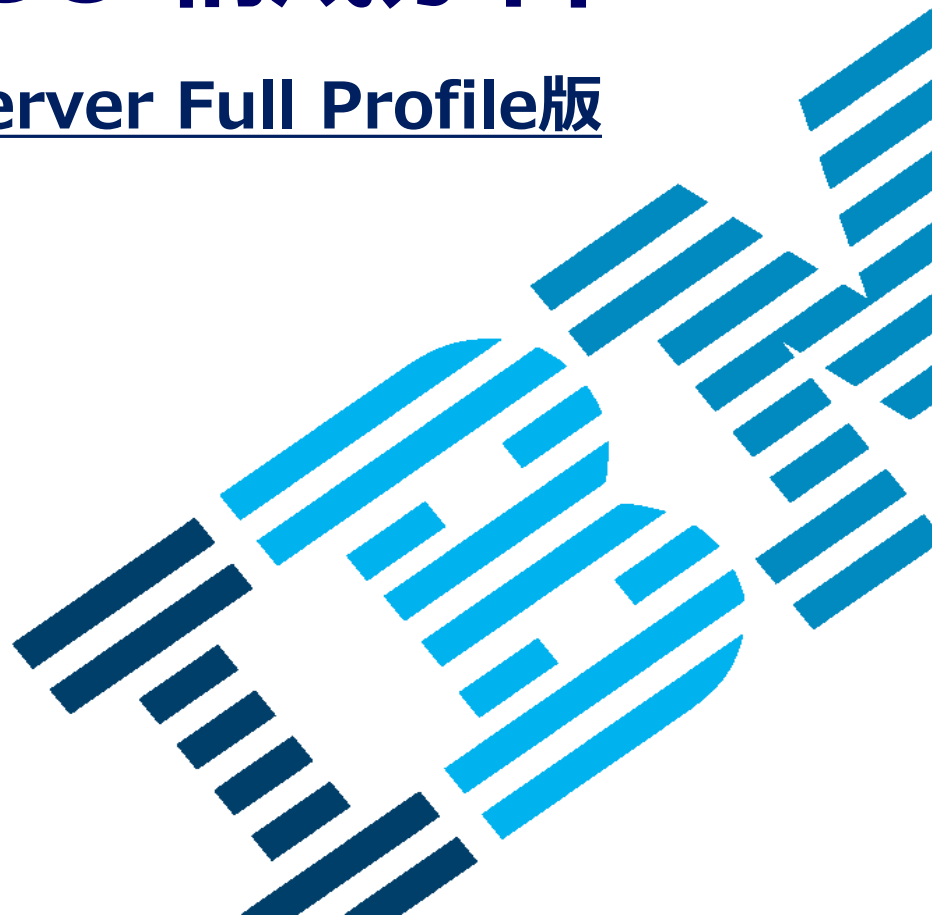
WAS V8.5.5 SAML Web SSO 構成ガイド

WebSphere Application Server Full Profile版

日本アイ・ビー・エム システムズ・エンジニアリング株式会社



Disclaimer

- この資料は日本アイ・ビー・エム株式会社ならびに日本アイ・ビー・エム　システムズ・エンジニアリング株式会社の正式なレビューを受けておりません。
- 当資料は、資料内で説明されている製品の仕様を保証するものではありません。
- 資料の内容には正確を期するよう注意しておりますが、この資料の内容は2015年12月現在の情報であり、製品の新しいリリース、PTFなどによって動作、仕様が変わる可能性があるのでご注意下さい。
- 今後国内で提供されるリリース情報は、対応する発表レターなどでご確認ください。
- ＩＢＭ、ＩＢＭロゴおよびibm.comは、世界の多くの国で登録されたInternational Business Machines Corporationの商標です。他の製品名およびサービス名等は、それぞれＩＢＭまたは各社の商標である場合があります。現時点でのＩＢＭの商標リストについては、www.ibm.com/legal/copytrade.shtmlをご覧ください。
- 当資料をコピー等で複製することは、日本アイ・ビー・エム株式会社ならびに日本アイ・ビー・エム　システムズ・エンジニアリング株式会社の承諾なしではできません。
- 当資料に記載された製品名または会社名はそれぞれの各社の商標または登録商標です。
- JavaおよびすべてのJava関連の商標およびロゴは Oracleやその関連会社の米国およびその他の国における商標または登録商標です。
- Microsoft, Windows および Windowsロゴは、Microsoft Corporationの米国およびその他の国における商標です。
- Linuxは、Linus Torvaldsの米国およびその他の国における登録商標です。
- UNIXはThe Open Groupの米国およびその他の国における登録商標です。

目次

1. SAML Webシングルサインオンの概要
2. WAS SAML Service Provider構成方法
3. Hint&Tips

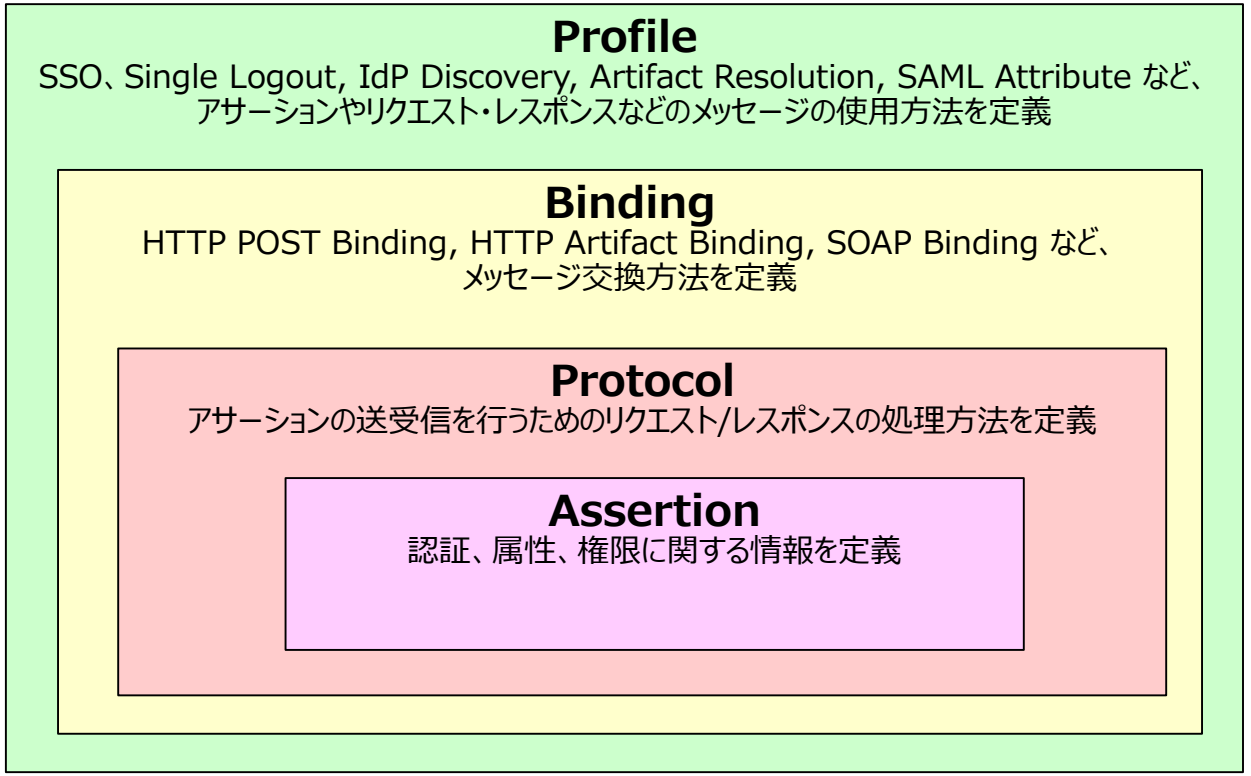
参考情報

1. SAML Webシングルサインオンの概要

SAMLの概要

- SAMLについて
  - **S**ecurity **A**ssertion **M**arkup **L**anguageの略で、標準化団体OASISによって策定されたセキュリティー関連の情報を交換するためのXMLベースのフレーム・ワーク
    - アサーションと呼ばれるXML文書を使用してユーザーのセキュリティー情報(認証、属性、権限に関する情報)を記述、伝達する

- SAML仕様は下記4要素で構成される
  - プロファイル(Profile)
    - Web Browser SSO Profile
    - Single Logout Profile など
  - バインディング(Binding)
    - HTTP POST Binding
    - HTTP Artifact Binding など
  - プロトコル(Protocol)
    - Authentication Request Protocol
    - Single Logout Protocol など
  - アサーション(Assertion)
    - Authentication Statement
    - Attribute Statement など



