2025 年 10 月 17 日
日本銀行決済機構局

ＣＢＤＣフォーラム ＷＧ７
「基本機能の事務フロー」
第６回会合の議事概要

１．開催要領

（日時）2025 年９月３日（水）14 時 00 分～16 時 30 分
（形式）対面形式及びＷｅｂ会議形式

２．日本銀行からの説明

事務局から、「ワーキンググループ（ＷＧ７）【基本機能の事務フロー】第６回会合日本銀行説明資料」（以下、事務局説明資料）に基づいて、実験用システムにおけるＣＢＤＣ送金にかかる共通の処理フロー（以下、ＣＢＤＣの送金処理フロー）を説明。これまでの会合で実施した議論内容およびエラーハンドルに関する論点を提示した（事務局説明資料は別添１を参照）。

３．プレゼンテーションおよびディスカッション

日本銀行からの説明に続き、株式会社三菱ＵＦＪ銀行および株式会社ＮＴＴデータの２社によるプレゼンテーションが行われ、その後、ディスカッションを実施。概要は以下のとおり。

（１）送金処理におけるエラーハンドリングの紹介（株式会社三菱ＵＦＪ銀行）
――　プレゼンテーション資料は別添２を参照。

本プレゼンテーションでは、当行における銀行間の送金処理のエラーハンドリングについて、銀行間の送金処理の大まかな流れと、障害発生箇所別のエラーハンドリングを説明のうえ、ＣＢＤＣの送金処理フローとの違い等から論点を提示する。

まず、インターネットバンキングを用いた銀行間の送金処理の大まかな流れを示す。本プレゼンテーションでは、各経路で電文途絶等のエラーが発生する際にどのようなエラーハンドリングを行うかといった点に焦点を当てる。

次に、各経路で行っているエラーハンドリングを説明する。電文途絶等のエラーが発生するタイミングが、仕向銀行の勘定系システムによるお客さまの

口座引き落とし処理完了前であれば取引中断となり、完了後であれば障害対応を行い前進処理する流れとなる。なお、被仕向銀行に電文が到達しているが振込先口座が確認できない等のエラーとなった場合には組み戻し電文が仕向銀行宛に発信される。この場合、組み戻し電文はこれまでの送金ルートを辿って仕向銀行に到達し、一部事務処理を挟んだうえで、お客さまの口座に組み戻し資金を入金するといった対応となる。

最後に、銀行間送金とＣＢＤＣの送金処理フローの違いから論点を５点（①送金先の事前確認、②送金可能かどうかの確認、③送金予告の有無、④顧客管理及び台帳管理の分離、⑤中央銀行の介在）挙げる。このほかの論点もあるとは思うが、本会合においてはこれらの論点を中心に議論したい。

（２）質疑応答およびディスカッション

（参加者）着金側ユーザーと被仕向銀行フロントシステムの間の電文途絶時のシステム処理として取引中断が挙げられているが、着金側ユーザーが自身の残高を確認できていない状態が続いた場合、何かの事象を契機に着金した取引が巻き戻るという意味ではないと理解して良いか。

（プレゼンタ）ご認識のとおり。あくまで着金側ユーザーのスマートフォン等の画面上で残高の確認ができない状況を指す。画面上では、「ただいまお取り扱いできません」のような表示を行い、時間を置いてから操作いただくように促す。

（参加者）仕向銀行の勘定系システムでお客さまの口座から資金を引き落とした後、被仕向銀行に電文が到達するまでの間で電文が途絶した場合、どのシステムでどのようなエラーが起きているかといった全体の処理を把握する主体は存在しないという理解でよいか。

（プレゼンタ）ご認識のとおりである。そもそも、被仕向銀行はデータを受領しない限り、自行に送金が実行されていることを知る術がない。また、仕向銀行勘定系システムと全銀システム間で障害対応が必要になった場合は、仕向銀行と全銀システムで、データの授受をどのようにして行うか議論し対応する。なお、仕向銀行でエラーを検知しても被仕向銀行にデータが到達している場合もある。このため、エラー発生時には基本的にまず仕向銀行が被仕向銀行に到達したか否か確認を行い、到達していない場合には全銀システムと協議するといった流れで、対応を検討することになる。

（参加者）仕向銀行がエラーにも関わらず、被仕向銀行に到達している状況について、どのように確認を取るのか。

（プレゼンタ）個別に被仕向銀行と連絡を取り合い、ログ上でデータの到達有無を確認していただく。データが到達しているのであれば、インターネットバンキング等を経由し、処理が出来ている旨をお客さまに伝達する等、仕向銀行のエラー自体への対応を行う。送金処理に関しては、正常に進行しているのであれば特段対応することはない。

（参加者）全銀システムと被仕向銀行勘定系システム間で障害が起きた際に、被仕向銀行に対して障害の発生を通知する仕組みはあるか。

（プレゼンタ）おそらく、自動的に障害の発生を通知する仕組みはなく、障害連絡網に沿って連絡されるのではないだろうか。なお、被仕向銀行に到達したのち、振込先口座を確認できないエラーが発生することもある。これは業務エラーとして日常的に発生し得るものであり、障害対応ではなく自動で組み戻し電文を送信するというエラーハンドリングになっている。

（参加者）今のケースで、プレゼンテーションの際に一部事務処理を行うと説明があったが、先ほどの回答で説明のあった自動処理との関係について確認したい。

（プレゼンタ）組み戻しの電文自体は被仕向銀行から自動で発出される。組み戻しの電文が仕向銀行に到達したのち、お客さまの口座に入金する前に一度処理は止まる。この際、組み戻し理由等がコードとして電文に組み込まれて返ってくるため、この内容を目視で確認のうえ事務処理を行うことでハンドリングするものと理解している。

（参加者）説明にあった「複数取引が同時に発生した場合等の考慮」とは何を示しているか。

（プレゼンタ）現状では、当行はフロントシステムおよび勘定系システムの両方を行内で管理しており、各システム間でリアルタイムに同期を行っている。この実務に照らせば、顧客管理システムと台帳管理システムでリアルタイムに同期を行う必要があるかもしれない。もっとも、ＣＢＤＣのシステム構成においては顧客管理システムと台帳管理システムを別の仲介機関で管理していると

仮定するならば、各システム間でリアルタイムに同期できない可能性がある。この場合、顧客管理システムが認識している状態を、台帳管理システムでは認識していないことになるかもしれない。複数取引が同時に発生した場合等にも、このリアルタイム性をどこまで確保できるか論点になり得るだろう。この論点において、各システムを同一主体内で管理しているか否かがポイントになると考える。

（参加者）仕向銀行勘定系システムと全銀システムの間で発生した障害と、全銀システムと被仕向銀行勘定系システムの間で発生した障害について、障害発生時はお客さまにどのような通知を行うのか。ＣＢＤＣでもバックエンド側で障害が発生した際には、ユーザーに何らかの通知が必要であると想定できるため、通知方法の検討も必要と考える。

（プレゼンタ）お客さまへの通知は、フロントシステムから勘定系システムに到達したタイミングを取引完了として、お客さまのインターネットバンキング等の画面上に取引成立の表示を行う。このため、仕向銀行勘定系システムと全銀システムの間で障害が発生しても、前進処理を行うのみであり、お客さまに改めて通知することはない。全銀システムと被仕向銀行勘定系システムの間で組み戻しが発生した場合は、インターネットバンキングのトップページ等にメッセージを表示するようなことは考えられるが、各行のサービスに拠るかもしれない。

（参加者）仕向銀行のフロントシステムと勘定系システムの間の障害の場合、フロントシステムではエラーとなり、勘定系システムでは処理が進行しているような事例はあるか。

（プレゼンタ）存在する。エラーが発生すると、関連するシステム同士でエラー内容について確認を取り合う。勘定系システムから全銀システムへ電文が発出されているログを確認次第、お客さまが二重取引をしないようにコールセンター等を経由して連絡を行う。当該事例はプログラム上予期できない範囲であり想定外エラーとなる。このため、自動でエラーハンドリングを行う形ではなく個別対応を行う。

（参加者）このような、フロントシステムではエラーと認識しながら前進処理をするケースは、事務によるリカバリ対応が大変だと推察される。

（参加者）提示いただいた論点について、顧客管理システムが送金先口座の事前確認や送金可否確認の負担を少なく行うことができるのであれば、台帳管理システムで確認する必要はないのではないだろうか。なお、このような確認を行ううえで、顧客管理システムにどのようなデータを持たせるか論点になるだろう。

（参加者）銀行間送金の処理フローとＣＢＤＣの送金処理フローを比較すると、例えば取引履歴を顧客管理システムで保持し、この取引履歴を基に各種制限の判定をしながらトランザクションを制御することは、銀行間送金の仕組みと異なり論点となるだろう。既存の銀行間送金では、フロントシステムで取引履歴を含むユーザーに関するマスタ情報は持たず、各種制限値を管理すると聞く。この実態を踏まえると、取引履歴を用いて各種制限の判定を行うと処理が重くなるのではないかと推察する。このため、各種制限の判定を行う際は、顧客管理システムで取引履歴を用いた判定を行うのではなく各種制限値のようなものを設けて管理することが望ましいかもしれない。そのほか、エンドポイントデバイスとの接点になる顧客管理システムでマスタ情報を保持することはセキュリティリスクの懸念が想定されるため、セキュリティ面で何らかの対応が必要になるのかもしれない。

（参加者）ＣＢＤＣの流通量と発行総量の整合性チェックについて、中央銀行ではＣＢＤＣの発行総量を何らかの形で把握していると想定するが、これは各台帳管理システムのＣＢＤＣ残高の合算値と同額となるはずである。仮に、中央銀行で把握している発行総量と各台帳管理システムのＣＢＤＣ残高の合算値が不一致となった場合、責任の所在を確認することが困難ではないだろうか。台帳管理システムが分散するほど、この難易度は上がるだろう。

（参加者）銀行間送金のシステム構成とＣＢＤＣ送金のシステム構成では、フロントシステムと勘定系システムの管理主体、および顧客管理システムと台帳管理システムの管理主体がそれぞれ同一か否かが大きな違いになるだろう。現状の銀行間送金のシステム構成では、管理主体が同一銀行であるが、ＣＢＤＣ送金のシステム構成では管理主体が異なり得ると仮定されており、この管理主体の分離が処理の難易度を高めているように感じる。同一主体が顧客管理システムや台帳管理システムを管理しながら、プライバシーを保護することも検討の一案ではないだろうか。現状の分散システムでは各システムの整合性を確保しながら即時性を高めることは非常に難易度が高いだろう。

（３）民間決済ネットワークにおけるエラー制御を踏まえたＣＢＤＣシステムへの示唆について（株式会社ＮＴＴデータ）

―― プレゼンテーション資料は別添３を参照。

本プレゼンテーションでは、ＣＢＤＣの送金処理フローを前提として、当社が提供しているクレジットカード決済の処理を参考に論点を提示し、課題解決に向けたアプローチを考察する。

まず、ＣＢＤＣの送金処理フローのエラーについて、発生が送金先増額記帳の前か後かで、大きく２つに分類できる。後者のエラーは、さらに２つのケースに整理できる。１つ目は、送金元の顧客管理システムが応答電文を返せないケース（以下、ケース①）で、２つ目は、送金元の台帳管理システムが応答電文を返せないケース（以下、ケース②）である。両ケースにおいて、送金側ユーザーと着金側ユーザーがともに決済が完了したか否か分からず、再実施等により二重決済を行ってしまう可能性があり、何らかの制御が必要になる。

次に、ケース①およびケース②に対して、既存のクレジットカード決済の処理を参考に比較を行う。ＣＢＤＣシステムにおける構成要素とカード決済における構成要素を関連付けた場合、端末センタと決済端末間のエラーが先ほどのケース①と対応し、決済ネットワークセンタと端末センタ間のエラーが同様に先ほどのケース②に対応すると考えられる。

こうした前提の下、まずＣＢＤＣ送金完了後の「取消」処理について検討する。クレジットカード決済における誤った取引についてはカードセンタでオーソリゼーションの許可処理後であっても、決済は完了していないため「取消」処理を行う事が出来る。他方でＣＢＤＣ送金の場合には、送金先の台帳管理システムの増額により決済完了となり、残高がすぐに利用されるリスクが存在するため「取消」処理は適さない。そのため、「取消」処理ではなく、「反対取引」処理が必要になるだろう。また、残高がすぐに利用されないように、送金先の台帳管理システムで「増額留保」を行い、送金先への応答通知後に端末発で残高を利用可能とする２段階での処理が考えられる。これにより、各システムが正常応答しない限り着金側の増額は次の支払に利用されないと保証できるのではないだろうか。もっとも、取引量を１万ＴＰＳ（Transactions Per Second、１秒あたりのトランザクション量）と仮定した場合に、再送や例外処理、接続維持の監視等、仲介機関が担う負荷は相応に高まることが想定できる。この点、システム間にネットワークシステムのようなハブを設けることが、ＣＢＤＣシステム全体の安定性や運用効率の確保に寄与すると考えられるだろう。

次に、増額完了後のエラー処理として反対取引を明細に反映するか否かを検討する。ＵＩ／ＵＸに自由度を持たせるためにも、反対取引明細を表示させ

ないモードと、反対取引も含めて全明細を表示するモードの切り替えを行える設計とする案が考えられるだろう。
　最後に、ディスカッションテーマとして、反対取引開始から終了までの間に残高が利用されるリスクへの対処と、反対取引のユーザーフィードバックのアプローチの２点を挙げる。このあとのディスカッションにて意見交換を行いたい。

（４）質疑応答およびディスカッション
（参加者）端末センタで処理全体のタイムアウト管理を行っているのか。

（プレゼンタ）既存のクレジットカード決済においては、端末センタは処理全体のタイムアウトを管理しておらず、各システムが外部にリクエストを発信する際に、それぞれ応答までの時間を管理している。

（参加者）ＣＢＤＣの送金処理フローの代案について、顧客管理システムの処理負担が大幅に増加する点が懸念点となるだろう。この懸念点は、ネットワークシステムが軽減し得ると理解したが、ネットワークシステムが単一障害点になり得る等、新しい論点も出てくると想定できる。この点はいかがか。

（プレゼンタ）ご認識のとおり。そもそも、各システムが分散されている構成において、複数の仲介機関が複雑に組み合わさることを想定すると、各システムが相互に通信して処理を行うことは現実的に相当困難ではないかと考える。ネットワークシステムを経由して処理を行うと難易度は下がるのかもしれない。なお、ネットワークシステムを経由する場合、例えば、ネットワークシステムが全面停止するなど、顧客管理システムを跨った処理が出来なくなることはあり得るだろう。こうした事態を想定して、対処方法を検討する必要があると考える。

