



| IBM Software Group

IBM WebSphere Application Server V6.1

Web Services – Resource Framework



@business on demand.

© 2006 IBM Corporation
Converted to video May 15, 2015

This presentation will explain the Web Services Resource Framework support in WebSphere Application Server V6.1.

Agenda

- Example
- Architecture
- Quality of Service
- Summary and Reference



This presentation will begin with an example of why the Web Services Resource Framework specification is needed. It will then discuss the architecture and workings of the specification and the benefits it has for quality of service.

Section

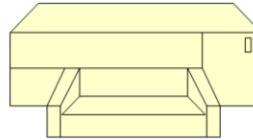
Example



Next is an example of the type of problem that the Web Services Resource Framework attempts to solve.

Example

- You want to expose a printer as a resource that can be managed using Web Services



- You can write a Web Service to represent the printer and publish a WSDL

```
<wsdl:portType name="printer">  
  <wsdl:operation name="GetPrinterProperties".../>  
  <wsdl:operation name="SetPrinterProperties".../>  
  ...  
</wsdl:portType>
```

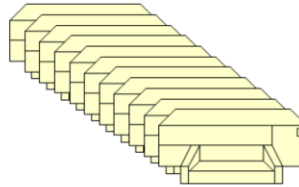
- A simple SOAP message exchange can be used to get or set the printer "state"

```
<soap:body>  
  <mysvc:GetPrinterPropertiesRequest/>  
</soap:body>
```

In order to explain the Web Services Resource Framework specification it is best to consider an example of using Web Services to manage stateful resources. For instance, imagine wanting to manage a printer resource using Web Services. A Web Service could be written that would represent the printer, and this could publish and make available a WSDL. Using this implementation, an exchange of SOAP messages could then be used to check and set the state of the printer.

Example continued

- But you have many printers in the department



- You need to be able to describe which printer resource you care about

- You can make the WSDL a little more complicated

```
<element name="GetPrinterPropertiesRequest">
  <complexType>
    <sequence>
      <element name="PrinterName" type="string"/>
    </sequence>
  </complexType>
</element>
```

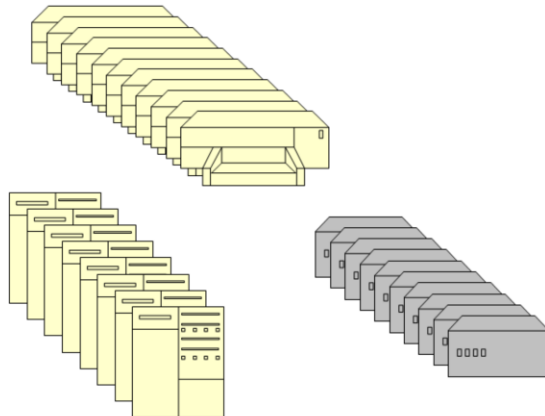
- A slightly more complex SOAP message contains the identity of the target printer resource

```
<soap:body>
  <mysvc:GetPrinterPropertiesRequest>
    <mysvc:PrinterName>
      Printer13
    </mysvc:PrinterName>
  </mysvc:GetPrinterPropertiesRequest>
</soap:body>
```

Now, expanding on the example, what if the environment contained numerous printers? The Web Service could be written with a parameter to specify a specific printer resource, this would in turn make the WSDL document slightly more complicated as is shown on the slide. The SOAP messages that are used would now need to contain a parameter for the identity of the printer that is being managed. This could be used to create a fairly robust Web Service for managing printer resources.

Example continued

- You also want to manage your servers, the applications on the servers, your storage devices
- Each resource type has its own interface and strategy for exposing identity and state
- How does a management agent deal with these different resource types



```
<soap:body>
  <abc:GetServerPropertiesRequest>
    <abc:clusterid>cl</abc:clusterid>
    <abc:serverid>s1</abc:serverid>
  </abc:GetServerPropertiesRequest>
</soap:body>
```

```
<soap:body>
  <mysvc:GetPrinterPropertiesRequest>
    <mysvc:PrinterName>
      Printer13
    </mysvc:PrinterName>
  </mysvc:GetPrinterPropertiesRequest>
</soap:body>
```

Now, to expanding upon the example a bit more, what if the Web Service also needed to manage other types of resources in the environment? Each other type of resource would have a different interface and methods for managing it, and the Web Service would need to be able to manage these different types of resources in a meaningful way. This adds significant complexity to the problem, and it is this type of problem that the Web Services Resource Framework specification tries to solve.

