

The SOA Gateway, for 'Green' Integration

**This document is intended to give IT personnel
an understanding of the how the SOA Gateway provides
a 'Green' alternative to traditional integration methodologies**

This document is distributed for information purposes only and does not form part of or constitute an agreement with Rissaris Ltd. Although Rissaris Ltd. uses reasonable efforts to include accurate and up-to-date information in this document, Rissaris makes no warranties or representations as to its accuracy. Rissaris Ltd. may also make improvements and/or changes to this document at any time without notice. The various approaches outlined in this document are put forward in good faith, but it remains possible that individual results may vary. For that reason and in accordance with standard practice, readers are encouraged to test any materials developed on the basis of this paper before putting them into productive use.

1. Overview

Organizations the world over are beginning to see the benefits of the 'Green' agenda whereby IT infrastructure and application decisions can have a significant impact on the environment. This is not just being driven by good intentions but also by sound financial principles. In many cases, organizations have expanded based on cost of equipment but never took the cost of running that equipment into account. In hard financial terms, it is possible to get significant financial savings from running less equipment or more energy efficient equipment without even taking into account the impact on the environment.

Organizations are now more aware than ever that adopting green policies for their IT decisions can save money as well as the environment. These costs are often measured in a 'carbon count' which currently is a virtual cost but many governments are moving towards the concept of 'carbon taxes' which will result in organizations being charged based on their carbon emissions. By adopting a green agenda, organizations can save money today in hard energy usage terms and such policies will also lead to savings down the road as governments begin to penalize companies that have not adopted a green agenda.

All companies have IT integration issues to some degree and this document will show how using the SOA Gateway can considerably simplify a number of integration scenarios while also adhering to the green agenda. So not only does an organizations get a quick and cost effective implementation, they can also immediately reduce their carbon count and have the potential to reduce this even further going forward.

2. Web Services and the Green Agenda

The use of standards bases web services (i.e. using SOAP or REST) is widely accepted to provide the flexibility required to move work loads based. In many instances, workloads cannot be moved as they are architecturally integrated in such a way that the must run on a specific hardware and/or software platform. When integration occurs through the use of a loose coupling standard like SOAP, or using the REST based approach which provides access to resources using URLs, it is possible to easily move those workloads to a greener, more energy efficient platform. As the Web Services will operate in an identical way on any software or hardware platform, the software using these services will operate unchanged.

The SOA Gateway exposes resources as SOAP and REST based web services and thus offers the ultimately flexibility to an organization as to where those services are run. It offers the potential to have a slow migration of these services one by one or application by application to avoid the pitfalls that can occur with a big bang approach. With this approach, the organization can choose the most appropriate platform on which to run these services thus offering the potential to move to greener technologies.

3. Using Web Services for Integration

It is clear from the previous section that SOAP and REST based Web Services provide a flexible integration approach which offers the potential to reduce energy consumption by

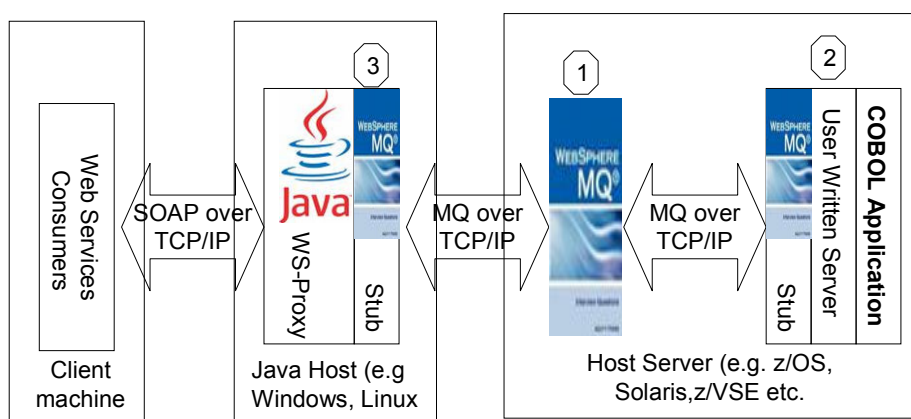
moving loads to more efficient platforms. This next section discusses how Web Services are being used today in integration projects and illustrates how Web Services can be created by the SOA Gateway and offer significant opportunities to save energy and reduce the overall cost of the traditional Web Services integration scenario.

