Reliable and Transacted Web Services Between Sun's "Project Tango" and Microsoft Indigo

Mike Grogan, Joe Fialli, Ryan Shoemaker

Sun Microsystems, Inc.

TS-1603

Goal of the Talk

Learn how to build reliable distributed systems composed of web services spanning Java™ platform and Microsoft platform

Agenda

Project Tango

Challenges in a Distributed System

Reliable Messaging

Demo

Distributed Transaction

Project Tango

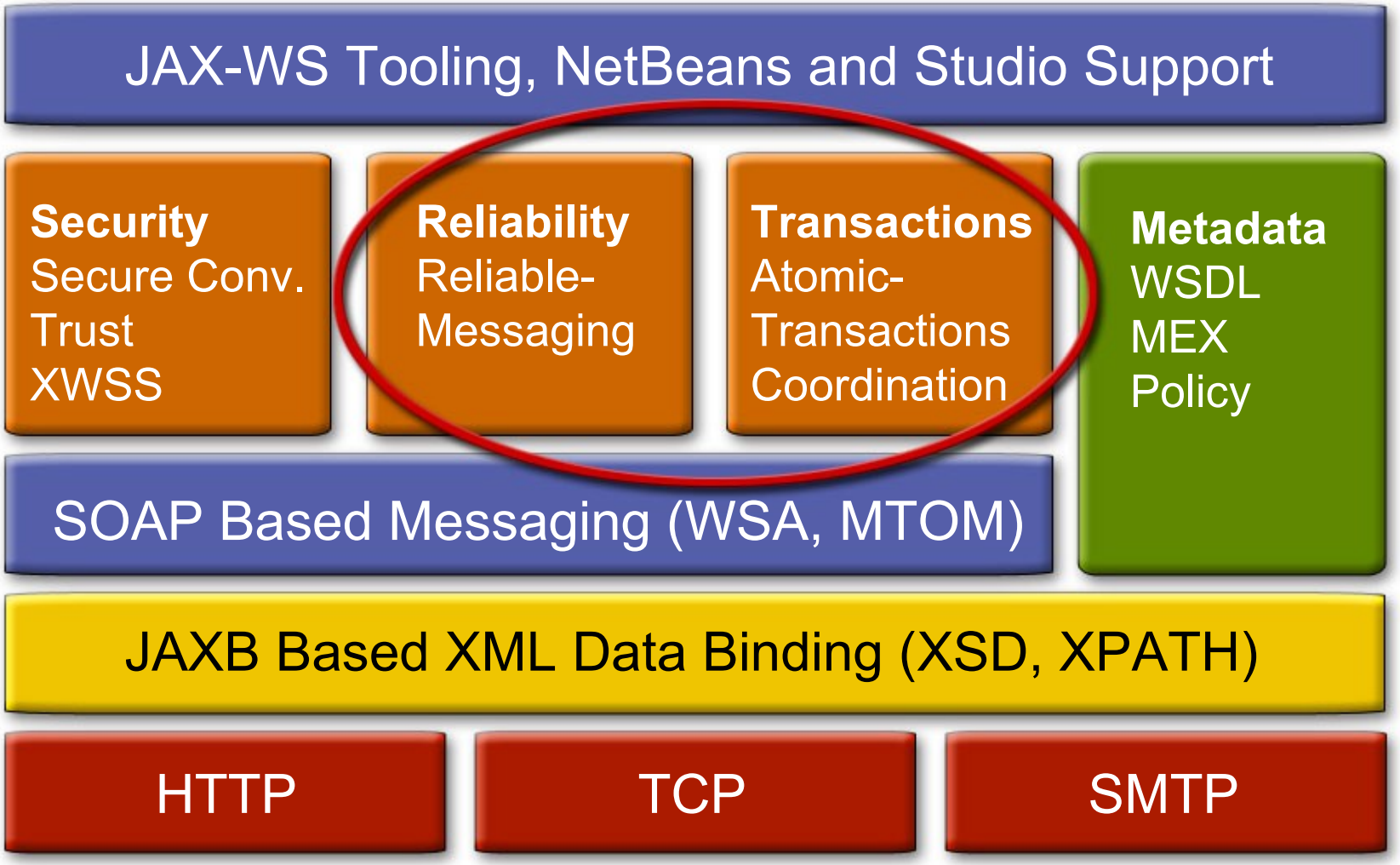
Background

- Goal
  - Deliver next generation Web Services technologies enabling first class interoperability between Sun's Java Products and Windows Operating environments supporting WCF[1]
- Implementation strategy
  - Build on Java API for XML Web Services (JAX-WS) and Java™ Architecture for XML Binding (JAXB) technologies
  - Work closely with Microsoft and perform product level testing
  - Build an active Open Source community centered around the Project GlassFish community

[1] Windows Communications Foundation a.k.a. "Indigo"

Web Services Stack

Quality of Services (QoS): Reliability, Transactions



Agenda

Project Tango

Challenges in a Distributed System

Reliable Messaging

Demo

Distributed Transaction

Challenges in Distributed Systems

- Communication issues
  - Network unavailable or connection dropped
  - Messages are lost
  - Related messages arrive out-of-order
- Processing issues
  - Messages can not be retried without side effects
  - Hardware failures result in lost data
  - Failures can leave system in inconsistent state

