九州大学学術情報リポジトリ
Kyushu University Institutional Repository

# 分散SSO機構を用いたコミュニティ認可に関する考察

小野, 昂
九州大学理学部物理学科情報理学コース : 学生

阿部, 英司
九州大学大学院システム情報科学府 : 修士

中國, 真教
九州大学情報基盤研究開発センター : 准教授 : 情報学

笠原, 義晃
九州大学情報基盤研究開発センター : 助教 : 情報学

他

https://hdl.handle.net/2324/15955

出版情報：情報処理学会火の国情報シンポジウム2009. B-6-2, 2009-03. 情報処理学会
バージョン：
権利関係：

KYUSHU UNIVERSITY

# 分散SSO機構を用いたコミュニティ認可に関する考察

小野昂† 阿部英司‡ 中國真教* 笠原義晃* 伊東栄典*

†九州大学理学部物理学科情報理学コース ‡九州大学大学院システム情報科学研究府情報理学専攻
*九州大学情報基盤研究開発センター 〒812-8581 福岡市東区箱崎 6-10-1

E-mail: †sc105013@s.kyushu-u.ac.jp, ‡eiji.abe@i.kyushu-u.ac.jp, *{nakakuni, kasahara, itou}@cc.kyushu-u.ac.jp

**あらまし** 近年, SNS (Social Network Service) やポータルなどの個人向け Web 情報サービス (Personalized service) が普及している。これらの情報サービスでは利用者特定に利用者認証を必要とする。要認証 Web 情報サービスのために，Shibboleth や OpenID といった分散 SSO 機構が実現されている。Shibboleth や OpenID では，サービス提供者（SP: Service Provider）と，認証者（IdP: Identity Provider）が分割されている。そのため， Shibboleth 等の利用者認証機構を用いれば，認証を要する情報サービスでも，サービス連携（Web Mash up）が実現できる。要認証 Web 情報サービスの連携を考える場合，認証やサービス自身の連携については障壁が低い。一方，ある情報資源に対する閲覧・編集といった操作を許すか否かという「認可」については問題が多い。利用者の所属部局や性別といった属性に応じて認可制御を行う方法が提案されている。属性認可は，利用者情報を均一かつ大規模に管理している情報サービスには適している。一方，利用者数が小規模な場合や，利用者の管理組織が複数の場合には適していない。利用者数が小規模な場合は，認可対象者をグループやコミュニティとして扱う方が良い。本研究では，分散 SSO 機構を用いる要認証サービスの間で，コミュニティを用いて情報資源の認可を行う仕組みについて検討する。検討結果と試作システムについても述べる。

**キーワード** Web 情報サービス，認証，認可，サービス連携，Web SSO，Shibboleth

# A study of community based authorization on distributed SSO platform

Akira Ono†, Eiji Abe‡, Masanori Nakakuni*, Yoshiaki Kasahara* and Eisuke Ito*

† Dept. of Informatics, Kyushu University ‡ Dept. of Informatics, Kyushu University,
* Research Institute for IT, Kyushu University, Hakozaki 6-10-1, Higashi-ku, Fukuoka, 812-8581 Japan

E-mail: †sc105013@s.kyushu-u.ac.jp, ‡eiji.abe@i.kyushu-u.ac.jp, * {nakakuni, kasahara, itou}@cc.kyushu-u.ac.jp

**Abstract** A lot of closed services are provided on the web, where closed service is personalized or membership oriented service such as SNS (Social Network Service) or Web portal. Most closed services have user authentication mechanism to identify user. Shibboleth and OpenID are proposed to realize distributed Web SSO (Single Sign-on). In Shibboleth or OpenID, SP (Service Provider) and IdP (Identity Provider) are separated because of load balancing. Distributed web SSO mechanism makes easy to mash up of closed Web services. For mashing up of closed services, user authentication is not so hard obstacle, but authorization becomes problem. There are some conventional authorization mechanisms such as access control list, and attribute based authorization. Attribute based authorization is suitable for internal closed service in large scale organization because the organization usually has user DB, and the DB scheme is well defined. On the other hand, ACL is suitable for small scale community services or inter-domain closed services. In this paper, we show a study of community based authorization mechanism on distributed Web SSO platform. We also show a prototype system and some results.

