hbstudy#22

クラウド時代の シングルサインオン

オープンソース・ソリューション・テクノロジ株式会社
野村 健太郎
2011/4/16

目次

- 自己紹介
- シングルサインオンとは？ なぜ今シングルサインオン？
- OpenAMの紹介
- シングルサインオンの方式
- SAMLによるシングルサインオン
- ID管理との組み合わせで導入効果倍増！

※プロトコル（SAML）の話が大半なので、眠くなるかも…。
適宜デモをはさんでいきます。

自己紹介

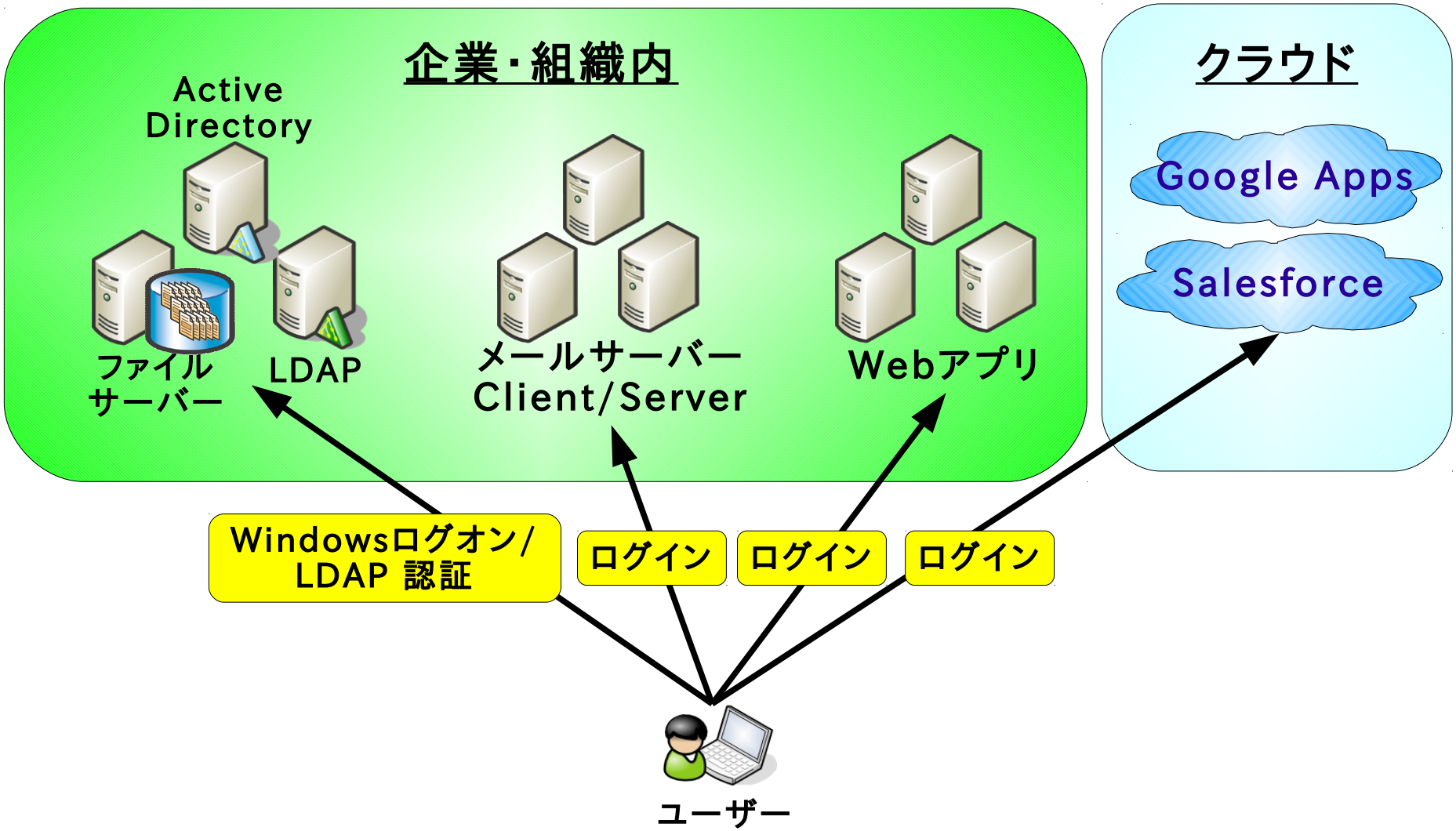
- 野村 健太郎
  - http://d.hatena.ne.jp/nomtech/
  - @nomnux
  - あまり更新してない…
- オープンソース・ソリューション・テクノロジ（株）に勤務
  - http://www.osstech.co.jp/
  - オープンソースや“認証”が得意な人が集まってま
    - OpenLDAP、OpenAM（SSO）、Samba
  - 主に SSO を担当
- Software Design 2010年9月号に統合認証・シングルサインオンの記事を執筆しました。

最初に教えてください

- シングルサインオンシステムを日常的に何かしら使っている
- シングルサインオンシステムを開発・構築している
- OpenAM（OpenSSO）を使ったことがある

