

# User-Managed Access に基づくクロスドメイン認可フレームワーク

江澤 友基<sup>†1,\*</sup> 掛井 将平<sup>†2</sup> 白石 善明<sup>†1, †3, †4</sup>  
瀧田 慎<sup>†5</sup> 毛利 公美<sup>†6</sup> 森井 昌克<sup>†1</sup>

**概要** : IoT の普及により登場した新たなサービスが蓄積するリソースを活用できれば、サービスのさらなる発展が期待される。しかし、リソースを活用するアプリケーションの増加は、認可を管理するユーザの負担の増加に繋がる。ユーザの負担軽減を考えた時、認可プロトコルの一つである User-Managed Access (UMA) の考え方は有用である。しかし、単純に UMA に従ってシステムを構築する場合、認可システムは単一の信頼された第三者によって管理されるので、悪質な主体による認可に関する情報の侵害を検知できないリスクがある。本稿では、ユーザの負担を軽減するより安全な認可フレームワークとして、ブロックチェーンを用いた UMA に基づくクロスドメイン認可フレームワークを提案する。

**キーワード** : User-Managed Access, ブロックチェーン, Hyperledger Fabric, 認可, 委任

## User-Managed Access-based Cross Domain Authorization Framework with Hyperledger Fabric

Yuki Ezawa<sup>†1,\*</sup> Shohei Kakei<sup>†2</sup> Yoshiaki Shiraishi<sup>†1, †3, †4</sup>  
Makoto Takita<sup>†5</sup> Masami Mohri<sup>†6</sup> Masakatu Morii<sup>†1</sup>

**Abstract**: If we can utilize resources accumulated by new services that have appeared as a result of the spread of IoT, we can expect further development of these services. However, an increase in the number of applications utilizing the resources leads to an increase in a burden on users to manage authorization. To reduce the cost to the users, a concept of User-Managed Access (UMA), which is an authorization protocol, is useful. However, if a system is simply built according to the UMA, there is a risk of not being able to detect breaches of information about authorization by a malicious actor because the authorization system is managed by a single trusted third party. In this paper, we propose a UMA-based cross-domain authorization framework using blockchain as a more secure authorization framework that reduces the burden on users.

**Keywords**: User-Managed Access, blockchain, Hyperledger Fabric, authorization, delegation

### 1. はじめに

Internet of Things (IoT)[1]は、建物や車、家電製品などに搭載されるセンサーデバイスやスマートデバイスによって構成される。IoT を活用することで、これまで数値化できなかったデータリソースを蓄積・処理・分析・連携でき、スマートホームやスマートグリッドのような様々な新しいサービスが実現されている[2]。

リソースへのアクセスを認可する仕組みに OAuth[3]がある。OAuth により、ユーザは自身が所有するリソースを別のサービスでも利用できる。しかし、第三者へのリソースの提供は考慮されていない。また、リソースを管理するシステムごとに認可の設定が必要であり、リソース管理システムの数に比例して認可の管理にかかるコストが増大する。

User-Managed Access (UMA)[4][5]は、リソース管理システ

ムがもつ認可の機構を分離・統合することで、リソース管理システムを横断した認可の管理を実現する。また、第三者への認可が許されており、データリソースの多様な連携に対応できる。しかし、統合された認可システムが単一信頼点となるので、内部不正やシステムの乗っ取りなどにより、設定された認可に関する情報が侵害されても検知できない。

本稿では、ブロックチェーン基盤の一つである Hyperledger Fabric[6]で UMA の認可システムを構成することで、リソース管理システムと認可システムを疎に結ぶ新たな分散型クロスドメイン認可フレームワークを提案する。ブロックチェーンにより認可システムを構成することで、ブロックチェーンを運用する複数の組織に信頼点を分散できる。設定された認可に関する全ての情報はブロックチェーンに記録されるので、認可システムが侵害を受けても設定された認可に関する情報の完全性は維持される。

<sup>†1</sup> 神戸大学大学院工学研究科電気電子工学専攻  
Graduate School of Engineering, Kobe University.

<sup>†2</sup> 名古屋工業大学サイバーセキュリティセンター  
Nagoya Institute of Technology.

<sup>†3</sup> 神戸大学数理・データサイエンスセンター  
Center for Mathematical and Data Sciences, Kobe University.

<sup>†4</sup> 国際電気通信基礎技術研究所  
Advanced Telecommunications Research Institute International.

<sup>†5</sup> 兵庫県立大学社会情報科学部  
School of Social Information Science, University of Hyogo.

