ID2208 Programming Web Services

Web Services Coordination

Mihhail Matskin:
http://people.kth.se/~misha/ID2208/

Spring 2016

Content

Coordination

- The need for coordination
- Modeling conversation
- Infrastructure for coordination
- WS-Coordination
- WS-Transactions
- Atomic transactions
- Business activity

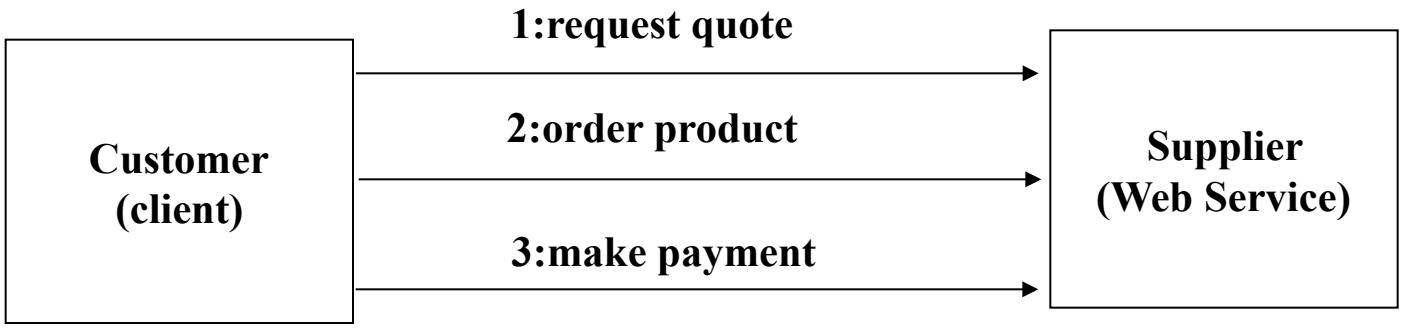
This lecture reference

Text-book **Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI, 2nd Edition**

Chapter 11

The Need for Coordination

A set of invocations
Some of them supported by supplier
and some are disallowed



The internal business logic of client
and Web service must support
the conversation

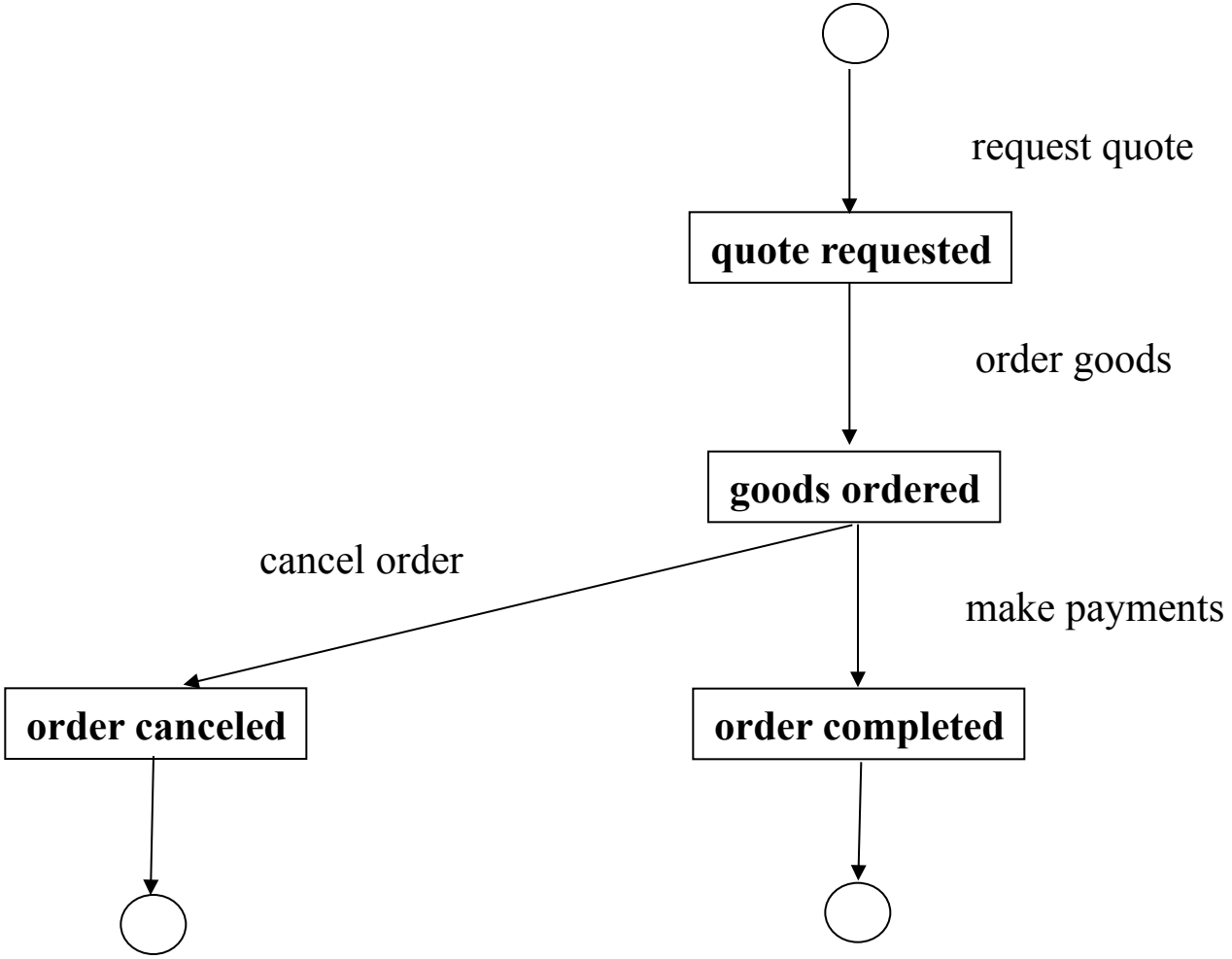
The Need for Coordination

- Internal perspective (implementation)
  - the client must be able to execute complex procedures in apporpriate order
  - the client must be capable of maintainig context information to be forwarded from one operation to another
  - an internal logic of a client is more complex

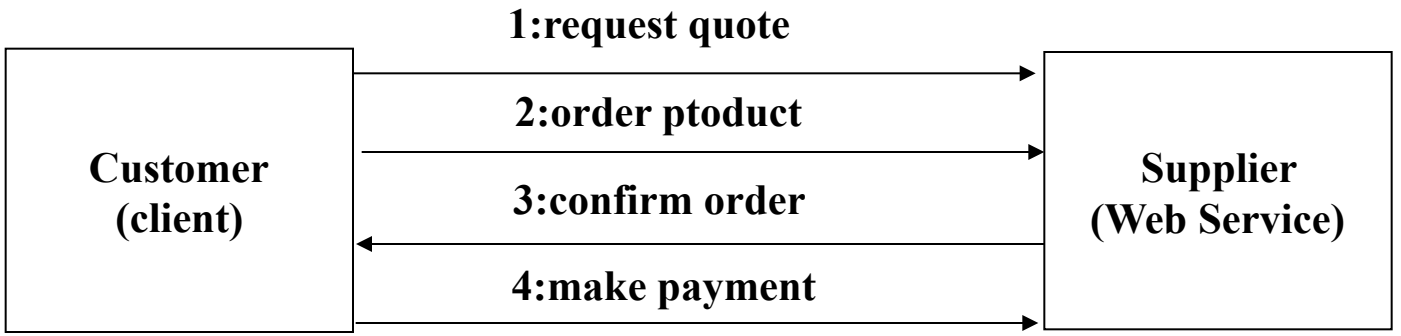
The Need for Coordination

- External perspective (interactions)
  - Ensuring order of operation (constraints)
  - Conversation rather than simple invocation (coordination protocol as a specification of the set of correct and accepted conversations)
  - How to present coordination protocols?

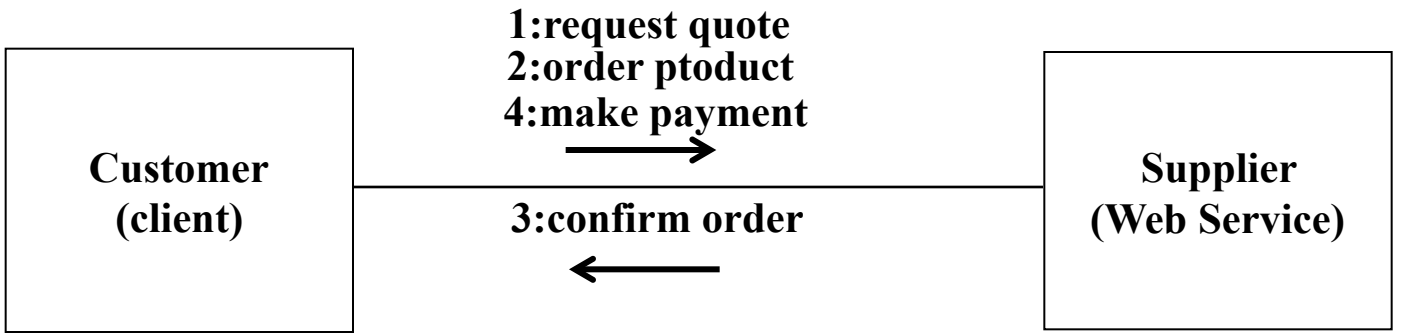
Modeling conversation (single client view)