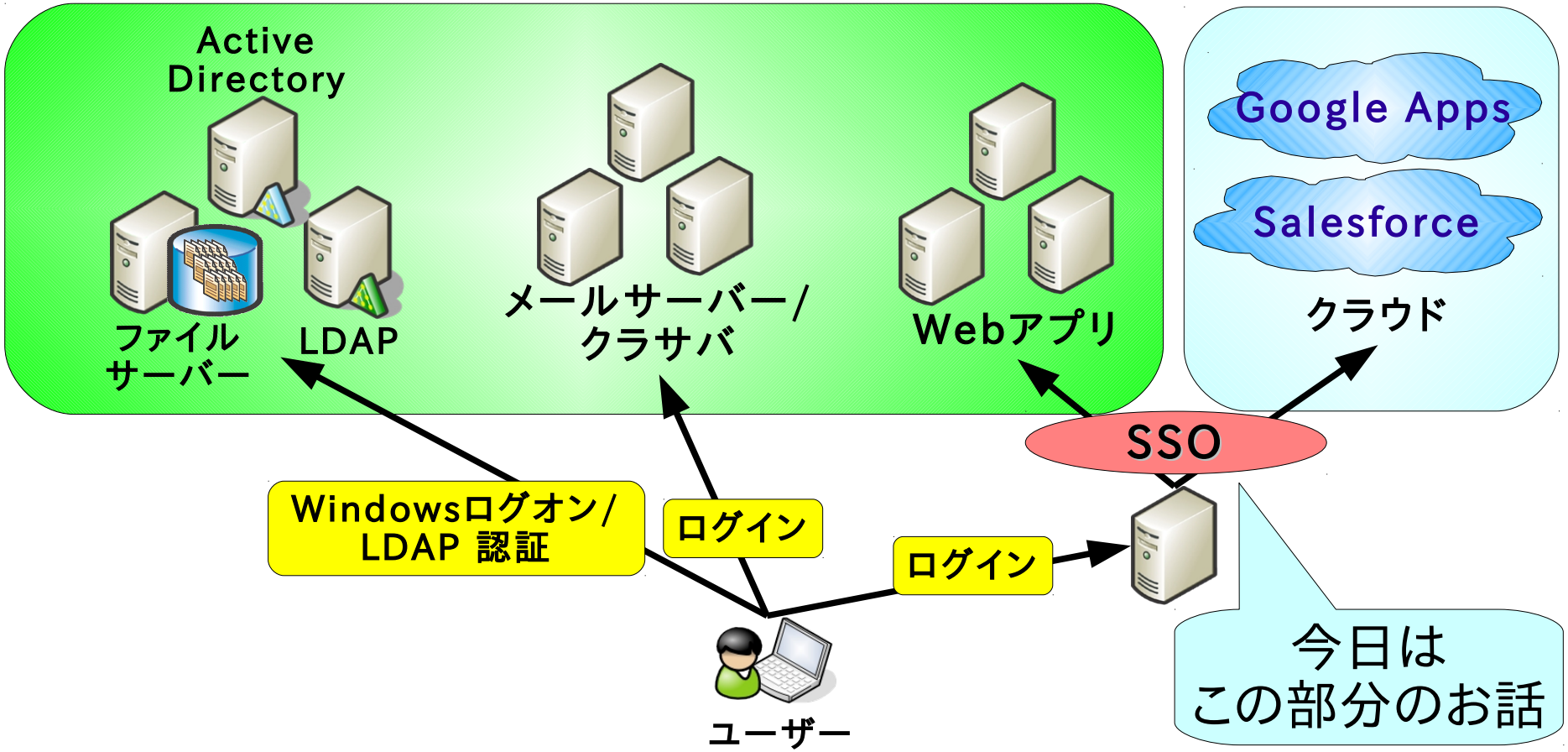
シングルサインオンとは？

システムには必ず必要な「認証」



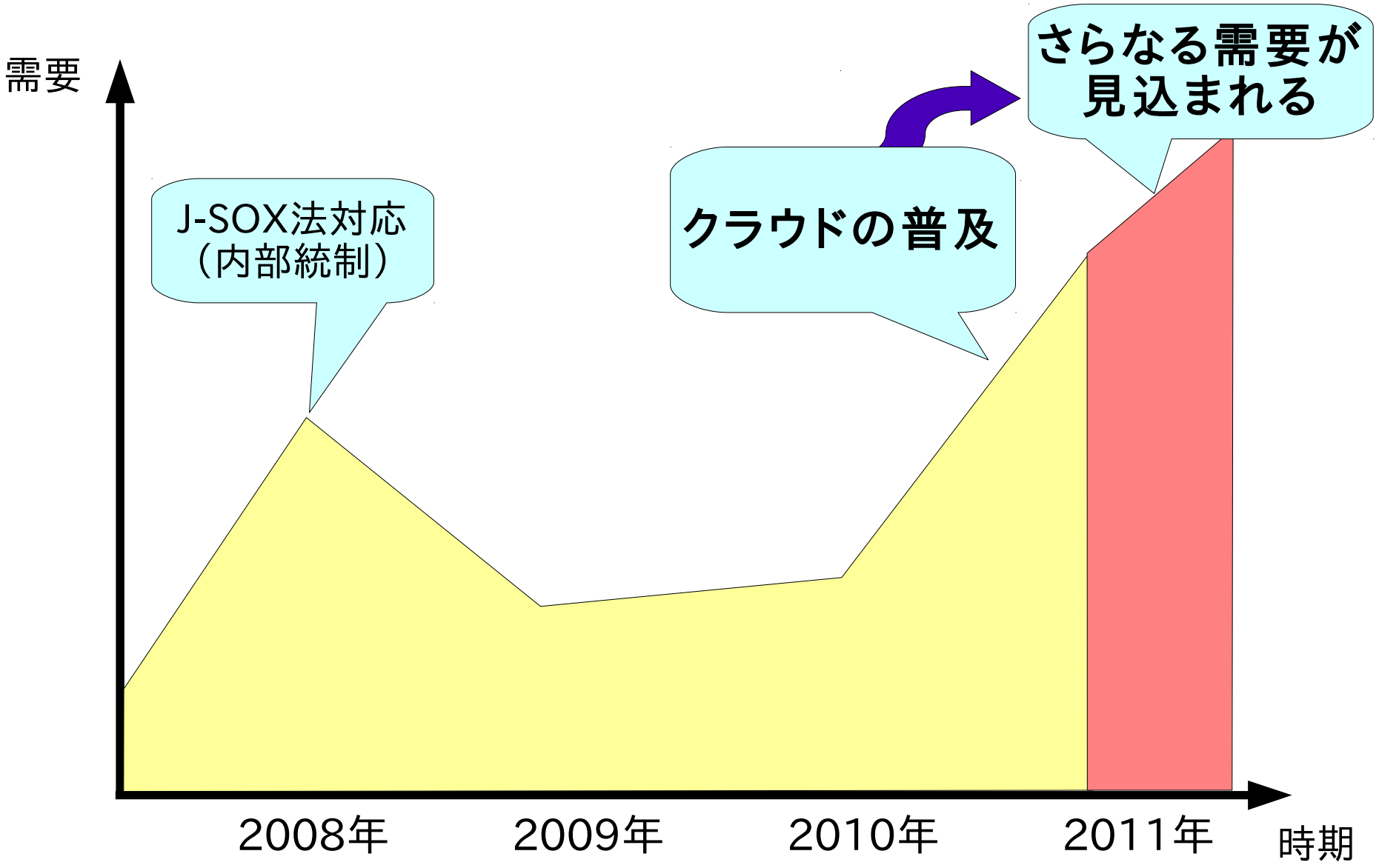
シングルサインオンとは

一度のログイン操作さえ完了すれば、複数のWebアプリケーションに認証操作することなくアクセスすることが可能になる。（以後、SSO と略すことも）



なぜ今シングルサインオン？

SSO（ID管理）の需要推移



今後の需要

- クラウド（外部のWebサービスの業務利用）が普及したことで、ID管理・シングルサインオンの需要が急上昇

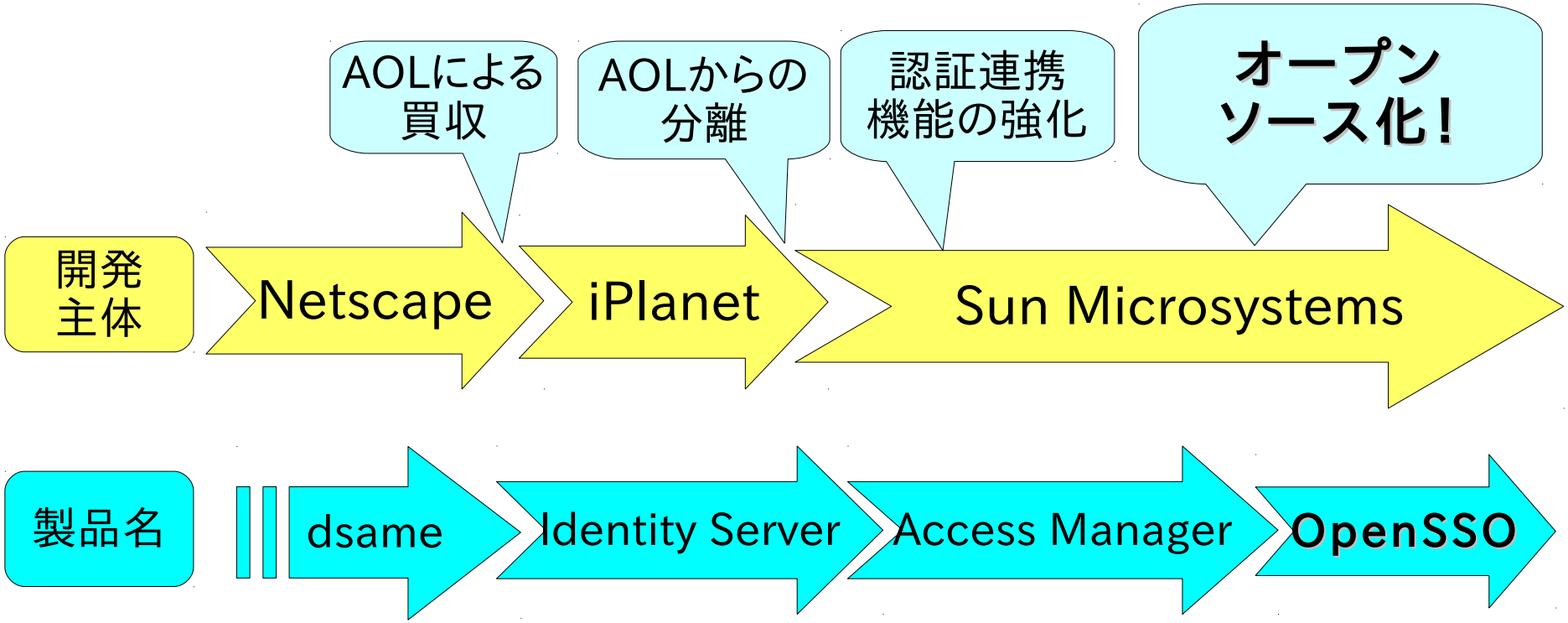
- よくある問い合わせ
  - 社内にある多数ありWebアプリ（オンプレミス）へのアクセスをシングルサインオンで管理し、利便性を向上させたい
  - 社内のWebアプリと**外部のWebサービス（Google Apps、Salesforceなど）をシングルサインオン連携したい**（クラウドサービス利用者）
  - **クラウド基盤の構成コンポーネント**の一つとして、シングルサインオンサービスを提供したい（クラウドサービス提供者）

OpenAM（旧OpenSSO）の紹介

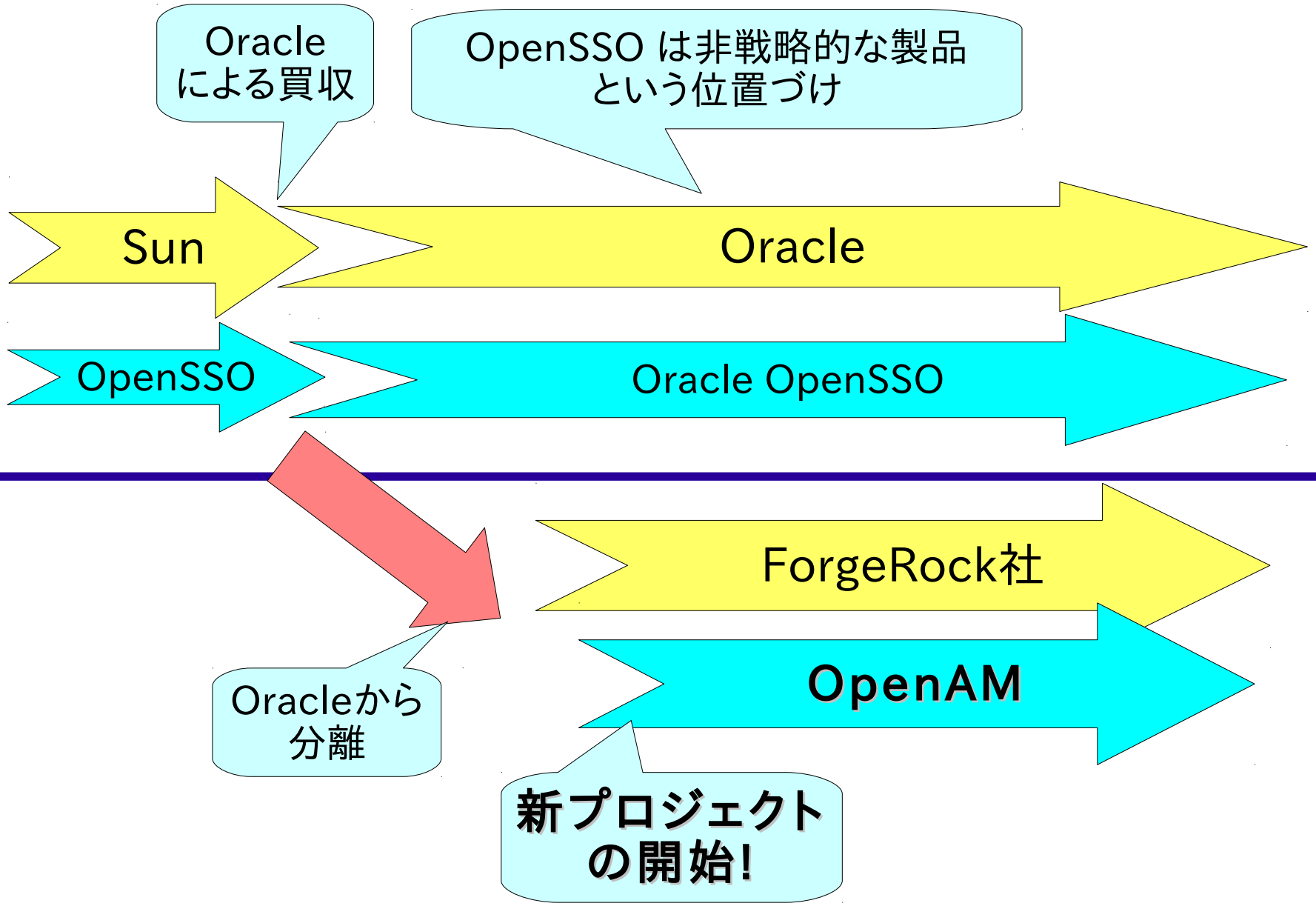
OpenAMとは

- Webアプリケーションにおける**シングルサインオン**を実現するためのプラットフォームとなるソフトウェア
- **SAML、OpenID、OAuth、ID-WSF**などの認証・認可に関連した複数のプロトコルをサポート
- 機能豊富で管理インタフェースも充実（悪く言えば複雑…）

OpenAMの歴史 - その1



OpenAMの歴史 - その2

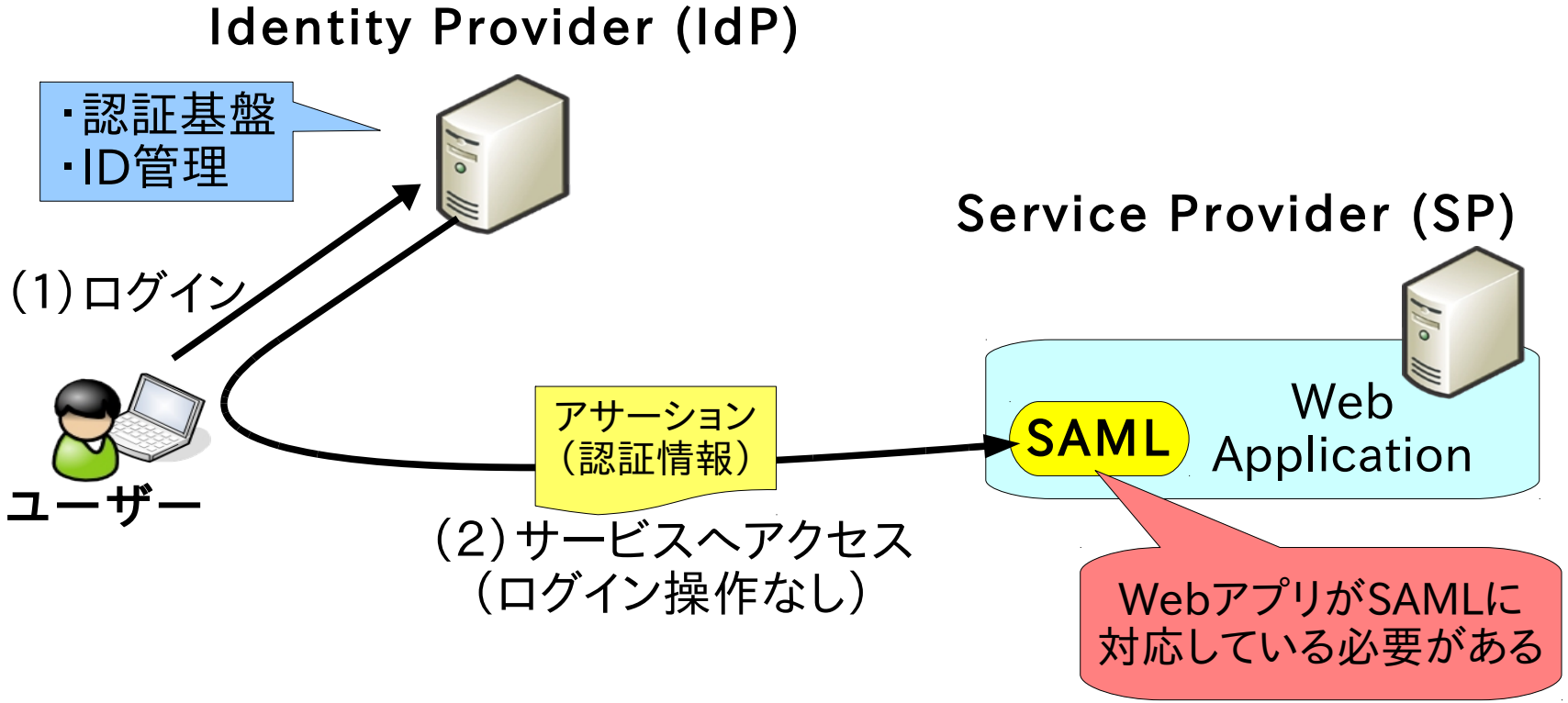


シングルサインオンの方式

※OpenAM を前提とした説明ですが、他の SSO ソフトウェア・SSO 製品でも
だいたい同じような感じのはずです

シングルサインオンの方式（1）

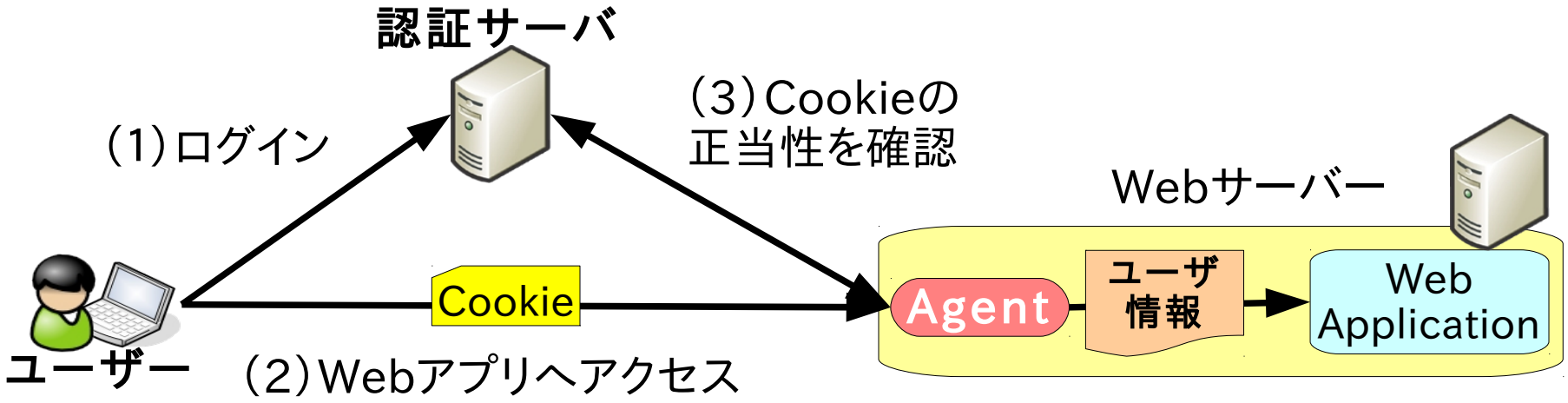
SAML



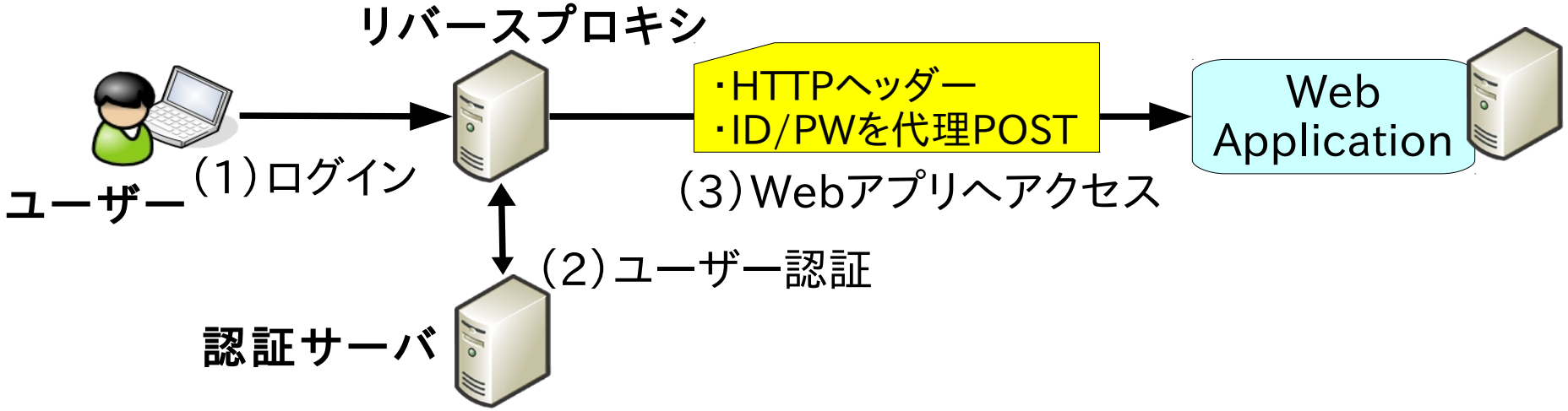
※この図は、HTTP Redirect Binding/HTTP POST Binding の場合の例です。