Agenda

Project Tango

Challenges in a Distributed System

Reliable Messaging

Demo

Distributed Transaction

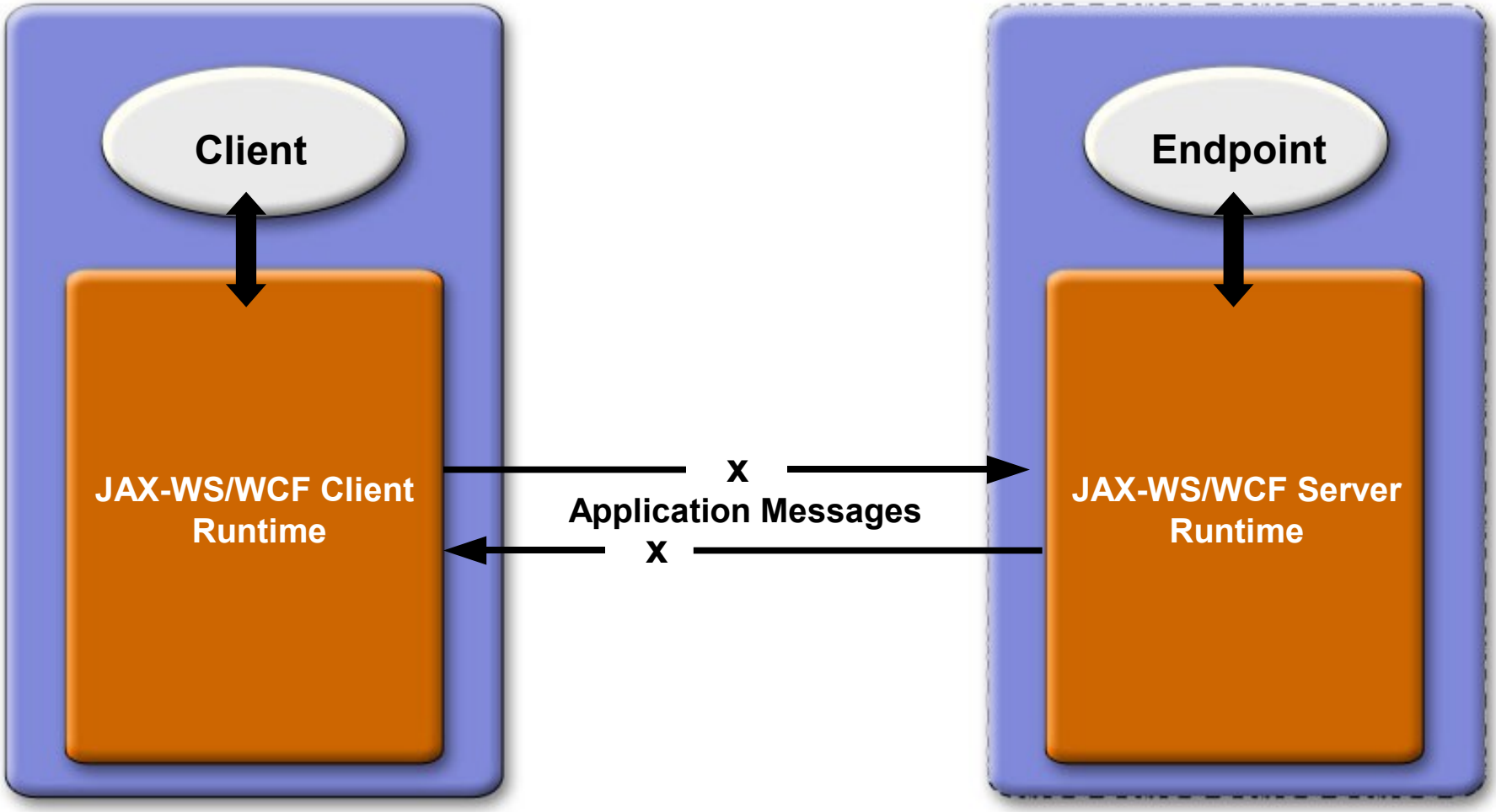
Reliable Messaging

- Uses open-standard SOAP-based protocol
- TCP-inspired
