2025 年８月４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５回会合の議事概要

１．開催要領

（日時）2025 年５月 16 日（金）14 時 00 分～16 時 30 分
（形式）対面形式及びＷｅｂ会議形式
（参加者）別紙のとおり

２．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５回会合日本銀行説明資料」（以下、事務局説明資料）に基づいて説明を実施。事務局説明資料は別添１を参照。

３．質疑応答およびディスカッション

事務局から説明後、質疑応答および参加者によるディスカッションを行った。概要は以下のとおり。

【ＣＢＤＣ口座の保有数】

（参加者）制度上、１人のユーザがＣＢＤＣ口座（以下、口座）を複数保有できるか否かは決まっていないと思うが、このＷＧでの議論においては１人のユーザが保有できる口座は１口座のみという前提で合っているか。

（日本銀行）ご認識のとおり。ここでの議論においては、送金側ユーザである太郎は１口座のみ保有し、これは台帳管理システムＣに記録され、顧客管理システムＡを通じてアクセスできる。着金側ユーザである花子も同様に１口座のみ保有し、これは台帳管理システムＤに記録され、顧客管理システムＢを通じてアクセスできるという前提である。ここでは、顧客管理システムが複数の台帳管理システムと接続されることは想定せずに議論いただきたい。

【ＣＢＤＣの送金にかかる共通の処理フロー[1]と対案】

（参加者）ステップ３では台帳管理システムＣの更新時に台帳管理システムＤが更新を許可したことを確認でき、ステップ３に対する対案３（別添１、８頁）では確認できないのはなぜか。

（日本銀行）実験用システムにおいて、取引を行う際には指示を受け付けた送金側の顧客管理システムがすべての取引に取引ＩＤを付番する。このため、対案３のように、台帳管理システムＤが取引ＩＤを保持しても、台帳管理システムＣにとっては、当該取引ＩＤを確認するだけでは、台帳管理システムＤが更新を許可したか否かを判別できない。一方で、ステップ３では、台帳管理システムＤが更新を許可する際には更新許可トークン（以下、トークン）が生成され電文に付与されるため、台帳管理システムＣにとっては、トークンの有無で台帳管理システムＤが更新を許可したか否か判別できる。ただし、トークンは台帳管理システムＤが生成したものであり台帳管理システムＣがそれを正しく検証しようとすると工夫が必要であるほか、実験用システムにおいては、台帳管理システムＤが自ら更新許可の判断をするところまでの実装は行っておらず、実装の簡単化のため顧客管理システムＢの指示に基づいてトークンを生成しているのみである。なお、今後の検討において、仮に台帳管理システムＤが自ら更新許可の判断をすることが必要となった場合には、トークンを生成する当該機能を用いて対応できると考えている。

（参加者）前提イ、ロ、ハ（別添１、２頁）を満たす処理フローとして、説明いただいた処理フローのほかに、まず顧客管理レイヤーで取引可否判定を行い、次に各顧客管理システムが並行的に各台帳管理システムに送金指示を行い、最後にそれぞれの送金指示を基に台帳管理レイヤーで増額と減額の指示を行うというフローも考えられる。しかし、このフローでは完了通知を行う方法が課題になるだろう。

（日本銀行）そのようなフローも考えられるが、顧客管理レイヤーでの合意後にパラレルに処理が行われると処理の複雑性が増すと考えたため、実験用システムでは、実装容易性の観点から、電文が途中で分岐しないような一

[1] 送金処理のために複数の主体が関与することを想定し、実験用システムにて構築した処理フロー（ステップ３：別添１、５頁）であり、順送金や逆引送金等のＣＢＤＣの送金を実行する際に行われる共通の処理を指す。

筆書きの処理を意識した。

【着金側における顧客管理システムから台帳管理システムへの指示】

（参加者）前提イについて伺いたい。顧客管理システムＡから台帳管理システムＣへの指示が必要である点は理解できたが、顧客管理システムＢから台帳管理システムＤへの指示は必要なのか。ＡＭＬ／ＣＦＴやセキュリティ等の観点も含めた取引可否判定を顧客管理レイヤーで行う前提を置いたとして、取引可と判定された取引のみが台帳管理システムＣに連携されるのであれば、顧客管理システムＢから台帳管理システムＤへの指示は不要ではないか。

（日本銀行）本ＷＧの議論においては、顧客管理システムや台帳管理システムはそれぞれ異なる主体である。異なる主体間で処理を行うため、台帳管理システムＤは台帳管理システムＣから受ける指示について実行して良い処理か否か判断する必要がある。この判断を行うために、台帳管理システムＤが自ら生成したトークンを電文に付与することを考えた。こうして、実験用システムでは、顧客管理システムＢが取引可否判定を行ったあとの処理として、顧客管理システムＢから台帳管理システムＤに直接指示を出し、台帳管理システムＤがトークンを生成し、電文に付与する処理フローとした。

（参加者）各システムがそれぞれ信頼できるか否かによって、顧客管理システムＢから台帳管理システムＤへの指示の必要性は変わり得るだろう。また、トークンを発行する場合、トークンの期限管理等、運用面も煩雑になる可能性が考えられる。

（日本銀行）各顧客管理システムや各台帳管理システムを全面的に信頼することができれば、対案４（別添１、９頁）のとおりトークン等を用いない案はあり得る。他方、顧客管理システムＡと顧客管理システムＢは（今次検討の想定として）それぞれ異なる主体であることを踏まえれば、組織間・システム間の信頼性をどのように確保するかが論点となるほか、問題発生時における責任分界が課題となるだろう。トークン発行の際の運用管理に留意点があるというのは、ご指摘のとおり。

（参加者）顧客管理システムＢにおける取引可否判定では、例えば一定期間における着金件数・金額の累計、取引ごとの金額等を確認し、判定するもの

と理解した。しかし、取引可否判定は問題がなく、顧客管理システムＢから台帳管理システムＤにトークンの発行依頼を行ったとしても、非同期処理であることからトークン発行時点とその後の送金実行の時点で着金側ユーザ口座の残高が変化していることも考えられる。そうすると、実際に台帳管理システムＤが残高を増額するタイミングにおいて着金側ユーザ口座の保有上限により着金が拒否される可能性があるのではないか。

（日本銀行）ご指摘のとおり、着金側に保有上限の条件がある場合には、台帳管理システムＤの増額処理の時点で拒否される可能性はある。そのような着金時に台帳管理システムＤにて保有上限判定等によってエラーになる可能性を考慮し、エラーになっても送金側の残高減額を元に戻せる仕組みを実装した。なお、（実験用システムには実装していないものの）トークン発行時に入金額分の枠の確保等を行えば、このような課題が解消されるかもしれない。