<sup>†6</sup> 岐阜大学工学部電気電子・情報工学科  
Faculty of Engineering, Gifu University.  
ezawa@stu.kobe-u.ac.jp

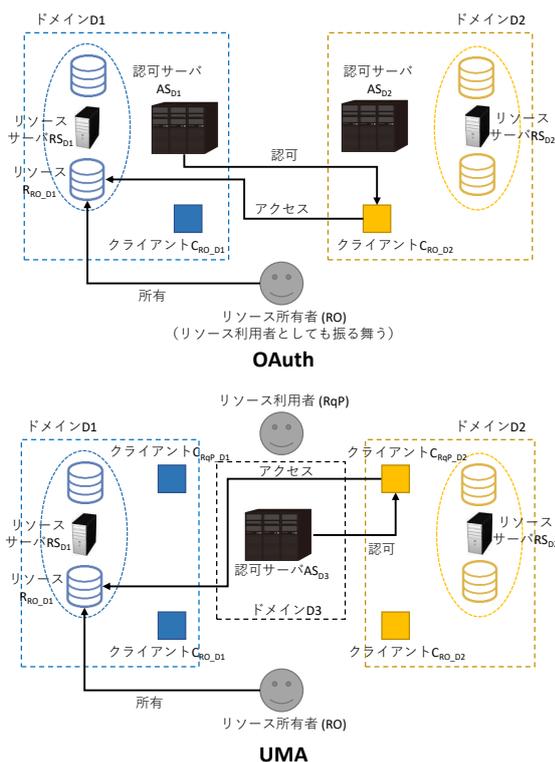


図 1. OAuth と UMA による認可システムの例.

## 2. 従来の認可システム

図 1 に, OAuth と UMA による認可システムの例を示す. リソース所有者は, ドメイン D1 とドメイン D2 に属するサービスにリソースを所有している. リソース利用者は, それらのリソースを専用のクライアントを用いてそれぞれにアクセスできる. OAuth では, 自身のリソースを別のサービスで活用することが想定されている. リソース所有者のクライアント  $C_{RO\_D2}$  がドメイン D1 の認可サーバから認可を受けて, 自身のリソース  $R_{RO\_D1}$  にアクセスできる. 一方で, UMA では, 他者とのリソースの共有も想定されており, リソース利用者のクライアント  $C_{RqP\_D2}$  がドメイン D1 の認可サーバ  $AS_{D1}$  から認可を受けて, リソース所有者のリソース  $R_{RO\_D1}$  にアクセスできる. ここで, リソース利用者によるリソースへのアクセスの分類を表 1 に整理する. ロールベースアクセス制御モデルや属性ベースアクセス制御モデルを用いた従来の認可システムでは, パターン 1 及びパターン 2 を実現できる. 同一ドメイン内におけるシステムであれば, アクセス権限の委任を独自に実装することで, リソースの所有者は他者へのアクセス権限を委任できる. 従来の認可システムに OAuth を取り入れることで, パターン 3 を実現できる. また, UMA を取り入れることで, パターン 3 とパターン 4 を実現できる.

OAuth では, 認可サーバはドメイン毎に運用されるので, リソース所有者はそれぞれで認可の状態を管理しなければならない. 例えば, Twitter, Facebook, Instagram といった

表 1. リソースへのアクセスの種類とその実現方法.

パターン	リソースの所有者	リソースとクライアントのドメイン	実現方法
1	自身	同一	独自実装
2	他者	同一	独自実装
3	自身	別	OAuth/UMA
4	他者	別	UMA

複数の SNS サービスに蓄積されるリソースを外部サービスで利用する場合, SNS サービスのリソースと外部サービスのクライアントの組み合わせの数だけ管理する必要があり, リソース所有者の負担となる. 一方で, UMA では, 認可サーバは独立したドメインで運用され, 複数のリソースサーバと連携できる. ドメイン横断的に認可サーバを利用することで, リソース所有者による認可の管理の負担が軽減される. また, リソース所有者はドメインを超えた他者へのリソースへのアクセス権限を許可できるので, 多様なリソースの連携が期待できる. しかし, OAuth と UMA において認可サーバが単一信頼点となる.

## 3. 関連研究

### 3.1 シングルドメインにおけるブロックチェーンベース認可

単一の信頼点に基づく認可システムにおいて, 認可サーバの管理者が不正にリソースへのアクセス権限を設定した場合に, リソースサーバはそれを検知できない. そこで, ブロックチェーンを用いた分散型認可システムが提案されている. これにより, ブロックチェーンネットワークに参加するノードの一つが不正にリソースへのアクセス権限の設定を試みたとしても, ブロックチェーンネットワーク全体としては完全性が維持される. 文献[7][8][9]はいずれも認可サーバの処理をブロックチェーンで実装することで認可処理の改ざんを防止する. しかし, いずれも権限の委任について考慮されておらず, 認可サーバから認可を付与される他に, リソース所有者から認可を取得する手段が提供されていない.

### 3.2 シングルドメインにおけるブロックチェーンベース認可・権限委任

ブロックチェーンベースの認可システムに, 権限委任の仕組みが統合されたシステムが提案されている. 文献[10]は, スマートコントラクトと OAuth を組み合わせることで, IoT デバイスにおける権限委任を実現する仕組みを提案している. 文献[11]は, ブロックチェーンと属性ベースアクセス制御を組み合わせることで, リソース所有者のリソースをリソース利用者に共有させる仕組みを提案している. 文献[12]は暗号化された属性をブロックチェーン上に記録し, スマートコントラクトを用いて暗号状態のまま属性を比較することでプライバシーの問題に対処しながら認可を行う

仕組みを提案している。文献[13]は、認可サーバが保護するリソースに対する不正な認可の付与に対処するために、ACE フレームワークにおける単一の信頼された認可サーバをブロックチェーンに置き換えることを提案している。文献[14]は、認可及び権限委任の一部のメカニズムをブロックチェーンに移すことで、信頼の一部を分散することを提案している。

