Conservatoire Nationale des Arts et Métiers

Centre d'enseignements de Grenoble

Année Universitaire: 2008-2009

SERVICE WEB – SOAP

Cours :NFE107 Urbanisation & Architecture des Systèmes d'Information

Auditeurs : Youssef BELAID

1 Introduction

Les Web Service sont un type d'architecture reposant sur les standards de l'internet. De ce fait Il s'agit d'une architecture orientée service permettant à des applications de communiquer entre elle directement entre elles sans se préoccuper des technologies sur lesquelles elles sont implémentées. Les Web Service se présentent comme étant la solution pour répondre à l'interopérabilité des systèmes.

Il existe trois grands standards dans les Web Service :

- REST
- XML-RPC
- SOAP : Simple

Parmis ces 3 méthodes nous avons d'un côté XML-RPC et SOAP qui sont basés sur des technologies XML et qui utilisent le protocole HTTP, mais qui peuvent se s'appuyer sur d'autre protocole de transport. REST à l'inverse est le mode natif de HTTP.

Même si les services Web ont le vent en poupe, ils peuvent être vu comme une évolution des technologies à composants distribués plus âgées comme les appels de procédures distances (RPC) ou encore la famille des RPC Objets (DCOM, Corda, Java RMI).
Nous allons dans le cadre de cette fiche de lecture décrire les différentes façons de communiquer avec un service Web.

2 Définitions d'un Web Service

Définition du W3C

« 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 »

Définition de 'diconet.com'

« Une technologie permettant à des applications de dialoguer à distance via Internet indépendamment des plates-formes et des langages sur lesquelles elles reposent »

Définition 'Wikipedia'

« 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 »

Les trois définitions ayant été publiées à différents stade de l'évolution des services Web, celle-ci peuvent prêter à confusion. Nous pouvons néanmoins trouver un consensus pour définir un service Web de façon plus générale.

Un web service est une technologie permettant à des applications de communiquer entre elles :

- en s'appuyant sur les standards du web (HTTP, XML)
- indépendamment de l'architecture sur lesquelles elles sont implémentées
- en échangeant des documents sous le format XML

3 – Architecture orientée service

Pour mieux appréhender la terminologie lié au Web Service et également situer tous les acteurs entrant en jeu dans les Web Service, nous allons donner un bref description de l'architecture orienté objet.

Une architecture orientée services (notée SOA pour Services Oriented Architecture) est une architecture logicielle s'appuyant sur un ensemble de services simples.

Lorsque l'architecture SOA s'appuie sur des web services, on parle alors de WSOA, pour Web Services Oriented Architecture).

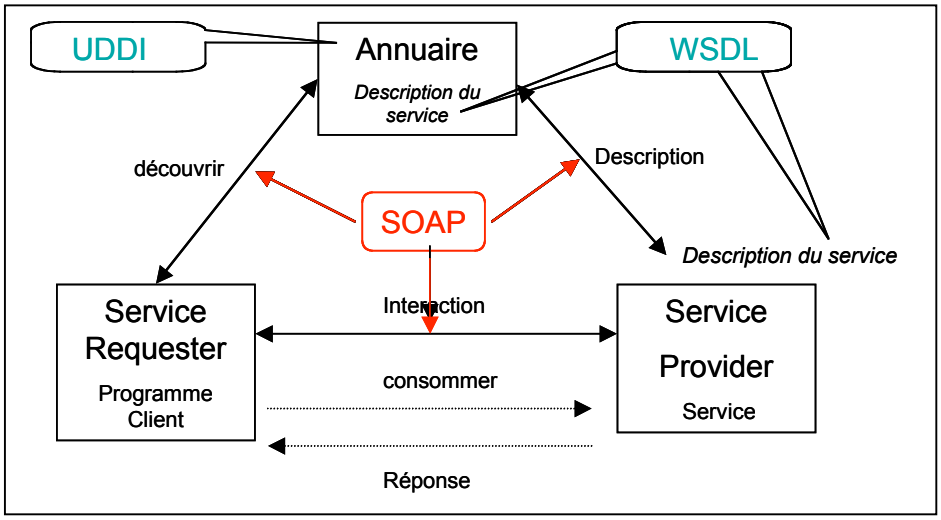

Figure 1 : Architecture WSOA

3-1 Terminologie dans une architecture SOA

- WSDL (Web Services Description Language) :

Le langage WSDL permet de décrire au format XML des Web Services en précisant les méthodes pouvant être invoquées, leur signature et le point d'accès (URL, port, etc..).

