Université Abderrahmane Mira de Béjaia
Faculté des Sciences Exactes
Département d'Informatique

Le WSDL
(Cours 3)

Dr H. EL BOUHISSI

Septembre 2023

Objectifs du cours

1. Découvrir le contrat de description d'un Service Web
2. Comprendre les briques de base d'un document WSDL
3. Apprendre comment extraire les informations utiles à partir d'un document WSDL

Plan

1. Définition et Versions
2. Contenu d’un document WSDL
3. Exemples de documents WSDL

WSDL : Définition

WSDL signifie Web Service Description Language.

Un métalangage permettant de décrire en détail les services Web. Un service Web est un service mis à disposition des clients par un serveur via Internet (ou un autre réseau). Les services Web sont multiplateformes, c'est-à-dire qu'ils fonctionnent avec les systèmes et les applications les plus divers.

Pour qu'un client puisse s'informer sur les possibilités et les processus du service Web, un fichier WSDL est disponible sur le serveur. Les détails contenus dans le fichier permettent au client de savoir comment consulter le service Web.

WSDL : Définition

Standard du W3C :

Version 1.1 en 2001
Version 2.0 en 2007, encore peu supporté par les outils

Objectif : décrire l'interface publique d'un Web Service (contrat de service)

Grammaire dérivée d'XML

Service Web = ensemble de ports de connexions mettant à disposition des opérations qui reçoivent et envoient des messages

Regroupe les informations nécessaires pour interagir avec le service (fonctionnelles et techniques) :
•Les méthodes, les paramètres et valeurs retournées, le protocole de transport utilisé, la localisation du service
Document indispensable au déploiement de Services Web
•Publication et recherche de services au sein de l'annuaire se font via les documents WSDL
• Pour l'accès à un service particulier, un client se voit retourné l'URL du fichier WSDL décrivant l'implémentation du service

Eléments du WSDL

<definitions> : racine du document
<types> : Types de données sous format schéma XML

<Part>

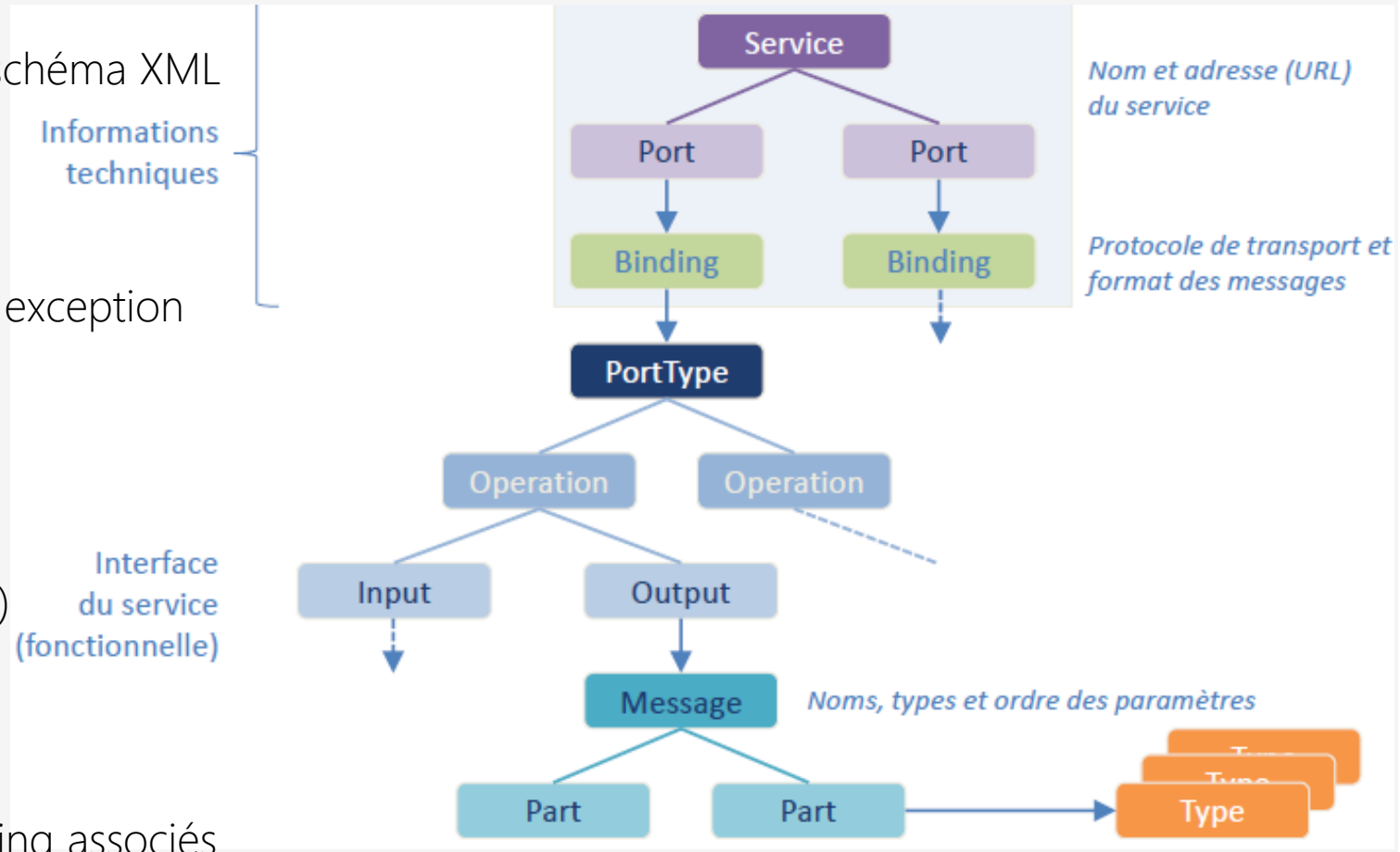
<Message> : paramètre, valeur de retour, exception
<Operation>

