「マルチメディア，分散，協調とモバイル
(DICOMO2021)シンポジウム」 令和3年6月-7月

# 自己主権型アイデンティティを用いた個人情報の流通方式の検討

大森芳彦[1]　　山下高生[1]

**概要：** インターネット上でのサービス間の ID 連携が普及していくにつれて，ユーザが利用する ID は，少数のサービス事業者が管理する ID に集約され，ID に紐づけられたユーザの個人情報は，マーケティングや提携会社との共同利用などの 2 次利用に使われている．この様な背景の中で，ID はユーザ自身で管理する SSI(Self- Sovereign Identity, 自己主権型アイデンティティ)のコンセプトが注目されている．

インターネット上での個人情報の流通，利用については，ユーザは，自身の個人情報の具体的な 2 次流通先を把握するのが困難な場合があり，プライバシーに関する懸念がある．サービス事業者も，取得した個人情報が変更されたことを把握するのが困難な場合があり，サービス提供に影響が発生することが考えられる．

本稿では，SSI を実現する技術として，ユーザのプライバシー保護，およびサービス事業者や 2 次流通先の運用性を向上させるための個人情報の流通方式について提案する．また，提案方式については，処理性能に関する基本測定と評価を行い，1 億人程度のユーザを収容できることが分かった．

# A Study on Sharing Method of Personal Information on Self-Sovereign Identity

YOSHIHIKO OMORI[1]　　TAKAO YAMASHITA[1]

## 1. はじめに

インターネット上で e コマース,SNS(Social networking service),インターネットバンキングなどのサービスが人々の生活に浸透していく中で，GAFA と呼ばれる少数の巨大 IT 企業を中心とするサービス間の ID 連携が普及している．サービスを利用するユーザは，サービス事業者が提供する ID 連携を利用することにより，特定のサービスを利用する時の ID を用いて，他のサービスも利用することができるようになる．そのため，ユーザは，利用するサービス毎に自身の ID を使い分ける必要がなくなり，インターネット上のサービスを利用するにあたって，ユーザの利便性が向上する．このような ID 連携は，OpenIDConnect[1]や SAML[2]などの標準化された技術を用いることで実現することができる．

ユーザがインターネット上のサービスを利用するためにユーザ登録をする際には，ユーザは，自身の名前や住所，生年月日などの個人情報をサービス事業者に提供する．このような個人情報はユーザの ID に紐づけられてサービス事業者で管理され，ユーザ毎のサービスのカスタマイズ，およびマーケティングや新サービスの開発などの 2 次利用に使われている．ID 連携が普及することで，ユーザが利用する ID が少数のサービス事業者が管理する ID に集約され，このようなサービス事業者が ID プロバイダ(IDP)として機能していくと，少数のサービス事業者に，ユーザ登録時にユーザが提供した個人情報の他にも，多数のユーザの多様な個人情報が集まることになる．例えば，ID 連携している他のサービス事業者のサービスについてもユーザの利用動向を収集，把握できるようになる．このように収集されたユーザの個人情報を提携会社と共有し，IT 技術を活用することで，フィンテック[3]やインシュアテック[4]などの金融サービス，ユーザの信用スコア[5]の提供など，新たなビズネスの創出やサービスの提供などに利用することができるようになる．

少数のサービス事業者による中央集権的な ID 管理が行われると，集約された個人情報の効果的なビジネス利用や，ユーザの利便性やサービス利用におけるエクスペリエンスの向上が可能になる．しかし，その一方で，ユーザがサービス事業者に提供した自身の個人情報が，どのように他のサービス事業者や提携会社に流通し，どのように利用されているのかを把握するのが困難になってくると考えられる．

インターネット上での個人情報の流通，利用については，サービス事業者がプライバシーポリシーを規定，公開して，ユーザのプライバシーに配慮している．図 1 に，プライバシーポリシーを例示する．図 1 では，個人情報の 2 次流通先として，サービス事業者のグループ会社や業務委託会社が記載されている．しかし，この例では，会社名などの具体的な 2 次流通先は記載されていなく，この様に，ユーザは 2 次流通先を把握するのは困難なことがあり，プライバシーの保護への懸念がある．

この様な背景の中で，少数のサービス事業者によって個

1 NTT ネットワークサービスシステム研究所

人情報が中央集権的に管理されることを避けるために，ユーザの ID や，その ID に紐づく個人情報は，ユーザ自身で管理する SSI(Self- Sovereign Identity,自己主権型アイデンティティ)のコンセプトが注目されている．

SSI を実現する技術としては，W3C(World Wide Web Consortium)が仕様を策定している分散型識別子(DIDs, Decentralized Identifiers)[6]がある．DIDs では，分散台帳などにユーザの ID を登録し，そのユーザの個人情報へのアクセスをユーザがコントロールすることで，インターネット上での ID の管理と個人情報の流通をユーザの権限に委ねている．その他の技術では，Linux Foundation が提唱している Hyperledger Indy[7]がある．Hyperledger Indy では，ユーザやサービス事業者，およびユーザの卒業証明書や免許証などの公的身分証明書を発行する機関などが，2 者間でのみ用いられる pseudonym DID(pseudonym Decentralized Identifier)を 2 者それぞれが生成する．この pseudonym DID は分散台帳に登録されて，pseudonym DID を生成したユーザなどの主体に，ID と個人情報の流通の管理が委ねられている．

