CASによるセキュアな全学認証基盤の構築

梶田将司　内藤久資
小尻智子　平野靖
間瀬健二

Ⅰ．はじめに

高等教育機関における情報基盤整備は，「学内のコンピュータネットワークの整備」から「大学における教育・研究を支えるアプリケーションの整備」に焦点が移りつつある。しかしながら，コンピュータネットワークとは異なり，アプリケーションは大学における教育・研究の業務プロセスに直結するため，その活用は，業務プロセスの見直しとITによる業務の効率化という「アカデミックリエンジニアリング」なしには進まない。このことは，WebCTに代表されるコース管理システムの活用が進む北米と比較して，我が国の大学にはなかなか導入が進まない状況[1]からも伺える。

しかしながら，コンピュータネットワークがそうであったように，アプリケーションの場合においても「個別対応から基盤対応へ」という流れができると考えられる。アプリケーションを実現する仕組みは，最近はほとんどのものがWeb技術であり，大学のような万単位のユーザを対象とする場合，図１のような負荷分散や高可用性のある機器構成を取る場合が多い。この構成は，アプリケーション間で共有できるため，まず，ここで基盤対応の可能性が考えられる。

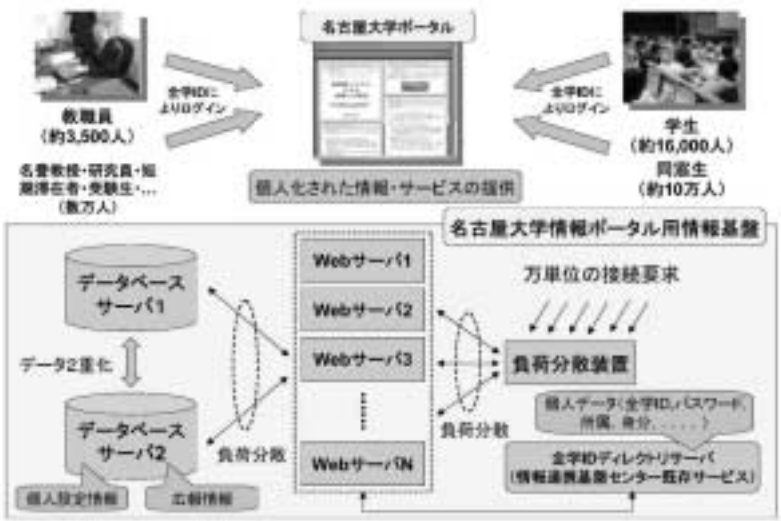

図１　名古屋大学ポータルの機器構成

また，最近では，LDAP（Light weight DirectoryAccess Protocol）サーバやKerberosサーバなどのディレクトリサーバのように，全学レベルでユーザ情報が共有され，情報基盤として運用されるようになってきている。しかしながら，これらの全学レベルでの認証システムで，ユーザIDとパスワードを共通化することができるが，ユーザは，アプリケーションごとに認証を行わなければならないとともに，認証情報を取り扱うため，アプリケーション側もHTTPSなどの暗号化通信によるセキュリティの強化を行わなければならない。このように，セキュア環境を容易に実現でき，かつ，一度ユーザ認証するだけで他のアプリケーションへのアクセス可能なシングルサインオン機能も実現できる全学的に統一された情報基盤の整備が求められている。

そこで本稿では，学内のさまざまな情報システムが共通に利用できるセキュアな全学認証基盤の実現の１つの方法として，Yale大学で開発されたCentral Authentication Service（CAS）[2]を用いて構築した全学認証基盤について述べる。そして，学部２年生から４年生を対象にした平成17年度前期履修登録手続きを通じて行ったCAS化した名古屋大学ポータルの実運用経験についてまとめる。

Ⅱ．CAS

１．概要

CASはYale University ITS Technology & Planningが開発した認証機構で，Webベースのアプリケーションに対してシングルサインオン環境を実現できる。現在は，Java Architecture Special Interest Groupのオフィシャルプロジェクトとして，継続的な開発が進められている。

