



# Chapitre 3. Services Web (standard)

Les web services peuvent être classés en deux grandes catégories:

Web service standard (orienté SOAP)

Web service REST (orienté ressource)



# Chapitre 3. Services Web (standard)

**Web Services standard :**

- basé sur XML (Extensible Markup Language)
- basé sur le **protocol SOAP (Simple Object Access Protocol)** •

**Tendance actuelle “RESTful Web Services”**

- basé sur le **protocol REST (Representational State Transfer)** •

**HTTP (GET, PUT, POST, DELETE)**

Dans ce chapitre, nous nous intéressons aux web services standard.



# Plan

- Notions de base et définitions
- Exemples
- Cycle de vie d'utilisation
  
- SOAP
- WSDL
- UDDI
- Conclusion
  
- Exemples de création des services Web (à voir en TP)
  - Axis1
  - Axis2
  - JavaEE

Conclusion



# Notions de base et définitions

- Les web services sont des composants logiciels. Ils encapsulent des fonctionnalités. Ils sont accessibles via des protocoles standard du web.
- Un web service est une fonction du web qui prend des paramètres en entrée, communique et renvoi un résultat.
- Les fonctionnalités sont créées sur un Serveur et consommées par des clients (utilisateurs). Un web service est alors accessible depuis n'importe quelle plate-forme ou langage de programmation.



# Notions de base et définitions

Un Web service standard est basé essentiellement sur trois spécifications XML (SOAP, WSDL, UDDI)

- SOAP (): est le protocole de transport de données et l'infrastructure de communication
- WSDL (Web Service Description Language): est le langage de description des fonctionnalités offertes par les services.
- UDDI: est l'annuaire qui sert à référencer les services par les fournisseurs. Il est utilisé par les utilisateurs pour la découverte et la sélection des services web appropriés afin de répondre à leurs besoins .

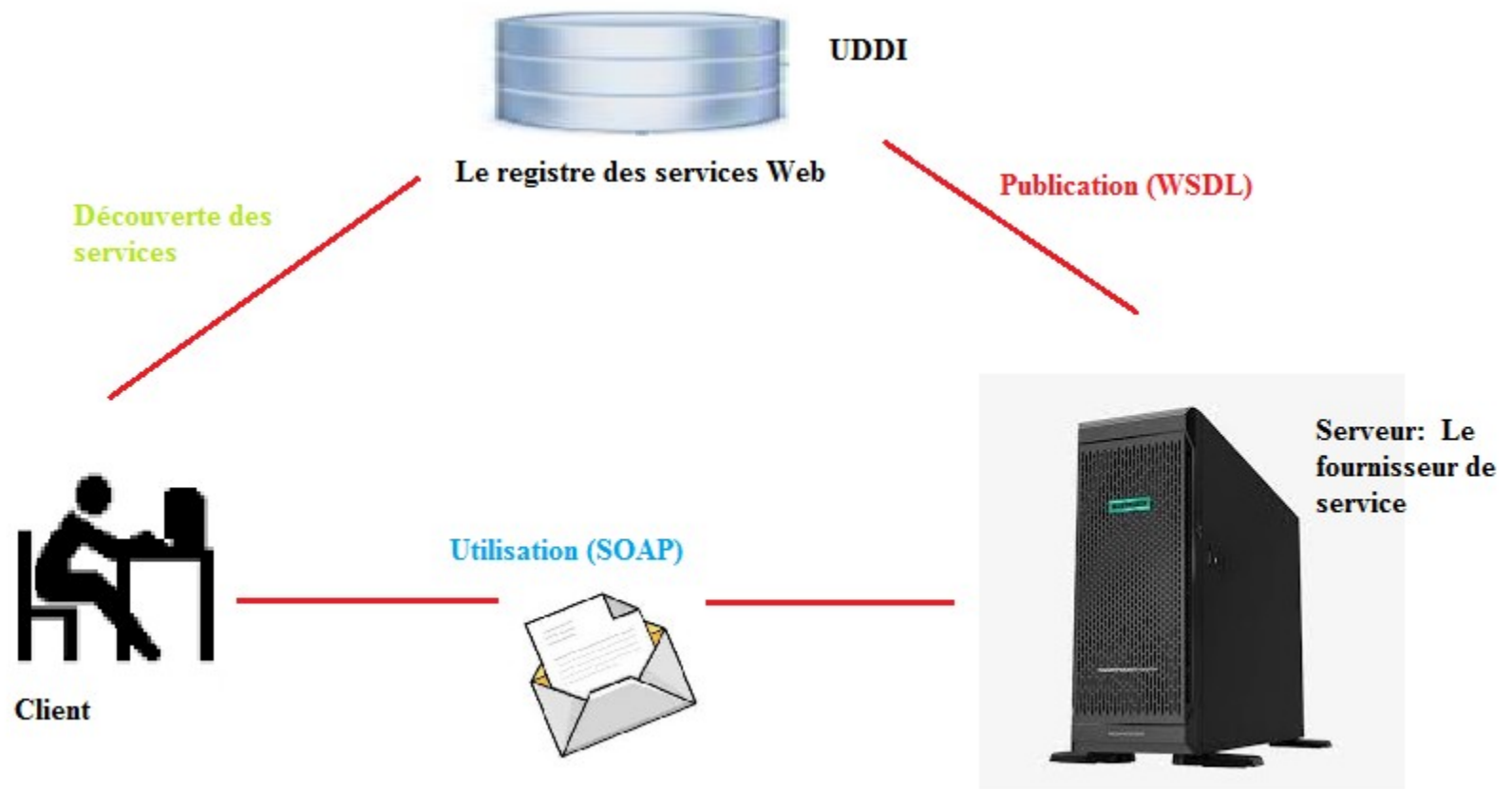


Schéma général d'un service web.



