CBDCフォーラム
【追加サービスとCBDCエコシステム】WG（WG2）
第5回会合　事務局説明資料

BISによるProject Rosalindのご紹介

2024年1月18日
日本銀行 決済機構局

1. プロジェクト概要

1-1. プロジェクト概要

- **CBDC（一般利用型）**のAPIに関する実証実験
- **APIプロトタイプ**の構築を通じて、CBDCシステムの機能面、アドプション、イノベーションを促進するために、APIがどのように活用できるか等を調査
- **公的セクターと民間セクターが協働**。これにより、APIのプロトタイプを**潜在的なユーザーニーズ**と合致するように設計・構築することが可能となったとしている
  - BISイノベーションハブ ロンドンセンターとBOEが主導（計画・開発）
  - 金融業、IT事業者、アカデミア、他中銀など、約30のエンティティが参加

Phase 1 (2022年7月 ～2023年1月)	● APIプロトタイプのデザインと構築、機能面の検証 ● ディベロッパー（API users group）や業界のエキスパート（Advisers group）と協働 ● “**Showcasing event**”を開催
Phase 2 (2023年2月～4月)	● エコシステム参加主体との協働を通じたユースケースの調査 ● 2023年3月 ”TechSprint”を開催 – 23チームが参加 – CBDCに関する多様なユースケースのデモンストレーションを実施 ● 同4月 ”**TechSprint demo day event**”を開催 – 選ばれた数チームが、複数の中央銀行から成るグループに、ソリューションをプレゼン

（参考）参加者一覧

API users group （5エンティティ）	TechSprint participants （19エンティティ）
Amazon Bank of Canada Barclays IDEMIA Mastercard	Amazon BMO Boom Budapest University of Technology and Economics eCora Global Cloud Payments IDEMIA Knox Networks Central Bank of Hungary Milicent Labs Nuggets OneID Onestep Financial Revolut SUPER HOW? Secretarium Thales Vayana Network Worldline
Advisers group （6エンティティ）	
R3 Stripe eCurrency OneID Google Visa	

1-2. プロジェクトの目的

- 機能面、相互運用性、アドプション、エコシステムの4観点から実験

①機能面（functionality）に関する目的

✓ どうすればAPIが中銀台帳と民間事業者との連携を可能にできるかを検証
（安全でセキュアな取引を促進するための様々なオプションを含む）

②相互運用性（interoperability）に関する目的

✓ どうすれば異なるシステムやアプリケーション間の相互運用性を確立できるかを検証
（相互運用性を実現する際のデザインやリスク、機会、トレードオフに関する知見の探求を含む）

