Les Services Web

Sana Sellami

sana.sellami@univ-amu.fr

2015-2016

PLAN

1. Qu’est ce qu’un service Web?
2. Architecture des Services Web
3. Création et déploiement des Services Web SOAP en Java
4. Services Web REST
5. Développement de Services Web REST en Java

1. QU’ EST CE QU’UN SERVICE WEB

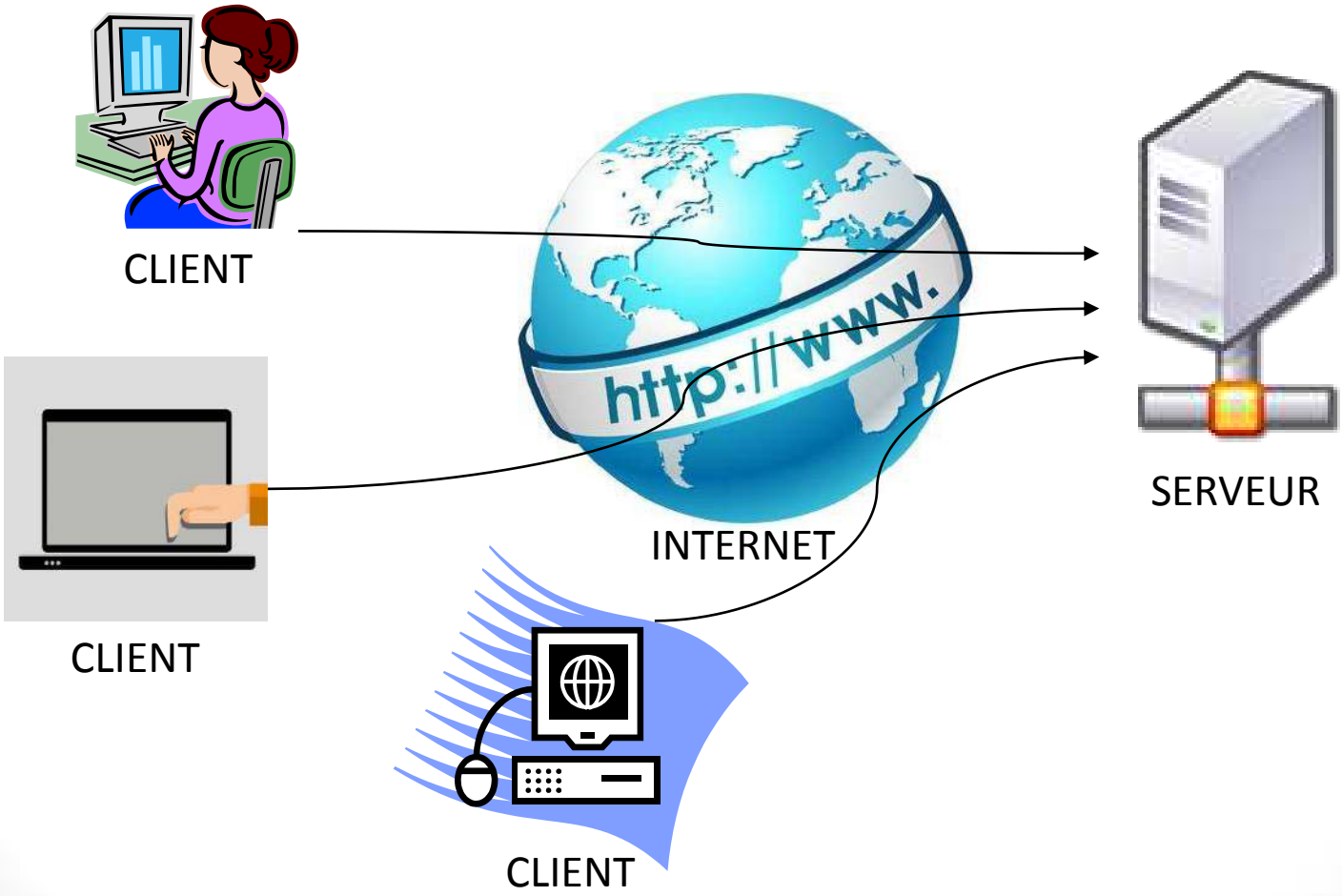
Un service Web .?

Définition du W3C (World Wide Web Consortium) :

Un service web est un système logiciel identifié par un URI, dont les interfaces publiques et les « bindings » sont définies et décrites en XML. Sa définition peut être découverte [dynamiquement] par d'autres systèmes logiciels. Ces autres systèmes peuvent ensuite interagir avec le service web d'une façon décrite par sa définition, en utilisant des messages XML transportés par des protocoles Internet.

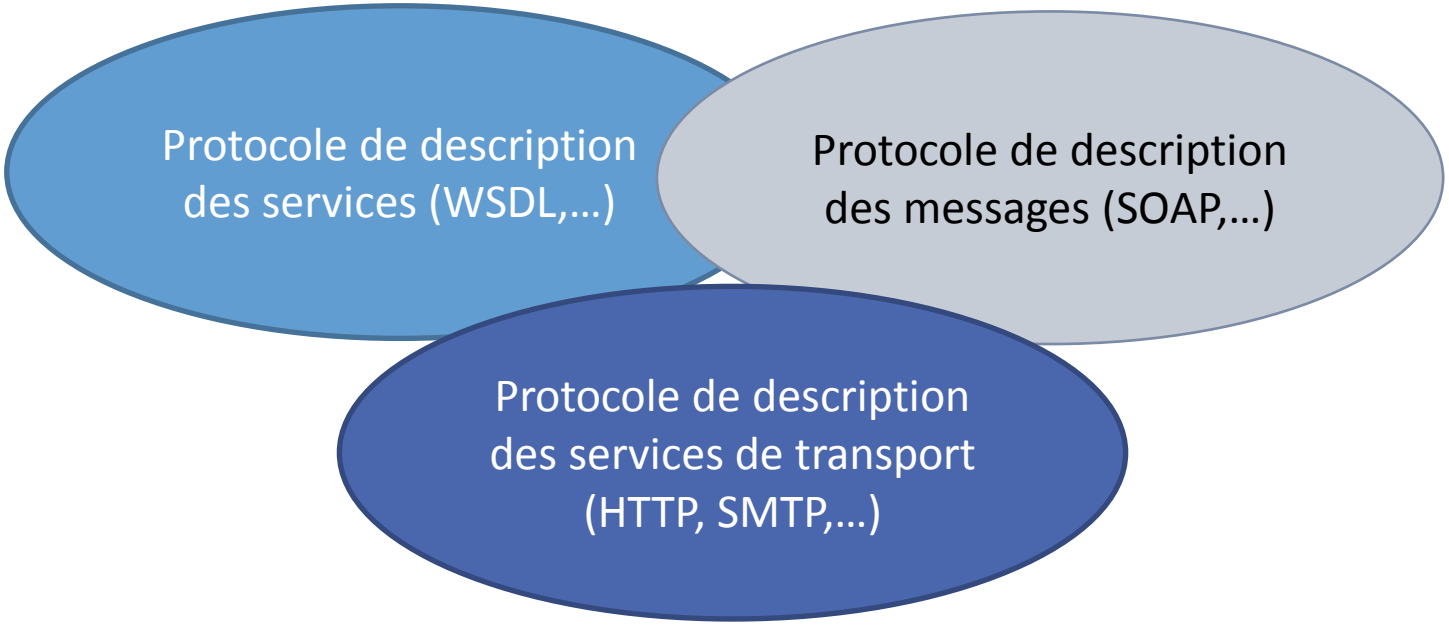
http://www.w3.org/TR/ws-gloss/

Un service Web .?



Les web services Ingénierie systèmes - Concevoir et utiliser des applications 2.0 C#, Java, PHP, API JavaScript, Android SDK, iOS SDK., Jonathan Fontanel, Philippe Lacomme, Libo Ren

Un service Web .?



Les web services Ingénierie systèmes - Concevoir et utiliser des applications 2.0 C#, Java, PHP, API JavaScript, Android SDK, iOS SDK., Jonathan Fontanel, Philippe Lacomme, Libo Ren

Un service Web .?

- Proposant diverses fonctionnalités que d'autres programmes peuvent
  - découvrir dynamiquement
  - et utiliser grâce à des protocoles
    - décrits en XML
    - basés sur l'échange de messages
      - écrits en XML
      - transmis par HTTP, FTP, SMTP, etc.

- Caractéristiques
  - Réutilisable
  - Indépendamment de
    - la plate-forme (UNIX, Windows, …)
    - l'implémentation (VB, C#, Java, …)
    - l'architecture sous-jacente (.NET, JEE, …)

Exemple de services existants

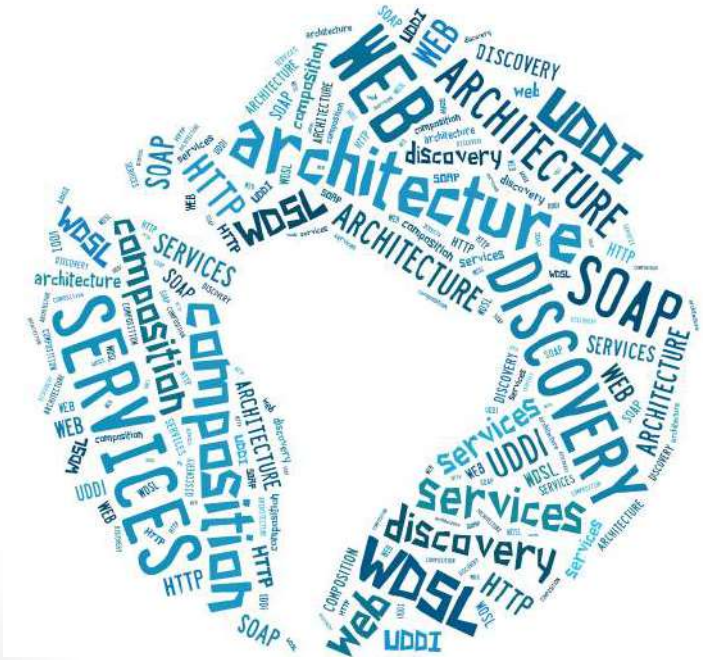
- Google (http://www.google.com/apis/) :
  - accès gratuit mais limité (1000 requêtes par jour après enregistrement)

- Amazon (http://aws.amazon.com/fr/)
  - accès gratuit mais limité (1 requête par seconde après enregistrement)

- bien d'autres ! (cf http://www.xmethods.com/ http://webservicex.net par exemple)

Exemple de services existants

- **Pour la création d'un nuage de tags:**
  - **Wordle:** http://www.wordle.net/
  - **Tag Cloud Generator:** http://www.tagcloud-generator.com/
  - **Tagxedo:** www.tagxedo.com
  - …..