La structure d'un document WSDL est assez complexe et on y retrouve un espace de nom spécifique à WSDL, un espace de nom SOAP, l'espace de nom des messages et l'espace de nom XML Schema permettant de spécifier des types normalisés.

Le WSDL décrit une Interface publique d'accès à un Service Web, notamment dans le cadre d'architectures de type SOA (Service Oriented Architecture).
C'est une description fondée sur le XML qui indique « comment communiquer pour utiliser le service »;
La constitution et la génération d'un fichier WSDL peut être réalisées par outils tel que DIA + UML2PHP5 ou ZendStudio.

- UDDI (Universal Description Discovery and Integration) :

UDDI est un annuaire de service fondé sur XML et plus particulièrement destiné aux services Web. Un annuaire UDDI permet de localiser sur le réseau le service Web recherché, il s'agit d'un élément clé dans les spécifications de Services, car il permet l'accès au répertoire des utilisateurs potentiels de services Web.

- SOAP (Simple Object Protocol communication) :

SOAP est un protocole de communication basé sur XML permettant la transmission de messages en différent service Web. Nous reviendrons plus en détail sur le protocole SOAP dans la suite de la fiche de lecture.

3-1 Les acteurs dans une architecture SOA

- Annuaire – Service Registry

L'annuaire de services référence l'ensemble des services (et des contrats associés) disponibles au sein du SI, il participe ainsi activement à la mise en œuvre d'une cartographie dynamique du SI. Dans un modèle de bus, l'annuaire peut être auto-alimenté par le service (enregistrement). Les annuaires UDDI forment aujourd'hui le standard de référencement des services.

(Définition Wikipédia)

L'annuaire peut avoir une portée au niveau d'une application, d'une entreprise ou mondial.

- Service Provider : (Fournisseur de service)

L'application s'exécute sur un serveur et comporte un module logiciel accessible en XML

- Service Requester

Application cliente se liant à un service et invoquant ses fonctions par des messages XML (REST, XML-RPC,SOAP)

4 – Communiquer avec les Web Service

4-1 Méthode REST (Representational State Transfer)

REST permet de construire une application pour les systèmes distribués comme le Web. Ce n'est pas un protocole ou un format mais une architecture (celle de http). On identifie alors :

- Une URI (Uniform Ressource Locator) qui permet d'identifier la ressource à laquelle on souhaite accéder ;
- Une méthode http (POST ou GET) pour définir quel opération on souhaite effectuer sur la ressource ;
- Des entête http pour gérer les métadonnées et les informations sur le transport ;

Consommer un service Web REST revient a appeler une simple URL en http, le serveur renvoie sa réponse, la plupart du temps en XML.

Exemple d'échange de données en 'mode REST' :

Le client émet la requête http suivante :

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

- le client adresse sa requête à l'URI ws.ct-goat.com/getCityInfos.asp en lui transmettant les paramètres uID=xxxxxxxxxxxx&comID=562

Le provider de service (le serveur de Web Service) retourne un fichier XML .

```
<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>
```

Figure 3 – Exemple de réponse au format XML

Dans l'exemple ci dessous un service Web 'getCityInfos' retourne les éléments d'une commune à partir d'un code INSEE. Cet exemple à été repris du site 'transisère.fr '

Cette architecture est très utilisée pour la réalisation de service Web destinés à la communication entre machine.

Il a l'avantage d'être simple. Il s'agit en réalité de l'utilisation du protocole http.

4-2 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.

Le XML-RPC permet la transmission du point A au point B. Il n'y a qu'un seul appel de méthode par message de requête et une seule valeur renvoyée, sous forme de tableau ou de structure pour transmettre plusieurs valeurs, par message de réponse.

Les processus d'invocation de service à distance utilisent le protocole http pour le transport des données et le langage XML pour leur codage (voir figure 3). Il est toutefois possible de faire du XML-RPC sur un autre protocole que http.

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.

