

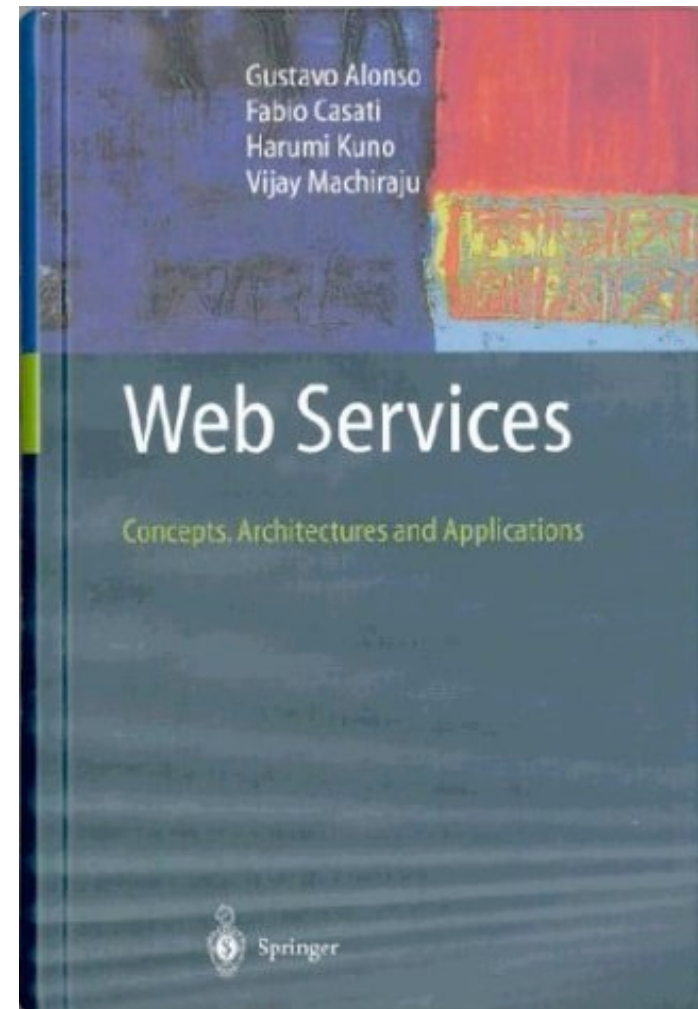
Standard Software for Enterprise Resource Planning

Lecturer: Prof. Dr. Ralf Möller
Lab classes: Rainer Marrone, Michael Wessel

Lecture: Thursdays (90 minutes)
Lab classes: Fridays (60 minutes)

Prerequisite:
Lecture on ECommerce

This lecture is based on:





ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Web Services: Concepts, Architecture and Applications

Part 8: Web services extensions

Gustavo Alonso and Cesare Pautasso
Computer Science Department
ETH Zürich
alonso@inf.ethz.ch
<http://www.inf.ethz.ch/~alonso>

Extending basic Web services



- ❑ SOAP, WSDL and UDDI provide the basic infrastructure for Web services. In real applications, however, additional functionality is needed
- ❑ Currently, there are literally several dozen proposals on different extensions to Web services covering almost every conceivable aspect. These proposals are evolving very quickly and disappear, merge, or change too rapidly to really influence actual systems
- ❑ Looking at these extensions provides a good idea of the mind set of some of the vendors pushing for Web services and where they hope to get an advantage over competitors
- ❑ As examples of some of these existing proposals, we will briefly review:
 - WS-Reliability
 - WS-Coordination
 - WS-AtomicTransactions and WS-BusinessActivity
 - WS-Security

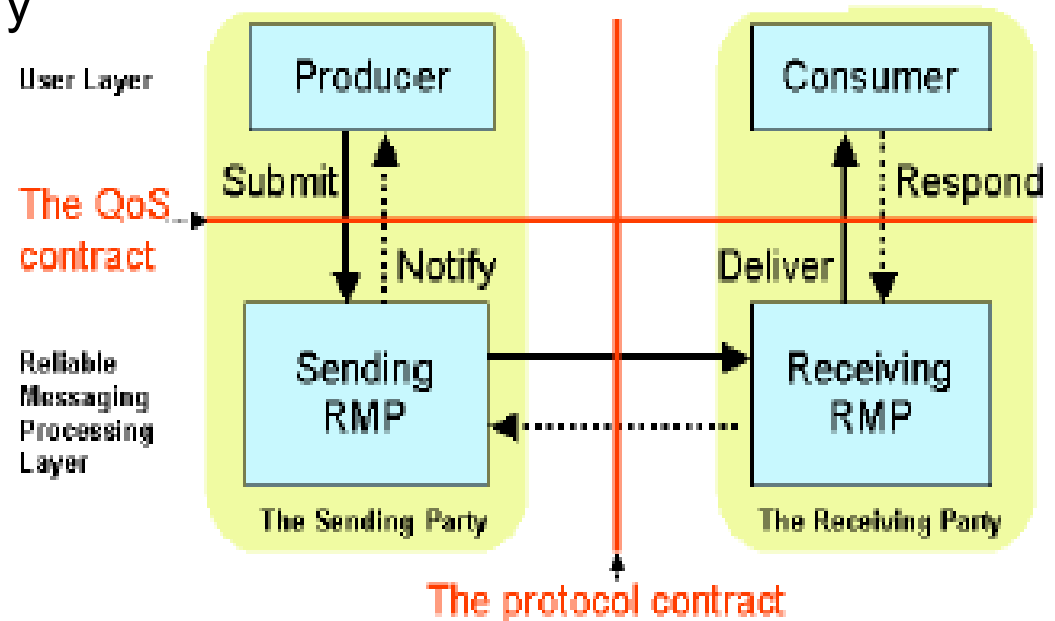
WS-Reliability

WS-Reliability

WS-reliability July 2004

- ❑ WS-Reliability extends SOAP with guarantees on the delivery of messages:

- At least once (guaranteed delivery)
- At most once (duplicate elimination)
- Exactly once (guaranteed delivery and duplicate elimination)
- Ordered delivered

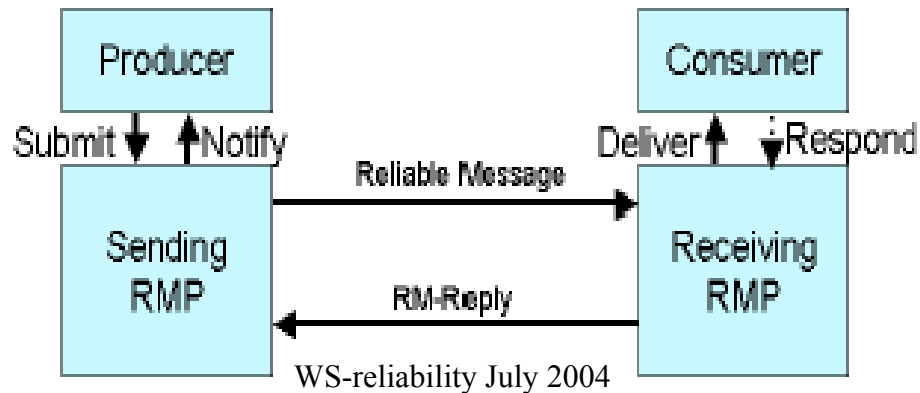


- ❑ WS-Reliability has two sides:
 - Protocol (with message headers, and message exchanges)
 - QoS (guarantees by each party on what to do with the messages as they arrive)

- ❑ WS-reliability assumes point-to-point communication (does not cover routing) and mirrors the guarantees offered in RPC systems

