

# Service Web (SOAP)

---

Urbanisation des SI – NFE107

Fiche de lecture

Y. BELAID

# Plan

---

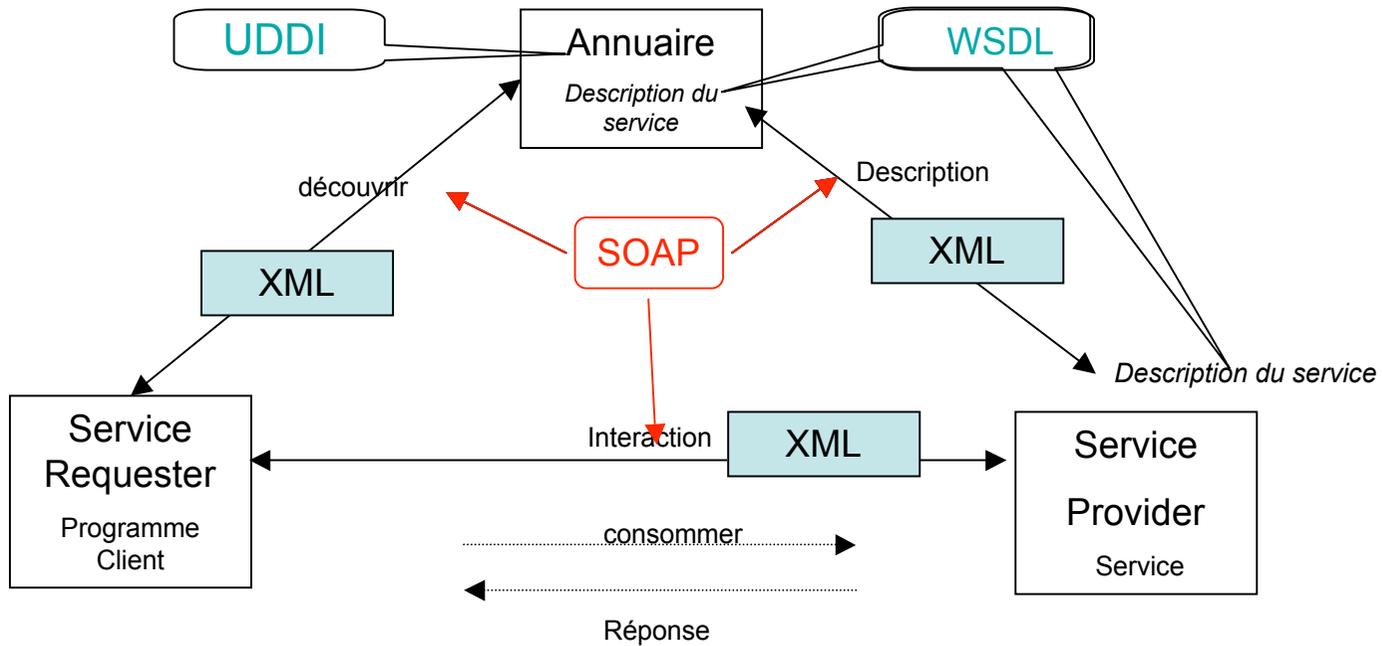
- Définitions
  - Web Service
  - Terminologie
- Communication avec les Web Service
  - REST
  - XML-RPC
  - SOAP

# Qu'est ce qu'un Web Service

---

- Un Web Service est un composant logiciel identifié par une URI, dont les interfaces publiques sont définies et appelées en XML.  
Sa définition peut être découverte par d'autres systèmes logiciels.  
Les services Web peuvent interagir entre eux d'une manière prescrite par leurs définitions, en utilisant des messages XML portés par les protocoles Internet.  
(W3C)
- Une technologie permettant à des applications de dialoguer à distance via Internet indépendamment des plates-formes et des langages sur lesquelles elles reposent.  
(définition : <http://www.dicodunet.com/>)
- Un service web est un programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, et en temps réel.  
(définition : [Wikipédia](#))

# Architecture des Web Service



# Architecture des Web Service (2)

---

- Les acteurs
  - Annuaire – Service Registry
    - Annuaire des services publiés par les providers (UDDI)
    - Géré sur un serveur niveau application, entreprise ou mondial
  - Service Provider
    - Application s'exécutant sur un serveur et comportant un module logiciel accessible par en XML
  - Service Requester
    - Application cliente se liant à un service et invoquant ses fonctions par des messages XML (REST, XML-RPC, SOAP)

