ローコード開発基盤

OutSystems で

シングルサインオンを実現する

tdi 情報技術開発株式会社

複数のシステム・サービスを利用する際に、一度の認証処理で、その認証に紐づけられている複数システム・サービスが利用可能になるシングルサインオン（以下、SSO）は、ユーザの利便性の向上や情報漏洩リスクの低減の観点から多くの企業で取り入れられてきました。近年では、Office365・Google WorkSpace・Dropbox など外部サービスの利用範囲の拡大や外部ストレージの利用に伴い、SSO の対応範囲が拡大しています。

本書では、企業システムのローコード開発基盤として注目されている OutSystems で SSO を実装する方法をご紹介いたします。既に OutSystems をご導入いただいているお客様、今後導入をご検討されているお客様のご参考になれば幸いです。

内容

SSO のメリット・デメリット

OutSystems における SSO の実装方法をご紹介するにあたり、SSO のメリット・デメリットについて記載いたします。

SSO は、一度の認証処理で複数のシステム・サービスにログインできることから、システムごとに ID・パスワードを覚えておく必要がなく、ユーザの利便性向上に役立つというメリットがあります。しかし、SSO で利用する ID・パスワードが流出すると、SSO で利用できる全てのシステムにアクセスできてしまうため、セキュリティリスク面でのデメリットがあります。また、SSO に対応できない全てのシステムやサービスが存在するケースもあり、SSO の導入、対象システム・サービスの選定には慎重な判断が必要です。

SSO を実現するための実装方式

SSO を実現するための代表的な 4 つの方式を紹介します。

1. リバースプロキシ方式　：
   リバースプロキシ経由で SSO 対象のシステムにアクセスし、認証を行う方式
2. エージェント方式　：
   Web アプリケーションサーバーに SSO を実行するための認証機能を導入する方式
3. 代行認証方式　：
   クライアント PC に専用のエージェントを導入し、そのエージェントが SSO 対象システムのログイン画面を監視しユーザの代理で ID・パスワードを打ち込む方式
4. SAML 認証方式　：
   クラウドサービスをつかった SSO の実装に使う仕組み
   SAML（Security Assertion Markup Language）とは、異なるインターネットドメイン間でユーザ認証を行うための標準規格

OutSystems では、SAML 認証方式で SSO を実装します。SAML 認証方式は、「Identity Provider（IdP）」と「Service Provider（SP）」で構成されており、各サービスを提供するインターネットドメイン間でユーザ認証を行なうための方式です。IdP は、認証情報を提供する側であり、SP へログインしたユーザが登録されたユーザであるかを確認し、認証結果を SP へ送信する役割を担います。SP は、認証情報を利用する側であり、SAML 認証で行われる IdP と SP 間の認証の過程は以下のようになります。

① ユーザが SP にアクセスを行います。
② ログインしていない場合、SP はユーザからのアクセス要求を受け取ることで、IdP に対して認証要求(SAML)を生成し、IdP にリダイレクトします。
③ SP からの認証要求を受けることで、IdP は認証処理を実施します。
④ IdP にてユーザ認証が成功すると、IdP は SP に対して認証応答(SAML)を生成し SP へリダイレクトします。
⑤ IdP からの認証応答を受け取り、SP はクラウドサービスへの自動ログインを実施します。
⑥ ログインに成功することで、ユーザにはクラウドサービスの画面が表示されます。

