

JUDCon

JBoss Users & Developers Conference

2012: Boston

Building Flexible Transactional Applications Using the Narayana TXFramework

Dr Paul Robinson

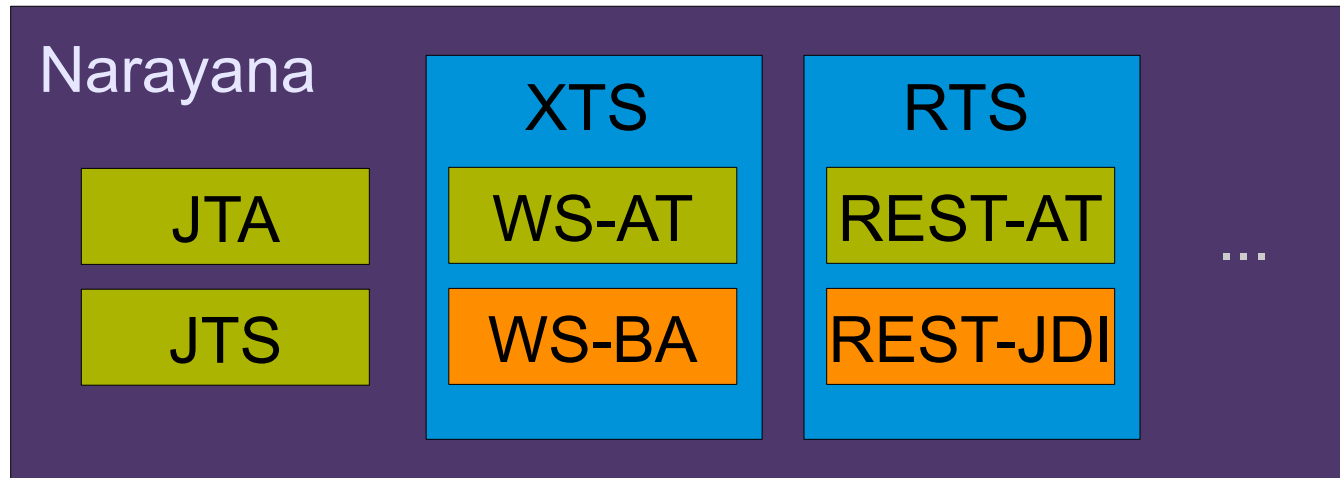
paul.robinson@redhat.com

@pfrobinson

Agenda

- Background
- Code examples
- Customization
- What we have today & planned
- Getting started

Introduction



- Issues
 - Varied APIs
 - Dated developer usability
 - Hard to switch/combine technologies

Extended Transactions

- Relax ACID properties

- ⊕ Long running

- ⊕ Scalable

- ⊕ Performant

- ⊖ Reduced guarantees

- Specifications:

