

# OpenAM最新情報

野村総合研究所  
生産革新ソリューション開発二部  
OSS推進グループ  
和田 広之

## ● 所属部署

- ▶ 生産革新ソリューション開発二部 OSS推進グループ
- ▶ OSSを使ったシステム構築から運用までワンストップでサポート
- ▶ 対象OSSは50種類以上

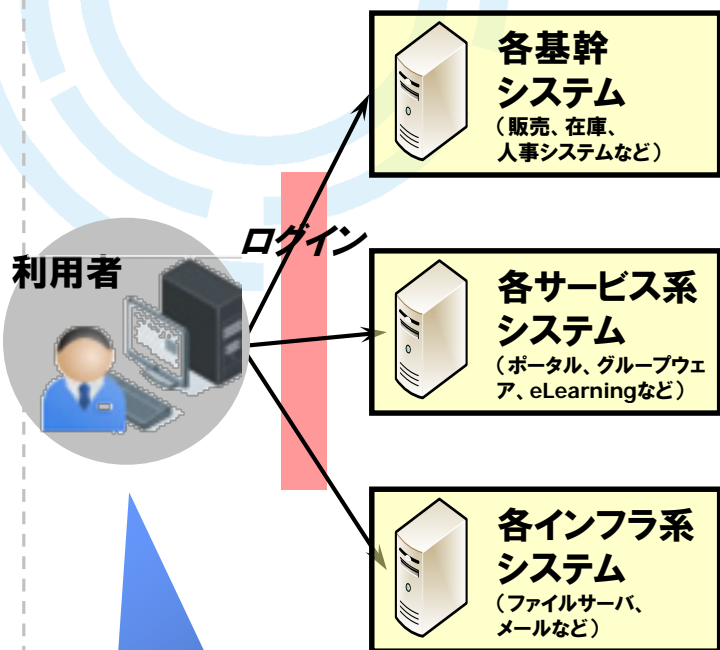
## ● 私の担当

- ▶ 各種OSSの技術的サポート
- ▶ OpenAM、OpenIDMの導入支援
- ▶ OpenAM、OpenIDMの機能拡張、バグ修正も実施
  - ▶ OpenStandiaチームから挙げたバグ・改善要望チケット数: 270件以上

# はじめに

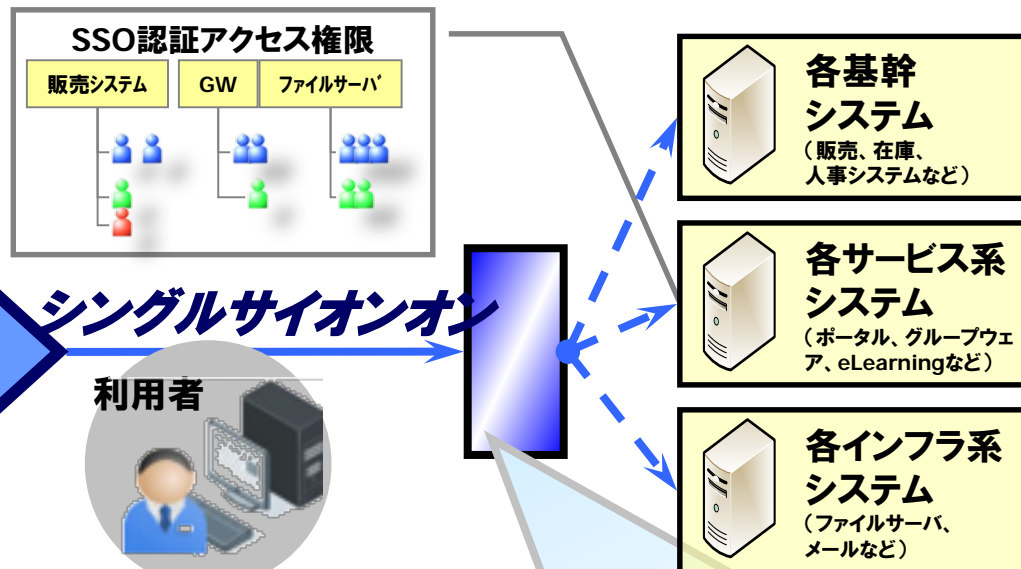
# シングルサインオン(SSO)について

## As-Is(現状運用)



各システム個別に、ID/パスワードで認証

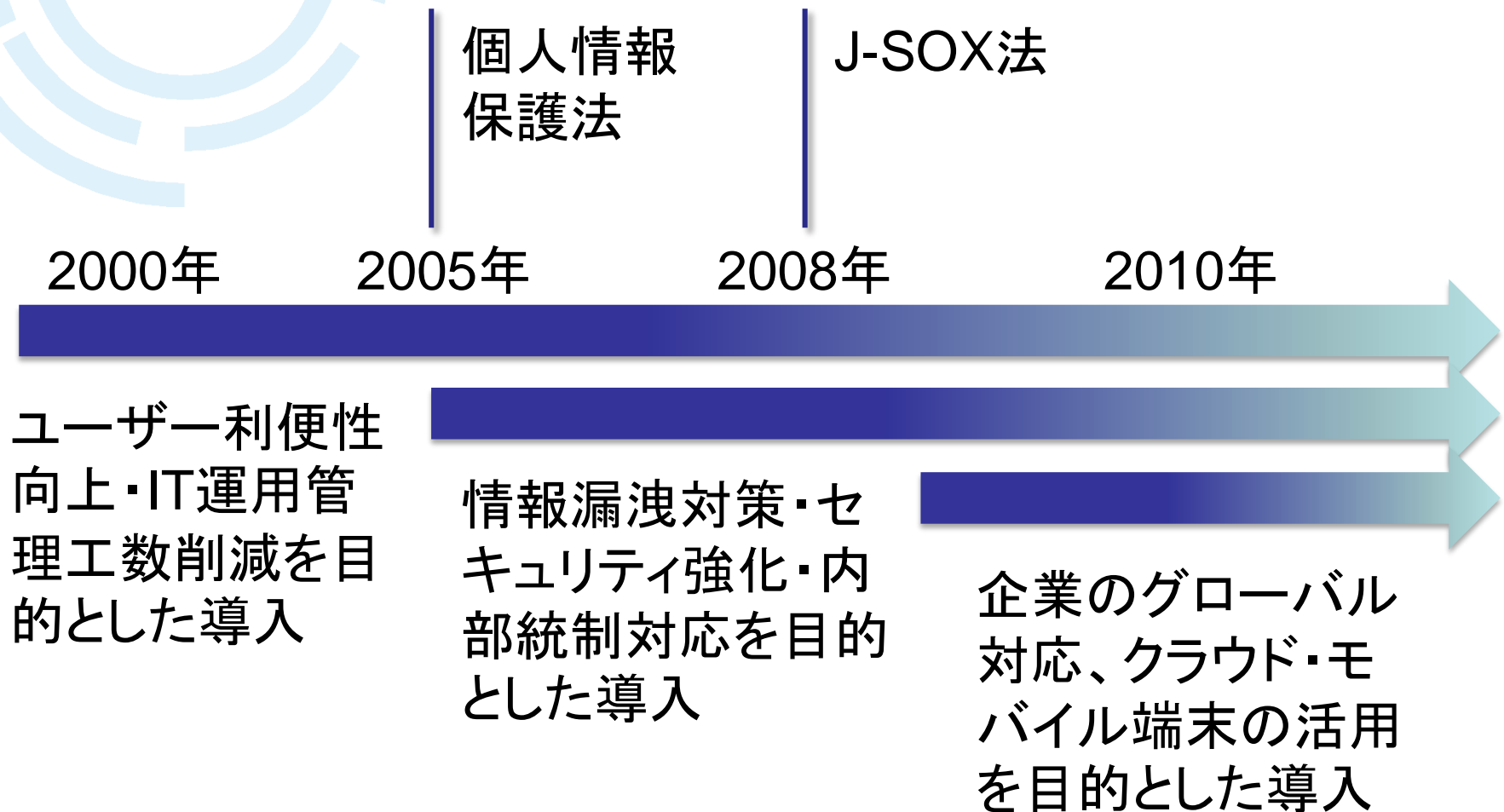
## To-Be(SSO導入後)



