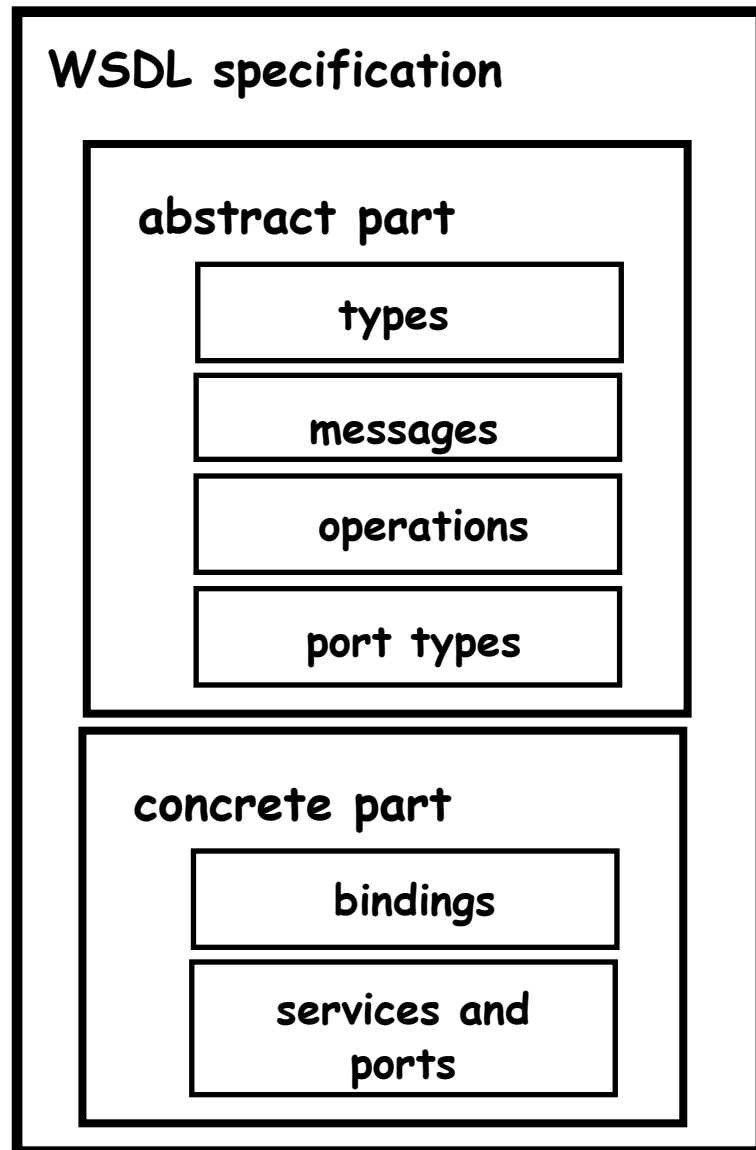


- [ACKM04] Ch 6, 7



- Abstract part
 - signatures, operations
 - like traditional IDL
- Concrete part
 - bindings / services / ports

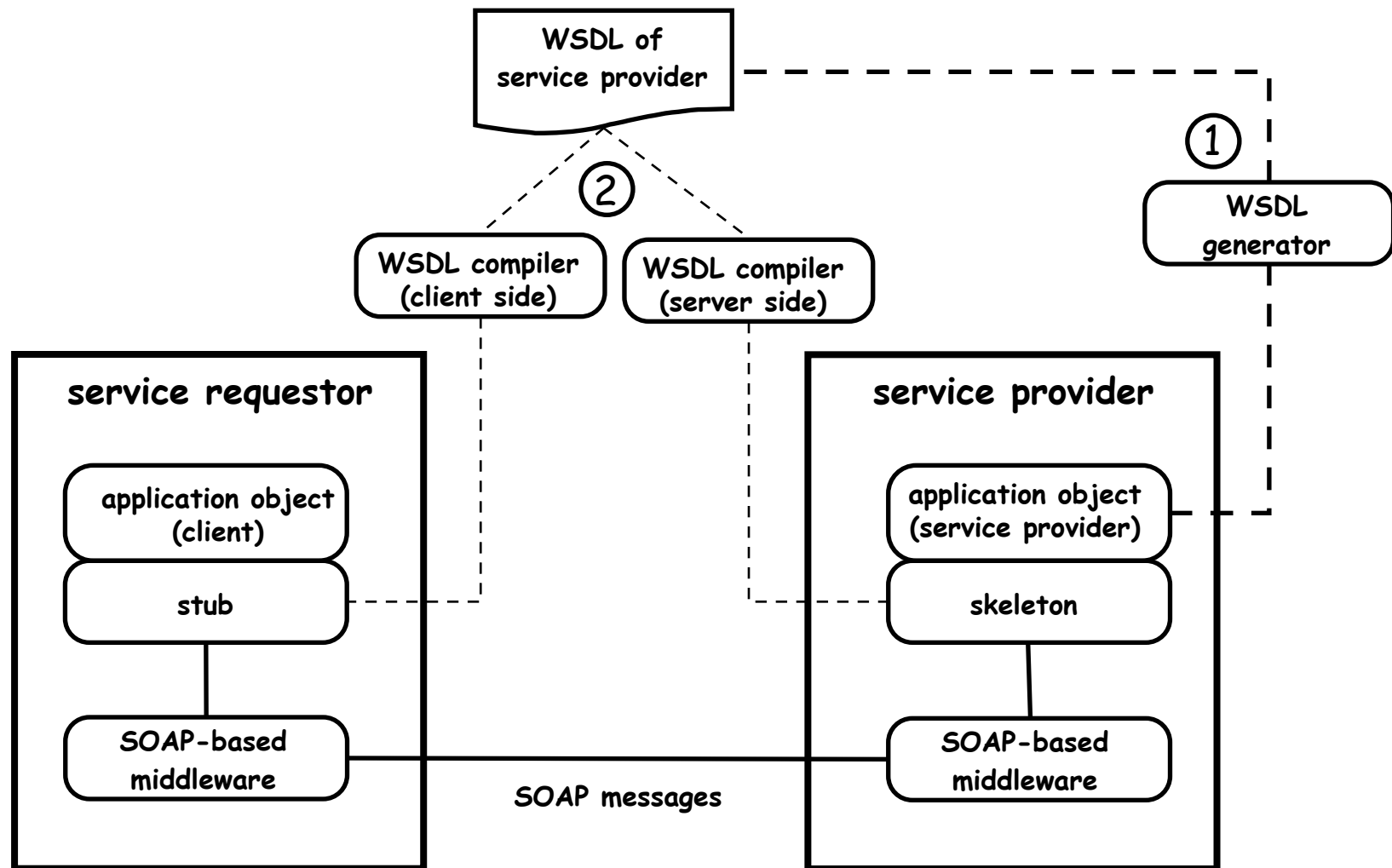
Concrete Part

- Interface Bindings
 - message encoding & protocol bindings
- Ports
 - interface binding + network address
- Services
 - logical groupings of ports

A WSDL Service Specification



Automatic Generation of WSDL

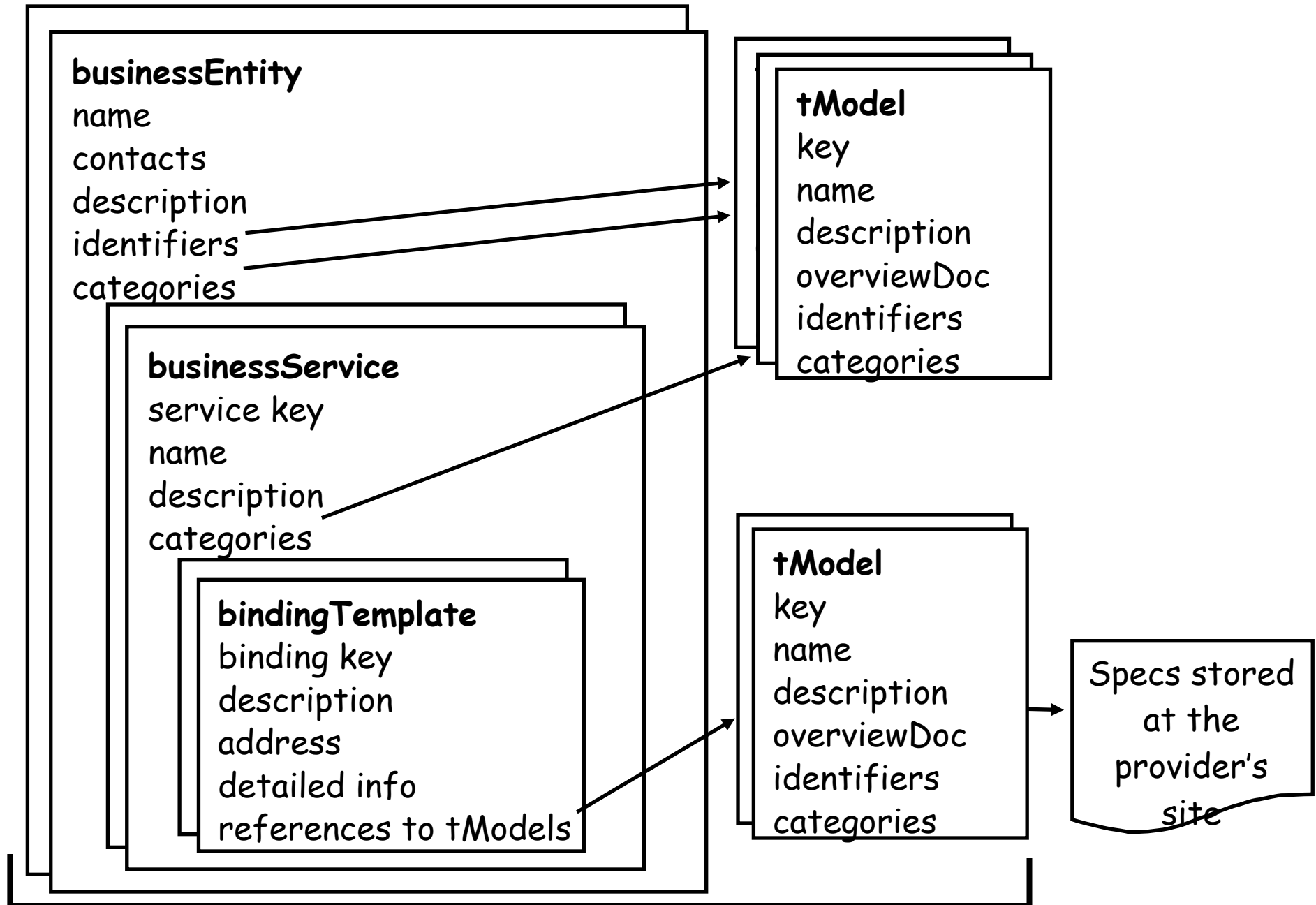


- The Web Services Directory Service
- Service Registry
 - for browsing by developers
 - for dynamic binding
- Business registry
 - original goal: global registry, every business and exported service registered there
 - now: support interaction between private and public UDDI registries

Information in UDDI

- Business entity
 - an organization that provides Web services
- Business Service
 - group of Web Services that cooperate in performing some business process
- bindingTemplate
 - technical information needed to use the service
- tModel
 - general container for a specification

Information in UDDI



Stored in the UDDI registry

tModel Description

overviewDoc
(refer to WSDL
specs and to
API specs)

```
<tModel tModelKey="uddi:uddi.org:v3_publication">
  <name>uddi-org:publication v3</name>
  <description>UDDI Publication API V3.0</description>
  <overviewDoc>
    <overviewURL useType="wsdlInterface">
      http://uddi.org/wsdl/uddi_api_v3_binding.wsdl#UDDI_Publication_SoapBinding
    </overviewURL>
  </overviewDoc>
  <overviewDoc>
    <overviewURL useType="text">
      http://uddi.org/pubs/uddi_v3.htm#PubV3
    </overviewURL>
  </overviewDoc>
</tModel>
```

classification
information
(specifies that this
tModel is about
XML, WSDL, and
SOAP specs)

```
<categoryBag>
  <keyedReference keyName="uddi-org:types:wsdl"
    keyValue="wsdlSpec"
    tModelKey="uddi:uddi.org:categorization:types"/>
  <keyedReference keyName="uddi-org:types:soap"
    keyValue="soapSpec"
    tModelKey="uddi:uddi.org:categorization:types"/>
  <keyedReference keyName="uddi-org:types:xml"
    keyValue="xmlSpec"
    tModelKey="uddi:uddi.org:categorization:types"/>
  <keyedReference keyName="uddi-org:types:specification"
    keyValue="specification"
    tModelKey="uddi:uddi.org:categorization:types"/>
</categoryBag>

</tModel>
```

