Common Architectures

Instructor:

Agenda

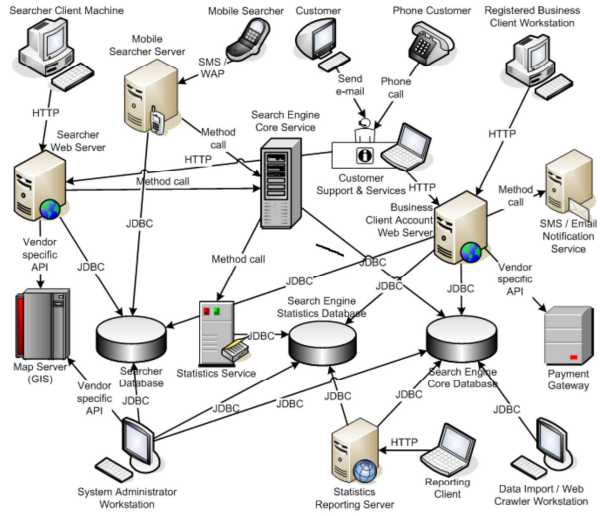
1. What is Software Architecture?
2. Client-Server Architecture
3. 3-Tier / Multi-Tier Architectures
4. MVC (Model-View-Controller)

What is Software Architecture?

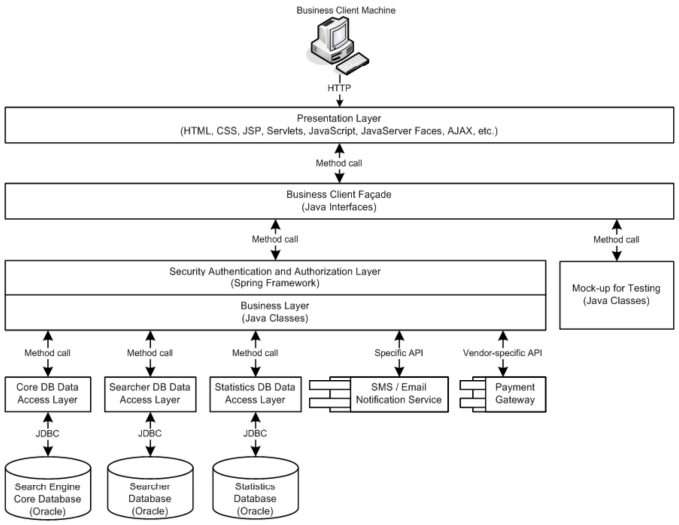
Software Architecture

- Software architecture is a technical blueprint explaining how the system will be structured
- The system architecture describes:
  - How the system will be decomposed into subsystems (modules)
  - Responsibilities of each module
  - Interaction between the modules
  - Platforms and technologies
- Each module could also implement a certain architectural model / pattern

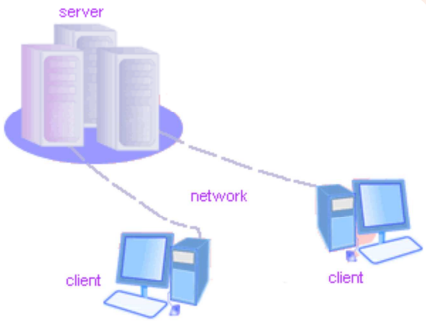
System Architecture Diagram – Example



Example of Multi-Tier Software Architecture



Client-Server Architecture

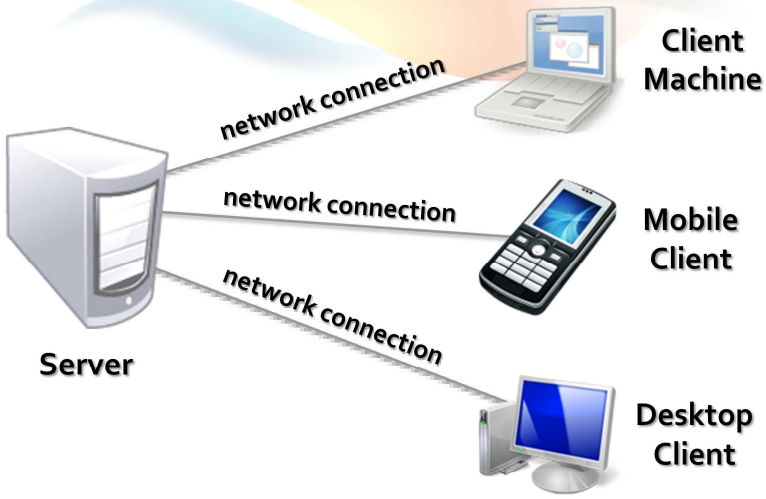


The Classical Client-Server Model

Client-Server Architecture

- The client-server model consists of:
  - Server – a single machine / application that provides services to multiple clients
    - Could be IIS based Web server
    - Could be WCF based service
    - Could be a services in the cloud
  - Clients –software applications that provide UI (front-end) to access the services at the server
    - Could be WPF, HTML5, Silverlight, ASP.NET, …

