IBM WebSphere

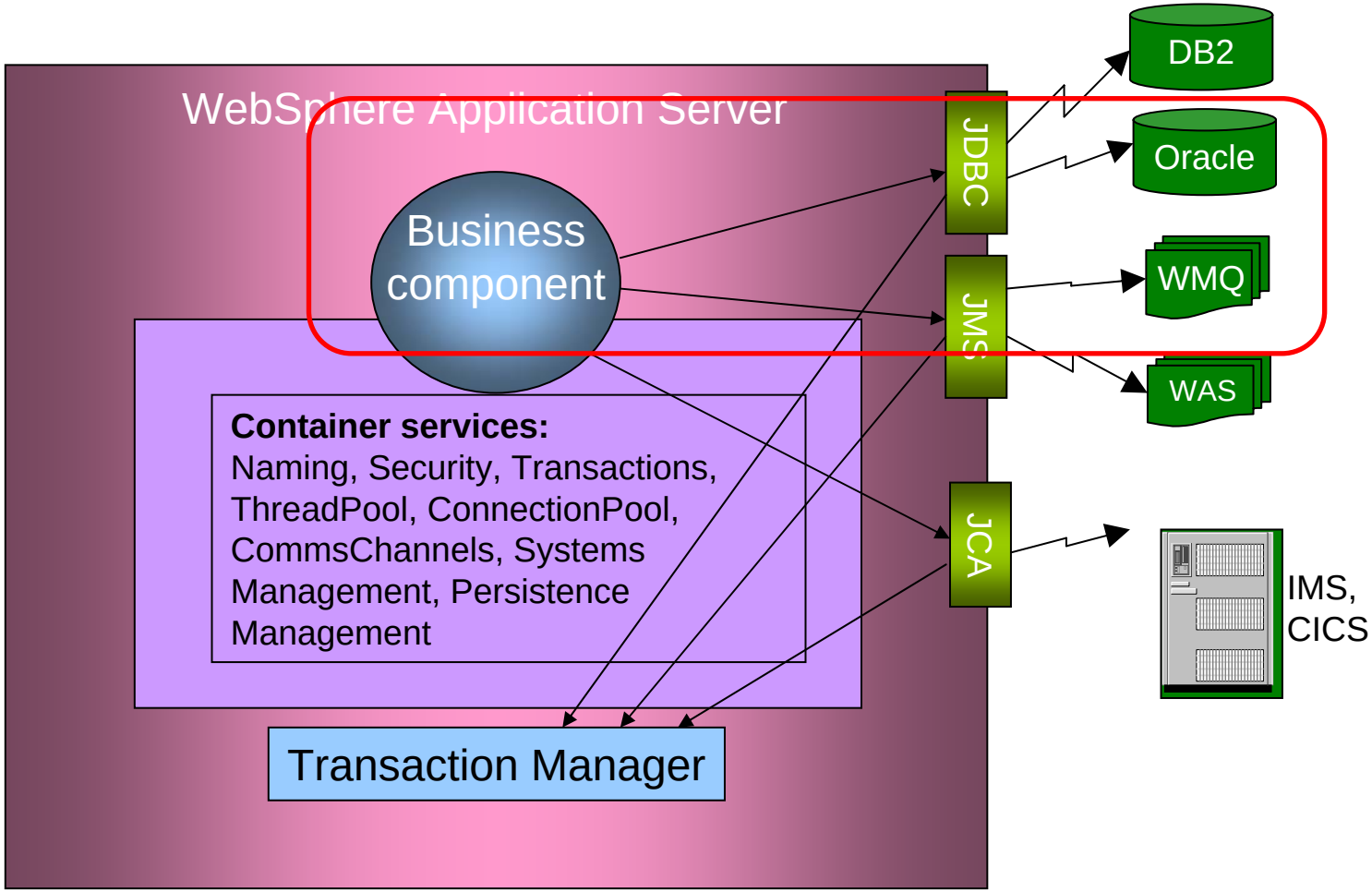
Real World Uses of Transactions in SOA

Ian Robinson, IBM Distinguished Engineer
WebSphere Transactions Architect, Chair OASIS WS-Tx TC

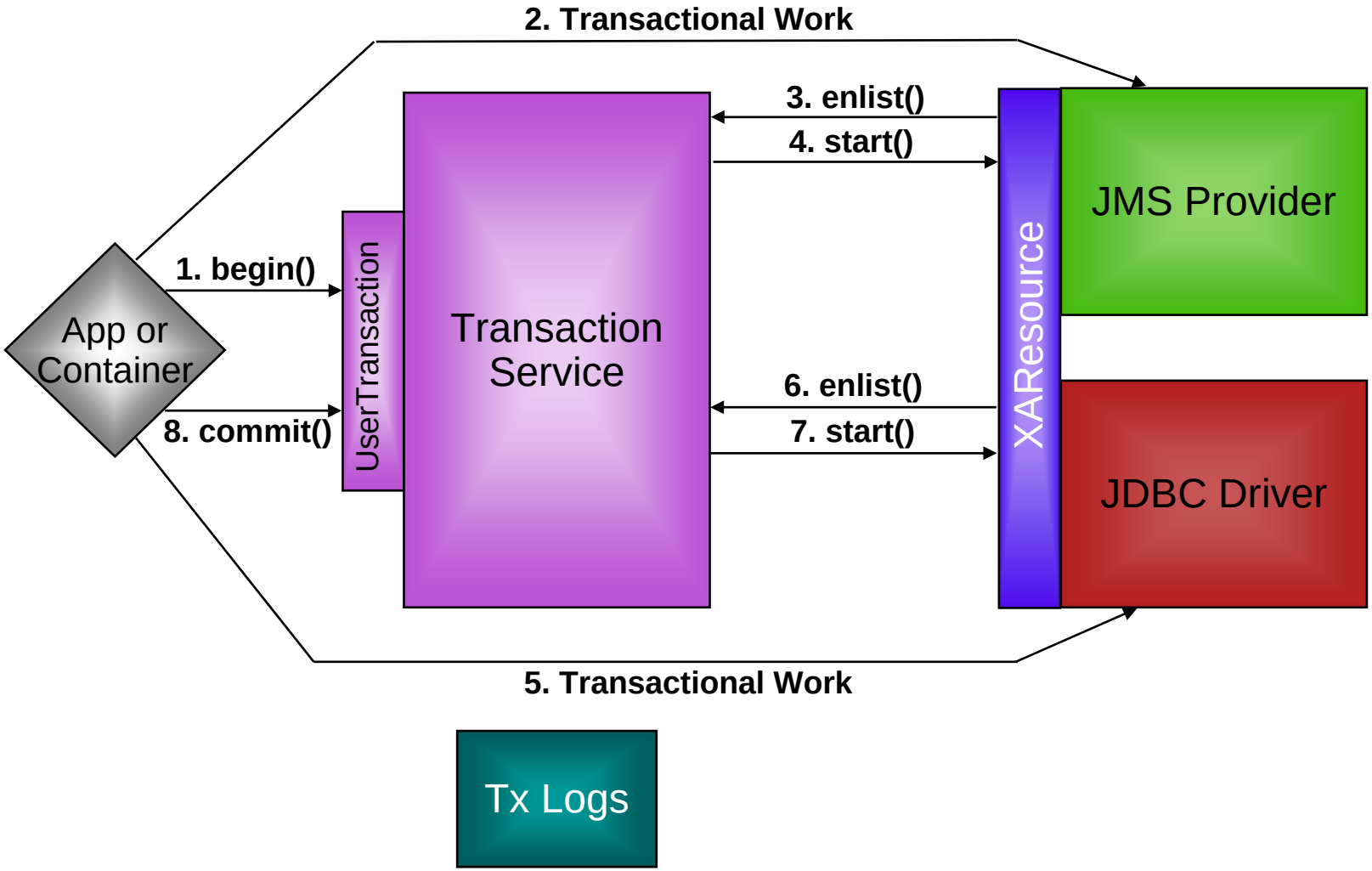
Agenda

- Transactions in Java EE and SOA
  - What is WS-Tx and how does it relate to Java EE?
- WAS support for WS-AT and WS-BA
  - How and when to use these in WAS.
- Qualities of service
  - What about the “NFRs” the specs don’t mention?
- Real world examples
  - Some real examples currently in use

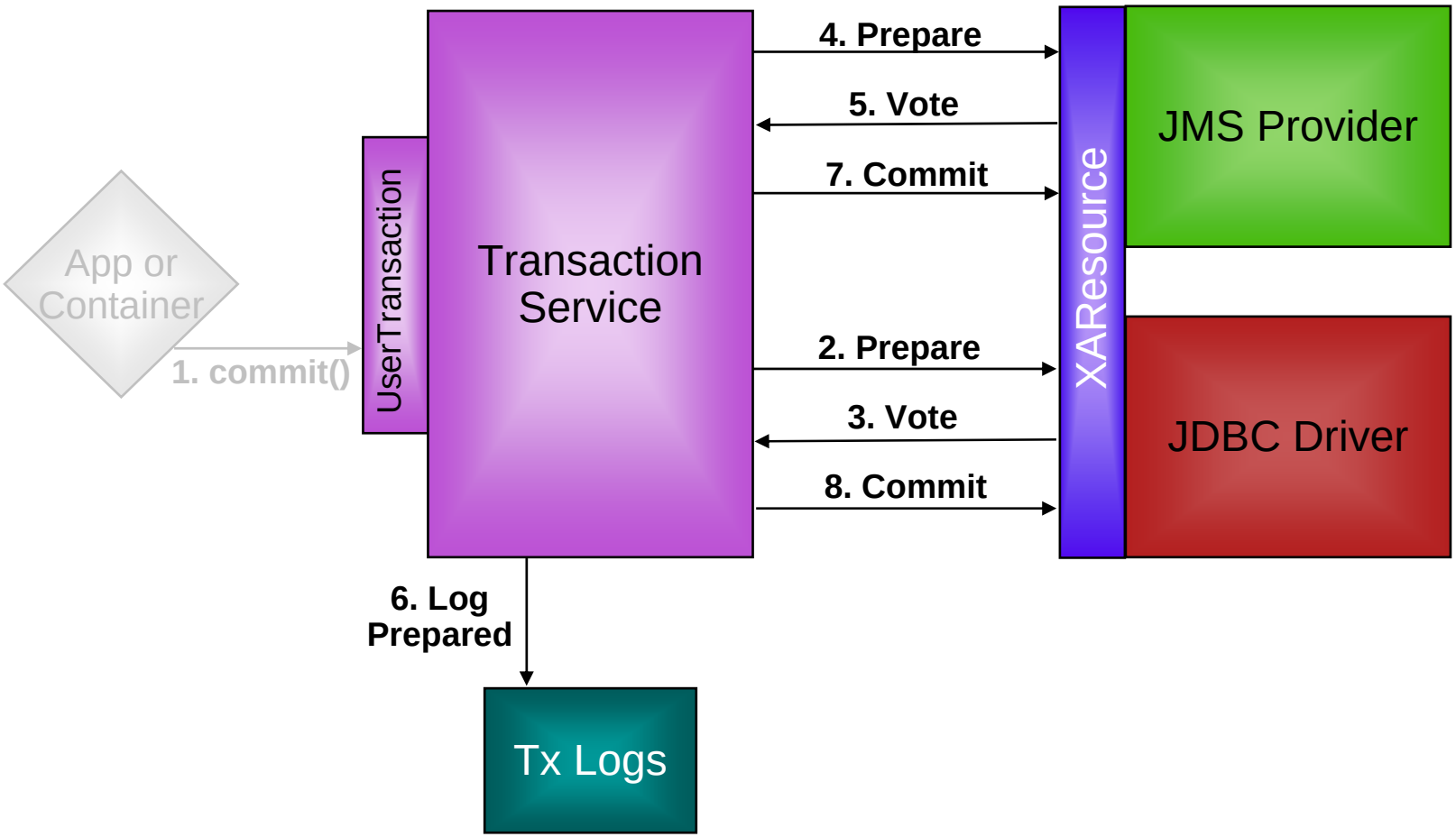
Local and Global transactions



Separation of concerns – interactions with RMs



Separation of concerns – 2-phase commit (2PC)

