SE 3112

SOFTWARE ARCHITECTURE

Prepared by: Dams Gabriel Lazarus
ICT University, Fall 2022

CHAPTER 2:

Modeling Software Architecture

Topics Overview

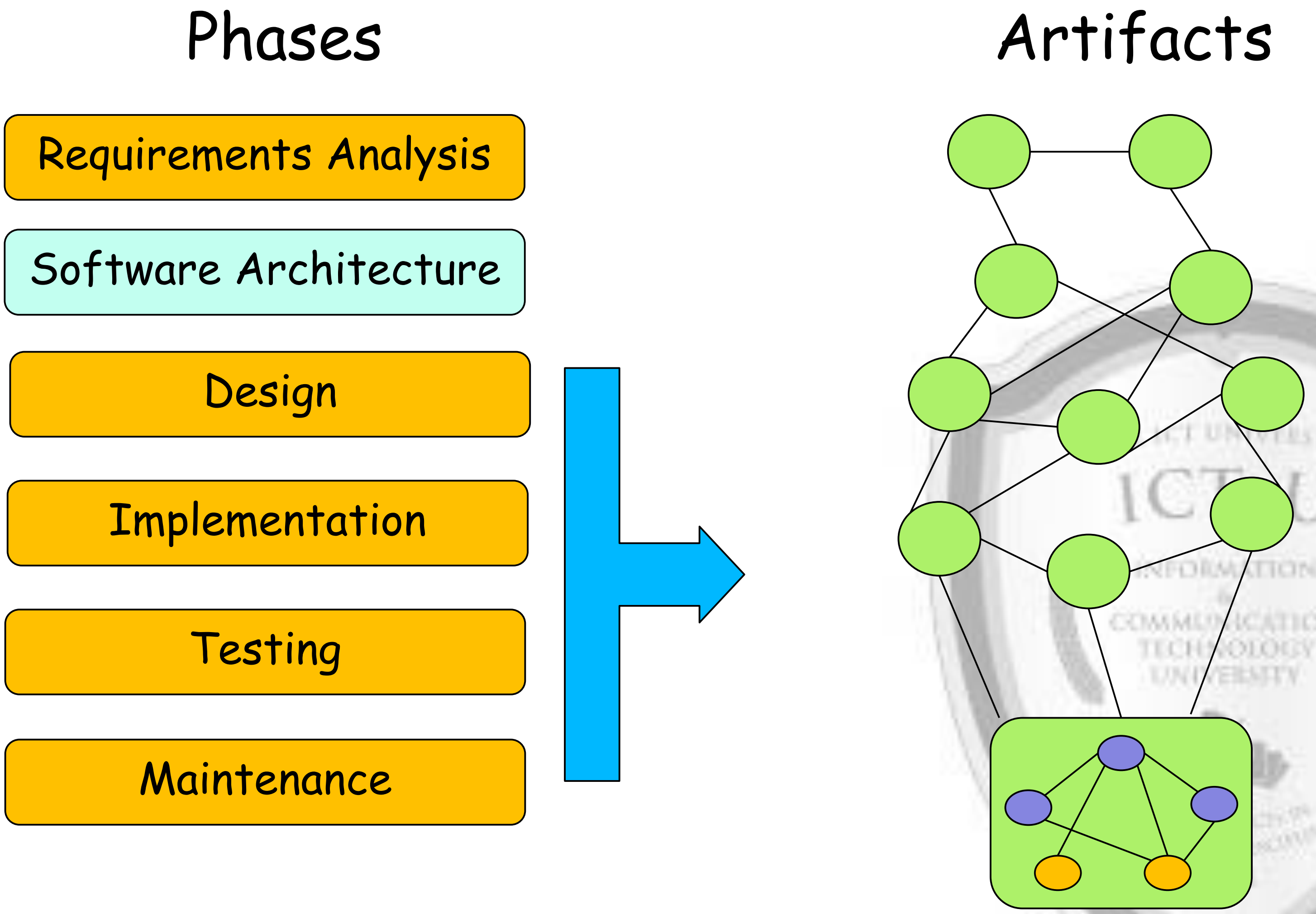
- Need for Modeling Architecture.
- Modeling Software Architecture in UML
- Architectural Modeling components
- Case Study Problem

Modeling is Essential

- Why do we need modeling in Software Architecture?

