A-Cloud Mail と GoogleApps を併用する大学情報インフラ

深町賢一*

2012 年 11 月 09 日

概 要

本学では、メールシステムの更改にあたり、認証システムを抜本的に再設計し、CTC A-Cloud Mail と GoogleApps for Education 両方を利用可能とした。併用を選択するにいたった理由と A-Cloud Mail への Provisioning システムの実装について紹介する。

目 次

1 切替前のシステム概要 1
1.1 切替前のシステム概要 1

2 メールサービスの更改 2
2.1 ユーザからの要望 2
2.2 運用側の要望 2
2.2.1 サポート体制 2
2.2.2 システム拡張と認証システム . 3
2.2.3 ネットワーク設計上の注意 . . . 3
2.3 GoogleApps の是非 3
2.3.1 GoogleApps 歓迎の論理 3
2.3.2 GoogleApps 拒否の論理 4
2.4 二つのクラウドを併用する案 4

3 実装 5
3.1 ID 管理システム 5
3.2 A-Cloud との連携 6

4 まとめ 6

*千歳科学技術大学グローバルシステムデザイン学科、〒 066-8655 北海道千歳市美々758-90、mailto:k-fukama@photon.chitose.ac.jp http://www.nsrg.fml.org/

1 切替前のシステム概要

本稿では、2012 年 4 月に切替をおこなった千歳科学技術大学におけるメールと認証システムの更改について述べる。

最初に、切替前における大学情報システムの概要についてまとめておく。

1.1 切替前のシステム概要

以下のシステム群は大学情報センター管轄の全学共用部分である。全システムは共通の学内認証システムを参照しているが、いわゆるシングルサインオン (以下、SSO) は実現されていない。

- ネットワークの基本構造は民間企業準拠。ファイアウォールがあり、学内、DMZ、インターネットを分離している。

  これは、開学時、すでに商用インターネットの時代だったためである。

- PC 教室のユーザ端末

  PC 教室は、実習・演習等で用いることを想定した全学利用のシステムである。初年時教育、各学科でのプログラミング実習等で用いられる。

  オペレーティングシステムは Windows と CentOS の dualboot になっている。ホームディレクトリはファイルサーバにあり、Windows と CentOS 両方から読み書きできる。

- 授業支援サービス

  授業支援のためにポータルシステム (以下ポータル) が運用されており、授業の予定、連絡、スケジュール等が利用できる。

e-Learning システム (以下 EL) は、大学独自開発のシステムで、本学独自開発のコンテンツを無料で広く社会に公開している。地域貢献の一貫として、小中高校との連携も広く行なわれている。

- 電子メール

  全教職員と全学生が利用可能で、メールアドレスは共に photon.chitose.ac.jp ドメインである。

なお、サーバ群は学内にあり、ハードウエア保守は原則オンサイトである。

上述のシステムは同じ学内認証システムを参照している。つまり、同じログイン名とパスワードで、PC 教室の Windows と CentOS、ポータル、EL、メールシステムなどを利用できる。

一方、学外 (ファイアウォールよりインターネット側) とは連携していない。学外側にあるサーバは学内の認証システムとは独立した別のシステムである。

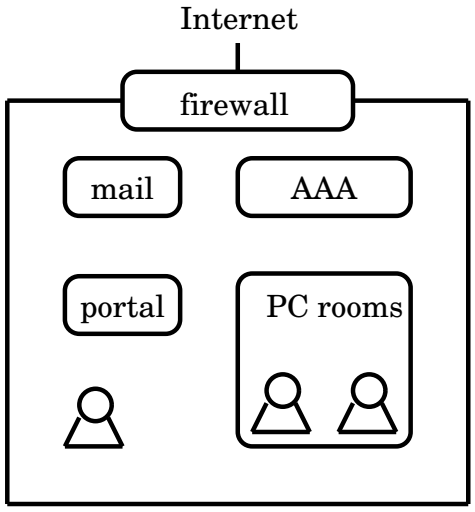

図 1: 切替前のシステム概念図: 学内 (キャンパス) ネットワークとインターネットは、ファイアウォール (firewall) で区切られている。PC 教室 (PC rooms) やポータル (portal)、メール (mail) など各サービス群は認証システム (AAA) へ問い合わせる。注: なお、この図は概念の紹介であり、詳細なネットワーク構成とは必ずしも一致しない。

2 メールサービスの更改

上述のように、学内におけるパスワード管理は統合されていたが、学外への認証系延長は技術上困難であった。しかしながら、クラウドが流行している昨今、コストメリットがあるなら、学外にあるサービスの利用も視野に入れておく必要があるだろう。

