学認春CAMP2014＠NII
2014/5/30(Fri)

金沢大学
KANAZAWA UNIVERSITY

Shibboleth用多要素認証導入のための技術ガイド

金沢大学　松平 拓也

発表の内容

金沢大学
KANAZAWA UNIVERSITY

- 金沢大学統合認証基盤について
- Shibbolethの認証方式について
- Shibbolethにおける多要素認証設定方法
- 開発した“認証方式選択プラグイン”の説明
- まとめ・今後の予定

はじめに

金沢大学
KANAZAWA UNIVERSITY

- 金沢大学では、これまで各部署・部局が独立して構築・運用していた情報システムの融合化の一環

- 金沢大学統合認証基盤を構築
  Kanazawa University Single Sign On（KU-SSO）
  - Shibbolethを採用
    - 認証サーバ：Identity Provider（IdP）を構築
    - 情報システム：Service Provider（SP）化
      ⇒シングルサインオン（認証）および属性共有（認可）を実現
  - 平成22年3月から本格運用を開始
  - 30以上の学内情報システムをShibboleth SP化

KU-SSOの現状

金沢大学 KANAZAWA UNIVERSITY

- KU-SSOのSP群
  - 学内IPアドレス ⇒ 全てアクセスを許可
  - 学外IPアドレス ⇒ アクセスを許可していないSPが多く存在（FWで遮断）
    - 例：給与明細オンライン、成績入力、予算執行支援システム など
      （学外用DNSに登録がないため、ポータルのリンクを辿ると「ページが見つかりません」となる・・）
  - 学外からアクセスするためにはVPNを利用する必要
    - VPNクライアントソフトがOSによって使えなかったりする
      （OSのバージョンアップで使えなくなったりする）
    - そもそも一般ユーザにとってはVPNそのものが敷居が高い ←重要

      ⇒日常的に金沢大学に通うことが物理的に困難な社会人学生、出張先からのアクセス需要が高い教員に対して不便を強いている

VPNを利用しなくても、すべてのSPを金沢大学外からでも利用可能な環境を実現したい

VPNなしに学外からのアクセスを実現するためには？

金沢大学 KANAZAWA UNIVERSITY

- これらのSPがVPNを必要とする理由
  - 本学の認証方式がUsername/Password認証だけで運用されていることが大きい
  - システム（SP）管理者はなかなか外部公開に踏み切れない
  - Username/Password認証の代わりに多要素認証に切り替え、セキュリティを向上

全てのSPがVPNを利用することなしに利用できる環境に移行できる

多要素認証

金沢大学 KANAZAWA UNIVERSITY

- 多要素認証とは？
  - 認証
    - 利用者が本人であるかどうかを確認する作業
  - 認証方式
    - 本人しか知らない知識によって認証（What You Know）
      - Password、秘密の質問など

      ※知識だけでは推測・漏洩などのリスクが高い
    - 本人しか持っていない所有物で認証（What You Have）
      - ICカード、スマートフォンなど
    - 本人の生体的特徴よって認証（What You Are）
      - 指紋、静脈、虹彩など（バイオメトリクス）

      ※替えが利かないリスクもある
  - 多要素認証
    - 上記3種類の認証方式のうち、2種類以上を必要とする認証
      - 例：ICカード（所有物）とPIN（知識）

Shibbolethにおける認証方式

金沢大学 KANAZAWA UNIVERSITY

- Shibbolethにおける認証の流れ

  1. ユーザがSPにアクセス
  2. SPはIdPに対して認証方式を指定
     （Authentication Context ClassとしてSAMLアサーション内に記述）
  3. IdPは自身に定義されている認証方式（Login Handler）から、SPから指定された認証方式を選択
  4. ユーザはIdPから提示された認証を行う
  5. 認証成功後、IdPからSPに対してユーザがどの認証方式で行ったかをSPに通知
     （Authentication Context ClassとしてSAML アサーション内に記述）

Shibboleth標準の認証方式（1）（Username/Password認証の例）

金沢大学 KANAZAWA UNIVERSITY

- Username/password login handlerを利用
  - SPの設定
    - 認証方式の指定（Authentication Context Classの指定）
      パスワード認証は“urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport”

shibboleth2.xml

```
<SessionInitiator type="Chaining" Location="/Login" isDefault="true" id="Intranet“
        authnContextClassRef="urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport"
    relayState="cookie" entityID="https://IdPserver/idp/shibboleth">
    <SessionInitiator type="SAML2" acsIndex="1" template="bindingTemplate.html"/>
    <SessionInitiator type="Shib1" acsIndex="5"/>
</SessionInitiator>
```

