Integración CAS-SIR-STORK.

El código depositado en la forja de RedIris (https://forja.rediris.es/projects/cas-sir-stork/) es resultado de un proyecto del Área TIC de la Universidad de Santiago de Compostela (USC–http://www.usc.es/atic). Se puede consultar información sobre una visión general de la identificación en la USC en la siguiente dirección: http://www.rediris.es/jt/jt2010/ponencias/jt2010-jt-serv_feder_1-2.pdf .

Secuencia de pasos que ocurren en una validación SIR (Servicio de Identidad de RedIris).

1. Pulsamos en el enlace de validación Sir.
2. Se ejecuta el método handleRequestInternal del controlador InitSirValidationController. Guarda en la sesión del usuario el valor del parámetro service del objeto HttpServletRequest con el nombe de atributo sirCasServiceValue. Redirecciona la petición a la url del SIR.
3. Seleccionamos comunidad autónoma.
4. Seleccionamos proveedor de identidad.
5. Nos autenticamos.....
6. Se ejecuta el método doFilter del filtro LoginSirFilter que verifica los parámetros recibidos en el Request de la petición recibida en /loginSir. Se recupera el parámetro service de la sesión del usuario y se añade como parámetro al Request de la petición, para fijar la url a la que nos redirijirá el CAS si la autenticación es correcta.
7. Se ejecuta el método constructCredentialsFromRequest de la clase SirNonInteractiveCredentialsAction. Se recuperan los datos devueltos por el SIR si es una respuesta SIR y se devuelve un objeto SirCredentials.

Secuencia de pasos que ocurren en una validación STORK (Proyecto Stork).

1. Pulsamos en el enlace de validación Stork.
2. Se ejecuta el método handleRequestInternal del controlador InitStorkValidationController. Guarda en la sesión del usuario el valor del parámetro service del objeto HttpServletRequest con el nombre de atributo sirCasServiceValue. Redirecciona la petición a la url del proyecto Stork.
3. Seleccionamos país.
4. Seleccionamos proveedor de identidad.
5. Nos autenticamos.....
6. Se ejecuta el método doFilter del filtro LoginSirFilter que verifica los parámetros recibidos en el Request de la petición recibida en /loginStork. Se recupera el parámetro service de la sesión del usuario y se añade como parámetro al Request de la petición, para fijar la url a la que nos redirijirá el CAS si la autenticación es correcta.
7. Se ejecuta el método constructCredentialsFromRequest de la clase StorkNonInteractiveCredentialsAction. Se recuperan los datos devueltos por la pasarela del proyecto Stork de RedIris si es unha resposta STORK y se devuelve un objeto StorkCredentials.

Requisitos del artefacto usc-cas-sir-stork

- Bouncy Castle Crypto APIs: Versión 1.45 del artefacto bcprov-jdk16.
- Versión 3.3.5 del artefacto cas-server-core.

```
…..............................
        <dependency>
            <groupId>org.jasig.cas</groupId>
            <artifactId>cas-server-core</artifactId>
```

```
            <version>3.3.5</version>
        </dependency>
        <dependency>
          <groupId>org.bouncycastle</groupId>
          <artifactId>bcprov-jdk16</artifactId>
          <version>1.45</version>
        </dependency>
…..............................
```

Modificaciones necesarias en la configuración del CAS.

Fichero pom.xml.

Añadimos la siguiente dependencia en el pom.xml del CAS:

```
<?xml version="1.0" encoding="UTF-8"?>
<project ..........................">
..............................
    <properties>
        ..............................
        <usc-cas-sir-stork-version>0.1</usc-cas-sir-stork-version>
    </properties>
......................
    <dependencies>
        ..............................
        <dependency>
            <groupId>es.usc.cas</groupId>
            <artifactId>usc-cas-sir-stork</artifactId>
            <version>${usc-cas-sir-stork-version}</version>
        </dependency>
    </dependencies>
..............................
</project>
```

Fichero server.xml de la instancia del tomcat.

