Package Diagrams

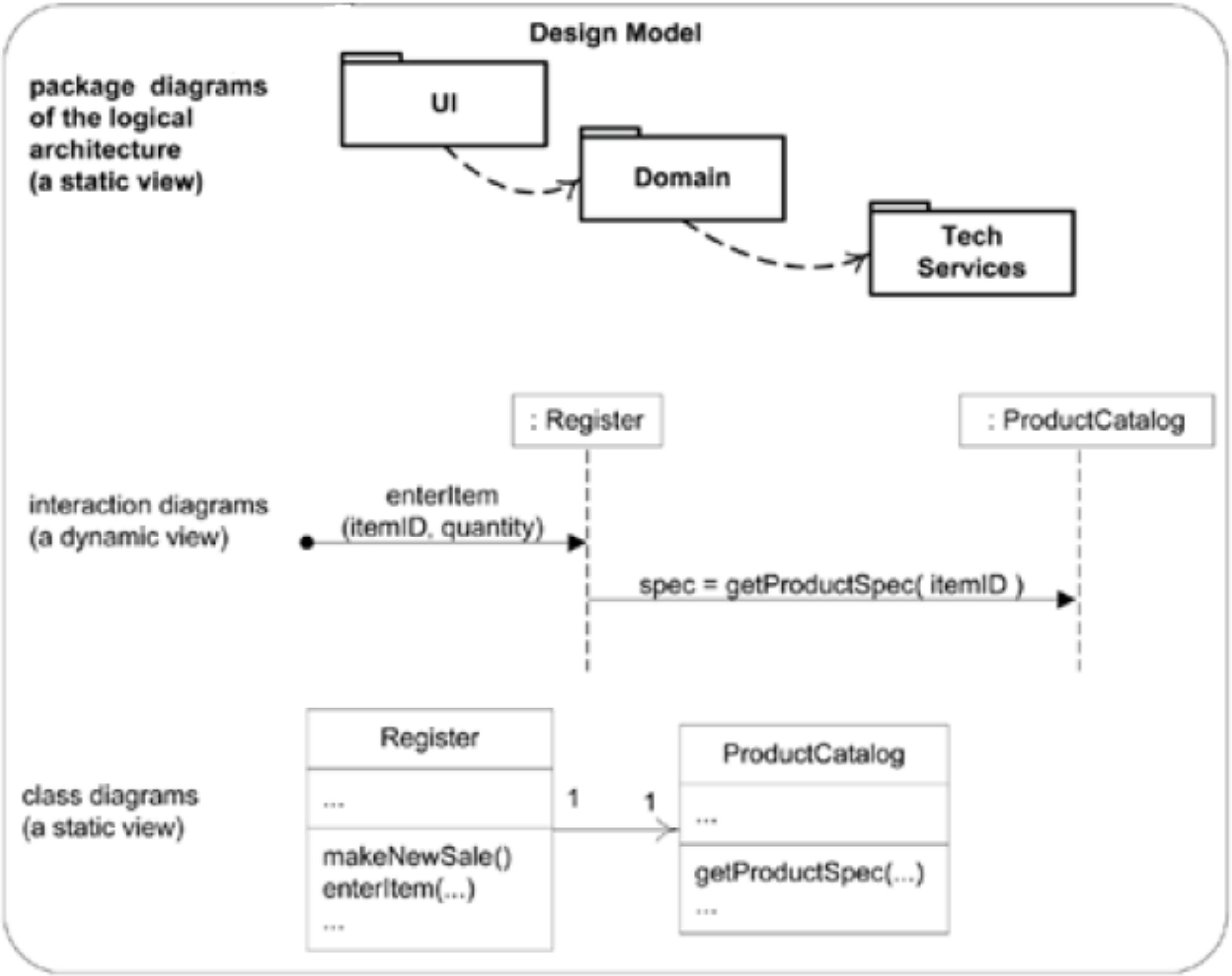
Software Design and Analysis

CSCI 2040

Objectives

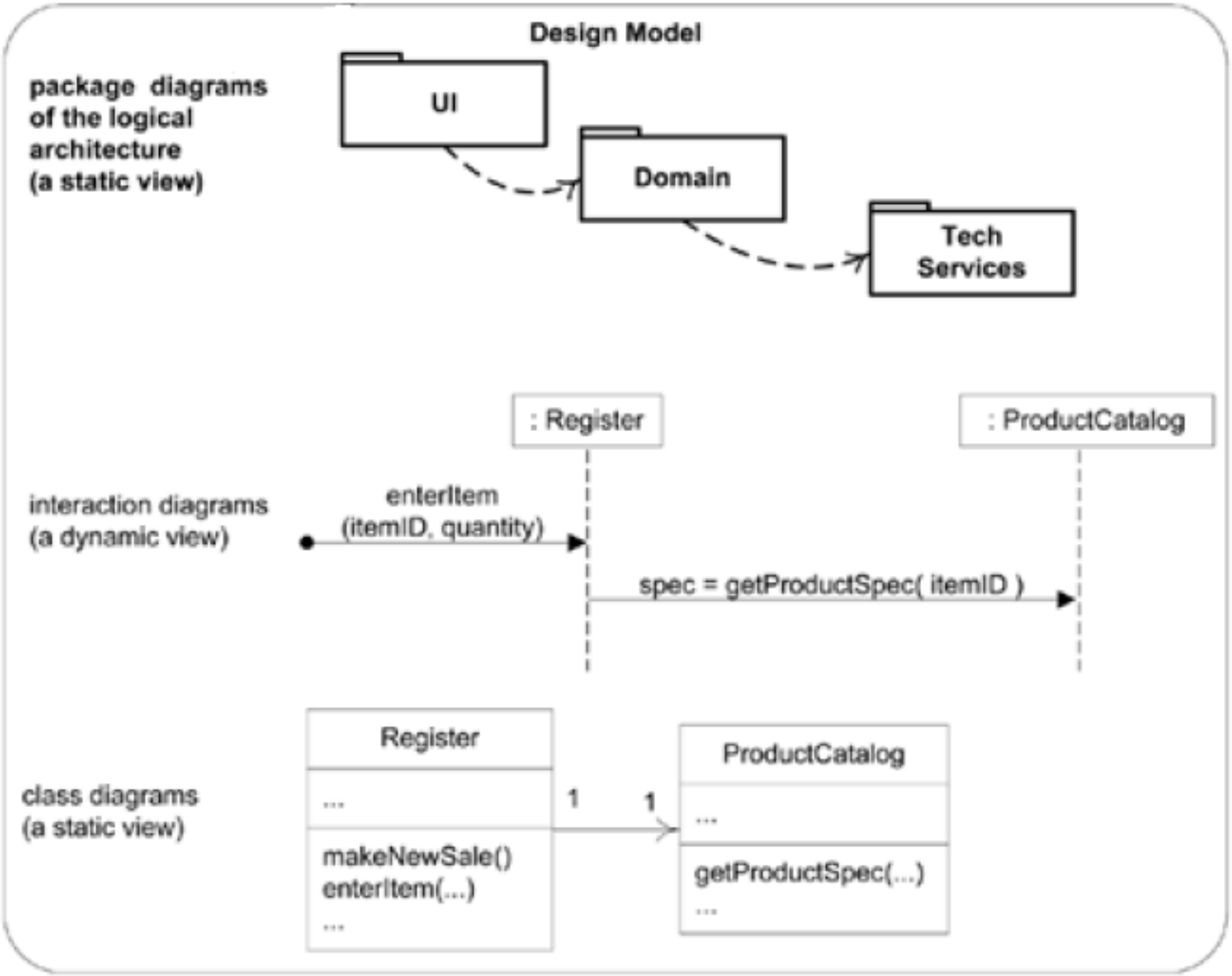
- Introduce a **logical architecture** using layers
- Illustrate the logical architecture with UML **Package Diagrams**

