2025 年度　卒業論文

DID に基づいた IoT データ管理システムの構築と評価

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

1 はじめに

近年,IoT 機器の爆発的な増加に伴い, 生成されるデータ量は急激に増加している．さらに,IoT は家庭や産業, 医療, 農業など多様な分野で活用されるようになり, 生成されるデータの種類や粒度も一層多様化している．

また, 従来の中央集権型による IoT データ管理には, 以下の 3 つの課題が存在する．第一に, スケーラビリティの問題である.IoT デバイスの急増により, 中央サーバーへの負荷が指数関数的に増大し, 処理能力の限界に達する可能性がある．第二に, セキュリティ上の問題である. 中央サーバーは単一障害点となりやすく, 攻撃対象として脆弱である．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 データが集中することで, 情報漏洩時の被害が甚大化するリスクがある．

これらの課題を解決するために, 本研究ではユーザー主権型 ID に基づいた分散型データ管理システムの実現を目指す．本システムは, 以下の三点を重視して設計されている．

1. データの分散管理：中央集権型から脱却し, 分散型ファイルシステムである InterPlanetary File System(以下 IPFS) を用いることで, 単一障害点を排除しシステムの堅牢性を向上させる．
2. ユーザーの真正性確保：分散管理環境におけるなりすまし防止のため, 分散型識別子である Decentralized Identity(以下 DID) を活用し, データ所有者の身元を保証する．
3. データの信頼性と改ざん防止：ブロックチェーンを活用し, データが改ざんされていないことを検証可能とする．

以上の要素を組み合わせることで,IoT データに対する分散型かつ信頼可能な管理基盤を構築することを目指す．

2 関連研究

IoT データ管理におけるプライバシー確保は, 従来より大きな研究課題とされている．現行の IoT システムは, 多くが集中型のクライアントサーバーモデルに依存しており, 生成される膨大なデータはサービスプロバイダを介して管理される．このような集中型モデルは, ユーザーの行動履歴や個人情報が第三者に漏洩・不正利用されるリスクを内包している．この問題に対処するため, 近年ではブロックチェーンを基盤とした分散型データ管理アーキテクチャの研究が進められている．

Ali ら [1] は, 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．このモデルでは,IoT デバイスをプライベートな「サイドチェーン」にグループ化し, アクセス制御の管理を「コンソーシアムブロックチェーン」によって実現する．サイドチェーンはセンサーデータ生成イベントを記録し, コンソーシアムブロックチェーンはアクセス要求の不変なログを保持することで, プライバシー保護とアカウンタビリティを両立させている．さらに, データ自体は IPFS 上に保存され, ブロックチェーンはハッシュのみを記録することで, ストレージ効率とセキュリティの両立を実現している．

評価実験として,Ethereum(PoW) および Monax(PoS/Tendermint) を用いた性能比較が行われている．その結果,Monax はサイドチェーンレベルで低い処理オーバーヘッドを示す一方で, コンソーシアムレベルでは高いネットワークトラフィックオーバーヘッドが課題となった．Ethereum はコンソーシアムレベルで通信効率に優れるものの,PoW に基づく高い計算コストが問題点として指摘されている．このように, ブロックチェーンのコンセンサスメカニズムの選択は,IoT 分散アーキテクチャの実用性に直接影響を与えることが明らかとなっている．

本研究は, この先行研究の知見を基盤としつつ,IoT データ管理における **DID** と **VC** の統合に焦点を当てる．既存研究が主に公開鍵基盤に依存したアクセス制御を行っていたのに対し, 本研究では DID/VC を導入することで, より柔軟かつ標準化された認証・検証モデルを提供する点に新規性がある．また,Ethereum および IPFS を用いたローカル環境での実装と性能測定を通じて, スケーラビリティと実用性の両面から評価を行うことを目的としている．

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,DID, およびブロックチェーンを用いる． 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 IPFS

IPFS は, 分散型のファイルシステムであり, コンテンツ指向のアドレッシング方式を採用している．従来の URL が「場所」に基づいてデータを参照するのに対し,IPFS ではデータ内容から生成されるハッシュ値を用いて「内容」を参照する．これにより, 同一のデータは同一のハッシュ値で一意に識別され, データの改ざん検知が容易になるとともに, ネットワーク上の複数ノードに分散保存することで耐障害性を高めることができる．本研究では,IPFS の実装として `go-ipfs` を用いた．これは IPFS の公式実装であり,Go 言語で開発されている．

3.2 DID

DID は, 自己主権型アイデンティティを実現するための分散型識別子である. 従来のインターネットにおける認証・識別は, 中央集権的な認証局やサービス提供者に依存していたため, 単一障害点や利用者のプライバシー保護に課題があった. これに対し DID は, ユーザー自身が管理可能な識別子を提供することで, 第三者に依存せずに真正性を保証できる仕組みを実現する. DID Document と呼ばれる文書には公開鍵やサービスエンドポイントなどが含まれ, これにより利用者の認証やデータ検証を行うことができる.

