ブロックチェーン導入における課題とその対応について

2016年8月23日

株式会社NTTデータ　赤羽喜治

NTT DATA

Agenda

NTT DATA

1．直近のブロックチェーン界隈のトピック
- Ethereumハードフォーク問題
- ビットコイン半減期問題
- Vault OSローンチ

2．当社の取組と実装事例
- 貿易金融
- 分散板寄せ

3．導入にあたって考慮すべき安全面の課題
- ブロックチェーンを構成する技術
- 各レイヤーごとの課題

4．実システムへの導入に向けて

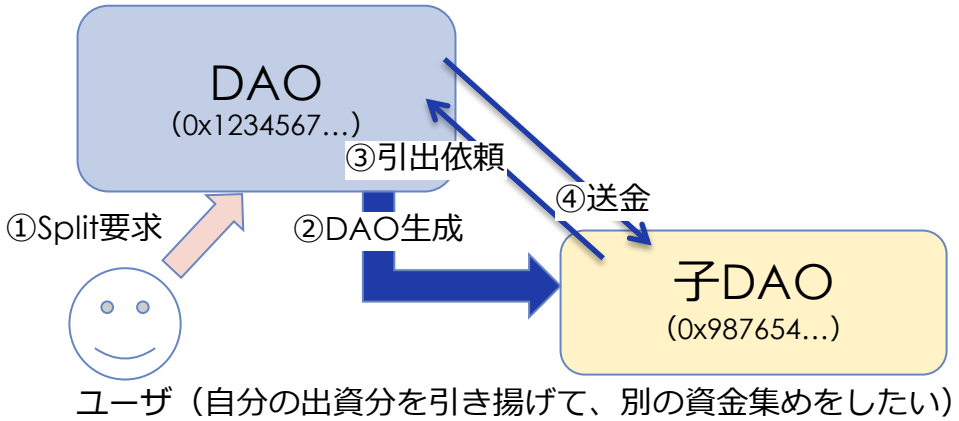
１．直近のブロックチェーン界隈のトピック

1.1 Ethereumハードフォーク問題

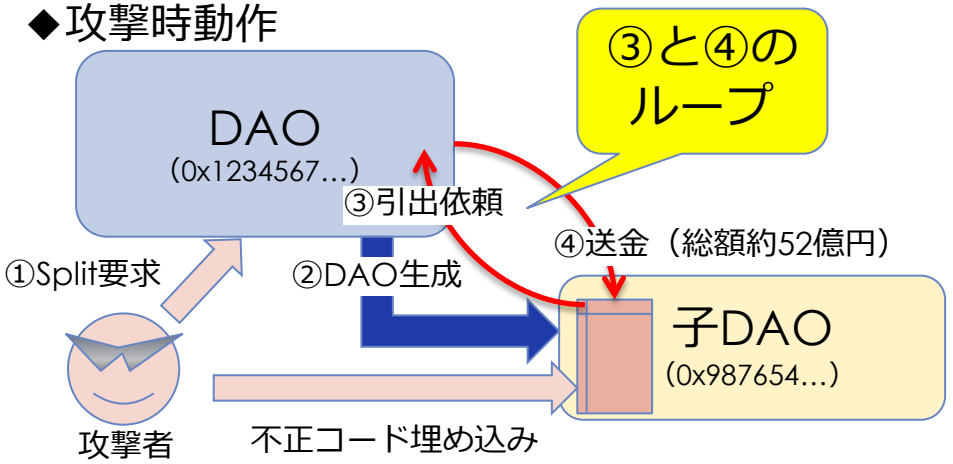
2016年6月　The DAO Attack問題発生（DAO・・・decentralized autonomous organization）

スマートコントラクトを使って作られた、資金集めプログラムの脆弱性をついて不正送金が繰り返され、約50億円もの詐取が行われかけた事件

◆正常時動作



◆攻撃時動作



2016年7月　Ethereumコミュニティがどのような対応を取るか注目されていたが、結局ハードフォーク（ブロックチェーンを不正の行われる以前に巻戻し）となった。反対運動も発生。

Mt Gox事件と同じく、Ethereumという基盤ではなく、DAOというプログラムの脆弱性が原因。とはいえ・・・

本件で垣間見えたように、何らかの原因で正しくない情報がブロックチェーンに書き込まれてしまった場合の運用対処方法は重要な観点となる

1.2 ビットコイン半減期問題

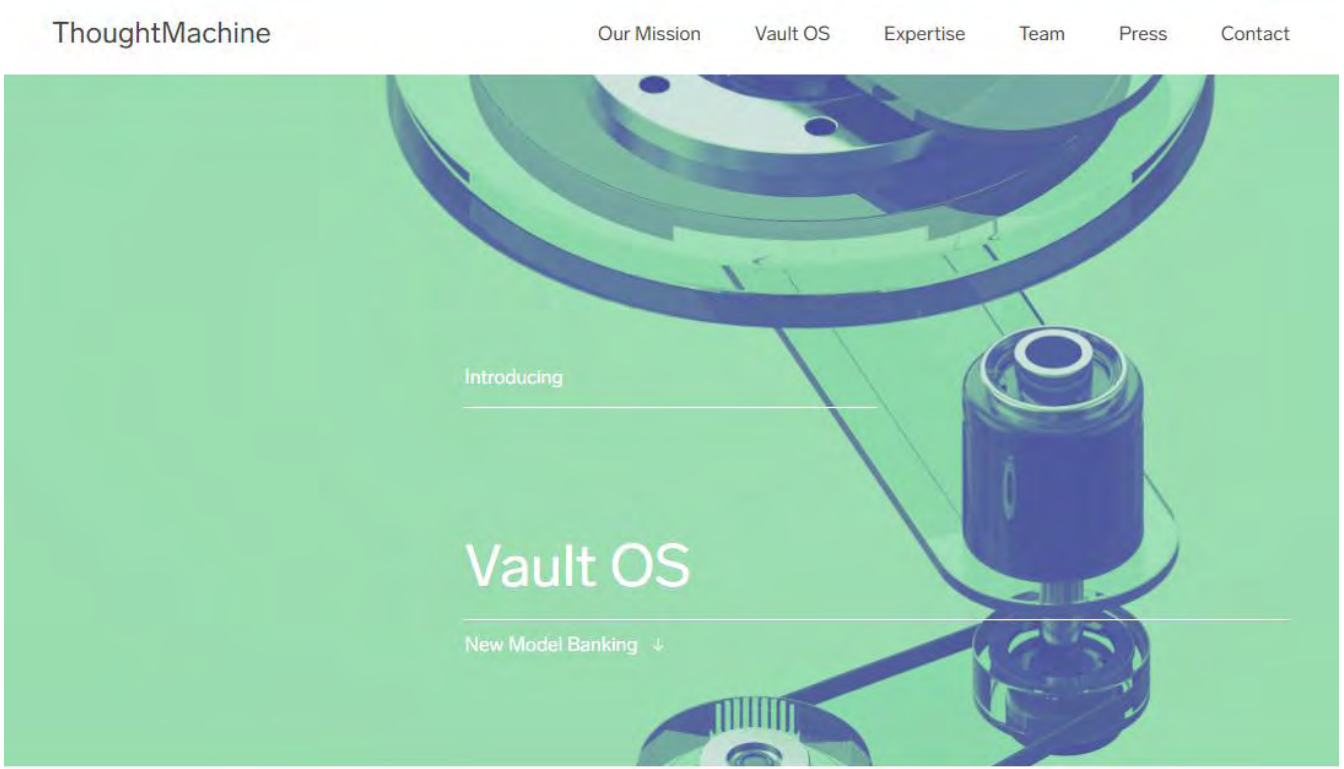
・2016年7月、ブロック生成の報酬が半額に(25BTC→12.5BTC)

➡ マイナー業界への影響はBTC相場の高騰で限定的(※)