# Notions de base et définitions

Un web service standard est un ensemble de fonctionnalités décrit dans un document XML (dit WSDL). Le document WSDL décrit les méthodes à invoquer, leur signature et les points d'accès du service à savoir l'URL et le numéro de port où se trouve le service. Le SOAP est utilisé comme moyen de transport des requêtes échangées entre serveur/clients. La requête et sa réponse est contenu dans un document XML (dit SOAP) transporté par HTTP.



## **Exemple 1:**

Un service Web qui affiche la moyenne générale d'un étudiants

- le client demande la moyenne d'un étudiant en envoyant un message sur le web. Ce message contient le matricule de l'étudiant. Le Web Service reçoit le matricule, effectue une recherche et renvoi la moyenne générale au client via un autre message.





## Programmation en Java

```
Public class Monservice{  
    public double Moyenne(int Matricule)  
    {  
        Double moy = Etudiant [matricule];  
        return moy;  
    }  
}
```



## Exemple 2:

Un service Web sans paramètres qui affiche « bonjour tout le monde »

```
public class ServiceHelloWorld {  
    public String message() {  
        return "Hello World"; }  
}
```



## Exemple2: Web service calculatrice

### Programmation en Java

```
public class Calculatrice {  
    public float plus(float i1, float i2) {return i1 + i2;}  
    public float moins(float i1, float i2) {return i1 - i2;}  
    public float mult(float i1, float i2) {return i1 * i2;}  
    public float div(float i1, float i2) {return i1 / i2;}  
  
}
```



# Cycle de vie

Étape 1 : Création du service web

Étape 2: Déploiement du service web (ceci dépend de la plateforme)

Étape 3: Enregistrement du service web (description du service WSDL référentiels: sur machine (locale), UDDI (Internet))

Étape 4: Découverte du service web

Étape 5 Consommation (Invocation) du service par le client



# SOAP

Simple Object Access Protocol



# SOAP

Protocole d'échange de messages basé sur XML,  
Porté sur HTTP, SMTP

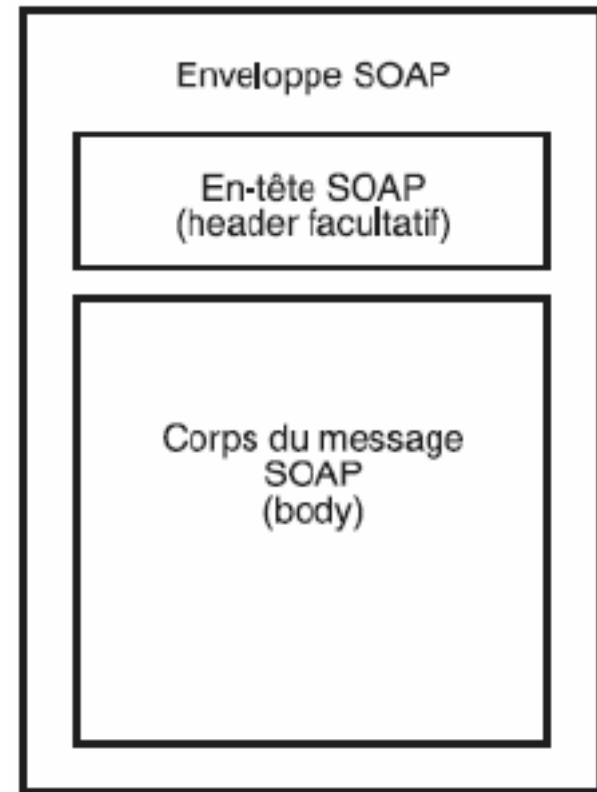
- Concepts :

Message = Enveloppe ( Header + Body )

# SOAP

## *structure de l'enveloppe SOAP*

Un message SOAP est un document XML, il est constitué d'une enveloppe contenant un entête (*header*) facultatif et le corps (*body*) du message





## **L'enveloppe SOAP : <SOAP-ENV:Envelope> :**

L'enveloppe est la racine du document XML contenant le message SOAP,

- l'espace de nommage (*Namespace*) définissant la version de SOAP utilisée, et le type d'encodage.

SOAP définit deux Namespaces fréquemment utilisés :

SOAP-ENV associé à l'URI *<http://schemas.xmlsoap.org/soap/envelope>* pour définir le namespace de l'enveloppe dans la version 1.1, et à *<http://www.w3.org/2001/06/soapenvelope>*

dans la version 1.2 reprise par le W3C.

