G CONSELLERIA
O ADMINISTRACIONS
I PÚBLIQUES I
B MODERNITZACIÓ
/ DIRECCIÓ GENERAL
MODERNITZACIÓ I
ADMINISTRACIÓ DIGITAL

Serveis d'Administració Electrònica en el Govern de les Illes Balears

Intermediació de dades

Eines d'integració SCSP

Palma, 11 de març de 2020

Índex de continguts

1. Introducció i objectius

1. Introducció i objectius

Eines de suport a la intermediació de dades SCSP

Objectius

SCSP

PINBAL

RECOBRIMENT

- Connexió SOAP al ws recobriment de pinbal i als procediments que ofereix.

INTEGRACIO EN APLICACIONS

- Simplificar l'us del recobriment per als casos més habituals. Desenvolupar tests per a cada cas d'us.

EMISERV

EMISERV BACKOFFICE

- Facilitar el desenvolupament dels backoffices EMISERV a les aplicacions corporatives.

SERVEIS ASÍNCRONS

- Ampliar l'abast de EMISERV als serveis asíncrons i als seus requeriments de negoci.

Emissors

SCSP BACKOFFICE

- Simplificar el desenvolupament de backoffices SCSP als casos que no s'ha implantat EMISERV i alleugerir l'ús de xml

Esquemes XSD

Datos Genéricos/Datos Específicos

- Aprofitar les funcionalitats de PINBAL i EMISERV que gestionen les parts del missatge SOAP SCSP que inclouen dades comunes per a tots els serveis (Datos Genéricos i proporcionar un conjunt de classes Java generades a partir dels esquemes de cada servei per als objectes Datos Especificos . A més a més, facilitar la gestió del xml per a qualsevosl missatge SCSP

Utilitats

Eines de propòsit general

- Proporcionar utilitats de configuració de connexions SOAP i tractament de XML mitjançant POJOs, i totes aquelles que calguin per aquesta finalitat.

Objectius planificats coberts
Objectius planificats no coberts
Nous objectius coberts

2. Productes que composen la solució

2. Productes que composen la solució

Eines de suport a la intermediació de dades SCSP

Tots els productes es troben al repositori GitHub del Govern

https://github.com/GovernIB/emissors i es divideixen en els projectes a continuació

scsp-pinbal

Client SOAP per al recobriment PINBAL

Abstract Facade i Adapter per integrar el client a les aplicacions

codapp-pinbal

Facade específic per al servei consultat

Classes de DatosEspecificos amb nom únic depenent del servei

Exemple de client especific que fa servir les classes anteriors

Tests Junit per als clients

scsp-schemas-xsd-main

Projectes Maven que subministren classes Java mitjançant JAXB per a cada servei SCSP configurable a PINBAL

scsp-schemas-gen

Eines de generació dels projectes scsp-schemas-xsd-{servei} a partir dels esquemes xsd configurats a pinbal

scsp-commons

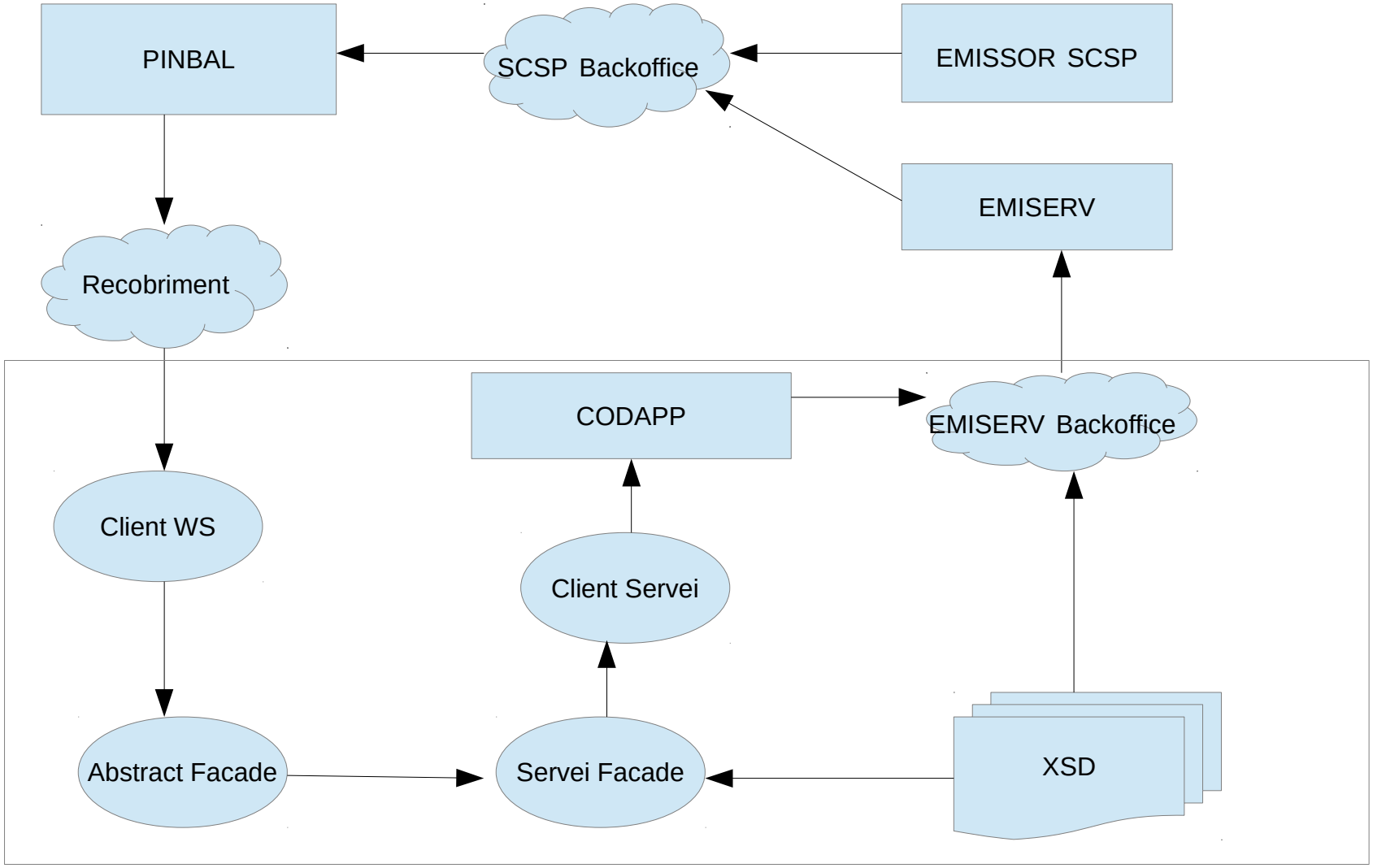
Eines de suport al tractament de XML i paràmetres de connexions de clients SOAP

