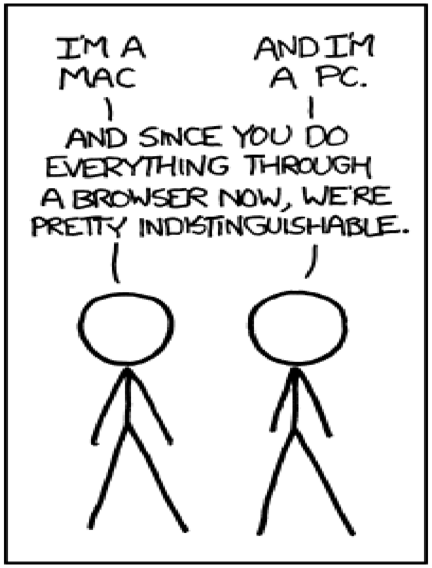
MIS6070

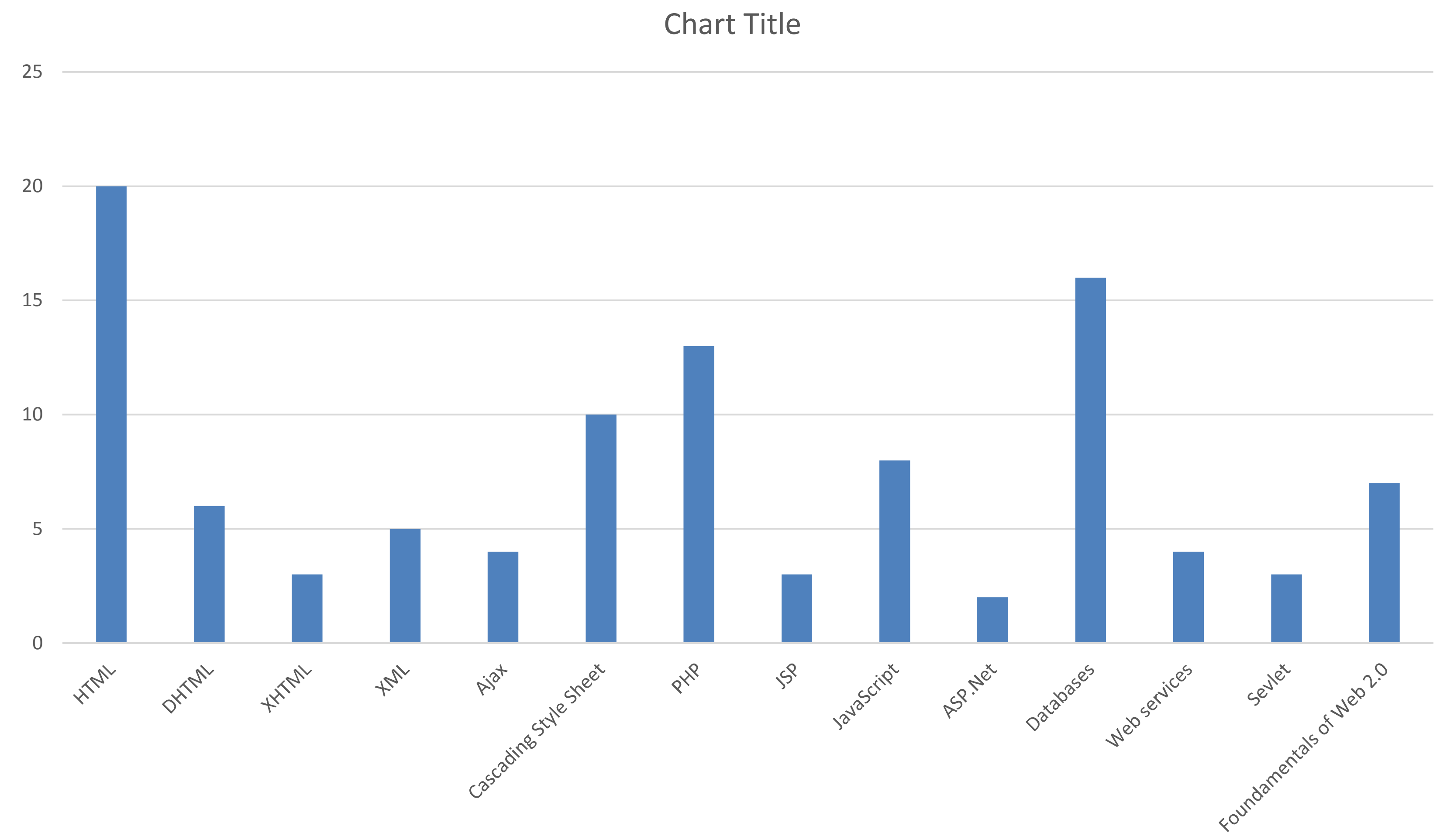
WEB BASED INFORMATION SYSTEMS


I'M A MAC
AND I'M A PC.
AND SINCE YOU DO EVERYTHING THROUGH A BROWSER NOW, WE'RE PRETTY INDISTINGUISHABLE.

HTML CSS JavaScript AJAX HTTP

Lesson 1

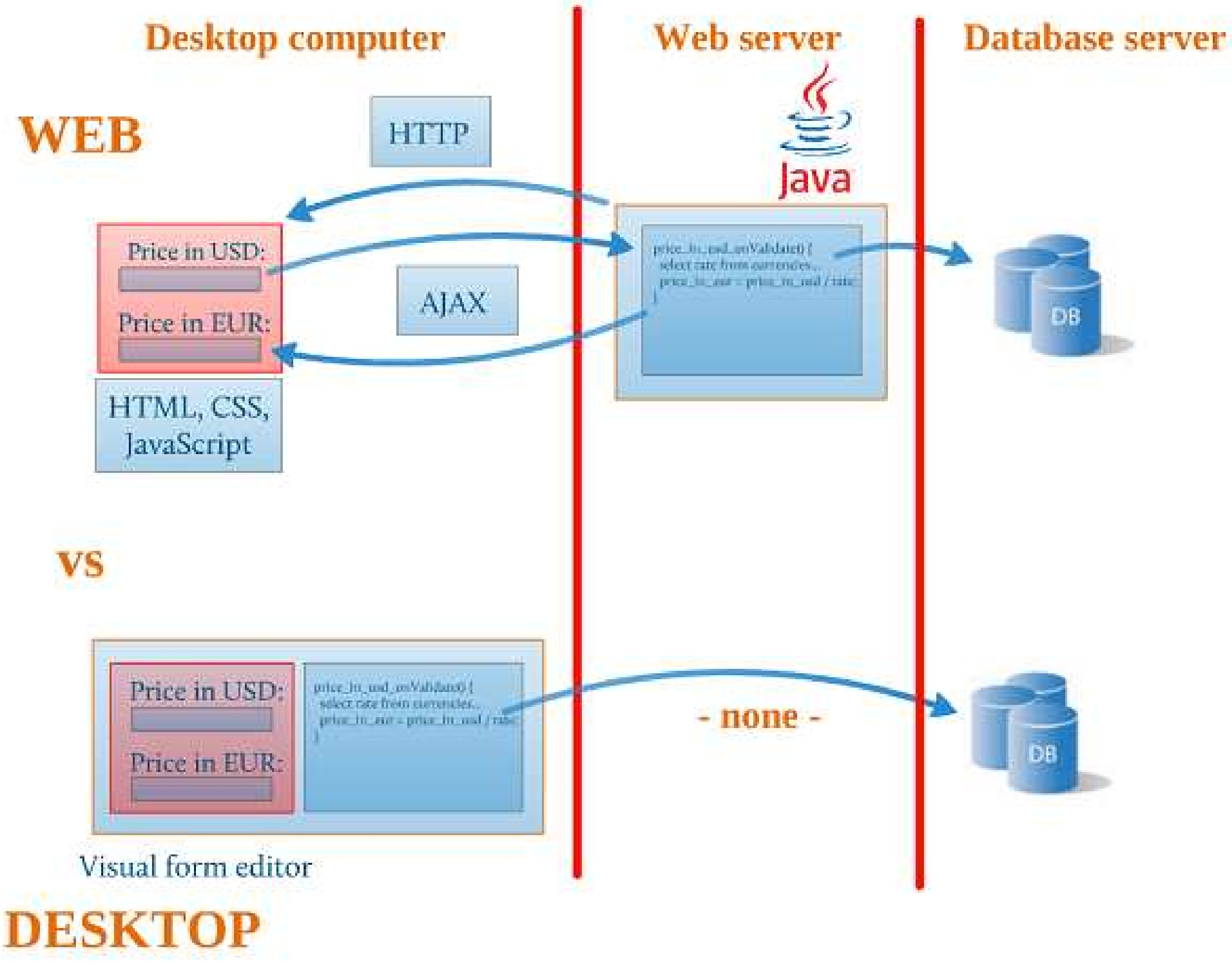
Web or Internet Applications



Web or Internet Applications

- A Internet application is a software package that can be accessed through the web browser.
  - It is half application and half web site
- The software and database reside on a central server rather than being installed on the desktop system and is accessed over a network.
  - Any organization can harness web technology to save time, save money, improve communication.