今回のメールシステム更改は抜本的な再構築のよいチャンスであった。そこで、メールサービスのクラウド化と認証システムの再設計を行ない、A-Cloud Mail と GoogleApps For Education を併用することにした。以下では、そこにいたる判断過程を概観する。

2.1 ユーザからの要望

一般ユーザからの要望は大分すると次の三点にまとめられるようだ。

- メール保存容量。

  メールスプールを大きくしてほしい。

- 使い勝手の継承。

  せっかく慣れたユーザインターフェイスを変更しないで欲しい。

- 使い勝手の改善。

  そろそろ他大学のように Gmail でいいのではないか？

保存容量は単にハードウエアの問題であるので対応が容易であるが、使い勝手/インターフェイスの問題は難しい。ソフトウエアのマイナーアップグレードであっても、ユーザインターフェイスに対する不満は、なにかしら出てくるものである。これについては、真摯なユーザサポート対応以外に解決策はない。Google の是非は 2.3 節で詳しく考える。

2.2 運用側の要望

運用側としては、サポート体制、そして将来拡張をみすえた認証システムの再構築が要望の主軸である。

2.2.1 サポート体制

専任のシステム対応用員はおらず、人手にも余裕はない。

ハードウエアはオンサイト保守であるが、いざという時に、オンサイト保守では対応時間がかかりすぎる。実際、PC 教室のファイルサーバ障害では、半日から一日授業にならないことが何度もあった。

さいわい、いままでメールシステムに大障害は起こっていなかったが、それは単に好運だったと思うべきである。メールにもファイルサーバと同様の障害が起こらないとはいえない。

本来であれば、インターネットサービスプロバイダ (以下 ISP) のサービスを契約したい。しかしながら、ISP のメールサービスは民間企業向けの仕様のものが多く、大学には向かない。

第一に価格帯が合わない。第二に年度末/年度はじめの一時的なユーザ数増への対応が難しい。卒業処理と入学処理は、きれいに年度末で切り替わらず、重なってしまう。大学の場合、百人、千人単位での一時的なユーザ数増になるため、ISP サービスの採用は困難である。

CTC A-Cloud Mail だけは、これらの要求を満たしているため、最大の切替先候補となった。しかしながら、既存の認証システムは学外との連携を想定しないため、認証システムの再構築をしないままでは、A-Cloud への切替えは運用の負荷を増やすだけになってしまう。

2.2.2 システム拡張と認証システム

ほぼどんな場合にも、コンピュータシステムに認証は必須である。それだけに、将来のシステムを想定することは難しい。たとえそうであっても、システム全体の設計者は、コストの許す範囲で、つねに最悪の事態を想定しておくべきである。

たいてい、システムは機能が追加されて拡張していくものである。常にシステムが一新されるのであれば問題はないが、一新されることは稀だ。

新しいシステムを追加する際に認証システムときちんと連動できるか？は、たとえ標準プロトコルを使っていても、毎回やってみるまで分からないものなのである。

分からない原因は、えてして「仕様を満たす最低限の設定や実装しか行なわれていない」からなのだが、それは納品する側からすれば資本主義経済下で当然の行動である。よって、システム全体の仕様策定者の将来構想が重要になるのだが、そうはいっても難しいものである。

この点において、システム拡張における認証系問題は「終らない悪夢」である。その「悪夢」を低コストで回避するためには再構築が必要だ。「銀の弾丸」はパスワードの保存形式にある。

認証システムのマスターデータにおいて「パスワードをハッシュ化して保存する」ということは Unix の伝統では正当な設計に見える。その一方、システム拡張の際には、足を引っ張る要素ともなっている。

追加する新システムと認証システムが連動できない場合、既存システム側の設定ファイルを一行変更すれば終りということは稀だ。「仕様を満たす最低限の設定や実装しか行なわれていない」システムは、大規模な変更を必要とする。追加費用も見込まなければならない。それ以上に保守の問題もあり、大規模の変更作業は難しい。

このような場合、認証システムとの間に PROXY を新規に作るほうが作業量が少ない。ただし、このシステムでは、そのシステム独自形式のデータを必要とすることも多い。このデータは平文パスワードを元に生成する必要があるが、パスワードのマスターデータがハッシュ化されている場合、対応しようがない。