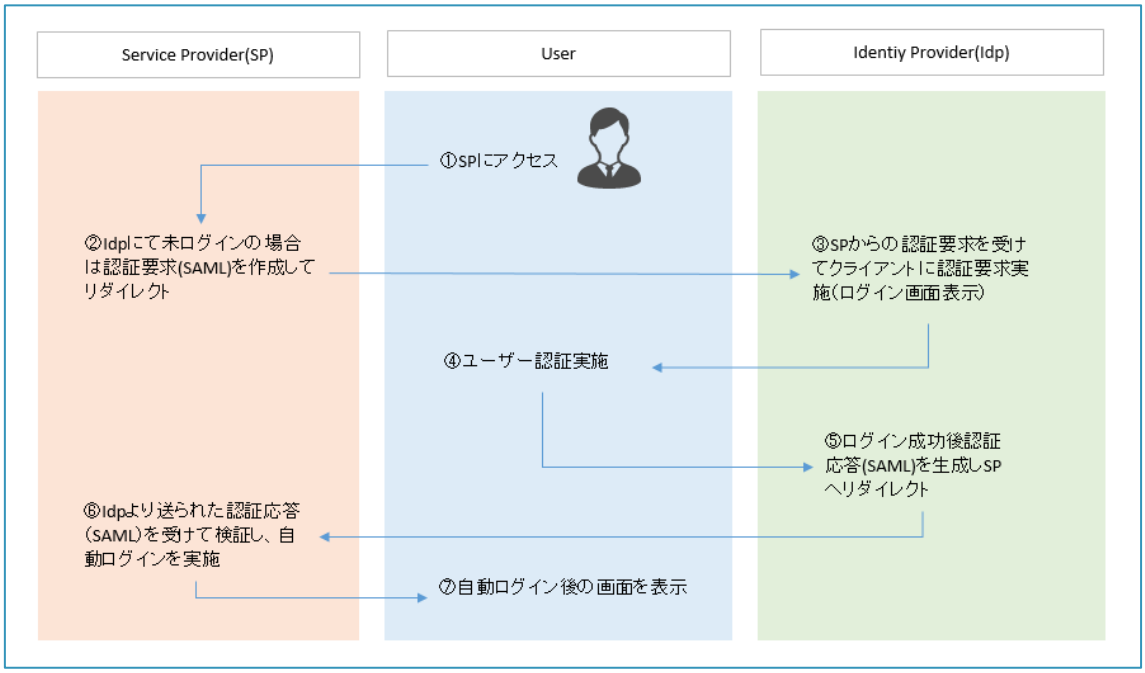

図 1　SAML 認証フロー

OutSystems で SAML 認証方式が便利な理由

OutSystems には、OutSystems が提供している共通ライブラリやパーツ、テンプレートスタイルなどを集めた「Forge」という Web サイトがあります。「Forge」では SAML 認証方式の構成に必要な部品が公開されており、簡単に SAML 認証方式の SSO を実装することができます。

「Forge」で公開されている SAML 認証方式の SSO を実現するためのパーツ（図 2 参照）を導入することで、商用 ID プロバイダが提供している認証機能を利用して、OutSystems の

アプリケーションの認証ができます。Forge には図 2 のパーツ以外にも SSO を実現するための複数のコネクタが公開されており、これらの Forge の部品を利用することで OutSystems のアプリで様々な外部の SSO 機能と連携することが可能となっています。