Section

Architecture



The next section will explain the architecture and components of the Web Services Resource Framework implementation.

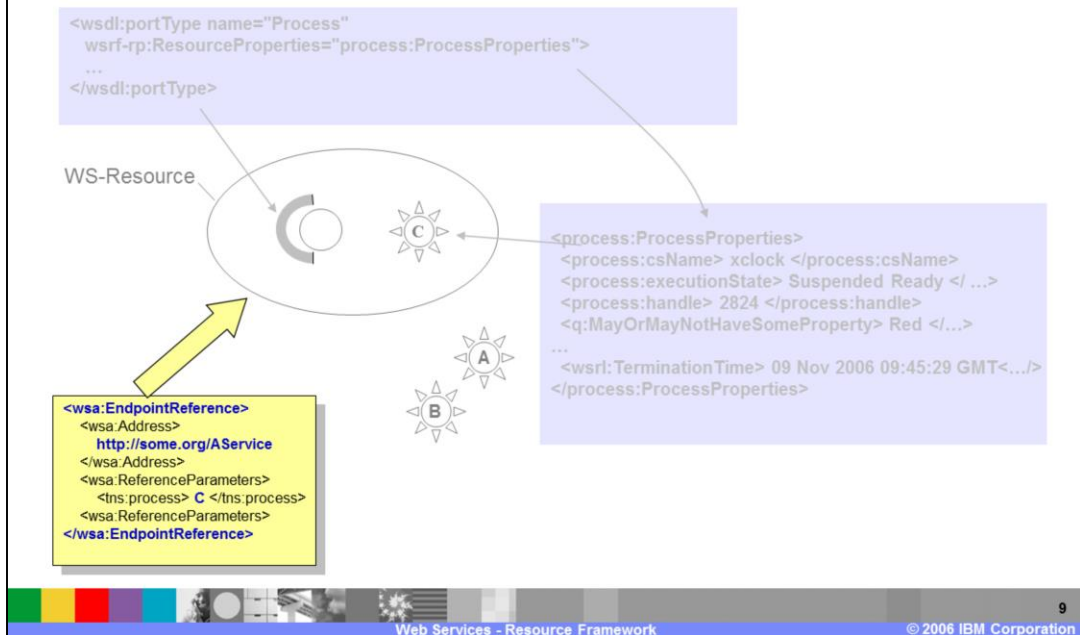
Motivation for WSRF

- Stateful entities exist in most systems
 - ▶ Data in a purchase order
 - ▶ Configuration of devices in a network
- Need a standard way to deal with state in Web Services context
 - ▶ Each system does it in a “unique way”
 - ▶ Complicates integration
- Goal:
 - ▶ Formalize a mechanism to represent “state” in Web Services



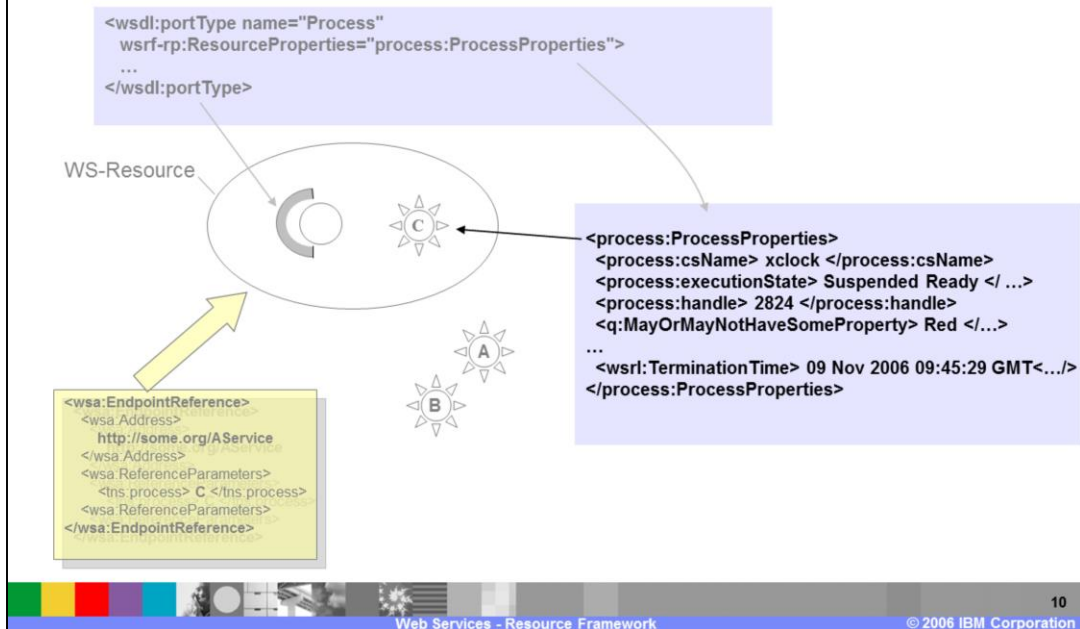
This concept of stateful entities or resources exist in most environments like a configuration for a printer or data in a purchase order. Web Services need a standard way to deal with state within the context of a Web Service application. The goal for the Web Services Resource Framework is to provide a standard way to represent state within a Web Service, thus simplifying the integration and management of resources. WebSphere Application Server Version 6.1 provides support for the WSRF specification, through a Java™ API for WS-Addressing end point references. The support is integrated within WebSphere Application Server's workload management and high availability components, allowing for the creation of highly available, scalable Web Services Resource applications.