【即時性を持つ分散システムにおけるエラーハンドリング】

（参加者）各システムを管理する仲介機関の事務負担を考慮すれば、極力エラーが少ないフローを構成することが望ましいが、実験用システムのように、システムを分散させる考え方ではエラーハンドリングの難易度が高くなるだろう。例えば、台帳管理システムＤを増額しファイナリティを付与した時点以降に、台帳管理システムＤから台帳管理システムＣへの電文が何らかの理由で到着しない、または途絶した場合の対応方法は悩ましい。取引をどのように追跡して不整合を認識し、どのようにして正しい状態にするのかを考えた場合、取引成立照会のような機能を設けての対応等が考えられるが、簡単ではないだろう。

（日本銀行）ご認識のとおり、台帳管理システムＤにおいてファイナリティを付与した後に、台帳管理システムＤから台帳管理システムＣに決済の完了を伝達する中で電文が途絶した場合や、台帳管理システムＤに障害が起きた場合等、ファイナリティ付与の前後でエラーになった場合のエラー検知方法やリカバリー方法については、重要かつ難しい課題と認識している。

（参加者）各システムが分散していることでシステム障害時等の対応が難しくなるのだろう。例えば、顧客管理システムＡから顧客管理システムＢに指示するタイミングや、顧客管理システムＢから台帳管理システムＤに指示するタイミング等、各システム間でデータの通信が発生する度に履歴が残

るようにすることで、障害発生時等においてどのタイミングまでデータが正しかったかが分かるのかもしれない。また、台帳は分散するよりも集中管理されている方が不整合のリスクが低減するだろう。

（日本銀行）台帳が複数に分かれて存在している前提を置く場合は、障害発生時における情報の収集方法や整合性のとり方について、検討が必要だと考えている。

（参加者）分散的なデータベースで即時性を高めることは大変だと思う。例えば、電文が途絶した際に、電文の取消や組戻し等の対応が必要となるだろう。

（日本銀行）ＣＢＤＣの送金にかかる共通の処理フローでは、顧客管理システムと台帳管理システムの２つのレイヤーが処理を行うフローにしているが、仮にＣＢＤＣが社会実装されたとすると、スマートフォンアプリや店舗端末等、フロント部分のレイヤーがさらに複雑になると想定される。市中における既存の決済サービスの仕組みは、即時に資金決済が行われないケースが多いため取引キャンセルを含めた異例対応等も対応し易いが、ＣＢＤＣにおける即時決済性を考慮すると、台帳管理レイヤーで資金決済を即時に完了させるため、顧客管理レイヤーやさらにフロント部分のレイヤーとの整合性の確保の難易度が高いと感じている。なお、実験用システムと、市中における既存の決済サービスの仕組みの異なる部分は、台帳管理システム、顧客管理システムがそれぞれ資金決済における役割を担いつつ、システムとして分散している点である。例えば、台帳が１つのデータベースで処理される場合は減額と増額を１トランザクションで同時に実行できるが、台帳管理システムとそのデータベースが分かれている場合は片方だけ成功するというケースが論理的には考えられるため、エラーハンドリングの問題が複雑性を増す。台帳管理システムＤで資金決済が完了したあとに、通信障害等により台帳管理システムＣに完了通知が届かなかった場合等、ユーザまで含めたエラーハンドリングの対応については別途議論の余地があると考える。

（参加者）ＣＢＤＣを現金の代替と捉えると、資金決済の即時性は必要だと考える。資金決済が即時で完了するということは、エラー時は即時でエラーが分かるということである。エラーの発生により資金決済が完了しないトランザクションが発生した場合、処理がエラーで終了していることをユー

ザ含めて検知できることが肝要である。

（日本銀行）「即時性」についての解像度を上げる必要があるのかもしれない。プライバシーに配慮しながら、複数の管理主体に跨るシステム間で、処理を即時で行うフローには相応の複雑さがある。実験用システムでは実装容易性の観点と処理のシンプル化を企図し、電文が途中で分岐しない一筆書きのフローを考えた。これを実現する際に、厳格に即時である必要性がどこまであるのか、改めて検討が必要かもしれない。例えば、親が子に小遣いを与えるという場面においては、子に着金した瞬間に使えるようになるという意味で即時の資金決済は必要だろう。他方、小売店舗でユーザが購買する場面においては、ユーザが購買を完了することが即時に実施されていればよく、ユーザが支払った瞬間に店舗がＣＢＤＣを使えるようにする必要性は高くないのかもしれない。

（参加者）小売店舗でユーザが購買する場面における即時性の利点として、悪意者が即時決済されないことを狙って不正取引を行おうとする可能性を鑑みれば、即時決済はこれを防止するための有効な手段になるのではないか。そのほか、中小企業をはじめとして、資金効率性を高める点も挙げられるのではないか。

４．次回予定

次回の会合は 2025 年９月３日（水）に開催予定。

以　上

別紙

ＣＢＤＣフォーラム ＷＧ７
「基本機能の事務フロー」
第５回会合参加者

（参加者）※五十音・アルファベット順
株式会社ことら
株式会社静岡銀行
一般社団法人しんきん共同センター
株式会社セブン銀行
一般社団法人全国銀行資金決済ネットワーク
株式会社千葉銀行
日本電気株式会社
日本アイ・ビー・エム株式会社
日立チャネルソリューションズ株式会社
株式会社ふくおかフィナンシャルグループ
株式会社三井住友銀行
株式会社三菱 UFJ 銀行
株式会社ゆうちょ銀行
株式会社りそなホールディングス
株式会社ローソン銀行
BIPROGY 株式会社
株式会社 NTT データ

