Faculté des Sciences de Gabès

Cours Sécurité des Services Orientés Web

Chapitre 4 : WSDL : Web Services Description Language

Faîçal Felhi

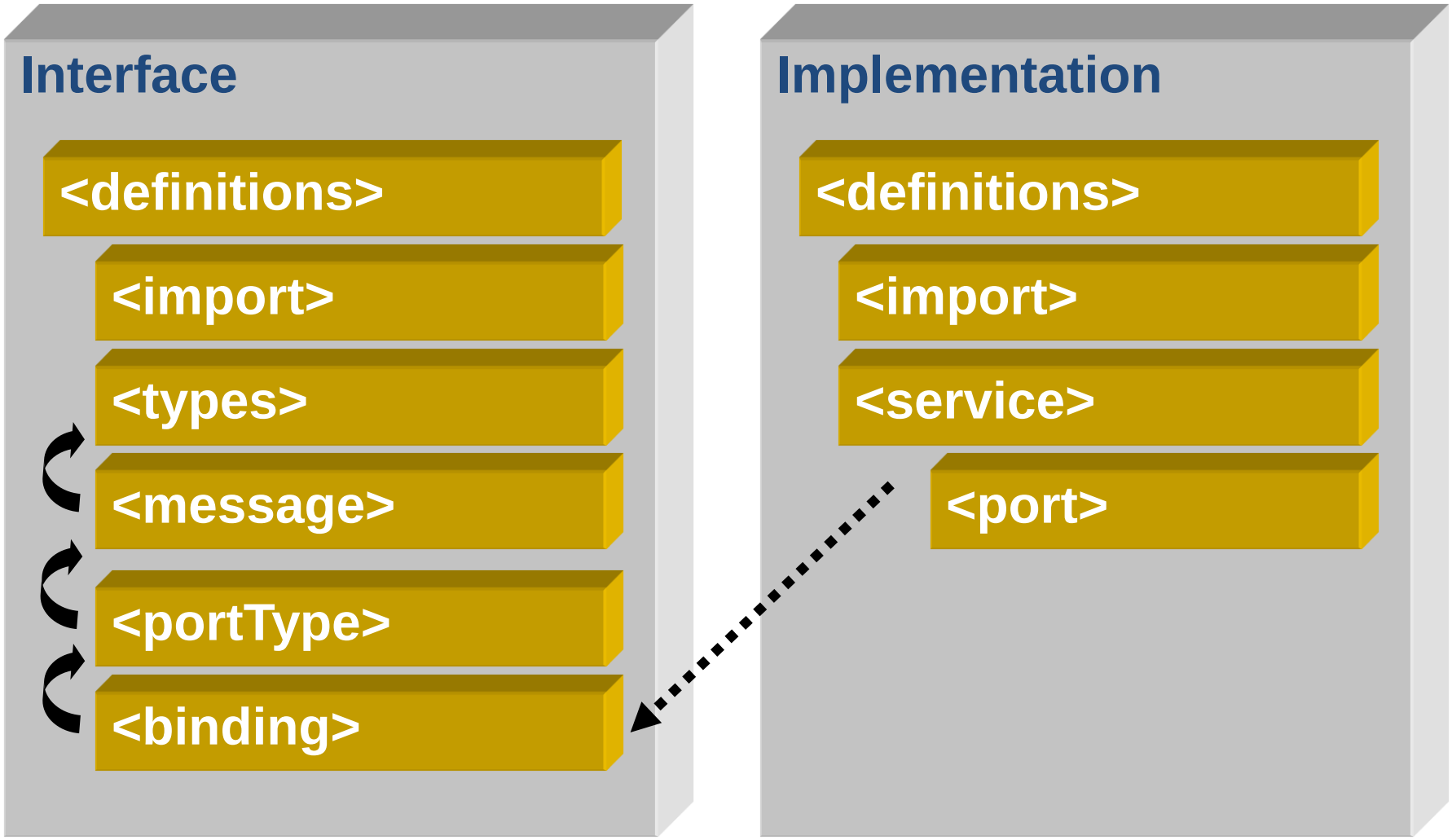
felhi_fayssal@yahoo.fr

WSDL

- Spécification (09/2000)
  - Ariba, IBM, Microsoft
  - TR W3C v1.1 (25/03/2001)
- Objectif
  - Décrire les services comme un ensemble d'opérations et de messages abstraits relié (*bind*) à des protocoles et des serveurs réseaux
- Grammaire XML (schema XML)
  - Modulaire (*import* d'autres documents WSDL et XSD)
- Séparation entre la partie abstraite et concrète

