

分散型台帳技術にかかる基礎実験

第3回FinTechフォーラム
『金融分野における分散型台帳技術の活用に向けて』

日本銀行決済機構局
河田雄次

2017年2月28日



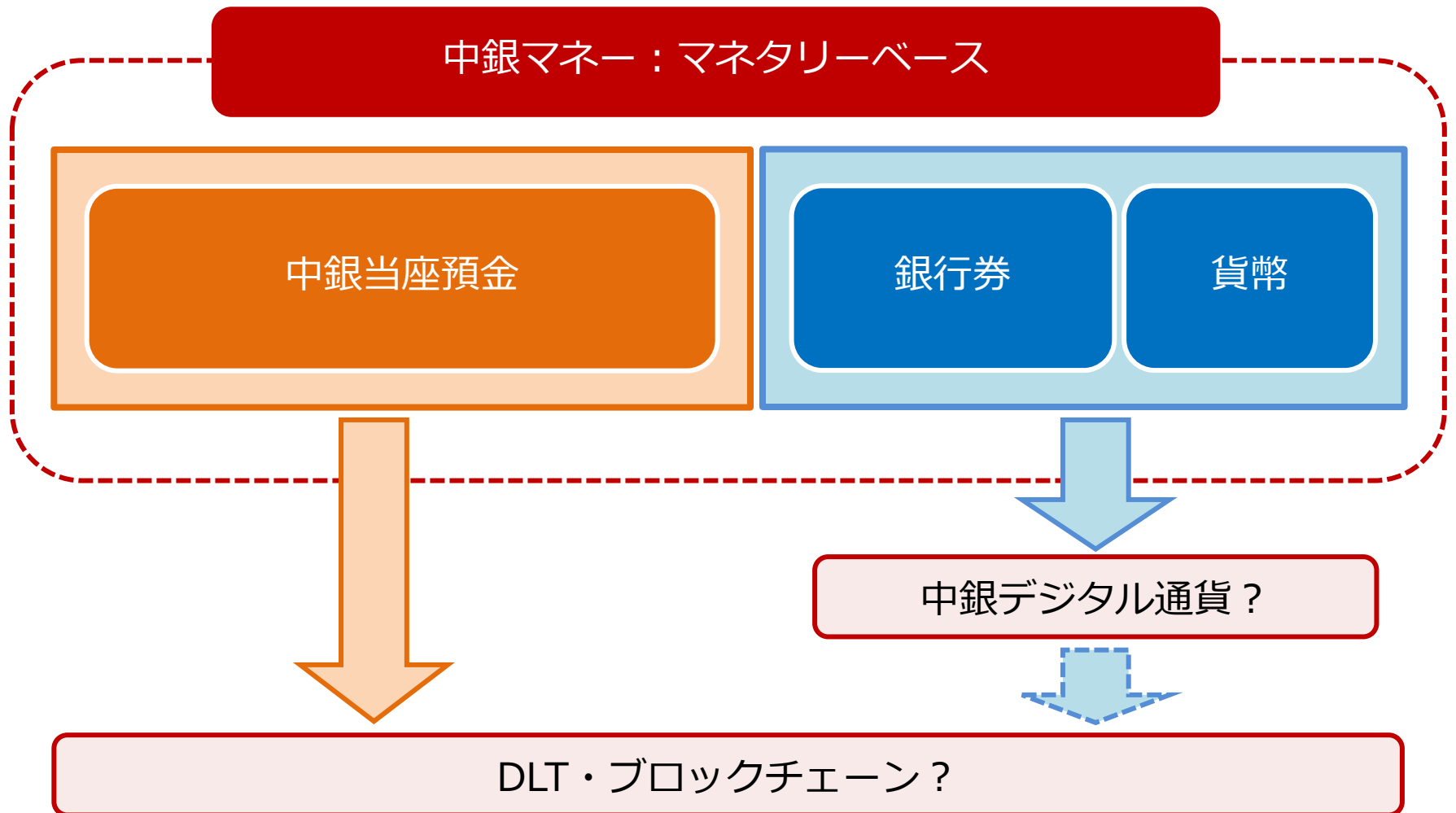
本日のテーマ

1. 中央銀行とDLT
2. DLTにかかる基礎実験
3. 金融インフラ分野における許可型DLTの活用可能性

(注) 本資料は、作成者の個人的見解を取りまとめたものであり、日本銀行の公式見解を表すものではありません。

1. 中央銀行とDLT

- 各国中銀にて研究活動が活発化。活用を巡る議論には「中央銀行デジタル通貨」に加え、「中央銀行当座預金（金融インフラ分野）のDLT化」も含まれる。



(参考) 主要中央銀行における取組み

DLTの活用可能性

中銀デジタル通貨

ECB：証券ポストトレードへの活用可能性に関する調査論文を公表（2016.4）

ユーロ圏

ECB：日本銀行との共同研究を公表（2016.12）
ドイツ連銀：ドイツ証券取引所と共同での実証実験を公表（2016.11）

英国

イングランド銀行：民間と共同で、DLTを使った仮想の金融資産の管理・移転計画を公表（2016.6）

ロンドン大学：中銀発行デジタル通貨 (RSCoin)の論文を公表（2016.2）

米国

FRB：NY連銀、シカゴ連銀と共同で、決済システムへの活用可能性に関する調査論文を公表（2016.12）

カナダ

カナダ中銀：民間と連携し、DLT上で中銀債務を発行・流通・決済する実証実験を開始（2016.6）

ロシア

ロシア中銀：DLTを用いた市場参加者間の情報伝達ツールの試作品を開発（2016.10）

中国

中国人民銀行：中長期的にデジタル通貨を発行する構想を発表（2016.1）

スウェーデン

リクスバンク：e-kronaの補完的な発行を検討するプロジェクト立ち上げを発表（2016.11）

