

# クラウド時代のID管理と SSO(シングル・サイン・オン) 最新技術動向と導入事例



**OSSTech**

オープンソース・ソリューション・テクノロジー株式会社  
代表取締役 チーフアーキテクト 小田切耕司

お問い合わせ [info@osstech.co.jp](mailto:info@osstech.co.jp)

# 講師紹介

## オープンソース・ソリューション・テクノロジー

### 会社紹介



**OSSTech**

# 講師紹介

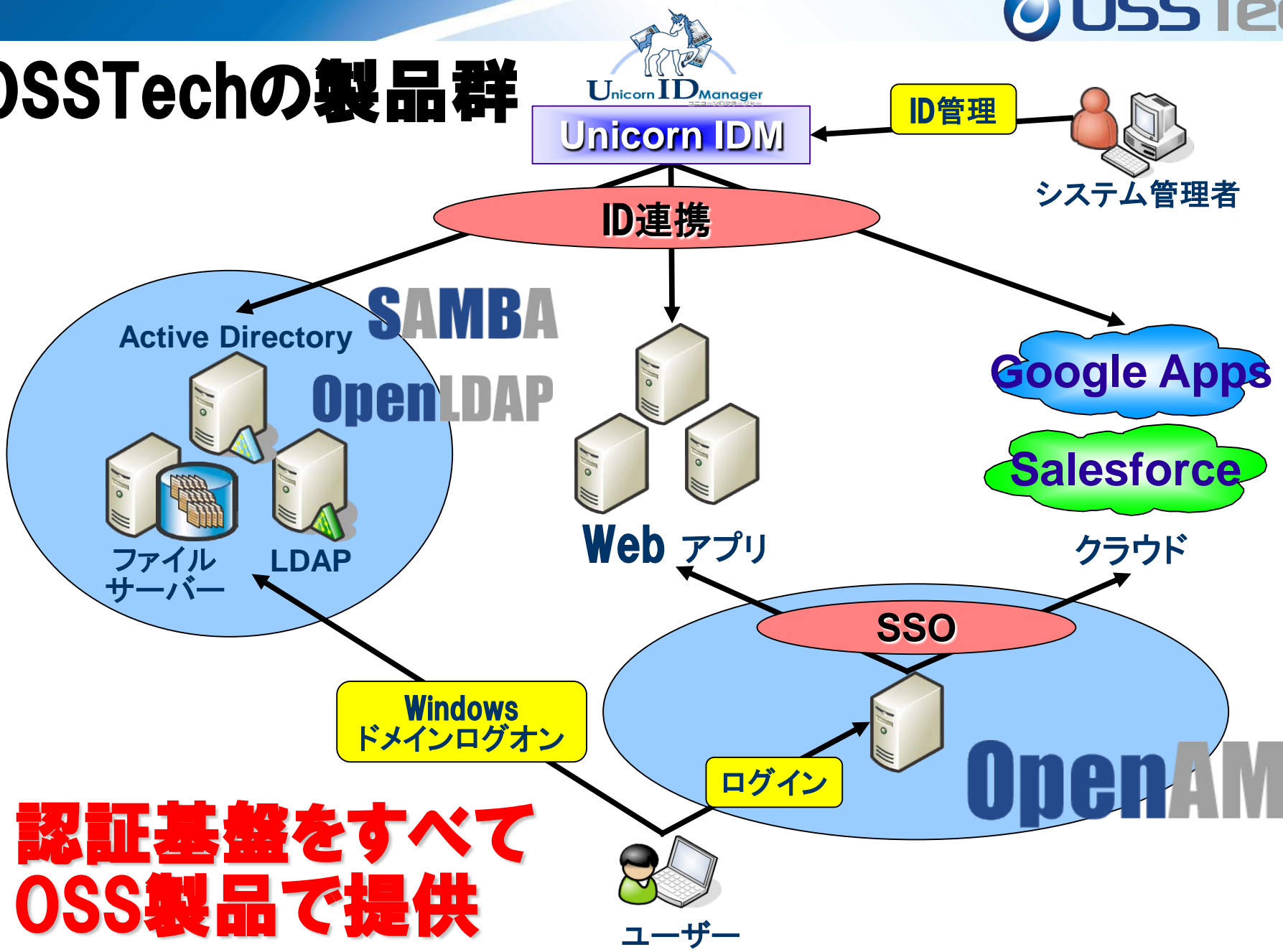
- 役職：代表取締役 チーフアーキテクト
- 氏名：小田切 耕司（おだぎり こうじ）
- 所属団体等
  - OpenAMコンソーシアム 副会長
  - OSSコンソーシアム 副会長
  - 日本LDAPユーザ会設立発起人
  - 日本Sambaユーザ会初代代表幹事
- 執筆関係
  - 日経Linux 2011年9月号～2012年2月号 連載中
    - 『Linux認証のすべて』（第1回～第6回）
    - <http://itpro.nikkeibp.co.jp/linux/>
  - ASCII.technologies 2011年2月号
    - 『キホンから学ぶLDAP』
    - <http://tech.ascii.jp/elem/000/000/569/569412/>
  - 技術評論社 Software Design 2010年9月号
    - 第1特集 クラウド対策もこれでOK！
    - 統合認証システム構築術
    - OpenAM/SAML/OpenLDAP/Active Directory
    - <http://gihyo.jp/magazine/SD/archive/2010/201009>
  - @IT やってはいけないSambaサーバ構築：2008年版
  - 2006年5月 技術評論社 LDAP Super Expert
    - 巻頭企画
    - [新規/移行]LDAPディレクトリサービス導入計画



# オープンソース・ソリューション・テクノロジー株式会社

- **OSに依存しないOSSのソリューションを中心に提供**  
Linuxだけでなく、AIX, Solaris, Windowsなども対応！
- **OpenAM, OpenLDAP, Sambaによる認証統合/  
シングル・サイン・オン、ID管理ソリューションを提供**
  - **製品パッケージ提供**  
機能証明、定価証明が発行可能
  - **製品サポート提供**  
3年～5年以上の長期サポート  
コミュニティでサポートが終わった製品のサポート
  - **OSSの改良、機能追加、バグ修正などコンサルティング提供**

# OSSTechの製品群



**認証基盤をすべて  
OSS製品で提供**

# OSSTechの製品群(すべてOSSで提供)

## 原則Linux/Solaris/AIX共にRPMで提供

- **Samba for Linux/Solaris/AIX**
  - ADの代替、高性能NASの代替
- **OpenLDAP for Linux/Solaris/AIX**
  - 認証統合、ディレクトリサービス、シングルサインオンのインフラ
- **OpenAM for Linux/Windows**
  - Tomcat, OpenLDAP対応で高機能なシングルサインオン機能を提供 (旧OpenSSO)
- **Unicorn ID Manager for Linux**
  - Google Apps, Active Directory, LDAPに対応した統合ID管理



# **OSSTechの製品群(すべてOSSで提供)**

## **原則Linux/Solaris/AIX共にRPMで提供**

- **Chimera Search(キメラサーチ) for Linux**
  - ・ アクセス権の無いファイルは表示されない全文検索システム
- **LDAP Account Manager for Linux**
  - ・ 管理機能の弱いOSSのLDAP/SambaにWebベースのGUIを提供
- **ThothLink(トートリンク) for Linux**
  - ・ WebブラウザからのWindowsファイルサーバアクセス機能を提供
  - ・ SSLBridge後継製品
- **Mailman for Linux**
  - ・ 日本語での細かな問題を解決
  - ・ YahooメールやGoogle Appsのメールングリスト機能を補完

# 統合認証とシングルサインオン



# 統合認証とシングルサインオンの必要性

- クラウド(外部のWebサービスの業務利用)が普及したことで、**統合認証/シングルサインオン/ID管理の必要性が急上昇**
- 社内Webアプリ(オンプレミス)の利便性・セキュリティ向上のための需要も同時に増加中
  - 社内にある多数ありWebアプリ(オンプレミス)へのアクセスをシングルサインオンで管理し、利便性を向上させたい
  - 社内のWebアプリと外部のWebサービス(Google Apps、Salesforceなど)をシングルサインオン連携したい(クラウドサービス利用者)
  - クラウド基盤の構成コンポーネントとして、OpenAMを利用したい(クラウドサービス提供者)

# 統合認証とシングルサインオンとID管理

- **統合認証とシングルサインオンとID管理は同時に使うことで最大の効果を発揮する**
  - ユーザーID/パスワードはシングルサインオンシステムで一元管理可能でも、各アプリケーション・サービス毎に必要なユーザー情報は、基本的には個々に管理される
  - ID管理ツールなどを利用した一元管理をしなければ、ID管理は破綻する
- **クラウドサービスにおいても、ID管理は必要**
  - クラウドサービスもユーザー情報を保存することから、ID管理の対象となる
  - ID管理用のAPI(プログラムインタフェース)を備えているものが多い(Google Apps、Yahoo! など)

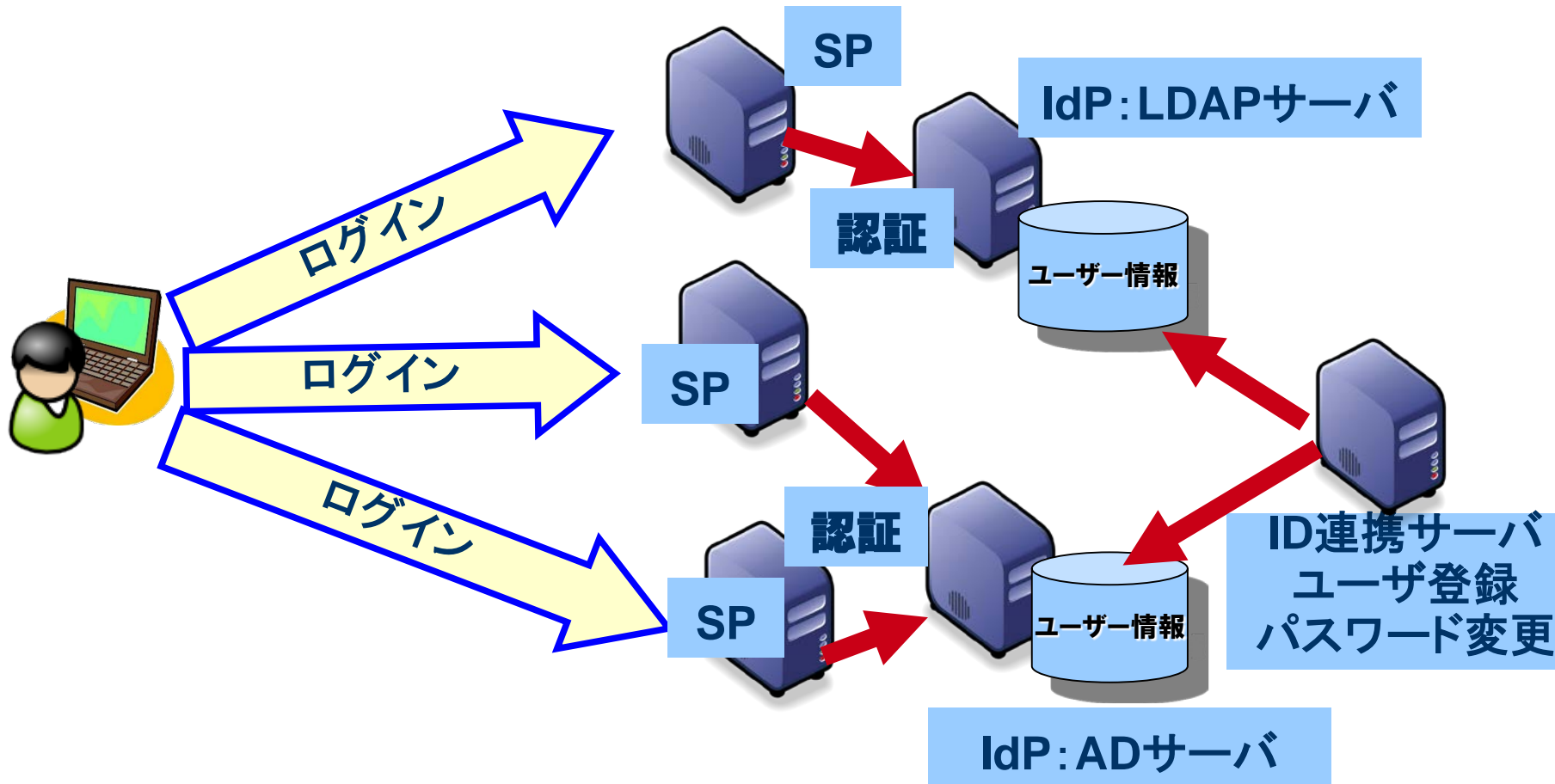
# 統合認証とは？

- WindowsやLinux/UNIXの認証をできる限りユーザの負担が少ない方法で提供する
- 色々なコンピュータへ同じアカウント名とパスワードでログインできる
- コンピュータのログインだけでなく、その上で動く様々なサービスへも同じパスワードでログインできる  
※例)
  - メールサーバ (POP,IMAP,SMTPサービス) の認証
  - PPPやVPNなどのリモート接続のためのRadius認証
  - ApacheやIISなどのWebサーバのBasic認証やForm認証

# プロビジョニング(ID配布)による統合認証

- 複数のシステムに同じアカウントとパスワードを設定
  - ID管理データベースは別々になっている
- システム毎にやるのは大変
  - 「統合ID管理機能」を持つソフトウェアを導入するのが一般的
- 統合ID管理ソフトは、複数システムのアカウントとパスワードを集中管理
  - 1ヶ所でIDを登録すると複数のシステムへ自動的にIDを一括登録する機能
  - ユーザが1ヶ所でパスワードを変更すると関連するすべてのシステムのパスワードを変更する機能
- 既存システムにできる限り手を加えずに実現できる方式として大変実用的
- パッケージソフト利用やSaaS利用において「ID統合による統合認証」に対応していない場合にも有用な方法

# プロビジョニング(ID配布)による統合認証



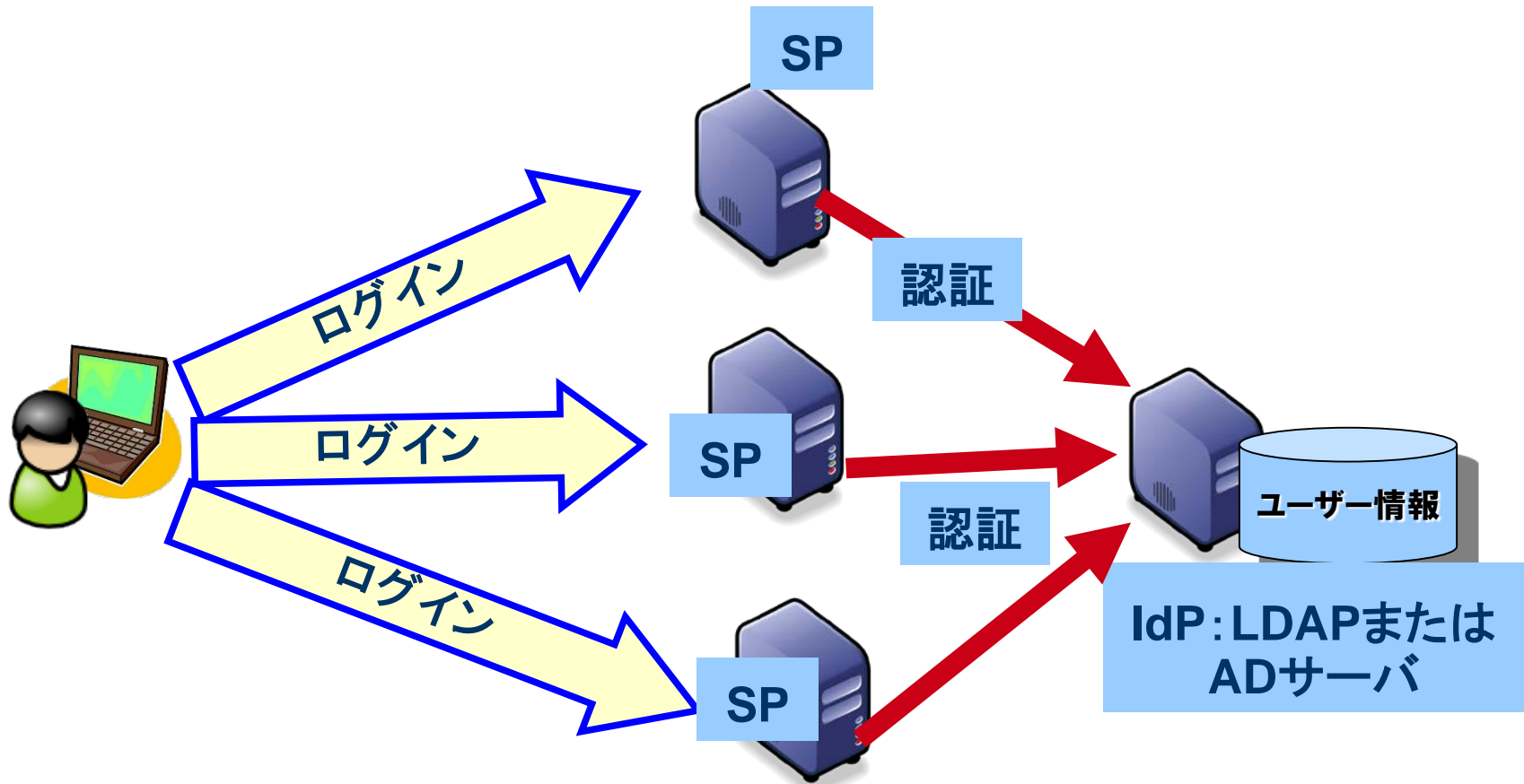
- ユーザ登録をADとLDAPに一括して行う
- パスワード変更も同時に行う

# ID統合による統合認証とは？

- 望ましい形はIDをひとつに集約し、これですべての認証を統合してしまうこと
- ID連携による統合認証は、導入費用が既存ソフトを改修する費用よりも安くないとメリットはない
- IDをひとつに統合する方法
  - ① UNIX/Linux上のLDAPによる統合認証
  - ② Windows ActiveDirectoryによる統合認証
  - ③ UNIX/Linux上のActiveDirectoryによる統合認証
  - ④ Windows上のLDAPによる統合認証

※RDBによるID統合は不可能ではないが容易ではない

# ID統合による統合認証





# UNIX/Linux上のLDAPによる統合認証

- サーバやクライアントにUNIX/Linux(Mac OS XもUNIX系)を利用している場合に推奨
- メールサーバやWebサーバ、アプリケーションサーバがUNIX/Linux上で動作している場合も推奨
- UNIX/Linuxが標準でPAM (Pluggable Authentication Module:IDとパスワードで認証するモジュール、ICカード認証や生体認証など他の認証方式もPAMがあれば実現可能)とNSS(Name Service Switch:ユーザやプロセスにUNIXのuid,gidを提供する)をサポートしている
- JavaやRuby, Perl, PHP, Pythonなどのプログラム言語がLDAPのクラスやモジュールを提供している
- PAMはUNIX/LinuxのOSログインから、その上で動く様々なサーバソフトの認証も制御することのできるモジュールで大変汎用的にできているため、PAMに対応しているvsftpdやsshd, postfix, dovecotなどは簡単にLDAP認証に切り替えることが可能
- ApacheやProFTPdやFreeRADIUSなどPAM経由だけでなく、直接LDAPのAPIを利用することでLDAP認証を実現しているものもある
- すべてのOSやプログラムが認証データベースもしくは認証プロトコルとしてLDAPのそれを利用することで同じIDとパスワードでOSやアプリにログインできることになる

# Windows ActiveDirectoryによる統合認証

- Windowsでは標準機能で実現可能
  - クライアントをADドメインに参加
- UNIX/Linuxの場合でもPAMを使うことで利用可能
  - ① LDAPのPAMを使う方法
  - ② KerberosのPAMを使う方法
  - ③ SambaのWinbind機能のPAMを使う方法

# LDAPのPAMを使う方法

- 認証後プロセスが動作するためのuid,gidが必要になるため、これをNSSから利用できるようにするにはAD側にSUA(Subsystem for UNIX-based Applications)などを使ってUNIX用の拡張スキーマを入れる必要がある
- ADを変更する必要があるので敬遠されがち

# KerberosのPAMを使う方法

- Kerberosサーバで認証後にチケットもらってサーバにアクセスする方法
- セキュリティが強固になるが、uid,gidが必要
- Kerberosサーバは提供してくれないため、AD側にUNIX用の拡張スキーマを入れ、NSSだけNISを使う
- Kerberosのチケットを使うことができるので、一度ログインすれば同じPAMを使うサービスに再度パスワードを入力せずに利用できるSSOも実現可能

# SambaのWinbind機能のPAMを使う方法

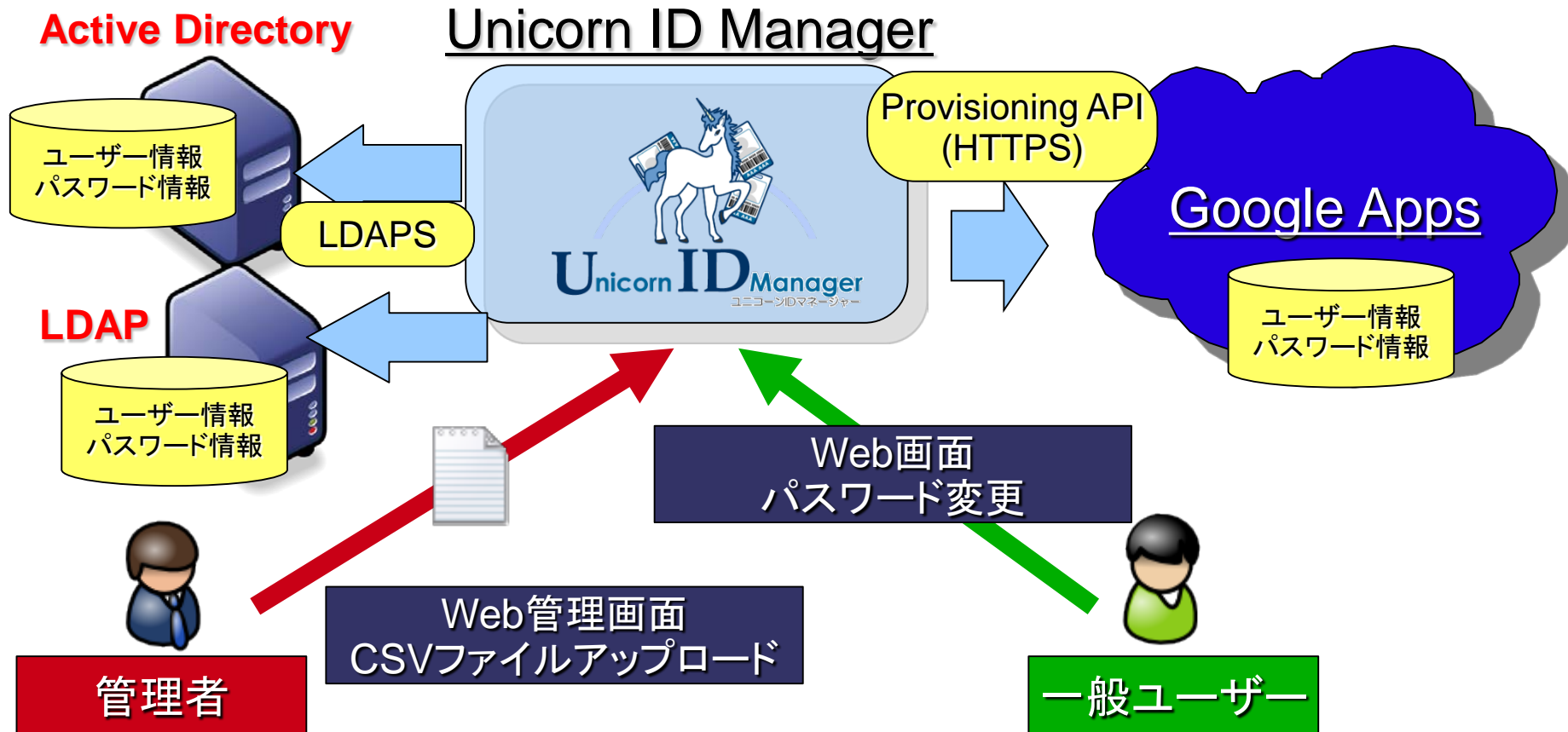
- Kerberosのチケット方式で認証し、uid,gidをWindowsのSID(セキュリティ識別子)から自動生成することが可能
- SUAのインストールは不要  
(SUAを入れてADのUNIX拡張スキーマでuid,gidを提供することも可能)
- PAMもNSSもSambaが提供するwinbindモジュールを使うことで実現可能
- 認証でKerberosのチケットを使うことができるので、一度ログインすれば同じPAMを使うサービスに再度パスワードを入力せずに利用できるSSOも実現可能

# それぞれの長所と短所

- LDAPの制限
  - Sambaを使えばWindowsクライアント統合認証も可能だがSamba3系ではADのグループポリシーをサポートしていない
- ADの制限
  - ADで統合認証するにはWindowsのCALが必要  
ユーザ数が増えるとコストがかさむ
  - Windowsサーバの信頼性がUNIXサーバより劣る場合が多く、UNIXサーバの認証をADに任せるとWindowsサーバの障害がUNIXサーバの障害へつながってしまう
- ADとLDAPの両方を導入し、WindowsクライアントにはADを使った統合認証を提供し、UNIX/Linux/Mac OSクライアントにはLDAPを使った統合認証を提供し、ADとLDAPの間をID連携による統合認証を行うという方式を取るユーザが多い

# Unicorn ID ManagerによるAD,LDAP連携

- WebブラウザからCSVを投入するだけで統合ID管理





# UNIX/Linux上のADによる統合認証

- ADとLDAP両方の欠点を補う方法
- UNIX / Linux上でADを動作させ統合認証する方法
- ADとほぼ同機能を持ったSamba4がリリースされた
- LDAPによる統合認証が実現できる
- UNIX/Linux (Mac) はLDAPクライアントになる
- Windows (Mac) クライアントはSamba4を通してLDAPの中に格納されたIDでAD(Kerberos)認証が可能

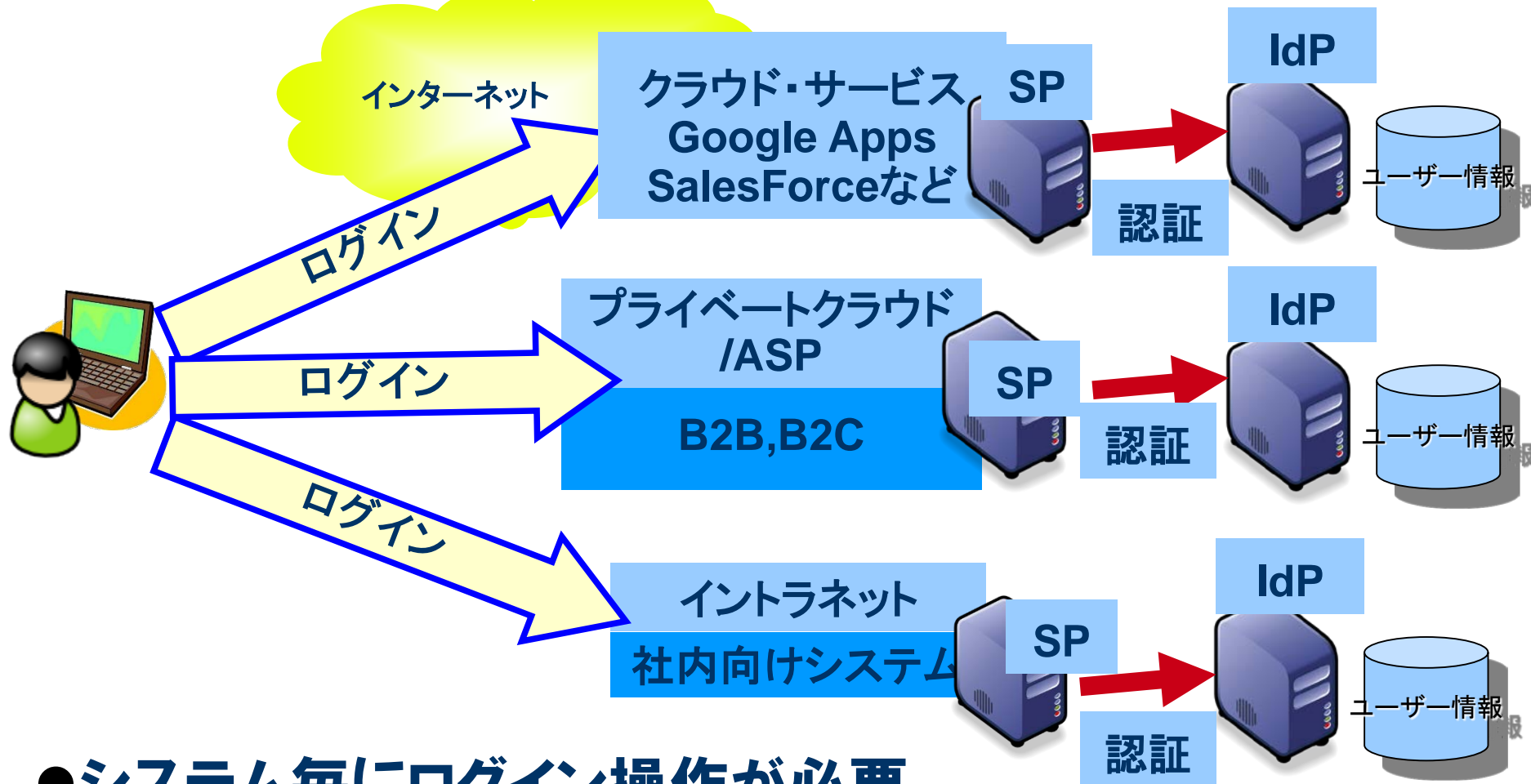
# Windows上のLDAPによる統合認証

- ADはLDAPの機能を包含しており、Windowsの上でADではないLDAPを動かす意義はあまり無い
- しかし、OpenLDAPなどのUNIX系LDAPの代わりにADを使うのは品質、性能、柔軟性の綿で不安要素が多い
- OpenLDAP for WindowsというOSSの製品もある
- Javaの上で動くOpenDS/OpenDJやOracle Directory Serverなど商用のLDAP製品も多数ある
- Windowsサーバの上でOpenLDAPを利用する場合もCALを購入する必要がある
- 商用製品の場合は該当製品のユーザライセンスに加え、WindowsのCALの2重の費用がかかる
- Windows XP,Vista,7クライアントの上にサーバソフトをインストールしてサーバ用途に利用することは費用がかかる以前にライセンス違反となる  
(Windowsクライアント製品の場合、サーバプロセスへの同時接続数に制限がかけられている)