（事務局）
日本銀行

別添１

ワーキンググループ（WG7）
【基本機能の事務フロー】
第５回会合
日本銀行説明資料

送金にかかる共通の処理フローの背景について

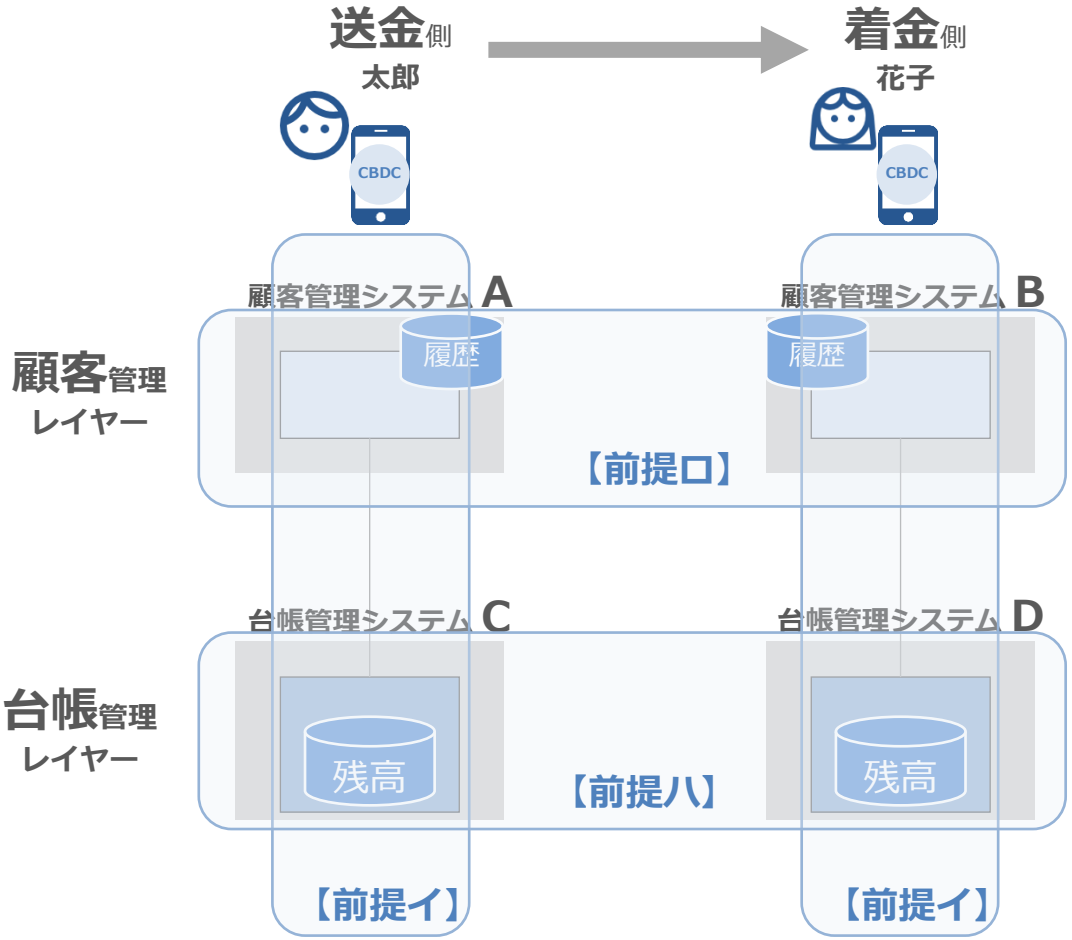
2025年５月

日本銀行 決済機構局

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】

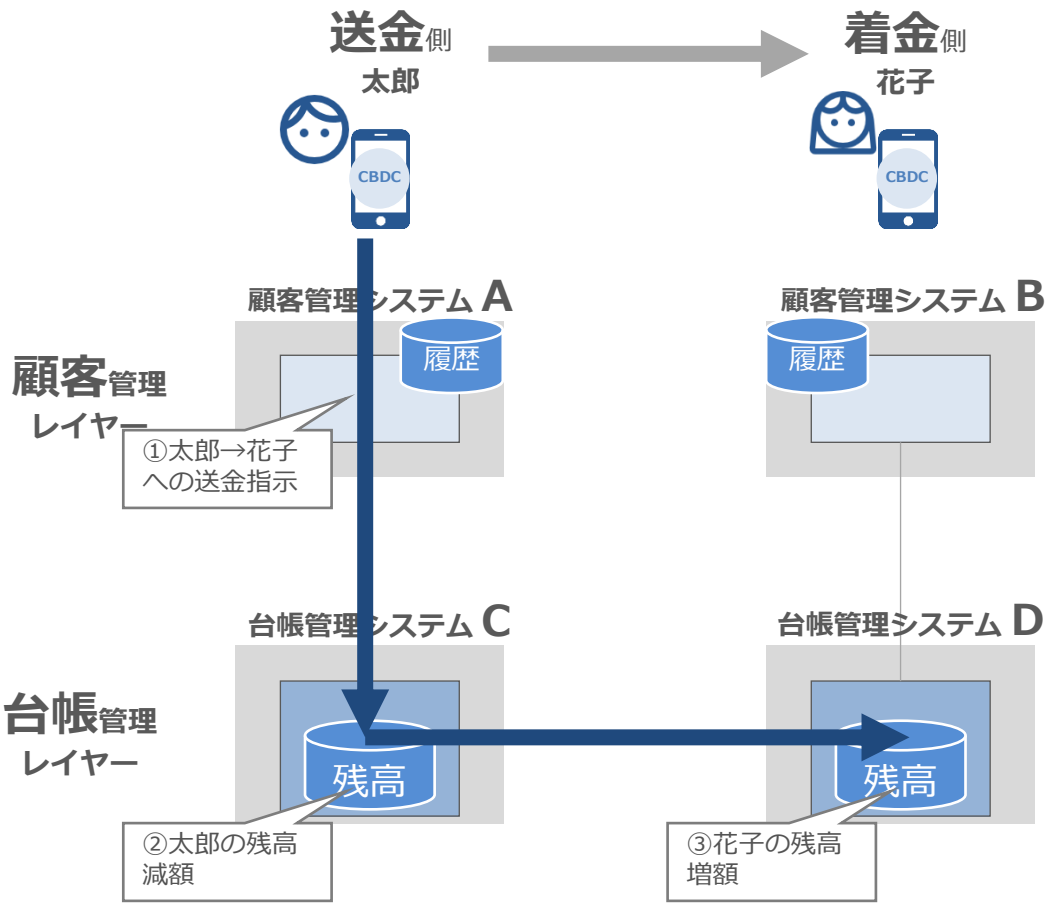
- **【前提イ】**：台帳を更新する際には、**顧客管理システムからの指示**が必要
  - 顧客管理システムの取引可否の判断のうえ台帳更新する
- **【前提ロ】**：台帳を更新する前に、**顧客管理レイヤーにて取引の是非の判断**（各種取引制限判定、AML/CFT等）を実施する
  - 顧客管理レイヤーでの取引の判断の後、台帳管理レイヤーでの処理を行う
- **【前提ハ】**：複数の台帳を更新する際には、**台帳間において整合性を取る必要**
  - 「台帳間の整合性」とは、Cの減額とDの増額がセットになるようにすること（例：Cの減額だけされているという状態を極力短くする）
  - 今回は、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

