Vol.2011-IS-115 No.11
2011/3/15

シングルサインオンにおけるID継続手法の考察

柿崎 淑郎[†1] 前田 千徳[†1] 岩村 恵市[†1]

シングルサインオンは1つのクレデンシャルで複数のWebサービスを利用可能とするID管理技術である．利便性が高い反面，認証サーバが利用できなくなった場合，Webサービスが正常稼働していても利用できなくなる．そのため，シングルサインオンにおいては，継続的に利用し続けられることが重要となる．本稿ではシングルサインオンにおけるID継続手法について考察する．ID継続手法として，冗長化SSO認証サーバ方式，別名方式，複数SSOID方式，SSOID移行方式の4つを挙げ，説明する．これら4方式を継続性，安全性，利便性の側面から考察する．またそれらの実現可能性について議論を行う．

A Consideration of Identity Continuance in Single Sign-On

KAKIZAKI Yoshio ,[†1] MAEDA Kazunari [†1]
and IWAMURA Keiichi[†1]

Single sign-on is an identity management technique that provides users the ability to use multiple Web services with one set of credentials. However, when the authentication server is down or unavailable, users cannot access Web services, even if the services are operating normally. Therefore, enabling continuous use is important in single sign-on. In this paper, we present an identity continuance method for single sign-on. We explain four such continuance methods: Redundant SSO Authentication Server method; Alias SSOID method; Multiple SSOID method; and SSOID Migration method. We consider these four methods from the viewpoint of continuity, security and efficiency. Moreover, we discuss the feasibility of each.

†1 東京理科大学
Tokyo University of Science

1. はじめに

我々がサービスを利用するとき，個別化された情報を利用するために，ログインなどのユーザ認証が必要となることが多い．複数のサービスを利用するとき，ユーザは各サービスで利用するIDとパスワードを記憶しておく必要がある．しかし安全性の観点から，複数のサービス提供者に対して，同一のIDとパスワードを設定することは好ましくない．そのため，複数のサービスを利用しようとするユーザは，多くのIDとパスワード対を記憶しておく必要があり，利便性が低い．

シングルサインオンとは，1度の認証によって，複数のアプリケーションやサービスを利用することが可能になる認証技術である．1度認証されることによって，さらなる認証を必要としないため，ワンストップ認証と呼ばれることもある．シングルサインオンでは，ユーザは認証サーバによって1度だけ認証を行う．その結果をユーザはサービス提供者に提示し，サービスを利用する．この際，複数の異なるサービス提供者を利用する場合でも，認証サーバによって認証されているユーザであれば，さらなる認証を必要とせずに，サービスを利用することができる．そのため，サービス提供者ごとに認証を行う方式に比べ，ユーザが記憶しておく必要があるIDとパスワードは大幅に減り，利便性が大きく向上する．

しかし一方で，認証サーバがなんらかの理由で停止した場合，ユーザは認証されることができずに，複数のサービスを利用することができなくなる問題がある．シングルサインオンを用いずに，各サービス提供者ごとで認証を行う方式では，認証が行えない場合の影響は，高々そのサービス提供者にしか及ばない．シングルサインオンの場合は，シングルサインオンが利用できる全てのサービスに影響が及ぶ．特に，サービス提供者が正常稼働していても，認証サーバが停止するだけで，ユーザはシングルサインオンを使えなくなる．そのため，認証サーバが停止した後においても，継続的にサービスを利用できるようにする方法が必要となる．

シングルサインオンにおいて，認証サーバが停止後にも，継続的にサービスを利用できるための要件を以下に挙げる．

要件1 問題発生後も継続してサービスを利用し続けられる（継続性）
要件2 第3者になりすましや乗っ取りが行われない（安全性）
要件3 ユーザの負担が少ない（利便性）

要件1は継続性であり，問題発生後においても，ユーザは継続してサービスを利用し続けられることが必要である．要件2は安全性であり，問題発生時に乗じて悪意ある第3者が

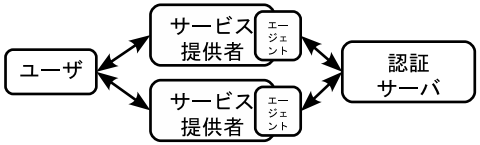

図 1 エージェント型シングルサインオン
Fig. 1 An agent type single sign-on.

