

## **UML PACKAGE DIAGRAMS**

**UML package diagrams** are often used to illustrate the logical architecture of a system-the layers, subsystems, packages . A layer can be modeled as a UML package; It is part of the Design Model and also be summarized as a view in the Software Architecture Document.

**Logical Architecture** is the large scale organization of the software classes into packages subsystem and layers It is called logical architecture because there's no decision about how these elements are deployed across different operating system processes or across physical computers in a network.

**Layer** is a coarse grained grouping of classes , packages, or subsystems that has cohesive responsibility for major aspect of the system .

There are 2 types of Layers.

- 1) Higher Layer (Contain more application specific services ex: UI layer)
- 2) Lower layer (Contain more generalized services ex: Technical Services layer )

Higher Layer calls upon services of lower layer , but vice versa is not .

Typically layers in the Object Oriented System has 7 standard layers. The important layers are

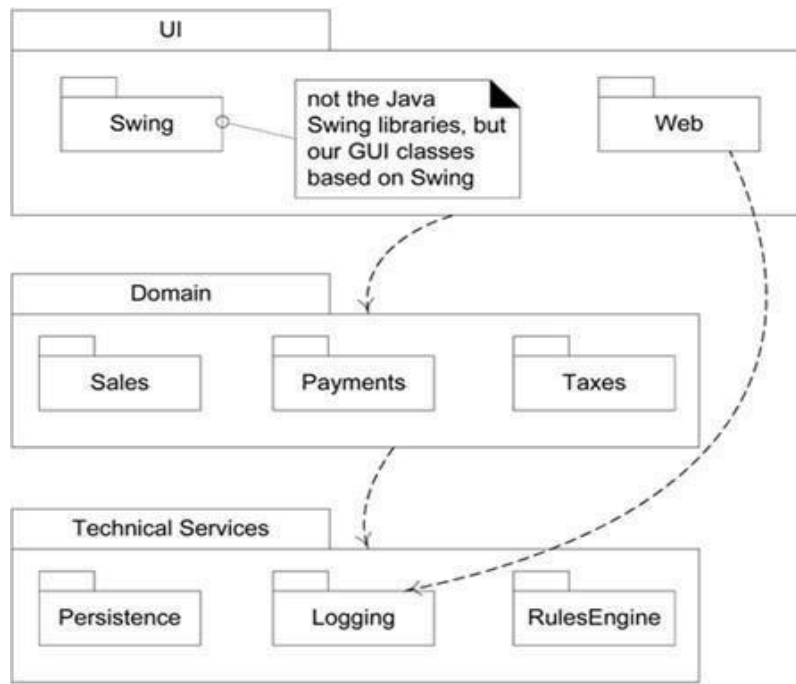
- **User Interface** – Has various I/O formats & forms.
- **Application Logic and Domain Objects** - software objects representing domain concepts (for example, a software class Sale) that fulfill application requirements, such as calculating a sale total.
- **Technical Services** general purpose objects and subsystems that provide supporting technical services, such as interfacing with a database or error logging.

### **Architecture Types**

**Strict layered architecture** - a layer only calls upon the services of the layer directly below it. This design is common in network protocol stacks, but not in information systems

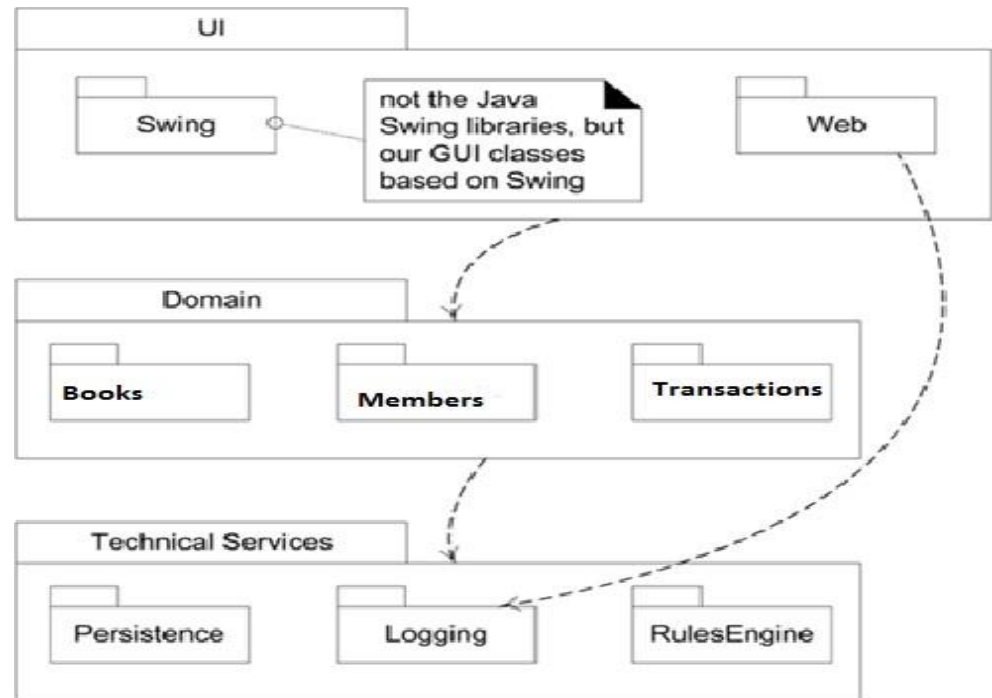
**Relaxed layered architecture** -a higher layer calls upon several lower layers. For example, the UI layer may call upon its directly subordinate application logic layer, and also upon elements of a lower technical service layer, for logging and so forth.

**Layers shown with UML package diagram notation.**



**Elements**

Name	Symbol	Description
Package		package can group anything: classes, other packages, use cases
Dependency		depended-on package
Fully qualified Name	<code>java::util::Date</code>	To represents a namespace (outer package named "java" with a nested package named "util" with a Date class)

**Package Diagram : Library Information System****WHEN TO USE PACKAGE DIAGRAMS**

1. It is used in large scale systems to picture dependencies between major elements in the system.
2. Package diagrams represent a compile time grouping mechanism.