# クラウドコンピューティング環境における 認証連携と属性利用技術の提案と評価

下道　高志†　　佐々木　良一†
†東京電機大学
120-8551 東京都足立区千住旭町 5 番
11udc01@ms.dendai.ac.jp,　sasaki@im.dendai.ac.jp

**あらまし** 複数のネットワークドメイン上で複数のサイトを連携するサービスは，クラウドコンピューティングの環境下で増加すると予想される．そこで必要とされる技術は，単なるサイト間の認証連携だけでなく，分散された属性情報を利用するサービスのための技術である．扱われる属性情報は静的属性情報に加え，動的属性情報も今後増加すると考えられる．本稿ではアイデンティティ管理／サービス技術であるSAML / ID-WSFに注目し，クラウド上への適用の実際と問題点を考察した上でID-WSFを拡張し，離れた地点においてもプライバシーを配慮しながら高速かつ安全に属性を利用できる技術の提案を行うとともに，提案する技術の有効性の評価も行った．

# A Proposal and an Evaluation of Technology on Federated Authentication and Usage of Attributes in Cloud Computing

Takashi Shitamichi†　　Ryoichi Sasaki†
†Tokyo Denki University
5 Senjyuasahi-cho, Adachi-ku, Tokyo, 120-8551, JAPAN
11udc01@ms.dendai.ac.jp,　sasaki@im.dendai.ac.jp

**Abstract** Federated services among multi-domain network are expected to be widely deployed onto cloud computing environment. Technology not only for federated authentication but also services using distributed attributes, that are not only static but also dynamic, are required. Focusing on SAML and ID-WSF, that are technology for identity management and services with privacy, this paper discusses deployments and problem of them in the real world to extend ID-WSF, proposes fast and safe technology which can safely use attributes, and evaluates the effectiveness of the proposed technology.

## 1　はじめに

さまざまな消費者向けサービスや企業情報システムでは，同一ネットワークドメイン上のサイトで完結するサービスだけでなく，複数のサイトがドメイン間で連携するサービスが増えている．連携されたサービスをユーザが利用する場合，複数サイト間での認証連携が必要となり，SAML (Security Assertion Markup Language)[1]，OpenID[2]，OAuth[3]といった仕様が規定されている．連携サービスを提供する各サイトが保有する個人属性情報の集合をアイデンティティと呼び，認証連携を行うことによって連携される個人属性情報の集合を，連携アイデンティティ(Federated Identity)と呼ぶ[4]．

連携アイデンティティにおける個人の属性情報利用のためには，個人情報保護やプライバシーを考慮したセキュアなプロトコルが必要であり，SAML や SAML を拡張した Shibboleth[5]を利用する属性転送方式や，SAML のアサ

ーションを利用しつつ属性利用の本人確認の仕様を備える Liberty ID-WSF[6]などが規定されている. SAML はクラウド間のセキュアな認証連携のために数多く研究され適用が進んでいる.

従来属性情報とは，氏名，住所，生年月日，性別といった静的な情報を指してきた. その一方で時間とともに変化する個人のライフイベントに関連した情報，たとえば位置情報，体温，脈拍，血圧等の生体情報，摂取した食事や飲物等の情報もある. これらは本人の嗜好や行動を示すアイデンティティであり属性である. 故に動的属性と定義することができる[7].

過去，静的属性を扱う認証連携の技術に関して，ユーザデバイス機能のローミングの研究が行われ，認証時間の短縮を行う技術は提案されている[8]. しかし認証連携と併せ，日々蓄積されていると推測される動的な属性情報を，速度と安全性を十分に配慮し扱う技術としては考案されていない[9][10][11][12].

属性情報を取り扱うサイトにおいては，ユーザのインタラクション（会話）時に属性転送を高速に行わなければ，ユーザエキスペリエンス（ユーザ体験）に影響を与える可能性がある. サイトがユーザと latency（ 遅延時間 ）が小さい地点に存在する場合には，属性転送時間は問題にならないと考えられるが，サイトがクラウド上の何処，地球の反対側のような地点に存在している場合は latency が大きくなると予想され，その結果，サービスを利用するユーザにとっての RTT (Round Trip Time)は長くなり，ユーザエキスペリエンスに大きな影響を与えると想定される.

そこで筆者は，認証連携および属性利用技術である SAML / ID-WSF を利用しつつ，静的な属性情報と動的な属性情報を高速かつ安全に扱うために，ID-WSF を拡張したアーキテクチャを考案した. 本稿では，アーキテクチャの概要を述べ，有効性を確認するために実際のクラウド上で実験を行い技術検証し，考察を行っている.