- Note overviewDoc has URL - refers to specification stored outside UDDI server
- Multiple overviewDocs are allowed

Sharing of tModels

- Each tModel is assigned a GUID when it is registered
- The tModels may be shared by multiple bindingTemplates
- Software can use UDDI to search for all instances of a given GUID

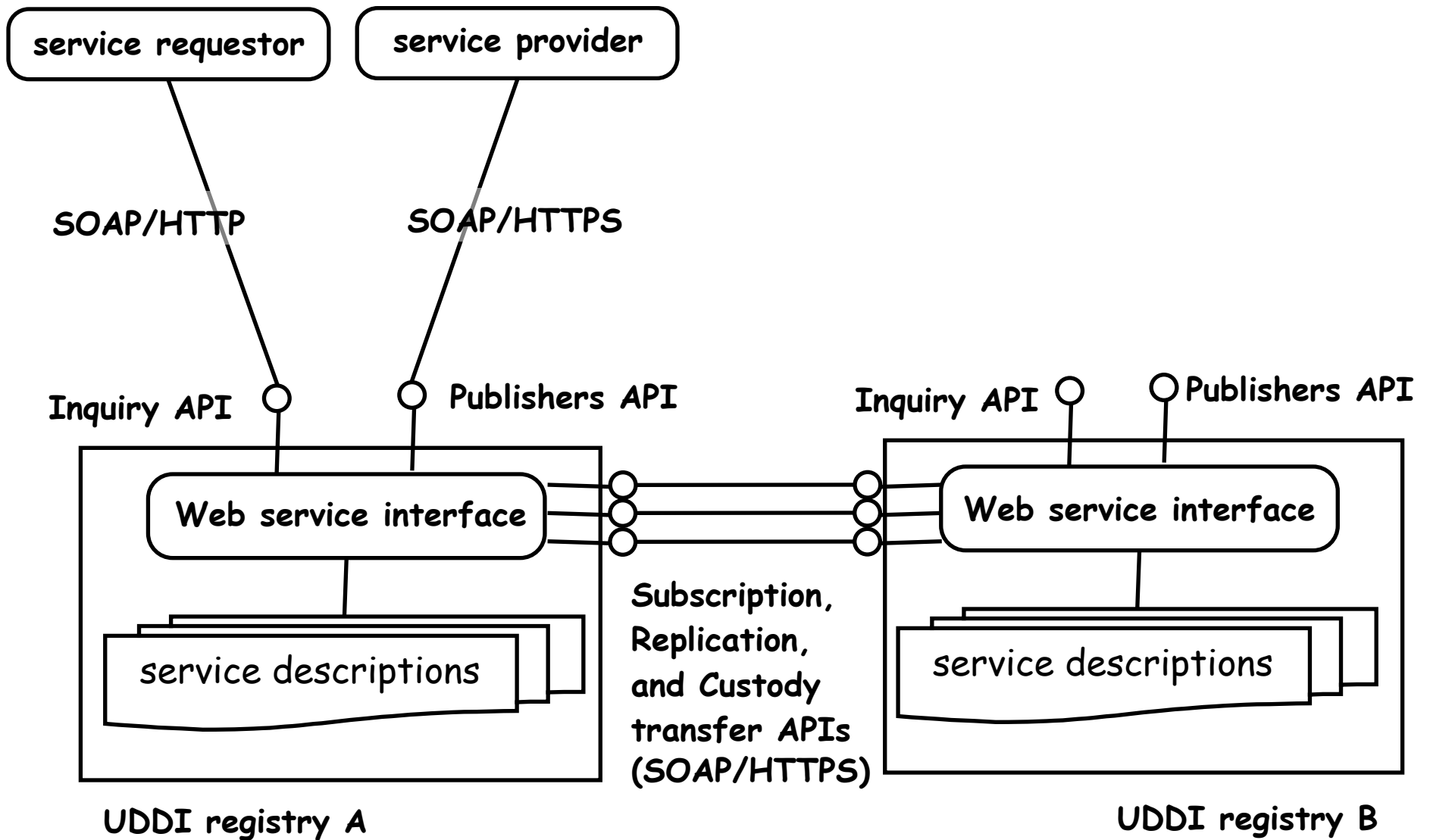
UDDI Registry API

- Inquiry
 - search
- Publisher
 - register and unregister operations

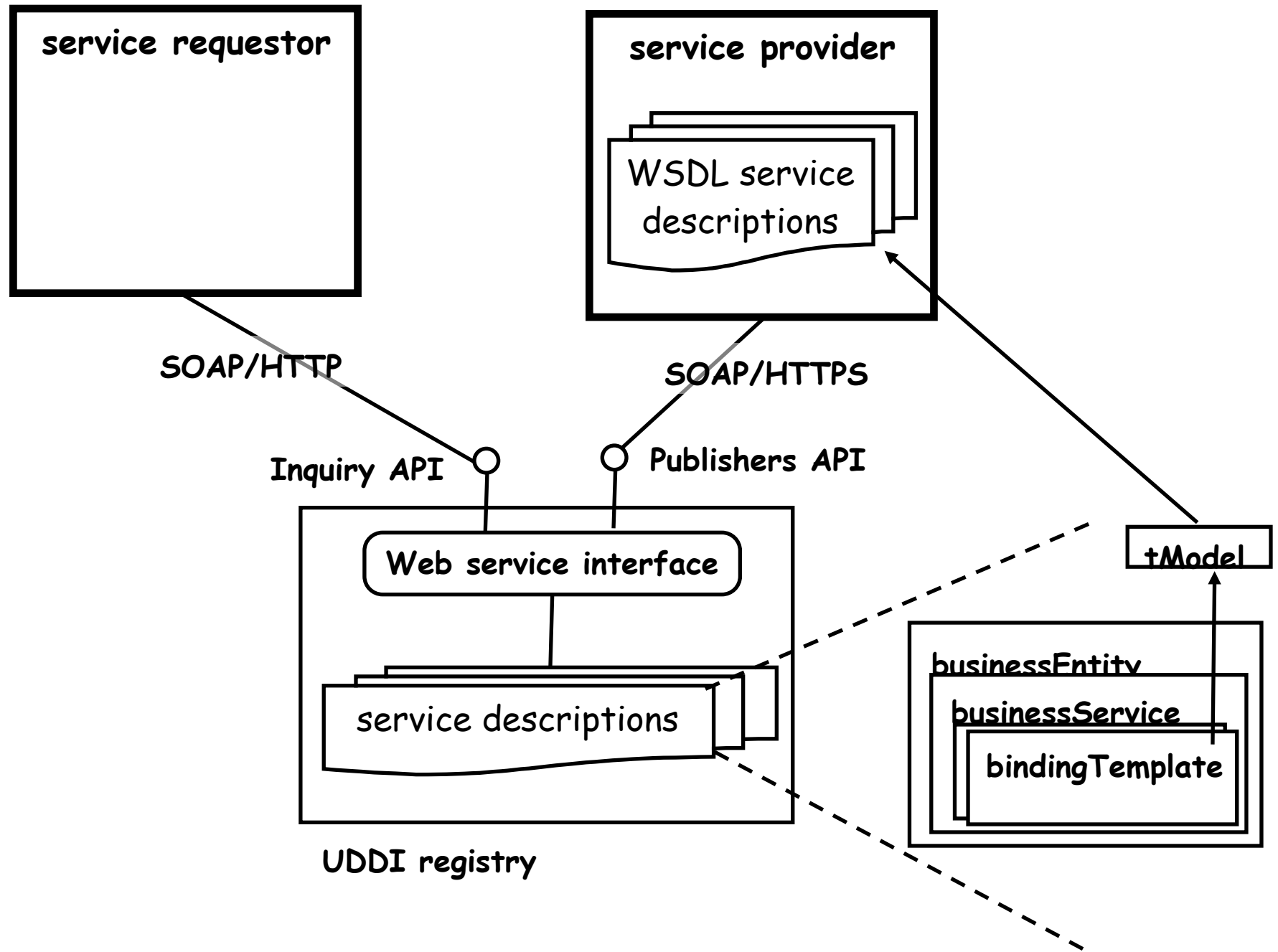
UDDI Registry API - II

- Security
- Custody and Ownership Transfer
 - create/destroy/move objects
- Subscription
 - monitor for changes in UDDI data
- Replication
 - readonly except at “owner” site

Architecture of Distributed UDDI



WSDL in UDDI

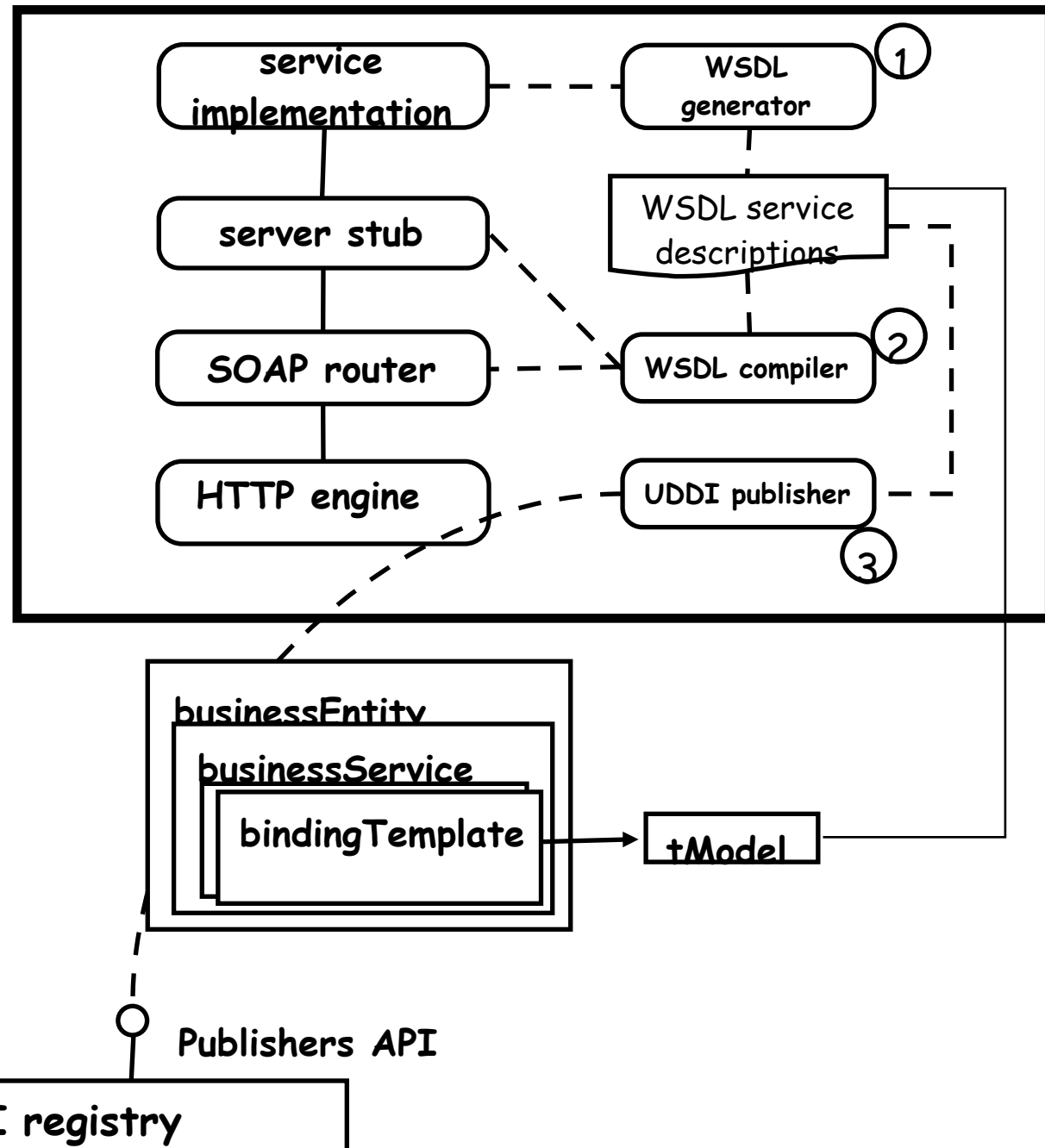


Private UDDI Registries

- Public Universal Business Registry not required for Web apps that are intra-enterprise or between trusted partners
- Three categories
 - public
 - private
 - shared
- Must support replication between registries where appropriate

Example: Database stored procedures

service
provider



“Advances” in SOAP

- BLOBs
 - now Web Services description yields interaction sequence but not data formats, which are agreed to by communicating applications
- Document exchange
 - constructing WSDL requires semantic understanding of the documents

Dynamic Binding

- Fully general
 - e.g. CORBA DII
 - now we have run-time types, introspection
- Restricted
 - e.g. all services using a given tModel
 - but enterprises usually won't accept a program effectively signing a contract ...

