ブロックチェーン連携に対応した
スマートコントラクトベースの
トークン取引システムに関する研究

２０２３年　３月

藤本　真吾

ブロックチェーン連携に対応した
スマートコントラクトベースの
トークン取引システムに関する研究

藤本　真吾

理工情報生命学術院
システム情報工学研究群
筑波大学

２０２３年　３月

目次

図の目次

表の目次

第1章 序論

1.1. 研究の背景

電子署名技術と P2P(Peer-to-Peer)通信ネットワーク技術の組み合わせを核とし，分散型台帳に分類されるブロックチェーン技術は，暗号通貨ビットコイン（Bitcoin）を中央集権的な管理者を持つことこなく管理，運営することを可能にするために生み出された．ビットコインは 2009 年の運用開始以来，現在まで一度も停止することなく稼働を続けていて，高い信頼性が確保されている．

その一方，暗号通貨のユーザ間取引に現状不可欠なサービスである暗号通貨取引所では，不正な通貨引出しや，本来は価値のないデジタル資産との交換を持ち掛ける詐欺，などブロックチェーンへの信頼を損なう被害事例が度々発生している.これらの被害が発生した原因は，ブロックチェーン技術に起因したものではなく，暗号通貨取引所などの運営側の管理体制に不正が発生しうる人的要素が含まれていることに起因するものである．

このような状況を打開する取り組みとして，イーサリウム（Ethereum）などのモダンなブロックチェーン基盤には，ブロックチェーン・ネットワークにあらかじめ配備しておくことで，分散台帳の操作やユーザ間の直接取引を安全に実行できる管理プログラム，スマートコントラクト機能が実装されるようになっている．スマートコントラクトとして実行されるプログラムの挙動は，ブロックチェーン・ネットワークに参加するノードと呼ばれる計算機によって個々に検証される．この検証では，スマートコントラクトを起動したユーザが取引承認の証として付与した電子署名のチェックに加えて，スマートコントラクトを複数個所で同時に実行した結果が一致するかをチェックし，ブロックチェーン・ネットワーク全体でスマートコントラクト自身が正しく稼働しているかを確認できるようになっている．このため，スマートコントラクトを使うユーザは，仲介サービスから管理者による不正な介入の可能性も排除でき，安全なユーザ間取引ができる．

また，安全性の向上以外にもスマートコントラクトの導入にはメリットがある．オンライン商取引などでは，ユーザ間の直接取引を望んだとしても両者が同時にオンラインとなっているタイミングを見つけるのが難しい場合もあるが，ブロックチェーン・ネットワーク上の仮想マシン上で稼働させるため常にオンラインであることが保証される，スマートコントラクトを介した取引を行うことで，取引当事者の両者が同時にオンラインである必要がなくなり有用である．

スマートコントラクトの活用でブロックチェーン利用における安全性と利便性の向上への期待が高まる一方で，現状のスマートコントラクトは，単一のブロックチェーン基盤内で閉じた処理しか扱えない，与えられた起動パラメタの正しさを検証する場合に単純な内部状態遷移に基づいた判断ロジックしか使えない，などの制限が存在している．

ところで，オリジナルのブロックチェーン（＝パブリックチェーン）では管理者不在の完全自律型運営を目指したのに対して，ブロックチェーン技術をベースに企業ユースに特有な

課題を解決するための拡張を加えたうえ，ブロックチェーン・ネットワークを企業が構築し，管理者を置いて運用する，エンタープライズ・ブロックチェーン，もしくはコンソーシアムチェーンと呼ばれる亜種も登場している．

コンソーシアムチェーンでは，参加するノードの数が運営体に選ばれた管理者の高々数個の計算機に絞ることで，パブリックチェーンでは数分から 1 時間程度かかる，取引の確定までの時間をミリ秒単位にまで向上させることができている．また，その利用についてもブロックチェーン・ネットワークに直接参加することなく，Peer ノードと呼ばれるブロックチェーン・ネットワークと外界を中継する計算機を介した利用を想定しており，一般ユーザはノードを運営するより要求される計算資源が少ないクライアント機能のみで利用できるように工夫されている．Peer ノードを介したコンソーシアムチェーンの利用であっても，分散台帳の操作や問い合わせには，パブリックチェーンと同様の暗号強度を持った電子署名技術が用いられており，その運営についてもコンソーシアムを構成する複数の企業が担当することから，パブリックチェーンに対しての安全性におけるデメリットは少なくなっている．しかしながらコンソーシアムチェーン運営の公正性に「運営体への信頼」という非技術要素が含まれるため，第三者機関からの監査を容易とするようなシステム設計が不可欠と言われている．

したがって，コンソーシアムチェーンをパブリックチェーンの代替として活用することには心理的なハードルが存在すると考えられるが，このコンソーシアムチェーンの特性を活かした有望な活用分野として，トークン・エコノミーがある．

トークン・エコノミーは，無形の価値をデジタル資産“トークン”として取引可能にするトークン化の技術を使い，さまざまなトークンを個人間でも流通可能にする社会システムを構築し，新たな経済活動を生み出す概念である．

トークン・エコノミー構築のメリットには以下の 3 つがある：

1. デジタル完結の即時決済で資金の調達が容易になる
2. 支払い能力や真贋判定のチェックを含む信用取引を自動化
3. 取引サービスでの口座開設やユーザ登録が不要

トークンの発行と管理をスマートコントラクトで行う“トークン化”は，特定のトークン発行体が提供するクローズドなサービスであるが，先に挙げたイーサリウムの開発者コミュニティでは API の標準化も進んでおり，貨幣ライクなトークンを標準的な API で交換できる ERC-20[1]や，同様に特定の資産にマッピングされる NFT(Non-Fungible Token：非代替性トークン)を標準的な API で交換できる ERC-721[2]が制定されている．これらの API 仕様は多くのトークン発行体が利用するトークン管理用スマートコントラクトで採用されたことで，この仕様をサポートしている OpenSea などのトークン交換プラットフォームでは，多様なトークンを扱うことができ，発行体の関与なしに個人ユーザ間での直接取引を仲介することが可能になっている．

特にインターネットの世界では，非代替性トークンとして発行される NFT アート（アーティスティックなサルの NFT アートが有名）が高額で取引されており，その取引市場が過熱的

に盛り上がっている．本来は，それと連動してトークン・エコノミーも盛り上がるはずであるが，現実にはそうはなっていない．これは初回の分譲で高値を付けた非代替性トークンであっても，再販される際には大幅な値下がりが発生することが多いためである．「OpenSeaの無料機能で発行されているNFTの80%が偽物」と報じられるように，非代替性トークンの本来の価値以上に喧伝する詐欺が横行しているともいわれている．実のところ非代替性トークンの発行において，トークン化される資産価値の裏付けは必須となっているわけではなく誰でも発行が可能なため，非代替性トークンの購入を検討する場合には，トークン発行体の身元や，そのトークンへマッピングされているコンテンツのリアル社会での所有権との関係をチェックしないといけない，という事実に気づかないユーザが多い．

この非代替性トークン化されるデジタル資産の価値への不安を取り除く金融規制当局の取り組みとして，セキュリティトークン（Security Token）がある．セキュリティトークンは，分類上非代替性トークンの一種となるが，トークンの発行体に対して，リアル資産へのマッピングやトークンの代わりに預けられた資産の運用，など関連法令へのコンプライアンスが義務付けられるという特徴がある．厳格な法規制の導入で，セキュリティトークンでは資産価値への信用不安に対するリスクが低減されたものの，法規制上の信頼担保をブロックチェーンの外の世界，“オフチェーン”環境に求めた結果，トークン化によるメリットが活かしにくい状況にある．たとえば日本では，セキュリティトークンでの権利移転は，ブロックチェーン台帳への操作の他に法律的な確定条件として，取引が行われる度に中央集権的に管理される「債権原簿」へ反映して両者を同期させることを義務づけられている．

セキュリティトークンで投資家が利益を得る手段として，償還時期まで資金を預けて利子を受け取ったり，二次流通取引を可能にして取引市場での売買で価格変動差益を得たりする，などが一般的である．これに対して，セキュリティトークンの長期保有を促すアイディアとして，複数の専門家が配当付きセキュリティトークンを提案している．つまり，セキュリティトークンを購入した投資家に対して，トークンの保有中に提供するサービスを充実させることに潜在的な可能性があることが示唆されている．このようなトークンの利用者に特別な機能やサービスを提供する決済システムを「プログラマブル・トークン」と呼ばれる．

1.2. 研究の目的と意義

本論文の目的は，現状のトークン・エコノミーの問題を解決し，多様なトークン同士を交換する取引の仲介や，投資プロジェクトの運営などをスマートコントラクトで自動化された“真のトークン・エコノミー”の実現に向けた取り組みとして，独自方式のスマートコントラクトベースのトークン取引システムを提案することにある．

具体的には実現手段は以下のとおり：

1. セキュリティトークンと別種のトークンを組み合わせた連携動作を可能にする「拡張スマートコントラクト」を実現
2. 提案した拡張スマートコントラクトと不足機能を補うサブシステムを追加して，セキュリティトークンのライフサイクル全体での運用管理を自動化する「スマートコントラクトベ

ースのトークン取引システム」を実現

本論文での提案の骨子としては，トークン取引システム自身が独自のブロックチェーン・ネットワークとして運用され，別のブロックチェーン基盤上の台帳操作を扱えるように拡張した拡張スマートコントラクトが動作するようにした．

この拡張スマートコントラクトの振る舞いは，システム稼働後に操作履歴付きでインストール，実行される動作シナリオで決定され，取引価格などのブロックチェーン内では決定できないオラクル情報を管理者が設定，運用できるようになっている．さらに，動作シナリオに含まれる条件分岐や，一連取引の進捗状況の検出結果に応じて，取引代金の一時預りと返金や，返金や決済の条件を組み込んだ定型処理ロジックとしたエスクロー取引なども実現可能にしている．拡張スマートコントラクトは，ユーザによる起動後は非同期で実行が継続され，動作シナリオの終了条件を満たすまで自動実行が継続される．その稼働履歴は，連携専用ブロックチェーンの台帳操作として記録されるので外部の監査主体からも参照できるようになっており，スマートコントラクトの振る舞いについての透明性が確保されるようにしている．提案した拡張スマートコントラクトの実現性を確認するための評価システムとして，ブロックチェーン連携基盤技術「ConnectionChain」を試作した．また，外貨建ての店舗決済システムや，デジタル証券の即時決済などのオンライン・サービスを ConnectionChain の拡張スマートコントラクトとしてシステムを構築，実証実験を実施して今回提案する拡張スマートコントラクトの有用性を検証した．

次に，セキュリティトークンのユースケースを収集・分析を行って，セキュリティトークンの運用管理に拡張スマートコントラクトの機能を使い，セキュリティトークンのライフサイクル全体をカバーする，ユニークなトークン取引システムを提案する．

具体的には，セキュリティトークンの初回分譲で集めた資金の運用や，利益配分などのトークンの運営が他所で未着手であることに着目し，トークン発行で集めた資金の管理運用を行うことで，投資プロジェクトを拡張スマートコントラクトで自律的運用されるようになる．

汎用的なブロックチェーン連携システムとして設計，試作した ConnectionChain について，セキュリティトークン運営のモデルに対応させるため，以下の 3 つの機能を ConnectionChain に追加して，「スマートコントラクトベースのトークン取引システム」を実現した：

1. 第三者に対するプライバシに配慮した取引内容証明
2. オラクル情報を使った取引可否の判断
3. トークン保有ユーザを対象にした配当トークンの定期的な配布

1.3. 本論文の構成

本論文の構成として，第 1 章の序論では本研究の背景を説明する．

第 2 章では本研究に関連するブロックチェーン技術とその応用例について解説する．

第 3 章では，本研究に関連する先行研究や取り組みについてのサーベイを行う．

第 4 章では，本研究の目的である，ブロックチェーン技術を用いたトークン・エコノミーの実現，に必要な技術として他のブロックチェーン台帳を操作できるように拡張を施した拡張スマートコントラクトを提案する．また提案方式の実現可能性を検証するために，コンソーシアム型ブロックチェーン上でスマートコントラクトとして動作するように設計，試作した評価システム ConnectionChain について説明する．

続く第 5 章では，本研究の最終的なゴールであるトークン・エコノミーの実現に向けて，前章で提案した拡張スマートコントラクトを，デジタル有価証券（セキュリティトークン）向けのトークン取引システムに組むこむ手法を考案し，現状のトークン・エコノミーでは実現できなかった投資プロジェクトの自動運用を可能とする，スマートコントラクトベースのトークン取引システムの提案を行う．

第 6 章では，本研究での提案したコア技術である拡張スマートコントラクトを有用性と安全性の面で評価する．

第 7 章では本研究の成果についての総括を行う．

第2章 ブロックチェーン技術とその応用例

2.1. ブロックチェーンの仕組み

本論文のテーマであるトークン・エコノミーを支える技術として，ブロックチェーンの起源と仕組みについて解説する．

2.1.1. 暗号通貨とブロックチェーン

暗号通貨(Cryptocurrency)は，お金の価値を暗号技術デジタル化・マネーの一種である．
暗号通貨は一般に以下の性質を持つ：

- 中央集権的な権限が存在せず，その価値が分散ネットワークを通じて維持される
- 暗号通貨の数量とその所有者の記録を維持する
- 新たな暗号通貨の作成ルールや，利用条件はシステムが決定する
- 暗号通貨の残高は，暗号学的な検証によって証明される
- 暗号通貨の所有権は，システムに送信されるトランザクション(transaction)によってのみ変更可能となっている

ブロックチェーンは，この暗号通貨の元祖とされるビットコイン(Bitcoin)を管理・運営するために考案された技術である．ビットコインの興味深いところは，管理者が不在でありながら，2009 年の運用開始からまだ一度もサービスが停止していない堅牢な金融システムであるということである．

また，マスメディアなどで報道されているビットコインの取引総額から考えれば，その資産を盗み出そうとする犯罪者のターゲットになりそうであるが，単純かつ巧妙に設計されたコンセンサス・アルゴリズムで実現される取引検証のメカニズムにより，今日までビットコインのブロックチェーン・ネットワークから不正に資金を引き出すことに成功した者はいないとされている．

こうした暗号通貨の安全性とその取引市場の成功を受け，ブロックチェーンの技術を通貨以外のさまざまな資産管理に応用しようとする取り組みがある．具体的には，ブロックチェーン上に実装されたスマートコントラクトの活用した，より複雑なデジタル資産管理システムの構築や，トークン化された異なるデジタル資産同士を交換する仕組みとしてトークン・エコノミーが検討されている．

2.1.2. ブロックチェーンを支える暗号技術とコンセンサス・アルゴリズム

ブロックチェーンは，公開鍵暗号やハッシュ関数などの暗号技術と，Peer-to-Peer ネットワークという，“成熟した”技術をユニークな形で組み合わせたブロックチェーン・ネットワーク上に実現されている．

ブロックチェーンに使われている暗号技術のひとつである公開鍵暗号は，暗号化と復号と

で異なる２つの鍵（秘密鍵と公開鍵）を使用する暗号化方式である．公開鍵暗号は，インターネット上で安全な通信を行うための通信プロトコル “TLS(Transport Layer Security)”でも秘密情報をやりとりする場合の鍵交換に使われている技術である．ある秘密鍵で暗号化したデータは，それと対をなす公開鍵でしか復号できない特徴を持つ．この特徴を活かして，秘密鍵を所持者が他人に知られないよう安全に管理したうえ，公開鍵を積極的に公開し，秘密鍵で平文を暗号化したメッセージを送信，受信されたメッセージを公開鍵で復号した結果として元の平文が復号できた場合には，平文の送信を秘密鍵の所有者が送ったという事実を確認できる．

また，ハッシュ関数はブロックチェーン・ネットワークの安全性を高めるために使われているもうひとつの暗号技術であり，さまざまな形で活用されている．

その基本的な機能は，任意の長さのメッセージを入力から，固定長の出力メッセージを算出するという単純なものである．このハッシュ関数のうち，出力メッセージ（＝ハッシュ値）から入力メッセージが逆算できないほど無秩序な出力を行う性質を持たせたハッシュ関数を特に暗号学的ハッシュ関数（cryptographic hash function）と呼ぶ．暗号学的ハッシュ関数では，入力メッセージが少しでも違うとハッシュ値が変わるので，公的なメッセージ発信やソフトウェアの配布時にデータが改ざんを受けていないかのチェックにも使われている．

暗号学的ハッシュ関数と公開鍵暗号を組み合わせたのが電子署名技術で，署名対象のメッセージを間違いなく本人が発信したことを確認できる．

電子署名データの作成の手順としては，署名対象のメッセージを入力に暗号学的ハッシュ関数で算出されたハッシュ値を，メッセージを発信しようとするユーザが管理する秘密鍵で暗号化して署名データとして，署名対象のメッセージに添付して送信する．メッセージを受信した側のユーザが行う署名の検証では，送信者から送られてきた署名データを本人の公開鍵で復号した結果得られる値と，受信したメッセージを入力として暗号学的ハッシュ関数を使って再計算したハッシュ値，これらが一致するかで確認する．

ハッシュ値は，入力メッセージの長さに依らず固定長のデータとなるため，メッセージに直接公開鍵暗号を適用するより，通信量の観点で効率のよい改ざん検知が可能で，その結果として送信データの改ざん防止が実現される．また，事前に公開鍵の出自が確認されていれば，電子署名で秘密鍵を管理するユーザの認証を行うこともできる．

ブロックチェーンにおけるハッシュ関数の別の使い方にマークルツリーによる分散台帳に格納するデータの変更履歴管理がある．マークルツリーは，公開鍵暗号の開発者の一人であるラルフ・マークルによって 1979 年に発明されたデータ構造で，主に大きなデータの要約と検証を行う際に利用される．マークルツリーの基本形は，２つのデータを一つにまとめる処理である．たとえば，A と B という２つのデータがあったとして，まずは A の値を入力にしたハッシュ値と、B の値を入力にしたハッシュ値をそれぞれ計算する．そして，さらにこの二つのハッシュ値を連結した値を入力したハッシュ値を計算して，A のハッシュ値と B のハッシュ値を子に持つ二分木のルートノードにする．マークルツリーを実際に使う場合には，この基本形の処理をペアにした大量のデータ群のルートノードへの再帰的な適用を繰り返すことで得られる値“マークルルート”を頂点とした複数段のツリー構造を構成して利用される．

マークルツリーに格納された値の更新を行う際には，ハッシュ値の再計算が必要となるが，マークルツリーでは，子ノードとの依存関係から子ノードや孫ノードなどで値が変更された場合のみノードの値が変化することから．この性質を利用すると値が変更されなかった部分木のルートの値は再利用が可能でハッシュ値の計算を省くことができる．また，変更前のツリーとの比較をルートノードから順次行っていくことでツリー構造のどの部分で変更が発生したかを容易に検出できる．つまりマークルツリーには，どんなデータを入力しても固定長のハッシュ値に要約することができ，かつデータの変更箇所を容易に検知できる特性がある．ブロックチェーンでは，台帳操作を行う“トランザクション・データ（データ[a]）”をマークルツリーのノードとして管理していて，あるトランザクション・データが承認済みか検証するための基点としてマークルルートの値が使われている．

ブロックチェーンにおけるさらにもう一つのハッシュ関数の利用法がハッシュチェーンである．電子署名は，その作成者が確かにそのデータを生成したという正当性は保証できるが，ある電子署名が別の電子署名より先に作成されたか，といった時間的順序の一貫性は通常保証できない．しかし，取引履歴を管理運用するブロックチェーンでは，ある取引が別の取引より前に行われたか後で行われたかによって，その取引の有効となったり，無効になったりするのでこの時間的順序の一貫性を担保することが非常に重要である．この一貫性を保証するのがハッシュチェーンである．ブロックチェーンにおけるハッシュチェーンの使い方は，特定の取引が承認済みであるか否かの取引検証のプロセスで使われる．検証ノードによって取引の正当性が確認できた取引が記載されたトランザクション・データは，同時期に承認された他の取引のトランザクション・データと共に分散台帳への記録単位である“ブロック”にまとめられる．これら同じブロックにまとめられた取引群は，検証ノードによって“同時に”承認されたとしてブロック全体のデータを入力としたハッシュ値と共に分散台帳に記録される．このとき，分散台帳に記録されるブロック間の時間的な順序関係を保証するために，それぞれのブロックデータは直前に配布されたブロックデータのハッシュ値を含め，ハッシュチェーンを構成するように設計されている．つまり，承認済みの取引群を分散台帳に追記するブロック間が，ハッシュチェーンを構成することで，結果的に承認された取引間の時間的順序の一貫性が保証される．

“Peer-to-Peer ネットワーク”は，略して P2P ネットワークとも呼ばれ，違法性の高いファイル共有ソフトで使われた技術として知られている．Peer という言葉には「対等な立場で情報共有を行う相手」という意味があり，機能や役割が同格な端末“ノード”が 3 台以上参加するネットワークを構成し，情報や機能がノードに分散配置されるシステムである．インターネットでは現在も主流であるクライアント・サーバ型と比較すると以下のような利点があるとされている．

① データの分散管理

P2P ネットワークでは，管理対象のデータはコピーが作成され，複数のノードがそれぞれ分担する形で保存，管理される．データを集中管理するサーバを運用する方式に比べ，

[a] 以下でトランザクション・データを略記する場合には慣例に従い TX データと表記する．

データの保全に関する責任も分散されるため，システム運用の管理負担が相対的に低減される．

② スケーラビリティと耐故障性の改善

P2P ネットワークでは，どのノードにアクセスしても結果的に同じ情報が得られるようになっている．サービス利用時にサーバへのアクセスが必須な方式に比べて，特定のノードへのアクセス集中や，ノードの故障でサービスが中断する事態を回避しやすい．

③ 匿名性の確保

P2P ネットワークでは，データの管理やアクセスがネットワーク上に広く分散されるため，情報やアクセスがサーバに集中する方式に比べ，外部から見て全体像を把握しにくいという特性がある．この特性は，利用者のプライバシを保証するときに便利である．

このようにブロックチェーンに使われている技術要素は，もともと別の用途のために考案され，実際に使われてきた実績がある“成熟した”技術を流用しているため，ブロックチェーンの暗号学的な安全性は非常に強固なものとなっている．

ところで，他からの流用ではないブロックチェーン独自の仕組みは，“コンセンサス・アルゴリズム”を使った取引の非中央集権的な検証にある（図 2-1 参照）．

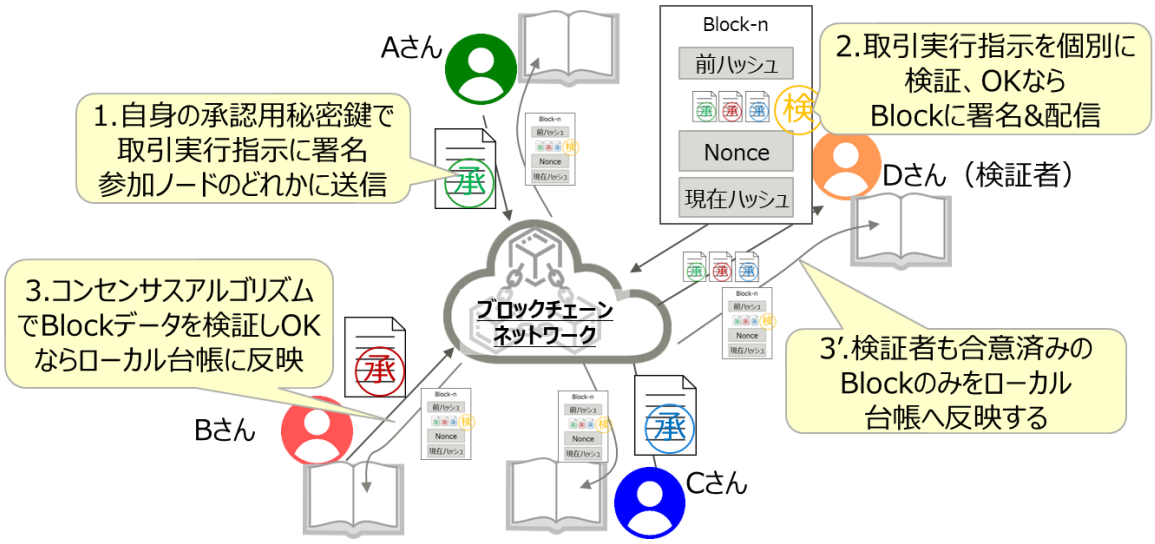


図 2-1：コンセンサス・アルゴリズムによる取引の検証

ブロックチェーン上で取引をしようとするユーザは，自身が発行する取引データ（例えばあるユーザのウォレット口座から別ユーザの口座への送金）に電子署名をしてから，署名済みの取引データをブロックチェーン・ネットワークに参加するノード経由で送信する（図 2-1 のステップ 1 ）．次に，コンセンサス・アルゴリズムの一部として規定された方法（例えば後述する Proof of Work などの方法）で取引の検証者が選択され，選ばれた検証者は取引の正当性を検証する．このとき検証するのは，電子署名だけではなく，参加ノード間で同期している台帳に記録されている取引履歴と不整合がないか検査する．これは，取引につけられた署

名が正しいものであっても，ウォレット口座の残高が送金予定額より小さかったり，二重取引をしていないかをチェックしたり，するためである．検証者が正当と判断した取引はブロックと呼ばれる取引データの集合に取り込まれ，他の参加ノードへ配信される（図 2-1 のステップ２）．そして，そのブロックデータは，次に選ばれた検証者のチェック対象になっており，次の検証者が正しいと認めたブロックデータのみが台帳の同期対象になる（図 2-1 のステップ３とステップ３’）．

ところで，この検証者による取引の検証結果を別の検証者がさらに検証する仕組みがあっても，検証者同士に結託できる余地があると検証の正当性に疑いが生じる．このため，検証者をできるだけ公平に選ぶための仕組みとしてプルーフ・オブ・ワーク(Proof of Work)が考案された．

ビットコインのブロックチェーン・ネットワークにトランザクションが投稿(Commit)されると，複数のトランザクション・データが定期的に承認される処理単位，ブロックにまとめられる． このブロックに含まれるトランザクションを検証し，承認を与えることで報酬が与えられる約束になっている．この報酬を受け取る権利は早いもの勝ちとされているので，検証者（マイナーと呼ばれる）同士がブロック検証のスピードを競い合うことになる．この検証作業に時間がかかるのには理由があり，承認対象となったブロックには複数のトランザクション・データの他に，ナンス(Nonce)と呼ばれる乱数パラメタを指定して，ブロックデータを入力データとして，ハッシュアルゴリズムの一つである SHA-256 で計算したハッシュ・ダイジェスト値，の先頭ビットが一定数の“0(ゼロ)”で埋めつくされたときのみ，その承認結果がブロックチェーン・ネットワークに受け容れられる．このような条件を満たすナンスを効率よく見つける方法は見つかっていないことから，マイナーはナンスの値を１ビットずつ調整してハッシュ・ダイジェストの値が条件を満たすまでハッシュ値の計算を繰り返すことになり，多大な計算力を費やすことになる．一方，ナンスが指定されたブロックのハッシュ値が条件を満たしていることは，一度だけのハッシュ値計算で検証できるので，与えられたナンス値が正当なものであることが即座に判定できる．こうして作られたブロックデータの最後には，この検証への報酬として各トランザクションの取引手数料と新規に発行されたビットコインが合算されて，マイナー所有のウォレット・アドレスへの送金するトランザクションが含まれている．ビットコインの発行量はこの検証作業によってのみ可能なので，検証作業もビットコインの採掘（マイニング）と呼ばれている．

このようにして，ブロックチェーンではユーザと検証者が互いに監視しあうことで非中央集権な形の運営でも安全に取引が実行できている．

2.1.3. スマートコントラクト

スマートコントラクトとは自動的な契約のことであり，契約とその履行条件をあらかじめプログラミングしておくと，契約条件が満たされた際に自動で取引が行われるような仕組み全般を指す．ブロックチェーンでのスマートコントラクトは，ブロックチェーン・ネットワークで動作するプログラムである．

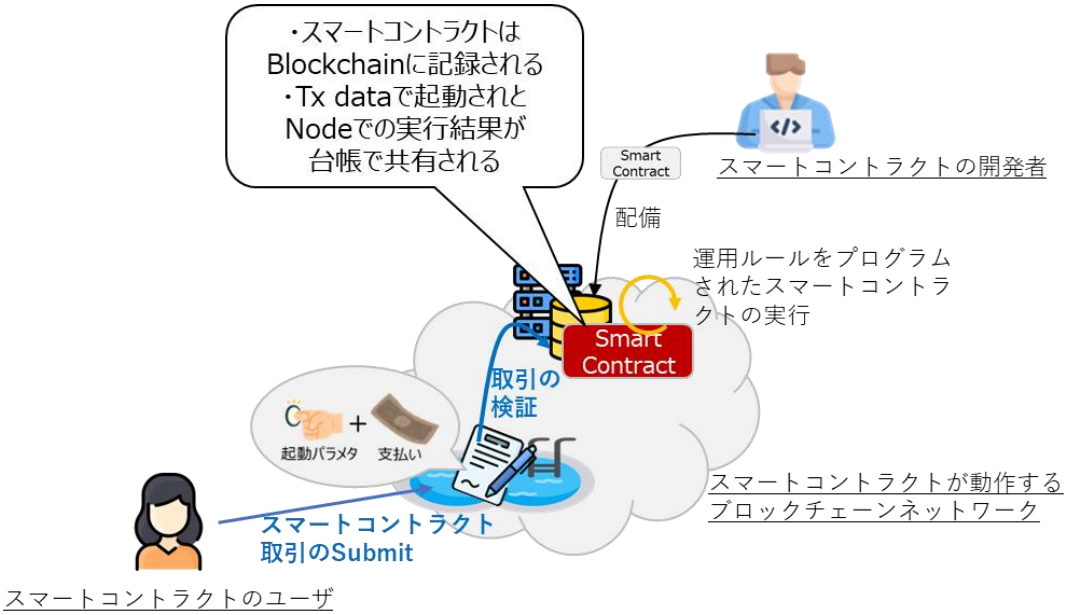


図 2-2：スマートコントラクトの動作

スマートコントラクトの振る舞いを理解するため，イーサリウムでのスマートコントラクト[3]の実装について詳しく説明する．

イーサリウムは，非中央集権型アプリケーションを構築するためのプロトコルとして作られた．イーサリウムには EOA(Externally Owned Account)とコントラクトアカウント(Contract account)という 2 つのアカウントがあり，どちらのアカウントも 20 バイトのアドレスと状態を持っている．EOA は秘密鍵の署名でコントロールされており，暗号通貨 ETH の残高は持つが，コードは持たない．一方，コントラクトアカウントはコードを持っており，このコードがいわゆるスマートコントラクトとして機能する部分となる．スマートコントラクトは，EVM(Ethereum Virtual Machine)と呼ばれる仮想マシン環境上で実行されることで，暗号通貨の送金だけでなくさまざまなオペレーションをイーサリアム・プラットフォーム上で行うことができる．ちなみにコントラクトアカウントはユーザが持つことはできず，EOA によって開始されたトランザクションによってイーサリアム・ネットワーク上で動作する．

スマートコントラクトを利用できるようにするには，まずスマートコントラクトの作成者が Solidity[4]などのプログラミング言語でプログラミングを行う必要がある．そして，そのソースコードを EVM で処理可能なバイトコードにコンパイルしたのち，このコンパイル済みコードを含むトランザクションをイーサリウム・ネットワークに送信し，採掘者によってブ

ロックチェーンに登録してもらって初めて他のユーザからアクセス出来るようになる．EOA の発行するトランザクションによって開始されたスマートコントラクトは、EVM 上で Gas（ガス）と呼ばれる手数料を消費しながら動作する．そして起動時に設定されるガスの消費上限（Gas Limit）を超えると処理の途中であっても停止してしまう．このためスマートコントラクトを実行するためには，Gas Limit を適切に設定しなければならないが、使われずに残った分は送信元の EOA に返金されるようになっている．ガス消費の仕組みは，スマートコントラクトを稼働させるための手数料徴収の仕組みでもあるが、プログラムが無限ループに陥らないための仕組みでもある．

イーサリウム・ネットワークで共有されている分散台帳では，アカウントステート(Account state)とワールドステート(World state)という二種類の状態が，新たなトランザクションの生成ごとに変化していき，その変化を効率よく管理するため前述のマークルツリーを改良した，マークルパトリシアツリー[５]というデータ構造が用いられている．スマートコントラクトの実行結果は，これらのステート情報として分散台帳に記録，保持されるため，次にそのスマートコントラクトがイーサリウム・ネットワークのどこから呼び出されても，直前の処理結果を反映した振る舞いが保証されるようになっている．

スマートコントラクトはこれまで第三者ユーザの仲介者を必要としていた取引プロセスを自動化できることから，その導入で決済期間の短縮や不正防止，システムの運用コスト削減といったメリットが期待できる．

2.1.4. ブロックチェーンの特長

ブロックチェーンは，技術的には昔からあり有用性が認知されている枯れた技術の集合体であると説明したが，システムにもたらす付加価値としての導入効果はとてもユニークなものになっている（図 2-3）．

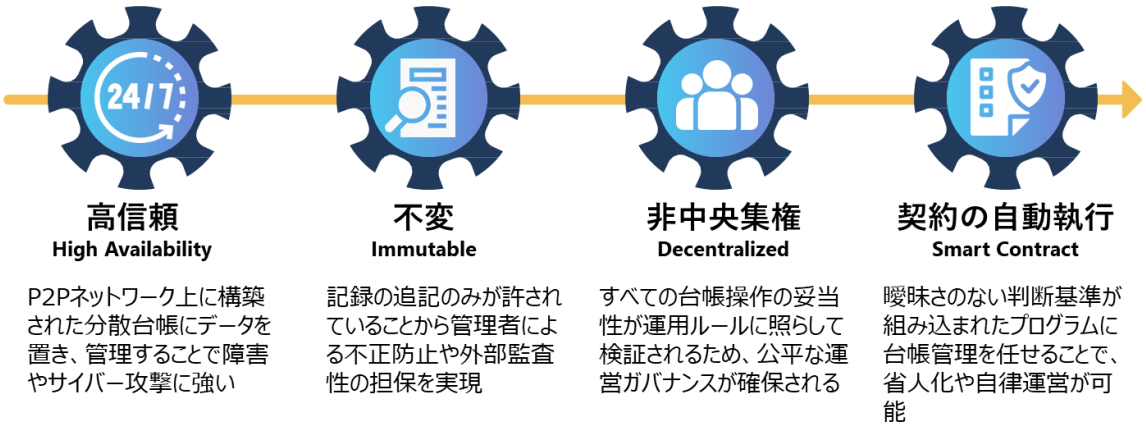


図 2-3：ブロックチェーンの特長

高信頼

ブロックチェーンはこれまで主流だった中央集権型システムとは異なり，それぞれが同等の機能・権限を持つ複数のノードが参加する P2P ネットワークで構成されている．このため特定のノードがダウンしても機能不全に陥る心配がなく，システムの安定的な運営が可能となっている．

