情報処理学会論文誌　デジタルプラクティス　Vol.6　No.1　87–98 (Jan. 2025)

分散台帳を用いた複数組織による機密情報の共同管理および利活用システムの検討

池川 航史[1,a)]

受付日 2024年6月18日, 採録日 2024年10月9日

概要：信頼性ある自由なデータ流通が注目されており，高い耐改ざん性と非中央集権的なデータ管理が可能な分散台帳技術の活用が進んでいる．しかし，分散台帳に書き込まれた情報は削除が困難であるため，機密情報を直接書き込むことは避けるべきである．本稿では，プライバシー保護機能を備えた許可型分散台帳基盤 Hyperledger Fabric を用いた，複数組織による機密情報の共同管理および利活用を実現するシステムの要件定義を示した．また，提案システムは医療分野における機密情報共有および活用が期待され，その具体的なユースケースとシナリオ例およびその実装を示した．さらに評価を実施し，既存システムと比較して処理遅延があるものの，想定する医療分野におけるユースケースに耐えうる性能を確認した．

キーワード：分散台帳，ブロックチェーン，スマートコントラクト，TEE，システム運用管理

Study of Joint Management and Utilization System of Sensitive Data by Multiple Organizations Using Distributed Ledger

Koshi Ikegawa[1,a)]

Received: June 18, 2024, Accepted: October 9, 2024

Abstract: The importance of Data Free Flow with Trust is gaining attention, and the use of distributed ledger technology, which offers high tamper resistance and decentralized data management, is becoming more widespread. However, since information written into a distributed ledger is difficult to delete, it is advisable to avoid directly writing confidential information. This paper defines the requirements for a system that enables multiple organizations to collaboratively manage and utilize confidential information using Hyperledger Fabric, a permissioned distributed ledger platform with robust privacy protection features. The proposed system is expected to benefit the sharing and utilization of confidential information in the medical field, and specific use cases, scenarios, and implementations are provided. Furthermore, evaluation experiments confirm that despite some processing delays compared to existing systems, the proposed system possesses sufficient performance for the anticipated use cases in the healthcare sector.

Keywords: distributed ledger, blockchain, smart contract, TEE, system operations management

1. はじめに

1.1 信頼ある自由なデータ流通（DFFT）

信頼ある自由なデータ流通（Data Free Flow with Trust, DFFT）[1] は，2019 年 1 月の世界経済フォーラム年次総会および同年 6 月の G20 大阪サミットにて日本国が提唱した概念である．DFFT は，プライバシーやセキュリティ，知的財産権に関する信頼を確保しながら，ビジネスや社会課題の解決に有益なデータが国境を意識することなく自由に行き来する，国際的に自由なデータ流通の促進を目指している．日本国は，DFFT の実現に向けて 2021 年 9 月にデジタル庁を設立し，提唱国として責任を持ち推進していくと宣言している [2]．また経済産業省は，DFFT の具体化に向けて「データの越境移転に関する研究会」を立ち上げ，越境データ移転の実態調査や有識者会議を行い，データ流通プラットフォームの構築に関して議論している [3]．特に，DFFT の具体的な実現に向けて，データを分散管理することが議論されており，分散台帳技術の活用が見込まれる．分散台帳に一度書き込まれたデータは一般的に消すことが不可能とされており，特に機密情

[1] 株式会社日立製作所
Hitachi, Ltd., Kokubunji, Tokyo 185–8601, Japan
[a)] koshi.ikegawa.mf@hitachi.com

報を取り扱う際にはプライバシーの保護を最優先に考える必要がある．

1.2　分散台帳技術と高信頼実行環境

本研究では，複数組織による機密情報の共同管理および利活用に焦点を当て，許可型分散台帳の Hyperledger Fabric と，高信頼実行環境（Trusted Execution Environment, TEE）の一種である Intel SGX を用いてスマートコントラクトを実行する拡張機能である Fabric Private Chaincode（FPC）を用いたシステムを構築した．本節では，分散台帳技術と TEE に関する基礎概念を示し，許可型分散台帳技術の Fabric と Fabric のプライバシー保護機能を説明する．

1.2.1　分散台帳技術

ブロックチェーンを筆頭とした分散台帳技術は，破壊的イノベーションとして金融分野や産業分野などへの応用が期待され，注目を集めている．金融分野では，第三者機関を経由して実施されてきた従来の取引手法を，利用者間のピアツーピア（Peer-to-Peer, P2P）通信で直接取引を行う分散台帳技術に代替することで，時間的および金銭的な取引コストが削減できると期待されている．分散台帳技術の特徴に，取引やデータの内容が分散台帳ネットワークに参加しているすべての組織に共有され，各組織が保有する分散台帳に書き込まれる点がある．この特徴により，悪意を持つ攻撃者はデータを改ざんするために各組織が保有する分散台帳すべてを書き換える必要があるため，高い耐改ざん性を有するとされる．一方で，医療データや個人情報などの機密情報は，個人情報保護法や EU 一般データ保護規則（GDPR）[4] などの各国が定める法律に基づいて厳重に扱う必要がある．

また，分散台帳におけるネットワークの構成方法は主に 2 種類に分けられる．1 つ目は，暗号資産のように不特定多数の計算機資源が形成する自由参加型分散台帳であり，たとえば Bitcoin [5] や Ethereum [6] などがある．2 つ目は，特定の企業や団体などの組織のみで形成されており，参加にはすでにネットワークに参加している組織の一部もしくはすべての許可が必要となる許可型分散台帳であり，たとえば Fabric [7] や Quorum [8]，Corda [9] などがある．

本研究にて取り扱う機密情報を扱うようなケースにおいては，許可型分散台帳を使用することで，公開範囲をネットワーク参加者のみに制限することを前提としているが，ネットワークに参加するすべての組織が信頼できるとは保証できないため，機密情報本体を直接分散台帳に書き込むことは避ける必要があるという課題がある．高い耐改ざん性および非中央集権的なシステムの運用を実現する分散台帳技術の利点を活かしつつ，プライバシーは保護したいという需要に応えるために，本稿ではそれらを実現可能とする分散台帳基盤技術である Fabric の利用を検討する．

1.2.2　高信頼実行環境

本研究の提案システムにおいて採用する Intel SGX を具体例として TEE の概念を説明する．TEE は，コンピュータシステム内で機密データやプログラムを保護するための隔離された実行環境を提供する機能である．この環境では，機密データが他のシステムコンポーネントからアクセスされることを防ぎ，外部からの攻撃に対する耐性を提供する．特にセキュリティが重視される分野で利用されており，金融サービス，医療データの保護，デジタル証明書の管理など，様々な用途において機密性を確保するために使用される．TEE は主に CPU ベンダ各社が製造および販売しており，Intel 社の Software Guard Extensions（SGX）や ARM 社 の TrustZone，AMD 社 の Secure Encrypted Virtualization（SEV）などが存在する．

