

特別賞

コンソーシアム型ブロックチェーン向け
非中央集権型システム運用管理技術 OpsSC

株式会社日立製作所 研究開発グループ

佐藤 竜也 下沢 拓 肥村 洋輔

1. 緒 言

近年、複数企業／組織間で信頼できるデータ共有や取引を可能とする技術としてコンソーシアム型ブロックチェーン(パーミッション型ブロックチェーンとも呼ばれる)が注目されている。Bitcoin[1]やEthereum[2]のような不特定多数の参加者からなるパブリック型ブロックチェーンとは異なり、コンソーシアム型ブロックチェーンでは、許可された組織間(コンソーシアムを形成)のみで限られた取引範囲を構築し、高い取引性能を達成することが可能である。主要なコンソーシアム型ブロックチェーンでは、契約や取引業務プロセスをプログラムで記述して自動実行する「スマートコントラクト(以下、SC)」を備え、組織間で合意形成しながらSCに記述された業務を自動処理することができる。これにより、従来の暗号資産だけでなく、貿易金融やサプライチェーンといった複数組織間での様々な業務プロセス改革による期待が高まっている。貿易金融では、国境を越える紙ベースの文書のやり取りをデジタル化することで、プロセスの効率化と透明性の向上が可能となり、これにより取引の速度を上げコストを削減することができる。サプライチェーンでは、生産から販売に至るまでのプロセスが透明化され、リアルタイムでの情報共有が可能となり、在庫管理の最適化、生産計画の精度向上、問題発生個所の特定などが容易となる。このように、コンソーシアム型ブロックチェーンは社会課題の解決に貢献する技術として期待が寄せられている。Linux Foundation[3]下で運営されるエンタープライズ向けブロックチェーン技術開発のオープンソースコミュニティであるHyperledger Foundation[4]のプロジェクトとして管理されるHyperledger Fabric[5][6]が最も有名なコンソーシアム型ブロックチェーンであり、様々な業種でB2Bユースケースを中心に実証実験や本番プロジェクトで活用されている。

コンソーシアム型ブロックチェーンを用いたシステム(以下、コンソーシアム型ブロックチェーンシステム、あるいは、単にブロックチェーンシステムと呼ぶ)の本番利用を実現するためにはシステム運用管理の確立が必須となる。一般に、システム管理とはITシステムを安定稼働させるために必要な作業群のことであり、システム運用とはシステム管理のために行う個々の作業のことである。システム運用の例としては、ソフトウェアのインストール、インストールされたソフトウェアのバージョン更新、新規ユーザのサーバ追加、バックアップの作成、バックアップからのリストアなどがある。ブロックチェーンシステムはまだ発展途中でありシステムアーキテクチャのような実行系に関するトピックが優先されていたため、運用管理はこれまで十分注目されてこなかった。しかし、最近になり少しずつ運用管理の重要性が徐々に認識され、体系化や整理に向けた取り組み[7]や研究[8]が登場している。

ブロックチェーンシステムは複数組織にまたがって構成され、非中央集権的にトランザクションを実行可能という価値を提供する。この価値を保つためにはシステム全体の非中央集権化が求められるが、運用管理面ではまだ中央集権的な側面が残っていた。システム運用が中央集権的になってしまうと、単一組織がシステム全体をコントロール可能となり、ブロックチェーンのデータ改ざんのリスクをはらむため、ブロックチェーンの導入目的や価値が損われる恐れがある。そのため、ブロックチェーン上で複数組織にまたがるシステム運用を非中央集権的に実施することが不可欠であり、その実現が大きな課題である。この数年間でのコンソーシアム型ブロックチェーンの進化に伴い、運用管理における非中央集権性の重要性も認知され始め、その実現が必須になっている。具体的にはHyperledger Fabricでは、これまで個々の管理コマンドに含まれていた中央集権的な要素の排除に向けた取り組みが進展

している[9]。さらに[10]ではシステム全体を含めた完全な非中央集権化のために不足している部分を明らかにし、この実現のためには運用管理が大きな鍵の一つとなることを示している。