2 - 1. DLTにかかる基礎実験 - 概要

- 目的：技術の理解深耕
 - 「銀行間資金決済システムの擬似環境」を用いて有効性や課題を評価。
- 評価項目：

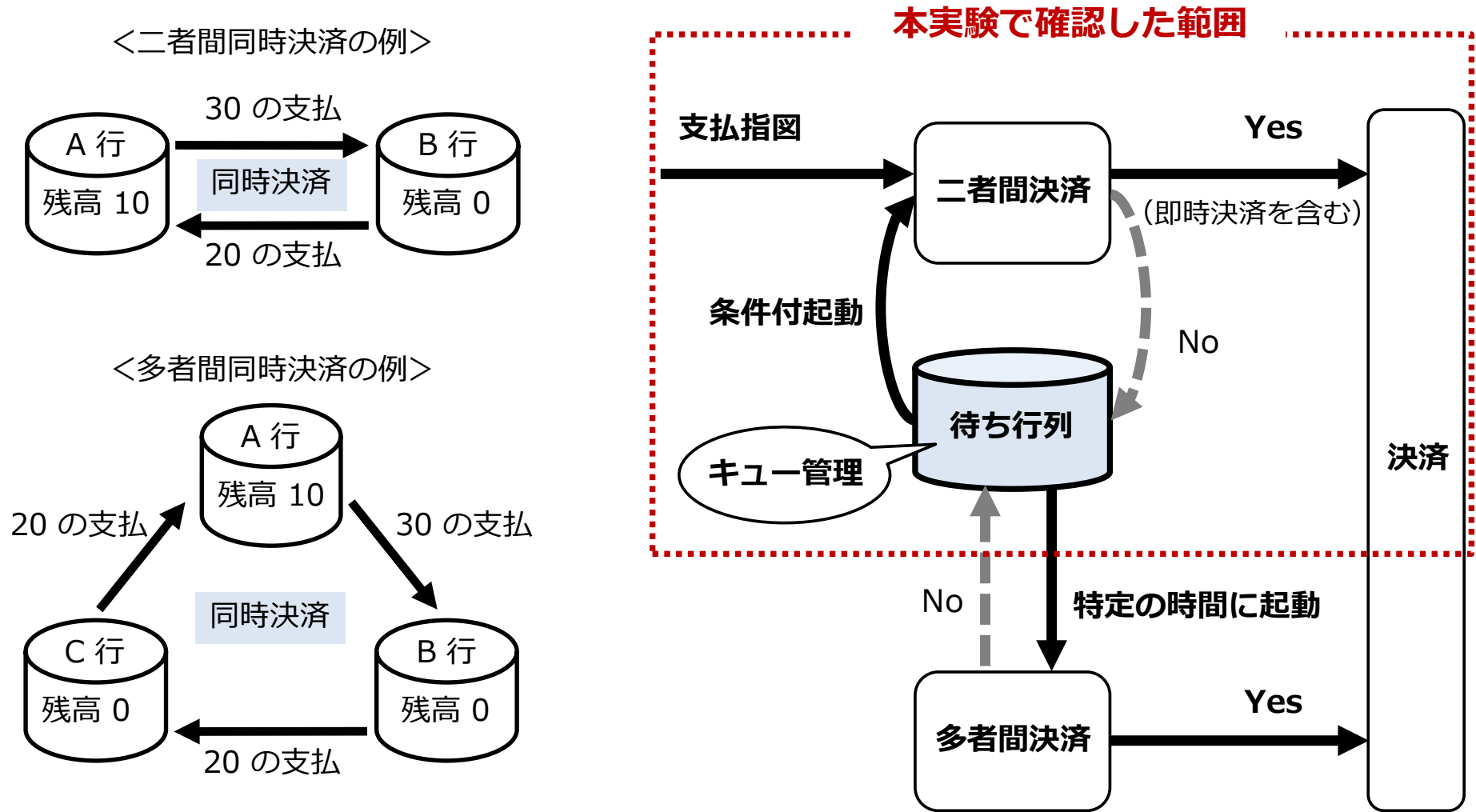
| 評価項目 | 実験内容 |
|--------------|--|
| ① スマートコントラクト | <ul style="list-style-type: none">• 複雑な業務処理の実装可能性• 非確定的な処理の影響 |
| ② 処理性能 | <ul style="list-style-type: none">• 検証ノード数やリクエスト数の影響• 日銀ネットの業務処理量における処理性能 |
| ③ 可用性 | <ul style="list-style-type: none">• 障害時および障害復旧後の業務継続性 |

- DLT基盤：Hyperledger Fabric v0.6.1-preview版
- 実験環境：スタンドアロン端末

(注) 基礎実験を進めるにあたっては、日本IBM、NTTデータ、日立製作所のスタッフの方々から、多くの有益かつ貴重なコメントを頂いた。

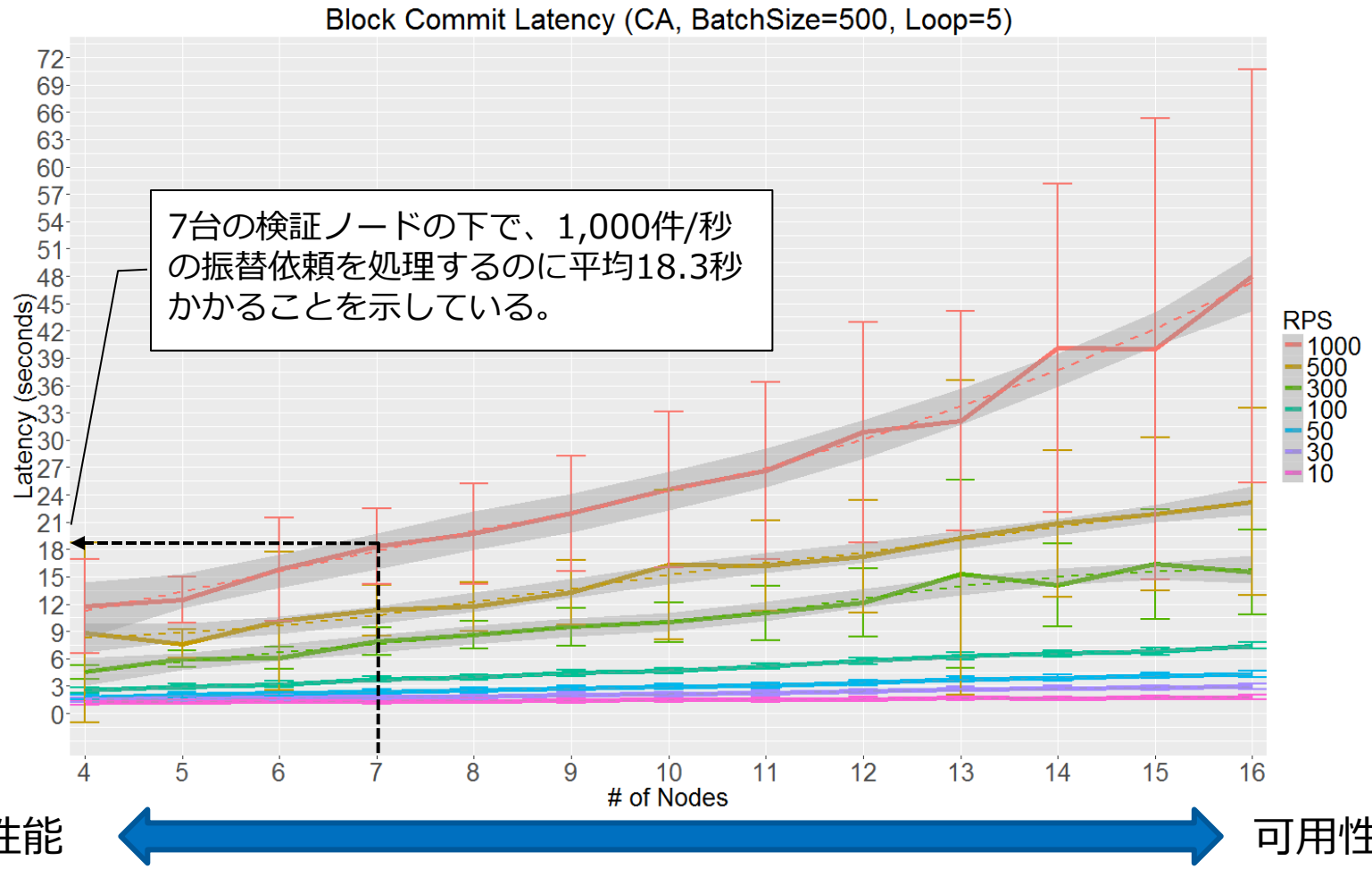
2-2. スマートコントラクトに関する暫定結果

- 日銀ネット同時決済口における流動性節約機能の一部を再現。
 - 時刻起動等の非決定的な処理や非停止性の処理などの実装には留意の必要。



2-3. 処理性能の「傾向」に関する暫定結果

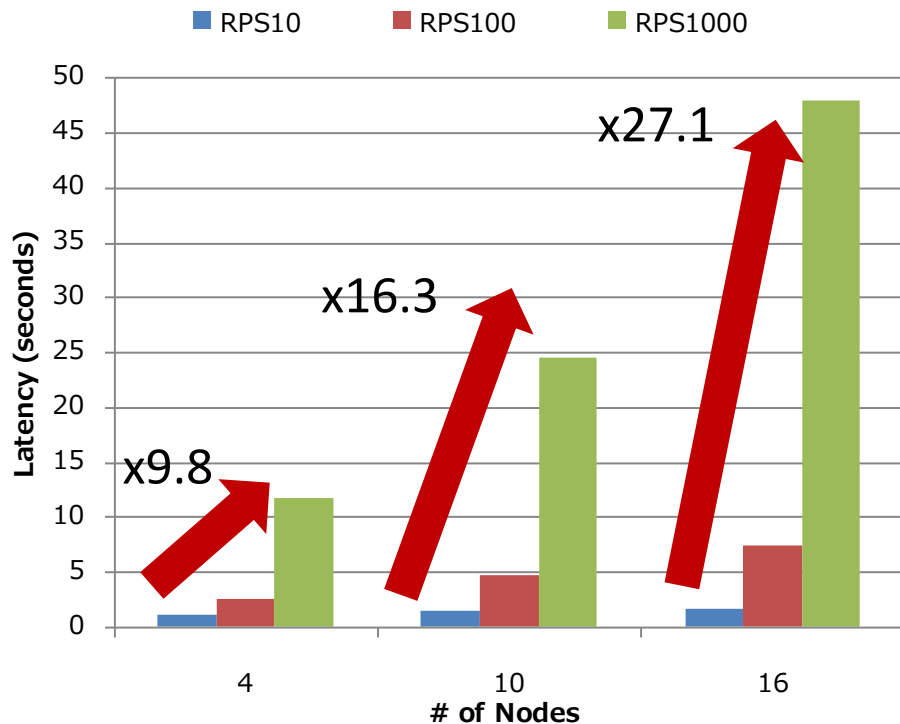
- 処理性能と可用性はトレードオフの関係にある。 可用性向上には、ノード数を増やし、かつ地理的・ハードウェア的にも分散させることが望ましいが、これは処理性能の悪化をもたらす。
 - 検証ノード数の増加に伴いレイテンシが拡大。この傾向は処理負荷が高まるほど顕著。