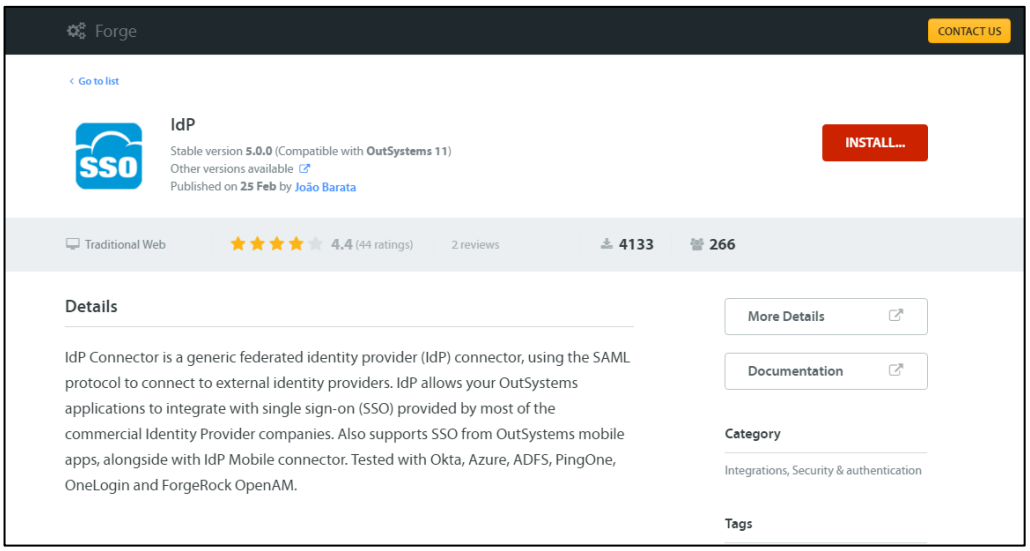

図 2　OutSystems「Forge」サイトで公開されている SAML 認証方式の SSO 用パーツ

SAML 認証方式を使用した SSO の実装例

OutSystems で SAML 認証方式を使用した SSO の実装例として、Azure AD 認証と連携するための実装方法を紹介します。

Azure AD 認証は、Microsoft が提供している「Azure Active Directory」を利用して SSO を実現する認証方式です。Azure AD は、Windows や Office などの Microsoft サービスとの相性がよいため、Microsoft サービスに対して SSO を導入する際に採用されるケースが多くあります。Microsoft 365 などの Microsoft サービスの有料サブスクリプションを保有している場合、Azure AD が付属するため無料で利用することができます。

手順 0.事前準備

実装に入る前に以下の準備をしていきます。

・Azure Active Directory の Web サイトから Azure の無料アカウントを発行する。（こちらから）

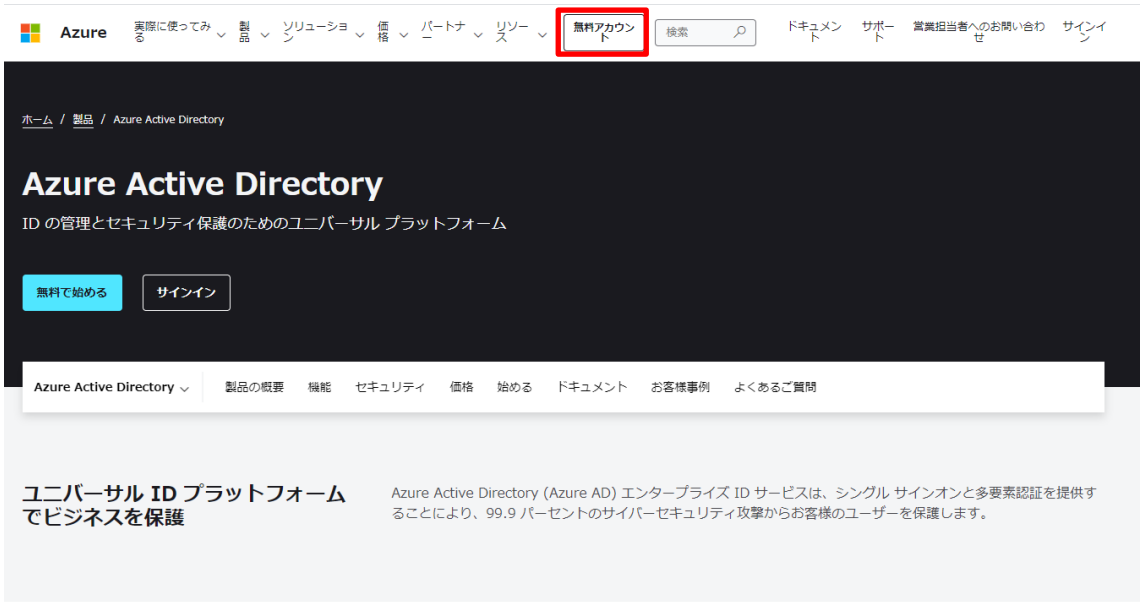

図 3　Azure Active Directory の公式サイト

・アプリケーションで利用する AzureAD ユーザを作成する。（こちらを参考）
・SSO を実装する仮のアプリケーションを TraditionalWeb アプリケーションで作成する。（OutSystems 標準のログインフローを使っていれば OK）

手順 1.IdP をアプリケーションに連携する

SAML 2.0 プロトコルを使用した SSO を実装するために、Forge「IdP」が必要です。Forge サイトからダウンロードします。(こちらから)

次に、「IdP」が持つログイン処理用のサーバーアクションを使って、アプリケーションのログインフローを書き換えていきます。

・Common フローの NoPermission 画面-Preparation を書き換える

OutSystems のアプリケーションでは、認証が必要なリソースに未認証のユーザがアクセスした場合、例外処理が行われます。例外処理では、NoPermission 画面に遷移し、画面の Preparation 内で未認証のユーザと判断され、ログイン画面に遷移するフローになっています。

SAML を利用した SSO を実装する場合、上記のフローで標準のログイン画面ではなく ID プロバイダにリダイレクトさせたいので、以下の画像のように Forge「IdP」部品内の ServerAction である「IdP_SSO_URL」を用いて処理を書き換えていきます。