- アクセス権限付与に基づく厳密なアクセス制御
- セキュリティポリシーに基づいた厳密な認証インターフェース
- システムへのアクセスの認証の統合化による利便性向上
- アクセスログの一括取得
- 多様化する認証方式への対応
  - ✓ 対象システム : Web系システム、C/S系システム、OS...
  - ✓ 認証連携インターフェース : ID/PWD、生体認証、PKIなど

## SSOを実現するソフトウェア=OpenAM

## ●時代とともにSSO/IDMの導入目的も変化



# シングルサインオン・ID管理が求められる時代の環境変化

- 近年さまざまな環境変化により、企業内システム利用の在り方、及びそれに基づくID管理の在り方、認証の仕組みが見直されてきている

## 社内環境の変化

- システム、ユーザアカウント、権限の複雑化
- 内部統制・コンプライアンス・個人情報保護の強化
- 採用形態の複雑化(グローバル人材、アウトソース、出向等)

## IT環境の変化

- クラウド時代の到来による「所有」から「利用」への流れ
- 社内システムのSaaS利用
- モバイル端末、スマートフォン、タブレットの利用拡大
- 今後はIoTの活用

## 事業環境の変化

- グローバル化
- M & A、企業合併によるグループ企業の統廃合
- 新規サービス事業の開始

# 今後、認証基盤に求められる要件

## ● 環境変化

モバイル端末、タブレットなどの利用拡大

グローバル対応

SaaS・クラウドの活用

IoTの活用

## ● 求められる要件

スケーラビリティ

高度認証

認可処理

標準プロトコルによる  
相互接続性

- 既に国内でも多数の導入実績があり、基本機能は枯れている
- 環境変化に合わせて継続的にアップデート
- IoT時代を見据えた機能の追加・最新の標準仕様にも対応
  - ▶ 開発元のForgeRock社は、標準仕様の策定にも深く関わっている

**最新版で追加された新機能について紹介**



# OpenAMの最新情報

- 13.0.0が1月26日にリリース !!

- リリースノート

- ▶ <https://backstage.forgerock.com/#!/docs/openam/13/release-notes>

- ▶ 非公式ですが、弊社OpenStandiaチームのメンバーによる日本語訳があります。

- ✓ <https://t246osslab.wordpress.com/2016/01/30/openam-13-0-0がリリースされました/>

- 過去バージョンで非推奨となっていたいくつかの機能が削除されているので注意

- ▶ <https://backstage.forgerock.com/#!/docs/openam/13/release-notes#removed-functionality>