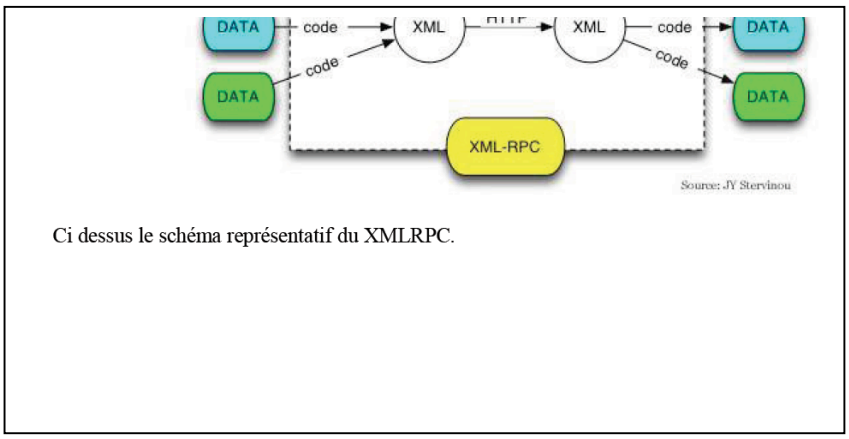

Figure 4 Schéma représentatif du XML- RPC

La spécification XML-RPC n'est plus maintenue depuis 1998 et elle est considérée comme l'ancêtre de SOAP.

Plutôt que d'analyser la spécification dans ses moindres détails, analysons un message concret pour en dégager les principales caractéristiques (voir figure 5 et 6).

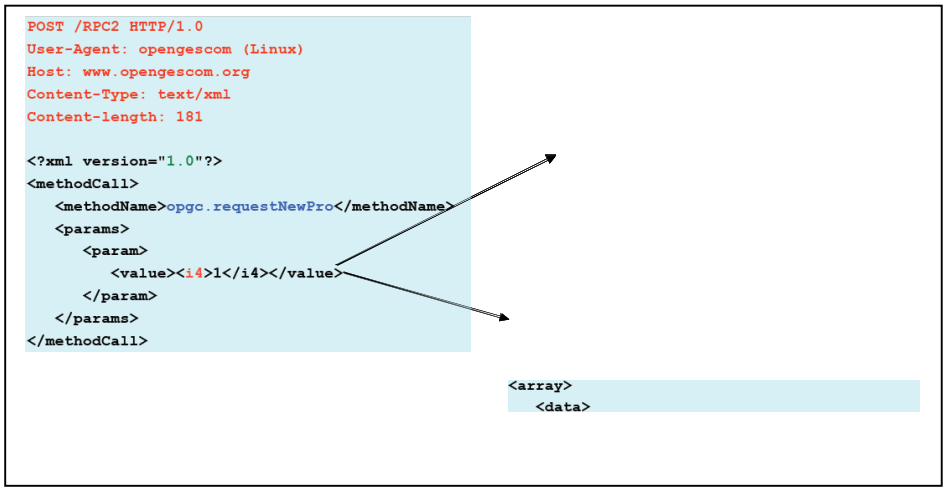

Figure 5 : Exemple de requête XML-RPC cliente

Figure 6 : Exemple de réponse

La figure ci-dessus représente la requête http, transportant un message XML-RPC.

La partie en rouge décrit l'entête du message http, il est indépendant de la spécification XML-RPC.

La partie inférieure du message représente le corps de la requête http, dans laquelle est encapsulée la requête XML-RPC.

L'une des premières choses que vous pouvez remarquer dans cette exemples, c'est que le code XML est totalement dépourvu d'attributs. Seuls des éléments sont utilisés. En outre, vous noterez rapidement que :

- chaque message de requête figure au sein d'un élément <methodCall>
- le nom de la méthode étant inclus dans un élément <methodName> enfant.
- de même, chaque message de réponse(voir figure 6) réside à l'intérieur d'un élément <methodResponse>.

Au-delà de cet aspect, vous pouvez voir que l'élément <value> est transmis dans les éléments <param> et que le type de valeur peut être spécifié à l'aide des éléments de type de données <int>, <double> et <string>, pour ne citer que ceux-là. Pour transmettre davantage de données, par exemple lorsque vous avez un nombre variable de paramètres ou voulez que votre paramètre soit une structure composée, vous pouvez voir dans l'exemple que XML-RPC supporte également les types de données <array> et <struct>.

XML-RPC a été conçu pour permettre à des structures de données complexes d'être transmises, exécutées et renvoyées très facilement.

L'exemple de la figure 5 donné à titre d'illustration ci dessus ne présente que 3 types de données pouvant être envoyées par un message. XML-RPC. Tous les types de données supportés par XML-RPC sont détaillés dans la spécification sur le site http:// xml-rpc.org.