シングルサインオンの方式（2）

エージェント方式



リバースプロキシ方式

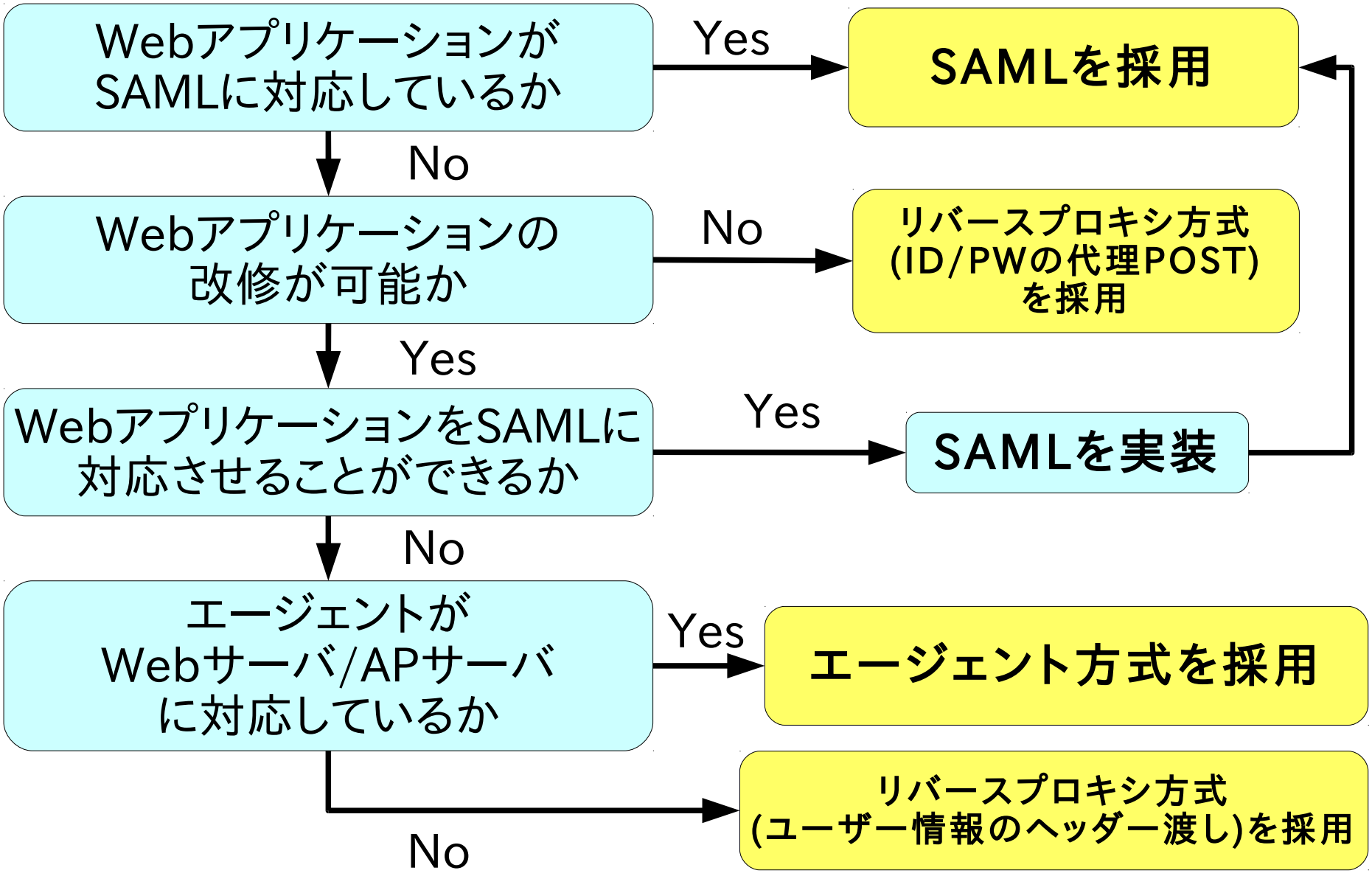


シングルサインオン方式の比較

方式	Webアプリケーションの改修	長所・短所
SAML	SAMLに対応していれば不要	■標準的な仕様に準拠したSSOシステムを構築可能。他製品・サービスとの互換性が高い ■WebアプリケーションがSAMLに対応している必要がある
エージェント方式	必要	■Webアプリケーションへの全ての通信をエージェントがフックする。細かなアクセス制御が可能 ■Webサーバー/APサーバーに対応したエージェントが必要
リバースプロキシ方式	必要/不要	■Webアプリケーションへのアクセスは必ずリバースプロキシを経由する。細かなアクセス制御が可能 ■ユーザー情報をHTTPヘッダーで渡す場合は改修が必要な場合あり ■ID/PWを代理でHTTP POSTする場合は改修不要 ■リバースプロキシがボトルネックになる可能性もある

※OpenAM を前提とした比較内容です

シングルサイオン方式の採用基準



設計上のポイント

ドメインやホスト名は充分吟味して決定する

- Webアプリの SSO においては、ドメインやホスト名は非常に重要（Cookie、SSLなどに影響）
- 一度 SSO 環境を構築すると変更するのが大変

Cookie ドメインの範囲は可能な限り限定する

- SSOの方式によっては、Cookieを複数Webアプリ間で共有することもある
- Cookie ドメインの範囲が広すぎると、やたらブラウザから Cookie が送出されて気持ち悪い…
  - .example.com とか、.example.co.jp とか
- かといって、ホスト Cookie ではSSO環境の構築が困難
- 理想は、シングルサインオン対象のWebアプリ間でのみ有効なドメインとするのがよいかも
  - .sso.example.com、.sso.example.co.jp など
- SAML に関してはこの限りではない（異なるドメイン間におけるSSOを想定している）

SAMLによるシングルサインオン

- 概要 -

SAMLとは

- SAMLとは
  - **S**ecure **A**ssertion **M**arkup **L**auguage
  - 認証、認可、ユーザ属性情報などをXMLで送受信するための仕様
  - 標準的な仕様にしたがって複数のWebサイト間におけるシングルサインオンを実現することが可能
  - Google Apps(SAML SP)、Salesforce(SAML SP/SAML IdP)、学術認証フェデレーションなどが採用
  - 公式サイト：http://saml.xml.org/
  - 仕様原文：http://www.oasis-open.org/specs/index.php#saml

抽象的でよくわからん…

独断と偏見により要約すると、

Webアプリにおける”認証処理”を、外部のWebアプリで代わりにやってもらうための仕組み

とひとまず覚えてくださいm(_ _)m

何が違う?「認証」と「認可」

- **認証(Authentication)**
  - 本人性を確認する
  - ID/パスワード認証、生体認証、ワンタイムパスワード認証
- **認可(Authorization)**
  - あるリソースへアクセスするための権限を与える(認証後のアクセス制御)

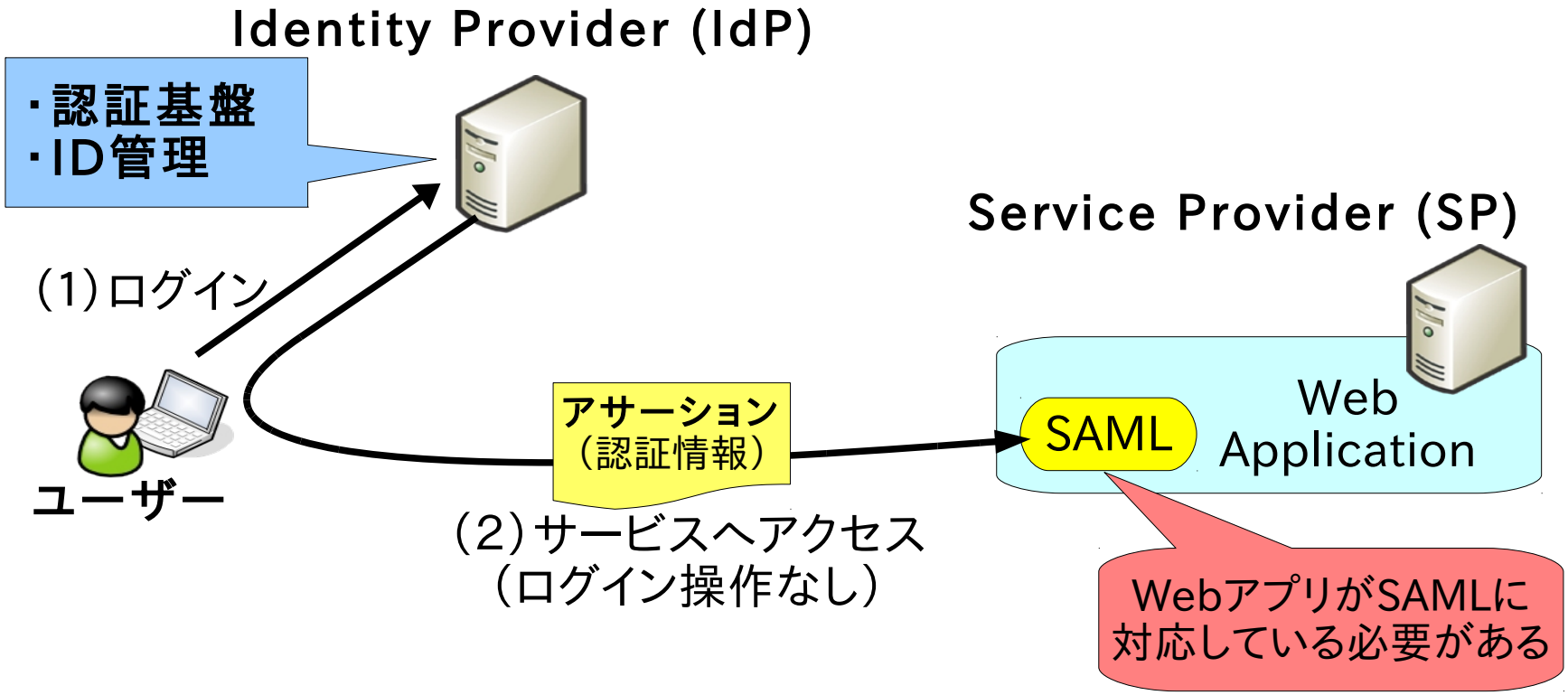
※SAMLでは認可に関連した仕様も定められている。
今日は認証関連のお話

SAMLキーワード

- **Identity Provider（IdP）**：認証・認可の情報を提供する役割を担う。IdPで認証されたユーザーは SP のサービスにアクセスできるようになる。
- **Service Provider（SP）**：シングルサインオン対象の Web アプリケーションなどを意味する。IdP が発行した認証・認可の情報に応じてクライアントにサービスを提供する。
- **アサーション**：・IdPが発行する認証・認可の情報。
- **トラストサークル（Circle Of Trust）**：IdP と SP の間で結ばれた信頼関係を意味する。シングルサインオンを実現するためには、IdP と SP との間で事前に信頼関係を結んでおく必要がある。
- **アカウント連携**：IdP と SP の間でユーザーアカウントを紐付けることを意味する。IdP と SP は信頼関係を結んだ後、アカウント連携を行う必要がある。
- **Federation**：「連携」の意味。SAML、OpenIDなどの認証・認可に関わるプロトコルやその仕組みの総称として使われることがある