**Keyword** Web information service, authentication, authorization, Web mash up, Web SSO, Shibboleth

## 1. はじめに

Web 上の利用者間情報交換･共有サービスでは,様々なコンテンツを扱うものがあり，その中にはコミュニティの形成を促すようなサービスがある。例えば mixi や Facebook などの SNS(Social Network Service)では，価値観や目的の似通う仲間との情報交換のために，コミュニティが形成できる。

既存の Web 情報システムは，システム毎にサービス内容が定まっており，かつ情報共有の場はシステムの

内部に限定されている。例えば前述の mixi の場合，日記や写真は共有できるものの,動画の共有はできない。動画共有を mixi コミュニティのメンバー間で行いたい場合は，外部の動画共有サービスにコミュニティの全メンバーが登録し，そちらでのコミュニティ再形成が必要となる。これは情報共有の大きな障害となる。

そこで，あるサービスで定義するコミュニティの内部に情報交換･共有範囲を限定しつつ,外部システムのサービスを利用することが求められている。しかしながら，既存のコミュニティ形成サービスや情報共有サービスは共有範囲等の取得に利用者認証を要する場合が多く，複数サービス間での連携は困難であった。

近年，一度の認証で，制御範囲内の情報システムを認証なしに横断的に使用できるシングルサインオン（SSO）が普及している。分散 SSO 機構である Shibboleth や OpenID では,負荷分散と構成の柔軟性確保のために，サービス提供者と認証者が分離されている。分散 SSO 機構であるコミュニティを形成するサービスと，動画共有などの情報共有サービスが，同一の分散 SSO 配下になっていれば,二つのサービスを連携することで,情報交換･共有範囲をコミュニティ内の者に限定しつつ，様々なシステムのサービスを提供することが可能になる。つまり,サービス A でコミュニティを作成し,サービス B の情報共有範囲を A のコミュニティに限定するといった制御が可能となる。

本研究では，分散 SSO 認証機構を用いたコミュニティ認可システムの実現を目指す。分散 SSO 技術として Shibboleth を使用する。

## 2. 認証と認可

### 2.1. 認証

認証（Authentication）とは利用者の真正性を確認する行為で，一般的な情報サービスでは，利用者が情報資源にアクセスする際に認証を行う。

認証手法は，知識を用いる手法，所有物を用いる手法，性質や特徴を用いる手法の三種類に大別できる。知識を用いる手法としては，パスワード認証がある。所有物を用いる手法としては，電子証明書による利用者認証がある。性質や特徴を用いる手法としては，指紋や虹彩などの生体情報を用いる生体認証がある。

### 2.2. 認可

認可（Authorization）とは，ある情報資源に対し，利用者がその情報資源を操作する権限を持つか否かを判断する行為である。

既存の認可手法としては，ACL（Access Control List）あるいは ACM（Access Control Matrix）を用いる方法と，利用者属性に基づく属性認可の方法がある。

### 2.3. Web SSO

SSO とは利用者が一度の認証だけで，以降，認証を行わず，許可されるすべてのサービスを利用できるシステムである。

現在，多くの個人向け情報サービスは，利用者の特定と個人情報保護のため利用者認証を行う。多くの場合，認証手法として ID・PW を用いる。そのため個人向け情報サービスが増えるたびに，利用者の管理すべき ID・PW の組が増加する。また，それぞれのサービスの利用開始時には，認証作業として ID・PW 入力が必要である。そのため，ID・PW の管理や，入力操作が煩雑になり,利用者からも管理者からも SSO によるサービス連携の要求が高まっている。

SSO にはエージェント型とリバースプロキシー型の二種類がある。エージェント型 SSO は，各 Web サーバーでの認証を代行するソフトウェアを組み込んで SSO を実現する。その仕組みとしては Cookie を用いる手法と SAML を用いる手法がある。リバースプロキシー型の SSO は,プロキシーが認証を代行することで SSO を実現する。リバースプロキシー型では，各情報提供システム側は少ない対応作業で SSO を実現できるという利点がある。しかし，リバースプロキシーへ負荷が集中する構造のため,規模適応性の問題がある。

### 2.4. Cookie を用いた SSO

Cookie とは Web サイト毎あるいは URL ごとに付加することができる付加情報で,特定の Web サーバーとブラウザーとの間で交換するテキスト情報である。ある一つの Cookie 対応 Web サーバーにアクセスすると，サーバーは利用者のブラウザーへ Cookie となるテキスト情報を送信し，Cookie に対応したブラウザーであれば受け取った Cookie 情報を保存する。Cookie には，Cookie 名と内容，発行したサイト名（ドメイン名）とパス名，有効期限，セキュリティ（SSL）設定に関する情報が含まれる。