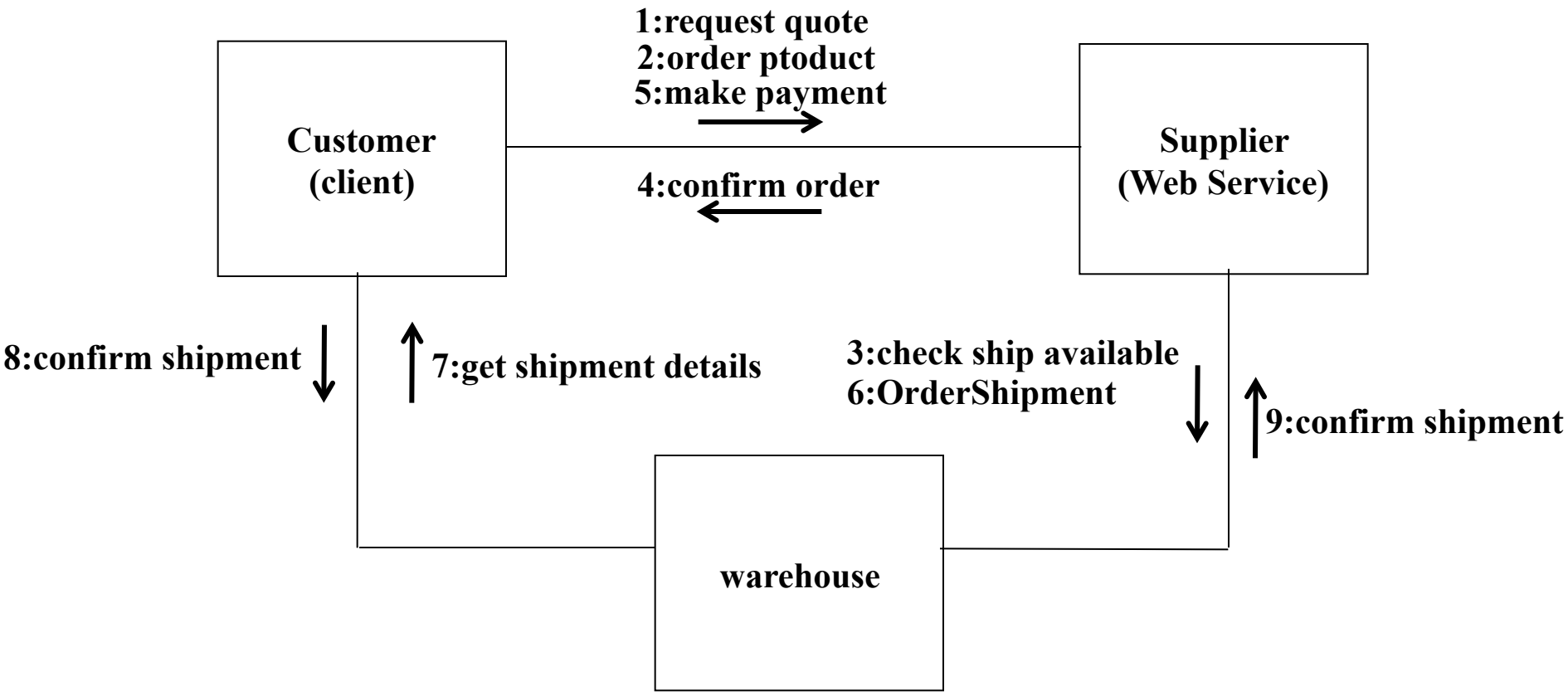
Modeling conversation (two parties)



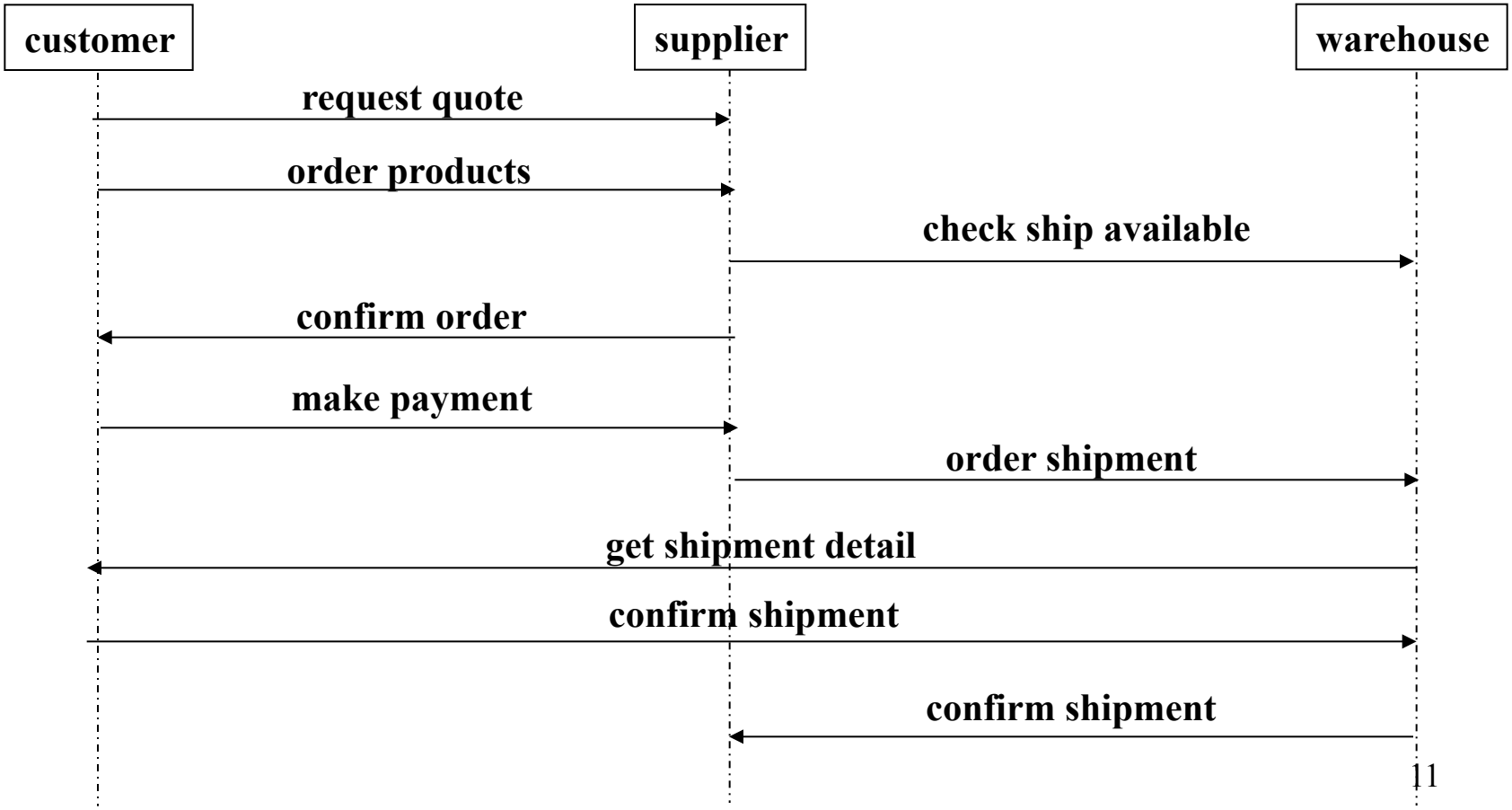
Modeling conversation (two parties)



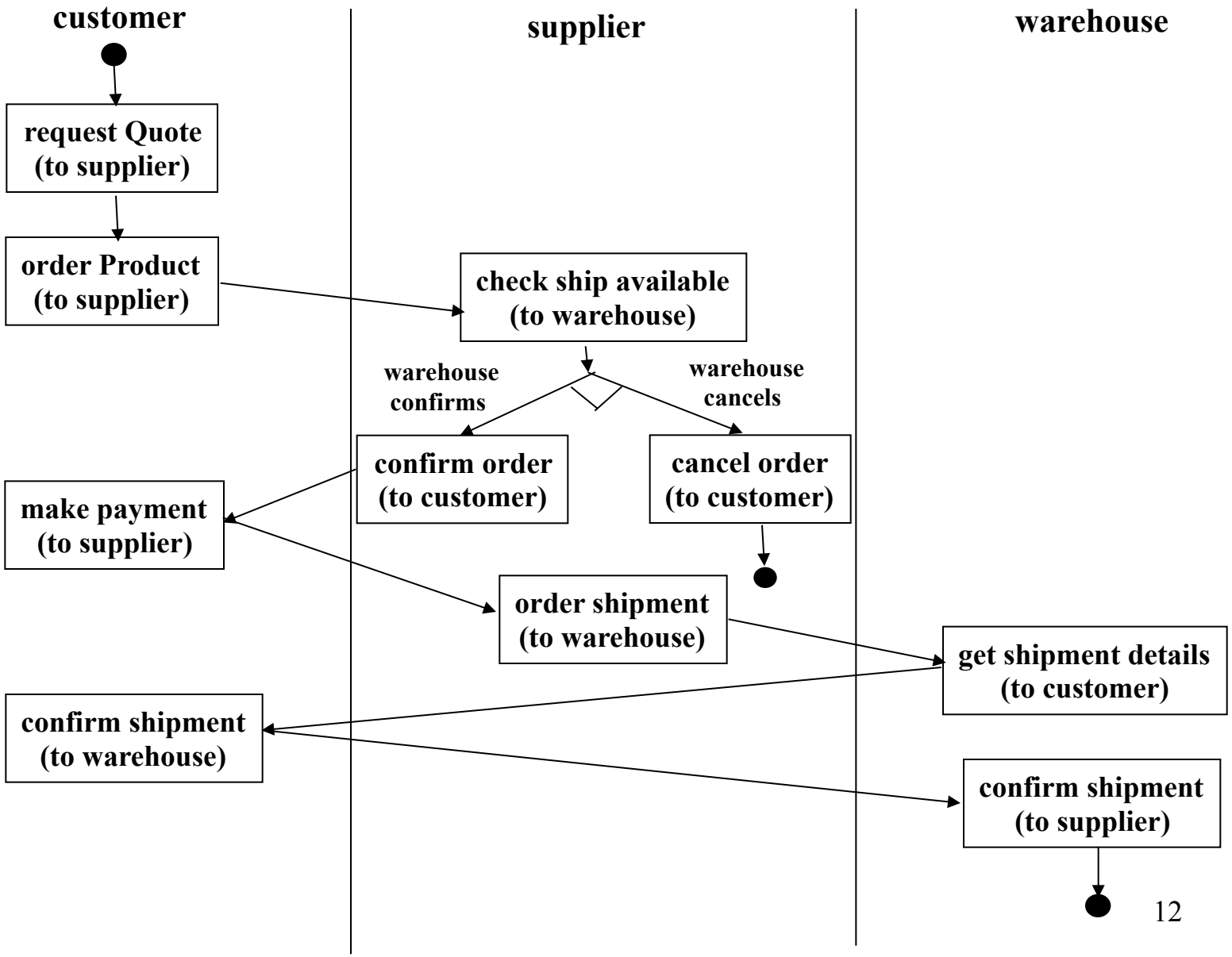
Modeling conversation (multiple parties)