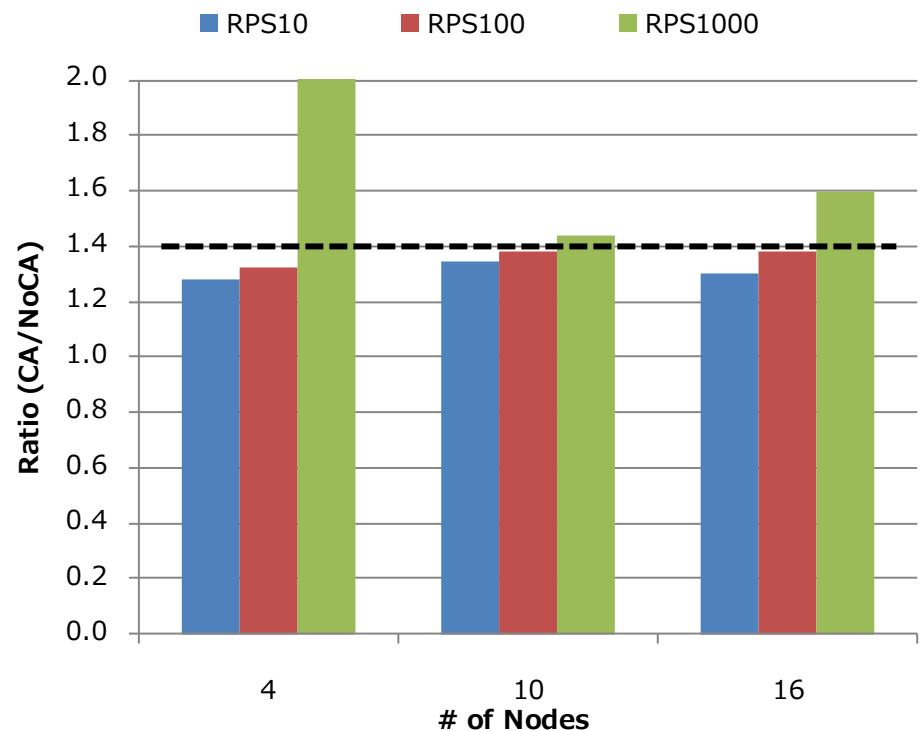
(参考) 処理性能の「傾向」にかかる暫定結果

- 検証ノード数が増えるほど、処理負荷増加による遅延度合いが拡大。
 - CPUがボトルネックとなり、正確な評価に至っていない可能性に留意の必要。
- 認証局は（取引認証書を発行するため）処理性能面ではボトルネックになり得るが、これまでのところその影響は概ね限定的。

Block Commit Latency (CA, BatchSize=500, Loop=5)

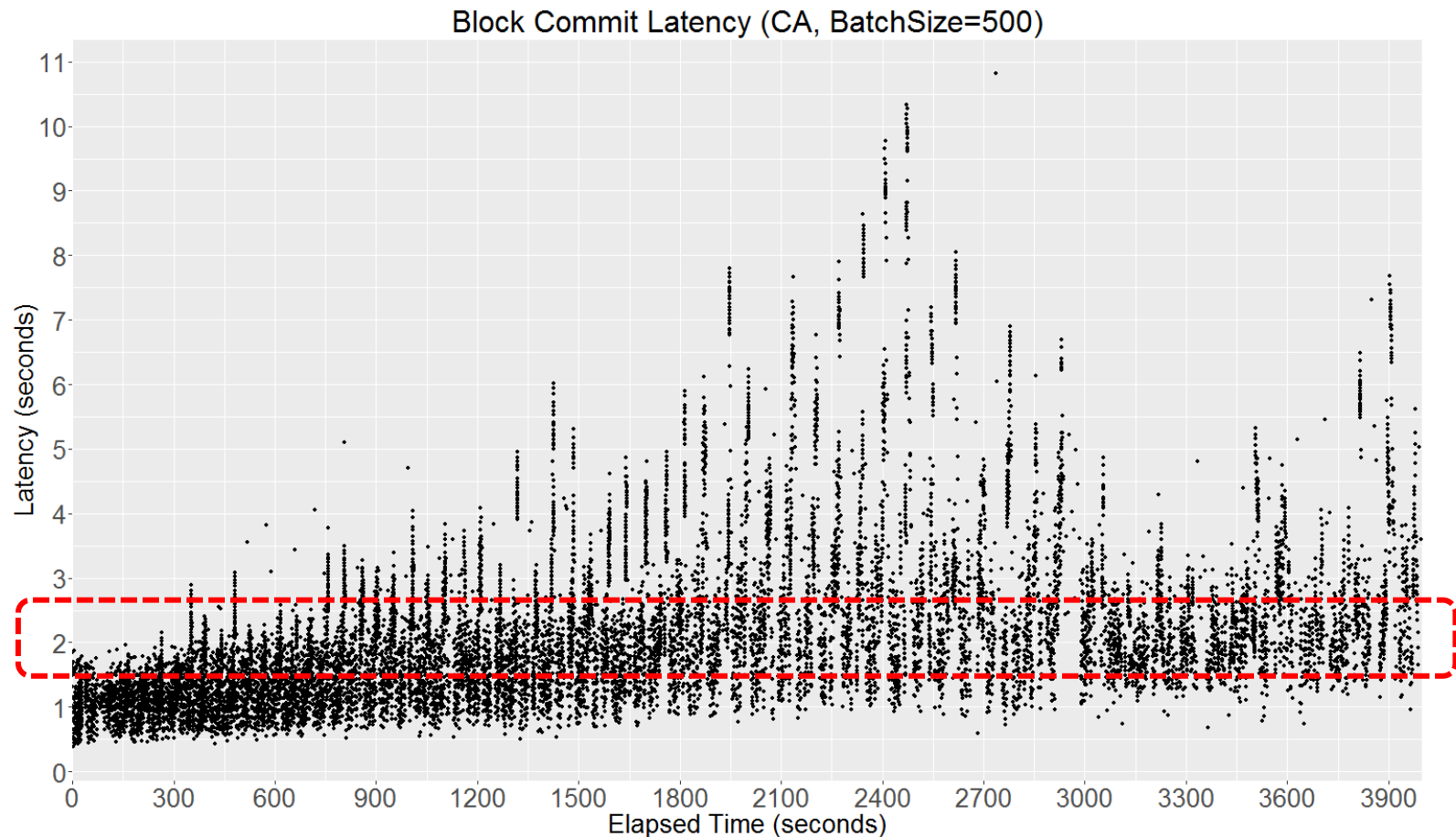


Effect of CA (BatchSize=500, Loop=5)



