

Les Services Web

Jean-Pierre BORG
EFORT

<http://www.efort.com>

1 Introduction

Un "Service Web" est une application logicielle à laquelle on peut accéder à distance à partir de différents langages basés sur XML.

Un "Service Web" est identifié par une URL, comme n'importe quel site Web. Il s'exécute sur un "Serveur d'Applications". Peu importe l'ordinateur, le système d'exploitation ou le langage utilisés par le Client ! Une application peut ainsi utiliser plusieurs "Services Web" s'exécutant sur des serveurs distants.

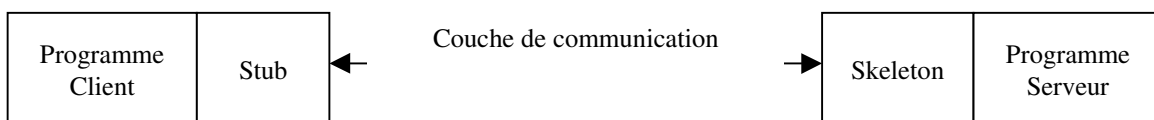
De nombreuses normes sous-tendent cette architecture : "SOAP" pour l'échange de messages, "XML" langage de base pour décrire tous les documents sur lesquels les messages sont construits, "HTTP" pour transporter les messages, "WSDL" pour décrire les services et enfin "UDDI" pour les publier.

L'approche "Services Web" constitue un changement fondamental dans la manière de concevoir et réaliser les applications informatiques et de programmer les ordinateurs. A court terme, il est probable que cette approche se substitue aux architectures et systèmes basés sur des LAN ou sur Internet.

2 Comparaison avec d'autres technologies

Exécuter des procédures à distance n'est pas une révolution. Plusieurs autres technologies le permettent. Ces technologies reposent sur deux types d'architecture :

2.1 Architecture "Stub/ Skeleton"



Cette architecture a pour but de rendre "simple" la partie "Client". Le "Stub" est une portion de code installée sur la machine client, qui identifie les objets et les méthodes disponibles sur le serveur, code les appels vers celui-ci dans un format que le serveur comprend et décode les réponses du serveurs dans un format accessible au client. Le "Skeleton" joue un rôle symétrique sur le serveur.

La connexion entre le client et le serveur est une connexion permanente pendant toute la durée de la session. Ceci limite le nombre de clients qu'un serveur peut traiter. Les appels des méthodes s'effectuent en "mode synchrone" et ignorent le "mode message" (MQSeries d'IBM ou Java Message Service).

Les données sont codées en binaire, créant ainsi des problèmes de compatibilité inter plates-formes. La validation des données est reportée au niveau applicatif.

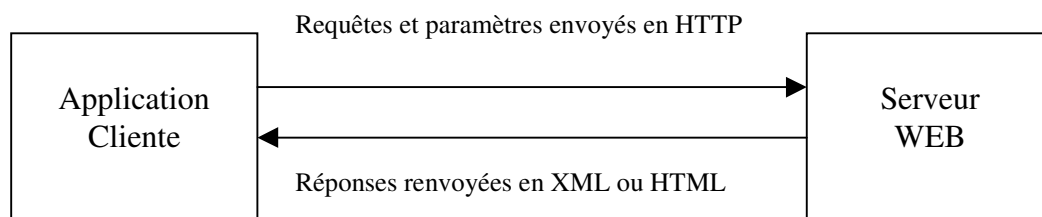
Les différences entre les solutions basées sur cette technologie proviennent de la façon dont les données sont codées, ainsi que des transports utilisés.

Les solutions les plus connues basées sur cette technologie sont CORBA, RMI et DCOM.

Par rapport aux Services Web, ces solutions, en particulier CORBA, présentent l'avantage d'être plus matures, mais elles sont rarement inter-opérables et peuvent poser des problèmes de sécurité car, la plupart du temps, les données sont transférées en clair sur le réseau.

