IBM 2007 IMPACT

IBM

CICS Web services atomic transactions

Nigel Williams (IBM Design Center)

nigel_williams@uk.ibm.com

Session 4134B

impact-venture*

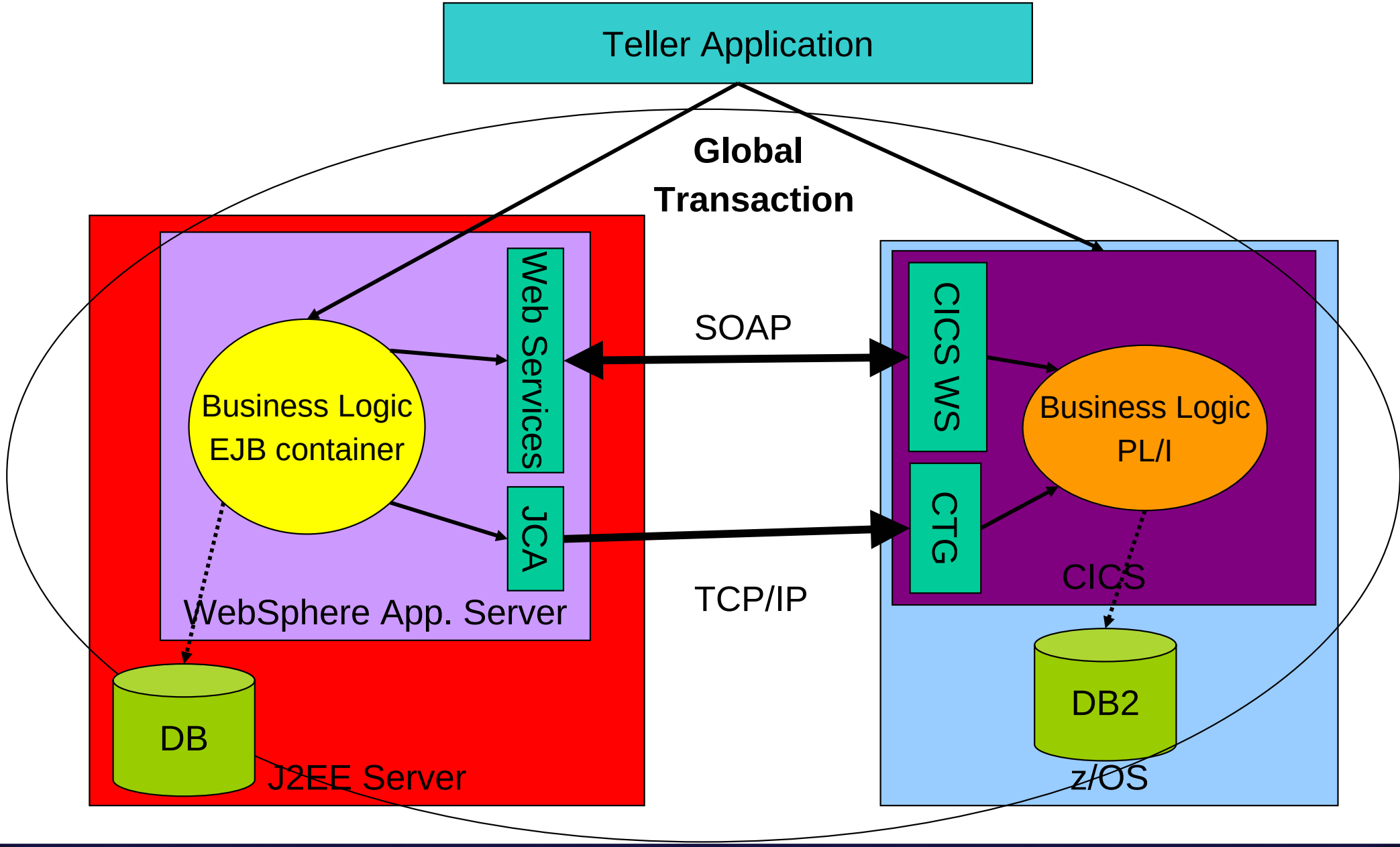
Background

- Savings Bank has existing core business applications based on PL/1 – CICS – DB2
- Plans to build new business logic in J2EE
- PL/1 and J2EE will coexist for a long time, so reliable transactions across both environments are required
- Transactions can start in CICS or in WebSphere so support for bi-directional communication with 2PC is needed (for **some** transactions)
- If no 2PC support, need to build compensation logic that increases development costs and complexity

Objectives

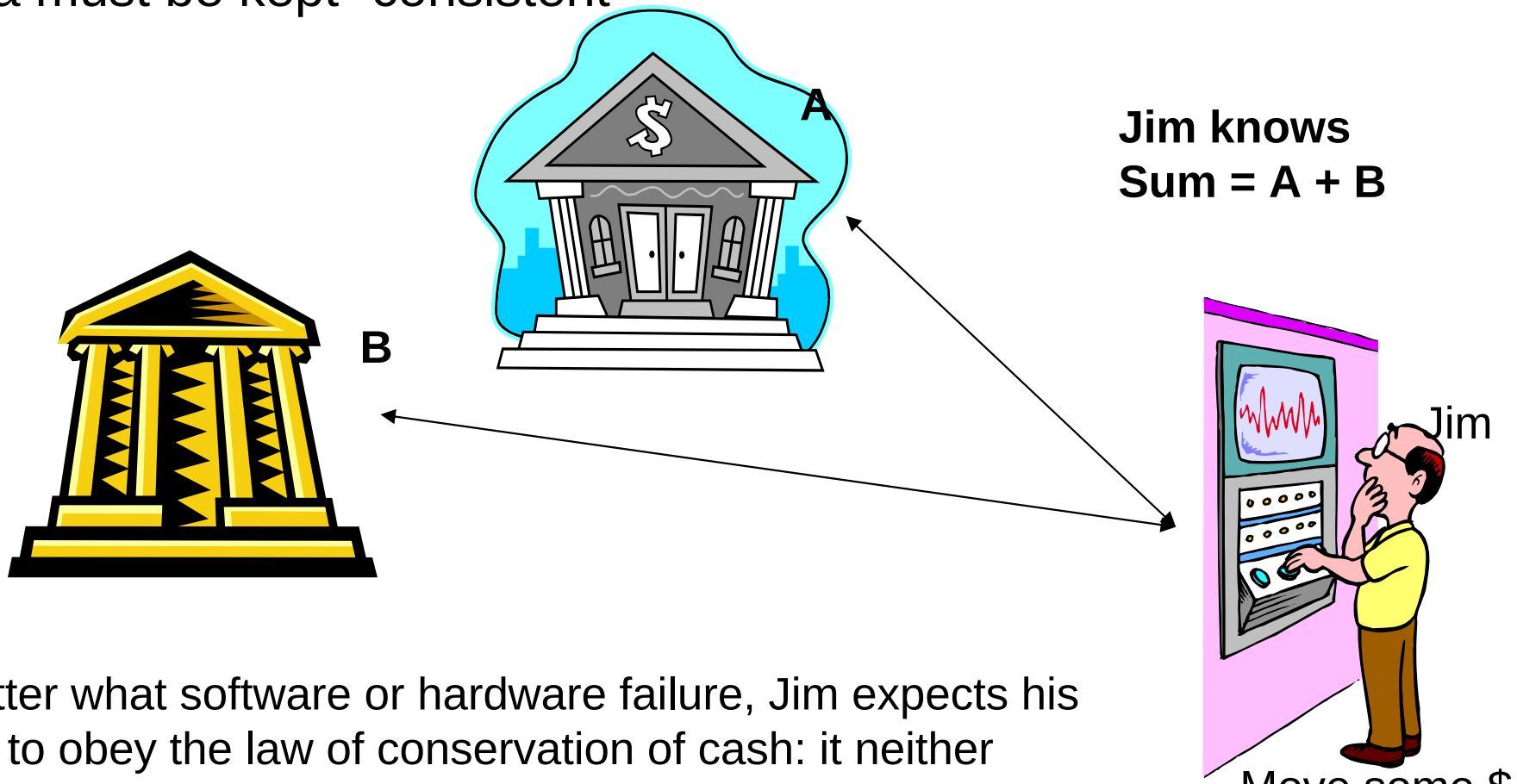
- Why distributed transactions are required at Savings Bank
- What technology options were considered for proof of concept
- Review of Web services atomic transactions
- Results of proof of concept
- Conclusions

Planned Architecture



Why 2PC ?

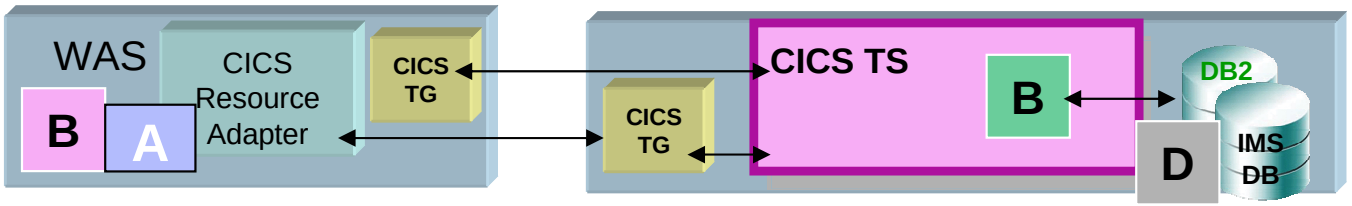
- Data must be kept “consistent”



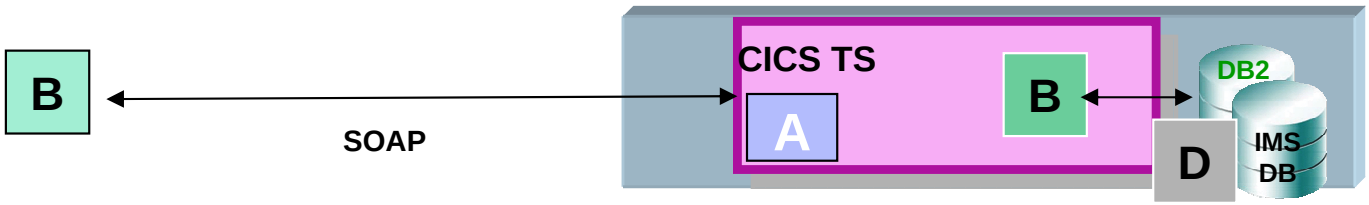
No matter what software or hardware failure, Jim expects his money to obey the law of conservation of cash: it neither evaporates nor suddenly appears from nowhere (the latter is acceptable to him, but not to the bank).