③アドプション（adoption）に関する目的

✓ 多様性があり革新的なCBDCのユースケースの開発を可能とするためのAPIの機能を検証

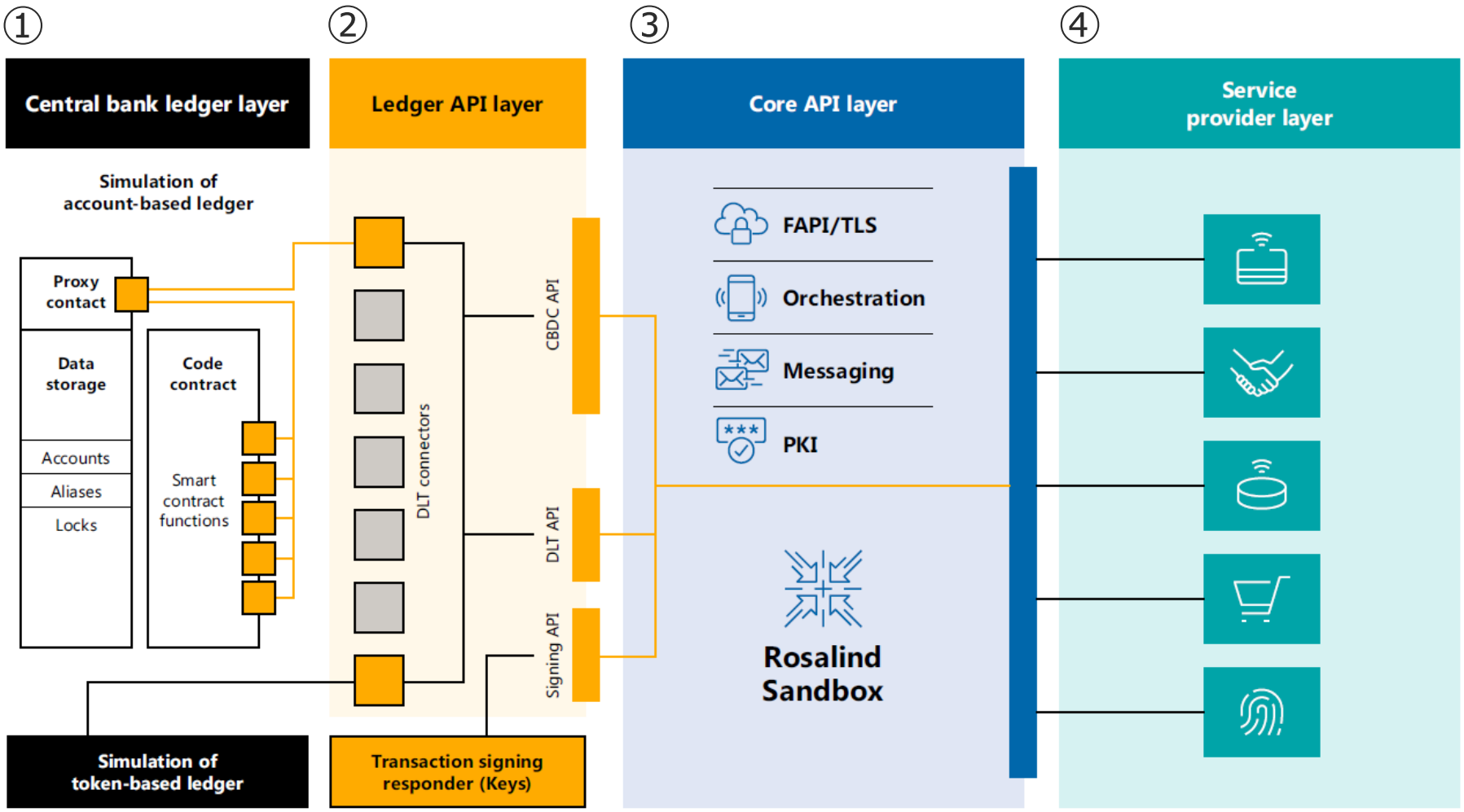
④エコシステム（ecosystem）に関する目的

✓ イノベーションを喚起し、デジタル包摂を支え、多様な送金手段の提供、優れたカスタマーアウトカムの実現のために、公的・民間セクターの参加者がどのように協働できるのかについての知見を獲得

2. 技術的設計

2-1. アーキテクチャ

- 以下の**4レイヤー**から構成



（出所）BIS Innovation Hub 「Project Rosalind final report」

2-2. 各レイヤーの概要

① Central Bank Ledger layer（中銀台帳レイヤー）

- 「アカウントベース」および「トークンベース」の両台帳を含む
  （**台帳構造の違いを吸収**できるかテストするため）
  - ■アカウントベースの台帳
    - Hyperledger Besuを使用
    - データストレージと一連のスマートコントラクトを含む
  - ■トークンベースの台帳
    - Hyperledger Fabricを使用
    - UTXOモデルをmimic（模倣/再現）するスマートコントラクトを実装

② Ledger API layer（台帳APIレイヤー）

- スマートコントラクトをAPIコールに変換し、変換後のAPIリクエストを中銀台帳が実行可能な形式に変換
- 複数のCBDC台帳の開発を加速させるため、Overledgerの技術を使用
  ※注：Overledger … 英国企業Quant Networkが開発したブロックチェーンに関するプラットフォーム
- 異なる種類の中銀台帳間の連携について調査

2-2. 各レイヤーの概要

③ Core API layer（コアAPIレイヤー）

- 本プロジェクトにおいて**重視**
- **機能性、相互運用性、ユースケース発見をサポート**するための重要な役割を担う
- すべてのAPIは、**FAPI**とTLS認証に準拠
- **End-to-Endの暗号化**をサポート
  （PII（personal identifiable information）が中銀に渡らないようにするため）
- **Rosalind Sandbox**（開発者ポータルや技術文書の中央レポジトリを含む）
- **6カテゴリ、33個のAPIエンドポイント**を通じ、メッセージとアクティビティをオーケストレーション