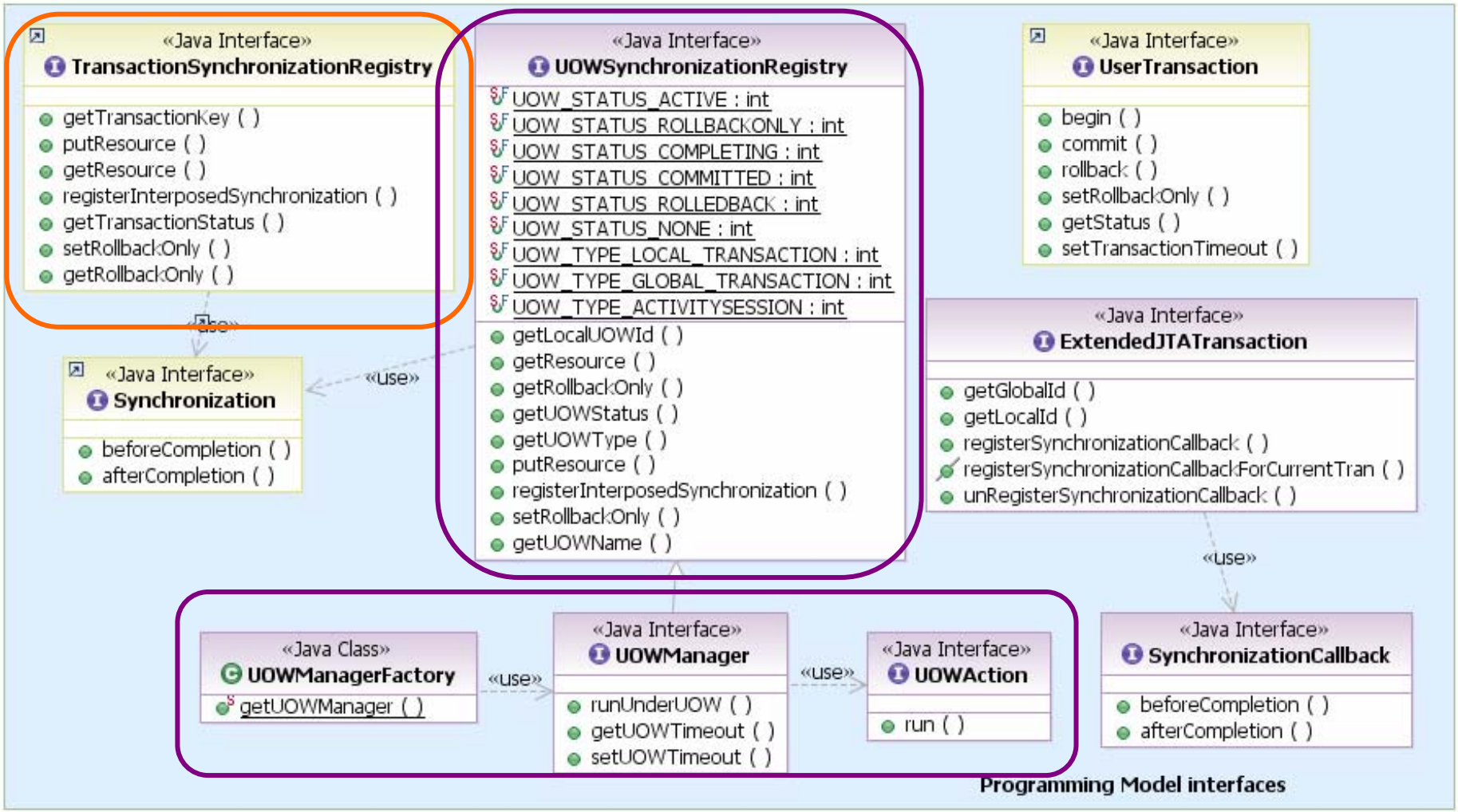


Bean or Container Managed Transactions?

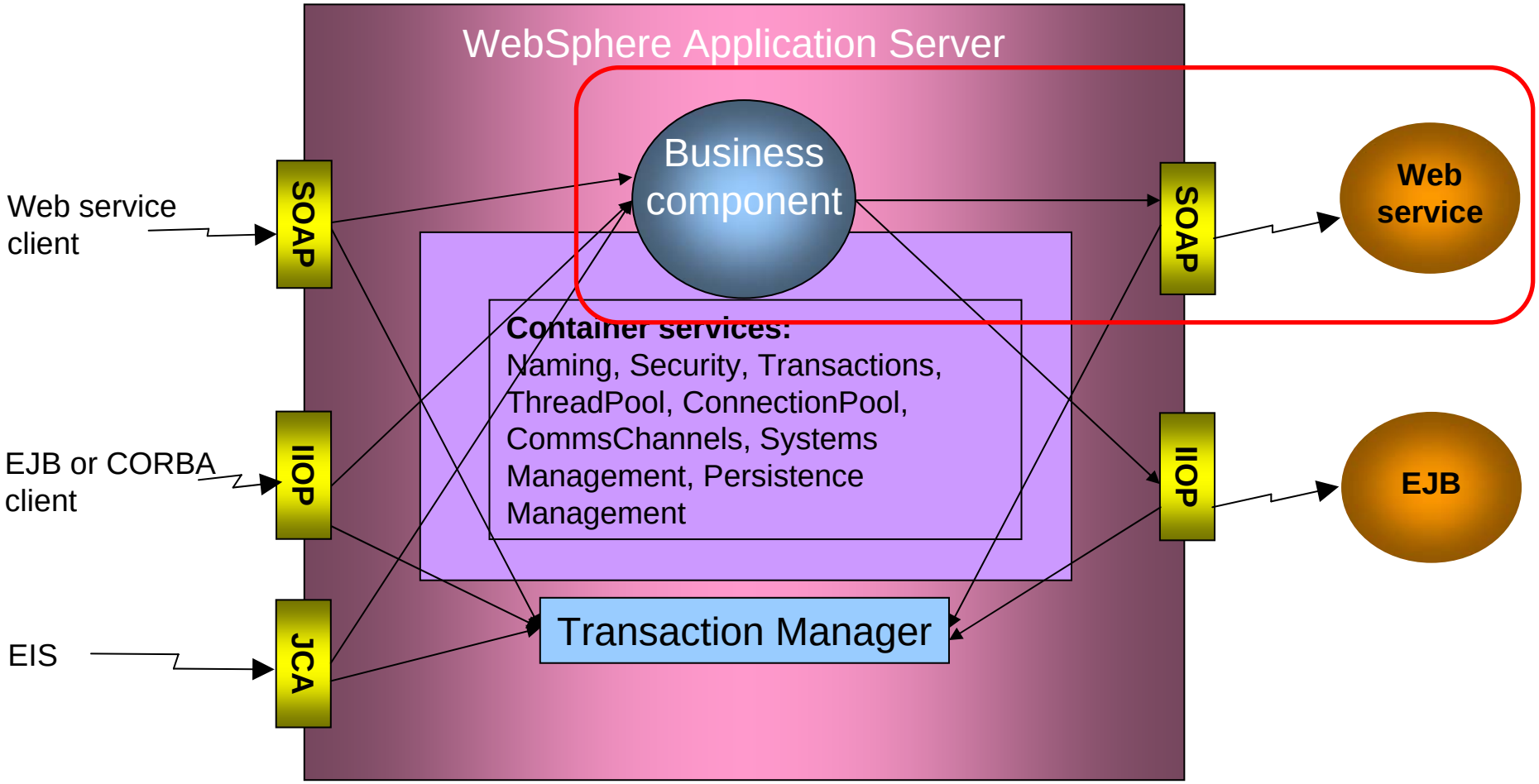
- Container managed (CMT)
  - Container manages transaction demarcation
  - Specified via application assembly tooling
    - Stored in deployment descriptors - No application code is required
  - Generally considered to be best practice
    - Easier to implement

- Bean managed (BMT)
  - Application code is responsible for demarcation of transactions
    - Via JTA UserTransaction API
      - begin(), commit(), rollback()
  - Complete set of APIs follow…

Java Transaction APIs (for Bean Managed Tx)



Distributed transactions



WS-Transaction: federating transactions across disparate systems

- WS-Tx defines the following concepts:
  - **An XML CoordinationContext** that identifies a transaction and which is passed implicitly in Web service messages without this context having to be declared as an explicit message parameter.
    - in the SOAP header for a SOAP binding
  - A generic **Coordination message set**
  - Protocol-specific messages sets that define the **AT and BA protocols**
- This XML context and XML messages are designed for simple transformation within different runtimes to map down to underlying transaction processing technologies.
  - For example, the J2EE WebSphere platform transforms WS-AT ← (JTA-based impl) → XA

Brief history of the WS-Transaction standard

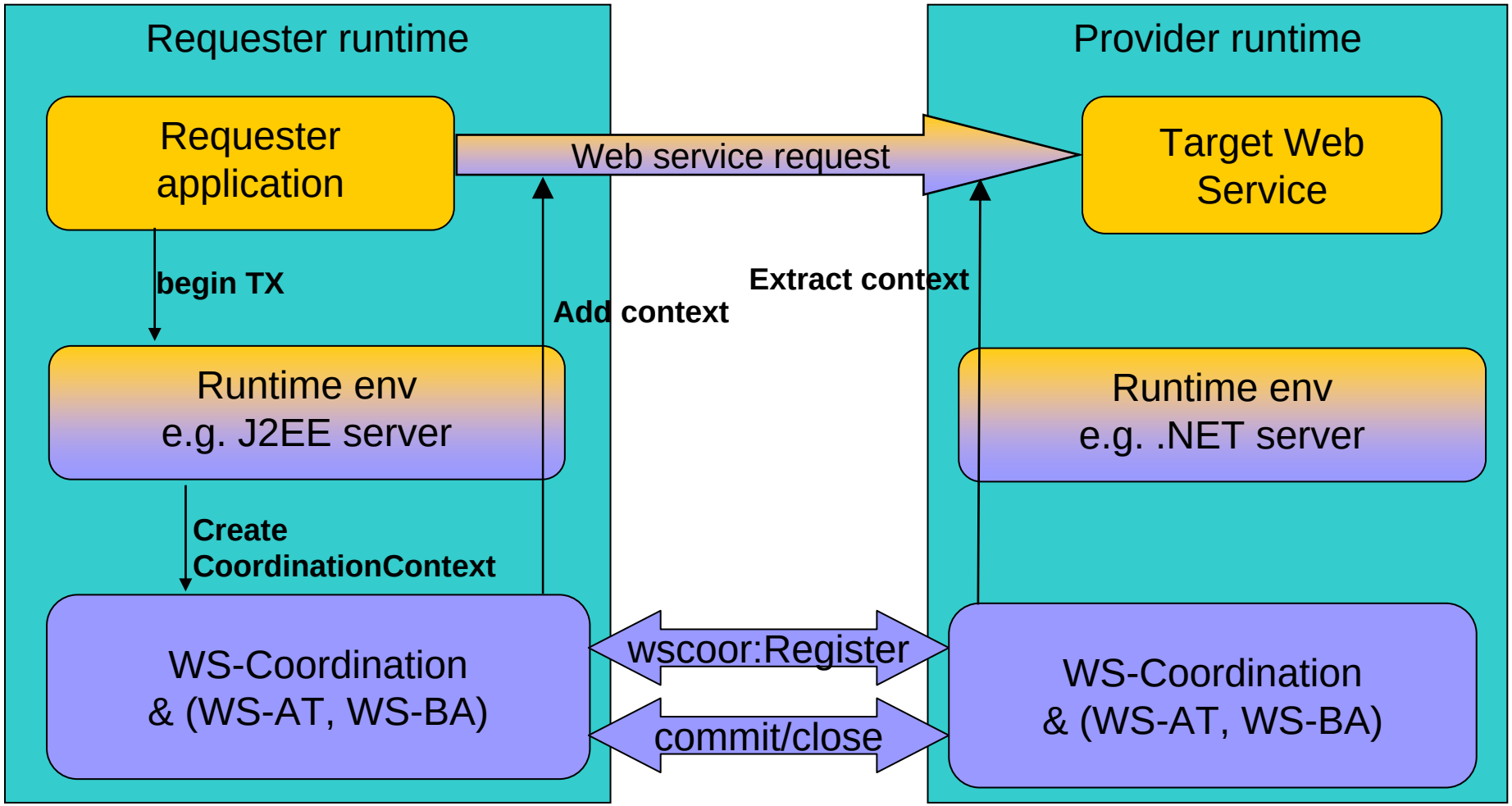
- Motivation: To provide a means to federate a variety of transaction models across different, existing systems
- WS-Tx = **WS-Coordination** +…
  - **WS-AtomicTransaction**
    - Atomic commit or rollback; 2PC
  - **WS-BusinessActivity**
    - Overall outcome atomic; business compensation
- IBM, Microsoft, BEA – first published draft Aug 2002.
- “Version 1.0” specs published Aug 2005
  - Input for OASIS WS-Tx Technical Committee
- OASIS WS-Tx TC V1.1 standard published April 2007