2-4. 処理性能の「水準」に関する暫定結果

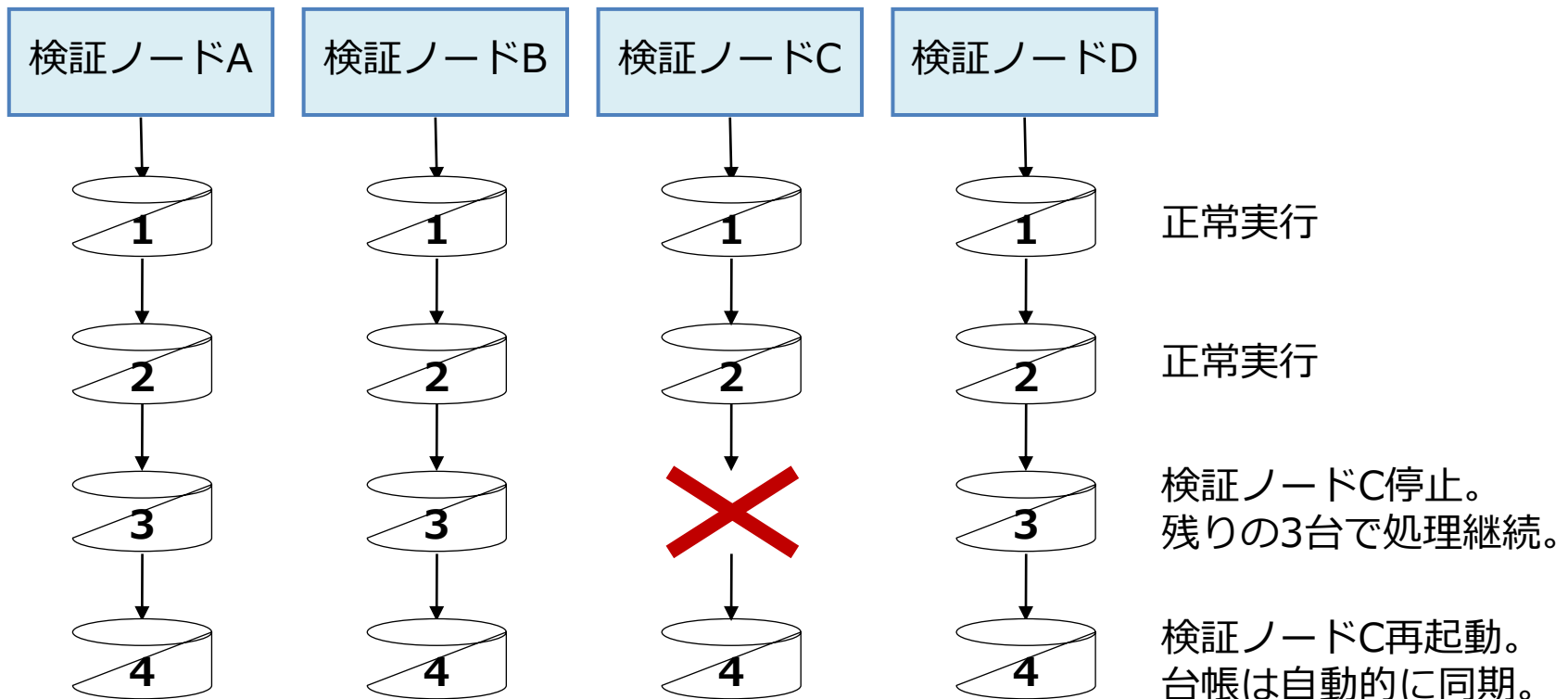
- 2016年3月31日9:15~9:30までの日銀ネット同時決済口データを用いたところ、振替依頼の送信からブロック確定までに要した時間は平均約2秒。
 - ただし、実験環境の制約から全体の取引処理に60分超を要したため、実験環境拡充後に再検証の必要。



2-5. 可用性に関する暫定結果

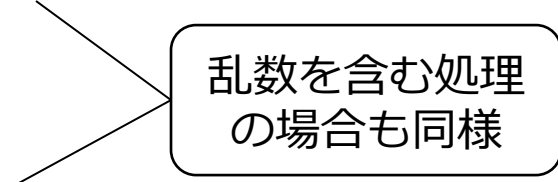
- 検証ノード4台・認証局1台の下では、検証ノードの障害は1台まで許容され、台帳間で同期がとられる。

- 検証ノード1台に障害が発生しても、残りの3台で処理を継続
- 検証ノード2台以上に障害が発生した場合は、処理は中断。
- 障害ノード2台のうち1台が復帰した場合は、障害中に処理されなかったトランザクションが処理され、併せて検証ノード間での台帳も同期。



(参考) 可用性にかかる暫定結果 (検証ノード4台、認証局1台)

- 1ノードの障害 (ハードウェア障害、ネットワーク障害、ビザンチン障害)
 - 残りのノードで処理は実行
 - ※ ただし、ネットワーク障害時は当該ノードには空のブロックが生成される
- 2ノードの障害
 - 処理は中断、ブロックも生成されない
 - ※ ただし、ビザンチン障害1台と他の障害1台の組合せでは、処理は継続するものの、一時的に台帳がずれる場合あり
- 障害中の2ノードからの1ノードの復帰
 - 中断中のトランザクションが即座に処理される
 - ※ 同期は一定間隔毎に行われる。差分同期が難しい場合は台帳ごと差し替えられる
- 認証局の障害 (単一障害点における障害)
 - 新たなユーザでのログインは不可、取引証明書が無くなるとそれ以上のトランザクションの送信は不可
 - 設定ファイルを書き変えて再起動すれば、任意のユーザを追加可能



乱数を含む処理
の場合も同様

2 - 6. DLTにかかる基礎実験 - まとめ

暫定的な結果

- ・ ネットィングなどの複雑な業務処理をスマートコントラクトで実装可能。
- ・ 従来システムに比べ、「処理性能」の面では見劣りするものの、「可用性」の面でメリットをもたらし得る可能性（ただし、処理性能と可用性はトレードオフ）。

今後の取組み

- ・ 評価基準の充実化：主に安全性に関わる面で、さらなる確認を要する点が多い。
- ・ 実験環境の向上：パフォーマンスを左右する要素の洗い出しには、実験環境によるボトルネックを極力排除することが望ましい。
- ・ 基盤技術革新への対応：例えば、Hyperledger Fabricにおける新しいコンセンサスモデルやチェーン間連携など。

など

3 - 1. 金融インフラ分野における許可型DLTの活用可能性 - 仲介者の存在

- 仲介者および単一障害点の存在 (注)

ビットコイン

イーサリアム

Fabric v0.6

Fabric v1.0

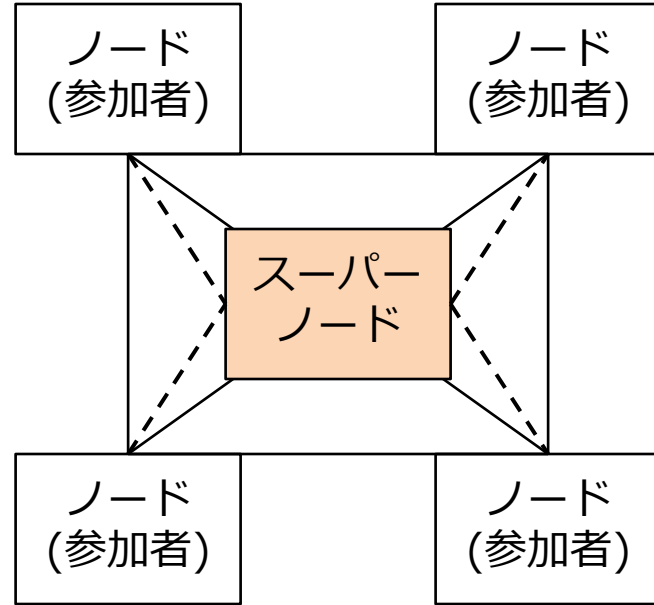
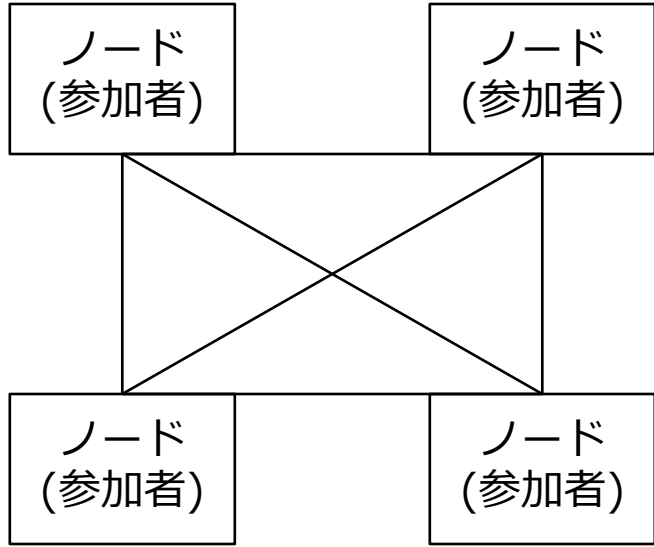
Corda v0.5

