API を用いることを想定した場合も、OAuth2.0 の脆弱性を極力排除するため、要求事項を十分に満たした実装が重要となる。また、CBDC のエコシステムにおける各主体間の接続においても同様に FAPI を活用することが有効と考えられる。

2.5.3 不正作出への対策

ここでは「不正作出」を台帳管理システム上に記帳された CBDC 残高を不正に改変することと定義し、不正作出のリスクおよび代表的な対策の検討を行った。

（1）事前対策

今回検討の前提とした実験用システムにおける送金の処理フローでは、取引単位で見て、減額、増額いずれか一方を実施するという処理は発生せず、同額の減額と増額を必ずセットで実施する形とした[90]。その結果、例えば「送金元での減額を伴わない送金先の増額」が取引として発生し得ないことから、このような処理フローとすることが対策の一つとなりうる。

（2）事後対策

(a)中央システムにおいて「発行額と還収額の差額（市中に流通している CBDC の総額の理論値）」と、「全台帳管理機関の集約口[91]と自己口の合計値（市中に流通している CBDC の総額の実績値）」とを比較し、仮に両者の差異が存在する場合には不正作出が行われたことを検知できる[92]。ただし、不正作出が発生した仲介機関の特定までは困難であるほか、総額と減額が整合するような不正として、ある口座の不正な増額と別の口座の不正な減額が同額で行われた場合は検出が不可能となる。

---

[89] 宇根正志、「オープン API のセキュリティ：認可処理における脆弱性と対策の高度化」、2025 年 1 月 https://www.imes.boj.or.jp/research/papers/japanese/kk44-1-2.pdf

[90] 詳細な処理フローは、2025 年進捗報告書の「2.1.2 送金の処理フロー」を参照。

[91] 仲介機関単位で当該仲介機関に口座を有するすべてのユーザーの口座残高を集約した口座。過去に実施した概念実証では「ユーザ口」と呼称していた。

[92] もっとも、24/365 でシステムを稼働させる場合、データ取得タイミングによっては仕掛中の処理（送金中の最終の処理等）が存在するため、それらへの配慮が必要となる。

また、(b)全台帳管理システムにおける取引履歴を集約して、送金の際に減額と増額がセットで行われていない取引が存在する場合には不正作出が行われたことを検知できる。この方法により、不正作出が発生した台帳管理システムを特定することは困難であるため、より詳細な調査のためには顧客管理システムにおける取引履歴との照合や、ユーザーの残高と取引履歴との照合も行うことが必要となる。

もっとも、特に(b)はシステム負荷が高くなる可能性がある点には注意が必要であるほか、各システムで記録している取引履歴自体が改ざんされた場合には、事後対策の難易度が高まることが挙げられる。このため、取引履歴の改ざんを検知する仕組み[93]の導入や、取引履歴を改ざんされにくくするため、全台帳管理システムの取引履歴を異なる技術を利用して構築したシステム[94]に集約して管理するといった手法も重要と考えられる。なお、これらは後述するシステム障害に伴う復旧作業にも資する可能性がある（2.6.2 障害停止を参照）。

（3）その他の観点

プライバシーに配慮する観点から、仲介機関の業務を顧客管理と台帳管理に分離しているなど、それぞれの業務を担うシステムを別々に構築することとなれば、攻撃者のインセンティブの観点で不正作出への対策の一助となる可能性がある。具体的には、実験用システムにおいては、台帳管理システムにおいて内部管理番号[95]が残高と紐づいているものの、内部管理番号が具体的にどのユーザーと紐づいているかは台帳管理システムにおいては管理しない設計としている。このため、台帳管理システムで（自身の口座の）不正作出を企図する攻撃者は、台帳管理システムに侵入するだけでは、どの内部管理番号の残高を改変させるべきか判断できず、その結果、攻撃のインセンティブを幾分低減させる効果があると考えられる。

