【CBDC フォーラム WG2】

CBDC の開発・発展のための
アーキテクチャ要求とプロセス要求

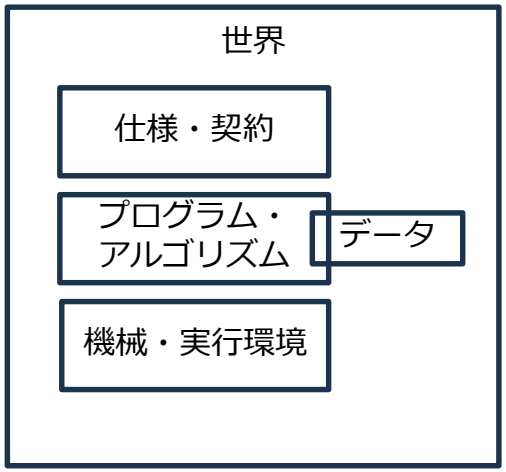
栗田 太郎
ソニー株式会社
2024 年 3 月 27 日

目次

- はじめに
- ソフトウェアの作り方
- システムの品質
- 構成・構造とプログラマビリティ

形式手法: 仕様とプログラム

- 仕様＝事前条件・事後条件・不変条件（≒契約）



- プログラムにより仕様を実現する（契約を履行する）
  （下記の組み合わせから成る．A～E は処理・計算）
  逐次実行: A して B する
  条件分岐: C の場合 D する
  繰り返し: E である間 F する

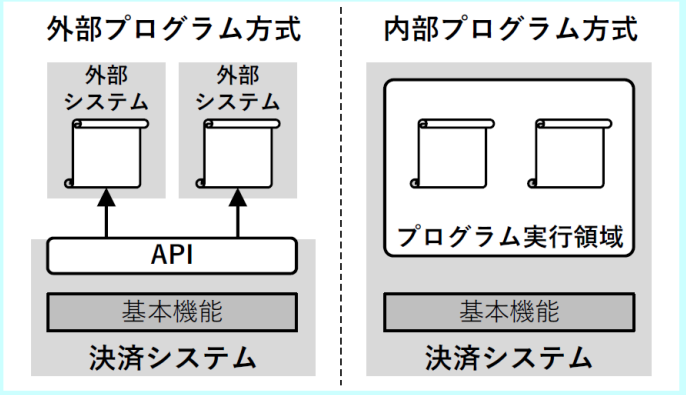
- ソフトウェアの検証は非常に難しい
  部分正当性: 事前条件が満たされるとき，事後条件が満たされる
  例: a>=0 であるとき RESULT * RESULT = a
  停止性: プログラムは必ず停止する

→ プログラムにバグがないことを証明することはほとんど不可能である

プログラマブル決済 と プログラマブルマネー

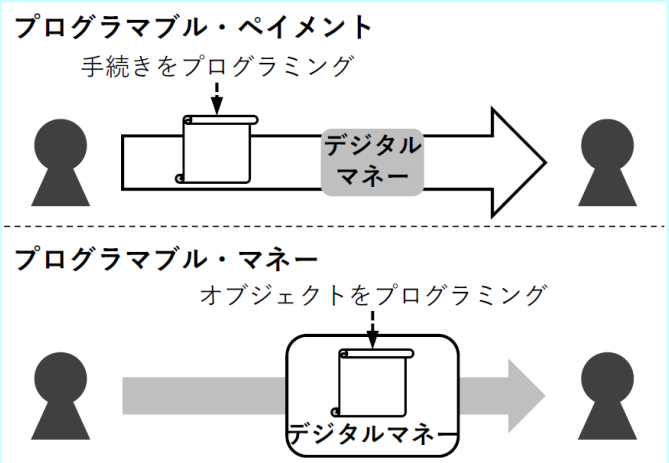
日本銀行『決済システムにおけるプログラマビリティの実現』から

【図表 4】外部プログラム方式と
内部プログラム方式



事例	
・銀行 API ・資金決済システムの API（NPP、全銀システム）	・オーバレイサービス（NPP）

【図表 5】プログラマブル・ペイメントと
プログラマブル・マネー



問い

L3 各種アプリケーション・サービス・デバイス
L2 日本銀行と仲介機関によるプラットフォーム（API・SDK・サンドボックス）
L1 CBDC コアシステム
L0 要素技術

「API は現在の銀行 API とイーサリアムの中間くらい」（栗田解釈）とはどのようなものになるのか?