# Architecture des Web Services

---

- Terminologie

- WSDL (Web Services Description Language) donne la description au format XML des Web Services en précisant les méthodes pouvant être invoquées, leur signature et le point d'accès (URL, port, etc..).
  - **Dialecte XML permettant de décrire un web service**
- UDDI (Universal Description, Discovery and Integration) normalise une solution d'annuaire distribué de Web Services, permettant à la fois la publication et l'exploration. UDDI se comporte lui-même comme un Web service dont les méthodes sont appelées via le protocole SOAP.
  - **Annuaire permettant d'enregistrer de rechercher des service web**
- SOAP (Simple Object Access Protocol) : Protocole de communication en service Web par échange de message XML.

# Communication Web Service

---

- Les différentes technologies pour accéder à un service web
  - REST (Représentational State Transfer)
    - Consommer un Webservice REST revient à appeler une simple URL en http (Post ou Get) , le serveur renvoie sa réponse, la plupart du temps en XML
  - Communication par échange de fichier XML
    - XML-RPC
    - SOAP (Simple Object Access Protocol)

# REST (Representation State Transfer)

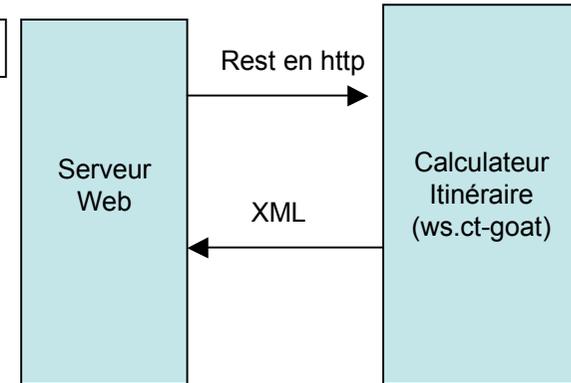
**Exemple :** Récupération des Informations d'une commune par une requête http en mode GET

## Requête

```
http://ws.ct-goat.com/getCityInfos.asp ?uid=xxxxxxxxxxxx&comID=562
```

## Réponse

```
<RETURN>
  <ERROR>
    <NUMBER>[numéro d'erreur]</NUMBER>
    <DESCRIPTION>[description de l'erreur]</DESCRIPTION>
  </ERROR>
  <RESULT>
    <ROWS>
      <ROWCOUNT>1</ROWCOUNT>
      <ROW>
        <COM_ID>[ID de la commune]</COM_ID>
        <COM_NAME>[nom de la commune]</COM_NAME>
        <COM_COMINSEE>[code INSEE de la commune]</COM_COMINSEE>
        <COM_PAE>[Id des points d'arrêts principaux de la commune]</COM_COMINSEE>
      </ROW>
    </ROWS>
  </RESULT>
</RETURN>
```