  - Sender and receiver can reconstruct stream of messages in the exact order they were sent
- “Delivery assurances”
  - At least once
  - At most once
  - In order

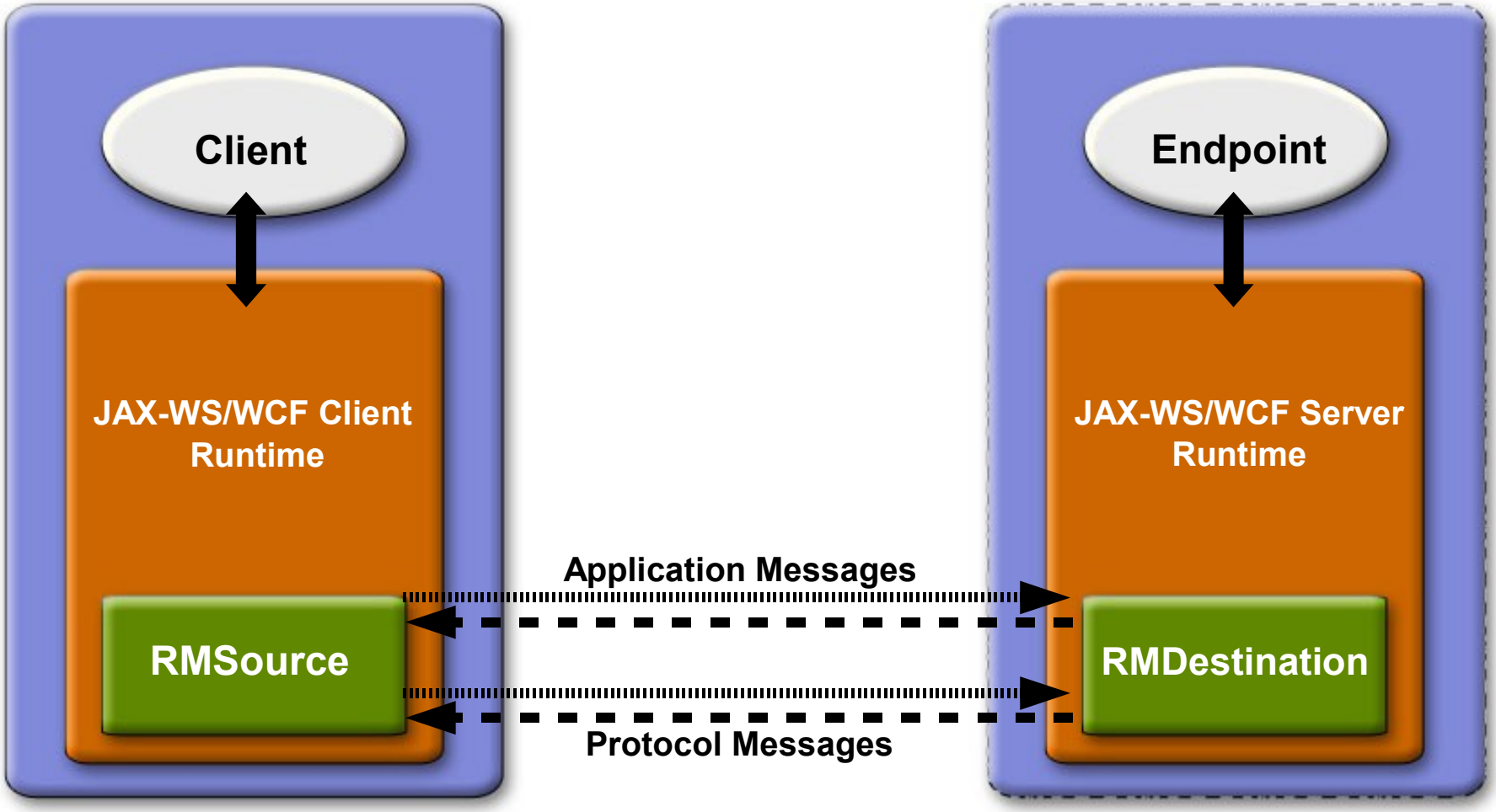
Standards: Web Service Reliable Messaging