## 2 認証連携と属性利用の技術

SAML は様々なサービスにおける認証連携に適用されてきた. 一方，前節で述べたように ID-WSF は属性利用の技術として利用されている. 本節ではセキュリティ情報交換の技術として SAML のセキュリティモデルと，ID-WSF の属性利用技術の考察を行う.

### 2.1 セキュリティ情報交換の技術

SAML は連携 SSO を行うための技術と捉えられることが多いが，元々はセキュリティ情報を，ネットワークを通して交換するためのフレームワークとして，2000 年代初頭に考案された.

SAML で定義される IdP（ Identity Provider)とは，トークンの要素である名前や年齢といったクレームを作成するオーソリティであり，STS(Security Token Service)を運用する. また SP(Service Provider)はクレームを利用することによってユーザを特定し，アプリケーション・サービスを提供する. このクレームを利用するための正当性を証明する IdP からの応答をアサーションと呼び，認証・認可・属性情報を XML で記述している.

SAML 仕様書ではアサーション，プロトコル，バインディングに加え SSO 等のユースケースをプロファイルとして規定している. Web ブラウザによる SSO のプロファイルとしては次の 3 種類を定義している[13].

(1) SP-initiated SSO: Redirect/POST Bindings
(2) SP-Initiated SSO: POST/Artifact Bindings
(3) IdP-Initiated SSO: POST Binding

SSL/TLS によるチャネルセキュリティの確保に加え，Web browser に対する柔軟性と信頼されたサイト間の SOAP (Simple Object Access Protocol)通信を用いることにより，セキュリティ面が考慮された (2)の方式が注目されている[1].

[1]SSL/TLS につては脆弱性やや公開鍵の共有の問題等が報告されている[14][15]. セキュリティ対策として SSL/TLS だけでなく，改竄防止と完全性を保証するために署名を付加した SOAP メッセージはより安全とされ，政府機関等で広く適用されている.

## 2.2　属性利用の技術

ID-WSF は異なるサイト間で，ユーザの意思に基づき属性情報を安全に流通させるための仕様である．安全性を保障するための仕組みとして通信プロトコルに SOAP を利用した上で，セキュリティメカニズム仕様を規定している．通信の秘匿性とメッセージの完全性を組み合わせて定義し，セキュリティの種別を“セキュリティメカニズム ID”と呼ばれる識別子で規定している．組み合わせには多くの方法があるが，たとえばメッセージの完全性を確保するために図1の構造の SAML トークンを使うことにより，以下を実現できる[16]．

(1)　アサーションによりユーザの特定
(2)　情報要求元である送信者の認証が XML 署名により検証
(3)　第三者機関(IDP)のアサーションに含まれる送信者の公開鍵により送信者の承認

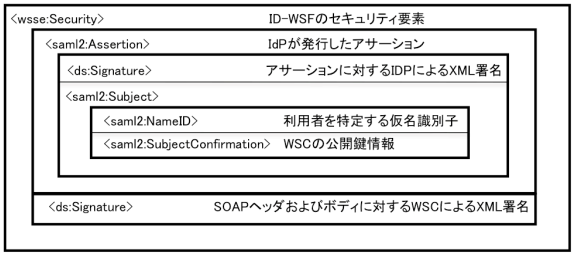
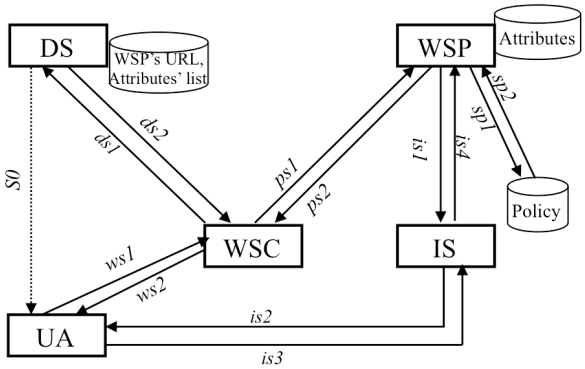



図 1　SAML トークンの構造

ID-WSF による属性利用のフローは図 2 である。ユーザが事前に特定の属性プロバイダである WSP(Web Service Provider )に属性を登録し，WSP の URL を検索サイトである DS (Discovery Service) に登録する．ユーザがサービス提供サイトである WSC（Web Service Consumer）を利用する時に，DS のポンタが WSC に通知されることによって，WSC は WSP の URL に登録されている属性情報の項目リストを DS から入手する．この時の WSC と DS 間の SOAP メッセージでのやり取りの内，DS のレスポンス概要を図 3 に示す．