DIDs や Hyperledger Indy では，ユーザの ID を分散台帳に登録することで，分散台帳に参加しているノードによって分散して ID を保持することから，特定のサービス事業者による ID の管理を避けて，ユーザ自身で ID を管理することができる．また，ユーザの ID が特定のサービス事業者によって管理されていないことから，複数のサービス事業者のサービスをそれぞれ同一の ID で利用しても，各サービスの利用履歴などを含めた個人情報が特定のサービス事業者に収集，把握されることを回避したり，制御したりすることができる．しかし，ユーザが提供した個人情報がサービス事業者や提携先にどのように流通しているかについては，ユーザは直接自身の個人情報を提供したサービス事業者については把握することができるものの，サービス事業から他のサービス事業者や提携先(以下，2 次流通先と呼ぶ)への流通については，把握することが困難である．

また，サービスを提供するサービス事業者の観点では，取得したユーザの個人情報は引っ越しや携帯電話の追加変更などにより，サービス提供中に変更されることがある．しかし，上述した SSI を実現する既存の技術では，サービス事業者や 2 次流通先がユーザの個人情報が変更されたことを把握するのが困難なため，サービス提供や個人情報の活用に影響が発生することが考えられる．

この様に，SSI のコンセプトを実現する従来の技術は，ユーザの ID や個人情報の管理をユーザ自身に委ねることができるものの，ユーザの個人情報の流通に関わるプライバシー保護，およびサービス事業者や 2 次流通先の運用性に関しては，課題があると考えられる．これらの課題については，インターネット上でサービスを提供したり，利用したりする際の従来からの課題であり，これらの課題を解決することができれば，ユーザやサービス事業者，2 次流通先にとって，SSI を利用するメリットが高まると考えられる．

本稿では，以上の背景を踏まえて，SSI を実現するための技術として，ユーザのプライバシー保護，およびサービス事業者や 2 次流通先の運用性を向上させるための個人情報の流通方式について提案，評価を行う．

本稿の構成について述べる．2 節では，SSI を実現する従来の技術について説明する．3 節では，2 節で説明した技術について，ユーザの個人情報の流通の観点での課題について説明する．4 節では，3 節で説明した課題を解決する個人情報の流通方式について提案する．5 節では，提案方式の評価行う．最後に，6 節で本稿のまとめを述べる．

プライバシーポリシー
・・・
・個人情報の提供
　1．お客様の同意をいただいた場合には，当社は，委託先等の第三者に個人情報を提供します．
　2．前項の他にも，当社は，グループ企業間で，必要な範囲で個人情報を共有します．
・・・

図1 プライバシーポリシーにおける個人情報の提供の規定例

## 2. 従来の技術

本節では，SSI を実現する従来の技術として，DIDs および HyperLedger Indy の概要について，ユーザの ID や個人情報の管理，流通方法を中心に説明する．これらの技術は，ユーザの個人情報を紐づける ID が，特定の IDP に依存せずに分散管理されることを特徴とする．

### 2.1 DIDs

DIDs は，W3C によって策定された自己主権型のアイデンティティ管理技術で，DID(Decetralized Identifier)によって，ユーザなどの主体を特定する．DIDs の概要について．図 2 を用いて説明する．

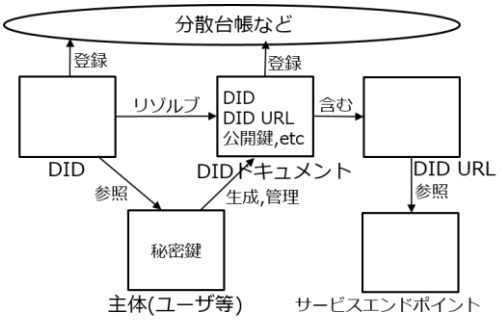


図2 DIDsの全体構成(概要)

図 2 で，ユーザの端末は，自身の識別子(DID)と DID に紐づく公開鍵/秘密鍵のペアを生成する．秘密鍵については，公開鍵/秘密鍵のペアを生成した端末のセキュア領域に保管する．DID ドキュメントは，ユーザの DID に関連する情報を含むドキュメントで，DID の他に，ユーザの端末が生成した公開鍵や，ユーザがインターネット上で提供する個人情報などの位置を示すサービスエンドポイント(SEP)の DID URL などが記述されている．DID ドキュメントは DIDs

を構成する分散台帳などに登録され，DID リゾルバに DID を入力することで分散台帳などから DID ドキュメントを取得することができる．DID や DID ドキュメントの生成，更新，非アクティブ化，DID から DID ドキュメントへのリゾルブの実装方法については， DID メソッドとして定義される．