Design Model


Design Model
package diagrams of the logical architecture (a static view)
UI
Domain
Tech Services
interaction diagrams (a dynamic view)
: Register
: ProductCatalog
enterItem (itemID, quantity)
spec = getProductSpec(itemID)
class diagrams (a static view)
Register
...
makeNewSale()
enterItem(...)
...
1
1
ProductCatalog
...
getProductSpec(...)
...

- Where the class Register should be?

Design Model


Design Model
package diagrams of the logical architecture (a static view)
UI
Domain
Tech Services
interaction diagrams (a dynamic view)
: Register
: ProductCatalog
enterItem (itemID, quantity)
spec = getProductSpec(itemID)
class diagrams (a static view)
Register
...
makeNewSale()
enterItem(...)
...
1
1
ProductCatalog
...
getProductSpec(...)
...

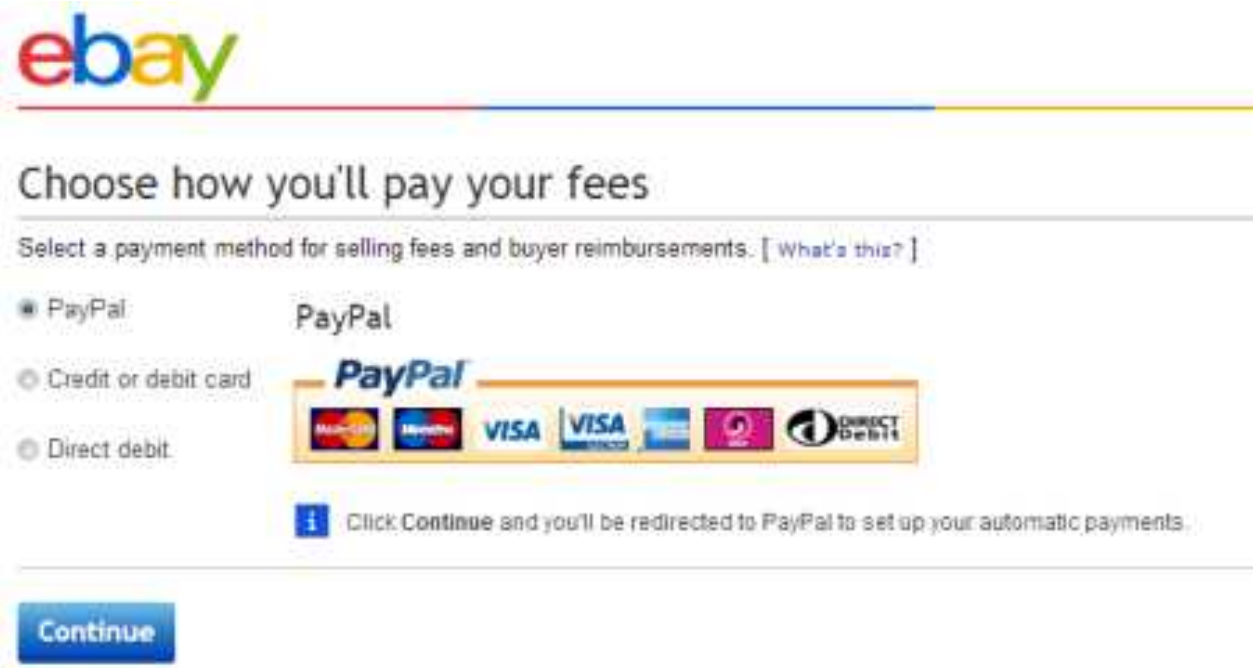
- Where class Register should be? Domain Package

Typical Layers

- **User Interface** - the means by which the user and a computer system interact
- **Domain Objects** – application and business logic
  - software objects representing domain concepts that fulfill application requirements, such as calculating a sale total (e.g., a software class *Sale*).
- **Technical Services**
  - general purpose objects and subsystems that provide supporting technical services, such as
    - interfacing with a database or
    - error logging.
  - These services are usually application-independent and reusable across several systems.

Paypal

- Which layer is Paypal?



Choose how you'll pay your fees

Select a payment method for selling fees and buyer reimbursements. [What's this?]