また、各ノードは過去の取引データの複製を保持しており，機器の故障やサイバー攻撃などで特定のノード上のデータベースが消失または改ざんされた場合でも，他のノード群が管理する取引データを検証したうえで取り込むことで，安全に復元できるようになっているので，ブロックチェーンで管理するデータベース（＝分散台帳）は消失する心配がない．

不変

不変である(Immutable)の本質は外部からの変更を受け付けないということである.計算機科学の定義では，不変オブジェクト(Immutable Object)とは，オブジェクトが作成された後にその状態を変更できないオブジェクトとされており，ブロックチェーンに記録される台帳がデータの上書きではなく，データの追記のみが可能なためこの特性が実現される．

データの上書きを許している通常のデータベースでは，現在の値がもとからその値であったのか，改ざんされたものかを簡単には判定できない．一方，ブロックチェーンでは台帳上で管理されるある値が変化する場合でも，値を変化させる原因となった取引を時系列で記録され，台帳の記録内容に監査が必要になったときにも検証にかかる時間とコストが削減できる．

またブロックチェーンでは，たとえサービス提供者であっても記録されたデータの改ざんや消去はできないし，参加者が自身の取引履歴を消すこともできない．この点がブロックチェーンとデータベースの最大の差となっていて，信用度の低い無名のサービス提供者であっても，暗号通貨の管理などの重要な業務を任せることができる根拠になる．

非中央集権

現行の IT システムのアーキテクチャは「集中型」「分散型」「非中央集権型」のどれかに分類できるので、非中央集権型というシステム運営は，けっして新しい概念ではない．ブロックチェーン・ネットワーク自体は，分散システムに分類される P2P ネットワークを使っているので，その上に構築されるアプリケーションのレベルで非中央集権型の運営が実現されている．

非中央集権型のシステムは，分散型のシステムと混同されることが多いが，分散型システムでは処理性能の向上に使えるのに対して，非中央集権型のシステムではスループットの低下などの弱点がある．しかしながら，ハッシュチェーンとコンセンサス・アルゴリズムの組み合わせにより，すべての台帳操作の妥当性が運用ルールに沿って照らして検証されるため，集中型や分散型では実現できない公平な運営ガバナンスを担保するアーキテクチャとして特異な価値を提供できる．

契約の自動執行

スマートコントラクトがコンピュータの世界でブロックチェーン上の取引を効率化することはすでに説明しているが，現実世界での“契約の自動執行”への応用がブロックチェーンのスマートコントラクトと相性が良いことから，近年注目を集めている．

スマートコントラクトは狭義には「設定されたルールに従い、ネットワーク上で取引を自動的に実行する仕組み」と言い換えることができる．この定義を拡大解釈するとブロックチェーンに契約の成立条件を設定しておき，その条件が満たされたときのみに契約で合意された処理を実行する，といったこともスマートコントラクトの機能となる．

従来の集中型 IT システムでも同様の処理が実行できる場合もあるが，その適用範囲は処理システム自体が契約の当事者としての役割を果たしている場合に限られ，当然契約の履行に対しての中立性は担保されない．一方，スマートコントラクトで自動化された契約では，曖昧さのない判断基準を規定出来るだけなく，合意した契約内容の改ざんや，管理者や当事者ユーザの不正操作が発生する心配もなくなる．

2.2. コンソーシアム型ブロックチェーンの登場と Hyperledger Foundation

ブロックチェーンの基本形について解説したが，ビットコインなどの暗号通貨の管理に特化したブロックチェーン・ネットワークは現実世界の問題を解決するには，不利なアーキテクチャになっていた．ビットコインなどから派生したコンソーシアムチェーンの生い立ちと，その特性について説明する．

2.2.1. ブロックチェーンの分類

ブロックチェーンと一口に言ってもアーキテクチャレベルで特性が違う種類が存在する．

ビットコインやイーサリウムなどの暗号通貨のためのブロックチェーン・ネットワークは，ノード運営者に支払われるマイニング報酬をインセンティブに共同運営を実現させている．しかし，暗号通貨の市場価格上昇に伴い，暗号通貨の送金やスマートコントラクトの呼出しなどでブロックチェーン・ネットワークを利用する際の取引手数料も上昇したことで「安価で利用できる」という利点が失われてしまった．さらに，ブロックチェーンの使途が暗号通貨から，より少額で取引されるデジタル資産の管理や，それ自体は資産価値を持たないデータの記録などにブロックチェーンの用途が拡がってくると，非中央集権なシステムであってもガバナンスや運用コストの問題で，暗号通貨の管理に特化した形で設計されていた旧来のブロックチェーン・ネットワークでの運用が適さないケースがでてきた．

このような市場ニーズを受ける形で，アーキテクチャレベル特性が異なるブロックチェーンの亜種が登場した．ビットコインやイーサリムは正統なブロックチェーンとしてパブリックチェーンと呼び、そこから派生したブロックチェーンにプライベートチェーンやコンソーシアム（共同事業体）チェーンと呼ばれている．

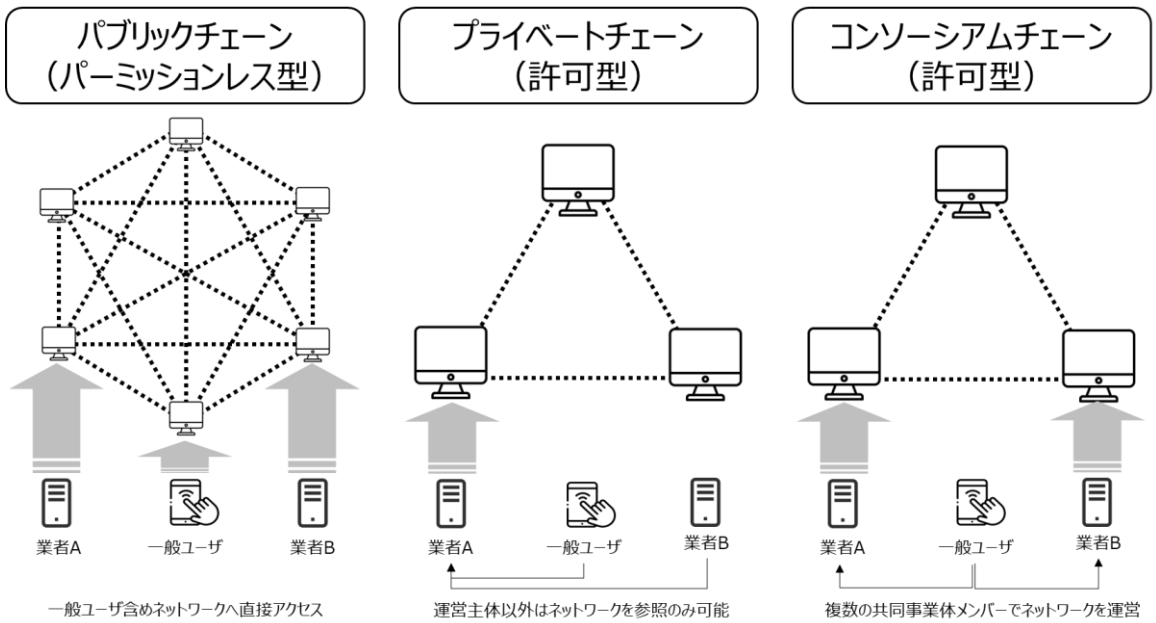


図 2-4：ブロックチェーンの分類とブロックチェーン・ネットワーク運営の違い

また，これら派生ブロックチェーンが持つ別の側面を捉えた分類として，パーミッションレス・チェーンとパーミッションド・チェーンの区別がある．パブリックチェーンが取引に対して台帳操作への権限がすべての参加者に同等に与えられることからパーミッションレス型チェーンと呼ばれるのに対して，プライベートチェーンやコンソーシアムチェーンでは通常アクセス制御が適用されていて，事前に許可を受けた特定ユーザのみが台帳を操作できることから，許可型チェーンと呼ばれる．

ブロックチェーンの分類と，分類した種類ごとのブロックチェーン・ネットワーク運営の違いについてのまとめを図 2-4 に示す．

パブリックチェーンには誰でも参加でき，ブロックチェーン上のデータの読み取り，書き込み，または監査を行うことができる．この特性は一見危うげに見えるが，台帳への書き込まれるデータは，その都度過半数の参加者が同意する運営ルールに従い，参加者自身によって運営ルールの検証を担保するコンセンサス・アルゴリズムの適用により，ルール違反をしようとする書き込みは，確実に排除されるようになっている．その一方で，コンセンサス・アルゴリズムによるデータ検証を確実に行うためには，データ検証を完全に独立した数多くのノードで行う必要があるため，データを処理する能力は数十 tps(Transactions Per Second)に限定される．

一方，プライベートチェーンやコンソーシアムチェーンでは，ブロックチェーンの持つ運営や記録データの透明性は継承しながらも，検証に関わる処理を特定の信頼に足る組織またはグループに委ねて記録データの検証に関する処理時間を短縮することで，数百から数千 tps と，パブリックチェーンと比べて非常に高いデータ処理能力を得ることができる．

2.2.2. ブロックチェーン関連 OSS 開発プロジェクト Hyperledger Foundation

プライベートチェーンやコンソーシアムチェーンの運営は，当初パブリックチェーンのソフトウェア実装をそのまま流用して行われていたが，処理性能の向上やアクセス制御の強化など，許可型チェーンのニーズに特化した専用ソフトウェア（ブロックチェーン基盤ソフト）が開発され，企業を中心に利用されるようになっていった.

しかし，ブロックチェーンの利用目的にはデジタル資産を非中央集権な形で管理することであるので，そのブロックチェーンを構築するための基盤ソフトウェアについても，動作ロジックの把握による透明性確保や，ソフトウェアにバグが発見された場合の修正作業など，インフラ維持に必要なメンテナンス作業への参加にオープン性が不可欠だと考える技術者が多かった.

このような動きを受け，有志企業からの資金援助を受けてブロックチェーン基盤ソフトをOSS(オープンソース)として共同開発するための組織として，“Hyperledger[b]“が 2016 年 2 月に設立された．オープンソースとは，ソフトウェアの動作を指示する設計図にあたる，ソースコードが公開されており，公開条件の則った改変や再利用が許されるソフトウェアのことである．また，Hyperledger は Linux Foundation の傘下にあり，100 以上の企業・団体からの開発者が参加するブロックチェーン関連のオープンソース開発者コミュニティとして最大の非営利団体となっている.

2.2.3. コンソーシアム型ブロックチェーン基盤 Hyperledger Fabric

Hyperledger では，設計思想や得意分野が異なる複数のブロックチェーン基盤ソフトが並行して開発されているが，そのなかでも高い処理性能と汎用性を備えることで知られる Hyperledger Fabric は，コンソーシアム型ブロックチェーンの代表格とみなされ，数多くの企業や学術機関で利用されている.

Hyperledger Fabric のブロックチェーン・ネットワークは許可型コンソーシアムチェーンで，メンバーシップサービスによって許可されたノードとユーザのみがアクセスできるようになっている．メンバーシップサービスは，公開鍵暗号基盤(PKI: Public Key Infrastructure)の上に構築されていて，コンソーシアムを構成する組織がそれぞれ運用する認証局（CA: Certification Authority）が発行する電子証明書で操作主体のアイデンティティを認証し，台帳操作で許可されるべき操作権限を決定している.

常時オンラインとなっているネットワーク要素は，分散台帳のステート（ワールドステート）を管理していてユーザからの取引依頼を受け付ける Peer ノードと，取引依頼を検証後に確定した取引結果をステートに反映させるための台帳操作をブロックデータとして Peer ノードに配布する Orderer ノードとで構成される．Peer と Orderer はネットワークに参加する組織が提供するもので，それぞれ最低 1 ノードを必要とする．また，Hyperledger Fabric のネットワークは，さらに細かいアクセス制御単位としてチャネル(Channel)が設定されており，

[b] 2021 年に“Hyperledger Foundation”と改称．以下では単に Hyperledger と表記.

分散台帳のステートの内容がチャネルごとに管理されるので，互いにプライバシが確保された独立したブロックチェーン・ネットワークとして運用できる．

そしてユーザは，認証と取引発行のプロトコルを実行ための専用クライアントから，特定のチャネル ID で識別されるブロックチェーン・ネットワークに参加する Peer ノードのひとつを介してネットワークにアクセスするので，常時オンラインでなくともブロックチェーン・ネットワークに参加、および利用できるように設計されている．

Hyperledger Fabric のブロックチェーン・ネットワークでの台帳操作は，chaincode と呼ばれるスマートコントラクトを介して行われる．この chaincode による台帳操作（取引）は，クライアント（fabric-sdk とも呼ばれる）と Peer、Orderer 協調動作することで，分散台帳を操作する取引の正当性検証を行うコンセンサス・アルゴリズムを形成しているので，その動作を詳しく説明しておくことにする．

Chaincode の配備(deploy)

Hyperledger Fabric での分散台帳の実体にあたるワールドステートは，キーと値(value)の組でデータを保持する，いわゆる key-value ストアで管理されている．そして，台帳操作の検証を行う chaincode には，台帳操作の命令セットを動作ロジックに従って実行するプログラムと，検証に参加すべき Peer のエンドーズメントに関する取り決め(endorsement policy)が含まれていて，管理者によって指定されたチャンネルに参加中のノードに配布され，配備が許可されるとユーザからの呼出しが可能になる．ここではチャネルに参加する Peer-A と Peer-B の両方が“合意価格による代金が支払われた”という動作ロジックの検証に成功したときだけ，当該の取引処理（Hyperledger Fabric ではトランザクションと呼ぶ）を有効とする endorsement policy が設定された“ラディッシュ購入アプリ”が配備されたと仮定した．

Chaincode の実行（トランザクションの開始）

ユーザがラディッシュ購入アプリを利用するときには，chaincode を呼び出し，台帳を操作するトランザクションを開始するメッセージ“Endorse Proposal”を fabric-sdk の機能を使って作成し，ユーザの電子署名を付与してからネットワークに参加する(Endorsing) Peer，今回は Peer-A と Peer-B，に送信する．

Endorse Proposal を受け取った Peer は，Proposal の正当性をチェック（データ構造の正しさ，リプレイ攻撃がないこと，電子署名が有効であること）を行ったのちリクエストされた chaincode の実行を行う．Endorsement プロセスでの chaincode の実行でエラーが発生せずに処理が完了した場合には，Read-Write-Set と呼ばれる差分データを生成する．Read-Write-Set は chaincode の動作ロジックを Peer が保持するワールドステートに対して適用した場合のシミュレーションを行った場合に，値を取り出す指示や値の更新が行われる（予定の）key-value ペアを差分データである．Read-Write-Setの生成に成功した場合には，これに Peer の電子署名をつけた “Proposal Response”が Proposal を送信してきた fabric-sdk へと返答される．

Endorsement Response の調査

Proposal Response を受け取った fabric-sdk では，付与されている署名が有効であるかチェックしてから Read-Write-Set が Endorsing Peer 間で一致していることを確認する．

Endorse Proposal が，台帳の更新を必要としない（つまり Write-Set が含まれない）場合には，ここで通信処理が完了し，問い合わせ結果を fabric-sdk を呼び出しているアプリケーションに返す.

トランザクション・メッセージの組み立て

Proposal が台帳の更新を意図していた場合，fabric-sdk は Endorsement policy で定められた条件を満たすまで（この例では Peer-A と Peer-B の両方のエンドーズメント署名を揃えることが要求されている）Endorsing Response を集め，Endorsement policy の条件が満たされたと判断すると，Transaction Proposal とその応答（具体的には Read-Write-Set）を組み込んだトランザクション・メッセージを生成し，Peer 経由でコンソーシアム参加組織の Orderer ノードで構成される Ordering サービスに向けてブロードキャストで通知する.

ブロックの生成と台帳の更新

Ordering サービスでは，fabric-sdk から通知される複数のトランザクション・メッセージをチャンネル毎に分別収集し、時系列で順序付けしてチャンネル毎のブロックにまとめたのち，ブロックデータを当該チャネルに参加する Peer ノード群に配布する．

ブロックデータを受信した Peer ではブロックデータが正当であることを検証したのち，それまでに受信したブロックで維持されてきたワールドステートの値を新たに受信したブロックで更新する.

Hyperledger Fabric におけるトランザクション処理のまとめ

上記で説明したトランザクション処理の流れをまとめると図 2-5 のようになる.

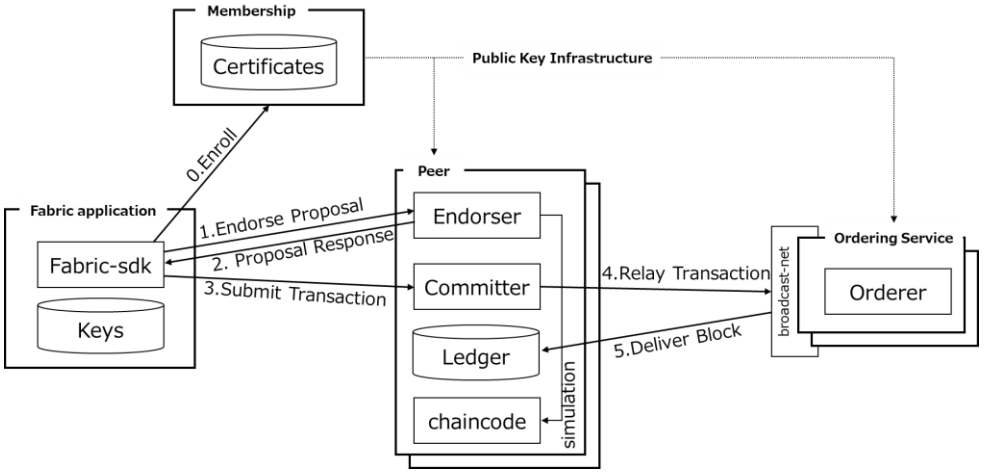


図 2-5：Hyperledger Fabric でのトランザクション処理の流れ

2.2.4. インターオペラビリティ問題と Hyperledger Cactus

スマートコントラクトを備えたイーサリウムのユーザは急激に増えている．あるユーザが保有するイーサリウムの残高の一部を，別のユーザに移す“送金”は美しい GUI を備えたウォレット・アプリ「メタマスク (MetaMask[6])」を使うととても簡単に実行できる．しかし，メタマスクは，ユーザの残高の一部をビットコインのアドレスへ送金することはできない．なぜなら，各ブロックチェーン・ネットワークは，独自の運用ポリシーと通信仕様に基づいて構成され，情報や資産価値を共有することができない．このように似たような暗号通貨を扱っていながらも相互に接続してされていないので，それぞれのブロックチェーン・ネットワークに参加しているユーザ間のコラボレーションが進んでいない．これをブロックチェーンにおけるインターオペラビリティ問題と呼んでいる．

Hyperledger では，組織が発足した当初からブロックチェーン技術にある様々な欠点を同時に解決することが理論上不可能であると考え，ブロックチェーン・ネットワークを構築するための基盤ソフトを互いに競合する可能性を排除せずに複数同時に開発する決断を下した．そして利用目的に合わせて特定の課題解決を図るため，アーキテクチャの異なる基盤ソフト 5 種類を同時に開発することになった．また，Hyperledger の組織外でもコンソーシアムチェーンの市場ニーズを汲み取る形で，CORDA や Quorum といった Hyperledger 以外で開発されたオープンソースの基盤ソフトも登場することになった．これらの企業向けブロックチェーン基盤ソフトの活用により，パブリックチェーンでは機能の欠如や処理性能不足などの問題で実現が難しかった分散台帳を活用した新しいサービスが次々と生まれた一方で，ブロックチェーン基盤ソフトが多様化したことによる弊害も生じている．ブロックチェーン・ネットワークで管理されている資産に対する信用は，そのブロックチェーンに参加しているメンバー間でのみ共有されることになるが，デジタル資産を売買するようなシーンで，ブロックチェーンの数が増えてくると取引相手が同じブロックチェーンに参加していないため取引が成立しないことが頻繁に発生するようになる．Hyperledger には，公式のブロックチェーン基盤が複数あることから，この問題が早い時期から認識された．そこで，この問題「ブロックチェーンのインターオペラビリティ」を解決するツールを開発するプロジェクト“Hyperledger Cactus”が 2020 年 5 月に設立された．Hyperledger Cactus は，ブロックチェーン・ネットワークへのアクセスを抽象化することで，10 種類以上の異なるブロックチェーン基盤ソフトで構築・運用される分散台帳同士を統合したサービスを構築できるようになっている．

2.3. トークン・エコノミーの現状

ブロックチェーンがビットコインなどのパブリックチェーンから，アーキテクチャレベルで違いのある，コンソーシアムチェーンがその派生形として生み出した．

種類の増えたブロックチェーンの新たな活用方法として，無形や有形の資産価値をスマートコントラクトでデジタル化し，仲介者を介することなくユーザ間で直接その資産価値を取引できるようにする，「トークン化」が注目されている．

2.3.1. ブロックチェーン技術による資産価値のトークン化

トークン化（tokenization）とは，物理的な資産や仮想的な資産を，売買可能なデジタル単位に変換することで，トークン化により地域的な障壁や仲介者を排除し，資産を細かく分割できるとされている[７]．トークンには用途や性質が異なるさまざまなトークンがあるが，そのトークンが別のものに代替可能かどうか（ファンジビリティ）という観点では FT と NFT に分類できる（表 2-1 ）．

表 2-1：ファンジビリティによるトークンの分類

分類	定義	例
FT (Fungible Token)	ある資産を同じ種類かつ同じ価値をもつ別の資産と交換できるトークン．	法定通貨，暗号通貨（仮想通貨）
NFT (Non-Fungible Token)	固有の価値を持ち，分割できず，相互交換もできない資産をトークン化したもの．代替不可トークンと呼ばれることが多い．	デジタル・アートの所有権，物理的な資産の所有権

ここで注目すべきなのは NFT である．NFT の特長として「ブロックチェーンが持つ非中央集権や透明性，トレーサビリティ，関係者間の直接的な情報共有および管理，対改ざん性といった技術特性を備えていることに加え，固有性，取引可能性，相互運用性，プログラマビリティといった特性も併せ持っている[８]」が挙げられている．現状ではこの特性の一部のみを使った NFT アートのような希少価値の高い“権利”の売買取引をオンライン完結な形でユーザ間が直接行えるようにする仕組みとしてのみ利用されているが，トレーサビリティやプログラマビリティなどは潜在的価値のままで活用ができていない．このトークンの潜在価値を引き出すことが，従来は難しかった“トークン・エコノミー”の実現につながると考えられている．

この分類に大きな意味を持たせているのが， ERC-20 や ERC-721 などのトークン操作用 API の標準化の動きである．ERC(Ethereum Request for Comment)は，イーサリウムの開発者コミュニティが規定する技術標準で，ERC-20 は，スマートコントラクトを使ってトークンを実装するために使用される技術標準で，基本トークン用の単純なインタフェースが規定されている． 一方，ERC-721 は ERC-20 を拡張した技術標準で，ERC-20 と似た操作感を維持しながら代替不可トークンの特性を活かすため，トークンを一意に識別するトークン ID を指定してトークンを操作するインタフェースを規定している．これらのトークン操作用 API を備えたスマートコントラクトでは，資産価値がマイニングに必要な計算量に裏付けられた形で生み出される暗号通貨（ネイティブ・トークン）とは異なり，Fungible Token と Non-Fungible Token に分類されるトークン（私製トークンとも呼ばれる）を管理する．私製トー

クンの世界では，その資産価値を保証する「トークン発行体」の社会的な信用が資産価値の裏付けをしており，トークンの利用や換金に関して管理者としての役割と権限を持つことができる．トークン管理用のスマートコントラクトに Fungible Token では ERC-20，Non-Fungible Token では ERC-721，に準拠したトークン管理用スマートコントラクトを使えば，トークン発行体によるトークン発行や破棄（Burning）などの特権的な操作も，すべてブロックチェーン上での監視対象となる，標準 API の呼出しを通してだけ可能となるので，トークン管理における運営ガバナンスへの透明性を担保することが可能となっている (図 2-6 参照)．

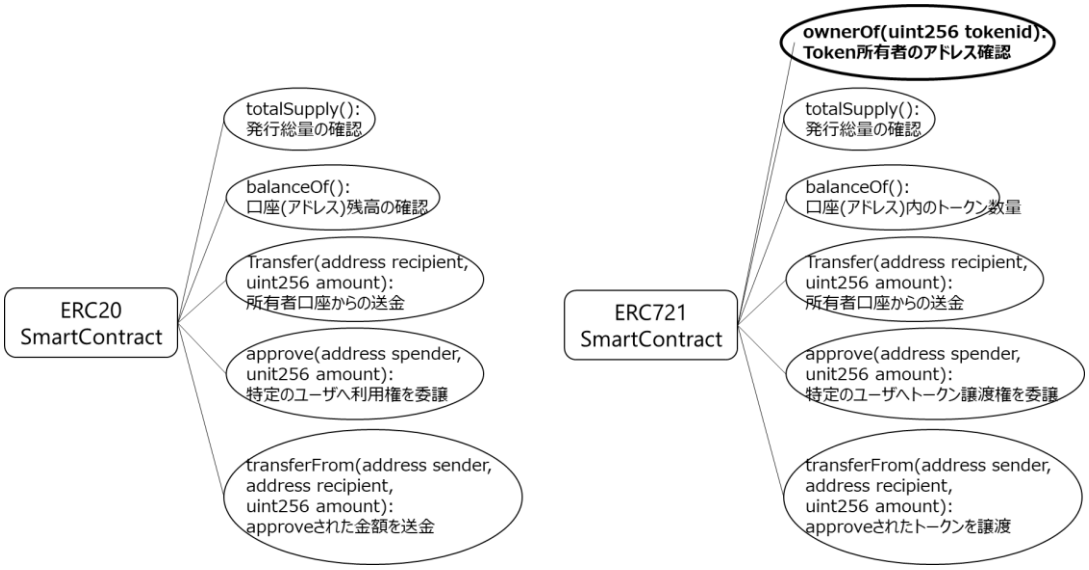


図 2-6：ERC20(数量型資産用)と ERC721（代替不可資産用）の操作用 API

具体的には，ネイティブ・トークンである ETH がマイナーによるマイニング作業で少量ずつしか発行できないのに対して，トークン発行体が ERC-20 や ERC-721 に準拠したトークン管理用スマートコントラクトを使えば，トークン発行体はユーザからの信用を担保にいつでも好きな量のトークンを発行できるのである．

また， ERC-20 や ERC-721 準拠のスマートコントラクトを使うメリットは他にもある．*approve()*と *transferFrom()*の API の組み合わせによる，権利移転代行機能の実現である．

ユーザ間で対価を要求する直接取引をブロックチェーンの世界で行うのは実は簡単ではない．まず，暗号通貨の送金とは違い，デジタル資産の売買では売り手と買い手の間でそれぞれが所有する 2 種類のトークンを互いに同時に送金しあう，いわゆる同時交換取引（金融業界の用語では DvP 取引: Delivery versus Payment と呼ぶ）が必要になるからである．DvP 取引は，売り手による代金の持ち逃げや，買い手による商品の持ち去りを防ぐ役割があるが，ブロックチェーン台帳の操作には DvP 機能がないので、現実的な解決法として取引を仲介する取引所を利用することになる．これを容易にするのが権利移転代行機能である．図 2-7 に示した ERC-721 準拠のスマートコントラクトを使った NFT 譲渡のフローでは、Step 1 で売り手の Alice が取引を仲介する NFT 販売所 Charlie に権利移転の権限を委譲し，Step4 で Charlie が権利移転の執行を行っている．

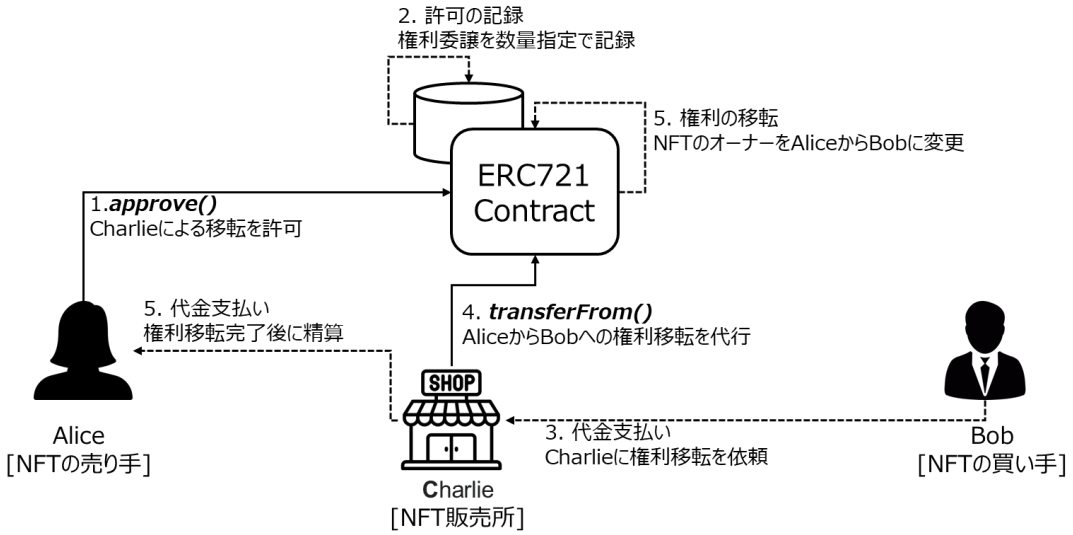


図 2-7：ERC721 を使った権利移転代行の例

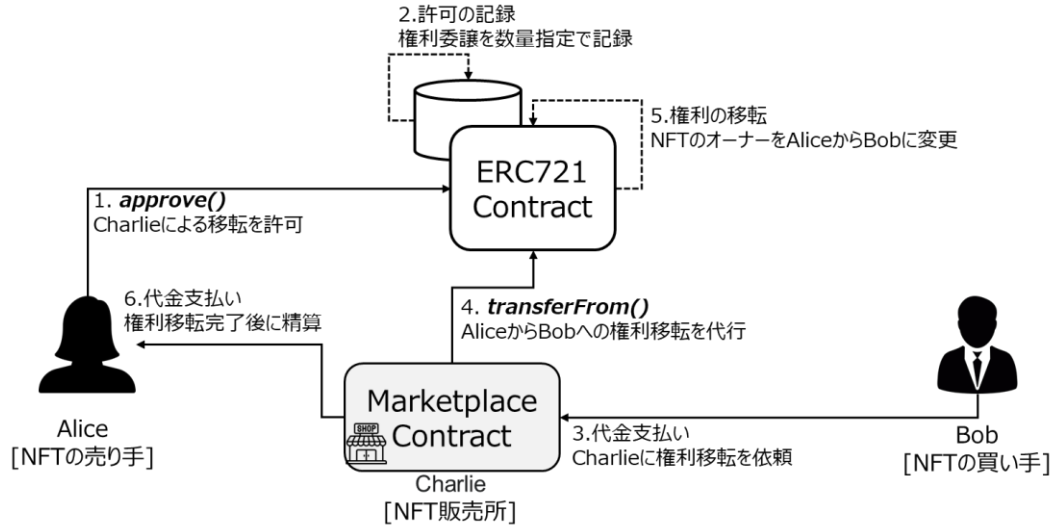


図 2-8：仲介業務のスマートコントラクト化

ここでは，Charlie がユーザである場合の例を示したが，スマートコントラクトは別のスマートコントラクトを呼び出す機能を持っているので，

図 2-8 のように、ユーザ Charlie が担っていた仲介者の役割をスマートコントラクトで置き換えてユーザの介在を減らすことができる．

代金支払いに ERC-20 標準に準拠した決済トークンを使えば，スマートコントラクトによる完全な自動化でユーザによる介在を排除した安全な取引が可能となる(図 2-9).

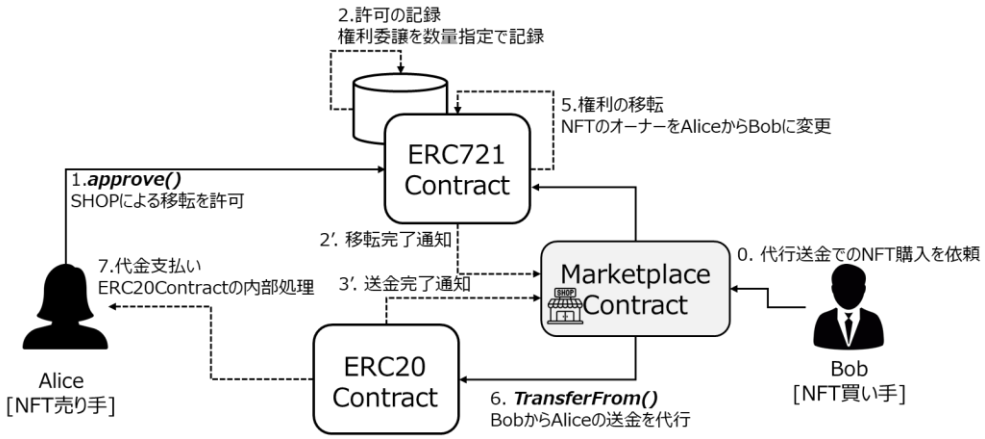


図 2-9：スマートコントラクトによる NFT 交換所の完全自動化

この仕組みを実際に利用しているのが NFT マーケットプレイス・サービス OpenSea であり取引に仲介者を挟まない直接取引による NFT の無人販売が可能となっている (図 2-10).

