Shibboleth IdPバージョン3に向けた NIIの取り組み

2016年5月27日　学術情報基盤オープンフォーラム2016
国立情報学研究所　西村 健

目次

学認から提供している情報の紹介を中心にIdPv3に向けた取り組みをご紹介します。

1. 冗長化のこと
2. uApproveJP 3.2.0
3. 複数の認証の組み合わせ
4. FPSP代替機能
5. 基本的なNameIDの取り扱い
6. 学認提供テンプレートの更新
7. 参考資料: v2からのアップグレードのプランニング

Shibboleth IdPバージョン3（IdPv3）

- 2014年12月22日リリース

- 送信属性選択・同意(uApproveJP相当)の標準搭載など，機能・メンテナンス性が向上
- 現在の最新バージョンは 3.2.1
  - 現在3.3.0に向けて精力的に開発継続中
- Shibboleth IdP ver.2.x系は 2016年7月31日 に開発元による全サポートを終了

IdPv3への移行（アップグレード）

- 現在 Shibboleth IdP 2.xをご利用の機関は，IdPv3へのアップグレードを行う必要があります。

- アプライアンス製品を利用して学認に参加している機関も，製品がShibboleth IdP 2.x系統をベースにしている場合，ご対応いただく必要があると考えられます。

- Shibboleth IdPのサポート終了だけではなく，OSのサポート期限にも注意してください。
  - CentOS 5　2017年3月31日まで
  - CentOS 6　2020年11月30日まで
  - （参考：CentOS 7　2024年6月30日まで）

新しく導入された概念

- *.properties
  - 「key = value」のフォーマットで書かれたファイル。XML編集なしで定型のカスタマイズを実現可能にするもの。
  - 使用箇所: idp.properties, ldap.properties, saml-nameid.properties, (services.properties)
    あと言語リソース

- Spring Web Flow
- bean
- Predicate (activationCondition)

いきなりですがクエスチョン

IdPv3にまつわるよくある誤解

- IdPv3はShibboleth IdP v2と互換性がない？

  →NO

- Shibboleth SPもバージョン3がリリースされている？

  →NO

- IdPv3へのアップグレードは、マイナーアップデート（例:2.3→2.4）と同じ？

  →NO

- IdPv3へのアップグレードは、設定をーからやり直す必要はない？

  →YES

  （一部の設定を引き継ぐことができます）

IdPv3の冗長化について

- バージョン2では4つの選択肢がありました。
  - Stateless Clustering方式
  - memcached方式
  - Terracotta方式
  - リレーショナルデータベース方式（後発）

IdPv3の冗長化について

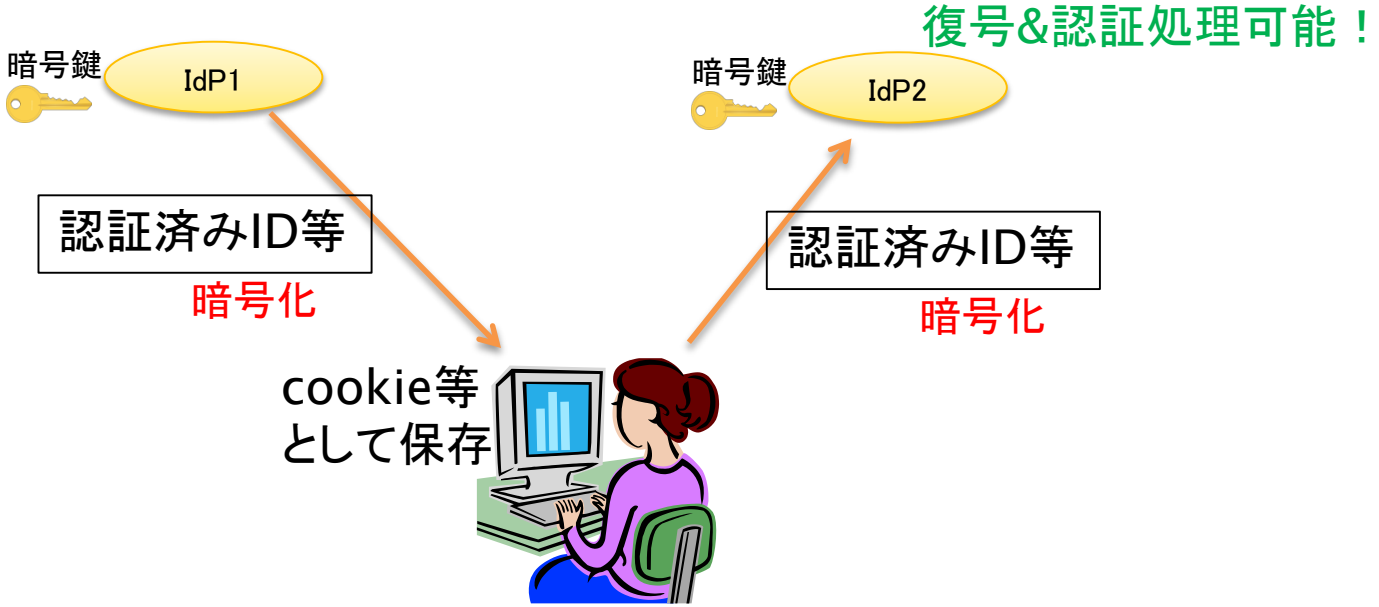
- IdPv3では以下のようになりました。
  - Stateless Clustering方式　→本体機能に取り込み、デフォルト化
  - memcached方式　→選択可
  - Terracotta方式　→廃止
  - リレーショナルデータベース方式~~(後発)~~　→選択可