● PayPal

○ Credit or debit card

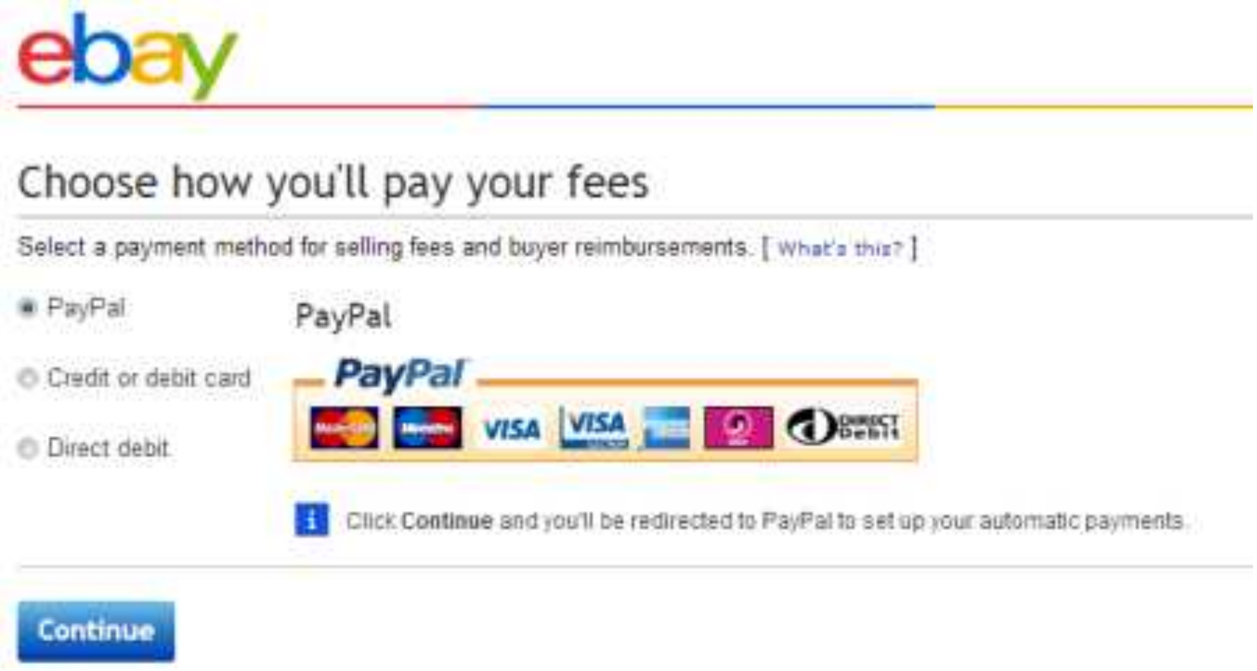
○ Direct debit

PayPal

Click **Continue** and you'll be redirected to PayPal to set up your automatic payments.

Paypal

- Which layer is Paypal?
  - Technical Services



Choose how you'll pay your fees

Select a payment method for selling fees and buyer reimbursements. [What's this?]

● PayPal

○ Credit or debit card

○ Direct debit

PayPal

Click **Continue** and you'll be redirected to PayPal to set up your automatic payments.

Oracle

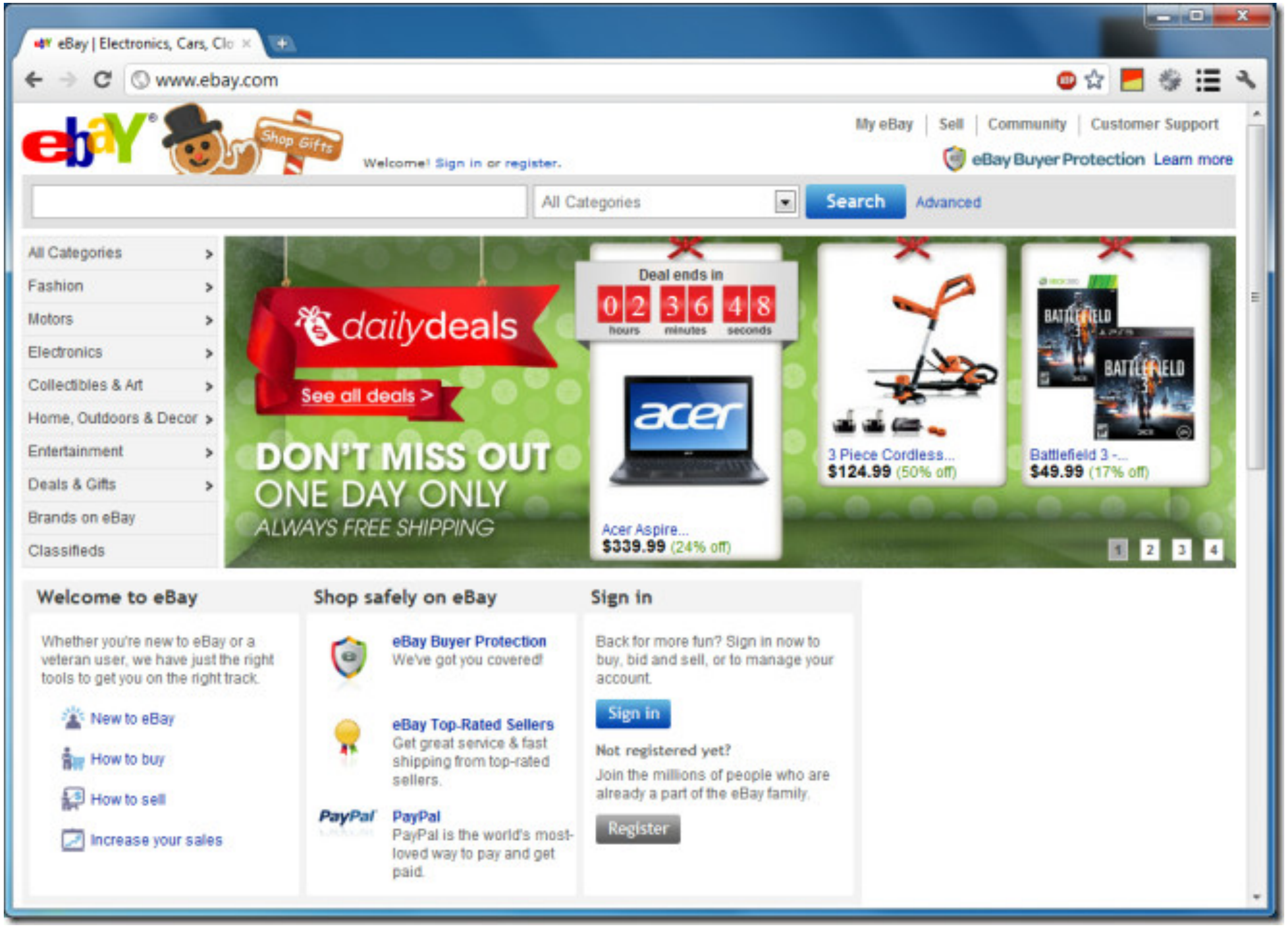
- Which layer is the Oracle database?

Oracle

- Which layer is the Oracle database?
  - Technical Services

eBay Web Interface

- Which layer is eBay web interface?



Welcome to eBay

Whether you're new to eBay or a veteran user, we have just the right tools to get you on the right track.

Shop safely on eBay

eBay Buyer Protection
We've got you covered!

eBay Top-Rated Sellers
Get great service & fast shipping from top-rated sellers.

PayPal
PayPal is the world's most-loved way to pay and get paid.