「IdP_SSO_URL」の出力で Common¥ExternalURL を呼び出します。

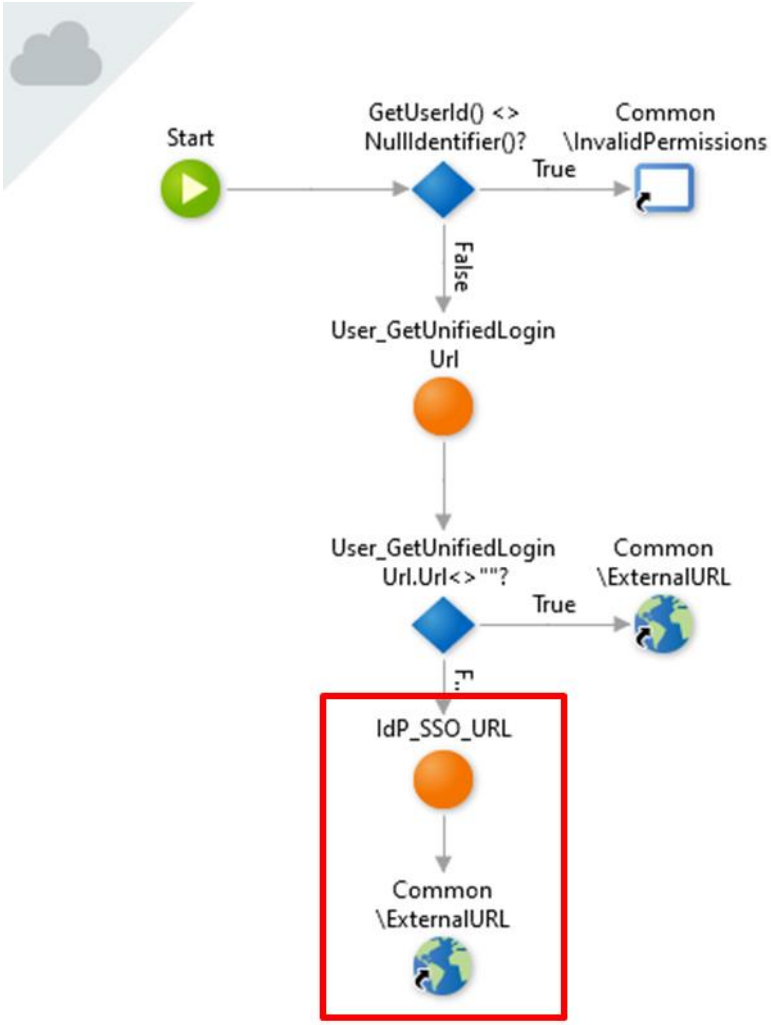

図４　Service Studio （Forge「IdP」_デフォルトのロジック全体図）

・LoginInfoWeb ブロックの LogOut アクションを書き換える

標準仕様では、ログアウトした際は上記のアクションから OutSystems のログインユーザアカウントのログアウトを行うフローになっています。SAML を利用した SSO を実装する場合、上記のフローで OutSystems のログインユーザーアカウントだけでなく、AzureAD 側のアカウントからもログアウトを実行させたいので、以下の画像のように Forge「IdP」部品内の ServerAction である「IdP_SingleLogout_URL」を用いて処理を書き換えていきます。