- ▶ /identity/attributes などのREST APIが削除に

## ● 本日紹介するアップデート内容

### 1. UI

✓ 管理コンソールの刷新

### 2. スケーラビリティ

✓ Stateless Session 機能の追加

### 3. 認証

✓ モバイルデバイスを使った2段階認証機能の強化

### 4. 認可

✓ 標準プロトコル UMA(User-Managed Access) に対応

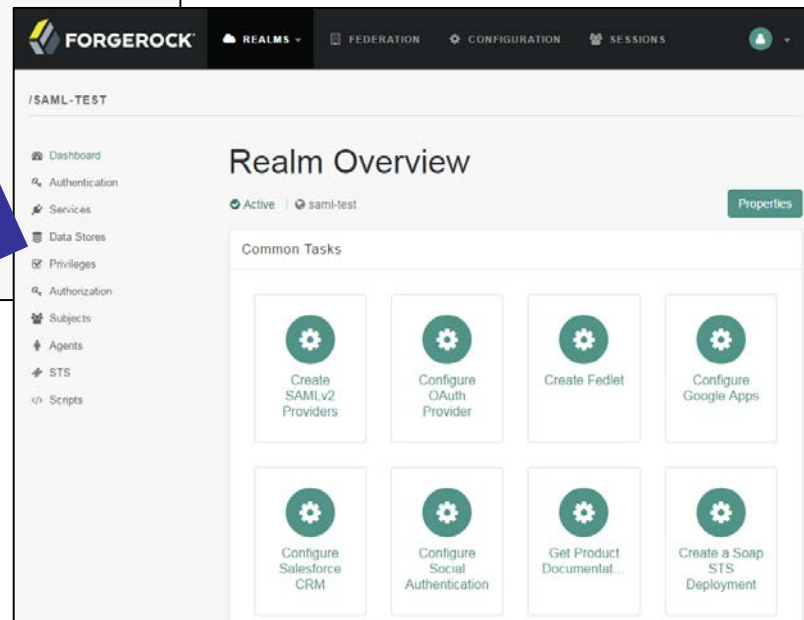
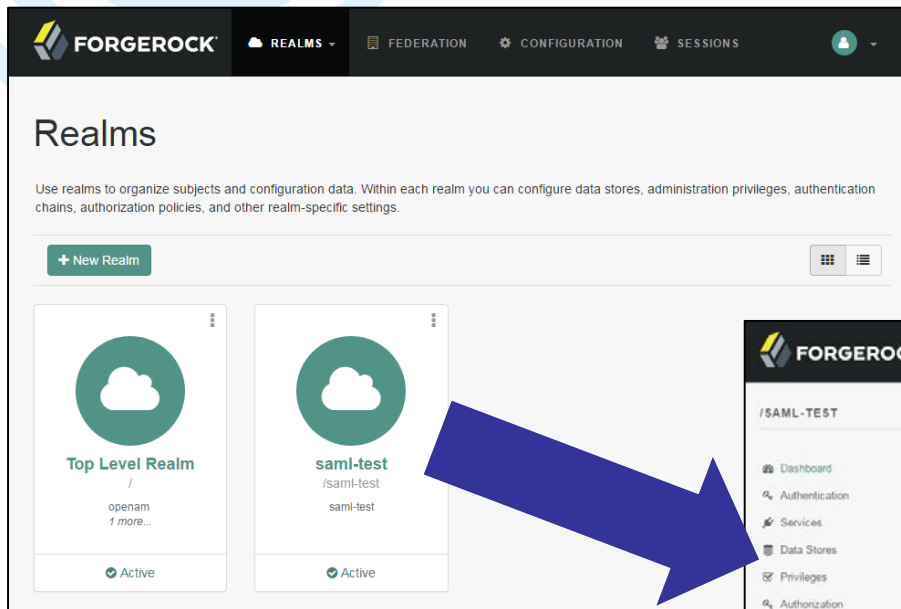
### 5. 開発者向け

✓ スクリプティングサービス

# 管理コンソールの刷新

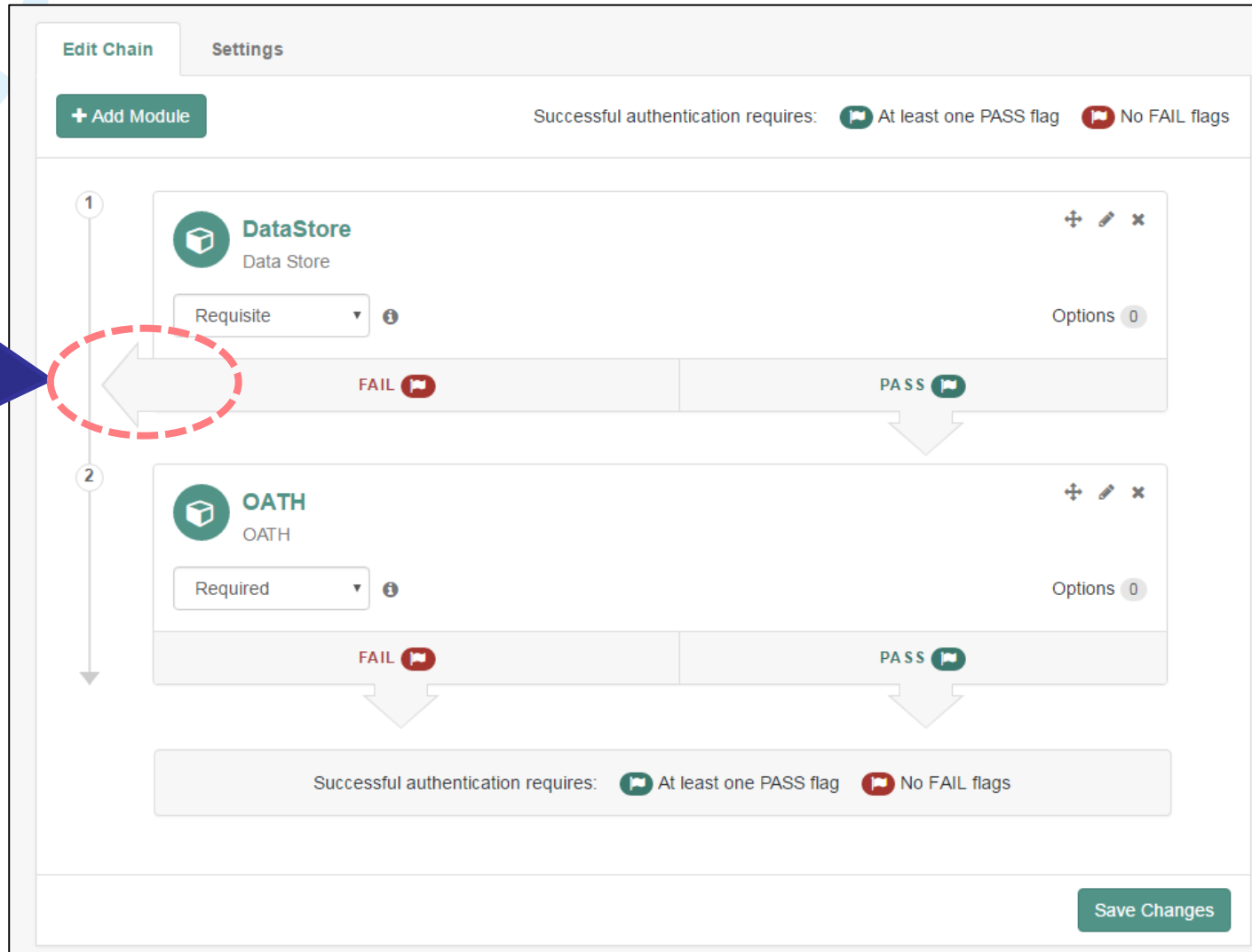
- デザインが今風のおしゃれな感じに
- レルムを中心としたUIに

最初にレルムを選択し、  
そのレルム内の設定を  
行うUIに



## ● 認証連鎖の設定はグラフィカルに

失敗すると即  
認証エラーと  
なること視覚  
的に表示



## ● ただし、まだレガシーUIの箇所も・・・

▶ 今後のさらなる改善に期待

The image shows a side-by-side comparison of the ForgeRock management console. On the left is the modern 'Realms' interface, featuring a clean design with a 'New Realm' button and two realm cards: 'Top Level Realm' and 'saml-test'. On the right is the legacy 'Trusts' interface, which is more cluttered and uses a different color scheme. A red dashed box highlights the navigation menu in the modern UI, and a large blue arrow points from it to the corresponding menu in the legacy UI, illustrating the transition.