※取引所「Bitfinex」への
ハッキング・盗難事件により
急落

1.3 Vault OSローンチ

銀行基幹システムをブロックチェーンで置き換えることを目指すVault OS発表



Googleをスピンアウトした技術者により設立されたThoughtMachine社が開発。
LINUXベース、クラウド（AWS）上で提供（BaaS）。
ブロックチェーンを謳う傍らで「中央集権型・パーミッション型」とも言っていて詳細は不明。

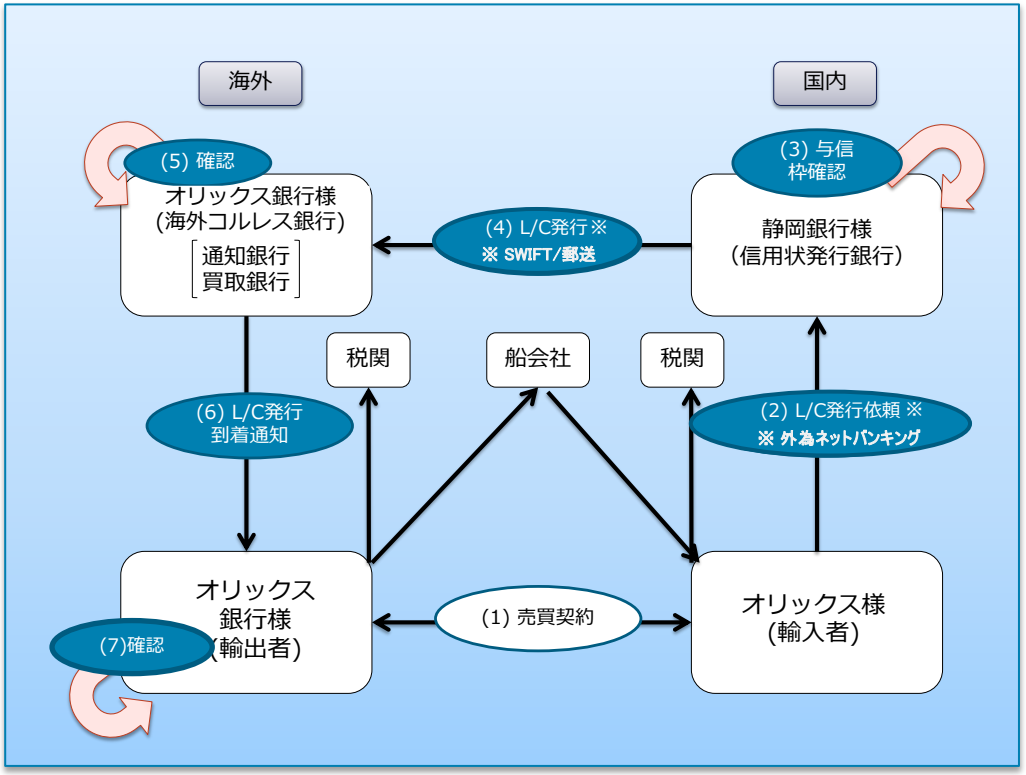
２．当社の取組と実装事例

貿易金融分野へのブロックチェーン技術の利用

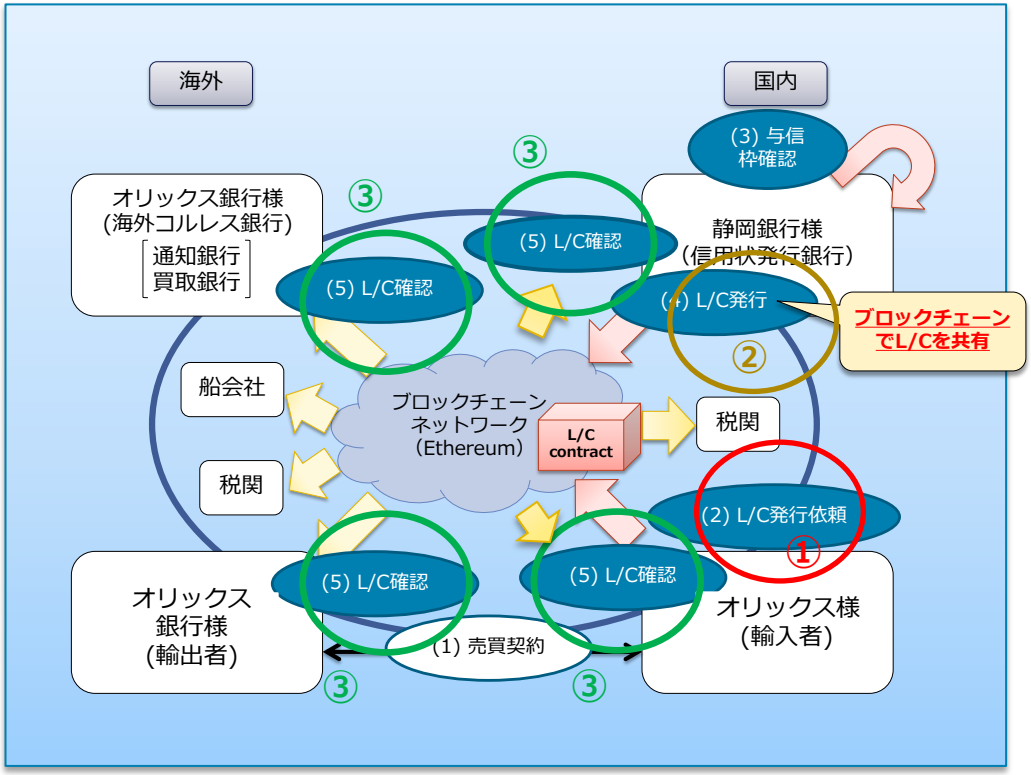
NTT DATA

Ethereum※を用いて、貿易金融のＬ／Ｃ発行に係る業務の一部をプロトタイプ実装し検証いたしました。
本PoCでの検証対象となる機能は、以下の通りです。
①信用状発行依頼
②信用状発行（開設依頼の受付）
③信用状確認（開設取引の照会）