---

[93] 各取引履歴にハッシュ値を算出しておく方法、各取引自体に電子署名を付す方法、各取引履歴のハッシュ値を直前の取引履歴を基に算出する方法、データの追加と参照のみが可能なテーブルに取引履歴を格納する方法、等が考えられる。

[94] メインのシステム（例：従来型の RDBMS）とは別の技術（例：DLT 技術）を別システムにおいて採用することにより、メインのシステムで何らかの手法により取引履歴の改ざんが行われた場合でも、別システムにおける取引履歴の改ざんが同じ手法では行われにくくすることが考えられる。

[95] 詳細は、2025 年進捗報告書【別冊】の「2. CBDC の口座開設の事務フロー例」を参照。

2.6 可用性

ここでは、台帳管理システムのインフラ面を例に、可用性確保策について整理する。具体的には、ここでは社会実装を意識した場合、システムが利用できない時間を極力ゼロとすることが求められると仮定し、計画停止[96]を極力ゼロに近づける施策についての検討を行ったうえで（以下 2.6.1）、システムに障害等が発生した場合でも極力サービスが停止しないための冗長化の施策として、特に、データセンターレベルでの機能停止（サイト機能の停止）を想定した冗長化の施策についても検討する（以下 2.6.2）。また、システムに障害等が発生した場合でも重要なサービスを最低限維持すべき水準において提供し続ける能力である「レジリエンス」についても検討する（以下 2.6.3）。

なお、検討にあたっては、実験用システムで仮置きした社会実装時の想定と同程度（通常時 1 秒あたり数万件、ピーク時同 10 万件以上）の事務量について、１件あたり数秒で処理が完了すると仮定した。

2.6.1 計画停止

（１）計画停止を極力少なくする施策（一般的な考え方）

計画停止を極力少なくする基本的な施策として、アプリケーションやソフトウェアのバージョンアップ時におけるローリングアップデート[97]やブルーグリーンデプロイメント[98]、ハードウェア交換時のホットスワップ[99]といった、コンポーネントを多重化したうえでシステムやサービスを停止させない範囲で一部分ずつ順繰りに停止して保守作業を行う手法がある。

こういった基本的な施策に加えて、大規模なシステムの更新など、システムを構成する幅広い機器等が影響を受ける作業においては、システムやサービスの停止を極力

[96] システムの保守作業等を行うために、事前の予定や計画に基づいて意図的に停止させること。

[97] システム全体を停止させることなく、システムの一部を段階的に新しいバージョンに更新することで、稼働状態を維持したままシステムを更新する手法。

[98] 現環境（ブルー）と新環境（グリーン）を準備し、直ちに新環境に切り替えることで、稼働状態を維持したままシステムを更新する手法。仮に問題が発生した場合には、瞬時にロールバックする（現環境に戻す）。

[99] サーバーを稼働した状態でサーバーの部品やコンポーネントを交換できるようにする仕組み。

少なくする施策として、バックアップサイト[100]の活用も考えられる。具体的には、メインサイトで保守作業等を行う際はバックアップサイトに切り替え、その後バックアップサイトで保守作業等を行う際はメインサイトに切り替えることにより、こうした保守作業等を行っている中もサービスを継続して提供するといった手法を採用することが考えられる。

（2）計画停止を極力少なくする施策（災害や障害発生時の対応力も意識した施策）

メインサイトとバックアップサイトという 2 サイト構成の場合には、メインサイトで大規模な保守作業等を行っている間、バックアップサイトにおいてサービスを提供することとなるため、その間のバックアップ機能を担うサイトが存在しない状態となる。この点、メインサイトで大規模な保守作業等を行っている間においても、災害や障害発生時への対応力を維持する観点では、３つのサイトを用意して、メイン用、バックアップ用および保守作業用等と役割をローテーションさせ、いずれかのサイトが保守作業等を行っている間であってもメインおよびバックアップの 2 サイト構成を維持する手法も考えられる。