Sign in

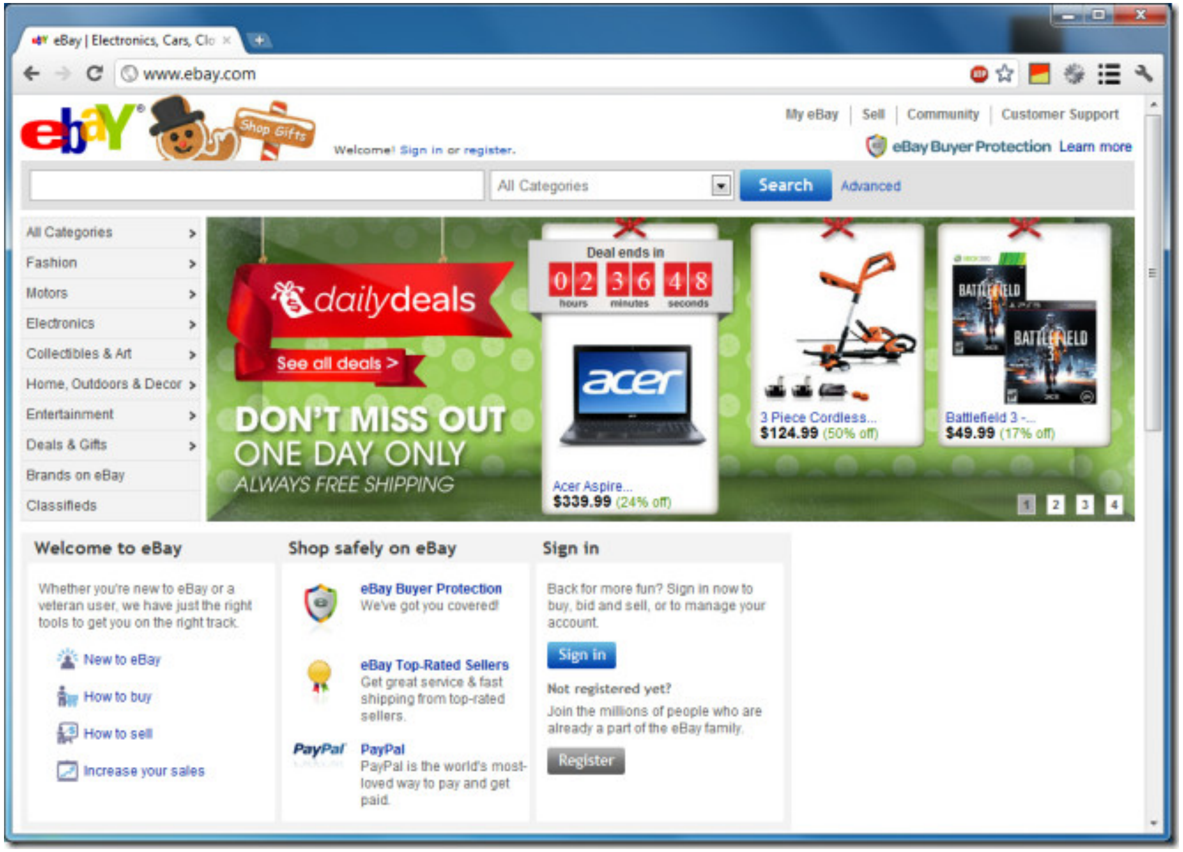
Back for more fun? Sign in now to buy, bid and sell, or to manage your account.

Not registered yet?

Join the millions of people who are already a part of the eBay family.

eBay Web Interface

- Which layer is eBay web interface?
  - User Interface



Welcome! Sign in or register.

Whether you're new to eBay or a veteran user, we have just the right tools to get you on the right track.

eBay Buyer Protection
We've got you covered!

eBay Top-Rated Sellers
Get great service & fast shipping from top-rated sellers.

PayPal
PayPal is the world's most-loved way to pay and get paid.

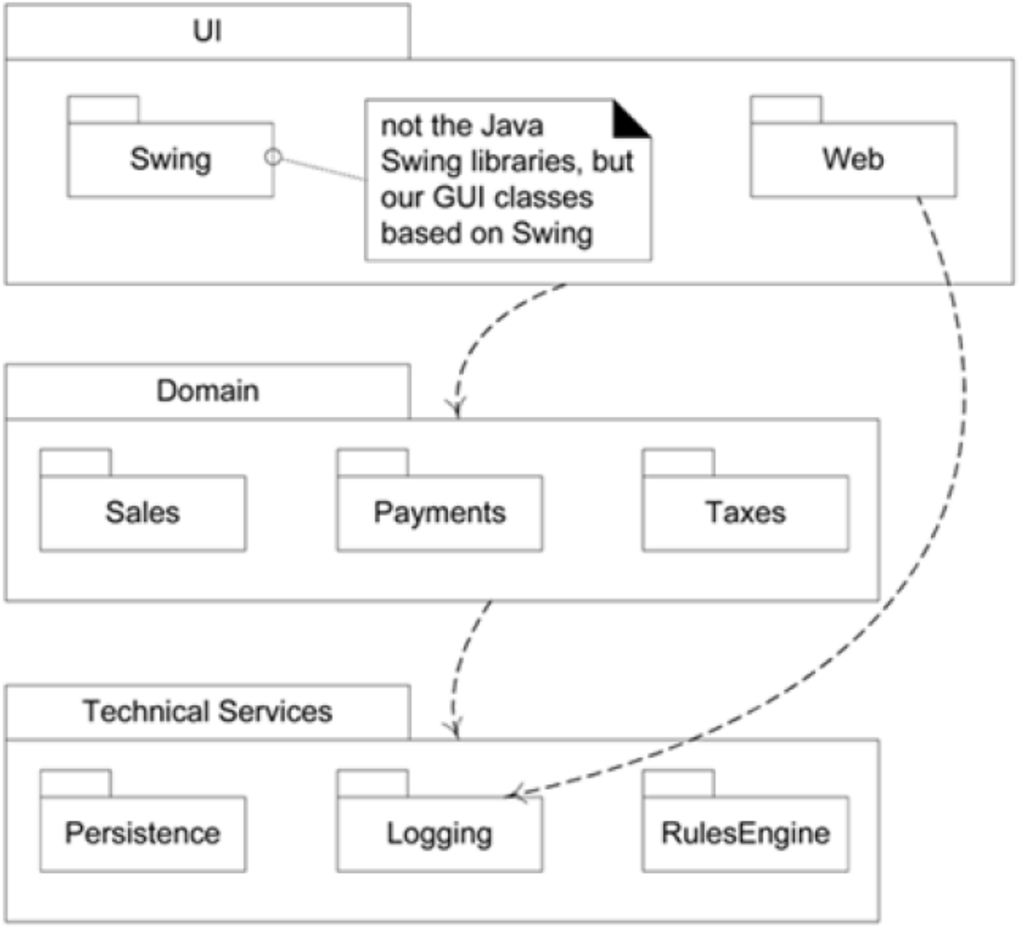
Back for more fun? Sign in now to buy, bid and sell, or to manage your account.