④ Service provider layer（サービスプロバイダーレイヤー）

- APIエンドポイントを通じて、コアAPIレイヤーと連携

2-3. 設計の前提

- CBDCは、発行中銀の直接の負債である
  - サービスプロバイダーは、自らの負債を発行しない
- すべての取引は、中銀台帳上でリアルタイム かつ 1対1で決済
  - 他取引とのバンドルやネッティングは行わない
- CBDCは付利されず、商業銀行マネーと1対1で交換可能
- 各ユーザの**CBDC保有額**や、各決済における**CBDC取引額**について、**制限なし**
- **APIや中銀台帳は、ユーザーのPIIや取引情報を閲覧/参照できない**
  - これらの情報はPIP（payment interface provider）レベルで保持され、APIレイヤー通過時に暗号化される

2-3. 設計の前提：サービスプロバイダーの役割

▼ サービスプロバイダーとして、以下の2種類が存在

PIP（payment interface provider）

- 個人や企業等のユーザーに対して、CBDCの口座と残高を管理するための「デジタルパススルーウォレット」 を提供
- ユーザーとのコネクションの管理や、各種指示（支払、他者に対する支払要求、受入）の実行を担う
- ユーザーに対して**のKYCの実施**や、**AML/CFT規制の遵守**が求められる
- Rosalind APIにおいては、PIPがデフォルトのサービスプロバイダーであり、**すべてのAPIエンドポイントを使用可能**
- PIPは、ユーザの取引履歴を保持する必要がある
  - Rosalind APIでは、取引履歴を取得するためのAPIは提供されない

ESIP（ecosystem service interface provider）

- ウォレットサービスを提供しない
- ビジネス分析、budgeting (資金管理) 、不正監視のためのツール提供等の**付加価値サービスに特化**
- **PIPと連携**して、**プログラマビリティサービスを提供**
  - 例：Three-party lockの意思決定者、送金依頼を起票する第三者

2-4. API設計の原則

原則	アプローチ
① **業界標準**に則る	• **REST API**（Open API Specification）、**JSON**を採用 • **FAPI認証**を実装
② 異なるシステムや技術間の**相互運用を可能**とする	• **アカウントベースとトークンベースの両台帳にて実験**
③ **標準化**する	• **ISO20022**（カットダウンバージョン）を採用 • A/Bテスト等を通して、**適切な大きさ** (right-sized) のAPIサービスを構築する方法を検討
④ **拡張可能**とする	• **拡張性最大化**のため、**サービスプロバイダーが**APIセット上に**カスタムメイドの機能性** (bespoke functionalities) **を追加することを許可** • **モジュール方式で設計**：シンプルかつ標準化されたAPIセットを様々な方法で組み合わせることで、**複雑で高度なユースケースをサポート**
⑤ **良質な開発者体験**を提供し、イノベーションを促す	• **Rosalind Sandbox**を用意：技術的書類やガイダンスの提供、ユーザーの問題提起やフィードバックのために使用

2-5. APIの機能

▼ APIカテゴリ：**Account**

サブカテゴリ	エンドポイント	説明
Account management	Open	中央銀行台帳上に、口座（parent account）を作成。口座の種類には、個人と法人がある
	OpenSubAccount	中央銀行台帳上に、特定の口座に紐付くサブ口座（sub-account）を作成。口座の種類には、個人と法人がある
	Disable	口座名義人は、中央銀行台帳上の口座、または、サブ口座を無効化することができる。一度無効化された場合、その口座では何もできなくなる
	Enable	無効化された中央銀行台帳上の口座、または、サブ口座を有効化
	Freeze	中央銀行台帳上の口座、または、サブ口座を凍結させることを、口座名義人に許可する。凍結された口座への入金は可能であるが、出金や支払いは不可となる。
	Close	口座を閉鎖
Alias	Alias	口座に紐付けるエイリアスを作成
	DeleteAlias	エイリアスを（論理的に）削除
	LookUpAlias	エイリアスの詳細を返す
Balances	Balances	口座の総残高を返す
	AvailableBalances	口座の総残高と利用可能残高（ロックされているものは除く）を返す

- **ユーザー (個人・企業)** は自身のCBDC口座の管理が可能 (例：口座開設、口座凍結、残高照会)
- ユーザーがCBDCにアクセスできる端末を紛失した場合、ユーザーは口座の**一時凍結リクエスト**を送信可能。
  なお、一時凍結中でも、ユーザーは他ユーザーからの支払い等でCBDCを受取可能
- **エイリアス機能**は、既存の決済インフラとの相互運用性の向上、およびプライバシーのサポートに寄与