SOAP-ENC associé à l'URI *<http://schemas.xmlsoap.org/soap/encoding>* pour la définition des formats de types de données dans la version 1.1, et à *<http://www.w3.org/2001/06/soapencoding>* dans la version 1.2.



## L'enveloppe SOAP :

L'enveloppe contient le *Namespace* définissant la version de SOAP utilisée, et les règles de sérialisation, et d'encodage.

## L'entête SOAP :

- Partie du message est optionnelle.
- contient des informations liées aux transactions, à l'authentification,
- sert à transmettre des informations nécessaires pour l'exécution de la requête SOAP

## Le Corps SOAP :

- constitué du seul élément **Body**, contenant un ou plusieurs sous-éléments.

Ces sous-éléments sont les suivants :

- **FAULT** : une erreur ou une défaillance en réponse à une requête.
- Des données destinées au destinataire du message avec un format défini par les règles de codage SOAP.

POST /Computer HTTP/1.1  
Host: bat710.univ-lyon1.fr  
Content-Type: text/xml;  
Charset = "utf-8"  
Content-Length: 10  
SOAPAction:

<SOAP-ENV:Envelope  
xmlns:SOAP-ENV = "http://www.w3.org/2001/06/soap-envelope"  
SOAP-ENV:encodingStyle = "http://www.w3.org/2001/06/soap-encoding"/>

<SOAP-ENV:Header>  
    // contenu du header  
</SOAP-ENV:Header>

<SOAP-ENV:Body>  
    // contenu du body  
</SOAP-ENV:Body>

# SOAP : Exemple

```
public class Echo {  
    public String echo(String expression) {  
        return "message reçu:" + expression ;  
    }  
}
```

# Requête SOAP

```
<SOAP-ENV : Body>
  <m:echo xmlns:m = "urn:Echo">
    <expression type= "xsd:string">
      Hello World
    </expression>
  </m:echo>
</SOAP-ENV : Body>
```

- `<m:echo xmlns:m="urn:Echo">` identification du service (« urn :Echo »).  
Le nom de l'élément (echo) correspond au nom de la méthode à invoquer;
- Les paramètres d'appel de la méthode : seul paramètre : « expression », de type string.

# SOAP réponse

```
<SOAP-ENV : Body>
  <m : echo xmlns : m = "urn : Echo">
    <return type="string">
      message reçu : Hello World
    </return>
  </m:echo>
</SOAP-ENV : Body>
```

Le résultat est dans un élément `<return>` :

- L'attribut `type` précise le type de retour de la méthode exécutée.
- Le contenu de l'élément est le résultat de l'exécution de la méthode.

# SOAP : gestion des erreurs

```
<SOAP-ENV : Body>
  <SOAP-ENV: Fault>
    <Faultcode> SOAP-ENV : Client </Faultcode>
    <Faultstring>
      La signature de la méthode ne correspond pas.
    </Faultstring>
  </SOAP-ENV : Fault>
</SOAP-ENV : Body>
```

Signaler des cas d'erreur.

- faultcode : un code permettant d'identifier le type d'erreur
  - Client, Server, VersionMismatch, MustUnderstand
- faultstring : une explication en langage naturel
- faultactor : une information identifier l'initiateur de l'erreur
- detail : Définition précise de l'erreur.