※同じ言葉でも、他のプロトコルでは意味が違うことがあるので注意

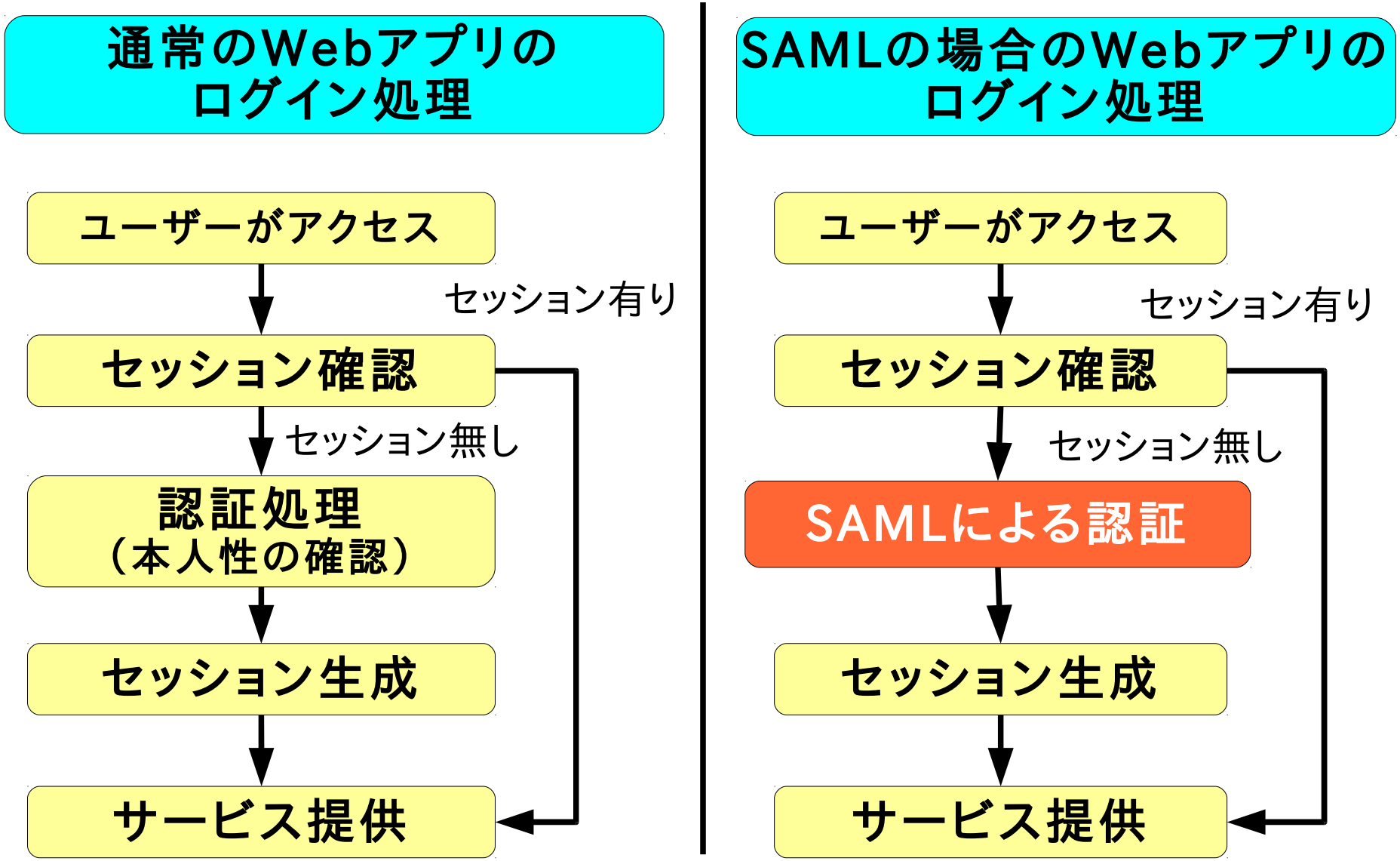
SAMLによるシングルサインオン



**基本的に、セッションはユーザーとIdP間、ユーザーとSP間
それぞれで管理**

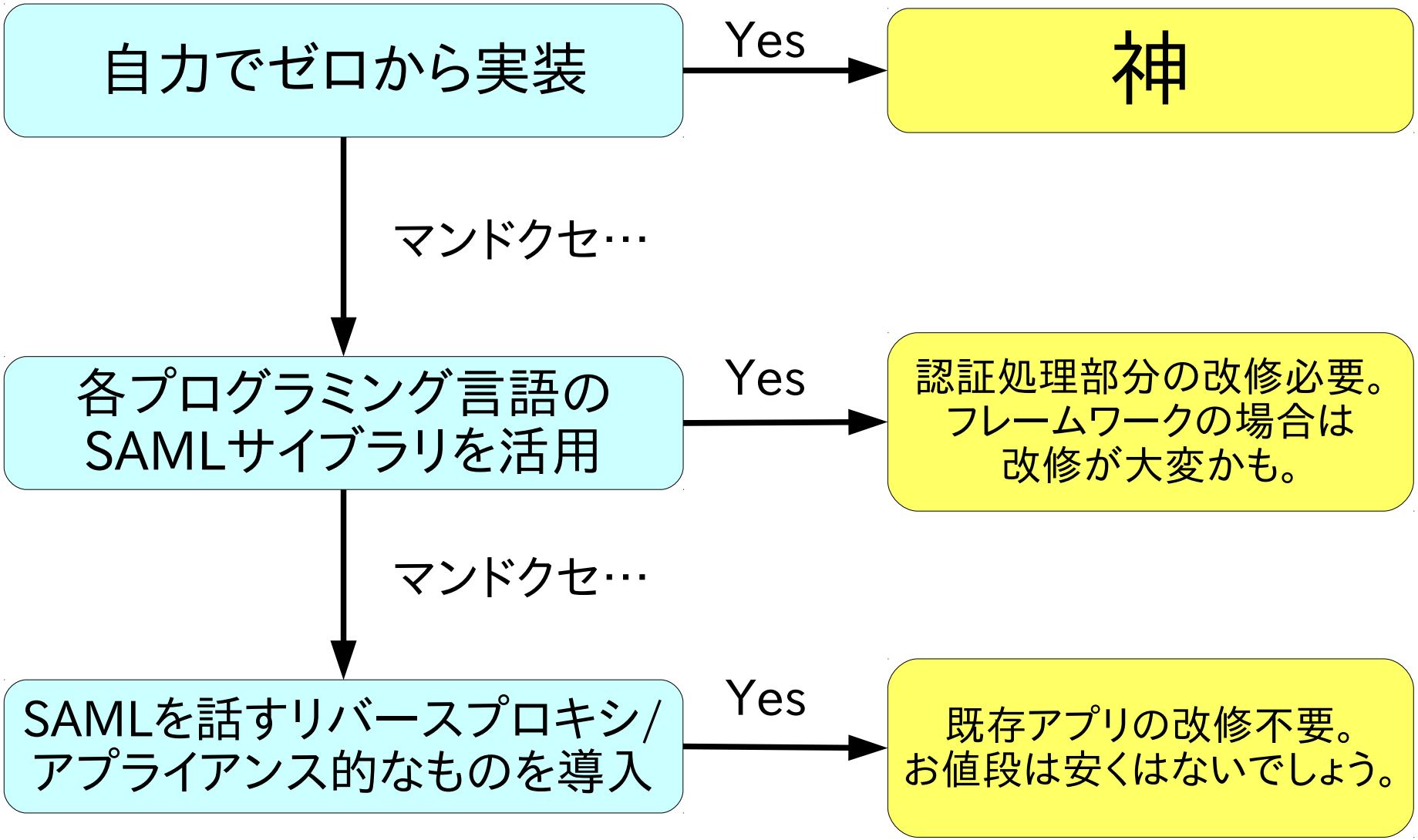
※この図は、HTTP Redirect Binding/HTTP POST Binding の場合の例です。

どこで SAML を使うか？



※Cookie を利用したセッション管理を行なう Web アプリの場合の例

WebアプリのSAML対応(SAML SP化)

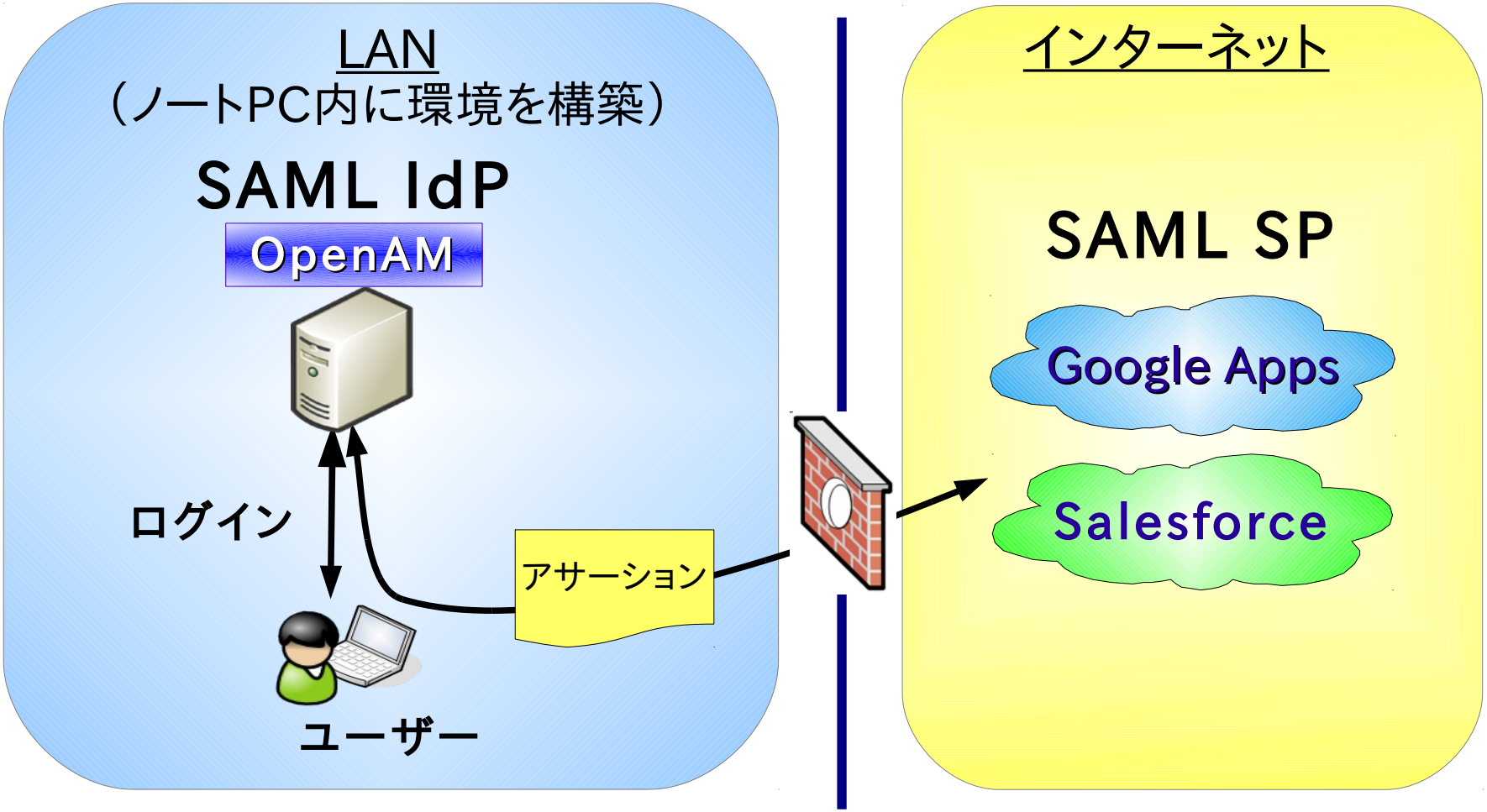


参考 - その他の認証・認可のプロトコル

- **認証（Authentication）**：本人性を確認する
- **認可（Authorization）**：あるリソースへアクセスするための権限を与える

プロトコル	役割	特徴
SAML	認証 認可	■B to B での採用が多い ■Google Apps、Salesforce、学術認証フェデレーション(Shibboleth) ■属性情報の提供方法なども定義されている ■実際のWebサービスなどでは、ほとんどが認証用途で使われている
OpenID	認証	■B to C での採用が多い ■Google、Yahoo!、mixi、はてな、livedoor、ATND ■属性情報の提供方法なども定義されている
OAuth	認可	■Twitter、Facebook、Google ■Webアプリ間でユーザー情報を共有する際に、ユーザー自身が、情報が共有されることを”認可”する ■WebAPI(REST)へのアクセス制御などに利用

