COSC 31093 / BECS 31213
Enterprise Software Design and Architecture

Architecture Patterns

Dr. B.M. Thosini Kumarika

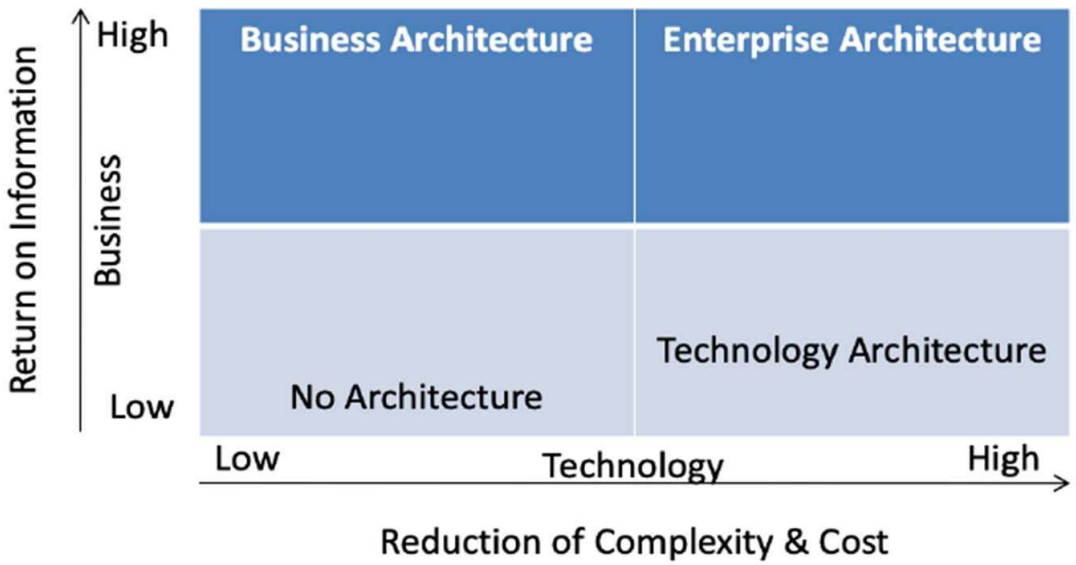
Outline

- Introduction to Architecture
- Key Attributes
- Issues without architectures
- Architecture Patterns
  - Layered
  - MVC
  - Component Architectures

Winchester Mystery House

- Mansion in California
- 147 builders and 0 architects
- The mansion contains 160 rooms, 40 bedrooms, 6 kitchens, 2 basements and 950 doors
- Of there 950 doors, 65 of them open to blank walls
- 13 staircases were built and abandoned
- 24 skylights were installed into various floors