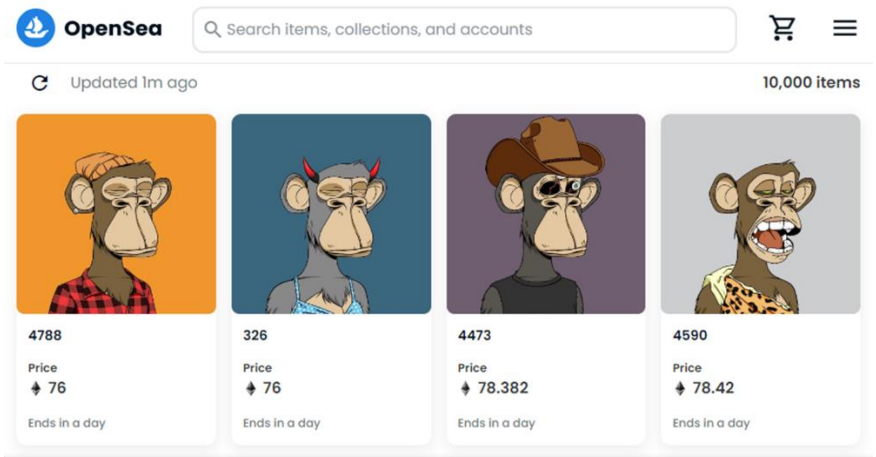


図 2-10：OpenSea での NFT 販売 (https://opensea.io/collection/boredapeyachtclub)

2.3.2. トークンの用途による分類

トークン・エコノミーとは，「トークンを用いた価値のエコシステムで，ブロックチェーンを基盤として個人や法人が資産をデジタル化して運用することにより，多くの人や企業がその資産価値を新たに認識し，活用や拡張，交換などの循環が活発化し生まれる豊かな経済圏[９]」，とされている．実のところ，暗号通貨の交換取引も広い意味ではトークン・エコノミーの一種である．

ファンジビリティとは別の視点として，トークン・エコノミーでの活用が期待されるさまざまなトークンを利用目的で分類したのが表 2-2 である．

表 2-2：トークンの分類

種別	目的	固有の価値
暗号通貨 (Cryptocurrency)	中央銀行とは独立して運営されており，製品やサービスとの交換の媒体として機能する.	なし - 取引市場での需給バランスで価値が決定される.
資産担保トークン (Asset-backed Token)	資産担保型トークンは，物理的な資産の所有権を表す.	リアル世界の原資産に基づいて，その価値を導き出すトークン．Fungible Token でも資産価値が法定通貨に基づくものはステーブルコインとよびこの一種.
ユーティリティトークン (Utility Token)	ユーティリティトークンは，ユーザに製品やサービスへアクセスする権利を表す	価値は，発行者が提供する製品やサービスに対する需要から派生する.
セキュリティトークン (Security Token)	出資先の法人に対する経済的な利害関係を証明する．トークンの保有者は，現金または別の金融資産を受け取ったり，議決権を行使したりできる.	価値は出資先の企業のビジネス上の成功から生まれる.

一般的にデジタルデータはコピーや改ざんが容易なため，現実の資産や販売物と比較して安全な相対取引が難しい．ブロックチェーンを使ってトークン化された資産は，暗号技術で取引対象となる資産の特定と，その所有権の検証を管理者不在で検証できるので，資産取引をデジタル空間で完結させることができる．また，取引相手との間で交換する資産の等価性に合意できれば「(独自の取引市場がある）金（ゴールド）で株式を買う，といったイメージのユーザ体験に近く，現金を使わずユーザ間での直接取引を行うことが多いので，物々交換のような形になる[10]」のように法定通貨に縛られない個々人が持つ価づけでの相対取引が可能となる．既存の商取引の決済に通貨が用いられるのは，通貨に何にでも交換できて誰もが欲しがる価値があり，いわゆる「欲望の二重の一致[c]」を不要にする機能が不可欠であったためである.「しかし，インターネットの普及で，今や二重の一致は奇跡ではなくなった．今後スマホ決済などによって売り手と買い手がインターネットで結ばれていけば，やがて物々交換や知識・労働の提供など，お金を用いない取引が拡大していくだろう[11]」といった予測もありトークン・エコノミーの実現により，金銭的価値では測りにくい感謝の気持ちを伝えたり，個人間での助け合いを促進したり，といった社会貢献にも期待が寄せられている.

2.3.3. トークン・エコノミーの事例

トークン化は，スマートコントラクトの助けを借りて，さまざまな価値を取引可能な単位

[c] 互いに自分が欲しいものを持っていないと物々交換が成立しない，という問題

である，トークンに変換する取り組みである．トークンは代替通貨と訳されることもあるが，

2.3.3.1. 不動産トークンの運営

トークン・エコノミーの事例として既に一定の成功を収めている取り組みが，セキュリティトークンを発行することで投資を募る STO(Security Token Offering)がある．

レポート「REAL ESTATE TOKENIZATION [12]」では，香港やシンガポールでの不動産にまつわる業務や取引の実例を参考に，不動産のセキュリティトークン化による業務効率化の可能性について論じている．

レポートでは，実際に行われた不動産に関するセキュリティトークン化から得られた知見を一般化し，５フェーズからなるセキュリティトークンのライフサイクルで提案している：

① 運用ルールの策定
② セキュリティトークン化
③ 投資家の募集
④ 投資プロジェクトの運営（不動産管理業務）
⑤ トークンの二次流通

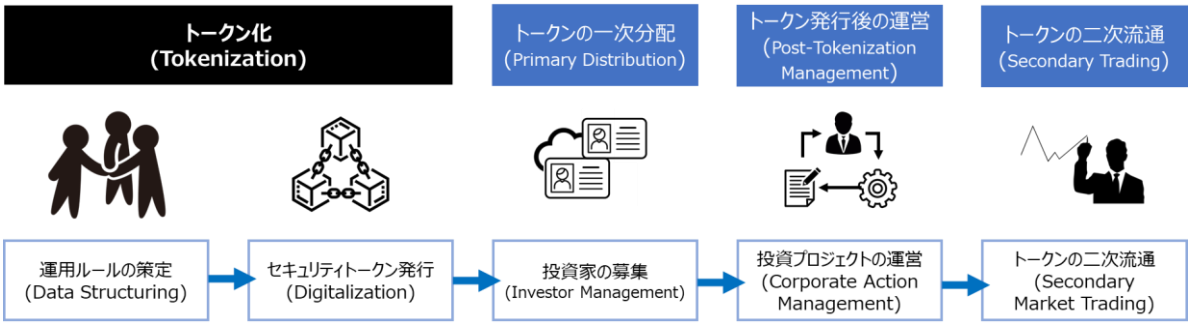


図 2-11：セキュリティトークンのライフサイクル（参考：[12]）

①運用ルールの策定

まずセキュリティトークンの運用ルールを契約条件として決定する必要がある．不動産セキュリティトークンの運用ルールには，不動産に関わる法規制へ準拠するものが多く含まれることになるはずである．これは，トークン化の技術が関連法規制への対応を不要にする技術ではなく，むしろ不可避の関連法規制への対応をいかに容易にするかに活用されるからである．

②セキュリティトークン化

元来，定型もしくは不定形の書類に記録されていた不動産に関する管理情報をイミュータブルな記録（報告書内では造語“Digital ROM”）としてブロックチェーン上にアップロードする処理．この情報は不動産管理業務をアシストする専用のスマートコントラクトが扱い，このスマートコントラクトがセキュリティトークンの実体となる．

レポートではセキュリティトークン化フェーズにおける業務効率化のカギとして，KYC や

AML 対策の自動化とスマートコントラクトによる即時決済の実現が業務効率化のカギになると提言している（原文："Smart contracts form the building blocks of programmable actions, which is the key to unlocking liquidity. Smart contracts are coded to execute compliance protocols, due diligence, KYC, and anti-money laundering (AML) procedures, as defined by regulatory requirements and further specified by the terms set by individual issuers. Smart contracts also play a role in facilitating near-instant settlement of transactions. "）.

③投資家の募集

レポートでは，不動産セキュリティトークンの潜在的価値を引き出すため，簡単には売買できないような高額物件への投資プロジェクトをセキュリティトークン化で効率化することを提案している．この分割所有(Fractal Ownership)のアプローチは，不動産業界では実践されている手法で，不動産投資信託(REITs: real estate investment trusts)として知られている．

『不動産は一般的に価格変動リスクが比較的小さい上，収益が安定して見込めるため，本来は優秀な投資対象である．投資家にはなかなか手が出しづらかった理由としては，投資家にとって魅力的な投資対象であっても物件価格が高いため購入がしにくいことが挙げられる．不動産所有者にとっても，売却、賃貸などでの資金調達では，所有権を手放す，あるいは使用権を失うため，不動産は流動性が低く，資金調達手段として難しい側面があるといえる．この難点を解消するスキームが REIT（リート）を含む不動産の証券化である．まず不動産を売却するため，合同会社や特定目的会社などの形式で事業体を作る．事業体は不動産を運用し，出資した投資家に，投資額に応じて家賃収入や運用益を分配するのだ．事業体が募る出資額は小口化されていることから出資しやすく，投資家にとって魅力的な投資対象となる．証券化された不動産は言うまでもなく流動性が高く，投資対象としてのデメリットは小さくなる．（出典：[13]）』.

レポートでは，投資プロジェクトのセキュリティトークン化することで，以下のような改善が見込めるとしている：

- 出資金額を厳格に分割したセキュリティトークン化で，投資プロジェクトへの参加や脱退が投資家自身の操作でトークンの移転操作で即座に容易に実現できる
- 証券の所有権が分散台帳で可視化されるので信用リスクの不安が払しょくされ，直接取引が可能となるため，これまで必須だった中間手数料を抑制
- 非常に高額な不動産物件１件にしか適用できなかった REIT を，複数の比較的安価な物件をまとめて証券化（セキュリティトークン化）することが可能になり，投資対象の幅が拡がる

④ 投資プロジェクトの運営

レポートでは，投資プロジェクトでは投資資金の運用フェーズに業務改善の余地が大きいとして，トークン発行後の管理業務をスマートコントラクトによる自動化で効率化することを

提案している．スマートコントラクトによって業務改善の可能性のある領域として挙げられているのは以下の４つである：

- 配当支払いや投票権の行使など，投資プロジェクトに参加する個々の投資家を対象に行われる定期的な事務処理の実行
- 投資資金を運用する際に発生する，プール口座と取引先銀行口座間での銀行間送金
- 人為的なミスを避けるための多重に設けられている承認プロセスの自動化
- プロジェクトの資金運用に関する外部監査を受けるための全取引を変更不可能な(Immutable な)台帳にログとして記録

⑤トークンの二次流通

より高度なトークンの活用法として，市場取引や相対取引による不動産セキュリティトークンの売買（＝二次流通）が提案されている．REIT（不動産投資信託）商品の取引に対する市場の需要が不動産取引に流動性を持たせることへの期待が表れている．REIT では資産運営業務の透明性が問題なっているがセキュリティトークン化でこれが改善できるとの期待がある．

レポートでは上記の５フェーズを考慮したセキュリティトークン発行の例として，香港ベースの SIDLEY 社(www.sidely.com)が 2019 年に構築したセキュリティトークン管理システムを紹介している（参考：図 2-12).

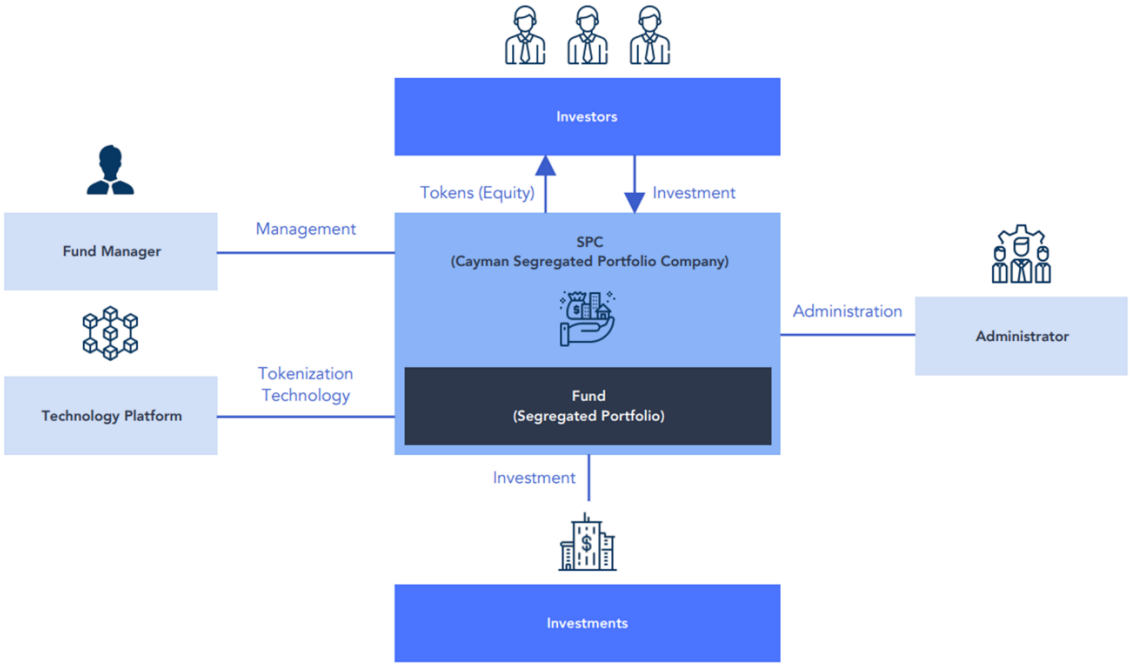


図 2-12：SIDLEY 社のセキュリティトークン管理システム（出典：[12]）

SIDELY 社のシステムでは，投資家(Investors)から集められた資金は一旦投資プロジェクトの運営事業者である SPC(分離ポートフォリオ企業: segregated portfolio company)に預託される．そして，投資資金を証券（Fund：ファンド）化したのち，各 Fund の所有権を ERC-

20 準拠のセキュリティトークンにマッピング（図中では"Tokenization Technology"と表記）している．ここで ERC20 を選んだ理由として，証券を取引しやすい少額単位に分け，投資家が保有する Fund の一部を他者に譲ったり，買い足したりできるようにするためだったと説明されている．また，Fund トークンの価値算出を容易にするための，工夫として証券を発行した後は追加の増資を行わないクローズドエンド・ファンドとして運営された．実際の資金運用は運営事業者の Fund Manager が投資資金を引き出し，その資金を運用した結果得られた利益は SPC に還元される．投資資金を運用益が発生する都度 Fund の価値に上乗せされており，Fund の価値が募集時の価値に期待される利益の配当分だけ増減した価値になっているとみなせるため，二次流通で移転される価値がわかりやすくなっている．このシステムでは，自動化できていない処理を補う目的で，投資資金の入金に応じて投資家名義セキュリティトークン発行したり，Fund Manager の入出金をセキュリティトークンの価値に反映されたり，といったセキュリティトークン台帳と，現実世界での法令に準拠するための book-entry 台帳との同期を図る Administrator の存在が不可欠になっている．

レポートの結論としては，「従来の不動産投資には，多額の金銭的コミットメント，長いプロセス，過剰な事務処理，サイロ化が伴う．共同出資型の不動産セキュリティトークンの導入は，不動産投資で集められた資金運用の効率化と情報の透明性を実現することで，これらの問題に対処する（抄訳）」として，セキュリティトークンを発行した後の管理業務に改善の余地がある，と主張している．

2.3.3.2. トークンへの投資

コインテレグラフ誌の記事 “Dividend Tokens, Explained[14] ”では，将来のトークン・エコノミーへの期待として，市場価格での変動差益以外の，所有し続けることで利益を生み出す「配当トークン」の概念を紹介し，トークン所有者間で配当を分配することで投資プロジェクトの利益を共有するビジネスモデルを提案している．

記事のなかで配当トークンの例として挙げられているのは NEXO 社のサービスで，NEXO のユーザは，自分が保有する暗号通貨を NEXO に預けると，年 8%程度の高額な利息を受け取れるとしている．ビジネスモデルはシンプルで，預金として預かった暗号通貨を暗号通貨取引所や，ベンチャー企業などに暗号通貨のままユーザに支払うより高い利率で貸出し，利率の差額で利益を上げている．

別の例としては，イーサリウム 2.0 でも採用されたコンセンサス・アルゴリズム “PoS(Proof of Stakes)”におけるステーキングが挙げられている．イーサリウム・ネットワークのネイティブ暗号通貨である ETH を一定期間ステーク(保有)することで，取引検証のプロセスに参加する権利が与えられる．イーサリウムのネットワークでは，取引検証のプロセスに参加したノードにブロック生成の結果生み出される暗号通貨が報酬として分配される．この仕組みはブロックチェーン独特であるが，経済学の観点ではステーキングが投資で、ステーキング中に得られる報酬は投資家への配当支払いに相当するとして配当トークンの例として説明されている．

記事では，配当トークンを活用するメリットについて次のように述べている：

「多くの種類の受動的収入戦略の中で，配当収入は間違いなく最良の形態です．受動的収入とは，過去に行った努力に対して継続的な報酬を受け取ることを意味し，追加の作業は必要ありません．同様に，配当トークンは，会社が市場のボラティリティにもかかわらず機能する健全なビジネスモデルを持っている場合，弱気市場であっても，追加の投資なしで定期的な報酬を所有者に支払います.」

そして主張のまとめとして，「配当トークンは，所有者がデジタル資産に結び付けた金銭的価値の制限を解除するものだ．所有者は，原資産の価値に加えて支払いを受け取りながら，引き続き配当トークンの保有を楽しむことができるからである.」と，配当トークンが長期投資に向いている理由を挙げている.

2.3.4. トークン・エコノミー実現の難しさ

2.3.4.1. 取引手数料の高騰

デジタル資産のトークン化を試みる場合に，多くの開発者によって選ばれているブロックチェーン基盤アプリは，イーサリウムである．ネイティブ通貨 ETH が多くの取引所で販売されていることや，ERC-20/ERC-721 などのトークン取引用の API をサポートしたトークン市場や，MetaMask のような操作用 GUI ツールが存在することも影響していると考えられる.

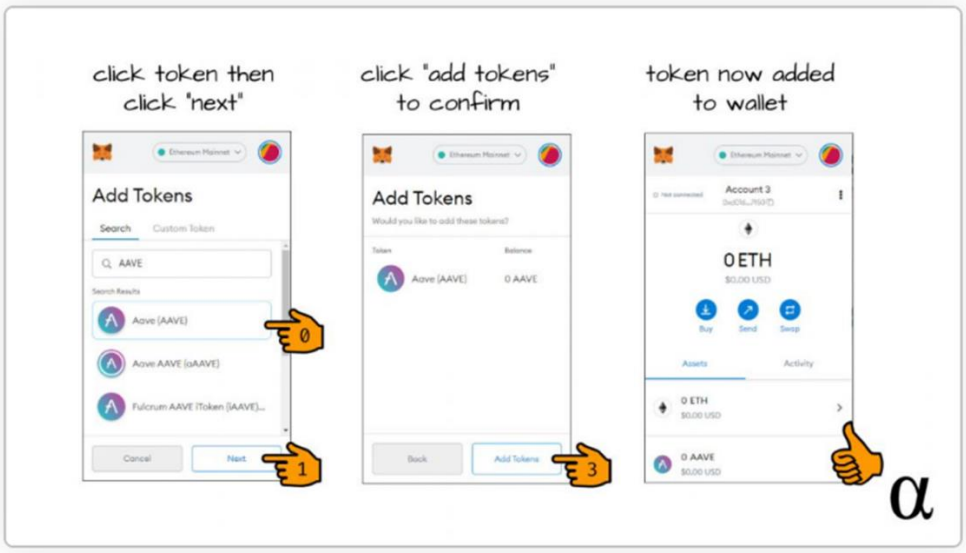


図 2-13：MetaMask への独自コイン（トークン）の登録手順(出典[6])

その一方でイーサリウムのメインネットでのスマートコントラクト稼働に必要な取引手数料ガス（gas）が高騰しており，資産価値が比較的小さいトークンの交換取引をスマートコントラクトの仲介で行うとしたら，そのスマートコントラクトの稼働に必要な取引手数料がトークンの価値に比べて高すぎて利用価値がなくなる問題が発生している.

この結果，多くのトークン化プロジェクトでは，メインネットに取引記録の要約のみを記

録して同期を図る “セカンドレイヤー”の使用や，イーサリウムの基盤ソフトウェアを使いながらメインネットには接続しない “プライベートネット運用”が選択されていることが多い．これらのアプローチでは，イーサリウムのメインネットが持つ “信頼性”を流用しようとしたものであるが，ICO プロジェクトや NFT アートの販売で発生している詐欺事例などをみる限り，信頼のチェーンは機能していない.

つまり，ETH のようなリアル社会で金銭的な価値を持つトークンと，メインネット上にない私製トークンとの安全，確実な交換取引（クロスチェーン取引）が実現できていない.

2.3.4.2. 関連規制への対応

規制当局による過度な干渉はブロックチェーン業界では好まれないが，トークン・エコノミーに参加する投資家を保護し，犯罪者の不正な取引を排除する仕組みを導入して合法的な運営を実現しなければ，トークン・エコノミーが持続的に社会に受け容れられることはないだろう.

具体的には，AML 対策に必要な本人確認(KYC 検証)は，暗号通貨の換金が行える取引所のサービスとしては行われているが，DEX サービスなどを利用すると暗号通貨の資金洗浄も理論上可能であるため，犯罪防止策として十分に機能していない.

また，将来のトークン・エコノミーでは，法定通貨を介さない物々交換も可能になることが予想されるので，暗号通貨の取引所だけでなく DEX などの無人取引所での本人確認を導入しないとトークン交換の仕組みを悪用する犯罪が増える可能性もある.

第3章 関連研究

3.1. ブロックチェーンのインターオペラビリティ

暗号通貨とは異なり，特定の組織が発行体としての役割を担う私製トークンの管理は，ビジネス上の友好関係が成立している企業間がアライアンスを組み，独立したブロックチェーン・ネットワーク（具体的にはプライベートチェーンやコンソーシアムチェーン）を構築して閉じた経済圏を形成していることが多い．しかし，参加者が少ない閉じた経済圏では大きな成長が見込めないので，複数の経済圏に参加するプレイヤーを横断的につなぐ仕組みが必要である．

ここで問題になるのがブロックチェーンのインターオペラビリティである．IT 業界での技術用語のインターオペラビリティ(Interoperability)は，あるシステムを外部システムと接続するために使われる通信プロトコル上の互換性を意味することが多いが，ブロックチェーン基盤同士のインターオペラビリティでは，通信プロトコルだけなく，合意形成のアルゴリズムや，トランザクション検証など上位アプリ層での意味上の整合性や，信頼を担保するための一貫性が求められるため，単純に繋げる以上の難しさがある．

ブロックチェーン経済圏拡大を目的としたインターオペラビリティに関する先行事例として，代表的な２つのアプローチ，“オラクルの導入”と，“クロスチェーン技術”を紹介し，これらの取り組みで未解決の課題を整理する.

3.1.1. オフチェーン情報の取り込み：オラクルの導入

ブロックチェーン・ネットワーク内の世界は「オンチェーン」と呼ばれ，これに対してブロックチェーン外部の世界は「オフチェーン」と呼ばれる．このような区別を行う用語が定義されているのは，リアル社会のさまざまな問題を解決するには，オンチェーンでの情報だけでは足りず，なんらかの形でオフチェーンにある情報やデータを参照する必要があるからである．しかし，ブロックチェーン上での安全な取引仲介を行う目的で利用が増えている通常のスマートコントラクトには，このオフチェーンの情報やデータを取り込み，動作ロジックに反映する機能はない.

そこでリアル社会で起きる事象をトリガーにしたスマートコントラクトの利用を可能にするのがオラクル[15]である．オラクルは，ブロックチェーンの世界と“ブリッジ(Bridge)”と呼ばれるノードを介してつながっており，スマートコントラクトの依頼を受けてオフチェーンにしかない情報やデータの取得を行い，スマートコントラクトの動作ロジックに反映する役割を果たす.

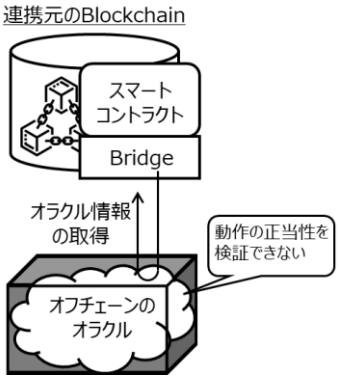


図 3-1：ブリッジを活用したオラクル情報の活用

オラクル導入のアプローチには，「中央集権型」と「分散型」の２種類がある．

3.1.1.1. 中央集権型オラクル：Provable

中央集権型のオラクルでは，１つの主体がオフチェーンへのアクセスに責任を持ち，通常その主体がブロックチェーンやスマートコントラクトを代表してオンライン上の情報源にアクセスし，得られた情報をブロックチェーンに伝える．

中央集権型オラクルの例として，Proavable(旧称：Oraclize)[16]がある．

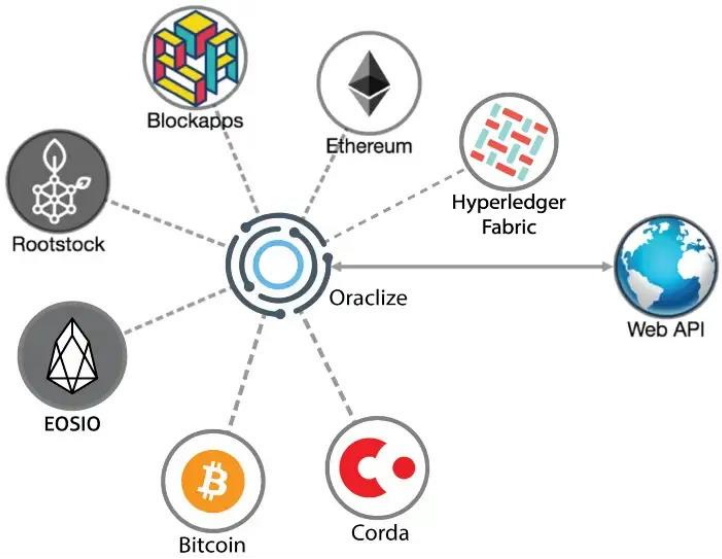


図 3-2：中央集権型オラクルの導入による Web API の参照 （出典: [16]）

Provable が提供する DApp 用ライブラリを使えば，スマートコントラクトから Web 上の情報を Web API 経由で以下に挙げたオフチェーンデータを簡単に参照できるようになる：

- URL: URL データ ソース タイプを使用すると，インターネット上の任意の API または Web ページにアクセスできるようになる．
- Random: 耐タンパー性のあるハードウェア疑似乱数生成器によるランダムビットデータを参照できるようになる．

- WolframAlpha: WolframAlpha データ ソース タイプを使用すると、WolframAlpha Knowledge Engine API(サードパーティ・サービス) に直接アクセスできるようになる.
- IPFS: IPFS データ ソース タイプを使用して、分散型ファイルストレージ IPFS ネットワーク上に置かれたファイルのコンテンツを証跡記録の対象として参照できるようになる.

中央集権型オラクルの場合，オラクルから得られた情報は検証なしに取り込まれるので，ブロックチェーンの振る舞いがオラクルの提供する情報に依存してしまうデメリットを生む．また，オラクル自体がシステム構成上の単一障害点となるため，ブロックチェーンを利用することの利点となるスケーラビリティや耐故障性を無効にしかねないという指摘もある．

3.1.1.2. 分散型オラクル：Chainlink

分散型のオラクルでは，中央集権型と同様にオフチェーンの情報源へアクセスして情報やデータを取得するが，オフチェーンから得られた情報を集めたうえ，複数の主体で集められた情報の正当性や正確性を検証してから取り込む点が異なる．このとき行われる検証が通常，それぞれの情報源から得られた結果がすべて一致することの確認で実現されているため，合意型のオラクルと呼ばれる場合もある．

分散型のオラクルの実装例として Chainlink[17][18]がある．

Chainlink では独自のユーティリティトークン LINK を発行しており，オラクル情報の検証を行うノード運営者への利用料報酬の支払いに使われる．この LINK は，パブリックチェーンのイーサリウム・ネットワーク上に取得した記録を残し，情報を請求したスマートコントラクトへの情報伝達メディアとしても使われる．

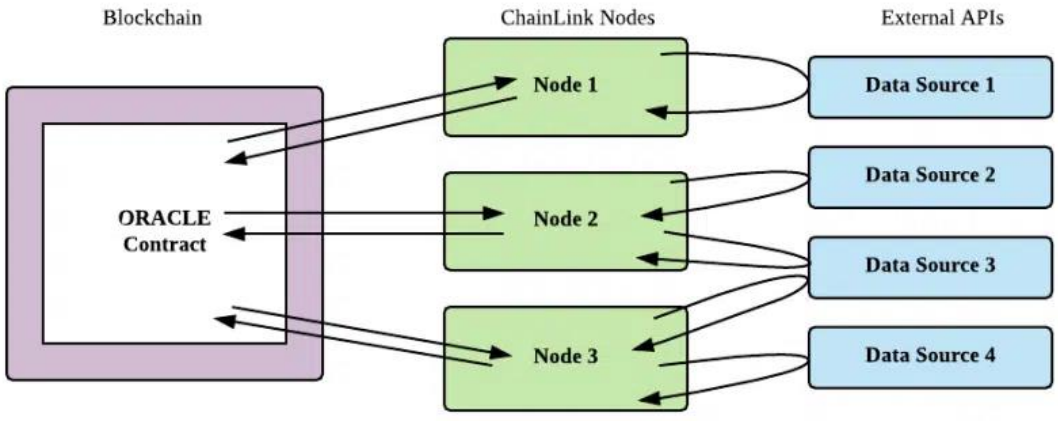


図 3-3：Chainlink Node によるデータソース参照結果の検証 （出典：[18]）

分散型のオラクルでは，中央集権型のオラクルよりブロックチェーンのトラストに関する考え方を反映しているといえるが，オフチェーンの情報源自体を信用することが絶対条件になるので，ブロックチェーンの外部依存度が高くなるデメリットは同様に発生する．つまり，

スマートコントラクトをリアル社会の問題に関わらせるためにオラクルの導入が行われているが，オラクル自体をどうやって信頼できるものにするか、という，いわゆる「オラクル問題」は依然未解決のままである.

3.1.2. ブロックチェーン経済圏の統合：クロスチェーン技術

スマートコントラクトをリアル社会への適応させる，オラクルの導入とは別のアプローチとして，ブロックチェーン連携（クロスチェーン）技術がある．ここでは，経済圏の統合を目指したブロックチェーン連携技術の代表事例を紹介する.

3.1.2.1. アトミックスワップ（Atomic Swap）

アトミックスワップ(Atomic Swap)[19][20]は，ユーザ間で互いが持つ異なる種類の暗号通貨を直接交換するためのプロトコルである．

暗号通貨では各ブロックチェーン・ネットワークでひとつのネイティブ通貨のみを管理するようになっているため，暗号通貨の交換には複数のブロックチェーンを同時に操作し，全体としての処理の整合性を図る，クロスチェーン取引が必須となる．クロスチェーン取引は，法定通貨のドルを円に両替するのに似ていて，両替商に相当する暗号通貨取引所の関与が不可避であった.

アトミックスワップは，暗号通貨間の両替において取引所の関与なしで実現しようとする試みで，２つの異なるブロックチェーン・ネットワークで発生する２つの送金取引があたかも１つの取引であるかのように両方が成立するか，両方ともキャンセルするか，を同時に(アトミックに)確定させる技術である.

所有するデジタル資産を交換したいユーザも，権限なしでアトミックスワップを実行できるようになっている．ただし，両方のブロックチェーン ネットワークでハッシュド・タイム・ロック・コントラクト(HTLC: Hashed Time Lock Contract)が使えることが前提条件になっている．交換トランザクションが，Alice(LTC: 実在する暗号通貨 LTC を所持している)と，Bob(ビットコイン: BTC を所持している)の間で互いが所有する暗号通貨を直接交換する取引を行うと想定してアトミックスワップの動作を説明する.

