OpenSSO社内勉強会第二回
- SAML -

OSSTech

オープンソース・ソリューション・テクノロジ株式会社
2009/12/1
野村健太郎

目次

- 概要
  - SAMLとは
  - SSOの全体像
  - SAMLによるSSO実現のための準備
  - SSOの開始
- 詳細
  - SAMLの構成要素
  - SAMLアサーション
  - SAMLプロトコル
  - SAMLバインディング
  - デモ
  - その他

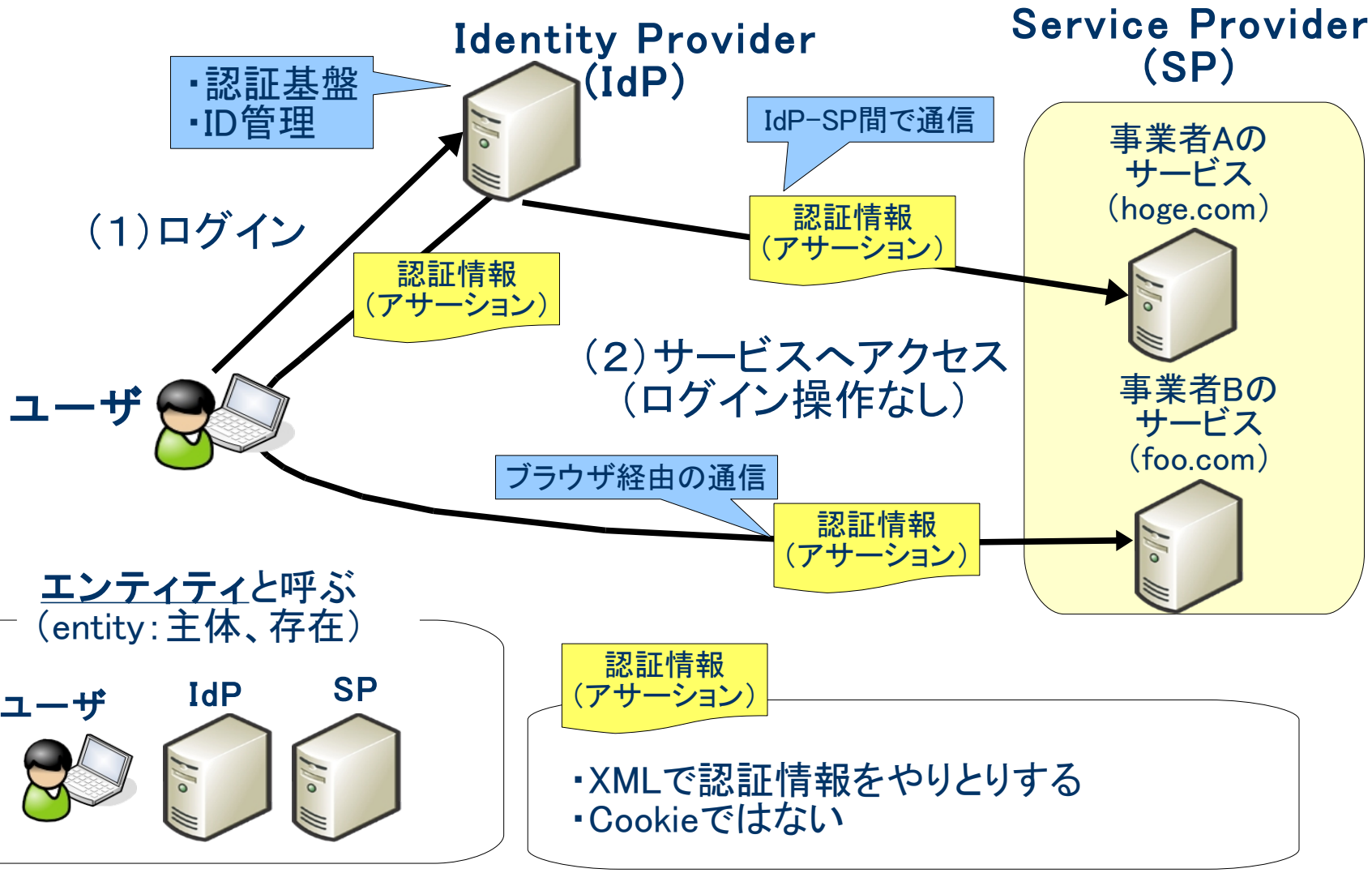
概要 - SAMLとは？

- SAML：Secure Assertion Markup Lauguage
- 認証、認可、ユーザ属性情報などをXMLで送受信するための**フレームワーク**（「Markup Lauguage」だが、言語だけの規約ではない）
  - 公式サイトより：「*XML-based framework for communicating user authentication, entitlement, and attribute information.*」
- 「認証情報」を、「どんなフォーマット（XML）」で、「どの通信プロトコルを使って」送受信するか規定する
- 最新バージョンがSAML2.0