Status of WS-Transaction in IBM Products

- WS-C, WS-AT 1.0 available since:
  - WAS v6.0 (Dec 2004) and later
  - CICS TS v3.1 (March 2005)
- All interoperate with MS WCF (.NET v3) WS-AT support using SOAP/HTTPS.
- WS-BA 1.0 available since:
  - WAS v6.1 (May 2006)

- OASIS Standards (V1.1) of all the above in WAS V7 (additive).
- WS-Policy support for WS-Transaction

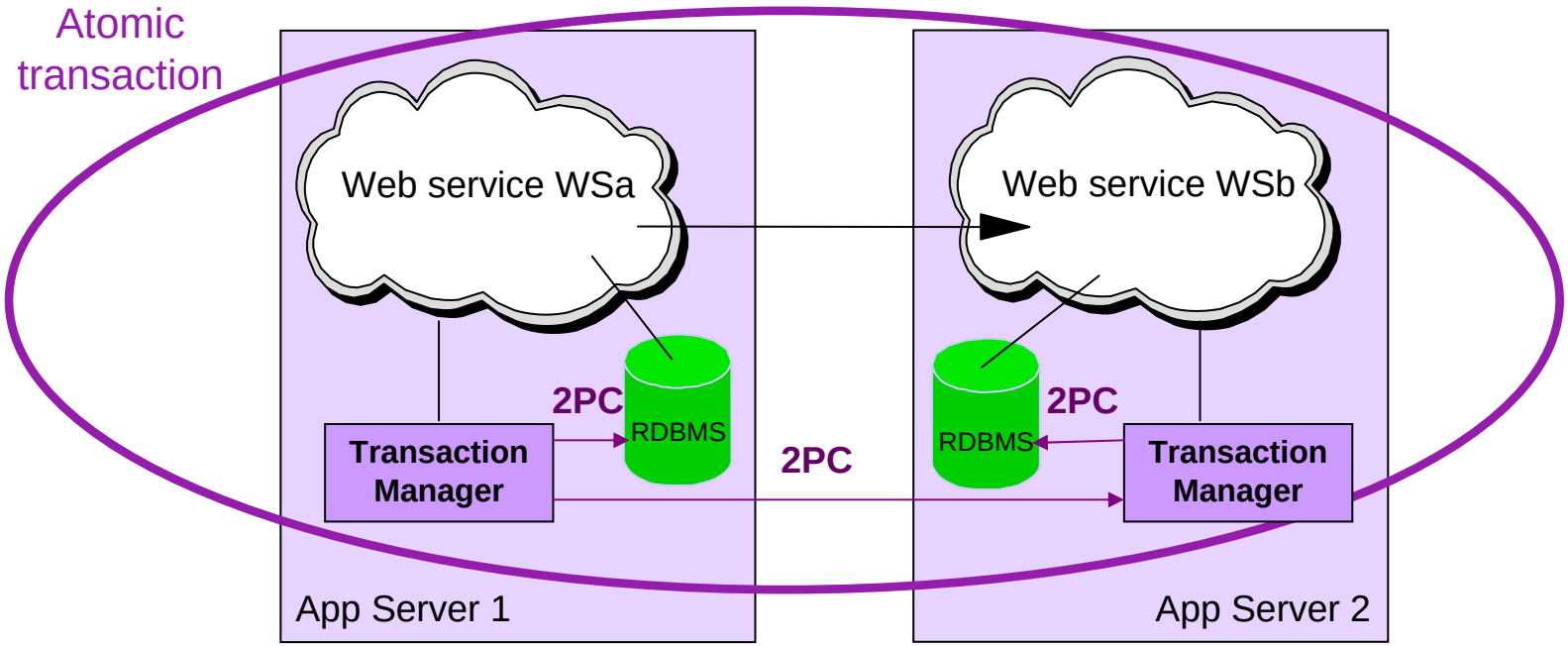
WS-Transaction architecture



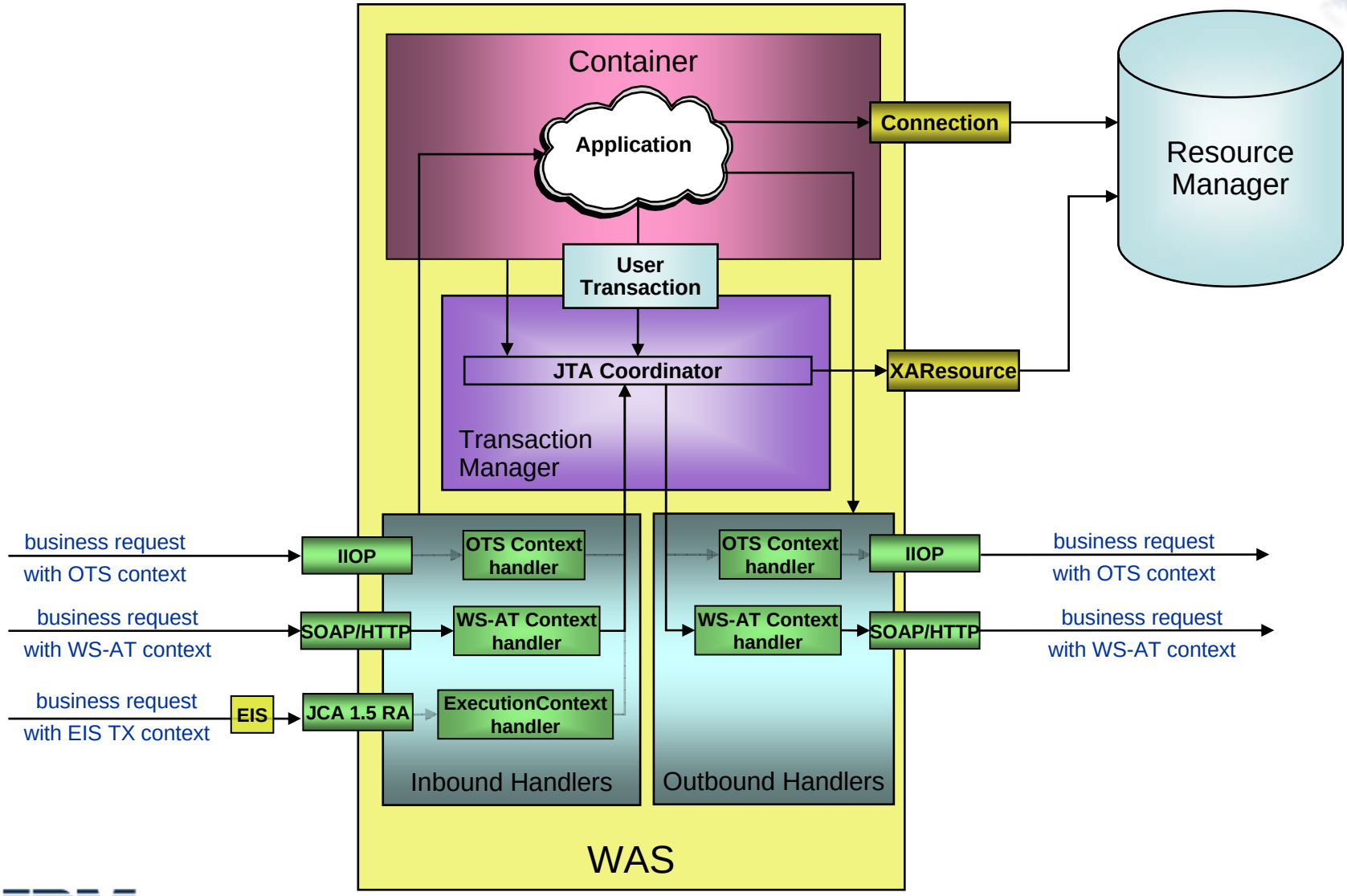
When is WS-AT useful?

- WS-AT enables the scope of ACID behaviour to be distributed between Web service components.
- BUT…
  - Resource locks are held throughout the transaction
  - Services that share an ACID transaction are not loosely coupled
- Primary uses:
  - transactions between services within a single organization domain
  - sometimes the *only* way to distribute an ACID transaction across different software stacks.