Components of WSRF



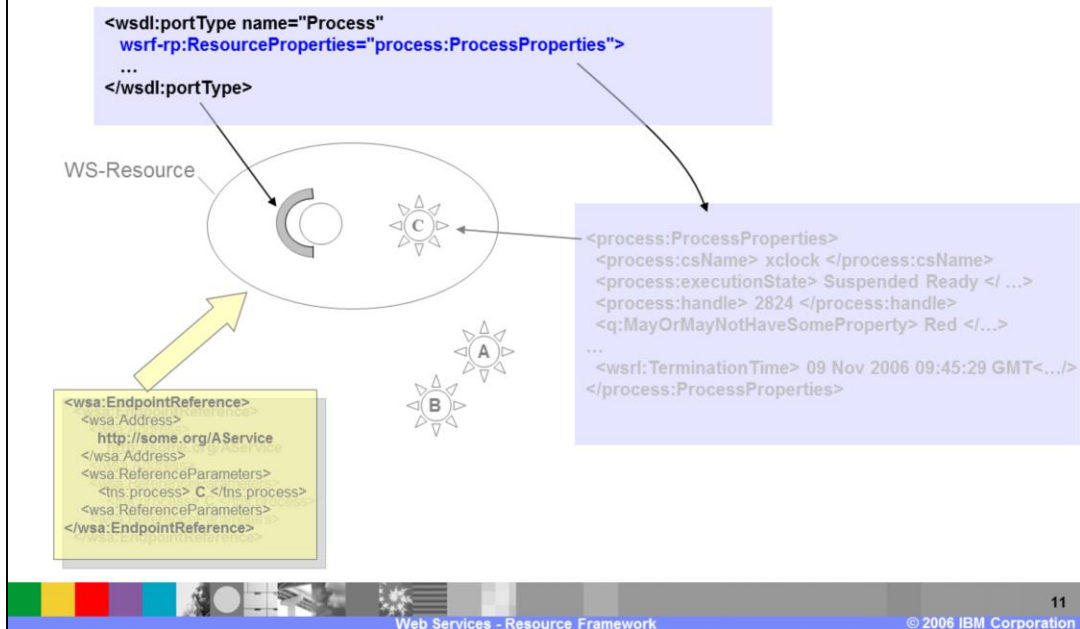
This example shows three resources; A, B and C. These could be any types of resources, such as printers from the example used earlier. There is one Web Services resource, C, which is a combination of the resource and the Web Service through which it is exposed. This resource is referenced through an end point reference based on the WS-Addressing specification. This end point reference contains an address or URI for the resource, it also contains zero or more reference parameters, which in this case is the actual resource C. This end point reference is created and made available by the service for the resource.

Components of WSRF



The end point reference also contains parameters that only the resource can understand. A number of processing properties are defined by the specification that handle how the properties for the resource are mapped to XML. The calling service does not need to be aware of the specifics of managing the resource, this can be managed by the runtime that handles the SOAP messages that are received. This resource properties document describes the state of the resource.

Components of WSRF



The specification also dictates how the processing properties are exposed within the WSDL document for the resources Web Service. So a WS resource will have a special WSDL attribute call for the resource properties, this will provide the WS consumer with information about the state of the WS resource.