我々はこの課題に世界に先駆けて着眼し、OpsSC と呼ばれる分散型組織間運用管理技術を開発してきた[11][12][13]。本技術を用いることで、組織をまたがる運用作業を非中央集権的かつ均一な手順や設定パラメータで効率的に実行可能となる。さらに、本技術の活用により、実行系だけでなく管理系も含めて、完全に非中央集権化されたコンソーシアム型ブロックチェーンシステムを実現することが可能である。本技術は主要なコンソーシアム型ブロックチェーンプラットフォームである Hyperledger Fabric 向けに実装、オープンソースソフトウェアとしてコミュニティに公開しており[14]、本技術開発を通じてコンソーシアム型ブロックチェーンおよび Hyperledger Fabric の発展および本番利用の促進に貢献する。

2. コンソーシアム型ブロックチェーンシステムとその運用における問題点

2.1 コンソーシアム型ブロックチェーンシステムの概要

図1に本研究で想定する典型的なコンソーシアム型ブロックチェーンシステムを示す。この図に基づいてブロックチェーンシステムがいかにより複数組織間でのデータ共有と処理を非中央集権的に実現するかを説明する。

ブロックチェーンシステムは、主にブロックチェーンプラットフォーム部分（主に Peer などのノード）と、SC を含むアプリケーション部分から構成される。SC は、事前に取り決められた業務ロジックをコード化したものであり、ブロックチェーン上で自動的に実行される。

スマートコントラクト(SC)により組織が互いに合意しながら各ノードで同一ロジックを确实実行可能
→ **ブロックチェーン導入価値**: 複数組織間で非中央集権的にトランザクション実行やデータ共有

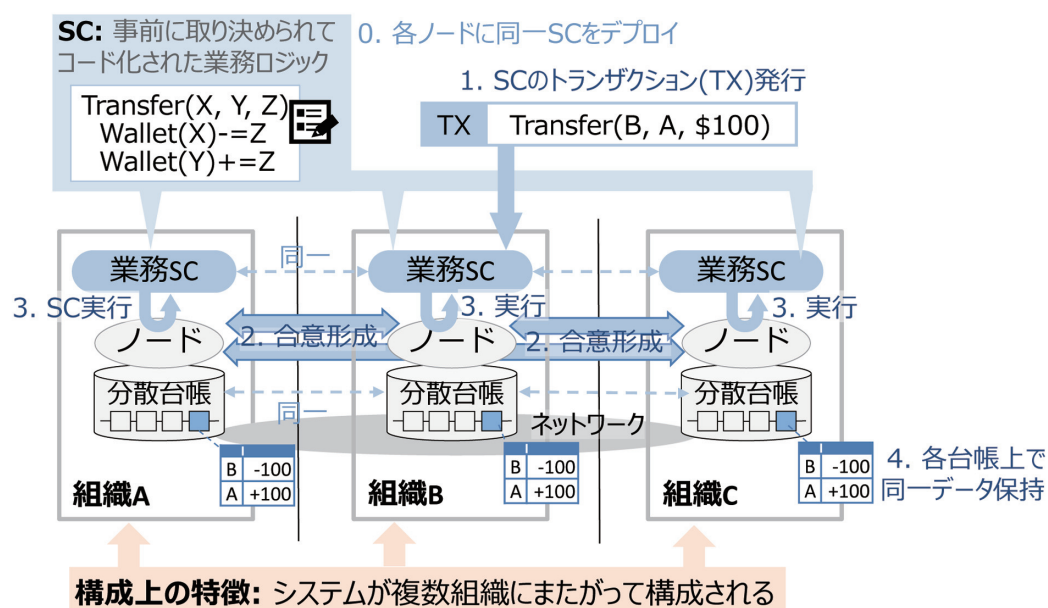


図1 コンソーシアム型ブロックチェーンシステムの概要

各組織がそれぞれノードを持ち寄ってブロックチェーンネットワークを形成する。これにより、単一のブロックチェーンシステムが複数組織にまたがって相互に利用される構造が実現される。