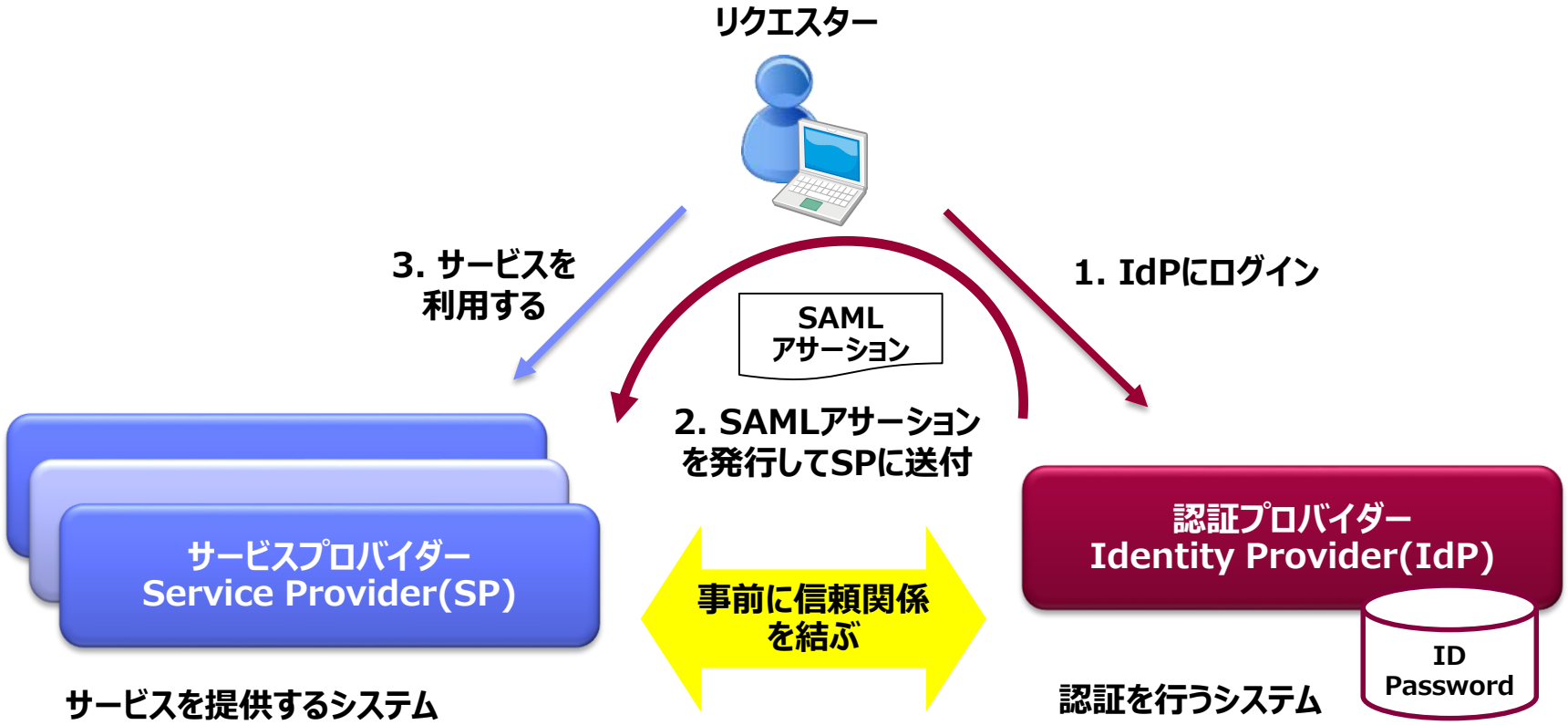
SAML Webシングルサインオンとは

- セキュリティードメインを超えたWebサイトのシングルサインオンが可能
  - SAML Web Browser SSO Profileで定義
  - 認証連携(フェデレーション)技術のひとつでエンタープライズ分野において多く活用されている
  - 企業間やクラウドとオンプレミス間のシングルサインオンを行いたい場合に有効
  - オープンスタンダードであるため異なるベンダーの製品間でも利用することが可能

- SAML Webシングルサインオンのメリット
  - エンドユーザーの観点：
    - 認証プロバイダーのユーザーIDでサービスが利用できるため、サイト毎のパスワードを覚える必要がない
    - フェデレーションプロトコルに対応するWebサイト間でシングルサインオンが可能
  - アプリケーション開発者およびサービス提供者の観点：
    - 認証機能のアウトソースすることによる実装コストの低減
    - ユーザーとサービスプロバイダーの間ではパスワード情報が流れず、サービス側でパスワード管理が不要
    - 認証プロバイダーで提供されている高度な認証機能（2要素認証など)を簡単に利用することが可能
    - エンドユーザーの利便性向上に伴うサービスの満足度向上

SAML Webシングルサインの構成要素

- 認証プロバイダー(IdP: Identity Provider)
  - リクエスターを認証し、SAMLアサーションを発行する役割を持つ
- サービス・プロバイダー(SP: Service Provider)
  - 実際のサービスを提供する側でSAMLアサーションを検証する役割を持つ

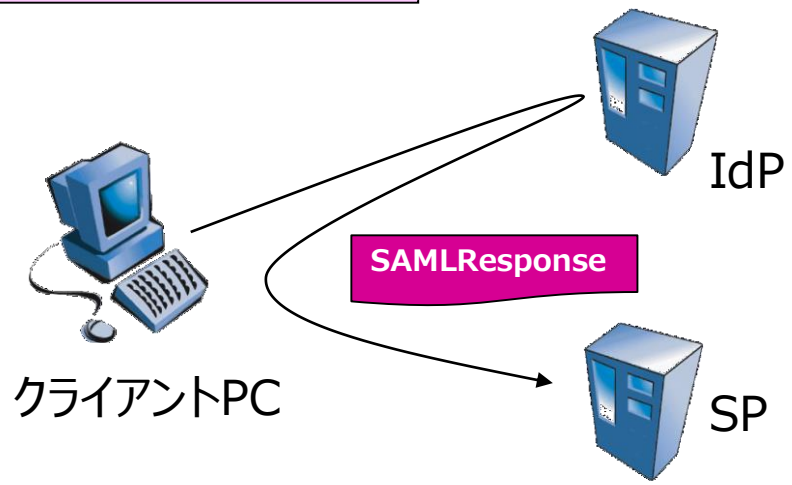


(*)WASがサポートするWeb Browser SSO Profile / HTTP POST Bindingを想定

SAML Web SSO 2種類の処理フロー

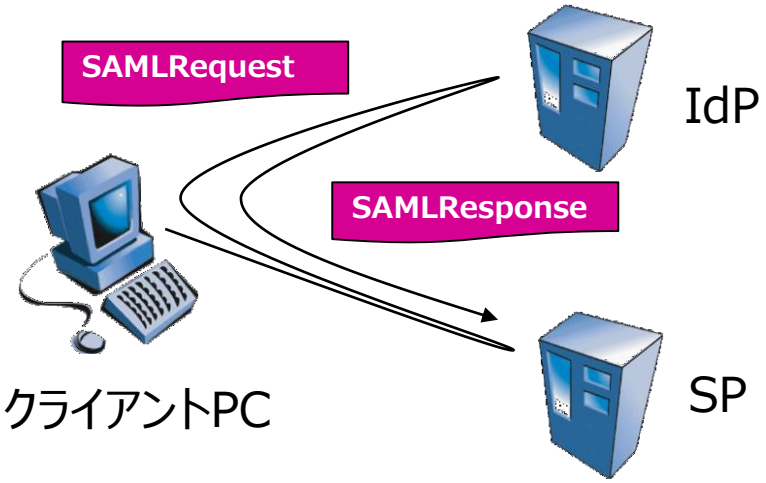
- SAML仕様では2種類の処理フローが定義されている

IdP-Initiated SSO



- ユーザーはSPにアクセスする前にIdPに明示的にアクセスして認証を受けておかなければならない。
- IdP-Initiatedの場合にはサインオン用のポータル画面を事前に作っておくといった工夫が必要。

SP-Initiated SSO



- ユーザーは使いたいサービスにアクセスして、認証されていなければ自動的にIdPにリダイレクトされる。
- ユーザビリティーの観点では、エンドユーザーは直感的に操作できる。

WAS Full ProfileにおけるSAML Webシングルサインオンのサポートについて (1/2)

- SAML Service Providerのサポート
  - WAS V8.5.5 / WAS V8.0 Fix Pack 4以降 / WAS V7.0 Fix Pack 23以降でSAML SPになることが可能
  - SAML TAIおよびAssertion Consumer Service(ACS)アプリケーションを提供
  - WebSphere PortalやBusiness Process Management(BPM)などWASベースの製品もSAML SPになることが可能
  - ただし、WASはIdPにはなれないため、別製品で構築する必要がある
    IBM製品：IBM Tivoli Federated Identity Manager あるいは後継製品の IBM Security Access Manager V9.0
    他ベンダー製品：Active Directory Federation Services (ADFS) など

WAS Full ProfileにおけるSAML Webシングルサインオンのサポートについて (2/2)

- サポートするSAML仕様
  - SAML 2.0 Web Browser SSO Profile および HTTP Post Bindingの組み合わせのみサポート
  - 処理フローはIdP-Initiated SSOのみサポート
  - SP-Initiated SSOはサポートされない(*)
    - SPに直接アクセスした場合にIdPにHTTPリダイレクトするように設定することが可能
    - WASはSP-Initiatedの仕様で定義されたSAML RequestをIdPに送るわけではなく、単純にリダイレクトしているだけ
    - リダイレクトされた後の処理フローはIdP-Initiated SSOとして動く
    - エンドユーザーの観点からはSPのURLさえ分かっていればよく、認証シーケンスもSP-Initiated SSOと同じように見せかけることが可能
  - 尚、WAS Liberty ProfileはIdP-Initiated SSOとSP-Initiated SSOの両方の処理フローをサポート

(*)WAS Full Profile V8.5.5.7以降でカスタムコードの作成によりSP-Initiated SSOのサポートが追加
詳細は下記APARを参照
PI34088: ERROR IN SAML WEB SSO TAI WITH CUSTOM SP-INITIATED SSO
http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg1PI34088

- 当ガイドでは上記SP-Initiated SSOのシナリオは対象外とし、SPに直接アクセスする場合はIdPにリダイレクトしてIdP-Initiated SSOで処理させる方法を紹介する