# シングルサインオンとは

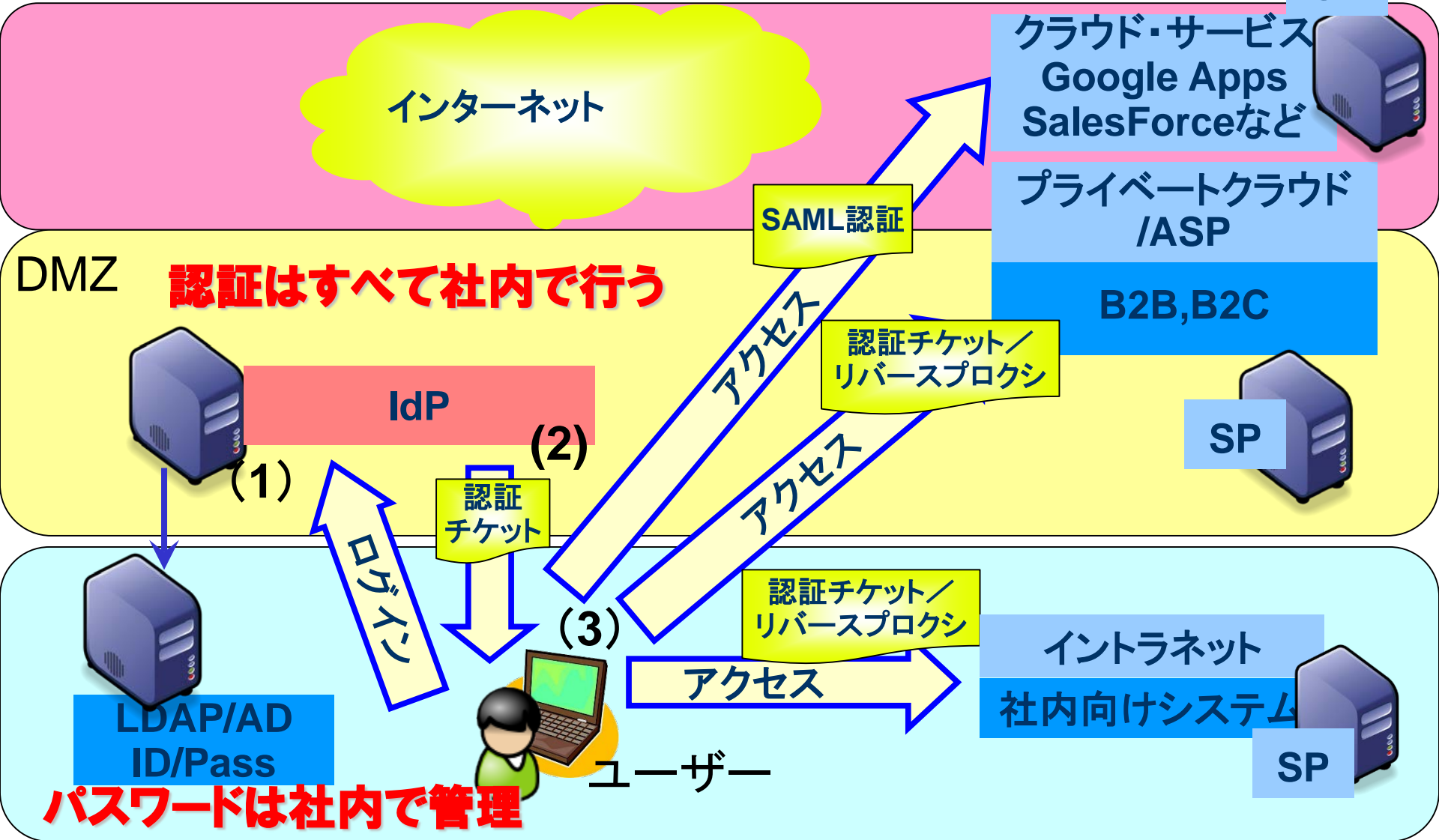
- 1回のパスワード入力で複数のシステムやサービスを同時利用
- 「ID統合を使った統合認証」ではIDとパスワードの管理を1カ所でできるためユーザの追加も楽、社員が退社した場合に1カ所IDを削除すれば、すべてのシステムが利用不可となる
- 近年クラウドサービス(SaaS, PaaS, IaaS, HaaSなど)の普及により、(社外にある)サービス毎にID／パスワードを登録しなければならないケースが増えており、「ID連携による統合認証」を使わざるを得ないケースが増えている
- ところがこのID連携が費用の問題や技術的な問題で完全に実現されていない場合、例えば社員が退社した時に社内システムのIDを削除しても、SaaS側のIDが残っているとクラウド側のシステムは社外から使えてしまう、といった問題が起きてしまう

# クラウドで統合認証ができていないと...



- システム毎にログイン操作が必要
- クラウドにID/パスワードとパスワードを置く必要がある  
(パスワードを社外に置くと不正ログインされる危険性が高い)

# クラウドで統合認証とSSOを実現する



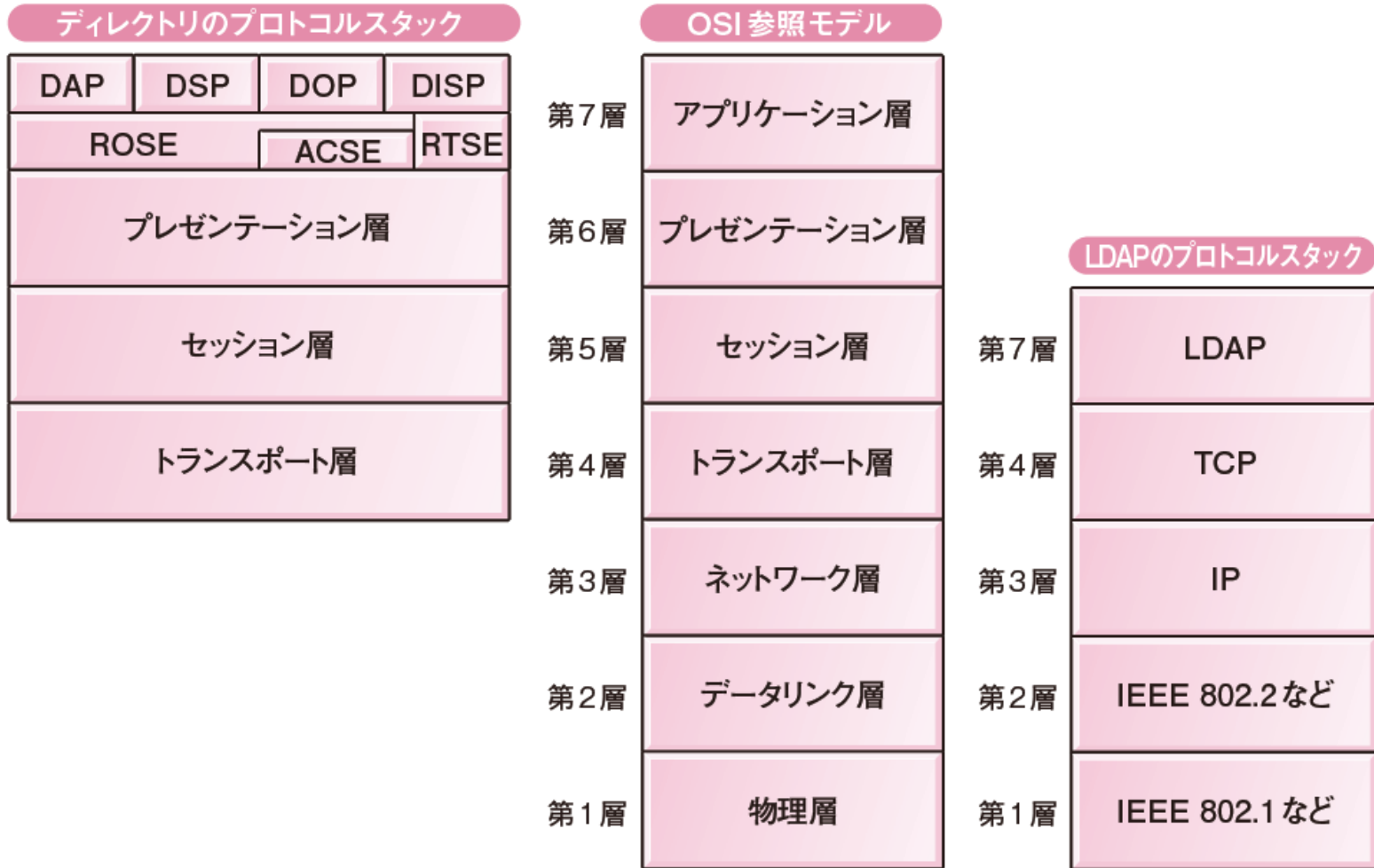
# LDAP概念と設計入門

# LDAPとは？

- ディレクトリサービスを利用するための規約の1つ (RFCで定義)
  - ディレクトリサービスとは、キーを基に関連情報を取り出す仕組み
  - ユーザ管理、電話帳、リソース管理などに利用
  - 高機能だが運用負荷や開発コストが高かったITU-T 勧告のX.500 ディレクトリ・サービスを「90%の機能を10%のコストで実現する」ために設計
- 商用LDAP製品も多数存在
  - Sun Java Directory Server, Red Hat Directory Server, Novell eDirectoryなど
  - MS Active DirectoryもLDAP準拠(認証はKerberos)
- オープンソースソフト
  - OpenLDAP
    - Linux ディストリビューションに同梱されるオープンソースのLDAP
  - Red Hat / Fedora Directory Server
    - かつてのNetscape Directory ServerをOSSにしたもの (RHは有償、Fedoraは無償)
  - Apache Directory Server
    - Apacheプロジェクトが進めるJavaで書かれたDS



# LDAPのプロトコルスタック(X.500との違い)



X.500でのDAPとLDAPのプロトコルスタックの違い

# LDAPにおいてX.500から削除されたもの

ROSE (Remote Operation Service Element)	遠隔操作サービス要素、処理の依頼と結果の通知という通信メカニズムを実現するプロトコル要素
RTSE (Reliable Transfer Service Element)	高信頼転送サービス要素、通信経路障害などによって情報の欠落や重複が起きないようにするプロトコル要素
ACSE (Association Control Service Element)	アソシエーション制御サービス要素、コネクションの確立、正常開放、異常解放を行なうサービス要素

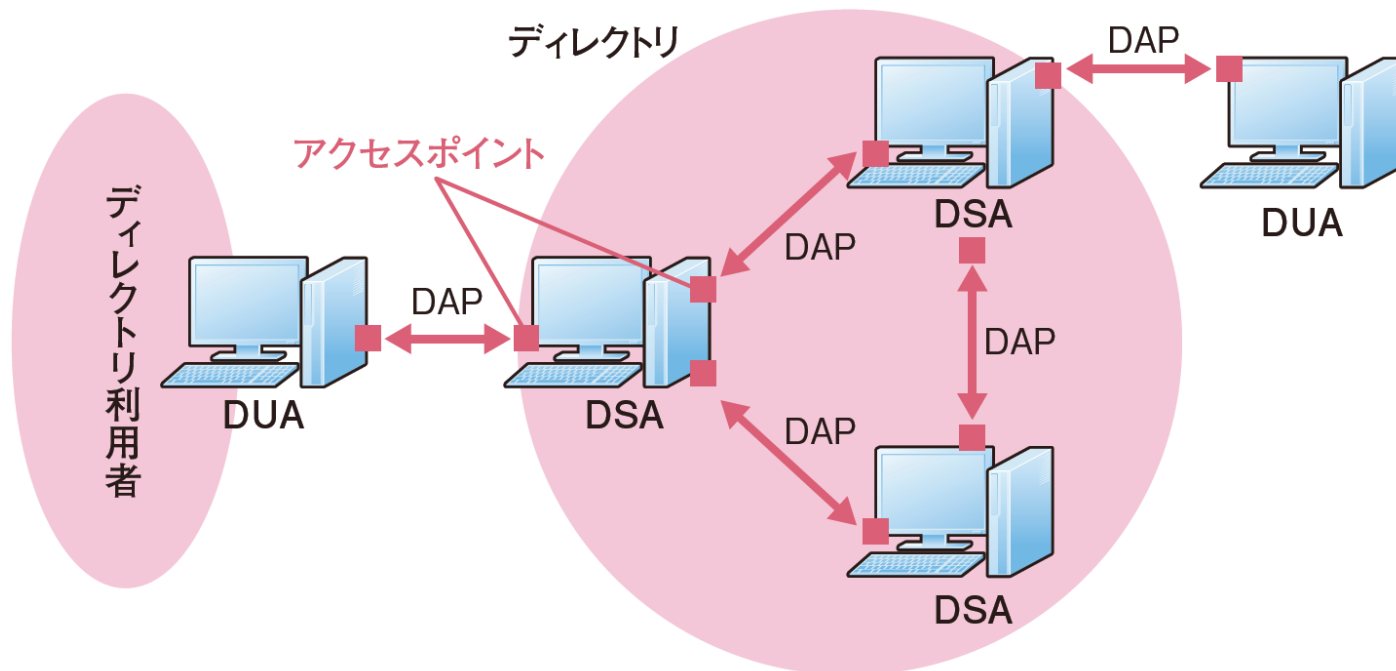
## TCP/IP上で動作するためにDAPから不要になった機能

DSP (Directory System Protocol)	DSA間で分散協調動作(連鎖や紹介)を行なうためのプロトコル
DOP (Directory Operational binding management Protocol)	ディレクトリ運用結合管理プロトコル。DSA間の運用結合の規定内容や状態の交換に用いられるプロトコル
DISP (Directory Information Shadowing Protocol)	DSA間で複製情報を交換するためのプロトコル

## DAP以外のX.500の機能

# ディレクトリの機能モデル

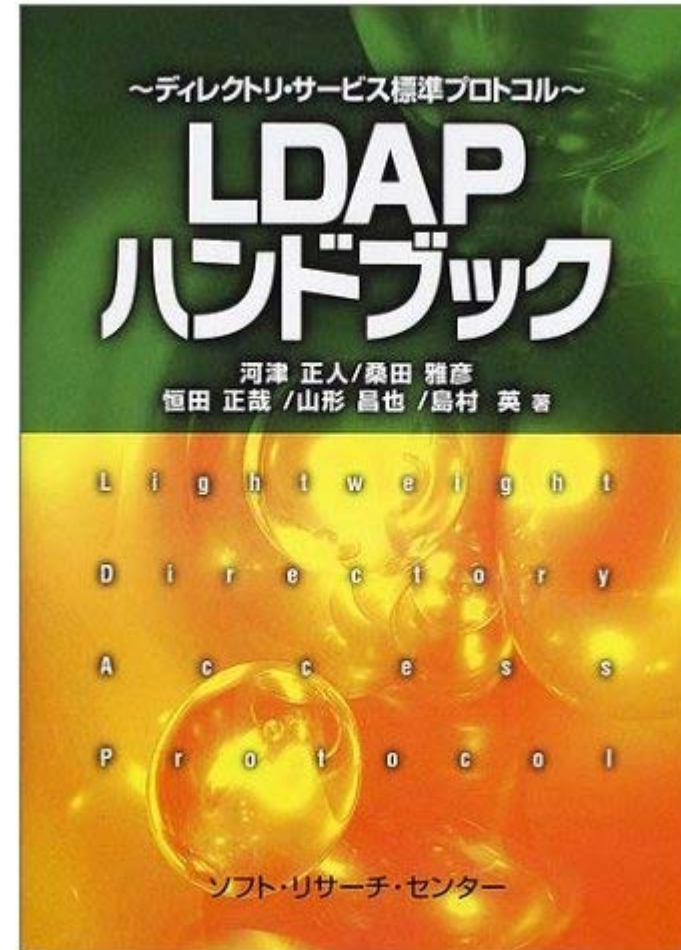
- DSA (Directory Service Agent) :ディレクトリ情報を管理する個々のシステム。ディレクトリはDSAの集合体として構成される。
- DUA (Directory User Agent) :ディレクトリの利用者に代わってディレクトリへアクセスする機能(プログラムやコマンド、ライブラリ)
- LDAPでは単純にLDAPサーバーとLDAPクライアントと呼ぶことが多い



ディレクトリの機能モデル

# LDAP概念を勉強のための参考書

- LDAPハンドブック  
ディレクトリ・サービス標準プロトコル
  - 出版社: ソフトリサーチセンター  
(2002/03)
  - 発売日: 2002/03



# LDAPとRDBMSの違い

- LDAPはネットワークプロトコル、SQLは言語

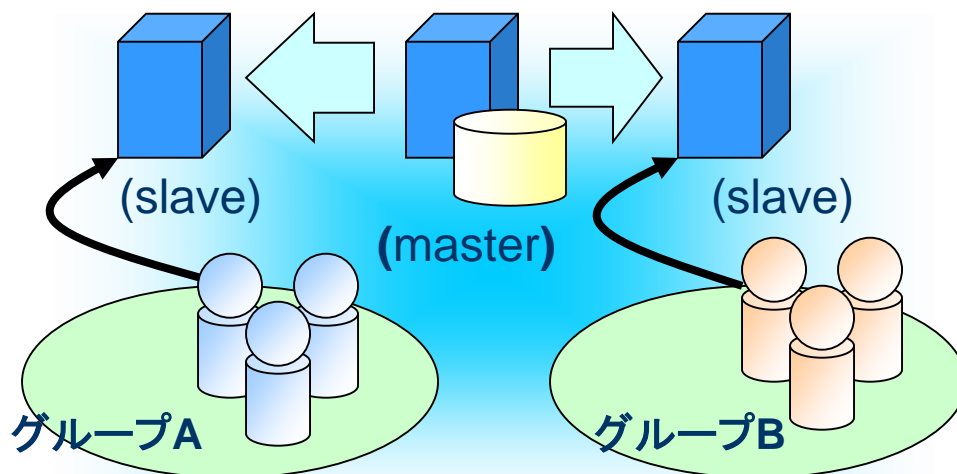
	LDAP	RDBMS
用途	検索性能重視、頻繁な更新には向かない	検索だけでなく頻繁な更新も重視
構造	木構造(行や列といった概念はない)	表構造(行や列が存在)
スキーマ	既存の登録済みスキーマ(ObjectClass)を利用するのが一般的	ユーザが業務に合わせて個別に設計し、利用する
更新	トランザクションの概念はない (トランザクション機能を持った製品もある) 大量更新には向かないので1時間に数件といった更新頻度のものに利用する	トランザクションの概念あり 1秒間に何十、何百もの更新に耐えられる設計となっている
分散	ツリーの枝単位で分散配置が可能	キーの範囲で分散配置が可能
操作	LDAP(ネットワークプロトコル)で操作 プロトコルは単純	SQL(プログラム言語)で操作 複雑な操作が可能
検索手法	木の枝葉をたどるイメージ	表の行を走査するイメージ

# LDAP概念に関する勘違い

- RDBMSは永続的なユーザ情報を蓄えるために使う、LDAPは管理情報を集約するために使う  
(社員DBはRDBMS、全社認証システムはLDAP)
- LDAPは検索重視となっているが、RDBより必ずしも早いわけではない
- LDAPはスケールアウト型負荷分散がやりやすいから
- 更新がすぐに反映されるとは限らない
  - ユーザ追加やパスワード変更がすぐにされないことがある(だからWindowsはパスワードをキャッシュする)
- マルチマスターの利用は要注意
  - トランザクションやロックの概念が弱い
  - uid,gidの自動割り振りをLDAPでやると危険

# 負荷分散方法1:レプリケーション

- 同じ内容のサーバを複数用意する
  - サーバを増やすだけでスケールアウトする
  - 負荷分散装置やldap.confで負荷を分散
  - 1つのサーバが持つデータ量は同じなので規模が大きくなると更新性能が低下
  - Syncreplではサブツリーだけを複製することも可能



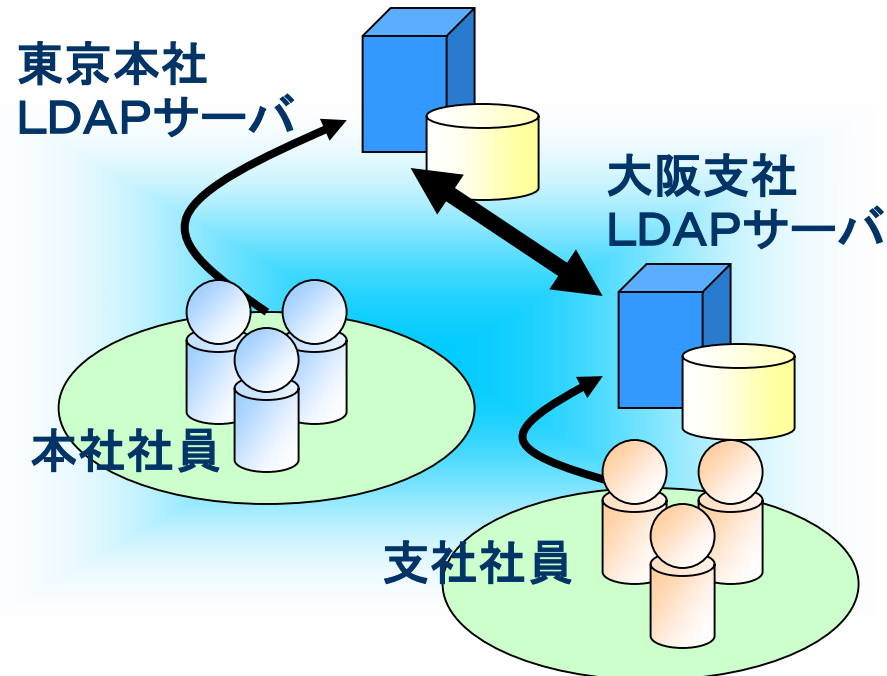


# 負荷分散方法2:リファラル

## ● サブツリー単位でサーバを分散する

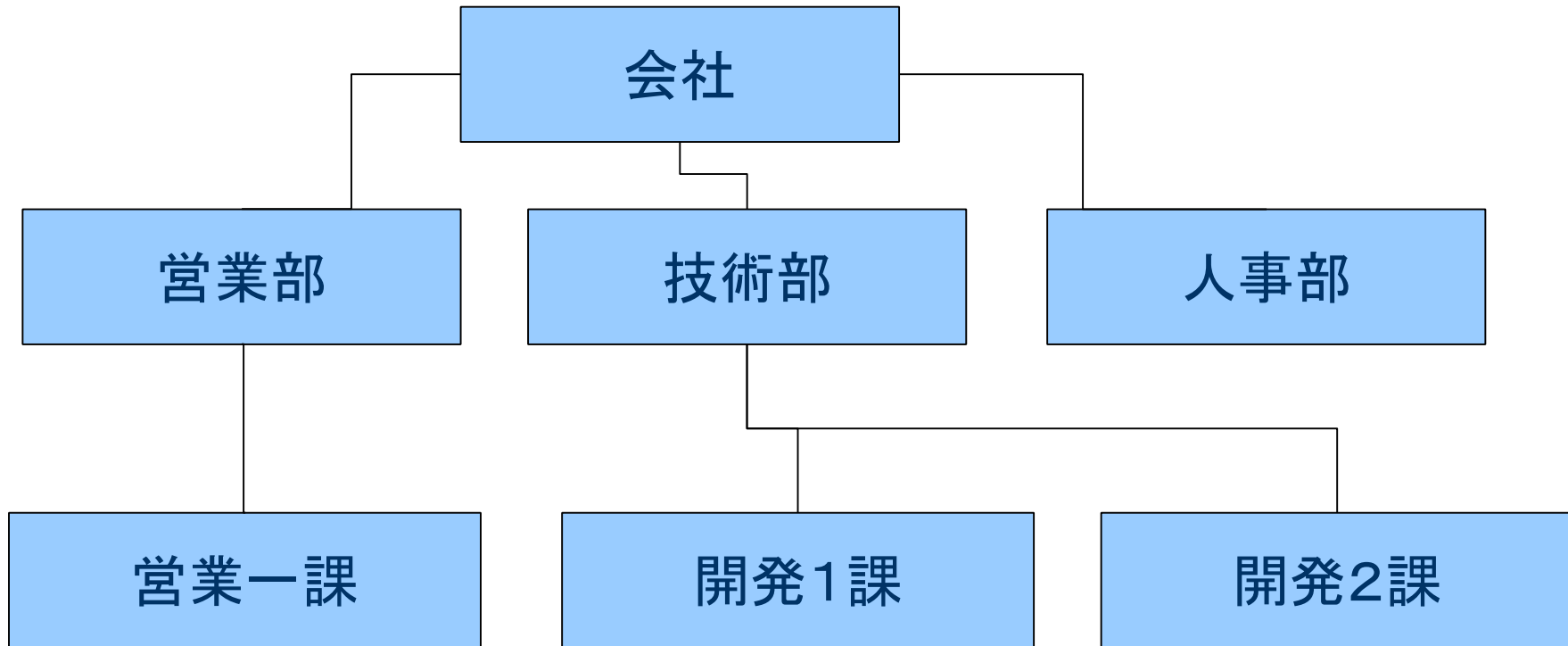
- ldap.confでbaseツリーを変える(負荷分散というよりも管理分散)
- 1サーバがもつデータ量が減るので更新性能も上がる
- referralが返ったら別なサーバを見に行くのはプログラム側の責任

### 分散管理(referral)



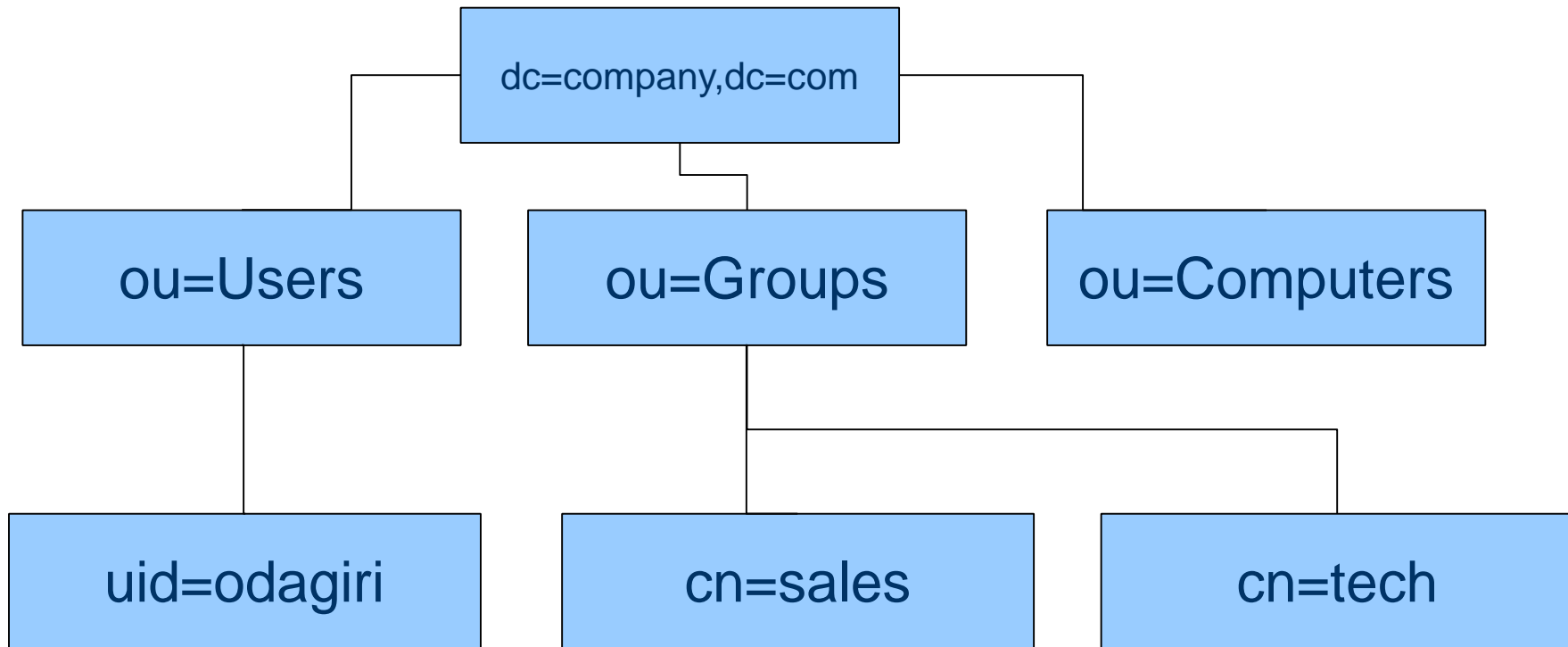
# DIT (Directory information Tree) の概念

- 概念として組織構造をあげる書籍が多いが...



