

Services Web

Samir Tata

Département INFormatique

GET/INT

<http://www-inf.int-evry.fr/cours/WebServices>

Services Web

Origines

Définition et architecture

WSDL

SOAP

UDDI

Orientation Service : les origines

- Programmation orientée structure
 - procédures, fonctions et données structurées
 - applications monolithiques, applications serveurs isolées
- Programmation orientée objet
 - classes (fonctions et données structurées)
 - applications monolithiques, applications c/s isolées
- Programmation orientée composant
 - composant (groupe d'interfaces)
 - implémentation agnostique
 - interaction application-application fonctionne

Orientation Service : les origines

- POS, POO, POC : pourquoi on veut améliorer ?
 - Paradigmes ?
 - On parle du code et non de l'architecture
 - l'architecture suit le modèle de programmation
 - et alors ?
 - Concepts abstrait et mapping concret
 - mapping vers Java, C++, CORBA, COM ...
 - Réutilisation entre domaines différents difficile sinon impossible
- Besoins ?
 - Support de l'hétérogénéité : plusieurs plate-formes logicielles et matérielles
 - Accéder et manipuler des données partout

Orientation Service : les origines

- C'est quoi alors un service
 - un block avec des propriétés fonctionnelles et non fonctionnelles
 - Localisations, plate-forme et code sont peu important
- implications sur
 - le service qui peut être
 - destinataire d'un message (application)
 - manipulateur d'un message (infrastructure)
 - transporteur d'un message (protocole)
 - l'utilisation d'un service qui
 - est toujours supposée distante
 - passe toujours au dessus des plate-formes
 - ignore l'implémentation

Service Web : les origines

- Le Web d'aujourd'hui
 - conçu pour une interaction client-application
 - développé autour de 2 standards http et html
 - permet de supporter le e-commerce
- Le Web de demain ?
 - Le Web est partout. Que peut-on faire avec ?
 - les places de marché électronique (E-marketplaces)
 - intégration des procédés métiers
 - partage des ressources
 - Les approches existantes sont ad hoc
 - e.g. interactions application-application avec des formulaires html
 - But: permettre une interaction application-application

Services Web

Origines

Définition et architecture

WSDL

SOAP

UDDI

Service Web : définition

A Web service is a software application identified by a URI, whose interfaces and binding⁽¹⁾ are capable of being defined, described and discovered by XML artefacts and supports direct interactions with other software applications using XML based messages via Internet-based protocols. (W3C definition)

