L-052

認可を必要とするシングルサインオンシステム
Single sign-on system requiring authorization

牧野 浩之 † **廣安 知之** ‡ **三木 光範** ‡
Hiroyuki MAKINO Tomoyuki HIROYASU Mitsunori MIKI

1. はじめに

インターネットが普及し，さまざまなサービスがインターネットを通じて提供されている．これらのサービスを利用するユーザはサービスごとにユーザ ID やパスワードなどのアカウント情報を持っており，利用する際にはサービスごとに認証を行う．しかし，利用するサービスの増加に伴い，多数のアカウント管理に悩むユーザも多い．そこで，一度の認証で複数のサービスが利用できるシングルサインオンシステムが必要となってくる．最近では企業や研究室などの組織内においても様々なイントラ向けサービスが提供されている．イントラ向けサービスに対してもシングルサインオンシステムは有用である．

一方で，組織内では管理者と利用者，上司と部下，教員と学生といった関係が存在し，ユーザにサービスを利用してもらう前にしてもらいたいタスクが存在する．既存のシングルサインオンシステムは一般ユーザを対象としたものばかりであり，組織内での関係性を認証に利用するというものはなかった．組織内においては，ユーザごとに特定のタスクを課し，そのタスクを完了させた場合にサービスへのアクセスを許可したいといったニーズがある．

そこで，組織内での関係性をもとに，シングルサインオンシステムとタスク管理システムを組み合わせ，ログイン時に課されているタスクをユーザが完了させないとサービスにアクセスできないようにする仕組みを提案する．

本研究では，シングルサインが可能な認証サービスとユーザごとにアクセス可能なリソースをタスクに応じて制限できる認可機構を持ったシステムを構築する．これにより，アカウント情報の集中管理ができるだけでなく，ユーザに特定のタスクを課してタスクを完了させた場合にのみユーザにサービスの利用権限を与えるといった利用が可能になる．

2. シングルサインオン (SSO: Single sign-on)

シングルサインオンとは，1 回の認証手続きで，複数のサービスやリソースなどにアクセスできること，またはそれを実現するための機能を表す．シングルサインオンによってユーザは複数の ID やパスワードを覚えておく負担から解放される．パスワードを 1 つ覚えておくだけで，厳格なパスワード管理も現実的なものとなり，より高いセキュリティを実現することが可能になる．ユーザは各種のサービスにアクセスする際に，認証画面で一度だけログインを行えば，許可されている全てのサービスが利用できる．また，システム管理者やアプリケーションの開発者はパスワードなどの認証情報の管理を一元化することで，複数の認証情報を管理したりアプリケーションごとに認証機能を開発する負担から解放される．すなわち，シングルサインオンを導入することにより以下のこと実現できる．

- 利用者の負担の軽減（簡略化）
- より安全な認証機能の実現（厳密化）
- システム管理者やアプリケーション開発者の負担軽減（共通化）

3. 認可 (Authorization)

認可とはユーザごとにアクセスできるリソースの検証を行うことである．実際のアプリケーションでは，認証後利用者の属性や資格によってアクセスできるサービスやサイトを制限したり，与えられたアクセス権限によってリソースへのアクセスを制御することが必要となってくる．これが認可である．ここでは，アクセスする権限を付与する条件として，管理者が課したタスクを，サービスにアクセスしようとするユーザが完了させたときとする．

4. 認可を必要とするシングルサインオンシステム

4.1 概要

今回構築した認可を必要とするシングルサインオンシステムは Web ブラウザ上でサービスへの認証および権限の管理ができ，利用するユーザに管理者が設定したタスクを課すことができるシステムである．システムのイメージを図 1 に示す．このシステムを認証サービスとして利用することにより，特定のユーザに対し特定のタスクを課し，そのタスクが完了した場合にのみユーザにアクセスする権限を与えるという利用が可能となる．

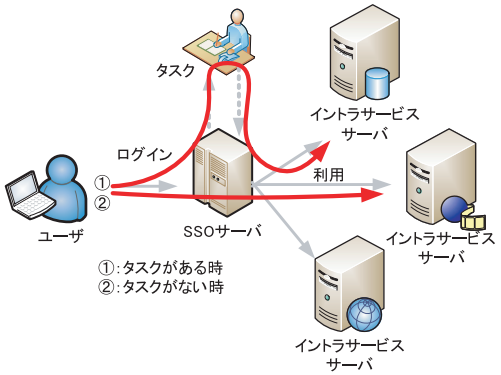

図 1: システムイメージ

†同志社大学大学院 工学研究科
‡同志社大学 工学部

4.2 利用シーン

私の所属する研究室を例に利用シナリオを述べる．本研究室には，イントラブログリーダやソーシャルブックマークなど多数の研究室内ツールが提供されている．本システムを利用することにより，例えば，年度が変わり住所録を更新したユーザのみイントラブログリーダの利用を許可することや，アンケートに答えた場合にのみサービスを継続して利用できるなどの認可が可能となる．また，教員が学生に対して研究の進捗状況を確認する場合にも質問をタスクとして課すといった利用が可能である．