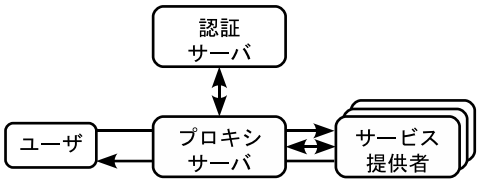

図 2 リバースプロキシ型シングルサインオン
Fig. 2 A reverse-proxy type single sign-on.

ユーザになりすましたり，乗っ取りを行ったりされないことが必要になる．要件 3 は利便性であり，問題発生時またはそれ以前からサービス利用再開までにかかるユーザの負担が少ないことが求められる．

本稿では継続的にサービスを利用できるように，シングルサインオンにおける ID 継続手法として，冗長化 SSO 認証サーバ方式，別名方式，複数 SSOID 方式，SSOID 移行方式の 4 つを挙げ，それぞれを比較検討する．これら 4 方式が上記 3 要件を充足するかを考察し，またその実現可能性について議論を行う．

2. シングルサインオン

シングルサインオンは大別するとエージェント型とリバースプロキシ型がある[1]．エージェント型は図 1 に示すような構成である．ユーザがサービスを利用しようとする場合，サービス提供者内部のエージェントがユーザと認証サーバの双方と通信し，認証結果などを受け取ることで，シングルサインオンが実現される．リバースプロキシ型は図 2 に示すような構成である．リバースプロキシ型ではプロキシサーバがユーザとサービス提供者の間に介在する．ユーザはプロキシサーバにログインすることで，プロキシサーバがユーザに代わってサービス提供者との認証を行う．

2.1 OpenID

OpenID[2),3)] は URI（Uniform Resource Identifier）または XRI（Extensible Resource Identifier）を識別子とするユーザ中心の分散認証サービスである．OpenID では以下の 3 つのエンティティを用いる．

OP OpenID Provider はユーザが主張する identifier を認証するエンティティ．

RP Relying Party はユーザを認証するために OP に認証を依頼するエンティティ．

ユーザ ユーザは自身の identifier を主張し OP より認証される．

ユーザ認証を必要とする場面では，認証者は被認証者を認証するために，例えば ID とパスワードの対などが必要となる．このとき，ユーザは自らの利便性のために，異なる認証者に対して同じ ID とパスワードの対を利用しているとする．この場合，悪意ある認証者が自らが持つユーザの認証情報を使って，被認証者になりすまして他の認証者を欺くことが可能となる．OpenID では，各 RP はユーザを認証するための認証情報は持たず，OP のみが認証情報を持っている．そのため，RP はユーザを認証することができないので，OP に対してユーザ認証を依頼し，RP はその認証結果を用いる．この際，ユーザは 1 組の ID とパスワードのみを用いており，OpenID の仕組みを用いる RP に対して，シングルサインオンが実現できる．

3. 準　　備

本稿では変形したエージェント型のシングルサインオンを抽象化し，その上での考察・検討を行う．

3.1 用　　語

本稿で用いる用語を以下に示す．

SSO 認証サーバ

SSO 認証サーバはシングルサインオン環境下において，ユーザを認証する主体である．SSO 認証サーバはユーザに対して SSOID を発行する．SSO 認証サーバは未認証 SSOID を主張するユーザについて，対応するパスワードなどの認証情報を用いて認証する．また，その結果を認証済 SSOID として，ユーザに発行する．

SSO クライアント

SSO クライアントはシングルサインオン環境下において，ユーザにサービスを提供する主体である．SSO クライアントは未認証 SSOID を主張するユーザの認証情報を持たないため，SSO 認証サーバに対して，ユーザの認証を要求する．SSO クライアントは SSO 認証

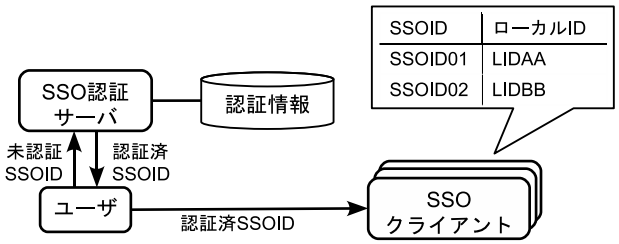

図 3 抽象化したシングルサインオン
Fig. 3 General model of single sign-on.