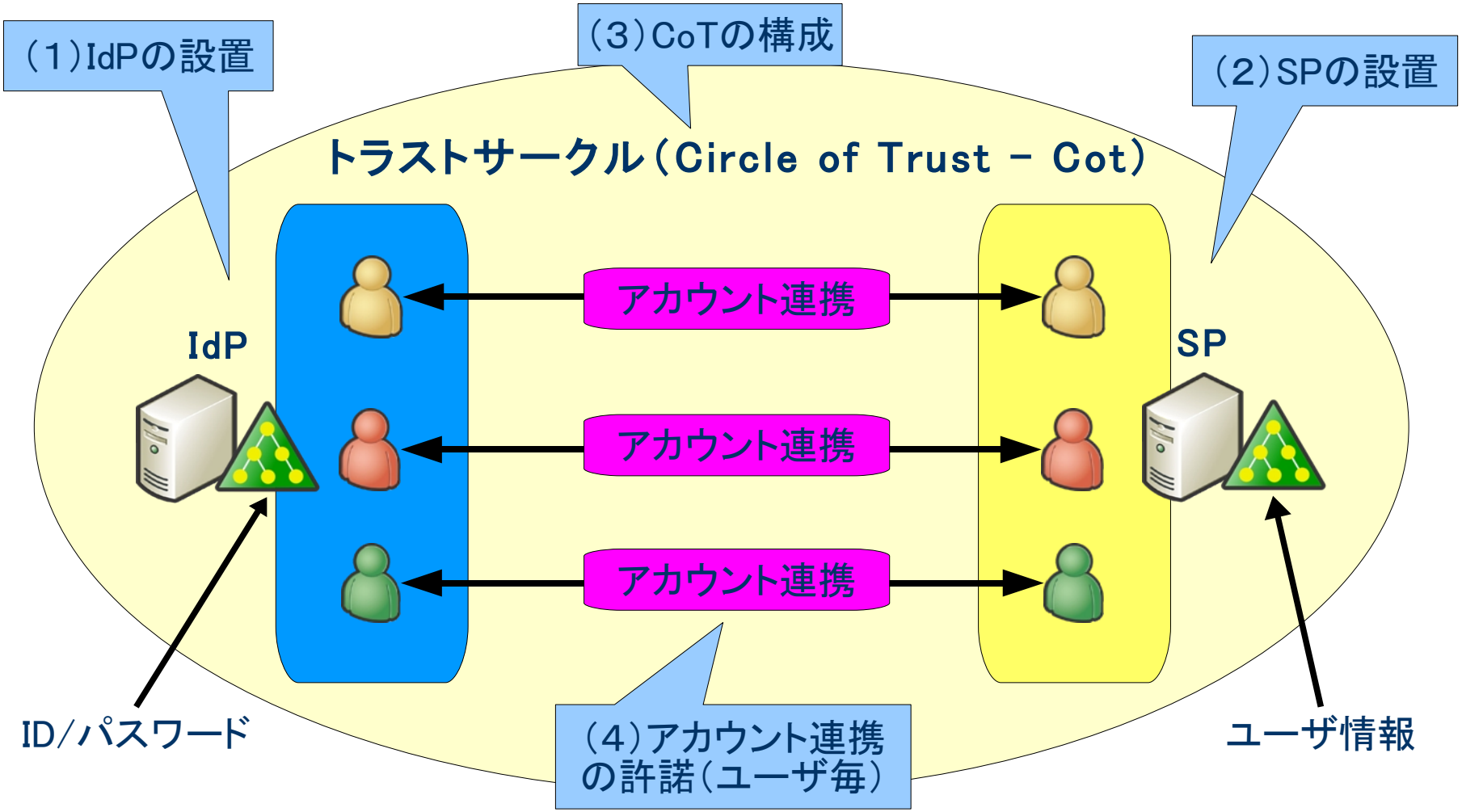
概要 - SAML仕様の原文

- http://www.oasis-open.org/specs/index.php から入手可能
  - saml-authn-context-2.0-os.pdf
  - saml-bindings-2.0-os.pdf
  - saml-conformance-2.0-os.pdf
  - saml-core-2.0-os.pdf
  - saml-glossary-2.0-os.pdf
  - saml-metadata-2.0-os.pdf
  - saml-profiles-2.0-os.pdf
  - saml-sec-consider-2.0-os.pdf
- それぞれが数十ページ。なんとか挫折せずに読める文章量…

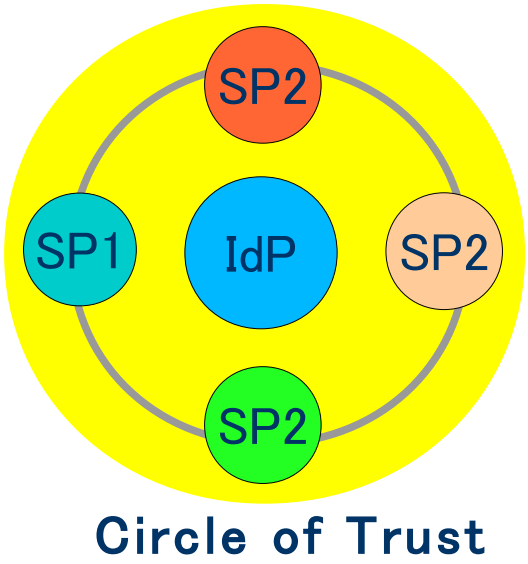
概要 - SAMLでのSSO全体像



概要 - SAMLでSSOを実現するための準備



概要 - トラストサークル（Circle of Trust - CoT）



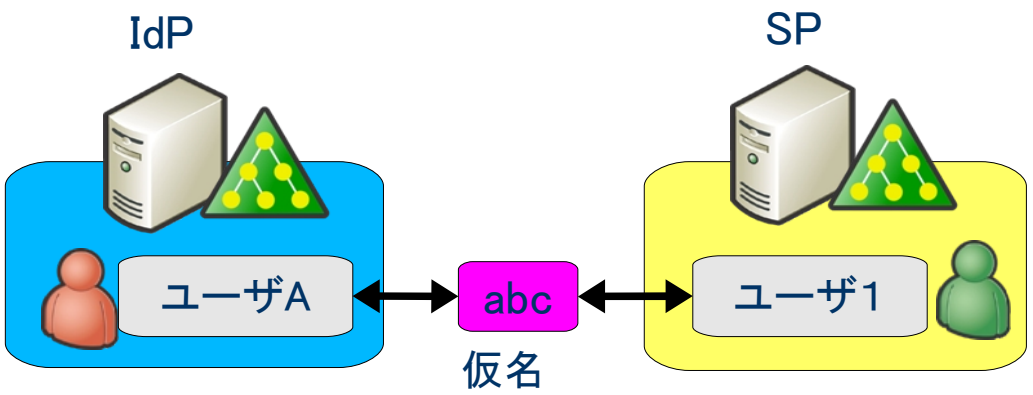
Circle of Trust

- 信頼の輪
- CoT内のSPに対してのみSSO可能
- IdP-SP間でお互いを事前に登録し、CoTを構成しておく必要がある
- 一つのCoT内に複数のIdPが存在することもある

概要 - アカウント連携(1)