The Client-Server Model

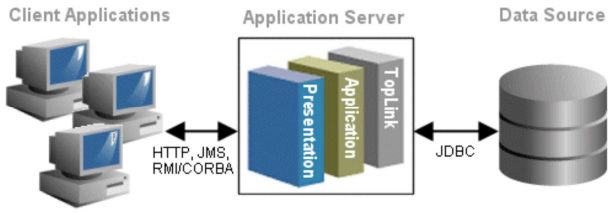


Client-Server Model – Examples

- Web server (IIS) – Web browser (Firefox)
- FTP server (ftpd) – FTP client (FileZilla)
- EMail server (qmail) – email client (Outlook)
- SQL Server – SQL Server Management Studio
- BitTorrent Tracker – Torrent client (µTorrent)
- DNS server (bind) – DNS client (resolver)
- DHCP server (wireless router firmware) – DHCP client (mobile phone /Android DHCP client/)
- SMB server (Windows) – SMB client (Windows)

3-Tier / Multi-Tier Architectures

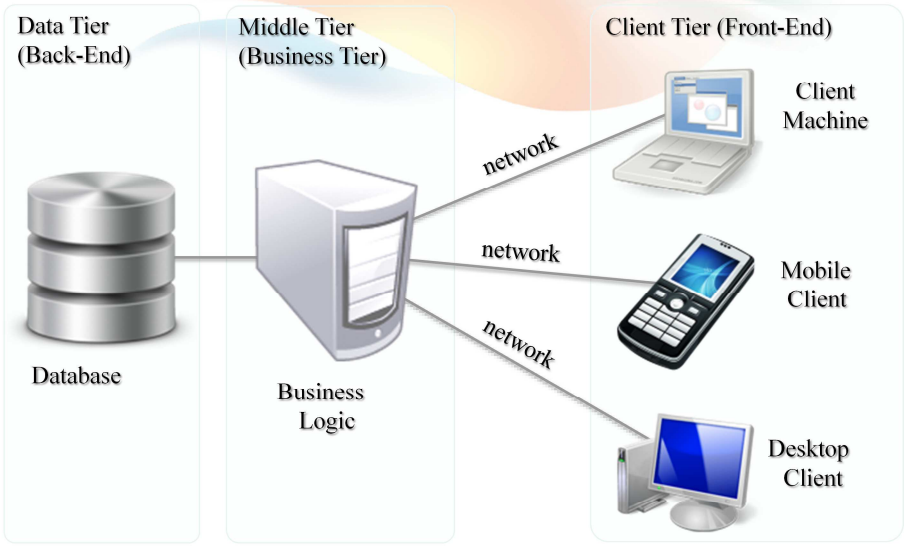
Classical Layered Structure of Software Systems



The 3-Tier Architecture

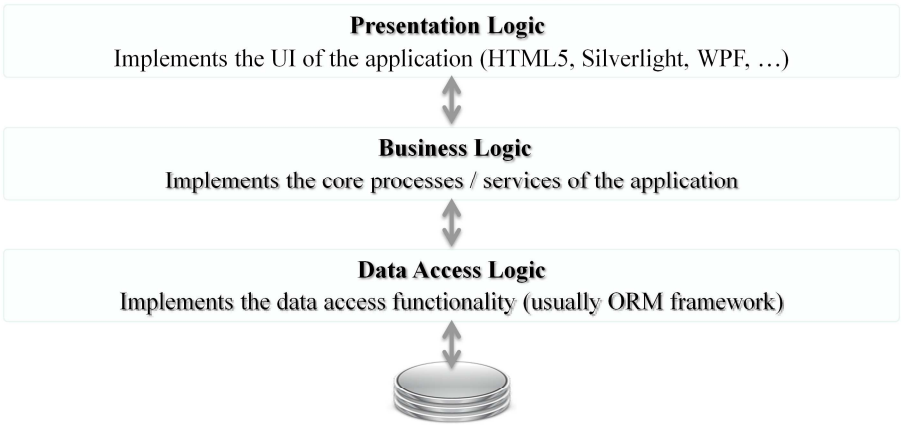
- The 3-tier architecture consists of the following tiers (layers):
  - Front-end (client layer)
    - Client software – provides the UI of the system
  - Middle tier (business layer)
    - Server software – provides the core system logic
    - Implements the business processes / services
  - Back-end (data layer)
    - Manages the data of the system (database / cloud)