2.2 Architecture basée sur des transactions HTTP

Cette architecture utilise un serveur Web, tel que IIS (Microsoft) ou Apache. Les clients communiquent avec le serveur en utilisant HTTP ou HTTPS. La connexion est rompue à la fin de la transaction (et pas de la session), ce qui permet de traiter un plus grand nombre de clients au détriment des performances si un client effectue un grand nombre de transactions. Cette architecture présente de plus l'inconvénient de ne pas avoir de répertoire référençant les services offerts. Ainsi, une collaboration étroite est nécessaire entre les clients et les fournisseurs de services.



Les solutions les plus connues basées sur cette technologie sont CGI, Servlets / JSP, ASP et PHP.

On a de nombreuses limitations, à savoir :

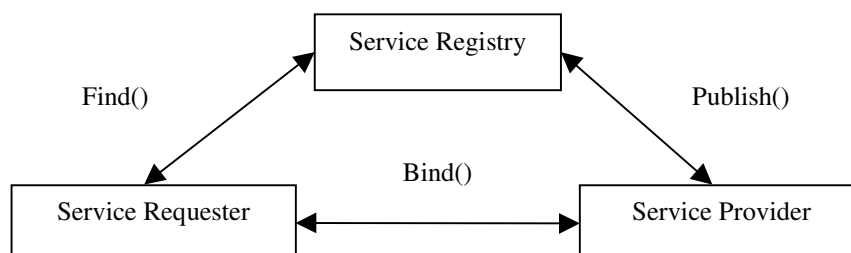
- types des données pauvres,
- pas de format accepté pour échanger des données,
- pas d'API bien documentée pour décrire les services,
- pas de répertoire pour trouver les services.

3 Architecture et composants clés

3.1 Architecture : le modèle SOA

L'objet des Services Web est la communication d'application à application (A2A) sur Internet. Le but est de faciliter l'intégration des applications d'entreprise (EAI) et le e-commerce spécialement en "Business To Business" (B2B). Pour ce faire,

- l'architecture des Services Web doit supporter les transactions asynchrones ("mode message") aussi bien que les transactions synchrones ("Remote Procedure Call" ou RPC),
- le codage des données est effectué en XML, ce qui conduit à des documents que l'homme et la machine peuvent interpréter, facilitant ainsi le débogage,
- des données arbitraires peuvent être représentées et validées au niveau système,
- les systèmes doivent être inter-opérables.



Le modèle SOA (Service Oriented Architecture) est le modèle des Services Web. C'est un modèle simple contenant trois entités et trois opérations :

Dans un environnement SOA, les ressources d'un réseau sont rendues disponibles en tant que services indépendants, auxquels on peut accéder sans rien connaître de la façon dont ils sont réalisés. Cet environnement a pour objet de permettre :

- la réutilisation de logiciels,
- un couplage "lâche" entre les serveurs et les clients,
- la découverte des services disponibles, afin d'être sûr que les utilisateurs potentiels découvriront le service dont ils ont besoin.

Le "Service Registry" est une "application bien connue" qui retourne au Requêteur les informations permettant de trouver un service à partir de critères de recherche, et de s'y connecter. Pour obtenir ce résultat, le fournisseur de service doit avoir publié ces informations préalablement.

Des organisations telles que OASIS et W3C dirigent les spécifications.

3.2 Composants mis en œuvre

3.2.1 SOAP 1.1 (Simple Object Access Protocol)

SOAP est un format de messages qui permet de transmettre en XML les appels de procédures distantes. SOAP permet aussi d'envoyer en XML un document entier d'un ordinateur à un autre.

SOAP 1.0 a été défini par l'IETF en Décembre 99. Après qu'IBM ait rejoint le mouvement, SOAP 1.1 a été spécifié par le W3C (World Wide Web Consortium) en Mai 2000. La version 1.2 est actuellement en cours de recommandation.

SOAP définit une enveloppe contenant un "en-tête" et un "corps". L'en-tête fournit les instructions indiquant comment traiter le message. Le corps contient l'appel de la procédure distante dans un sens et la réponse du serveur dans l'autre.

L'envoi de documents attachés à un message SOAP se fait en encapsulant le message SOAP et les documents attachés dans un document au format MIME (Multipurpose Internet Mail Extensions – rfc 2387) ou au format DIME (Direct Internet Message Encapsulation), qui est moins souple mais plus simple que MIME.