（参加者）本プレゼンテーションにおける考えられるエラーケースは、送金先台帳管理システムの増額が行われて決済完了となったのち、送金元台帳管理システムに完了通知が到達する前提となっていると理解した。今回の議論の対象外かもしれないが、送金先台帳管理システムでは増額完了していながら、送金元台帳管理システムに完了通知が到達しないケースも考え得るだろう。この場合、送金先台帳管理システムが増額完了となり、この残高を次の取引に使用できることになる一方、送金元台帳管理システムがエラーと判断することになる。このとき送金元台帳管理システムではロールバックの処理が進行するので

はないだろうか。すなわち、送金先の残高は増額のうえ使用され、送金側の減額留保が元の残高に戻ることも可能性として起こり得る。整合性の確保を検討するうえではこの点の考慮も重要になると考える。

（プレゼンタ）解像度を上げればご指摘のパターンも有り得るが、本プレゼンテーションでは台帳管理システムから発出される応答電文以降の処理フローを考察するため、台帳間は整合するものとし、議論の対象外とした。

（参加者）プレゼンテーションいただいた処理フローについて、ネットワークシステムのようなハブの設置によって、安定性や運用面、ハンドリング面に関する利点があることは承知している。他方、システムリソースの観点では、顧客管理システムの負担がネットワークシステムによってどれほど効率化されるかは疑問である。システム負荷がネットワークシステムに集中するのみで、全体の処理負担は減少しない可能性があるのではないだろうか。

（プレゼンタ）システムリソースについては、ネットワークシステムを設けてもトランザクションの総量が減ることはなく負荷を絶対的に減らせるものではないだろう。一方、各システムが各種制御や処理を行ううえで、ネットワークシステムを設置することで顧客管理システムや台帳管理システムが個々に負担する処理量は相対的に減少し得るのではないかと考える。なお、負荷が集中する処理をネットワークシステムに負担させることができるのであれば、各仲介機関は高度な仕組みを構築せずに運用できるという利点も考え得るのではないだろうか。そのほか、各仲介機関が独自に各システムを管理して運用する難易度を考慮すれば、ネットワークシステムを経由することで通信方法や障害時対応等を統一的なルールで運用し易くなり、統制等も図り易くなるのではないだろうか。

（参加者）ＣＢＤＣの送金処理フローの代案について、送金先の台帳管理システムの増額留保を解除したタイミングで決済完了となり、システム的には、この処理以降であれば着金側ユーザーは増額分の残高を使用できるようになると理解した。他方、着金側ユーザーのエンドポイントデバイスでは、いつ残高が増額されたと把握できるのか。着金側ユーザーへの通知は増額留保を解除する前と後に存在しているため、どちらかで把握できると理解している。

（プレゼンタ）増額留保を解除した後の通知で把握することになるだろう。これを着金側ユーザーにどのように分かり易く伝えるか検討の余地があると考え

ている。

（参加者）着金側ユーザーは、増額留保を解除した後の通知を受けて、取引成立および決済完了と捉えるものと理解した。一方、増額留保を解除したタイミング以降ではシステム的には残高が使用でき、着金側ユーザーが増額された残高を認識するタイミングと時間差が生じることになる。増額留保が解除されてから、着金側ユーザーに通知されるまでの間で障害が発生すると、残高が使用可能になったか否かを着金側ユーザーが確認できず、ＣＢＤＣの送金処理フローと同様にユーザー間で二重払いをするかもしれないといった論点等が発生してしまうのではないだろうか。

（プレゼンタ）ご理解のとおりであり、そこで増額留保を解除する前に着金側ユーザーに通知するフローが必要になると考えた。そもそも、ＣＢＤＣの送金処理フローでは、次に挙げる３つの要素を同時に実現することが求められるのだろう。１つ目は、台帳管理システム間の整合性を必ず確保すること。２つ目は、取引を一定の時間内で完了させること。３つ目は、増額された残高は増額された時点で使用可能となること。しかし、これらをすべて同時に実現するには、非常に高速かつ緻密に処理を実行する以外の方法はなく、実現が困難ではないだろうか。そこで、クレジットカード決済を事例に、増額された残高をすぐに使用可能とせずに、一時的に留保することも一案ではないかと考えた。これは、まず、増額留保を行った状態で取引の成否をユーザーに通知し、次に、増額留保を解除してユーザーに決済完了を通知するという二段階の処理に分けるということである。前者のタイミングで着金側ユーザーは取引成立を把握できる。そして後者の処理を高速で行えば、増額された残高をすぐに使用できるようになるため、３つの要素を同時に実現することに近くなるのではないだろうか。

（参加者）決済完了をユーザーが認識するタイミングについて、送金元ユーザーには増額留保を行った状態で通知され、送金先ユーザーには増額留保を行った状態での通知と増額留保を解除した後の通知の二段階であるため、このような時間差はユーザーが混乱する可能性があるだろう。増額留保解除後の通知が必要であれば、同タイミングで送金元ユーザーにも通知する必要があるのかもしれない。

（参加者）
ＣＢＤＣ送金の処理フローの代案について、３つの要素を同時に成立すること

は困難であるとのことだが、受領したＣＢＤＣが即時で使用可能とならない場合も、数秒程度であれば許容範囲ではないだろうか。

（参加者）送金フローにおいて、台帳側で残高が使えるタイミングとユーザーが増額されたことを知るタイミングとの間に差が生まれる箇所についてどのように考えるか、難しい問題だと理解している。

（参加者）何らかの理由で二重払い等が起きてしまった場合、増額した着金側ユーザーの残高に対し、自動的に反対取引を行うような処理は考え得るか。

（プレゼンタ）増額された残高が使用されていないのであれば処理としては反対取引を自動で行うことは可能かと推察するが、銀行間送金の組み戻しのような、戻す場合もあれば戻さない場合もある現在の業務運用から想像すると、自動実行は難易度が高いのではないか。

（参加者）仮に、ＣＢＤＣ送金のエラー対応として取消ではなく反対取引が必要だとしても、反対取引をユーザーに見せる必要はないのではないだろうか。

（参加者）エラーハンドリングに主眼を置いて議論を進めるならば、システム構成を単純化するかシステム自体を単純化することに帰着するのではないだろうか。なお、システムが分散されて管理主体が多く関与する前提に立てば、処理の複雑さが増すと考えられるため、プレゼンテーションいただいた、ネットワークシステムのようなハブの設置が一案になるだろう。

４．次回予定

次回会合で取り扱うテーマについて、日本銀行より概要を説明（別添４参照）。次回の会合は 2025 年 11 月５日（水）に開催予定。

以　上

別添１

ワーキンググループ（WG7）
【基本機能の事務フロー】
第６回会合
日本銀行説明資料

2025年9月

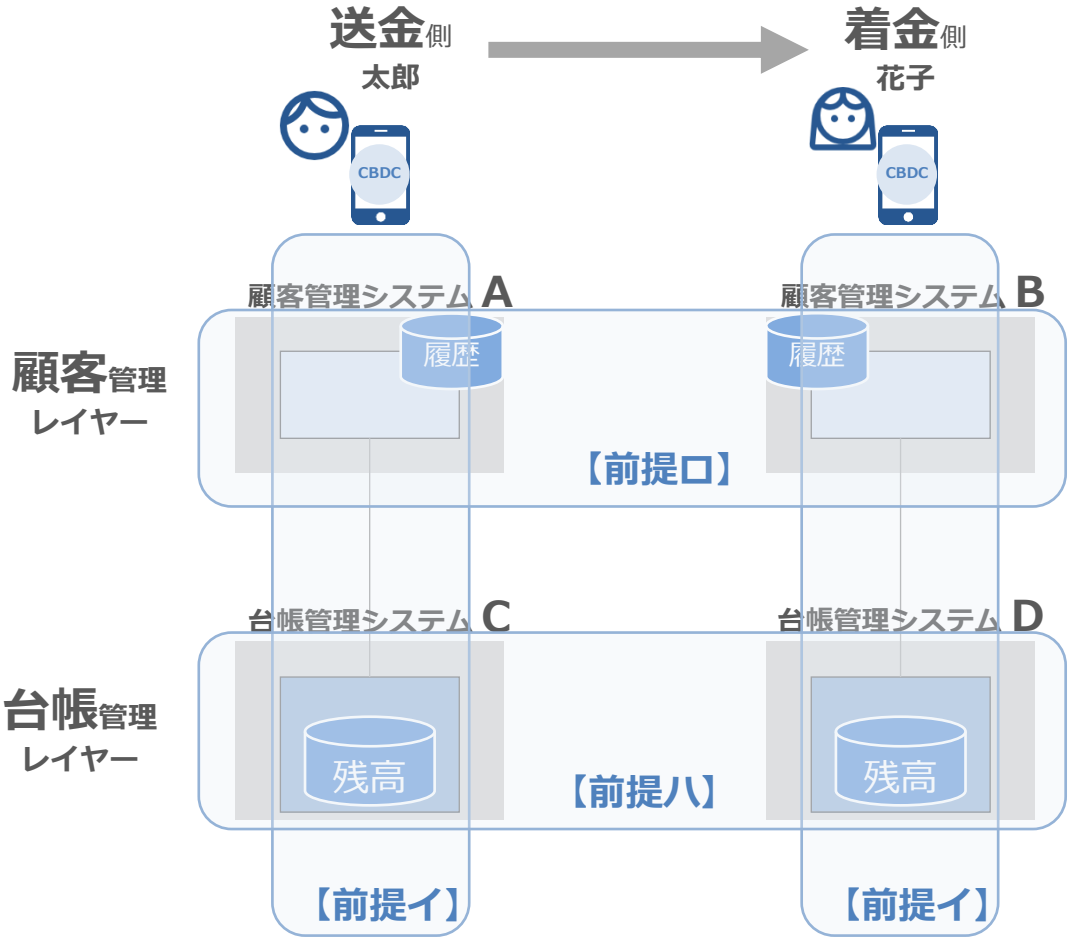
日本銀行 決済機構局

CBDC送金にかかる共通の処理フローとエラーハンドル

【再掲】セットアップ：議論の前提

【バックグラウンド】

- 非同期通信（リソースの有効活用）
- 実装容易性の観点から、処理を一筆書きで実装
- 各主体は独立した機関として振る舞う
- 台帳設計パターン２（台帳がCとDに分離）にて実験用システムを構築



【議論の前提】

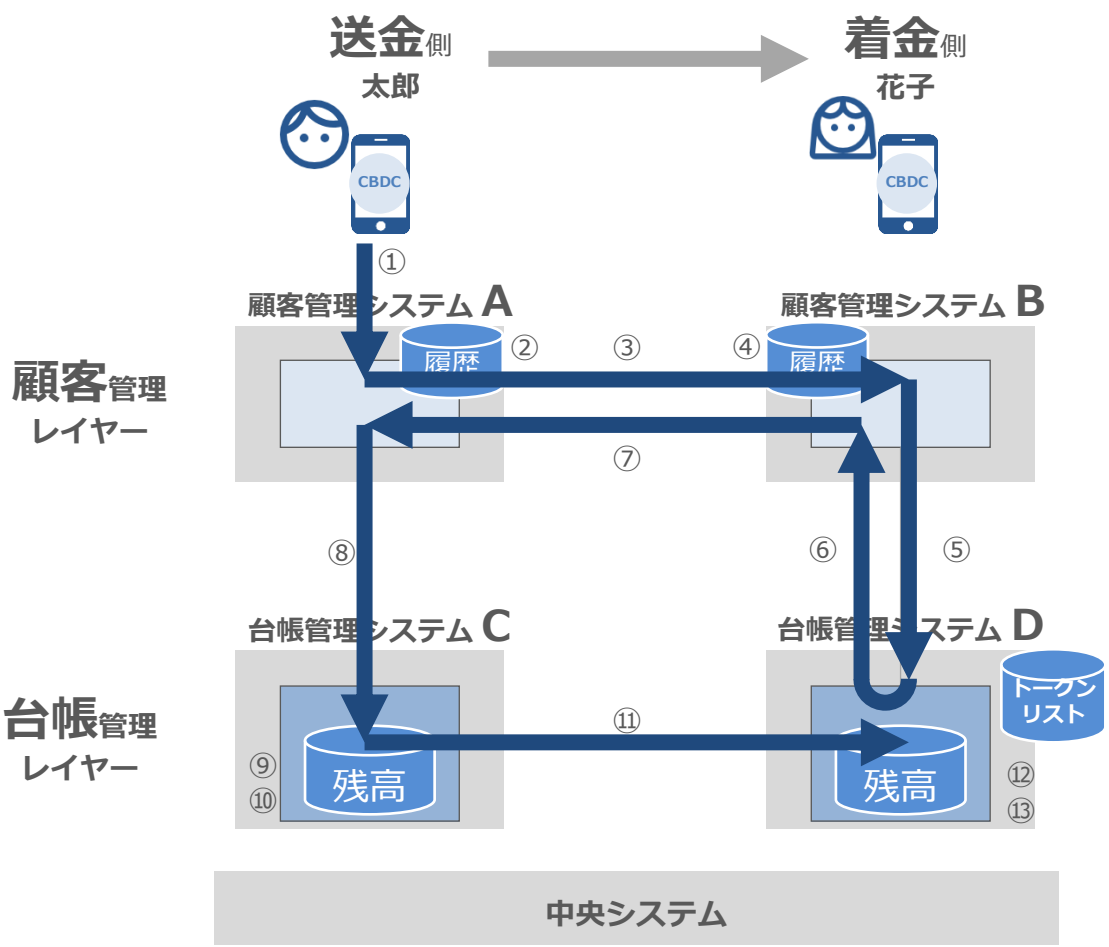
- **【前提イ】**：台帳を更新する際には、**顧客管理システムからの指示**が必要
  - 顧客管理システムの取引可否の判断のうえ台帳更新する
- **【前提ロ】**：台帳を更新する前に、**顧客管理レイヤーにて取引の是非の判断**（各種取引制限判定、AML/CFT等）を実施する
  - 顧客管理レイヤーでの取引の判断の後、台帳管理レイヤーでの処理を行う
- **【前提ハ】**：複数の台帳を更新する際には、**台帳間において整合性を取る必要**
  - 「台帳間の整合性」とは、Cの減額とDの増額がセットになるようにすること（例：Cの減額だけされているという状態を極力短くする）
  - 今回は、CとDとの間で、中央集権的な主体を仮定せずに整合性を担保する（コレオグラフィ*の考え方を援用）方法を考えた

*分散システムにおけるサービス連携のための制御方法の一つで、イベントドリブンのフロー管理（メッセージのやり取り）で業務を実現するもの。これに対する概念として、中央集権的に全体フローを管理する方法である「オーケストレーション」の考え方がある