（このようなものは突然完全なかたちで現れないとして）

- CBDC に関係するシステムはどのような構成になり，どのように発展していく（ようにする）のか?
- システムのアーキテクチャ要求・プロセス要求としてどのようなものがあるのか?
- ペイメントやマネーのプログラマビリティはどこにどのようにあるのか，ないのか?
- WG2 の活動（API サンドボックスプロジェクト等）において検証したい仮説はあるのか?

クネビンフレームワーク

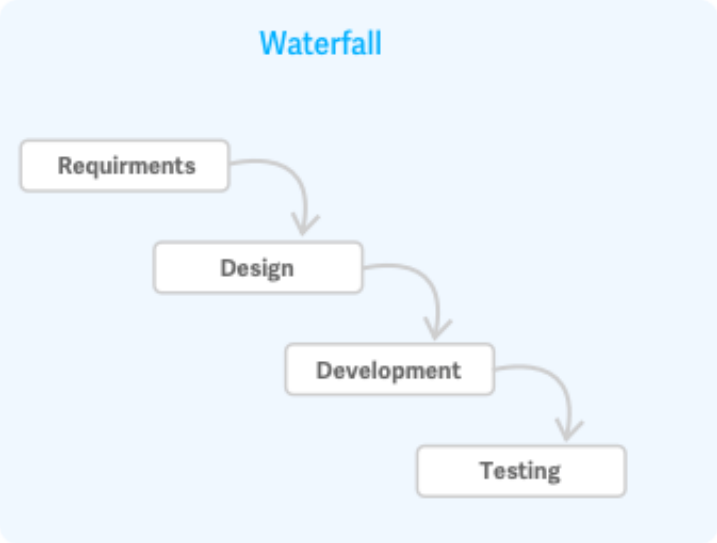
- 単純（Simple）な問題: 解決策は既知
  ベストプラクティスがあるかもしれない.

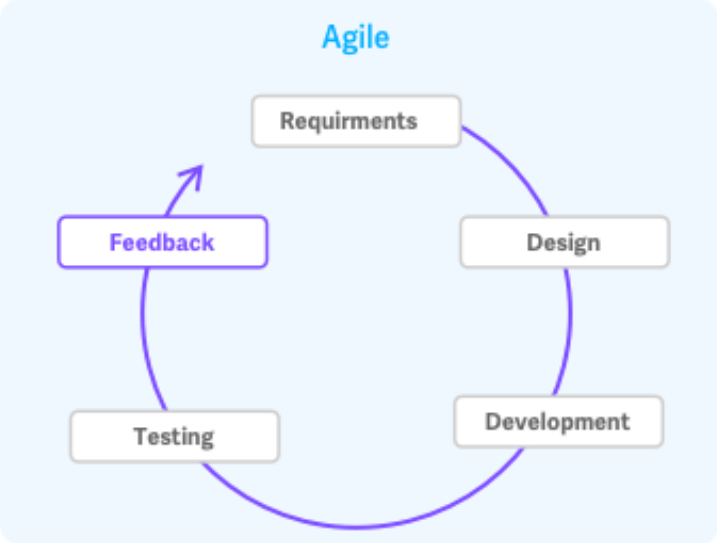
- 込み入った（Complicated）問題: 解決策を求め得る
  個々の問題の解は既知であるが，解決すべき問題が複数あり，それらの解が競合する可能性がある．例えば，手軽さとセキュリティとか.

