



WSDL

Module SOA
A.U 2019-2020



Objectifs

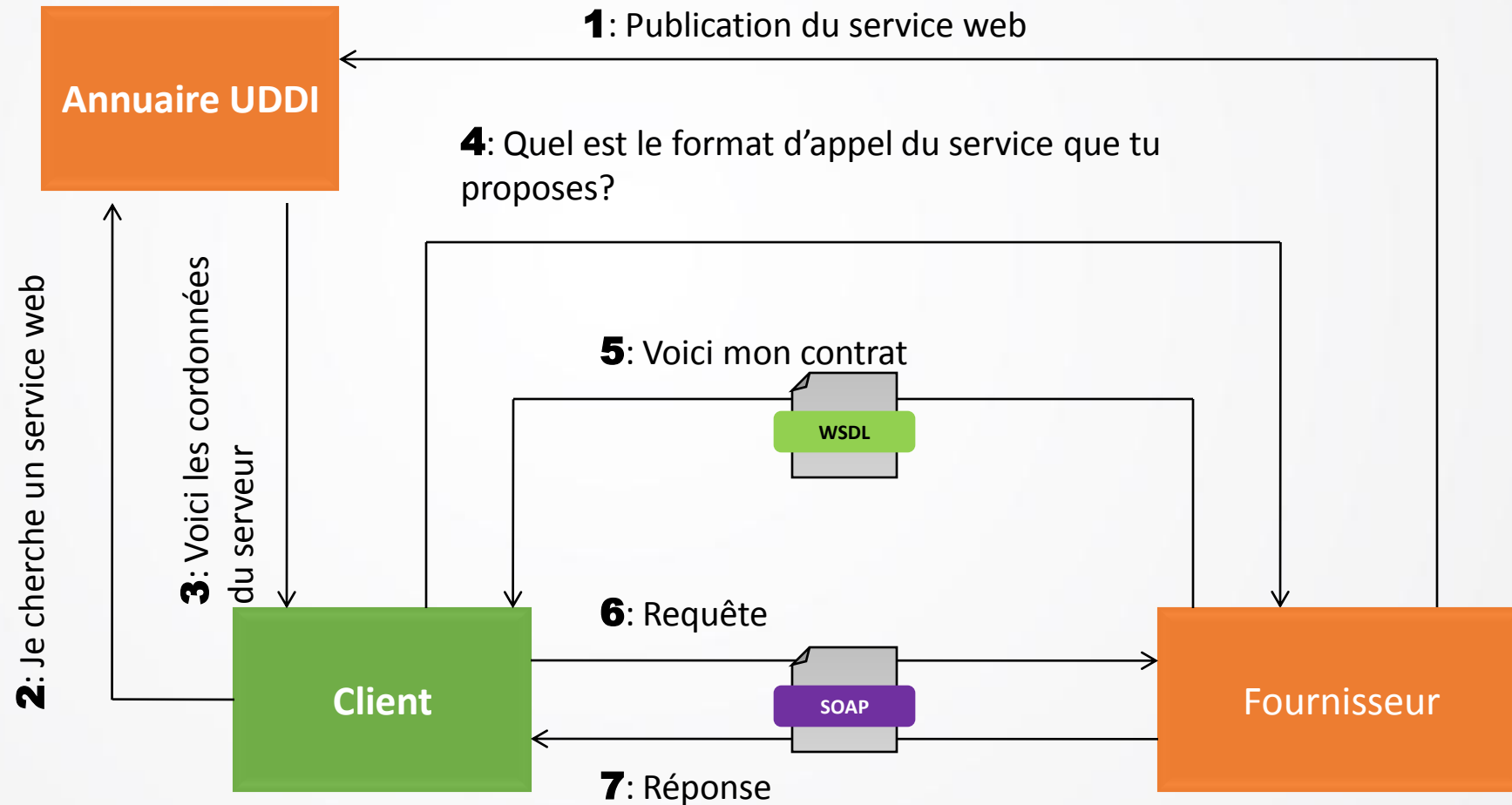


- Comprendre la structure et les différents éléments d'un document WSDL



Introduction

Un service web en action





Introduction



Afin de mettre en œuvre un service web, trois composants sont nécessaires :

- Un langage pour décrire le service web: WSDL
- Un protocole de communication pour écrire les messages échangés entre le consommateur et le fournisseur: SOAP
- Un protocole de transport afin de faire circuler les informations sur Internet



Présentation de WSDL



- **Web Service Description Language**
- Un langage de description des services web
- Une spécification W3C
- Basé sur XML
- Une description de type « boîte noire » du service web:
 - Quelles sont les opérations disponibles, leurs entrées, leurs sorties, leurs types?
 - Comment y accéder : adresse(s), protocole(s) utilisé(s),...
 - Quel est le format des échanges : les messages (requête/réponse)

Structure d'un contrat WSDL

Un document WSDL se structure en **2 parties** :