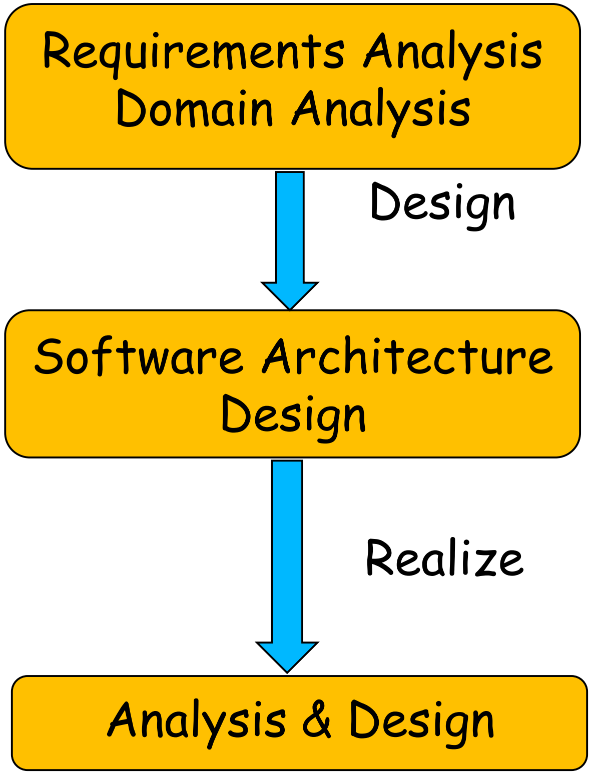
- To visualize the product before it is produced or developed.
- Models provide template for guiding the production
- Models document the decisions that are made.
- Models helps to communicate our ideas.

Models in Software Engineering



Architecture Modeling

Models (Artifacts)



Requirements Analysis Domain Analysis

Design

Software Architecture Design

Realize

Analysis & Design

Use case diagram/ Domain Model

Architecture Model

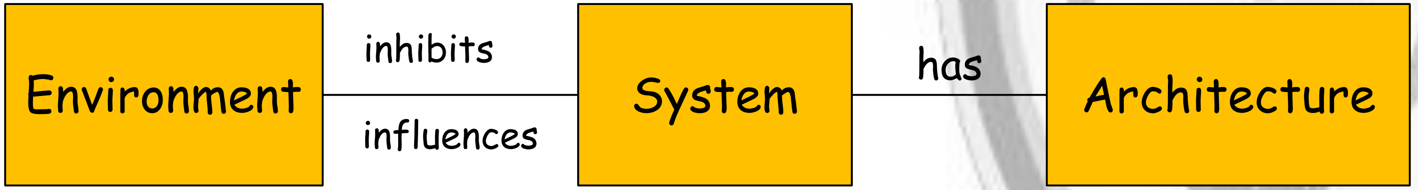
Class diagrams

Architecture Context

- Every system has an architecture

- Every system is composed of elements and there are relationships among them.