Section

Quality of service



The next section will explain the quality of service options based on the Web Services Resource Framework specification.

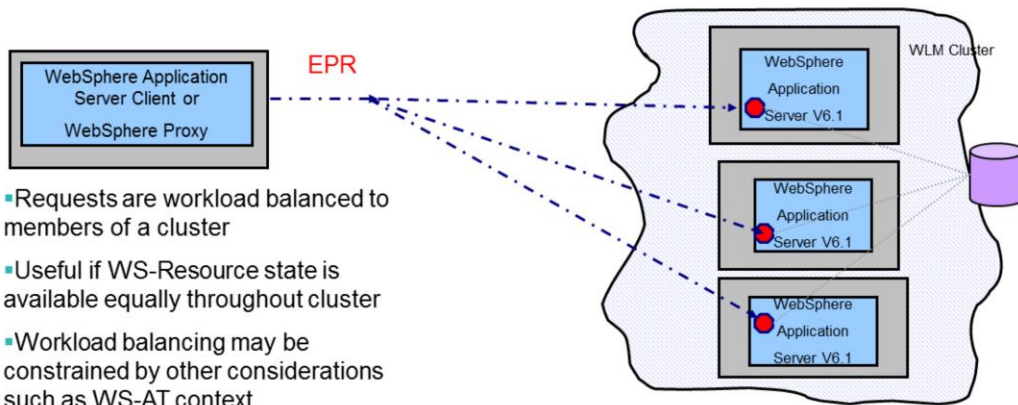
EPR: Qualities of service

- WebSphere Application Server supports three qualities of service for End Point References (EPRs):
 - ▶ Workload-balanced EPRs (default)
 - Workload balanced across cluster
 - Useful for scenarios that have no volatile state cached
 - `createEndpointReference(QName serviceName, String port)`
 - ▶ Fragile EPRs
 - EPRs created to non-load-balanceable endpoint – that is, with server affinity
 - Useful for scenarios that employ volatile state or caches
 - `createEndpointReference(QName serviceName, String port, null)`
 - ▶ Highly available (fragile) EPRs
 - Endpoints have server affinity, but affinity can be failed-over
 - EPRs created with singleton cluster-id, routed by WLM selection service in the client or WebSphere proxy
 - Useful for scenarios that favor affinity but can be failed over
 - SFSB implementation of WS-Resource using DRS to replicate state
 - `createEndpointReference(QName serviceName, String port, SessionBean sfsb)`



WebSphere Application Server supports three levels for the quality of service for an end point reference. The first option is for a workload balanced end point reference, where the workload can be balanced across a cluster. This is appropriate when there is no volatile state that may be cached within a particular server. Another option is for fragile end point references. These end point references can maintain affinity with a server. The last option is for end point references that are fragile, but must also be made highly available. This option allows the server affinity for the end point reference to fail over to another server in the cluster.

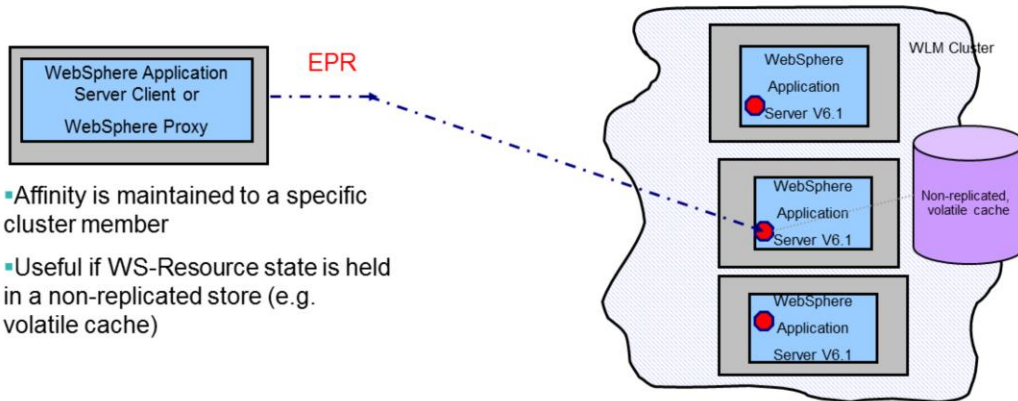
Workload balanced EPR



- Requests are workload balanced to members of a cluster
- Useful if WS-Resource state is available equally throughout cluster
- Workload balancing may be constrained by other considerations such as WS-AT context