従来型システムの処理の流れ

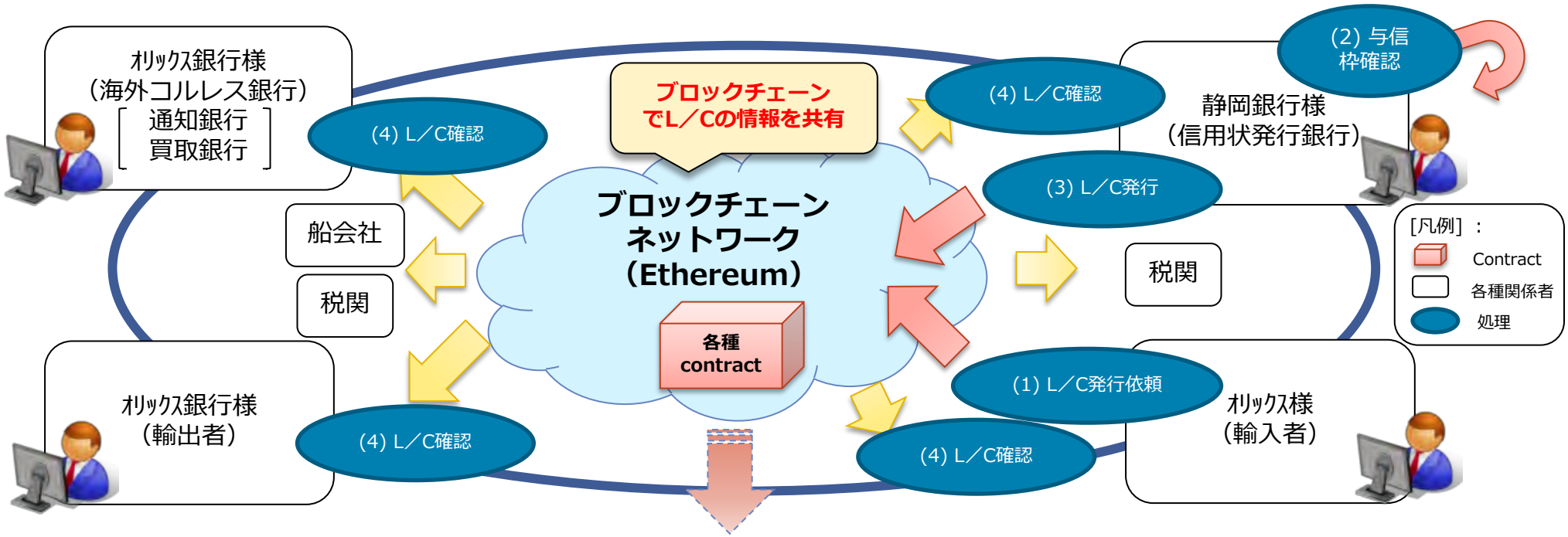


Ethereumのブロックチェーン技術を活用した処理の流れ



※ 分散型アプリケーションの構築プラットフォーム
ブロックチェーンの支払いの仕組み以外に、独自の振る舞いを持つコントラクトをプログラマブルに定義可能となっており、様々な拡張が容易であることが特徴

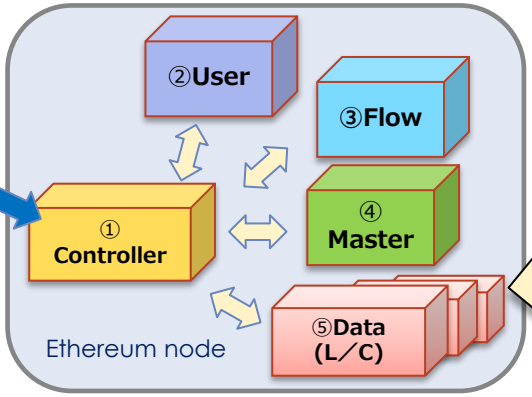
検証内容・検証範囲・アプリケーション概要



【Contract構成】
①**Controller**Contract
　全てのContract処理の起点(呼び出しの窓口)
②**User**Contract
　ユーザ毎の権限設定を管理
③**Flow**Contract
　全てのL／Cのアドレスを保有
④**Master**Contract
　画面に共通的に表示するコードや値を保持
⑤**Data**Contract(L／C)
　L／Cの内容を保持

【システム構成】

データ取得
(JSON-RPC接続)

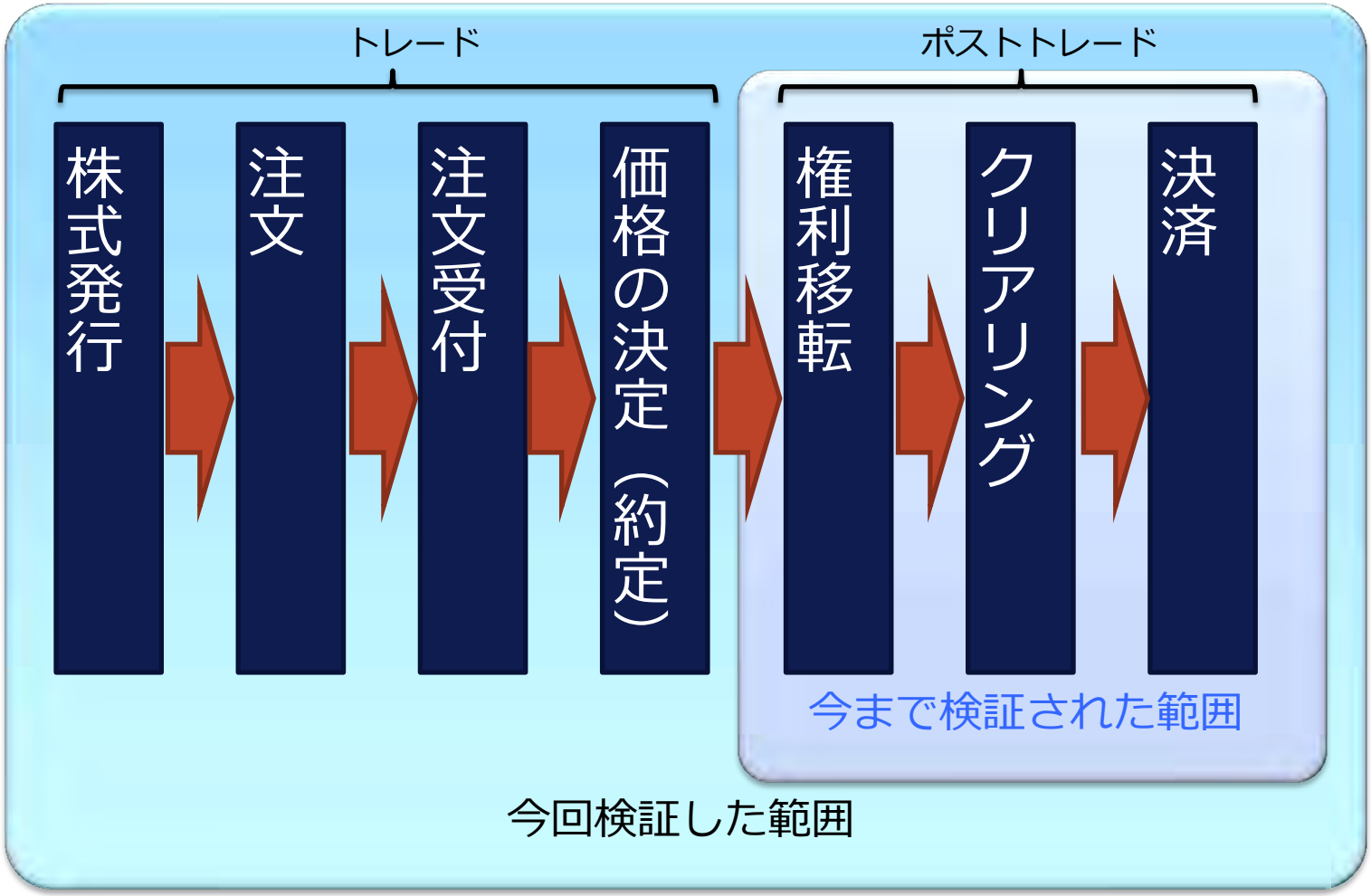


ブロックチェーンの各ブロックへ個別に格納された各種コントラクト（アプリケーションプログラム）が連携し、L／Cを生成し、フロー制御でステータスを更新する。更新結果はブロックチェーンに格納、保持される。