Basics of WS-Reliability

□ Basic notation



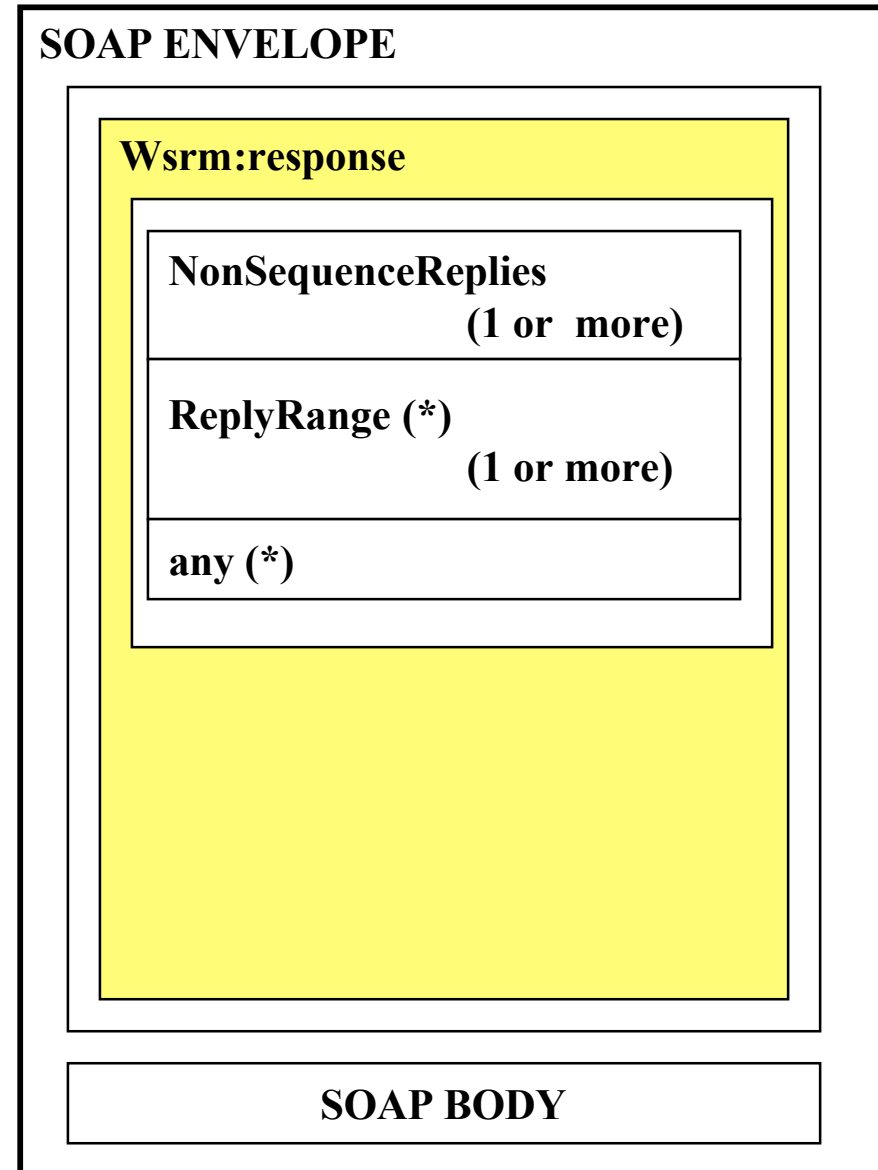
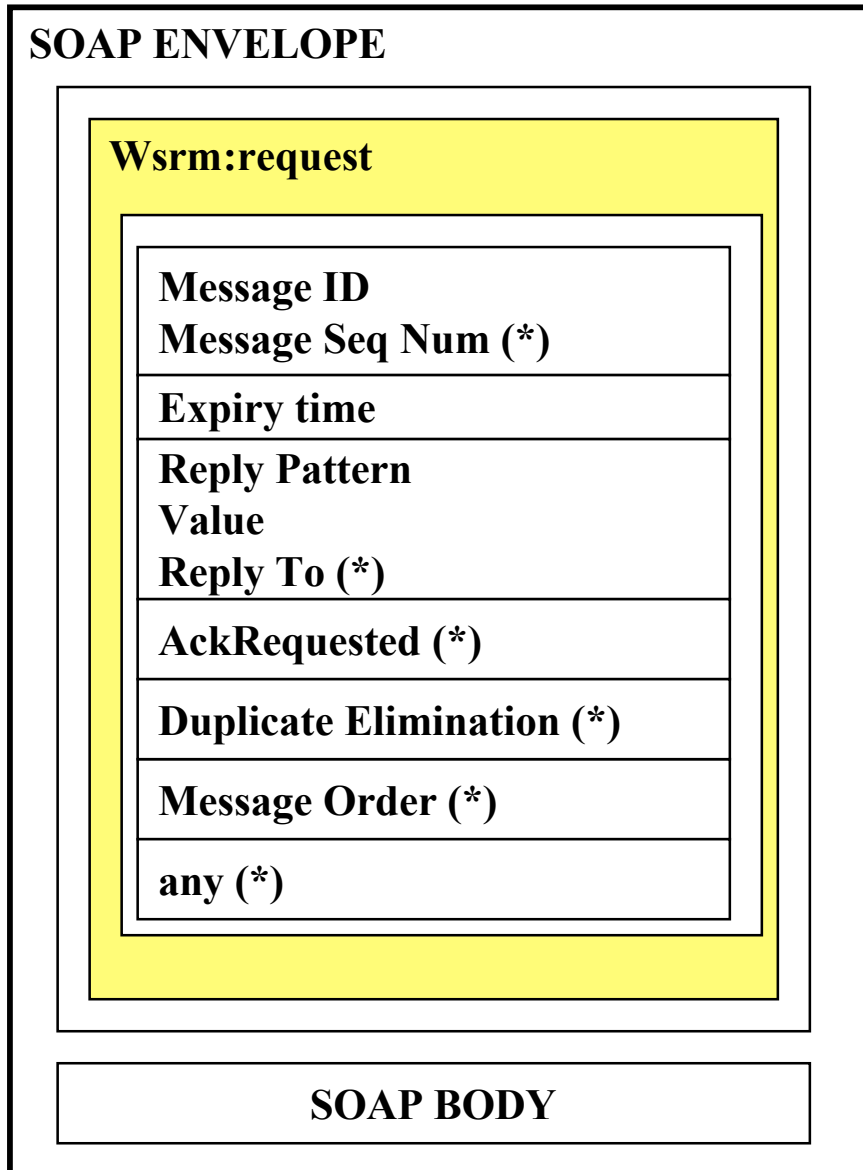
□ WS-Reliability extends SOAP:

- Uses SOAP headers
- Uses SOAP message exchange patterns (one way and request-response)

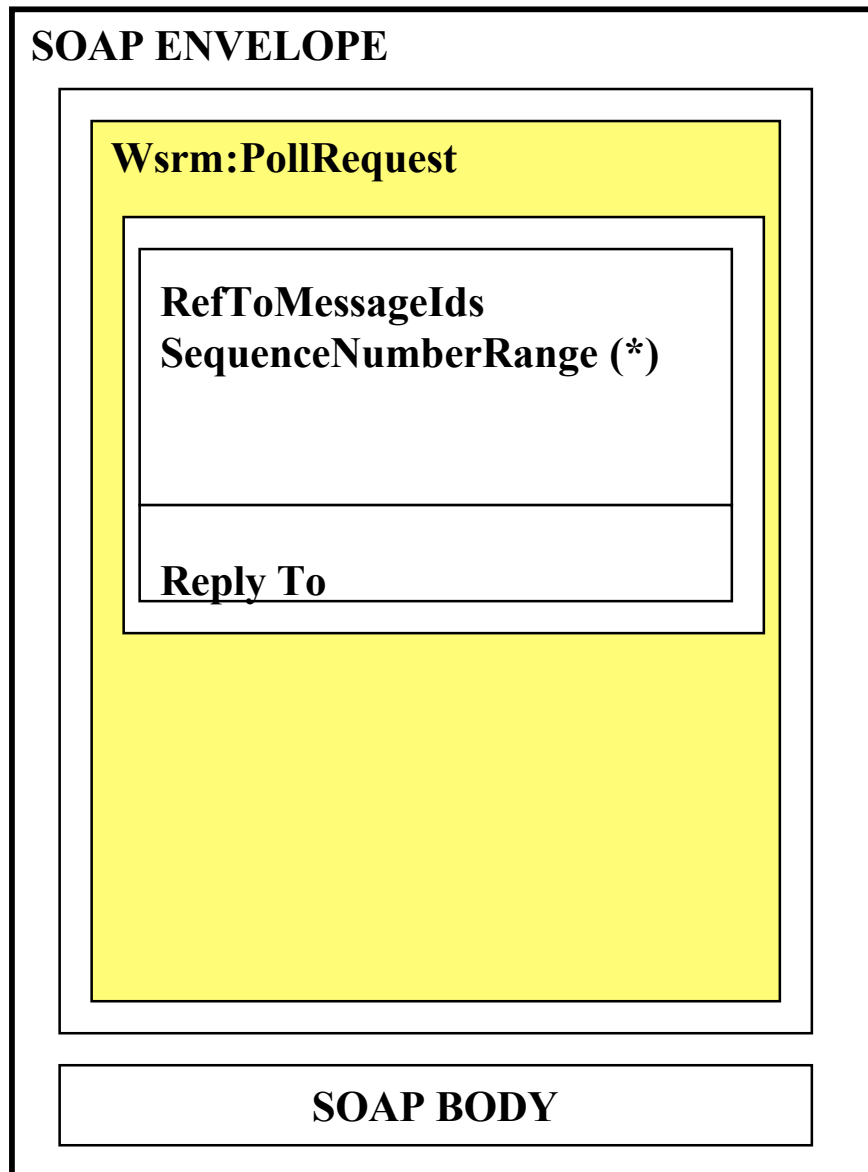
□ Message patterns:

- Request response (the information about reliability is attached to the SOAP request, the RM-reply is attached to the SOAP response)
- Callback (the information about reliability is attached to the SOAP request, the Reply is attached to the SOAP response of another invocation or to a specific message)
- Poll (the information about reliability is attached to the SOAP request, then the sender must issue a Poll message, and then the receiver responds with the RM-reply; the Poll message can be synchronous or asynchronous)

WS-Reliability information



WS-Reliability information



- ❑ Every reliable message must have:
 - A message ID that identifies it as a unique member of a group of messages
- ❑ A group of messages is a sequence of messages sent from the sender to the receiver
- ❑ The message identifier includes the group identifier and a sequence number within the group

