古い認証システムの捨て方

細川達己　金子康樹

慶應義塾インフォメーションテクノロジーセンター本部

{hosokawa,yasuki.kaneko}@keio.jp

How to Throw Away a Legacy Authentication System

Tatsumi Hosokawa, Yasuki Kaneko

Information Technology Center, Keio University

概要

慶應義塾においては、約 10 年前に構築した keio.jp という独自実装の全学基盤認証システムが存在し、その上で 50 以上のアプリケーションが動作していたが、クラウドメールの利用のため、この認証システムを Shibboleth ベースのものに置き換えることを決定した。

その後、認証システム自体の切替えから約 2 年半をかけて、ユーザにあまりシステムの変更を意識させることなく移行作業を続けており、現在はその最終段階にある。

本稿では、独自実装から Shibboleth を用いた認証方式への移行をする際の留意点や課題について、実例を挙げながら考察する。

1　はじめに

慶應義塾では、2002 年に開発を開始し、2005 年に本格運用が始まった、keio.jp と呼ばれる独自仕様の全学基盤認証システム（以下、旧 keio.jp）が存在しており、これで認証を行う 50 を越えるアプリケーションが存在していた。

しかし、運用開始から 10 年近くを経て、特に独自仕様である点が既存のパッケージや SaaS のクラウドサービスを導入する際などに大きな弊害となっていた。

そこで 2014 年、全学メールシステムを、本学向けにカスタマイズされた DEEPMail ベースのシステムから、Google Apps for Education（現在 G Suite for Education）に移行するにあたり、認証システムを、学術機関の認証システムとして広く利用されている Shibboleth による、SAML2 ベースの認証システム（以下、新 keio.jp）に移行することとした。

そして本年度末である 2017 年 3 月までを移行期間と定めて、既存のアプリケーションの全面移行を行うこととした。

本稿では、移行時に特に問題になった点とその解決策について、具体的に紹介していく。

2　旧 keio.jp のシステム

旧 keio.jp は、独自の実装がされたもので、次のような仕組みで認証が行われるものであった。

1. ユーザを ID とパスワードで認証するとメニュー画面（ポータル）が表示される。
2. その画面からアプリケーションに遷移するときに一時的なランダム文字列がアプリケーションに渡される。
3. アプリケーションは、渡されたランダム文字列を認証システムに対して照会し、ユーザの属性などを得る。

ユーザ ID のフォーマットは、「ユーザが好きに付けられる文字列@サブドメイン.keio.jp」という形式であり、サブドメインはシステム側でランダムに割り振られるようになっていた。たとえば「keio-taro@a2.keio.jp」のような形式である。これが同時に導入された全学メールのメールアドレスとなっていた。

ユーザの認証情報や属性は SQL のデータベースに保存され、認証や属性の取得、アカウント初期化などの API は、データベースのストアドプロシージャとして実装されていた。アプリケーションはこれらの API を Web サービス経由でアクセスするように定められていた。

このような仕組みで認証を行うため、ポータルを経由しないページ遷移では、シングルサインオンができない、などといった問題も存在していた。

3　アプリケーションの移行

先述の通り、認証システムの移行を検討してい

る時点で、様々な部門が開発したものを含む 50 を越えるアプリケーションが旧 keio.jp 上で動作していた。これら全てを、タイミングを合わせて一気に新システムに移行することは技術的に困難であり、また大きなリスクでもある。

そこで、移行に関しては次の方針を立てた。

- 旧 keio.jp のコアのデータベースと各種 API は当面の間機能を維持する。
- 旧 keio.jp のログイン用 API と、旧アプリケーションに画面遷移するポータル機能の一部を切り出して、Shibboleth SP として再実装する。このシステムによって、旧アプリケーションはあたかも新 keio.jp で認証しているかのように動作する。
- ある程度（2017 年 3 月まで）の移行期間を持たせて、その間に原則全アプリケーションをネイティブな Shibboleth SP に移行してもらう。