3.2.2 XML 1.0 (eXtensible Markup Language - 2^o édition)

XML est un langage "extensible" contenant des "tags", comme HTML. XML décrit la signification des données codées, indépendamment de la façon de les représenter (qui est décrite par d'autres spécifications, telles que CSS ou XSL, non utilisés par les Services Web). Contrairement à HTML dont les balises n'ont pour objet que de définir la "présentation du document", XML permet d'inventer à volonté de nouvelles balises pour décrire, de façon arborescente, toutes les informations élémentaires expliquant la signification des données.

Son objectif initial est de faciliter l'échange automatisé de contenus entre systèmes d'informations hétérogènes.

XML contient aussi un aspect supplémentaire appelé "schéma", qui précise la structure des "tags" pour qu'un document soit valide. Il est ainsi possible de restreindre les données acceptées dans un document.

Un document XML peut contenir d'autres documents XML. Pour éviter toute confusion entre les "tags" utilisés, chaque document contient un "espace de noms", ce qui rend ainsi les "tags" uniques à l'intérieur d'un document.

Les spécifications de XML sont aussi maintenues à jour par le W3C.

3.2.3 HTTP 1.1 (Hypertext Transport Protocol)

HTTP est aussi géré par le W3C. Plus aucune modification n'est prévue maintenant car HTTP est considéré complet et stable.

HTTP est un protocole permettant l'échange des requêtes et des réponses entre clients et serveurs, quel que soit le type des données. Extrêmement utilisé dans le monde du Web, il fonctionne au-dessus de TCP-IP (port 80). La plupart des pare-feux autorisent le transfert des messages correspondant à ce protocole.

SOAP 1.2 autorise d'autres protocoles que HTTP, mais actuellement, l'immense majorité des messages des Services Web est transportée par HTTP.

HTTP fonctionne en mode "Requête – Réponse". Une connexion HTTP ne dure que le temps d'émettre une requête et de recevoir entièrement sa réponse. Après que le client ait acquitté cette réception, le serveur ferme la connexion.

HTTP permet de gérer des sessions "avec états" par le mécanisme des "cookies" (rfc 2965), en ajoutant trois en-têtes contenant des informations d'état concernant les clients et serveurs participant à la session.

3.2.4 WSDL 1.1 (Web Services Description Language)

Cette spécification du W3C précise comment décrire un Service Web :

- méthodes acceptées et paramètres correspondants,
- réponses fournies en retour,
- protocoles et formats de données pouvant être traités,
- URLs du service.

Un document WSDL est un document XML contenant toutes les informations nécessaires pour contacter et utiliser un service, indépendamment de toute plate-forme ou langage de programmation.

L'utilisateur potentiel d'un Service Web doit d'abord obtenir le WSDL de ce service. Il créera ensuite (sur la plate-forme et dans le langage de son choix) le logiciel "Client" correspondant à son application, utilisant les "méthodes" que propose(nt) le (ou les) Service(s) Web, comme s'il s'agissait de "méthodes locales à son ordinateur". Plusieurs outils logiciels existent, permettant de traiter la communication avec le serveur, en utilisant les composants décrits précédemment. Voir § 4.

L'exemple ci-dessous expose une partie du document WSDL décrivant un Service Web "getReceivedSMS" permettant de lire un SMS reçu :

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- March 8, 2007 -->
<wsdl:definitions
  name="parlayx_sms_receive_interface"
  targetNamespace="http://www.csapi.org/wsdl/parlayx/sms/receive/v3_1/interface"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema/"
  xmlns:parlayx_sms_receive="http://www.csapi.org/wsdl/parlayx/sms/receive/v3_1/inte
rface"
  .....

  <wsdl:types>
    <xsd:schema elementFormDefault="qualified"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://www.csapi.org/schema/parlayx/sms/receive/v3_1/loc
al">
```

```

.....
    <xsd:complexType name="getReceivedSms">
        <xsd:sequence>
            <xsd:element name="registrationIdentifier"
                type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
</wsdl:types>

<wsdl:message name="ReceiveSms_getReceivedSmsRequest">
    <wsdl:part name="parameters"
        element="parlayx_sms_receive_local_xsd:getReceivedSms"/>
</wsdl:message>

.....

