「ThemiStruct Identity Platform」クックブック 2018

テミストラクト　アイデンティティプラットフォーム

～ 共通ID基盤、API連携認証システムを題材に、デモを交えて～

株式会社オージス総研
事業開発本部 テミストラクトソリューション部

鈴川 竜司

© 2018 OGIS-RI Co., Ltd.

統合認証パッケージ
ThemiStruct Identity Platform のご紹介

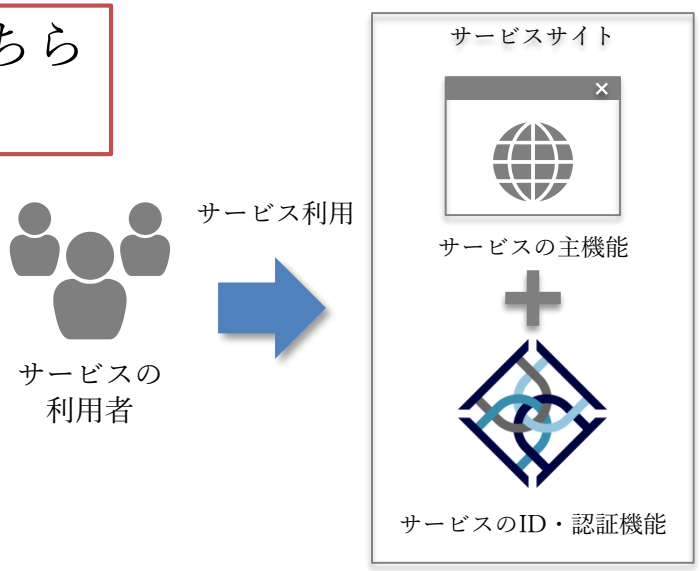
© 2018 OGIS-RI Co., Ltd.

統合認証パッケージ ThemiStruct Identity Platform

ThemiStruct Identity Platform は、ITシステム利用者のアイデンティティ管理と認証の機能を提供します。顧客向けにサービスを展開したり、従業員向けにアプリケーションを展開する際に必要となる、共通ID基盤の構築に活用いただけます。

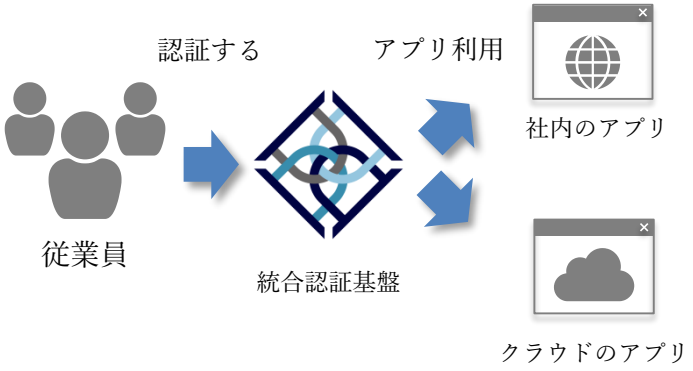
本ご紹介ではこちらを中心に説明

顧客向けサイト領域に活用



サービスサイトの
ID、認証の共通機能として利用

エンタープライズ領域に活用



エンタープライズアプリ群の
SSOやアクセス制御の基盤として利用

© 2018 OGIS-RI Co., Ltd.

『顧客向けサイト』に向けた3大特長

ロ　短期間での導入が可能

- AWSマネージドサービスのコンポーネントを活用し、導入時におけるキャパシティやアベイラビリティプラニングから解放します。スモールスタート、突発的アクセス集中に対応できる伸縮可能かつ事業継続性を考慮した認証基盤が、2時間でセットアップできます。

ロ　IDとユーザー認証、アクセス認可の機能をアプリに組み込み

- 利用者のID情報の管理や認証など利用者IDに関連する共通機能の実装を支援する『フレームワーク』と『Web API』を提供します。これらを活用し、貴社のサービスサイトに認証機能やパスワード変更機能、アプリケーションへのID連携機能などを組み込むことができます。

ロ　Web APIの認証・アクセス認可に対応

- OAuth 2.0 の技術仕様に基づいた認証・アクセス認可機能を提供します。OAuth 2.0 を使ったAPI連携認証システムが実現でき、モバイルアプリ、オープンAPIを安全に公開できます。

© 2018 OGIS-RI Co., Ltd.

本講演でのご紹介内容

ロ　短期間での導入が可能

- AWSマネージドサービスのコンポーネントを活用し、導入時におけるキャ
  ご紹介１：セットアップしてみる
  します。スモールスター
  つ事業継続性を考慮した
  認証基盤が、2時間でセットアップできます。

ロ　IDとユーザー認証、アクセス認可の機能をアプリに組み込み

- 利用者のID情報の管理や認証など利用者IDに関連する共通機能の実装を支
  ご紹介２：共通ID基盤として構成してみる
  を活用し、
  ケーショ
  ンへのID連携機能などを組み込むことができます。

ロ　Web APIの認証・アクセス認可に対応

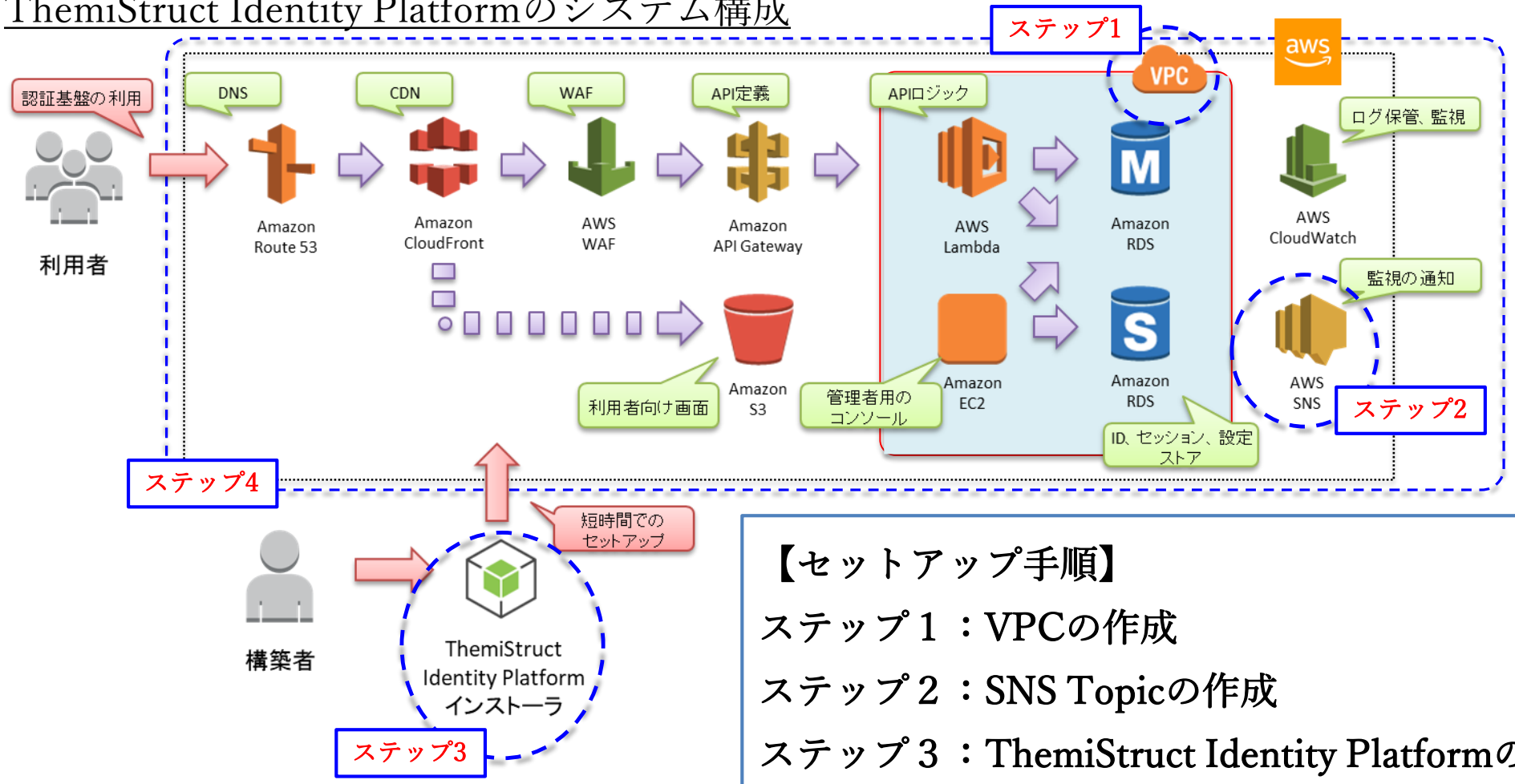
- OAuth 2.0 の技術仕様に基づいた認証・アクセス認可機能を提供します
  ご紹介３：API連携認証システムとして構成してみる
  オープンAPIを安全に公開できます。

© 2018 OGIS-RI Co., Ltd.