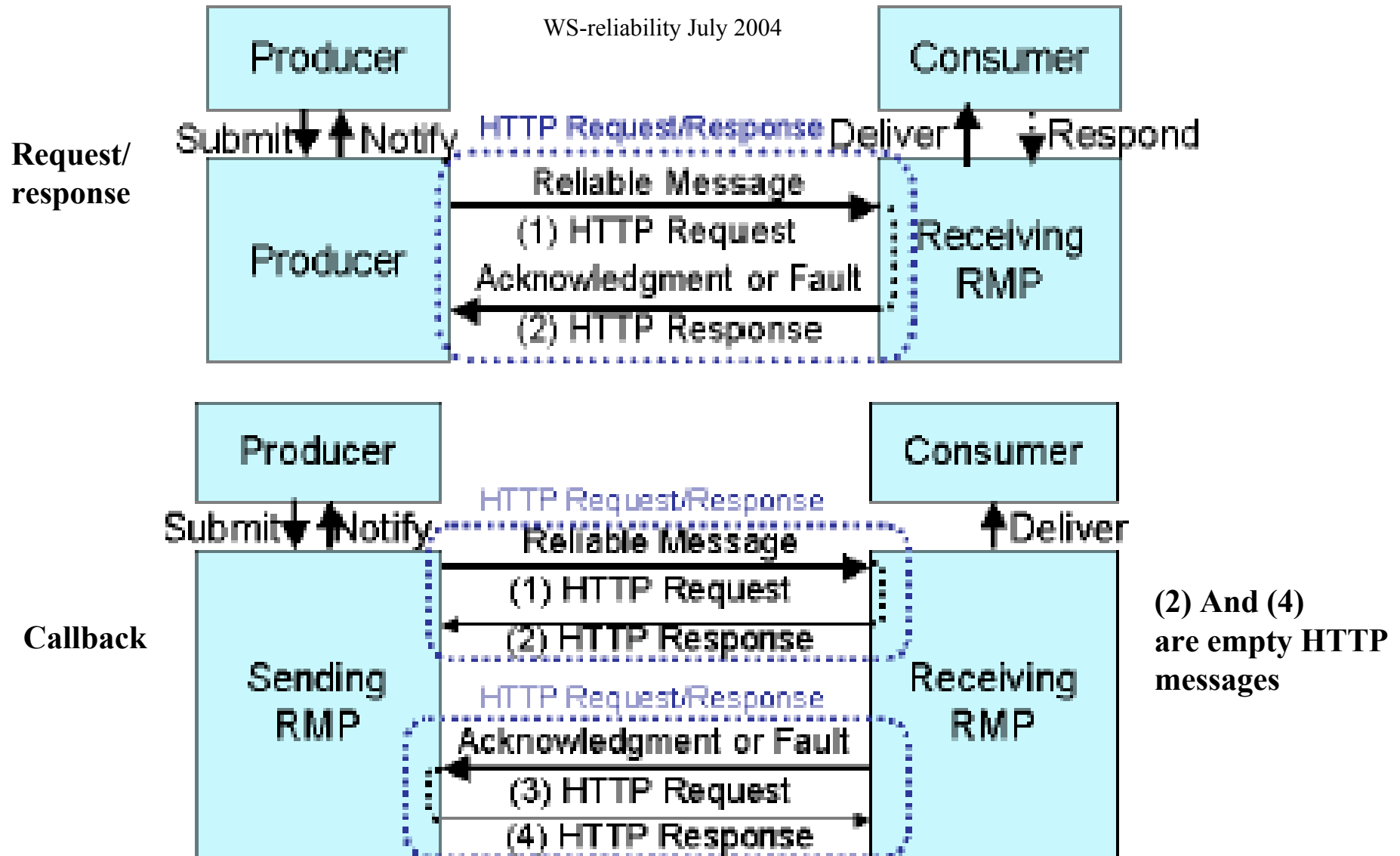
RM Agreement

Name	Value	Definition
GuaranteedDelivery	enabled/disabled	For setting Guaranteed Delivery.
NoDuplicateDelivery	enabled/disabled	For setting message delivery without duplicates, or Duplicate Elimination.
OrderedDelivery	enabled/disabled	For setting Guaranteed Message Ordering.
GroupMaxIdleDuration	number of seconds	For setting the elapsed time limit from the last message sent or received in a group, after which the group can be terminated. The value MUST NOT be zero or smaller.
GroupExpiryTime	date/time	For setting the date and time after which the group can be terminated.
ExpiryTime	date/time	For setting the date and time after which a message must not be delivered to the
ReplyPattern	"Response", "Callback", "Poll"	For setting the mode of response for Acknowledgments or Faults.

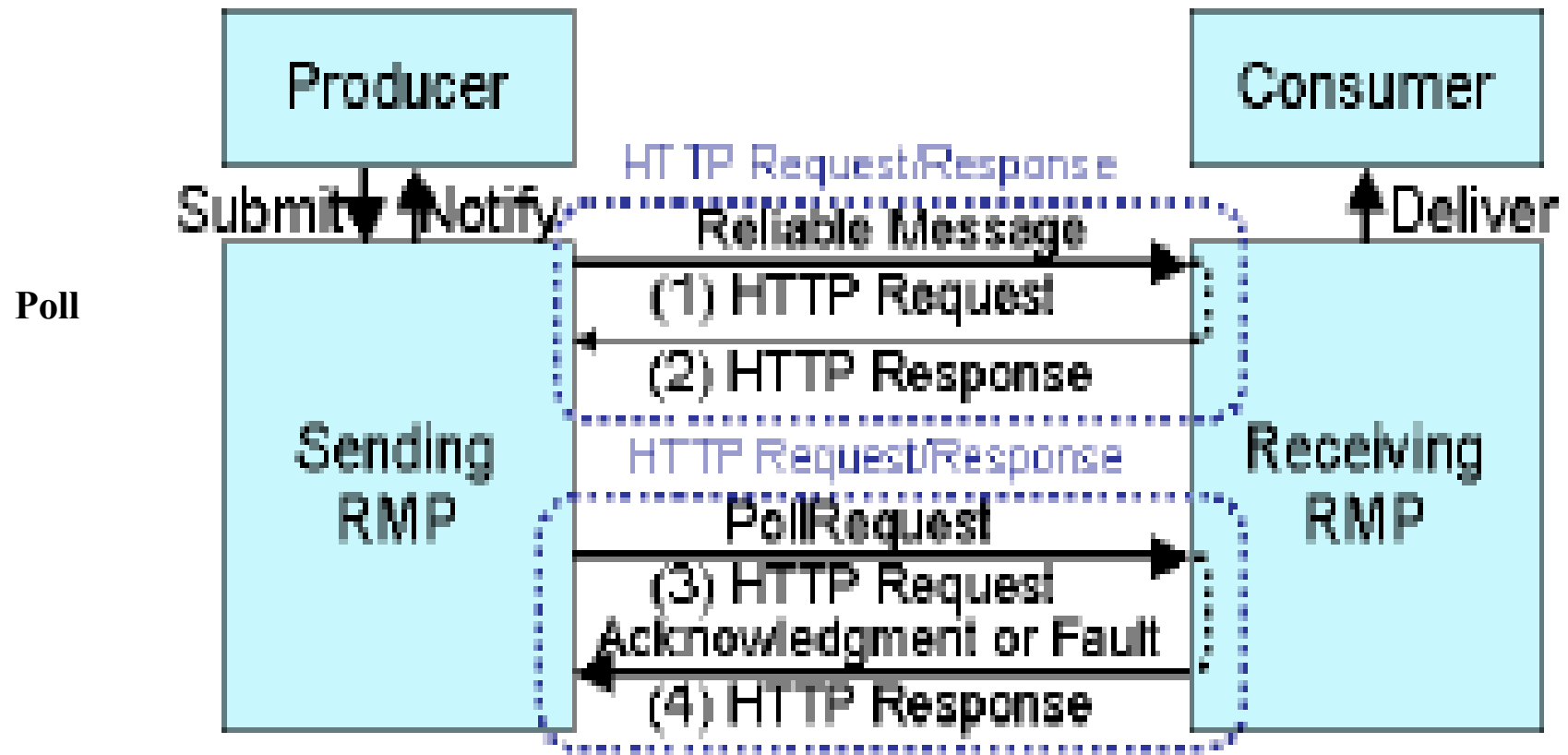
Main features

- ❑ Guaranteed delivery
 - Either the payload is delivered or the sender is notified of the failure
 - It mandates that the receiver produces a RM-reply (ACK) for the message
- ❑ Duplicate elimination
 - A message with a given message ID is delivered at most once
 - It implies that the receiver remembers the IDs of the messages and checks each message to see if it has already been delivered
 - The sender must make sure the IDs are correctly assigned to messages (same message, same id)
- ❑ Guaranteed Message ordering:
 - Messages send within a group are delivered in the sequence that they were sent
 - It implies the receiver keeps track of the sequence, requests the resend of missing messages in the sequence, and delivers the messages in the sequence order