2. Productes que composen la solució

Eines de suport a la intermediació de dades SCSP

Tots els productes es troben al repositori GitHub del Govern

https://github.com/GovernIB/emissors i es divideixen en els projectes a continuació



3. Sol·lucions als objectius

3. Sol·lucions als objectius

Eines de suport a la intermediació de dades SCSP

Suposicions prèvies

Tots els projectes tenen l'estructura habitual:

- Projecte maven principal
- Subprojecte scripts
- Un o més projectes que contenen les funcionalitats.
- La versió de CXF es la habitual

3. Sol·lucions als objectius

Eines de suport a la intermediació de dades SCSP

Generació del Client SOAP a partir del WSDL

Generació del client SOAP a partir del WSDL: Afegim el codi al fitxer pom.xml del projecte client

```
<plugin>
      <groupId>org.apache.cxf</groupId>
      <artifactId>cxf-codegen-plugin</artifactId>
      <version>${cxf.version}</version>
      <executions>
          <execution>
              <id>pinbal-ws-recobriment</id>
              <phase>generate-sources</phase>
              <configuration>
                  <!--sourceRoot>${basedir}/target/generated/cxf-codegen-plugin</sourceRoot-->
                  <wsdlOptions>
                      <wsdlOption>
                          <!--wsdl>${basedir}/src/main/resources/wsdl/recobriment.wsdl</wsdl-->
                          <wsdl>https://proves.caib.es/pinbal/ws/recobriment?wsdl</wsdl>
                          <extraargs>
                              <extraarg>-xjc-Xts</extraarg>
                              <!--extraarg>-xjc-npa</extraarg-->
                          </extraargs>
                      </wsdlOption>
                  </wsdlOptions>
              </configuration>
              <goals>
                  <goal>wsdl2java</goal>
              </goals>
          </execution>

      </executions>
      <dependencies>
          <dependency>
              <groupId>xerces</groupId>
              <artifactId>xercesImpl</artifactId>
              <version>2.9.1</version>
          </dependency>
          <dependency>
              <groupId>org.apache.cxf</groupId>
              <artifactId>cxf-xjc-ts</artifactId>
              <version>${cxf.version}</version>
          </dependency>
      </dependencies>
  </plugin>
```

3. Sol·lucions als objectius

Eines de suport a la intermediació de dades SCSP

Generació del Client SOAP a partir del WSDL

Obtenció del wsdl en entorns autenticats: Afegim el codi al fitxer pom.xml del projecte client

```
<plugin>
        <groupId>org.codehaus.mojo</groupId>
        <artifactId>exec-maven-plugin</artifactId>
        <version>1.4.0</version>
        <executions>
            <execution>
                <phase>generate-sources</phase>
                <goals>
                    <goal>java</goal>
                </goals>
            </execution>
        </executions>
        <configuration>
            <mainClass>es.caib.scsp.utils.cxf.authentication.AuthenticatorReplacer</mainClass>
            <arguments>
                <argument>$codapp_pinbal</argument>
                <argument>codapp_pinbal</argument>
            </arguments>
        </configuration>
    </plugin>
```

Els detalls de implementació de la classe AuthenticationReplacer es troben al projecte scsp-commons.

Compilam el projecte i les classes generades es troben a la carpeta Generated Sources (cxf) al package es.caib.pinbal.ws.recobriment

3. Sol·lucions als objectius

Eines de suport a la intermediació de dades SCSP

Implementació del client fent servir patrons Proxy i Singleton.

