京都産業大学
KYOTO SANGYO UNIVERSITY

学認のサービス連携推進のための機能開発

秋山 豊和
京都産業大学

Keep Innovating.

内容

- 背景と課題
- 取り組んだ内容（１）
  - 拡張機能を用いた利便性・セキュリティの向上
- 取り組んだ内容（２）
  - Shibboleth SPとアプリケーションフレームワークの親和性向上
- 取り組んでいる内容
  - 新しい形式のサービスへの対応
- まとめ

背景

- WebアプリケーションとSSO
  - Webアプリケーションの増加
  - SSOによる認証統合
- SSO化のメリット
  - パスワードの強化
  - 多要素認証の導入による認証強化
    - SSOで統合しておけば、導入コストは低減可
    - リスクベース認証の併用により利用コストも低減
  - ID統合管理
  - アプリケーション間のデータ連携
  - etc.

背景

- 連携サービスの増加
  - SSOによる利点の拡大
  - 組織としてのサービス改善
- 連携サービスの対象
  - 学内サービス
  - 学外サービス

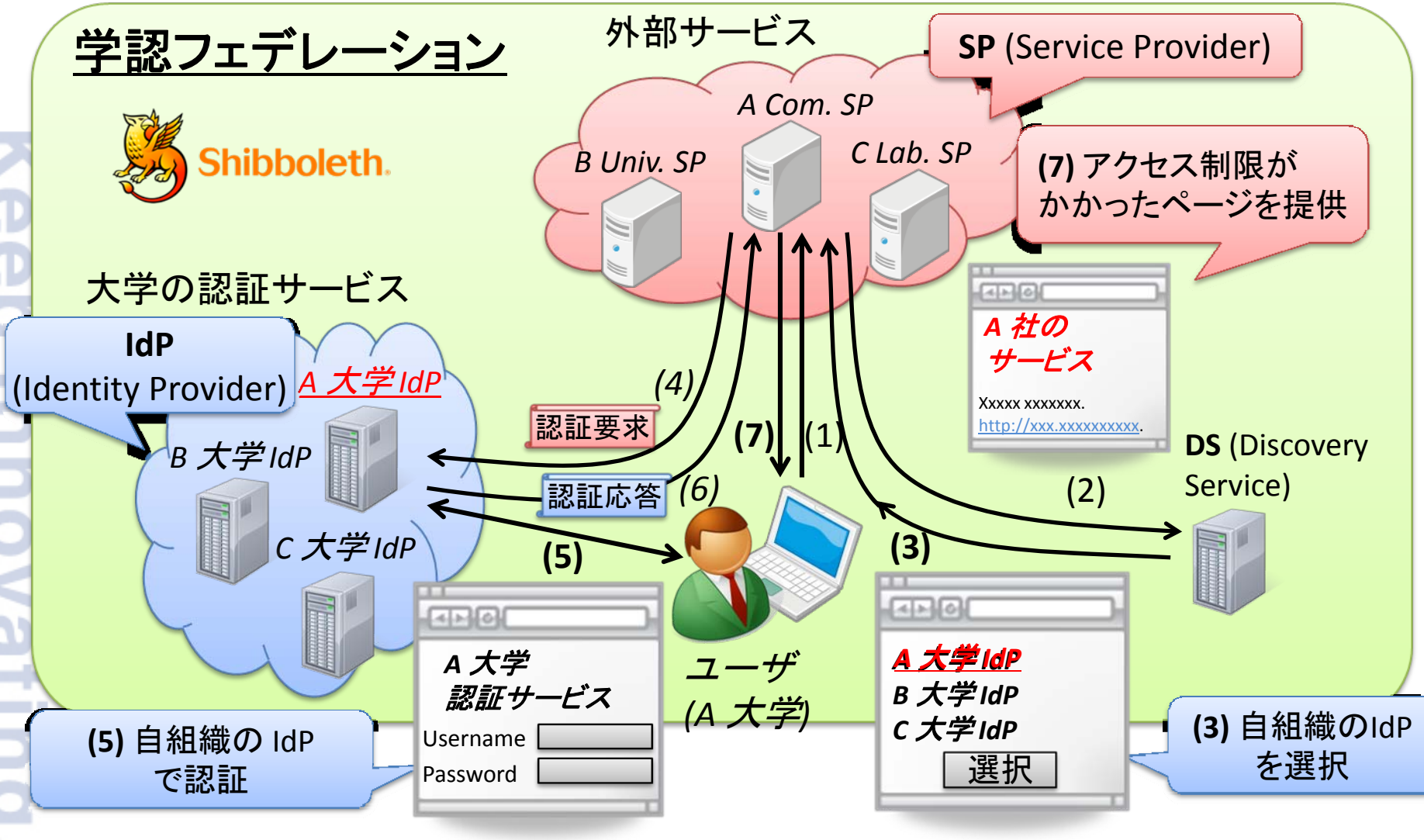
学認によるメリットが大きい部分

SSOの課題

(1)「連携サイトの増加」に伴う
利便性・セキュリティの課題

(2)「連携サービスの追加」に伴う
アプリケーション開発の課題

ShibbolethのSSO手順



（1）連携サイトの増加

連携サイトの増加により共通基盤の価値が向上

連携サイトの増加により下記ようなの懸念が増大

主にエンドユーザの課題

Embedded DS

(1) 所属組織のIdP選択の手間の増加

・ユーザ教育コスト増加の懸念

問題B

(2) ID・パスワードの資産価値の増大

問題A

(3) 組織外サービスの利用機会の増加

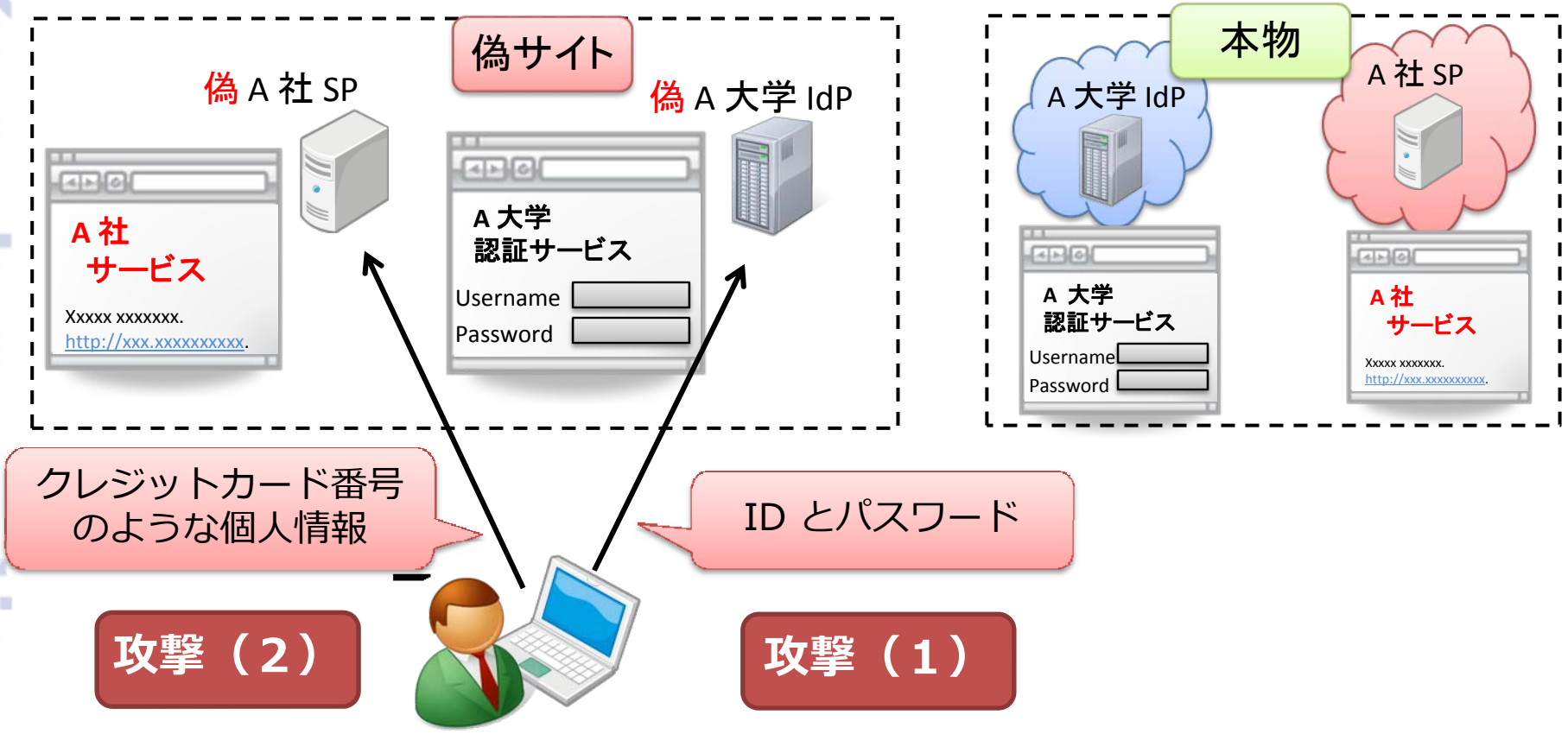
・フィッシングの懸念

必ずしも対策できていない

ブラウザ拡張機能を使って解決できないか？

問題A フィッシング詐欺

- フィッシング詐欺で盗まれるもの



フィッシング詐欺対策

- URL Reputationサービス
  - 自動的に悪意のあるURIに対して警告を出してくれる
  - Internet Explorer： SmartScreen
  - Google Chrome： Safe Browsing API
    - Firefox, Safariも使用

Good!