En saisissant un ensemble de mots clés

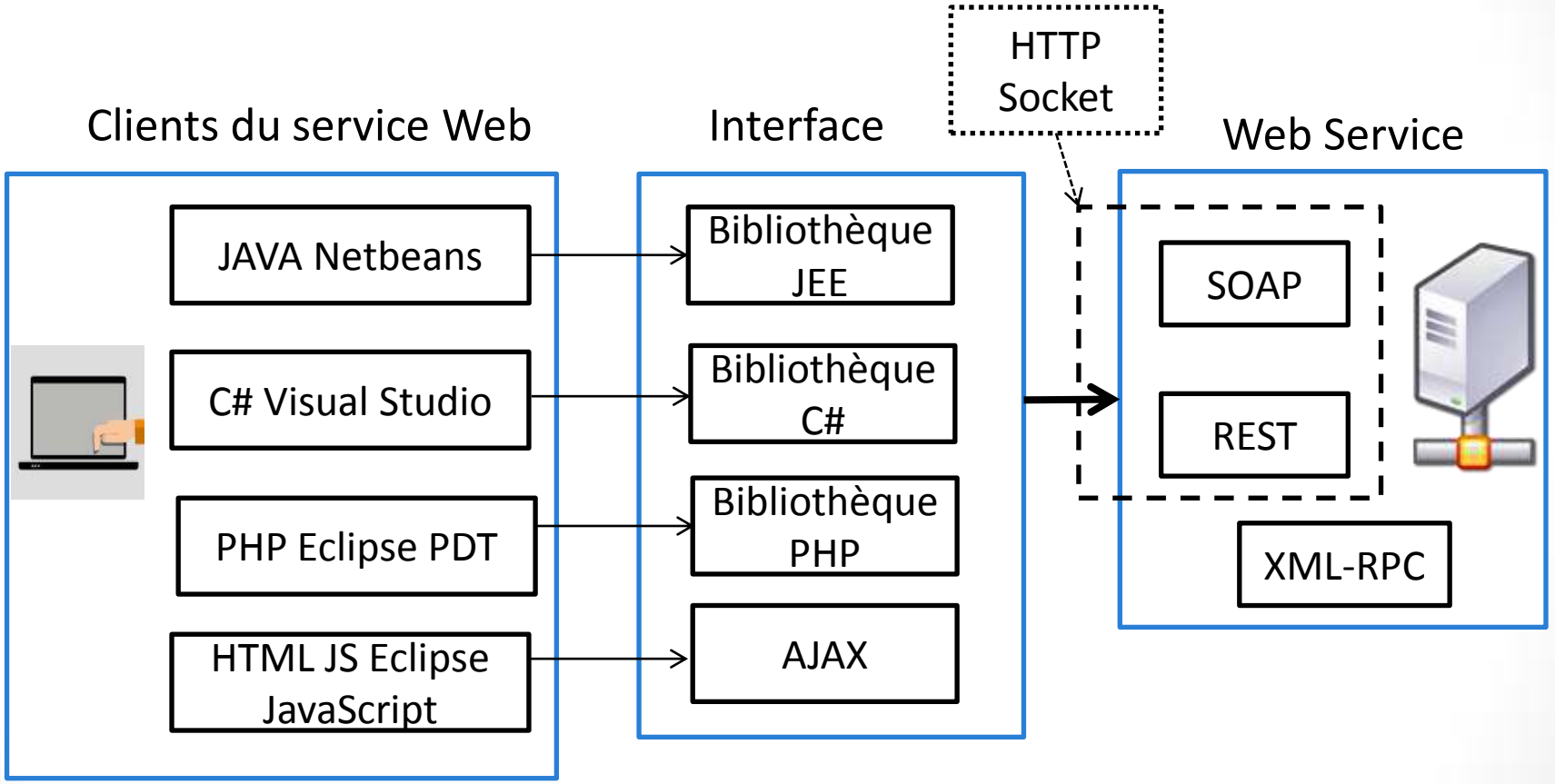
En saisissant l'url de facebook

Exemple de services existants

- Outils vous permettant de vous organiser, de communiquer et de planifier votre travail:
  - **Google Docs**: pour travailler de manière collaborative sur les documents
  - **SlideRocket:** pour créer des présentations plus élaborées que celles de power point
  - **Toggl (Free Time Tracking Software):** pour mesurer le temps passé à réaliser une tâche
  - **Evernote:** pour prendre des notes et pouvez y accéder de n'importe où.
  - Etc…

2. ARCHIECTURES DES SERVICES WEB

Architecture générale des services Web et des clients



Les web services Ingénierie systèmes - Concevoir et utiliser des applications 2.0 C#, Java, PHP, API JavaScript, Android SDK, iOS SDK., Jonathan Fontanel, Philippe Lacomme, Libo Ren

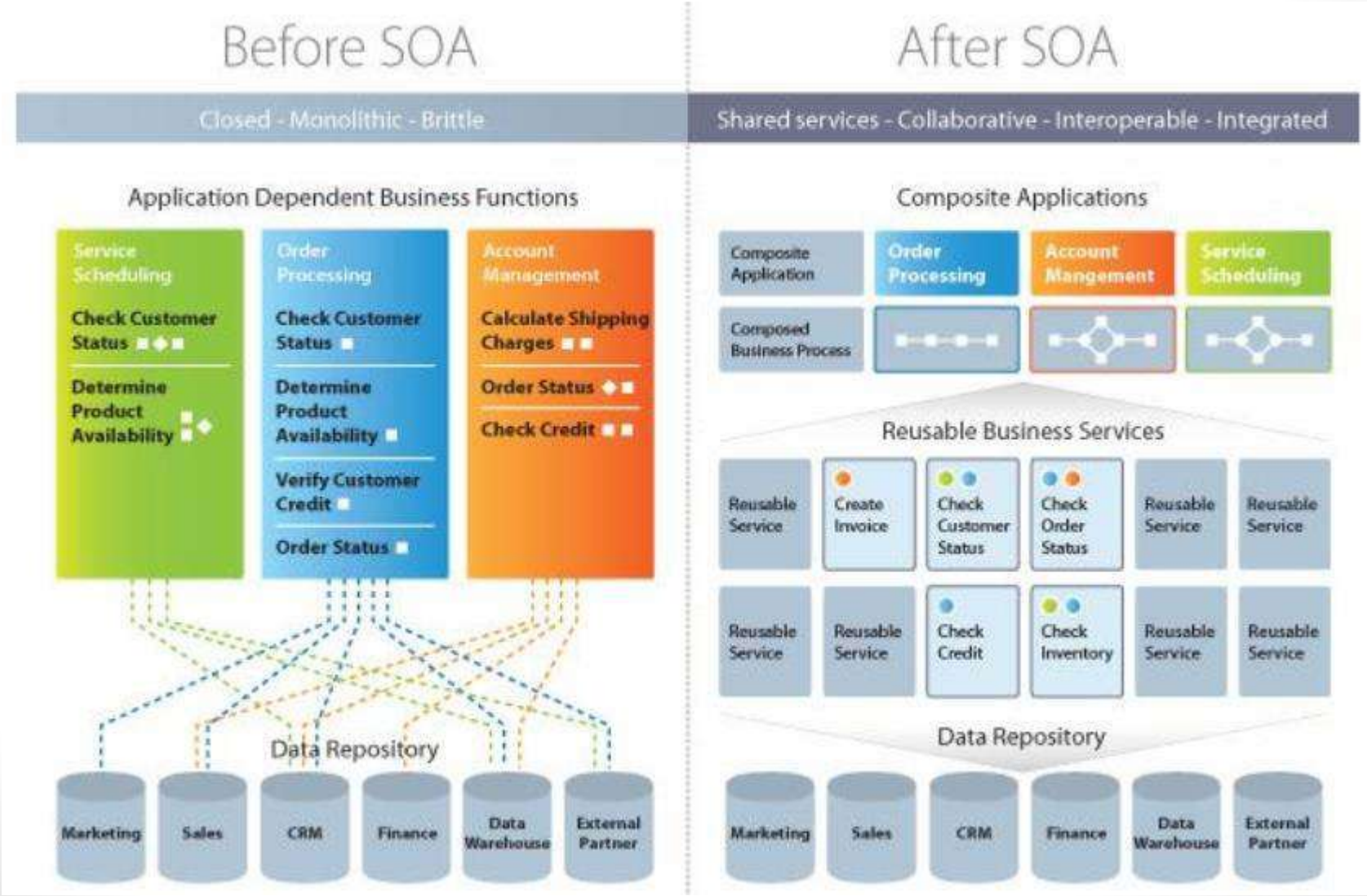
Architecture Orientée Services (SOA)

→Transformer les composants d'un système d'information en services, intégrables à la volée, pour construire des processus métier transverses.

- La notion de SOA renvoie à une nouvelle manière d'intégrer et de manipuler les différentes briques et composants applicatifs d'un système informatique (comptabilité, gestion de la relation client, production, etc.) et de gérer les liens qu'ils entretiennent.

- Repose sur la réorganisation des applications en ensembles fonctionnels appelés services.

Architecture Orientée Services (SOA)

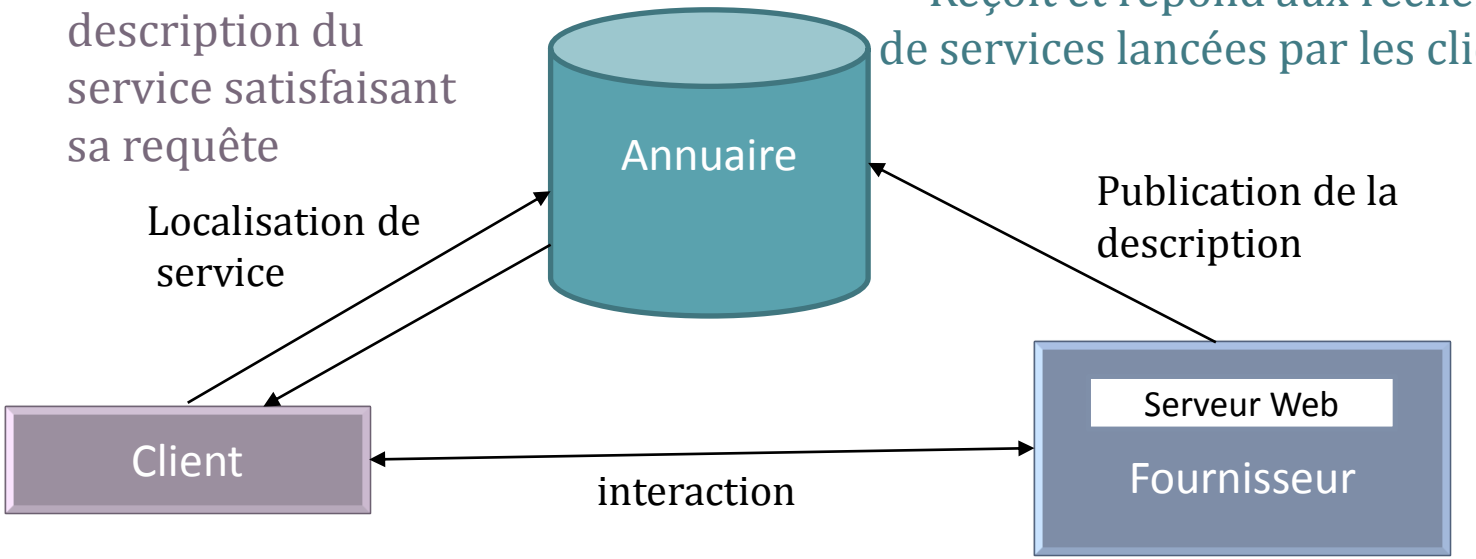


Architecture Orientée Service (SOA)

- Trois acteurs: le fournisseur, annuaire et client

- Reçoit et enregistre les descriptions de services publiées par les fournisseurs
- Reçoit et répond aux recherches de services lancées par les clients

- Obtient la description du service satisfaisant sa requête



- Définit le service
- Publie sa description dans l'annuaire
- Réalise les opérations

Architecture Orientée Service (SOA)

Ordonnancement des actions

- **1. Définition, déploiement et description du service**

Quelle est la fonctionnalité fournie et comment y accéder (description WSDL)

- **2. Publication de la description du service**

Envoi de la description dans un registre (annuaire).