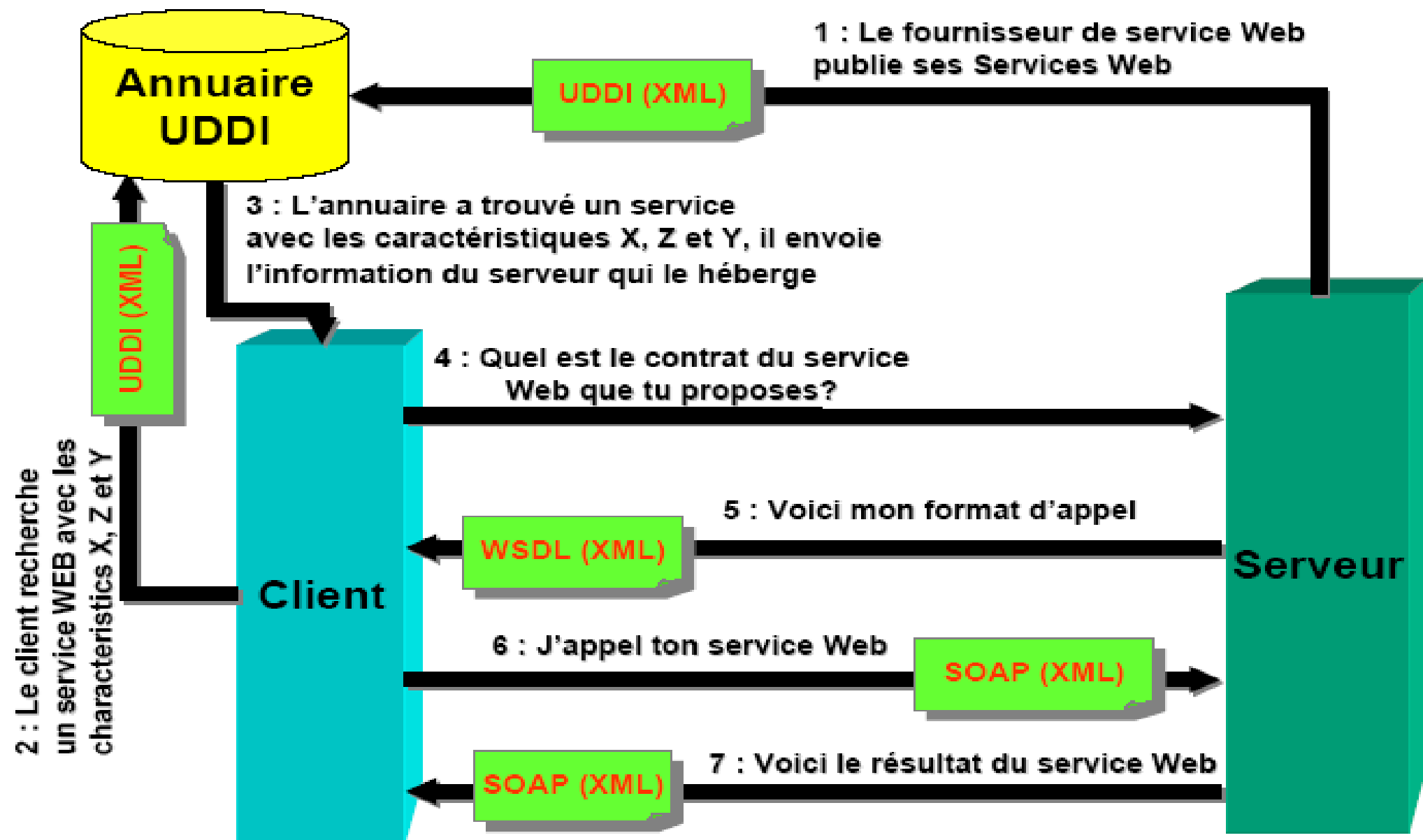
Un service Web est une application logicielle identifiée par un URI dont les interfaces et les liaisons sont définies, décrites et découvertes en XML et supporte une interaction directe avec les autres applications logicielles en utilisant des messages XML via un protocole Internet.

(1) An association between an Interface, a concrete protocol and a data format

Service Web : définition

- Les caractéristiques:
 - Les services Web sont réutilisables
 - Indépendamment de :
 - la plate-forme (UNIX, Windows,...)
 - leur implémentation (Java, C++, Visual Basic,...)
 - l'architecture sous-jacente (.NET, J2EE,...)

Architecture orientée service



Architecture orientée service

- Les partenaires
 - Le **fournisseur de service** crée le service Web, puis publie son interface ainsi que les informations d'accès au service, dans un annuaire de services Web.
 - L'**annuaire de service** rend disponible l'interface du service ainsi que ses informations d'accès, pour n'importe quel demandeur potentiel de service.
 - Le **demandeur de service** accède à l'annuaire de service pour effectuer une recherche afin de trouver les services désirés. Ensuite, il se lie au fournisseur pour invoquer le service.
- Les standards
 - Langage XML : Décrit les informations
 - Protocole SOAP : Exécute les services à distance
 - Langage WSDL : Décrit l'interface des services
 - Norme UDDI : Trouve les services dont on a besoin