「IdP_SingleLogout_URL」の出力で Common¥ExternalURL を呼び出します。

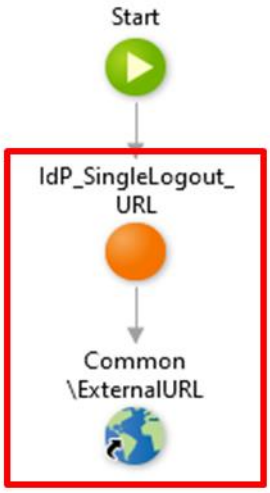

図 5　Service Studio （Forge「IdP」_LogOut アクションの編集）

手順 2. AzureAD 側の設定

AzureAD ポータルから、プロバイダ側の設定をしていきます。

1.OutSystems Azure AD アプリケーションをギャラリーに追加

画面左側メニューより[エンタープライズ アプリケーション]に移動

図 6　Azure Active Directory Portal

↓

表上部の[新しいアプリケーション]をクリック

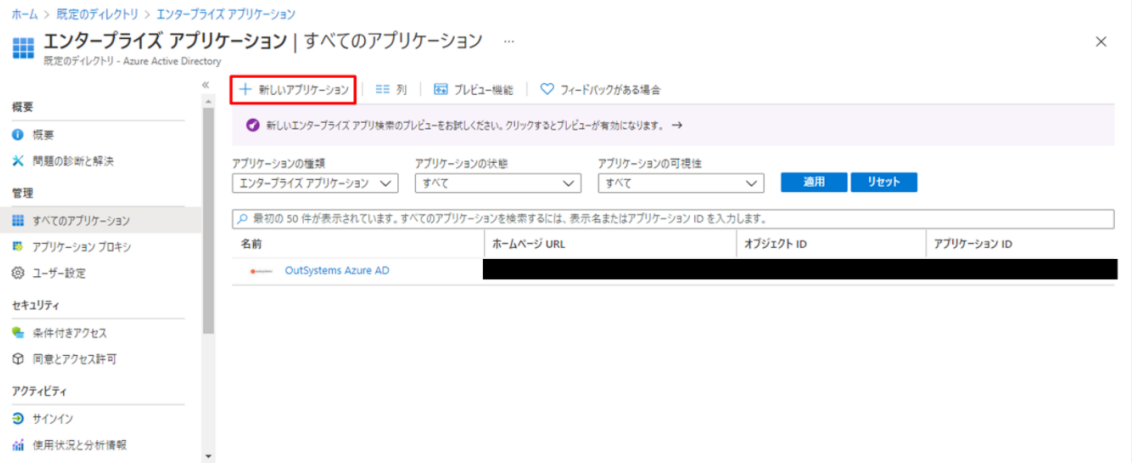

図 7　Azure Active Directory Portal（アプリケーションの追加①）

↓

OutSystems Azure AD で検索

アプリケーションを見つけたら選択し、[作成]をクリック

図 8　Azure Active Directory Portal（アプリケーションの追加②）

2.SAML による SSO の設定

OutSystems Azure AD アプリケーションダッシュボードの画面左側メニューから[シングルサインオン]に移動

↓

[SAML]をクリック

↓

設定項目「①基本的な SAML 構成」 の編集画面に移動

以下の項目を設定する。

・識別子 (エンティティ ID)

https://[環境情報]/IdP

・応答 URL (Assertion Consumer Service URL)