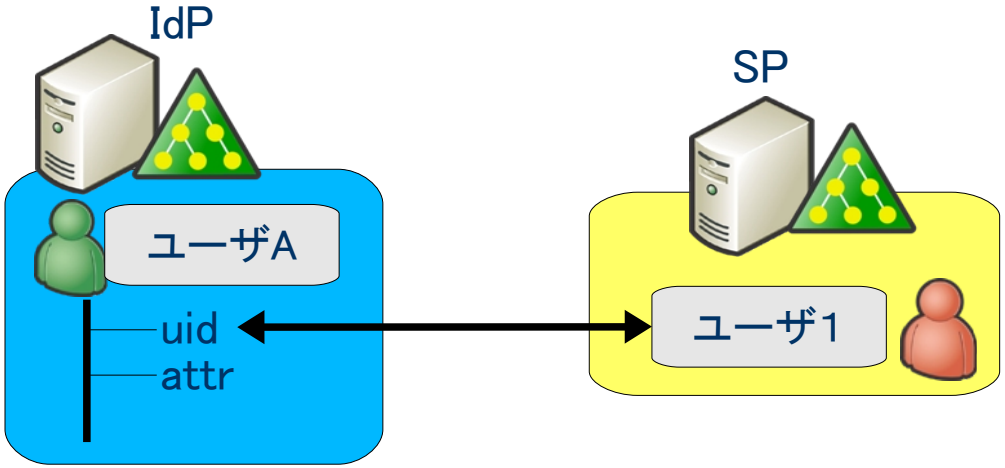
- IdPのアカウントとSPのアカウントを紐付ける
- NameIDというユーザ識別子をIdPとSP間で共有することで実現する
- NameIDには以下のものが使用される
  - メールアドレス
  - X.509のSubject
  - ユーザ属性情報(ユーザIDなど。Google Appsはこのタイプ)
  - 仮名:ランダムな文字列によるユーザ識別

概要 - アカウント連携(2)



仮名による連携

- IdPのアカウントとSPのアカウントを仮名(仮IDのようなもの)で紐付ける
- IdP/SP内のアカウント情報(ユーザIDなど)を隠蔽したままアカウント連携可能
- ユーザ毎に設定する:初回のみ、IdPとSPにそれぞれのID/パスワードでログインする必要あり