HTTP Binding



HTTP binding

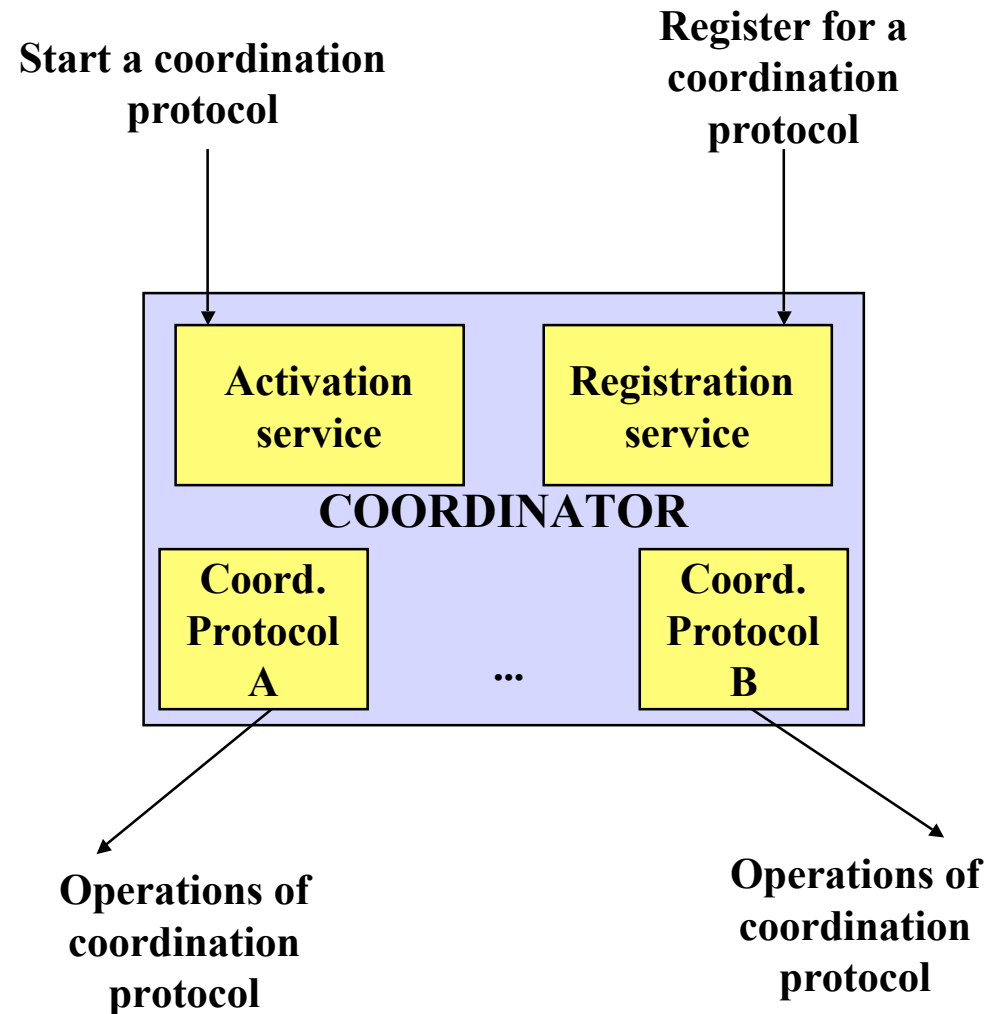


(2) is an empty HTTP message

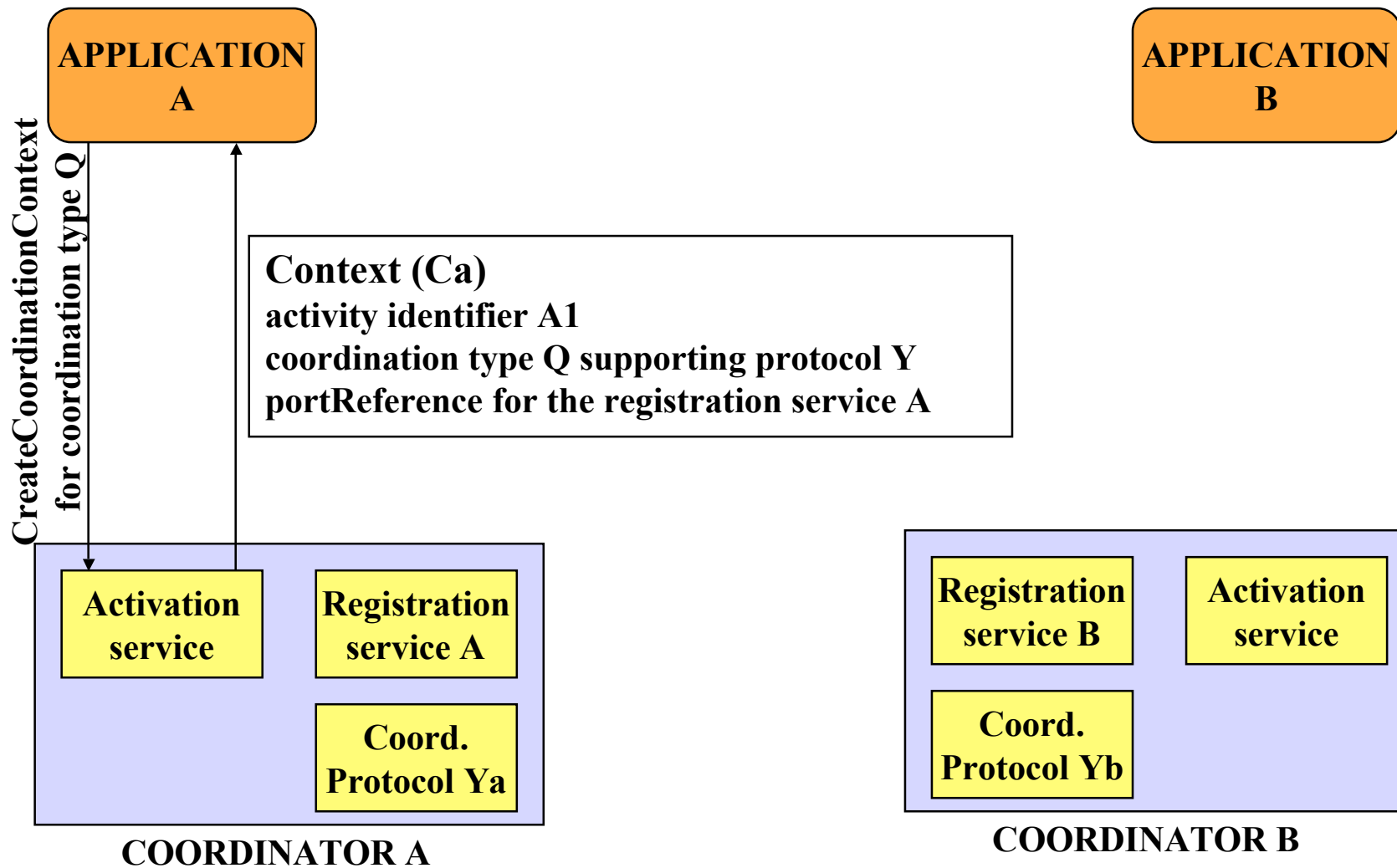
WS-Coordination

WS-Coordination

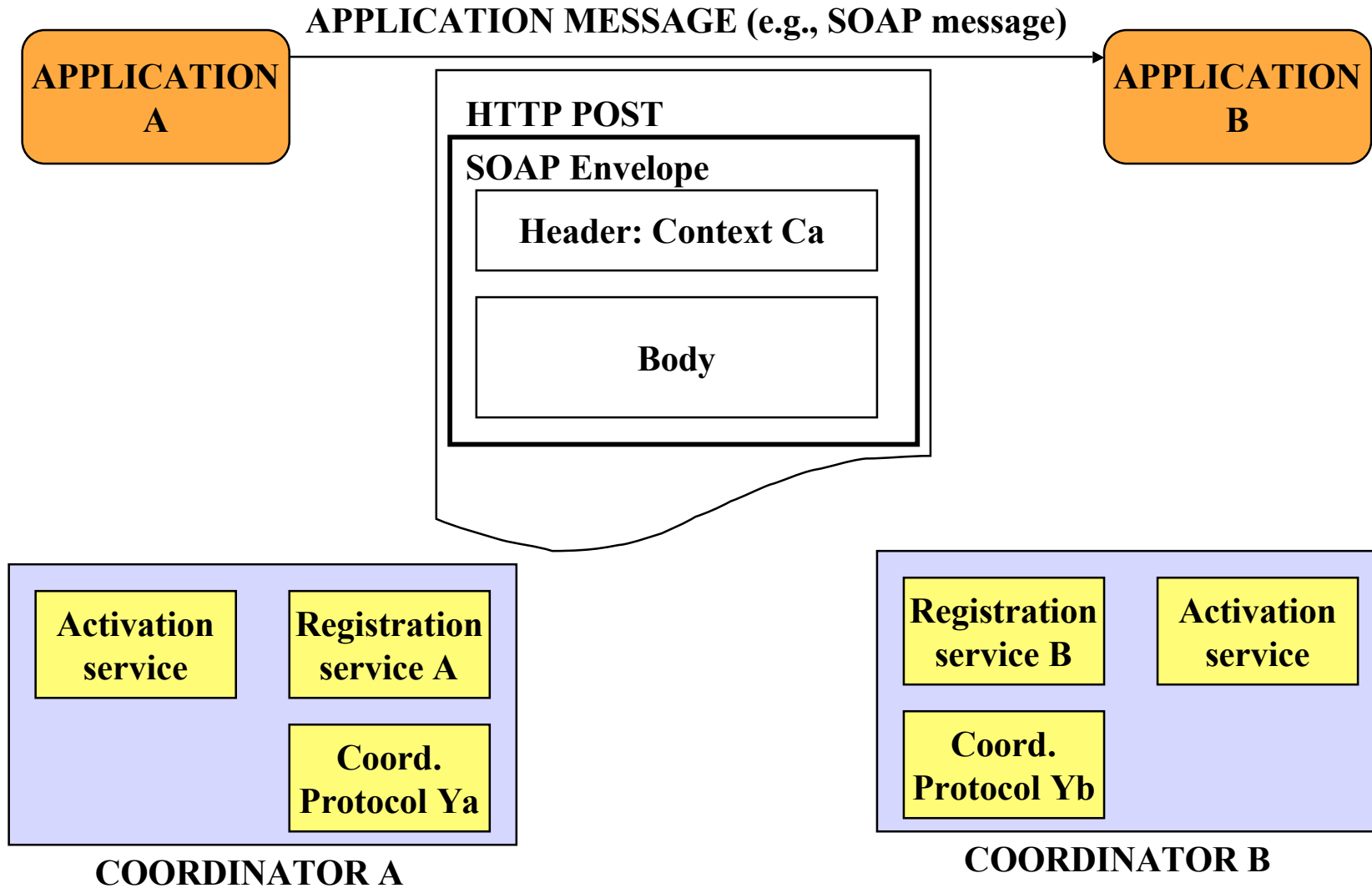
- ❑ WS-Coordination is intended as a generic infrastructure to implement coordination protocols between Web services
- ❑ Its main goal is to serve as a generic platform for implementing advanced transaction models but it can be used to implement a wide variety of coordination protocols between services (including some forms of conversations)
- ❑ WS-Coordination encompasses a set of behaviors and APIs which enable a module to extend Web services with coordination capabilities
- ❑ It mirrors the behavior of transactional services in conventional middleware platforms



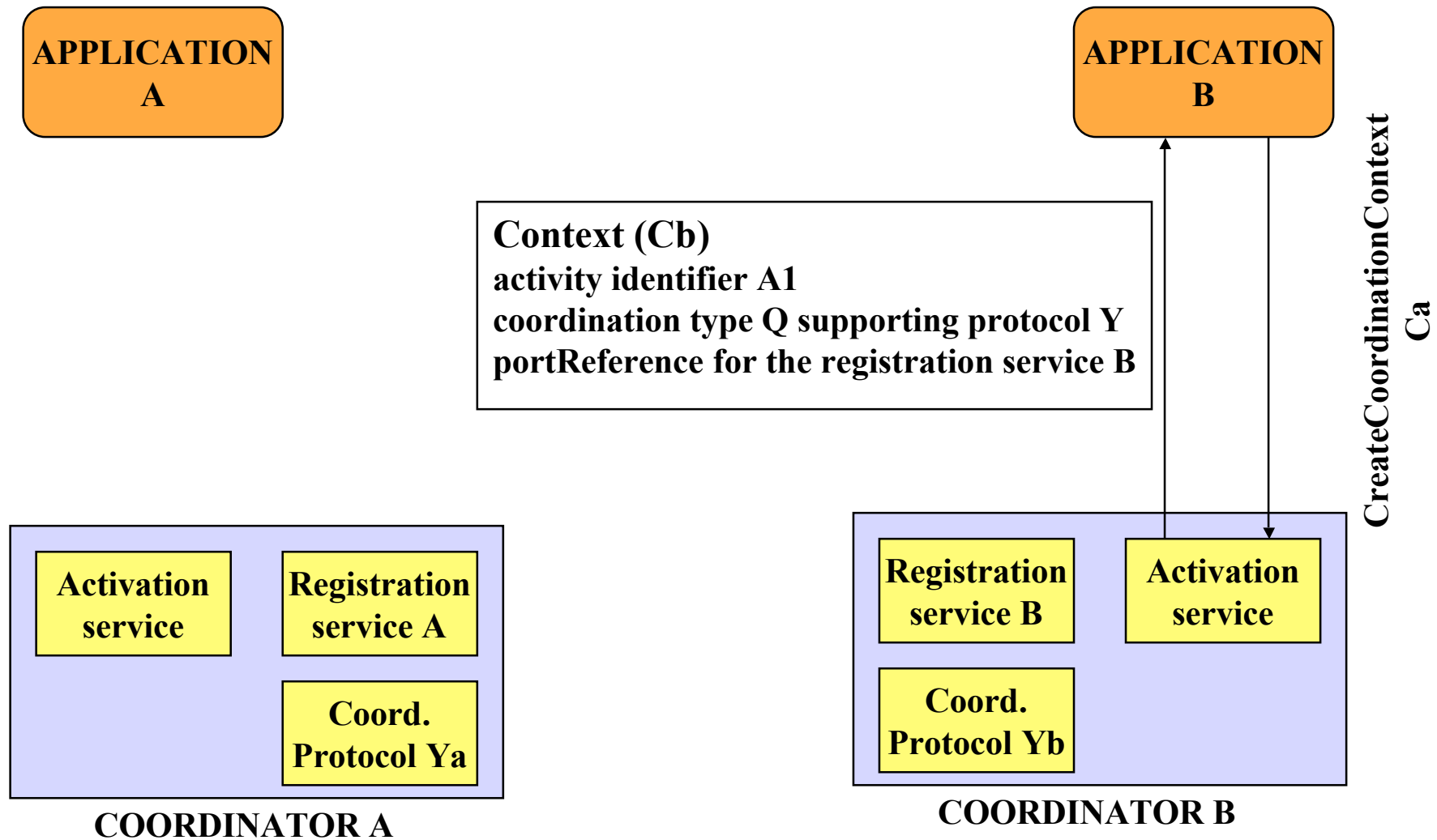
Basics of WS-Coordination (1)