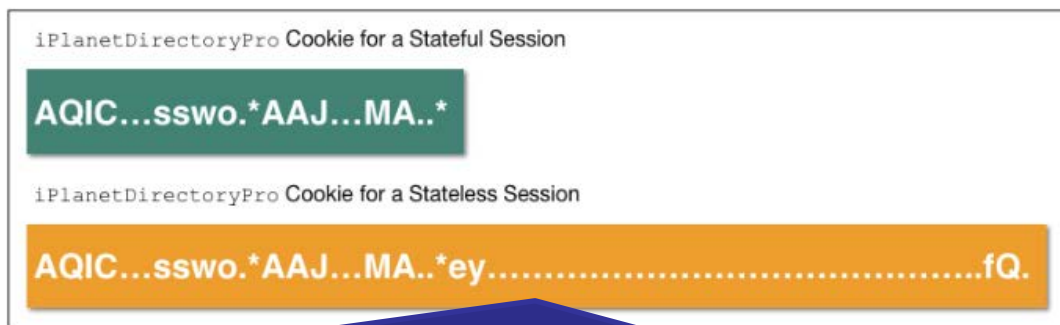
**Modern UI (Left):**

- Navigation: REALMS, FEDERATION, CONFIGURATION, SESSIONS
- Section: Realms
- Content: + New Realm, Top Level Realm, saml-test

**Legacy UI (Right):**

- Navigation: アクセス制御, 連携, 設定, セッション
- Section: トラストサークル設定
- Content: トラストサークル (0 項目), エンティティプロバイダ (0 項目), SAML 1x の設定

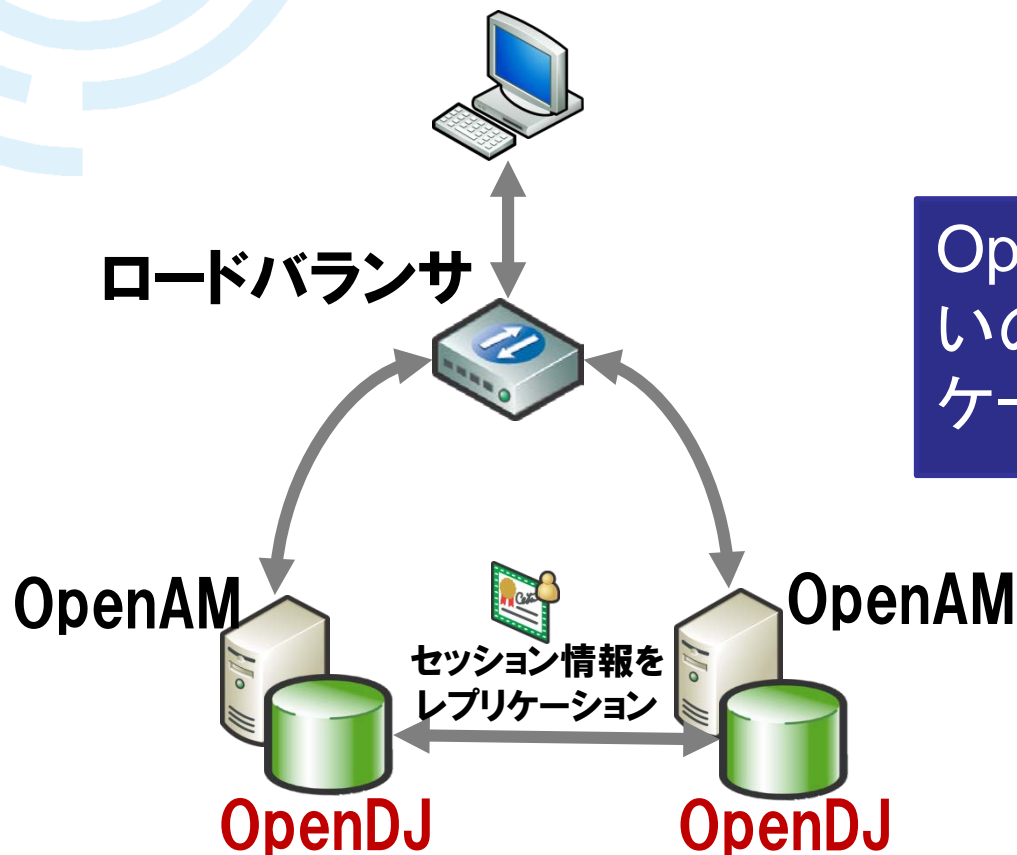
- セッションタイプにStatelessが選択可能に
  - ▶ 従来は、Stateful モードのみ
- Cookieの中にセッション情報を保存し、OpenAMサーバのメモリ上には持たない
  - ▶ スケールアウトが必要な大規模構成で有効
  - ▶ セッション情報はJWT(JSON Web Token)で格納されており、デジタル署名(改ざん防止)、暗号化も可能



