OpenAMトレーニング

OpenAMでシングルサインオンを実現しよう！

野村総合研究所のOpenStandia（オープンスタンディア）は、おかげさまで、2006年のサービス開始から2011年までの5年間で契約数累計が1,000件を突破いたしました！

株式会社 野村総合研究所 情報技術本部 オープンソースソリューションセンター（OSSC）
Mail：ossc@nri.co.jp　Web：http://openstandia.jp/

オープンソースまるごと
OpenStandia™
Open Source Technology

NRI 未来創発
Dream up the future.

アジェンダ

- Section0：自己紹介
- Section1：OpenAM概要
- Section2：OpenAMインストール
- Section3：連携先システムとのSSO
- Section4：まとめ

Section0

自己紹介

自己紹介

- 林田　敦

- 所属部署
  - オープンソースソリューション推進室（OSSC）
  - OSSを使ったシステム構築から運用までワンストップでサポート
  - 対象OSSは50種類以上

- 担当
  - SI
  - システム運用維持管理
  - 製品開発
  - OSSサポート

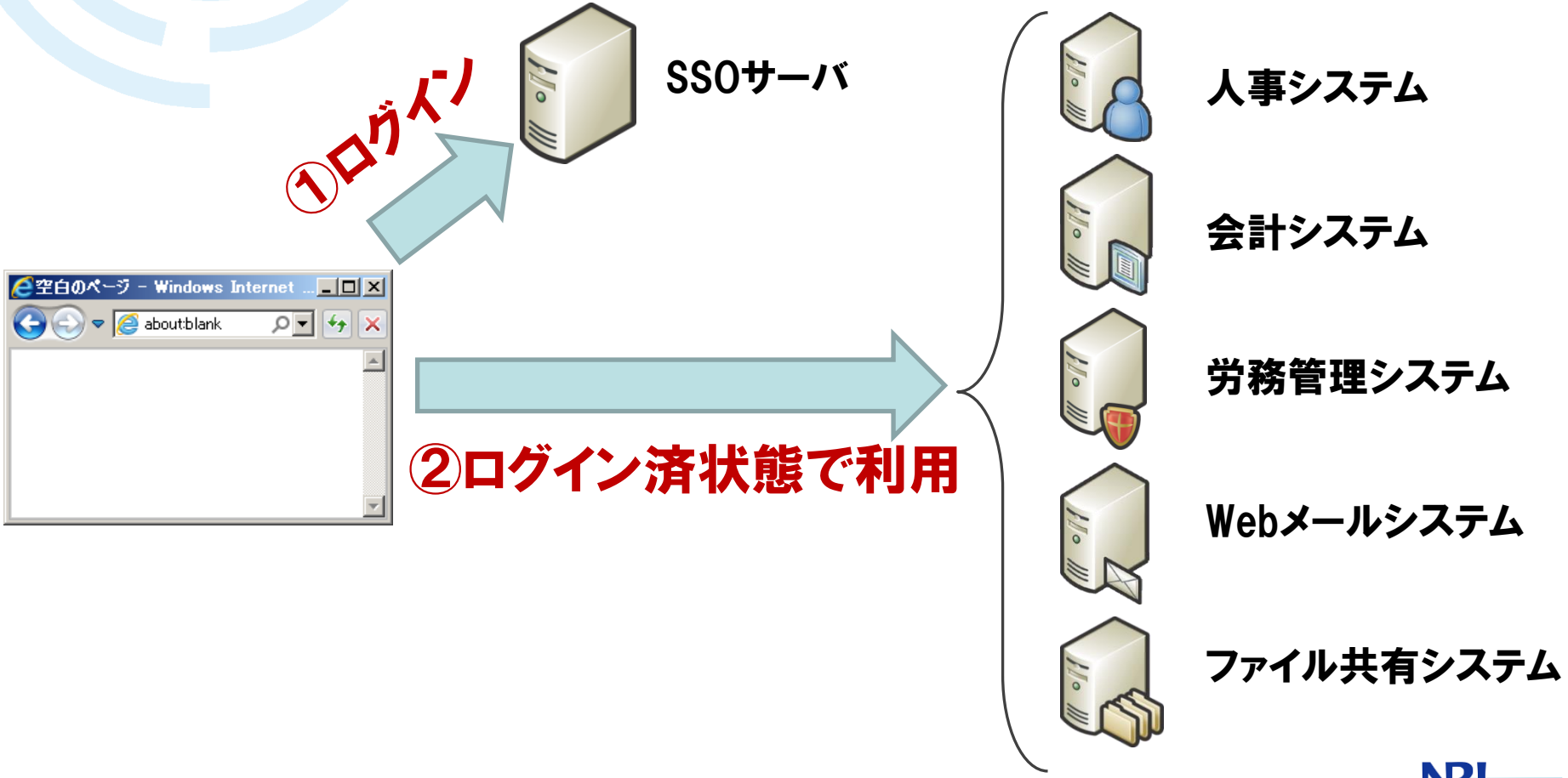
Section1

OpenAM概要

シングルサインオン(SSO)とは

オープンソースまるごと
OpenStandia™
Open Source Technology

- 一度のログイン(サインオン)で複数のアプリケーションがログイン済状態で利用できる仕組み



NRI 未来創発
Dream up the future.

シングルサインオン(SSO)とは

- ユーザにとってのメリット
  - システムごとの認証が無くなって手間が減る
  - ID/PWの管理(記憶や更新)が楽になる
- システム管理者にとってのメリット
  - アカウント情報の一元化により、運用の手間が減る
  - ユーザの認証方法を変更しやすい(指紋認証機能の追加など)
  - ユーザのID/PW管理一元化に伴うセキュリティの向上

シングルサインオン（SSO）とは

SSO需要の高まり

▶企業内システム数の増加

✓複雑化したシステムをよりスマートに利用／管理したい

▶クラウドサービスの増加

✓salesforceなどのクラウドサービスも企業内システムとシームレスに使いたい

▶求められる企業コンプライアンスの高まり

✓不正ログイン等のセキュリティリスクを低減したい

シングルサインオン（SSO）とは

OpenStandiaのSSO導入事例

ヘルスケア

✓課題

- 顧客の利便性を向上させるため、複数の自社サービスと、顧客システムとをシングルサインオン対応したい。

✓ユーザ数

- 10,000人

大手医療機器メーカー

✓課題

- 様々なアプリケーションに対応でき、将来のサービス追加にも柔軟に対応できる社内認証基盤が欲しい。

✓ユーザ数

- 10,000人

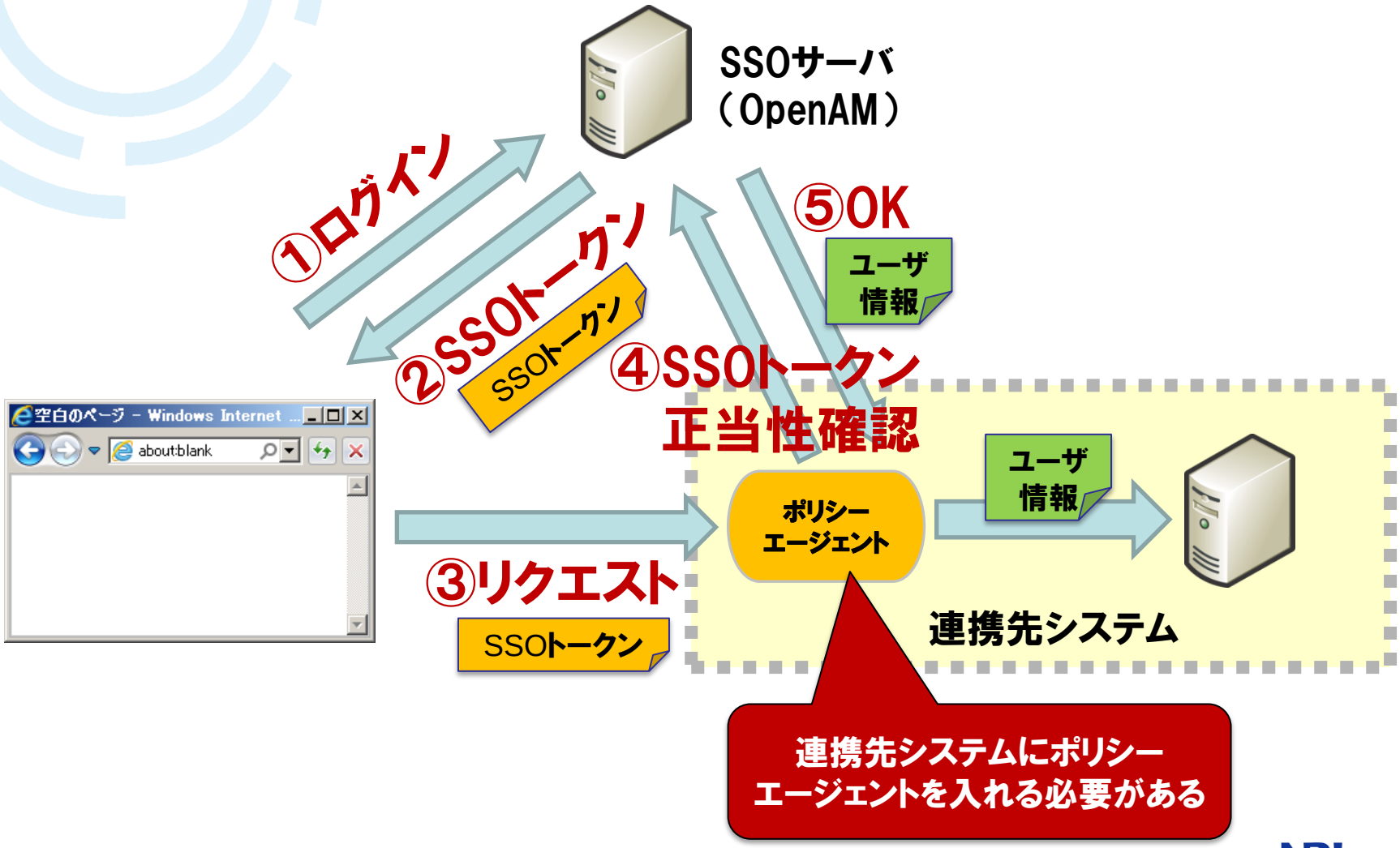
OpenAMとは

- SSOを実現するためのOSS
  - 旧Sun Microsystems社の商用製品(OpenSSO)がベースであるため高品質かつ多機能
  - ForgeRock社が継続開発中
  - Javaで実装されたWebアプリケーションでOS非依存
  - CDDL(Common Development and Distribution License)ライセンスで、ソースコードを無償で使用、改変、再配布可能
  - 最新の安定バージョンは10.0.1