- 複雑な（Complex）問題: 解決策を知り得ない
  個々の問題の解が未知であるだけではなく，解が競合する可能性がある．また，問題を取り巻く環境は変化し続ける．解決案を適用・評価し，観察・課題抽出することを繰り返す必要がある.

- 混沌とした（Chaotic）とした問題: そもそもの問題が不明
  問題が不明，解決策は未知，環境も変化．現在の技術の守備範囲外である.

ウォーターフォール と アジャイル





- フェーズに分かれる
- フェーズは，論理的に関連のある活動の集まりであり，1 つ以上の成果物の完成をもって終了する
- 比較的大規模な仕事に向いている
- 後戻りは難しい
- 初期の計画とリスク管理が重要となる

- 何らかのフードバックを得ながら高速に実験する
- 2～3週間～2～3ヶ月でループする
- フィードバックの例としては，チーム外の関係者の意見，成果物や生産性の定量的な品質データなどがある
- 少人数のチームの仕事に向いている

計画重視型 と 繰り返し型

- 最初の計画を重視する
  - 大規模な仕事に適している
  - 想定外に弱い．リスク管理を重視する
  - 見積もりは難しい
  - 従来のプロジェクト．計画があると大人数・複数組織で分担・管理しやすい

- 繰り返す（計画がないわけではない）
  - 状況変化に「素早く」対応できる
  - 大きな手配や覚悟ができないかもしれない
  - 行き当たりばったりになる可能性がある
  - 行き着いた先が成果物になる．成果物がこじんまりとする可能性
  - 少人数の自律型（自己管理型）のチームだと「アジャイル」になる
  - 2000 年代以降，ソフトウェア開発を中心に発展してきた

システム・ソフトウェア品質について

品質

ISO 9000（品質マネジメントシステムに関する標準）における定義：対象に本来備わっている特性の集まりが，要求事項を満たす程度

ISO/IEC 25010（ソフトウェア製品の品質要求および評価に関する標準，通称 SQuaRE) における定義：様々なステークホルダーの明示的または暗黙的なニーズを満たす，すなわち価値を提供する程度

Verification と Validation（V&V）

- Verification=検証
  「正しく作っているか」
  “Are we building the product right?”

- Validation=妥当性確認
  「正しいものを作っているか」
  “Are we building the right product?”