または、shib.conf

```
<Location /secure>
  AuthType shibboleth
  ShibCompatWith24 On
  ShibRequestSetting requireSession 1
  ShibRequestSetting authnContextClassRef urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
  require shib-session
</Location>
```

Shibboleth標準の認証方式（1）（Username/Password認証の例）

金沢大学 KANAZAWA UNIVERSITY

- IdPの設定
  - SPから送信されてきたAuthentication Context Classを基に、認証方式をユーザに提示
  - この場合は、“urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport”

handler.xml

```
<ph:LoginHandler xsi:type="ph:UsernamePassword"
        jaasConfigurationLocation="file:///opt/shibboleth-idp/conf/login.config">
  <ph:AuthenticationMethod>
        urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
  </ph:AuthenticationMethod>
</ph:LoginHandler>
```

- AuthenticationEngineがAuthenticationMethodで指定されたLoginHandlerを選択
- 選択されたLoginHandlerが認証処理を実行
- SPからの指定がない場合はパスワード認証がデフォルトで行われる

Shibboleth標準の認証方式（2）（IP address認証の例）

金沢大学 KANAZAWA UNIVERSITY

あくまでLogin Handlerを解説するための参考用！

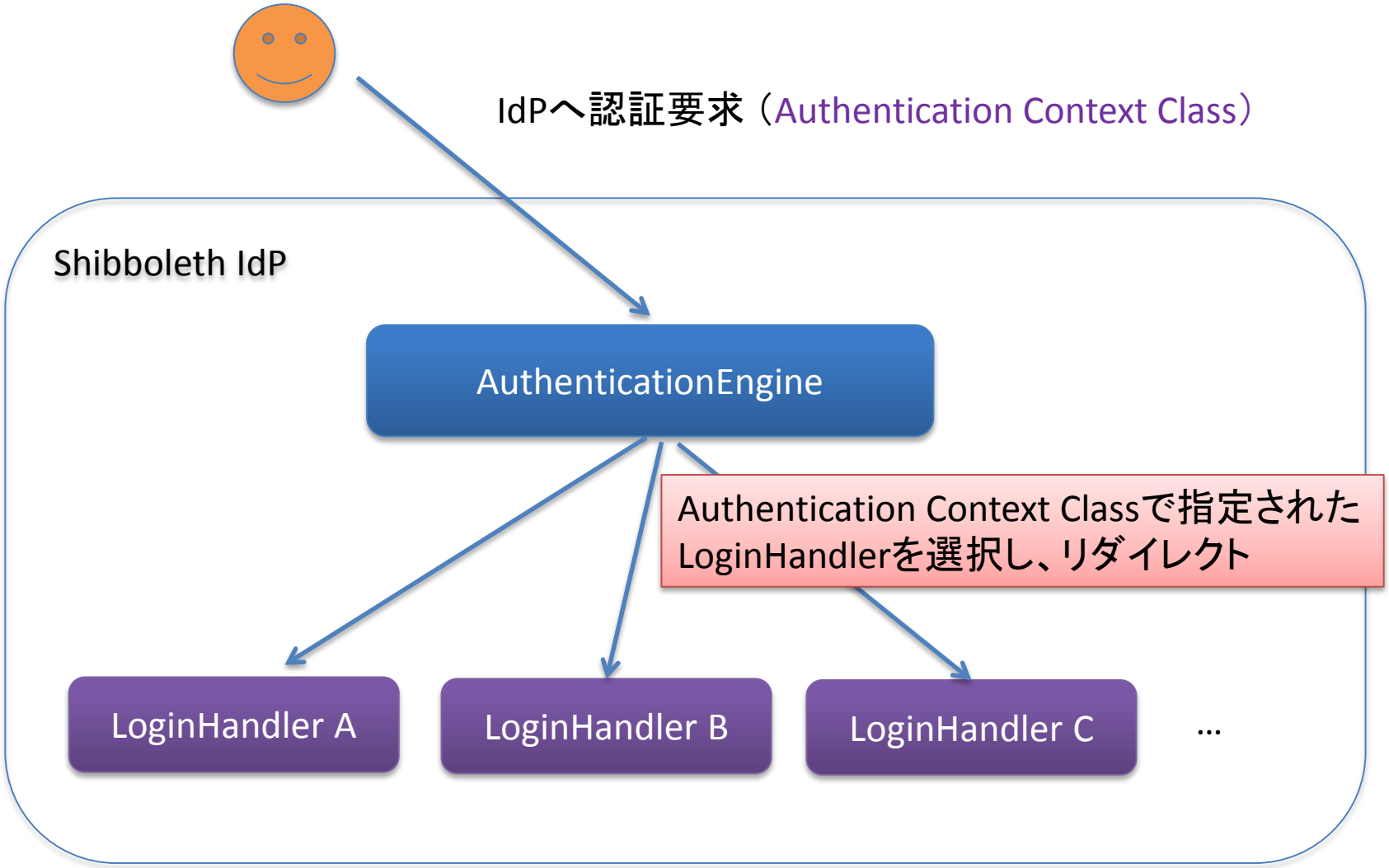
- IP address login handlerを使用
  - SPの設定
    - “urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol”をセット
  - IdPの設定
    - handler.xmlに以下を記述

```
<ph:LoginHandler xsi:type="ph:IPAddress" username=“test" defaultDeny="true">
    <ph:AuthenticationMethod>
        urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol
    </ph:AuthenticationMethod>
    <ph:IPEntry>192.168.0.0/24</ph:IPEntry>
</ph:LoginHandler>
```