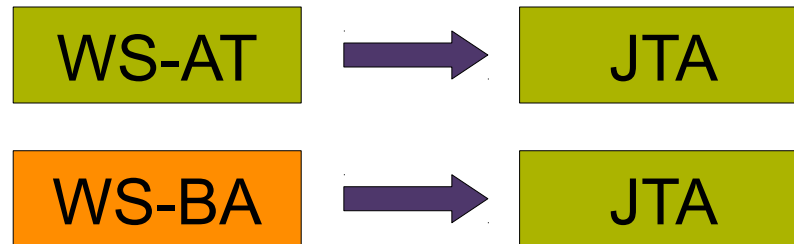
WS-BA

REST-JDI

Transaction Bridging

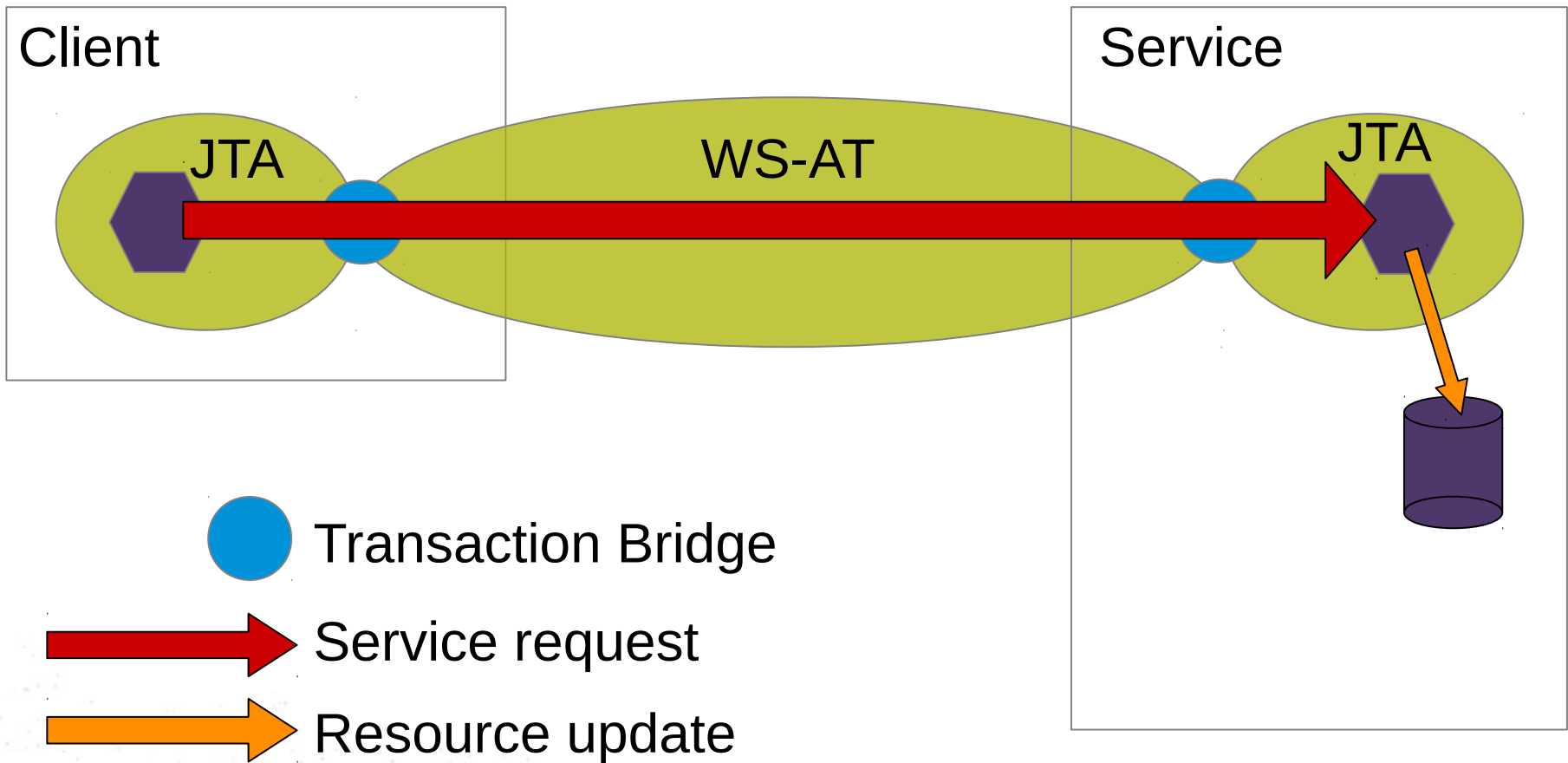
- Distributed & heterogeneous

E.g.