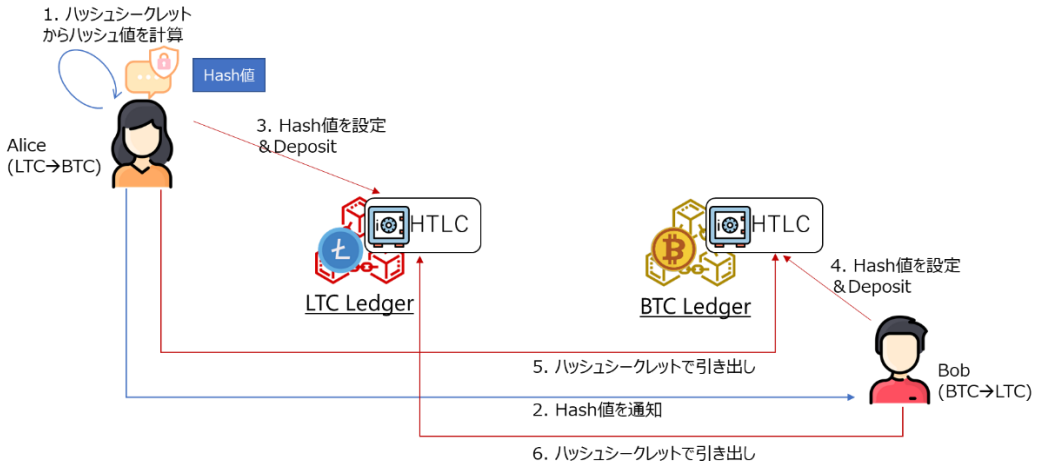


図 3-4：HTLC を活用したアトミックスワップ

最初に Alice は，取引で引き渡す予定の LTC コインを保持するためのセーフティボックスとして機能する HTLC を作成する．HTLC の作成時には，初期化のパラメタとしてハッシュ値が与えられる．このハッシュ値は，Alice のみが知る秘密“ハッシュ・シークレット”を使って作成されており，のちに HTLC に預託された資金を引き出す claim()操作を実行する際に，このハッシュ・シークレットが提示された場合にのみ，これを承認する．Bob は，HTLC のハッシュ値のみを知っており，預託された LTC を要求するためのハッシュ・シークレットをまだ知らないことに注意してほしい．

次に Bob は，Alice が共有するハッシュ値を使用して BTC を預託するためのもうひとつの HTLC を作成し，BTC をここに預ける． Alice が Bob の入金を確認した後，Alice は Bob が作成した HTLC で claim() 操作を実行し，Bob によって入金された BTC を受け取る．Bob は Alice の claim() 操作でハッシュ・シークレットを観察できるため， Bob は LTC 預託用の HTLC で claim() 操作を実行して，Alice が預託した LTC を受け取ることができる．アトミックという言葉は，これらのペアのトランザクションがいずれかで成功するという事実に関連しており．または全てがキャンセルされることを意味する．つまり，一方が期待された役割を果たせなかった場合には契約はキャンセルされ，資産は自動的に所有者に返却される．

3.1.2.2. Polkadot

Polkdot [21][22][23]は，任意のデータをブロックチェーン間で転送できるようにするネットワーク・プロトコルを使って，プライベートチェーンからデータを取得し，パブリックチェーンでそのデータを使用する，といったマルチチェーン・アプリケーション・システムを構築できる．

Polkadot では，パラチェーンもしくはパラスレッドと呼ばれる異種ブロックチェーンのネットワークを，リレーチェーンと呼ばれる Polkadot のメインネットワークに安全に統合する．

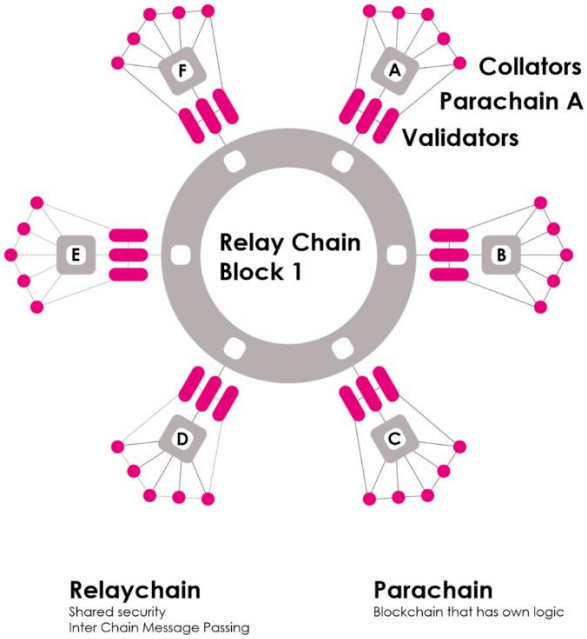


図 3-5：Polkadot ネットワーク (出典: [23])

任意のパラチェーンをリレーチェーンに接続することで，ブロックチェーン・サービスをカスタマイズすることができる．また，パラチェーンでの恩恵としては，リレーチェーンの一部として動作することでパラチェーンのトランザクションもリレーチェーンと同等のセキュリティを確保することができる．

さらにパラチェーンは，リレーチェーンとの連携により，システム全体でのスケーラビリティの改善を可能にする．一般的なブロックチェーン・ネットワークでは，一秒あたり数十トランザクションしか処理できないが，多数のパラチェーンとメインチェーンが連携動作することで，毎秒数千のトランザクションに対応できるようになると Polkadot の Whitepaper のなかで説明されている．

3.1.2.3. Plasma

Plasma[24]は 2017 年 8 月に提唱された，イーサリウムのスケーラビリティ問題を解決するためのサイドチェーン技術である．

スケーラビリティ問題とは，利用者の増加によって処理速度が遅くなり，円滑な取引ができなくなる，またその遅延を取り戻すために相対的に取引手数料が高くなり利便性が損なわれることを指す．ブロックチェーンは，従来の銀行などと比べて安価な手数料で場所を問わず世界中の誰に対しても短時間で，かつ安価な手数料で送金ができることが大きなメリットになっているが，スケーラビリティ問題が解決されなければその魅力が失われてしまい利用価値がなくなってしまう．

また，サイドチェーン技術とは，その名から想像されるようにメインのブロックチェーンとは異なるブロックチェーン上でトランザクションを処理する仕組みである．

Plasma は以下のような特長を持っている：

- パブリックチェーンと同レベルのセキュリティ
- 送金手数料（ガス）がかからない
- ファイナリティ（ある取引が正当であると検証される）までが最短 200ms
- 暗号通貨 ETH がそのまま使える

Plasma の動作原理は，メインのブロックチェーンから独立した環境で計算処理を行い，最終的な結果をメインのブロックチェーンに記録する，というものである．サイドチェーン「プラズマ・ネットワーク」のノードは，イーサリウムのノードとは別に用意されるが，イーサリウム上に設置された連携用のスマートコントラクトを介して上記の同期処理が実現されている．

プラズマ・ネットワークで送金を行えるようにするには，連携用のスマートコントラクトに ETH をデポジット（入金）手続きをとる必要があり，逆にプラズマ・ネットワークに移していた資金を元のメインのブロックチェーンでも通用する形に戻る際には，イグジット(exit:出金)手続きをとる必要がある．プラズマ・ネットワークの世界では処理すべき取引が限られるのでより高速な送金が可能となるが，メインのブロックチェーンへの書き戻し処理(具体的には，出金やプラズマ・ネットワーク内部での送金の確定)では，メインのブロックチェーンのスマートコントラクトで合意をとるため取引の確定までに時間がかかる欠点がある．

Plasma の White Paper ではスケーラビリティ問題以外での貢献の将来性についても説明されており，ETH の資産価値を中心にしたトークン・エコノミーの将来像が示されている．

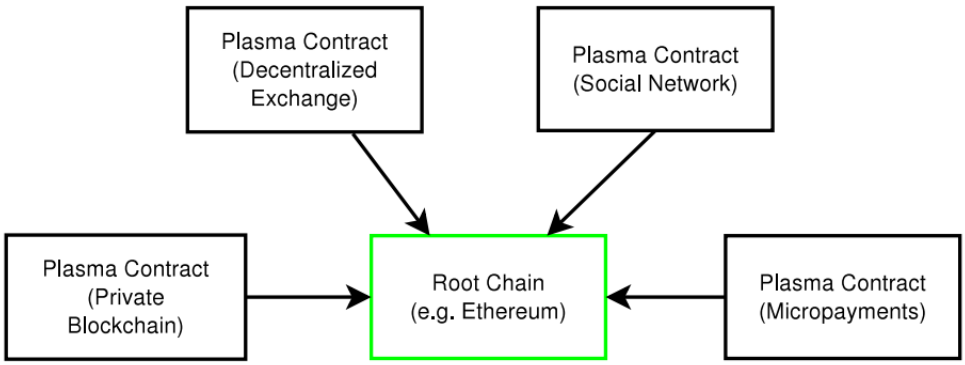


図 3-6：Plasma Contract の連携イメージ （出典：[24]）

3.2. トークン・エコノミーに関するブロックチェーン技術の動向

トークン化がスマートコントラクトで実現されることから，そのトークンを交換するトークン・エコノミーの実現に必要な要素技術が開発され，インターネット上で実際に使われている．

3.2.1. 暗号通貨取引所の自動運用 – DEX

DEX(Distributed Exchange：分散型取引所)は，資金の管理を仲介者に任せることなく，暗号通貨間の直接取引を可能にするピア・ツー・ピア市場である．DEX の実体はスマートコントラクトになっており，プログラム・コードで記述された運用ルールに従って自動的に為替取引サービスを提供する．

DEX 上の取引では，ユーザが自身で管理する秘密鍵の管理を行うことでその主権が十分に尊重されるように設計されている．DEX が提供するサービスは，リアル世界での金融機関が提供する金融商品の交換サービスに似ているが，取引手数料を安価にする代償として，ユーザが秘密鍵を紛失したり，送金の宛先を間違って入力したりしても，これを取り戻す手段は提供されないことに注意が必要である．

初期の DEX システムでは，売り手と買い手のマッチングが必要なオーダーブック型の DEX であったが， AMM(Automated Market Maker)型 DEX が現在の主流になっている．AMM は，Bancor で初めて導入されて [25]から，他の DEX にも採用が広がって現在も主流になっている．AMM 採用のメリットは，流動性プール(Liquidity Pool)と呼ばれるスマートコントラクトが仮想的な買い手となることで，売買に参加するユーザが少ない場合でも交換取引に必要な暗号通貨の供給量を確保できるようにした点にある．

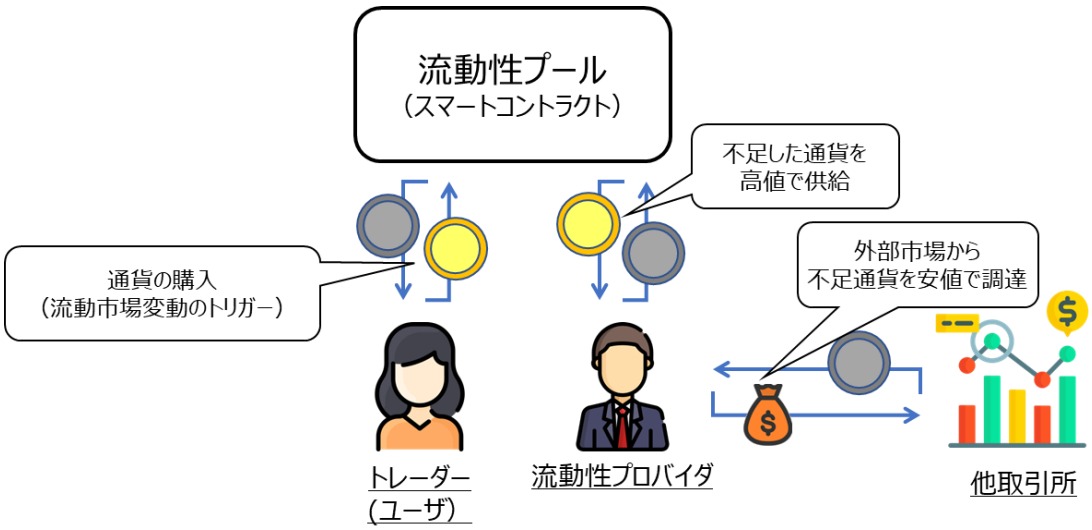


図 3-7：流動性プールによる交換対象通貨の供給量確保

流動性プールでは DEX 内での通貨間交換レートをその占有率で割り出していて，暗号通貨を売買しようとするユーザは，この流動性プールで売りたい暗号通貨を入金すると，交換レ

ートに従って計算された量の受け取りに指定した暗号通貨が自動的に払い出される．また，交換レートが外部の取引所の交換レートと乖離せず，交換に必要な暗号通貨の供給を確保するために流動性プロバイダと呼ばれる事業者を DEX に参加させている．流動性プロバイダは，流動性プールでの交換レートを見ながら，供給量が不足して取引価格が高騰した暗号通貨をそれより安価な交換レートで外部市場から調達し，流動性プールに供給することで為替差益を得ることができるように工夫されている (図 3-7 参照).

3.2.2. 匿名送金 - Zcash

暗号通貨の送金では，ある程度の匿名性が確保されるが，送金額などが平文で共有されるためトラフィック分析などの手法で，プライバシ情報が漏洩する危険性がある．Zcash[26] は，ビットコインで使われているトークン残高管理方式の UTXO(Unspent Transaction Output)を採用しているパブリックチェーンであるが，本当の送金先アドレスや送金額を秘匿したまま安全に暗号通貨の送金を行う特別な機能を提供するために考案された.

Zcash の送金では，transparent アドレスと shielded アドレスが 使われ，transparent アドレスから shielded アドレスに送金された後，shielded アドレス間で行われる送金の流れは，ブロックチェーンで共有される取引データからはわからなくなる．このとき shielded アドレス間の送金に使われるのが zk-SNARKs と呼ばれるゼロ知識証明の手法である.

Zcash でのプライバシ送信では，自身が shielded アドレスにプールされた資金を引き出すための鍵を所持していること，またプールされている残高から指定した量の資金を引き出して，その資金を指定するアドレスに送ることに合意したこと，を zk-SNARKs で証明する．

この証明内容を，検証ノードが管理する nullifier と呼ばれる資金引き出しの記録リストと照合し，不整合がなければこれを受理することで資金の移動が承認される.

Zcash の方式では，現在主流の残高管理方式の暗号通貨には適用できないが，ブロックチェーンではすべての情報を公開しなければならない，という常識を覆した画期的な技術である.

3.2.3. 本人確認と分散 ID 管理

顧客の誰と取引しているかを知ることは，銀行や証券取引所などの金融機関にとって最も重要な業務のひとつである．KYC (Know Your Customer)検証プロセスと呼ばれる本人確認の手続きは，資金洗浄などの犯罪抑止のため各国の規制で義務付けられているため，手間がかかっても避けることはできない.

たとえば，ある顧客が銀行Ａに口座を開設する場合には，銀行Ａに身分を証明するための書類一式を送付する必要がある．顧客の書類を受け取った銀行Ａは，それらを確認してはじめて銀行 A に口座を開設することが許可される．ここまでは良いとして，この顧客が他の銀行，たとえば銀行Ｂや銀行Ｃとも取引を必要となった場合にも，銀行 B と C のそれぞれでまた一から KYC 検証プロセスを実行する必要があるのが現状である.

このような旧来の KYC 検証は，銀行での口座開設時などのレアケースにのみ必要であった

が，オンラインペイメント・サービスや，暗号通貨でのサービス支払いが普及しつつあるなか，より厳格かつ頻繁な本人確認が規制当局から求められるようになっている．このようなニーズの変化を踏まえ，現行 KYC 検証の非効率を改善するため，ブロックチェーン技術を応用した KYC 検証の取り組みとして，2017 年 7 月から 2018 年 3 月にかけて実施された「本人確認（KYC：Know Your Customer）高度化プラットフォームにおけるブロックチェーン技術の適用に関する実証」がある（参考：[27]）．

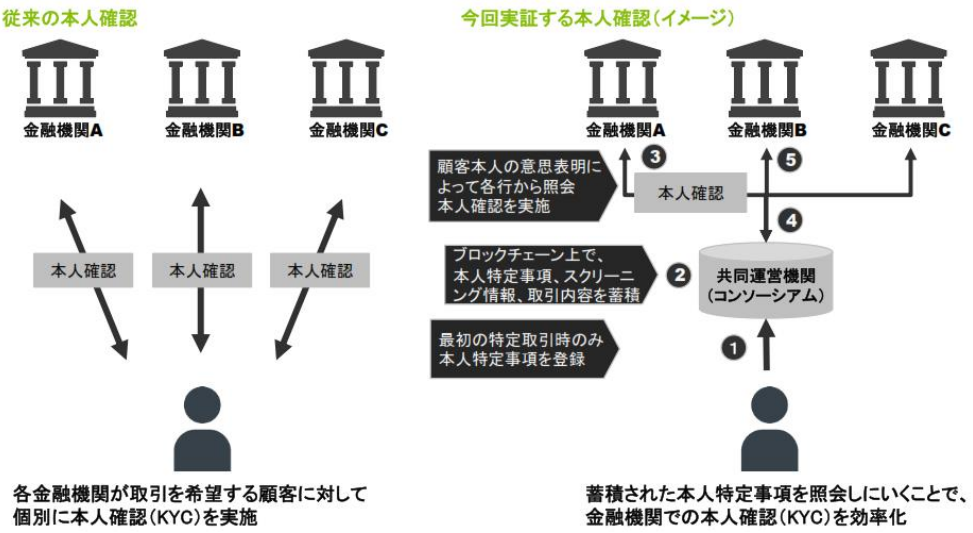


図 3-8：本人確認（KYC）の仕組みの模式図（出典：[27]）

この実証実験の報告書では，初めて金融機関と取引を行う顧客が KYC 検証を受けたとき，その提出した確認書類の情報（本人特定事項）や，審査結果などをコンソーシアム型ブロックチェーンに記録しコンソーシアム内に公開することで，以降別の金融機関との取引を開始する際には，ブロックチェーン上の記録を再利用することで，KYC 検証に必要な確認書類や，その検証結果などを再利用することができる，とされている．

ところで，実際の金融機関が KYC 検証を行う際には，検証の精度を上げるためにユーザ自身に提出を求める身分証明情報などだけでなく，調査機関に依頼して収集する勤務先や年収などのプライバシ情報なども使われている．このため，これらのプライバシ情報の共有は最低限に抑えつつ，オンライン完結な KYC 検証を実現したい，というニーズがある．

このニーズに応えるのが「自己主権型アイデンティティ（SSI：Self-Sovereign Identity)」で，プライバシ保護のため個人が自身の個人データの共有や使用方法を完全に管理すべきである，という考え方である．また，この SSI の考え方を技術的に実現する仕組みとして，W3C の Verifiable Credentials Working Group[28]で標準化された，検証可能な資格情報（VC：Verifiable Credential）[29] と，その VC を特定するための DID(Decentralized

Identifiers)[30]を扱うためのツールとして Hyperledger Indy[d]が利用可能である．

図 3-9：DID のフォーマット（出典：[30]）

DID(図 3-9)で記述される"Scheme"，は DID の上位概念である URN でのデータ仕様を区別するためのもので"did"で固定となる．一方"DID method"はどのブロックチェーン・ネットワーク上管理される分散 ID 体系であるかを識別するユニークな ID で，”DID Method Specific String”との組み合わせでグローバルユニークな ID となる．

前述の Hyperledger Indy を用いて構築される VC ベースの分散 ID 管理システムのアーキテクチャを図 3-10 に示す．

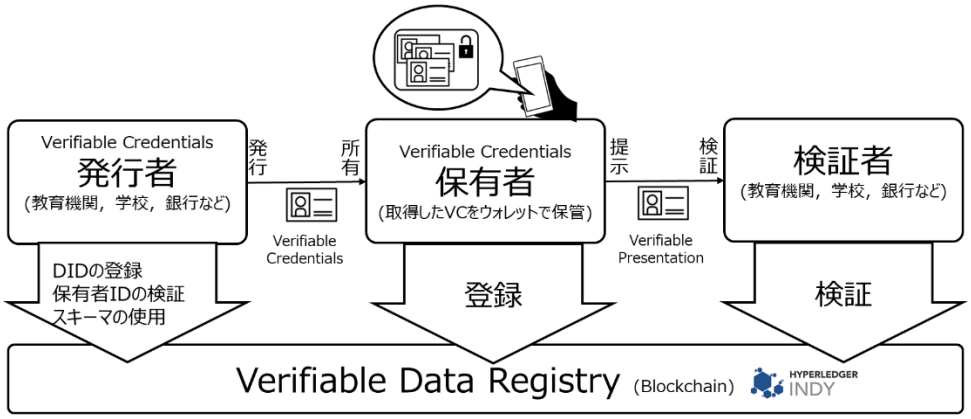


図 3-10：Hyperledger Indy を用いた分散 ID 管理システム（参考：[29])

このアーキで分散 ID 管理サービスのユーザが担うのは「保有者（holder)」の役割で，VC の発行を依頼し，発行された VC を「ウォレット」と呼ばれるアプリ内で管理する．VC はユーザに関するさまざまな証明内容（資格の保持や，卒業証明など）を格納できるデータ・コンテナで，VC に格納される証明内容に電子署名を付与して，証明内容の正しさを保証する主体を「発行者（issuer)」と呼ぶ．そして発行された VC を検証する役割を担うのが「検証者（verifier)」である．VC は，保有者の要求に合わせて発行者が作成されるが，身分証明の都度発行を受けなくても，あらかじめ発行されウォレット内に保管されている VC を活用することができる．具体的には，検証者からの身分証明要求があった際に，保有者は自身の都合にあわせて，ウォレット内の VC の一部に目隠しなどの加工を施したり，いくつかの VC をまとめたりして，検証者への提示に適した Verifiable Presentation を仕立てることができる．

このように VC ベースの分散 ID 管理システムは，プライバシに配慮した認証・認可を実現

[d]分散 ID 管理システムの OSS 実装. DID や VC を扱うためのツール、ライブラリなどを提供する． https://wiki.hyperledger.org/display/indy/Hyperledger+Indy

する．

3.2.4. トークンへのプログラマビリティの追加

現状，暗号通貨を含むブロックチェーン・トークンの売買は取引所で行われており，決済システムとしての利用は少ない．その一方で，DEX の例にもあったようにトークンの貨幣としての価値が確定したり，トークンの物々交換などが広く認知されるようになったり，すれば決済システムの在り方を一変させる可能性がある．

2022 年 6 月に日本銀行が発行した報告書「決済システムにおけるプログラマビリティの実現[31]」では，ブロックチェーンが将来担うべき役割として決済システムにプログラマビリティを挙げ，「将来の決済システムの検討においては，プログラマビリティを高めるアプローチを模索しつつ技術研究を進め，デジタル社会にふさわしい決済手段の実現を目指していくことが重要と考えられる」と述べられている．

この報告書では，決済システムで強制的に運用ルールを適用するためのプログラマビリティを決済システムへ追加手法として「内部プログラム方式」と「外部プログラム方式」を挙げているが，それぞれに利点と課題が存在するとして課題の解決に期待が述べられている．

本報告書が期待する決済システムでは，トークンそのものが運用ルールを自動的に遵守するモデルを想定しており，ユーザから見てこれらを意識することなく使えるようにすることを提案している．この提案内容を図で表現したのが図 3-11 である．

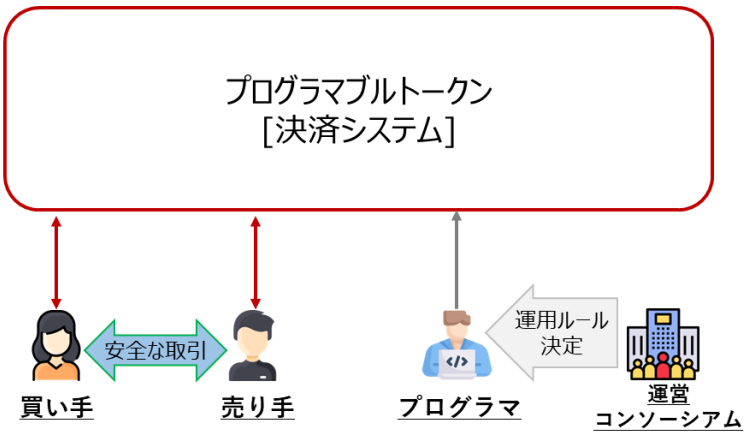


図 3-11：プログラマビリティをトークン（決済システム）への組み込むイメージ

つまり，プログラマブル・トークン利用の特長は，売り手や買い手などの取引当時者は特に運営ルールについて意識しなくても，トークンの決済システムにプログラムとして組み込まれたルールが強制的に適用される決済システムを指す．

プログラマブル・トークンは，トークン取引システムの理想形であるといえる．

第4章 拡張スマートコントラクト

資産価値をスマートコントラクトでユーザ間が直接取引可能になる世界，トークン・エコノミーの実現に向けて，複数の経済圏を横断的につなぐ取り組みでの未解決の課題を解決する技術として，拡張スマートコントラクトを提案する．

4.1. トークン・エコノミー実現に向けた課題の整理

ブロックチェーンとスマートコントラクトの技術により，管理者不在な形でもさまざまな資産価値をデジタル化し，その資産価値をブロックチェーン・ネットワークに参加する一般ユーザ同士で仲介者なしで直接移転できるようになった．

また，企業体により管理運営されるコンソーシアムチェーンの登場や，発行体の社会的信用を資産価値の裏付けとする私製トークンの発行により，現実世界にある有形，無形のさまざまな資産価値をオンライン上で取引可能にする「トークン化」が注目を集めている．

現在，ブロックチェーンの基盤ソフトは多数があるが，ことトークン化に限っては，時価総額がビットコインに次ぐ第二位とされているイーサリウムに利用が集中している．これはイーサリウムが私製トークンの発行・移転に不可欠なスマートコントラクト機能を備えた初のブロックチェーンであったことと，ネイティブ・トークンである ETH が取引市場での市場流動性の高い暗号通貨であるためトークン売買の支払い手段として有用である，ことが他を圧倒する強みとなっている．

しかし，イーサリウムの人気が高まるにつれ，トランザクション数が急増したことによる取引完了までの遅延や，ETH の取引市場での高騰によりスマートコントラクトの稼働に必要なガス(gas)代がその処理内容にそぐわないほど高価になった，また ETH の取引価格が安定しておらず価値貯蔵の役割が機能していない，などトークン・エコノミーを実現するうえで致命的な問題が顕在化しつつある．

このような状況を受け本研究では， トークン化の未来として，トークンの発行形態やその価値保証の担保に依らず，当事者間の同意があれば自由に，安全に取引可能にする世界を真のトークン・エコノミーと定義し，これを実現する手段としてスマートコントラクトベースのトークン取引システムを構築することにした．

トークンの発行や運営の容易さから，私製トークンは発行体が構築したプライベートチェーン，もしくはコンソーシアムチェーン上で運営されることが多い．これらの私製トークン同士を交換できるようにするためにはまず，両者が運営されているブロックチェーン同士が連携できるようになっていなければならない．このための技術であるクロスチェーン技術の使用は必須である．また，片方のブロックチェーンの世界から見ると，交換取引でペアとなる別のブロックチェーンは，外の世界「オフチェーン」となるので，オフチェーンとの連携を仲介するブリッジや，交換取引全体を俯瞰的に把握して取引をサポートするオラクルの導入も検討する必要がある．

しかしながら，オラクル導入とオフチェーンの先行事例では，トークン・エコノミーの実現に必要な機能要件が満たされていない．

オラクル導入のアプローチでは，オフチェーンの情報を取り込むための現実的な解ではあるといえる．しかし，オラクルを利用する側のブロックチェーン参加者がオラクルの動作を絶対的に信用する必要があり，オフチェーン情報を取り入れる場合には使えるが，トークン取引の仲介で中立性を求める場合には，ユーザやサービス事業者の信頼を得にくいという問題がある．また，オラクル自身が独自の経済圏を形成している場合には，運営ガバナンスの分担がオラクル側に依りすぎていてオラクルを利用する側のブロックチェーン経済圏の支持を得にくい．

一方，二つのブロックチェーンを直接相互接続するクロスチェーンのアプローチでは，連携するブロックチェーンの関係性が問題となる．Atomic Swap では，２種類の暗号通貨を等価な分量だけ HTLC コントラクトに預託して，取引当事者間で直接通信することなしに交換取引を行えるが，交換レートというものは市場価格として変動するものであるから，HTLC コントラクトは送金があったというだけではなく，送金額が当事者間で同意された金額と同額であるかのチェックを行い，送金額に過不足があった場合には送金を無効とする必要がある．つまり，連携するブロックチェーン同士が対等な場合には，交換レートなどを恒久的なルールに同意することが難しく，結果として HTLC コントラクトを使い捨てにしなければならないなど，サービス提供の安定性に問題が生じる．Polkadot では，汎用性を持たせるために，中央に置いたリレーチェーンを信頼点としながらリレーチェーン自身は連携ロジックそのものには一切関与せず，連携先ブロックチェーンであるパラチェーン間でやりとりされる連携取引を記録するだけにとどまっている．結果として，Polkadot のアーキテクチャでは取引全体の整合性を取る役割は信頼点であるはずのリレーチェーンにはなく，他のサイドチェーンの機能や役割を認識した特別なサイドチェーンに任されることになるので，連携先のブロックチェーンがその特別なサイドチェーンで稼働するスマートコントラクトを絶対的に信頼する必要がある．このような信頼関係は本来取引当事者同士の合意のもとで構築されるべきものであり，スマートコントラクトの実装に任されているのは運営ガバナンス上の問題がある．また，Plasma のようにメインネットに対するサイドチェーンといった上位と下位の概念がある連携は，処理速度の向上などスケーラビリティの課題解決には使えるが，価値の自由な流通をめざすトークン・エコノミーのコンセプトからすると経済圏を拡げる取り組みにするのは難しいと考えられる．

4.2. 拡張スマートコントラクトの提案

スマートコントラクトは，トークン取引自動化の可能性を示したが，ユーザ同士が取引条件を決め，決済を行うような真のトークン・エコノミーは実現できていない．

4.1 節で整理したトークン・エコノミー実現に向けた課題を解決するアプローチとして，ブロックチェーンとブロックチェーンを連携専用の拡張機能を備えたスマートコントラクトが動作するようにしたブロックチェーンで連携させるアプローチを考案し，拡張スマートコントラクトとして提案する．

4.2.1. 課題解決へのアプローチ

トークン・エコノミーでスマートコントラクトを活用するには，現状のオラクル導入やクロスチェーン技術だけでは難しいことがわかった.

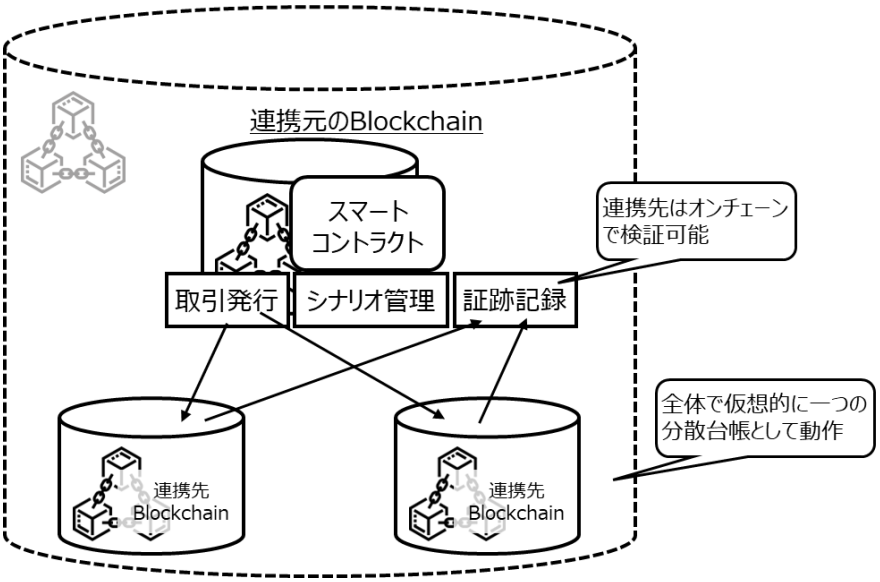


図 4-1：拡張スマートコントラクトの提案

今回提案する拡張スマートコントラクトは，スマートコントラクトの考え方や機能を踏襲し，外部のブロックチェーン・サービスを統合する際，統合対象となるブロックチェーンへの取引発行や，ブロックデータのモニターとその結果の記録（証跡記録）が行えるように拡張しており，連携専用のコンソーシアム型ブロックチェーン上で動作するように設計した.

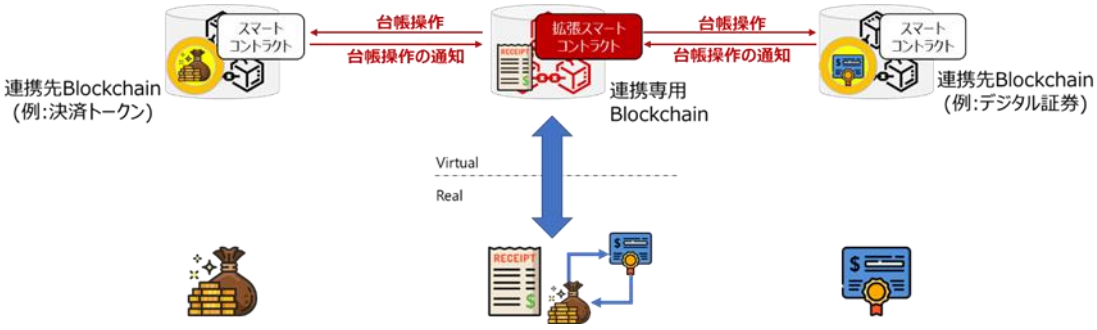


図 4-2：拡張スマートコントラクトとリアル世界での経済活動との関係

取引の発行と証跡記録をスマートコントラクト自身が行うことで，オフチェーンとなる部分がなくなり，拡張スマートコントラクトを利用するユーザの視点では，シナリオの一部を構成する外部ブロックチェーンでの取引記録が，拡張スマートコントラクトの動作するブロックチェーン上には証跡記録を行ったトランザクションとして取り込まれているので，仮想

的に一つの分散台帳が動作しているように見える（図 4-1 での拡張スマートコントラクトを参照）.

拡張スマートコントラクトの実行環境にコンソーシアム型のブロックチェーンを選択したのは，拡張スマートコントラクトはシナリオ上の処理条件を決定的に判断する必要があり，これを担保できるのが現状参加組織間での合意内容についてファイナリティ性が確保されたコンソーシアム型ブロックチェーン状のスマートコントラクトに限られているからである.

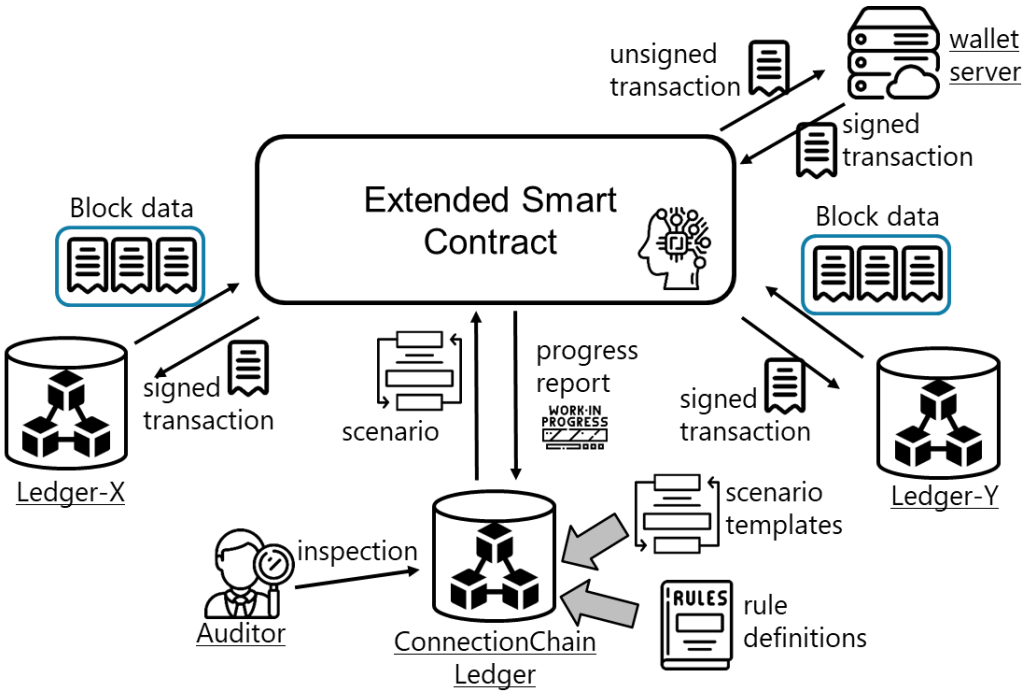


図 4-3：拡張スマートコントラクトのアーキテクチャ図

この拡張スマートコントラクトの振る舞いは，「デジタル資産の移転」などの定型動作を動作時に決定されるパラメタが変数の形で埋め込まれた「動作シナリオ」を用意した.

