



Les services Web

WSDL : Web Services
Description Language

Éléments d'une description de service

■ Une description de service définit :

- ❑ Quelles sont les opérations disponibles, leurs entrées, leur sorties, leur types
- ❑ Comment y accéder : adresse(s), protocole(s) utilisé(s),...
- ❑ Quel est le format des échanges : les messages (requête /réponse)

■ Une description de service

- ❑ ne décrit pas de façon sémantique ce que fait un service
- ❑ Modélise/structure les informations définies ci-avant dans un ou plusieurs documents WSDL

Document WSDL

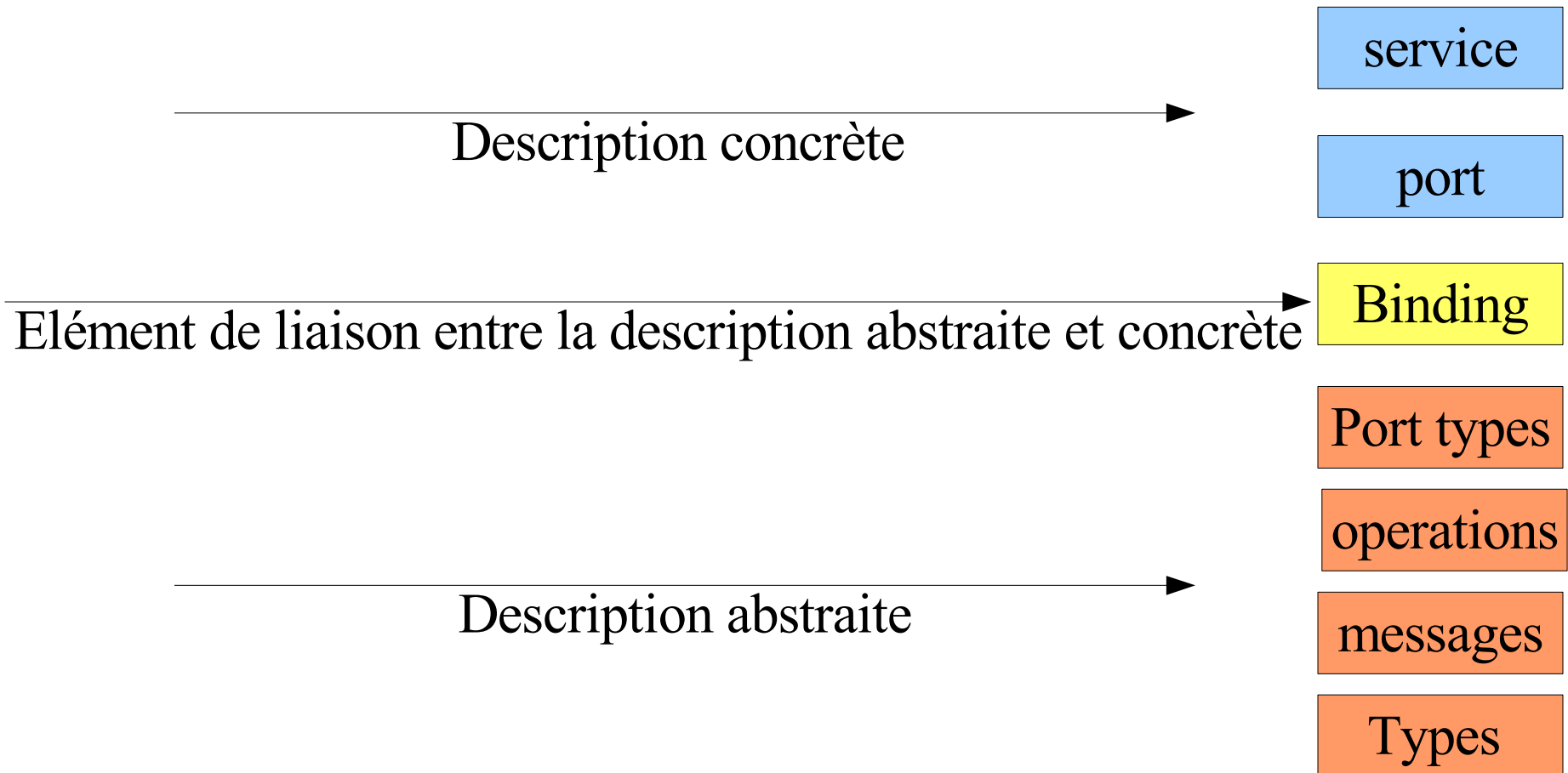
- Un document WSDL
 - est un document XML
 - fournit une description technique du service (en opposition à une description commerciale)
- Un document WSDL est utile pour
 - générer une souche et/ou un squelette
 - Enregistrer & découvrir le service.

Structure d'un document WSDL – description abstraite versus concrète

- Un document WSDL se structure en **2 parties** :
 - Une description concrète (grosso modo, un protocole d'accès et une URI à partir de laquelle on peut accéder aux méthodes)
 - Une description abstraite (nom de la méthode, paramètres d'entrée, de sortie, structure des messages)
- Avantages d'une structuration en 2 parties :
 - la description abstraite peut être réutilisée facilement
 - La description abstraite peut être mutualisée entre plusieurs services et reste réutilisable même si le service est redéployé sur un (ou plusieurs) autre hôtes

Structure d'un document WSDL – description abstraite versus concrète

Le document XML contient 7 éléments :



Structure d'un document WSDL –

service

Service et Port

port

- Un service peut résulter de la **composition** de plusieurs services
 - ➔ Une description WSDL peut inclure/faire référence à plusieurs documents WSDL (d'où l'intérêt d'XML)
- Plusieurs protocoles (HTTP, SMTP, ...) peuvent être utilisés pour accéder à une méthode du service
 - ➔ Description des protocoles et des messages indépendante de la description des opérations