つまり、平文パスワードを取り出せる下準備さえあれば、文字列変換するシステムの開発だけで、新規システムとの連動が容易に達成できる。この仕組みさえあれば、認証系を学外へ延長していくことも容易である。実際、後述する A-Cloud への Provisioning は、この仕組みの応用例である。

2.2.3 ネットワーク設計上の注意

民間企業と異なり、大学構内の多くは事実上屋外と同様のセキュリティレベルと考えなくてはならない。よって、学内からの攻撃も考慮し、学内に対してもファイアウォールを持つべきである。

当然、上述の「マスターデータをもつサーバ」や「平文のパスワードを各形式へ変換処理するサーバ群」は学内の最深部に設置し、対学内ファイアウォールで守

る。外部からのアクセスはもちろん、学内からのアクセスすら許すべきではない。

2.3 GoogleApps の是非

2.3.1 GoogleApps 歓迎の論理

費用 (GoogleApps For Education は無料) が最大の魅力であることは間違いない。

固定費だけでなく教育コストも低いと見積もられる。個人の Google アカウントで Gmail や Google カレンダーを利用しているユーザも多い。そのため、操作法について教えるコストがかからないと考えられるからだ。

もっとも、すでに個人で利用しているのなら、あえて GoogleApps For Education を導入する積極的な理由はない。

しかしながら、全学生が利用可能な情報システムとして GoogleApps を利用可能としておくことは有意義と考えた。

たとえば、Google Drive は最も利用価値のある例である。

従来より、実習等の課題の続きが自宅で出来ないという問題があった。これは PC 教室のホームディレクトリを学外からアクセスできないためだ。

Google Drive を利用すれば、この問題を解決できる。大学の PC 教室でやりかけの課題を Google Drive に保存し、自宅で続きをする。自宅でも Google Drive に保存し、大学で続きをするといった具合だ。

また、学生の書くレポートは、Google Drive (旧 Google ドキュメント) の Word や Excel、Powerpoint 互換機能で十分である。学部低学年で、自宅用の Microsoft Office を購入する必要などない。

サークルのサイトを作りたいということであれば、Google サイトが利用できる。独自サーバを運用したりする必要はない。

サークル内でカレンダーの共有も便利だ。

同窓会も同様の運用を行なえる (将来構想)。

このように、GoogleApps For Education を導入したい理由は、メール (Gmail) 以外のアプリケーションの利用にある。“学生” 向けに GoogleApps を利用できる環境整備は有意義である。大いに Google を歓迎するべきだろう。

だが、それは本当なのだろうか？

2.3.2 GoogleApps 拒否の論理

メールについては、以前より「そろそろ (他大学のように) Gmail でよいのではないか？」という意見が出ていた。しかしながら、筆者は、次のような理由により Google を採用しない。

第一に、サポートの品質。ISP に勤めていた人間から見て、Google のサポート体制は、お粗末である。無料だから利用する気になるのであって、有料ではありえない。もっとも GoogleApps For Education は無料なので、割り切って利用するという選択肢はあるだろうが、筆者は選ばなかった。

第二に、ログの問題。非常時にはシステムログの分析が重要であるが、Google ではログの取得が可能かどうかすらわからない。取得できたとして、その対応時間も不明である。もちろん Google のサーバに対しては国内法も適用されないので、何かあった場合には、どうにもならないだろう。

第三に、メールスプールの設置場所。これは情報の安全保障にかかわる問題である。たとえば、大学のメールには特許に関わる情報も入りこみ得る。確かに Google の既約にはプライバシー保護がうたわれている。だが、アメリカにあるハードウエアに対するプライバシー保護は、テロ捜査という印籠があれば簡単に反古にされるだろう。

よって、スプールを国外に置くことは考えられない。

2.4 二つのクラウドを併用する案

まとめると、「(特に) 学生向けに GoogleApps For Education を利用できる」体制を整えることは有意義であるが、法律的にも安全保障上もメールスプールは国内にあるべきである。

この矛盾した命題を解決するために特殊な構成も考えたが、運用を複雑にしては意味がない。コストさえ許せば、単純であるほうがよい。

よって二つのクラウドサービスを導入した。

- CTC A-Cloud Mail をメインのメールサービスとして採用した。

  すべての教職員と学生のメールサービスを A-Cloud へ移行した。つまり A-Cloud で photon.chitose.ac.jp ドメイン宛のメールを送受信できるよう設定した。

