

# Business Processes and Service Specifications

Paul C. Brown



# Why Focus on Business Processes?

- **Business processes are the mechanisms that provide enterprise value**
  - Consistent execution is vital
  - IT mission is to facilitate business process execution
- **A service provides value when it becomes part of a business process**
  - No use = no value
- **The value of a service depends on its ability to fit one or more business processes**
  - Lack of fit = wasted investment

***Services must be designed to fit business processes!***

# What Must Be Considered to Specify a Service?

- ❑ **Basic functional requirements**

- What does the business process require?

- ❑ **Non-functional requirements**

- May differ from process to process

- ❑ **Business process delivery channels**

- Non-functional requirements may vary

- ❑ **Coordination**

- How is the service work coordinated with the business process?

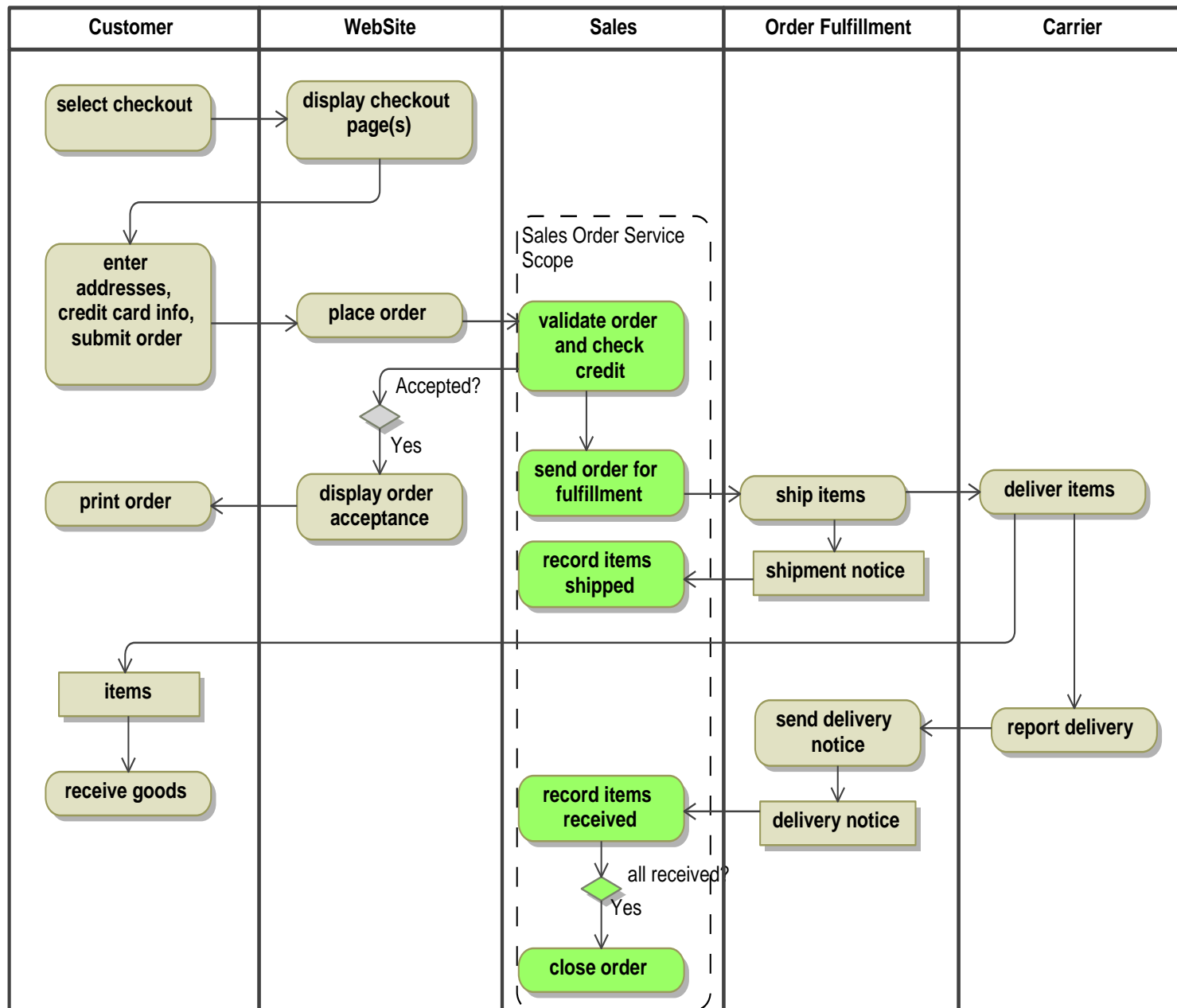
- ❑ **Service architecture**

- Not entirely a black-box exercise

# Basic Functional Requirements

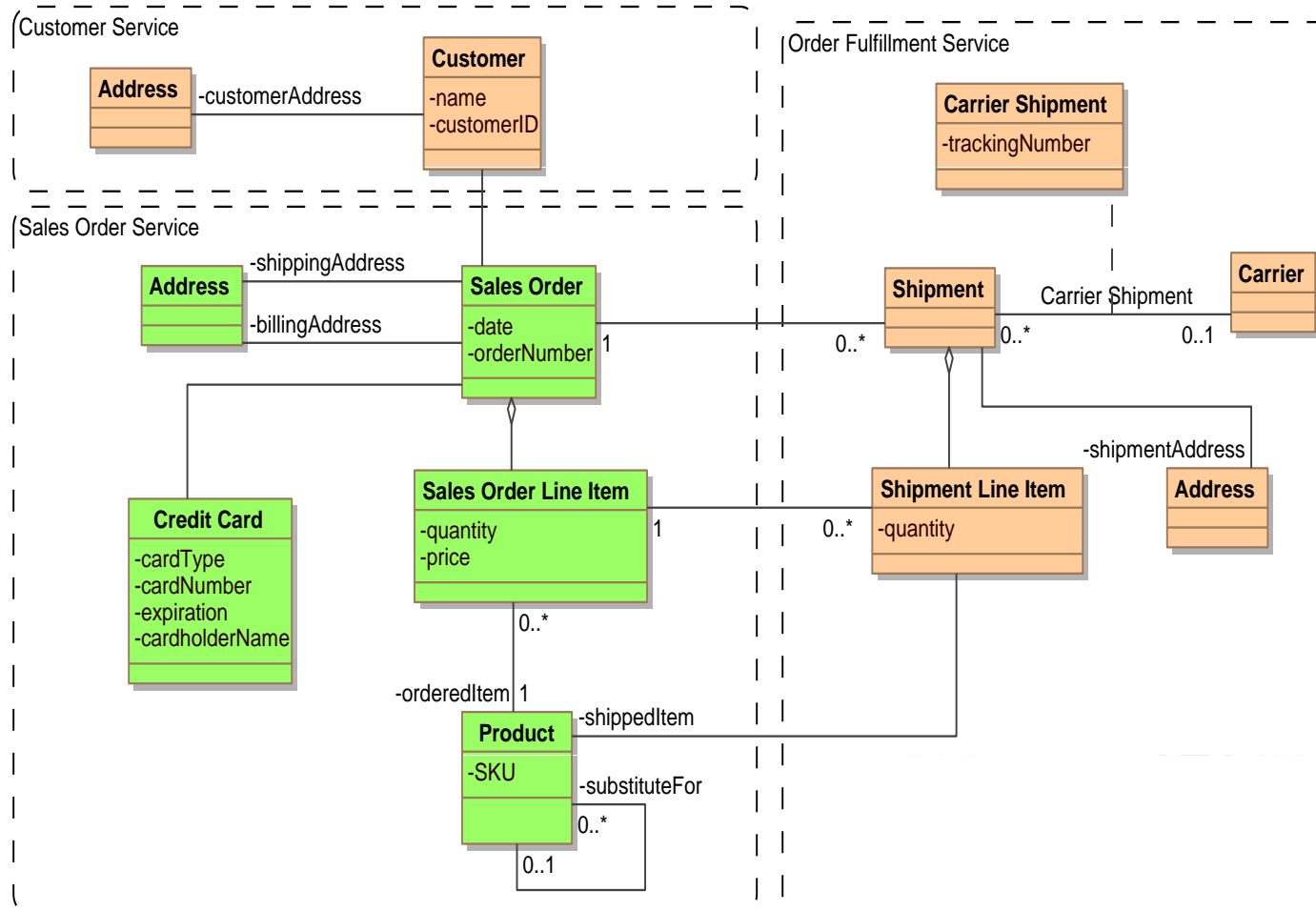


# Business Process Context



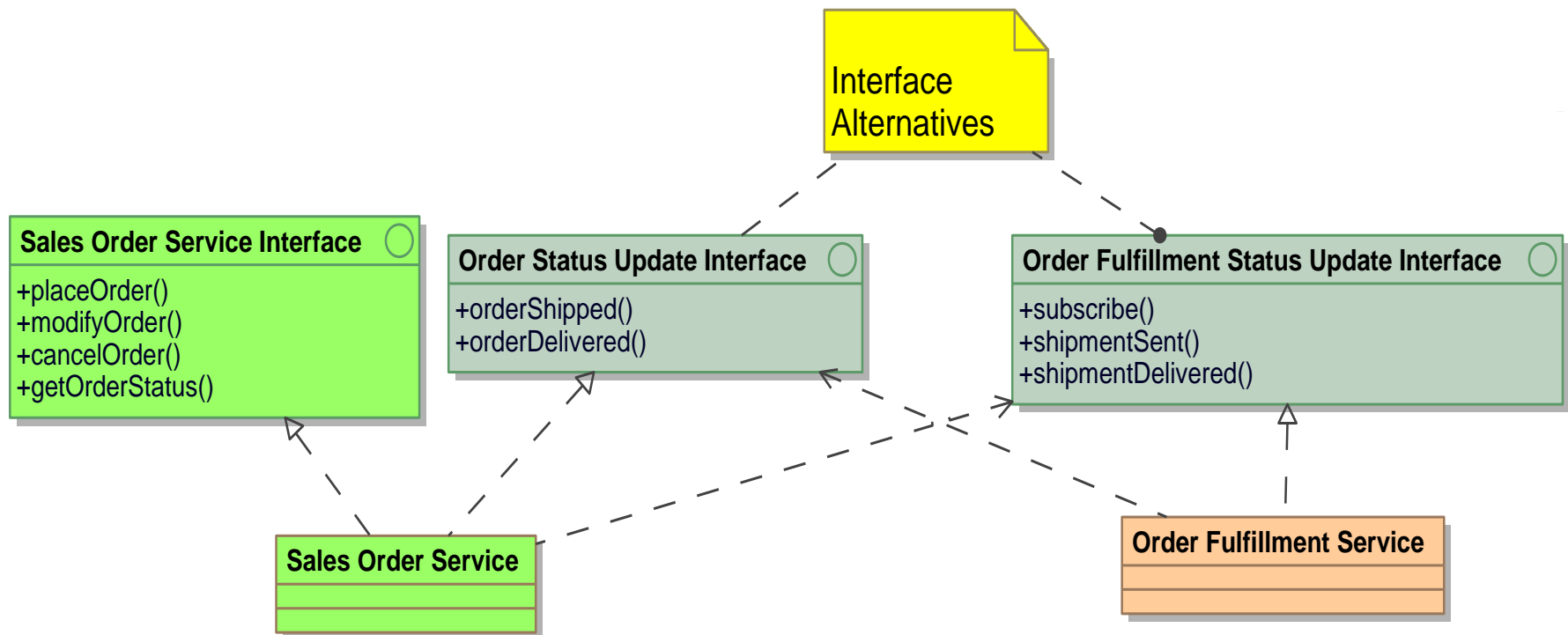
# Information Requirements

- ❑ What information does the service manage?
- ❑ What information does it use but not own?
  - Is the information cached? How is the cache maintained?