次に，図 3 を用いて，DIDs を用いた場合に想定される，インターネット上のサービスへのユーザ登録例を示す[8]．まず，事前準備として，ユーザの端末は DID と公開鍵/秘密鍵を生成し，その後に DID ドキュメントを作成して分散台帳に登録する(図 3 の①)．また，自身の個人情報を記載したドキュメントを作成し，認定機関の署名を付けて，ストレージなどの SEP に保管する(同②,③)．その後，ユーザがインターネット上のサービスにアクセスしてユーザ登録を行う場合，ユーザはサービス事業者に DID を提示し(同④)，サービス事業者は DID メソッドを用いて，DID リゾルバを経由して，DID で特定される DID ドキュメントを分散台帳から取得する(同⑤,⑥)．そして，サービス事業者は DID ドキュメントから公開鍵を取得して，ユーザが保持する秘密鍵とともにチャレンジレスポンスを行うことにより，取得した DID ドキュメントの所有者は DID を提示したユーザであることを検証する(同⑦)．その後，サービス事業者は DID ドキュメントに記載されている SEP にアクセスしてユーザの個人情報を取得し，ユーザ登録を行う(同⑧)．なお，SEP がサービス事業者からのアクセスを許可するために，ユーザがあらかじめ，SEP にサービス事業者のアクセス権限を設定する等のアクセス制御を行う．

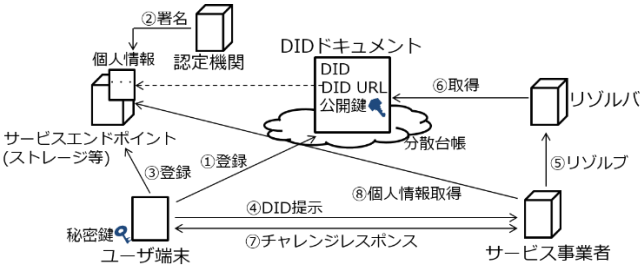

図3 DIDsを用いた場合のユーザ登録例

### 2.2 Hyperledger Indy

Hyperledger Indy は，Linux Foundation が提唱している分散型 ID 管理技術で，汎用サーバで動作するソフトウェアツールやライブラリの提供も Linux Foundation によって行われている．Hyperledger Indy の概要を図 4 に示すケーススタディによって説明する．

図 4 のケーススタディでは，就職活動をしているユーザが，オンラインで求人募集をしている会社に応募するにあたって，自身の個人情報である，電子的な卒業証明書(以下，卒業証明書と呼ぶ)を会社に提出するために Hyperledger Indy を用いている．Hyperledger Indy には，ユーザ，会社，大学が参加し，Hyperledger が提供する分散台帳に接続している．ここで，大学は信頼できるノードとして分散台帳に接続し，Hyperledger 内で公開される自身の分散型 ID (以下，verynim DID と呼ぶ)，verynimDID に紐づく公開鍵，および卒業証明書の形式を定義した情報(以下，クレデンシャル定義と呼ぶ)を分散台帳に登録している(図 4 の①)．なお，公開鍵とペアになる秘密鍵は大学のノードのセキュア領域に保管されている，まず，ユーザは，会社に提出する卒業証明書を大学から取得する．そのために，ユーザは，分散台帳からクレデンシャル定義を取得し(同②)，大学に卒業証明書を要求する．大学はユーザの卒業証明書を作成した後に，verynimDID に紐づく秘密鍵で署名をして(同③)，ユーザに署名付き卒業証明書を返信する(同④)．ユーザは，署名付き卒業証明書を会社に提出すると(同⑤)，会社は verynimDID に紐づく公開鍵を分散台帳から取得して卒業証明書を検証して，真正性を確認する(同⑥,⑦)．なお，ユーザ，大学，会社間の各通信では，それぞれ相互認証を可能とするために，各 2 者間でのみ用いられる分散型 ID(以下，pseudonym DID と呼ぶ)，および pseudonym DID に紐づく公開鍵/秘密鍵のペアを 2 者それぞれが生成し，pseudonym DID と公開鍵を分散台帳に登録する．秘密鍵については，端末あるいはサーバのセキュア領域にそれぞれ保管する．相互認証の際には，2 者間で pseudonym DID を相互に提示し，相手の pseudonym DID に紐づく公開鍵を分散台帳から取得して，チャレンジレスポンスで認証を行う．

