

Veille technologique REST vs SOAP :

Introduction :

Simple Object Access Protocol (SOAP) et Representational State Transfer (REST) sont deux solutions pour un même objectif : accéder à des services Web.

SOAP est un protocole d'accès aux services Web basé sur des normes qui existe depuis un certain temps et bénéficie de tous les avantages de l'utilisation à long terme.

REST est le nouveau venu. Il vise à résoudre les problèmes rencontrés avec SOAP et fournir une méthode vraiment simple, d'accéder à des services Web.

Les deux techniques ont des problèmes à prendre en compte au moment de décider quel protocole utiliser.

I - Présentation de REST et de SOAP

Caractéristiques de SOAP	Caractéristiques de REST
<ul style="list-style-type: none">- SOAP est un protocole- SOAP repose exclusivement sur XML pour fournir des services de messagerie.- Un client développé avec le protocole SOAP ressemble à un logiciel d'ordinateur, car il est étroitement lié au serveur. Si une modification est effectuée d'un côté ou de l'autre, l'ensemble peut ne plus fonctionner. Il faut effectuer des mises à jour du client s'il y a des changements sur le serveur et vice-versa.- Le WSDL (Web Services Description Language) est un autre fichier qui est associé à SOAP. Il donne une définition de la façon dont le service Web fonctionne, de sorte que lorsqu'on crée une référence à	<ul style="list-style-type: none">- REST (REpresentational State Transfer) constitue un style architectural et un mode de communication fréquemment utilisé dans le développement de services Web.- REST utilise différents formats : HTML, XML, JSON- REST est souvent utilisé dans les applications mobiles, les sites Web des réseaux sociaux, les outils composites (mashup) et les processus métier automatisés.- Un client de type REST sait utiliser un protocole et des méthodes standardisées. Son application doit rentrer dans ce modèle. On ne crée pas de méthodes supplémentaires, on utilise les méthodes standardisées que l'on développe

<p>celui-ci, l'IDE peut complètement automatiser le processus. Donc, la difficulté d'utiliser SOAP dépend dans une large mesure du langage utilisé.</p> <ul style="list-style-type: none"> - Il n'est pas nécessaire d'utiliser SOAP avec le protocole HTTP (HyperText Transfer Protocol). Il existe une possibilité pour l'utilisation de SOAP avec le protocole SMTP (Simple Mail Transfer Protocol). - Concernant la sécurité, SOAP supporte WS-Security et SSL. 	<p>pour le type de média dont on a besoin.</p> <ul style="list-style-type: none"> - Application respectant pleinement REST = application RESTful : client ne doit connaître que le point d'entrée de l'API ainsi que le type de données qui est attendu. - REST peut utiliser quatre différents méthodes HTTP (GET, POST, PUT et DELETE) pour effectuer des tâches (voir Annexe 1). - Concernant la sécurité, REST supporte SSL et HTTPS.
---	---

II - SOAP, avantages et inconvénients

SOAP	
<u>Avantages</u>	<u>Inconvénients</u>
<ul style="list-style-type: none"> - SOAP peut générer directement un client à partir de métadonnées générées côté serveur à l'aide du langage WSDL (Web Service Description Language). - SOAP est fortement extensible, mais on peut utiliser uniquement les parties dont on a besoin pour une tâche particulière. - Des outils permettent de générer l'XML des requêtes, ce qui peut aider à réduire l'effort nécessaire pour créer la requête et pour analyser la réponse. Et quand on travaille avec les langages comme le .NET, nous ne voyons même pas le XML. - Concernant la gestion des 	<ul style="list-style-type: none"> - Un client SOAP doit tout savoir des éléments qu'il va utiliser pendant son interaction avec le serveur, sinon cela ne fonctionnera pas. - Plus lourd que REST car consomme plus de bande passante et de ressources. - Les services SOAP sont beaucoup plus difficiles à faire évoluer que les services RESTful. - Le XML utilisé pour envoyer des requêtes et recevoir des réponses dans SOAP peut devenir extrêmement complexe. Dans certains langages de programmation, nous devons

<p>erreurs, s'il y a un problème avec la requête, la réponse contient des informations d'erreur que nous pouvons utiliser pour résoudre le problème. Étant donné que nous ne pourrions accéder au service Web, cette particularité est extrêmement importante, sinon nous devrions deviner les raisons pour lesquelles les requêtes ne fonctionnent pas. Le rapport d'erreurs fournit même des codes normalisés de sorte qu'il est possible d'automatiser des tâches de gestion des erreurs dans le code.</p> <ul style="list-style-type: none"> - Standardisé 	<p>construire ces requêtes manuellement, ce qui devient problématique parce que SOAP ne tolère pas les erreurs.</p> <ul style="list-style-type: none"> - SOAP est encombrant et difficile à utiliser. Par exemple, travailler avec SOAP en JavaScript signifie écrire beaucoup de code pour effectuer des tâches extrêmement simples parce qu'il faut créer à chaque fois la structure XML requise. Il faut avoir des connaissances solides avant d'utiliser SOAP.
---	---

III - REST, avantages et inconvénients

REST	
<p><u>Avantages</u></p> <ul style="list-style-type: none"> - Il y a beaucoup moins de couplage entre le client et le serveur : un client peut utiliser un service de type REST sans aucune connaissance de l'API. - Plus léger que SOAP car consomme moins de bande passante (utilisation plus pratique sur Internet). - Aucun besoin d'outils coûteux pour interagir avec le service Web. - La courbe d'apprentissage semble relativement réduite. Ainsi, les développeurs sont capables d'utiliser REST dans 	<p><u>Inconvénients</u></p> <ul style="list-style-type: none"> - REST est moins sécurisé que SOAP. Il utilise l'HTTPS mais si le serveur n'est pas en HTTPS, alors le web service non plus.

<p>des applications plus rapidement qu'avec SOAP car il n'est pas nécessaire de connaître l'API. Cela se traduit par un gain de temps, et donc d'argent.</p> <ul style="list-style-type: none"> - REST est conçu pour une utilisation sur un Internet/Web ouvert. Il constitue un meilleur choix pour les applications Web et à plus forte raison pour les plateformes en Cloud. - Au lieu d'utiliser XML pour faire une demande, REST repose sur une simple URL. - Contrairement à SOAP, REST n'a pas à utiliser XML pour fournir la réponse. On peut trouver des services Web en REST qui ont comme sortie des données au format CSV (Command Separated Value), JavaScript Object Notation (JSON) ou encore Really Simple Syndication (RSS). On peut donc obtenir la sortie dont on a besoin sous une forme qui est facile à analyser avec le langage dont nous avons besoin pour notre application. 	
---	--

Conclusion :

Certaines personnes estiment que l'un des protocoles est meilleur que l'autre, cette affirmation reste incohérente. En effet, chaque protocole présente certains avantages et inconvénients tout aussi contraignants. Vous devez choisir entre SOAP et REST en vous basant sur le langage de programmation que vous utilisez, l'environnement dans lequel vous l'utilisez ainsi que les exigences de l'application.

A l'avenir, REST devrait continuer son expansion, lorsque les entreprises chercheront des interfaces ouvertes et bien définies pour leurs services d'application

et d'infrastructure. L'essor du Cloud computing public et Cloud privé, qui détermine en grande partie cette demande est une garantie de croissance future.

Pour moi, la technologie REST est la plus efficace pour notre PPE car c'est une méthode simple et pratique pour accéder à des web services. De plus, REST peut être sécurisé grâce au HTTPS et supporte de nombreux formats.

Annexe 1 :

Protocol Layering