セットアップしてみる

セットアップの概要：セットアップは４ステップで実現

ThemiStruct Identity Platformのシステム構成



【セットアップ手順】

ステップ１：VPCの作成

ステップ２：SNS Topicの作成

ステップ３：ThemiStruct Identity Platformのインストーラー配備

ステップ４：インストーラで残るコンポーネントを構築

© 2018 OGIS-RI Co., Ltd.

ステップ１．VPCを作成

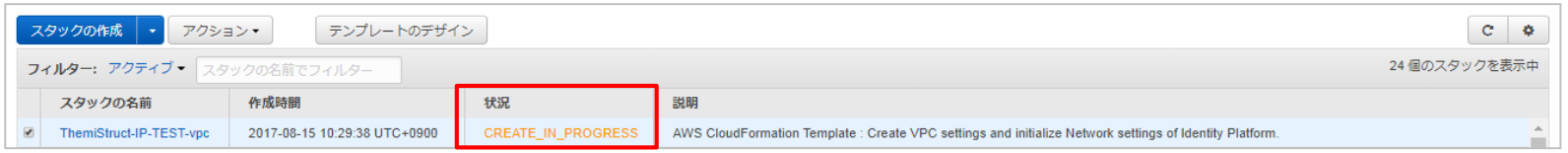
■CloudFormationとテンプレートを用いてVPCを作成

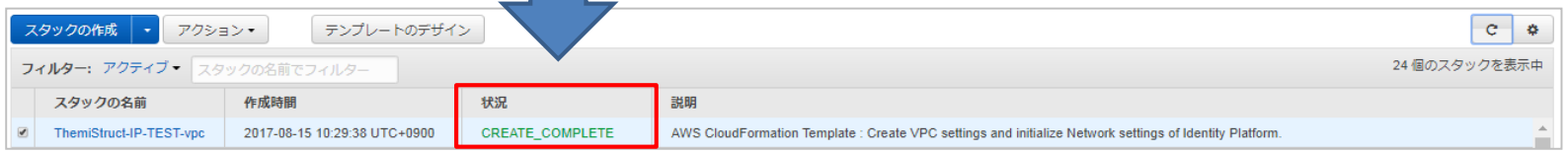
ThemiStruct Identity Platformより提供 → identityplatform-vpc.json

VPCのサブネット情報入力
Identity Platform専用のネットワークなため、基本は入力値のまま

© 2018 OGIS-RI Co., Ltd.