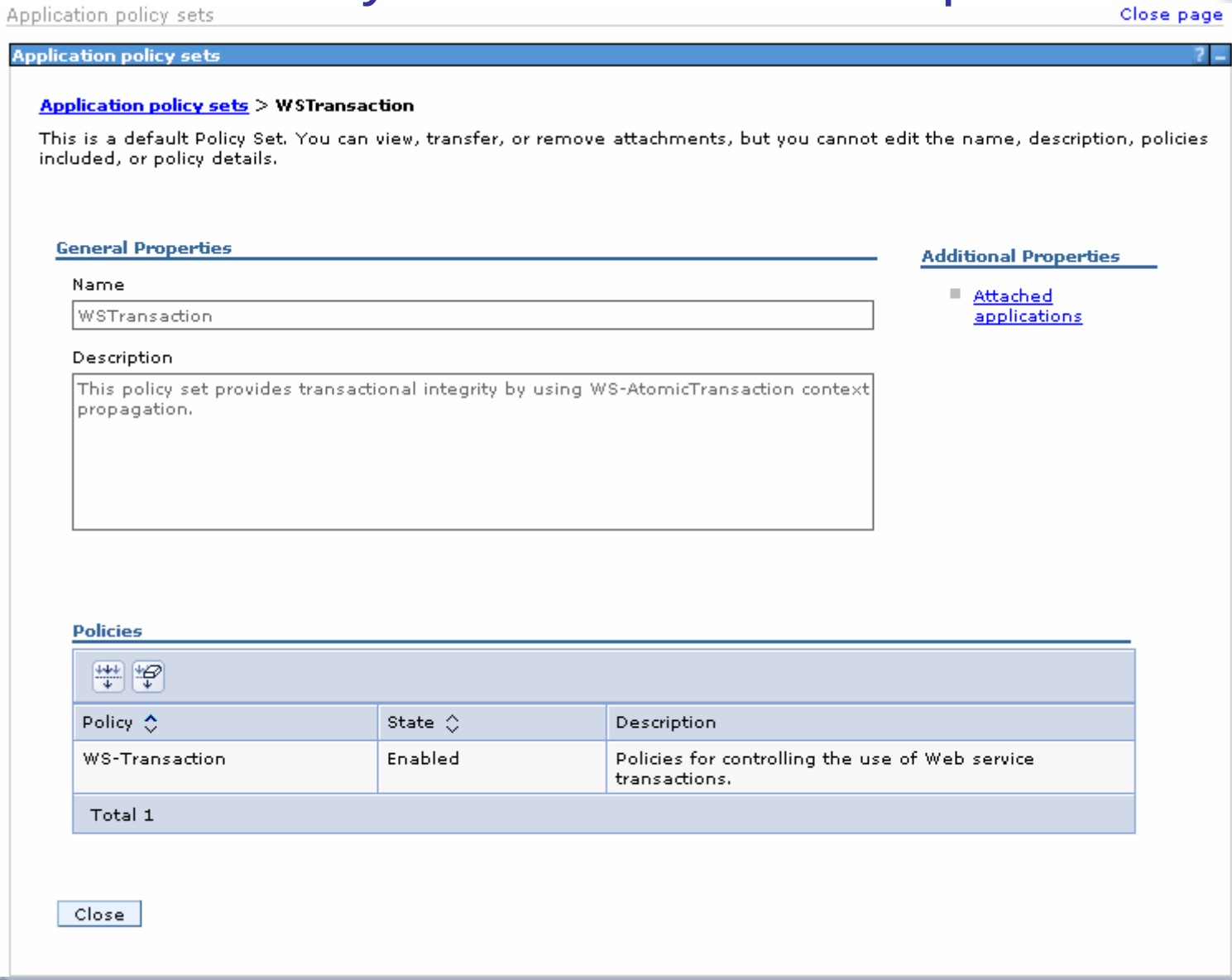
When is WS-AT useful?



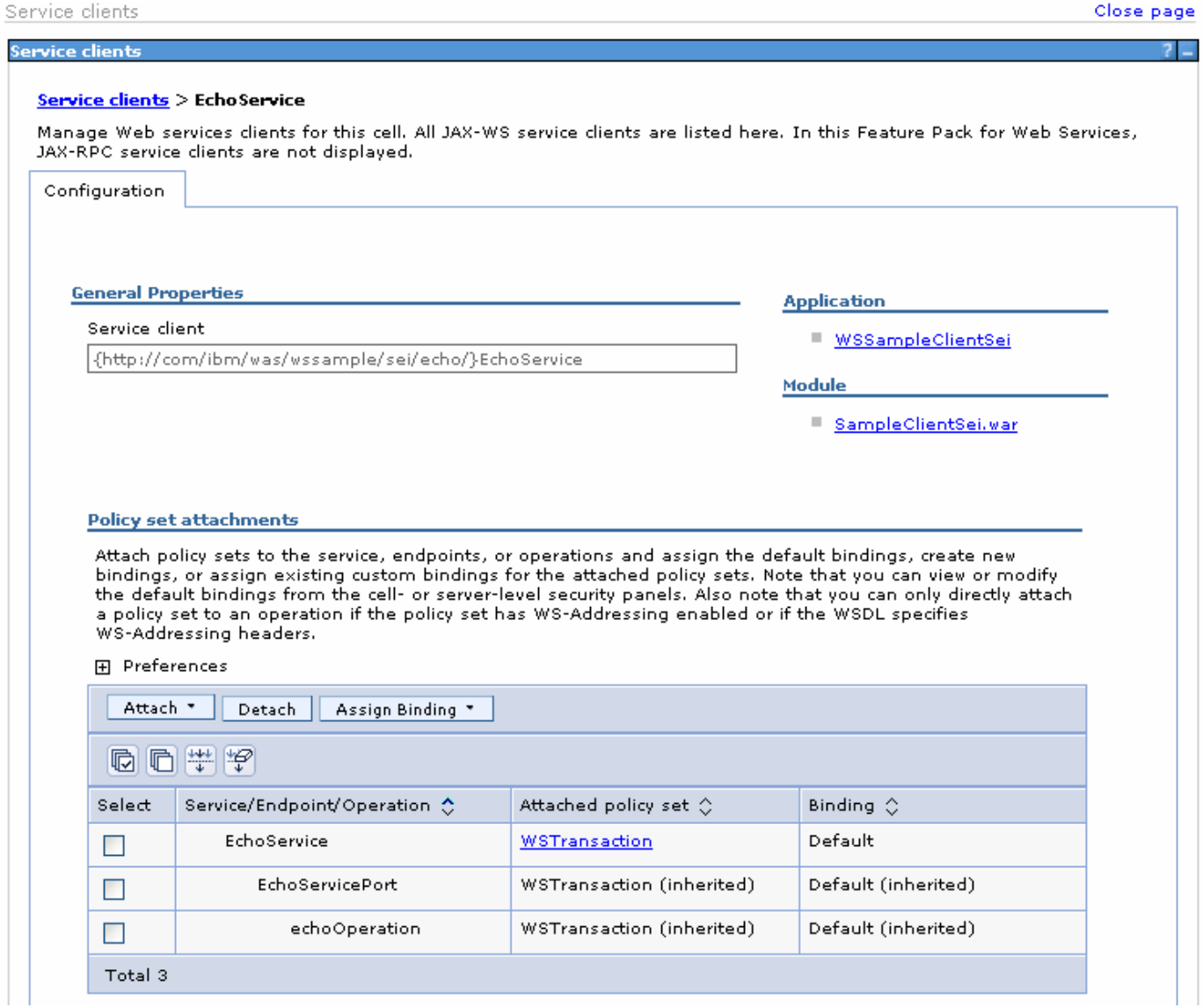
WS-AT in WAS: No coding to engage WS-AT



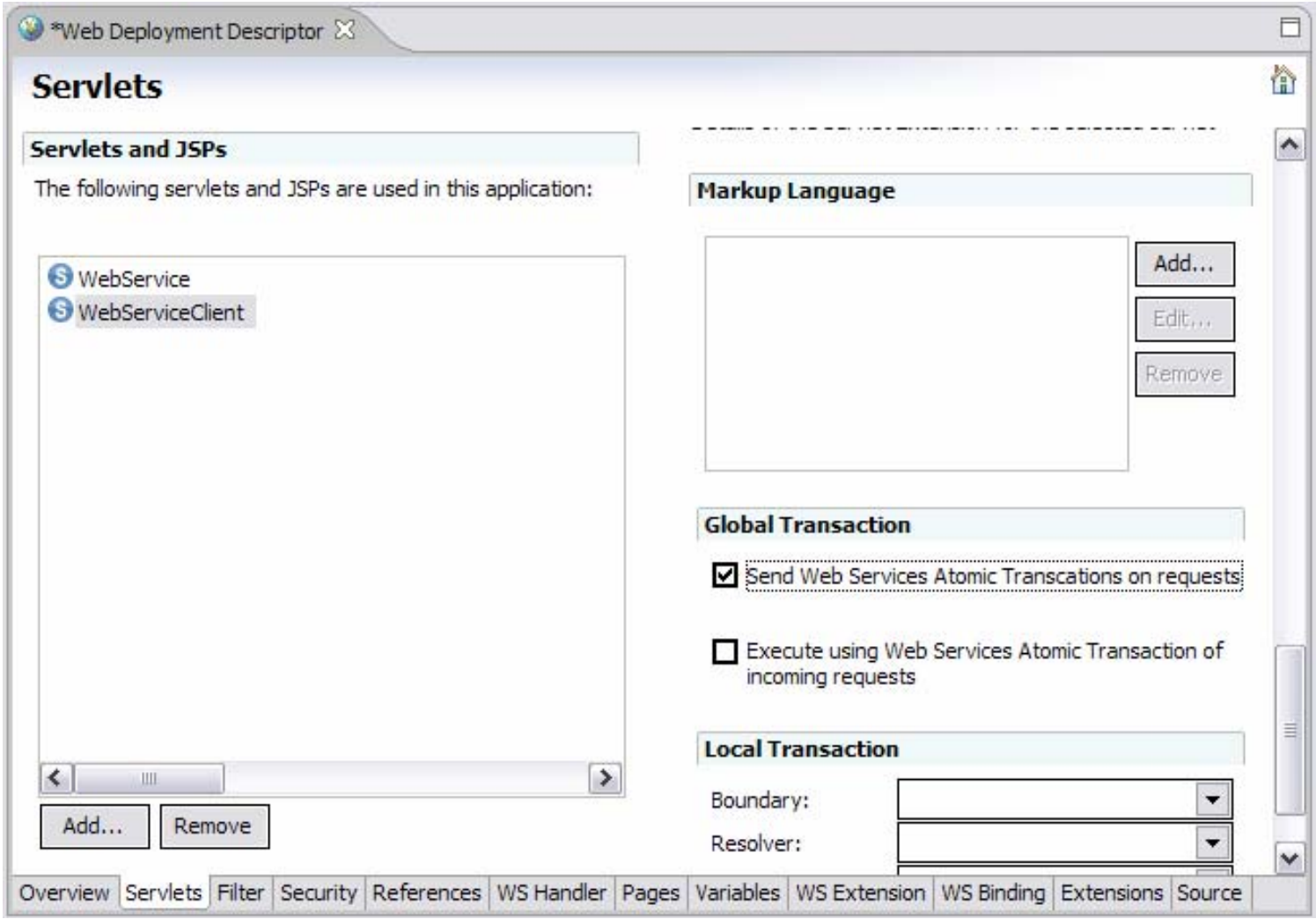
WS-AT assembly for JAX-WS clients/providers