- 学認では1番目と4番目の手順を公開しています。
  https://meatwiki.nii.ac.jp/confluence/x/25sxAQ

セッション情報を Client Storageに保存してノード間共有

Stateless Clusteringで使用していた下記機能が冗長化しなくても有効になっています

- IdPv3ではデフォルトでcookie/transientId等に情報を保存する
- 前回処理したサーバに依存しないため冗長化に向いている
- （ただしtransientId等のサイズが大きくなる）
- このためにインストール時に暗号鍵(sealer.jks)が生成される



uApproveJP 3.2.0

https://meatwiki.nii.ac.jp/confluence/x/ZwLO
IdPv3標準機能でもある程度送信属性同意機能は実現できますが、以下の点を強化しました。

- 必須属性でも送信拒否することが可能 の問題を解消
- 属性使用用途を表示できる
- サーバサイドで同意状況を記憶
  - 標準機能ではブラウザが記憶（デフォルト。変更可）
- 日本語化

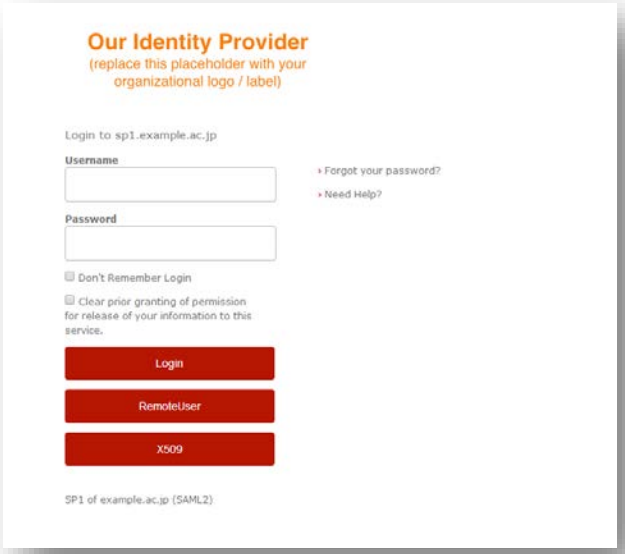
バージョン2版との相違点:

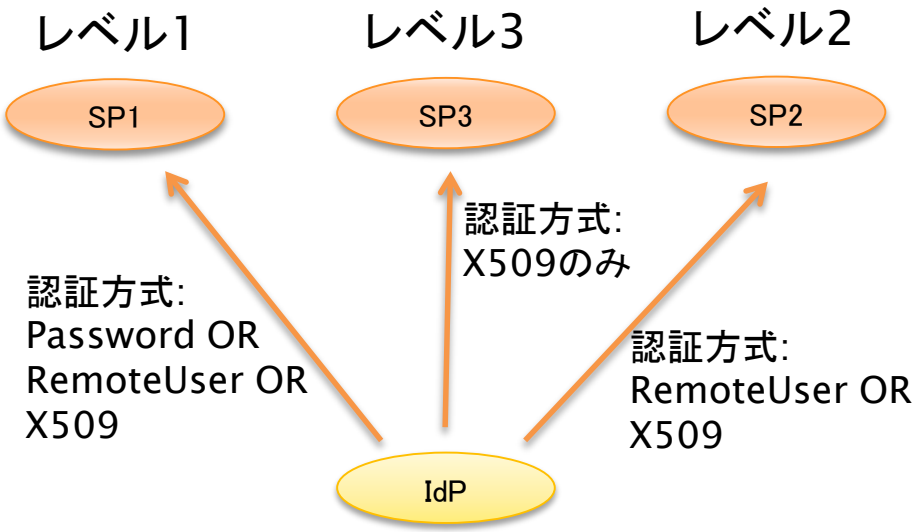
- チェックボックスについて前回選択状態で表示しない
- 属性送信確認画面（「この内容で送信しますよ」の確認）が挿入されない
- 過去の同意状況を一覧で表示するサーブレットがない
- ユーザ全員に再同意強制（DBフォーマット不一致のため）
- 属性の並び順が異なる
- 属性の内容が変化した場合に黄色三角マークを表示する機能がない
- 属性自体の説明が表示されない（要.vm修正）
- 利用規約表示機能の分離（本体機能として実装されたため）

複数の認証の組み合わせ

- https://meatwiki.nii.ac.jp/confluence/x/15cxAQ

- IdPv3の本体機能として追加された、認証のExtended Flowについて解説しています。
- SPごとに要求するレベルを設定し、レベルに応じて認証方式を提示するものです。





プラグイン代替例（FPSP相当）

- https://meatwiki.nii.ac.jp/confluence/x/GIAxAQ
- FPSPと書式は全く変わりますが、SP entityIDとePPNを指定して特定のユーザのみアクセス許可することが可能。

```
   <list>
<!--
     <bean parent="shibboleth.Conditions.RelyingPartyId" c:candidates="#{ 'https://sp.example
.org' }" />
-->
     <bean parent="shibboleth.Conditions.RelyingPartyId" c:candidates="#{ 'https://ex-sp-test
01.gakunin.nii.ac.jp/shibboleth-sp' }" />
     <bean class="net.shibboleth.idp.profile.logic.SimpleAttributePredicate">
       <property name="attributeValueMap">
         <map>
<!--
           <entry key="eppn">
-->
           <entry key="eduPersonPrincipalName">
             <list>
<!--
               <value>*</value>
-->
               <value>test002</value>
             </list>
           </entry>
         </map>
       </property>
     </bean>
   </list>
```

基本的なNameIDの取り扱い

- SAML仕様にはIdP/SP間の情報受け渡し方法として属性を使ったもの以外にNameIDによるものが定義されている。
- Shibboleth開発元ではeduPersonTargetedIDがdeprecatedとされ、今後はNameIDによる受け渡しが主流になるかも
- それ以外にも、すでにNameIDを使った受け渡しをしている場合にはIdPv3で扱いが異なる部分があるので注意しておくのが良い。

- 詳細は
  https://meatwiki.nii.ac.jp/confluence/x/iIooAQ
  の「//saml2:Subject/saml2:NameID」項目
  - transient-idが送られる条件
  - persistent-id（eduPersonTargetedIDと同等）の送りかた
    - computedId
    - storedId
  - 設定ファイル: saml-nameid.properties, saml-nameid.xml

学認提供テンプレートの更新

- 学認が
  https://meatwiki.nii.ac.jp/confluence/x/34S5
  で提供しているIdP向けの設定ファイルテンプレートが3.2.x向けに更新されました。
  `attribute-resolver-template.xml`
  `attribute-filter-template-prodfed.xml`
  `attribute-filter-template-testfed.xml`
  - ldap.propertiesを参照するように変更
  - eduPersonTargetedIDをpersistent-idと統一的に扱うための例追加
  - LDAPのStartTLSを使うための設定方法コメント追加
  - その他各種改善