Cream un objecte proxy que ens centralitzarà tota la interacció amb el WS recobriment. El singleton ens proporciona una còpia única del objecte. Els detalls de connexió es troben a DadesConnexioRecobriment que s'estén de la classe DadesConnexioSOAP de scsp-commons. Cal configurar les propietats següents

es.caib.codapp.pinbal.client.username=$codapp_pinbal
es.caib.codapp.pinbal.client.username=codapp_pinbal
es.caib.codapp.pinbal.client.baseURL=«https://proves.caib.es/pinbal»

```
  private DadesConnexioSOAP dadesConnexio;

  public DadesConnexioSOAP getDadesConnexio() {
      return dadesConnexio;
  }

  public void setDadesConnexio(DadesConnexioSOAP dadesConnexio) {
      this.dadesConnexio = dadesConnexio;
  }


 private static final RecobrimentClient client = new RecobrimentClient();


  private RecobrimentClient() {
      super();
  }


  private static RecobrimentClient _getClient(DadesConnexioRecobriment dadesConnexio) {
      client.setDadesConnexio(dadesConnexio);
      return client;
  }


  public static RecobrimentClient getClient(DadesConnexioRecobriment dadesConnexio) {
      DadesConnexioRecobriment dct = (dadesConnexio != null) ? dadesConnexio : new DadesConnexioRecobriment("");
      return _getClient(dct);
  }
```

3. Sol·lucions als objectius

Eines de suport a la intermediació de dades SCSP

Obtenció del client amb els paràmetres de connexió configurats.

Obtenim la interfície del client (Recobriment) on configuram els endpoints, usuari i password. A més a més, gestionam els certificats de servidor amb AuthenticatorReplacer.

```
private static final QName SERVICE_NAME = new QName(DadesConnexioRecobriment._QNAME,
         DadesConnexioRecobriment._SERVICE_NAME);

private Recobriment getServicePort() {

     AuthenticatorReplacer.verifyHost();

     URL wsdlURL = null;

     try {
         LOG.info(dadesConnexio.getWsdlLocation());
         wsdlURL = new URL(dadesConnexio.getWsdlLocation());
     } catch (MalformedURLException ex) {
         Logger.getLogger(RecobrimentClient.class.getName()).log(Level.SEVERE, null, ex);
     }

     String userName = dadesConnexio.getUserName();
     String password = dadesConnexio.getPassword();

     AuthenticatorReplacer.setAuthenticator(userName, password);

     LOG.log(Level.INFO, "Servicio:  {0}", SERVICE_NAME);
     LOG.log(Level.INFO, "URL: {0}", wsdlURL);

     RecobrimentService ss = new RecobrimentService(wsdlURL, SERVICE_NAME);
     Recobriment port = ss.getRecobrimentServicePort();

     Map<String, Object> req = ((BindingProvider) port).getRequestContext();

     req.put(BindingProvider.ENDPOINT_ADDRESS_PROPERTY, dadesConnexio.getEndPoint());

     req.put(BindingProvider.USERNAME_PROPERTY, dadesConnexio.getUserName());
     req.put(BindingProvider.PASSWORD_PROPERTY, dadesConnexio.getPassword());

     return port;

 }
```

3. Sol·lucions als objectius

Eines de suport a la intermediació de dades SCSP

Obtenció del client amb els paràmetres de connexió configurats.

Si cal accedir els missatges SOAP de petició i resposta, afegim el handler RecobrimentSOAPHandler al BindingProvider un cop obtingut el ServicePort

```
private Recobriment getHandledServicePort() {

      Recobriment port = getServicePort();

      RecobrimentSOAPHandler sh = new RecobrimentSOAPHandler();

      List<Handler> handlerChain = new ArrayList<Handler>();
      handlerChain.add(sh);

      BindingProvider bindingProvider = ((BindingProvider) port);

      bindingProvider.getBinding().setHandlerChain(handlerChain);

      return port;

  }

  public class RecobrimentSOAPHandler implements
      javax.xml.ws.handler.soap.SOAPHandler<SOAPMessageContext> {...}
```

3. Sol·lucions als objectius

Eines de suport a la intermediació de dades SCSP

Invocació dels mètodes del recobriment.

El proxy ofereix uns mètodes públics (amb i sense handler) que fan ús de un únic mètode privat.

El métode que oferim es la petició síncrona.

```
public Respuesta peticionSincrona(Peticion pet) {
     Recobriment port = getServicePort();
     Respuesta response;
     response = peticionSincrona(port, pet);
     return response;
 }



 private static Respuesta peticionSincrona(Recobriment port, Peticion pet) {
     LOG.log(Level.INFO, "Invoking port...");
     Respuesta _peticionSincrona__return = port.peticionSincrona(pet);
     LOG.log(Level.INFO, "Return port...");
     return _peticionSincrona__return;
 }
```

3. Sol·lucions als objectius

Eines de suport a la intermediació de dades SCSP

Invocació dels mètodes del recobriment.

