

2025 年度 卒業論文

分散型 IoT データ管理に向けた IPFS を用いた
DID 発行・検証システム

2026 年 2 月 10 日

システム工学部システム工学科
(学生番号: 60276128)

竹内 結哉

和歌山大学システム工学部

目 次

1	はじめに	3
2	関連研究	5
3	準備	6
3.1	InterPlanetary File System (IPFS)	6
3.2	ブロックチェーン	6
3.3	Decentralized identifier (DID)	6
4	提案システム	7
4.1	システム概要	7
4.2	システム構成要素	7
4.3	システム全体の処理フロー	8
4.4	検証フェーズ	11
4.5	スマートコントラクトによる登録処理	13
5	実験と考察	14
5.1	実験目的	14
5.2	実験環境	14
5.3	実験方法	15
5.4	実験結果	16
5.5	考察	21
6	まとめ	23

概要

近年、IoT 機器の普及に伴い、生成・収集されるデータ量は急激に増加しており、これらのデータを安全かつ効率的に管理するための基盤技術が重要な研究課題となっている。従来の中央集権型データ管理方式では、スケーラビリティ、セキュリティ、プライバシー保護の観点で課題が指摘されており、その解決策としてブロックチェーンや分散型ストレージを用いた分散型データ管理手法が注目されている。

一方で、ブロックチェーンと IPFS を組み合わせた既存研究の多くは、データの改ざん耐性や分散管理の有効性を示しているものの、分散環境において「誰がそのデータを生成・保有しているか」を中央管理者に依存せずに検証する仕組みについては十分に扱われていない。

本研究では、この課題に着目し、Decentralized Identifier (DID) および Verifiable Credential (VC) を導入した **DID 発行・検証システム**を提案する。提案システムでは、IoT データ本体を InterPlanetary File System (IPFS) に保存し、その参照情報である CID とデータ所有者の識別子である DID をブロックチェーンに記録する。さらに、発行者が IoT データの真正性を保証する VC を発行し、検証者が DID Document および VC を用いてデータおよび主体の正当性を検証することで、分散環境においても第三者による検証可能性を実現する。

本研究では、提案システムを実装し、DID の発行・記録、IoT データの IPFS への保存、CID および DID のブロックチェーンへの記録、VC の発行・署名・検証に至る一連の処理について機能動作試験を行った。また、IPFS のデータアップロード処理、ブロックチェーンへの CID 記録処理、VC の発行および検証処理に要する時間を計測し、DID/VC の導入がシステム全体の性能に与える影響を評価した。

その結果、提案システムは想定通り正しく動作することが確認され、VC の発行処理は性能上のボトルネックとならない一方、VC の検証処理は他の処理と比較して相対的に長い処理時間を要することが明らかとなった。ただし、検証処理はデータ取引時など限定的なタイミングで実行されることを想定しており、通常の運用においてはシステム全体の性能に与える影響は限定的であると考えられる。

以上より、本研究は DID および VC を統合することで、IoT データの真正性とデータ保有者の正当性を分散環境において検証可能とする **DID 発行・検証システム**を構築し、その実装可能性および性能特性を示した。

1 はじめに

近年, IoT 機器の爆発的な増加に伴い, 生成・収集されるデータ量は急激に増加している [1]. IoT は家庭, 産業, 医療, 農業など多様な分野で活用されるようになり, センサデータやログデータなど, 生成されるデータの種類や粒度も一層多様化している. このような背景から, 膨大かつ多様な IoT データを安全かつ効率的に管理するための基盤技術が重要な研究課題となっている. 従来, IoT データはクラウドサーバを中心とした中央集権型アーキテクチャによって管理されることが一般的であった. しかし, このような中央集権型データ管理には主に以下の 3 つの課題が指摘されている. 第一に, スケーラビリティの問題である. IoT デバイス数の増加に伴い, データを集中的に保存・処理する中央サーバへの負荷が増大し, 処理能力や通信帯域の限界に達する可能性がある. 第二に, セキュリティの問題である. 中央サーバは単一障害点となりやすく, 攻撃や障害が発生した場合, システム全体に甚大な影響を及ぼす. 第三に, プライバシー保護の問題である. 個人情報を含む IoT データが一か所に集中することで, 情報漏洩時の被害が大規模化するリスクが高まる.

これらの課題を解決するためのアプローチとして, 近年ではブロックチェーン技術と IoT を組み合わせたデータ管理手法が注目されている. ブロックチェーンは, 改ざん耐性や透明性を備えた分散型台帳技術であり, IoT データの真正性検証やアクセス制御, データ履歴の追跡などへの応用が検討されている. 特に, IoT データのハッシュ値やメタ情報をブロックチェーン上に記録することで, データ改ざんの検知や信頼性の担保を実現する研究が進められている.

このような背景のもと, ブロックチェーンを用いた IoT データ管理およびアクセス制御に関する研究が数多く報告されている. Liu らは, Hyperledger Fabric と属性ベースアクセス制御を組み合わせた分散型アクセス制御システムを提案している [2]. また, Ma らは, 許可型ブロックチェーンに基づく IoT ビッグデータ管理スキームを提案している [3]. さらに, Reyna らによる研究では, スケーラビリティ, セキュリティ, プライバシーに加え, 主体識別やアイデンティティ管理が重要な課題であることが指摘されている [4]. Patil らも, IoT アクセス制御やセキュリティを中心としたブロックチェーン応用に関する既存研究を整理しており, 分散化と改ざん耐性の有効性を示している [5]. Ayoade らは, ブロックチェーンと TEE を組み合わせた IoT データ管理手法を提案しているが, 主体の自己主権的な識別という観点では制約が残る [6].

一方で, ブロックチェーンは取引データを全ノードで共有・保持する特性を持つため, 大容量データや個人情報をそのまま格納することは, スケーラビリティやプライバシーの観点から現実的ではない. この問題に対する解決策として, データ本体は分散型ストレージに保存し, ブロックチェーンには検証に必要な最小限の情報のみを記録するという設計思想が広く採用されている. その代表例が, 分散型ファイルシステムである InterPlanetary File System [7] (以下 IPFS) を併用する構成である. IPFS はコンテンツアドレス方式によりデータを識別するため, 大容量データの分散保存と改ざん検知を両立できる点に特徴がある. しかしながら, ブロックチェーンと IPFS を組み合わせた既存研究の多くは, 「そのデータが誰のものであるか」, あるいは「正当な主体が提示しているか」といったデータ提供主体の識別と正当性検証については十分に扱っていない. 分散環境においては, 中央管理者による認証に依存しない主体識別の仕組みが不可欠であり, この点が分散型 IoT データ管理における未解決課題となっている.

以上のように, 既存研究では, ブロックチェーンを用いた IoT データの改ざん耐性確保や分散型管理の有効性が示されている一方で, 分散環境において「誰がそのデータを生成・保有しているのか」を中央管理者に依存せずに証明可能とする仕組みについては, 十分に確立されているとは言い難い. 本研究は, この課題に着目し, データ管理と主体識別を同時に満たす DID 発行・検証システムの構築を目指す点に特徴がある.