WSC は項目リストにより属性を WSP に要求し，ユーザの属性情報を入手する．この際，




***S0:* an SSO assertion containing a bootstrap for DS**
***ws1*: access WSC's service with the assertion**
***ws2*: return with the result of WSC's processing**
***ds1:* require the location of WSPs orWSPd or WSPr**
***ds2*: return the locations**
***ps1*:require the user's static attributes**
***ps2*: return with the user's static attributes**
***sp1*:check the policy**
***sp2*:return the policy**
***is1*:require user consent**
***is2*:interaction with user to get his/her consent**
***is3*:return the result of the interaction**
***is4*:return his/her consent**

図 2 ID-WSF のフロー

```
<Envelope>
 <Header>
      <To>wsc URI ( e.g. https://wsc.com/wsc ) </To>
 </Header>
 <Body>
    <QueryResponse>
       <EndPointReference>
          <Address> WSP's endpoint address </Address>
          <Metadata>
               <ProviderID> WSP's service URI </ProviderID>
               <ServiceType>
                    urn:liberty:id-sis-pp:2003-08
               </ServiceType>
               <SecurityContext>.....</SecurityContext>
               <Token>
                 <SecurityMechID>
                 urn:liberty:security:2006-08:ClientTLS:SAMLV2
                 </SecurityMechID>
                 <Assertion>
                      <ds:Signature> .....</ds:Signature>
                      <saml2:Subject>.....</saml2:Subject>
                 </Assertion>
               </Token>
          </Metadata>
       </EndPointReference>
    </QueryResponse>
 </Body>
</Envelope>
```

図 3 DS のレスポンス概要

WSP は WSC から属性要求があった場合，次の 2 通りの動作を行う．

- 事前に定めたポリシーに従い属性を返す
- 属性の持ち主に可否を逐次問い合わせる (IS：Interaction Service)

属性の管理／利用技術として，ID-WSF はさまざまな分野に適用されている．コンテンツ視聴のための複数デバイス間の情報連携や，機微情報を扱う医療分野等，国内での多くの研究と実績がある[17][18][19][20][21]．

### 2.3 クラウド適用における問題点

ID-WSF は属性情報を取り扱うために，SAML トークンの利用や本人同意の下での属性転送等，セキュリティやプライバシー面での技術が確立されているが次の問題点がある.

#### 2.3.1 動的属性情報の取扱いの問題

クラウド環境では，企業が自社で管理できる信用境界を越えて，個人のプライバシーを含む情報が行き来することもあり，各国のプライバシーや個人情報保護に関連した法制度面での問題に直面する可能性もある[22]. 特に動的属性情報は個人の生活上でのプライバシーに関連する情報も数多いと考えられるため，可能な限り自分で自分の情報を信用のおけるサイトで管理したいと考えるユーザの潜在的要求も高いと考えられる. しかしながら ID-WSF は，静的属性情報の取扱いを中心として設計されており，刻々と蓄積される動的属性情報を扱うことは考慮されていない[2].

#### 2.3.2 latency の問題

さまざまなサービスがクラウド環境上に構築されると考えられるが，必ずしも国内に閉じた環境ではなく，世界のどこかのデータセンターで稼働すると考えるべきである. その場合，サイトが世界中に散在すると，“ネットワークの遠さ”= latency（ 遅延時間 ）がサービス提供において問題となると考えられる.

たとえば，内容的には全く同じサービスが，在処が違うサイトから提供されるとする. 東京在住者が東京のサイトが提供するサービスを利用するのと，ロンドンのサイトが提供するものとでは，ユーザエキスペリエンスに差が出てくるが，距離と中継設備による latency が主な原因と考えられる. ユーザエキスペリエンス上 latency は非常に重要と考えられている. [3]

[2] Liberty 仕様の中には，ある時点における位置情報だけの取得を目的とした Geo Location Service 仕様が存在する[23].

[3] “Web を利用するユーザは，読み込みに 3 秒以上かかると苛立ちを感じ，47%のユーザが 2 秒以内の Web ページ読み込みを期待し，Web ページの読み込み時間に 3 秒以上かかるとユーザの 40 %がそのサイトを去る”との調査結果が報告されている[24]

ID-WSF においてサービスを提供する WSC は，サービスアプリケーションの挙動によって随時条件を変えながら，WSP に（特に動的）属性情報を問い合わせることも考えられる. この場合，WSC と WSP 間の通信の latency を小さくすることは，ユーザエキスペリエンス向上に必要と考えられるが，ID-WSF では latency を小さくする技術的対策は考慮されていない.