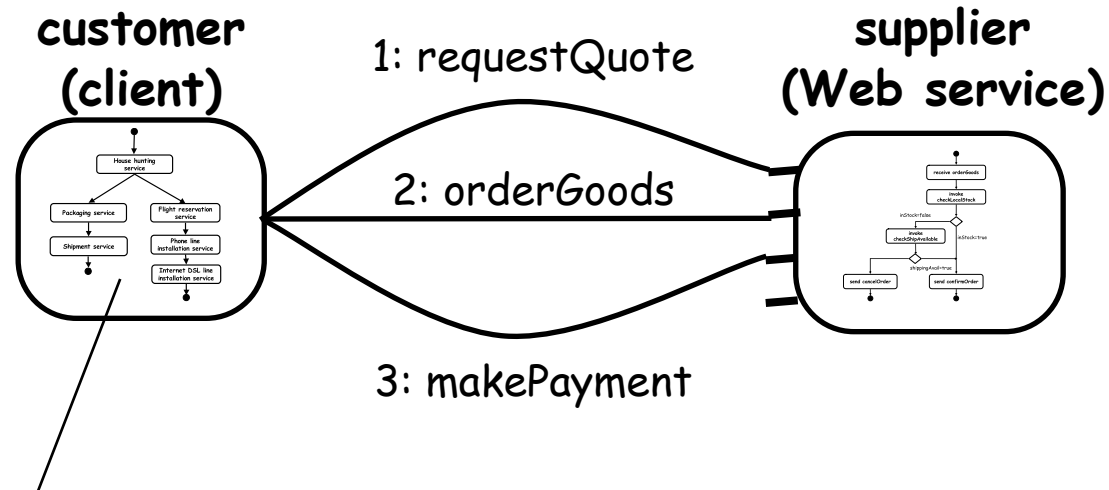
Service Coordination

- Not Yet Implemented!
- But needed - business transactions comprise multiple method invocations that are not independent
- How specify valid and invalid sequences of invocations?

A Simple Conversation

The interaction between clients and services is often formed by a set of operation invocations (i.e., it is a conversation).

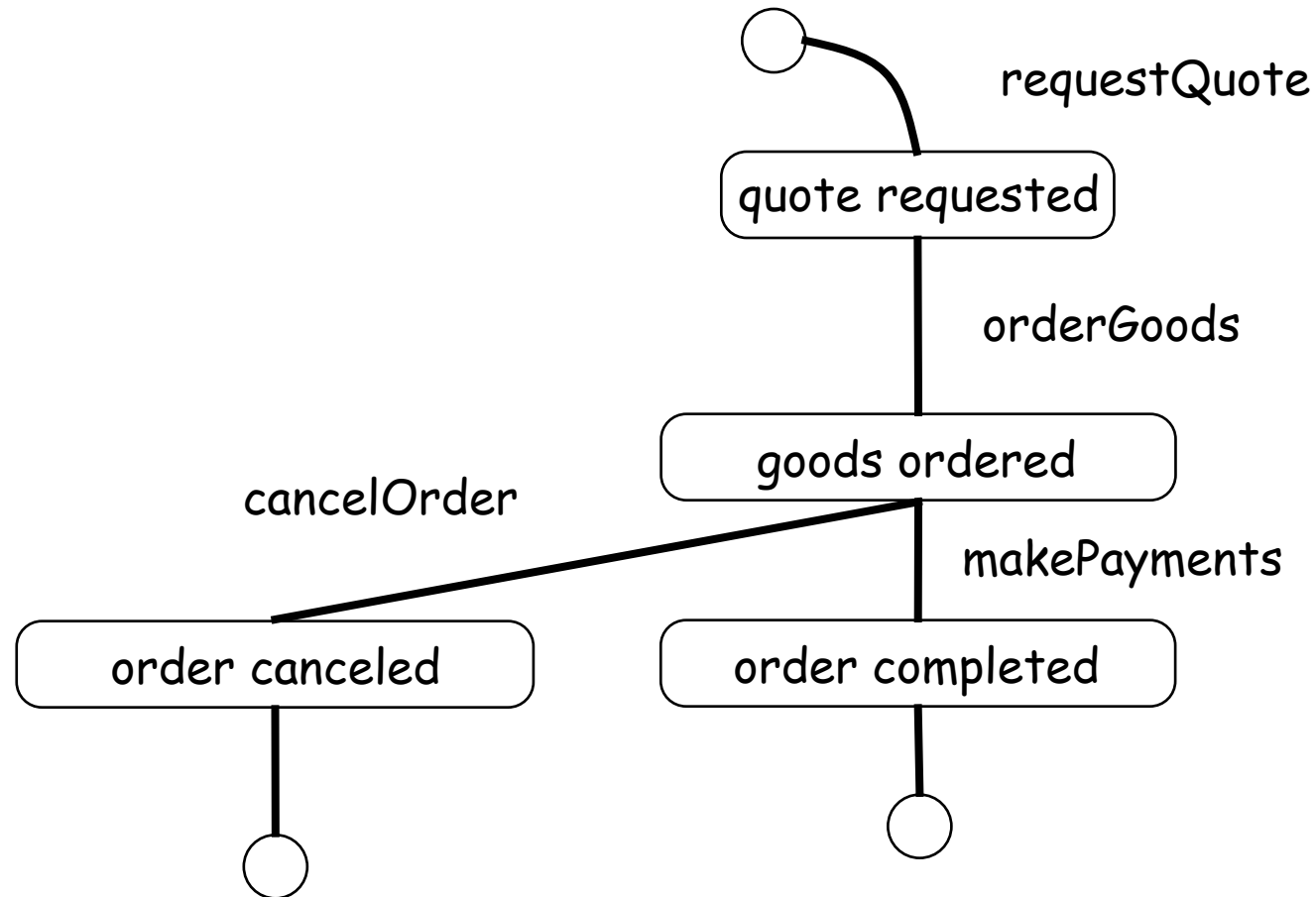
A service provider may support some conversations while disallowing others.



The internal business logic of clients and Web services must support the conversation, and maintain the state across different operation invocations belonging to the same conversation.

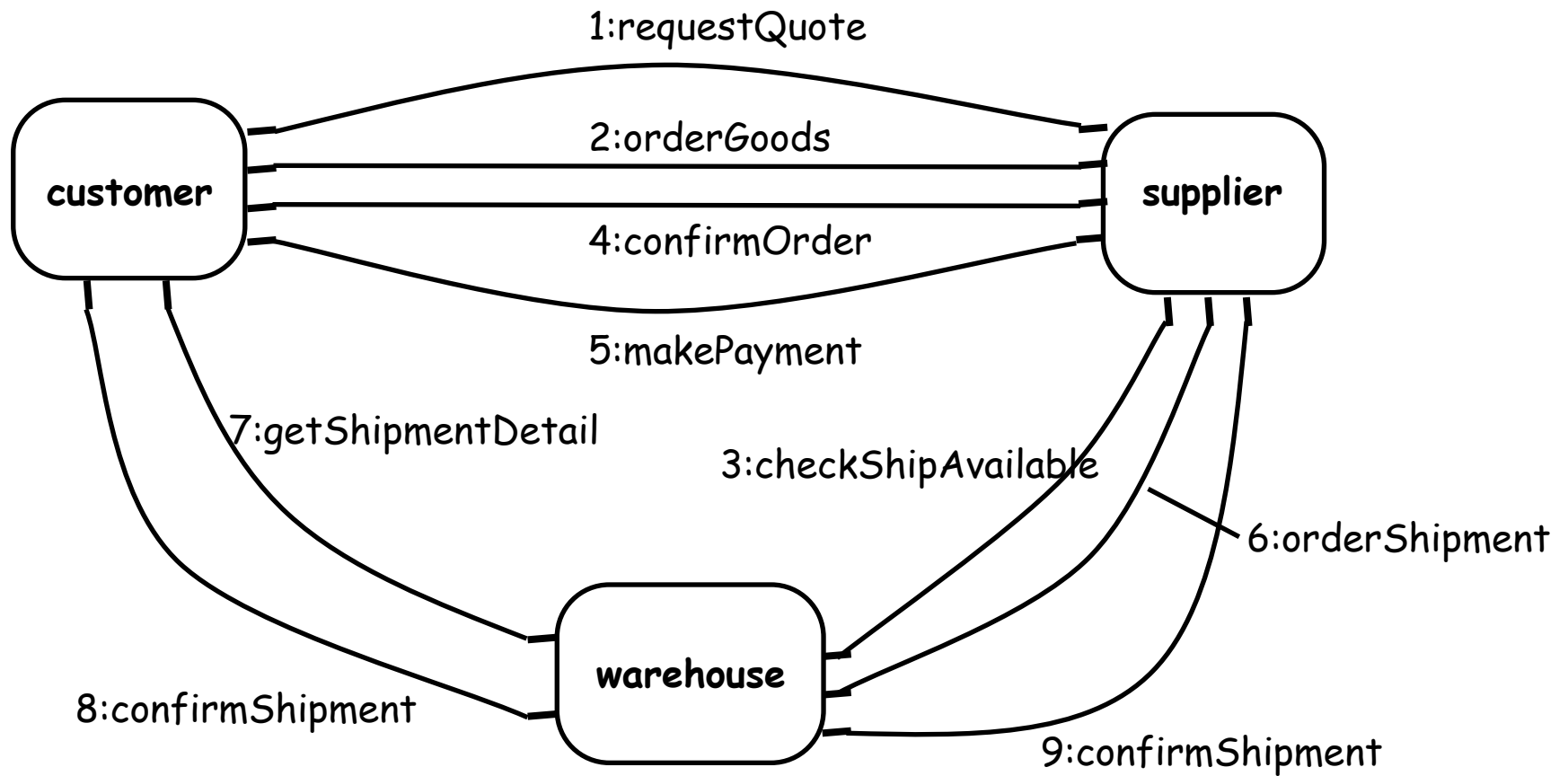
- How ensure e.g. requestQuote happens before orderGoods?