Services Web

Origines

Définition et architecture

WSDL : Web Service description Language

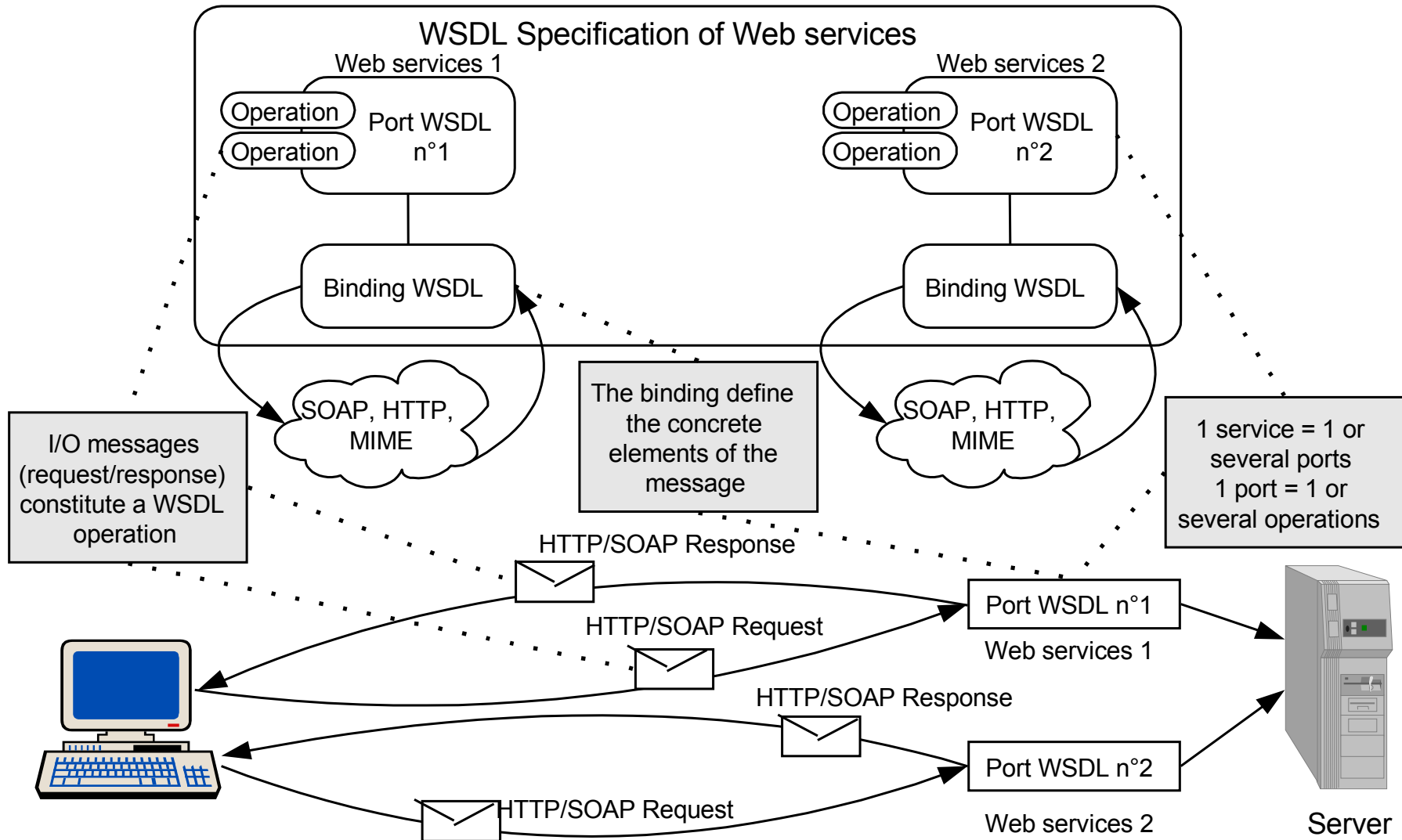
SOAP

UDDI

Rôle du langage WSDL

- WSDL est un langage qui permet de décrire :
 - un service Web,
 - comment invoquer un service Web
- Objectif :
 - Décrire les services comme un ensemble d'opérations et de messages abstraits relié à des protocoles et des serveurs réseaux
- Permet de décharger les utilisateurs des détails techniques de réalisation d'un appel
- WSDL est un langage qui standardise les schémas XML utilisés pour établir une connexion entre émetteurs et récepteurs.

Web service description using WSDL



Structure d'un document WSDL

```
<definitions>  
  <types>définition des types</types>  
  <message>définition des messages</message>  
  <portType>définition des opérations </portType>  
  <binding>définition des bindings</binding>  
  <services>définition de endpoint </endpoint>  
</definitions>
```

Exemple: annuaire (1/4)

Opérations :

- Ajouter une entrée:
 - Name : INT
 - Address :
 - city: Evry
 - number : 9
 - road : rue Charles Fourier
 - telephone:
 - code: 33
 - number : 1 60 76 00 00
- Chercher une entrée:
 - Entrée:
 - Name : INT
 - Sortie:
 - Address
- Classes:
 - Directory
 - Address
 - Phone

Example: annuaire (1/3)

```
package directory;
import java.util.Hashtable;
public class Directory {
    private Hashtable  entries;

    public Directory (){
        entries = new Hashtable();
    }

    public void addEntry (String name, Address address){
        entries.put(name, address);
    }

    public Address lookup(String name){
        return (Address) entries.get(name);
    }
}
```