</wsdl:definitions>

```

3.2.5 UDDI 2.04 API (Universal Description, Discovery and Integration)

La spécification UDDI décrit un type de registre listant les Services Web disponibles. Différentes informations permettent de faciliter la recherche d'un service particulier. Les services enregistrés peuvent appartenir à l'une des trois catégories suivantes :

- "Public" : service ouvert à tous. Plusieurs éditeurs majeurs proposent de tels services, par exemple IBM et Microsoft.
- "Private" : service accessible seulement à l'intérieur d'une compagnie. Pour celle-ci, le catalogue UDDI permet la réutilisation de logiciels en interne.
- "Restricted" : service accessible seulement par certaines organisations autorisées (par exemple les fournisseurs et certains clients d'une entreprise).

Un fournisseur, ayant créé un Service Web "public", doit l'enregistrer auprès du "Service Registry" de son choix, par exemple le "Microsoft public registry". Ce "Registry" réplique chaque nuit ses entrées sur les autres "public registry" existants. Ainsi, un client potentiel, interrogeant quelques jours après le "IBM public registry", trouvera ce nouveau service Web et pourra l'utiliser s'il le souhaite.

La spécification UDDI part d'une initiative lancée par ARIBA, Microsoft et IBM. Cette spécification n'est pas gérée par le W3C mais par le groupe appelé OASIS (Organization for the Advancement of Structured Information Standards).

3.3 Sécurité des Services Web

Les points importants à prendre en compte, en ce qui concerne les Services Web, sont :

- l'authenticité : le client et le serveur sont ceux qu'ils affirment être,
- la discrétion : les requêtes et les réponses ne sont pas écoutées par des personnes non autorisées,
- l'intégrité : ces requêtes et réponses ne sont pas altérées,
- la non-répudiation : le client et le serveur peuvent prouver, chacun pour sa partie en cas de contestation, que la transaction a réellement été effectuée.

L'infrastructure de sécurité d'Internet a été créée bien avant l'apparition des Services Web. Cette sécurité repose sur des "Certificats à Clés Publiques" (ITU-T X 509).

Un des moyens les plus simples pour sécuriser les messages d'un Service Web est l'utilisation des SSL (Secure Sockets Layer). Le protocole de transport utilisé est alors HTTPS au lieu de HTTP. Toutes les données qui s'échangent sur Internet sont cryptées. Par contre, cette méthode ne permet pas l'authentification des parties, ne garantit pas leur intégrité et ne protège pas contre la répudiation.

Plusieurs autres solutions émergent pour traiter ces problèmes, mais il est encore trop tôt pour déclarer un vainqueur. Citons :

- la signature XML (OASIS), permettant de signer une partie du document,
- le cryptage XML (W3C),
- l'authentification par le protocole SAML (Security Assertion Markup Language -- OASIS).

Enfin, un dernier standard apparaît, appelé "WS-Security", qui adressera l'ensemble de ces problèmes.