State Machine Approach

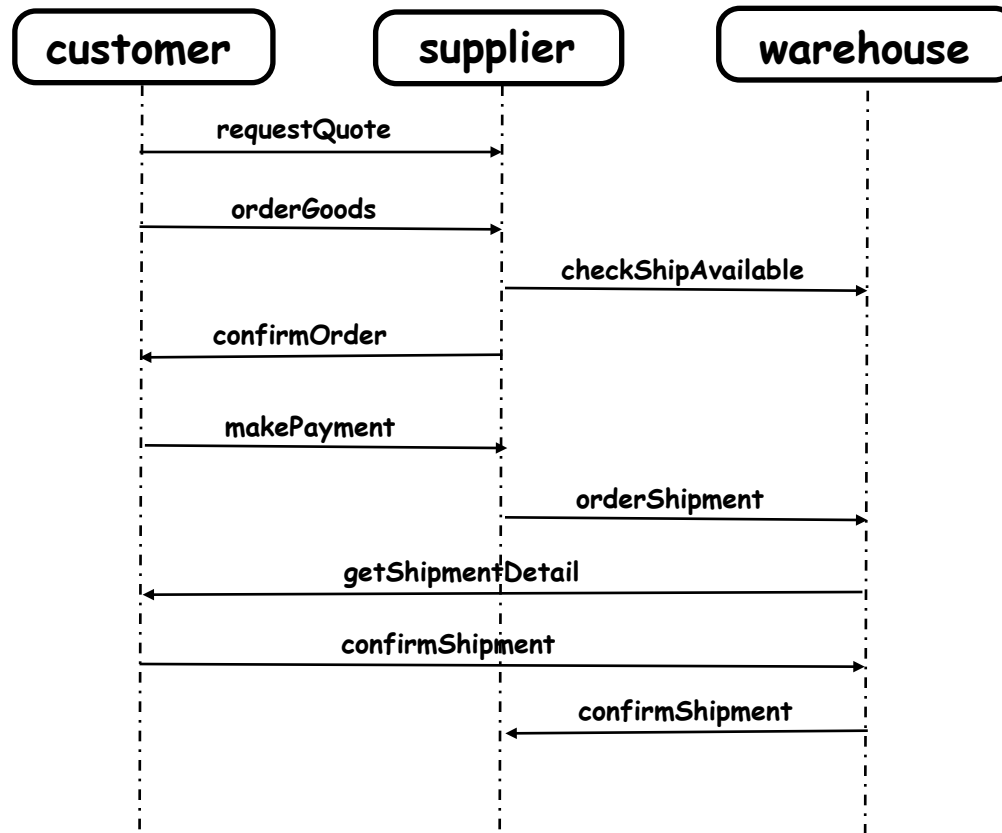


- boxes = states
- edge labels = operation calls

Multiple Servers?