Example: annuaire (2/3)

```
package directory;
public class Phone {
    int code ;
    String number ;

    public Phone (int codep, String numberp){
        code = codep;
        number = numberp;
    }

    public int getCode(){
        return code;
    }

    public void setCode(int codep){
        code = codep;
    }
    // ... getter and setter of number
}
```

Example: annuaire (3/3)

```
package directory;
public class Address {
    String road ;
    int number;
    String city;
    Phone telephone;

    public Address ( String roadp, int numberp, String cityp, Phone telephonep){
        road=roadp; number=numberp; city=cityp;
        telephone=new Phone(telephonep.getCode(), telephonep.getNumber());
    }

    public String getRoad(){
        return road;
    }

    public void setRoad(String roadp){
        road = roadp;
    }
    //... getter and setter of number, city, and telephone
}
```

WSDL : <definition>

<wsdl:definitions

```
targetNamespace="http://localhost:8081/axis/services/DirectoryService"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="http://localhost:8081/axis/services/DirectoryService"
xmlns:intf="http://localhost:8081/axis/services/DirectoryService"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns1="urn:DirectoryServiceTypes"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
<wsdl:types> </wsdl:types>
```

```
<wsdl:message> </wsdl:messages>
```

```
<wsdl:service> </wsdl:service>
```

```
<wsdl:portType> </wsdl:portType>
```

```
<wsdl:binding> </wsdl:binding>
```

</wsdl:definition>

WSDL : <types>

```
<wsdl:types>
  <schema targetNamespace="urn:DirectoryServiceTypes"
    xmlns="http://www.w3.org/2001/XMLSchema">
    <complexType name="Phone">
      <sequence>
        <element name="code" type="xsd:int"/>
        <element name="number" type="xsd:string"/>
      </sequence>
    </complexType>
    <complexType name="Address">
      <sequence>
        <element name="city" type="xsd:string"/>
        <element name="number" type="xsd:int"/>
        <element name="road" type="xsd:string"/>
        <element name="telephone" type="tns1:Phone"/>
      </sequence>
    </complexType>
  </schema>
</wsdl:types>
```

WSDL : <message>

```
<wsdl:message name="addEntryRequest">  
  <wsdl:part name="name" type="xsd:string"/>  
  <wsdl:part name="address" type="tns1:Address"/>  
</wsdl:message>
```

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

```
<wsdl:message name="lookupResponse">  
  <wsdl:part name="lookupReturn" type="tns1:Address"/>  
</wsdl:message>
```

WSDL : <porttype>

- Les types de port (port Type) décrivent un ensemble d'opérations. Peuvent être comparés à une interface Java
 - **One-way**
 - le service reçoit un message en entrée (<input>).
 - **Request-response**
 - le service reçoit un message en entrée (<input>) et renvoie un message en sortie <output> ou un ou plusieurs messages d'erreur (<fault>)
 - **Solicit-response**
 - Le service envoie un message (<output>) et reçoit un message corrélé (<input>) ou un ou plusieurs messages d'erreur (<fault>).
 - **Notification**
 - le service envoie un message de notification (<output>)

WSDL : <portType>

```
<wsdl:portType name="Directory">
```

```
  <wsdl:operation name="addEntry" parameterOrder="name address">
```

```
    <wsdl:input message="impl:addEntryRequest" name="addEntryRequest"/>
```

```
  </wsdl:operation>
```

```
  <wsdl:operation name="lookup" parameterOrder="name">
```

```
    <wsdl:input message="impl:lookupRequest" name="lookupRequest"/>
```

```
    <wsdl:output message="impl:lookupResponse" name="lookupResponse"/>
```

```
  </wsdl:operation>
```

```
</wsdl:portType>
```

WSDL : <binding>

```
<wsdl:binding name="DirectoryServiceSoapBinding" type="impl:Directory">
  <wsdlsoap:binding style="rpc"
    transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="addEntry">
      <wsdl:input name="addEntryRequest">
        <wsdlsoap:body
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          use="encoded"/>
        </wsdl:input>
      </wsdl:operation>

      <wsdl:operation name="lookup">
        <wsdl:input name="lookupRequest">
          <wsdlsoap:body ... />
        </wsdl:input>
        <wsdl:output name="lookupResponse">
          <wsdlsoap:body ... />
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
```

WSDL : <service>

- Le port est un point de terminaison identifié de manière unique par la combinaison d'une adresse Internet et d'une liaison
- Un Service est une collection de points d'entrée (endpoint) relatifs

```
<wsdl:service name="DirectoryService">  
  <wsdl:port  
    binding="impl:DirectoryServiceSoapBinding"  
    name="DirectoryService">  
    <wsdlsoap:address  
      location="http://inf-6820:8081/axis/services/DirectoryService"/>  
    </wsdl:port>  
  </wsdl:service>
```