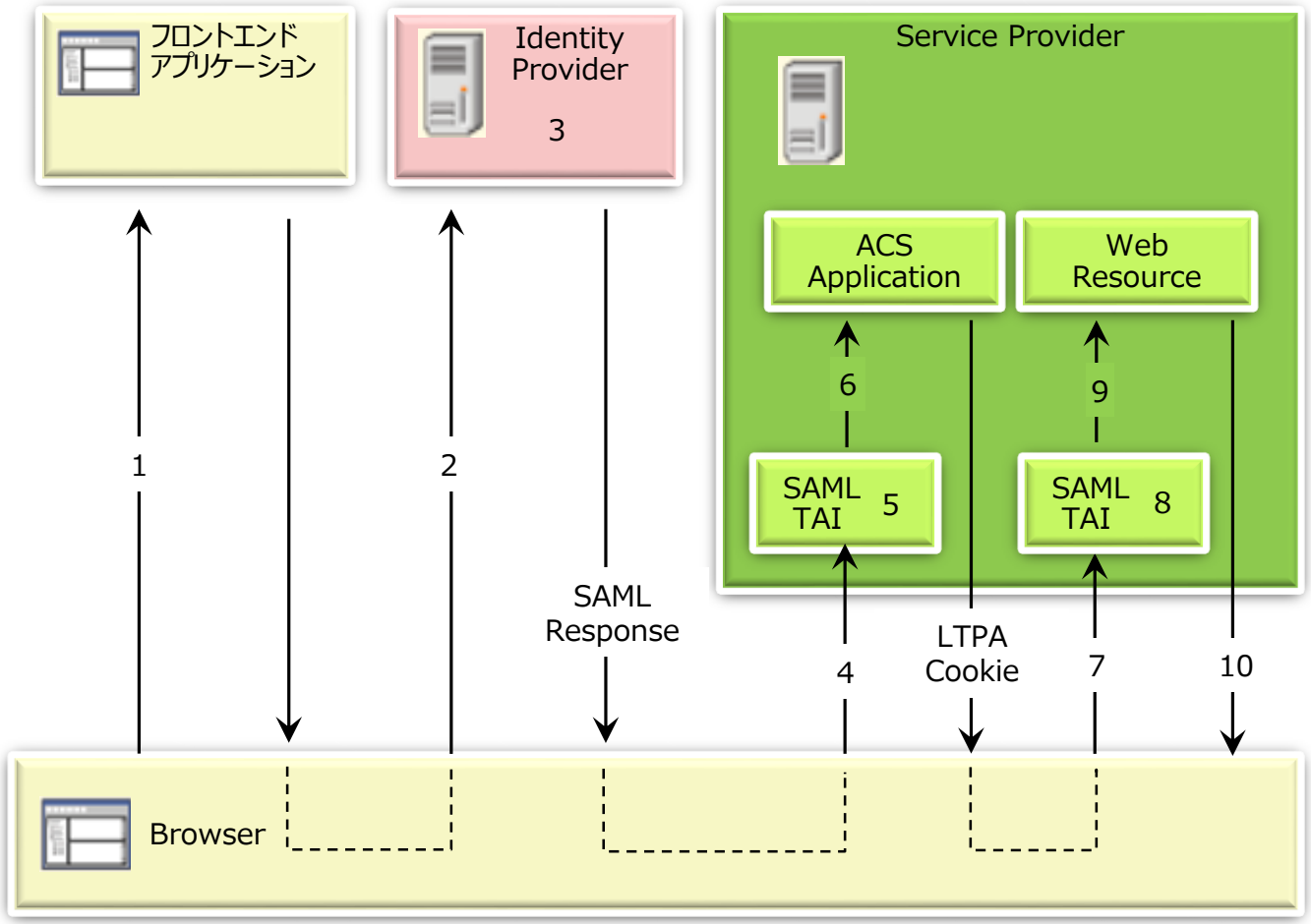
WAS SAML Webシングルサインオン機能

- WAS V8.5.5 SAML Web SSOには以下のような機能が含まれている
  - 複数のIdPでシングルサインオンが可能
  - IDアサーションおよびWASのユーザーレジストリーとSAMLアサーションをマッピングするオプションが提供されている
  - SAMLアサーション属性をセキュリティー・コンテキスト内のrealm、principal、unique Id、groupにマッピングあるいは追加することができる
  - カスタムIDマッピングを行うことができるプラグポイントが提供されている
  - SPのユーザーレジストリーからIDのグループメンバーシップ情報を取得してセキュリティーコンテキストに追加することができるオプションが提供されている
  - RSA-SHA1 および RSA-SHA256の署名アルゴリズムをサポート
  - SAMLアサーションはSubjectのPrivateCredentialに保持されており、ダウンストリームのWebサービス呼び出しやEJBの認証に利用することが可能
  - リクエストがIdPから送られなかった場合にリクエストを適切なIdPにリダイレクトするIdP選択フィルターが提供されている
  - ビジネス・アプリケーション URL が AssertionConsumerService URL の役割を果たすようにできるので、IdP は SAMLResponse をビジネス・アプリケーション URL に直接送ることができる
  - 監査への対応

- サポート機能と制約の詳細は下記を参照
  - http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.doc/ae/cwbs_samlssosummary.html

IdP-Initiated SSO 処理フローの流れ

- ユーザーがIdPの認証を受けてSPのサービスにアクセスするまでの処理フローは以下のとおり
  - 処理フローはIdP-Initiated SSOを想定

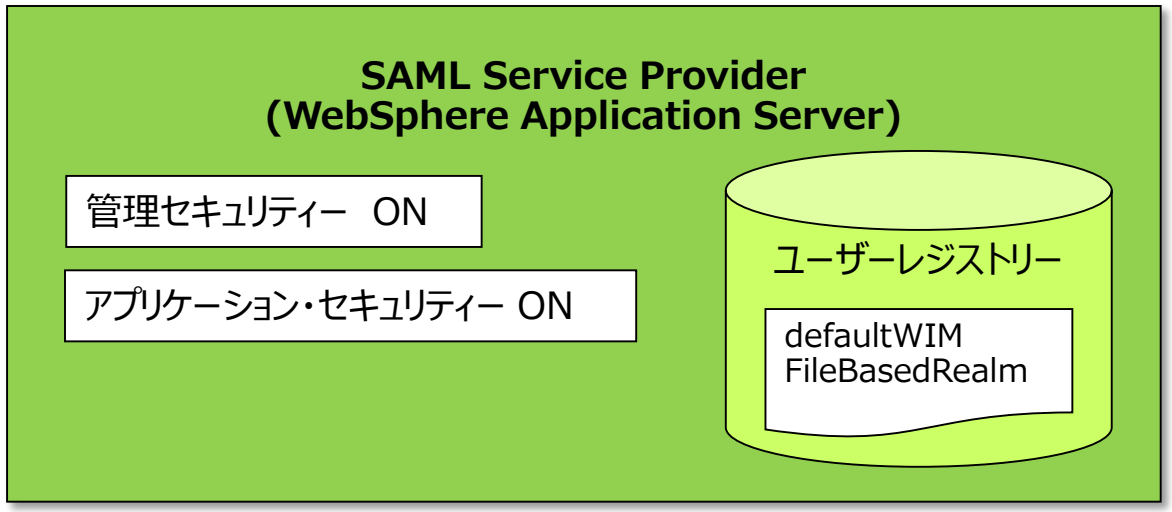


1. ユーザーがIdPのURLリンクが含まれるフロントエンドアプリケーションにアクセスする
2. フロントエンドアプリケーションはユーザーをIdPへリダイレクトさせてユーザーはIdPにログインする
3. IdPはユーザーの認証に成功するとSAMLアサーションを発行する
4. IdPはリクエストをSP側のAssertion Consumer Service(ACS)にリダイレクトさせて、HTTP POSTでSAMLResponse(SAMLアサーション)を送信する
5. SAML TAIがSAML Responseを受信してメモリー上にJAAS SubjectやLTPAトークンを作成する
6. リクエストはACSアプリケーションにディスパッチされる
7. HTTP応答にLTPA Cookieを追加して、リクエストをWebリソースにリダイレクトさせる
8. SAML TAIが再度呼ばれてリクエストにSAMLResponseが含まれているかどうかを確認し、SAMLResponseではなくLTPA Cookieが含まれるため通常のLTPAベースのログインプロセスとなる
9. LTPA Cookieをセキュリティー・コンテキストにマップしてリクエストされたWebリソースに対するユーザーアクセスを許可する
10. WASはユーザーに対してHTTP応答を送信する

2. WAS SAML Service Provider 構成方法

前提構成

- 当ガイドでは認証プロバイダーとしてTivoli Federated Identity Managerを使用する
- WASの前提構成は以下のとおり
  - 1台のアプリケーションサーバー構成
  - 管理セキュリティーおよびアプリケーション・セキュリティーの有効化
    - アプリケーション(EAR/WAR)とWASランタイムのセキュリティー・ロールのマッピングも必要
  - ユーザーレジストリーの構成(オプション)

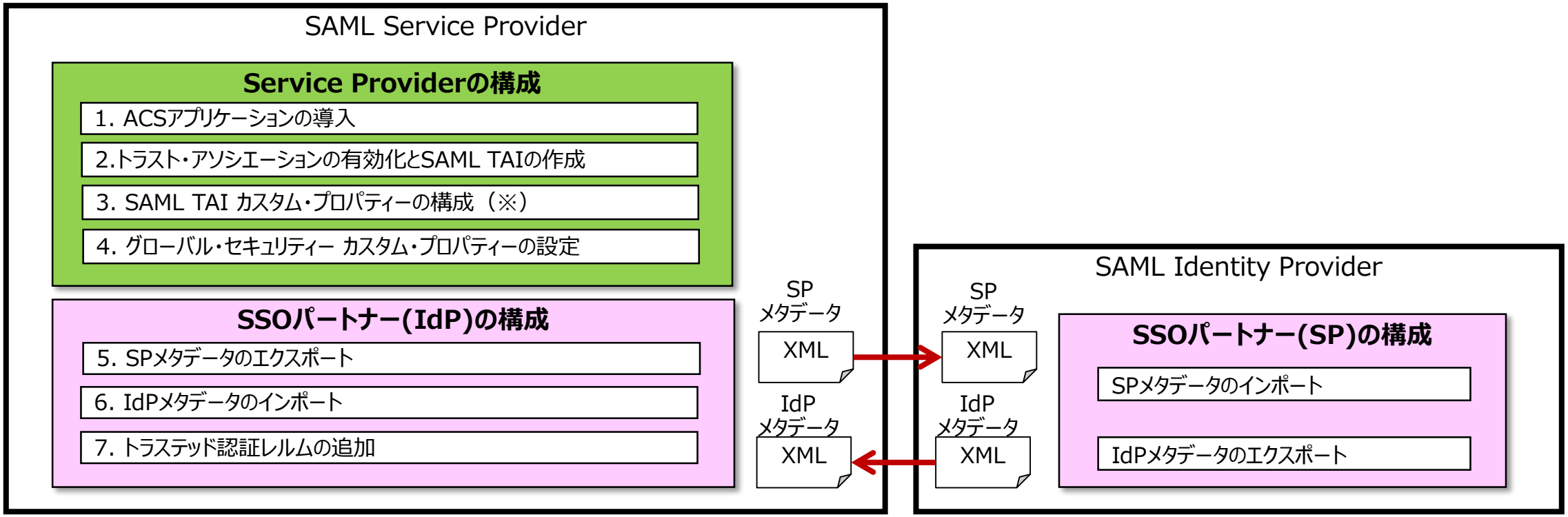


SAML Identity Provider
(Tivoli Federated Identity Manager)

