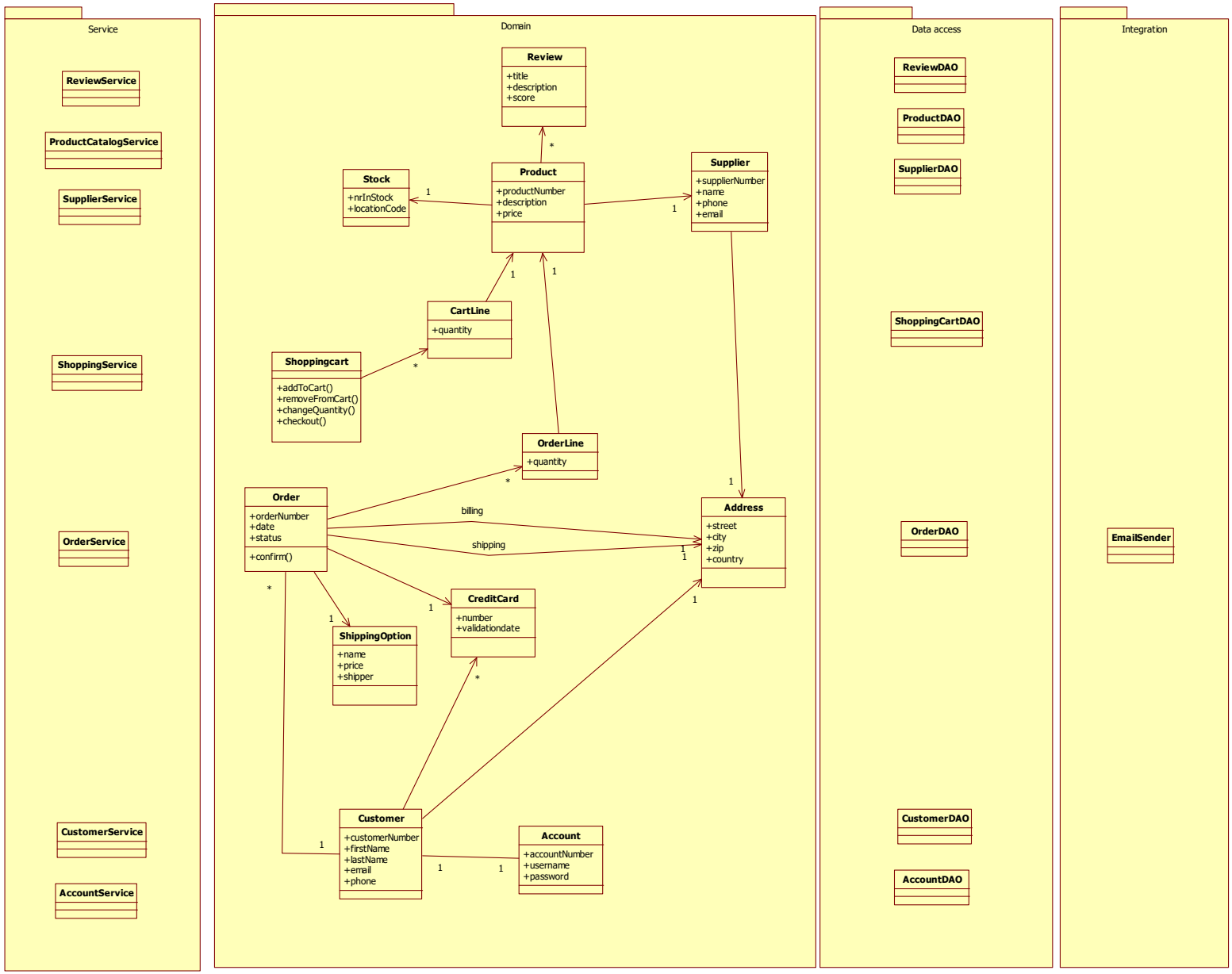
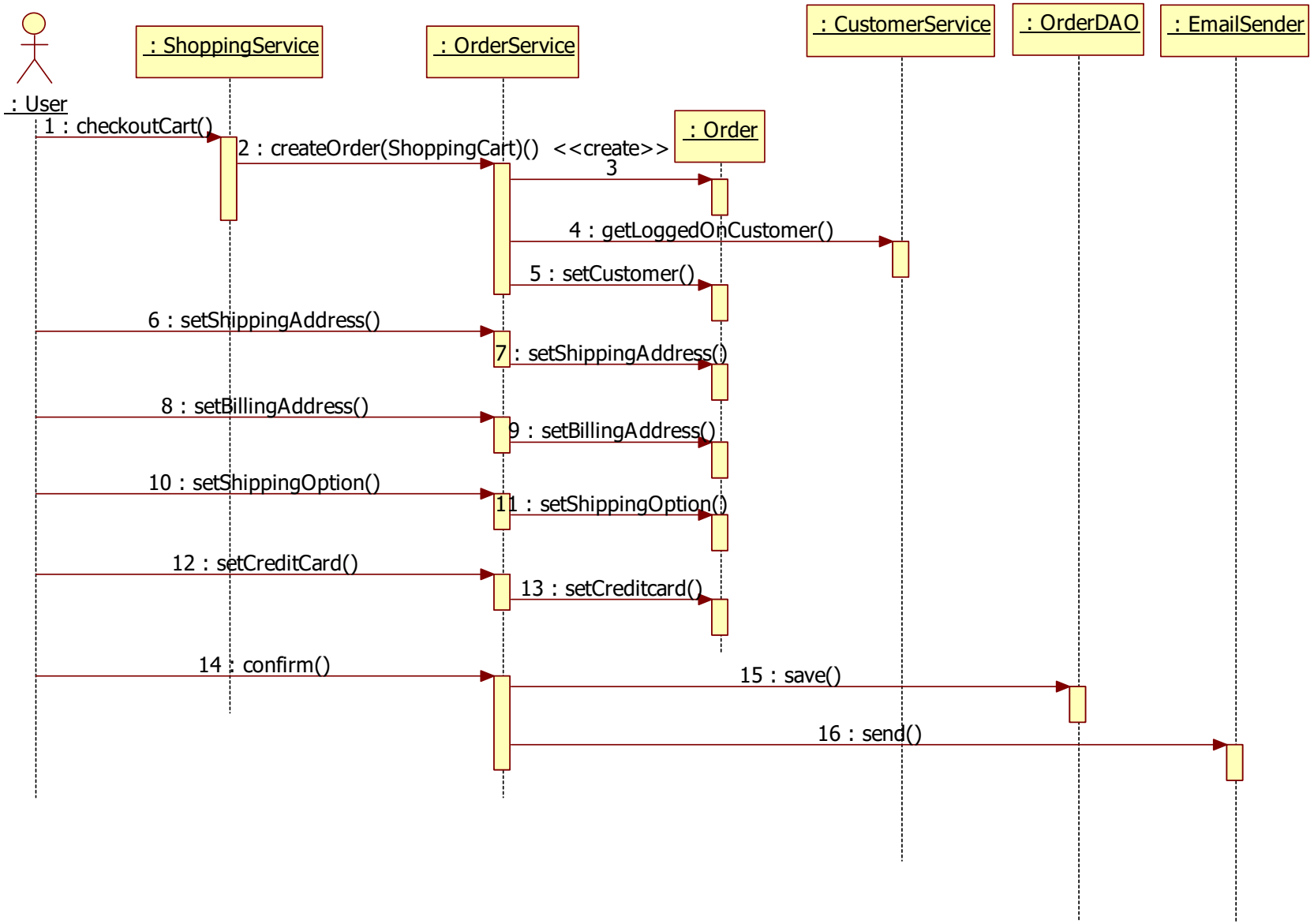