CBDC の基本的特性・システム面の特性

中央銀行グループ・BIS「中央銀行デジタル通貨: 基本的な原則と特性」から

CBDC の基本的特性　表 1

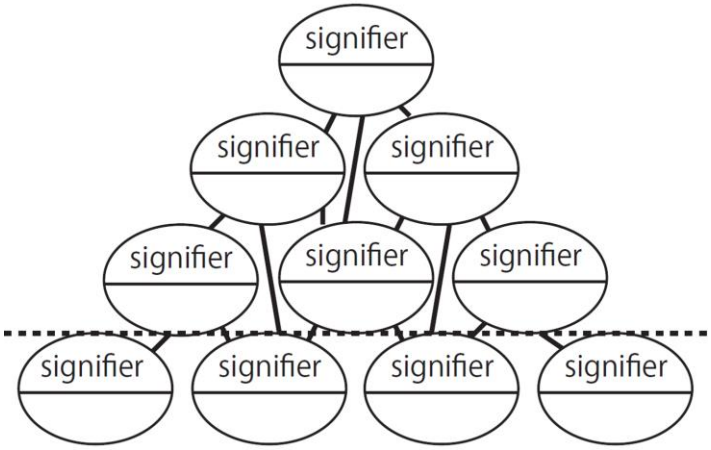
機能面の特性	
交換可能性	**通貨の単一性を維持**するため、CBDC は、現金および民間マネーと等価で交換されるべきである。
利便性	CBDC による支払は、幅広い利用とアクセス可能性を促進する観点から、**現金の利用**、カードのタッチ、あるいはモバイル端末のスキャンと**同様に、簡便**であるべきである。
受容性および利用可能性	CBDC は、現金と同様に、店頭および個人間取引を含む多くの取引に利用可能であるべきである。これには、一定の**オフライン取引**（利用期間の制限および予め定められた閾値を上限とすることが考えられる）**を行う機能**も含まれる可能性がある。
低コスト	CBDC による支払は、**エンドユーザーにとって、非常に低いコストか無償**であるべきである。また、エンドユーザーに求められる技術的投資は最小限であるべきである。
システム面の特性	
安全性	CBDC システムのインフラおよび参加者の双方は、**サイバー攻撃**やその他の脅威**に対し、極めて強靭**であるべきである。また、偽造に対する効果的な防止策も確保されるべきである。
即時性	即時あるいはほぼ**即時のファイナリティのある決済**をシステムのエンドユーザーに対し提供すべきである。
強靭性	CBDC システムは、**運行上の障害や中断**、自然災害、停電およびその他の問題**に対して極めて強靭**であるべきである。ネットワークに接続が出来ない場合、エンドユーザーがオフライン決済を行えるよう、何らかの能力を有するべきである。
利用可能性	エンドユーザーは、システムにおいて、**24 時間 365 日**、常に決済可能であるべきである。
処理性能	システムは、**極めて大量の取引**を処理することが可能であるべきである。
拡張性	**将来、取引量が増加した場合においても処理可能**であるよう、CBDC システムは拡張可能であるべきである。
相互運用性	システムには、民間部門のデジタル決済システムとの十分な相互作用メカニズムや、**システム間の資金フローを容易**にする取扱いが必要である。
柔軟性および適応性	CBDC システムは、**環境変化**や政策要請**に柔軟に適応**できるべきである。
制度面の特性	
頑健な法的枠組み	中央銀行は、CBDC の発行を支える**明確な権限**を有するべきである。
基準	CBDC システム（インフラおよび参加主体）は、**適切な規制上の基準に適合**する必要がある（例：CBDC の移転、保蔵あるいはカストディ業務を提供する主体は、現金や既存のデジタルマネーにおいて類似のサービスを提供する企業と同等の規制、監督基準に服さなければならない）。

ISO15408 第三者セキュリティ評価・認証

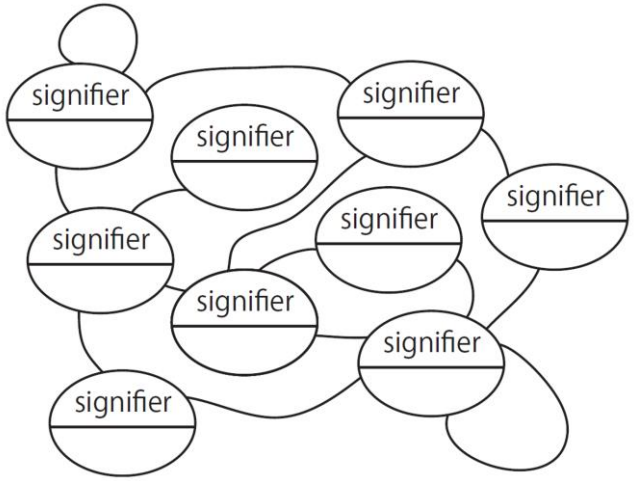
- 規格に基づく第三者によるセキュリティ検証
- レベルが 1～7 まである
- セキュリティターゲットと呼ばれる仕様により利用者とコミュニケートする