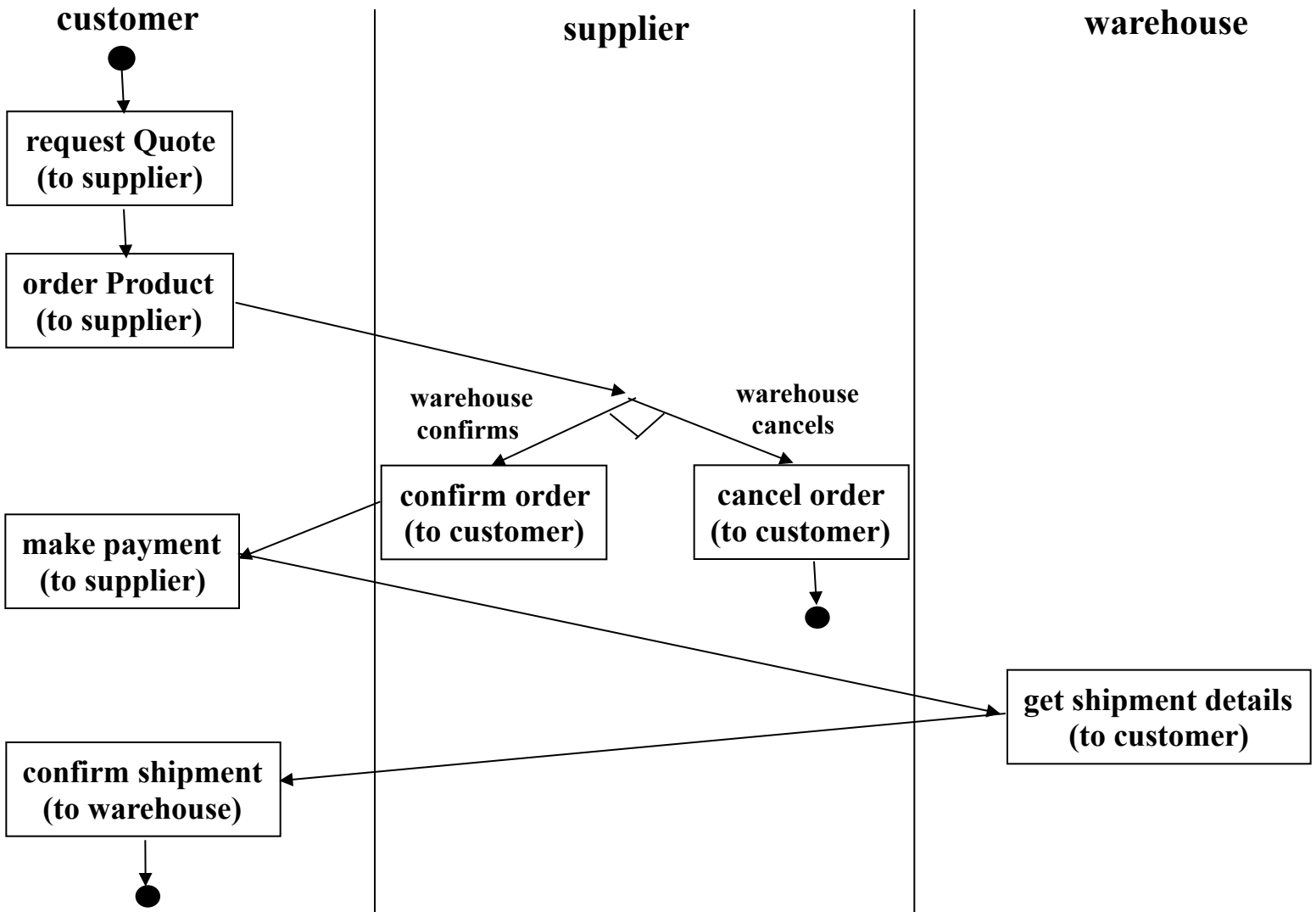
Modeling conversation (multiple parties, sequence diagram)



Modeling conversation (multiple parties, activity diagrams)



Role Specific Views



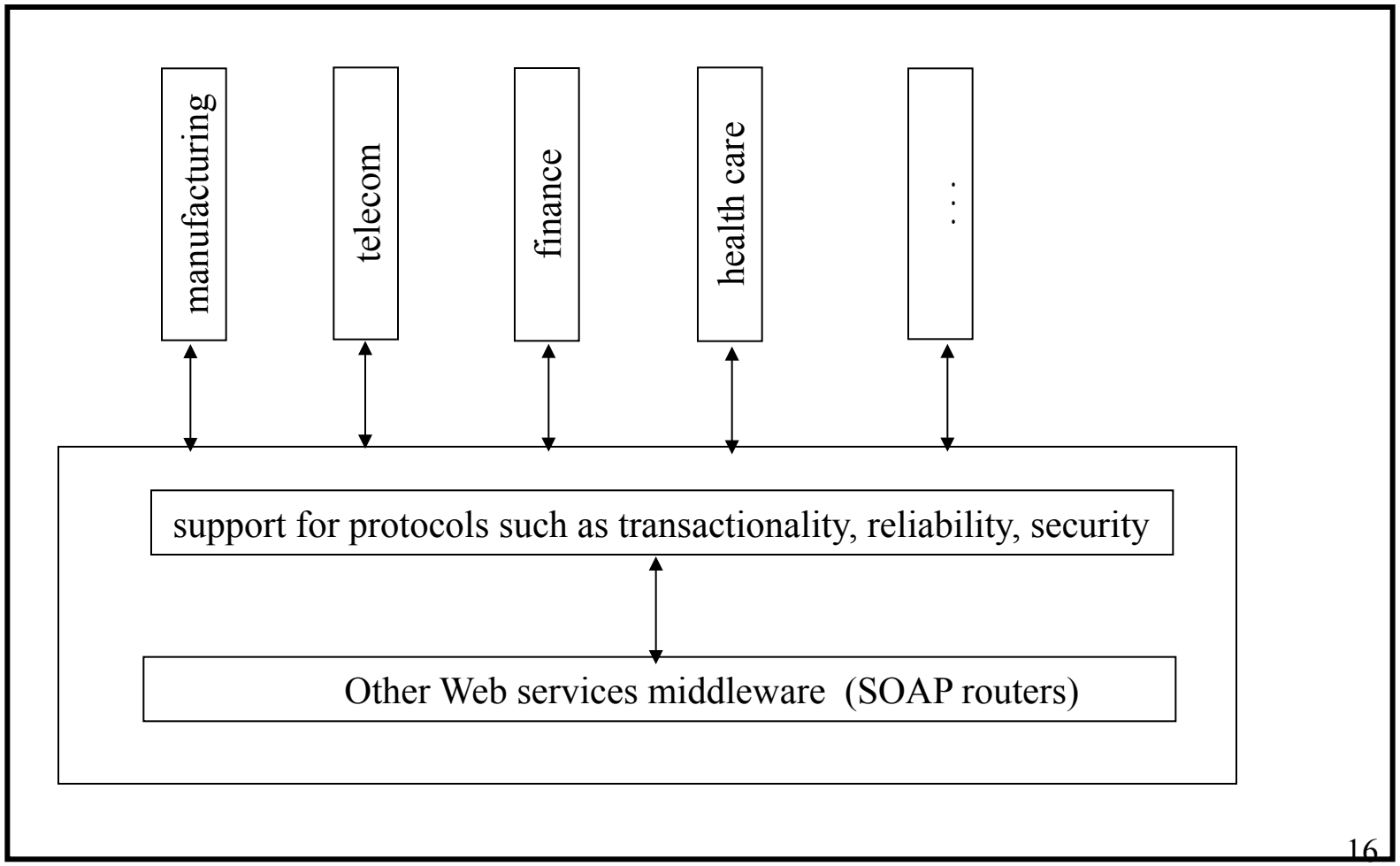
Service Interface and Coordination Protocols

- Similarity between service interface and coordination protocols
- Coordination protocol can be, for example, published as **`tModels`**
- The fact that a Web service plays a certain role can be described as **`bindingTemplate`** referring to the **`tModel`** containing the coordination protocol

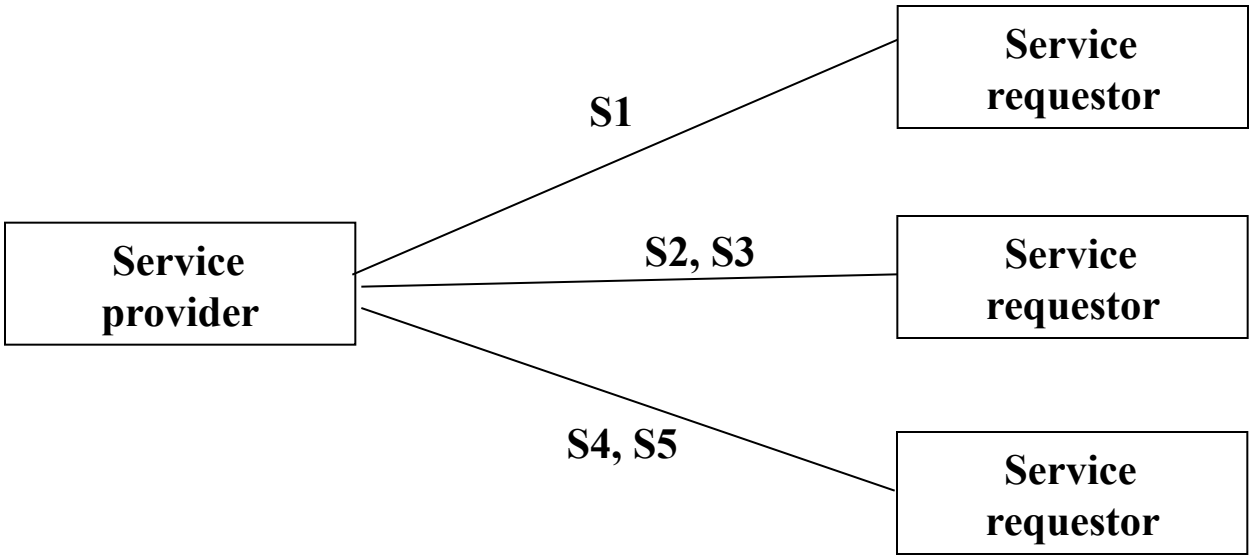
Classification of Web services protocols

- Vertical protocols - specific to business areas and applicable in some domains
  - do not necessarily include all low level details (for example, how to exchange messages)
  - they concern much more with semantic aspects of the exchanges
  - they typically rely on infrastructure based on horizontal protocols
- Horizontal (middleware) protocols - define a common infrastructure independent of the application area
  - Endow exchanges between services with higher-level abstractions implemented by the Web services middleware in a transparent manner
  - provide reliability and service management functionality