Technology options considered

JCA Connector:
CICS Transaction Gateway

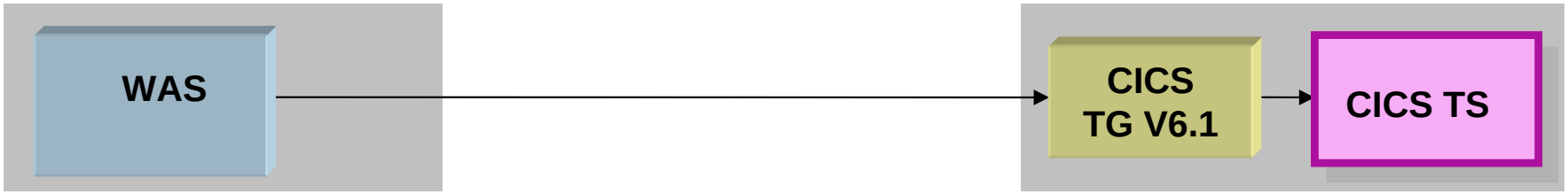


SOAP Access:
CICS Web services Support



CICS TG 2PC transaction support

Distibuted platform **z/OS**



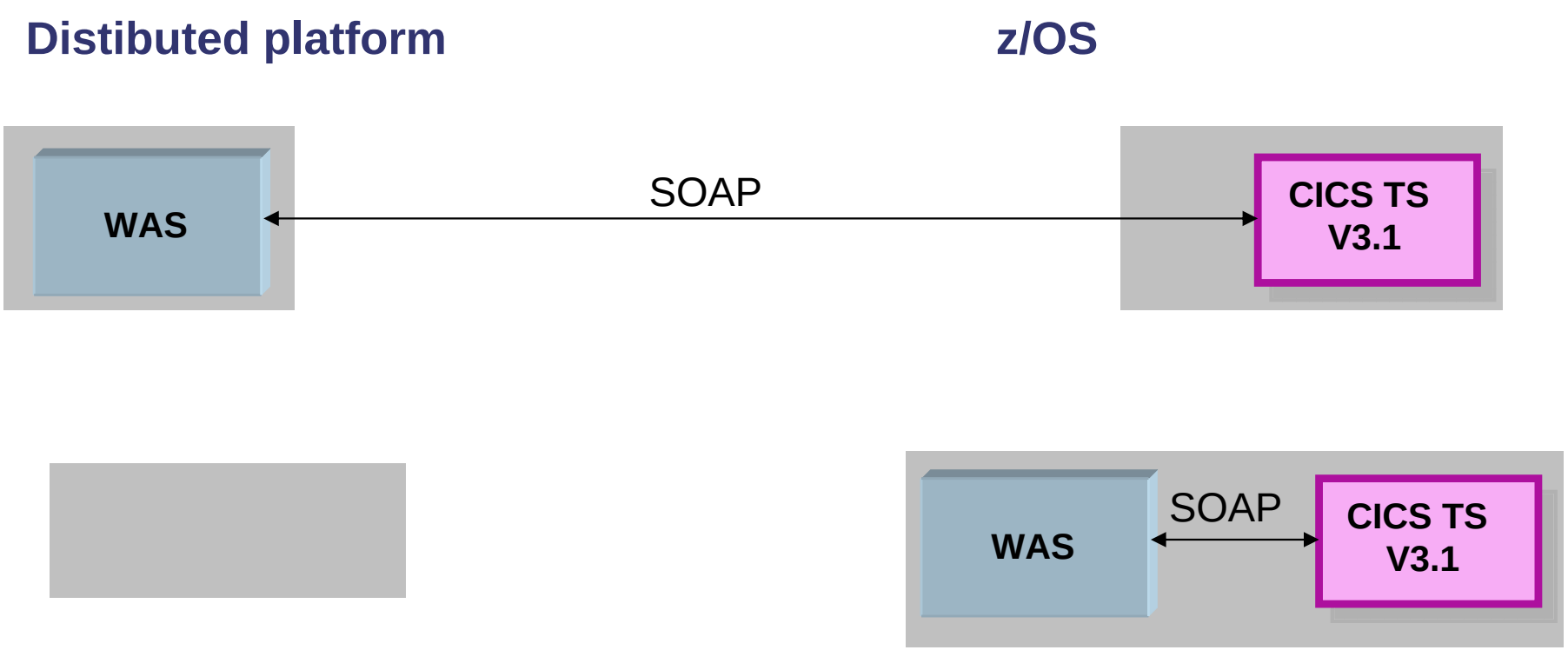
-2PC introduced with XA support in CTG V6.1

-CICS TG must run on z/OS

-2PC through RRS

- Requires CICS TG 'local' mode (not through TCP/IP)

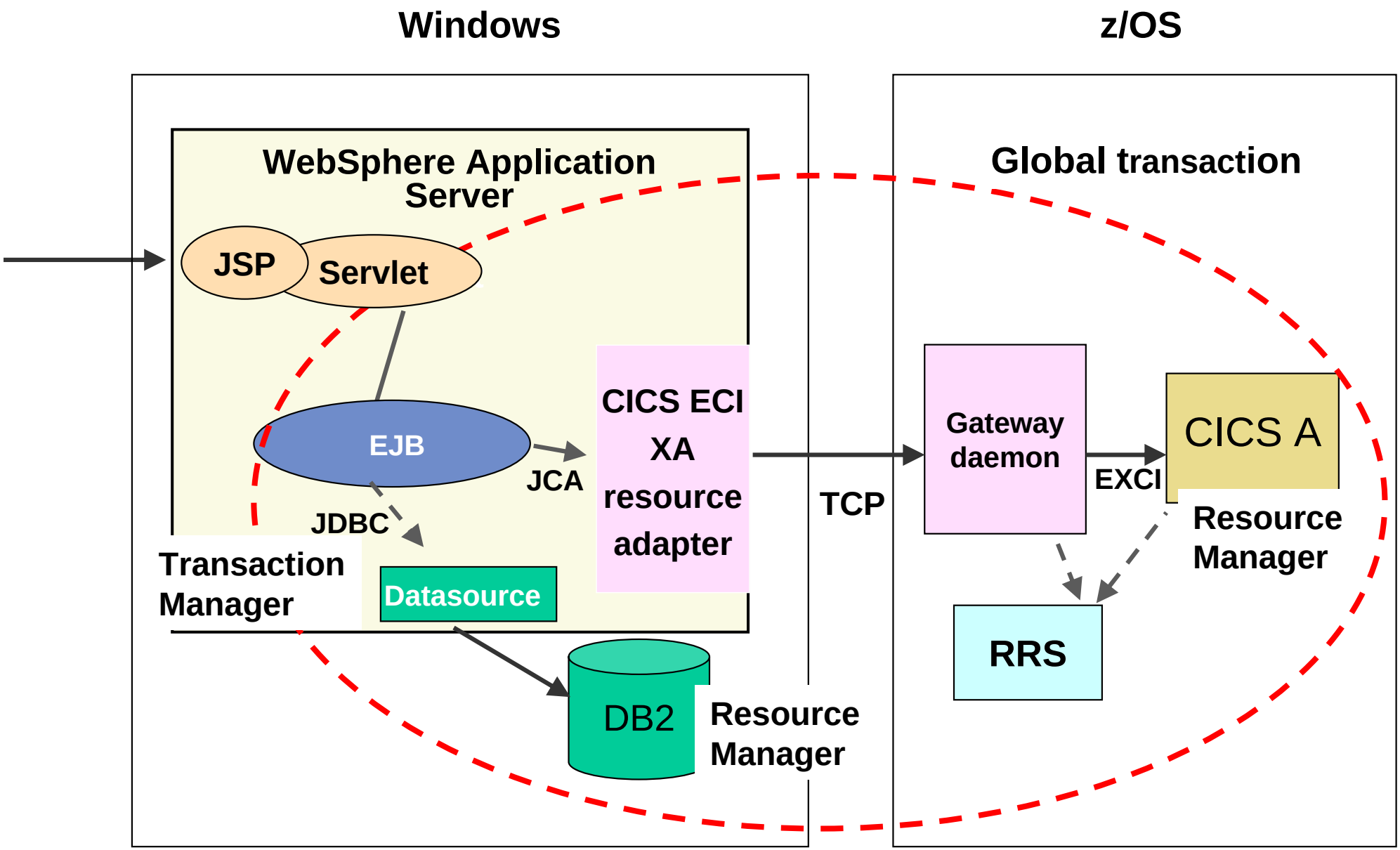
CICS Web services 2PC transaction support



-2PC through WS-AtomicTransaction support in CICS TS V3.1 (as service provider and as service requester)

- WAS V6 and later supports WS-AtomicTransaction (WS-AT)

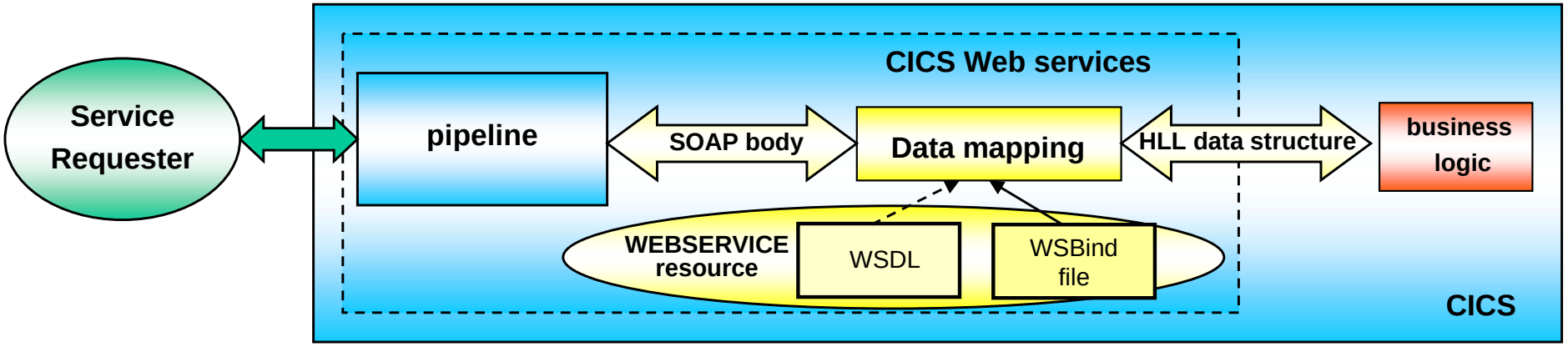
CICS TG V6.1 2PC support



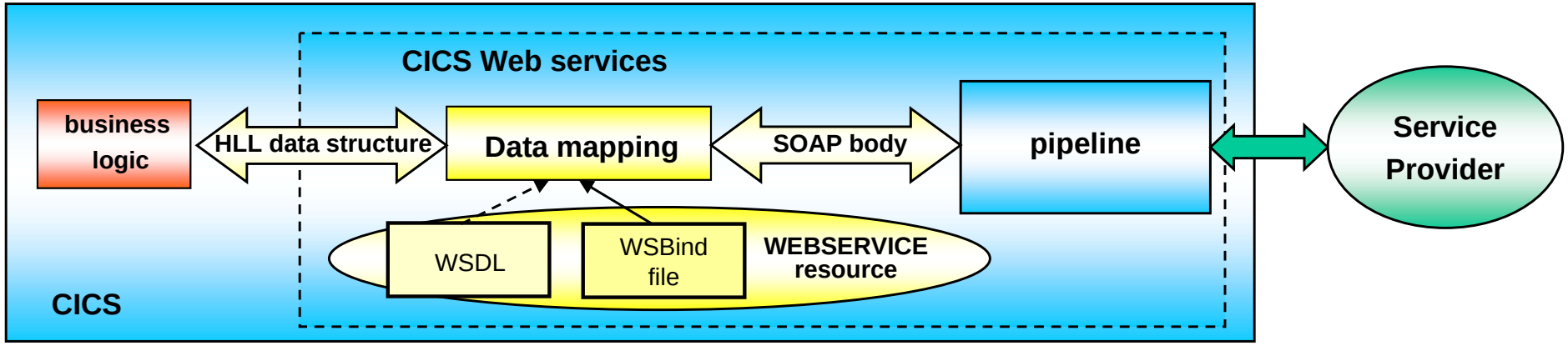
Review: Web services atomic transactions

CICS Web Services support (overview)