この課題に対する有力なアプローチとして, 近年, 自己主権型 ID (Self-Sovereign Identity) の概念が注目されている. 自己主権型 ID とは, 特定の中央管理者に依存せず, 個人や組織が自らの識別子を生成・管理するという考え方であり, Web3.0 が目指す分散型インターネットにおける基盤技術の一つと位置付けられている. Decentralized Identifier [8] (以下 DID) は W3C によって定められている標準規格であり, ブロックチェーンなどの改ざん耐性を持つ基盤と組み合わせることで, 第三者により検証可能な主体識別を実現できる.

本研究では, これらの背景を踏まえ, IPFS, ブロックチェーン, および DID/VC を統合した DID 発行・検証システムを提案する. 提案システムでは,

- 大容量かつ個人情報を含む IoT データ本体は IPFS に保存する
- データの参照情報や検証情報はブロックチェーンに記録する

- データ所有者および生成主体の正当性は DID および VC により検証可能とする

という役割分担を採用することで、スケーラビリティ、セキュリティ、および主体の真正性検証を両立する分散型データ管理基盤の実現を目指す。

本論文の構成は以下のとおりである。第 2 章では、IoT データ管理およびブロックチェーン、IPFS、DID/VC に関する関連研究について整理する。第 3 章では、本研究で用いる基盤技術の概要と実験環境について説明する。第 4 章では、提案する DID 発行・検証システムの構成および処理フローについて詳述する。第 5 章では、構築したシステムに対する機能動作試験および性能評価の結果を示し、考察を行う。最後に第 6 章では、本研究のまとめと今後の課題について述べる。

2 関連研究

IoT データ管理に関する研究では、デバイス数の増加に伴うスケーラビリティの確保や、データの完全性・真正性をいかに保証するかが重要な課題として指摘されてきた。特に、従来の中央集権型アーキテクチャに基づくデータ管理では、管理主体への依存や単一障害点の存在により、可用性やセキュリティ、プライバシーの観点でリスクが生じやすいことが問題とされている。これらの課題を背景として、近年ではブロックチェーンや分散型ストレージを活用した、中央集権的管理に依存しない IoT データ管理手法が数多く提案されている。

Ali ら [9] は、中央集権的な IoT データ管理における管理主体への依存やスケーラビリティの問題を指摘し、ブロックチェーンと IPFS を組み合わせたモジュラー・コンソーシアム・アーキテクチャを提案している。この手法では、IoT データ本体を IPFS に保存し、ブロックチェーンにはそのハッシュ値のみを記録することで、データ管理の分散化と改ざん耐性を両立している。また、IoT デバイス群ごとにサイドチェーンを構成する設計により、単一のブロックチェーンに処理が集中することを回避し、分散的な IoT データ管理におけるスケーラビリティ確保の重要性を示している。

Krejci ら [10] は、IoT データ配信におけるデータ完全性とリアルタイム性のトレードオフに着目し、ブロックチェーンと IPFS を用いたデュアルチャネル型のデータ配信手法を提案している。提案手法では、リアルタイム性を重視した MQTT による配信と並行して、IPFS 上に保存したデータのハッシュ値をブロックチェーンに記録することで、後からデータの改ざん検証を可能としている。評価実験では、ブロックチェーンを用いた完全性検証が高頻度に実行される処理には不向きである一方、取引時など限定的なタイミングでの検証には有効であることが示されている。

一方、ブロックチェーンを IoT へ適用する際の課題として、コンセンサス処理に起因する遅延やスループット低下が指摘されている。Haque ら [11] は、Delegated Proof of Stake (DPoS) を用いた軽量コンセンサスと IPFS を組み合わせることで、多数の IoT デバイスが存在する環境においてもスケーラブルなデータ管理を可能とするフレームワークを提案している。さらに Haque ら [12] は、DPoS に加えてシャーディング技術を導入し、トランザクション処理の並列化による性能向上を図っており、IoT 環境における処理性能とスケーラビリティ改善の方向性を示している。

これまでに述べた、ブロックチェーンと IPFS を組み合わせることで、中央集権的管理に依存せずに IoT データの完全性および真正性を確保しようとする研究の流れを医療分野へ適用したものとして、Kebira ら [13] は、IoT、ブロックチェーン、IPFS を統合した分散型ヘルスケアシステム「BlockMedCare」を提案している。BlockMedCare は、慢性疾患患者の遠隔モニタリングを対象とし、従来のクライアント/サーバー型医療情報管理における単一障害点やプライバシー侵害のリスクを課題として位置づけている。提案システムでは、医療データ本体を暗号化した上で IPFS に保存し、ブロックチェーンにはそのハッシュ値のみを記録することで、スケーラビリティとデータ完全性を両立している。また、Proof of Authority (PoA) を採用したプライベートブロックチェーンにより、医療用途に求められる処理性能の向上を実現している。さらに、プロキシ再暗号化やスマートコントラクトを用いたアクセス制御により、医療データ共有におけるセキュリティ確保の有効性を示している。

以上の先行研究から、IoT データ管理においては、データ本体を分散ストレージに保存し、ブロックチェーンは参照情報や検証情報の管理に限定して用いる設計が、スケーラビリティと信頼性の両方に有効であることが分かる。これらの研究は中央集権型管理に内在するリスクを回避する手段として、ブロックチェーンおよび IPFS を用いた分散型 IoT データ管理の有効性を示すものである。一方で、データの所有者や提供主体の正当性、すなわち「誰のデータであるか」や「正当な主体が提示しているか」といった主体に関する検証については、分散環境における主体識別の標準化まで含めて統合的に扱う研究は十分とは言えない。

本研究では、これらの知見を踏まえ、DID 発行・検証システムの構築を目指す。DID により分散環境における主体識別を明確化し、IoT データの真正性を第三者が検証可能とすることで、既存研究では十分に扱われてこなかった主体の正当性検証を含むデータ管理について検討する点に本研究の特徴がある。

3 準備

本研究では、分散型データ管理の基盤技術として IPFS、ブロックチェーンおよび DID を用いる。本章では、これらの技術の概要を説明し、さらに本研究で使用した実験環境について述べる。

3.1 InterPlanetary File System (IPFS)

IPFS は世界中のコンピュータ（ノード）に分散的にデータを保存する P2P 型のファイルシステムである。IPFS では中央集権的なサーバーを介さず、データを分散的に管理しており、耐故障性、負荷分散、耐検閲性、改ざん耐性に優れている。特徴は「コンテンツアドレス方式」である点で、保存されたファイルはその内容を基に計算される Content Identifier（以下 CID）によって参照される。CID はファイル内容のハッシュ値であるため、以下の性質を持つ。

- 同一内容のファイルは必ず同じ CID となる。
- 1 ビットでも内容が変更されれば別の CID となる。
- CID から元のデータを推測することはできない。

この仕組みにより、ファイルが改ざんされていないかを CID の比較によって確認できるため、改ざん耐性に優れる。

本研究では、ユーザが保有する IoT データを IPFS に保存し、得られた CID をブロックチェーンに記録することで、データの真正性と参照可能性を確保している。