### 3.3 クロスドメインにおけるブロックチェーンベース認可・権限委任

クロスドメインでリソースへのアクセスができる認可システムを構築できれば、リソースをより有効に活用できると期待される。文献[15][16]はともに、ブロックチェーンを用いることで安全にクロスドメインにリソースを共有させる仕組みを提案している。しかし、どちらの提案もポリシー決定点である認可サーバとポリシー施行点であるリソースサーバが密に結合している。すなわち、ある認可サーバを用いて特定のリソースサーバに属するリソースだけにしか認可を設定できず、多数のドメインにリソースを所有するリソース所有者は認可の管理に大きな負担がかかる。

## 4. 提案フレームワークの概要

UMA による認可システムは、異なるサービスの認可の機構を統合できる一方で、認可サーバが単一信頼点となる。認可サーバが侵害された場合、その影響は関係する全てのサービスに波及する。このようなネガティブな要素はクロスドメインアクセスを阻害する原因となりうる。本稿ではブロックチェーン基盤である Hyperledger Fabric を用いて認可システムを構成することで、認可サーバの信頼点を分散する。

### 4.1 Hyperledger Fabric

Hyperledger Fabric は、オープンソースで開発されているブロックチェーン基盤である。ブロックチェーンは、特定の中央機関を信頼することなく、複数の端末（ノード）が同じ内容の台帳を分散して保持できる仕組みであり、分散台帳技術とも呼ばれる。

Hyperledger Fabric では、台帳に対する操作をトランザクションと呼び、トランザクションにより更新された値 (RW-Set) は State DB に格納される。一連のトランザクションはブロックとしてまとめられ、ハッシュチェーンにより連鎖されることでトランザクションの耐改ざん性を高められる。Hyperledger Fabric では、この連鎖したデータ構造をブロックチェーンと呼ぶ。台帳に対して許可する操作はチェーンコードとして実装され、チェーンコードを利用することで台帳に対する任意の操作ができないように制限されている。以下に、Hyperledger Fabric を構成するエンティティについて説明する。

**Certification Authority (CA)** : Hyperledger Fabric ネットワークに参加するノードに対して発行されるデジタル証明

書を管理する。

**Peer** : ブロックチェーンネットワークにおける対等なノードであり、ブロックチェーン、State DB、及びチェーンコードを保有する。Peer には Endorser と Committer の2つの役割が割り当てられる。Endorser は、クライアントからの要求に基づいてトランザクションをシミュレートし、RW-Set を作成してこれに同意する。Committer は、トランザクションと実行結果の正当性を検証し、ブロックチェーンと State DB を更新する。

**Orderer** : Endorser によって同意されたトランザクションの結果を、ブロックチェーンに書き込む順番を制御するノードである。一定のルールに従ってソートされたトランザクションは Committer へ送信される。Orderer は Hyperledger Fabric ネットワーク上の全トランザクションの順序を制御するため、単一障害点にならないように冗長構成にする必要がある。

**Organization** : Hyperledger Fabric ネットワークに参加する組織を表す論理的な単位である。各 CA, Peer, 及び Orderer は Organization に所属する。

### 4.2 システム構成

提案フレームワークのアーキテクチャを図2に示す。提案するフレームワークを介してリソースの所有者がリソースの認可をリソースの利用者に与える。本提案フレームワークのエンティティを以下に示す。

**Resource Owner (RO)** : リソース及びリソースに関する全権利を所有するエンティティ。サービスのエンドユーザである。

**Resource Server (RS)** : RO が所有するリソースが蓄積されているサーバ。リソースは Authorization Blockchain によって保護される。

**Authorization Blockchain (AB)** : RO が RS に蓄積するリソースへのアクセスを許可するかどうか決定するための情報をアクセス要求者から収集し、それらに基づいて判断されたアクセス可否の結果を RS へ提供するエンティティ。複数の個人や組織によって構成される。

**Requesting Party (RqP)** : RO の許可を受けて、RS へのアクセスを要求するエンティティ。サービスのエンドユーザである。

**Client** : RqP の指示を受けて、代理として RS 及び AB との間で通信を行うエンティティ。例えば、RqP が Web ブラウザ上で操作するクライアントアプリケーションなどが該当する。

認可ブロックチェーンは以下の7つのスマートコントラクトによって構成される。従来の UMA モデルの認可サーバに実装されるべき5つのエンドポイントに加え、UMA の仕様範囲外である Protection API Token (PAT) を登録するためのエンドポイントとしての PAT Creation Contract 及びポリシーを設定するためのエンドポイントとしての Policy

