Introduction

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)

Web Services peut accélérer l'intégration de différents types de systèmes gouvernements, des entreprises et de l'entreprise, réduire les coûts et améliorer la prestation, mais ils conduisent également à de nombreux problèmes de sécurité. L’intérêt pour les services Web a rapidement augmenté de leur début d'utilisation.

Un service Web est un ensemble de méthodes Web qui sont hébergés sur un serveur d'application. Méthodes Web sont des méthodes qui peuvent être invoquées à distance sur un réseau ou Internet. Pour invoquer une méthode Web, il faut créer un message et de l'envoyer au service Web via HTTP ou un autre mécanisme de transport. Le plus souvent, un service Web envoie un message de réponse en retour pour chaque appel de méthode; le message de réponse contient des informations utiles ou des informations de défaut en cas d'erreur.

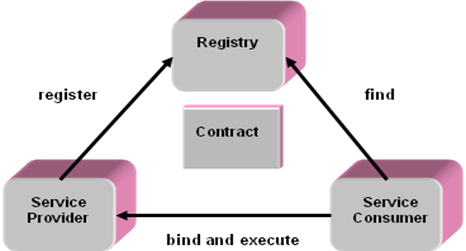
# Architecture du service web

L'architecture de services Web SOAP est illustrée dans la figure 1.0. Qui définit 3 entités:

1. Fournisseurs de services
2. Registre de service
3. Demandeur de services

Le fournisseur de services est l'entité adressable du réseau qui accepte et exécute les requêtes du consommateur. Le demandeur de service est une application, un service ou un autre type de module logiciel qui a besoin d'un service.

Un registre de service est un annuaire basé en réseau qui contient des services disponibles. Le consommateur de service trouve la description du service dans le registre qui est publié par le fournisseur de service. Grâce à cette description, le consommateur commence à interagir avec le service. La communication entre ces entités est basée sur XML et protocole SOAP.



|  |  |  |
| --- | --- | --- |
|  | Point de vue commercial | Point de vue technique |
| **Fournisseur de services** | Propriétaire de la de service | Plate-forme qui héberge le service |
| **Demandeur de service** | Entreprise qui nécessite certaines fonctionnalités | l'application qui appelle ou interagit avec le service |
| **Registre du service** | Registre consultable des descriptions de service où les fournisseurs de services publient leurs descriptions de service | |

Définition SOAP

**SOAP** (Object Access Protocol) est un protocole de **RPC** (Remote Procedure Call) permettant d'invoquer des méthodes d'objets distants. Il est comparable à DCOM ou CORBA mais contrairement à eux, il s'appuie sur des standards très connus. Il utilise XML pour définir les fonctions et les définitions disponibles. Il prend en charge divers protocoles de transport, tels que HTTP et SMTP, ainsi que différents formats comme MIME. Ces derniers sont très répandus sur de multiples plates-formes, ce qui donne à SOAP une grande portabilité et interopérabilité.

SOAP est une spécification non propriétaire. Il n'est pas lié à un protocole particulier. Il n'est pas non plus lié à un système d'exploitation ni à un langage de programmation.

SOAP étant un protocole d'échange d'informations entre diverses machines sur un réseau, elle nécessite un format pour transporter les données. Pour cela elle utilise des messages SOAP qui sont en fait des documents XML.

# Historique de SOAP

**SOAP 0.9** (Septembre 1999)

* Editeur (Microsoft, DeveloppMentor, UserLand)

**SOAP 1.0** (Novembre 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 ...)

Les spécifications de SOAP 1.2 sont composées de plusieurs parties :

* [Partie 0 : Primer](http://www.w3.org/TR/soap12-part0/)
* [Partie 1 : Messaging Framework](http://www.w3.org/TR/soap12-part1/)
* [Partie 2 : Adjuncts](http://www.w3.org/TR/soap12-part2/)
* [Specification Assertions and Test Collection](http://www.w3.org/TR/soap12-testcollection/)