現時点でCBDCを発行する計画はなく、CBDCに関して前提や仮定を置く場合も、実験用システムやWGの議論のための仮置きに過ぎない

CBDCの送金にかかる共通の処理フロー

- 共通の処理フローについて、「送金先の台帳を増額記帳するまで」の流れは以下のとおり。



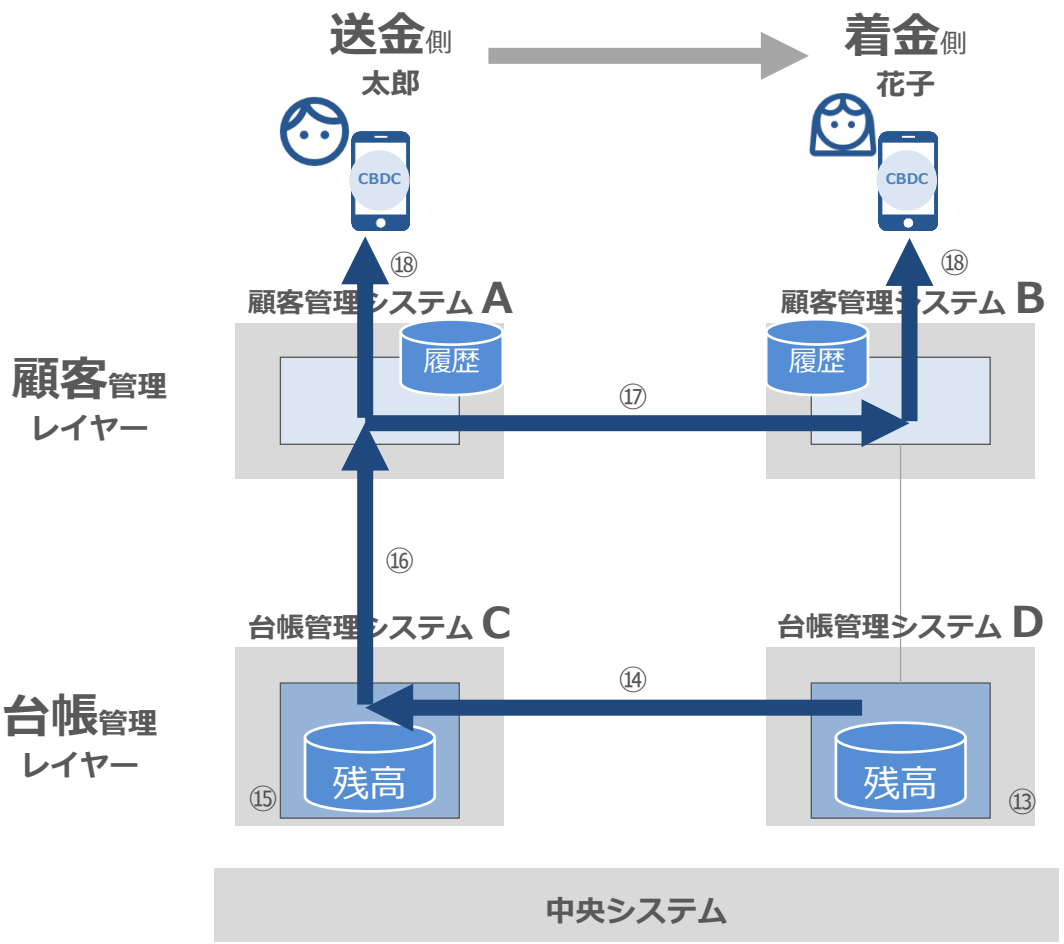
	行為者	処理内容
①	太郎	Aに送金指示
②	A	取引履歴を確認し、太郎が送金可能かを判定
③	A	送金先（花子）を特定し、Bに送金を通知
④	B	取引履歴を確認し、花子が受取可能かを判定
⑤	B	Dに台帳更新許可トークンを発行依頼
⑥	D	Bに更新許可トークンを発行
⑦	B	Aに了解を返送
⑧	A	Cに送金指示
⑨	C	送金にあたって残高が不足していないか太郎の台帳を確認
⑩	C	太郎の台帳を留保付で減額記帳
⑪	C	Dに増額指示
⑫	D	⑥と⑪を突合のうえ、保有上限確認
⑬	D	花子の台帳を増額記帳（決済ファイナル）

※この図において②以降は送金にかかる共通の処理フローを示しているが、「①送金指示」は順送金の処理フローのものを例示している。

現時点でCBDCを発行する計画はなく、CBDCに関して前提や仮定を置く場合も、実験用システムやWGの議論のための仮置きに過ぎない

CBDCの送金にかかる共通の処理フロー

- 共通の処理フローについて、「送金先の台帳を増額記帳するまで」の流れは以下のとおり。



	行為者	処理内容
⑬	D	花子の台帳を増額記帳（決済ファイナル）
⑭	D	Cに完了を通知
⑮	C	太郎の台帳の留保を取る
⑯	C	Aに完了通知
⑰	A	Bに完了通知
⑱	A、B	Aは太郎宛、Bは花子宛に取引の完了を通知

現時点でCBDCを発行する計画はなく、CBDCに関して前提や仮定を置く場合も、実験用システムやWGの議論のための仮置きに過ぎない

日本銀行における検討と問題意識

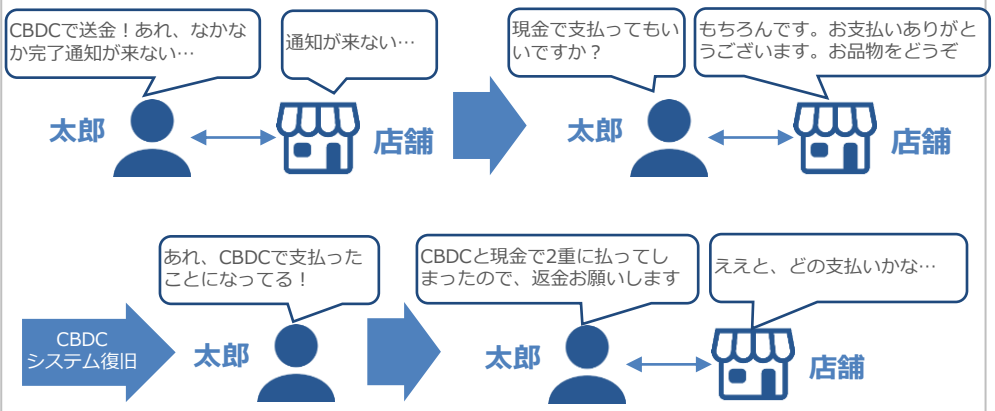
- 日本銀行における検討と問題意識

【検討】

- 日本銀行ではまずは台帳管理部分に関するエラーハンドリングを検討した。結果として、通信に時間がかかることで、処理完了まで時間がかかるケースは考えられるものの、最終的に台帳間で整合性を保つことは可能であるとした。
- 他方、顧客管理部分やエンドポイントデバイスまで考慮した際には、異例時対応が生じると考えられる。

【問題意識】

- 例えば、支払い時にCBDCシステムに障害が発生すると、決済完了通知が個人・店舗になかなか届かず、個人・店舗が待ちきれずに別の手段で支払いを済ますことが想定される。その際、台帳内の送金が完了した後の障害であると、（別の手段で支払い済みにも関わらず）復旧後にCBDCシステムから決済完了通知が個人・店舗に届き、店舗から個人に返金する必要が生じる。



ユーザーは通信中等の状況が続き、決済完了したか不明

➢ **上述の問題意識について、市中の決済サービス（銀行振込・口座振替・キャッシュレス決済等）ではどのようにエラーハンドリングしているのか、本会合にて深掘りしたい。**

現時点でCBDCを発行する計画はなく、CBDCに関して前提や仮定を置く場合も、実験用システムやWGの議論のための仮置きに過ぎない

世界が進むチカラになる。

MUFG

別添２

三菱UFJ銀行
送金処理における
エラーハンドリングの紹介

2025年9月3日

三菱UFJ銀行　システム企画部
瓦谷 佳祐、辻本 皓亮

目次

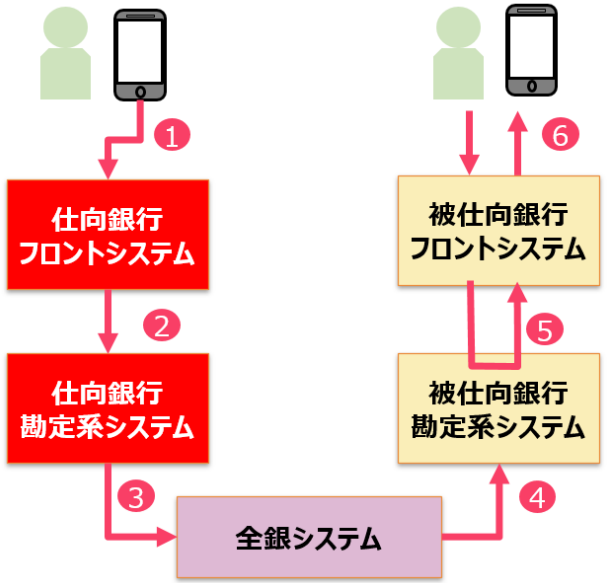
目次に戻る

銀行間の送金イメージ

概略

- 銀行間で送金を行う際、仕向銀行送金側のフロントシステムからバックエンドに存在する勘定系システムへ送金指示を行う
- 送金指示を受けた勘定系システムは、仕向口座の引き落とし処理を実施
- 仕向銀行から全銀システムを経由して被仕向銀行宛に電文を送信することで、振込処理を実現している

イメージ図



※フロントシステム・・・インターネットバンキング等

障害発生箇所（電文が届かない箇所）を以下に分けて説明（後述）

① お客さま端末(仕向)と仕向銀行フロントシステムの間

② 仕向銀行フロントシステムと仕向銀行勘定系システムの間

③ 仕向銀行勘定系システムと全銀システムの間

④ 全銀システムと被仕向銀行勘定系システムの間

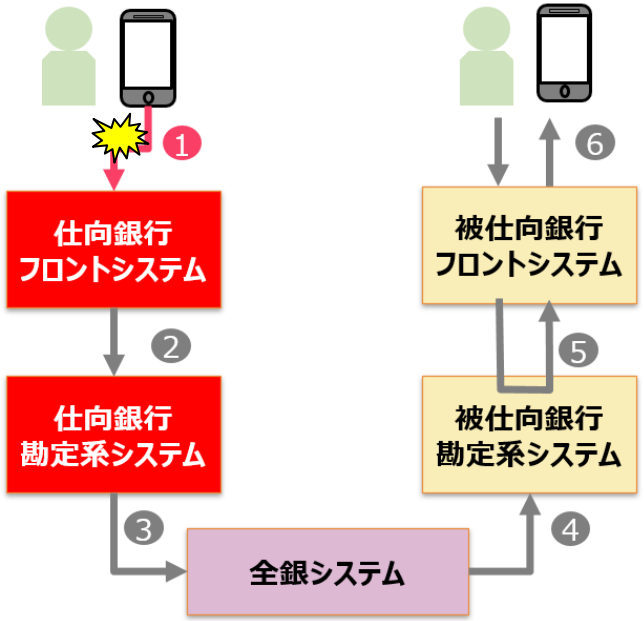
⑤ 被仕向銀行勘定系システムと被仕向銀行フロントシステムの間

⑥ 被仕向銀行フロントシステムとお客さま端末(被仕向)の間

目次に戻る

障害発生箇所別のエラーハンドリング①

①お客さま端末(仕向)と仕向銀行フロントシステムの間



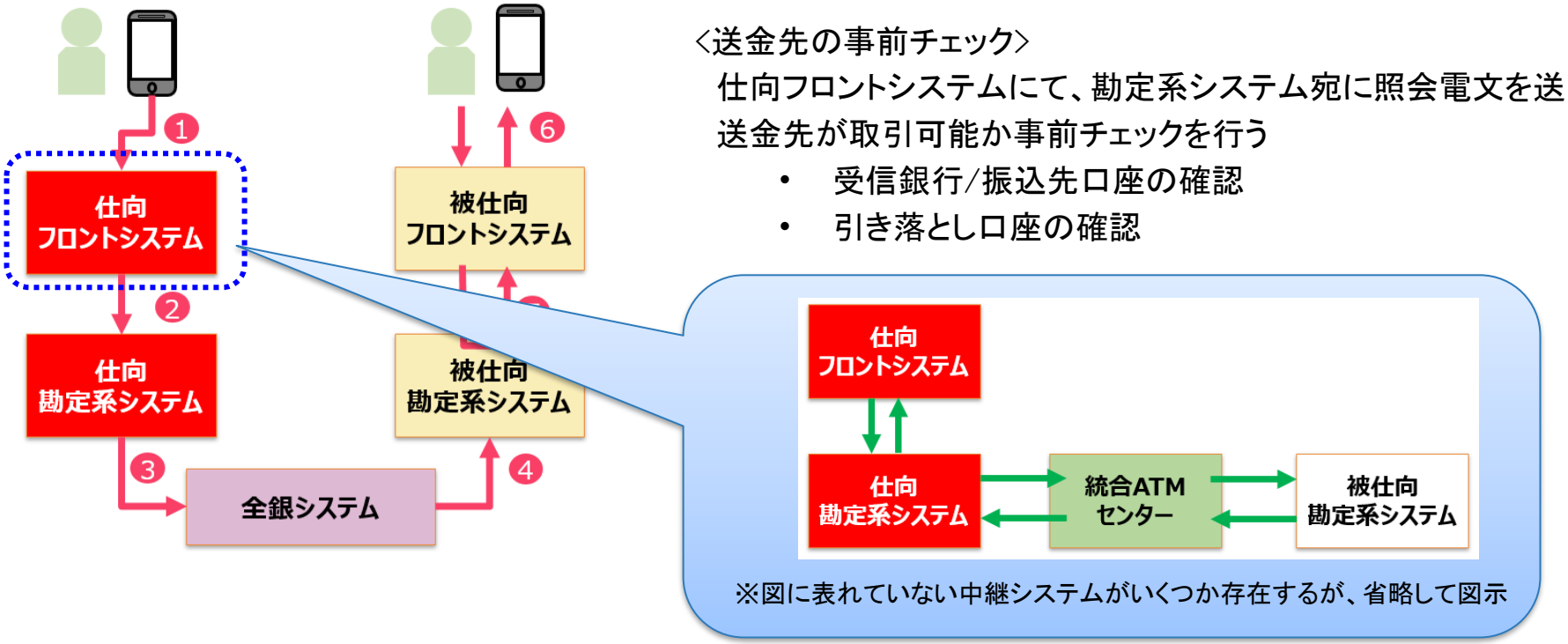
〈電文途絶時のシステム影響〉
送金指示が当行宛に届いていないため、システム上は影響なし