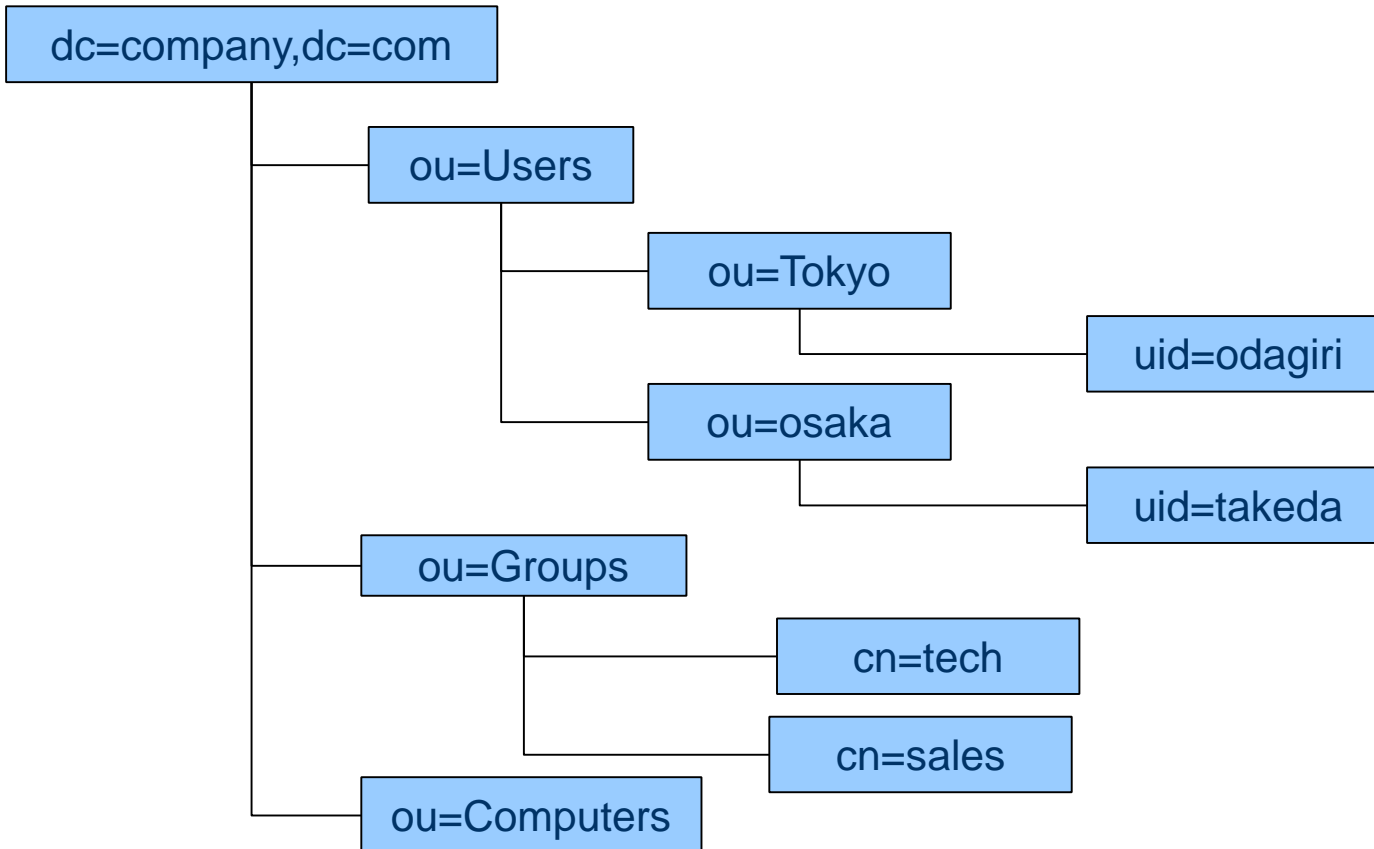
# DIT (Directory information Tree) の概念

- 実構造としては管理単位で分ける



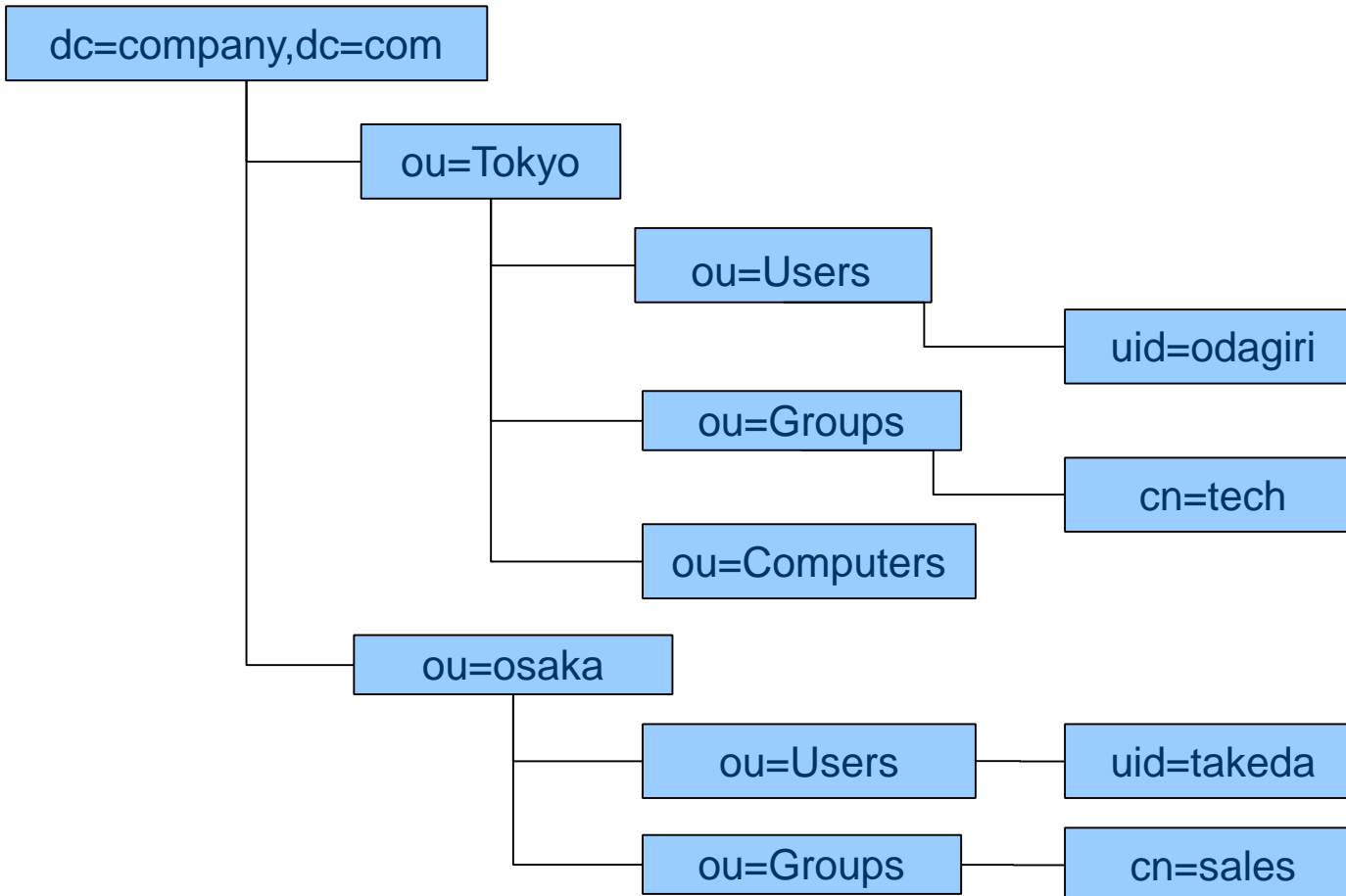
# DIT (Directory information Tree) の設計 (1)

- 組織構造にマッピングしないこと、管理対象で分ける



# DIT (Directory information Tree) の設計 (2)

- 組織構造にマッピングしないこと、管理対象で分ける



# LDAPで何ができるか？

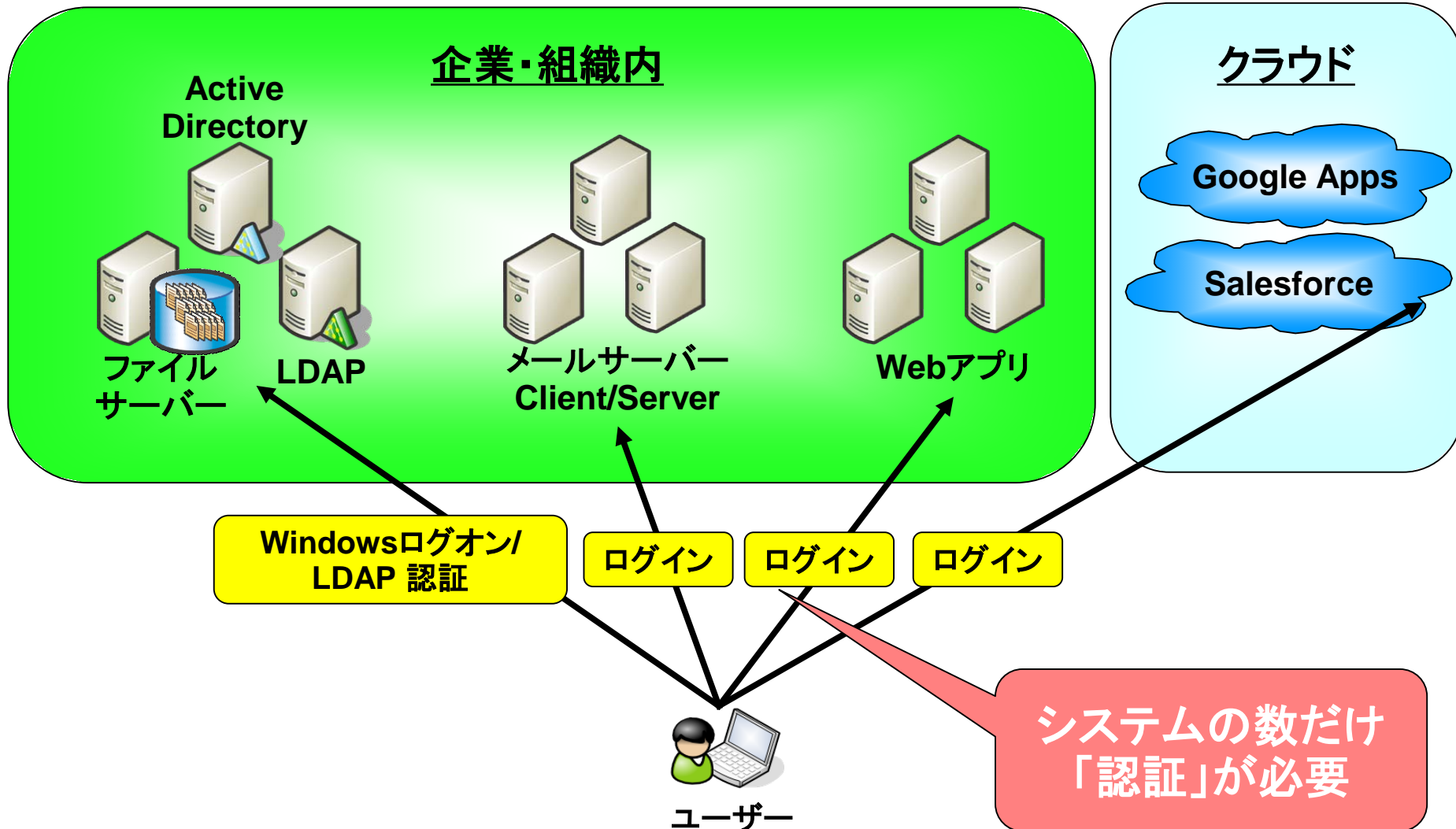
- **Linuxユーザの統合管理  
(Mail,FTP,Telnet,Proxy,sshなど)**
- **Samba/Windowsユーザの統合管理**
- **Webサーバ(Apache)のアクセス制御**
- **電話帳、メールアドレス帳**
- **PKI(公開キー)の保管場所として**
  
- **LDAPのスキーマはむやみに拡張しない  
本当に必要か精査する**

# クラウドと共に普及する シングルサインオン



OSSTech

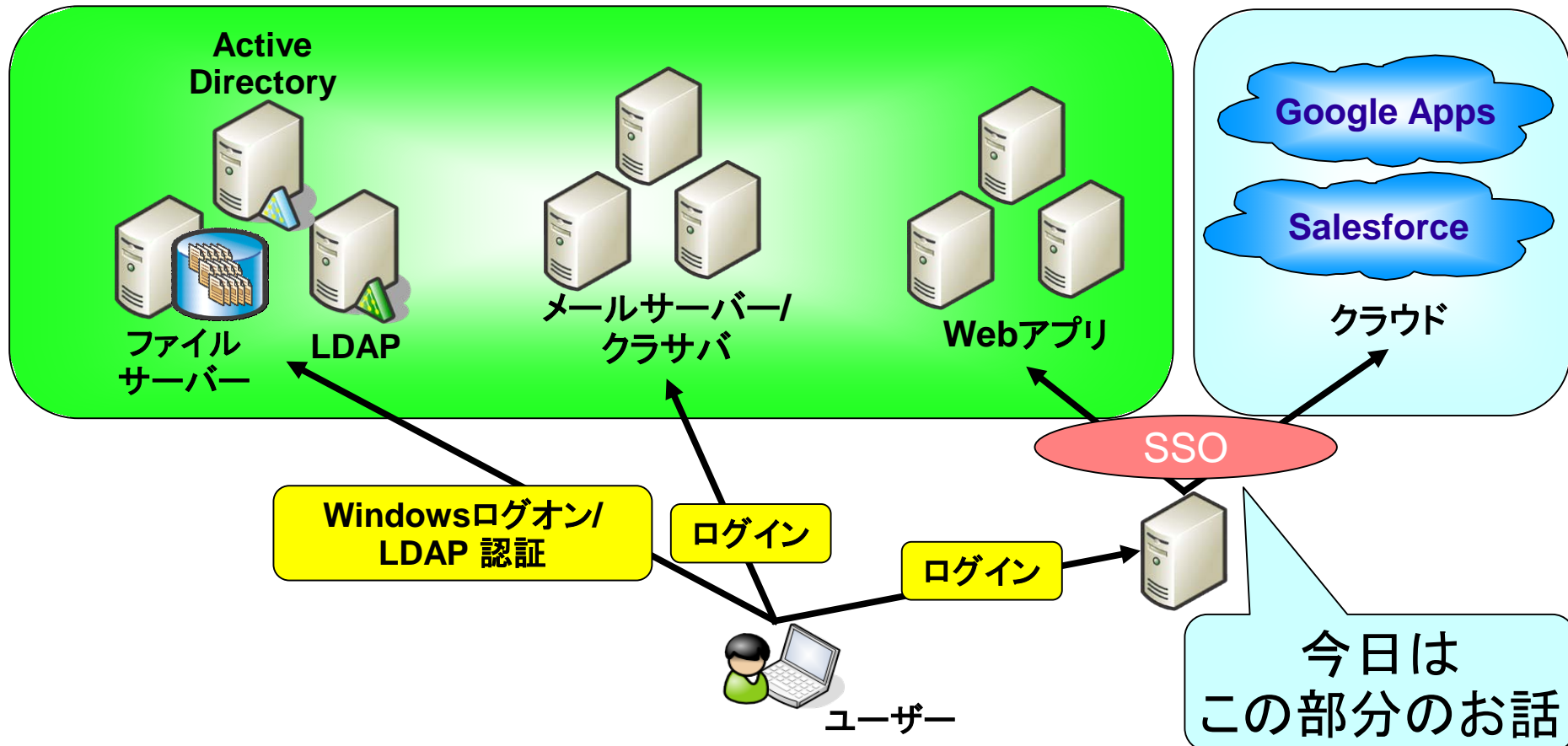
# サービスを利用するには必ず必要な「認証」





# SSO: シングルサインオンとは

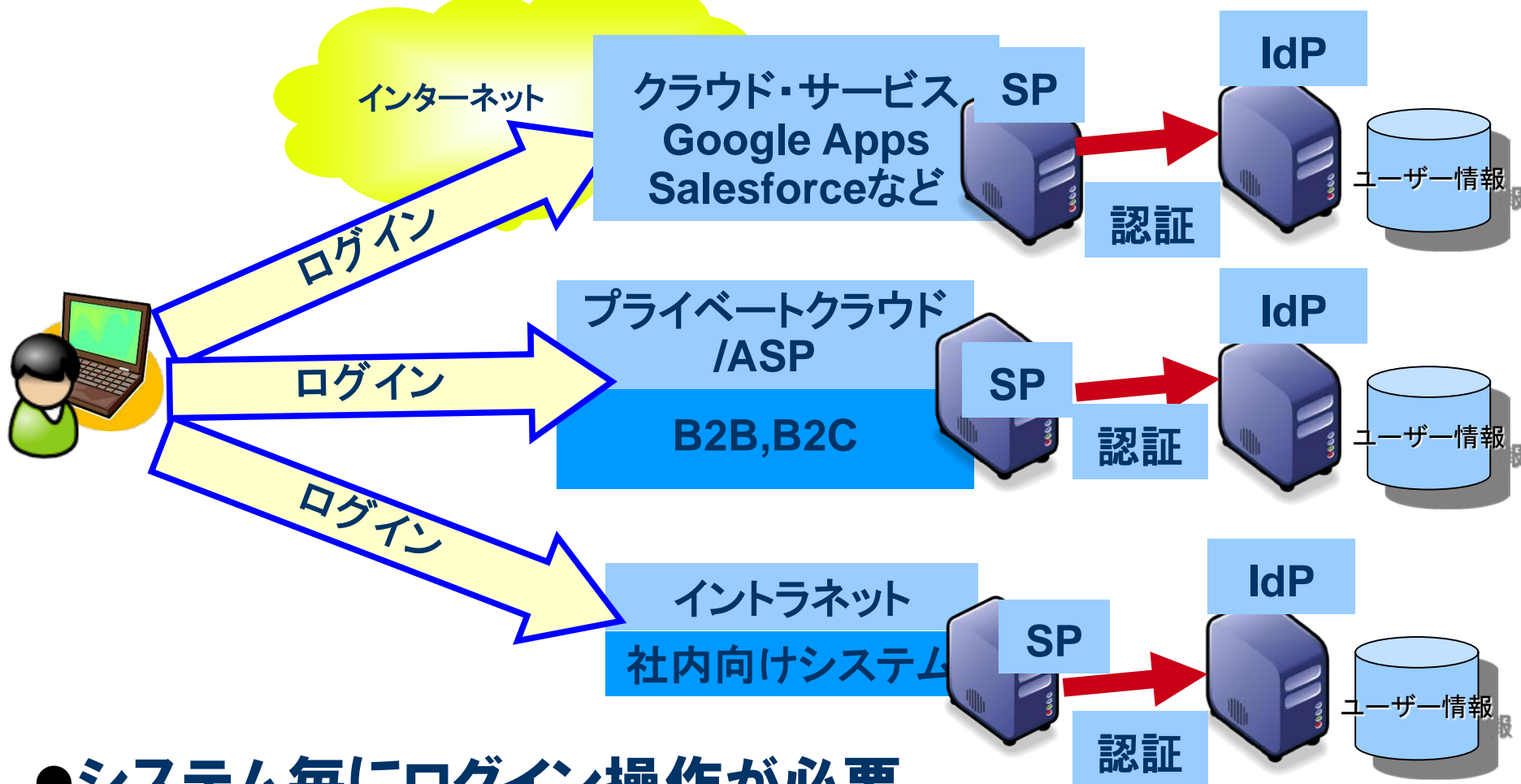
一度のログイン操作さえ完了すれば、複数のアプリケーションに認証操作することなくアクセスすることが可能になる。



# SSO(シングルサインオン)とは

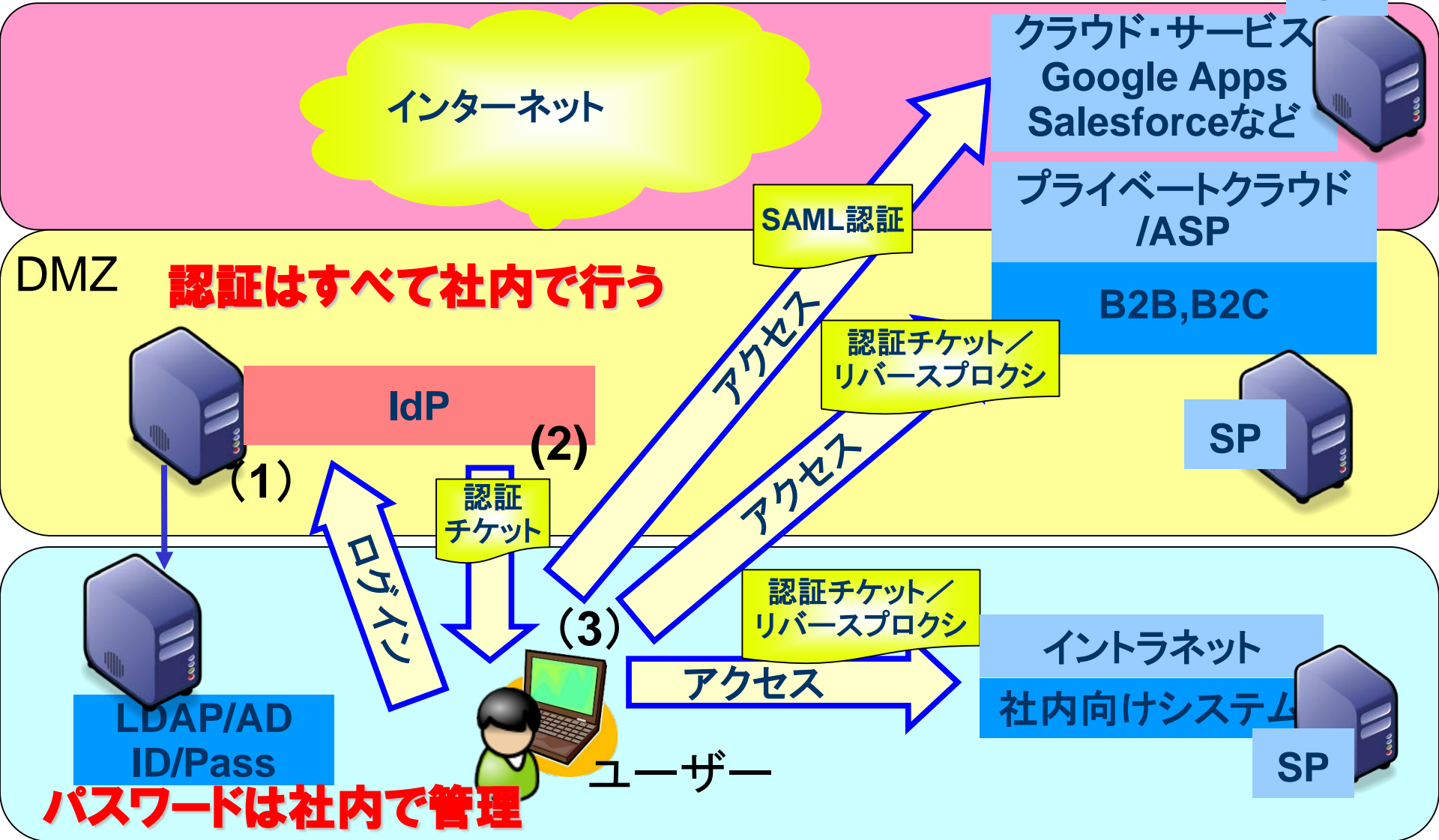
- 1回のパスワード入力で複数のシステムやサービスを同時利用
- 「ID統合を使った統合認証」ではIDとパスワードの管理を1カ所でできるためユーザの追加も楽、社員が退社した場合に1カ所IDを削除すれば、すべてのシステムが利用不可となる
- 近年クラウドサービス(SaaS, PaaS, IaaS, HaaSなど)の普及により、(社外にある)サービス毎にID／パスワードを登録しなければならないケースが増えており、「ID連携による統合認証」を使わざるを得ないケースが増えている
- ところがこのID連携が費用の問題や技術的な問題で完全に実現されていない場合、例えば社員が退社した時に社内システムのIDを削除しても、SaaS側のIDが残っているとクラウド側のシステムは社外から使えてしまう、といった問題が起きてしまう

# クラウドで統合認証ができていないと...



- システム毎にログイン操作が必要
- クラウドにID/パスワードとパスワードを置く必要がある  
(パスワードを社外に置くと不正ログインされる危険性が高い)

# クラウドで統合認証とSSOを実現する



# シングルサインオンを実現するソフトウェア

## OpenAM

- Webアプリケーションにおけるシングルサインオンを実現するためのプラットフォームとなるソフトウェア
- 現在はオープンソースだが、元はSun Microsystems社の商用製品（Access Manager）
- 弊社で製品パッケージを提供



## Shibboleth.

- SAMLを扱えるオープンソースのソフトウェア
  - Shibboleth1.3以前のバージョンがSAML1.1を実装
  - Shibboleth2.0よりSAML2.0を実装
- 学認フェデレーションでの主な認証ミドルウェアとして使用

# シングルサインオン 技術動向



OSSTech

# SSO(OpenAM)導入動向

- クラウドの普及により、SSO(シングルサインオン)が急速に普及中
- IaaSやPaaSも増えつつあるが、やはりSaaSのGoogle Apps(大学／企業)とSalesforce(企業)をまず導入するケースが多い
- 企業ではSalesforceのセキュリティ強化を目的にOpenAM導入するケースが多い
- 大学ではGoogle AppsとイントラネットやShibbolethを連携させるケースが多い
- 企業ではM&Aや会社合併のために増えすぎたアプリやIDを統合するためにSSOを導入
- IaaSやPaaSも普及し始め、これらの上で構築された社内向け個別アプリのSSOも普及しだしてきた。

# OpenAMで実現する シングルサインオン・ハブ



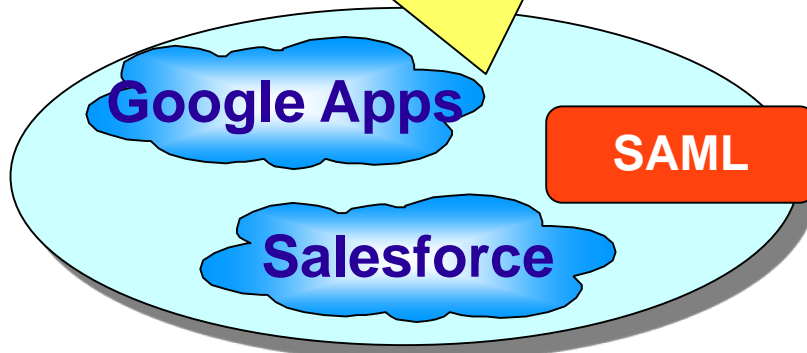
OSSTech

オープンソースのOpenAMだから  
高機能・安価に実現できる



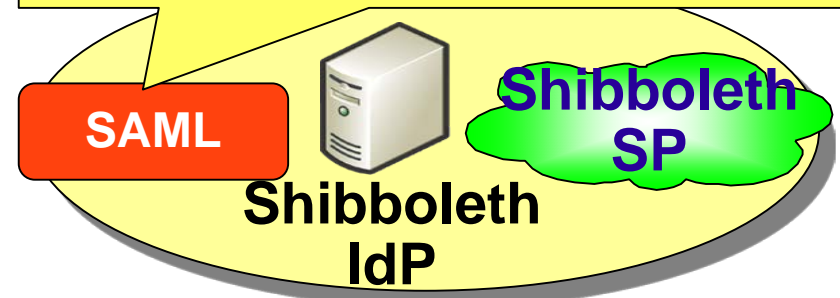
# 混在する複数のSSO環境

SAML IdP を導入して  
SSO を実現



クラウドSSOセグメント

Shibboleth IdP で SSO を実現  
(Shibboleth は SAML を利用し  
ているが、仕様上 OpenAM では  
代替不可能)



学認 (Shibboleth)  
SSOセグメント

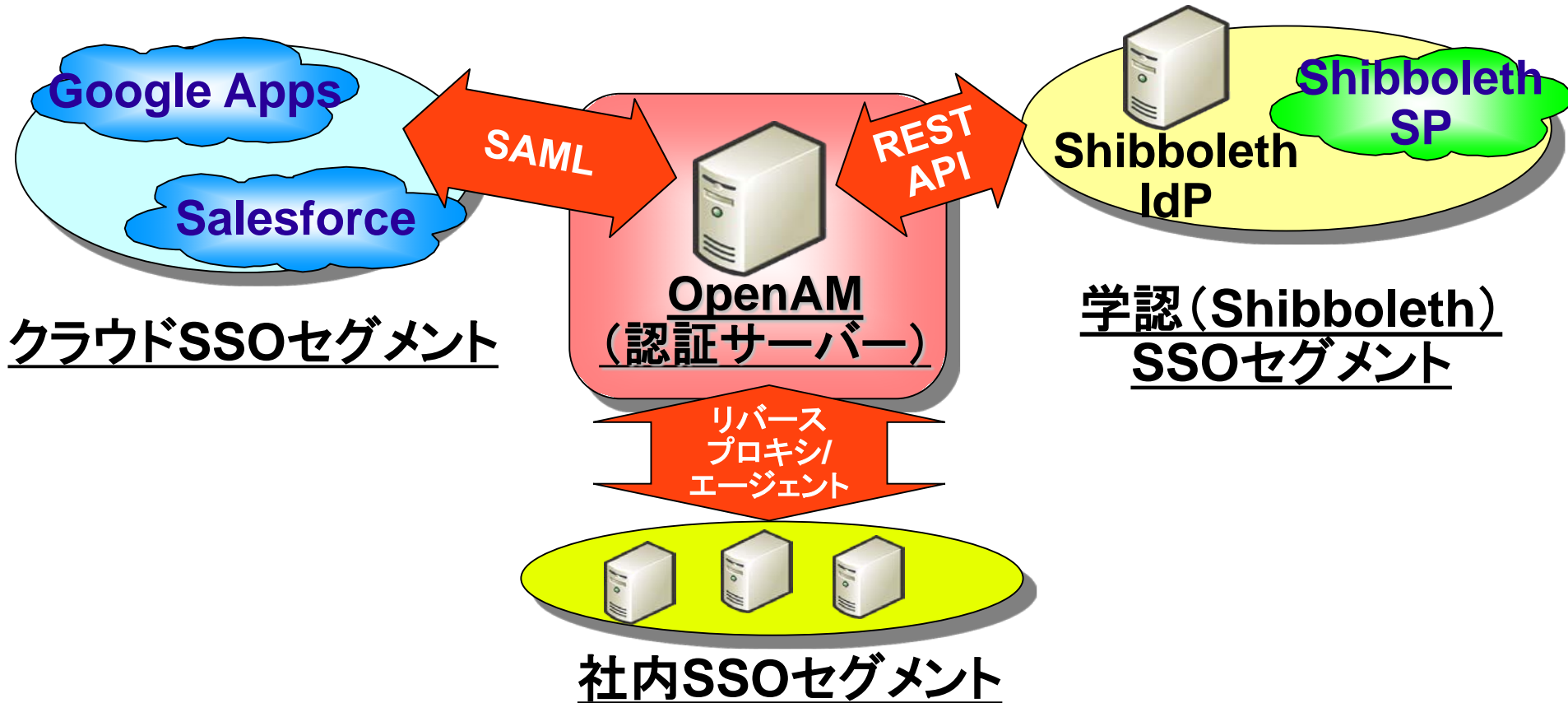
リバースプロキシ/  
エージェント



社内SSOセグメント

大幅な改修はしたくないため、エー  
ジェント型/リバースプ  
ロキシ型で SSO を  
実現

# OSSで実現するシングルサインオン・ハブ



SSO セグメントを結合するハブとして OpenAM を利用。  
 ユーザーは OpenAM へのログインさえ完了していれば、  
 全てのアプリに SSO 可能

## シングルサインオン・ハブを実現するための機能

- **認証機能**

- ユーザーの本人性を確認する。セキュリティ強化のために、多要素認証が望ましい。

- **ユーザー情報保存機能**

- 認証情報や他システムに連携するユーザー情報を保存する

- **外部システムと連携可能なインタフェース**

- フェデレーション(SAML, OpenID, OAuthなど)
- REST API
- SDK

# シングルサインオン方式

# シングルサインオン方式の詳細(1)

- フェデレーション: SAMLによるシングルサインオン
  - Secure Assertion Markup Language
  - 認証、認可、ユーザ属性情報などをXMLで送受信するためのフレームワーク
  - 標準化団体OASISにより策定
  - GoogleApps, Salesforceなどが採用
  - 今後OpenIDの普及によりOpenAMでOAuth実装予定
- エージェント方式
  - SSO対象のWebアプリが動作するサーバー上にアクセス制御用のモジュールを配置する方式
  - サーバーのバージョンに影響を受ける