利用者が同じ Web サイトへアクセスすると,ブラウザーはページ取得要求と共に Cookie を送信する。Web サーバーは送信された Cookie を用いて,利用者に合わせたサービスの提供などを行う。

Cookie を用いた SSO の実現方法について説明する。利用者が SSO に対応した Web サーバーのリソースにアクセスする。リソースは利用者が認証済みでないことを確かめた後，利用者に認証を要求する。利用者は ID とパスワードを，Web サーバーを通して SSO サーバーに送信する。ID とパスワードを受信した SSO サーバーは利用者を認証し，Cookie を利用者に送信する。この Cookie が SSO 環境で使用される。

Cookie を用いた SSO では SSO の対象となるサービ

スか同一ドメインに限定される。Cookie には Cookie が有効となるドメインが記載され，ブラウザーはそこで指定されたドメインのサーバーにのみ Cookie を送信する。例えば，有効なドメインとして，＊.kyushu-u.ac.jp が指定されていれば，＊.kyushu-u.ac.jp のサーバーへは Cookie が送付される。従って，Cookie を用いた SSO では，同一ドメイン内のサーバー間では SSO を実現できものの，他のドメインでは Cookie が使えないため，ドメイン間での SSO は実現できない。

### 2.5. SAML

SAML(Security Assertion Markup Language)は，異なるシステム間において，ID やパスワード等の認証情報を安全に交換するプロトコルで，2002 年 11 月に XML 関連の標準化団体である OASIS(Organization for the Advancement of Structured Information Standards)によって version 1.0 が策定され，2005 年 3 月に version 2.0 が策定された。

SAML はメッセージの送受信方法として，HTTP を使用し，メッセージの記述形式として SOAP（Simple Object Access Protocol）を使用する。SOAP は XML をベースとした記述形式で，他のコンピュータに存在するデータやサービスの呼び出しに使用される。SAML を HTML や SOAP のメッセージに付加することにより，あるサーバーで認証を受けた利用者の認証情報を異なるサーバーに共有する。そのため，SSO を実現することができる。

SAML による SSO の実現方法を説明するため，まず SAML で使用する用語を説明する。SAML は SAML オーソリティーとポリシー実行点というシステム実体と，要求に対する応答であるアサーション(Assertion)を規定する。

アサーションとは，要求に対する応答・返事である。アサーションには対象となる利用者の認証や属性，リソースに関する認可権限が含まれており，SAML オーソリティーによって発行される。

SAML オーソリティーとは，アサーションを発行するためのシステム実体である。SAML オーソリティーには認証オーソリティー，属性オーソリティー，ポリシー決定点(PDP: Policy Decision Point)の三つがあり，各部が役割に沿ったアサーションを提供する。認証オーソリティーは利用者の認証権限を認証アサーション(Authentication Assertion)として返す。属性オーソリティーは利用者の登録されている属性情報を属性アサーション(Attribute Assertion)として返す。ポリシー決定点はポリシーに基づいて，利用者に認可権限が与えられているかどうか判断し，認可決定アサーション(Authorization Decision Assertion)を返す。

ポリシー実行点(PEP: Policy Enforcement Point)とは，認可決定アサーションに従い，実際のアクセス制御が行われるシステム実体である。

SAML オーソリティーとポリシー実行点は，異なるサーバーに配置される。SAML オーソリティーが配置されたサーバーは Identity Provider(IdP)，ポリシー実行点が配置されたサーバーは Service Provider(SP)と呼ばれる。IdP は利用者の認証を行い，アサーションを発行する。SP は認可判断を行い，実際にサービスを提供する。

## 3. コミュニティ認可

### 3.1. 認可手法の選択

コミュニティ認可を実現するために，コミュニティを形成する組織数，コミュニティに属する利用者の規模を想定した上で，認可手法を選定する。

#### 3.1.1. コミュニティおよびコミュニティの規模

コミュニティとは利用者の集まりである。利用者が同一の枠組みを持つことがコミュニティを形成する条件である。例えば，九州大学の学生は九州大学という同一の枠組みを持つコミュニティである。