CASの特徴をまとめると以下のとおりである。

- HTTPリダイレクション，Ticket Granting Cookie，Service Ticket（URLパラメータ）という標準的で一般的なWeb技術を駆使するため，処理が極めて軽く，インストール及び設定が簡単である。CAS認証を利用するためには，CASサーバとCAS認証用のライブラリを利用して認証を行うことになる。
- CASサーバはJava Servletで実現されており，アプリケーションが利用することになるCAS認証を行うためのライブラリとしてJava，PHP，Perl，PL/SQL，ASP，Python用が，また，静的なファイルへのアクセスコントロールが行えるApache mod_casモジュールや，Zope・Plone用のCASライブラリ，uPortal用のモジュール，PAM用モジュールも用意されており，いずれもオープンソースで公開されている。
- 認証に必要なユーザIDとパスワードは，CASサーバにしか送られないため，最低限CASサーバとエンドユーザ間のみHTTPSにより暗号化通信が行われればよい。
- 豊富な実績。30を越える大学で利用されている（2005年４月現在）。

２．CASの認証メカニズム

まず，アクセスのためにCAS認証を必要とするアプリケーションに，ユーザが初めてアクセスした場合の動作を説明する。

CAS認証（その1）

ユーザはアクセスしたいアプリケーションのURLを指定（例えば，`https://mynu.jp/Login`）し，Webブラウザでアクセスする（図２参照）。CAS認証を経ていない場合，アプリケーション側はCASサーバへHTTPリダイレクション機能を使ってアクセスを転送する。その際，`service`パラメータを用いて，CASサーバが認証すべきサービスを伝える。CASサーバは，Webブラウザに保存されているTicket Granting Cookie（TGC）を確認し，TGCがない場合は，ユーザ認証がまだ終わってないと判断，認証画面（図３参照）を表示する。

CAS認証（その2）

ユーザは認証画面で，ユーザIDとパスワードを入力し送信する（図４参照）。入力された認証情報は，LDAPなどの認証源に問い合わされ，認証結果を得る。認証源として，LDAP，NIS，RDBMS，通常のファイルなど利用可能なCAS Generic HandlerをESUP-Portail consortiumが提供している[3]。

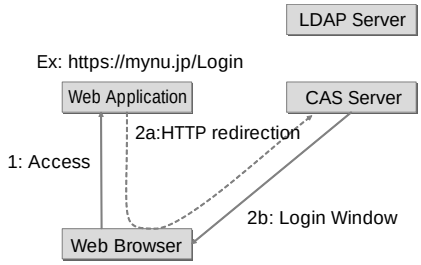

図２　CAS認証を経ていないアプリケーションへのアクセスはCASサーバへ自動的にリダイレクトされ，認証画面が表示される

図３　名古屋大学CASログイン画面

CAS認証（その3）

ユーザが正しく認証されると，CASサーバはWebブラウザに対してセキュア属性[1]をつけたTGCを発行するとともに，URLパラメータ`ticket`に

```
ticket=ST-415240-RGhNbrthiZKezNr9AA7t
```

のようなService Ticket（ST）をセットし，再度，呼び出されたアプリケーションにHTTPリダイレクトする（図５参照）。

1　HTTPSでの接続でなければWebブラウザはWebサーバにCookie情報を送信しない。

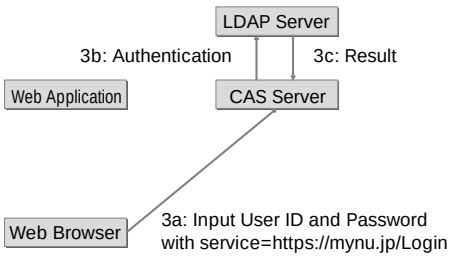

図４　入力されたユーザIDとパスワードは，認証源に問い合わされ認証結果を得る

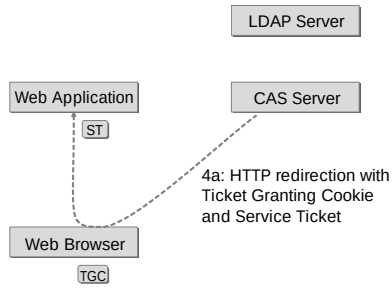

図５　認証されると，呼び出されたアプリケーションへ自動手的にリダイレクトされる。その際，TGCとSTが発行される

CAS認証（その４）

アプリケーションは取得したSTを検証するため，CASサーバに対してSTを送信する（図６参照)。CASサーバでは，メモリ上のST発行情報をもとにチケットの正当性を検証する。この際，名古屋大学のCASサーバでは，LDAP属性や時間などによるアクセス制限を課すことも容易に実現できるようにした。このCASにおけるアクセス管理については，次節で詳細に述べる。