Not registered yet?

Join the millions of people who are already a part of the eBay family.

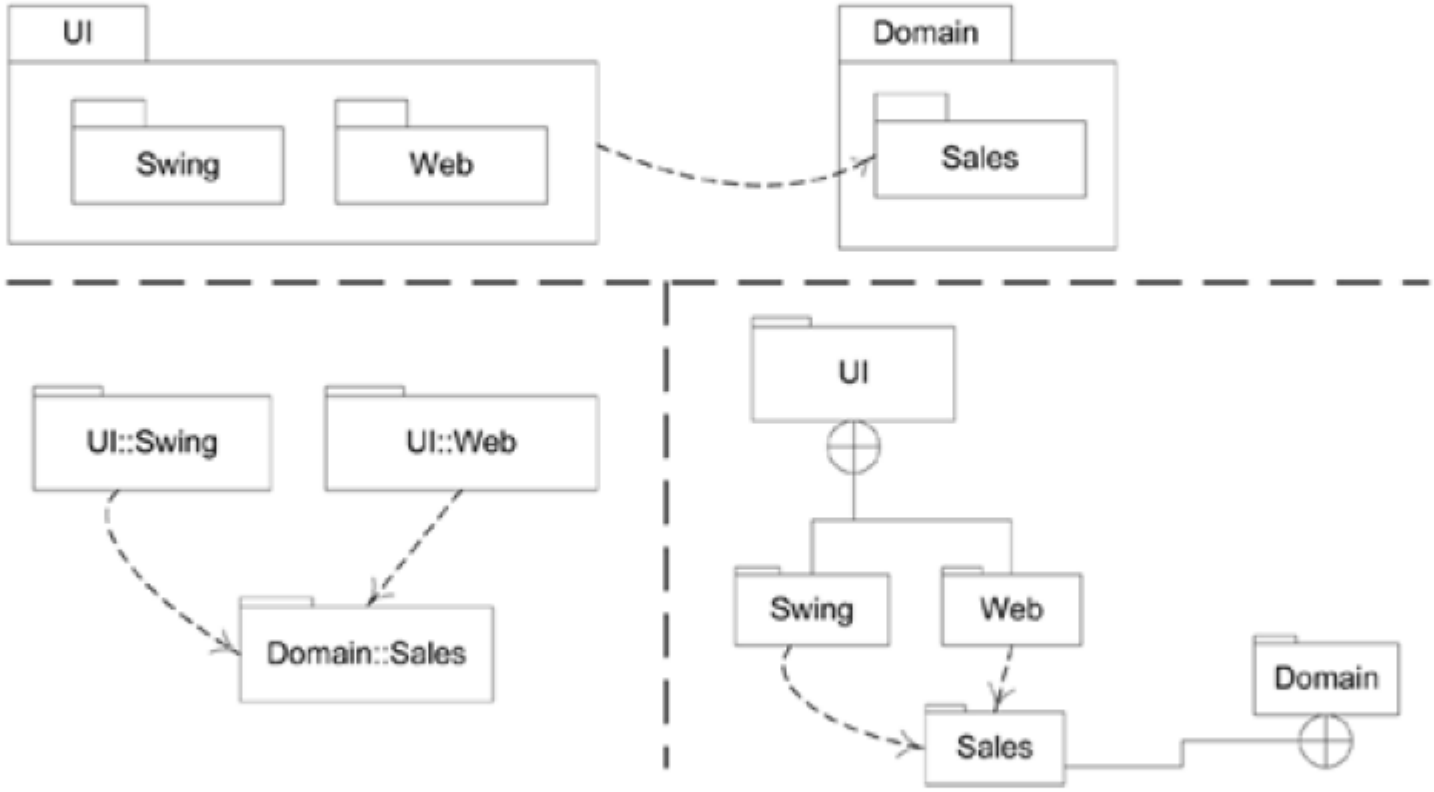
Layers in UML Package Diagram

- A logical architecture does not have to be organized in layers, but it is *very* common:
  - Especially for large and complex systems (e.g., banks)



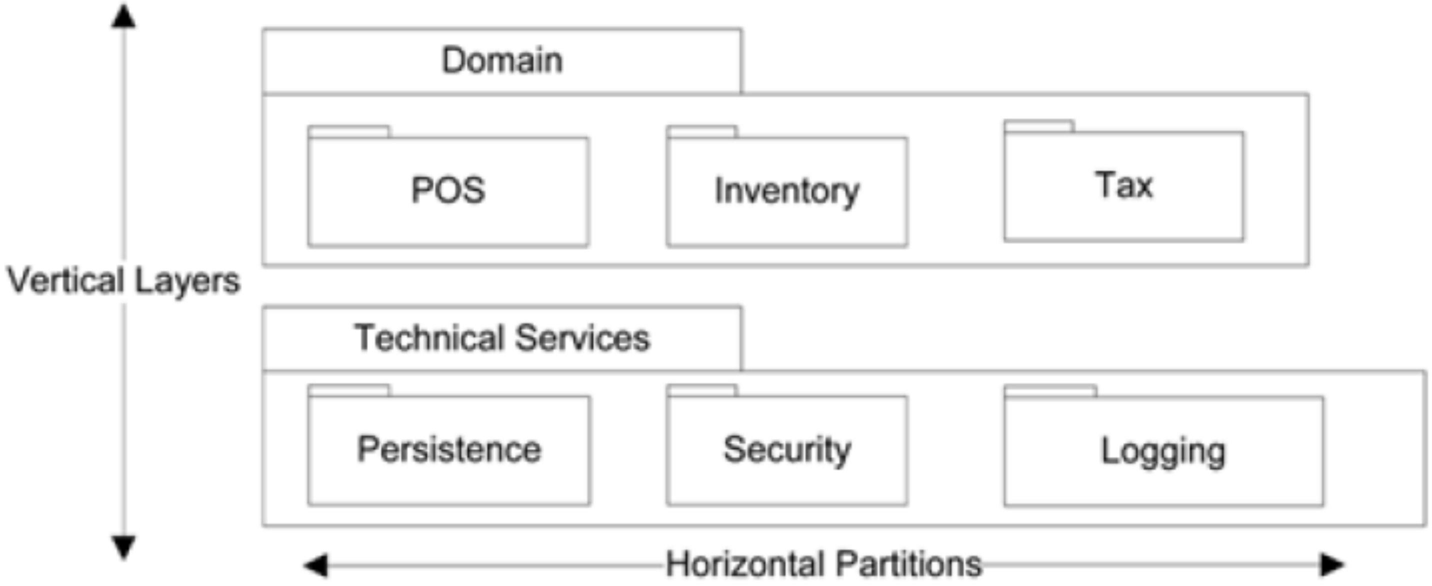
Alternate UML Approaches

- A UML package represents a **namespace**.
  - Dependency is represented with **dependency line**.