（3）遠隔サイト間のデータ同期方式

システムやサービスの停止時間を短縮する観点では、遠隔のサイト間における常時のデータ同期方式による両現用構成[101]が考えられる。これにより、接続元のシステムが接続先をメンテナンス予定のサイトから他のサイトに切り替えることでシステムやサービスの停止なくメンテナンスを実施できる可能性がある。

ただし、CBDC の社会実装時において、大量のトランザクションを短時間で処理する必要があることを踏まえると、遠隔地にバックアップサイトが存在する場合、距離によるネットワークの遅延によって処理時間が長くならないよう、メインサイトで処理を完了した後にバックアップサイトへ当該処理に関するデータを送信する方法（データ非同期方式：メイン-バックアップ構成）を採用することも想定される。その場合、サイトの切り替えには、稼働中のシステムが処理したデータの伝送、切り替え先のシ

---

[100] 災害や障害発生時に備えて、通常使用するシステムを稼働させるデータセンター（メインサイト）から独立したデータセンターにシステムを１セット準備したもの。

[101] 例えば、ユーザーからすると、サイト A にあるサーバーに接続してサービスを利用しても良いし、サイト B にあるサーバーに接続してサービスを利用しても良い（ここで、サイト A とサイト B は遠隔地にあるとする）という構成になる。

ステムにおける受領データのチェック、接続元システムにおける接続サイトの変更といった作業が必要であるため、一定のサービス停止が必要となる。

2.6.2 障害停止

システム障害には、個別の機器の故障から電源や通信の断絶によるサイト（データセンター）機能の停止といった大規模な障害まで、様々なケースが考えられる。本検討では、より難易度の高い、サイト機能の停止を想定した施策を整理した。

（1）障害停止と計画停止の差異

障害停止を極力少なくする施策として、仕掛中のデータの消失への対応が重要である。すなわち、サイト機能が停止した中においてシステムやサービスの目標停止時間（RTO[102]）を極力短縮するための施策として、計画停止と同様にバックアップサイトへの切り替えが考えられる。その際に仕掛中の処理を意識する必要があるかどうかが、障害停止と計画停止との大きな違いである。

仕掛中の処理については、データがサーバーのメモリ上にあるなど、全てのデータをバックアップサイトに伝送することは困難であり、そういったデータはサイト機能の停止により消失しうる。加えて、2.6.1 計画停止で述べた、メインサイトで処理を完了した後にバックアップサイトに当該処理のデータを送信する方法（データ非同期方式：メイン-バックアップ構成）を採用する場合、メインサイトにおいて処理が完了していたとしても、バックアップサイトへのデータ伝送が未了であれば、メインサイトの機能停止に伴い当該データは消失しうる。

（2）データ消失対策

消失したデータについては、自身では復旧することができないため、当該データに関係するシステム（顧客管理システムや他の台帳管理システムといった自身以外のシステム）からのデータ再送信や、相手システムに対する問合せ等を通じ、相手システムの処理状況を把握したうえで、必要に応じ何らかの追加処理を行うことにより、相

[102] Recovery Time Objective（目標復旧時間）の略で、障害が発生してから業務を復旧するまでの目標時間のこと。

手システムと整合的な状態に復旧させる必要がある。なお、これは RPO[103]を極力ゼロに近づけると言い換えることができる。

なお、台帳管理システム間の整合性に着目すると、データ非同期方式において台帳管理システムに障害が発生した場合、ある口座における複数の減額・増額の処理にかかるデータが消失することも考えられる。この場合、復旧の過程で取引の前後関係を取り違えると、誤った残高不足判定による処理のエラーが生じうる[104]。このため、復旧に際しては、消失した取引データにかかる取引履歴を全て取り揃えた後、取引の前後関係を正確に認識したうえで対処する必要がある。