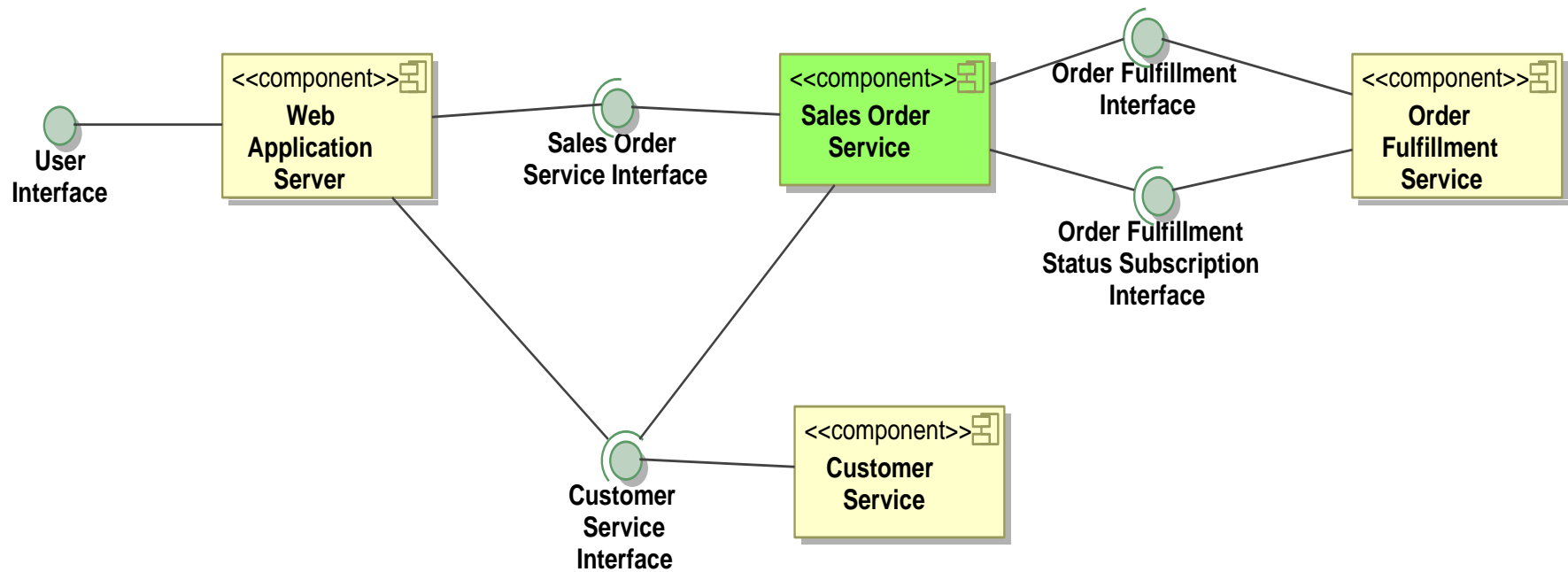


# Capability Requirements

- Provided operations
- Don't forget about multi-step processes

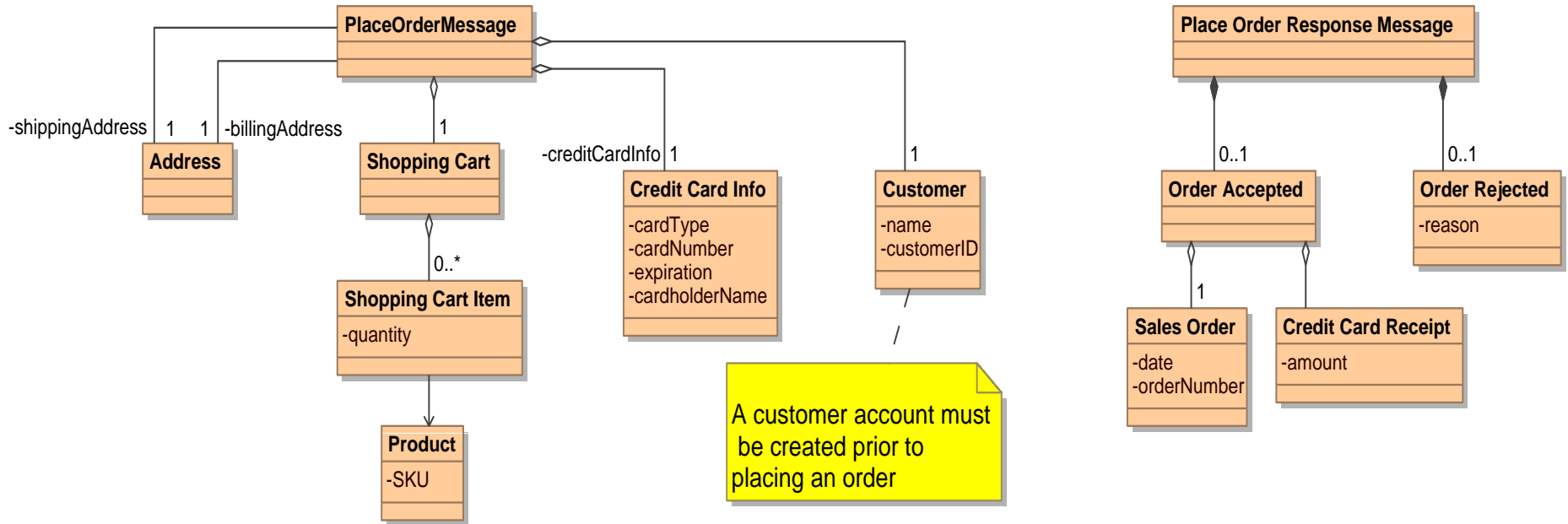


# Partial Solution Architecture



# Message Data Structures

- Information required
- Information returned
- Common data models
  - Whole messages?
  - Common sub-structures



# Granularity

- ❑ **One or many items per order?**
- ❑ **One or many orders at a time?**
  - Multiple shopping carts
  - Common shipping address?
  - Common billing information?
  - Common customer?
- ❑ **One or many shipments per order?**
  - Affects interfaces between Customer Order Service and Order Fulfillment Service