〈電文途絶時のシステム処理〉
- 顧客宛にエラー画面を応答
- 取引中断

目次に戻る

(参考)仕向フロントシステムでの送金先チェック

①お客さま端末(仕向)と仕向銀行フロントシステムの間



〈送金先の事前チェック〉

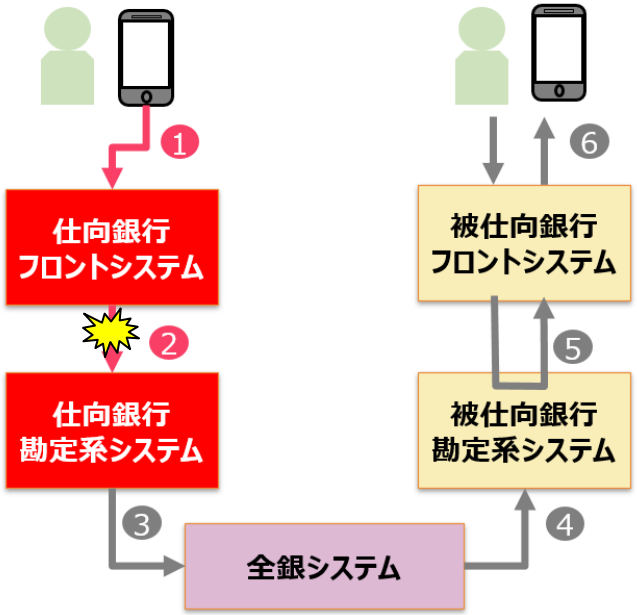
仕向フロントシステムにて、勘定系システム宛に照会電文を送り、送金先が取引可能か事前チェックを行う

- 受信銀行/振込先口座の確認
- 引き落とし口座の確認

目次に戻る

障害発生箇所別のエラーハンドリング②

②仕向銀行フロントシステムと仕向銀行勘定系システムの間



〈電文途絶時のシステム影響〉

- 送金指示は仕向銀行宛に届いている
- 仕向銀行勘定系システムに電文が届いておらず、口座引き落とし処理は未済（不整合は発生せず）

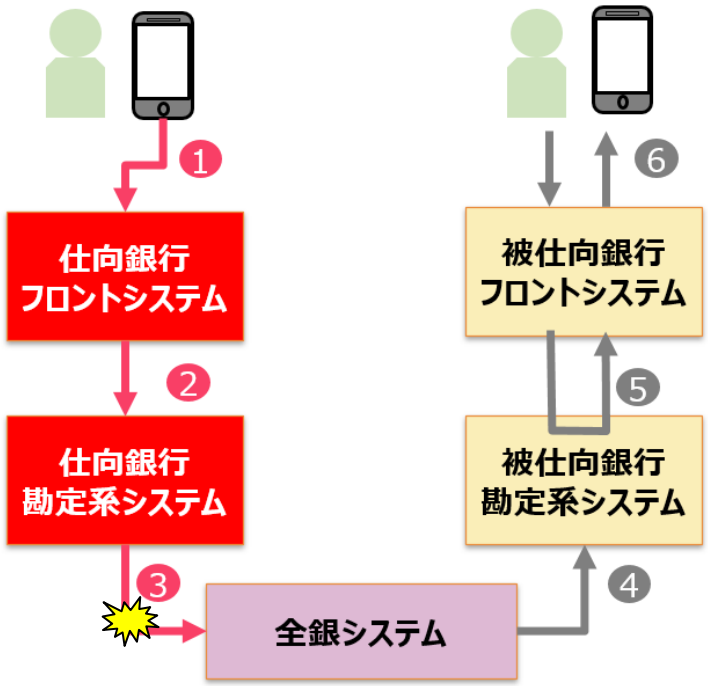
〈電文途絶時のシステム処理〉

- 顧客宛にエラー画面を表示
- 取引中断

目次に戻る

障害発生箇所別のエラーハンドリング③

③仕向銀行勘定系システムと全銀システムの間



<電文途絶時のシステム影響>
- 仕向銀行勘定系システムにて、口座引き落とし処理は完了
- エラー発生時は、仕向銀行・被仕向銀行での不整合を避けるため、障害対応を行い、処理を前進めする方針
- 全銀システム宛に電文は届いていない

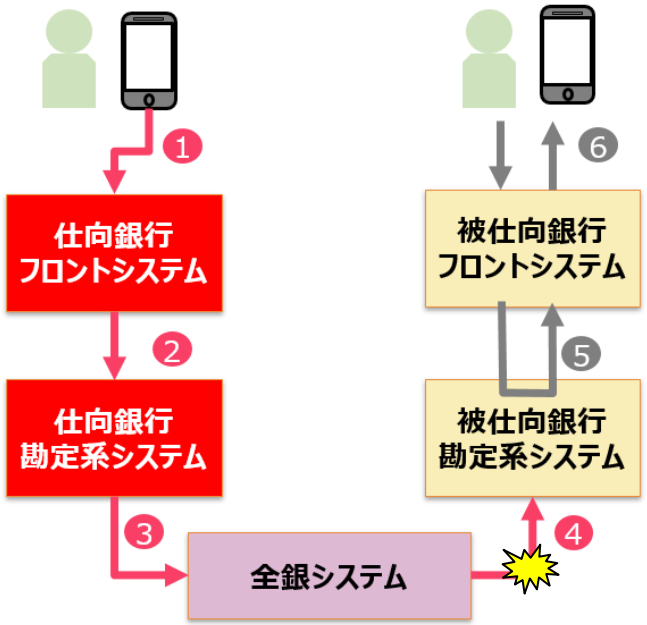
<電文途絶時のシステム処理>
- データを自動再送
  （自動再送も失敗する場合、障害対応を行う）

<障害対応>
- 手動でデータ送信処理を起動
- 手動のデータ送信処理も不可の場合、コマンドによるファイル送信
- ファイル送信も不可の場合、
  テープ媒体をTAXIで運び、物理的に持ち込む

目次に戻る

障害発生箇所別のエラーハンドリング④

④全銀システムと被仕向銀行勘定系システムの間



〈電文途絶時のシステム影響〉

- 全銀システム宛に電文は届いている
- 被仕向銀行宛に電文は届いていない
  （仕向側と整合性があっていない状態）

〈電文途絶時のシステム処理〉

- 全銀システム側にてエラーハンドリング

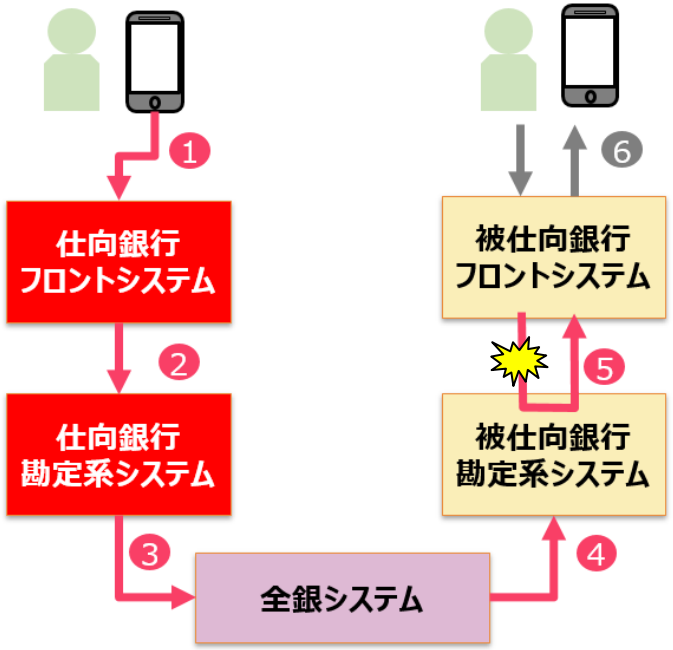
〈業務的なエラー発生時のシステム処理〉

- 振込先口座が確認できない等のエラー発生時は、
  組み戻し処理を実施

目次に戻る

障害発生箇所別のエラーハンドリング⑤

⑤被仕向銀行勘定系システムと被仕向銀行フロントシステムの間



〈電文途絶時のシステム影響〉

- 被仕向銀行勘定系システム宛に電文は届いているため、着金処理は完了（不整合は発生せず）

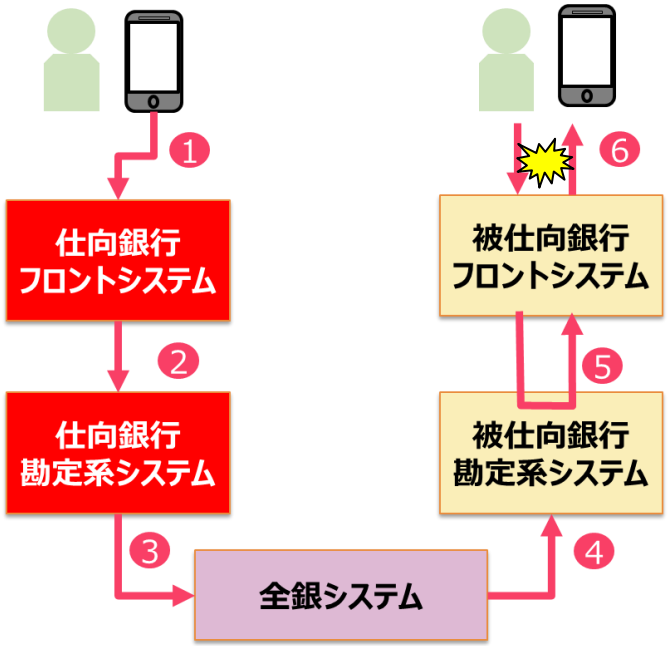
〈電文途絶時のシステム処理〉

- 被仕向フロントシステムにて、残高が取得できず、顧客宛に、残高を非表示マークで表示した画面を応答

目次に戻る

障害発生箇所別のエラーハンドリング⑥

⑥被仕向銀行フロントシステムとお客さま端末(被仕向)の間



〈電文途絶時のシステム影響〉

- 被仕向銀行勘定系システム宛に電文は届いているため、着金処理は完了（不整合は発生せず）

〈電文途絶時のシステム処理〉

- 顧客宛にエラー画面を応答
- 取引中断

目次に戻る

ディスカッションポイント

論点

#	銀行間送金との相違点	当行仕様(現行銀行間送金)	CBDC仕様（想定）	ディスカッションポイント
1	送金先の事前確認	フロント・勘定系の両方で確認	顧客管理システムでのみ確認	【台帳管理システム上での事前確認要否】 台帳管理システム間での事前確認は不要で良いか（複数取引が同時に発生した場合等の考慮）
2	送金可能かどうかの確認データ	勘定系システム(あるいは類似のホストシステム)にて管理	顧客管理システムでのみ確認	【台帳管理システム上での送金可能確認要否】 台帳管理システム間での送金可能確認は不要で良いか（複数取引が同時に発生した場合等の考慮）
3	送金予告の有無	予告なし	台帳管理システムにてトークン発行	【送金予告の要否】 トークン発行後、ロストやキャンセル等が発生した場合の挙動について考慮が必要
4	顧客管理及び台帳管理の分離	システムは分離されているが同一組織内で管理	別事業者の可能性有	【顧客に紐づく台帳の整合性チェック】 顧客管理にて保有する取引履歴と台帳管理上の台帳データの整合性をどう担保するか
5	中央銀行の介在	日銀による介在有	介在無し	【CBDC流通量の整合性チェック】 中央銀行を介在しないことにより、流通するCBDCと実際に発行されているCBDCの整合をどう担保するか

MUFG

Beyond Finance®

みらいの社会のつくり手に

NTT DATA

別添3

Discussion Paper Confidential

【WG7】第6回会合 民間決済ネットワークにおけるエラー制御を踏まえたCBDCシステムへの示唆について

2025年09月03日
株式会社NTTデータ
第三金融事業本部　決済ITサービス事業部

アジェンダ

Beyond Finance®
みらいの社会のつくり手に

1 プレゼンの目的と前提となる考え方

今回のプレゼンの目的と、弊社の説明において、前提となる考え方をご説明します。

2 CBDC実験用システムを例とした課題認識

CBDC実験用システムの店舗利用における送金のケースを例に、今回の課題のポイント・運用への影響を整理します。

3 課題へのアプローチかかる弊社事例紹介と論点抽出

弊社の決済サービス提供経験と照らして、課題へのアプローチの参考となる事例をご紹介し、論点を抽出します。

4 抽出論点を踏まえたCBDCシステムにおけるアプローチ例

抽出論点を踏まえ、CBDCシステムにおけるアプローチについて、案ベースの例をあげて、考察します。

5 ディスカッションテーマ

弊社が検討した各論点のアプローチ案を踏まえ、案の問題点や異なるアプローチなどを議論させてください。

NTT DATA

01

プレゼンの目的と前提となる考え方

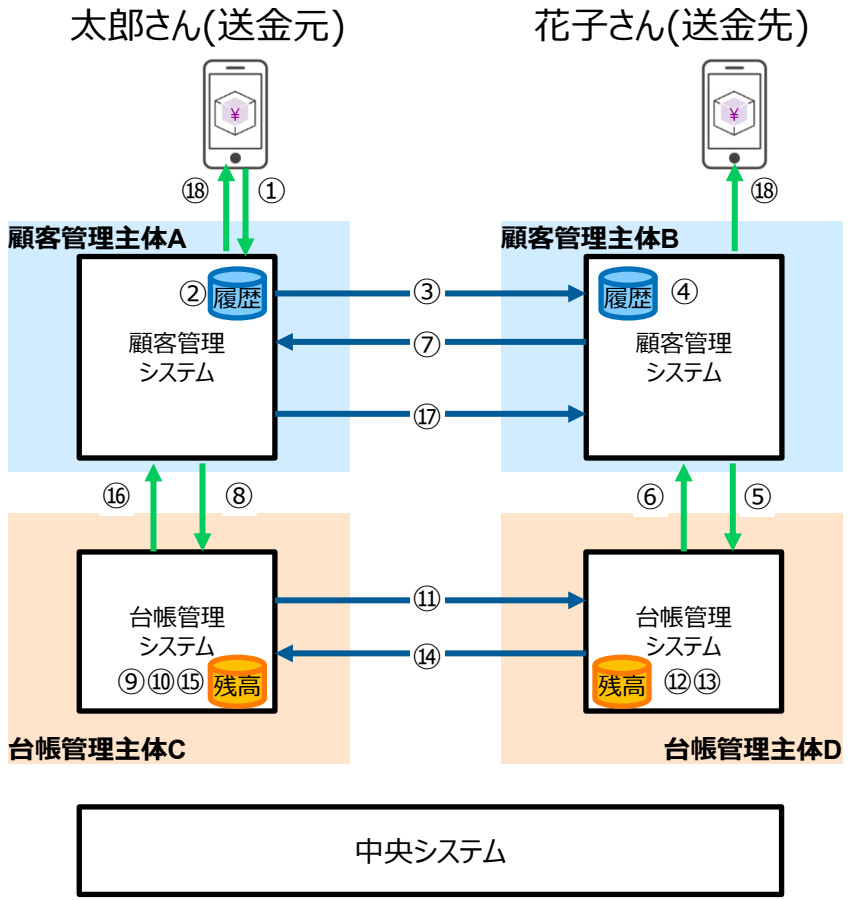
1.1. 題材とするユースケースについて

Beyond Finance®
みらいの社会のつくり手に

実験用システムにおける以下の代表ケースを題材に、エラーケースを抽出します

題材とするユースケース例

送金元アプリで「振込」を行い
「送金元への応答」と「送金先への通知」で相互に目視確認



	行為者	処理内容
①	太郎	Aに送金指示
②	A	取引履歴を確認し、太郎が送金可能かを判定
③	A	送金先（花子）を特定し、Bに送金を通知
④	B	取引履歴を確認し、花子が受取可能かを判定
⑤	B	Dに台帳更新許可トークンを発行依頼
⑥	D	Bに更新許可トークンを発行
⑦	B	Aに了解を返送
⑧	A	Cに送金指示
⑨	C	送金にあたって残高不足していないか太郎の台帳を確認
⑩	C	太郎の台帳を留保付で減額記帳
⑪	C	Dに増額指示
⑫	D	⑥と⑪を突合のうえ、保有上限確認
⑬	D	花子の台帳を増額記帳（決済ファイナル）
⑭	D	Cに完了を通知
⑮	C	太郎の台帳の留保を取る
⑯	C	Aに完了通知
⑰	A	Bに完了通知
⑱	A、B	Aは太郎宛、Bは花子宛に取引の完了を通知