IPアドレスが192.168.0.0/24のユーザは”test”としてログインされる
（その他のIPからはアクセスできない）

- これらの他にも標準でいくつかサポートされている
  - https://wiki.shibboleth.net/confluence/display/SHIB2/IdPUserAuthn

IdPの標準的な流れ


IdPへ認証要求（Authentication Context Class）
Shibboleth IdP
AuthenticationEngine
Authentication Context Classで指定された
LoginHandlerを選択し、リダイレクト
LoginHandler A
LoginHandler B
LoginHandler C
…

Shibbolethでの多要素認証

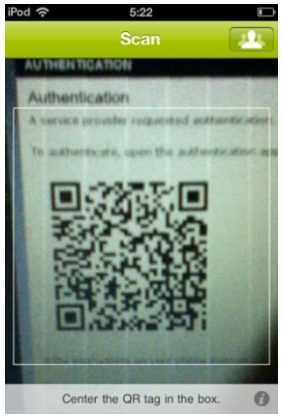
金沢大学 KANAZAWA UNIVERSITY

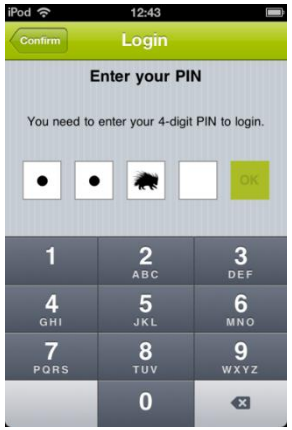
- Shibbolethの標準的な認証方式の中には多要素認証はない
  - Shibbolethが提供している認証方式以外を利用する必要がある
- Shibbolethにおける認証拡張方式
  1. Remote User Login Handler
     REMOTE_USER環境変数にIDを入れることが可能な認証実装の場合に利用可能
     例）Apache Basic認証、クライアント証明書認証
     - https://wiki.shibboleth.net/confluence/display/SHIB2/IdPAuthRemoteUser
     - https://meatwiki.nii.ac.jp/confluence/pages/viewpage.action?pageId=12158431
  2. External Authentication Login Handler
     JAVAのコードでIDを受け渡せる認証実装の場合に利用可能
     ⇒ 今回はこの方式でtiQr認証を適用させた例を説明
  3. Login Handlerを独自に実装
     任意の認証実装にも利用可能であるが複雑

認証実装の仕組みに合わせて3つから選択可能

tiQr認証

- tiQr
  - より優れた利便性を追求しSURFnetで開発された方式
    - スマートフォンのアプリとして実装（iPhone（日本語対応済）、Android）
    - スマートフォン（所有物）とPIN（知識）の多要素認証
    - QRコードを用いてユーザのログイン操作を軽減
    - https://tiqr.org/