La majorité des langages de programmation, implémentent la spécification XML-RPC. Nous pouvons cité à titre d'exemple : ASP, Delphi, Eiffel, Perl, Python, .NET, PHP, …

Le listing suivant représente un exemple d'appel d'une méthode sur le modèle XML-RPC en JAVA.

```
import org.apache.xmlrpc.*;
 import java.util.*;

 public class TestMeerkat {
    static final String MEERKAT_URL = "http://www.oreillynet.com/meerkat/xml-rpc/server.php";

    public static void main(String[] args) {
        try {
                XmlRpcClient rpc = new XmlRpcClient(MEERKAT_URL);

            // I don't have any params for this call, but the second arg of
            // execute() expects a vector of params,
            // so I'll just create a new vector that will disappear.
            Vector result = (Vector) rpc.execute("system.listMethods",new Vector());

            System.out.print("system.listMethods result:\n" + result.toString() + "\n\n");

            result = (Vector)rpc.execute("meerkat.getChannels",new Vector());

            System.out.print("system.getChannels result:\n" + result.toString());

        } catch (XmlRpcException rpcEx) {
            System.err.print("XmlRpc exception code " + rpcEx.code + ": " + rpcEx.getMessage());
        } catch (java.net.MalformedURLException malEx) {
            System.err.print("Malformed Url: " + malEx.getMessage());
       } catch(java.io.IOException ioEx) {
            System.err.print("IO exception: " + ioEx.getMessage());
        }
    }
 }
```

Figure 7 : Exemple d'mplémentation de XML-RPC en java

4-3 SOAP (Simple Object Access Protocol)

4-3-1 Généralités

Des 3 méthodes présentées dans le cadre de cette fiche de lecture, SOAP est la spécification la plus récente, la plus complète et la plus complexe.

Il s'agit du type de service le plus courant. Il bénéficie d'une spécification complète et détaillée dans les normes éditées par le W3C.

Avant d'aller plus en avant, revenons sur l'historique de la spécification SOAP :

- SOAP 0.9 (Septembre 1999) : Editeurs : Microsoft,DevelopMentor, UserLand
- SOAP 1.0 (novembre 1999) : IETF
- SOAP 1.1 (Avril 2000) : IBM et soumission au W3C
- En septembre 2000 : Repris par le W3C
- Mai 2002 : refonte de SOAP 1.1 et publication de la version SOAP1.2

Les spécifications de SOAP auprès du W3C sont identifiables par leur nom WS-*, elles sont réparties dans 3 grandes familles :

- SOAP 1.2 Messaging Framework :la structure pour les échanges de messages
- SOAP 1.2 Adjuncts : Ajouts
- Primer : Préliminaires (sous forme de tutoriel.)

L'objectif de cette fiche de lecture n'étant pas de détaillé les moindres détails des spécifications, nous allons au travers d'exemples parcourir les contours afin d'en dégagé les principale fonctionnalités.

Avant d'entrer plus en détail dans la structure d'un message SOAP nous allons rappeler que :

- SOAP définit le cadre général pour l'échange de données structurées en XML;
- SOAP permet d'échanger des structures de données complexes en XML ;
- SOAP repose sur XML et sur quelques standards dérivés, les NamesSPaces, XML Schema en particulier ;
- SOAP est indépendant des langages de programmation ou des systèmes d'exploitation sur lesquels il est implémenté ;
- SOAP est implémenté dans tous les langages de programmation : JAVA, PHP, .NET, PERL …. ;

4-3-2 Structure d'un message SOAP

Un message SOAP est un document XML constitué d'une enveloppe, contenant un entête (facultatif) et le corps du message (voir figure ci dessous)