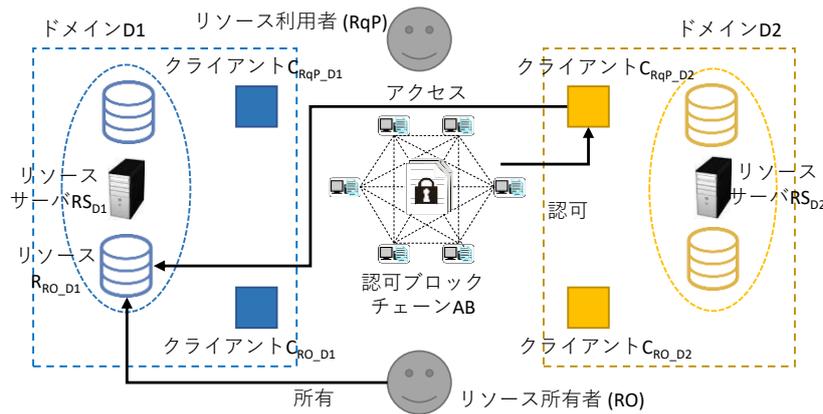


図 2. 認可サーバとリソースサーバを疎に結ぶ分散型クロスドメイン認可フレームワーク (提案フレームワーク).

Creation Contract を実装する.

**PAT Creation Contract** : これは AB に RO のリソースを登録するために使用されるトークンである PAT を要求するスマートコントラクトである. RO は, PAT Creation Contract を実行して得られた PAT を RS に与えることで AB へのリソースの登録を許可する.

**Resource Registration Contract** : これは RO のリソースを AB に登録するスマートコントラクトである. RS は, PAT を引数に Resource Registration Contract を実行して, 登録したいリソースに関する情報を AB に登録する. 登録されたリソース情報は, リソース ID により識別される.

**Policy Creation Contract** : これは AB に登録されたリソース情報に対し, 認可ポリシーを設定するスマートコントラクトである. RO は, ロールベースアクセス制御モデルや属性ベースアクセス制御モデルを用いて, リソースへのアクセスを認めるエンティティの条件を設定する.

**Permission Contract** : これは AB に対して Client の認可を要求するスマートコントラクトである. RS は, Client のリソースへのアクセス要求に応じて Permission Contract を実行し, 認可処理を開始するためのチケットを生成する.

**Token Contract** : これは RS へのアクセスに必要なトークンである Requesting Party Token (RPT) を要求するスマートコントラクトである. Client は, Token Contract を実行して, RPT を取得する. Token Contract の実行において, AB は認可された Client であるかをクレームトークンにより確認する. クレームトークンは, Client の認証情報を証明するもので, OpenID Connect などを利用した外部 ID プロバイダによる ID トークンが利用できる. また, 次に示す Claim Interactive Contract が利用できる. クレームトークンを示さずに Token Contract を実行した場合, AB は外部 ID プロバイダか Claim Interactive Contract のどちらのクレームトークンが必要かを応答する.

**Claim Interactive Contract** : これは AB が必要とするクレームトークンの発行点を示す URI を取得するスマートコントラクトである. 提案システムでは, クレームトークンの取得に OpenID Connect 以外のサービスを利用できる. Client は, Claim Interactive Contract を実行して, 自身に割り当てられた ID 及びリダイレクト用の URI を取得する. クロスサイトリクエストフォージェリの防ぎため, リダイレクト用 URI は必ず事前登録されている必要がある. Client 用 ID の登録処理及びリダイレクト先 URI での認証情報の収集プロセスについては提案内容からは外れるため, 本稿では Client 用 ID は事前に登録され, かつ, リダイレクト先 URI では適切な処理の後にクレームトークンが発行されるものとする.

**Introspection Contract** : これは RPT に紐づく情報を取得するためのスマートコントラクトである. RS は, RPT を引数に Introspection Contract を実行することで, Client の RS へのアクセスを可否を判断する.

### 4.3 システムの流れ

本節では, 提案アーキテクチャ内の各エンティティと認可ブロックチェーンとの間で実行される操作のフローについて述べる. システムの動作フローを図 3 に示す.

#### 4.3.1 リソース登録フェーズ

本フェーズでは, 最初に RO が RS にリソースの登録を許可するために, Step-RR1 で PAT Creation Contract を用いて AB から PAT を発行してもらう. その後, Step-RR3 で RO は発行された PAT を RS に付与する. PAT を受け取った RS は, Step-RR4 で Resource Registration Contract を用いて RO のリソースを AB に登録できる. リソース登録後, RO は AB 上で保護されるリソースに対し, Step-RR6 で Policy Creation Contract を用いて認可ポリシーを設定できる.

#### 4.3.2 認可処理フェーズ

本フェーズではまず, リソースの利用者である RqP が Client に対し, RS にアクセスするように指示する. RS は許可チケットを Step-AP3 の Permission Contract で AB から取得し, これを Client に発行する. 許可チケットは, Client