ユーザの事前知識がなくても適切に
フィッシングサイトを除外してくれる

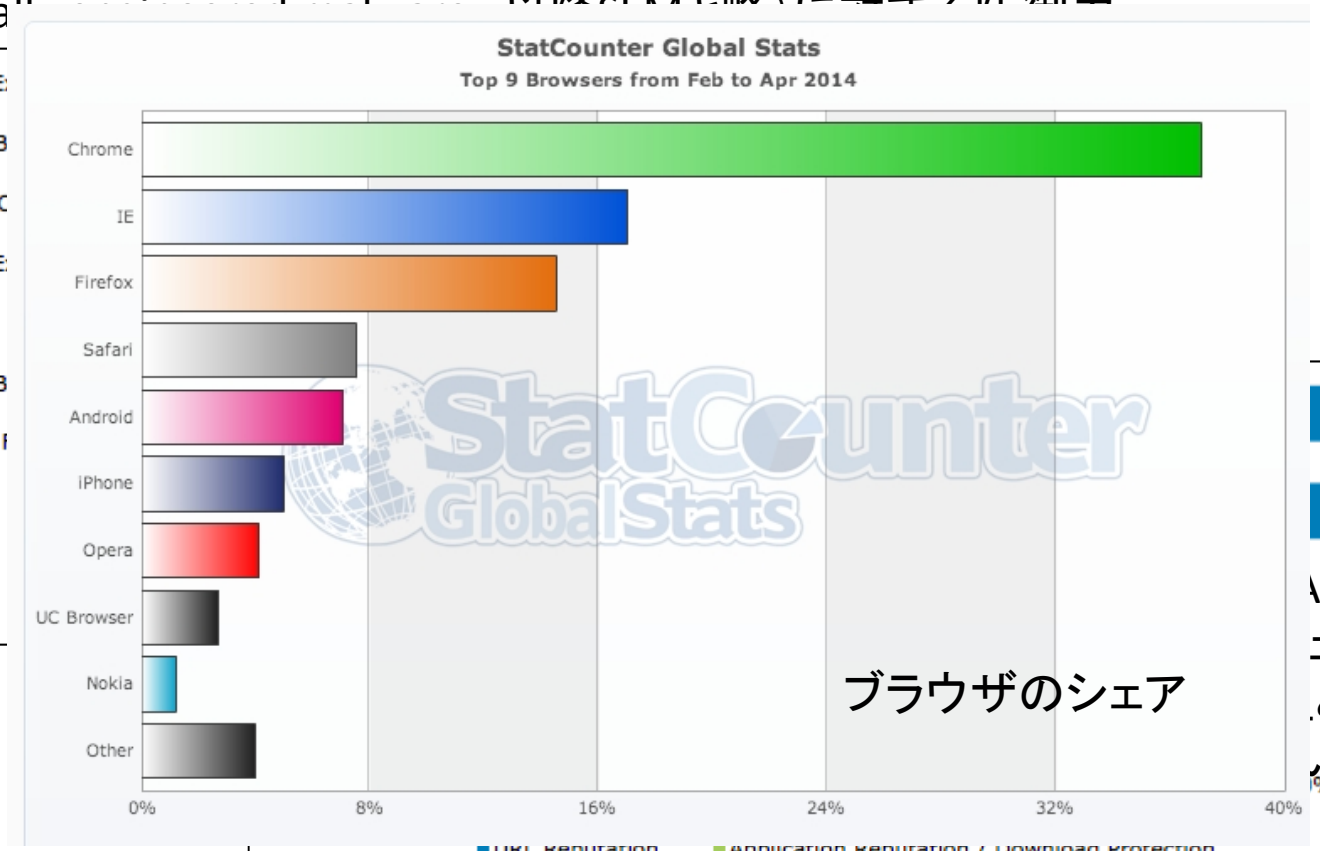
しかし. . .

