Web Services

InfoRep – TDM4

I. Conception d'architectures répartie

- **Conception :** Diagramme d'interaction entre entités distribuées
  - Sémantique des messages (type, contenu)
  - Protocole de communication (enchainements des messages)
  - Protocole de transport
- **Propriétés :** Transparent, Abstrait (connaissance interface uniquement), …
- **Organisation :** Client –Serveur / Serveurs coopérants / Objets partagés / flots de communication / code mobile / agents

II. Web Services

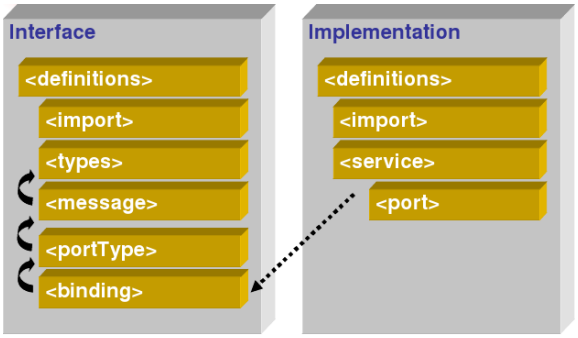
Fait transiter des données XML via HTTP

1. Vocabulaire

- **Annuaire UDDI :** Universal Description, Discovery and Integration
- **XML-RPC, SOAP (Simple Object Access Protocol) :** protocoles pour Web Services

2. WSDL : Web Service Description Language

- Décrit le service
- wsdl2java / java2wsdl



a. Approche bottom-top

- Développer une interface
- Développer l'implémentation de l'interface
- Développer le service web (war)
- Déployer le war
- Générer le stub client depuis WSDL
- Développer le client

3. Passage d'objets

Les objets doivent respecter les contraintes JavaBeans :

- Constructeur sans argument
- get et set pour chaque attribut privé
- Le client peut créer des objets par les classes regénérées (et ObjectFactory ?)

4. Implémentation

a. Développer une interface et une implémentation (annotation pour WSDL)

```
@WebService(name="HelloWorldWS") // PortType
@SOAPBinding(style=SOAPBinding.Style.RPC) // Binding
public interface HelloWorldWS {
	@WebMethod(action="sayHello") // Operation
	public String sayHello(@WebParam(name = "name") String name);
}

@WebService(endpointInterface="helloWebService.HelloWorldWS") // Port
public class HelloWorldWSImpl implements HelloWorldWS {
	public String sayHello(String name) {
		return "Hello " + name + " !";
	}
}
```

b. Description du servlet dans web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
	<servlet>
		<servlet-name>HelloWorldWS</servlet-name>
		<servlet-class>helloWebService.HelloWorldWSImpl</servlet-class>
	</servlet>
	<servlet-mapping>
		<servlet-name>HelloWorldWS</servlet-name>
		<url-pattern>/HelloWorldWS</url-pattern>
	</servlet-mapping>
</web-app>
```

c. Création d'un war

```
jar -cf HelloWorldWebService.war WEB-INF
```

d. Génération du stub client

```
$JBOSS_HOME/bin/wsconsume.sh http://localhost:8080/HelloWorldWebService/HelloWorldWS?wsdl
```

e. Création d'un client

```
public class Client {
	public static void main(String[] args) throws Exception {
		HelloWorldWS hello = (new
HelloWorldWSImplService()).getHelloWorldWSImplPort();
		System.out.println(hello.sayHello(args[0]));
	}
}
```