- **3. Recherche du service**

Le client envoie une requête définissant ses besoins au registre, il reçoit en retour une liste de services

Architecture Orientée Service (SOA)

- **4. Récupération de la description du service**

Le client récupère par le registre le lien vers le fichier décrivant le service sélectionné. Il ”sait” maintenant comment accéder au service (comment ”l'invoquer”)

- **5. Exécution (invocation) du service Web**

- Le client peut directement envoyer une requête au service pour réaliser la fonctionnalité
- Il peut aussi récupérer plusieurs descriptions de services différents et les composer pour obtenir une fonctionnalité avancée (ex: voyage)

Standards

- Protocole : **SOAP** = HTTP + XML
  - Requête/réponse = message XML
  - Cadre général permettant l'échange de données structurées au format XML
  - Protocole de transport de ces données basé sur HTTP
- **WSDL** (Web Service Description Language)- Description de service web
  - Description des interfaces des services
- **UDDI** - Découverte automatique des services (dynamicité)
  - Annuaire contenant les interfaces (Pages Jaunes, Vertes, Blanches), permettant d'enregistrer et de rechercher des descriptions de services web

SOAP

- Simple Object Access Protocol, norme W3C
  - SOAP 1.0: 1999, basé sur HTTP
  - SOAP 1.1: 2000, plus générique, autres protocoles
  - SOAP 1.2: recommandation W3C, 2007

- Est un protocole de transmission de messages

- Permet des appels de procédures à distance (RPC) s'appuyant principalement sur le protocole HTTP et sur XML, mais aussi SMTP et POP.

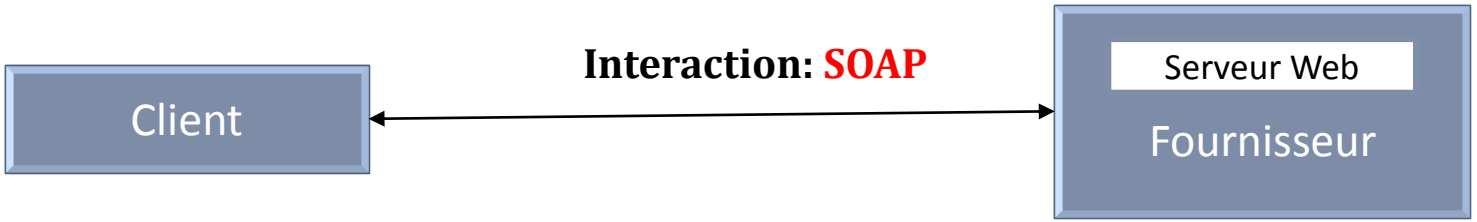
SOAP

- S'appuie sur le protocole HTTP:
  - HyperText Transfert Protocol
  - Tim Berners-Lee, fin 89 - début 90
  - Protocole de communication client/serveur basé sur TCP/IP
  - Simple pour le récupération de documents (GET, HEAD)
  - Simple pour la transmission de données (GET, POST)

- Echange classique avec un service Web
  - Connexion du client vers le serveur
  - Demande d'un document via une méthode GET
  - Renvoi du document, erreur ou information sur le document
  - Déconnexion

SOAP

La requête SOAP intervient sur le réseau
entre le client et le serveur



SOAP Côté client

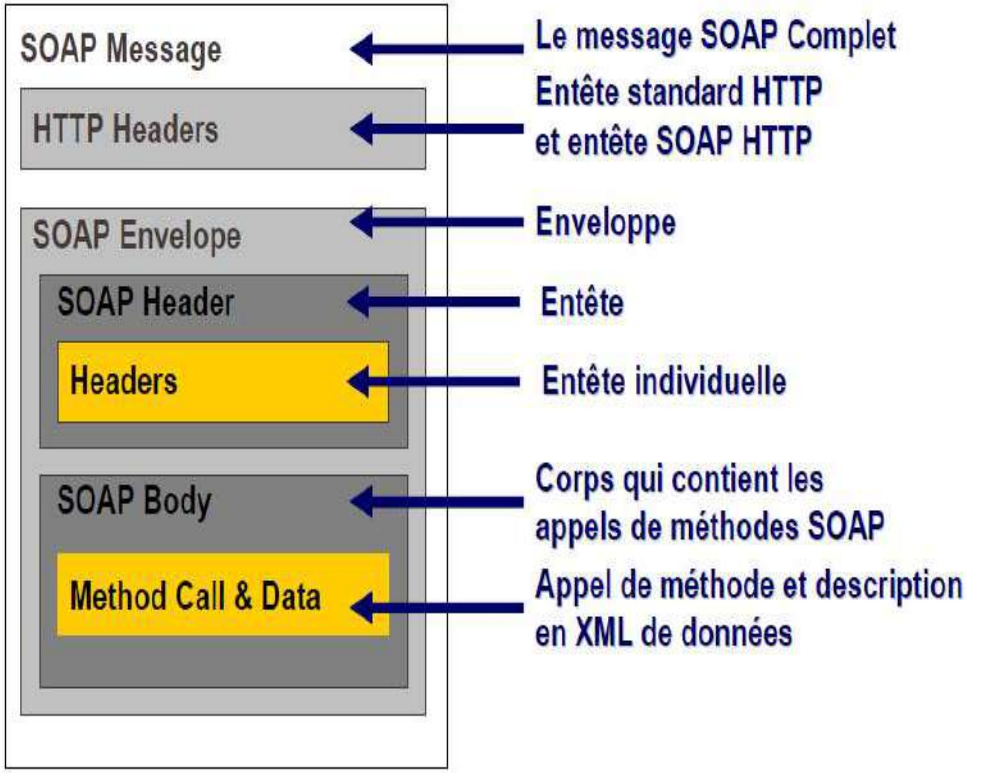
- Ouverture d'une connexion HTTP
- Requête SOAP: document XML décrivant
  - La méthode à invoquer sur la machine distante
  - les paramètres de la méthode

SOAP Côté Serveur

- Récupère la requête
- Exécution de la méthode avec les paramètres
- Renvoie une réponse SOAP (document XML) au client

SOAP: Structure

- **Envelope** expliqant comment la requête doit être traitée et présentant les éléments contenus dans le message.
- **Header** (en-tête) est un mécanisme générique permettant d'ajouter des fonctions à un message SOAP de façon décentralisée sans accord préalable entre les parties en communication.
- **Body** (Corps) contient les informations obligatoires destinées à l'ultime destinataire du message.



SOAP: Exemple

Requête SOAP getsomme envoyée à un service Sommer

```
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:def="http://DefaultNamespace">
   <soapenv:Header/>
   <soapenv:Body>
      <def:getsomme soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
         <a xsi:type="xsd:int">1</a>
         <b xsi:type="xsd:int">3</b>
      </def:getsomme>
   </soapenv:Body>
</soapenv:Envelope>
```

SOAP: Exemple (suite)

Réponse SOAP

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <soapenv:Body>
      <ns1:getsommeResponse soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:ns1="http://DefaultNamespace">
         <getsommeReturn xsi:type="xsd:int">4</getsommeReturn>
      </ns1:getsommeResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

WSDL (Web Services Description Language)

- Langage de description des services web
- Depuis 2007, WSDL 2.0 est une recommandation W3C
- **Objectif:** Décrire les services comme un ensemble d'opérations et de messages abstraits reliés (bind) à des protocoles et des serveurs réseaux
- Basé sur le langage XML (schéma XML)

WSDL (Web Services Description Language)

- Regroupe les informations nécessaires pour interagir avec le service :
  - 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

WSDL (Web Services Description Language)

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

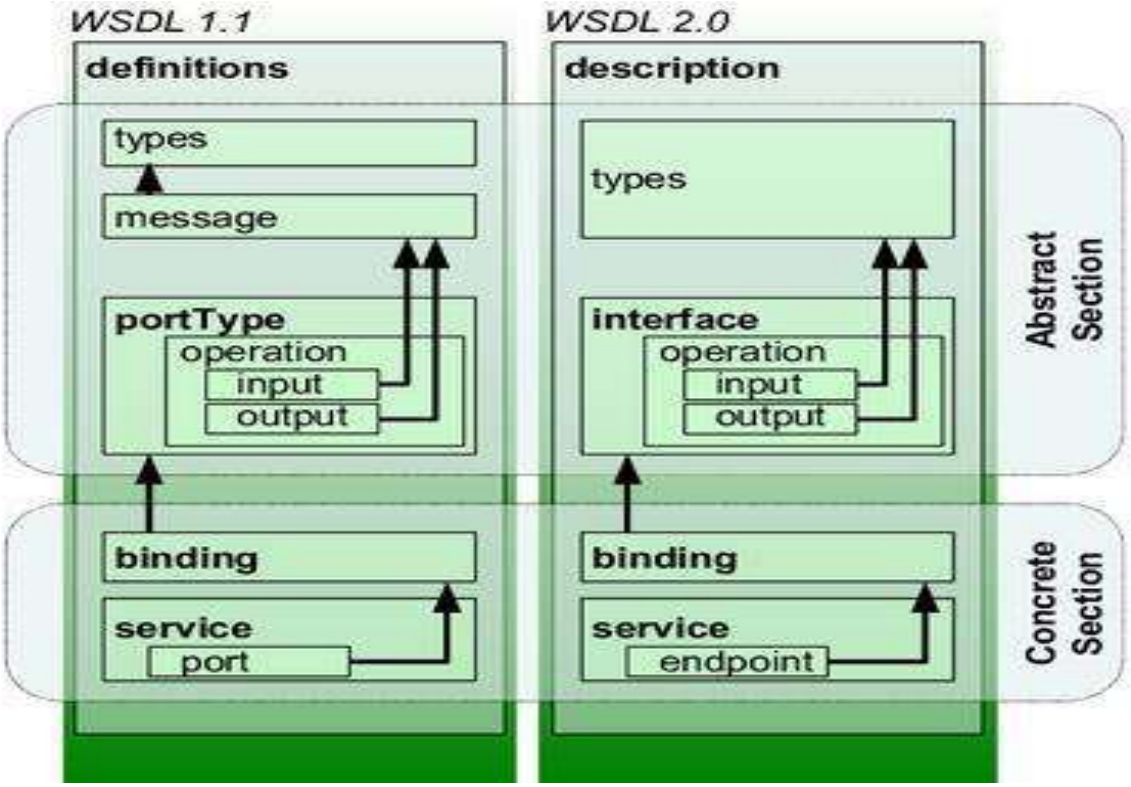
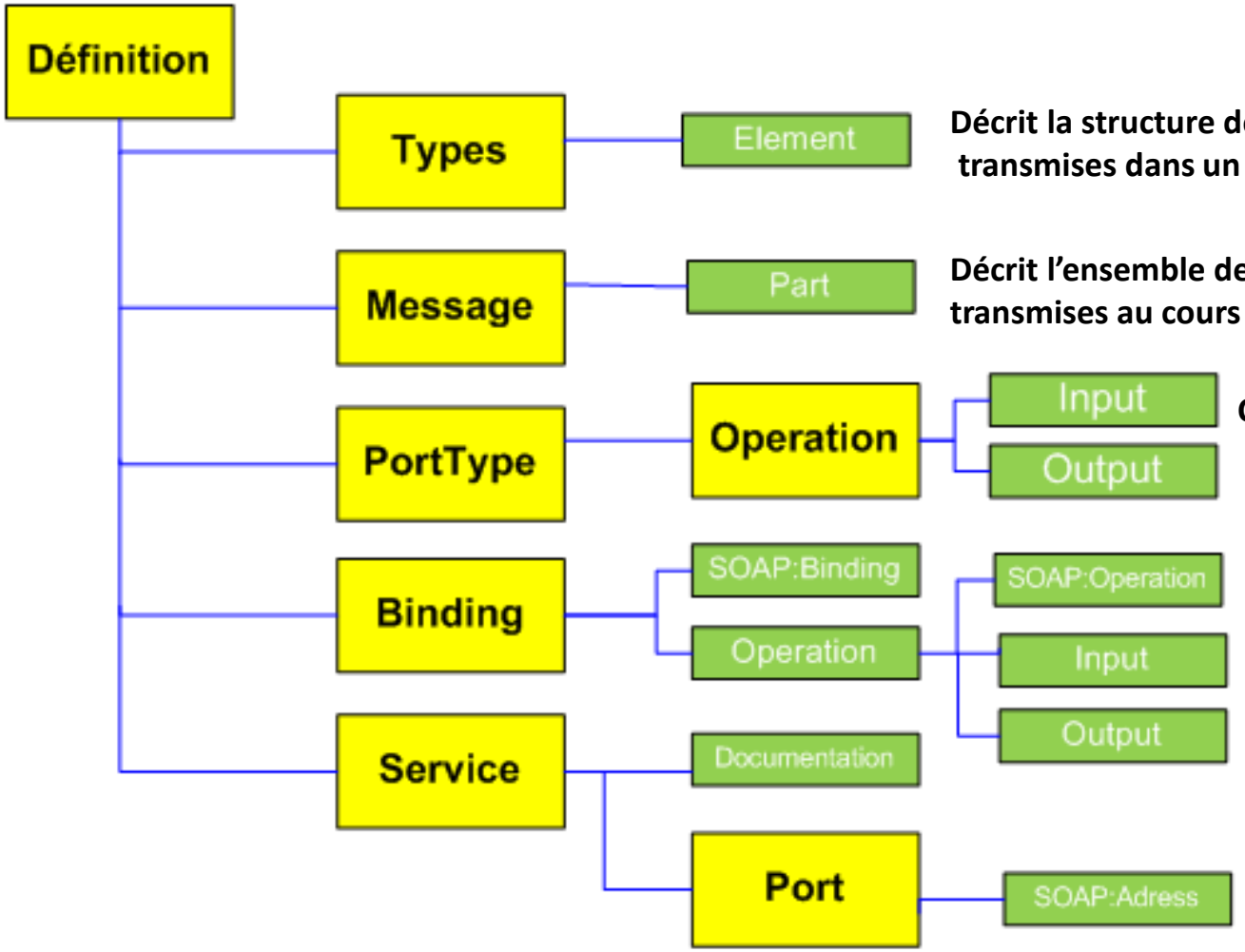