Internet Applications



```
price_in_usd_onValidate() {
  select rate from currencies...
  price_in_eur = price_in_usd / rate
}
```

```
price_in_usd_onValidate() {
  select rate from currencies...
  price_in_eur = price_in_usd / rate
}
```

Web Based Applications

- Most Internet applications are determined by organization’s business requirements.
- It could be a customized web application
  - Such as an e-commerce system.
- Such an application enables the employees to interact with the organization’s data in a fluid and highly responsive manner.
- This gives the organization an edge over its competitors by allowing the business to consolidate and streamline its systems and processes and reduce costs.

Web Based Applications

- Internet or Web based applications have evolved significantly over recent years
  - Improvements in information systems security
  - Improvement in technology
- There are plenty of scenarios where traditional software based applications and systems could be improved by migrating them to a web based application

Examples of Internet or Web Based Applications

- SMS messaging integration
  - M-banking
- Online Product Database
  - http://www.alibaba.com
- Online Data Collection, Analysis and Reporting
  - Online questionnaire
- Online Customer Support Software
- Secure Areas to allow collaboration with Business Partners
- Online Quote Software

Why Web based applications?

- Web based applications are the ultimate way to take advantage of today's technology to enhance organizations productivity and efficiency.
- Web based application gives the opportunity to access business information from anywhere in the world at any time.
- It also facilitates the organization to save time and money and improve the interactivity with customers and partners.

Why Web based applications?

- It allow administration staff to work from any location and sales staff to access information remotely 24 hours a day, 7 days a week.
  - With a computer connected to the Internet, a web browser and the right user name and password one can access the systems from any location.
- Web-based applications are easy to use and can be implemented without interrupting existing work process.
- Delivering applications via the web eliminates the problems of versioning
- Harness web technology to improve data and information access.
  - This will save money and time while improving communication

Core benefits of web based applications

- **Cross platform compatibility.**
  - Most web based applications are far more compatible across platforms than traditional installed software.
  - Typically the minimum requirement would be a web browser of which there are many. (Internet Explorer, Firefox, Netscape, Safari).
  - These web browsers are available for a multitude of operating systems and so whether you use Windows, Linux or Mac OS you can still run the web application.

Core benefits of web based applications

- **More manageable**
  - Web based systems need only be installed on the server placing minimal requirements on the end user workstation.
  - This makes maintaining and updating the system much simpler as usually it can all be done on the server.
  - Any client updates can be deployed via the web server with relative ease.

Core benefits of web based applications

- **Highly deployable**
  - Due to the manageability and cross platform support deploying web applications to the end user is far easier.
  - They are also ideal where bandwidth is limited and the system and data is remote to the user.
  - You simply need to send the user a website address to log in to and provide them with internet access.
  - This has huge implications allowing you to widen access to your systems, streamline processes and improve relationships by providing more of your customers, suppliers and third parties with access to your systems.

Core benefits of web based applications

- **Secure live data**
  - Typically in larger more complex systems data is stored and moved around separate systems and data sources.
    - In web based systems these systems and processes can often be consolidated reducing the need to move data around.
  - Web based applications also provide an added layer of security by removing the need for the user to have access to the data and backend servers.

Core benefits of web based applications

- **Reduced costs**
  - Web based applications can dramatically lower costs due to reduced support and maintenance, lower requirements on the end user system and simplified architecture.
  - By further streamlining your business operations as a result of your web based application additional savings can often be found.

Core benefits of web based applications

- **Usage Accounting:**
  - Developers can find out how many times the web application is accessed, something which may not be possible when distributing standalone applications to customers. The vendor can also offer a per-usage fee structure.
- **Less Expensive Infrastructure:**
  - Since every desktop can be equipped with a free browser, the users only need inexpensive, low maintenance computers that run the browser software.

Core benefits of web based applications

- **Rapid Software Distribution:**
  - There is no need to distribute new software to any of the client browsers. Once programs are updated on the server, users can instantly see the changes the next time they load the page into their browser.
  - Troubleshooting is also improved since most issues can be investigated and resolved on the server.

Core benefits of web based applications

- **Low Network Usage:**
  - In a web environment, the client application (browser) doesn't access the database over the network to retrieve or update the data.
  - The browser only sends instructions to the server with as much information as necessary to execute the server program.
  - The server software is responsible for communicating with and retrieving any content from the database and composing the page to be sent back to the client; there is no need to exchange large amounts of data.
  - This is why web applications often run faster than applications based on fat clients.

Core benefits of web based applications

- **Platform Independence:**
  - Most web applications can be accessed through a variety of web browsers. Web browsers exist for virtually any operating system.
  - Since they receive only the HTML document it is irrelevant if the document is being viewed on Windows, Linux/UNIX, Mac, or any other system.
  - Additionally, certain web applications are accessible through wireless devices equipped with a browser and Internet connection.

Disadvantages of Web Based Applications

- Slower, as they run over the internet
- Interfaces not as sophisticated
- Can take longer to develop
  - as they are more complex,
  - have to work on different browsers, and different versions of browsers,
  - only writable by programming experts
- Security risks

Challenges Encountered Using Web Technologies

- **Limited Interactivity:**
  - Interactivity is limited since a web page cannot communicate with the server in real-time except when the page is submitted to the server by the user.
  - For example, if a program needs to populate a particular list box field, it needs to refresh the whole page when the user reaches a particular field on the screen.
- **Lack of Skills:**
  - The web is still foreign to many developers who have been busy developing mainframe, client/server, and standalone applications.
  - These developers are often unfamiliar with web technologies, web programming, and web scalability issues.

Challenges Encountered Using Web Technologies

- **Security Risks:**
  - Utilizing the web exposes the company's network to new security risks and programming vulnerabilities that can expose private data to the public.
- **Browser Compatibility:**
  - Unfortunately, the competition between web browsers has resulted in many differences in supported standards, thus creating confusion between users, and more importantly, compatibility issues with web applications.
  - It is recommended that you test your web application with several browser versions, on several operating systems.

Challenges Encountered Using Web Technologies

- **Lack of Web Development Tools Until Recently:**
  - The development of web applications is time-consuming due to the lack of professional web development tools.
  - The available tools are usually coding environments that require a solid programming foundation and experience.
  - Only recently have new tools become available that facilitate the creation of web applications with little programming experience.
  - These include Macromedia Dreamweaver MX, Microsoft Visual Studio.NET, and YesSoftware CodeCharge products.