El proxy ofereix uns mètodes públics (amb i sense handler) que fan ús de un únic mètode privat.

```
public Respuesta peticionSincrona(Peticion pet, boolean handled) {

      if (!handled) return peticionSincrona(pet);

      Recobriment port = getHandledServicePort();

      Respuesta response;

      response = peticionSincrona(port, pet);

      Map<String, Object> res = ((BindingProvider) port).getResponseContext();

      for (TransmisionDatos transmisionDatos:response.getTransmisiones().getTransmisionDatos()){
              Transmision transmision = transmisionDatos.getDatosGenericos().getTransmision();

              String key = RecobrimentSOAPHandler.DATOS_ESPECIFICOS + "."
                      + transmision.getIdSolicitud() + "." + transmision.getIdTransmision();
              Element datosEspecificos = XmlUtils.node2Element((Element)res.get(key));
              transmisionDatos.setDatosEspecificos(datosEspecificos);
      }
      return response;
  }
```

3. Sol·lucions als objectius

Eines de suport a la intermediació de dades SCSP

Còm assignar els valors als objectes Peticion. Facade i Adapters

La classe RecobrimentUtils proporciona utilitats per assignar valors als objectes generats. La abstracció dels detalls la obtindrem amb el component RecobrimentFacade. Els detalls mes importants son el constructor i els métodes abstractes. A més a més ens ofereix métodes per als casos més habituals: la petició de una única transmissió.

```
public abstract class RecobrimentFacade<TDatosEspecificosPeticion, TDatosEspecificosRespuesta> {

    private static String APP = "es.caib.scsp.";

    protected static final Logger LOG = Logger.getLogger(RecobrimentFacade.class.getName());

    private RecobrimentClient client;

    public RecobrimentFacade(){
        this(APP);
    }

    public RecobrimentFacade(String app){
        DadesConnexioRecobriment dadesConnexio = new DadesConnexioRecobriment(app);
        this.client = RecobrimentClient.getClient(dadesConnexio);
    }

    public Respuesta peticionSincrona(Peticion peticion){
        return this.client.peticionSincrona(peticion);
    }

    public RespuestaClientAdapter<TDatosEspecificosRespuesta> peticionSincrona(PeticionClientAdapter<TDatosEspecificosPeticion> peticionClient){
        Peticion peticion = peticionClient2Peticion(peticionClient);
        Respuesta response = peticionSincrona(peticion);
        RespuestaClientAdapter<TDatosEspecificosRespuesta> respuesta = respuesta2RespuestaClientAdapter(response);
        return respuesta;
    }

    protected abstract RespuestaClientAdapter<TDatosEspecificosRespuesta> peticionSincronaEspecifica(..)

...

protected abstract Element datosEspecificos2Element(TDatosEspecificosPeticion datosEspecificosPeticion);

protected abstract TDatosEspecificosRespuesta element2DatosEspecificos(Element elementDatosEspecificos);
```

3. Sol·lucions als objectius

Eines de suport a la intermediació de dades SCSP

Còm implementar els mètodes abstractes de facade a un servei específic.

El projecte d'exemple codapp-pinbal ens proporciona implementacions de facade per serveis específics i formarà part de la aplicació client.
Ens caldrà substituir el codi codapp per els codis d'aplicació al fitxer pom i als packages, afegir les dependències maven i definir les classes DatosEspecificos. Exemple de dependències al pom.xml

```
        <dependency>
            <groupId>es.caib.scsp</groupId>
            <artifactId>scsp-utils</artifactId>
            <version>1.0.0</version>
        </dependency>
        <dependency>
            <groupId>es.caib.scsp</groupId>
            <artifactId>scsp-schemas-xsd-SCDHPAJUv3</artifactId>
            <version>1.0.0</version>
            <type>jar</type>
        </dependency>
        <dependency>
            <groupId>es.caib.scsp</groupId>
            <artifactId>pinbal-ws-client</artifactId>
            <version>1.0.0</version>
            <type>jar</type>
        </dependency>
```

Exemples de DatosEspecificos peticio i resposta depenent del servei.

```
@XmlRootElement(name = "datosEspecificos")
public class SCDHPAJUv3PeticionDatosEspecificos
        extends es.caib.scsp.esquemas.SCDHPAJUv3.peticion.datosespecificos.DatosEspecificos {
    public SCDHPAJUv3PeticionDatosEspecificos(){
        super();
    }
}

@XmlRootElement(name = "datosEspecificos")
public class SCDHPAJUv3RespuestaDatosEspecificos
        extends es.caib.scsp.esquemas.SCDHPAJUv3.respuesta.datosespecificos.DatosEspecificos {
    public SCDHPAJUv3RespuestaDatosEspecificos(){
        super();
    }
}
```

3. Sol·lucions als objectius

Eines de suport a la intermediació de dades SCSP

Còm implementar els mètodes abstractes de facade a un servei específic.