3.3 ブロックチェーン

ブロックチェーンは, 分散型台帳技術の一種であり, 取引情報をブロックとして記録し, チェーン上に連結することで, データの改ざん耐性を保証する. 各ブロックは暗号学的ハッシュにより前後のブロックと結合されており, 一部のデータが改ざんされると以降のブロックすべてが不整合となるため, 改ざん検知が容易である. また, ネットワークに参加する複数ノード間で合意形成を行い, 信頼できる台帳を分散的に維持する仕組みを持つ. 本研究では実験環境として Ethereum 互換のローカルブロックチェーン環境である Ganache を使用した. Ganache は開発用に特化したブロックチェーンシミュレータであり, 高速なテスト実行やトランザクションの記録確認を容易に行うことができる.

4 システム構成

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,DID,IPFS, およびブロックチェーン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 全体像

本研究で提案するシステムの全体像を図 1 に示す. 1 に示すように, ユーザー 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ハッシュ値 (以下 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, 発行者であるユーザー B がユーザー A の真正性を保証する Verifiable Credential(以下 VC) を発行し, 検証者であるユーザー C がブロックチェーン上の DID 情報と VC を突き合わせることで, ユーザー A の正当性を検証する.

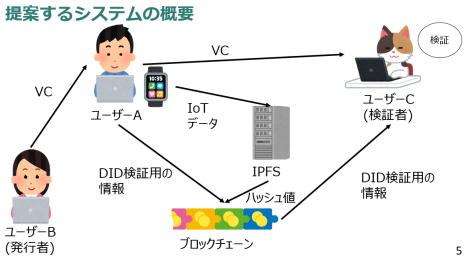

図 1 提案システムの全体像

4.2 ユーザー A(データ所有者)

ユーザー A は, 自身の識別子として DID を保持し, その検証用情報 (公開鍵など) をブロックチェーンに格納する. さらに, 所持している IoT データを IPFS に格納し,IPFS から返される CID を取得する. この CID はデータの一意なハッシュ値であり, データ改ざん検知に利用できる. ユーザー 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で, データの所有者であることを保証する.

4.3 ユーザー B(発行者)

ユーザー B は, 発行者として, ユーザー A の真正性を保証する役割を担う. ユーザー B は, ユーザー A の DID に基づき VC を発行する. この VC には, ユーザー A が正当な主体であることを示す署名が含まれており, 第三者による検証を可能にする.

4.4 ユーザー C(検証者)

ユーザー C は, 検証者として, ユーザー A から提示された VC を受け取り, ブロックチェーン上に記録された DID 検証用情報と突き合わせて検証を行う. これにより, ユーザー A が正当なデータ所有者であることを確認できる. この仕組みによって,IoT データの分散的な流通においてもユーザーの信頼性とデータの真正性が担保される.

4.5 データフロー

本システムの具体的な処理の流れを図 2 に示す. 1 に示すように, ユーザー A は IoT データを IPFS に格納し, 得られた CID と自身の DID をブロックチェーンに登録する. その後, ユーザー B が VC を発行し, ユーザー A はこれをユーザー C に提示する. ユーザー C はブロックチェーン上の DID 情報と VC を照合することで, ユーザー A の正当性を検証する.

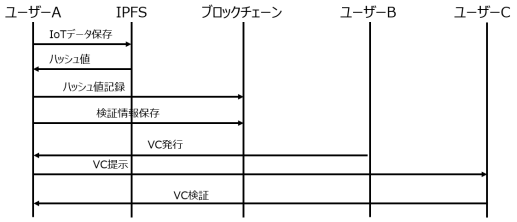

図 2　データフロー図

5　実験と考察

5.1　実験目的

本実験の目的は, 提案システムにおける各処理の政党を定量的に評価し, 実社会での適用に耐えうる処理速度を有しているかを検証することである. 特に,IoT データの分散保存にかかる処理時間, DID および VC を用いたユーザー検証の処理時間, およびブロックチェーンにおけるトランザクション処理時間を測定対象とした.

5.2　実験環境

実験は以下の環境で行った.

- 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

5.3　実験方法

IPFS,DID,VC, ブロックチェーンに対応する各処理について, ベンチマークスクリプトを用いて 10 回の測定を行い, 平均時間を算出した. 測定対象は以下のとおりである.

1. IoT データ (約 2.7KB) の IPFS アップロード処理
2. ユーザー A に対する VC 発酵処理
3. ユーザー A の VC 検証処理
4. DID 登録および発行に関するブロックチェーン処理

5.4　実験結果

表 1 に各処理の平均処理時間を示す.

表 1　各処理のベンチマーク結果 (平均値)

処理内容	平均処理時間	備考
IPFS アップロード	63.0ms	2.7KB データ,0.044 MB/s
VC 発行	5.99ms	10 回平均
VC 検証	93.7ms	10 回平均
DID 登録	88.2ms	gasUsed = 253,446
DID 発行	46.6ms	gasUsed = 49,928

IPFS へのアップロードはデータサイズ 2.7KB の場合に平均 63.0ms であり, スループットは 0.044 MB/s であった. VC 発行処理は平均 5.99ms と高速に実行可能であったのに対し,VC 検証処理は平均 93.7ms を要した. また, ブロックチェーンへの DID 登録は平均 88.2ms,DID 発行処理は 46.6ms であり, いずれも 100ms 未満で処理可能であった.

5.5　考察

6　まとめ

参 考 文 献

[1] 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, 2017. Association for Computing Machinery.