<portType> : ensemble d'opérations

<binding> : détail technique (protocoles…)

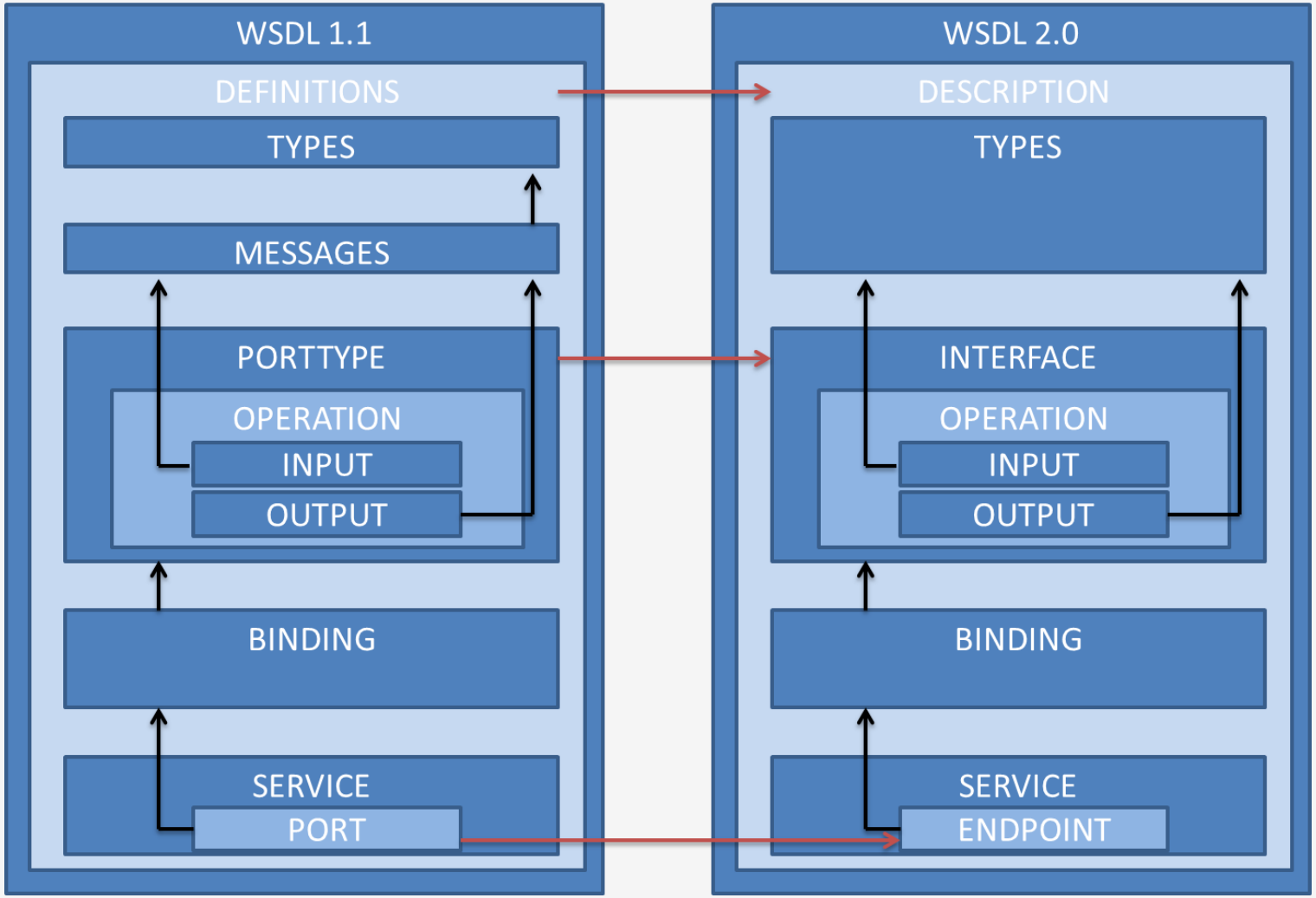
<port> : point d'accès

<service> : ensemble de port et leur binding associés



Description à 2 niveaux: Séparation entre la partie abstraite et concrète

WSDL : Version



Exemple : service Carnet d'adresses

3 opérations à mettre en œuvre :

- addPerson : Entrée (Person) + Sortie (booléen pour indiquer l'état de création)
- addPerson : Entrées (3 string : name, address et birthyear)
- getPersonByName : Entrée (string) + Sortie (Person)
- getPersons : Sortie (tableau d'objets Person)

Protocole SOAP : pour décrire les messages

Protocole HTTP : pour l'échange de messages

WSDL : l'élément types

2 variantes :

- Contient la définition des types de données (Schéma XSD, schéma XML…)
- Facultatif si les types sont simples (Integer, Boolean…)

- Importe un fichier Schéma XML contenant la définition des types
- Avantage : réutiliser des types et d'alléger le fichier WSDL

Exemple WSDL : définition des types

```
<definitions
      name="Notebook"
      targetNamespace="http://notebookwebservice.lisi.ensma.fr/"
      xmlns="http://schemas.xmlsoap.org/wsdl/"
      xmlns:tns="http://notebookwebservice.lisi.ensma.fr/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
      <types>
            <xsd:schema targetNamespace="http://notebookwebservice.lisi.ensma.fr/">
                  <xsd:complexType name="person">
                        <xsd:sequence>
                              <xsd:element name="address" type="xs:string" minOccurs="0"/>
                              <xsd:element name="birthyear" type="xs:string" minOccurs="0"/>
                              <xsd:element name="name" type="xs:string" minOccurs="0"/>
                        </xsd:sequence>
                  </xsd:complexType>
                  <xsd:complexType name="personArray" final="#all">
                        <xsd:sequence>
                              <xsd:element name="item" type="tns:person" minOccurs="0" maxOccurs="unbounded" nillable="true"/>