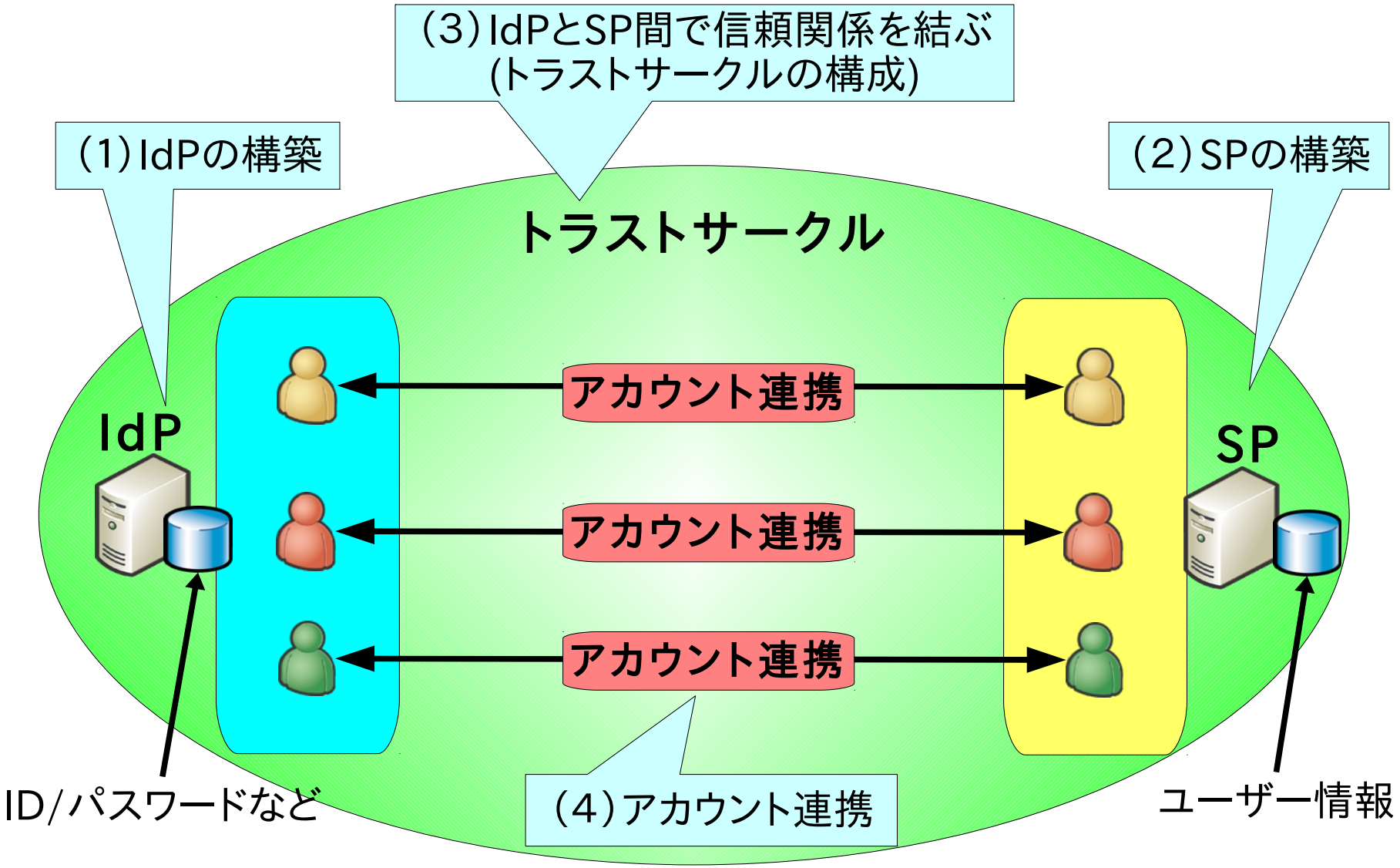
実際にSAMLでシングルサインオンしてみる



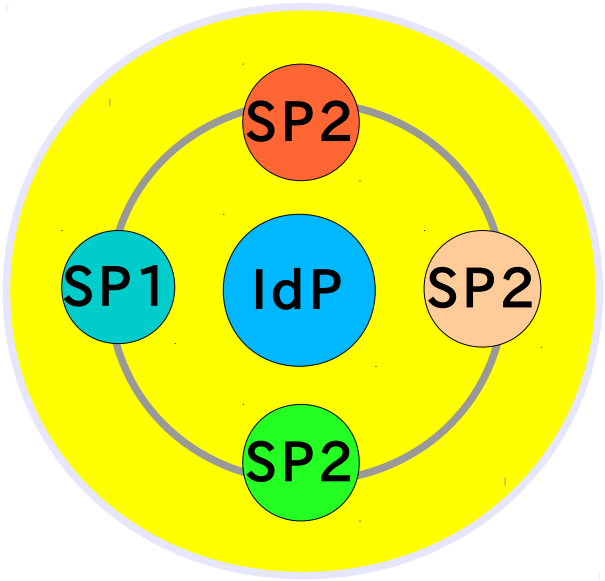
SAMLによるシングルサインオン

- シングルサインオン環境の構築 -

SAMLによるシングルサインオン環境の構築



SAML - トラストサークル



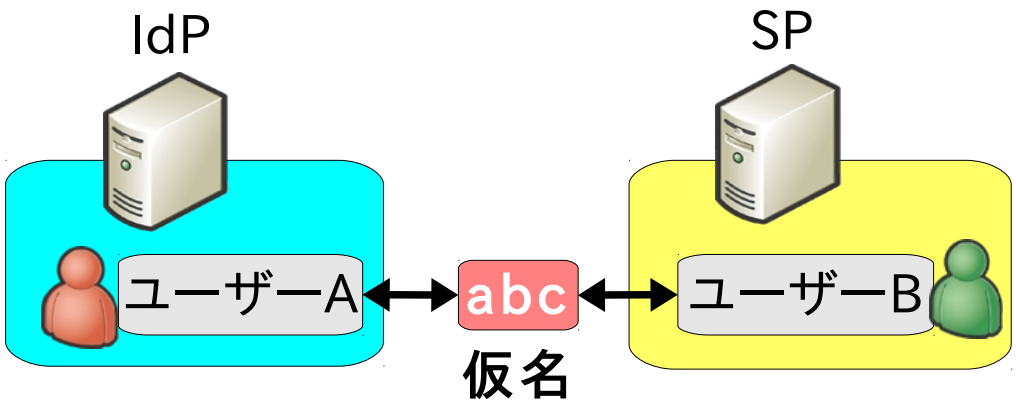
トラストサークル

- **信頼の輪**を意味する（Circle Of Trust - CoT）
- トラストサークル内の SP に対してのみ SSO 可能
- IdP-SP 間でお互いを事前に登録し、トラストサークルを構成しておく必要がある
- お互いの証明書を交換する
- 一つのトラストサークル内に複数の IdP が存在することもある

SAML - アカウント連携(1)

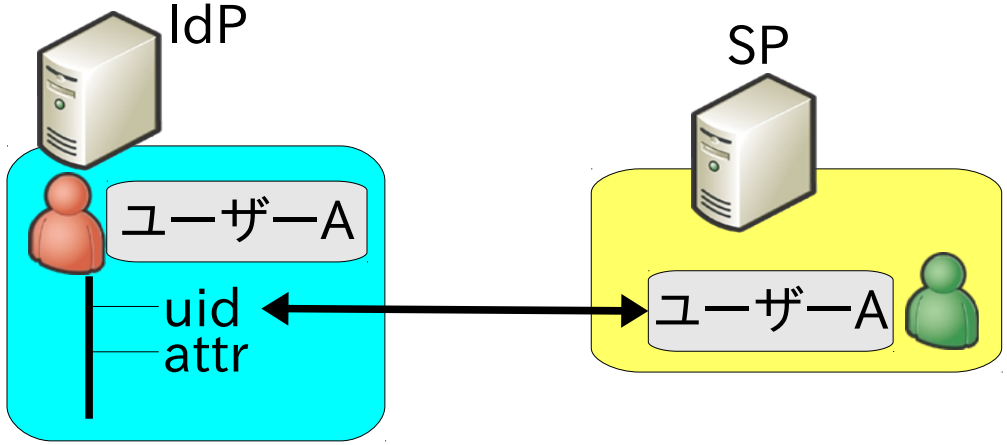
- **アカウント連携**:IdP のアカウントと SP のアカウントを紐付ける
- **NameID** というユーザー識別子を IdP と SP 間で共有することで実現する
- NameID には以下のものが使用される
  - メールアドレス
  - ユーザー属性情報(ユーザー名など。Google Apps はユーザー名を Name ID として使用する)
  - 仮名:ランダムな文字列によるユーザー識別
  - X.509 の Subject

仮名による連携



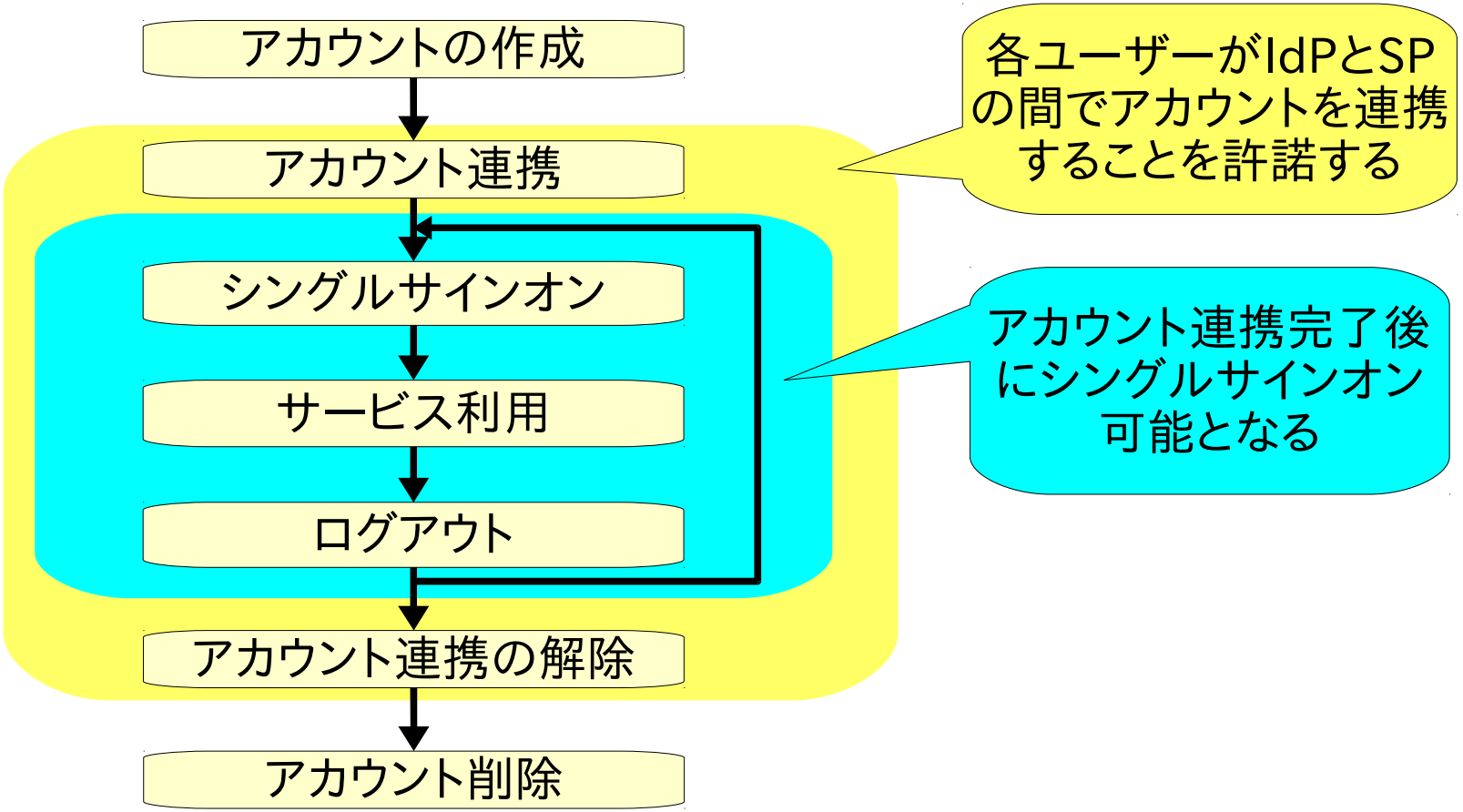
- IdP のアカウントと SP のアカウントを**仮名**(仮 ID のようなもの)で紐付ける
- 基本的にユーザー毎に設定する。初回のみ、IdP と SP にそれぞれのID/パスワードでログインする必要がある
- IdP/SP 内のアカウント情報(ユーザー ID など)を隠蔽したままアカウント連携可能