Note that for the purposes of illustration, a COBOL application is used in this example, however, the same concept applies for integration with applications in other languages such as Natural or PL1, other environments, such as CICS or IMS/DC and other databases such as DB2, ADABAS, VSAM and others.

3.1.1. Traditional Web Services Integration

MQ Series can be configured in various ways, however, in order to provide the flexibility offered by the use of Web Services, the configuration required to make COBOL services available with SOAP is used here as an example.

A typical integration scenario looks like this:



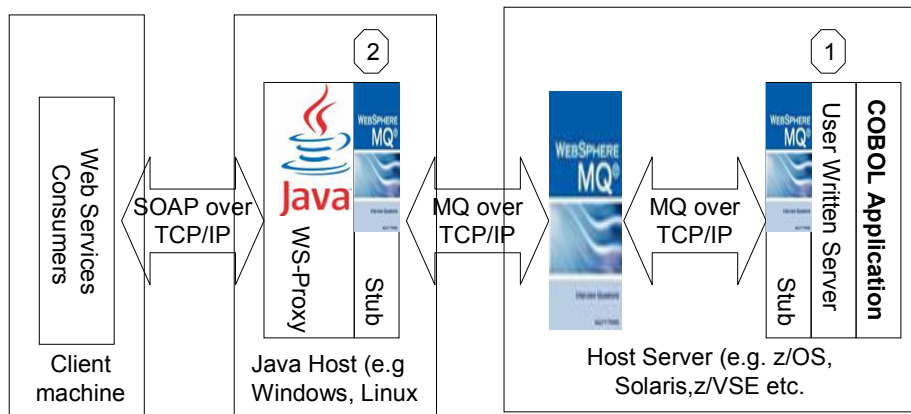
In order to achieve the above configuration, the following steps are required:

1. The MQ Server must be installed and configured on the host server where your COBOL business logic is running.
2. A user written server or application container must be built or installed to handle the MQ Series messages and to invoke the COBOL application.

3. The MQ Series stub must be installed on the Java host where you will be running your JAVA Web Services proxy for the service. This will generally run on a different machine to your host server but can potentially run on the same machine.

3.1.2. Creating a Web Service for your COBOL program

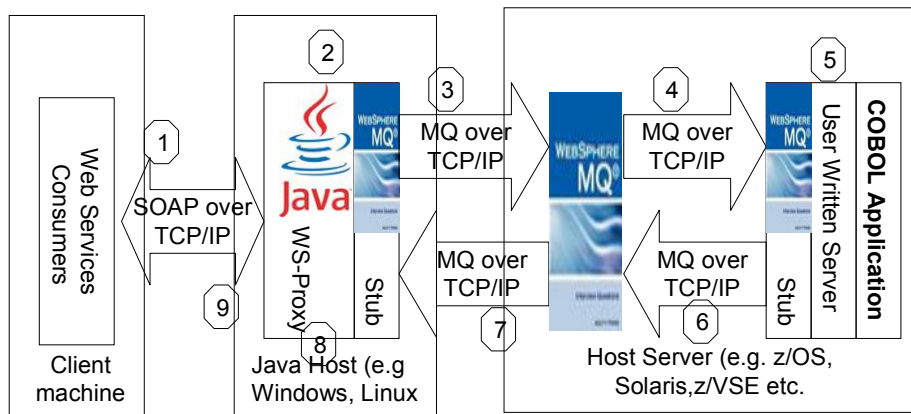
In order to make a Web Service available for a COBOL program with such a configuration, it is necessary to proceed as follows:



1. Custom coding must be added to the user written server to handle each COBOL program to be made available.
2. The user must create a JAVA Web Services proxy for the service and must then deploy this on the Java host.