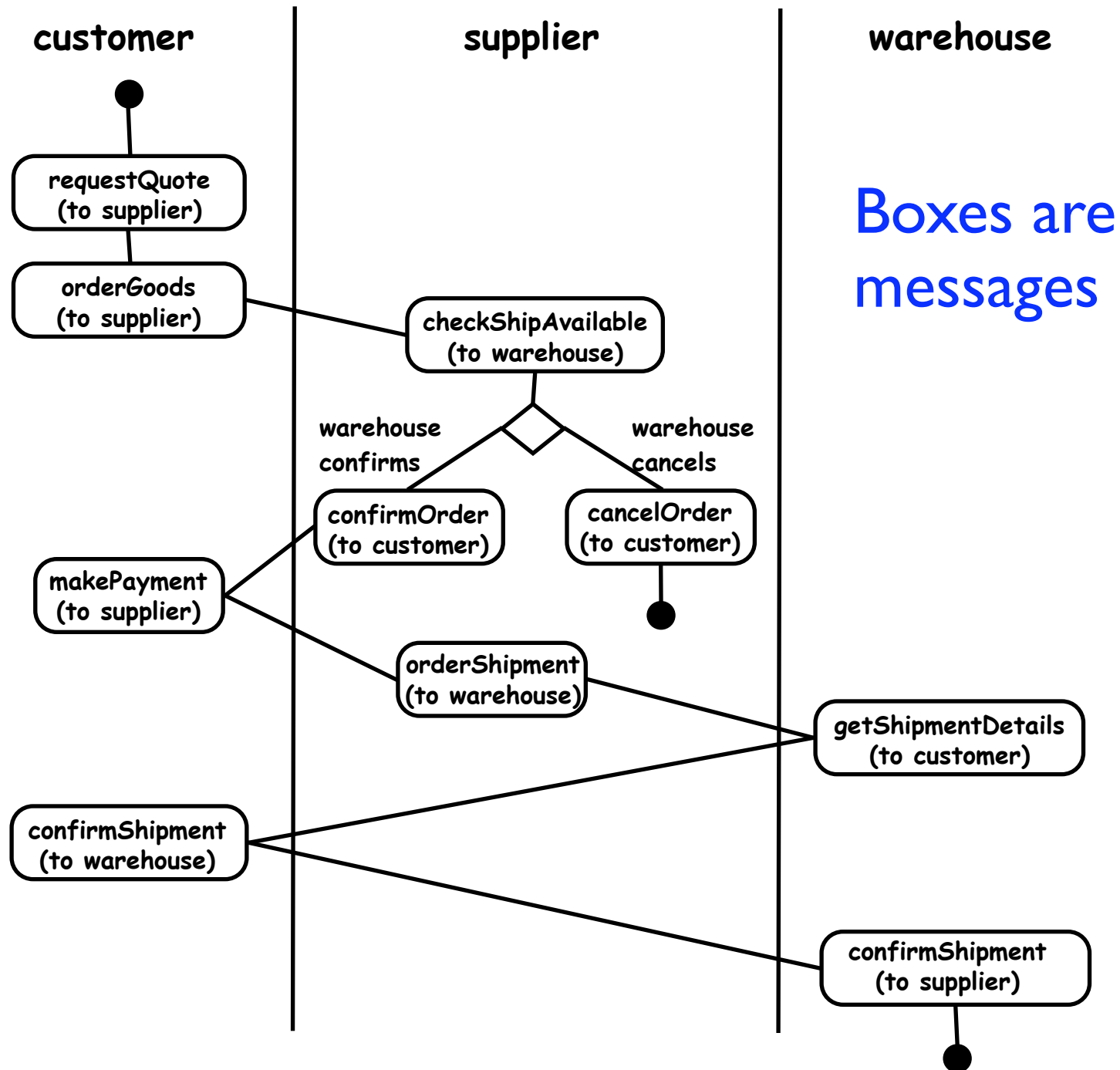


Sequence Diagram

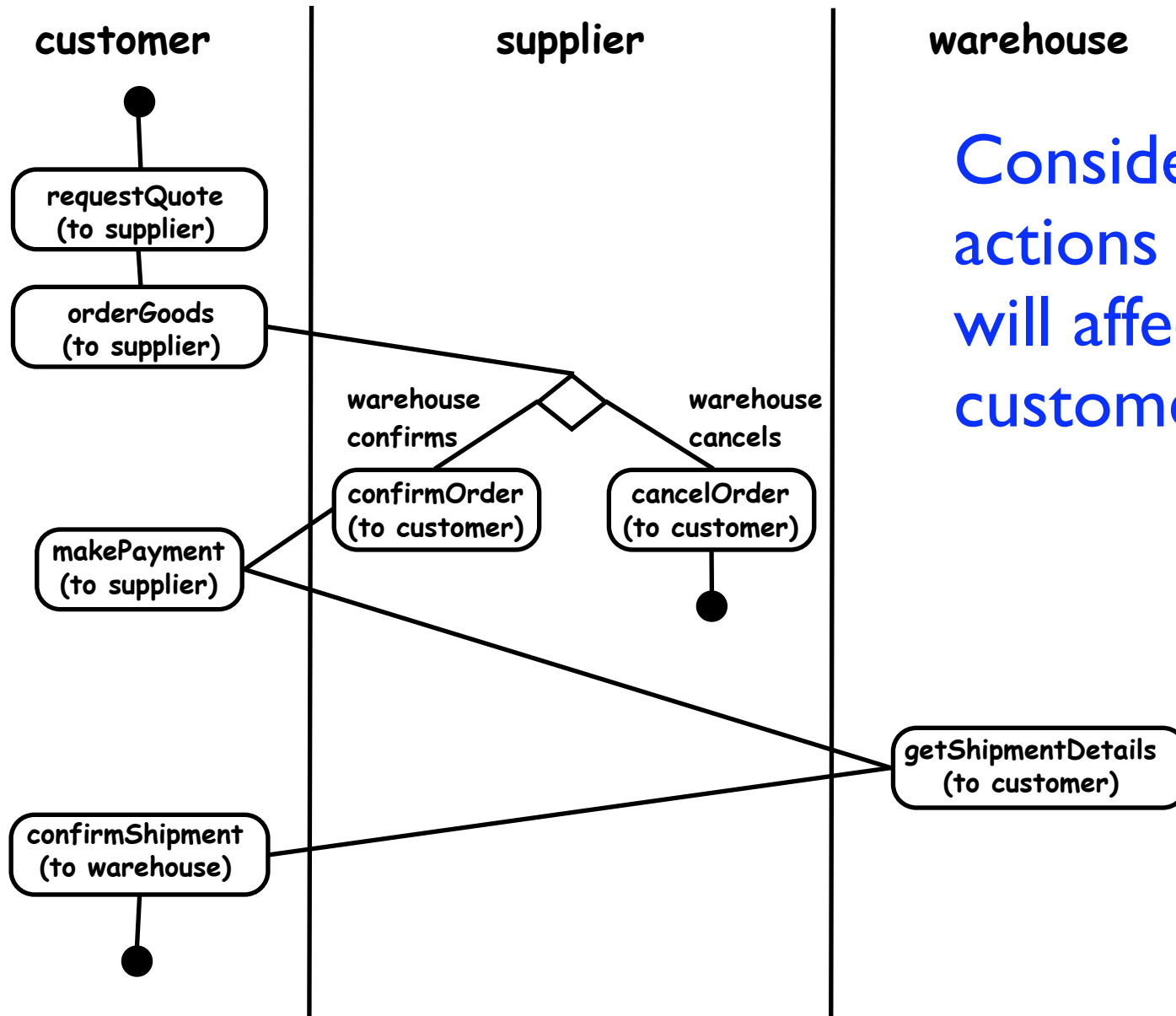


- Simple and intuitive ...
- Drawback: cannot handle alternatives

Activity Diagram



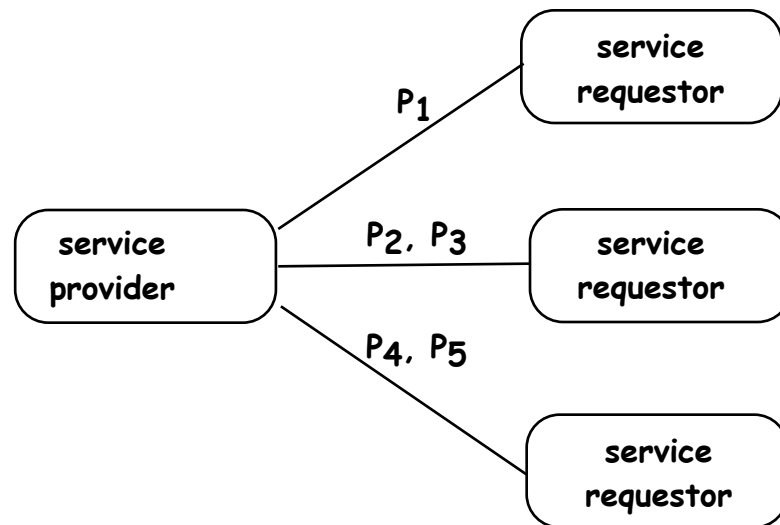
Activity Diagram Slice



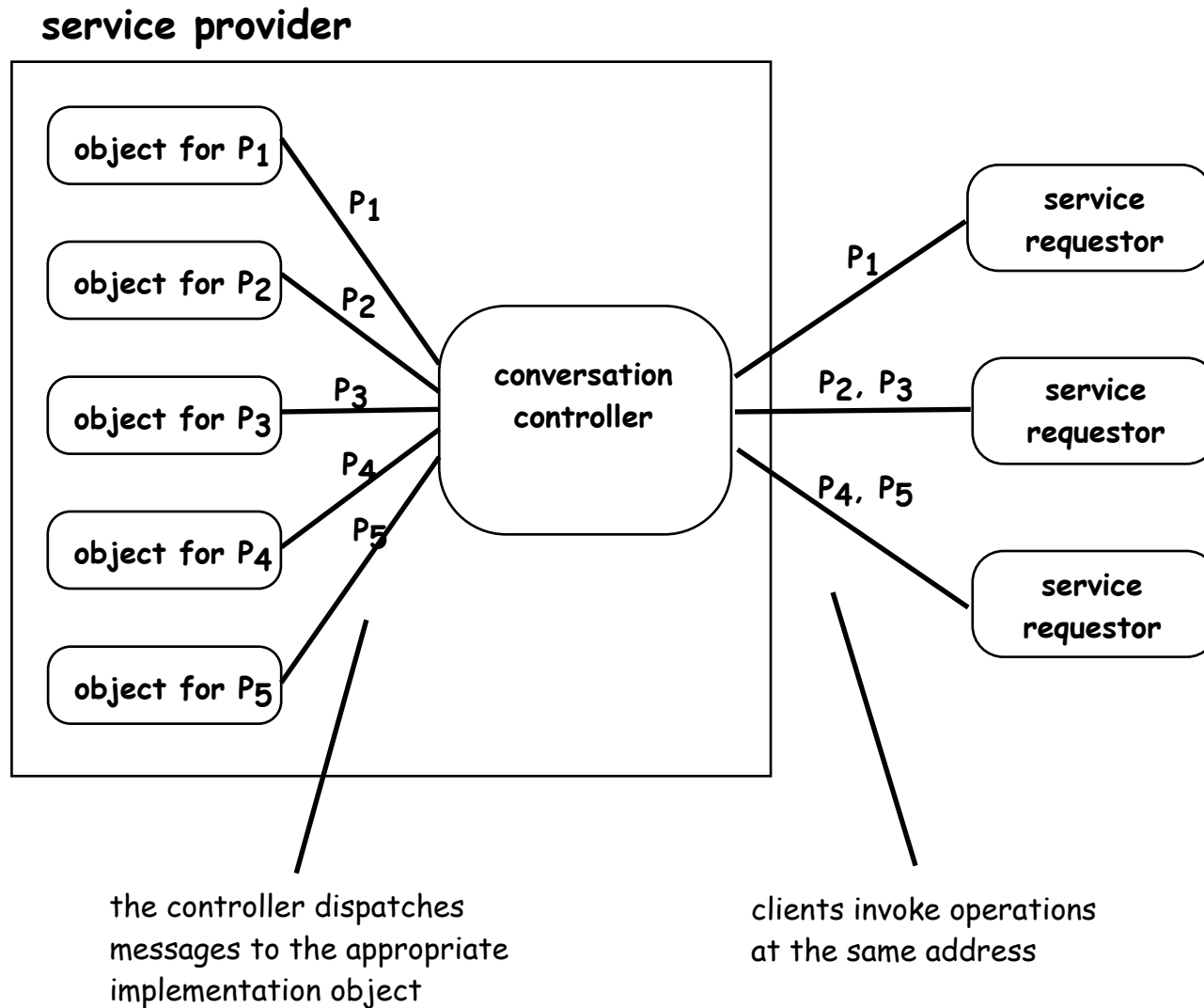
Consider only
actions that
will affect
customer ...

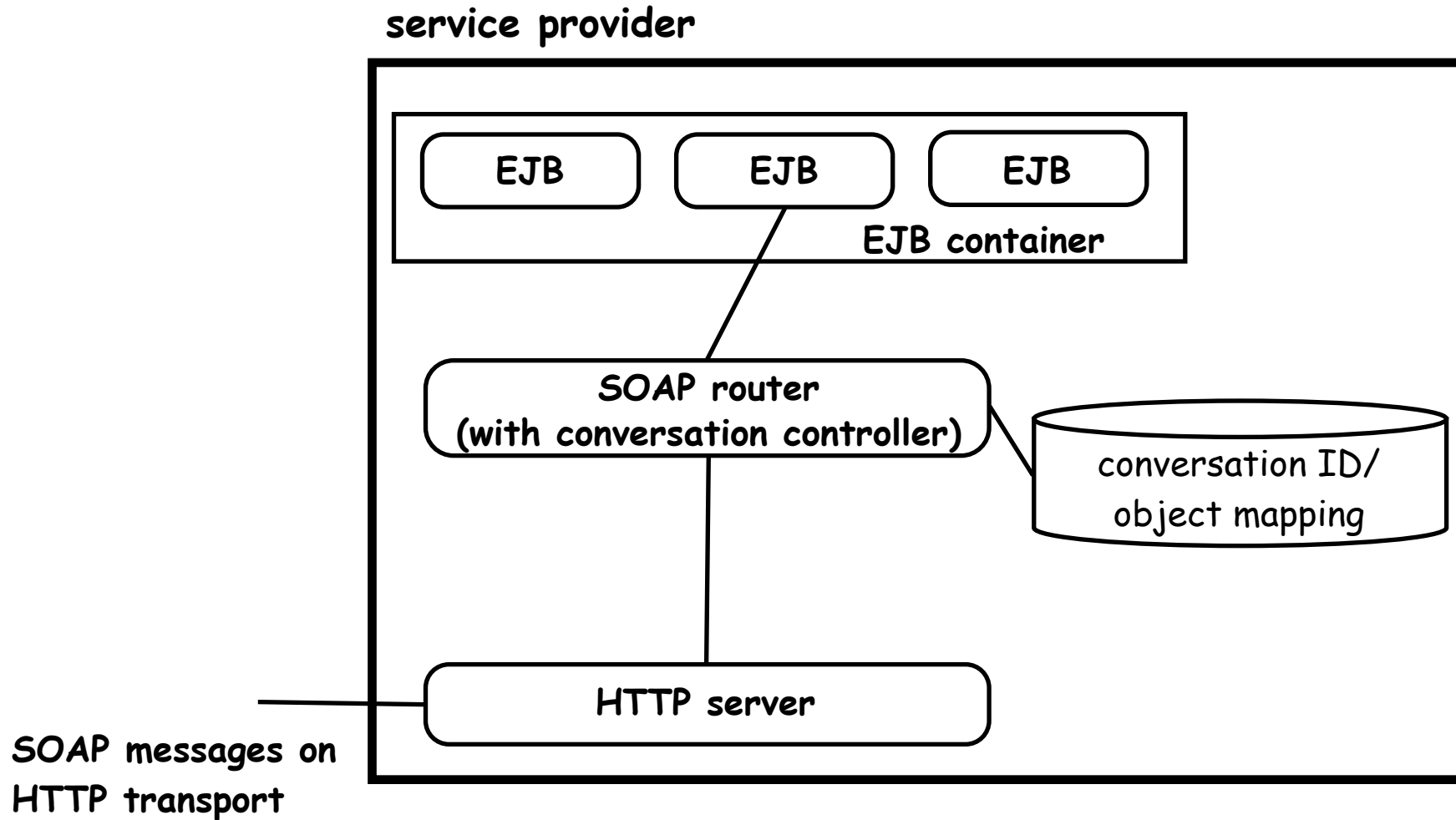
Conversation Controller

- Conversation routing
- Protocol compliance checking
- Example: many clients all use protocol P

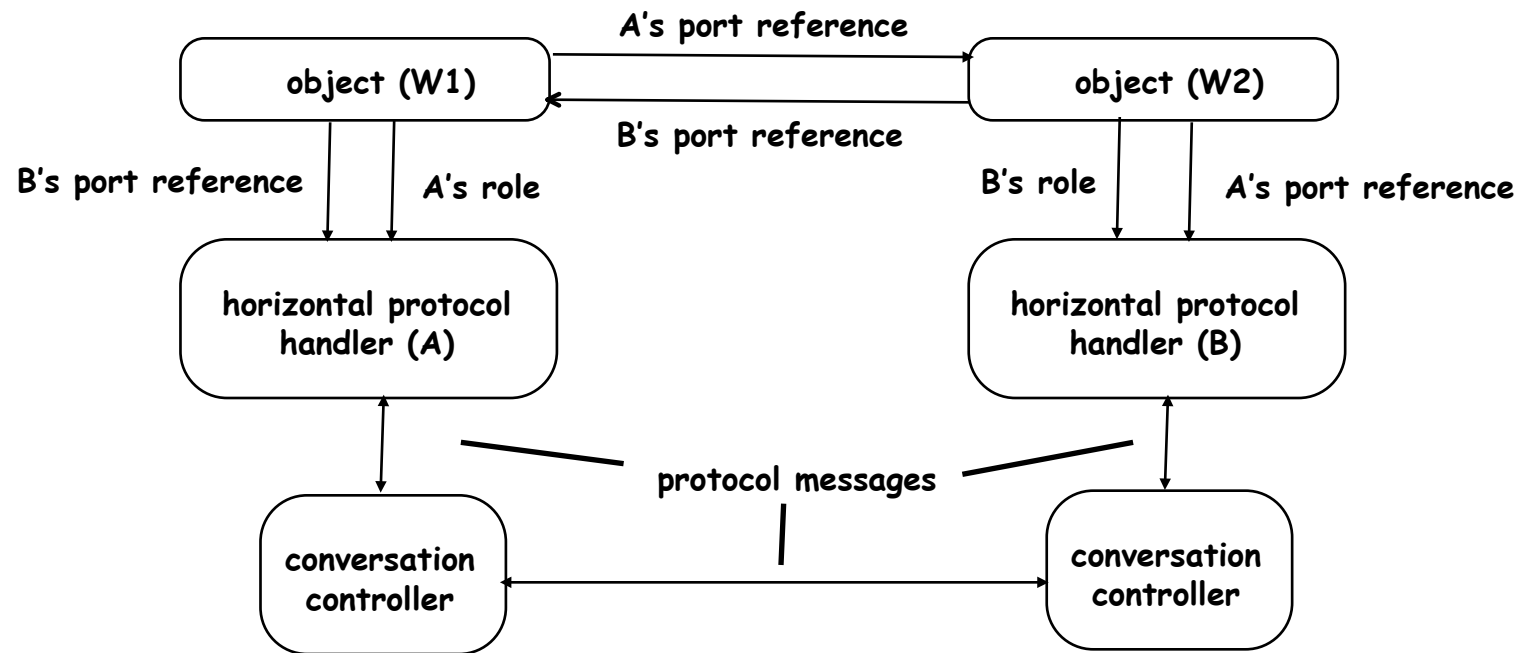


Conversation Controller - Routing





Identity of Conversation

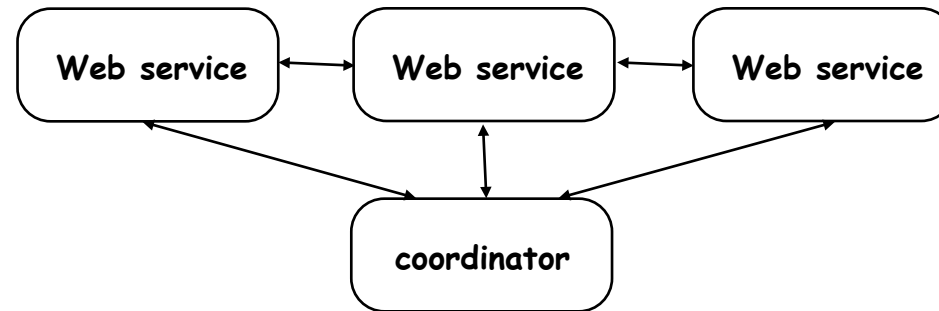


- Web services must communicate port references and role information to the conversation controller infrastructure

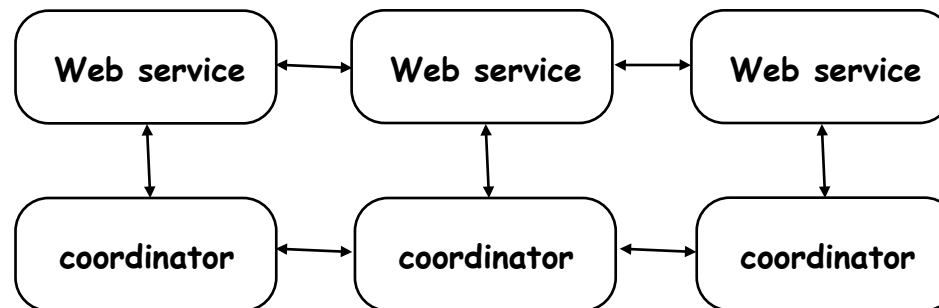
- A framework for supporting coordination
- Standardizes:
 - passing GUID between interacting Web services
 - informing protocol handler about ports
 - informing protocol handler about roles