# シングルサインオン方式の詳細(2)

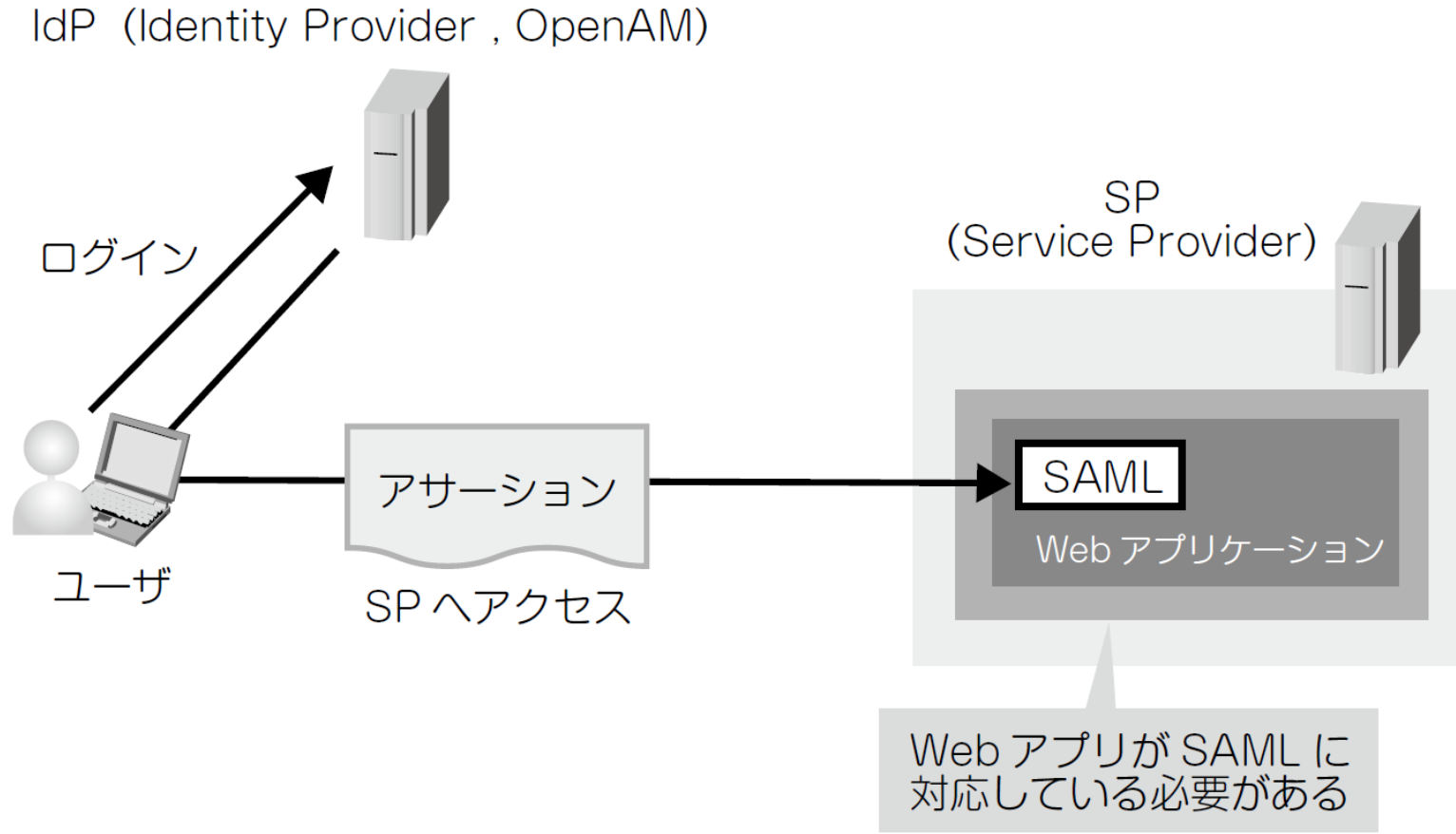
- リバースプロキシ方式

- リバースプロキシを使用してアクセス制御を行う
- ユーザーデータの受け渡しはHTTPヘッダーを利用
- SSO対象Webアプリのバージョンや設定変更の影響が少ない
- リバースプロキシが性能上のボトルネックになる可能性がある

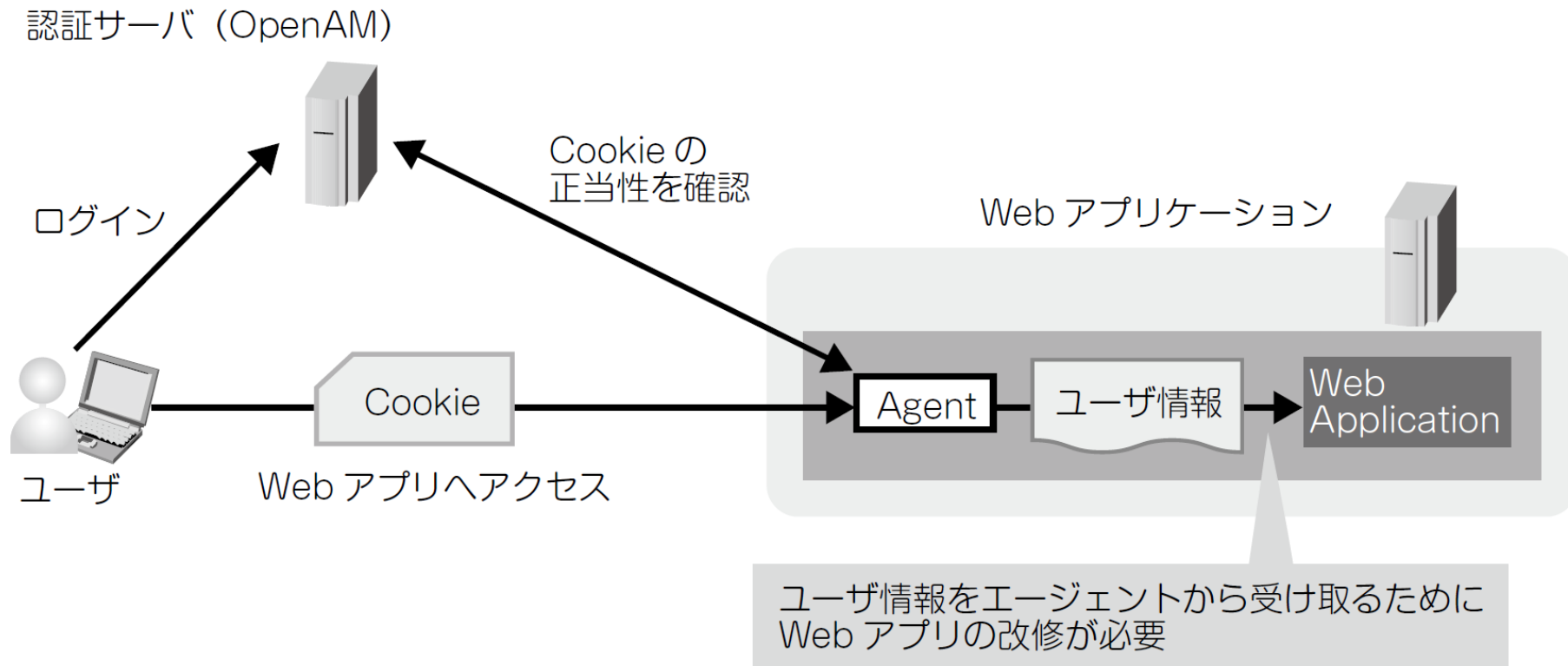
- 代理認証方式

- SSO対象Webアプリの既存ログイン画面に対して、OpenAMがユーザーの代理でログインID/パスワードを送信する
- SSO対象Webアプリの改修が不要
- 細かなアクセス制御はできない(ログイン処理の代理実行のみ)

# フェデレーション: SAMLによるシングルサインオン

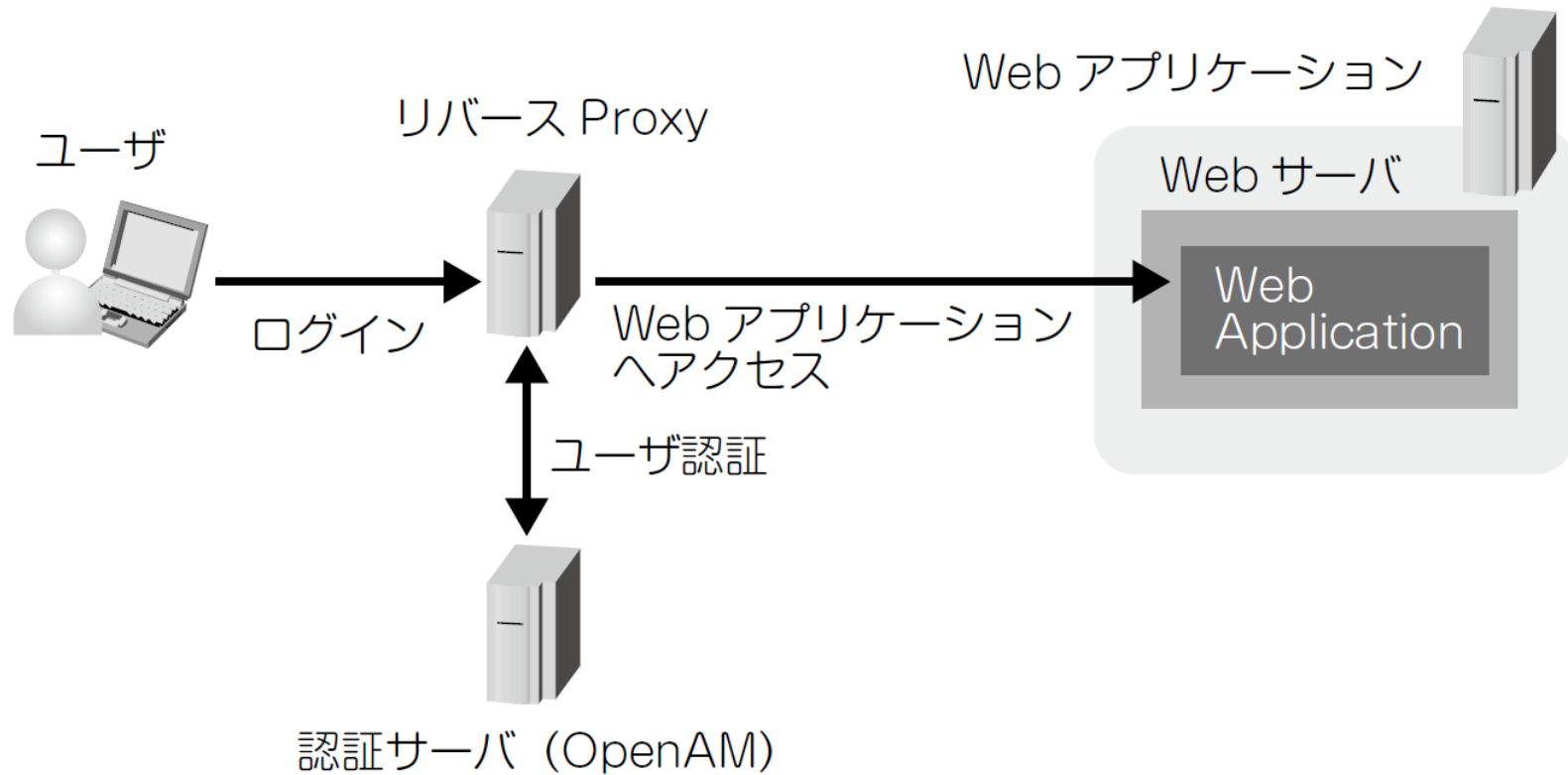


# エージェント型



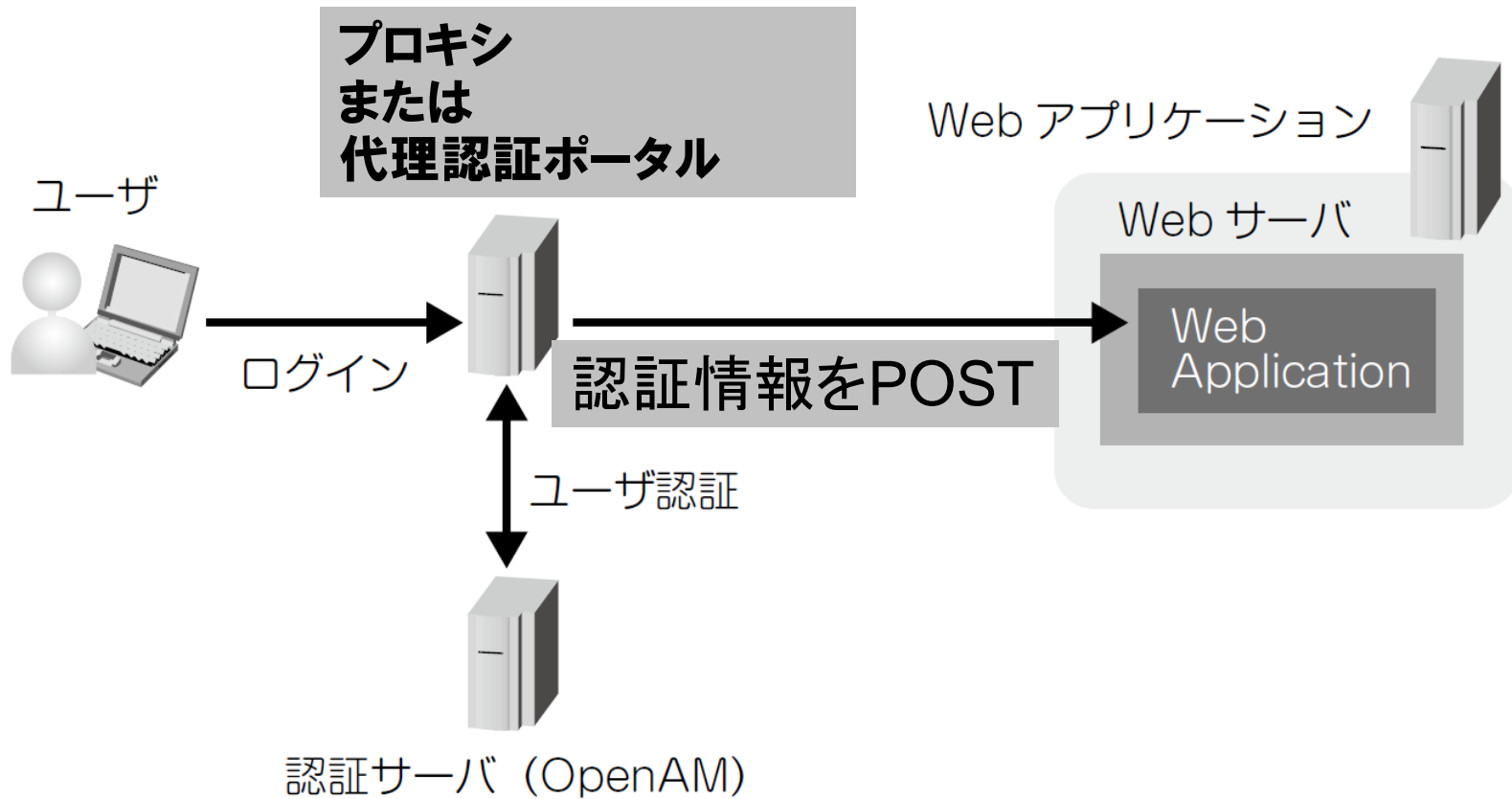


## リバースプロキシ型



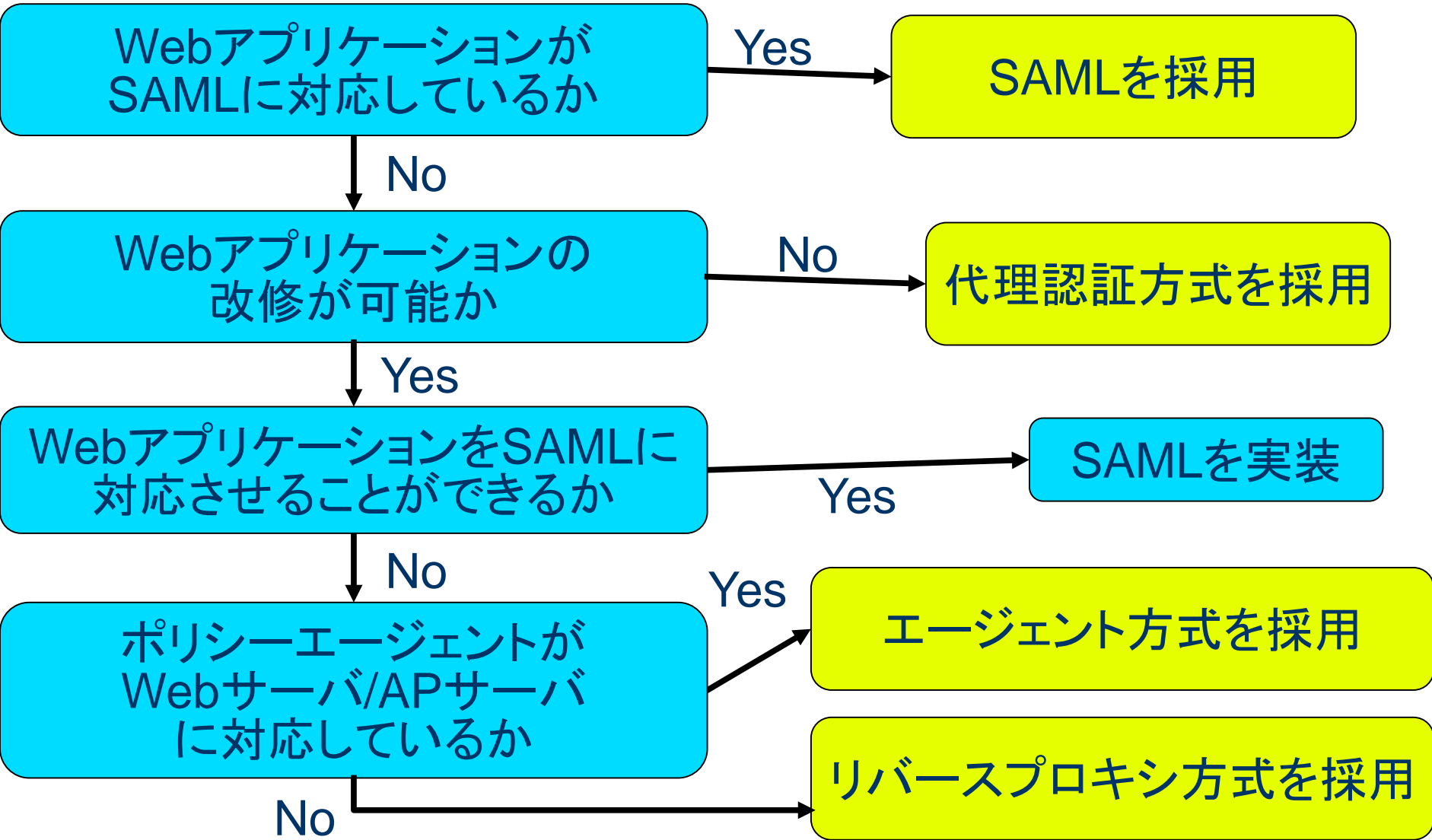
- 後方のサーバを仮想的に1台に見せることも可能
- 認証とサーバへのアクセス制御はプロキシサーバで行う
- 後方のサーバは認証なしもしくはBasic認証でアクセス可能

## 代理認証方式



- 認証サーバで認証したら後方のサーバへ認証情報をPOSTして認証する
- 後方のサーバが独自の認証画面を持っていてもSSO可能

# シングルサイン方式の採用基準



# 本当はやってはいけない「代理認証」

「既存アプリに手を入れられない」という理由で代理認証を採用するユーザーは多いが本当はやってはいけない！

- IDとパスワードを(HTTPSでも)ネットワークに何度も流すのは良くない。(SSO入り口の1カ所に限定すべき)
- 代理認証はイントラネットのみに限るべき
- クラウドへの代理認証は危険
- SAMLに対応しているGoogle AppsやSalesforceに対して、代理認証は絶対にやってはいけない！  
(SAMLを使ってIdPを社内に置けばパスワードはクラウドに流れない)

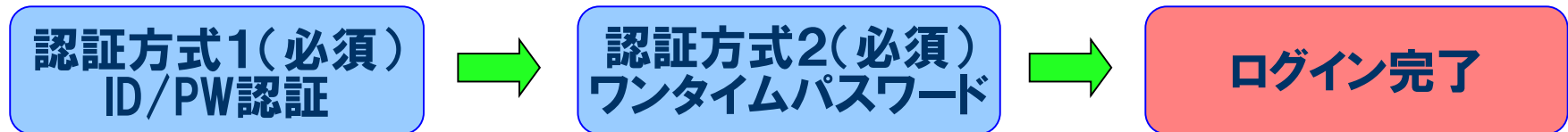
# OpenAMの認証方式 (多要素認証)



OSSTech

# OpenAMの機能 - 認証連鎖

- 多要素認証の必要性
  - 複数の認証方式を組合わせて認証を行うことにより個々の認証方式の欠点を補完
- 認証連鎖
  - 複数の認証方式を組み合わせて利用可能
  - 認証方式にはそれぞれ適用条件を指定する
    - 必須: 失敗したらそこで終了
    - 十分: 成功したらそこで終了
    - 必要: 成功しても失敗しても次に継続
    - 任意: 認証結果には関係しない付随的な処理



# 多要素認証

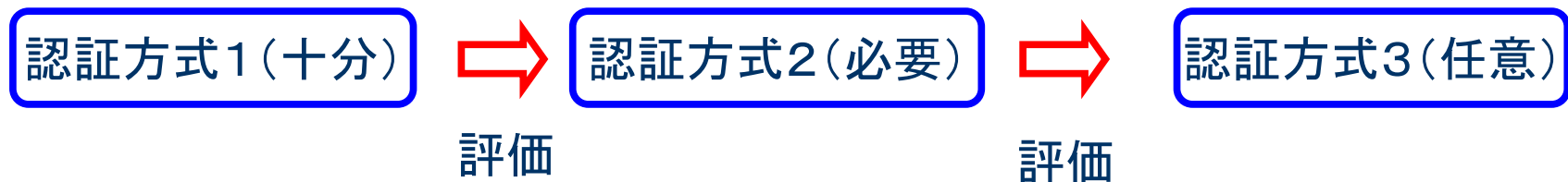
## 複数の認証方式を組合わせて認証を行うことにより 個々の認証方式の欠点を補完

- 厳密なユーザ認証
  - 異なるタイプの認証方式を組合わせることが重要
- 使い勝手の向上
  - いつも同じ認証方式が使えるとは限らない
  - 状況により要求される認証の精度が異なる
- 認証方式間での連携
  - 組合わせて使うことを前提にしている認証方式もある

# 認証連鎖

認証方式を組み合わせる方法を指定する

- 認証方式にはそれぞれ適用条件を指定する
  - 十分: 成功したらそこで終了
  - 必要: 成功しても失敗しても次に継続
  - 必須: 失敗したらそこで終了
  - 任意: 認証結果には関係しない付随的な処理
- 認証成功時には認証方式に応じて認証レベルが設定される






# 例1. Windows Desktop SSO

Windows Server  
2000/2003/2008

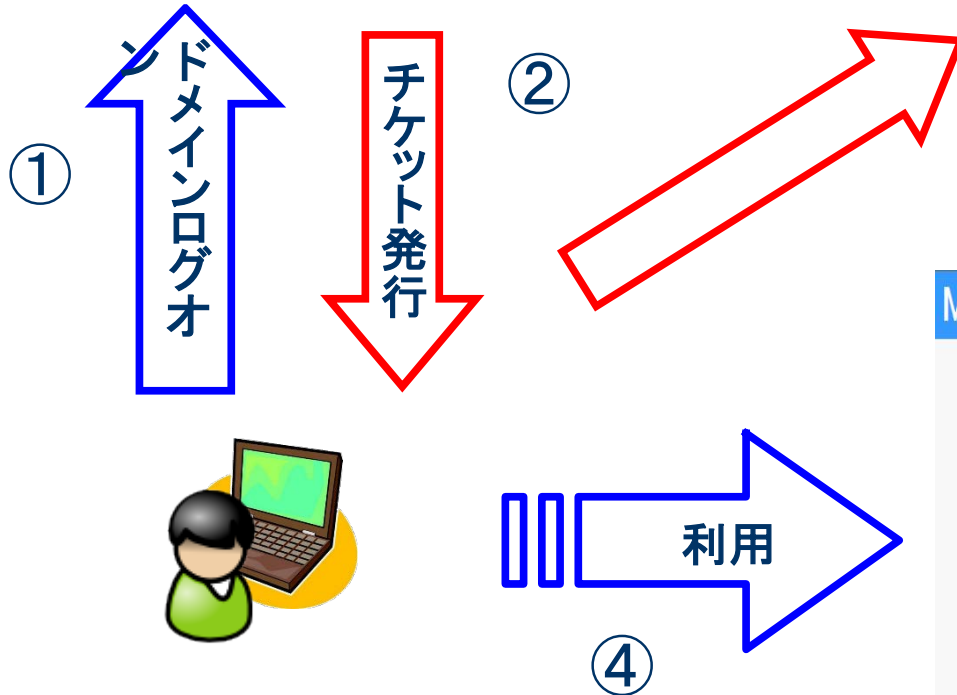


Active Directory

自動チケット送付



OpenAM



③ 認証、認可、属性情報

MosP勤怠管理 メニューガイド v3.20 ユーザー名: 人事 一郎

メニューガイド

勤怠入力

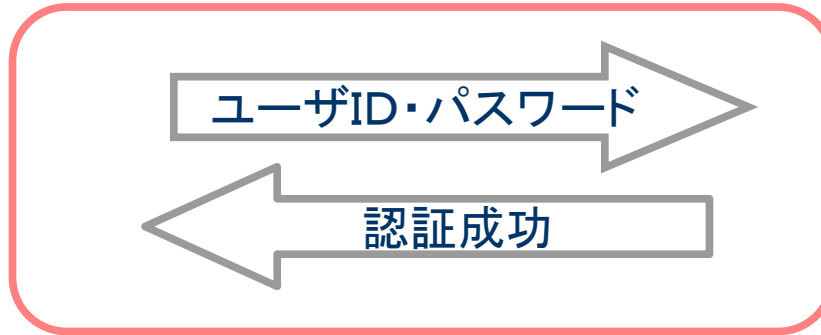
勤怠管理 給与管理 人事管理

# 例1. Windows Desktop SSO

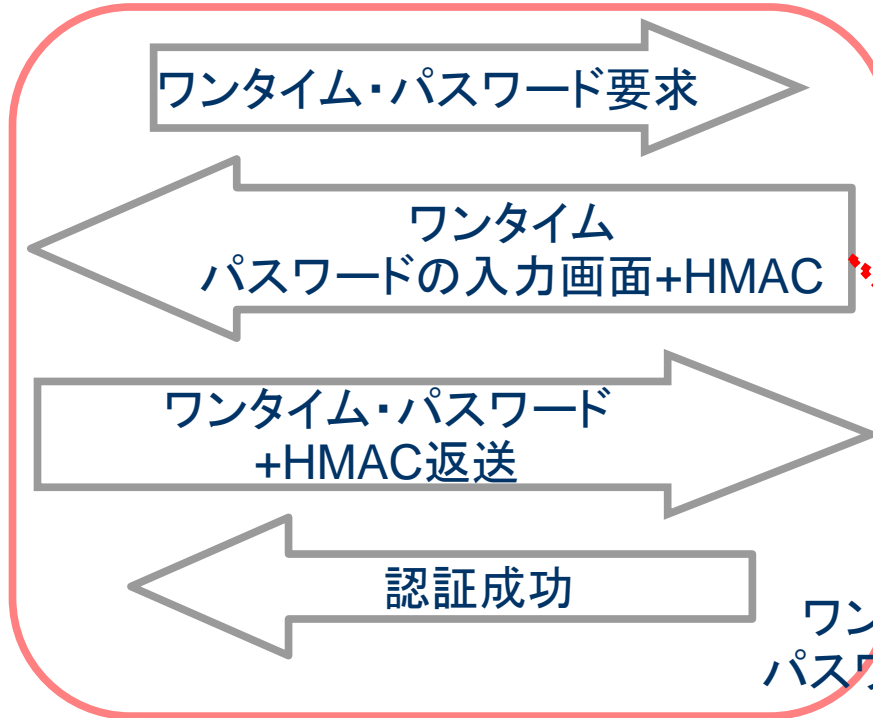
WindowsドメインログオンするだけでWebアプリケーションにもSSOが可能になる便利な方式

- **いつも、全てのユーザがドメインログオン可能であるとは限らない**
  - リモート・アクセスの場合
  - 非常勤社員の場合
- **通常のユーザID・パスワードによる認証と組み合わせて以下のように認証連鎖構成する**
  - Windows Desktop SSO: 十分
  - ユーザID・パスワードによる認証: 必須

## 例2. 携帯電話を使ったワンタイム・パスワード



通常のユーザID・パスワード  
による認証



同時に携帯電話へ  
ワンタイム・  
パスワードを送付

ワンタイム・  
パスワード認証

## 例2. 携帯電話を使ったワンタイム・パスワード

- 所持物認証と知識認証の組み合わせによる厳密なユーザ認証が可能
- 携帯電話を使うことによる利点
  - 導入コストの低減
  - 所持品の軽減
- フィッシングへの対応
  - HMAC(RFC2104:Keyed-Hashing for Message Authentication)を利用
  - 両方のパスワードが盗まれた場合は問題
  - 参考:RSAセキュリティ(株)による月例記者会見

[http://internet.watch.impress.co.jp/docs/news/20100728\\_383861.html](http://internet.watch.impress.co.jp/docs/news/20100728_383861.html)

# 応用例

- Windows Desktop SSOによる認証は便利なのでぜひ使いたいが全てのユーザがドメインログオン可能とは限らない
- ワンタイム・パスワードは厳密な認証が出来る点は良いが、いつも携帯電話を開いてパスワードを確認するのは面倒だ



- **2つを組み合わせることにより便利かつ厳密な認証を行うことが可能**
  - Windows Desktop SSO: 十分
  - ユーザID・パスワードによる認証: 必須
  - ワンタイム・パスワードによる認証: 必須