- Add WS-ReliableMessaging support to Java EE platform
  - http://specs.xmlsoap.org/ws/2005/02/rm/ws-reliablemessaging.pdf
  - http://specs.xmlsoap.org/ws/2005/02/rm/WS-RMPolicy.pdf
- Support OASIS WS-RX when completed
  - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ws-rx

Messaging System Without RM



Reliable Messaging System



Reliable Messaging Trade-offs

- Benefits
  - Communication failure recovery
  - Helps manage stateful endpoints
- Disadvantages/limitations
  - Processing cost
  - Non-portability
  - No processing failure protection
    - Non-persistent
    - Cannot restore system to original state

Communication Failures

How Does RM Help?

- At least once
  - Are there one-way messages?
    - Provides way to ensure one-way message have arrived
- At most once
  - Does processing duplicate messages have harmful side-effects?
    - account.debit(100000);
- In order
  - Is ordered delivery important?
    - For stateless endpoints, maybe not

How to Enable Reliable Messaging

- Endpoint
  - Use NetBeans to enable and configure
- Client
  - No choice—depends on Policy Assertions in Endpoint's WSDL

Reliable Messaging Programming Model

- Almost completely invisible
- Client
  - proxy.close()
    - Returns after all messages have been Acknowledged
    - Tells Endpoint no more messages are coming
- Endpoint
  - **com.sun.xml.ws.sessionid** WebServiceContext property
    - Uniquely identifies client instance
    - Use to maintain state per client

Agenda

Project Tango

Challenges in a Distributed System

Reliable Messaging

Demo

Distributed Transaction

DEMO

Reliable Messaging

http://weblogs.java.net/blog/bhaktimehta/

Agenda

Project Tango

Challenges in a Distributed System

Reliable Messaging

Demo

Distributed Transaction

Distributed Transactions

- Enable transactions flowing over Web Services

Atomic

Consistent

Isolated

Durable

- Provide fault tolerance across a heterogeneous system

Adding Web Service Transaction Support

Relation to Existing Standards

- WS-Coordination specification defines
  - Wire protocol for distributed coordinated activity
  - Participant registration protocol with activity
- WS-Atomic Transaction specification defines
  - Policy assertions
  - Coordinated protocols:
    - Completion and 2 Phase Commit (2PC)
- Support OASIS WS-TX when its completed
- Expose transaction manager in Sun Java System Application Server as WS-AT specified web services

No New Java-based APIs Necessary!*

* Some Configuration Required.

Enabling WS-Atomic Transactions

Programming Model

- Transacted Web Service
  - Starting from Java Source
    - Stateless EJB™ using Container Managed Transaction (CMT)
  - Starting from WSDL
    - Transacted operations denoted with WS-AT Policy Assertion
- Web Service Consumer (Client)
  - Create a transaction and demarcate its boundaries
    - `javax.transaction.UserTransaction` in Web or EJB tier
    - CMT in EJB tier
  - Invoke transacted web method(s) in transaction scope

Enabling WS-Atomic Transactions

Programming Model (Cont.)