NTT DATA

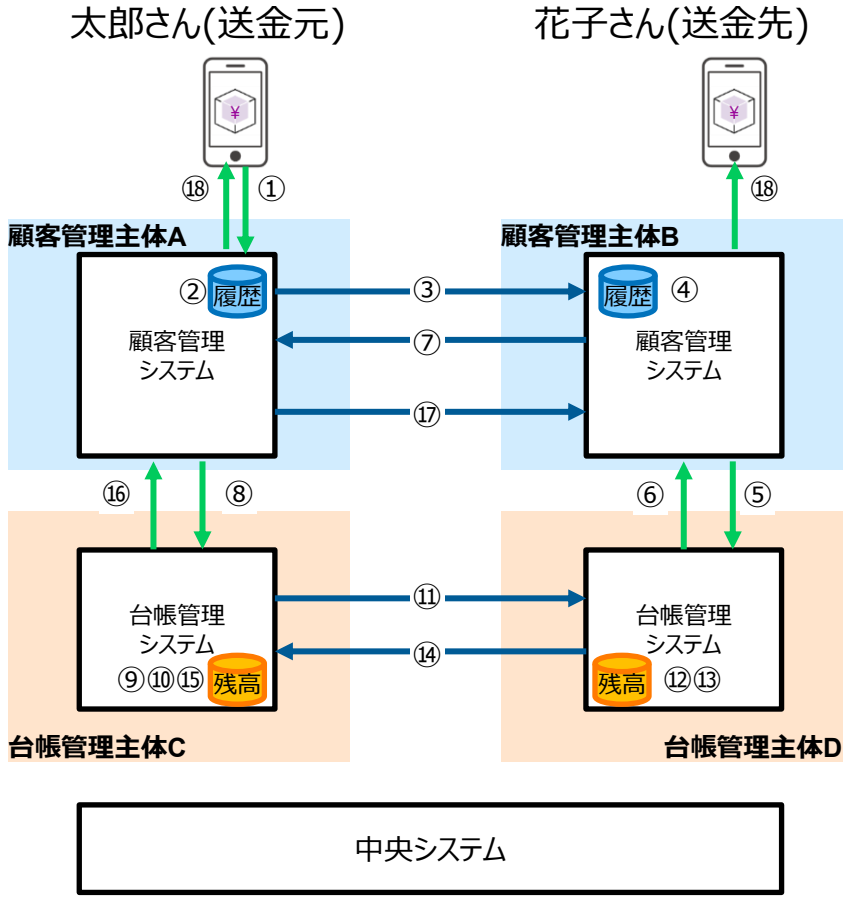
1.2. プレゼンの目的と前提となる考え方

Beyond Finance®
みらいの社会のつくり手に

以下の視点を出発点として、考察を経て、論点を抽出し、アプローチの方法や視点を拡げることを目的としています

題材とするユースケース例

送金元アプリで「振込」を行い
「送金元への応答」と「送金先への通知」で相互に目視確認



出発点とする視点・考え方

1 送金元・送金先双方の確認を前提

対面での決済シーンにおいては、送金元と送金先の双方のアプリにて、正常に決済処理が完了した旨の通知を確認することで、完了する流れを前提とする

2 オフライン決済機能は考慮しない

エラーの発生により、オンラインでの取引結果が曖昧な場合でも、アプリ側でオフラインで取引を扱い、あとでオンラインで同期をとって、結果整合性を確保するなどの発想も考えられるが、今回は、オンライン処理のみで、整合性を確保する前提とする

3 台帳更新後の結果の通知は、顧客管理主体Aを起点に行う

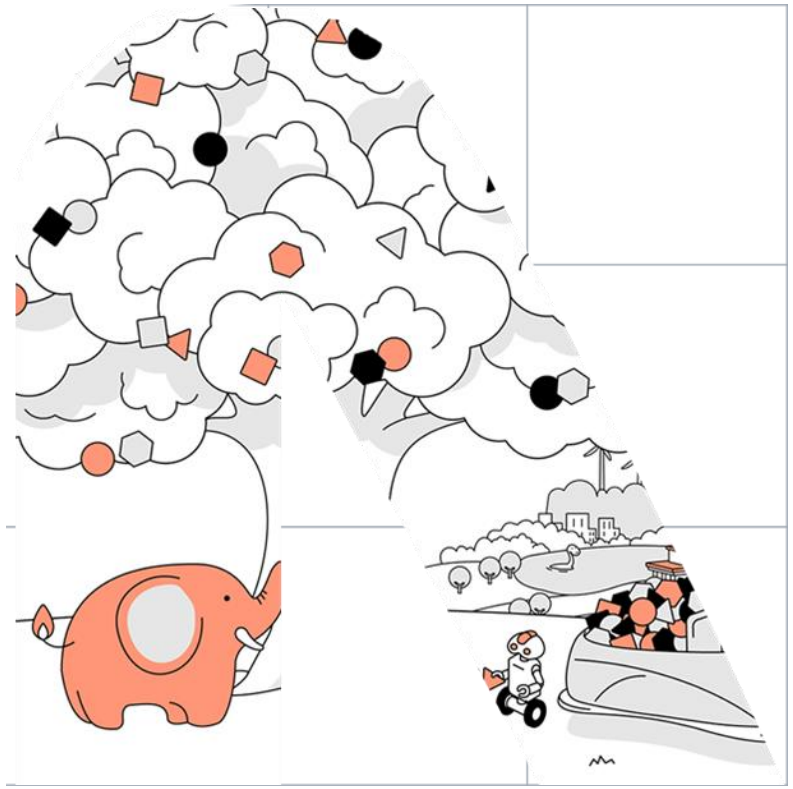
処理手順のとおり、結果の通知は顧客管理システムAを起点に行うものと考え、顧客管理システムAが正常に処理できない場合は、顧客管理システムBにもその影響が波及するものと考える

4 送金先に入金された残高は、即時に利用される可能性がある

⑬で入金完了し、ファイナリティが確定した後は、不可逆であり、残高は即時利用される可能性があることを前提とする

NTT DATA

02

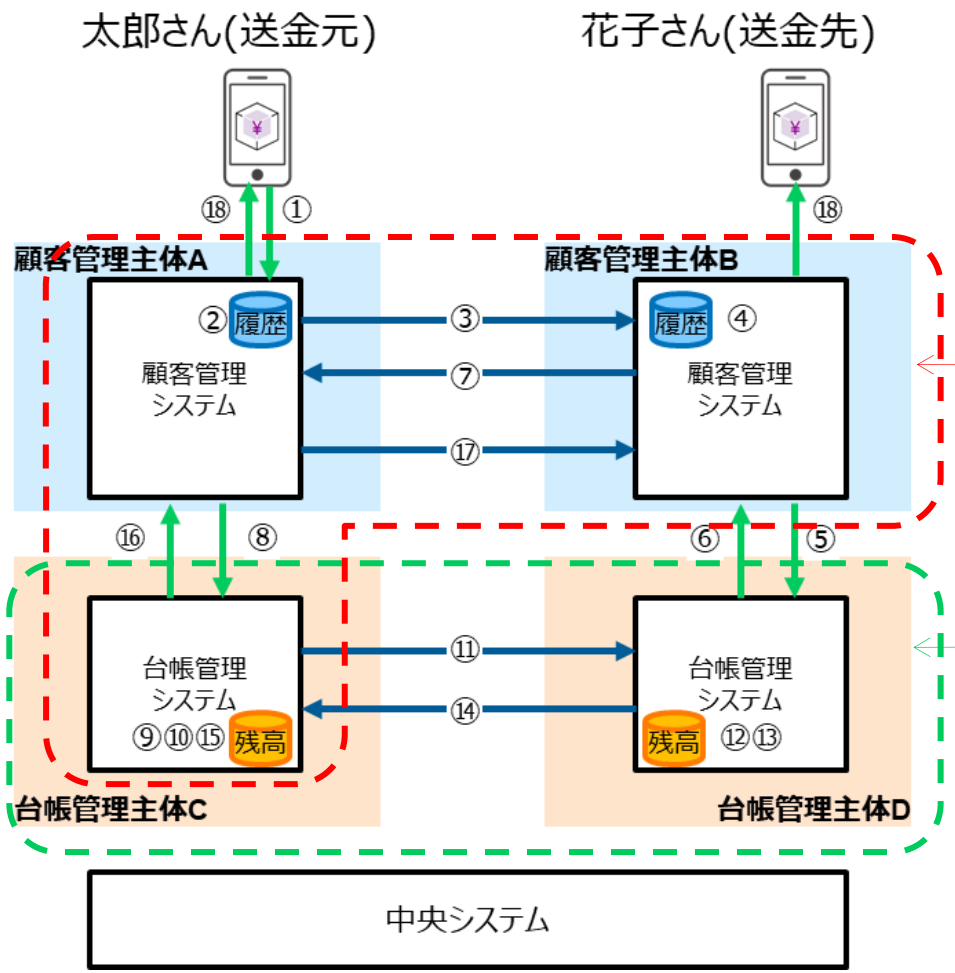


CBDC実験用システムを例とした課題認識

2.1. 考えられるエラーケースについて（1/2）

Beyond Finance®
みらいの社会のつくり手に

ロールバック可否が分岐する時点で2つに分けたうえで、①のケースに着目して、不整合になるケースを抽出する



❶ 送金先増額記帳以降に発生するエラー

ファイナリティが確定しており、不可逆なためロールバック不可

不整合な状態が発生するケースがある。

なお、エラー時には、⑰⑱は、エラー通知が行われるものと考える

❷ 送金先増額記帳前までに発生するエラー

ロールバックで対処することで、整合確保可能

NTT DATA

2.1. 考えられるエラーケースについて（2/2）

Beyond Finance®
みらいの社会のつくり手に

下記3パターンの不整合となるエラーの発生が想定され、障害発生箇所で分類した場合、2ケースに大別されます

⑬台帳管理システムD	⑭→	台帳管理システムC	⑯→	顧客管理システムA⑱	⑰→	顧客管理システムB⑱	ステータス
花子の台帳を増額記帳（決済ファイナル）	Cに通知	太郎の台帳の留保を取る	Aに通知	Bに完了通知後、太郎宛に通知	Bに通知	花子宛に取引の完了を通知	
完了通知可（増額完了）	→	Dから完了通知を受領/Cが完了通知	→	Cから完了通知を受領/Aが完了通知	→	Aから完了通知を受領	完了
	→		→	Cから完了通知を受領/Aが完了通知しない	→	Aから完了通知を受領しない	エラー
	→	Dから完了通知を受領/Cが完了通知しない	→	Cから完了通知を受領しない/Aがエラー通知	→	Aからエラー通知を受領	エラー
	→		→	Cから完了通知を受領しない/Aがエラー通知しない	→	Aからエラー通知受領しない	エラー

Cから通知がこないが、Aが適切にエラーハンドリング

【ケース①】
顧客管理システムA
の障害

【ケース②】
台帳管理システムC
の障害

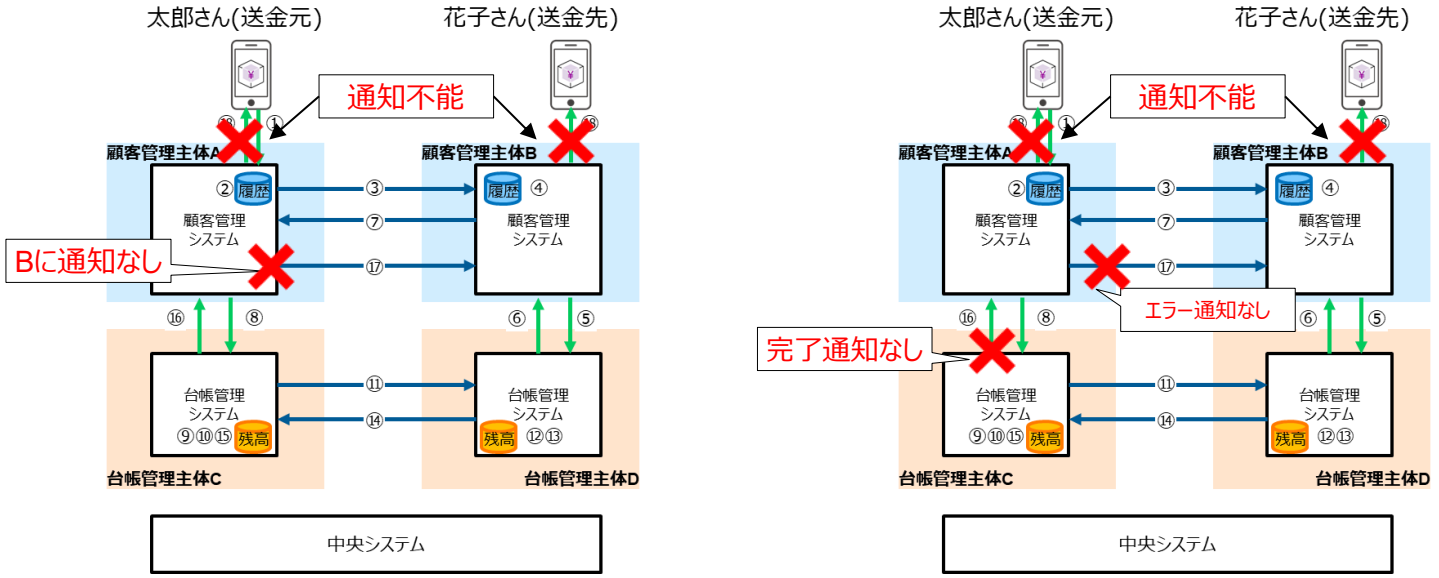
NTT DATA

2.2. CBDC実験用システムを例とした課題認識

Beyond Finance®
みらいの社会のつくり手に

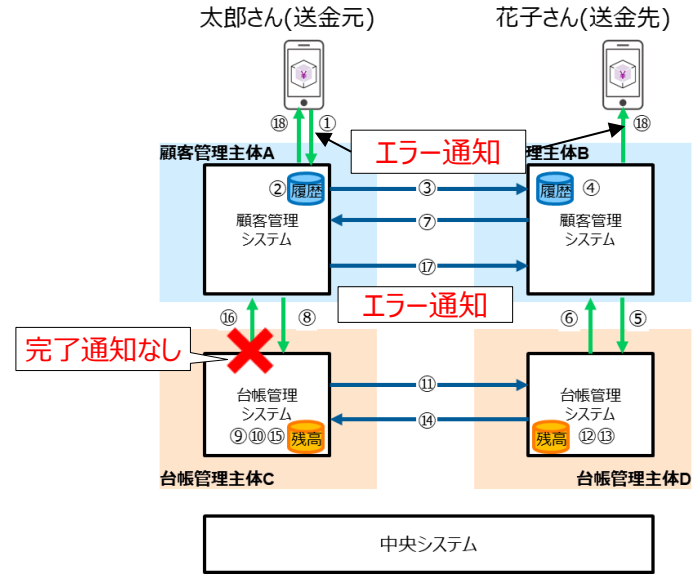
いずれのケースも、現場で再実施をした場合2重決済となる影響が懸念されるため、これに対処する制御が必要

