Orchestration, Choreography and Collaboration in Web Services and ebXML

Ron Ten-Hove
Staff Software Engineer
Sun Microsystems

Goals

Examine message sequencing in Web Services and Business Web Services

Ron Ten-Hove

- Staff Engineer, Sun Java™ Web Services Business Integration Product Development
  - Developed EAI Products for Forté Software (now part of Sun Microsystems)
  - EAI Web Services
  - ebXML BPSS
  - WSCI
  - JSR 208
- Et cetera

The next important area of development of Web Services is so-called Choreography. Many new concepts and specifications are emerging.

Agenda

- Definitions
- Abstract Models
- Web Services
- Business Web Services
- Summary

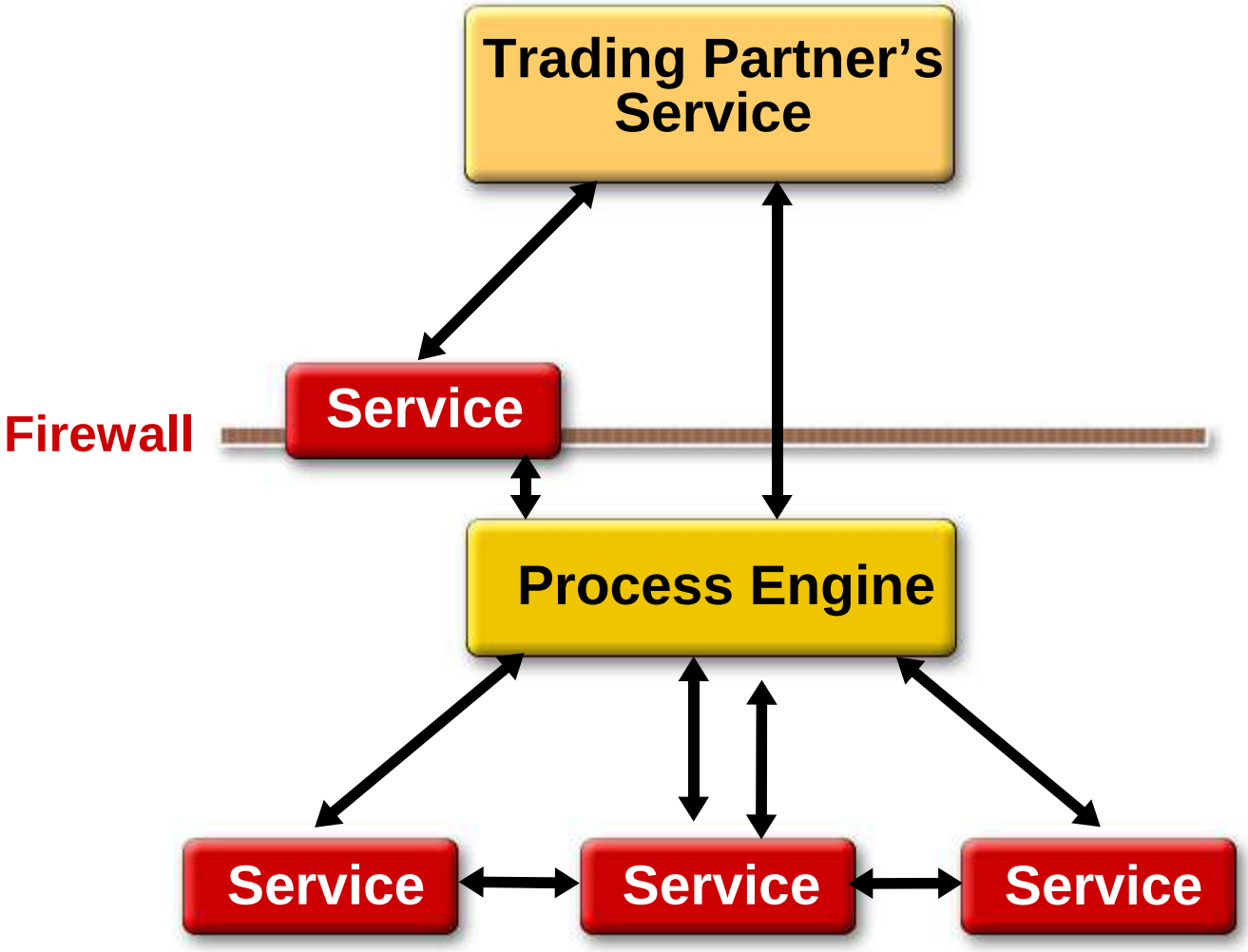
Service-Oriented Architecture (SOA)

- Services, Services Everywhere
- Business Processes:
  - Use Services
  - Provide Services
- Useful EAI Approach
- SOAP + WSDL = Web Services
- ebXML Business Web Services

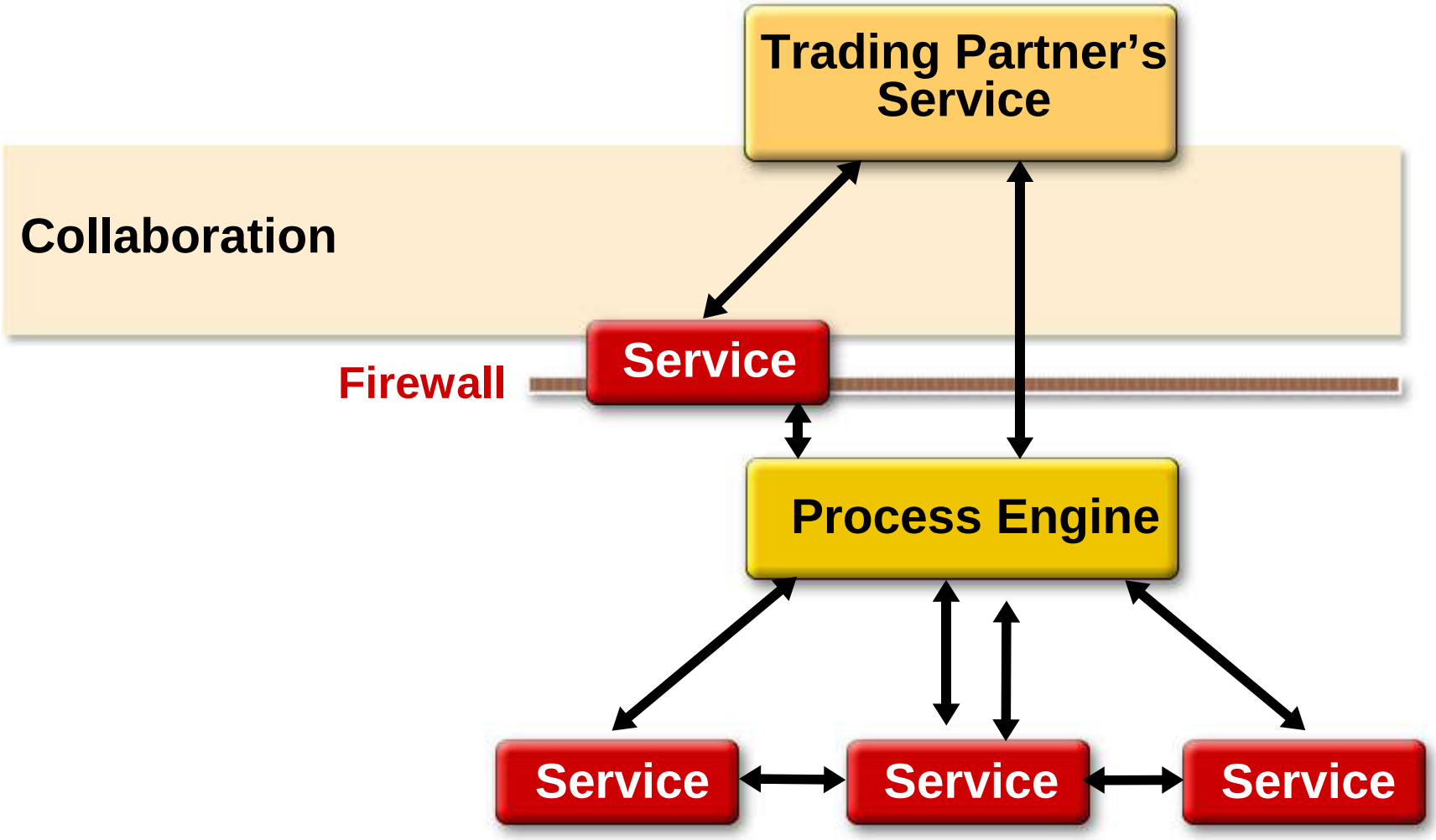
Some Terminology (WS Flavored)

- Collaboration
  - Two or more peers
  - Business-oriented goals
- Orchestration
  - Running business processes
- Choreography
  - Sequencing “interface”
  - Client/Server

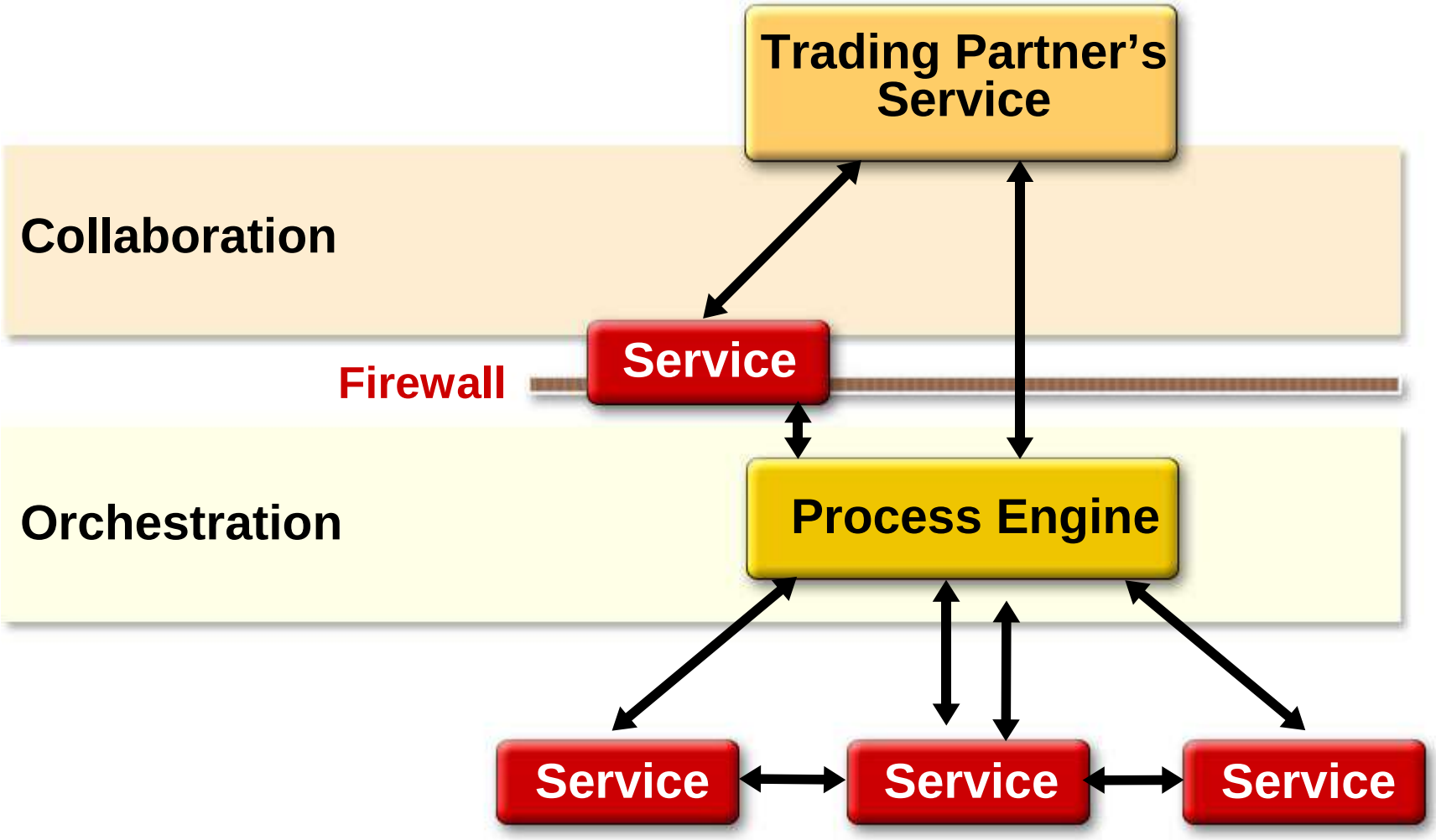
Definitions (Cont.)