※Rosalind APIでは、取引履歴の取得機能は提供しない

2-5. APIの機能

▼ APIカテゴリ：**Payments**

サブカテゴリ	エンドポイント	説明
Push payments	Pay	CBDCを他の口座に移転
	SplitPay	CBDCを複数の口座に移転
Request to pay	RequestToPay	他口座に対して支払を要求
	Authenticated RequestToPay	“RequestToPay”の要求に 支払人の認証情報を含められるようにすることで、当該支払を支払人のPIPが自動的に承認できるようにする（POS端末への適用等を想定）
Fund and defund	Fund	口座にCBDCを入金
	Defund	口座からCBDCを出金

- **Request to pay (RTP)** と**Authenticated request to pay (ARTP)** をサポート
  - RTP … 受取人が支払人に対する支払リクエストを作成する。支払人は支払人のPIPを介して支払リクエストを承認または拒否する。承認されれば、支払人から受取人に対しての支払（push payment）が行われる
  - ARTP … 受取人が支払人の認証情報を含めた支払リクエストを作成する。認証情報を用いて、支払人のPIPによる支払リクエストの自動承認（支払人による確認を介さない承認）がなされ、支払人から受取人に対しての支払（push payment）が行われる
- 逆引送金 (pull payments) はサポートせず
- ESIP等のサードパーティーによる送金指示の起票をサポート
- 受取人がオンラインだが支払人がオフラインの場合におけるPOS取引をサポートしうる

2-5. APIの機能

▼ APIカテゴリ：**Programmability**

サブカテゴリ	エンドポイント	説明
Set locks	RequestToLock	アカウント内の資金を下記3種類のいずれかのメカニズムでロックするリクエストを送信
	TwoParty	口座内の一定量のCBDCをロック。ロックの解除および資金の解放の決定は、受取人のPIPが行う
	ThreeParty	口座内の一定量のCBDCをロック。ロックの解除および資金の解放の決定は、適切な権限を持つサードパーティのPIPが行う
	HTLC	HTLCを用いて、口座内の一定量のCBDCをロック
Cancel locks	CancelLock	口座に設定されているロックを解除
	DrawDownLock	口座に設定されているTwoPartyロック、もしくはThreePartyロックを解除し、資金を引き出す
	DrawDownHTLC	口座に設定されているHTLCロックを解除し、資金を引き出す
Locks information	LockbyLockID	指定したアクティブなロックの情報を取得する
	LockbyPIP	指定したPIPを持つ全ての口座に関して、アクティブなロックの情報を取得する
	LockbyAccount	指定した口座に関して、設定されているアクティブなロックの情報を取得する

- 3つのロックメカニズムをテスト。すべてに有効期限がある（無期限にロックされることはない）

■ **Two-party lock**
- ✓ **取引当事者 (PIP2者) 間の高度な信頼関係**があることが前提。受取人のPIPは、支払条件が満たされたかどうかを判断する責任を負う

■ **Three-party lock**
- ✓ **取引当事者(PIP2者)間の信頼関係は不要**。支払実行のタイミングを決定する**サードパーティー**を導入

■ **HTLC (hash timelock contract) lock**
- ✓ **アトミックスワップ**をサポート。HTLCをサポートした他システムと統合しうるポテンシャルがある

2-5. APIの機能

▼ APIカテゴリ：**Participants**

サブカテゴリ	エンドポイント	説明
–	Key	特定のPIPの公開鍵を返す。この鍵は、PIP間で安全なデータを送信するために使用
	Notification	API経由で通知を引き出す。同規格のウェブフックも提供

- サービスプロバイダー間の暗号化のための公開鍵の取得
- APIコール送受信後の通知

▼ APIカテゴリ：**ESIPs**

サブカテゴリ	エンドポイント	説明
Connectivity	ConnectAccount	口座をサードパーティーのアプリケーションやマーチャントに接続
	DisconnectAccount	口座をサードパーティーのアプリケーションやマーチャントから切断。これは、口座名義人、もしくは、接続先によって呼び出し可能

- ユーザーのCBDCアカウントをサードパーティーアプリに接続/切断することを可能にする
- サードパーティーによる多数のユースケースをサポートできる(例：支払開始/スマートコントラクト/budgeting)