拡張スマートコントラクトでは，台帳操作などの「動作シナリオ」を順番に沿って実行していくが， 実際のブロックチェーン台帳の操作は，台帳の操作を指示するトランザクション・データ（transaction）をブロックチェーン・ネットワークに送信しただけでは反映されず，コンセンサス・アルゴリズムを使った検証プロセスを経ないと確定しない．このため，拡張スマートコントラクトの処理エンジンは，ブロックチェーン・ネットワークに送信したトランザクション・データを特定する情報を記憶しておき，そのトランザクション・データが確定するまで待ち受けを行うようにした.

このとき注意が必要なのは，ブロックチェーン台帳への操作は非中央集権な形で運営されているので，トランザクション・データによる台帳操作が承認されない場合への考慮である．台帳操作が承認されない単純なケースは送金元口座の残高不足であり，事前のチェックを行えば回避できる．一方で台帳が管理する特定の値を複数個所から同時に書き換えようとする，いわゆる“操作の競合”の発生は予測が不可能で不可避のエラーが発生しうるのである．つ

まり，こうした複数の台帳操作が関わる取引の最中にエラーが発生した場合，その原因によってトランザクション・データの再送，トランザクション・データのパラメタを変更して作り直す，その時点までに確定した台帳操作の取り消しを行う，などエラーの発生原因に応じて対応を変える必要がある．

特に問題になるのが台帳操作の取り消しで，ブロックチェーンにおけるセキュリティの根幹が一度確定した台帳操作が決して元に戻らない、という特性の上に成立しているからである．この台帳操作取り消しへの対処として拡張スマートコントラクトではエスクロー取引機能で対処した．エスクロー取引とは，ネットオークションや通信販売などで用いられている取引手法で，商品の買い手が仲介者へ代金を預託し，商品の受け渡しが正しく行われたと仲介者が判断したときだけ預託された代金が売り手に引き渡される．そして，商品に問題があった場合には，商品の返品などの返金条件が満たされたと仲介者が判断した時点で預託された代金の返金が行われる．

拡張スマートコントラクトにおけるエスクロー取引機能では，一連の取引のなかで元の状態への復元が必要な台帳操作について，その操作が一時保留にするトランザクション・データの実行と，拡張スマートコントラクトがその操作を確定させるトランザクション・データの実行に分け，拡張スマートコントラクト処理エンジンが監視対象とした取引の状態変化などのシナリオ判定に応じて，当該操作を確定するトランザクション・データか，保留中の取引を無効化するためのトランザクション・データのいずれかを発行する（図 2-1）．

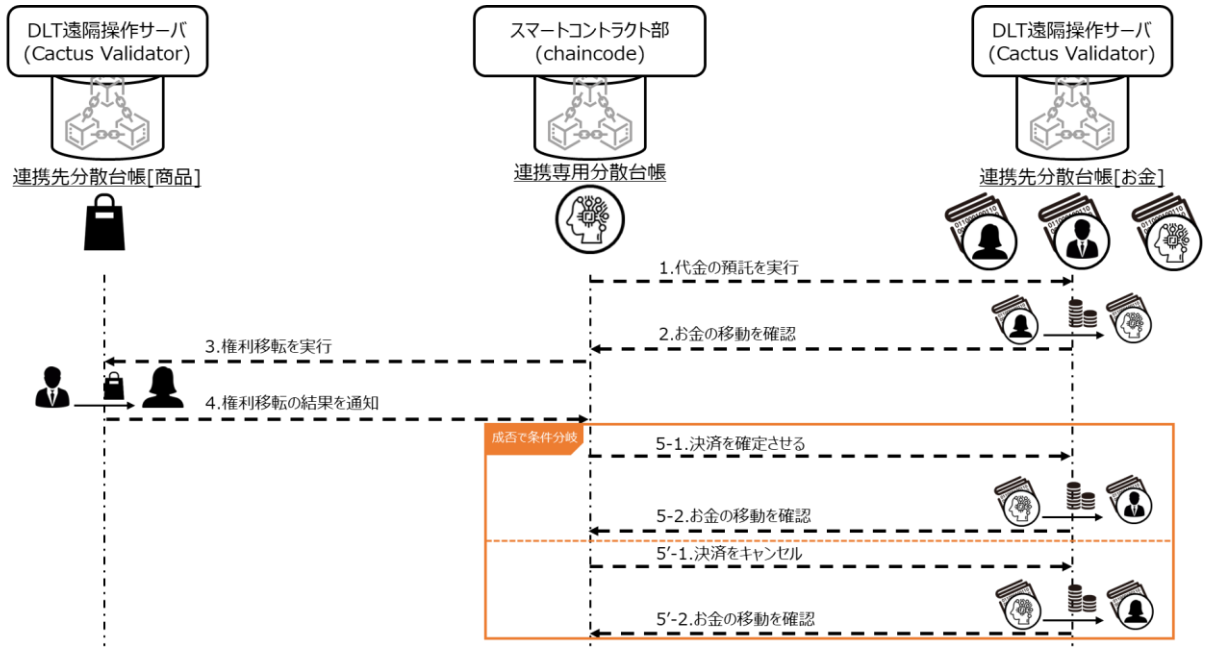


図 4-4：エスクロー取引の処理シーケンス

拡張スマートコントラクト処理エンジンが判断する条件には，先に挙げたエラー発生時のリカバリ処理だけでなく，ユーザが拡張スマートコントラクトの起動時に与えられるユーザ変数や，台帳に記録されたシステム変数の参照結果も含められるようにして，拡張スマート

コントラクトのアプリ開発が柔軟な処理ロジックをプログラミングできるようにした．

このように様々な判定条件で関係づけられた複数の動作シナリオを 1 つに組み上げたものを「シナリオテンプレート(scenario template)」の形で拡張スマートコントラクトに設定するが，拡張スマートコントラクトの運営ガバナンスでの透明性を高めるために，scenario template の設定・変更操作は管理者の特権としつつ，専用のスマートコントラクトを経由させることで連携専用ブロックチェーン上に履歴が残るように設計した．

また，拡張スマートコントラクトにオフチェーンで決定されるオラクル情報を安全に取り込むための仕組みとして，「ルール定義(rule definition)」設定機能を用意した．

ルール定義にはシナリオ ID で特定される scenario template に依存した複数のシナリオ変数が定義されており，管理者は専用スマートコントラクトを介してこれらのシナリオ変数を設定・変更することで拡張スマートコントラクトの振る舞いを scenario template を変更することなく動的に制御できるようにしている．

そして，拡張スマートコントラクトは，シナリオ ID で特定される動作シナリオ依存で定義される複数のユーザ変数をユーザが与えて拡張スマートコントラクトを起動すると，システム変数，ユーザ変数が scenario template で空欄になっていたパラメタに適用されることで，拡張スマートコントラクトが読み込む動作シナリオが確定する．

例えば，異なる暗号通貨同士の交換を行う拡張スマートコントラクトで，シナリオ変数に交換レート，ユーザ変数に交換後の暗号通貨を送金するアドレス、代金支払いのための交換前の暗号通貨の支払い承認データがあったとする．拡張スマートコントラクトは，コンソーシアムの運営により開発・設置され，交換レートは交換業務を行うサービスプロバイダが変更する．シナリオ ID で識別される拡張スマートコントラクトに設定されている交換レートは，便宜上 Web インタフェースなどで表示されるが，マスターデータは拡張スマートコントラクトの管理用分散台帳で記録・公開されるため，シナリオ変数を問い合わせるスマートコントラクトによっていつでもその値を確認できるようにして，ガバナンスを確保できる．このようにして設置された拡張スマートコントラクトをユーザが利用する際には，利用しようとする拡張スマートコントラクトの起動に必要なユーザ変数の収集を，サービスアプリがユーザとの対話操作で行い，すべての準備が整ったのちにユーザの最終承認を得て拡張スマートコントラクトが起動される．

ところで非中央集権型の管理を旨とするブロックチェーンの世界ではユーザの電子署名のみが承認の証となる．しかし，拡張スマートコントラクトでは，起動後に動作シナリオで実行される台帳操作の詳細が決定されるので，拡張スマートコントラクトの起動時には取引に電子署名を行うことはできない．一方で，起動時に拡張スマートコントラクトに渡されるパラメタ変数は，コンソーシアムへの参加メンバーで共有されるので，署名を行うための秘密鍵を処理エンジンに預けることもできない．この対策として，ユーザの代わりに必要な電子署名を行うエンティティとして“署名サーバ（図 4-3 中では wallet server と表記)”を導入した．

署名サーバは，拡張スマートコントラクトの処理エンジンが動作シナリオの実行中に行われる条件判定の結果，発行が決定されたトランザクション・データ（transaction data）に対

してユーザから預かった秘密鍵を使って電子署名を付与する．

拡張スマートコントラクトは，起動後ユーザとの対話的操作が不要となるようにしたので，事前にプログラミングされた動作シナリオ中の判定条件にしたがって，次に起動すべきシナリオを選択，選択したシナリオを自動実行する．この処理を繰り返した結果，すべての動作シナリオの実行が完了すると，拡張スマートコントラクト実行の成否が確定する．

ユーザの視点に立つと，拡張スマートコントラクトの実行ステータスは API 経由で常に，モニターできるようになっていて，処理中/キャンセル/成功，のいずれかのステータスであると確認できる．

4.2.2. 評価システムの実装

前節で提案した拡張スマートコントラクトの実現性を評価するため，評価システム ConnectionChain(コネクションチェーン)を試作した．

拡張スマートコントラクトの主要機能は，コンソーシアム型ブロックチェーン基盤ソフト Hyperledger Fabric でのスマートコントラクト機能“chaincode”で実装された「スマートコントラクト部」と，これを拡張スマートコントラクトとして動作させる「拡張スマートコントラクト実行部(以下では拡張ＳＣ実行部と表記)」で構成され，この部分はコンソーシアム型の管理者団体により共同運用されることを想定した．

また，上記２つの構成要素に加え，コンソーシアムの外にユーザがサービス利用契約を結ぶ特定の事象者が運営するサブシステムとして「署名サーバ部」と「フロントサービス部」があり，これらを組み合わせてユースケースに合わせたシステムに統合し，Web サービスとして利用可能なＷｅｂサービスを構築する．

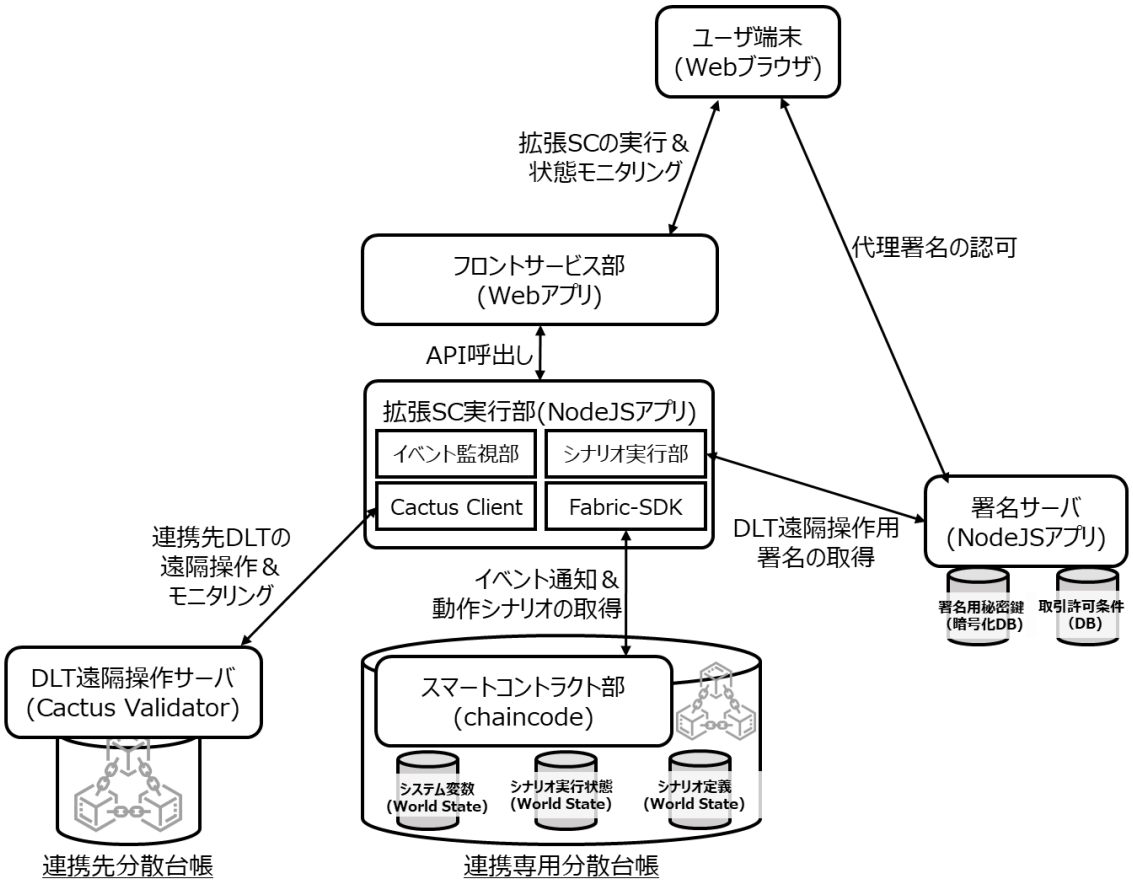


図 4-5：評価システム ConnectionChain のアーキテクチャ図

ConnectionChain の主機能は，拡張 SC 実行部が担っている．

拡張 SC 実行部はユーザに対して，フロントサービス部経由の API 呼出により，拡張スマートコントラクトの動作シナリオの起動や処理状態の問い合わせ，を受け付ける．このとき，起動された動作シナリオには取引 ID が割り当てられる．拡張スマートコントラクトの実行は，非同期で処理されるためユーザは取引の結果をすぐに知ることはできないが，シナリオ起動時に取引 ID が通知されるので，取引 ID を使って動作中の拡張スマートコントラクトの状態監視を行うことができ，動作が完了した後，取引の成否を取引 ID で問い合わせるようにした．

連携先となる分散台帳（以下連携先 DLT と表記する）に対して，拡張 SC 実行部は動作シナリオで検出した現在状況に応じて決定される連携取引の実行に必要なトランザクション・データを生成する．この取引データは，動作シナリオの起動時に設定されていたシステム変数と，ユーザパラメタをシナリオ定義のテンプレートに当てはめ，完成させた取引データを連携先 DLT に送信することで台帳を遠隔操作する．

ところで拡張ＳＣ実行部による遠隔操作では，ユーザ口座からの資産移転など，拡張ＳＣ実行部による代理操作でユーザが承認した証，つまりユーザの秘密鍵による電子署名が必要となる．しかし，拡張スマートコントラクトの実行は非同期で実行され，処理の途中で電子署名が必要になっても秘密鍵を管理するユーザに電子署名を請求することができない．ユー

ザがシステムと対話できない状況でも署名を行えるように，秘密鍵をサービス提供者に預けることも考えられるが，このアプローチでユーザは，資産の管理を拡張 SC に一任することになり運営ガバナンスの透明性が失われる．ConnectionChain では，台帳操作を行うトランザクション・データへの電子署名の付与する機能を，拡張 SC 実行部から見てシステム外部に，独立したサーバとして「署名サーバ」を置く設計にした．この署名サーバは，ユーザの代わりに台帳操作を実行するための秘密鍵を預かって，拡張 SC 実行部が台帳操作を行うために生成したトランザクション・データに電子署名の付与を依頼するようにした．この署名サーバは，ユーザが代理署名の条件を確認して権限を委譲するための仕組みとして，インターネット標準の OAuth2.0 プロトコル[32]を採用している．OAuth2.0 は、ユーザが Web ブラウザを使った UI 上での対話操作を行って，そのユーザの代わりに特定のリソースへのアクセス，ここでの使い方では電子署名を付与する機能の呼出し，を許可するためのプロトコルである．OAuth2.0 の仕組みにより，ユーザは署名サーバによる署名付与を許可する条件を細かく設定可能になり，設定された許可条件を認証トークン・データの形で，署名サーバ利用するアプリケーション，つまり拡張 SC 実行部，へ安全に引き渡すことが可能となる．

ユーザが代理署名を必要とする拡張スマートコントラクトを利用する際には，フロントサービスとの対話操作のなかで，拡張スマートコントラクトの呼出しに必要な他のパラメタ設定したあと，拡張スマートコントラクトの起動を確認する処理の一貫で，Web ブラウザの接続が一時署名サーバへ転送され，署名サーバと OAuth2.0 プロトコルに準拠した対話操作で電子署名付与に必要な認証トークンを取得する．Web ブラウザは，認証トークンの取得後に再度フロントサービスへ再転送され，設定済みの他の起動パラメタと共に拡張スマートコントラクトの起動を要求する（図 4-6）．

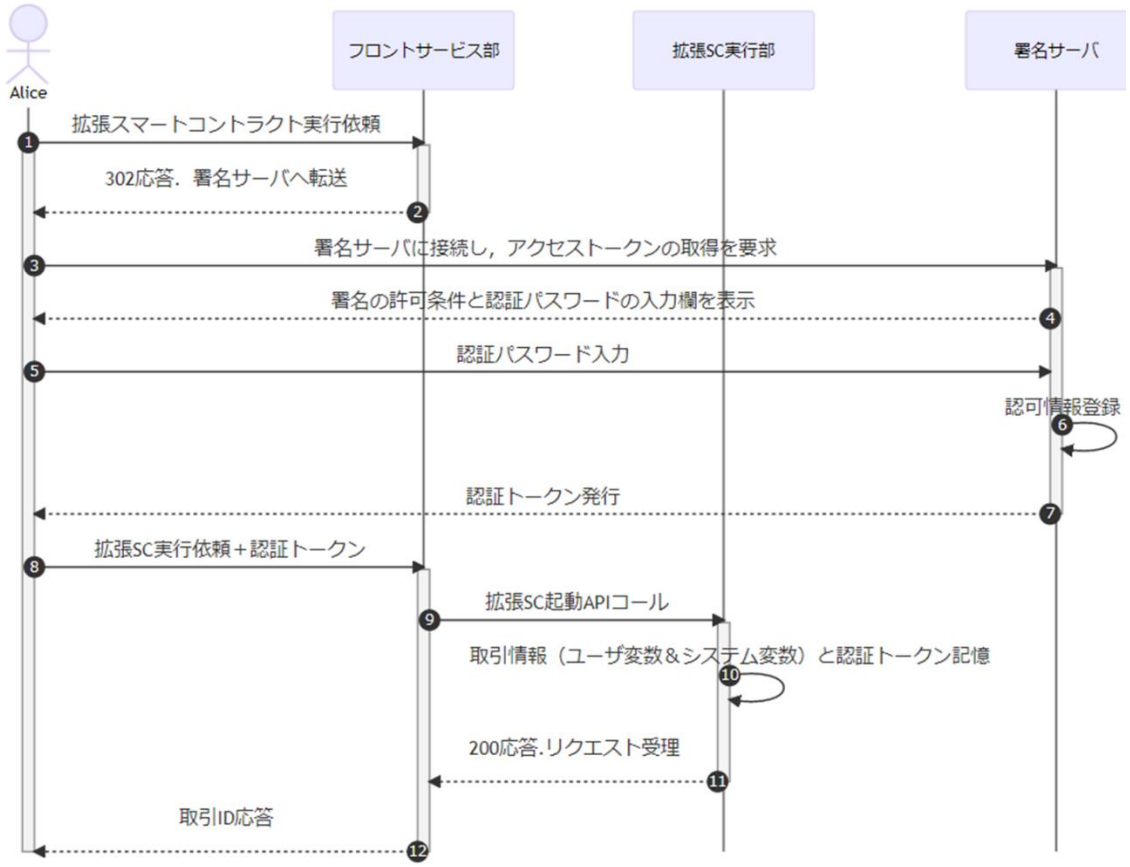


図 4-6：OAuth2 プロトコルに沿った認証トークンの受け渡し処理

まず，外部のブロックチェーン・ネットワークとの通信手段であるが，多くのブロックチェーンの実装では、ブロックチェーンの起源とみなされているビットコインの設計を参考しているため，分散台帳の実現方法に違いはあるが，台帳操作やスマートコントラクトの呼出しなど共通点が多い．

スマートコントラクト部は，動作シナリオ上の内部状態の変化やシナリオ変数の管理を専用の台帳に記録する役割と，登録されている動作シナリオと，内部状態の変化に応じて次に実行すべき処理を判断する役割を担っている．

拡張 SC 実行部は，連携先となる他のブロックチェーンで検知された取引結果などの外界イベントをスマートコントラクトへ通知したり，スマートコントラクトの下した判断に従って連携先のブロックチェーンを制御したりする．

取引データを準備したり，取引の実行承認に必要な電子署名の付与を行ったりするなど拡張スマートコントラクトが他ブロックチェーンへ行う遠隔操作を支援する．

4.2.3. ConnectionChain デモアプリ

ConnectionChain は汎用プラットフォームで，動作シナリオを実装してはじめてサービス

を提供可能となる．ここでは，ConnectionChain の動作を説明するために試作したデモアプリについて説明する．

デモアプリが実現しようとするサービスの想定ユースケースでは，特定の商店街にある商店でのみ使用できるなどの使途に制限が加えられた地域通貨と，相互送客に関連したサービスを考案した．

デモアプリの背景シナリオとして，地理的には離れていて姉妹商店街の関係にある商店街同士が，それぞれ独自の地域通貨をブロックチェーンで運用しているとした．この地域通貨は商店街内の店舗に限定される代わりに，法定通貨からの交換時に１０％のインセンティブを付与して発行されるようになっている．今回，年末時期の購買喚起を狙った期間限定のキャンペーンとして，互いの地域通貨ウォレットを持つユーザに対してさらに１５％のインセンティブを付与して購入時の法定通貨から換算すると実質２５％のインセンティブが付与される特別交換ルールを適用できるようにすることにした．

ただし，それぞれの商店街の店舗が備える決済システムは，当該商店街専用の地域通貨決済にのみ対応できるようになっているため，姉妹商店街の地域通貨の決済は行えないものとする．この条件下で，各店舗の決済システムには変更を加えずに，特別キャンペーンを ConnectionChain の機能だけで実現する．

追加のデモシナリオとして，商店街Ａでは中原ドル，商店街Ｂでは富士通円という地域通貨が導入されているとした．商店街Ａでは法定通貨１円に対して１中原ドルが発行されるようになっている．これは商店街Ａでは値札に日本円での値段とインセンティブが提供された後の中原ドルでの値段を併記しておくと，米ドルでの表記に近くなり外国人訪問客が値ごろ感をつかみやすいという副次的な効果を狙ったためである．ただし，商店街Ａの店舗では中原ドル決済端末で中原ドルでの金額を入力する必要がある．一方で，商店街Ｂでは地域通貨富士通円を法定通貨と同じように使えるようにするため発行時にインセンティブを付与して発行している．具体的には，法定通貨１円に対して 1.1 富士通円が発行することでインセンティブ付与を通貨の発行時に適用している．このように２つの地域通貨は，通貨単位が異なり，また各商店での決済操作も異なるため，さきの特別キャンペーンの適用で他方の地域通貨を直接利用できるようにするには困難が予想される．

以上のような前提を置いて，ConnectionChain を使って，通常の決済ルールにはない，地域通貨の優遇レートを適用した交換，決済店舗が所属していない組織（具体的には姉妹商店街）が運営する地域通貨のウォレットのみを持つユーザが店舗での買い物で支払いができるようにすることをデモアプリで達成すべきゴールとした．

まず，商店街Ａが決済プラットフォームに採用したイーサリウムと，商店街Ｂが決済プラットフォームに採用した Hyperledger Fabric の差異は，Hyperledger Cactus の Ledger Plugin で吸収する．Cactus では，連携先ブロックチェーンが異なる場合でも，付属するユーティリティライブラリの助けを借りると，連携先ブロックチェーンで必要なトランザクションの遠隔起動が可能になる．

このデモアプリで連携させる２つの地域通貨は，それぞれイーサリウムと Hyperledger Fabric と異なるブロックチェーン基盤を使って実装した．イーサリウムをベースにしたブロ

ックチェーン基盤では，ERC-20 標準に準拠したトークン管理用スマートコントラクト用ライブラリ，OpenZeppelin[33]を使い，単体で動作可能な地域通貨決済システムを構築した．一方，Hyperledger Fabric では，通信プロトコルのレベルでは互換性がないものの，操作 API が ERC-20 ライクなオープンソースのチェーンコード token-erc-20[34]を流用して，単体で動作可能な地域通貨決済システムを構築した．

次に，通貨交換の動作ロジックを拡張スマートコントラクトで実装した．今回のデモシナリオでは，店舗での買い物を想定しているので，ユーザ端末から店舗ＩＤと支払いに指定すべき地域通貨ＩＤを指定してから支払い金額入力すれば，ユーザが所有する地域通貨のウォレット口座から店舗が指定した口座に指定された地域通貨建ての決済トークンが送金されるようにする．

本デモアプリでは，ユーザによる端末操作を最小化するため，店頭に掲示するＱＲコード[e]をスマートフォンで読み込むと端末内蔵のＷｅｂブラウザが起動され，フロントサービス部を構成する Web アプリの決済システムのポータルサイトへ誘導されるようになっている．

ポータルサイトでは，決済サービスに事前登録した ID とパスワードでログインできるようになっていて，QR コードで読み取られた URL にアクセスすると，店舗 ID と支払い先の地域通貨 ID が指定された状態の送金依頼のフォームが表示される．

ユーザは，画面に表示される店舗名が正しいことを確認し，ブラウザに表示される入力欄に店員が指示する地域通貨の金額を入力して，「決済実行ボタン」を押下して地域通貨での決済を地域通貨決済システムに依頼する．

この送金手続きでは，ユーザが保有する地域通貨の口座からの送金が必要になるので，決済実行ボタンの押下後に，リクエストを受信したポータルサイトにより署名サーバへリダイレクト転送が実行される．署名サーバでは，送金金額の見積もり額とその送金を承認するための承認用パスワードを入力する欄が表示される．

ユーザが正しいパスワードを入力すると，再度ポータルサイトへ再転送が発生し，送金手続きが受理され，決済処理が進行中である旨が画面に表示される．このとき，署名サーバでは認証用パスワードでの認証結果に応じて取引データに承認を与えるための認証トークン・データが発行され，ユーザ端末の Web ブラウザのセッション情報として記憶されることになる．そして，認証サーバがサポートする OAuth2 プロトコルの規約に則った HTTP レスポンスを使った再転送で，署名サーバが発行した認証トークンの値がポータルサイトへ渡される．

こうしてユーザが保有する地域通貨の口座からの送金に必要な認証トークンが用意されたので，ポータルサイトとの対話操作で設定されたユーザパラメタに認証トークンを添えて，拡張スマートコントラクトの起動が行われる．

ブロックチェーンにおける取引は，即時には確定しないので，しばらくは処理中のままとなるが，エラーが発生しなかった場合には１０秒ほどで決済取引が完了したとの表示される

[e] QR(Quick Response)コードは，マトリクス型二次元バーコードの一種．端末内蔵のカメラを使い，手入力では困難な長い URL 文字列を共有するのに適している．

（図 4-7 の左図）．また，送金元の口座の残高が足りなかったなどのエラーが発生した場合には，取引が完了しなかったエラー原因と共に，途中まで実行された取引の実行結果と，場合によっては取引を無効化するためのリカバリ取引の実行結果も併せて表示される（図 4-7 の右図）．

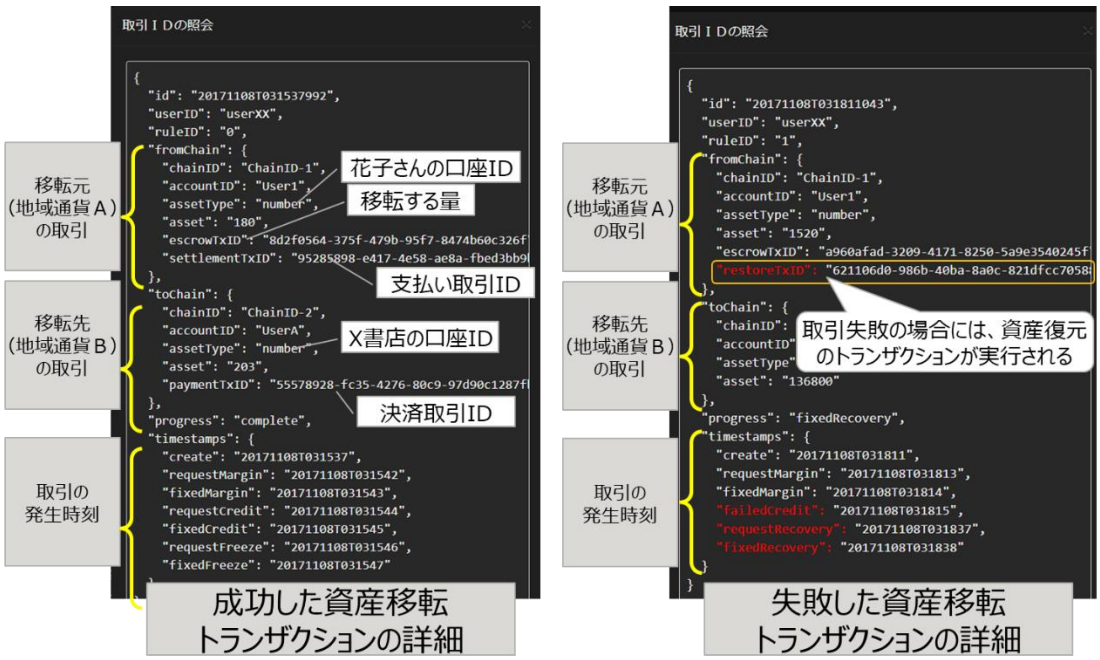


図 4-7：動作シナリオの実行結果（証跡データ）の例

デモアプリでは，ConnectionChain のエラー処理で使われているエスクロー取引シナリオの振る舞いを可視化するために，買い手ユーザ「花子さん」の地域通貨 A での残高と，売り手ユーザ「書店」の地域通貨 B での残高を時系列で管理者用ダッシュボード上で可視化した（図 4-8）

図 4-8：管理者ダッシュボードで表示された送金取引ステータスの一覧画面

4.3. 評価システムを使った実証実験

ConnectionChain を使い，提案したブロックチェーン連携システムの有用性を検証するために複数の実証実験を行った．以下では，実証実験の目的と評価システム導入の効果について説明する．

4.3.1. 旅行者向け地域通貨の決済

近年商店街などでの地元での消費促進を目的として，提携店舗での商品購入に限って利用可能“地域振興券”などの“地域通貨”を発行する試みが数多く行われている．しかし，偽造防止の対策や専用決済システムの構築などのコストが見合わないとして，簡易な偽造防止機能を導入して印刷したクーポン券での非デジタルな地域通貨の運用に留まっている．また，多くのデジタル地域通貨では，ユーザ間での譲渡などの二次流通までは考慮されておらず，デジタル通貨の点々流通性などの特性を活かしにくい状況にあることも普及を妨げる一因となっている.このような状況を踏まえ，筆者らは地域通貨利用の促進策として，海外から日本に来訪する旅行客の消費拡大を狙う地域通貨決済システムの実験を行うことにした．

ユーザにとっては不必要な外貨を保有することなく，旅行先の商店などでの商品購入やサービス利用の対価を外貨（商店にとっては邦貨）建てで支払い可能にする機能を実現した.