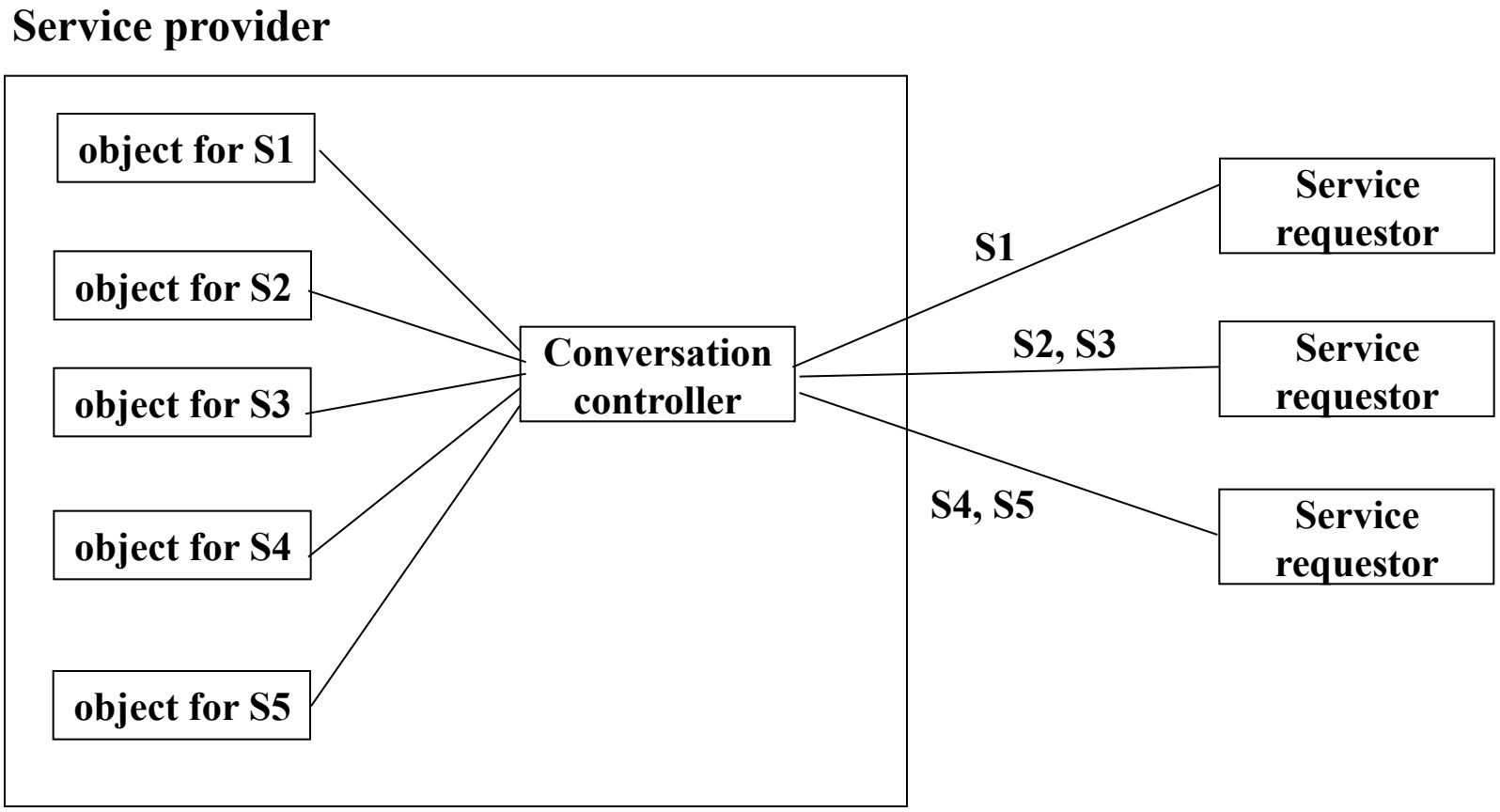
Horizontal and vertical protocols



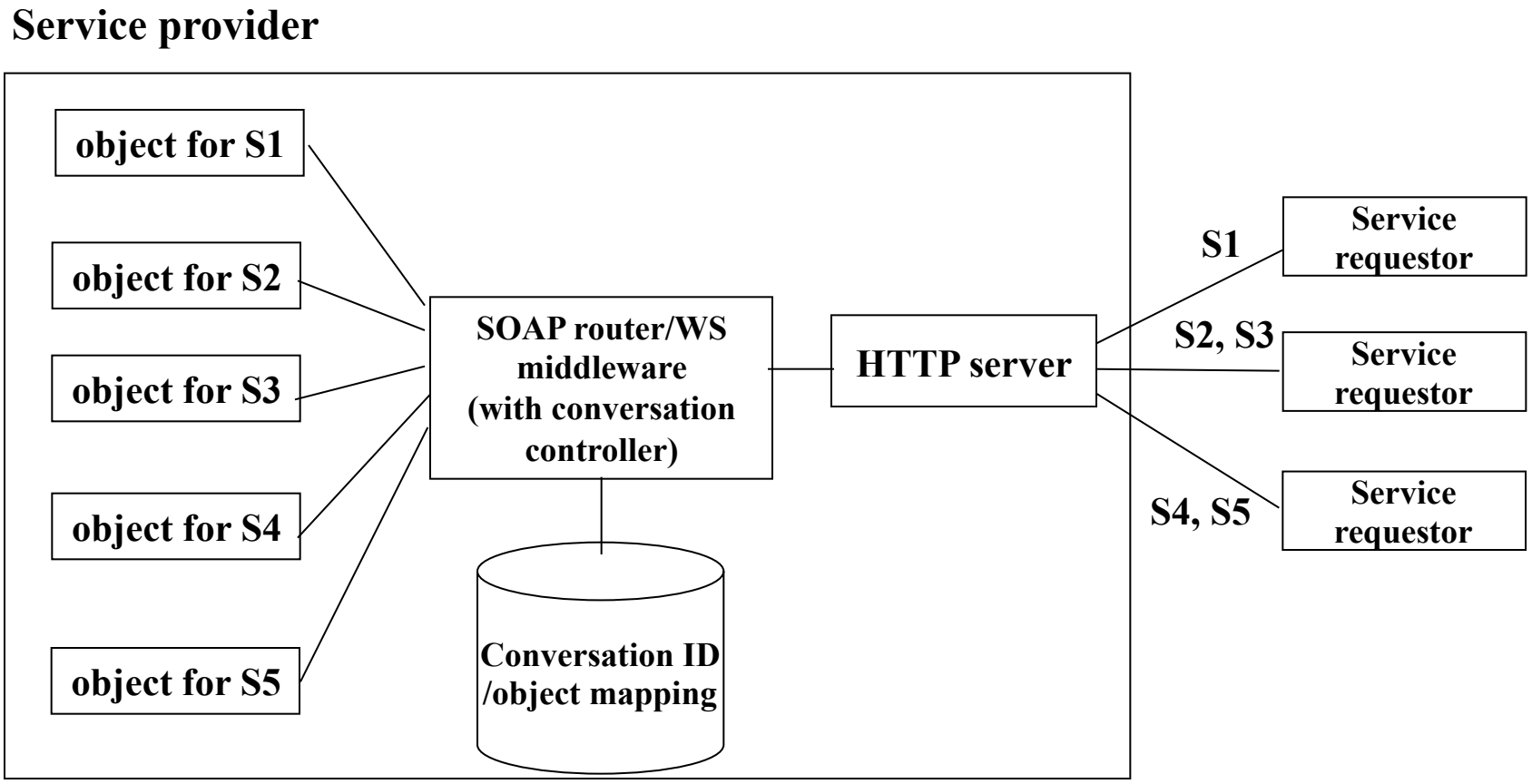
Infrastructure for coordination protocols



Infrastructure for coordination protocols (conversation controllers)



Infrastructure for coordination protocols (conversation controllers)



Standardization requirements

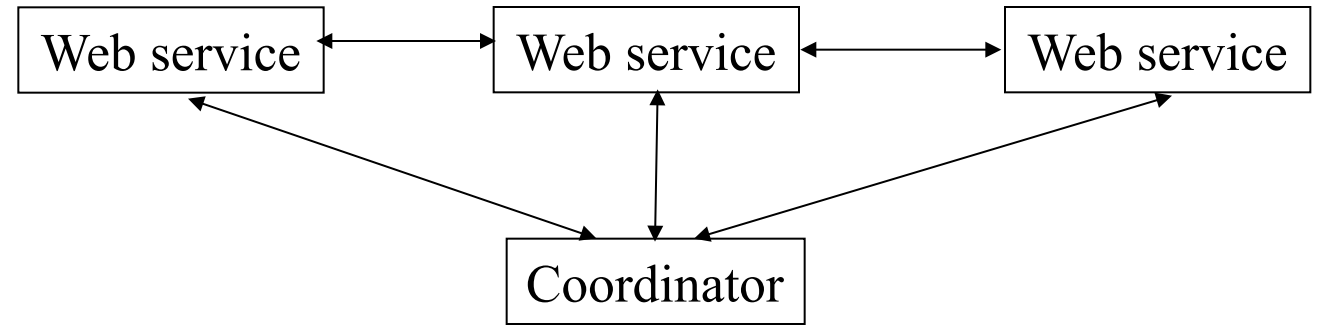
- It requires a way to generate and transport unique conversation identifiers in the headers of SOAP messages
- It requires a framework
- It requires horizontal protocols to be standardized

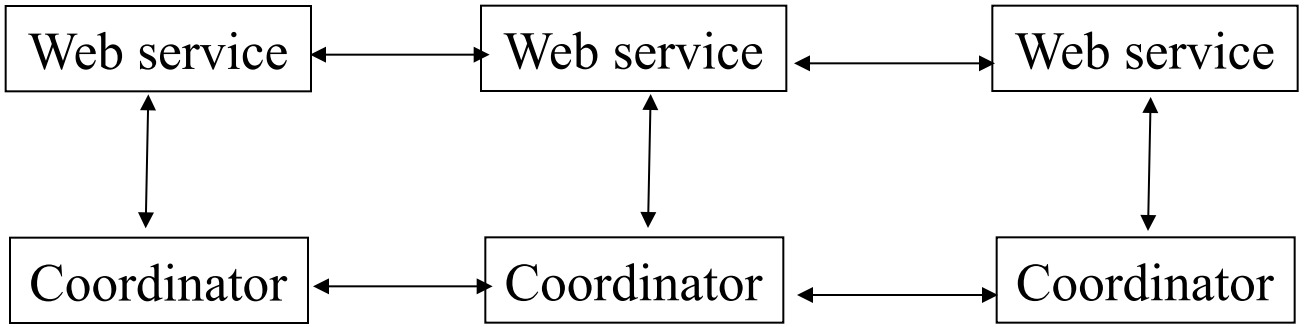
WS-Coordination (goals)