OpenAMのSSO方式

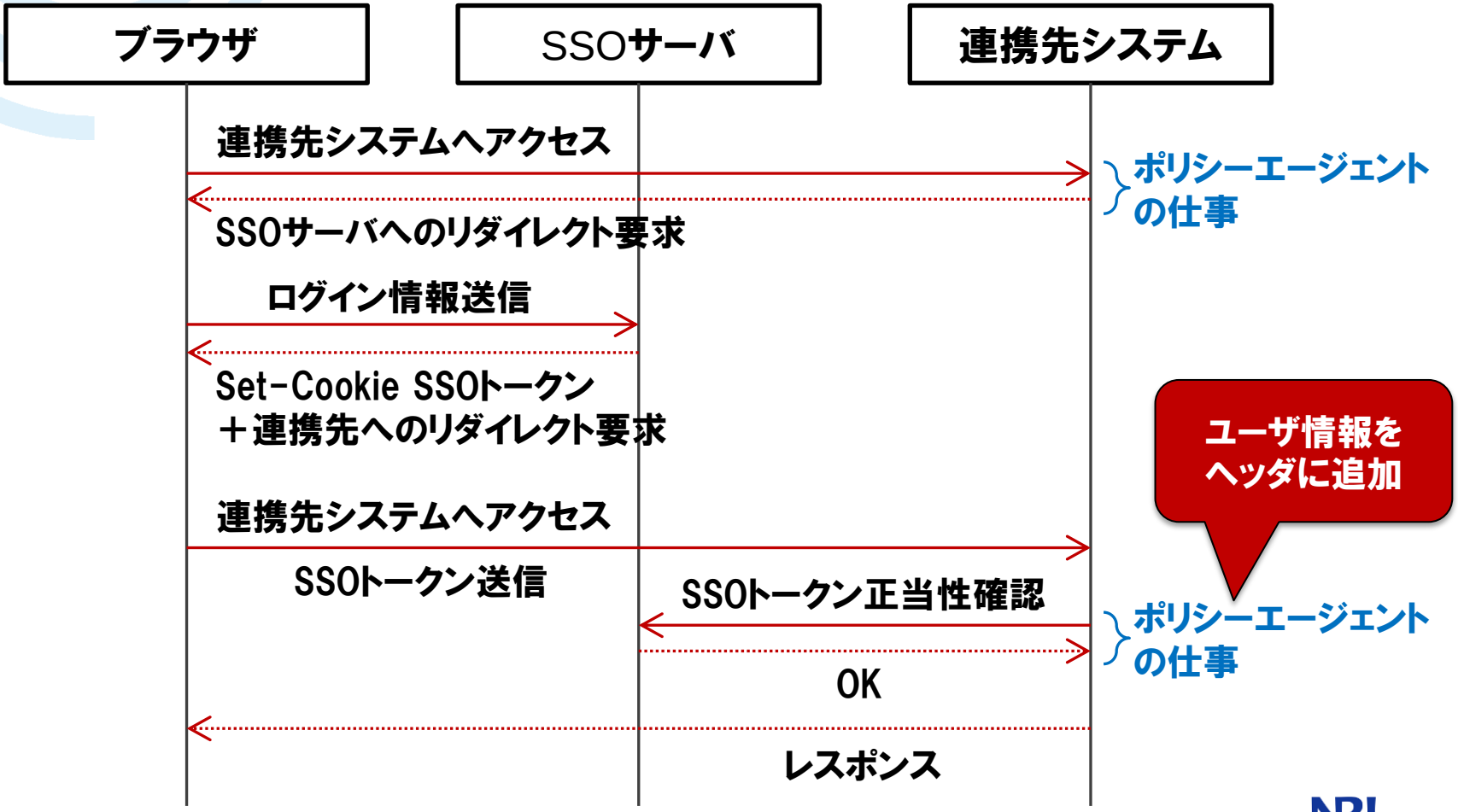
SSO方式	説明
エージェント方式	アプリケーションが動作するサーバに直接エージェントを導入する方式。
リバースプロキシ方式	リバースプロキシサーバ(通常はApache)にエージェントを導入し、バックエンドにいる複数のアプリケーションサーバに対してリバースプロキシする方式。
[参考] SAML	SAMLとは認証情報を表現するためのXML仕様。おもにクロスドメインのサイトやSalesforce、GoogleAppsとSSOする際に採用する方式。
[参考] 代理認証方式	代理認証とは、ユーザからのログインリクエストをエミュレートし、認証を代行すること。OpenIGと連携することで、代理認証が可能となる。 連携先システムで、HTTPヘッダから認証情報を取得するカスタマイズが出来ない際に採用する方式。

エージェント方式



SSO方式

- OpenAMでのログインが完了したらSSOトークン（Cookie）を発行します



SSOトークンについて

SSOトークンの正体

- 認証トークン、認証クッキーとも呼ばれる
- 標準では「iPlanetDirectoryPro」という名前のクッキー

認証されたユーザの識別方法

- ポリシーエージェントがHTTPリクエストヘッダにユーザ識別情報(例:ログインID)を付与する
- アプリケーションはHTTPリクエストヘッダからユーザ識別情報を取得する

```
GET / HTTP/1.1
Accept                          : text/html
Accept-Language                 : ja-JP
Connection                      : Keep-Alive
Cookie :iPlanetDirectoryPro=KU(GKU(#LGSJVUUR749
```

エージェント通過後

```
GET / HTTP/1.1
Accept                          : text/html
Accept-Language                 : ja-JP
Connection                      : Keep-Alive
Cookie : iPlanetDirectoryPro=KU(GKU(#LGSJVUUR749
LOGINID                         : a-hayashida@hoge.com
```

ポリシーエージェントについて

ポリシーエージェントとは

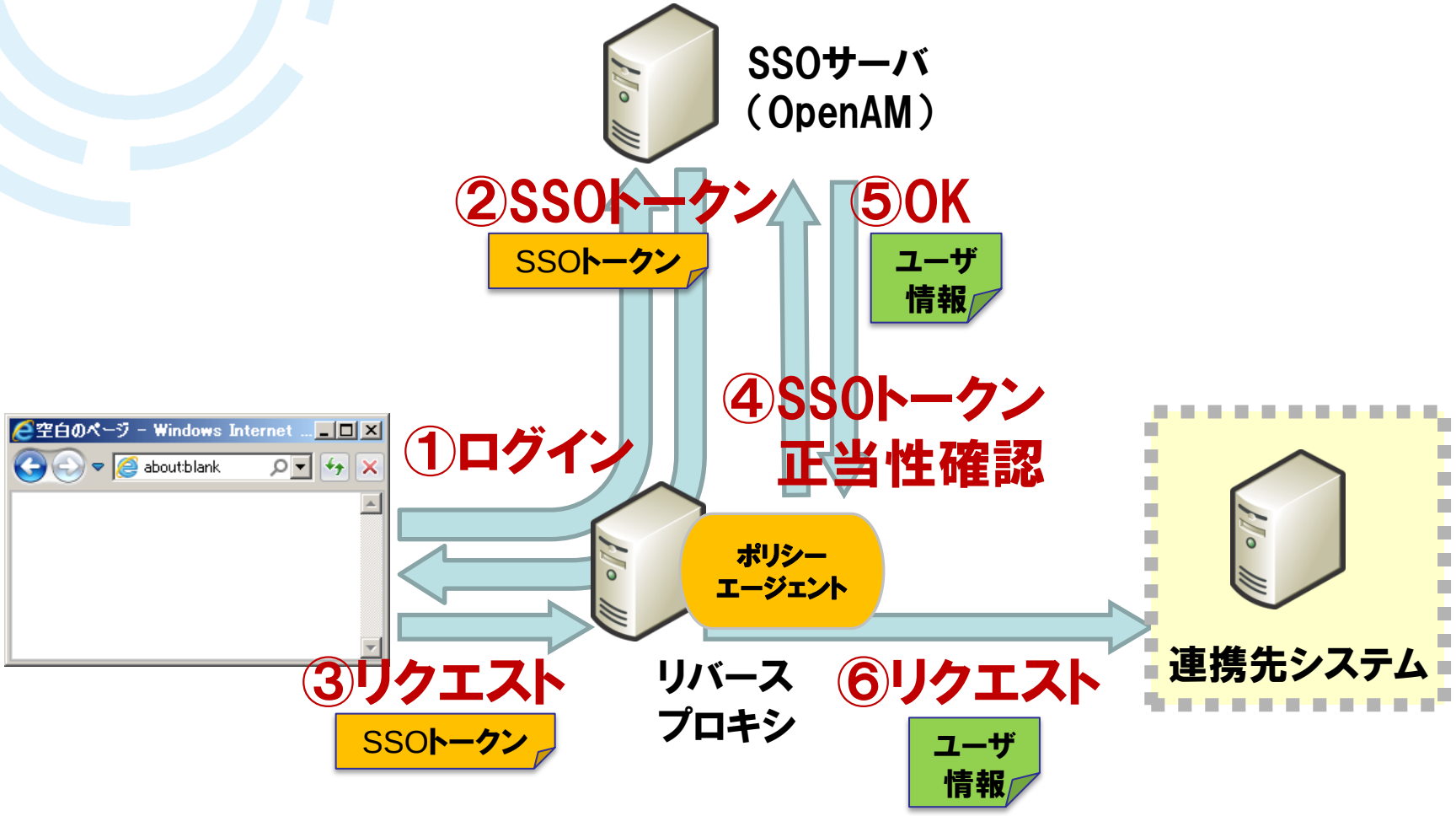
- SSO対象の連携先サーバへインストールするモジュール
- SSOサーバと通信し、認証／認可に必要な情報を取得する
- ユーザからリクエストがあるとそのURLを評価し、拒否／許可を判定する

提供されているエージェントの種類

- Webポリシー エージェント
  - ✓Apache 2.0、2.2用
  - ✓Microsoft IIS 6.0、 7.0 等
- J2EEポリシー エージェント
  - ✓Tomcat 6.0
  - ✓JBoss v 4.2 & v 5.x 等

