

Computer Systems Architecture

Adapted from material provided by Sang Shin at www.javapassion.com

Objectives

- **To introduce computer system architecture concepts.**
- **To discuss logical and physical tiers**
- **To look at a typical layered architecture for a single tier.**
- **To present a typical N-Tier architecture that uses Java technology.**

Introduction to Computer Systems Architecture 1

- **The structural arrangement of the major components of both the hardware and software from which a computer system is composed defines the architecture of a system.**
- **Since networking is often a key aspect of most modern computer systems the functionality provided is often distributed across multiple computer systems.**

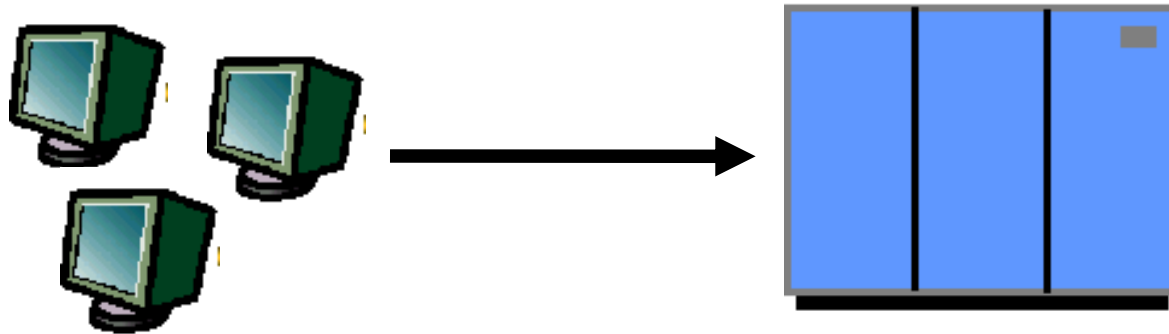
Introduction to Computer Systems Architecture 2

- **The layers of software present on a specific system to provide services to software components are often referred to as a tier.**
- **Complicated computer systems often employ multiple logically separated and/or physically separated tiers.**
- **Lower level layers in a tier provide services to the higher level layers in a tier.**

Terminology

- **Presentation Logic**
 - Functionality which deals with the display of information.
- **Business Logic**
 - Functionality which deals with the processing of information, the execution of commands and the coordination of application execution
- **Data Access Logic**
 - Functionality that makes it easy to access the data held in a data store such as a database.

Single Tier (Mainframe-based)



- **Dumb terminals are directly connected to a mainframe**
- **A centralized model as opposed to a distributed model.**
- **Presentation logic, business logic, and data access logic are intertwined in one monolithic mainframe application**

Single Tier (Workstation Based)

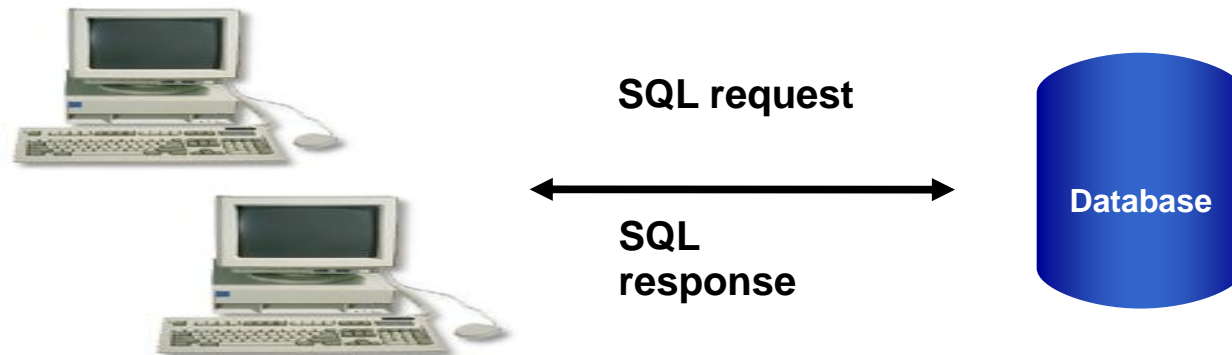


- **The user employs a single workstation e.g. PC, Mac and a single application e.g. Microsoft Word, Photoshop.**
- **Not a distributed architecture since the application does not employ network connectivity.**
- **Presentation logic, business logic, and data access logic are intertwined in a single 'Fat Client' application.**

