

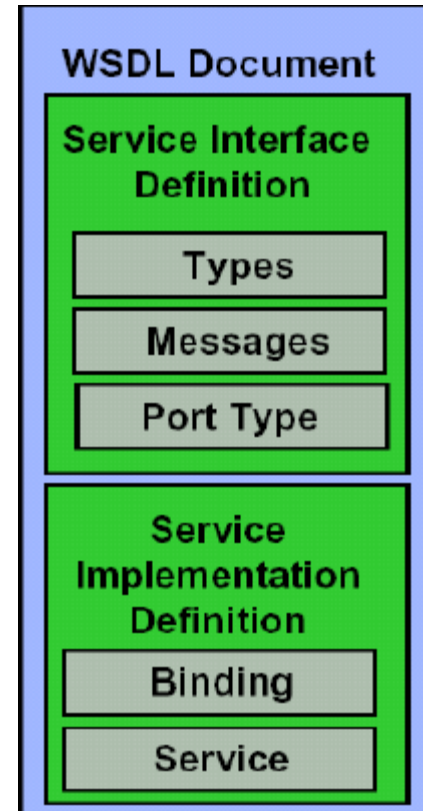
Le contrat d'un service web (WSDL: Web Service description Language)

C'est quoi le WSDL?

- WSDL= Web Service Description Language
- Langage de définition des contrats des Web
- Services Basé entièrement sur XML

Structure d'un fichier WSDL

- Format XML de description de services Web
- Une description WSDL:
 1. Décrit les opérations d'un service web (méthodes, types des paramètres)
 2. Décrit les aspects techniques d'implantation d'un service web (quel est le protocole utilisé, quel est le l'adresse du service)
Cette description sert à se connecter concrètement à un service web.



Structure du fichier WSDL

```
<definitions>  
  <types>  
    définition des types.....  
  </types>  
  <message>  
    définition des messages ....  
  </message>  
  <portType>  
    définition des ports (opérations et messages).....  
  </portType>  
  <binding>  
    Les protocoles de communication utilisés par le web-service  
  </binding>  
  <service>  
    Une adresse physique où accéder au service.  
  </service>  
</definitions>
```

Elément <definitions> (1/2)

- Il contient les namespaces faisant référence aux types des éléments utilisés dans le document.
 - o wsdl : <http://schemas.xmlsoap.org/wsdl/WSDL> namespace pour le framework WSDL.
 - o soap : <http://schemas.xmlsoap.org/wsdl/soap/WSDL> namespace pour le binding WSDL SOAP.
 - o http : <http://schemas.xmlsoap.org/wsdl/http/WSDL> namespace pour le binding WSDL HTTP GET & POST.
 - o mime : <http://schemas.xmlsoap.org/wsdl/mime/> WSDL namespace pour le binding WSDL MIME.
 - o soapenc : <http://schemas.xmlsoap.org/soap/encoding/> namespace d'encodage SOAP 1.1.
 - o soapenv : <http://schemas.xmlsoap.org/soap/envelope/> namespace de définition de l'enveloppe SOAP 1.1.
 - o xsi : <http://www.w3.org/2000/10/XMLSchema-instance> namespace d'instance XSD.
 - o xsd : <http://www.w3.org/2000/10/XMLSchema> namespace de schéma XSD .
 - o tns : "this namespace" (tns) référence le document courant par convention.

Elément <definitions> (2/2)

L'entête d'un document WSDL comporte des références à différents espaces de nommage (namespace) identifiés par une URL et un préfixe:

```
<?xml version="1.0" ?>
<definitions
  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/">
  ...
</definitions>
```

Elément <Types>

- Types : Permet la définition des types complexes. Il utilise généralement la grammaire XSD car cette dernière supporte un grand nombre de type de données.

```
<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>
```

Élément <message> (1/2)

- Les messages décrivent les noms et types d'un ensemble de champs à transmettre. Peuvent-êtré comparés aux paramètres d'un appel de procédure.
- Message(name, (part()))
 - Deux types de message Input et Output.
 - Peut êtré composé de plusieurs parties (Parts) correspondant aux paramètres passés aux fonctions.
- Part(nom, type)
 - Il peut êtré défini comme un type (simple ou complexe)
 - L'ordre des « parts » dépend de la signature de la méthode