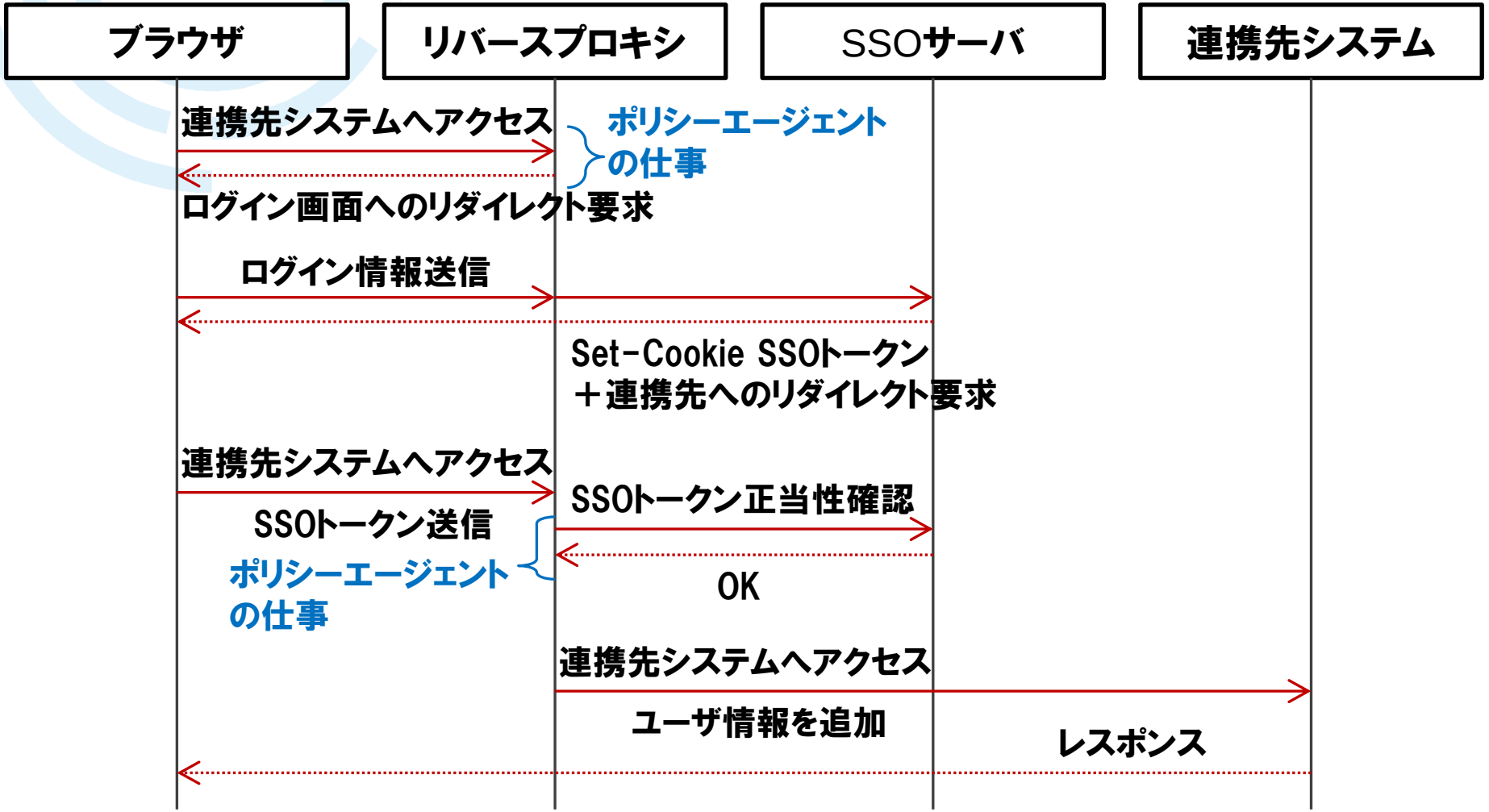
SSO方式

リバースプロキシ方式



SSO方式

リバースプロキシがSSOをドライブします



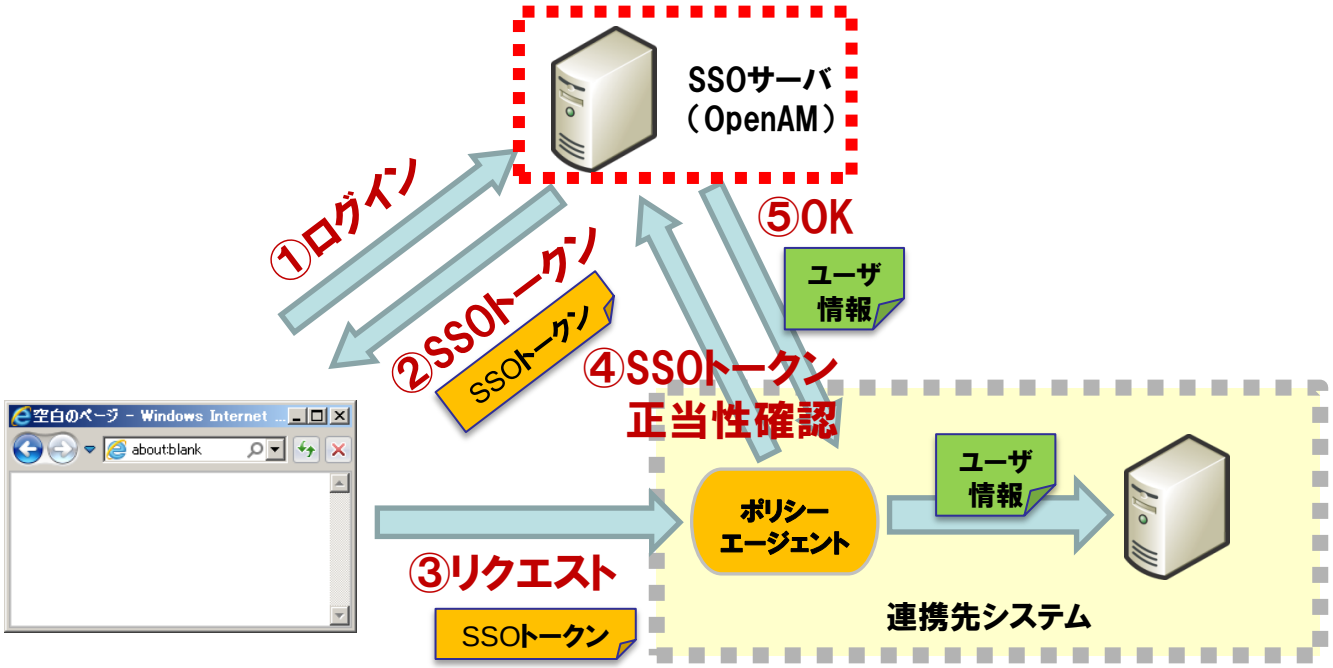
Section2

OpenAMインストール

OpenAMインストール

概要

- OS
  - ✓CentOS6
- OpenAM
  - ✓10.0.1
  - ✓ダウンロード元：http://forgerock.org/openam-archive.html



OpenAMインストール

ネットワークのセットアップ（FQDNの登録）

- /etc/sysconfig/network

```
HOSTNAME=openam.nriossc.co.jp
```

- /etc/hosts

```
192.175.204.101   openam.nriossc.co.jp
```

環境整備（動作検証のため）

- /etc/sysconfig/selinux

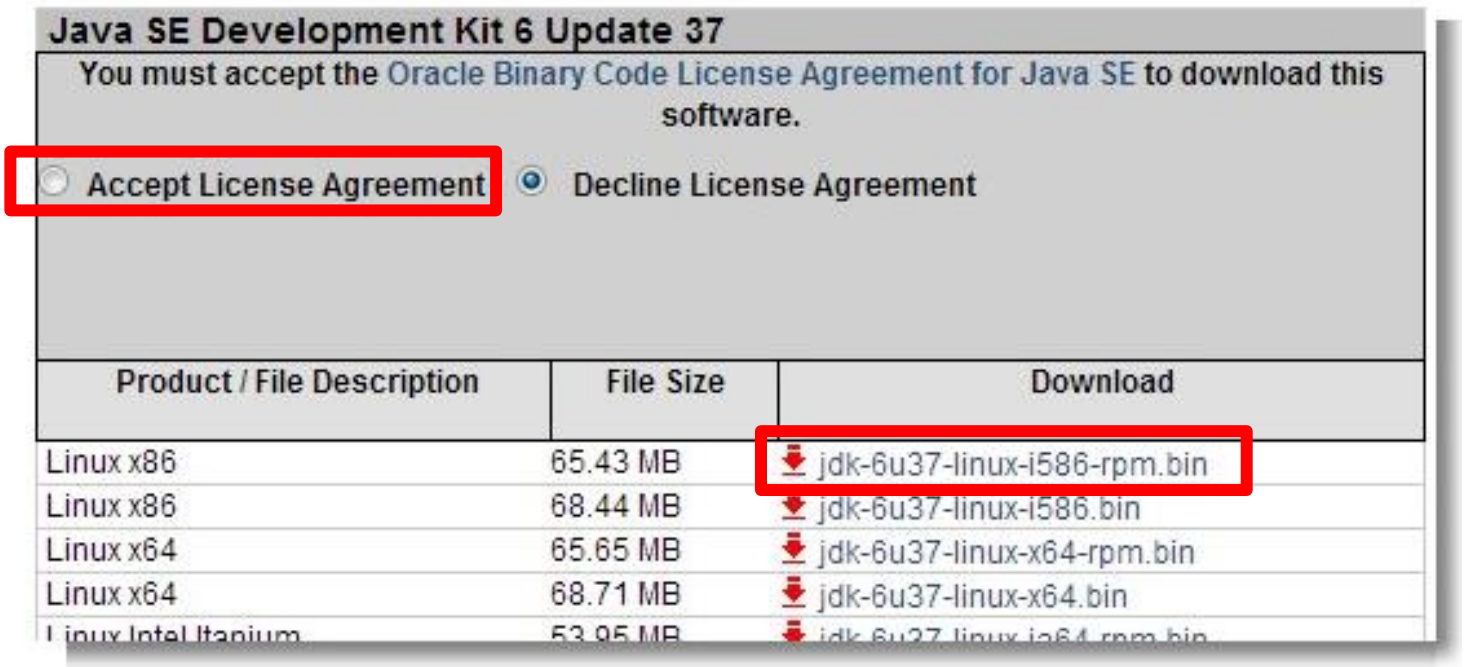
```
#SELINUX=enforcing
SELINUX=disabled
```

- ファイアウォールをOFF

```
$ service iptables stop
$ chkconfig iptables off
```

JDKのダウンロード

- ORACLE公式サイトからJDKをダウンロード
- http://www.oracle.com/technetwork/java/javase/downloads/jdk6u37-downloads-1859587.html
  - ✓「Accept License Agreement」をクリックしてから「jdk-6u37-linux-i586-rpm.bin」をクリック



Java SE Development Kit 6 Update 37

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement　Decline License Agreement

Product / File Description	File Size	Download
Linux x86	65.43 MB	jdk-6u37-linux-i586-rpm.bin
Linux x86	68.44 MB	jdk-6u37-linux-i586.bin
Linux x64	65.65 MB	jdk-6u37-linux-x64-rpm.bin
Linux x64	68.71 MB	jdk-6u37-linux-x64.bin
Linux Intel Itanium	53.95 MB	jdk-6u37-linux-ia64-rpm.bin

JDKのインストール

インストール

```
# chmod 755 jdk-6u37-linux-i586-rpm.bin
# ./jdk-6u37-linux-i586-rpm.bin
# java -version
java version "1.6.0_37"
```

以下のようなエラーが出た場合は、ld-linux.so.2をインストールする

```
/lib/ld-linux.so.2: bad ELF interpreter: No such file or directory

# yum install ld-linux.so.2
```

Tomcatのインストール

Tomcatをダウンロードしてインストール

```
# wget
http://ftp.tsukuba.wide.ad.jp/software/apache/tomcat/tomcat-
6/v6.0.37/bin/apache-tomcat-6.0.37.tar.gz

# tar zxvf apache-tomcat-6.0.37.tar.gz
# mv apache-tomcat-6.0.37 /usr/share/tomcat6
```