# 学術認証フェデレーション

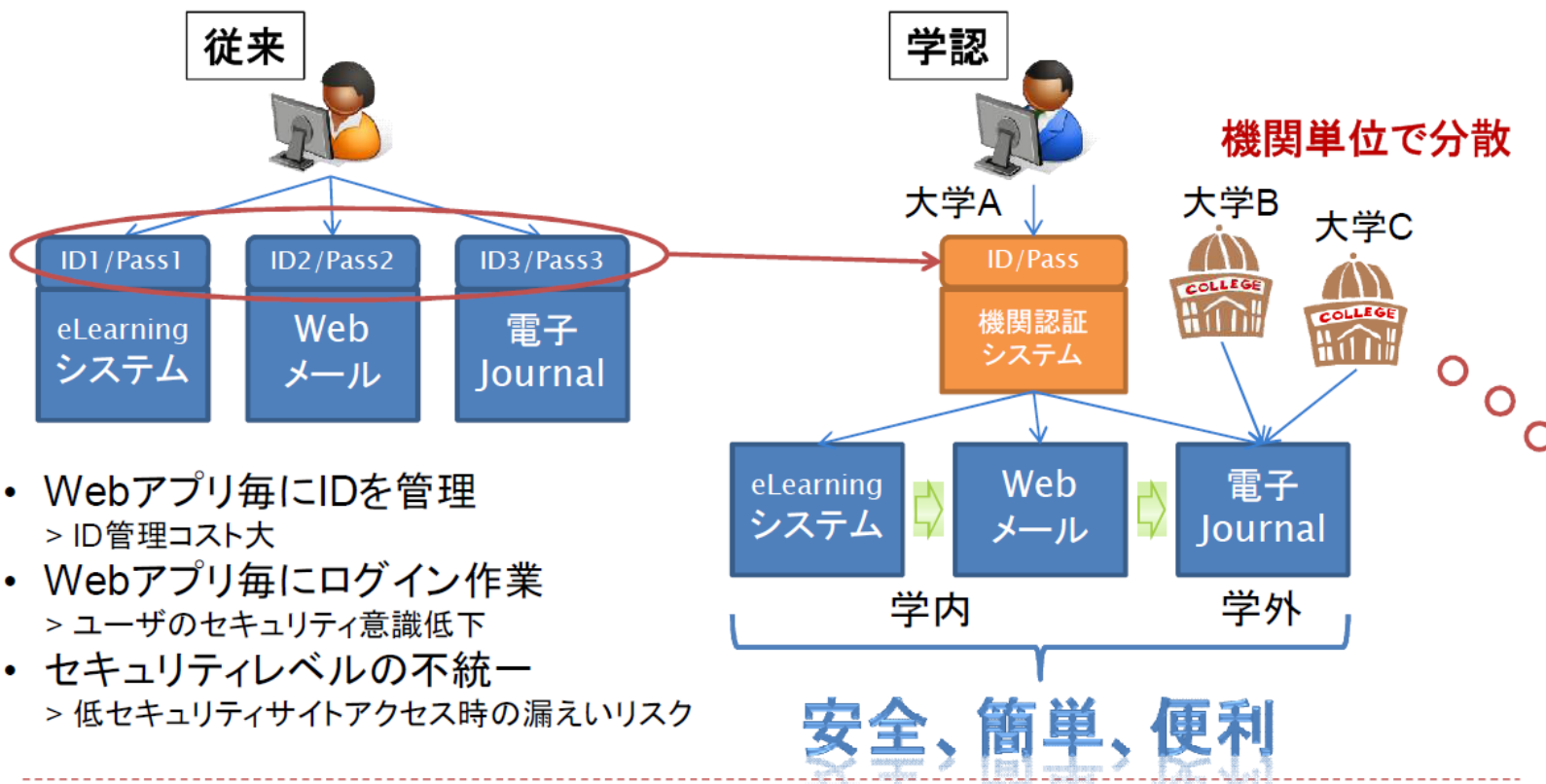
## 学認：GakuNinとは？



GakuNin

## 学術認証フェデレーション「学認」とは

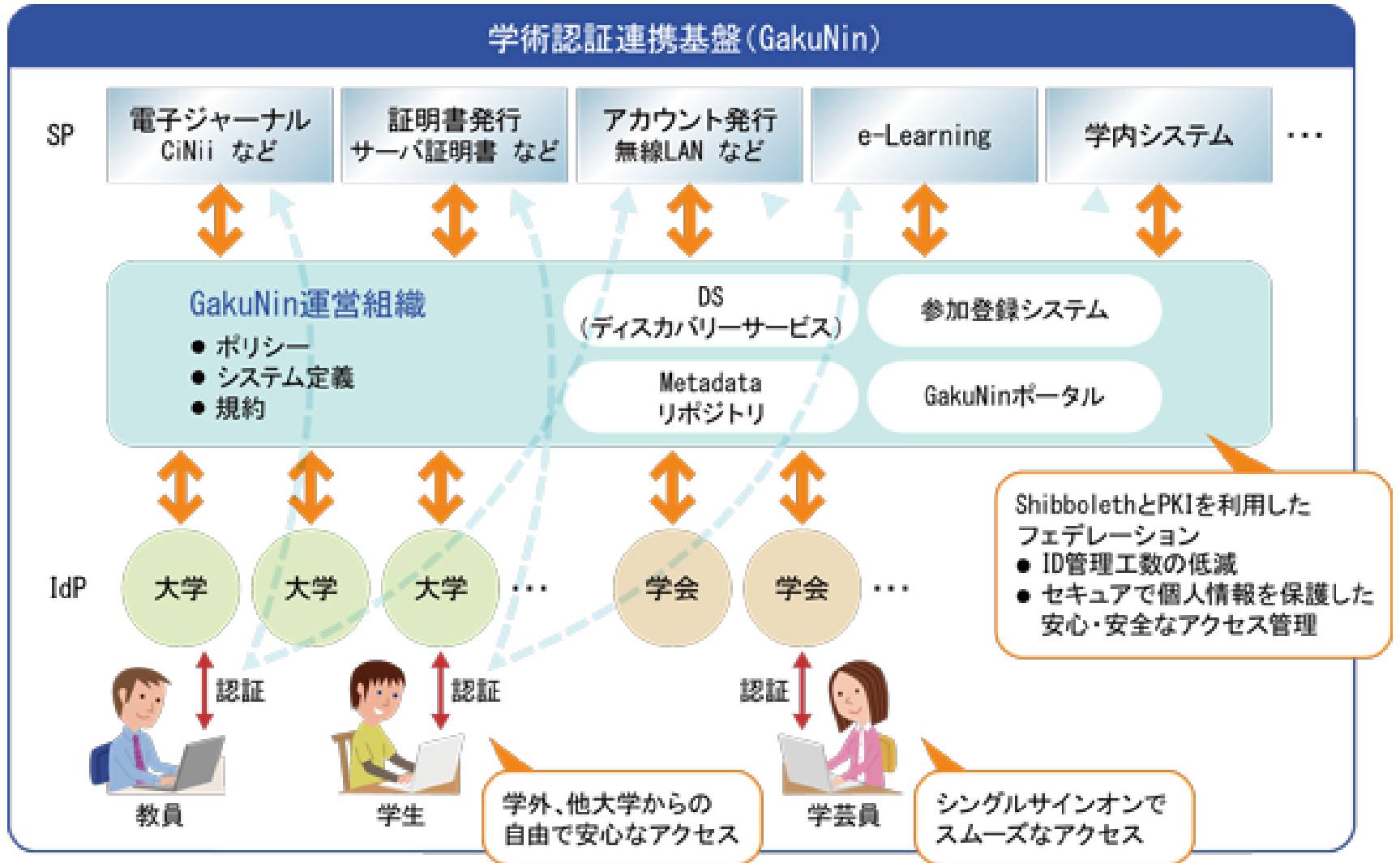
- ▶ Webアプリケーションへのシングル・サイン・オン(SSO)技術を、組織を越えて活用する分散型認証基盤



- Webアプリ毎にIDを管理  
> ID管理コスト大
- Webアプリ毎にログイン作業  
> ユーザのセキュリティ意識低下
- セキュリティレベルの不統一  
> 低セキュリティサイトアクセス時の漏えいリスク

参考)「学術認証フェデレーションシンポジウム」の資料より

<https://www.gakunin.jp/docs/open/3>



参考) 学術認証フェデレーションの資料より

<https://www.gakunin.jp/>



# 学認で利用されている技術

- SAML( Security Assertion Markup Language )
  - 認証情報の連携を行うプロトコル
  - 学認のSP - IdPのやりとりはSAMLで行うと取り決め
- 学認に参加し、SPやIdPと連携するためには・・・
  - SAMLを扱える認証基盤の構築が必要
  - ソフトウェアはSAMLを扱えれば何でも良い
- しかし、実際はほとんどShibbolethで構築
  - 学認のシステム運用基準で推奨されている

GakuNin では、フェデレーション内で利用するソフトウェアとして、上記プロトコルの実装例であるShibboleth を利用することが推奨される。

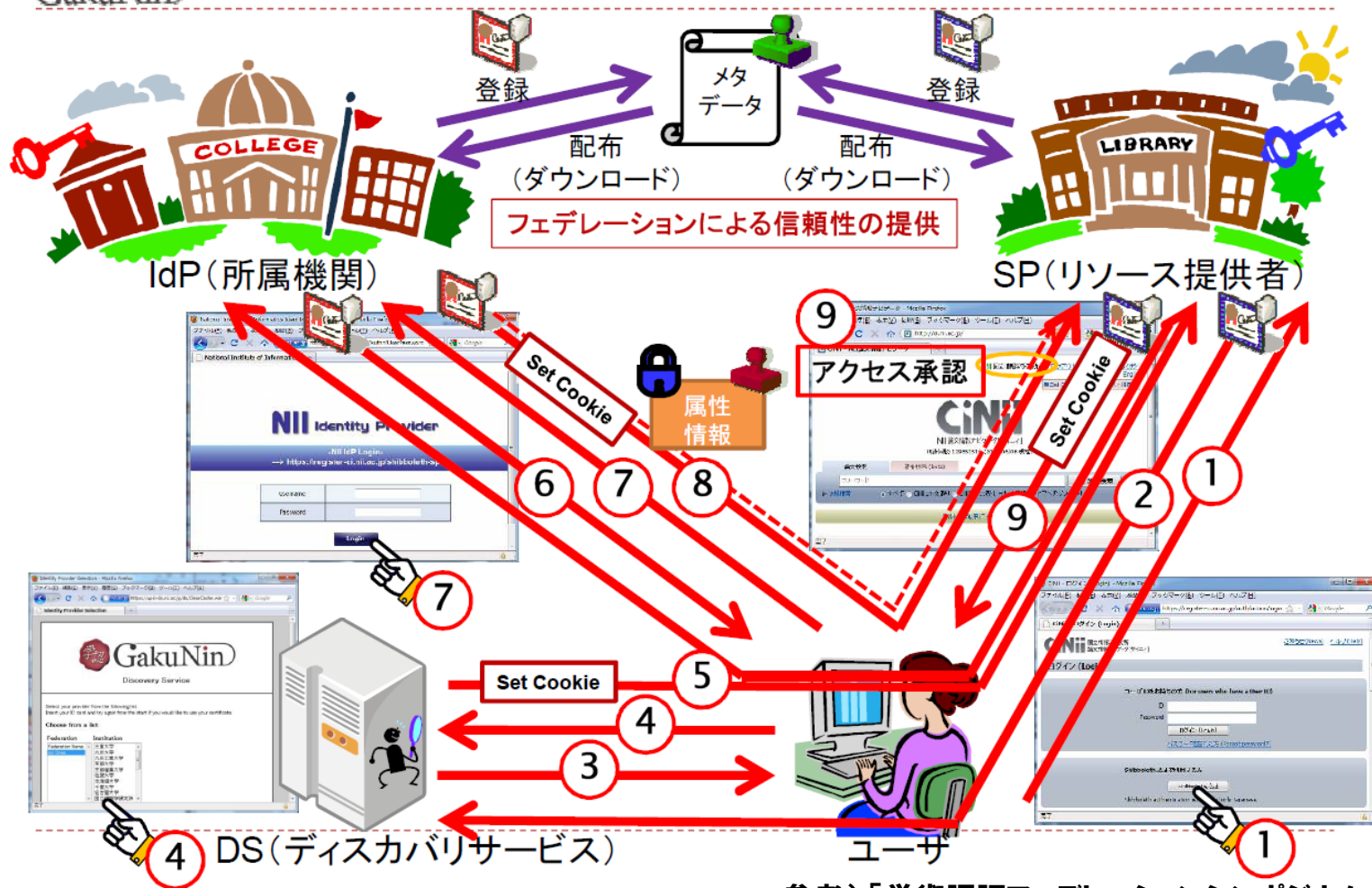
(※上記プロトコルとはSAMLのこと)

学術認証フェデレーションシステム運用基準(Ver 1.2)より引用  
[https://www.gakunin.jp/docs/files/GakuNin\\_System\\_SpecV1.2.pdf](https://www.gakunin.jp/docs/files/GakuNin_System_SpecV1.2.pdf)

- 学認のWebページにShibbolethの情報があり、構築しやすい



## Shibbolethの動作の仕組み



参考)「学術認証フェデレーションシンポジウム」の資料より

<https://www.gakunin.jp/docs/open/3>

# OpenAMとShibbolethの連携

# 学内の認証をShibbolethで？ (1)

- 学認へ参加する場合はShibbolethで構築するのが良い
  - Shibbolethが学認推奨のソフトウェアである点
  - メタデータの取扱いや属性情報の用意等、Shibboleth独自の機能を使った方が運用上において楽な点

【例えばOpenAMで学認に参加することを考える】

SAMLでやりとりを行うため、OpenAMと学認との連携は技術的には可能。

しかし、OpenAMで学認との連携するシステムを運用していくためには、Shibbolethの独自機能を補う仕組みを用意する必要がある。

- Shibbolethによる学内のシングルサインオン
  - 学認用に導入したShibboleth IdPを使い、学内のシングルサインオンを実現できないか？(せっかくシングルサインオン製品を導入したのだから)

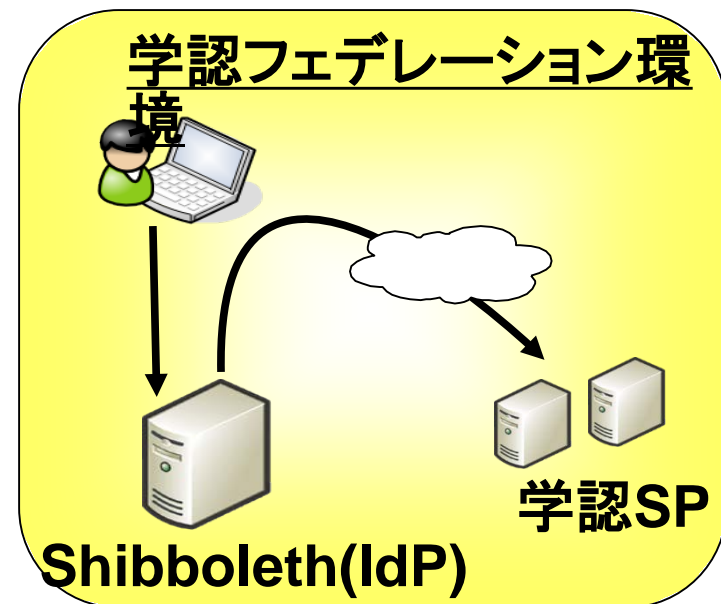
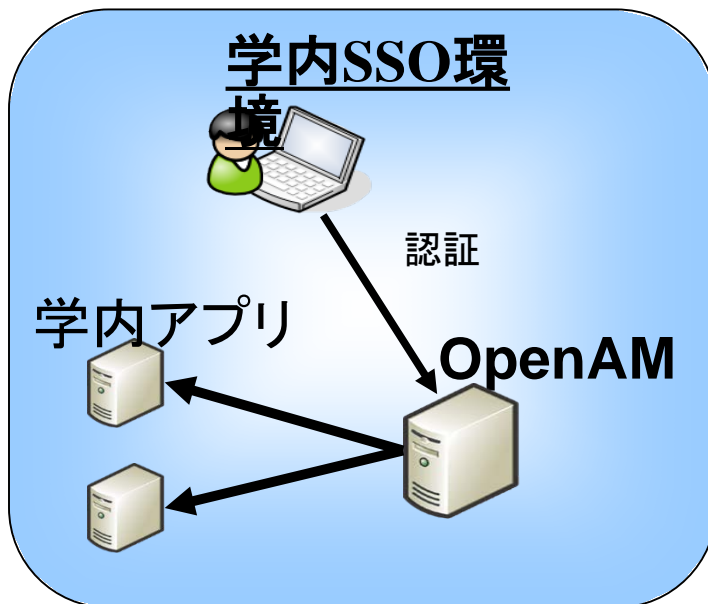
# 学内の認証をShibbolethで？ (2)

- 学内のアプリケーションをSAML化すればShibbolethでのシングルサインオン環境を実現できる
  - SAMLに対応していないアプリは**改修**が必要
- 他にもこんな懸念も・・・
  - 学認との接続ノウハウの情報はあるが、**学内アプリのSSO化の情報**が少ない
    - ◆ アプリケーションのSAML化ってどうすれば良いかわからない
    - ◆ アプリケーションの改修にコストがかかる
  - Shibbolethの標準機能では認証方式はID/Password形式
    - ◆ デスクトップSSO/クライアント証明書等柔軟な認証方式を採用できない
    - ◆ 複数の認証方式の組み合わせ認証連鎖を行えない
  - Shibbolethでは認証のみ提供。認可の提供はない。

Shibbolethは学内アプリケーションのシングルサインオンに不向き

# 学内のシングルサインオンはOpenAM

- 学内のシングルサインオンはOpenAMで行うのが良い
  - OpenAMはシングルサインオンの方式としてSAML以外にも用意
    - ◆ 様々な方式を用意し**アプリの改修なし**でシングルサインオンを実現可能
  - 多様な認証方式を用意
    - ◆ ID/パスワード認証以外に認証方式を標準で備えており、**システムのセキュリティや用途にあった認証方式の選択が可能**
- **学内はOpenAM、学認はShibboleth**という構成がベスト



# •OpenAMとShibbolethの連携

- OpenAMとShibbolethが別々では、それぞれで認証が必要
  - シングルサインオン製品を活かせていない！



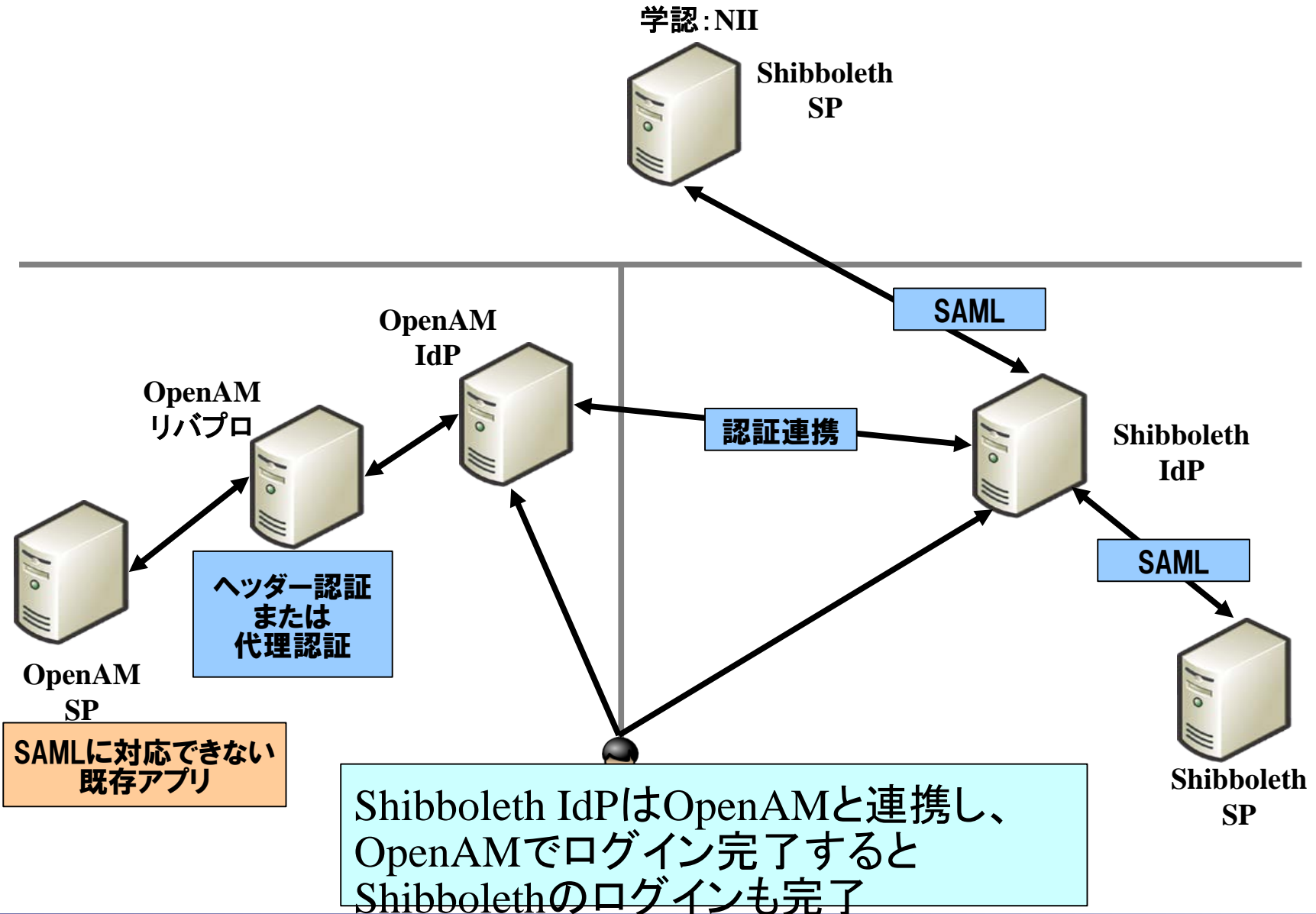
## • そこでOpenAMとShibbolethを連携する

- **Shibboleth IdPの認証をOpenAMで行う**
  - ◆ 具体的にはShibbolethがOpenAMにRest APIによる認証有無の問い合わせを行い、ユーザーはShibbolethでのログイン操作は行わなくて済む

ユーザーは一度の認証で、学認、学内のアプリケーションが使用可能



# OpenAMとShibbolethの構成





# OpenAMによるシングルサインオン システム導入事例



OSSTech

# 某通信会社グループ共通 シングルサインオンシステム

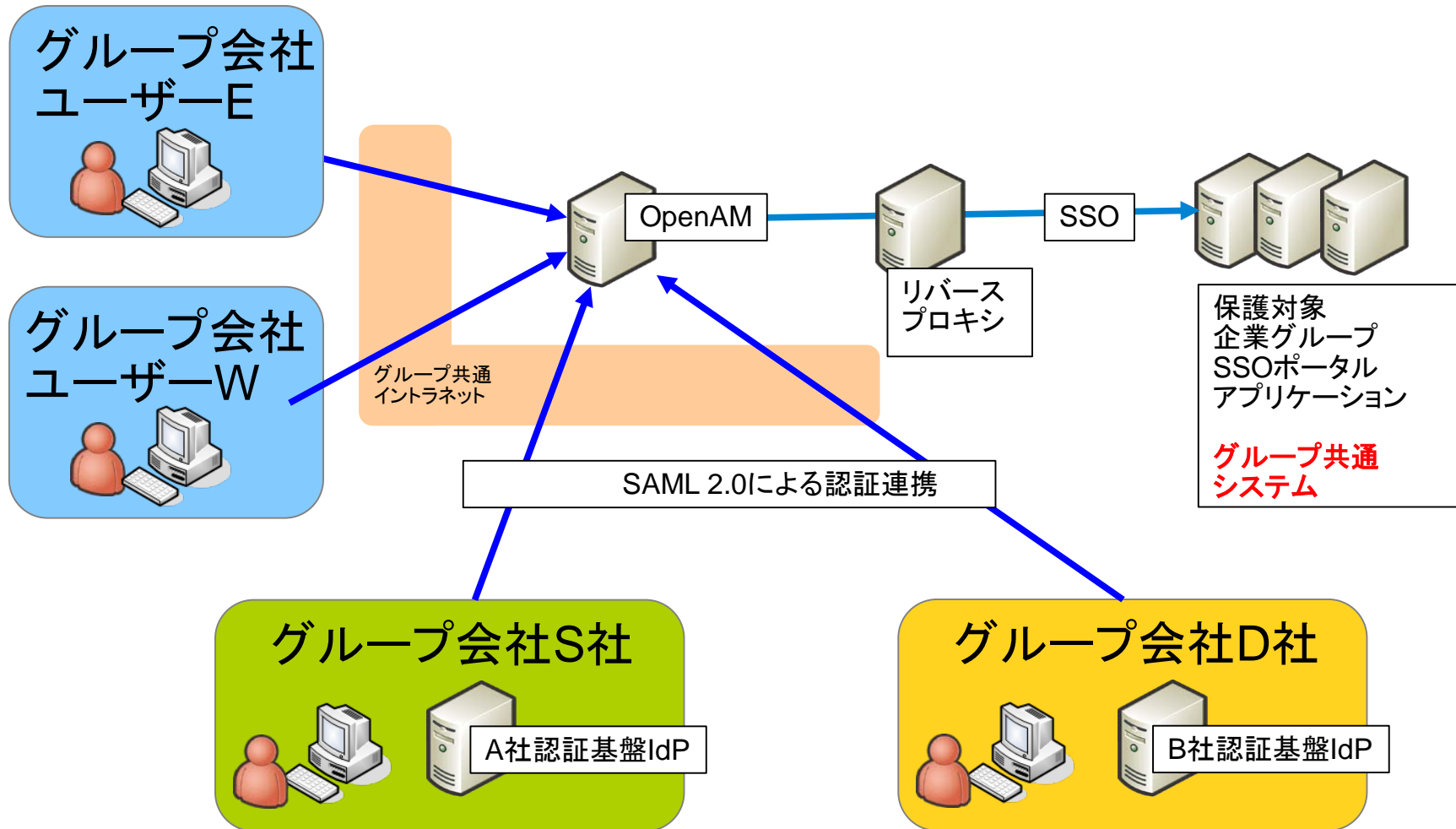


OSSTech

# 某通信会社グループ共通 シングルサインオンシステム

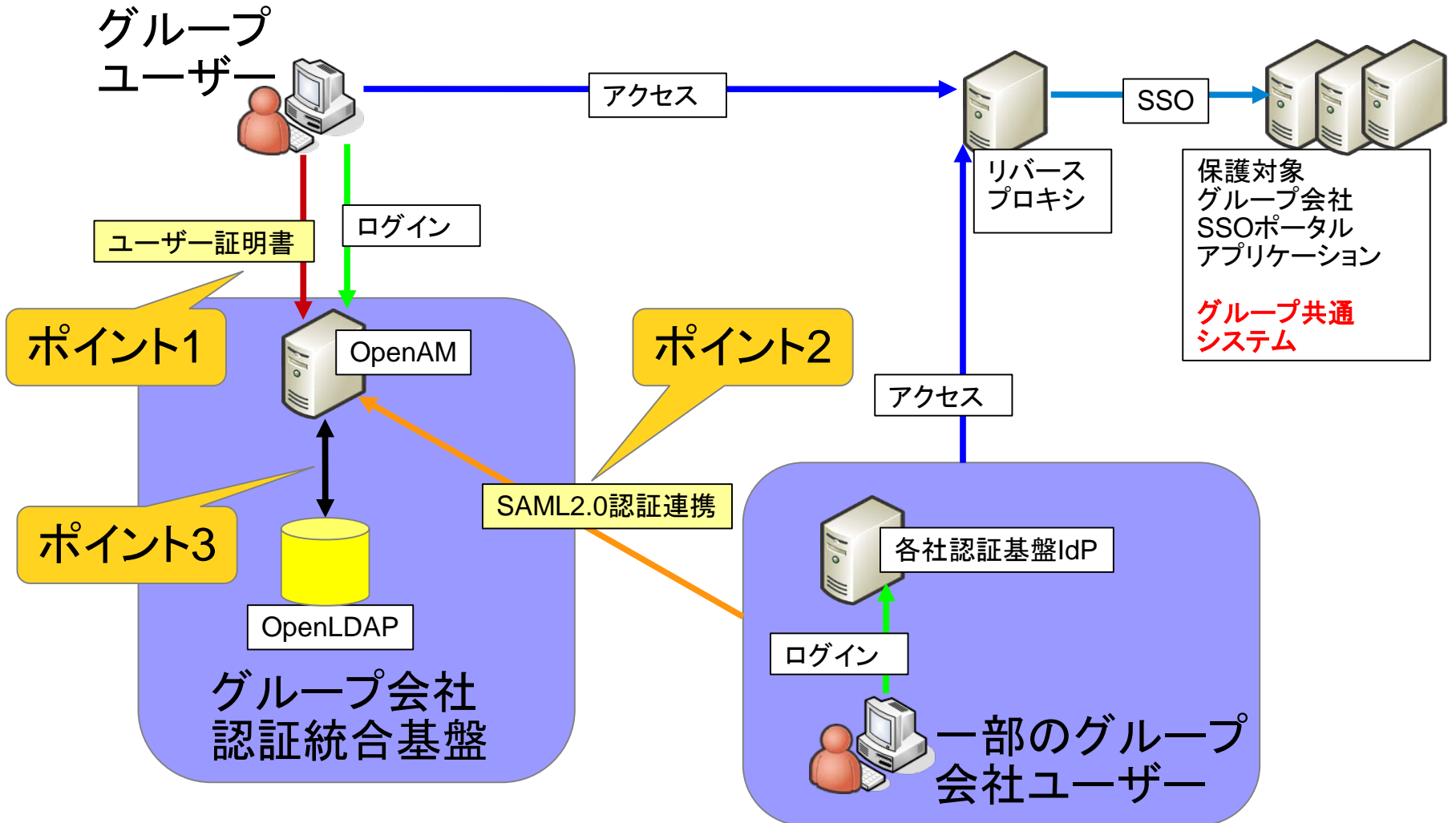
- ・ ユーザー総数 約25万人
- ・ ID/パスワードとユーザー証明書の多要素認証（認証連鎖）
- ・ 一部グループ会社ユーザーはSAML 2.0対応IdPによる認証連携
- ・ OpenLDAPのパスワードポリシー対応モジュールの開発
- ・ 保護対象アプリケーションとの連携はPolicyAgentを用いたリバースプロキシ型