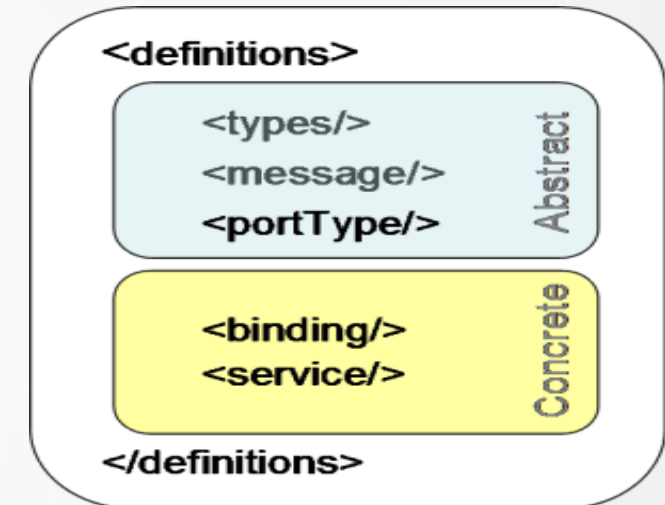
- **Une description concrète**

- Définition du protocole d'accès et de l'URI à partir de laquelle on peut accéder au service web

- **Une description abstraite**

- nom des opérations, paramètres d'entrée, de sortie, structure des messages

Contrat WSDL





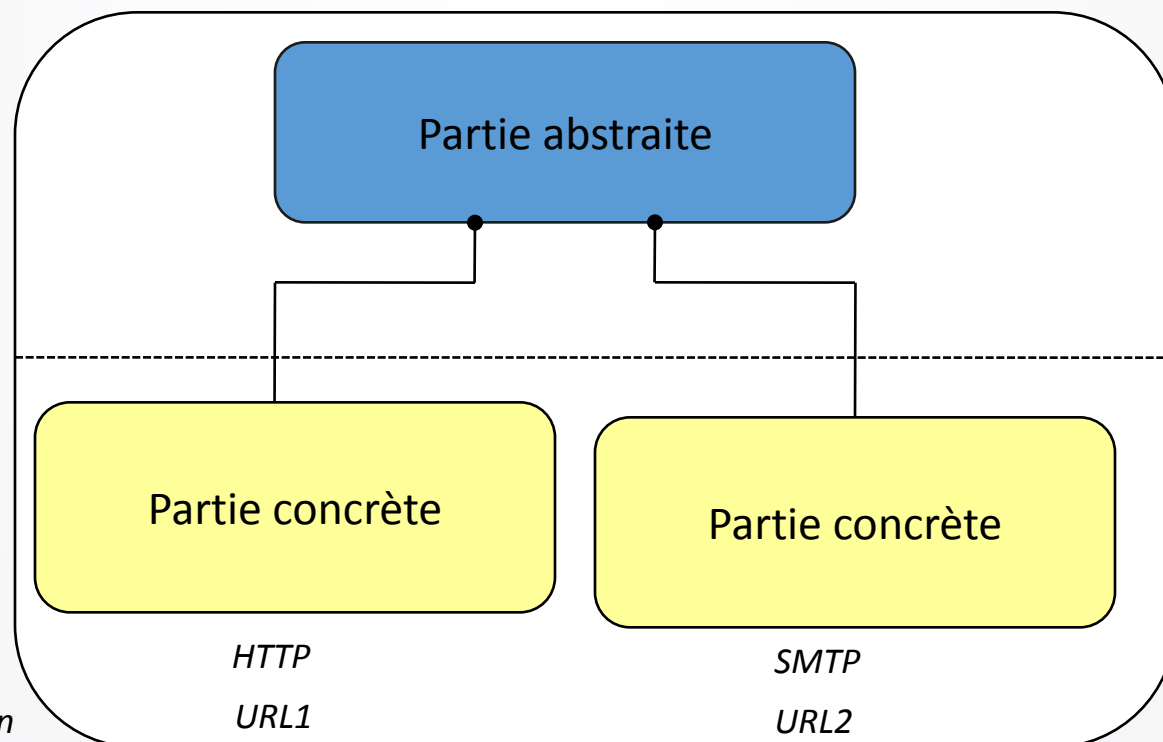
Structure d'un contrat WSDL

+ Avantages d'une structuration en 2 parties :

On peut associer la description abstraite avec plusieurs descriptions concrètes.