/root/.bash_profile

```
export JAVA_HOME=/usr/java/default/
export JAVA_OPTS="-Xmx1024m -XX:MaxPermSize=256m"
```

OpenAMのインストール

warファイルをダウンロードしてTomcatにデプロイする

```
# wget
http://download.forgerock.org/downloads/openam/openam10/
10.0.1/openam_10.0.1.war

# cp -p openam_10.0.1.war  /usr/share/tomcat6/webapps/
```

Tomcatを起動する

```
# /usr/share/tomcat6/bin/startup.sh
```

OpenAMインストール

OpenAM初期設定：Step1

- http://openam.nriossc.co.jp:8080/openam/

OpenAM初期設定：Step2

- amAdmin（OpenAM管理者）のパスワードを設定（例：adminpassword）

OpenAM初期設定：Step3

- サーバーURL　　：http://openam.nriossc.co.jp:8080
- Cookieドメイン　　：.nriossc.co.jp
- プラットフォームロケール、設定ディレクトリはそのままでOK

OpenAM初期設定：Step4

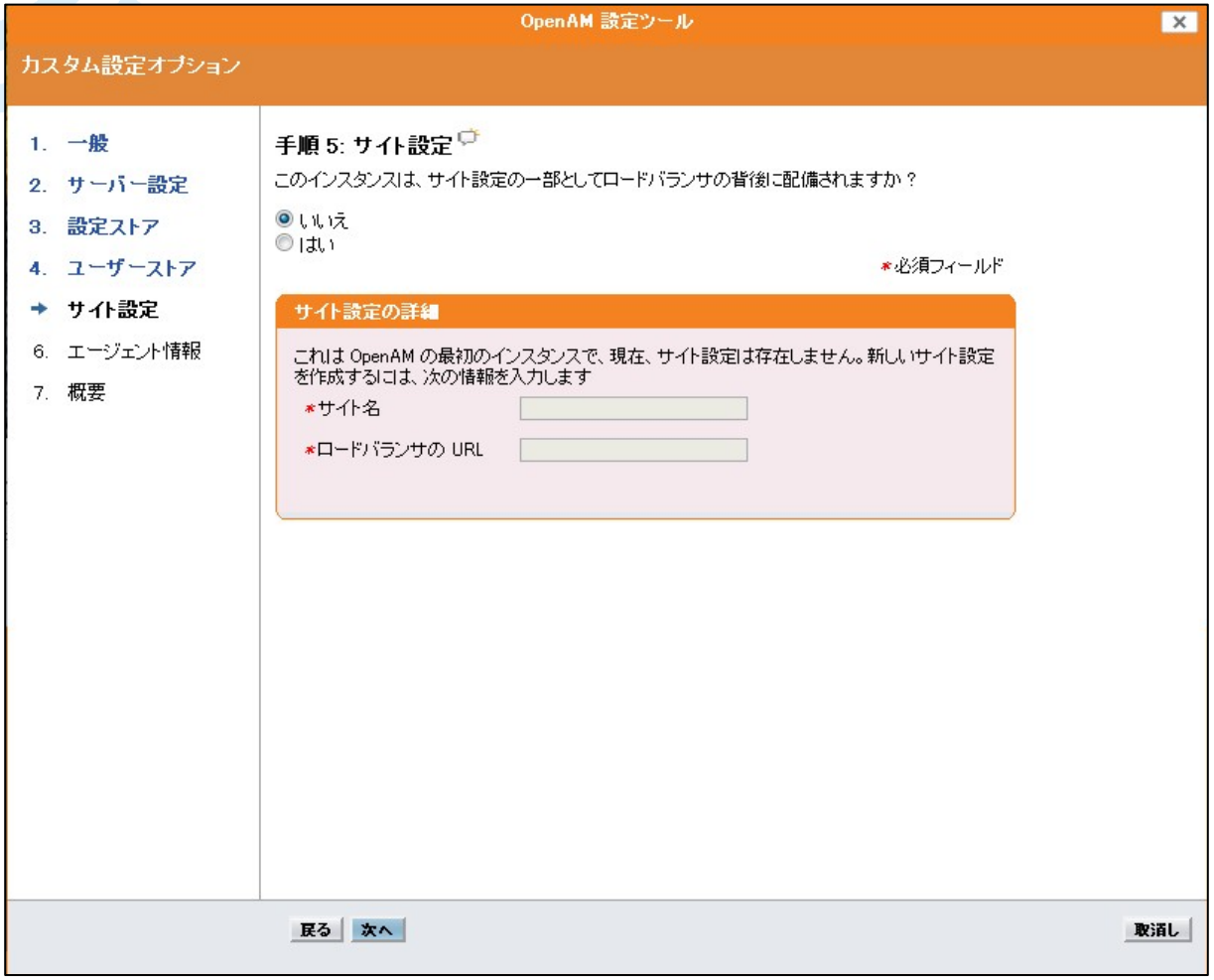
- 「最初のインスタンス」を選択して次へ

OpenAM初期設定：Step5

- 「OpenAMのユーザーデータストア」を選択して次へ

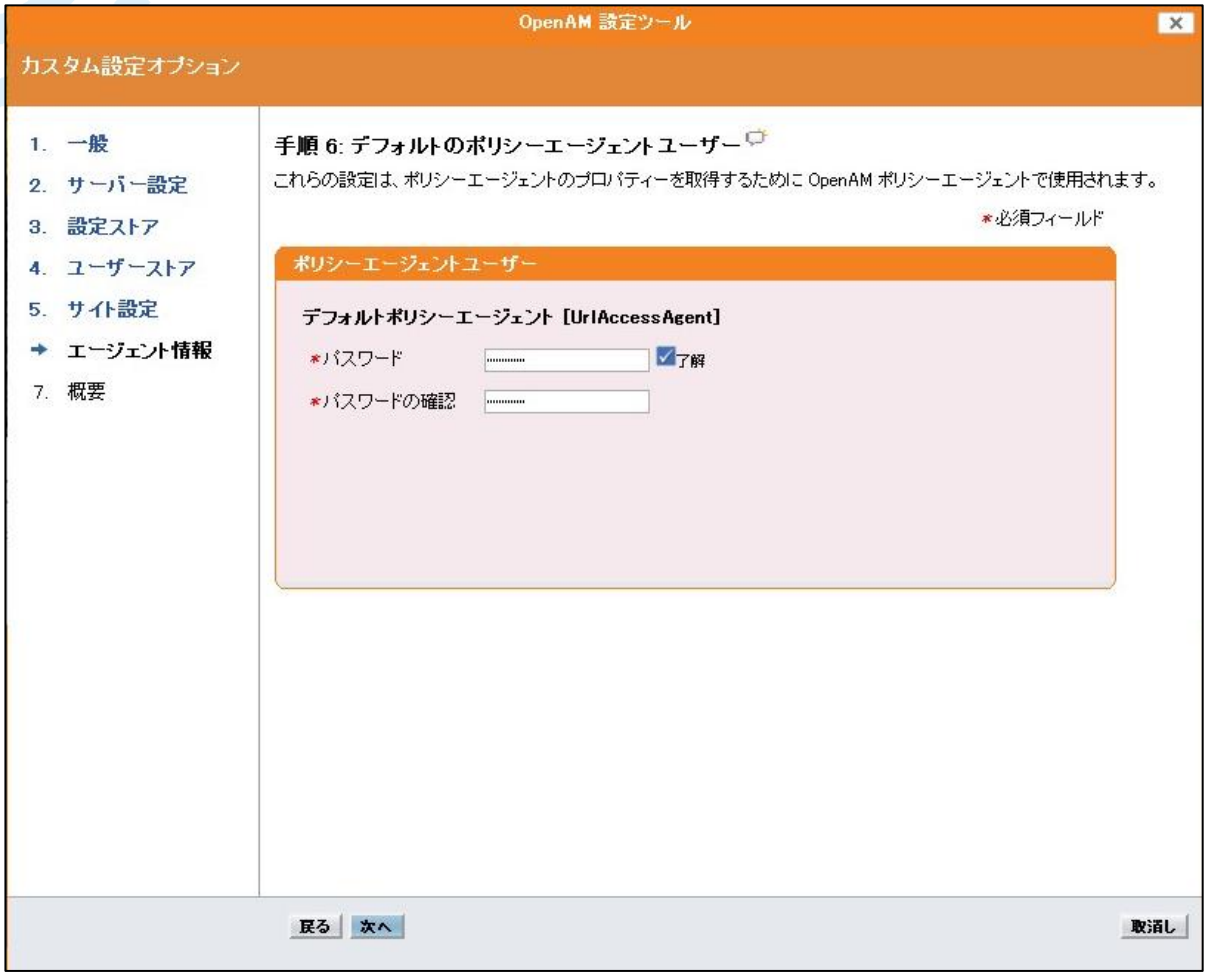
OpenAM初期設定：Step6

- 「いいえ」を選択して次へ



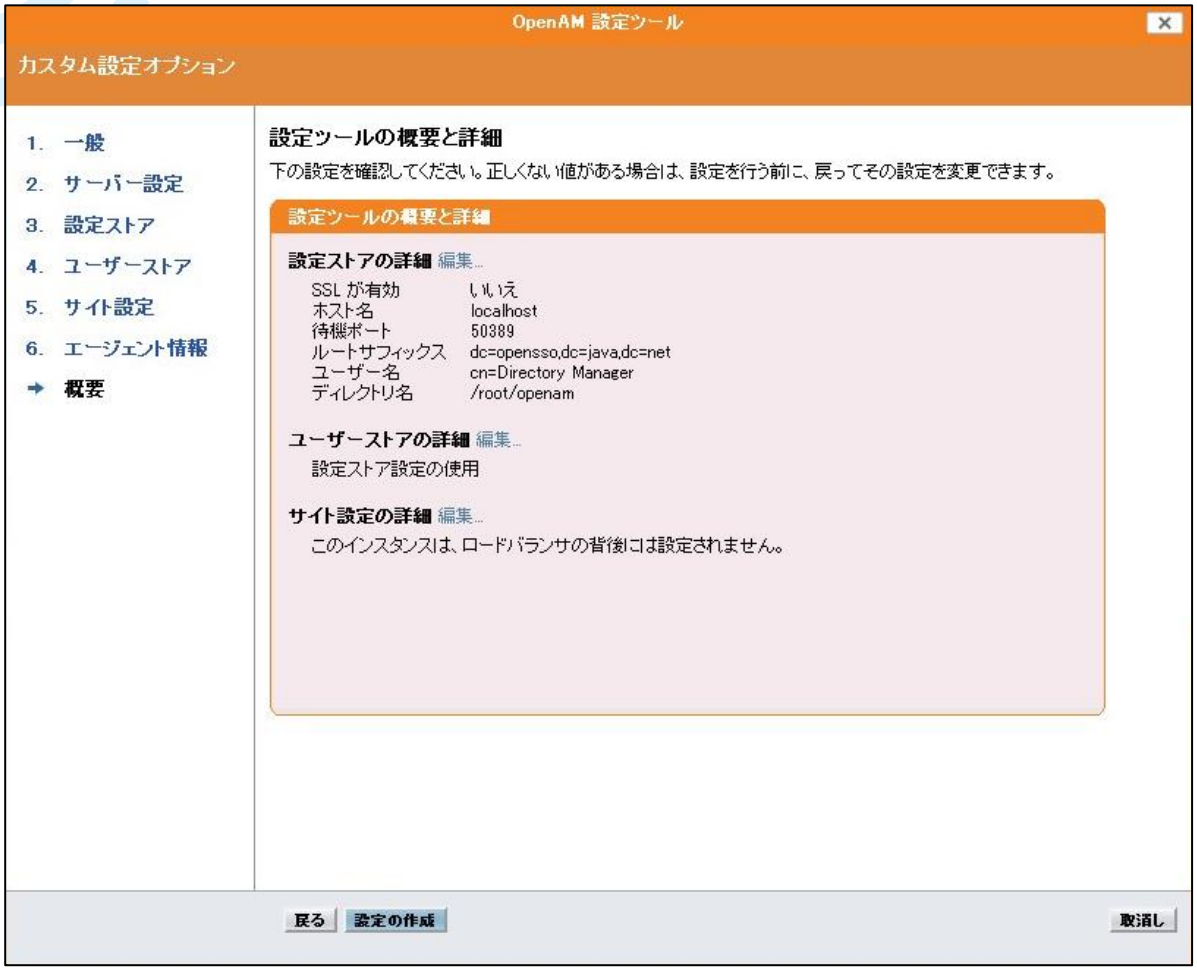
OpenAM初期設定：Step7

▶デフォルトポリシーエージェントのパスワードを設定（例：agentpassword）



OpenAM初期設定：Step8

▶設定内容を確認して「設定の作成」



OpenAM初期設定：Step9

- 設定が実行されます

OpenAMにログイン

- http://openam.nriossc.co.jp:8080/openam にアクセスし、amadmin ユーザーでログインします