Structure d'un document WSDL –

service

Service et Port

port

- Un **service** peut être **distribué**, c-à-d accessible à partir de plusieurs hôtes
 - Il faut pouvoir définir plusieurs points d'accès
 - Approche : les opérations peuvent être regroupées par point d'accès
- ➔ Un **service** est vue comme un ensemble de point d'entrée
- ➔ Un **port**
 - ➔ correspond à un hôte mettant à disposition des opérations
 - ➔ est identifié par une URI et est accessible par un protocole

Port : point d'accès du service

service

```
<port name="StockQuotePort" binding="tns:StockBinding">  
  <soap:address location="http://ex.com/stockquote"/>  
</port>
```

port

Binding

■ Un port est

Port types

operations

messages

Types

- ❑ Identifié par un **nom**
- ❑ Est **lié** à une partie abstraite
- ❑ Est accessible :
 - À partir d'une **URI**
 - Via un **protocole**

Service : regroupe un ensemble de ports

```
<service name="StockQuoteService">
  <documentation>My StockQuote Service</documentation>
  <port name="StockQuotePort" binding="tns:StockBinding">
    <soap:address location="http://example.com/stockquote"/>
  </port>
  <port name="StockPort" binding="tns:StockBinding">
    <soap:address location="http://example.com/stock"/>
  </port>
</service>
```

service

port

Binding

■ Un service est

- ❑ Est identifié par un **nom**
- ❑ Regroupe un ensemble de **port** (points d'accès)

Port types

operation

messages

Types

Types : définition des types à transmettre

- Un type englobe
 - Les espaces de nommage
 - Les types

service

port

Binding

Port types

operation

messages

Types

```
<types>
  <schema targetNamespace="http://ex.com/stockquote.xsd"
    xmlns="http://www.w3.org/2000/10/XMLSchema">
    TradePriceRequest<element name="TradePriceRequest">
      <complexType>
        <element name="tickerSymbol" type="string"/>
      </complexType>
    </element>
    TradePrice
      <element name="TradePrice">
        <complexType>
          <element name="price" type="float"/>
        </complexType>
      </element>
    </schema>
  </types>
```

string tickerSymbol

float price

Message : collection de types

Deux messages :

TradePriceRequest

string tickerSymbol

GetLastPriceInput

TradePrice

float price

GetLastPriceOutput

```
<message name="GetLastTradePriceInput">  
  <part name="body" element="xsd1:TradePriceRequest"/>  
</message>
```

```
<message name="GetLastTradePriceOutput">  
  <part name="body" element="xsd1:TradePrice"/>  
</message>
```

service

port

Binding

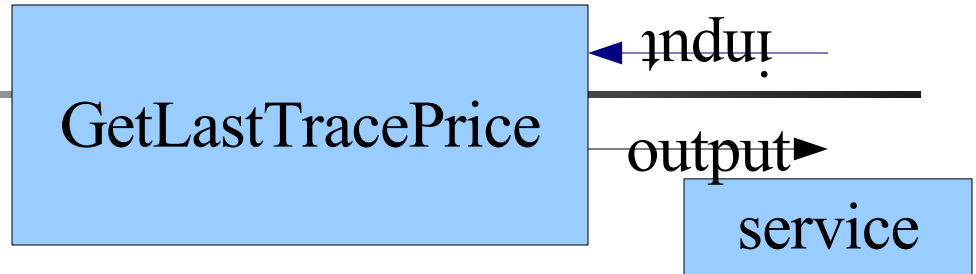
Port types

operation

messages

Types

Opération : collection de types



■ Une opération :

- Est identifiée par un **nom**
- Contient ou non une ou plusieurs **entrée(s)**
- Contient ou non une ou plusieurs **sorties**

port

Binding

Port types

operation

messages

Types

```
<operation name="GetLastTradePrice">
```

```
<soap:operation soapAction="http://ex.com/GetLastTradePrice"/>
```

```
<input>
```

```
<soap:body use="literal"/>
```

```
</input>
```

```
<output>
```

```
<soap:body use="literal"/>
```

```
</output>
```

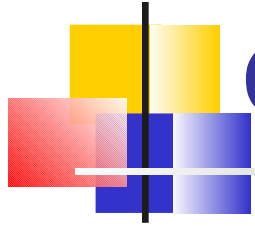
```
</operation>
```



Opération

- 4 types d'opérations :
 - One way : réception requête, pas d'envoi de réponse
 - Requête-réponse : réception requête, envoie d'un réponse
 - Solicite une réponse : envoie d'une requête en vue d'une réponse
 - Notification : envoie d'un message sans attendre la réponse

PortType : collection d'opérations



- Un PortType (sorte d'interface):
 - Est identifiée par un **nom**
 - Contient une collection d'opération(s)

```
<portType name="StockQuotePortType">
  <operation name="setTradePrice">
    <input name="newTradePrice" message="newTradeValue"/>
  </operation>
  <operation name="GetLastTradePrice">
    <input
message="tns:GetLastTradePriceInput"/>
    <output
message="tns:GetLastTradePriceOutput"/>
  </operation>
</portType>
```

Binding

Port types

operation

messages

Types



Binding : liaison

■ Binding :

- Est identifiée par un **nom**
- Associe un **type de port** à un protocole particulier

```
<binding name="StockQuoteSoapBinding"
  type="tns:StockQuotePortType">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="GetLastTradePrice">
    <soap:operation soapAction=
      "http://ex.com/GetLastTradePrice"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
```

Binding

Port types

operation

messages

Types