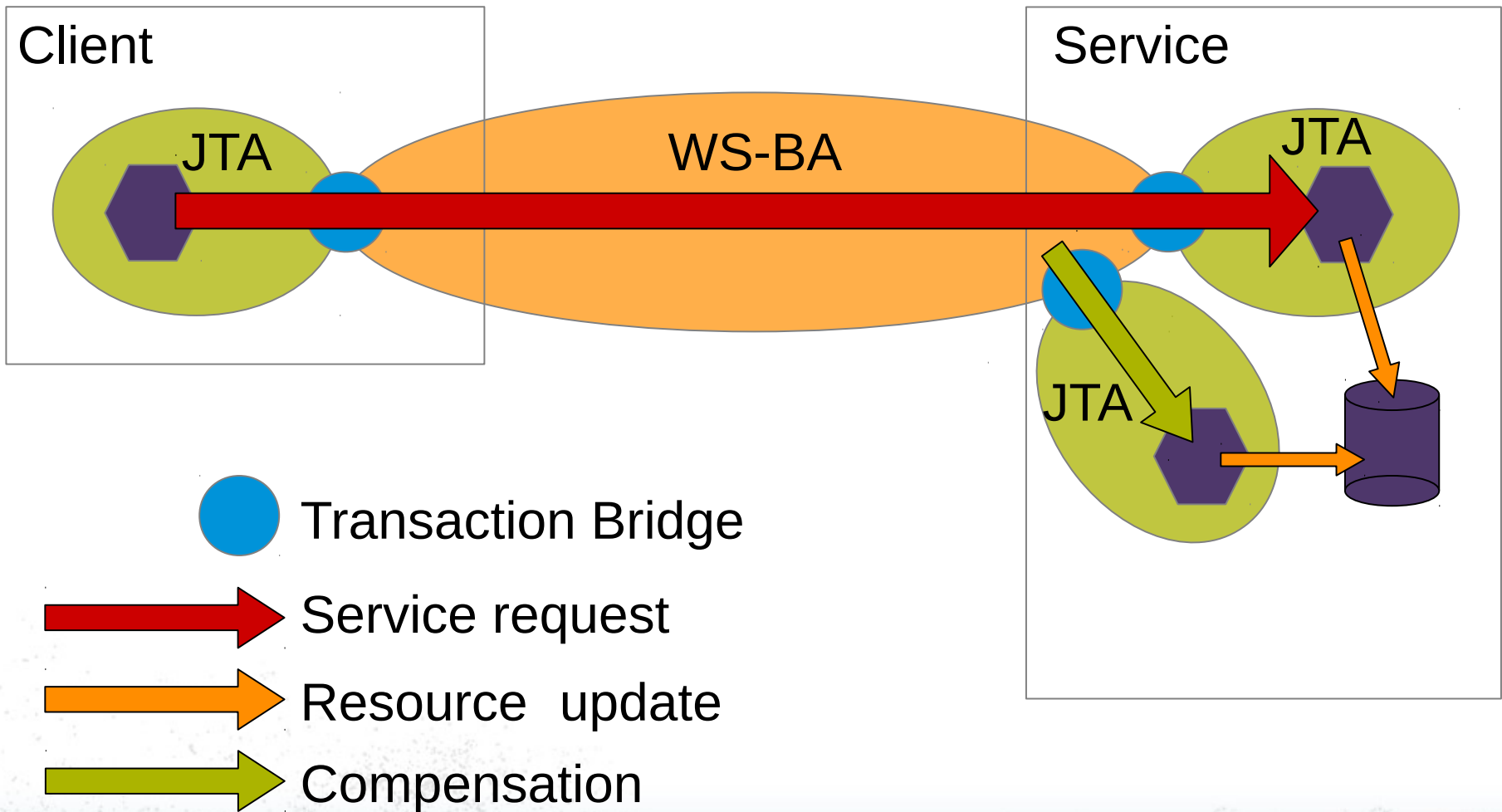


- Usage
 - Different internal/external type
 - Integration

ACID Protocol Bridging



BA to ACID Bridging



The JEE 6 Way

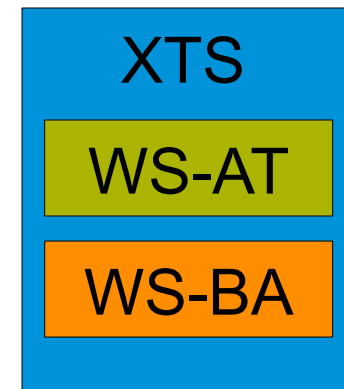
```
@Stateless
public class UserManagement
{
    @Inject
    EntityManager em;

    public void addUser(User user) {
        em.persist(user);
    }
}
```