コミュニティを大規模なコミュニティと小規模なコミュニティに分類する。大規模なコミュニティの例として，学生，企業，自治体等がある。大規模なコミュニティは，下位に複数の小規模なコミュニティを内包する。下位のコミュニティは，上位のコミュニティより，具体的な目的を持つ。内部のコミュニティを階層コミュニティと言う。階層コミュニティもまた，内部に階層コミュニティを形成する場合がある。例えば，大学では学部という階層コミュニティが存在し，その内部に学科という階層コミュニティが存在する。階層化されているため，九州大学の例では，物理学科に属する学生は理学部に属することになる。

大規模なコミュニティに該当しないコミュニティを小規模なコミュニティとする。多くの場合，小規模コミュニティは大規模コミュニティに属する利用者が個別に集まったものである。例えば，大学のサークルは小規模コミュニティである。大学は大規模なコミュニティであり，大学に属する学生により，サークルが作られる。しかし，異なる学部・学科の学生が同一サークルに属するため，所属サークルから上位のコミュニティは判断できない。サークルは大学の階層とは異なるコミュニティである。

#### 3.1.2. コミュニティを形成する組織数

コミュニティを形成する組織数は単一，もしくは複

数を考える。

単一組織からコミュニティを形成する場合，そのコミュニティの利用者は，すべて単一のコミュニティに属する。例えば，A 大学のサークルを考える。サークルに属する学生がすべて A 大学の学生である場合，単一組織からなるコミュニティである。

複数組織からコミュニティを形成する場合，複数組織の利用者が集まり，コミュニティを形成する。例えば，各大学の学生が集合したサークルを考える。サークルの加入者は同一サークルに属するが，利用者の所属する大学（組織）は，複数存在する。

### 3.2. コミュニティにおける認可手法

コミュニティ内での情報共有実現に使う認可手法として，Access List, User Class, 属性認可を考察する。ただし，認可手法の考察において，Access List，User Class はどちらも一覧で表現できるため，Access List と User Class をまとめて「Group」と記述する。Access List や User Class による認可を「Group 認可」と呼ぶ。

### 3.3. 組織数・規模に応じた認可手法の考察

コミュニティの組織数・規模により場合分けを行い，それぞれ場合における認可手法を検討する。①単一・大規模，②複数・大規模，③単一・小規模，④複数・小規模の順に検討する。

表 1 コミュニティの組織数・規模による場合分け

| 規模 | 単一組織 | 複数組織 |
|---|---|---|
| 大規模 | ①単一・大規模 | ②複数・大規模 |
| 小規模 | ③単一・小規模 | ④複数・小規模 |

#### 3.3.1. 単一組織・大規模コミュニティ

単一・大規模なコミュニティは階層を形成する特徴を持つため，Group 認可は困難である。Group による認可手法は一覧形式で利用者を管理するため，認可の際に検索が遅くなることと，メンバー管理が煩雑化するためである。階層が多いほど，作成・追加・削除等の管理は煩雑化し，検索に時間を要する。そのため，大規模コミュニティにおいて，Group による認可は困難である。

Group による認可では，階層化したコミュニティを表現するために,コミュニティごとに一覧を作成する。例えば，利用者が理学部物理学科に属することを Group により表現する場合，理学部の Group と物理学科の Group の二つを作成する。さらに，利用者情報を変更する際には，利用者の関係するコミュニティの Group も変更しなければならない。例えば，ある利用者が理学部物理学科から工学部電気情報工学科へ異動する場合，利用者名を基に，理学部，物理学科，工学部，電気情報工学科の Group を編集しなければならず煩雑である。

これに対し属性認可は，RDB（Relational Database）が保持する表の属性で利用者の属する各階層のコミュニティを表現できるため，階層を形成する大規模コミュニティに適している。一般的な会社などの事業組織では，所属構成員を管理する人事 DB を保持しており，その組織に下位部局があれば，各構成員の所属情報が正確に保持されている。また，最近では利用者認証のために LDAP サーバーへ利用者情報を登録する場合が多い。このような人事 RDB や LDAP サーバーを参照することで，各階層のコミュニティに属するかどうかを判断できる。例えば，理学部物理学科に属する利用者のみを認可する場合,所属部局の属性を参照し，利用者の学部属性が「理学部」で，学科属性が「物理学科」であるか否かを判断すれば良い。

さらに，RDB や LDAP サーバーは，検索や参照が高速であるため大規模コミュニティに適している。そのため，単一組織・大規模コミュニティの認可手法には属性認可が有効である。