3.1.3. Using the COBOL Web Service

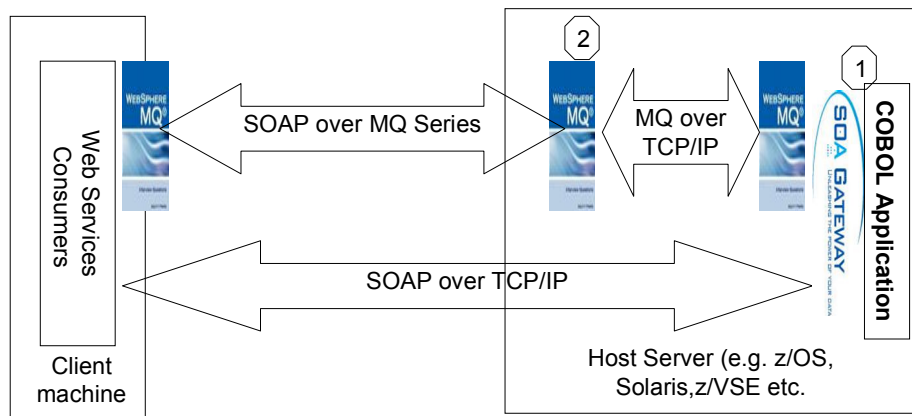
The Web Service may now be called from a Web Services consumer as follows:



1. The Web Services consumer will send the SOAP request to the Java proxy using TCP/IP.
2. The Java proxy will take the SOAP request and transform it to a MQ Series request.
3. The Java proxy will send the MQ request to the input queue defined in the MQ Series server via TCP/IP.
4. The MQ Series server will send the MQ request via TCP/IP to the User Written Server application.
5. The User Written Server will interpret the MQ request and call the COBOL program and get the result. It will then map the result back into a MQ response.
6. The User Written Server will send the MQ response back to the output queue defined on the MQ Series Server.
7. The MQ Series Server will send the MQ response back to the Java proxy.
8. The Java proxy will take the MQ response and transform it to a SOAP response.
9. The Java proxy will send the SOAP response back to the Web Services Consumer.

3.2. SOA Gateway Architecture

The SOA Gateway equivalent configuration to the above is as follows:



3.2.1. Installation and Configuration Requirements

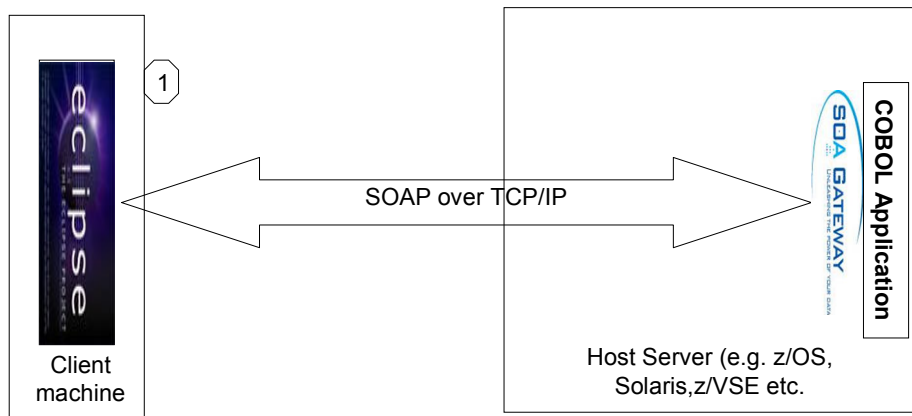
In order to achieve the above configuration, the following steps are required:

1. The SOA Gateway Server is installed on the Host server.
2. Where MQ is to be used as the transport, the MQ Server must be installed and configured on the host server where the SOA Gateway is running.