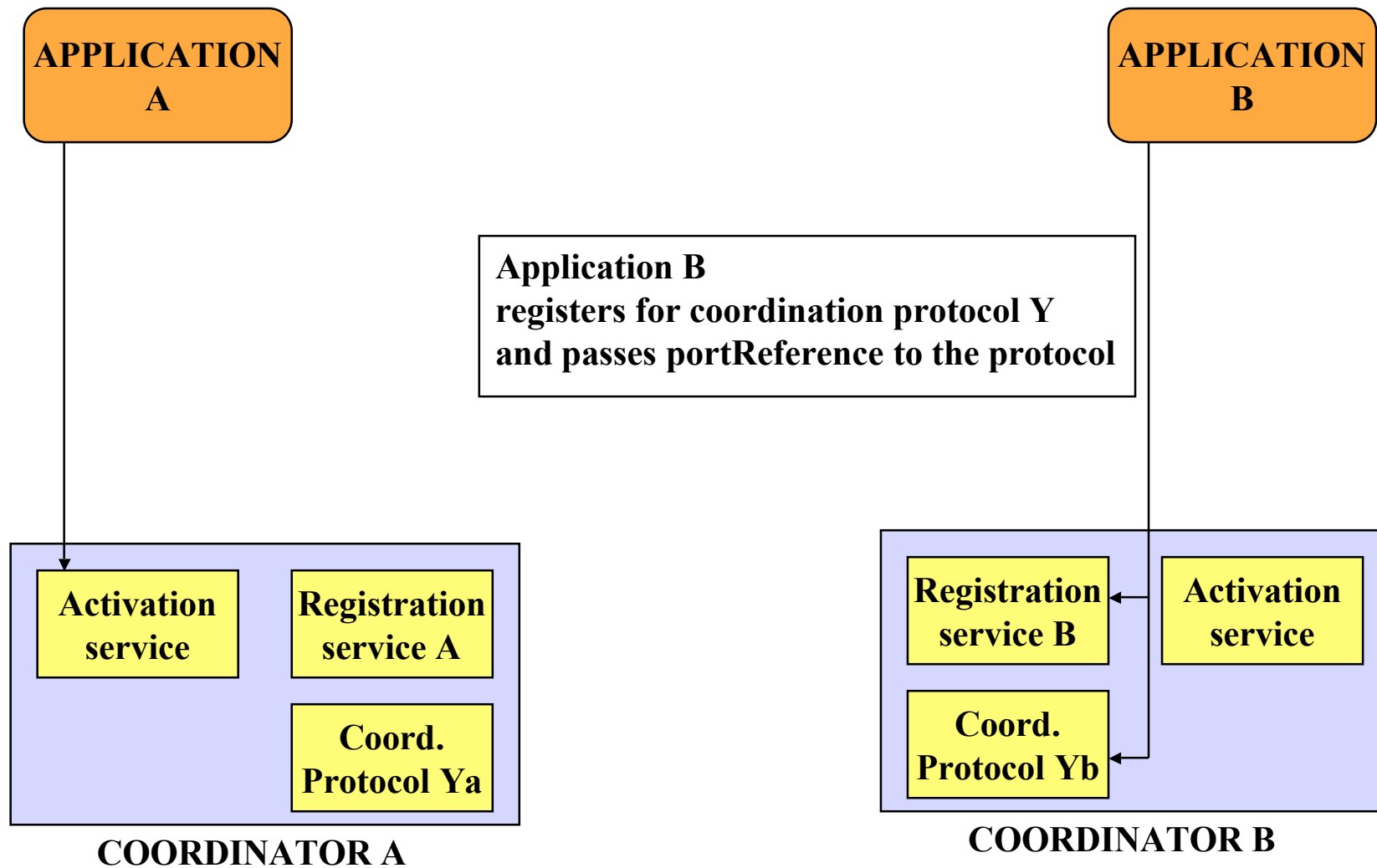
Basics of WS-Coordination (2)



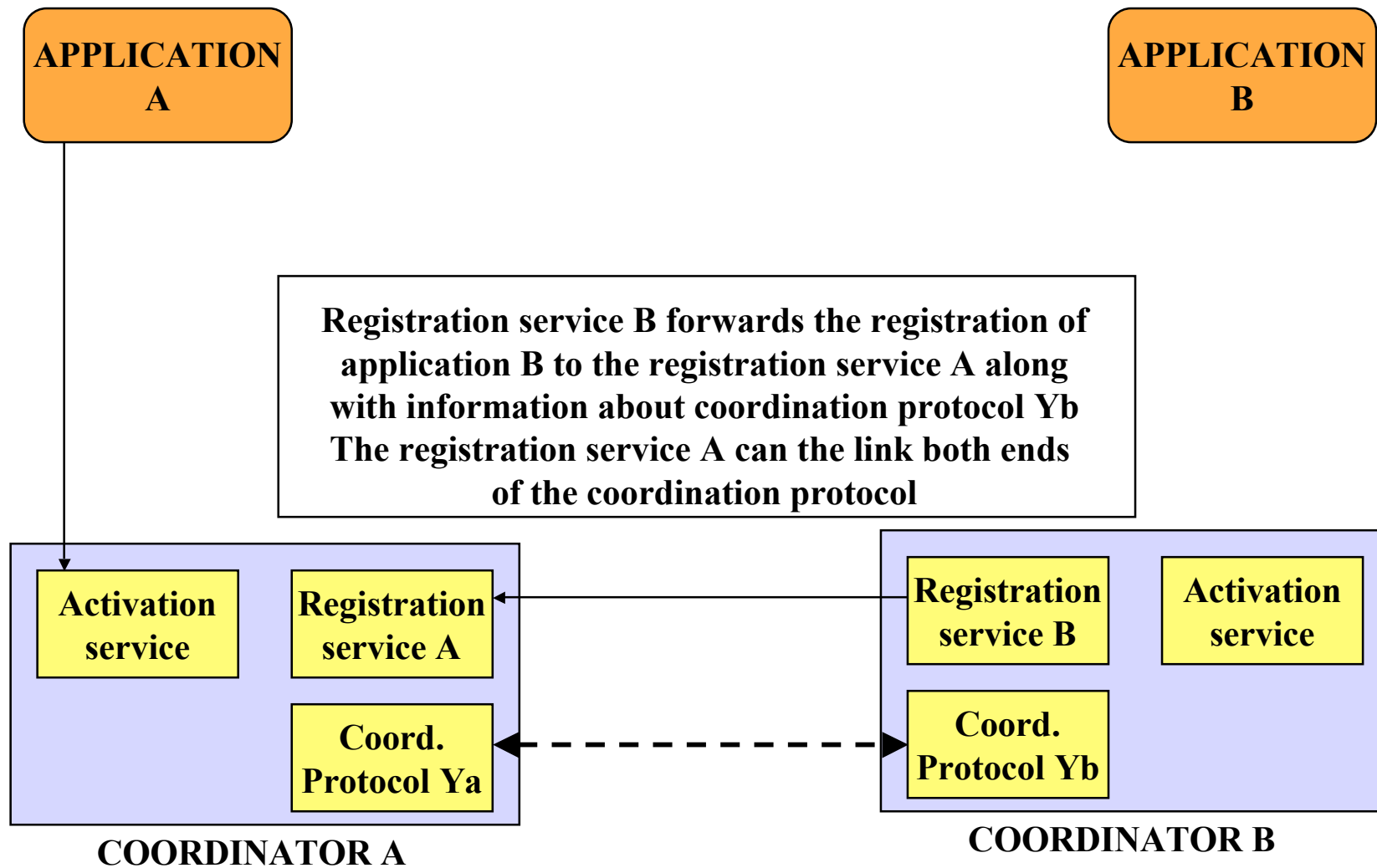
Basics of WS-Coordination (3)



Basics of WS-Coordination (4)



Basics of WS-Coordination (5)



Messages and interfaces



- ❑ The coordinator defined by WS-Coordination is described using WSDL and offers a number of services to the application.
- ❑ The application accesses these services by sending, e.g., SOAP messages to the coordinator which then responds with new SOAP messages. Interactions with the protocol would then also be in terms of SOAP messages (but other protocols are possible, one simply needs to provide alternative bindings for the coordinator services)
- ❑ The example shown considers the case where application B decides to use its own coordinator. Application B could also decide to use the same coordinator as application A but in the cases where A and B are independent services, provided by different organizations, a coordinator per application makes more sense
- ❑ WS-Coordination is an attempt at standardizing:
 - the use of SOAP headers for coordination protocols
 - the basic operations for most coordination protocols
 - the functionality a Web service middleware platform must support for allowing coordination protocols to be implemented

WS-Coordinator in XML



ACTIVATION SERVICE:

```
<wsdl:portType name="ActivationCoordinatorPortType">  
  <wsdl:operation name="CreateCoordinationContext">  
    <wsdl:input message="wscoor:CreateCoordinationContext"/>  
  </wsdl:operation>  
</wsdl:portType>
```