Coordinators and Participants

- Coordinators and participants
- Central or distributed coordinator



(a) central coordination

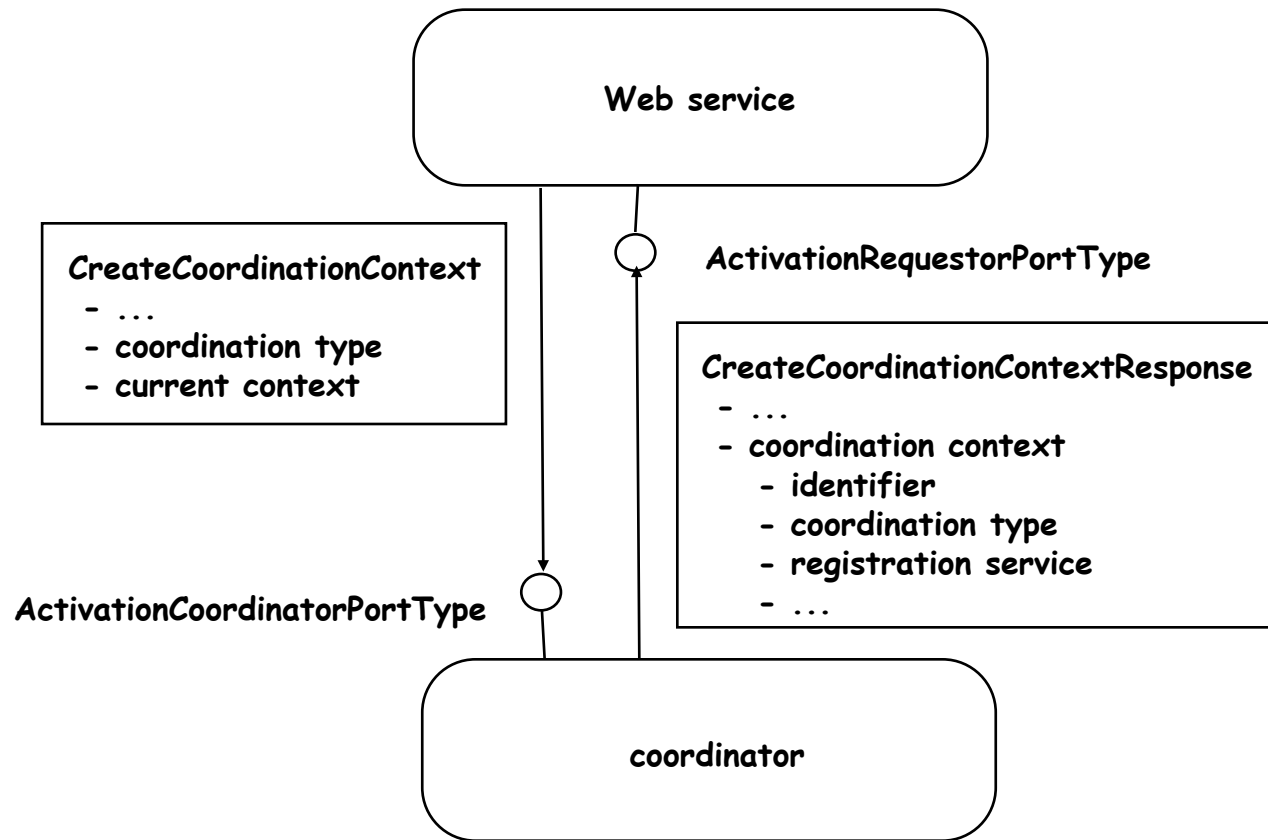


(b) distributed coordination

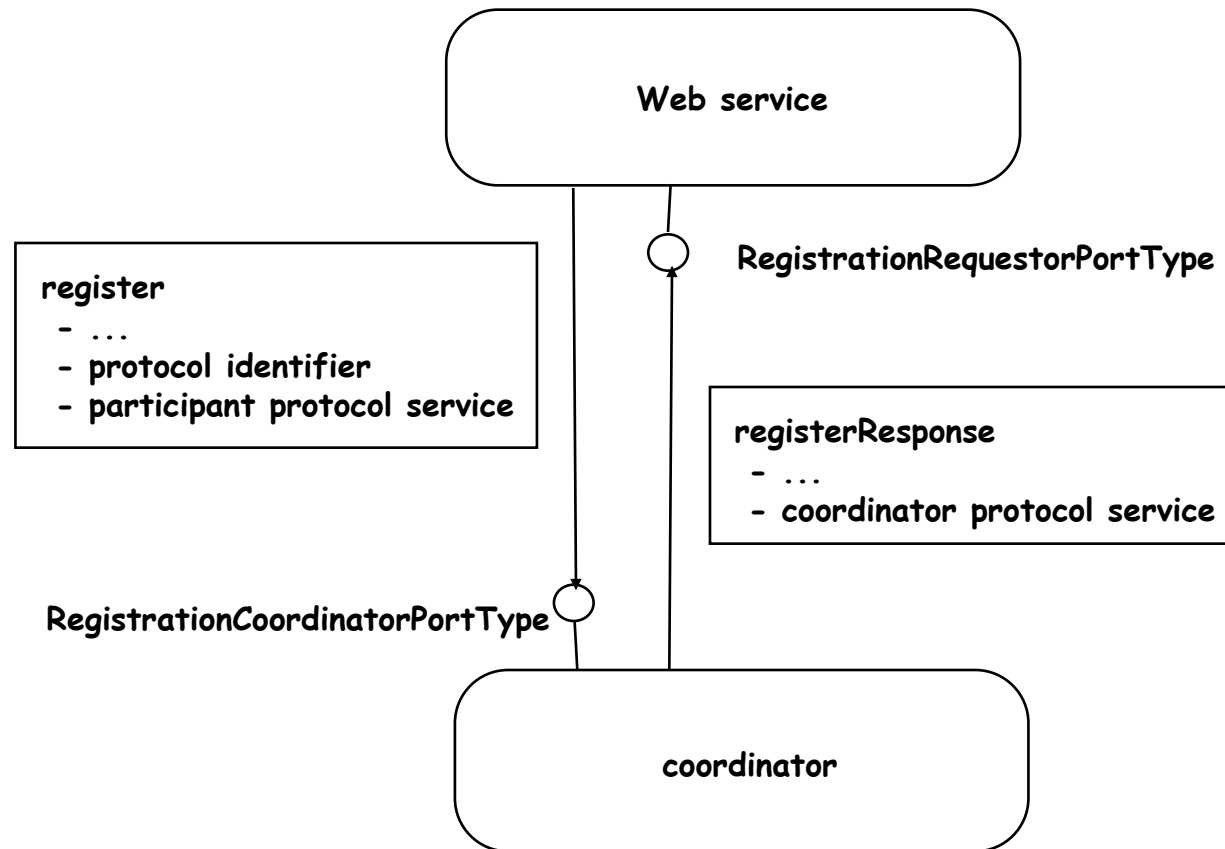
- Coordination protocol
 - e.g. 2PC
- Coordination type
 - e.g. atomic transaction
- Coordination context
 - data structure identifying messages as belonging to a conversation

- Activation
 - participant requests new context
- Registration
 - participant registers self with coordinator
- Protocol specific interactions
 - interactions specific to the protocol :-)