CookieにJWT形式でセッション情報を保存

## 従来のセッション管理方式

- OpenAMが内蔵している組み込みLDAP(OpenDJ)にセッション情報を保存してレプリケーション

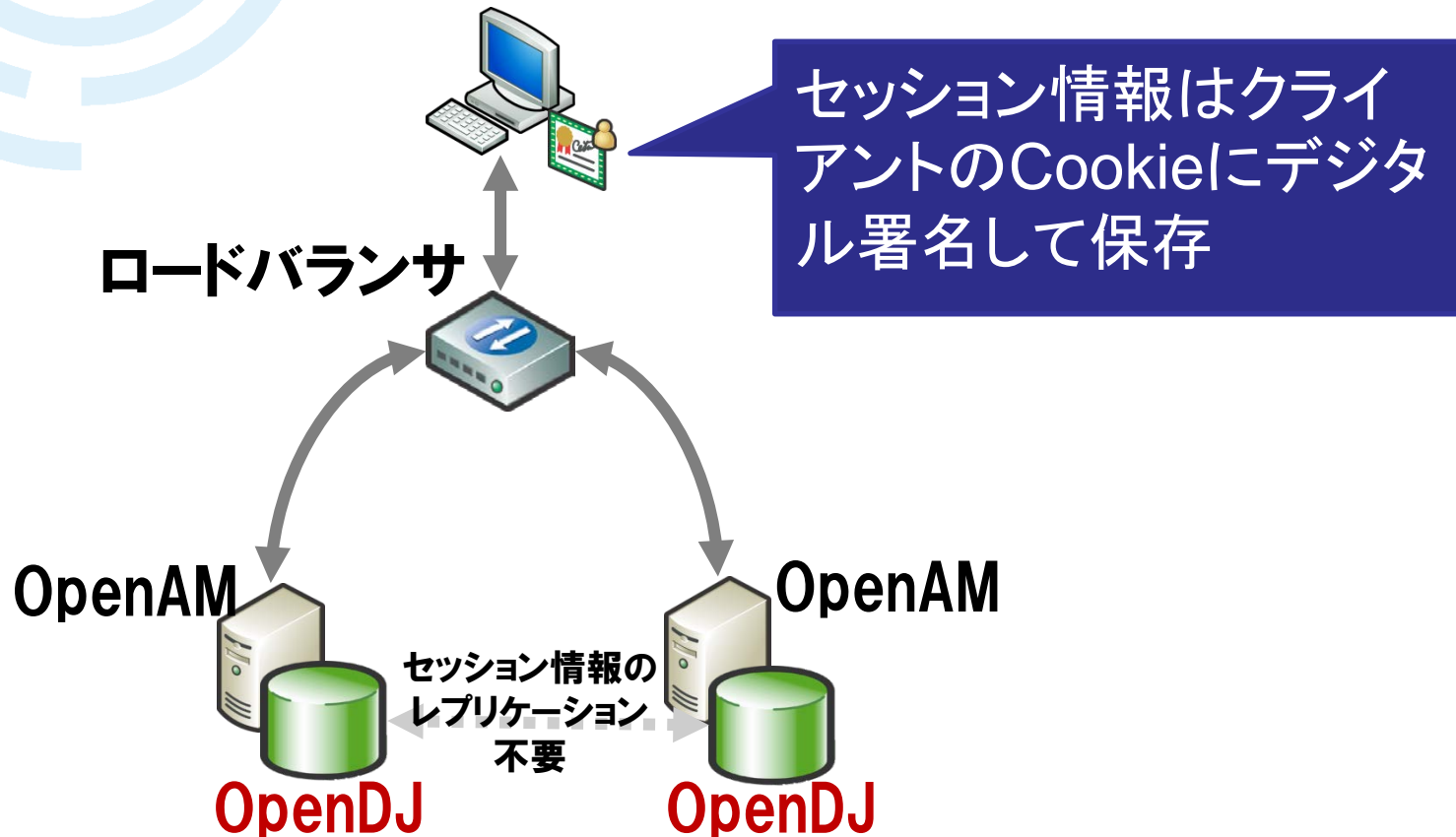




- 
- The diagram illustrates a multi-region OpenAM/OpenDJ architecture. It shows two regions: New York and Tokyo, each connected to a mobile device (smartphone) via a 'Preferred connection' and an 'Alternate connection' (represented by a map of Japan).
- New York Region:**
- Contains multiple OpenAM hosts (represented by rounded rectangles) and OpenDJ Directory Server hosts (represented by triangles).
  - Each OpenAM host is connected to an OpenDJ Directory Server host via LDAP (solid line with dots).
  - Each OpenDJ Directory Server host is connected to an OpenDJ Replication Server host (represented by a cylinder) via Directory replication (dashed line with arrows).
- Tokyo Region:**
- Contains multiple OpenAM hosts and OpenDJ Directory Server hosts.
  - Each OpenAM host is connected to an OpenDJ Directory Server host via LDAP.
  - Each OpenDJ Directory Server host is connected to an OpenDJ Replication Server host via Directory replication.
- Central Cloud:**
- Represents the network connecting the two regions.
  - Has bidirectional connections (dashed lines with arrows) to the OpenDJ Replication Server hosts in both the New York and Tokyo regions.
- Legend:**
- OpenAM host
  - OpenDJ Directory Server host
  - OpenDJ Replication Server host
  - HTTP
  - LDAP
  - Directory replication (not showing all connections)

## 新しいStateless Sessionの方式

- サーバ側でセッション情報を一切共有しないため、大規模構成でも容易にスケールする




## ● OpenAMの2段階認証

- ▶ 通常のユーザID/パスワード認証に加え、ワンタイムパスワード(OTP)を利用した2段階認証
- ▶ OTPの方式は標準仕様のOATHに準拠
  - ✓ HOTP、TOTPに対応
  - ✓ Android/iOSアプリのGoogle AuthenticatorなどをOTP発行機として利用可能

## ● OpenAM13での強化ポイント

- ▶ OPT発行のデバイス登録・管理のUIが標準で追加

# 利用イメージ



**FORGEROCK™**


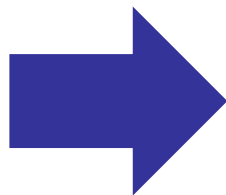
OPENAM へのサインイン

test1

\*\*\*\*\*

☐ Remember my username

LOG IN



**FORGEROCK™**


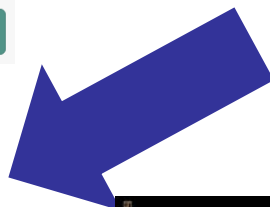
FORGEROCK AUTHENTICATOR (OATH)

Enter verification code

SUBMIT

REGISTER DEVICE


SKIP THIS STEP




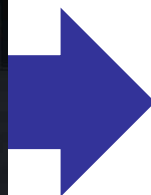
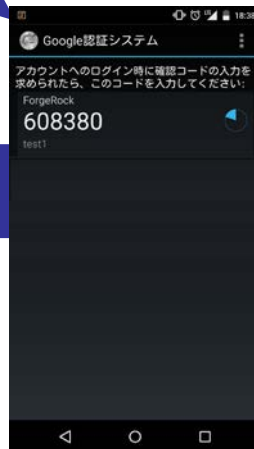
**FORGEROCK™**

REGISTER YOUR DEVICE WITH OPENAM

Scan the barcode image below with the ForgeRock Authenticator App. Once registered click the button to enter your verification code and login.