ユーザー属性情報による連携

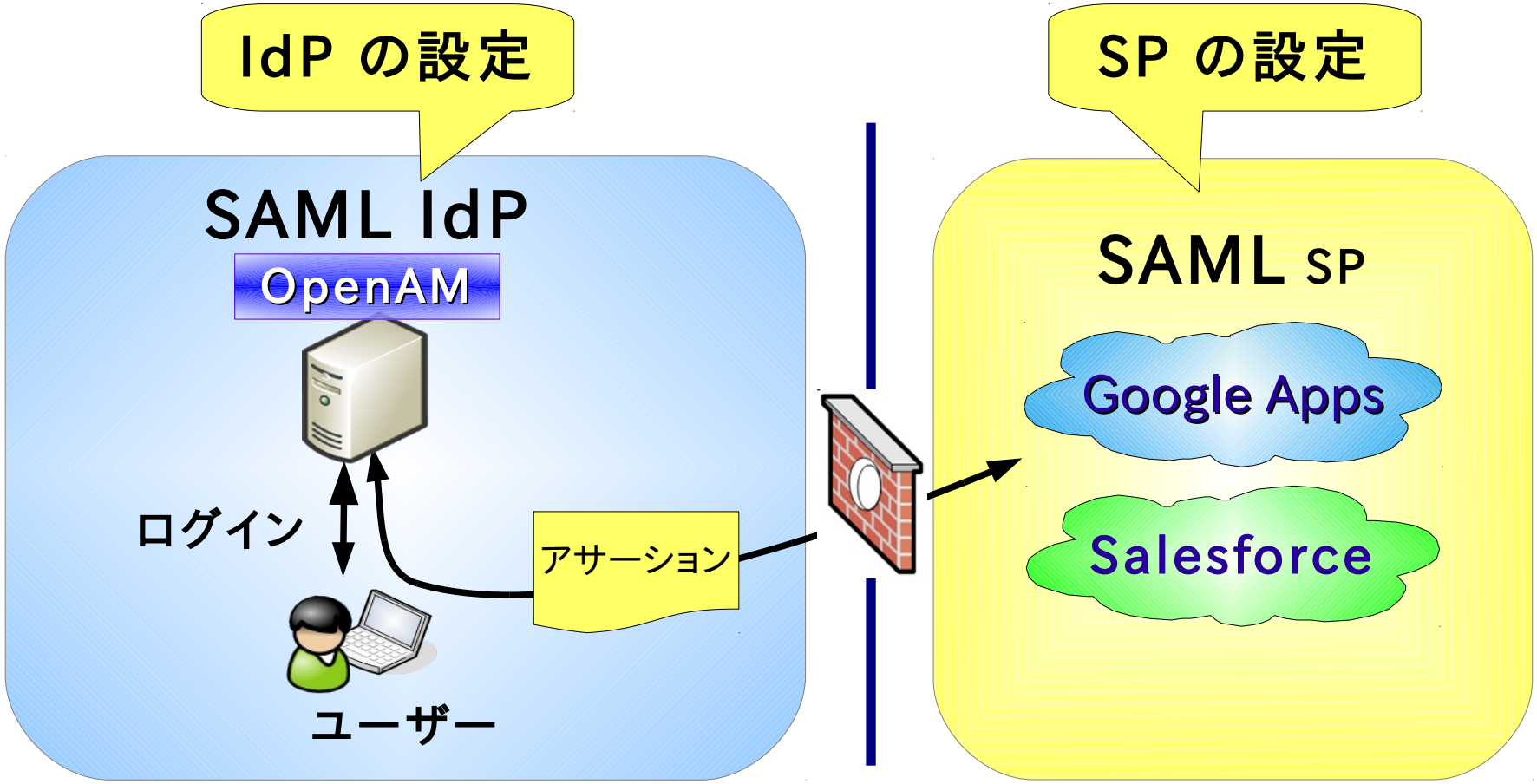


- IdP のアカウントと SP のアカウントをユーザー属性で**直接連携**
- Google Apps はこの方式(トラストサークルに入ることで自動的に全ユーザーのアカウント連携が有効化)
- 自システム内のユーザー属性情報の一部を相手に知られる

SAML を利用したシングルサインオン環境における [アカウント作成→SSO→アカウント削除] のサイクル



実際にSAMLのシングルサインオン設定をやってみる



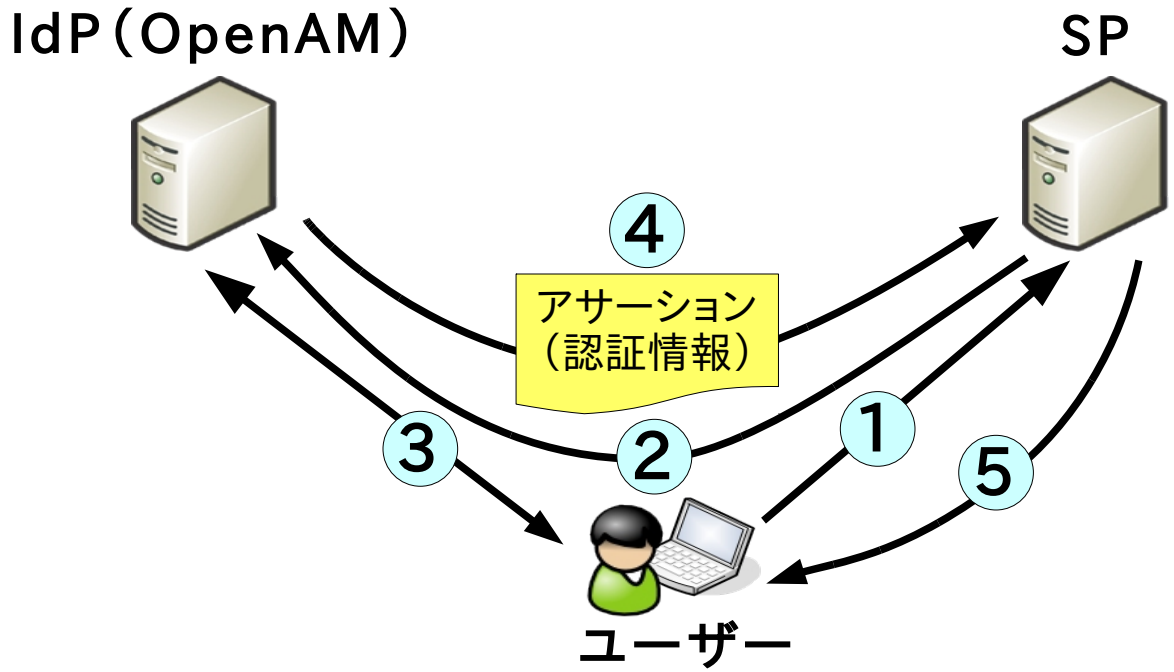
SAMLによるシングルサインオン

- シーケンス -

2通りの認証シーケンス

- SP-initiated SSO
  - ユーザーは最初にSPにアクセスし、IdPでの認証に成功した後に、再びSPにアクセスする
- IdP-initiated SSO
  - ユーザーは最初にIdPにアクセスし、IdPでの認証に成功した後にSPにアクセスする

SAMLにおけるSSO（SP-initiated SSO）



1.ユーザーが未認証の状態で SP にアクセスする

2.SP は SAML 認証要求を IdP に送信する

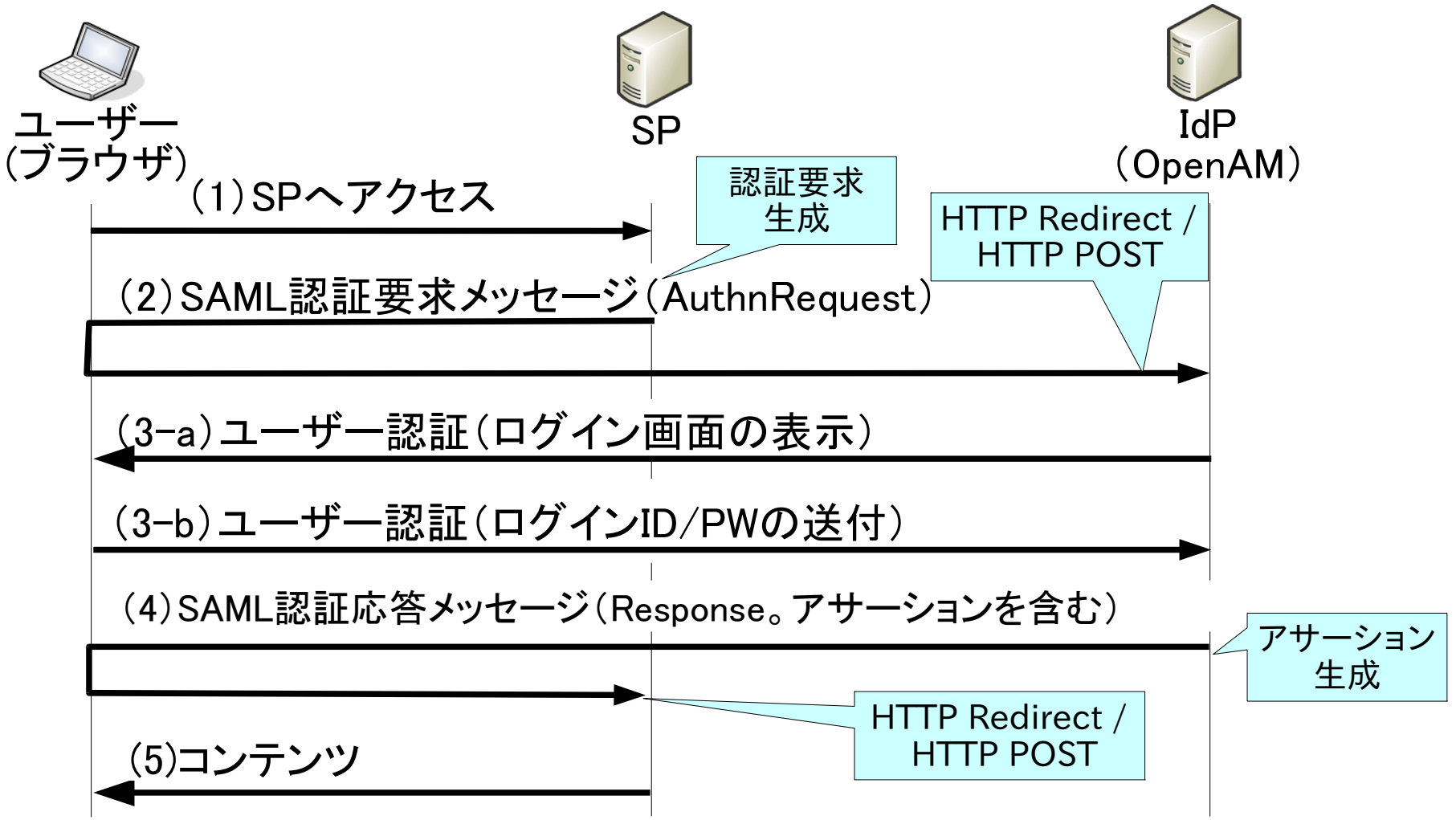
3.IdP はユーザーを認証する

4.IdP での認証に成功すると、IdP は SP に SAML 認証応答（アサーションを含む）を送信する

5.SP は認証応答を受け取るとユーザーにコンテンツを提供する

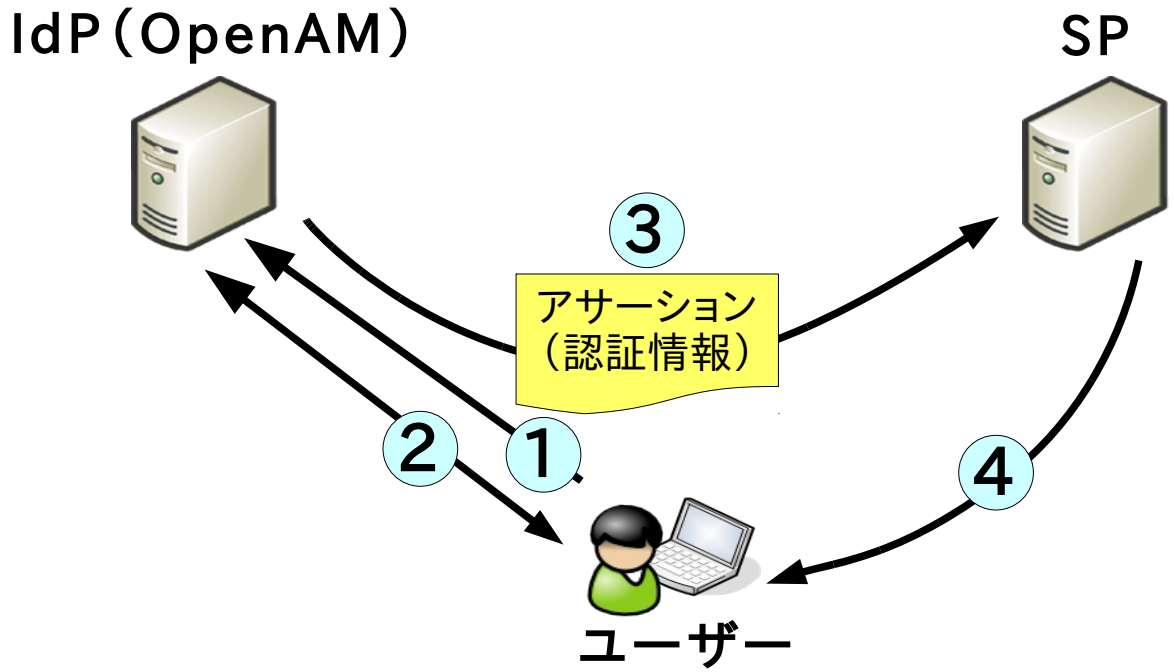
※この図は、HTTP Redirect Binding/HTTP POST Binding の場合の例です