- IdPv3をお使いの場合はresolverについては今後このテンプレートをお使いいただくことをお勧めします。

その他のカスタマイズ

- ログイン画面のカスタマイズ
  - ロゴの差し替えは簡単
    https://meatwiki.nii.ac.jp/confluence/x/iIooAQ の「ロゴの変更」項目
  - JSP(.jsp)からVelocity template(.vm)に変更になっているので凝ったことをしている場合は移行が面倒
  - .vmはviews/配下にあり変更時にWARファイルの再作成等不要
- SPメタデータ追加
  - https://meatwiki.nii.ac.jp/confluence/x/iIooAQ
    の「ローカルSPメタデータ」項目
- ToU: messages/consent-messages.properties に記述
  https://wiki.shibboleth.net/confluence/display/IDP30/ConsentConfiguration （英語）
- v2向けIdPプラグインによるカスタマイズはどうすればよい？
  - 各プラグインのIdPv3対応版をインストールするのが基本
    - ログインハンドラについてはv2のAPI(RemoteUser/ExternalAuth)を使っているものは移行できそう
  - もしくはIdPv3本体機能で代替する
    - X509ログインハンドラとか

IdPv3関連情報リンク集

- IdPv3インストール方法
  - 新規でShibboleth IdPバージョン3を構築する手順
    https://meatwiki.nii.ac.jp/confluence/x/eIExAQ
  - 既存のShibboleth IdPバージョン2からアップグレードする手順
    https://meatwiki.nii.ac.jp/confluence/x/tYwoAQ
- 実習セミナー活用編（の一部）
  https://meatwiki.nii.ac.jp/confluence/pages/viewpage.action?pageId=12158239
  - 上記アップグレード手順　設定ファイルの整理
  - 送信属性同意機能の有効化
  - FPSP代替
- GakuNinShare:Shibboleth IdP 3
  https://meatwiki.nii.ac.jp/confluence/display/GakuNinShare/Shibboleth+IdP+3
  - 上記に含まれないたくさんの情報

- そのほか，最新の情報は学認情報交換メーリングリストで随時ご案内いたします。
  - 学認情報交換メーリングリストへの参加方法は，こちらをご覧ください。
    https://www.gakunin.jp/ml/

学認はこれからも必要な情報を提供していきます

- 以下のドキュメントの提供を予定しています
  - 特殊なNameID、SAML1フロントチャネル送信、アサーション非暗号化
  - 日本語ロケールリソースの展開
  - sealer.jksの定期更新手順

参考資料: v2からのアップグレードのプランニング

まずは現行環境の確認

- まずは「自機関のIdP環境を確認」しましょう。
  - 実行環境のバージョンは？ （OS, Java, Tomcat）
    - Shibboleth IdP ver.3系の動作環境はJava 7以上，Tomcat 7以上です。
    - バージョンが低い場合は，まずは環境を整備してください。
      https://meatwiki.nii.ac.jp/confluence/x/tYwoAQ
    - OSのEOLも考慮しましょう
  - カスタマイズはしていますか？
    - ログイン画面（ロゴの利用など）
    - 属性送信（SAML1フロントチャネル送信，特殊なNameIDの利用，平文アサーション）
    - ローカルSPなど，学認以外のSPメタデータ読み込み
  - IdPのプラグインは利用していますか？
    - プラグインごとに対処方法が異なります。
      学認が提供しているものも含め、代替を探してください。

実行環境のバージョンの選択と将来設計

- OS / Java / Tomcat
  - OSのEOL考慮
  - OSの標準パッケージの検討
  - 現在の学認技術ガイドは CentOS 6 / OpenJDK 7 / Tomcat 7

- 余談: 将来設計として
  - CentOS 7? Java 8? Tomcat 8? Jetty?
  - スクリプトを使っている場合Java 8は注意！
  - 同時にいろんなバージョンを上げるとトラブル時に困る
    - 証明書更新等も同様

移行後マシンの選択肢

- ①現行マシンを使い続けるか、
  ②新規マシン/VMを用意するか、
  ③VMクローンするか
  - ダウンタイムとトラブル対応を考えると1つ目は避けたい
  - OSをバージョンアップする場合は2つ目の選択肢しかない