Architecture of Web Applications

- In general, web applications use the client/server architecture (i.e., two-tier) where the browser acts as a thin client.
- The three-tier architecture is becoming more popular and even necessary when implementing enterprise systems that require better scalability.
- A multitier (N-tier) architecture is an expansion of the 3-tier architecture, in one of several different possible ways
  - Replication of the function of a tier
  - Specialization of function within a tier

System Architecture

- The architecture of a computer system is the high-level (most general) design on which the system is based

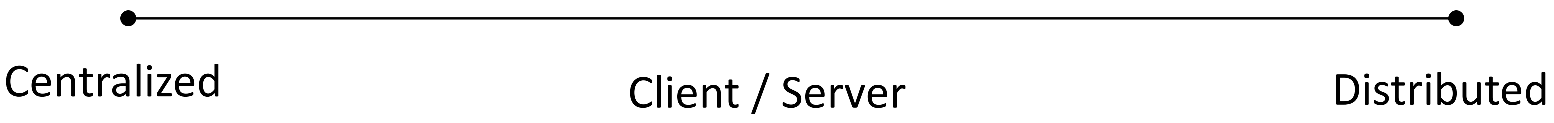
- Architectural features include:
  - Components
  - Collaborations (how components interact)
  - Connectors (how components communicate)

Client-Server Architecture

- Each component of a client-server system has the role of either client or server
  - Client: a component that makes requests *clients are active initiators of transactions*
  - Server: a component that satisfies requests *servers are passive and react to client requests*

Centralized / Distributed

- The client-server architecture can be thought of as a median between
  - **Centralized processing:** computation is performed on a central platform, which is accessed using “dumb” terminals
  - **Distributed processing:** computation is performed on platforms located with the user



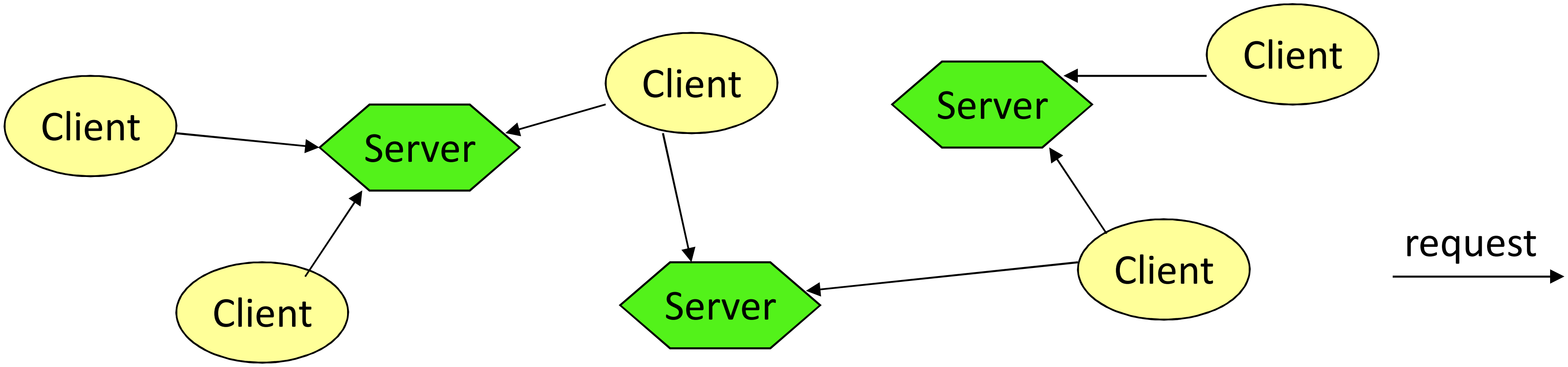
Centralized

Client / Server

Distributed

Client-Server Architecture

- The Web is a client-server system
- Web browsers act as clients, and make requests to web servers
- Web servers respond to requests with requested information and/or computation



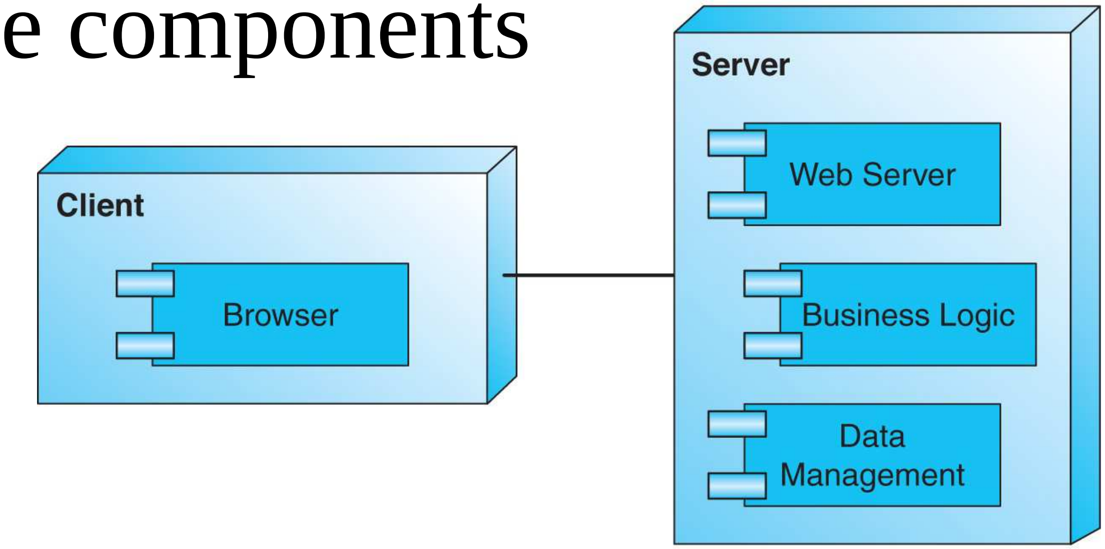
Tiered Web Architectures

- Web applications are usually implemented with 2-tier, 3-tier, or multitier (N-tier) architectures

- Each tier is a platform (client or server) with a unique responsibility

2-Tier C-S Architecture

- Tier 1: Client platform, hosting a web browser
- Tier 2: Server platform, hosting all server software components

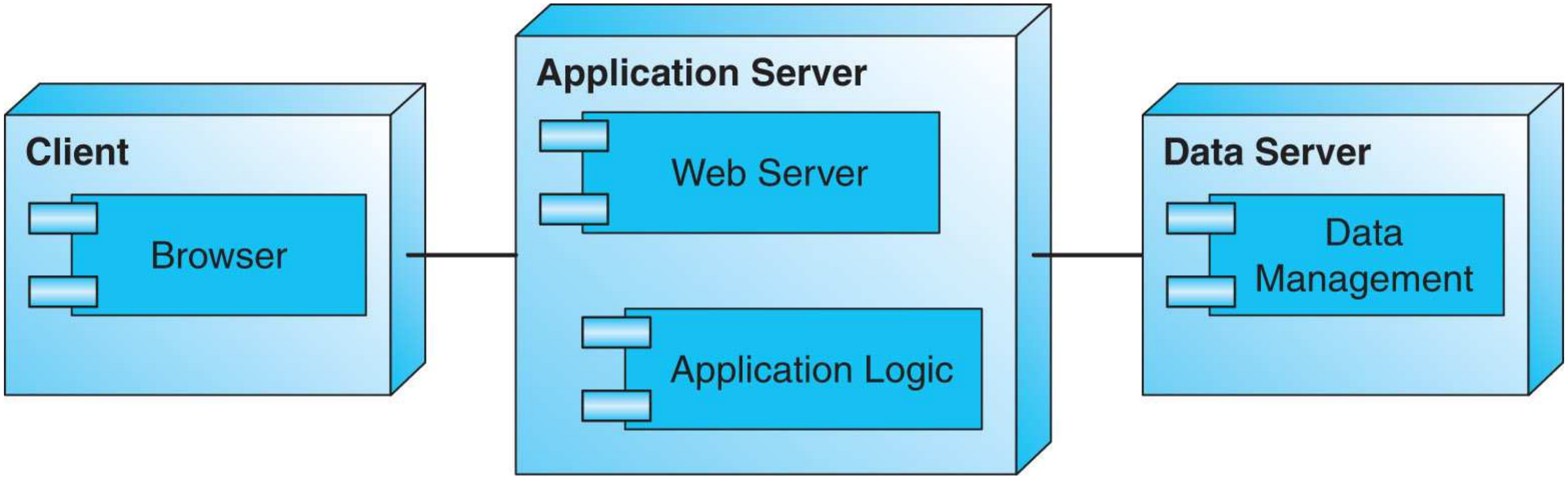


2-Tier Characteristics

- Advantage:
  - Inexpensive (single platform)
- Disadvantages
  - Interdependency (coupling) of components
  - No redundancy
  - Limited scalability
- Typical application
  - 10-100 users
  - Small company or organization, e.g., law office, medical practice, local non-profit

3-Tier C-S Architecture

- Tier 3 takes over part of the server function from tier 2, typically data management