The 3-Tier Architecture Model

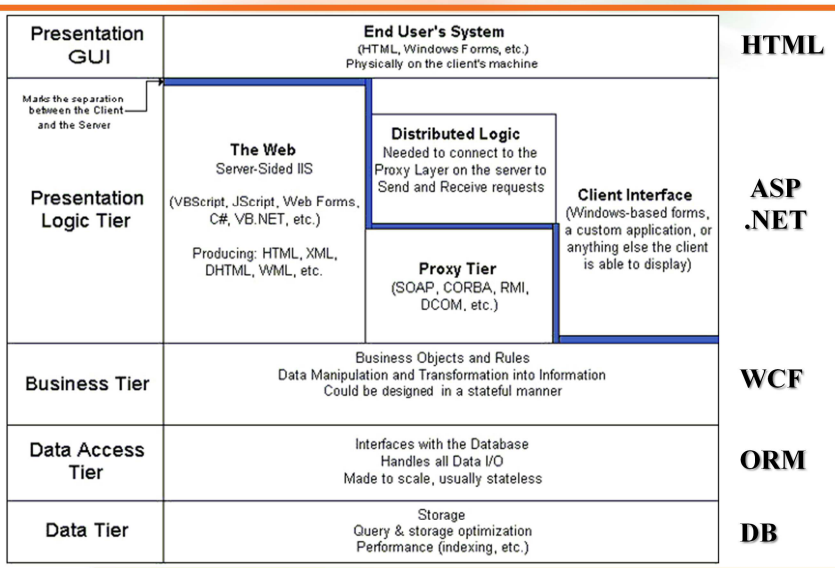


Typical Layers of the Middle Tier

- The middle tier usually has parts related to the front-end, business logic and back-end:

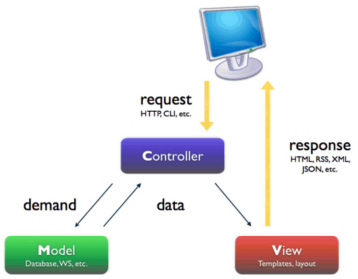


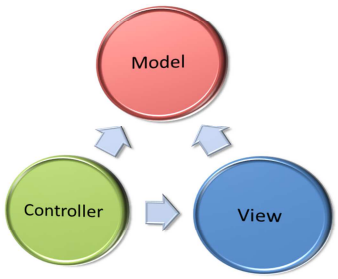
Multi-Tier Architecture



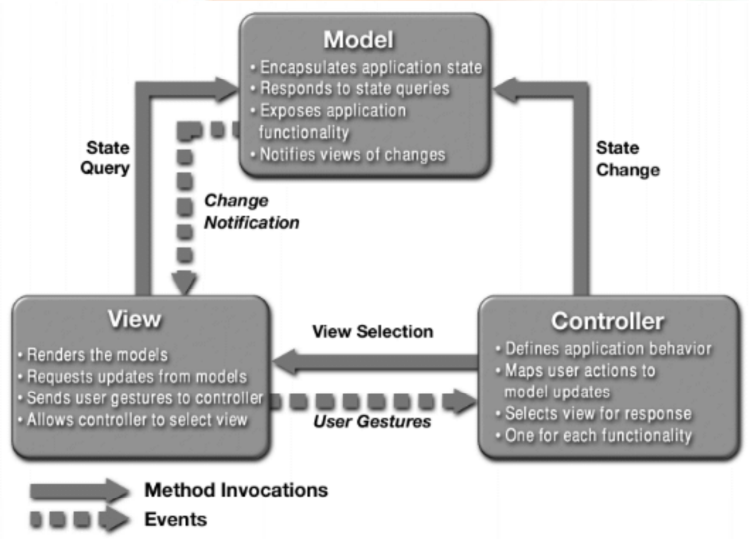
MVC (Model-View-Controller)

What is MVC and How It Works?