https://[環境情報]/IdP/SSO.aspx

図 9　Azure Active Directory Portal（SAML 構成）

↓

設定項目「③SAML 署名証明書」 より [フェデレーションメタデータ XML] 横の [ダウンロード] をクリック。ファイルをダウンロードする。

図 10　Azure Active Directory Portal（SAML 構成_メタデータのダウンロード）

3.AzureAD ユーザを OutSystemsAzureAD アプリケーションに追加

AzureAD の登録ユーザを OutSystemsAzureAD アプリケーションで利用できるように追加していきます。

OutSystems Azure AD 画面 左側メニューより[ユーザとグループ]に移動

↓

画面上部[ユーザまたはグループの追加]をクリック

図 11　Azure Active Directory Portal（ユーザ追加）

↓

ユーザ：[選択されていません]よりサブメニューを開く

↓

アプリケーションで利用するユーザを選び、[選択]をクリック

↓

画面下部[割り当て]をクリック

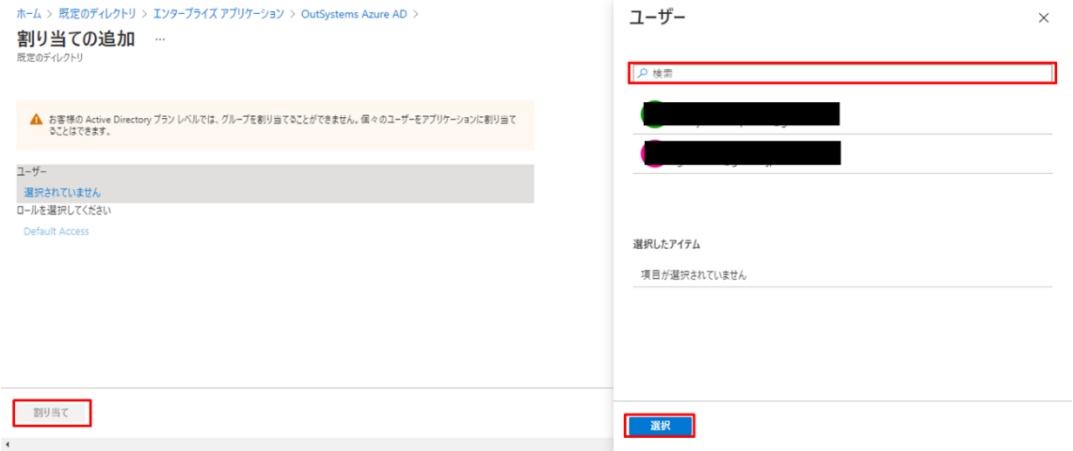

図 12　Azure Active Directory Portal（ユーザ追加_アプリケーション割り当て）

※ユーザのプロビジョニング設定を変更

上記の処理では、AzureAD で認証が許可された後、OutSystems 側で Microsoft ユーザに対応するユーザを自動生成するようになっています。その際、初期設定では User エンティティ - External_Id アトリビュート に AzureAD のプリンシパル名が対応して作成されます。この External_Id アトリビュートには 36 字以内の text 型データを指定する必要がありますが、Azure の標準のプリンシパル名がこの文字数を超えることがあります。その場合、SQL 書き込みエラーが発生し、正しく SSO の処理を終えることが出来ません。

AzureAD ポータルからプリンシパル名を変更する、または External_Id に対応する AzureAD 側属性を変更する、等の回避策を実施する必要があります。

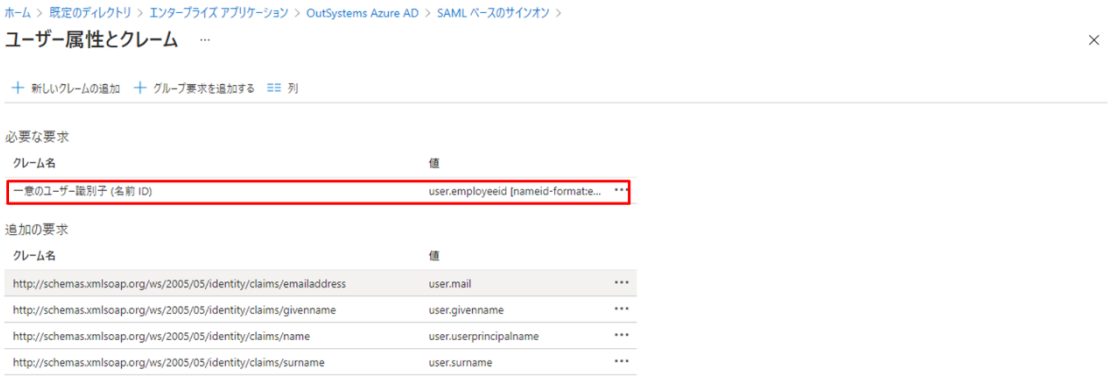

図 13　Azure Active Directory Portal（ユーザ追加_プロビジョニング設定の変更）
（今回は仮にプリンシパル名の代わりに従業員 ID を External_Id の対応属性に利用する設定とした）

手順 3.IdP の接続情報の設定

手順 2 で AzureAD の準備が完了したので、その情報を Forge「IdP」側に設定していきます。

1.Users アプリケーションより、自身のユーザに「IdP_Administrator」ロールを付与

図 14　Users アプリケーション（ロールの付与）（Idp 画面に入るための専用ロール）

2.Forge「IdP」の Web 画面から、手順 2 で作成したプロバイダ情報を設定

Web 画面は、ServiceStudio で Forge「IdP」アプリケーションを開き、「Open in Browser」よりアクセスできます。

[Add SAML App]をクリック

↓

以下の項目を設定し、[Save]をクリック

Provider - [AzureAD/ADFS]