External Authentication Login Handler

金沢大学 KANAZAWA UNIVERSITY

- External Authentication Login Handlerの一般的な流れ
  - IdPおよびSPの設定
    - 認証方式の指定（Authentication Context Classの指定）
      例） urn:mace:gakunin.jp:idprivacy:ac:classes:Sample
  - SPの設定
    - 前述のようにAuthentication Context ClassをIdPに送信
  - IdPの設定
    1. handler.xmlにExternal Authentication Login Handlerの追加
    2. Handlerで指定したPathを呼び出す（Filter or Servlet or JSP）
    3. 2のプログラムから外部認証を呼び出す
    4. 成功後、AuthenticationEngineにIDを返す

tiQrのShibboleth対応方法の詳細
https://meatwiki.nii.ac.jp/confluence/display/tiqr/Home

tiQrのShibboleth対応（1）

金沢大学 KANAZAWA UNIVERSITY

- SPの設定
  - 認証方式の指定（Authentication Context Classの指定）
    “ urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract ”をセット

shibboleth2.xml

```
<SessionInitiator type="Chaining" Location="/DS" isDefault="true" id="tiqrshiblogin"
    authnContextClassRef="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract">
        <SessionInitiator type="SAML2" template="bindingTemplate.html"/>
        <SessionInitiator type="Shib1"/>
        <SessionInitiator type="SAMLDS" URL="https://DS SERVER/ds/WAYF"/>
</SessionInitiator>
```

tiQrのShibboleth対応（2）

金沢大学 KANAZAWA UNIVERSITY

- IdPの設定
  1. handler.xmlにExternal Authentication Login Handlerの追加

handler.xml

```
<ph:LoginHandler xsi:type="ph:ExternalAuthn"
                 externalAuthnPath="/Authn/TiqrShib" >

   <ph:AuthenticationMethod>
      urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract
   </ph:AuthenticationMethod>
</ph:LoginHandler>
```

  2. Handlerで指定したPathを呼び出す（Filter）

https://IDP_SERVER/idp/Authn/TiqrShib が呼び出される

tiQrのShibboleth対応（3）

金沢大学 KANAZAWA UNIVERSITY

- IdPの設定
  3. 2のプログラムから外部認証を呼び出す

/TOMCAT_HOME/webapps/idp/WEB-INF/web.xml

```
<servlet>
      <servlet-name>TiqrShibAuthHandler</servlet-name>
      <servlet-class>tiqrshibAuthn</servlet-class>
</servlet>

  <servlet-mapping>
     <servlet-name>TiqrShibAuthHandler</servlet-name>
     <url-pattern>/Authn/TiqrShib</url-pattern>
  </servlet-mapping>
```

※ /TOMCAT_HOME/webapps/idp/WEB-INF/classes/ にtiqrshibAuthn.classを配置

tiQrのShibboleth対応（4）

金沢大学 KANAZAWA UNIVERSITY

tiqrshibAuthn.java

```
import javax.servlet.*;
import javax.servlet.http.*;
import edu.internet2.middleware.shibboleth.idp.authn.AuthenticationEngine;
import edu.internet2.middleware.shibboleth.idp.authn.LoginHandler;

public class tiqrshibAuthn extends HttpServlet {
  protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException,
        IOException {

  /*tiQr認証済みならばtiQrのCookieがある*/

  if(tiQr認証済みCookieがない) {
     tiQr認証にリダイレクト
  }

  if(tiQr認証済みCookieがある)
     request.setAttribute(LoginHandler.PRINCIPAL_NAME_KEY, “tiQr認証時に使ったID”);
     AuthenticationEngine.returnToAuthenticationEngine(request, response);
  }
}
```

4．成功後、AuthenticationEngineに処理を返す

※PRINCIPAL_NAME_KEY（値：principal_name）を属性名、認証で使用したIDを値としてsetAttributeメソッドにセットし、　returnToAuthenticationEngineメソッドを呼び出す

外部認証のShibboleth対応は結構簡単！

AuthenticationEngineの現状

金沢大学 KANAZAWA UNIVERSITY

- Shibbolethでの多要素認証は可能
  - 多要素認証
    - Username/Password認証より手間がかかる
    - 特定の所有物（スマホ、ICカードなど）がないと認証できない
  - 多要素認証をUsername/Password認証と全て置き換えるのはあまり現実的ではない
    - 常に多要素を要求するのではなく、特定の条件下に限って要求するのがベスト
- 現在のShibboleth（AuthenticationEngine）の仕様
  - 現在のShibbolethの仕様では、IdPは1つのSPに対して1つの認証方式でしか対応できない