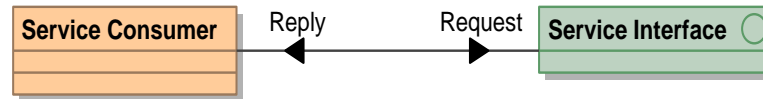
# Coordination



# Message Exchange Patterns

## □ In-Out

- Request reply



## □ In-Only

- Fire-and-forget



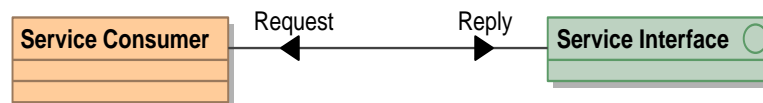
## □ Out-Only

- Subscription



## □ Out-In

- Warranty recall

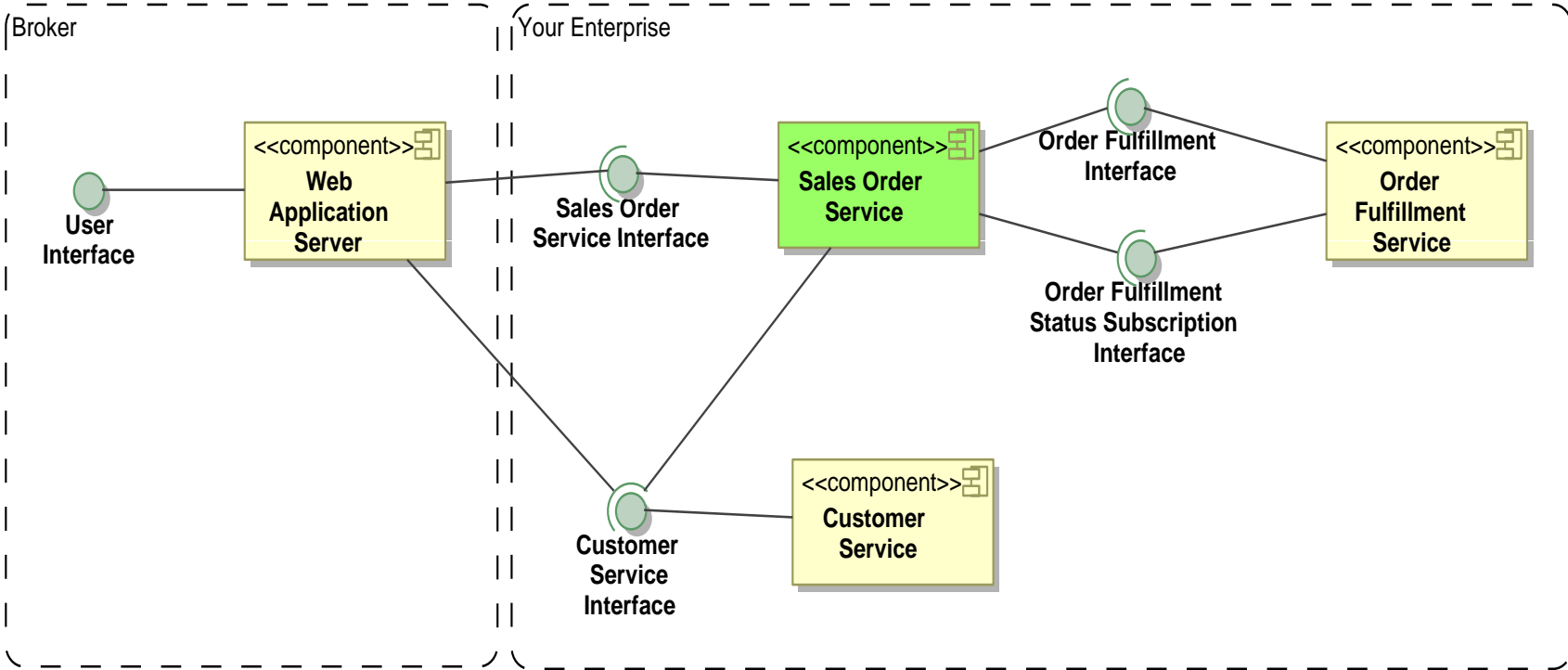


□ In-\* and Out-\* patterns differ in which party specifies the interface

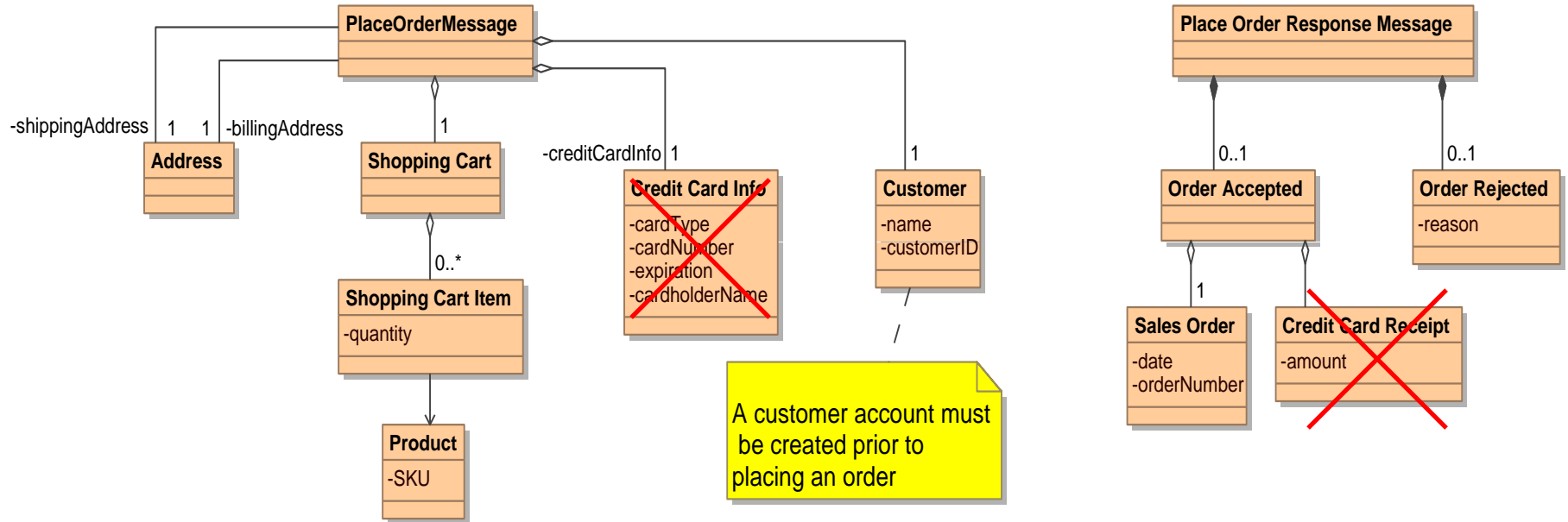
# Different Channels Alter the Business Process