3-Tier Characteristics

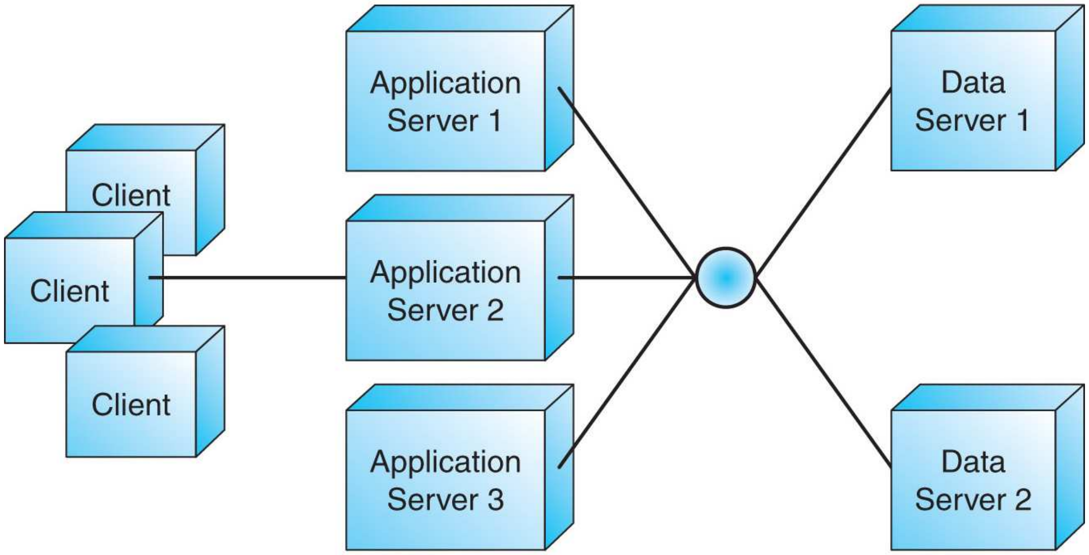
- Advantages
  - Improved performance, from specialized hardware
  - Decreased coupling of software components
  - Improved scalability
- Disadvantages
  - No redundancy
- Typical Application
  - 100-1000 users
  - Small business or regional organization, e.g., specialty retailer, small college

Multitier C-S Architecture

- A multitier (N-tier) architecture is an expansion of the 3-tier architecture, in one of several different possible ways
  - Replication of the function of a tier
  - Specialization of function within a tier
  - Portal services, focusing on handling incoming web traffic

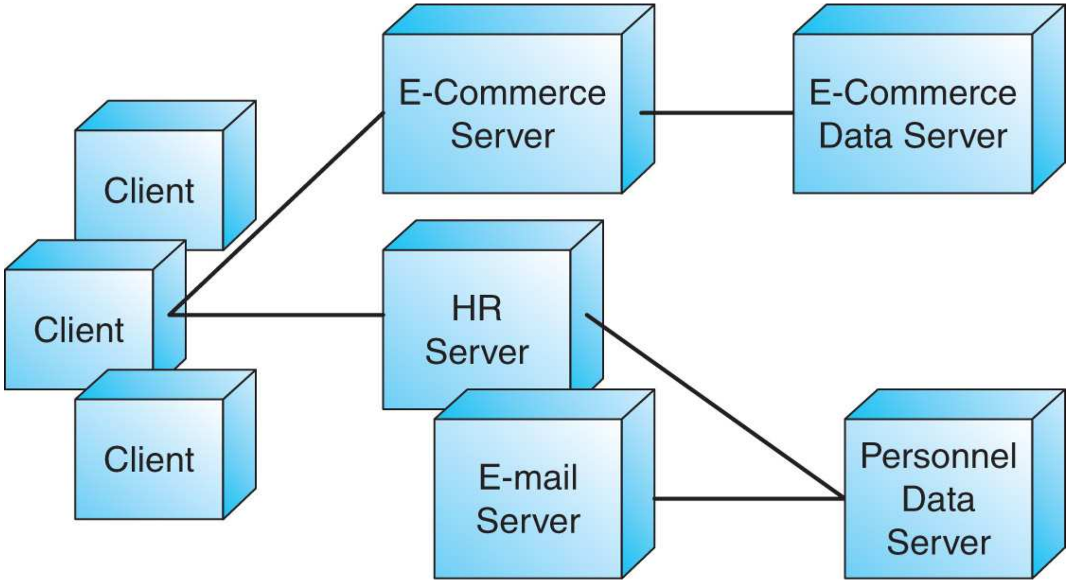
Replication

- Application and data servers are replicated
- Servers share the total workload



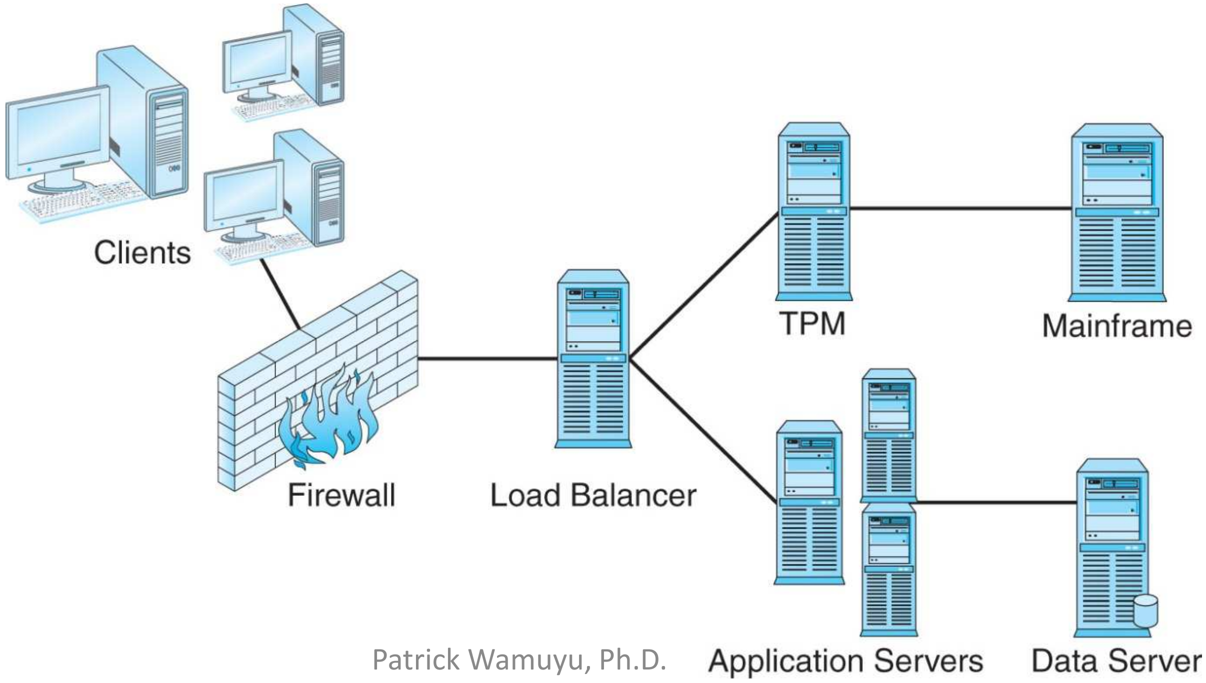
Specialization

- Servers are specialized
- Each server handles a designated part of the workload, by function