分散証券取引プラットフォーム検証範囲

NTT DATA

- 証券取引全体にブロックチェーン技術を適用



分散証券取引におけるスマートコントラクト構成

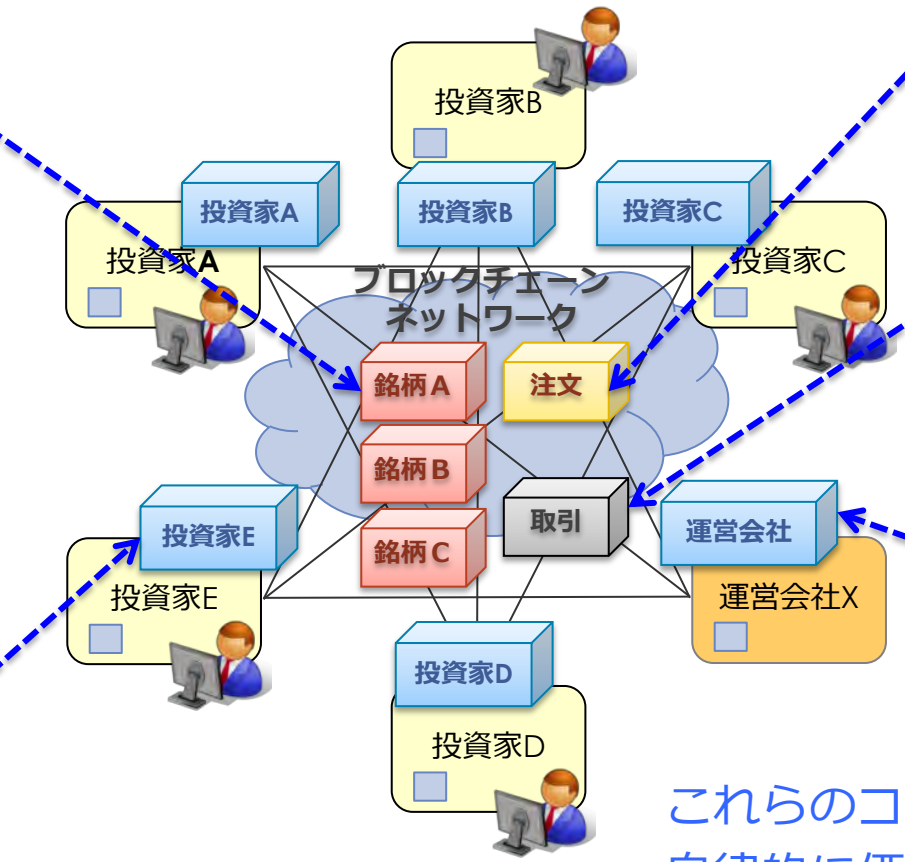
NTT DATA

銘柄コントラクト

〈変数〉
・銘柄名
・銘柄コード
・所有者リスト
・発行済株数
・現在価格　など
〈メソッド〉
・投資家追加
・権利移転
・保有株情報取得
・銘柄情報取得
・所有者リスト取得
・増資
・減資

投資家コントラクト

〈変数〉
・氏名
・住所
・残高
・買付余力
・所有銘柄リスト
〈メソッド〉
・買い注文
・売り注文
・注文履歴取得
・取引履歴取得
・所有銘柄情報取得



[凡例]　：Ethereum node

注文コントラクト

〈メソッド〉
・注文受付
・注文内容取得
・注文内容更新
・注文削除

取引コントラクト

〈メソッド〉
・価格決定（板寄せ）

運営会社コントラクト

〈変数〉
・管理銘柄一覧
・残高
〈メソッド〉
・板寄せ実行
・クリアリング実行
・決済

これらのコントラクトが互いに連携し、
自律的に価格を決定し、その結果より
所有権の移転・決済を実行する。

詳しくは日経FinTech5月号をご参照ください

Technology

テクノロジー ｜ ブロックチェーン

イーサリアムで株式売買システム

ブロックチェーン技術はどこまで金融業務をカバーできるのか

NTTデータが2016年4月、ブロックチェーンを使ったスマートコントラクトの実行基盤「イーサリアム」で株式売買システムを検証した。そこから見えてきた課題と、今後の可能性を探る。

NTTデータ金融事業推進部の開発チームは2016年4月、ブロックチェーン技術を用いた未公開株式市場向けの証券売買システムについて、システム構築の可能性や課題について発表した。

検証の範囲は売買注文の受付から、取引価格の決定（約定）、クリアリング（清算）や決済までの全工程。各工程の処理ロジックをコントラクトとして実装し、ブロックチェーン上で連携して自動執行させるサービスを実現した。

ブロックチェーンを証券取引業務に応用する試みでは、これまでにも「株式とひも付けたコインをブロックチェーン上で発行し、権利移転に使う」「取引後の清算や決済にブロックチェーンを使う」など、ポストトレードを対象とした検証実験は行われていた。

投資家による注文から決済までの全業務をブロックチェーン上で実現させた取組みは珍しい。実証実験の詳細と、実験から得られた知見や今後の課題を紹介する。

●ビットコインとイーサリアムの違い

なぜ売買システムを検証したか

NTTデータの開発チームが今回、株式売買の全工程を検証の対象にしたのは次のような理由からだ。

「ブロックチェーンを用いた開発の可能性を検証する中で、スマートコントラクトは重要なテーマ。複雑な業務システムをブロックチェーンやコントラクトを用いて実現できれば、様々な領域の顧客に新たなサービスを提供する可能性が広がる。分散証券取引はそれを検証するに格好の題材だった」（NTTデータ 金融事業推進部 技術戦略推進部システム企画担当 部長の赤羽真治氏）。

ブロックチェーンを売買システムに応用する検証実験の多くは、取引成立後の「権利移転」「クリアリング」「決済」といったポストトレード業務を対象にしている。

今回の検証で、NTTデータはポストトレードに加え、プロント部分に当たる「株式発行→注文→注文受付→板寄せによる価格の決定（約定）」も含めた全行程を対象とした。

「ブロックチェーン導入のメリットを享受するためにはポストトレード

6 ｜ Nikkei FinTech 2016.05.