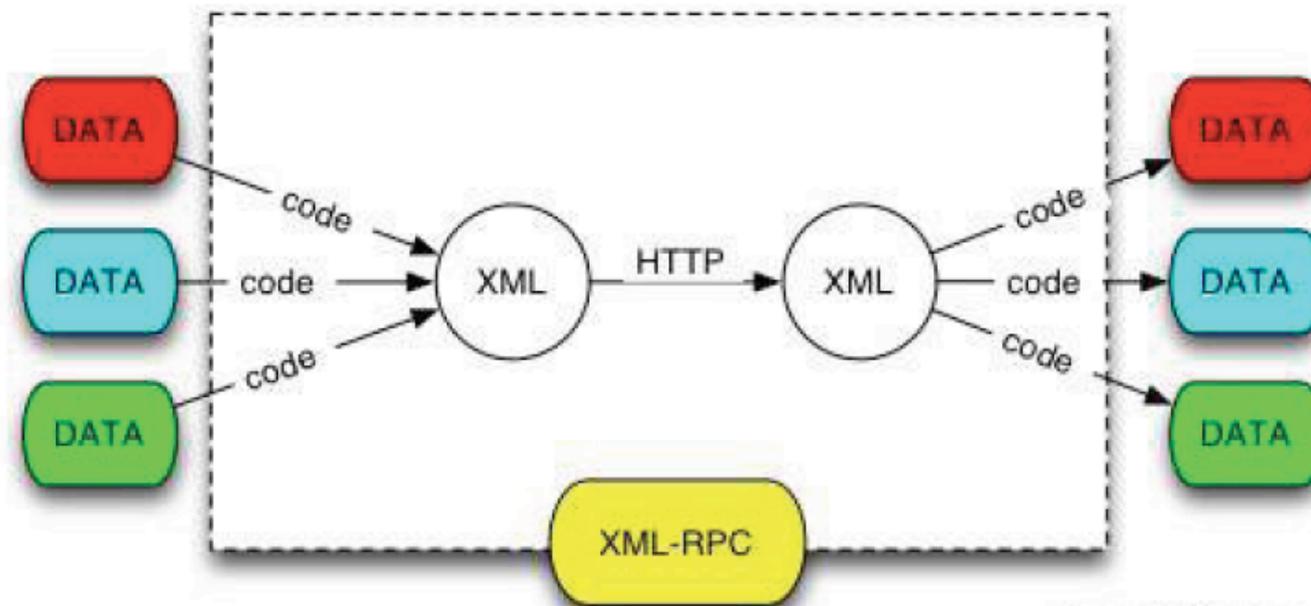


# REST (Representation State Transfer)

---

- N'est pas un protocole ou un format, mais une architecture (celle de HTTP).
- Chaque 'méthode' ou 'service' est attaché à une URL
- Consommer un Webservice REST revient à appeler une simple URL en http (Post ou Get) , le serveur renvoie sa réponse, la plupart du temps en XML
- Très utilisé pour la communication entre machines.

# XML-RPC



Source: JY Stervinou

Ci dessus le schéma représentatif du XMLRPC.

Source : <http://opengescom.org/docs/spec-xmlrpc.pdf>

# XML-RPC

## Requête

```
POST /RPC2 HTTP/1.0
User-Agent: opengescom (Linux)
Host: www.opengescom.org
Content-Type: text/xml
Content-length: 181

<?xml version="1.0"?>
<methodCall>
  <methodName>opgc.requestNewPro</methodName>
  <params>
    <param>
      <value><i4>1</i4></value>
    </param>
  </params>
</methodCall>
```

```
<struct>
  <member>
    <name>minimum</name>
    <value><i4>18</i4></value>
  </member>
  <member>
    <name>maximum</name>
    <value><i4>139</i4></value>
  </member>
</struct>
```

```
<array>
  <data>
    <value><i4>12</i4></value>
    <value><string>Alencon</string></value>
    <value><boolean>0</boolean></value>
    <value><i4>-31</i4></value>
  </data>
</array>
```

- Le message d'une requête est inclus dans un élément <methodCall >
- Le message d'une réponse est inclus dans une élément <methodResponse>
- L'élément <value> est transmis dans un élément <param>
- Le type des valeurs peut être défini avec des éléments de type <string> <double>

....

# XML-RPC

## Réponse :

Réponse Correct

```
HTTP/1.1 200 OK
Connection: close
Content-Length: 158
Content-Type: text/xml
Date: Tue, 22 Feb 2005 18:30:08 GMT
Server: OpenGesCom/1.0.1-Linux Debian

<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <struct>
        <member>
          <name>code produit</name>
          <value><string>AB000010</string></value>
        </member>
        <member>
          <name>code barre</name>
          <value><string>3270190113508</string></value>
        </member>
      </struct>
    </param>
  </params>
</methodResponse>
```

Réponse en erreur

```
HTTP/1.1 200 OK
Connection: close
Content-Length: 426
Content-Type: text/xml
Date: Tue, 22 Feb 2005 18:30:08 GMT
Server: OpenGesCom/1.0.1-Linux Debian

<?xml version="1.0"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value><int>4</int></value>
        </member>
        <member>
          <name>faultString</name>
          <value><string>Too many parameters.</string></value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>
```

# XML-RPC

- XML-RPC est un protocole RPC (Remote procedure call), une spécification simple et un ensemble de codes qui permettent à des processus s'exécutant dans des environnements différents de faire des appels de méthodes à travers un réseau.
- XML-RPC permet d'appeler une fonction sur un serveur distant à partir de n'importe quel système (Windows, MacOSX, Linux) et avec n'importe quel langage de programmation. Le serveur est lui-même sur n'importe quel système et est programmé dans n'importe quel langage.
- Cela permet de fournir un Service Web utilisable par tout le monde sans restriction de système ou de langage.
- Les processus d'invocation à distance utilisent le protocole HTTP pour le transport des données et la norme XML pour le codage des données.
- XML-RPC est conçu pour permettre à des structures de données complexes d'être transmises, exécutées et renvoyées très facilement.
- XML-RPC est l'ancêtre de SOAP.