アクセスの正当性が確認されると，ユーザIDがアプリケーション側に通知され，その情報に基づいてアプリケーションはユーザにサービスを提供する（図７参照)。

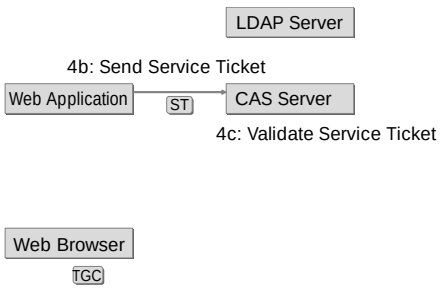

図６　アプリケーションに送られたSTはCASサーバに再度送られ正当性が検証される.

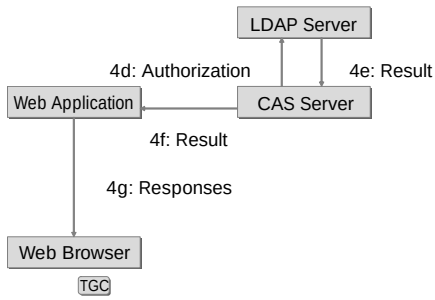

図７　アクセスしようとしているアプリケーションへの権限があるかどうかLDAPで検証。OKであれば，アプリケーションに必要なLDAP属性が送信され，それに基づいてアプリケーションはサービスを提供する

Ⅲ. CASによる権限管理

我々はCASに対して，以下の意味での権限管理機構を導入した:

1. STのValidation要求を受けた際に，アクセスを行おうとするユーザが該当のURLに対するアクセス権を持つかどうかを判断。

ıbアクセス権がある場合には，CASクライアントであるアプリケーションに送信するユーザデータベースの属性値を任意に設定可能。

この権限管理機構によるアクセス制限のリストは外部データベースで設定し，CASの起動時及び起動後の任意の時点で設定可能である。このアクセス制限リストを**CAS Access Control List（CASACL）**と呼ぶこととする。今回，名古屋大学ポータルではCAS-ACLはLDAP DIT内に格納した。

LDAPを利用した場合，CAS-ACLデータベース内の各エントリは以下の３種類に分類される。

１．標準的なCAS-ACLエントリ：Service Validation requestに対してAuthorizationを行うCAS-ACLエントリ。cas-auth-typeの属性値がbasicとなっている。

```
dn: cn=uPortal,ou=uPortal,ou=cas,o=NU
cn: uPortal
description: uPortal
cas-auth-type: basic
cas-attributes: uid,.....
cas-service: https://mynu\.jp/uPortal/.*
cas-allow: （dn=.+,ou=place.?,o=nu）
objectClass: top
objectClass: cas
```

２．basic CAS-ACLエントリのためのマクロエントリ：cas-auth-typeの属性値がaccess_filterとなっている。

```
dn: cn=nagoya-univ-students-b,ou=cas,o=NU
cn: nagoya-univ-students-b
cas-auth-type: access_filter
cas-allow: （&（dn=.+,ou=place.?,o=nn）...
objectClass: top
objectClass: cas
description: Nagoya University User
```

３．CAS-ACLデータベースをアップデートする権利を持つユーザを指定するCAS-ACLエントリ：cas-auth-typeの属性値がtrustedとなっている。

```
dn: ou=uPortal,ou=cas,o=NU
objectClass: top
objectClass: organizationalunit
objectClass: cas
cas-allow: （uid=XXXXX）
ou: uPortal
cn: trusted
description: trust for uPortal
```

このCAS-ACLエントリの構造から，自然に「正規表現で記述されたURLのグループ」が構成される。我々は，このグループのことを**CAS Access Control Class（CAS-ACC）**と呼んでいる。すなわち，CAS-ACCとは，アクセス条件とWebアプリケーションに対して送信する属性値が同一となる「正規表現で記述されたURLのグループ」である。