提案システムにおいて採用する Intel SGX は，アプリケーションが信頼できない OS やハイパーバイザの下でも機密データを保護できるようにするために設計された．McKeen ら [10] は，Intel SGX の設計と実装について詳細に説明しており，SGX がどのようにして隔離された実行環境を提供し，攻撃に対する耐性を強化するかを論じている．SGX は，Enclave Registration という仕組みにより主記憶装置上に暗号化された Enclave と呼ばれる保護領域を生成し，その領域にプログラムやデータを読み込むことによって機密情報を保護しつつプログラムを実行することを可能にする．Remote Attestation という仕組みを用いて，Intel 社のサーバと通信することで，セキュアな Enclave および TEE 環境が正しく設定され，信頼できる状態であることを第三者に証明するプロセスを有する．

1.3　Hyperleder Fabric

Fabric は許可型分散台帳であり，TEE を使用したスマートコントラクトの実行が可能となる拡張機能を持つ．本節にて，Fabric の基礎概念と，Fabric が持つプライバシー保護機能を説明する．

1.3.1　Fabric の基礎概念

Fabric は，Hyperledger Foundation により管理されているオープンソースソフトウェア（Open Source Software, OSS）の許可型分散台帳基盤である．

図 1 は，Fabric を用いて分散台帳ネットワークを立ち上げた際の簡略図である．組織は，分散台帳ネットワーク内の独立したエンティティであり，ネットワークの構成メンバーとして機能する．各組織は，それぞれが独立してメンバーシップサービスプロバイダ（Membership Service Provider, MSP）および認証局（Certificate Authority, CA）を持ち，認証と ID 管理を担当する．また，各組織が保有する Peer ノードは Ledger，State DB，および Chaincode の実行環境を含む．Ledger はネットワーク全体にわたって取引記録を分散して保持する台帳であり，分散

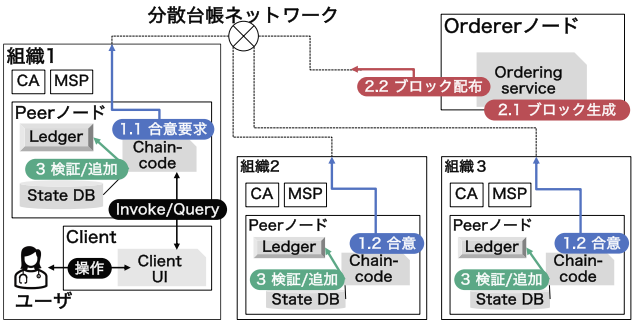

図 1　Fabric 基礎外観図：Fabric を用いた分散台帳ネットワークのトランザクション発行時における参加組織間の合意形成

Fig. 1　Basic overview of Fabric: consensus algorithm among participating organizations during transaction issuance in a distributed ledger network using Fabric.

台帳である．State DB は Ledger が管理しているトランザクションを実行した際の結果の最新情報を保存するデータベースである．Chaincode は Peer にインストールされる Fabric におけるスマートコントラクトであり，複数組織の Peer にインストールされた Chaincode の実行結果が一致することによって信頼性を確保している．Orderer はトランザクションの順番を決定し，Peer にブロックを配布する役割を担うノードである．Client はユーザが操作するインタフェースであり，ユーザは Client を介して Chaincode に記述された各関数を選択し呼び出しを行う．

また，Fabric では Execute-order-validate 構成と呼ばれる 3 段構成の合意形成アルゴリズムを用いてトランザクション書き込みおよび実行の合意を取得している．図 1 を用いて Fabric における合意形成の流れを示す．まず（1）Execute では，ユーザが Client を介して Chaincode を呼び出しトランザクションの実行を命令すると，分散台帳ネットワークに参加する他の組織に合意を要求し，他の組織の Peer が，Chaincode 上でトランザクションの検証および署名を実施することで合意する．次に（2）Order では，Orderer は署名を集めたトランザクションを受け取り，トランザクション同士の順序関係を解決して複数のトランザクションを梱包したブロックを生成し，そのブロックを各組織の Peer に配布する．最後に（3）Validate では，Orderer から渡されたブロックを各 Peer が検証し台帳に書き込む．ブロック内に書き込まれたトランザクションを実行し，State DB に結果を書き込む．

1.3.2　Fabric におけるプライバシー保護技術

Fabric には，データを秘匿しつつ共有するための標準機能や Fabric を拡張する形で実装された OSS が存在する．本節にて 3 種類のプライバシー保護技術を紹介し，提案システムの構成に使用する技術の選定指針を示す．

1.3.2.1　Channel

Channel 機能を用いることで同一の分散台帳ネットワークに参加する特定の組織間でのみ共有される台帳を作成することができる．その Channel に属していない他の組織からは台帳を見ることができないような仕組みを実現することが可能となる．Fabric の標準的な機能として実装されており，本機能を用いてトランザクションおよびデータの開示先を制限するような機能を持つシステムを簡単に構築することが可能となる．しかし，数多くの組織が参加する分散台帳ネットワークにおいて，2 つの組織間のみでのやり取りを秘匿するためには，膨大な数（具体的には，分散台帳ネットワークに参加する組織数が N の場合，N（$N-1$）個）の Channel を作成する必要があり，参加組織が増加することが前提の運用には向かない課題がある．

1.3.2.2　Private Data Collection

Private Data Collection（PDC）機能は，他の組織に対して秘匿したい機密情報を台帳外の外部記憶装置に保存する機能である．Benhamouda らの研究 [19] にて提案され，Fabric の標準機能としても実装されている．PDC 機能を用いて保管されているデータを他の組織に共有する際は分散台帳ネットワークを介さず P2P の直接通信で送受信を行う．しかし，機密情報を P2P 通信で他の組織に送信するため，他の組織に送られたデータがどのように活用されるか，提供者は分からないという課題がある．また，PDC 機能は数テラバイトオーダーのデータを取り扱うことは考慮されておらず，1 つのファイルが数ギガバイトオーダー以上の大容量のデータを扱うユースケースには適応が非現実的である．

1.3.2.3　Fabric Private Chaincode

Fabric に標準搭載されている Chaincode の実行環境においては，台帳に書き込まれるトランザクションおよびデータは平文の状態で保存されている．Chaincode を介さずとも分散台帳や State DB にユーザが直接アクセスすることでそれらの内容を確認することが可能となっているため，機密情報を取り扱うユースケースには適していない．Brandenburger らの研究 [20] にて提案された Fabric Private Chaincode（FPC）は，SGX を用いて Chaincode の実行を可能にする技術である．トランザクションおよびデータが Enclave 内で暗号化された後に台帳に書き込まれることにより秘匿化を実現している．FPC は Fabric の拡張機能として実装され，OSS として提供されている．

FPC では各 Peer に Chaincode をインストールした後に初期化処理として，その Chaincode 専用の Enclave が作成される．次に，Enclave 固有の暗号鍵が生成され各 Peer が保有する Enclave Registry Chaincode（ERCC）に登録される．Chaincode を呼び出し，分散台帳および State DB に書き込まれるトランザクションおよびデータは ERCC に登録された暗号鍵を用いて暗号化される．よって，悪意を持った攻撃者およびユーザさえも，それらに直接アクセスしてもトランザクションおよびデータを取り出すことはできない．

