Lecture # 2

Web Technologies

Credit Hours: 3(2, 1)

Course Instructor: MUSTAFA ALI BAMBOAT

“Categories of Web Applications and Web Architectures”

Introduction

- Overview of web applications and architectures

Web applications use different structures to work efficiently and meet various needs. These structures, or architectures, help decide how the application is built, how it performs, and how well it can grow or handle complex tasks.

- Importance of understanding different categories and architectural patterns

Understanding different categories and architectural patterns is crucial for designing effective, scalable, and user-friendly web applications tailored to specific needs and constraints.

Categories of Web Applications

Document-Centric

Social Web

Semantic Web

Document-Centric Web Applications

- Focus on creating, managing, and sharing documents
- Examples: Google Docs, Microsoft Office Online, Dropbox Paper

Features of Document-Centric Apps

Real-time collaboration

Version control

Document sharing

Editing tools

Social Web Applications

- Platforms for user interaction and content sharing
- Examples: Facebook, Twitter, Instagram, LinkedIn

Features of Social Web Apps

User profiles

News feeds

Social interactions (likes, comments, shares)

Notifications

Semantic Web Applications

Uses data with semantic meaning to enhance user experience

Examples of semantic web applications include:

- **Wolfram Alpha**: Provides answers to queries based on structured data and computational algorithms.
- **Google Knowledge Graph**: Enhances search results with contextual information about entities.
- **Schema.org**: Facilitates the use of structured data to improve search engine understanding of web content.

Features of Semantic Web Apps

ENHANCED SEARCH CAPABILITIES

DATA INTEGRATION

CONTEXT-AWARE INFORMATION RETRIEVAL

IMPROVED DATA RELATIONSHIPS

Web Architectures Overview

- Introduction to architectural patterns:
  - Layered Architecture
  - One-Tier
  - Two-Tier
  - Three-Tier
  - N-Layered Architecture