(définition : Wikipédia)

# XML-RPC

---

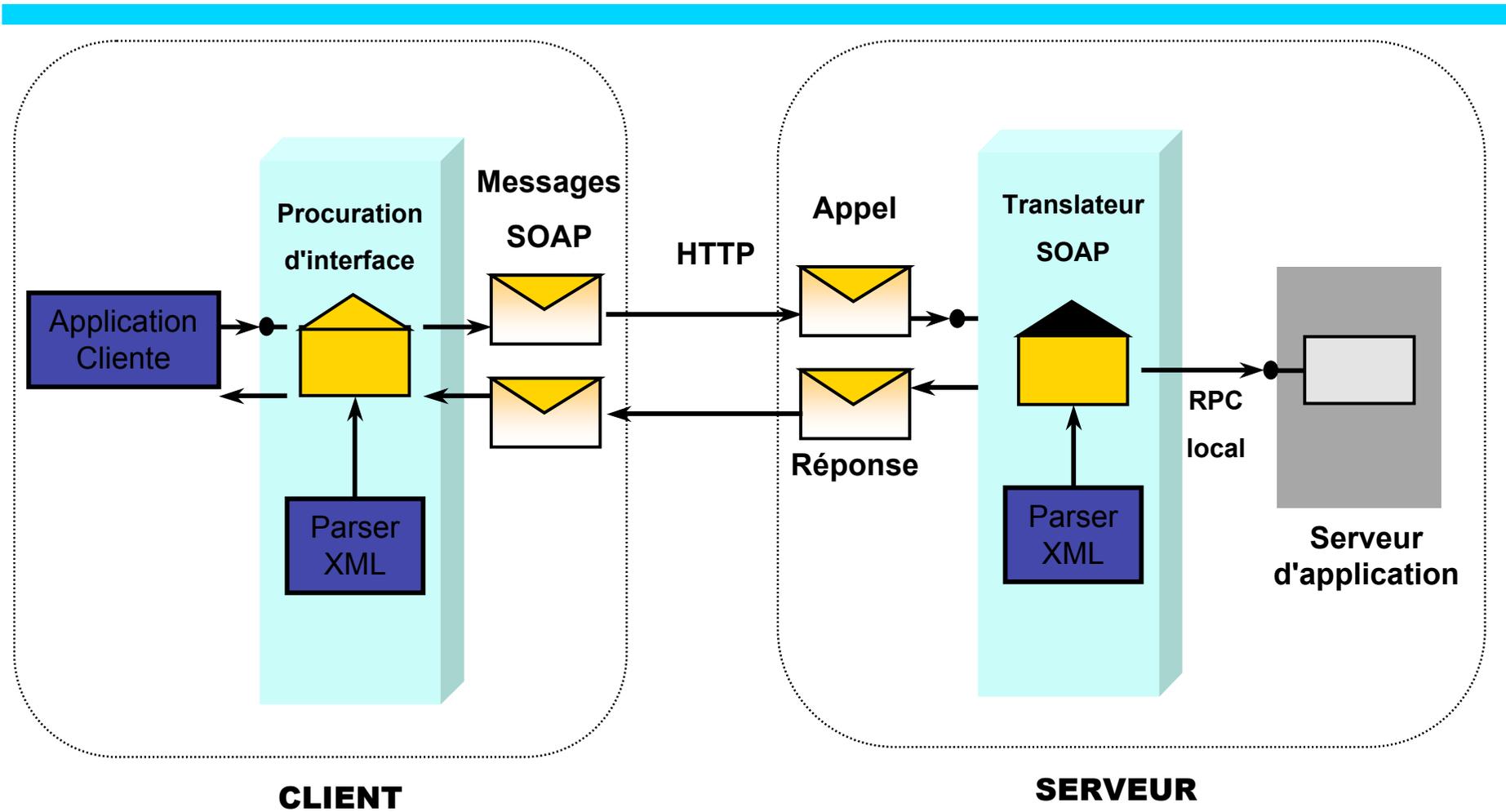
Exemple de requête en Java :

```
XmlRpcClient xmlrpc = new XmlRpcClient("http://www.localhost.com/xmlrpc/server.asp" );  
  
Vector params = new Vector();  
  
params.addElement( 123456 );  
  
Integer retVal = (Integer)xmlrpc.execute( "GetCurrentBalance", params );
```