Activation

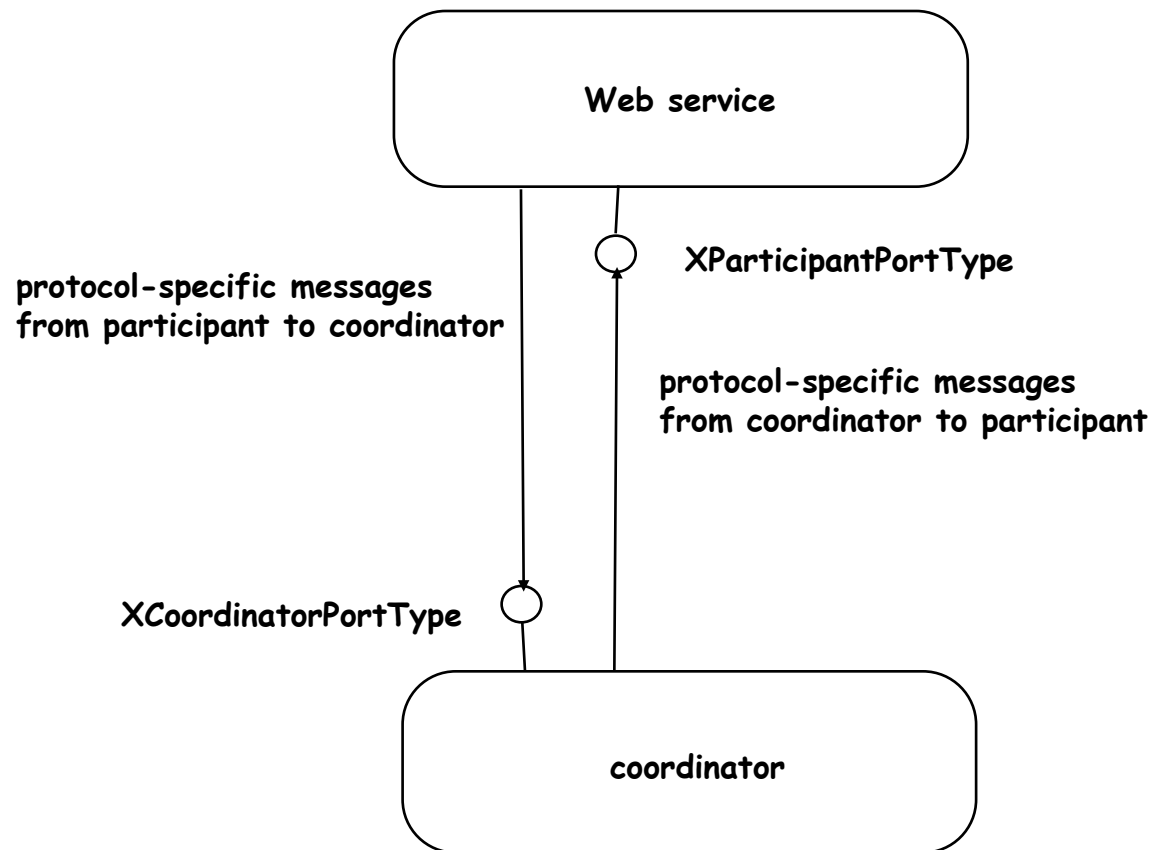


Registration

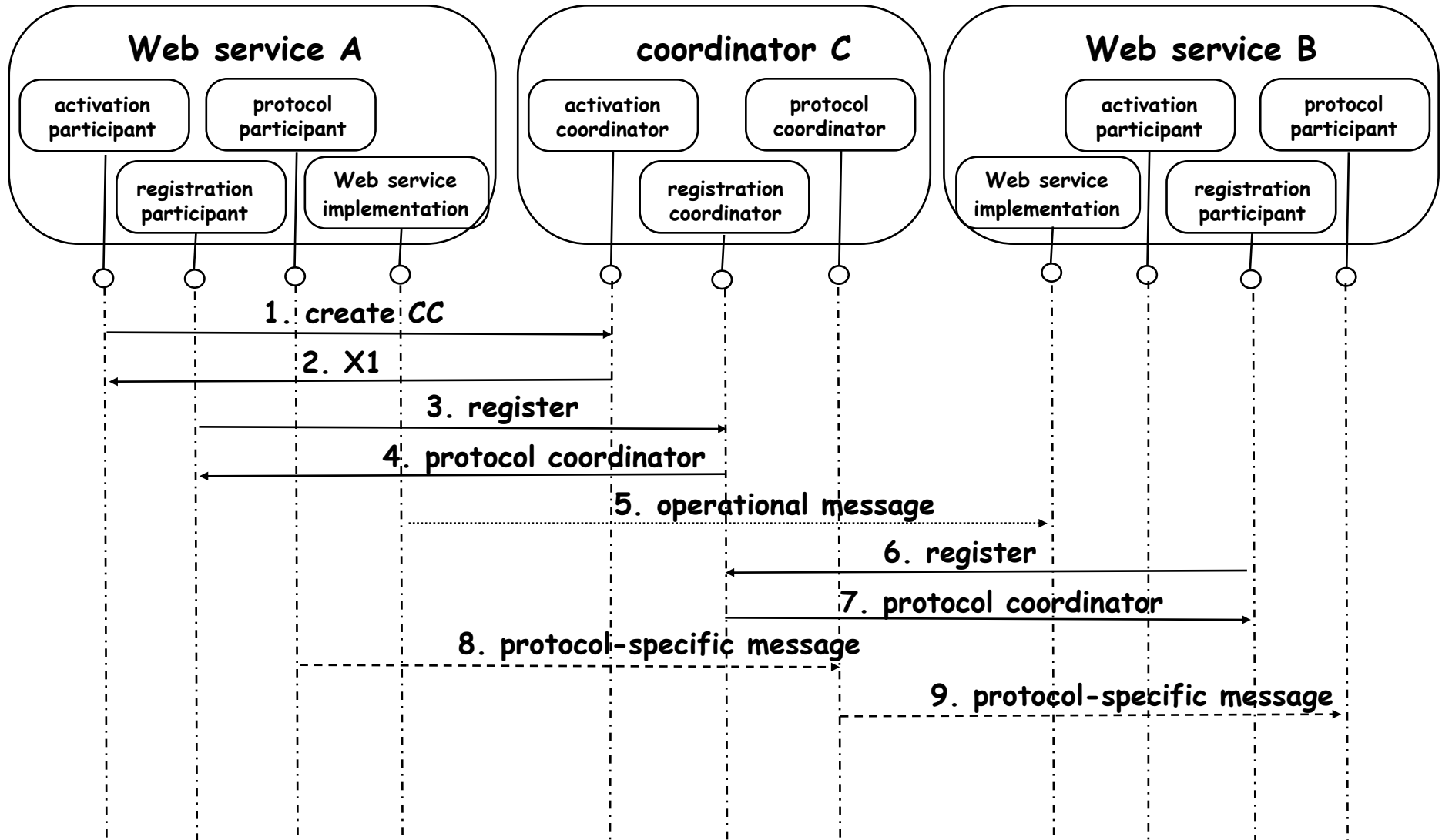


Protocol-Specific

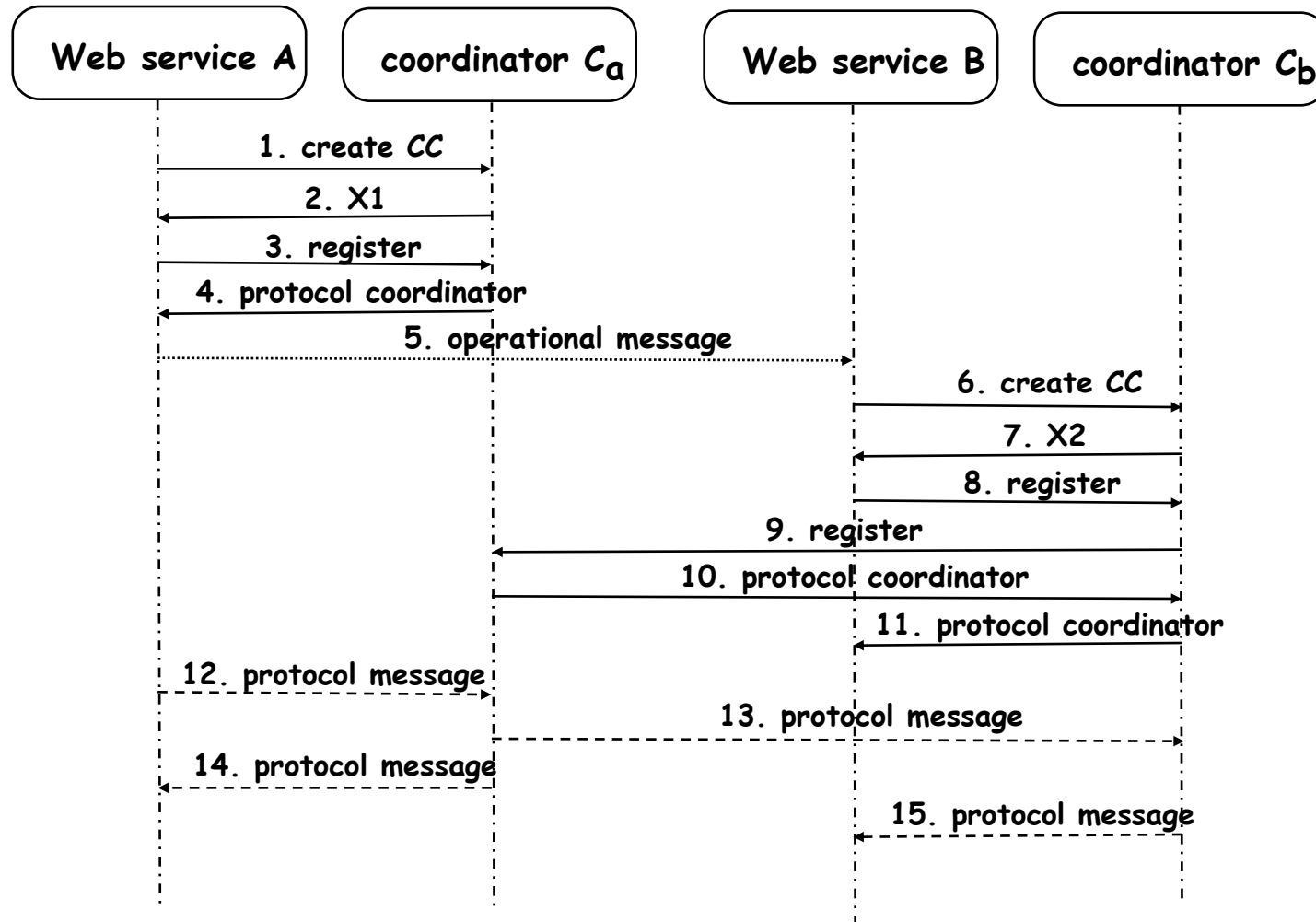
- Convention for protocol specific messages



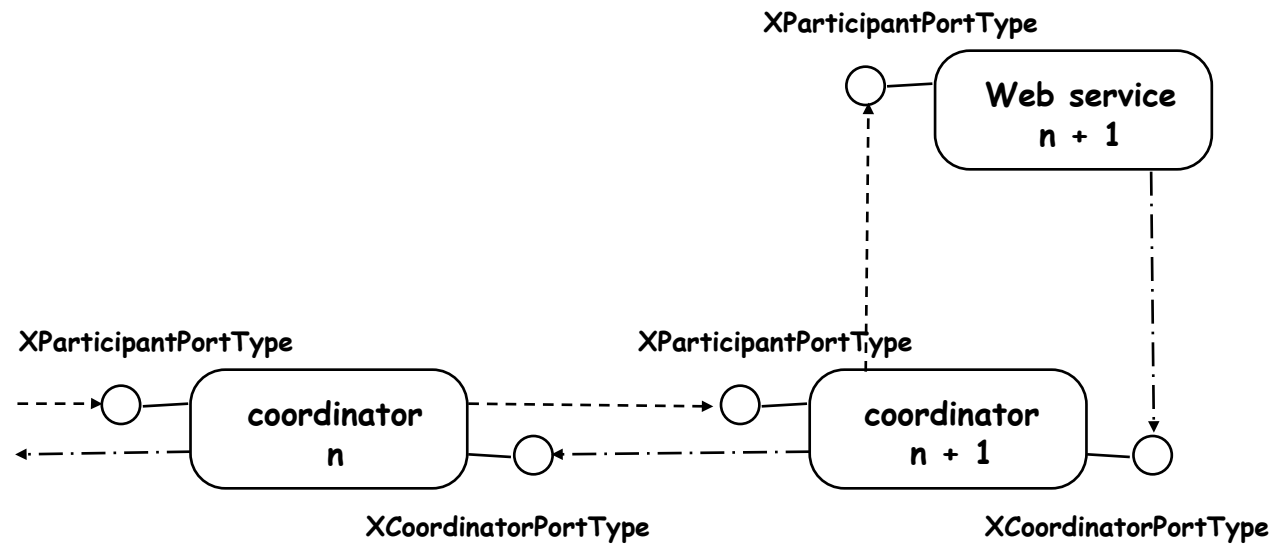
Central Coordinator



Distributed Coordinator



Coordinator Chaining

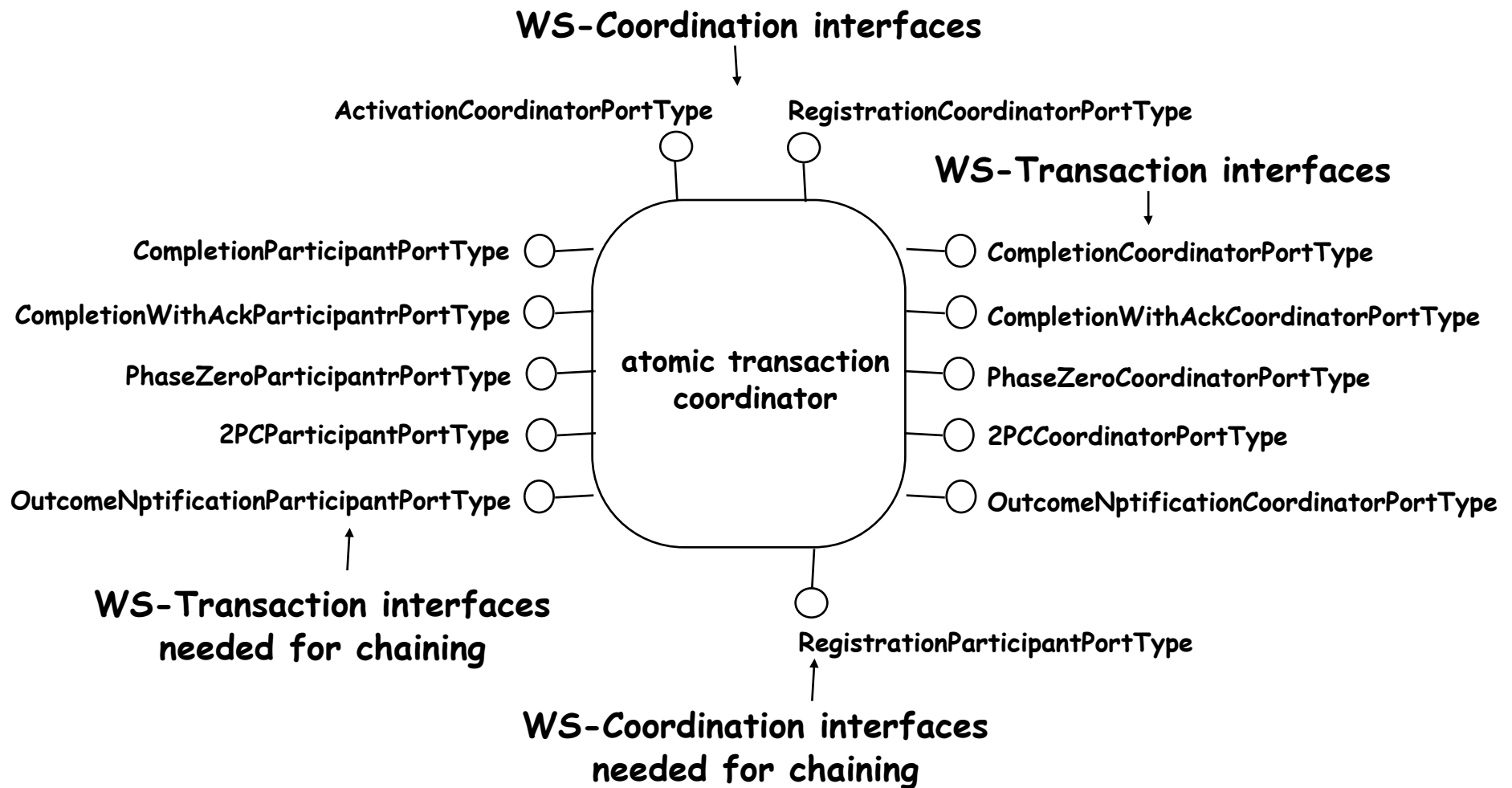


- Defines needed SOAP extensions
- Metaprotocols
 - activation
 - registration
- Middleware for central or distributed coordination