OpenAMインストール

OpenAMにログイン

- OpenAM管理コンソールのメインメニュー画面が表示されます

Section3

連携先システムとのSSO

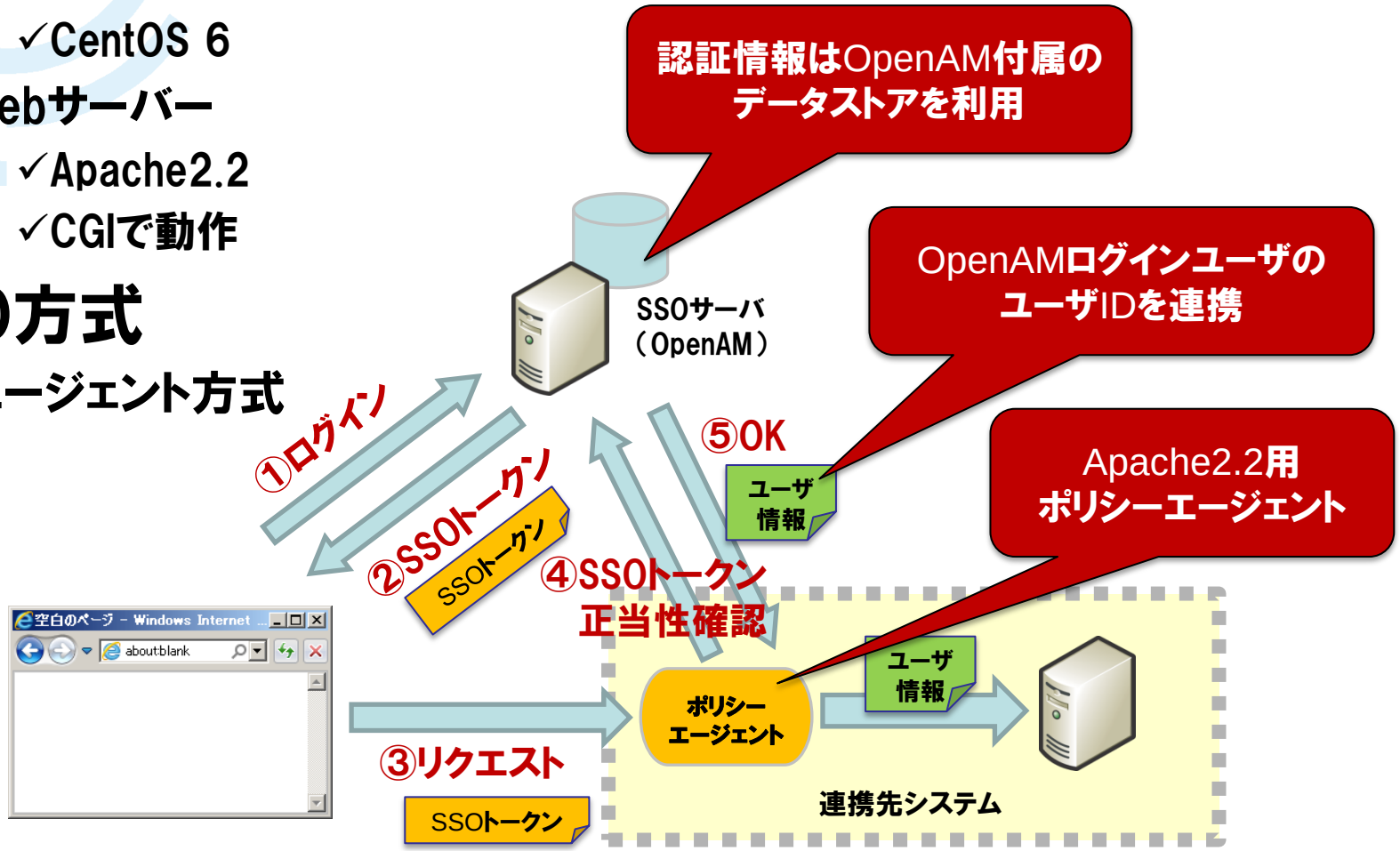
連携先システムとのSSO

連携先システム概要

- OS
  - ✓CentOS 6
- Webサーバー
  - ✓Apache2.2
  - ✓CGIで動作

SSO方式

- エージェント方式



連携先システムとのSSO

連携先システム

- クライアントからのリクエストヘッダの内容を表示するだけのアプリケーション

管理者ユーザー向けサイト HTTPヘッダ SSO実行結果

想定通りのパラメータが送信されていることを確認してください。

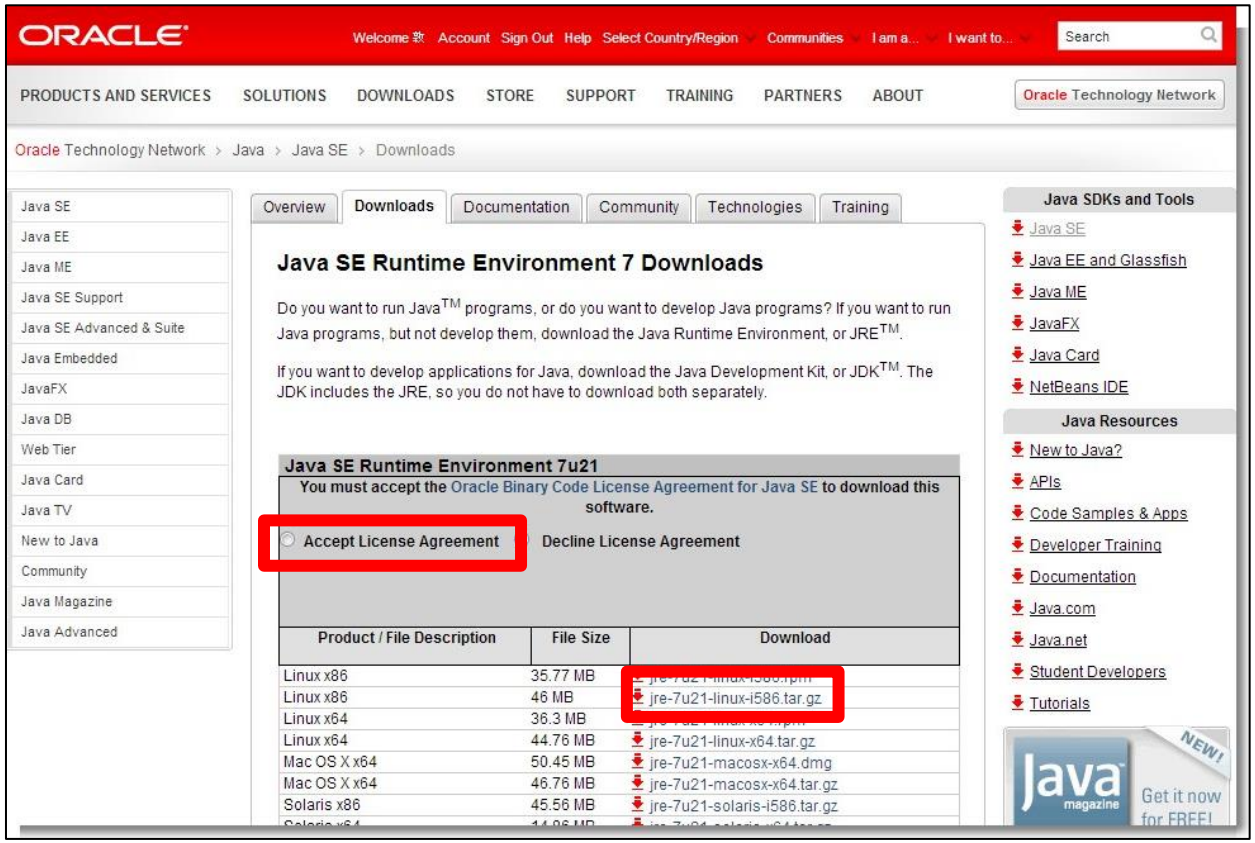
---------- HTTPヘッダ ----------
ACCEPT = text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
ACCEPT_CHARSET = Shift_JIS,utf-8;q=0.7,*;q=0.3
ACCEPT_ENCODING = gzip,deflate,sdch
ACCEPT_LANGUAGE = ja,en-US;q=0.8,en;q=0.6
CONNECTION = keep-alive
HOST = openam-traning-app.nriossc.co.jp
USER_AGENT = Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.31 (KHTML, like Gecko) Chrome/26.0.1410.64 Safari/537.31

---------- リクエストパラメータ ----------

連携先システムの構築

JREをダウンロード

- ORACLE公式サイトからJREをダウンロード（エージェントインストール用）
- http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html



JREを連携先サーバにインストール

ファイルを展開

```
# ls -l
jre-7u21-linux-i586.gz

# tar zxvf jre-7u21-linux-i586.gz
# ls -l
jre1.7.0_21
```