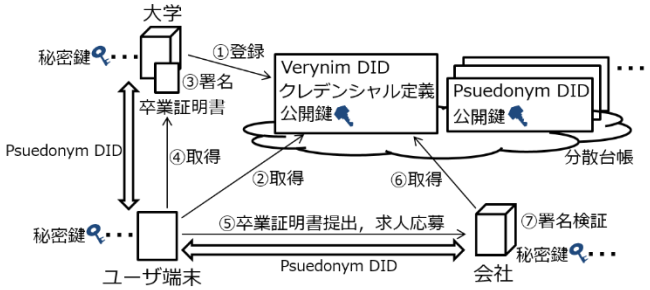

図4 Hyperledger Indyを用いた場合のケーススタディ

## 3. 課題

2 節で説明したように， SSI を実現する従来の技術をインターネット上でのサービスに用いた場合，ユーザなどの ID を DID として分散台帳で管理することで，特定のサービス事業者による中央集権的な ID 管理を回避することができる．また，複数のサービス事業者のサービスをそれぞれ同じ DID を用いて利用した場合でも，DID に紐づくユーザの個人情報が，各サービス事業者間で名寄せされることを抑止し，一定範囲のプライバシーを保護することが可能であると考えられる．

しかし，これらの技術では，ユーザの ID に紐づく個人情報がサービス事業者に提供された後，サービス事業者から 2 次流通先への個人情報の流通をユーザが管理，制御する仕組みはない．そのため，ユーザが提供した個人情報のうち，具体的にどの個人情報がどの 2 次流通先に流通しているのかをユーザが把握するのは困難である．

また，2 次流通先において，サービス事業者が取得したユーザの個人情報を共有する仕組みはないことから，別途，ユーザの個人情報を適切な範囲でサービス事業者と共有するための方法が必要となる．

さらに，サービス事業者が取得，共有した個人情報に関して，ユーザが住所や連絡先等を変更した場合，DIDs では，最新の個人情報は SEP から取得することになる．そのため，サービス事業者は適宜，SEP にアクセスして取得済の個人情報が更新されていないかを確認する必要がある．しかし，SEP では，セキュリティ上の観点から，サービス事業者のアクセス権限に期限を設定して運用することが考えられる．Hyperleder Indy では，ユーザの個人情報は，ユーザの端末で保持するため，サービス事業者から適宜ユーザの端末にアクセスして，個人情報が更新されていないかを確認するのは，ユーザの端末のセキュリティや処理負荷の観点で困難であると考えられる．

本稿では，以上の問題点を踏まえて，SSI における個人情報の流通に関して次の課題を設定する．

[ユーザのプライバシーの保護の観点]

(T-1) ユーザがサービス事業者に提供した個人情報について，ユーザ自身が 2 次流通先への流通の管理，制御ができること．

[サービス事業者と 2 次流通先の運用性の観点]

(T-2) 2 次流通先が利便性よく個人情報を取得できること．

(T-3) サービス事業者と 2 次流通先は，取得した個人情報を適宜最新化できること．

## 4. 提案方式

本節では，3 節で設定した 3 つの課題(T-1)〜(T-3)を解決するための個人情報の流通方式について説明する．

### 4.1 課題解決の前提

ユーザがインターネット上のサービスを利用するにあたり，ユーザの ID はユーザ自身で管理する SSI のコンセプトを用いることを前提に，W3C が策定した DIDs を拡張する．ここで，サービス事業者と 2 次流通先は，提案方式が用いる分散台帳に，信頼できるノードとして接続する．また，サービス事業者と 2 次流通先は，それぞれ DID および DID に紐づく公開鍵/秘密鍵のペアを生成する．サービス事業者は，サービス事業者が取得したユーザの個人情報の，全ての 2 次流通先の DID を知っているものとする．

### 4.2 課題解決へのアプローチ

課題(T-1)については，DIDs が定義している，DID の主体に関連する情報を含む DID ドキュメントの他に，サービス事業者や 2 次流通先におけるユーザの個人情報の扱いに関連する情報を含むサービスドキュメントを新たに定義する．このサービスドキュメントの生成，更新，非アクティブ化，リゾルブは，DID ドキュメントと同様に，DID メソッドを用いて行う．サービスドキュメントには，図 5 に示す通り，サービス事業者や 2 次流通先が取得するユーザの個人情報の項目や，その 2 次流通先を記載する．また，DID ドキュメントと同様に，サービス事業者や 2 次流通先の DID や，それらが生成した公開鍵/秘密鍵のペアのうちの公開鍵を記載する．ユーザは，このサービスドキュメントの内容を確認することで，自身の個人情報の流通先を把握する．

また，ユーザの個人情報の保管や，個人情報へのアクセス制御を行うユーザエージェントを新たに導入する．ユーザエージェントの機能概要を図 6 に示す．図 6 で，ユーザエージェントは，複数のユーザの個人情報の保管，アクセス制御を可能とし，ユーザ毎の公開鍵/秘密鍵ペアの生成，分散台帳上での DID の処理等，ストレージ等のサービスエンドポイントにはない処理を行う．ユーザエージェントでの，ユーザの個人情報へのアクセス制御については，ユーザがサービス事業者や 2 次流通先に，自身の個人情報へのアクセス権限を移譲することで行う．