# SOAP

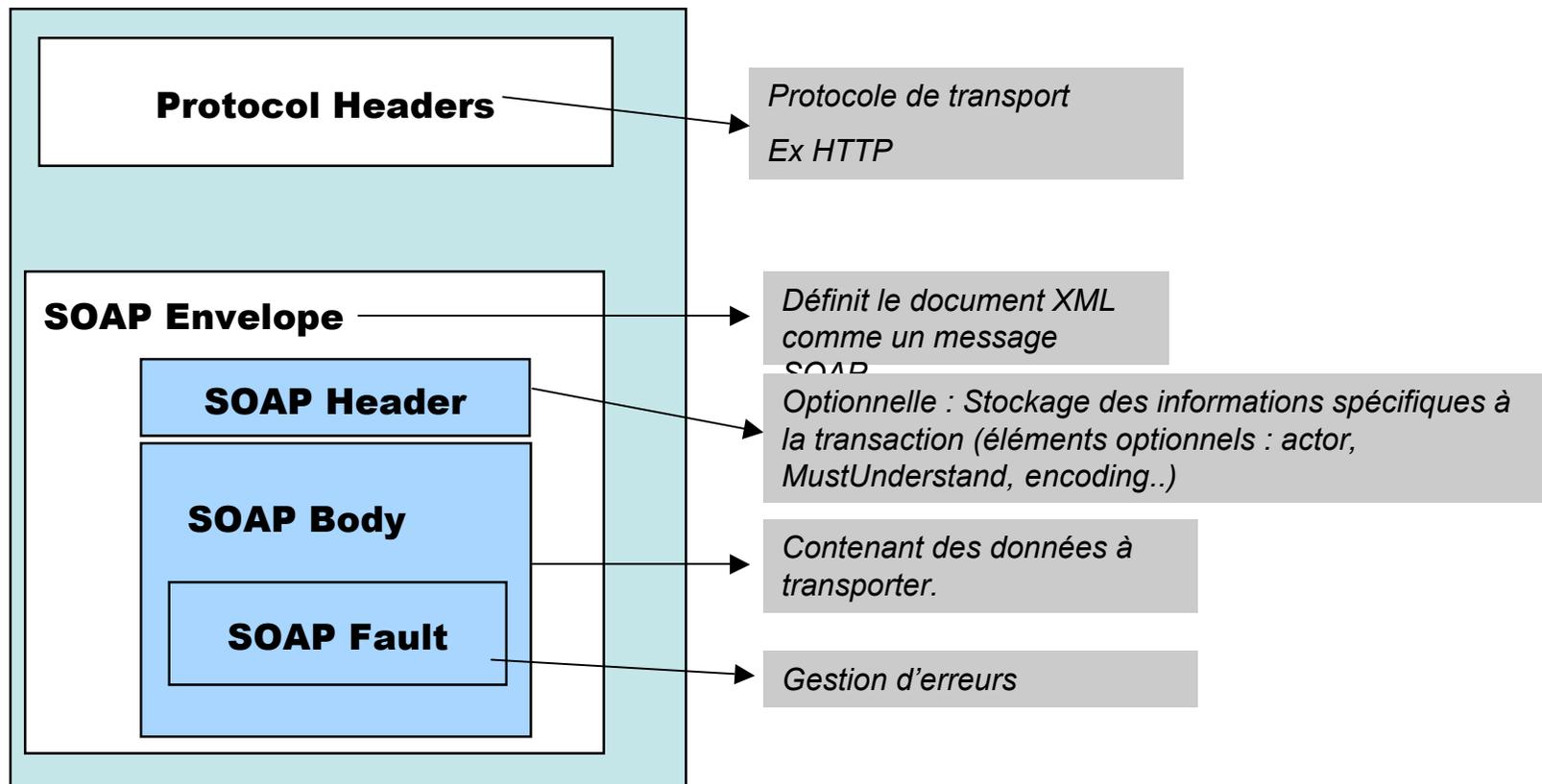
- SOAP définit le cadre général pour l'échange de données structurées en XML
- SOAP permet d'échanger des structures de données complexes en XML avec les Namespaces, et la spécification XML Schéma
- Historique de SOAP
  - SOAP 0.9 (septembre 1999)
    - Editeur (Microsoft, DeveloppMentor, UserLand)
  - SOAP 1.0 (Nov 1999)
    - IETF
  - SOAP 1.1 (Avril 2000)
    - IBM & Soumission W3C
  - W3C SOAP 1.2 (Sept 2000 à Mai 2002)
    - Refonte SOAP 1.1
    - Groupe de travail : 40 entreprises (IBM, Microsoft, Sun, Intel, Xerox, Canon ...)
- Spécification du W3C (WS-\*)
  - SOAP 1.2 Messaging Framework : la structure pour les échanges de messages
  - SOAP 1.2 Adjuncts : Ajouts
  - Primer : Préliminaires (sous forme de tutoriel..)