IdPインストール方法の選択肢

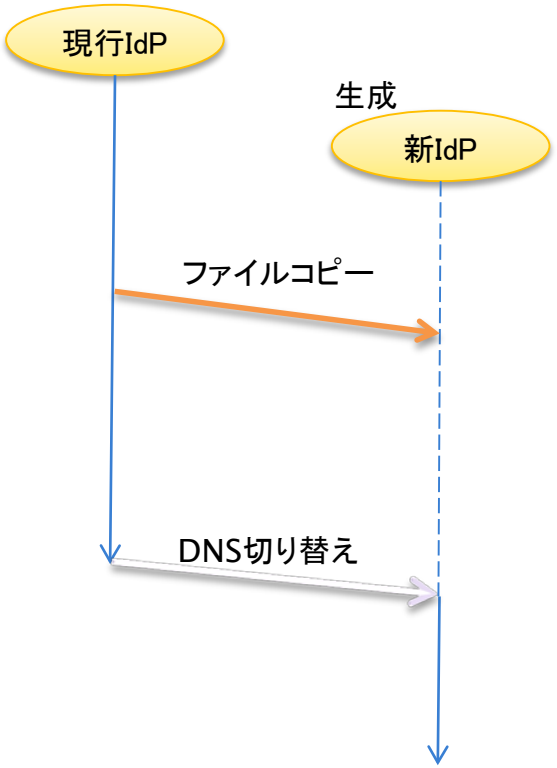
- IdPv3のインストールには、
  「新規インストール」
  「既存のIdPからのアップグレード」
  の2種類があります。
  - いずれにしても同一ホスト名、同一entityID、同一証明書を使うのがお勧めです。

- Pros/Cons
  - 新規インストール
    - クリーン
  - アップグレード
    - 手を入れるところ最小限（？）

- 前頁「②新規マシン/VM」を選択した上で「アップグレード」を選択したいのだがどうすればいい？
  →必要なファイルをマシン間コピーしてから

モデルケース

- 「新規VM」「IdPv3新規インストール」を選択した場合



①VM生成

②新規インストール

③attribute-resolver/filter/証明書等コピー

④テスト(SAML2)
※ブラウザのhostsで新IdPを設定しておく

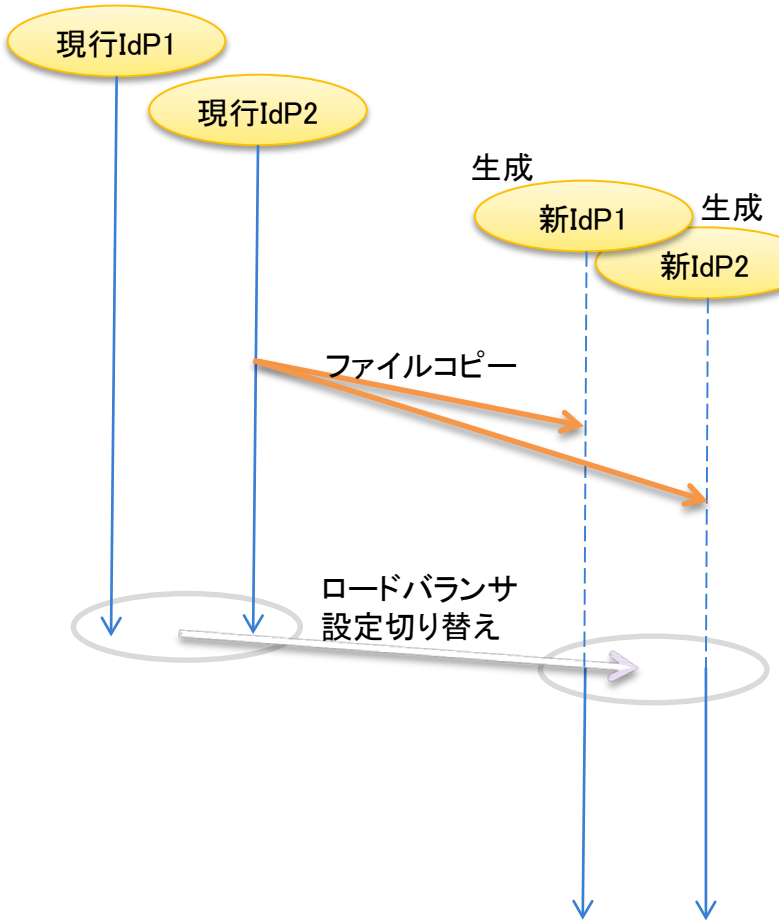
⑤DNS切り替え

⑥再度テスト

※バックエンドにDBがある場合は工夫が必要

冗長化モデルケース

- 「新規VM」「IdPv3新規インストール」を選択した場合



①VM生成

②新規インストール

③attribute-resolver/filter/証明書等コピー

④テスト(SAML2)
※ロードバランサ配下にテスト環境としておいた上でブラウザのhostsで新IdPを設定しておく

⑤新IdP群を本番にするようロードバランサ設定切り替え

⑥再度テスト