一方で，FPC の本稿執筆時の最新バージョンである v1.0-RC3 では，他の Peer の Enclave に対する暗号鍵の配布機能は非対応となっており，暗号化を実行した Peer 上でのみデータの復号化および Chaincode の実行が行われ，単一の Peer でのみ合意形成アルゴリズムが実行されるものとなっている本来であれば他の Peer の Enclave に対して暗号鍵を配布し，すべての Peer でデータの復号化および Chaincode が実行でき合意形成ができるような仕組みで動作する必要があることに注意する必要がある．本稿では，近い将来 OSS である FPC が成熟し，暗号鍵の配布機能や複数組織の Peer による合意形成が実行可能となることを期待し，本技術を選定することとした．

2. 関連研究

分散台帳におけるプライバシー保護を秘密計算技術によって実現する研究および技術が数多く存在する．秘密計算手法は主にゼロ知識証明（ZKP），マルチパーティ計算（MPC），および TEE に大別される．本節では，**表 1** に関連研究をまとめ，提案システムと比較を行う．はじめに，ZKP および MPC によるプライバシー保護に関する研究を紹介し，TEE を用いる場合との比較を示す．次に，TEE によるプライバシー保護に関する研究を示し，本研究の位置づけを示す．

2.1 ZKP および MPC によるプライバシー保護

ZKP は，相手に情報の内容を明かすことなく，その情報が正しいことを暗号学的に証明する技術である．文献 [11], [12] は，ZKP を用いて情報を暗号化した状態で分散台帳に保存し，プライバシー保護を実現している．一方で，ZKP は計算コストが非常に高く，特に複雑な証明では効率が低下する可能性を有する．

MPC は，複数の参加者が個々のデータを秘密にした状態で共同で計算を行い，その結果を得る技術である．文献 [13], [14] は，MPC と分散台帳技術を組み合わせたプライバシー保護を実現している．一方で，ZKP 同様計算コストが高く，更に複数の参加者が協力して計算を実行するため通信のコストが高いことから，複雑な計算では効率が低下する可能性を有する．

加えて，ZKP および MPC ともに実装には TEE と比較して暗号技術に対する高度な専門知識必要となることが課題となっている．様々なユースケースに対応することが可能な基盤システムにおいて，技術者による実装の容易さは重要な焦点となる．本研究の提案システムは TEE を採用しており，分散台帳に精通している技術者であれば追加で専門的な知識は不要となる点が利点としてあげられる．一方で，TEE を用いる手法にもハードウェア依存により計算機の準備が困難である課題が存在する．本研究では Azure を用いたパブリッククラウド VM による SGX ハードウェアによる実装手法の提示していることから，その課題を解決している．

2.2 TEE によるプライバシー保護

TEE を用いて分散台帳に書き込まれるデータを秘匿する手法にもいくつか種類がある．LayerX 社が開発した Anonify [15] は TEE を用いてオフチェーン領域におけるプログラムを正しく実行できることを保証している．一方で，本研究の提案システムに用いる FPC はオンチェーン領域におけるチェーンコードの実行に TEE を用いている点に違いがある．Cheng らの研究 [16] では，Ethereum と TEE を組み合わせ，プライバシー保護を実現するとともに高速な処理かつ低遅延を実現している．一方で，本研究の提案システムは，医療分野における研究組織から構成さ

表 1　分散台帳におけるプライバシー保護に関する研究と提案システム
Table 1　Related work on privacy protection in distributed ledgers and the proposed system.

文献	台帳	保護技術	説明
[11] Kosba et al.	Ethereum	ZKP	分散台帳上の金融取引を平文で保存しない分散型スマートコントラクトシステム．
[12] Kang et al.	Fabric	ZKP	Fabric の拡張機能として実装され，取引データを暗号化し，検証可能な ZKP を構築．
[13] Benhamouda et al.	Fabric	MPC	Fabric の Chaincode 上で動作する MPC を実装し，機密データの取引をサポート．
[14] Zhou et al.	Fabric	MPC	加法ホモモルフィック暗号を用いた MPC で，大規模ネットワークでの計算効率向上．
[15] LayerX	Quorum	TEE	オフチェーン領域におけるプログラムが正しく実行されることを保証．
[16] Cheng et al.	Ethereum	TEE	Ethereum と TEE を組み合わせ，プライバシー保護を実現するとともに高速な処理を実現．
[17] Desai et al.	Fabric	TEE	競売時のスマートコントラクトにおけるセキュリティ上の課題を解決，入札情報の秘匿と透明性を両立．
[18] Wu et al.	Fabric	TEE	複数のサイトで収集された臨床試験データを管理，共有，および分析する方法を提示．ローカルマシン上による SGX ハードウェア実行．パブリッククラウドベースの実装は AWS を用いた SGX シミュレーションを使用して検証．
提案システム	**Fabric**	**TEE**	**全ゲノム解析ユースケースに特化したデータ共同管理および利活用システム．Azure を用いたパブリッククラウド VM による SGX ハードウェアによる実装手法の提示．パブリッククラウド VM 上の実装による実行環境条件を揃えた評価実験．**

れるコンソーシアムによる機密情報の共同管理を推進するために許可型分散台帳 Fabric を使用する．

分散台帳と TEE を組み合わせる研究の中でも，提案システムと同様に Fabric と FPC を利用する研究も存在する．Desai らの SECAUCTEE [17] は分散台帳ネットワークを用いて競売を実現する際に使用するスマートコントラクトのセキュリティ上の課題を解決するため，FPC を使用することを提案している．これにより，入札の情報を秘匿し談合などの不正な入札を防止することと，競売の透明性を維持することを両立している．Wu らの研究 [18] は，FPC を用いて複数のサイトで収集された臨床試験データを管理，共有，および分析する方法を提示している．また，Amazon Web Service（AWS）クラウドとローカルマシンの 2 種類の環境を使用した実装を提案しているが，AWS クラウドには SGX を搭載した環境を作成できないため SGX シミュレーションを使用した実装となっている．

本稿の提案システムは，医療分野における「がんの全ゲノム解析」のユースケース実現に特化したシステム要件を定義し，実装および評価を行っている．また，システムの実運用を見据えたハードウェアの選定を実施し，SGX が利用可能でありパブリッククラウドとして提供されている Microsoft Azure で作成した VM 上にシステムを構築した．比較対象となるすべての分散台帳環境を構築し，条件を正しく揃えた実験を設計し評価を行った．

3. 本研究の貢献

本研究は，分散台帳技術を用いて複数の組織で医療データの 1 つであるゲノムデータを管理および利活用を目的としている．既報 [21] では機密情報を各組織が保有する外部記憶装置に格納し，そのデータのメタデータのみを分散台帳に格納することにより，機密情報の複数組織管理を実現していた．しかし，この管理および利活用基盤では，データの利活用に関係しない組織が解析依頼や結果情報の閲覧が可能となる．ゲノム情報を扱う研究者にとって，研究対象としているゲノムデータに関する情報は秘匿したいという課題がある．

その課題に対して，研究速報として既報 [22] にて FPC を用いたトランザクションおよびデータを秘匿しつつデータを複数組織管理および利活用の促進を実現するシステムのプロトタイプ構想が示された．本稿では，既報 [22] の内容を拡張し，適応ユースケースの具体化，実運用に向けたシステム要件の定義，および提案システムの実機評価を行った．