- To create a framework for supporting coordination protocols

- It is intended as a meta-specification that will govern specifications that implement concrete forms of coordination (for example, transactional coordination)

Components of WS-Coordination





WS-Coordination (abstractions)

- A coordination protocol – a set of rules governing conversation
- A coordination type – a set of coordination protocols logically related to each other
- A coordination context – a data structure used to mark messages belonging to the same conversation

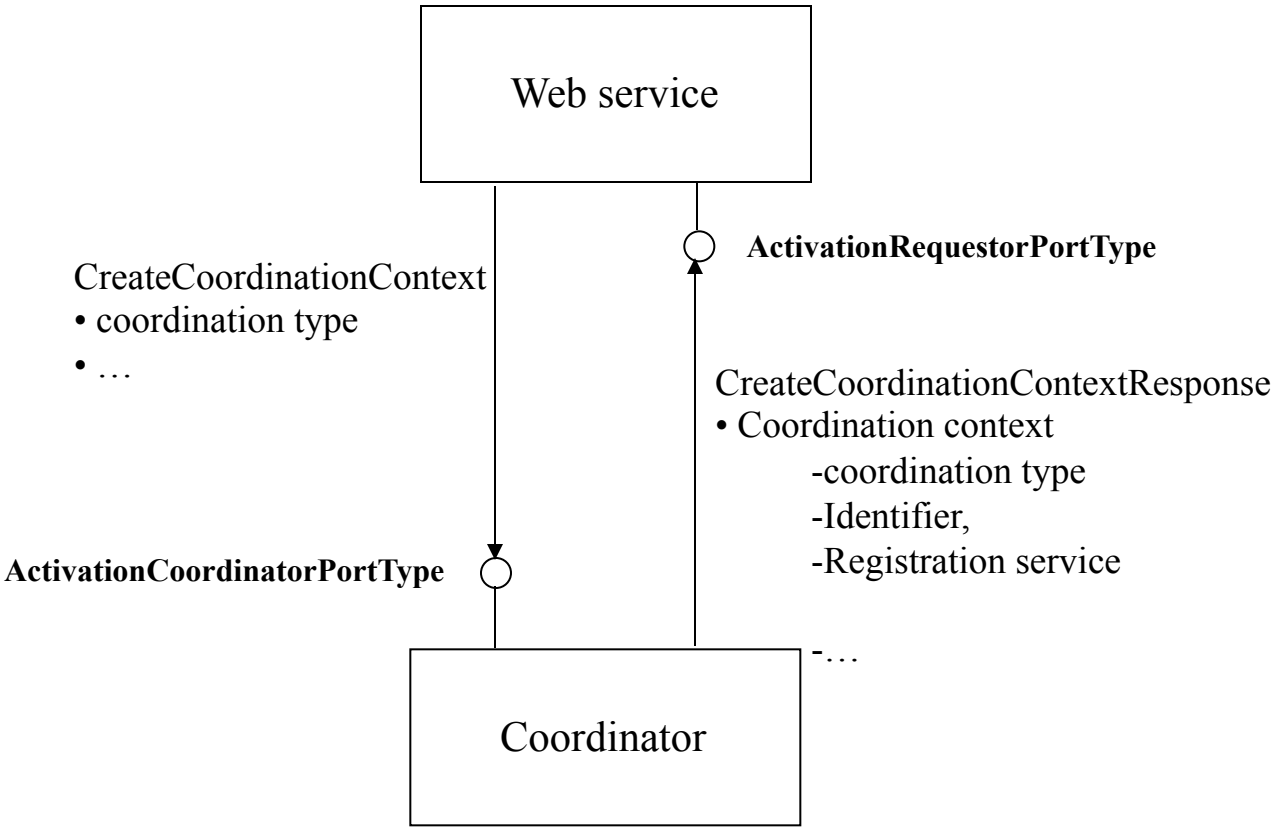
WS-Coordination (forms of interactions)

- Activation - a participant requests a coordinator to create a new coordination context
- Registration – a participant registers as a coordination protocol participant with a coordinator
- Protocol-specific interactions – the coordinator and its participants exchange messages that are specific to a coordination protocol

wscoor.wsdl

```
<?xml version="1.0" encoding="utf-8" ?>  <!-- Copyright (c) OASIS Open 2006.   -->
  <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:wscoor="http://docs.oasis-
   open.org/ws-tx/wscoor/2006/06" targetNamespace="http://docs.oasis-open.org/ws-tx/wscoor/2006/06">
 <wsdl:types>
  <xs:schema>
    <xs:import namespace="http://docs.oasis-open.org/ws-tx/wscoor/2006/06" schemaLocation="http://docs.oasis-open.org/ws-tx/
   wscoor/2006/06/wstx-wscoor-1.1-schema-200701.xsd" />
  </xs:schema>
 </wsdl:types>
  <wsdl:message name="CreateCoordinationContext">
    <wsdl:part name="parameters" element="wscoor:CreateCoordinationContext" />
  </wsdl:message>
  <wsdl:message name="CreateCoordinationContextResponse">
    <wsdl:part name="parameters" element="wscoor:CreateCoordinationContextResponse" />
  </wsdl:message>
  <wsdl:message name="Register">
    <wsdl:part name="parameters" element="wscoor:Register" />
  </wsdl:message>
  <wsdl:message name="RegisterResponse">
    <wsdl:part name="parameters" element="wscoor:RegisterResponse" />
   </wsdl:message>
   <wsdl:portType name="ActivationPortType">
    <wsdl:operation name="CreateCoordinationContextOperation">
      <wsdl:input message="wscoor:CreateCoordinationContext" />
      <wsdl:output message="wscoor:CreateCoordinationContextResponse" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:portType name="RegistrationPortType">
    <wsdl:operation name="RegisterOperation">
      <wsdl:input message="wscoor:Register" />
      <wsdl:output message="wscoor:RegisterResponse" />
    </wsdl:operation>
  </wsdl:portType>
  . . .
</wsdl:definitions>
```

WS-Coordination (activation)



WS-Coordination (`CreateCoordinationContext`)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
   soap/envelope/"
   xmlns:wscoor="http://schemas.xmlsoap.org/ws/2002/09/wscoor">
   <soapenv:Body>
      <wscoor:CreateCoordinationContext>
               <wscoor:CoordinationType>
                     http://schemas.xmlsoap.org/ws/2002/09/wsat
               </wscoor:CoordinationType>
      </wscoor:CreateCoordinationContext>
   </soapenv:Body>
</soapenv:Envelope>
```

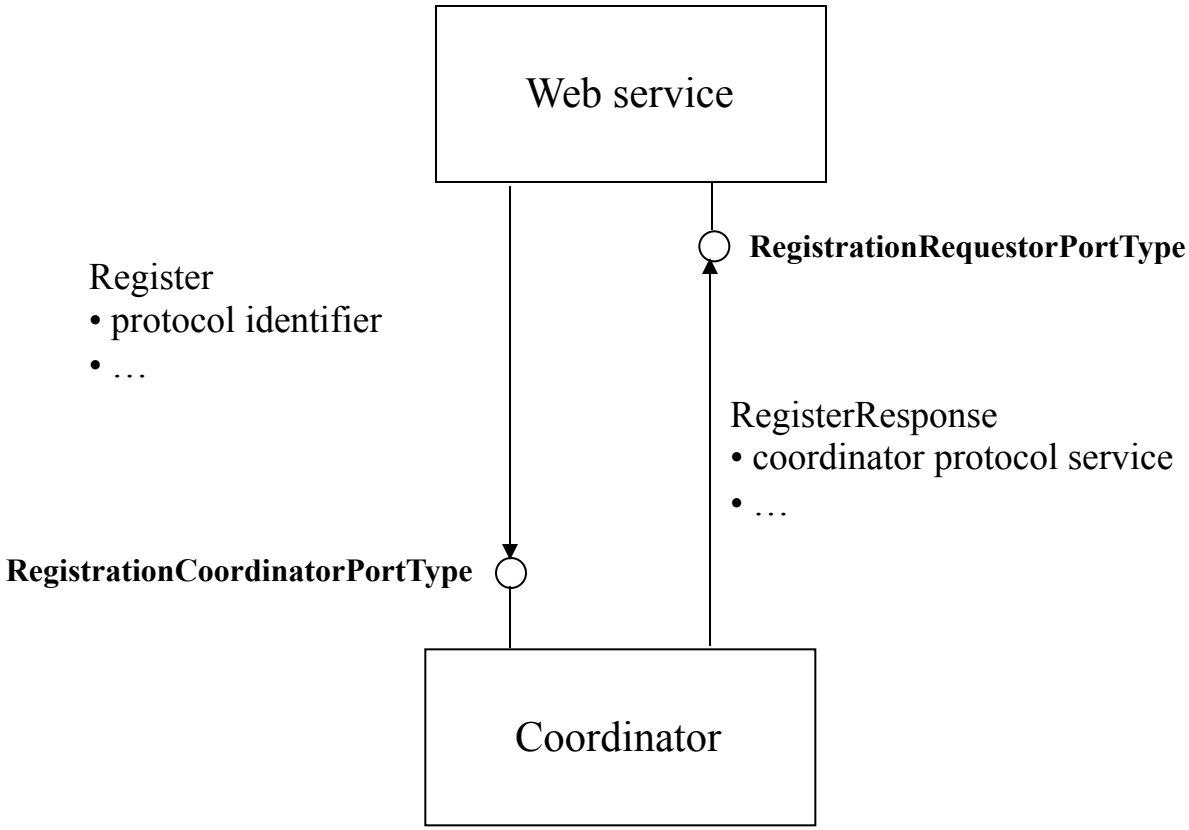
WS-Coordination (`CoordinationContext`)