Faîçal Felhi - felhi_fayssal@yahoo.fr

WSDL


Interface
<definitions>
<import>
<types>
<message>
<portType>
<binding>
Implementation
<definitions>
<import>
<service>
<port>

Éléments d'une définition WSDL

- <types>
  - Contient les définitions de types utilisant un système de typage (comme XSD).
- <message>
  - Décrit les noms et types d'un ensemble de champs à transmettre
    - Paramètres d'une invocation, valeur du retour, …
- <porttype>
  - Décrit un ensemble d'opérations. Chaque opération a zéro ou un message en entrée, zéro ou plusieurs message de sortie ou de fautes
- <binding>
  - Spécifie une liaison d'un <porttype> à un protocole concret (SOAP1.1, HTTP1.1, MIME, …). Un <porttype> peut avoir plusieurs liaisons !
- <port>
  - Spécifie un point d'entrée (endpoint) comme la combinaison d'un <binding> et d'une adresse réseau.
- <service>
  - Une collection de points d'entrée (endpoint) relatifs.

Faîçal Felhi - felhi_fayssal@yahoo.fr

Élément <types>

- Contient les définition de types utilisant un système de typage (comme XSD).
- **Exemple**

```
<!-- type defs -->
<types>
  <xsd:schema targetNamespace="urn:xml-soap-address-demo"
                              xmlns:xsd="http://www.w3.org/1999/XMLSchema">
    <xsd:complexType name="phone">
      <xsd:element name="areaCode" type="xsd:int"/>
      <xsd:element name="exchange" type="xsd:string"/>
      <xsd:element name="number" type="xsd:string"/>
    </xsd:complexType>
    <xsd:complexType name="address">
      <xsd:element name="streetNum" type="xsd:int"/>
      <xsd:element name="streetName" type="xsd:string"/>
      <xsd:element name="city" type="xsd:string"/>
      <xsd:element name="state" type="xsd:string"/>
      <xsd:element name="zip" type="xsd:int"/>
      <xsd:element name="phoneNumber" type="typens:phone"/>
    </xsd:complexType>
  </xsd:schema>
</types>
```


Faîçal Felhi - felhi_fayssal@yahoo.fr

Élément <message>

- Décrit les noms et types d'un ensemble de champs à transmettre
  - Paramètres d'une invocation, valeur du retour, …
- Exemple

```
<!-- message declns -->
<message name="AddEntryRequest">
   <part name="name" type="xsd:string"/>
   <part name="address" type="typens:address"/>
</message>

<message name="GetAddressFromNameRequest">
   <part name="name" type="xsd:string"/>
</message>

<message name="GetAddressFromNameResponse">
   <part name="address" type="typens:address"/>
</message>
```


Faîçal Felhi - felhi_fayssal@yahoo.fr

Élément <porttype>

- Décrit un ensemble d'opérations.
- Plusieurs types d'opérations
  - **One-way**
    - Le point d'entrée reçoit un message (<input>).
  - **Request-response**
    - Le point d'entrée reçoit un message (<input>) et retourne un message corrélé (<output>) ou un ou plusieurs messages de faute (<fault>).
  - **Solicit-response**
    - Le point d'entrée envoie un message (<output>) et recoit un message corrélé (<input>) ou un ou plusieurs messages de faute (<fault>).
      - Binding HTTP : 2 requêtes HTTP par exemple
  - **Notification**
    - Le point d'entrée envoie un message de notification (<output>)
- Paramètres
  - Les champs des messages constituent les paramètres (in,out, inout) des opérations

Faîçal Felhi - felhi_fayssal@yahoo.fr

Élément <porttype>

- Exemple

```
<!-- port type declns -->
<portType name="AddressBook">

   <!– One way operation -->
   <operation name="addEntry">
     <input message="AddEntryRequest"/>
   </operation>

   <!– Request-Response operation -->
   <operation name="getAddressFromName">
      <input message="GetAddressFromNameRequest"/>
      <output message="GetAddressFromNameResponse"/>
   </operation>

</portType>
```