【ケース①】顧客管理システムAの障害



【課題】
決済完了後、結果を通知する前に障害が発生した場合に、どのように対処するか
【運用への影響】
- 仕向・被仕向ともに通知がいかないため、決済されたか否か確認できず、商品の受け渡しが完了しない
- 現場で再実施した場合、2重決済となりえる

【ケース②】台帳管理システムCの障害



【課題】
決済完了しているが、エラー通知が発生することにより、台帳状態とクライアントへの通知内容が不整合となる
【運用への影響】
- 仕向・被仕向ともにエラー通知がくるため、決済されていないと判断される。
- 現場で再実施した場合、2重決済となりえる

NTT DATA

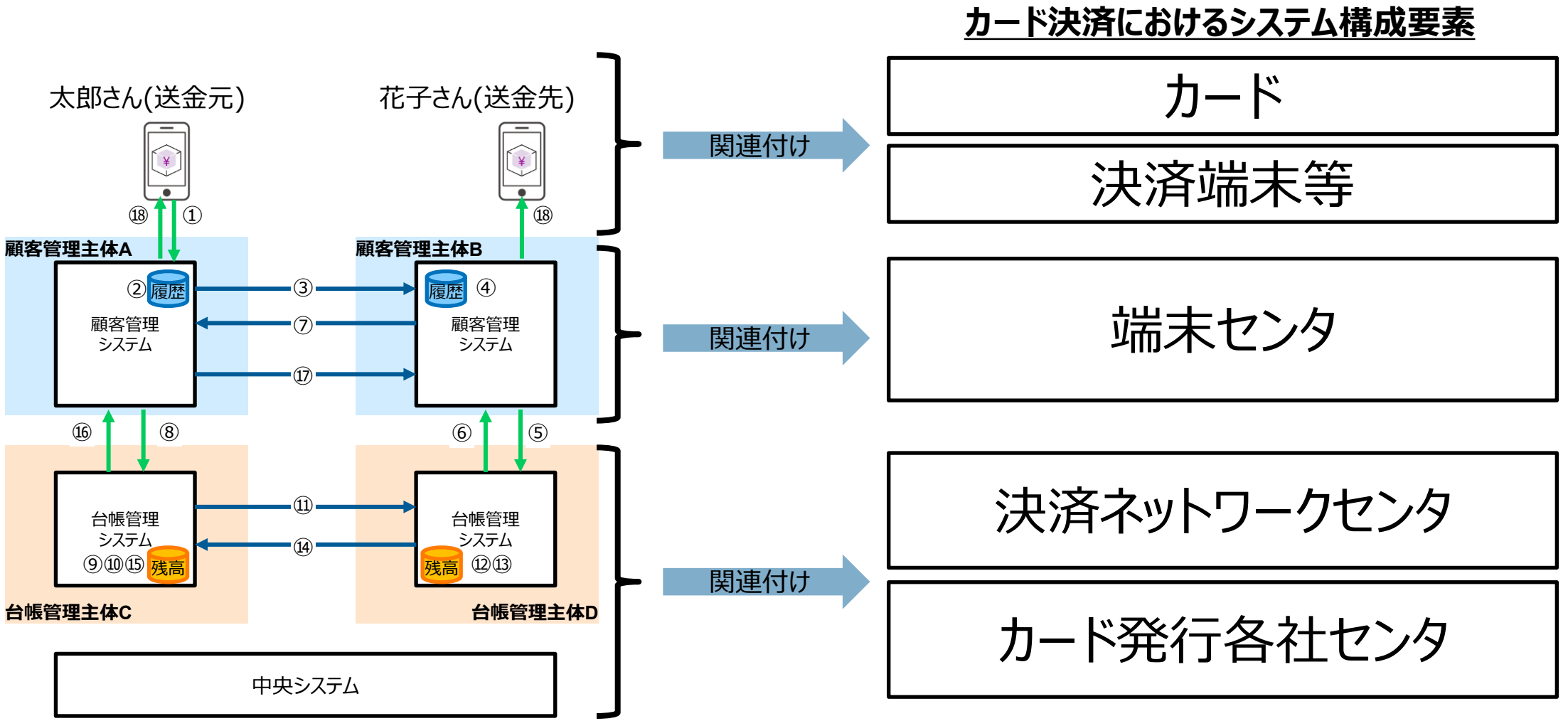
03

課題へのアプローチかかる
弊社事例紹介と論点抽出

3.1. CBDCシステムとカード決済システム構成への関連付け

Beyond Finance®
みらいの社会のつくり手に

民間のカード決済処理を構成するネットワーク上の各構成要素にCBDCシステムの構成要素を関連付けます



NTT DATA

3.2. カード決済における正常処理のシーケンスとエラーケースの対応

Beyond Finance®
みらいの社会のつくり手に

カード決済における正常シーケンスにおいて、CBDCシステムの各エラーケースを対応付ける場合、下記のとおり

投影のみ

3.4. NWセンタより結果受信後、端末センタから決済端末に正常応答不能

Beyond Finance®
みらいの社会のつくり手に

決済端末にエラー応答を返却することで失敗を通知し、非同期処理で取消処理を実行することで整合性を確保

投影のみ

3.5. NWセンタより結果受信できず、端末センタが決済端末にエラー応答

Beyond Finance®
みらいの社会のつくり手に

同様に決済端末にエラー応答を返却することで失敗を通知し、非同期処理で取消処理を実行することで整合性を確保

投影のみ

3.6. その他民間システムにおけるエラーハンドリングの工夫点（1/2）

Beyond Finance®
みらいの社会のつくり手に

取消指令は、必ず完了を確認する仕組みとなっており、万一完了せず処理経路を圧迫する場合は運用等で対処します

投影のみ

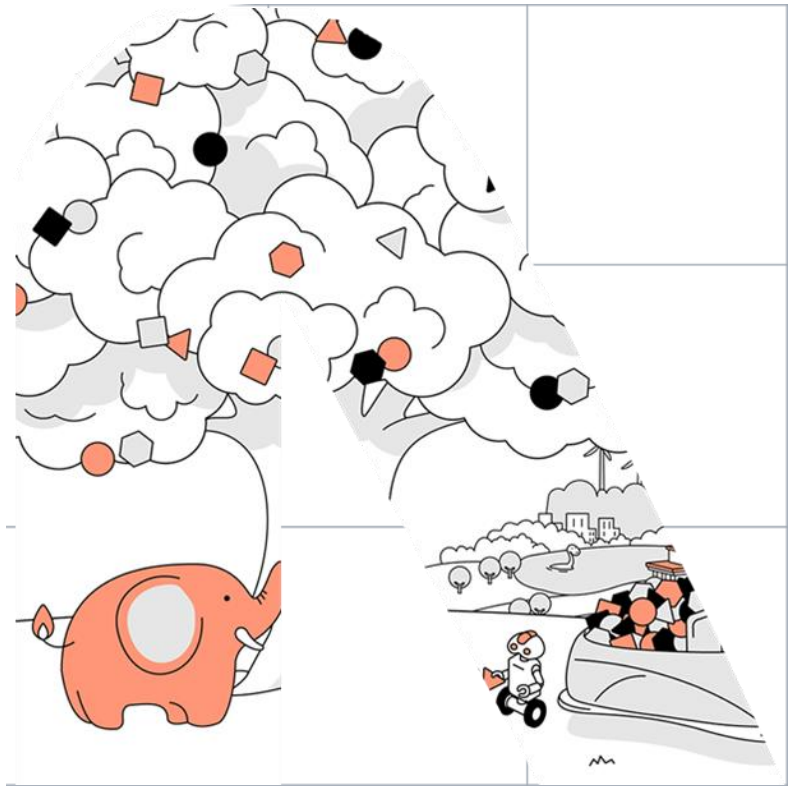
3.6. その他民間システムにおけるエラーハンドリングの工夫点（2/2）

Beyond Finance®
みらいの社会のつくり手に

センタ処理でOKとなったが、カード側でエラー判定となった場合に端末発で自動取消を行うケースもあります

投影のみ

04



抽出論点を踏まえたCBDCシステムにおけるアプローチ例

4.1. 抽出論点にかかる対応案に関する切り口（例）

Beyond Finance®
みらいの社会のつくり手に

抽出した論点に潜むバイアスを捉え、その逆の切り口からの発想で、エラーハンドリングを行う１案が考えられます

CBDCシステムに準えた場合の論点

1 決済完了後に「取消」はできるか

CBDCシステムの現状の処理フローでは、送金先への入金後は、不可逆であるため、**「取消」という操作は適さない**と考えられる。よって、「反対取引」としての取り扱いが適切

2 「反対取引」の非同期処理の可否

一方、反対取引を非同期で処理する場合において、入金後の残高を即時利用可能である性質を考慮したとき、**反対取引の開始～終了の間に、次の取引が発生する可能性も考慮**に入れて、整合性を確保に留意する必要。

3 エラー通知の発生

CBDCシステムの場合、送金元と送金先双方にエラー通知が送信される。送金先が、入金後残高を使うとすれば、**「この通知を受信してなお、次の取引を行う場合」や、「異なるクライアントから、同一CBDC口座の残高を利用する導線がある場合」に限定**される。

4 タイムアウト時の反対取引指令

端末センタが、顧客管理システムに該当することを考慮した場合、タイムアウト（一定時間経過しても応答がない）ケースを考慮した場合、この時点で、**自動で反対取引を発生させて整合性を確保する必要**があるのではないか

5 自動反対取引時のエラー通知内容

カード決済の場合、自動での取消指令はシステム制御上の処理であるため、カードホルダーが確認できる取引履歴上の明細には、記録されない。一方で、**CBDCでは、反対取引として記録される**ことになる。そして、タイムアウト時点では、取引の成否を、顧客管理システムは把握できない。つまり、タイムアウトした場合に、顧客管理システムがエラー通知を送信するときは、**ユーザーに対して、反対取引が発生するかもしれないし、発生しないかもしれないという、曖昧なフィードバックしかできない**。

論点を踏まえて必要な対処の方向性

「取消」ではなく、「反対取引」として台帳に記録

- ✓ 論点に沿って、**反対取引の操作として処理**する扱いとする

送金先通知後に端末発で残高を利用可能化する２段階処理

- ✓ エラー通知を受けても、次の取引を行う可能性が残存するというリスクを解消できないというバイアスを捉える。
- ✓ **「送金先が正常終了通知を受信後しか次の取引の残高を使えない」**という逆の切り口から、処理シーケンスの構造化を試行する。
- ✓ こうすることで、エラーが生じた場合に、**自動・非同期の反対取引でエラーハンドリングを行った場合も、カード決済システムと同様に整合性を確保できる**と考えられる。

アカウント視点とトランザクション視点の分離設計

- ✓ **「反対取引として記録された内容を、ユーザーに正確に通知する」という原則**に沿った場合に、**「エラー通知時点では、状況が正確に把握できない」**ゆえに、曖昧なフィードバックしかできないというジレンマをバイアスと捉える。
- ✓ **「アカウント視点（最終的な有効残高変動のみ表示）」と「トランザクション視点（全取引・反対取引も表示）」でユーザーへの取引のフィードバックの概念を分割**し、システム上取引明細に区分を設ける。
- ✓ これにより、区分に応じてUIを出しわける自由度が出せるため、よりユーザーフレンドリーなフィードバック方法を模索できる