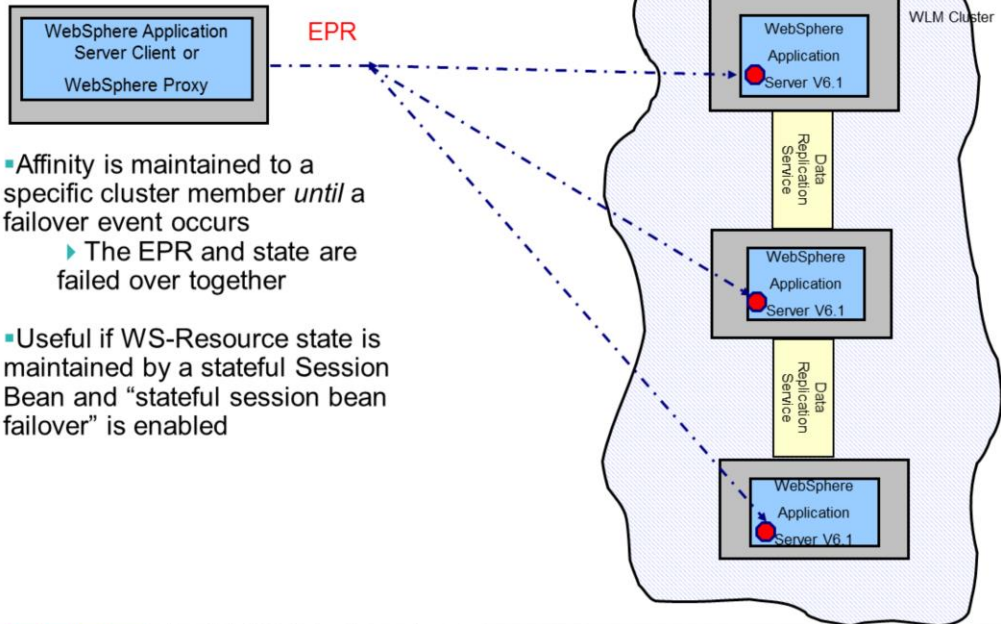
The workload balanced end point reference is the default, and also preferred behavior. Incoming requests will be balanced across members in the cluster. In this case the state of the WS resource must be available equally throughout the cluster. The balancing may be limited by other considerations such as being part of an atomic transaction.

Fragile EPR



For a fragile end point reference, affinity will be maintained to a specific application server. The WS resource state in this case is most likely stored within some type of volatile cache local to each application server in the cluster. Each application server will only be aware of specific state information.

Highly available (fragile) EPR



- Affinity is maintained to a specific cluster member *until* a failover event occurs
 - ▶ The EPR and state are failed over together
- Useful if WS-Resource state is maintained by a stateful Session Bean and “stateful session bean failover” is enabled

The most common example of a highly available fragile end point reference uses a stateful session bean to maintain the state information. The information stored within the stateful session bean is maintained across the cluster by the data replication service. Stateful session bean failover occurs on passivation boundaries of a stateful session bean; that is, a failed-over stateful session bean is activated in the same state it was in at the last successful passivation. By default, stateful session bean passivation occurs on transaction boundaries.

Section

Summary and reference

The next section is a summary of the presentation.

Summary and references

- WSRF:
 - ▶ Defines a standardized means to represent state in Web Services, including its lifecycle, using WS-Addressing EPRs
 - ▶ Defines a standard set of application-level message to operate on the state



The Web Services Resource Framework is a standard way for representing state in Web Services. It also defines a set of application level messages that can be used to operate on the state of resources.

Industry Status of WSRF and WS-Addressing

- WSRF OASIS TC
 - ▶ formed April 2004
 - ▶ http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf
 - ▶ Committee specs approved Jan 2006
 - ▶ Complete OASIS standard expected March 2006



- WS-Addressing WG
 - ▶ formed Aug 2004
 - ▶ <http://www.w3.org/2002/ws/addr/>
 - ▶ Candidate Recommendation (CR) approved Aug 2005
 - ▶ Proposed Recommendation (PR) expected June 2006



This slide has additional information related to the WS-RF specification.

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	CICS	IMS	MQSeries	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
eflago/business	DB2	iSeries	OS/400	xSeries
ALX	DB2 Universal Database	Lotus	pSeries	zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2006. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.