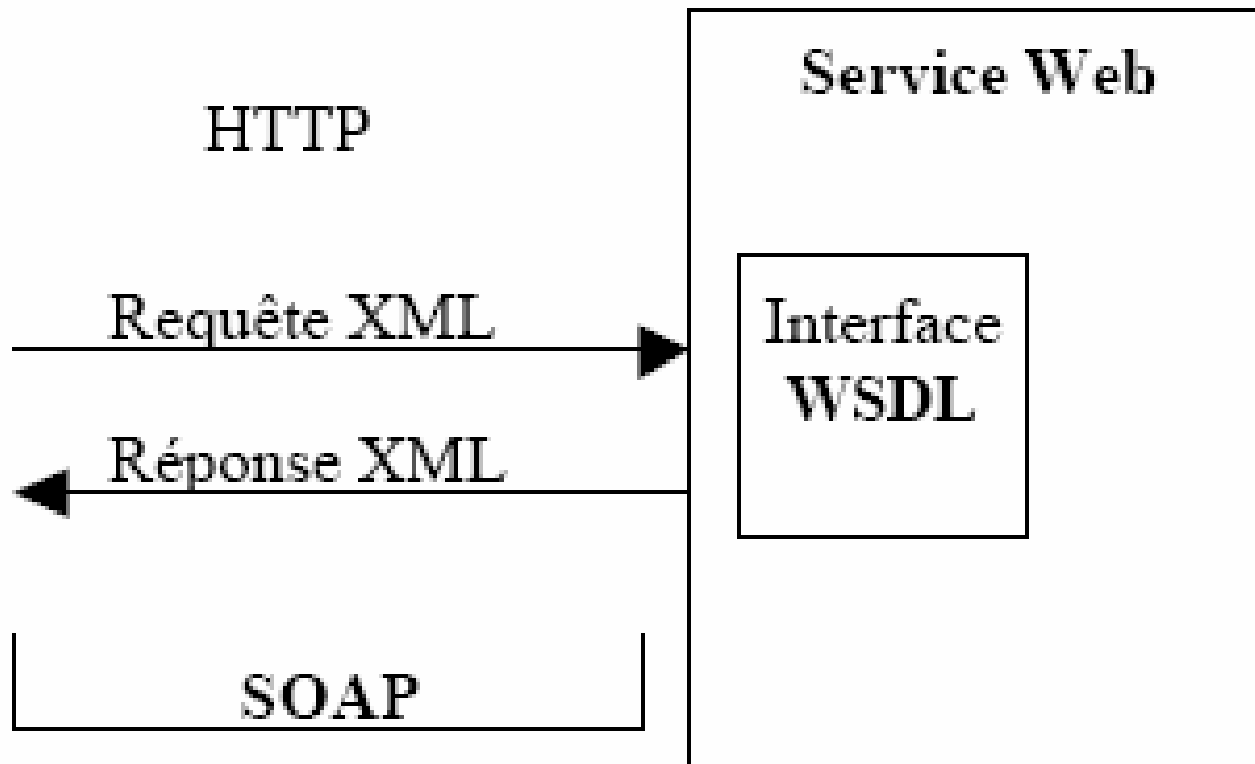
# WSDL

## ■ Langage de définition de Web Services

Un langage qui permet de décrire de façon précise les Web Services, en incluant des détails :

Les protocoles, les serveurs, les ports utilisés, les opérations qui peuvent être effectuées, les formats des messages d'entrée et de sortie, ect

# WSDL



WSDL est le point d'entrée de n'importe quel Web Service,  
On trouve dans WSDL : la localisation du service, et les opérations à invoquer en utilisant le SOAP.



# WSDL

## *Structure d'un document WSDL*

```
<definitions>
  <message>
    ...
  </message>
  <portType>
    <operation> ... </operation>
    <operation> ... </operation>
    ...
  </portType>
  <binding>
    ...
  </binding>
  <service>
    <port> ... </port>
    <port> ... </port>
  </service>
</definitions>
```



**<definitions>** : la définition du service.

## *Structure d'un document WSDL*

Cet élément est la racine de tout document WSDL.

- Il contient les attributs précisant le nom du service, et les *Namespaces*.

- **<message>** et **<portType>** : Ces éléments définissent les opérations offertes par le service, leurs paramètres d'entrée et de sortie, etc.

**<message>** : un paramètre d'entrée ou de sortie d'une **<operation>**.

**<portType>** définit un ensemble d'opérations.

**<operation>** un couple message entrée / message sortie.

Une opération : est une méthode

Un portType : une interface.



## *Structure d'un document WSDL*

- **<binding>** : le lien entre les **<portType>** et le protocole SOAP.
- **<service>** : il spécifie les informations complémentaires nécessaires pour invoquer le service,

À savoir : l'URI du destinataire.

Un **<service>** est modélisé comme une collection de ports, un **<port>** étant l'association d'un **<binding>** à un URI

## Exemple

```
public class Echo {  
    public String echo( String message ) {  
        return "message reçu : " + message ;  
    }  
}
```

1. *<definition*

2. xmlns : xsd = "http://www.w3.org/1999/XMLSchema" ← les *Namespaces*

3. xmlns : soap = "http://schemas.xmlsoap.org/wsdl/soap/">

4. *<message* name = "echoInput">

5. *<part* name = "expression" type = "xsd:string"/>

6. *</message>*

7. *<message* name = "echoOutput">

8. *<part* name = "expression" type = "xsd:string"/>

9. *</message>*

Paramètres d'entrée et de sortie de la méthode **echo** du service : **echoInput** et **echoOutput**.

10. *<portType* name = "EchoPortType">

11. *<operation* name = "echo">

13. *<input* message = "echoInput"/>

14. *<output* message = "echoOutput"/>

15. *</operation>*

16. *</portType>*

**PortType**, l'interface du service qui encapsule la définition de l'opération **echo**. faire référence aux (paramètres d'entrée et de sortie de l'opération). On obtient une description abstraite du service, indépendante de tout protocole de communication.

17. <**binding** name = "EchoSoapBinding" type = "tns:EchoPortType">

18. <soap:binding style = "document"

19. transport = "http://schemas.xmlsoap.org/soap/http"/>

20. <**operation** name = "echo">

21. <soap:operation soapAction = "urn:ServiceEcho"/>

22. <**input**>

24. <soap:body use = "encoded"

25. encodingStyle =

26. "http://schemas.xmlsoap.org/soap/encoding"/>

27. </**input**>

28. <**output**>

29. <soap:body use = "encoded"

30. encodingStyle =

31. "http://schemas.xmlsoap.org/soap/encoding"/>


32. </**output**>

33. </**operation**>

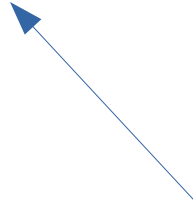
33. </**binding**>

Associer le service au  
protocole SOAP

Le binding qui définit  
les paramètres  
d'invocation du service  
spécifiques au  
protocole SOAP  
utilisé.



```
34 <service name = "EchoService">
35. <port name = "EchoSoap" binding = "tns:EchoSoapBinding">
36. <soap:address
37. location = "http://www.improve.fr/ServiceEcho"/>
38. </port>
39. </service>
40 .</definition>
```



définir l'adresse URL du service à invoquer.