こうした復旧作業を実現するにあたっては、常時、全ての台帳管理システムから取引履歴を収集し蓄積しておくような仕組みを構築することで、データ消失が発生した場合には、当該取引履歴を直ちに参照するといった方法が考えられる。これは、2.5.3（２）事後対策でも述べた、不正作出の事後対策としても有用であると考えられる。また、当該取引履歴を蓄積する方法として、DLT 技術を活用し、全台帳管理システムの取引履歴を１つに集約したうえで分散管理するといったことも考えられる。その場合、当該取引履歴を常時蓄積するためのシステムのリソースや可用性など技術的に検討すべき論点が存在する。

2.6.3 レジリエンス

レジリエンス（オペレーショナル・レジリエンス）とは、金融庁の「オペレーショナル・レジリエンス確保に向けた基本的な考え方[105]」によると、仮に何らかの原因でシステム障害が発生してしまったとしても、「重要な業務」を最低限維持すべき水準（「耐性度」）で「提供し続ける能力」のことをいう。レジリエンスを構成する要素のうち、「重要な業務」と「耐性度」については社会実装時のユースケース・要件に依存するため、現時点で確定的な議論を行うことは困難である。このため、ここでは「（業務を）提供し続ける能力」を中心に検討する。

---

[103] Recovery Point Objective（目標復旧時点）の略で、障害が発生してから業務を復旧する際にどの時点までのデータを復旧させるかの目標値のこと。

[104] 例えば「残高 0 円の口座に、100 円が着金し（取引 A）、その後 50 円を送金した（取引 B）」という一連の取引を復旧したい場合、先に取引 B の履歴のみを参照すると「残高 0 円の口座から 50 円を送金した」形となり、この時点で残高不足エラーが発生する。

[105] 金融庁、「オペレーショナル・レジリエンス確保に向けた基本的な考え方」、2023 年 4 月 https://www.fsa.go.jp/news/r4/ginkou/20230427/02.pdf

CBDC のリテール決済サービスという特徴を踏まえると、（1）早期復旧や（2）代替手段の確保、（3）迅速な広報（ユーザーへの迅速な周知）によりユーザー目線で影響を限定的にするという施策が重要となる。以下では、上記施策それぞれにつき、CBDC システムでの実装を検討する。

（1）早期復旧

障害からの早期復旧にあたっては一般的な冗長構成の確保が重要であることはもちろんであるが、複数主体が関与している CBDC システムでは、障害が他主体へ波及してしまうと復旧に時間を要することが考えられるため、波及を防止する仕組みの構築が重要となりうる。例えば、サーキットブレーカー[106]のようなマイクロサービス・アーキテクチャー[107]におけるベストプラクティスを検討・導入することが重要と考えられる。

なお、特定地域の被災を想定した場合には、システムの各コンポーネントを地理的に分散することも障害の全体波及を防ぐための有効な施策と考えられる。台帳管理・顧客管理システムがそれぞれ独立であることを想定している実験用システムのアーキテクチャーは、地理的分散が比較的実現しやすい構成であるといえる。

早期復旧の確保にあたって他に考慮すべき CBDC の特徴としては、取引量が極めて多くなる可能性があることが挙げられる。相当高水準な事務量のもとでの復旧作業においては、障害の内容次第では復旧が必要となる取引が停止時間とともに増加していくため、人手のみでエラーログの解析・データ補正を早期に実施することは現実的ではない。このような環境下で早期復旧を担保するには、システムの稼働状況や個別取引のステータスを常時監視し、障害発生やデータ不整合時にはデータ補正をサポートできるような何らかのシステムが必要であると考えられる。