課題(T-2)については，2 次流通先も，サービス事業者と同様の手続きで，ユーザの個人情報を取得可能とする．そのために，サービス事業者の DID に 2 次流通先の DID を紐づけて，ユーザがこれらの DID の主体に自身の個人情報へのアクセス権限をまとめて移譲する．

課題(T-3)については，サービス事業者や 2 次流通先はユーザエージェントにアクセスして，適宜，取得済みのユーザの個人情報が更新されていないかを確認することで，ユーザの端末での個人情報の提供に関する処理を不要とする．また，ユーザエージェントが，サービス事業者や 2 次流通先毎に，ユーザの個人情報へのアクセス権限に期限を設定し，適宜アクセス権限を更新する．

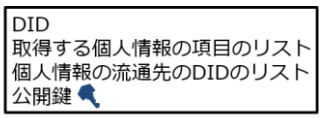

図5 サービスドキュメントの構成概要

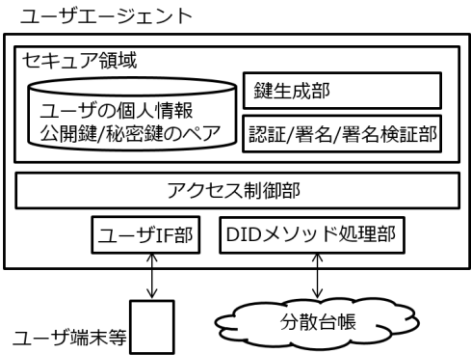

図6 ユーザエージェントの構成概要

### 4.3 提案方式の処理

4.2 節を踏まえて，ユーザがサービス事業者のサービスにユーザ登録する場合を例にして，図 7 を用いて，提案方式の処理について説明する．なお，本節で用いる記号については，表 1 に一覧として記載する．

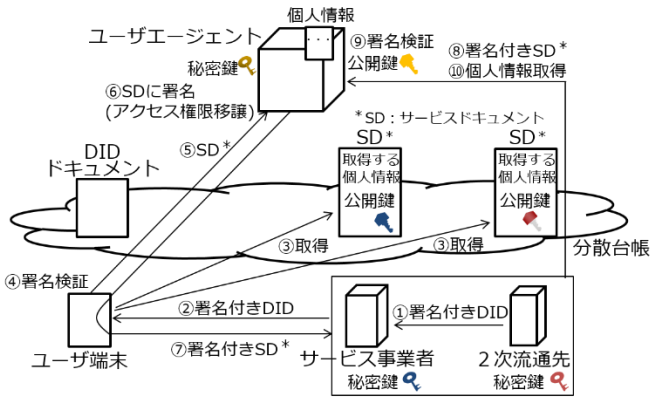


図7 提案方式の処理

表1 記号の説明

| 記号 | 説明 |
|---|---|
| DIDs | サービス事業者に紐づくDID |
| DIDc | 2次流通先に紐づくDID |
| SDs | サービス事業者のサービスドキュメント |
| SDc | 2次流通先のサービスドキュメント |
| Kss | サービス事業者が生成した秘密鍵 |
| Kps | サービス事業者が生成した公開鍵 |
| Ksc | 2次流通先が生成した秘密鍵 |
| Kpc | 2次流通先が生成した公開鍵 |
| Ksu | ユーザエージェントが生成した，ユーザに紐づく秘密鍵 |
| Kpu | ユーザエージェントが生成した，ユーザに紐づく公開鍵 |
| X+Y | データXとデータYを結合したデータ |
| M(X, Ks) | Xに，秘密鍵KsによるXへの署名を付加したデータ |

#### 4.3.1 DID の生成と個人情報の登録

サービス事業者と 2 次流通先は，あらかじめ，自身の DID と DID に紐づく公開鍵/秘密鍵のペアを生成した後に，サービスドキュメントを生成して，分散台帳に登録する．サービスドキュメントには，4.2 節で述べた通り，自身の DID，DID に紐づく公開鍵，取得するユーザの個人情報，2 次流通先の DID と流通させる個人情報の項目などを記載する．

ユーザは，ユーザエージェントにアカウントを作成し，自身の個人情報をユーザエージェントに保管する．このとき，ユーザエージェントは，ユーザに紐づく公開鍵/秘密鍵のペアを生成し，ユーザの個人情報を公開鍵で暗号化する．

ユーザエージェントは，アカウントを作成したユーザの DID と DID に紐づく公開鍵/秘密鍵のペアを生成した後に，ユーザの DID ドキュメントを生成して，分散台帳に登録する．

#### 4.3.2 ユーザの個人情報へのアクセス権限の移譲

ユーザがサービス事業者にユーザ登録を要求すると，サービス事業者は 2 次流通先の DIDc を用いて，DID リゾルバを経由して，分散台帳から 2 次流通先のサービスドキュメント SDc を取得する．また，サービス事業者は，2 次流通先から，DIDc に自身が生成した秘密鍵 Ksc で署名をした M(DIDc, Ksc)を取得する(図 7 の①)．