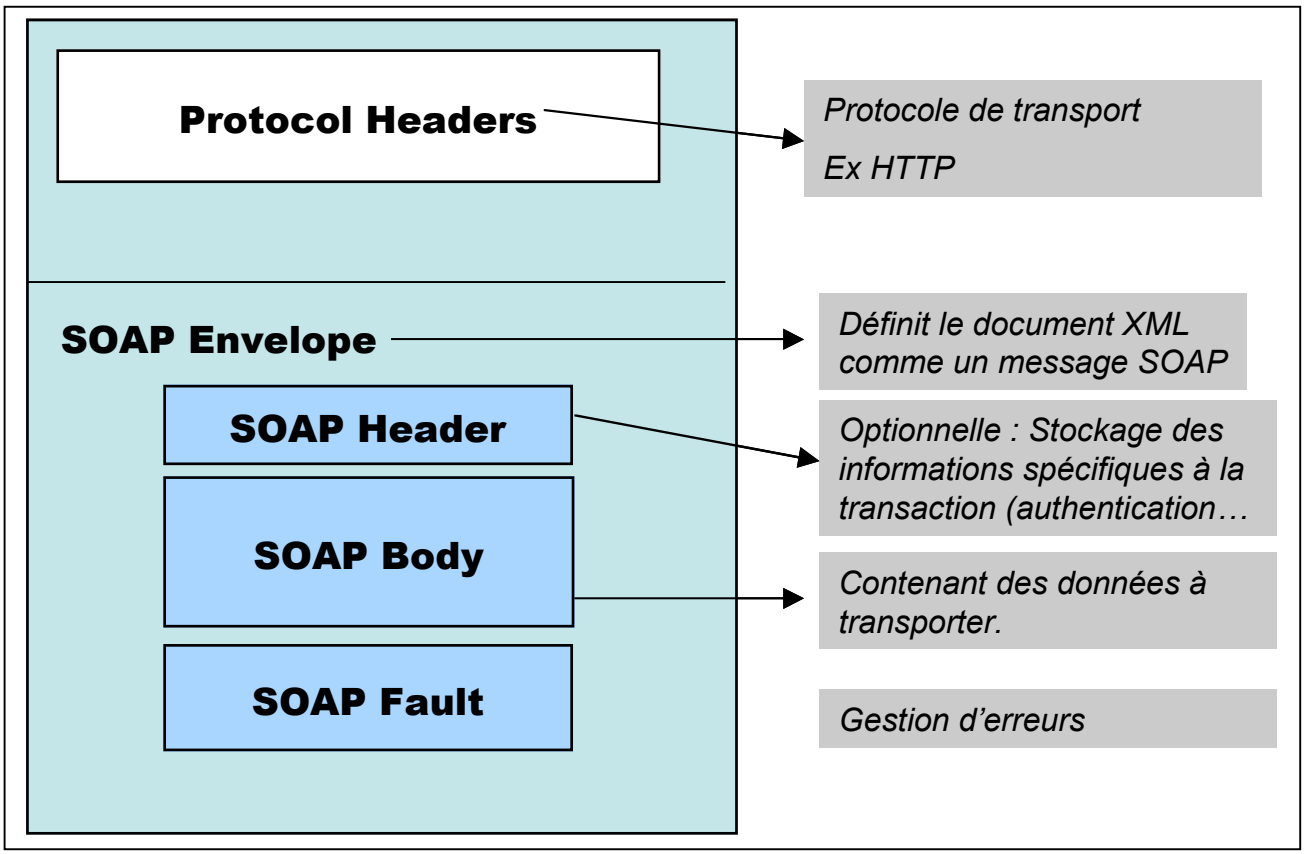

Figure 8 : Message SOAP

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

L'enveloppe est la racine du document XML contenant le message SOAP, marquée par la balise <Envelope>. La spécification impose que tous les attributs de cette balise et celles imbriquées soient explicitement associés à un namespace, de manière à supprimer toute ambiguïté. On peut cité un namespace fréquemment utilisé :

- SOAP-ENV associé à l'URI http://www.w3.org/2001/06/soap-encoding

L'entête SOAP <SOAP-ENV :Header>

La balise <Header> permet de passer dans le message des informations complémentaires sur ce même message. Cet élément est facultatif mais s'il est présent, il doit être le premier

présent dans l'enveloppe SOAP du message. L'entête Header peut avoir plusieurs usages, il peut par exemple contenir des informations d'authentification de l'émetteur, ou bien le contexte d'une transaction dont le message n'est qu'une des étapes. Pour les couches de transports (comme FTP) ne fournissant pas, à l'émission, d'adresse de retour, on peut aussi utiliser l'entête pour identifier l'émetteur du message SOAP.

Un message peut avoir à traverser plusieurs intermédiaires (voir figure 9) avant d'atteindre son destinataire final. Dans ce cas l'entête, l'entête joue un rôle important. A chaque arrêt le long de l'itinéraire, le serveur intermédiaire extrait de l'entête qui le concerne en propre et ajoute ce qui est nécessaire au serveur intermédiaire suivant. Les éléments concernant un serveur intermédiaire sont marqués par la présence de l'attribut 'SOAP-ENV :Actor'