本研究の貢献を以下に示す．

- 全ゲノム解析ユースケースの具体的なシナリオ提示
- ユースケース実運用に向けたシステム要件定義
- 適切なハードウェア・ソフトウェア選定および実装
- 実機評価による性能フィージビリティ検証

4. ユースケースの前提とシナリオ

本章にて，医療分野における「がんの全ゲノム解析」をユースケースとした分散台帳を用いた複数組織によるがんゲノムデータの共同管理および利活用システムの要件を定義する．はじめに，ユースケースの前提となる全ゲノム解析についての説明と分散台帳を用いた共同管理を実現するための条件を示す．次に，本ユースケースにおけるユーザおよびシステムの動きをシナリオとして説明する．最後に，ユースケース実装の条件を踏まえたシナリオを実現するためのシステム要件をハードウェア構成とソフトウェア構成に分けて説明する．

4.1 ユースケースの前提

医療分野ではがんの研究において，発がん要因等を発見するために全ゲノム解析が行われている [23], [24]．全ゲノム解析は，リファレンスとなるゲノムデータとがんを発症した患者から提供されたゲノムデータを用いて，がんによる DNA 情報の変異を検出する処理である．がんの全ゲノム解析を進めるにあたり，ワークフローは大きく 5 つで構成され，(1) 患者への説明と同意取得，(2) ゲノム読み取り，(3) 全ゲノム解析，(4) 解析結果の分析，(5) 担当医による患者への薬剤という流れで解析処理が進められる．

4.2 ユースケースシナリオ

本研究は，これまで 1 つの組織内で閉じた形で管理されていたゲノムデータを分散台帳上で管理し，複数の組織間でのゲノムデータの利活用を促進することを目的としたユースケースの実現を目指す．複数組織での非中央集権的なシステム運用に加えて，改ざん不可能な堅牢なデータの利用証跡を管理することも分散台帳利用のモチベーションとなる．医療関連の組織，たとえば病院や大学，医療研究機関が分散台帳ネットワークに参加し，他の組織が保有するゲノムデータを利活用するための仕組みを実現するシナリオを示す．

まず前提として，ゲノムデータの特性を考慮する．ゲノムデータは，1 つのファイルで数百ギガバイトから数テラバイトに及ぶ膨大なデータ量を持ち，また個人情報が含まれた機密性の高いデータである．よって，生のデータを他の組織に直接渡すことはデータ転送にかかる時間の観点やプライバシー保護の観点からも避けるべきである．そこで，ゲノムデータに関するデータのハッシュ値や患者の国籍，性別，年齢，発症したがんの種類といったメタデータを管理する「ゲノムデータカタログ」を分散台帳上で運用する．データを活用したい他の組織のユーザは，データを保有する組織に許諾を得た後に解析処理を依頼し，結果のみを返すような運用を実現する．

本ユースケースの具体的なシナリオを示す．図 2 に本ユースケースのシナリオを実現する提案システムの概観を示す．本シナリオでは，組織 1 にデータを利活用したいユーザが所属しており，組織 2 がそのデータを保有していることとする．まず，組織 2 に所属するユーザががん患者から同意を得て，ゲノムデータを自組織のストレージに保存する．同時に，このデータに対応するハッシュ値および患者に関するメタデータを，分散台帳上のゲノムデータカタログに記録する．次に，組織 1 に所属するユーザがこのカタログを照会し，研究を推進するための適切なメタデータ条件を指定してデータの検索をする．必要なデータを発見した組織 1 のユーザは，組織 2 が保有するそのゲノムデータの利活用を希望し，分散台帳で管理されるワークフローシステムを通じて利用申請を行う．

次に，組織 2 のユーザは，組織 1 からの利用申請を確認し，承認した場合はその旨をワークフローシステムに記録する．組織 1 のユーザは，自身の申請が承認されたことを確認した後，分散台帳で管理される全ゲノム解析タスク管理システムに，解析条件を含むタスク情報を記録する．組織 2 の全ゲノム解析管理システムは，組織 1 から依頼されたタスク情報を取得し，ワークフローシステム上で利用許諾が得られていることを確認した後に解析処理を実行する．最終的に，解析結果は組織 1 のユーザに提供され，一連のプロセスが完了する．

このように，研究者が分散台帳上のゲノムデータカタログを照会し，必要なデータを効率的に検索し，適切な解析を依頼し，結果のみを受け取るというプロセスを安全かつ確実に実現することを目指す．また，全ゲノム解析は，使用するアルゴリズムや比較対象となるリファレンスゲノムデータなど解析条件を詳細に設定する必要があり，これらの条件設定はがんゲノム研究を推進するうえでの重要な要素となっている．研究者からはこの全ゲノム解析に関する解析情報はなるべく秘匿したいという要望があり，分散台帳上で管理されるメタデータやワークフロー，解析依頼の発行および結果の共有に関する情報は，関係組織以外に閲覧されないような仕組みが必要となる．

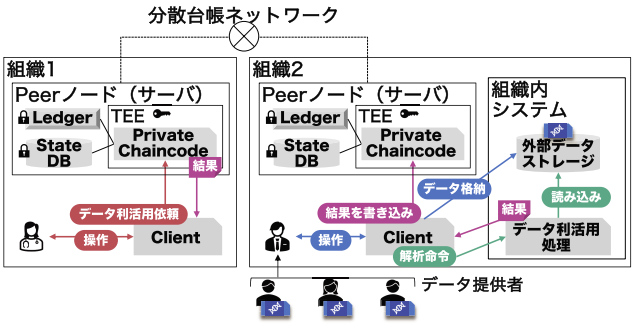

図 2　提案システム概観：分散台帳を用いた複数組織による機密情報の共同管理および利活用基盤システム

Fig. 2　Overview of the proposed system: joint management and utilization system of sensitive data by multiple organizations using distributed ledger.

5.　提案システム要件

本章にて，ユースケースとシナリオから必要となるシステム要件を定義する．提案システムは主に分散台帳部，各組織がそれぞれ保有するクライアント部，ゲノムデータの保存および全ゲノム解析を実行するデータ保存解析部に分け，それぞれに対する要件を示す．

はじめに，本ユースケースにおいては複数組織が同一権限で参加する非中央集権的運用が可能なシステムが必要とされており，改ざん不可能かつ堅牢なデータ利用証跡管理を目指すことからも，許可型分散台帳の利用を要件とする．また，ゲノムデータを利活用する際の組織間の取引は当事者間のみが閲覧できるようしたいというニーズがあり，プライバシー保護機能を備えた許可型分散台帳基盤を使用する必要がある．2 章に述べたとおり，分散台帳におけるプライバシー保護技術として秘密計算技術と組み合わせる手法がいくつか存在するが，容易な Chaincode 実装方式および今後の拡張性を考慮し，TEE による手法を要件とした．一方で，TEE による手法は専用のハードウェアを準備する必要があるという課題を抱えており，なるべく準備コストを書けないようなハードウェア選定を必要としたい．そこで，TEE を利用可能なパブリッククラウドサービス上で提案システムの分散台帳部を構築することを要件とした．