                        </xsd:sequence>
                  </xsd:complexType>
            </xsd:schema>
      </types>
      ...
</definitions>
```

Une personne est définie par une *adresse*, une *année de naissance* et un *nom*

Définition d'un type tableau de personne

WSDL : l'élément message

Décrit un message échangé par le service et est défini par les attributs :

- Name : nom du message

- 1 ou plusieurs éléments <part> : inputs/outputs ou exception d'une opération

Chaque élément <part> est défini par les attributs :
- name : nom du paramètre
- type : type de données du paramètre

WSDL : l'élément message

```
<definitions
      targetNamespace="http://notebookwebservice.lisi.ensma.fr/"
      name="Notebook"
      ...>
      <types>
            ...
      </types>
      <message name="addPersonWithComplexType">
            <part name="newPerson" type="tns:person"/>
      </message>
      <message name="addPersonWithComplexTypeResponse">
            <part name="addPersonWithComplexTypeResult" type="xsd:boolean"/>
      </message>
      <message name="addPersonWithSimpleType">
            <part name="name" type="xsd:string"/>
            <part name="address" type="xsd:string"/>
            <part name="birthyear" type="xsd:string"/>
      </message>
      <message name="getPerson">
            <part name="personName" type="xsd:string"/>
      </message>
      <message name="getPersonResponse">
            <part name="getPersonResult" type="tns:person"/>
      </message>
      <message name="getPersons"/>
      <message name="getPersonsResponse">
            <part name="getPersonsResult" type="tns:personArray"/>
      </message>
</definitions>
```

Message utilisé pour l'appel d'une opération avec une seule partie

Message utilisé pour le résultat d'une opération avec une seule partie

Message utilisé pour l'appel d'une opération avec trois parties

Une partie qui pointe sur un type défini par l'élément *<types>*

WSDL: les éléments portType et opération

Interface contenant les prototypes d'opérations fournies par le service et définie par les attributs :