本当にフィルタ結果を信用できるのか？

学認を通じて信頼できるサイトの情報を得ている
それを利用できないか？

URL Reputationサービス

ソーシャルエンジニアリングを用いるマルウェア
（socia



ブラウザのシェア

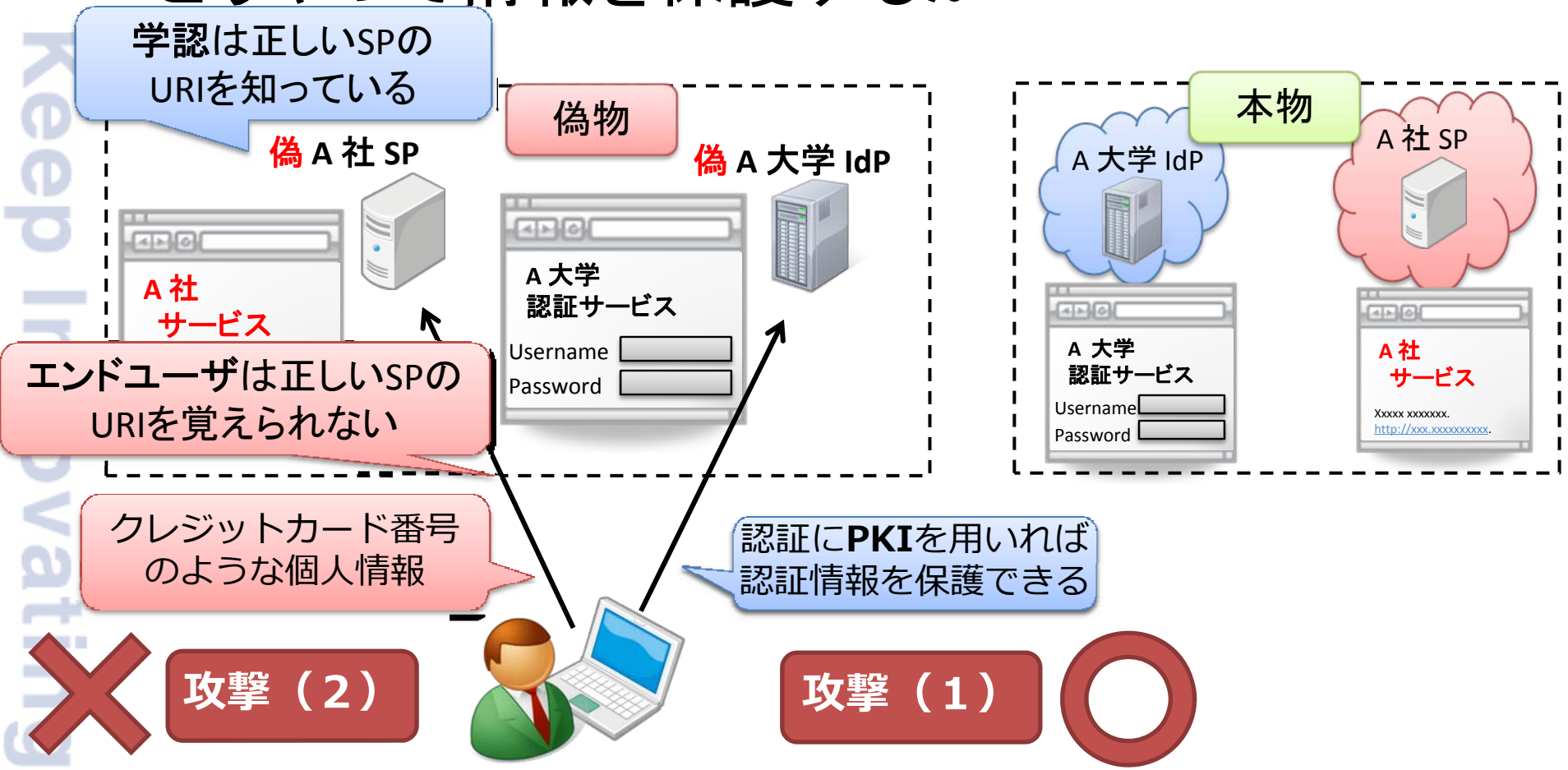
AMPという
ェア対策
みで防御
いる

フィッシング対策として取り組んだ内容

- PKIを用いたアプローチ
- チャレンジレスポンスを利用したアプローチ

PKI を用いたアプローチ

- どうやって情報を保護するか？

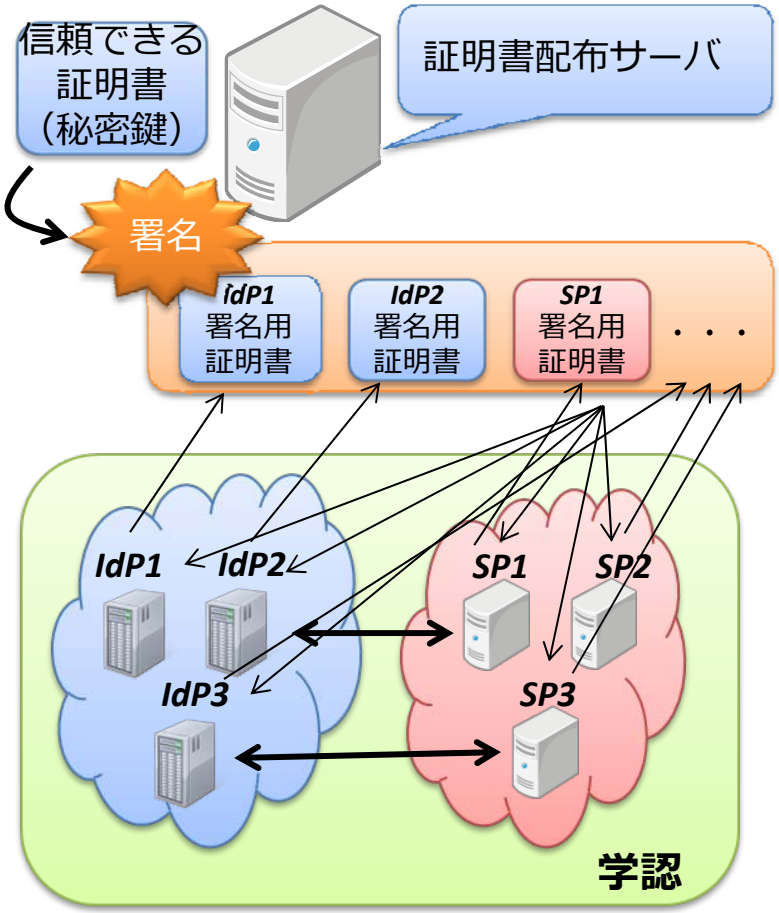


学認はどのようにして メンバIdP，SPを記憶しているか？

- メタデータ(IdP/SP 証明書)の共有

ただしサーバ間のみ

1. IdP/SPの証明書を収集 (SAMLメタデータ)
2. IdP/SPの証明書に信頼できる証明書で署名
3. IdP/SP証明書の配布



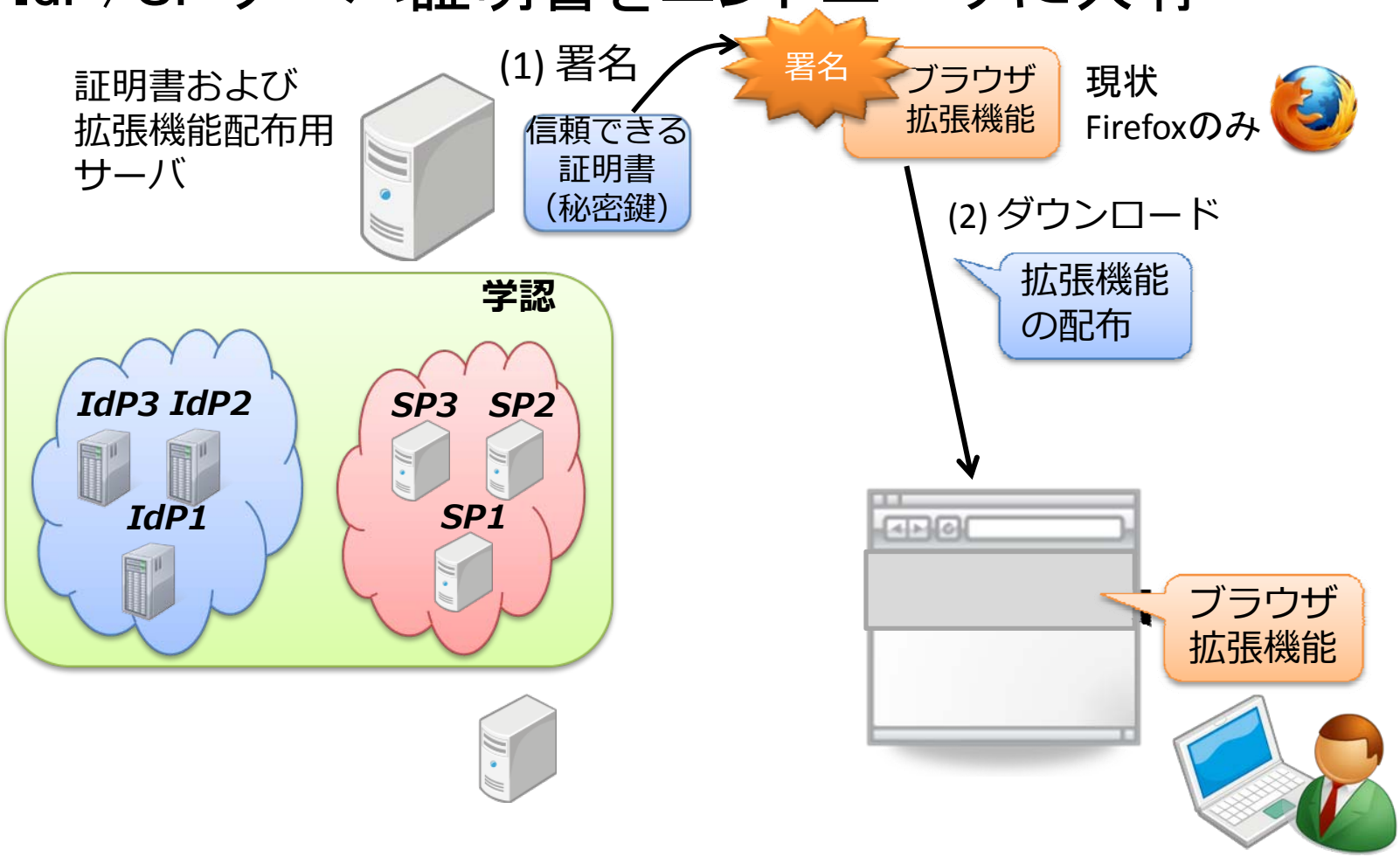
IdP/SP は互いに識別可能

SAML認証要求･応答は IdP/SP証明書を検証に利用

証明書共有範囲の拡張

エンドユーザまで

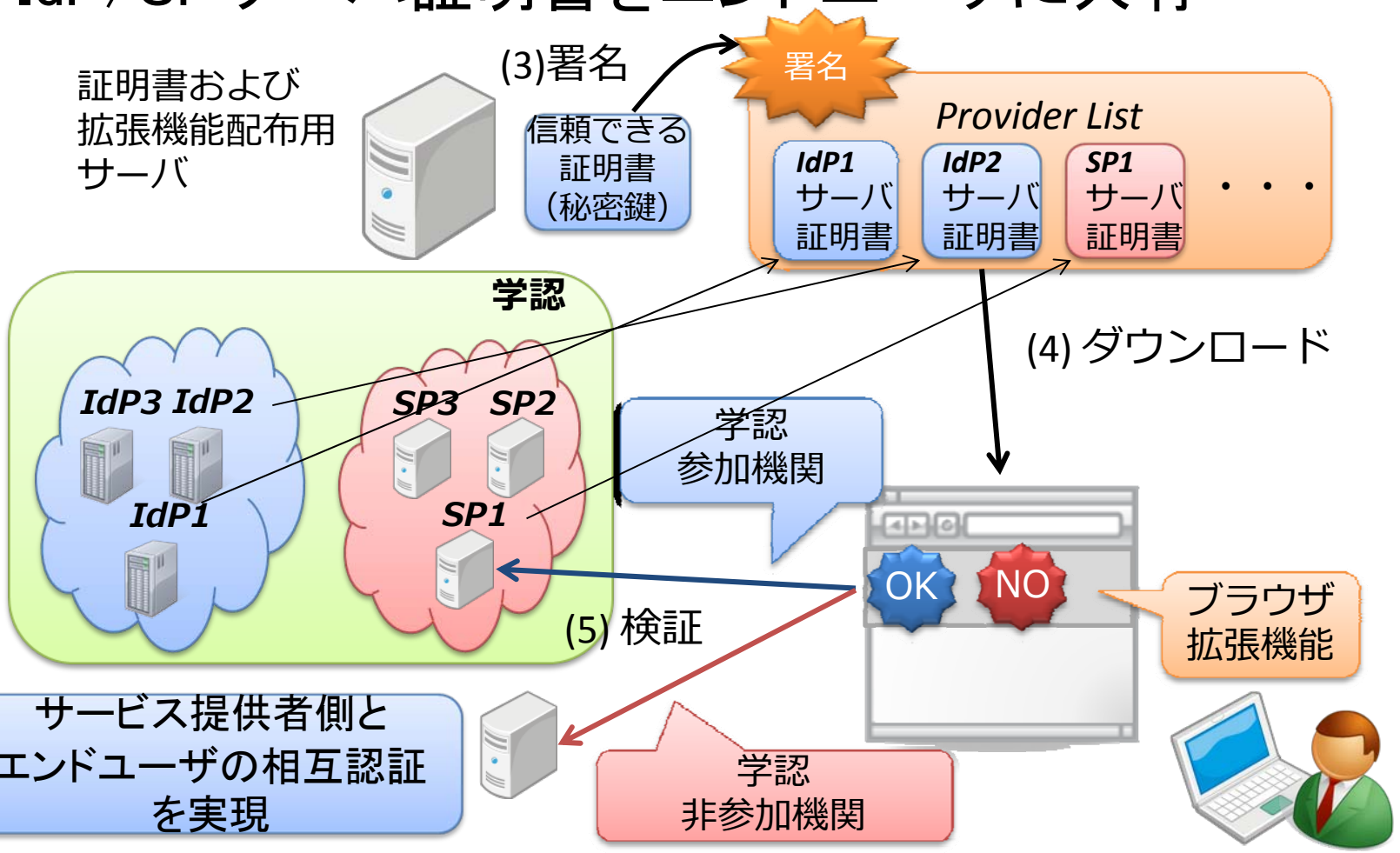
- IdP/SPサーバ証明書をエンドユーザに共有



証明書共有範囲の拡張

エンドユーザまで

- IdP/SPサーバ証明書をエンドユーザに共有



ユーザ認証にPKIを用いる場合の課題

- 厳密なPKI は高い運用コストが要求される
  - 例）ICカード

Good!

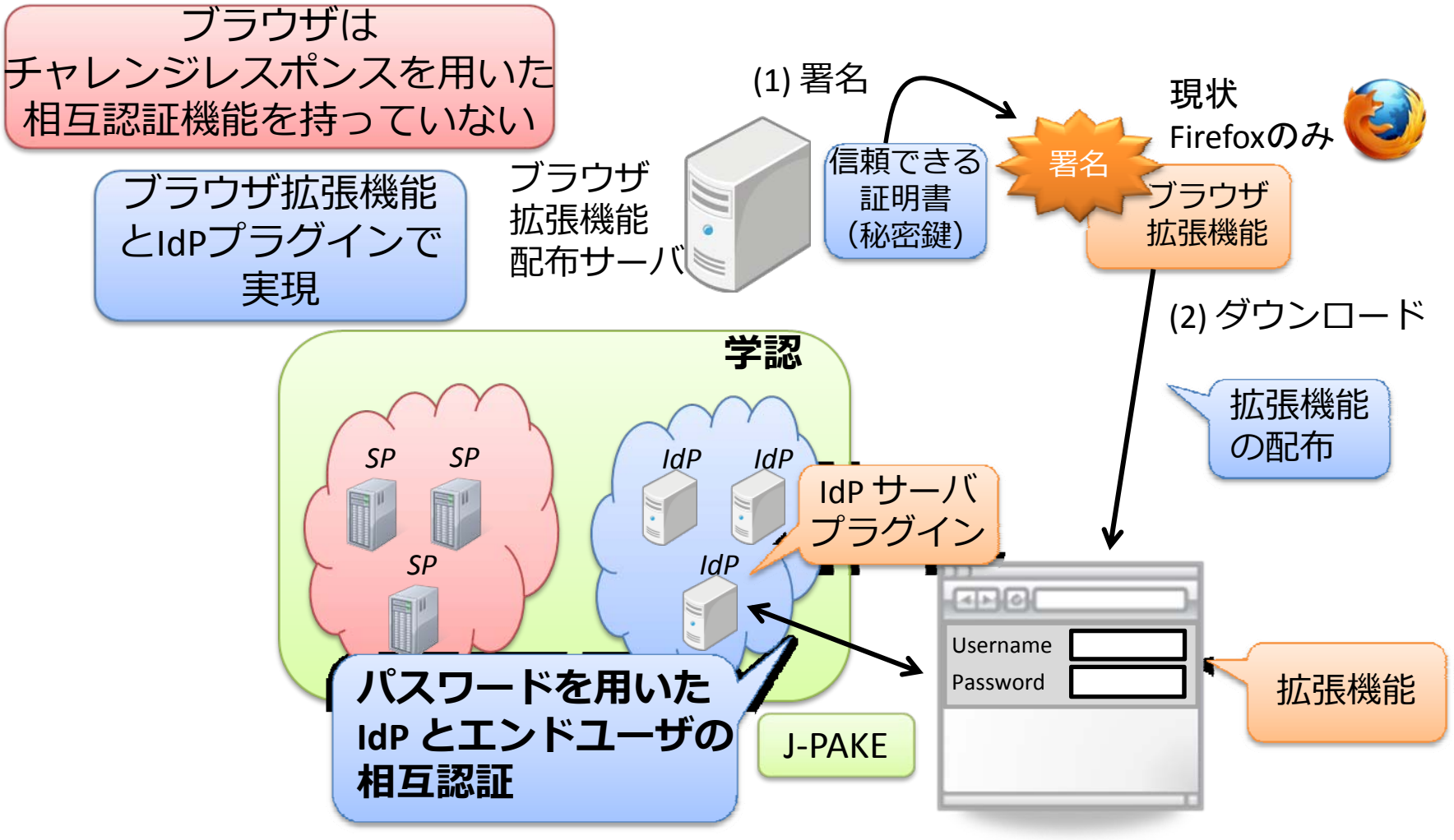
- 簡易PKI はPKIの運用コストを低減できる
  - オンラインで公開鍵・秘密鍵のペアを配布
  - 秘密鍵はパスワードで暗号化