# **UDDI**

**Universal Description Discovery and Integration**





# UDDI

**Standard proposé en septembre 2000 à l'instigation des sociétés Ariba, IBM et Microsoft, visant à établir un format d'annuaire des Web Services.**

<http://www.uddi.org>



# **UDDI**

**UDDI est composé de deux parties :**

- **L'UDDI Business Registry : annuaire d'entreprises et de Web Services;**
- **Les interfaces d'accès à ces annuaires, et les modèles de données.**



# **UDDI** *Business Registry*

- Pages blanches
- Pages jaunes
- Pages vertes



# **UDDI** *Business Registry*

*Pages blanches :*

**Noms, adresses, contacts, identifiants,... des entreprises enregistrées.**

**Ces informations sont décrites dans des entités de type**

***Business Entity.***



# **UDDI** *Business Registry*

*. Pages jaunes :*

**Détails sur le métier de l'entreprise, les services qu'elle propose.**

**Ces informations sont décrites dans des entités de type**

***Business Service ;***

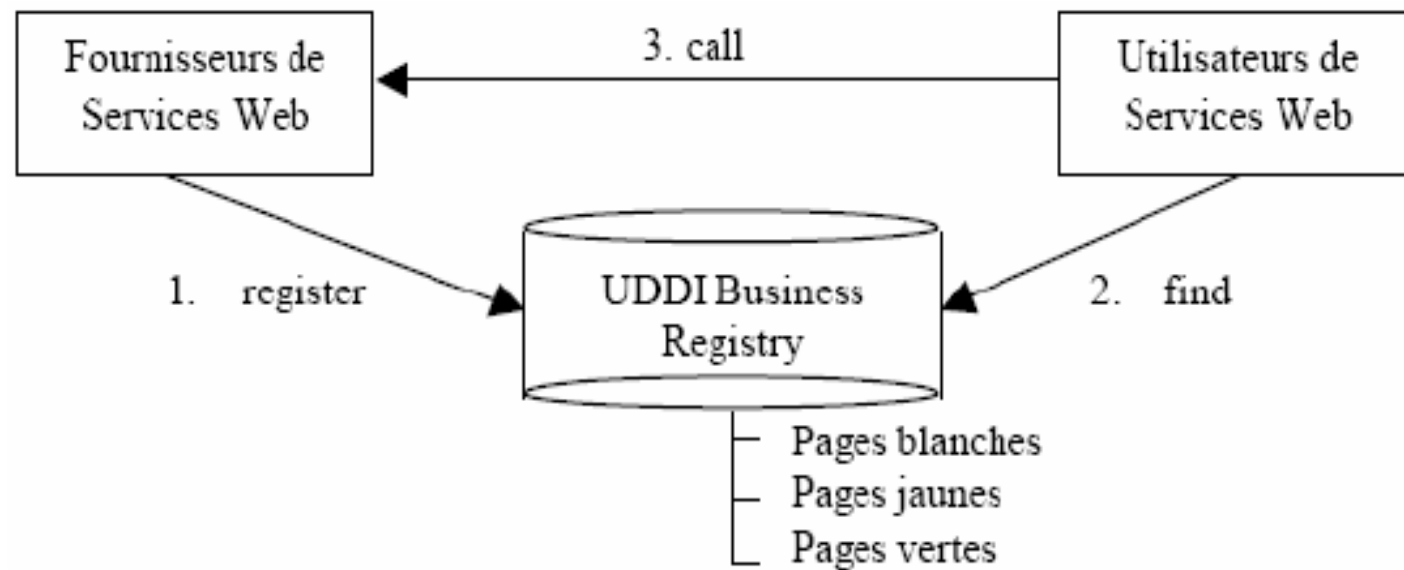


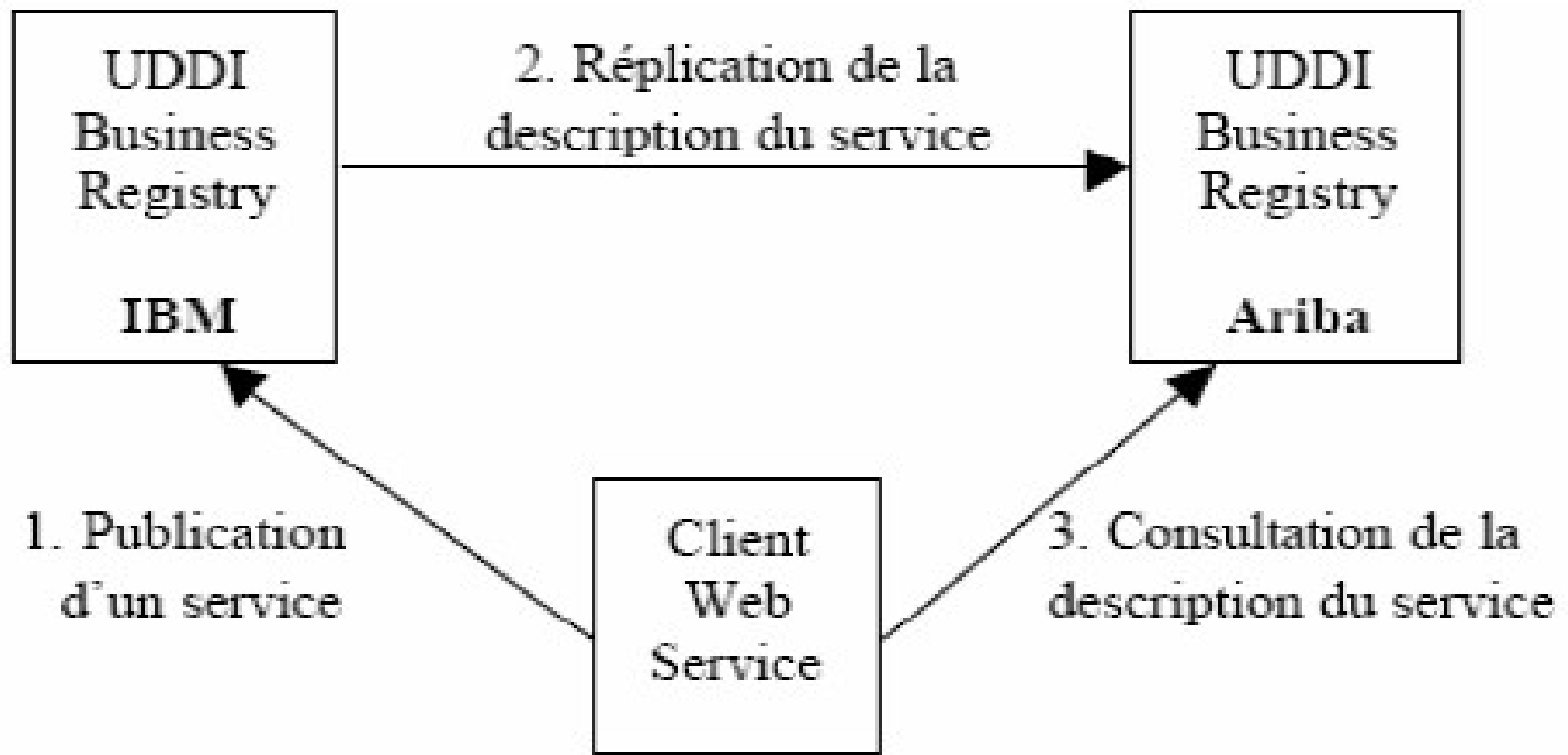
# **UDDI** *Business Registry*

**Pages vertes :**

**Informations techniques sur les services proposés.  
incluent des références vers les spécifications des Web  
Services,**

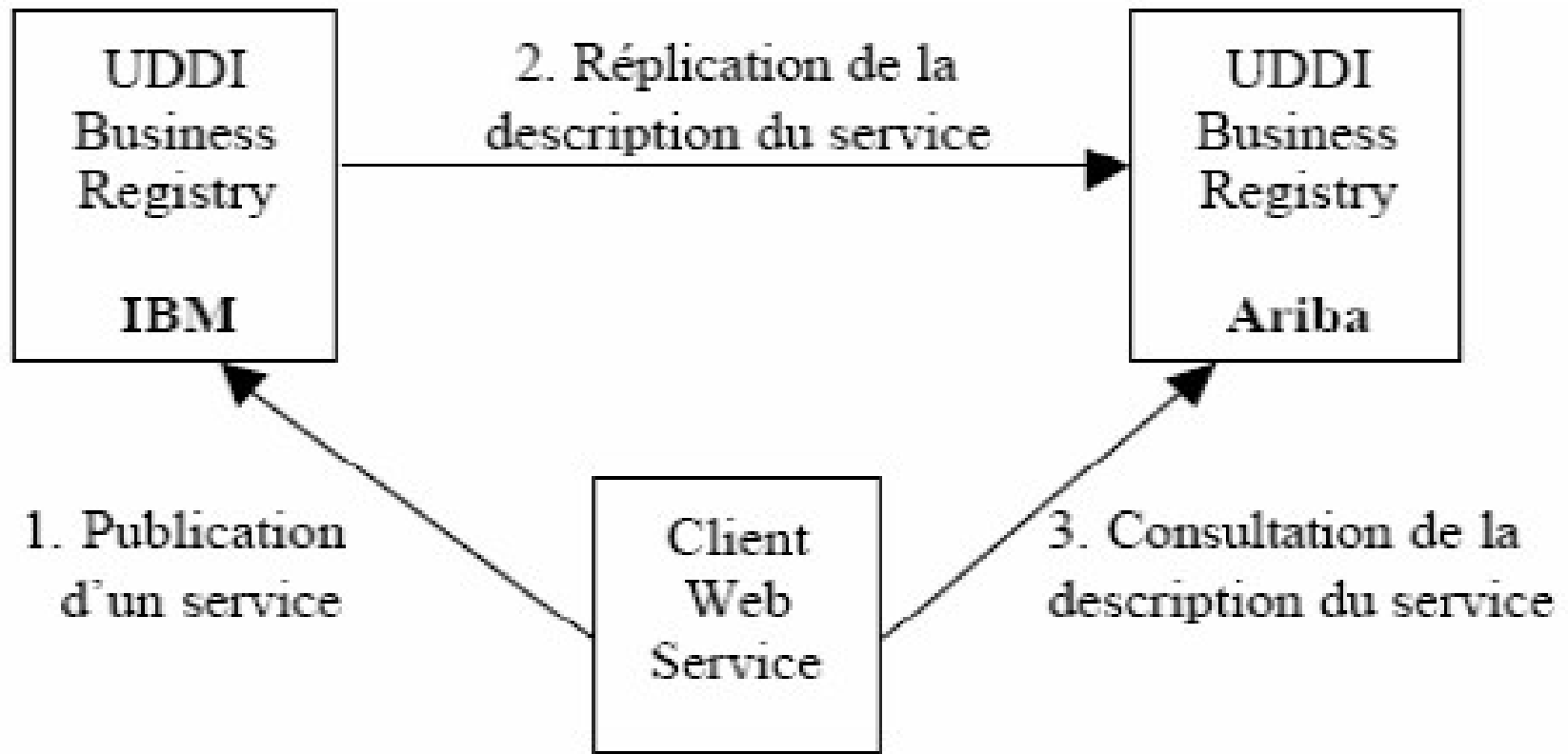
Ces informations sont  
décrites dans deux documents :  
un *Binding Template*, et un *Technology Model (tModel)*.