- CICS as a service provider

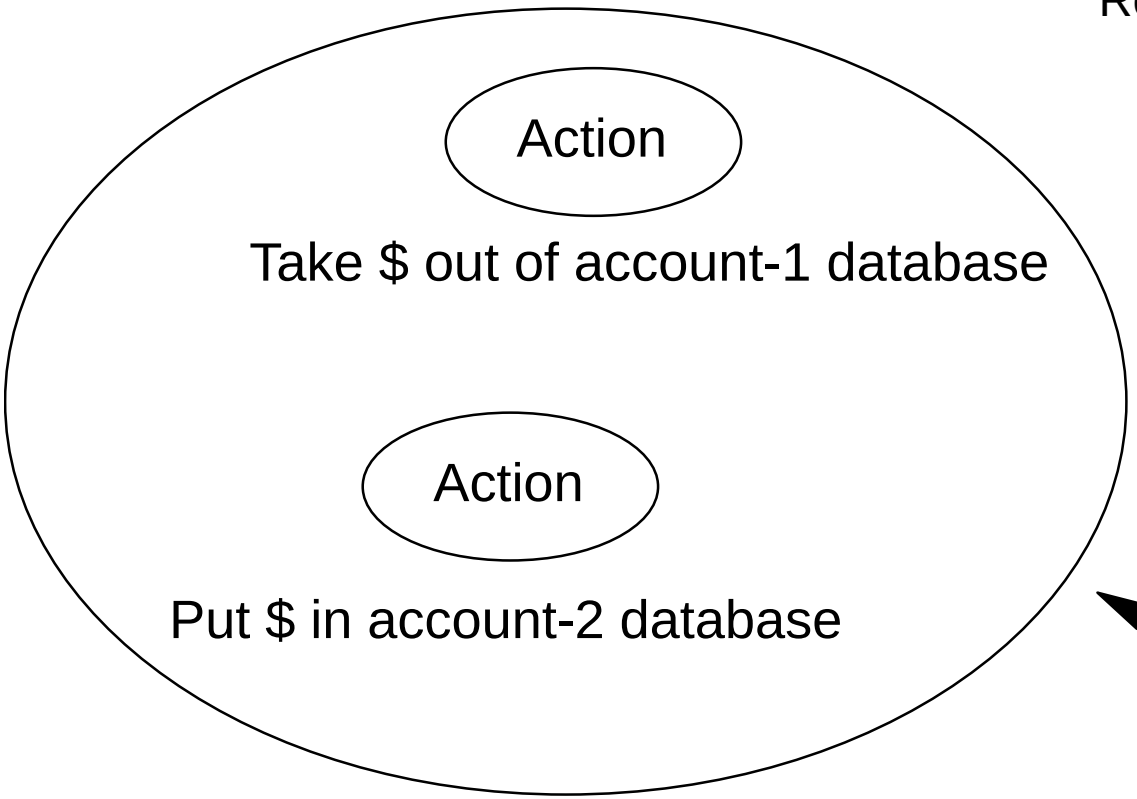


- CICS as a service requester



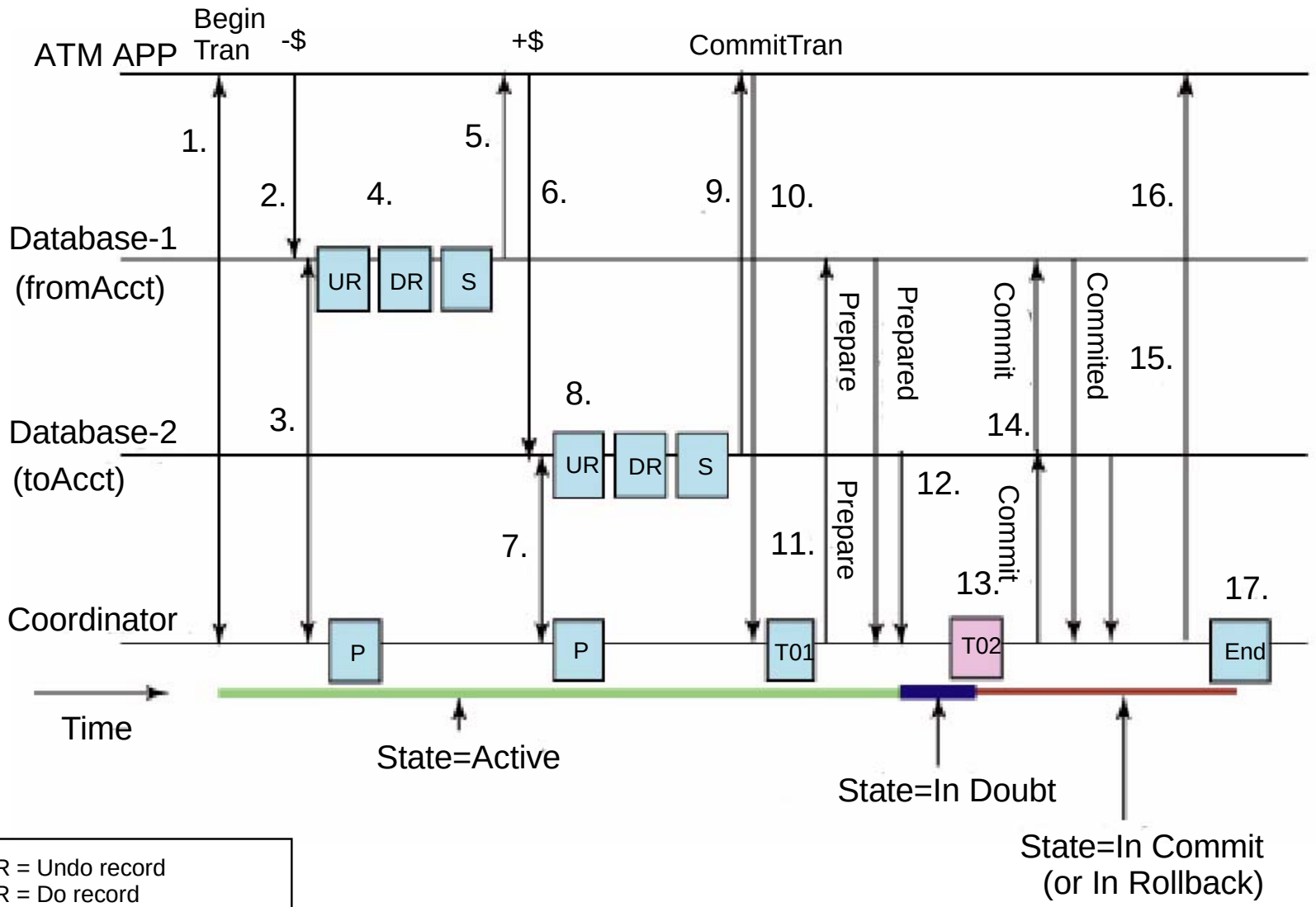
Simple to code a transaction

```
TransferCash(fromAcct, toAcct, amount)
    BeginTransaction
        fromAcct = fromAcct - amount
        toAcct = toAcct + amount
    CommitTransaction
Return
```



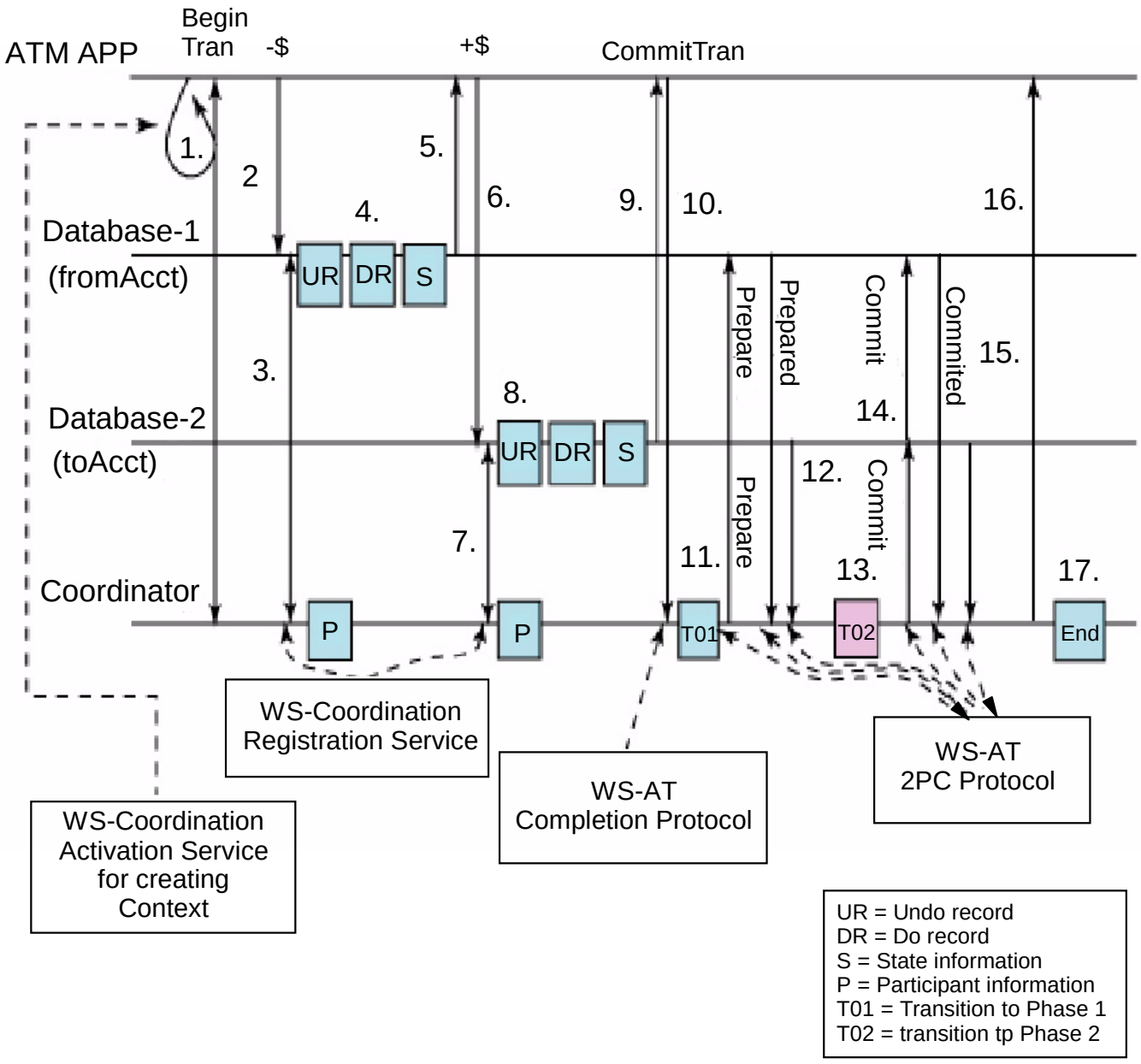
All or none outcome

Classic transaction



UR = Undo record
DR = Do record
S = State information
P = Participant information
T01 = Transition to Phase 1
T02 = transition tp Phase 2

WS-Atomic transaction



WS-Atomic transaction (cont…)

- Web services can be configured to take part in an extended or global unit of work, known as an ***Atomic Transaction***
  - Recoverable updates are not committed or backed out until the Web service is instructed to do so
  - Provision is made to allow updates to be completed manually or automatically if they cannot be successfully coordinated in a reasonable timeframe