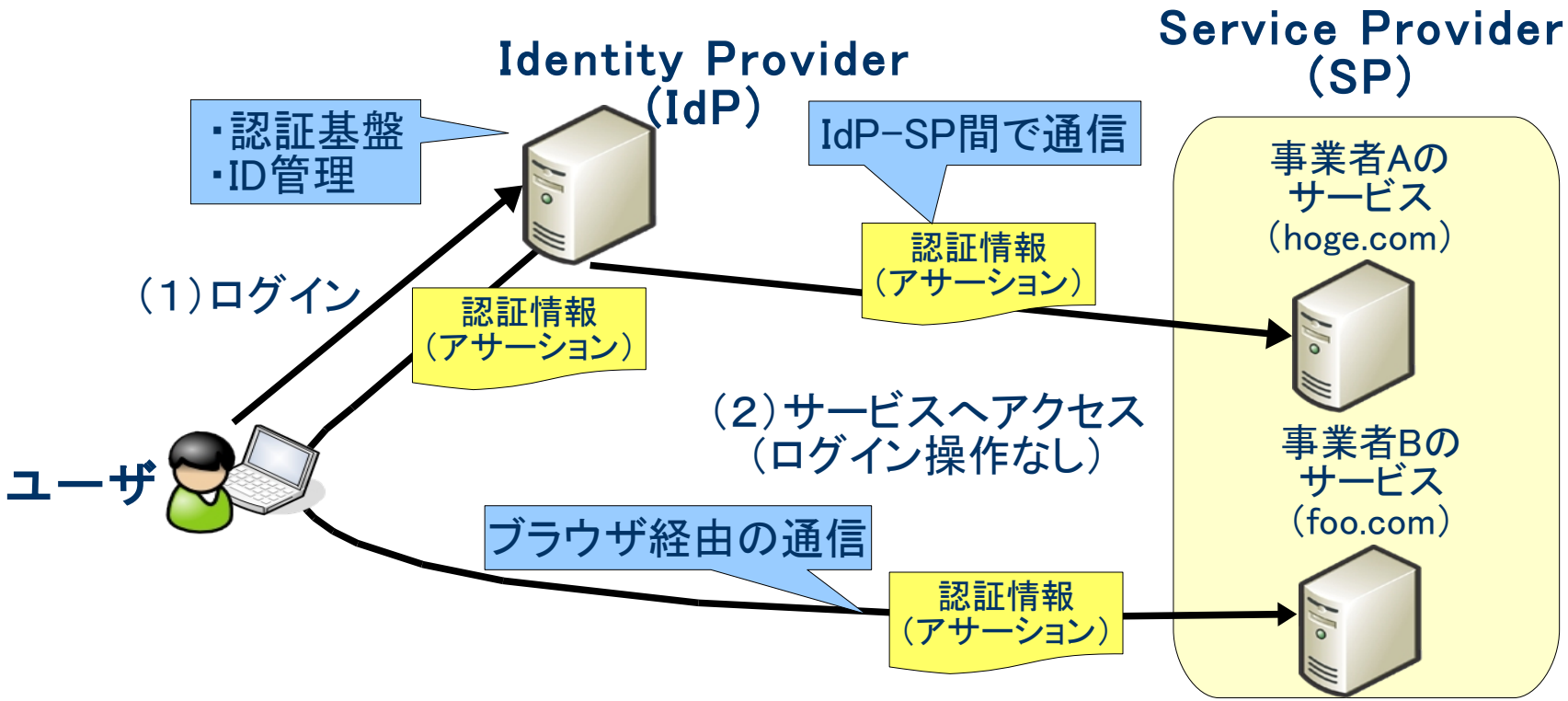


ユーザ属性情報による連携

- IdPのアカウントとSPのアカウントをユーザ属性で直接連携
- 自システム内の情報の一部を相手に知らせる必要がある
- Google Apps はこの方式

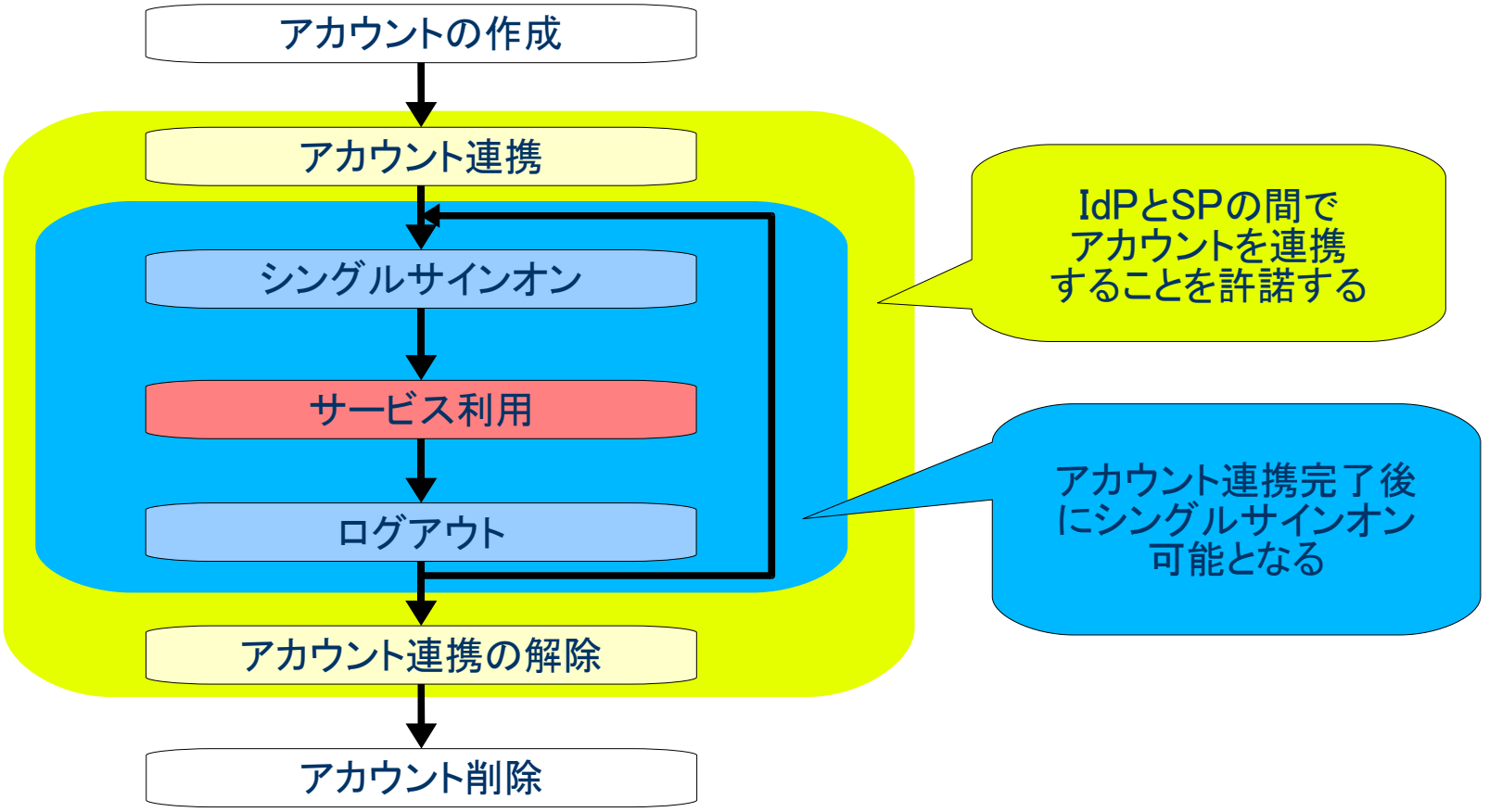
概要 - 認証連携（SSO）

- CoTの構成・アカウント連携が完了して、SSO可能になる

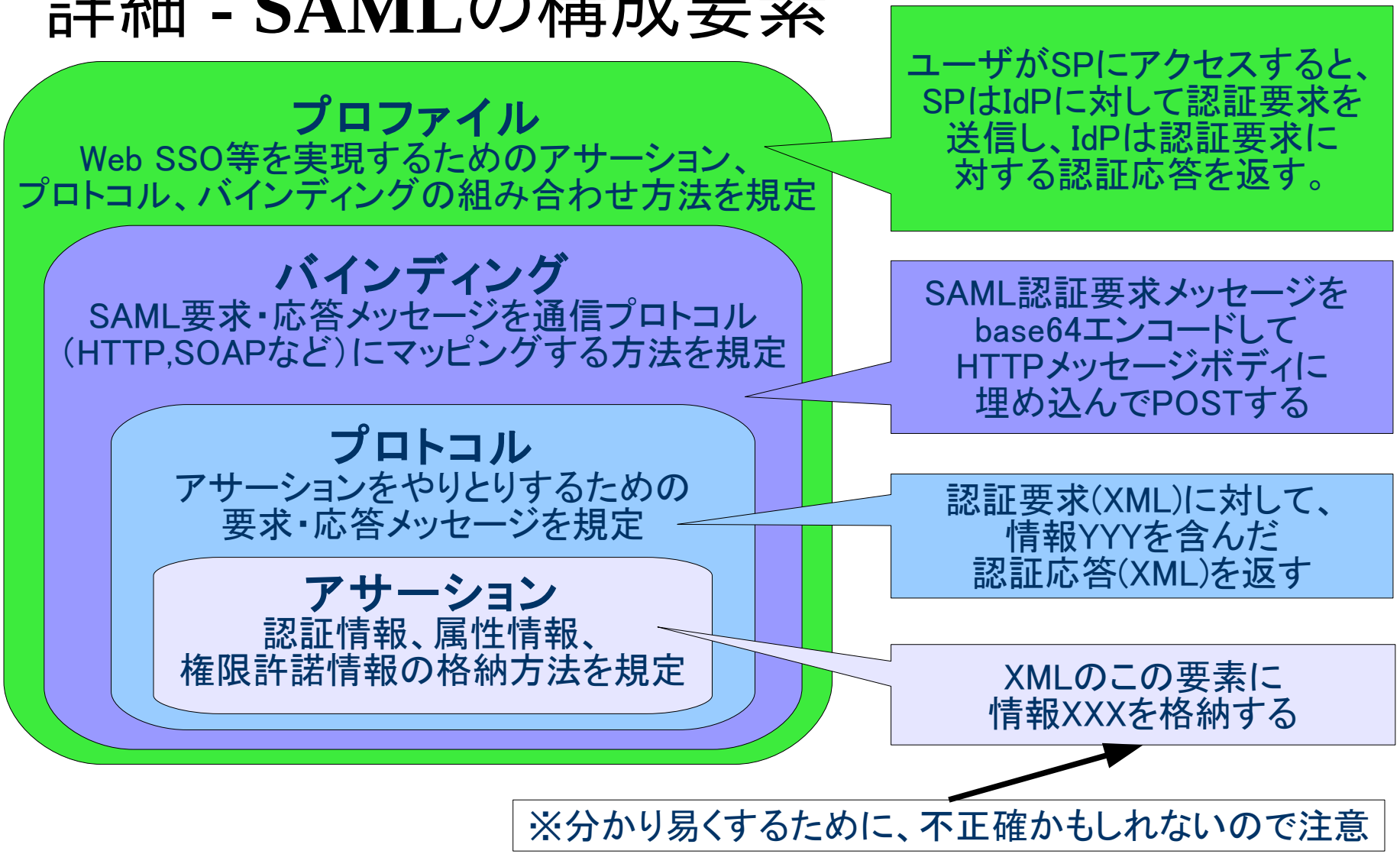


概要 - アカウントのライフサイクル

- アカウント作成～SSOの利用～アカウント削除のサイクル



詳細 - SAMLの構成要素



詳細 - アサーション

- IdPが発行するユーザに関する証明情報のXML

```
<saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" Version="2.0"
ID="s2907181983bc6f588aeb045fca183d671224506ec" IssueInstant="2009-11-18T08:28:09Z">

    アサーション発行者
    アサーションのデジタル署名
    アサーションの利用条件
    ユーザ識別子(NameID)

</saml:Assertion>
```

詳細 – SAMLプロトコル

- 認証要求（AuthnRequest）
  - SPがIdPに対して、ユーザの認証情報を要求する