Élément <message> (2/2)

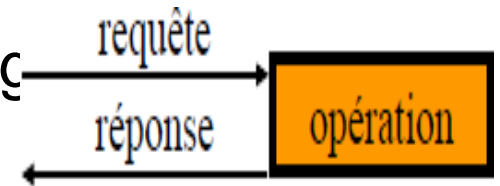
```
<message name="AddEntryRequest">  
  <part name="name" type="xsd:string"/>  
  <part name="address" type="typens:address"/>  
</message>
```

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

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

Élément <operation>

- Chaque opération est essentiellement définie par ses
- messages Trois types d'opération selon le séquençement et la nature des messages



Elément <portype> (1/2)

- Ensemble d'opérations
 - Groupes de messages à envoyer/recevoir
- portType (name, operation(name,(input msg, output msg)))
 - A chaque portType sont associées des *opérations, correspondant aux méthodes.*
 - Pour chaque méthode, on définit le message d'entrée et/ou de sortie

Élément <portype> (2/2)

```
<portType name="Annuaire">  
  <!-- One way operation -->  
  <operation name="ajouterEntree">  
    <input message="ajouterEntree"/>  
  </operation>  
  
  <!-- Request-Response operation -->  
  <operation name="chercherAdresse">  
    <input message="chercherAdresseAppel"/>  
    <output message="chercherAdresseReponse"/>  
  </operation>  
</portType>
```

Élément <binding> (1/3)

- Séparation entre la définition abstraite du service et la manière de la consommer
- Chaque <portType> peut avoir plusieurs <binding>
- C'est un élément qui définit les détails techniques nécessaire pour consommer le service:
 - Style de consommation des opérations
 - L'encodage des messages requête & réponse.
 - En général:
 - SOAP
 - HTTP

Elément <binding> (2/3)

Binding(name, type, (soap:binding, style, transport),(operation(soap:operation, soap:action (input(soap:body, encodingStyle, namespace, use), output(soap:body, encodingStyle, namespace, use)))

- *Nom de binding*
- *Le type permet de spécifier le portType(classe) auquel on fait*
- *référence soap :binding permet de déterminer le protocole*
- *d'invocation utilisé*

Style: permet de préciser si le message(body de soap) contient des

- *données ou un document xml*
- *Transport spécifie le protocole d'invocation.*
- *soap:action précise le contenu de l'entête du message d'invocation*
- *SOAP soap:body précise la forme des « messages parts » présents*
- *dans le corps du message SOAP.*

Use est utilisé pour spécifier la forme des données, un namespace (encoded) ou des paramètres (literal)

Élément <binding> (3/3)

```
<binding name="AnnuaireSOAPBinding" type="Annuaire">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="ajouterEntree">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="encoded"
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
      </input>
    </operation>
    <operation name="chercherAdresse">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="encoded"
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
      </input>
      <output>
        <soap:body use="encoded"
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
      </output>
    </operation>
  </binding>
```