Definimos en el fichero de configuración server.xml de la instancia del tomcat, las entradas JDNI necesarias para que el CAS se conecte a SIR y STORK:

- cas-server-webapp-local/sir/gpoa/uri - Uri del GPOA de SIR.
- cas-server-webapp-local/sir/gpoa/pubkey_filename - Path al fichero que contiene la clave pública del GPOA de SIR.
- cas-server-webapp-local/sir/uriCasRecepcionSir - Uri del cas que recibirá las respuestas de SIR.
- cas-server-webapp-local/stork/gpoa/uri - Uri del GPOA de STORK.
- cas-server-webapp-local/stork/gpoa/pubkey_filename - Path al fichero que contiene la clave pública del GPOA de STORK.
- cas-server-webapp-local/stork/uriCasRecepcionStork - Uri del cas que recibirá las respuestas de STORK.

```
  <GlobalNamingResources>
  ..............................
    <!-- CAS
############################################################################################################################################ -->
      ..............................
      <Environment name="cas-server-webapp-local/sir/gpoa/uri" type="java.lang.String"
value="http://www.rediris.es/SIRGPoA/papiPoA" />
      <Environment name="cas-server-webapp-local/sir/gpoa/pubkey_filename" type="java.lang.String"
value="Sir_GPoA_pubkey.pem" />
      <Environment name="cas-server-webapp-local/sir/uriCasRecepcionSir" type="java.lang.String"
value="https://www.usc.es/cas/loginSir" />
      <Environment name="cas-server-webapp-local/stork/gpoa/uri" type="java.lang.String"
value="http://www.rediris.es/sir/stork/" />
      <Environment name="cas-server-webapp-local/stork/gpoa/pubkey_filename" type="java.lang.String"
value="Stork_GPoA_pubkey.pem" />
      <Environment name="cas-server-webapp-local/stork/uriCasRecepcionStork" type="java.lang.String"
value="https://www.usc.es/cas/loginStork" />
    <!-- Fin CAS
############################################################################################################################################ -->
    ..............................
  </GlobalNamingResources>
```

Fichero META-INF/context.xml.

Definimos en el fichero META-INF/context.xml los mapeos con las variables JNDI que guardan la configuración de SIR y STORK:

```
.....................
  <ResourceLink global="cas-server-webapp-local/sir/gpoa/uri" name="cas-server-webapp-local/sir/gpoa/uri"
type="java.lang.String"/>
  <ResourceLink global="cas-server-webapp-local/sir/gpoa/pubkey_filename" name="cas-server-webapp-
local/sir/gpoa/pubkey_filename" type="java.lang.String"/>
  <ResourceLink global="cas-server-webapp-local/sir/uriCasRecepcionSir" name="cas-server-webapp-local/sir/uriCasRecepcionSir"
type="java.lang.String"/>
  <ResourceLink global="cas-server-webapp-local/stork/gpoa/uri" name="cas-server-webapp-local/stork/gpoa/uri"
type="java.lang.String"/>
  <ResourceLink global="cas-server-webapp-local/stork/gpoa/pubkey_filename" name="cas-server-webapp-
local/stork/gpoa/pubkey_filename" type="java.lang.String"/>
  <ResourceLink global="cas-server-webapp-local/stork/uriCasRecepcionStork" name="cas-server-webapp-
local/stork/uriCasRecepcionStork" type="java.lang.String"/>
.....................
```

Fichero WEB-INF/web.xml.