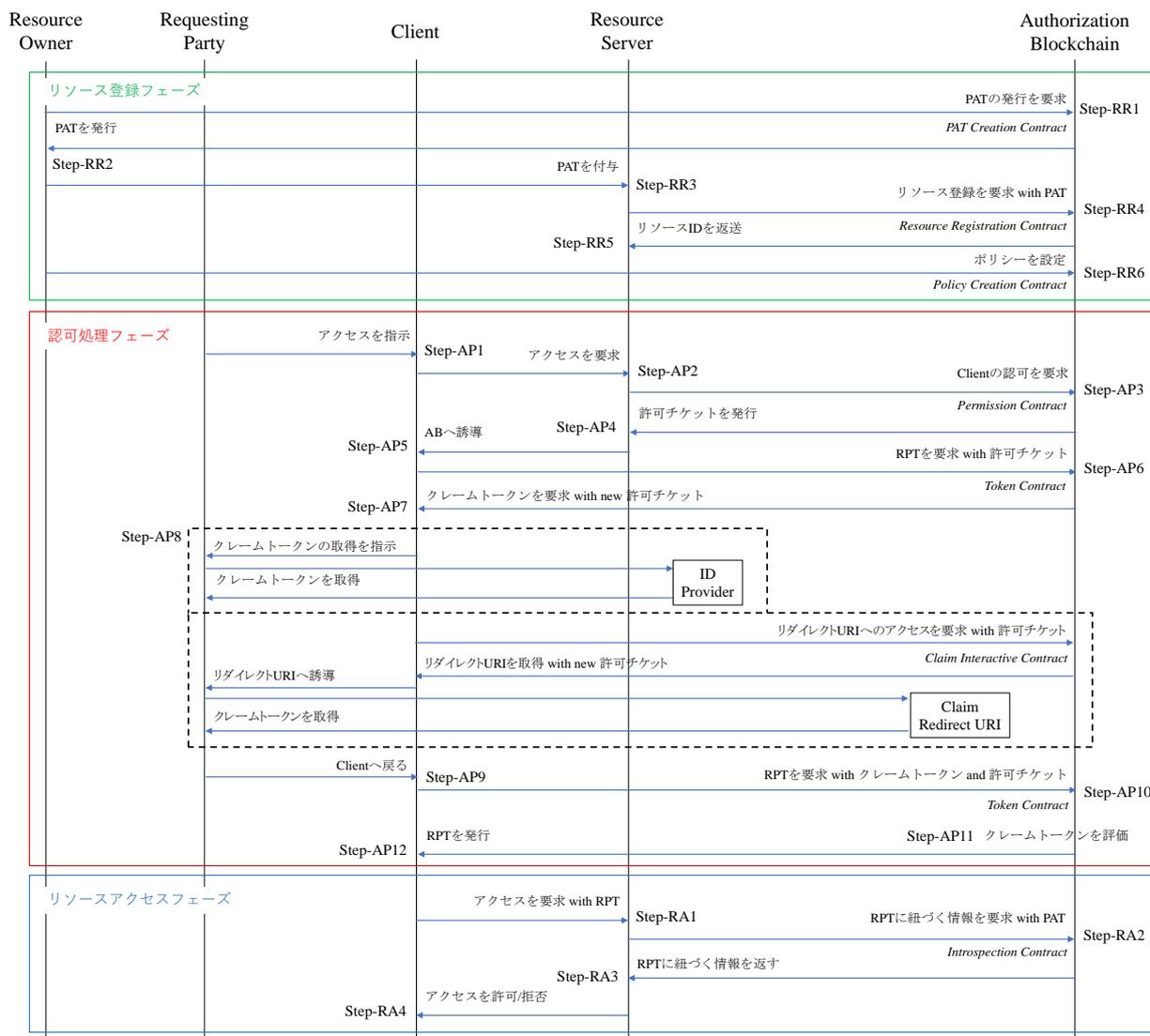


図 3. 提案フレームワークのシーケンス.

がどの RS に対する認可処理かを AB に示すチケットで、以降の処理では、都度 AB に提示する。次に、Client は、Step-AP6 の Token Contract により RPT を要求する。この時点では、Client はクレームトークンを所持していないので、AB は Client にクレームトークンの取得を指示する。クレームトークンの取得処理は図 3 の Step-AP8 に示すように、外部 ID プロバイダか Claim Interactive Contract を使う方法が指示される。クレームトークンを取得した Client は、Step-AP10 において Token Contract を用いて AB に対し RPT を要求する。AB は事前に RO によって設定されたポリシーに照らし合わせてクレームトークンを評価し、正しいことが確認できれば RPT を発行する。

### 4.3.3 リソースアクセスフェーズ

本フェーズでは、Client は認可処理フェーズで取得した RPT を使って RS にアクセスを要求する。アクセス要求を受けた RS は Step-RA2 において、Introspection Contract を

用いて RPT に紐づく情報を AB から取得する。RS は AB から取得した情報をもとに、Client のアクセスの可否を判断する。

## 5. 実装評価

### 5.1 実装環境

本稿では、単一の計算機 - Ubuntu 18.04 LTS 上に VirtualBox v5.2.42 を用いて仮想環境 - Ubuntu 18.04 LTS を構築し、その上で Docker v19.03 を用いて提案フレームワークを実装した。本システムは、“fabric orderer”及び“fabric peer”コンテナを含む Hyperledger Fabric のサンプル実装を用いた。

### 5.2 パフォーマンス評価

本節では、提案フレームワークにおいて、異なる RO-RS の組の数の増加に伴う一連の処理の実行時間、及びその時の台帳サイズの増加量を評価した。

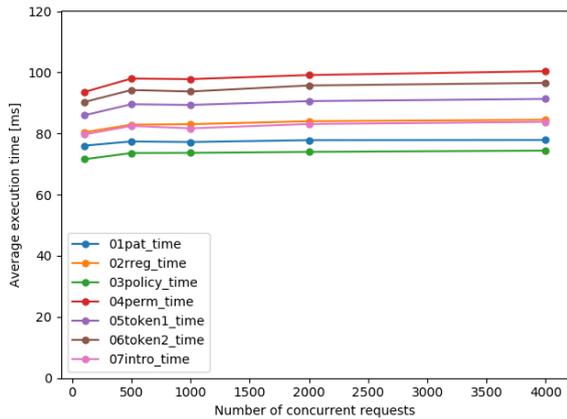


図 4. アカウント数の増加に伴う各チェーンコードの平均実行時間の傾向 - ケース 1.