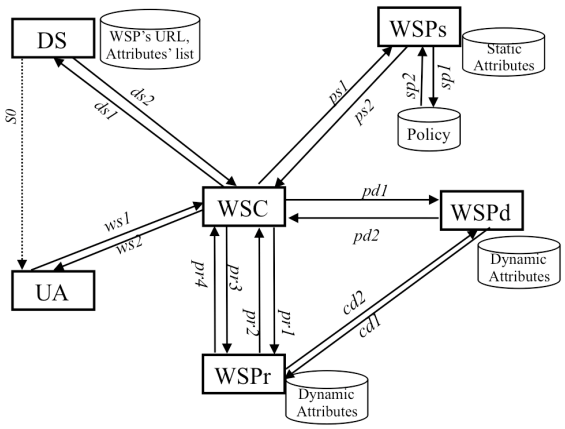
## 3 提案の方式

### 3.1 アーキテクチャの概要

前節の 2 つの問題点を解決するために ID-WSF を次のように拡張した.

(1) WSP を静的属性情報を扱うサイト WSPs と動的属性情報を扱うサイト WSPd とに分離する.
(2) 動的属性情報は WSC と latency の小さいサイト WSPr へローミングし，WSC と WSPd 間の latency を解決する.

アーキテクチャとメッセージフローの概要および追加したメッセージフローを図 4 に示す. 図 2 ID-WSF フローに追加したフローが図 4 の pd1~pd2, pr1~pr4, cd1~cd2 である.




***pd1*: require the user's dynamic attributes**
***pd2*: return with cache option instead of attributes**
***pr1*: require preparing cached dynamic attributes**
***pr2*: return with the result of preparation**
***pr3*: require the user's dynamic attributes**
***pr4*: return with the dynamic attributes**
**(*pr3 and pr4are recurring if WSC wants the attributes)***
***cd1*: require bulk of the user's dynamic attributes**
***cd2*: transfer bulk of the user's dynamic attributes**

図 4 動的属性情報ローミング方式フロー図

## 3.2 プロファイルの定義

ID-WSF で規定している ID-SIS を拡張し，id-sis-wps(静的属性)， id-sis-wpd（動的属性)， id-sis-wpr（ローミング）の 3 つのプロファイルを規定する．id-sis-wps は ID-SIS Personal Profile である id-sis-pp の上位互換とする．要素<DyamicAttributes>を id-sis-pp の XML スキーマに追加し，動的属性プロファイル名を次の形式で記述する．

```
<wps>
  <InformalName>shita</InformalName>
   <CommonName>
      <CN>Takashi Shitamichi</CN>
      <AnalyzedName nameScheme=" ">
          <PersonalTitle>Mr.</PersonalTitle>
          <FN>Takashi</FN>
          <SN>Shitamichi</SN>
      </AnalyzedName>
          ……..
   <DynamicAttributes>
     <DProfileName>urn:id-sis-wpd</DProfileName>
     ……
   </DynamicAttributes>
</wps>
```

## 3.3 動的属性呼び出しとローミング

WSC による WSPs の呼び出し時において，<DynamicAttributes>で id-sis-wpd が示されているので，WSC から DS に対し WSPd の在処を次の形式で問い合わせる．

```
<disco:Query xmlns:disco="urn:liberty:disco:2005-11" id="discReq">
  <disco: ResourceID>https://idp-ds.com/ dca24a6f3f</disco:ResourceID>
  <disco:RequestedService>
    <disco:ServiceType>urn:id-sis-wpd:2012-06</disco:ServiceType>
    <disco:SecurityMechID>
        urn:liberty:security:2006-08:ClientTLS:SAMLV2
    </disco:SecurityMechID>
    <disco:Action>urn:id-sis-wpd:2012-06:GetAttributes</disco:Action>
  </disco:RequestedService>
</disco:Query>
```

DS は次のレスポンスを返す．

```
<disco:QueryResponse   disco="urn:liberty:disco:2005-11" >
 <Status code="OK"/>
 <disco:ResourceOffering entryID="1">
  <disco:ResourceID>
       https://sp.wspd.com/dda825cfef</disco:ResourceID>
  <disco:ServiceInstance>
   <disco:ServiceType>
         urn:id-sis-wpd:2012-06</disco:ServiceType>
   <disco:ProviderID>https://www.wspd.com</disco:ProviderID>
   <disco:Endpoint>https://sp.wspd.com:443/soap</disco:ProviderID>
   <wsa:Metadata>
    <disco:SecurityContext>
     <disco:SecurityMechID>
         urn:liberty:security:2006-08:ClientTLS:SAMLV2
     </disco:SecurityMechID>
     <sec:Token xmlns:sec="urn:liberty:security:2006-08" usage=.. >
      <sa:Assertion xmlns:sa="urn:oasis:names:tc:SAML:2.0:assertion">
                ….Assertion data…</sa:Assertion>
     </sec:Token>
    </disco:SecurityContext>
   </wsa:Metadata>
  </disco:ServiceInstance>
 </disco:ResoueceOffering>
</disco:QueryResponse>
```