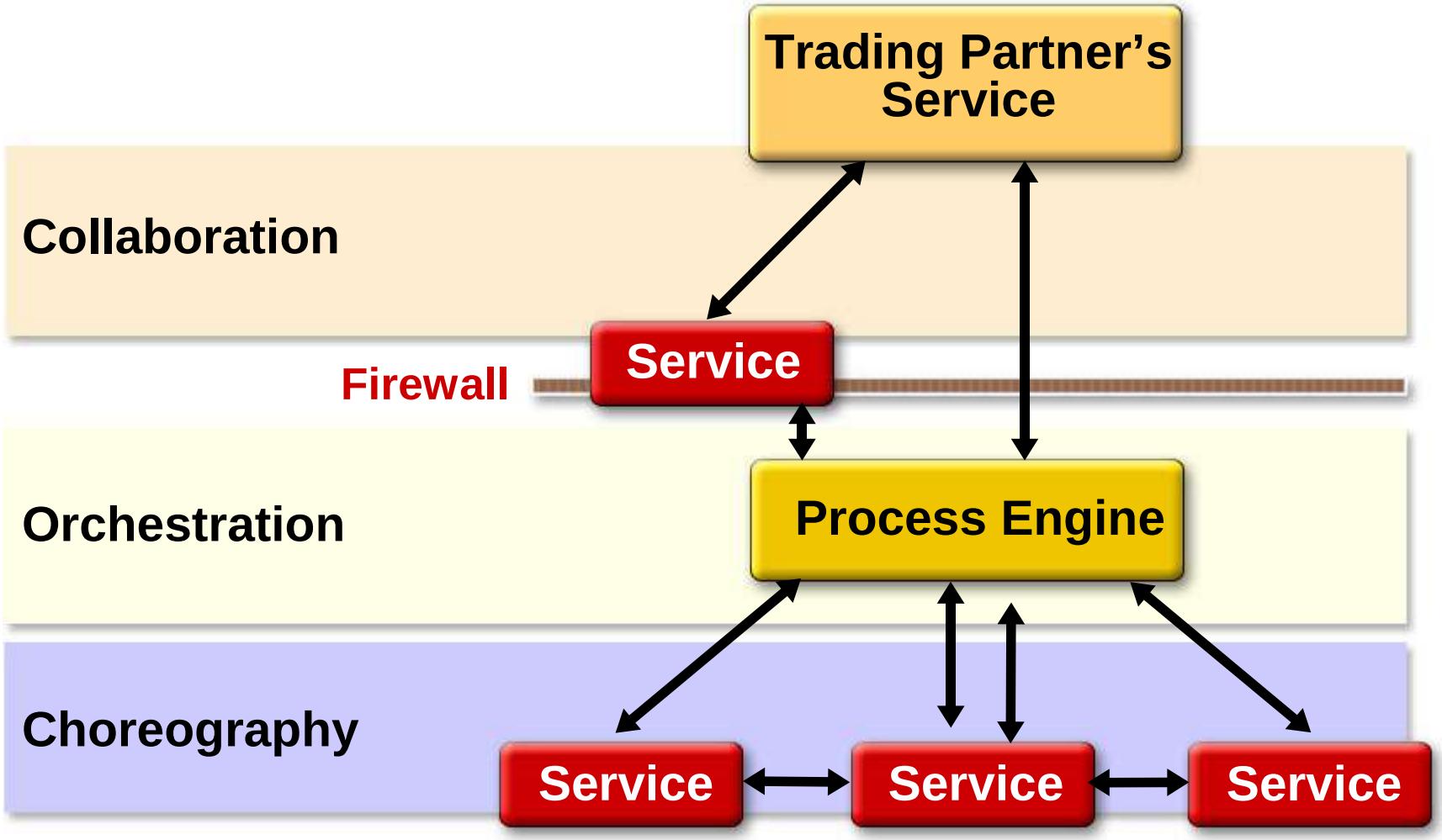
Definitions (Cont.)



Definitions (Cont.)



Definitions (Cont.)



Choreography

- Sequencing of Service operations
- Direct dependencies
- Interface
  - Observable messages
- Constrains client using Service

Orchestration

- Process Engine
  - Invokes Services
  - Offers Services
  - Stateful
  - Complex rules:
    - Sequencing
    - Concurrent sequences (flows)
    - Synchronization of flows
- Conforms to Choreography for each Service used

Collaboration

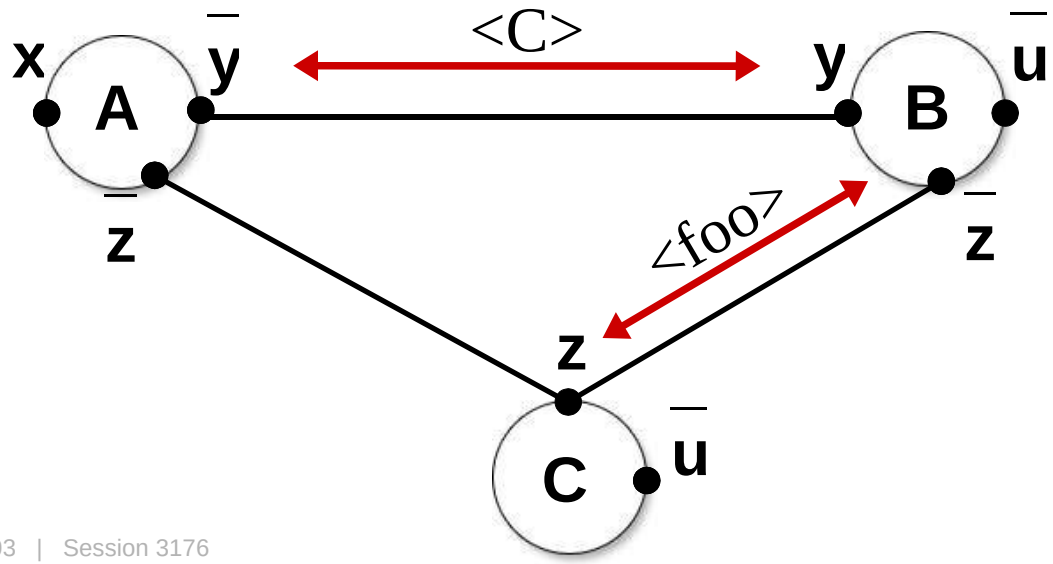
- B2B (Peer-to-peer)
- Message exchange only
- Like paired services, but
  - Uses Internet
  - Trading Partner Agreement
  - Business Semantics
- Electronic Data Interchange (EDI)

Abstract Models

- Abstract Models
  - Influence Specifications
  - Model Checking
  - Emphasize certain aspects
- π-Calculus
- Petri Nets
- State Diagrams/Activity Diagrams

π-Calculus

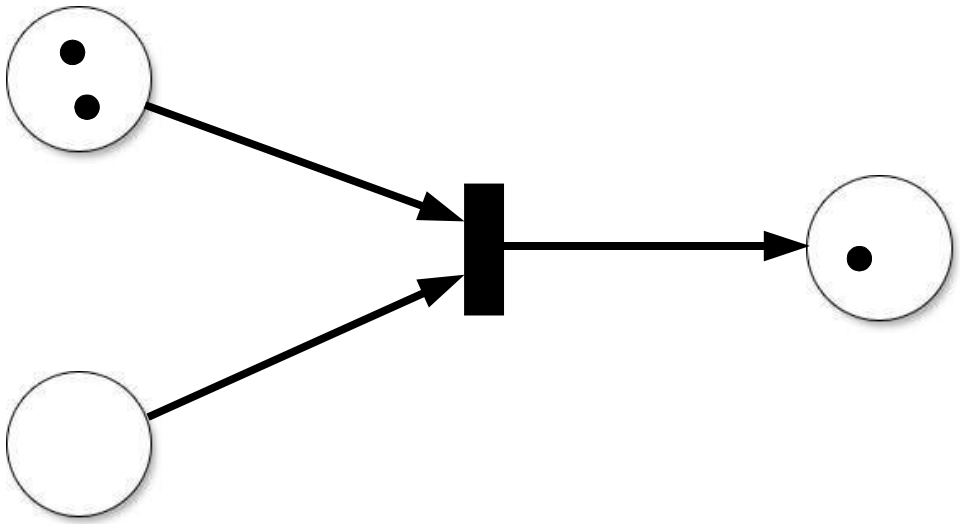
- Model of concurrent computation
- Interacting, named agents
- Dynamic configuration
- “Calculus of Mobile Processes”
- Influence on newer process specifications



Petri Nets

- Model of concurrent, interacting systems
- Graph of
  - Places
  - Transitions
  - Directed arcs connecting transitions and places
- Models
  - Collaboration
  - Orchestration
- Good for modeling, not execution

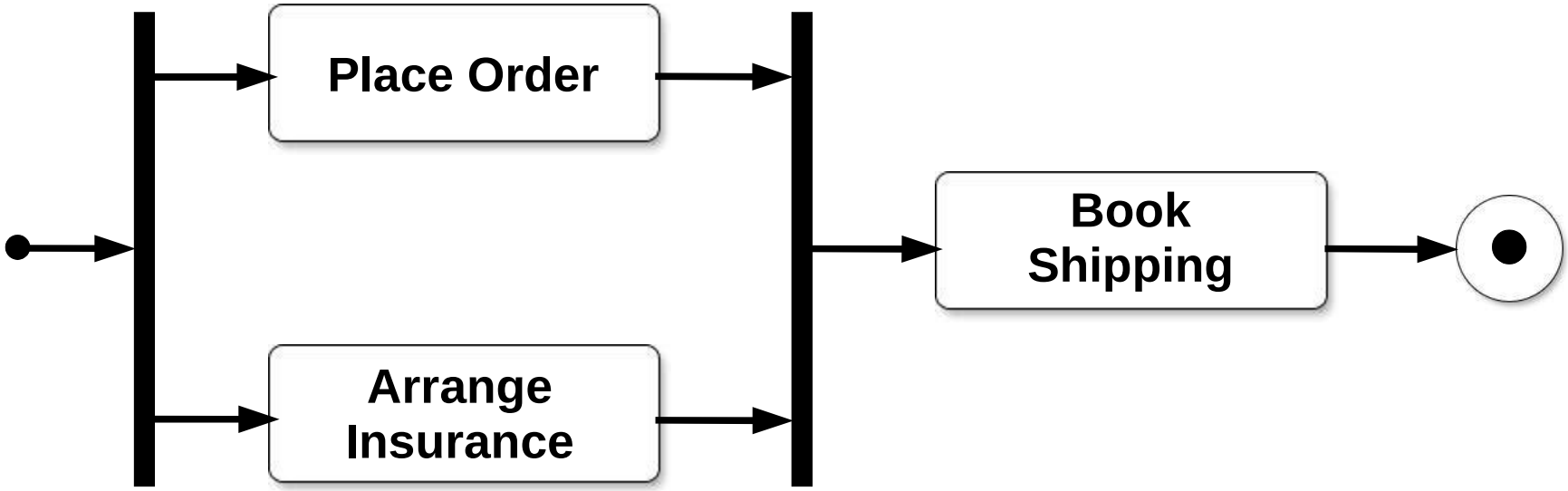
Petri Nets



State Machine

- States linked by directed transitions
- More detail than Petri net
- Simple synchronization primitives
- Models
  - Simple orchestration
  - Most choreographies
- UML StateChart, Sequential Function Chart, Grafcet, etc.