JREを連携先サーバにインストール

JREにPATHを通す

```
# vi ~/.bash_profile
export JAVA_HOME=/ [jreを展開したフルパス] /jre1.7.0_21←追記
export PATH=$PATH:$JAVA_HOME/bin←追記

# source ~/.bash_profile
# java -version
/lib/ld-linux.so.2: bad ELF interpreter←ld-linux.so.2が無くエラー
# yum install ld-linux.so.2

# java -version
java version "1.7.0_21"
Java (TM) SE Runtime Environment (build 1.7.0_21-b11)
Java HotSpot (TM) Client VM (build 23.21-b01, mixed mode)
```

ネットワーク設定

- SSOサーバの/etc/hostsファイルに連携先サーバを追記する

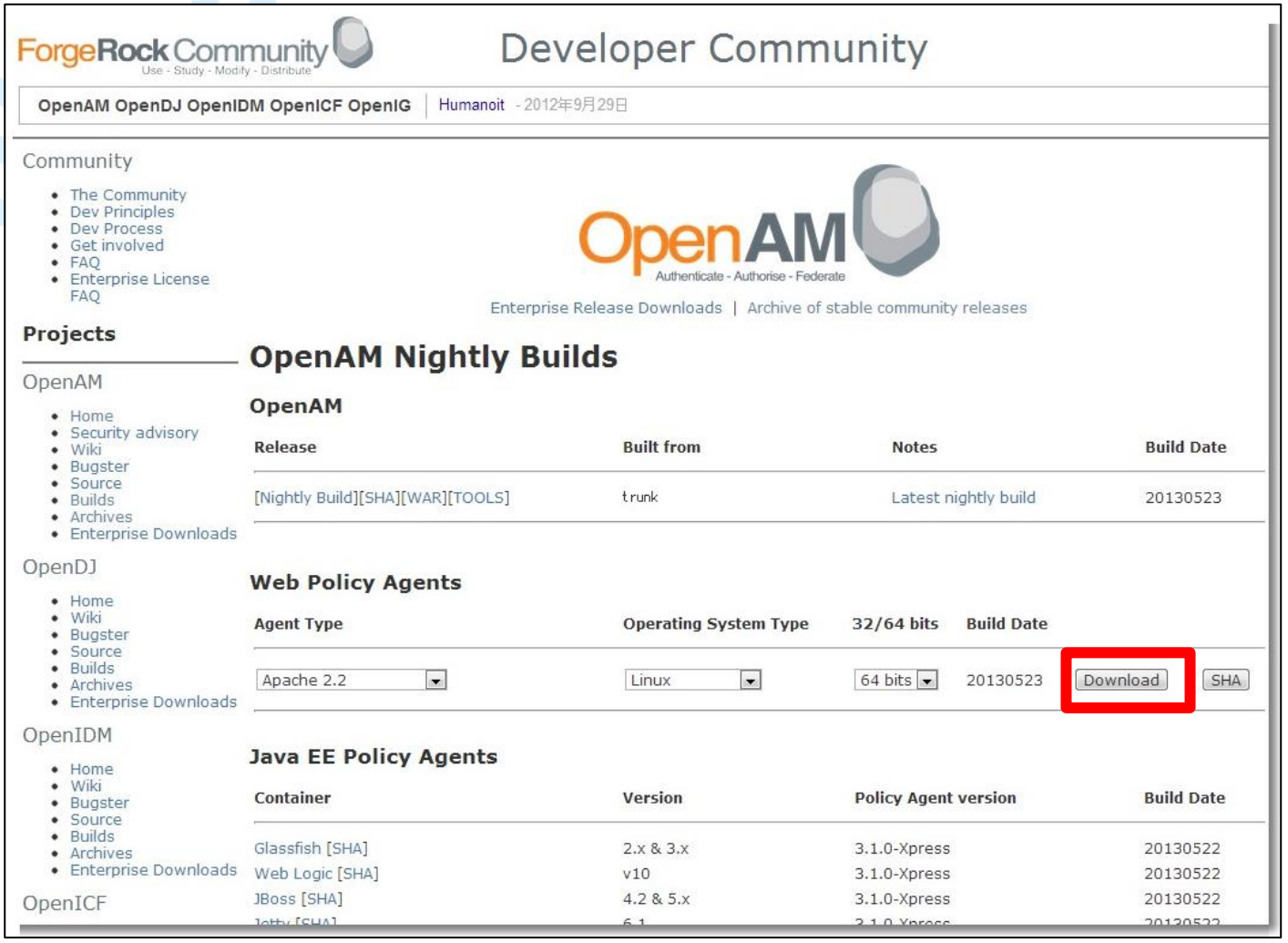
```
172.105.126.67   openam-traning-app.nriossc.co.jp
```

- 連携先サーバの/etc/hostsファイルにSSOサーバを追記する

```
192.175.204.101 openam.nriossc.co.jp
```

連携先システムの構築

ポリシーエージェントをダウンロード



ポリシーエージェントを連携先サーバにインストール

ファイルを解凍

```
# ls -l
apache_v22_Linux_64_agent_3.1.0-Xpress.zip

# unzip apache_v22_Linux_64_agent_3.1.0-Xpress.zip
# ls web_agents
apache22_agent
```

Apacheを止める

```
# /etc/init.d/httpd stop
```

パスワードファイルを作成する

```
# cd web_agents/apache22_agent/bin
# echo [デフォルトポリシーエージェントのパスワード] > password.txt
```

ポリシーエージェントを連携先サーバにインストール

ポリシーエージェントをインストール

```
# ./agentadmin --install
Please read the following License Agreement carefully:
 [Press <Enter> to continue...] or [Enter n To Finish]
（ライセンスが表示されるので、Enterかnで読み進める）
Agreement (yes/no): [no]: yes ←入力

Enter the Apache Server Config Directory Path
 [/opt/apache22/conf]: /opt/OpenStandia/apache-2.2/conf
OpenAM server URL: http://openam.nriossc.co.jp:8080/openam
Agent URL: http://openam-traning-app.nriossc.co.jp:80
Enter the Agent Profile name: openam-traning
Enter the path to the password file: [password.txtへのフルパス]

Please make your selection [1]: ←Enter
```

成功すると、apache22_agentディレクトリにAgent_001が作成される

エージェントプロファイルの作成

- http://openam.nriossc.co.jp:8080/openam からOpenAMにログイン

エージェントプロファイルの作成

- アクセス制御＞(最上位のレルム)＞エージェント と遷移
- エージェントの新規ボタンをクリック

エージェントプロファイルの作成

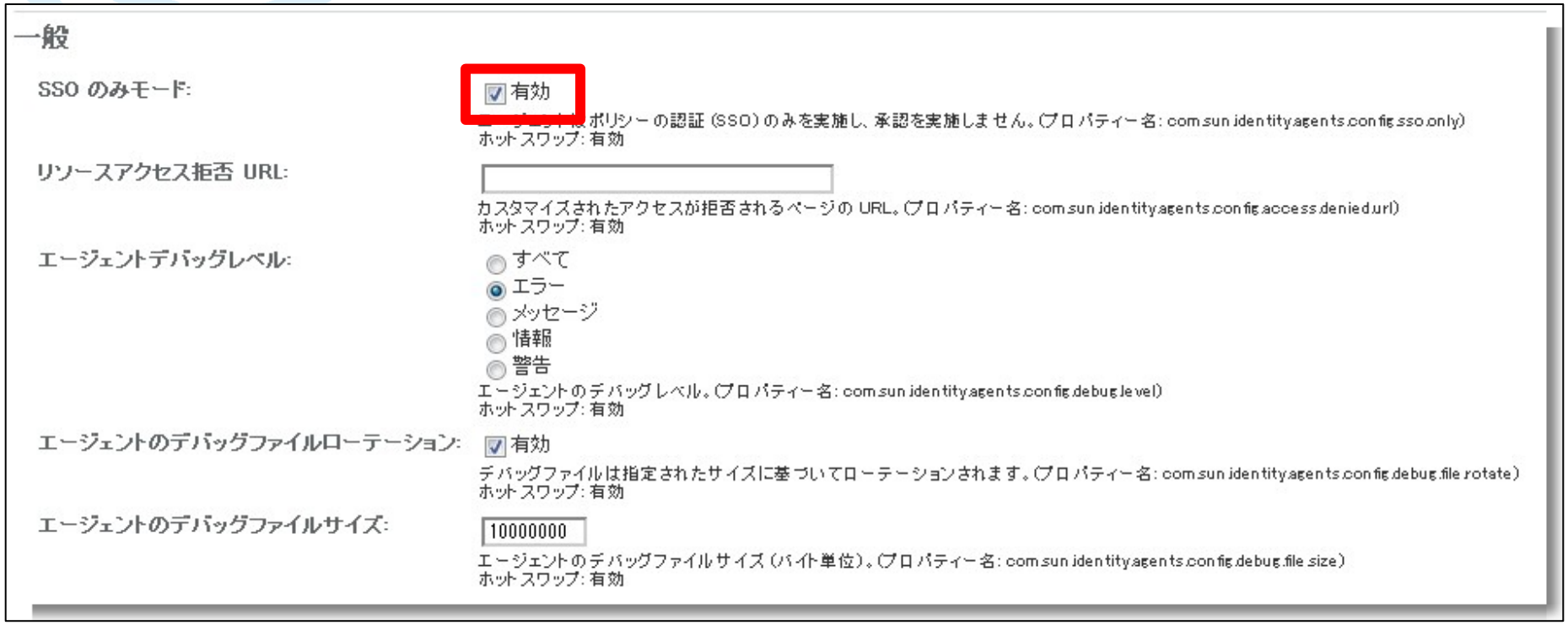
- エージェントの情報を入力して「作成」ボタンをクリック

エージェントプロファイルの作成

- 作成されたエージェント名をクリック

エージェントプロファイルの作成

- 「SSOのみモード」の「有効」にチェックを入れて「保存」ボタンをクリック
- その後ログアウト



連携確認

- 連携先システムにアクセス
  - ✓http://openam-traning-app.nriossc.co.jp/app01
- OpenAMのログインページにリダイレクトされることを確認
  - ✓http://openam.nriossc.co.jp:8080/openam/UI/Login?goto=http%3A%2F%2Fopenam-traning-app.nriossc.co.jp%2Fapp01