仲介者なし

認証局

認証局
Orderer

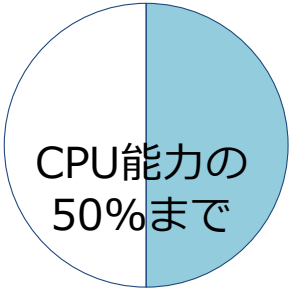
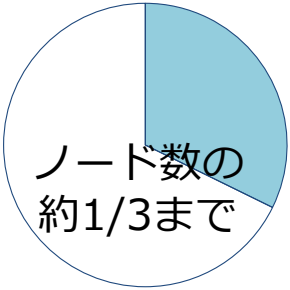
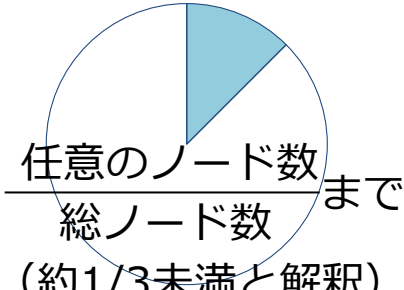
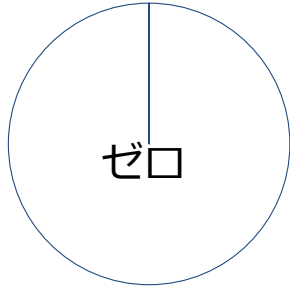
Notary Service,
Network-Map
Service



(注) FabricおよびCordaでは、単一障害点解消のため、仲介者の分散化が検討されている(2017年1月時点)。12

3-2. ビザンチン障害耐性、改竄耐性

- コンセンサスアルゴリズムとビザンチン障害に類する障害への耐性 (注)

| <u>ビットコイン</u> | <u>イーサリアム</u> | <u>Fabric v0.6</u> | <u>Fabric v1.0</u> | <u>Corda v0.5</u> |
|---|---|---|---|-------------------|
| PoW | Ethash | PBFT | 重み付き多数決 ないし条件式 (Endorser間) | RAFT (Notary間) |
| ブロック生成難度の累積値が 最も大きい帳簿が正 | | | | |
|  |  |  |  | |
| CPU能力の 50%まで | ノード数の 約1/3まで | 任意のノード数 総ノード数 まで (約1/3未満と解釈) | ゼロ | |

(注) コンセンサスアルゴリズムが用いられる箇所において、どれだけの割合の障害まで耐えられるかを図示したもの。なお、FabricおよびCordaは代表的なコンセンサスアルゴリズムを挙げている(2017年1月時点)。

- ブロック生成時間と改竄耐性

| <u>ビットコイン</u> | <u>イーサリアム</u> | <u>Fabric v0.6</u> | <u>Fabric v1.0</u> | <u>Corda v0.5</u> |
|---------------|--------------------------|--------------------|--------------------|-------------------|
| 約10分 | 約12秒 (フォーク頻度 の上昇等) | 任意 | 任意 | ブロックという 形は採らない |

3-3. 共有範囲

- 台帳およびスマートコントラクトの共有範囲 (注)

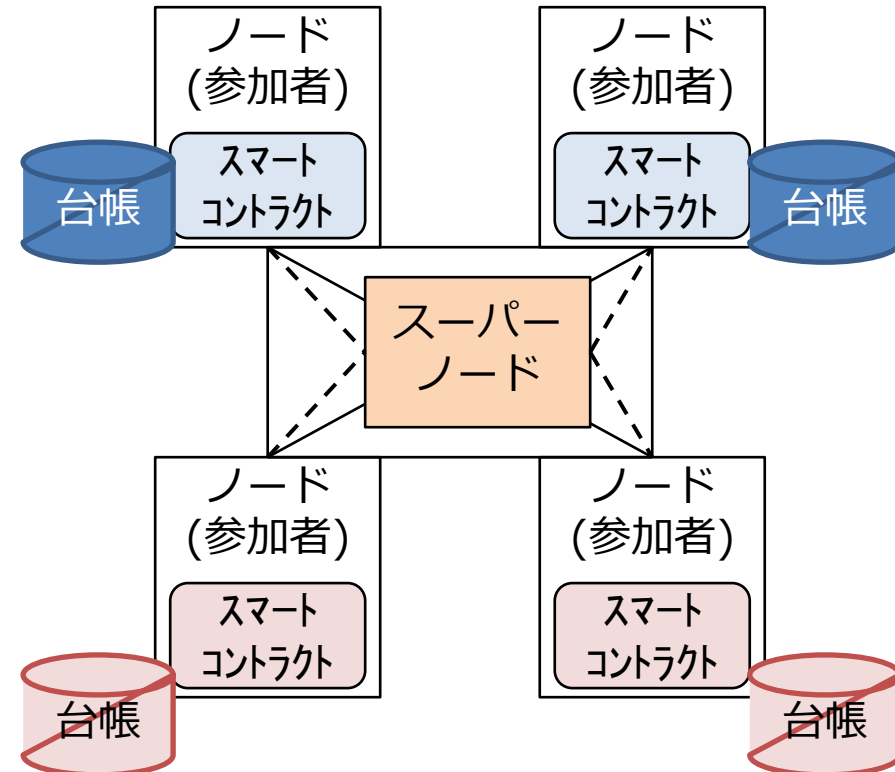
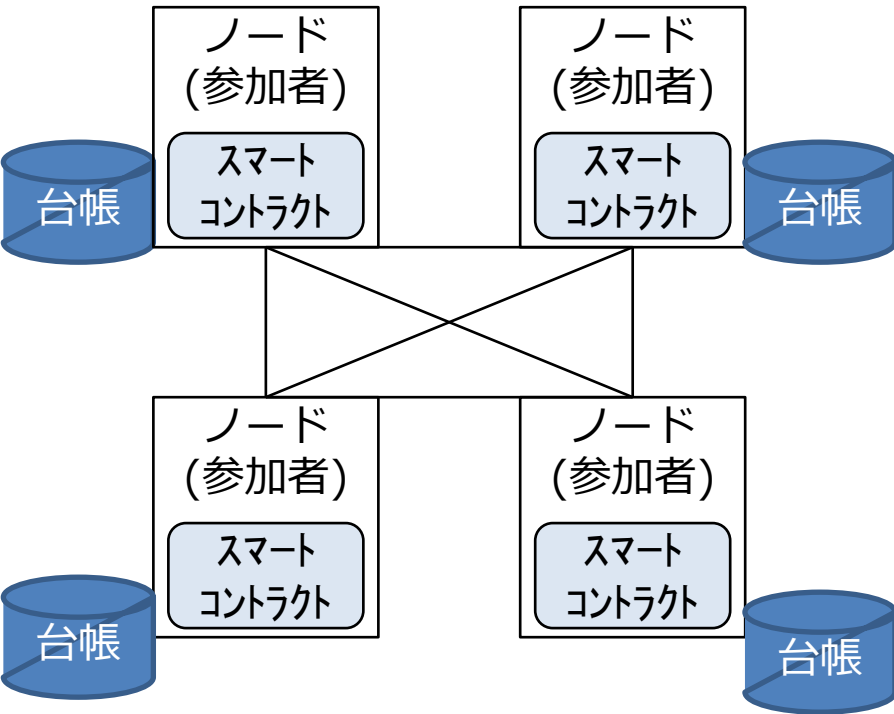
ビットコイン イーサリアム Fabric v0.6