WS-AT assembly for JAX-WS clients/providers



WS-AT assembly for JAX-RPC clients/providers

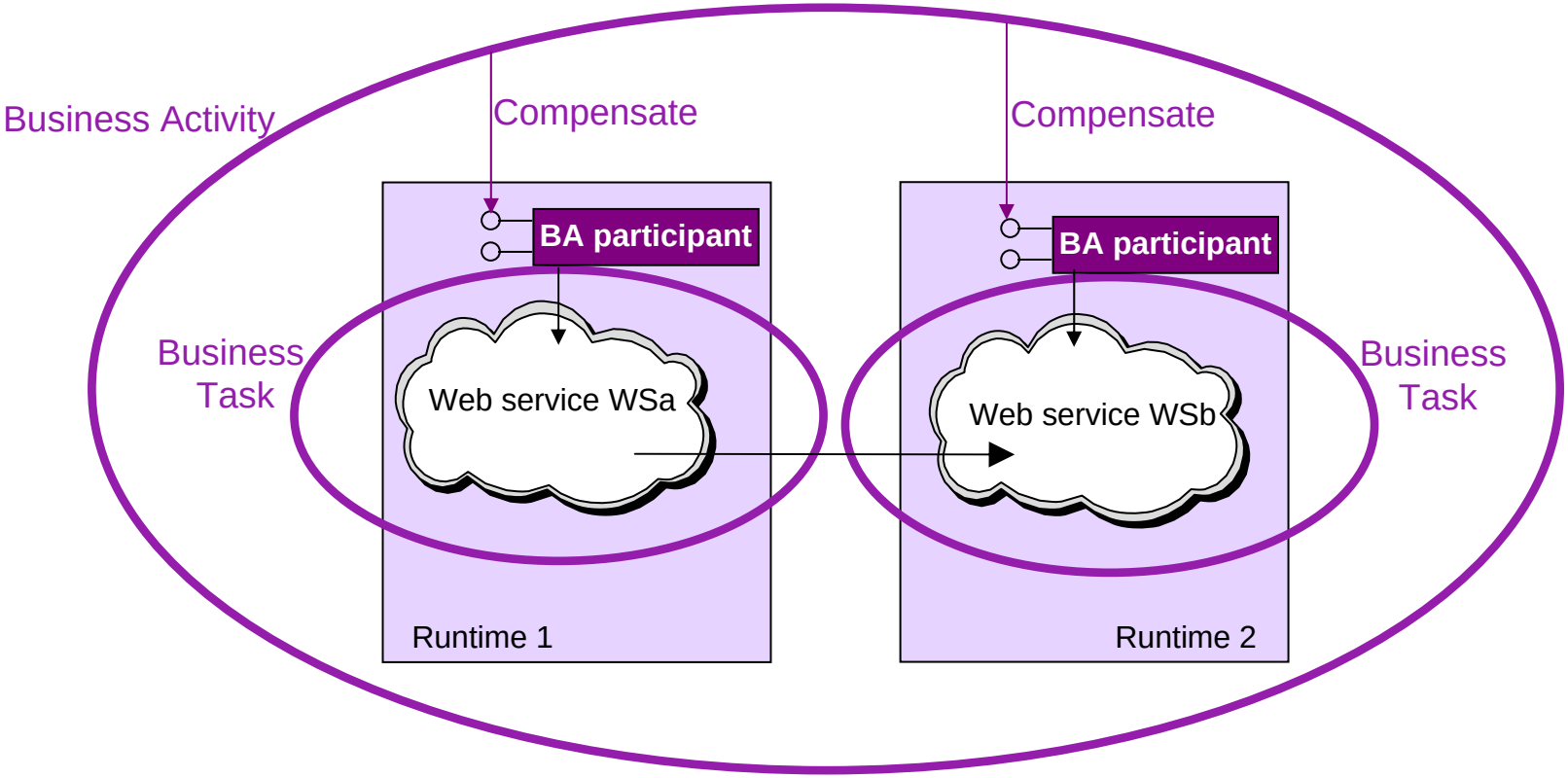


When is WS-BA useful?

- WS-BA provides a different form of atomicity from WS-AT. Participant are still brought to an atomic outcome but:
  - there is no isolation of resources
  - application-level compensation is required instead of resource manager rollback
- Also appropriate for longer-running and more loosely-coupled interactions.
  - but don't equate compensation with "looser coupling".
- A non-process-oriented approach *c.f.* WS-BPEL
- Primary uses:
  - business transactions that span organizational boundaries
  - distributed processes that use non-transactional resources

When is WS-BA useful?

- Business Tasks execute within the scope of a Business Activity
- Updates are exposed before Business Activity completes
- Completed participants receive common outcome



WAS *"BAScopes"*

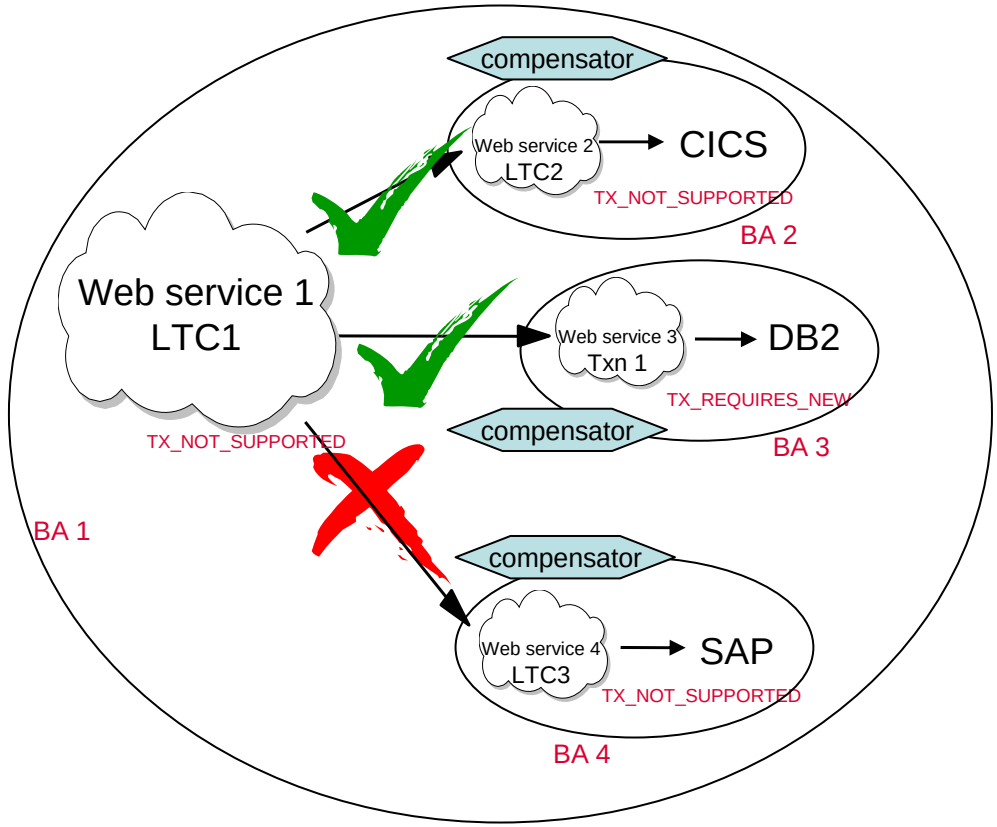
- …provides a compensating transaction model for **EJBs** that may (but do not have to) be exposed as Web services
  - *Uses WS-BA across Web service protocols*
- …enables 1PC or 2PC work to be committed as part of a larger activity and later compensated if the overall activity fails
  - *Activation of the CompensationHandler can be transacted as part of the forward transaction that it compensates.*
- …follows the BPEL compensation model for nested scope activities.

More on BAScopes

- The scope of a business activity, a BAScope, is that of a core WebSphere Application Server unit of work (UOW).
  - JTA Transaction
  - ActivitySession
  - Local Transaction Containment (LTC)
- A BAScope is *not* a new UOW; it is an attribute of an existing core UOW, therefore there is a one-to-one relationship between a BAScope and a UOW.
  - The outcome of a BAScope mirrors the outcome of its associated core UOW.
- A *child* BAScope is implicitly created if an EJB running under core UOW *A* calls an EJB running under core UOW *B*.
  - CompensationHandlers are implicitly promoted from a child to a parent BAScope.

Nested BAScopes

- An activity is a scope within which a single consistent outcome is provided.
- Successful work has compensators promoted
- Compensators are closed or compensated according to direction of top-level scope.



Using WS-BA functionality in WAS

1. Enable the application server.
   - By default the WS-BA functionality is disabled. This needs to be enabled on each application server planning to exploit the WS-BA function.
2. Create a CompensationHandler class.
   - An implementation of the CompensationHandler interface needs to be created for the WS-BA application component to reference and use at runtime.
3. Enable the application components.
   - Each application component needs to be configured using the RAD/AST tooling to enable WS-BA on the component and by setting a CompensationHandler class for that component.

Programming Model: WS-BA APIs

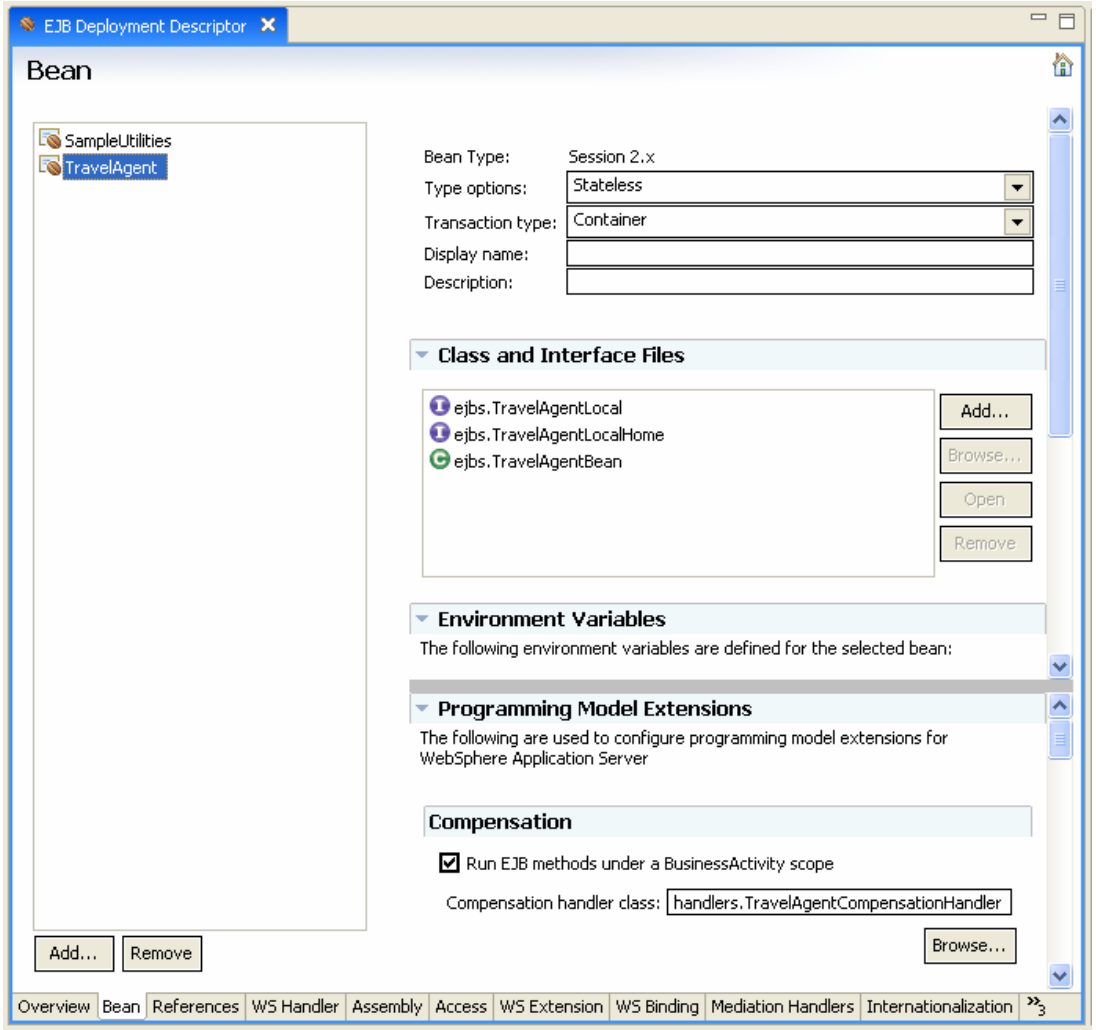
```
package com.ibm.websphere.wsba;
public interface UserBusinessActivity
{
  boolean isCompensateOnly() throws java.lang.IllegalStateException
  void setCompensateOnly() throws java.lang.IllegalStateException
  void setCompensationDataAtCommit(Serializable compensationData)
                  throws java.lang.IllegalStateException, java.io.NotSerializableException
  void setCompensationDataImmediate(Serializable compensationData)
                  throws java.lang.IllegalStateException, java.io.NotSerializableException
}
```