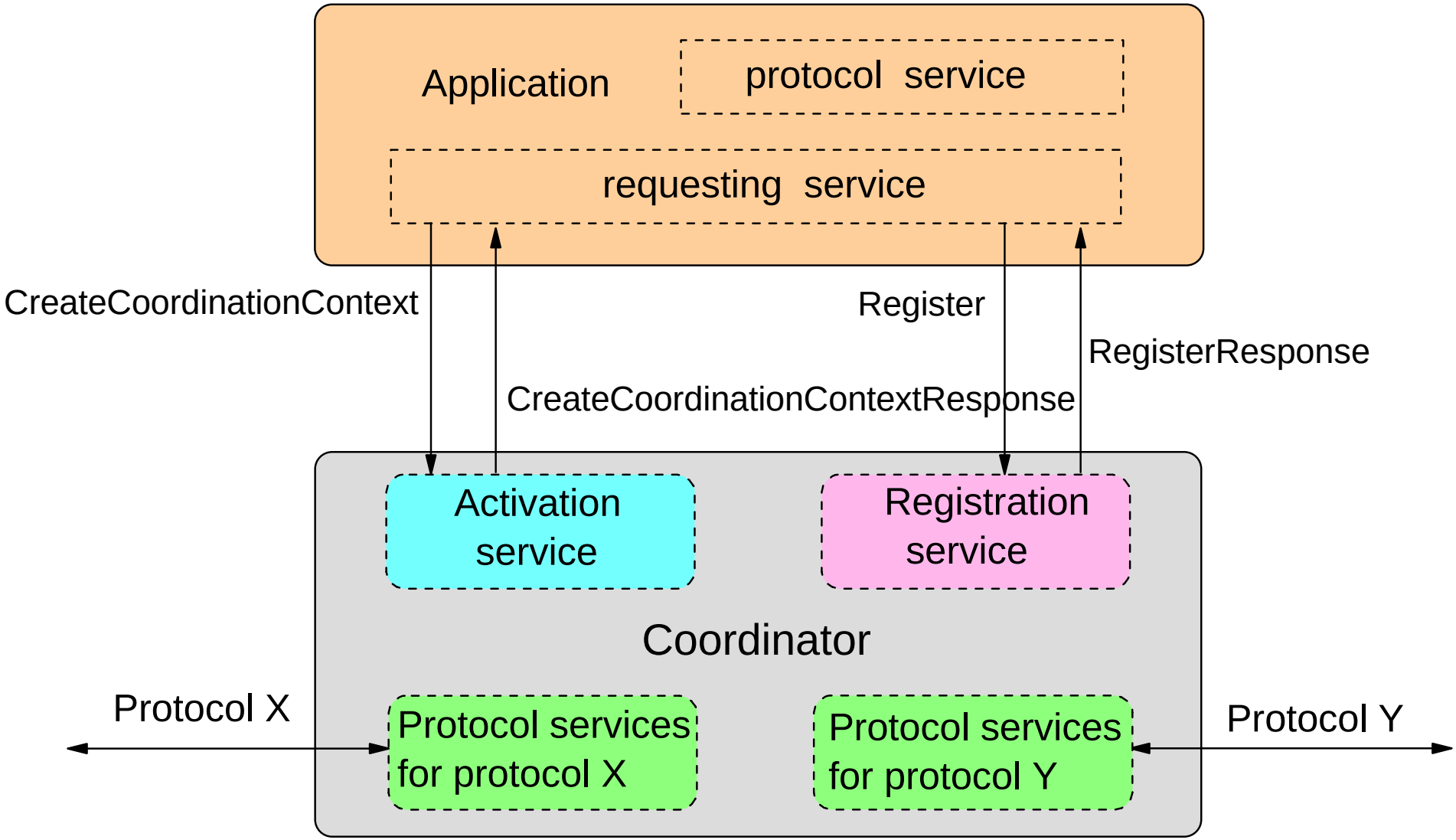
Specifications

- **Web Services Addressing (WS-A)**
  - How to provide information about the destination, where to reply to, and where faults should be sent
- **Web Services Coordination (WS-C)**
  - Define protocols for Web services interoperability
    - Coordination Context
    - Registration Services
- **Web Services Atomic Transactions (WS-AT)**
  - The WS-AT specification builds on WS-C by providing the definition of the atomic transaction coordination type

See Redbook 'Implementing CICS Web Services'

http://www.redbooks.ibm.com/abstracts/sg247206.html?Open

Coordinator services



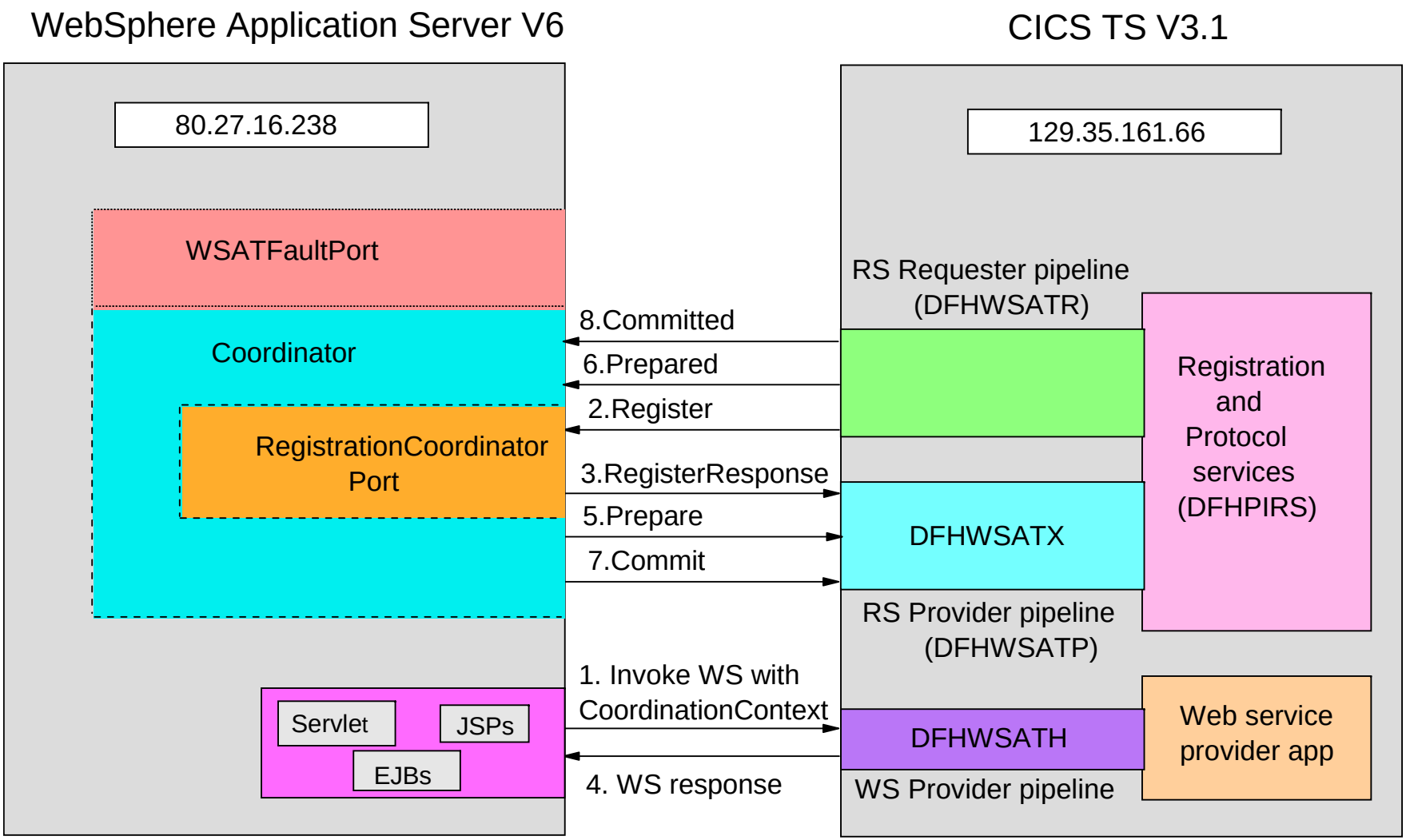
Sample CoordinationContext created by CICS

```
<wscoor:CoordinationContext>
<wscoor:Expires>00004080000</wscoor:Expires>
<wscoor:Identifier>PIAT-CCON-CICSACB2-
        0335969539286 5C</wscoor:Identifier>
<wscoor:CoordinationType>http://schemas.xmlsoap.org/ws/2004/10/wsat
</wscoor:CoordinationType>
<wscoor:RegistrationService>
        <wsa:Address>http://129.35.161.66:13301/cicswsat/Registration
        Service</wsa:Address>
        <wsa:ReferenceProperties>
                <cicswsat:Netname>CICSACB2</cicswsat:Netname>
                <cicswsat:Token>F0F0F0F0</cicswsat:Token>
                <cicswsat:UOWID>BEFA00A71EAACD0E
                </cicswsat:UOWID>
        </wsa:ReferenceProperties>
</wscoor:RegistrationService>
</wscoor:CoordinationContext>
```

Atomic Transaction Services

- **Activation Service**
  - When the application sends a *CreateCoordinationContext* element, the Activation service creates a new activity and returns its coordination context
- **Registration Service**
  - The Registration service defines a **Register** operation that allows a Web service to register to participate in a coordination protocol
- **Protocol Services**
  - A set of coordination Protocol services for each supported coordination type (e.g WS-AT)
  - Control of recoverable updates during task termination
  - Receive and respond to messages requesting that specific actions are carried out before termination
  - Provide support for resync processing

How CICS supports atomic transactions

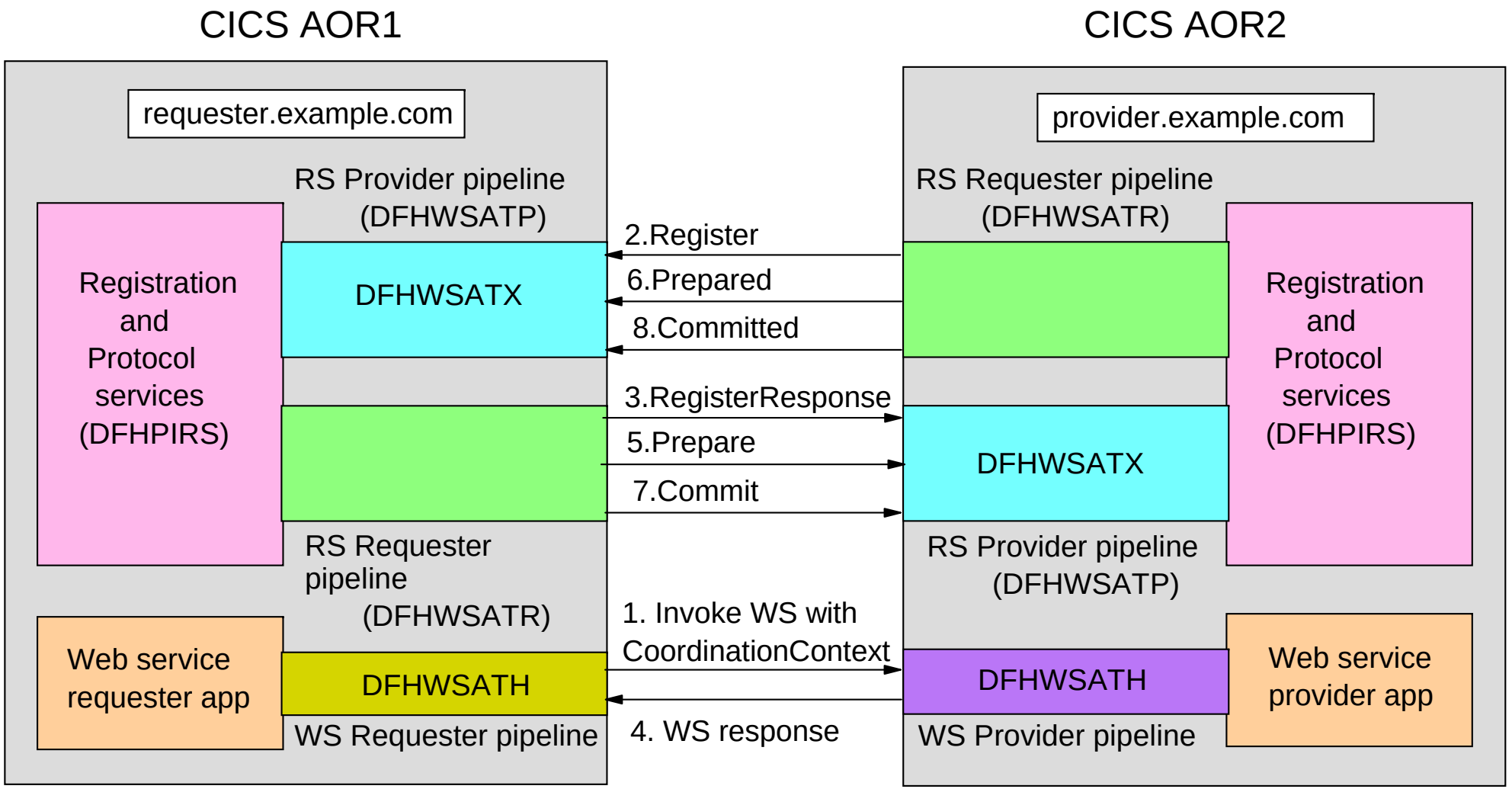


1. Invoke
2. Register

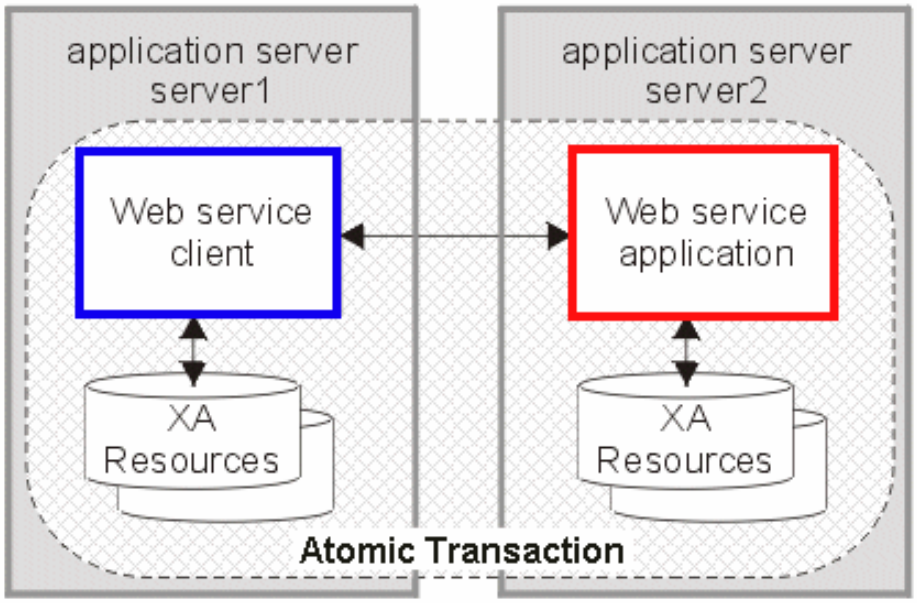
CICS WS-AT pipelines

- Pipeline DFHWSATP
  - Provider pipeline for registration and protocol processing
  - Invokes the CICS-supplied message handler DFHWSATX as the last message handler in the pipeline
- Pipeline DFHWSATR
  - Requester pipeline for registration and protocol processing