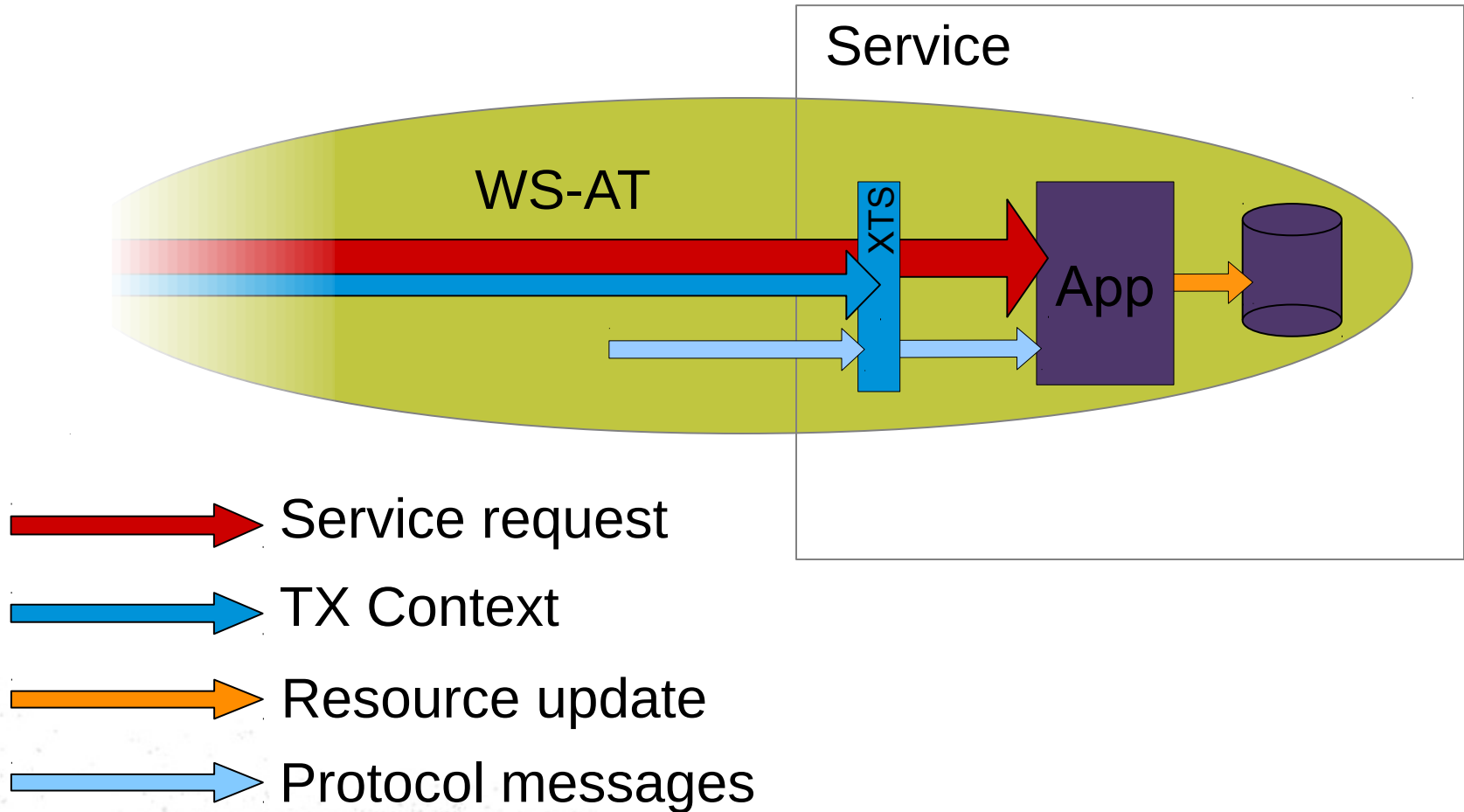
- Annotations
- Sensible defaults
- XML optional