JNDI location: java:comp/websphere/UserBusinessActivity

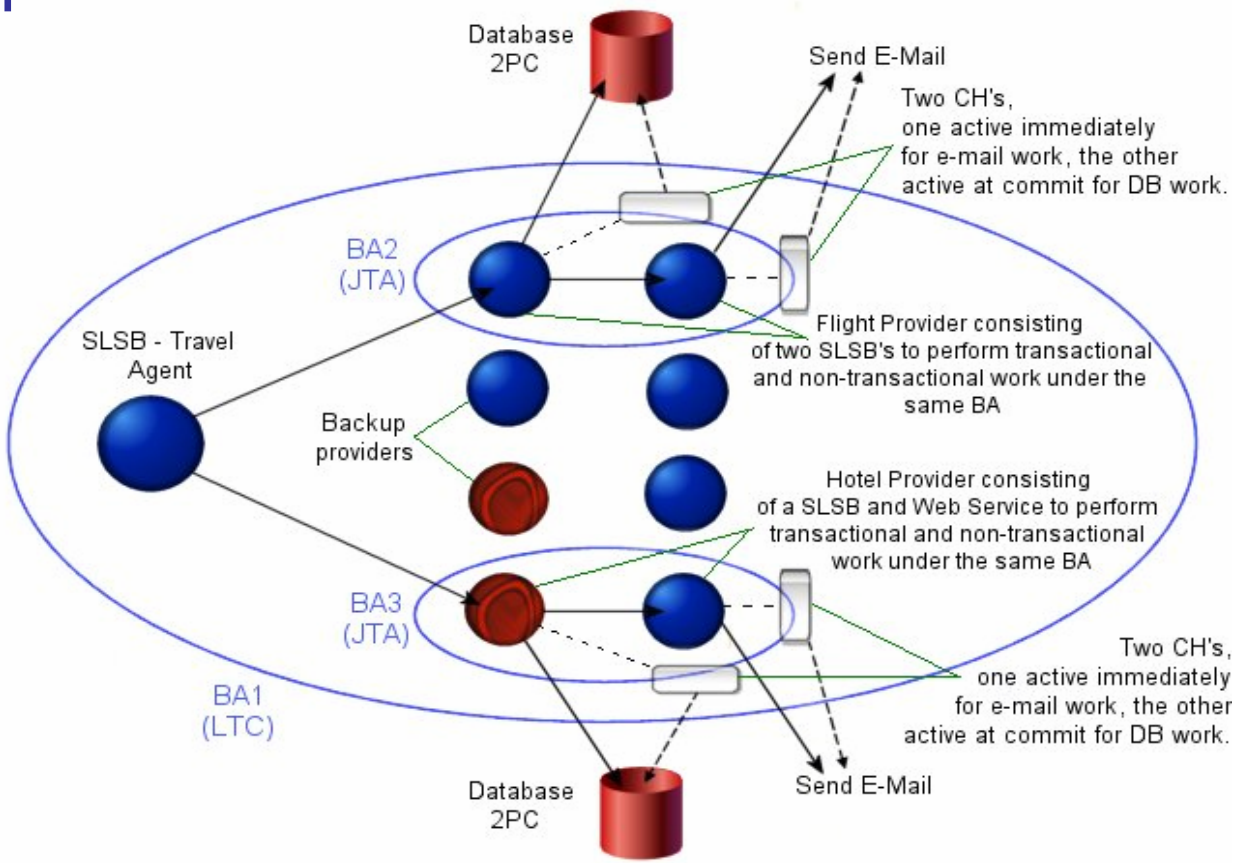
```
package com.ibm.websphere.wsba;
public interface CompensationHandler
{
  public void close(Serializable compensationData) throws
          RetryCompensationHandlerException, CompensationHandlerFailedException
  public void compensate(Serializable compensationData) throws
          RetryCompensationHandlerException, CompensationHandlerFailedException

}
```

WS-BA Step 3: Enable the application components



WS-BA Sample: Compensation and forward progress with multiple resources



http://www.ibm.com/developerworks/websphere/library/samples/wsba.html

What about the “NFRs”?

The TX specs define interoperable Web service transaction protocols. They don’t define “non-functional requirements” such as:

- Proxies and firewalls
- High availability and failover of TX endpoints
- Transaction-based workload management affinity

These are outside the scope of the specification. But any enterprise-level SOA runtime needs to accommodate these.

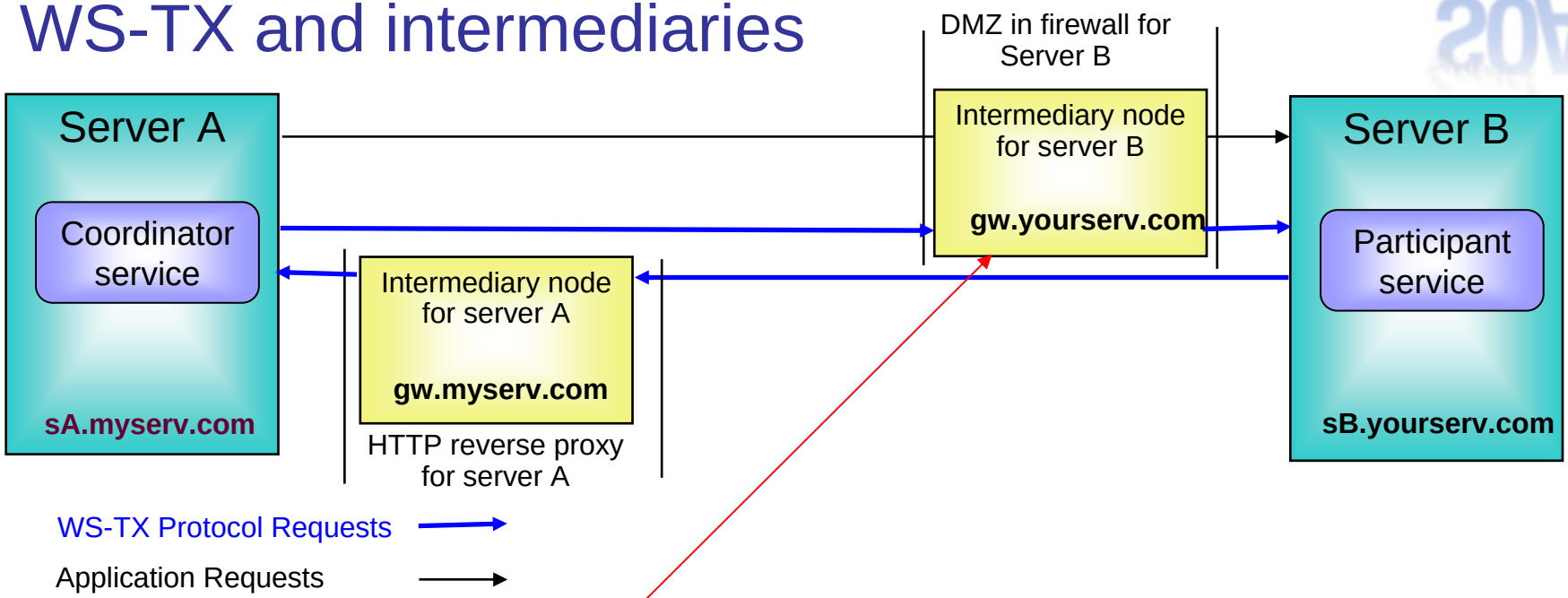
Where are these considerations factored and do they affect interoperability?

WS-Addressing EndpointReferences

- The TX endpoints exchanged are WS-Addressing EndpointReferences:
  - A “++” XML pointer.
  - Contains an address URI for the endpoint service
  - Contains a set of ***opaque*** “ReferenceParameter” tokens that augment the address with anything required by the target service or its runtime environment.

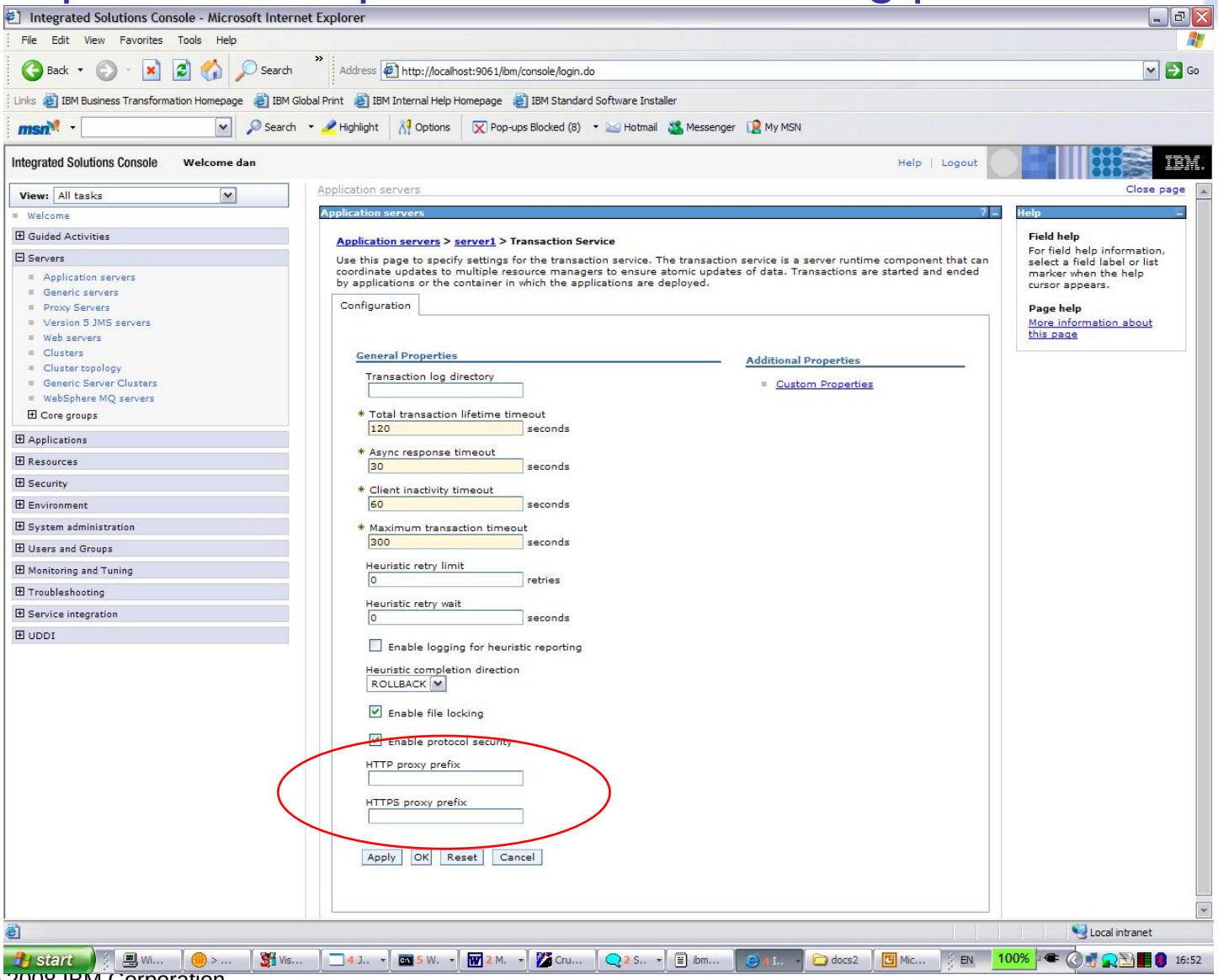
```
<wsa:EndpointReference>
    <wsa:Address>http://wsgw.fabrikam.com/_IBMSYSAPP/wsat/services/Participant</wsa:Address>
    <wsa:ReferenceParameters>
        <someNS:txID>111-222-333</someNS:txID>
        <was:HAclusterId>47563940008426597832759 3</was:HAclusterId>
     </wsa:ReferenceParameters>
</wsa:EndpointReference>
```