典型的なトランザクション処理の流れを説明する。SC に対するトランザクションが発行されると、ネットワーク内の各組織のノード間で合意形成を行いながら各組織のノード上で SC が実行される。トランザクションが受け入れられた場合にはその実行結果をブロックという単位でまとめて各組織のノード上の分散台帳に追記する。ブロックをハッシュで連結し一度追加されると変更不可 (Immutable) とすることで耐改ざん性が確保される。また、ネットワーク内の全ノードが同一のデータを保持することで、データの一貫性と透明性が保証される。このようにしてブロックチェーンは複数組織間で非中央集権的にトランザクション実行やデータ共有が可能という価値を提供する。なお実際には、これらの内容はブロックチェーンプラットフォームによって多少の違いがあるが基本的な部分は概ね共通である。

2.2 コンソーシアム型ブロックチェーンシステムの運用における問題点

ブロックチェーンシステムの場合、システム運用には以下の2種類がある：

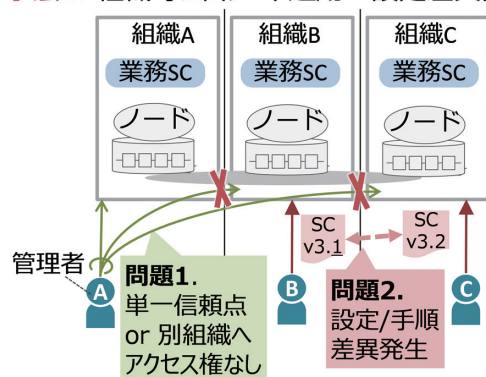
- ・ 単一組織に閉じるシステム運用(従来型のシステム運用)
 - 例：各組織のポータルアプリ(フロントエンド)の修正
- ・ 組織またぎのシステム運用：複数組織間での協調動作が必要となる運用(ブロックチェーンシステム特有の作業)
 - 例：ブロックチェーンネットワーク上の SC 更新(全組織が同じタイミングで同じ SC をデプロイする必要がある)

従来型の運用管理ツールは、ジョブ管理サーバ、Infrastructure as Code(IaC) ツールなど数多く存在する。これらは単一組織に閉じる運用の効率化には有効だが、組織またぎの運用には対応していない。

図2左側には、組織またぎの運用を実施する従来手法とその問題点を示している。一つ目の方法として、単一の管理者が全組織のノードを運用する手法(手法1)が考えられる。この方法の問題点は、管理者がブロックチェーンネットワークにおける単一信頼点となり、すなわち運用が中央集権化してしまうことである(問題1)。これは、この管理者が悪意を持って行動する場合にブロックチェーンのデータを改ざんするリスクを生じさせる。また、実際には、ある組織の管理者が他の組織が所有するノードに対するアクセスすることは通常許可されないため、そもそもこの方法は実行が困難である。二つ目の方法としては、各組織の管理者が自組織のノードをそれぞれ運用する手法が挙げられる(手法2)。この手法では、組織間での運用のずれ(運用手順、タイミング、設定パラメータのずれ)が生じる可能性があり、それに伴ってシステム全体が機能しなくなる可能性がある(問題2)。例えば、SC の更新を考えた場合、SC のバージョンや更新タイミングに関する設定パラメータが組織間で統一されていないと、SC が一時的に使用不可になる(間違ったバージョンを使用している組織がある場合)リスクや、ブロックチェーンネットワーク上で長期間にわたって新しい SC バージョンに更新されない状態が発生するリスクがある。ユーザ視点ではこれらはサービス品質の低下や不安定性につながることを意味する。

従来手法

- 手法1. 単一管理者が運用→運用中央集権化
手法2. 組織毎に自ノード運用→設定差異発生



提案手法: OpsSC

【補足】SC × BC: 業務ロジックを組織間で合意形成しながら確実に実行

運用ワークフローをスマートコントラクト(SC)として定義、各組織はそのSCに従い自ノード運用

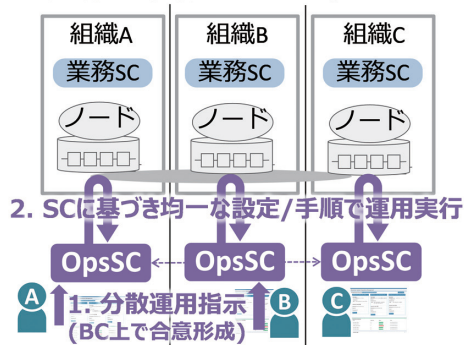


図2 OpsSC の概要

3. 提案技術：Operation Smart Contract (OpsSC)