構成方法については当ガイドの対象外

SAML Web SSO の構成方法

- SAML Web SSOの構成の流れは下記図のとおり
  - SP側ではService Providerの構成とSSOパートナーであるIdPの構成を行う
  - IdP側ではWASのSPメタデータをインポートしてもらう必要がある
- 以下のような要件への対応として、オプション構成としてSAML TAIカスタム・プロパティー（※）を追加・変更していくことになる
  - SPへ直接アクセスする処理フロー / ユーザーレジストリーとのIDマッピング / SAMLアサーションの暗号化 / 複数IdPが存在するケース など



1. SAML ACSアプリケーションのインストール

Service Providerの構成

1. 管理コンソールにログインする
2. WASが提供するSAML ACSアプリケーションをアプリケーションサーバーあるいはクラスターにインストールする
   - アプリケーションは以下のEARファイルを指定する
     - <was_install_root>/installableApps/WebSphereSamlSP.ear
       - アプリケーション名：WebSphereSamlSP
       - コンテキストルート：samlsps

- wsadminコマンドでアプリケーションをデプロイすることも可能

単体構成の場合

```
wsadmin -f installSamlACS.py install <nodeName> <serverName>
```

クラスター構成の場合

```
wsadmin -f installSamlACS.py install <ClusterName>
```

【Tips】ビジネスアプリケーションが直接SAMLアサーションを受け取ることも可能

- HTTP POSTで送られるSAMLアサーションをサービスを提供するビジネスアプリケーションが受け取ることもできる
- その場合、製品提供のSAML ACSアプリケーションを導入する必要はない
- 後述のSAML TAIカスタム・プロパティー「sso_1.sp.acsUrl」にビジネスアプリケーションのURLを指定する必要がある

2. トラスト・アソシエーションの有効化とSAML TAIの作成

Service Providerの構成

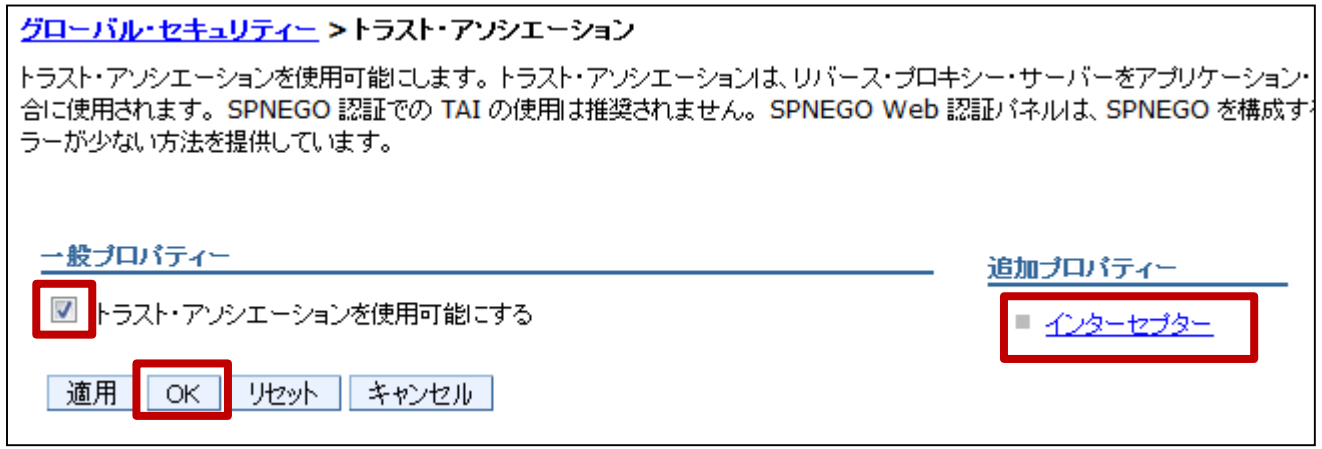
1. 「セキュリティー」>「グローバル・セキュリティー」>「認証」>「Web および SIP セキュリティー」 >「トラスト・アソシエーション」を選択して、「トラスト・アソシエーションを使用可能にする」にチェックを入れて「OK」ボタンをクリックする

2. 「トラスト・アソシエーション」 >「インターセプター」を選択する

3. 「新規作成」ボタンをクリックする

4. インターセプター・クラス名として以下の値を入力して「OK」ボタンをクリックする

   com.ibm.ws.security.web.saml.ACSTrustAssociationInterceptor

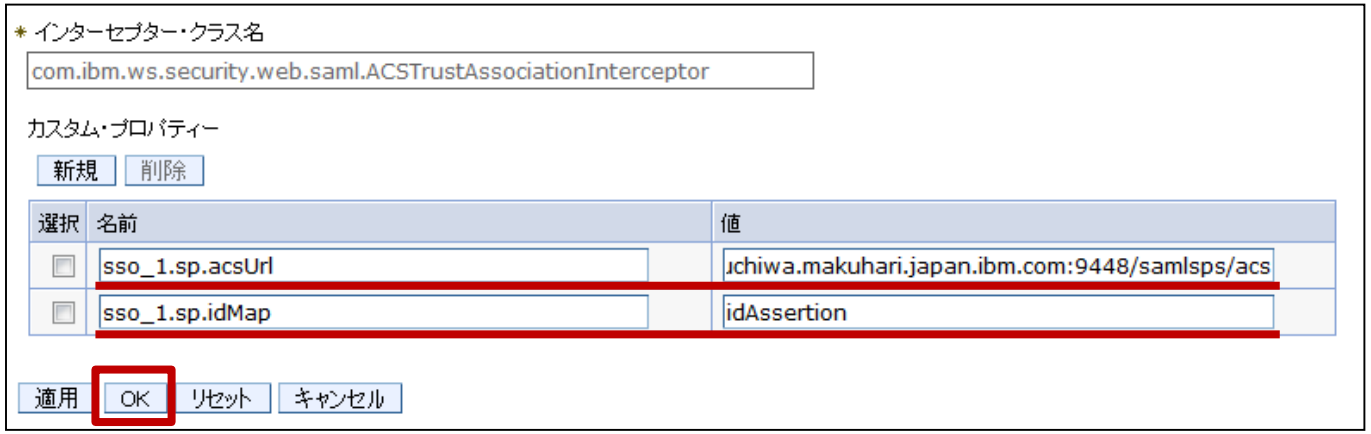


3. SAML TAIカスタム・プロパティーの設定

Service Providerの構成

1. 「トラスト・アソシエーション」 >「インターセプター」>「com.ibm.ws.security.web.saml.ACSTrustAssociationInterceptor」を選択する
2. 以下のカスタム・プロパティーを定義して「OK」ボタンをクリックする

名前	値	説明
sso_1.sp.acsUrl	https://<hostname>:<port>/samlsps/<any URI pattern string> 例) https://webserver01/samlsps/acs https://appserver01:9443/samlsps/acs	ACSアプリケーションのURLを指定する。<any URI pattern string>はSP毎に一意となるように設定する。IdPに提供するSPメタデータの情報に含まれ、IdPがHTTP POSTでSAMLアサーションを送信する際の宛先URLとして使用される。 尚、ACSアプリケーションを導入せずにSAML HTTP POSTリクエストを直接ビジネスアプリケーションで受け付ける構成にする場合はビジネスアプリケーションのURLを指定する。 クラスター構成で負荷分散構成になっている場合、<hostname>:<port>は代表アドレスおよびポート番号を指定する。
sso_1.sp.idMap	idAssertion	WASのユーザーレジストリーとのIDマッピングを行うかどうかの設定。idAssertionはIDマッピングを行わなずにIDアサーションユーザーとしてWAS上のターゲットアプリケーションにログインする。

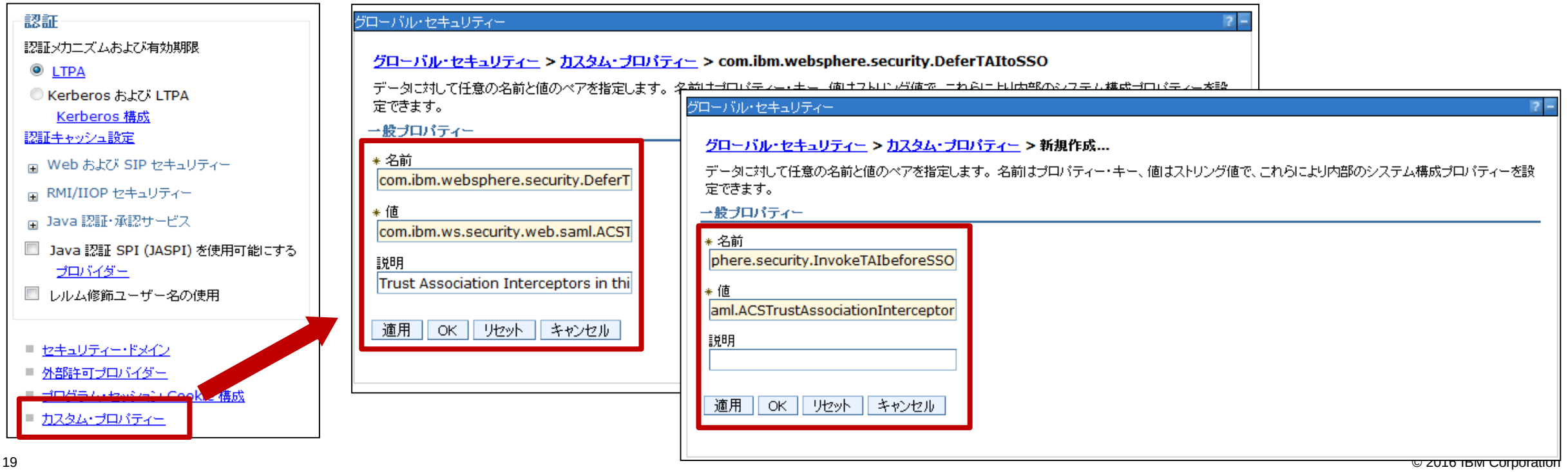


4. グローバル・セキュリティーのカスタム・プロパティーの設定

Service Providerの構成

1. 「セキュリティー」>「グローバル・セキュリティー」>「カスタム・プロパティー」を選択する
2. 下記2つのカスタム・プロパティーの値を変更する。カスタム・プロパティーが存在しなければ新規追加する。