CICS as a requester and provider



WAS support for WS-C and WS-AT



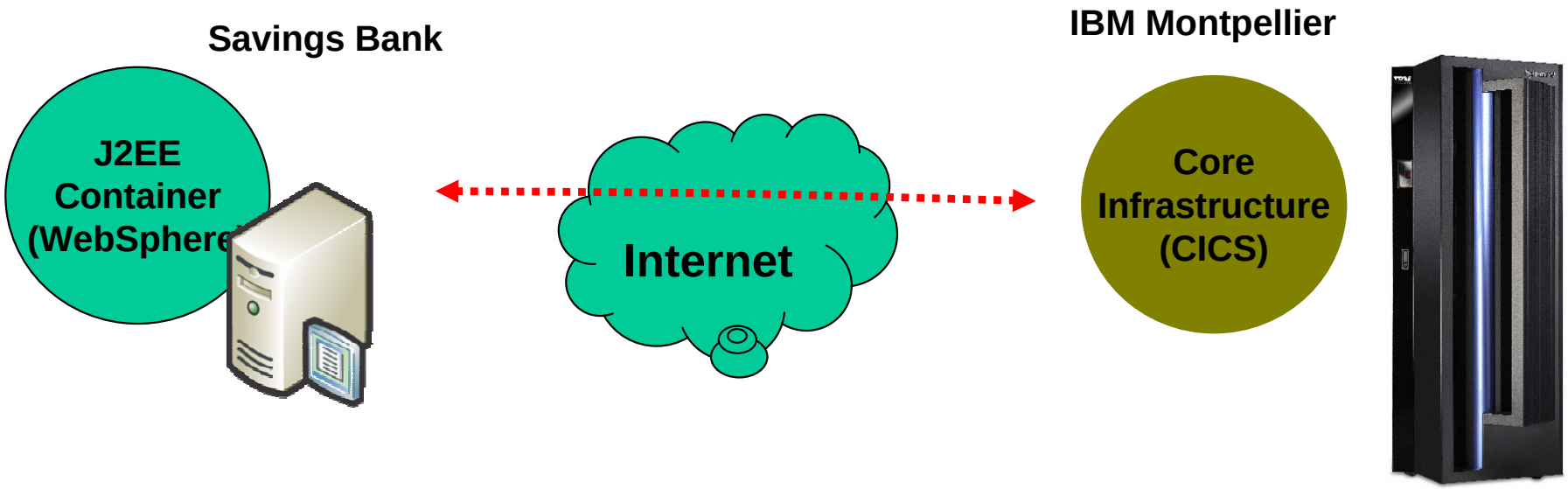
- WAS runtime takes responsibility for the registration of WS-AT participants in the same way as the registration of XA resources
- At transaction completion time all XA resources and WS-AT participants are atomically coordinated by the WAS transaction service
- Configure using application deployment descriptors

Proof of concept

Testing environment

- Due to no availability of CICS 3.1 at customer site the testing has been done with a System z machine provided by IBM Montpellier

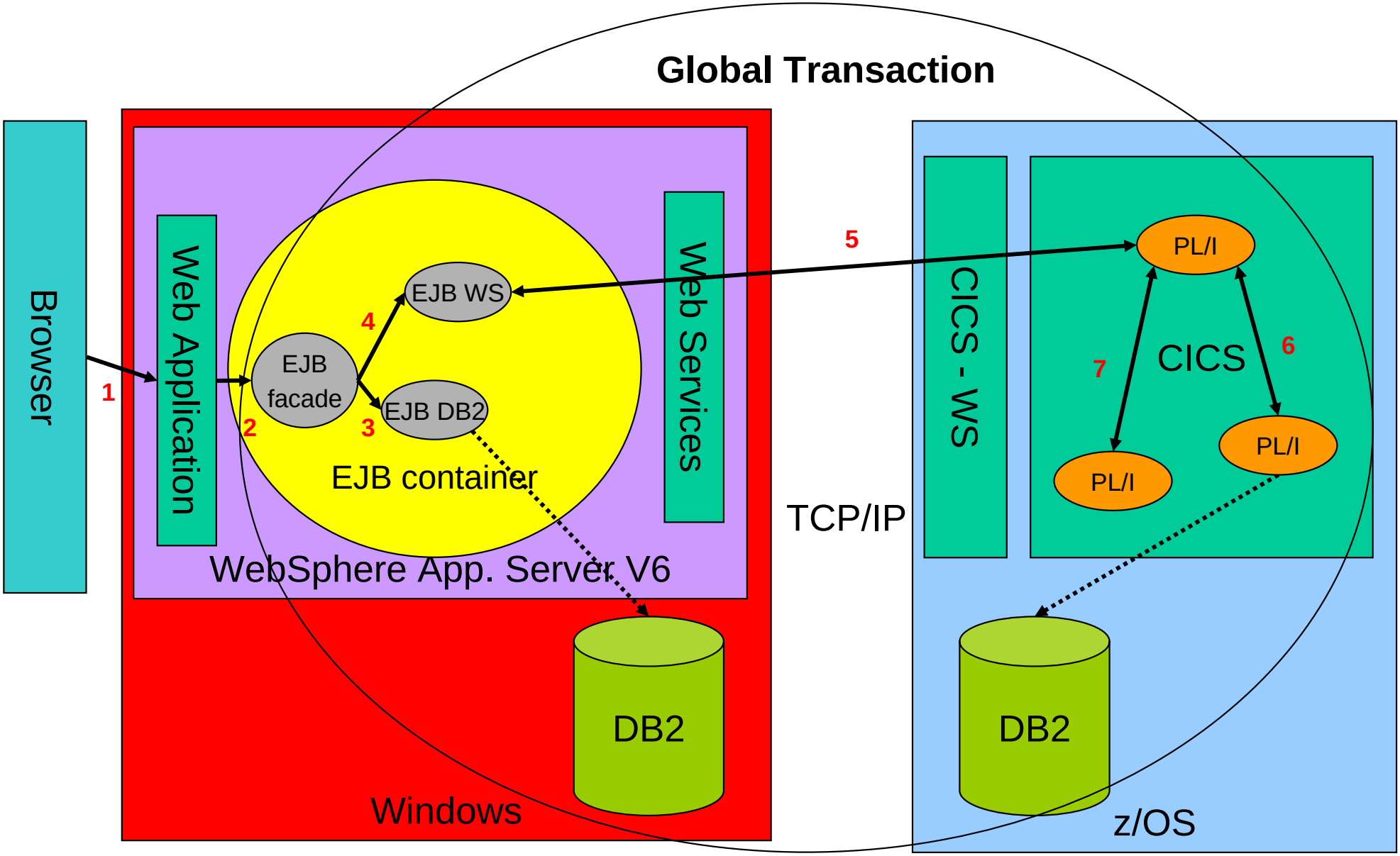
- WAS Windows server runs at customer site



PoC Support

- IBM Montpellier has facilities to do all kinds of testing providing us a complete z/OS stack available from internet
- IBM Design Center have skills to help with PoC

PoC Development - Scenarios



PoC Development

- WebSphere Application Server
  - Session EJBs acting as Web Services requester and provider
  - Session EJBs writing data in DB2
  - Web Application to invoke tests
- CICS – PL/I – DB2
  - PL/I programs acting as Web Services provider (and requester)
  - PL/I transactions write data in DB2 zSeries
- Action in each environment (CICS and WAS) to insert record in DB2 table
  - Errors caused (division by zero) to see how the atomic transaction is rolled back

CICS Web Services tooling

- Used tools DFHLS2WS and DFHWS2LS to generate the necessary files to convert XML data (described in the WSDL) to language structures (PL/I) and vice-versa
  - DFHLS2WS: generate WSDL and WSBind from language structure
  - DFHWS2LS: generate LS and WSBind from WSDL

- To invoke a Web Service from CICS:
  - EXEC CICS INVOKE WEBSERVICE ( ) CHANNEL ( ) URI ( ) OPERATION ( )

Software Maintenance

- Required CICS TS V3.1 maintenance :
  - PTF UK12958/APAR PK20166
  - PTF UK11916/APAR PK19530

    INVREQ on EXEC CICS INVOKE WEB SERVICE
  - PTF UK13664/APAR PK22600

    CICS transaction abend if service provider updates DB2

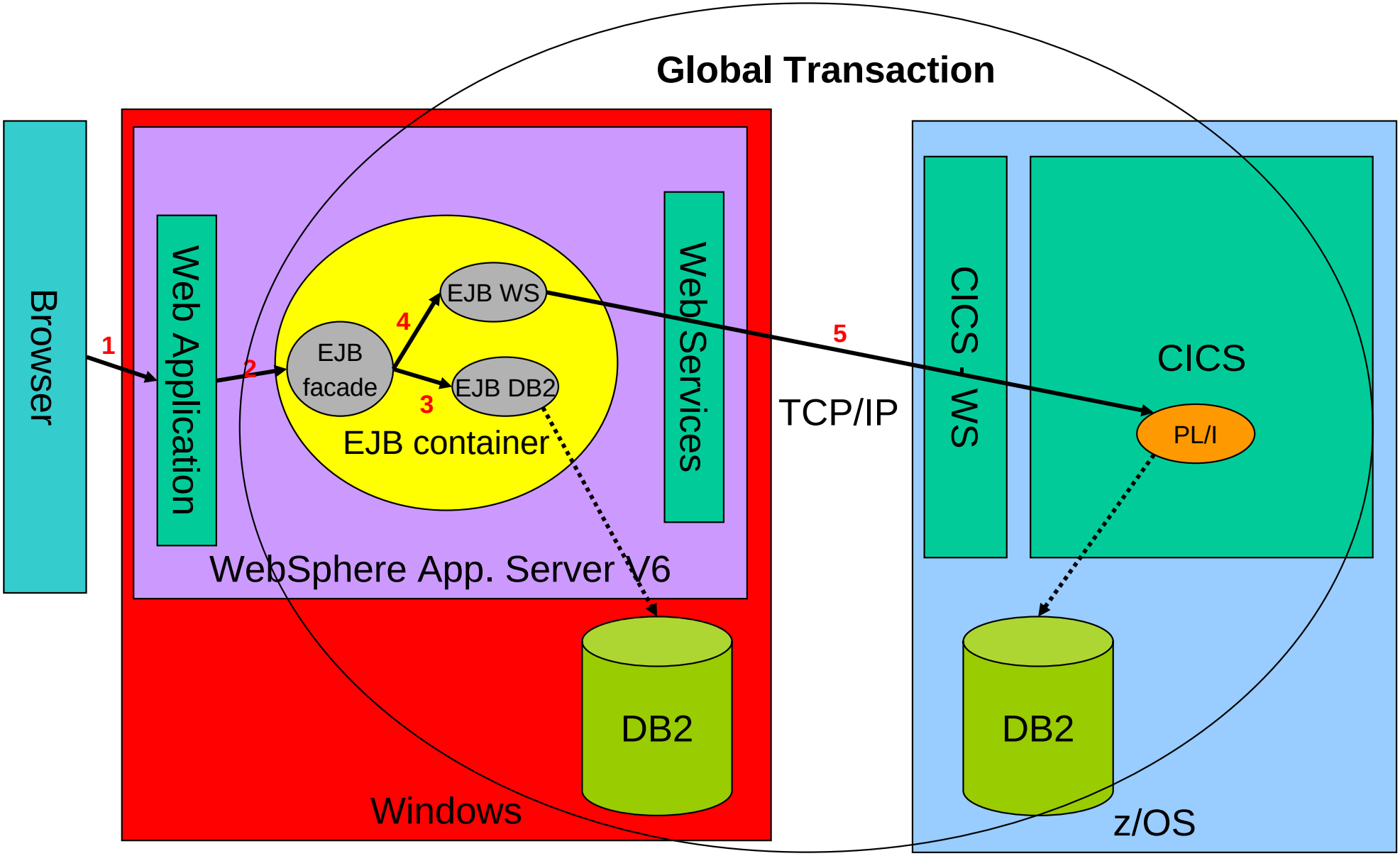
- WAS maintenance level, require version over 6.0.2.7 (we used 6.0.2.9)
  - Includes fix for APAR PK16509