参加者全員での共有 (Global Sharing/Global Execution)

Fabric v1.0

Corda v0.5

関係者に限定した共有 (Local Sharing/Local Execution)



(注) Fabric v1.0とCorda v0.5の差異については、本資料では割愛している。 14

3 - 4. 許可型DLTにおける最近の流れ

| ビットコインを他分野へ適用する際の主な論点 | 許可型DLTにおける最近の流れ |
|--------------------------------------|--|
| ① ビザンチン障害の考慮 | 簡素化 |
| ② 結果整合性の採用 | 強い一貫性 |
| ③ ブロックサイズ/生成時間の制約 | 制約の撤廃 |
| ④ 送信順序と処理順序の相違 | 相違の発生抑制 |
| ⑤ 機能拡充の制約 | 制約の撤廃 |
| ⑥ ストック情報の未検証 | ストック情報の検証追加 |
| ⑦ プライバシーの割り切り | Local Sharing (関係者に限定したデータ共有) |
| スマートコントラクト対応に伴う主な論点 | 許可型DLTにおける最近の流れ |
| ⑧ 非決定性・非停止性の処理の制限 | 当該処理の禁止 (事前ないし事後の確認) |
| ⑨ 処理ノードの冗長性 | Local Execution (関係者に限定したスマートコントラクト実行) |
| ⑩ 直列処理の制約 | ある程度の並列化 |
| ⑪ ライブラリの未成熟 | 枯れた技術の採用 |
| 許可型ネットワークに伴う主な論点 | 許可型DLTにおける最近の流れ |
| ⑫ 単一障害点 | ある程度の許容 |
| ⑬ 改竄耐性の劣化 | 改竄の阻止から改竄の検知へ |
| Local Sharing/Local Executionに伴う主な論点 | 許可型DLTにおける最近の流れ |
| ⑭ 可用性担保の必要性 | ある程度の許容 |
| ⑮ ネットワーキング処理の制約 | ある程度の許容 |

(注) 効率性に関わる面を中心に、ビットコイン・イーサリアム・Fabric・Cordaの状況から取り纏めたもの(2017年1月時点)。

3 - 5. 許可型DLTの方向性

- 金融インフラへの活用に向けて、許可型DLTは以下の方向で検討が進む模様。

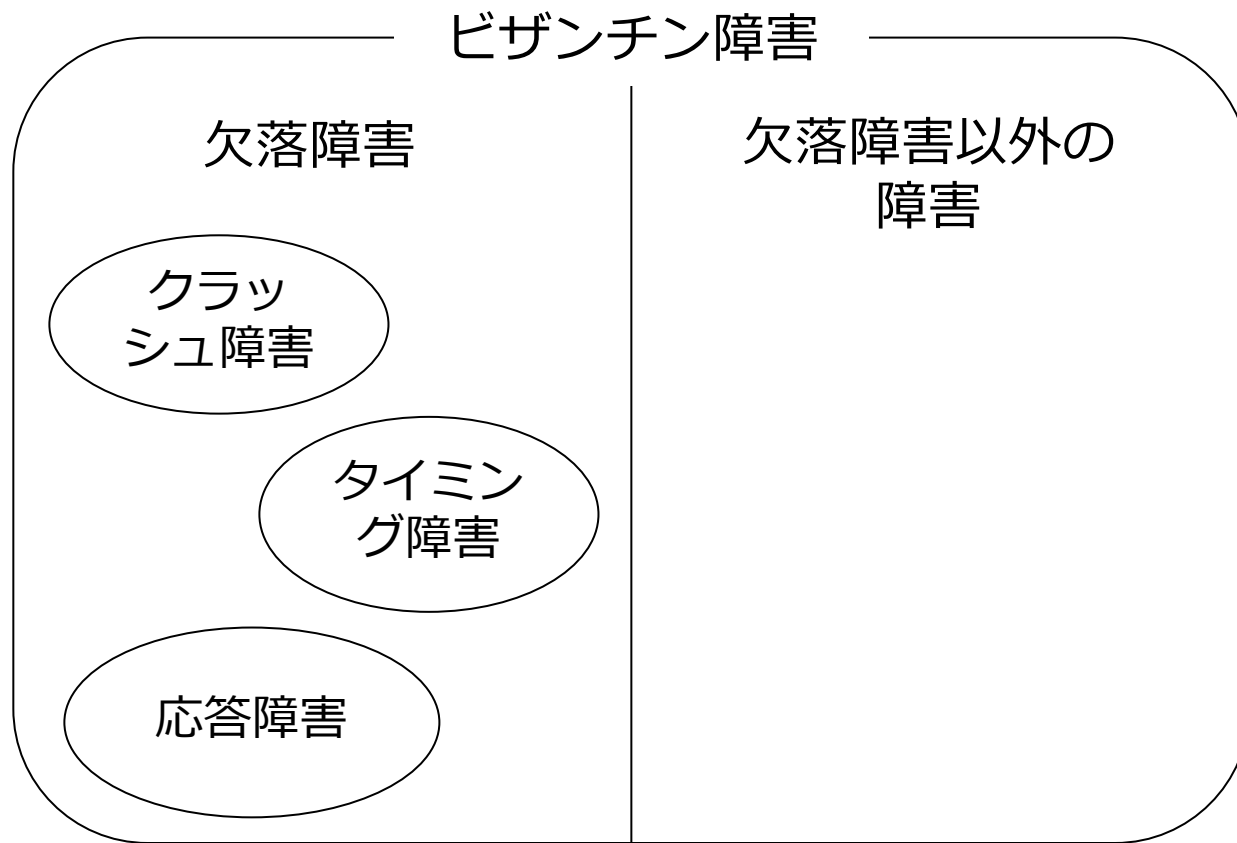
| ビットコインで挙げられる主な特徴 | 許可型DLTの方向性と新たな論点の例 |
|----------------------------|--|
| 仲介者の排除 | 信頼できる仲介者の導入 (単一障害点における可用性等担保の必要性) |
| 不特定多数による協働、 ビザンチン障害耐性 | 信頼できる参加者間での協働で対処 (多重投票の発生を検知する仕組みの必要性) |
| 改竄困難 | 改竄検知で代替 (改竄を検知する仕組みの必要性) |
| 実質ゼロダウンタイム、 参加者全員での台帳共有 | 関係者に限定したデータ共有・スマートコントラクト 実行で代替 (各ノードにおける可用性等担保の必要性) |



- 信頼のおける仲介者の下、相当程度信頼のおける参加者間での、資産や取引毎に関係者を限定したデータ共有・スマートコントラクト実行 (Local Sharing/Local Execution)
 - 今後の取組みでは、効率性・安全性含むトータルでの費用対効果の検証が進むのではないかと考えられる。