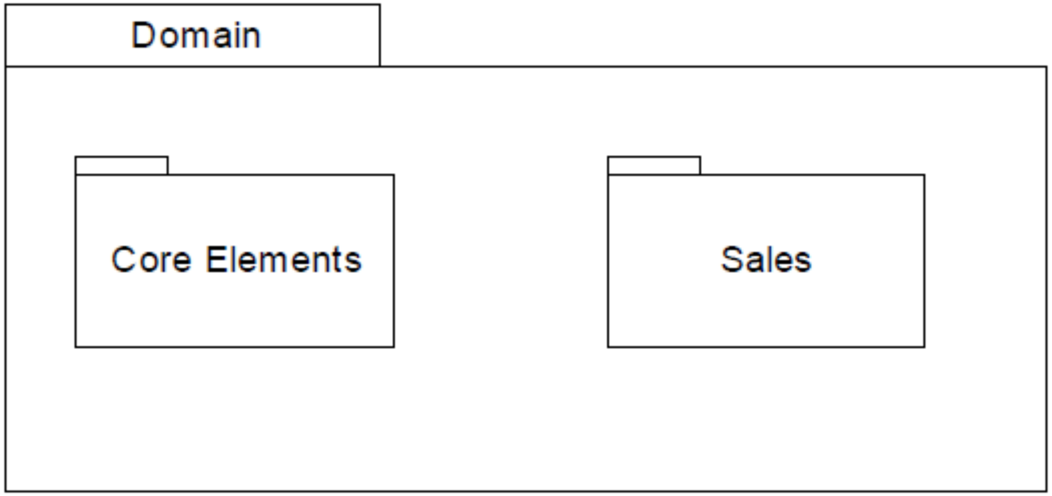


Layers and Partitions

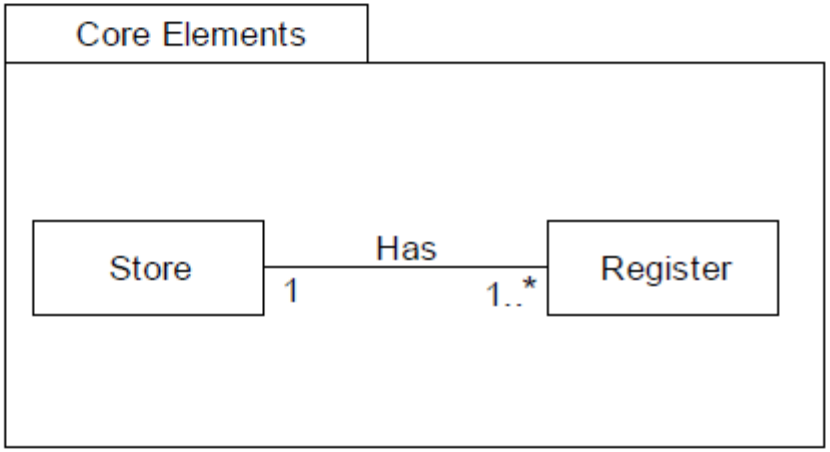
- The **layers** of an architecture are said to represent the **vertical slices**.
- While **partitions** represent a **horizontal division** of subsystems of a layer.



UML Package



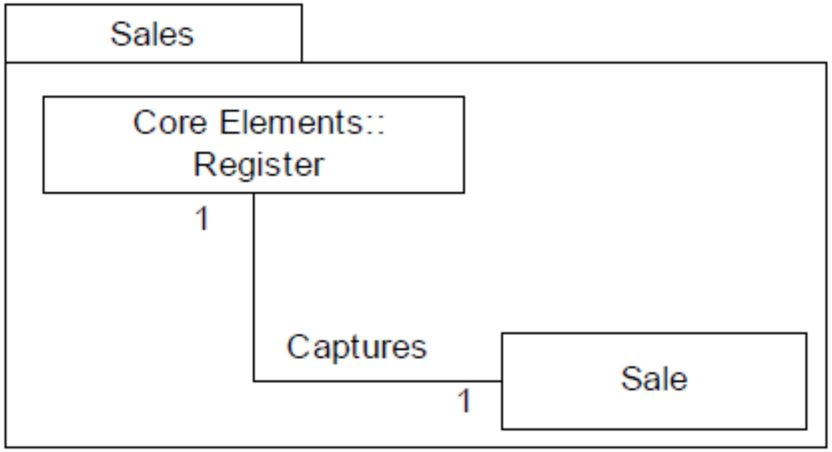
Classes in Packages



- Register class is part of Core Elements package.

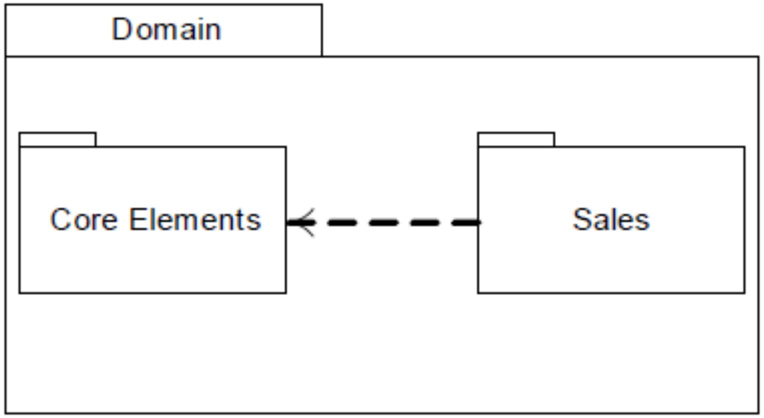
Referenced Class in Package

- A class shown in a foreign package may have new associations, but must otherwise remain unchanged.
  - Use “::” symbol to indicate it