Figure : Concepts définis par WSDL 1.1 et WSDL 2.0

http://www.w3.org/TR/wsdl20/

Source: http://en.wikipedia.org/wiki/Web_Services_Description_Language

WSDL (Web Services Description Language)



Décrit la structure des données transmises dans un message

Décrit l'ensemble des données transmises au cours de l'opération

Composé de l'ensemble des opérations abstraites

Décrit la façon dont un ensemble d'opérations abstraites, appelé « type de port », est lié à un port selon un protocole réel

Un ensemble de points finaux du réseau appelés « ports » permettant de déterminer la localisation du service

Exemple WSDL 1.1: sommer.wsdl

```
<wsdl:definitions targetNamespace="http://localhost:8080/axis/sommer.jws">
<!--
   WSDL created by Apache Axis version: 1.4
   Built on Apr 22, 2006 (06:55:48 PDT)
-->
 <wsdl:message name="getsommeResponse">
    <wsdl:part name="getsommeReturn" type="xsd:int"/>
 </wsdl:message>
 <wsdl:message name="getsommeRequest">
    <wsdl:part name="a" type="xsd:int"/>
    <wsdl:part name="b" type="xsd:int"/>
 </wsdl:message>
 <wsdl:portType name="sommer">
  <wsdl:operation name="getsomme" parameterOrder="a b">
     <wsdl:input message="impl:getsommeRequest" name="getsommeRequest"/>
     <wsdl:output message="impl:getsommeResponse" name="getsommeResponse"/>
    </wsdl:operation>
 </wsdl:portType>
 <wsdl:binding name="sommerSoapBinding" type="impl:sommer">
    <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="getsomme">
     <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="getsommeRequest">
       <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://DefaultNamespace" use="encoded"/>
     </wsdl:input>
    <wsdl:output name="getsommeResponse">
       <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://localhost:8080/axis/sommer.jws" use="encoded"/>
     </wsdl:output>
    </wsdl:operation>
 </wsdl:binding>
 <wsdl:service name="sommerService">
  <wsdl:port binding="impl:sommerSoapBinding" name="sommer">
     <wsdlsoap:address location="http://localhost:8080/axis/sommer"/>
    </wsdl:port>
 </wsdl:service>
</wsdl:definitions>
```

Racine du document

Définition des types de données (facultatif)

Message: Définition des messages échangeables

PortType: définition des ensembles d'opérations

service: localisation des services web

Exemple : Gestion de compte

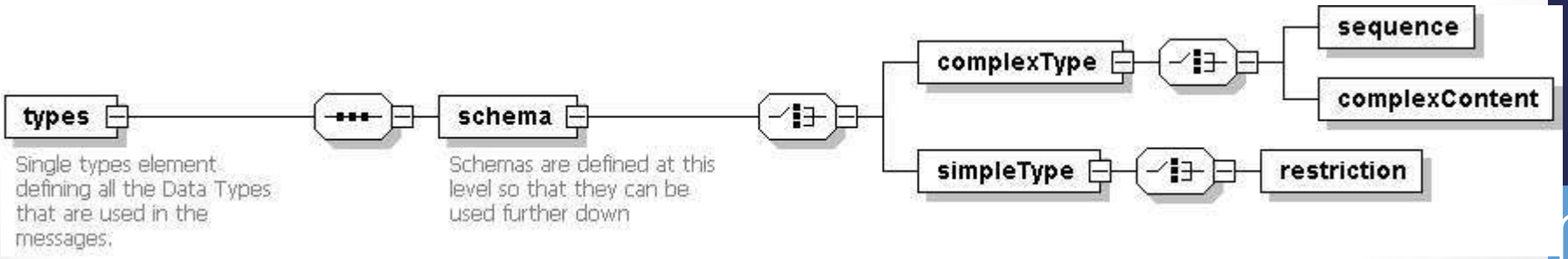
- **Description: Une application de gestion de compte**

- **Définition de l'interface Java:**

```
import java.util.*;
public interface CompteInterface {
public void depotDe(int montant);
public boolean retraitDe(int montant);
public int valeurDuSolde();
public Vector listeMouvements();
}
```

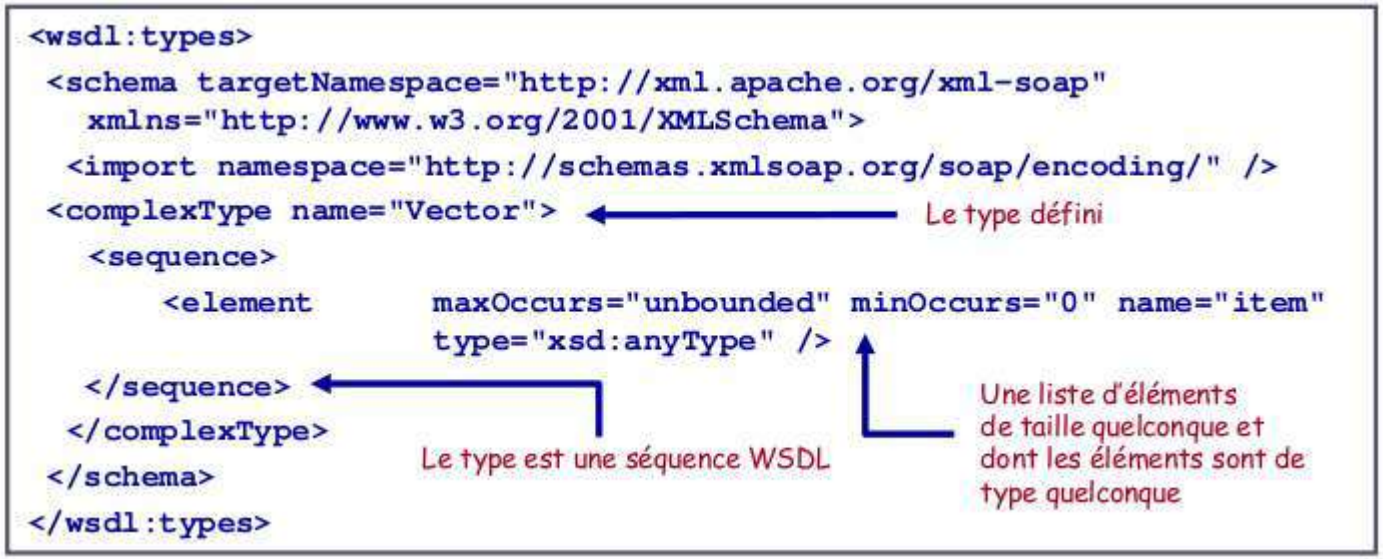
Structure d'un WSDL : <Types>

- Un type décrit la structure de données transmises dans un message.
- Contient les définition de types utilisant un système de typage par défaut XML Schema (XSD).
- Pouvant contenir des types simples et complexes



Structure d'un WSDL : <Types>

- Par exemple, dans la gestion de compte, la méthode listeMouvements retourne un Vector. Nous aurons alors la description de ce type, comme illustré ici :



Structure d'un WSDL : <message>

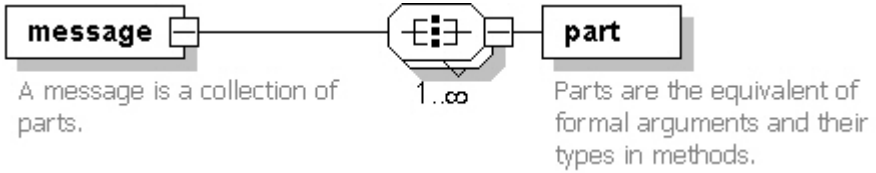
- Décrit les données associées à une opération (1 requête et 1 réponse HTTP par opération, 1 message d'erreur "fault" optionnel).
- Un document WSDL peut contenir zéro ou plusieurs messages.
- Par exemple, la méthode listeMouvements disposera de deux messages (un pour l'appel et un pour la réponse):

```
<wsdl:message name="listeMouvementsRequest" />

<wsdl:message name="listeMouvementsResponse">
  <wsdl:part name="listeMouvementsReturn" type="apachesoap:Vector" />
</wsdl:message>
```