得られた ResourceID 等の情報により，WSC は WSPd を次の形式で呼び出す．

```
<wpd:Query xmlns:wpd="urn:id-sis-wpd:2012-08">
<wpd:ResourceID>https://sp.wspd.com/dda825cfef</wspd:ResouceID>
  <wpd:QueryItem itemID="activity">
      <wpd:Select>/wpd:WPD/wpd:Activities</wpd:Select>
 </wpd:QueryItem>
</wpd:Query>
```

WSPd はこの呼び出しを受け，動的属性を返すか，もしくはローミングへ誘導する．ローミングへ誘導する場合は以下を返す．

```
<wpd:QueryResponse xmlns:wpd="urn:id-sis-wpd:2012-08">
  <wpd:Status code="ROAM"/>
</wpd:QueryResponse>
```

WSPd から"ROAM"が返されることにより，WSC はローミングの動作を開始する． まず ServiceType を id-sis-wpr に指定し DS に問い合わせを行う．DS は WSPr の ResouceID 等の情報を返す．WSPd に WSPr へのローミングを依頼するために WSC は WSPr に対して次の形式で呼び出しを行う．

```
<wpr:Query xmlns:wpd="urn:id-sis-wpr:2012-08">
<wpr:ResourceID>https://sp.wspr.com/dfa8c5djaf</wspr:ResouceID>
  <wpr:QueryItem itemID="roam">
      <wpr:Select>/wpr:WPR/wpr:Roam</wpr:Select>
  </wpr:QueryItem>
</wpr:Query>
```

WSPr は BulkRequest を指定し，WSPd を次の形式で呼び出す．

```
<wpd:Query xmlns:wpd="urn:id-sis-wpd:2012-08">
<wpd:ResourceID>http://sp.wspd.com/dda825cfef</wspr:ResouceID>
  <wpd:BulkRequest>
   <wpd:QueryItem itemGroup="activity">
      <wpr:Select>/wpr:WPR/wpr:Activities</wpr:Select>
  </wpd:QueryItem>
  </wpd:BulkRequest>
</wpd:Query>
```

ローミングは WSPd から WSPr へ以下の形式の bulk データ送信によって行う．

```
<wpd:QueryResponse xmlns:wpd="urn:id-sis-wpd:2012-08">
<Status code="OK"/>
 <wpd:BulkRequestResponse>
   <wpd:QueryItem itemGroup="activity">
      <wpd:GroupEntity id=1>
        <wpd:timestamp>2012-08-12T23:21:09Z</wpd:timestamp>
        <wpd:vital_signs>
           <wpd:blood_pressure>132:82</wpd:blood_pressure>
            ……..
        </wpd:vital_signs>
        ……..
      </wpd:GroupEntity id=1>
      <wpd:GroupEntity id=2>
      ……
      </wpd:GroupEntity id=n>
  </wpd:QueryItem>
  </wpd:BulkRequestResponse>
</wpd:QueryResponse>
```

ローミング終了時に WSC へ成功のステータスを返すことにより，WSC は WSPr を WSPd のプロキシとして認識する．その後，UA からのリクエストに対しては，WSPr はあたかも WSPd と同様に振る舞うことが可能となる．

# 4　評価

## 4.1　評価対象の抽出

提案したアーキテクチャの有効性を評価するために実験を行った．事前実験として SAML SSO Artifact Profile の実験をパブリッククラウドである AWS(Amazon Web Services)の世界 7 カ所間で行い，SOAP 通信の latency 特性について次が分かった．

- 世界 7 カ所のサイト（Ireland, Sao Paulo, Virginia, Tokyo, Oregon, California, Singapore）間で latency が最大となるのは Singapore/Ireland 間である．
- その際，往復の SOAP 通信にかかる時間は 765ms である．
- 一方，同じ Tokyo サイト内同士の SOAP 通信では 49ms であった．
- SAML トークンの処理等，重いとされている処理であっても 30ms で終了している．