LOGIN USING VERIFICATION CODE



**FORGEROCK™**

FORGEROCK AUTHENTICATOR (OATH)

288705

SUBMIT

SKIP THIS STEP

デバイス登録画面を標準で用意

## ● UMA Authorization Server 機能の追加

- ▶ UMAとは、Kantara Initiativeのワーキンググループで仕様策定されたWebベースのアクセス管理プロトコル
- ▶ 現在Webで一般的に使われる認可プロトコルのOAuth2を拡張したもの
  - ✓ Facebookの例

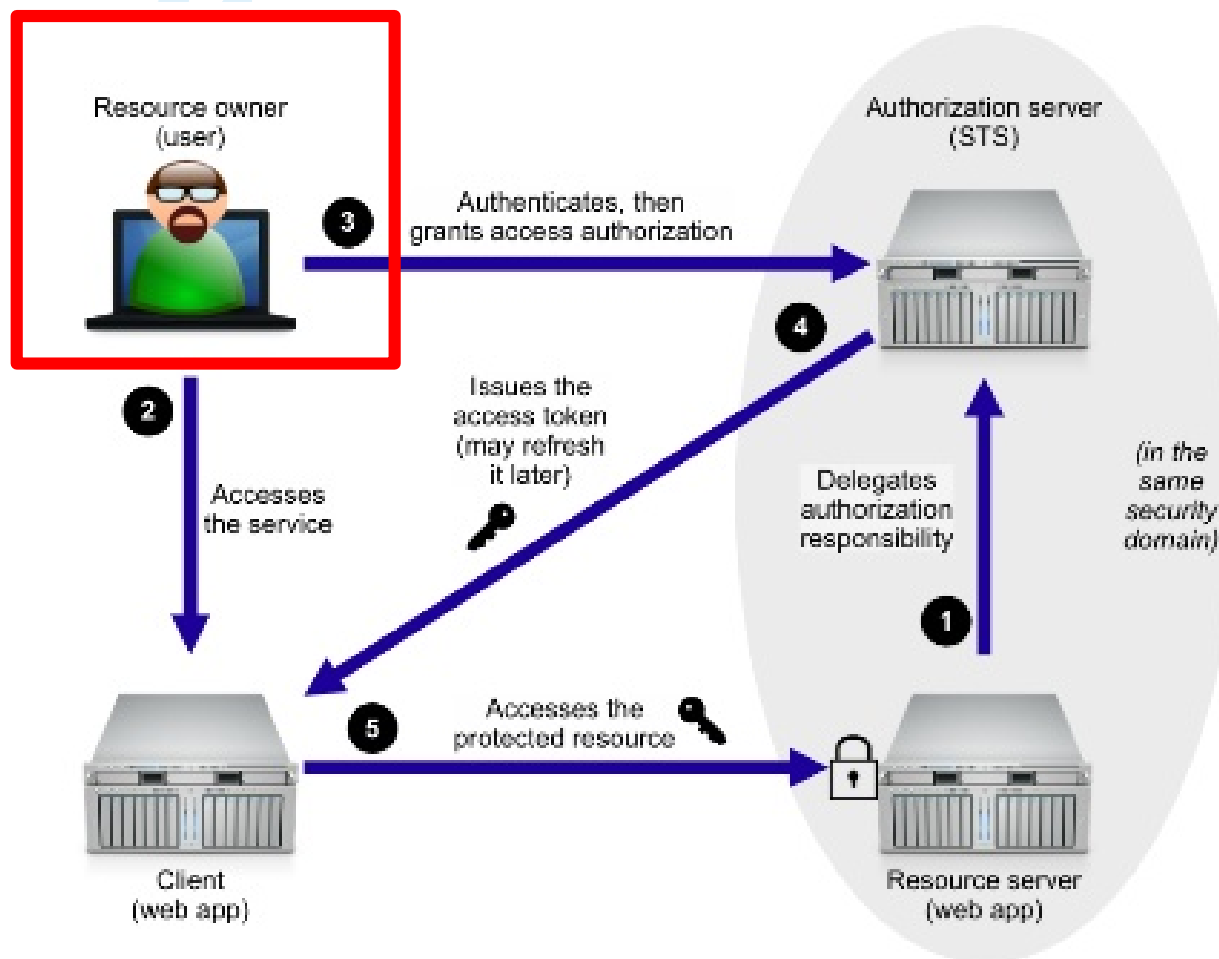


出所: <http://blog.unfindable.net/archives/1891>

## ▶ OAuth2との大きな違い

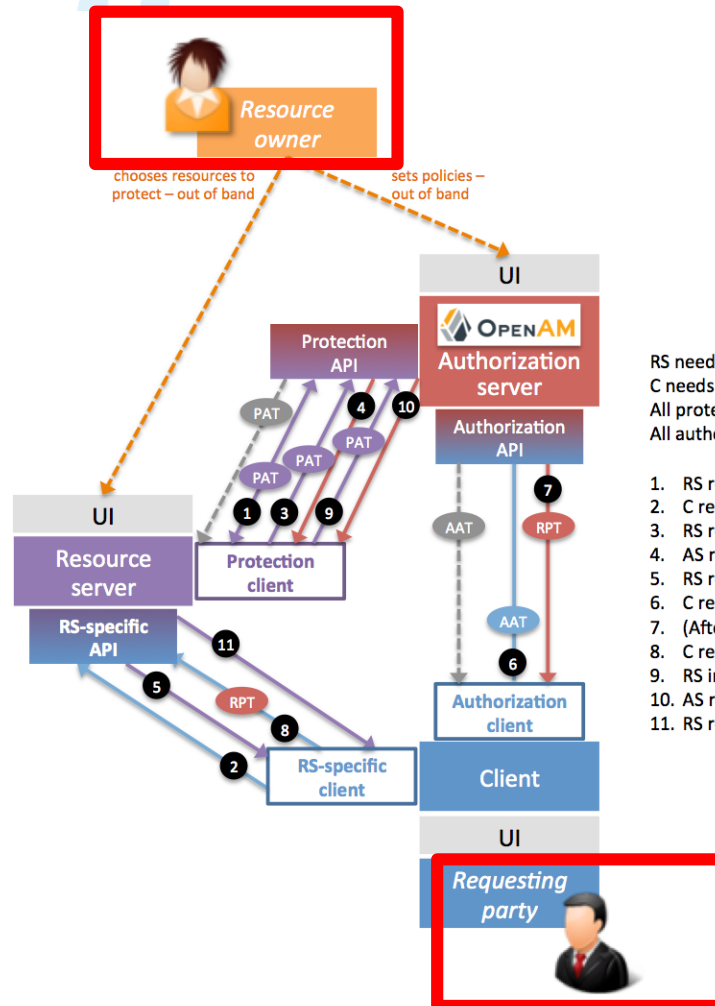
- ✓ OAuth2はユーザーが外部のアプリケーションから自分のデータにアクセスするユースケースを想定
- ✓ UMAは自分のデータを他人が見れるように権限を設定するといった、他人とデータを共有するユースケースも想定