ステップ１：シンプルなフロー

- まず、シンプルに考えると、「台帳管理システムCの減額と台帳管理システムDの増額」を行うフローとして、以下がありうる



ステップ１の課題

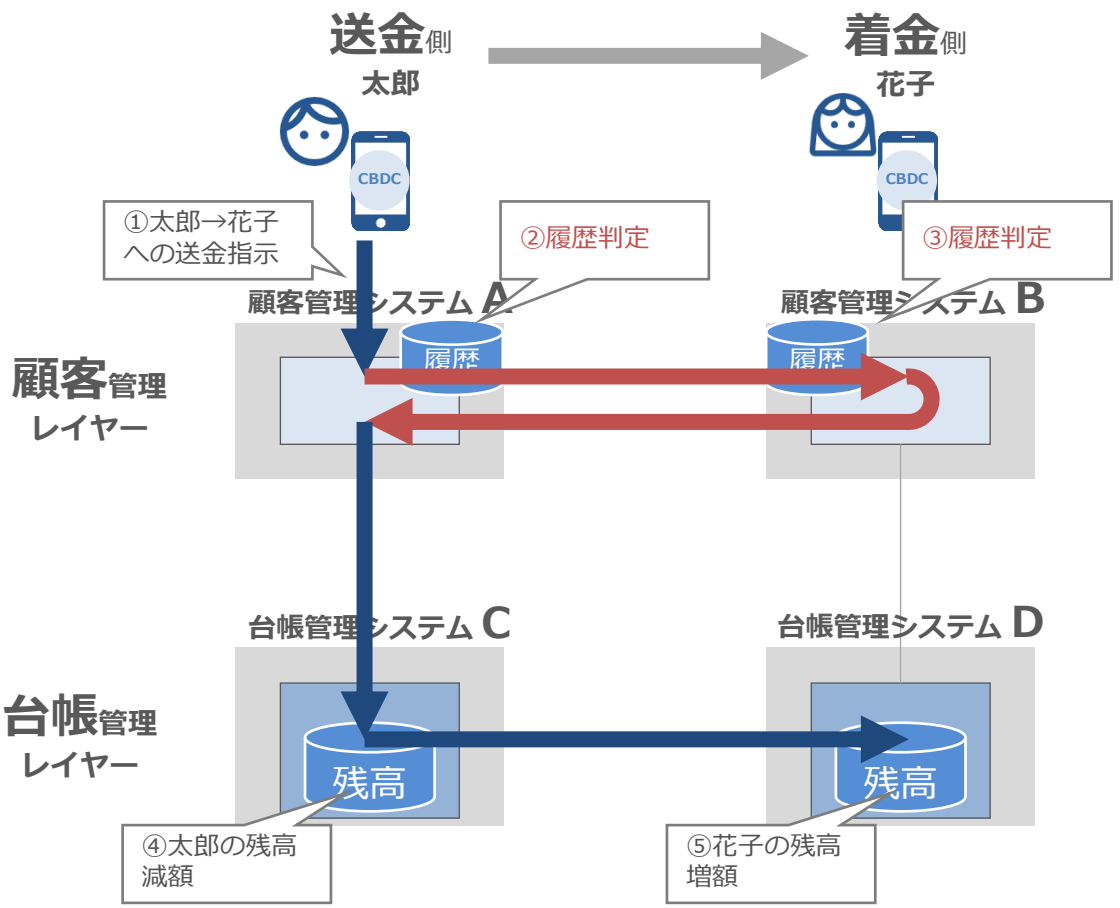
- 履歴判定（累計取引額制限、回数制限）を経ずに台帳更新を行うことになる
  → **【前提ロ】に抵触**
  - ✓ 台帳更新はコストがかかる処理であるため、各種判定を経てから台帳更新を行いたい

ステップ２にて解決を試みる

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

ステップ２：履歴判定を予め行うフロー

- ステップ１の課題を解決するため、顧客管理システムA、Bにおいて、履歴判定を行ったうえで、台帳の更新を行うフローを考える
  - 赤色部分がステップ１との差分



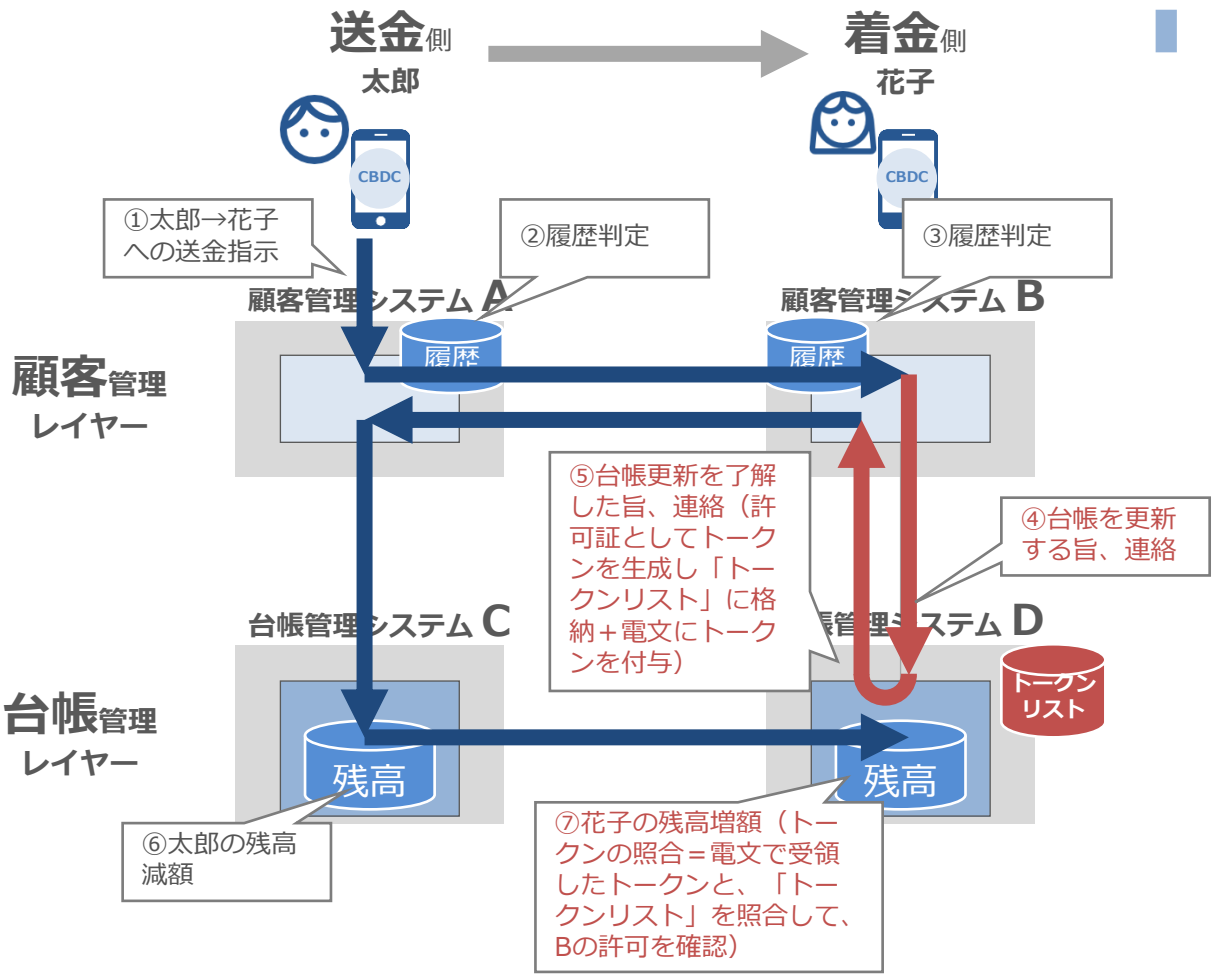
ステップ２の課題

- 顧客管理システムBにて着金可否を判定したか分からない状態で、台帳管理システムDは更新を迫られる
  → **【前提イ】に抵触**
  - ✓ 顧客管理システムからの指示の有無を各台帳管理システムが認識する必要