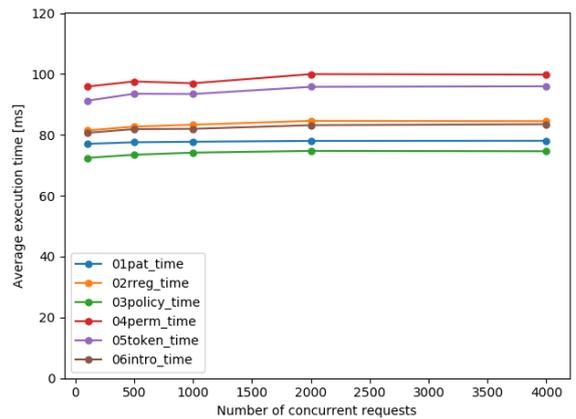


図 6. アカウント数の増加に伴う各チェーンコードの平均実行時間の傾向 - ケース 3.

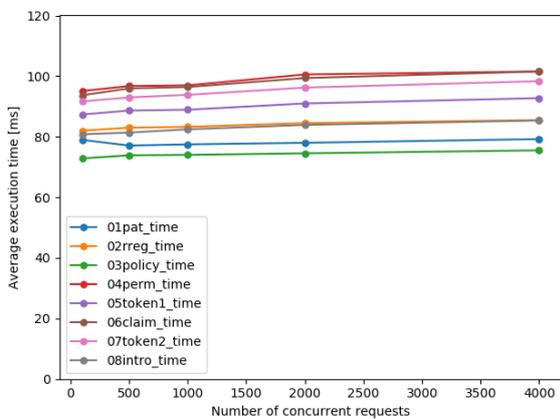


図 5. アカウント数の増加に伴う各チェーンコードの平均実行時間の傾向 - ケース 2.

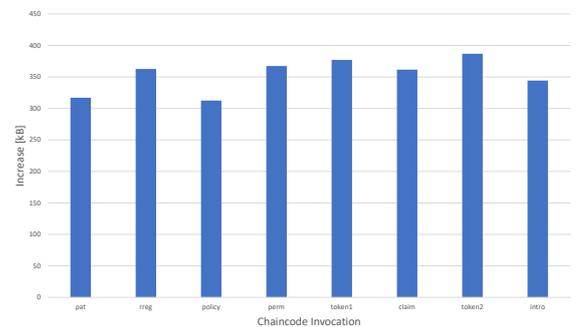


図 7. アカウント 100 件の登録に伴う台帳サイズの増加量.

### 5.2.1 アカウント数の増加に伴うチェーンコードの平均実行時間の傾向

提案フレームワークでは時間の経過とともにシステムを利用するエンティティの増加が予想される。本節では、Client が外部 ID プロバイダからクレームトークンを取得する場合（ケース 1）、AB と対話的にクレームトークンを取得する場合（ケース 2）、及び Client が予めクレームトークンを取得している場合（ケース 3）の 3 つのケースに対して、エンティティが増加した際に、リソース所有者が PAT の発行を要求する時点から、最後にリソースサーバがリクエストングパーティの持つ RPT の情報を要求する点までの一連の処理の実行にかかる時間をそれぞれ計測した。図 4-6 は、一組の RO-RS あたりの各ケースにおけるチェーンコードによるリクエストの処理にかかる時間の平均値の推移を表す。本実験では、おおよその推移を見るために、同時リクエスト数を 100,500,1000,2000,4000 に絞って測定した。実験の結果、どのケースのどのチェーンコードにおいても、リクエスト数の増加に伴う処理時間の変化はほとんど観測されないことがわかった。

### 5.2.2 アカウント数の増加に伴う台帳サイズの増加量

提案システムを利用するエンティティの増加に伴い、エンティティの認可処理に必要な情報を格納する台帳のサイズが増加することが予想される。図 7 に、最も多くの処理手順を必要とするケース 2 における、リクエスト 100 件あたりの台帳サイズの増加量の測定結果を示す。実験の結果、どの処理においても台帳サイズの増加量は 300~400 kbytes の範囲内に収まっていることがわかった。

### 5.3 提案アーキテクチャと既存のアーキテクチャの比較

提案するフレームワークと関連する既存のフレームワークの比較結果を表 2 に示す。次の項目に基づいて関連フレームワークを調査した。

#### A) 認可システムは権限委任をサポートしているか

あるユーザが所有する権限を別のユーザに委任できるかについて評価する。ユーザ間での柔軟なアクセス制御の実現には、システムが一方的にユーザに対して権限を付与するだけではなく、エンティティ間で権限を委任できることが望ましい。

#### B) 認可システムはシングルドメイン認可をサポートしているか

リソースサーバで保護されるあるユーザのリソースに対

表 2. 提案フレームワークと既存のフレームワークの比較.