AppName- 任意（IdP の一覧表示でのみ利用する）

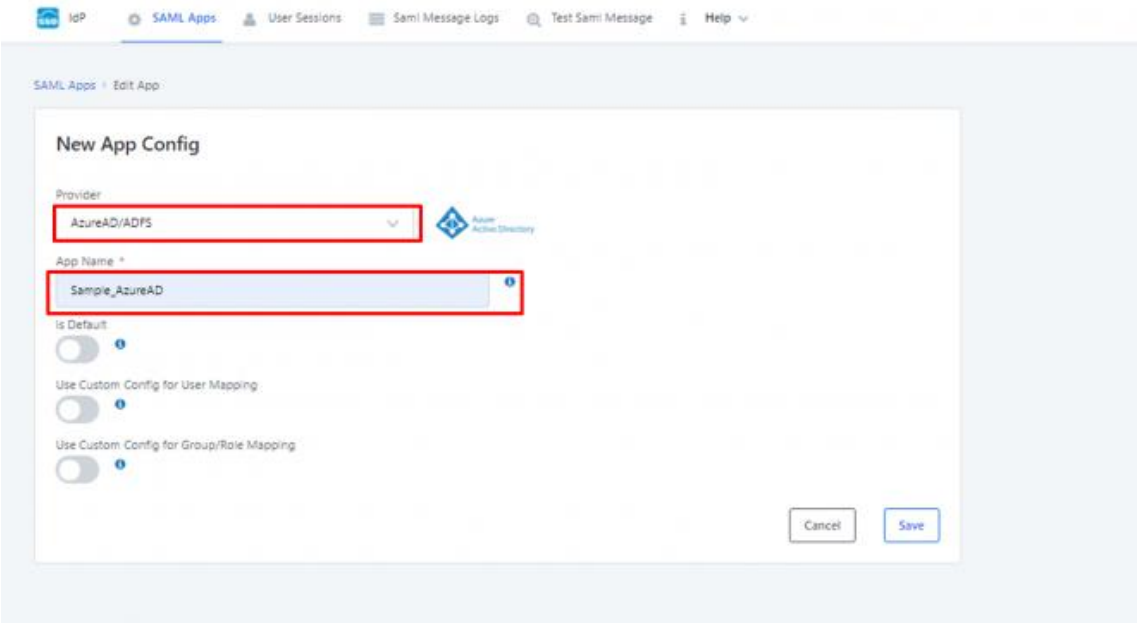

図 15　Idp アプリケーション（プロバイダ情報の設定）

↓

一覧画面から作成されたアイコンをクリック

↓

タブを[Idp Server and Settings]に切り替える

[Import from IdP/federation metadata xml]をクリック

↓

手順 2 でダウンロードしたメタデータファイルをアップロード

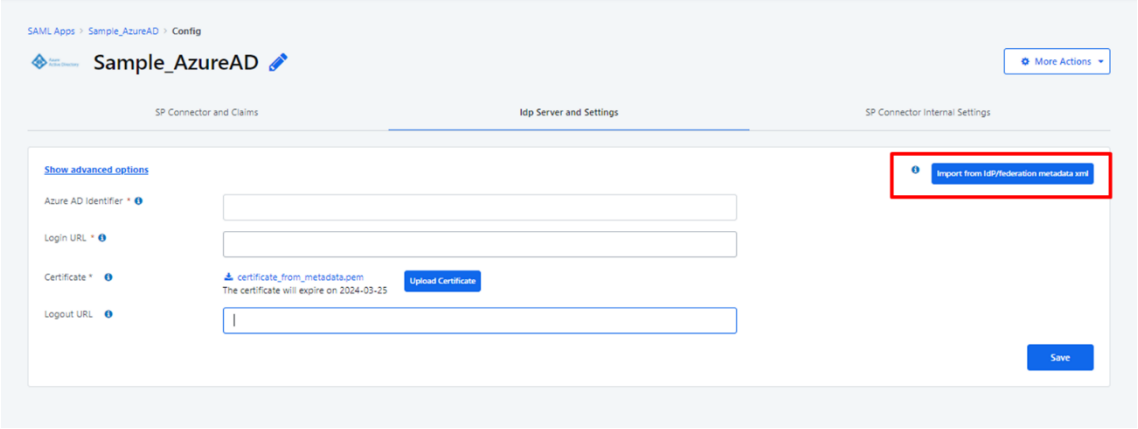

図 16　Idp アプリケーション（メタデータファイルのアップロード）

完成したアプリケーションを確認

以上で SSO の実装が完了しました。実装先の OutSystems アプリケーションをブラウザで開いてみます。

アプリケーションが起動したタイミングで AzureAD の認証画面に遷移するはずです。

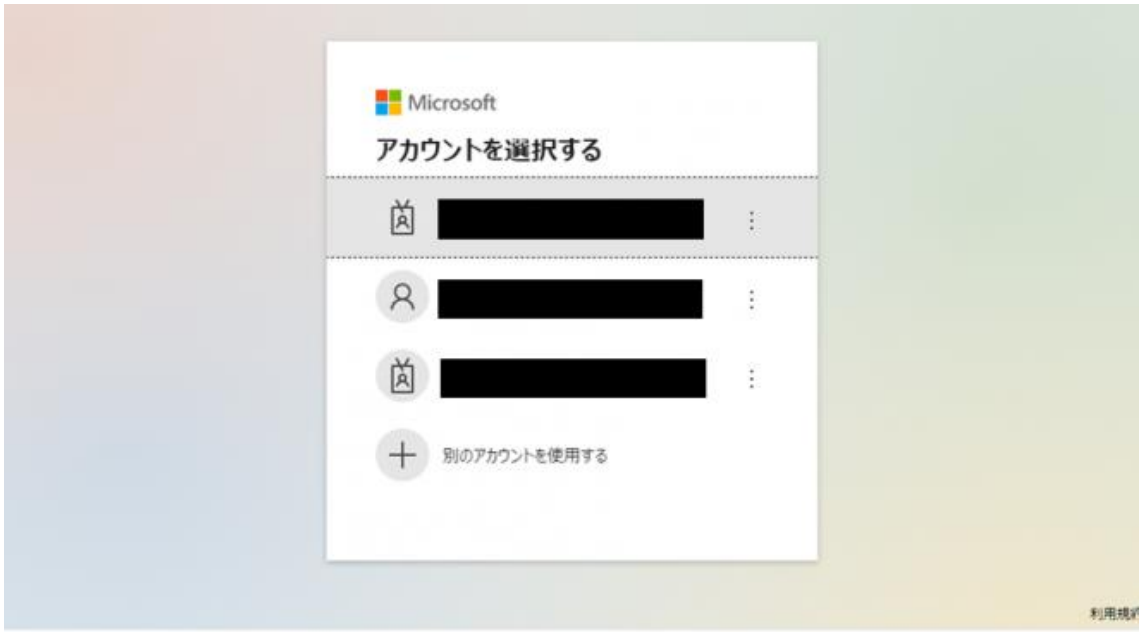

図 17　SSO の実装を組みこんだアプリケーション

認証後、対象のアプリケーション画面にリダイレクトすれば成功です。同様に、ログアウトした際の挙動も確認してみましょう。

今回は AzureAD を利用した SSO 機能を実装してみました。今回利用した Forge「IdP」は Okta, One Login など、他の SSO サービスにも対応しており、幅広い実装が可能です。

Forge を利用することで、OutSystems 標準機能の範囲だけでなく、他サービスと連携した機能の実装も幅広くカバーできます。