# SOAP



# SOAP

- Structure d'un message SOAP



# SOAP

## • Structure d'un message SOAP

- **Envelope**: c'est lui qui contient le message et ses différentes sous-blocs. Il s'agit du bloc racine XML. Il peut contenir un attribut *encodingStyle* dont la valeur est une URL vers un fichier de typage XML qui décrira les types applicables au message SOAP.
- **Header**: c'est un bloc optionnel qui contient des informations d'en-têtes sur le message. Si il est présent, ce bloc doit toujours se trouver avant le bloc **Body** à l'intérieur du bloc **Envelope**.
- **Body**: c'est le bloc qui contient le corps du message. Il doit absolument être présent de manière unique dans chaque message et être contenu dans le bloc **Envelope**. SOAP ne définit pas comment est structuré le contenu de ce bloc. Cependant, il définit le bloc **Fault** qui peut s'y trouver.
- **Fault**: ce bloc est la seule structure définie par SOAP dans le bloc Body. Il sert à reporter des erreurs lors du traitement du message, ou lors de son transport. Il ne peut apparaître qu'une seule fois par message. Sa présence n'est pas obligatoire.

# SOAP-Header

## Exemple : Bloc Header – Message à destination de Plusieurs Nœud SOAP

```
<soap:Header>
  <m:User xmlns:m="http://www.exemple.com/rights/"
    soap:actor="http://www.exemple.com/rights/RightsManager">
    Charles
  </m:User>
  <m:Session xmlns:m="http://www.exemple.com/session/"
    soap:mustUnderstand="1">12AE3C
  </m:Session>
  <m:Lang xmlns:m="http://www.exemple.com/lang/"
    soap:actor="http://schemas.xmlsoap.org/soap/next" soap:mustUnderstand="1">
  </m:Lang>
</soap:Header>
```