3.1 OpsSC のコンセプト

前節に述べた問題を解決するため、我々は運用ワークフローを SC として定義し、各組織（の管理者またはエージェント）がこの SC に従って各自のノードを運用することを特徴とする、非中央集権型の組織間運用手法を提案する [11]。SC 上に定義された運用ワークフローには、具体的には、運用手順（コマンド列等）、設定パラメータ、実行タイミング情報が含まれる。本提案手法により、組織またぎの運用作業を特定の組織の意思決定に依存することなく（すなわち非中央集権的に）均一な手順 / 設定パラメータで効率的に実行することが可能になる。「運用そのものを SC として定義する」というコアアイデアに基づき、本提案手法を Operations Smart Contract (OpsSC) と名付けている。

図2右側に示される OpsSC の概念図を用いて処理の流れを説明する。各組織のノードは、OpsSC に対する起動トランザクションを受信すると、ブロックチェーン上で互いに合意形成を確立し、SC 上で設定パラメータを共有およびワークフローの制御を行う。各組織のノードは SC からの指示に基づいてそれぞれ自組織のノードに対して運用を実行する。以上のように、本提案手法では、ブロックチェーン元来の特性である合意形成メカニズムを活用することで単一信頼点や認証情報の共有なしに組織またぎの運用を可能とし、SC 元来の特性を活用することで均一な設定パラメータおよび手順による運用実行を実現する。

3.2 Hyperledger Fabric 向け非中央集権型システム運用ツールとしての OpsSC

我々は OpsSC を Hyperledger Fabric 向け運用ツールとして開発している [12] [13]。Hyperledger Fabric (以下、単に Fabric と呼ぶ) [5] [6] は、最も主要なコンソーシアム型ブロックチェーンプラットフォームのオープンソースソフトウェアであり、エンタープライズ利用に求められる要件に対応した機能 / 非機能を提供し、様々な業種で B2B ユースケースを中心に活用されている。

Fabric の運用管理について述べると、各種管理コマンドが提供されており、各組織の運

用管理者はこれらのコマンドを組み合わせて運用作業を遂行する。2019年頃にv1.4系からv2.0系に進化する過程で一部管理コマンドに存在していた中央集権部分が排除された。このようにコミュニティにおいても運用管理に関しても非中央集権性を保つことの重要性が認識されつつある。一方、実際の運用作業は依然としてこれらのコマンドを組み合わせて実施する必要があり、その中で各組織が行うべき作業やその際に生じる組織またぎでの調整作業（例：情報共有 / パラメータ合わせ）が残っている。

そこでEnd-to-Endの運用ワークフローを非中央集権かつ自動的に実行可能にするためにOpsSCを適用する。開発したツールでは、現在までにFabricを用いたシステムで共通のかつ不可欠であるブロックチェーンネットワーク管理の典型的な運用フローを効率化する機能を提供する。

- ・ チェーンコード (Fabric の SC) 運用：チェーンコードのデプロイやアップデートをサポートする
- ・ チャネル (Fabric のサブネットワーク) 運用：チャネルの新規生成、チャネルに対する組織追加 / 削除、チャネルの設定更新などをサポートする

図3には、Fabricの典型的なブロックチェーンネットワーク運用におけるOpsSC導入前後の比較を示す。Fabric単体では個々のタスク（コマンド）レベルでのみ非中央集権的な実行がサポートされている。そのため、従来の典型的な運用フロー（図3左）では、まず(1)各組織の管理者がOff-chain（すなわちブロックチェーンネットワーク外）で設定パラメータを共有・調整する。その後、(2)各組織の管理者が適宜、他の組織とタイミングなどを合わせながら(1)で調整したパラメータを使ってコマンドを手動実行する。