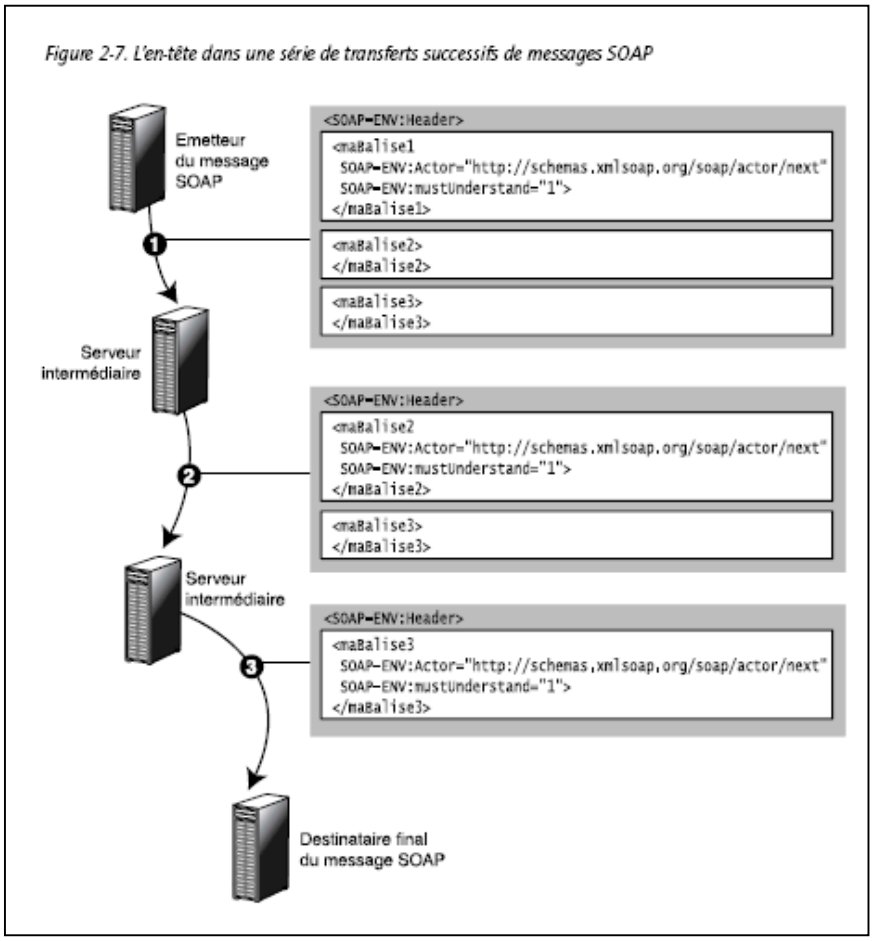

Figure 9 : Exemple illustrant l'usage de l'entête SOAP pour une série de transferts successifs de messages suivants

Le corps du messages SOAP <SOAP-ENV :Body>

Le corps du message est constitué du seul élément Body, contenant un ou plusieurs sous-éléments. Ces sous éléments sont les suivants :

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

Le corps du message peut passer des appels de procédures distants (au sens OMG : un objet, une méthode et les valeurs de ces arguments), des résultats ou des messages d'erreur. Mais l'usage s'est élargi et le corps du message est souvent utilisé pour simplement passer des données structurées entre application.

Le bloc FAULT est utilisé pour retourner à l'émetteur le type d'erreur dans le cas d'une défaillance. Les codes erreurs sont détaillées dans la spécification.

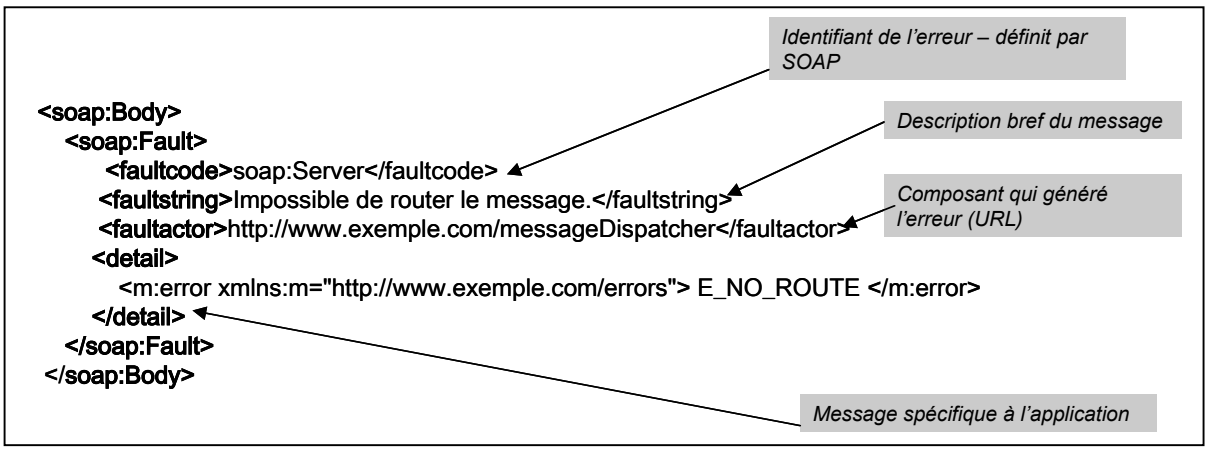

Figure 10 : Exemple bloc Fault

4-3-3 Exemple d'implémentation SOAP

L'exemple en lui même n'a pas beaucoup d'importance mais il m'a permis d'implémenté un cas réel de client SOAP.