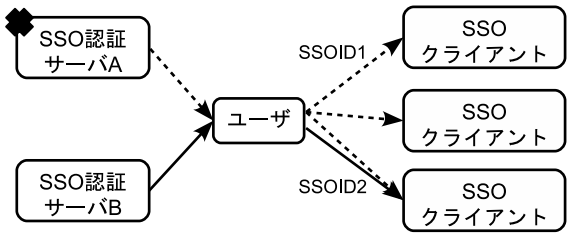

図 4 取り扱う問題の概略図
Fig. 4 Overview of problem.

サーバから認証結果である認証済 SSOID を受け取り，ユーザに対してサービスを提供する．SSO クライアントはユーザとその情報を管理するために，ローカル ID を認証済 SSOID と紐付けて管理する．なお，SSO クライアントはエージェントの機能を内包している．

ユ ー ザ

ユーザはシングルサインオン環境下において，SSO クライアントが提供するサービスを利用しようとする主体である．ユーザは SSO クライアントでのサービスを利用するにあたって，SSO 認証サーバにおいて認証を受ける必要がある．また，SSO 認証サーバより認証された場合，シングルサインオンによって異なる複数の SSO クライアントを利用することができる．

SSOID

SSOID はシングルサインオン環境下において，各ユーザに一意に与えられる識別子である．SSO 認証サーバと SSO クライアントは，SSOID によって，ユーザの識別を行う．SSOID は永続的に用いる一意の識別子であり，異なる SSOID は異なるユーザを意味する．また特に，SSO 認証サーバから認証を受けていない場合は未認証 SSOID，SSO 認証サーバから認証されている場合は認証済 SSOID と呼ぶ．

ローカル ID

ローカル ID は各 SSO クライアントで利用される識別子であり，各 SSO クライアントのみで有効であり，他の SSO クライアントでは利用できない．ローカル ID は SSO クライアントにおいて，ユーザとその情報を管理するために，認証済 SSOID に紐付ける．

3.2 抽象化したシングルサインオン

抽象化したシングルサインオンの構成図を図 3 に示す．

ユーザは SSO 認証サーバから SSOID を発行されており，その SSOID を用いて SSO クライアントのサービスを利用する．ユーザは SSO クライアントのサービスを利用する手順は 2 通りある．

手順 A SSO クライアントに未認証 SSOID を提示する場合

手順 B SSO クライアントに認証済 SSOID を提示する場合

手順 A の場合，SSO クライアントはユーザを SSO 認証サーバにリダイレクトする．SSO 認証サーバは提示された未認証 SSOID に対応する認証情報を用いて，ユーザを認証し，認証済 SSOID をユーザに返す．ユーザは認証済 SSOID を SSO クライアントに提示することで，SSO クライアントのサービスを利用することができる．

手順 B の場合，**手順 A** の認証済 SSOID 取得後と同じと考えることができる．そのため，ユーザは SSO 認証サーバによって認証済 SSOID を発行されて以降は，異なる複数の SSO クライアントに対して，さらなる認証を必要とすることなく，サービスを利用することができる．

SSO クライアントは認証済 SSOID に SSO クライアント内でのみ有効なローカル ID を紐付けることで，ユーザを管理する．図 3 の例では，SSOID01 はローカル ID として LIDAA が紐付けられており，SSOID02 には LIDBB が紐付けられている．SSO クライアントでは異なるローカル ID が割り当てられているため，SSOID01 と SSOID02 は異なるユーザであると識別する．

3.3 問題の設定

ユーザは SSO 認証サーバ A から SSOID1 を発行され，SSOID1 を利用して複数の SSO クライアントからサービスを受けている．今なんらかの理由によって，SSO 認証サーバ A が機能を停止したとする．このとき，SSO 認証サーバ A から発行されている SSOID1 は認証できなくなるため，ユーザは SSOID1 を利用して SSO クライアントからサービスを受け

