Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université AMO de Bouira
Faculté des Sciences et des Sciences Appliquées
Département d'Informatique
****

Correction : Introduction aux Architectures Orientées Services

Examen Niveau : **M1 – ISIL** Année : **2017-2018**

Exercice 1 : 6 points

1. Trois types de contrat sont à distinguer :
   a) Lié à la syntaxe du service (opération, messages d'entrée, messages de sortie, …)
   b) Lié à la sémantique du service (définition de règles et de contraintes d'usage, …)
   c) Lié à la qualité de service (temps de réponse attendu, procédures en cas de panne, temps de reprise après interruption,…).

2. Différence :
   a) Registry : annuaire listant tous les services (contrats) disponibles exposés.
   b) Repository : référentiel contenant tous les services exposés avec leurs codes, versions, etc.

3. Les exigences :
   a) Techniques : Maitrise de l'hétérogénéité, Communication, Redondance, etc.
   b) Métiers : Agilité, Intégration d'acteurs externes, etc.

Exercice 2 : 9 points

1. L'interface :

```
@WebService
@SOAPBinding(style=Style.RPC, use=Use.LITERAL,
                                parameterStyle=ParameterStyle.WRAPPED)
public interface TwitterAPI {
      @WebMethod
      public ArrayList<Tweet> searchTweets(String user, int nbr);
      @WebMethod
      public ArrayList<Tweet> filterTweets(String hashtag, int nbr);
      @WebMethod
      public void websiteAPI(String mode, String url);
}
```

2. Modification de l'opération :

```
@WebMethod(operationName="rechercherTweets")
@WebResult(name="ListeDesTweets")
public Song getRecommendedSong(String user, int nbr);
```

3. Changements :

```
@WebService(endpointInterface = "org.soa.ws.examen.TwitterAPI",
             serviceName="TwitterAPIService", portName="TwitterAPIPort")
public class TwitterAPIImpl implements TwitterAPI {

    //TODO

}
```

4. Client :

```
public class TwitterAPIClient {
      public static void main(String[] args) {
            TwitterAPIService service = new TwitterAPIService();
            TwitterAPI stub = service.getTwitterAPIPort();

            ArrayList<Tweet> list = stub.rechercherTweets("User", 5);
            for(int i = 0; i < list.size(); i++)
               System.out.println("Tweet N°  " + i + " : \n \t Contenu du
                       tweet n° " + i + " : " + list.get(i)+getContenu());
}
```

5. Classe Tweet :

```
@XmlRootElement(name="TweetClass")
@XmlType(propOrder={"user", "contenu", "hashtag"})
public class Tweet {
      private String user;
      private String contenu;
      private ArrayList<String> hashtag;

      public Tweet() {
      }

      //Constructeur init

      @XmlElement(name="UserTweet")
      public String getUser() {return this.user;}

      @XmlElement(name="ContenuTweet")
      public String getContenu() {return this.contenu;}

      @XmlTransient
      public ArrayList getHashtag() {return this.hashtag;}

      //setters

}
```

6. Modifications :

```
<portType name="TwitterAPI">
      <operation name="filterTweets">
           <input wsam:Action="http://examen.ws.soa.org/
               TwitterAPI/filterTweetsRequest"  message="tns:filterTweets"/>
      </operation>
</portType>
```

Exercice 3 : 5pts

1. Automatisation des processus métier dans SOA : Orchestration de services.

2. Rôle ESB : La communication et l'échange de messages entre les fournisseurs de services et les consommateurs. L'utilisation d'un service évite que les services aient besoin de connaître les autres services. Caractéristique : couplage lâche.

3. Code de sécurité :

```
@Resource
WebServiceContext wsContext;

...

public String invokedOperation(){

      MessageContexte messageContext = wsContext.getMessageContext( );

      Map<String, List<String>> httpHeaders = (Map)

           messageContext.get(MessageContext.HTTP_REQUEST_HEADERS) ;

      List<String> codeList = (List) httpHeaders.get("Code");

      if(! codeList.contains("CodeCorrect"))
          return "Le code que vous avez indiqué n'est pas correct" ;
      else
         //TODO invokedOperation traitements + return …
}
```