▼ APIカテゴリ：**Offline**

サブカテゴリ	エンドポイント	説明
Download and upload	Download	口座名義人のオンラインウォレットからCBDCを引き落とし、口座名義人のオフラインウォレットにCBDCを追加
	Upload	口座名義人のオフラインウォレットからCBDCを引き落とし、口座名義人のオンラインウォレットにCBDCを追加

- ユーザーによるCBDCのオフライン決済を可能にする

3 .ユースケース

3. ユースケース一覧

- 上述のAPIエンドポイントを活用し、以下のユースケースを検証

A: “Showcasing event” にて提示

#	ユースケース	
1	支払い	• 顧客が店舗のQRを読取 • オンライン支払で、顧客が店舗からの支払依頼を承認 • 店舗が顧客のQRを読取（顧客の承認操作は無し） • 電子レシートの授受
2	親子ウォレット	• 詳細不明（親が子のデジタル支払い経験をガイドしたり、親が子にお金を稼ぐことや責任ある支出について教育するための機能）
3	代金相当額のリザーブ	• 商品注文時に顧客口座のうち**代金相当額をリザーブ**（留保）、商品の到着と同時にリザーブを解除 • CBDCと民間マネーとの相互連携についても検証
4	オフライン台帳	• 即時・最終決済を伴う連続的なオフライントランザクションを実行
5	既存カードネットワークとの接続	• 個人が外国を旅行した場合に**CBDC決済**を実施
6	マイクロペイメント	• 企業のサステナビリティ戦略に関するもの • 駐車アプリ（分数に応じて課金）

3. ユースケース一覧

B: “TechSprint demo day event” にて提示

#	ユースケース	
1	ワンクリック支払い・ 声紋認証支払い	• サブスクリプション（定期支払）や請求書支払いにおいて、以下を利用する場合の顧客ウォレットとマーチャントの紐付け ✓ ワンクリック支払い ✓ **声紋認証支払い**（スマートアシスタント機能を利用）
2	エネルギー関連請求に対する政府支援	• 市民宛ての**エネルギー関連請求**に対する、**政府からの動的（dynamic）かつリアルタイムの支援**
3	オフライン決済によるポイント付与	• オフライン決済で**ポイントを付与**。蓄積したポイントは、慈善事業に利用可能。ユーザーはNFTの報酬を得る
4	DIDs・VCとの連携	• **DIDs**（decentralized identifiers）・**VC**（verifiable credentials）**との連携**について、以下を検証 ✓ 中銀APIを通じて伝達・蓄積されるデータの最小化 ✓ 顧客導入プロセスの迅速化
5	貿易	• 三者間エスクロー決済（輸入代金のロック、および貨物受取時の自動ロック解除）
6	個人による個人情報の利用	• オープンバンキングを通じて、個人が、銀行によってverifyされた個人情報をCBDC口座開設や支払いに利用

3. ユースケース一覧

#	ユースケース	
7	仕事の完了段階に応じた支払い	• ノマドワーク（itinerant economy）やギグエコノミーにおいて、**仕事の完了に応じてプログラマブルペイメントを履行**
8	鉄道チケットの購入および即時返金	• 鉄道チケットの購入、および**遅延時の即時返金**。返金プロセスはすべての関係者が閲覧可能
9	旅行者対応	• 旅行者が、オンライン・オフライン両方でのセキュアな支払いのために、e-Simをセットアップ。資金を消費する”time limit”がある中で、CBDCを安全に保管する
10	オフラインスマートカード	• **オフラインスマートカード・オンラインCBDCウォレット間の資金移動**。偽造や二重支払いは、オフライン台帳を通して監視
11	グループ内での貯蓄と借入	• 個人ユーザーに選ばれた**特定グループ内での貯蓄と借入**
12	POSとの連携	• 既存のPOSインターフェースとの連携

４ .ユーザーフィードバックおよび結論

4-1. ユーザーフィードバックと将来改善しうる点

ユーザーフィードバック

- Rosalind APIは比較的シンプルで使いやすく、複数の支払レグ（multiple payment legs）をしっかりと（robust）処理できた
- 効率性とパフォーマンスを向上させるには、非同期的なAPIモデルが探求されうる
- 基本的な機能（分割払い、ロック/解除、４桁の小数等）を組み合わせて利用することは、多くのイノベーティブなユースケースをサポートするポテンシャルを持つ