構造 と 構成

【還元論（reductionism），構成，絶対】



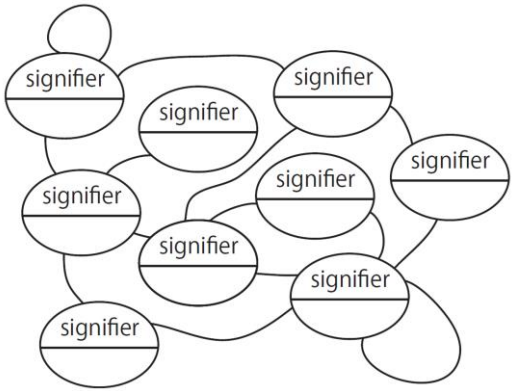
【全体論（Holism），構造，相対】

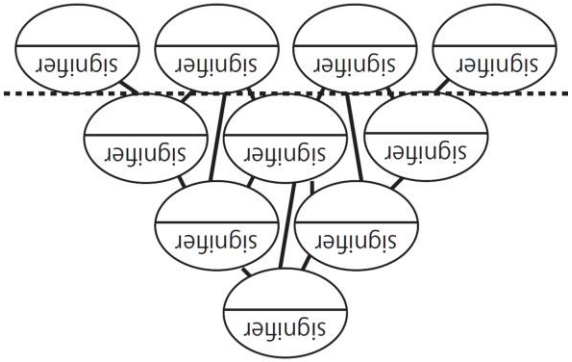


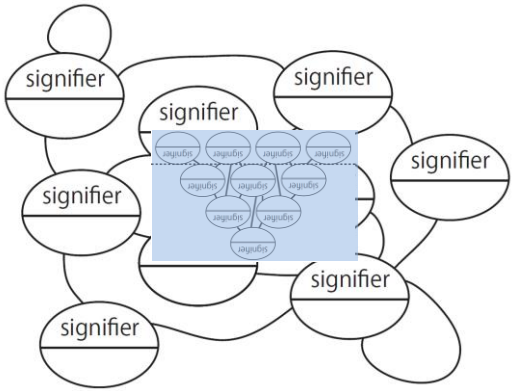
- 還元論では，全体は部分に分解（還元）できるとする
- 全体論では，各要素が有機的に結びつく．システムには創発特性がある

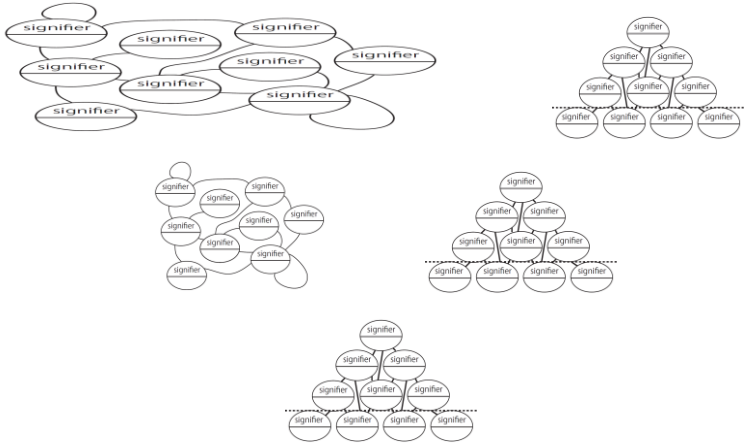
- あらかじめ計画できないものごとの構成を決めきることは難しいし，拡張により破綻することがある
- 全体論は発展が可能だが，カオスにならない構造の制約を見いだす必要がある