    WebSphere sends SOAP actor role in CoordinationContext which is rejected by CICS

Test scenarios

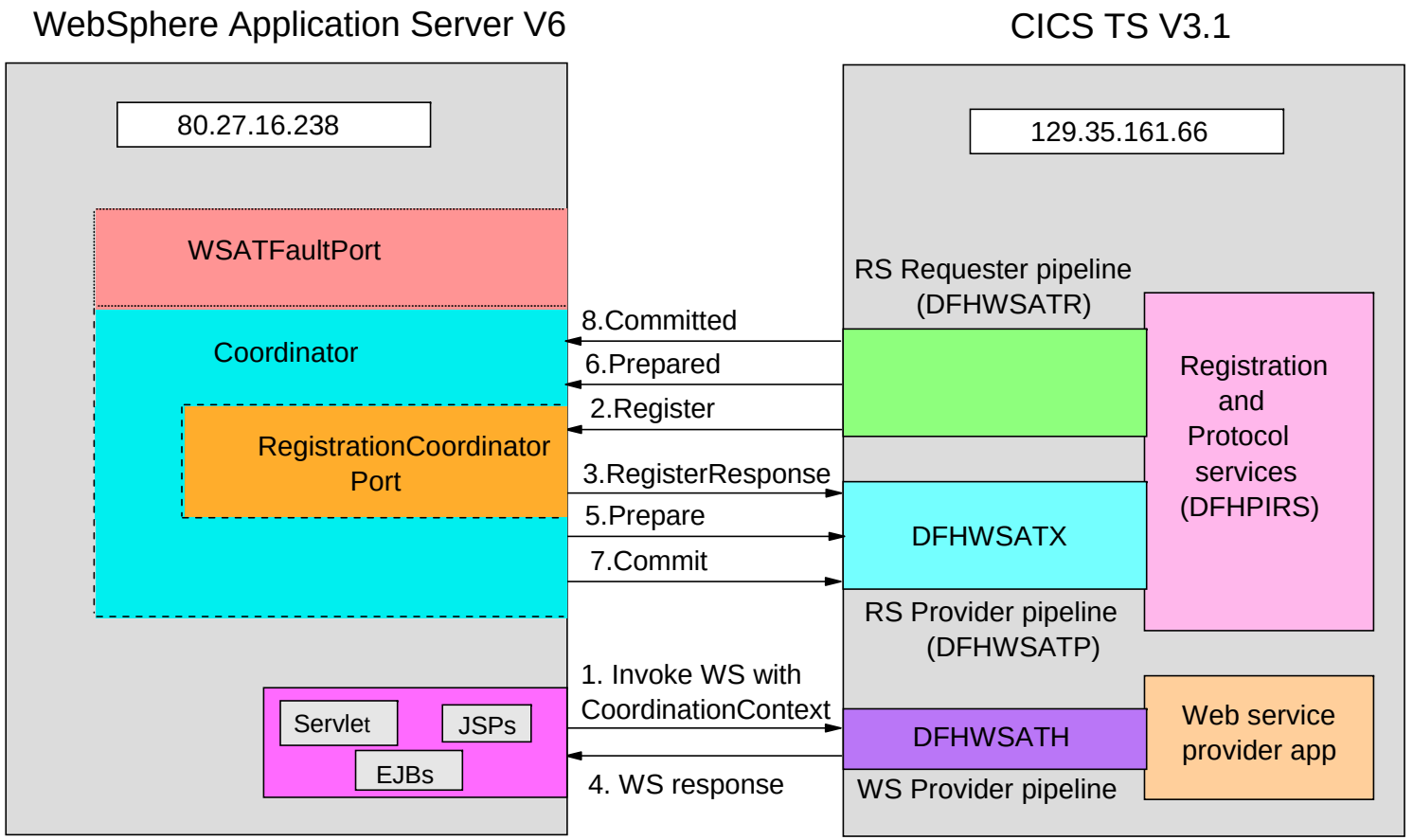
- Scenario 1: WAS -> CICS
- Scenario 2: WAS -> CICS -> WAS
- Scenario 3: CICS -> WAS
- Scenario 4: CICS -> WAS -> CICS

Scenario 1: WAS - CICS



WS-AT protocol between servers: Commit

1. Invoke
2. Register
3. RegisterResponse
4. Response
5. Prepare
6. Prepared
7. Commit
8. Commited



Configuring CICS for WS-AT

- Install group **DFHWSAT** which contains the CICS resource definitions needed
  Note: this group is not part of DFHLIST.

- The HFS locations for the pipeline files for the **DFHWSATR** and **DFHWSATP** pipelines may need to be changed if they have not been installed in the default locations
  - Copy the contents of group DFHWSAT to another group
  - Change the CONFIGFILE attribute in the PIPELINE resources

- Update service provider pipeline for (optional) invocation of **DFHWSATH** header processing program whenever the SOAP message contains a CoordinationContext header

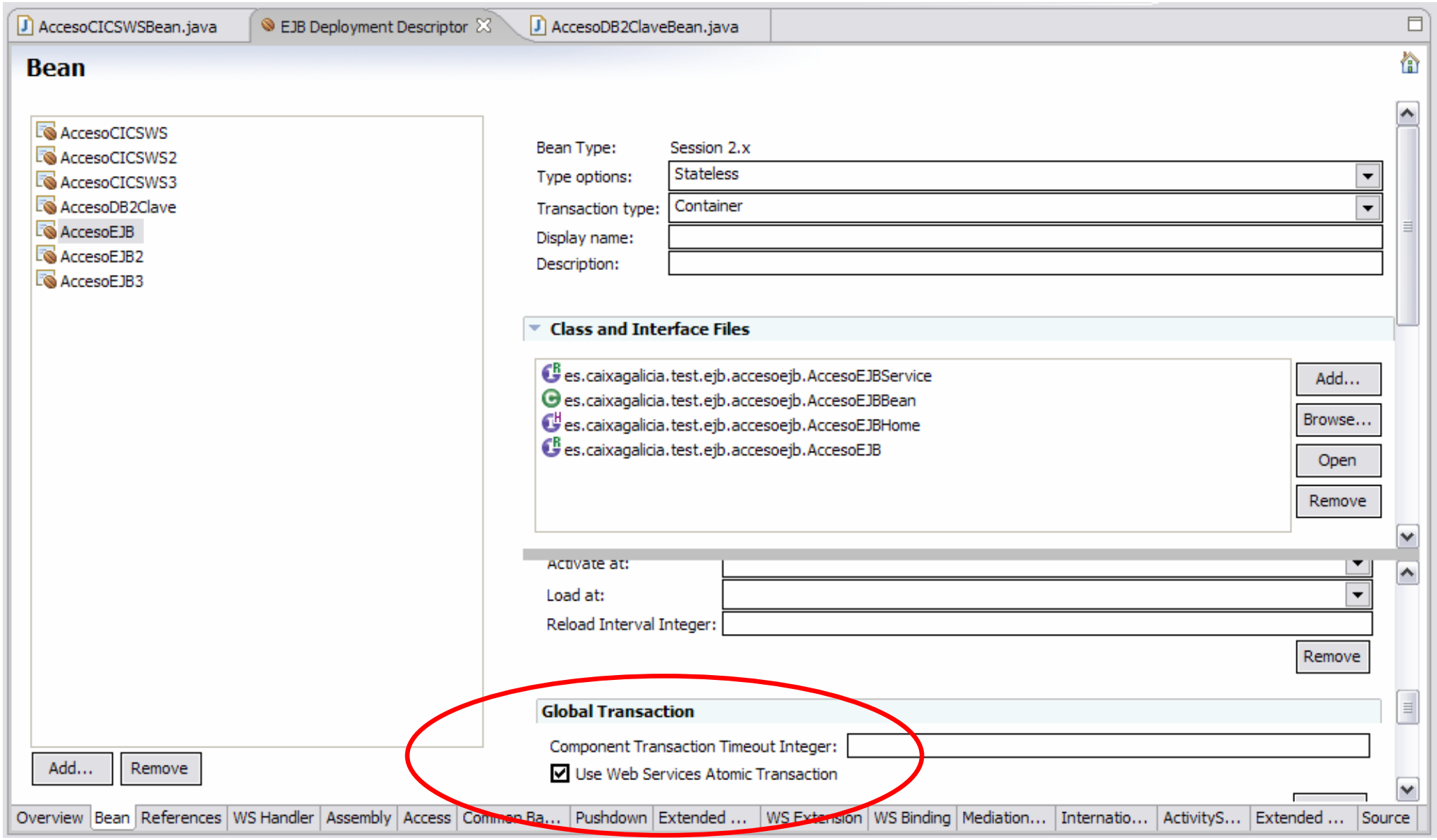
Service provider pipeline config file

```
<service>
  <terminal_handler>
    <cics_soap_1.1_handler>
      <headerprogram>
        <program_name>DFHWSATH</program_name>
        <namespace>
          http://schemas.xmlsoap.org/ws/2004/10/wscoor
        </namespace>
        <localname>CoordinationContext</localname>
        <mandatory>false</mandatory>
      </headerprogram>
    </cics_soap_1.1_handler>
  </terminal_handler>
</service>
<apphandler>DFHPITP</apphandler>
<service_parameter_list>
  <registration_service_endpoint>
  http://129.35.161.66:13301/cicswsat/RegistrationService
  </registration_service_endpoint>
</service_parameter_list>
```