→ Réutilisabilité de la description abstraite

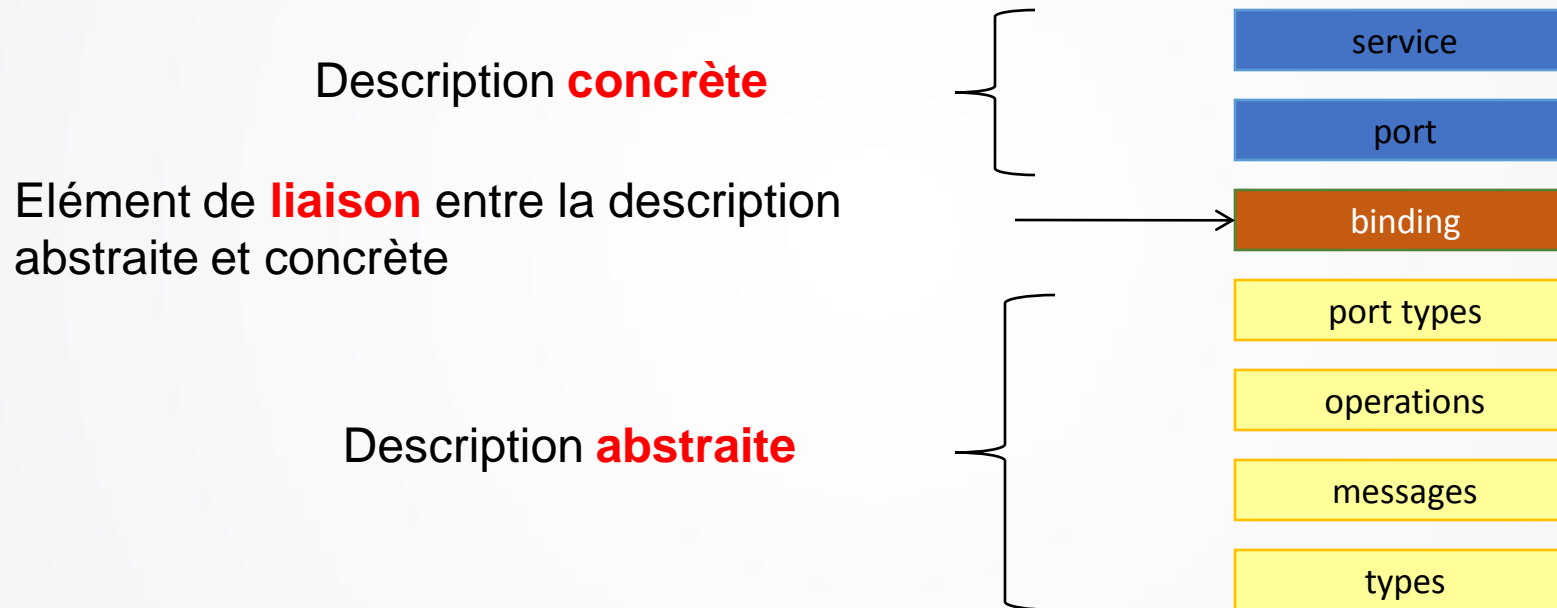
Contrat WSDL



Deux parties concrètes sont associées avec la même partie abstraite.

Les éléments WSDL

- Le document WSDL contient 7 éléments :

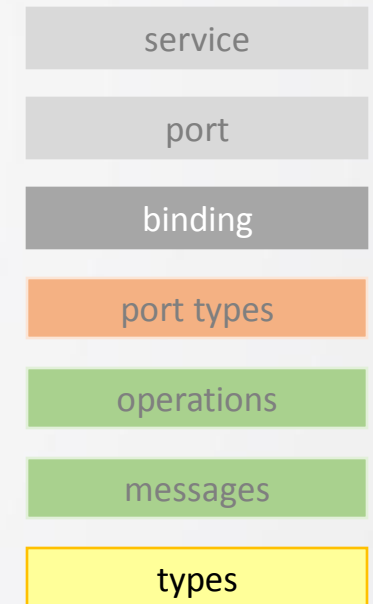




Les éléments WSDL



- Partie abstraite
- **Types**
 - Contient la définition des types de données à transmettre
 - Exprimé en XSD



Les éléments WSDL

■ Partie abstraite

• Types

Exemple:

```
<wsdl:types>
<s:schema elementFormDefault="qualified" targetNamespace="http://www.webservice.net">
  <s:element name="Translate">
    <s:complexType>
      <s:sequence>
        <s:element name="LanguageMode" type="tns:Language"/>
        <s:element name="Text" type="s:string"/>
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:simpleType name="Language">
    <s:restriction base="s:string">
      <s:enumeration value="EnglishTOFrench"/>
      <s:enumeration value="EnglishTOGerman"/>
      <s:enumeration value="EnglishTOItalian"/>
    </s:restriction>
  </s:simpleType>
  <s:element name="TranslateResponse">
    <s:complexType>
      <s:sequence>
        <s:element name="TranslateResult" type="s:string"/>
      </s:sequence>
    </s:complexType>
  </s:element>
</s:schema>
</wsdl:types>
```

Translate

LanguageMode: Language
Text: String

TranslateResponse

TranslateResult: String

Les éléments WSDL

■ Partie abstraite

• Messages

- Contient la description des messages échangés avec le service web
 - Paramètres d'entrée des opérations
 - Paramètres de sortie

```
<wsdl:message name="TranslateInput">
  <wsdl:part name="parameters" element="tns:Translate"/>
</wsdl:message>
<wsdl:message name="TranslateOutput">
  <wsdl:part name="parameters" element="tns:TranslateResponse"/>
</wsdl:message>
```