認証方式選択プラグインの開発

金沢大学 KANAZAWA UNIVERSITY

- AuthenticationEngineの改善点
  1. ユーザのIPアドレスに応じて要求する認証方式を変えたい
     - 例　学内　Username/Password認証　　学外　tiQr認証

       ⇒ リスクベース認証を行いたい

  2. 複数の認証方式を選択できるようにしたい（and/or 条件）
     - tiqrやICカード認証はスマホやICカードを持つ人しか認証できない
     - 大学で多要素認証を普及させるには、如何にして使えないユーザを作らないかが重要
       - 例：　tiqr認証 or ICカード認証
     - 非常に重要な情報を扱うSPは強固な認証を要求したい
       - 例：　tiqr認証 and ICカード認証

IdPが複数の認証方式からSPの要求に応じた認証方式をユーザに提示できる“認証方式選択プラグイン”を開発

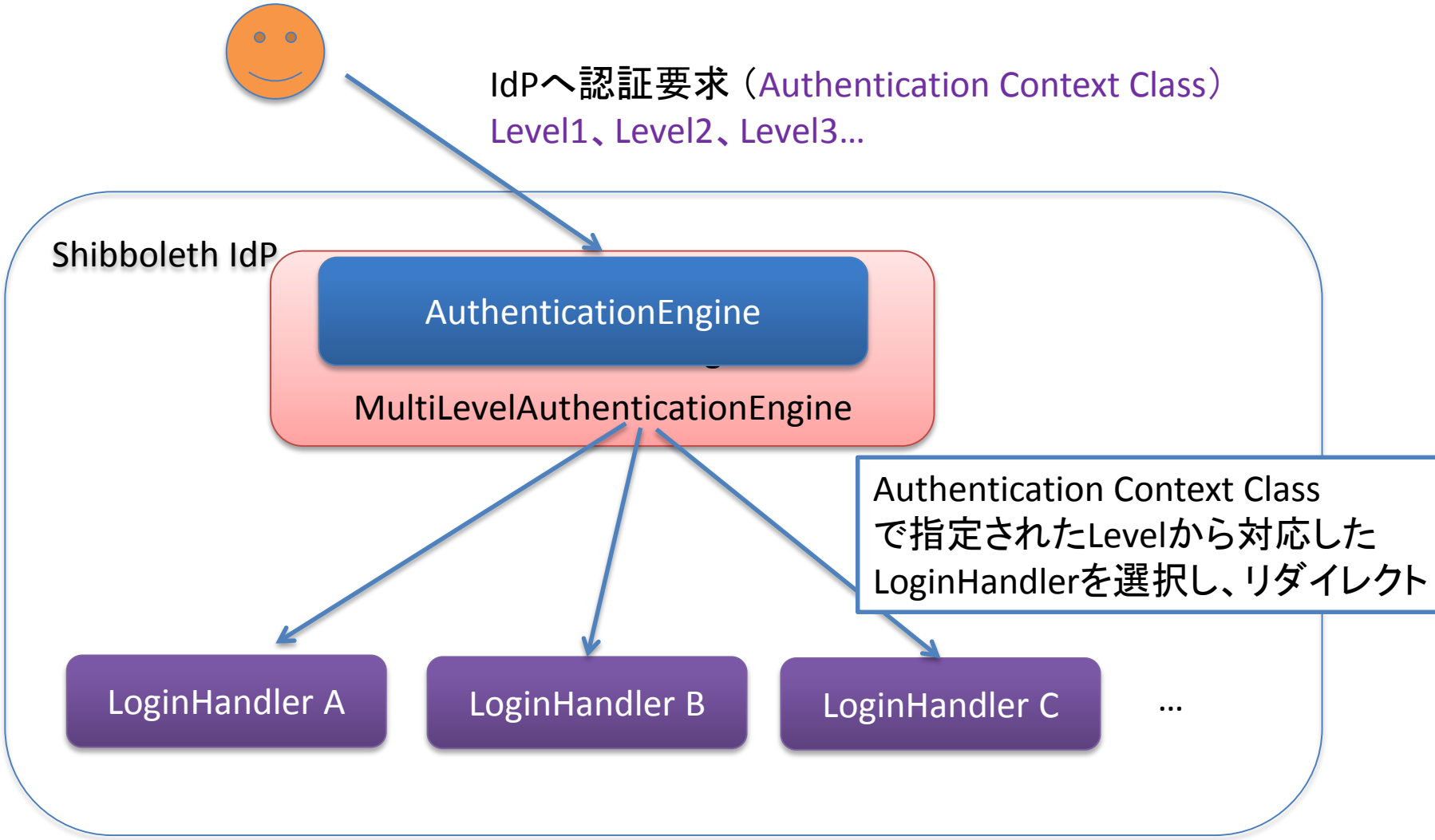
プラグインの設定方法

金沢大学 KANAZAWA UNIVERSITY

- 必要な設定
  - SPの設定
    - Authentication Context Classの指定において、認証方式の代わりにレベル情報（Level1/Level2/…/levelN）をセット
  - IdPの設定
    1. IPアドレスのホワイトリスト設定
       - ホワイトリストのIPアドレスを “inside”、それ以外を“outside”とする
         例：学内ネットワーク(X.X.X.X/16)をホワイトリストとして登録
    2. 各レベルの認証方法を決める
       - Level1 Username/Password 認証(both inside and outside)
       - Level2 Username/Password 認証or tiQr 認証 (inside)
         tiQr 認証 (outside)
       - Level3 Username/Password 認証 and tiQr 認証(both inside and outside)