この事前実験の結果を踏まえて，提案するアーキテクチャが有効であるかを評価することとした．すなわち WSC がローミング機能を利用することによる効果によって，分離された動的属性情報を取得する際の latency が実際に小さくなるかの確認を行うべく実験を行った．

図 5 が提案したアーキテクチャ全体のシーケンス図である．ここでシーケンスの経過時間を属性情報全般（ID-WSF 仕様），動的属性情報固有，ローミング固有に 3 分類し，各々（$Ts1$, $Ts2$），（$Td1$, $Td2$），（$Tr$）としている．ローミングを行わない場合，WSC が必要とする動的属性を全部得るために $S8$, $S9$ が $n$ 回実行される．

ここで S8，S9 の開始時間／終了時間を $td1/td2$, $td3/td4$，内部処理時間を $tdp$，総経過時間である RTT を $Tdt$ とすると，

$$Tdt = \sum_{i=1}^{n}(\Delta S8 + \Delta S9 + tdp)i = \sum_{i=1}^{n}((t2 - t1) + (t4 - t3) + tdp)i$$

となる．ローミングの RTT を $Trt$ とすると，

$$Trt = \sum_{i=12}^{17}\Delta Si + \sum_{i=1}^{5}(trp)i = (tr2 - tr1) + (tr3 - tr2) + \ldots$$
$$+(tr12 - tr11) = tr12 - tr1$$

である．この比較部分を対象に実験用プログラムを作成し，AWS 上で実行した．

## 4.2　AWS 上での実験と結果

シナリオを作成し WSC，WSPd，WSPr を AWS の 3 つのサイトで稼働させた．

### (1)　実験環境

サービス提供サイト WSC を Singapore，属性情報提供サイト WSPd を Ireland，Tokyo，Singapore，ローミング・サイト WSPr は WSC と同じデータセンターに，DS/IdP は Tokyo に構築した．実験システムは C 言語で記述した．稼働条件は以下の通りである．

[AWS インスタンス]

1GB mem, CentOS5.7, Apache 2.2, MySQL5.1

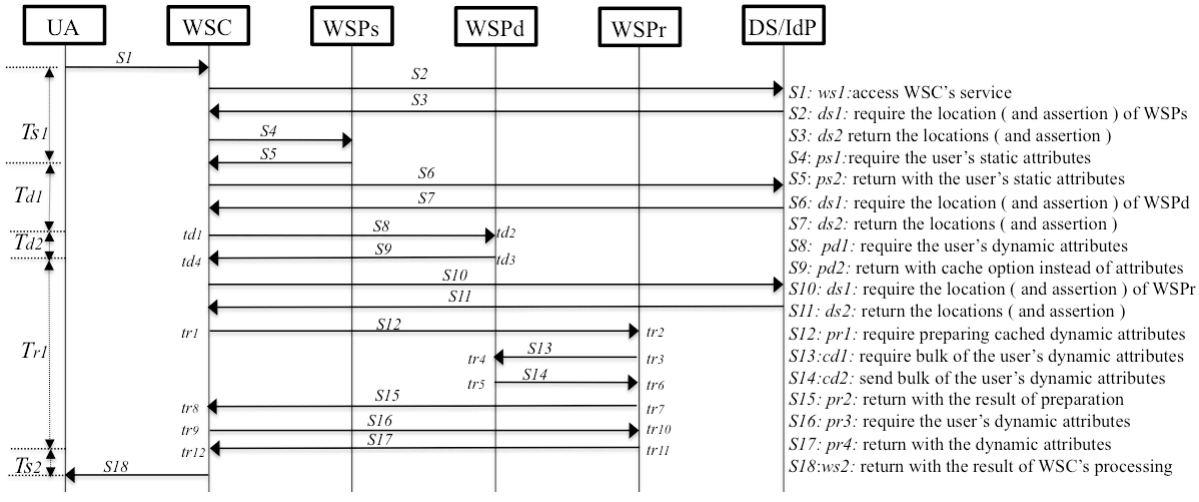



図 5 ローミング方式のシーケンス図

[動的属性情報]

レコード長 1024, 2048, 8192, 16384 byte

(2) 実験の方法

4 種類のレコード毎に,レコード数 N を N=1~10, 20, 50, 100, 200, 500, 1000 と変化させ，非ローミング方式の RTT である Trt とローミング方式の RTT である Tdt を計測した. WSC は Singapore 固定とし，WSPd を Singapore, Tokyo, Ireland 各々について 10 回実行し，平均値を取得した.

(3) 実験結果と考察

実験の結果，提案するローミング方式は，非常に有効であることが確認された. レコード長 8192byte での測定結果を図 6，表 1 に示す.