OpsSCではこのようなEnd-to-Endの運用ワークフローを非中央集権かつ自動的に実行することができる。典型的なユースケース（図3右）では、まず(1)各組織の管理者がOn-chainで、具体的にはOpsSC用チェーンコード上で設定パラメータを調整 / 共有する。典型的にはある組織の管理者が運用を提案し、他の組織がそれに対して賛成 / 反対することで執り行われる。(2) OpsSC用チェーンコードは所定の組織からの賛成（例えば多数決）が得られたことを確認し、提案された運用に対するイベントを発行する。各組織のために用意されたエージェントプログラムがOpsSC用チェーンコードからの指示（イベント）を受け取り、それに基づ

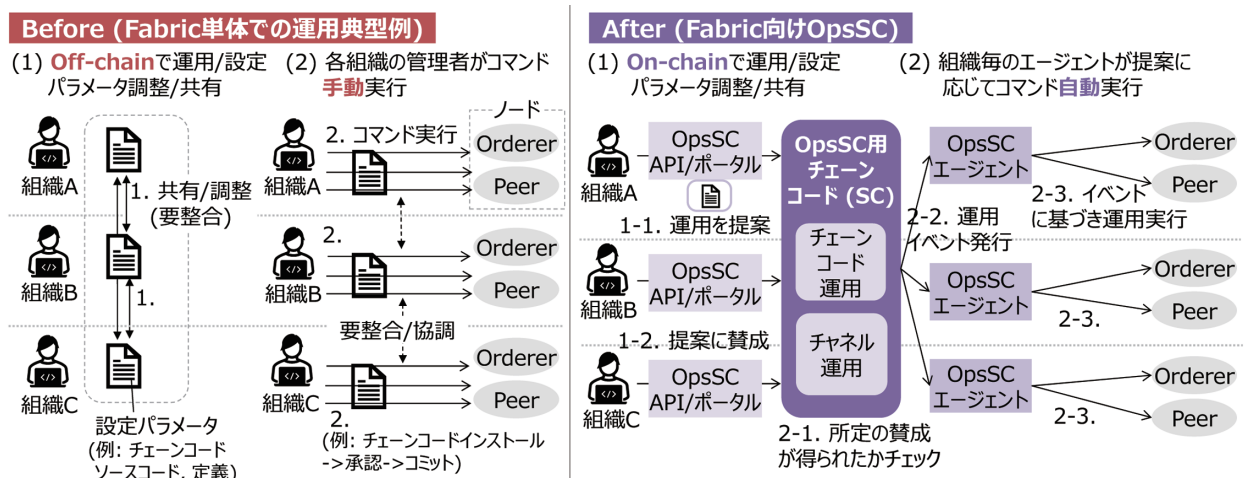


図3 Hyperledger Fabric 向け OpsSC

いて自組織のノードに対してコマンドを自動的に実行する。

FabricにOpsSCを導入することで、End-to-Endでの運用フローの非中央集権性を実現し、手順／設定パラメータの均一性の確保が可能となるが、これらの効果に加えて、運用の効率も大幅に向上するというプラスのベネフィットがある。この点を明らかにするために、Fabricの典型的なブロックチェーンネットワーク運用シナリオを対象に、提案手法（OpsSC利用）と従来手法（スクリプトベース運用）を運用工数削減の観点から比較分析評価した。従来手法では、管理者が手動で運用ステップを実行する方法が採用されているが、各ステップが合理的に整理され、スクリプトによって自動化されていることを前提とし、単純な手動運用に比べて最大限の効率性を確保している。これにより提案手法と従来手法との間でフェアな比較基準を設けている。本評価では、各運用シナリオにおける両手法のコストを、運用ステップ数ベースでモデル化し、それに基づき試算を実施した。試算結果は図4に示すとおりであり、提案手法が従来手法に比べて、運用工数を顕著に削減できることが確認された。