将来改善しうる点

- 条件付分割支払いのサポートのために、プログラマビリティ（ロック/解除フロー等）をさらに発展させうる
- Rosalind APIにおけるエイリアスと暗号化は初期的なものであり、本番環境で実装するためにはさらに多くの開発が必要
- HTLCロックについては、秘密に32バイトの値を16進数にエンコードしたものを、ハッシュアルゴリズムにSHA256を採用。今後異なる秘密の長さやハッシュアルゴリズムを採用しうる
- Rosalind APIは正常系（CBDC決済がスムーズに完了するパターン）のみにフォーカスしたため、エラーステータス設定、代替path、復旧処理といった点では不十分
- Rosalind Sandboxについては、開発者向けのドキュメンテーションを強化する余地がある（テストスクリプトやマルチメディアコンテンツの提供等）

4-2. 示唆および結論

示唆

- シンプルかつ中核的な（core）API機能のセットは、多様なユースケースをサポートしうる
- Rosalind APIの多くの機能は、アカウントベース、トークンベースどちらの台帳とも動作しうる
- オフライン機能については多くの課題（challenges）が残る
- API設計において、拡張性と一貫性はトレードオフ
- エコシステム参加者全員について、オペレーション上の役割と責任を定義する必要性がある
  - 例として、１ユーザーを複数のサービスプロバイダーに紐付け、サービスプロバイダーを簡単に切替可能にするケースでは、デフォルトの送金受取人を誰にするか、ユーザーの代理として送金を起票可能なサービスプロバイダーをどう設定するか、問題が発生した際にどの主体が責任を負うべきか、等を定義する必要がある

結論

- 本プロジェクトにより、よく設計されたAPIレイヤーは、中銀台帳とサービスプロバイダーを安全に連携させうることが示された

※本プロジェクト中、累計で、APIコール 541,865件、通知 7,652件、エラー 1,595件が作成/記録された。

参考資料

（参考） プライバシー

- 各参加者は共に、必要最小限の情報にのみアクセスすべきであるという考えのもと、以下のようなプライバシーモデルが検討された
  - Rosalindでは、仮名IDのみ生成される
  - PIIは、支払データや取引履歴と共に、サービスプロバイダーによって保持される
  - サービスプロバイダーの身元は、中銀に対して匿名化される
  - ユーザーデータは、ユーザーの許可を得ることで、サービスプロバイダー間で共有可能
    - ✓ 中銀と共有することはできない
    - ✓ サービスプロバイダー間のデータ共有時は、暗号化のうえ、コアAPIレイヤーを通過する
  - トランザクションは完全には匿名化されない

（参考） セキュリティ

▼ 4分野のベストプラクティスを適用

①ユーザーの権限についての認可(User authorization)

- Rosalind APIでは、 PIP がユーザーに代わり、資金の支払いやロックなどのAPIコールを実行できるようにするための、技術的な実現可能性を検討

②支払取引の認証 (Authentication of payment transactions)

- FAPI標準を適用
  - APIコールは、TLS1.2 以降を使用して HTTPS 経由で行われた

③否認防止 (Non-repudiation)

- JSON Web Signatures (JWS) を用いて取引を署名し、否認防止を検証
  ※否認防止 … 取引当事者が、メッセージを送信した事実の否定、あるいは、支払を後で拒否できない等を保証する機能
  - なお、今回はユースケース開発の加速のため、否認防止機能は無効化された
- トランザクションの処理速度・能力について、否認防止機能による低下が懸念される

④アンチリプレイ (Anti-replay)

- 悪意のある攻撃、あるいは、アクシデントによる二重決済を防ぐため、各書き込みAPIコールに、ユニークかつ冪等なID (unique idempotency ID) を含めることを求めた
  - 同一IDのAPIコールは受け付けない

（参考） 業界標準の適用

▼ API開発の出発点として、関連する業界標準をどのように実装できるかを検討

ISO20022の適用

- 下記のISO20022のメッセージング標準の実装を試行し、Rosalind API上で動作することを確認
  - pain.013：Request to Pay (RTP) API
  - pacs.008：payment APIs
  - camt.103：ロックのリクエスト、および、ロックに関するAPI

取引主体識別子（LEI）の適用

- サービスプロバイダーの識別のため、取引主体識別子 (LEI: Legal entity identifier) も使用

※LEI… ISO17442に基づく20文字の英数字コード