Single-Tier: Pros & Cons

- **Pros:**
 - **Data consistency is easy to achieve**
- **Cons:**
 - **Functionality (presentation logic, business logic , data access logic) intertwined**
 - **Difficult to apply updates**

Two-Tier



- **Fat client software applications talking to a database server**
 - SQL queries sent, raw data returned
- **Presentation Logic ,Business logic and Data Access Logic intertwined in the fat client applications**

Two-Tier: Pros and cons

- **Pros:**

- Shared Database between clients

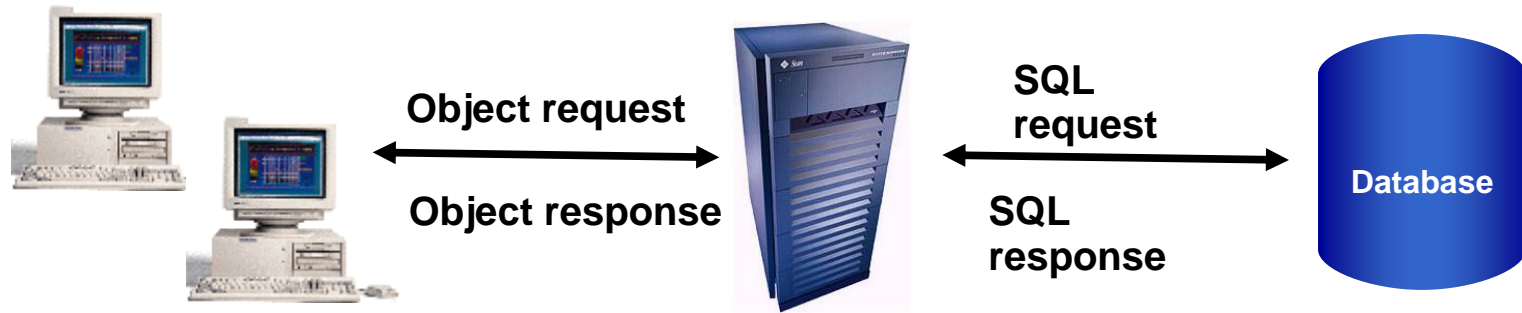
- **Cons:**

- Presentation logic, business logic and data access logic are intertwined (at client side), difficult for updates to be applied
- Data Model is “tightly coupled” to every client: If the Database Schema changes, all clients break
- Database connection for every client, thus difficult to scale (Database connections use a lot of system resources)
- Raw data transferred to client for processing causes high network traffic

Three-Tier

- **Many different technologies can be used in a three tier computer system**
- **Three Tier examples include:**
 - **Three tier: Remote Object based**
 - **Three tier: Web/Application server based**

Three-Tier (Remote Object based)



- **Business logic and data access logic are encapsulated in objects hosted on a middle tier server.**
- **OMG CORBA, Sun Microsystems RMI and Microsoft .Net remoting are the typical technologies used**