Un cop tenim resoltes les dependències del servei implementam els mètodes abstractes de RecobrimentFacade i els mètodes privats especifics del servei. La classe XmlManager es un conjunt de utilitats a scsp-commons.

```
public class SCDHPAJUv3RecobrimentFacade
        extends RecobrimentFacade<
        SCDHPAJUv3PeticionDatosEspecificos, SCDHPAJUv3RespuestaDatosEspecificos> {

    public SCDHPAJUv3RecobrimentFacade(String app) {
        super(app);
    }

    @Override
    protected Element datosEspecificos2Element(SCDHPAJUv3PeticionDatosEspecificos datosEspecificosPeticion){

        Element elementDatosEspecificos;
        try {
            XmlManager<SCDHPAJUv3PeticionDatosEspecificos> manager
                    = new XmlManager<SCDHPAJUv3PeticionDatosEspecificos>(SCDHPAJUv3PeticionDatosEspecificos.class);
            elementDatosEspecificos = manager.generateElement(datosEspecificosPeticion);
            return elementDatosEspecificos;
        } catch (JAXBException ex) {..}
        return null;
    }

    private SCDHPAJUv3PeticionDatosEspecificos establecerDatosEspecificosPeticion(
            String municipioSolicitud, String numeroAnyos, String provinciaSolicitud,
            String nombreTipoDocumentacion, String valorDocumentacion, String NIA){..}

    public RespuestaClientAdapter<SCDHPAJUv3RespuestaDatosEspecificos> peticionSincrona(...)

    @Override
    protected SCDHPAJUv3RespuestaDatosEspecificos element2DatosEspecificos(Element elementDatosEspecificos) {

        SCDHPAJUv3RespuestaDatosEspecificos datosEspecificos;
        try {
            XmlManager<SCDHPAJUv3RespuestaDatosEspecificos> manager
                    = new XmlManager<SCDHPAJUv3RespuestaDatosEspecificos>(SCDHPAJUv3RespuestaDatosEspecificos.class);
            datosEspecificos = manager.generateItem(elementDatosEspecificos);
            return datosEspecificos;
        } catch (JAXBException ex) {..}
        return null;
    }
```

3. Sol·lucions als objectius

Eines de suport a la intermediació de dades SCSP

Còm fer ús dels objectes facade.

La aplicació codapp inclourà el projecte codapp-pinbal com subprojecte i podrà fer ús de la classe client {Servei}Client que a la vegada invocarà el mètode més convenient del facade específic.

```
public class SCDHPAJUv3Client {

    private static String APP = "es.caib.codapp.";

    private SCDHPAJUv3RecobrimentFacade facade;

    public SCDHPAJUv3Client(){
        this(APP);
    }

    public SCDHPAJUv3Client(String app){
        this.facade = new SCDHPAJUv3RecobrimentFacade(app);
    }

    public void dummy(){
        _dummy();
    }

    private static void _dummy() {
        LOG.log(Level.INFO, "Invoking dummy...");
    }

    public RespuestaClientAdapter<SCDHPAJUv3RespuestaDatosEspecificos> peticionSincrona(){

    RespuestaClientAdapter<SCDHPAJUv3RespuestaDatosEspecificos> respuestaClient =
            facade.peticionSincrona(
             codigoEstado, codigoEstadoSecundario, literalError, literalErrorSec, tiempoEstimadoRespuesta,
             codigoCertificado, idPeticion, numElementos, timeStamp, nifEmisor, nombreEmisor, nifFuncionario,
             nombreCompletoFuncionario, seudonimo, codProcedimiento,nombreProcedimiento, codigoUnidadTramitadora,
             consentimiento, finalidad, idExpediente, identificadorSolicitante, nombreSolicitante, unidadTramitadora,
             apellido1, apellido2, documentacion, nombre, nombreCompleto, tipoDocumentacion, fechaGeneracion,
             idSolicitud, idTransmision, municipioSolicitud, numeroAnyos, provinciaSolicitud, nombreTipoDocumentacion,
             valorDocumentacion, NIA);
        return respuestaClient;
    }

    public static void main(String args[]) throws Exception {
    }
}
```

3. Sol·lucions als objectius

Eines de suport a la intermediació de dades SCSP

Còm fer ús dels objectes facade.

Finalment cream una instància del {Servei}Client als components de negoci de la aplicació codapp i executam els mètodes que calguin.

Exemple de test

```
public class SCDHPAJUv3ClientTest {

    private SCDHPAJUv3Client client;

    public SCDHPAJUv3ClientTest() {}

    @Before
    public void setUp() {
        String app = "es.caib.codapp.";
        DadesConnexioRecobriment dadesConnexio = new DadesConnexioRecobriment(app);
        System.setProperty(app  + "pinbal.client.username", "$codapp_pinbal");
        System.setProperty(app  + "pinbal.client.password", "codapp_pinbal");
        System.setProperty(app  + "pinbal.client.baseURL", "https://proves.caib.es/pinbal");
        client = new SCDHPAJUv3Client(app);
    }

    @Test
    public void testPeticionSincrona() {
        System.out.println("peticionSincrona");

        RespuestaClientAdapter<SCDHPAJUv3RespuestaDatosEspecificos> expResult = null;
        RespuestaClientAdapter<SCDHPAJUv3RespuestaDatosEspecificos> result = client.peticionSincrona();

        System.out.println("APELLIDO  " +
result.getTransmisionesClient().get(0).getDatosEspecificos().getResultado().getApellido1());
    }
```

3. Sol·lucions als objectius

Eines de suport a la intermediació de dades SCSP

Còm fer ús dels objectes facade.

Finalment cream una instància del {Servei}Client als components de negoci de la aplicació codapp i executam els mètodes que calguin.