Structure d'un WSDL : <message>

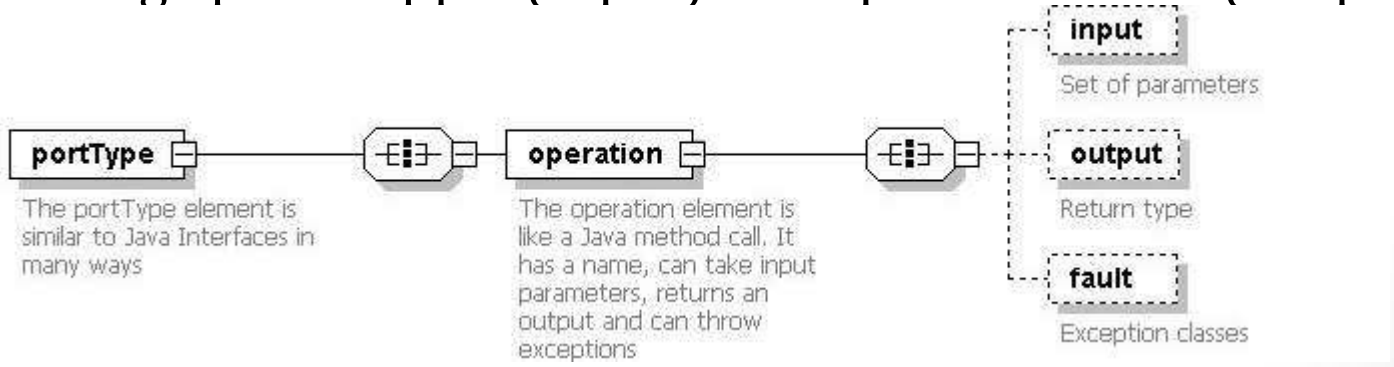
- Chaque message peut être un composé de plusieurs parties




```
  <wsdl:message name="listeMouvementsRequest" />
+ <wsdl:message name="depotDeRequest">
    <wsdl:part name="in0" type="xsd:int" />
  </wsdl:message>
+ <wsdl:message name="listeMouvementsResponse">
    <wsdl:part name="listeMouvementsReturn" type="apachesoap:Vector" />
  </wsdl:message>
+ <wsdl:message name="valeurDuSoldeResponse">
    <wsdl:part name="valeurDuSoldeReturn" type="xsd:int" />
  </wsdl:message>
  <wsdl:message name="depotDeResponse" />
  <wsdl:message name="valeurDuSoldeRequest" />
+ <wsdl:message name="retraitDeResponse">
    <wsdl:part name="retraitDeReturn" type="xsd:boolean" />
  </wsdl:message>
+ <wsdl:message name="retraitDeRequest">
    <wsdl:part name="in0" type="xsd:int" />
  </wsdl:message>
```

Structure d'un WSDL : <portType>

- Un document WSDL peut contenir 0 à plusieurs portType
- L'élément portType contient un seul attribut name. La convention de nommage nameOfWebService PortType.
- Composé d'un ensemble d'opérations abstraites (i.e. signature de la méthode). Une opération est composée d'un message pour l'appel (Input) et un pour le retour (Output).



Structure d'un WSDL : <portType>

- Définition d'un seul type de port, avec les 4 opérations abstraites, correspondant aux 4 déclarations de méthodes dans l'interface Java :

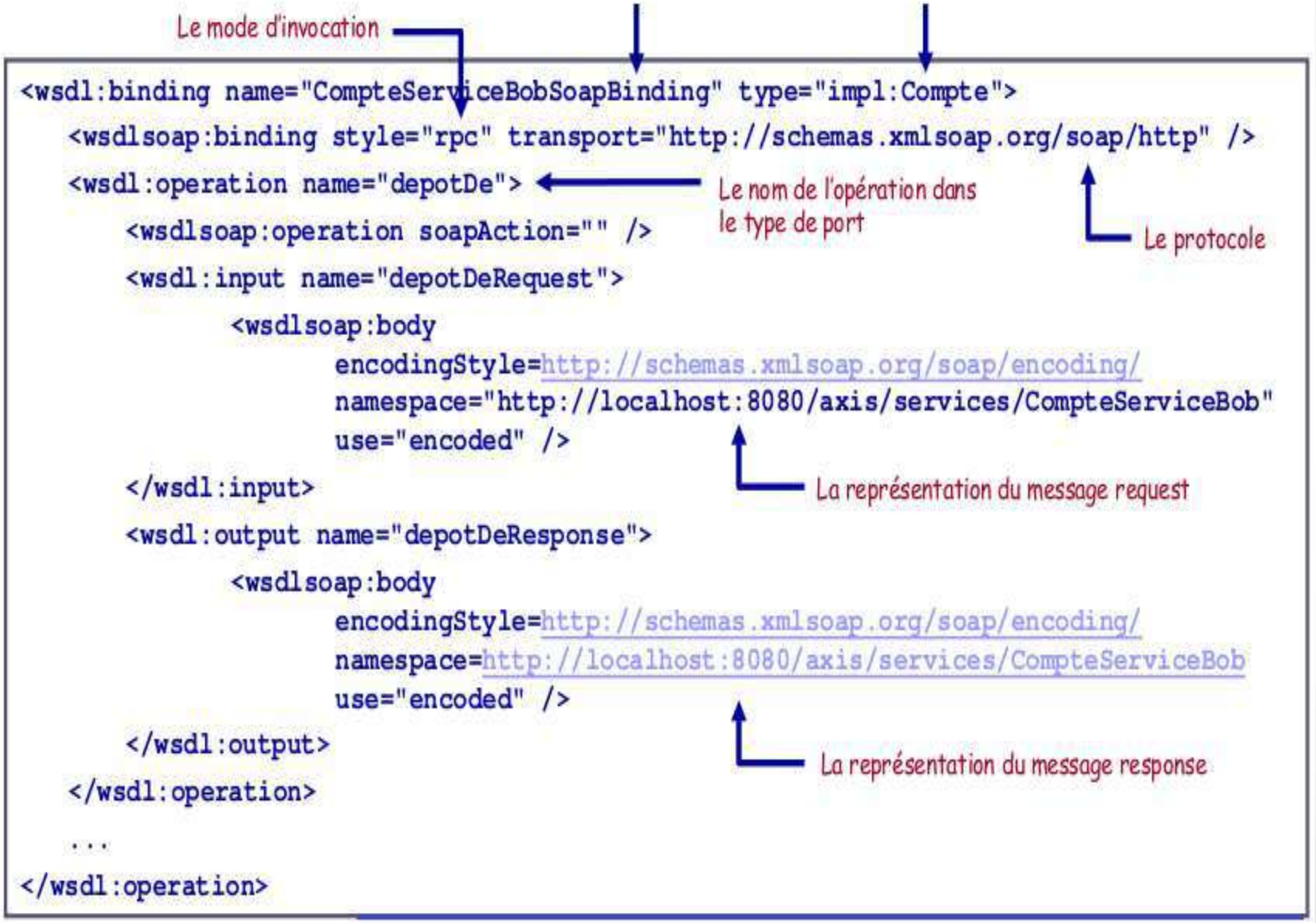

```
<wsdl:portType name="Compte">
+ <wsdl:operation name="depotDe" parameterOrder="in0">
    <wsdl:input message="impl:depotDeRequest" name="depotDeRequest" />
    <wsdl:output message="impl:depotDeResponse" name="depotDeResponse" />
  </wsdl:operation>
+ <wsdl:operation name="retraitDe" parameterOrder="in0">
    <wsdl:input message="impl:retraitDeRequest" name="retraitDeRequest" />
    <wsdl:output message="impl:retraitDeResponse" name="retraitDeResponse" />
  </wsdl:operation>
+ <wsdl:operation name="valeurDuSolde">
    <wsdl:input    message="impl:valeurDuSoldeRequest«
                   name="valeurDuSoldeRequest" />
    <wsdl:output   message="impl:valeurDuSoldeResponse«
                   name="valeurDuSoldeResponse" />
  </wsdl:operation>
- <wsdl:operation name="listeMouvements">
    <wsdl:input    message="impl:listeMouvementsRequest"
                   name="listeMouvementsRequest" />
    <wsdl:output message="impl:listeMouvementsResponse«
                   name="listeMouvementsResponse" />
  </wsdl:operation>
</wsdl:portType>
```

Structure d'un WSDL : <binding>

- Une liaison (ou binding) décrit la façon dont un portType(en d'autres termes l'abstraction du service, i.e. ses opérations abstraites) est mis en oeuvre pour un protocole particulier (HTTP par exemple) et un mode d'invocation (RPC par exemple).

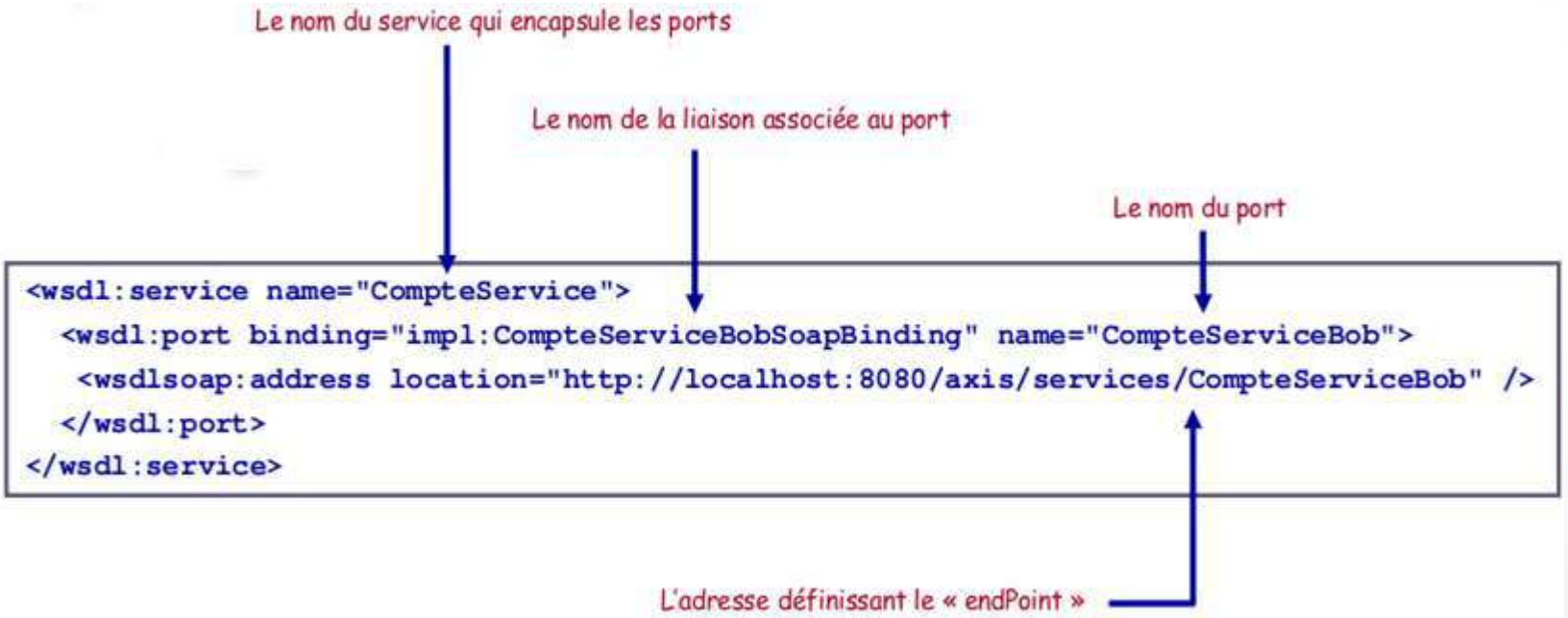
- Pour un portType, on peut avoir plusieurs liaisons, pour différencier les modes d'invocation (RPC ou autres) ou de transport (HTTP ou autre) des différentes opérations.