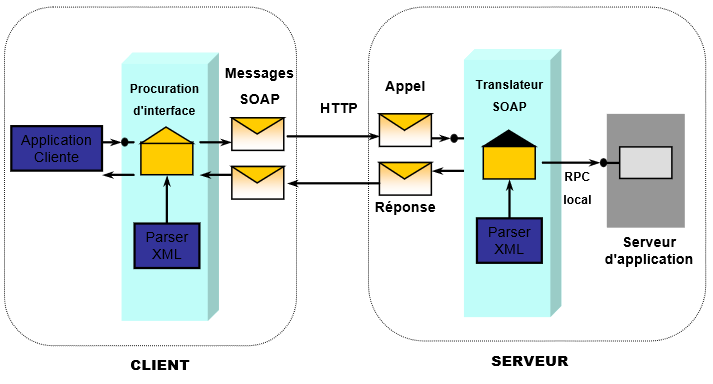
La version 1.2 des spécifications de SOAP est plus précise pour réduire les ambigüités qui pouvaient conduire à des problèmes d'interopérabilité entre différentes implémentations.

SOAP 1.2 propose un support pour des protocoles de transport différents de HTTP. La sérialisation de messages n'est pas obligatoirement en XML mais peut utiliser des formats binaires (XML Infoset par exemple).

Fonctionnement SOAP

Les services web de type Soap permettent l'appel d'une méthode d'un objet distant en utilisant un protocole web pour le transport (http en général) et XML pour formater les échanges. Les services web fonctionnent sur le principe client / serveur :

* un client appelle les services web
* le serveur traite la demande et renvoie le résultat au client
* le client utilise le résultat



L'appel à un service web de type SOAP suit plusieurs étapes :

1. Le client instancie une classe de type proxy encapsulant le service Web XML.
2. Le client invoque une méthode du proxy.
3. Le moteur SOAP sur le client crée le message à partir des paramètres utilisés pour invoquer la méthode
4. Le moteur SOAP envoie le message SOAP au serveur généralement en utilisant le protocol HTTP
5. Le moteur SOAP du serveur réceptionne et analyse le message SOAP
6. Le moteur fait appel à la méthode de l'objet correspondant à la requête SOAP
7. Le moteur SOAP sur le server crée le message réponse à partir de la valeur de retour
8. Le moteur SOAP envoie le message SOAP contenant la réponse au client généralement en utilisant le protocole http
9. Le moteur SOAP du client réceptionne et analyse le message SOAP
10. Le moteur SOAP du client instancie un objet à partir du message SOAP contenant la réponse

Un des intérêts des services web est de masquer aux développeurs la complexité de l'utilisation des standards sous-jacents. Ceci est réalisé grâce aux développements d'API et de moteurs pour la production et la consommation de services web.

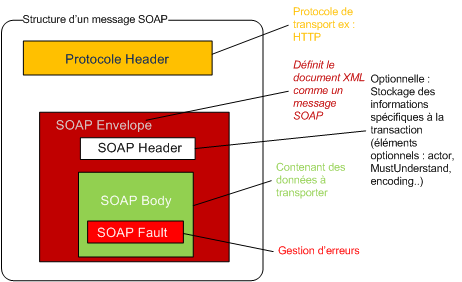
Ces API sont dépendantes des plate-formes utilisées (Java, .Net, PHP, Perl, ...) mais elles mettent toutes en oeuvre avec plus ou moins de complétude les standards de l'industrie relatifs aux services web notamment SOAP et WSDL.

Ainsi les développeurs peuvent se concentrer sur l'écriture des traitements proposés par les services et par leur consommation sans se soucier de la tuyauterie sous-jacente. Un minimum de compréhension est cependant nécessaire pour bien appréhender les mécanismes mis en oeuvre.

Architecture

La grammaire de SOAP est assez simple à comprendre. Elle procure un moyen d'accès aux objets par appel de méthodes à distance. Les deux plus fortes fonctionnalités de SOAP sont sa simplicité et le fait que tout le monde a accepté de l'utiliser.

Un message SOAP est composé de deux parties obligatoires **: l'enveloppe SOAP** et **le corps SOAP** ; et une partie optionnelle : **l'en-tête SOAP.**

****

# **SOAP envelope** (enveloppe) :

Est l'élément de base du message SOAP. L'enveloppe contient la spécification des espaces de désignation (**[namespace](http://fr.wikipedia.org/wiki/Namespace%23En_informatique)**) et du codage de données.

# **SOAP header** (entête) :

Est une partie facultative qui permet d'ajouter des fonctionnalités à un message SOAP de manière décentralisée sans agrément entre les parties qui communiquent. C'est ici qu'il est indiqué si le message est mandataire ou optionnel. L'entête est utile surtout, quand le message doit être traité par plusieurs intermédiaires.