Modificamos el fichero WEB-INF/web.xml. Añadimos el filtro LoginSirFilter y nuevos mappings para el servlet cas:

```
      ..............................
   <filter>
      <filter-name>LoginSir Filter</filter-name>
      <filter-class>es.usc.cas.sir.LoginSirFilter</filter-class>
   </filter>
      ..............................
   <filter-mapping>
      <filter-name>LoginSir Filter</filter-name>
      <url-pattern>/loginSir</url-pattern>
   </filter-mapping>
   <filter-mapping>
      <filter-name>LoginSir Filter</filter-name>
      <url-pattern>/loginStork</url-pattern>
   </filter-mapping>
      ..............................
   <servlet-mapping>
      <servlet-name>cas</servlet-name>
      <url-pattern>/loginSir</url-pattern>
   </servlet-mapping>
   <servlet-mapping>
      <servlet-name>cas</servlet-name>
      <url-pattern>/initSirValidation</url-pattern>
   </servlet-mapping>
      <servlet-mapping>
      <servlet-name>cas</servlet-name>
      <url-pattern>/loginStork</url-pattern>
   </servlet-mapping>
   <servlet-mapping>
      <servlet-name>cas</servlet-name>
      <url-pattern>/initStorkValidation</url-pattern>
   </servlet-mapping>
      ..............................
```

Fichero WEB-INF/cas-servlet.xml.

Modificamos el fichero WEB-INF/cas-servlet.xml:

```
      ........................
<!-- Handler Mapping -->
<bean id="handlerMappingB" class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
    <property name="mappings">
   <props>
            <prop key="/login">loginController</prop>
            <prop key="/loginX509">loginController</prop>
            <prop key="/loginSir">loginController</prop>
            <prop key="/loginStork">loginController</prop>
   </props>
    </property>
    <property
   name="interceptors">
   <list>
      <ref bean="localeChangeInterceptor" />
      <ref bean="x509Interceptor" />
   </list>
    </property>
</bean>
<bean id="handlerMappingInitSirValidation"
   class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
   <property
      name="mappings">
      <props>
         <prop
```

```
            key="/initSirValidation">
            InitSirValidationController
          </prop>
      </props>
   </property>
</bean>
<bean id="handlerMappingInitStorkValidation"
   class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
   <property
      name="mappings">
      <props>
         <prop key="/initStorkValidation">
            InitStorkValidationController
         </prop>
      </props>
   </property>
</bean>
.........................................
<bean id="InitSirValidationController" class="es.usc.cas.sir.InitSirValidationController"
      p:peticionSir-ref="peticionSir" />
<bean id="InitStorkValidationController" class="es.usc.cas.stork.InitStorkValidationController"
      p:peticionStork-ref="peticionStork" />
.........................................
<bean id="sirCheck"
        p:centralAuthenticationService-ref="centralAuthenticationService"
        class="es.usc.cas.sir.SirNonInteractiveCredentialsAction" >
        <property name="centralAuthenticationService" ref="centralAuthenticationService"/>
        <property name="peticionSir" ref="peticionSir"/>
</bean>
<bean id="storkCheck"
        p:centralAuthenticationService-ref="centralAuthenticationService"
        class="es.usc.cas.stork.StorkNonInteractiveCredentialsAction" >
        <property name="centralAuthenticationService" ref="centralAuthenticationService"/>
        <property name="peticionStork" ref="peticionStork"/>
</bean>
.......................
```

Fichero WEB-INF/deployerConfigContext.xml

Modificamos el fichero WEB-INF/deployerConfigContext.xml:

- Añadimos nuevos credentialsToPrincipalResolvers para tratar credenciales Sir y Stork.
- Añadimos nuevos authenticationHandlers para autenticar credenciales Sir y Stork.
- Añadimos nuevos beans para tratar peticiones Sir y Stork.

```
    <bean id="authenticationManager"
          class="org.jasig.cas.authentication.AuthenticationManagerImpl">
        ..........................................
        <property name="credentialsToPrincipalResolvers">
            <list>
                ..............................
                <bean
                    class="es.usc.cas.ActiveDirectoryUsernamePasswordCredentialsToPrincipalResolver">
                    <property name="attributeRepositorys">
                        <list>
                            <ref bean="attributeRepository" />
                            <ref bean="attributeRepository2" />
                        </list>
                    </property>
                </bean>
                ..............................
                <bean
                    class="org.jasig.cas.authentication.principal.HttpBasedServiceCredentialsToPrincipalResolver" />
                ..............................
                <bean
                    class="es.usc.cas.x509.X509CertificateCredentialsToPrincipalResolver">
                    <property name="attributeRepositorys">
                        <list>
                            <ref bean="attributeRepository3" />
                            <ref bean="attributeRepository4" />
                        </list>
                    </property>
                </bean>
                <bean
                    class="es.usc.cas.sir.SirCredentialsToPrincipalResolver">
                </bean>
                <bean
                    class="es.usc.cas.stork.StorkCredentialsToPrincipalResolver">
                </bean>
            </list>
        </property>
        ........................
        <property name="authenticationHandlers">
            <list>
                ..............................
                <bean class="org.jasig.cas.authentication.handler.support.HttpBasedServiceCredentialsAuthenticationHandler"
                      p:httpClient-ref="httpClient" />
                ..............................
                <bean class="es.usc.cas.sir.SirCredentialsAuthenticationHandler"/>
                <bean class="es.usc.cas.stork.StorkCredentialsAuthenticationHandler"/>
            </list>
        </property>
    </bean>
```

```
    ....................................
    <bean id="utilesRsa" class="es.usc.cas.utiles.UtilesRsa" />
    <bean id="peticionSir" class="es.usc.cas.sir.PeticionSir">
        <property name="uriGpoaSir">
            <jee:jndi-lookup jndi-name="java:comp/env/cas-server-webapp-local/sir/gpoa/uri"/>
        </property>
        <property name="pubkeyFilenameGpoaSir">
            <jee:jndi-lookup jndi-name="java:comp/env/cas-server-webapp-local/sir/gpoa/pubkey_filename"/>
        </property>
        <property name="uriCasRecepcionSir">
            <jee:jndi-lookup jndi-name="java:comp/env/cas-server-webapp-local/sir/uriCasRecepcionSir"/>
        </property>
        <property name="utilesRsa" ref="utilesRsa"/>
    </bean>
    <bean id="peticionStork" class="es.usc.cas.stork.PeticionStork">
        <property name="uriGpoaStork">
            <jee:jndi-lookup jndi-name="java:comp/env/cas-server-webapp-local/stork/gpoa/uri"/>
        </property>
        <property name="pubkeyFilenameGpoaStork">
            <jee:jndi-lookup jndi-name="java:comp/env/cas-server-webapp-local/stork/gpoa/pubkey_filename"/>
        </property>
        <property name="uriCasRecepcionStork">
            <jee:jndi-lookup jndi-name="java:comp/env/cas-server-webapp-local/stork/uriCasRecepcionStork"/>
        </property>
        <property name="utilesRsa" ref="utilesRsa"/>
    </bean>
    ........................
```

Fichero WEB-INF/login-webflow.xml.