Pour des raisons de commodités j'ai utilisé le langage PHP, mais cette démonstration aurait pu être faire dans n'importe quels langage de programmation. (voir listing ci joint figure 11)

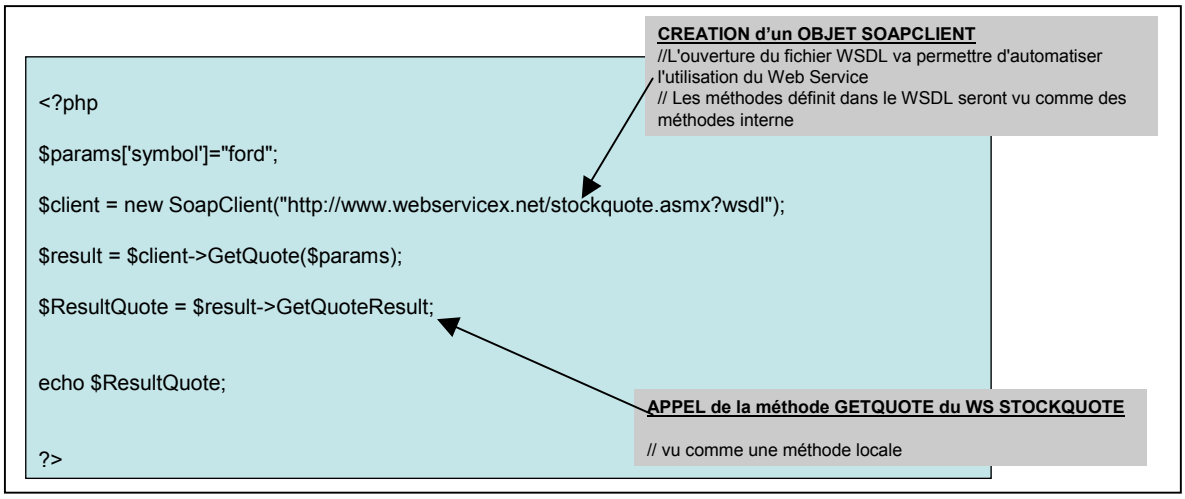

Figure 11 : Appel d'un web Service SOAP en PHP

On commence par créer un objet de type SoapClient en donnant l'adresse de la déclaration du WSDL. Une fois cet objet créé, la méthode GetQuote sera vu comme une méthode locale de notre objet.

La variable $client est un objet qui représente la liaison avec le service Web 'stockquote'. Chaque méthode 'appelable' sur le service est représentée par une méthode sur cet objet. Dans notre cas la méthode 'GetQuote' prend en paramètre le nom de société et retourne un objet représentant les éléments de côte boursière.

L'exemple ci dessus va se traduire par la requête http représenté dans la figure 12 ci dessous.