（2）代替手段の確保

障害からの早期復旧が何らかの事情により難しい状況でも、代替手段への切り替えで処理を継続できる場合には、ユーザーへの影響は最小限に抑制することが可能と考えられる。

---

[106] あるサービスで障害が検出された後、当該サービスへのアクセスを一時的にブロックする仕組み。

[107] 複数の小規模・軽量で個々が独立したサービスを組み合わせて、システムを構築する方法。

システム全体の障害に対する代替手段としては、バックアップサイトの整備のほか、サイバー攻撃に対する代替手段としては Non-Similar Facility（NSF）[108]の整備が考えられる。また、システムの一部障害への代替手段としては、例えばネットワーク部分の障害時に既存の民間決済インフラネットワークを利用して迂回するという施策が一例として挙げられる。また、オフライン決済の仕組みを別途構築しておき、それを代替手段として利用するということも考えられる（詳細は、BOX．オフライン決済を参照）。さらに、それらのシステム的な代替手段でも早期復旧ができない状況を想定すると、CBDC という決済手段の代替として、現金での支払いも考えられる。

（3）迅速な広報（ユーザーへの迅速な周知）

ユーザーの目線からすると、システム障害が発生しているにも関わらず、決済処理を待ち続けるという状況は望ましくない。ユーザーが、例えば代替手段として現金での決済を選択するためには、まずは障害発生時に CBDC が利用不可となっている旨をユーザーが迅速に認知できるような仕組みを具備することが重要である。この点、一定時間が経過すると自動で処理を中断・エラー応答を返すようなタイムアウトの仕組み（2.2.1 タイムアウト管理機能と留保機能の整理 でも議論）は、ユーザーを必要以上に待たせないという観点から重要な要素と考えられる。

以　上

[108] メインシステムとは異なる製品・技術を用いて構築された、当該システムの代替となるシステム。

BOX．オフライン決済

（1）導入目的・ユースケース、検討の前提

CBDC のオフライン決済については、導入目的やユースケース等に応じて、様々な実現方法や制度設計が考えられる。この点、「CBDC に関する関係府省庁・日本銀行連絡会議　中間整理」では、「自然災害などによって生じる通信障害や電力途絶といった場面でも CBDC の利用を可能にするもの」と整理されている。こうした利用シーンを想定し、ここでのオフライン機能は、「ある程度長期間にわたって通信障害等が発生した場合に、CBDC のエンドポイントデバイス（個人のスマートフォンや店舗決済端末等）がオフライン状態でも、個人間送金や店舗支払いを実施可能とするための仕組み」と仮置きする[109]。

（2）オフライン決済の実現方法

上述の仮置きを前提とした場合には、実現方法として、例えば、NFC（近距離無線通信）を利用した IC 決済（非接触方式）、すなわちエンドポイントデバイス同士を近づけることで、デバイス内に安全に格納された CBDC の価値データ（以下、「オフライン CBDC」という）をデバイス間で安全に移転して決済を行う方法が考えられる。この方法では、デバイス間でオフライン CBDC 関係の通信を行う際にインターネット接続は不要となるものの、個人は自身のオンライン CBDC[110]の口座から、事前に（オンライン状態で）価値データをオフライン CBDC に移転させておく必要が生じうる[111]。

---

[109] ここでは、個人側・店舗側の端末がともにオフライン状態であり、それが一定期間継続することを前提とした。もっとも、①個人端末はオフラインだが店舗側の端末は通信可能なケースや、②通信途絶が短時間に止まるケースなどでは、必要なオフライン機能が異なる可能性がある。例えば、IC 決済の場合は、個人側がオフライン状態であっても店舗側がオンライン状態なら特段の問題は生じない。また、コード決済のうち CPM 方式については、店舗側がオンライン状態であれば、個人端末で 2 次元コードを表示のうえ店舗決済端末が当該コードを読み取ることで、決済を行うことができる（この場合、不正送金対策等の観点から、オフライン決済の金額に上限を設ける等の対応により取引のリスク度合いを軽減することが考えられる）。このほか、Bank of England の Digital Pound における検討では、個人側・店舗側がともにオフライン状態であることを前提とした場合でも、「端末間で何らかのオフライン決済を行った後に、店舗側がオンラインに復帰した時点で送金を実行する」といった、送金予約型のオフライン決済（ディファードオフライン決済）も考えられている。もっとも、この方法では、店舗がオンライン状態に復帰して送金処理が開始する際、個人側の CBDC 残高が不足して送金が実行されない可能性があり、そうした残高不足時の対応が制度設計面等で適切に手当てされる必要がある（通信途絶の期間が長いほど残高不足のリスクが高まる）。