WS-TX and intermediaries




```
<wsa:To>
   <wsa:Address>http://gw.yourserv.com/_IBMSYSAPP/wsat/services/Participant</wsa:Address>
</wsa:To>
<someNS:txID wsa:IsReferenceParameter='true'>111-222-333</someNS:txID>
<was:HAclusterId wsa:IsReferenceParameter='true'>475639400084265978327593</was:HAclusterId>
<wsa:From>
   <wsa:Address>http://gw.myserv.com/_IBMSYSAPP/wsat/services/Coordinator</wsa:Address>
   …
</wsa:From>
```

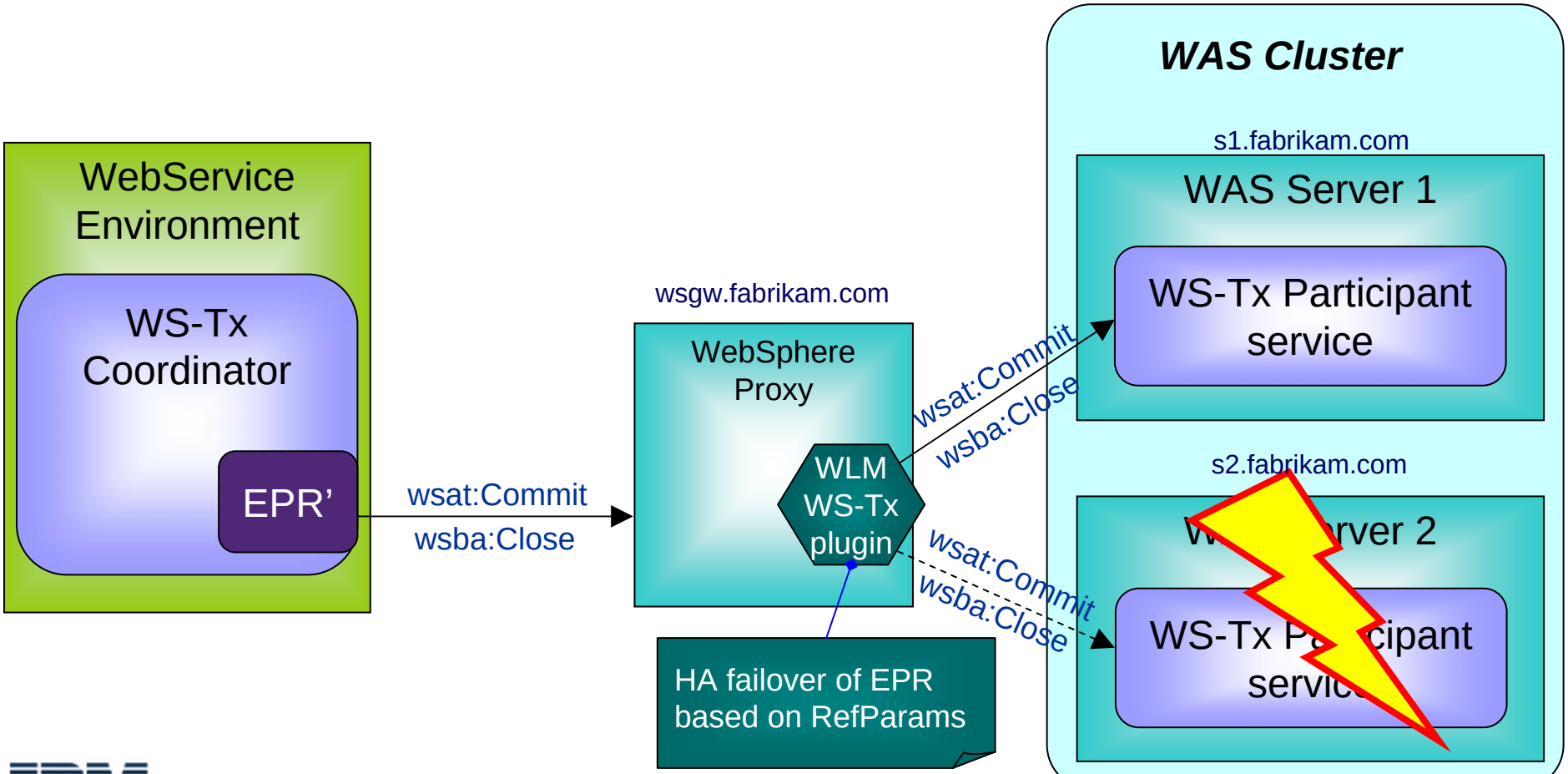
WS-TX protocol endpoints accommodating proxies



High availability failover of protocol endpoints

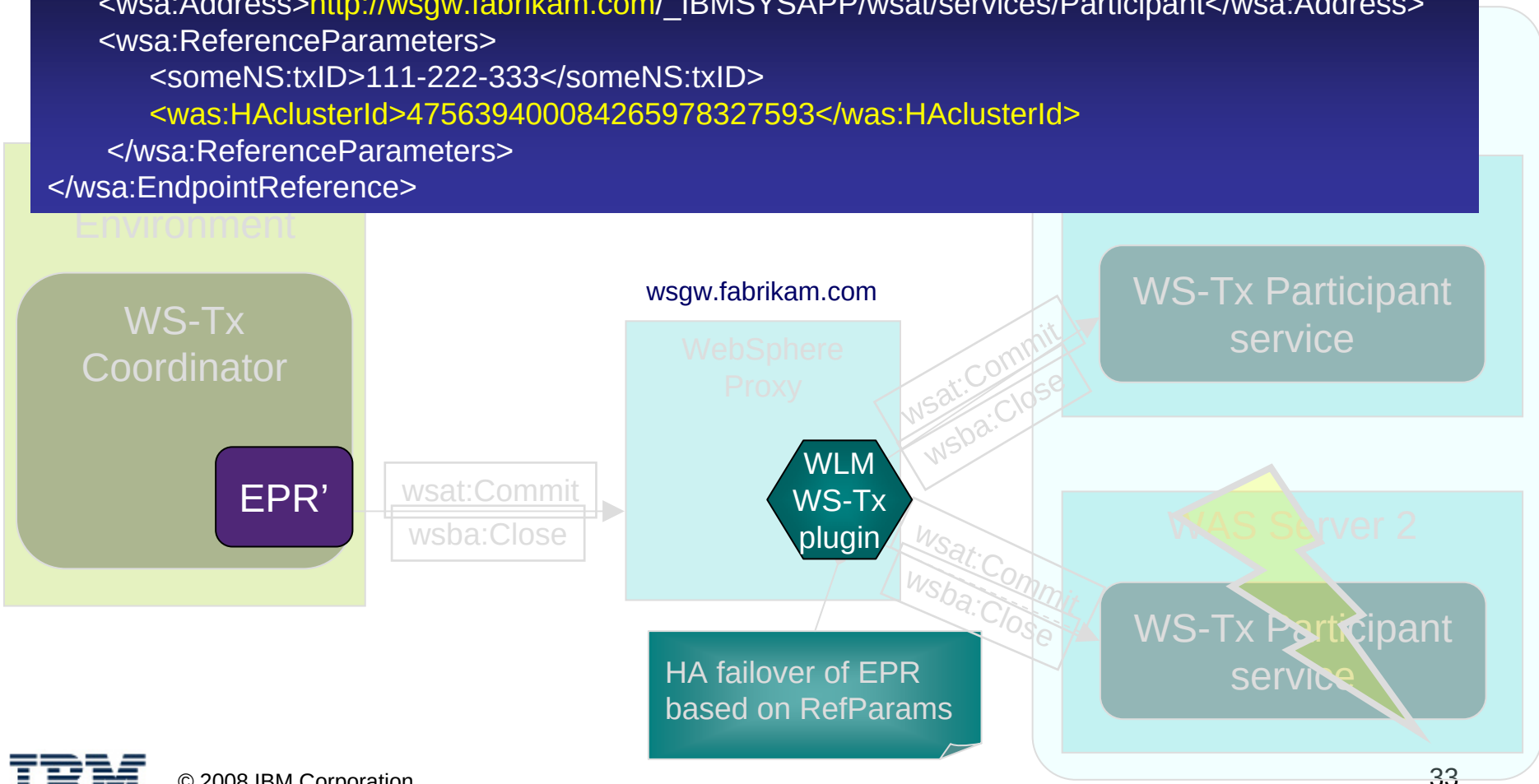
EPRs from each server contain a virtual host address mapped by the proxy.

RefParams in the EPR are used by WLM plug-in to “follow” HA failover.



High availability failover of protocol endpoints

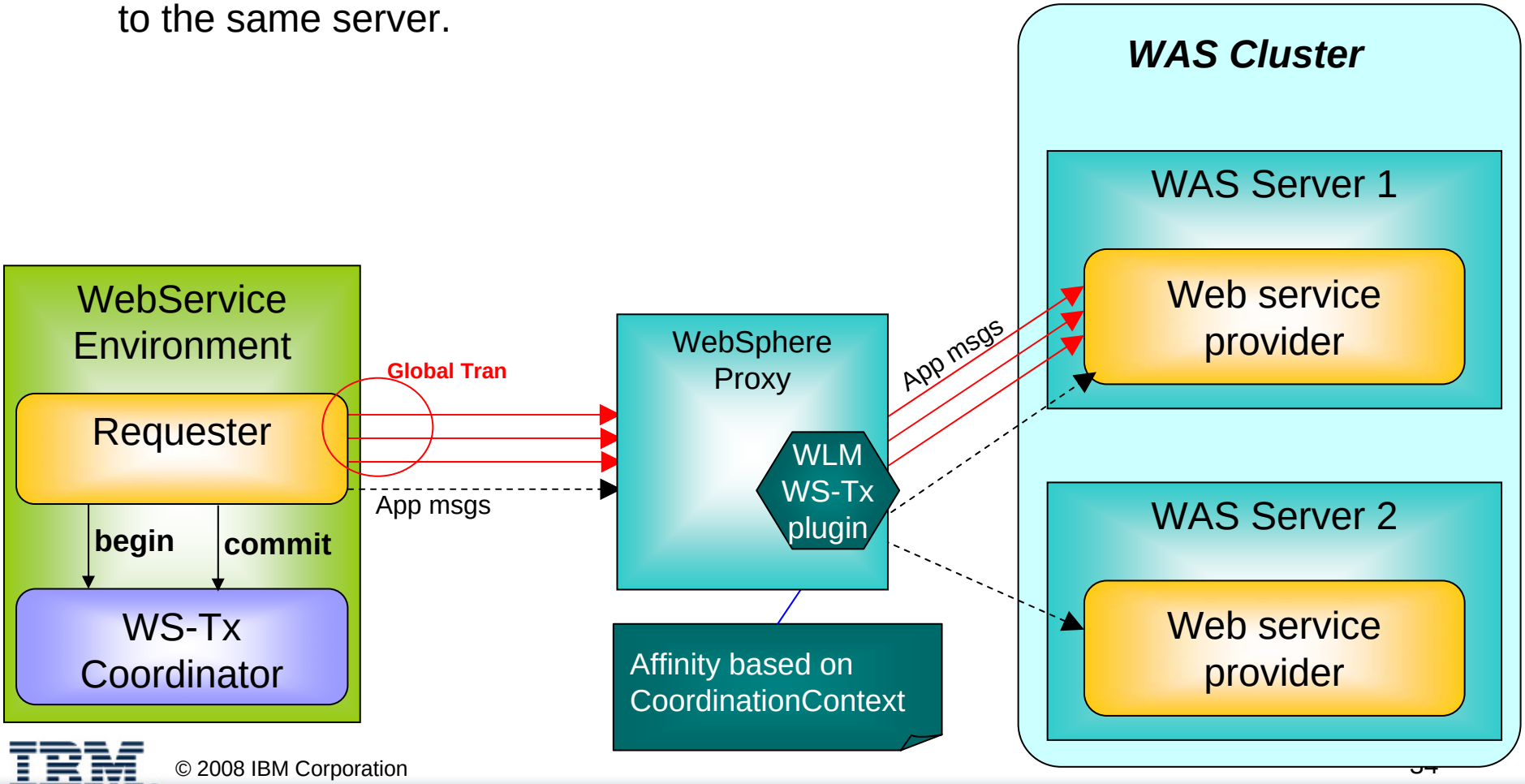
```
<wsa:EndpointReference>
    <wsa:Address>http://wsgw.fabrikam.com/_IBMSYSAPP/wsat/services/Participant</wsa:Address>
    <wsa:ReferenceParameters>
        <someNS:txID>111-222-333</someNS:txID>
        <was:HAclusterId>47563940008426597832759 3</was:HAclusterId>
     </wsa:ReferenceParameters>
</wsa:EndpointReference>
```



Transaction-based affinity routing

WLM transaction affinity constraints.