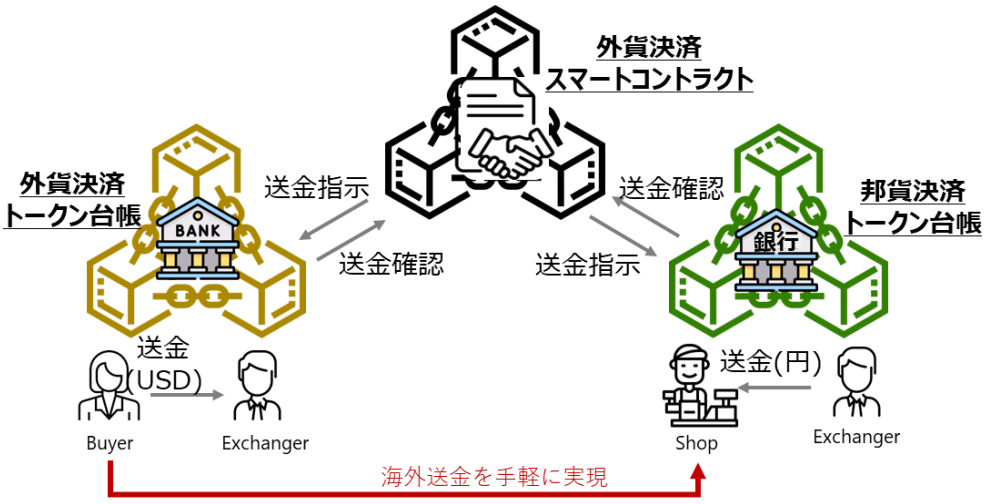


図 4-9：旅行者向け地域通貨決済システムの概要

サービス利用の前提として，旅行者は居住地でのローカル通貨建てでの決済用地域通貨を調達しておく．今回の実験では，支払いを行う外貨（例：US ドル）の金額から計算した分の金額の資金を銀行口座から決済用通貨の口座に都度チャージする利用を想定して行った．また，店舗では支払いを受けるためのローカル通貨建て（例：日本円）の地域通貨口座を通貨 ID と口座 ID の組み合わせで表現した情報を二次元バーコード化したステッカーを店頭に掲示しておく．

この決済サービスを利用しようとするユーザは，決済用アプリを起動して先の二次元バーコードを読み取らせると，店舗にとってのローカル通貨（例：日本円）建ての金額と，決済手数料を含む旅行者にとってのローカル通貨（例：US ドル）建ての金額を自動計算する UI が表示され，決済金額の入力をユーザに促す．

ユーザが代金として請求された金額（例：990 円）を入力すると，決済用アプリは決済事業者が定めた為替レート（1 ドル＝144 円）にしたがって計算した送金額（例：990 x 1.44 = 6.67 USD）を表示し，ユーザに承認用の PIN 入力を求める．

ユーザが正しい PIN を入力した場合には，認証サーバから電子署名付与を行うための認証コードが発行され，外貨建ての地域通貨決済用の拡張スマートコントラクトの実行が開始される．

この拡張スマートコントラクトでは，シナリオ変数として決済事業者毎に異なる為替レートを決済事業者自身が設定・変更できるようになっている．また，拡張スマートコントラクトの起動時に指定されるユーザ変数には，送金先口座の通貨 ID と口座番号，決済事業者 ID，送金元口座の通貨 ID と口座番号，邦貨建てと外貨建ての送金金額と，送金元口座からの決済トークン引き出しを許可する認証トークン，が指定できるようになっている．

先に述べた決済アプリのユーザ操作では，あらかじめ決済事業者が為替レートを設定しており，アプリでは決済事業者ＩＤが固定される実装になっていた．それ以外のユーザ変数は，旅行者ユーザのＵＩ操作によって決定され，シナリオＩＤで特定される拡張スマートコントラクトの起動時パラメタとして ConnectionChain に送信される．

以下では起動された拡張スマートコントラクトの振る舞いを説明する．

シナリオＩＤを使って拡張スマートコントラクトが呼び出されると，シナリオ処理エンジンの内部状態が ConnectionChain の台帳上で動作するスマートコントラクトを介して初期化され，シナリオＩＤに紐づく形で ConnectionChain に登録されている初動シナリオがシナリオ処理エンジンにロードされ，送金元口座の地域通貨を拡張スマートコントラクトが管理するエスクロー口座に移転する許可を与える取引データが自動生成される．この取引データを実行するためには電子署名が必要なため，ConnectionChain は，生成した取引データをユーザの署名鍵を預かっている署名サーバに拡張スマートコントラクトの起動時パラメタとして与えられた認証トークンのデータを添えて署名の付与を依頼する．

署名サーバは，取引データと認証トークンの組をチェックした結果，正当なリクエストであると判断すると，取引データにユーザの署名をつけて ConnectionChain に返答する．

ConnectionChain は，ここまでの処理で初動シナリオの実行に必要な処理は完了したとして，この取引データが確定した場合の通知待ち受けを活性化させたのち，署名が付与された取引データを送金元口座が存在するブロックチェーンに送信して処理を完了させる．

ConnectionChain では，連携先となっているブロックチェーンの取引状態を常時モニタリングしており，監視対象の取引が確定したり，処理に失敗したり，といったイベントが発生すると待ち受けの処理ロジックに設定されているシナリオＩＤを使って現在の待ち受け状態を確認，シナリオの一部として記述された条件分岐ルールに従って，内部状態の遷移を ConnectionChain の台帳上で稼働するスマートコントラクトに要求する．スマートコントラクト側では，検出されたブロックチェーン取引の確定通知などのイベント情報と，現在の内部状態で決定されるシナリオに照合し，指定された内部状態への状態遷移が妥当だと判断できた場合にのみ内部状態を更新する．この内部状態の更新成功が，シナリオの更新と連動しており，ConnectionChain が次に実行すべき処理内容を決定する．

本実証実験でプログラミングされたシナリオは以下の通りで，シナリオに設定された処理が順次実行される．

1. 外貨建ての決済トークンをエスクロー口座に預託
2. 邦貨建ての店舗への支払いを決済事業者の所有する残高で決済
3. （預託されていた）外貨建ての決済トークンを決済事業者に送金

上記１の処理で決済トークンの預託を行っているが，これはなんらかの理由で途中の処理が失敗した場合には，預託した資金を送金元の口座へ送り返すことで，すべての取引をキャンセルできるようにするためである．

この実証システムの導入効果としては，異なる種類の地域通貨（具体的には異なる台帳で管理されている通貨単位の異なる決済トークン）の交換取引を拡張スマートコントラクトが仲介することで，ユーザに複雑な操作や手続きを強いることなく，為替レートに沿った両替が伴う外貨建ての決済システムを構築できることが確認できた．

4.3.2. デジタル証券売買の決済

本実証実験の背景として，セキュリティトークンが投資家の注目を集めていることがある．

セキュリティトークンとは，ブロックチェーンで有価証券の所有権の管理を行えるようにしたものである．ブロックチェーンを使った資金調達の仕組みとしては，ICO(Initial Coin Offering)が先行していたが，活動実態のない投資プロジェクトの存在や詐欺の横行による被害が数多く発生し，社会問題にもなった．ICO での資産価値の裏付けに関する問題の解決を図る目的で使われるのがセキュリティトークンによる資金調達の仕組み， STO(Security Token Offering)である．セキュリティトークンの発行や販売は，各国の法規制に基づいて証券会社などの事業者免許を取得した金融機関によって行われる．2020 年 5 月に施行された金融商品取引法の改正により，日本でもセキュリティトークンの発行や販売が合法的に可能になっている．STO を実施することのメリットには，取引履歴の公開による監査業務の効率化もあるが，ブロックチェーンによるトークン化の効果として，売買取引に仲介者を必要としない直接即時決済，DvP(Delivery versus Payment)を導入することで，所有権移転と決済のスピードアップすることへの期待が大きい．

しかし，セキュリティトークンの発行は運営ガバナンス確保の理由から，プライベートチェーンやコンソーシアムチェーンなどの非パブリックチェーンで運営されており，ICO のときのように暗号通貨を決済に使うことが難しい．これはデジタル証券の譲渡を行うブロックチェーンと代金支払いを行うブロックチェーンが分かれてしまうと，既存のスマートコントラクトでは自身が稼働するブロックチェーンの外側にある他のブロックチェーンへの台帳操作や，口座状態の確認などを行うことができないからある．

本実証実験では，異なるブロックチェーンに対する台帳操作や取引状態の監視が可能な ConnectionChain を使い，セキュリティトークンを管理するブロックチェーンと決済トークンを管理するブロックチェーンを連動させて，デジタル証券の所有権移転と代金支払いを同時に実行する DvP 即時決済についての技術検証を行った．

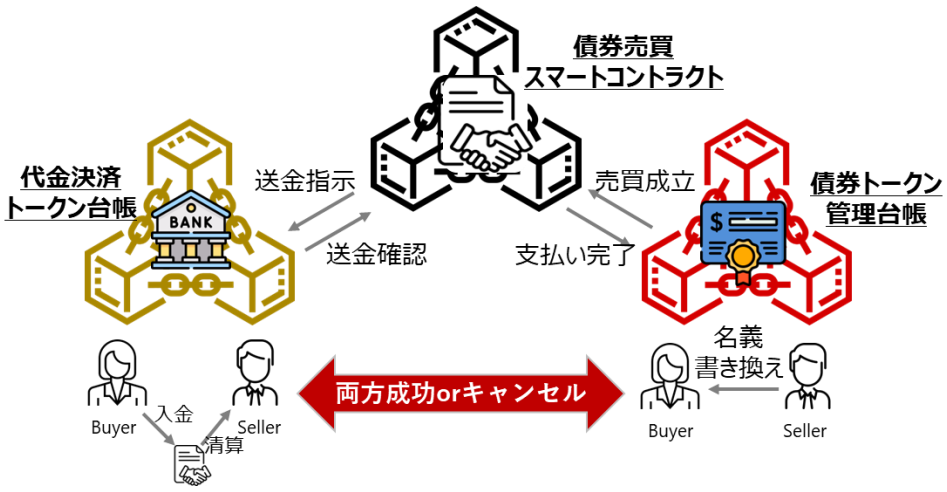


図 4-10：デジタル証券の売買

本実証実験でプログラミングされたシナリオは以下の通りで，シナリオに設定された処理が順次実行される．

1. 譲渡対象となったデジタル証券（セキュリティトークン）の所有権の状態を移転準備中にする
2. 買い手の決済トークン口座から売り手の口座への送金を行う
3. ステップ２の送金が成功したときにのみデジタル証券の所有権を売り手の口座から買い手の口座に移転する．送金が何らかの原因で行われなかった場合には，デジタル証券の所有権の状態を取引開始前の状態へ戻す

この実証実験により，譲渡価格・数量・決済手段についてデジタル証券の当事者間で同意したという前提のもとで，拡張スマートコントラクトを介したユーザ間での直接 DvP 取引が可能であることが検証できた．

第5章 拡張スマートコントラクトのトークン管理への応用

5.1. トークン・エコノミー実現に向けた残課題の整理

異なるブロックチェーン基盤で発行，運営されるトークンの交換や連動操作するための仕組みとして，前章で説明した拡張スマートコントラクトを考案し，その実証システムとしてConnectionChain を実装した．

次なるステップとして，この拡張スマートコントラクトを使い，実社会での経済活動に貢献する真のトークン・エコノミーを実現する，スマートコントラクトベースのトークン取引システムを構築する．

現状トークン・エコノミーを実現するにはさまざまなハードルが存在している.また，現実世界での経済活動には取引に関与する当事者が多数居るので，お金の流れが見えにくくなっている．そこで「2.3.3.1 不動産トークンの運営」で紹介したセキュリティトークン（デジタル証券）のユースケースを参考に，トークン・エコノミーをモデル化し，トークン取引システムに期待される機能を整理することにした．

2.3.3.1 節ではセキュリティトークンのライフサイクルを 5 フェーズに分類していた：

① 運用ルールの策定
② セキュリティトークン化
③ 投資家の募集
④ 投資プロジェクトの運営
⑤ トークンの二次流通

今回のユースケース検討では，セキュリティトークンが独立したブロックチェーン基盤で運営されている現状を踏まえたうえで，トークン発行後に投資プロジェクトの運営で得られた収益を配当として分配や，初期のトークン発行で分譲を受けたユーザが別のユーザに譲渡する二次流通のユースケースなどを追加している.

5.1.1. セキュリティトークンのユースケース分析

運用ルールの策定フェーズ

まず，セキュリティトークンの仕組みを整理する．セキュリティトークンは，発行体の募集に応じた投資家が銀行へ入金を行うと，投資プロジェクトの投資対象となるトークン発行体が，投資された資金を受領した証としてセキュリティトークンを発行する．投資された資金は，投資の募集時に定められた償還期限まで銀行にプールされており，トークン発行体はプールされた資金を元手にして，社員の雇用や設備購入などの営業活動を行い，営業利益を創出する．この利益の一部が投資家へと還元される．

投資家は投資対象とする，投資プロジェクトの情報を持たないので，証券会社などの取引

仲介を受けることが必要になっている．また，投資プロジェクトへの参加で得られる利益は，配当や，運用実績により増減するため，運用実績が悪化した場合や，投資家の都合で資金の引き揚げが必要になった場合などには，投資家同士での譲渡取引が発生する場合もある．

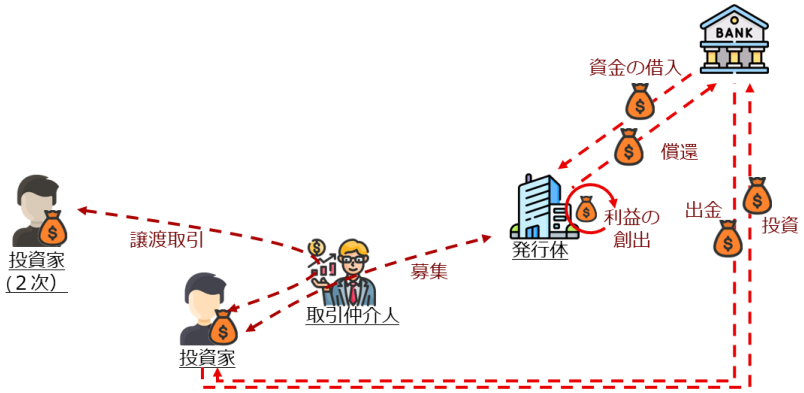


図 5-1：投資予算の裏付けによりトークン発行を許可する

法規制への対処の不可欠である．投資プロジェクトが健全に運営されるためには，投資プロジェクトの運営事業体（＝セキュリティトークン発行体）は，規制当局が求めると AML 対策として投資家の身分を確認する KYC サービスの利用を運用ルールに組み込む必要がある．

セキュリティトークン化

セキュリティトークンの発行では，その価値の裏付けが投資された資金の価値によって担保されるため，当該業務に必要な免許や資格によって，複数のトークンを使い分ける必要がでてくる．

具体的には，投資家から集められた投資資金をプールして管理する銀行は，投資資金の移動を容易するために，銀行は投資家から投資資金の入金を受けると，法定通貨の価値をトークン化した決済トークンを発行し，投資家ユーザの口座へ入金された金額と同量の決済トークンを移転する．また，投資家の権利を保証する証書である証券は，発行体の管理下にあるべきで，こちらは決済トークンから分離されている，セキュリティトークンの一種である「証券トークン（ここでは固有名詞としてこう呼ぶ）」が発行される．

そして，この証券トークンの発行時には，発行時の売り出し価格や償還期間などの運用条件をセキュリティトークン管理システムに通知して，証券トークン発行の準備が整う．

投資家の募集

セキュリティトークン発行の準備が整い，投資家ユーザがトークン管理システムに対して売り出し価格に相当する決済トークンを送金すると，それと引き換えに証券トークンの所有権が投資家ユーザに書き換えられる．

また，5.1.1 で策定される運用ルールの徹底の観点で，投資家ユーザがセキュリティトークンの購入を申し込んだタイミングで， セキュリティトークンを管理するブロックチェーンから見るとオラクルとなる，KYC(Know Your Customer:身元確認)サービスで適格性を審査す

る必要がある．

投資プロジェクトの運営

セキュリティトークンの発行と引き換えに集められた資金の運用を自動化するのは，現時点では現実的ではない．このため，今回検討するトークン取引システムでも，投資資金を引き出し，運営を行う人間が務めるファンドマネージャ(Fund Manager)の存在は不可欠となる．

ファンドマネージャは，プールされている投資資金の一部を運転資金として引き出して営業活動を行い，その活動の成果として得られた利益を資金プールに還元することになる．この入出金は人間が行うものであるから，間違いや不正が起こりやすいので，外部監査での透明性を高めるために，Immutable な台帳に入出金時の状況をできるだけ詳細に記録することが大事である．

また，償還期間が経過したあとの払い戻しに相当する「償還」の手続きでは，証券トークンの発行手順を逆順で実行する形で，償還金の支払いが行われる．具体的には，セキュリティトークン管理システムに，証券トークンの所有権を放棄して，償還をうける手続きを依頼すると，投資プロジェクトの運営結果として得られた配当金が加えられた決済トークンが投資家ユーザの口座へ送金される．ここで注意が必要なのは，投資プロジェクトでは契約上で元本の償還が約束されている場合を除き，償還される決済トークンの量は，投資プロジェクトの成否にかかっており，場合よっては投資した金額より少ない金額が払い戻される可能性があるということである．投資家ユーザが現金を入手するためには，受け取った決済トークンをさらに銀行の換金システムを使って現金の出金を行う必要がある．投資資金は，現金化をすると別の投資プロジェクトへの再投資で決済トークンへの交換が必要となるが，再投資の予定がはじめからある場合には，受け取った決済トークンをそのまま別の投資プロジェクトへ投資することで，換金と再投資にかかる時間を節約できる可能性がある．

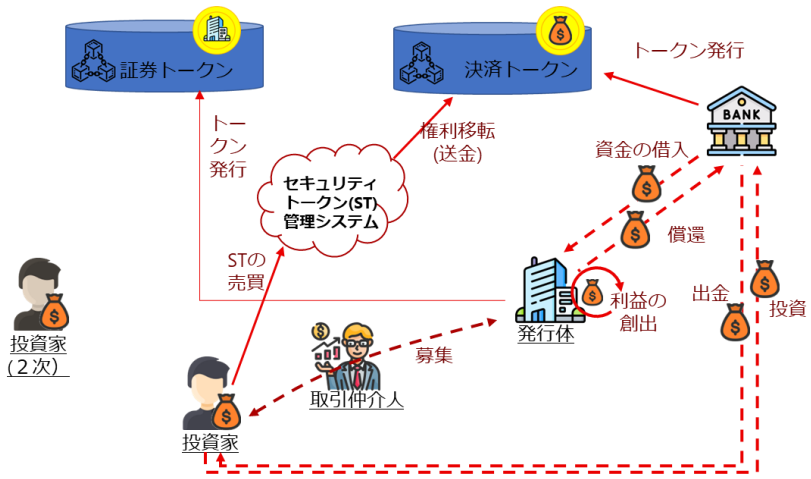


図 5-2：トークン発行体が決めた価格で証券を譲渡

投資プロジェクトにおけるトークン・エコノミーの将来像として，配当金の途中配分が考えられる．投資プロジェクトにおける証券の償還期間は通常数年間となっているので，投資

家は投資期間中まとまった金額の投資資金を長期間活用できず不便である．一方，現実世界での投資プロジェクトの一種である株式投資では，半期から一年の期間ごとに運用実績に応じて，投資の結果得られた収益の一部を投資家に還元する配当金の支払いが行われる場合が多い．現状のセキュリティトークン管理システムが，セキュリティトークンの所有権書き換えを唯一のトリガーにして資金の引き出しが行われているので，配当金の支払いを実現できていないが，管理システムがセキュリティトークンを現在所有している投資家を特定できるようになり，また運用実績でえられた収益の定期的な分配ルールを設定できれば，セキュリティトークン管理システムでも，配当金の支払いが可能となり，投資プロジェクトへ参加しようとする投資家への資金力へのハードルを下げることができると考えられる．

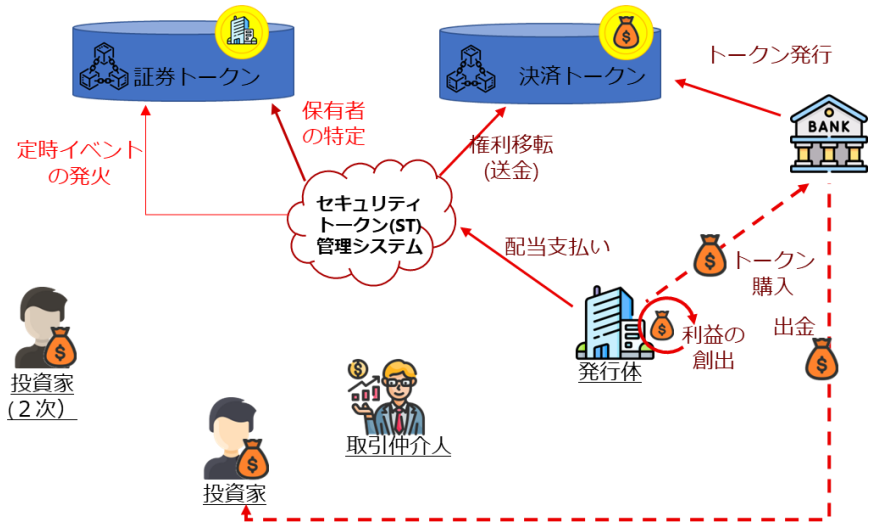


図 5-3：定期的な配当の分配

二次流通

先に述べたように，投資プロジェクトの償還額は，運用実績に依存するので，投資家はプロジェクトへの参加継続をあきらめて，投資した資金を引き揚げることを検討するかもしれない．プロジェクトからの離脱を判断する投資家も居れば，別の投資家の業績見込みとし今後の運用実績が現状より好転するとして新たな参加を希望する投資家も居るはずである．

こうした将来の運用実績への期待の違いが市場価値を生み，証券トークンの途中譲渡，いわゆる二次流通のニーズが生まれる．逆に言えば，現状のセキュリティトークン発行では，投資資金の収集と，償還をトークンで行うことに特化しており，運用実績に応じた償還金の支払いがなされていないので，二次流通のニーズは生まれにくいと考えられる．

また，二次流通のニーズが高まったとしても，投資プロジェクトを開始したときとは違い，プロジェクト開始後の証券トークンの価値は，投資家の現状評価によってばらつきが発生することが予想されるが，現状のセキュリティトークン取引システムでは，トークンの売り手と買い手が価格を交渉した結果合意された金額でセキュリティトークンの所有権を譲渡するような機能が欠けている．

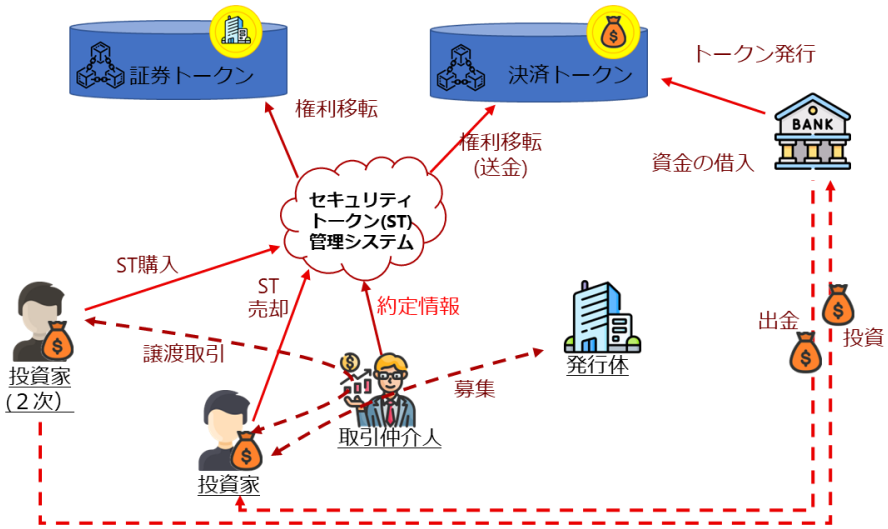


図 5-4：売り手と買い手が合意した価格で譲渡

5.1.2. トークン管理システムの機能要件抽出

トークン・エコノミーのユースケースと，既存金融サービスや関連法令への配慮からトークン・エコノミーに関するデジタル資産取引への機能要件を抽出し，表 5-1 にまとめた．

表 5-1：ユースケースから抽出したトークン取引システムに対する機能要件

ユースケース	機能要件	不足	備考
（全てに共通）	取引履歴の記録	✓	・取引内容（トークン ID や取引価格など）を取引当事者や監査役以外には秘匿したい
運用ルールの策定	トークン運用ルールの設定		・裏付け資産の正当性検証が必要 ・償還が不可能な時期の設定 ・利払いや償還のタイミング
新発債購入	約定情報設定	✓	・トークン発行体が決定することが多い．取引価格は取引システムの外部から与える必要がある
	購入者の本人確認(KYC)	✓	・取引市場に複数のトークン発行体が存在する場合に，プライバシと確実性のバランスに配慮が必要
既発債購入	約定情報設定	✓	・売り手と買い手が市場取引や相対取引などの取引システムの外部で合意した譲渡条件（約定情報）を取引システムへ反映する必要がある
	購入者の本人確認(KYC)	✓	・相対取引での本人確認では取引相手にはプライバシ情報を秘匿
サービス利用料支払い	利用料設定		・交換レートは公開されるが，利用状況は非公開とすべき
セキュリティトークンの利子支払い	定期的な利子の支払い	✓	・デジタルな契約としてトークン発行体とホルダーの双方から中立な形で記録・運営が望ましい
（債権の）償還	債権の破棄		・償還手続きの完了と債権の破棄を連動させる必要がある ・償還可能な金額の決定ロジックには透明性が必要
トークンホルダー優待	優待ルール設定	✓	・トークンを保有する投資家には，ホルダー優待として割引購入や，プロジェクトが関連する別サービスの利用を可能にするユーティリティトークンの発行を行う

上記の機能要件抽出ではトークン・エコノミー特有の機能要件として「取引当事者以外には取引内容の秘匿が必要」と，「監査目的での取引チェック機能が必要」という一見相反した要件が挙がっている．特に，コモディティ，通貨に分類されるトークンが関わるトークン交

換取引については，その市場流動性の高さからトークンのウォレット口座番号が公開されていると，口座に紐づくユーザの購買行動の追跡が可能になるため，ブロックチェーン台帳を利用することの最大のメリットである取引の透明性は担保しつつ，ウォレット口座情報などの個人情報につながる情報は，取引の当事者以外には秘匿する必要があることがわかった．また，さらに，既存のブロックチェーン連携では取引が電子署名のチェックなどの暗号技術でのチェックで十分だったが，トークン・エコノミーのトークン交換では，トークンが持つ特性に踏み込んだチェックが必要であることもわかった．具体的には，セキュリティトークンの発行，運用で，トークン発行体の属する各国の，資金洗浄などの犯罪防止や，外国籍投資家の出資比率への規制などの法令遵守が求められるため，トークンホルダー情報の参照や，記録が必要となることが判明した．さらに，セキュリティトークンの管理業務には利払日や償還日などの，時間と連動したイベントと連動した運用ルールがあるが，既存のスマートコントラクトでは時間連動イベントの処理は苦手なのでその対応も必要となる．

5.1.3. トークン管理システム実現に向けた残課題の整理

5.1.3.1. 抽出した残課題の説明

取引履歴の記録

証券取引での最大の関心事は，その証券をどのユーザがいくらで購入したかという約定情報であり，約定情報から購買傾向が推測されるとフロントランナー攻撃（[35]）を誘引することになるので，売買取引の約定金額は取引の当事者以外には知られてはならない．

その一方で，証券の売買取引で得た利益は多くの国で課税対象であり，利益がでない場合にも税控除を受けることもあるので， セキュリティトークンの取引でも売買金額の証拠を残す必要がある．

約定情報の設定

スマートコントラクトで売買取引を仲介させる場合には，取引価格の情報が不可欠である．ただし，セキュリティトークンを二次流通させる場合には，販売価格がトークン毎に異なるので，販売価格を処理ロジックに反映させる必要がある．

購入者の本人確認

ブロックチェーン・ネットワークでは送受信を行うためのアカウントが比較的容易に作れるので，法定通貨に交換できるセキュリティトークンのホルダーの身元は厳格にチェックすべきである．従来の証券会社では，口座開設のタイミングで身元確認が行えていたが，セキュリティトークンの世界では，ユーザ自身が生成した秘密鍵で所有権の譲渡を受けられるため，口座開設というユースケースが存在しない．

資金洗浄などの犯罪を抑制するためには，公的な機関による身元確認の結果を参照して，取引の可否を判断する必要がある．

定期的な利子の支払い

現在の証券取引では，証券の保管や売買時の名義書換や，発行体へのホルダー情報の通知，など行う仲介者（日本では「ほふり」と呼ばれる）が存在する．セキュリティトークンの世界では，この役割を仲介者なしで実現しようとするものである．

発行体がホルダー情報を必要とするのは，定期的な配当や利子の支払い，株主総会での議決を求めるためである．こうしたホルダー情報は，セキュリティトークンそのものにも記載されないので，セキュリティトークンを所有するホルダーの側から利子受取りの手続きを取るように変更する必要がある．

優待ルール

ゴルフの会員権などのように，購入したセキュリティトークンを保持している間だけ，サービスを利用する権利をトークン化した“ユーティリティトークン”が発行されるサービスを想定した．ただし，ユーティリティトークンの発行は，セキュリティトークンの発行体の意思が強く反映されており，敢えてスマートコントラクトでユーティリティトークンの発行を仲介させる意味があるかには意見が分かれそうだ．

5.1.3.2. 今回対処する残課題

「約定情報の設定」は，証券トークンの売買取引の準備段階であり，価格決定処理と，取引価格に基づいた決済処理に分けられる．後者の決済処理については，拡張スマートコントラクトの先行事例で実現されているため対処は不要と判断した．また，前者の価格決定処理についても，非ブロックチェーンのものも含め，既存の売買マッチングシステムの流用で問題ないと考えられるので，この機能要件は今回の機能拡張の対象から外した．

「優待ルール」については，証券トークン発行体の責任のもとで実施されるサービスの側面が強く，優待サービス利用の結果がブロックチェーンで検証できれば問題ないと考えられ，その運営や優待適用を拡張スマートコントラクト化するメリットは少ない判断し，こちらも今回の機能拡張の候補から除外した．

結果として，「取引履歴の記録」，「（証券トークン）購入者の身元確認」，「定期的な利子の支払い」の３つの手続きに関する課題を解決するための機能拡張を行うことにする．

5.2. 投資プロジェクトの自律的な運営をめざしたトークン管理システムの設計

機能要件分析の結果を踏まえて，トークン・エコノミーの実現に必要な機能要件を備えたスマートコントラクトによるトークン取引システムの提案を行う．

5.2.1. トークン管理システムへの拡張スマートコントラクトの適用

ブロックチェーン取引において，当事者以外に信頼がおける信頼点となる存在がスマートコントラクトである．このスマートコントラクトの処理において，台帳上に記録されていない情報はすべてオラクル情報(3.1.1「オフチェーン情報の取り込み：オラクルの導入」参照)

とみなされ判断の基準に取り込めない．一方で，トークン・エコノミーでの交換取引ではトークンの管理には直接関係しない，台帳外の情報参照や，証跡としての安全な記録メディアが必要である．

これらの機能要件をふまえて，前章で説明した拡張スマートコントラクトを使い，投資プロジェクトの運営を効率化することにした．

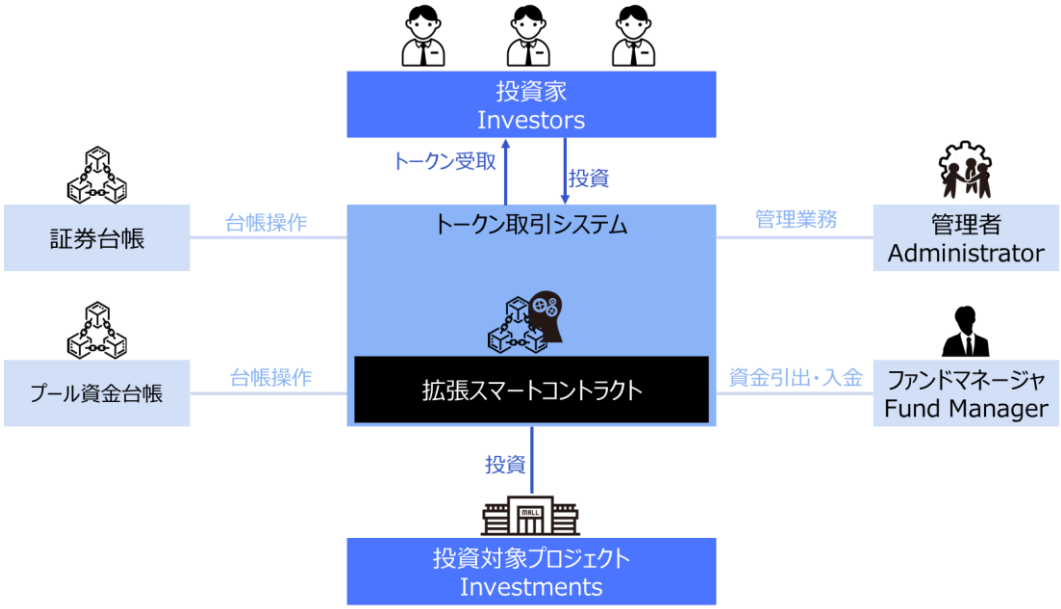


図 5-5：トークン取引システムのアーキテクチャ

投資家の視点では，図 2-12 で紹介した SIDLEY 社のセキュリティトークン取引システムと使い勝手は変わらないが，投資プロジェクトの運営事業体が拡張スマートコントラクトで置き換わっている点が異なる．

SIDLEY 社のシステムでは，資金の流れがシステム内に閉じていてファンドマネージャによる資金運用に不透明さがあったが，今回提案するトークン取引システムでは，投資家から集めた資金をブロックチェーン台帳として構築されたプール資金台帳で残高管理することで，ファンドマネージャの資金引出や運営益の入金などを拡張スマートコントラクトが仲介することで，運営ガバナンスの徹底と資金運用状況の透明性確保が期待される．

資金プール台帳で管理するトークンは，基本投資家がプロジェクトに参加する際に資金を預託するために使われる．この資金プール台帳で管理されるトークンの価値を，法定通貨の預託を信用の根拠とするステーブルコインとして構築することで，投資家は投資プロジェクトへの参加や離脱のタイミングで投資資金の入出金を行う場合に，法定通貨への換金を行うことなく暗号通貨として，受け取った暗号通貨建ての決済代金を別プロジェクトへの投資に流用できるようになり，デジタル完結なトークン・エコノミーが実現できると考えた．

しかし，拡張スマートコントラクトとそのプロタイプ実装である ConnectionChain は，ブロックチェーン連携の汎用プラットフォームとして設計されているため，そのままでは今回の検討で抽出された機能要件のうち以下のような機能要件を満たしていない：

①取引の追跡性に関するプライバシ保護への配慮
②オラクル情報（法令対応に必要な KYC 情報など）を使った取引可否の判断
③時間に依存した取引条件に関する判断

以下の節では，これら 3 つの残課題解決のアプローチについて説明する．

5.2.2. 取引履歴記録に関するプライバシ保護への対処

拡張スマートコントラクトを稼働させるための汎用プラットフォームとして試作した評価システム “ConnectionChain”では，拡張スマートコントラクトの動作を可視化するために，連携シナリオとしてプログラムされた連携先ブロックチェーンで発生する全イベントについて中央台帳に一元管理している．