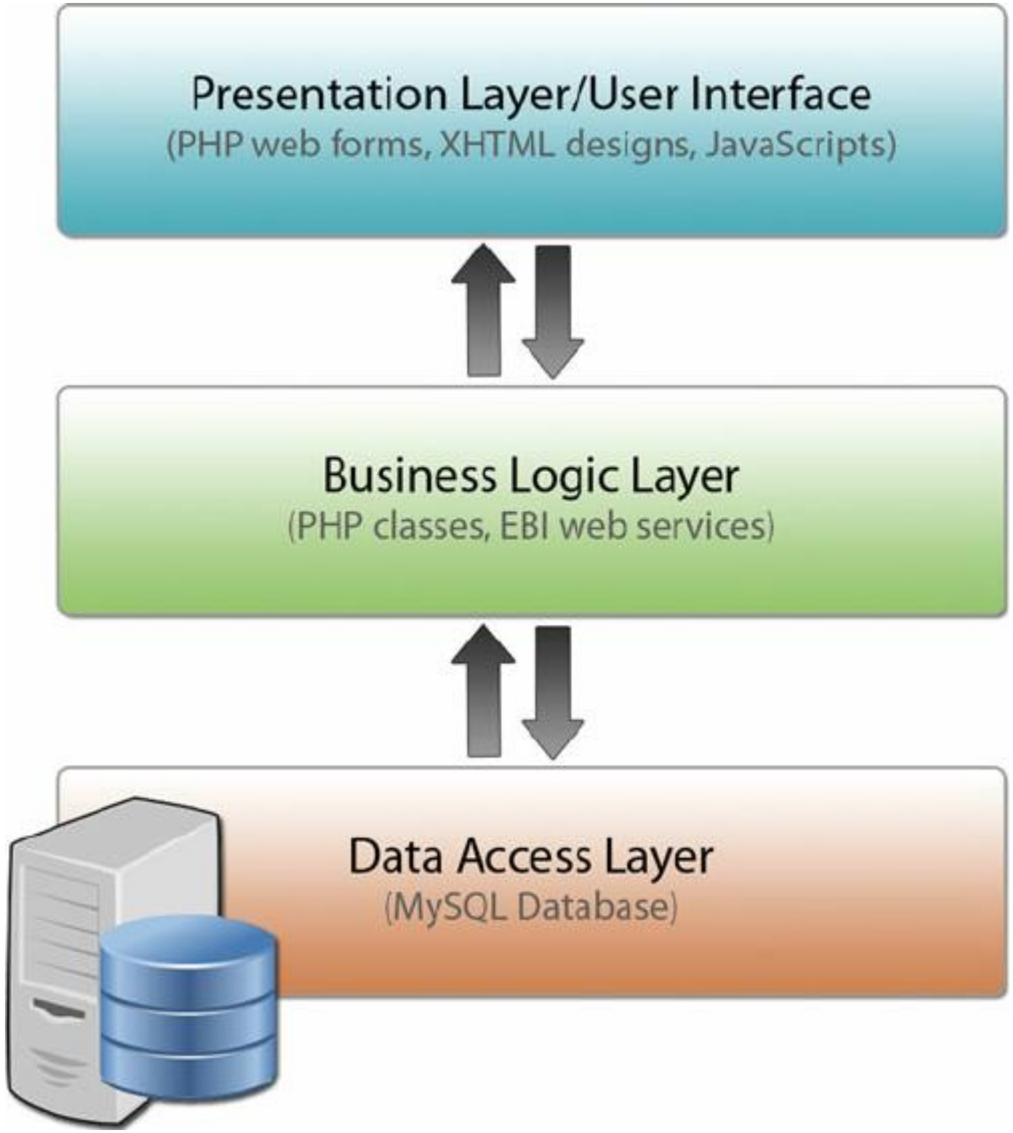
Layered Architecture

Structure: Presentation Layer, Business Logic Layer, Data Layer

Benefits: Separation of concerns, modularity, scalability

- **Separation** of concerns involves dividing a system into distinct sections, each handling a specific aspect of functionality;
- **modularity** allows for independent development and maintenance of these sections
- **Scalability** ensures that the system can handle increased loads or complexity effectively.

Diagram of Layered Architecture



Presentation Layer/User Interface

(PHP web forms, XHTML designs, JavaScripts)

Business Logic Layer

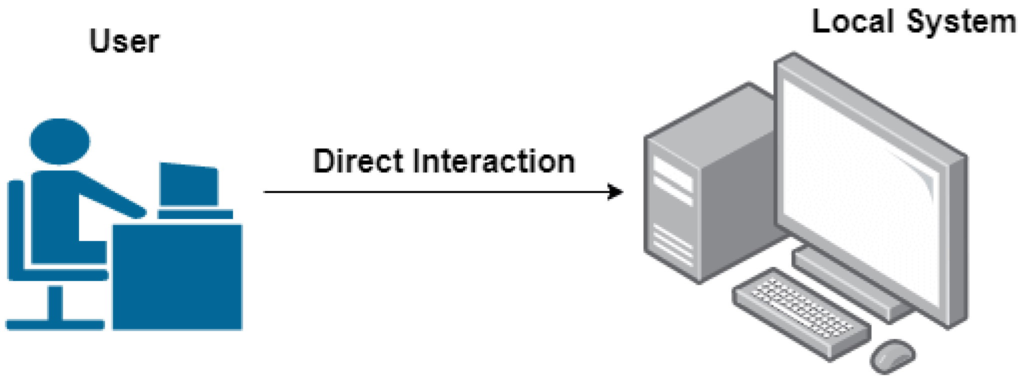
(PHP classes, EBI web services)

Data Access Layer

(MySQL Database)

One-Tier Architecture

- Also known as Single-Tier
- All components integrated into a single system
- Examples: Desktop applications or simple applications