Resultat del test

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running es.caib.codapp.pinbal.ws.recobriment.client.SCDHPAJUv3ClientTest
peticionSincrona
mar 10, 2020 5:05:02 PM es.caib.scsp.pinbal.ws.recobriment.client.RecobrimentClient getServicePort
INFORMACIÓN: https://proves.caib.es/pinbal/ws/recobriment?wsdl
mar 10, 2020 5:05:02 PM es.caib.scsp.pinbal.ws.recobriment.client.RecobrimentClient getServicePort
INFORMACIÓN: Servicio:  {http://www.caib.es/pinbal/ws/recobriment}RecobrimentService
mar 10, 2020 5:05:02 PM es.caib.scsp.pinbal.ws.recobriment.client.RecobrimentClient getServicePort
INFORMACIÓN: URL: https://proves.caib.es/pinbal/ws/recobriment?wsdl
mar 10, 2020 5:05:04 PM es.caib.scsp.pinbal.ws.recobriment.client.RecobrimentClient peticionSincrona
INFORMACIÓN: Invoking port...
mar 10, 2020 5:05:25 PM es.caib.scsp.pinbal.ws.recobriment.client.RecobrimentClient peticionSincrona
INFORMACIÓN: Return port...
APELLIDO  REBASSA
dummy
mar 10, 2020 5:05:25 PM es.caib.codapp.pinbal.ws.recobriment.client.SCDHPAJUv3Client _dummy
INFORMACIÓN: Invoking dummy...
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 24.843 sec

Results :

Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
```

4. Esquemes XSD i generació amb JAXB

4. Esquemes XSD i generacio amb JAXB

Eines de suport a la intermediació de dades SCSP

Còm fer ús dels esquemes.

Els objectes especifics de cada servei els trobam als projectes sota la carpeta scsp-schemas-xsd-main. Tots els subprojectes generen els objectes Java a partir dels esquemes xsd corresponents mitjançant el plugin Maven maven-jaxb2-plugin al fitxer pom.xml

```
            <plugin>
               <groupId>org.jvnet.jaxb2.maven2</groupId>
               <artifactId>maven-jaxb2-plugin</artifactId>
               <version>0.13.3</version>
               <executions>
                   <execution>
                       <id>SCDHPAJUv3_peticion</id>
                       <phase>generate-sources</phase>
                       <goals>
                           <goal>generate</goal>
                       </goals>
                       <configuration>
                           <xjcArgs>
                              <xjcArg>-XautoNameResolution</xjcArg>
                           </xjcArgs>
                           <schemaDirectory>src/main/resources/jaxb/SCDHPAJUv3/peticion</schemaDirectory>
                           <schemaIncludes>
                              <include>*.xsd</include>
                           </schemaIncludes>
                           <generateDirectory>${project.build.directory}/generated-sources/xjc-SCDHPAJUv3-peticion</generateDirectory>
                           <episodeFile>
                                 ${project.build.directory}/generated-sources/xjc/META-INF/jaxb-schemas-SCDHPAJUv3-peticion.episode
                           </episodeFile>
                           <bindingDirectory>src/main/resources/jaxb/SCDHPAJUv3/peticion</bindingDirectory>
                           <bindingIncludes>
                               <include>*.xjb</include>
                           </bindingIncludes>
                       </configuration>
                   </execution>
                   ....
           </plugin>
```

4. Esquemes XSD i generacio amb JAXB

Eines de suport a la intermediació de dades SCSP

Còm fer ús dels esquemes.

Els objectes especifics de cada servei els trobam als projectes sota la carpeta scsp-schemas-xsd-main. Tots els subprojectes generen els objectes Java a partir dels esquemes xsd corresponents mitjançant el plugin Maven maven-jaxb2-plugin al fitxer pom.xml