次に，分散台帳およびゲノム解析を命令するためのユーザが操作するクライアント端末が必要となる．ユーザが分散台帳部への Chaincode 関数呼び出しを実行するための操作を可能とするユーザインタフェースを含むクライアントソフトウェアの実装を要件とする．

続いて，ゲノムデータは 1 つのファイルで数百ギガバイトから数テラバイトに及ぶデータサイズであることから，膨大な容量のストレージを必要とする．また，全ゲノム解析を実行するための計算機を必要とする．全ゲノム解析に必要となる一連のソフトウェアを含むものとする．なお，既報 [21], [22] において全ゲノム解析に必要となるストレージ，計算機，およびソフトウェアに関する環境は構築済であり，本稿では既存環境を流用することとした．

以下に，提案システムの要件をまとめ列挙する．

- 分散台帳部
  - 許可型分散台帳によるネットワーク構築
  - TEE によるプライバシー保護
  - パブリッククラウドサービス上の構築
  - シナリオ実現のための Chaincode
- クライアント部
  - ユーザが操作するクライアント端末
  - UI を含むクライアントソフトウェア

- データ保存解析部
  - ゲノムデータ保管のための大容量ストレージ
  - 全ゲノム解析のための計算リソース
  - 全ゲノム解析を実行する一連の解析ソフトウェア

6. 提案システム実装

本章にて提案システムの要件定義に沿った具体的な実装を示す．

6.1 ハードウェア構成

図 3 に，提案システムのハードウェア構成を示す．分散台帳部，クライアント部，およびデータ保存解析部に分けそれぞれの実装について示す．

6.1.1 分散台帳部

分散台帳部は許可型分散台帳によるネットワーク構築，TEE によるプライバシー保護，およびパブリッククラウドサービス上の構築を要件としている．Fabric は許可型分散台帳であり，中でも TEE を活用した拡張機能である FPC を有している．FPC を動作させるにあたり，ハードウェアとして Intel 製の CPU の中でも SGX が搭載されたものを選定する必要がある．さらに，マザーボードが SGX 対応しており，さらに BIOS の設定で SGX が実行できるよう設定する必要がある．Intel SGX が搭載されたハードウェアを提供するクラウドサービスは数少なく，Amazon，Google，および Microsoft の世界的有力クラウドベンダ 3 社の中では Microsoft の Azure のみで利用可能である．よって，本研究では Azure を採用し，Azure 上で SGX が使用可能となる DCsv2 シリーズの VM 上で Fabric および FPC を動作させることとした．

ユースケースの実現にあたり，メタデータを書き込むゲノムデータカタログ機能，他の組織が保有するゲノムデータの利活用申請および承認を実施するワークフロー機能，ゲノムデータ解析依頼の発行および結果の共有機能を持つ

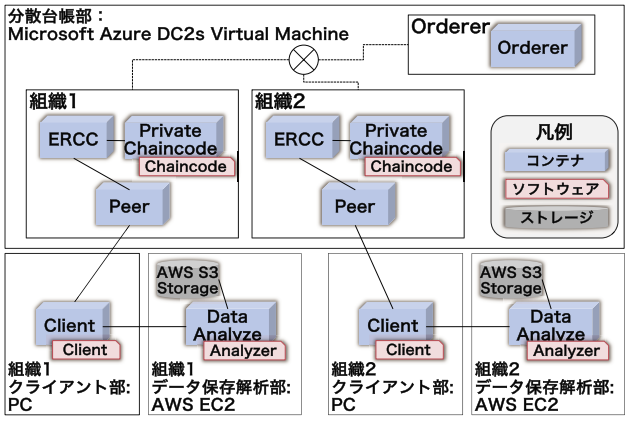

図 3　提案システム構成：ハードウェアおよびソフトウェア配置

Fig. 3　Proposed system configuration: hardware and software arrangement.

Chaincode が必要となる．提案システムではそれぞれ機能が独立した Chaincode プログラムを作成し，各組織の Peer にそれぞれインストールすることとした．Fabric は各機能を持つモジュールが docker を用いたコンテナ化がされており，本システムも docker を用いたコンテナベースのシステムで動作する．docker を用いて 2 組織から構成される Fabric の分散台帳ネットワークを構築した．この分散台帳ネットワークを構成する各組織は Peer コンテナを 1 つ持つ．各 Peer は分散台帳本体とその分散台帳で管理されているトランザクションを実行した結果となる最新のデータを管理する State DB を所有する．また，この分散台帳ネットワークは 1 つの Orderer を持つ．各組織が保有する Peer はそれぞれ Enclave Registry Chaincode（ERCC）コンテナを介して Private Chaincode コンテナに接続される．ERCC は，TEE の外部で実行される Chaincode であり，Private Chaincode と関連する TEE の Enclave に関する情報や公開鍵情報などを管理する．Private Chaincode コンテナには 6.2.1 節にて述べる Chaincode がインストールされる．

6.1.2 クライアント部

次に，各組織のユーザが直接操作するクライアント部は，各ユーザが保有する PC を利用することを想定する．Client ソフトウェアが動作するコンテナが配置され，ユーザはこのソフトウェアを使用することで Peer コンテナに対する分散台帳に関する操作およびデータ保存解析部のデータ解析ソフトの操作が可能となる．

6.1.3 データ保存解析部

続いて，各組織が保有するデータ保存解析部は，ゲノムデータを保存しておく膨大な容量のストレージおよび解析処理を実行するための専用の計算機を必要とする．データ保存解析部は，既報 [21], [22] にてがんゲノム解析環境が AWS 上に構築されている，ストレージには AWS S3，解析プログラムを配置する VM には AWS EC2 を使用することとした．データ解析を行う Analyzer ソフトウェアを含む Data Analyze コンテナは Client と接続されユーザからの操作を受け付ける．

6.2 ソフトウェア構成

提案システムにおけるソフトウェアは，ハードウェア構成における分散台帳部にて使用する Chaincode ソフトウェア，クライアント部にて使用する Client ソフトウェア，およびデータ保存解析部にて使用する Analyzer プログラムの 3 つを設計および実装した．本章において，各ソフトウェア事に設計および実装を示す．

6.2.1 Chaincode

提案システムの Chaincode はシナリオ例に沿って実装された．シナリオ例の実現には，(1) データカタログ管理を行う機能，(2) データの権限管理を行う機能，および

(3) タスク管理を行う機能が必要となる．それぞれの機能を実現するため各関数の詳細を**表 2** に示す．各関数は台帳への書き込みおよび State DB の更新を伴う関数（Invoke）と State DB の情報を読み出しのみを行う関数（Query）に分けられる．

6.2.2 Client

Client はユーザが提案システムを操作するために用意されたコマンドラインベースの CUI ソフトウェアである．本ソフトウェアは，各組織にそれぞれ独立して配置される．患者から提供されたゲノムデータをストレージに保存する機能や，分散台帳ネットワークの Peer を介して Chaincode 内の関数を呼び出すための機能，Analyzer に対して解析処理を依頼する機能など，ユーザがシステムに対して操作を必要とする場合は Client を介して実行する．