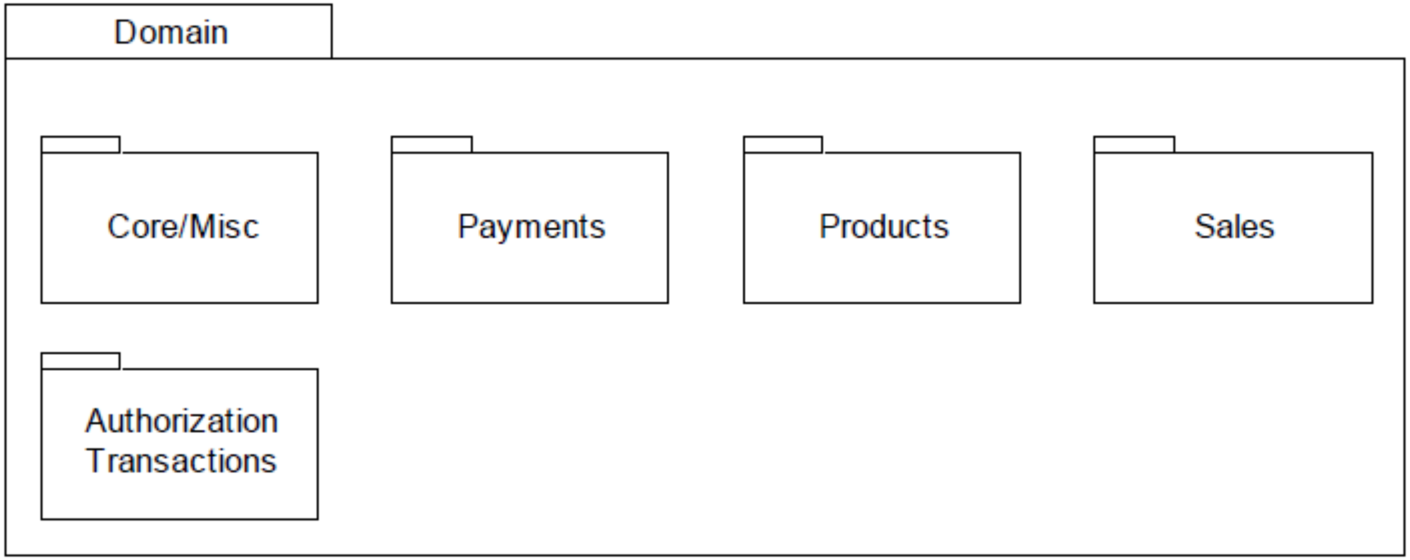
Package Dependency

- If a model element is dependent on another, the dependency may be shown with a dependency relationship with an arrowed line.
  - For example, if a package references an element owned by another, a dependency exists.

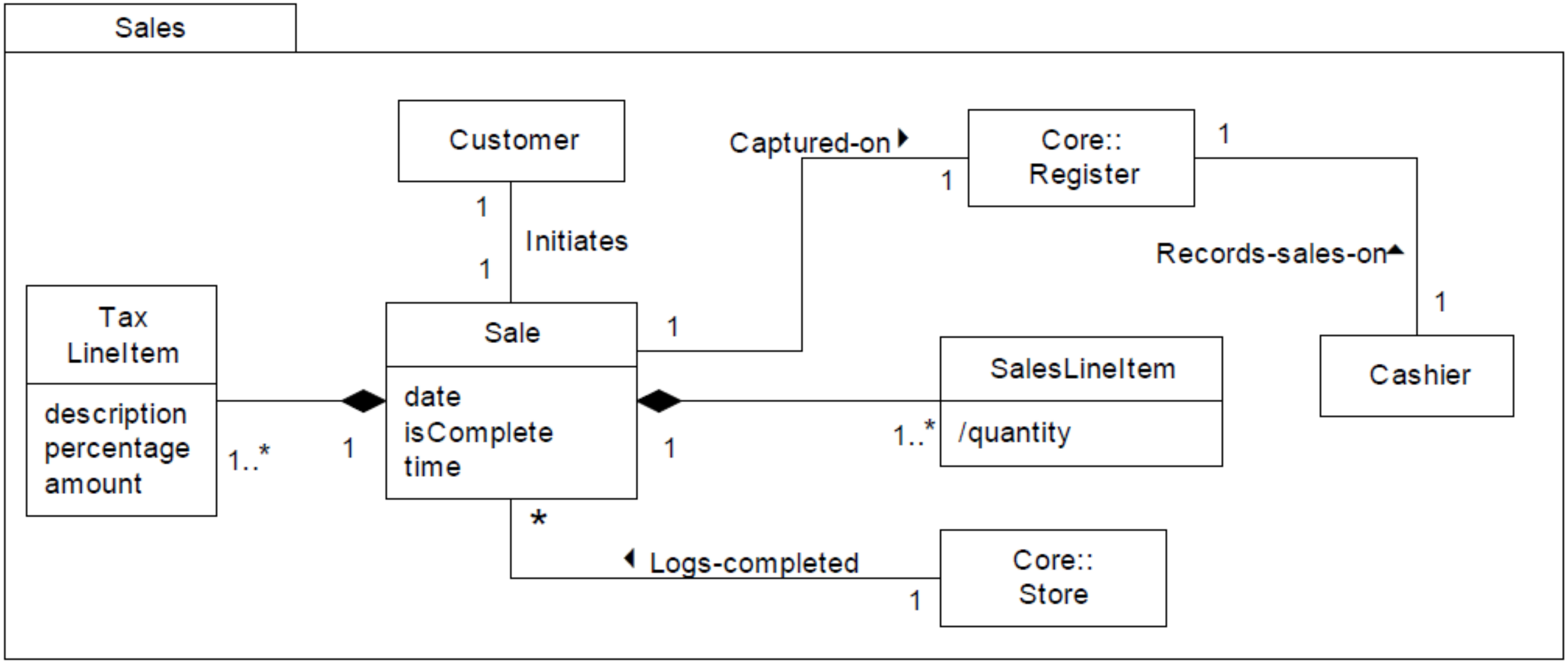


POS Domain Model Package

- *A* Core/Misc package is useful to own widely shared concepts or those without an obvious home.
  - In later references, the package name will be abbreviated to *Core.*