ステップ３にて解決を試みる

ステップ３：履歴判定を予め行う＋顧客管理での許可を行うフロー

- ステップ２の課題を解決するため、 許可トークンを電文に付与する送金フローを考える（現在実験用システムで採用しているフロー）
  - 赤色部分がステップ２との差分



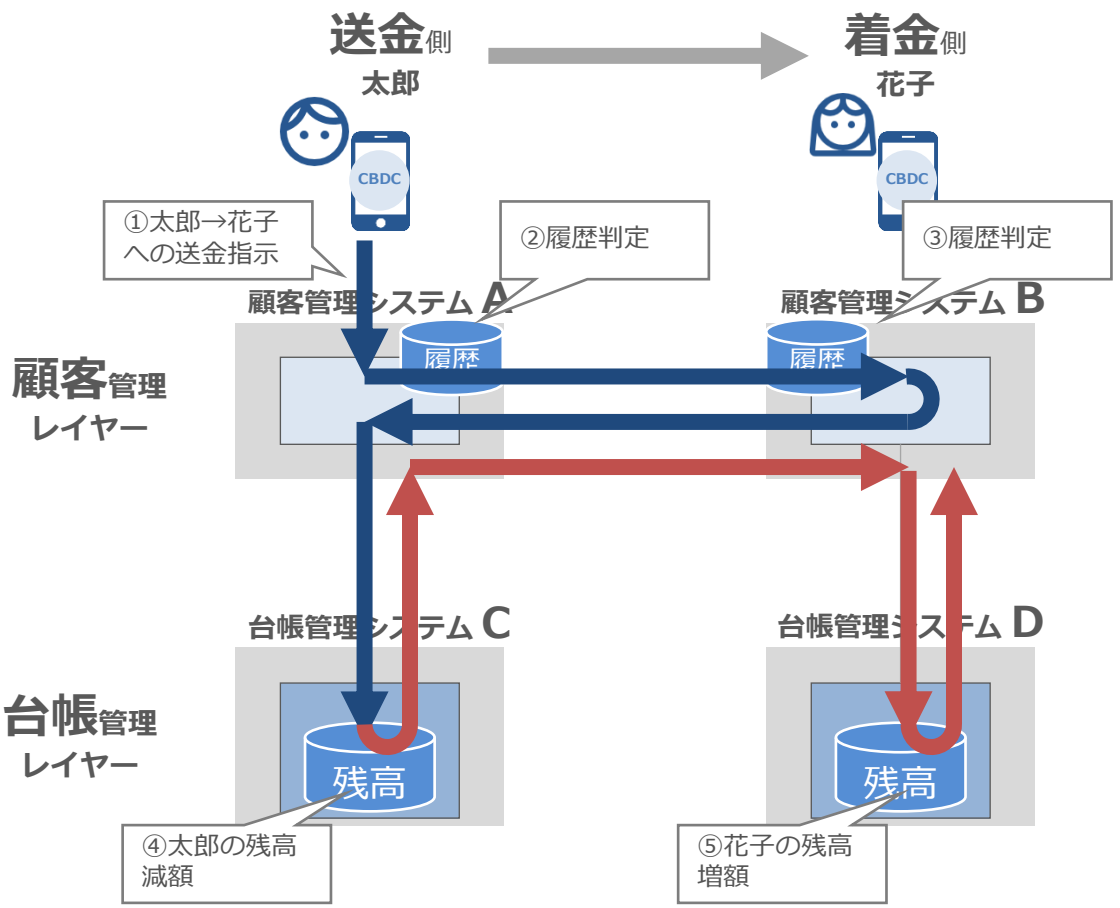
ステップ３の留意点

- フローの複雑性に起因する、性能面の影響、障害時のリカバリー処理の難易度などが潜在的な課題
  →【前提イ】【前提ロ】を解決するためには甘受せざるを得ない
  - ✓ 性能面の影響については、実験用システムを用いた性能テストにて確認