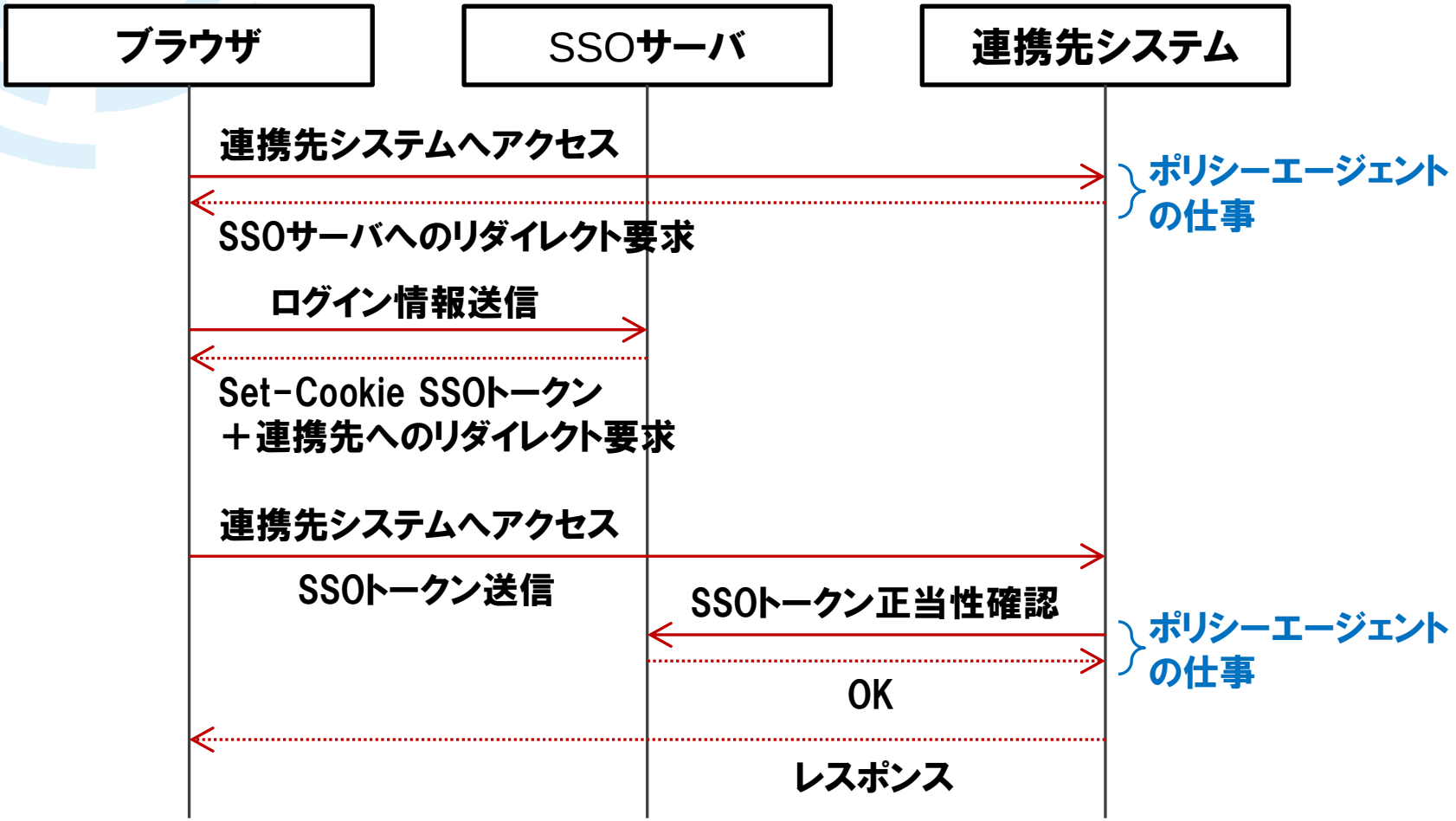
連携確認

- OpenAMのユーザでログイン（amadmin/adminpassword）すると、app01の画面が表示される
- このとき、HTTPヘッダに「iPlanetDirectoryPro」というSSOトークン（Cookie）が追加されていることを確認

```
管理者ユーザー向けサイト HTTPヘッダ SSO実行結果

想定通りのパラメータが送信されていることを確認してください。

---------- HTTPヘッダ ----------
ACCEPT = text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
ACCEPT_CHARSET = Shift_JIS,utf-8;q=0.7,*;q=0.3
ACCEPT_ENCODING = gzip,deflate,sdch
ACCEPT_LANGUAGE = ja,en-US;q=0.8,en;q=0.6
CACHE_CONTROL = max-age=0
CONNECTION = keep-alive
COOKIE = amlbcookie=01; iPlanetDirectoryPro=AQIC5wM2LY4SfcwtvDqwLv93UI9Q8wnkEsWboMzyjXA5tTM.*AAJTSQACMDE.*
HOST = openam-traning-app.nriossc.co.jp
REFERER = http://openam.nriossc.co.jp:8080/openam/UI/Login?goto=http%3A%2F%2Fopenam-traning-app.nriossc.co.jp%2Fapp01&gx_charset=UTF-8
USER_AGENT = Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.31 (KHTML, like Gecko) Chrome/26.0.1410.64 Safari/537.31

---------- リクエストパラメータ ----------
```

連携確認

- 構築したシステムは、以下のような動作をしています



ログインユーザのIDを連携

オープンソースまるごと
OpenStandia™
Open Source Technology

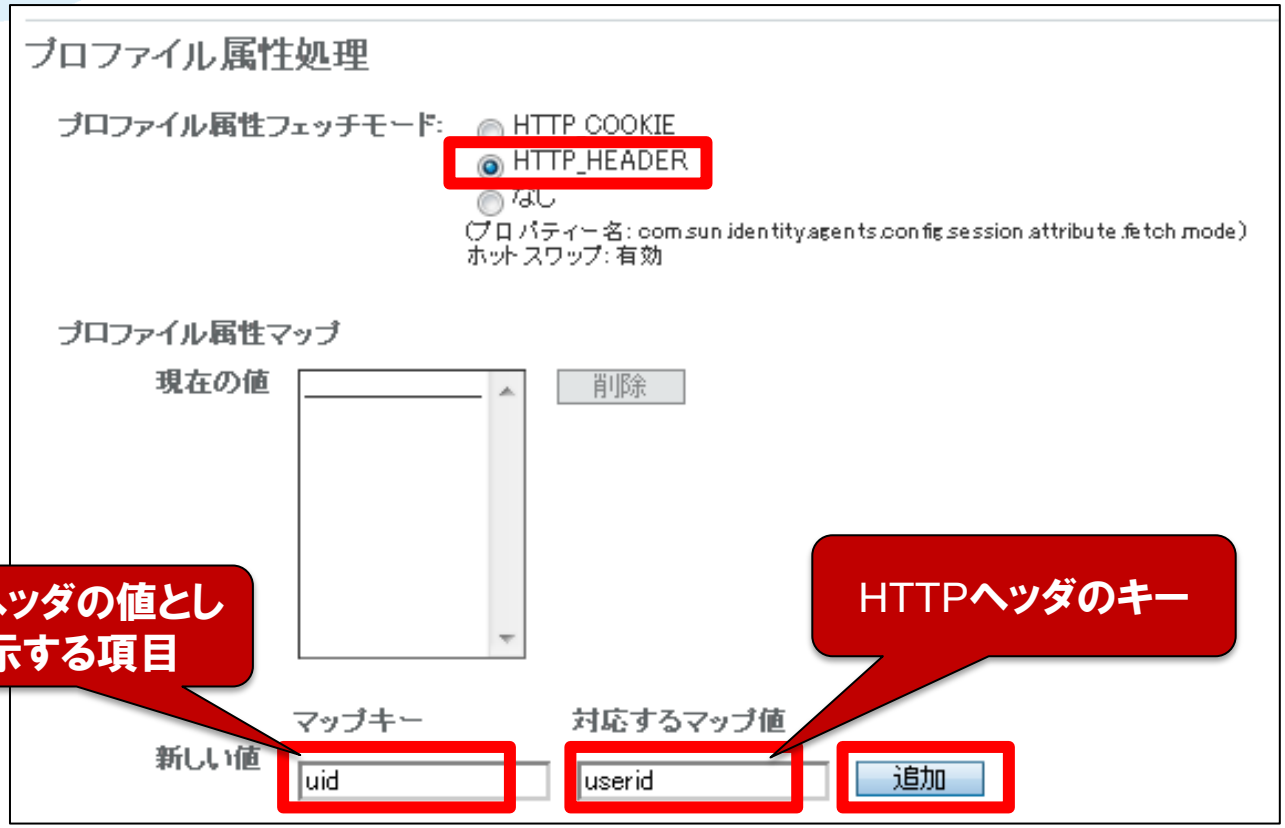
- HTTPヘッダにユーザIDを追加して連携する
  - アクセス制御＞(最上位のレルム)＞エージェント と遷移
  - エージェントの名前をクリックしてアプリケーションタブを開く

NRI 未来創発
Dream up the future.

ログインユーザのIDを連携

エージェントの設定を変更

- プロファイル属性処理を以下のように変更
- 「追加」をクリックした後、ページ上部の「保存」をクリック



エージェントを再起動

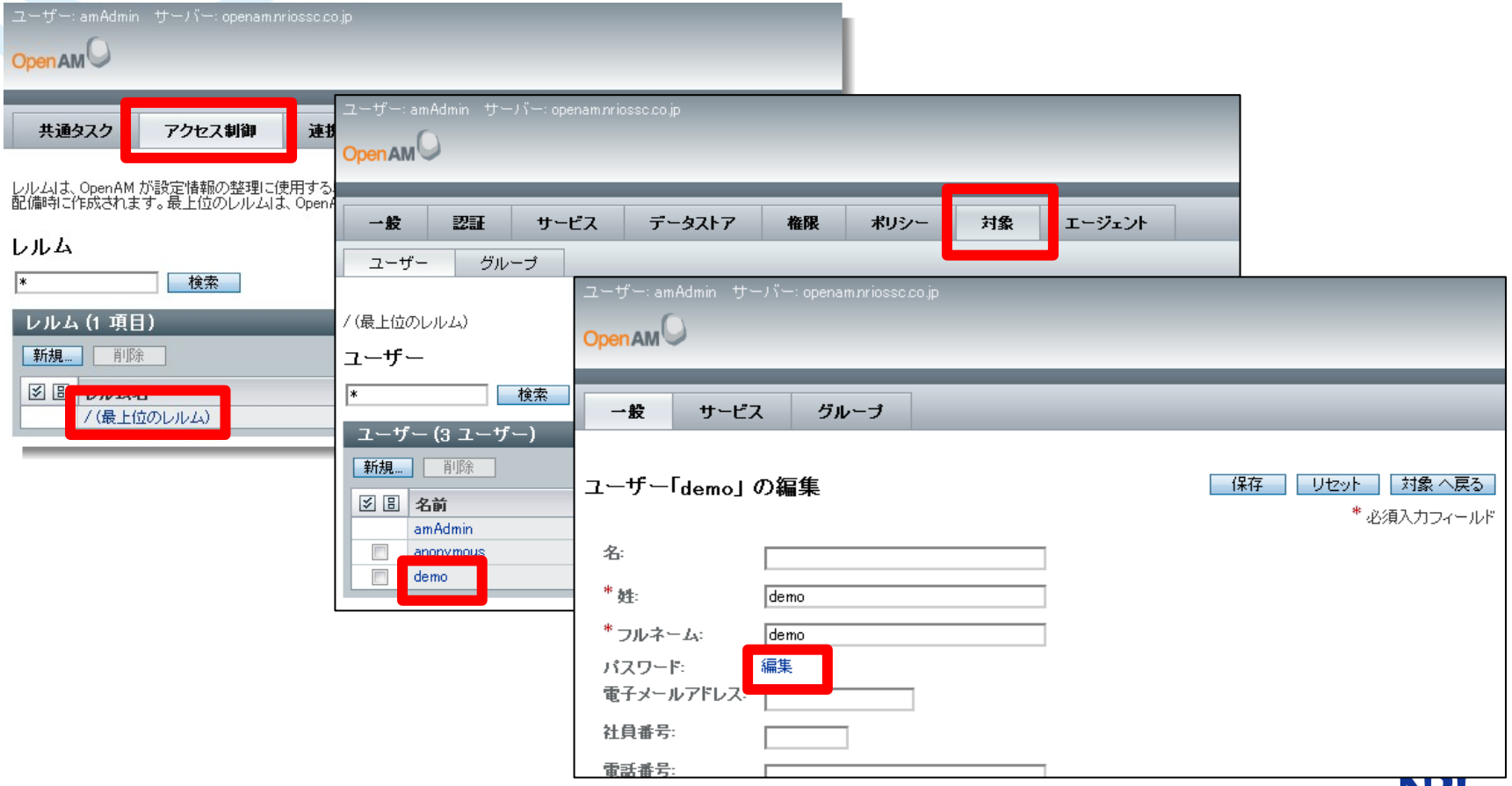
連携先システムのApacheを再起動

```
# /etc/init.d/v-httpd restart
httpd を停止中:                                            [  OK  ]
httpd を起動中:                                            [  OK  ]
```