Structure d'un WSDL : <binding>



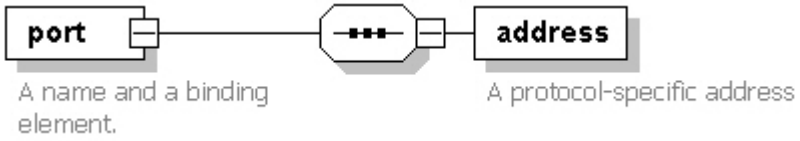
Structure d'un WSDL : <service>

- Un service est décrit comme un ensemble de points finaux du réseau appelés « ports »



Structure d'un WSDL : <port>

- Un port spécifie une URL qui correspond à l'implémentation du service par un fournisseur.
- Le port est associé à un « binding » définissant ainsi un simple point de terminaison (endpoint:@ où se situe le WS)



Le nom de la liaison associée

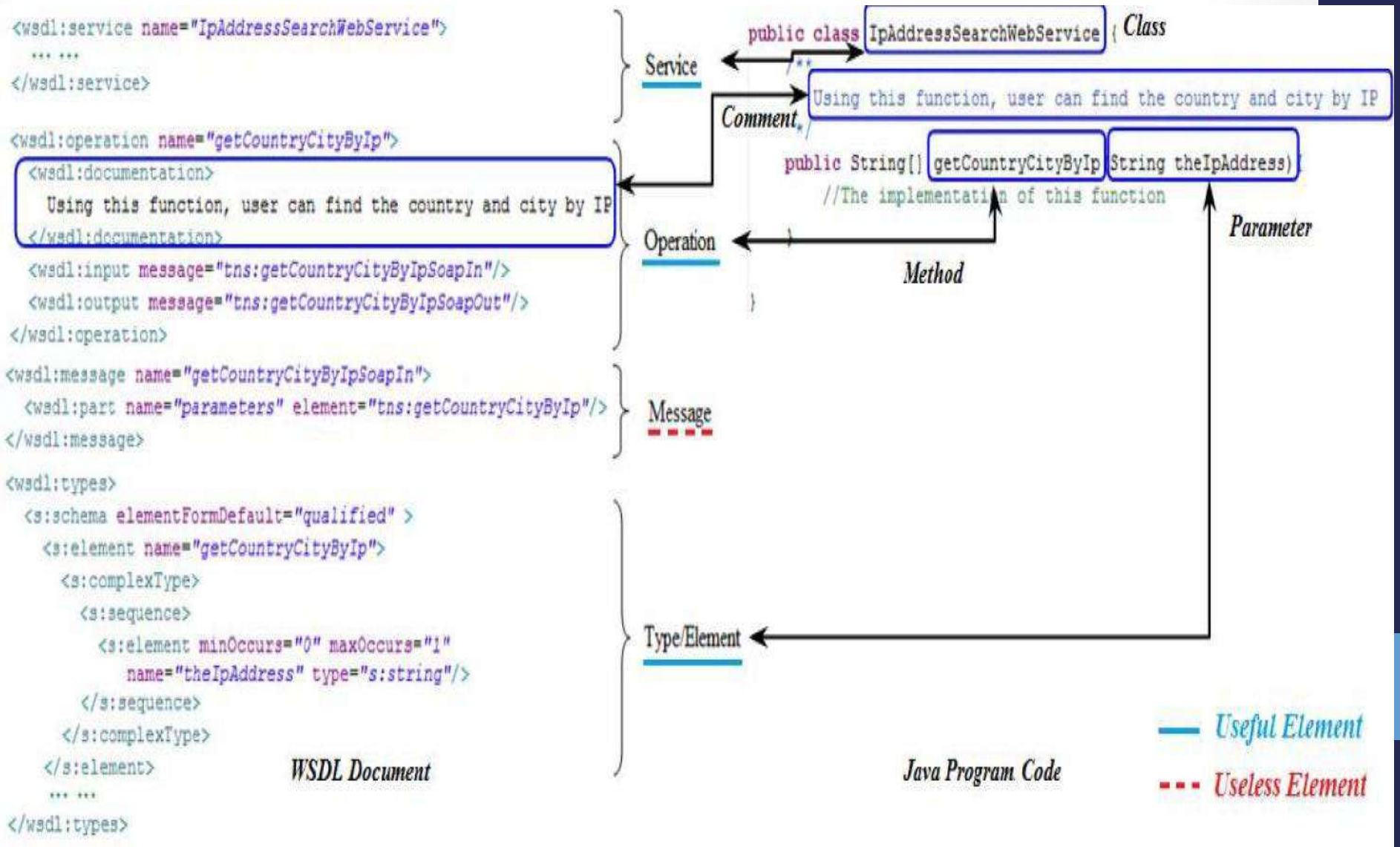
Le nom du port

```
<wsdl:port binding="impl:CompteServiceBobSoapBinding "name="CompteServiceBob">
  <wsdlsoap:address location="http://localhost:8080/axis/services/CompteServiceBob" />
</wsdl:port>
```

Comme indiqué précédemment, pour un même port (Compte par exemple), on peut avoir plusieurs liaisons qui correspondent à des mises en œuvre protocolaires différentes des opérations du service. On retrouve donc ici, les ports associés aux différentes liaisons définies.

L'adresse définissant le « endPoint »

Mapping Java←→ WSDL



Mapping Java←→ WSDL

Code Java	Fichier WSDL
Classe Java public class IpAddressSearchWebService	**Service** <wsdl: service name="IpAddressSearchWebService"> </wsdl:service>
Commentaire /*using this function, user can find the country and city by Ip */	**Documentation** <wsdl:documentation> using this function, user can find the country and city by Ip </wsdl:documentation>
Méthode public String[] getCountryCityByIp(String theIpAddress)	**Operation** <wsdl:operation name="getCountryCityByIp"> <wsdl:input ….> <wsd:output….> </wsdl:operation>
Paramètre String theIpAddress	**Types** <wsdl:types> <s:schema elementFromDefault=“qualified"> <s:element name="getCountryCityByIp"> <s:complexType> <s:sequence> <s:element minoccurs=“0" maxoccurs=“1" name=“theIpAddress" type=“s:string" /> </s:sequence> </s:complexType>…</s:element> </wsdl:types>

UDDI : Universal Description, Discovery and Integration

- Universal Description, Discovery and Integration

- Historique
  - À l'origine: annuaire universel pour les services web (à la Google)
  - Aujourd'hui: vise plutôt les environnements privés, à petite échelle
  - Raisons: peu d'annuaires généraux UDDI (IBM, Microsoft, …), contenu pauvre et non fiable
    - Meilleure fiabilité en environnements contraints, privés (~EAI)
    - Élément d'infrastructure qui aide aussi à stocker des infos absentes en WSDL

- Versions
  - Version 1: les bases d'un annuaire de services
  - Version 2: adaptation à SOAP et WSDL
  - Version 3: redéfinition du rôle UDDI, accent sur les implémentations privées, sur l'interaction entre annuaires privés et publics

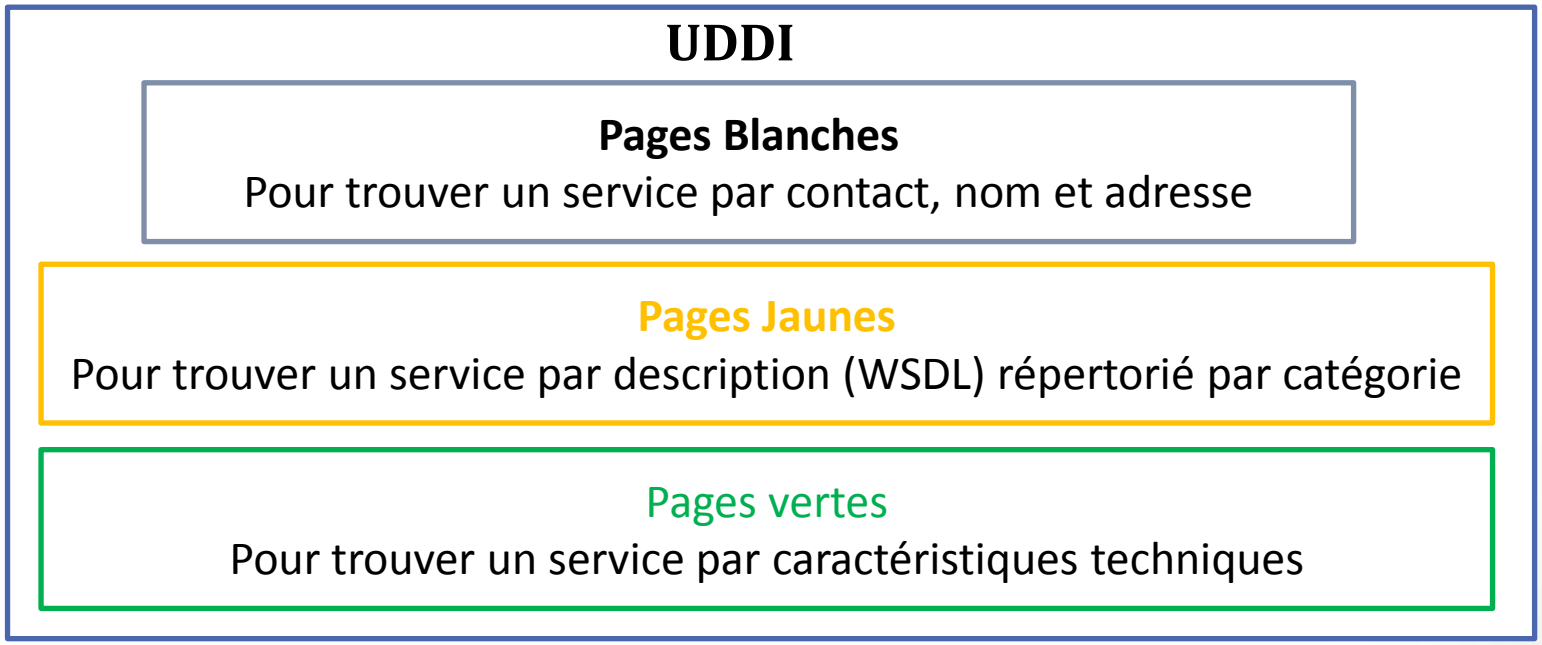
UDDI : Universal Description, Discovery and Integration

- L'annuaire UDDI permet de :
  - Publier, découvrir des informations sur une entreprise et ses services
- L'inscription sur UDDI permet à une entreprise de se présenter ainsi que ses services
- L'adoption de UDDI facilite le développement des échanges de type « B2B »
  - L'enregistrement des services dans un annuaire s'effectue auprès d'un opérateur (Microsoft ou IBM actuellement) à travers son site mais on peut créer ses propres registres UDDI (UDDI4J, jUDDI)
  - http://uddi.xml.org/
- Un annuaire à l'aide d'un browser en ligne:

  http://soapclient.com/UDDIAdv.html

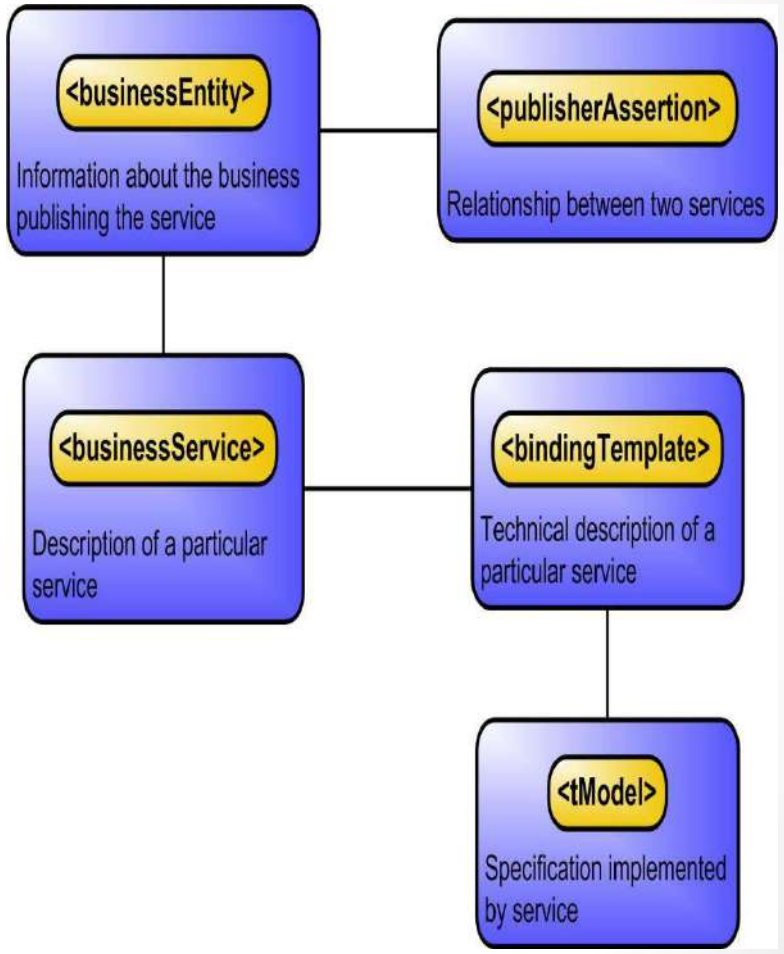
UDDI : Universal Description, Discovery and Integration