3.2 ブロックチェーン

ブロックチェーンは、ネットワーク上の複数のノードが同一のデータを共有し、合意形成に基づいて取引履歴を記録する分散型台帳技術である。記録されるデータは複数の取引をまとめた「ブロック」に格納され、各ブロックは直前のブロックのハッシュ値を保持することで鎖状に連結される。この構造により、一部のブロックが改ざんされると以降すべてのブロックの整合性が崩れるため、改ざんは即座に検知される。ここで重要なのは、ハッシュ値の性質である。ハッシュ値はデータから一方向的に算出される識別子であり、内容にわずかな変更があっても全く別の値となる。また、ハッシュ値から元のデータを復元することはできない。ブロックチェーンではこの性質を利用し、データの完全性を保証している。

本研究では、IoT データに対応する CID および後述する DID というユーザの識別子をスマートコントラクト経由でブロックチェーンへ記録することで、データ登録の証跡を改ざん不能に保持できるようにしている。

さらに、既存研究においても議論されているように、Proof-of-Work（以下 PoW）はエネルギー消費が大きく、リソース制約のある IoT 環境には適さないことが指摘されている [10]。そのため、本研究においても PoW ベースのブロックチェーンは採用せず、より実装や評価に適した Ethereum 環境を主に使用することとした。

3.3 Decentralized identifier (DID)

DID は特定の中央管理者に依存せずに個人が自分自身で生成・管理できる識別子であり、分散型デジタルアイデンティティの基盤となる技術である。DID はその DID に紐づく公開メタデータである DID Document が存在している必要がある。DID Document には識別子の情報や DID の所有者が使用する公開鍵の情報が含まれており、DID の正当性や鍵の正しさを検証する場面では DID に紐づいた DID Document を取得して使用される。DID から DID Document を取得するという処理を DID 解決と呼ぶ。DID Document はブロックチェーンなどの改ざん耐性を持つ基盤に保存されることで、第三者がその内容を検証可能となる。

本研究では、ユーザ A および発行者が発行した DID と DID Document をブロックチェーンに登録し、IoT データの所有者であることを証明するための基礎情報として利用する。

さらに本研究のシステムでは、発行者がユーザに対して Verifiable Credential（以下 VC）と呼ばれる証明書を発行し、IoT データが正当なデバイスによって生成されたものであることを保証する仕組みを構築する。VC の検証時には、DID Document に記録された公開鍵により署名を確認し、データの真正性を確認することができる。

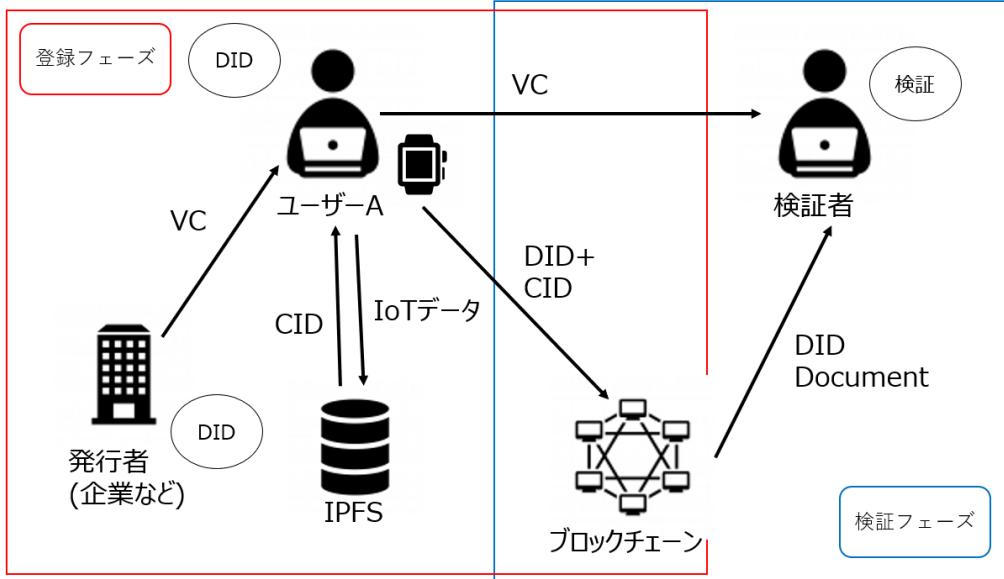


図 1 提案システムの概要図

4 提案システム

本研究で提案するシステムは、IoT データを分散的に管理するために、IPFS、ブロックチェーンおよび DID を連携させたものである。本章では、まずシステム全体の流れを示した後、各要素の役割について説明する。

4.1 システム概要

本研究で提案するシステムの全体像を図 1 に示す。本研究で提案するシステムは、IoT データを保有するユーザ A、データの真正性を確認し VC を発行する発行者、および最終的に VC を検証する検証者により構成される。図 1 に示すように、ユーザ A は自身の IoT データを IPFS に格納し、その結果として得られる CID を、自身の識別子である DID とともにブロックチェーンに記録する。その後、発行者が IoT データの真正性を保証する VC を発行し、検証者がブロックチェーン上の DID Document と VC を突き合わせることで、ユーザ A およびデータの正当性を検証する。

4.2 システム構成要素

4.2.1 データ保有者（ユーザ A）

ユーザ A は、自身の識別子として DID を保持し、DID Document をブロックチェーンに格納する。さらに、保有している IoT データを IPFS に格納し、IPFS から返される CID を取得する。ユーザ A は、CID と自身の DID を組み合わせてブロックチェーンに登録すること、および後に発行者から取得する VC に自身の秘密鍵で署名することで自身が IoT データの正当な保有者であることを保証する。

4.2.2 発行者（企業など）

発行者とは、ユーザ A に対して VC を発行する主体である。本研究ではユーザ A が所持している IoT 機器の製造元企業などを想定している。発行者はユーザ A が保有している IoT データが自社製品によって生成されたデータであることを確認し、その真正性を保証する VC をユーザ A に発行する。

4.2.3 検証者

検証者は、ユーザ A と IoT データを取引する相手、すなわちデータの受領者を想定している。検証者は、ユーザ A から提示された VC を受け取り、ブロックチェーン上の DID Document と照合することで、ユーザ A が真正なデータ保有者であることや、IoT データについて企業が保証していることについて確認することができる。本研究で提案するシステムにおいては、検証者が VC の正当性を確認した上でデータの取引を実行することを想定している。

4.3 システム全体の処理フロー

システムの処理は、登録フェーズと検証フェーズの 2 つに大別される。登録フェーズでは DID・IoT データがブロックチェーンおよび IPFS に記録され、発行者による VC 発行までが行われる。一方、検証フェーズでは、提示された VC の署名検証を通じてユーザ A が真正なデータ提供者であることを確認し、安全なデータ取引を可能にする。

以下では、両フェーズの詳細な処理について説明する。

4.3.1 登録フェーズ

登録フェーズのフローチャートを図 2 に示す。本フェーズは DID の生成・登録、IoT データの保存、CID の登録、および発行者による VC 発行、ユーザ A による VC への署名までの流れで構成される。