しかし．．．

サービスとユーザの相互認証を実現するには
証明書共有範囲の拡張が必要

他にユーザ情報が登録されているIdP
を認証する方法はないか？

チャレンジレスポンスを用いたアプローチ



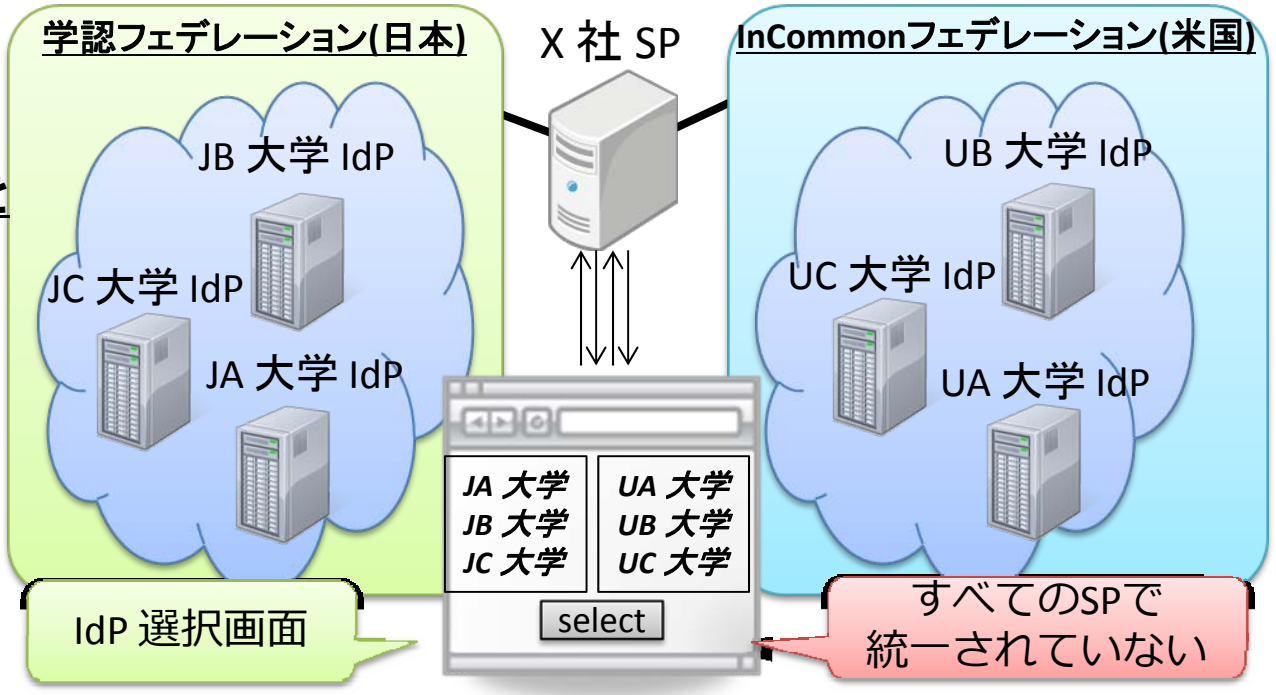
問題B IdP選択の手間

- IdP/SPの増加により発生
  - DSにおけるIdPリストが長くなる
  - 複数フェデレーションに対応したSPでは、IdP独自の選択画面をもつ

e.g. 出版社等

フェデレーションをまたがるDSがないため

複数フェデレーションをサポートするSPの例



問題B IdP選択の手間

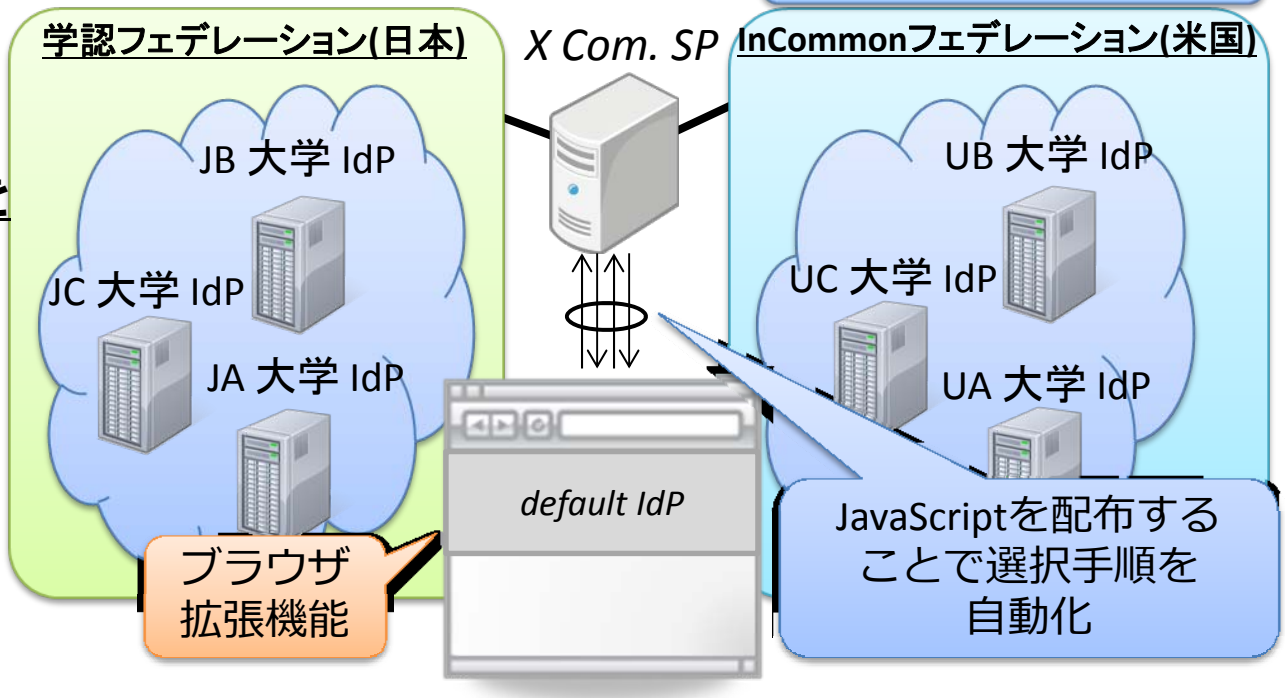
- IdP/SPの増加により発生
  - DSにおけるIdPリストが長くなる
  - 複数フェデレーションに対応したSPでは、IdP独自の選択画面をもつ