WSDL : Conclusion

- Avantages
 - Description de services assez simple
 - Facile à interroger sur l'Internet
 - Extensible
- Inconvénient
 - Peu de fonctionnalités offertes

Services Web

Origines

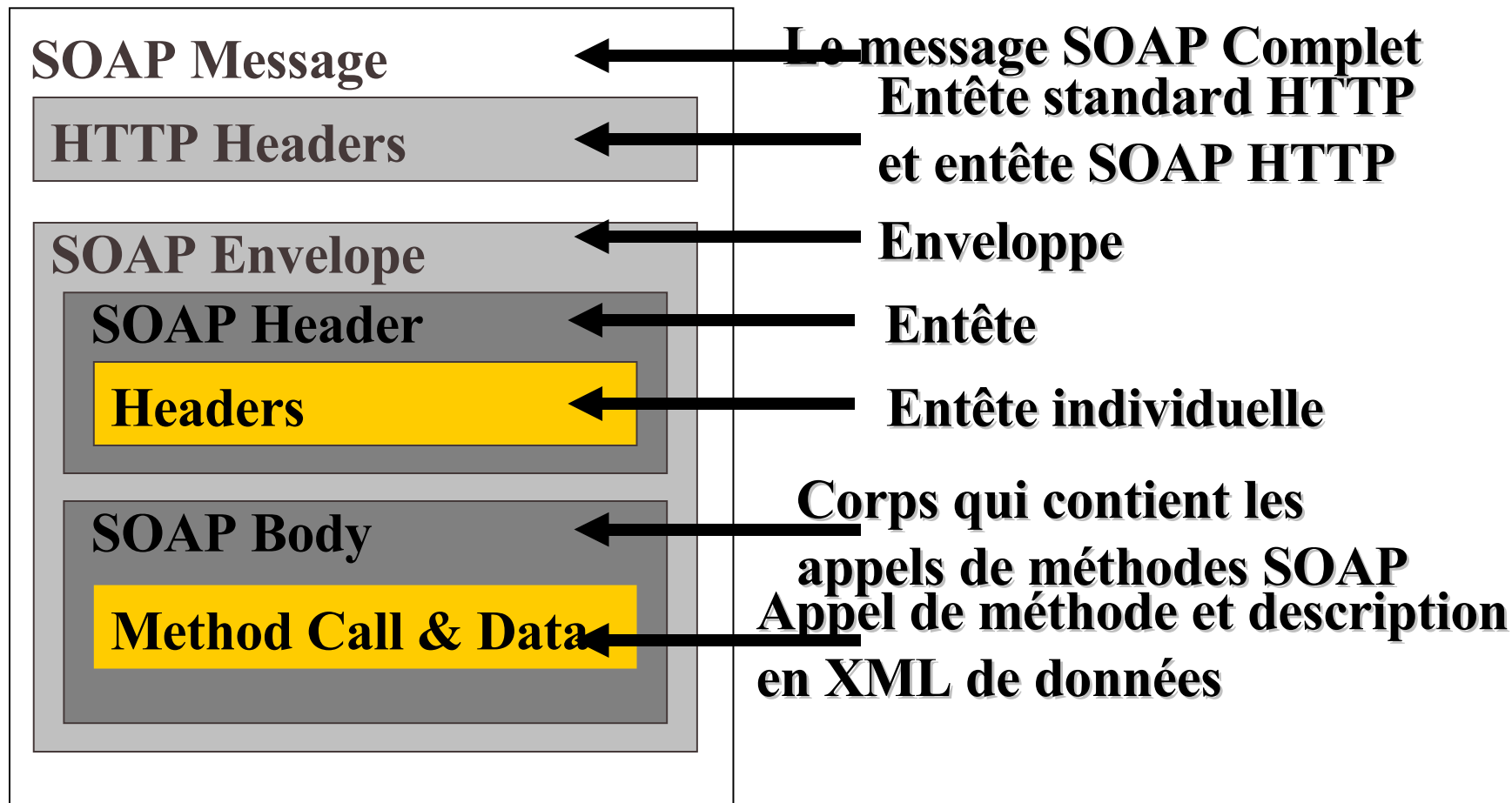
Définition et architecture

WSDL

SOAP: Simple Object Access Protocol

UDDI

Structure d'un message SOAP



SOAP : Message

```
<soap:Envelope  
  <soap:Header>  
    ... entête...  
  </soap:Header>  
  <soap:Body>  
    ... Corps du message...  
    <soap:Fault>  
      ... Erreurs éventuelles ...  
    </soap:Fault>  
  </soap:Body>  
</soap:Envelope>
```