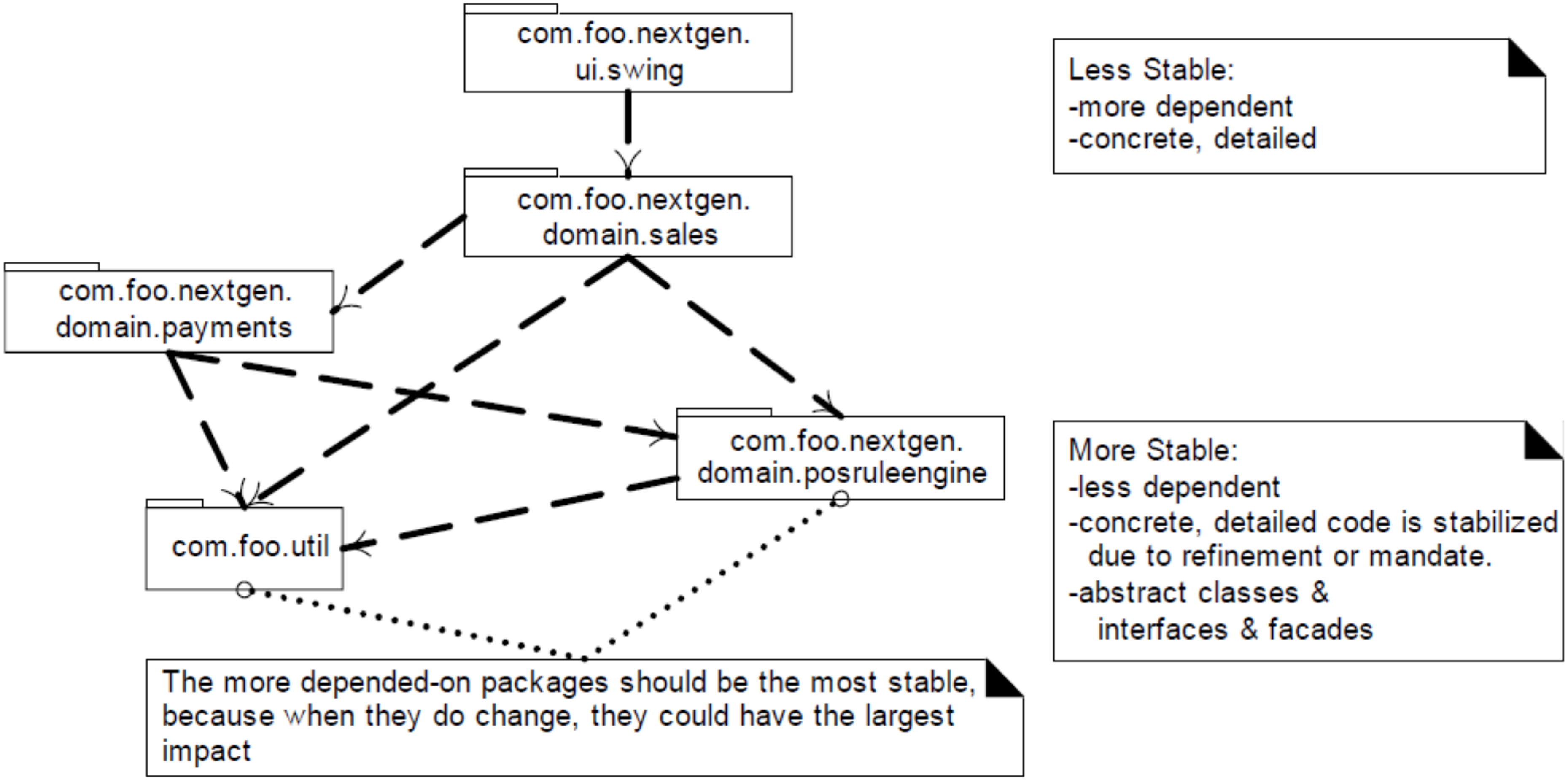


Sales Package



Most Responsible Are Most Stable

- As an extreme case, if a widely used utility package such as *com.foo.util* changed frequently, many things could break.



com.foo.nextgen.
ui.swing

com.foo.nextgen.
domain.sales

com.foo.nextgen.
domain.payments

com.foo.nextgen.
domain.posruleengine

com.foo.util

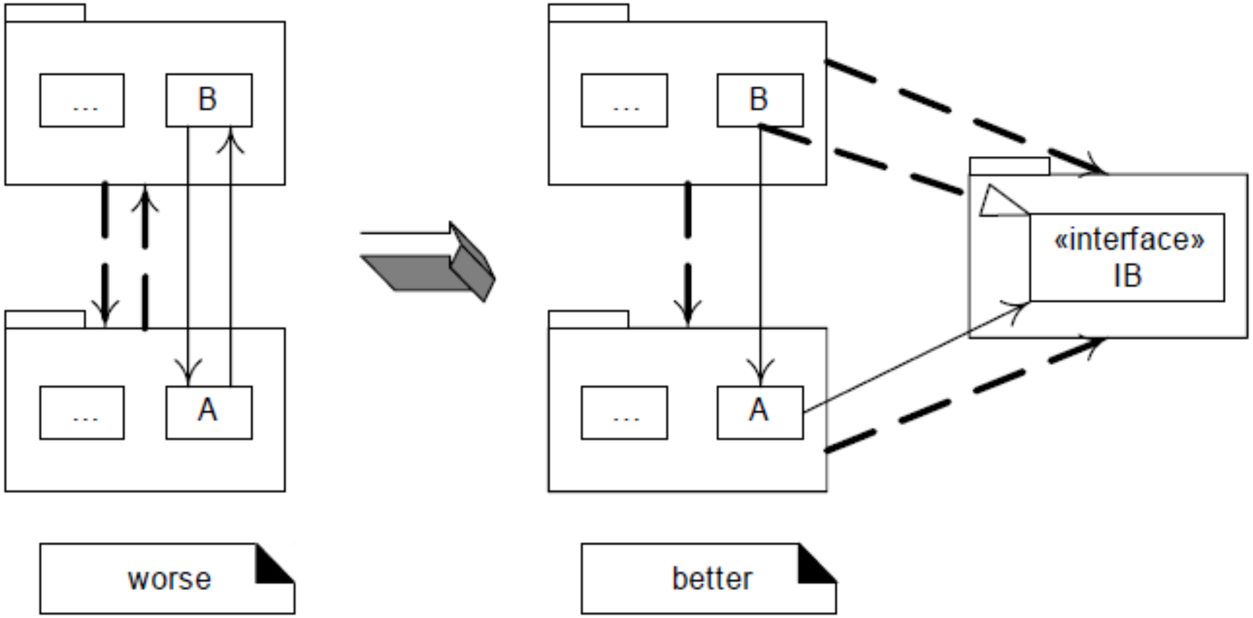
Less Stable:
-more dependent
-concrete, detailed

More Stable:
-less dependent
-concrete, detailed code is stabilized
due to refinement or mandate.
-abstract classes &
interfaces & facades

The more depended-on packages should be the most stable, because when they do change, they could have the largest impact

Breaking Cyclic Dependency

- If a group of packages have cyclic dependency then they may need to be treated as one larger package in terms of a release unit.
  - Break the cycle with an interface.



Quiz

- Why are package diagram useful in practice?
- How to break cyclic dependencies in package diagrams?
- Which packages should be the most stable?
- What are the alternative approaches to show package structure?

Actions

- Review Slides.
- Read Chapter 31 about package diagrams
  - *Applying UML and Patterns*, Craig Larman