Portal Services

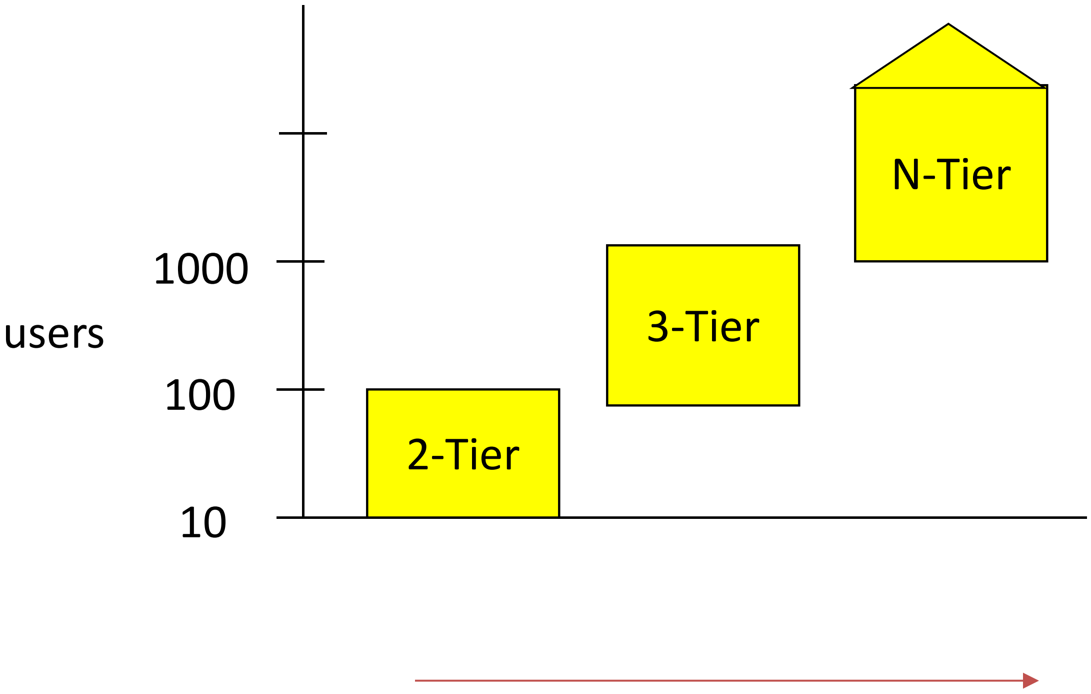
- Portal servers handle incoming traffic, reducing application server load
  - e.g., firewall, load balancer, transaction processing manager



Multi-Tier Characteristics

- Advantages
  - Decoupling of software components
  - Flexibility to add/remove platforms in response to load
  - Scalability
  - Redundancy
- Disadvantages
  - Higher costs (maintenance, design, electrical load, cooling)
- Typical Application
  - 1000+ users
  - Large business or organization

Characteristics Summary

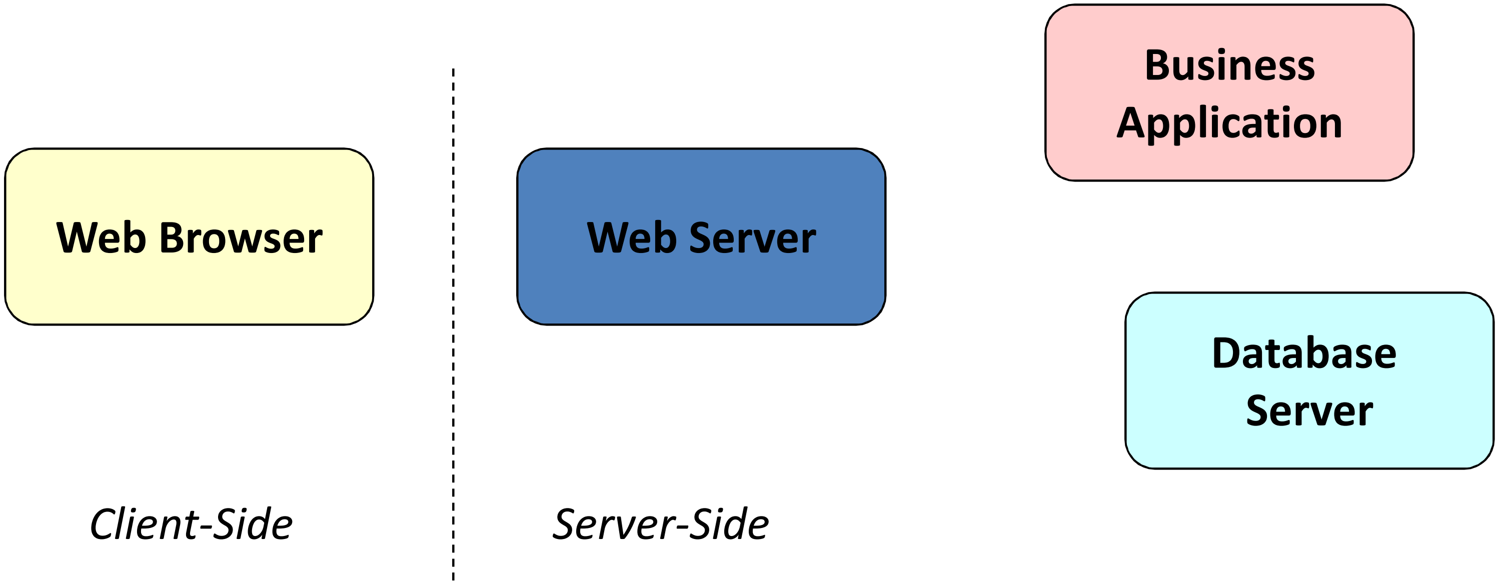


capacity
scalability
redundancy
cost

- large e-commerce, business, or organization
- small e-commerce, regional business or organization
- local business or organization

Internet Applications Components

- Four important components of a web application:



Web Browser

Web Server

Business Application

Database Server

Client-Side

Server-Side

Internet Applications Components

- **Web Browser:** presents the user interface
- **Web Server:** processes HTTP requests
- **Business Application:** processes requests at the application level by providing a service
- **Database Server:** maintains the database by processing query and update requests from the application

Web Browser

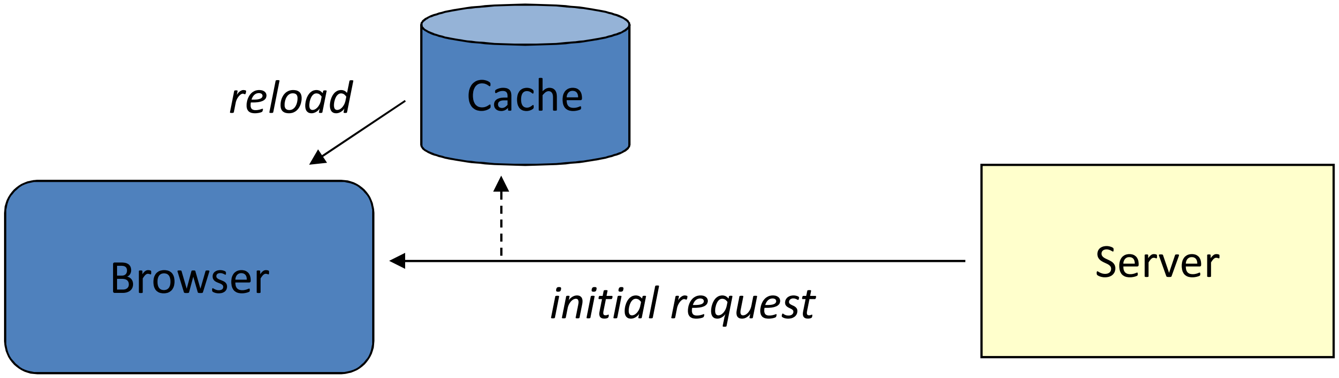
- A Web browser, also called a browser, is a program that interprets and displays Web pages and enables you to view and interact with a Web page
  - Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, and Apple Safari
- A Uniform Resource Locator (URL) is the address of a document or other file accessible on the Internet
  - http://www.usiu.ac.ke/
  - A hyperlink, also called a link, is an element used to connect one Web page to another Web page that's located on the same server or used to link Web pages located on a different Web server located anywhere in the world

The Six Web Browser Responsibilities

- User Interface Presentation
  - Parse HTML and CSS code
    - handle errors
  - Format and present a graphical display
  - Handle user interactions
    - scroll, mouse movement, click, etc.
- Client-Server Communication (HTTP)
  - Format HTTP request
  - Handle HTTP response
    - including redirects, errors, etc.
  - Request subordinate items
    - images, style sheets, etc.