アクセス制御メカニズム	権限委任	シングルドメイン認可	クロスドメイン認可	リソース所有者とは異なるユーザへの権限委任	PDP-PEP間の疎結合	リソース保護メカニズム	PDPの非中央集権性
[3]	Yes	Yes	Yes	No	No	Cookie	No
[4][5]	Yes	Yes	Yes	Yes	Yes	Cookie	No
[17]	Yes	No	Yes	No	Yes	Cookie	No
[7]	No	Yes	No	No	No	公開鍵	Yes
[8]	No	Yes	No	No	No	No	Yes
[9]	No	Yes	No	No	No	No	Yes
[10]	Yes	Yes	No	No	No	スマートコントラクト	Yes
[11]	Yes	Yes	No	Yes	No	トランザクション	No
[12]	Yes	Yes	No	No	No	スマートコントラクト	No
[13]	Yes	Yes	No	No	No	公開鍵	No
[14]	Yes	Yes	No	No	No	No	No
[15]	Yes	Yes	Yes	Yes	No	スマートコントラクト	No
[16]	Yes	Yes	Yes	Yes	No	スマートコントラクト	Yes
提案フレームワーク	Yes	Yes	Yes	Yes	Yes	Cookie	Yes

して、同じドメインに所属する別のユーザからのアクセスを許可あるいは拒否できるかについて評価する。

C) 認可システムはクロスドメイン認可をサポートしているか

リソースサーバで保護されるユーザのリソースに対して、異なるドメインに所属する別のユーザからのアクセスを許可あるいは拒否できるかについて評価する。

D) リソース所有者から別のユーザへの権限委任ができるか

リソースを所有するあるユーザは別のユーザに対しリソースへのアクセスを許可できるかについて評価する。

E) PDP-PEP 間は疎結合であるか

認可システムにおいて、ポリシーに基づいて決定を下すエンティティである Policy Decision Point (PDP)とその決定を適用するエンティティである Policy Enforcement Point (PEP)が疎結合であるかについて評価する。PDPとして振る舞うエンティティとして、一般の認可システムにおける認可サーバや、本稿の提案フレームワークにおける認可ブロックチェーンが当てはまる。また、PEPとして振る舞うエンティティはリソースサーバである。

F) リソースサーバはどのようなメカニズムで保護されるか

リソースはインターネット上に公開されるので、不特定のエンティティによるアクセスからリソースを保護するためのメカニズムを備えていることが望ましい。

G) PDP は分散型であるか

認可システムにおいてアクセス要求者の認可を決定するための PDP が分散管理されているかについて評価する。

PDP がブロックチェーンによって分散管理される場合、認可システムの認可決定に関わるアルゴリズム及び実行結果は改ざん耐性を持つ。

文献[7][8][9]ではブロックチェーンベースの認可システムを提案しているが、既存の認可システムの PDP を分散化するというアイデアにとどまっており、権限の委任について考慮されていない。文献[10][11][12][13][14]は認可システムにブロックチェーンを取り入れる提案をしているが、いずれもクロスドメイン認可をサポートしておらず、シングルドメインでの認可に限定される。文献[17]では、PDP である権限委任サーバをドメインから分離するクロスドメイン認可システムを提案している。しかし、権限委任サーバは信頼された単一のエンティティによって運営されている。ケイパビリティベースのクロスドメイン認可システムを提案している文献[15]では、ケイパビリティトークンの管理にブロックチェーンを採用しているが、このトークンを発行する PDP は集中型サーバによって構成されており、トークン発行プロセスが保証されない。文献[16]では分散型 BC ベースのクロスドメインリソース共有のための認証認可システムを提案しているが、PDP-PEP 間の構成が密結合になっているため、リソース所有者はドメインごとにリソースの認可を設定する必要がある。

提案アーキテクチャは、ユーザ間で権限の委任ができるブロックチェーンベースのクロスドメイン認可システムである。そのため、システムのユーザは自身が所属しないドメインのリソースに対し、アクセス認可を受けることができる。反対に、リソースへのアクセスを要求するエンティティがユーザの所属するドメインに所属していない場合で

も、ユーザはこのエンティティにアクセスを認可することもできる。PDP-PEP 間は疎結合であるため、リソースサーバ及びリソース所有者の認可に関する処理負担を軽減する。PDP は分散型 BC ベースの構成であるため、認可の決定に関わるアルゴリズムや実行結果は改ざんのないことが保証される。