サービス事業者は，分散台帳から取得した SDc に記載されている 2 次流通先の公開鍵 Kpc 用いて M(DIDc, Ksc)の署名を検証して，DIDc は 2 次流通先と紐づいていて，2 次流通先がなりすましていないことを確認する．その後に，サービス事業者は，DIDs と M(DIDc, Ksc)に，自身が生成した秘密鍵 Kss で署名をした M(DIDs+M(DIDc, Ksc), Kss)をユーザに提示して，ユーザの個人情報へのアクセス権限の移譲を要求する(同②)．

ユーザは，M(DIDs+M(DIDc, Ksc), Kss)から DIDs を取得し，DID リゾルバからサービス事業者と 2 次流通先のサービスドキュメント SDs と SDc を取得する(同③)．そして，SDs に記載されているサービス事業者の公開鍵 Kps を用いて M(DIDs+M(DIDc, Ksc), Kss)の署名を検証して，このメッセージはサービス事業者が作成したことを確認する(同④)．確認後，ユーザは，SDs から，サービス事業者が取得するユーザの個人情報と 2 次流通先を確認し，これらのサービス事業者と 2 次流通先に，自身の個人情報へのアクセス権限を移譲することに同意する場合には，ユーザエージェントに，M(DIDs+M(DIDc, Ksc), Kss)とともに SDs と SDc を送付し，サービス事業者と 2 次流通先のサービスドキュメントへの署名を依頼することで，アクセス権限を移譲する(同⑤)．

ユーザエージェントは，M(DIDs+M(DIDc, Ksc), Kss)から DIDs と DIDc を取得し，SDs に記載されているサービス事業者の公開鍵 Kps を用いて M(DIDs+M(DIDc, Ksc), Kss)の署名を検証して，このメッセージはサービス事業者が作成したことを確認する．確認後，ユーザエージェントは，ユーザがアカウント作成時に生成した秘密鍵 Ksu を用いて，SDs と SDc にそれぞれ署名を行って，M(SDs, Ksu)と M(SDc, Ksu)をそれぞれユーザに返信する(同⑥)．ユーザは，M(SDs, Ksu)と M(SDc, Ksu)をサービス事業者に送信する(同⑦)．サービス事業者は，M(SDc, Ksu)を 2 次流通先に送信する．

なお，ユーザエージェントが SDs と SDc に署名をした際に，ユーザ毎に SDs と SDc を関連付けて保管することで，ユーザの個人情報の流通元と流通先を把握することが可能となる．

#### 4.3.3 ユーザの個人情報の取得

ユーザの個人情報へのアクセス権限を移譲されたサービス事業者は，署名付きサービスドキュメント M(SDs, Ksu)をアクセストークンとして，ユーザエージェントから個人情報を取得する．サービス事業者は，M(SDs, Ksu)をユーザエージェントに送信して，個人情報を要求する(図 7 の⑧)．ユーザエージェントは，受信した M(SDs, Ksu)から SDs を取得し，SDs に記載されている公開鍵 Kps を用いて，チャレンジレスポンスにより，サービス事業者を認証して，サービス事業者と SDs の紐づけを確認する，確認後，ユーザエージェントは，M(SDs, Ksu)から Ksu を取得し，Ksu とペアとなる公開鍵 Kpu を用いて，M(SDs, Ksu)の署名を検証して，ユーザエージェントが署名したメッセージであるこ

とを確認する(同⑨). その後, ユーザエージェントは SDs に記載されているユーザの個人情報をサービス事業者に送信する(同⑩).

2 次流通先がユーザの個人情報を取得する場合も同様である．2 次流通先は，M(SDc, Ksu)をユーザエージェントに送信して，ユーザの個人情報を要求する．ユーザエージェントは，SDc に記載されている 2 次流通先の公開鍵 Kpc を用いて 2 次流通先を認証する．次に，M(SDc, Ksu)の署名を検証した後に，SDc に記載されているユーザの個人情報を 2 次流通先に送信する．

なお，署名付きサービスドキュメントをアクセストークンとした，ユーザの個人情報へのアクセス権限に，期限を設定することが可能である．一例として，ユーザエージェントがユーザに紐づく公開鍵 Kpu と秘密鍵 Ksu を定期的に更新し，サービス事業者，2 次流通先とサービスエージェント間で，サービスドキュメントへの署名を期限前に更新する．署名を更新することで，ユーザエージェントは，同一アクセストークンによる，ユーザの個人情報への無期限のアクセスを回避することができる．

以上のように，提案方式では，ユーザは，サービス事業者と 2 次流通先の署名付き DID とサービスドキュメントを取得して，自身の個人情報に対するサービス事業者と 2 次流通先のアクセス権限を制御する．また，サービス事業者と 2 次流通先は，ユーザエージェントが署名をしたそれぞれのサービスドキュメントをアクセストークンとして用いて，それぞれ同様の処理で，ユーザエージェントからユーザの個人情報を適宜取得，最新化する．これらの処理により，3 節で設定した 3 つの課題(T-1)〜(T-3)を解決することができる．