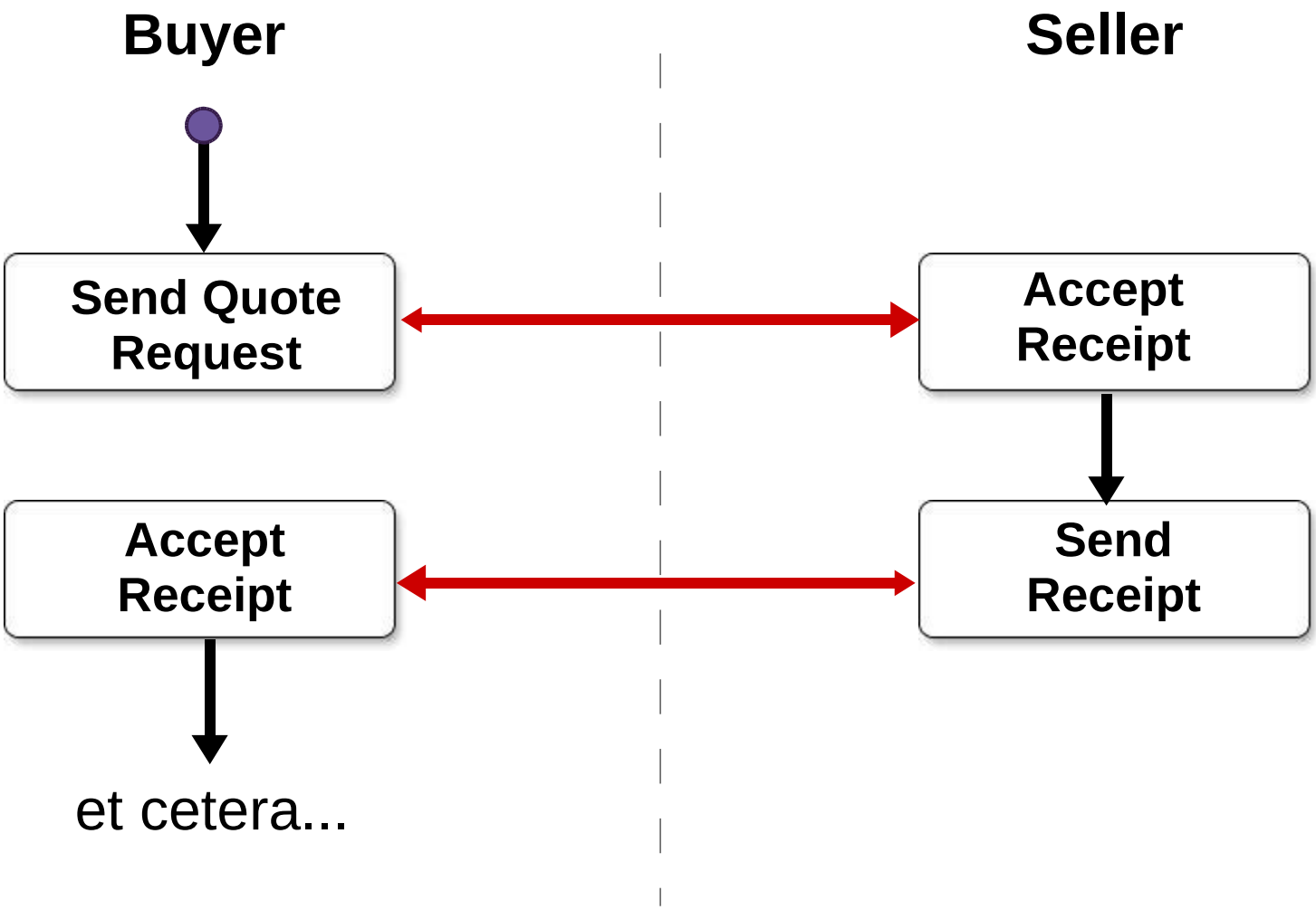
State Diagram



Activity Diagrams

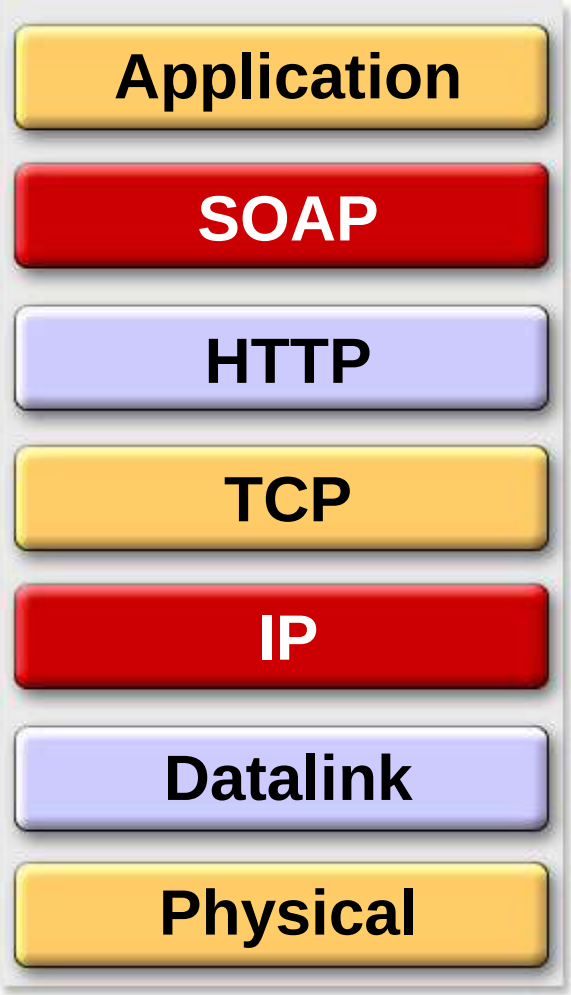
- UML variation of State Machine
- Roles (“Swim lanes”)
- Initiating role
- Message exchange between roles
- Models
  - Collaboration
  - Orchestration (optionally)

Activity Diagrams



Web Services

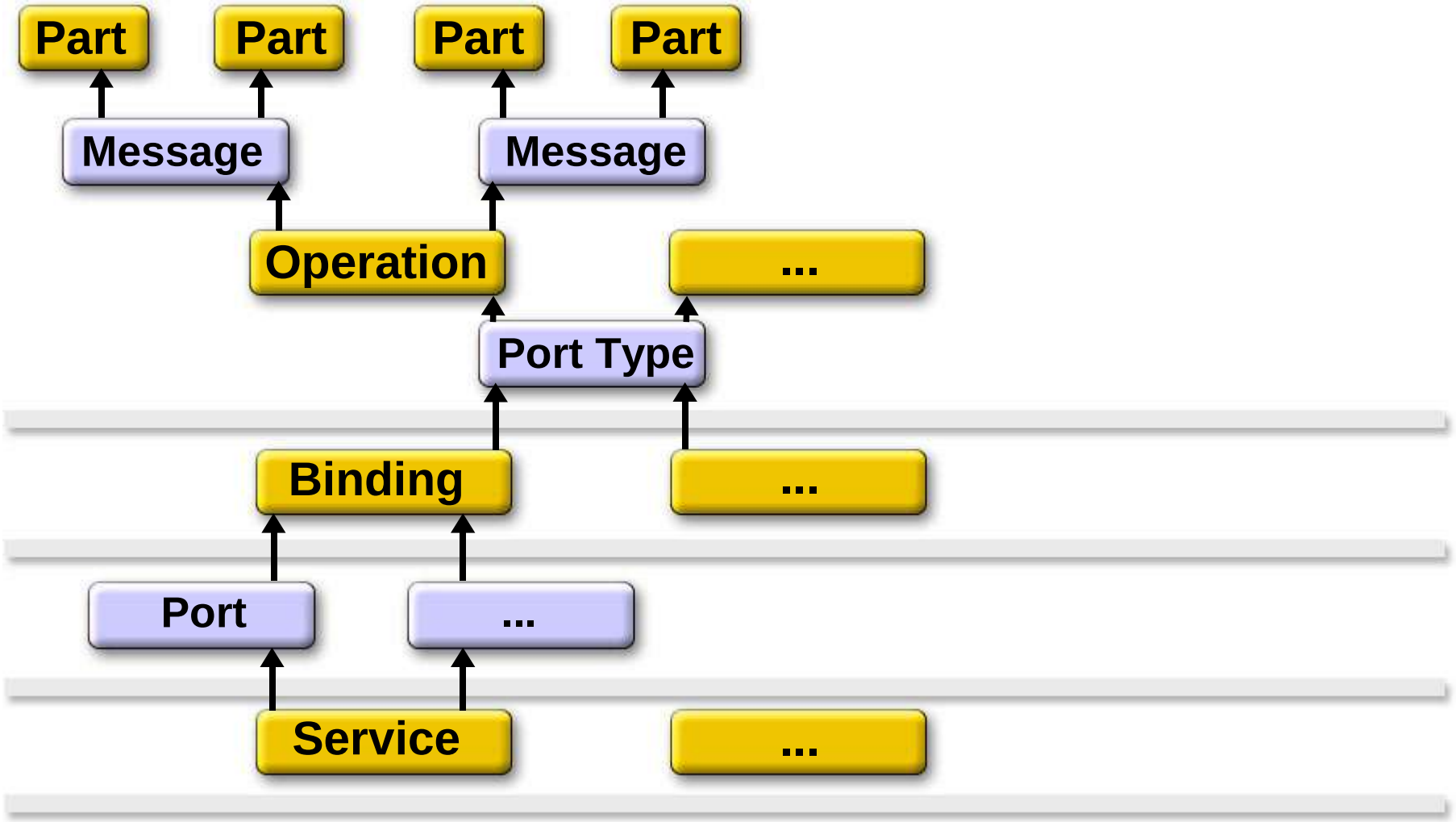
- Technology stack
- Based on web protocols
- Port 80 as a tunnel
- SOAP 1.1
  - RPC
  - Message exchange
- WSDL 1.1
  - Description of service
  - Bind service to SOAP