Modificamos el fichero WEB-INF/login-webflow.xml:

```
.............................................
    <decision-state id="gatewayRequestCheck">
   <if test="${externalContext.requestParameterMap['gateway'] != '' && externalContext.requestParameterMap['gateway'] != null
&& flowScope.service != null}" then="redirect" else="isLoginX509" />
    </decision-state>
.............................................
    <decision-state id="renewRequestCheck">
   <if test="${externalContext.requestParameterMap['renew'] != '' && externalContext.requestParameterMap['renew'] != null}"
then="isLoginX509" else="generateServiceTicket" />
    </decision-state>
.............................................
    <decision-state id="isLoginX509">
      <if test="${externalContext.requestMap['org.springframework.web.servlet.HandlerMapping.pathWithinHandlerMapping'] != ''
&& externalContext.requestMap['org.springframework.web.servlet.HandlerMapping.pathWithinHandlerMapping'] == '/loginX509'}"
then="startAuthenticateX509" else="isLoginSir" />
    </decision-state>
    <decision-state id="isLoginSir">
      <if test="${externalContext.requestMap['org.springframework.web.servlet.HandlerMapping.pathWithinHandlerMapping'] != ''
&& externalContext.requestMap['org.springframework.web.servlet.HandlerMapping.pathWithinHandlerMapping'] == '/loginSir'}"
then="startAuthenticateSir" else="isLoginStork" />
    </decision-state>
    <decision-state id="isLoginStork">
      <if test="${externalContext.requestMap['org.springframework.web.servlet.HandlerMapping.pathWithinHandlerMapping'] != ''
&& externalContext.requestMap['org.springframework.web.servlet.HandlerMapping.pathWithinHandlerMapping'] == '/loginStork'}"
then="startAuthenticateStork" else="viewLoginForm" />
    </decision-state>
.............................................
    <action-state id="startAuthenticateX509">autenticación
   <action bean="x509Check" />
   <transition on="success" to="sendTicketGrantingTicket" />
   <transition on="error" to="x509CertificateErrorView" />
    </action-state>
    <action-state id="startAuthenticateSir">
   <action bean="sirCheck" />
   <transition on="success" to="sendTicketGrantingTicket" />
   <transition on="error" to="sirErrorView" />
    </action-state>
    <action-state id="startAuthenticateStork">
   <action bean="storkCheck" />
   <transition on="success" to="sendTicketGrantingTicket" />
   <transition on="error" to="storkErrorView" />
    </action-state>
.............................................
    <end-state id="x509CertificateErrorView" view="x509CertificateErrorView" />
    <end-state id="sirErrorView" view="sirErrorView" />
    <end-state id="storkErrorView" view="storkErrorView" />
.............................................
```

Fichero WEB-INF/view/jsp/protocol/2.0/casServiceValidationSuccess.jsp.

Añadimos los nuevos tipos de identificación.

```
    ..............................
<c:choose>
     <c:when test="${'es.usc.cas.ActiveDirectoryFastBindLdapAuthenticationHandler' == auth.attributes['authenticationMethod']}"
>
         <cas:tipoAutenticacion>Formulario</cas:tipoAutenticacion>
     </c:when>
     <c:when test="${'es.usc.cas.x509.X509CertificateCredentialsAuthenticationHandler' ==
auth.attributes['authenticationMethod']}" >
         <cas:tipoAutenticacion>CertificadoX509</cas:tipoAutenticacion>
```

```
    </c:when>
    <c:when test="${'es.usc.cas.sir.SirCredentialsAuthenticationHandler' == auth.attributes['authenticationMethod']}" >
        <cas:tipoAutenticacion>Sir</cas:tipoAutenticacion>
    </c:when>
    <c:when test="${'es.usc.cas.stork.StorkCredentialsAuthenticationHandler' == auth.attributes['authenticationMethod']}" >
        <cas:tipoAutenticacion>Stork</cas:tipoAutenticacion>
    </c:when>
    <c:otherwise>
        <cas:tipoAutenticacion>Otras</cas:tipoAutenticacion>
    </c:otherwise>
</c:choose>
   ..................................
```