しかし，セキュリティトークンの取引市場には，ビジネス上の競合プレイヤーも存在することから，取引の事実について第三者による監査性は維持しながらも取引金額などのプライバシ情報は必要以上に公開することは望ましくない．

この問題に対処するために，zk-SNARKs のゼロ知識証明手法を使って，監査での情報開示が必要になったときに開示する取引情報と，証跡として記録されたデータ(コミットメント)との整合性を，取引に関与しない監査人ユーザがチェックできるようにする．

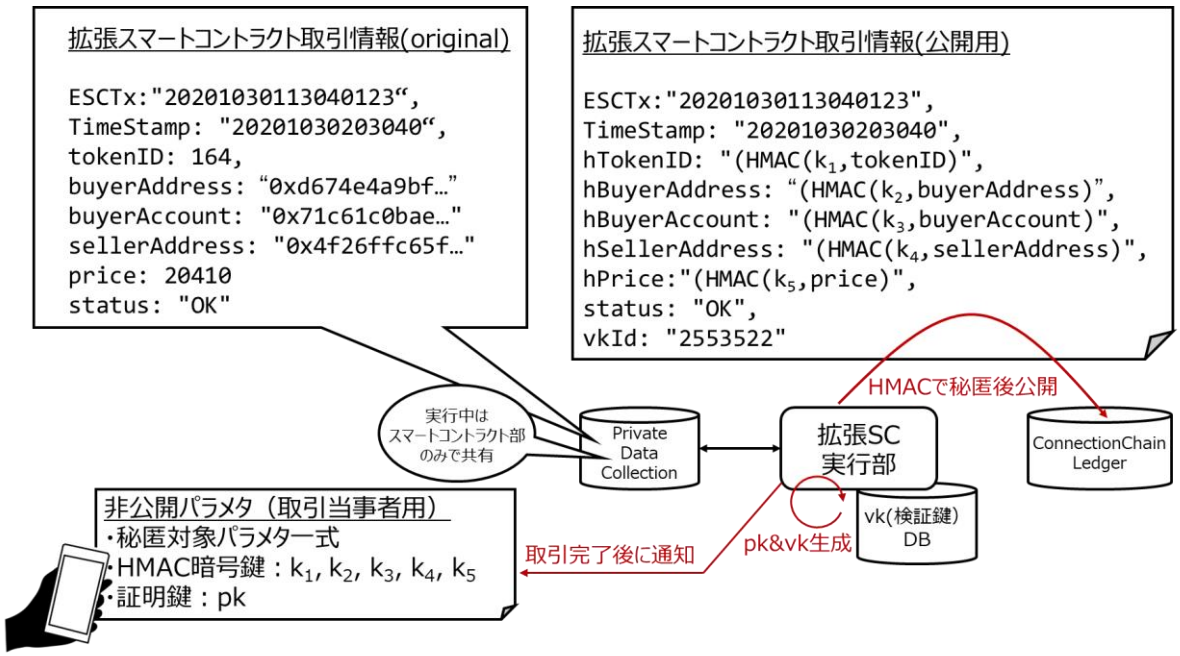


図 5-6：取引証跡の秘匿化処理

図 5-6 に示したように，拡張スマートコントラクト部に改造を加えたトークン取引システムで，シナリオベースの連携処理を司る「拡張 SC 実行部」は，Hyperledger Fabric が提供

する検証ノード間でのみ参照可能な DB，"Private Data Collection"機能を使って，拡張スマートコントラクトのステート遷移をシナリオ実行の妥当性を検証しながら実行する．そして取引が正常に実行された場合にのみ取引の当事者以外には秘匿すべき，売買当事者の ID や約定価格などの情報（図中では秘匿対象パラメタ一式と表現）を，暗号鍵で秘匿化したハッシュ値（HMAC の計算値）に変換された後に ConnectionChain 台帳にトークン売買の取引証跡として記録される．

また，トークン取引内容の証明が後で必要になった場合に備えて，「秘匿対象パラメタ一式」，「HMAC 暗号鍵(k_x)」，「証明鍵(pk)」からなる「非公開パラメタ」が取引当事者に共有される．この証明鍵は，検証鍵とペアになっており，この取引証明における信頼点となる拡張 SC 実行部によって生成されるものである．

取引当事者が納税申告などで取引内容を証明する場合には，先の「非公開パラメタ」を知っているというゼロ知識証明の proof データを作成して拡張スマートコントラクトの取引 ID「ESCTx」の値を添えて証明サーバに提示する．これに対して，証明サーバでは，ESCTx で参照される取引証跡記録トランザクションを ConnectionChain 台帳から検索し，ハッシュ化された値との整合性をゼロ知識証明用の検証鍵 vk を使って検証することで，proof データを提示したユーザの主張が正しい，と判定できる．

5.2.3. オラクル情報を使った取引可否の判断への対処

既存のクロスチェーン技術では，トークン価値の同時交換を安全に行うことに主眼が置かれている．しかし，今後のトークン・エコノミーでは換金を必要としないトークンの交換取引も増えることが予想される．また，トークン・エコノミーをより身近なものとするために，証券会社に口座を開設するような手間もなくしたい．

その一方で，資金洗浄などの犯罪行為を防止するためセキュリティトークン取引システムの利用では，規制当局からの要望として口座開設時だけでなく，トークンの交換取引が発生する度に取引当事者の身元確認を行うニーズもある．

現状のトークン交換システムにおいて自動で身元確認を行える取引システムは存在しないが，将来を見据えた仕組みとして，トークン取引のシナリオにブロックチェーンベースの分散ＩＤ管理システムとの連携機能を組み込んで，取引の都度身分確認が自動で行えるようにした．

この機能拡張には，3.2.3 節で紹介したブロックチェーン基盤ソフトウェア"Hyperledger Indy"の VC ベースの分散ＩＤ管理システムと連携動作させるようにした．図 5-7 に分散 ID 管理システムとの連携で KYC 検証を行う場合の処理シーケンスを示す．

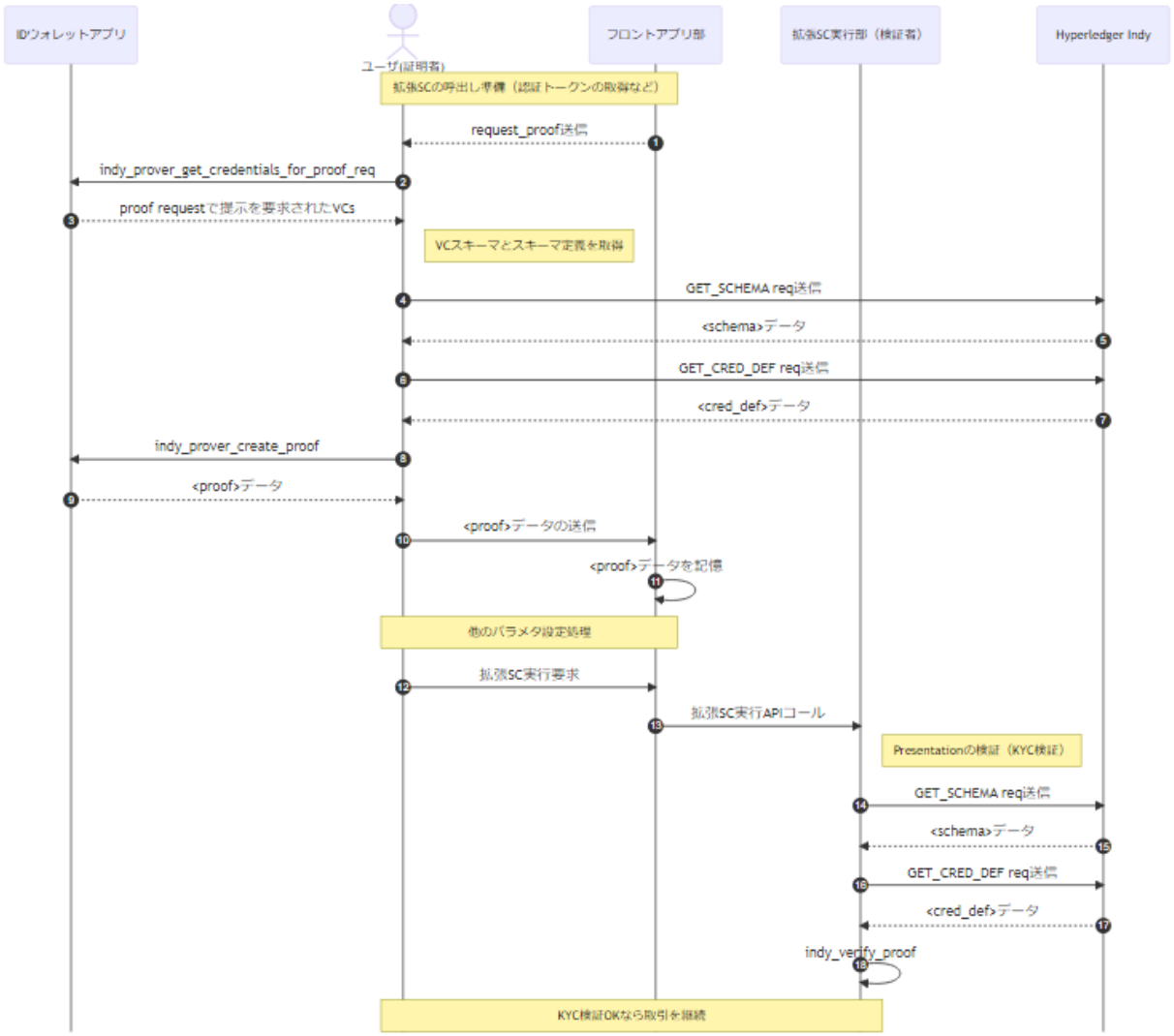


図 5-7：KYC 検証機能の動作シーケンス

5.2.4. 時間に依存した取引条件に関する判断への対処

セキュリティトークン運用のユースケースでは，利益の配当や利子の配布などのトークン発行後の運用管理サービスへのニーズが高いとされているので，トークン管理システムでもユーザや管理者の操作を不要とした取引の自動実行で，自律運用化実現することを考えた．

しかし，既存のクロスチェーン技術では，時間の同期について信頼性が担保されないノードでのスマートコントラクト設計のベストプラクティスとして，時間に関する処理ロジックを避けるべきとされてきた．

スケジュール実行を実現する手段として以下のような機能強化を行った：

1. 事前に設定された日時に取引を実行するためのスケジュール実行取引シナリオを追加
2. スケジュール実行取引シナリオで指定可能なシステム変数として，“実行日時”を用意
3. サブシステムとしてスケジュール実行サーバ(Scheduled Task Server)を置き，証跡台

帳で実行日時を指定する取引が合意されたときに，実行日時と当該取引への参照リンクを紐づけた“タスク情報”を登録可能にする

4. スケジュール実行サーバ上の時計でタスクに含まれる実行日時を過ぎた場合には，当該サーバからスケジュール実行取引シナリオを起動する API コールが発行される．なお，API コールにはスケジュール実行サーバが付与する電子署名とタイムスタンプが含まれる．
5. API コールで呼び出されるシナリオでは，証跡台帳に記録されている実行日時を指定する取引を参照し，スケジュール実行サーバが送信した電子署名やタイムスタンプを使ってシナリオ実行の妥当性をチェックし，OK ならシナリオに定義されている取引を自動実行する．

このようにして，拡張スマートコントラクトとスケジュール実行サーバとの連携で，従来は実現が難しかった，投資家への配当支払いが自動化できるようになった．

5.3. スマートコントラクトベースのトークン取引システムの運用

ConnectionChain がもとから備えていた機能に加えて，投資プロジェクトの自律運営化実現のための機能拡張を行った，トークン取引システムについて説明する．

5.3.1. トークン取引システムの構成

トークン取引システムを図 5-5 のアーキに沿って ConnectionChain を使って構築した．

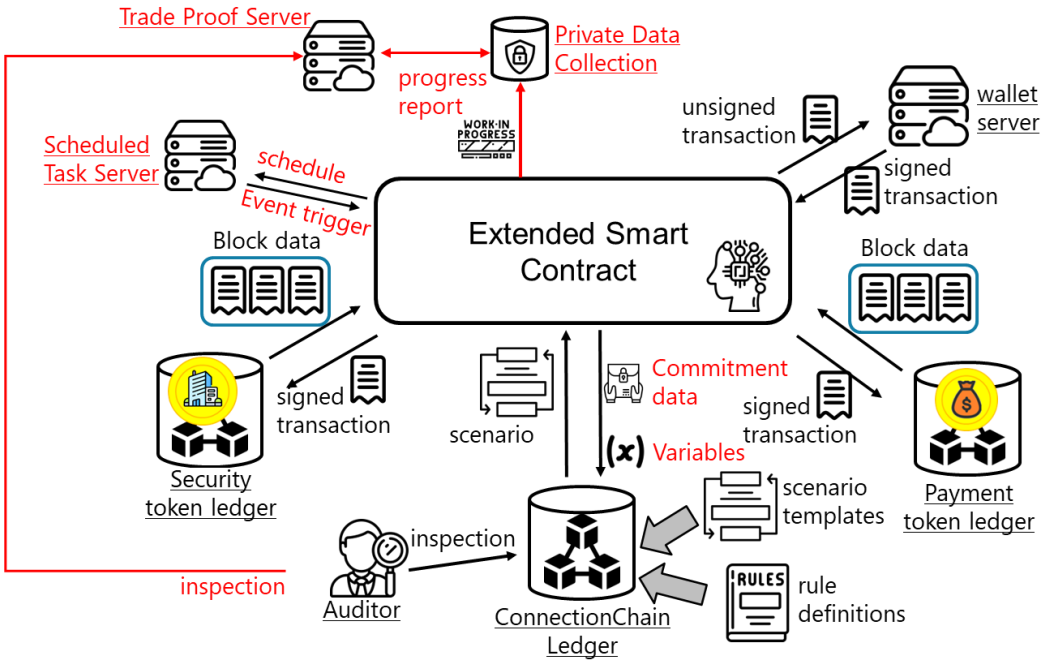


図 5-8：トークン取引システムの構成図

セキュリティトークンのライフサイクルを参考に，本論文で提案するトークン取引システムの機能を説明する：

① 運用ルールの策定
② セキュリティトークン化
③ 投資家の募集
④ 投資プロジェクトの運営
⑤ トークンの二次流通

5.3.2. トークン取引システムの機能

5.3.2.1. 運用ルールの策定

システム変数の設定

セキュリティトークンとして発行される，証券トークンでは運営ガバナンスを徹底するため，運用ルールを拡張スマートコントラクトでプログラミングする．

運用ルールには，募集時の証券トークンの分譲価格，集められた投資金を運用するファンドマネージャが資金を引き出す上限額，配当支払いの条件（支払い日時や配当額の決定ロジック），などが含まれ，管理者が証跡台帳（図中では ConnectionChain Ledger と表記）に．シナリオ・ロジックとシステム変数の組み合わせで設定される．

5.3.2.2. セキュリティトークン化

決済トークンの払い出し

証券トークンの購入に先だって投資家は投資資金入金のために法定通貨を決済用の決済トークンに交換する．

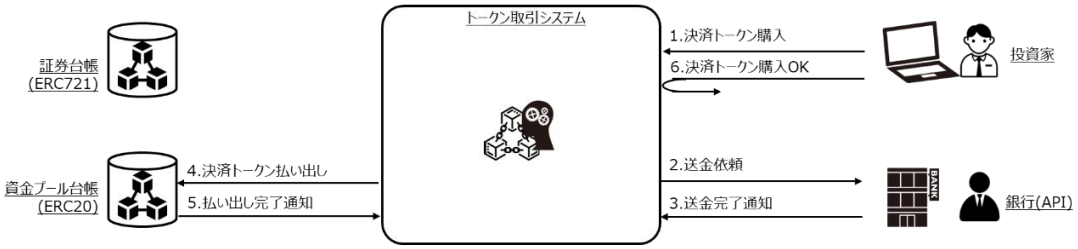


図 5-9：決済トークンの払い出し

証券トークンの払い出し

投資家の募集前に証券トークンを発行する場合には，証券トークンの発行と交換する対象は特に存在しない．しかし，投資プロジェクト運営の透明性を図るため，証券トークンの発

行は取引として可視化されることが望ましい．

今回のトークン取引システムでは，ERC721 コントラクト仕様ではオプションとなっている_mint()（英語で mint は鋳造するの意）API を使って発行体が用意した証券トークンのプール口座に事前にアナウスした数量のセキュリティトークンを発行する．

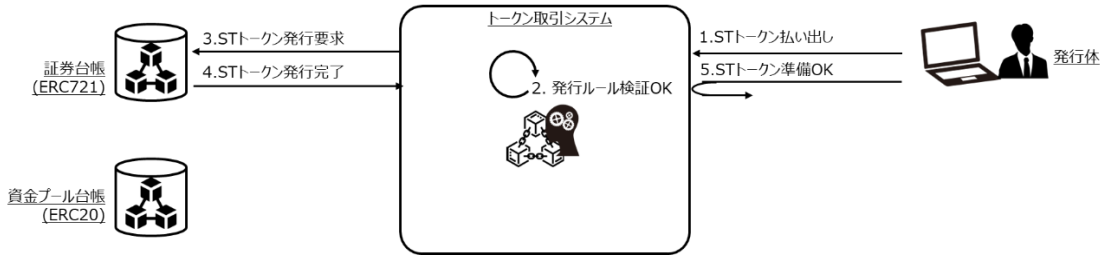


図 5-10：証券トークンの払い出し

5.3.2.3. 投資家の募集

証券トークンの購入

投資家が投資プロジェクトに参加しようとするときには，証券トークンの払い出しで発行された「既発行証券トークンの購入」で，証券トークンを入手する．トークン発行体が新規で投資を募集する際には，証券トークンの分譲価格がシステム変数としてプログラムされた DvP 決済シナリオが実行され，投資家ユーザの資金口座にプールされている資金プール台帳上の決済トークンの残高と引き換えに証券台帳上の証券トークンの所有権が移転される．

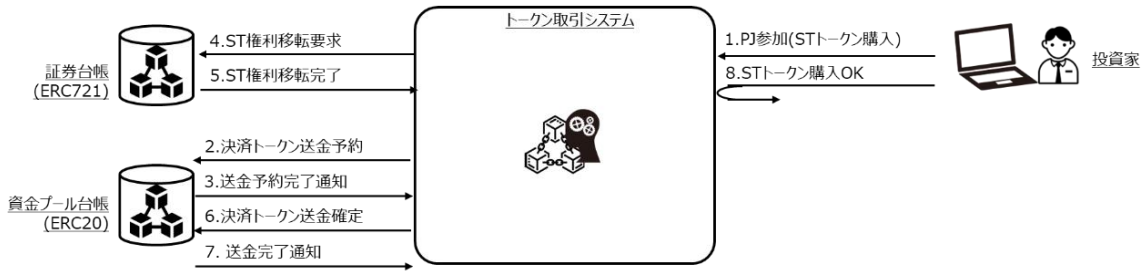


図 5-11：証券トークンの購入

5.3.2.4. 投資プロジェクトの運営

投資資金の運用

投資家が投資した資金はプロジェクトで管理される資金プール口座に集められる．この資金を運用するファンドマネージャは，出金の手続きを取ることでプールされている資金の一部を引き出せる．

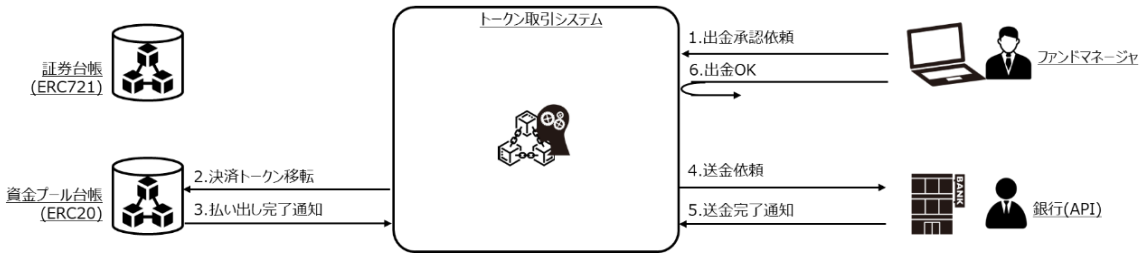


図 5-12：プロジェクト運営のためにプール資金を出金

配当支払い

現実社会での長期にわたる投資プロジェクトでは，運用実績に合わせて中間配当が分配されることが多い，しかし，既存のトークン取引システムでは，主に決済手続きのみをカバーするものであった．今回提案するトークン取引システムでは，スケジュール実行サーバとの連携で，運用ルール設定でプログラムされた決算日に，ほぼ自動で配当が分配されるようにした．

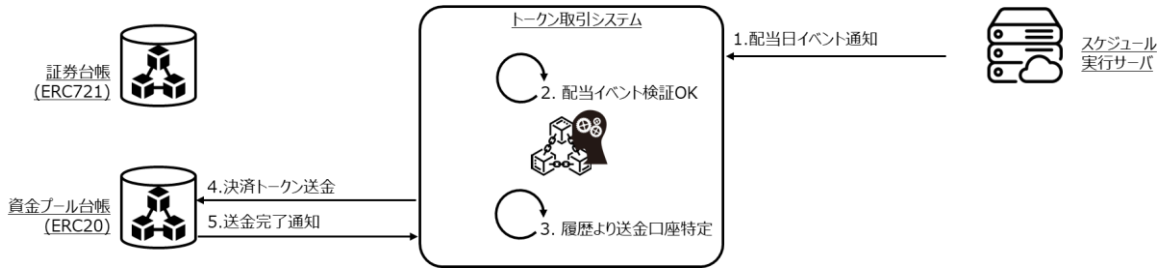


図 5-13：配当の自動支払い

償還手続き

投資プロジェクトの終了時には，プログラミングされた運用ルールに従った分配ルールに沿ってプールされていた資金が証券トークンを保有していた投資家に分配される．

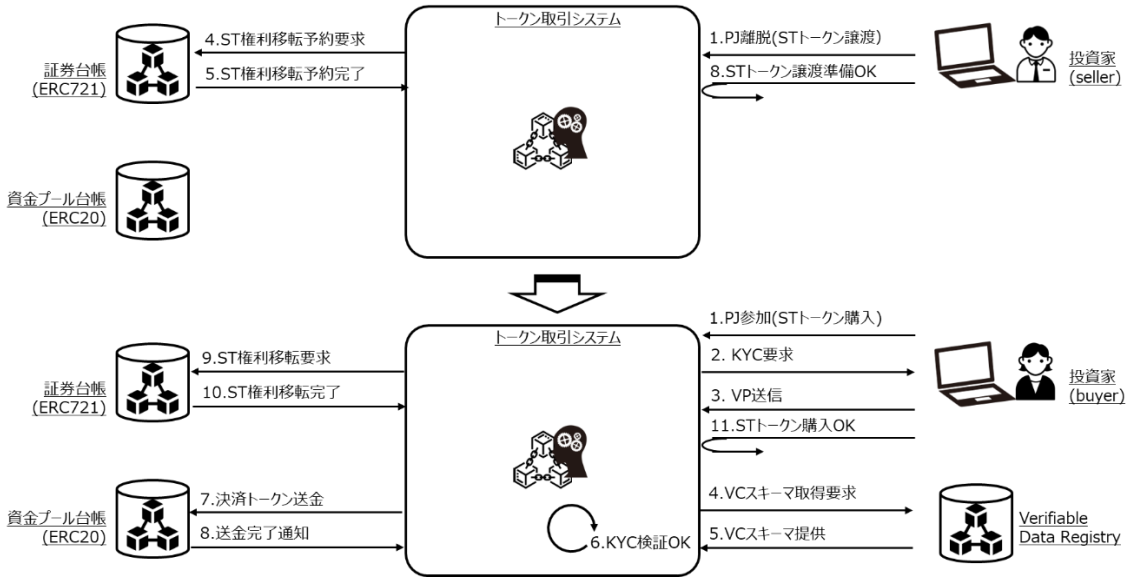
5.3.2.5. トークンの二次流通

ユーザ間の直接売買

投資プロジェクトの運用成果の成否で配当や償還金額が変わることから，投資家はプロジェクト開始の脱退や，参加を希望することが予想される．しかし，プロジェクトの途中で証券トークンが追加発行されることは稀であるため，既発行の証券トークン保有者から譲渡を受けることになる．この既発行の証券トークン譲渡で，仲介者を介すると人件費がかさむので，提案するトークン取引システムでは，発行体の関与を排して投資家同士が直接証券トークンの売買を行えるようにした．

ただし，セキュリティトークンの運用で規制当局から義務付けられる，新たな投資家の身

元確認はエンドユーザである現投資家では行うことができない．そこで，VC ベースの分散 ID 管理システムを導入し，住所や氏名などの個人情報を取引相手に知らせずに，身元確認が行えるようにした．



5.4. まとめ

拡張スマートコントラクトの実行基盤である ConnectionChain に，セキュリティトークンのライフサイクルに即した機能拡張を加えることで，セキュリティトークンを導入しても効率化が図れていなかった投資プロジェクトの運営や二次流通などの，トークン発行後に発生する様々な手続きを自動化することに成功した．

第6章 拡張スマートコントラクトに関する考察

本研究では，異なるブロックチェーンで管理されているトークンを，連携させることで真のトークン・エコノミーを実現する「拡張スマートコントラクト」を提案し，評価システム ConnectionChain システムを試作した．

本章では，この研究の意義をあきらかにするための考察を行う．

6.1. 有用性についての考察

評価システム ConnectionChain の有用性についての複数の観点から評価する考察を行う．

6.1.1. 目的適合性に関する評価

ConnectionChain がトークン交換システムとして目的を満たしているかを評価するために 3.1.2 に挙げたクロスチェーン技術との比較を行った．

表 6-1：機能面での比較

	ConnectionChain	Atomic Swap	Polkadot	Plasma
異なる DLT 基盤間の連携	○	△	○	△
複数ステップからなる取引	○	△	×	×
非通貨(NFT 等)が扱える	○	×	○	×
Private-chain と接続可能	○	×	△	×
取引手数料の安定性	○	×	×	△
運営ガバナンスの透明性	△	○	△	○
連携取引が確定する速度	○	△	○	○

プライベートチェーンやコンソーシアムチェーンの増加により，異なる DLT 基盤間の連携機能へのニーズが高まっている．この点でビットコイン系ブロックチェーンに適用が限られる Atomic Swap や，イーサリウム系に適用が限られる Plasma に対して，ConnectionChain では，OSS である Cactus による台帳操作の抽象化により，基盤ソフトの種類が全く違うブロックチェーン同士を連携させることができることは有利である．

また，複数のブロックチェーンが関与する取引では，一部の取引が成功しないケースが増えるので，取引全体のキャンセルなど複数ステップからなる取引への対応が重要になる．ConnectionChain では，定型シナリオでエスクロー取引に対応しており，通常は取り戻しが不可能なブロックチェーンでの台帳取引で，資産の預託を利用して疑似的に元の状態へと復

元することが可能になっている点は機能面での大きな利点となっている.

一方，コンソーシアムチェーンを利用している ConnectionChain では，パブリックチェーンのような分散型の運営ガバナンス確保のレベルに及ばない．しかし，セキュリティトークンの取引システムなどでは，システムの運営を担うコンソーシアムの構成メンバーを免許や資格を持ち社会的信用を得ている企業体が務めることにより，運営ガバナンスの弱点は許容範囲とみなすことができると考え，利用価値が高いと判断した.

6.1.2. 拡張性に関する評価

トークン・エコノミーでは，トークンが持つ価値を多面的に捉え，別の価値を持つトークンへ変換・交換する経済活動ができることが期待される．このため，そのためのトークン取引システムには，ユーザや取引サービスの事業者の求めに応じて，取引対象となるトークンの種類や，価値判断のロジックにおける拡張性が重要である.

Atomic Swap は，HTLC(Hashed Time-Lock Contract)の仲介により，当事者間で等価とみなされる 2 つのデジタル資産を同時に交換する技術である．Atomic Swap では，互いが預託する資産の価値が等価であることがセキュリティの前提となっているため，この前提が成立しないブロックチェーン連携には適用できない．具体的にはサービス利用への従量課金は実現できない.

Polkadot は，パブリックチェーンとして運用されている「リレーチェーン」が，互いに信頼関係にない連携先ブロックチェーン「パラチェーン」とのブリッジ機能を提供し，同士を連携させる技術である．リレーチェーンの役割をパラチェーンの信頼性を保証することに絞っているため，より高度な判断を行う外部システムを取り込むことで，機能拡張が容易になっている．つまり，ブロックチェーン基盤の違いによって生じる接続性の問題は解決しうるが，トークン取引のような上位サービスの提供はできない.

Plasma は，サイドチェーンでトランザクションを処理して，最終的なデータだけをメインのブロックチェーンに記録する技術である．サイドチェーンで動作する Plasma Contract は，イーサリウムにあった処理能力の限界や，ガバナンスの問題から解放され，DApp を実現できる可能性があるが，決済手段がブリッジ・コントラクトに預託されるネイティブ・トークン ETH に限定されてしまうため，多様な価値観を許すようなトークンの運営は難しい.

これらの先行技術に対して， ConnectionChain は，エスクロー取引機能を含む，動作シナリオのカスタマイズで，Atomic Swap では不可能な 3 つ以上のブロックチェーン操作の連携や，連携処理の一部でエラーが発生した場合の取り戻し処理が可能な点で優位となる．また，連携に関する台帳操作を連携専用のブロックチェーン台帳に逐次記録することで，Polkadot や Plasma と同様の汎用性のある証跡記録機能を持つ．さらに，処理ロジックのなかに拡張スマートコントラクトの管理者から与えられるシステム変数や動作シナリオを介してオラクル情報の取り込みができるため，Polkadot や Plasma では実現できない処理ロジックの高度化や自動化で優位となる.