## 6. おわりに

IoT を活用してこれまで数値化できなかったデータリソースを蓄積・連携することで、新たなサービスの実現が期待される。UMA は、リソース管理システムがもつ認可の機構を分離・統合することで、リソース管理システムを横断した認可の管理を実現する。また、第三者への認可が許されており、データリソースの多様な連携に対応できる。しかし、統合された認可システムが単一信頼点となるので、内部不正やシステムの乗っ取りなどにより、設定された認可に関する情報が侵害されても検知できない。本稿では、ブロックチェーン基盤の一つである Hyperledger Fabric で UMA の認可システムを構成することで、リソース管理システムと認可システムを疎に結ぶ新たな分散型クロスドメイン認可フレームワークを提案する。認可システムは、ブロックチェーンによりその機能が実現されることで、ブロックチェーンを運用する複数の組織に信頼点が分散される。設定された認可に関する全ての情報はブロックチェーンに記録されるので、認可システムが侵害を受けても設定された認可に関する情報の完全性は維持される。提案フレームワークを用いた認可システムを実装し、パフォーマンス評価を行ったところ、ユーザ及びリソースの増加に伴う認可処理の実行時間の増加は見られなかった。また、関連するアクセス制御メカニズムと性能を比較し、本提案が PDP-PEP 間の疎結合を実現する唯一の分散型クロスドメイン認可フレームワークであることを示した。

## 参考文献

- [1] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.
- [2] B. L. R. Stojkoska, K. V. Trivodaliev, "A review of Internet of Things for smart home: Challenges and solutions," *Journal of Cleaner Production* 140, 2017, pp. 1454–1464.
- [3] D. Hardt, "The OAuth 2.0 Authorization Framework," *Internet Requests for Comments, RFC Editor, RFC 6749*, October 2012, <http://www.rfc-editor.org/rfc/rfc6749.txt>. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc6749.txt>
- [4] E. Maler, "User-Managed Access (UMA) 2.0 Grant for OAuth 2.0 Authorization," January 2018, <https://docs.kantarinitiative.org/uma/wg/rec-oauth-uma-grant-2.0.html>. [Online]. Available: <https://docs.kantarinitiative.org/uma/wg/rec-oauth-uma-grant-2.0.html>
- [5] E. Maler, "Federated Authorization for User-Managed Access (UMA) 2.0," January 2018, <https://docs.kantarinitiative.org/uma/wg/rec-oauth-uma-federated-authz-2.0.html>. [Online]. Available: <https://docs.kantarinitiative.org/uma/wg/rec-oauth-uma-federated-authz-2.0.html>
- [6] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, and Y. Manevich, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, 2018, p. 30.
- [7] S. Wang, X. Wang and Y. Zhang, "A Secure Cloud Storage Framework With Access Control Based on Blockchain," in *IEEE Access*, vol. 7, pp. 112713–112725, 2019, doi: 10.1109/ACCESS.2019.2929205.
- [8] D. Di Francesco Maesa, P. Mori and L. Ricci, "Blockchain Based Access Control Services," 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, 2018, pp. 1379–1386, doi: 10.1109/Cybermatics\_2018.2018.00237.
- [9] H. Liu, D. Han and D. Li, "Fabric-iot: A Blockchain-Based Access Control System in IoT," in *IEEE Access*, vol. 8, pp. 18207–18218, 2020, doi: 10.1109/ACCESS.2020.2968492.
- [10] V. A. Siris, D. Dimopoulos, N. Fotiou, S. Voulgaris and G. C. Polyzos, "OAuth 2.0 meets Blockchain for Authorization in Constrained IoT Environments," 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 2019, pp. 364–367, doi: 10.1109/WF-IoT.2019.8767223.
- [11] Y. Zhu, Y. Qin, G. Gan, Y. Shuai and W. C. Chu, "TBAC: Transaction-Based Access Control on Blockchain for Resource Sharing with Cryptographically Decentralized Authorization," 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, 2018, pp. 535–544, doi: 10.1109/COMPSAC.2018.00083.
- [12] O. Mounnan, A. A. E. Kalam and L. El Haourani, "Decentralized Access Control Infrastructure using Blockchain for Big Data," 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates, 2019, pp. 1–8, doi: 10.1109/AICCSA47632.2019.9035221.
- [13] O. Alphand et al., "IoTChain: A blockchain security architecture for the Internet of Things," 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, 2018, pp. 1–6, doi: 10.1109/WCNC.2018.8377385.
- [14] N. Tapas, G. Merlino and F. Longo, "Blockchain-Based IoT-Cloud Authorization and Delegation," 2018 IEEE International Conference on Smart Computing (SMARTCOMP), Taormina, 2018, pp. 411–416, doi: 10.1109/SMARTCOMP.2018.00038.
- [15] R. Xu, Y. Chen, E. Blasch and G. Chen, "BlendCAC: A Blockchain-Enabled Decentralized Capability-Based Access Control for IoTs," 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, 2018, pp. 1027–1034, doi: 10.1109/Cybermatics\_2018.2018.00191.
- [16] G. Ali et al., "xDBAuth: Blockchain Based Cross Domain Authentication and Authorization Framework for Internet of Things," in *IEEE Access*, vol. 8, pp. 58800–58816, 2020, doi: 10.1109/ACCESS.2020.2982542.
- [17] M. Alam, X. Zhang, K. Khan, and G. Ali, "XDAAuth: A scalable and lightweight framework for cross domain access control and delegation," in *Proc. 16th ACM Symp. Access Control Models Technol. (SACMAT)*, 2011, pp. 31–40.