ることはできなくなる．

もし，ユーザは SSO 認証サーバ A とは異なる SSO 認証サーバ B から SSOID2 を発行されれば，SSO クライアントのサービスを利用することができるようになる．しかし，SSO クライアントの視点では，SSOID1 のユーザと SSOID2 のユーザは異なるユーザと識別する．そのため，ユーザは SSO クライアントにおいて SSOID1 で利用していた情報や履歴を利用することができない．

このような SSOID1 が利用できなくなった状況において，ユーザが従来より使い続けてきた情報や履歴を継続して利用する方法を問題と設定する．以上の取り扱う問題の概略図を図 4 に示す．

この問題は，SSO クライアント自身がユーザを認証することができないために発生する．SSO クライアントはユーザを認証するための認証情報を持っていないために，ユーザを認証することができない．そのため，SSO クライアントは未認証 SSOID を主張するユーザを SSO 認証サーバに転送し，SSO 認証サーバから送られる認証結果を利用して，ユーザにサービスを提供する．よって，SSO クライアントは異なる SSOID を持つユーザに対して，たとえ同一の主体であったとしても，異なるユーザとして振る舞う．なおこの際，各 SSO クライアントが持つローカル ID を利用して，各ユーザを認証する解決方法も考えられる．しかし，各 SSO クライアントごとに認証情報を設定する必要があり，またシングルサインオンとしての機能を果たさないため，検討対象外とする．

以上より，この問題を解決し，サービスを継続的に利用するためには，ID の継続性が重要となる．

4. シングルサインオンにおける ID 継続手法

本章では，3.3 節で述べた問題を解決する手法について述べる．

4.1 冗長化 SSO 認証サーバ方式

冗長化 SSO 認証サーバ方式は認証サーバを二重化または多重化する方式である．冗長化構成にすることで，全てのサーバが機能停止しない限り，継続的に機能を利用可能である．冗長化 SSO 認証サーバ方式の概略図を図 5 に示す．図 5 では，SSO 認証サーバ A は内部で SSO 認証サーバ A1 と A2 に二重化されている．このとき，SSO 認証サーバ A1 または A2 のどちらか一方でも稼働していれば，SSO 認証サーバ A は継続して利用可能である．

シングルサインオンにおいて，冗長化 SSO 認証サーバ方式を実現するには，以下の 2 通りが考えられる．

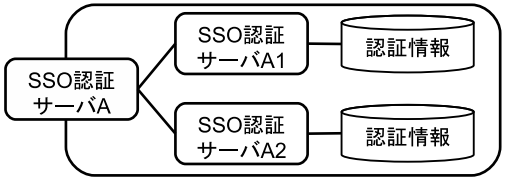

図 5 冗長化 SSO 認証サーバ方式
Fig. 5 Redundant SSO Authentication Server method.

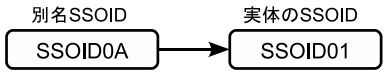

図 6 別名方式
Fig. 6 Alias method.

- 同一ドメイン内で冗長化する．
- 異なるドメイン間で冗長化する．

同一ドメイン内で行う場合，単純な冗長化構成で対応できる．しかしながら，冗長化サーバを束ねるスイッチなどの上位ネットワークに障害が発生した場合，各冗長化サーバが正常稼働していても，利用できなくなることがある．対して，異なるドメイン間で冗長化を行う場合，この問題は発生しにくいが，異なるドメインにユーザの認証情報を引き渡す必要があり，運用上の問題がある．

4.2 別名方式

別名方式は SSOID とは異なる仮名の SSOID を用いる方式である．この仮名の SSOID を別名 SSOID と呼ぶこととする．SSOID と別名 SSOID の関係を図 6 に示す．

図 6 における SSOID0A が別名 SSOID であり，実体の SSOID として SSOID01 が紐付けられている．SSOID を利用する場合，SSO クライアントには SSOID0A が提示される．ユーザの認証は実体である SSOID01 を発行する SSO 認証サーバが行う．SSOID01 が使えなくなった場合，SSOID0A に紐付ける実体の SSOID を切替えても，SSO クライアントには同じ別名 SSOID である SSOID0A が提示される．

4.3 複数 SSOID 方式

複数 SSOID 方式はローカル ID に複数の SSOID を紐付けることで，ローカル ID に対応する SSOID を多重化する方式である．1 つの SSOID が利用できなくなった状況下でも，

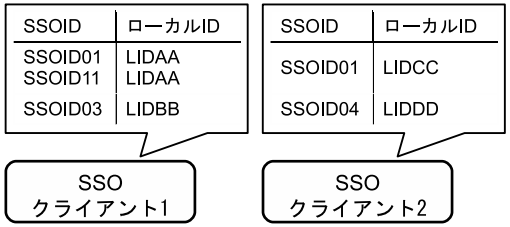

図 7 複数 SSOID 方式
Fig. 7 Multiple SSOID method.