message: **TranslateInput**

Translate

LangageMode: Langage
Text: String

message: **TranslateOutput**

TranslateResponse

TranslateResult: String

service

port

binding

port types

operations

messages

types

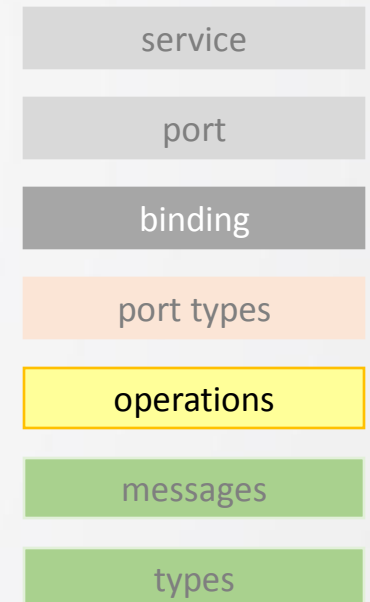
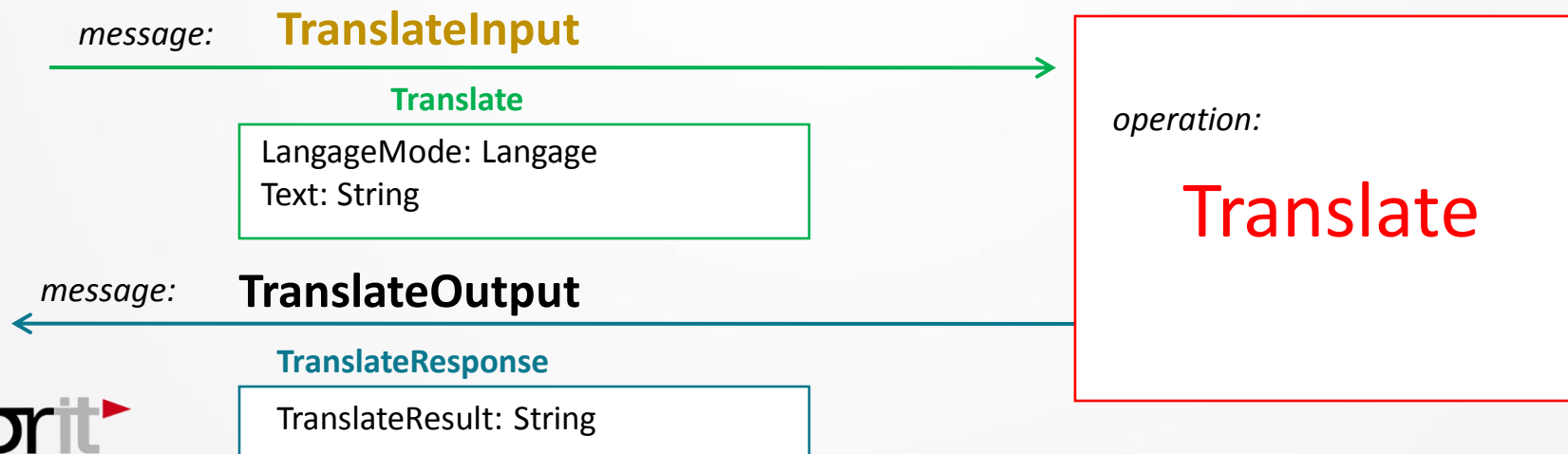
Les éléments WSDL

■ Partie abstraite

• Operations

- Une opération est comparable à une méthode en Java
 - Est identifiée par un nom
 - Contient ou non une ou plusieurs entrée(s)
 - Contient ou non une ou plusieurs sorties

```
<wsdl:operation name="Translate">  
  <wsdl:input message="tns:TranslateInput"/>  
  <wsdl:output message="tns:TranslateOutput"/>  
</wsdl:operation>
```





Les éléments WSDL



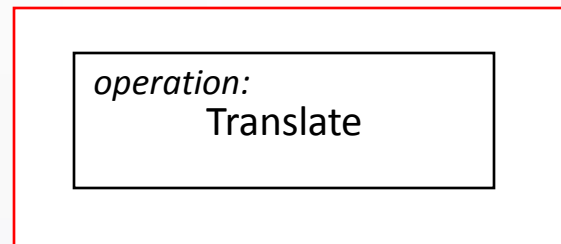
■ Partie abstraite

• Porttype

- Un porttype est comparable à une interface en Java
 - Est identifiée par un nom
 - Contient un ensemble d'opérations

```
<wsdl:portType name="TranslateServiceSoap">  
  <wsdl:operation name="Translate">  
    <wsdl:input message="tns:TranslateInput"/>  
    <wsdl:output message="tns:TranslateOutput"/>  
  </wsdl:operation>  
</wsdl:portType>
```