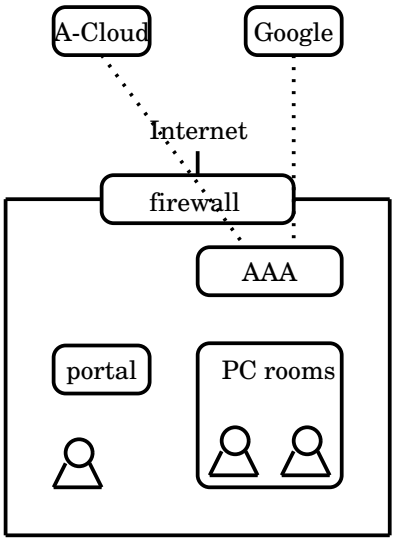

図 2: 新メールシステム概念図: 図 1 の学内にあるメールシステム (mail) が、学外サービスの A-Cloud に移行した。認証システム (AAA) は A-Cloud および Google と連携するように拡張された (図中点線)。PC 教室 (PC rooms) やポータル (portal) などは認証システム (AAA) へ問い合わせる。ここは従来と同じである。AAA と Google をつなぐ点線の端が責任分解点にあたる。

なお、ドメインが増えると契約数が増えるため、ドメインが一つしかなかったことは幸いであった。

- GoogleApps For Education はオプションとして利用する。

  利用するドメインは photon.chitose.ac.jp とは “異なる” ドメインである。

  同窓会ドメインも別途用意するので、GoogleApps For Education は二契約となる。SSL サーバ証明書の費用見積りを忘れないように注意が必要だ。

このように二つのクラウドの間に特別な関係はなく、単純に二つあるだけである。各クラウドとの連携システムも二つ独立して存在している。ステータス管理をしている ID 管理システム (3.1 節を参照) だけが二つのシステムにまたがっている。

なお、GoogleApps では Gmail をはじめ何でも利用可能だが、利用するかどうかは、あくまでも自己責任である。オプションサービスなので、利用したい人が利用すればよい。特に、すでに Google アカウントを持つユーザが、大学独自ドメインへの移行を無理に行なう必要はない。

情報センターでは責任分解点を次のように考えている。GoogleApps For Education の認証は情報センターの責任範囲。GoogleApps For Education というサービスそのものは保証外。自己責任で使うもの。つまり、この Google への認証部分が責任分解点となっている。

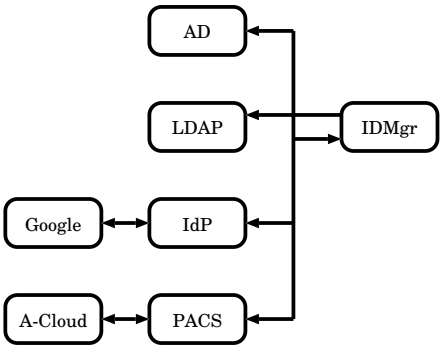

図 3: 認証システムとの連携概念図: ID 管理システム (IDMgr) はマスターデータを加工し、LDAP や ActiveDirectory (AD) などの参照システムへデータを投入する。LDAP は一方通行であるが、AD からはパスワード変更の要求が上がるため双方向通信である。Google や A-Cloud との連携システムへのデータも IDMgr が加工し、データを投入する。ステータス管理をするため、各連携システムとの間は双方向に通信が行なわれる。Google や A-Cloud と各連携システム間も同様にステータス管理があるため、双方向通信である。注: ただし、図の矢印の向きは通信の発呼方向とは一致していない。

3 実装

メールサービスのプライマリは A-Cloud へ変更するが、Gmail も含め GoogleApps For Education も利用可能とする。実装上の課題は「ID 管理システムの再設計」と「A-Cloud との連携」の二点に集約される。

3.1 ID 管理システム

認証系の再構築における要点は 2.2 節で述べた。

新たに ID 管理システム製品 (以下 IDMgr) を核として、マスターデータをもつシステムを再構築する。

以下の仕様を満たす IDMgr 製品を採用した。

1. マスターデータにおけるパスワードが可逆であること。
2. 既存システム (LDAP や ActiveDirectory) との連動が可能であること。
3. GoogleApps との連携サポートオプションがあること。
4. 平文パスワードを逆生成し、他システムと連携することが可能であること。

LDAP や ActiveDirectory など参照される認証サーバ群との連携は IDMgr のプラグイン (オプション) である。これらサーバ群に必要なデータは IDMgr が生成し供給する (図 3)。クライアント PC 側で参照先の認証サーバを切替える必要はない。

GoogleApps との連携には IDMgr のプラグインの他に GoogleApps への Provisioning と IdP を行なうソフトウエアが必要である。特別なことはしていないので他の GoogleApps 連携関連の文献を参照されたい。