[110] オフライン CBDC に対して、オンライン状態（台帳管理システムにおける CBDC 口座に接続可能な状態で利用する状況）における CBDC のことを指す。

[111] 個人がオフライン CBDC を取得するためには、①自身のオンライン CBDC の残高から事前に移転させておく方法のほかに、②他の個人から個人間送金によってオフライン CBDC を取得する、③現金

オフライン CBDC を受領した個人や店舗は、オンライン状態に復帰した後、オフライン CBDC をオンライン CBDC の口座に移転することが可能となる。

上述の方法でオフライン決済を実現する場合、セキュリティ面では、オフライン CBDC が不正に生成・複製されるリスクや、使用済みのオフライン CBDC が端末に残存して二重使用されるリスクが想定される。こうしたリスクに対しては、2.3 エンドポイントデバイスでも述べたとおり、スマートフォン等に搭載された IC チップが耐タンパ性を有することや、IC チップに格納されているアプリやスマートフォン等の脆弱性対策等を講じることにより、それらリスクを低減することが重要となる。もっとも、一般にどのような対策を講じてもリスクを完全にゼロにすることは難しく、事後対策として、仮に不正なオフライン CBDC が発生した場合でも、それを何らかの方法で検知する仕組みを具備しておくことが必要となりうる。

（3）オフライン決済に関するセキュリティ対策

不正なオフライン CBDC を検知する方法として、①オフライン決済時に検知（リアルタイム検知）、②オフライン状態からオンライン状態になった時に検知（事後検知）の 2 通りが想定される。

①の方法については、リアルタイム検知は技術的に困難と考えられる。例えば、上述の「使用済みのオフライン CBDC が端末に残存して二重使用されるリスク」に関し、端末に残存したオフライン CBDC は、セキュアエレメントに格納されたオフライン CBDC が二重使用されるという状況まで想定すると、オフライン環境において不正な CBDC であることを外形上から判別することが困難と考えられる。