ステップ１. VPCを作成



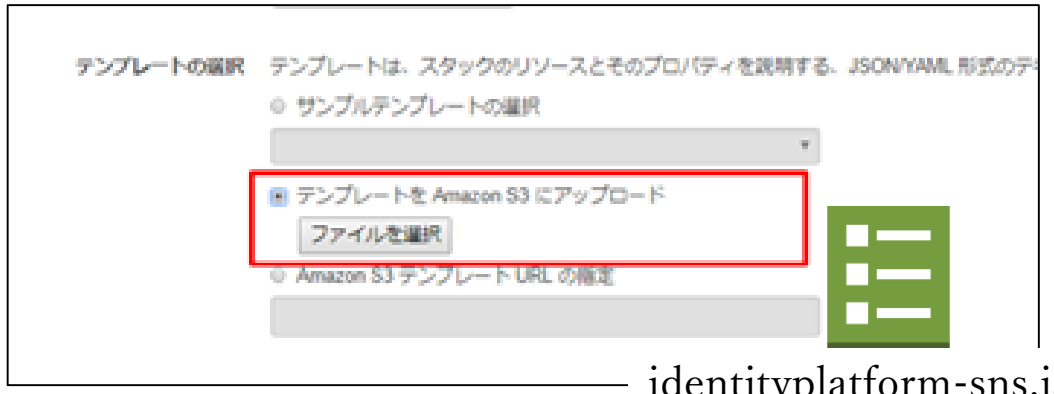


約10分で作成完了

作成結果の出力は後工程の入力で利用

2. SNS Topicを作成

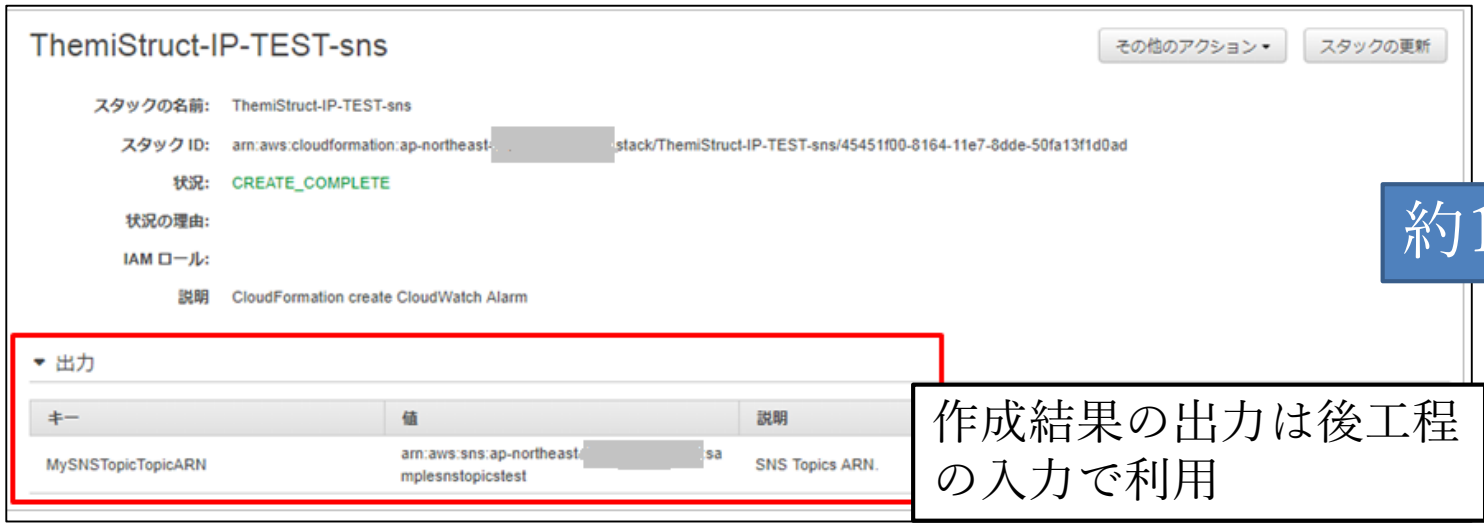
■CloudFormationとテンプレートを用いてSNS Topicを作成



identityplatform-sns.json

MailAddress（監視アラート通知先）

TopicName（トピックの識別子）



約10分で作成完了

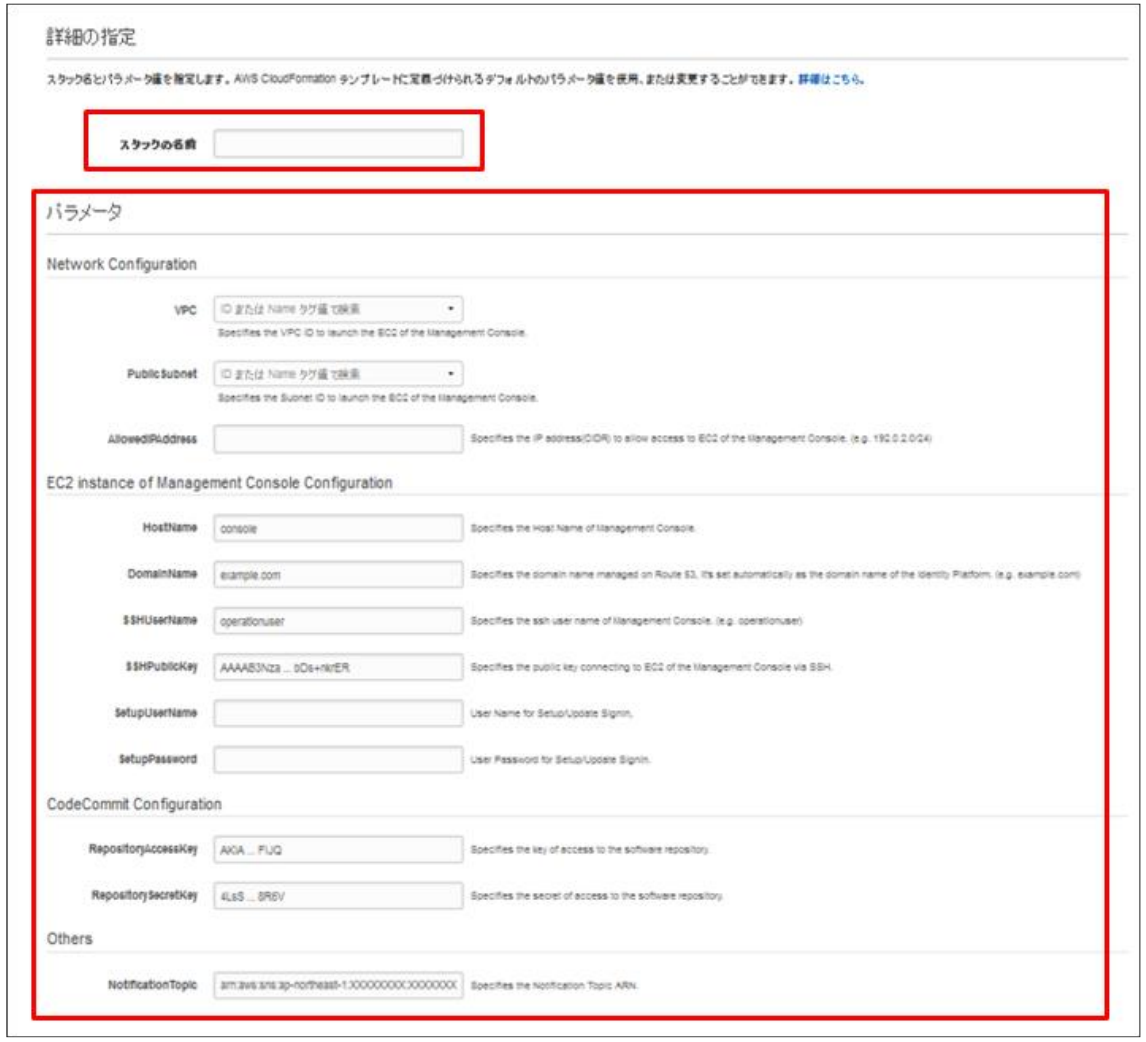
作成結果の出力は後工程の入力で利用

© 2018 OGIS-RI Co., Ltd.

3. インストーラの配備

■CloudFormationとテンプレートを用いてThemiStruct Identity Platformインストーラを配備

identityplatform-setup.json



- Identity Platformのソフトウェアダウンロードキー
- インストーラーのホスト名／ドメイン名
- インストーラーへログインするためのユーザ名／パスワード
- インストーラー用サーバ証明書（HTTPS実現用）
- インストーラーのOSのユーザ名
- インストーラーへアクセス（SSH）用の公開鍵

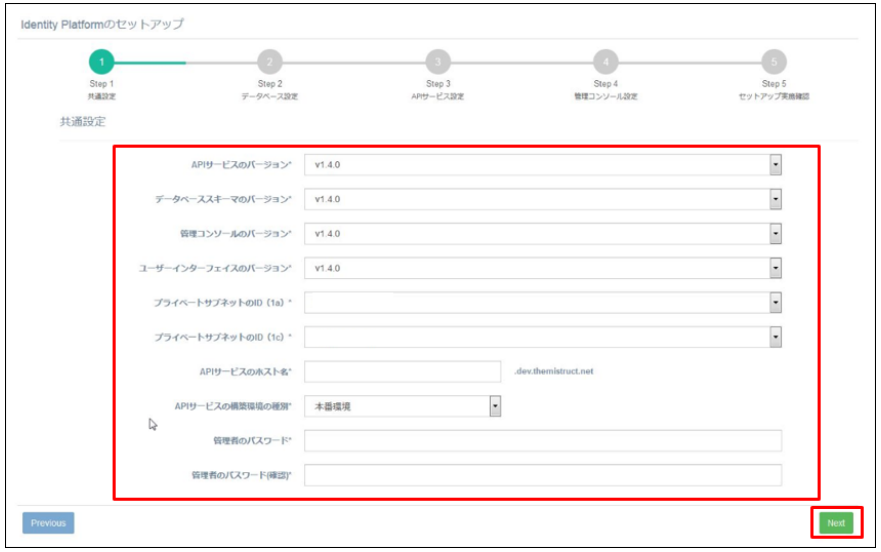
（前工程完了時の出力結果）

© 2018 OGIS-RI Co., Ltd.

4. インストーラを実行

約20分で作成完了

4. インストーラを実行

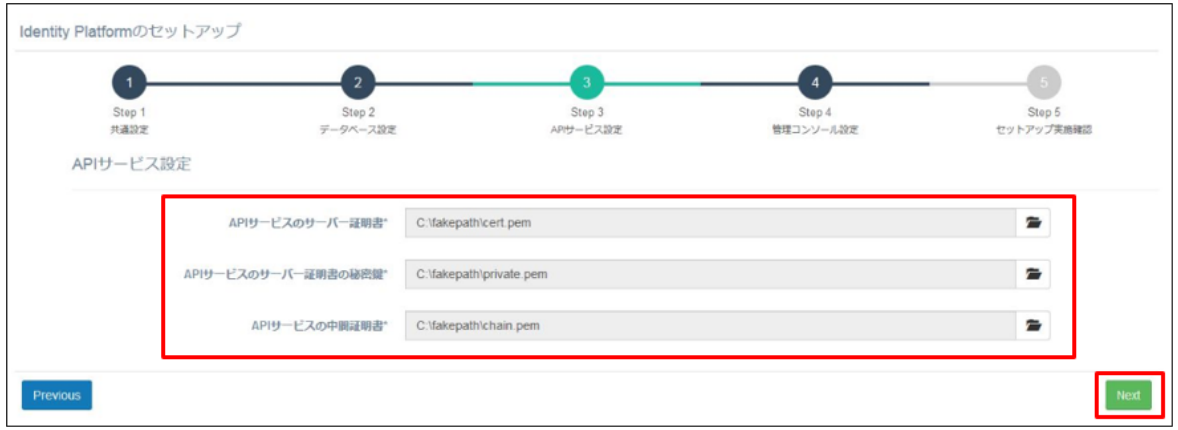


共通の設定
認証／APIサービスのホスト名
構築環境の種別
システム管理者のパスワード
（前工程完了時の出力結果）

DBの設定
データストアタイプの選択 冗長性：Amazon Aurora 低冗長性：RDS for MySQL
データベースのMulti-AZ配置（有効or 無効）
データベースのインスタンスクラス
データベースのストレージサイズ
データベースのマイナーバージョン自動更新（有効or無効）
週次メンテナンス期間：上記の実施曜日／時間
バックアップの保存期間
データベースリソース／通信の暗号化（有効or 無効）

© 2018 OGIS-RI Co., Ltd.

