



Diploma Project

Telecommunication equipment
management using web services

Clients tools for MileGate

Version V1 (C.Tools)

Professeurs:

Philippe Joye
François Buntschu

Mandatory:

Daniel Gachet

Expert:

Nicolas Mayencourt

Students:

Thierry Kiki
David Schneider



Table of Contents

1	Introduction	3
2	Interoperability common problems between web services and SOAP protocol	3
2.1	Transport Problem	4
2.2	XML Problems	4
2.3	Problèmes du protocole SOAP	5
3	Web Services Interoperability Organisation (WS▶I)	6
4	Présentation de quelques frameworks	7
5	Etude frameworks	8
5.1	Axis2	8
5.2	CXF	8
5.3	WSO2	8
5.4	METRO	8
5.5	WCF	8
6	Tests Tools	8
6.1	WS-I tools	8
6.2	Others tools	8
7	Conclusion	8
	Annexes	9
7.1	Revision history	9
7.2	References	9



1 Introduction

There are few available platforms to create web applications. In the past, each application used its own specific protocol for service integration between devices. For that reason, applications that used different platforms could not communicate or shared data. This problem is called interoperability, which is the ability to communicate and share data effectively and efficiently. The awareness of those unfortunate limitations led to the standardization of data structure and shared data; thus, the study of web services observed in this project.

The objective of this project is to emphasize to the best of our (my) abilities the different aspects that could either prevent a good integration of web services or represent an obstacle to reach efficient software and traditional materials. In fact, the idea is to produce an interactive « Milegate » capable of understanding the service web standardized language that works with any client device regardless of the web tools used by the device.

2 Interoperability common problems between web services and SOAP protocol

Service web technologies such as protocol SOAP, WSDL Language and HTTP protocol are currently used to transfer messages between machines. Those messages vary in complexity. It ranges from the type of methodology to the submission of an order. A common function – of higher level – of web service requires the implementation of RPC communication type (Remote Procedure Call, a long distance procedure call that allows a computer to run a program on another computer.

For the rest of the paper, we will focus on a list of practical and frequently asked questions concerning interoperability ; mainly questions that relate to RPC communication type using protocol SOAP. « La messagerie avec SOAP, WSDL (Interface publique avec laquelle le client invoque le service dont il a besoin). »

Commentaire [HK1]: Ce n'est pas une phrase

It is important to remember that interoperability problems are not often linked to SOAP itself because it is « normalized »; on the contrary, it is linked to transport and core protocol XML used.

Therefore, we have the following problems:

- Transport (HTTP, SMTP, FTP..).
- Core XML



- Partial implementation or confusing integration of SOAP specification.

2.1 Transport Problem

The transport used for web services is very important. HTTP represents the most popular transport for RPC calls via SOAP. In addition, the embarked server that we will use for the prototype is HTTP based. This means that a reliable interoperability must exist between HTTP and SOAP.

A simple but widely popular example of HTTP interoperability problem relates to the usage of « SOAPAction » (only valid for SOAP 1.1). « SOAPAction » is a HTTP header that needs to be present in SOAP messages via HTTP. Numerous weights can be attributed to this header. An example is : SOAPAction: "http://tempuri.org/"

According to W3C specification, the weight of « SOAPAction » should be in quotes unless that weight is zero.

The most often encountered problem is the following : if a server requires that « SOAPAction » with weight zero, some clients will not be able to solve that case because all API client HTTP cannot define headers with weight zero. In this case, there are two possible solutions :

- **Settle** API clients (an actually difficult task)
- And/or make sure that none of the servers requires a « SOAPAction » of weight zero.

This problem is nonexistent for SOAP 1.2 because a new code HTTP (427) was introduced and submitted to IANA to show the client that the server application requires a « SOAPAction » replaced with an optional parameter « action ». See specification for more details.

2.2 XML Problems

The second type of interoperability problems relates to XML analysis and the management of XSD diagrams. SOAP uses at its base XML and XML diagrams ; so its interoperability depends on both interoperabilities.