Contrast the above with the installation and configuration steps required with the traditional solution. The task of installation involves less steps, less hardware and less software thus reducing the effort to get the solution installed. This is effort that is saved during initial installation and configuration and periodically as maintenance is required to keep the technology up to date.

3.2.2. Creating a Web Service for COBOL

In order to make a Web Service available for a COBOL program with such a configuration, it is necessary to proceed as follows:

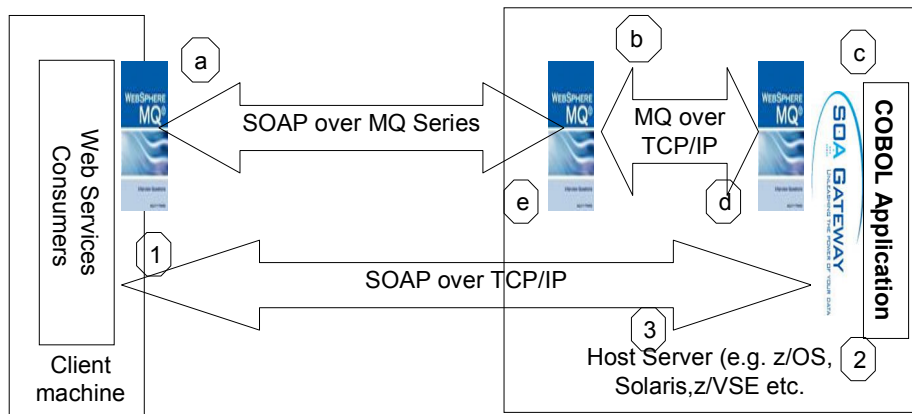


1. The COBOL program is defined as a service to the SOA Gateway using the Eclipse based Control Centre

Creating Web Services from the legacy artifacts is a point and click exercise and thus takes minutes instead of hours, days or even weeks to set up and configure the web service for the traditional scenario. In this way, effort can be saved each and every time integration of data or an application is required.

3.2.3. Using the COBOL Web Service

The Web Service may now be called from a Web Services consumer as follows:



When using the TCP/IP transport, usage occurs as follows:

1. The Web Services consumer will send the SOAP request to the SOA Gateway Server using TCP/IP.
2. The SOA Gateway Server will interpret the SOAP request, call the COBOL program and get the result. It will then map the result back into a SOAP response.
3. The SOA Gateway Server will send the SOAP response back to the Web Services Consumer.

When using the MQ transport, usage occurs as follows:

- a) The Web Services consumer will send the SOAP request to the MQ series Server as payload using the MQ transport protocol.
- b) The MQ Series server will send the SOAP request payload to the SOA Gateway Server.
- c) The SOA Gateway Server will interpret the SOAP request, call the COBOL program and get the result. It will then map the result back into a SOAP response.
- d) The SOA Gateway Server will send the SOAP response payload back to the output queue using the MQ transport protocol.

- e) The SOA Gateway Server will send the SOAP response back to the Web Services Consumer using MQ.

The above logic illustrates how less effort, hardware and software is required **for each request** thus reducing the carbon cost of the day to day use of the integrated data or business logic.

4. Summary

This document shows that using the Web Services and the SOA Gateway can make a real difference to your carbon footprint when integrating existing data or business logic. The green benefits that can be achieved by using this methodology are as follows:

- No additional hardware and software is required to complete the integration effort.
- A simpler initial installation and configuration means less effort is required to get the software installed and configured.
- Simple, single step creation of Web Services saves time and effort over the traditional approach.
- No installation and maintenance of software is required on the client systems thus saving effort on an ongoing basis.
- Savings are made with each invocation of the service due to the two tiered approach to the solution avoiding energy consuming hardware and software required by 3 tiered integration solutions.