以下では A-Cloud との連携について述べる。

3.2 A-Cloud との連携

A-Cloud は SAML 対応予定となっているが、現状は未サポートである。いまのところ独自に Provisioning を行なわなければならない。

Provisioning システム (以下、PACS) は筆者が実装した (図 3)。

PACS は、プログラミング言語 Perl で書かれており、ソースコードは 10000 行弱である。そのうち PACS 固有のコードは 4000 行で、残り 6000 行は fml8[1] のコードを流用している。ターゲット OS は NetBSD だが、Unix クローンであれば問題なく動くだろう。

PACS は単なるキュー管理システムである。キュー管理システムのスケジューラが定期的にサブモジュール群を起動する。モジュールを追加することで、理論上、任意のシステムと連携可能だ。

[1] http://www.fml.org/software/fml8/index.html.ja

将来のライセンス問題を考慮し、サブモジュールはベンダーごとのモジュールとなっており、連携先固有のコードは、すべてこのモジュール内に書かれている。小さなスクリプトを呼び出す処理なども各モジュールの判断で行なわれる (いつものことだが、こういった Unix 的な作業と組み合わせる方が開発は容易である)。

A-Cloud 連携は次のように行なわれる。

1. IDMgr からのデータ転送。

   PACS の IDMgr 固有モジュールが定期的に IDMgr と通信し、未処理キューがあるかどうかを調べる。未処理キューがあればファイルをダウンロードし、PACS 形式に変換後、PACS のキューに入れる。

   この IDMgr からダウンロードするデータは、IDMgr の CSV プラグインを用いて生成している。

2. PACS キュー管理システム呼び出し。

   キュー管理システムは一意な ID をふり、モジュール間ルーティングを制御する。IDMgr からのデータは、最終的に A-Cloud へ送信するキューに入れられる。

3. A-Cloud へのデータ転送。

   A-Cloud モジュールが呼び出され、A-Cloud 形式へ変換後、A-Cloud へ転送する。

4. A-Cloud からの処理結果の転送。

   定期的に A-Cloud モジュールが呼び出され、A-Cloud からの処理結果が返ってきたかをチェックする。処理結果があればファイルをダウンロードし、PACS 形式に変換後、PACS のキューに入れる。

5. PACS キュー管理システム呼び出し。

   キュー管理システムがルーティング制御し、IDMgr へ結果を返すキューに入れる。

6. IDMgr へ処理結果の転送。

   IDMgr モジュールが呼び出され、IDMgr 形式へ変換し、IDMgr へ処理結果を転送する。

ステータス管理は、すべて IDMgr 側で行なう。よって PACS 側で再送処理は試みない。ユーザの作成、削除、パスワードの変更など、必要な処理は IDMgr から要求が発行される。エラーは、一度 IDMgr に返し、IDMgr 側からの再送処理依頼を待つ。

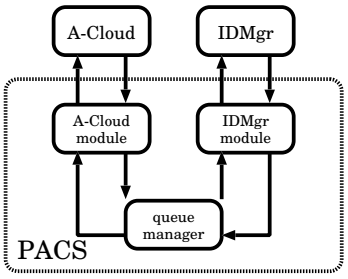

図 4: PACS の内部概念図: IDMgr から処理要求が出されると、モジュール (IDMgr module) が、その要求をデータ転送し、PACS キュー管理システム (queue manager) へ入れる。queue manager は A-Cloud モジュールを呼び出し、A-Cloud へのデータ転送 (Provisioning) と A-Cloud からの処理結果を待つ。その処理結果は、ふたたび queue manager を経て、IDMgr モジュールを呼び出し、IDMgr へ処理結果を返す。注: ただし、図の矢印の向きは通信の発呼方向とは一致していない。

4 まとめ

A-Cloud と Google を併用するシステムを構築した。

2.3 節で述べた理由により、Google には機密情報を置くべきではない。これはユーザにも周知しなければならない。

3.2 節で述べた A-Cloud 連携ソフトウエアはフリーソフトウエアとしてリリースする予定である。動作自体は安定しているので、品質に問題はないだろう。デバッグコードを削除し、レビューをすれば、リリースできる。レビュー最大の目的はベンダー固有コード (たとえば A-Cloud モジュール) が、きちんとソースコードの中で分離されているかである。ベンダー固有モジュールは NDA が必要であるため、リリース版に含まれない (A-Cloud ユーザには提供可能であるので担当営業に相談されたい)。