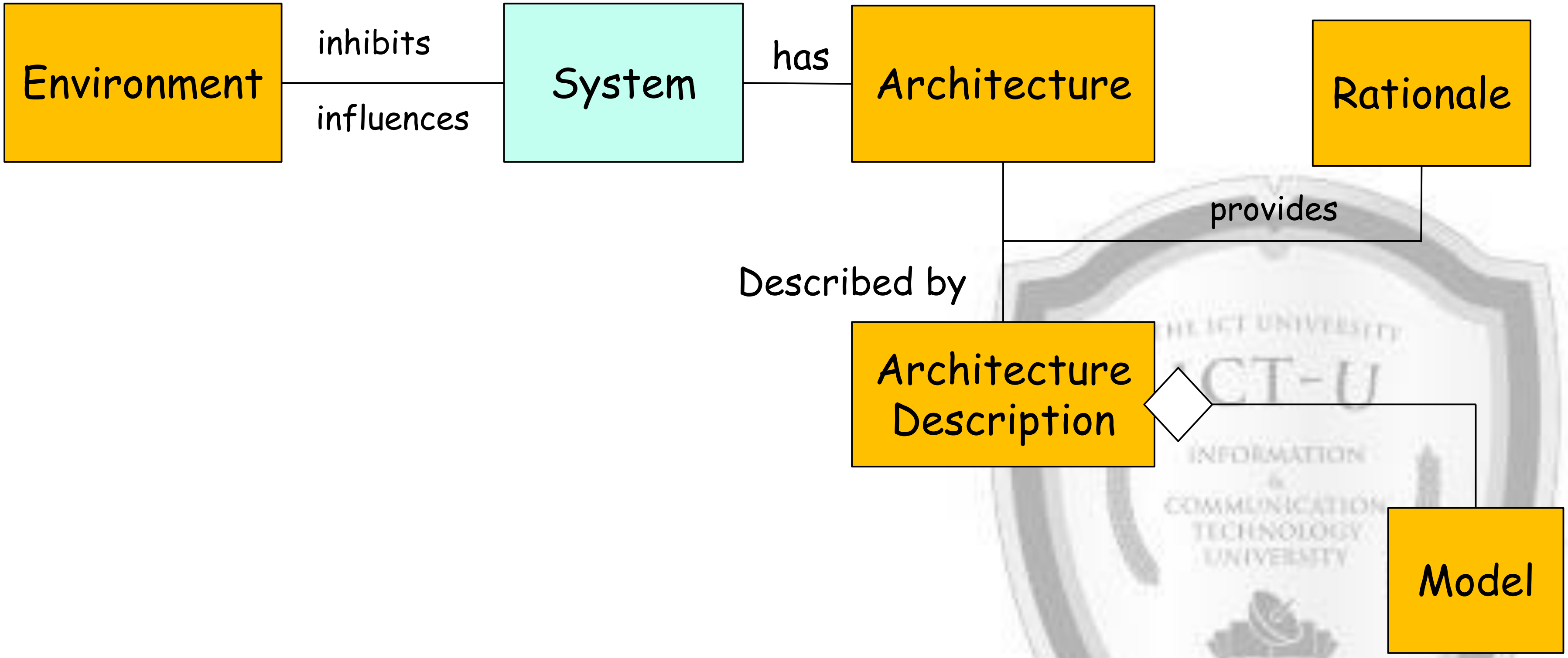
- In the simplest case, a system is composed of a single element, related only to itself.



- **Note**: if you want to control the system, it is better to specify the architecture.

Architectural Description

- Every system has an architecture

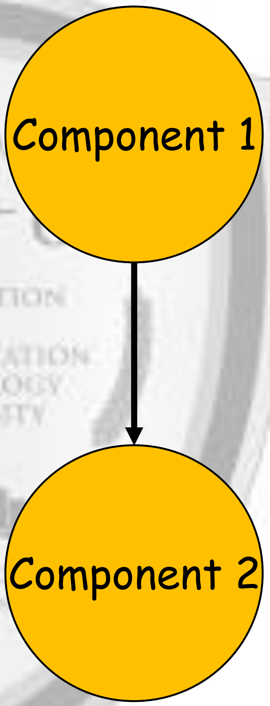


Software Architecture – IEEE 1471-2000

- *Software architecture is the fundamental organization of a system, embodied in its components,*
- *Their relationships to each other and the environment,*
- *And the principles governing its design and evolution.*
- How do we represent such definition in a model?

Using Informal Notation - Discussion

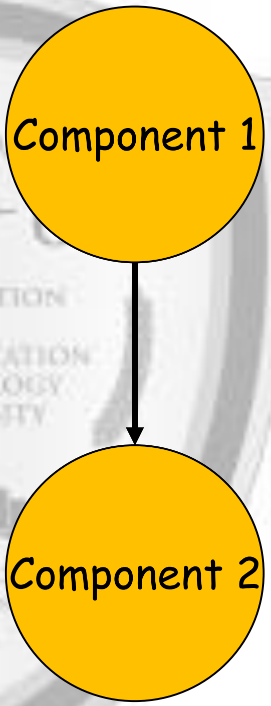
- *Typically, we can use some lines and boxes to form an architecture.*
- *The informal notation typically represents some architecture.*



Using Informal Notation - Discussion

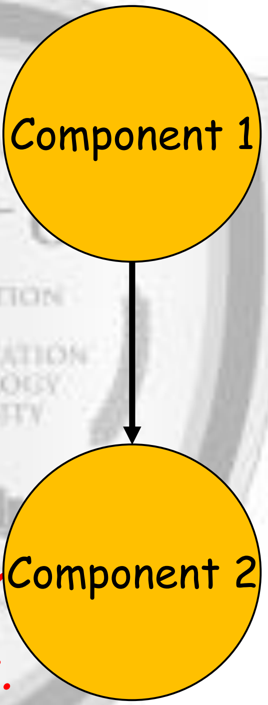
- *Suppose that someone (in an effort to communicate his idea to you) shows you the diagram/model below and says that this is an architecture.*