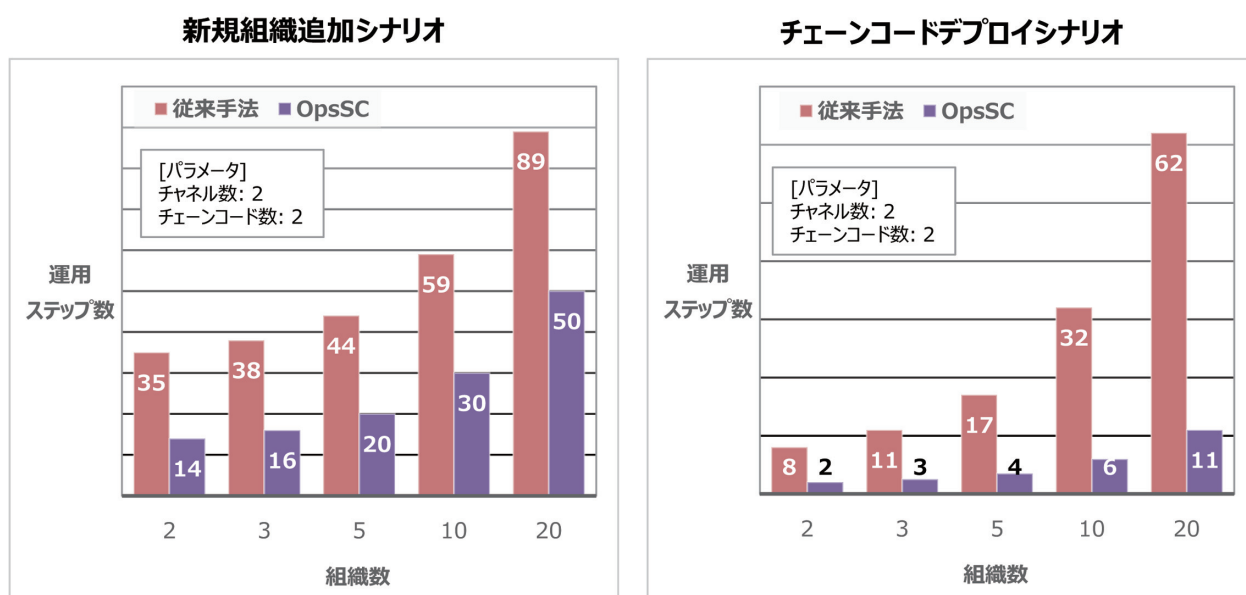


図4 評価結果([12][13]より抜粋)

4. 本技術の普及活動

我々はFabric向けのOpsSC実装をオープンソースソフトウェアとして公開し[14]、2021年2月にHyperledger Foundation内の実験的プロジェクトをホストするHyperledger Labs[15]の一プロジェクトとして提案／承認された。オープンソースソフトウェア化後も、Fabric本体の開発状況に追従しながらエンハンスとメンテナンスを継続し、2024年3月時点でのFabricの最新Long Term Support版であるv2.5系をサポートしている。

2024年3月現在、OpsSCはHyperledgerコミュニティの中で認知されており、図5に示すとおりFabricのエコシステムの一部としても位置付けられている。

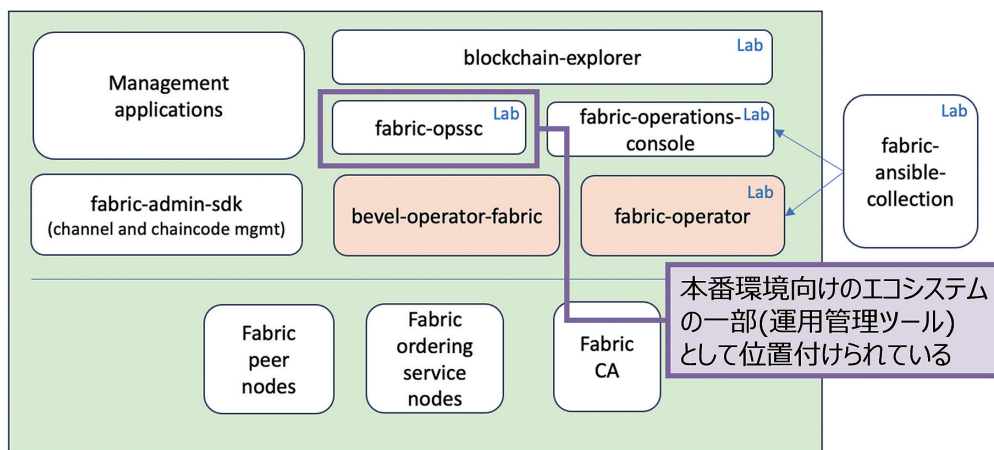


図5 Fabric Ecosystem 中での位置づけ ([16]より引用 / 注釈追加)