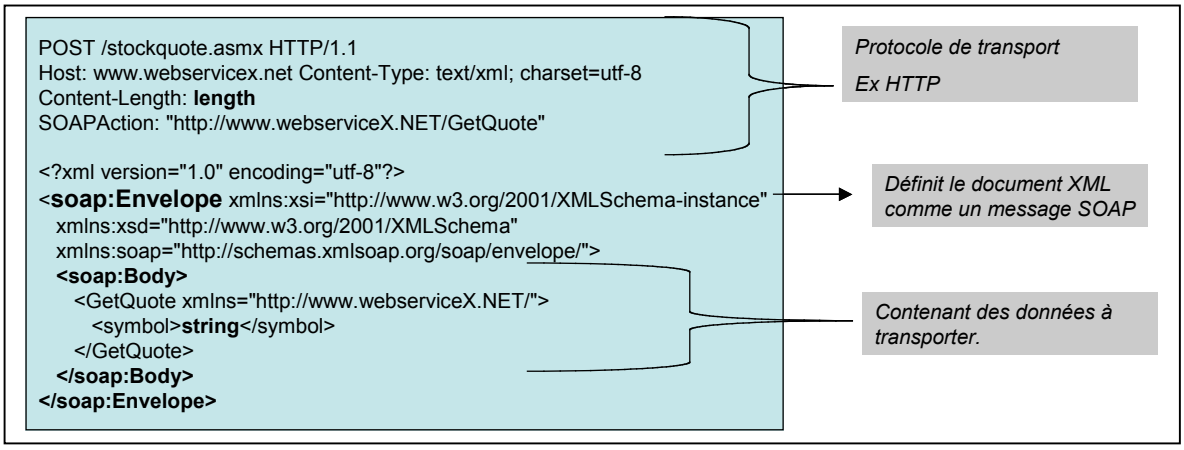

Figure 12 : Requête http d'un message SOAP

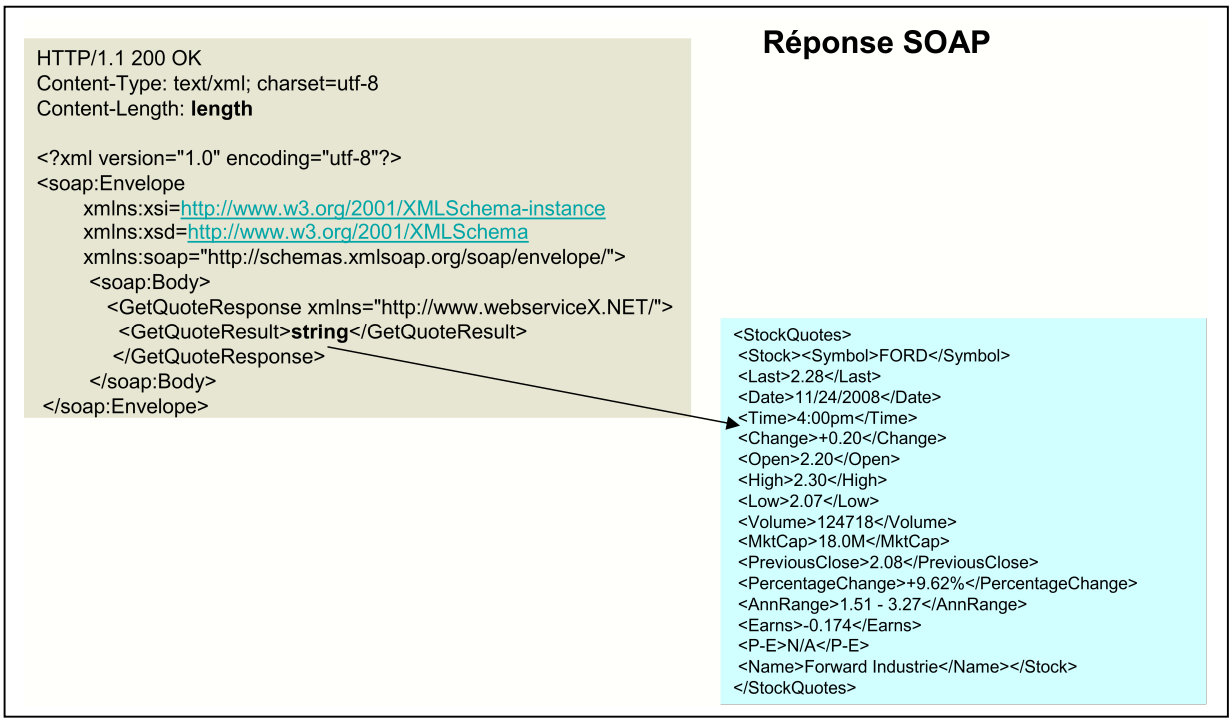

Figure 13 : requête http d'un réponse SOAP

5 - Conclusion

Les trois méthodes de Web Service présentées ne sont pas à mettre en opposition ni en comparaison. Chacune d'entre elle peut être adaptée à une situation, l'idéal étant d'implémenter les trois lors de la création d'un 'Web service'.

Un grand avantages des services Web c'est qu'il de repose sur les standards de l'internet et notamment le protocole http. De ce fait l'utilisation des Web Service n'est pas bloquée par proxies et Firewall,

L'implémentation des différentes spécifications sur la majorité des langages de programmation à pour avantage de rendre extrêmement simple l'utilisation des Web Service.

Ces seules raisons peuvent justifier de préférer l'utilisation des Web Service à d'autre technologie comme CORBA ou RMI.

6 - 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://www.commentcamarche.net
•http://www-adele.imag.fr/users/Didier.Donsez/cours
•http://www.w3schools.com/soap/