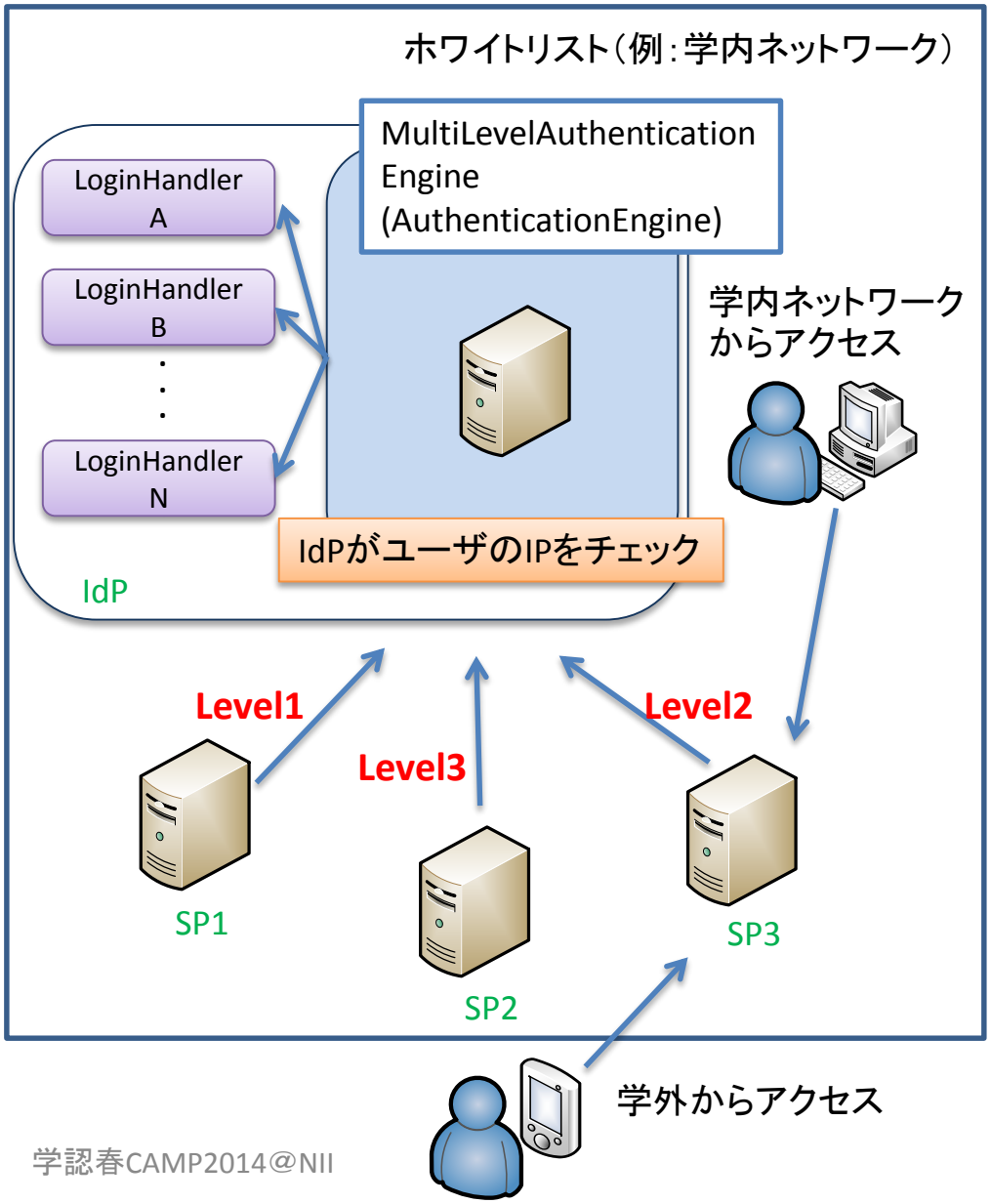
IdPの流れ（プラグイン導入後）

金沢大学 KANAZAWA UNIVERSITY



プラグインの特徴

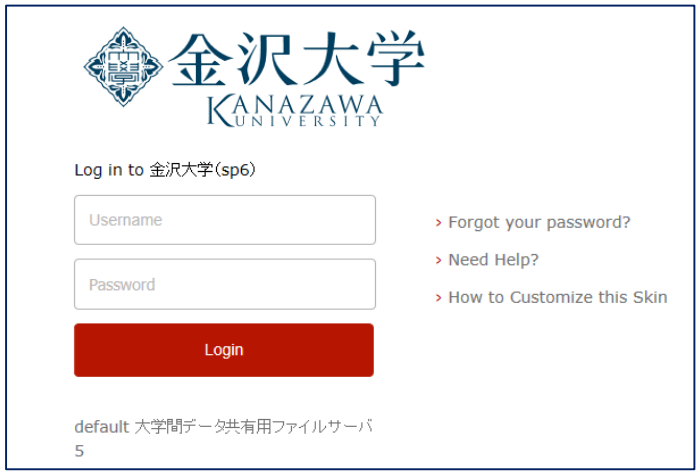
- 従来の認証レベルで十分なSPはパスワード認証（Level1）で対応
- 高いレベルを要求するSPにおいては、ホワイトリストのIPアドレスに応じて多要素認証（Level2以上）を提示
- 一度上位レベルで認証に成功した場合、別SPにアクセスした際に下位レベルの認証を要求されてもシングルサインオン（SSO）できる



SPがLevel1を要求した場合

- Level1 Username/Password 認証(both inside and outside)