The Problem With Web Services

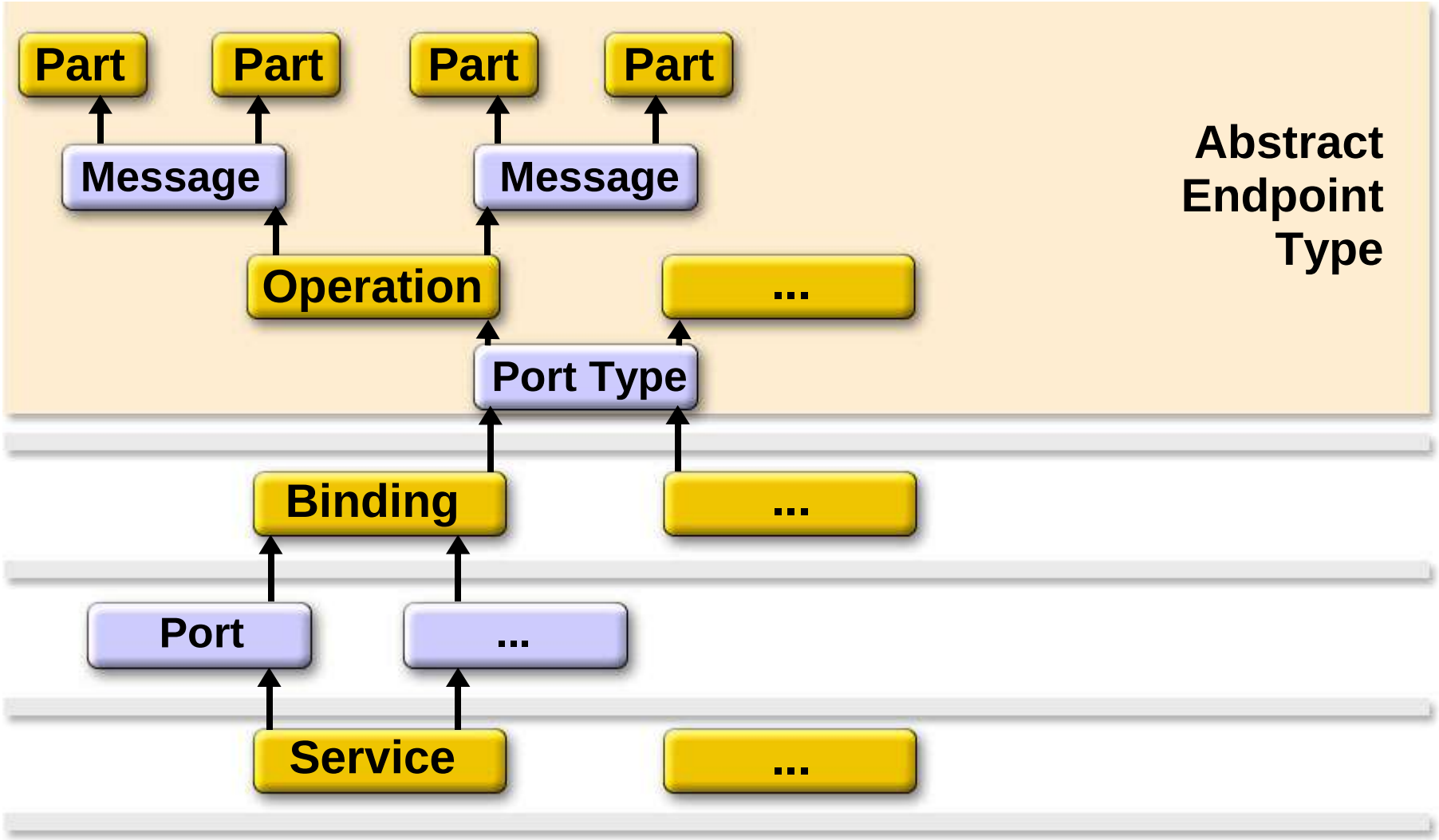
- Web Services Description Language 1.1
  - Operations, Operations Everywhere
  - Services modeled as message exchanges
  - Unordered
- No Choreography
- No Collaboration
  - Paired services pattern
- Orchestration: Not applicable