名前	値	説明
com.ibm.websphere.security.DeferTAItoSSO	com.ibm.ws.security.web.saml.ACSTrustAssociationInterceptor	LTPA CookieもSAMLアサーションも送らずにSPに直接アクセスした際にSAML TAIが呼ばれる
com.ibm.websphere.security.InvokeTAIbeforeSSO	com.ibm.ws.security.web.saml.ACSTrustAssociationInterceptor	LTPA Cookieを保持しており、新しいSAMLアサーションをACSにPOSTした場合にLTPA Cookieを無視して新しいSubjectを作成する



5. SPメタデータのエクスポート

SSOパートナー(IdP)の構成

1. wsadminコマンドを利用してSPのメタデータをエクスポートする

```
AdminTask.exportSAMLSpMetadata('-spMetadataFileName /work/<SPMetaDataFile> -ssoId 1')
```

実行例

```
#/usr/IBM/WebSphere/AppServer/profiles/samlBase01/bin/wsadmin.sh -lang jython -user wasadmin –password xxxx
WASX7209I: ノード samlNode のプロセス "server1" に、SOAP コネクターを使用して接続しました。プロセスのタイプは UnManagedProcess です。
WASX7031I: ヘルプを表示するには、"print Help.help()" と入力してください。
wsadmin>AdminTask.exportSAMLSpMetadata('-spMetadataFileName /work/was_sp_metadata.xml -ssoId 1')
'true'
wsadmin>
```

2. エクスポートしたSPメタデータのファイルをIdPに提供する（その後、IdP側でSPメタデータのインポートを実施する）
   – 上記コマンドでエクスポートしたSPメタデータのサンプルは以下のとおり

was_sp_metadata.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:EntityDescriptor xmlns="http://www.w3.org/2000/09/xmldsig#" xmlns:ns2="urn:oasis:names:tc:SAML:2.0:metadata"
xmlns:ns3="http://www.w3.org/2001/04/xmlenc#" xmlns:ns4="urn:oasis:names:tc:SAML:2.0:assertion"
entityID="https://uchiwa.makuhari.japan.ibm.com:9448/samlsps/acs">
   <ns2:SPSSODescriptor AuthnRequestsSigned="true" WantAssertionsSigned="true" protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
      <ns2:AssertionConsumerService index="0" Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://uchiwa.makuhari.japan.ibm.com:9448/samlsps/acs"/>
   </ns2:SPSSODescriptor>
</ns2:EntityDescriptor>
```

6. IdPメタデータのインポート (1/4)

1. IdPのメタデータ(XMLファイル)を入手する
2. IdPメタデータファイルを次ページのwsadminコマンドを実行する区画に配置する（当ガイドで使用するTFIMのメタデータは以下のとおり）

WASSAML_ISE_metadata.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<md:EntityDescriptor entityID="https://idp.ise.com:444/was/sps/WASSAML/saml20" xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata">
<md:IDPSSODescriptor WantAuthnRequestsSigned="true" protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
<md:KeyDescriptor use="signing"><KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#"><X509Data><X509Certificate>MIICBzCCAXCgAwIB(中略
)dzxdhikBMZPgdyQ==</X509Certificate></X509Data></KeyInfo></md:KeyDescriptor>
<md:KeyDescriptor use="encryption"><KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#"><X509Data><X509Certificate>MIICBzCCAXCgAwIB(中略
)dzxdhikBMZPgdyQ==</X509Certificate></X509Data></KeyInfo><md:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/></md:KeyDescriptor>
<md:ArtifactResolutionService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP" Location="https://idp.ise.com:444/was/sps/WASSAML/saml20/soap" index="0"
isDefault="true"/>
<md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact" Location="https://idp.ise.com:444/was/sps/WASSAML/saml20/slo"/>
<md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="https://idp.ise.com:444/was/sps/WASSAML/saml20/slo"/>
<md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP" Location="https://idp.ise.com:444/was/sps/WASSAML/saml20/soap"/>
<md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:persistent</md:NameIDFormat><md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-
format:transient</md:NameIDFormat><md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</md:NameIDFormat>
<md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:encrypted</md:NameIDFormat><md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-
format:unspecified</md:NameIDFormat>
<md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact" Location="https://idp.ise.com:444/was/sps/WASSAML/saml20/login"/>
<md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="https://idp.ise.com:444/was/sps/WASSAML/saml20/login"/>
</md:IDPSSODescriptor>
<md:Organization><md:OrganizationName xml:lang="ja">ISE</md:OrganizationName><md:OrganizationDisplayName
xml:lang="ja">ISE</md:OrganizationDisplayName><md:OrganizationURL xml:lang="ja"/></md:Organization>
<md:ContactPerson
contactType="technical"><md:Company>ISE</md:Company><md:GivenName/><md:SurName/><md:EmailAddress/><md:TelephoneNumber/></md:ContactPerson>
</md:EntityDescriptor>
```

6. IdPメタデータのインポート (2/4)

SSOパートナー(IdP)の構成

3. IdPから提供されたメタデータをwsadminコマンドでインポートして保存する
   - <IdPMetaDataFile>はIdPのメタデータのファイル名を記載する。
   - <idpAlias>はIdpから提供される証明書の別名として任意の値を指定する。WASのトラストストアにIdPの証明書が署名者証明書として追加されることになる。その署名者証明書の別名として使用される。

```
AdminTask.importSAMLIdpMetadata('-idpMetadataFileName <IdPMetaDataFile> -idpId 1 -ssoId 1 -signingCertAlias <idpAlias>')
```

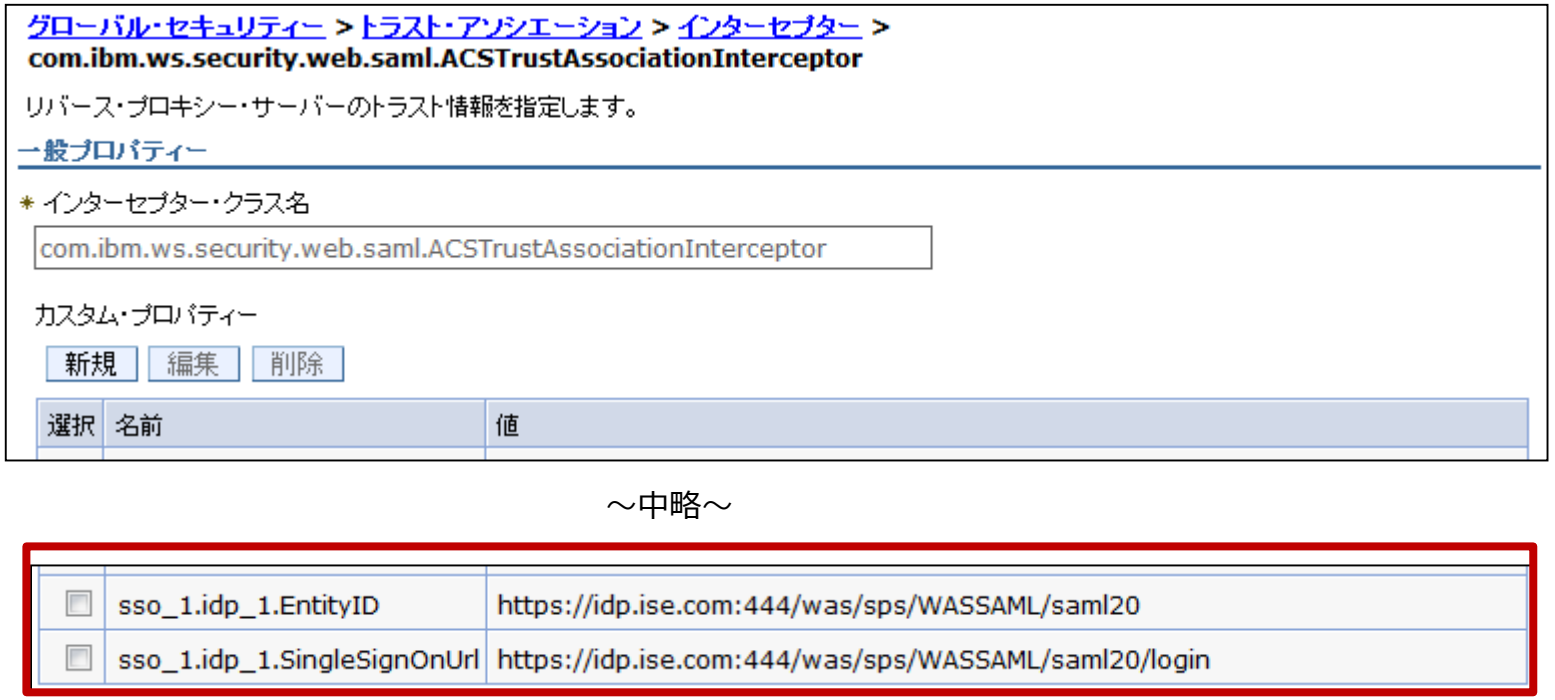
実行例

```
#/usr/IBM/WebSphere/AppServer/profiles/samlBase01/bin/wsadmin.sh -lang jython -user wasadmin -password xxxx
WASX7209I: ノード samlNode のプロセス "server1" に、SOAP コネクターを使用して接続しました。プロセスのタイプは UnManagedProcess です。
WASX7031I: ヘルプを表示するには、"print Help.help()" と入力してください。
wsadmin>AdminTask.importSAMLIdpMetadata('-idpMetadataFileName /work/WASSAML_ISE_metadata.xml -idpId 1 -ssoId 1 -
signingCertAlias FIMCert')
'true'
wsadmin>AdminConfig.save()
''
wsadmin>
```