この移行のための互換レイヤは、旧 keio.jp のログイン用 API のストアドプロシージャのソースを読み、それを Shibboleth SP 上のスクリプトとして再実装することで実現した。スクリプトは Shibboleth SP による認証を信頼し、旧 keio.jp 本来の認証プロセスをスキップして、偽の認証済みランダム文字列を旧 keio.jp のデータベースに書き込む。この処理が正常に終了した後、生成されたランダム文字列を自動的に POST しながら、ブラウザを旧アプリケーションに遷移させる。

全ての旧アプリケーションは、この偽データによって問題なく動作することが検証できたため、旧アプリを十分な期間を設けて移行作業を行うことが可能となった。

さらに、ポータルを経由しないページ遷移ではシングルサインオンが出来ないという問題点も、この機構によって解決することができた。

4　ユーザ名体系の移行

メールシステムを Google Apps for Education（現在 G Suite for Education）に移行するにあたって、旧 keio.jp のユーザ名体系が、データの共有などにおいて弊害となる可能性が、この移行作業を行っていたサイオステクノロジー社[1]から示された。

具体的には、「ユーザが好きに付けられる名前@サブドメイン.keio.jp」の「サブドメイン」の部分である。Google Apps の設計上、1 つのテナントは 1 つのドメイン内に収まるように設計することが自然であり、ランダムに割り振られるサブドメインは、今後未知の弊害を起こす可能性も考えられた。

また以前から、このランダムに割り振られたサブドメインはユーザから不評であり、このサブドメイン部分を忘れたので教えてほしいという問い合わせがヘルプデスクに頻繁に寄せられていた。

以上のような理由から、新 keio.jp においては、プライマリなユーザ名体系を「ユーザが好きに付けられる文字列@keio.jp」という、サブドメインを省く形に設定することとなった。

ただし、先述の通り、旧 keio.jp を前提としたアプリケーションは残り続けるため、旧形式のユーザ名もシステム内部に持ち続けることとした。また、システムの移行後にアカウントを取得したユーザは、新システムに対応したユーザ名しか認識していないが、システムの内部には旧形式のユーザ名を持ち続けることになっている。

逆に、既存ユーザ名に関しても新形式のユーザ名を割り当てることが必要であったため、次の方針で割り当てを行った。

- @の左側の文字列が、全サブドメイン間で重複していない場合は、そのままサブドメインを外した文字列を新形式とする（旧形式が「keio-taro@a2.keio.jp」であれば、新形式は「keio-taro@keio.jp」となる）。
- サブドメイン間での重複がある場合は、「ユーザの割り当てた文字列.サブドメイン名@keio.jp」を新形式とする（旧形式が「keio-taro@a2.keio.jp」であれば、新形式は「keio-taro.a2@keio.jp」となる）。
- 割り当てられた新形式ユーザ名を、一度だけ好きな文字列に変更することができるシステムを一定期間提供する。

ログイン ID としては新形式・旧形式のどちらでも利用できるようにした。これは Shibboleth IdP のバックエンド LDAP 内に、新旧両形式を uid として持つ 2 つのエントリを用意することで実現した。新旧両形式のユーザ名は、uid とは別のユーザ属性として、ログインに利用した uid の形式にかかわらず、Shibboleth SP 経由で同一の値を取得することが可能であり、アプリケーションは uid ではなくこれらの属性値をユーザ名として利用する。この仕組みによって、どちらの形式でログインし

ても同一ユーザとしてアプリケーションからは扱われることになる[2]。

メールシステムに関しても、旧形式のメールアドレスをそのまま、受信先アドレスとしてだけではなく、Web メールの送信元アドレスとしても使い続けることができるようにシステムを構築した。

5　パスワードハッシュの移行

旧システムは独自実装であったため、パスワードハッシュも独自形式のものであった。しかし、Shibboleth IdP で認証させるには、バックエンドの OpenLDAP が BIND（認証）可能なパスワードハッシュ形式である必要があるため、平文パスワードを保存していない状態では簡単には移行することはできない。