NTT DATA

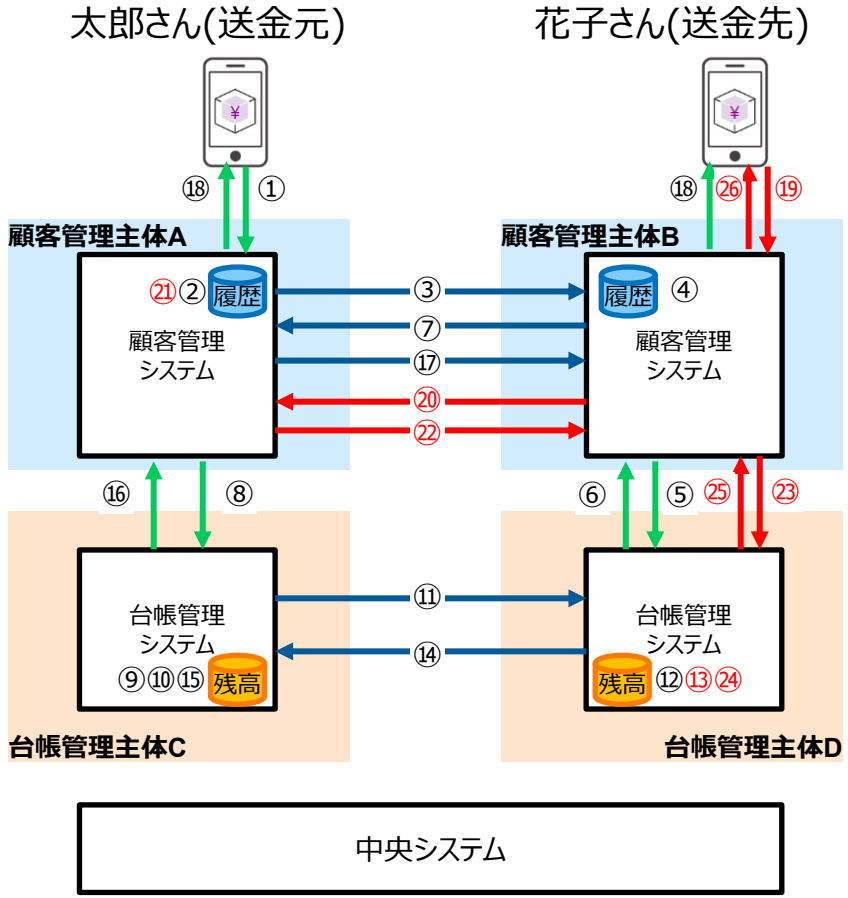
4.2. 送金先通知後に端末発で残高を利用可能化する２段階処理

Beyond Finance®
みらいの社会のつくり手に

送金先台帳を留保付で増額記帳し、通知受信後、送金元の通知完了を確認の上、留保を取る2段階処理を行う

題材とするユースケース例

送金元アプリで「振込」を行い
「送金元への応答」と「送金先への通知」で相互に目視確認



	行為者	処理内容
		①～⑫の処理は省略
⑬	D	花子の台帳を留保付で増額記帳
⑭	D	Cに完了を通知
⑮	C	太郎の台帳の留保を取る
⑯	C	Aに完了通知
⑰	A	Bに完了通知
⑱	A、B	Aは太郎宛、Bは花子宛に取引の完了を通知
⑲	花子	完了通知受信後、Backgroundの自動処理でBに残高利用確認要求を指示
⑳	B	Aに完了通知確認要求を依頼
㉑	A	送金元に完了通知の送信を正常に完了していることを判定
㉒	A	Bに完了通知確認応答を返送
㉓	B	Dに利用可能許可要求を依頼
㉔	D	花子の台帳の留保を取る（決済ファイナル）
㉕	D	Cに利用可能許可完了応答を返送
㉖	B	花子宛に残高利用確認要求の完了を通知

NTT DATA

4.3. 2段階処理の副作用（1/3）

Beyond Finance®
みらいの社会のつくり手に

経産省の試算式をもとに、CBDC利用状況に仮定を置いて、1秒当たり取引件数（Z）を概算

キャッシュレス比率（**50%仮定**）＝ キャッシュレス支払額（X＝**164.5兆円**） ／ 民間最終消費支出（**329兆円**） ※2024年度

キャッシュレス支払額（X＝**164.5兆円**）÷ 決済単価（**¥500仮定**）＝ 年間取引件数（Y＝**3,290億件**）

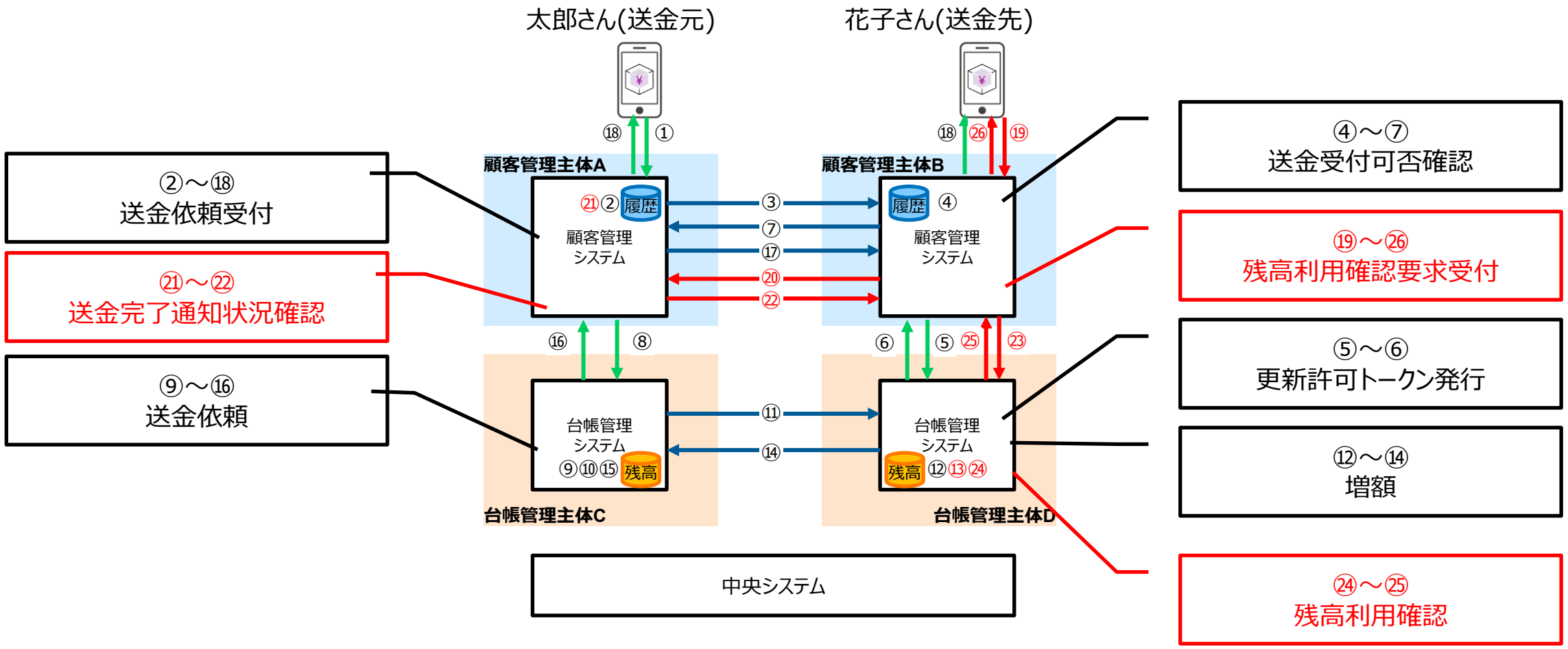
年間取引件数（Y＝**3,290億件**）÷ 1秒あたり（**3,153.6万秒**）＝ 秒間件数（Z＝**10,433件**）

NTT DATA

4.3. 2段階処理の副作用（2/3）

Beyond Finance®
みらいの社会のつくり手に

処理シーケンスから各システムの処理単位を抽出。２段階処理の追加により、各システムに１処理追加されます



NTT DATA

4.3. 2段階処理の副作用（3/3）

Beyond Finance®
みらいの社会のつくり手に

下記2点の副作用などが想定されるため、仲介機関ごとの個別最適化を超えて、NWシステムのようなハブを設けることが、CBDCシステム全体の安定性や運用効率の確保に寄与すると考えられる

1

通信・再送・例外処理・接続維持等の周辺的な負荷の増加

２段階処理で、さらにシステム間に跨る通信が増加するため、通信・再送・例外処理・接続維持等の監視など、仲介機関が担う、周辺的な負荷は増すと考えられる

2

各システムのトラフィック量の増加

全体負荷を１万tpsと仮定した場合において、各システムの負荷は、1万tpsをベースとして、仲介機関の分散度合いに応じて、追加となり、トラフィック量も相応となり、安定性や運用効率を確保するための難易度は相対的に高くなると考えられる

NTT DATA

4.4. アカウント視点とトランザクション視点の分離設計のイメージ

Beyond Finance®
みらいの社会のつくり手に

取引履歴明細に種別を設けることで、下図のイメージでビューを分けるなど、UI/UXの設計に自由度を出せると想定

まとめ表示モード（アカウント視点）

最終状態にまつわる取引明細のみ表示されるモード

（送金と反対取引はセットで**相殺されて非表示**となる）

全明細表示モード（トラン視点）

すべての取引履歴が表示されるモード

（送金と反対取引はセットで表示される）

一意となるキー(例)

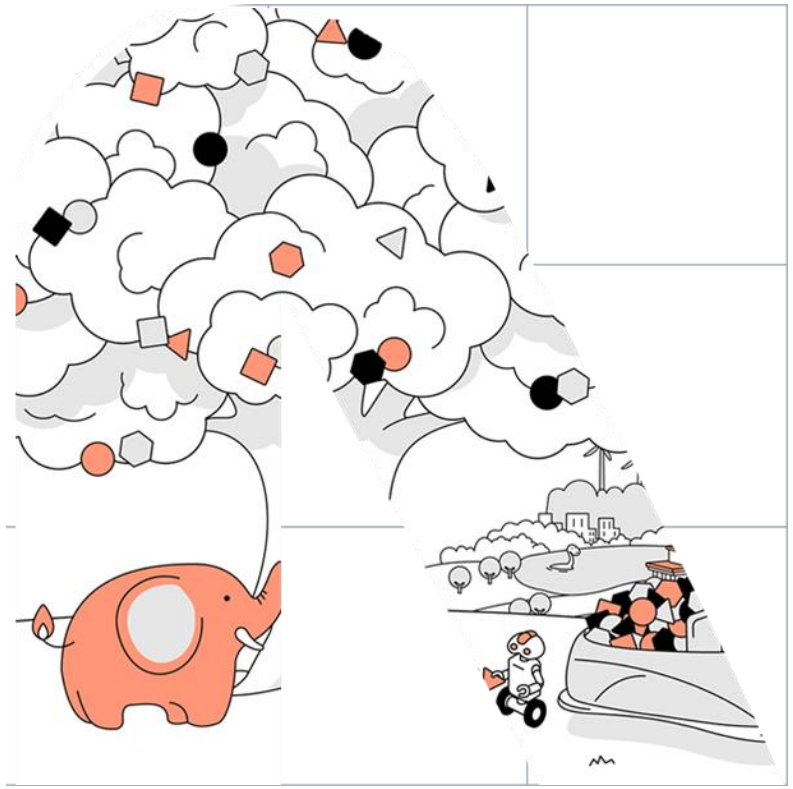
取引ID	取引種別	送金元	送金先	取引金額
Tx-1	送金	太郎	花子	−100
Tx-1	制御(反対取引)	花子	太郎	+100

顧客管理システムA

取引履歴

NTT DATA

05



ディスカッションテーマ

5.1. ディスカッションのテーマについて

Beyond Finance®
みらいの社会のつくり手に

下記3つのテーマにて、グループディスカッション→内容共有の形式でディスカッションさせてください

1 反対取引開始～終了までに残高が利用されるリスクへの対処

２段階処理を行うアプローチ案に言及いたしましたが、システム面・運用面いずれでも構いませんので、その他のアプローチや、２段階処理の問題点がございましたら、ご意見ください。

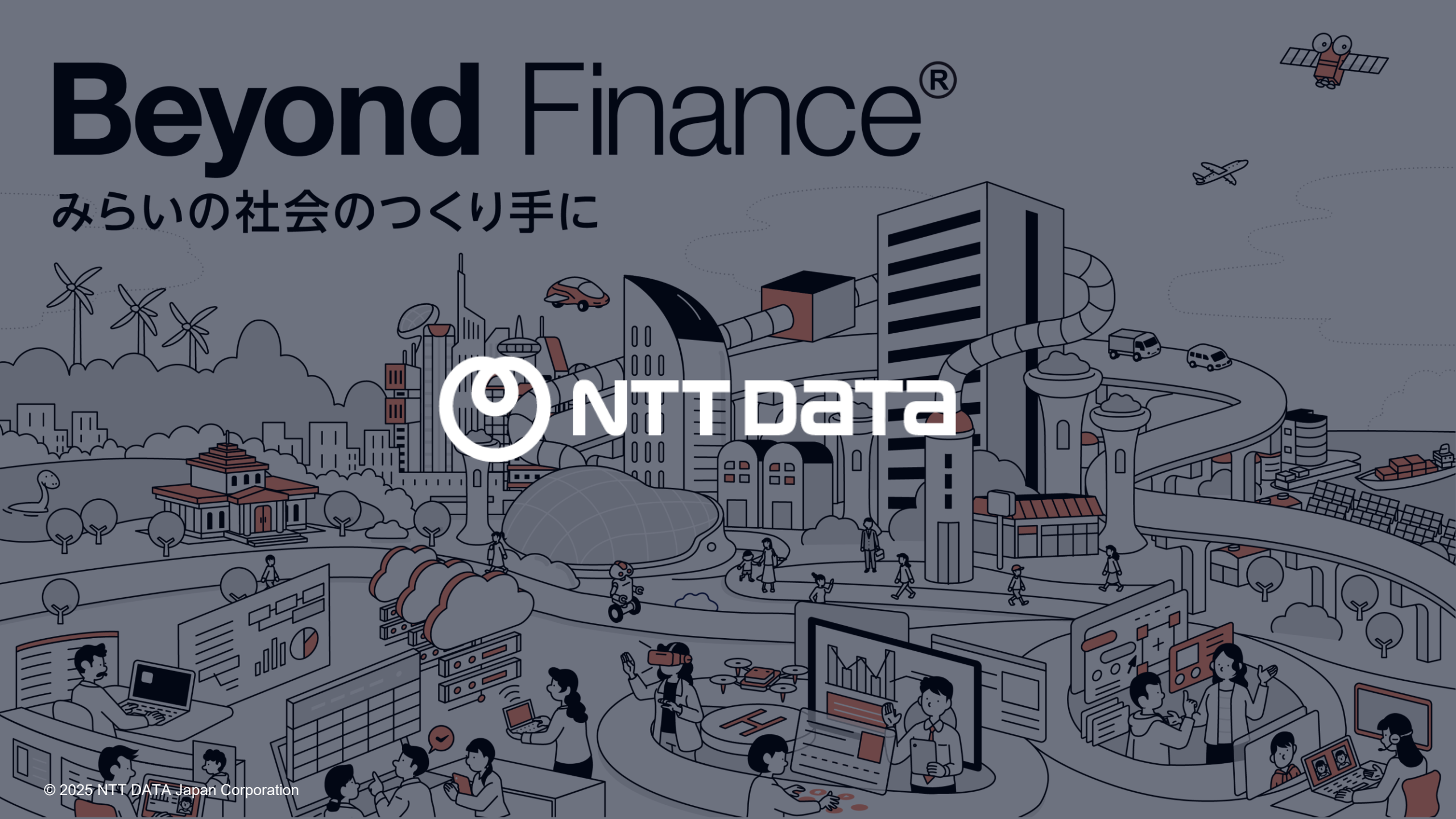
2 反対取引のユーザーフィードバックのアプローチ

「アカウント視点とトランザクション視点の分離設計」を行うアプローチ案に言及いたしましたが、システム面・運用面いずれでも構いませんので、その他のアプローチや、分離設計案の問題点がございましたら、ご意見ください。