6. IdPメタデータのインポート (3/4)

SSOパートナー(IdP)の構成

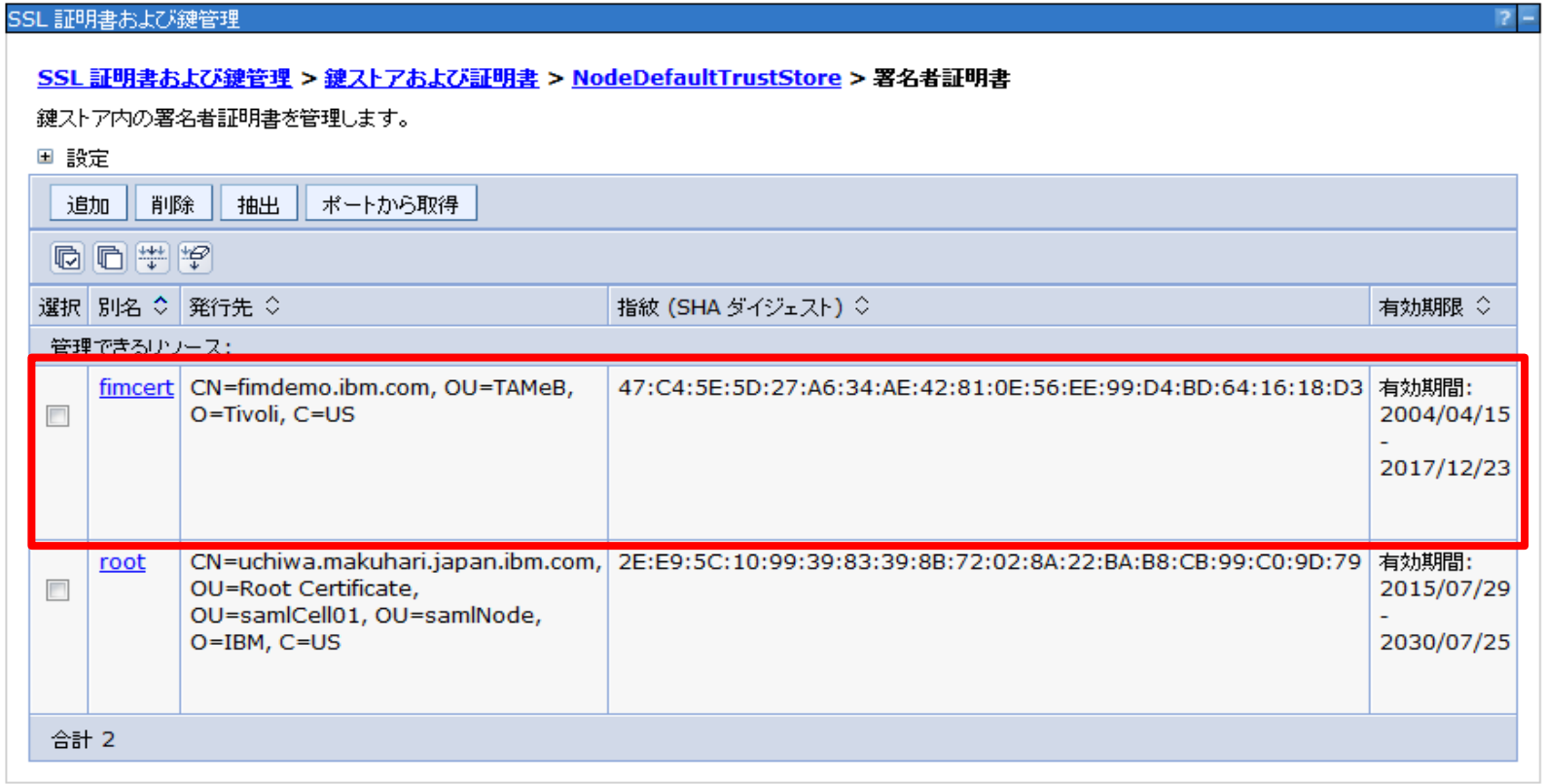
4. 管理コンソールにログインして、「セキュリティー」>「グローバル・セキュリティー」> 「トラスト・アソシエーション」>「インターセプター」>「com.ibm.ws.security.web.saml.ACSTrustAssociationInterceptor」を選択し、SAML TAIカスタム・プロパティーとして以下の項目が追加されていることを確認する
   - sso_1.idp_1.EntityID
   - sso_1.idp_1.SingleSignOnUrl



6. IdPメタデータのインポート (4/4)

SSOパートナー(IdP)の構成

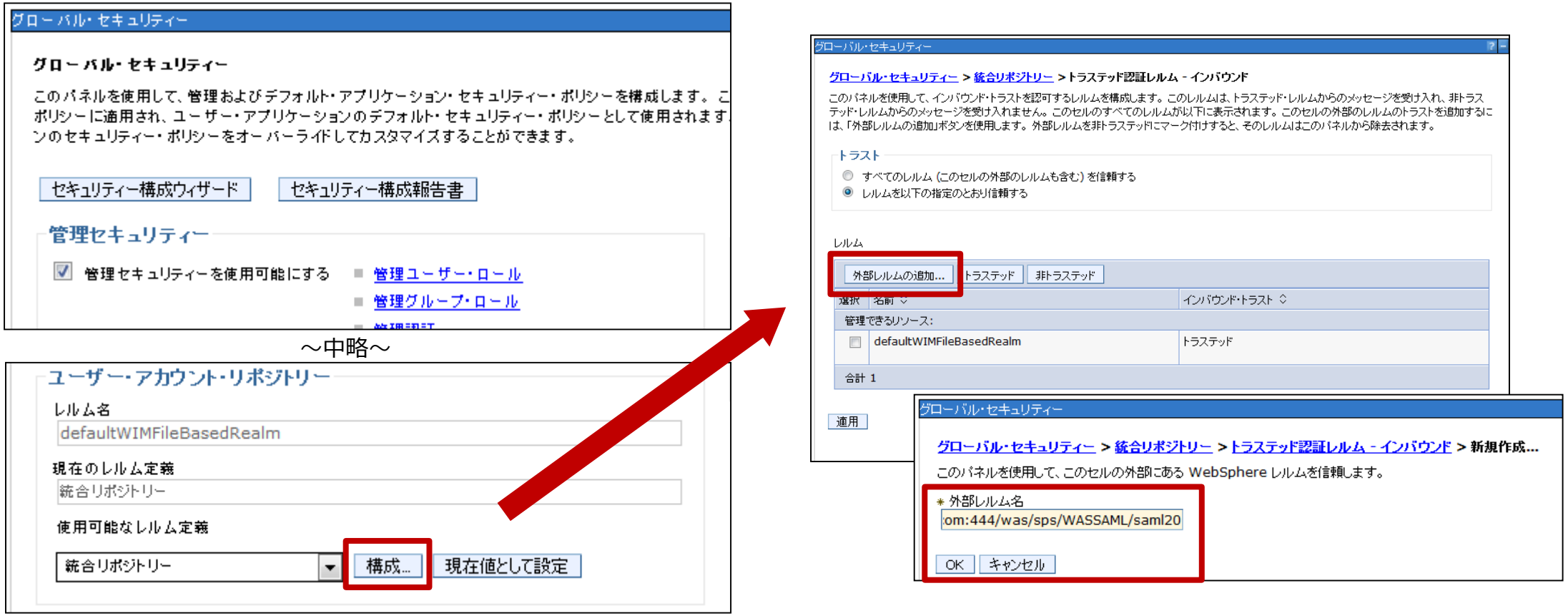
5. 管理コンソールにログインして、IdPの証明書がインポートされていることを確認する
   - 単体構成の場合は「セキュリティー」>「SSL 証明書および鍵管理」 >「鍵ストアおよび証明書」>「NodeDefaultTrustStore」>「署名者証明書」
   - クラスター構成の場合は「セキュリティー」>「SSL 証明書および鍵管理」>「鍵ストアおよび証明書」>「CellDefaultTrustStore」>「署名者証明書」



7. トラステッド認証レルムの追加

SSOパートナー(IdP)の構成

1. 管理コンソールにて、「セキュリティー」>「グローバル・セキュリティーー」>「ユーザー・アカウント・リポジトリー」>「構成」>「トラステッド認証レルム - インバウンド」を選択して「外部レルムの追加」をクリックする
2. 外部レルムを入力して「OK」ボタンをクリックする。レルム名の指定については次ページ参照。



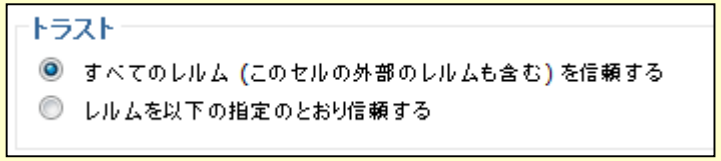
7. トラステッド認証レルムとして追加するIdPのレルム名について

SSOパートナー(IdP)の構成

- WASはデフォルトではIdPの「Issuer Name」の値がレルム名となる
  - TFIMの場合、IdPメタデータの「EntityID」の値がレルム名になっている
- IdPのレルム名はカスタマイズが可能
  - SAML TAIカスタムプロパティー「sso_<id>.sp.realmName」にて任意のSAML属性を指定することができる

【Tips】IdPのレルム名が不明、あるいは一致せずに認証エラーになる場合

- 「トラステッド認証レルム - インバウンド」にて「すべてのレルム (このセルの外部のレルムも含む) を信頼する」を選択することでエラーを回避できる
- ただしセキュリティーの観点で「レルムを以下の指定のとおり信頼する」が推奨
- サーブレットに以下のようなコードを追加することで、IdPのレルム名を確認することが可能であり、確認したレルム名をトラステッド認証レルムとして指定する



