Chapitre 8: Architecture Web Service

D’après le cours de Patrick Pleczon

Web service

- Définition du W3C
  - Entité logicielle conçue pour supporter des interactions machine-machine à travers un réseau.
  - A une interface décrite dans un formalisme interprétable par une machine (WSDL).
  - Permet à d'autres logiciel d'interopérer avec lui en suivant la prescription définie dans sa description et en utilisant des messages SOAP.

XML

- Langage issu de SGML. (Standard Generalized Markup Language)
- Contient des données étiquetées:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <!-- Edited with XML Spy v4.2 -->
    - <CATALOG>
    - <CD>
      <TITLE>Empire Burlesque</TITLE>
      <ARTIST>Bob Dylan</ARTIST>
      <COUNTRY>USA</COUNTRY>
      <COMPANY>Columbia</COMPANY>
      <PRICE>10.90</PRICE>
      <YEAR>1985</YEAR>
      </CD>
    - <.....
```

Communications basées sur XML

- XML-RPC
  - \+ Simple
  - \+ Léger
  - − N'évolue plus
- SOAP (Simple Object Access Protocol)
  - \+ Léger
  - \+ Plus complet que XML-RPC (gestion erreur, gestion types utilisateur)
  - ± En évolution
  - − Plus complexe que XML-RPC

Origine de SOAP

- Créé pour les Services Web
- Besoins :
  - Protocole multi-plateformes, multi-langages
  - Peu coûteux à mettre en oeuvre
  - Possibilité de fonctionner en environnement sécurisé (firewalls)
  - Respect des formats d'échanges du Web (XML)
  - Possibilité de choisir différents protocoles transport
  - Specification non propriétaires (w3c)

Autour de SOAP

- SOAP n'est pas utilisé seul dans le monde des services Web :

La « pile des services Web »

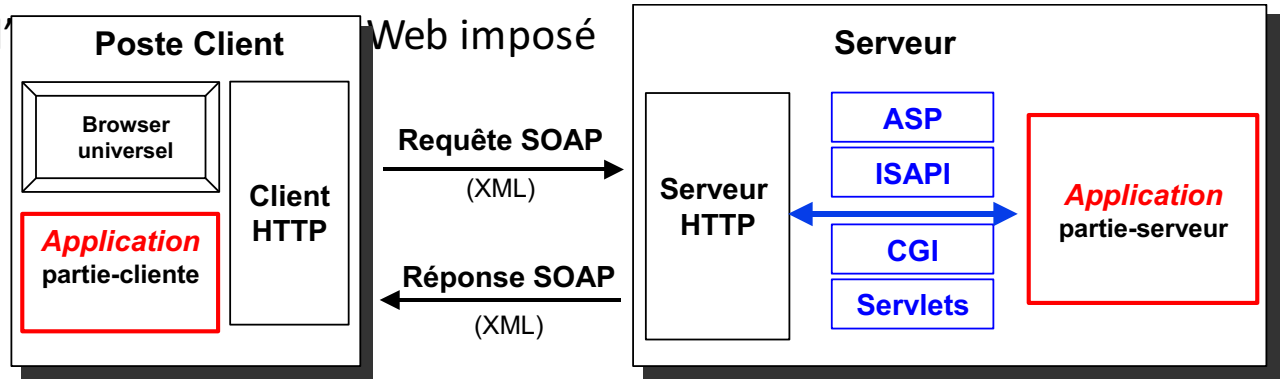
Annuaire : publication et recherche de services	**UDDI**
Description des services	**WSDL**
Description des échanges	**SOAP**
Format de données	**XML**
Communication	**Internet**

SOAP : les dates

- XML : Fev. 1998 → v1.0
- SOAP :
  - Déc. 1999 → v1.0
  - Mai 2000 → v1.1
  - Déc. 2001 → v1.2 Draft
  - Oct. 2003 → v1.2

SOAP

- Protocole de transmission de messages s'appuyant sur XML + protocole transport
- Transmission unidirectionnelle ou RPC
- Indépendant du langage/plate-forme
- Pas de modèle de programmation imposé
- Pas d'API/Run-Time imposé
- Pas d' Web imposé



SOAP : quelques kits de développement

- Le plus répandu : Apache Axis (Java).
- JWSDP de SUN.
- GLUE de The Mind Electric (Java).
- WASP de Systinet (Java & C++).
- GSOAP (C++).

Performances de SOAP

Extrait étude IBM (http://www-106.ibm.com/developerworks/webservices/library/ws-pyth9)

Techno	Temps CNX	Envoi chaîne (21,000 char)	Réception chaîne (22,000 char)	Envoi de 5,000 entiers	Taille msg Envoyant 1,000 char	Taille msg Envoyant 100 int
Raw sockets	0.002242	0.001377	0.001359	6.740674	2 279	85 863
CORBA	0.000734	0.004601	0.002188	1.523799	2 090	27 181
XML-RPC	0.007040	0.082755	0.050199	100.33721	4 026	324 989
SOAP	0.000610	0.294198	0.279341	1 324.296	4 705	380 288

Sécurité dans SOAP

- SOAP assure uniquement la transmission de données
  - Le cryptage peut être assuré soit:
    - par le protocole transport (HTTPS, SMTP/PGP, …)
    - Soit par l'application .

- Firewalls
  - SOAP a été conçu pour pouvoir passer les firewalls (utilisation port HTTP)

    ➔ des éditeurs ont développés des firewalls « applicatifs » (ex : Westbridge Tech., Reactivity)

Message SOAP

Structure message SOAP :

- une déclaration XML (optionnelle) suivie de
- une Enveloppe SOAP (l'élément racine) composée de:
  - Un En-tête SOAP (optionnel)
  - Un Corps SOAP

Il n'est pas indispensable de connaître la structure des messages SOAP pour développer, les toolkits masquant tout cela !

Utilisation de SOAP = invocation d'un appel prenant comme paramètres la méthode distante (RPC) et ses paramètres.

Requête SOAP

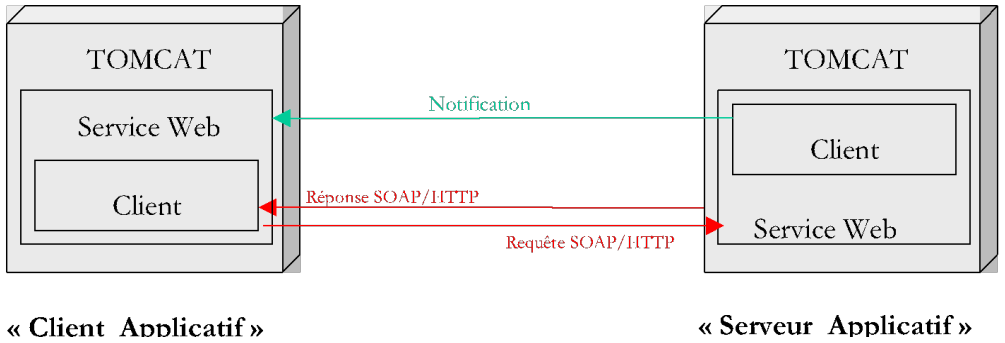
```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  <soap:Body xmlns:m="http://www.stock.org/stock">
    <m:GetStockPrice>
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>
</soap:Envelope>
```

Réponse SOAP

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-
encoding">
  <soap:Body xmlns:m="http://www.stock.org/stock">
    <m:GetStockPriceResponse>
      <m:Price>34.5</m:Price>
    </m:GetStockPriceResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP : asynchronisme (Callbacks)

- Ce mécanisme, peut être complexe à mettre en oeuvre avec certains toolkits.
- Pas de mécanisme standard.
- Souvent nécessaire d'utiliser un serveur WEB côté client.



SOAP : fiabilité

- SOAP/HTTP pas fiable pas nature
- peut le devenir à travers une séquence de confirmations et d'acquittements SOAP
  - inconvénient : la partie applicative doit gérer ce mécanisme

- SOAP/SMTP pas plus fiable (aucune garantie qu'un e-mail arrive à bon port)
  - mécanisme de confirmation équivalent plus lourd et plus complexe à mettre en œuvre.

Exemple avec Axis : code serveur