RESPONSE ACTIVATION SERVICE

```
<wsdl:portType name="ActivationRequesterPortType">  
  <wsdl:operation name="CreateCoordinationContextResponse">  
    <wsdl:input message="wscoor:CreateCoordinationContextResponse"/>  
  </wsdl:operation>  
  <wsdl:operation name="Error">  
    <wsdl:input message="wscoor:Error"/>  
  </wsdl:operation>  
</wsdl:portType>
```

From Web Services Coordination (WS-Coordination) 9 August 2002

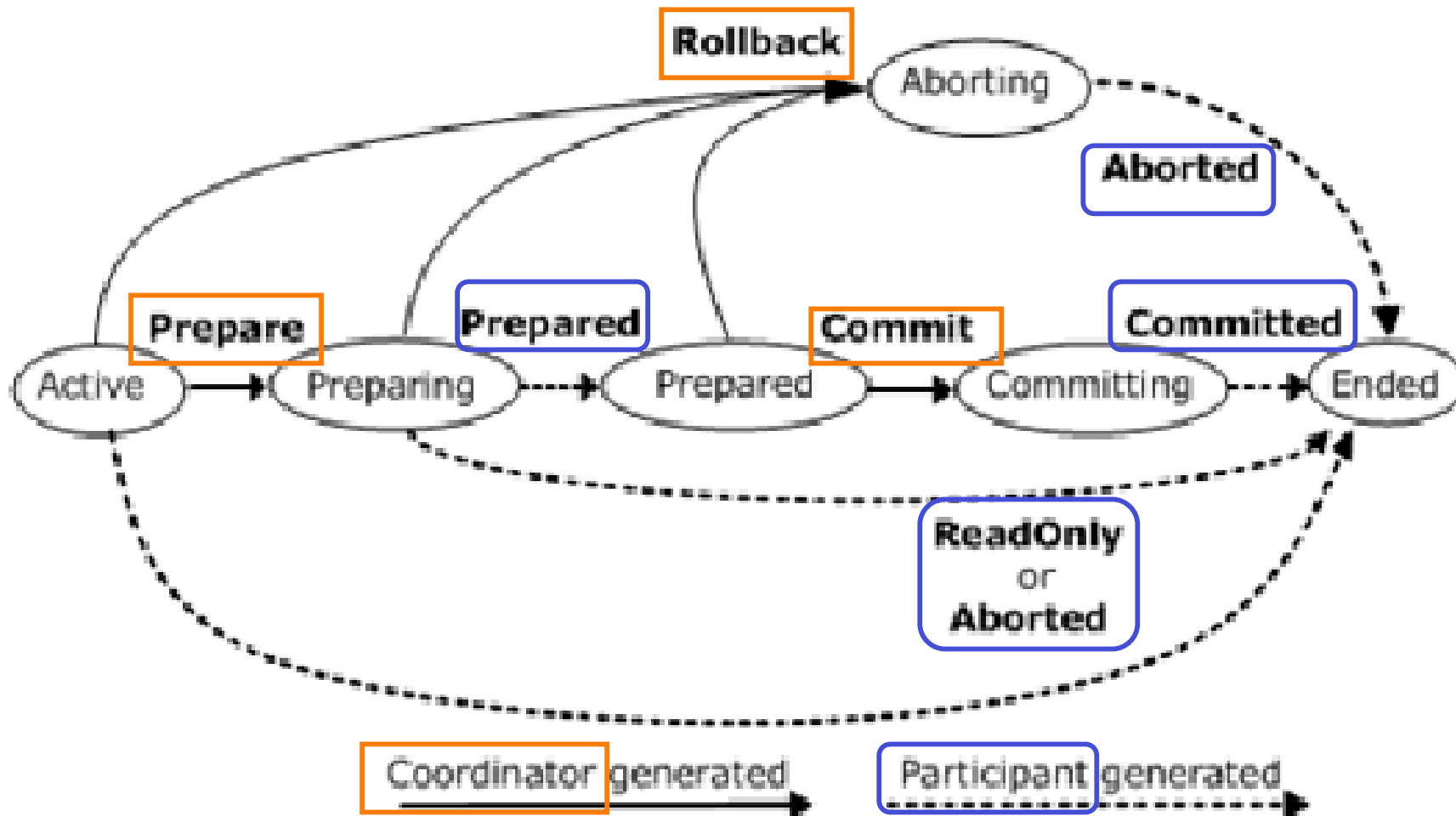
WS-AtomicTransactions

WS-Transactions



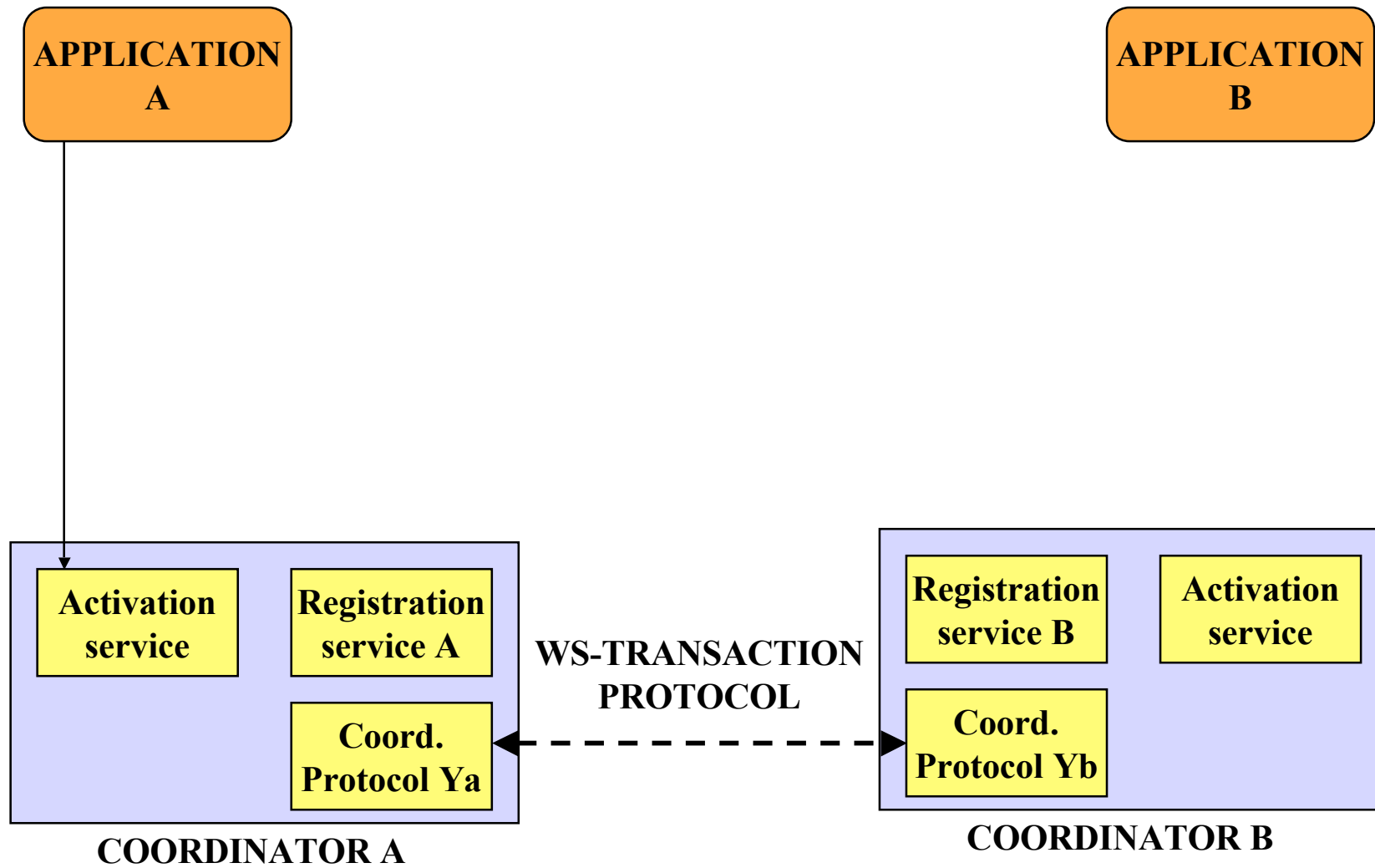
- ❑ WS-Transactions builds directly upon WS-Coordination to specify different coordination protocols related to transaction processing
 - atomic transactions (governed by 2 Phase Commit)
 - business activities (transactional but based on compensation activities)
 - business agreement
 - business agreement with complete
- ❑ WS-Transactions specify the coordination protocol to be used as part of WS-Coordination. The specification deals with the nature of the interaction, the syntax and semantics of the messages to be exchanged as part of the coordination protocol, and the expected responses of all participants involved
- ❑ Like WS-Coordination, WS-Transactions follows very closely the transactional model found in conventional middleware platforms

Coordination protocol for 2PC



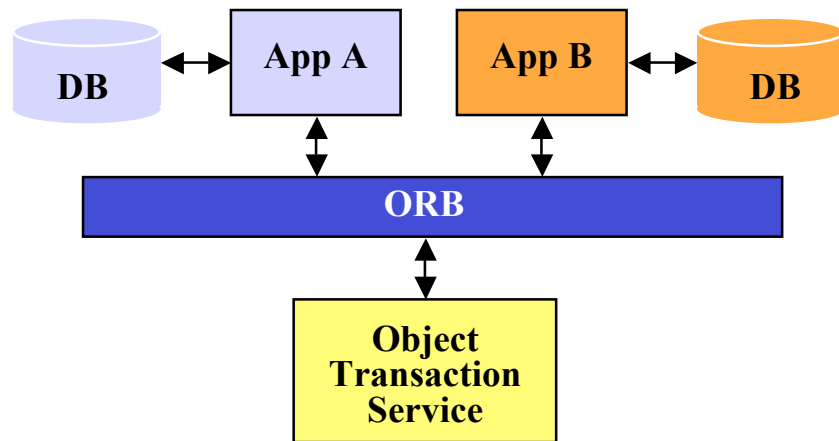
From WS-AtomicTransactions September 2003