6.2.3 Analyzer

提案システムの Analyzer は Client を介して自身が保有するデータに対して利活用依頼が来ているかを定期的に確認し，依頼が来ていた場合そのタスクを実行するためのアプリケーションである．Analyzer に接続された Storage から解析を依頼されたデータを読み出し，タスクを実行する．本稿では，医療分野におけるゲノムデータの管理および利活用を例にしたシナリオを想定しており，OSS である Burrows-Wheeler Aligner（BWA）[25], Sumtools [26], IGV.js [27] を使用することで全ゲノム解析に必要となる一連の処理を実行することとした．

6.3 シナリオに沿ったソフトウェア操作

表 3 にトランザクションの最新データを管理する State DB の内容を示す．

6.3.1 データカタログの操作

はじめに，組織 2 の医師は患者から提供を受けたゲノムデータを外部記憶装置に格納する．同時に，組織 2 の医師は Client を操作して Chaincode の（i）SetData 関数を呼び出し，ゲノムのメタデータを分散台帳に書き込む（表 3(a)）．次に，組織 1 の医師は Client を操作して Chaincode の（ii）GetData 関数を呼び出し，ゲノムのメタデータを参照し，活用したいデータを検索する．活用したいデータを見つけた場合，（b）の処理に移る．

6.3.2 データ利活用権限の申請および承認

組織 1 の医師は Client を操作して Chaincode の（iii）RequestPermission 関数を呼び出しデータ所有者である組織 2 にデータ利用権限を求める申請を書き込む（表 3(b)(1)）．次に，組織 2 の医師は Client を操作して，自身が所有するゲノムデータに関する利活用権限の申請を確認する．組織 2 の医師は Client を操作して Chaincode の（iv）ApprovePermission 関数を呼び出して組織 1 の医師に対して申請を承認することでデータ利活用権限を付与する（表 3(b)(2)）．組織 1 の医師はデータ権限の申請が承認されたことを確認し，（c）の処理に移る．

6.3.3 タスクの実行依頼および結果の書き込み

組織 1 の医師は Client を操作して Chaincode の（v）

表 3　State DB に書き込まれたデータ
Table 3　State DB Data.

(a) データカタログ管理 Chaincode が管理する State DB

ID	データ所有者 (1)	ハッシュ値 (2)	その他 (3)
Data-01	組織 2	00aa11bb...	Data info

(b) データ権限管理 Chaincode が管理する State DB

ID	権限依頼者 (1)	権限所有者 (2)
Data-01	組織 1 医師 A MSPID	組織 1 医師 A MSPID

(c) タスク管理 Chaincode が管理する State DB

ID	依頼者 (1)	タスク (2)	結果 (3)
Data-01	組織 1 医師 A MSPID	解析条件	結果

表 2　Chaincode の関数リスト
Table 2　Chaincode functions list.

関数名 (タイプ)	説明
(i) **SetData** (Invoke)	データ ID，データ名，データのハッシュ値，およびデータに関するその他情報を入力とする．データカタログにデータを追加する関数．同時にユーザの MSPID および呼び出し元 Client に割り当てられている Client ID を取得しデータ所有者の情報として書き込みを行う．
(ii) **GetData** (Query)	データ ID を入力として，データカタログに登録されているそのデータ ID のデータの情報をユーザに返す．
(iii) **RequestPermission** (Invoke)	データ ID を入力として，利活用を行いたいデータの利用権限の要求を行う．
(iv) **ApprovePermission** (Invoke)	データ ID を入力として，データの所有者が要求者に利用権限を付与する．
(v) **RequestTask** (Invoke)	データ ID および依頼するタスクの内容を入力とする．依頼者が該当データの利用権限を持つか確認し，そのタスクの内容を書き込む．
(vi) **GetTask** (Query)	データ ID を入力として，該当データの所有者が自身であり，該当データに関する利活用タスクが書き込まれている場合，その該当データのタスクの内容をユーザに返す．
(vii) **WriteTaskResult** (Invoke)	データ ID およびタスクの実行結果を入力として，該当データの所有者が自身であり該当データに関する利活用タスクが書き込まれている場合，そのタスクの実行結果を書き込む．

RequestTask 関数を呼び出して組織 2 にゲノムデータの解析処理を依頼する（表 3(c)(2)）．次に，組織 2 のゲノム解析処理装置が Client を操作して Chaincode の（vi）GetTask 関数を呼び出して，組織 1 の医師から依頼された解析処理の内容を読み込む．ゲノム解析処理装置は，指定されたゲノムデータを外部記憶装置から読み込み，解析処理を実行する．最後に，ゲノム解析処理装置は Client を操作して Chaincode の（vii）WriteTaskResult 関数を呼び出して，解析処理の結果を分散台帳に書き込む（表 3(c)(3)）．

7. 評価

7.1 提案システムの要件に対する実装評価

分散台帳部は Azure による SGX が動作する VM 上に Fabric ネットワークを構築することで，要件を満たした構成を実現した．また，全ゲノム解析ユースケースに必要となる Chaincode も実現した．Chaincode は全ゲノム解析ユースケースに必要となるゲノムデータカタログ，ワークフロー管理，タスク管理それぞれの機能を要件としており，実装後の実機検証により必要な機能要件を満たしていることを確認した．

クライアント部に関しては，本研究では各ユーザが保有する PC をハードウェア要件とし，Client ソフトウェアをインストールする構成とした．ユーザは CUI を操作し Client ソフトウェアにより Chaincode の関数呼び出しを実現した．

データ保存解析部に関しては，既報 [21], [22] において全ゲノム解析に必要となるストレージ，計算機，およびソフトウェアに関する環境は構築済であり，本稿では既存環境を流用することとしたため，既に要件を満たしている．本稿にて実装した分散台帳部およびクライアント部とは正しく接続し，一連の全ゲノム解析処理を実行できることを確認した．

7.2 性能フィージビリティ評価

7.2.1 実験設計

本研究では提案システムのシナリオ例の正常系の実行時に要する実行時間を計測し比較を行った．提案システムのシナリオ例の実行時間を計測することにより，本システムの実用性を示すことを目的とする．従来の実装手法の実行時間と比較することによって，提案システムのボトルネック箇所を解析し，提案手法のさらなる改善を目指すことを目的とする．また，本稿のシナリオとして例示しているゲノム解析のユースケースにおいては，ゲノムデータ解析処理に膨大な時間がかかることは自明であるため，評価実験を実施する際は解析処理をスキップし，解析結果はダミーデータを書き込むこととした．

実験には 6.1 節にて述べたハードウェア構成を使用した．評価実験には**表 4** に示す 3 つの環境を用意した．（1）と（2）を比較することによって，SGX ハードウェアを使用することによる遅延時間を測定する．また，（1）と（3）を比較することによって，SGX のハードウェアおよび暗号化や鍵管理，証明書管理等の SGX 全体の処理による遅延時間を測定する．Fabric 標準として動作する Chaincode の呼び出しには，Fabric 標準として提供されている peer コマンドを利用し，実行時間の測定を行った．

評価実験は 6.3 節にて述べたシナリオ例の正常系を実行した際の実行時間をそれぞれの環境で 3 回計測する．また，シナリオ例の正常系を進めるにあたり実行される各関数の実行時間も同時に計測した．

7.2.2 結果および考察

表 5 に実験結果を示す．表 5a は，シナリオ例の正常系を実行した際の実行時間をそれぞれの環境で 3 回計測した結果である．環境（1）では 10.649 秒，（2）は 10.628 秒と差はなかった．環境（3）の場合，0.235 秒と（1）と比較すると提案システムの実行時間は約 45 倍かかった．次に，表 5b は，各関数を Invoke と Query に分類し，平均実行時間を求めた結果である．Invoke に関しては，環境（1）の結果は 2.111 秒，（2）の結果は 2.108 秒と差はなかった．環境（3）の結果は 0.034 秒と（1）と比較すると

表 4　実験環境
Table 4　Experimental environment.