# 某通信会社グループ 全体構成図



一部グループ会社では各社の認証基盤をIdPとしてOpenAMと連携

# 某通信会社グループ 構築のポイント



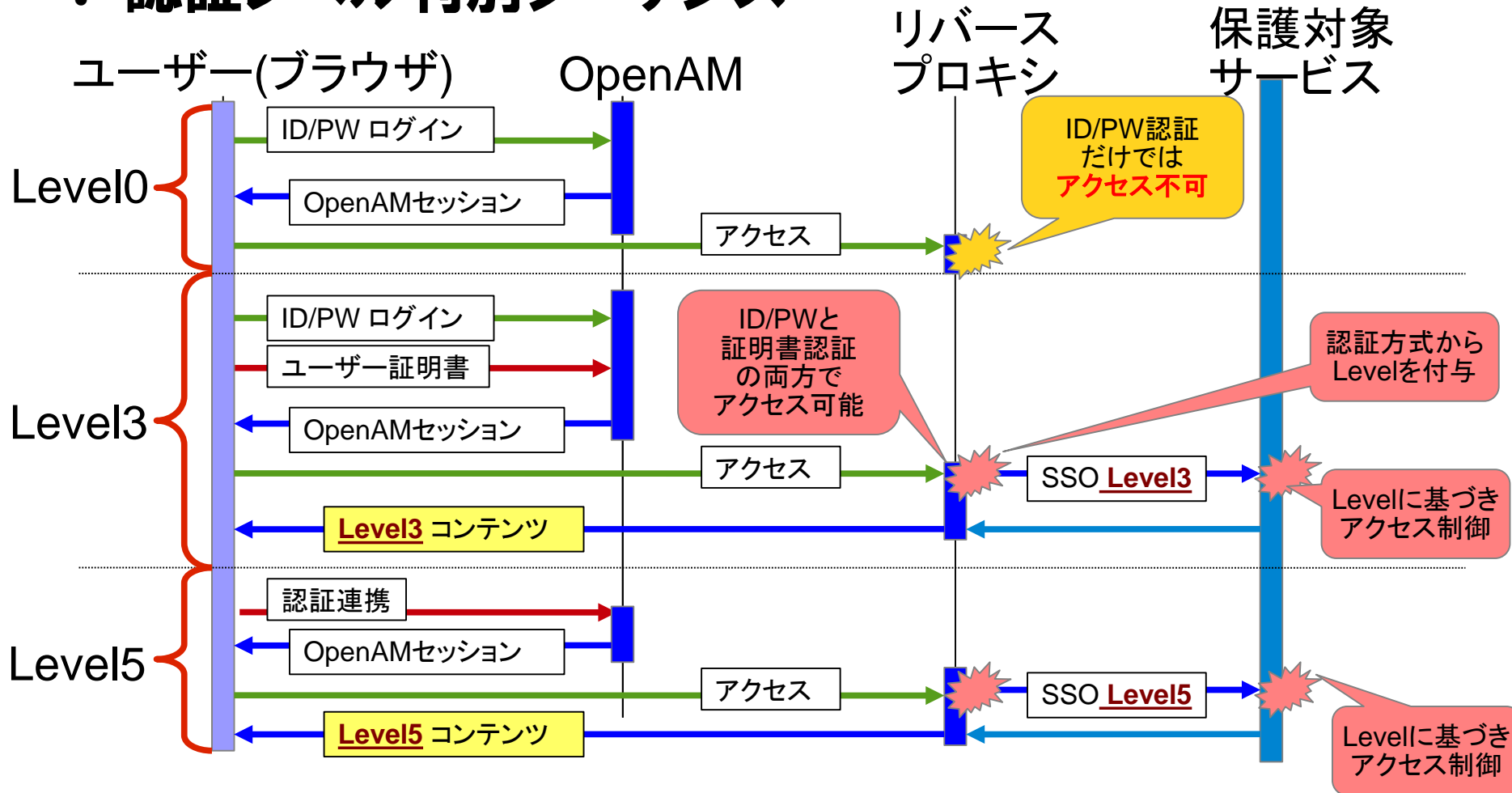
# 多要素認証

## ・ポイント1

- ID/パスワードとユーザー証明書を用いた多要素認証
- 「認証連携」での接続方法も、同等の認証レベルをセットするカスタム認証モジュールを開発
- OpenAMリバースプロキシのポリシーでレベルをチェックしアクセス制御

# 多要素認証時の認証・認可シーケンス

## ・ 認証レベル判別シーケンス



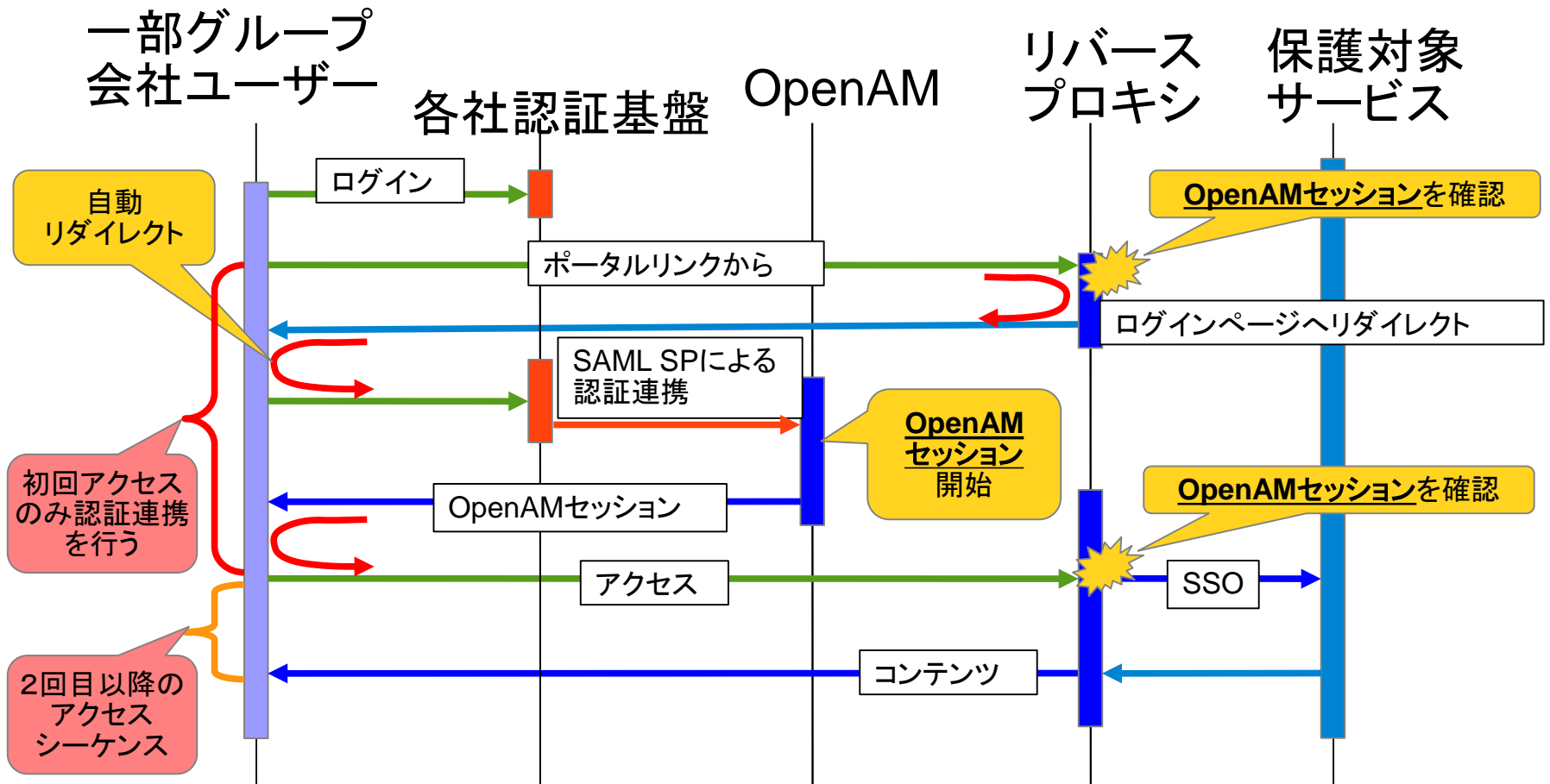
# 異なるIdP製品との認証連携

## ・ポイント2

- 一般的にユーザーはOpenAMで認証を行う。
- 一部のグループ会社ユーザーは各社認証基盤のIdPで認証を行い、OpenAM保護下のグループ会社SSOポータルアプリケーションとはSAML認証連携でアクセス可能とする。



# 異なるIdP製品との認証連携シーケンス



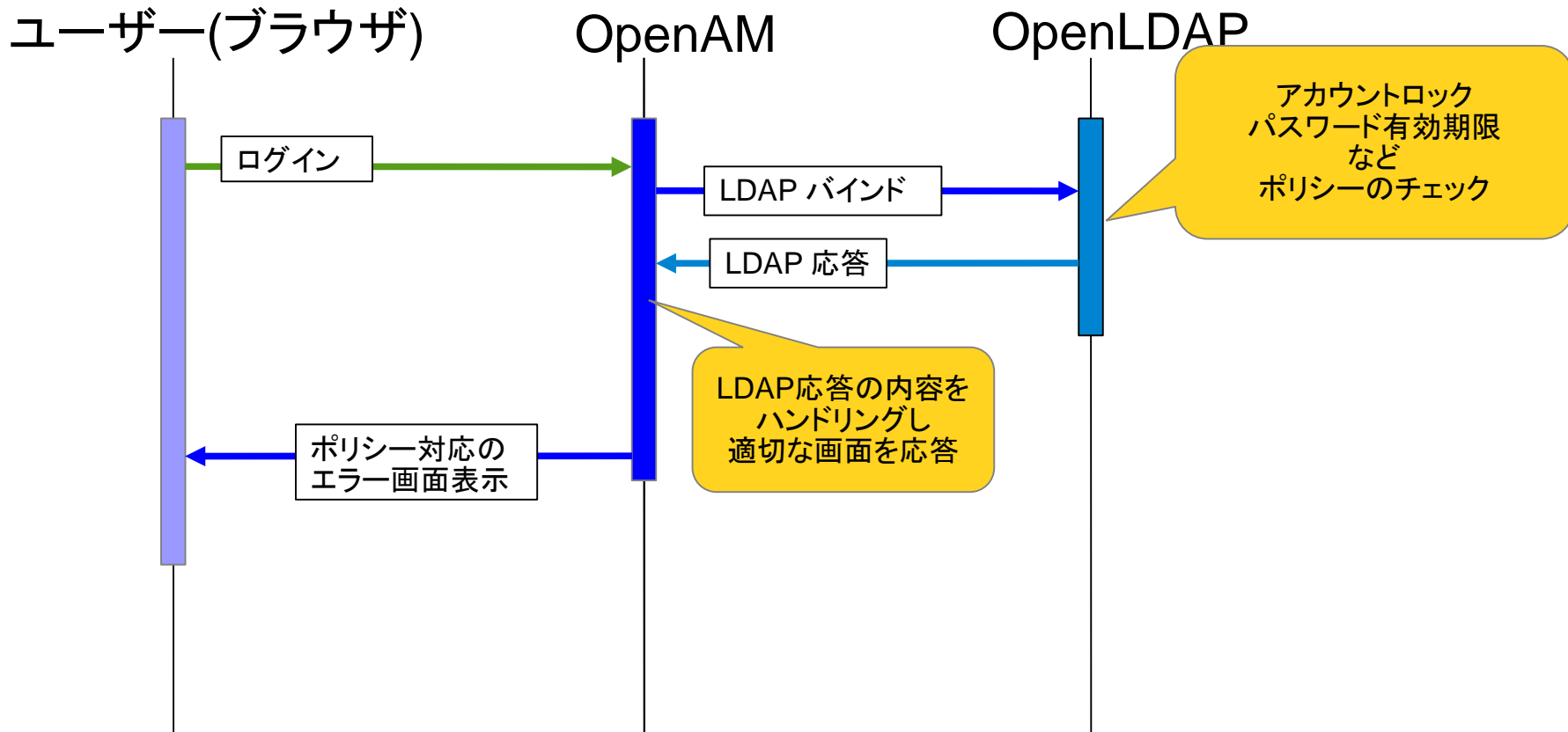
# OpenLDAPポリシーへの対応

## ・ポイント3

- OpenAM 9系では対応していないOpenLDAP (RFC標準) のアカウントポリシーエラー対応のためOpenAMの拡張開発を行った。
- 拡張を行ったOpenAMは、パスワード有効期限切れなどOpenLDAPからの戻り値を判定し、任意のURLへ遷移する。

# OpenLDAPポリシーへの対応

## ・ OpenLDAPエラー情報判定シーケンス



# 某総合電機メーカー シングルサインオン システム

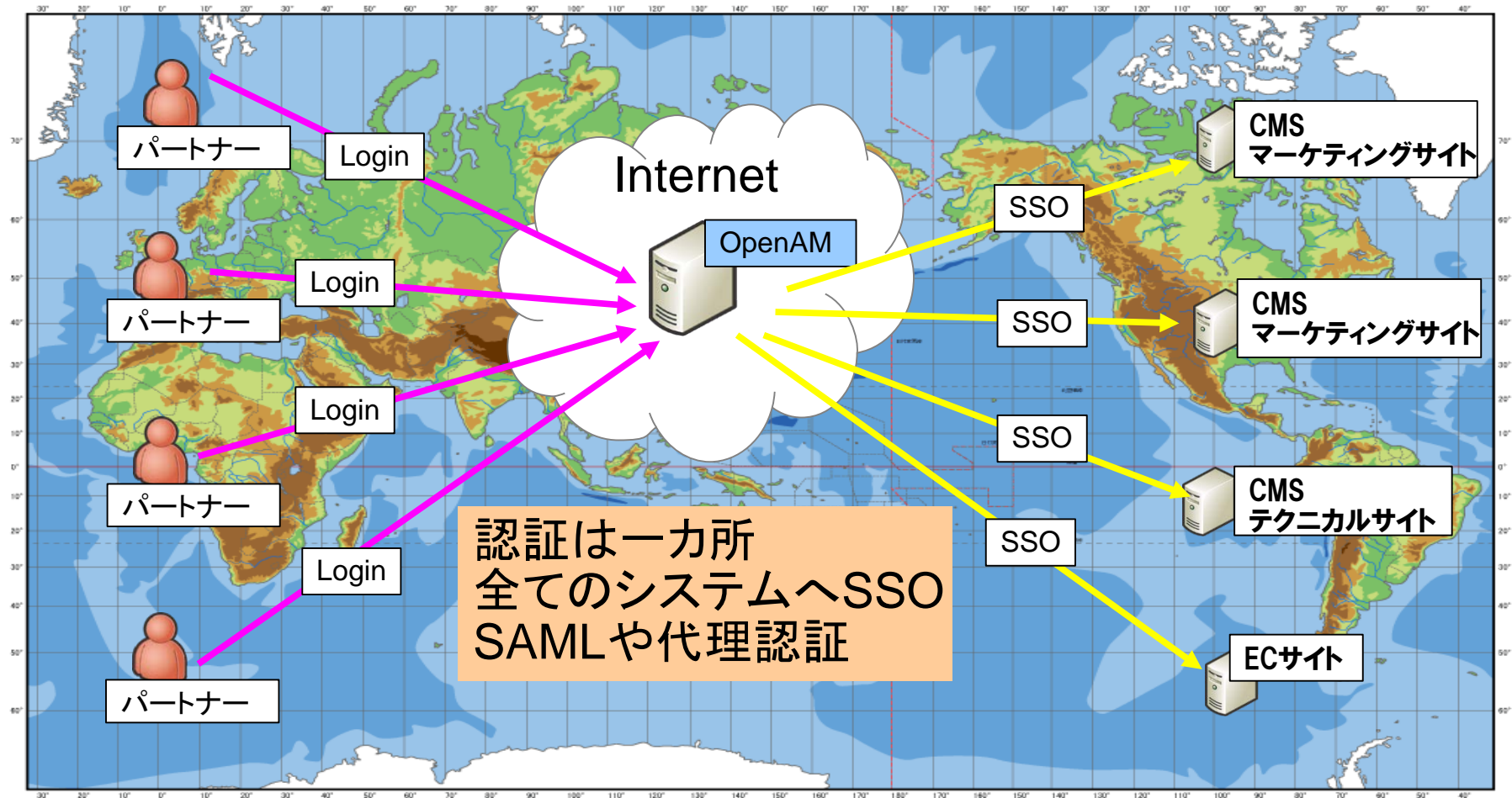


OSSTech

# 某総合電機メーカー シングルサインオンシステム

- 規模:グループ企業7社、約5000人、海外22拠点  
今後拡大予定
- 海外ディーラー向けの技術情報やマーケティング情報のCMSおよびECサイトへのシングルサインオン
- CMS, ECサイトとの連携はOpenAM PolicyAgentとお  
客様開発の連携モジュール
- SAML認証と代理認証を利用
- 対象ユーザー、保護対象アプリケーションはインター  
ネット上に点在

# 某総合電機メーカー 構成図



# 国立大学法人 名古屋工業大学



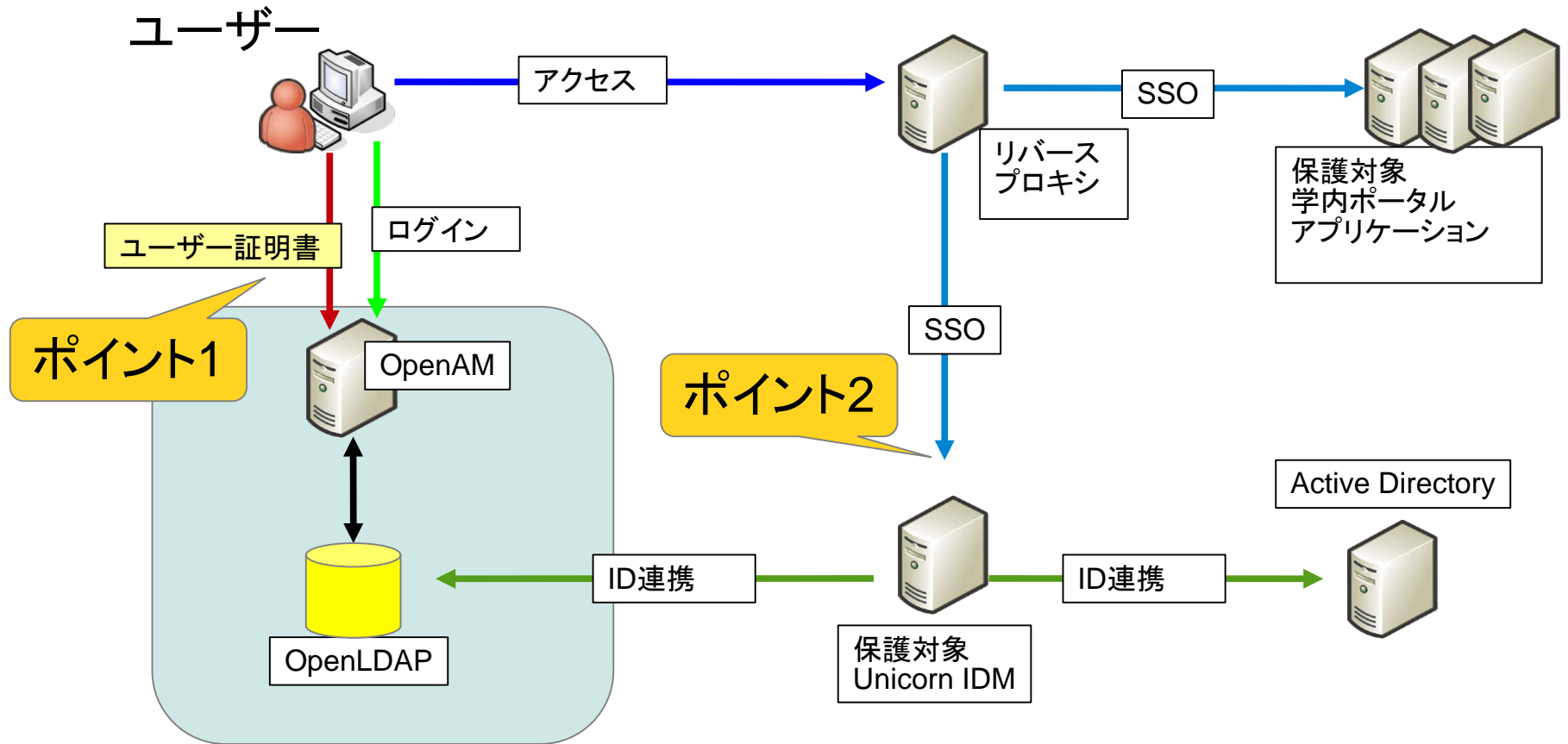
OSSTech

# 名古屋工業大学様 事例のポイント

- 規模 学生数 約5,800人 教職員数 約510人
- 旧Sun製品の置き換え
  - 旧Sun製品(Sun Java System Access Manager)からの移行を実現
  - 旧Sun製品のOracle後継製品を導入する場合はコスト高
  - Sun Java System Access Managerの後継であり、OSSのOpenAMを採用
  - 他にもLDAPにOpenLDAP, ID管理にUnicorn IDMと積極的にOSSを採用
- ICカードによる認証とID/パスワードによる認証の使い分け
  - アクセスリソースに対しての認証レベルの使い分け
  - 「ICカードによる証明書認証」と「ID/パスワードによる認証」の二つの認証方式を用意
  - 重要なリソースへのアクセスの際にはより安全なICカードで認証したユーザーのみをアクセス可能とした
- **日立製作所**と**オープンソース・ソリューション・テクノロジー**で実現



# 名古屋工業大学 構成図

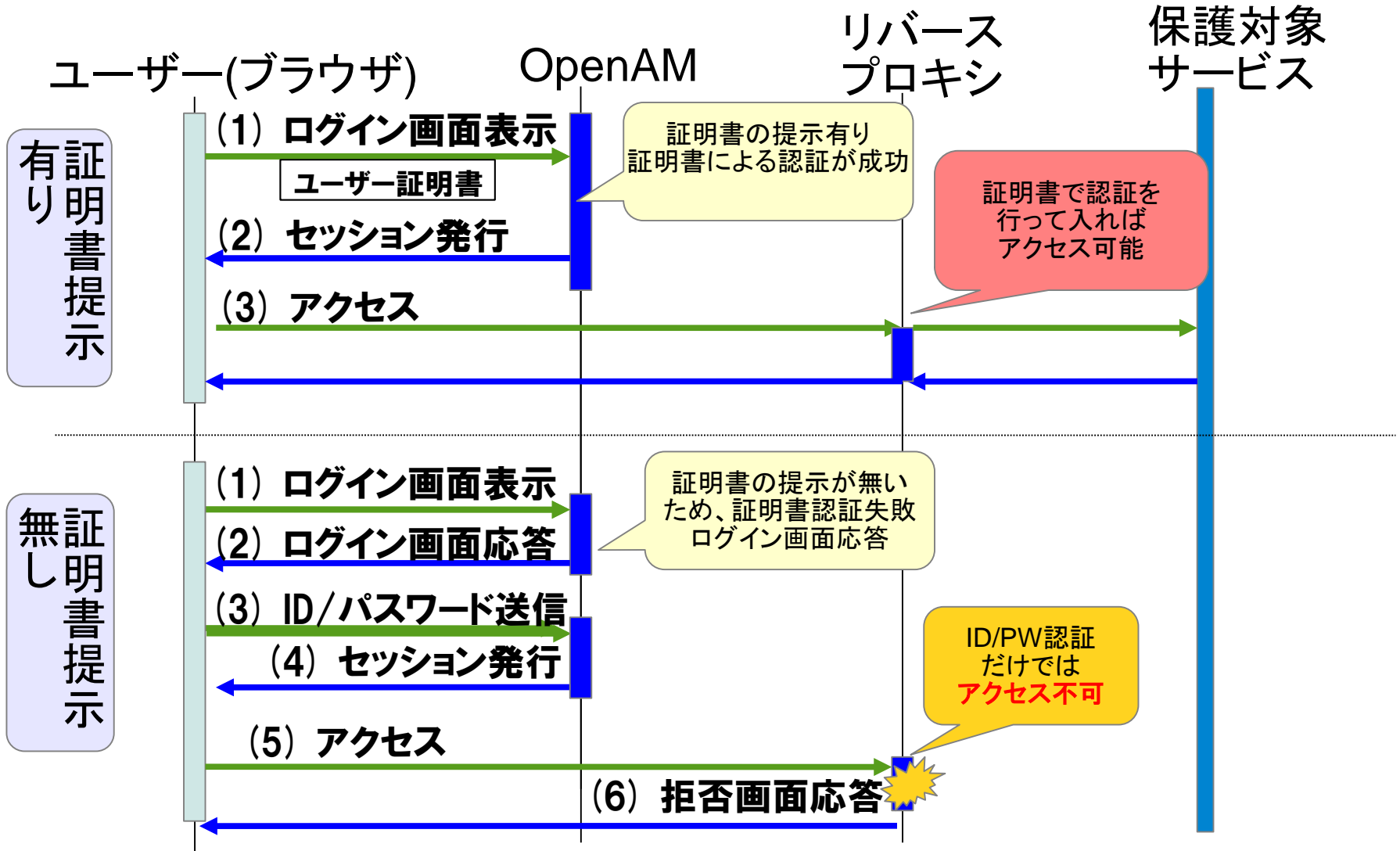


# 名古屋工業大学 認証の使い分け

## ・ポイント1

- ICカードを使った証明書認証を基本とする
- 証明書認証に失敗した場合(証明書の提示が無い)にログイン画面を表示しID/パスワードを用いた認証
- 証明書認証とID/パスワード認証では異なる認証レベルをセット
- OpenAMリバースプロキシのポリシーでレベルをチェックしアクセス制御

# 名古屋工業大学 認証シーケンス

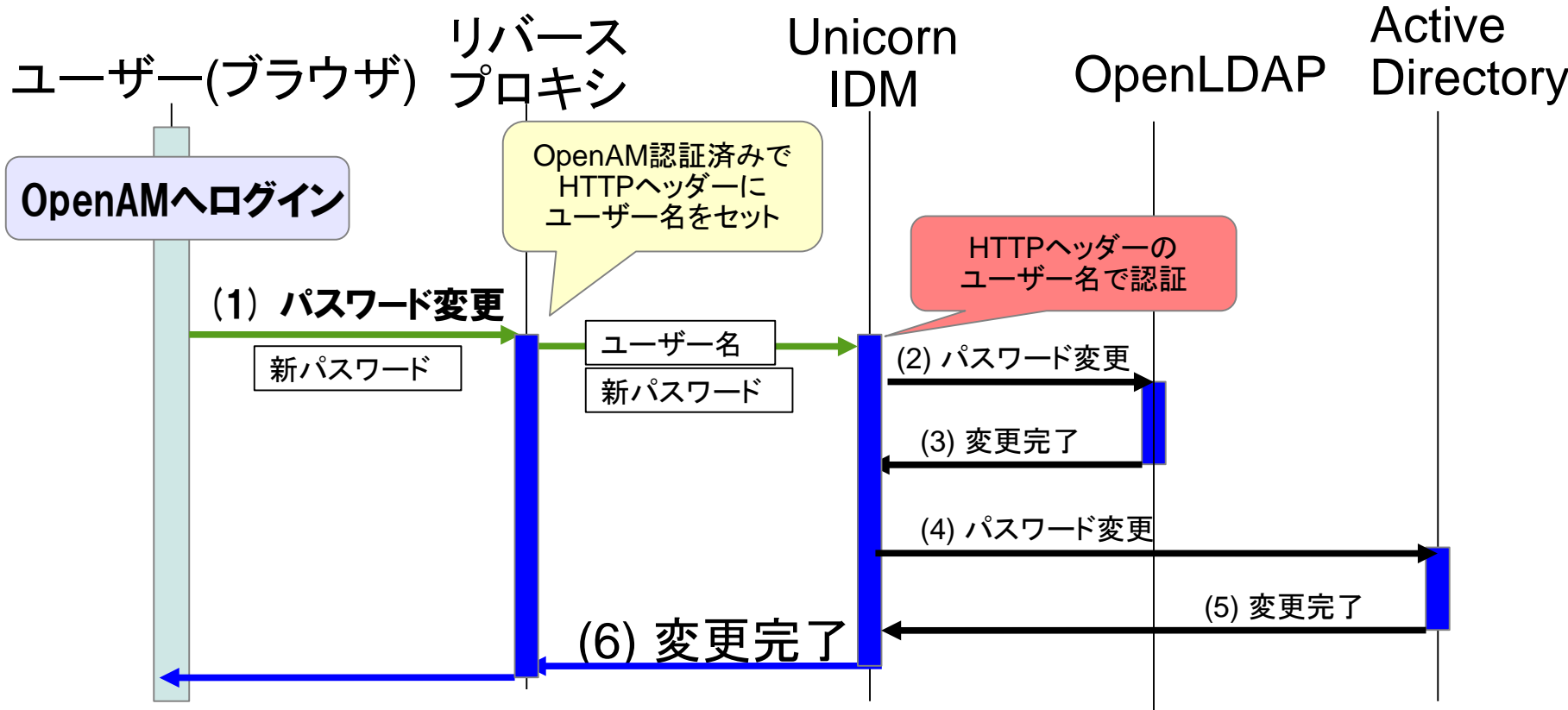


# 名古屋工業大学 ID管理

## ・ポイント2

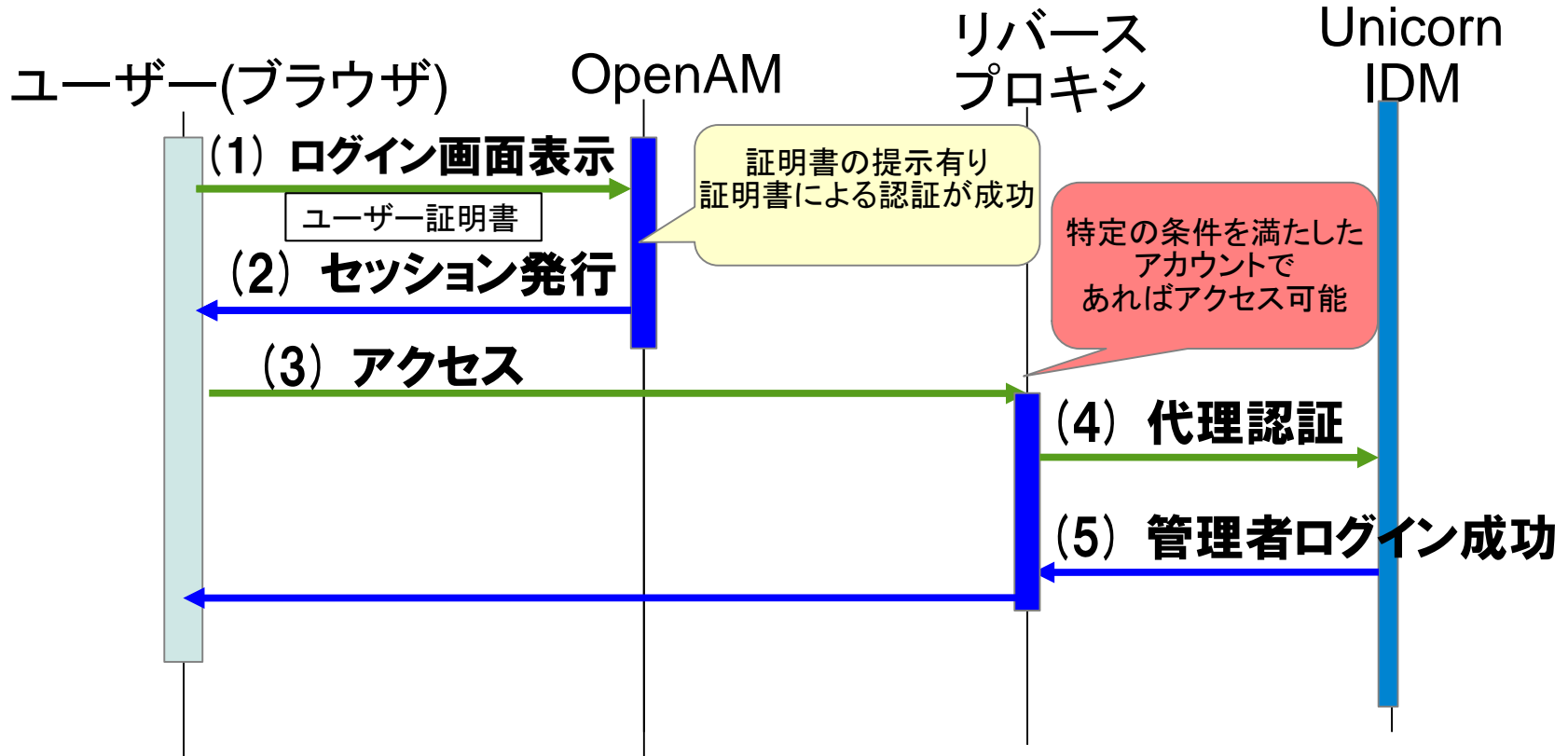
- Unicorn IDMによるID連携を実施
  - Active Directory と OpenLDAPのアカウントを同期
- OpenAMとのシングルサインオンを実現
  - ユーザーはOpenAMにログイン済みであれば、再度の認証無しでパスワードの変更が可能
  - UnicornIDMの管理者アカウントもシングルサインオンを実現