4.3 利用方法

本システムを利用する際の流れを図 2 に示す．まず始めに管理者がユーザのアカウントを登録しておく．ユーザは通知されたアカウント情報をもとに本システムのログイン画面にアクセスし，認証を行う．ログイン完了後，メニューが表示される．ここで管理者がユーザに対し特定のタスクを課していれば，タスクを完了させなければサービス利用権限が与えられない旨の表示を行う．そこで，ユーザはアンケートや情報更新など管理者が課したタスクを完了させる．タスクが完了できれば，利用可能サービスのメニューが表示され，コンテンツやサービスにアクセスが可能となる．

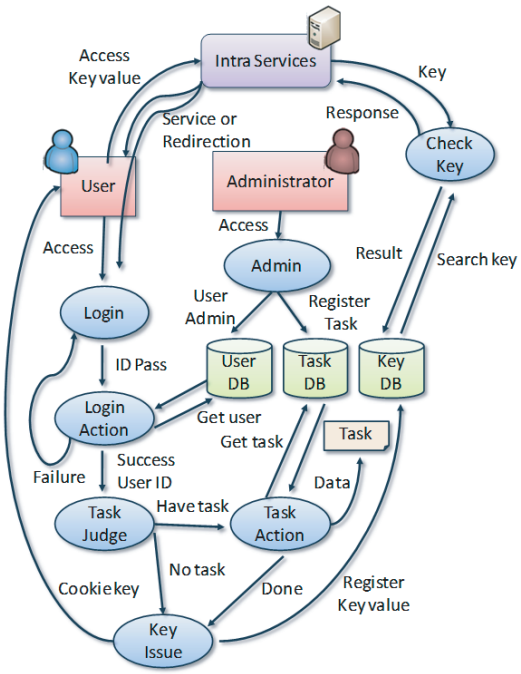

図 2: システム利用の流れ

4.4 機能

本システムには，従来から様々なシステムで提供されているシングルサインオンの機能だけでなく，サービスへのアクセス権限を設定できる認可の機能を持ち合わせている．認可は管理者が課したタスクの状態によって行われる．管理者が課すタスクは管理画面よりユーザごとに設定ができる．タスクにはアンケートや同意を求める質問などを自由に記述することが可能となっており，また集計も行える機能も提供する．これにより，認証管理とタスク管理が一つのシステムで同時に行えるようになる．

4.5 システム構成

本システムは以下の言語，サーバを利用して構築した．

- 開発言語:PHP (Ethna フレームワーク)
- データベースサーバ:MySQL
- Web サーバ/モジュール:Apache (mod_rewrite)
- 動作 OS:Linux, UNIX, Windows など

本システムの動作メカニズムについて述べる．ログイン画面でユーザ認証を完了させると，ユーザに対し一定時間有効なキーを発行する．キーはサーバ上のデータベースとユーザクライアント側の Cookie に記憶され，リソースにアクセスする際に Web サーバがキーを要求し，キーをもとに認証が完了しているか，また，アクセス権限があるかをデータベースと照合して確認し，それにパスできない場合は認証画面へ mod_rewrite でリダイレクトする．キーの確認は mod_rewrite で認証プログラムにフックすることによって行う．そのため，シングルサインオン機構を利用するサービスへの適用が比較的容易に行える．本システムを利用して，サービスにシングルサインオン機構を導入する一番簡単な方法は，認証を適用するディレクトリに.htaccess ファイルを置き RewriteEngine を有効にして，Cookie に保存されたキーをもとにデータベースに照会を行い，キーが有効でないならログイン画面へリダイレクトするように設定するだけである．

5. 今後の展望

今回構築したシステムでは，ユーザアカウントの管理はデータベースを利用して行っているが，今後は，ユーザアカウントの管理に適したディレクトリサービスである LDAP との連携を行い，アカウントの統合管理に対応したい．また，タスクの管理においては各ユーザの処理状況や進捗状況を管理画面で閲覧できるようにし，グループウェア要素を高めることで組織での生産性向上に寄与していきたい．

6. まとめ

オンラインサービスの増加に伴いシングルサインオンへの需要は高くなってきている．企業や組織でのイントラ向けサービスにおいてもシングルサインオンの導入が進んできている．一方で，組織内においては管理者が利用者に対し処理してもらいたいタスクが存在し，ユーザごとにタスクを課したいというニーズがある．そこで，本提案システムでは，組織内での関係性を認証に反映できるようシングルサインオンシステムとタスク管理システムを融合する．ユーザがログインしてサービスを利用する際にタスクを提示し，条件を満たした場合にアクセス権限の付与が行われる認可機構を持たせたシステムを構築した．

参考文献

[1] @ IT：なぜ「シングル・サインオン」が必要なのか？，
http://www.atmarkit.co.jp/fnetwork/rensai/dirt01/01.html