NTT DATA

Beyond Finance®

みらいの社会のつくり手に



別添４

ワーキンググループ（WG7）
【基本機能の事務フロー】
第６回会合
日本銀行説明資料

第７回会合に向けた事前説明

2025年９月

日本銀行 決済機構局

CBDC送金にかかる共通の処理フローの別パターン

送金にかかる共通の処理フローの論点

- 実験用システムにおける、送金にかかる共通の処理フローでは、論点があると推察される。

【論点①】

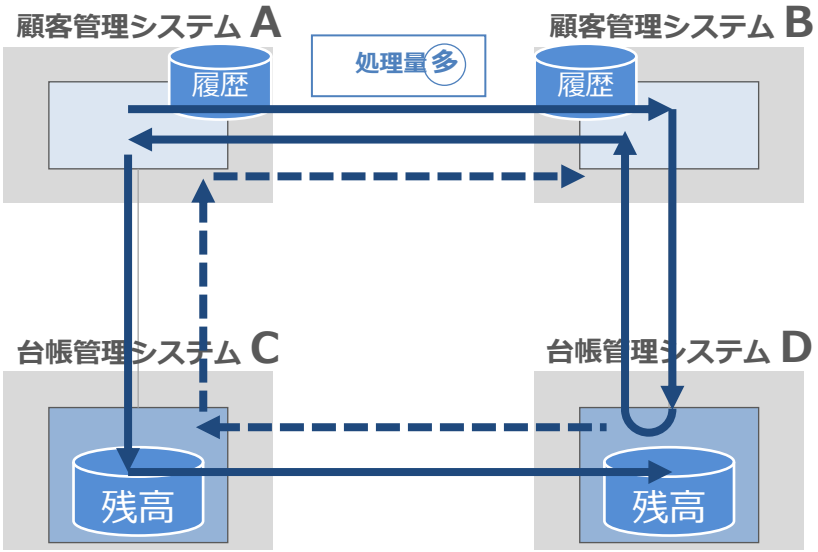
台帳間のエラーハンドリングとして
タイムアウトエラー処理を実装のうえ、
整合性の確保が必要

- 着金側台帳管理システムから応答電文の通信が途絶する等、何らかの理由で通信に失敗した場合は、送金側の台帳管理システムは待ち続けることしかできず処理を進められない。このため、一定時間が経過しても処理が進まない場合に決済を取り消すタイムアウトエラー処理の実装が必要である。他方、仮にタイムアウトエラー処理を行う場合、着金側の台帳管理システムが増額していると、この残高が別の支払に利用されることで送金元台帳との間で不整合が起きうる。

【論点②】

顧客管理システムの処理量が多い

- 実験用システムの性能試験の結果を踏まえると、顧客管理システムは履歴の更新や他の顧客管理システム含む他システムへのメッセージングが主たる処理となっているなか、台帳管理システムよりも大きなリソースが必要になると推察される。



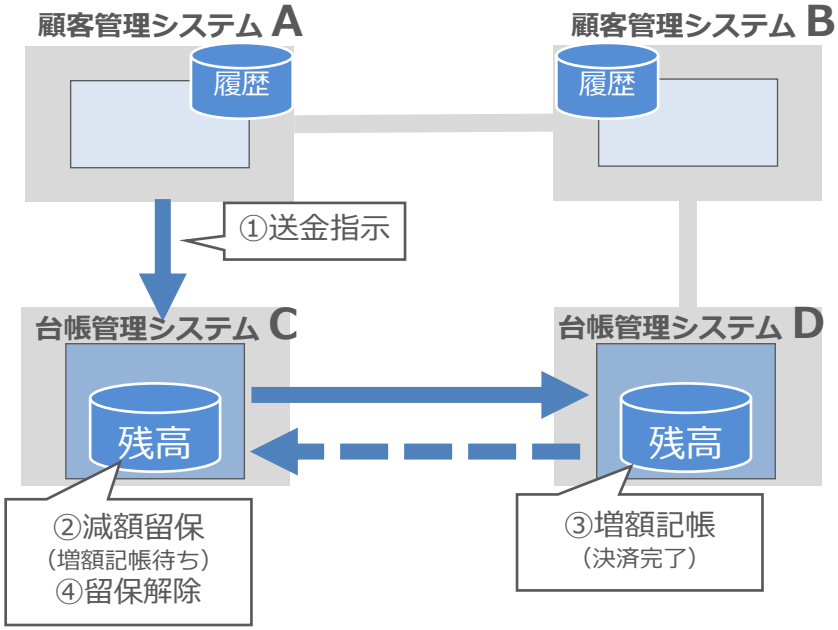
現時点でCBDCを発行する計画はなく、CBDCに関して前提や仮定を置く場合も、実験用システムやWGの議論のための仮置きに過ぎない

共通の処理フローの論点①：タイムアウトエラーと整合性確保

- タイムアウト管理と、台帳間の整合性確保を実現するアイデアとして減額留保と同様の考え方で増額留保を検討。

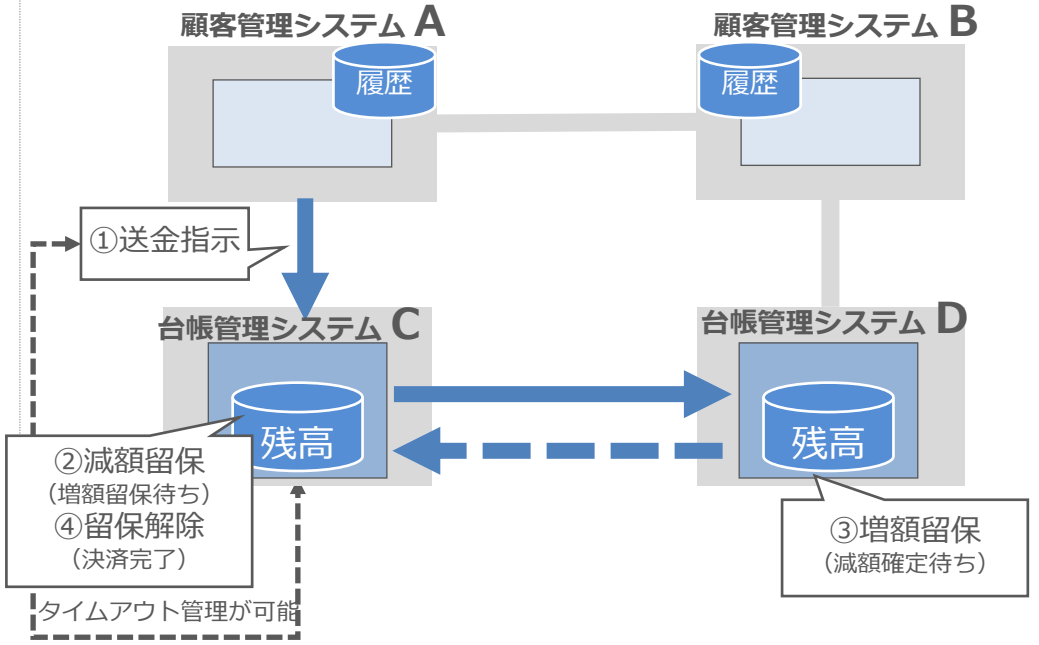
実験用システムの台帳部分のフロー

- 増額記帳をもって決済完了。送金元は増額確定の通知を受領次第、留保を解除。



増額留保を行うフロー

- 減額確定をもって決済完了。着金側は減額確定の通知を受領次第、留保を解除。
- 送金指示受領から減額確定（下図①～④）までの時間について台帳管理システムCがタイムアウト管理を可能

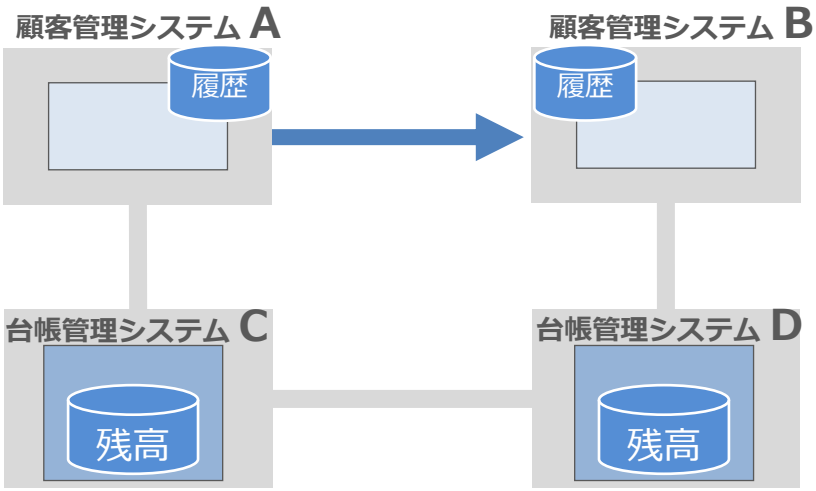


現時点でCBDCを発行する計画はなく、CBDCに関して前提や仮定を置く場合も、実験用システムやWGの議論のための仮置きに過ぎない

共通の処理フローの論点②：顧客管理システムの処理量

- 顧客管理システムの処理負担を軽減するために、顧客管理レイヤーで取引是非の判断を先に行う実験用システムのフローと、送金側の残高を先に確保するフローを検討。

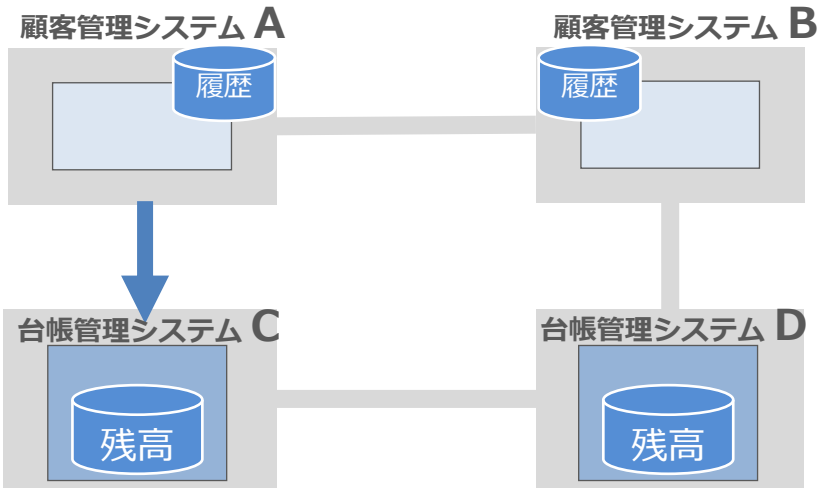
実験用システム
（送金にかかる共通の処理フロー）



先に行う処理

顧客管理システムによる取引の是非の判断

検討案



先に行う処理

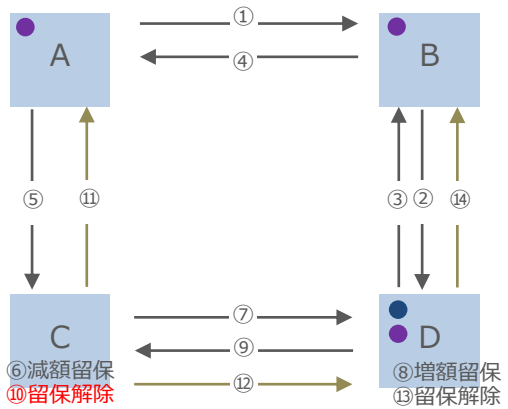
送金側の残高の確保

現時点でCBDCを発行する計画はなく、CBDCに関して前提や仮定を置く場合も、実験用システムやWGの議論のための仮置きに過ぎない

送金にかかる共通の処理フローの別パターン

- 別パターンとして、実験用システムを基本としたフロー（案①・案②）と顧客管理システムの処理量を軽減するフロー（案③）を検討。なお、増額留保はすべての案に具備する前提で検討。
  - 案①・案②については、先に顧客管理における取引の是非の確認を終え、最後に決済する点が特徴。一方、案③では、まず送金分の残高を確保（減額留保）する点が特徴。

案①＜実験用システムのフロー＞



DB更新：● 取引制限判定、保有上限判定
● トークン登録

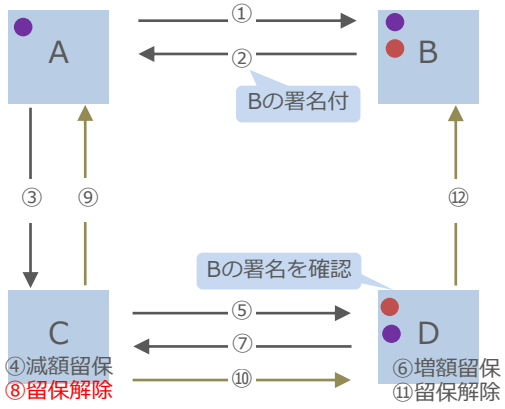
赤字で決済完了につき、以降はロールバックをしない

実験用システムのフローに増額留保の処理を加える

Pros：
送金側によるタイムアウト管理

Cons：
実験用システムよりも処理量が増える可能性

案②＜案①の通信経路変更＞



DB更新：● 取引制限判定、保有上限判定
● 署名・署名検証

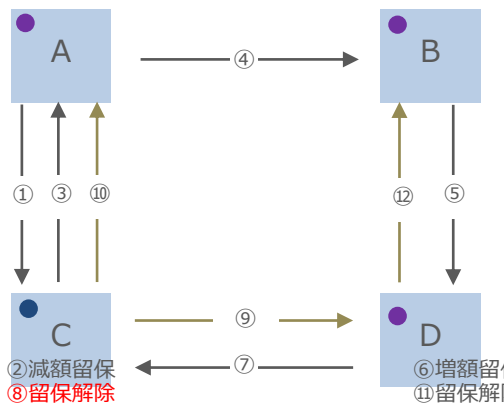
赤字で決済完了につき、以降はロールバックをしない

電子署名を活用することで許可発行にかかる処理・通信を省略

Pros：
顧客管理・台帳管理間の通信回数

Cons：
電子署名にかかる処理負荷

案③＜送金分の残高を先に確保＞



DB更新：● 取引制限判定、保有上限判定
● トークン登録

赤字で決済完了につき、以降はロールバックをしない

送金側の残高を先に確保し、顧客管理システム間の通信を削減

Pros：
顧客管理間、台帳管理間の通信回数

Cons：
減額留保以降のフローが長く
タイムアウト時の処理が複雑

現時点でCBDCを発行する計画はなく、CBDCに関して前提や仮定を置く場合も、実験用システムやWGの議論のための仮置きに過ぎない