CBDC のアーキテクチャパターン

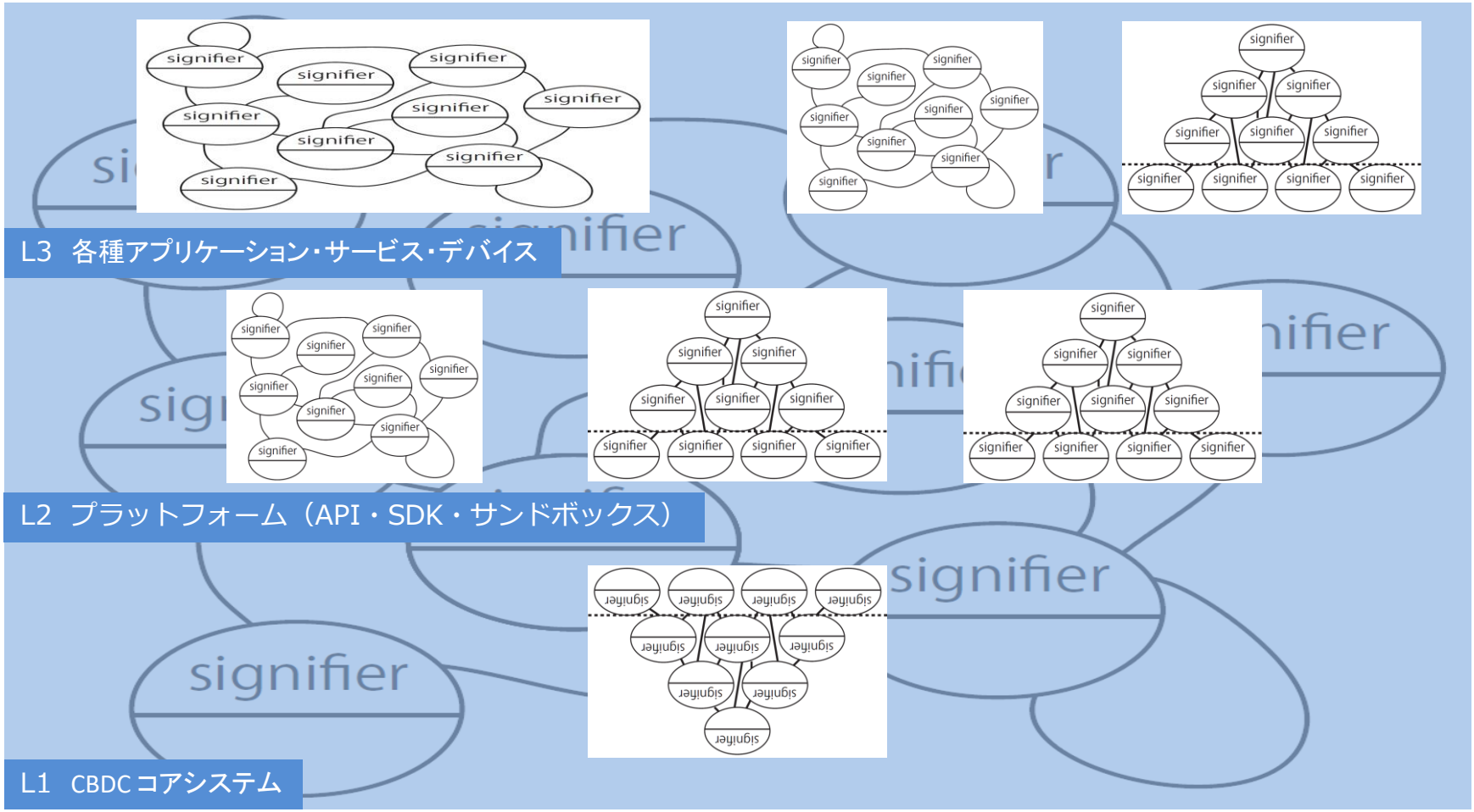








CBDC アーキテクチャパターンのひとつ



構成・構造の要素と要素間のパターン

- データ（データ中心（日本語））
- プロセス（プロセス中心）
- データ + プロセス（オブジェクト（＝型）指向）

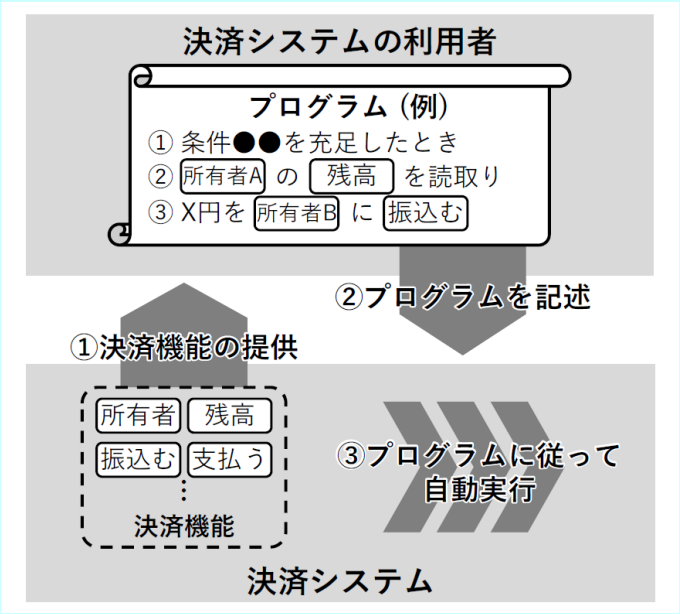
- プロセスの発動方法の例
  - ある○が複数の○に対して指示を送っていく
  - ある○がメタプログラムを別の○に送る
  - ある○であるオブジェクトが動作する

- プログラマブル : A programmable machine can be programmed, so that for example it will switch on and off automatically or do things in a particular order. (COBUILD)