Fichero WEB-INF/view/jsp/usc/ui/casLoginView.jsp

Ejemplo de personalización del formulario de login con añadidos HTML para mostrar los nuevos tipos de identificación.

```
<%@page contentType="text/html" pageEncoding="ISO-8859-1"%>
<jsp:directive.include file="includes/top.jsp" />
<div id="contenedor_cas">
        <form:form method="post" commandName="${commandName}" htmlEscape="true">
            <div id="cabecera_informacion">
                <h1><spring:message code="screen.welcome.instructions" /></h1>
                <p><spring:message code="screen.welcome.intruccionsUSC1" /></p>
                <c:set var="texto_mensaje" scope="request">
                   <spring:theme code="screen.welcome.intruccionsUSC2"/>
                </c:set>
                <p><spring:message code="${texto_mensaje}" /></p>
                <p><spring:message code="screen.welcome.intruccionsUSC3" /></p>

                <form:errors path="*" cssClass="loginError" id="loginError" element="div" htmlEscape="false"/>

                <c:if test="${param.cambiaMetodoIdentificacion == 'true'}">
                   <p class="loginError" id="loginError" ><spring:message code="screen.identificacion.rechazada.message" /></p>
                </c:if>
            </div>
            <c:set var="theme_name" scope="request">
                  <spring:theme code="theme.name" />
            </c:set>
            <div id="capa_credenciais"  <c:if test="${(theme_name == 'usc') or (theme_name ==
'xc')}">style="width:100%;"</c:if> >
              <fieldset class="login">
                <h2><spring:message code="screen.welcome.credenciaisDaUsc"/></h2>
                <p>
                <c:if test="${not empty sessionScope.openIdLocalId}">
                    <strong>${sessionScope.openIdLocalId}</strong>
                    <input type="hidden" id="username" name="username" value="${sessionScope.openIdLocalId}" />
                </c:if>

                <c:if test="${empty sessionScope.openIdLocalId}">
                    <label for="userName"><spring:message code="screen.welcome.label.netid"/></label><br />
                    <spring:message code="screen.welcome.label.netid.accesskey" var="userNameAccessKey" />
                    <form:input id="username" size="25" tabindex="1" accesskey="${userNameAccessKey}" path="username"
autocomplete="false" htmlEscape="true" /> <span><spring:message code="screen.welcome.exemplo" /></span>
                </c:if>
                </p>
                <p>
                    <label for="userPassword"><spring:message code="screen.welcome.label.password"/></label><br />
                    <spring:message code="screen.welcome.label.password.accesskey" var="passwordAccessKey" />
                    <form:password cssClass="required" cssErrorClass="error" id="password" size="25" tabindex="2"
path="password"  accesskey="${passwordAccessKey}" htmlEscape="true" />
                </p>
                <p>
                <input id="warn" name="warn" value="true" tabindex="3" accesskey="<spring:message
code="screen.welcome.label.warn.accesskey" />" type="checkbox" />
                        <label for="warn"><spring:message code="screen.welcome.label.warn" /></label>
                </p>
                <p>
                    <input type="hidden" name="lt" value="${flowExecutionKey}" />
                    <input type="hidden" name="_eventId" value="submit" />
                    <input name="submit" accesskey="l" value="<spring:message code="screen.welcome.button.login" />"
tabindex="4" type="submit" />
                </p>
              </fieldset>
            </div>

            <c:if test="${theme_name == 'all'}">
                <div id="capa_otros_metodos" style="background: url(<%=request.getContextPath()%>/<spring:theme
code='pixel_n'/>) top left repeat-y #FFF ;">
                    <h2><spring:message code="screen.welcome.outrosMediosAutenticacion"/></h2>
                                <dl id="metodos">
                            <dt><a href="loginX509?service=${param.service}"><img src="<%=request.getContextPath()
%>/<spring:theme code="logo.edni"/>" alt="logo.dnie"/><spring:message
code="screen.welcome.outrosMediosAutenticacion.x509"/></a></dt>
                        <dd><spring:message code="screen.welcome.outrosMediosAutenticacion.x509.descripcion"/></dd>

                            <dt><a href="initStorkValidation?service=${param.service}"><img src="<%=request.getContextPath()
%>/<spring:theme code="logo.stork"/>" alt="logo.stork"/><spring:message
code="screen.welcome.outrosMediosAutenticacion.stork"/></a></dt>
                        <dd><spring:message code="screen.welcome.outrosMediosAutenticacion.stork.descripcion"/></dd>
```

```
                        <dt><a href="initSirValidation?service=${param.service}"><img src="<%=request.getContextPath()
%>/<spring:theme code="logo.rediris"/>" alt="logo.rediris" /><spring:message
code="screen.welcome.outrosMediosAutenticacion.sir"/></a></dt>
                                <dd><spring:message code="screen.welcome.outrosMediosAutenticacion.sir.descripcion"/></dd>
                    </dl>
                </div>
            </c:if>

 </form:form>
</div>
<script language="JavaScript">
<!--
   document.getElementById("${commandName}").username.focus();
//-->
</script>
<jsp:directive.include file="includes/bottom.jsp" />
```