Introduction to Architecture



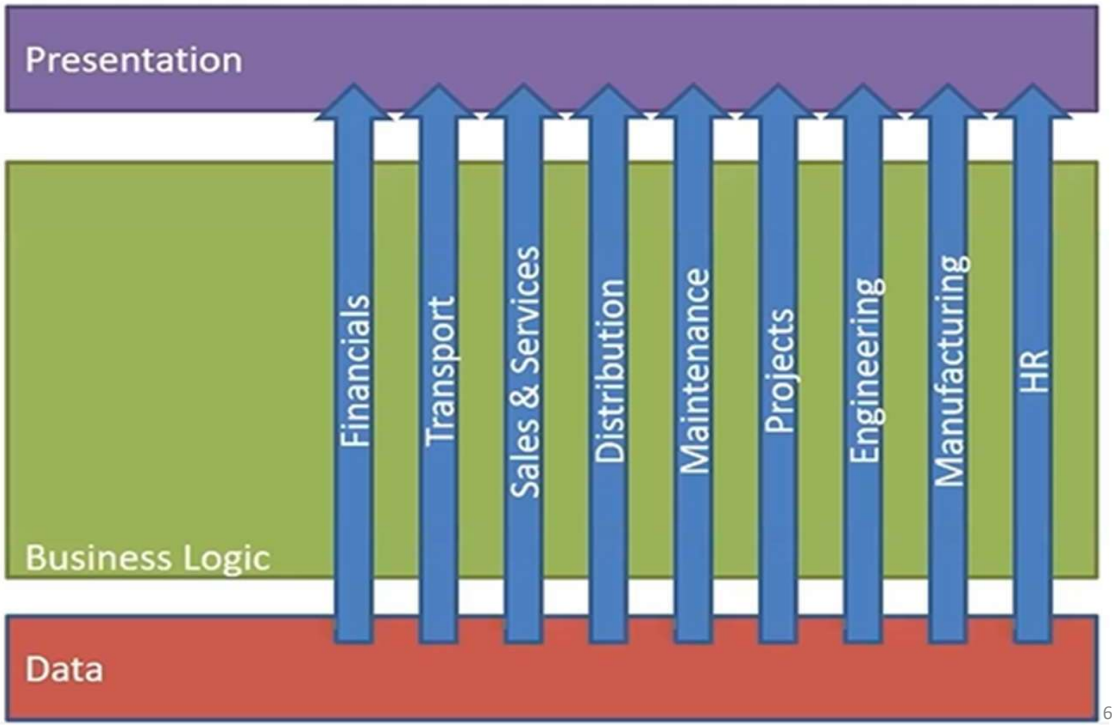
Enterprise Architecture

Definitions

- Enterprise Architecture is about understanding all of the different elements that go to make up the enterprise and how those elements interrelate. [The Open Group]

- Enterprise Architecture is a strategic information asset base, which defines the business mission, the information necessary to perform the mission, and the transitional processes for implementing new technologies in response to the changing mission needs. [USA Federal CIO Council]

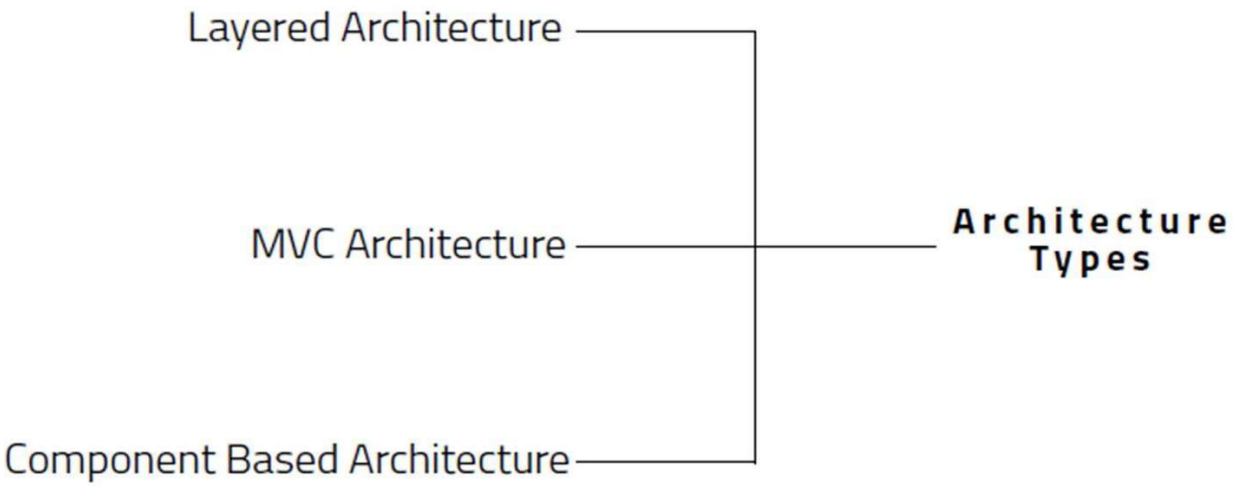
Logical Components of an Enterprise Architecture



Without Architectures

Unpredictable Solutions
Poor Quality and Reduced User Experience
Stunted Evolution
Rigid Systems (hard to reuse)

Types of Architectures

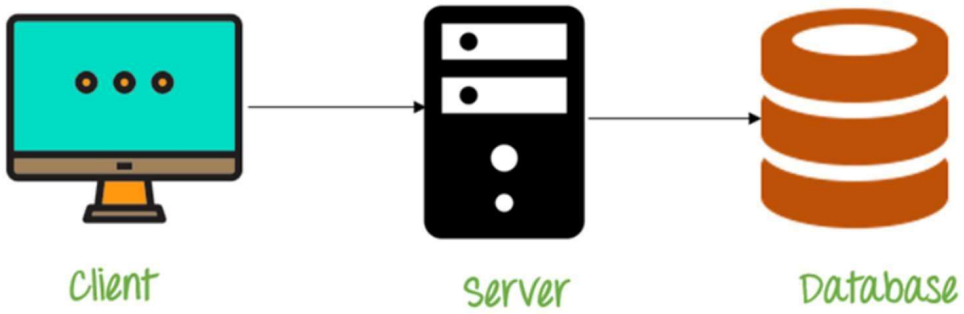


Layered Architecture

- **Started with Client/Server**
  - Thin Client – Web clients [Google Docs/Photo Editor web]
  - Fat Client – Window Clients [Word Document/Photo Editor/Desktop]