XTS API Issues

- Tied tightly to XTS
- Cumbersome API
- Lots of Boilerplate



XTS Architecture



Using WS-AT without TXFramework

```
@WebService
@HandlerChain(file="/context-handlers.xml", name="Context Handlers")
public class BookServiceImpl implements BookService {

    @WebMethod
    public void orderBook(Order order) throws OrderException {
        String transactionId;
        try {
            transactionId = UserTransactionFactory.userTransaction().toString();
            OrderParticipant orderParticipant =
                new OrderParticipant(transactionId);
            TransactionManager transactionManager =
                TransactionManagerFactory.transactionManager();
            transactionManager.enlistForDurableTwoPhase(orderParticipant,
                "BookServiceImpl:" + UUID.randomUUID());
        } catch (Exception e) {
            throw new OrderException("Error when enlisting participant", e);
        }
        OrderManager.orderBook(order, transactionId);
    }
}
```

Using WS-AT without TXFramework

```
<handler-chains
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee javaee_web_services_1_2.xsd"
...>
  <handler-chain>
    <protocol-bindings>##SOAP11_HTTP</protocol-bindings>
    <handler>
      <handler-name>WebServicesTxContextHandler</handler-name>
      <handler-class>
        com.arjuna.mw.wst11.service.JaxWSHeaderContextProcessor
      </handler-class>
    </handler>
  </handler-chain>
</handler-chains>
```

Using WS-AT without TXFramework

```
public class OrderParticipant implements Durable2PCParticipant {  
  
    Vote prepare() throws WrongStateException, SystemException;  
    void commit() throws WrongStateException, SystemException;  
    void rollback() throws WrongStateException, SystemException;  
    void unknown() throws SystemException;  
    void error() throws SystemException;  
}
```