WSDL Structure



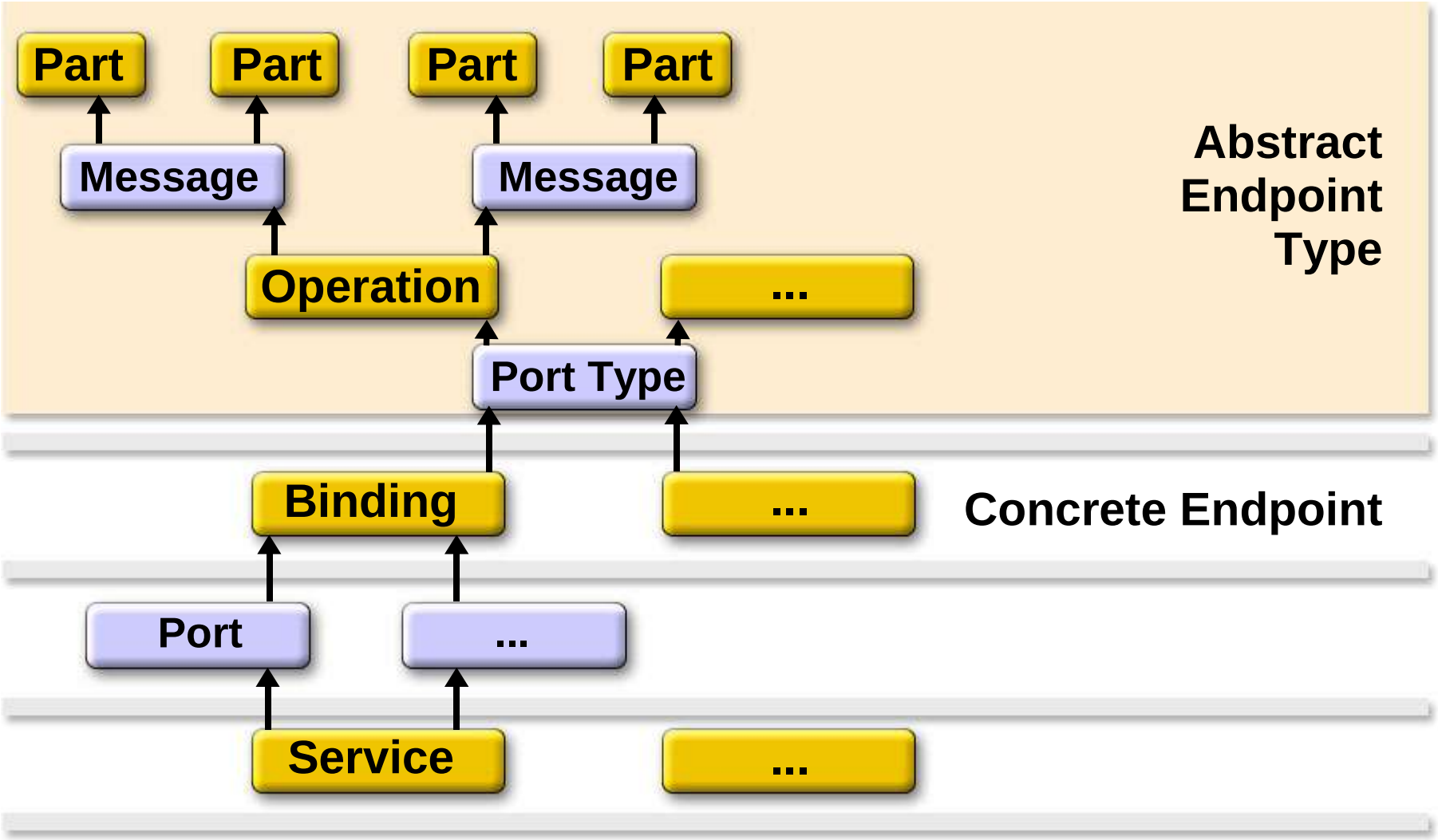
Adapted from: Microsoft MSDN

WSDL Structure



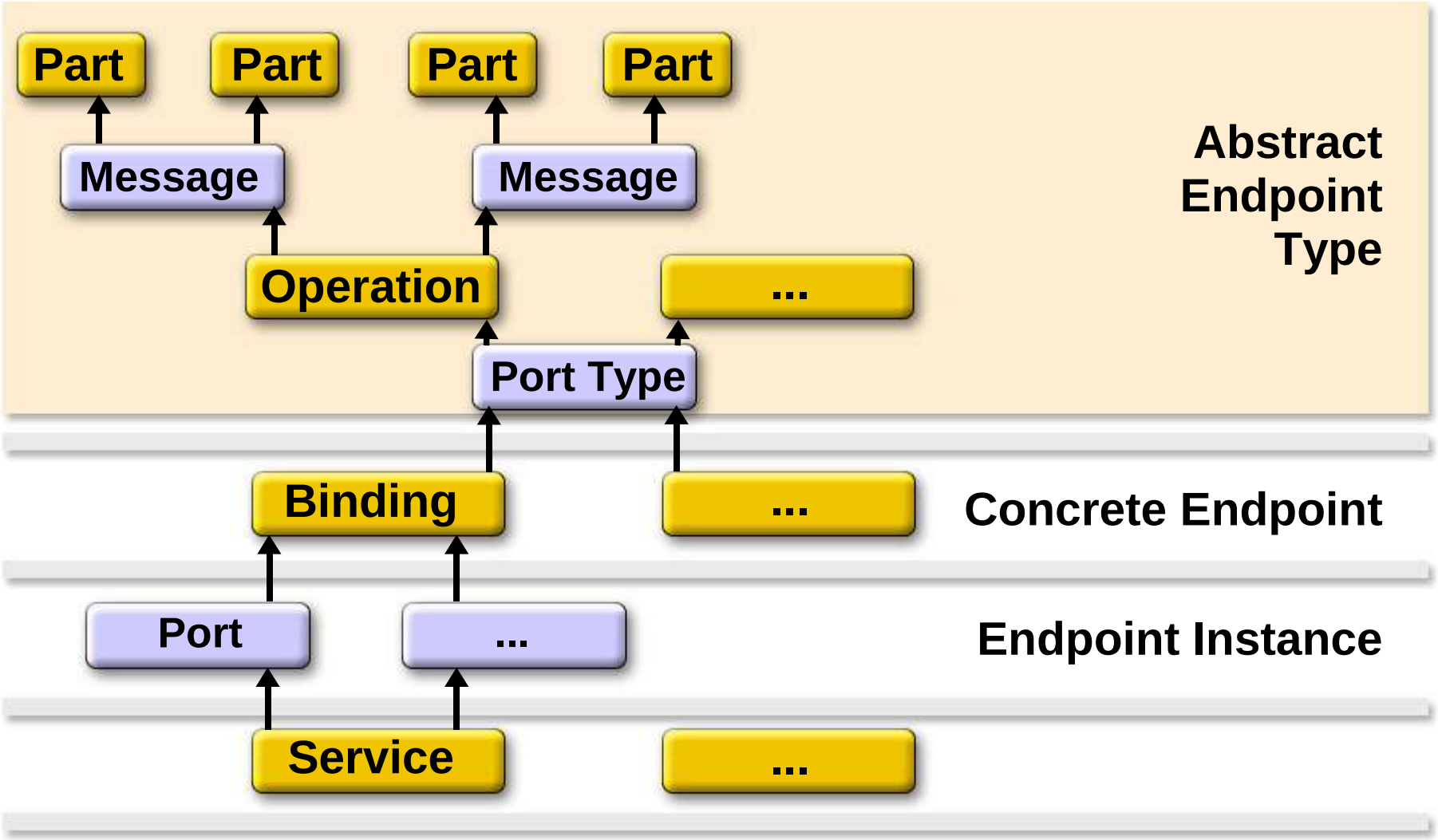
Adapted from: Microsoft MSDN

WSDL Structure



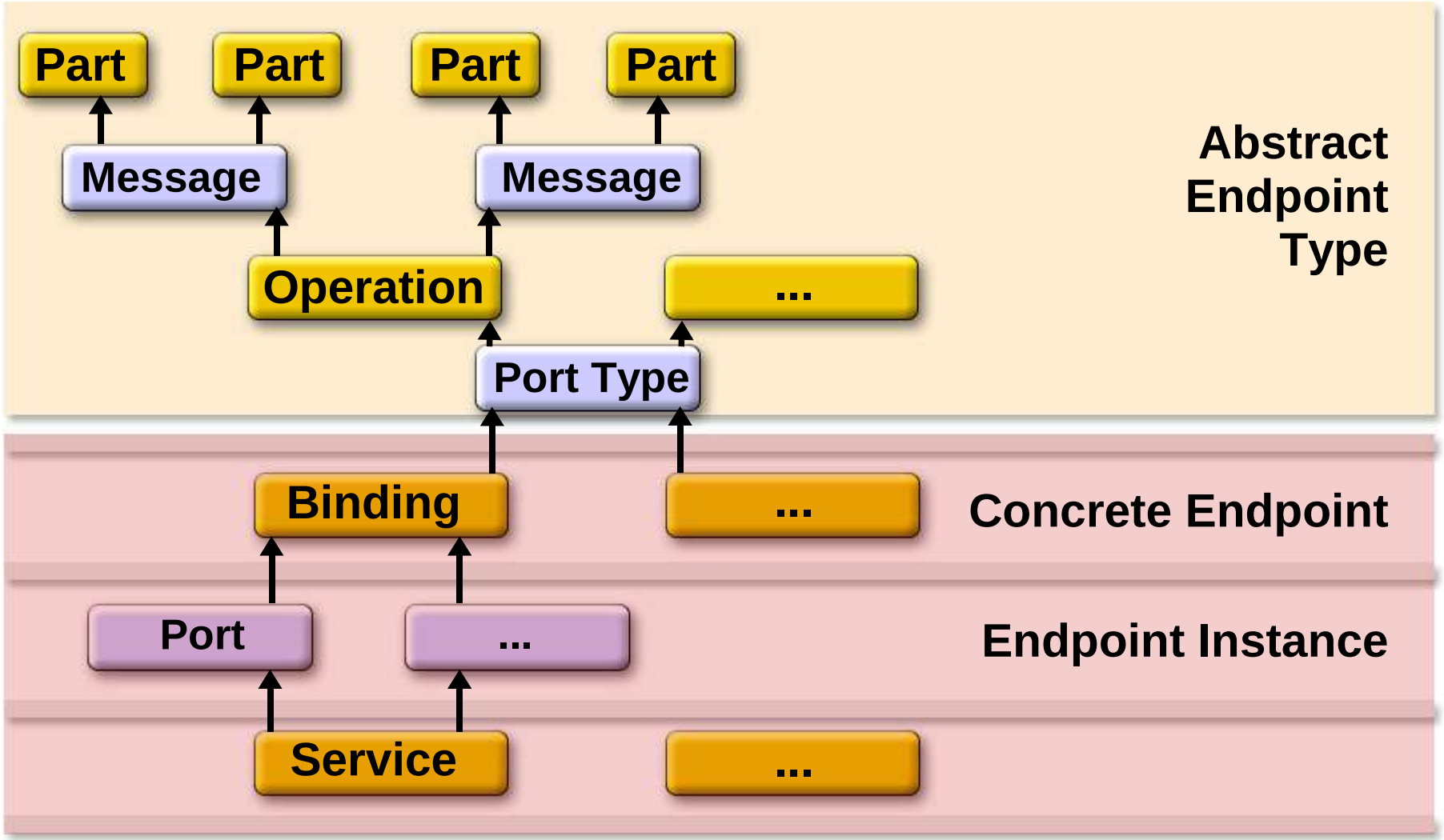
Adapted from: Microsoft MSDN

WSDL Structure



Adapted from: Microsoft MSDN

WSDL Structure

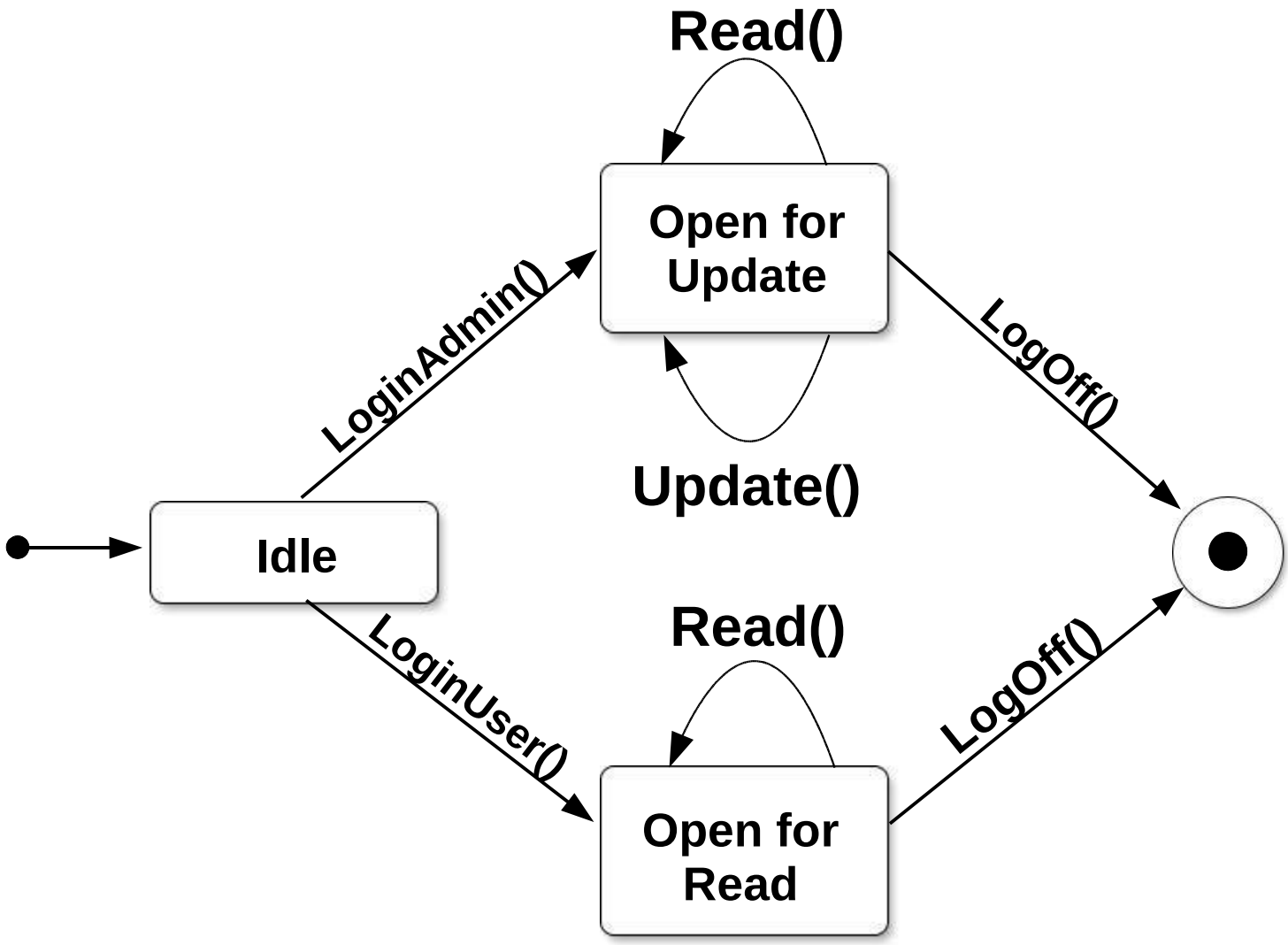


Adapted from: Microsoft MSDN

Web Service Choreography

- Layered on top of PortType
  - Reuse
- Client/Service contract
- Ordering of Operation invocations

Choreography Example



Simple Choreography Isn’t Enough

- Pure Client/Service
  - Only one initiator
  - Synchronous Messaging
- Single Service Only

Specifications to the Rescue

- WSCL 1.0
  - Basic choreography language
  - Layered directly atop WSDL 1.1
- WSCI 1.0
  - Rich choreography language
  - Layered directly atop WSDL 1.1
- BPEL4WS
  - Broad orchestration language
  - Choreography support
  - Layered directly atop WSDL 1.1