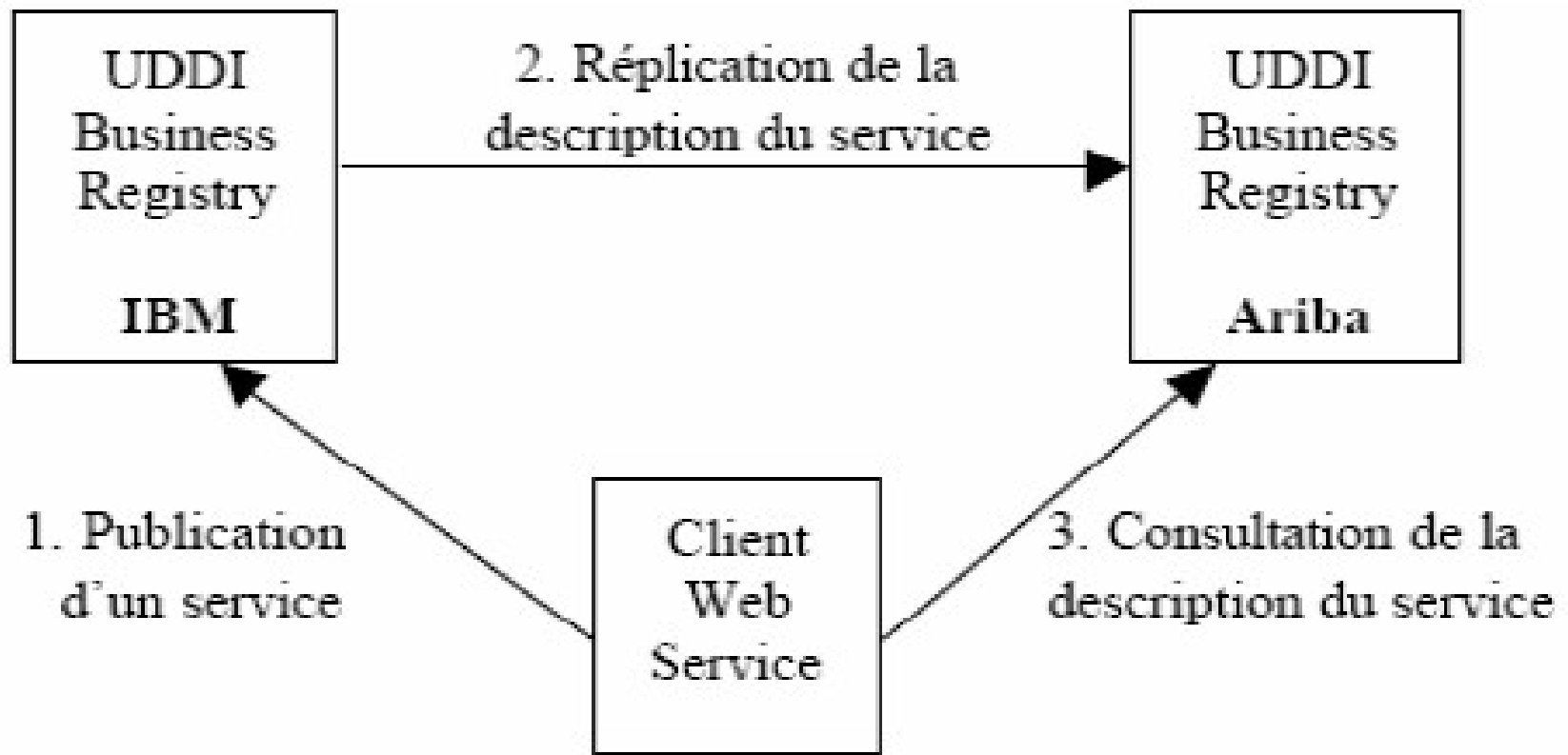


- 1- Les fournisseurs des services web s'enregistrent auprès de l'UDDI *Business Registry* (sur les pages blanches et jaunes)
- ajoutent les services en complétant les pages jaunes et en renseignant les détails techniques sur ces services dans les pages vertes.