Ⅳ．Web履修登録による名古屋大学ポータルの実運用

最後に，平成17年度前期履修登録手続きを通じて行ったCAS化した名古屋大学ポータルの実運用経験についてまとめる。

１．CAS化した名古屋大学ポータルの構成

名古屋大学ポータルは，Webサーバ群とデータベースサーバ群で構成される（図１参照）。Webサーバ群では，Sun Microsystems社のSunFire V210を５台，CAS・LDAPサーバとしてSunFire V480を１台，データベースサーバ群では，SunFire V240 ２台とStorEdge 3150FCを用いている。Webサーバ群に対する負荷分散はNortel Networks Alteonが，データベースサーバ群に対しては，Oracle 10g Real Application Clusterが負荷分散を行っている。なお，大学ポータルフレームワークであるuPortalを用いて名古屋大学ポータルを構築している（図３参照）。

Web履修登録処理を行う新教務システムは，Sun-Fire V120 ２台及びV210 ２台上でOracle Application Serverを動かし，SunFire V240 １台上でOracle 9iのPL/SQLにて独自開発したソフトウェアを用いて実現されている。Oracle AS サーバの負荷分散は，名古屋大学ポータルと同じAlteonで行った。

CAS認証については，uPortalについてはuPortalパッケージ及びJavaクライアントを，新教務システムについてはPL/SQL用CASクライアントを用いて個別にCAS認証を行うように構成した。

２．Web履修登録処理の実運用

名古屋大学では，２年生から４年生までの学部学生約6,500名を対象にWebによる履修登録を2005年３月22日から30日にかけて行った。期間中，学内だけでなく，学外からのアクセスも許可し運用した。

限界性能試験

履修登録運用に先立ち，e-Testを用いた負荷実験を行った。その結果を表１に示す。測定は，ボトルネックとなることがあらかじめ分かったデータベースサーバのCPU使用率が85%以上に達した場合とした。表から分かるように，CAS認証を行ったとしても高負荷が予想される履修登録において十分な性能が得られることが分かった。なお，ポータルが履修登録に比べてスループットが悪いのは，ログイン時の処理に時間がかかることが分かっている。

表１　CAS化した名古屋大学ポータルと新教務システムの限界性能。ポータルは，CASが関係するログイン・ログアウトのみを行った。各値は約５分間の平均値

処理	スループット	レスポンス
内容	［ページ/秒］	［秒］
履修登録	37.5	1.5
集中登録	60.0	1.1
ポータル	17.5	2.4

ユーザの利用状況

CAS認証の回数から見たユーザのアクセス状況を図８に示す。今回は，学外からの利用も許可したため，深夜にわたる利用があったことがよく分かる。

また，ユーザが利用しているWebブラウザはつぎのとおりであった。

```
4199 （57.0%） Windows.XP.MSIE
1757 （23.9%） Windows.2000.Netscape
 757 （10.3%） Windows.98.MSIE
 201 （ 2.7%） Windows.2000.MSIE
 126 （ 1.7%） Linux..Netscape
```

２位のNetscapeは情報メディア教育センターの端末室からの利用を反映している。

アクセス元・経路の状況

アクセスログに基づいて，ユーザのアクセス経路を調査した。経路の同定手順はつぎのとおり。

１．アクセスログに記載されているIPアドレスからFQDNを求める。ただし，10秒以上経ってもレゾルブできない場合はunknownとした。

２．JPIX名古屋に接続されている地域ISP（７社）のドメイン，及び，中部テレコミュニケーションに接続されているドメインに属するホストからのアクセスを，JPIX名古屋経由のアクセスと判断した。

３．IPアドレスが133.６.０.０/16の場合は，名古屋大学内からのアクセスと判断した。

４．上記以外の経路については，すべてSINET経由のアクセスと判断した[2]。

ただし，学内の端末システムからのアクセスなど，同一ホスト（あるいは，IPアドレス）からのアクセスが想定されるため，アクセス時の全学IDとホストのFQDNを対にし，アクセス元の総数を同定した。