プログラマビリティ

日本銀行『決済システムにおけるプログラマビリティの実現』から

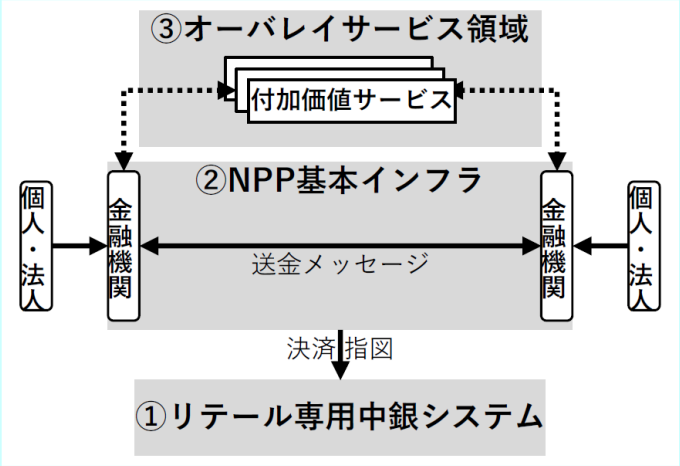
【図表 1】決済システムにおける
プログラマビリティ



【図表 2】銀行 API により実現される
組込型金融

（注）各種資料を参考に作成

【図表 3】NPP のシステムアーキテクチャ



①リテール専用中銀システム：豪準銀が NPP に合わせて提供を開始した、24/7 稼動の小口専用 RTGS システム

②NPP 基本インフラ：各参加者を繋ぎ、送金メッセージの伝送等の共通機能を提供するネットワーク

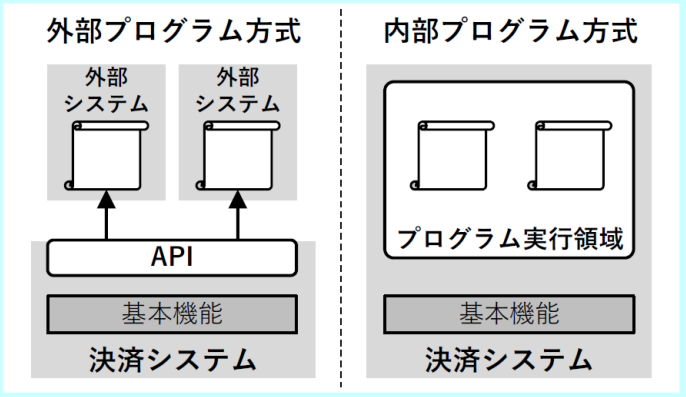
③オーバレイサービス領域：フィンテック企業等の事業者が付加的なサービス機能を組込む領域