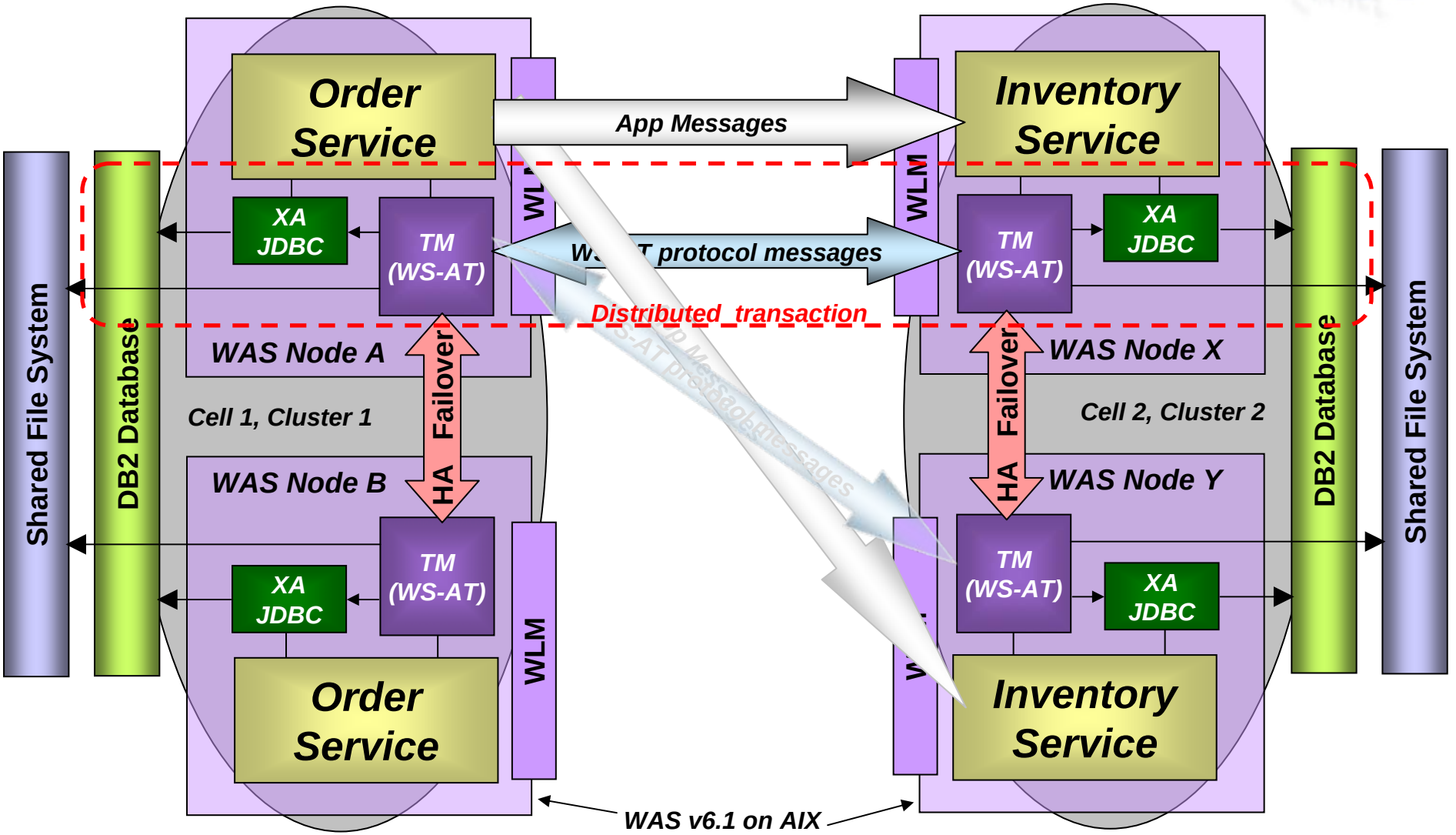
All requests to WLM-able EPRs within the same WS-Tx context targeted to the same server.



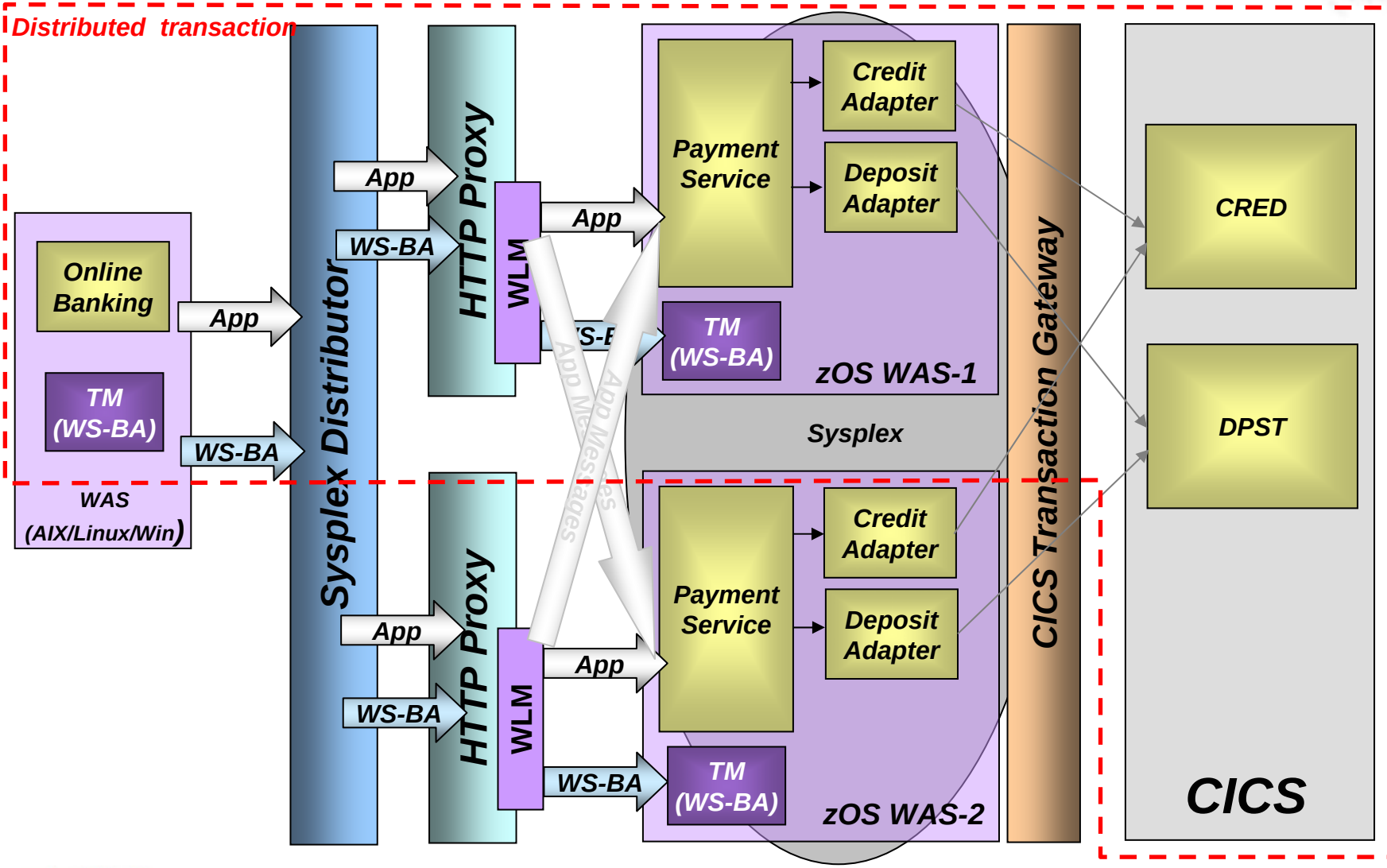
Transaction-based affinity routing

```
<soap:Header>
  <wscoor:CoordinationContext>
     <wscoor:Identifier>
              uuid:33ca57d4-eaab-4939-8177-77351e6e63c7
     </wscoor:Identifier>
     <wscoor:CoordinationType>
              http://docs.oasis-open.org/ws-tx/wsat/2006/06
     </wscoor:CoordinationType>
     …
  </wscoor:CoordinationContext>
  …
</soap:Header>
```

Order Management Application: HA WS-AT deployment



Online Banking: Proxied/WLM'd WS-BA deployment



Summary

- WS-Transaction defines a generic framework for Web service coordination protocols, as well as two concrete protocols for atomic and compensating transactions.
- The WAS support additionally provides for all the “non-functional requirements” typically necessary for enterprise deployments, in both a transparent and an interoperable fashion.
- Applications that use WS-AT and WS-BA and exploit these enterprise qualities of service have been successfully deployed and demonstrate the viability of these advanced technologies in the real world.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.
Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including 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 results similar to those stated here.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries. For a complete list of IBM trademarks, see www.ibm.com/legal/copytrade.shtml
AIX, CICS, CICSPlex, DB2, DB2 Universal Database, i5/OS, IBM, the IBM logo, IMS, iSeries, Lotus, OMEGAMON, OS/390, Parallel Sysplex, pureXML, Rational, RCAF, Redbooks, Sametime, Smart SOA, System i, System i5, System z , Tivoli, WebSphere, and z/OS.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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

Coordination Context (I)

```
<wscoor:CoordinationContext>
   <wscoor:Identifier>
         uuid:33ca57d4-eaab-4939-8177-77351e6e63c7
   </wscoor:Identifier>
   <wscoor:Expires>60000</wscoor:Expires>
   <wscoor:CoordinationType>
         http://schemas.xmlsoap.org/ws/2004/10/wsat
   </wscoor:CoordinationType>
   <wscoor:RegistrationService>
      <wsa:Address>http://wsatserver/wscoor/Registration</wsa:Address>
      <wsa:ReferenceParameters>
         <tm:txId>246</tm:txId>
      </wsa:ReferenceParameters>
   </wscoor:RegistrationService>
   …
</wscoor:CoordinationContext>
```

Note: Applications don’t see or care about this

Coordination Context (II)

Flows as a SOAP header on application requests

```
<s:Envelope>
   <s:Header>
      <wsa:Action>http://tempuri.org/application/action</wsa:Action>
      <wsa:To>http://server/service/</wsa:To>
      <wscoor:CoordinationContext s:mustUnderstand="true">
          ...
      </wscoor:CoordinationContext>
   </s:Header>
   <s:Body>
      <!-- Application body content -->
   </s:Body>
</s:Envelope>
```

Note: Applications don't see or care about this