3．導入にあたって考慮すべき安全面・運用面の課題

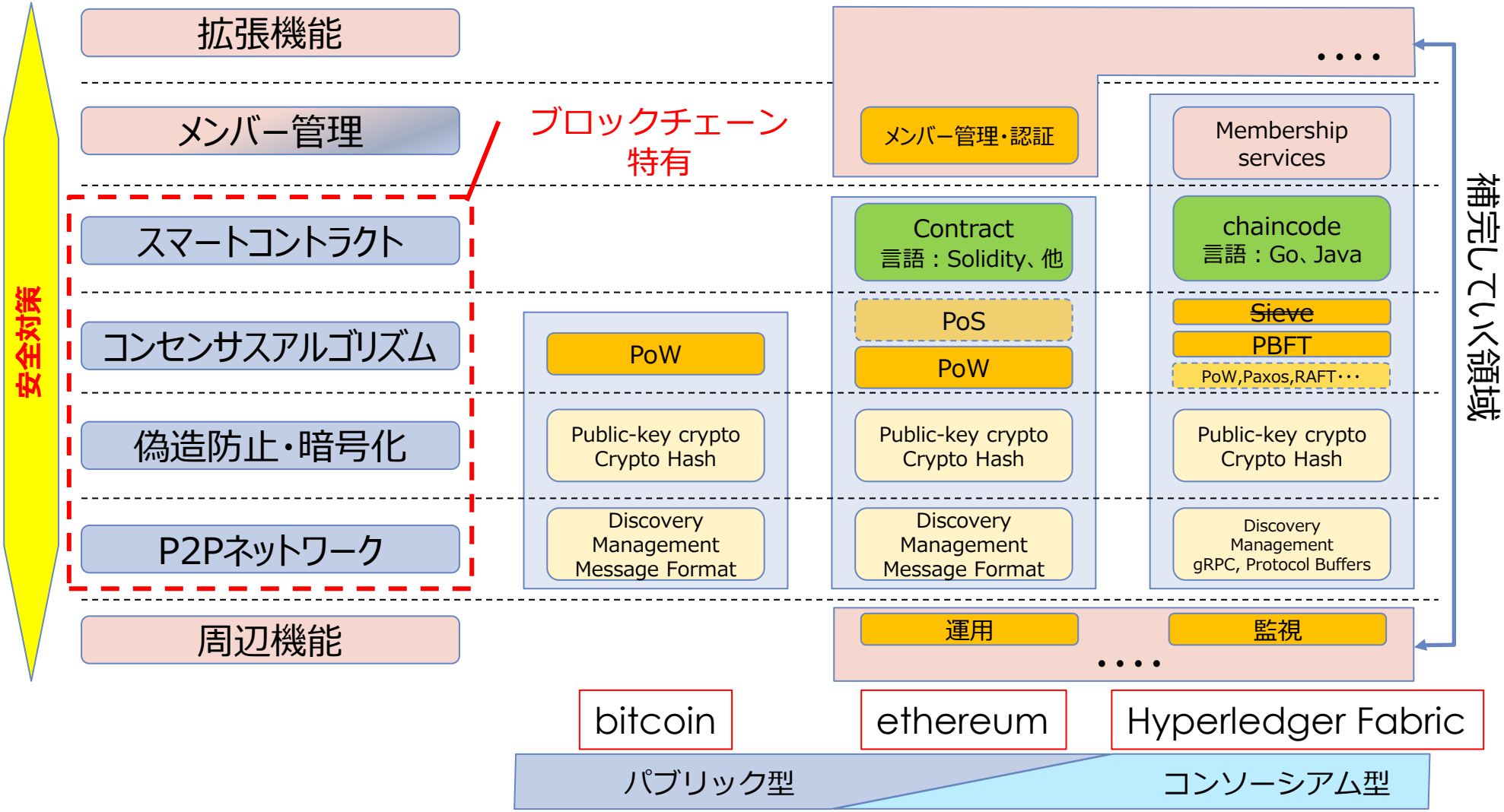
主要なブロックチェーンプラットフォーム実装

NTT DATA

名称	開発元	特徴
Bitcoin Core	Bitcoin Core	オリジナルのビットコインクライアントの後継
Ethereum	Ethereum Foundation	スマートコントラクトを使用してコインの移転以外にも広く使える
Hyperledger Fabric	Hyperledger Project	PoWではなくBFTを使うのでブロックの生成が速い。認証の仕組みを標準で持つ
Corda	R3	参加者間ですべてのデータが共有されるわけではない
Chain OS 1	Chain	一貫性の維持にSimplified BFTを使う
mijin	テックビューロ	Proof of Stakeを使用するが、高速な処理を指向
Orb	Orb	ブロックチェーンのフォークによるトランザクションの手戻りリスクの低減
Eris	Eris Industries	Ethereumから派生し、許可型ネットワーク向けに改造したもの

各ブロックチェーン実装の構成比較と課題

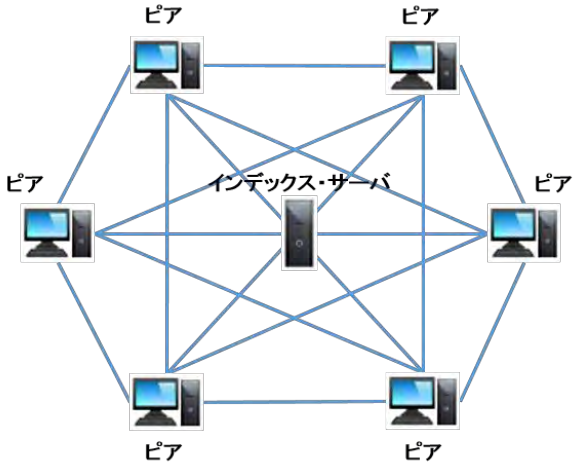
ブロックチェーンを構成する各技術のレイヤごとに技術の深堀と検証が必要
メンバーシップ管理など、従来システムと同様の安全対策が求められるレイヤーと
ブロックチェーンならではの観点が求められるレイヤーとがあり、特に後者についての蓄積が求められている。



P2Pネットワーク

P2Pネットワークの分類と特徴（１）

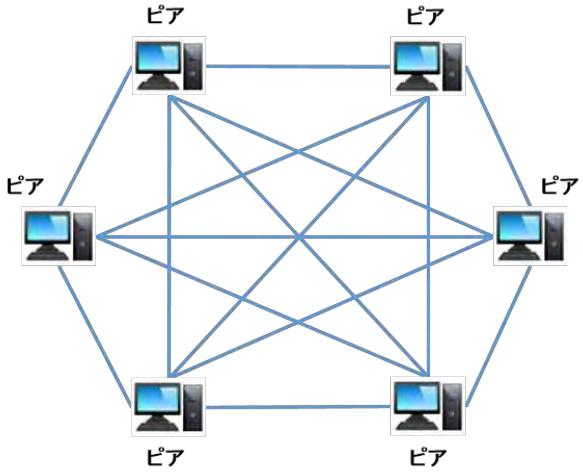
主なP2Pネットワークの形態



ハイブリッドP2P

・探索用のインデックス・サーバを持つ。

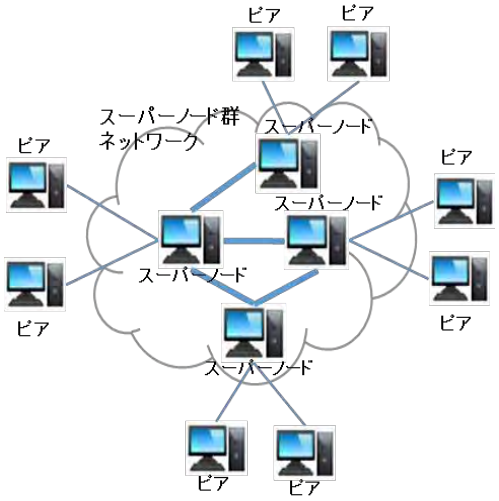
○シンプルでシステムを管理しやすい
×システムに中心を持つため、スケーラビリティや耐障害性が十分に発揮されにくい



ピュアP2P

・全ノードが同じ役割

○スケーラビリティや耐障害性が高い
×実装が複雑、ノード数が増えた場合に、マシンリソース消費拡大の懸念



スーパーノード型

・メンバシップ管理など特定のノードが管理機能を持つ

○ハイブリッドP2Pとピュア P2Pの長所を併せ持つ。
×スーパーノードがSPOFになりやすいため対策が必要

安全面・運用面で考慮すべきこと

NTT DATA

パフォーマンス

・転送回数やネットワーク遅延等
  – P2Pネットワーク上で動作するブロックチェーンは、クライアント・サーバ型のシステムと比較して遅延が懸念されるため、リアルタイム性を求められる領域での適用が難しいとされている

確実性

・ブロードキャスト
  – ブロックチェーンネットワーク全体で同期が可能か
  – 到達保証（受信確認）はどうするか

・ノードやネットワークの信頼性
  – ブロックチェーンが有効に動作するために最低限必要なノード数や、これを下回った際の運用ルールの策定が必要
  – ノード障害を考慮したネットワーク設計

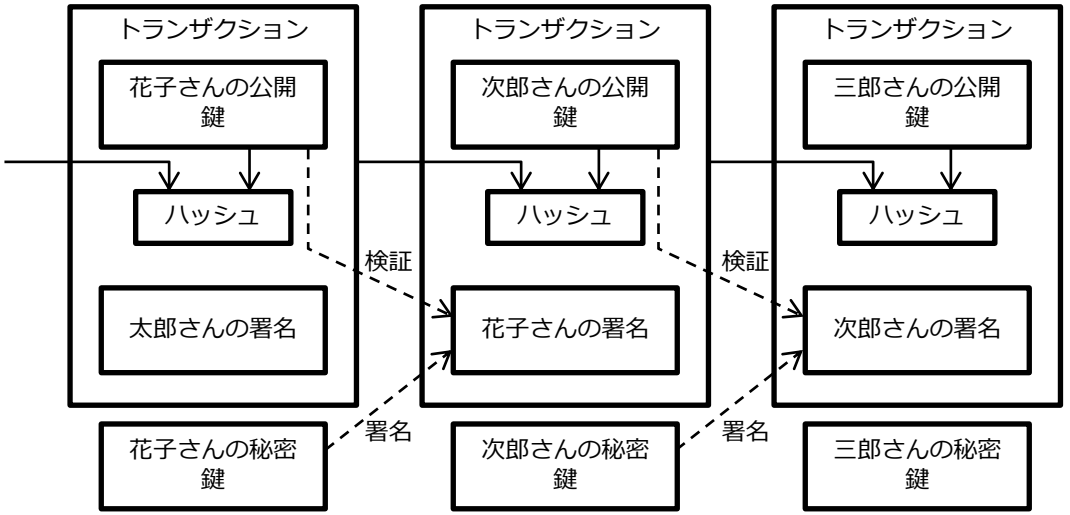
偽造防止・暗号化

ブロックチェーンにおける電子署名の利用

ブロックチェーンにおける偽造・改ざん防止は、既知の暗号化技術である電子署名とハッシュを組み合わせることで実現している。

電子署名による本人性保証

トランザクションにおける電子署名の利用



ブロックチェーンでは各トランザクションに1つずつ電子署名が付与される。
また、電子署名を検証するための公開鍵もセットで付与される。
ビットコインを例にとると、電子署名と公開鍵がセットで付与されることで、過去ビットコイン上で行われた全ての取引を順次検証することができる。
ビットコインの電子署名を検証することで、以下を確認することができる。