- Configuration of WS-AT coordination service
  - Via Application Server's admin console
    - Application Server → Configuration → Transaction Services
  - New properties needed
    - Configure for secure transaction protocol between Sun and Microsoft WS-AT coordination services

Implementing Transacted Web Service

Composition of Java EE and Web Service Annotations

```
@javax.jws.WebService
@javax.ejb.Stateless
@javax.ejb.TransactionManagement(CONTAINER) [1]

public class Bank {

   @javax.jws.WebMethod
   @javax.ejb.TransactionAttribute(REQUIRED) [1][2]
   void transferFunds(...) throws ... ;
}
```

[1] stateless EJB default, annotation added to be explicit
[2] Implementation restriction:
transaction not propagated with a one way message

Defining a Transacted Web Service

```
<wsdl:definitions>

    <!-- Define WS-AT policy assertion -->
    <wsp:Policy wsu:Id="TransactedPolicy1" >
       <wsat:ATAssertion wsp:Optional="true"/>
       <wsat:ATAlwaysCapability/>
    </wsp:Policy>

  <wsdl:binding name="Bank" type="tns:BankPortType" >

      <!-- Marked Transacted by Operation Policy Subject-->
      <wsdl:operation name="transferFunds" >
        <wsp:PolicyReference URI="#TransactedPolicy1"
                             wsdl:required="true" />
        ...
      </wsdl:operation>
  </wsdl:binding>
</wsdl:definitions>
```

Web Service Client

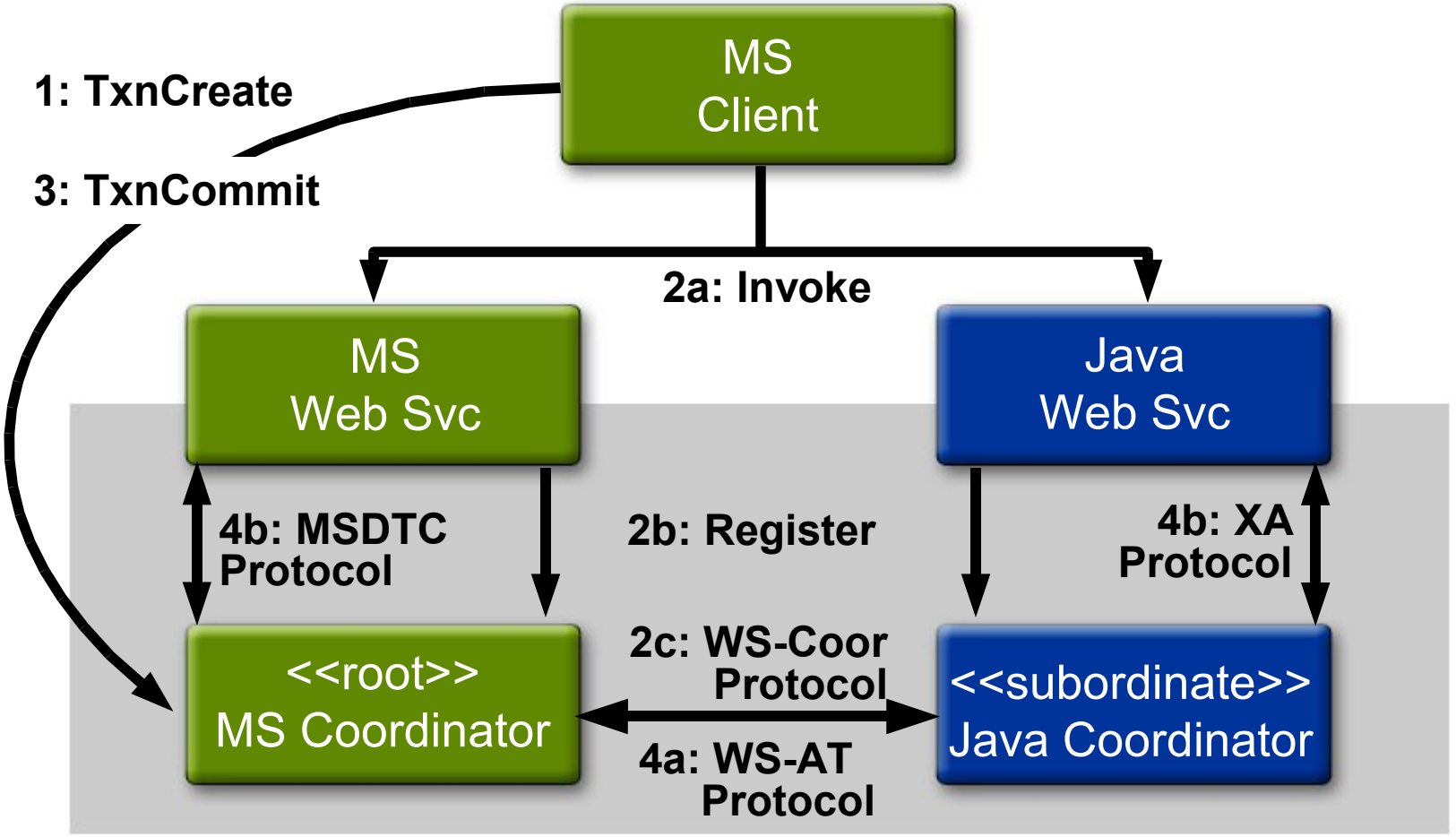
```
import javax.annotation.Resource;

public class ATMClient {
  @javax.jws.WebServiceRef
  static BankService service;

  public void selectedTransferFunds() {
     @Resource javax.transaction.UserTransaction ut;

     Bank bank = service.getBank();
     ut.begin();
         bank.transferFunds( ... );
         // call other transacted ejb/web methods
     ut.end();
  }
}
```