終わりに

tdi グループは、 OutSystems の機能や技術について十分な知識を持った多くの技術者を有しており、資格保有者数は国内トップクラスです。ローコード開発が一般的に注目される以前（2016 年）から重ねた OutSystems 開発の実績をもとに、IT 戦略コンサルタントや OutSystems 導入から運用までをトータルサポートします。また、お客様に合わせた人財育成や内製化もご支援いたします。

お困りのご担当者の方は、どうぞお気軽にお問合せください。

【ローコード開発基盤「OutSystems」】

https://www.tdi.co.jp/outsystems/

【お問い合わせ】

https://tdi.smktg.jp/public/application/add/1095

情報技術開発株式会社 営業本部

東京:〒163-1332 東京都新宿区西新宿六丁目 5 番 1 号 新宿アイランドタワー32 階
TEL.03-5325-4811(代表) FAX.03-5325-4812

中部:〒451-6027 愛知県名古屋市西区牛島町 6 番 1 号 名古屋ルーセントタワー27 階
TEL.052-571-6871(代表) FAX.052-571-3856

関西:〒530-0005 大阪府大阪市北区中之島二丁目 2 番 7 号 中之島セントラルタワー20 階
TEL.06-6201-7739(代表) FAX.06-6201-7740

九州:〒812-0013 福岡県福岡市博多区博多駅東二丁目 10 番 1 号 福岡ビル S 館 7 階
TEL.092-451-8218(代表) FAX.092-474-7379