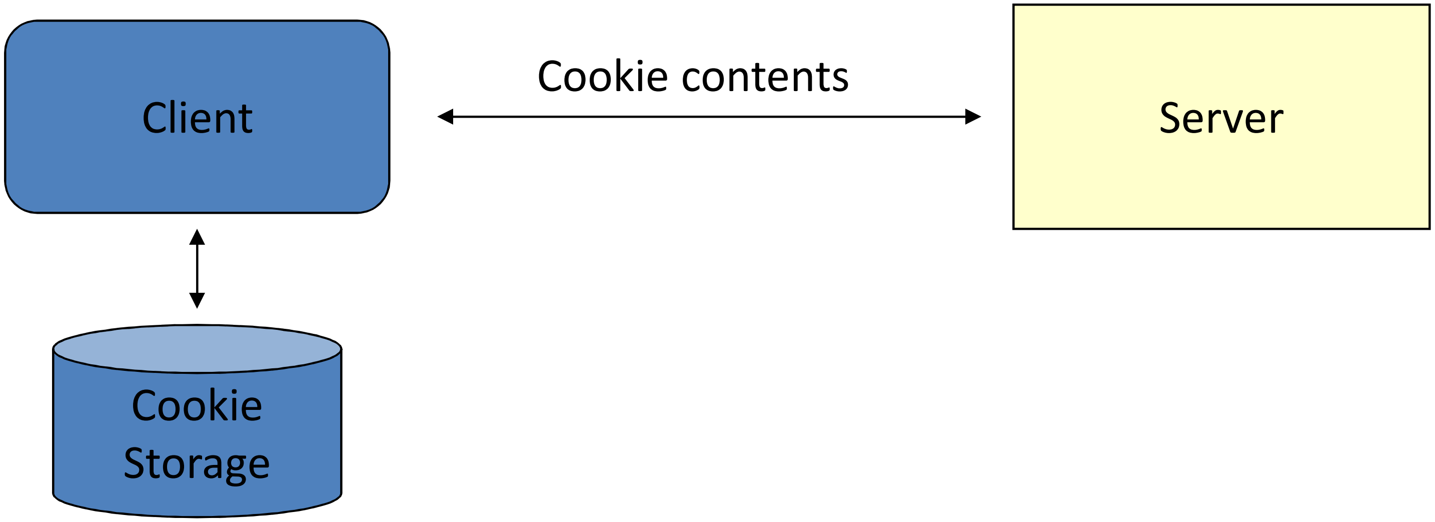
The Six Web Browser Responsibilities

- Cache Control
  - Cache is a local memory for recently accessed web pages, images, etc.
  - Cache enables pages to be reloaded quickly, without another HTTP transaction



The Six Web Browser Responsibilities

- Cookie Management
  - An HTTP cookie is a small file that is
    - provided by the server as an HTTP response header
    - stored by the client
    - returned to the server as an HTTP request header



Cookies and Sessions

- HTTP Session: a series of HTTP transactions between a client and a server
- Cookies allow servers to identify sessions

- Typical usage
  - initial transaction: server creates session (server-side) and stores session-id in a cookie (client-side)
  - subsequent transactions: client sends session-id with each request, allowing server to locate session data

The Six Web Browser Responsibilities

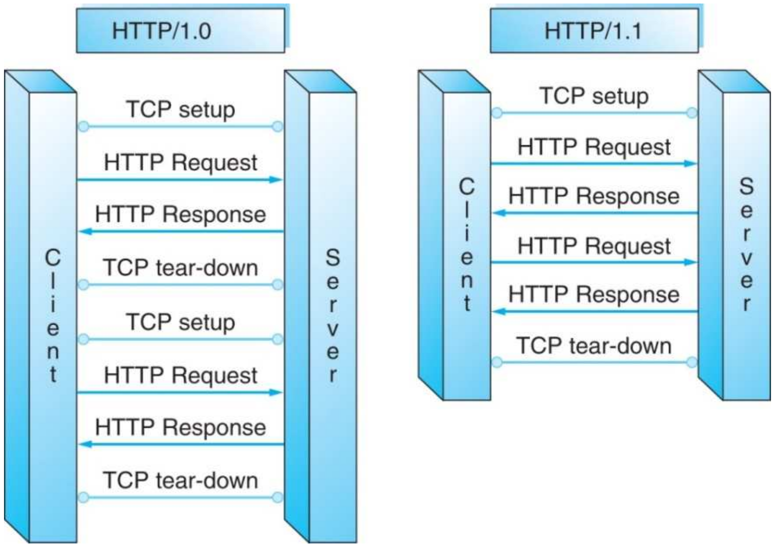
- Handling Embedded Objects
  - Objects referenced by an HTML document (images, scripts, style sheets, etc.) must be loaded separately by the browser
    - A plug-in or helper application may be used
  - Helper application: a program that handles content in a separate window
    - e.g., audio player, movie player
  - Plug-in: a program that handles content within the browser
    - e.g., animation viewer or virtual machine for applets

The Six Web Browser Responsibilities

- Script Interpretation
  - Most browsers interpret JavaScript and its variants Scripting languages are powerful, so interpreters are necessarily complex
  - Script interpreters are not entirely standardized across browsers, so script programmers must test scripts on many browser versions
    - The “write once, test many” principle in action

Web Server Responsibilities

- Connection Management
  - With HTTP/1.1, servers are responsible for setting up and tearing down TCP connections over a series of requests
    - improves efficiency



Web Server Responsibilities

- HTTP Request Handling
  - Servers must correctly interpret HTTP request headers and respond appropriately
  - Static content includes fixed files that are returned without any further computation
  - Dynamic content is generated on request by an application component that is invoked by the server

Database Server Responsibilities

- Relational Database Management System (RDBMS)
- Maintains data storage for an application
  - processes queries and updates

    Provides a standard interface for application programs
  - Open DataBase Connectivity (ODBC)
  - Java DataBase Connectivity (JDBC)

    Supports standard query language for data query and manipulation
  - Structured Query Language (SQL)

Web Application Infrastructure

- There are three categories of infrastructure components that can help to provide the basis for web application development:
  - Platforms, (LAMP, Microsoft/.NET, Java EE)
  - Frameworks (.NET Framework, Ruby on Rails, Struts)
  - Content management systems (Joomla, Vignette)

Web Application Infrastructure

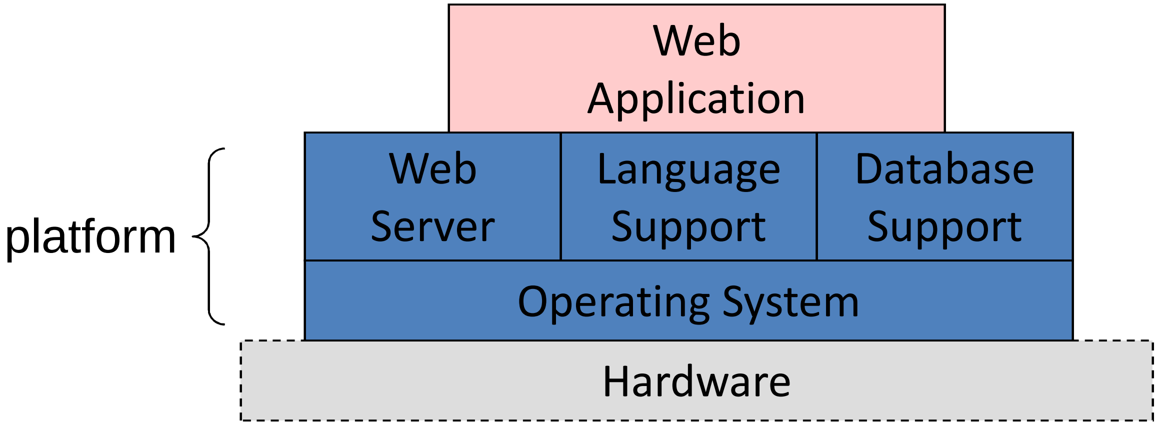
- A web application platform is a host environment for web application deployment.
  - This include an OS, web server, programming languages and compilers, interpreters and other language run-time support systems, database drivers, and Database Management Systems
  - Not all of these components are essential to every platform

Web Application Platform

- The web application platform serves as the host environment for web application deployment.
- It contains a variety of components which include an operating system, a web server, programming languages and compilers, interpreters and other run-time support systems, database drivers, and database management systems.