Using WS-AT with TXFramework

```
@Transactional
@Stateless
@WebService
public class BookServiceImpl implements BookService {

    @DataManagement private Map txDataMap;

    @WebMethod
    @ServiceRequest
    public void orderBook(Order order) throws OrderException {
        txDataMap.put("order", order);
        //Create the order by updating some resource
    }

    @Prepare public Boolean prepare() {
        Order order = (Order) txDataMap.get("order");
        //Prepare the resource and return outcome
    }

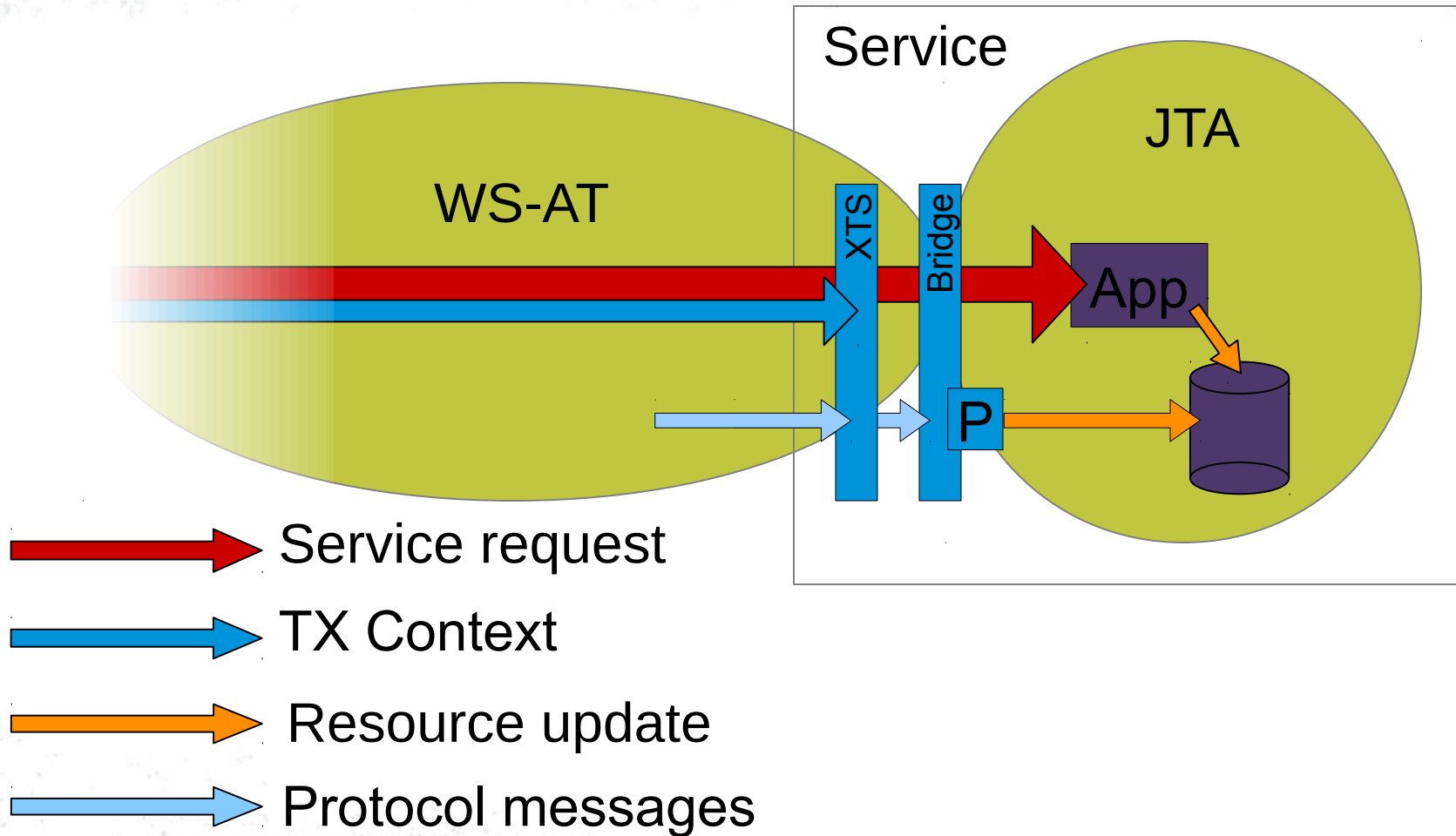
    @Commit public void commit() {...}

    @Rollback public void rollback() {...}
}
```

Bridging WS-AT to/from JTA

- Developer manages handler chain
 - Required XML
 - Error prone
 - Extra work
 - Transport dependent

XTS Bridging Architecture



Bridging without TXFramework

```
@Stateless
@WebService
@HandlerChain(file = "jaxws-handlers-server.xml")
public class BookServiceImpl implements BookService {
{
    @PersistenceContext protected EntityManager em;

    @WebMethod
    public void orderBook(Order order) throws OrderException {
        //Use em to add order to DB
    }
}
```

Bridging without TXFramework

```
<handler-chains
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee javaee_web_services_1_2.xsd"
...>
<handler-chain>
  <protocol-bindings>##SOAP11_HTTP</protocol-bindings>
  <handler>
    <handler-name>TransactionBridgeHandler</handler-name>
    <handler-class>
      org.jboss.jbossts.txbridge.inbound.JaxWSTxInboundBridgeHandler
    </handler-class>
  </handler>
  <handler>
    <handler-name>WebServicesTxContextHandler</handler-name>
    <handler-class>
      com.arjuna.mw.wst11.service.JaxWSHeaderContextProcessor
    </handler-class>
  </handler>
</handler-chain>
</handler-chains>
```