```
<wscoor:CoordinationContext>
   <wsu:Identifier>
        http://skatestown.com/trans2
   </wsu:Identifier>
   <wsu:Expires>2012-12-31T09:00:00Z</wsu:Expires>
</wscoor:CoordinationContext>
```

WS-Coordination(createCoordinationContextResponse)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
   soap/envelope/"
   xmlns:wscoor="http://schemas.xmlsoap.org/ws/2002/09/wscoor"
   xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility"
   xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing">
   <soapenv:Body>
      <wscoor:CreateCoordinationContextResponse>
         <wsu:Identifier>
            http://skatestown.com/trans2
         </wsu:Identifier>
         <wsu:Expires>2004-12-31T09:00:00Z</wsu:Expires>
         <wscoor:CoordinationType>
            http://schemas.xmlsoap.org/ws/2002/09/wsat
         </wscoor:CoordinationType>
         <wscoor:RegistrationService>
            <wsa:Address>
               http://coordinator.com/register
            </wsa:Address>
         </wscoor:RegistrationService>
      </wscoor:CreateCoordinationContextResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

WS-Coordination (registration)



WS-Coordination (registration)

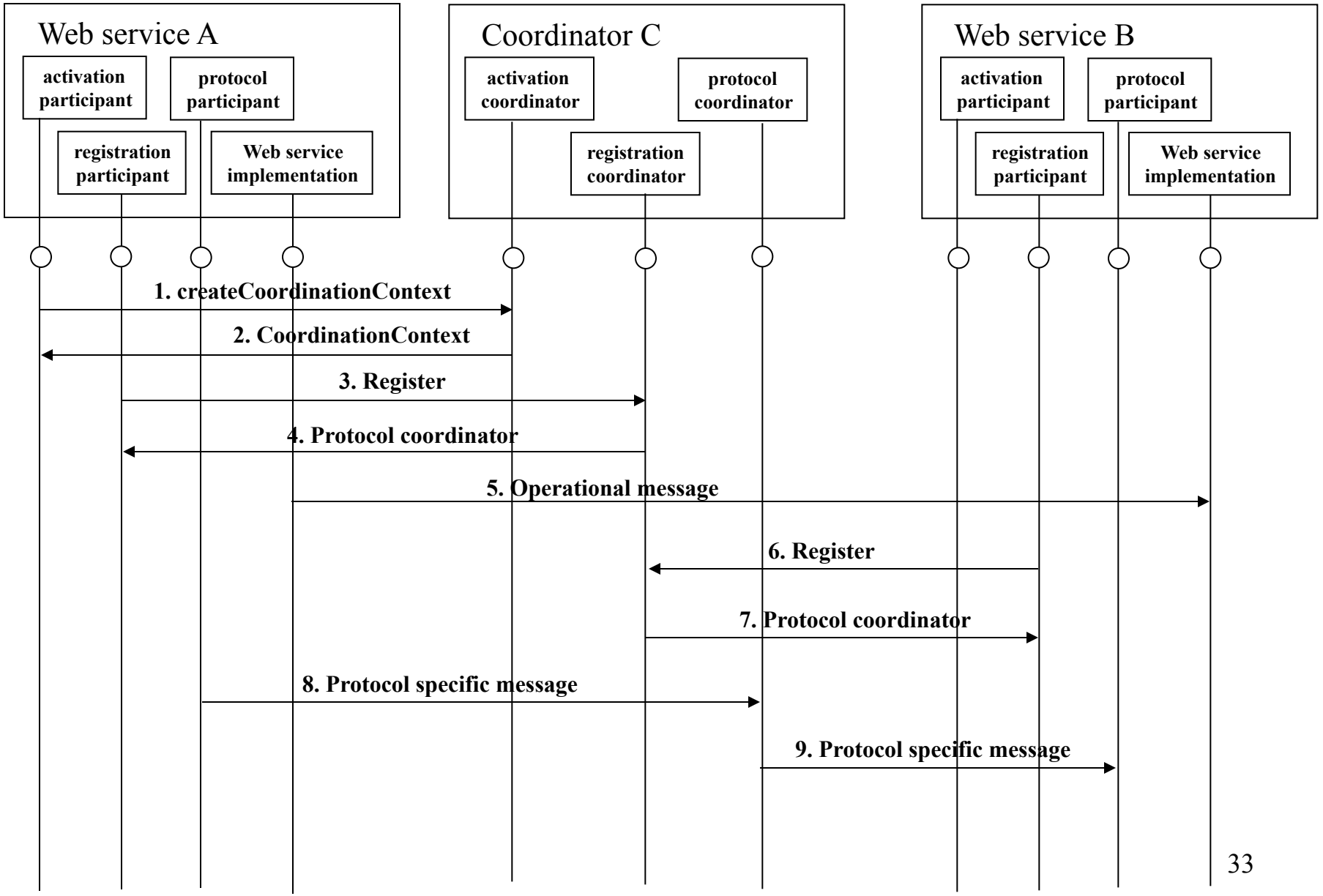
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/
   envelope/"
   xmlns:wscoor="http://schemas.xmlsoap.org/ws/2002/09/wscoor"
   xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility">
   <soapenv:Header>
     <wscoor:CoordinationContext mustUnderstand="true">
          <wsu:Identifier>http://skatestown.com/trans2</wsu:Identifier>
           <wsu:Expires>2004-12-31T09:00:00Z</wsu:Expires>
           <wscoor:CoordinationType>
                  http://schemas.xmlsoap.org/ws/2002/09/wsat
           </wscoor:CoordinationType>
           <wscoor:RegistrationService>
              <wsa:Address>http://coordinator.com/register</wsa:Address>
           </wscoor:RegistrationService>
     </wscoor:CoordinationContext>
   <soapenv:Body>
     <wscoor:Register>
           <wscoor:ProtocolIdentifier>
                http://schemas.xmlsoap.org/ws/2003/09/wsat#Durable2PC
           </wscoor:ProtocolIdentifier>
           <wscoor:ParticipantProtocolService>
               <wsa:Address>http://Wheels.com/2pcservice</wsa:Address>

           </wscoor:ParticipantProtocolService>
     </wscoor:Register>
   </soapenv:Body>
</soapenv:Envelope>
```

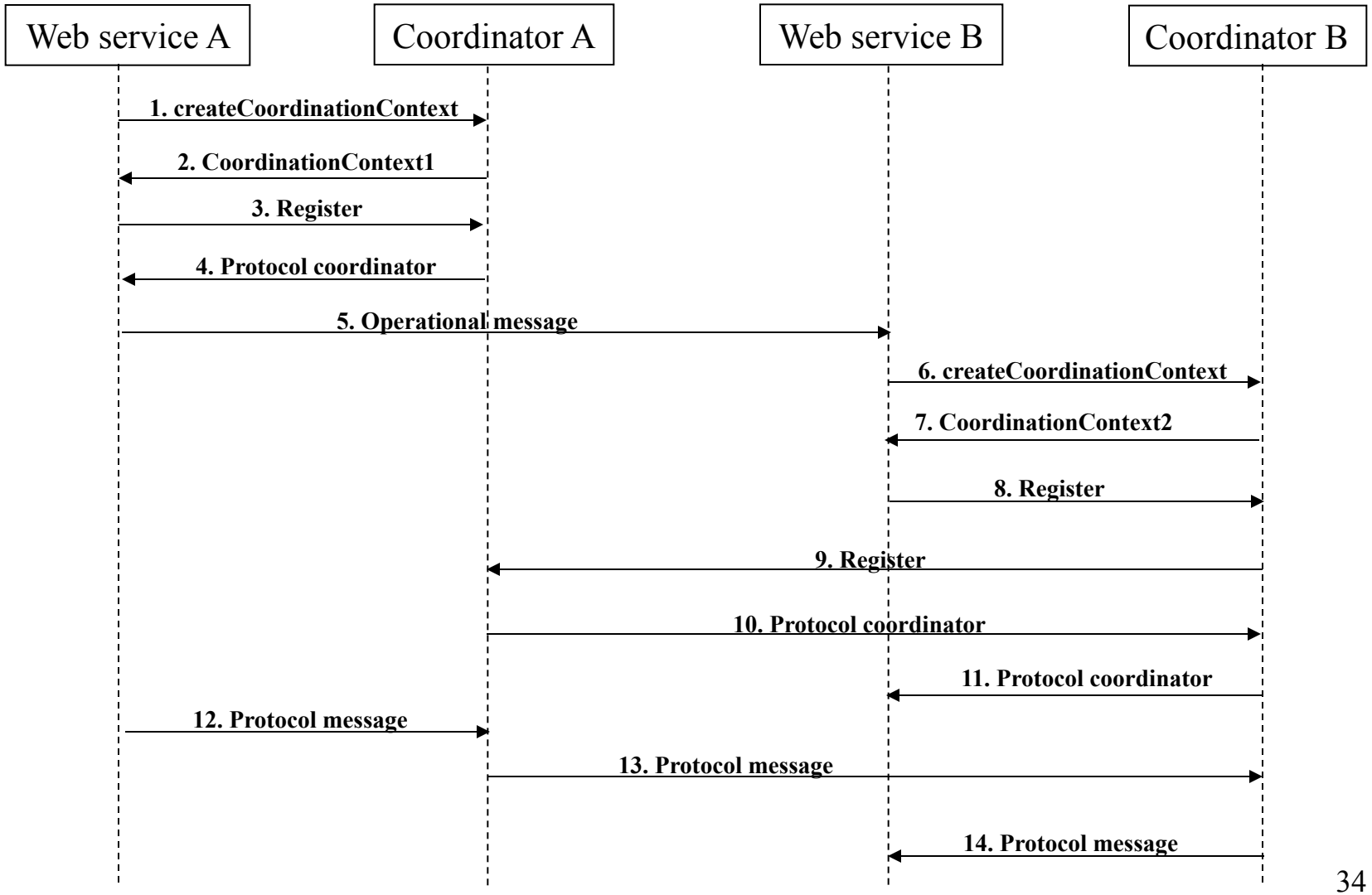
WS-Coordination (**registrationResponse**)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/
   envelope/"
   xmlns:wscoor="http://schemas.xmlsoap.org/ws/2002/09/wscoor"
   xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility">
   <soapenv:Header>
     <wscoor:CoordinationContext mustUnderstand="true">
       <wsu:Identifier>http://skatestown.com/trans2</wsu:Identifier>
         <wsu:Expires>2004-12-31T09:00:00Z</wsu:Expires>
         <wscoor:CoordinationType>
              http://schemas.xmlsoap.org/ws/2002/09/wsat
         </wscoor:CoordinationType>
         <wscoor:RegistrationService>
            <wsa:Address>http://coordinator.com/register</wsa:Address>
       </wscoor:RegistrationService>
     </wscoor:CoordinationContext>
   <soapenv:Body>
     <wscoor:RegisterResponse>
        <wscoor:CoordinationProtocolService>
           <wsa:Address>http://coordinator.com/coordinator</wsa:Address>
        </wscoor:CoordinationProtocolService>
     </wscoor:RegisterResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