幸運なことに、学術認証フェデレーション（以下、学認）に正式加入することにした 2013 年に、将来的な Shibboleth での認証に備えて、旧 keio.jp のログイン API に関するストアドプロシージャを改修し、正常にログインがなされた場合には、パスワードとして入力された文字列を OpenLDAP で BIND 可能な形式に再ハッシュ化し、データベースに格納するようになっていた。

この形式で 1 年以上システムを運用し続けたことで、その間に一度でも旧 keio.jp にログインしたユーザに関しては、OpenLDAP で BIND 可能なハッシュが取得されていた。

そのため、ほとんどのユーザに関しては、自動的に新 keio.jp にパスワードを引き継ぐ事が可能となった。それ以外のユーザに関しては、窓口で通常のパスワード再発行の手続きを行うこととしたが、対象が少なかったこともあり、大きな混乱は生じなかった

6　プロビジョニングシステムの移行

旧 keio.jp は、構築時にあまりユーザプロビジョニングについて深く考えられていなかったため、学事や人事のデータベースから提供されたデータを元に、非常に複雑な手順で反映される、独自のシステムで運用されていた。

この反省から、2011 年に導入された「ITC システム」という、学内設置 PC を利用する際に必要となるアカウント等のサービスを提供しているシステムでは、よりシステム化されたユーザのプロビジョニングが実現され、属性データもより実用的な形式で格納されるようになっていた。

そこで新 keio.jp においては、この「ITC システム」のデータを元に、ユーザや各種属性のプロビ

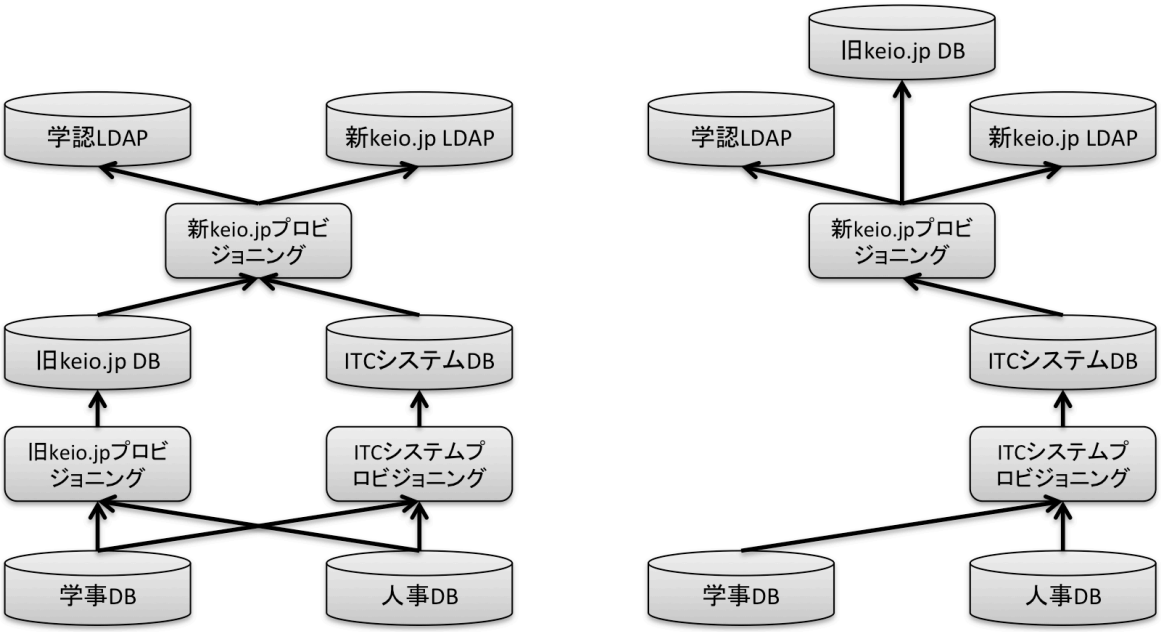

図 1. keio.jp 用プロビジョニングシステムの移行（手順 2.～3.の段階）

ジョニングを行う事を決定した。

一方で、keio.jp のユーザ名やパスワードの情報は、旧 keio.jp システムにしか存在しなかったため、次のような手順でプロビジョニングシステムを移行することとした。