porttype: **TranslateServiceSoap**



service

port

binding

port types

operations

messages

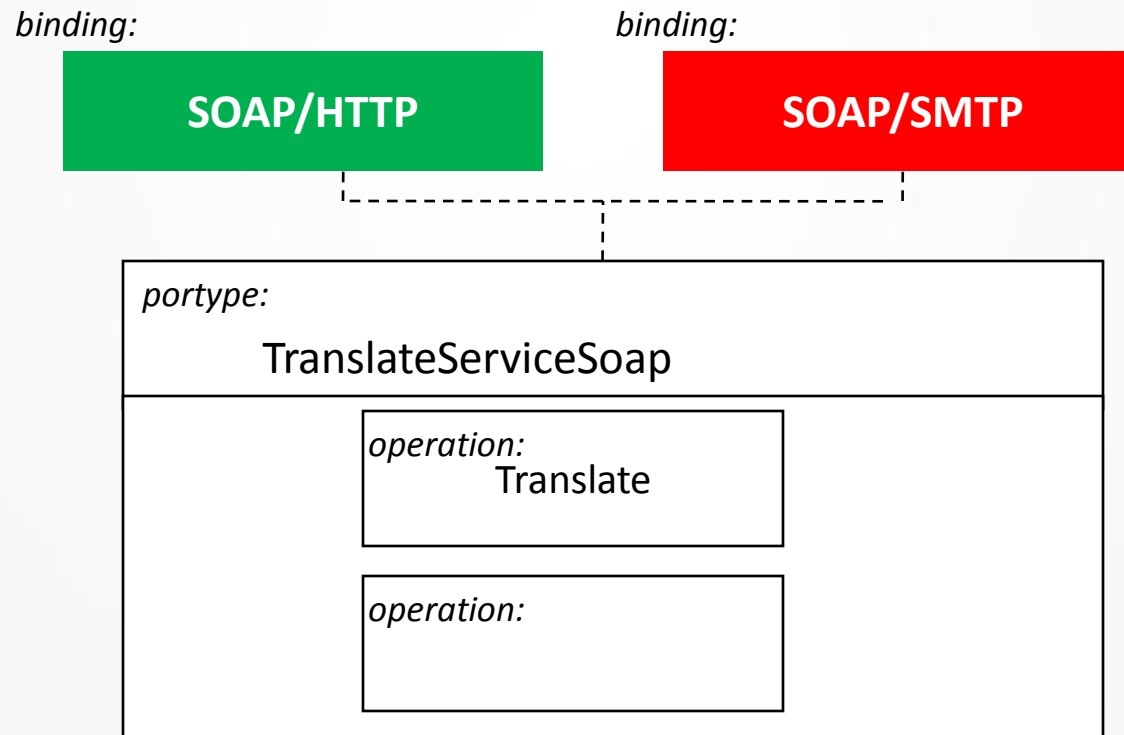
types

Les éléments WSDL

■ Partie concrète

• binding

- Permet de définir pour un porttype
 - Le format des messages échangés
 - Le protocole de transport (HTTP, SMTP, FTP, ...)
- On peut associer plusieurs binding à un même porttype



service

port

binding

port types

operations

messages

types

Les éléments WSDL

■ Partie concrète

- **binding**

```
<wsdl:binding name="TranslateServiceBinding" type="tns:TranslateServiceSoap">  
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>  
  <wsdl:operation name="Translate">  
    <soap:operation soapAction="http://www.webservicex.net/Translate" style="document"/>  
    <wsdl:input>  
      <soap:body use="literal"/>  
    </wsdl:input>  
    <wsdl:output>  
      <soap:body use="literal"/>  
    </wsdl:output>  
  </wsdl:operation>  
</wsdl:binding>
```

Protocole de transport des messages SOAP

Format des messages échangés

Les éléments WSDL

■ Partie concrète

- **binding**

Deux bindings pour un même porttype

```
<wsdl:binding name="TranslateServiceBinding1" type="tns:TranslateServiceSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="Translate">
    ...
  </wsdl:operation>
</wsdl:binding>

<wsdl:binding name=" TranslateServiceBinding2" type="tns:TranslateServiceSoap"/>
  <soap:binding transport="http://stockquote.com/smtp" />
  <wsdl:operation name="Translate">
    ...
  </wsdl:operation>
</binding>
```

Un 1^{er} binding qui définit le protocole HTTP pour le transport des messages SOAP

Un 2^{ème} binding qui définit le protocole SMTP pour le transport des messages SOAP



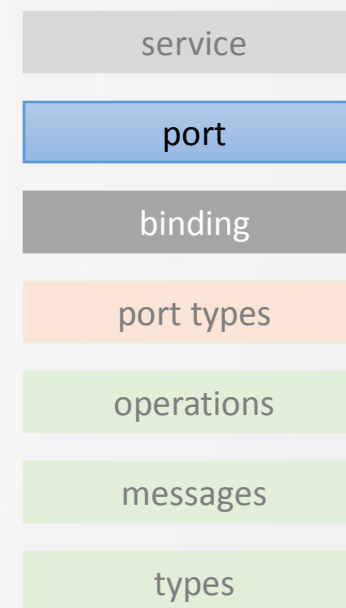
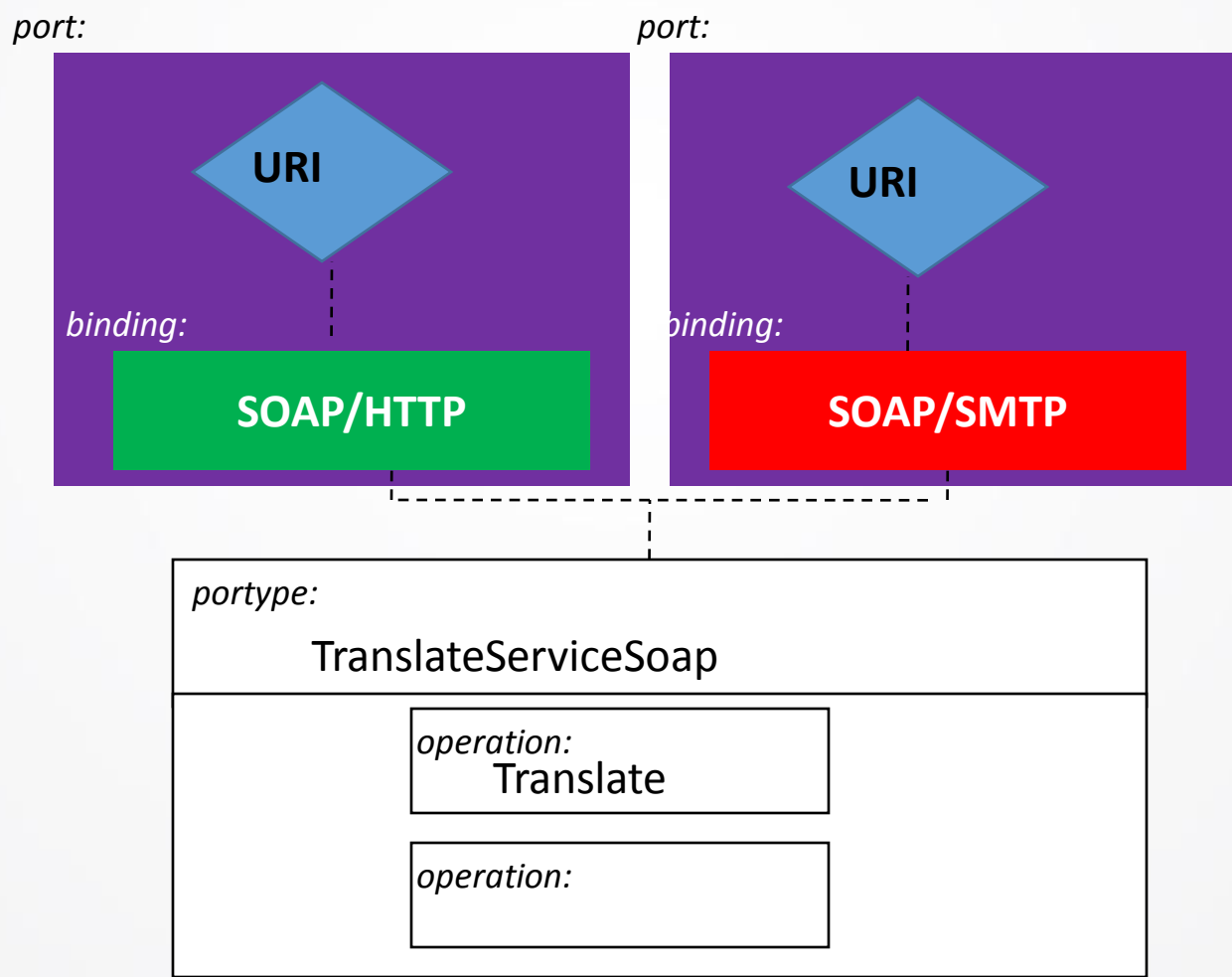
Les éléments WSDL