- Name : nom du portType
- <opération> : 1 ou plusieurs décrivant chacun le prototype d'une méthode fournie par le service

Chaque élément <opération> est défini par les attributs :

- name : nom de l'opération
- <input message>/<output message>/ <fault> : référence à 1 message décrivant les paramètres d'E/ S et le message d'erreur de l'opération

Exemple WSDL : l'élément portType

```
<definitions
      targetNamespace="http://notebookwebservice.lisi.ensma.fr/"
      name="Notebook"
      ...>
      <types>
            ...
      </types>
      <message>
            ...
      </message>
      <portType name="Notebook">
            <operation name="addPerson">
                  <input message="tns:addPersonWithComplexType"/>
                  <output message="tns:addPersonWithComplexTypeResponse"/>
            </operation>
            <operation name="addPerson" parameterOrder="name address birthyear">
                  <input message="tns:addPersonWithSimpleType"/>
            </operation>
            <operation name="getPerson">
                  <input message="tns:getPerson"/>
                  <output message="tns:getPersonResponse"/>
            </operation>
            <operation name="getPersons">
                  <input message="tns:getPersons"/>
                  <output message="tns:getPersonsResponse"/>
            </operation>
      </portType>
</definitions>
```

L'opération *addPerson* est surchargée

Possibilité de fixer l'ordre des paramètres définis par cette opération

Types d'échanges assurés par une opération

One-way : Message input sans réponse

```
<operation name="addPerson" parameterOrder="name address birthyear">
    <input message="tns:addPersonWithSimpleType"/>
</operation>
```

Notification : Seul un message <output> est utilisé

```
<operation name="personStatus">
    <output message="trackingInformation"/>
</operation>
```

Request/Response : <input>, <output> et <fault>
Le service reçoit un message du client et répond à sa Requête

```
<operation name="addPerson">
    <input message="tns:addPersonWithComplexType"/>
    <output message="tns:addPersonWithComplexTypeResponse"/>
</operation>
```

Solicit - response : <input>, <output> et <fault>
Le client reçoit un message du service et répond au service

```
<operation name="clientQuery">
    <output message="bandWithRequest"/>
    <input message="bandwidthInfo" />
    <fault message="faultMessasge" />
</operation>
```

WSDL : l'élément binding

Décrit un portType du point de vue technique , les informations sur le protocole de transport utilisé ;
Protocole (SOAP 1.1 ou 1.2, HTTP GET & Post : ex, transfert d'images) utilisé pour manipuler un <portType>

Est défini par les attributs : name : nom du binding , type : portType concerné
Sa structure (éléments qu'ils contient) dépend du protocole utilisé
Plusieurs <binding> peuvent être définis pour appeler un portType de différentes manières

```
<definitions>
     ...
     <binding name="NamePortBinding" type="tns:portType">
          <!-- Décrit le protocole à utiliser -->
          <operation name="operation1">
                    <!-- Action du protocole sur l'opération -->
               <input>
                    <!-- Action du protocole sur les messages d'entrés (input) -->
               </input>
               <output>
                    <!-- Action du protocole sur les messages de sorties (ouput) -->
               </output>
               <fault>
                    <!-- Action du protocole sur les messages d'erreurs (fault) -->
               </fault>
          </operation>
     </binding>
     ...
</definitions>
```

Ces informations sont spécifiques au protocole utilisé

WSDL : Binding SOAP

A généralement la forme <soap:binding> et est défini par l'espace de noms :

http://schemas.xmlsoap.org/wsdl/soap/

Utilise les principales balises suivantes :

- <soap:binding>
- <soap:operation>
- <soap:body>, <soap:header>, <soap:headerfault>
- <soap:fault>

WSDL : les éléments service et port

Un élément <service> définit les ports (points d'accès) du service
Chaque <port> spécifie 1 adresse pour 1 binding donné.
Un élément <port> est défini par les attributs :

- name : nom du port
- binding : nom du binding (défini précédemment)

La structure de l'élément <port> est spécifique au protocole utilisé dans le binding
- <soap:address> : URI du port dans le cas d'un binding SOAP

```
<definitions ...>
     <!-- Définition de la partie Abstraite du WSDL -->

     <binding ...>
     </binding>
     <service name="Notebook">
          <port name="NoteBookPort" binding="tns:NoteBookPortBinding">
               <soap:address location="http://localhost:8080/NotebookWebService/notebook"/>
          </port>
     </service>
</definitions>
```