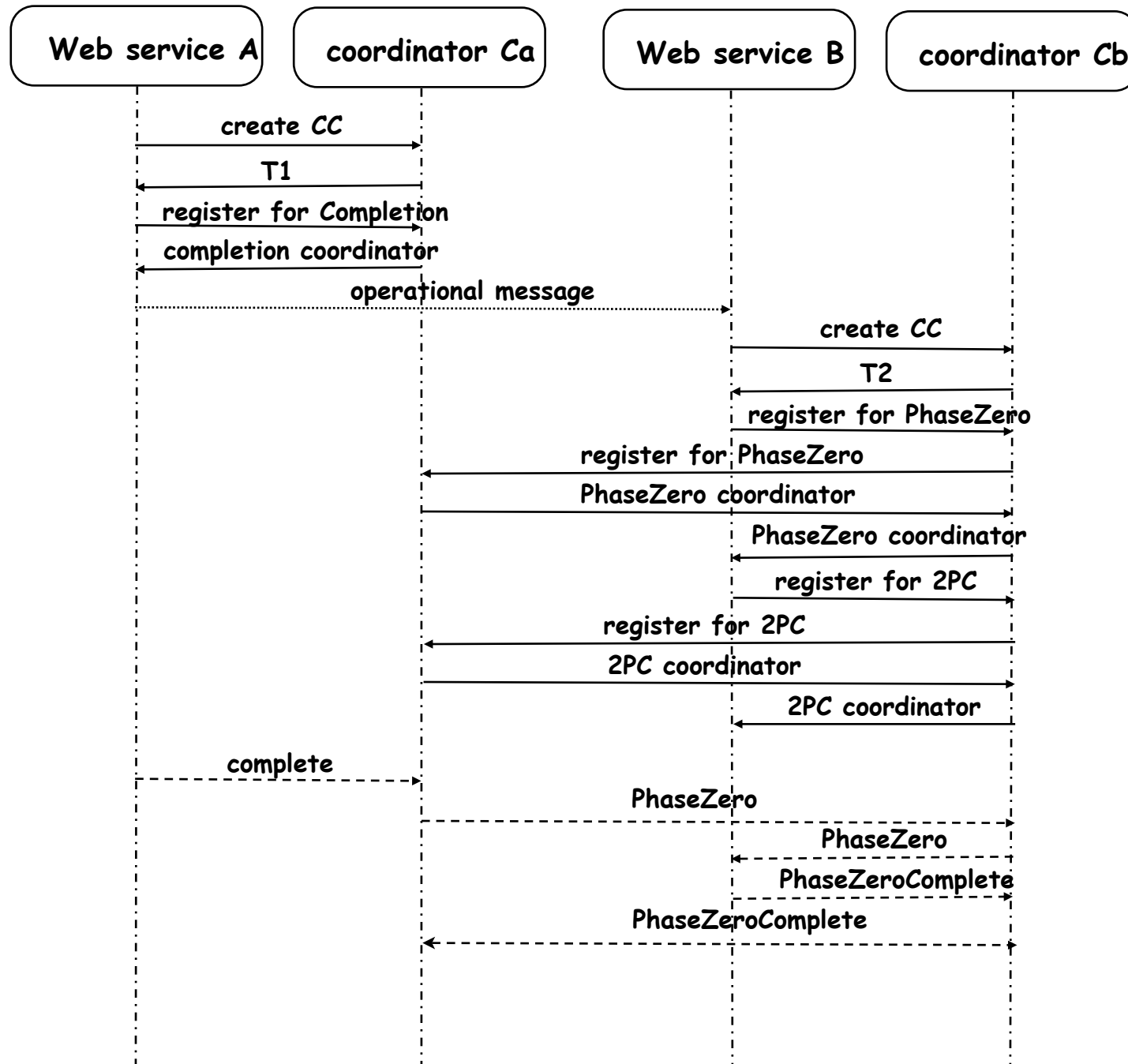
WS-Transaction

- Extension/Use of WS-coordination
- Two flavors:
 - Atomic transaction
 - Business activity

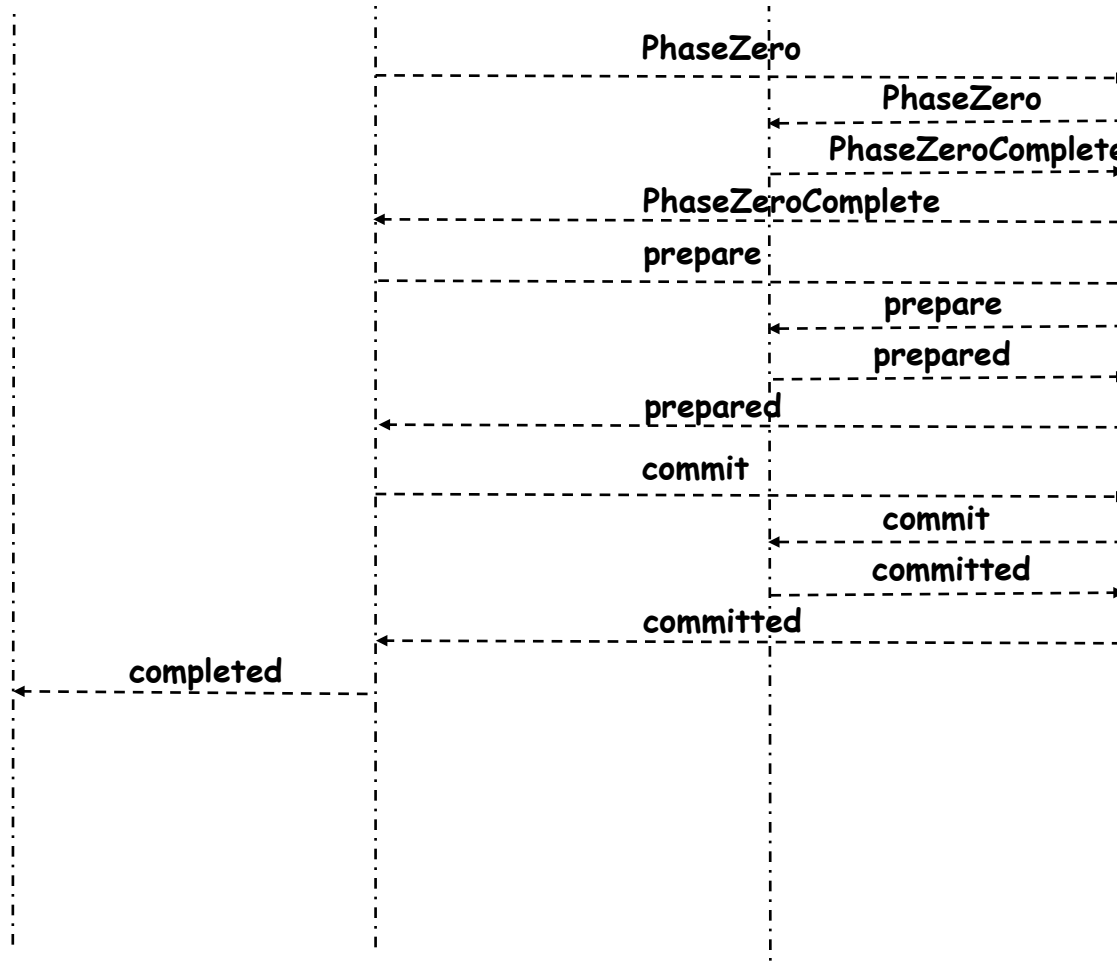
Atomic Transaction - Ports



Atomic Transaction - Protocol



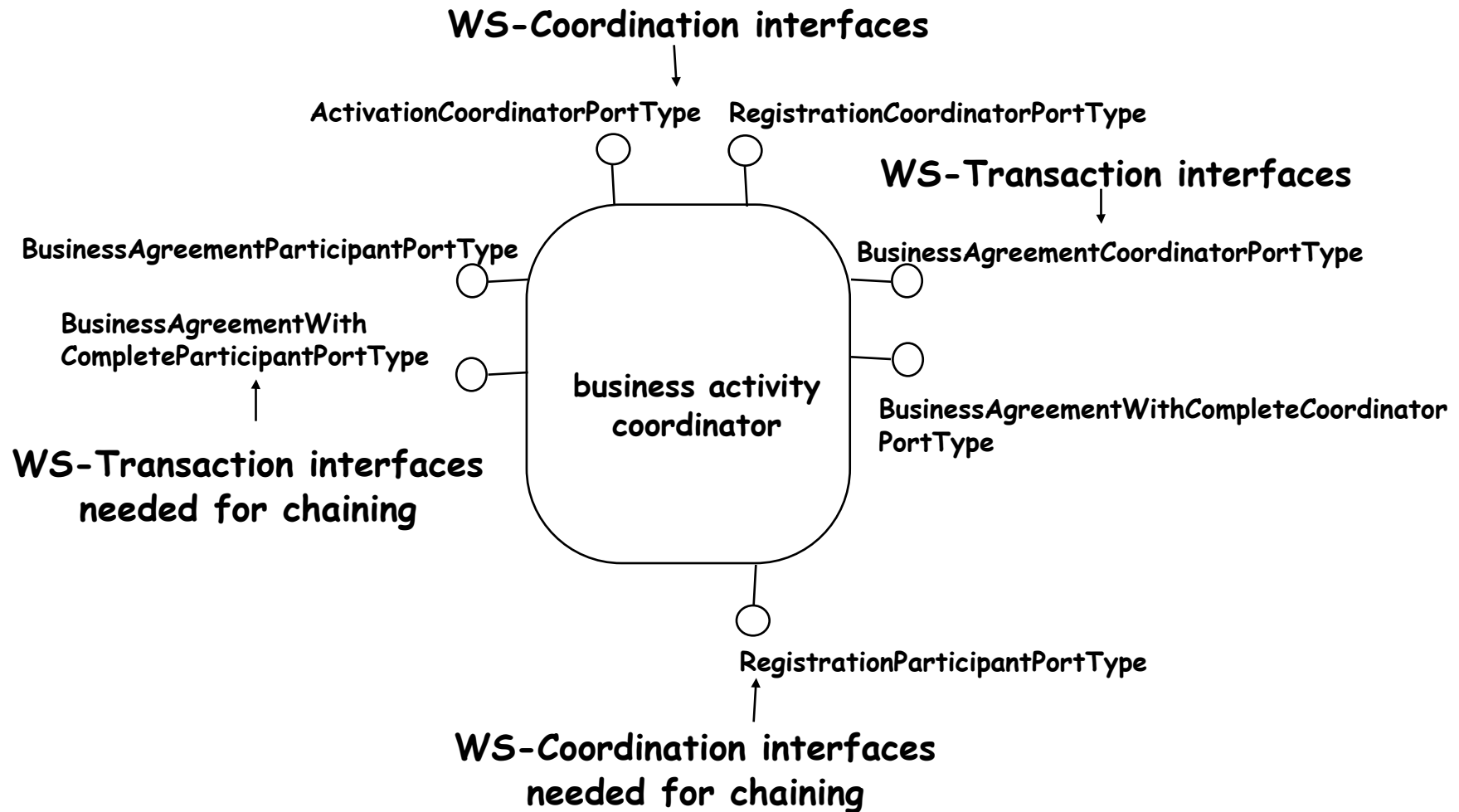
Atomic Transaction - Protocol



Business Activities

- Business Agreement (with Complete)
 - participant initiates by informing the coordinator of its status:
 - exited/completed/faulted
 - Coordinator replies
 - close/complete/compensate/forget

Port Types



Execution