それ以外の紐付けられた SSOID によって，継続的にサービスを利用することができる．複数 SSOID 方式の概略図を図 7 に示す．

複数 SSOID 方式はローカル ID に複数の SSOID を紐付けることで，紐付けられた SSOID からローカル ID を参照してユーザを特定し，サービスを提供する．図 7 の例では，異なる SSO 認証サーバにより発行されている SSOID である SSOID01 および SSOID11 は，ともに LIDAA というローカル ID に紐付けられている．ここでユーザは SSOID01 でログインしても SSOID11 でログインしても，SSO クライアント 1 は LIDAA という同一のユーザとして取り扱うことができる．そのため，SSO クライアントが対応するだけで利用可能となる．また，SSO 認証サーバには変更の必要がなく，どの SSO 認証サーバが発行する SSOID でも利用可能である．

その反面，SSO クライアント側での対応となるため，全ての SSO クライアントで同様の作業をする必要性がある．このように，シングルサインオンの利便性が損なわれる問題がある．また，図 7 の例では，SSOID01 を持つユーザは SSO クライアント 1 と 2 を利用していた．このユーザは SSOID11 を SSO クライアント 1 に追加したが，SSO クライアント 2 には追加をしなかった．そのため，SSOID11 で SSO クライアント 2 にログインした場合，別のローカル ID（例えば LIDEE）が付与され，SSO クライアント 2 は LIDCC とは別のユーザとして識別する．

4.4 SSOID 移行方式

SSOID 移行方式は SSO 認証サーバ同士が事前に合意することで，ある SSO 認証サーバに合意した他の SSO 認証サーバの SSOID を移行する方式である．説明のために，SSO 認証サーバ A を移行元，SSO 認証サーバ B を移行先とする．SSOID 移行方式の概略図を図 8 に示す．

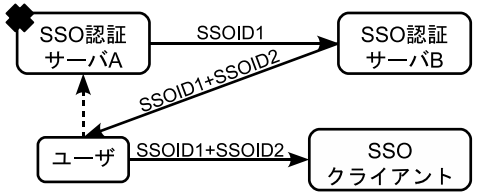

図 8 SSOID 移行方式
Fig. 8 SSOID Migration method.

SSOID 移行方式は冗長化 SSO 認証サーバ方式と同様に，複数の SSO 認証サーバを用いる．で説明したように，冗長化 SSO 認証サーバ方式では認証情報を渡す必要があり，ドメインを越えた運用に問題がある．SSOID 移行方式では，ユーザの合意の元で，SSO 認証サーバ A が発行する SSOID1 と SSO 認証サーバ B が発行する SSOID2 を紐付ける．

ここで，SSOID1 を SSOID2 に紐付ける場合を考える．SSO 認証サーバ B は SSO 認証サーバ A とは異なる主体であるので，SSOID1 についての認証情報を持っておらず，ユーザを認証することはできない．そこで，SSO 認証サーバ A は SSO 認証サーバ B に対して，SSOID1 と SSOID2 が紐付けられていることを示す付加情報を渡す．この付加情報には以下の情報が含まれる．

- SSOID1 は SSOID2 に紐付けられていることを示す情報．
- その紐付けはユーザの合意の元で SSO 認証サーバ A が許可したことを示す情報．
- 付加情報が不正に改ざんなどされていないことを示す情報．

SSO 認証サーバ B は SSOID2 に加えて SSOID1 と付加情報をユーザへ渡す．ユーザは SSO クライアントに SSOID2 と SSOID1 と付加情報を提示し，SSO クライアントは付加情報を検証し，SSOID2 と SSOID1 が正当に紐付けられていることを確認する．これによって，SSO クライアントは SSOID1 および SSOID2 に同じローカル ID を割り当てることができる．