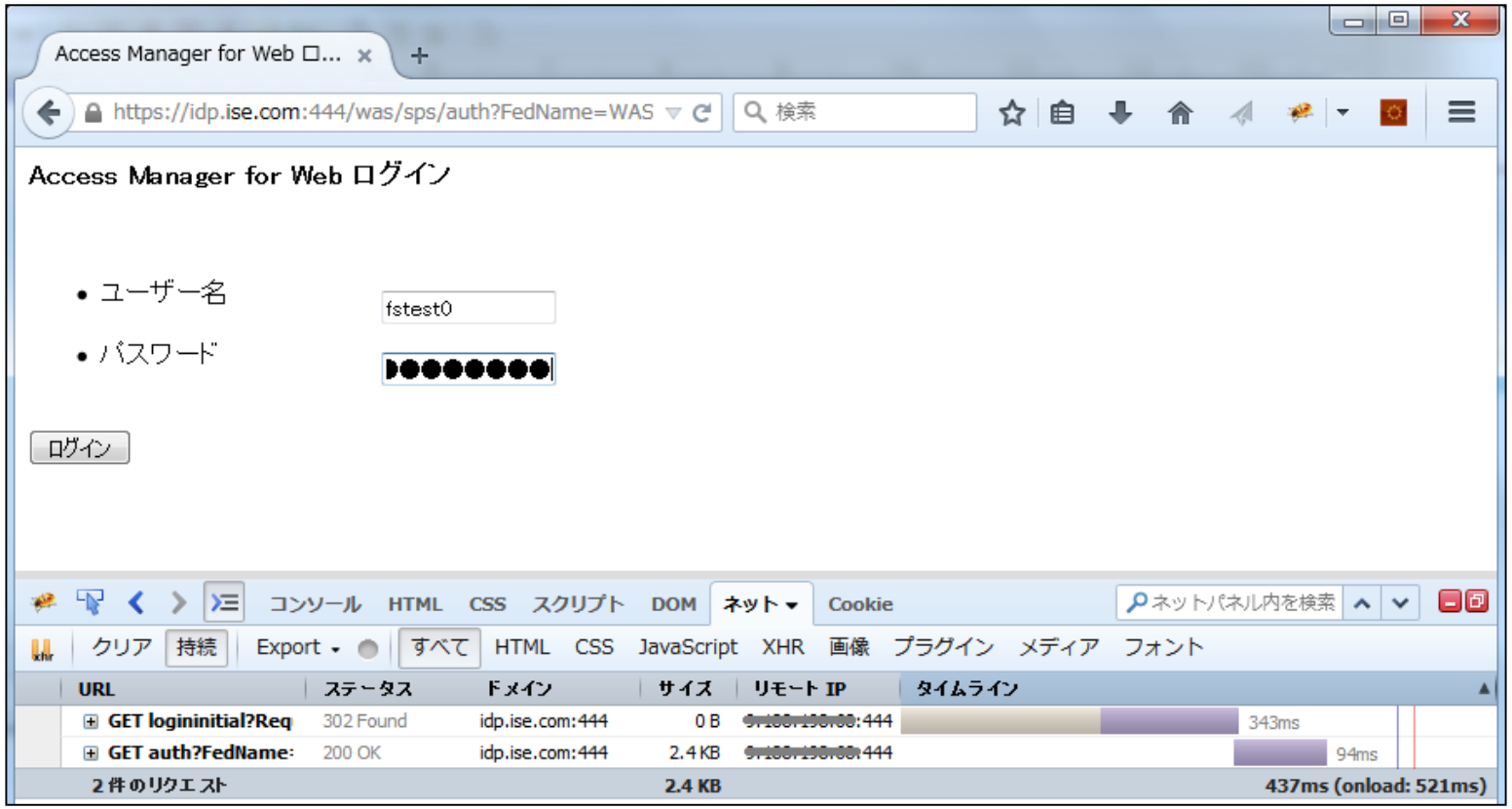

```
import javax.security.auth.Subject;
import com.ibm.websphere.security.auth.WSSubject;
import com.ibm.websphere.security.cred.WSCredential;
                ...中略...
Subject callerSubject = WSSubject.getCallerSubject();
Set<WSCredential> wsPublicCredentials = callerSubject.getPublicCredentials(WSCredential.class);
Iterator<WSCredential> wsPublicCredentialsIterator = wsPublicCredentials.iterator();
if (wsPublicCredentialsIterator.hasNext()) {
    WSCredential wsPubCred = wsPublicCredentialsIterator.next();
    String realmName = wsPubCred.getRealmName();
    System.out.println("RealmName : "+ realmName);}
```

8. 構成の保存および反映

1. 最小構成の設定が完了したため、構成を同期、保存する
2. サーバーを再起動する

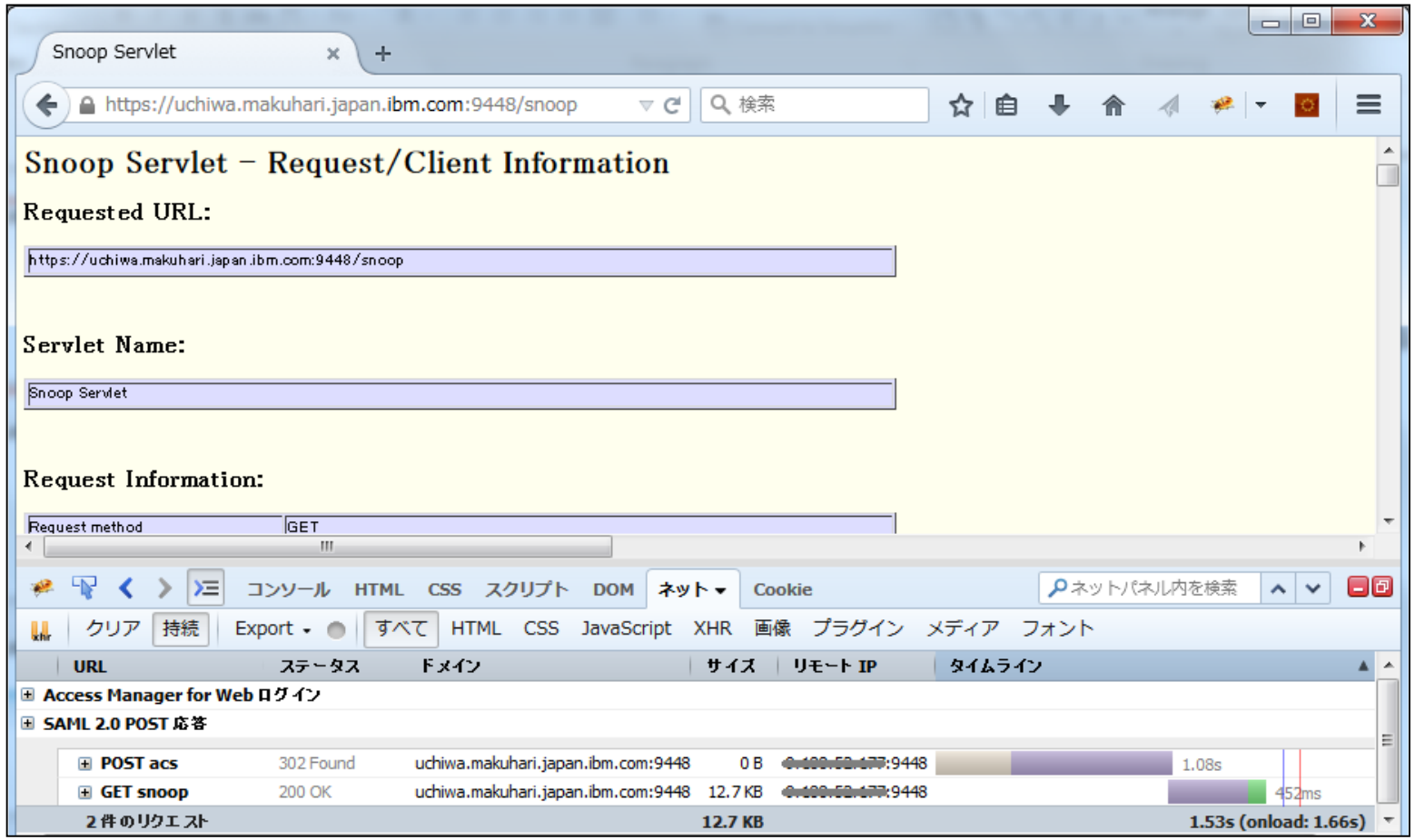
9. 動作確認

1. IdP Initiate SSO処理フローの動作確認をする
2. WebブラウザーでIdPのログインURLにアクセスする
   - https://idp.ise.com:444/was/sps/WASSAML/saml20/logininitial?RequestBinding=HTTPPost&NameIdFormat=email&PartnerId=https://uchiwa.makuhari.japan.ibm.com:9448/samlsps/acs&Target=https://uchiwa.makuhari.japan.ibm.com:9448/snoop
3. ユーザー名およびパスワードを入力してログインする



9. 動作確認

4. アプリケーションが表示されることを確認する
   - IdPへのログイン後にACSアプリケーションにSAMLアサーションをPOSTしてLTPAトークンが発行される
   - アクセスしたいアプリケーションにLTPA Cookieを送付することでアプリケーションにアクセスすることができる



SAML Web SSO オプションの検討項目

- 以上の最小構成にてSAML Web SSOの稼働確認が可能となった
- 最小構成に加えて必要に応じてSAML TAIカスタム・プロパティーを追加・変更することでSAML認証の挙動を柔軟に変更することが可能になっている
  - 設定可能なSAML TAIカスタム・プロパティーの詳細は下記URLを参照
    http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.doc/ae/rwbs_samltaiproperties.html?cp=SSAW57_8.5.5&lang=en

- 当ガイドで紹介する検討項目は以下のとおり
  - SPに直接アクセスする処理フローがある場合
  - SP側のユーザーレジストリーとのIDマッピングについて
  - SAMLアサーション属性とJAAS Subjectの対応関係について
  - SAML Web認証を行う場合のLTPA Web SSOの挙動について
  - SAMLアサーションの保護（署名・暗号化）について

SPに直接アクセスした場合の処理フローとフィルター機能について

- SPに直接アクセスした場合の処理フローについて
  - WAS Full ProfileはSAML仕様のSP-Initiated SSOをサポートしていないが、同様の処理フローを実現するためにSP側のアプリケーションに直接アクセスした場合にIdPのログインURLにHTTPリダイレクトさせることが可能になっている(Knowledge CenterではBookmark style SSOと表現されている)
  - SAML TAIカスタムプロパティー「sso_<id>.sp.login.error.page」に未認証の際のリダイレクト先を設定する

- フィルター機能について
  - WAS上のアプリケーションに直接アクセスした場合にSAML Web SSOの対象とするHTTPリクエストでフィルタリングすることが可能
  - SAML TAIカスタム・プロパティー「sso_<id>.sp.filter」でフィルター条件を指定することが可能