環境 (1)	SGX にて動作する提案システム．
環境 (2)	SGX を仮想的に動作させるシミュレーションモードを用いて動作する提案システム．
環境 (3)	既存システム：提案システムの Chaincode と同等の機能を Fabric に標準搭載されている Chaincode 仕様にて実装．

表 5　評価実験の結果
Table 5　Experimental results.

a) 各関数ごとの平均実行時間 [単位：秒]

	環境 (1)	環境 (2)	環境 (3)
1. SetData	2.116	2.106	0.027
2. GetData	0.043	0.043	0.032
3. RequestPermission	2.112	2.103	0.037
4. ApprovePermission	2.110	2.109	0.035
5. RequestTask	2.117	2.118	0.035
6. GetTask	0.049	0.045	0.032
7. WriteTaskResult	2.102	2.102	0.036
合計	10.649	10.628	0.235

b) 関数の分類ごとの平均実行時間 [単位：秒]

	環境 (1)	環境 (2)	環境 (3)
Invoke (1,3,4,5,7)	2.111	2.108	0.034
Query (2,6)	0.046	0.044	0.032

提案システムの実行時間は約 62 倍かかった．Query に関しては，環境（1）の結果は 0.043 秒，（2）の結果は 0.043 秒，（3）の結果は 0.032 秒と，（3）が若干高速であったものの大きな差は生じなかった．

表 5a における，環境（1）と（2）の測定結果より，SGX ハードウェアの使用自体には遅延がないことが示唆された．また，環境（2）と（3）の測定結果より，Enclave Registration や Remote Attestation など SGX を実行する際に必要となる処理に関してシミュレーションであっても処理遅延が発生することが示唆された．特に Remote Attestation は Intel 社が所有するサーバと通信を行うことで信頼できる状態であることを第三者に証明する仕組みであるため処理に時間がかかると考察する．また，表 5b の結果より，特に動作遅延となっている原因は分散台帳および State DB の更新を伴う Invoke の関数であることが分かる．Query の関数に関しては，Enclave にてデータを復号化するのみの処理であるため，事前にローカル登録された証明書や鍵を用いているため外部のサーバとの通信は不要であることから，大きな遅延が生じないと考察する．

以上より，提案システムにおいては一部の処理で従来システムと比較すると若干の遅延は生じた．一方で，提案システムはゲノムデータを活用した処理であり，解析処理自体に数時間以上の処理時間を要することから，10 秒程度の遅延はシステムの運用に影響を与えない．

7.3　機密データの安全性に関する評価

データの利活用を行った組織を組織 1，データを所有する組織を組織 2，一連の処理とは関係がない組織を組織 3 とし，機密データの安全性に関する評価を以下に示す．

はじめに，提案システムにおける Ledger および State DB を直接参照するケースを評価する．提案システムにおける Ledger および State DB に書き込まれるデータは TEE 上で動作する Chaincode により暗号化されて保存されている．よって，Chaincode を介さない直接の Ledger および State DB に対するデータアクセスを行ったとしても組織 1,2,3 すべてのユーザにおいて内容を確認することはできない．このことは，実際に提案システムを実機検証し，一通りユースケースシナリオを実行した後に Ledger および State DB に直接参照することで確認できた．

次に，提案システムにおける Ledger および State DB に対して Chaincode を介して参照するケースを評価する．Chaincode 関数における（vi）GetTask 関数の挙動に焦点を当てて評価をする．GetTask 関数は自身が保有するデータ ID に対して解析タスクの依頼が行われていた場合，そのタスクを読み出す機能である．よって，自身が保有するデータ ID，もしくは自身がタスクを依頼したデータ ID 以外を引数に関数を呼び出しを実行した場合はエラーとして処理される．提案システムを実機検証し，組織 1 が（v）RequestTask 関数により組織 2 が保有するデータに対する解析処理を依頼した．次に，データを所有する組織 2 がそのデータ ID を引数に（vi）GetTask 関数を実行したとき，正常処理として正しくデータが渡されることを確認した．また，一連の処理とは関係がない組織 3 が同様の（vi）GetTask 関数を実行した際はエラーとして処理されることを確認した．

一方で，本提案システムおよびユースケースにおいては，組織に属するユーザが操作するクライアント PC のユーザ権限が奪われ，そのユーザになりすまして Chaincode 関数呼び出しをするようなケースの安全性は確保できない．クライアントソフトウェアにアクセスするパスワードや秘密鍵は，提案システムの保護機能とは別に厳重に管理する必要がある．

8.　まとめ

8.1　得られた知見

本研究では，プライバシー保護機能を備えた分散台帳技術である Fabric を用いて，複数組織が参加するネットワークを構築し，機密情報を共同管理および利活用するシステムを検討した．提案システムは，医療分野の機密情報共有および利活用への実用化が期待されており，ゲノムデータを取り扱うユースケースおよびそのシナリオを具体的な事例として示した．

また，ユースケースシナリオの実運用に向けたシステム要件定義を行った．ゲノムデータを取り扱うユースケースにおいて，研究者は研究対象としているゲノムデータに関する情報は秘匿したいというニーズがある．分散台帳に書き込まれるデータを秘匿する手法として ZKP や MPC，TEE などの秘密計算技術を用いる研究が数多く存在するが，実装時に比較的暗号学分野の専門知識が不要である TEE を採用することとした．加えて，ハードウェアに依存するため計算機を準備するコストがかかるという TEE の課題に対しても，パブリッククラウドである Azure を使うことで解決した．そこで，複数のプライバシー保護機能を備えた Fabric を使用することとし，その中でも FPC を使用することが適していることを明らかにした．

さらに，提案システムの要件に対する実装評価，提案システムと既存システムの実機評価による性能フィージビリティ検証，機密データの安全性に関する評価を実施した．提案システムは既存システムより処理遅延は存在するものの，想定するヘルスケア分野のユースケースにおいては十分耐えうる性能を持つことを明らかにした．

8.2　今後の課題

今後は，提案システムの実証実験や実運用を進め，社会実装した際の運用した際の課題を発見し解決していく．また，FPC の開発コミュニティにおいては，現在 Fabric 標

準の Chaincode と同様に Go 言語で書かれた Chaincode を FPC に使用可能にする開発プロジェクトが進んでいる．Go 言語は Fabric における Chaincode 開発のためのプログラミング言語としてデファクトとなっており，Fabric Chaincode エンジニアは Go 言語でコーディングできることを好む傾向にある．今後は，提案システムを Go 言語で実装し，追加の評価を行う．

また，SGX は Intel 社を信頼することを前提とした設計となっており，非中央集権的なシステム構成および運用を目指す分散台帳の基本的な理念と相反する点が課題としてあげられる．近年では Intel 社だけでなく AMD 社や arm 社が設計する CPU アーキテクチャも普及しはじめており，各社それぞれ別の手法で TEE を実装している．それらが相互運用可能にすることで，TEE の観点において非中央集権的なシステム構築を可能にすると予想する．今後は，各 CPU ベンダの TEE を利用可能とする FPC 実装を OSS コミュニティに提案し，実装を推進する．

参考文献

[1] World Economic Forum 2019: Data Free Flow with Trust (DFFT): Paths towards Free and Trusted Data Flows (2020).