5. 検討・考察

本章では，1 章で挙げた要件を各方式が充足するかを考察する．また，効果と影響の範囲，実現可能性について検討を行う．表 1 は各方式の比較結果を示したものである．

5.1 要件 1（継続性）

冗長化 SSO 認証サーバ方式は主 SSO 認証サーバから副 SSO 認証サーバへ切り替わるこ

表 1 比較
Table 1 comparison

	要件 1	要件 2	要件 3	範囲	実現可能性
冗長化 SSO 認証サーバ方式	○	5.2 節を参照	○	○	上位主体が必要
別名方式	○	5.2 節を参照	別名 SSOID を解決する主体が必要	○	HTML-Based Discovery[3)]
複数 SSOID 方式	5.1 節を参照	○	各 SSO クライアントで手続きが必要	手続きを行った SSO クライアントのみ	sourceforge.jp など
SSOID 移行方式	5.1 節を参照	○	○	○	文献 4)

とで，ユーザは継続的にサービスを利用することができる．

別名方式も紐付けた実体の SSOID を変更することで，別名 SSOID を変更することなく，継続的にサービスが利用できる．

複数 SSOID 方式では SSOID ではなく，各 SSO クライアントで使われる LocalID を用いて認証される．そのため，各 SSO クライアントが稼働していれば，SSO 認証サーバが停止していても，サービスは利用できる．しかしながら，LocalID を用いているため，シングルサインオンが利用できなくなる問題が残る．

SSOID 移行方式は SSO クライアントが付加情報による移行を受け入れれば，ユーザは継続的にサービスを利用することができる．しかし，SSO クライアントが付加情報による移行を受け入れない場合，SSOIDb1 と SSOIDa1 の関係性を検証できないので，ユーザは継続的にサービスを利用することはできない．

5.2 要件 2（安全性）

冗長化 SSO 認証サーバ方式は認証情報の引き渡しの問題がある．異なるドメイン間で冗長化する場合は，認証情報を安全に受け渡す必要がある．また，受け渡し先の認証ポリシーが受け渡し元の認証ポリシーと同じとは限らず，その場合，認証結果の信頼性に問題がある．

別名方式は紐付けを変更できるのがユーザだけであるならば，安全性は満たされる．ユーザ以外が変更できる場合，容易に実体の SSOID を変更できるため，なりすましが行える．

複数 SSOID 方式でなりすましが行われる場合，その影響は各 SSO クライアントにとどまる．また，その安全性は通常の認証方式と変わらない．そのため，複数 SSOID 方式の安全性は高い．

SSOID 移行方式では SSOIDa1 と SSOIDb1 の関係を示す付加情報が用いられる．この付加情報が不正に変更される場合，なりすましが行われる．しかし，4.4 節で述べたように，付加情報は改ざん検出のための情報を含んでおり，改ざんは行われないため，なりすましは行われない．また，SSOIDb1 を不正に利用しようとする場合，その安全性は通常の認証方式と同じである．

5.3 要件 3（利便性）

冗長化 SSO 認証サーバ方式は未認証 SSOID を認証する SSO 認証サーバを選択する必要がある．この処理は SSO 認証サーバ側で準備されるもので，ユーザと SSO クライアントに負担はない．対して，SSO 認証サーバ側では適切な SSO 認証サーバを選択する機能を準備する必要がある．

別名方式ではユーザが別名 SSOID を準備し，実体の SSOID と紐付ける手続きが必要である．そのため，別名 SSOID を解決する主体が必要となる．また，他の別名 SSOID に紐付け替える場合も，ユーザが手続きをする必要がある．

複数 SSOID 方式は SSO 認証サーバに追加の処理は発生しないが，SSO クライアントは複数の SSOID をローカル ID に紐付ける手続きが必要になる．また，ユーザは利用しようとする全ての SSO クライアントに対して，同様の手続きを行う必要があり，ユーザの手続きは大変煩雑である．

SSOID 移行方式は複数 SSOID 方式とは逆に，SSO クライアントとユーザの負担は少ない．対して，SSO 認証サーバは紐付けられた SSOID と付加情報を管理し，必要に応じてそれらを提示する必要がある．しかし，一般的に SSO クライアントは複数存在するため，複数 SSOID 方式よりも，SSOID 移行方式の方が利便性は高い．