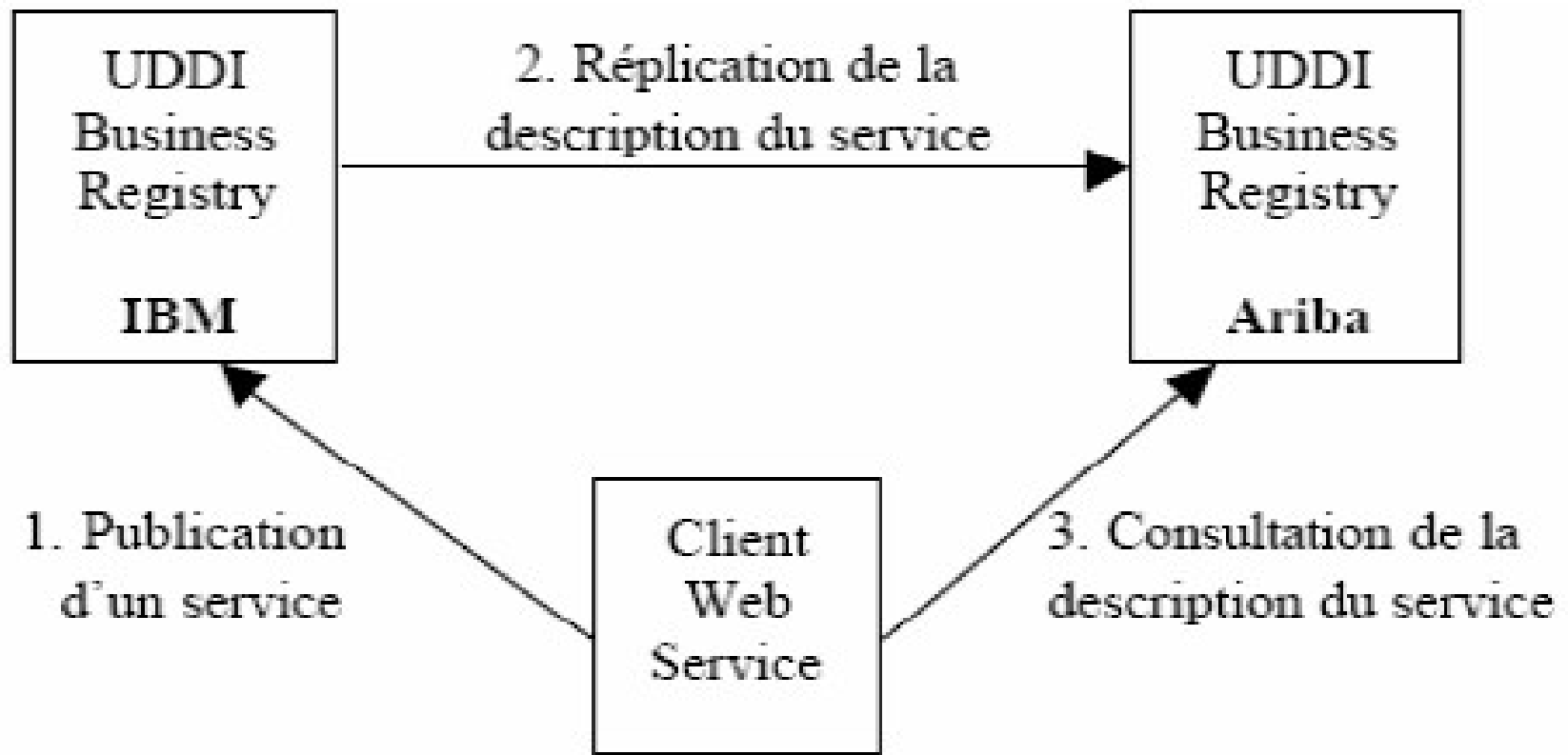




- Un seul enregistrement d'un service se fait sur référentiel (UDDI),
- UDDI est un service distribué sur Internet,
- Tous les changements sont répercutés sur les autres référentiels



2 - le client recherche une entreprise fournissant un service donné, Il récupère la description de l'entreprise via les pages blanches et jaunes, et des détails de l'invocation du service offert via les pages vertes,



3- Le client peut alors invoquer le Web Service distant en utilisant SOAP.



**FIN du chapitre**



# Références

- SOAP : <http://www.w3.org/TR/SOAP/>
- WSDL :  
<http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
- UDDI : <http://www.uddi.org/>
- Apache SOAP : <http://xml.apache.org/soap/>