```
<samlp:AuthnRequest ID="xxx" Version="2.0"
Destination="http://idp.osstech.co.jp/idp/sso">
    認証要求情報が入る
</samlp:AuthnRequest>
```

- 認証応答（Response）
  - IdPがSPにユーザの認証情報（アサーション）を送付する

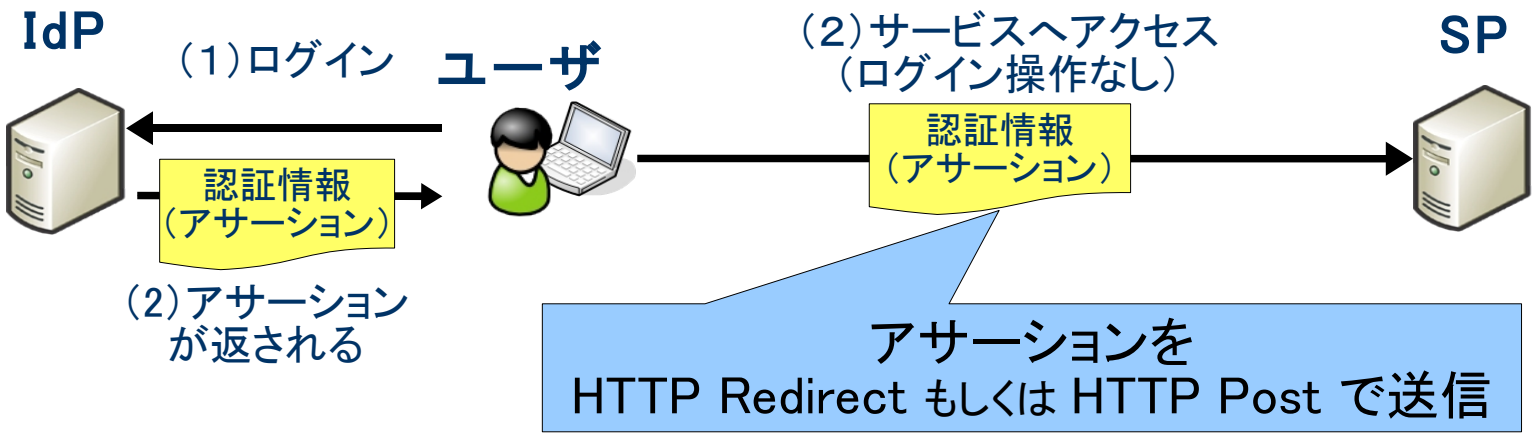
```
<samlp:Response ID="xxx" Version="2.0" Destination"http://sp.osstech.co.jp/sp/sso">
    < saml:Assertion ...>
        アサーションが入る
    </saml:Assertion>
</samlp:AuthnRequest>
```

詳細 - バインディング（Binding）

- SAMLメッセージを既存の通信プロトコル（HTTP、SOAPなど）にマッピングする（埋め込む）方法を規定
- IdP-SP間の通信有無で分けてみる（一番使いそうなやつだけ抜粋）
  - 無：HTTP Redirect、HTTP POSTバインディング
    - HTTP Redirect：Google Apps（認証要求）で利用されている
    - HTTP POST：Google Apps（認証応答）、salesforceで利用されている
  - 有：HTTP Arifactバインディング