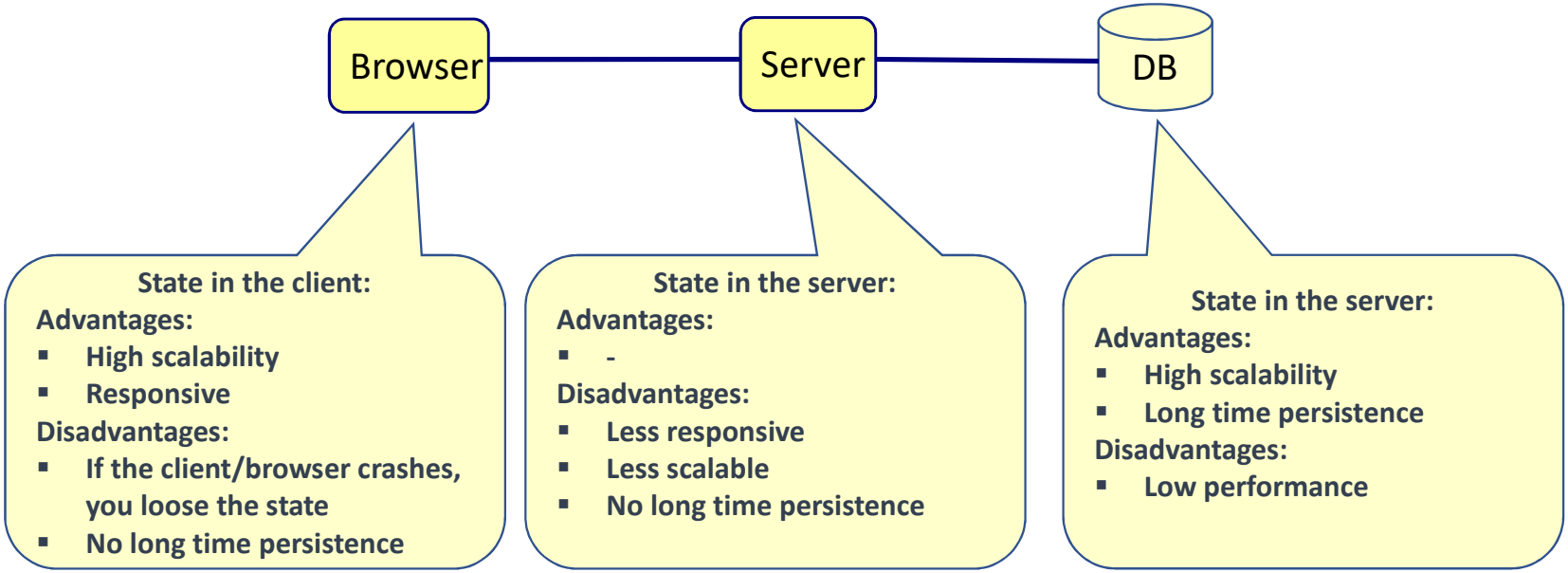