フェデレーションをまたがるDSがないため

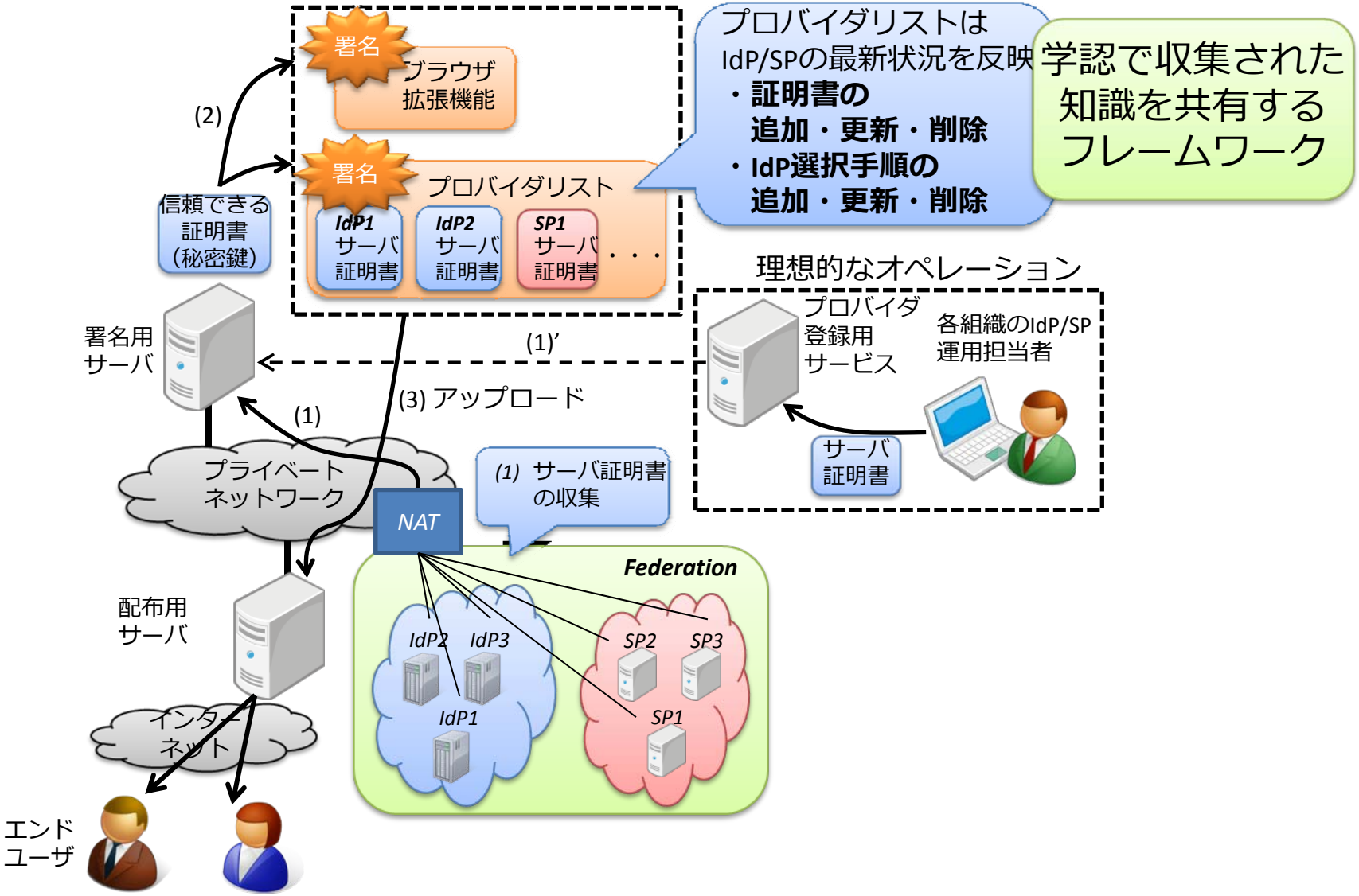
e.g. Publisher

IdP選択手順を簡易化

複数フェデレーションをサポートするSPの例



提案する構成



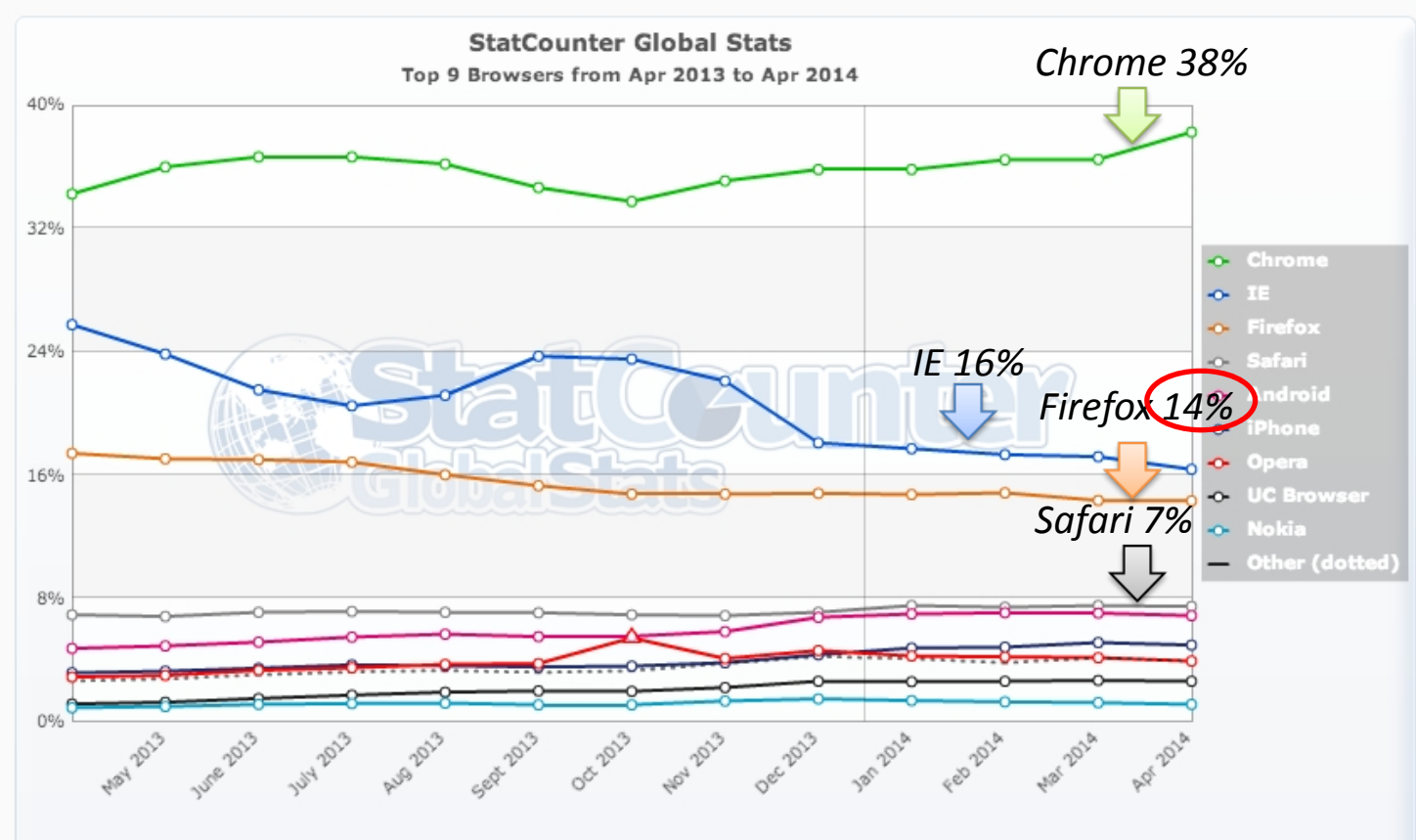
Keep Innovating.

プロトタイプ実装に関する考察

- 提案フレームワークの課題
  - ブラウザ拡張機能の実装
  - ブラウザ拡張機能の配布
- チャレンジレスポンスに対する攻撃
  - ブラウザ拡張機能に対するGUI攻撃
- 標準技術への適合
  - プロバイダリストの配布フォーマット

Statistics of Desktop Browsers

(Apr. 2013 – Apr. 2014)

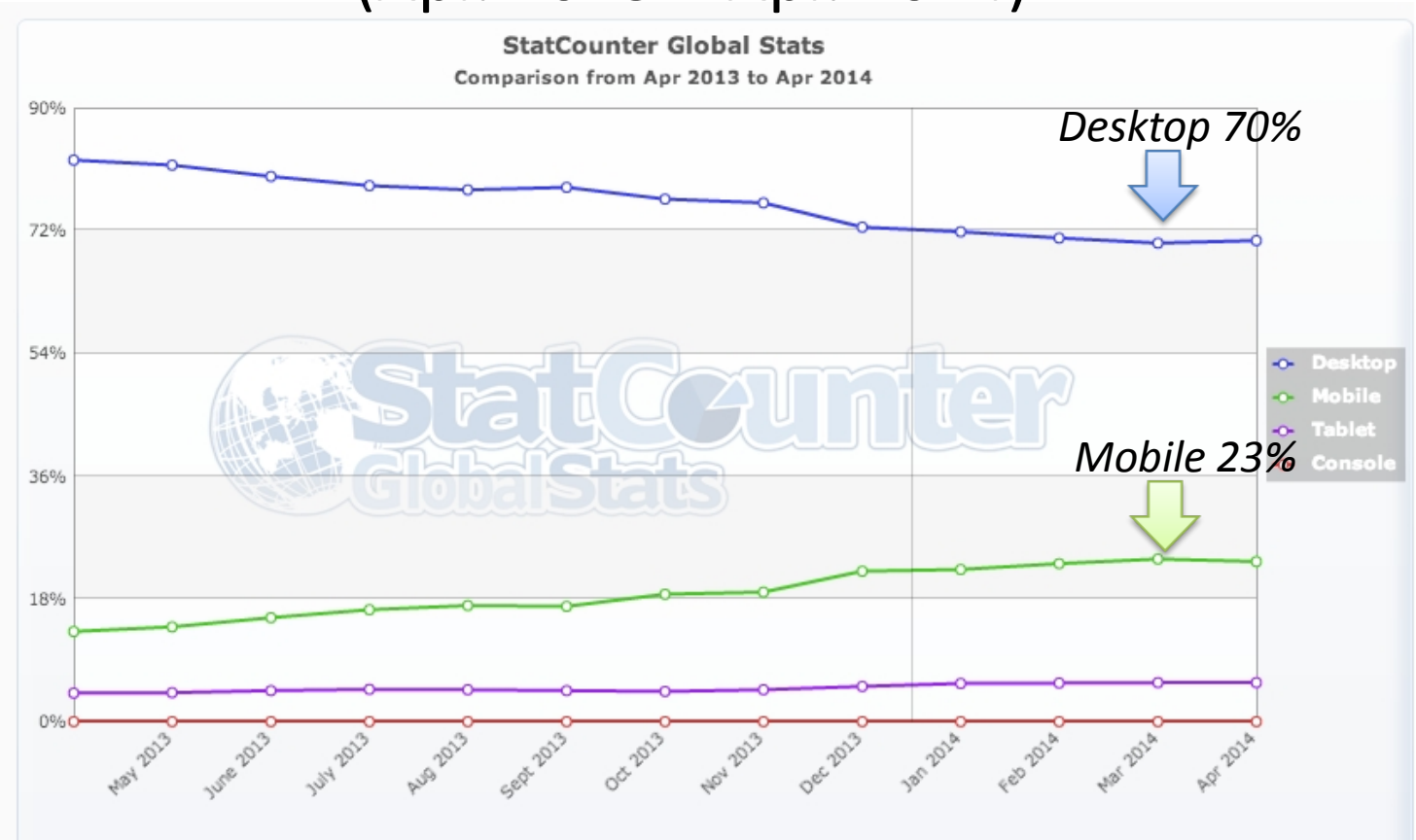


*Market share estimates for **desktop browsers***
from: http://gs.statcounter.com/

Firefoxのシェアは徐々に減っている

Statistics of Mobile vs Desktop

(Apr. 2013 – Apr. 2014)



*Market share estimates for **mobile vs desktop***
from: http://gs.statcounter.com/

モバイルはもっと増えていく

ブラウザ拡張機能のサポート

- Firefox
  - JavaScript, RDF, XUL
  - XPCOM (Nativeメソッドインタフェース)
    - nsIX509CertXX
    - nsISyncJPAKE
- Internet Explorer
  - Browser Helper Object (C#)
  - CryptAPI
- Chrome
  - JavaScript, HTML5, CSS
  - JavaScriptから利用できるのはブラウザAPIのみ
  - NativeインタフェースNSPIの利用は非推奨
- Safari
  - JavaScript, HTML5, CSS
  - JavaScriptから利用できるのはブラウザAPIのみ