- **On-line direct ordering**
  - Immediate synchronous response
- **Email ordering**
  - Delayed asynchronous response
- **Phone ordering**
  - Operator requires immediate synchronous response
  - Secondary feedback channel may be required (e.g. email)
- **B2B ordering**
  - Batch – asynchronous response
  - Interactive – synchronous response
- **Agent ordering**
  - Passes customer payments through
  - Separate accounting for agent fees
- **Broker (reseller) ordering**
  - Broker handles customer payments
  - Broker pays for products at a discount

# Partial Architecture with Broker



# Place Order Message Structures with Broker



□ An alternate mechanism for payments is required!

# Request-Reply Considerations

## □ Synchronous interaction

- In-Out pattern
- Allowed completion time must be specified

## □ Asynchronous interaction

- In-Out with WS-Addressing for response
- In-Only and Out-Only with WS-Addressing for response

## □ Can the same operation support both?

- Use presence of WS-Addressing reply-to destination to indicate asynchronous response is required? (untried!)

## □ How to handle large data sets in reply

- WS-Enumeration?

***Different business processes and channels may require different coordination patterns!***

# Subscription Services

## □ Managing the subscription

- WS-Eventing and associated interfaces

## □ Interaction patterns for delivering the subscription

- Out-Only (typical)
- Out-In
  - Price list updates to agents and brokers
  - Product recalls to customers

# Exception Handling

## □ Don't use SOAP faults for expected conditions

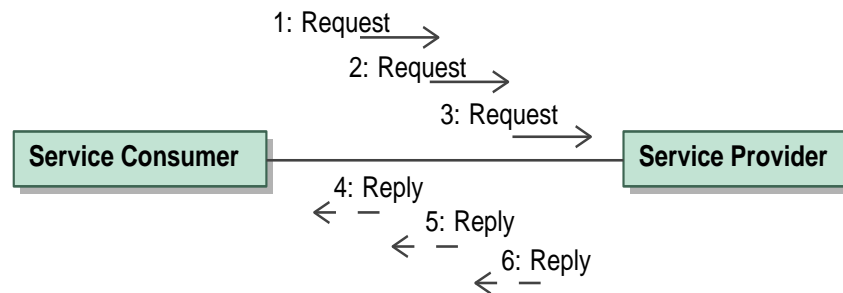
- May be difficult to continue business process flow with SOAP faults
- Expected conditions should return expected results
  - e.g. Bad credit card number on placeOrder() operation

## □ Anticipate business process requirements for responding to exceptions

- Information required in expected results
  - e.g. primary keys, problem details
- Additional service operations
  - e.g. query, edit, delete, hold, and resume operations
- Options for resuming business process
  - e.g. engage customer service representative
  - Support must be provided for these activities

# High Throughput Considerations

- ❑ Focus on true peak rate at which response time must be met
- ❑ Consider back-end limitations!
- ❑ High throughput does not necessarily require short response time
  - Asynchronous responses allow multiple requests to be in progress



- ❑ Do requests need to be serialized?
  - E.g. create, delete, and update operations on the same order must be processed in the order received

# Other Non-Functional Requirements to Consider

## □ Response time

- Synchronous and asynchronous

## □ Availability

- Required availability during normal working hours
  - Maximum time to recover
- Allowed outage periods

## □ Security

- Authentication and authorization
  - Variations for different business processes
- In-transit data encryption
- At-rest data encryption

## □ Most requirements derive from business processes

# Service Architecture Alternatives

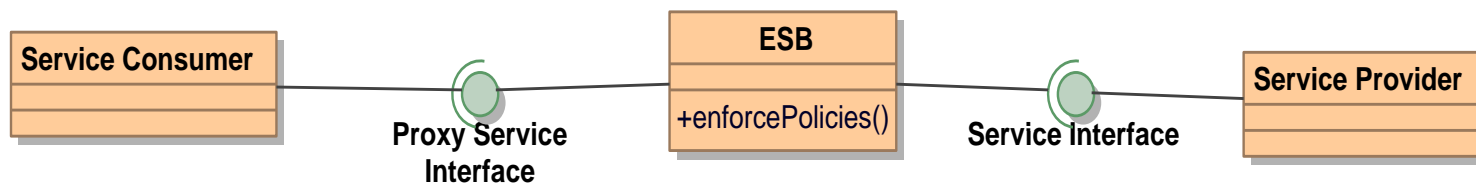
## □ Back-end system wrapper

- How will scaling work?