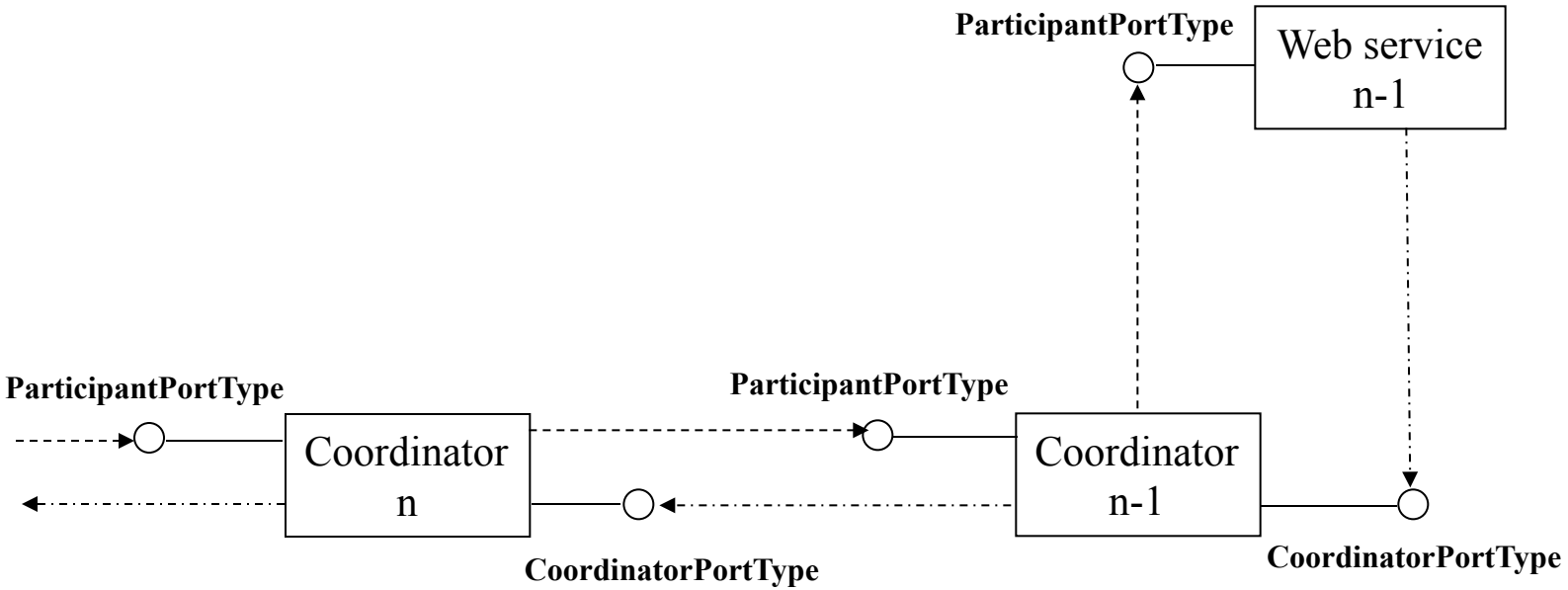
Central Coordinator



Distributed Coordinator



Distributed Coordinator



Summary of WS-Coordination

- It defines SOAP extensions for presenting coordination issues
- It defines meta-protocols for creating coordination context (**`Activation`**) and for binding coordinators and participant to each other (**`Registration`**)
- It defines basic components for implementing central or distributed coordination

Transactions

- A unit of work with a distinct start and end
- The notion of transaction was developed in the context of DBs and refers to a set of DB operations with ACID (Atomicity, Consistency, Isolation, Durability) properties
- Either all operations in the set are successfully executed of none of them is
- In the case of failure completed operations are rolled back
- Atomic transaction – viewing all work between start and end of transaction as a single unit of work
- Commit changes and unlock resources when the work is completed

Web services and transactions

- Web services are often “long-running transactions”
- ACID property can be relaxed in some cases
- Business Activity as a type of transaction for long-running transactions
- Compensation mechanism can be used
- Atomic transactions can also be needed for Web services executing short-duration transactions within limited trust domains

WS-Transaction and WS-Coordination

- WS-Transaction defines a set of protocols that require coordination among multiple parties
- It is naturally built upon WS-Coordination framework
- WS-Transaction details the structure of the coordination context and defines standard WSDL interfaces to be implemented by coordinators and participants
- Transactional semantics are achieved through a combination of WS-Coordination and WS-Transaction protocols executed with support of the coordinators

Atomic Transactions – WS-AT (operations)

Commit operation:

```
<soapenv:Body>
  <wsat:Commit/>
</soapenv:Body>
```

Rollback operation:

```
<soapenv:Body>
  <wsat:Rollback/>
</soapenv:Body>
```

Atomic Transactions (protocols)

- Participants can play different roles within each type of protocol
- For Atomic Transaction a participant can choose to register under:
  - Completion
  - Durable Two-Phase Commit
  - Volatile Two-Phase Commit

Atomic Transaction (protocols)

- Completion – participant wishes to be notified at the end of transaction
- Durable Two-Phase Commit (2PC) – completion the transaction in two phases
  - First phase – coordinator asks participant to prepare to commit the work under transaction
  - Possible returned votes in 2PC
    - Prepared – acknowledgement
    - Abort – stop transaction
    - ReadOnly – acknowledge but doesn't wish to participate in the second phase
  - Second phase – coordinator sends a commit message to each particupant
- Volatile Two-Phase Commit

wsat.wsdl

```
<?xml version="1.0" encoding="utf-8" ?>  <!-- Copyright (c) OASIS Open 2006.   -->
  <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:wsat="http://docs.oasis-open.org/ws-tx/wsat/2006/06" targetNamespace="http://docs.oasis-open.org/ws-tx/
    wsat/2006/06">
     <wsdl:types>
       <xs:schema>
          <xs:import namespace="http://docs.oasis-open.org/ws-tx/wsat/2006/06" schemaLocation="http://docs.oasis-
    open.org/ws-tx/wsat/2006/06/wstx-wsat-1.1-schema-200701.xsd" />
       </xs:schema>
     </wsdl:types>
     <wsdl:message name="Prepare">
       <wsdl:part name="parameters" element="wsat:Prepare" />
     </wsdl:message>
     <wsdl:message name="Prepared">
       <wsdl:part name="parameters" element="wsat:Prepared" />
     </wsdl:message>
     <wsdl:message name="Aborted">
       <wsdl:part name="parameters" element="wsat:Aborted" />
     </wsdl:message>
     <wsdl:message name="ReadOnly">
       <wsdl:part name="parameters" element="wsat:ReadOnly" />
     </wsdl:message>
     <wsdl:message name="Commit">
       <wsdl:part name="parameters" element="wsat:Commit" />
     </wsdl:message>
     <wsdl:message name="Rollback">
       <wsdl:part name="parameters" element="wsat:Rollback" />
     </wsdl:message>
     <wsdl:message name="Committed">
       <wsdl:part name="parameters" element="wsat:Committed" />
     </wsdl:message>
```

wsat.wsdl

```
<wsdl:portType name="CompletionCoordinatorPortType">
   <wsdl:operation name="CommitOperation">
      <wsdl:input message="wsat:Commit" />
   </wsdl:operation>
   <wsdl:operation name="RollbackOperation">
     <wsdl:input message="wsat:Rollback" />
   </wsdl:operation>
 </wsdl:portType>
 <wsdl:portType name="CoordinatorPortType">
   <wsdl:operation name="PreparedOperation">
     <wsdl:input message="wsat:Prepared" />
   </wsdl:operation>
   <wsdl:operation name="AbortedOperation">
     <wsdl:input message="wsat:Aborted" />
   </wsdl:operation>
   <wsdl:operation name="ReadOnlyOperation">
     <wsdl:input message="wsat:ReadOnly" />
   </wsdl:operation>
   <wsdl:operation name="CommittedOperation">
     <wsdl:input message="wsat:Committed" />
   </wsdl:operation>
 </wsdl:portType>
```

…

wsat.wsdl

```
  <wsdl:portType name="ParticipantPortType">
      <wsdl:operation name="PrepareOperation">
        <wsdl:input message="wsat:Prepare" />
      </wsdl:operation>
      <wsdl:operation name="CommitOperation">
        <wsdl:input message="wsat:Commit" />
      </wsdl:operation>
      <wsdl:operation name="RollbackOperation">
        <wsdl:input message="wsat:Rollback" />
      </wsdl:operation>
   </wsdl:portType>
</wsdl:definitions>
```

Commitment (registration)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
   soap/envelope/"
   xmlns:wscoor="http://schemas.xmlsoap.org/ws/2002/09/wscoor"
   xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility">
   <soapenv:Header>. . .
   <soapenv:Body>
     <wscoor:Register>
      <wscoor:ProtocolIdentifier>
          http://schemas.xmlsoap.org/ws/2003/09/wsat#Durable2PC
       </wscoor:ProtocolIdentifier>
      <wscoor:ParticipantProtoolService>
            <wsa:Address>
               http://Wheels.com/2pcservice
            </wsa:Address>
      </wscoor:ParticipantProtocolService>
     </wscoor:Register>
   </soapenv:Body>
</soapenv:Envelope>
```

Commitment (prepare)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/
   envelope/"
   xmlns:wscoor="http://schemas.xmlsoap.org/ws/2002/09/wscoor"
   xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility">
   <soapenv:Header>
        <wscoor:CoordinationContext mustUnderstand="true">
          <wsu:Identifier>
               http://skatestown.com/trans2
          </wsu:Identifier>
          <wsu:Expires>2004-12-31T09:00:00Z</wsu:Expires>
          <wscoor:CoordinationType>
               http://schemas.xmlsoap.org/ws/2002/09/wsat
          </wscoor:CoordinationType>
          <wscoor:RegistrationService>
            <wsa:Address>http://coordinator.com/register</wsa:Address>
          </wscoor:RegistrationService>
        </wscoor:CreateCoordinationContext>
   </soapenv:Header>
   <soapenv:Body>
        <wsat:Prepare/>
   </soapenv:Body>
</soapenv:Envelope>
```