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

ステップ３に対する対案１：台帳間の指示を行わないフロー

- 前回フォーラムにて指摘のあった「台帳間の指示（C→D）を行わない」フローを考察
  - 赤色部分がステップ３との差分



メリット

- 台帳管理レイヤー間のネットワークが不要となる

対案１の課題

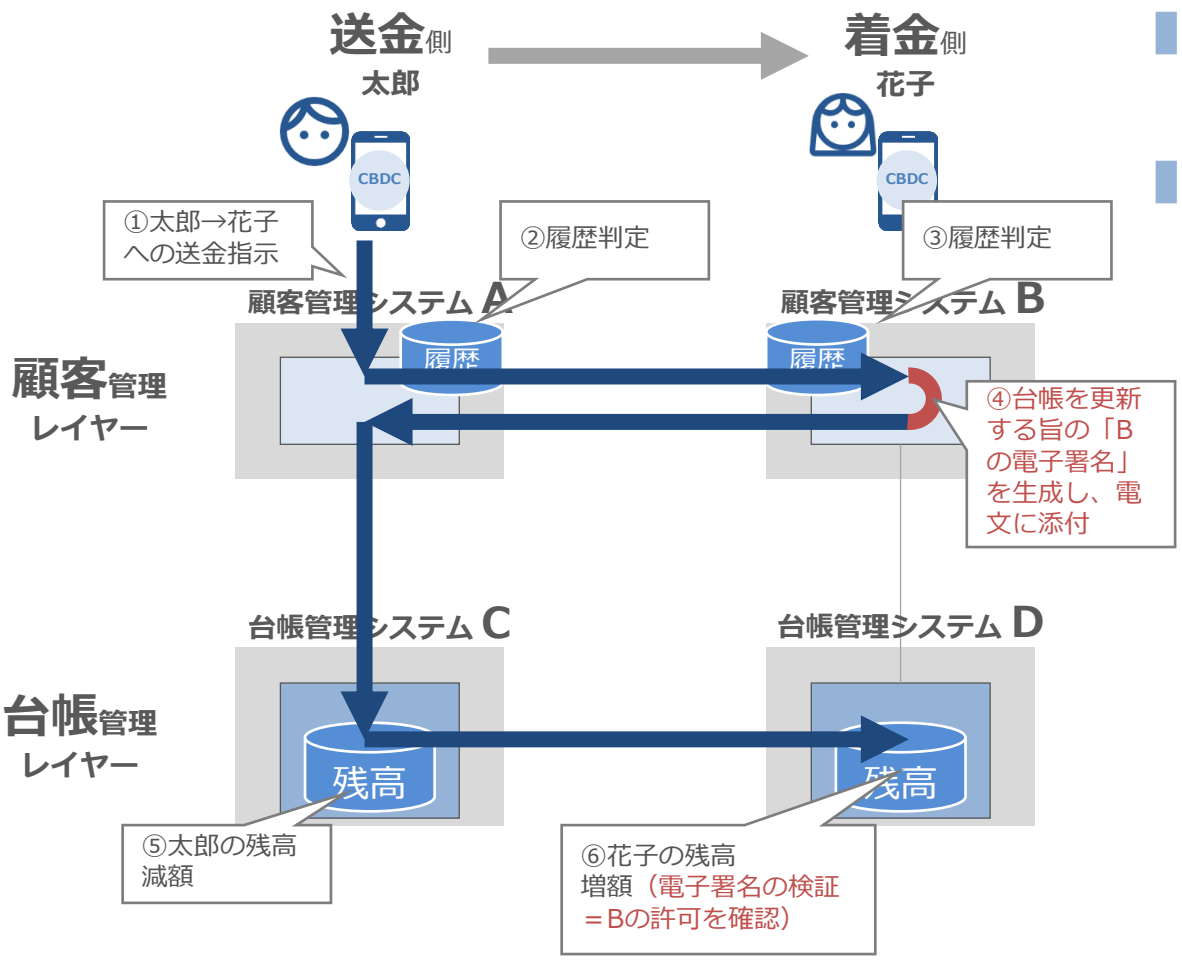
- ABを経由することで関わる主体が多くなり、「Cの減額とDの増額」のタイムラグがステップ３対比で大きくなる
- Dからみて「Cが減額済であること」を確認することが難しくなる
  →**【前提八】に抵触する可能性**

対案１のその他の留意点

- ステップ３では、顧客管理システムでの処理フェーズ（顧客管理レイヤーにて取引の是非の判断）と、台帳管理システムでの処理フェーズ（各台帳の更新：台帳間整合性確保）が分かれており、役割分担とフェーズが明確であったものの、対案１ではそれが曖昧になる（台帳間の更新プロセスに顧客管理が絡んでしまうことになる）
- 対案１では、④⑤の後に、台帳管理システムCにおいて「⑥太郎の留保解除」もあるため、⑥についても顧客管理システムを経由すると相当複雑なフローとなる

ステップ３に対する対案２：Bの許可を電子署名で示すフロー

- 前回フォーラムにて話題に出た「Bが増額の許可を出したということに対して、トークンではなく、Bにて電子署名を付すことでDが確認する」フローを考察
  - 赤色部分がステップ３との差分



メリット

- 通信回数を減らすことができる

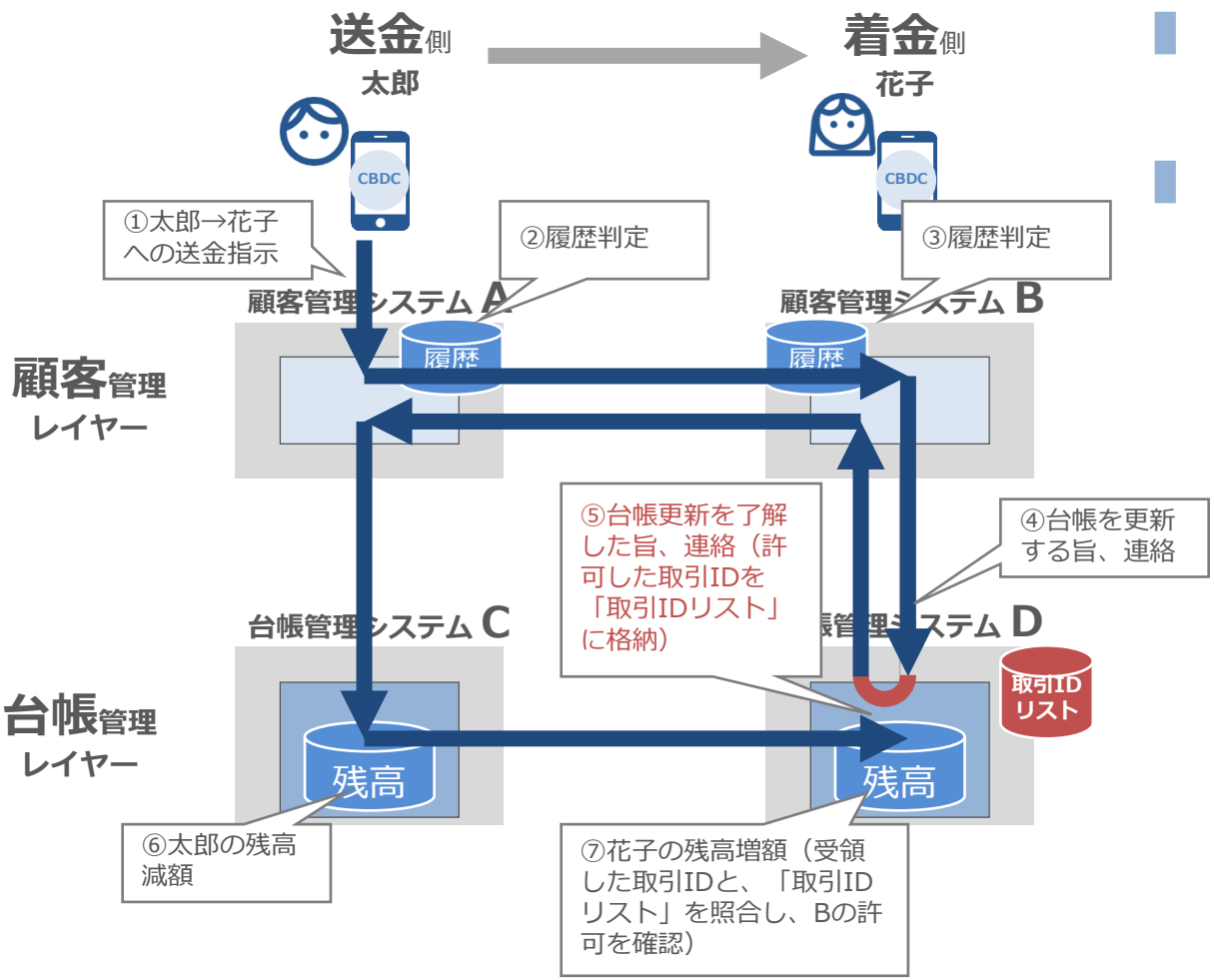
対案２のその他の留意点

- Cの更新時に、Dの更新がreadyであること（拒否していないこと）を確認することが出来ない
- 署名作成・検証に伴う性能面、運用面（公開鍵の配布）で留意が必要