4.3.2 DID の生成と DID Document の登録

まず、ユーザ A および発行者はそれぞれ DID を生成し、それに対応する DID Document を作成する。作成した DID Document はブロックチェーンへ登録される。DID Document のブロックチェーンへの登録手順はスマートコントラクト `registerDIDDocument()`(algorithm 1) で規定される。

本研究では、DID の制御主体として Ethereum アドレスを用いる設計を採用している。Ethereum アドレスは ECDSA(secp256k1) 鍵ペアに基づいて生成されるため、当該アドレスで署名可能であることは対応する秘密鍵を保持していることを意味する。この対応関係をブロックチェーン上に登録することで、後続処理における署名検証の基盤を構築する。

以下にユーザ A が生成した DID Document の一例を示す。本 DID Document は最小構成とし、DID とその制御主体である Ethereum アドレスのみを記載している。

```
{
  "id": "did:example:userA",
  "controller": "0x2c854F81C990fDD856fC360f17Dc592366711f08"
}
```

4.3.3 IoT データの保存と CID の取得

次に、ユーザ A は自身が保有する IoT データを IPFS に保存する。この処理により、IoT データはブロックチェーンとは独立した分散ストレージ上に格納され、データ本体を直接ブロックチェーンに保存する必要がなくなる。

IPFS への保存が完了すると、保存されたデータに対応する CID が取得される。本研究では、この CID を IoT データを一意に識別する参照情報として扱い、後続の処理においてユーザ A の DID と組み合わせてブロックチェーンへ記録する。

4.3.4 CID と DID のブロックチェーンへの記録

ユーザ A は取得した CID と自身の DID をブロックチェーンへ記録する。CID と DID のブロックチェーンへの記録はスマートコントラクト `registerIoTData()`(algorithm 2) で規定される。これにより、「どの DID がどの IoT データの所有者であるか」が改ざん耐性を持って記録され、第三者は所有者を検証可能となる。

4.3.5 発行者によるデータ真正性の確認と VC 発行

発行者は、ブロックチェーン上の DID と CID の整合性を確認し、ユーザ A が記録したデータが真正であるかを検証する。正当性が確認された場合、企業はユーザ A に対して VC を発行する。VC には発行者の DID による署名が付与され、内容の真正性が保証される。

4.3.6 ユーザ A による VC への追加署名

最後に、ユーザ A は自身の DID に紐づく秘密鍵を用いて VC に追加署名を行う。これにより、企業とユーザ A の双方が署名した VC が完成し、検証フェーズにおいて提示・検証可能な証明情報となる。

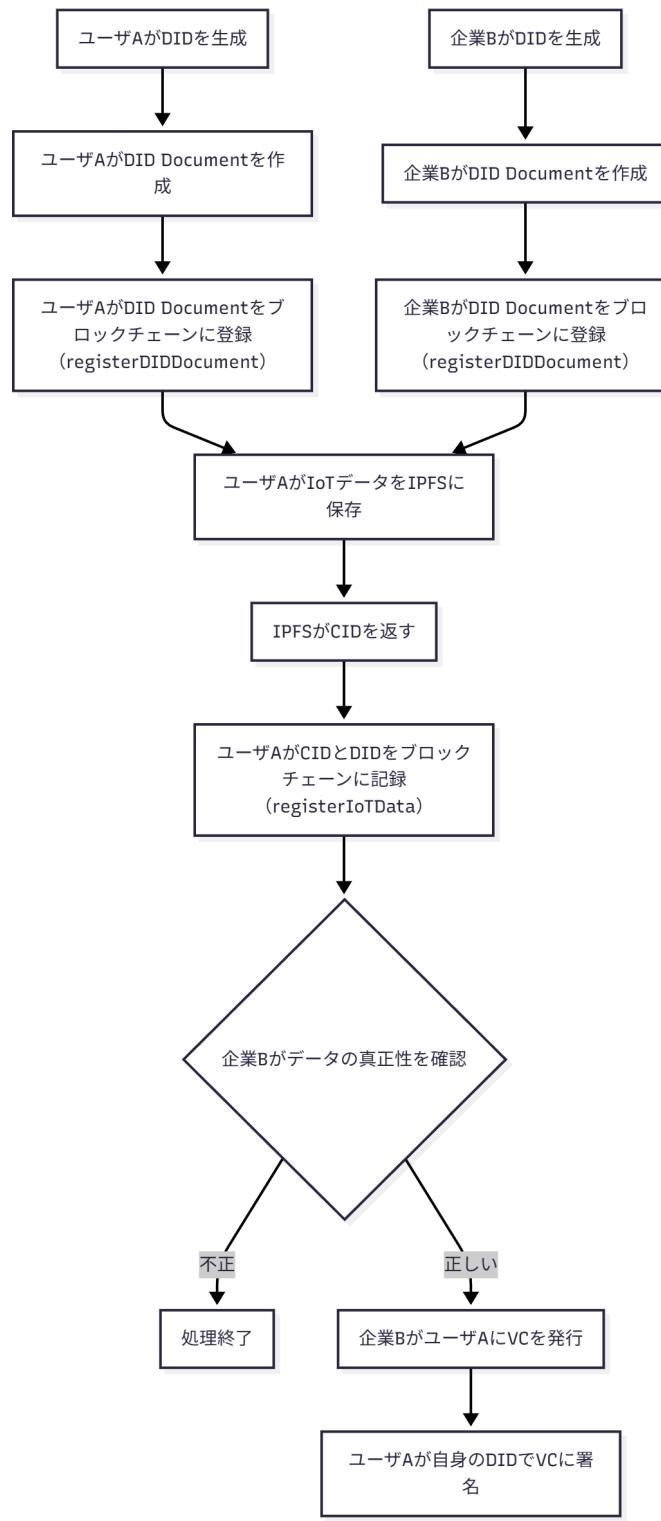


図2 登録フェーズのフローチャート

4.4 検証フェーズ

検証フェーズのフローチャートを図3に示す。本フェーズでは、ユーザAから提示されたVCに含まれる署名を段階的に検証することにより、IoTデータが正当なデバイスによって生成されたものであること、およびユーザAが当該データの正当な保有者であることを確認する。

4.4.1 VCの提示と受領

ユーザAは、発行者と自身の署名が付与されたVCを検証者に提示する。検証者は当該VCを受領し、VCに含まれる署名情報および関連付けられたDID Documentを確認することで、後続の検証処理に備える。

4.4.2 企業の署名検証

検証者はまずVCに含まれる発行者の署名を検証する。署名の公開鍵は、ブロックチェーンへ登録された企業のDID Documentから取得される。署名が不一致であった場合、VCは不正と判断され処理を終了する。

4.4.3 ユーザAの署名検証

企業の署名が正当であった場合、次にユーザAの署名が検証される。ユーザAの公開鍵は同様にDID Documentから取得され、署名が一致する場合、ユーザAがIoTデータの正当な保有者であることが確認される。不一致の場合、処理は終了する。

4.4.4 データの取引の実行

企業およびユーザAの署名がいずれも正当であれば、検証者はユーザAが真正なデータ提供者であると判断できる。この確認を基に、検証者はユーザAと安全にデータ取引を実行する。