Commitment (prepared)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/
   envelope/"
   xmlns:wscoor="http://schemas.xmlsoap.org/ws/2002/09/wscoor"
   xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility">
   <soapenv:Header>
        <wscoor:CoordinationContext mustUnderstand="true">
          <wsu:Identifier>
               http://skatestown.com/trans2
          </wsu:Identifier>
          <wsu:Expires>2004-12-31T09:00:00Z</wsu:Expires>
          <wscoor:CoordinationType>
               http://schemas.xmlsoap.org/ws/2002/09/wsat
          </wscoor:CoordinationType>
          <wscoor:RegistrationService>
            <wsa:Address>
               http://coordinator.com/register
            </wsa:Address>
          </wscoor:RegistrationService>
        </wscoor:CreateCoordinationContext>
   </soapenv:Header>
   <soapenv:Body>
        <wsat:Prepared/>
   </soapenv:Body>
</soapenv:Envelope>
```

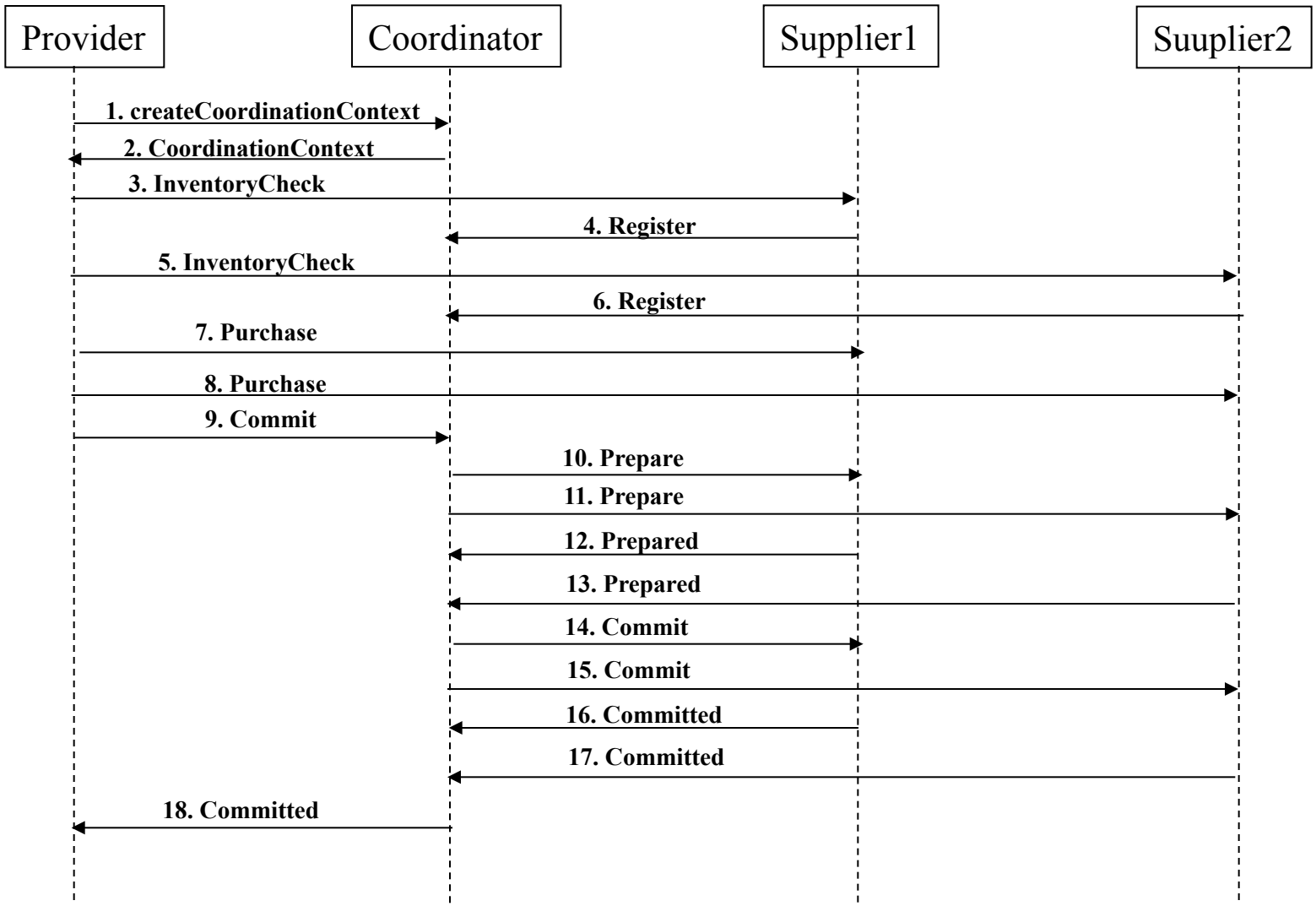
Commitment (commit)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/
   envelope/"
   xmlns:wscoor="http://schemas.xmlsoap.org/ws/2002/09/wscoor"
   xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility">
   <soapenv:Header>
        <wscoor:CoordinationContext mustUnderstand="true">
            <wsu:Identifier>
                http://skatestown.com/trans2
            </wsu:Identifier>
            <wsu:Expires>2004-12-31T09:00:00Z</wsu:Expires>
            <wscoor:CoordinationType>
                http://schemas.xmlsoap.org/ws/2002/09/wsat
            </wscoor:CoordinationType>
            <wscoor:RegistrationService>
                <wsa:Address>
                    http://coordinator.com/register
                </wsa:Address>
            </wscoor:RegistrationService>
        </wscoor:CreateCoordinationContext>
   </soapenv:Header>
   <soapenv:Body>
        <wsat:Commit/>
   </soapenv:Body>
</soapenv:Envelope>
```

Commitment (committed)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/
   envelope/"
   xmlns:wscoor="http://schemas.xmlsoap.org/ws/2002/09/wscoor"
   xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility">
   <soapenv:Header>
     <wscoor:CoordinationContext mustUnderstand="true">
        <wsu:Identifier>
               http://skatestown.com/trans2
        </wsu:Identifier>
        <wsu:Expires>2004-12-31T09:00:00Z</wsu:Expires>
        <wscoor:CoordinationType>
               http://schemas.xmlsoap.org/ws/2002/09/wsat
        </wscoor:CoordinationType>
        <wscoor:RegistrationService>
          <wsa:Address>http://coordinator.com/register</wsa:Address>
        </wscoor:RegistrationService>
     </wscoor:CreateCoordinationContext>
   </soapenv:Header>
   <soapenv:Body>
        <wsat:Committed/>
   </soapenv:Body>
</soapenv:Envelope>
```

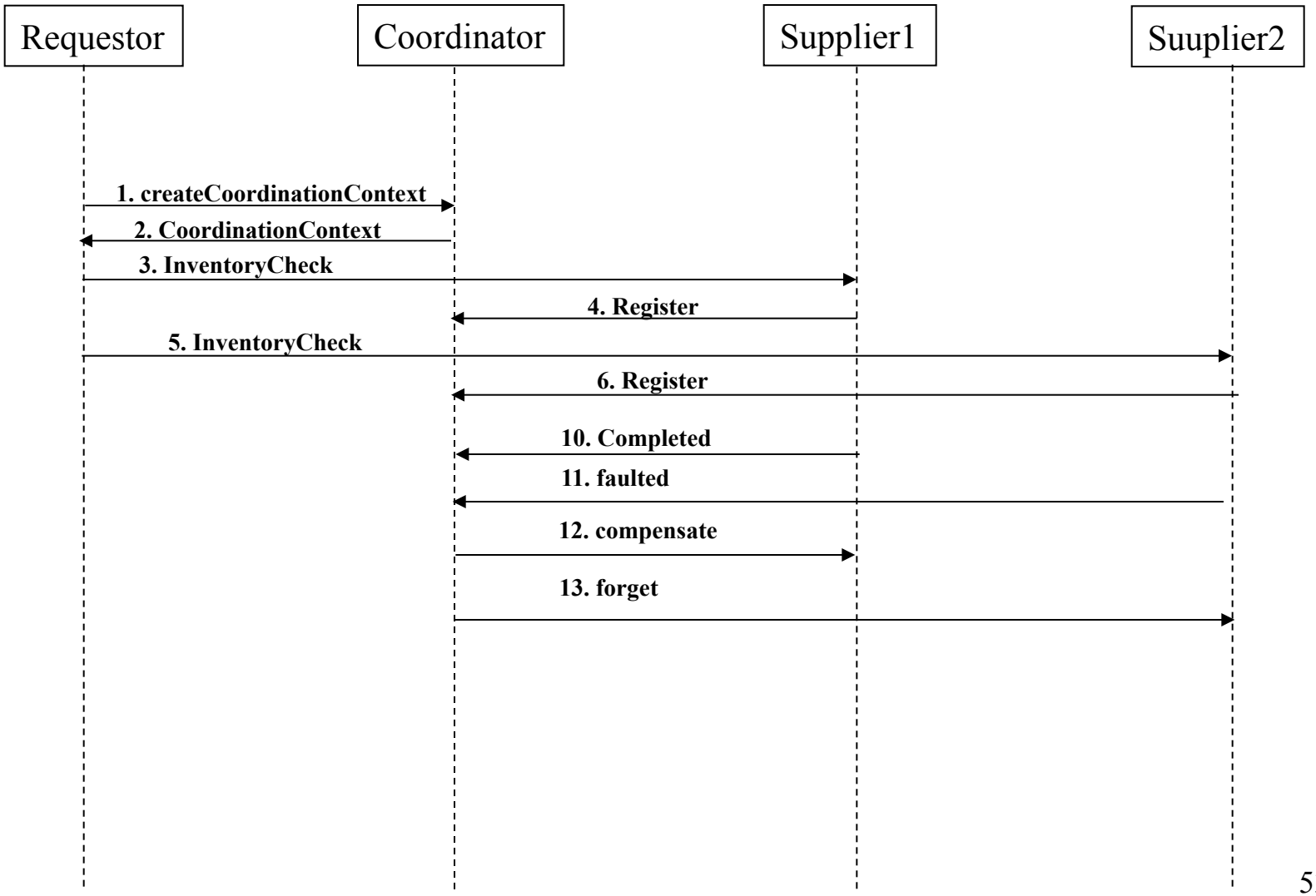
Complete AT example



Business Activity

- Long-running transaction, allows do not lock resources
- Each individual Web service call is conceptually independent and complete
- Compensation mechanism
- Operations
  - **`Complete`** instead of **`Commit`**
  - **`Compensate`** instead of **`RollBack`**

Business Activity



Next lecture

- Web Services Composition

Text-book **Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI, 2nd Edition**

Chapter 12