ところで，Polkadot は，外部の独立したブロックチェーンとの連携機能を持つが，開発者

が Polkadot 専用のアダプテーション機能（具体的には Validator と Collator）を実装する必要がある．これに対して ConnectionChain は，Hyperledger Cactus の開発成果を外部 DLT 基盤との DLT 遠隔操作サーバとして活用しており，Hyperledger の OSS 開発者コミュニティの力を借りて，新たな DLT 基盤ソフトに自動的に対応することが可能となっている．

これらの差異化機能は，ConnectionChain を使った複数の実証実験で実際に利用されており，異なる産業分野や，用途で有用であった実験参加者より高い評価を得ている．

6.1.3. 操作性に関する評価

ブロックチェーン利用のデメリットとして，利用上のリスクや不便への対処がユーザの自己責任である点がよく指摘される．ブロックチェーンは非中央集権な形で運営されるが，参加ノードを構築・運営するには秘密鍵の厳重な管理や，常時のインターネット接続が必要な P2P ネットワークへの参加，などのハードルが高いため本当の意味でブロックチェーンに参加しているユーザは少なく，大衆が利用法に慣れ親しんでいる Web サービス経由で利用している．

この現実を踏まえ，ConnectionChain ではブロックチェーン・ネットワークの特長（参照 2.1.4 ブロックチェーンの特長 ）を尊重しながら，ユーザの不便を補い操作性を向上させる以下の機能を備えた Web サービス（API ホスティング）を提供している：

秘密鍵の管理（ウォレットサービス）

ブロックチェーンの台帳操作では，Tx データへの署名のみが認証・認可の基礎になっている．一方，現実問題として，秘密鍵の紛失や，オフライン中の取引承認などに配慮すると，ユーザ自身が秘密鍵を管理すると不便が多いため，サービス事業者の多くでユーザの秘密鍵を預かる運営を行っている． この運営には取引を仲介するサービス事業者がユーザの秘密鍵まで預かると，サービス運営者への権限委譲の度合いが大きすぎるとの批判もある．

ConnectionChain のサービス事業者に運営が任される，フロントサービス部（図 4-5 参照）では，ユーザ口座情報としてユーザ鍵を管理し電子署名の付与を代行する署名サーバを登録できるようにした．

この署名サーバは，連携先ブロックチェーン毎に ConnectionChain のフロントサービス部とは独立して運営されるアーキになっており，外部のサーバやユーザ端末上のプロセスなどのシステム構成上の選択肢を用意して，運営ガバナンスへユーザが積極的に関われるように工夫されている．

シナリオベースの非同期処理

ブロックチェーン台帳の操作は，Tx データの送信後，参加ノードによるコンセンサス・アルゴリズムで Tx データの正当性検証を経て確定する．この検証に要する時間はブロックチェーン・ネットワークの運営ポリシーに依存しており，数秒から 1 時間程度と幅がある．

ConnectionChain の拡張スマートコントラクトの実装では，拡張スマートコントラクト実

行部（拡張ＳＣ実行部）がシナリオベースの内部状態管理を行い，台帳操作確定やエラーの発生に，ユーザがオフライン中であっても対処できる．この内部状態管理における状態遷移はコンソーシアムチェーン上で稼働するスマートコントラクトで検証・記録されるので，シナリオの実行に関する透明性も確保される．

一方，スマートコントラクトによる検証・記録はデータベース記録に比べて低速であり，シナリオ中に設定された各チェックポイントで，２秒程度のオーバーヘッドが発生する欠点もある．このオーバーヘッドがユーザの利便性に影響することを避けるため，拡張スマートコントラクトの呼出しと，その状態確認を非同期処理として，ユーザが体感する見かけ上の待ち時間を数秒以内にできている．実際の処理時間はシナリオの複雑さにも依るが，エスクロー型の相対取引であれば，連携先ブロックでのブロック確定間隔にプラス１０秒前後で処理を完了できている．また，拡張スマートコントラクトの処理ロジックは，インフラに使うコンソーシアムチェーンが処理可能な取引数の上限分だけ並行実行が可能で，Hyperledger Fabric では毎秒数百件のトランザクション処理性能があることから，この数に近い拡張スマートコントラクトを並行実行できる．

連携先ブロックチェーンの信頼性担保

ブロックチェーン台帳のシステムでは，システムが適切に管理され，意図しないデータの改ざんや漏洩を防ぐ体制になっていることを内部統制によって担保できなければ，技術的なメリットを生かした適正な運用とシステムの信頼性が保証できない．この運営ガバナンスに関する問題は，複数のブロックチェーンを仮想的に一つに統合して運営する拡張スマートコントラクトではより深刻な課題となる．

ConnectionChain では，連携先ブロックチェーンに配備する Validator に署名鍵を持たせ，ブロックチェーン・ネットワーク上で観測される台帳の変化を Validator の署名で確定されることでこの問題に対処した．仮にブロックチェーンにフォークが発生して，送金取引が取り消された場合にも，サービス事業者が設置した Validator の署名で確定させたことを信頼の根拠として扱うので，サービス事業者の責任範囲としてファイナリティの問題にも対処できるようになっている．

以上の機能により，ConnectionChain では，ブロックチェーンの操作性を高めることができている，と評価した．

6.2. システムの安全性についての考察

トークン・エコノミーで取引対象とトークンは，実社会でも資産価値を持つものが多くなることが予想される．ユーザが安心してトークン・エコノミーに参加できることを保証するためには，トークン管理システムの安全性の評価が不可欠である．

6.2.1. 安全性評価の手順について

今回の考察では，アーキテクチャレベルでの脅威の認識を目的とするため，DFD の作図と STRIDE 分析による簡易な脅威モデリング手法を用いることにした．

脅威モデリングは，設計対象のシステムやソフトウェアに関するセキュリティ上の脅威を抽出し，抽出した脅威への対策の要否を判断するために利用される．

具体的な脅威分析の手順は以下のとおり：

1. 分析対象のシステム（今回の対象は ConnectionChain）の DFD を作成，Trust Boundary を記述
2. Trust Boundary とデータフローが交差している箇所が脅威の発生しうる場所と考える
3. データフローの両端の要素 (origin/destination)の相関関係から脅威を列挙する
4. 見つけた脅威を表にまとめる
5. 脅威ごとの深刻度を図り，深刻度に応じて対策を検討する

6.2.2. DFD によるデータフローの整理

システム構成要素と，外部システムがどのようにデータをやり取りするかを可視化するデータフロー図(DFD: Data Flow Diagram)を作成し，ConnectionChain のシステム構成を把握する．

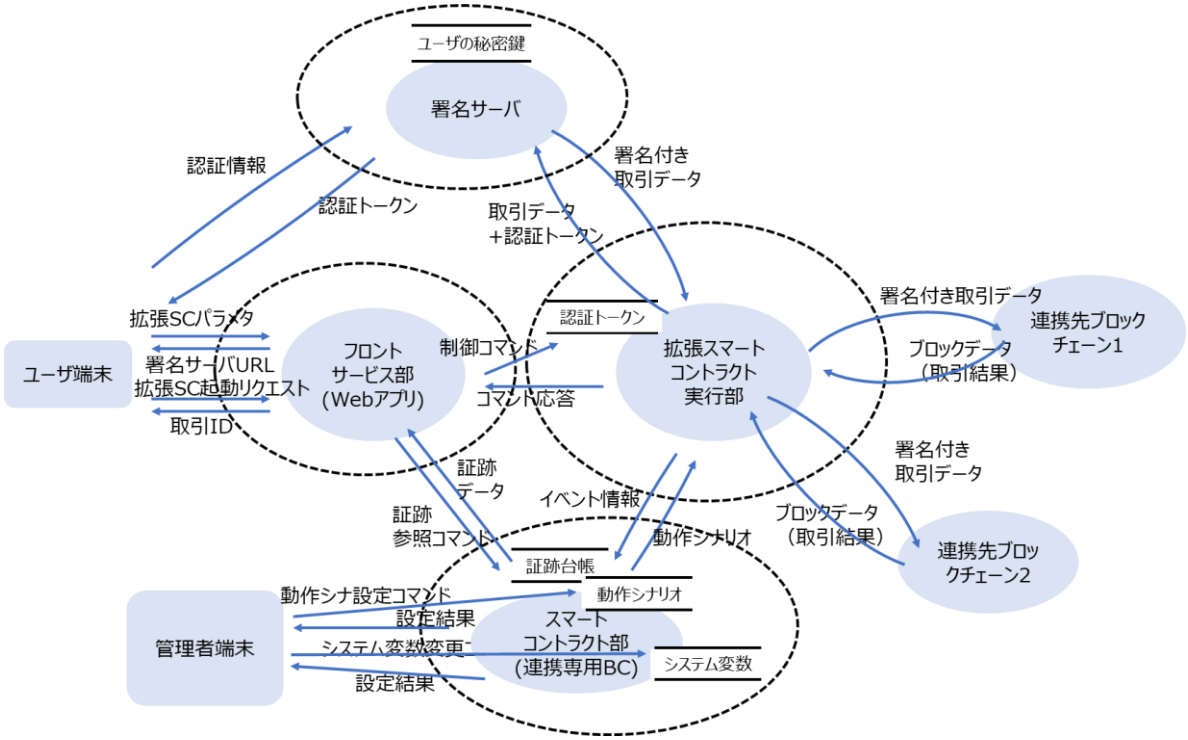


図 6-1：ConnectionChain のデータフロー図

6.2.3. STRIDE モデルを使った脅威の抽出

今回提案したトークン管理システムについて STRIDE モデル(参考：[36])を使った脅威分析を行う.

STRIDE モデルは，Microsoft 社が提唱する脅威の分類方法で，分類名の Spoofing（なりすまし)，Tampering（改ざん)， Repudiation（否認)，Information Disclosure（情報漏洩)，Denial of Service（サービス拒否)，Elevation of Privilege（特権の昇格）の頭文字をとって命名されている．STRIDE モデルは，分類に沿って想定される脅威を列挙していくことで脅威を網羅的に抽出するのに適している.

表 6-2：STRIDE モデルの分類（出典: [36]）

カテゴリー	説明
なりすまし	他のユーザの認証情報 (ユーザ名，パスワードなど) に不正にアクセスし，それを使用する行為など
改ざん	悪意のあるデータの変更など．例としては，データベースに保持されているような永続的なデータに対する許可されていない変更や，インターネットなどのオープン ネットワーク経由で 2 台のコンピュータ間を流れるデータの変更などがある
否認	反証できる関係者がいない状況でアクションの実行を否定するユーザに関連するもの．たとえば，禁止されている操作を追跡できる機能がないシステムでユーザが不正な操作を行うような場合がある．否認防止とは，否認の脅威に対抗するシステムの機能のこと．たとえば，商品を購入するユーザは，受け取り時に署名をする必要がある．販売者は，署名された受領書をユーザが荷物を受け取ったことの証拠として使用することができる.
情報漏洩	情報へのアクセスが想定されていない個人への情報の暴露などがこれにあたる．たとえば，アクセスが許可されていないファイルをユーザが読み取ることができたり，侵入者が 2 台のコンピュータ間で送信されるデータを読み取ることができたりする場合などが考えられる.
サービス拒否	サービス拒否 (DoS) 攻撃では，有効なユーザへのサービスが拒否されてしまうこと．たとえば，Web サーバを一時的に使用できなくする行為がこれにあたる．システムの可用性と信頼性を向上させるために，特定の種類の DoS 脅威からシステムを保護する必要がある.
特権の昇格	特権のないユーザが特権的なアクセスを取得すると，システム全体を侵害したり破壊したりできるようになる．特権の昇格の脅威には，攻撃者が効果的にすべてのシステム防御を破り，信頼されているシステム自体の一部となる，本当に危険な状況が含まれる.

STRIDE による脅威の洗い出しにはいくつかの方法論が存在し，代表的なものには STRIDE-per-Element と STRIDE-per-Interaction がある．STRIDE-per-Element は，DFD の要素すべてに対して STRIDE を適用し，虚位を洗い出す手法であり，STRIDE-per-

Interaction は信頼境界上の脅威を洗い出す手法である．この考察では主にシステム外部からの脅威洗い出しを目的とするので，STRIDE-per-Interaction の手法にしたがって，ConnectionChain における脅威を列挙する.

なりすましに関する脅威

表 6-3：なりすましに関して抽出した脅威

項番	抽出した脅威
S-1	ユーザ端末に対するフロントサービス部（Web アプリ）へのなりすまし
S-2	署名サーバに対するユーザ端末へのなりすまし
S-3	ユーザ端末に対する署名サーバへのなりすまし
S-4	フロントサービス部に対する拡張スマートコントラクト実行部へのなりすまし
S-5	拡張スマートコントラクト実行部に対するフロントサービス部へのなりすまし
S-6	スマートコントラクト部に対する管理者端末へのなりすまし
S-7	署名サーバに対する拡張スマートコントラクト部へのなりすまし

改ざんに関する脅威

表 6-4：改ざんに関して抽出した脅威

項番	抽出した脅威
T-1	拡張スマートコントラクト実行部における認証トークンの改ざん
T-2	スマートコントラクト部におけるシステム変数の改ざん
T-3	スマートコントラクト部における動作シナリオの改ざん
T-4	スマートコントラクト部における証跡台帳の改ざん

否認に関する脅威

表 6-5：否認に関して抽出した脅威

項番	抽出した脅威
R-1	ユーザが拡張スマートコントラクトを利用した事実を否認
R-2	ユーザが連携先ブロックチェーン（1 もしくは 2）で実行された取引を承認した事実を否認

情報の漏洩に関する脅威

表 6-6：情報の漏洩に関して抽出した脅威

項番	抽出した脅威
I-1	署名サーバからのユーザの秘密鍵が漏洩
I-2	拡張スマートコントラクト実行部から認証トークンが漏洩
I-3	スマートコントラクト部から証跡台帳の記載内容が漏洩
I-4	スマートコントラクト部から動作シナリオが漏洩
I-5	スマートコントラクト部からシステム変数の値が漏洩

サービス拒否に関する脅威

表 6-7：サービス拒否に関して抽出した脅威

項番	抽出した脅威
D-1	署名サーバへのサービスへのサービス拒否攻撃
D-2	フロントサービス部へのサービス拒否攻撃
D-3	拡張スマートコントラクト実行部へのサービス拒否攻撃
D-4	スマートコントラクト部へのサービス拒否攻撃

特権の昇格に関する脅威

表 6-8：特権の昇格に関して抽出した脅威

項番	抽出した脅威
E-1	拡張スマートコントラクトユーザがサービス運用者の操作権限を奪取する

6.2.4. リスクの評価

6.2.3 節で抽出した脅威のリスクについて，発生頻度，危害の程度の指標についてそれぞれを 5 段階で評価し，システム運営へ及ぼす脅威のインパクトに応じて A（対策必須），B(要対策)，C（許容範囲）で総合評価した.

表 6-9：リスクの評価基準

発生頻度						
	5：頻発する	C―1	B―1	A―1	A－2	A―3
	4：しばしば発生	C―2	B―2	B―3	A―4	A―5
	3：めったにない	C―3	C―4	B―4	B―5	A―6
	2：可能性あり	C―5	C－6	B－6	B－7	B―8
	1：ありえない	C―7	C－8	C－9	C―10	C－11
		Ⅰ：無傷	Ⅱ：軽微	Ⅲ：中程度	Ⅳ：重大	Ⅴ：致命的

危害の程度（危害の重大性）

A－1～6：対策必須．対策しなければ危険

B―1～8：要対策．対策の検討を行うべき

C－1～11：許容範囲

なりすましに関するリスク評価

表 6-3 に挙げたリスクの評価を行った．

表 6-10：なりすましに関するリスク評価の結果

対象	評価	根拠	備考
S-1	C-3	Web アプリが動作するフィッシングサイトで Web アプリ上のアカウント情報（アドレス帳など）を盗み出す攻撃はありえるが，通常の Web サイトの運営と同様で特別な配慮は不要と判断．	
S-2	B-7	なりすましが発生すると任意の取引を実行できるようになる	PIN による認証は避け，推測困難なパスワードの使用や，多要素認証の導入を推奨する
S-3	C-3	署名サーバを語るフィッシングサイトで認証情報を盗み出す	
S-4	C-6	拡張 SC 実行部は，フロントサービス部がサービス運営側の内部ネットワークを介して通信を行う相手でなりすましが非常に困難	
S-5	C-5	フロントサービス部は，特権持っていないため発生頻度の低さも加味して脅威はないと判断	
S-6	C-6	スマートコントラクト部では公開鍵ベースの厳重な認証を行っており，なりすましはほぼ不可能と判断	
S-7	C-5	正しい認証トークンを持っていない状態での拡張 SC 実行部へのなりすましでは何もできない	

改ざんに関するリスク評価

表 6-4 に挙げたリスクの評価を行った.

表 6-11：改ざんに関するリスク評価の結果

対象	評価	根拠	備考
T-1	C-6	認証トークンを操作する API はないので通常経路での改ざんはありえない．また改ざんが起こった場合でも特定の動作シナリオが異常終了するだけなので危害も軽微と判断した.	
T-2	B-5	システム変数の改ざんは，サービス運営のガバナンスへの影響が大きい．ただし，システム変数の変更にはトランザクションへの公開鍵署名が必要なので発生確率は低いと判断した.	管理者の秘密鍵の厳重な管理と，システム変数の操作ログの監視を推奨
T-3	B-5	動作シナリオの改ざんは，サービス運用のガバナンスへの影響が大きいが，管理体制の強化で対処可能	秘密鍵の厳重な管理と，操作ログの監視を推奨
T-4	C-10	証跡台帳は，本システムにおける信頼の要であり，改ざんが発生した場合の危害は重大である．一方で，ブロックチェーンの特性として改ざんが不可能と謡われていることから正しく構築されたシステムでは発生しない脅威と判断した.	

否認に関するリスク評価

表 6-5 に挙げたリスクの評価を行った.

表 6-12：否認に関するリスク評価の結果

対象	評価	根拠	備考
R-1	C-7	ブロックチェーンに記録されるトランザクションにはユーザの電子署名が残るので，これを否認するのは無意味で危害なしかつありえないとした.	
R-2	C-5	連携先ブロックチェーンでの台帳操作の検証には限界があり，ユーザが取引を否認しうる可能性は残る．ただし，連携先ブロックチェーンの正当性を検証する責任は拡張スマートコントラクト側にはないため，危害はないと判断した.	

情報の漏洩に関するリスク評価

表 6-6 に挙げたリスクの評価を行った.

表 6-13：情報の漏洩に関するリスク評価の結果

対象	評価	根拠	備考
I-1	A-4	ユーザの署名用秘密鍵が漏洩した場合には，連携先ブロックチェーンで管理している資産が失われるなど重大な危害が発生する．ただし，その影響範囲は当該鍵が紐づく口座のみで，システム全体への影響は限定される.	秘密鍵は，最低でも暗号化して保存されるべきである．また，システムへの攻撃者の侵入を許した場合に備えて，署名が可能な状態でも秘密鍵が取り出せない HSM(Hardware Security Module)の導入を検討すべき
I-2	B-4	拡張 SC 実行部は，システムの深部にあり攻撃を受けにくい構成要素であるが，繰り返し利用できる認証トークンが漏洩した場合には，署名用秘密鍵が漏洩した場合と同様の危害が発生する.	認証トークンを主記憶メモリ上でのみ保持するようにするか，二次記憶で保持する場合には暗号化ストレージの使用することを推奨する
I-3	C-1	コンソーシアムチェーンにおける台帳に記載される情報であるので情報が漏洩しても大きな問題にならない.	連携先ブロックチェーンの取引内容によってはプライバシ保護の配慮が必要であるが危害とはみなされないのでここでは無害と評価した
I-4	C-1	拡張スマートコントラクトの動作には透明性が求められるので，その動作シナリオに関する情報が漏洩しても問題ない	
I-5	C-1	拡張スマートコントラクトの動作に影響するシステム変数の公開は，動作シナリオの透明性を確保する一要因となっており，その値が漏洩しても問題ない.	

サービス拒否に関するリスク評価

表 6-7 に挙げたリスクの評価を行った.

表 6-14：サービス拒否に関するリスク評価の結果

対象	評価	根拠	備考
D-1	A-4	署名サーバは，拡張 SC 実行部がユーザから委譲された連携先ブロックチェーンでの取引を実行するために不可欠の構成要素である．署名サーバがサービス不能となった場合には，新規の拡張スマートコントラクトが起動不可となるほか，動作中のシナリオ実行の継続が困難となるため，攻撃を受けた場合のダメージを重大と判断した.	認証トークンの発行と，認証トークンを使った代理署名の 2 つに IF を分けて，攻撃を受けた場合には送信元の限定が難しい前者向け IF へのネットワーク接続を遮断することで，稼働中の拡張スマートコントラクトの実行へのダメージを低減することは可能と考えられる
D-2	A-5	フロントサービス部への攻撃が発生した場合には，攻撃のダメージを低減する手段がほとんどないため，攻撃を受けた場合のダメージを致命的と判断した.	
D-3	C-4	拡張 SC 実行部は，外部ネットワークに直接接していないため，攻撃によるダメージを軽微と判断した.	
D-4	B-6	シナリオ動作の透明性を保証するために，証跡台帳を公開している場合には，影響を受ける可能性がある.	証跡台帳の公開時には，書き込み権限を与える必要はないので，台帳の読み出しだけができるノードの設置を推奨する

特権昇格に関するリスク評価

表 6-8 に挙げたリスクの評価を行った.

表 6-15：特権昇格に関するリスク評価の結果

対象	評価	根拠	備考
E-1	C-1	システム管理者の操作は，スマートコントラクト部で直接実行されるため，署名用の秘密鍵を奪取されない限りは権限の昇格はありえない	システム管理者が，フロントエンドサーバなどを介してスマートコントラクト部を操作する形態ではなりすまし攻撃との併用で致命的なダメージを受ける可能性があるので注意が必要

6.2.5. リスク評価のまとめ

リスク評価の結果では，Ａランクの脅威３件と，Ｂランクの脅威４件を確認した.

特に注意が必要なのは，Ａランクと評価した２つの脅威（I-1 と D-1），Ｂランクと評価した２つの脅威（S-2 と I-2）と抽出した重大な脅威の多くが，署名サーバに関わっていたことである．今回の試作では，署名サーバの実装において想定されるリスクへの対策が十分でないことが判明したので，実運用に供するための署名サーバでは，想定されたリスクに対処するための機能を盛り込む必要があるとわかった.

一方，拡張スマートコントラクトの主機能を担う，拡張ＳＣ実行部とスマートコントラクト部については，Ｂランクの評価となったリスクが見つかったものの，管理者が台帳操作用の署名鍵を適切に扱うことでリスクを低く抑えられることがわかった，これは，ブロックチェーンの分散システムとしての特性を活かした成果であると考えている.

第7章 結論

本論文では，序論に示した論文の構成に則して，第２章から第６章まで研究を進めた結果，以下のような研究成果を得た．

第２章では，ブロックチェーン技術とその応用例について解説した．解説のなかでは，ブロックチェーンの動作原理や，本研究のテーマに深く関わる「契約とその履行条件をあらかじめプログラミングしておくと，契約条件が満たされた際に自動で取引が行われる」スマートコントラクトの概念と導入の意義を説明した．また，暗号通貨で注目されることが多いパブリックチェーン以外の選択肢として，パブリックチェーンと比べて非常に高いデータ処理能力を得ることができるプライベートチェーンやコンソーシアムチェーンを活用することが重要であることを確認した．そして，暗号通貨以外のブロックチェーンの有望な使途として，トークン化を取り上げた．トークン化は，NFT アートが高値で販売されるなど話題を呼んでおり，その取引を容易にするための仕組みとして，ERC20 や ERC721 といった管理用スマートコントラクトの標準化が進んでいると，という業界動向を説明した．トークン化の先行事例として，セキュリティトークンを発行することで投資を募る STO の現状と課題をまとめたレポート「REAL ESTATE TOKENIZATION」について，同レポートが提案する，セキュリティトークンの５段階のライフサイクルを中心に紹介した．

第３章では，本研究に関連する先行研究や取り組みについてのサーベイを行った．サーベイの対象として，私製トークンの発行を行うプライベートチェーンやコンソーシアムチェーンのエコシステムをつなぐことを目的とした「インターオペラビリティ」の技術と，「トークン・エコノミーに関連するブロックチェーンの技術動向」を選んだ．インターオペラビリティに関する研究サーベイでは，未だ暗号通貨を対象にした技術が多勢を占めているものの，Polkadot のように金銭的な価値以外の情報を管理するブロックチェーンをパブリック，プライベートの区別なく接続しようとする新しい流れがみられた．一方，ブロックチェーンの技術動向として，ゼロ知識証明の技術を使いながら，匿名送金，本人確認，などの使い方が異なる事例を紹介し，その活用の可能性を確認できた．また，DEX やプログラマブル・トークンといった先行事例の調査で，スマートコントラクトの自律動作のメリットを引き出すことにトークン・エコノミーの潜在的価値があるとわかった．

第４章では，本研究の目的である，ブロックチェーン技術を用いたトークン・エコノミーの実現を目指して，取引の自動化に欠かせないスマートコントラクトの考え方や機能を踏襲しつつ，他のブロックチェーンで管理されるトークンやオラクル情報を利用できるような機能拡張を施した，「拡張スマートコントラクト」を提案した．提案した拡張スマートコントラクトは，スマートコントラクトの考え方や機能を踏襲し，外部のブロックチェーン・サービスを統合する際，統合対象となるブロックチェーンへの取引発行や，ブロックデータのモニターとその結果の記録が行えるように拡張しており，連携専用のコンソーシアム型ブロックチェーン上で動作する．また，提案した拡張スマートコントラクトの有用性を評価するための評価システム ConnectionChain をコンソーシアムチェーン Hyperledger Fabric と，OSS のブロックチェーン連携ツール Hyperledger Cactus を使って設計，実装した．また，拡張ス

マートコントラクトの有用性を示すため，ConnectionChain を使って行った実証実験の成果についても報告した．

第５章では， ConnectionChain をさらに機能拡張し，実社会での経済活動に貢献する真のトークン・エコノミーを実現する，スマートコントラクトベースのトークン取引システムを構築した．機能拡張の候補選定にあたり，セキュリティトークンのユースケースを分析し，投資プロジェクトを自律運用するトークン取引システムの実現に必要な機能要件を抽出し，元の ConnectionChain だけでは不足する機能を特定した．この不足機能は，「取引履歴の記録」，「（証券トークン）購入者の身元確認」，「定期的な利子の支払い」で，ゼロ知識証明などブロックチェーン関連の新技術を積極的に取り込んで，これからの残課題を解決した．

第６章では，本研究での提案したコア技術である拡張スマートコントラクトを有用性と安全性の面で評価した．有用性の評価は，目的適合性，拡張性，操作性について評価し，いずれの項目でも比較対象より優れた評価結果をだすことができた．一方，安全性の評価では，アーキテクチャレベルでの脅威の認識を目的とした，DFD の作図と STRIDE 分析による簡易な脅威モデリング手法を用いて行った．リスク評価の結果としては，対策必須となる A ランクのリスクが１件見つかった．情報漏洩に関する脅威「I-1: 署名サーバからのユーザの秘密鍵が漏洩」で，ConnectionChain システム本体に関する脅威ではないが，署名サーバの利用はサービスの利便性維持のためには不可欠な要素であるので，秘密鍵の漏洩を防ぐ HSM(Hardware Security Manager)の導入を検討すべき，とわかった．B ランクのリスクも４件見つけたが，操作ログの監視など一般的な IT システム向けのセキュリティ対策が本システムの運営でも有効とわかった．

本論文を通して，ブロックチェーンのスマートコントラクトの可能性を最大化する「拡張スマートコントラクト」の有用性と，将来にむけた可能性を確認することができた．

対外発表論文

本論文の主要部分は下記参考論文の(1)〜(3)として公表済みである.

参考論文

(1) S.Fujimoto, and T.Takeuchi, Y.Higashikado: “Secure Blockchain Interworking Using Extended Smart Contract,” IEICE Transactions on Information and Systems, 2022, vol. E105.D, Issue 2(2022), pp. 227-234.

(2) S.Fujimoto, and K.Omote, “Proposal of a smart contract-based security token management system,” in 2022 IEEE International Conference on Blockchain, Aug. 2022, pp. 419–426.

(3) S.Fujimoto, Y.Higashikado and T.Takeuchi, “ConnectionChain: the Secure Interworking of Blockchains,” in 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), 2019, pp. 514-518

査読付き発表論文

(1) S.Fujimoto and M.Takenaka, “Adoption of the IPsec-VPN for the ubiquitousnetwork,”International Symposium on Applications and the Internet (SAINT'06),2006, pp. 77-81.

査読のない発表論文

(1) 藤本真吾，面和成： スマートコントラクトによるデジタル資産取引におけるプライバシに配慮した取引仲介の実現に向けて, 第 95 回コンピュータセキュリティ研究会（CSEC），2021 年 11 月

(2) 藤本真吾，鎌倉健：ブロックチェーンの安全な連携方式の提案, 2018 年暗号と情報セキュリティシンポジウム(SCIS2018), 2018 年 1 月

(3) 藤本真吾，小暮淳：ブロックチェーンを安全につなげるセキュリティ技術「コネクションチェーン」, 雑誌富士通 vol.70, No.2, 2019 年 4 月

(4) S.Fujimoto and J.Kogure, "ConnectionChain: Security Technology for Securely Linking Blockchains", FUJITSU TECHNICAL JOURNAL vol.55, No.5, 2019

(5) 小櫻文彦，藤本真吾，野村佳秀，山下一寛：ブロックチェーンの信頼性を向上する脅威分析手法およびスマートコントラクト検証，雑誌富士通 vol.70，No.4, 2019 年 9 月

謝辞

本論文をまとめるにあたり，指導教官である筑波大学 システム情報系 教授　面和成博士には共同研究を開始してから今日に至るまで，真摯な傾聴と示唆に富んだご助言で，また時には情熱に溢れた議論を行うことで，研究活動を導いてくださいました．ここに深甚なる謝意を表します．

本論文の論文審査でご指導を賜りました，情報セキュリティ大学院大学　教授 大塚玲博士，筑波大学 システム情報系 准教授　木村成伴博士，同准教授　片岸一起博士，同准教授　西出隆志博士に厚く御礼申し上げます．

自己啓発の一貫としての筑波大学大学院への通学を許可し，研究活動を応援してくださった富士通株式会社 富士通研究所 研究本部 データ＆セキュリティ研究所の上司や，同僚の方々に感謝いたします．

研究活動の一環として検証システムの試作，実証実験の実施で利用したブロックチェーン関連のオープンソース・ソフトウェアを開発してくださった Hyperledger Foundation および OpenZeppelin プロジェクトに所属する開発者の方々と，説明図のなかで利用させていただいた flaticon.com をはじめとするクリップアート作成者の方々に感謝いたします．

そして最後に，会社業務と学業の両立を本当に支えてくれた家族に深謝いたします．特に，日頃から筆者を心身ともに支援し続けた妻　藤本 真理に深く感謝します．

参考文献

[1] "EIP-20: Token standard," Ethereum Improvement Proposals. [Online]. Available: https://eips.ethereum.org/EIPS/eip-20. [Accessed: 4-Jan-2023]

[2] "EIP-721: Non-Fungible token standard," Ethereum Improvement Proposals. [Online]. Available: https://eips.ethereum.org/EIPS/eip-721. [Accessed: 4-Jan-2023].

[3] Vitalik Buterin, "A NEXT GENERATION SMART CONTRACT & DECENTRALIZED APPLICATION PLATFORM," Ethereum White Paper, [Online]. Available: https://www.weusecoins.com/assets/pdf/library/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf. [Accessed: 4-Jan-2023].

[4] Solidity Team, “Solidity programming language,” Solidity Programming Language. [Online]. Available: https://soliditylang.org/. [Accessed: 4-Jan-2023].

[5] "PATRICIA MERKLE TREES," Ethereum White Paper, [Online]. Available: https://ethereum.org/en/developers/docs/data-structures-and-encoding/patricia-merkle-trie/. [Accessed: 4-Jan-2023].

[6] "How to Add Custom Tokens to MetaMask," alpharithms, [Online] Available: https://www.alpharithms.com/adding-custom-tokens-to-metamask-153821/. [Accessed: 4-Jan-2023].

[7] 「トークン化」の仕組みとは, [Online]. Available: https://coinpost.jp/?p=234391, [Accessed: 4-Jan-2023].

[8] NFT（非代替性トークン）を活用したデジタル世界の未来. [Online] Available: https://www.pwc.com/jp/ja/knowledge/column/disruptive-technologyinsights/blockchain-featured1.html. [Accessed: 4-Jan-2023]

[9] プレスリリース：ブロックチェーン推進協会(BCCC)がトークンエコノミー部会を新設, [Online] Available: https://prtimes.jp/main/html/rd/p/000000007.000035659.html. [Accessed: 4-Jan-2023].

[10] 究極のフィンテックは「物々交換」？, [Online] Available: https://www.sbbit.jp/article/fj/39629. [Accessed: 4-Jan-2023].

[11] キャッシュレスの次は「マネーレス」が到来する, [Online] Available: https://president.jp/articles/-/31561?page=2. [Accessed: 4-Jan-2023].

[12] P. Pang, H. F. Tang, J. Lam, J. Chan, N. Hobler, K. K. K. Kan, H. Jeong, and R. Lau, "Real estate tokenization-assets." [Online]. Available:

https://assets.kpmg/content/dam/kpmg/cn/pdf/en/2020/04/real-estate-tokenization.pdf. [Accessed: 4-Jan-2023].
[13] 三菱地所リアルエステートサービス, 不動産の証券化とはなにか, The Watch – Trend and Marketing Report, [Online]. Available: https://www.mecyes.co.jp/library/watch/074 [Accessed: 4-Jan-2023].
[14] A. Milev, "Dividend tokens, explained," Cointelegraph, 04-Jul-2019. [Online]. Available: https://cointelegraph.com/explained/dividend-tokens-explained. [Accessed: 4-Jan-2023].
[15] G. Iredale, "Understanding Oracles, Smart Contracts, And The Oracle Problem," 101 Blockchains article. [Online]. Available: https://101blockchains.com/blockchain-oracle-problem/ [Accessed: 4-Jan-2023].
[16] pNetwork Team, "Launching the Oraclize service on Hyperledger Fabric," Provable Blog Article. [Online]. Available: https://medium.com/pnetwork/launching-the-oraclize-service-on-hyperledger-fabric-c336c2d7d9b1 [Accessed: 4-Jan-2023].
[17] "Securely connect smart contracts with off-chain data and services," [Online]. Available: https://chain.link/ [Accessed: 4-Jan-2023].
[18] S. Walters, " Chainlink Review: Smart Contract Solutions for any Blockchain," [Online]. Available: https://www.coinbureau.com/review/chainlink-link/ [Accessed: 4-Jan-2023].
[19] M. Herlihy, "Atomic Cross-Chain Swaps," Proceeding of the 2018 ACM Symposium on Principles of Distributed Computing. Pp.245-254, 2018.
[20] Matthew Zipkin etc., “Cross-Chain Atomic Swaps,” Bcoin article. [Online]. Available: https://bcoin.io/guides/swaps.html [Accessed: 4-Jan-2023].
[21] Web3 Foundation, "Polkadot: Decentralized Web 3.0 Blockchain Interoperability Platform," 2021. [Online]. Available: https://polkadot.network/. [Accessed: 4-Jan-2023].
[22] "What is Polkadot? (DOT)," Kraken’s Crypto Guides, [Online]. Available: https://www.kraken.com/en-us/learn/what-is-polkadot-dot. [Accessed: 4-Jan-2023].
[23] Vitaly Romanov, "Polkadot. The Polka Dot Blockchain Story," [Online]. Available: https://everkit.org/en/articles/polkadot-the-polka-dot-blockchain-story. [Accessed: 4-Jan-2023].
[24] J. Poon and V. Buterin, "Plasma: Scalable Autonomous Smart Contracts", Whitepaper, pp.1-47, 2017, [online] Available: https://plasma.io/plasma.pdf [Accessed: 4-Jan-2023].
[25] "Bancor network." [Online]. Available: https://www.bancor.network/. [Accessed: 4-Jan-2023].

[26] "Zcash's zero knowledge proofs, ZK Snarks, and more," Gemini. [Online]. Available: https://www.gemini.com/cryptopedia/zcash-zero-knowledge-proof-zk-snarks-mining#section-future-of-zero-knowledge-proofs. [Accessed: 4-Jan-2023].
[27] ブロックチェーン技術を活用した本人確認（KYC）高度化プラットフォーム構築の実証に係る報告書，ブロックチェーン研究会 [Online] Available: https://www2.deloitte.com/content/dam/Deloitte/jp/Documents/about-deloitte/news-releases/jp-nr-nr20180713-report.pdf . [Accessed: 4-Jan-2023].
[28] "Verifiable Credentials Working Group Homepage," W3C(World Wide Web Consortium) [Online] Available: https://www.w3.org/2017/vc/WG/ [Accessed: 4-Jan-2023].
[29] "Verifiable Credentials Data Model v1.1," W3C Recommendation, [Online]. Available: https://www.w3.org/TR/vc-data-model/ [Accessed: 4-Jan-2023].
[30] "A Primer for Decentralized Identifiers," Draft Community Group Report 11 November 2021, [Online]. Available: https://w3c-ccg.github.io/did-primer/. [Accessed: 4-Jan-2023].
[31] 決済システムにおけるプログラマビリティの実現, 日銀レビュー, 日本銀行 [Online] Available: https://www.boj.or.jp/research/wps_rev/rev_2022/data/rev22j12.pdf. [Accessed: 4-Jan-2023].
[32] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, [Online]. Available: https://www.rfc-editor.org/info/rfc6749 [Accessed: 4-Jan-2023].
[33] "OpenZeppelin," The standard for secure blockchain application, [Online]. Available: https://www.openzeppelin.com/ [Accessed: 4-Jan-2023].
[34] "ERC-20 token scenario," Hyperledger Fabric Samples, [Online]. Available: https://github.com/hyperledger/fabric-samples/tree/main/token-erc-20 [Accessed: 4-Jan-2023].
[35] "DeFi Security Lecture 8 — Front Running Attack", beaver-smartcontract-security article [Online]. Available: https://medium.com/beaver-smartcontract-security/defi-security-lecture-8-front-running-attack-3247045dd9cd [Accessed: 4-Jan-2023]
[36] Microsoft Threat Modeling Tool の脅威, Microsoft, [Online]. Available: https://learn.microsoft.com/ja-jp/azure/security/develop/threat-modeling-tool-threats [Accessed：4-Jan-2023].