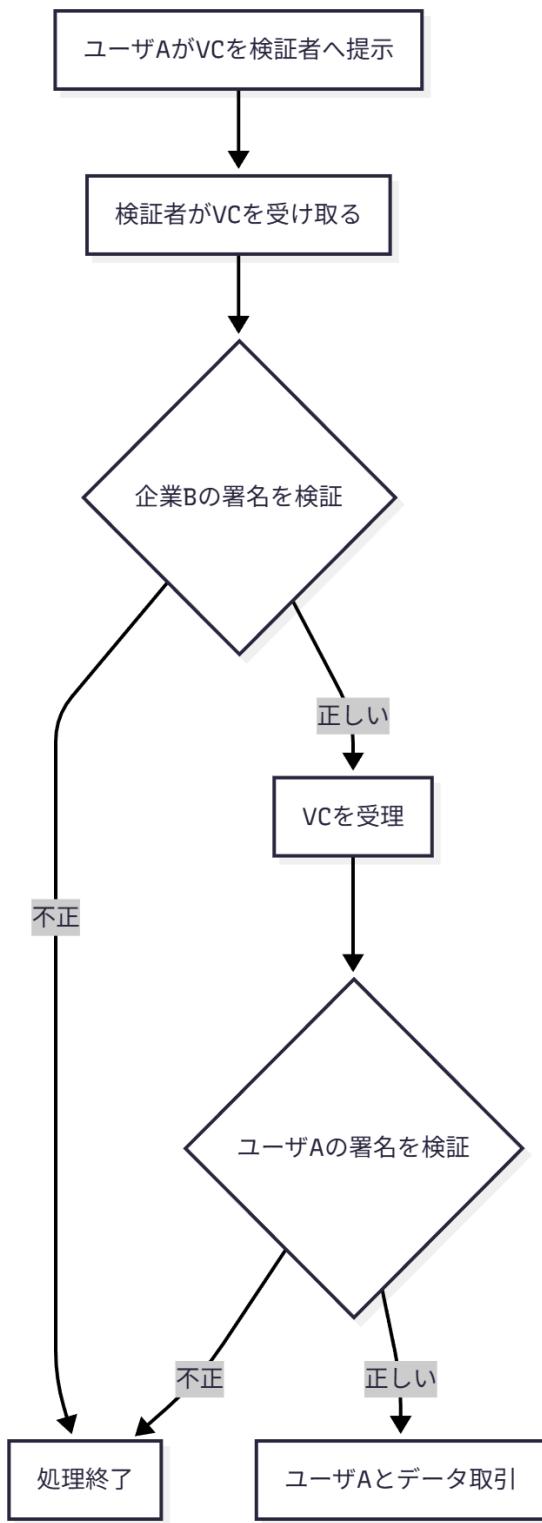


図3 検証フェーズのフローチャート

4.5 スマートコントラクトによる登録処理

本研究で使用したスマートコントラクトでは、ユーザの DID Document および IoT データ (CID) をブロックチェーンに記録するために `registerDIDDocument` および `registerIoTData` の 2 つの関数を提供している。それぞれの処理内容を擬似コードとして以下に示す。

Algorithm 1 DID Document の登録処理 (`registerDIDDocument`)

Require: DID, DID Document (JSON 形式)

- 1:呼び出し元アドレスを取得する（これをユーザ識別子として扱う）
 - 2:DID Document を以下の形式で保存する：
 - 3: `records[呼び出し元アドレス]. append ((DID, DID_Document))`
 - 4:DID 登録イベントを発行する
-

Algorithm 2 IoT データ (CID) の登録処理 (`registerIoTData`)

Require: DID, CID (IPFS で取得したハッシュ値)

- 1:呼び出し元アドレスを取得する
 - 2:IoT データを以下の形式で保存する：
 - 3: `IotData[呼び出し元アドレス]. append ((DID, CID))`
 - 4:IoT データ登録イベントを発行する
-

5 実験と考察

本章では、本研究で構築した IPFS、ブロックチェーン、DID/VC を用いた分散型データ管理システムについて、機能動作試験および性能評価を通してその有効性を検証する。

5.1 実験目的

本研究の実験目的は、大きく 2 つである。1 つ目は提案システムが問題なく動作することを検証することである。具体的には、DID の生成から DID Document の登録、IoT データの IPFS への保存、取得した CID のブロックチェーンへの記録、発行者による VC の発行、および検証者による VC の検証に至るまでの一連の処理を実装し、それらが想定通り正しく実行されることを確認する。これにより、IPFS、ブロックチェーン、DID/VC を統合した提案システムが、データとその所有者の正当性を一貫して保証できることを検証する。

2 つ目は、提案システムの性能を定量的に評価し、DID および VC を統合したことがシステム全体の性能に与える影響を明らかにすることである。特に、IPFS およびブロックチェーンを用いた IoT データ管理に関する先行研究が多数存在することを踏まえ、これらを基準的な処理として位置づけたうえで、DID および VC に関連する処理が新たな性能上のボトルネックとなり得るかどうかに着目して評価を行う。

5.2 実験環境

本研究における実験は、ノートパソコンおよび Raspberry Pi を用いたローカル環境で実施した。図 4 に実際の実験環境の外観を、図 5 に実験環境の構成を示す。

Raspberry Pi は実際の IoT デバイスの代替として使用し、IoT データの生成およびノートパソコンへのデータ送信を行う役割を担う。一方、データ受信後の処理はすべてノートパソコン上で実行しており、IPFS へのデータ保存、ブロックチェーンへの登録、DID および DID Document の作成・登録、並びに VC の発行および検証処理を行っている。ノートパソコン上で使用したソフトウェアおよび実行環境を以下に示す。

- OS:Windows 11 Home
- CPU:AMD Ryzen 5 PRO 7530U with Radeon Graphics (2.00Hz)
- メモリ:16GB
- IPFS:go-ipfs v0.35.0
- ブロックチェーン環境:Ganache v7.9.2
- Solidity: v0.5.16
- Node.js: v18.20.7
- Truffle: v5.11.5
- Web3.js: v1.10.0

また、Raspberry Pi で使用したソフトウェアおよび実行環境を以下のとおりである。

- 機種:Raspberry Pi5
- OS:Raspberry Pi OS (64-bit)
- デスクトップ環境:Raspberry Pi Desktop



図 4 実験環境の外観

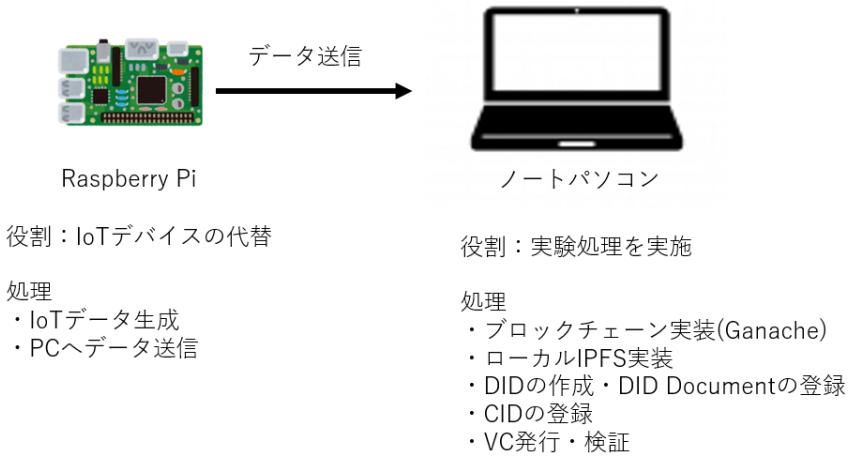


図 5 実験環境の構成

5.3 実験方法

本節では、本研究で構築したシステムに対して実施した実験方法を述べる。実験は大きく次の 2 段階に分けて行った。

1. 機能動作試験

DID Document の登録, IoT データの IPFS への保存, CID のブロックチェーンへの記録, 発行者による VC 発行, 検証者による VC 検証という一連の処理が正しく実行されることを確認した.