SAML 認証のシーケンス（SP-initiated SSO）



※この図は、HTTP Redirect Binding/HTTP POST Binding の場合の例です

SAMLにおけるSSO（IdP-initiated SSO）



1.ユーザーが未認証の状態で IdP にアクセスする

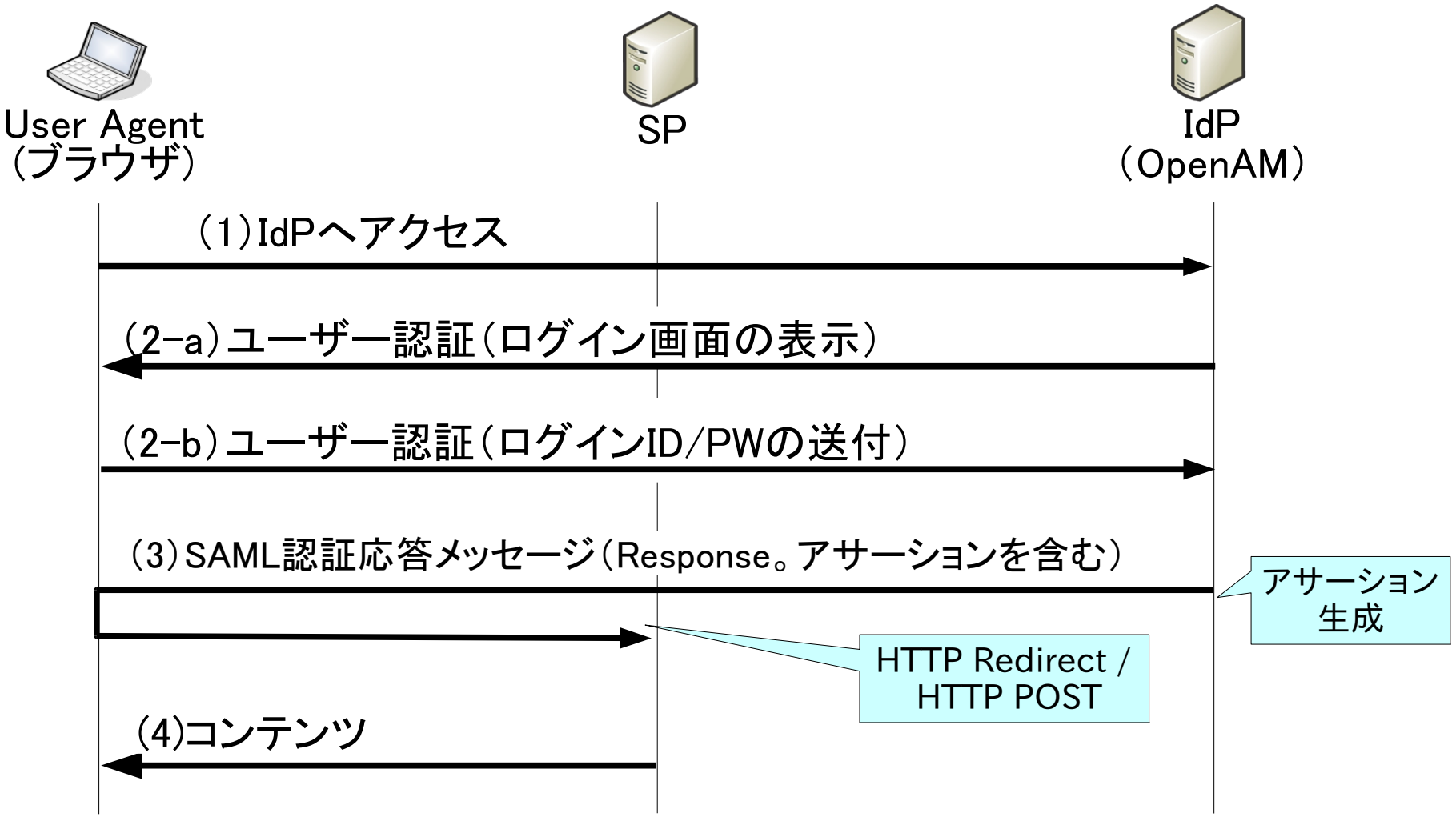
2.IdP はユーザーを認証する

3.IdP での認証に成功すると、IdP は SP に SAML 認証応答（アサーションを含む）を送信する

4.SP は認証応答を受け取るとユーザーにコンテンツを提供する

※この図は、HTTP Redirect Binding/HTTP POST Binding の場合の例です

SAML 認証のシーケンス（IdP-initiated SSO）



※この図は、HTTP Redirect Binding/HTTP POST Binding の場合の例です

SAML – メッセージの送受信方法

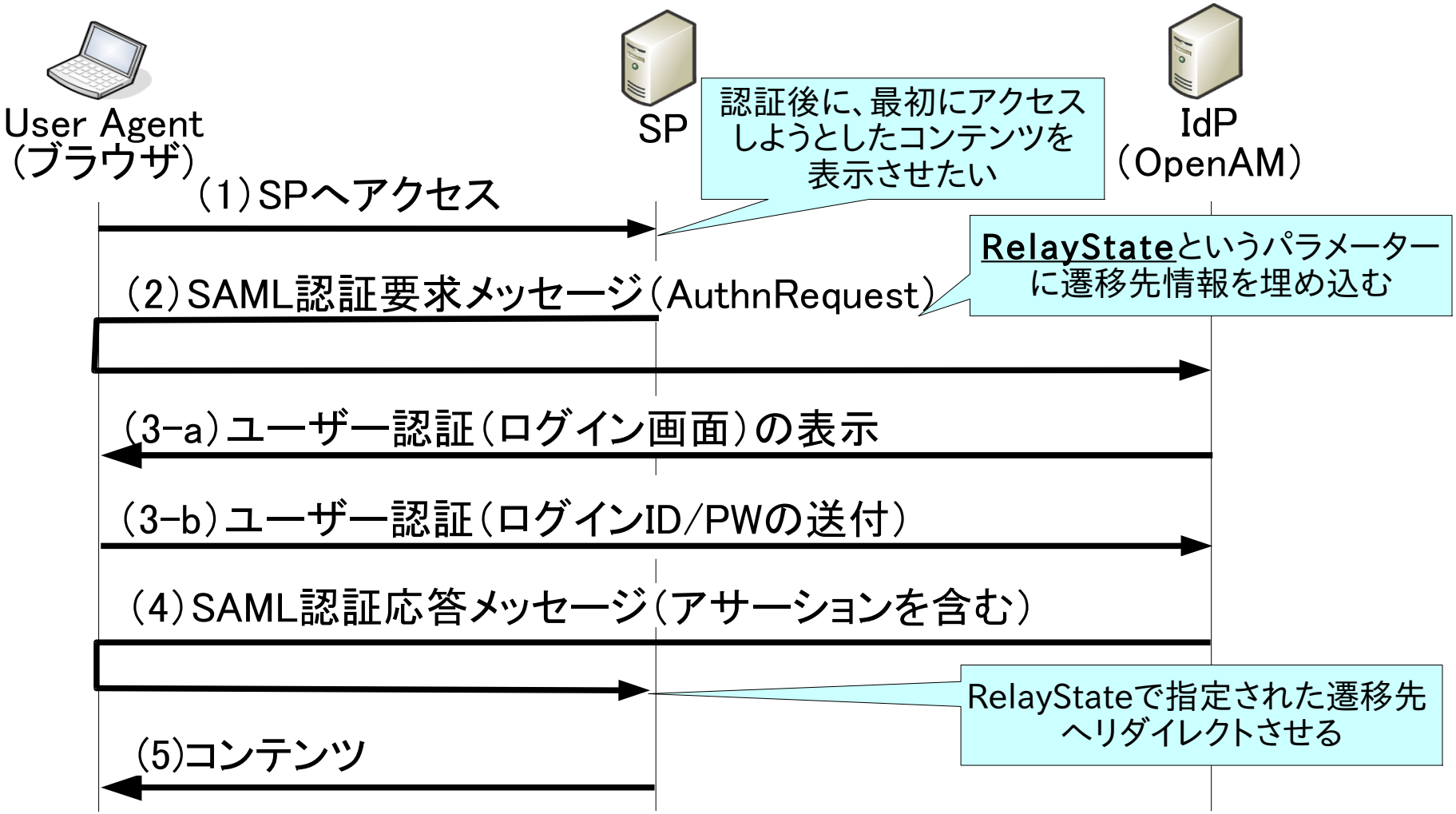
- HTTP Redirect/HTTP POST Binding
  - ブラウザが通信を中継する（HTTP Redirect/HTTP POST を利用）
  - IdP-SP間の直接的な通信が発生しない

方式	説明	特徴
HTTP Redirect	SAML メッセージを Base64 エンコードし URL パラメータに埋め込んで GET メソッドで送信(HTTP ステータスコード 302/303 を利用)。Google Apps は SAML認証要求で使用。	URL が長すぎると、ブラウザの URL の長さ制限に抵触する可能性がある。古い携帯ブラウザでは使えないことも。
HTTP POST	Base64 エンコードした SAML メッセージを HTMLフ Form に埋め込んで POST メソッドで送信。Google Apps、Salesforce が採用。Google Apps はSAML認証応答で使用。	IdP へのログイン→SP への遷移を自動化するには、JavaScript を利用して自動的に POST リクエストを送信させる必要がある。

- HTTP Artifact Binding
  - IdP-SP間の直接的な通信が発生する
  - アサーションへのリファレンスである Artifact をブラウザを介してIdPとSPの間で送受信する。IdPと SP は Artifact を利用して直接相手に SAML 認証要求/認証応答メッセージを問い合わせる。Artifact のデータサイズは小さい。

HTTP POST Binding で Javascript が使われている場合は、セキュリティ系のツールに引っかかるかも…

IdPで認証が完了した後に、SPの特定のURLに遷移させる



※この図は、HTTP Redirect Binding/HTTP POST Binding の場合の例です

SAMLによるシングルサインオン

- アサーション -

SAML - アサーション

- IdP が発行する、ユーザーに関する認証情報の XML
- アサーションの改竄によるユーザー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
  - 事前に IdP の証明書を SP に登録しておく必要がある

```
<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
    ユーザー識別子(NameID)

</saml:Assertion>
```

SAML - 認証要求・認証応答

- **認証要求（AuthnRequest）**
  - SPがIdPに対して、ユーザーの認証情報（アサーション）を要求するメッセージ

```
<samlp:AuthnRequest ID="xxx" Version="2.0" Destination="http://idp.osstech.co.jp/idp/sso">

  認証要求情報

</samlp:AuthnRequest>
```

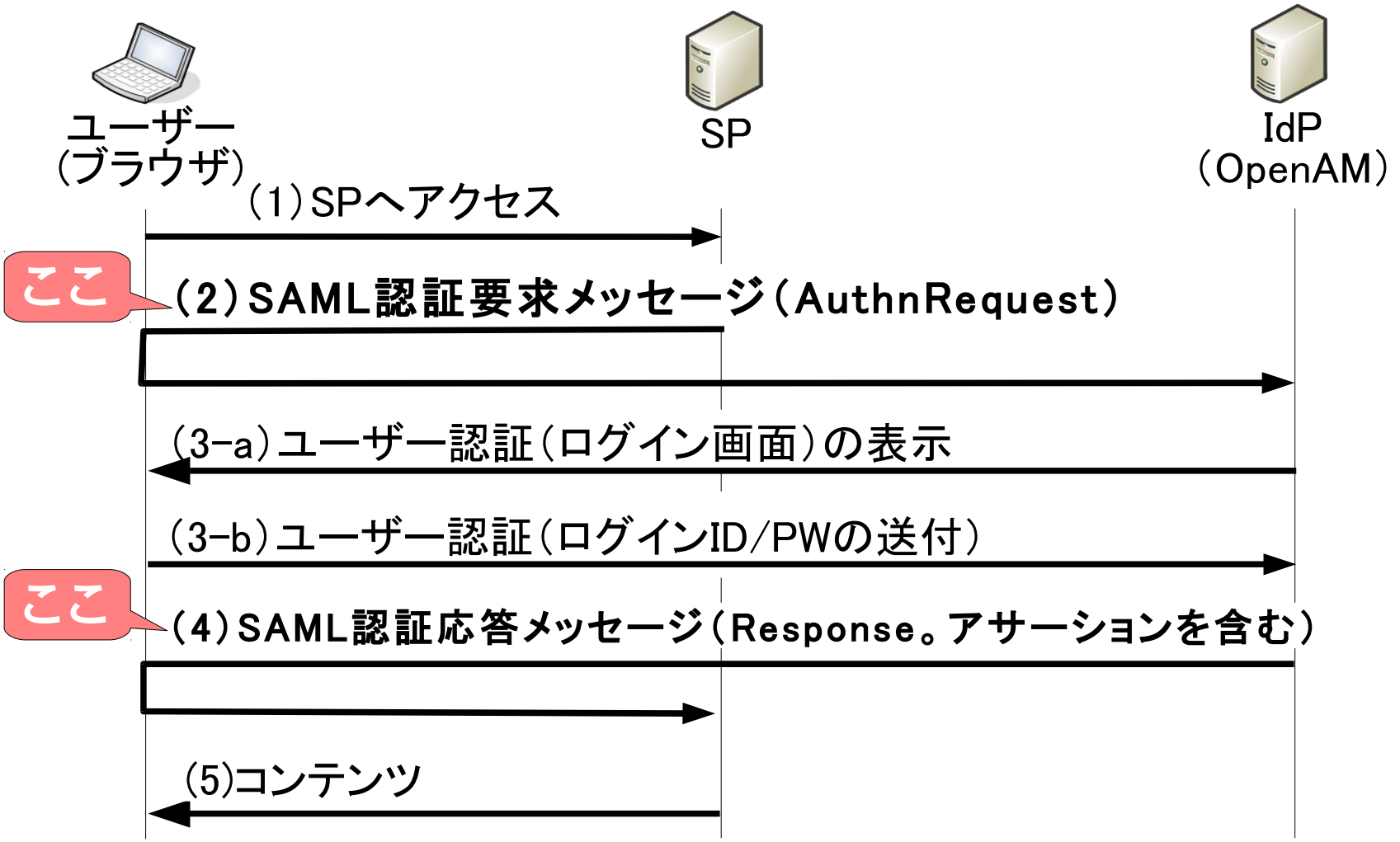
- **認証応答（Response）**
  - IdPがSPにユーザーの認証情報（アサーション）を送付するメッセージ

```
<samlp:Response ID="xxx" Version="2.0" Destination"http://sp.osstech.co.jp/sp/sso">
  < saml:Assertion ...>

    アサーション

  </saml:Assertion>
</samlp:AuthnRequest>
```