- *What is wrong with this?*



Using Informal Notation - Discussion

- *What is wrong with this?*
  - *Ambiguous and often leading to confusion*
  - *What are the components?*
    - *Class*
    - *Processes*
    - *Data Stores etc..*
  - *What are the relations?*
    - *Data flow*
    - *Control flow*
    - *Dependency link etc..*
  - *The model is open to different interpretation*
    - *This model cannot guide the software development.*



Architectural Modeling

- Two basic means of formally modeling software are: Visually or Textually.
- Visual
  - UML-based
  - Other
- Textual
  - Architecture Description Languages (ADLs)

Modeling Software Architecture in UML

- Why UML?
  - Is a de facto **standard**
  - **General purpose** modeling language that supports diverse application areas
  - Is **based on experience and needs** of the user community (OMG)
  - **Widespread use** - not only applicable to software design but also in non-software disciplines such as systems engineering and in the business domain.
  - **Supported by many tools**
  - Support the entire software development lifecycle.

How to Model Architectures in UML

- 1st Approach: Use existing UML models 'as is'

- 2nd Approach: Use UML's built-in extension mechanisms to refine/extend existing models

- 3rd Approach: Introduce new notations to represent architectural elements (change meta-model).

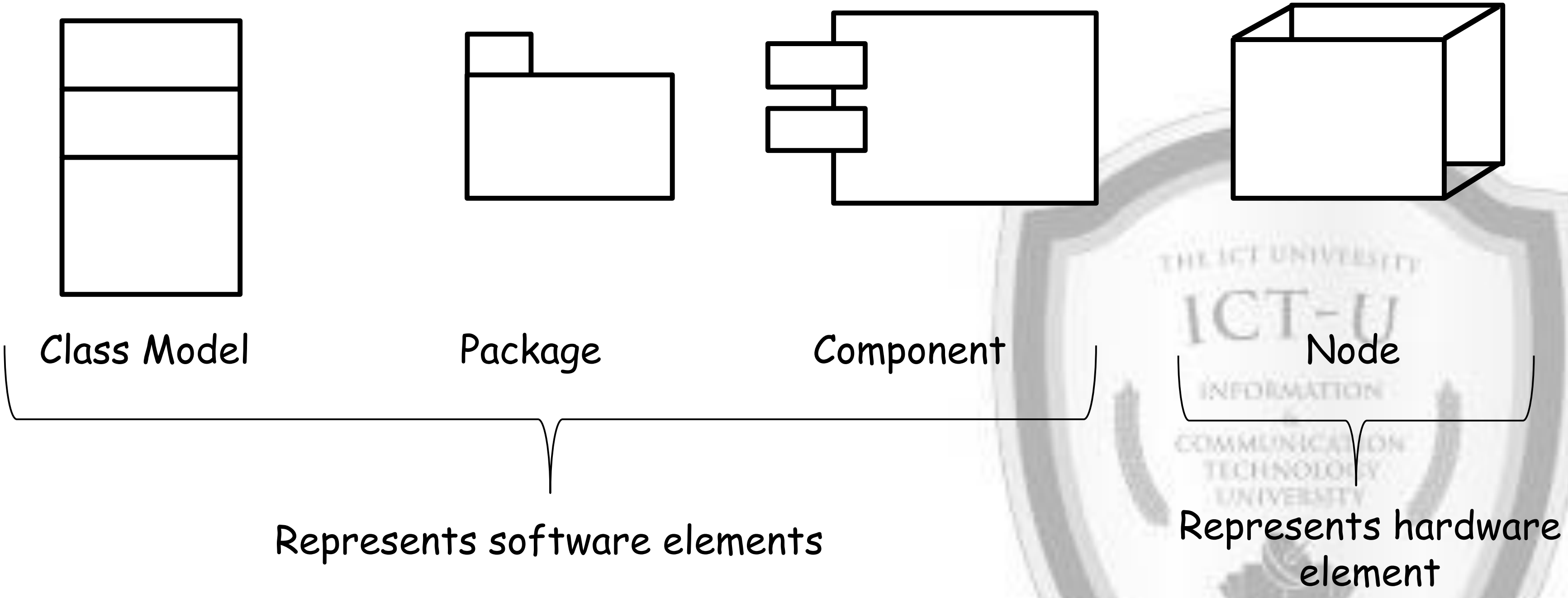
1st Approach: Use existing UML models 'as is'

- Using the defined notations from textbooks or some source to model the architecture.