SPに
アクセス

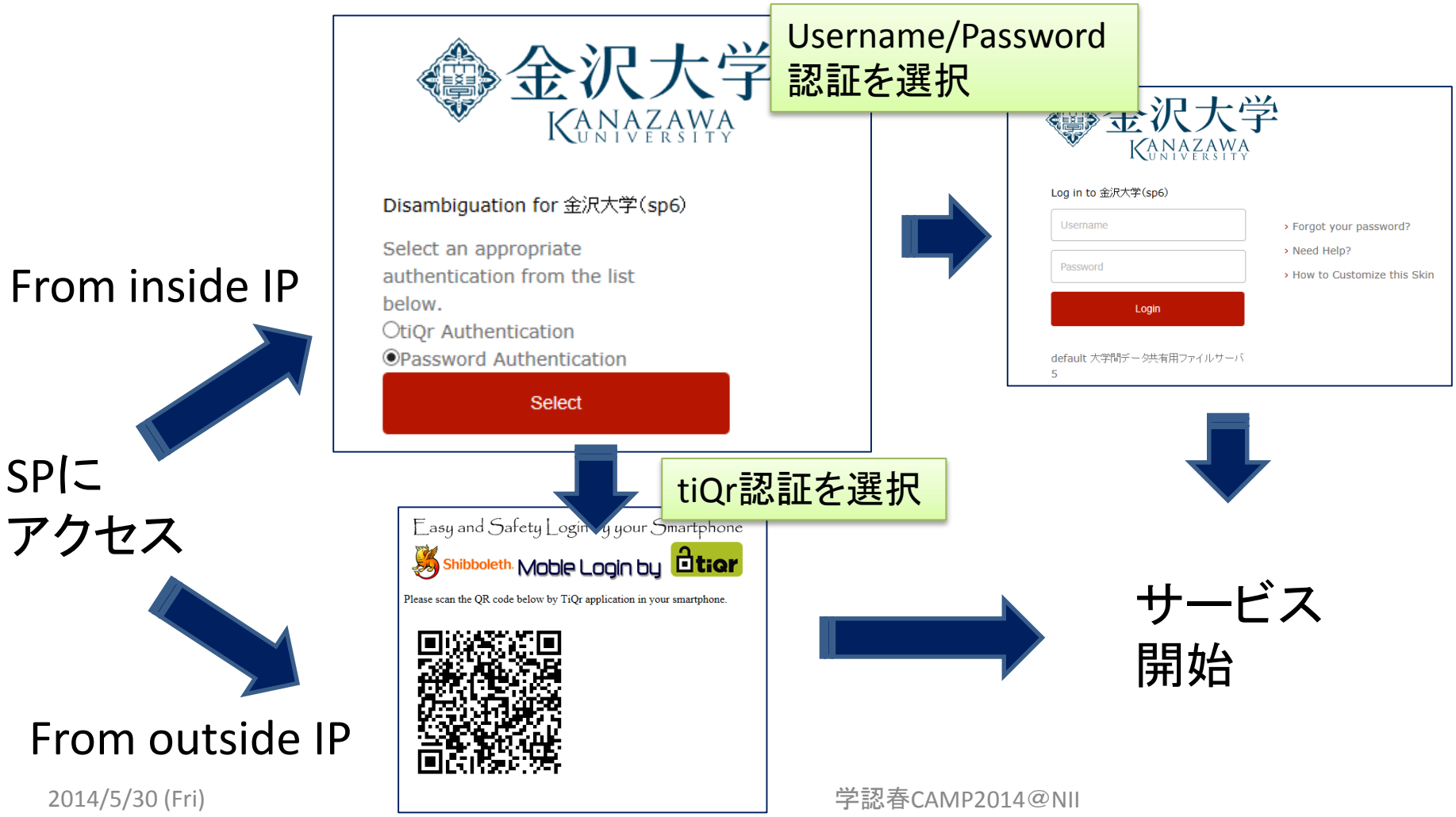


サービス
開始

これまでと同様の認証

SPがLevel2を要求した場合

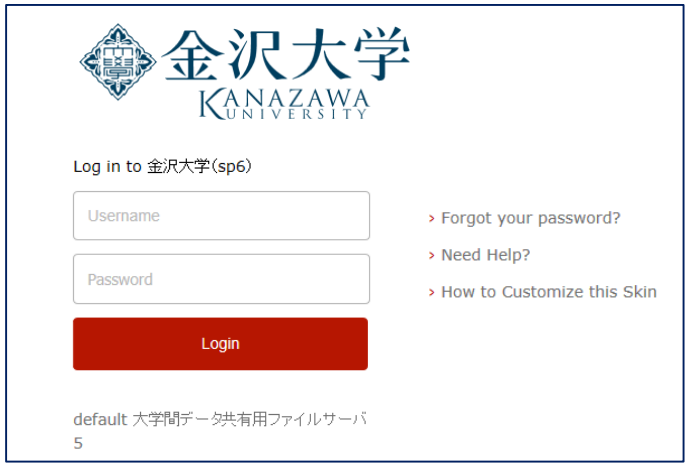
- Level2 Username/Password 認証or tiQr 認証 (inside)
  tiQr 認証 (outside)



SPがLevel3を要求した場合

- Level3 Username/Password 認証 and tiQr 認証(both inside and outside)

SPに
アクセス



サービス
開始

まとめ

金沢大学 KANAZAWA UNIVERSITY

- Shibbolethでは外部認証を使ってログインすることが可能

  ⇒ Shibbolethで容易に多要素認証を組み込むことが可能

- 認証方式選択プラグインを開発
  - 認証方式をレベルとして抽象化

  ⇒ 取り扱う情報の重要度に応じてSP群をレベル分けしておけば、あるレベルの認証方式に追加・変更があっても、そのレベルのSPだけに直ちに適用できる

今後の予定

金沢大学 KANAZAWA UNIVERSITY

- 学内統合認証基盤において本プラグインを試行導入するため準備中
  - Level1 Username/Password認証
  - Level2（inside）Username/Password認証　　（outside）tiQr認証
    - VPNを使うことなく、学外からアクセスできるようにする

- 試行導入が上手く行き次第、本プラグインを提供予定

- 最終的に、学認で利用するためには
  1. レベルごとのAuthentication Context Class名の決定
  2. 各レベルに期待する認証強度の決定
  3. 各大学のIdPへの本プラグイン導入および各レベルに対応した認証方式の組み込み
  4. 各SPのレベル決定