#### 3.3.2. 複数組織・大規模コミュニティ

複数組織・大規模コミュニティは，複数組織の利用者が集まり，階層コミュニティを形成する。例えば，A 大学と B 大学が連携し，両大学の理学部を一つのコミュニティとする場合である。階層コミュニティを形成するため，単一組織・大規模コミュニティと同様，Group による認可は不適切である。従って，属性による認可を行うことになる。

複数組織の間で属性認可を実現するには，属性項目の標準化や共通化が課題となる。単一組織の場合と同様に，複数組織での認可も指定した階層の利用者を認可する。各組織はそれぞれ独自に階層を形成しているため，属性項目を組織間で標準化・共通化していない場合は，属性認可のために複数の異なる階層コミュニティを指定するが，コミュニティは複数の組織で構成されるため，複雑な指定が必要である。また，例外的な属性に対する認可は難しい。

例として，複数の大学が単位互換協定を結び，同系統の学科所属者に授業コンテンツを提供しあう場合を考える。単位互換を行う各大学がすべて同じ組織階層を持っていれば単純であるものの，現実には大学毎に部局名は異なっている。A 大学には理学部物理学科があっても，B 大学には理工学部応用物理学科しかないかもしれない。つまり，大学ごとに属性を指定する必要がある。

サービスやコンテンツを提供する側で属性認可を

行うためには，LDAP 認証サーバー内の属性項目のパスや，RDB の属性名を指定する必要がある。認証サーバーが数個であれば手作業で指定できるかもしれないが，認証サーバーが多くなると指定は困難である。サービスやコンテンツを提供する側で，利用者の属性による認可制御を効率化するためには，認証サーバー側での利用者属性データ格納の構造を標準化・共通化する必要がある。しかし，既存組織は組織の内部構造が決まっているため，標準化は難しい。

### 3.3.3. 単一組織・小規模コミュニティ

小規模コミュニティにおいて属性認可は適していない。多くの小規模コミュニティは下位の階層コミュニティ（属性）を持たないし，また持つ場合でも利用者数は少数である。そのため，大規模情報への高速な参照は必要でない。

単一・小規模コミュニティでは，Group による認可手法の効率が良い。階層コミュニティを表現する必要がないため，整合性を気にせず，利用者情報の変更ができる。また，階層コミュニティを含む場合にも，利用者数が少ないため，変更処理を短時間に抑えることができる。利用者数が少ないため，検索時間は短時間である。

### 3.3.4. 複数組織・小規模コミュニティ

複数組織・小規模コミュニティは複数組織の利用者が集まるコミュニティである。複数組織・小規模の例としては，各大学の学生で集まったサークルが考えられる。単一組織・小規模コミュニティと同様に，階層コミュニティを含まず，含む場合も利用者は少数である。そのため，単一・小規模コミュニティと同様に，Group による認可手法が適している。

複数組織の間でコミュニティを形成する場合，コミュニティの管理方法が曖昧になる。そのため，複数組織間で一つのコミュニティを管理する仕組み（コミュニティ管理サービス）が必要である。

## 3.4. 既存コミュニティを用いた認可機構

コミュニティが小規模な場合，Group による認可が適切である。一方，情報サービスのために，Group の管理機構を新たに構築することは煩雑である。現在までに，様々な情報サービスで Group を管理維持する機構が構築されてきた。例えば，SNS では Group の作成・削除，メンバーの追加・削除などを管理する方法が提供されている。WebCT などの e ラーニングシステムでも，受講者というグループを管理する方法が提供されている。電子メールでも，メーリングリストというグループ管理機構が存在している。本研究が対象とするグループ認可機構においても，既存のグループ管理機構を用いる方が容易にシステムを構築できる。

3 章で説明したように，近年 Web 系の情報サービスでは SSO が普及している。つまり，Web 上の情報サービスでは，認証作業を共通化する基盤が整ってきている。そこで，まず Web SSO の基盤の上で，複数の情報サービス連携を実現する。次に，サービス A での認可機構に，サービス B のグループを用い，既存のコミュニティ管理サービスを使用した認可機構の実現を試みる。

# 4. 試作システムの開発

## 4.1. 試作システムの概要

本研究では，コミュニティ認可機構を組み込んだシステムを試作した。試作システムでは認証機構として Shibboleth を用いる。また，認可機構に用いるコミュニティ管理システムも試作した。各システムのソフトウェア構成などを表 2 に示す。