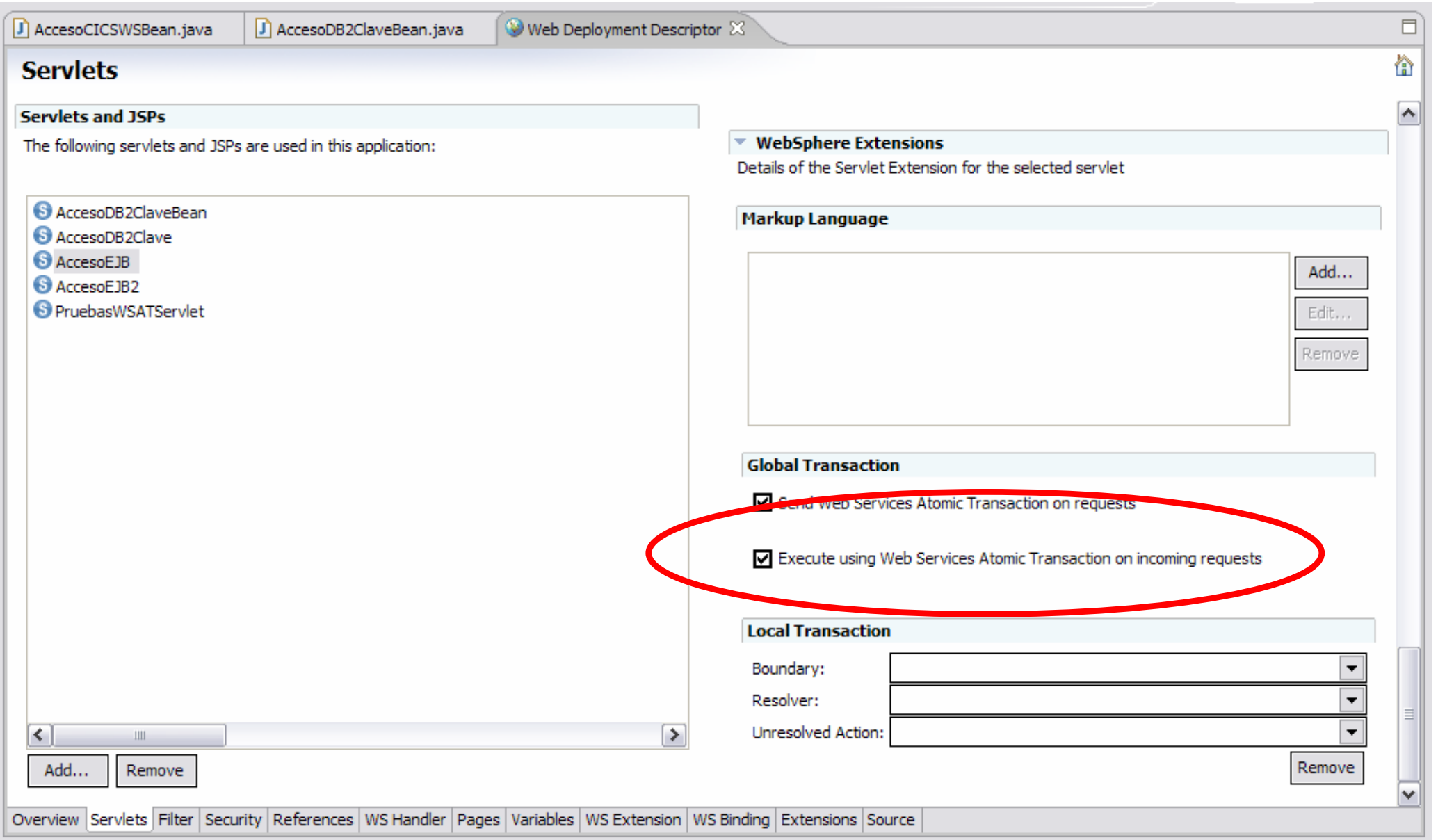
Configuring WAS for WS-AT

- In a Web module that invokes a Web service, specify ***Send Web Services Atomic Transaction Context*** *on outbound requests*
- In a Web module that implements a Web service, specify ***Execute using Web Services Atomic Transaction*** *on incoming requests*
- In an EJB module that invokes a Web service, specify ***Use Web Services Atomic Transaction*** to propagate the EJB transaction to the target Web service
- In an EJB module, bean methods must be specified with transaction type **Required**, which is the default, to participate in a global atomic transaction