▪ Partie concrète

• Port

Permet de spécifier une adresse pour un binding donné





Les éléments WSDL



■ Partie concrète

- **service**

Contient une collection de ports.

```
<wsdl:service name="TranslateService">  
  <wsdl:port name="TranslateServiceSoap1" binding="tns:TranslateServiceBinding1">  
    <soap:address location="http://www.webservicex.net/TranslateService.asmx"/>  
  </wsdl:port>  
  <wsdl:port name="TranslateServiceSoap2" binding="tns:TranslateServiceBinding2">  
    <soap:address location="mailto:subscribe@translate.com"/>  
  </wsdl:port>  
</wsdl:service>
```

service

port

binding

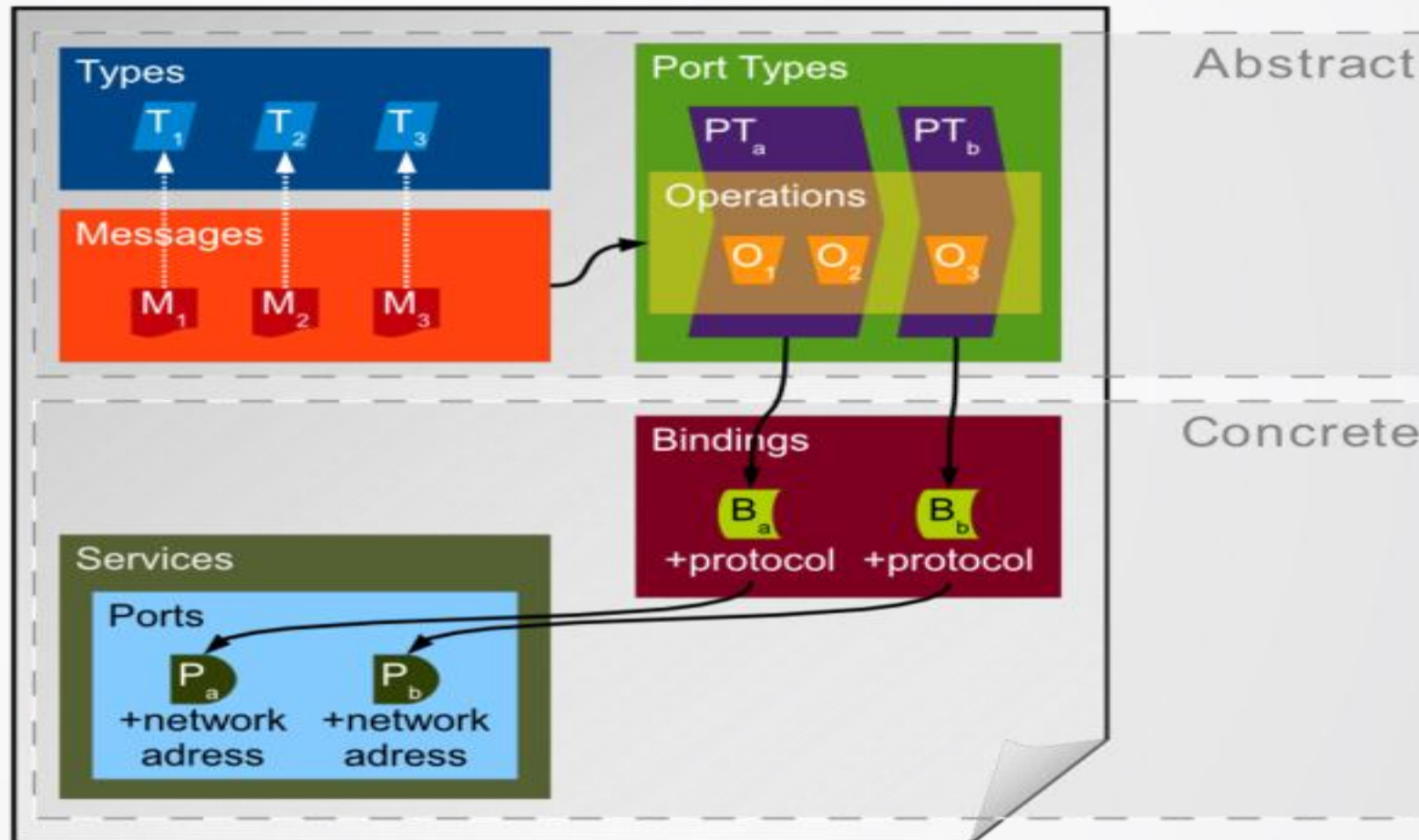
port types

operations

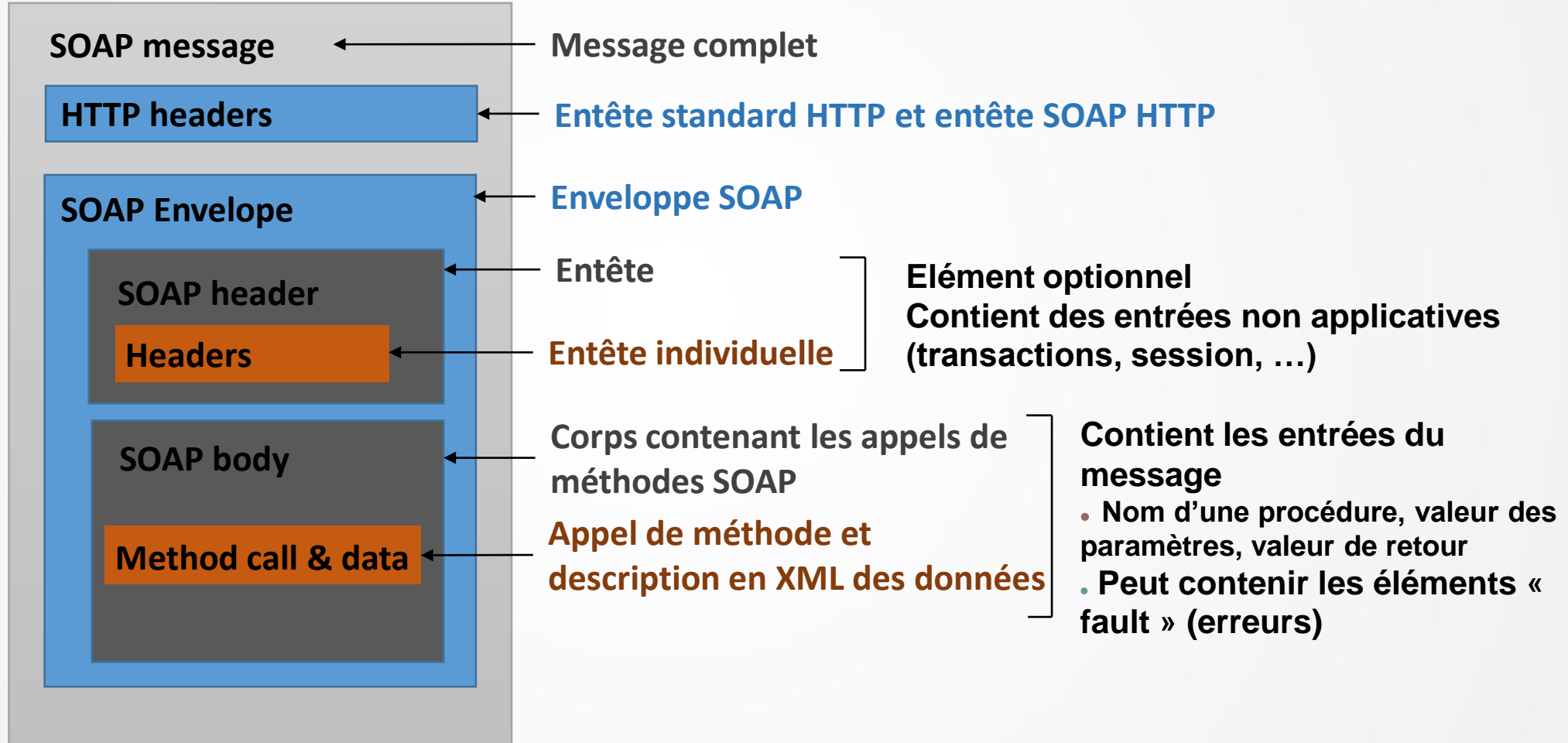
messages

types

Les éléments WSDL



Les messages SOAP (Simple Object Access Protocol)





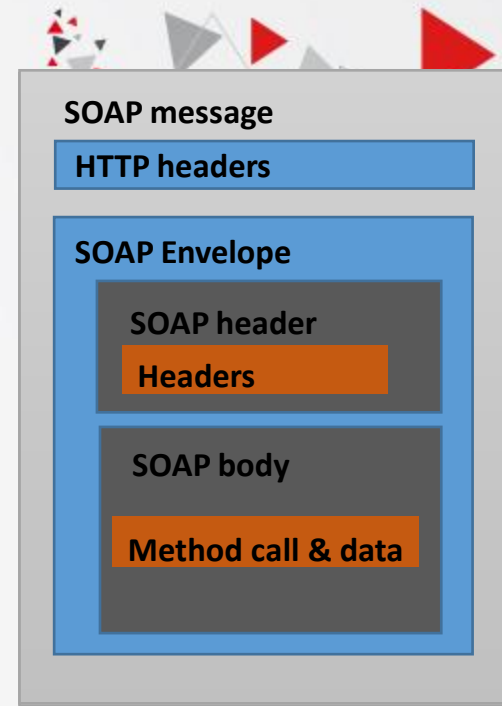
L'enveloppe SOAP

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope">
  <SOAP-ENV:Body>
    <!-- Contenu de la requête-->
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Message SOAP d'une requête