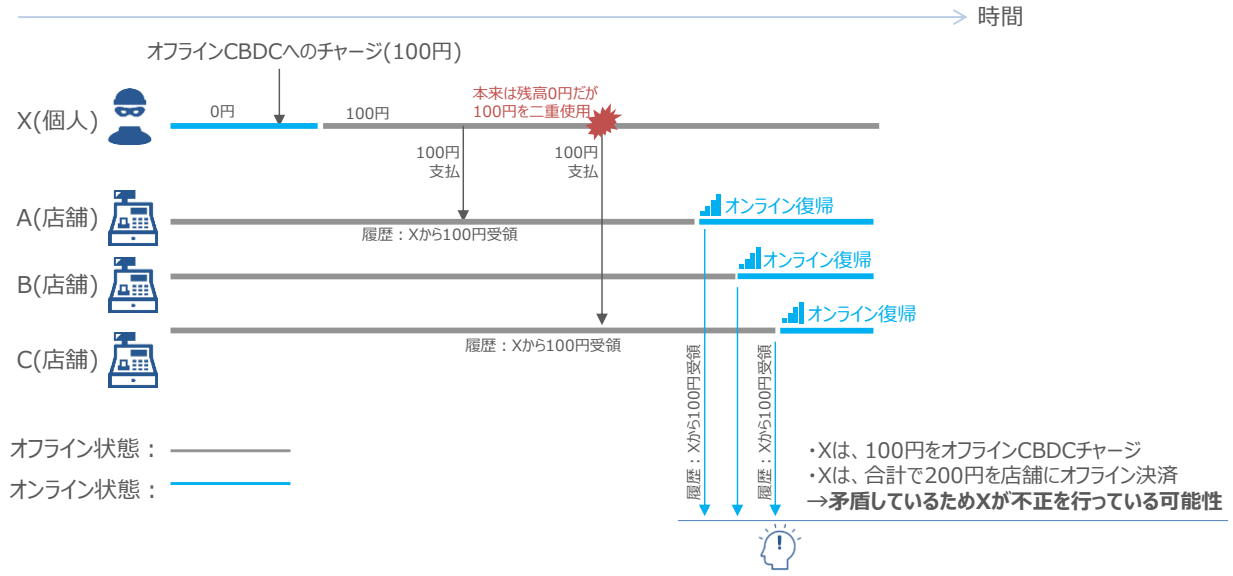
②の方法については、同様に二重使用のケースを考えると、オンラインの CBDC システム側でオフライン CBDC の二重使用の形跡を確認することが必要となる。ここで、仮にオフライン CBDC の転々流通（オフライン決済で受領したオフライン CBDC を、再び他のオフライン決済に利用すること）を行わない、かつ、オフライン決済による個人間送金を行わない前提であれば、「(a)オンラインの CBDC システム側で記録された、個人のオフライン CBDC の取得履歴（オンライン CBDC からオフライン CBDC への移転）」と「(b)当該個人がオフライン CBDC を利用した可能性のある全ての店舗における、当該個人から受領したオフライン CBDC の取引履歴」を突合することにより、

から個人端末（スマートフォンやカード型デバイス）に CBDC をチャージする機械を用意したうえで、個人が当該機械を用いて現金からオフライン CBDC に直接チャージする、といった方法も考えられる。

不正なオフライン CBDC を事後的に検知することが可能となる[112]。もっとも、この事後検知についても、当該個人がオフライン CBDC を利用した可能性のあるすべての店舗における CBDC の取引履歴を突合する必要があることから、多くの店舗がオンライン復帰した時点でないと事後検知の可能性が低くなってしまう点には留意が必要である。

また、オフライン CBDC の転々流通やオフラインにおける個人間送金を前提とする場合は、店舗だけでなく個人ユーザーも含めた多数のオフライン CBDC の受領履歴を確認しない限り、二重使用の形跡を確認することが難しく、事後検知の難易度が極めて高くなる。

図表 24　不正なオフライン CBDC の事後検知の方法



以上を踏まえれば、仮に上述の方法でオフライン決済を実現する場合、不正なオフライン CBDC の事後検知を行う観点からは、例えば「オフライン決済は店舗支払いのみをユースケースとし、オフライン CBDC を受領した店舗はオンライン状態に復帰しない限りオフライン CBDC を利用不可とする（オフライン CBDC の転々流通は不可とする）」といった制約を設けることや、オフライン CBDC に移転できる金額の制限をリスクに応じて設定することも、技術的には一案と考えられる。ただし、この場合でも、多くの店舗がオンライン復帰しない限り不正なオフライン CBDC を事後検知する

[112] 例えば、個人がオフライン CBDC を所有していない状態において、オンライン CBDC からオフライン CBDC に 100 円移転し（オフライン CBDC 残高 100 円）、同個人がオフラインにおいて店舗 A に 100 円および店舗 B にそれぞれ 100 円支払っている取引履歴を確認した場合、不整合を検知することができる。

ことは困難なケースが多く、セキュリティ面でのリスクが残存する点は課題と考えられる。

　このように、オフライン決済の検討にあたっては、オフライン決済の目的やユースケースを明確にしたうえで、適切な実現方法を検討することが重要であるほか、実現方法に応じて生じうるセキュリティ面でのリスクと、オフライン決済によって得られるベネフィットを比較しながら、オフライン決済の必要性も含めて検討を深めていくことが求められる。