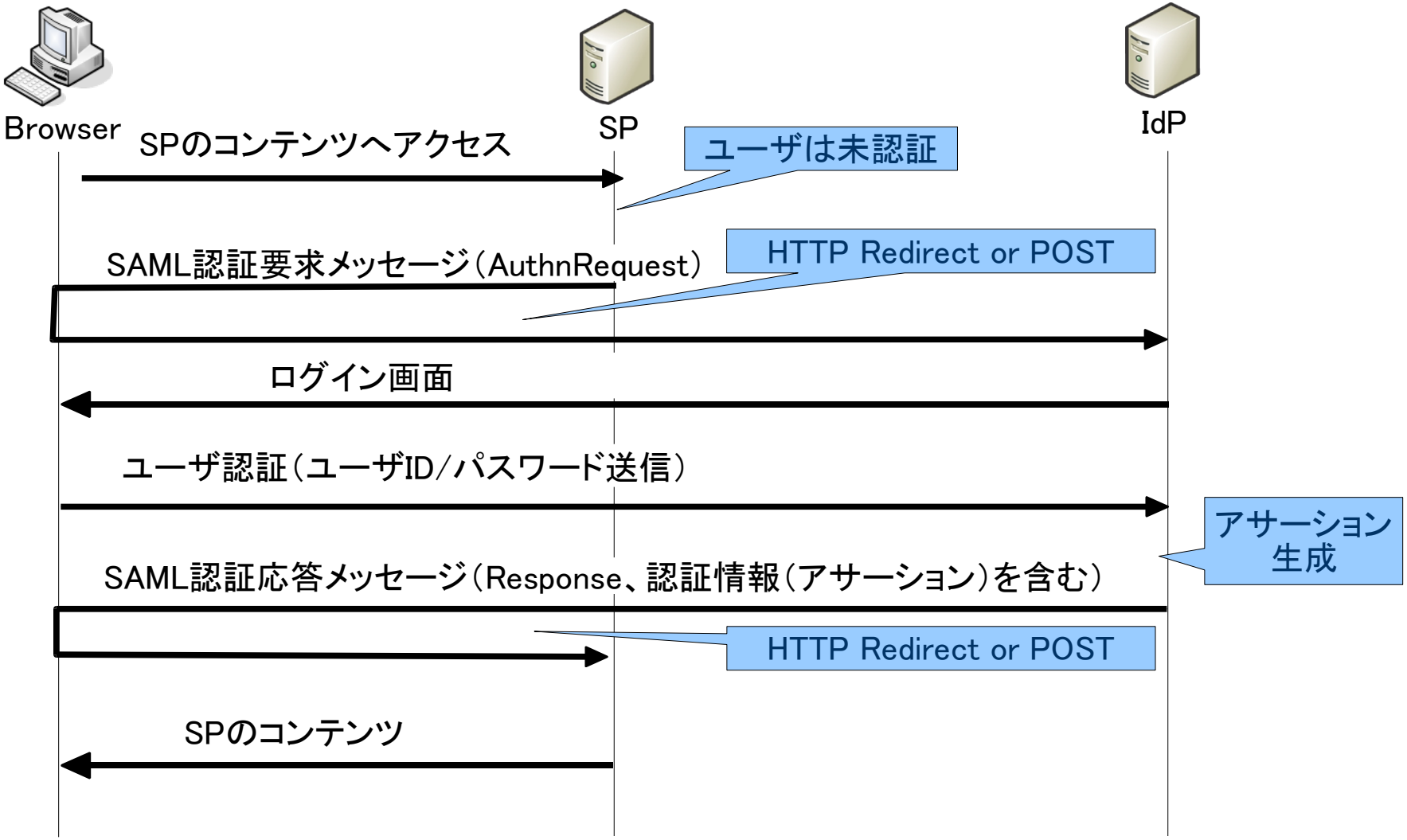
詳細 - HTTP Redirect/POST バインディング

- ブラウザが通信を中継し、IdP-SP間の通信が発生しない



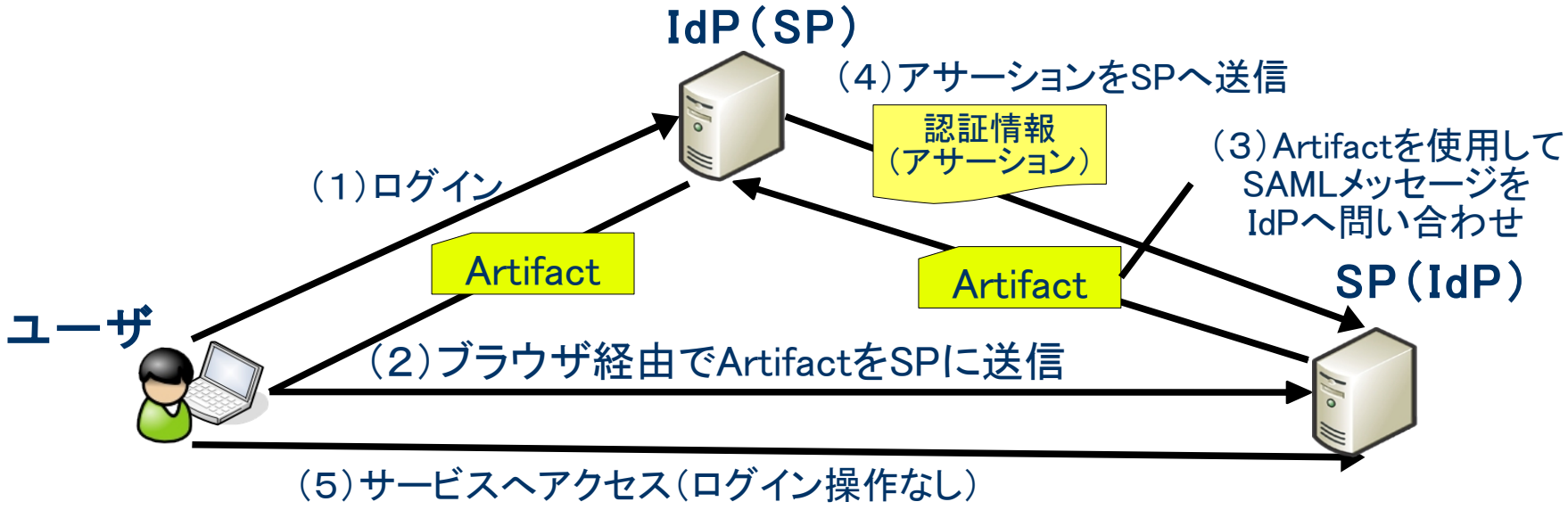
方式	説明	特徴
HTTP Redirect	SAMLのメッセージをBase64エンコードしてURLパラメータに載せてGETメソッドで送信(HTTP 302 を利用)。	URLが長すぎると、ブラウザのURLの長さ制限に引っかかる可能性がある(IEは2,083文字らしい)
HTTP POST	Base64エンコードしたSAMLメッセージをHTMLフォームに載せてPOSTメソッドで送信。JavaScriptで自動的にPOSTリクエストを送信させるのが一般的？ Google Apps、salesforceはこのタイプ。	IdPへのログイン→SPへの遷移を自動化するには、ブラウザでJavaScriptが実行可能となっている必要がある