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope">
  <SOAP-ENV:Body>
    <!-- Contenu de la réponse -->
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Message SOAP d'une réponse



Les messages SOAP

Requête vers le service HelloWorld

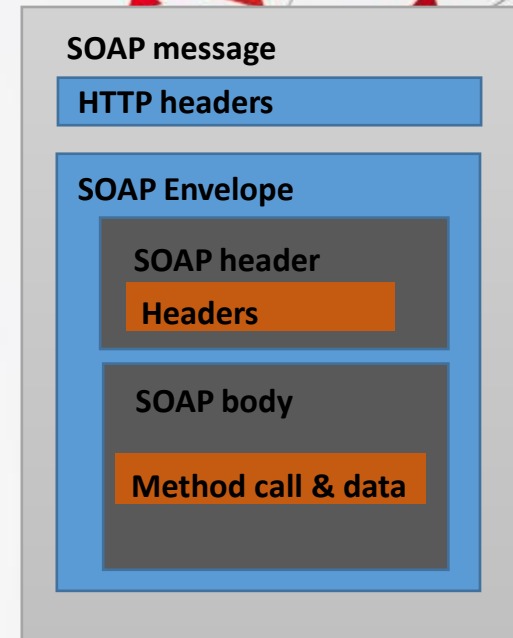
Exemple: Appeler les opérations du service HelloWorld

```
<soap-env:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:hel="http://helloworldservice... .com/">
  <soapenv:Header/>
  <soapenv:Body>
    <hel:makeHelloWorld>
      <value> toto </value>
    </hel:makeHelloWorld>
  </soapenv:Body>
</soapenv:Envelope>
```

**Message SOAP pour appeler
l'opération *makeHelloWorld*
contenant un paramètre *value***

```
<soap-env:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:hel="http://helloworldservice... .com/">
  <soapenv:Header/>
  <soapenv:Body>
    <hel:simpleHelloWorld>
  </hel:simpleHelloWorld>
  </soapenv:Body>
</soapenv:Envelope>
```

**Message SOAP pour appeler
l'opération *simpleHelloWorld*
ne contenant pas de
paramètres**



Les messages SOAP

Reponse vers le service HelloWorld

Exemple (suite) : Messages retour de l'appel des opérations du service HelloWorld

```
<soap-env:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope">
  <soapenv:Body>
    <ns2:makeHelloWorldResponse xmlns:ns2="http://helloworldwebservice.... .com/">
      <helloWorldResult>Hello World toto </helloWorldResult>
    </ns2:makeHelloWorldResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<soap-env:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope">
  <soapenv:Body>
    <ns2:simpleHelloWorldResponse xmlns:ns2="http://helloworldwebservice.... .com/">
      <helloWorldResult>Hello World Everybody </helloWorldResult>
    </ns2:simpleHelloWorldResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Les messages SOAP



Exemple : corps de messages SOAP pour appeler des opérations du service Web Notebook

```
<soap-env:Envelope
xmlns:soapenv=http://schemas.xmlsoap.org/soap/envelope
xmlns: not="http://notebookwebservice.... .com/">
  <soapenv:Header/>

  <soapenv:Body>
    <not:addPersonWithComplexType>
      <newperson>
        <name>toto</name>
        <adress>France</adress>
        <birthyear>1980</birthyear>
      </newperson>
    </not:addPersonWithComplexType>
  </soapenv:Body>

</soapenv:Envelope>
```

```
<soap-env:Envelope
xmlns:soapenv=http://schemas.xmlsoap.org/soap/envelope
xmlns: not="http://notebookwebservice.... .com/">
  <soapenv:Header/>

  <soapenv:Body>
    <not:addPersonWithSimpleType>
      <name>toto</name>
      <adress>France</adress>
      <birthyear>1980</birthyear>
    </not:addPersonWithSimpleType>
  </soapenv:Body>

</soapenv:Envelope>
```


Les messages SOAP



Exemple (suite) : corps de messages SOAP pour le résultat des opérations du service Web Notebook

```
<soap-env:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope">
  <soapenv:Body>
    <ns2:addPersonWithComplexTypeResponse xmlns:ns2="http://notebookwebservice.... .com/">
      <addPersonWithComplexTypeResult> true </addPersonWithComplexTypeResult>
    </ns2: addPersonWithComplexTypeResponse >
  </soapenv:Body>
</soapenv:Envelope>
```

```
<soap-env:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope">
  <soapenv:Body>
    <ns2:addPersonWithSimpleTypeResponse xmlns:ns2="http://notebookwebservice.... .com/">
      <addPersonWithSimpleTypeResult> true </addPersonWithSimpleTypeResult>
    </ns2: addPersonWithSimpleTypeResponse >
  </soapenv:Body>
</soapenv:Envelope>
```

Les paramètres de sorties suivent la même convention que les paramètres d'entrées



Références



- <http://www.w3.org/TR/wsdl>
- <http://www.webservicex.net/ws/WSDetails.aspx?CATID=12&WSID=63>
- <http://www.slideshare.net/baronm/wsdl-dcrire-et-configurer-4677823>