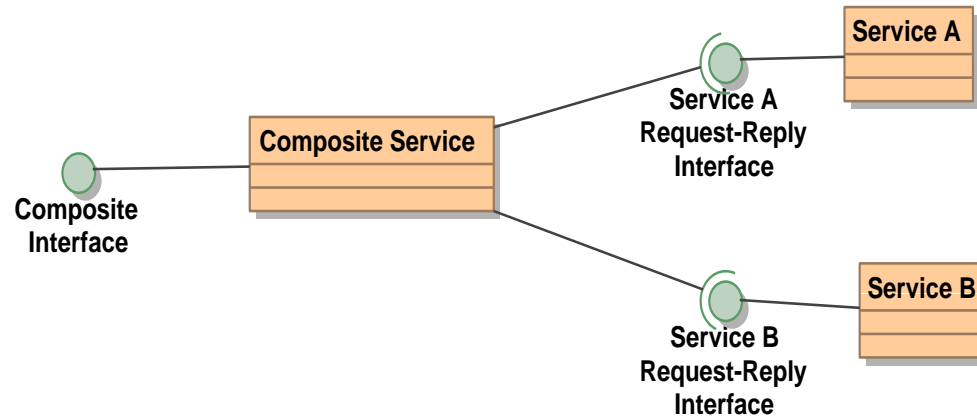
## □ Policy-based access

- Security
- Results filtering

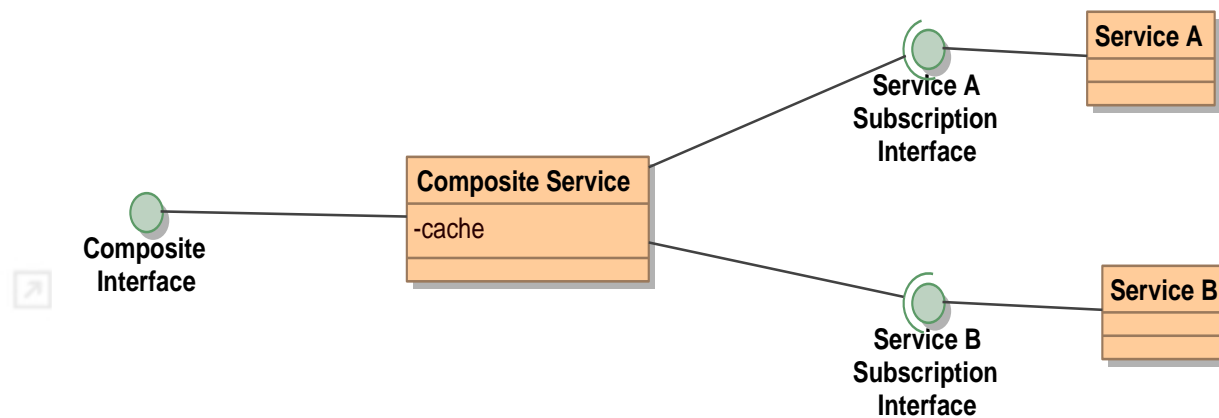


# Composite Services

## □ Traditional architecture



## □ Cache-based architecture



# Summary

- ❑ **Business process needs drive most service requirements**
- ❑ **Reuse depends on satisfying the requirements of more than one business process**
  - Granularity and interaction patterns may be different
- ❑ **Most services manage information**
  - References to another service's information require additional interfaces to maintain consistency
- ❑ **Channels have a huge impact on requirements**
  - Changing enterprise boundaries always changes interfaces!
- ❑ **The internal architecture of the service is important**

# Questions?

