

SOAP ou REST, QUE CHOISIR?

Eric van der Vlist (vdv@dyomedea.com)

SOAP ou REST, que choisir?

Web Services Convention

Juin 2004

COMPARER CE QUI EST COMPARABLE

Principes architecturaux v.s. spécifications

On a d'un côté REST qui est un ensemble de principes architecturaux et d'un autre côté SOAP, WSDL et UDDI qui font partie d'une « pile » de spécifications décrivant très concrètement des formats de messages et des protocoles.

On ne peut donc pas directement comparer REST et SOAP/WSDL/UDDI.

SOAP/WSDL/UDDI sont-ils REST?

La première question qu'il semble légitime de se poser pour clarifier le débat est d'examiner dans quelle mesure SOAP, WSDL et UDDI sont (ou peuvent être) conformes aux principes énoncés par REST.

SOAP est-il REST?

Pour examiner dans quelle mesure SOAP est compatible avec les principes de REST, nous allons confronter cette spécification aux principales caractéristiques de REST.

Sans état et sans connexion : OK

La spécification SOAP par elle-même n'introduit pas de gestion d'état ou de connexion et elle est donc conforme à REST à ce niveau.

Utilisation des méthodes HTTP : très mauvaise

Toutes les requêtes SOAP se font en utilisant la méthode POST pour envoyer un document XML. A ce niveau, on peut donc dire que SOAP utilise très mal les différentes méthodes HTTP puisque même quand il s'agit de consulter une ressource (comme un cours de bourse), un client SOAP va « poster » un message au lieu de lire la ressource au moyen d'une requête « GET ».

Utilisation des URIs : mauvaise

Rien n'interdit bien entendu d'utiliser des URIS dans un message SOAP, mais elles sont masquées dans le message au lieu d'être exposées dans les requêtes HTTP.

Elles échappent donc aux outils d'administration, de journalisation et d'optimisation (cache) du web.

Bilan : SOAP est mal adapté à REST

On peut donc dire que SOAP est mal adapté à l'implémentation de systèmes conformes

aux principes de l'architecture REST.

WSDL est-il REST?

Sans état et sans connexion : OK

Les bindings de WSDL (SOAP, HTTP et Mime) sont tous des bindings sur des protocoles sans état et sans connexion. WSDL est donc conforme à REST sur ce point.

Utilisation des méthodes HTTP : mauvaise

Les bindings HTTP de WSDL ne supportent que les méthodes « GET » et « POST ». Cela constitue certes un progrès par rapport à SOAP qui n'utilise que la méthode « POST », mais une utilisation optimale des méthodes HTTP supporterait l'ensemble des méthodes HTTP et notamment les méthodes « PUT » et « DELETE ».

De plus, ces bindings ne donnent pas accès aux headers HTTP, ce qui ne permet pas de spécifier des mécanismes tels que la négociation de contenu ou la gestion du cache.

Utilisation des URIs : moyenne

Les URIs interviennent à deux niveaux dans WSDL : dans les messages (de manière opaque, comme dans le cas de SOAP) et en tant qu'URI des requêtes HTTP.

Les règles de constitution des requêtes HTTP semblent très figées : l'URL calculée est nécessairement relative à l'URL de base du service et la manière de la déterminer doit rentrer dans un des deux cas de figure prévus par la spécification.

Cette rigidité rendrait difficile mais sans doute pas impossible l'utilisation de WSDL pour décrire les deux exemples de services web REST de Paul Prescod.

Bilan : WSDL est très moyennement adapté à REST

Si WSDL est sans aucun doute un peu mieux adapté à REST que SOAP, l'adéquation reste néanmoins très moyenne.

UDDI est-il REST?

La vocation de UDDI étant très différente de celle de SOAP et WSDL, nous allons modifier la nature des questions à nous poser.

Modèle de données UDDI : OK

Le modèle de données UDDI est suffisamment générique pour que l'on puisse stocker la description de Services Web REST dans un annuaire UDDI (nous avons vu que la description technique (le tModel) n'était qu'une référence à une description exprimée de

manière quelconque.

Interface d' accès à l'annuaire : non conforme à REST

Par contre, l'interface permettant d'accéder à l'annuaire n'est pas conforme à REST puisqu'elle est entièrement basée sur SOAP et WSDL.

Bilan : UDDI peut stocker les descriptions de Services Web REST mais ne permet pas d'y accéder en suivant les principes de REST

Un annuaire UDDI est adapté au stockage de descriptions de Services Web REST (il reste néanmoins à définir le format que pourrait prendre une telle description) mais l'accès au répertoire ne peut pas se faire en suivant les principes de REST (sauf à définir une nouvelle interface pour cela).