## ●リソースオーナー = 利用者



出所: <http://image.slidesharecdn.com/uma-for-ace-150114044632-conversion-gate02/95/uma-for-ace-10-638.jpg?cb=1421210822>

## ●リソースオーナー ≠ 利用者 (でも良い)



RS needs OAuth client credentials at AS to get PAT

C needs OAuth client credentials at AS to get AAT

All protection API calls must carry PAT

All authorization API calls must carry AAT

1. RS registers resource sets and scopes (ongoing – CRUD API calls)
2. C requests resource (provisioned out of band; must be unique to RO)
3. RS registers permission (resource set and scope) for attempted access
4. AS returns permission ticket
5. RS returns error 403 with as\_uri and permission ticket
6. C requests authz data, providing permission ticket
7. (After claims-gathering flows not shown) AS gives RPT and authz data
8. C requests resource with RPT
9. RS introspects RPT at AS (if using default “bearer” RPT profile)
10. AS returns token status
11. RS returns 20x

出所: <https://forgerock.org/app/uploads/2014/10/uma-info-highres.png>

## ● UMAの処理は大きく3つに分かれる

### ▶ 1: リソースの登録

✓リソースオーナー/リソースサーバ/認可サーバ  
でのやりとり

### ▶ 2: 認可処理

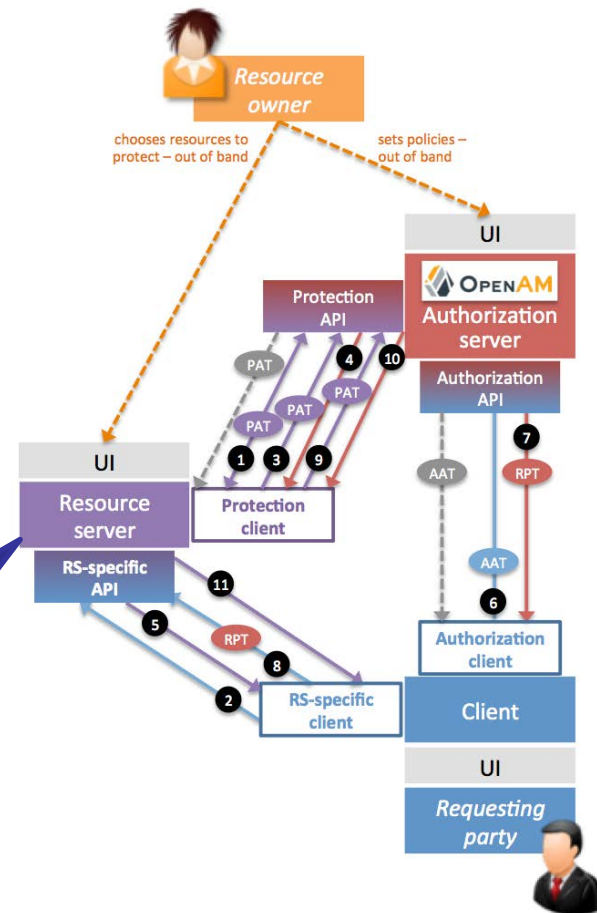
✓リソースサーバ/認可サーバ/クライアント/  
Requesting Partyでのやりとり

### ▶ 3: リソースアクセス

✓Requesting Party/リソースサーバ/認可サーバ  
でのやりとり

最終的には、リソースサーバにて、クライアントから渡された  
トークン (RPT) から権限情報を取りアクセス制御する

```
{
  "active": true,
  "exp": 1447051542,
  "iss": "http://openam.example.org:8080/openam/oauth2",
  "token_type": "requesting_party_token",
  "permissions": [
    {
      "resource_set_id": "592e20b8-8f43-48a0-90cf-eae722ced36b0",
      "scopes": [
        "http://photoz.example.com/dev/scopes/view",
        "http://photoz.example.com/dev/scopes/all"
      ]
    }
  ]
}
```





### ● 写真を知り合いに共有、といった個人データの共有

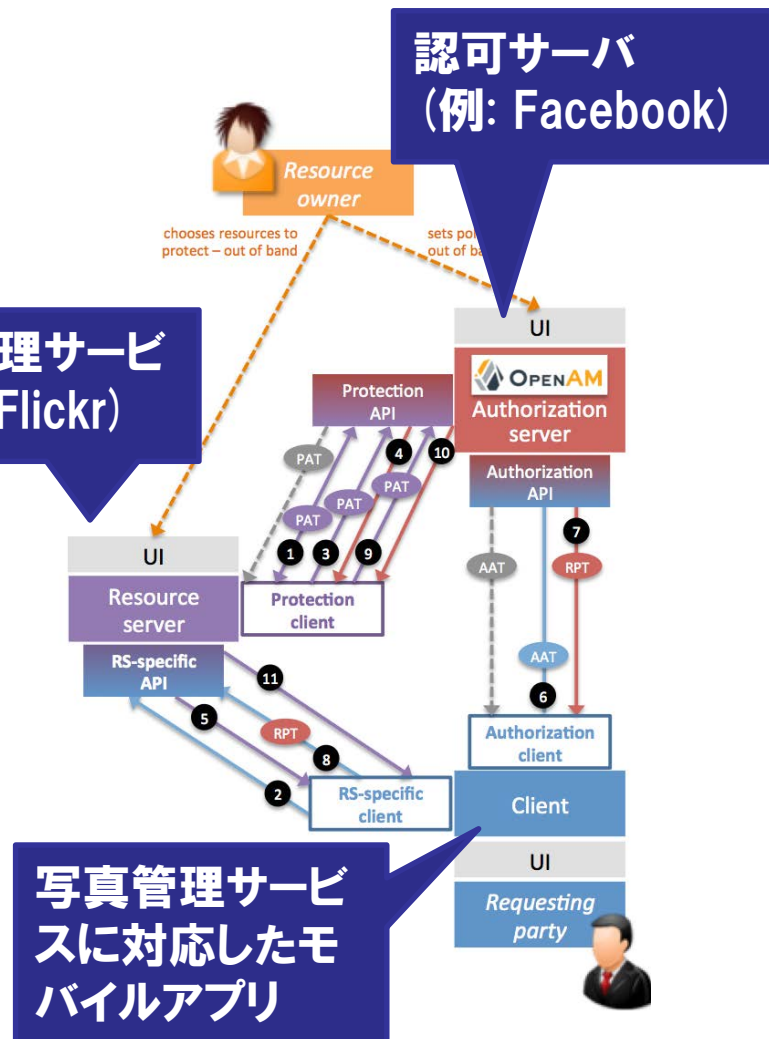
- ▶ よくあるのは、自動生成されたURLをメール等で送信して共有するなど
- ▶ これだと、URLが他人に渡ると見れてしまう