1. 慶應義塾大学の学認 IdP 用のプロビジョニングシステムは、旧 keio.jp と ITC システムの双方からデータを得て、Shibboleth IdP のバックエンドの OpenLDAP にユーザプロビジョニングを行っていた[3]。新 keio.jp は Shibboleth IdP を利用することとなったので、まずはこの学認のプロビジョニングシステムをベースに新 keio.jp のための改修を加えることとした。
2. この新 keio.jp 用プロビジョニングサーバに対して、学認には不要ではあるが新 keio.jp に必要な属性や、様々な関連データを扱えるように改修を行い、その結果を新 keio.jp 用 LDAP にプロビジョニングするようにした、この LDAP をベースとする Shibboleth IdP を、新 keio.jp として 2014 年 11 月に公開した（図 1 左）。
3. さらに改修を行い、旧 keio.jp の DB も新 keio.jp 用プロビジョニングシステムがプロビジョニングを行うようになり、データの流れが単純化された。この変更は 2015 年 3 月に行った（図 1 右）。
4. 2017 年 3 月に、全ての旧アプリケーションの Shibboleth SP 化が完了した時点で、不要となる旧 keio.jp のデータベースを廃止する予定となっている。

このようにして、学認用のプロビジョニングシステムをベースに、スムーズに新 keio.jp のプロビジョニングシステムを構築することができた。

現在は、最終段階の手順 4.の準備中である。

7 ログイン・ポータル画面の移行

旧 keio.jp においては、ログイン画面とポータル画面は同一の Web アプリケーションとして実装されていたが、これをほとんど変わらないデザインで、ログイン画面は Shibboleth IdP に、ポータル画面は新しいポータルアプリケーション（Shibboleth SP）に分離した。

旧 keio.jp のログイン画面をブックマークしていたユーザのために、旧 URL は新ポータルアプリケーションへのリダイレクトを設定した。さらに、通常はブックマークしてもうまくその後のログインができない Shibboleth IdP のログイン画面をブックマーク可能にするハック[4]を加えることで、ユーザが旧 keio.jp の時と同様に、自然な手順でログイン画面をブックマークできるようにした。

8 今後の課題

その他にも多くの課題・問題があったが、無事に本学全学基盤認証システムを。独自システムから Shibboleth ベースのシステムに、比較的スムーズに移行することができており、周辺アプリケーションの移行も順調に進んでいる。

現在直近の課題としては、作業のベースとなった学認の IdP を廃止し、新 keio.jp の中に学認 IdP を取り込むことであるが、これは次のような要因で細心の作業が必要であると考えている。

1. 学認 IdP の entityID の変更を含む作業となる（ePPN 等の属性値は変化させない）。
2. 学認のポリシーに合わないユーザの属性を学認 SP に送信しないように設定することが必要である。

また、多くのユーザやアプリケーションに利用されている基盤システムであっても、スムーズに更改することが可能であったというこの事例を、学内に多く残る他のレガシーな基盤システムの更改に向けて役立てていけたらと考えている。

参考文献

[1] サイオステクノロジー株式会社「サイオスが開発したワンタイムパスワード認証システム、慶應義塾の認証基盤に採用 ～ Shibboleth（シボレス）認証用ワンタイムパスワードモジュールとワンタイムパスワード用秘密鍵発行システムを提供～」, http://i.sios.com/news/press/20141022-otp.html, 2014

[2] 慶應義塾 ITC 本部技術メモ「Shibboleth で複数の ID を同一人物としてログインさせる」, http://memo.itc.keio.ac.jp/blog/?p=101, 2014

[3] 慶應義塾 ITC 本部技術メモ「プロビジョニングの道具としての OpenLDAP back-sql」, http://memo.itc.keio.ac.jp/blog/?p=88, 2014

[4] 慶應義塾 ITC 本部技術メモ「Shibboleth のログインページをブックマーク可能にする」, http://memo.itc.keio.ac.jp/blog/?p=416, 2014