（注）NPP に関する公表情報を参考に作成

プログラマブル決済 と プログラマブルマネー

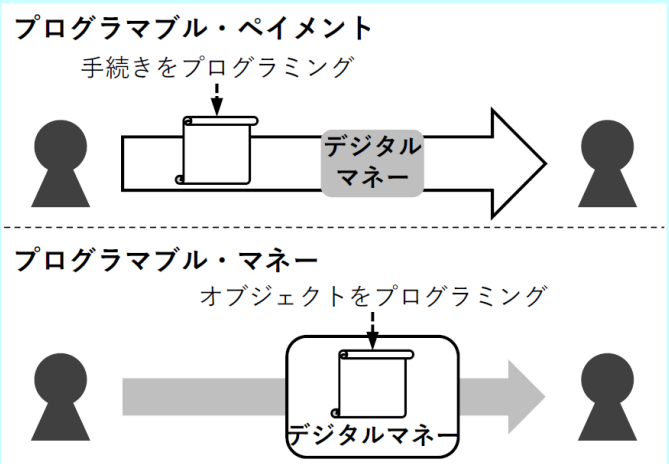
日本銀行『決済システムにおけるプログラマビリティの実現』から

【図表 4】外部プログラム方式と
内部プログラム方式



事例	
・銀行 API ・資金決済システムの API（NPP、全銀システム）	・オーバレイサービス（NPP）

【図表 5】プログラマブル・ペイメントと
プログラマブル・マネー



プログラマブルに対する要求

- だれ（専門家・非専門家・国民全員）でも書く（書かせる）ことができる．実行できる
- システムの利用に自由度がある（機能を動的に追加して実行することができる）
- いつ何に対してどのようなことが行われるのかといった情報情報がまとまっていて利用者に理解可能である

- プログラムが検証可能であることが望ましいが，前述の通り，難しい
- 「プログラマブルな機能の多用は，より高い水準のシステムの処理性能を必要とするほか，開発費用の増加や強靭性の低下を生む」（「中央銀行デジタル通貨：基本的な原則と特性」エグゼクティブ・ペーパー）

【参考】ニック・サボによるスマートコントラクトの要求

観測可能性: 当事者全員にとっての透明性・理解可能性
客観的検証可能性: 当事者と仲裁者による検証
契約当事者関係: 仲裁者以外が介入できないこと．機密やプライバシーが守られること
強制力: インセンティブや罰則も含めて強制力が働くこと

CBDC のアーキテクチャ要求・プロセス要求

- CBDC コアシステムの要求
- CBDC エコシステムの要求

- 可用性・高速処理・高負荷対応・高セキュリティ等々

- 改善・発展可能である
- コアシステムはシンプルにする
- エコシステムとして多様に応用できる．多様なシステムとつながる
- プログラマブルである（さまざまな方式・レベルがある）

- セキュリティを含む品質が担保される
- 説明責任が果たされる
- 利用者にとってシステムや各処理が理解できるものである

- （変わっていく）規格や法律や政策を満足する

論点

- 開発者・運用者はシステム全体をどう捉え，マネージするのか?
- 利用者はシステムをどう認識するのか?

- （そもそも CBDC システムとは?（コアシステムの API のことだけではのないだろう））

- 各レイヤの API は標準化するのか，できるのか
- 将来何か良いことが起きることを具体的に考えることはできないとしても，どのような創発特性を期待するのか?（あるいはしないのか?）
- システムの品質・セキュリティをどのように達成するのか?
- 利用者にとっての安心・安全をどのように実現するのか?

- CBDC ではないと実現できない，良いことが起こりそうなプラットフォームとは?
- コア台帳にどのような機能を有すると発展していくのか? あるいはどのような制約・型が必要なのか?