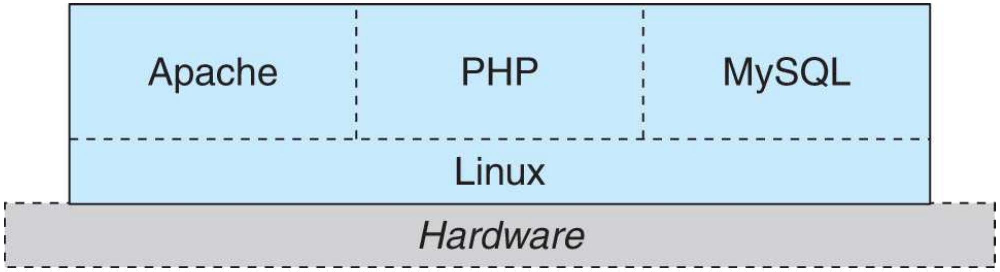
Web Application Platform

- A web application **platform** is the host environment for application development and operation
- The platform includes
  - Operating system, web server, language support, database support



LAMP

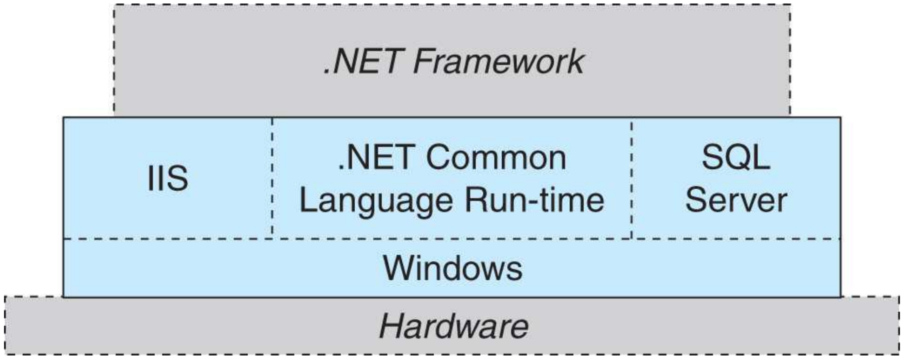
- Linux, Apache, MySQL, PHP/Perl/Python



- The LAMP platform appeared in mid 1990's and has become very popular
- LAMP is open-source free software, which is one reason for its popularity

Microsoft / .NET

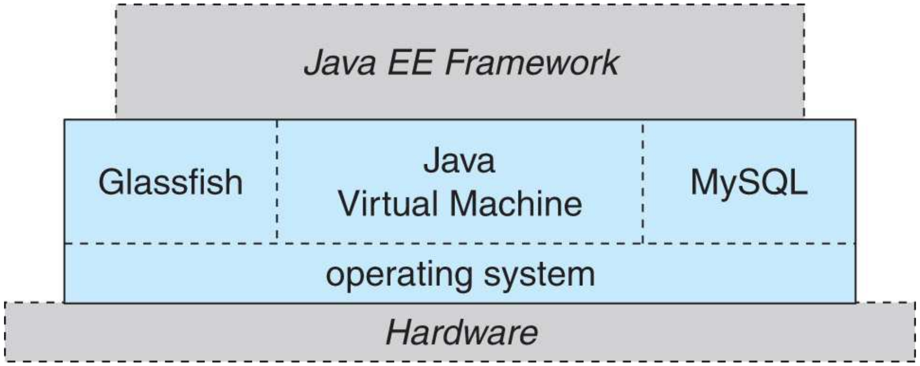
- All Microsoft products (licensed)



- .NET supports multiple languages
- Runs primarily on Windows Server O/S

Sun Microsystems/Java EE

- Supports Java language development



- Supported by multiple operating systems
- Proprietary, free license

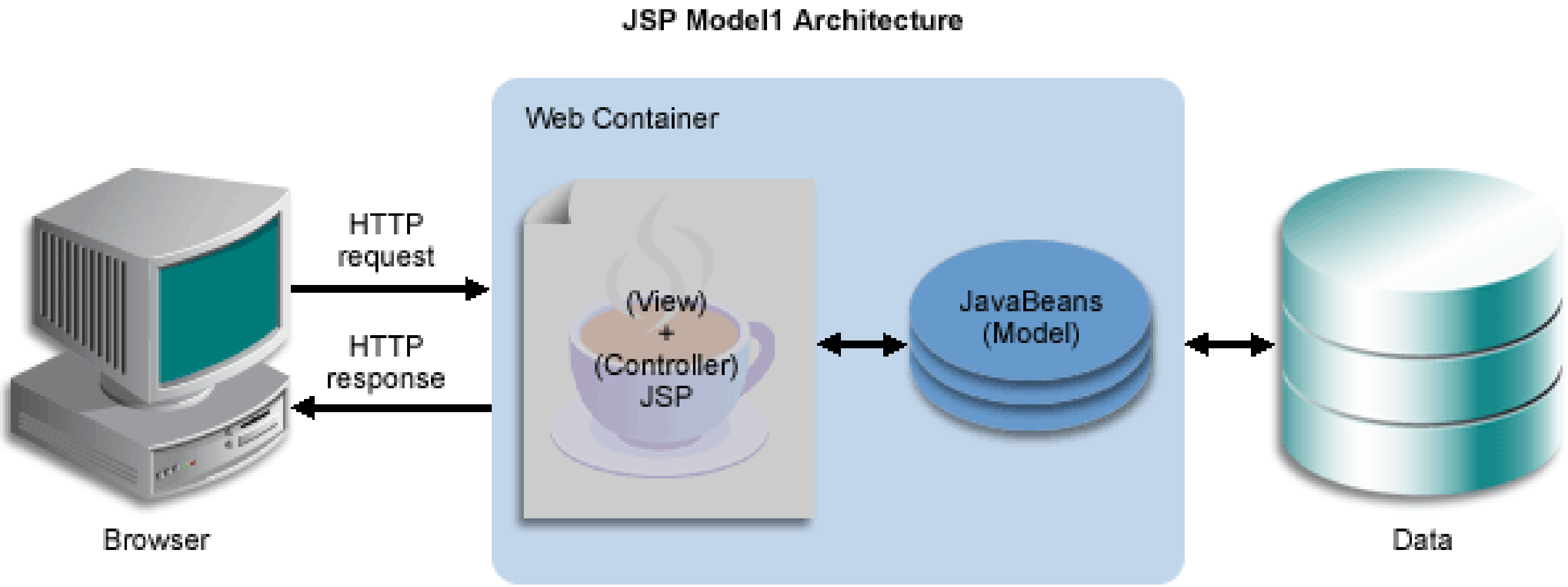
Web Application Framework

What is Framework?

- In information systems environment, a framework is **a defined support structure in which other software applications can be organized and developed**.
- A **software framework** is **a reusable design and building blocks for a software system and/or subsystem.**
  - The software frameworks significantly reduce the amount of time, effort, and resources required to develop and maintain applications.

Web Application Framework

- A Web Application Framework (WAF) is a reusable, skeletal, semi-complete modular platform that can be specialized to produce custom web applications , which commonly serve the web browsers via the Http's protocol.



Web Application Framework

- A web application framework is a web application development environment that provides reusable components and designs that are used to implement the application requirements.
  - May provide user interface templates, user authentication and authorization services or common transaction processing logic that can be used directly in an application.

Web Application Infrastructure

- Logically, **a framework operates on top of a platform** and it depends upon services provided by a platform, and **below the application that it supports**.
- The distinction between the platform and framework is not precisely applied in practice and some platforms also include framework features and the extent of services offered varies among platforms and framework
- WAFs store important data in a relational database and they interact with users via a web-based user interface.
- Any application written on top of a Web Application Framework can transparently and immediately take advantage of these basic services.

Web Application Framework

- Web application frameworks enable developers to work at a higher level of abstraction and coding.
- They provide designs, tools, and common solutions to help develop web applications.
- A web application framework is a set of tools that support web application development with:
  - A standard design model (e.g., MVC)
  - User interface toolkit
  - Reusable components for common functions (authentication, e-commerce, etc.)
  - Database support
  - Support for distributed system integration

Why we use WAF?

- Virtually all web applications have a common set of basic requirements, such as user management e.g., secure user login, password recovery), group management, and access authorization.
- **A Web Application Framework usually includes all these functionalities, refined through hundreds of production deployments**, freeing developers to focus on the needs of their specific application.

Why we use WAF?