2. 性能評価

提案システムにおける主要な処理について処理時間を計測し, VC の発行および検証処理に要する時間と他の処理時間を比較することで, DID/VC の導入がシステム全体の性能に与える影響, 特に処理のボトルネックとなり得るかについて評価を行った. 本評価における測定対象を以下に示す.

- IoT データの IPFS アップロード時間
- ブロックチェーンへの CID 記録時間
- VC 発行処理時間
- VC 検証処理時間

なお, 本研究における「データ 1 件」とは, センサから取得された 1 時点分の IoT データを表す JSON オブジェクト 1 つ分を指す. 具体的には, タイムスタンプ, 温度, 湿度, 照度を含む以下の形式の JSON データ 1 つを 1 件とする擬似データを使用している.

```
{  
    "timestamp": "2025-12-14 18:32:58",  
    "temperature": 21.65,  
    "humidity": 63.41,  
    "light": 926  
}
```

したがって, データ数 10 件, 100 件, 1,000 件とは, 上記形式の JSON データをそれぞれ 10 個, 100 個, 1,000 個まとめたファイルを対象として測定を行ったことを意味する.

5.4 実験結果

5.4.1 機能動作試験の結果

本研究で構築したシステムが, 設計時に想定した一連の処理を正しく実行できるかを確認するため, 各処理を順に実行し, コマンドプロンプト上の出力結果をもとに機能動作試験を行った.

まず, 図 6 に示すように, ユーザおよび発行者がそれぞれ DID を生成し, 対応する DID Document を作成した後, スマートコントラクトを通じてブロックチェーンへ登録する処理を実行した. 出力結果から, DID および DID Document を正常に生成され, トランザクションが成功していることを確認できる.

次に, 図 7 に示すように, IoT データを IPFS へアップロードする処理を実行した. コマンドプロンプトには, アップロード対象ファイルに対応する CID が表示されており, データが IPFS 上に正常に保存されたことが確認できる.

続いて, 図 8 では, IPFS 上に保存されたデータの CID と, そのデータの所有者を示す DID をブロックチェーンへ記録する処理を示している. 出力結果から, CID および DID が対応付けられた形でスマートコントラクトに記録され, 処理が正常に完了していることが確認できる.

図 9 は, 発行者がユーザに対して VC を発行する処理の実行結果を示している. コマンドプロンプトには, VC の生成および発行処理が正常に完了した旨が表示されており, 発行者による VC 発行が正しく行われたことが確認できる.

次に, 図 10 に示すように, ユーザ A が発行された VC に対して署名を行う処理を実行した. 出力結果から, ユーザ A の秘密鍵を用いた署名処理が成功していることが確認でき, VC がユーザ本人によって承認された状態となっていることが分かる.

最後に, 図 11 では, 検証者が VC の検証処理を実行した結果を示している. コマンドプロンプトの出力から, VC の署名検証および DID Document との照合が成功し, VC の正当性が確認されたことが分かる.

以上の結果より, DID の生成・登録, IPFS へのデータ保存, CID と DID のブロックチェーン記録, VC の発行・署名・検証という一連の処理が, 提案システムにおいて問題なく実行できることを確認した.

```
=====
■ Step1: DID ドキュメント 生成 & 登録
=====

[1] DID Document を作成しました。

  ■ UserA DID: did:example:userA
  ■ Company DID: did:example:company
[2] UserA の DID をブロックチェーンへ登録中...
  → UserA の DID を登録しました。

[3] Company の DID をブロックチェーンへ登録中...
  → Company の DID を登録しました。

=====
■ DID ドキュメント 登録 完了
=====
```

図 6 DID および DID Document の生成・登録処理の実行結果

```
=====
■ Step2: IoTデータをIPFSへアップロード
=====

[1] ローカル IPFS ノードへ接続しています...
  → 接続成功

[2] IoTデータファイルを読み込みました
  対象ファイル : demo/data/iot-data.json

[3] IPFS へデータをアップロード 中...

[4] アップロード完了！
  → 取得した IPFS CID: QmRh3fCVx3AuedwpND6TRKpZKVo9TdbtK8cjNDxP9d7TS3

=====
■ IPFS アップロード処理 完了
=====
```

図 7 IPFS へのデータアップロードの実行結果

```

=====
■ Step3: IoTデータ(CID)をブロックチェーンへ登録
=====

[1] UserA の Ethereum アドレス : 0x2c854F81C990fDD856fC360f17Dc592366711f08
[2] IPFS から取得した CID:
    → QmRh3fCVx3AuedwpND6TRKpZKV9TdbtK8cjNDxP9d7TS3
[3] UserA の DID:
    → did:example:userA
[4] ブロックチェーンへ登録処理を送信中...
[5] 登録完了！
    → IoTデータ (DID, CID) をブロックチェーンに保存しました。

=====
■ Step3 完了: IoT データ登録成功
=====
```

図 8 CID および DID のブロックチェーンへの記録処理

```

=====
■ Step4: 企業が IoT データの真正性を保証する VC を発行
=====

[1] Company の Ethereum アドレス :
    → 0x041653bCaB0e61d24236212DdcECc8F9b9E4745A
[2] VC 生成に使用する情報:
    CID      → QmRh3fCVx3AuedwpND6TRKpZKV9TdbtK8cjNDxP9d7TS3
    User DID → did:example:userA
    Company DID → did:example:company
[3] VC オブジェクトを生成中...
[4] 発行者が秘密鍵による VC 署名処理を実行中...
[5] VC の発行が完了しました！

=====
Step4 完了: VC 発行処理が正常に終了しました
=====
```