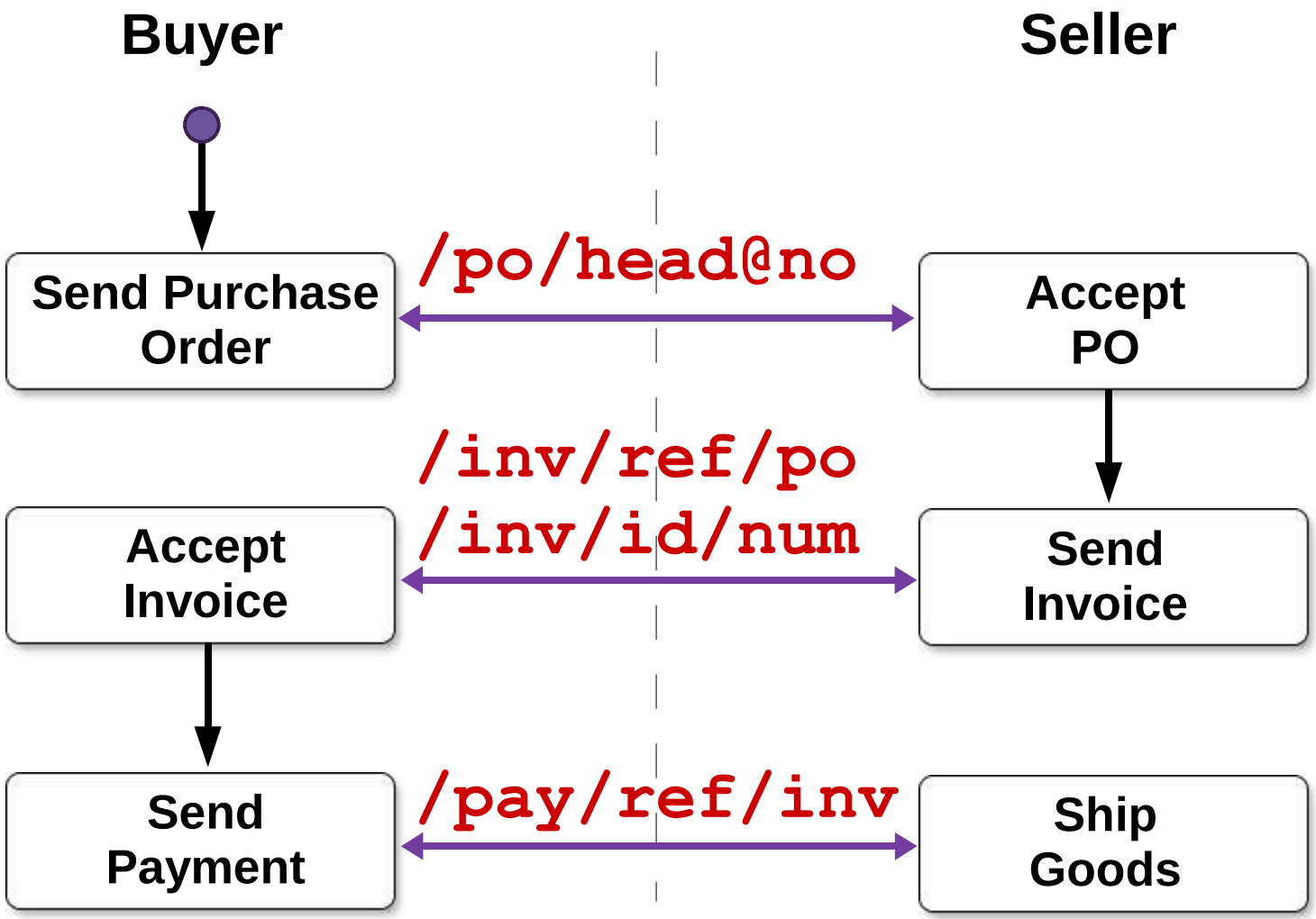
Added Features

- Correlation
- Transactions
- Exceptions
- Compensation
- Roles
- Dynamic Participation

Correlation

- Asynchronous responses
  - How to correlate to request?
- ebXML message meta-data
  - Unique message IDs
  - Ref-to message ID
- Use message content
  - E.g., purchase order number
- Multiple correlation items possible

Correlation Example



Transactions

- Multiple operations complete or fail like a single one
- Various models:
  - ACID—Atomic
    - Locks resources
    - Costly for long-running operations
  - Nested (Open)
    - Relaxes isolation requirement
    - Complex error recovery
- In tension with loose coupling

Exceptions

- Choreography distinguishes:
  - "Happy path"
  - Error paths
    - Technical problems
    - Fault response from Service
      - Multiple types
- Error paths can
  - Retry
  - Complain
  - Clean up

Compensation

- Undo a committed transaction/operation
  - Needed in open transaction model
  - Distinct from rollback
- Explicit compensation handler
  - Sequence of operations to achieve “undo”

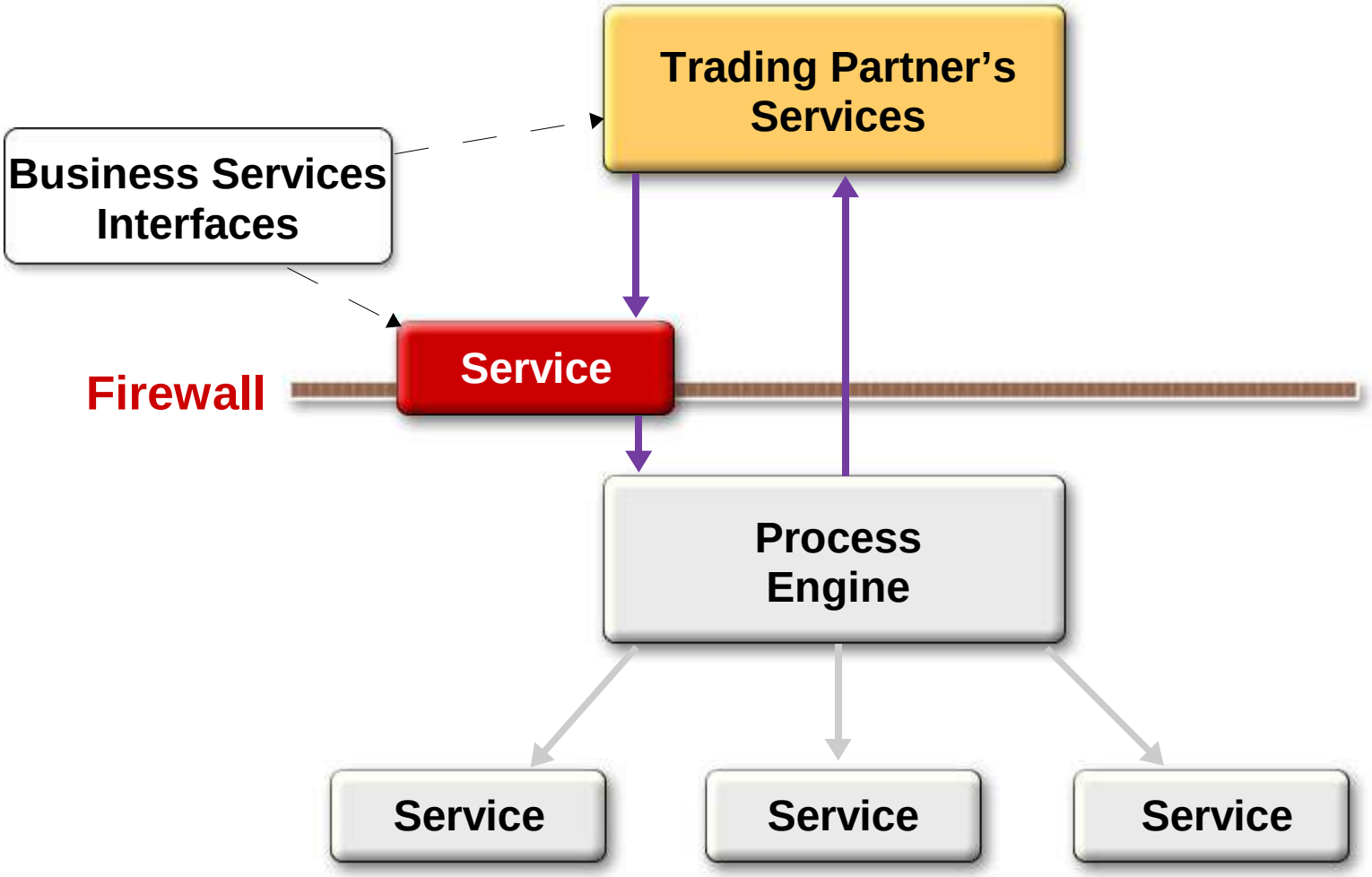
Roles

- Assigning role names to port types
- Pairing roles
  - E.g., Buyer, Seller
  - Links port types directly

Dynamic Participation

- Selection of target service at run-time
- Uses dynamic data
  - Message contents
  - Previous message contents
- π-Calculus influence
  - Service address is a channel
  - Port type is channel type

Business Web Services



Business Web Services

- Web Services Plus:
  - Reliability
  - Security
  - Direct support for business semantics
- EDI—Electronic Data Interchange
  - Electronic business documents
  - Many data formats
  - Value Added Network—VAN
  - Expensive

Business Web Services

- Binary or multi-party collaborations
- Complex sequencing
- Trading partner agreements (contracts)
- Peer-to-peer shared views:
  - State management
  - Business semantics, rules, constraints
- Integration with internal business systems:
  - Business processes
  - Applications

Business Web Services Are Different

- B2B Transactions
  - Coarse-grained
    - Map to business activities
  - Have economic and legal consequences
  - No shared infrastructure
- Internet
  - Insecure
  - Unreliable
  - Ubiquitous, cheap

ebXML for Business Web Services

- “Cheap” EDI:
  - Internet protocols
    - Reliable, secure messaging atop SOAP
  - XML for data
  - Standard business documents:
    - CC—Core Components
    - UBL—Universal Business Language
  - Standards based, interoperable

ebXML for Business Web Services

- Suite of loosely-coupled specifications
  - CPP/CPA
    - Trading partner agreement
  - Message Handler
    - Secure, reliable messaging over Internet
    - CPA configures handler quality-of-service for each transaction
  - BPSS
    - Binary and multi-party collaboration
    - Business transaction model
  - Core Components, UBL

CPP/CPA

- Ultimate WSDL
- Trading partner agreement
  - Roles assigned to partners
  - All messages described
  - Endpoints defined
  - All message exchanges defined
    - Quality of service
      - Security
      - Reliability

Message Handler

- Configured by CPA(s)
- Provides
  - Security
    - Confidential
    - Tamper-proof
    - Authentication of sender
  - Reliability
    - Once-and-only-once delivery
- Interoperable
  - Many implementations available

BPSS

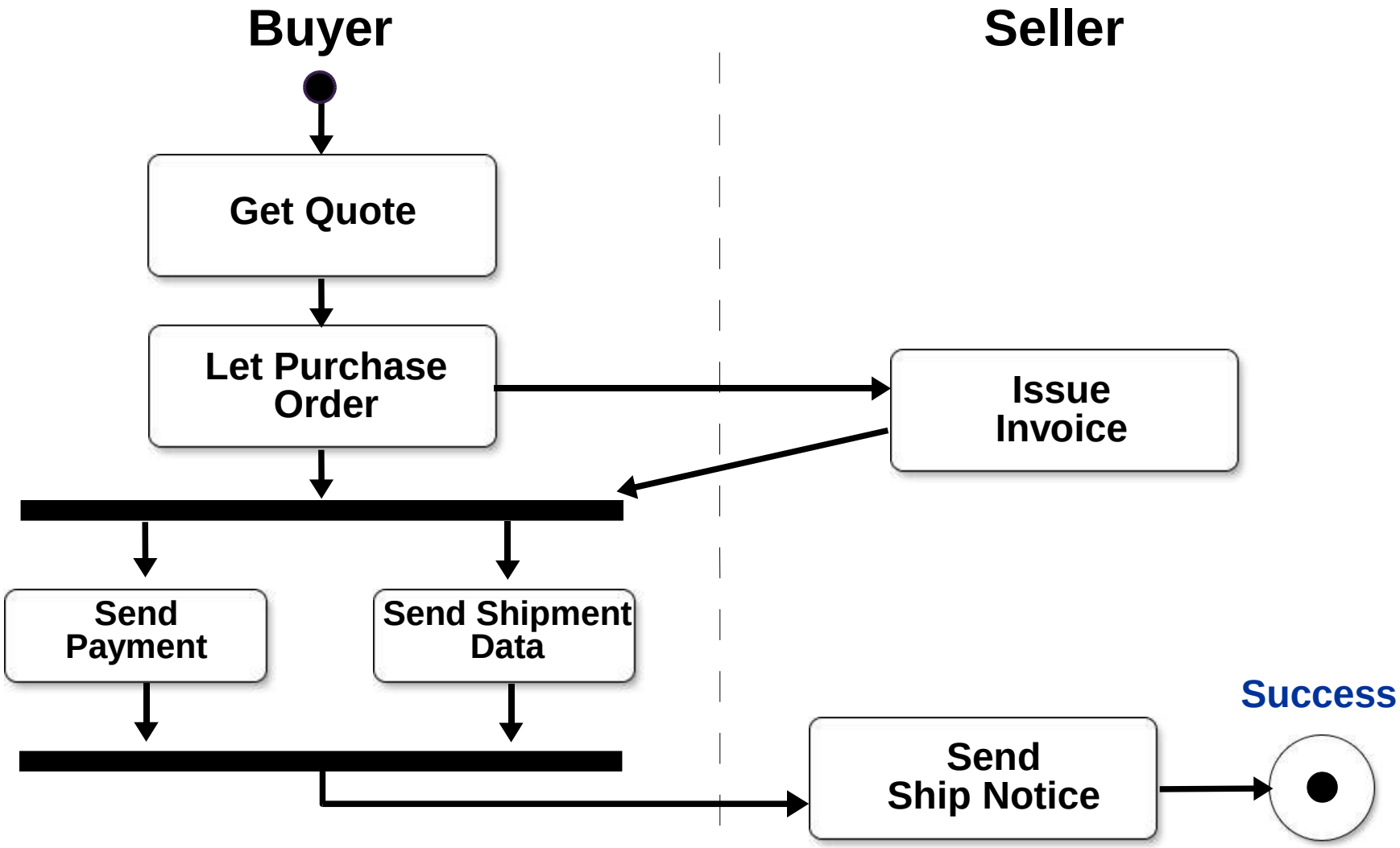
Business Process Specification Schema

- Collaborative business process definition
  - Role-based
  - Binary and multi-party
- Defined atop abstract model of MSH
- Shared view of process state
  - Message exchange
    - business signal protocol
  - Common process definition
- Rich set of business-oriented features