Three-tier (Remote Object based): Pros & Cons

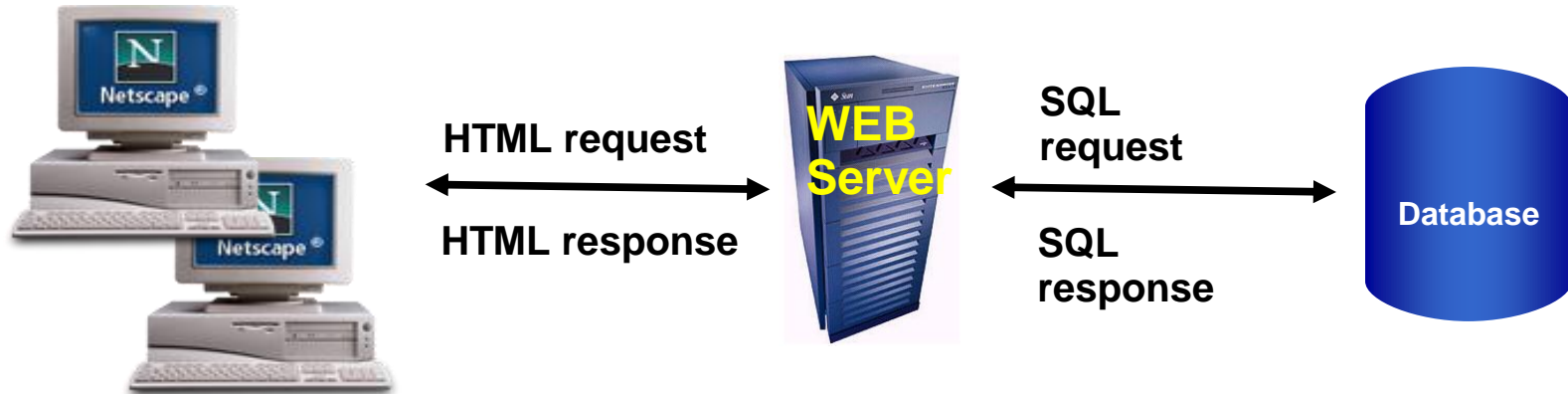
- **Pro:**

- Middle Tier server code can be shared between clients

- **Cons:**

- Complexity can be a problem in the middle-tier

Three-Tier (Web/Application Server)



- **Browser (A Thin Client) only handles presentation of web pages provided by a server. The browser may also execute some presentation logic code embedded in the pages it presents.**
- **HTTP protocol used for browser/ server interaction**
- **Business logic and data access logic are handled by “dynamic content generation” technologies (CGI, Servlet/JSP, ASP.NET)**

Three-tier (Web/Application Server based): Pros & Cons

- **Pros:**

- **Ubiquitous client types**
- **Zero client management**
- **Supports various client hardware platforms**
 - cell-phones and other hand held devices.

- **Cons:**

- **Complexity can be a problem in the middle-tier since it hosts the presentation logic, business logic and data access logic. Moving to n-tier can overcome this.**

Trends

- **The industry is moving from:**
 - **Single-tier and two-tier to multi-tier architecture**
 - **Procedure based application development to Object-Oriented application development using software components**
 - **Fat client applications to thin client applications that use HTML to present their user interface.**

Single Tier / Multi Tier Comparison

Single tier

- No separation between presentation logic, business logic and database access logic
- Hard to maintain

Multi-tier

- Separation between presentation logic, business logic and database access logic
- More flexible when making changes, i.e. presentation logic can change without affecting the other tiers

Monolithic / Object-based Software Comparision

Monolithic

- 1 Binary file
- Recompiled, relinked and redeployed every time there is a change

Object-based

- Pluggable Components
- Reusable Components
- Promotes better design
- Easier to update
- Implementation can be separated from interface, only the interface is published

Layers

- **A common approach to software architecture involves arranging software into a set of layers.**
- **Lower level layers in a layer stack provide services and functionality to higher level layers.**
- **Examples of the services that might be provided by a layer are the rendering of graphics objects on a display and the passing of information, without any corruption, between the software tiers present on different computer systems.**

Layering Examples

- **The seven layer International Standards Organisation (ISO) Open Systems Interconnect (OSI) model.**
- **The TCP/IP five layer model often called the Internet reference model.**
- **The Direct X and OpenGL computer graphics programming interfaces.**

TCP/IP 5-Layer Reference Model

- Also called 'Internet Reference Model'

Application Layer : contains functionality used by users of the system or which hosts or contains components that provide functionality
e.g Web Browser, email client , FAT client in two tier system

Transport Layer : Contains the functionality to allow data to be transferred reliably between tiers

Internet Layer : Contains the functionality to allow data to be routed between tiers

Network Access Layer : Provides the functionality to allow a tier to pass information over its physical network hardware.

Physical Layer : allows the data bits and bytes to be passed across the physical connections between tiers.

Example N-tier Architecture (Using Java Technology)