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

ステップ３に対する対案３：Bの許可を取引IDで示すフロー

- 「Bが増額の許可を出したということに対して、取引IDにてDが確認する」フローを考察
  - 赤色部分がステップ３との差分
  - 取引IDとは、すべての取引電文にユニークに付されるID（下図のフローでは１つの取引IDが付される）



メリット

- トークンの生成・検証が不要となる

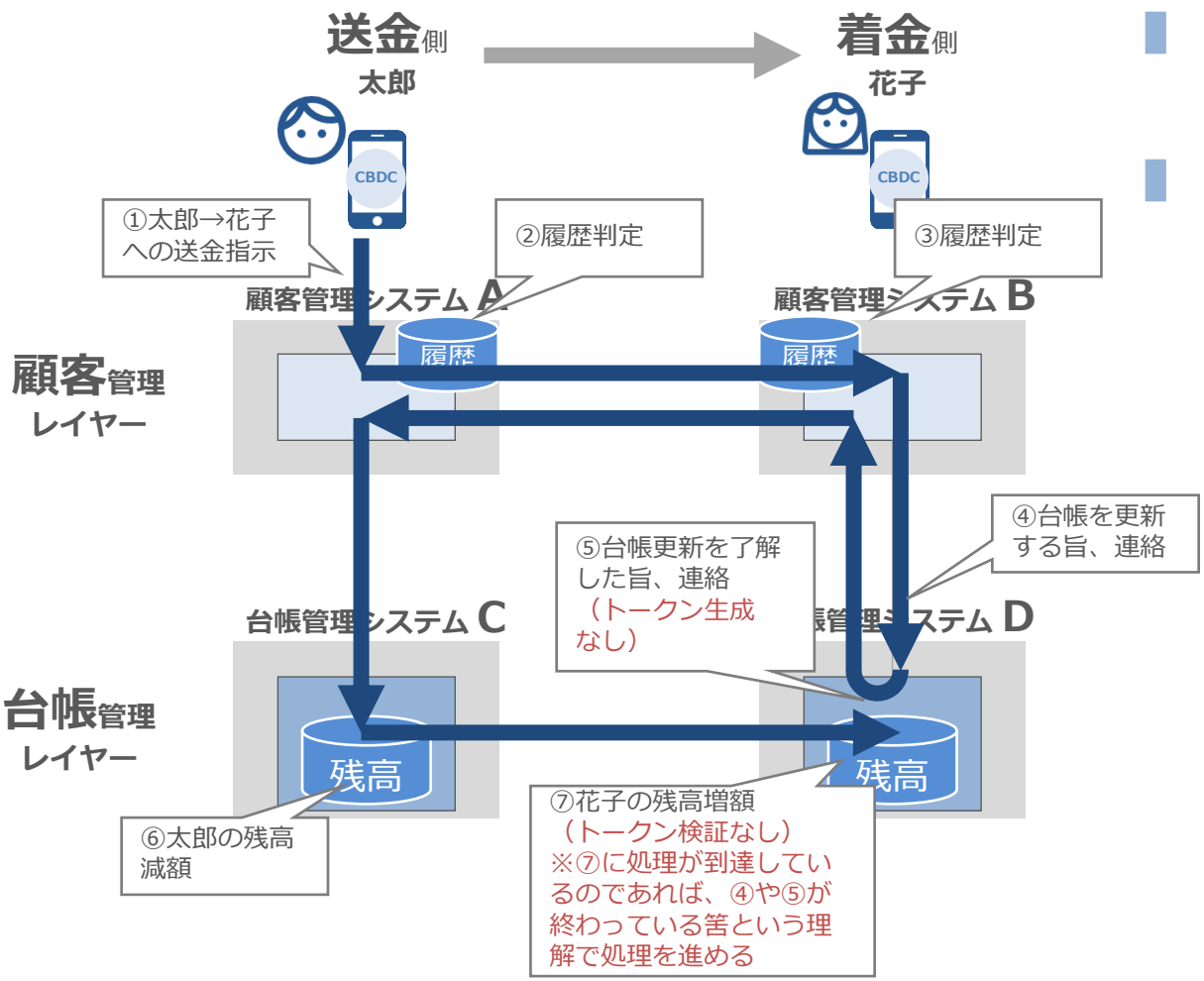
対案３のその他の留意点

- Cの更新時に、Dの更新がreadyであること（拒否していないこと）を確認することが出来ない

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

ステップ３に対する対案４：各主体の処理結果を信頼するフロー

- 「⑦に処理が到達しているのであれば、その前段の④でBの許可や、⑤にてDの了解が終わっているはず」との考えから、トークン生成・検証を不要とするフローを考察
  - 赤色部分がステップ３との差分



メリット

- トークンの生成・検証が不要となる

対案４の課題

- 他の仲介機関の処理結果通知を全面的に信頼して、各主体が処理を行う必要がある
  - ✓ 対案４は「各主体は独立した機関として振る舞うものとして構成している」という前提と反する
  - ✓ 例えば、仮に、Aが①の後に、③④⑤を行わずに、Cへ指示（⑥）を出してしまった場合に、CやDがそれに気づけない