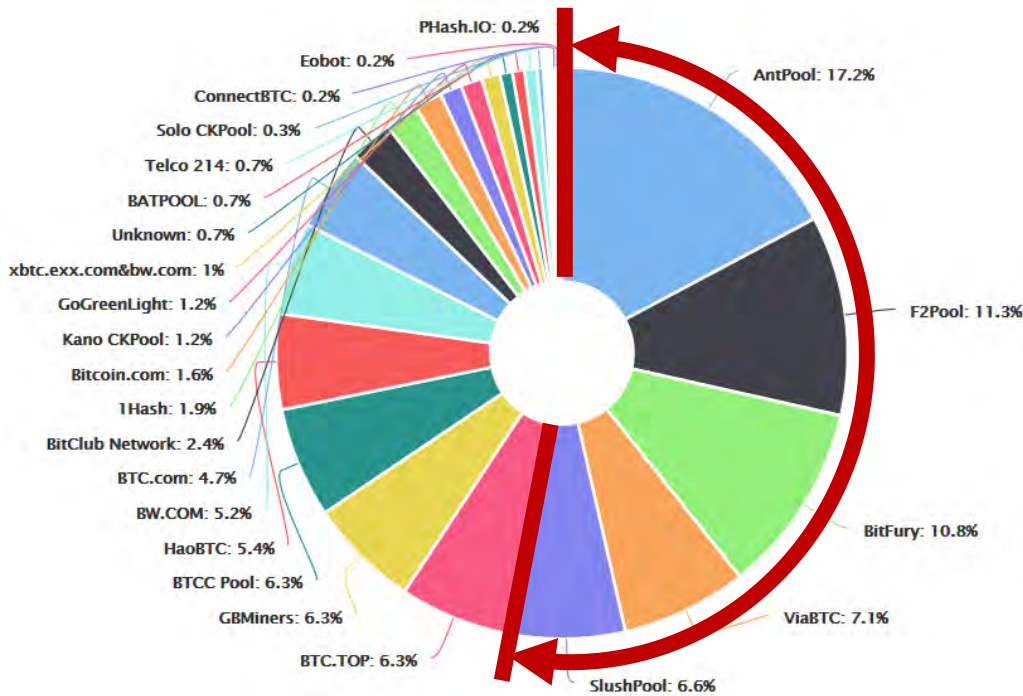
(参考) ビザンチン障害

- 任意障害とも言われ、P2Pネットワーク上で発生し得るあらゆる障害を指す。
- ビザンチン障害耐性が重要とされるユースケースは、ビットコイン以外では、金融以外のミッションクリティカルな分野（Honeywell SAFEbus、NASA Spider等）が知られる。

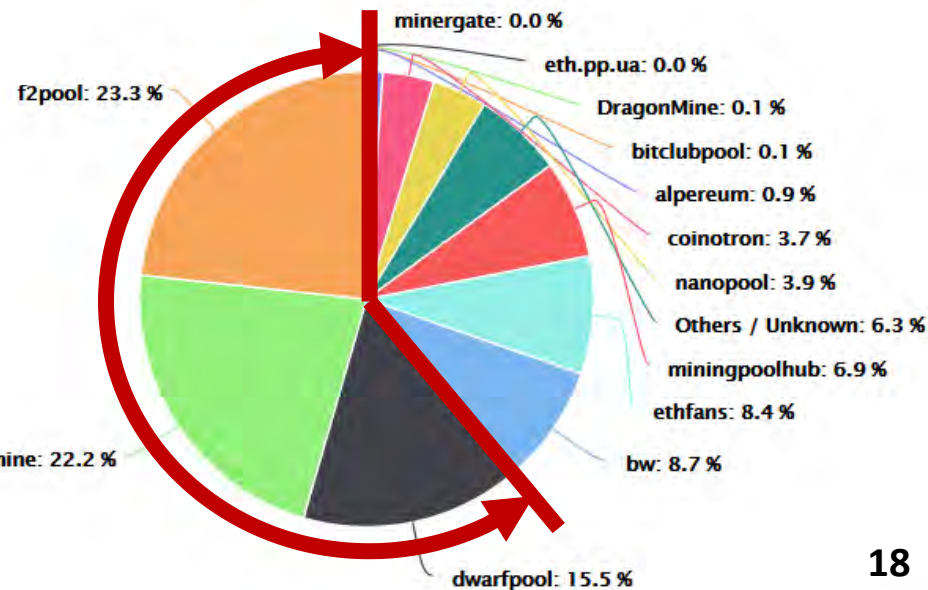


(参考) ビットコインとイーサリアムにおけるマイナーの分布

<ビットコイン>



<イーサリアム>



(注) ビットコインはblockchain.infoより転記。
イーサリアムはetherchain.orgより転記。
(2017年2月21日時点)

3 - 6. 金融インフラ分野における許可型DLTの活用可能性 - まとめ

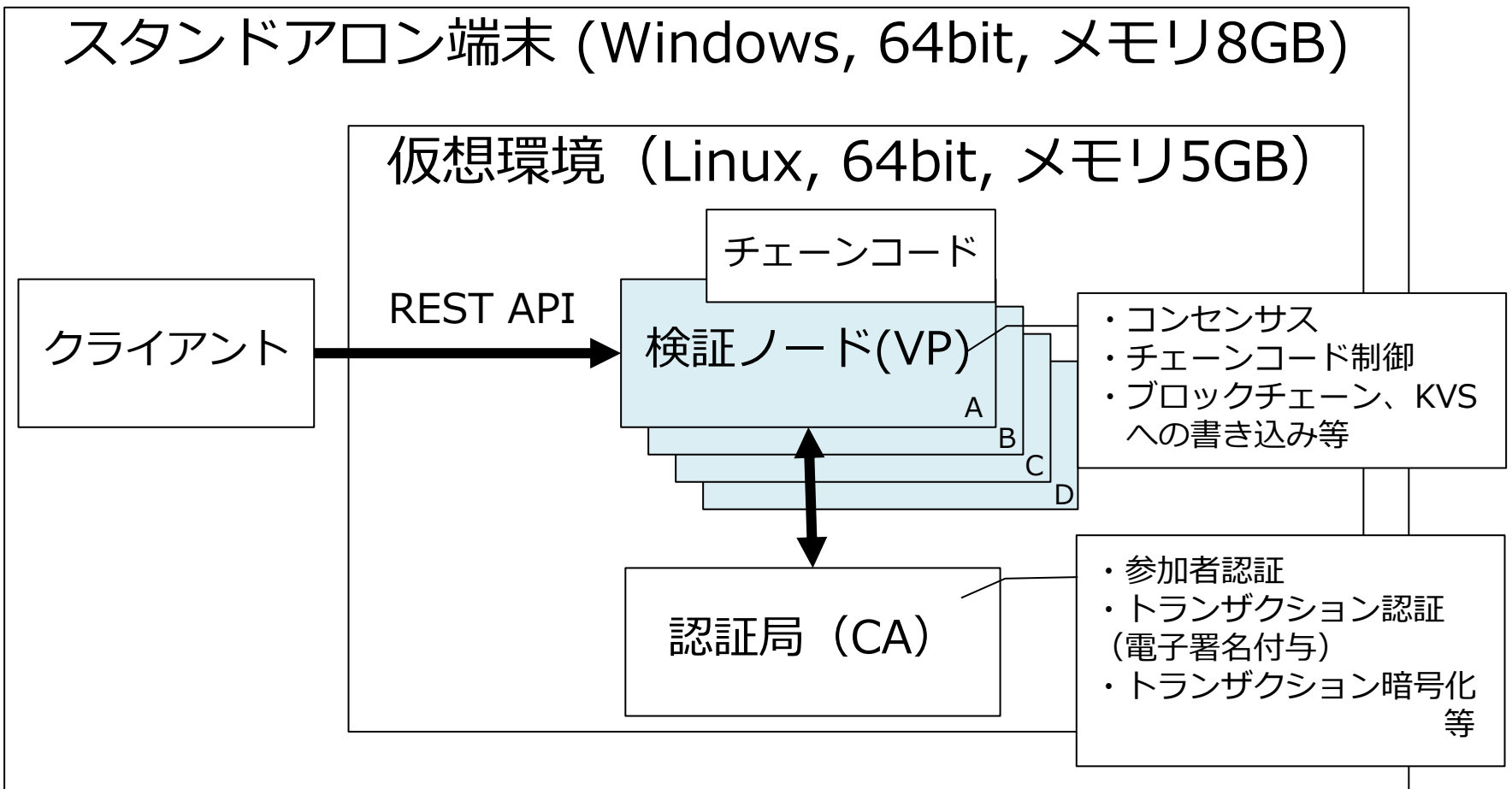
- 実用化が見込まれる多くのユースケースでは、ビザンチン障害耐性の代わりに、（従来システムと同じく）参加者間の信頼で担保する方向と見られる。
- 一方で、その普及にあたっては、従来システムと比べて具体的に分かりやすいメリットを、ユースケースに結び付けていく必要があると考えられる。