[2] デジタル庁：デジタル社会の実現に向けた重点計画（2022）.

[3] 経済産業省：データの越境移転に関する研究会報告書（2022）.

[4] Voigt, P. and Bussche, A. v. d.: *The EU General Data Protection Regulation (GDPR): A Practical Guide*, Springer Publishing Company, Incorporated, 1st edition (2017).

[5] Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System (2009).

[6] Buterin, V.: A Next Generation Smart Contract and Decentralized Application Platform (2013).

[7] Androulaki, E., Barger, A., Bortnikov, V. et al.: Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains, *Proceedings of the Thirteenth EuroSys Conference*, EuroSys '18, New York, NY, USA, Association for Computing Machinery, pp.1–15 (online), DOI: 10.1145/3190508.3190538 (2018).

[8] ConsenSys: Quorum, ConsenSys (online), available from 〈https://github.com/ConsenSys/quorum〉 (accessed 2024–08–27).

[9] Brown, R., Carlyle, J., Grigg, I. and Hearn, M.: The Corda Platform: An Introduction (2016).

[10] McKeen, F., Alexandrovich, I., Berenzon, A. and et al.: Innovative instructions and software model for isolated execution, *Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy*, HASP '13, New York, NY, USA, Association for Computing Machinery, (online), DOI: 10.1145/2487726.2488368 (2013).

[11] Kosba, A., Miller, A., Shi, E., Wen, Z. and Papamanthou, C.: Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts, *2016 IEEE Symposium on Security and Privacy (SP)*, pp.839–858 (online), available from 〈https://doi.org/10.1109/SP.2016.55〉 (2016).

[12] Kang, H., Dai, T., Jean-Louis, N., Tao, S. and Gu, X.: FabZK: Supporting Privacy-Preserving, Auditable Smart Contracts in Hyperledger Fabric, *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp.543–555 (online), DOI: 10.1109/DSN.2019.00061 (2019).

[13] Benhamouda, F., Halevi, S. and Halevi, T.: Supporting private data on hyperledger fabric with secure multiparty computation, Vol.63, No.2–3 (online), DOI: 10.1147/JRD.2019.2913621 (2019).

[14] Zhou, J., Feng, Y., Wang, Z. and Guo, D.: Using Secure Multi-Party Computation to Protect Privacy on a Permissioned Blockchain, *Sensors*, Vol.21, No.4 (online), DOI: 10.3390/s21041540 (2021).

[15] LayerX: Anonify, LayerX (online), available from 〈https://www.anonify.layerx.co.jp〉 (accessed 2024–08–27).

[16] Cheng, R., Zhang, F., Kos, J., He, W., Hynes, N., Johnson, N., Juels, A., Miller, A. and Song, D.: Ekiden: A Platform for Confidentiality-Preserving, Trustworthy, and Performant Smart Contracts, *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp.185–200 (online), DOI: 10.1109/EuroSP.2019.00023 (2019).

[17] Desai, H. and Kantarcioglu, M.: SECAUCTEE: Securing Auction Smart Contracts using Trusted Execution Environments, *2021 IEEE International Conference on Blockchain (Blockchain)*, pp.448–455 (online), available from 〈https://doi.org/10.1109/Blockchain53845.2021.00069〉 (2021).

[18] Wu, Y., Liu, C., Sebald, L., Nguyen, P. and Yesha, Y.: Apply Trust Computing and Privacy Preserving Smart Contracts to Manage, Share, and Analyze Multi-site Clinical Trial Data, *The International Conference on Deep Learning, Big Data and Blockchain (DBB 2022)*, Cham, Springer International Publishing, pp.3–14 (online), available from 〈https://doi.org/10.1007/978-3-031-16035-6_1〉 (2023).

[19] Benhamouda, F., Halevi, S. and Halevi, T.: Supporting Private Data on Hyperledger Fabric with Secure Multiparty Computation, *2018 IEEE International Conference on Cloud Engineering (IC2E)*, pp.357–363 (online), DOI: 10.1109/IC2E.2018.00069 (2018).

[20] Brandenburger, M., Cachin, C., Kapitza, R. and Sorniotti, A.: Blockchain and Trusted Computing: Problems, Pitfalls, and a Solution for Hyperledger Fabric (2018).

[21] Koshi, I., Nao, N., Yoji, O. and et al.: Secure and Traceable System for Genomic Data Sharing Using Hyperledger Fabric Blockchain, *Informatics In Biology, Medicine and Pharmacology 2020*, IIBMP 2020 (2020).

[22] 池川航史，西島　直：ブロックチェーンを用いた信頼ある機密データの管理および利活用基盤，電子情報通信学会論文誌 D, Vol.105, No.11, pp.653–656（オンライン），DOI: 10.14923/transinfj.2021SGL0001（2022）.

[23] Senkin, S., Moody, S., Díaz-Gay, M. et al.: Geographic variation of mutagenic exposures in kidney cancer genomes, *Nature*, Vol.629, No.8013, pp.910–918 (online), DOI: 10.1038/s41586-024-07368-2 (2024).

[24] Horie, S., Saito, Y., Kogure, Y., Mizuno, K., Ito, Y., Tabata, M., Kanai, T., Murakami, K., Koya, J. and Kataoka, K.: Pan-Cancer Comparative and Integrative Analyses of Driver Alterations Using Japanese and International Genomic Databases, *Cancer Discovery*, Vol.14, No.5, pp.786–803 (online), DOI: 10.1158/2159-8290.CD-23-0902 (2024).

[25] Li, H. and Durbin, R.: Fast and accurate short read alignment with Burrows-Wheeler transform, *Bioinformat-*

ics, Vol.25, No.14, pp.1754–1760 (2009).
[26] Danecek, P., Bonfield, J. K., Liddle, J., Marshall, J., Ohan, V., Pollard, M. O., Whitwham, A., Keane, T., McCarthy, S. A., Davies, R. M. and Li, H.: Twelve years of SAMtools and BCFtools, *GigaScience*, Vol.10, No.2, p. giab008 (online), DOI: 10.1093/gigascience/giab008 (2021).
[27] Robinson, J. T., Thorvaldsdottir, H., Turner, D. and Mesirov, J. P.: igv.js: an embeddable JavaScript implementation of the Integrative Genomics Viewer (IGV), *Bioinformatics*, Vol.39, No.1, pp.1–2 (online), DOI: 10.1093/bioinformatics/btac830 (2022).

池川 航史

2019 年筑波大学大学院システム情報工学研究科博士前期課程修了．同年株式会社日立製作所入社．Web3/ブロックチェーンに関する研究開発に従事．