表 2 試作システムと関連するシステムの構成

| システム | ソフトウェア環境 |
|---|---|
| IdP<br>（認証機構） | OS: CentOS 5.2<br>IdP 構築用ソフト: Apache2.2, Tomcat5.6, Shibboleth2.0 |
| 試作コミュニティ管理システム（認可機構） | OS: FreeBSD 6.2<br>Web: Apache 2.2<br>開発言語: Ruby |
| SP<br>（リソース提供） | OS: CentOS 5.2<br>SP 構築用ソフト: Apache, Tomcat, Shibboleth SP |

利用者が SP の Shibboleth により保護されているリソースにアクセスをすると，Shibboleth の機能により，IdP へリダイレクトされ認証を行う。その後，コミュニティ管理システムにリダイレクトされ，利用者自身が所属するコミュニティのリストを得て（Cookie として保存），SP へリダイレクトされ，最初にアクセスしたリソースへ，再度アクセスを試みる。リソースを公開する対象となるコミュニティに利用者が属している場合は，リソースがユーザーへ提供される。認可処理の流れを図 1 に示す。

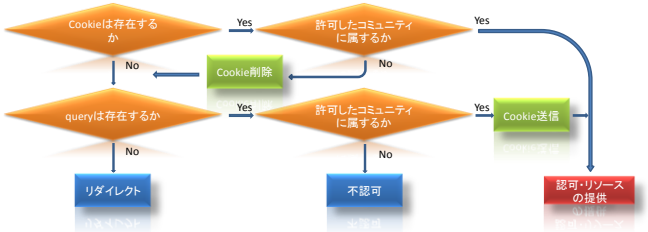


図 1 コミュニティ認可処理の流れ

## 4.2. 試行と評価

### 4.2.1. 試行

実際に試作システムを稼動させ，認可の実験を実施した。図 2 に Shibboleth 認証画面，図 3 にリダイレクトの画面，図 4 と図 5 に認可成功画面を示す。

図 2 Shibboleth 認証画面

図 3 リダイレクト画面

図 4 認証成功画面

図 5 Cookie による認証成功画面

### 4.2.2. 試作システムの評価

試作システムでは，利用者がコミュニティに属するか否かでリソースへのアクセス制御ができることを確認できた。本システムでは，Cookie に関する課題が残されている。試作システムでは，利用者が目的の Cookie を保有する場合，認可判断を行う。Cookie の有効期限を短時間に設定し，利用者がコミュニティから退会した後に誤ってサービスを許可する問題に対応している。Cookie の有効期限が短いために，次回の認可時もリダイレクトと探索の作業が必要となり，退会した場合でも，利用者の属するコミュニティを判別することができる。しかし，利用者が取得した Cookie 情報を改ざんするなどで不正に利用し，コミュニティに属さずサービスを利用するという課題を解決しなければならない。

## 5. おわりに

本論文では，分散 SSO 機構を利用し，コミュニティに属する利用者のみを対象に情報リソースを提供するコミュニティ認可機構を提案した。そして，組織間連携の上でコミュニティ認可機構を実現するシステムを試作した。Group の概念を取り入れることにより，所有者や所属コミュニティなどにより，認可判断が可能となる。

今後，コミュニティ管理者としてのメンバーの設定，コミュニティ内における権限委譲，コミュニティ外メンバーの追加，一部のコミュニティメンバーの認可拒否について検討を進め，適切に認証と認可ができるシステムを目指す。

## 文　献


[1] 伊東　栄典，全学共通認証事業室：“九州大学全学共通認証基盤と全学共通 ID「SSO-KID」の紹介”，九州大学情報統括本部 IT マガジン Vol.1，No.2，pp.42-48，2007

[2] Web サービスのセキュリティ -@IT：http://www.atmarkit.co.jp/fsecurity/rensai/webserv04/webserv01.html

[3] SAML とは -IT 用語辞典：http://e-words.jp/w/SAML.html

[4] OASIS ：http://www.oasis-open.org/home/index.php

[5] Web サービスと Web サービスのセキュリティ：http://www.soi.wIDe.ad.jp/class/20030038/slIDes/74/index_17.html

[6] シングルサインオン -キーマンズネット：http://www.keyman.or.jp/3w/prd/69/30002269/

[7] Understanding Shibboleth ：https://spaces.internet2.edu/display/SHIB2/UnderstandingShibboleth

[8] 甲斐啓文：セキュアな協調作業空間のためのグループ管理システムの設計と構築 ,2005

[9] LDAP とは 【Lightweight Directory Access Protocol】 - 意味・解説 ： IT 用語辞典：http://e-words.jp/w/LDAP.html