S’il est présent, ce bloc doit toujours se trouver avant le bloc **Body** à l'intérieur du bloc **Envelope**.

# **SOAP body** (corps) :

Est un container pour les informations mandataires à l'intention du récepteur du message, il contient les méthodes et les paramètres qui seront exécutés par le destinataire final.

# **SOAP fault** (erreur) :

Est un élément facultatif défini dans le corps SOAP et qui est utilisé pour reporter les erreurs.

Par la suite, nous étudierons plus en détail le contenu de **l’enveloppe SOAP**, **le corps SOAP**, **l’entête SOAP**.

# L’enveloppe SOAP

L'enveloppe SOAP joue le rôle d’un conteneur pour les autres éléments du message SOAP, elle est définie au début par la balise **<soap:Envelope>** et se termine par la balise **</soap:Envelope>.**

Les messages SOAP ne peuvent pas être envoyés en lots, autrement dit l'enveloppe contient un seul message constitué d'un entête facultatif (SOAP header) et d'un **corps obligatoire** (SOAP body).



Toutes les balises XML associées à SOAP ont le préfixe ***soap*** .L'entête est **<soap:Header>** et le corps **<soap:Body>.**

SOAP repose entièrement sur les espaces de noms XML. Dans cet exemple, les espaces de noms sont introduits à l'aide d'un mot-clé « xmlns » *XML namespace* qui signifie espace de noms XML. L'espace de noms est utilisé pour identifier toutes les balises afin d'éviter les conflits. La spécification impose que tous les attributs contenus dans l'enveloppe SOAP soient explicitement associés à un *namespace*, de manière à supprimer toute ambiguïté. Par convention, la spécification SOAP définit deux *namespaces* fréquemment utilisés :

1. **soap-env ou soap** associé à l'URI « [..]schemas.xmlsoap.org/soap/envelope » pour définir le *namespace* de l'enveloppe dans la version 1.1, et à « [..]wwww.w3.org/2001/06/soap-envelope » dans la version 1.2 reprise par le W3C.
2. **soap-enc:encodingStyle**  associé à l'URI « [..]schemas.xmlsoap.org/soap/encoding » pour la définition des formats de types de données dans la version 1.1, et à « [..]www.w3.org/2001/06/soap-encoding » dans la version 1.2.

Deux autres espaces de noms fortement utilisés dans SOAP sont « **xsd** » et « **xsi** ».

* **xsd namespace** précise que les balises proviennent de la définition de schéma XML.
* **xsi namespace** indique que les balises viennent d'une instance d'un schéma XML.

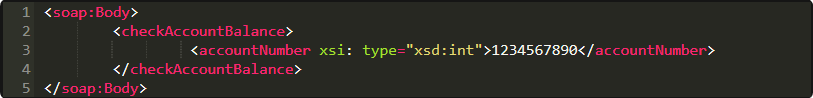
# Le corps SOAP

Le corps SOAP est un élément obligatoire dans le message SOAP. Il contient l'information destinée au receveur.

Le corps (*body*) doit fournir le nom de la méthode invoquée par une requête ainsi que les paramètres associés a celle-ci.

Le contenu du corps SOAP est utilisé pour spécifier un appel de méthode à un ordinateur distant avec les valeurs de paramètre. Par exemple, la demande du solde d'un compte bancaire.

L'extrait suivant représente un corps SOAP qui fait appel de procédure distante (RPC) à une méthode appelée **checkAccountBalance().**



Le corps du message SOAP commence avec la balise **<soap:Body>** et se termine avec la balise **</soap:Body>.**

L'élément **<checkAccountBalance>** fournit le nom de la méthode à appeler : **checkAccountBalance**. L'élément **accountNumber** est un paramètre qui est passé dans la méthode **checkAccountBalance**.

Les attributs **xsi** et **xsd** définissent les espaces de noms qui vont être utilisés dans le corps du message. La définition de xsi permet d'utiliser **xsi:type** dans le corps du message, **le xsd:int** signifie que cette valeur est de type **entier**. 1234567890 est la valeur donnée au paramètre.

L'ensemble de ces caractères représente un appel de méthode qui a la forme suivante en langage C :

exampleC.png

Une différence importante entre SOAP et XML-RPC est que les méthodes SOAP prennent des paramètres nommés. L'ordre des paramètres, lui, n'a pas d'importance, contrairement à XML-RPC.