### ● UMAを使うと...

- ▶ ユーザーは共有したい写真に対する参照権限を認可サーバに登録
- ▶ 認可サーバにて、参照権限のポリシーを設定(公開範囲を友人、など)
- ▶ 友人は認可サーバにて許可を得ることで、クライアントアプリは写真管理サービスからデータを取得可能になる

写真管理サービス  
(例: Flickr)

認可サーバ  
(例: Facebook)



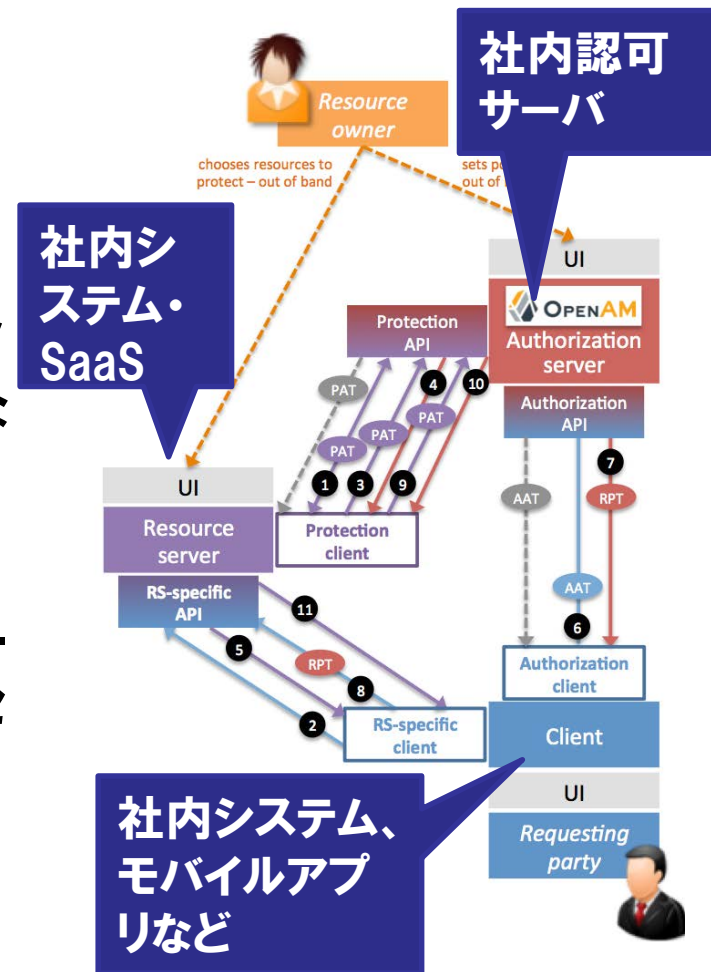
写真管理サービス  
に対応したモ  
バイルアプリ

## ● 企業内のリソースに対するアクセス制御に利用

- ▶ 従来型の認証基盤だと、リバプロ/エージェントの導入が必要
- ▶ URLベースでの制御しかできない
- ▶ モバイルアプリや外部サービス等、リバプロ/エージェントが導入できない環境では使えない

## ● UMAを使うと...

- ▶ リソースオーナーは企業であり、リソースサーバ(社内システム・SaaSなど)に対するアクセス権限を認可サーバに設定する
- ▶ 認可サーバにて権限のポリシーを設定(〇〇部は見れる、部長以上は承認できる、など)
- ▶ 社員は認可サーバにて許可を得ることでアクセス可能になる



出所:

<https://forgerock.org/app/uploads/2014/10/uma-info-highres.png>

オープンソースまるごと

## ● 従来

- ▶ Java 言語で認証モジュールなどのカスタマイズが可能

## ● OpenAM13での強化ポイント

- ▶ 下記をスクリプト(JavaScriptまたはGroovy)にてカスタマイズ可能に
  - ✓ 認証ロジック
  - ✓ 認可ポリシー条件
  - ✓ OpenID Connectクレーム

## 例) 認証ロジックのカスタマイズ

- OpenAM管理コンソールよりスクリプトを定義
- AM9時～PM5時の間しか認証できないようにするサンプルソースが付いている

SCRIPT

</>

Scripted Module - Server Side

Delete

Name

Scripted Module - Server Side

Description

Default global script for server side Scripted Authentication Module

Script Type

Server-side Authentication

Change

Language

☐ JavaScript

☒ Groovy

Script

```
14
15 if (username) {
16     // Fetch user information via REST
17     var response = httpclient.get("http://localhost:8080/openam/json/users/" + username,
18         cookies : [],
19         headers : []
20     );
21     // Log out response from REST call
22     logger.message("User REST Call. Status: " + response.getStatusCode() + ", Body: " + r
23 }
24
25 var now = new Date();
26 logger.message("Current time: " + now.getHours());
27 if (now.getHours() < START_TIME || now.getHours() > END_TIME) {
28     logger.error("Login forbidden outside work hours!");
29     authState = FAILED;
30 } else {
31     logger.message("Authentication allowed!");
32     authState = SUCCESS;
33 }
```

Upload

Validate

Edit Fullscreen

Save

# まとめ

- **環境変化により、企業内の認証基盤に求められる要件も変化**
  - ▶ 近年のキーワードとしては、SaaS/グローバル/モバイル/IoT
- **オープンソース認証基盤であるOpenAMは変化にいち早く対応**

本資料に掲載されている会社名、製品名、サービス名は各社の登録商標、又は商標です。

- OpenStandiaは、「攻めのIT」を支援します。
- オープンソースのことなら、なんでもご相談ください！



お問い合わせは、NRI OpenStandiaチームへ



osscc@nri.co.jp



<http://openstandia.jp/>