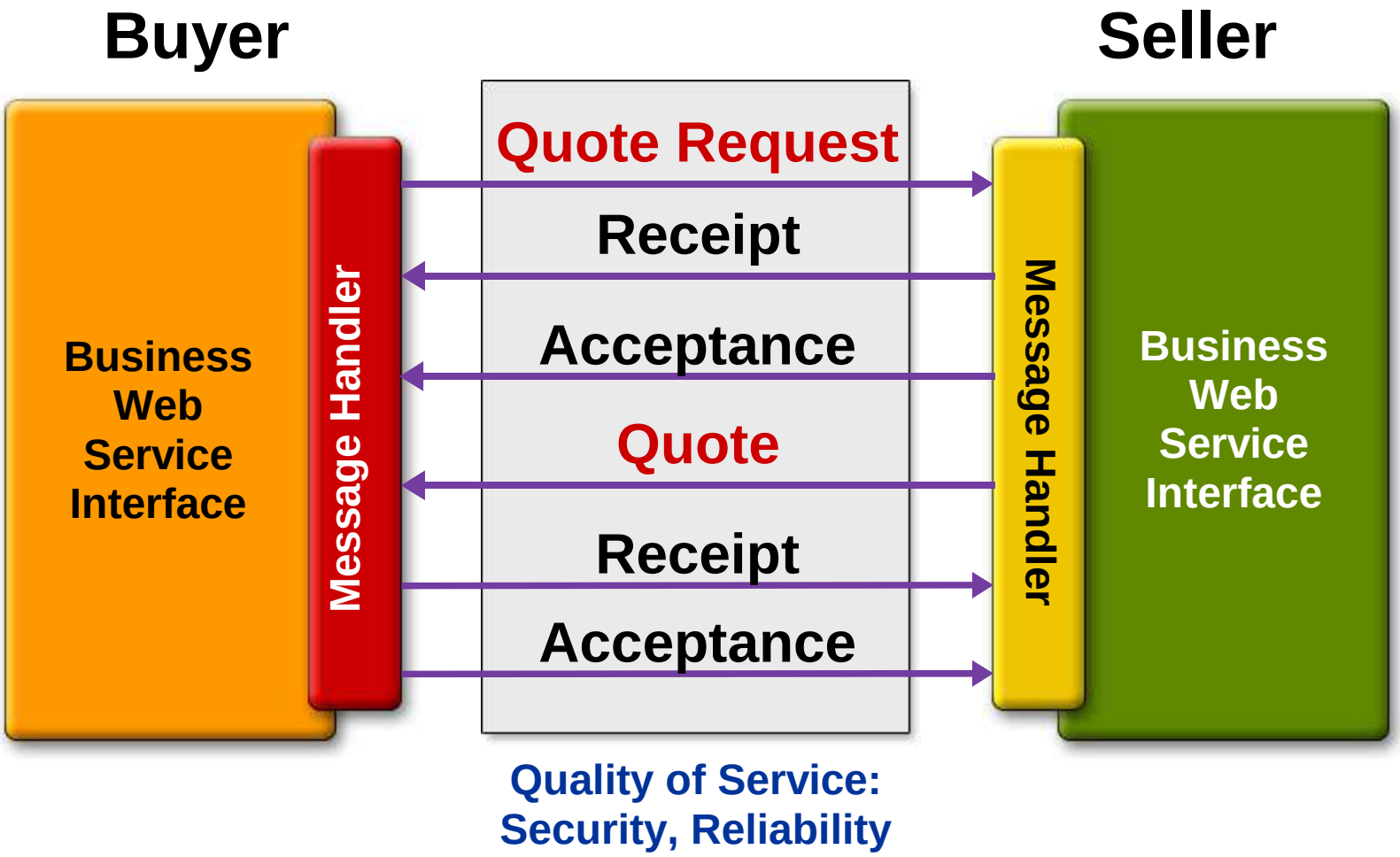
BPSS

- Process models similar to UML Activity Diagrams
  - Stereotyped activity “Business Transaction”
    - Business protocol
      - Multiple document exchanges
      - Time outs
    - Simple set of result types
      - Succeeded,
      - Failed for technical reasons, failed for business reasons, etc.
  - Roles mapped to swim lanes
  - Concurrent activities