SOAP/WSDL/UDDI sont-ils REST?

Nous pouvons maintenant répondre à la question :

SOAP : non

WSDL : très moyennement

UDDI : peut l'être

SOAP/WSDL/UDDI ou REST?

Puisque nous venons de voir que ces deux architectures sont largement incompatibles, la question se pose donc bien en ces termes : faut-il choisir REST ou SOAP/WSDL/UDDI pour déployer des Services Web?

Quelle est votre vision?

Pour ce type de choix, il est bon de revenir aux visions qui sont à l'origine des différentes propositions. Sans vouloir se laisser intoxiquer par la « vaporware » qui peut embrouiller les messages, les visions originelles ont en général des implications pratiques dont il serait dommage de ne pas tenir compte!

Ouvrir le web aux applications (REST)

Si vous avez une stratégie Internet forte et que votre but principal est d' ouvrir l'accès de vos sites Web (internes ou externes) aux applications, vous êtes dans le champ de la vision REST.

Utiliser le Web pour échanger des objets (SOAP/WSDL/UDDI)

Si vous avez déployé des applications utilisant des technologies orientées objet de manière intensive et que votre but principal est de faire comme CORBA ou DCOM (éventuellement teinté d'une touche de « workflow ») en empruntant le Web, vous êtes dans le champ de la vision SOAP/WSDL/UDDI.

Administration Web, administration EAI

C'est une des conséquences les plus immédiates des différences de vision se retrouve au niveau de l'administration.

REST s' administre comme le Web

Puisque la base des Services Web REST est la requête HTTP avec sa sémantique d'origine, l'administration de ses services se fait de manière classique comme celle des pages web.

SOAP/WSDL/UDDI s' administrent comme un EAI

Dans la mesure où les requêtes sont cachées dans des documents XML, l'administration classique des serveurs web est déjouée et il faut concevoir et utiliser des outils et des procédures d'administration spécifiques qui s' apparentent plus à celles d'un EAI qu'à celle d'un site Web.

Informatique lourde, informatique légère

On voit se dessiner une autre ligne de fracture entre REST et SOAP/WSDL/UDDI autour des notions d'infrastructure ou de développements lourds (heavyweight) ou légers (lightweight).

REST est « tendance léger »

Un service REST se met en place comme une page Web. Cela ouvre la porte à des développements rapides, avec des cycles courts, au prototypage, aux tests possibles avec un simple navigateur et à l'utilisation de langage de scripts (PHP, Perl, Python, Rubis...).

SOAP/WSDL/UDDI sont « tendance lourde »

Un service SOAP se met en place comme le déploiement d'un objet. Cela ouvre la porte à une phase préalable de modélisation, des développements aux cycles plus longs et à l'utilisation de langages de programmation à typage statique (Java, C#, ...).

L'EXPÉRIENCE D'AMAZON.COM

Exposer le SI aux partenaires

Le succès d'amazon.com s'appuie largement sur la participation de ses partenaires et amazon.com a souhaité exposer une partie de son système d'information à ses partenaires.

Cela couvre bien entendu l'accès au catalogue mais également la vente (la gestion du panier de commande a ainsi été exposée) et les achats (amazon.com sert de plateforme de vente pour de nombreux partenaires qui commercialisent des produits sur amazon.com).

Sous les deux architectures (REST et SOAP)

Amazon.com a décidé de fournir les mêmes fonctionnalités sous forme de services Web SOAP et REST.

REST = 80 %, SOAP = 20% des accès

Les utilisateurs de ces services Web ont massivement opté pour l'architecture REST puisque 80% des accès sont effectués par le canal des Services Web REST.

SOAP semble l'apanage de JAVA et C#

Amazon.com indique que la majorité des utilisateurs de SOAP sont des programmeurs en langages statiquement typés (Java ou C#) alors que les utilisateurs de REST semblent plus éclectiques.

Mieux intégré au Web, REST permet de construire des sites Web virtuels

Outre la simplicité (notamment au niveau des tests), un des intérêts de l'interface REST est qu'elle s'intègre mieux au Web. Ainsi elle permet de faire réaliser une transformation XSLT sur les résultats d'une requête effectuée sur le serveur amazon.com.

Cette fonctionnalité permet aux partenaires d'amazon.com de créer des sites web virtuels fonctionnant sur le principe que nous avons montré dans l'exemple « orienté GET » de Paul Prescod : le résultat de la transformation XSLT appliqué sur une requête contient les liens qui permettent à l'utilisateur d'effectuer d'autres requêtes.