金融インフラ分野における許可型DLTのメリットとは何か？

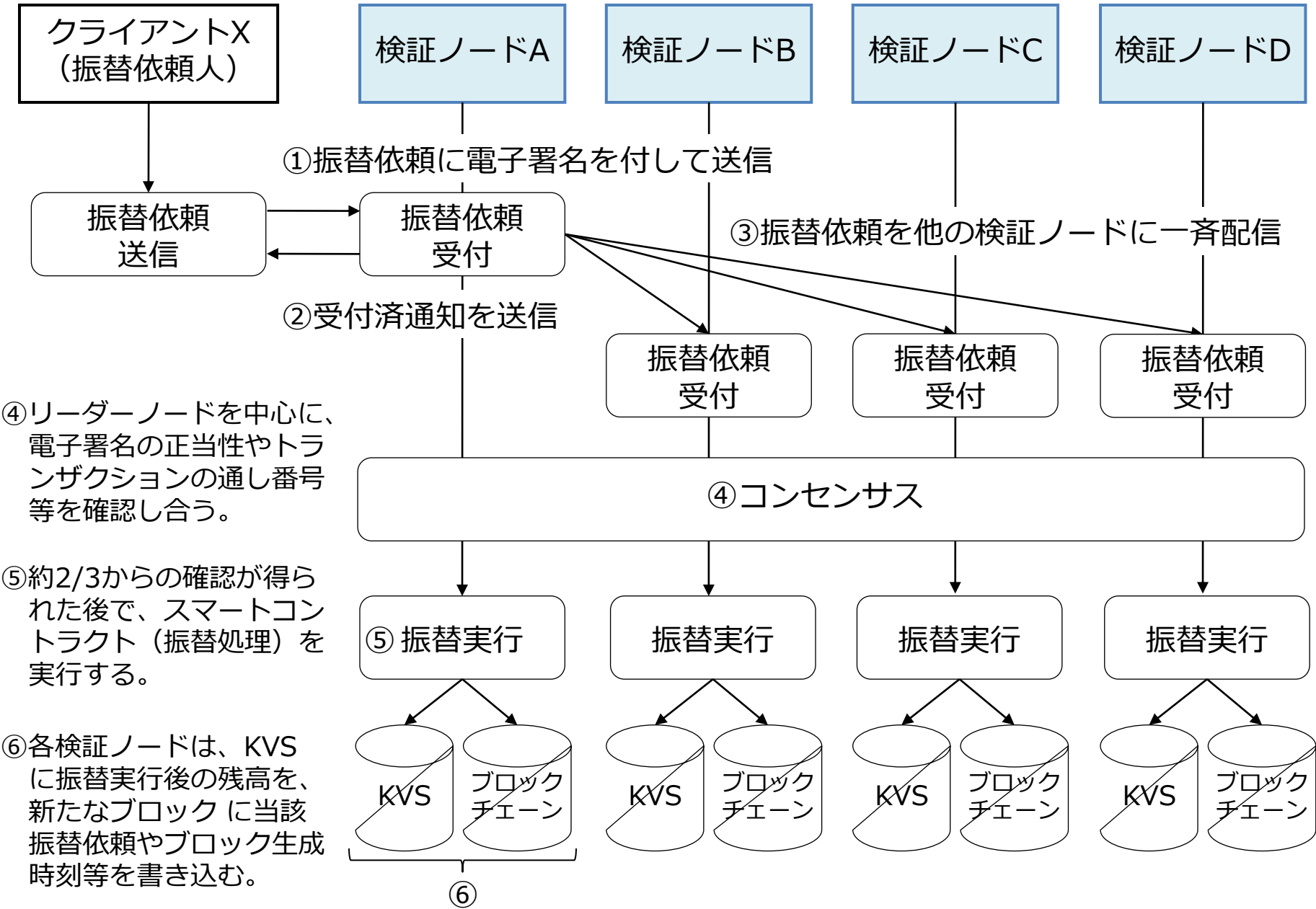
參考資料

基礎実験の環境

- 実験基盤: スタンドアロン端末上の仮想環境
- 検証ノード数: 1 (認証局) + 4~16 (検証ノード)
- コンセンサスアルゴリズム: PBFT (Practical Byzantine Fault Tolerance)



Hyperledger Fabric v0.6.1 での処理の流れ



クライアントX
(振替依頼人)

検証ノードA

検証ノードB

検証ノードC

検証ノードD

①振替依頼に電子署名を付して送信

振替依頼
送信

振替依頼
受付

②受付済通知を送信

③振替依頼を他の検証ノードに一斉配信

振替依頼
受付

振替依頼
受付

振替依頼
受付

④リーダーノードを中心に、
電子署名の正当性やトランザクションの
通し番号等を確認し合う。

④コンセンサス

⑤約2/3からの確認が得られた後で、
スマートコントラクト(振替処理)を実行する。

⑤振替実行

振替実行

振替実行

振替実行

⑥各検証ノードは、KVSに振替実行後の残高を、
新たなブロックに当該振替依頼や
ブロック生成時刻等を書き込む。

KVS
ブロック
チェーン

KVS
ブロック
チェーン

KVS
ブロック
チェーン

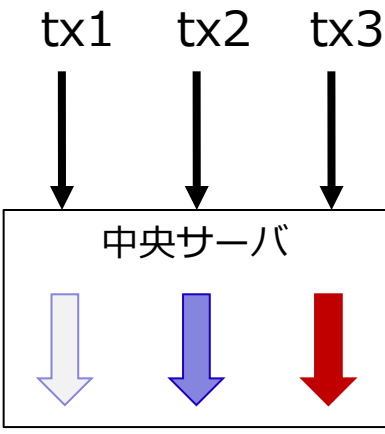
KVS
ブロック
チェーン

⑥

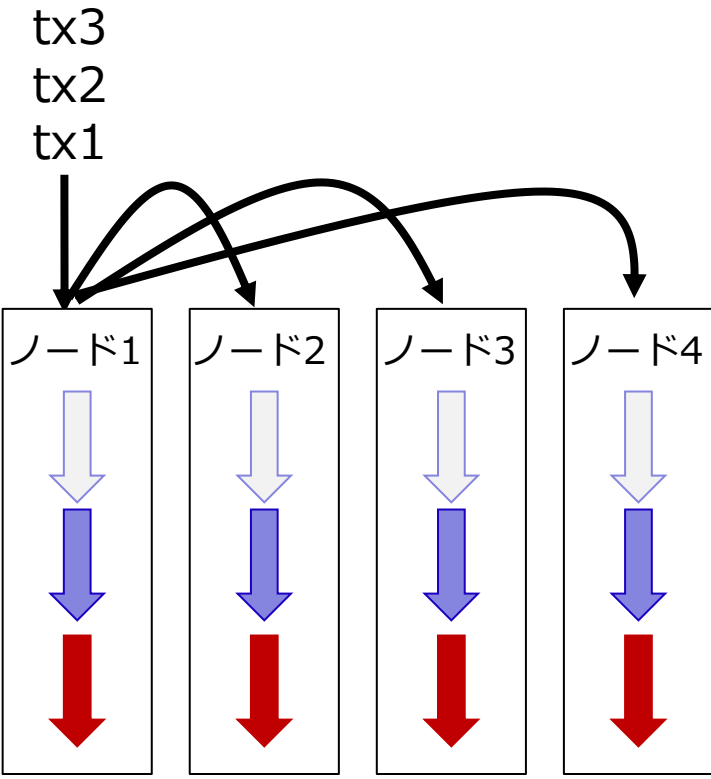
処理内容の制限、処理ノードの冗長性、直列処理の制約

- 必ず同じ結果になることが保証されている処理を、全てのノードが、直列に処理する必要。

<集中管理型システム>



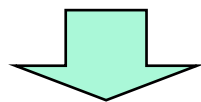
<イーサリアムおよびFabric v0.6>



「触媒」としての役割

「運営者」としての役割

「オーバーシーアー」としての役割



決済システム・金融インフラ

安全性の確保・向上

効率性の向上