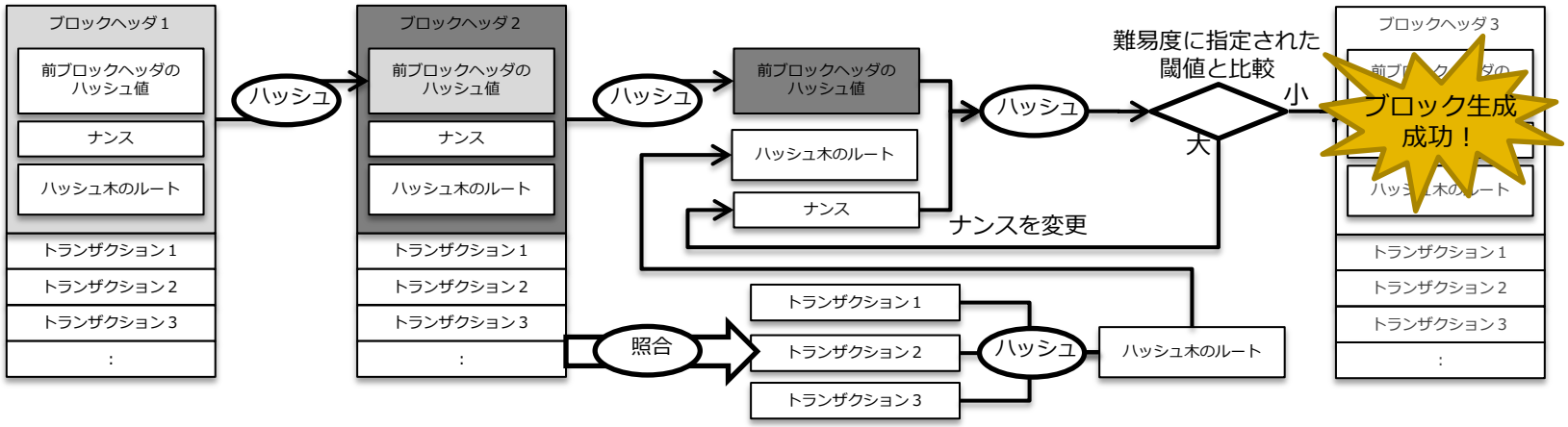
- 第三者が取引内容を偽造・改ざんしていないこと
- 第三者がなりすましを行って取引を行っていないこと
- コインの正しい所有者が確かに取引を行ったこと
  （そんな取引はしていないと否認することを防止）

ブロックチェーンにおけるハッシュの利用

ブロックチェーンにおける偽造・改ざん防止は、既知の暗号化技術である電子署名とハッシュを組み合わせることで実現している。

ハッシュによる改竄防止

ブロック生成時におけるハッシュの利用



ブロックチェーンでは複数のトランザクションをまとめたブロックを作り、ブロックには前のブロックのハッシュを付与する。
また、ハッシュの計算に使用するナンスと呼ばれる値もセットで付与される。
ブロックに付与されるハッシュは、1つ前のブロックをハッシュ関数に入力することで生成される。
そのため、あるブロックの内容を偽造・改ざんすると、ハッシュ関数の特性により、その次のブロックに付与するハッシュが変わり、同様に、以降全てのブロックに付与するハッシュが変わる。
偽造・改ざんを成功させるためには、これら全てのハッシュを再計算しなければならず、偽造・改ざんを困難にする。

安全面・運用面で考慮すべきこと

NTT DATA

秘匿情報をどのように扱うか

・ブロックチェーンにおける暗号化技術の利用は、偽造・改ざんを防止するためのものであり、
取り扱うデータそのものは暗号化されていない。
そのため、ブロックチェーンで機密情報や個人情報等を扱いたい場合に、どのように情報を
秘匿化するか検討する必要がある。

暗号技術を利用したシステムにおける運用課題

・鍵管理
－鍵ペアの有効期間の管理、新しい鍵への置き換え等
・暗号技術の危殆化
－ハッシュ関数や電子署名は、時間の経過と共にその強度が弱くなる運命
－量子コンピュータが登場すると電子署名の有効性が失われる
（ブロックチェーンは長期的に運用される前提にも関わらず検討がされていない。
セキュリティ界隈では取り組みは始まっている（英Post Quantum等））

実装面での脆弱性

・“トランザクション展性“のような実装面で脆弱性が入り込んでいないかの検証
※ビットコインで問題となった、トランザクションの一部が署名対象となっていなかった
ことに起因する脆弱性。デジタル署名が検証可能のままで取引データが改ざん可能だった。
MtGOX事件などでも取り上げられた。

コンセンサスアルゴリズム

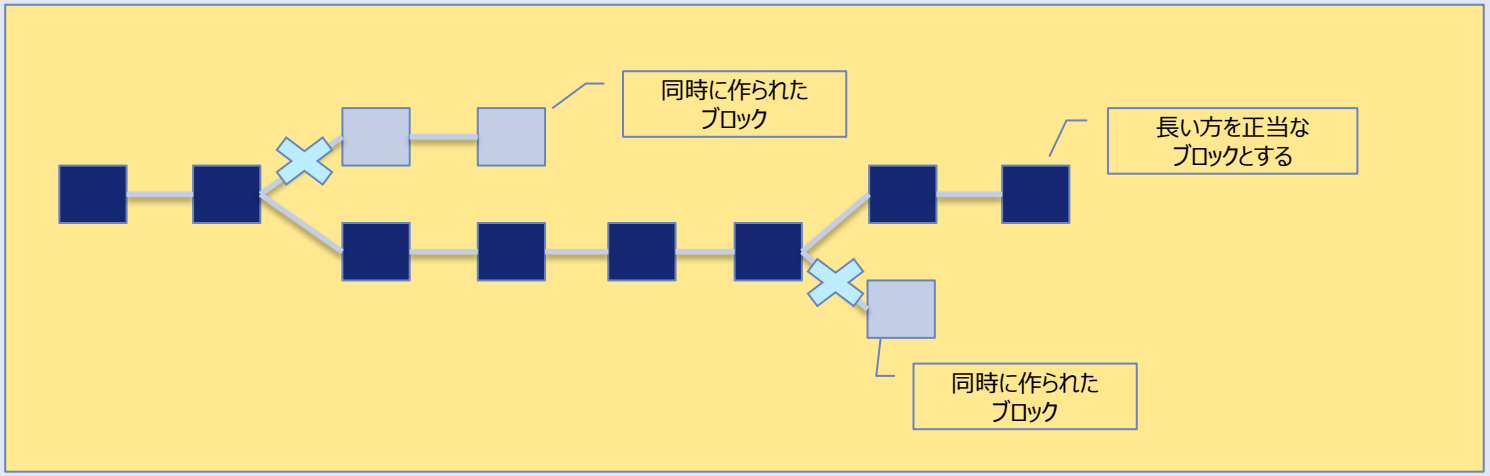
コンセンサスアルゴリズムとは

分散ネットワーク上で各ノードが合意形成をするためのアルゴリズム。

コンセンサスアルゴリズムの一つとしてPoWがある。

- PoW(Proof of Work)