An interesting example of interoperability problem involving both XML analysis and HTTP transports is « **la marque d'ordre de tri** » (BOM, Byte Order Mark). When data is sent through HTTP, it is possible to state the coding – such as UTF-16 or UTF-8 – in the header Content-Type. It is also possible to state the coding of a XML fragment by inserting a set of bytes. When UTF-16 is sent, BOM is necessary regardless of the presence of

Commentaire [HK2]: Traduction à revoir...
Supprimer l'exemple et donner un aspect beaucoup plus général sur l'encapsulation des msg soap dans le header http.

Commentaire [HK3]: Présenter aussi le problème sous un aspect plus général



coding in the header Content-Type (to point out big-endian or little-endian) ; but it is not the case with UTF-8. For example,

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

n++<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

  xmlns:xsd="http://www.w3.org/2001/XMLSchema"

  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tns="http://soapinterop.org/"

  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<tns:echoStringResponse>
<Return>string</Return>
</tns:echoStringResponse>
</soap:Body>

</soap:Envelope>
```

Les trois premiers caractères sont hexadécimaux pour la marque d'ordre de tri indiquant UTF-8, mais comme vous pouvez le voir, le Content-Type en fait également part. Certaines implémentations envoient la BOM pour UTF-8, même si elles n'en ont pas besoin. D'autres ne peuvent pas traiter XML avec une BOM. La solution est de ne l'envoyer que lorsque c'est nécessaire et de la gérer correctement. En effet, il est essentiel de bien gérer la BOM pour le traitement des messages UTF-16, car elle est requise dans ce cas. Indéniablement, il n'existe pas de solution idéale pour régler ces problèmes avant l'heure. Mais une fois qu'ils sont identifiés, le mieux est de se référer aux spécifications exactes (généralement consultables au W3C) qui décrivent les standards, puis de les appliquer scrupuleusement.

2.3 Problèmes du protocole SOAP

En lui-même, SOAP est relativement simple. Il requiert que les messages soient placés dans une enveloppe, avec le texte du message inclus dans un élément body.

Malheureusement, l'imprécision de certaines spécifications sont souvent la source de problème ; ce qui peut rendre parfois difficile la détermination correcte d'un comportement. En d'autres termes l'interprétation divergente de la spécification donne naissance à diverses implémentations.

En effet, SOAP propose des éléments facultatifs tels que les en-têtes et donne toute latitude quant à ce qui peut entrer dans l'élément body.



Un problème parmi tant d'autres souvent rencontré lié à la spécification est l'utilisation et l'interprétation de l'en-tête SOAP qui est facultative. L'attribut « mustUnderstand » en fonction de sa valeur spécifie si l'en-tête SOAP est optionnelle ou obligatoire.

Dans le cas où l'attribut « mustUnderstand » défini sur "1" ou "True" selon la version SOAP, le récepteur doit reconnaître l'information présente dans l'en-tête.

Un premier problème se situe déjà au niveau de la reconnaissance de l'en-tête et surtout de son traitement afin d'y extraire l'information nécessaire.

Un autre problème survient quand cet en-tête devra traverser plusieurs serveurs intermédiaires. Dans ce cas chaque serveur intermédiaire extrait ce qui le concerne et si nécessaire rajoute de l'information pour le serveur suivant.

Cette déduction logique mais non précisée par la spécification n'est pas toujours implémentée dans les API clients.

On voit de suite quelle peut être la conséquence d'une telle imprécision si une action particulière et importante doit être exécutée par un serveur final.