```
public class ElectionWS {

        public ElectionWS() {
                super();
        }

        public long votes(String candidat) {
                .....
        }
}
```

Exemple avec Axis : Déploiement (version simple)

- Renommer le fichier ElectionWS.java en ElectionWS.jws.
- Le copier dans le répertoire axis.

Exemple avec Axis : code client

```
import javax.xml.namespace.QName;
import org.apache.axis.client.Call;
import org.apache.axis.client.Service;
public class TestElectionWS {
        public TestElectionWS() {
                super();
        }
          public static void main(String [] args) {
          try {
          String endpoint = "http://localhost:8080/axis/ElectionWS.jws";
           Service  service = new Service();
           Call     call    = (Call) service.createCall();
           call.setTargetEndpointAddress( new.java.net.URL(endpoint) );
           call.setOperationName(new  QName("http://soapinterop.org/", "votes"));

           Long ret = (Long) call.invoke( new Object[] { "LeCandidat" } );
           System.out.println("Sent ' LeCandidat', got '" + ret.longValue() + "'");
           } catch (Exception e) {…
```

Axis : autres outils

- WSDL2Java
  - Génération de classes Java à partir de WSDL
- Java2WSDL
  - Génération de WSDL à partir d'une interface Java.
- SOAPMonitor
  - Monitoring des messages SOAP

Conclusion sur SOAP

- Jeune (?) (v1.2 a mis 2 ans pour passer de la 1ère « draft » à la définitive)

+ Simplicité de mise en œuvre (avec le bon toolkit),
+ Prévu pour fonctionner en environnements hétérogèmes (indépendant des langages, plate-formes, …)
+ Service et Clients sont déchargés de la gestion des connexions,
+ Possibilité de manipuler des types complexes, des fichiers joints, des callbacks, …

− Pas de mode connecté ➔ problème de gestion de la fiabilité des transmissions,
− Maturité toute relative, faible niveau documentaire de certains outils

SOAP : le bilan

- Intéressant si :
  - pas besoin de mode connecté
  - essentiellement communication de type requêtes/réponses
  - Pas de flux très contraints.
- A garder à l'esprit :
  - verbeux et moyennement performant
  - Bien choisir le toolkit !