Bridging without TXFramework

```
//Get the service port
```

```
BookService bookService = ...
```

```
//Create handler chain
```

```
List<Handler> handlers = new ArrayList<Handler>();
```

```
handlers.add(new JaxWSTxOutboundBridgeHandler()); //JTA-WSAT bridge handler
```

```
handlers.add(new JaxWSHeaderContextProcessor()); //XTS handler
```

```
//Obtain the BindingProvider and set the handler chain
```

```
BindingProvider bindingProvider = (BindingProvider) bookService;
```

```
bindingProvider.getBinding().setHandlerChain(handlers);
```

AT Bridging with TXFramework

```
@Transactional //By default bridge to JTA
@Stateless
@WebService
public class BookServiceImpl implements BookService {
    {
        @PersistenceContext protected EntityManager em;

        @WebMethod
        @ServiceRequest
        public void orderBook(Order order) throws OrderException {
            //Use em to add order to DB
        }
    }
}
```

BA Bridging with TXFramework

@Compensatable

@Stateless

@WebService

```
public class BookServiceImpl implements BookService {  
{
```

```
    @DataManagement private Map txDataMap;
```

```
    @PersistenceContext protected EntityManager em;
```

```
    @WebMethod
```

```
    @ServiceRequest
```

```
    public void orderBook(Order order) throws OrderException {  
        txDataMap.put("order", order);  
        //Use em to add order to DB  
    }  
}
```

```
    @Compensate
```

```
    private void compensate() {  
        Order order = (Order) txDataMap.get("order");  
        //Use em to cancel order in DB  
    }  
}
```

```
}
```

Bridging Customizations

- Prevent bridging

```
@Transactional(bridgeType = BridgeType.NONE)
```

```
public class BookServiceImpl
```

```
@ServiceRequest(bridgeType = BridgeType.NONE)
```

```
public void orderBook(Order order)
```

- Specify target type

```
BridgeType.JTA
```

```
BridgeType.WSAT
```

```
BridgeType.RESTAT
```

Switching Protocols

- Switching WS-AT to REST-AT
 - Just replace JAX-WS annotations with JAX-RS annotations

WS-AT Service

```
@Transactional(bridgeType=BridgeType.NONE)
@WebService
public class BookServiceImpl implements BookService {

    @WebMethod
    @ServiceRequest
    public void orderBook(Order order) throws OrderException {
        txDataMap.put("order", order);
        //Create the order
    }

    @Prepare public Boolean prepare() {...}
    @Commit public void commit() {...}
    @Rollback public void rollback() {...}
}
```

REST-AT Service

```
@Transactional(bridgeType=BridgeType.NONE)
@Path("/")
public class BookServiceImpl implements BookService {

    @POST
    @ServiceRequest
    public void orderBook(Order order) throws OrderException {
        txDataMap.put("order", order);
        //Create the order
    }

    @Prepare public Boolean prepare() {...}
    @Commit public void commit() {...}
    @Rollback public void rollback() {...}
}
```

Other Customizations

- Participant optimizations

```
@ReadOnly public boolean isAvailable(Book book)
@Completes public void checkout(Integer basketId)
```

- External Participants

```
@Transactional(participant=MyParticipant.class)
public class BookServiceImpl
```

```
@ServiceRequest(participant=MyParticipant.class)
public void orderBook(Order order)
```

- Parametrized lifecycle methods

```
@ServiceRequest
public void orderBook(Order order)
@Compensate
public void cancelOrder(Order order)
```

What We Have Today

- Early release in Narayana master
- Quickstarts
 - WS-AT, WS-BA, REST-AT
 - Bridging WS-AT from/to JTA
- No other documentation yet :-)

What we Have Planned

- Documentation
- Shipped with JBoss AS
- New Features
 - More bridging
 - Recovery
 - Continued simplification

Getting Started

- Explore the quickstarts
 - <http://github.com/jbosstm/quickstart/>
 - TXFramework/README.md
- Give feedback
 - <http://community.jboss.org/en/jbosstm>
- Track issues
 - <http://issues.jboss.org/browse/JBTM>
 - TXFramework component
- Subscribe to Blog
 - <http://jbossts.blogspot.co.uk/>
- Contact me
 - Paul.robinson@redhat.com, @pfrobinson