- **Evolution**
  - Two Tier – User Interface and Database
  - Three Tier – UI, Business Logic Layer, Database

Client – Server Architecture



10

Another Type of Layered Architecture

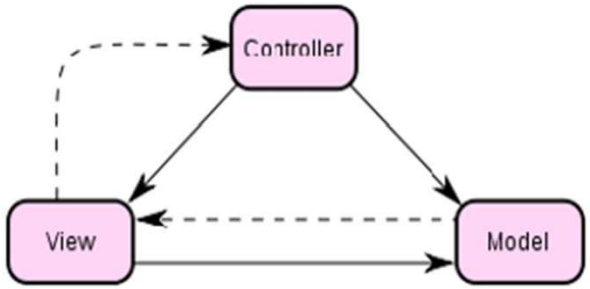
Multilayer Architecture

- Presentation layer (UI layer/View layer/Presentation layer)
- Business Logic Layer (Business Layer)
- Application layer (Service Layer)
- Data Access Layer (Persistence Layer)

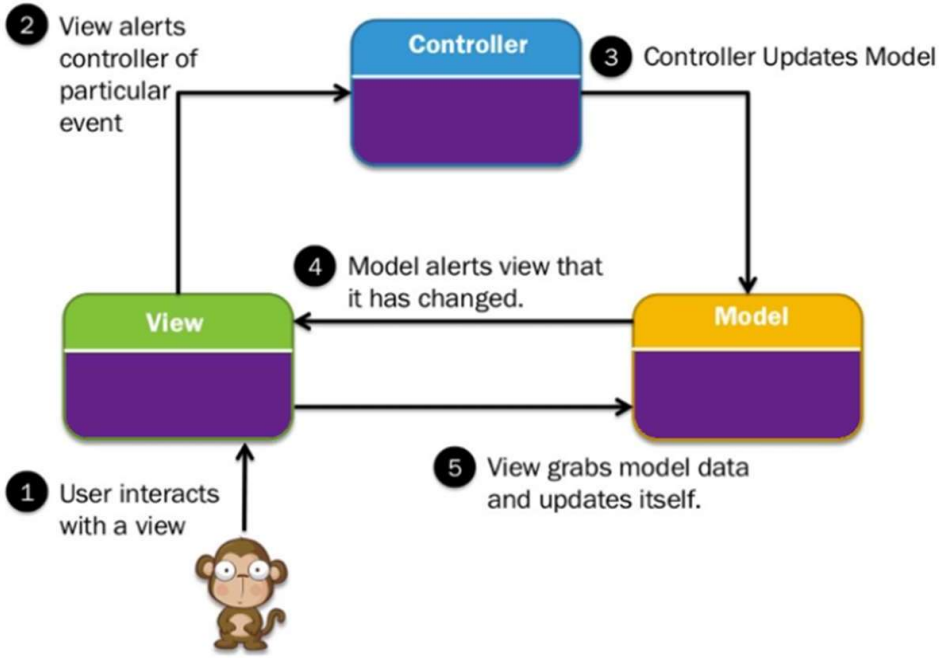
Model, View and Controller (MVC) Architecture

Model, View and Controller

- Model - A model represents an application's data and contains the logic for accessing and manipulating that data
- View - The presentation semantics are encapsulated within the view
- Controller - The controller is responsible for intercepting and translating user input into actions to be performed by the model