Mapping Between Java EE Transaction and WS-Atomic Transaction

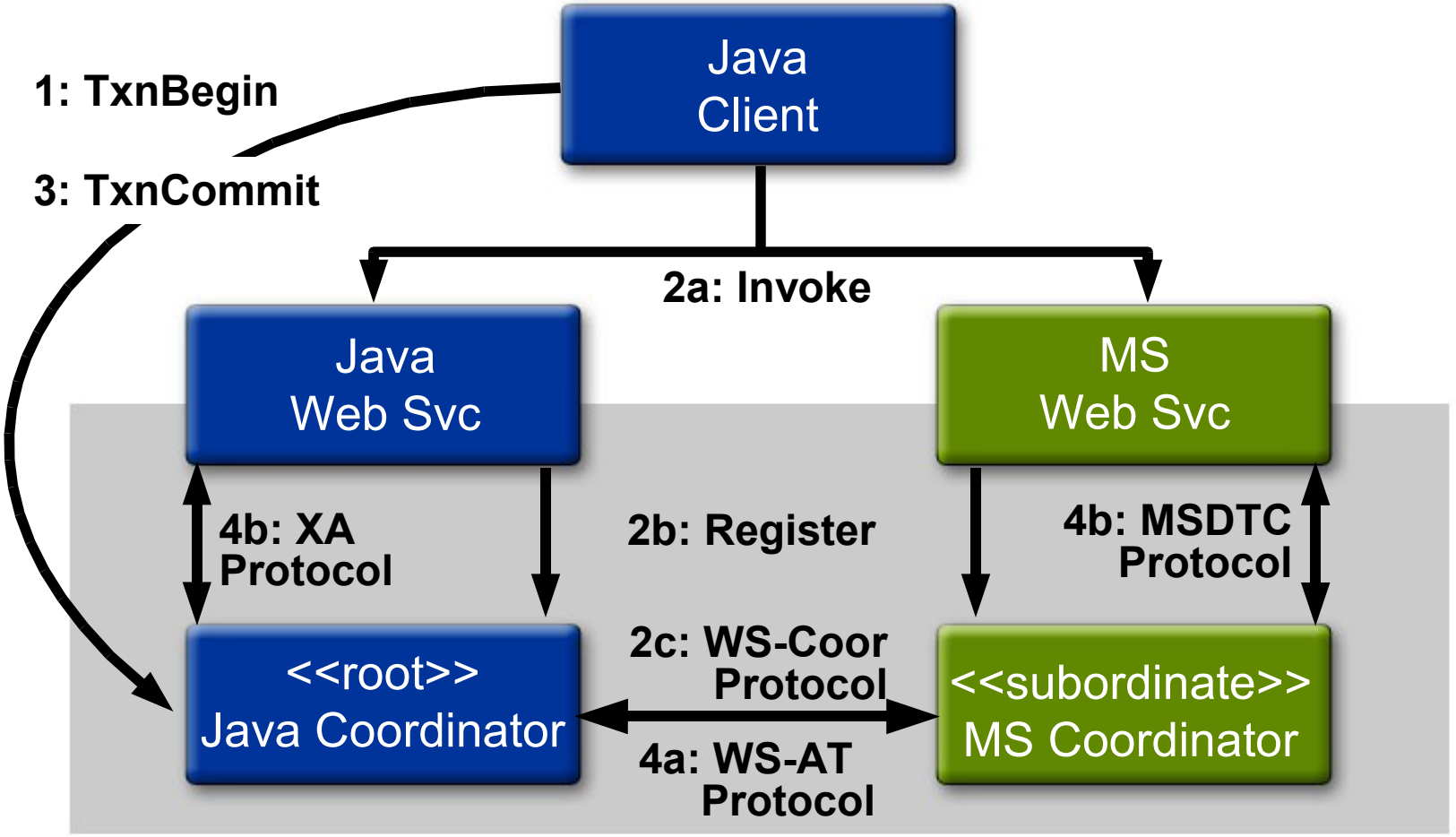
Java Web Method Annotation[1] @javax.ejb.TransactionAttribute value	WS-AT Subject Policy for bound wsdl:operation
MANDATORY	<wsat:ATAssertion/>
REQUIRED[2]	<wsat:ATAssertion wsp:Optional="true"/> <wsat:ATAlwaysCapability/>
REQUIRES_NEW	<wsat:ATAlwaysCapability/>
SUPPORTS	<wsat:ATAssertion wsp:Optional="true"/>
NOT_SUPPORTED	NONE[3]
NEVER	NONE