基本的には JavaScript , HTML5, CSSで実装

↓

セキュリティライブラリの APIはブラウザごとに 異なっている

↓

Web Cryptography API (W3C) の標準化により 統一される可能性

Webアプリケーションにおける 暗号演算のJavaScript API

*David Dahl, **Mozilla** Corporation*
<ddahl@mozilla.com>
*Ryan Sleevi, **Google**, Inc.*
<sleevi@google.com>

http://www.w3.org/TR/WebCryptoAPI/

ブラウザ拡張機能の配布

課題

偽
配布サーバ

偽
ブラウザ
拡張機能

偽　ブラウザ
拡張機能
配布の危険性

運用による対策

配布サーバ

学認関連組織
による間接的で
限定的な配布

A 大学

署名

ブラウザ
拡張機能

運用担当者

署名の
検証

エンド
ユーザ

B 大学

Signature

ブラウザ
拡張機能

運用担当者

署名の
検証

エンド
ユーザ

コストに見合うかどうかが課題

ブラウザ拡張機能へのGUI攻撃

- 遠隔の悪意のあるWebアプリケーションが見た目の似たGUIを提供する可能性がある

本物のGUIの例

偽物のGUIの例

この部分が遠隔からダウンロードされたJavaScriptによって表示されている

もしすべてのユーザが同じインタフェースを使っていると簡単に攻撃できる

対策として、インタフェースの見た目のランダム化機能や設定機能を提供

プロバイダリストの配布形式

- プロバイダリストに含まれるもの
  - IdP/SPのentityIDとSAMLパラメータ
  - IdP/SPの証明書
  - 配布者の署名
  - 自動IdP選択コード

ほぼShibbolethのメタデータと同じ

- メタデータフォーマット(XML)が利用可能
  - JavaScriptに良いXML処理系がない
  - JSONフォーマットを用いるのが簡単
- プロトタイプ実装
  - JSON Simple Sign (Internet Draft) で実装
  - **JSON Web Signature** (Internet Draft) へのアップデートが必要か？

ひとまずまとめ

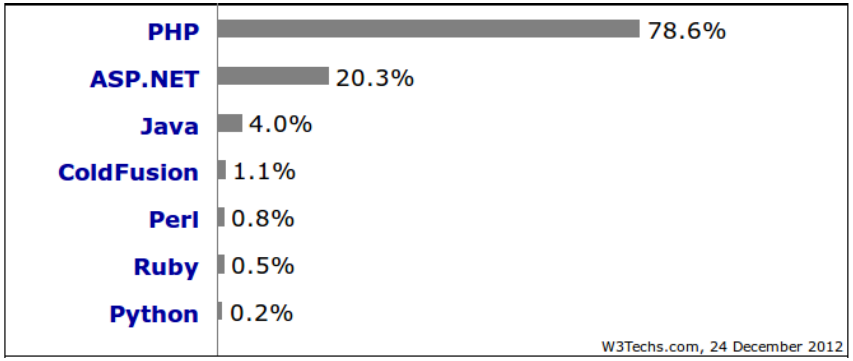
- ブラウザ拡張機能を用いた
  学認の利便性・セキュリティの改善
- 成果
  - 技術的な実現可能性についてはプロトタイプ実装で示せた
  - 実用化には様々な課題が残る

（2）連携サービス追加

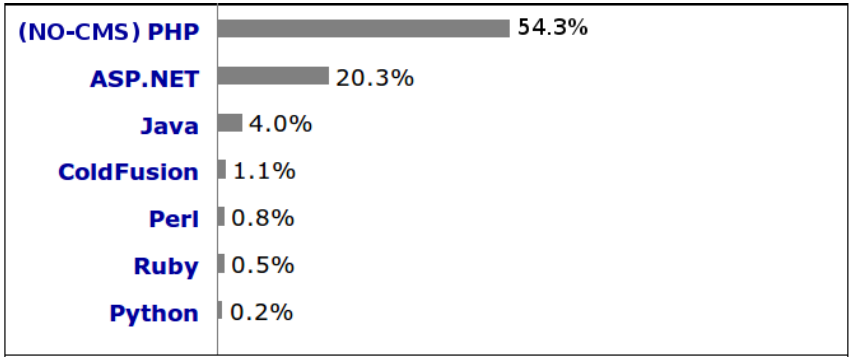
- サーバ側アプリケーション実装のSP対応
  - 新たなWebアプリケーション開発環境の台頭
  - 新たな通信形式の登場