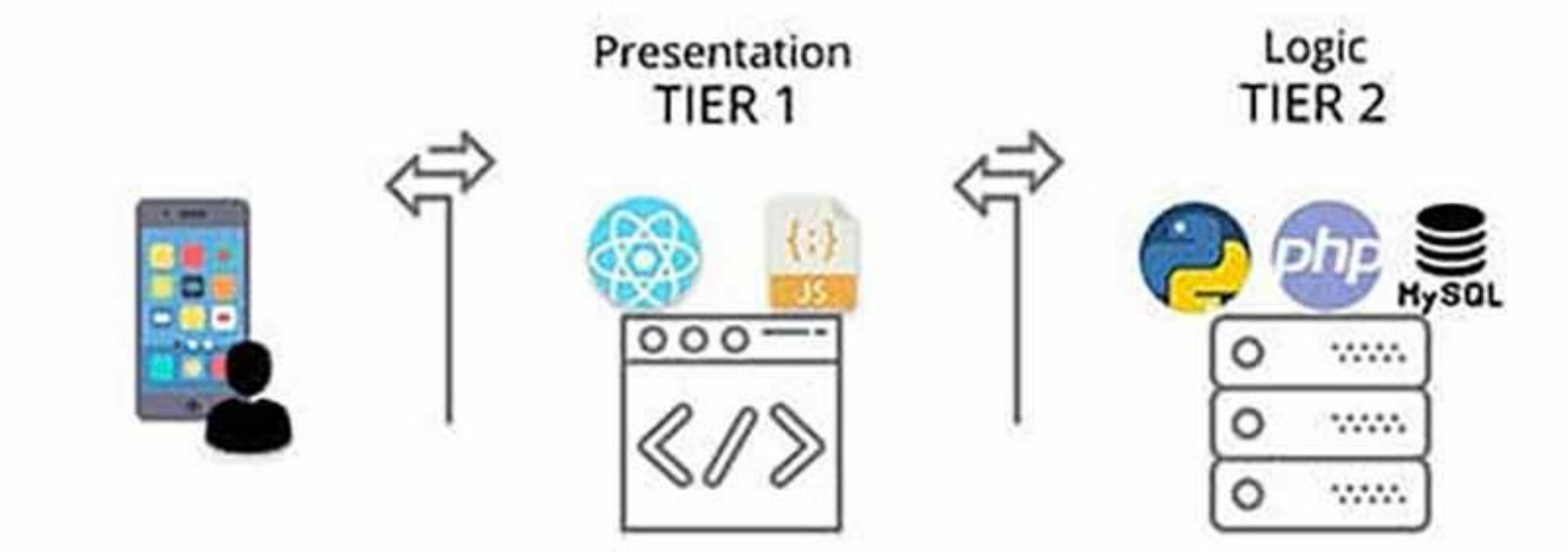


Two-Tier(C-S) Architecture

- Structure: Client-Server
- Client handles presentation; server manages data
- Examples: Basic client-server applications, such as online forms