# 名古屋工業大学 パスワード変更



- ユーザーはOpenAMログイン済みなので新パスワードのみでパスワード変更可能
- Unicorn IDMによりOpenLDAPとActive Directoryのパスワードが同時変更

# 名古屋工業大学 管理者シーケンス



特定の条件を満たしたアカウントはOpenAMにログインすることで、Unicorn IDMの管理者としてログインすることができる。

**大学法人 福岡大学 様**

# 福岡大学様 システムの特徴

## 規模

9つの学部、2つの病院、22の付置施設で構成される総合大学  
学生数 約21,000人  
教職員数 約3,000人

## ミッション

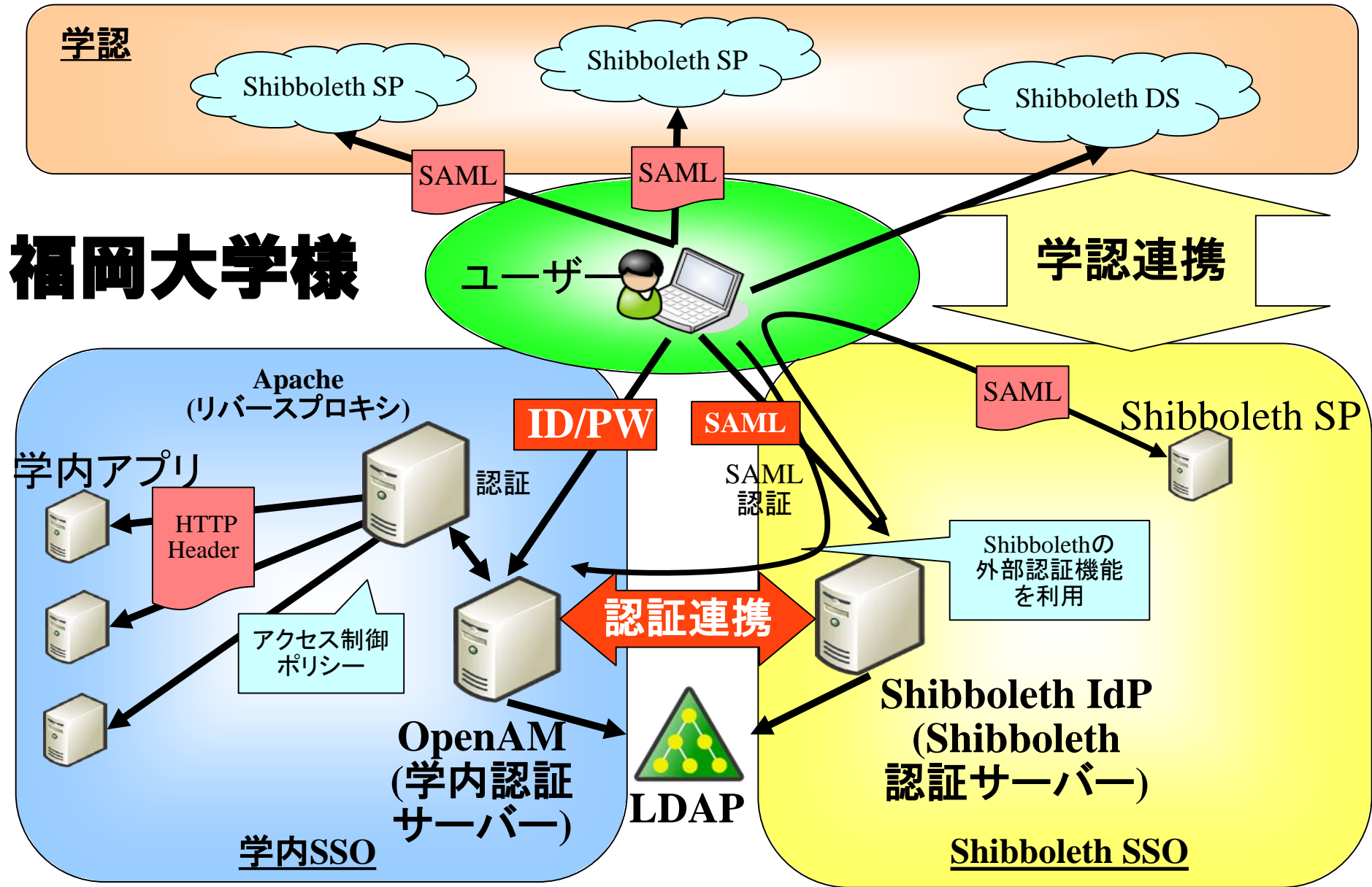
高い拡張性と柔軟性を持つ先進的SSO基盤の構築

**日立製作所**と**オープンソース・ソリューション・テクノロジー**で実現

### OpenAMとShibbolethによるハイブリッド型SSO基盤

- ・ システムのシングルサインオンを実現する認証基盤をOpenAMとShibbolethを使って実現
- ・ 様々なアプリケーションとのシングルサインオンを実現する基盤
- ・ ユーザーは1度の認証で学認と学内のアプリケーションを利用可能



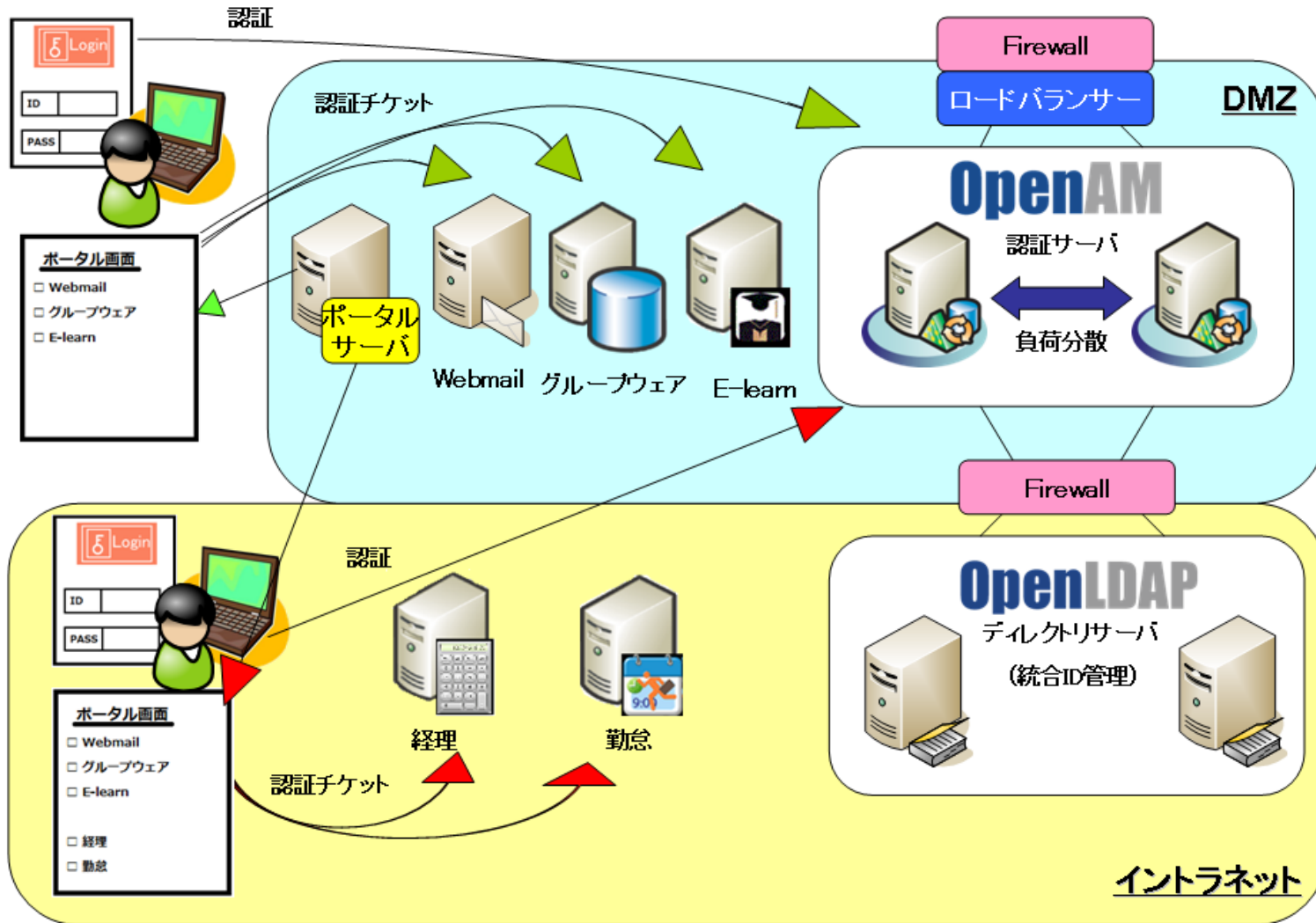


**国立大学法人 北見工業大学 様**

# 北見工業大学様 システムの特徴

- ユーザー(学生や教職員)はOpenAMに一度ログインすると、複数のWebアプリケーションをログイン操作なしで利用できます。
- ログインするとポータルメニューが表示されますが、ユーザー権限やログイン場所(学内/学外)によって表示されるメニューが変化します。
- ログインしたユーザーが利用できないアプリケーションは表示されず、インターネットからログインするとイントラネット専用アプリケーションも表示されません。
  - システム全体設計やプロジェクトとりまとめは、兼松エレクトロニクス株式会社が行いました。
  - シングルサインオン システム構築は、オープンソース・ソリューション・テクノロジ株式会社が行いました。

# 北見工業大学様



# OpenAM 10

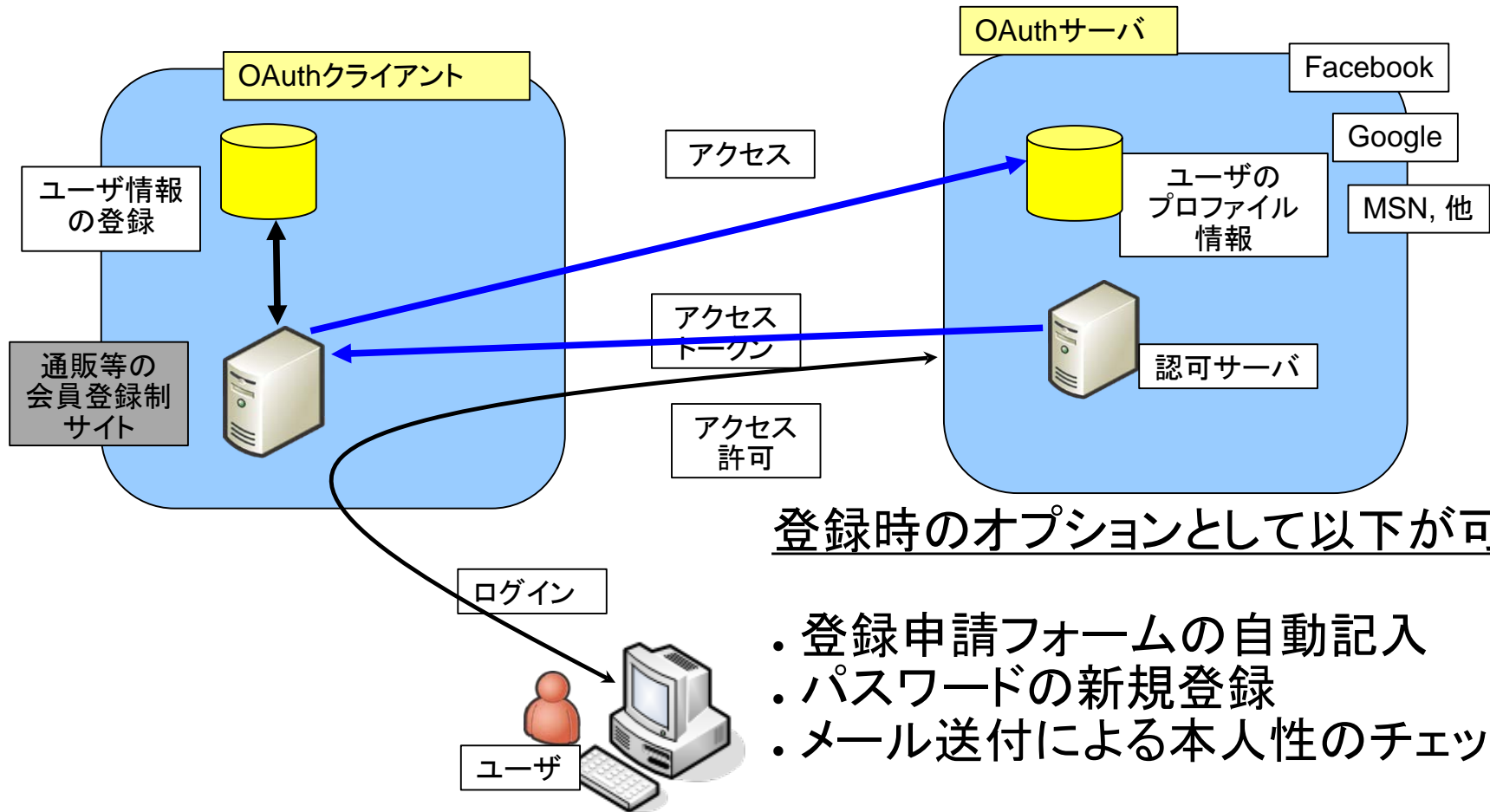
## OAuth 2.0の機能



OSSTech

**Facebook認証のアプリを簡単開発！**

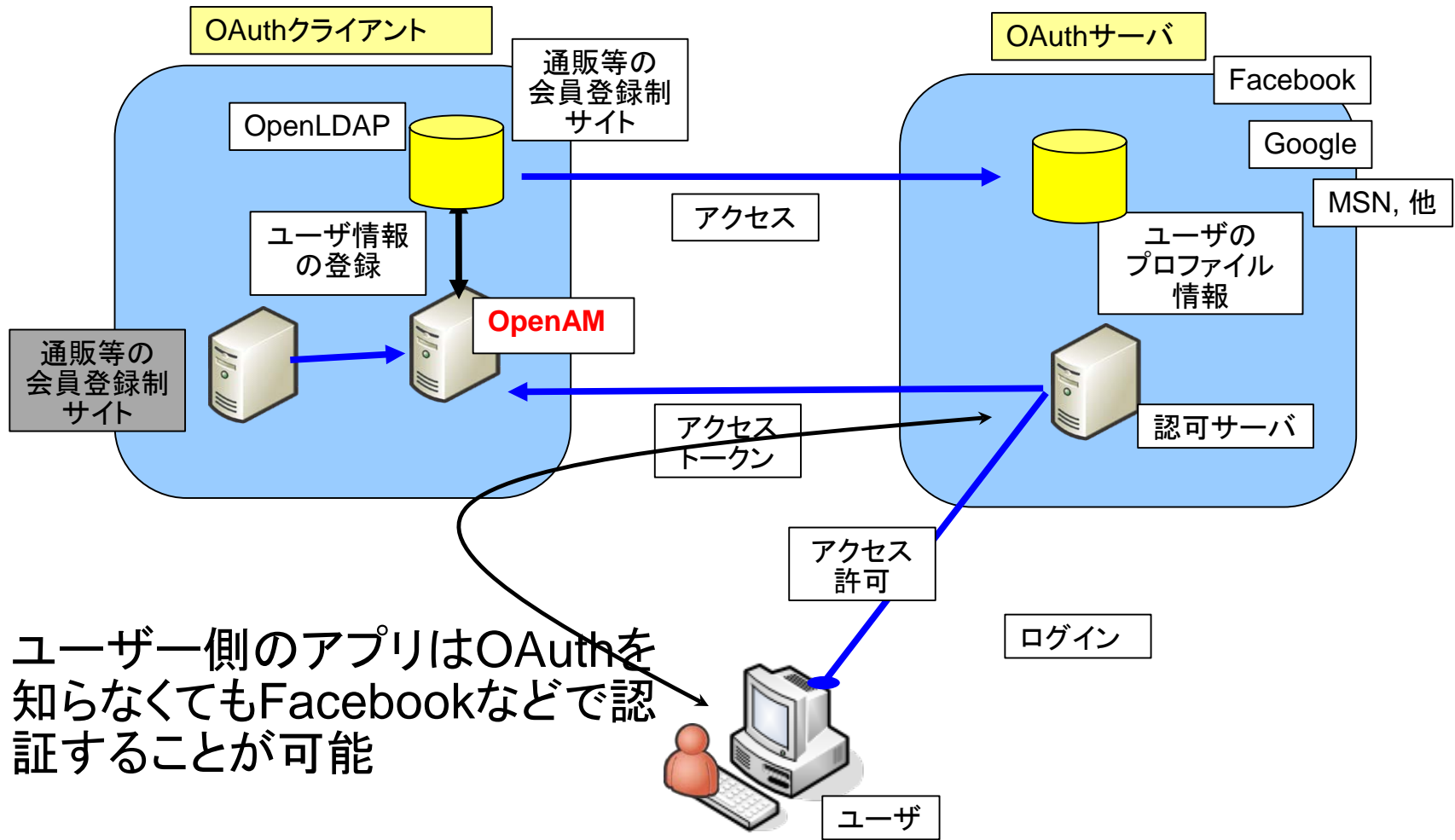
# 自社のサービスにFacebookなどのアカウントでログインしてもらうOAuth クライアント機能



登録時のオプションとして以下が可能

- ・登録申請フォームの自動記入
- ・パスワードの新規登録
- ・メール送付による本人性のチェック

# OAuth クライアントとしてOpenAMを使う



# OAuth 2.0 のクライアントとして使う設定

管理コンソールで

簡単設定

詳細手順は弊社ホームページで紹介中

## OAuth 2.0

保存 リセット 認証へ戻る

### レールム属性

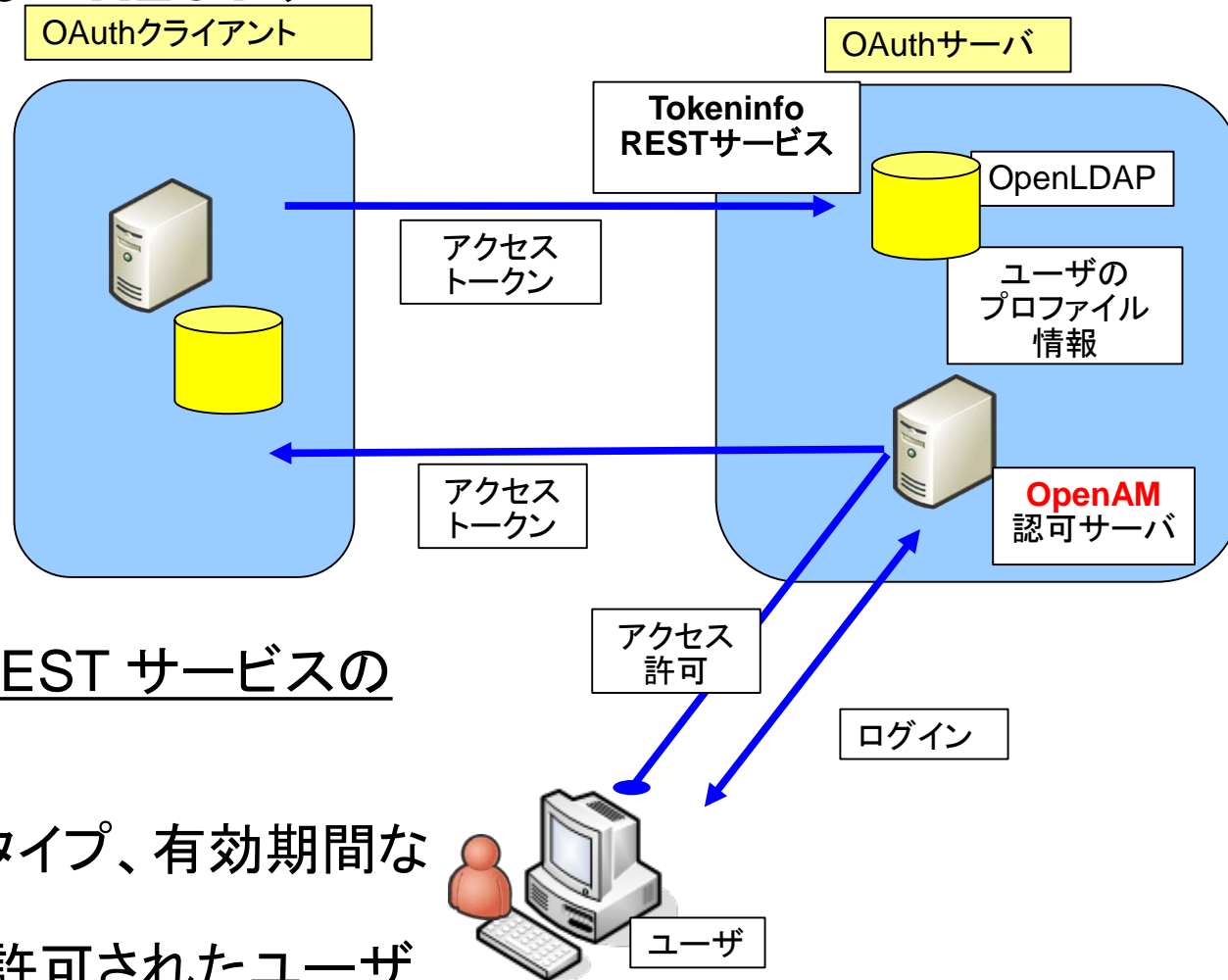
クライアント ID:	<input type="text"/>	<small>i OAuth client_id パラメーター。</small>
クライアントシークレット:	<input type="text"/>	<small>i OAuth client_secret パラメーター。</small>
クライアントシークレット (確認):	<input type="text"/>	
認証エンドポイント URL:	<input type="text" value="https://www.facebook.com/dialog/oauth"/>	<small>i OAuth 認証エンドポイント URL。</small>
アクセストークンエンドポイント URL:	<input type="text" value="https://graph.facebook.com/oauth/access"/>	<small>i OAuth アクセストークンエンドポイント URL。</small>
ユーザープロフィールサービス URL:	<input type="text" value="https://graph.facebook.com/me"/>	<small>i ユーザープロフィール情報 URL。</small>
スコープ:	<input type="text" value="email,read_stream"/>	<small>i OAuth スコープ; ユーザープロフィールプロパティのリスト</small>
プロキシURL:	<input type="text" value="https://openam.servername/openam/oa"/>	<small>i OpenAM OAuth プロキシ JSP への URL。</small>
アカウントマッパー:	<input type="text" value="org.forgerock.openam.authentication.modu"/>	<small>i アカウントマッピングを実装するクラスの名前。</small>

### アカウントマッパー設定

現在の値



# 認可サーバとしての OpenAMと “tokeninfo” RESTサービス



## Tokeninfo REST サービスの 応答内容

- ・トークンのタイプ、有効期間など
- ・スコープで許可されたユーザ情報

# OpenAMのREST API

- ・ RFCで規定されたもの
  - ・ access\_token エンドポイント
  - ・ authorize エンドポイント
- ・ OpenAM独自
  - ・ tokeninfo エンドポイント
    - ・ トークンの検証
    - ・ 属性の設定
  - ・ トークン管理用エンドポイント

# OAuth 2.0の認可サーバとして使う 設定

管理者用GUIを  
使って簡単に設  
定可能

## Configure OAuth2

作成 取消し

Use this page to configure OAuth2

\* 必須入力フィールド

\* レalm:

## Configure OAuth2 Service

\* Refresh Token Lifetime (seconds):   
The time in seconds a refresh token is valid for

\* Authorization Code Lifetime (seconds):   
The time in seconds a authorization code is valid for

\* Access Token Lifetime (seconds):   
The time in seconds an access token is valid for

\* Issue Refresh Tokens:   
Check to enable generation of refresh tokens

\* Scope Implementation Class:   
The class that contains the required scope implementation

スコープを  
カスタマイズ  
したクラス

## Configure OAuth2 Authorization End Point Protection Policy

A policy to protect the OAuth2 authorization end point will be created. This policy will be named OAuth2ProviderPolicy. The policy will protect the endpoint `http://openam.server.name.com/openam/oauth2/authorize`. The purpose of this policy is to redirect clients to the OpenAM login page to authenticate a resource owner each time they go to the authorize end point. To do advanced policy management can be done using the policies tab for each realm.

## Register OAuth2 Client(s)

The last step is to register client(s) for the OAuth2 Provider to issue tokens to. Clients can be registered by navigating to the OpenAM agents tab and selecting OAuth 2.0 Client. A Client can also be registered by visiting the a jsp registration page at `http://openam.server.name.com/openam/oauth2/registerClient.jsp`

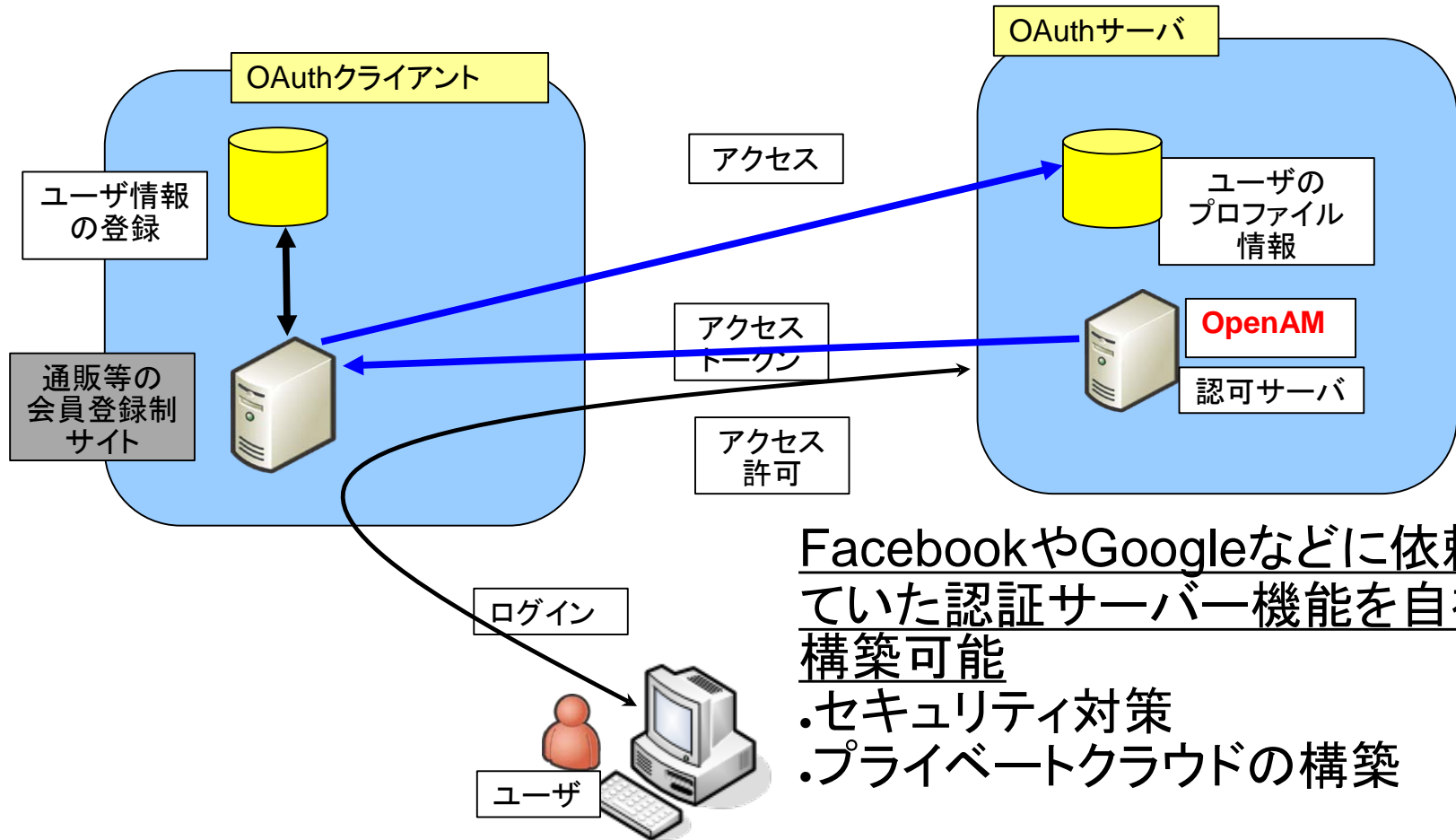
# OpenID Connect は対応中！



OSSTech

- ・ OAuth 2.0 の土台の上にOpenIDを構築
- ・ OpenAM 10.2 (2013, Q2) で対応予定
- ・ Facebook相当のOpenIDの認証サーバーを  
OpenAMを使って自社(オンプレミスやプライベートクラウド)で構築可能になる