*Elément USER :*

*A destination du Nœud  
RightManager*

*Elément Session*

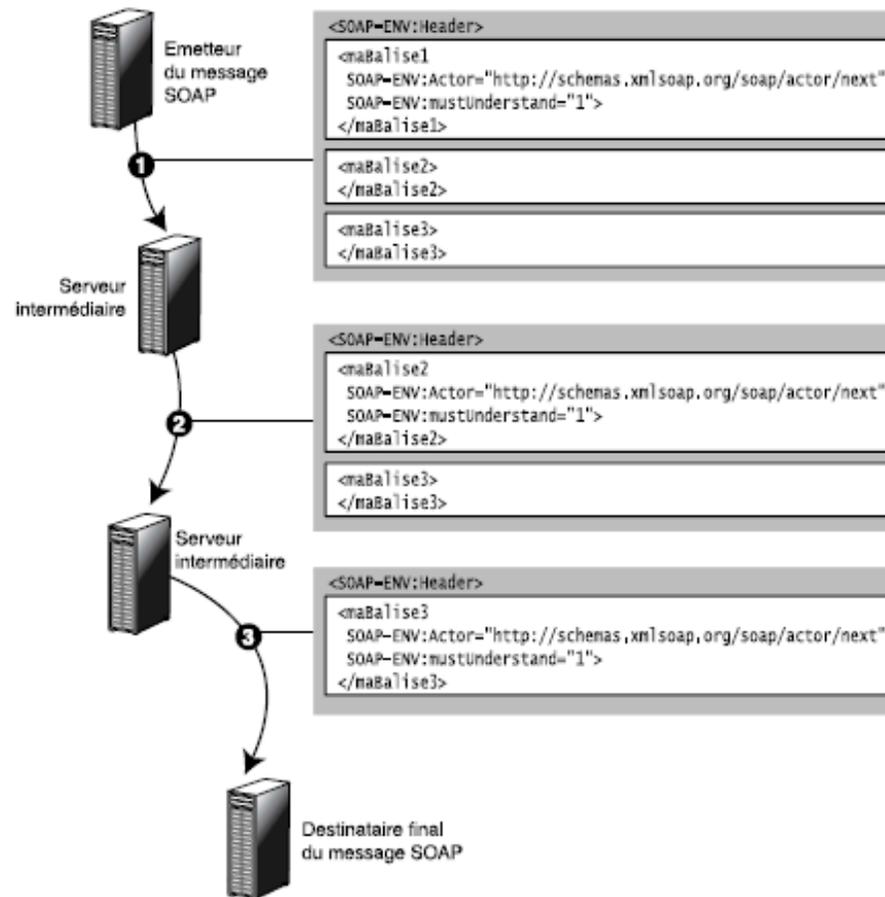
*A destination du Nœud Final*

*Elément USER :*

*A destination du prochain  
Nœud*

# SOAP Header

Figure 2-7. L'en-tête dans une série de transferts successifs de messages SOAP



# SOAP – Fault

## Exemple : Bloc Fault

```
<soap:Body>
  <soap:Fault>
    <faultcode>soap:Server</faultcode>
    <faultstring>Impossible de router le message.</faultstring>
    <faultactor>http://www.exemple.com/messageDispatcher</faultactor>
    <detail>
      <m:error xmlns:m="http://www.exemple.com/errors"> E_NO_ROUTE </m:error>
    </detail>
  </soap:Fault>
</soap:Body>
```

Identifiant de l'erreur – définit par SOAP

Description bref du message

Composant qui généré l'erreur (URL)

Message spécifique à l'application

# SOAP Exemple

<http://www.webservicex.net/>

## REQUETE

```
POST /stockquote.asmx HTTP/1.1
Host: www.webservicex.net Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.webserviceX.NET/GetQuote"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetQuote xmlns="http://www.webserviceX.NET/">
      <symbol>string</symbol>
    </GetQuote>
  </soap:Body>
</soap:Envelope>
```

*Protocole de transport  
Ex HTTP*

*Définit le document XML  
comme un message  
SOAP*

*Contenant des données à  
transporter.*

# SOAP Exemple

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xmlns:xsd=http://www.w3.org/2001/XMLSchema
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetQuoteResponse xmlns="http://www.webserviceX.NET/">
      <GetQuoteResult>string</GetQuoteResult>
    </GetQuoteResponse>
  </soap:Body>
</soap:Envelope>
```

## Réponse SOAP

```
<StockQuotes>
  <Stock><Symbol>FORD</Symbol>
  <Last>2.28</Last>
  <Date>11/24/2008</Date>
  <Time>4:00pm</Time>
  <Change>+0.20</Change>
  <Open>2.20</Open>
  <High>2.30</High>
  <Low>2.07</Low>
  <Volume>124718</Volume>
  <MktCap>18.0M</MktCap>
  <PreviousClose>2.08</PreviousClose>
  <PercentageChange>+9.62%</PercentageChange>
  <AnnRange>1.51 - 3.27</AnnRange>
  <Earnings>-0.174</Earnings>
  <P-E>N/A</P-E>
  <Name>Forward Industrie</Name></Stock>
</StockQuotes>
```

# SOAP(Exemple)

- Appel du Webservice stockquote en PHP

```
<?php
$params['symbol']="ford";

$client = new SoapClient("http://www.webserviceX.net/stockquote.asmx?wsdl");

$result = $client->GetQuote($params);
$resultQuote = $result->GetQuoteResult;

echo $resultQuote;