Diagram of Two-Tier Architecture

- Visual representation of Client and Server layers

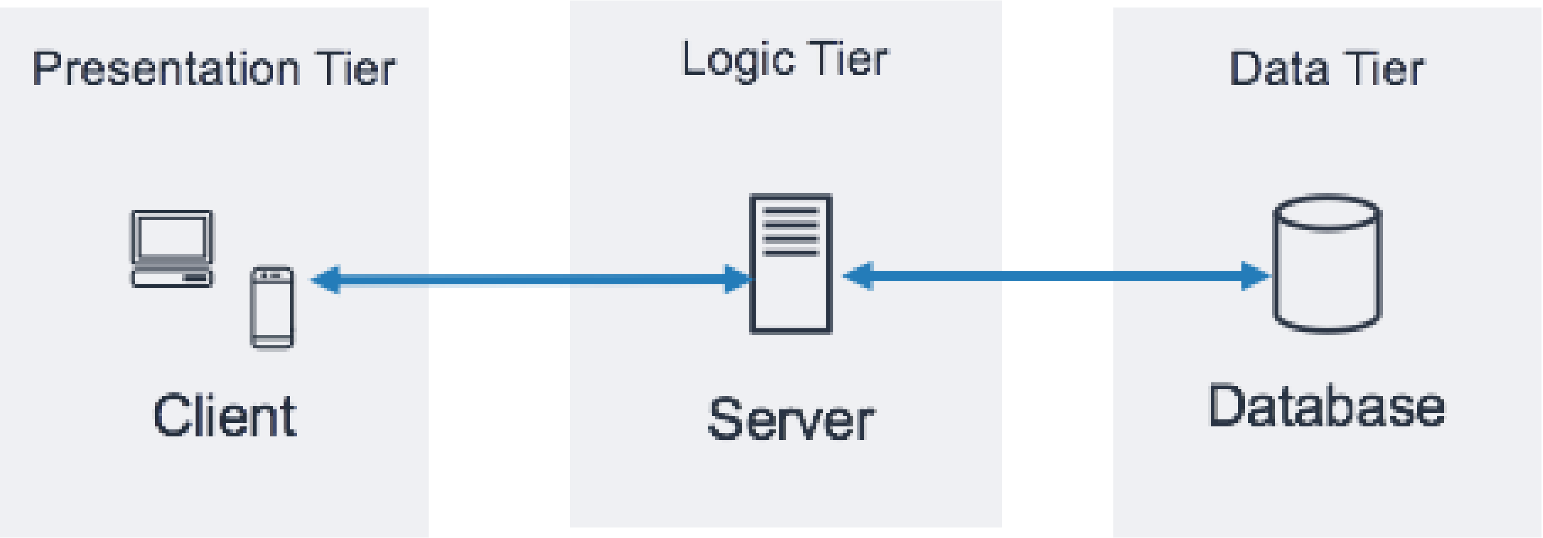


Three-Tier Architecture

- Structure: Presentation Tier, Application Tier, Data Tier
- Benefits: Improved scalability, separation of concerns, easier maintenance

Diagram of Three-Tier Architecture

- Visual representation of Presentation, Application, and Data Tiers

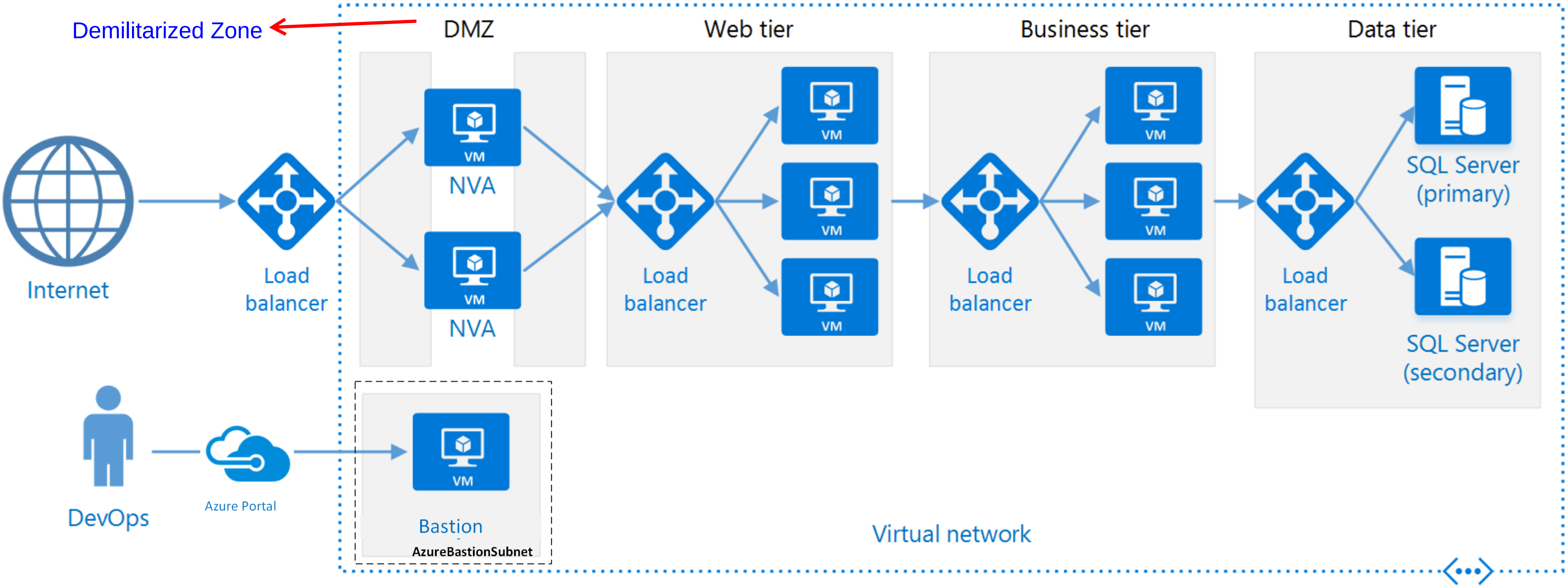


N-Layered Architecture

- Extension of layered architecture with multiple layers:
  - Presentation
  - Application
  - Domain
  - Data Access
- Benefits: Flexibility, modularity, scalability, maintainability

Diagram of N-Layered Architecture

- Visual representation of multiple layers and their interactions


Demilitarized Zone
DMZ
Web tier
Business tier
Data tier
Internet
Load balancer
VM
NVA
VM
NVA
Load balancer
VM
VM
VM
Load balancer
VM
VM
VM
Load balancer
SQL Server (primary)
SQL Server (secondary)
DevOps
Azure Portal
VM
Bastion
AzureBastionSubnet
Virtual network