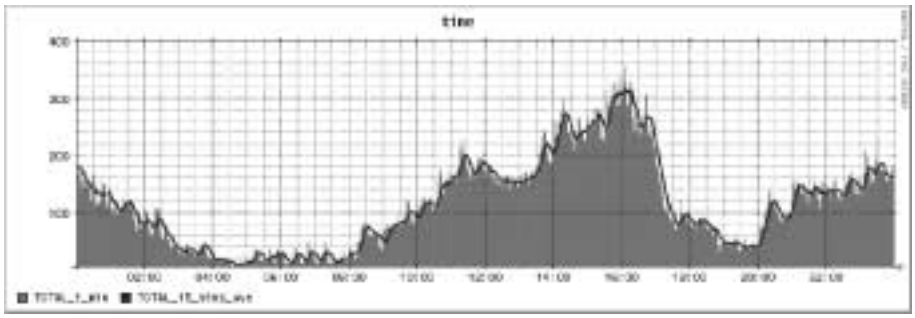

図８　CAS認証の回数から見たユーザのアクセス状況。履修登録期間中の平均。18時から20時は保守時間としたためアクセスは少ない。

2　名古屋大学は，JPIX名古屋接続以外はSINETにしか外部経路を持たない。

まず，10位までのアクセス元を調べたところ，下記のようになった:

```
1922 （26.1%） jp.ac.nagoya-u.media
 984 （13.4%） net.bbtec
 656 （ 8.9%） jp.ne.dion
 641 （ 8.7%） jp.ne.ocn
 403 （ 5.5%） jp.ne.starcat
 229 （ 3.1%） unknown_in_nu
 211 （ 2.9%） jp.ne.so-net
 199 （ 2.7%） jp.ne.aitai
 184 （ 2.5%） jp.ad.mesh
 167 （ 2.3%） jp.or.plala
```

この結果から，今回の履修登録に関しては，１/４程度のユーザが学内の情報メディア教育センターの端末室からアクセスしていることが分かった。

また，上記の結果から，JPIX名古屋を経由するアクセス，学内のアクセスの比率，それ以外を表２に示す。表から分かるように，10.2%がJPIX名古屋経由となっている。これは，アクセス元の多くを占めるISPからのアクセスがSINET経由となっているためであり，直接のピアリングあるいは地域IXを利用するISPが増えることで，さらなる改善が期待できる状態であることが分かった。

表２　名古屋大学ポータルへのアクセス経路

アクセス経路	アクセス数
JPIX名古屋経由	966（10.2％）
SINET経由	5,672（59.9％）
学内から	2,836（29.9％）
総数	9,474

Ⅴ．まとめ

本稿では，学内のさまざまな情報システムが共通に利用できるセキュアな全学認証基盤の実現の１つの方法として，CASを用いて構築した全学認証基盤について述べた。

我々が行ったLDAPによる権限管理強化は，カナダのQueen ’s Universityや他の大学でも似通った拡張が行われており，Central Authentication Serviceだけでなく，Central Authorization Serviceのニーズも明確に存在していると言える。コードのオープンソース化など，CASコミュニティへのフィードバックは必須であろう。

謝辞

本研究は，文部科学省平成16年度「知的資産の電子的な保存・活用を支援するソフトウェア技術基盤の構築」研究開発課題「ユビキタス環境下での高等教育機関向けコース管理システム」（研究代表者：間瀬健二），及び，文部科学省科学研究費基盤研究（A）「地域学術コンソーシアムに

おけるe-Learning地域ハブに関する研究」（研究代表者：梶田将司，課題番号：15200054）の助成を受けて実施されている。

参考文献

［1］日本WebCTユーザ会，http://www.webct.jp/
［2］Yale University ITS Technology & Planning, http://tp.its.yale.edu/tiki/tiki-index.php
［3］CAS Generic Hander, http://esup-casgeneric.sourceforge.net

（かじた　しょうじ：名古屋大学情報連携基盤センター情報基盤システムデザイン研究部門）
（ないとう　ひさし：名古屋大学多元数理科学研究科）
（こじり　ともこ：名古屋大学情報連携基盤センター情報基盤システムデザイン研究部門）
（ひらの　やすし：名古屋大学情報連携基盤センター大規模計算支援環境研究部門）
（ませ　けんじ：名古屋大学情報連携基盤センター情報基盤システムデザイン研究部門）