L'enveloppe et l'entête

- L'enveloppe : définit le document XML comme étant un message SOAP

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/
encoding/">
</soap:Envelope>
```
- L'entête : contient des éléments définis par l'utilisateur

```
<soap:Header>
  <m:local xmlns:m="http://www.Computer.com/local/">
    <m:language>fr</m:language>
    <m:currency>Euro</m:currency>
  </m:local>
</soap:Header>
```

Le corps : Appel

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" >
  <soapenv:Body>
    <addEntry soapenv:encodingStyle="
      http://schemas.xmlsoap.org/soap/encoding/" >
      <name xsi:type="xsd:string">INT</name>
      <address href="#id0"/>
    </addEntry>
    <multiRef id="id0" >
      <city xsi:type="xsd:string">Evry</city>
      <number xsi:type="xsd:int">9</number>
      <road xsi:type="xsd:string">Rue Charles Fourier</road>
      <telephone href="#id1"/>
    </multiRef>
    <multiRef id="id1">
      <code xsi:type="xsd:int">33</code>
      <number xsi:type="xsd:string">160760000</number>
    </multiRef>
  </soapenv:Body>
</soapenv:Envelope>
```

Le corps : Appel

```
<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>
    <lookup
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" >
      <name xsi:type="xsd:string">INT</name>
    </lookup>
  </soapenv:Body>
</soapenv:Envelope>
```

Le corps : réponse

```
<soapenv:Envelope>
<soapenv:Body>
  <lookupResponse
    soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" >
    <lookupReturn href="#id0"/>
  </lookupResponse>
  <multiRef id="id0">
    <city xsi:type="xsd:string">Evry</city>
    <number xsi:type="xsd:int">9</number>
    <road xsi:type="xsd:string">Rue Charles Fourier</road>
    <telephone href="#id1"/>
  </multiRef>
  <multiRef id="id1">
    <code xsi:type="xsd:int">33</code>
    <number xsi:type="xsd:string">160760000</number>
  </multiRef>
</soapenv:Body>
</soapenv:Envelope>
```

Services Web

Origines

Définition et architecture

WSDL

SOAP

UDDI : Universal Description, Discovery and Integration

- UDDI

- Annuaire mondial d'entreprises basé sur le Web
- Il permet d'automatiser les communications entre prestataires, clients, etc..
- Très orienté « affaires » (vente, prestation)
- Accessible grâce à SOAP

- Structure de données

Intégrant toutes sortes d'entrées (nom, carte d'identité des sociétés, description des produits et des services, etc.),

- Qui : Le nom de l'entreprise, les contacts ...
- Quoi : Les classes, les noms des services
- Où : Les adresses d'accès aux services
- Comment : Les informations concernant les interfaces, les propriétés

Les pages de UDDI

- Pages blanches
 - Pour trouver un service par contact, nom et adresse
- Pages jaunes
 - Pour trouver un service par sujet (domaine) basé sur une taxonomie des services publiés dans l'annuaire
- Pages vertes
 - Pour trouver un service par ses caractéristiques techniques

Naviguer dans une base de services web

- <http://www.xmethods.net>
- Sélectionner un service
- Accéder au WSDL
- L'essayer

Références

- Web

- Site de W3C: www.w3.org
- Site de Zvon: <http://www.zvon.org/>
- Apache XML: <http://xml.apache.org/>
- Apache WS: <http://ws.apache.org/>

- Livres

- Georges Gardarin : Des bases de données aux services Web, Dunod, Paris, 2002
- Hubert Kadima et Valérie Monfort : Les services Web: techniques, démarches et outils, Dunod, Paris, 2003
- Gustavo Alonso, Fabio Casati, Harumi Kuno, and Vijay Machiraju : Web Services: Concepts, Architecture and Applications, Springer-Verlag, New York, 2004
- Jorge Cardoso and Amit P. Sheth : Semantic Web Services, Processes and Applications (Semantic Web and Beyond: Computing for Human Experience), Springer-Verlag, New York, 2006