## 5. 提案方式の評価

4 節で述べた提案方式では，ユーザエージェントを新たに導入し，ユーザエージェントがサービス事業者や 2 次流通先にユーザの個人情報へのアクセス権限の移譲を行う．ユーザエージェントでは，複数のユーザの個人情報を保管して，それらの個人情報へのアクセス権限の移譲を行うことから，ユーザエージェントに提案方式の処理負荷が集中すると考えられる．また，ユーザエージェントでは，ユーザの個人情報へのアクセス制御のために，電子署名に関する署名や分散台帳へのアクセス処理を行い，これらの処理は個人情報の流通の処理負荷の要因になることが想定される．そこで，本節では，ユーザエージェントを新規に導入することによる設備の増加に着目し，ユーザエージェントを対象に処理性能に関する基本測定をして，評価を行う．なお，ユーザエージェントでは，分散台帳からサービスドキュメントを取得していることから, 今回の基本測定では，分散台帳の処理能力も含まれる．

### 5.1 システム構成

本稿では，図 8 に示す通り，汎用機で動作するソフトウェアツールやライブラリを提供している Hyperledger Indy のソフトウェアを流用して，提案方式を実装した．実装範囲は, 4.3 節で説明した提案方式の処理を含むこととする．ここで，DID ドキュメントやサービスドキュメントに記載する DID には，Hyperledger Indy の verinym DID を用いた．また，ユーザ，ユーザエージェント，サービス事業者，および 2 次流通先の間の通信に用いる DID には psuedonym DID を用いた．これらの DID を用いたのは，Hyperledger Indy での DID の実装を流用するためである．なお, 分散台帳には, Hyperledger Indy が提供する独自の分散台帳を用いた．

ユーザ端末からの処理要求は, Apache が提供する性能測定ツール(JMeter)を使用した．また，ユーザ端末からは，マルチスレッドにより同時に複数の処理を要求し，スレッド数を増やすことで処理負荷を増やした．

ハードウェアは，汎用機を用いて，ユーザ，ユーザエージェント，サービス事業者をそれぞれ異なる汎用機に 1 つずつ実装した．分散台帳はノード数を 4 とし，1 つの汎用機に 4 ノードを実装した．表 2 に汎用機の諸元を示す．なお，ユーザ，ユーザエージェント，サービス事業者，および分散台帳を実装した汎用サーバ機は，それぞれ同一の諸元である．

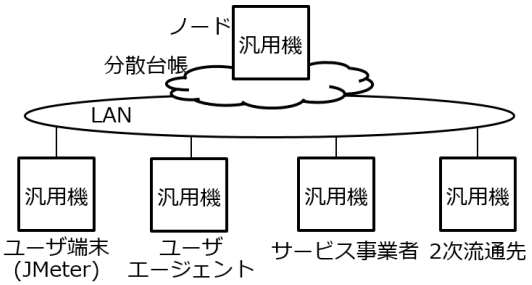


図8 性能評価のシステム構成

表2 性能評価に用いた汎用機の諸元

| 項目 | 諸元 |
|---|---|
| CPU | Intel(R) Core(TM) i9-9900(3.1GHz)<br>8Cores/16Threads<br>Cache Memory:16MB |
| メモリ | 16GB DDR4 SDRAM(2666MT/s 2DIMM) |
| OS | Ubuntu 16.04 LTS |

### 5.2 評価項目

ユーザエージェントの処理性能として，ユーザがサービス事業者のサービスにユーザ登録をする際などの個人情報の提供を想定し，個人情報へのアクセス権限の移譲時および，サービス事業者によるユーザの個人情報の取得時の処理を対象にする．この処理性能の項目として，スループットを測定する．その測定結果から，ユーザエージェントが収容できるユーザ数を評価する．

### 5.3 評価結果

図 9 に，JMeter のスレッド数と，ユーザエージェントでのスループットの関係を示す．図 9 より，本稿で用いた汎用サーバ機の諸元の場合，ユーザエージェントが 1 秒間で処理できるユーザ数は，個人情報へのアクセス権限の移譲では約 9 ユーザ，ユーザの個人情報の取得では約 15 ユーザの処理が可能であった．

以上の測定結果を踏まえて，ユーザエージェントが収容できるユーザ数を評価する．ユーザエージェントの諸元は，本稿で用いた汎用サーバ機の諸元と同等し，1 秒間で処理できるユーザ数を，個人情報へのアクセス権限の移譲の処理数とユーザの個人情報の取得の処理数の調和平均の 1/2 とし，今回の測定値から 5.625 ユーザとする．なお，ユーザがインターネットを利用する時間帯を考慮して，ユーザ登録は，インターネットの利用率が数%から 15%程度に立ち上がる 7 時台から，再度 15%程度から数%に落ち込むまでの 23 時台に集中すると仮定して評価する[9].