Service
ReviewService
ProductCatalogService
SupplierService
ShoppingService
OrderService
CustomerService
AccountService
Domain
Review
+title
+description
+score
Stock
+nrInStock
+locationCode
Product
+productNumber
+description
+price
Supplier
+supplierNumber
+name
+phone
+email
CartLine
+quantity
Shoppingcart
+addToCart()
+removeFromCart()
+changeQuantity()
+checkout()
OrderLine
+quantity
Order
+orderNumber
+date
+status
+confirm()
billing
shipping
Address
+street
+city
+zip
+country
CreditCard
+number
+validationdate
ShippingOption
+name
+price
+shipper
Customer
+customerNumber
+firstName
+lastName
+email
+phone
Account
+accountNumber
+username
+password
Data access
ReviewDAO
ProductDAO
SupplierDAO
ShoppingCartDAO
OrderDAO
CustomerDAO
AccountDAO
Integration
EmailSender


: User
: ShoppingService
: OrderService
: Order
: CustomerService
: OrderDAO
: EmailSender
1 : checkoutCart()
2 : createOrder(ShoppingCart)()
<<create>>
3
4 : getLoggedOnCustomer()
5 : setCustomer()
6 : setShippingAddress()
7 : setShippingAddress()
8 : setBillingAddress()
9 : setBillingAddress()
10 : setShippingOption()
11 : setShippingOption()
12 : setCreditCard()
13 : setCreditcard()
14 : confirm()
15 : save()
16 : send()

Where to store state?



Integration techniques

technique	advantage	disadvantage	When to use?
RMI		Only Java to Java High coupling	Never
Messaging	Buffer Low coupling Asynchronous	You need messaging middleware	Between applications within the same organization When you need asynchronous requests
SOAP	Standards for security, transactions, etc. Standard for interface description	Complex	When you need standards
REST	Simple	No standards for security, transactions, etc No standard for interface description	Everywhere you need synchronous requests
Serialized objects over HTTP	Simpler as RMI Webcontainer functionality	Only Java to Java High coupling	For fast Java to Java integration between applications managed by the same project
Database integration	Simple	High coupling	Never
File based integration		High coupling	If you have to communicate large files

Distributed systems

- Advantages
  - More resources available (CPU's, memory, ports, etc)
- Disadvantages
  - Very complex
  - Lots of thinks that can go wrong
  - Remote calls are extremely slow
  - Security is more complex
  - Transactions are more complex