# Oauthサーバ機能： Facebookなどの認証サーバーを独自に構築可能



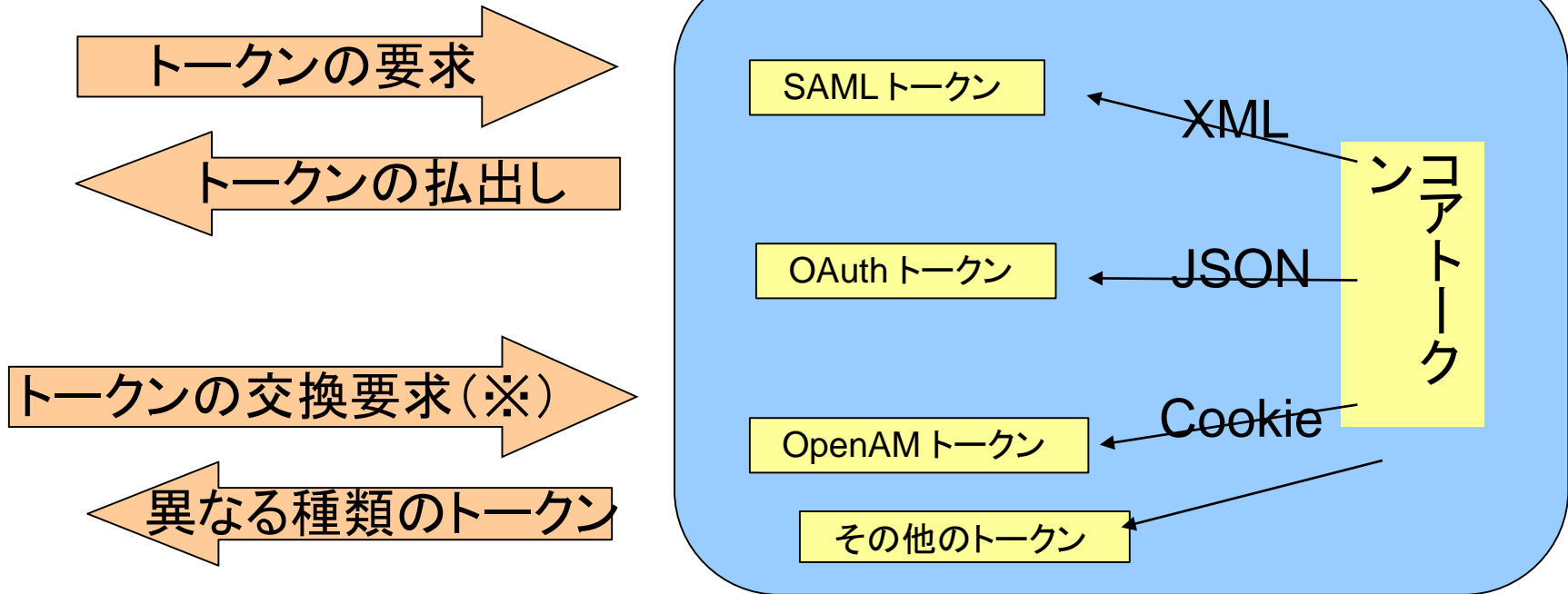
FacebookやGoogleなどに依頼し  
ていた認証サーバー機能を自社で  
構築可能

- ・セキュリティ対策
- ・プライベートクラウドの構築

# コアトークン・サービス

(様々な認証トークンに柔軟に対応)

コアトークンサービスの概念図



(※) token exchange はOpenAM 10.9で予定されている機能で現在は使用できません。

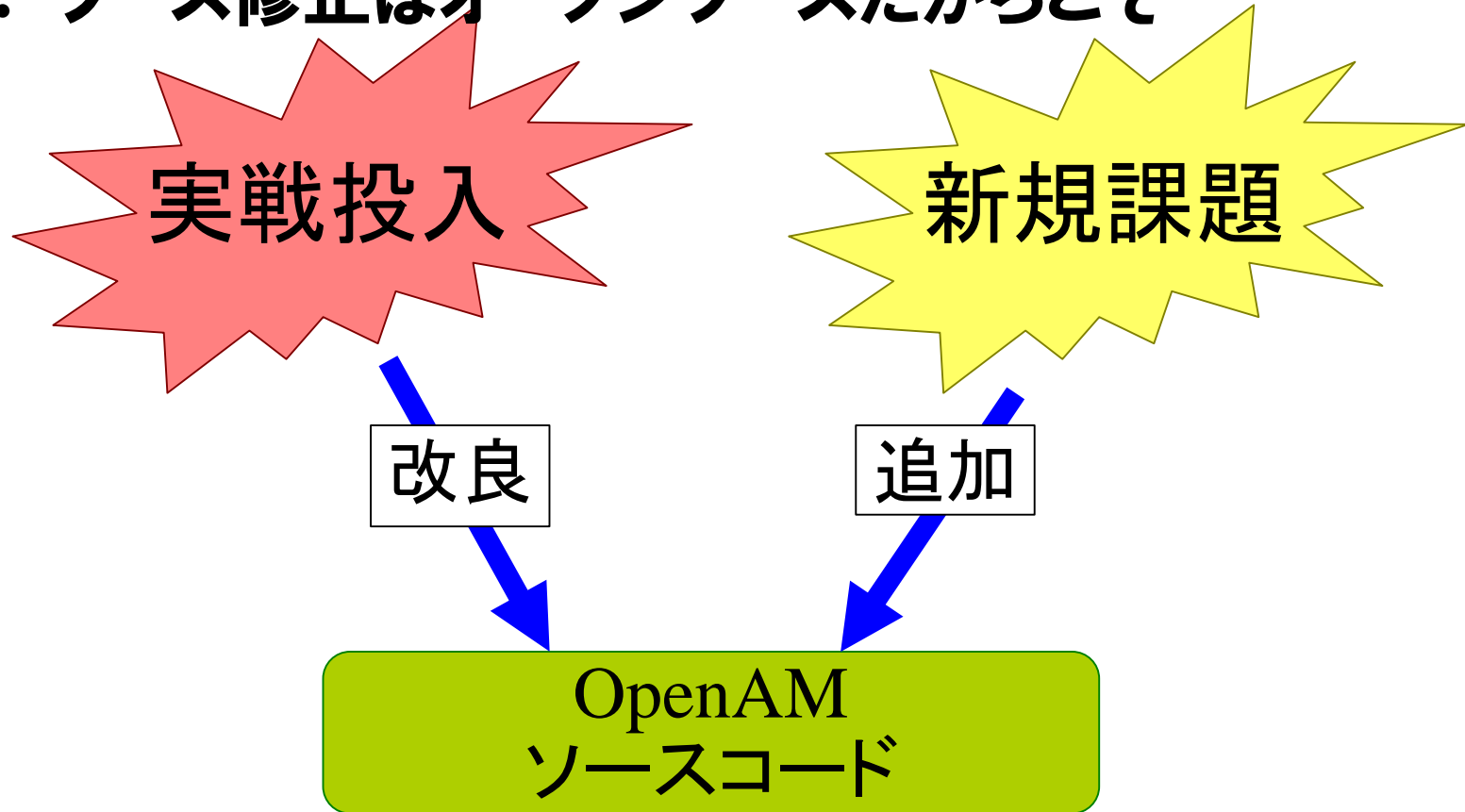
# OSSTechのOpenAMへの取り組み



OSSTech

# オープンソースであること

- ・ ソース修正はオープンソースだからこそ

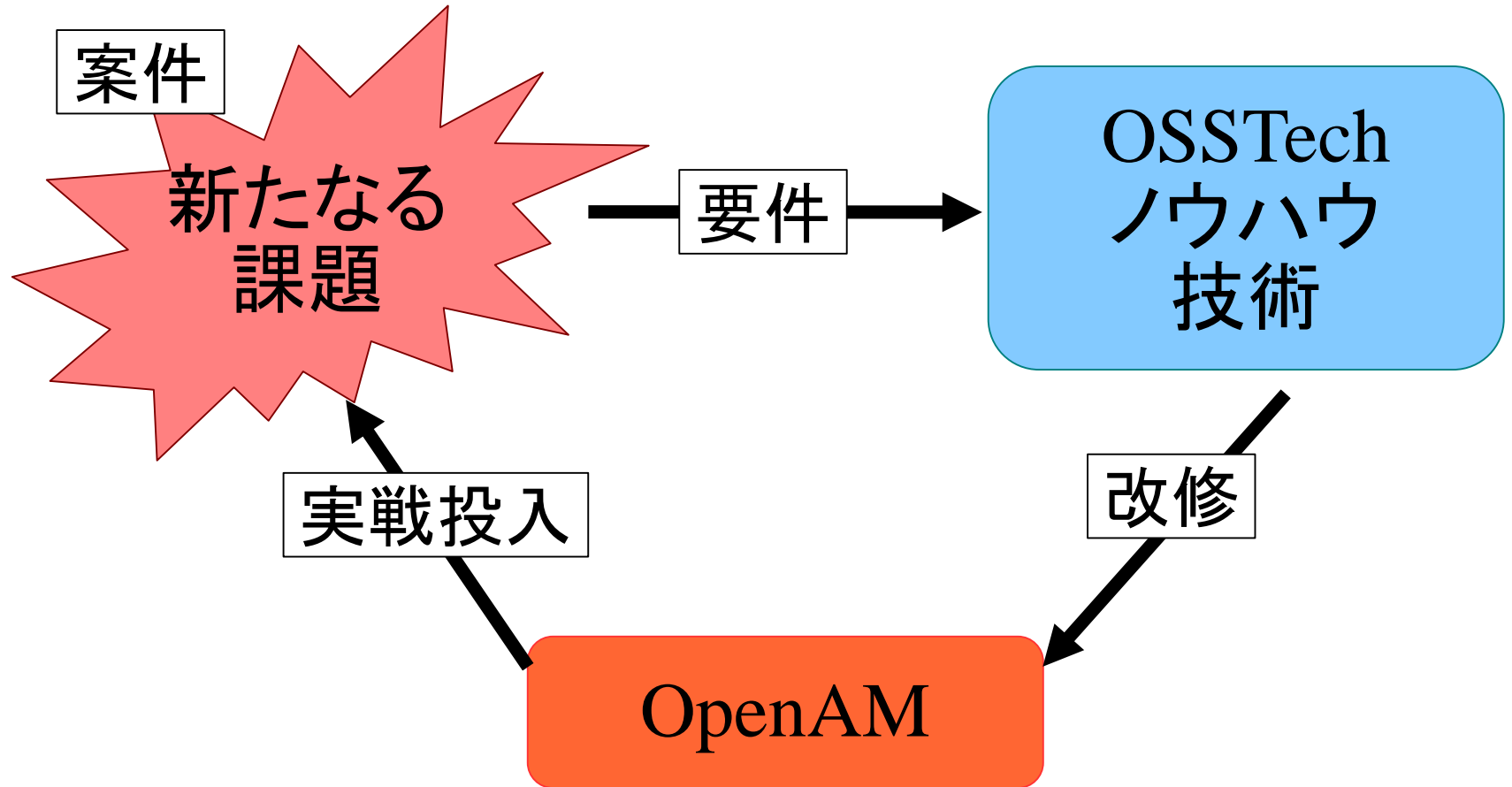


問題点の改良、要件に合わせた機能追加



# オープンソースであること

- ・ ソース改修力は課題解決力



全てのサイクルをサポートできる

# OpenAM ソースコード への貢献

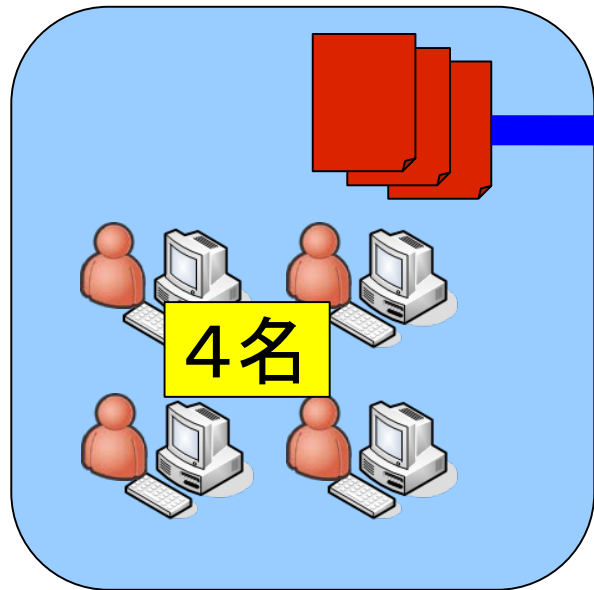


OSSTech

# コミッター数

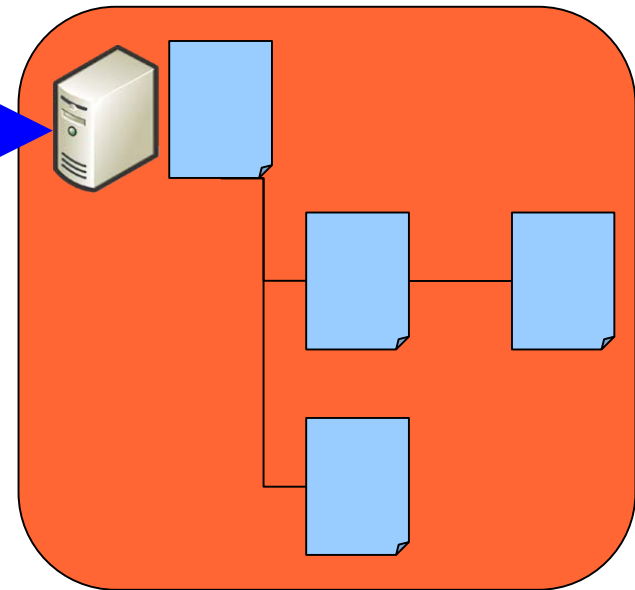
- ・ 社内開発者6名
- ・ コミット権限のある社内開発者4名
- ・ OpenAM コミッター総数37名

OSSTech社



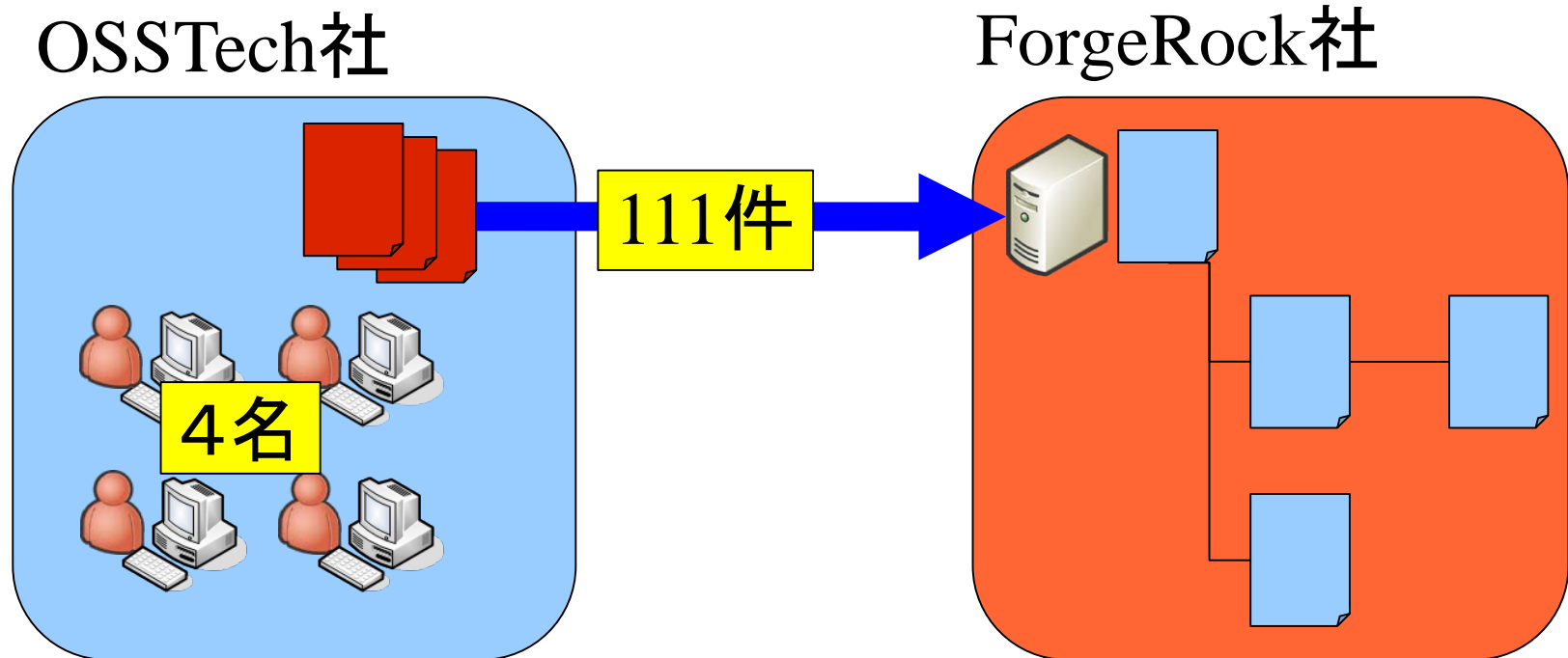
commit

ForgeRock社



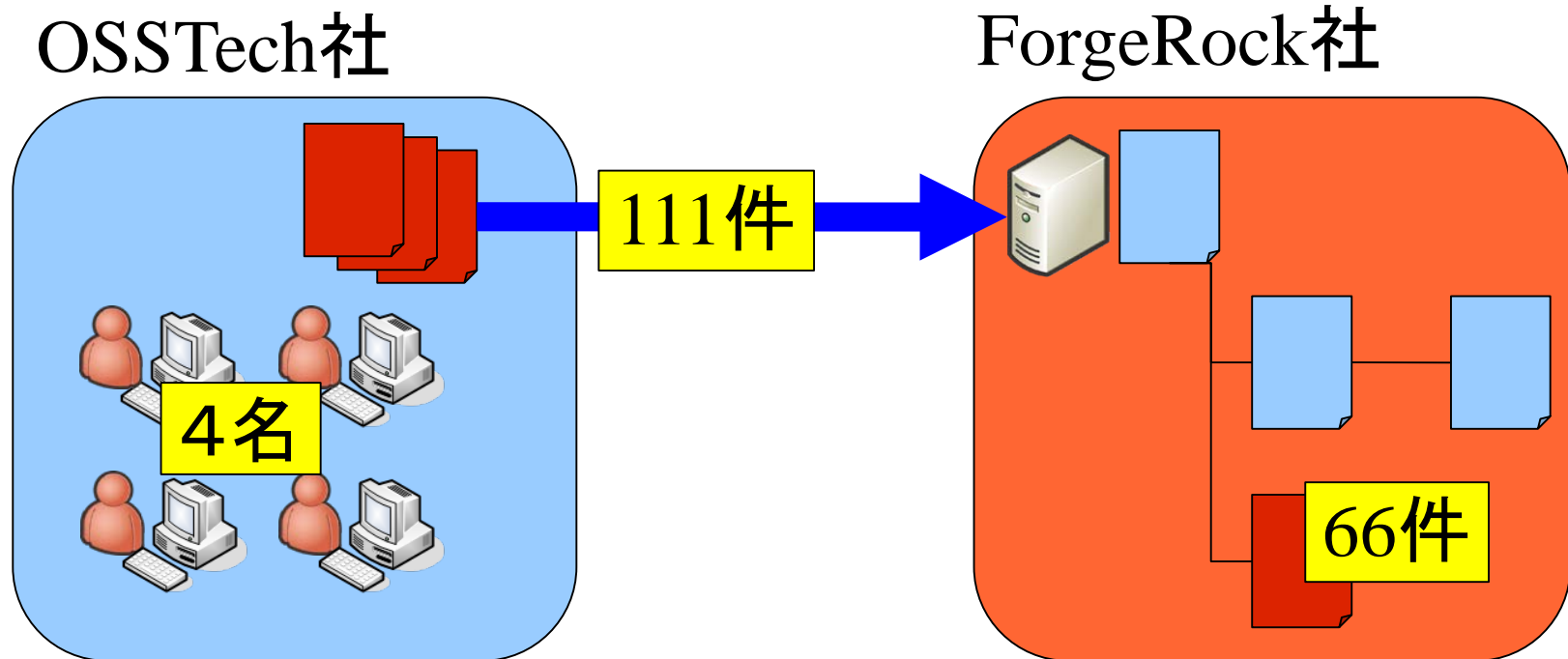
# コミット件数

- ・ 社内コミッターによるコミット件数111件
- ・ ForgeRockツリーのコミット総数は3346件



# 問題修正件数

- ・ 社内コミッターによる修正件数66件
- ・ 現在のチケット数1739(重複等含む)



## コミット例

- **nginxエージェント開発**
- **ユーザーデータストアの修正**
  - **JDBCの設定が共有されてしまう問題**
  - **ログ出力の改善**
- **コマンドツール(ssoadm など)の修正**
  - **エージェントの設定に不正な値が登録される問題**
  - **Windows 用ツールが動作しない問題**
- **ファイルアップロードの修正**
  - **Safari・Chromeでファイルアップロードが動作しない問題**

# コミット例

- ・ **メモリーリーク関連の修正**
  - エージェントのメモリーリーク
  - HTTPコネクションのリーク
  - DBコネクションのリーク
- ・ **マルチバイト文字**
  - マルチバイト文字の表示の問題
- ・ **日本語化**
  - 日本語表示の追加
  - 誤訳の修正

# OSSTech版カスタマイズ



# OSSTech版カスタマイズ

- OpenLDAPと親和性向上 > **自社ソリューション**
  - OpenAMにOpenLDAP専用の設定を追加
  - OSSTech社製OpenLDAP向け拡張スキーマを用意
  - OSSTech社製OpenLDAPをSHA-2対応にアドオンモジュール開発

OpenAM

ステップ 1/2: データストアのタイプを選択

戻る 次へ 取消し

\* 必須入力フィールド

\* 名前:

\* タイプ:

- Active Directory
- Active Directory アプリケーションモード (ADAM)
- OpenAM スキーマを含んだ Sun Directory Server
- OpenDS
- OpenLDAP
- Tivoli Directory Server
- データベースリポジトリ (アーリーアクセス)

# OSSTech版カスタマイズ

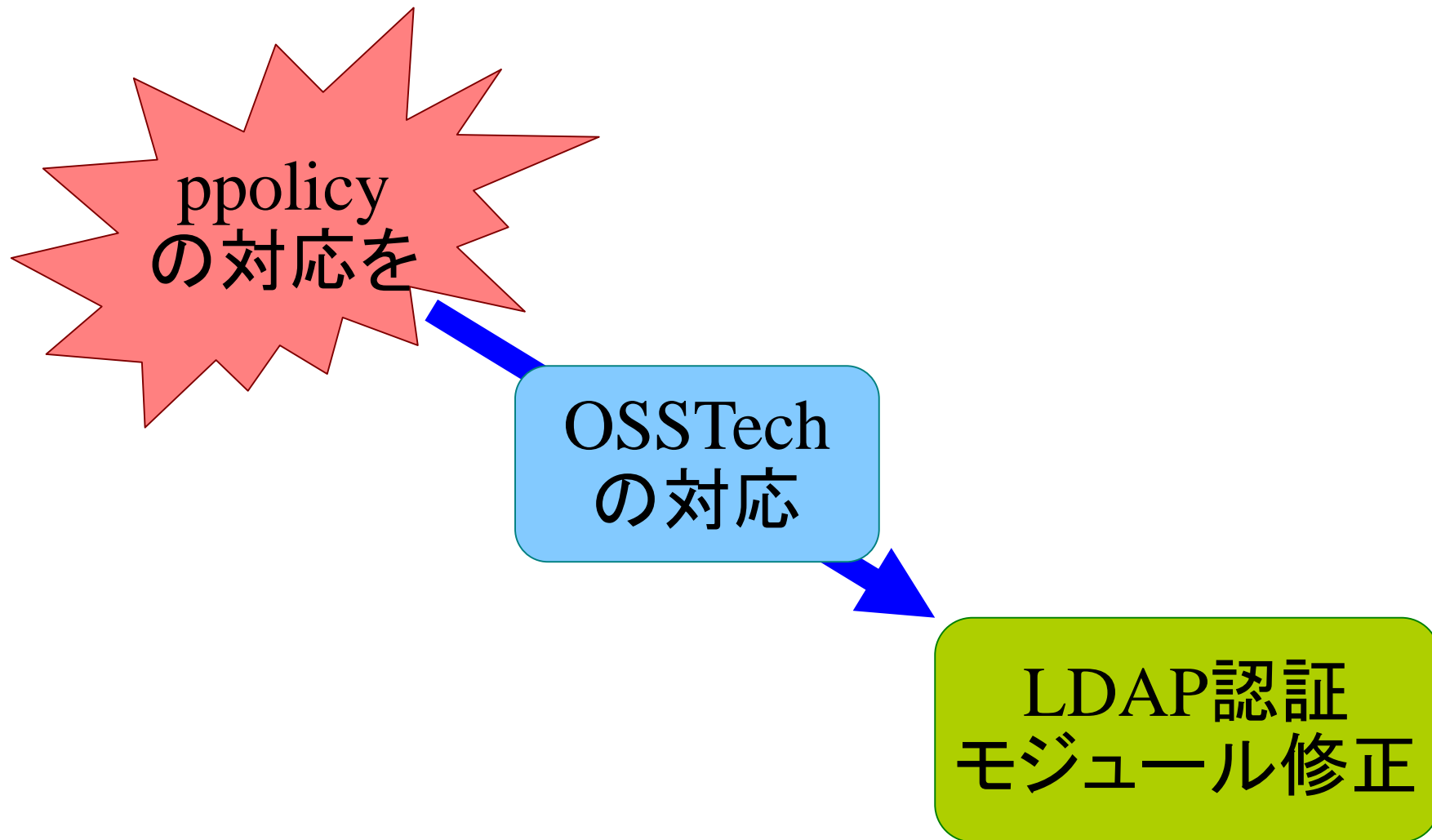
- Tomcatとの親和性向上 **> 環境の統一化**
  - OpenAM向けにパラメータを調整したTomcatをOpenAMとセットで提供
  
- パッケージング **> セットアップ容易化**
  - RPMパッケージとして提供
  - Windowsインストーラー提供

# OSSTech版カスタマイズ

- OpenAM10からのバックポート
  - 重要な修正、必要な機能をバックポート
  - 多重構成でのセッション数の共有
  - ポリシーの設定方法の改善
  - メモリリークの修正
- プラットフォーム毎にエージェントを提供
  - RHEL5でも動作可能なApache2.2エージェントの提供
- 日本語化
  - 画面の文字化け対策

# 案件個別カスタマイズ

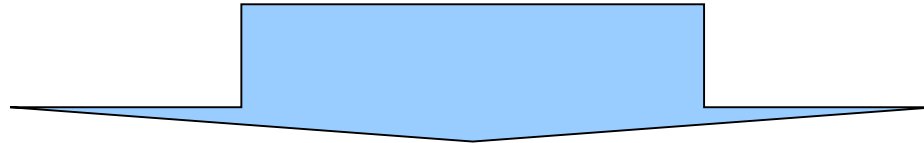
- OpenLDAP パスワードポリシー対応



# カスタマイズ事例

## ・ OpenLDAP パスワードポリシー対応

- ・ OpenAM-9.5系ではLDAP標準のアカウントポリシー※に未対応
- ・ 全て同一の認証エラーとなるという課題



- ・ LDAP認証モジュールにLDAPのアカウントポリシー(パスワード有効期限、ロック等)エラーをハンドリングし、個別のエラー遷移するよう改修

※アカウントポリシーとはパスワード有効期限、アカウントロックアウト等  
SunJavaDSやOpenDSを利用する場合は独自実装で対応されていた。  
OpenAM-10から対応開始

# カスタマイズ事例

- ・ 認証パラメータの追加

追加認証  
パラメータ

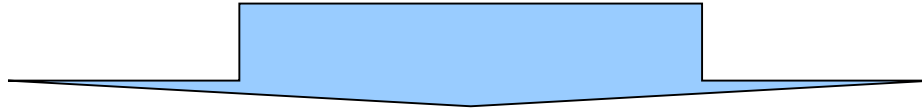
OSSTech  
の対応

LDAP認証  
モジュール修正

# カスタマイズ事例

## ・ 認証パラメータの追加

- ・ LDAP認証時に入力するユーザー名、パスワード以外に別パラメーターをドロップダウンリストとして表示し入力したい



- ・ LDAP認証モジュールを別パラメーター取得、表示可能なよう改修した

**OpenAM**

OpenAM へのサインイン

ユーザー名:

パスワード:



**OpenAM**

OpenAM へのサインイン

ユーザー名:

パスワード:

グループ:



# カスタマイズ事例

## ・ 認証モジュールの開発

特別な  
認証方式を

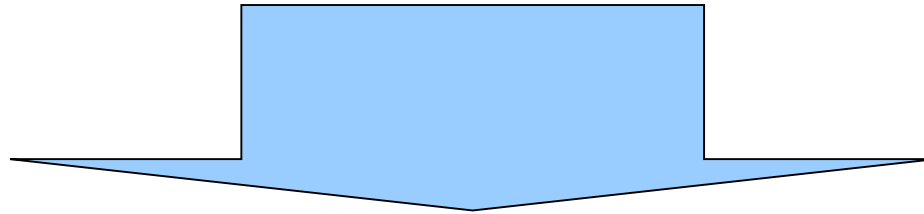
OSSTech  
の対応

新規認証  
モジュール開発

# カスタマイズ事例

## ・ 認証モジュールの開発

- ・ 既存認証モジュールでは対応できない認証方式(リクエストヘッダ)で認証したい



- ・ 要件に合った認証モジュールを開発し、既存モジュールとの認証連鎖とした。

# 開発事例

- Apacheより早いリバースプロキシを構築したい

リバース  
プロキシに  
パフォーマンスを！

OSSTech  
の対応

nginx用  
PolicyAgent開発

# 開発事例

- Apacheより早いリバースプロキシを構築したい

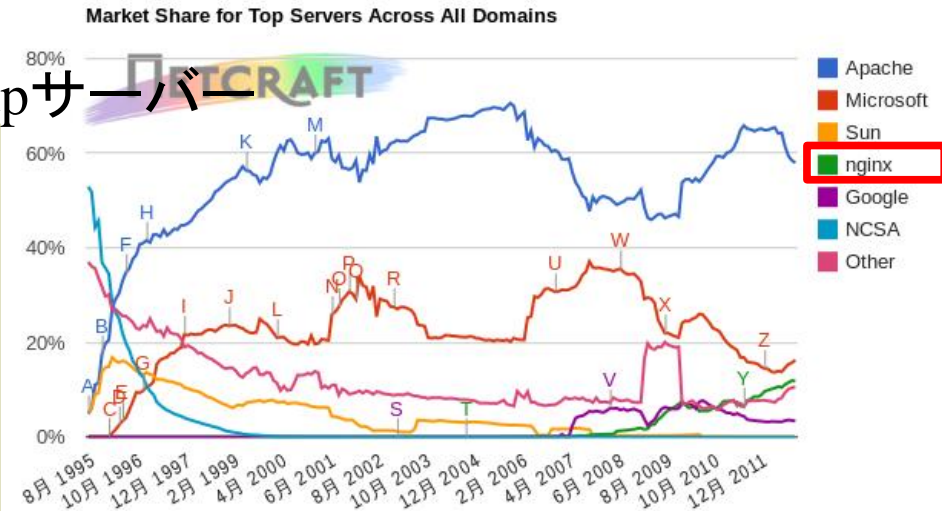
- Apacheによるリバースプロキシよりスケラビリティが欲しい



- nginx※用 Policy Agentの開発

※nginxとはスケラビリティ、パフォーマンスに優れる第三のhttpサーバー

netcraftの2011年資料  
第3位にnginxが伸びてきている  
apacheほど多機能ではないが、  
リバースプロキシ利用では  
十分な機能を持たせられる





OSSTech

**オープンソース・ソリューション・テクノロジー株式会社**

<http://www.osstech.co.jp/>

お問い合わせ [info@osstech.co.jp](mailto:info@osstech.co.jp)