```
            <plugin>
               <groupId>org.jvnet.jaxb2.maven2</groupId>
               <artifactId>maven-jaxb2-plugin</artifactId>
               <version>0.13.3</version>
               <executions>
                   ...
                   <execution>
                       <id>SCDHPAJUv3_respuesta</id>
                       <phase>generate-sources</phase>
                       <goals>
                           <goal>generate</goal>
                       </goals>
                       <configuration>
                           <xjcArgs>
                               <xjcArg>-XautoNameResolution</xjcArg>
                           </xjcArgs>
                           <schemaDirectory>src/main/resources/jaxb/SCDHPAJUv3/respuesta</schemaDirectory>
                           <schemaIncludes>
                               <include>*.xsd</include>
                           </schemaIncludes>
                           <generateDirectory>${project.build.directory}/generated-sources/xjc-SCDHPAJUv3-respuesta</generateDirectory>
                           <episodeFile>
               ${project.build.directory}/generated-sources/xjc/META-INF/jaxb-schemas-SCDHPAJUv3-respuesta.episode
                           </episodeFile>
                           <bindingDirectory>src/main/resources/jaxb/SCDHPAJUv3/respuesta</bindingDirectory>
                           <bindingIncludes>
                               <include>*.xjb</include>
                           </bindingIncludes>
                       </configuration>
                   </execution>
               </executions>
               <configuration>
                   <forceRegenerate>false</forceRegenerate>
                   <strict>false</strict>
                   <verbose>true</verbose>
               </configuration>
           </plugin>
```

4. Esquemes XSD i generacio amb JAXB

Eines de suport a la intermediació de dades SCSP

Generacio dels projectes scsp-schemas-xsd-{servei}

Els projectes per obtenir les classes Java a partir dels esquemes xsd es generen automàticament amb les eines del projecte scsp-schemas-gen.

Executam la linea de comandes o el fitxer schemas.bat (schemas.sh)

mvn exec:java -Dexec.mainClass="es.caib.scsp.genschemas.GenSplits"

Les utilitats relacionades generen els projectes a partir dels esquemes http://maven.apache.org/xsd/maven-4.0.0.xsd i bindingschema_2_0.xsd i es fan servir els objectes XmlManager per generar el XML

El contingut del fitxer pom.xml es configura a la classe MainProjectGenerator per al projecte scsp-schemas-xsd-main, i a la classe ServiceProjectGenerator per als subprojectes. Ambdues s'estenen de la classe abstracta ProjectGenerator

El procés consisteix en obtenir el contingut dels esquemes xsd dels fonts de Pinbal a la carpeta schemas.

CodeSource src = XmlHelper.class.getProtectionDomain().getCodeSource();

Per a cada servei cream un projecte i copiam els seus esquemes.

Tots els detalls es poden trobar a les classes ProjectGenerator i les seves extensions.

5. Problemes pendents

5. Problemes pendents

Eines de suport a la intermediació de dades SCSP

Problemes pendents de resoldre

Pendent la estandarització del desenvolupament de EMISERV Backoffice per al que fa als DatosEspecificos

Pendent la estandarització del desenvolupament de SCSP Backoffice per totes les parts del missatge SOAP

No s'ha implementat cap servei asíncron

No es recupera el justificant del certificat Pinbal

6. Properes passes

6. Properes passes

Eines de suport a la intermediació de dades SCSP

Passes a realitzar

1) Desenvolupament de un model de EMISERV Backoffice

2) Anàlisi de còm recuperar els esquemes xsd actualitzats en comptes de substituir-los manualment

3) Anàlisi de serveis asíncrons a Emiserv amb els objectes generats

4) Desenvolupament de la funcionalitat de recuperació del justificant de la consulta al recobriment i al abstract facade de scsp-pinbal.

7. Propostes de solució

7. Propostes de solució a mig termini

Eines de suport a la intermediació de dades SCSP

1) Suport als Ajuntaments, Consells i altres entitats que montin emissors SCSP

   Implantació de EMISERV en comptes de emissor SCSP

   Scripts de desplegament de imatges Docker per reduïr la dependència de la infraestructura

      Ja desenvolupat per BBDD Postgres 9.5

      Pendent de construir Dockerfile per JBOSS 5.1 o Wildfly 15

2) Desenvolupament de un model de EMISERV Backoffice

   Desenvolupament d'una aplicació J2EE mínima per exposar un EMISERVBackoffice

   Generació de DatosEspecíficos amb les eines XML JAXB ja desenvolupades per scsp-pinbal

7. Propostes de solució a mig termini

Suport als Ajuntaments, Consells i altres entitats que mantenen emissors SCSP.

Dockerfile o configuració de VM ja desenvolupat per BBDD Postgres 9.5

Docker-postgresql

Dockerfile

```
FROM postgres:9.5

# Environment variables required for this build (do NOT change)
# -------------------------------------------------------------

ENV  PGTABLESPACES $PGDATA/tablespaces

# Copy files needed during database creation
# ------------

COPY ./docker-entrypoint-initdb.d/00_initdb.sh /docker-entrypoint-initdb.d/
COPY ./docker-entrypoint-initdb.d/scripts/ /docker-entrypoint-initdb.d/scripts/
#COPY ./docker-entrypoint-initdb.d/sql/ /docker-entrypoint-initdb.d/sql/

# Define default entrypoint

ENTRYPOINT ["docker-entrypoint.sh"]
EXPOSE 5432
CMD ["postgres"]
```

```
./fresh_build_and_run.sh -codapp=emiserv -app=ems  -pass=secret  -p=5436
./persistent_build_and_run.sh -codapp=emiserv -app=ems  -pass=secret  -p=5436
```

Pendent de construir Dockerfile per JBOSS 5.1 o Wildfly 15

7. Propostes de solució a mig termini

Desenvolupament de un model de Emiserv Backoffice.

Aplicació J2EE mínima per exposar un EMISERVBackoffice

.

Plugin per generar les classes al servidor

```
<plugin>
      <groupId>org.codehaus.mojo</groupId>
      <artifactId>jaxws-maven-plugin</artifactId>
      <executions>
            <execution>
                  <id>generate-reports-ws-code</id>
                  <phase>generate-sources</phase>
                  <goals>
                        <goal>wsimport</goal>
                  </goals>
                  <configuration>
                        <wsdlUrls>
                              <wsdlUrl>${project.basedir}/src/main/resources/backoffice.wsdl</wsdlUrl>
                        </wsdlUrls>
                        <vmArgs>
                              <vmArg>-Djavax.xml.accessExternalSchema=all</vmArg>
                        </vmArgs>
                        <sourceDestDir>${basedir}/src/main/java</sourceDestDir>
                  </configuration>
            </execution>
      </executions>
</plugin>
```

7. Propostes de solució a mig termini

Desenvolupament de un model de Emiserv Backoffice.

Aplicació J2EE mínima per exposar un EMISERVBackoffice

Cada organisme te una plataforma tecnològica diferent. Integram la funcionalitat dins un ws que implementa EMISERVBackoffice i delega les particularitats de la consulta a un EJB