- まず非ローミング方式を測定した. WSC, WSPd が同じ Singapore の場合，動的属性取得に要する RTT は， N=1000 まで直線的に増加している. これは作成したプログラムが正しく稼働していることを示す.
- 非ローミングで WSC を Singapore,WSPd を Tokyo, Ireland と変えた場合の RTT も，N が増えるに従って N=100 迄ほぼ直線的に増加している.（N>100 以上の場合，計測回毎に大きくばらつきがあった. 特に遠隔地である Ireland との間ではコリジョンの影響を受けた可能性が高い）
- 次にローミング方式を測定した. Tokyo, Ireland 共に N=3 の時点で，Tdt< Trt となり，ローミングした方が高速となった.

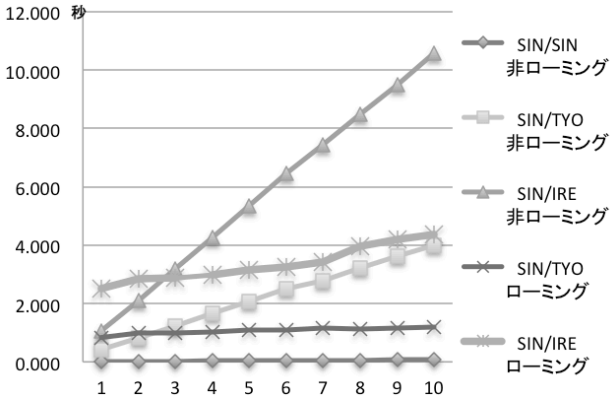



図 6 N=1~10 における各方式の RTT 比較

- ローミング方式は bulk 転送のため，非ローミング方式と比較すると転送データを一纏めにするオーバーヘッドがあるが，1 回だけで行われるので，N が大きくなればその影響は軽微であった.
- bulk 転送では N が大きい場合，一度の送信で大きなデータを送る. N=1000 の場合には，8192 x 1000=8MB のデータを一括して転送したが，RTT は N=10 のほぼ 10 倍であり，実験で利用した範囲のデータ量では問題は起きなかった.

## 4.3 ローミング方式についての考察

クラウド環境上でのサービスにおいては，静的属性だけでなく動的属性を扱うサービスが今後増えると考えられるが，大量の動的属性情報を司るサービスでは，サービス提供サイトと動的属性提供サイトの間の latency が非常に重要となると考えられる.

本実験では，Singapore / Tokyo 間において 100 件のレコードでの RTT は 2.79 秒であり，このサイト間でのローミング方式適用は十分に実用的であることを実証できた. 一方，Singapore / Ireland 間については，十分に速

表 1 各方式の RTT（単位：秒）(レコード長=8192 byte)

| | SIN./SIN | SIN/TYO | SIN/IRE | SIN/TYO | SIN/IRE |
|---|---|---|---|---|---|
| N | 非ローミング | | | ローミング | |
| 1 | 0.011 | 0.426 | 1.067 | 0.835 | 2.513 |
| 2 | 0.016 | 0.815 | 2.098 | 0.986 | 2.858 |
| 3 | 0.025 | 1.211 | 3.192 | 1.003 | 2.874 |
| 4 | 0.028 | 1.660 | 4.256 | 1.018 | 2.977 |
| 5 | 0.040 | 2.066 | 5.335 | 1.077 | 3.166 |
| 6 | 0.046 | 2.515 | 6.463 | 1.084 | 3.264 |
| 7 | 0.054 | 2.768 | 7.442 | 1.144 | 3.436 |
| 8 | 0.060 | 3.215 | 8.476 | 1.134 | 3.961 |
| 9 | 0.071 | 3.614 | 9.514 | 1.172 | 4.203 |
| 10 | 0.078 | 3.996 | 10.595 | 1.184 | 4.356 |
| 20 | 0.144 | 8.150 | 21.969 | 1.492 | 6.600 |
| 50 | 0.358 | 20.493 | 54.021 | 1.941 | 6.644 |
| 100 | 0.723 | 40.733 | 107.388 | 2.749 | 8.274 |
| 200 | 1.412 | 85.978 | 243.231 | 3.724 | 10.569 |
| 500 | 3.899 | 220.732 | 642.321 | 7.522 | 19.447 |
| 1000 | 8.260 | 410.343 | 1326.847 | 16.563 | 42.480 |