-Proof-of-Workアルゴリズムは、取引情報(Block)を時系列にチェインし、
ひとつ前の取引情報のハッシュ値(タイムスタンプを含む)を元に、自取引のハッシュ値を生成/設定する仕組み。
-改ざん/複製する場合、一部だけの改ざんでは矛盾が発生する為、過去に遡ってハッシュ値を書き替える必要がある。
過去に遡ってハッシュ値を書き替える為には、膨大なコンピューターリソースが必要。
-BitcoinはコンセンサスアルゴリズムとしてPoWを用いている



- PoS(Proof of Stake)、PoI（Proof of Importance)

-Proof of Workへの代替案（マイニングによる消費電力がない等）
-コインを持っている割合(Stake)や“重要性“でブロックの承認の割合を決める

コンセンサスアルゴリズムとは

NTT DATA

BitcoinではPoWでコンセンサスを成立させていたが、ブロックチェーン的には他にも様々なコンセンサスアルゴリズムが提唱、実装されている。

- Raft
  - -P2Pなアルゴリズムと異なり、Leaderが存在する
  - -CandidateからLeaderを選び、 Leaderを中心にデータを送信しコミットしてから合意に達する
- Paxos
  - -L. Lamportが提案
  - -State MachineはProposers、Acceptors、Learnersのいづれかの役割を果たし コンセンサスの合意を目指す
- PBFT(Practical Byzantine Fault Tolerance)
  - -M. Castro と B. Liskovが提案
  - -Client、Validator、Execution、Agreementから構成
- Sieve
  - -PBFTを拡張したアルゴリズム
  - -Client、Validator、Replicaから構成

Hyperledger Fabricではこれらの実装が差し替え可能となることを目指す

アルゴリズムの例：PBFTの動作概要

NTT DATA

1. クライアントが命令をすべてのサーバー（Replica 0～3）に送る。

“リーダー”による合意形成

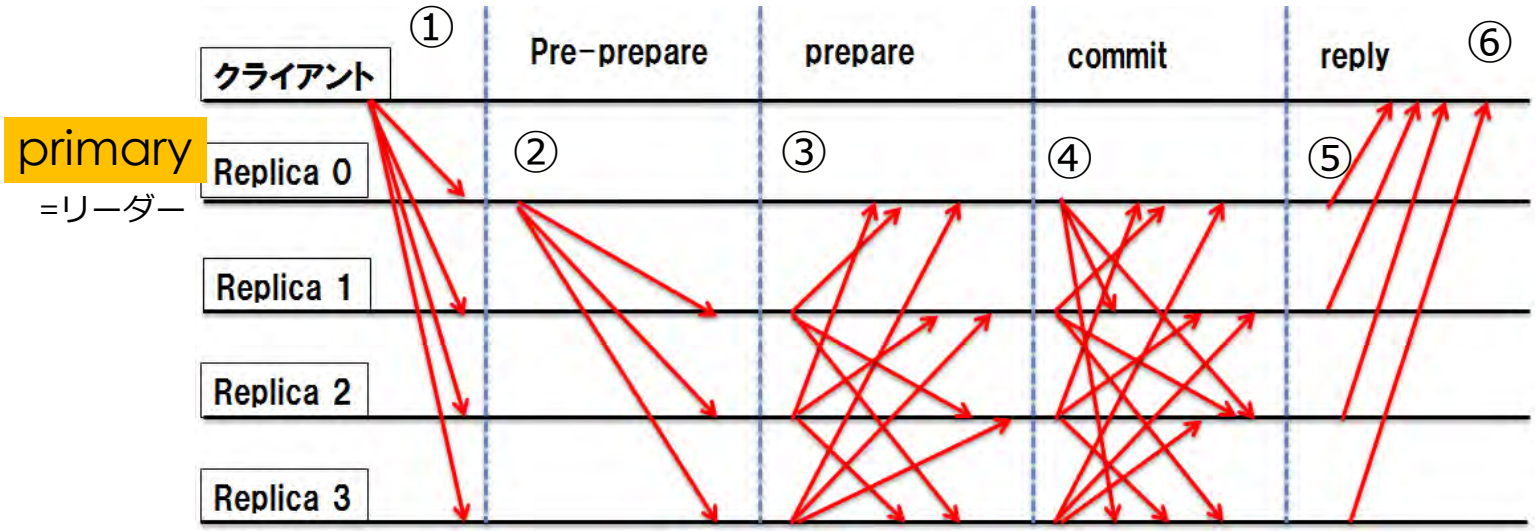
2. primary は実行順序nをつけた上で命令を他のすべてのサーバーへ送付する。

3. 各サーバー は命令を受け取ったら、他のサーバー に受け取った合図を送る。

4. 各サーバーは 3 で他のサーバーが送付した PREPARE メッセージを受け取る。
   ある一定数以上の他のサーバーからのPREPAREメッセージが集まったら、他のサーバーに受け取った合図を送付する。

5. 各サーバーは 4 で他のサーバーが送付したCOMMITメッセージを受け取る。
   ある一定数以上の他のサーバーからの COMMIT メッセージが集まったら、そのサーバーのコミット命令として登録する。
   実行順序n 未満のコミット命令がすべて実行されていれば、この n 番目の命令を実行する。
   そうでなければ、n未満の数値の命令がコミットされるまで、この番号の命令に関しては実行を保留する。
   実行結果をクライアントに送る（REPLYメッセージ）。

6. クライアントは各サーバーが送付したREPLYメッセージを受け取る。ある一定数以上のサーバーからのメッセージが集まったら、
   中身がすべて同じか確認する。同じ REPLYメッセージがある一定数以上あれば REPLY の値として これを実行結果とする



安全面・運用面で考慮すべきこと

NTT DATA

コンセンサスアルゴリズム毎の属性（耐障害性・対攻撃性）把握

・Primaryノード障害発生時のリーダー交替プロセスの振る舞いや
　各ノードの異常動作時（リーダー、リーダー以外）の振る舞いについての検証が必要。

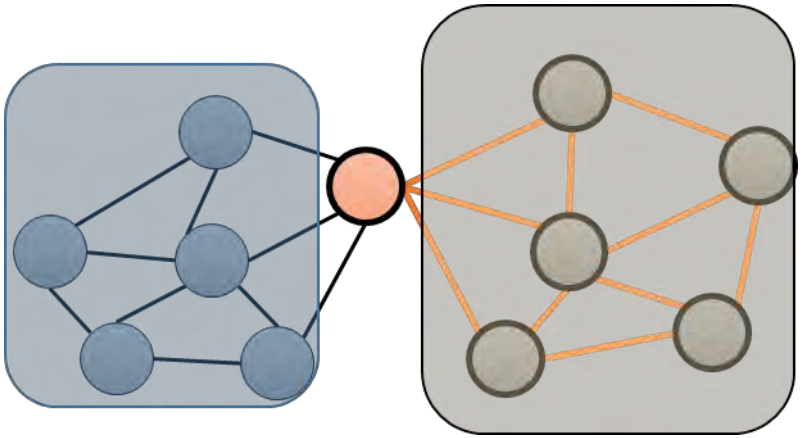
　ex. ノード障害により、リーダーが交替し続け、コンセンサス形成に非常に時間がかかる事象等