Webアプリケーション フレームワークの利用状況

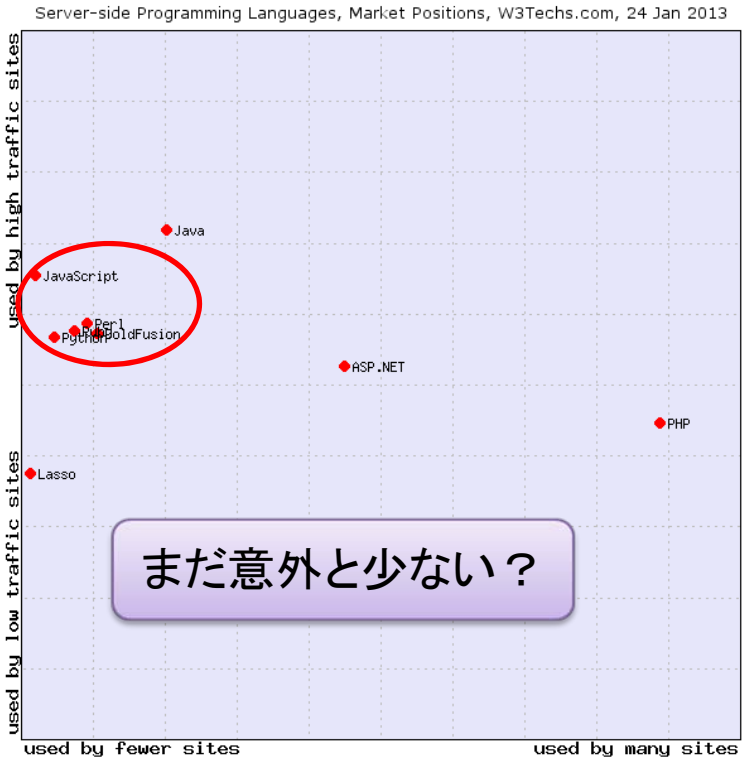
プログラミング言語の利用状況



プログラミング言語の利用状況（CMS抜き）



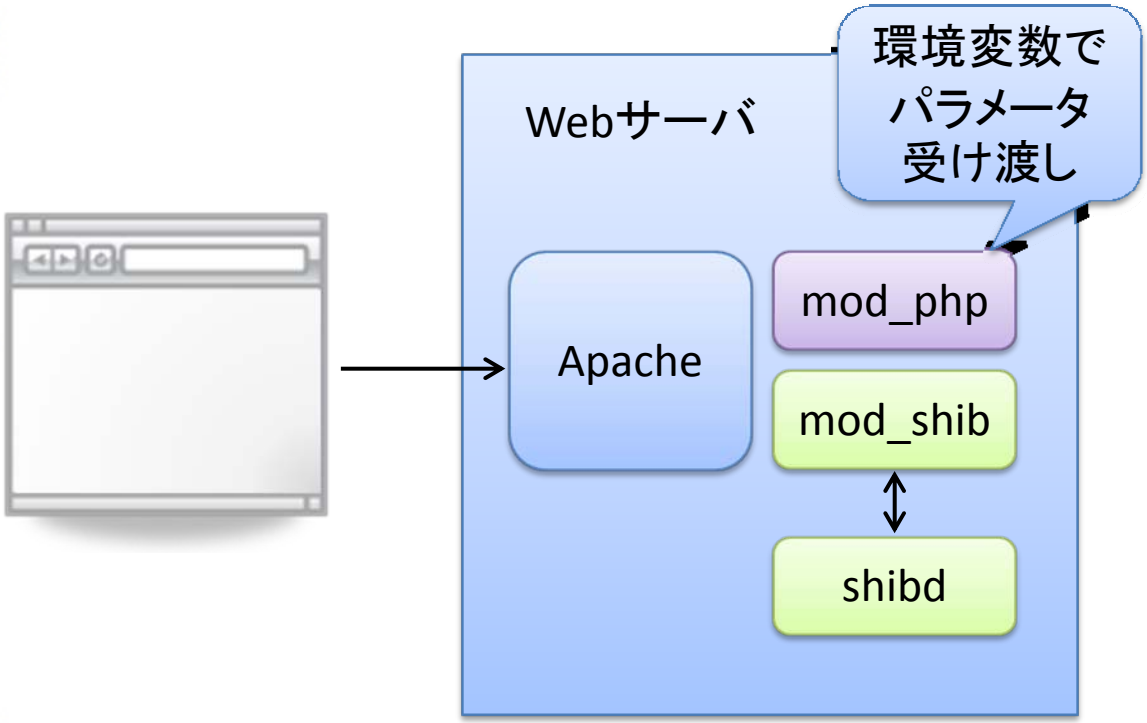
高トラフィックサイトでの利用と
多くのサイトでの利用



引用元： http://blog.websitesframeworks.com/

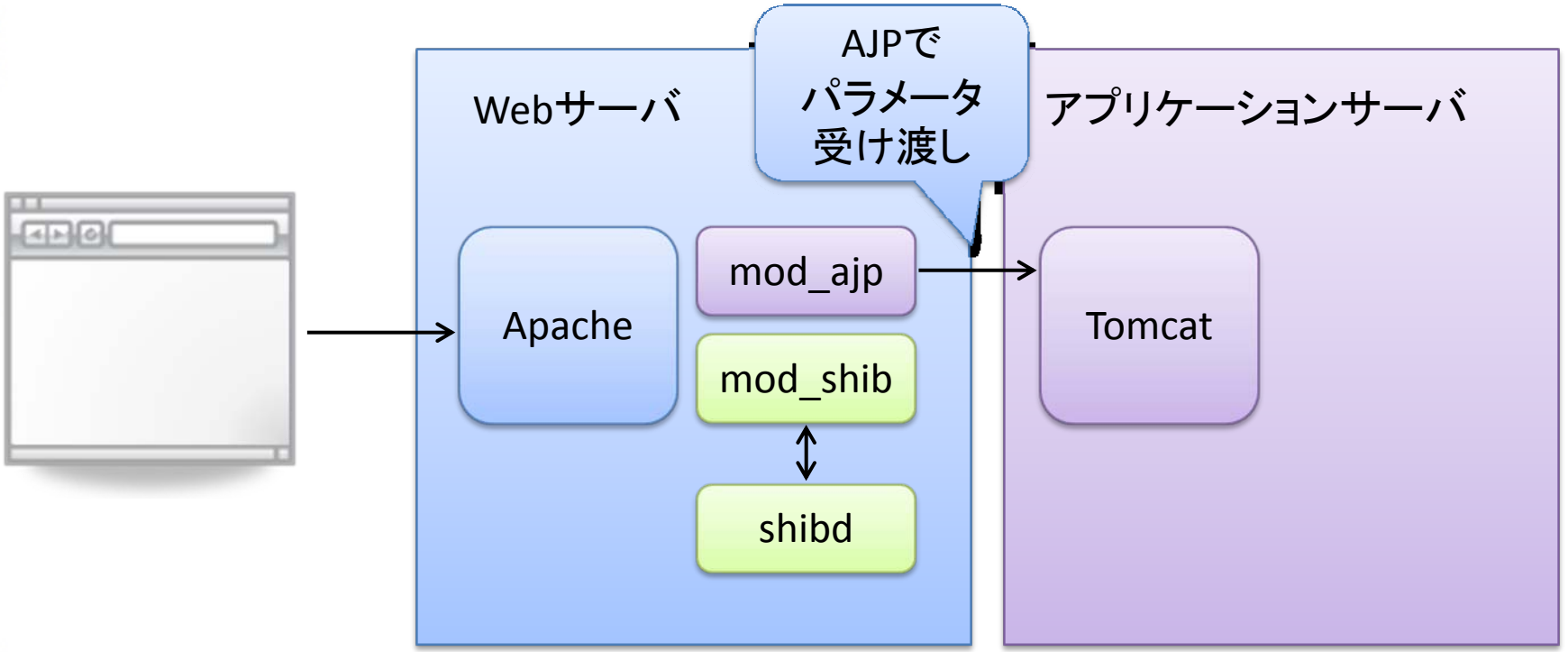
アプリケーションのSSO連携

- PHPの場合



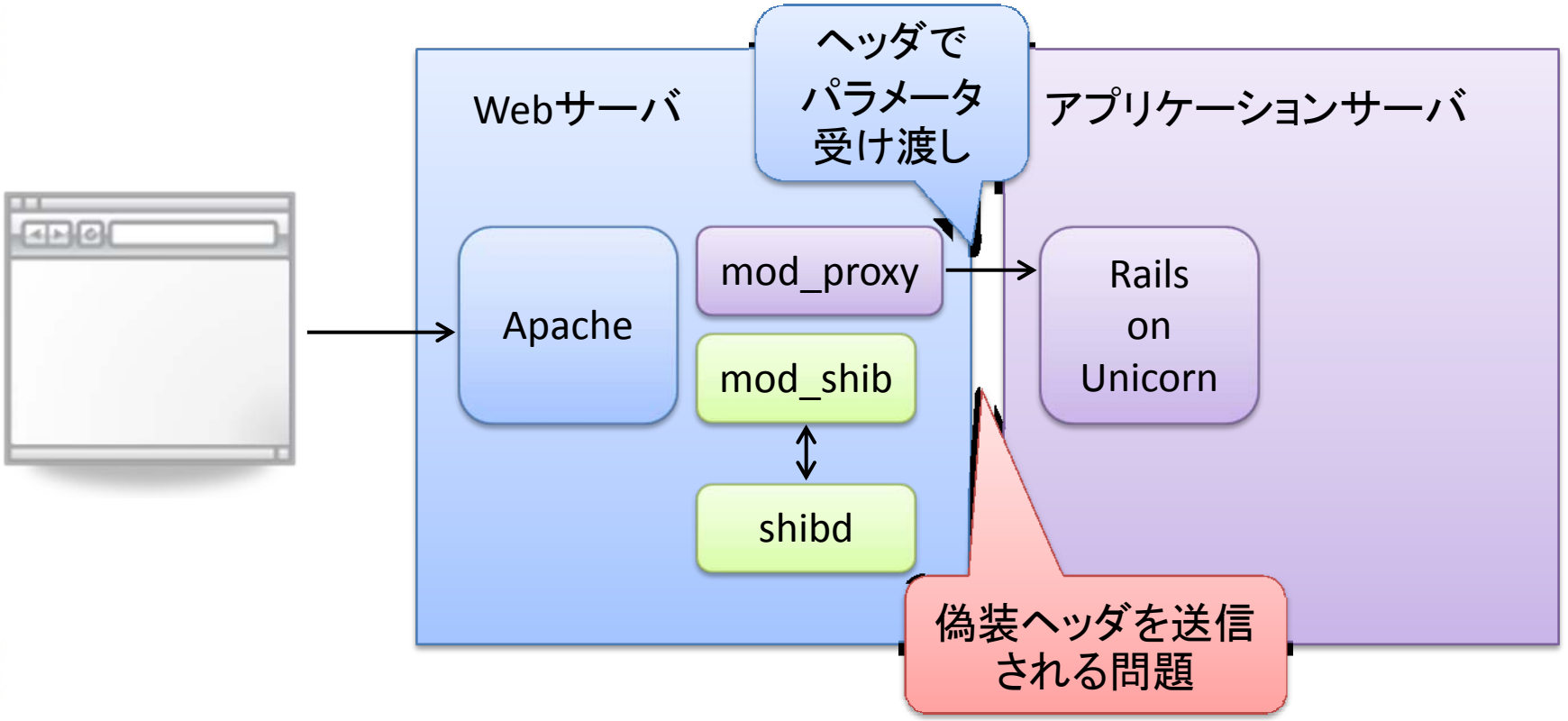
アプリケーションのSSO連携

- Javaの場合



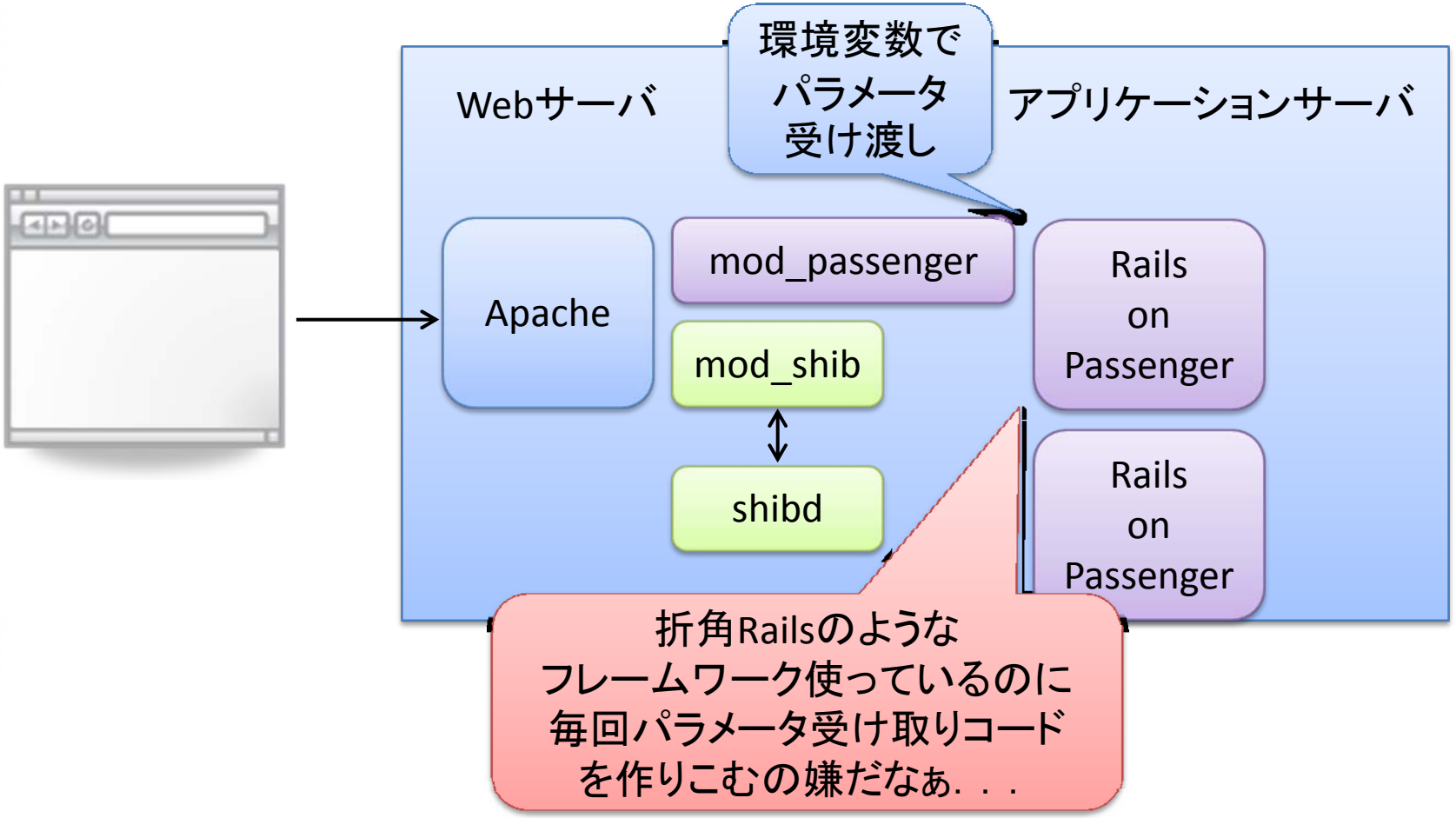
アプリケーションのSSO連携

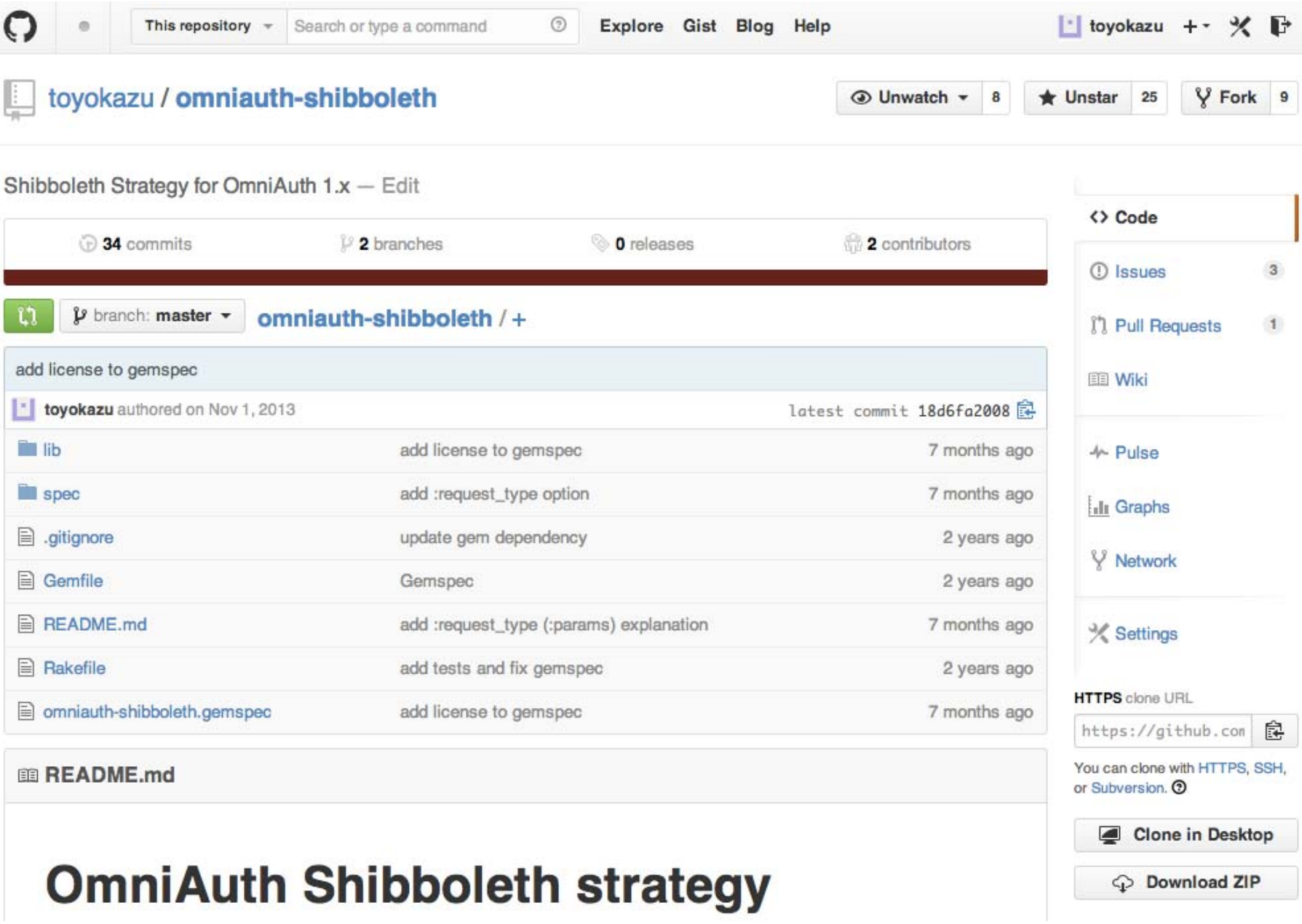
- Rubyの場合（プロキシ型）



アプリケーションのSSO連携

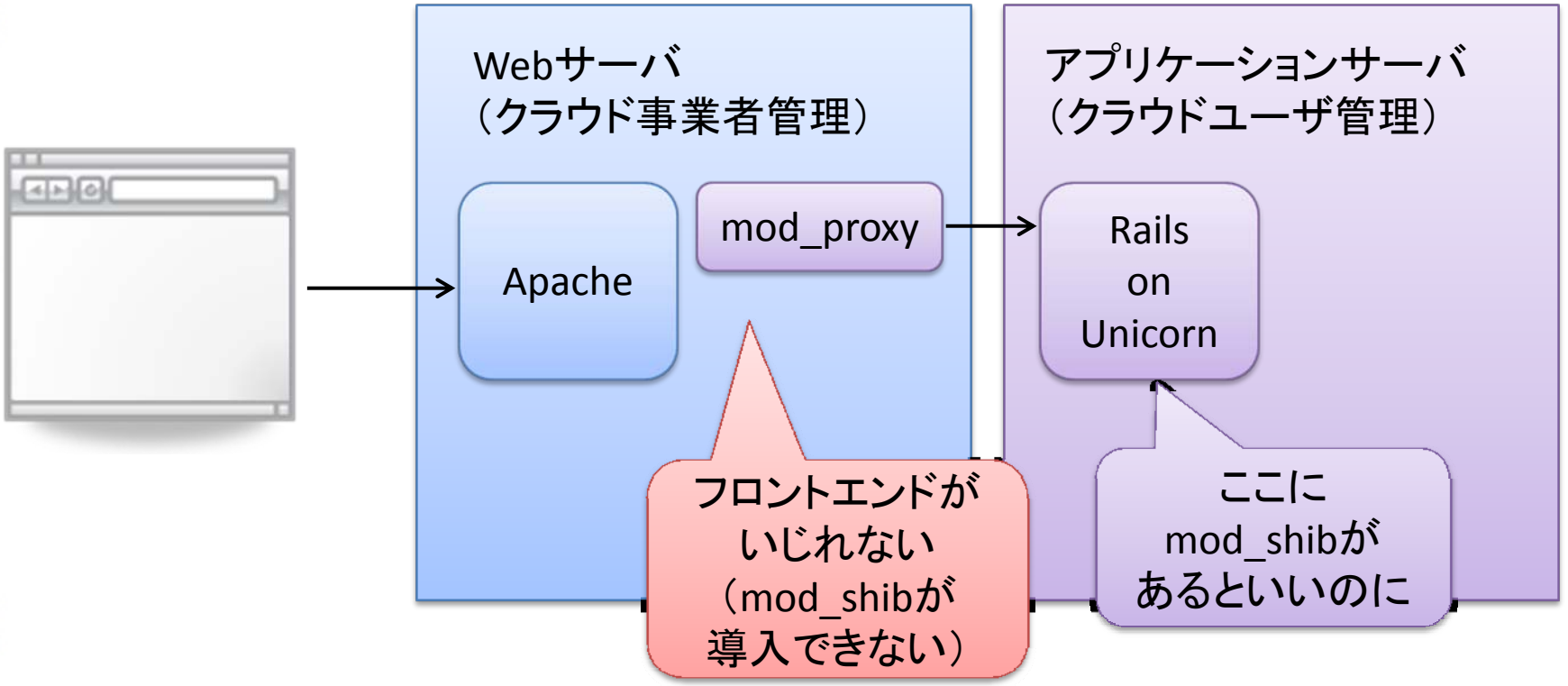
- Rubyの場合（モジュール型）




This repository
Search or type a command
Explore
Gist
Blog
Help
toyokazu
PUBLIC
toyokazu / omniauth-shibboleth
Unwatch 8
Unstar 25
Fork 9
Shibboleth Strategy for OmniAuth 1.x — Edit
34 commits
2 branches
0 releases
2 contributors
branch: master
omniauth-shibboleth / +
add license to gemspec
toyokazu authored on Nov 1, 2013
latest commit 18d6fa2008
lib add license to gemspec 7 months ago
spec add :request_type option 7 months ago
.gitignore update gem dependency 2 years ago
Gemfile Gemspec 2 years ago
README.md add :request_type (:params) explanation 7 months ago
Rakefile add tests and fix gemspec 2 years ago
omniauth-shibboleth.gemspec add license to gemspec 7 months ago
README.md
OmniAuth Shibboleth strategy
Code
Issues 3
Pull Requests 1
Wiki
Pulse
Graphs
Network
Settings
HTTPS clone URL
https://github.com
You can clone with HTTPS, SSH, or Subversion.
Clone in Desktop
Download ZIP

アプリケーションのSSO連携

- Rubyの場合（クラウド（例：Heroku））



京都産業大学
KYOTO SANGYO UNIVERSITY

This repository | Search or type a command | Explore | Gist | Blog | Help | toyokazu