- Comporte plusieurs catégories de données: Informations organisées en trois méthodes

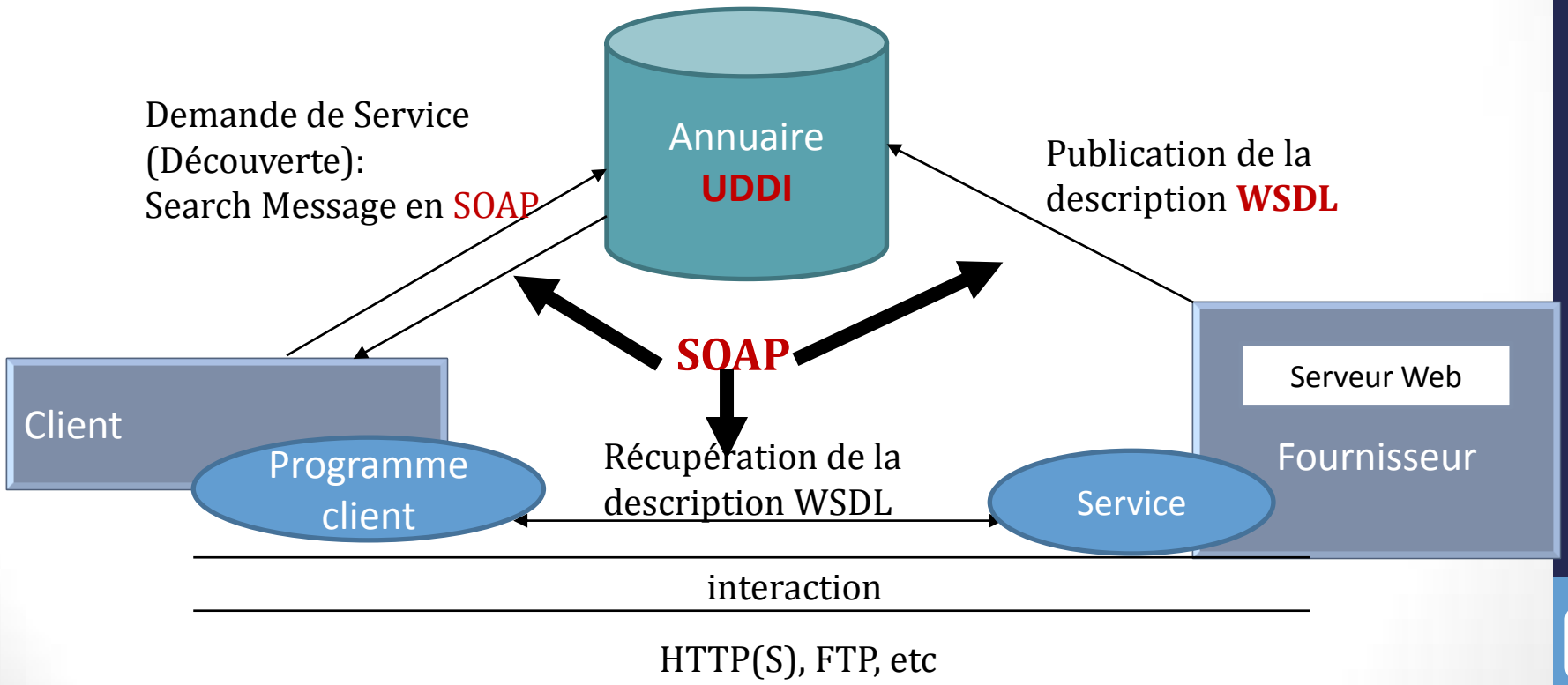


UDDI

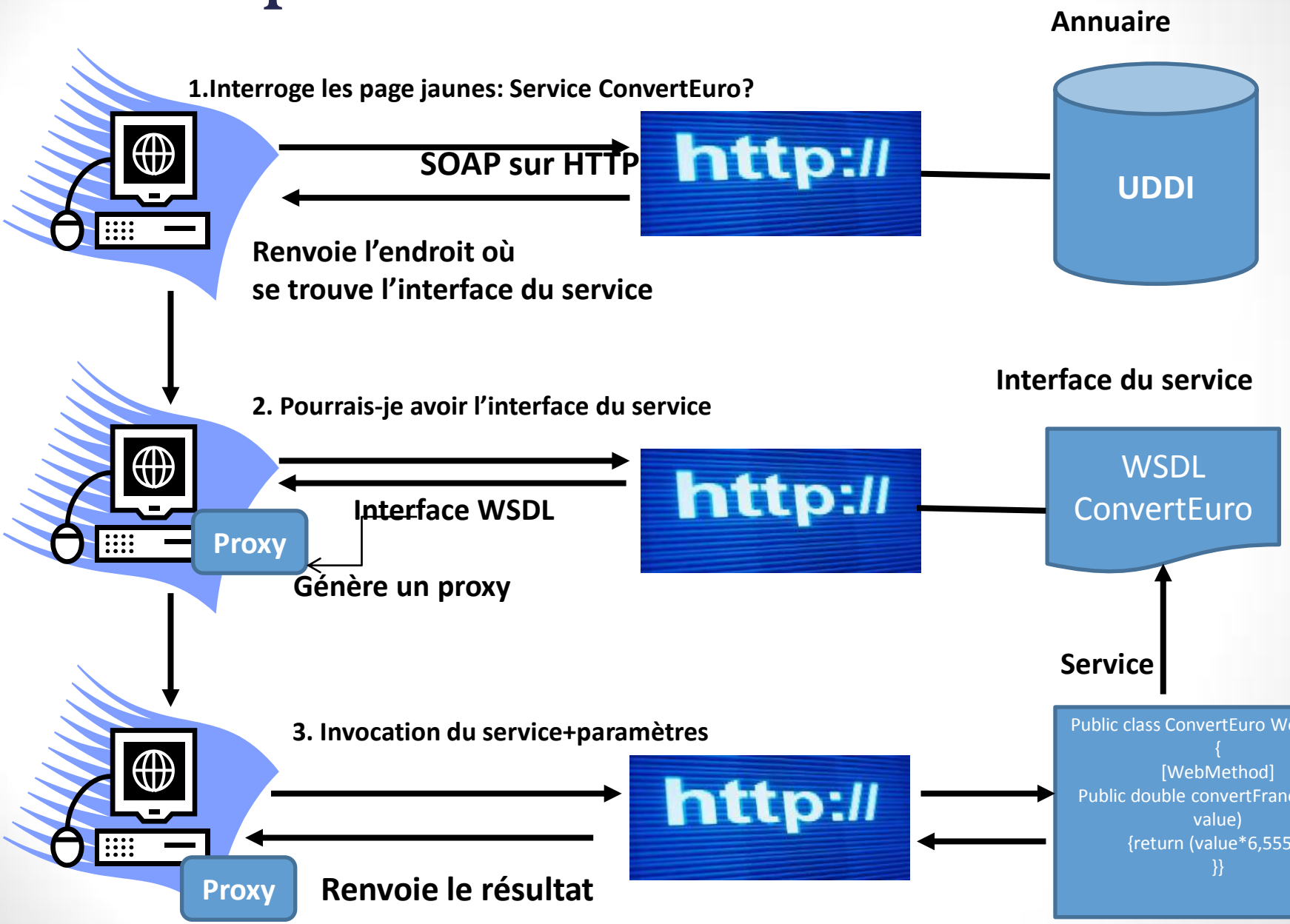
- Le modèle UDDI comporte 5 structures de données principales décrites sous forme de schémas XML :
  - **BusinessEntity** : ensemble d'informations sur l'entreprise qui publie les services dans l'annuaire
  - **BusinessService** : ensemble d'informations sur les services publiés par l'entreprise
  - **BindingTemplate** : ensemble d'informations concernant le lieu d'hébergement du service (i.e. adresse du fournisseur)
  - **tModel** : ensemble d'informations concernant le mode d'accès du service (définitions WSDL)
  - **publisherAssertion** : ensemble d'informations contractuelles entre partenaires en échanges commerciaux