図 10 に，1 ユーザが個人情報を提供する頻度と，1 ユーザエージェントが収容可能なユーザ数の関係を示す．図 10 より，1 ユーザが 1 か月に 1 度程度，個人情報を提供する場合であれば，数台～10 台程度のユーザエージェントで，日本国内のユーザ(1 億人程度)を収容することができると考えられる．なお，キャパシティプランニングの観点では，ユーザエージェントは，ユーザ毎に異なるユーザエージェントに収容可能であり，ユーザ数に比例したスケーラビリティがあると考えられる．

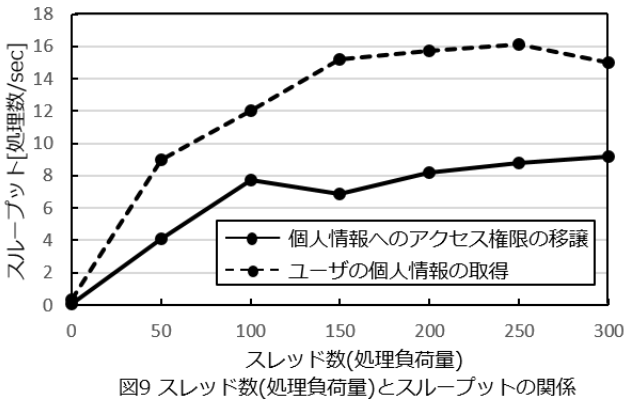

図9 スレッド数(処理負荷量)とスループットの関係

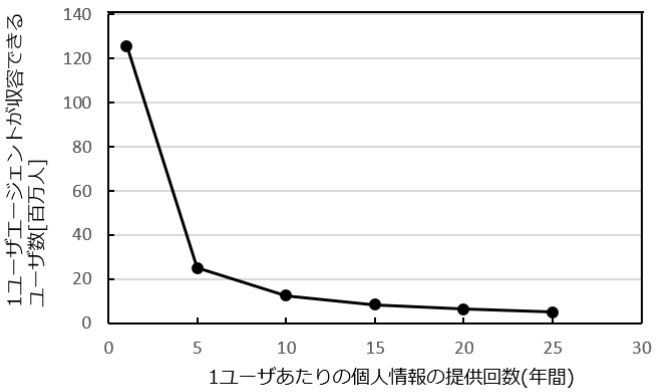

図10 個人情報を提供する頻度とユーザエージェントの収容ユーザ数の関係

## 6. まとめ

本稿では，SSI を実現する技術として，ユーザがインターネット上のサービスを利用するにあたってサービス事業者に提供する個人情報の流通について，ユーザのプライバシー保護，およびサービス事業者や 2 次流通先の運用性を向上させる方式を提案した．提案方式では，ユーザの個人情報を管理し，ユーザが移譲した個人情報のアクセス権限にもとづいてアクセス制御を行うユーザエージェントと，個人情報の流通を把握するためのサービスドキュメントを新たに導入して，ユーザが 2 次流通先への個人情報の流通の管理，制御を可能とするとともに，2 次流通先が利便性よく個人情報を取得することを可能とした．また，サービス事業者と 2 次流通先は，取得した個人情報を適宜最新化することも可能とした．

提案方式の評価では，汎用機を用いた実装を行い，ユーザが 1 か月に 1 度程度，個人情報を提供する場合であれば，日本国内のユーザを数台～10 台程度のユーザエージェントで収容できることが分かった．

今後は，ユーザエージェントの実装方法について詳細検討を進めていく予定である．

## 参考文献

[1] OpenID Foundation：OpenID Coonect 1.0 (2014).
[2] OASIS：Assertions and Protocols for the OASIS Security Assetion Markup Language(SAML)V2.0 (2005).
[3] 経済産業省：FinTech ビジョン(FinTech の課題と今後の方向性に関する検討会合 報告)，経済産業省(オンライン)，https://www.meti.go.jp/report/whitepaper/data/pdf/20170508001_1.pdf（参照 2021-04-22）.
[4] 矢野経済研究所：生命保険領域における国内 InsurTech 市場に関する調査を実施（2018 年），矢野経済研究所(オンライン)，https://www.yano.co.jp/press-release/show/press_id/1972（参照 2021-04-22）.
[5] 山本龍彦：信用スコアの課題と今後，月間 経団連(オンライン)，https://www.keidanren.or.jp/journal/monthly/2019/10/p26.pdf（参照 2021-04-22）.
[6] World Wide Web Consortium：Decentralized Identifiers(DIDs) v1.0 (2021).
[7] Linux Foundation：HYPERLEDGER INDY，Linux Foundation，https://www.hyperledger.org/use/hyperledger-indy (accessed 2021-04-22）.
[8] 大森芳彦，山下高生：自己主権型アイデンティティにおけるユーザのクレデンシャル管理方法の検討，電子情報通信学会ソサイエティ大会，B-7-6，(2019）.
[9] 総務省：令和 2 年版 情報通信白書，総務省(オンライン)，https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r02/pdf/index.html（参照 2021-04-22）.