図 9 発行者が VC を発行する処理

```

=====
■ Step5: UserA による VC 署名
=====

[1] UserA の Ethereum アドレス :
    → 0x2c854F81C990fDD856fc360f17Dc592366711f08

[2] 企業が発行した VC を読み込み中...

● 読み込んだ VC の概要:
  Issuer DID → did:example:company
  Subject DID → did:example:userA
  CID          → QmRh3fCVx3AuedwpND6TRKpZKVo9TdbtK8cjNDxP9d7TS3

[3] UserAによるVC署名処理を実行中...

[4] UserA による署名が完了しました。
  以下が UserA の署名情報です :

===== ● UserA の 署名情報 =====
{
  "type": "EcdsaSecp256k1",
  "created": "2025-12-12T07:49:43.109Z",
  "verificationMethod": "did:example:userA#key-1",
  "hash": "0x9911a002516a2b7e7009192b89c07131fdc5561d3c4fee8677f2929d07d61268",
  "signature": "0x73ff394e52afac70b0f57c99c019bdc9ebf05039175574313a3aafb8fa895e770c1b2ec394fb710fc352
3f420b3ec9f0e00909b3eeb3b6e695af447bf730d3431c"
}
=====

=====
Step5 完了: UserA による VC 署名処理が正常に終了しました
=====
```

図 10 UserA が VC に署名する処理

```

=====
Step6: Verifiable Credential(VC)の検証
=====

[1] 検証対象の VC:
{
  "id": "vc:device-auth:userA",
  "issuer": "did:example:company",
  "subject": "did:example:userA",
  "claim": {
    "cid": "QmRh3fCVx3AuedwpND6TRKpZKV09Tdbtk8cjNDxP9d7TS3",
    "verifiedByDevice": "company-original"
  },
  "proof": {
    "type": "EcdsaS256k1",
    "created": "2025-12-12T06:56:57.139Z",
    "verificationMethod": "did:example:company#key-1",
    "hash": "0x01d28843637f345e1dc64a46a2fb298a5113d2114923dc04e9e690ca595a6709",
    "signature": "0xae86371302c24991e4d86938fbe18c18f091a0b89db56af1ecd82c22a6fd3920a9a86358e571a7471
3fadd767835bd80d2ece4b782c366cfb30837a7f8ddad1c"
  },
  "userProof": {
    "type": "EcdsaS256k1",
    "created": "2025-12-12T07:49:43.109Z",
    "verificationMethod": "did:example:userA#key-1",
    "hash": "0x9911a002516a2b7e7009192b89c07131fdc5561d3c4fee8677f2929d07d61268",
    "signature": "0x73ff394e52afac70b0f57c99c019bcd9ebf05039175574313a3aafb8fa895e770c1b2ec394fb710fc3
523f420b3ec9f0e00909b3eeb3b6e695af447bf730d3431c"
  }
}

[2] Issuer の DID Document を検索...
▶ 所有者アドレス : 0x041653bCaB0e61d24236212DdcECc8F9b9E4745A
▶ DID Document: {
  '0': 'did:example:company',
  '1': '[{"id": "did:example:company", "controller": "0x041653bCaB0e61d24236212DdcECc8F9b9E4745A"}]',
  __length__: 2,
  did: 'did:example:company',
  doc: '[{"id": "did:example:company", "controller": "0x041653bCaB0e61d24236212DdcECc8F9b9E4745A"}]'
}

[3] Subject(UserA) の DID Document を検索...
▶ 所有者アドレス : 0x2c854F81C990fDD856fC360f17Dc592366711f08
▶ DID Document: {
  '0': 'did:example:userA',
  '1': '[{"id": "did:example:userA", "controller": "0x2c854F81C990fDD856fC360f17Dc592366711f08"}]',
  __length__: 2,
  did: 'did:example:userA',
  doc: '[{"id": "did:example:userA", "controller": "0x2c854F81C990fDD856fC360f17Dc592366711f08"}]'
}

[4] IoTデーティアの記録を検索...
▶ DID: did:example:userA
▶ CID: QmRh3fCVx3AuedwpND6TRKpZKV09Tdbtk8cjNDxP9d7TS3

[5] Issuer の署名を検証中...
▶ recover結果: 0x041653bCaB0e61d24236212DdcECc8F9b9E4745A
▶ 登録Issuerアドレス : 0x041653bCaB0e61d24236212DdcECc8F9b9E4745A
 Issuer の署名は正しい

[6] UserA の署名を検証中...
▶ recover結果: 0x2c854F81C990fDD856fC360f17Dc592366711f08
▶ UserAアドレス 0x2c854F81C990fDD856fC360f17Dc592366711f08
 UserA の署名は正しい

=====
Step6 完了: VC検証処理が正常に終了しました
=====
```

図 11 検証者が VC を検証する処理

5.4.2 性能評価の結果

本研究では、提案システムの実運用を想定した際に、DID/VC を統合したことが性能上のボトルネックとなり得るかを評価することを目的として、各処理に要する時間の計測を行った。

性能測定は、各処理は 20 回繰り返し実行した際の平均処理時間を算出することで行った。まず、IPFS への IoT データアップロード処理の性能を評価した。その結果、データ数 10 件 (931bytes) から 1,000 件 (96,793bytes) までのいずれの場合においても、平均処理時間は約 19~21ms で推移しており、データサイズの増加に対して大きな遅延は確認されなかった。また、スループットはデータ数の増加に伴い向上しており、IPFS が IoT データの集約保存に対して十分な性能を有することが確認できた。

次に、ブロックチェーンへの CID 記録処理に要する時間を計測した。本処理では、データ内容に依存しない条件下において、平均処理時間は 63.60ms であり、トランザクション処理として一定の時間を要するものの、安定した性能を示した。

これらの処理と比較し検討するため、VC の発行処理および検証処理に要する時間を計測した。

まず、VC の発行に要する時間を計測した結果、平均処理時間は 10.45ms であり、IPFS アップロード処理およびブロックチェーンへの記録処理と比較しても短い処理時間であった。

一方、VC の検証処理に要する平均時間は 155.78ms であり他の処理と比較して相対的に長い処理時間を要する結果となった。

以上の性能評価結果を表 1 にまとめる。

表 1 各処理における性能評価結果

処理内容	条件	平均処理時間
IPFS アップロード	10 件 (931bytes)	19.11ms
IPFS アップロード	100 件 (9,279bytes)	20.78ms
IPFS アップロード	1,000 件 (96,793bytes)	21.20ms
ブロックチェーン登録	—	63.60ms
VC 発行	—	10.45ms
VC 検証	—	155.78ms

5.5 考察

本章では、前章で示した実験結果を踏まえ、提案システムにおいて DID および VC を統合したことがシステム全体の性能に与える影響について考察する。

5.5.1 DID/VC が性能に与える影響

まず、VC の発行処理に着目する。性能評価の結果より、IPFS への IoT データアップロード時間は約 19~21ms で推移し、ブロックチェーンへの CID 記録処理は平均 63.60ms を要することが確認されている。これらの処理と比較すると、VC の発行処理に要する平均時間は 10.45ms と短く、システム全体の処理性能に与える影響は小さい。このことから、VC の発行処理が提案システムの動作において性能上のボトルネックとなる可能性は低いと考えられる。

次に、VC の検証処理に着目する。VC の検証にかかる処理は 155.78ms であり、IPFS へのアップロード処理やブロックチェーンへの CID 記録処理と比較して、相対的に長い処理時間を要する結果となった。