toyokazu / rack-saml

Unwatch 3 | Unstar 16 | Fork 3

SAML (Shibboleth SP) middleware for Rack — Edit

18 commits | 1 branch | 0 releases | 3 contributors

branch: master | rack-saml / +

add license to gemspec

Limitations

AuthnRequest Signing and Response Encryption

Current implementation supports only Onelogin SAML assertion handler. It does not support to sign AuthnRequest and encrypt Response. So thus, the assertion encription function should be disabled at IdP side for rack-saml SPs.

rack-saml.gemspec

README.md

SAML (Shibboleth) SP middleware for Rack

Code
Issues 2
Pull Requests 0
Wiki

https://github.com/

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop

Download ZIP

Ruby-opensamlのようなNative SAML実装が求められている

Keep Innovating.

新たな通信形式の登場

- ハイパフォーマンスブラウザネットワーキング
  - HTTP/2.0
  - WebSocket
  - WebRTC

Node.jsのようなサーバでのサービス提供

2014年5月発売

PersonaとShibbolethの連携

- Persona
  - Mozillaプロジェクトが立ち上げたSSOのフレームワーク
  - BrowserIDというプロトコルを用いてSSOを実現
  - 一旦IdPで認証されると、その後はブラウザとSPが直接認証処理を行える
    - プライバシの確保（賛否両論）
    - P2Pのように接続先（検証者）が多い場合有効
  - Node.jsベースで認証サーバを実装
- 調査結果
  - 連携ができることを確認
  - 想定していたユーザ・SP間のローカル認証機能は未実装
  - 現在残念ながらコミュニティプロジェクトに格下げ

WebRTCでの認証

- SkyWay
  - NTTコミュニケーションズのブラウザ間リアルタイム通信フレームワーク
  - http://nttcom.github.io/skyway/
  - PeerJSを介してセッションを確立
    - ブラウザ間でP2P通信が可能
  - DTLS-SRTPにより端末間の相互認証・通信路暗号化を実現

P2PでもPeerJSを用いる場合は、PeerJSにpassport-samlのようなSAML認証機能を追加すれば十分？

まとめ

- 学認のサービス連携推進のための機能開発
- 取り組んだ内容（1）
  - 拡張機能を用いた利便性・セキュリティの向上？
- 取り組んだ内容（2）
  - Shibboleth SPとアプリケーションフレームワークの親和性向上？
- 取り組んでいる内容
  - 新しい形式のサービスへの対応？