実際にSAMLメッセージを覗いてみる



SAML 製品/サービス選定のポイント

- SAML仕様をフルスペックで実装している製品/サービスは少ないと思われる

- 特に、SAML SP 側ではその（SPが提供する）サービスに必要なSAML 仕様しか実装されていないことも
  - IdP-initiated SSO には対応しているが、SP-initiated SSO には対応していない
  - RelayState に対応していない

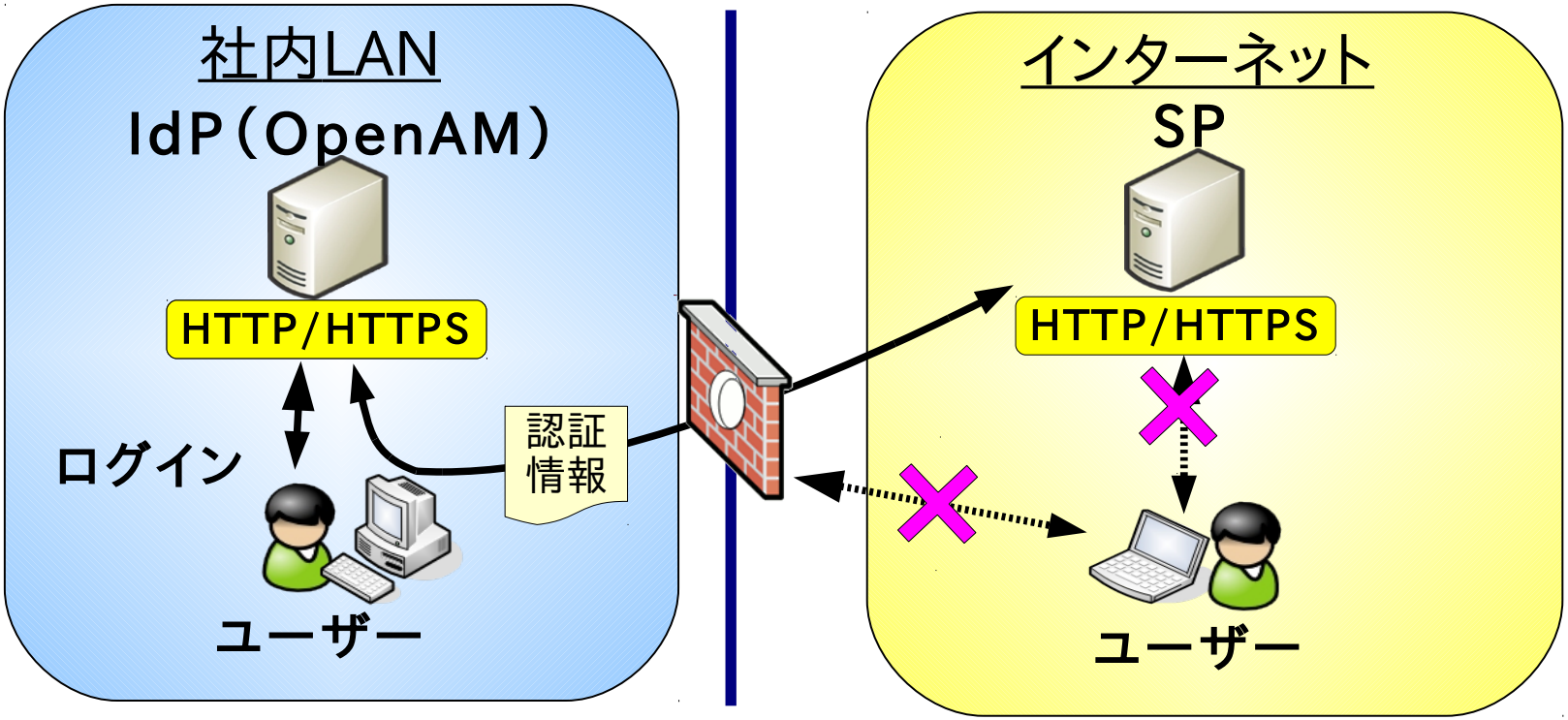
SAML 対応製品/サービスを選定する際は
SAML 仕様がどこまで実装されているか
確認することが大切
（“イケてない”SSO環境になってしまうことも…）

SAMLによるシングルサインオン

- ネットワーク構成 -

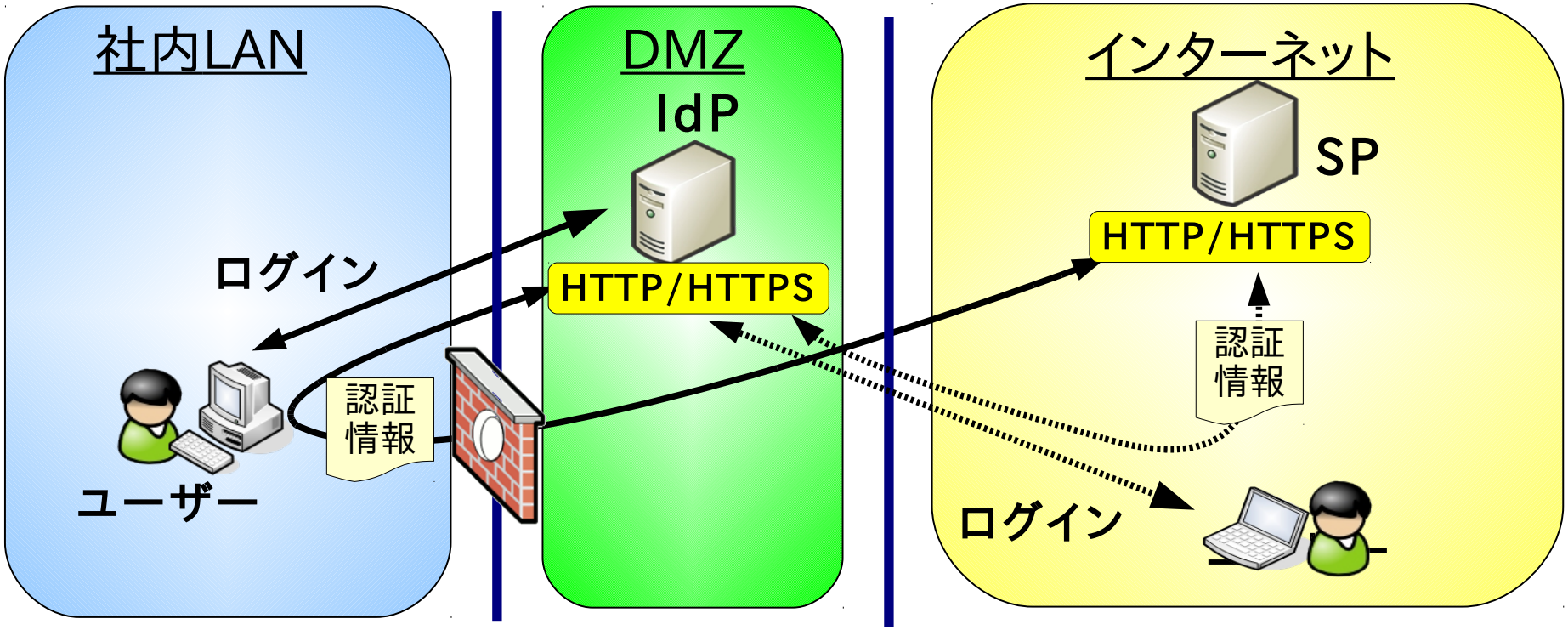
SAML IdP のネットワーク上の配置 - その1

- 認証を行なうSAML IdP（OpenAM）を社内LANに設置することで、SAML SP（Google Apps、Salesforceなど）へのアクセスを社内のみからに制限することが可能
- 「**俺専用 IdP**」を作ることも可能（実用性はあまり無い…）



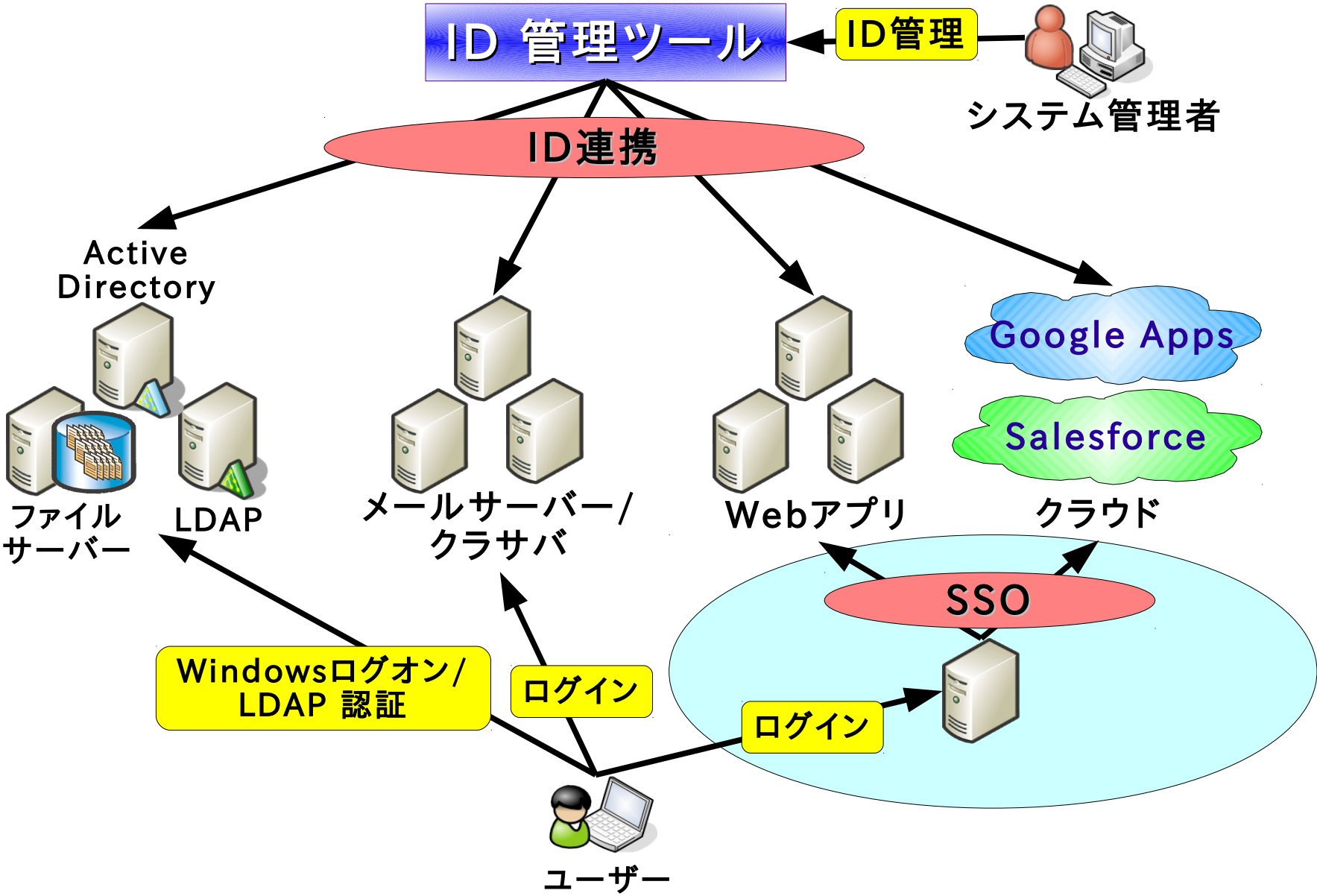
SAML IdP のネットワーク上の配置 - その2

- 社外からも SAML SP（Google Apps、Salesforceなど）にアクセスする場合は、SAML IdP（OpenAM）を社外からアクセス可能な場所に設置する（DMZなど）



ID管理との組み合わせで効果倍増!

ID 管理 & SSO



ID管理との連携

- シングルサインオンとID管理は一緒に使うことで最大の効果を発揮する
  - ユーザーID/パスワードはシングルサインオンシステムで一元管理可能でも、各アプリケーション/サービスに必要なユーザー情報は、基本的には個々に管理される
  - ID管理ツールなどを利用したID一元管理をしなければ、シングルサインオンは破綻することも

- クラウドサービスにおいてもID管理は必要
  - クラウドサービス側にもユーザー情報を保存することから、ID管理の対象となる
  - ID管理用のAPI（プログラムインタフェース）を備えているものも多い（Google Apps、Yahoo! など）

参考情報

- LIBERTY ALLIANCE のセミナー資料（SAML）
  - http://wiki.projectliberty.org/images/9/94/080215_JapanSIG_Technical_Seminar.pdf
  - SAML を調べるのであれば、まず最初に読むのがおすすめ
- SAML 公式サイト
  - http://saml.xml.org/
  - オープンソースの SAML 実装：http://saml.xml.org/wiki/saml-open-source-implementations
- SAML 仕様の原文
  - http://www.oasis-open.org/specs/index.php#saml
- Google Apps の SAML シングルサインオンの解説
  - http://code.google.com/intl/ja/apis/apps/sso/saml_reference_implementation.html
- Salesforce の SAML SSO 設定（IdPとしてOpenSSOを想定）
  - http://wiki.developerforce.com/index.php/Single_Sign-On_with_SAML_on_Force.com
- OSSTechのOpenSSO勉強会資料
  - http://www.osstech.co.jp/techinfo/opensso

ご清聴ありがとうございました