Faîçal Felhi - felhi_fayssal@yahoo.fr

Élément <binding>

- Exemple de binding sur SOAP et HTTP (HyperText Transfer Protocol)

```
<!-- binding declns -->
<binding name="AddressBookSOAPBinding" type="AddressBook">
 <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
 <operation name="addEntry">
  <soap:operation soapAction=""/>
  <input> <soap:body use="encoded" namespace="urn:AddressFetcher2"
                encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/> </input>
  <output> <soap:body use="encoded" namespace="urn:AddressFetcher2"
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/> </output>
 </operation>
 <operation name="getAddressFromName">
  <soap:operation soapAction=""/>
  <input> <soap:body use="encoded" namespace="urn:AddressFetcher2"
                encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/> </input>
  <output> <soap:body use="encoded" namespace="urn:AddressFetcher2"
                encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></output>
 </operation>
</binding>
```


Faîçal Felhi - felhi_fayssal@yahoo.fr

Élément <binding>

- Exemple de binding avec SOAP et SMTP (Simple Mail Transfer Protocol)

```
<definitions ...>
  <types>
    <schema targetNamespace="http://stockquote.com/stockquote.xsd"
                xmlns="http://www.w3.org/2000/10/XMLSchema">
      <element name="SubscribeToQuotes">
        <complexType><all><element name="tickerSymbol"
   type="string"/></all></complexType></element>
      <element name="SubscriptionHeader" type="uriReference"/>
    </schema>
  </types>
  <message name="SubscribeToQuotes">
    <part name="body" element="xsd1:SubscribeToQuotes"/>
    <part name="subscribeheader" element="xsd1:SubscriptionHeader"/>
  </message>
  <portType name="StockQuotePortType">
    <operation name="SubscribeToQuotes">
      <input message="tns:SubscribeToQuotes"/>
    </operation>
  </portType>
  ...
```


Faîçal Felhi - felhi_fayssal@yahoo.fr

Élément <binding>

```
…
<binding name="StockQuoteSoap" type="tns:StockQuotePortType">
  <soap:binding style="document" transport="http://stockquote.com/smtp"/>
  <operation name="SubscribeToQuotes">
    <input message="tns:SubscribeToQuotes">
      <soap:body parts="body" use="literal"/>
      <soap:header message="tns:SubscribeToQuotes" part="subscribeheader"
                   use="literal"/>
    </input>
  </operation>
</binding>
<service name="StockQuoteService">
  <port name="StockQuotePort" binding="tns:StockQuoteSoap">
    <soap:address location="mailto:subscribe@stockquote.com"/>
  </port>
</service>
</definitions>
```


Faîçal Felhi - felhi_fayssal@yahoo.fr

Élément <service>

- Une collection de points d'entrée (endpoint) relatifs
- Exemple :

```
<?xml version="1.0" ?>
<definitions name="urn:AddressFetcher"
                    targetNamespace="urn:AddressFetcher2"
                    xmlns:typens="urn:xml-soap-address-demo"
                    xmlns:xsd="http://www.w3.org/1999/XMLSchema"
                    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
                    xmlns="http://schemas.xmlsoap.org/wsdl/">
  …
  <!-- service decln -->
  <service name="AddressBookService">
    <port name="AddressBook" binding="AddressBookSOAPBinding">
      <soap:address location="http://www.mycomp.com/soap/servlet/rpcrouter"/>
    </port>
  </service>
</definitions>
```


Faîçal Felhi - felhi_fayssal@yahoo.fr

WSDL

- Ouvert - permet d'autres namespace et donc très extensible
- Possibilité d'importer d'autres schémas et WSDL
- Fournit une "recette" pour les services Web
- Fournit les détails de l'interface et de l'implémentation
- Permet la séparation des deux

Faîçal Felhi - felhi_fayssal@yahoo.fr

Outils

- Générateur WSDL à partir de déploiement SOAP ou EJB, …
- Générateur de proxy SOAP à partir de WSDL
- Toolkits (Wsdl2Java / Java2Wsdl, …)
  - Propriétaires (non normalisés)

Faîçal Felhi - felhi_fayssal@yahoo.fr

Génération WSDL