5. 今後の構想

OpsSC には、コンソーシアム型ブロックチェーンの普及 / 発展および安定稼働に向けた潜在的な可能性があると考えられる。この文脈において、我々は以下の発展を構想している。

Fabric エコシステム内での発展/成熟. Fabric では、次期バージョンである v3 にて BFT Orderer と呼ばれる機能が実装され、トランザクション順序決定サービスについてビザンチン障害耐性がサポートされる [17][18]。これにより完全に非中央集権的なシステムを構築可能となるため、OpsSC のような仕組みがより重要となる。今後は OpsSC でも BFT Orderer を利用したシステムにおけるチャネル運用をサポートなどの対応を進めていき、エコシステムの一部として Fabric とともに発展していきたい。

様々なブロックチェーンプラットフォームへの適用. OpsSC は、Fabric 以外の SC ベースのコンソーシアム型ブロックチェーンにも汎用的に適用可能であると考えられる。例えば、Ethereum 領域では、コンソーシアム型実装として注目を集めている Hyperledger Besu [19] への適用が考えられる。

様々な運用管理プロセスへの拡張. OpsSC はもともと組織またぎでの運用実行に焦点を当てた仕組みである。しかし、OpsSC のコンセプトは他の運用管理プロセスにも応用可能であると考えられる。表1には、ブロックチェーンシステムで検討すべき運用管理プロセスを列挙している。表中では実行時に関わる組織に応じて「単一組織」「組織横断」「両方」という分類を行っており、「組織横断」と「両方」が OpsSC の対象になり得る。もちろん、単純な適用だけでは不十分であり各運用管理プロセスに特化した設計が必要となるが、我々の提案した「運用ワークフローの SC 化」というコアアイデアはシンプルかつ非常に強力であり、これを足掛かりにすることで全部解決できるポテンシャルがあると考えている。例えば課金管理では、OpsSC の概念を拡張してコンソーシアムにおける各組織の貢献度とリソース消費量を計算する課金用 OpsSC を用意することで組織横断での公平性のある課金を実現できる。別の例として可用性 / キャパシティ管理に関しては、監視用 OpsSC を用意し、各組織が他組織ノードの健全性(死活、性能など)を監視した結果をそれぞれ SC 上に登録、登録結果を相互検証することで、客観性のある情報に基づいたシステム全体としての健全性監視を実現

できる。このように各運用管理プロセスに対しても検討を進め、将来的には全プロセスをカバーできるように段階的に拡充することで、運用のさらなる高度化／自動化を図る予定である。

表1 ブロックチェーンシステムで最終的に検討すべき運用管理プロセス一覧

区分	管理機能/プロセスの分類	概要	タイプ
ブロックチェーンプラットフォーム/ネットワーク	チェーンコード運用管理	チェーンコードのデプロイ、アップデートなどを行う。	組織横断
	チャネル運用管理	チャネル生成、チャネルに対する組織/Ordering Node追加/削除、設定更新などを行う。	組織横断
	台帳運用管理	台帳のスナップショット作成、ページ、リストアなどを行う。	両方
	ノード運用管理	各組織が自身のノード(CA, Peer, Ordering Nodeなど)をデプロイ、管理する。	単一組織
	ID運用管理	各組織のIDやそれに紐づく秘密鍵/証明書を発行/管理する。	単一組織
ブロックチェーンシステム	アプリ(含む)運用管理	アプリ(APIサーバ含む)のデプロイや更新などを行う。加えて、システム全体としてみたときに必要となる運用作業も対象とする。	両方
	可用性/キャパシティ管理	システム全体として必要十分な可用性と性能を確保する。	両方
	インシデント/問題管理	システム内で複数組織に影響を及ぼす障害/異常の検知、情報収集、原因切分/分析などを行う。	両方
	リリース管理(構成/変更管理)	システムのデプロイプロセスを管理する。Continuous Integration/Deliveryパイプラインの整備やシステム全体としてのインフラバージョン整合/アップグレードなども含まれるべきである。	両方
コンソーシアム	コンソーシアム/ガバナンス運用管理	- ライフサイクル管理: メンバー管理や加入ワークフローなど。 - 意思決定/合意支援: コンソーシアム内での決めごとや設定値の調整支援する汎用的な手段。	組織横断
	課金/インセンティブ管理	システムを運営するための課金計算および決済などを行う。 (複数組織のリソースを相互利用する特性のため、リソースや費用の負担公平性確保に課題)	組織横断
	サービスレベル管理	システム全体としてのService Level ObjectiveやService Level Agreementを管理/監視する。	組織横断