5.4 効果と影響の範囲

効果と影響の範囲とは各方式によって問題を解決する際に発生する作用が及ぶ範囲とする．

冗長化 SSO 認証サーバ方式，別名方式，SSOID 移行方式は，その効果は全ての SSO クライアントに及ぶ．影響範囲は冗長化 SSO 認証サーバ方式と別名方式では，認証済 SSOID を取得後は全ての SSO クライアントに及ぶ．SSOID 移行方式は認証済 SSOID に付加情報などを付け加えるため，それらの情報を提示した SSO クライアントのみに影響範囲はとどまる．

対して，複数 SSOID 方式は効果と影響の範囲は手続きを行った SSO クライアントのみにとどまり，他の SSO クライアントには作用を一切及ぼさない．そのため，複数 SSOID 方式は効果と影響の範囲を局所的にとどめることができるが，それ以外の 3 方式では広範に及ぶ．

5.5 実現可能性

各方式によって問題を解決する際にあたって，新たに発生する問題や運用の容易さを実現可能性とする．また，実際の利用例についても言及する．

冗長化 SSO 認証サーバ方式は，5.3 でも述べたように，未認証 SSOID を認証する SSO 認証サーバを選択する必要がある．この処理は SSO 認証サーバ側で準備され，図 5 の例では SSO 認証サーバ A がその処理を行う．冗長化 SSO 認証サーバ方式の実現には，SSO 認証サーバ A のような上位主体が必要となる．

別名方式の例として，OpenID 2.0[3] の HTML-Based Discovery がある．HTML-Based Discovery は HTML ドキュメントの HEAD 要素内に LINK 要素として OP エンドポイント URL を示すことで，identifier を発見する方式である．HTML ドキュメントの URL を別名，OP エンドポイント URL を実体として捉えれば，HTML-Based Discovery は別名方式の 1 つの実現方法と言える．

複数 SSOID 方式の例として，OpenID において sourceforge.jp[⋆1]や Fastladder[⋆2]などで実際に利用されている．これらでは OpenID で Web サービスにログイン中に，さらに他の OpenID を紐付けることで，紐付けたいずれの OpenID でもログインが可能となっている．

SSOID 移行方式の例として，OpenID への適用が前田らによって提案されている[4]．前田らの方式では，紐付ける OpenID の Identifier，移行先ドメイン名，それらに対する署名情報などを付加情報として用いている．また，OpenID のプロトコルで付加情報を利用するために，OpenID 拡張機能を準備し，移行を実現する方式となっている．

6. ま と め

本稿では，シングルサインオンの認証サーバが停止などの理由により，利用できなくなった状況下において，シングルサインオンを利用するサービスの継続利用について検討を行った．シングルサインオンにおける ID 継続手法として，冗長化 SSO 認証サーバ方式，別名方式，複数 SSO-ID 方式，SSO-ID 移行方式の 4 つを挙げ，継続性，安全性，利便性，効果と影響の範囲，実現可能性について比較検討を行った．各方式ともに一長一短であるが，別名方式と SSOID 移行方式がシングルサインオン環境下において低負担で利用可能である．今後は，これらの検討を基に，継続的にサービスが利用可能なシングルサインオン方式の検討を行いたい．

⋆1 http://sourceforge.jp/
⋆2 http://fastladder.com/

参 考 文 献

1) D.Nobayashi, Y.Nakamura, T.Ikenaga, and Y.Hori. Development of single sign-on system with hardware token and key management server. *IEICE Transactions on Information and Systems*, Vol. E92-D, No.5, pp. 826–835, 2009.
2) D.Recordon and D.Reed. OpenID 2.0: a platform for user-centric identity management. In *Proc. of the second ACM workshop on Digital identity management (DIM '06)*, pp. 11–16, 2006. ACM.
3) OpenID Authentication 2.0, 2007. http://openid.net/specs/openid-authentication-2_0.html.
4) 前田千徳, 柿崎淑郎, 岩村惠市. OpenID における ID 継続の提案. コンピュータセキュリティシンポジウム 2010, pp. 441–446, 2F2-4, 2010.