4. インストーラを実行



認証画面やAPIのHTTPS設定
APIサービスのサーバ証明書秘密鍵
APIサービスのサーバ中間証明書
APIサービスのサーバ証明書

管理コンソールのHTTPS設定
管理コンソールのサーバ証明書秘密鍵
管理コンソールのサーバ中間証明書
管理コンソールのサーバ証明書

© 2018 OGIS-RI Co., Ltd.

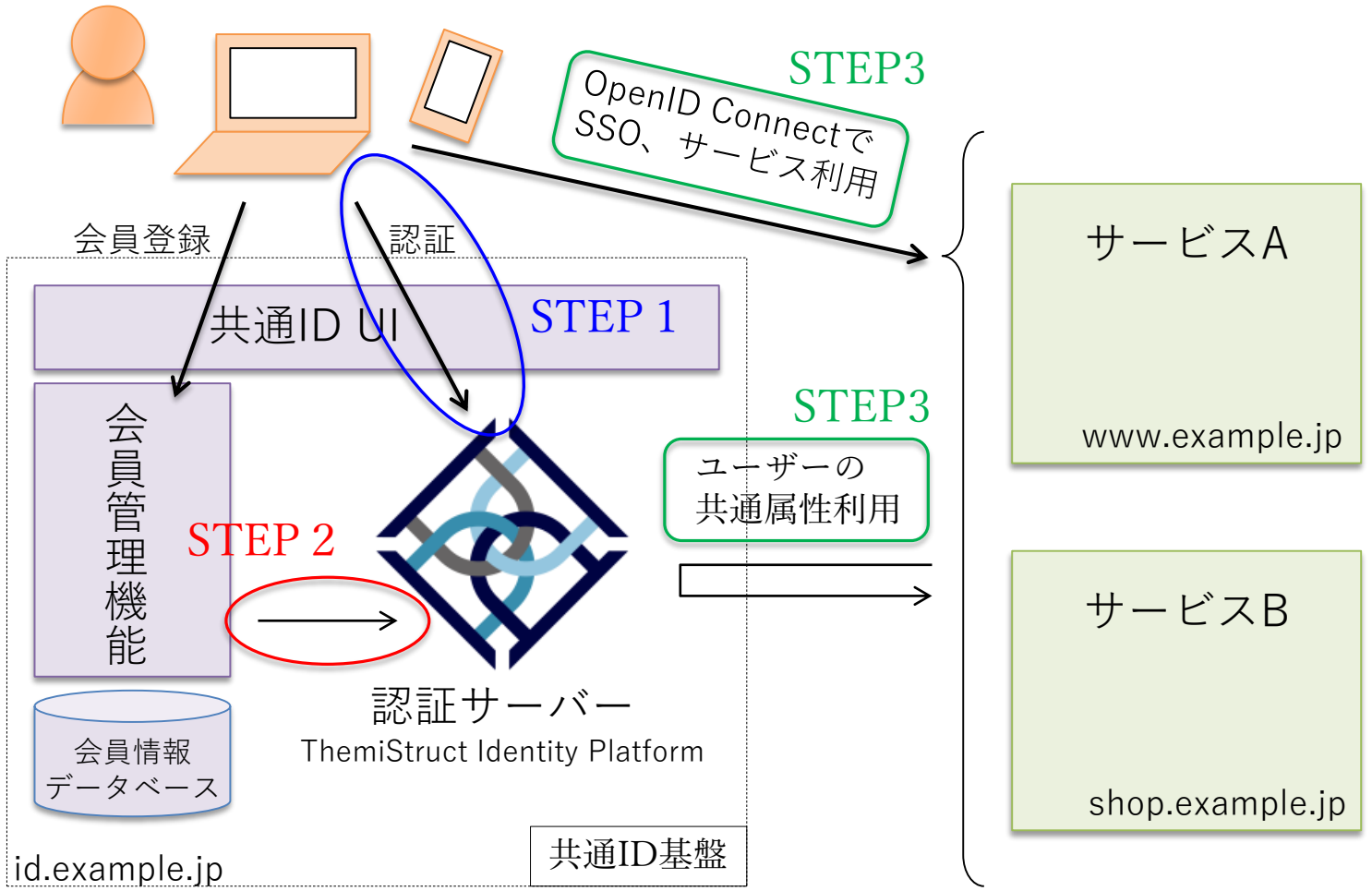
4. インストーラを実行

約80分で作成完了

合計約120分でセットアップ完了

共通ID基盤として構成してみる

共通ID基盤の構成イメージ



構築 STEP 1

✓ サインイン画面にThemiStruct Identity Platform の認証機能を組み込む。

構築 STEP 2

✓ 会員管理機能で登録・更新される情報をThemiStruct Identity Platformに連携する。

構築 STEP 3

✓ サービスサイトと認証サーバでID連携を行い、SSOやユーザの共通属性利用を実現する。

© 2018 OGIS-RI Co., Ltd.

STEP1：サインイン画面に認証機能を組み込む

ThemiStruct Identity Platform が提供するSDKを利用して認証機能を組み込み

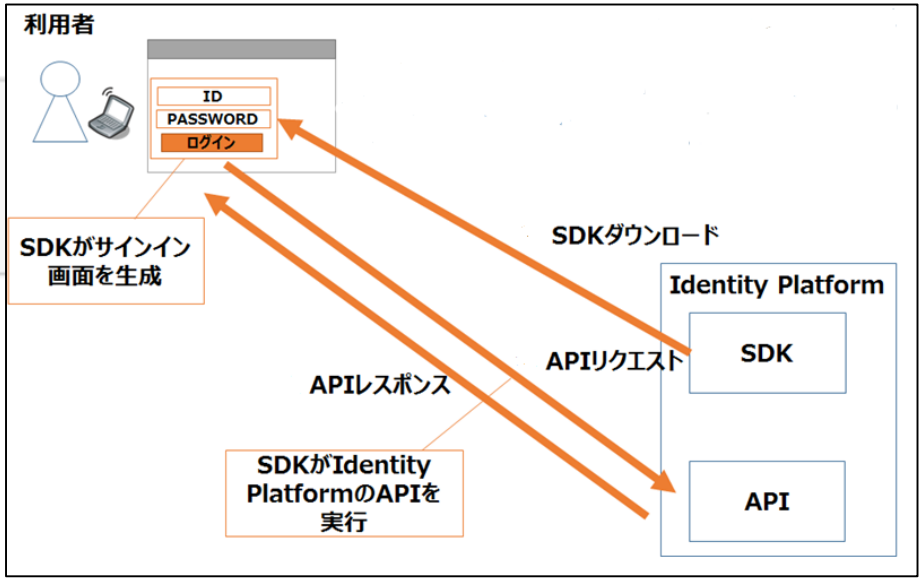
1. サインイン画面のHTMLに認証機能（認証フォーム）を配置したい箇所に以下のタグを追加する

```
<div id="signin-container"></div>
```

2. SDKの読み込みタグを追加する

```
<script
 src="https://${Identity PlatformのFQDN}/public/javascripts/signin.bundle.min.js">
</script>
```