Model, View and Controller (MVC) Architecture

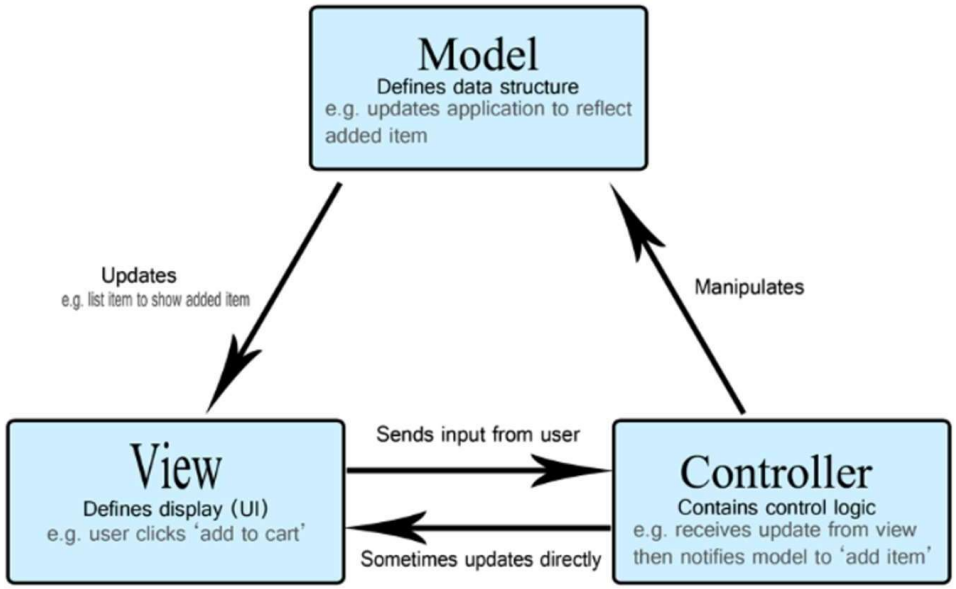


Model, View and Controller (MVC) Architecture

Web Browser and Web Page example

- Model - Encapsulate data in the database.
- View - Presentation layer, web pages, CSS, Java Script.
- Controller - Process and respond to events.

Model, View and Controller (MVC) Architecture



MVC Architecture – Model and View

```
package mvc.models;

public class Model {

    private int x;

    public Model(){
        x = 0;
    }

    public Model(int x){
        this.x = x;
    }

    public void incX(){
        x++;
    }

    public int getX(){
        return x;
    }
}
```

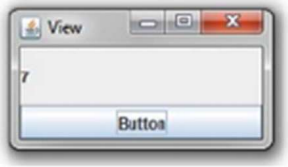
```
package mvc.views;

import javax.swing.*;
import java.awt.BorderLayout;

public class View {

    private JFrame frame;
    private JLabel label;
    private JButton button;


    public View(String text){
        frame = new JFrame("View");
        frame.getContentPane().setLayout(new BorderLayout());

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(200,200);
        frame.setVisible(true);

        label = new JLabel(text);
        frame.getContentPane().add(label, BorderLayout.CENTER);

        button = new JButton("Button");
        frame.getContentPane().add(button, BorderLayout.SOUTH);
    }

    public JButton getButton(){
        return button;
    }

    public void setText(String text){
        label.setText(text);
    }


}
```

MVC Architecture – Controller

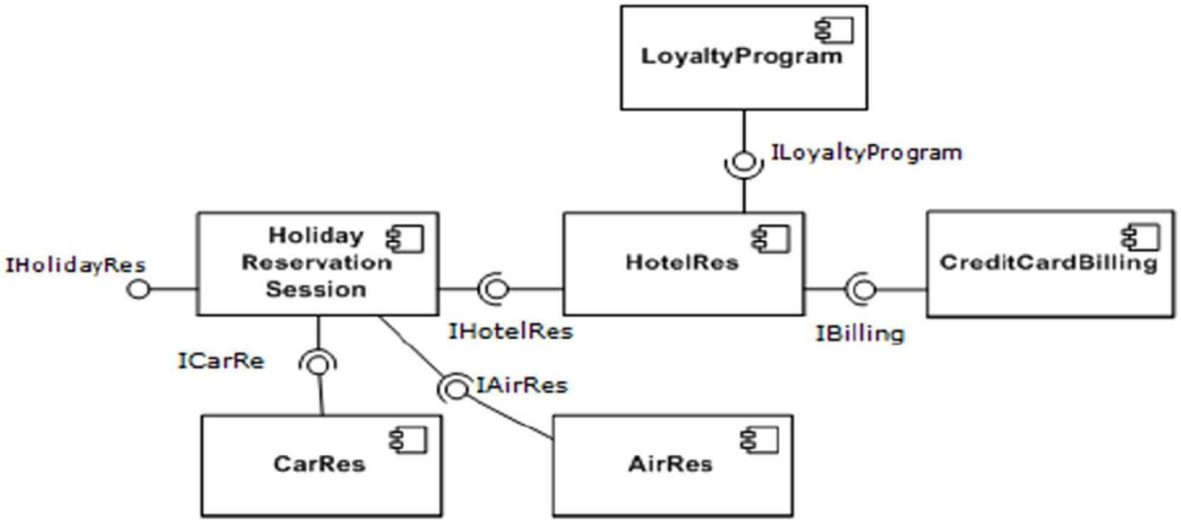
```
package mvc.controllers;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import mvc.models.*;
import mvc.views.*;

public class Controller {

    private Model model;
    private View view;
    private ActionListener actionListener;

    public Controller(Model model, View view){
        this.model = model;
        this.view = view;

    }

    public void contol(){
        actionListener = new ActionListener() {
              public void actionPerformed(ActionEvent actionEvent) {

                  linkBtnAndLabel();
              }
        };
        view.getButton().addActionListener(actionListener);
    }

    private void linkBtnAndLabel(){
        model.incX();
        view.setText(Integer.toString(model.getX()));
    }
}
```