詳細 - HTTP Redirect/POST バインディング

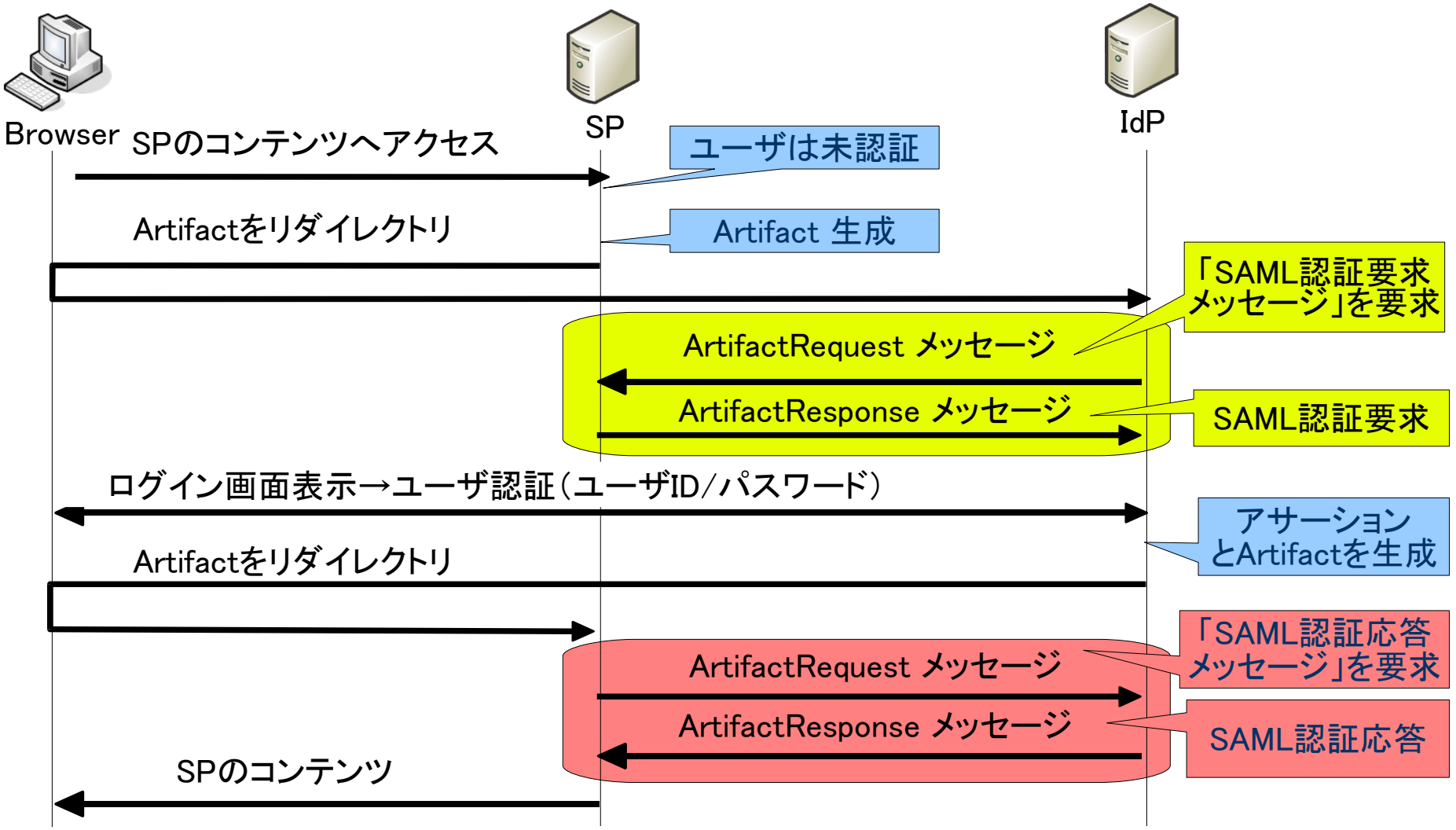


詳細 – HTTP Artifactバインディング

- Artifact：SAMLメッセージ識別用のランダム文字列
- Artifactを利用して、IdP-SP間で直接SAMLメッセージを送受信する（SOAPを利用）
- アサーションがクライアントに渡ることがない



詳細 – HTTP Artifact バインディング

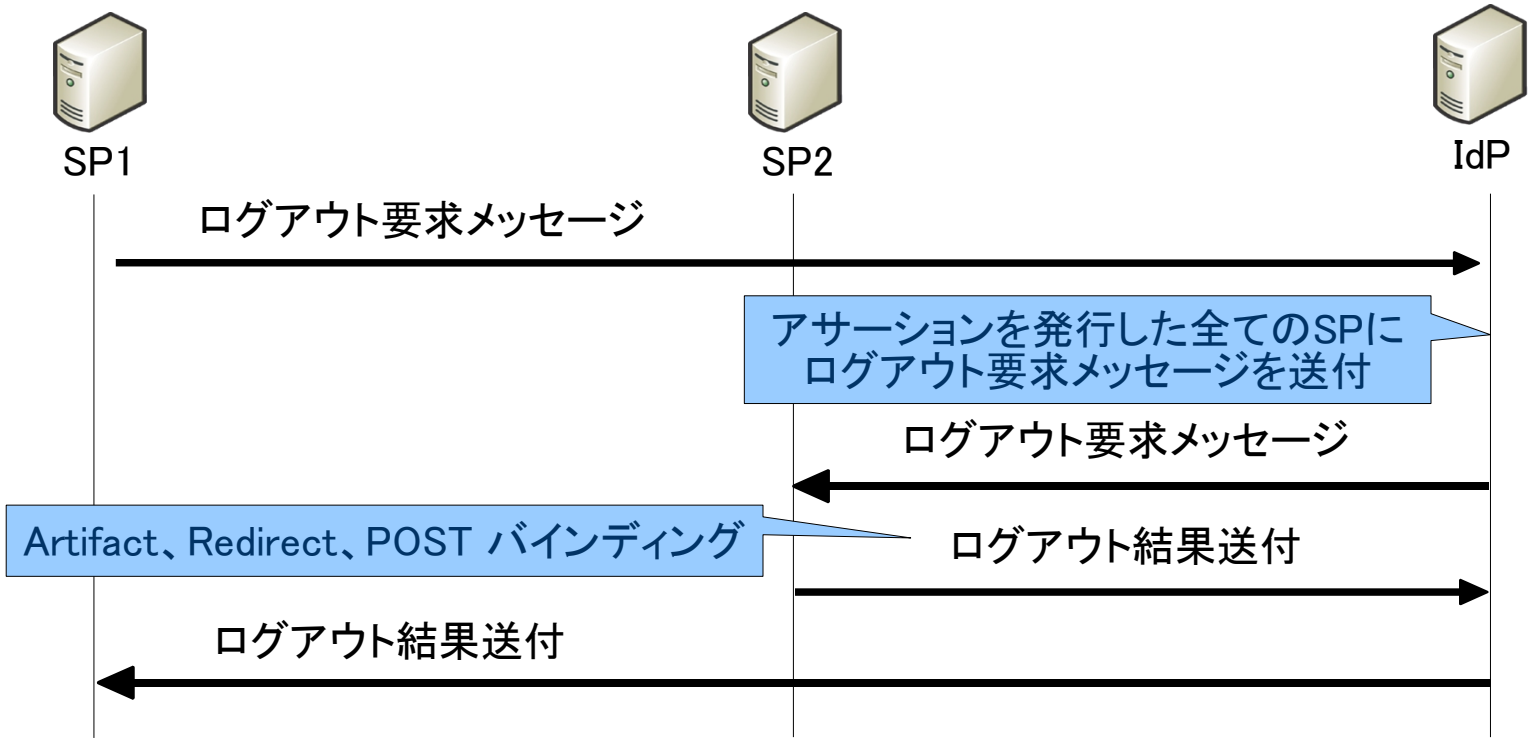


デモ

1. Google AppsのSAML認証がHTTP POSTバインディングで行われる様子を見てみる

2. LDAPに保存されているNameIDを直接いじってみる
   - OpenSSOにはユーザAでログインし、Google Appsにはユーザ Bでログインする

詳細 – シングルログアウトプロファイル



- OpenSSO ＋ Google Apps ＋ Liferay ＋ Alfresco のデモで、Google Apps からログアウトすると Liferay からもログアウトするが、これはSAMLのシングルログアウトではない。