WS-Transactions

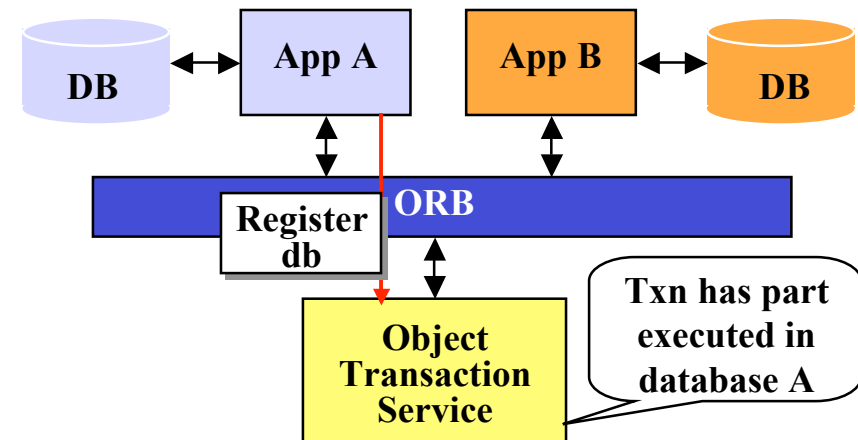


CORBA transactions (1)

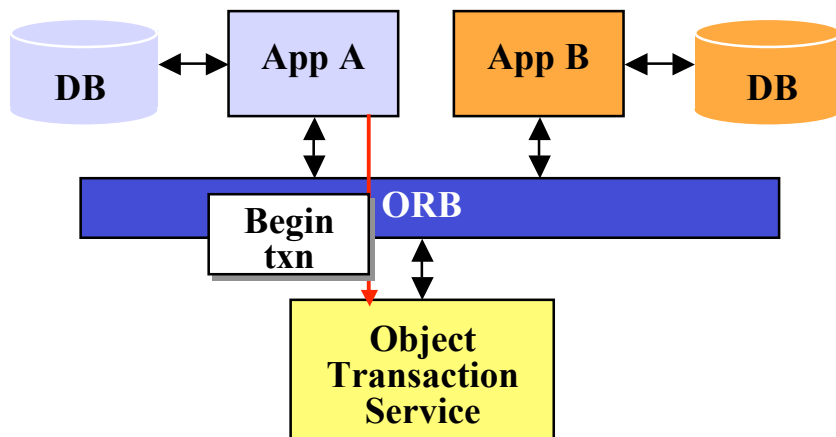
1) Assume App A wants to update databases A and B



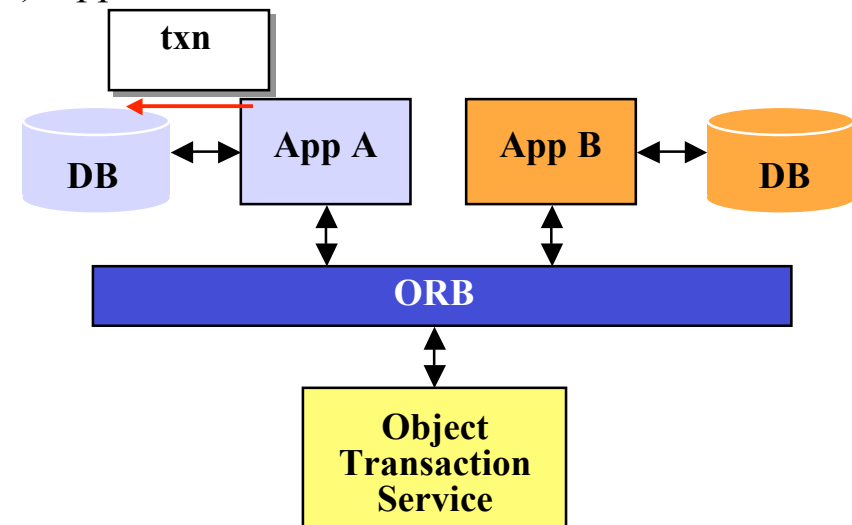
3) App A registers the database for that transaction



2) App A obtains a txn identifier for the operation

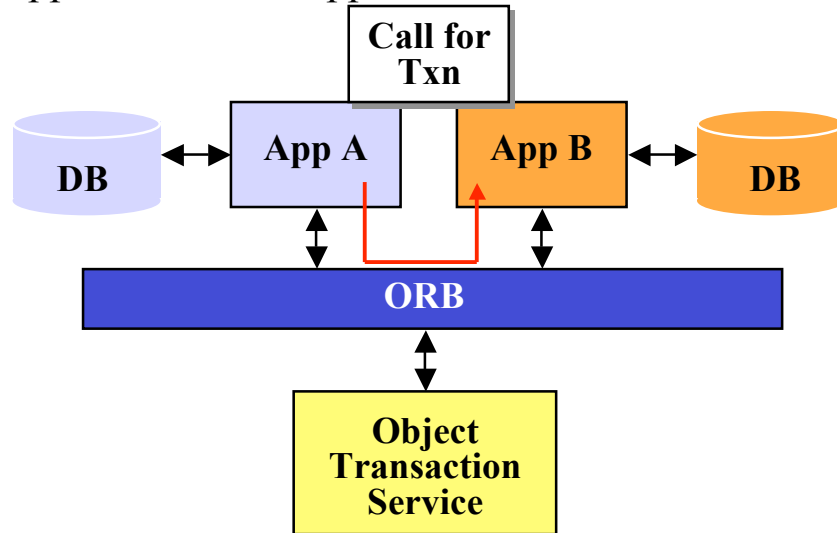


4) App A runs the txn but does not commit it

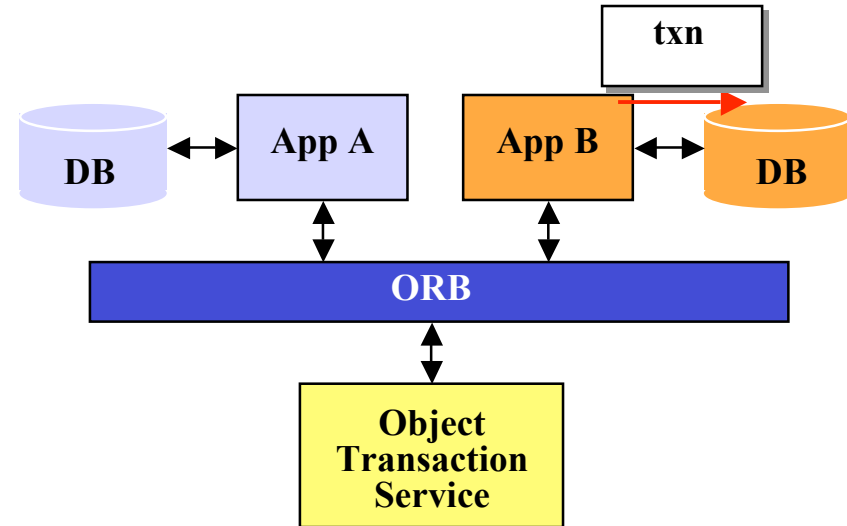


CORBA transactions (2)

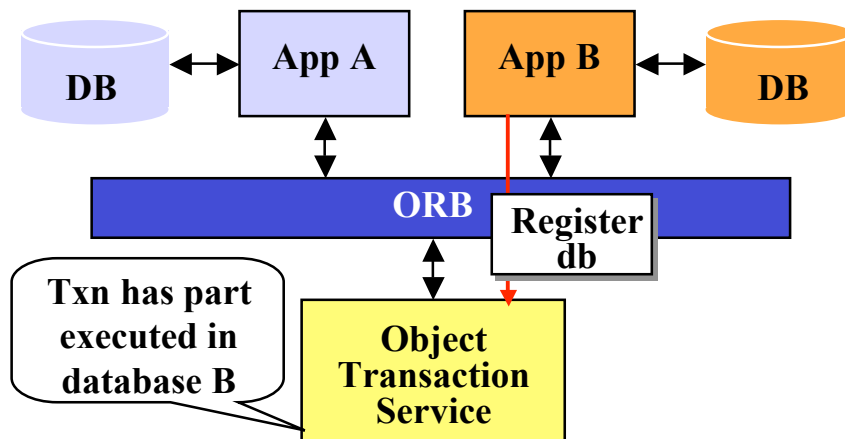
5) App A now calls App B



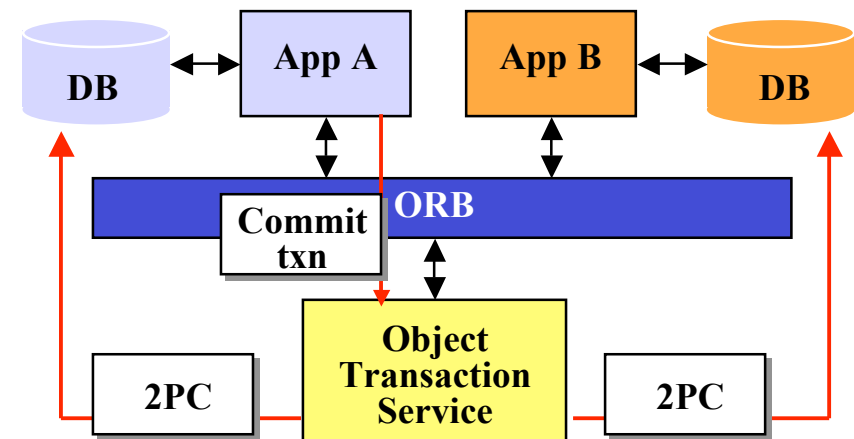
7) App B runs the txn but does not commit it