La liste suggestive ci-dessous propose une série de différents problèmes d'interopérabilité. Il s'agit notamment d'archives de recherche :

<http://groups.yahoo.com/group/soapbuilders>

<http://www.mssoapinterop.org/>

<http://www.xmethods.net/ilab/> .

En règle générale, la seule façon d'éviter tous ces genres de problèmes parmi une infinité de cas est de s'assurer que l'API utilisée est robuste et a déjà fait ses preuves sur le Web. Cependant de tels problèmes peuvent toujours persister malgré tout et dans ce cas le seul remède existant sera de procéder à des tests réels une fois l'application réalisée et le service défini afin de prévenir de telles anomalies.

3 Web Services Interoperability Organisation (WS▶I)

WS▶I is an open industry organization chartered to establish and document Best Practices for Web services interoperability.

It provides a Profiles and Testing tools that can be use by web service community to aid in developing and deploying interoperable Web services.

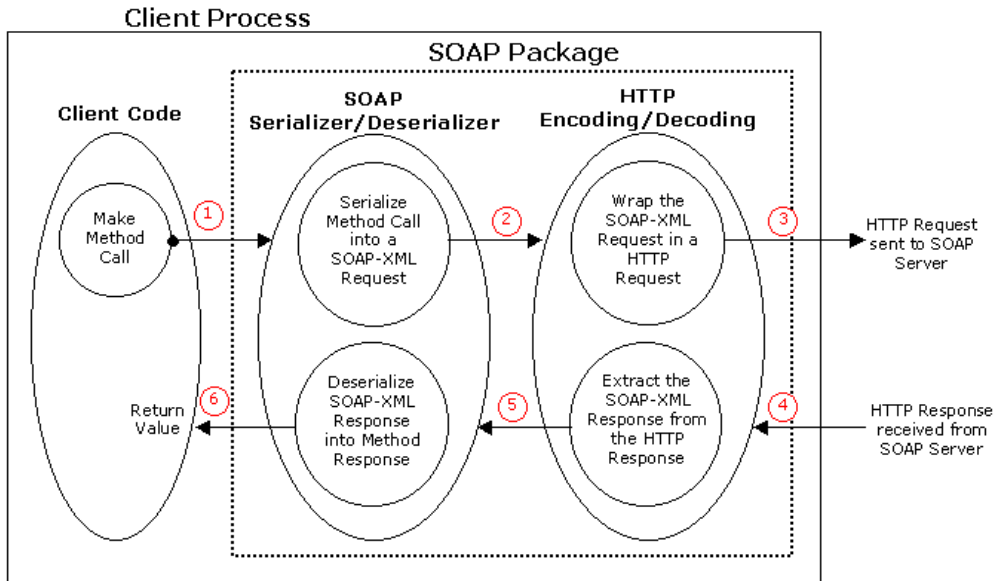
MISSING PROFILE DESCRIPTION

Commentaire [B4]: Décrire les 4 profils (ClientToolsComparison) et définir le but précis de ces profils



4 Présentation de quelques frameworks

Commentaire [B5]: Change client process to IDE/API SOAP Package →Framework



What client Side framework should be allowed...

Summary explanation of the diagram above:

The client code (written in various languages available like Java, C / C + +, Perl, Python, PHP ...) makes a service call by invoking the appropriate method in the SOAP package (1).

The SOAP package via SOAP serializer converts this invocation into a SOAP request and sends that to the HTTP encoder (2).

SOAP serialization converts (serializes) the public fields and properties of an object, or the parameters and return values of methods, into an XML stream that conforms to a specific XML Schema definition language (XSD) document. SOAP serialization results in strongly typed classes with public properties and fields that are converted to a serial format (in this case, XML) for storage or transport.

As XML is an open standard, the XML stream can be processed by any application, as needed, regardless of platform.

It is a simple powerful concept (Serialization & De-serialization), which allows an object to retain its form even across a network.

The HTTP encoder wraps the SOAP message in a HTTP request and sends it to the SOAP server (3). the SOAP server is simply an application server (appserver) running on the milegate.

The response is received from the SOAP server by the HTTP encoder/decoder(4) which decodes it and extracts the SOAP response which it hands to the SOAP deserializer (5).



The SOAP deserializer deserializes the message and gives the result to the client code (6) as the return value of the original invocation (1).

5 Etude frameworks

See ClientToolsComparison

5.1 Axis2

5.2 CXF

5.3 WSO2

5.4 METRO

5.5 WCF

6 Tests Tools

Coming soon

6.1 WS-I tools

6.2 Others tools

7 Conclusion

Commentaire [B6]: Outils de tests destinés à vérifier la conformité aux profils WS-I, OASIS... et non la syntaxe des spécifications W3C et format des messages comme l'a dit David...



Annexes

7.1 Revision history

Doc ID	Revision		Short description of the modification	Prepared by	Checked by	Approved
	Version	Date				
C.Tools	V1	15-06-09	Clients Tools (Draft)	T. KIKI		
C.Tools	V1.1	23-06-09	<ul style="list-style-type: none">• Remove server app. Description.• Diagram description correction.• Add Framework survey	T. KIKI		
C.Tools	V1.2	27-06-09	<ul style="list-style-type: none">• Framework survey correction• Framework Avaluation	T. KIKI		

7.2 References