２つのタグを画面に埋め込むことで独自画面に
認証機能を組み込むことが可能！



STEP1：サインイン画面に認証機能を組み込む

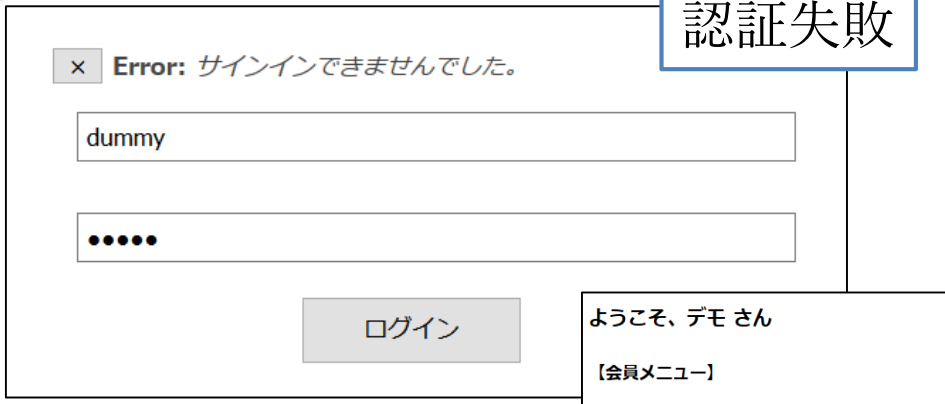
■認証機能を実際に組み込んだ設定と動作画面

```
<h1 id="page_title">会員ログイン（ThemiStruct Identity Platform組込みあり） </h1>
        <div class="type1 noMargin_b">
                <div class="panel">
                        <div class="login overlap">
                        <!--tsIP Login form start -->
                        <div id="signin-container">
                        <script src="public/javascripts/lang.js"></script>
                        <script src="https://x.x.x.x/public/javascripts/signin.bundle.min.js"></script>
                </div>
        </div>
```

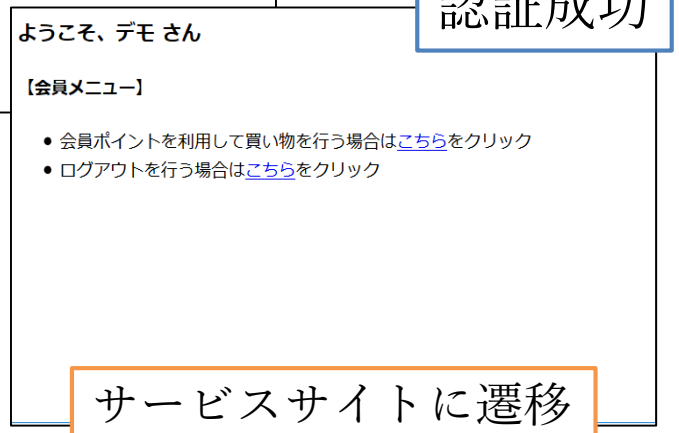
SDKが認証フォームを自動生成

```
<div class="signin-input">
  <input id="userName" class="form-control" data-i18
  <input id="password" class="form-control" name="pa
  <button id="signin-button" class="btn btn-info" da
</div>
```

認証失敗



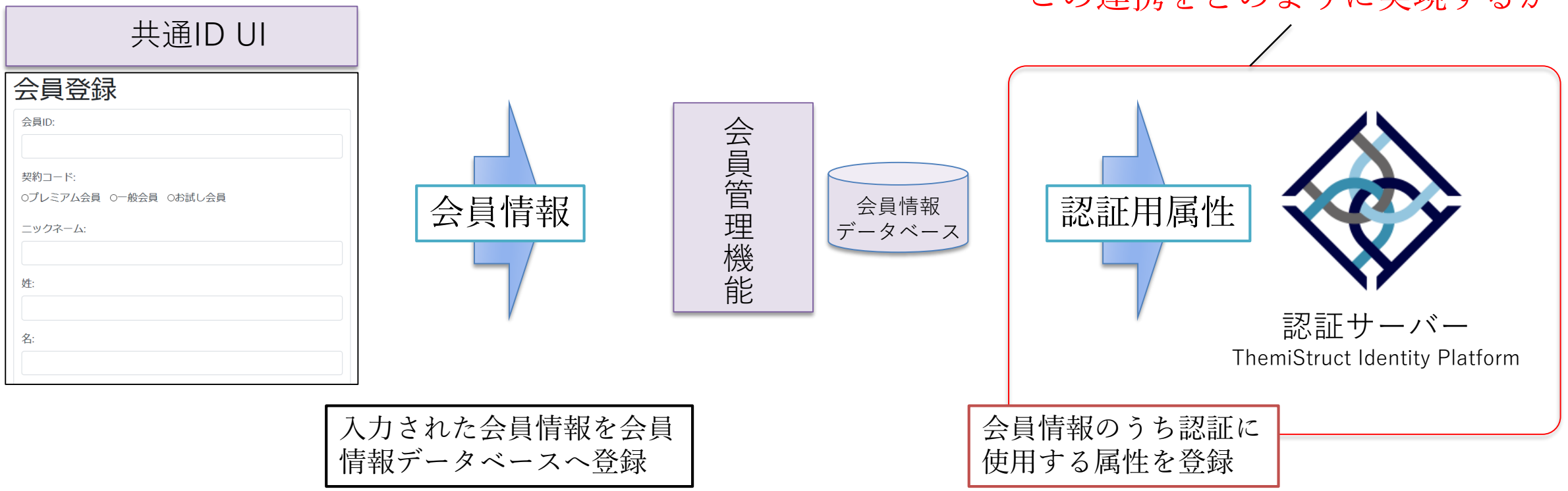
認証成功



サービスサイトに遷移

STEP2：会員管理機能で登録・更新される情報を連携する

会員登録の実現イメージ



STEP2：会員管理機能で登録・更新される情報を連携する

ThemiStruct Identity Platform のアカウント管理APIを利用

■ SCIM（System for Cross-domain Identity Management）に準拠したアカウント管理APIを提供

エンドポイント	メソッド	概要
/account/v2/Users	POST	リクエストパラメータに指定されたユーザ情報でユーザを登録する
/account/v2/Users/{id}	GET	「{id}」で指定されたユーザのユーザ情報を取得する
/account/v2/Users/{id}	PUT	「{id}」で指定されたユーザのユーザ情報を更新する
/account/v2/Users/{id}	DELETE	「{id}」で指定されたユーザを削除する

© 2018 OGIS-RI Co., Ltd.

