

# Web Services Container Reference Architecture (WSCRA)

Arulazi Dhesiaseelan, Venkatavaradan Rangunathan  
Hewlett-Packard Company  
[arulazi.d@hp.com](mailto:arulazi.d@hp.com), [venkat.rangunathan@hp.com](mailto:venkat.rangunathan@hp.com)

## Abstract

*Today, web services are pervasive and omnipresent in the Internet and within enterprises. Even though there are massive web services specifications development underway, early adoption by developers and tool vendors is becoming a need. The potential growth of this technology is highly predictable because of its universal acceptance and use among the developer community. This industry may expect to grow enormously based on the support from various communities that benefit from this technology. Researches are carried out in various standards bodies on various aspects of web services such as definition, architecture, security, discovery, interoperability, etc. As we are committed to the success of this technology, we need to research on service oriented containers that makes web services potential more constructive. This paper proposes a container for web services which can manage and monitor the state and behavior of web services, which may address the Quality of Service (QoS) factor for web services.*

## 1. Introduction

Web services deployment in huge enterprises is becoming vogue. Most of the enterprises provide wrapper to their legacy systems for better usability and increased utilizations among the collaborating entities. Due to its standard way of interacting with providers and consumers, the use of this technology could be varied. This technology may provide a unique solution for businesses to collaborate seamlessly with the existing infrastructure. The use of web services are becoming a reality in today's enterprises. There may be some drawbacks in its premature adoption. But these can be overcome easily as the technology heavily relies on HTTP, the underlying protocol used by the World Wide Web, XML for data representation. Unlike traditional computing systems, web services

share business logic, data through programmatic interfaces across a network. The key to success of this technology is to provide a dedicated container architecture which may address various factors associated with the state and behavior of web services. The requirement for such container is the central theme of this paper.

## 2. Web Services Containers

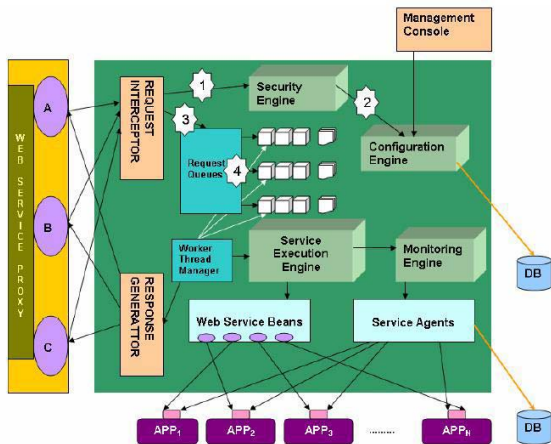
Today's container architectures provide only a basic subset of components such as logging, security, and configuration and implement the component interactions in their own proprietary ways. J2EE/.NET containers for web services are good enough for early adoption. Use of best of breed components for enterprise application integration is becoming a nightmare among the technology adopters due to its tight coupling between the container and the services they offer. Some of the challenges that are not addressed by the today's containers are:

- J2EE/.NET containers are language/platform dependent.
- Unable to address broken SLAs.
- No well-defined standards for container-component interaction.
- No well-defined standards for hosting web services.
- Service Oriented Architecture (SOA) exists only in concept, not in reality.

## 3. WSCRA Proposal

Web Services Container Reference Architecture (WSCRA) proposal outlines the need for managing and monitoring the state and behavior of Web services which are hosted within a platform and language neutral, logical container. This reference architecture also allows addressing various QoS factors that determine the efficiency factor of a web service. Some of the QoS factors that influence the rapid adoption of

this technology include: performance, scalability, reliability, availability, security, and, interoperability.



Web Service Virtual Machine (WSVM) is the runtime environment for languages such as Perl, Python, Java, C#, etc. Each of these language bindings are modeled as WSVMs, that can run as independent containers controlled by a logical container which maintains references to the deployed web services. Tools can monitor this logical container for the state and behavior of the deployed web services which can span across different machines, and across networks. Web Service Beans (WSB) represent instance of these web services which can be pooled for maximum availability and accessibility. The configuration in the logical container helps to locate the instance of WSB and their corresponding bindings associated with QoS parameters. This container approach allows multiple VMs to interoperate transparently irrespective of programming languages and platforms. For instance, services could be deployed either in WebLogic container or Python container which can be managed and monitored in an uniform manner, thus abstracting the backend for these services represented as WSB.

The proposed conceptual architecture addresses the following factors that influence the architecture definition.

- To monitor the availability of the web services across multiple physical containers.
- To prioritize the access to the web services.
- To load balance the web services for scalability.
- To monitor and control the performance of web services.
- To provide secure access to the web services.
- To monitor the SLAs for web services.

- To allow adopters to choose their preferred partners.
- To allow service pooling/filtering.
- To standardize the container-component interactions.
- To enable service oriented computing.

#### 4. Competitive Approaches

As of today, no similar approaches are available. But this problem would arise as Web services adoption is becoming more ubiquitous, and the need for such a container would be highly unavoidable.

#### 5. Current Status

Currently, we are involved in refining the architecture to provide a better picture for the web services community.

#### 6. Future Work

- To draw support for this initiative from web services community and industry leaders.
- To define standards for Container-Security Service component interaction, Container-Load Balancer component interaction, Container-Configuration component interaction etc.
- To prototype a cross-platform service container that may manage and monitor XML web services (J2EE, .NET, Python, PERL).

#### 7. References

OASIS Web Services Distributed Management  
[http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wsdm](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsdm)

Web Services and Service Oriented Architectures  
<http://www.service-architecture.com/>

W3C Web Services Architecture Working Draft  
<http://www.w3.org/TR/2003/WD-ws-arch-20030808/>

J2EE vs. Microsoft.NET: A comparison of building XML-based web services  
<http://www.theserverside.com/articles/pdf/J2EE-vs-DotNET.pdf>