?>
```

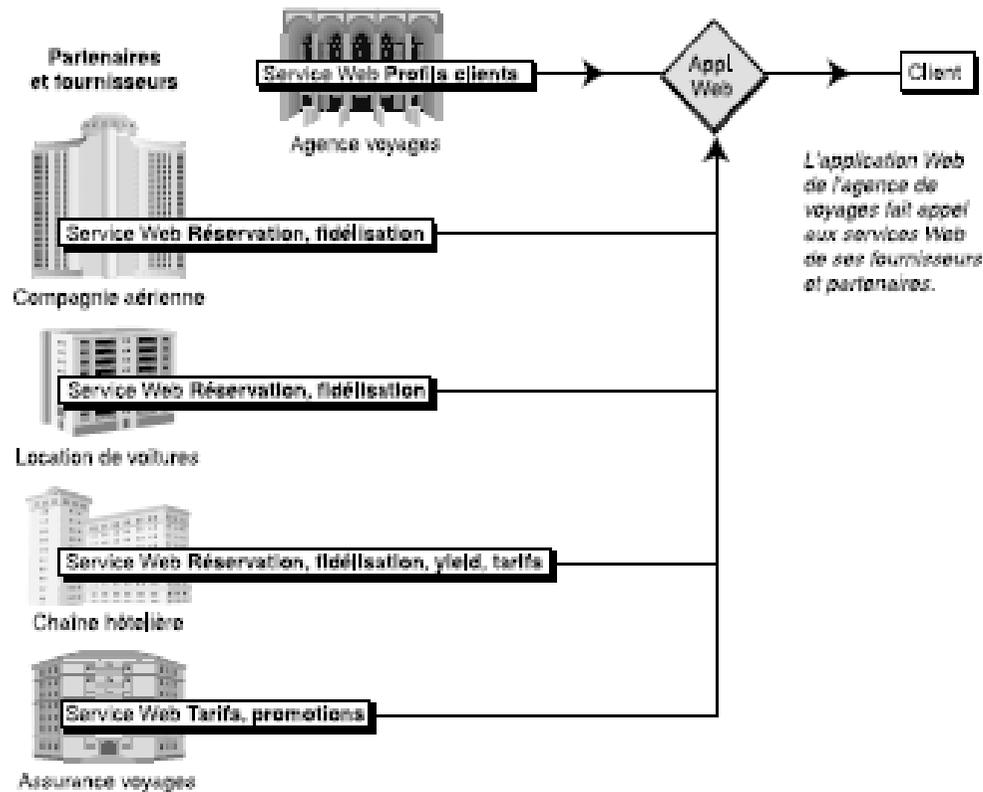
**CREATION d'un OBJET SOAPCLIENT**  
//L'ouverture du fichier WSDL va permettre d'automatiser  
l'utilisation du Web Service  
// Les méthodes définies dans le WSDL seront vu comme des  
méthodes interne

**APPEL de la méthode GETQUOTE du WS STOCKQUOTE**  
// vu comme une méthode locale

# SOAP

## Exemple d'application

Figure 1-2. Application Web comme agrégation de services Web.



# SOAP

---

- Implémentations
  - JAVA (API et outils associés)
    - JAX-RPC (Java XML – based RPC) : utilisation de SOAP en mode RPC
    - JAXR (JA XML registries) : utilisation de UDDI
    - JAXM (JA XML Messaging) : utilisation de SOAP en mode message
  - Microsoft (technologie .NET)
    - Api dans la bibliothèque de classes de .NET
  - Classes PHP SOAP : Divers projets Open Source
  - Perl : SOAP::Lite, UDDI::Lite, XMLRPC::Lite
  - ...

# Bibliographie

---

- « Services Web avec SOAP, WSDL, UDDI, ebXML » de Jean Marie CHAUVET.
- <http://www.irisa.fr/coo/2001/W3CSOAP1.pdf>
- <http://www.w3.org/2003/06/soap12-pressrelease.html.fr>
- [http://igm.univ-mlv.fr/~dr/XPOSE2005/rouvio\\_WebServices/soap.html](http://igm.univ-mlv.fr/~dr/XPOSE2005/rouvio_WebServices/soap.html)
- <http://www.commentcamarche.net>
- <http://www-adele.imag.fr/users/Didier.Donsez/cours>
- <http://www.w3schools.com/soap/>

# Questions

- 
- ?