STEP2：会員管理機能で登録・更新される情報を連携する

アカウント登録（POST /account/v2/Users） のリクエストデータ

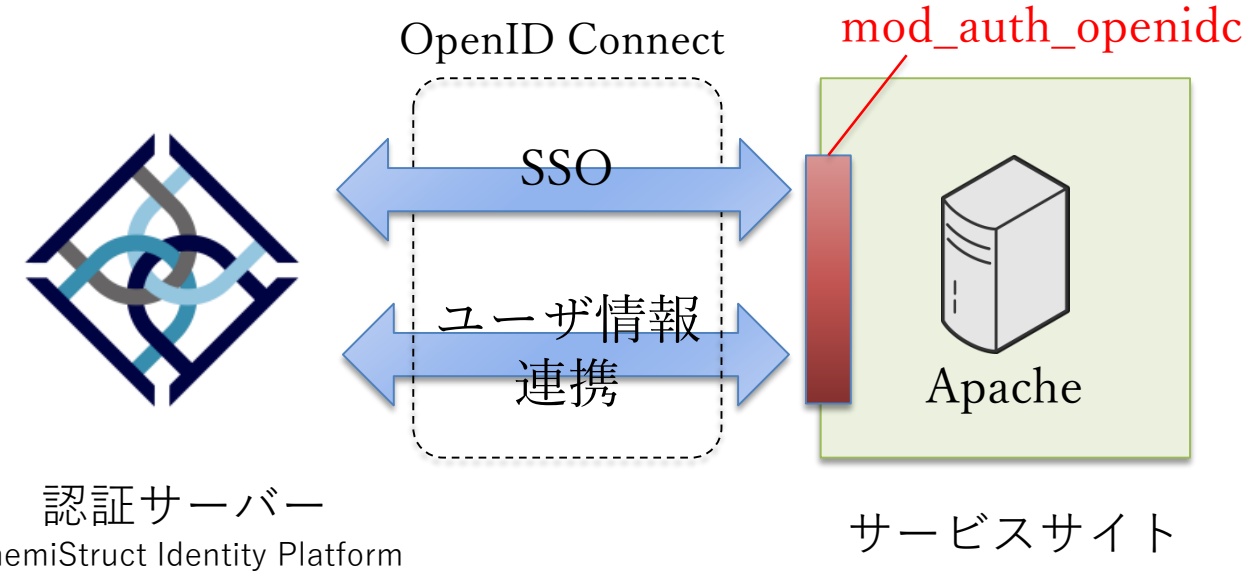
```
{
   "schemas": [
      "urn:ietf:params:scim:schemas:core:2.0:User",
      "urn:custom:scim:schemas:1.0:User",
      "urn:identityplatform:scim:schemas:1.0:User"
   ],
   "userName": “emp0001",
   "nickName": “Tanaka-san",
   "active": true,
   "password": “I0darAVFGV0I5Emzclj",

【続き】
```

```
   "urn:identityplatform:scim:schemas:1.0:User": {
      "jpName": {
         "familyName": "田中",
         "familyNameKana": "タナカ",
         "givenName": "哲也",
         "givenNameKana": "テツヤ"
      }
   },
   "urn:custom:scim:schemas:1.0:User": {
         "employeeType": "A100",
         "organaizationCode": "B2000",
         "positionCode": "C30"
   }
}
```

STEP3：サービスサイトと認証サーバでID連携を行う

サービスサイトにOpenID ConnectでのID連携組み込み



Apacheで構築されたサービスサイトにApacheのモジュール「mod_auth_openidc」を組み込み、ID連携を行う仕組みであるOpenID Connectを利用したID連携を構築してみます

■インストール

```
# yum install –y mod_auth_openidc
```

■インストール後の構築手順

1．ThemiStruct Identity PlatformへID連携を行うクライアントとして登録する
2．mod_auth_openidcの設定を実施する
3．サービスサイト側でmod_auth_openidcから渡されるユーザ情報を受け取る処理を組み込む

STEP3：サービスサイトと認証サーバでID連携を行う

■認証サーバへID連携を行うクライアント（サービスサイト）を登録

登録

ID連携用にこのクライアントに以下のスコープ（権限）を設定
　openid：クライアントがOpenID ConnectによるID連携を可能　※必須
　profile, email, address, phone　※アクセスを許可する範囲に応じて設定（独自定義の追加も可）

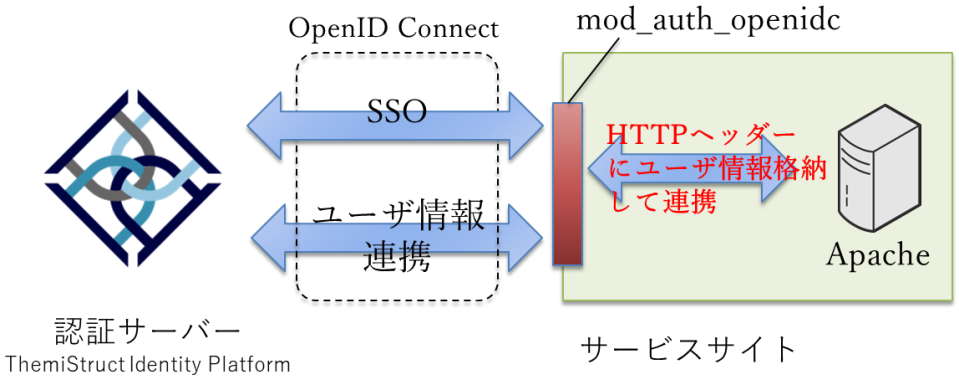
© 2018 OGIS-RI Co., Ltd.

STEP3：サービスサイトと認証サーバでID連携を行う

■mod_auth_openidcの設定

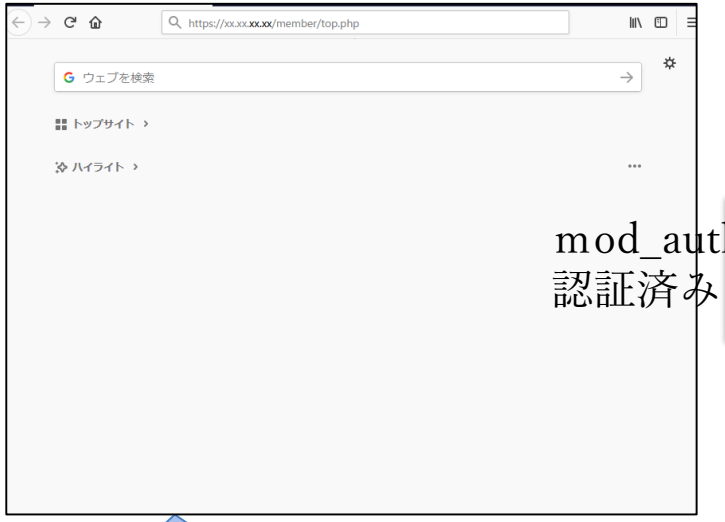
```
# cat /etc/httpd/conf.d/openidc.conf
OIDCProviderMetadataURL  https://＜認証サーバURL＞/oauth/v2/.well-known/openid-configuration
OIDCClientID  ＜払い出されたクライアントID＞
OIDCClientSecret  ＜払い出されたクライアントシークレット＞
OIDCRedirectURI  https://www.example.com:443/callback
OIDCScope "openid profile"
<Location />
  AuthType openid-connect
  Require valid-user
</Location>
```

■サービスサイトコンテンツでユーザ情報連携を組み込み



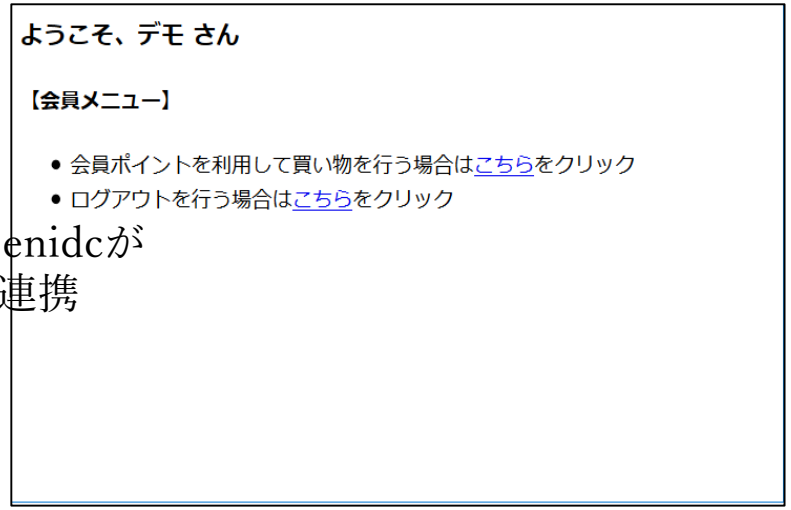

```
# vi /var/www/html/member/top.php
<TITLE>Demo</TITLE>
</HEAD>
<BODY>
<h3>ようこそ、
        <?php print $_SERVER['OIDC_CLAIM_family_name']; ?> さん
</h3>
<h4>【会員メニュー】</h4>
・・・
```