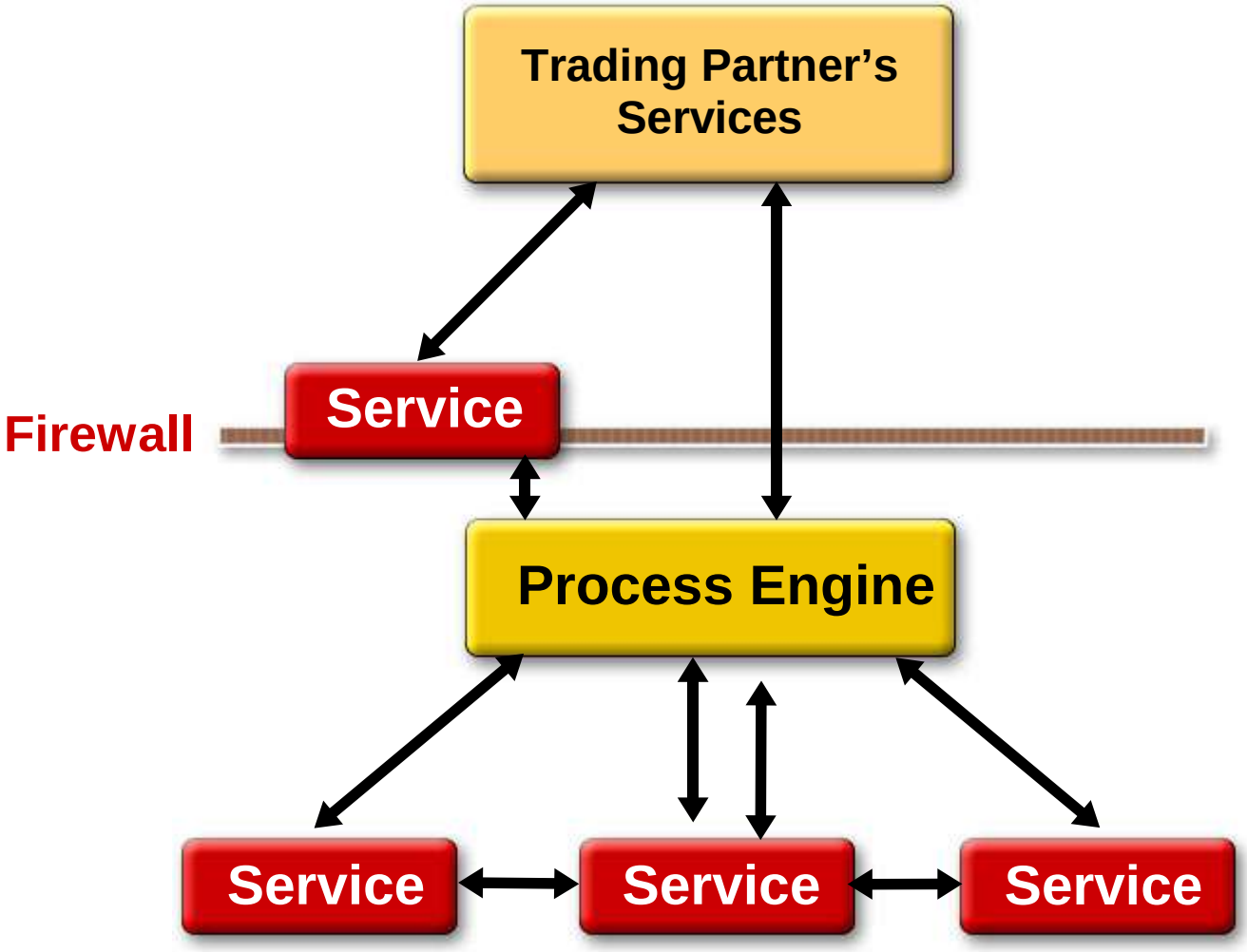
BPSS Example



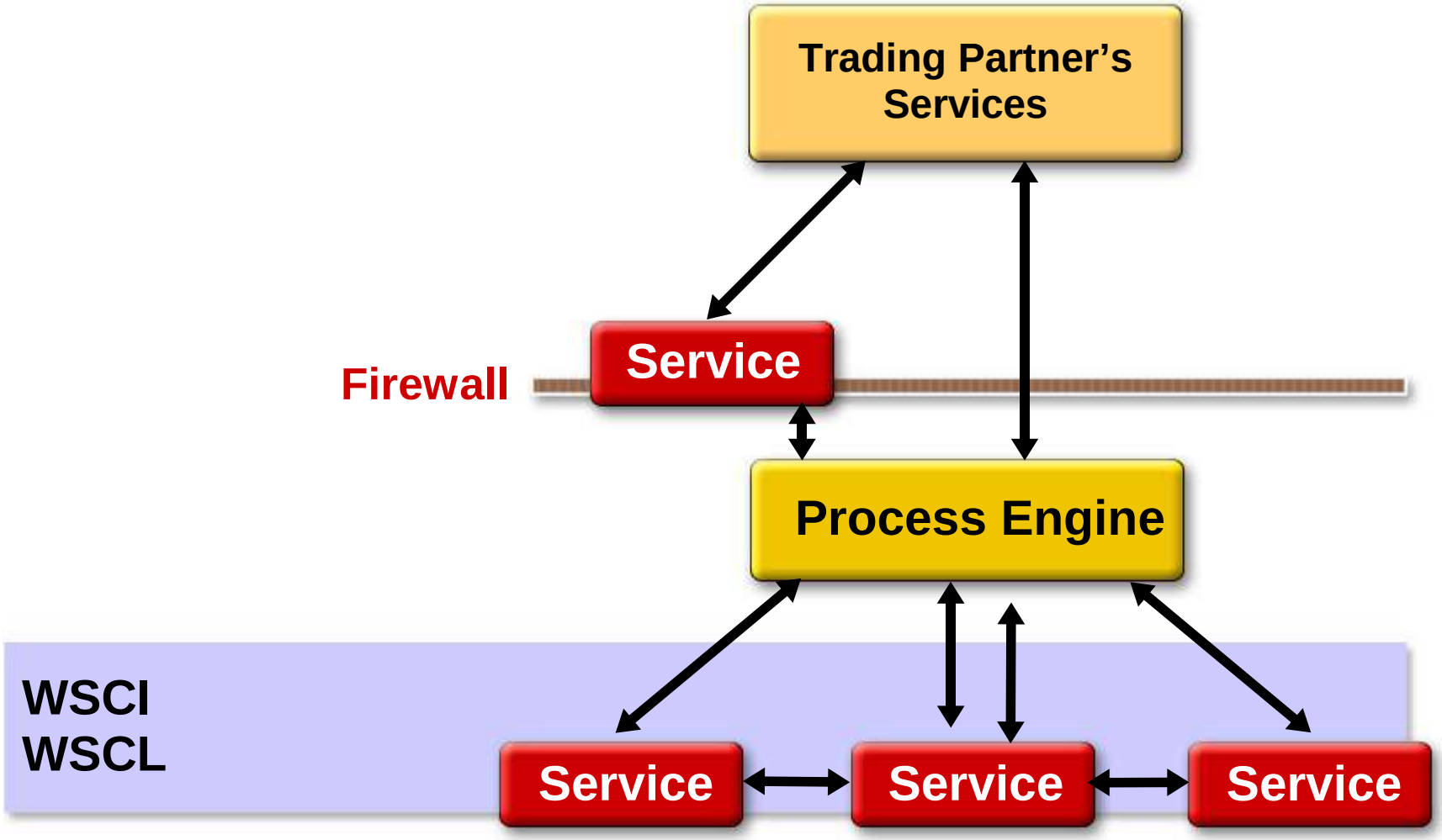
Business Transaction Activity: Get Quote



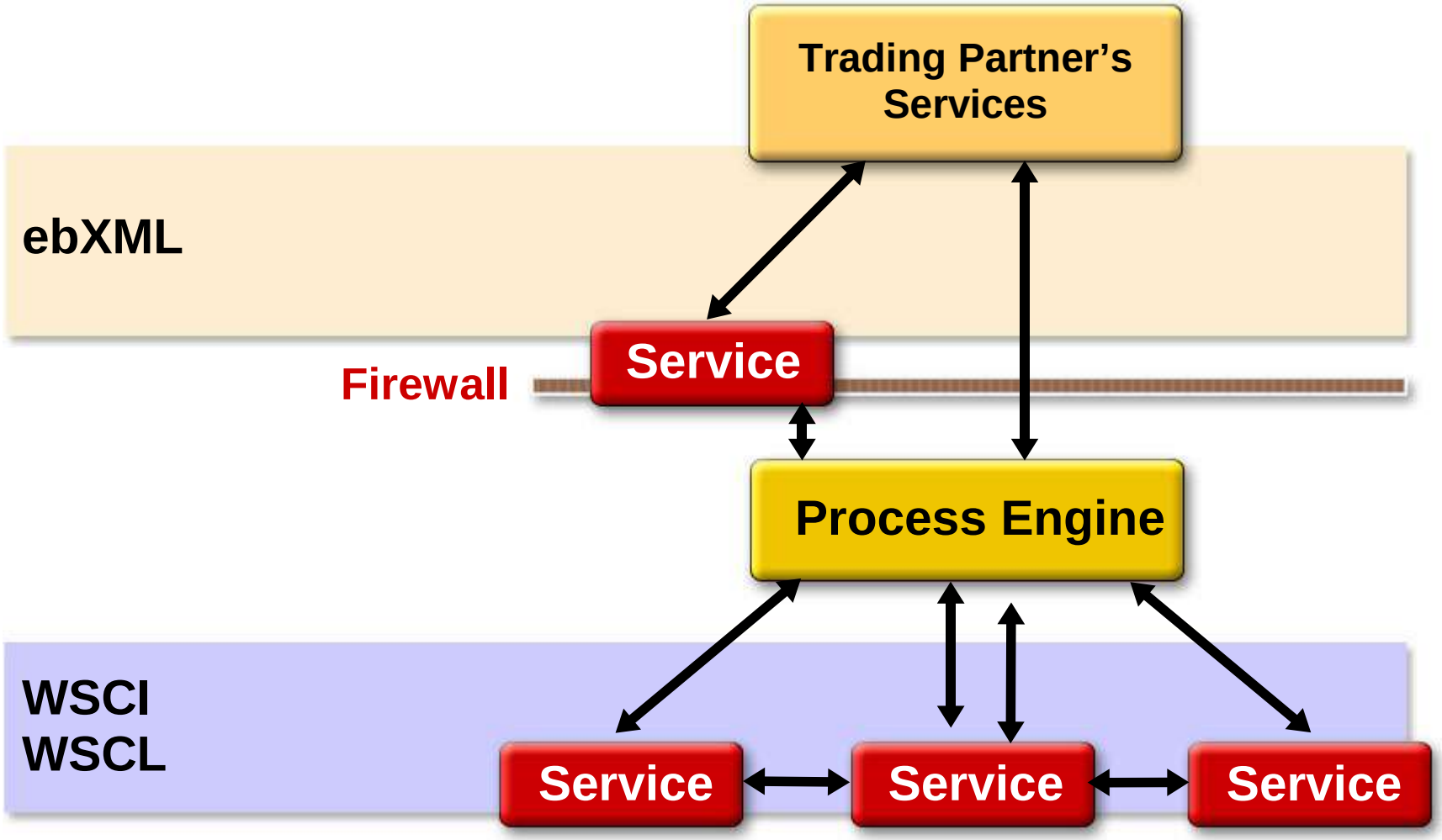
Summary


Trading Partner's Services
Firewall
Service
Process Engine
Service
Service
Service

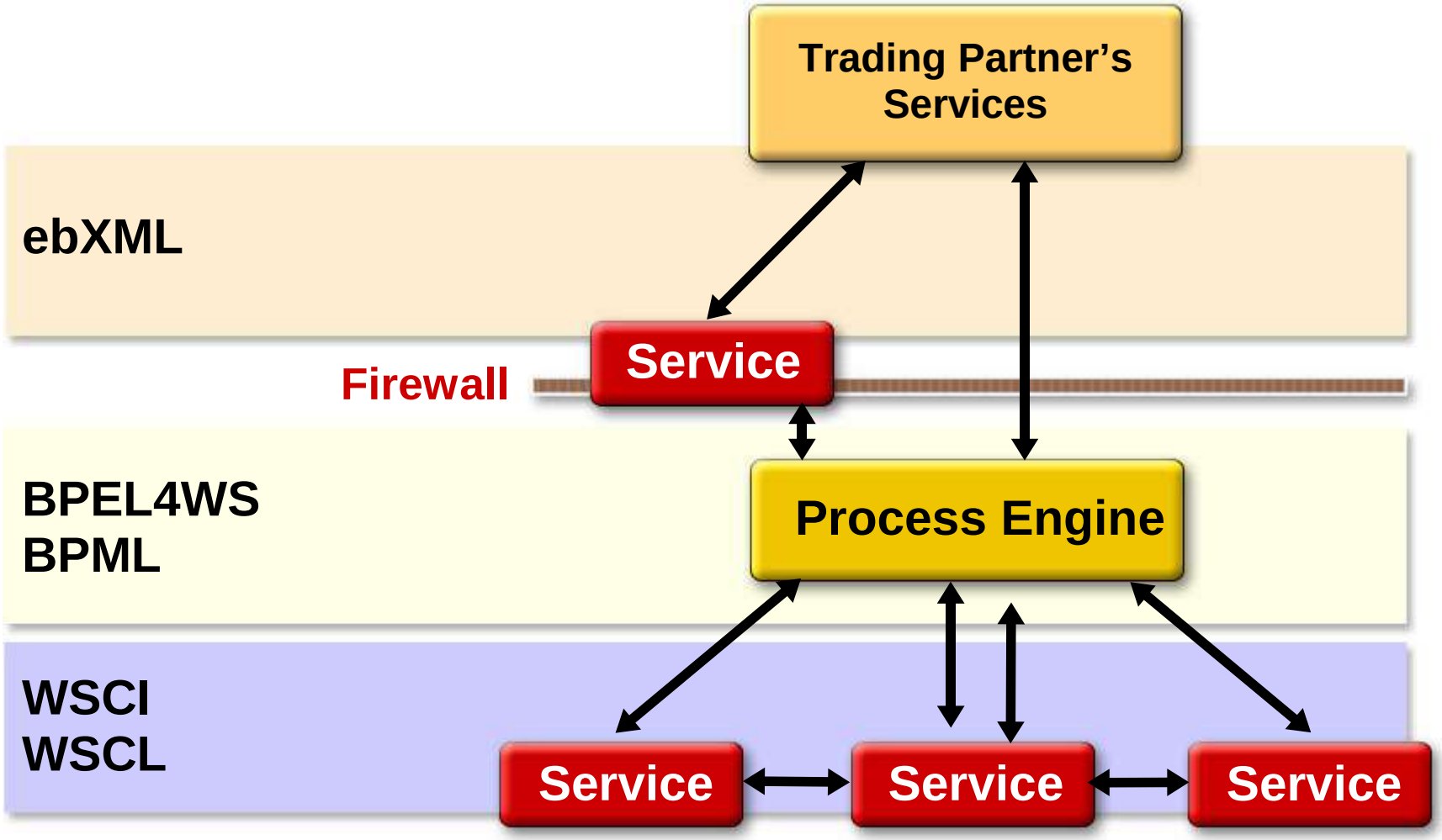
Summary


Trading Partner's Services
Firewall
Service
Process Engine
WSCI
WSCL
Service
Service
Service

Summary


Trading Partner's Services
ebXML
Service
Firewall
Process Engine
WSCI
WSCL
Service
Service
Service

Summary


Trading Partner's Services
ebXML
Service
Firewall
BPEL4WS
BPML
Process Engine
WSCI
WSCL
Service
Service
Service

Summary

- Distinct technical meanings
  - Collaboration
  - Orchestration
  - Choreography
- Abstract models
  - Help understand specifications
  - Help drive specifications

Summary

- Web Services
  - Service-oriented architecture
  - WSDL
    - Simple structure
    - Everybody is “fixing” it
  - Expanding Orchestration/Choreography features:
    - Correlation
    - Transactions/Exceptions/Compensation
    - Roles
    - Dynamic participation

Summary

- Business Web Services
  - Different from Web Services
  - EDI by Internet
  - ebXML provides
    - Reliable, secure messaging
    - Trading partner agreements
    - Shared business processes
    - Business semantics

If You Only Remember One Thing…

We don’t have a Web Services silver bullet.

Different solutions are needed in different parts of an enterprise’s EAI and B2B structures.

Q&A

JavaOne℠

Sun's 2003 Worldwide Java Developer Conference℠

java.sun.com/javaone/sf