WAS app deploy descriptor for service requester

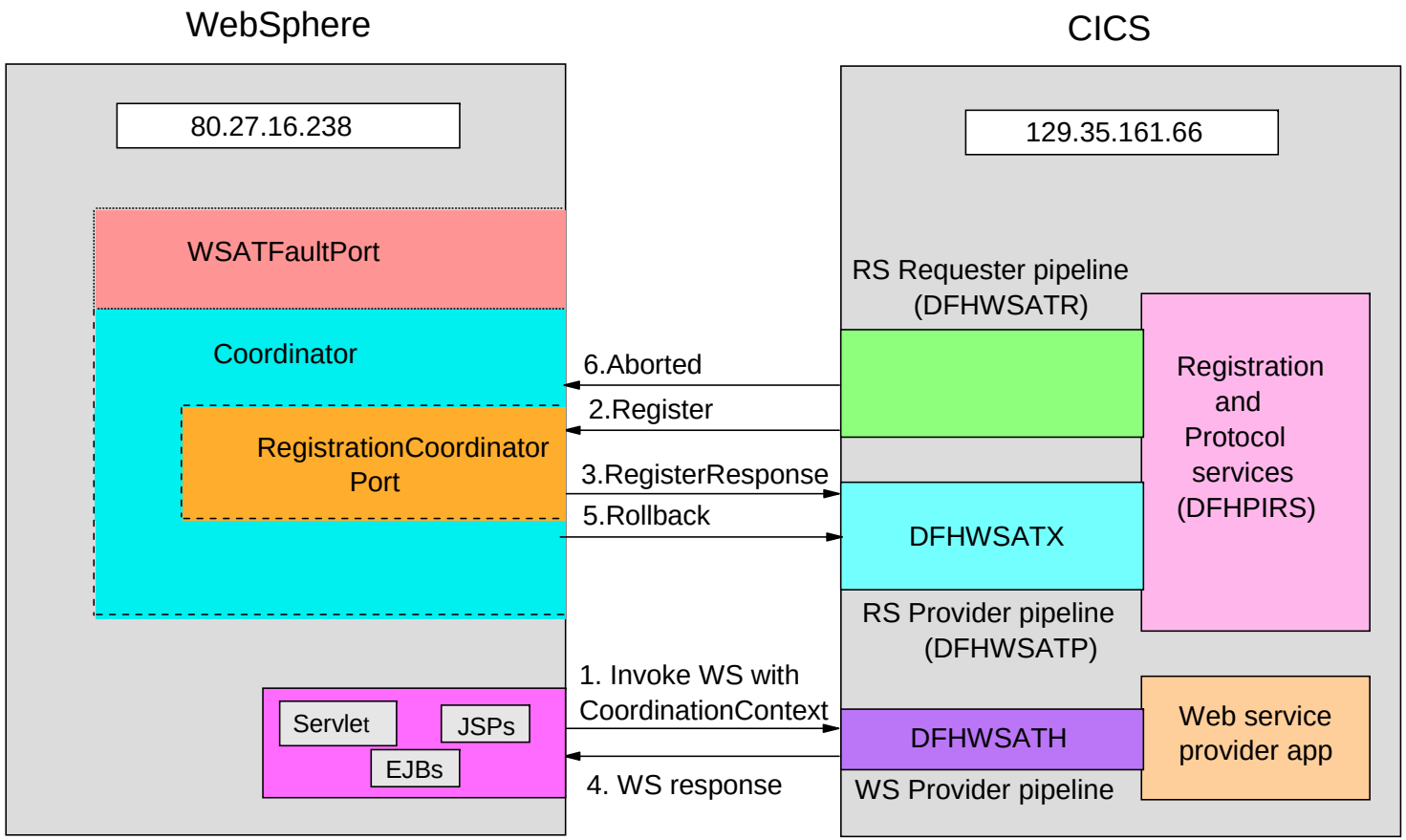


WAS app deploy descriptor for service provider

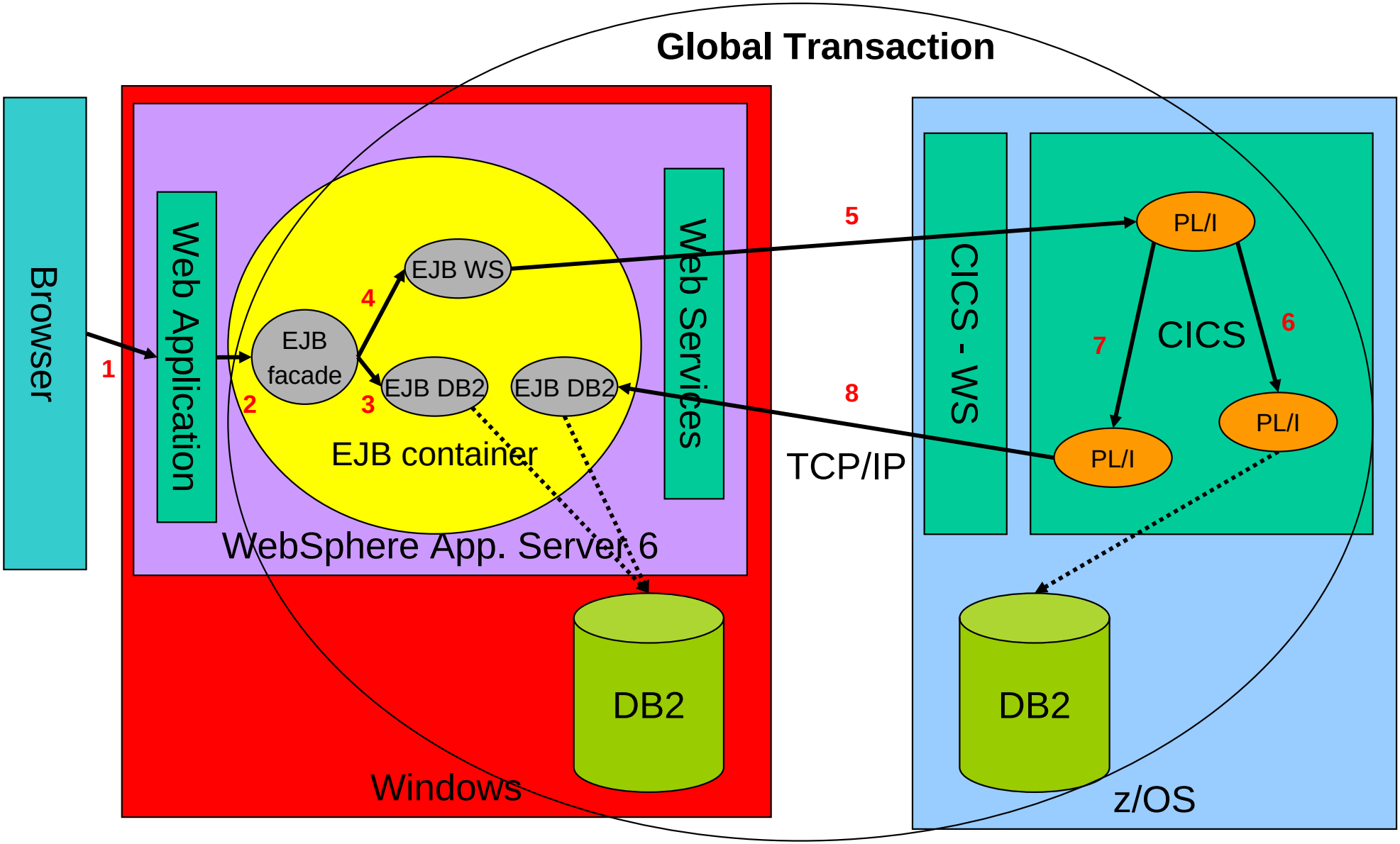


WS-AT protocol between servers: Rollback

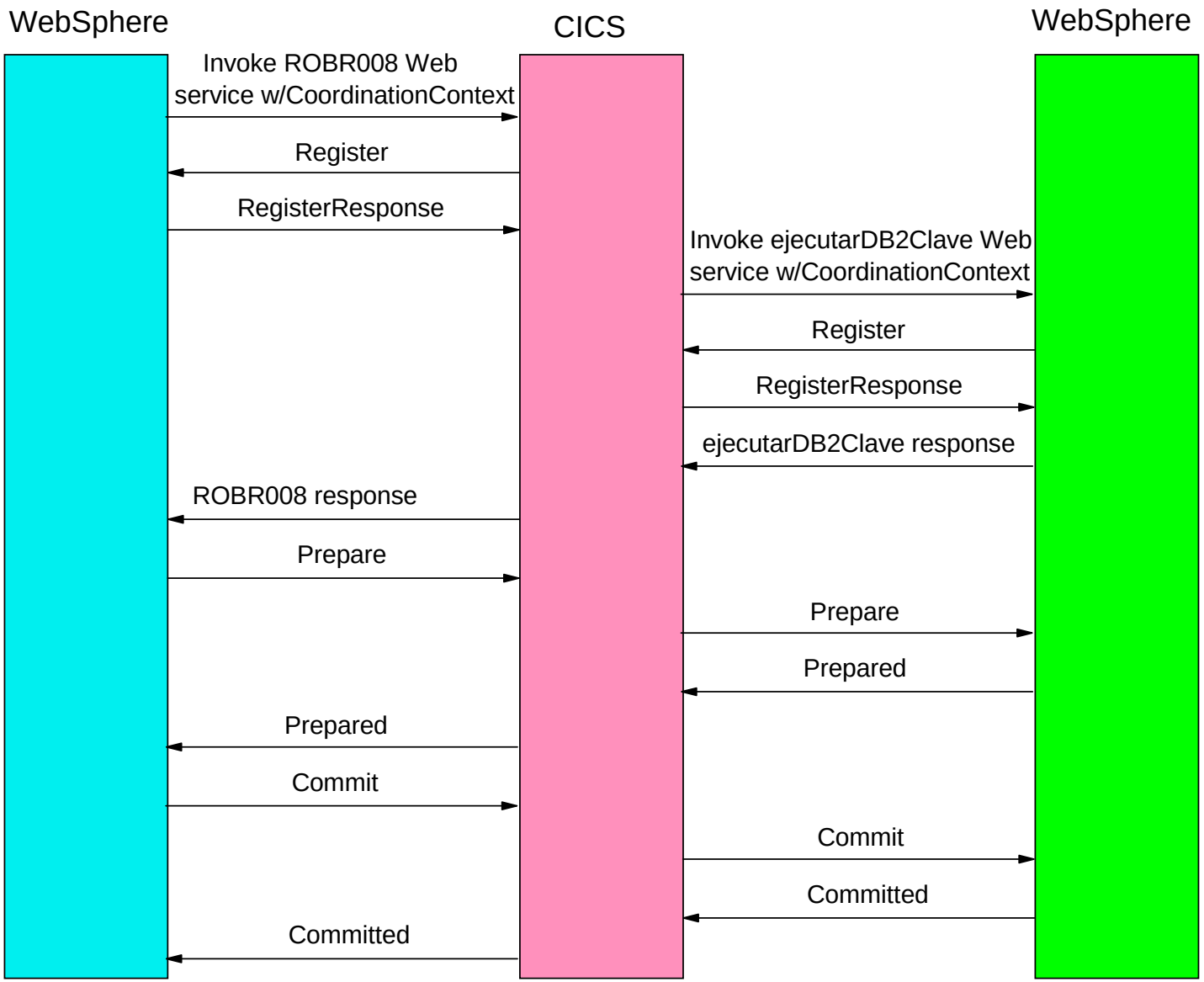
1. Invoke
2. Register
3. RegisterResponse
4. Response
5. Rollback
6. Aborted



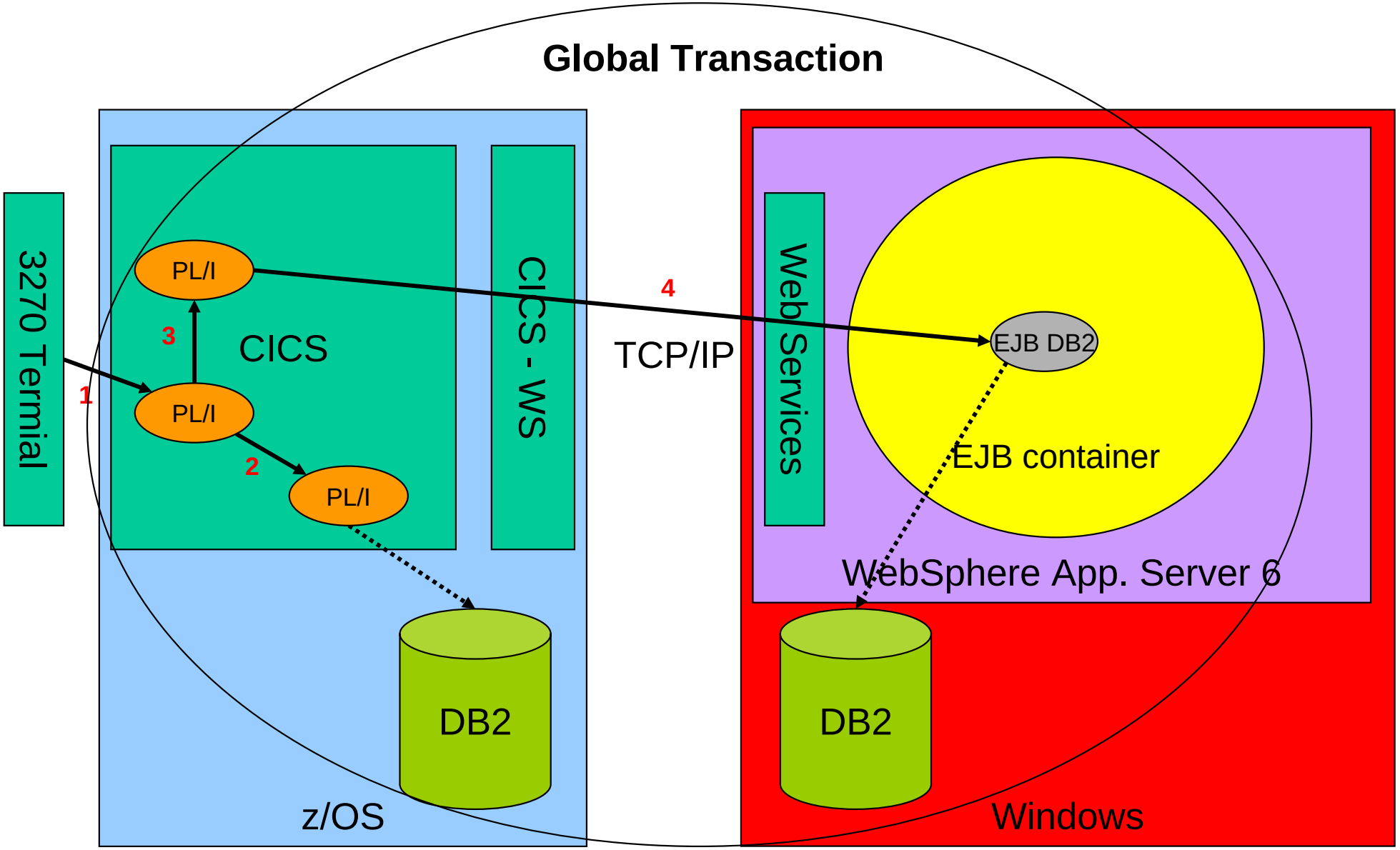
Scenario 2: WAS - CICS - WAS



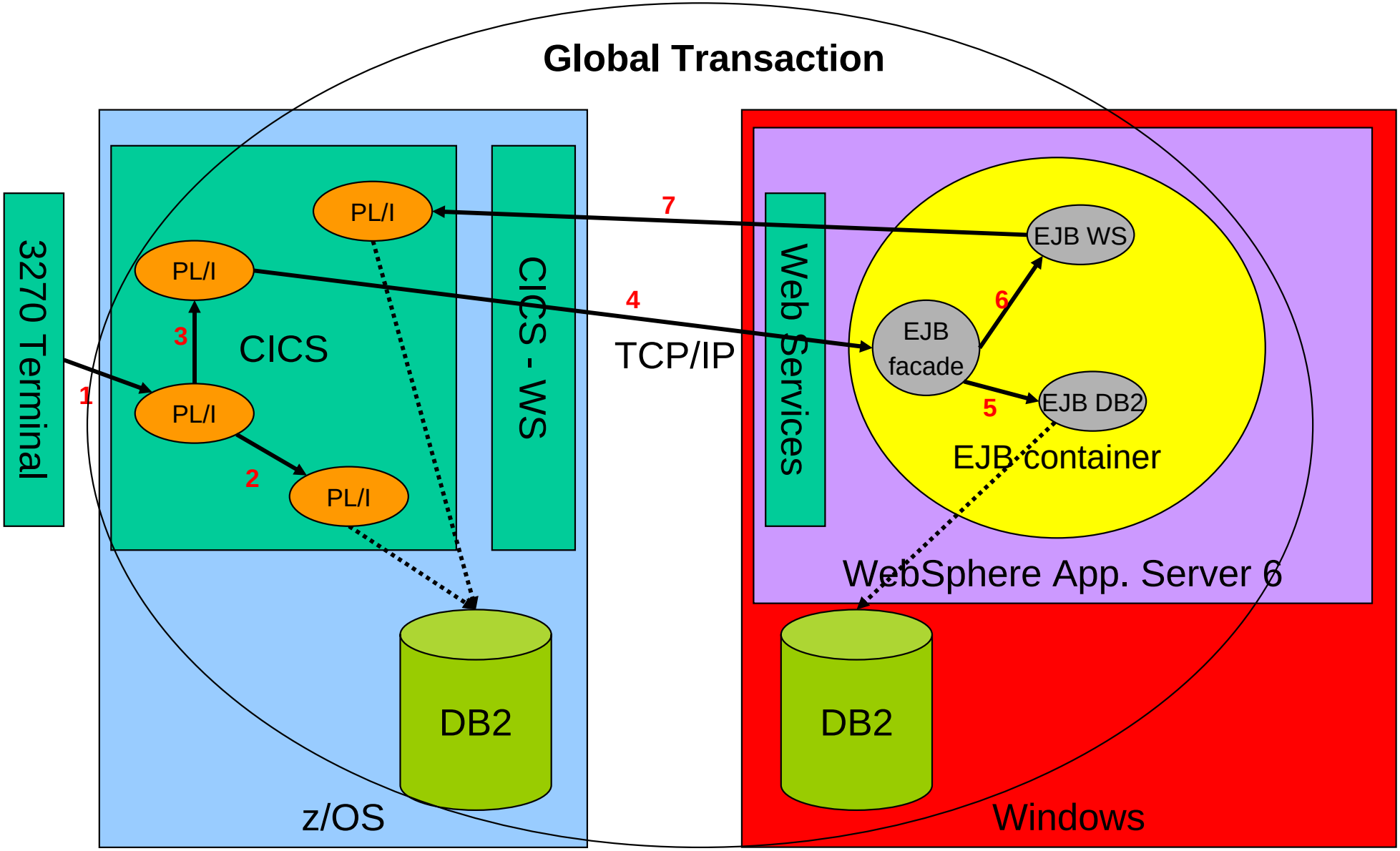
Scenario 2 flows



Scenario 3: CICS - WAS



Scenario 4: CICS - WAS - CICS



Lessons learnt

- Useful detailed scenarios in Redbook ‘Implementing CICS Web Services’
  http://www.redbooks.ibm.com/abstracts/sg247206.html?Open
- Recommend to use latest maintenance levels of CICS and WAS
- In WAS V6.1 improved support for WS-AT (e.g you can define the coordinator endpoint, in WAS V6 system IP of WAS is used)
- Use of a network sniffer (e.g TCP/IP monitor) to monitor the SOAP messages
- Web services support in CICS TS 3.1 easier to use than expected
- WS-AT is supported in IBM GA products, not betas

Conclusions

- WS-AT is a two-phase-commit protocol for short life transactions
- WS-AT should only be considered for transactions inside the company
- Inter-enterprise transactions typically require a looser semantic than 2PC
- PoC results confirm that 2PC is possible between WAS and CICS bi-directionally
- If single direction only is required first consideration should be CICS TG (V6.1 and V7)

Further Information

ITSO Redbooks

- “Implementing CICS Web Services” (SG24-7206-**2**)
- “Web Services Handbook for WebSphere Application Server 6.1” (SG247257)

Information Centers

- CICS

http://publib.boulder.ibm.com/infocenter/cicsts/v3r1/index.jsp

- WebSphere

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp

IBM 2007 IMPACT

IBM

Questions
and
Answers

impact-venture*

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, 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.