分断耐性

　－分断時に複数のブロックチェーンに分岐が発生し、分断解消時に上書き等の問題が発生する

耐攻撃性

　－クエリー内容の改ざんやエクリプス攻撃等への対応



eclipse攻撃のイメージ

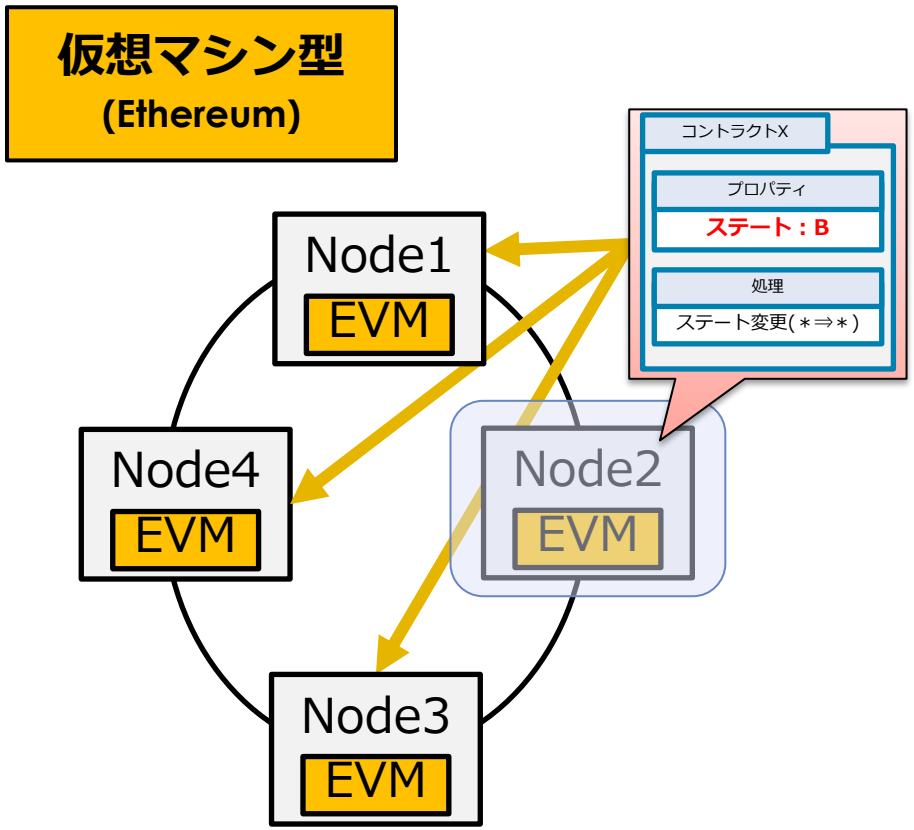
スマートコントラクト

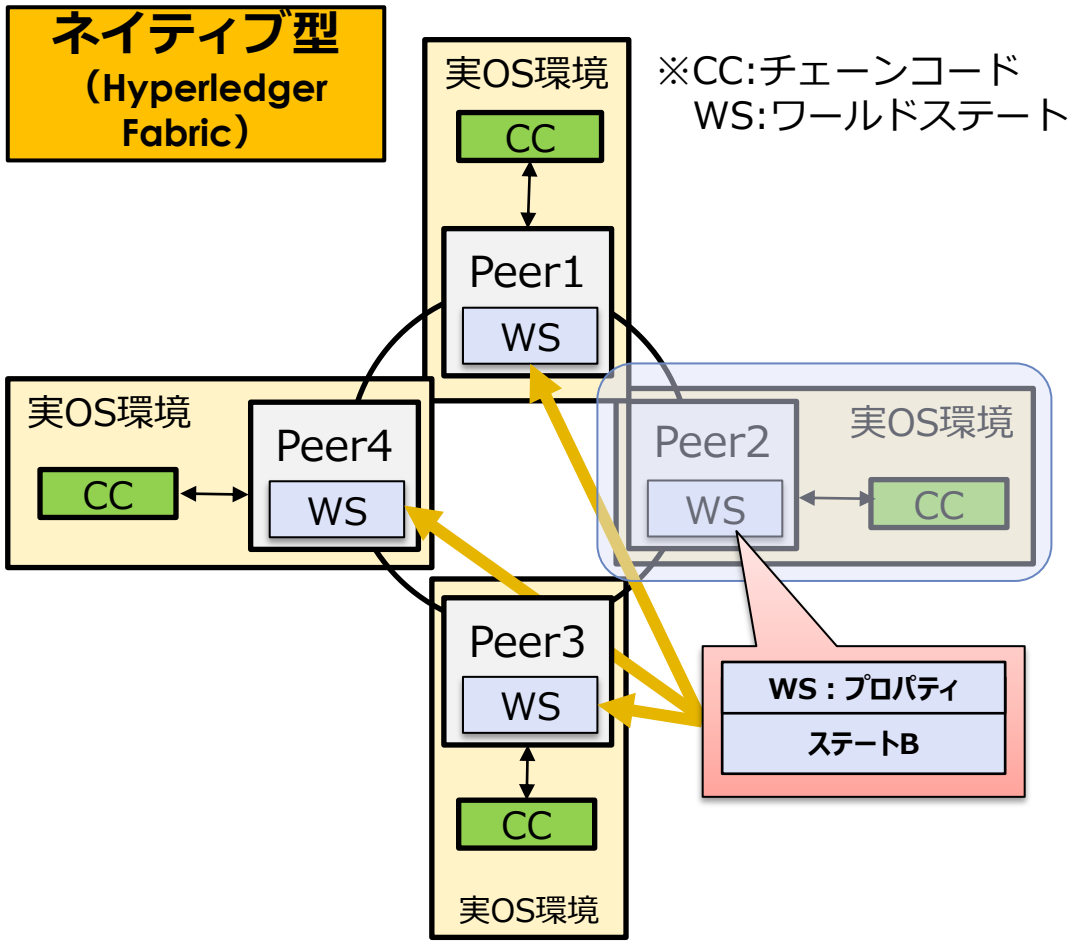
スマートコントラクトの分類

スマートコントラクトの実行環境は、実装によって様々な形態がある

例１）Ethereum：EVMという「仮想マシン」上でプログラムを実行するEthereum
例２）ノードの実際のOS環境上でネイティブなプログラムを実行するHyperledger Fabric

⇒ある程度の制約のかかるEthereumと一般のプログラムと同じ自由度のあるHyperledger Fabric





安全面で考慮すべきこと

NTT DATA

コントラクト自体の脆弱性

・コントラクトコードのバグ・脆弱性をついて、不正な処理を実行されることが考えられる。
例）The DAO Attack事件
ブロックチェーン自体の脆弱性ではなくとも、コントラクトの脆弱性により誤った記録がブロックチェーンに書き込まれるという事象。攻撃ではなくともコントラクト自体のバグにより、同様の問題が発生するリスクがある（従来のシステムと同様のリスクが存在）

スマートコントラクトの実行環境・配布方式

・自由度の高さと安全性は基本的に二律背反。プログラムの安全性がどのように担保されているのか、実装ごとに確認が必要
・仮想マシンで実行環境を分離したり、機能やリソースを制限したりすることにより不具合や悪意のあるコードへのある程度の問題の緩和はできるが、
開発生産性や実行効率に影響がある。
スマートコントラクトプログラムを一般的なプログラミング言語で書き、
コンピュータ上で直接動作させるのは効率は良いが、
不具合や悪意のあるコードへの考慮がより重要になる。

安全面・運用面の課題　まとめ

NTT DATA

安全面の課題

・ブロックチェーンに関する安全性の定義が定まっておらず、その結果十分な安全性の検証がなされているとは言いがたい。

・ブロックチェーン基盤だけでなく、その上で実行されるプログラムの安全性を担保する手段についても十分な検証が必要。

運用面の課題

・ブロックチェーンに誤った情報が書き込まれた際の対応については検討と検証が必要

・P2Pネットワークに分断が発生した際に運行を続行するか停止するかといったルール作りも必要となる

・コンソーシアム型、パブリック型といった切り口でも運用の考え方は大きく変わる

“ブロックチェーンは安全面の課題がある危ない技術”というメッセージではない。
過去に出現した様々な新規技術と同様に検証すべき項目が数多く残されているということ。
“原理的に大丈夫”という言葉の及ぶ範囲、実装とのGAPをしっかりと意識する必要がある。
ブロックチェーンはパーツの一つでしかない。

NTT DATA

Global IT Innovator