STEP3：サービスサイトと認証サーバでID連携を行う



mod_auth_openidcが
認証済みかチェック

mod_auth_openidcが
ユーザ情報を連携



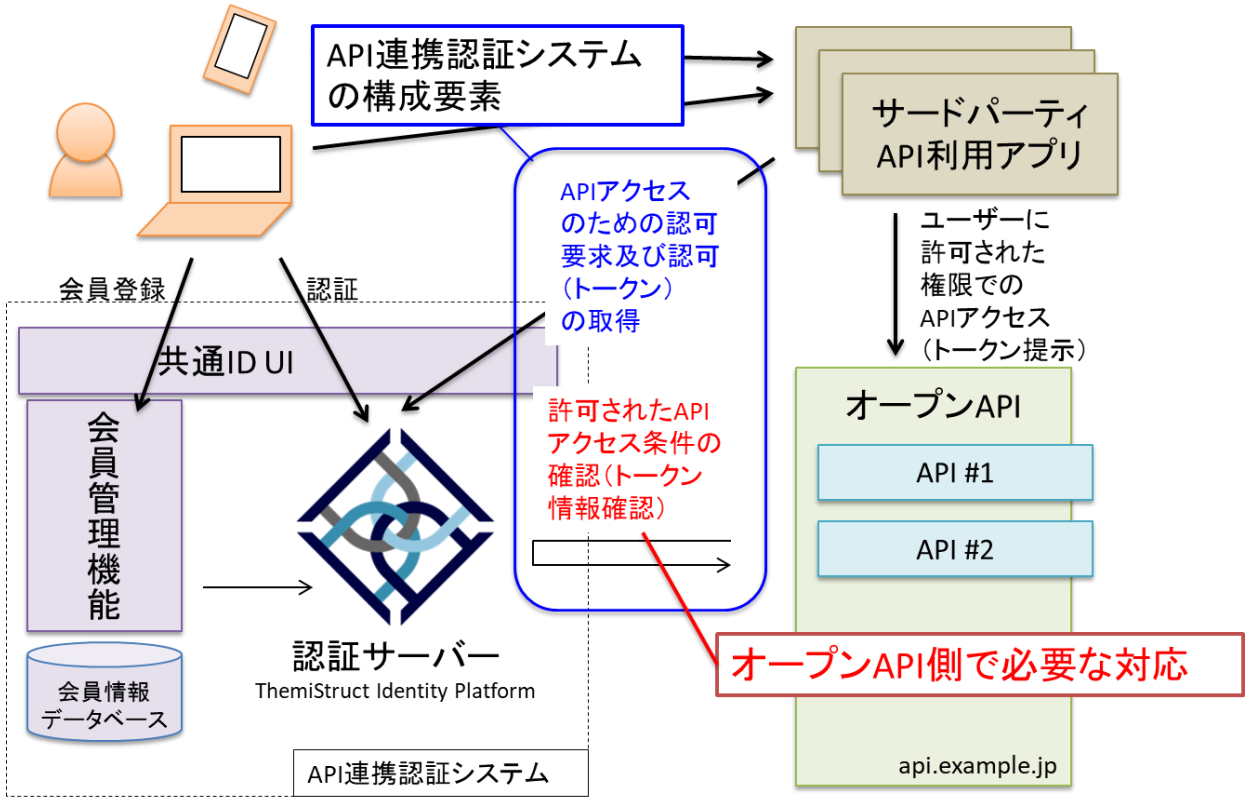
未認証でサービスサイトに
アクセスすると

認証画面が表示され
認証を行うと

サービスサイトに遷移し、
サイト側でヘッダーに格納さ
れたユーザ情報を受け取るこ
とができます

API連携認証システム
として構成してみる！

API連携認証システムの構築イメージ



実現するために必要な対応

■ThemiStruct Identity Platform側の設定
① APIをリソースとして登録
② スコープ（※）の定義
③ API利用アプリをクライアントとして登録
※トークンに付与される権限表す

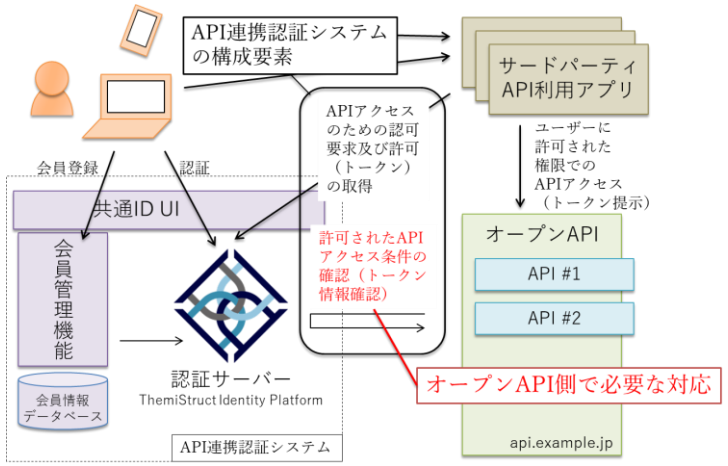
■オープンAPIでの実装
① トークンを受け取る
② トークン情報を取得する
③ 取得したトークン情報を検証する
④ APIの処理／応答を定義する

■API利用アプリでの実装
① 認可要求を行う
② トークンを取得する
③ トークンを付与してAPIを呼び出す

© 2018 OGIS-RI Co., Ltd.

許可されたアクセス条件の確認の実現

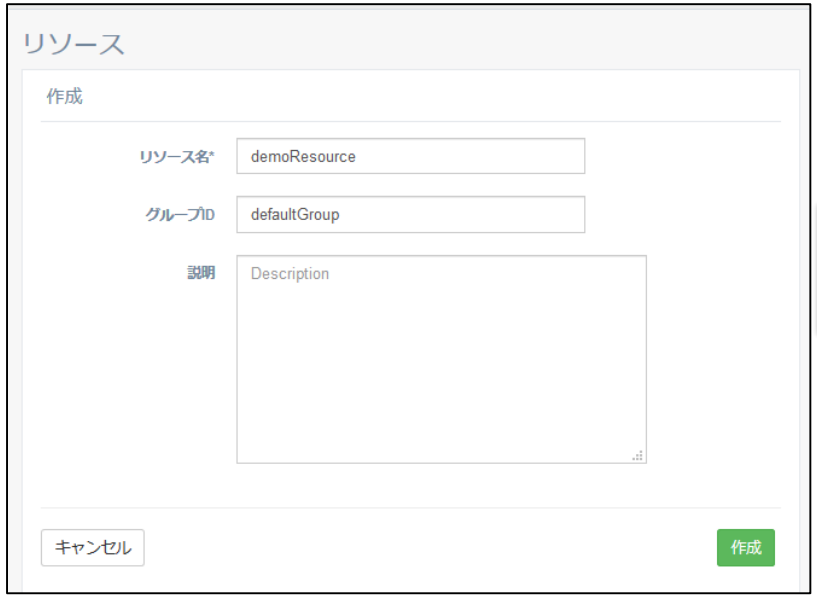
オープンAPIの公開に必要な設定 & ThemiStruct Identity Platform が提供するAPI



■ThemiStruct Identity Platform
① APIをリソースとして登録

■オープンAPI
① トークンを受け取る
② トークン情報を取得する

■APIをリソースとして登録



■トークンの詳細情報取得のエンドポイントを提供

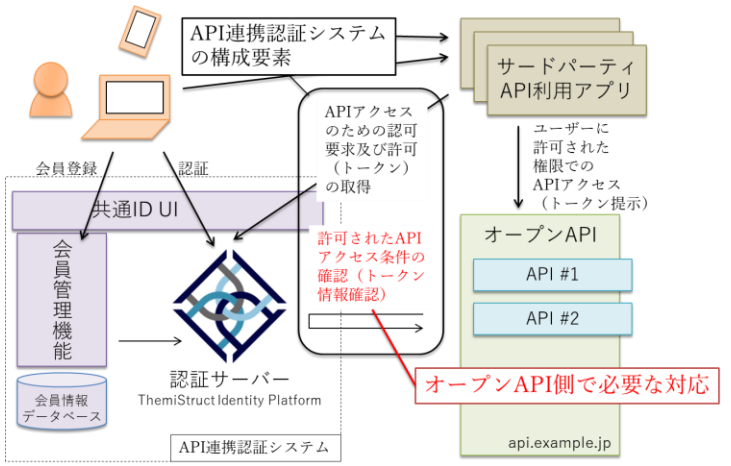
種別	エンドポイント	説明
トークン情報取得	/oauth/v2/token/introspect	提示したトークンの詳細情報を取得します

© 2018 OGIS-RI Co., Ltd.