Revenons à l'architecture

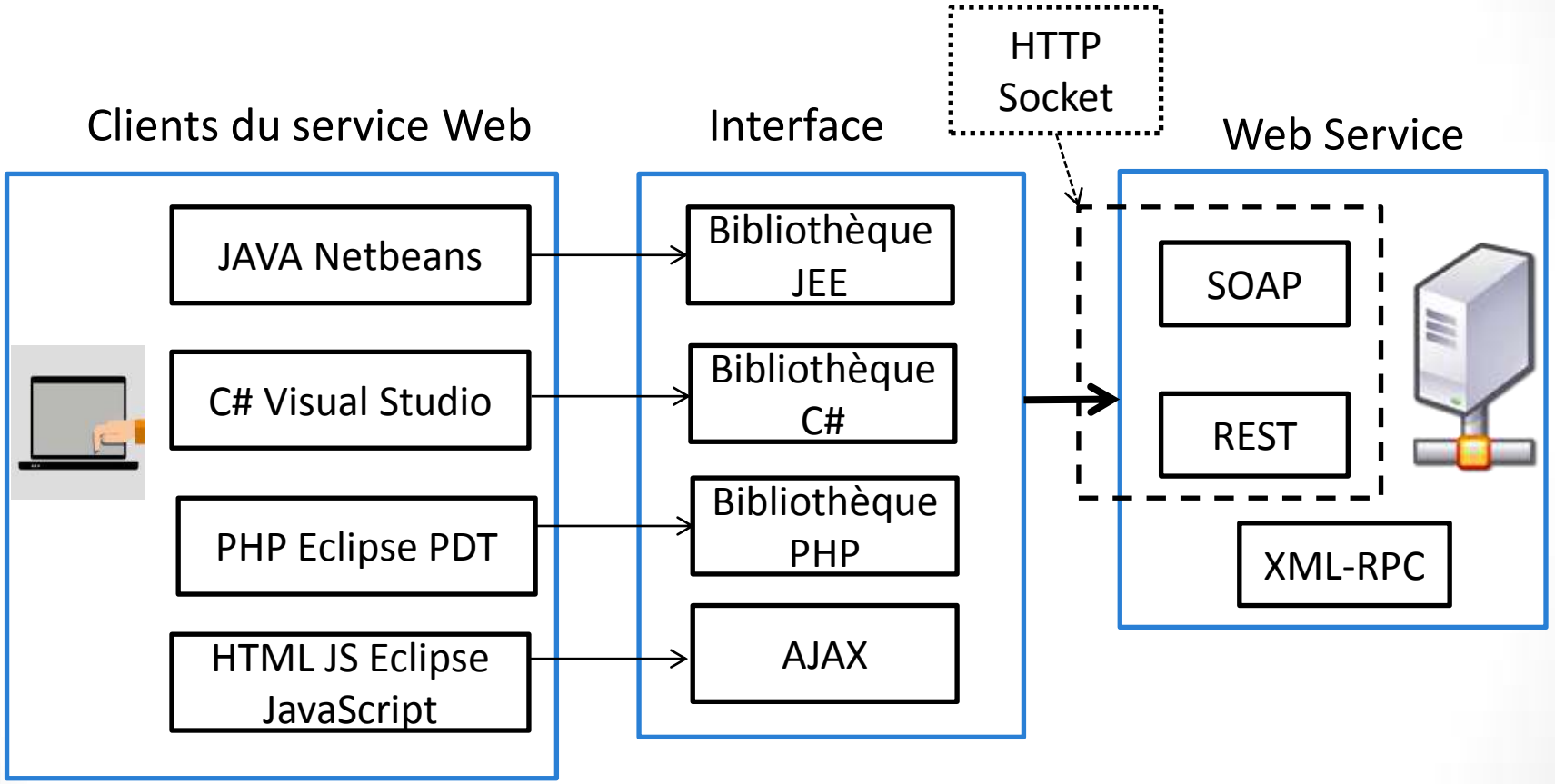


Exemple


Annuaire
1.Interroge les page jaunes: Service ConvertEuro?
SOAP sur HTTP
http://
UDDI
Renvoie l'endroit où
se trouve l'interface du service
Interface du service
2. Pourrais-je avoir l'interface du service
Interface WSDL
http://
WSDL
ConvertEuro
Proxy
Génère un proxy
Service
3. Invocation du service+paramètres
http://
Public class ConvertEuro WebService
{
[WebMethod]
Public double convertFranc(double
value)
{return (value*6,5559)
}}
Proxy
Renvoie le résultat

3. CRÉATION ET DÉPLOIEMENT DE SERVICES WEB EN JAVA

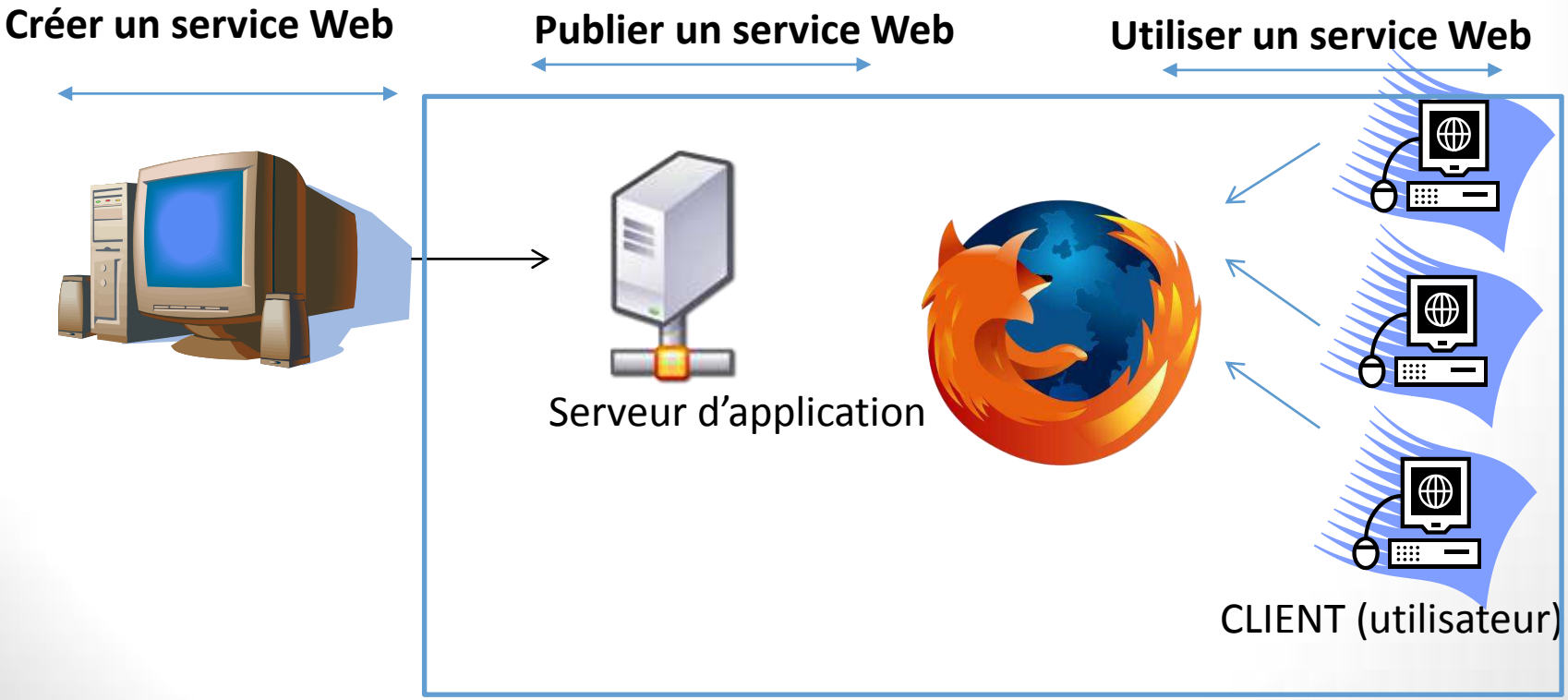
Architecture générale des services Web et des clients



Création de services web

La conception d'un service Web nécessite les étapes suivantes :

- Définir et créer un service Web
- Publier le service Web sur le serveur d'application
- Utiliser un service Web en créant un client.



Création de services web en Java

- Deux approches pour la création de SW:
  - Approche Bottom-up (ou code first): Création d'une classe Java→déploiement→ WSDL
  - Approche top-down (ou contract-first) : Développer un Service Web à partir de sa description WSDL.
- **Outils utilisés:**
  - Frameworks: Axis2 , Apache CXF (framework Java/Web Services) , Spring WS
  - API JAX-WS et JAX-RS
  - Etc.
- **Serveurs:** Tomcat, Glassfish, etc.

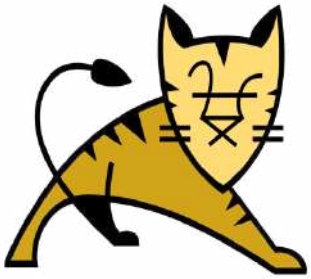
Axis

- **Axis** est un engin permettant la création des web services en technologie SOAP, réalisé par Apache Software Foundation.

  Offre :

  - un environnement pouvant fonctionner comme un serveur SOAP/Rest indépendant
  - une API pour développer des services web SOAP RPC ou à base de messages SOAP,
  - des outils pour créer automatiquement les WSDL correspondant à des classes Java, ou inversement, pour créer les classes Java sur la base d'un WSDL (classe proxy en quelque sorte, qui fait le lien entre l'application Java cliente et le service distant),
  - des outils pour déployer, tester et monitorer des web-services,
  - Etc…
- http://axis.apache.org/axis2/java/core/

Tomcat



- **Apache Tomcat** est un conteneur de servlet JEE. Issu du projet Jakarta, Tomcat est désormais un projet principal de la fondation Apache. Tomcat implémente les spécifications des servlets et des JSP de Sun Microsystems. Il inclut des outils pour la configuration et la gestion, mais peut également être configuré en éditant des fichiers de configuration XML. Comme Tomcat inclut un serveur HTTP interne, il est aussi considéré comme un serveur HTTP.

- http://tomcat.apache.org/

Glassfish

- **GlassFish** (Glassfish Enterprise Server) est un serveur d'applications Open Source de Sun et offre une implémentation complète de la norme Java EE (EJB, JPA, JAX-RS, servlets, JAX-WS etc.)

- Téléchargement:
  - https://glassfish.java.net/

Apache CXF

- Framework open Source en langage Java pour le développement des Services Web
- Inclut la norme JAX-WS, JAX-RS, etc. et le support des standards des Services Web e.g. SOAP, etc.
- Il supporte le mode « code-first », c'est-à-dire: il autorise que le développeur commence par l'implémentation du webservice, avant la création du contrat d'interface.

JAX-WS (Java API for XML WS)

- Java API for XML WS
- Ensemble d' APIs pour la programmation de services web sur JEE (incluant JAXB(Java Architecture for XML binding) et SAAJ)
- Utilisation simplifiée par des annotations dans les classes Java
  - Spécification de la correspondance XML - Objets
  - Facilitation de la programmation des points d' entrée de services

Annotations JAX-WS

- JAX-WS repose sur l'utilisation massive d'annotations pour la configuration d'un Service Web

- Les principales annotations sont les suivantes:
  - *@WebService :* Implémentation d'un Service Web
  - *@WebMethod :* Paramétrer une opération
  - *@WebParam :* Paramétrer un message
  - *@WebResult :* Paramétrer un message de sortie
  - *@WebFault :* Paramétrer un message fault

- A noter que seule l'utilisation de l'annotation @WebService est nécessaire (utilisation de valeurs par défaut)

Annotation @WebSERVICE

- Annote une classe Java pour définir l'implémentation du Service Web
- Annote une interface Java pour définir la description du Service Web
- Attributs de l'annotation @WebService
  - String name : nom du Service Web
  - String endpointInterface : nom de l'interface décrivant le Service Web
  - String portName : nom du port
  - String serviceName : nom du service du Service Web
  - String targetNamespace : le namespace du Service Web
  - String wsdlLocation : l'emplacement du WSDL décrivant le Service Web

Annotation @Webmethod

- Annote une méthode d'une classe Java exposée comme une opération du Service Web
- Attributs de l'annotation : @WebMethod
  - String action : l'action de l'opération. Dans le cas d'un binding SOAP, cela détermine la valeur de l'action SOAP
  - boolean exclude: précise que la méthode ne doit pas être exposée comme une opération. Ne pas utiliser dans une interface Java
  - String operationName : précise le nom de l'attribut name défini dans l'élément operation du document WSDL

Annotation @Webparam

- Décrit la relation entre un paramètre d'entrée d'une méthode et un message part d'une opération
- Attributs de l'annotation
  - boolean header : précise si le paramètre doit être transmis dans l'en-tête du message (true) ou dans le corps (false)
  - WebParam.Mode mode : précise le type d'accès au paramètre (IN, OUT ou INOUT)
  - String name : nom du paramètre
  - String partName : le nom du wsdl:part représentant ce paramètre
  - String targetNamespace : l'espace de nommage de ce paramètre

Annotation @Webresult

- Décrit la relation entre le paramètre de sortie d'une méthode et un message part d'une opération
- Attributs de l'annotation
  - boolean header : précise si le paramètre de sortie doit être transmis dans l'en-tête du message (true) ou dans le corps (false)
  - String name: nom du paramètre de sortie
  - String partName: le nom du wsdl:part représentant ce paramètre de sortie
  - String targetNamespace: l'espace de nommage de ce paramètre de sortie

Et concrètement ça donne quoi?

- Vous allez faire le TP qui est divisé en deux parties:

1. Tester des services web existants
2. Création et déploiement de services web
   - Java
   - L'IDE Netbeans
   - Le serveur d'applications Glassfish