(1) Also specifiable in deployment descriptor
(2) Default for Container Managed Transaction (CMT) EJB architecture
(3) Closest mapping for WSDL to Java binding

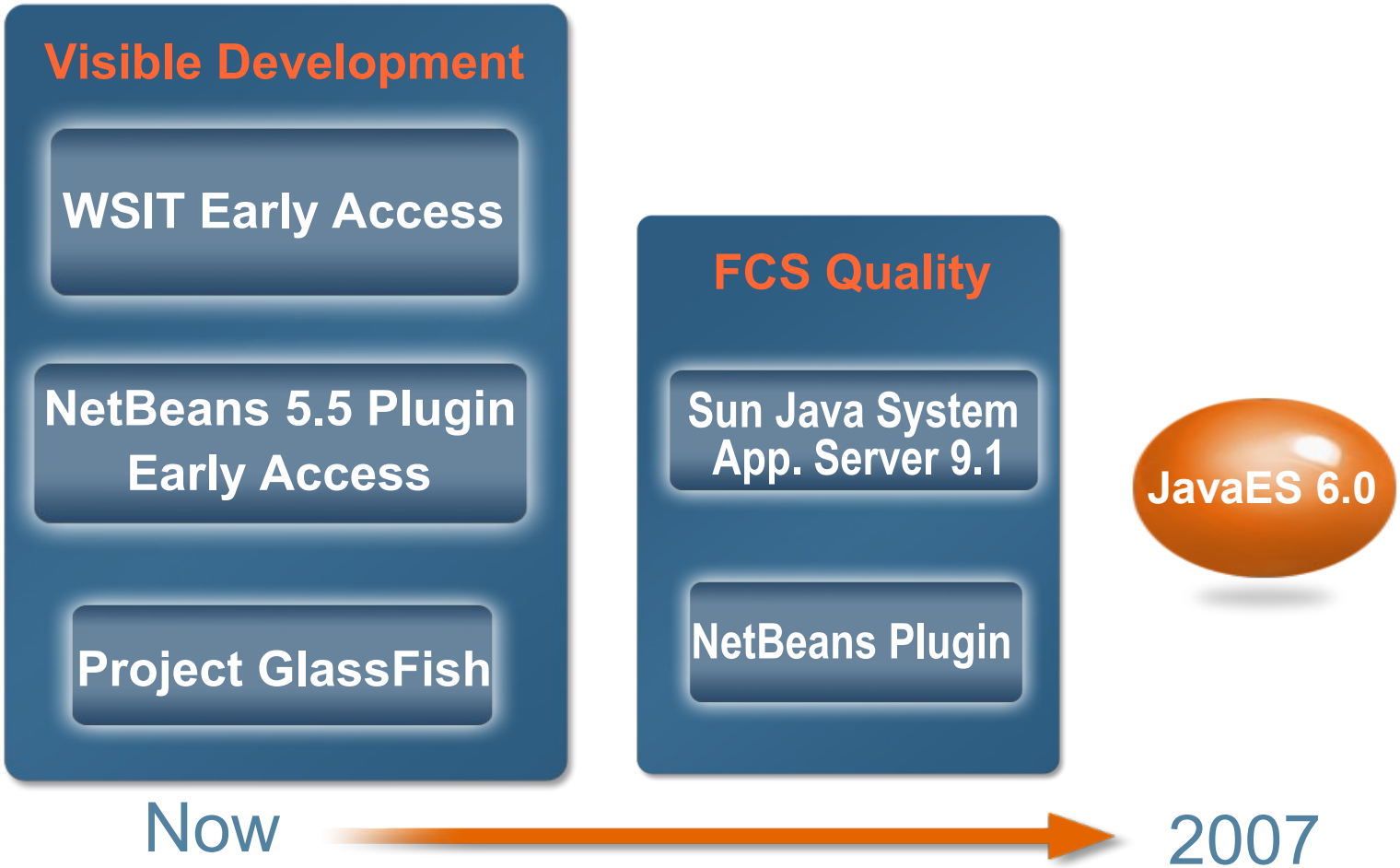
MS Indigo Coordinated Transaction



Java EE/Project Tango Coordinated Transaction



Project Tango: Technology and Product Delivery Plans



Note: transaction functionality is not part of Early Access

Project GlassFish

Simplifying Java application development with **Java EE 5 technologies**

Includes JWSDP, EJB 3.0, JSF 1.2, JAX-WS and JAX-B 2.0

Supports > **20** frameworks and apps

Open source CDDL license

Basis for the **Sun Java System Application Server PE 9**

Free to download and **free** to deploy

Over **1,200** members and **200,000** downloads

Integrated with **NetBeans**

News: blogs.sun.com/theaquarium

Building a Java EE 5 platform open source application server

java.sun.com/javaee/GlassFish

Summary

- Project Tango adds 2 Quality of Service (QoS) technologies for web services to Project Glassfish/Application Server
  - Reliable Messaging
  - Distributed Transactions
- These QoS technologies enable building more reliable distributed systems that use web services
  - Minimal or no new APIs to learn!

Resources

Related Project Tango sessions and labs:

- TS-4661
  - Composable Web Services Using Interoperable Technologies from Sun's "Project Tango"
- TS-5540
  - Making Java™ Technology-Based/.NET Web Services Interoperability Real
- TS-3473
  - Building Secure and Trusted Web Services using Project Tango
- LAB-4335
  - Developing Interoperable Next Generation Web Services with Project Glassfish℠, Netbeans™ IDE and WSIT

Web sites

- http://java.sun/com/webservices/wsinterop

Resources (Cont.)

- WS-Coordination specification
  http://specs.xmlsoap.org/ws/2004/10/wscoor/wscoor.pdf
- WS-Atomic Transaction specification
  http://specs.xmlsoap.org/ws/2004/10/wsat/wsat.pdf
- OASIS WS-TX technical committee
  http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ws-tx

Q&A

Mike Grogan
Joe Fialli
Ryan Shoemaker

Reliable and Transacted Web Services Between Sun's "Project Tango" and Microsoft Indigo

Mike Grogan, Joe Fialli, Ryan Shoemaker

Sun Microsystems, Inc.

TS-1603