MVC Architecture - Main Class

```
package mvc;

import javax.swing.SwingUtilities;

import mvc.models.*;
import mvc.views.*;
import mvc.controllers.*;

public class Main
{
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                Model model = new Model(0);
                View view = new View("-");
                Controller controller = new Controller(model,view);
                controller.contol();
            }
        });
    }
}
```



Component Based Architecture



Component Based Architecture (Example)

Component Based Architecture

Why Component architecture?

- Enterprise Applications Involved high complexity
- Adopt new functionality in low-cost manner
- Easier for new customers to implement
- Existing customer should be able to upgrade to new functionality

Component Based Architecture

What is a component?

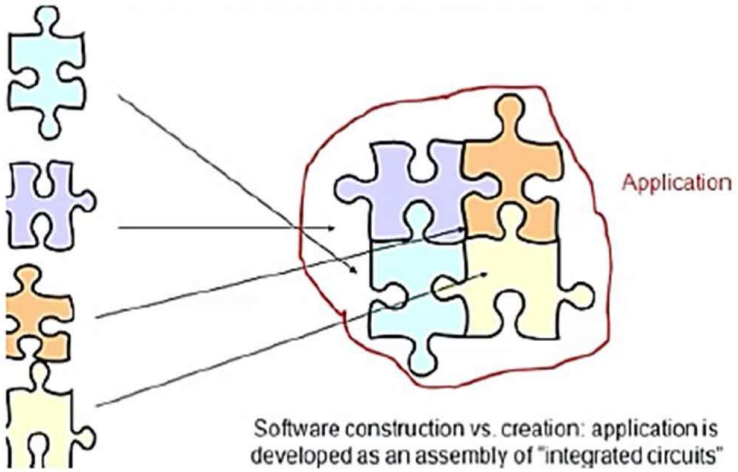
- Individual module which has following attributes

✓ Separation – All data and functions are semantically related.
✓ Interfacing – Can upgrade with new components.
✓ Standardization – Agreed Standard interface.

- Example – PC
  Constructed by different manufactures.
  Assembled very rapidly
  Assembler is not an expert in internals of each component.

Component Based Architecture Goals

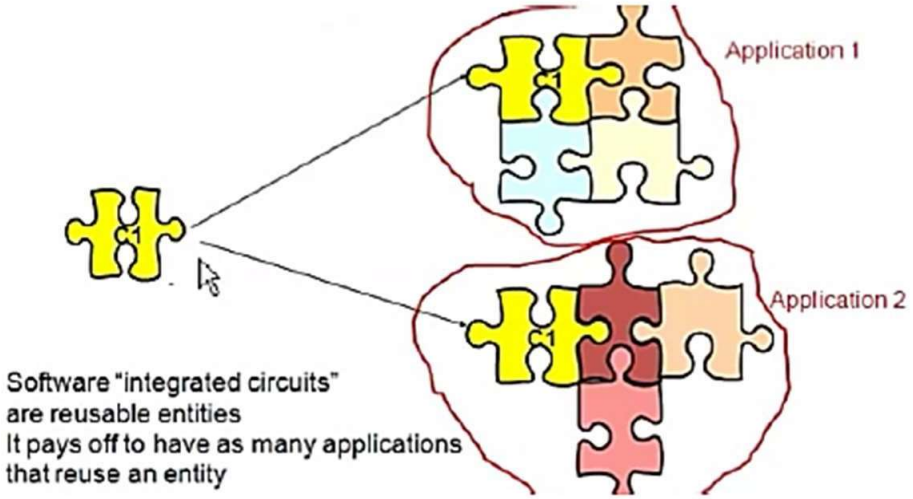
- Goal 01: Software Construction



Software construction vs. creation: application is developed as an assembly of "integrated circuits"

Component Based Architecture Goals

- Goal 02: Reuse



Component Based Architecture Goals

- Goal 03: Maintenance and Evolution