- In doing that we need to typically answer two (2) questions:
  - What are the candidate UML models for representing architectural components?
  - What are the candidate UML relations for representing architectural relations?

Using Structural Models

- Basically there are four (4) types of structural models



Class Model

Package

Component

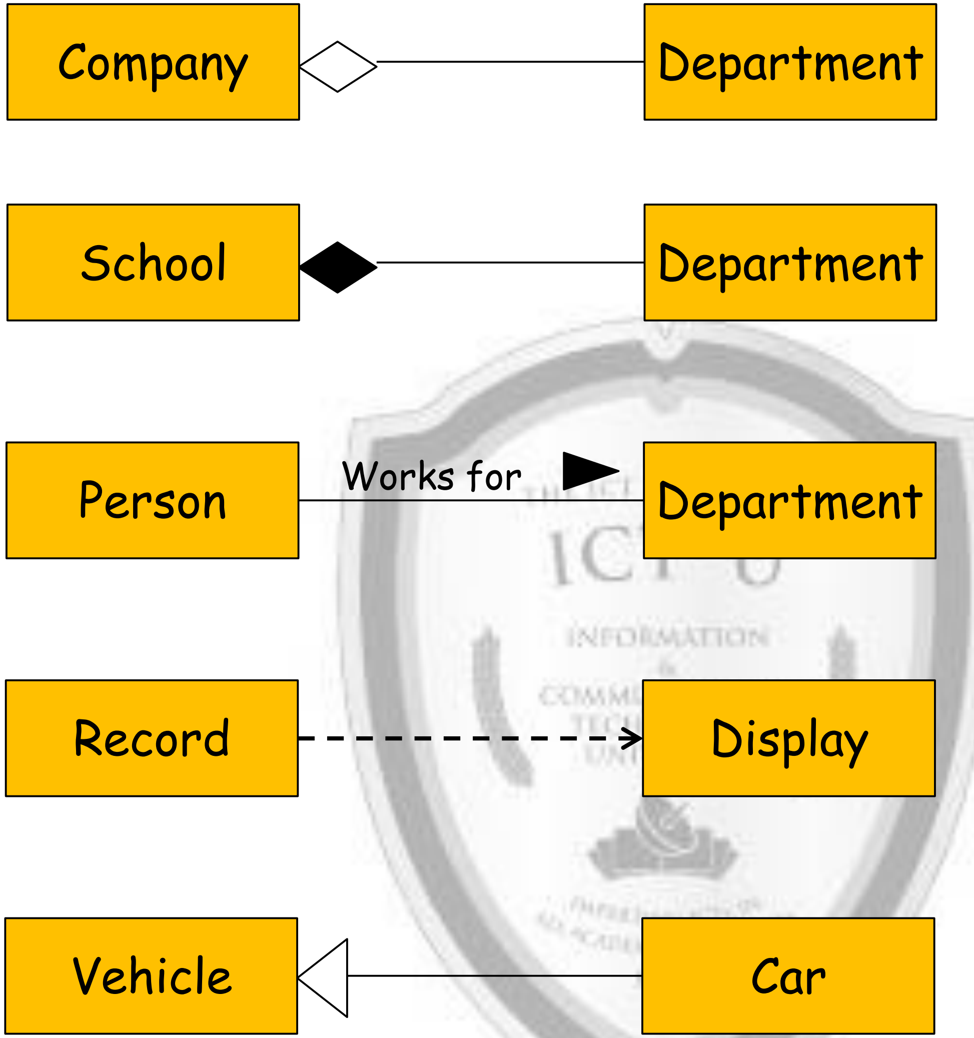
Node

Represents software elements

Represents hardware element

UML Relations

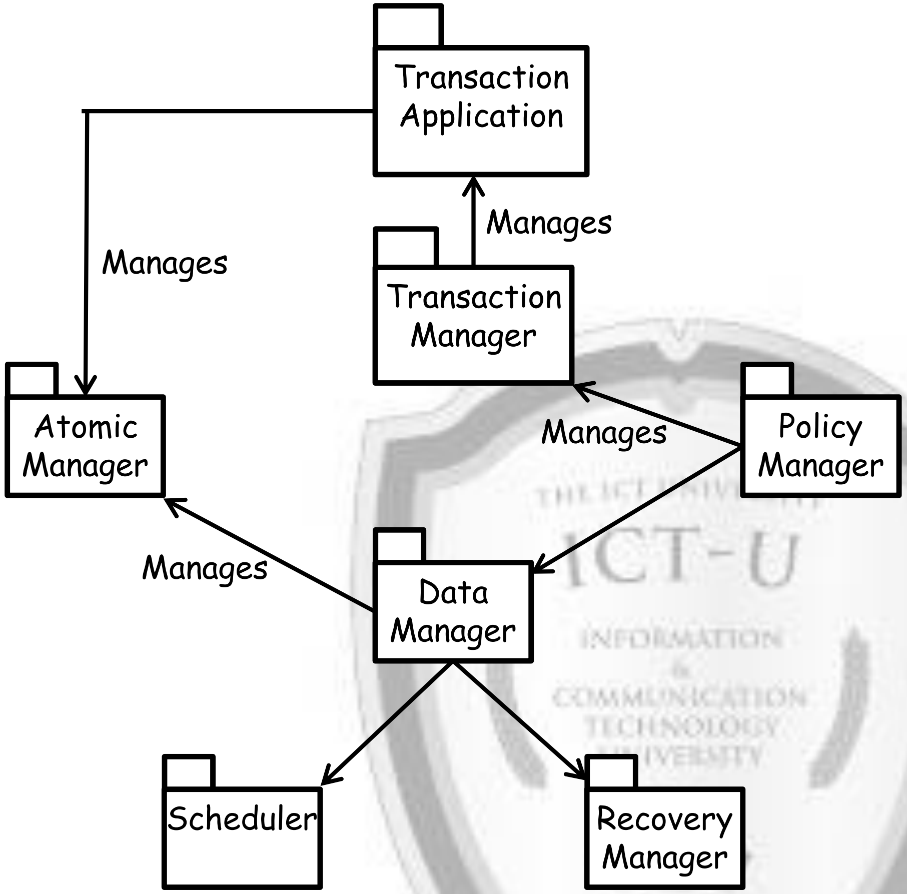
- Aggregation
- Composition
- Association
- Dependency
- Generalization



Architectural Components as Packages

Example:

Transaction System Architecture



- Packages are only grouping constructs;
- Less useful

2nd Approach: Using UML extensions

- Three built-in extensibility mechanisms
- Stereotypes
  - Refines existing models to define new notations
- Tagged values
  - Adds new properties to existing notations
- Constraints
  - Extends semantics of existing notations

Need for Modeling using as Architects

- Model is a way to communicate to audience and stakeholders what we are building and what we are going to build.
- Model will give shape to the abstraction or abstract requirement
- Model helps to decompose big problems
- Its a good tool to identifying non functional requirements (NFR's)

Who are the Audiences?

- Non Technical
- CEO
- Marketing Team
- Product Owners
- Business

- Technical Team
- Architects
- Developers
- Testers

Architectural Views

- When you create architectural diagrams/models, consider the different views of the audience.
- There are tons of UML diagrams but you don't need to use all.
- You are to use the ones that brings value to the project.

Case Study (Autocab - A Self Drive Taxi Company)

- Context
  - AutoCab is a traditional business company where Taxis can be booked by calling call center. The company wants to expand her business to a self driving Taxi system. The company wants booking of the self driving taxi (when launched) to be done via AutoCab web or mobile application. The self driving taxi will confirm the trip, will pick up the customer at his/her desired location. Once the trip is finished, customer will pay online and the trip is considered completed and the self taxi is ready for the next pick up.

Case Study (AutoCab - A Self Drive Taxi Company)

- Requirements
  - AutoCab expects the application to be up 24/7.
  - Self driving Taxi is going to generate loads of data, and these data need to be analyzed for performance, engine health.
  - The application should be able to handle millions of customers
  - All the transactions need to be secured and should be audited.

Case Study (Autocab - A Self Drive Taxi Company)

- Additional info.
  - AutoCab wants to launch the application first in Yaoundé and she has a plan to expand to the rest of Cameroun by the end of the year.
  - Support push capabilities of messages and updates to customer application and self driving taxi by the Admin
  - The user interface of the application should be straight forward and not difficult to use
  - The payment is using existing payment gateway system

Q & A