6. 結 言

我々は、コンソーシアム型ブロックチェーンシステムにおける組織またぎのシステム運用を非中央集権的かつ効率的に実行するために OpsSC と呼ばれる手法を提案した。本手法に基づく Hyperledger Fabric 向けのブロックチェーンネットワーク運用ツールはオープンソースソフトウェアとしてコミュニティに公開している。今後もコンソーシアム型ブロックチェーンおよび Hyperledger Fabric の発展および本番利用の促進に貢献するべく、OpsSC の拡張や適用範囲拡大を進めていく。

7. 参考文献

- [1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2009.
- [2] V. Buterin, “A next-generation smart contract and decentralized application platform,” White paper, pp.1-36, 2014.
- [3] The Linux Foundation, <https://www.linuxfoundation.org/>
- [4] Hyperledger Foundation, <https://www.hyperledger.org/>
- [5] Hyperledger Fabric, <https://github.com/hyperledger/fabric>
- [6] E. Androulaki et al., “Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains,” EuroSys’18, pp. 1-15, 2018.
- [7] Accenture and DTCC, “Governing DLT networks,” https://www.accenture.com/_acnmedia/accenture/redesign-assets/dotcom/documents/global/2/accenture-governing-dlt-networks.pdf, 2019.
- [8] T. Hoiss et al., “Blockchain service operations: A structured approach to operate a blockchain

- solution,” IEEE DAPPS 2021, pp.11-19, 2021.
- [9] What’s new in Hyperledger Fabric v2.x,
<https://hyperledger-fabric.readthedocs.io/en/release-2.2/whatsnew.html#decentralized-governance-for-smart-contracts>
 - [10] T. Sato et al., “Toward fully-decentralized system with hyperledger fabric,” IEICE ComEX, Vol.12, No.5, p. 223-229, 2023.
 - [11] T. Sato et al., “Smart-Contract based System Operations for Permissioned Blockchain,” IFIP NTMS 2018, p. 6, 2018.
 - [12] T. Sato et al., “OpsSC: Decentralized Blockchain Network Operation Workflow for Hyperledger Fabric,” IEEE Blockchain 2021, pp. 287-294, 2021.
 - [13] T. Sato et al., “Operations Smart Contract to Realize Decentralized System Operations Workflow for Consortium Blockchain,” IEICE Transactions on Communications, vol. E105.B, Issue. 11, pp. 1318-1331, 2022.
 - [14] Operations Smart Contract (OpsSC) for Hyperledger Fabric,
<https://github.com/hyperledger-labs/fabric-opssc>
 - [15] Hyperledger Labs, <https://labs.hyperledger.org>
 - [16] Fabric Ecosystem, <https://wiki.hyperledger.org/display/fabric/Ecosystem>
 - [17] J. Sousa et al., “A Byzantine Fault-Tolerant Ordering Service for the Hyperledger Fabric Blockchain Platform,” IEEE/IFIP DSN 2018, pp. 51-58, 2018.
 - [18] A. Barger et al., “Byzantine Fault-Tolerant Consensus Library for Hyperledger Fabric,” IEEE ICBC 2021, pp. 1-9, 2021.
 - [19] Hyperledger Besu, <https://www.hyperledger.org/projects/besu>