4 Outils de création des Services Web

Différents éditeurs proposent des outils pour créer des Services Web. Citons en quelques uns :

- Apache AXIS (Apache eXtensible Interaction System) : outil "Open Source", pouvant être téléchargé librement,
- Java Web Services Developer Pack, fourni par SUN,
- Microsoft Plate-forme .NET (par exemple Visual Studio 2005 ou 2008),
- BEA Plate-forme WebLogic,
- IBM WebSphere,
- SOAP::Lite, plate-forme utilisant le langage PERL.....

5 PARLAY-X : les Services Web dans le monde des télécommunications

Les spécifications PARLAY X 3.0 ont été définies conjointement par l'ETSI (European Telecommunications Standards Institute) et le 3GPP (Third Generation Partnership Program). Elles sont maintenant publiques.

Selon la mise à jour de Juin 2007, les services Web sont regroupés en vingt parties, couvrant une partie importante des télécommunications et des services, en particulier dans le monde du mobile :

Part 1: Common

Part 2: Third Party Call

Part 3: Call Notification

Part 4: Short Messaging

Part 5: Multimedia Messaging

Part 6: Payment

Part 7: Account Management

Part 8: Terminal Status

Part 9: Terminal Location

Part 10: Call Handling

Part 11: Audio Call

Part 12: Multimedia Conference

Part 13: Address List Management

Part 14: Presence

Part 15: Message Broadcast

Part 16: Geocoding

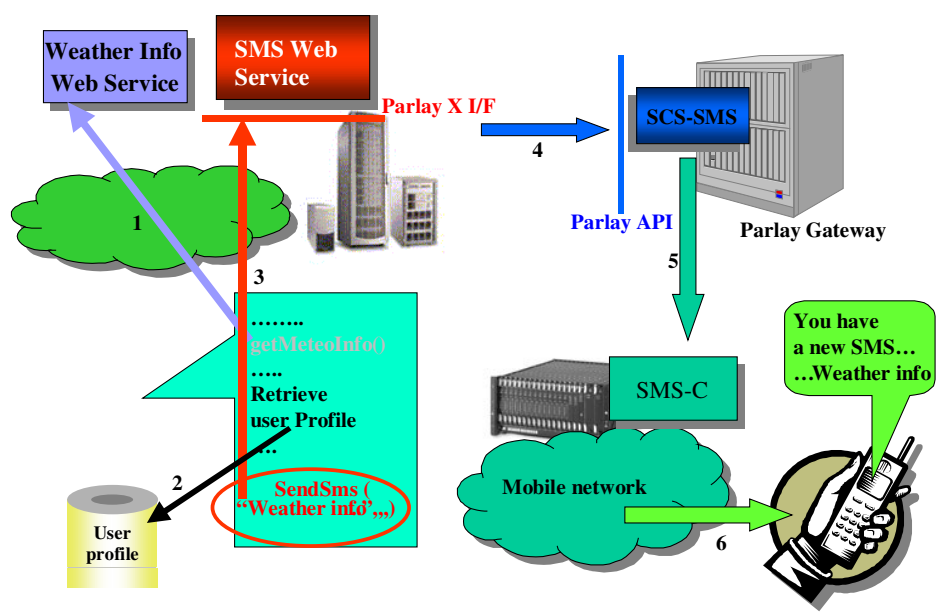
Part 17: Application-driven Quality of Service

Part 18: Device Capabilities and Configuration

Part 19: Multimedia Streaming Control
Part 20: Multimedia Multicast Session Management

Le schéma ci-dessous montre le fonctionnement d'une application basée sur des Services web, qui a pour objet de prévenir par mail ses abonnés (par exemple des agriculteurs de Champagne) en cas d'apparition de certains événements Météo (ex. prévision de gel nocturne) :

- 1/ interrogation d'un Service Web fournissant les prévisions Météo,
- 2/ recherche des caractéristiques des abonnés concernés,
- 3/ composition d'un SMS adapté,
- 4/ envoi de ce SMS en utilisant un Service Web parmi ceux proposés par les spécifications Parlay X,
- 5/ acheminement de ce SMS par le réseau de l'opérateur mobile concerné, comme s'il avait été émis d'un portable quelconque.



6 Références

- <http://www.w3.org/TR/2007/REC-sawsdl-20070828/> : Semantic Annotations for WSDL and XML Schema (W3C Recommendation 28 August 2007).
- <http://www.w3.org/TR/2007/REC-soap12-part0-20070427/> : SOAP Version 1.2 (W3C Recommendation 27 April 2007).
- <http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626> : WSDL Version 2.0 W3C Recommendation 26 June 2007).
- <http://www.w3.org/TR/REC-xml> : spécification XML.
- <http://www.w3.org/TR/xmlschema-1> et <http://www.w3.org/TR/xmlschema-2> : Schémas XML.
- <http://www.w3.org/TR/WD-xsl> : spécification XSL.
- <http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.htm> : spécification UDDI (OASIS).