6) App B registers the database for that transaction

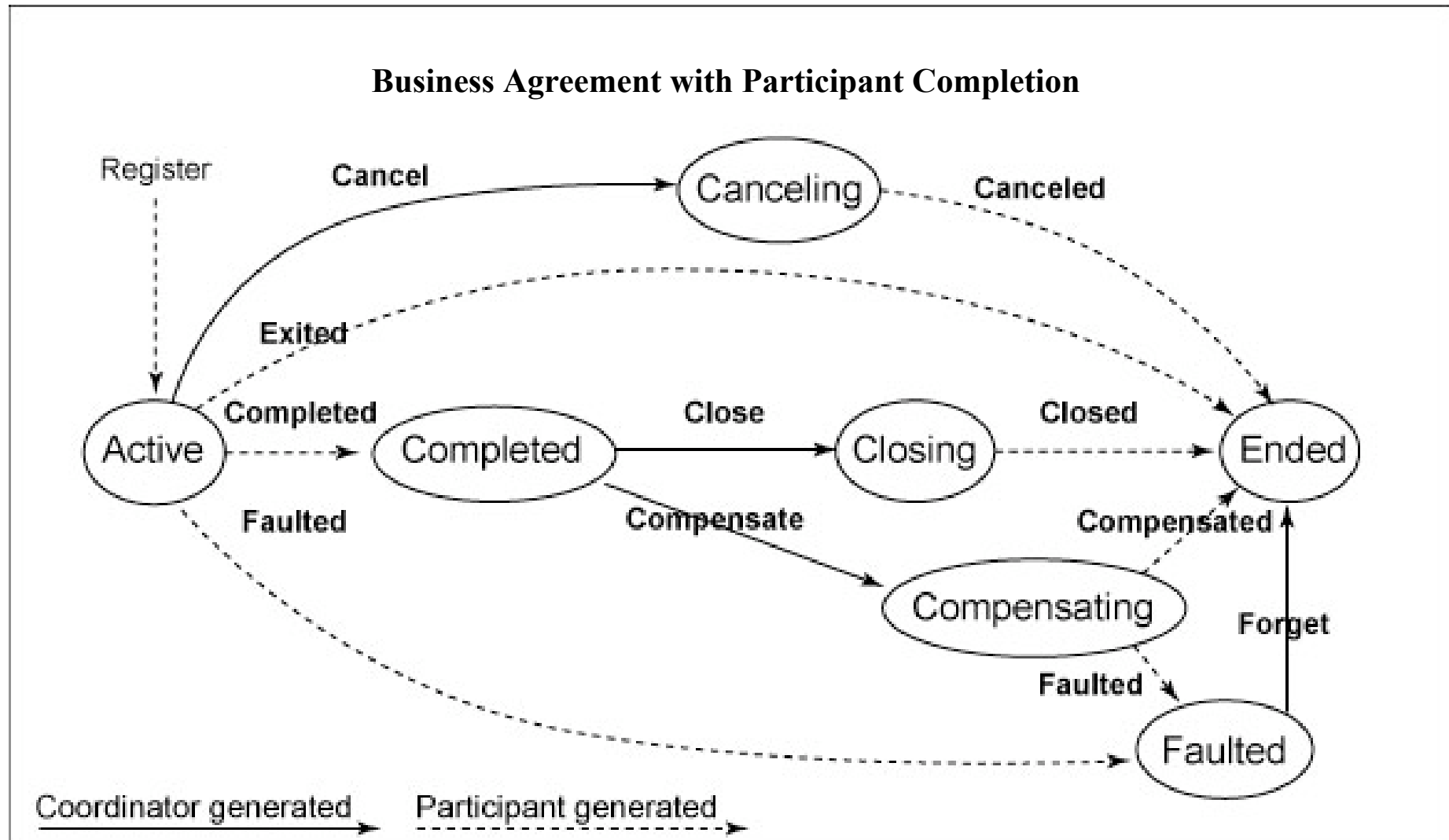


8) App A request commit and the OTS runs 2PC



WS-BusinessActivity

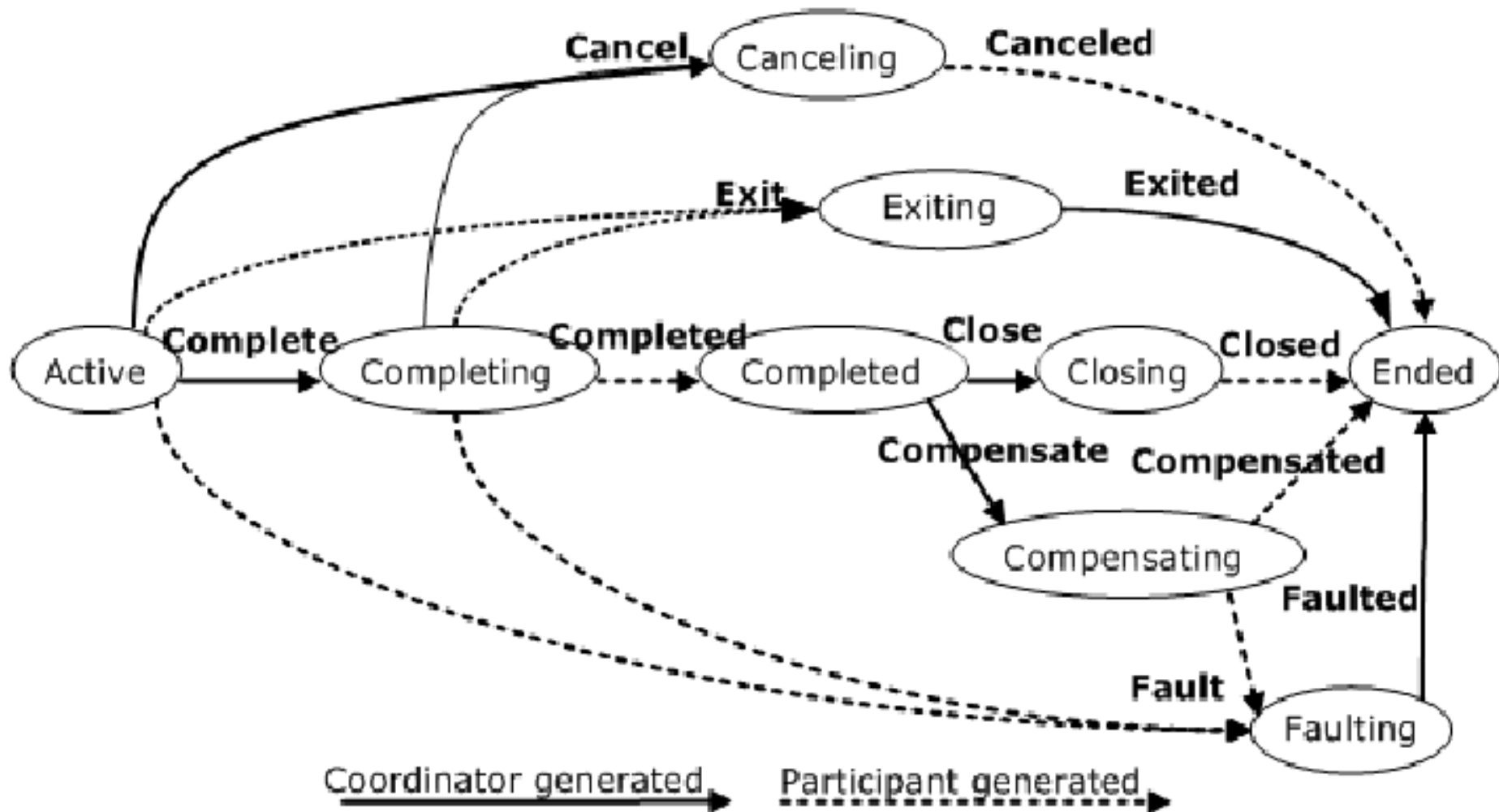
Business agreement



From WS-BusinessActivity September 2003

Business agreement

Business Agreement with Coordinator Completion



WS-Security

WS-Security

- ❑ WS-Security is intended as an extension to SOAP that can be used to implement message integrity, message confidentiality, and single message authentication. It also adds mechanisms for passing security tokens as part of SOAP messages and describes the encoding for such tokens
- ❑ WS-Security is geared towards the exchange of security tokens (e.g., an id or a password) along with digital signatures to confirm the possession of a key for that security token
- ❑ Message integrity is implemented by means of XML signature and message confidentiality is implemented through XML encryption
- ❑ WS-Security specifies the different SOAP header blocks in use to achieve the desired degree of security

```
    <S:Envelope>  
      <S:Header>
```

```
        ...  
        <Security S:actor="..." S:mustUnderstand="...">  
          ...  
        </Security>
```

```
      </S:Header>
```

```
    </S:Envelope>
```

From Web Services Security (WS-Security) Version 1.0 05 April 2002

Example of token

```
<S:Envelope xmlns:S="http://www.w3.org/2001/12/soap-envelope"  
  xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
```

```
<S:Header>
```

```
...
```

```
<wsse:Security>  
  <wsse:UsernameToken>  
    <wsse:Username>Zoe</wsse:Username>  
    <wsse:Password>ILoveDogs</wsse:Password>  
  </wsse:UsernameToken>  
</wsse:Security>
```

```
...
```

```
</S:Header>
```

```
...
```

```
</S:Envelope>
```

From Web Services Security (WS-Security) Version 1.0 05 April 2002

Encrypting a body block

<S:Envelope

```
xmlns:S="http://www.w3.org/2001/12/soap-envelope"  
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"  
xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"  
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
```

<S:Header>

```
<wsse:Security>  
  <xenc:ReferenceList>  
    <xenc:DataReference URI="#bodyID"/>  
  </xenc:ReferenceList>  
</wsse:Security>
```

</S:Header>

<S:Body>

```
<xenc:EncryptedData Id="bodyID">  
  <ds:KeyInfo>  
    <ds:KeyName>CN=Hiroshi Maruyama, C=JP</ds:KeyName>  
  </ds:KeyInfo>  
  <xenc:CipherData>  
    <xenc:CipherValue>...</xenc:CipherValue>  
  </xenc:CipherData>  
</xenc:EncryptedData>
```

</S:Body>

</S:Envelope>

From Web Services Security (WS-Security) Version 1.0 05 April 2002

Web services extensions



- ❑ WS-Coordination, WS-Transactions, and WS-Security show that SOAP, WSDL, and UDDI are still in their early stages and there is still quite a way to go until there are standardized solutions that provide all the necessary functionality
- ❑ Interesting at this stage is how these specifications build upon each other and on SOAP and WSDL. This is, in fact, one of the advantages of, e.g., SOAP. It is generic enough to allow multiple extensions, because, in principle, the only thing needed is an agreement on the information on the header and how to treat it
- ❑ These extensions are also a source of controversy as in several cases they are tied to intellectual propriety rights held by the companies proposing these extensions. Should the extensions become standards, then any traffic based on such extensions will have to pay royalties to the companies holding the intellectual propriety rights
- ❑ WS-Coordination, WS-Transactions, and WS-Security are mainly the work of IBM and Microsoft; two of the companies that have not yet entirely agreed to royalty free contributions to Web services standards