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

以降は参考資料

（第4回会合資料）実験用システムにおける共通の処理フロー①

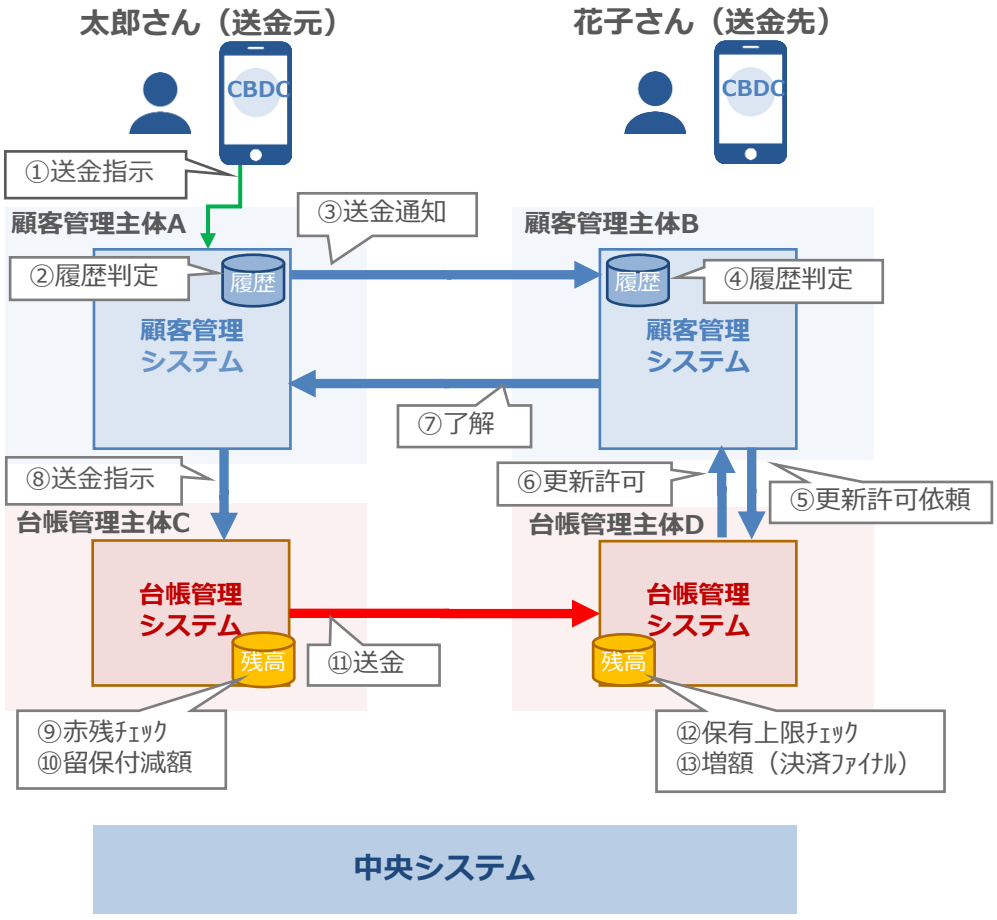
- 共通の処理フローを考える上での前提は以下のとおり。
  - ユーザは当該ユーザの顧客管理を通じて、取引を行う。
  - 中央管理を行う存在は極力排除し、分散トランザクションを意識。

- 上記の考え方に基づき、処理フローを設定。
  - ユーザの送金指示をもとに、送金元ユーザの顧客管理と、送金先ユーザの顧客管理の双方で各種チェックを行い、送金の可否を判定。
  - 送金元ユーザと送金先ユーザの顧客管理が異なる場合、送金先ユーザの台帳を更新するためには、送金先ユーザの顧客管理の了解が必要。
  - あるユーザの台帳の減額・増額を行う際には、当該ユーザを管理する顧客管理からの指示に基づいて、台帳管理が実施。
    - ✓ それぞれの台帳管理は各種チェックを行い決済を実行。（送金元：台帳減額、送金先：台帳増額）。
  - なお、実験用システムでは、パフォーマンスの向上を企図して、顧客管理間や顧客管理－台帳管理間の通信に関し、非同期通信方式を採用。
    - ✓ 加えて、実験用システムでは、個々の顧客管理システムや台帳管理システム内の処理についても、可能な範囲で非同期通信方式を採用。

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

（第4回会合資料）実験用システムにおける共通の処理フロー②

- 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。



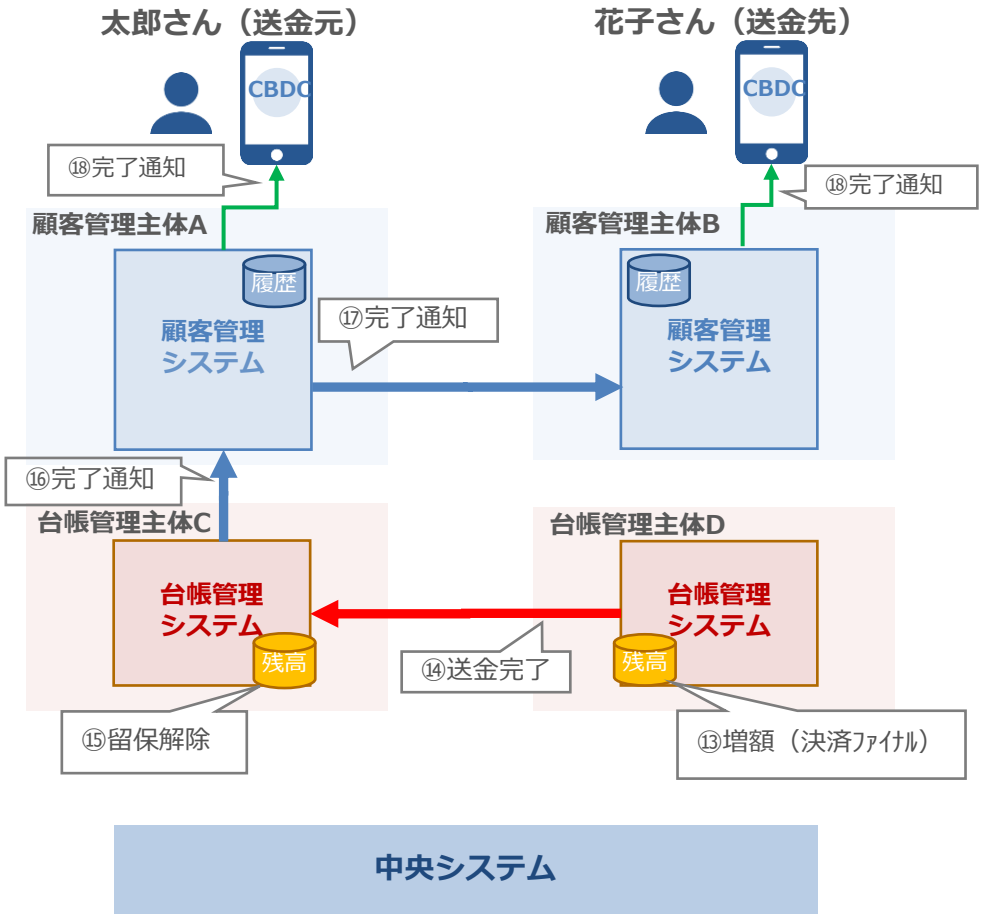
	行為者	処理内容
①	太郎	Aに送金指示
②	A	取引履歴を確認し、太郎が送金可能かを判定
③	A	送金先（花子）を特定し、Bに送金を通知
④	B	取引履歴を確認し、花子が受取可能かを判定
⑤	B	Dに台帳更新許可トークンを発行依頼
⑥	D	Bに更新許可トークンを発行
⑦	B	Aに了解を返送
⑧	A	Cに送金指示
⑨	C	送金にあたって残高が不足していないか太郎の台帳を確認
⑩	C	太郎の台帳を留保付で減額記帳
⑪	C	Dに増額指示
⑫	D	⑥と⑪を突合のうえ、保有上限確認
⑬	D	花子の台帳を増額記帳（決済ファイナル）

※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

（第4回会合資料）実験用システムにおける共通の処理フロー③

- 共通の処理フローについて、「送金先の台帳を増額記帳してから各ユーザに完了通知を行うまで」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