Le portType Notebook est accessible en SOAP/HTTP via cette URL

Un autre service : HelloWorld

2 opérations : 1) makeHello : Entrée (string) et Sortie (string) 2) simpleHello : Sortie (string)
Protocole SOAP : pour décrire les messages , et le Protocole HTTP : pour l'échange de messages

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<definitions name="HelloWorld"
     targetNamespace="http://helloworldwebservice.lisi.ensma.fr/"
     xmlns="http://schemas.xmlsoap.org/wsdl/"
     xmlns:tns="http://helloworldwebservice.lisi.ensma.fr/"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  <types/>
  <message name="makeHelloWorld">
    <part name="value" type="xsd:string"/>
  </message>
  <message name="makeHelloWorldResponse">
    <part name="helloWorldResult" type="xsd:string"/>
  </message>
  <message name="simpleHelloWorld"/>
  <message name="simpleHelloWorldResponse">
    <part name="helloWorldResult" type="xsd:string"/>
  </message>
  <portType name="HelloWorld">
    <operation name="makeHelloWorld">
      <input message="tns:makeHelloWorld"/>
      <output message="tns:makeHelloWorldResponse"/>
    </operation>
    <operation name="simpleHelloWorld">
      <input message="tns:simpleHelloWorld"/>
      <output message="tns:simpleHelloWorldResponse"/>
    </operation>
  </portType>
```

```
<?xml version="1.0" encoding="
<definitions name="AktienKurs"
  targetNamespace="http://loca
  xmlns:xsd="http://schemas.xmlsoap.or
  xmlns="http://schemas.xmlsoap.org/wsd
  <service name="AktienKurs">
    <port name="AktienSoapPort" binding
      <soap:address location="http://loc
    </port>
    <message name="Aktie.HoleWert">
      <part name="body" element="xsd:Tra
    </message>
    …
  </service>
</definitions>
```

WSDL

Un autre service : HelloWorld

```
<binding name="HelloWorldPortBinding" type="tns:HelloWorld">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc"/>
  <operation name="makeHelloWorld">
    <soap:operation soapAction=""/>
    <input>
      <soap:body use="literal" namespace="http://helloworldwebservice.lisi.ensma.fr/"/>
    </input>
    <output>
      <soap:body use="literal" namespace="http://helloworldwebservice.lisi.ensma.fr/"/>
    </output>
  </operation>
  <operation name="simpleHelloWorld">
    <soap:operation soapAction=""/>
    <input>
      <soap:body use="literal" namespace="http://helloworldwebservice.lisi.ensma.fr/"/>
    </input>
    <output>
      <soap:body use="literal" namespace="http://helloworldwebservice.lisi.ensma.fr/"/>
    </output>
  </operation>
</binding>
<service name="HelloWorld">
  <port name="HelloWorldPort" binding="tns:HelloWorldPortBinding">
    <soap:address location="TODO"/>
  </port>
</service>
</definitions>
```

Exercice

Annoter ce WSDL de vos commentaires :à quoi il sert, comment s'en servir d'un point de vue client, quelles données vont circuler quand on l'utilise, etc.

Cliquez ici pour visualiser le Fichier WSDL

Solution

Ce service web offre une unique opération : sur envoi d'un message de type GetLastTradePrice, l'opération nommée GetLastTradePrice va s'exécuter. L'URL pour déclencher cette opération depuis un navigateur web devrait :

http://example.com/stockquote/method=GetLastPrice?tickerSymbol=MonEntreprisePreferee

Le message sera véhiculé par une enveloppe SOAP (on a défini un soap binding). Le type de donnée en entrée est un tickerSymbol prenant une valeur de type chaine de caractères. En réponse, on obtient un price que l'on peut interpréter comme un flottant.

Outils pour manipuler des documents WSDL

Edition

Notepad++ (texte : XML)
Eclipse JavaEE
Netbeans
Visual Studio
Environnements de développement de SW

Validation

www.validwsdl.com

Test

SOAPUI

En résumé.....

En résumé WSDL c'est un contrat entre un client et un serveur qui fait état :

- Des spécifications d'interfaces qui décrivent toutes les méthodes publiques,
- Des spécifications relatives aux types de donnée de messages mis en œuvre dans les questions-réponses.
- Des informations liées au protocole de transport utilisé.
- Des informations d'adresse permettant de localiser le service décrit.

En un mot, WSDL définit le contrat existant entre un client et un serveur sans dépendance particulière pour une plateforme ou un langage.