Élément <service>

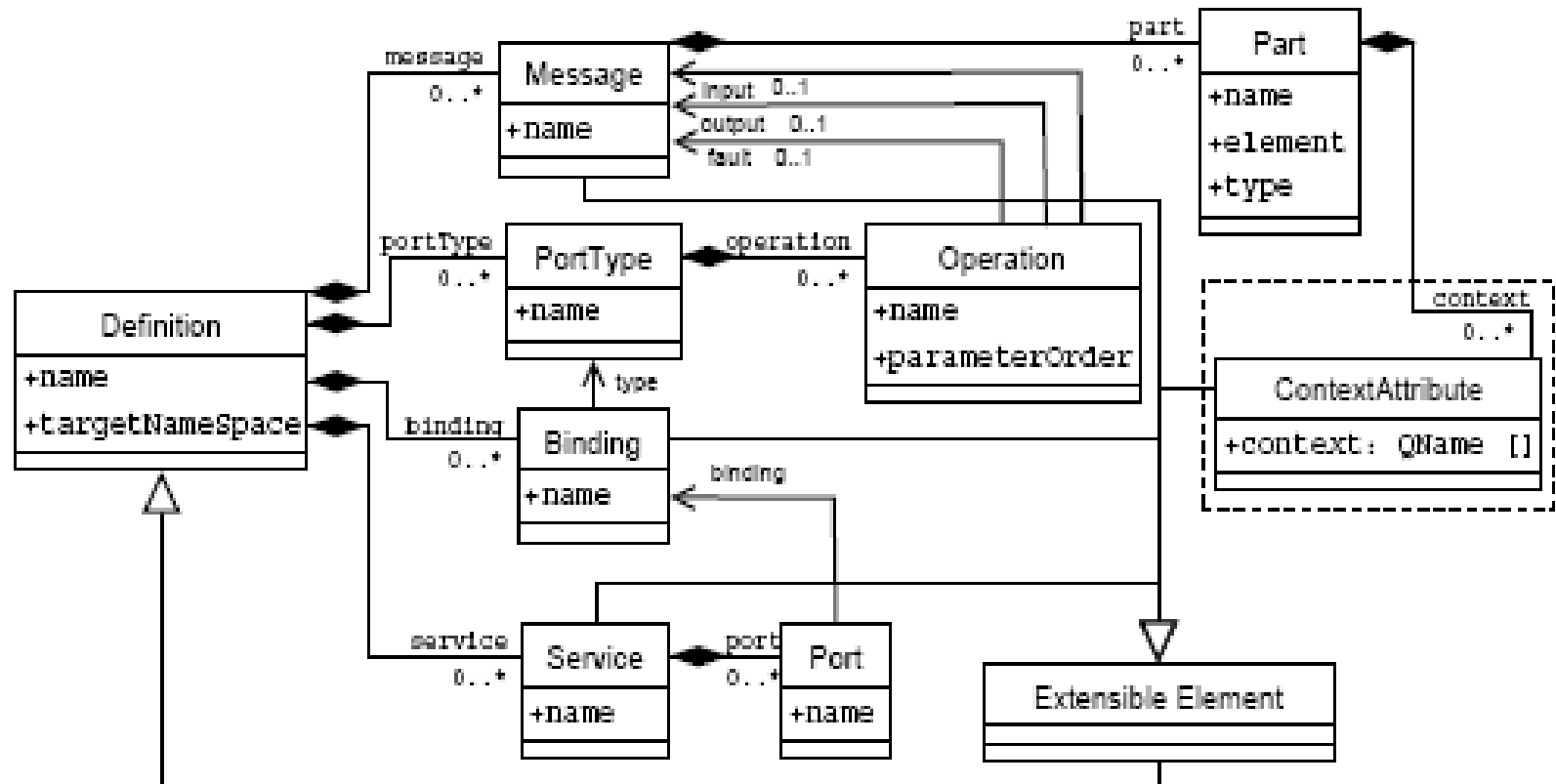
- Permet de spécifier l'url où envoyer les messages d'invocation.
- Service(name, port(name, binding, :address(location)))
 - Port spécifie l'adresse (URI) à laquelle un service peut être invoqué.
 - L'élément binding permet d'associer le nom du port au binding du fichier wsdl.
 - soap:address spécifie l'adresse du service

```
<!-- service decln -->  
<service name="serviceAnnuaire">  
  <port name="Annuaire" binding="AnnuaireSOAPBinding">  
    <soap:address location="http://www.monentreprise.com/annuaire"/>  
  </port>  
</service>
```


WSDL: structure

- Description de l'interface du service :
 - Types : description des données (de paramètres) utilisées dans les messages
 - Messages : description structurée des données échangées
 - Opérations : description abstraite des actions supportées par un service.
 - Type de port : ensemble d'opérations abstraites supportées par un ou plusieurs services Web
 - Binding : un protocole concret et une spécification de format de données pour un type de port particulier.

WSDL



Représentation types de données en XSD

IEP Data Type	XSD Data Type
INTEGER	int
BIGINT	long
DOUBLE	double
VARCHAR	string
DATE	date
TIMESTAMP	dateTime

Exercice

- Soit une application simple de gestion du compte dont l'interface java est la suivante :

```
import java.util.*;

public interface CompteInterface {

    public void depotDe(int montant);

    public boolean retraitDe(int montant);

    public int valeurDuSolde();
}
```

- Écrire le WSDL correspondant