MVC Architecture Blueprint



Model-View-Controller (MVC)

- Model-View-Controller (MVC) architecture
  - Separates the business logic from application data and presentation
- Model
  - Keeps the application state (data)
- View
  - Displays the data to the user (shows UI)
- Controller
  - Handles the interaction with the user

MVC - Model

- Models data and behavior behind business process
- Manages Information - If Changes
- Contains data and Related Functionality
- Maps Real-World Entities
- Performing DB Queries
- Calculating Business Process
- Encapsulates Domain Logic which are independent of Presentation

MVC - View

- Obtains data from model & presents to the user
- Represents Output/Input of the application
- Display results of Business Logic
- Free Access to Model
- Reads Data from Model – Using Query Methods

MVC - Controller

- Serves logical connection between user's interaction and the business process
- It receives and Translates input to request on model or view
- Input from user and instructs the model and view to perform action
- Responsible for making decision among multiple presentation
- Maps the end-user action to the application response

Relationship between Components

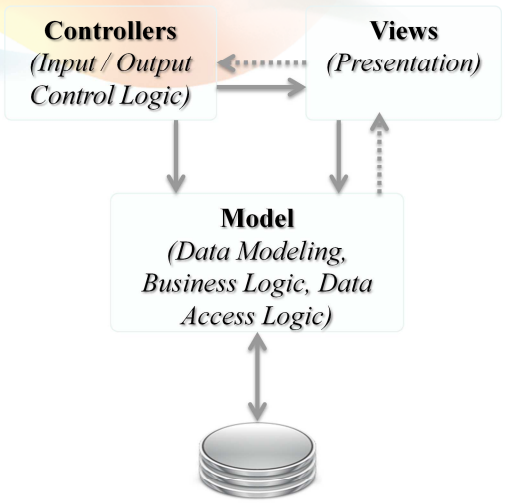
- View and Controller
  - Controller is responsible for creating or selecting view
- Model and Controller
  - Controller depends on model
  - If a change is made to the model then there might be required to make parallel changes in the Controller
- Model and View
  - View depends on Model
  - If a change is made to the model then there might be required to make parallel changes in the view

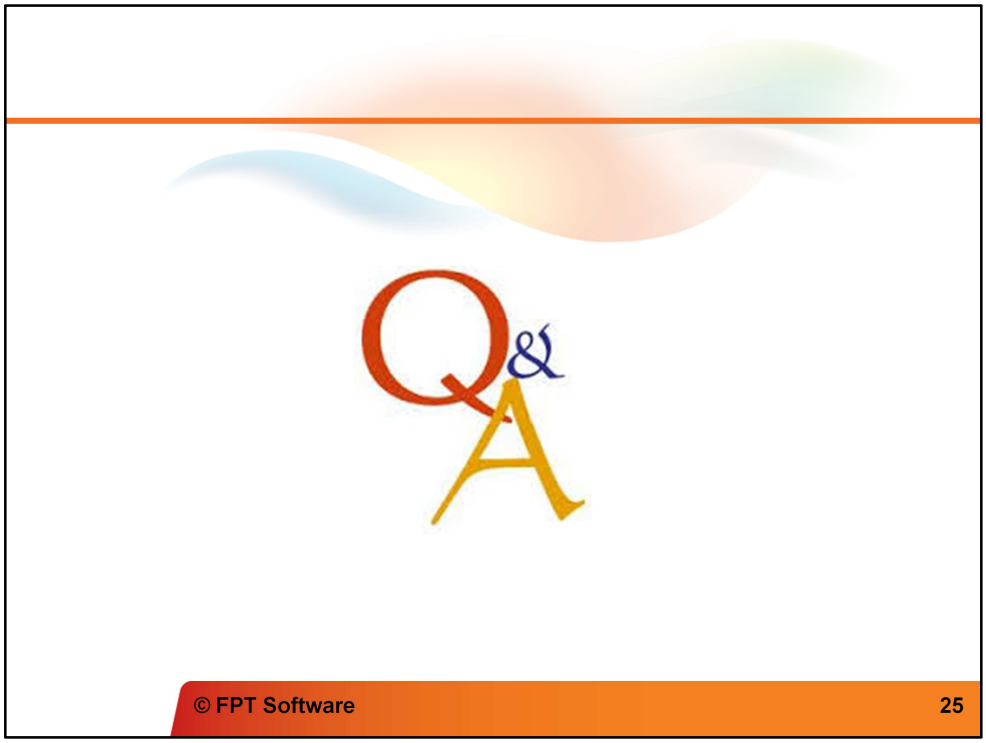
MVC – Based Frameworks

- .NET
  - ASP.NET MVC, MonoRail
- Java
  - JavaServer Faces (JSF), Struts, Spring Web MVC, Tapestry, JBoss Seam, Swing
- PHP
  - CakePHP, Symfony, Zend, Joomla, Yii, Mojavi
- Python
  - Django, Zope Application Server, TurboGears
- Ruby on Rails

MVC & Multi-Tier Architecture

- MVC does not replace the multi-tier architecture
  - Both are usually used together
- Typical multi-tier architecture can use MVC
  - To separate logic, data and presentation




Q&A