詳細 - メタデータ

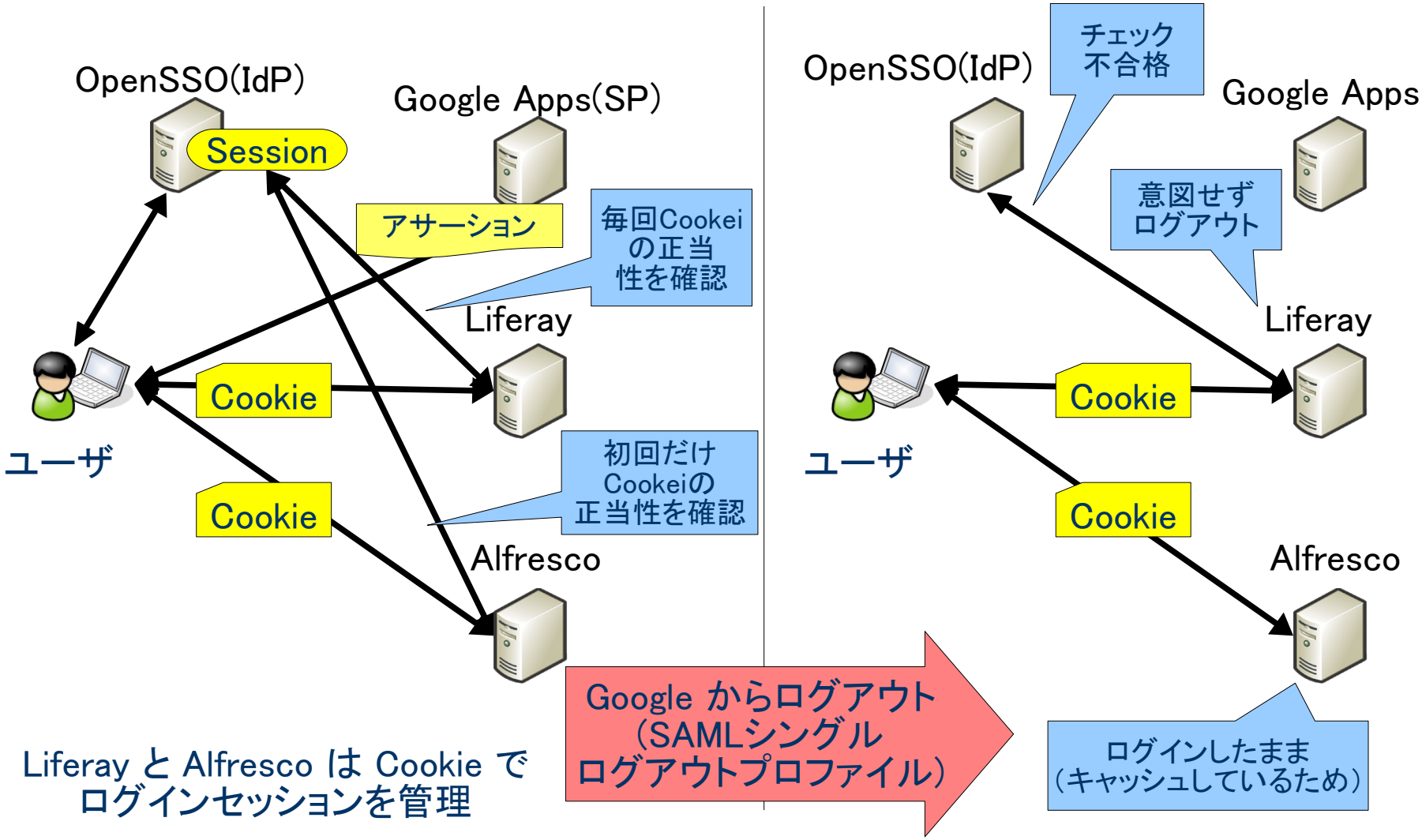
- SAMLでSSOを利用するためには、あらかじめCoTの構成、アカウント連携などを行っておく必要がある。これらの作業に必要な情報をまとめたXMLデータをメタデータという。
- メタデータがあれば、IdP、SPの構築作業が楽になる
- 例：Google Apps のメタデータ↓

```
<EntityDescriptor entityID="google.com" xmlns="urn:oasis:names:tc:SAML:2.0:metadata">
  <SPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">

    <NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</NameIDFormat>

    <AssertionConsumerService index="1"
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
      Location="https://www.google.com/a/ドメイン名/acs" />
  </SPSSODescriptor>
</EntityDescriptor>
```

詳細 - アサーションへのXMLデジタル署名

- アサーションの改竄によるユーザなりすましなどを防ぐために、XMLデジタル署名を付加する
- IdPの証明書をSPに登録しておく必要がある

参考 – Liferay と Alfresco のログアウト



参考文献

- LIBERTY ALLIANCE のセミナー資料。最初に読むならこれがおすすめ（今回の勉強会資料の元ネタ）
  - http://wiki.projectliberty.org/images/9/94/080215_JapanSIG_Technical_Seminar.pdf
- SAML仕様の原文
  - http://www.oasis-open.org/specs/index.php#saml
- 実システムでのSAML
  - Google Apps：
    - http://code.google.com/intl/ja/apis/apps/sso/saml_reference_implementation.html
    - http://code.google.com/intl/ja/apis/apps/articles/shibboleth2.0.html
  - salesforce
    - http://www.salesforce.com/community/crm-best-practices/it-professionals/application-development/sso.jsp