```
@SecurityDomain(Constants.SECURITY_DOMAIN)
@Stateless(name = AplicacioEmiservBackoffice.NAME + "Ejb")
@RolesAllowed({Constants.CMI_USER, Constants.CMI_ADMIN})
@SOAPBinding(style = SOAPBinding.Style.RPC)
@org.apache.cxf.interceptor.InInterceptors(interceptors =
{"es.caib.aplicacio.ws.utils.AplicacioInInterceptor"})
@org.apache.cxf.interceptor.InFaultInterceptors(interceptors =
{"es.caib.aplicacio.ws.utils.AplicacioInInterceptor"})
@WebService(name = AplicacioEmiservBackoffice.NAME_WS, portName = AplicacioEmiservBackoffice.NAME_WS,
serviceName = AplicacioEmiservBackoffice.NAME_WS
        + "Service")
@WebContext(contextRoot = "/aplicacio/ws", urlPattern = "/v1/"
        + AplicacioEmiservBackoffice.NAME, transportGuarantee = TransportGuarantee.NONE, secureWSDLAccess =
false, authMethod = "WSBASIC")
public class AplicacioEmiservBackoffice extends AuthenticatedBaseWsImpl implements EmiservBackoffice {

    @EJB(mappedName = "aplicacio/DominiConsultaAplicacioLogicaEJB/local")
    private DominiConsultaAplicacioLogicaLocal dominiConsultaAplicacioLogicaEjb;

    public static final String NAME = "AplicacioEmiservBackoffice";

    public static final String NAME_WS = NAME + "Ws";

    @Resource
    private WebServiceContext wsContext;

....

@RolesAllowed({...})
@WebMethod
@Override
public PeticionSincronaResponse peticionSincrona(...)
```

7. Propostes de solució a mig termini

Generació de DatosEspecíficos amb les eines XML JAXB ja desenvolupades per scsp-pinbal. Còm fer ús dels esquemes.

Els objectes especifics de cada servei els trobam als projectes sota la carpeta scsp-schemas-xsd-main. Tots els subprojectes generen els objectes Java a partir dels esquemes xsd corresponents mitjançant el plugin Maven maven-jaxb2-plugin al fitxer pom.xml

Substitucio de la generació manual de DatosEspecificos

```
private Element createDatosEspecificos(
                      SolicitudTransmision solicitudTransmision,
                      int versioNum) throws ParserConfigurationException {
               DocumentBuilderFactory fac = DocumentBuilderFactory.newInstance();
               Document doc = fac.newDocumentBuilder().newDocument();
               Element datosEspecificos = doc.createElement("DatosEspecificos");
               Element retorno = doc.createElement("Retorno");
               Element estado = doc.createElement("Estado");
               Element codigoEstado = doc.createElement("CodigoEstado");
               codigoEstado.setTextContent("0003");
               estado.appendChild(codigoEstado);
               Element literalError = doc.createElement("LiteralError");
               literalError.setTextContent("Tramitada");
               estado.appendChild(literalError);
               retorno.appendChild(estado);
               datosEspecificos.appendChild(retorno);
               doc.appendChild(datosEspecificos);
               return doc.getDocumentElement();
       }
```

7. Propostes de solució a mig termini

Generació de DatosEspecíficos amb les eines XML JAXB ja desenvolupades per scsp-pinbal. Còm fer ús dels esquemes.

Els objectes especifics de cada servei els trobam als projectes sota la carpeta scsp-schemas-xsd-main. Tots els subprojectes generen els objectes Java a partir dels esquemes xsd corresponents mitjançant el plugin Maven maven-jaxb2-plugin al fitxer pom.xml

Substitucio de la generació manual de DatosEspecificos

```
SCDHPAJUv3RespuestaDatosEspecificos datosEspecificosRespuesta

private Element createDatosEspecificos(
                      SolicitudTransmision solicitudTransmision,
                      int versioNum) throws ParserConfigurationException {

     return datosEspecificos2Element(SCDHPAJUv3PeticionDatosEspecificos datosEspecificosPeticion)

       }


@Override
protected Element datosEspecificos2Element(SCDHPAJUv3RespuestaDatosEspecificos datosEspecificosRespuesta){

        Element elementDatosEspecificos;
        try {
            XmlManager<SCDHPAJUv3PeticionDatosEspecificos> manager
                    = new XmlManager<SCDHPAJUv3PeticionDatosEspecificos>(SCDHPAJUv3PeticionDatosEspecificos.class);
            elementDatosEspecificos = manager.generateElement(datosEspecificosPeticion);
            return elementDatosEspecificos;
        } catch (JAXBException ex) {..}
        return null;
}
```

https://github.com/GovernIB/emissors