しかしながら、VC の検証処理は、IoT データの保存や更新といった高頻度に実行される処理とは異なりデータの取引時といった特定のタイミングでの実行を想定している。そのため、通常の運用形態においてはシステム全体の性能に与える影響は限定的であると考えられる一方、検証処理の実行頻度が高くなる場合には、性能上のボトルネックとなる可能性も否定できない。

5.5.2 今後の課題

本稿で提案したシステムは、特定の中央集権的管理主体に依存せず、分散的に IoT データを管理することを目的として設計した。

しかしながら、ユーザが保有する IoT データの正当性を保証する手段として VC を用いる場合、その発行主体として企業などの特定の組織に依存せざるを得ない。この点において、提案システムは一部に中央集権的な管理構造を内包しているといえる。

したがって、完全に分散化されたデータ管理システムの実現という観点からは、データの正当性保証を特定の組織に依存せずに実現する仕組みについて、今後さらなる検討が必要である。

また、IPFS はコンテンツアドレス型の分散ストレージであり、CID を知る第三者が当該データを取得可能であるという特性を有する。そのため、CID が第三者に漏洩した場合においてもデータの内容が不正に閲覧されないよう、データの暗号化などによって機密性を確保する仕組みを併せて検討する必要がある。

6 まとめ

本研究では、IoT 機器の増加に伴い顕在化するデータ管理のスケーラビリティ、セキュリティ、およびプライバシー保護といった課題に対し、ユーザ主権型 ID に基づく DID 発行・検証システムを提案した。提案システムは、IPFS、ブロックチェーン、DID/VC を連携させることで、中央集権的管理主体に依存せずに IoT データの管理と真正性検証を可能とすることを目的として設計した。

本研究では、まず IPFS を用いて IoT データ本体を分散的に保存し、得られた CID をブロックチェーン上に記録することで、データの改ざん耐性と参照可能性を確保した。さらに、DID および DID Document を導入することで、分散環境においてもデータ保有者の識別と公開鍵の正当性を検証可能とした。加えて、発行者が VC を発行し、ユーザ自身も署名を付与する仕組みを構築することで、IoT データが正当なデバイスによって生成されたものであること、ユーザが当該データの正当な保有者であることを第三者が検証できる構成を実現した。

提案システムに対して実施した機能動作試験の結果、DID の生成・登録、IoT データの IPFS への保存、CID と DID のブロックチェーンへの記録、VC の発行・署名・検証という一連の処理が、想定通り正しく実行されることを確認した。これにより、IPFS、ブロックチェーン、DID/VC を統合した分散型データ管理基盤が、データとその保有者の正当性を一貫して保証できることを示した。

また、性能評価を通じて、各処理に要する時間を計測した結果、VC の発行処理は IPFS へのアップロード処理やブロックチェーンへの CID 記録処理と比較して短時間で完了し、システム全体の性能に与える影響は小さいことが確認された。一方、VC の検証処理は他の処理と比較して相対的に長い処理時間を要する結果となったが、本研究で想定する運用形態では高頻度に実行される処理ではないため、通常の利用においては性能上のボトルネックとなる可能性は限定的であると考えられる。

以上より、本研究は、DID/VC を用いることで、IoT データの真正性と保有者の正当性を分散的に検証可能な管理体制を構築できることを示し、その実装可能性および性能特性を明らかにした点に意義がある。

一方で、今後の課題として、データの正当性保証を VC の発行主体となる特定の組織に依存している点が挙げられる。完全に分散されたデータ管理システムの実現を目指すためには、単一の発行主体に依存しない正当性保証の仕組みを検討する必要がある。

また、IPFS の特性上、CID を知る第三者がデータを取得可能であるため、CID の漏洩時にもデータの機密性を確保するための暗号手法やアクセス制御の導入が今後の重要な課題である。さらに、多数の IoT デバイスやユーザが同時に利用する実環境を想定した場合のスケーラビリティや性能への影響についても、より大規模な実験を通じた検証が求められる。これらの課題に取り組むことで、提案システムは実社会における IoT データ管理基盤としてより実用性の高いものへ発展すると考えられる。

参考文献

- [1] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, Vol. 29, No. 7, pp. 1645–1660, September 2013.
- [2] Han Liu, Dezhi Han, and Dun Li. Fabric-iot: A blockchain-based access control system in IoT. *IEEE Access*, Vol. 8, pp. 18207–18218, January 2020.
- [3] Ma Zhaofeng, Wang Lingyun, Wang Xiaochang, Wang Zhen, and Zhao Weizhe. Blockchain-enabled decentralized trust management and secure usage control of IoT big data. *IEEE Internet of Things Journal*, Vol. 7, No. 5, pp. 4000–4015, December 2020.
- [4] Ana Reyna, Cristian Martín, Jaime Chen, Enrique Soler, and Manuel Díaz. On blockchain and its integration with IoT. Challenges and opportunities. *Future Generation Computer Systems*, Vol. 88, pp. 173–190, November 2018.
- [5] Pradnya Patil, M. Sangeetha, and Vidhyacharan Bhaskar. Blockchain for IoT access control, security and privacy: A Review. *Wireless Personal Communications*, Vol. 117, No. 3, pp. 1815–1834, April 2021.
- [6] Gbadebo Ayoade, Vishal Karande, Latifur Khan, and Kevin Hamlen. Decentralized IoT data management using blockChain and trusted execution environment. In *Proceedings of 2018 IEEE International Conference on Information Reuse and Integration (IRI)*, pp. 15–22, July 2018.
- [7] Juan Benet. IPFS - Content Addressed, Versioned, P2P File System, July 2014.
- [8] World Wide Web Consortium(W3C). Decentralized identifiers (dids) v1.1. <https://www.w3.org/TR/did-1.1/>. 閲覧日 : 2026/1/10.
- [9] Muhammad Salek Ali, Koustabh Dolui, and Fabio Antonelli. IoT data privacy via blockchains and IPFS. In *Proceedings of the Seventh International Conference on the Internet of Things*, IoT '17, New York, NY, USA, October 2017. Association for Computing Machinery.
- [10] Simon Krejci, Marten Sigwart, and Stefan Schulte. Blockchain- and IPFS-based data distribution for the Internet of Things. In *Service-Oriented and Cloud Computing*, pp. 177–191, Cham, March 2020. Springer International Publishing.
- [11] Ehtisham Ul Haque, Adil Shah, Jawaid Iqbal, Syed Sajid Ullah, Roobaea Alroobaea, and Saddam Hussain. A scalable blockchain based framework for efficient IoT data management using lightweight consensus. *Scientific Reports*, Vol. 14, No. 1, p. 7841, April 2024.
- [12] Ehtisham Ul Haque, Waseem Abbasi, Ahmad Almogren, Jaeyoung Choi, Ayman Altameem, Ateeq Ur Rehman, and Habib Hamam. Performance enhancement in blockchain based IoT data sharing using lightweight consensus algorithm. *Scientific Reports*, Vol. 14, No. 1, p. 26561, November 2024.
- [13] Kebira Azbeg, Ouail Ouchetto, and Said Jai Andaloussi. BlockMedCare: A healthcare system based on IoT, blockchain and IPFS for data management security. *Egyptian Informatics Journal*, Vol. 23, No. 2, pp. 329–343, July 2022.