許可されたアクセス条件の確認の実現

■スコープ（アクセス権限）の登録

独自のスコープを定義することが可能。



■ThemiStruct Identity Platform側の設定

① APIをリソースとして登録

② スコープ（※）の定義

③ API利用アプリをクライアントとして登録

※トークンに付与される権限表す

許可されたアクセス条件の確認の実現

オープンAPI側の実装

```
/* API(1)
 * ポイント照会
 */
router.get('/point', function(req, res, next) {

  var authzHeader = req.get('Authorization');
  if(!authzHeader) {
    ・・・・
  }
  intro.tokenintro(req, clientId, clientSecret).then((ret) => {

    var info = ret;
    if(info.code == 200) {
      if(checkActive(info.active) &&
         checkScope(info.scope, GETPOINT_SCOPE) &&
         checkIss(info.iss, GETPOINT_ISS) ) {
          resource.getpoint(info.sub).then((rows) => {
          res.header('Content-Type', 'application/json; charset=utf-8');
          res.send(rows[0]);
        }).catch((err) => {
          badRequest(res);
          return;
        });
      } else {
        invalidRequest(res);
        return;
      }
```

アクセストークン受け取り（Authorizationヘッダー）

アクセストークンの情報取得

アクセストークンの検証

■オープンAPIでの実装
① トークンを受け取る
② トークン情報を取得する
③ 取得したトークン情報を検証する
④ APIの処理／応答を定義する

■アクセストークン情報

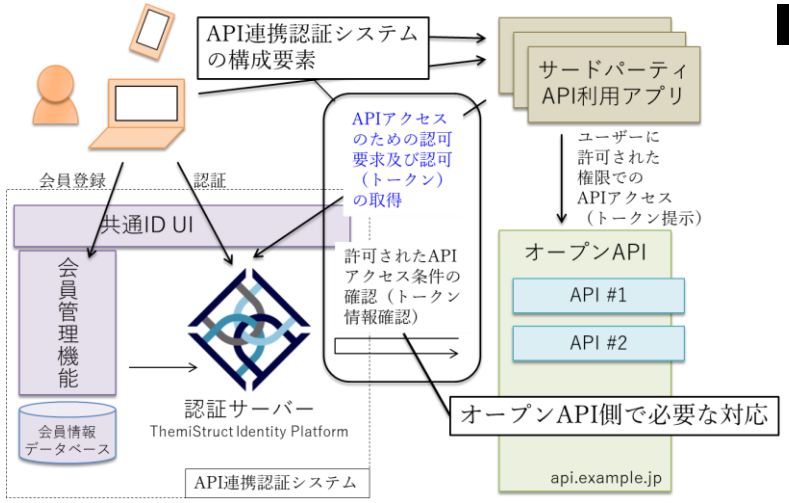
項目名	格納値
active	トークンの有効性
client_id	トークンを要求したクライアント
scope	トークンが許可するアクセス範囲を表すScopeのリスト
sub	ユーザの識別子
iss	トークン発行元
exp	トークンの有効期限
iat	トークンの発行日時

※上記はJSON形式で返却される

© 2018 OGIS-RI Co., Ltd.

APIアクセスのための認証要求の実現

ThemiStruct Identity Platformで提供されるエンドポイント＆導入に必要な設定



■認証要求を許可（トークンを発行）するクライアントを登録

設定項目例
スコープ（許容するアクセス範囲）
リフレッシュトークン有無
発行されたトークンの有効期限

■ThemiStruct Identity Platform
③ API利用アプリをクライアントとして登録

■API利用アプリでの実装
① 認可要求を行う
② トークンを取得する

■認証要求及びトークン取得のエンドポイントを提供

No	エンドポイント	説明
APIアクセスのための認証要求	/oauth/v2/authz	リクエストパラメータを基に必要な認証の要求を行います。未認証の場合は認証画面を表示します。 認証成功すると登録されたコールバック先に認可コードやトークンを送信する。
	/oauth/v2/token	認証成功後に受け取った認可コード等からアクセストークンを取得します。

© 2018 OGIS-RI Co., Ltd.

APIアクセスのための認証要求の実現

API利用アプリ側の実装

Google OAuth2 API Client Library for Java利用時の実装例

■認可要求の実装例

```
@RequestMapping("/xxxxx")
public String xxxxx(Model model,Principal principal,
    HttpServletRequest request) throws ServletException {

    AuthorizationCodeRequestUrl codeUrl
        = new AuthorizationCodeRequestUrl(AUTHORIZATION_SERVER_URL, CLIENT_ID);
    codeUrl.setScopes(Arrays.asList(SCOPE_REF));
    codeUrl.setResponseTypes(Arrays.asList("code"));
    codeUrl.setRedirectUri(REDIRECT_URL);
    codeUrl.setState(createState(request));
    codeUrl.set(NONCE,createNonce());

    return "redirect:" + codeUrl.build();
}
```

■認証成功後のコールバック実装例

```
public static Credential oauth2Callback(HttpServletRequest request) {

    StringBuffer fullUrlBuf = request.getRequestURL();
    ・・・
    AuthorizationCodeResponseUrl authResponse
        = new AuthorizationCodeResponseUrl(fullUrlBuf.toString());
    ・・・
        // 認可コードを取得
        code = authResponse.getCode();
    }
    ・・・
    try {
        // 認可コードからアクセストークン取得
        cre = requestAccessToken(code);
    ・・・
```

■認可コードからアクセストークンを取得する実装例

```
private static Credential requestAccessToken(String code) throws IOException {
    TokenResponse response = null;
    try {
        response = new AuthorizationCodeTokenRequest(new NetHttpTransport(), new JacksonFactory(),
            new GenericUrl(TOKEN_SERVER_URL), code)
                .setRedirectUri(REDIRECT_URL)
                .setClientAuthentication(new BasicAuthentication(CLIENT_ID, CLIENT_SECRET))
                .execute();
```

■API利用アプリでの実装

① 認可要求を行う
② トークンを取得する
③ トークンを付与してAPIを呼び出す

© 2018 OGIS-RI Co., Ltd.

本ご紹介のまとめ

本ご紹介のまとめ

ロ ThemiStruct Identity Platformのセットアップ

- AWSマネージドサービスを利用したシステムを4つのステップで構築
- セットアップはCloudFormationやインストーラーを利用して自動化
- 上記により高可用・スケーラブルな環境を約2時間で構築可能

ロ 共通ID基盤として構成する

- サービスサイトへの認証機能の組み込みを簡略化するためのSDKを提供
- アカウント情報の連携を可能とするアカウント管理APIを提供
- ID情報連携をOpenID Connectや対応ミドルウェア等を利用して実現することが可能

ロ API連携認証システムとして構成する

- OAuth2.0に準拠した認証要求やアクセス条件確認を行うためのエンドポイントを提供
- 各種エンドポイントを利用するためのクライアントやスコープ設定を管理GUIから設定可能
- オープンAPIの公開でAPI側はトークンの受け取り／情報取得／検証の3ステップを組み込み

© 2018 OGIS-RI Co., Ltd.

ご清聴ありがとうございました

【お問い合わせ先】
株式会社オージス総研
TEL: 03-6712-1201 / 06-6871-8054
mail: info@ogis-ri.co.jp

© 2018 OGIS-RI Co., Ltd.