い RTT とは言えないが，100 件のレコードについて非ローミングと比較して 13 倍の RTT が得られており，高速化が有効に機能している．

## 5　おわりに

本稿では，クラウドコンピューティングの環境を想定し，SAML / ID-WSF を拡張し，静的属性と動的属性を分離した．さらにサービス提供サイトとの間の latency の低いサイトに動的属性情報をローミングし，RTT を短くすることによって，ユーザエキスペリエンスを向上させる方式を提案した．地理的にも遠いサイト間での通信を，実際のパブリッククラウド環境である AWS に構築したサイトで実験することにより，ローミング方式がユーザエキスペリエンス向上に対し，有効となる可能性が高いことを確認できた．これらの実験の結果から得た定量的データのさらなる分析を進め，ローミング方式を改良しながら更なる実装に反映していく．

## 参考文献


[1] OASIS Security Services (SAML) TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

[2] OpenID Authentication 2.0 – Final, http://openid.net/specs/openid-authentication-2_0.html

[3] The OAuth 1.0 Protocol, http://tools.ietf.org/html/rfc5849

[4] 下道，佐々木: クラウドコンピューティングにおける認証連携と属性利用技術に関する考察，研究報告コンピュータセキュリティ（CSEC）, 2012-CSEC-56(42),

[5] Internet2 Shibboleth Project http://www.internet2.edu/shibboleth

[6] Liberty Alliance ID-WSF 2.0 Specifications including Errata v1.0 Updates, http://www.projectliberty.org/resource_center/specifications/liberty_alliance_id_wsf_2_0_specifications_including_errata_v1_0_updates/

[7]下江達二：アイデンティティ管理技術の進展と変遷(<特集>Web アイデンティティと AI)，人工知能学会誌 24(4)，社団法人人工知能学会

[8] Y.Takeda, S.Kondo, Y.Kitayama, M.Torato, T.Motegi：Avoidance of Performance Bottlenecks Caused by HTTP Redirect in Identity Management Protocols, DIM '06 Proceedings of the second ACM workshop on Digital identity management, ACM

[9] 千葉，他：属性情報プロバイダ：安全な個人属性の活用基盤の提言，情報処理学会論文誌，Vol 47 No.3，Mar. 2006

[10] 畠山：異なる連携プロトコルを仲介するプロキシ型属性情報管理システム，情報処理学会創立 50 周年記念（第 72 回）全国大会，5F-1

[11] 牧，他：ID マッピング情報の登録方式に関する一考察，情報処理学会第 73 回全国大会，4E-2

[12] 鷲尾，他：クラウド向け認証基盤プラットフォームの実装と検証，情報処理学会第 73 回全国大会，4E-1

[13] Security Assertion Markup Language (SAML) V2.0 Technical Overview, 25 March 2008, OASIS

[14] 須賀：SSL/TLS renegotiation 機能の脆弱性に伴なう移行における問題点，IPSJ SIG Notes 2010-CSEC-50(12), 1-4, 2010-06-24

[15] Nadia Heninger, Zakir Durumeric, Eric Wustrow, J. Alex Halderman, "Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices", USENIX Security '12

[16] 菅野：ID-WSF2.0 を利用したセキュアな情報流通，カンターラ・イニシアチブ・セミナー2011

[17] 藤井，石川，他：複数デバイス間での認証情報連携によるシームレスなコンテンツ視聴サービス，社団法人映像情報メディア学会技術報告，2008 年 9 月

[18] 爰川，他：医療・健康情報の流通・活用に向けた情報連携基盤の提案，情報処理学会研究報告，Vol.2009-DPS-141 No.14

[19] 堀川：コンシューマ向け ID 連携サービスの構築・運用の実際とその戦略性, 信学技法, IN2009-117(2010-1), 電子情報通信学会

[20] 山村，他：放送を起点とした個人向け通信サービス利用におけるユーザー機器認証フレームワーク，FIT2010（第 9 回情報科学技術フォーラム），L-035

[21] M.Hatakeyama, S.Shima: Privilege Federation between Different User Profiles for Service Federation, DIM '08 Proceedings of the 4th ACM workshop on Digital identity management, ACM

[22] 下道：クラウドコンピューティングの現状と欧米におけるプライバシーへの取組み（《特集　ネット検索サービス事業の諸問題》），法とコンピュータ学会誌 No.28 July 2010，法とコンピュータ学会

[23] Liberty ID-SIS Geolocation Service Specification, http://www.projectliberty.org/resource_center/specifications/liberty_alliance_id_sis_1_0_specifications/

[24] Forrester Research：eCommerce Web Site Performance Today - An Updated Look At Consumer Reaction To A Poor Online Shopping Experience - , August 17, 2009