ログインユーザのIDを連携

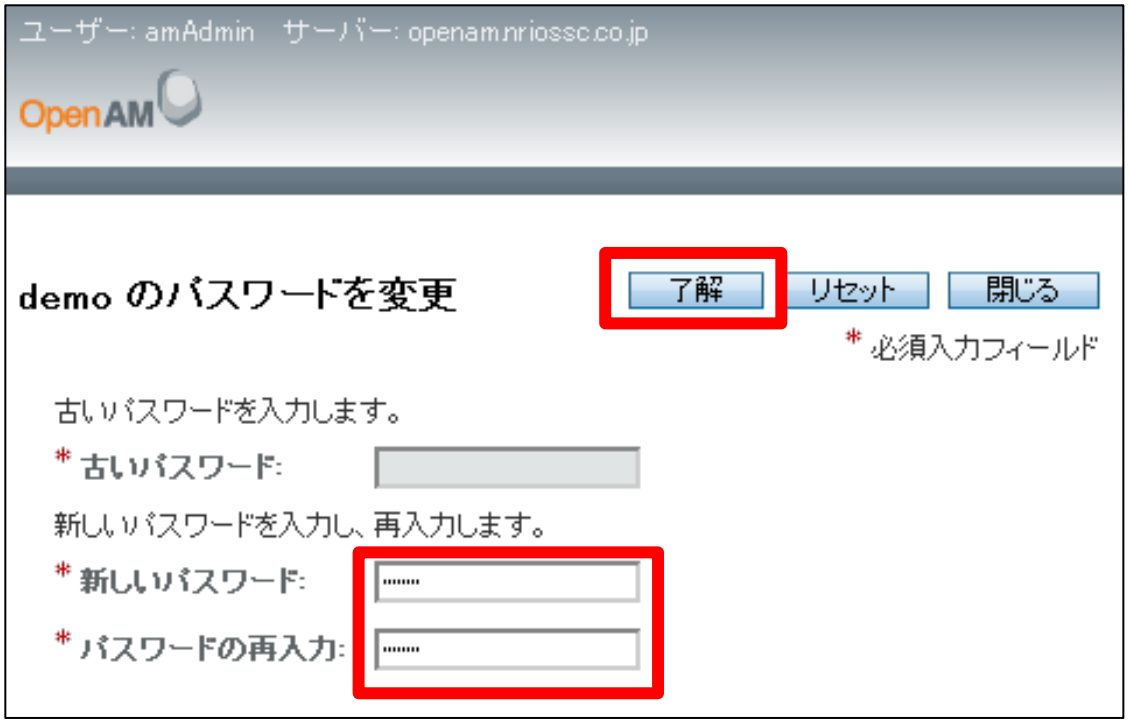
一般ユーザの確認

- アクセス制御＞（最上位のレルム）＞対象 と遷移
- 今回はdemoユーザを使うので、demoをクリックしてパスワードを編集



一般ユーザのパスワード設定

- 新しいパスワード（demodemo）を設定して「了解」をクリック
- その後ログアウト



ログインユーザのIDを連携

- 連携先システムにアクセス
  - http://openam-traning-app.nriossc.co.jp/app01
  - demoユーザ（demo / demodemo）でログイン

ログインユーザIDの連携を確認

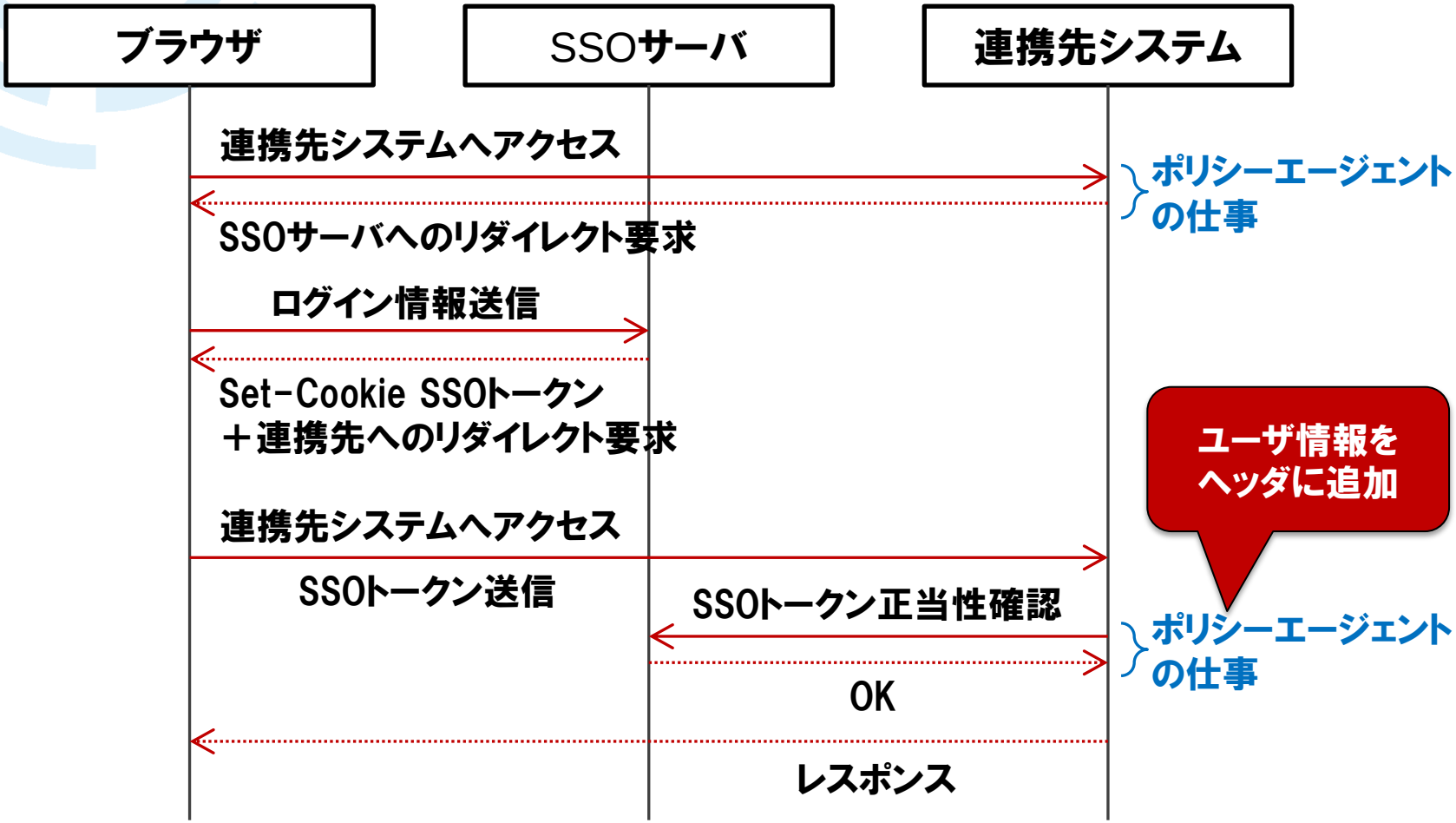
- HTTPヘッダに「USERID=demo」が追加されていることを確認

```
管理者ユーザー向けサイト HTTPヘッダ SSO実行結果

想定通りのパラメータが送信されていることを確認してください。

---------- HTTPヘッダ ----------
ACCEPT = text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
ACCEPT_CHARSET = Shift_JIS,utf-8;q=0.7,*;q=0.3
ACCEPT_ENCODING = gzip,deflate,sdch
ACCEPT_LANGUAGE = ja,en-US;q=0.8,en;q=0.6
CACHE_CONTROL = max-age=0
CONNECTION = keep-alive
COOKIE = amlbcookie=01; iPlanetDirectoryPro=AQIC5wM2LY4SfcxTaMZufuR6YGssjuid9PZdXud6ZFZxsAw.*AAJTSQACMDE.*
HOST = openam-traning-app.nriossc.co.jp
REFERER = http://openam.nriossc.co.jp:8080/openam/UI/Login?goto=http%3A%2F%2Fopenam-traning-app.nriossc.co.jp%2Fapp01
USERID = demo
USER_AGENT = Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.31 (KHTML, like Gecko) Chrome/26.0.1410.64 Safari/537.31

---------- リクエストパラメータ ----------
```

- これで、連携先システムがHTTPヘッダのUSERIDを参照してログインできる仕組みであれば、該当ユーザとしてログイン可能

連携確認

- 構築したシステムは、以下のような動作をしています



Section4

まとめ

まとめ

- Section1：OpenAM概要
  - SSOのメリットについて説明しました
  - OpenAMで実現可能なSSO方式をまとめました
- Section2：OpenAMインストール
  - OpenAMのインストールの流れを説明しました
- Section3：連携先システムとのSSO
  - エージェント方式のSSOの設定の流れを説明しました
  - ユーザIDを連携先システムに連携する方法を説明しました

- OpenStandiaは、「攻めのIT」を支援します。
- オープンソースのことなら、なんでもご相談ください！

オープンソースまるごと

OpenStandia™

Open Source Technology

お問い合わせは、NRIオープンソースソリューションセンターへ

ossc@nri.co.jp

http://openstandia.jp/

NRI 未来創発 Dream up the future.