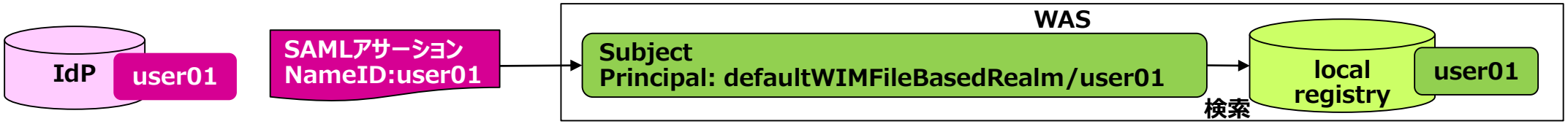
名前	値	説明
sso_<id>.sp.login.error.page	デフォルト値：なし	認証されていないリクエストのリダイレクト先URLを指定する。 エラーページやIdPログインページあるいはカスタムマッピングクラスを指定する。 IdPが複数ある場合、IdPログインページへのリンクがリストされているURLを指定する
sso_<id>.sp.acsErrorPage	デフォルト値： sso_<id>.sp.login.error.pageで指定した値	login.error.pageにIdPログインURLを指定した場合、SAMLアサーションの検証エラーの場合にもIdPのログインURLにリダイレクトされて無限ループに陥る可能性があるため、acsErrorPageを設定したほうがよい
sso_<id>.sp.filter	デフォルト値：なし	SAML Web SSOの対象となるHTTPリクエストの条件(HTTPヘッダーやリクエストURLなど)を指定してフィルタリングを行うことができる 例) request-url!=test105;From==jones@my.company.com5

ユーザーレジストリーとのIDマッピングについて ー SAMLアサーションがどのようにSubjectにマップされるのか？

- SAMLアサーションからセキュリティー・コンテキストを生成する際、WASのユーザーレジストリーとのIDマッピングを行うかどうかはSAML TAIカスタムプロパティー「sso_<id>_idMap」で設定することができる

1. **sso_<id>_idMap=idAssertion**
   - ローカルレジストリとのマッピングは行わずにIDアサーションユーザーとして扱う
   - ローカルレジストリーにユーザーが存在するかどうかのチェックを行わずにSAMLアサーションの情報からSubjectを生成する
   - デフォルトではSAML NameIDがSubjectのPrincipalにマップされる

2. **sso_<id>_idMap=localRealm**
   - ローカルレジストリーとSAMLアサーションのユーザーをマッピングさせて、ローカルレルムユーザーとして扱う
   - ユーザーレジストリーに対してユーザー(デフォルトではSAML NameID)が存在するかどうかを検証し、ローカルレジストリーからSubjectを生成する



3. **sso_<id>_idMap=localRealmThenAssertion**
   - ローカルレジストリーにユーザーが見つかった場合はローカルレルムユーザー、見つからなかった場合はIDアサーションユーザーとして扱う

SAMLアサーションの属性とJAAS Subjectの対応関係について

- SAMLアサーションの属性とSubject(principalおよびgroup、realmなど)の対応関係についてはSAML TAI カスタム・プロパティーで柔軟に変更することが可能

  例えば下記表のようなカスタム・プロパティーが提供されている

名前	値	説明
sso_<id>_sp.principalName	任意のSAML属性を指定する デフォルト値：SAML NameID	SubjectのPrincipalとして使用される
sso_<id>_sp.groupName	任意のSAML属性を指定する デフォルト値：なし	Subjectのgroupとして使用される
sso_<id>.sp.realmName	任意の文字列を指定する デフォルト値：SAML Issuer name	Subjectのrealmとして使用される
sso_<id>.sp.userMapImpl	カスタムマッピングクラスを指定する デフォルト値：なし	SAMLアサーションのユーザーをローカルレジストリー内の異なるユーザーにマッピングしたい場合に使用する。カスタムマッピングクラスはcom.ibm.wsspi.security.web.saml.UserMappingインターフェースを実装しなければならない。

- IdPから提供されたSAMLアサーションの情報はSubjectのPrivateCredentialとして保管されている
  - アプリケーションはSAMLトークンにアクセスすることが可能であり、ダウンストリームのEJB認証やWebサービス認証に利用することができる(SAML属性情報の取得方法は後述)

LTPAトークンとSAMLアサーションの関係

- SAML認証後にはLTPAトークンが発行されるためCookieによってログインセッションが維持される
- SAML認証後に発行されたLTPAトークンとSAML認証以外(Basic認証など)で発行されたLTPAトークンを同等に扱うかどうかは、SAML TAIカスタムプロパティーで制御される

1. sso_<id>.sp.enforceTaiCookie=true (デフォルト)
   - LTPA Cookieを受け取るとそのユーザーのSubjectにSAMLResponseが含まれているかどうかをチェックし、含まれていない場合はIdPへリダイレクトする
   - 例）app1がSAML認証の対象であり、app2が対象外の場合
   - app2にログインして発行されたLTPA Cookieをapp1に送った場合はSubjectのSAMLResponseがないためIdPへリダイレクトされる

2. sso_<id>.sp.enforceTaiCookie=false
   - SubjectにSAMLResponseが含まれているかどうかのチェックは行わないため、全てのLTPAトークンは同じものとして扱われる
   - 例) app2にログインして発行されたLTPA Cookieをapp1に送った場合は認証済みユーザーとしてLTPA Web SSOができる

3. sso_<id>.sp.cookieGroup=<任意のString>
   - SubjectにSAMLResponseに加えてcookieGroupの値を追加することでログインセッションをグルーピングすることが可能
   - SAML AudienceRestrictionに準ずる構成
   - 例）app1とapp2はSAML認証の対象だが、異なるIdPで認証を受けているケースで他のIdPで認証後に発行されたLTPAの場合はIdPにリダイレクトする
   - sso_1.sp.cookieGroup=app1 / sso_2.sp.cookieGroup=app2 と設定した場合
   - app1にログインした後にapp2にログインした場合はapp2のIdPにリダイレクトされる

SAMLアサーションの保護（署名・暗号化）

- 通信の暗号化
  - ACSアプリケーションへのHTTP POSTリクエストはHTTPS接続で送信されるべきである
- SAMLアサーションの署名
  - SAMLアサーションはIdPによって署名されているため、コンテンツ(アサーション情報)を改ざんすることはできない
- SAMLアサーションの暗号化
  - HTTPS接続によって暗号化されるため必須ではないが、SAMLアサーション自体を暗号化することが可能
    - IdPはリクエストをWebブラウザー経由でリダイレクトするため、ユーザーはブラウザーのプラグインを使用することでコンテンツを参照することが可能
  - 暗号化方法
    - SP側の秘密鍵と公開鍵を使用して暗号化を行う
    - IdPはSPから提供された証明書に含まれる公開鍵を使用してコンテンツを暗号化する
    - SPは公開鍵に対応する秘密鍵を持っているため復号化できる
    - 以下のSAML TAI カスタム・プロパティーが提供されている

名前	値
sso_<id>_sp.keyStore	個人証明書を含むキーストアの名前を指定する
sso_<id>_sp.keyAlias	鍵のエイリアスを指定する
sso_<id>.sp.keyPassword	キーストアのパスワードを指定する
sso_<id>.sp.keyName	鍵のDNを指定する(オプション)

3. Hint&Tips

アプリケーションからSAMLアサーションの属性情報を取得する方法

- IdPから受け取ったSAMLアサーション情報はSubjectのPrivateCredentialに保持されている
- アプリケーションはSAML Token Library APIを利用してSAMLアサーション情報を取得することができる
  - http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.doc/ae/rwbs_libraryapisaml.html

```
import javax.security.auth.Subject;
import com.ibm.websphere.security.auth.WSSubject;
import com.ibm.websphere.security.cred.WSCredential;
import com.ibm.websphere.wssecurity.wssapi.token.SAMLToken;
import com.ibm.wsspi.wssecurity.saml.data.SAMLAttribute;
.....中略.....
try {
  Subject callerSubject = WSSubject.getCallerSubject();   //Subjectを取得
  Set<SAMLToken> privateCredentials = callerSubject.getPrivateCredentials(SAMLToken.class); //SAMLTokenを取得
  for (SAMLToken samlToken : privateCredentials) {
    List<SAMLAttribute> allAttributes = samlToken.getSAMLAttributes(); //SAMLTokenの属性情報を取得
    for (SAMLAttribute anAttribute : allAttributes) {
    String attributeName = anAttribute.getName();   //属性名を取得
    String[] attributeValues = anAttribute.getStringAttributeValue(); //属性値を取得
    if (attributeValues.length > 1) {
        StringBuilder value = new StringBuilder();
        for (String attributeValue : attributeValues) { value.append(" [" + attributeValue + "]"); }
        System.out.println("AttributeName="+ attributeName + ", AttributeValue=" + value.toString());
    } else if(attributeValues.length == 1){
        System.out.println("AttributeName=" + attributeName + ", AttributeValue= [" + attributeValues[0] + "]");
    } else {
    System.out.println("AttributeName==" + attributeName + ", AttributeValue = [no values]"); }
}}} catch (WSSecurityException e) {
  e.printStackTrace();}
```

```
[16/02/12 12:03:01:389 JST] 000000f3 SystemOut     O AttributeName=group_name, AttributeValue= [fgroup]
[16/02/12 12:03:01:390 JST] 000000f3 SystemOut     O AttributeName=mail, AttributeValue= [fstest0_mail@tfim.com]
[16/02/12 12:03:01:390 JST] 000000f3 SystemOut     O AttributeName=employeenumber, AttributeValue= [fstest1234]
```

SystemOut.log抜粋

トラブルシューティング – SAML TAI トレースストリング

- SAML認証で問題が発生している場合にはサポート部門の指示に従ってトレースを取得する
- TAIに関するMustGatherが公開されている

MustGather: Web Single Sign-on TAI problems with WebSphere Application Server
https://www-304.ibm.com/support/docview.wss?uid=swg21971762#show-hide
----------抜粋----------
SAML Web Single Sign On
=info:com.ibm.ws.security.web.=all:com.ibm.websphere.wssecurity.*=all:com.ibm.ws.wssecurity.*=all:com.ibm.ws.wssecurity.platform.audit.*=off
----

参考資料

- WebSphere Application Server Knowledge Center – SAML web single sign-on
- http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.doc/ae/cwbs_samlssoconcepts.html?cp=SSAW57_8.5.5%2F1-13-2-4-21&lang=en

- Understanding the WebSphere Application Server SAML Trust Association Interceptor
- http://www.ibm.com/developerworks/websphere/techjournal/1307_lansche/1307_lansche.html

- Tivoli Federated Identity Manager Knowledge Center
- http://www-01.ibm.com/support/knowledgecenter/SSZSXU/welcome