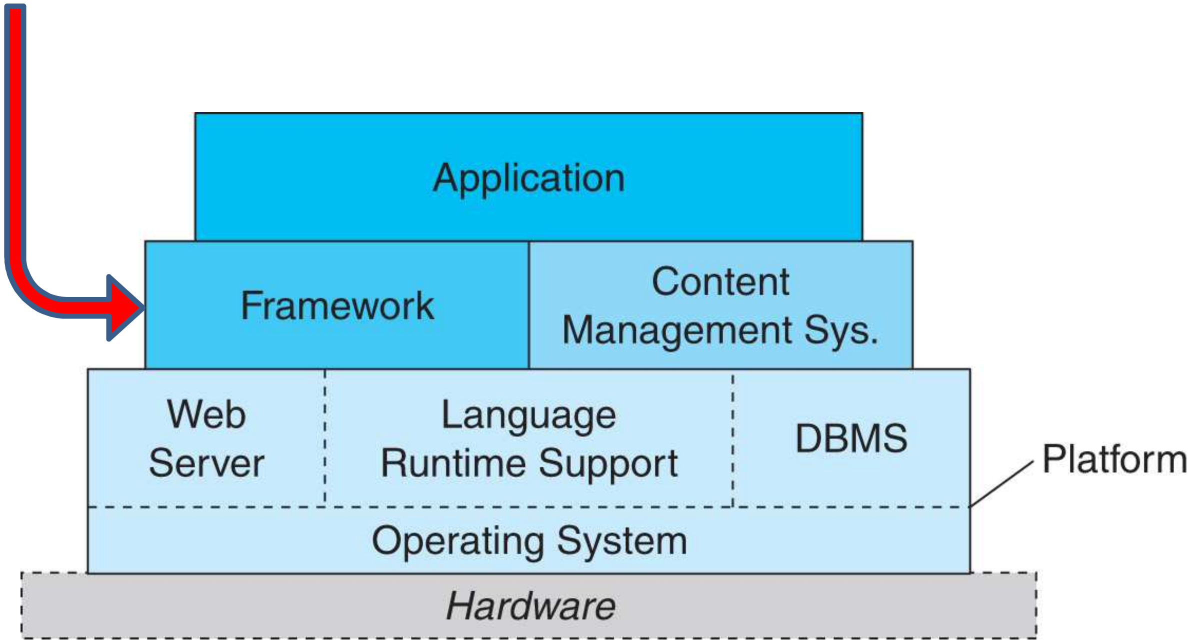
- WAFs store important data in a relational database and they interact with users via a web-based user interface.
- Any application written on top of a Web Application Framework can transparently and immediately take advantage of these basic services.

Why we use WAF?

- A framework dictates the overall architecture of the application.
- A framework predefines features in the form of reusable classes, utility classes, and base classes for developers to extend and utilize.
- Frameworks become popular because they solve common problems in a simplified way and do so without seriously compromising the intent of the application they support.

Web Application Framework

- Frameworks give application developers more powerful building blocks to work with



Java Web Aplication Framework

- The Model View Controller pattern based Java web frameworks hold a lot of potential to make the developer's life easier, their development time faster, and their application more maintainable.
- So the time invested in deciding on which framework to use is worthwhile.
- **The success of the Web MVC has triggered a proliferation of the Java web presentation frameworks.**

Java Web Aplication Framework

- There are four types of Java Web Aplication Framework
  - Request-based Framework
  - Component-based Framework
  - Hybrid – Meta Framework
  - RIA-based Framework

Java Web Aplication Framework

- A Request-based Framework uses controllers and actions that directly handle incoming requests.
- Examples
  - Struts
  - WebWork
  - Beehive
  - Stripes

Struts

- Struts was originally developed by Craig McClanahan and donated to the Apache Foundation in May 2000.
- Struts has been a de facto framework with a strong and vibrant user community.
- Struts uses and extends the Java Servlet API to adopt the "Model 2" approach, a variation of the classic Model-View-Controller (MVC) design pattern.

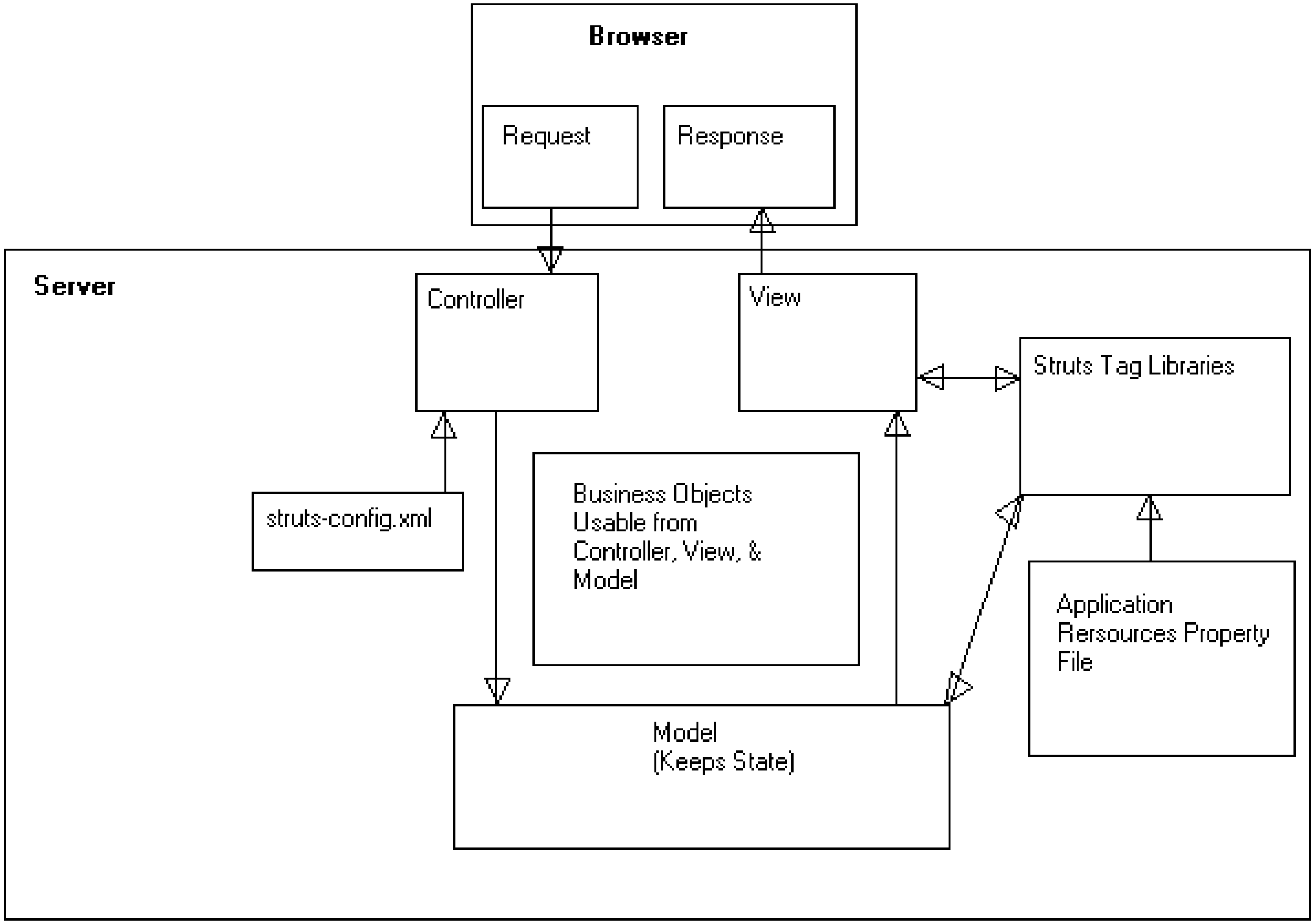
Struts

- Features:
  - use of MVC design paradigm
  - Centralized XML-based application configuration that can define many functions
  - Action definitions that link user interface events to Controller and View modules
  - Extensive use of custom JSP tags in the view component to support HTML form processing,
  - event handling, internationalization, etc.

Struts

- A user sitting at a browser clicks a button, link, or types in a url.
- A Struts Servlet receives that request. It normally hands over the request to a FormAction class.
- The FormAction class is responsible for validating any form input.
- Control is then passed on to an Action class.
- It's the responsibility of the Action class to call any Model components, such as classes responsible for updating information in a database.
- The final step is typically to forward to another JSP, thus a response being sent back to the browser.

Struts


Browser
Request
Response
Server
Controller
View
Struts Tag Libraries
struts-config.xml
Business Objects Usable from Controller, View, & Model
Application Rersources Property File
Model (Keeps State)

WebWork

- WebWork was originally developed by Rickard Oberg in 2001, and released as an open source project on **SourceForge** in March 2002.
- WebWork provides robust support for building reusable UI templates, such as form controls, UI themes, internationalization, dynamic form parameter mapping to JavaBeans, and robust client and server side validation.

Component-based Framework

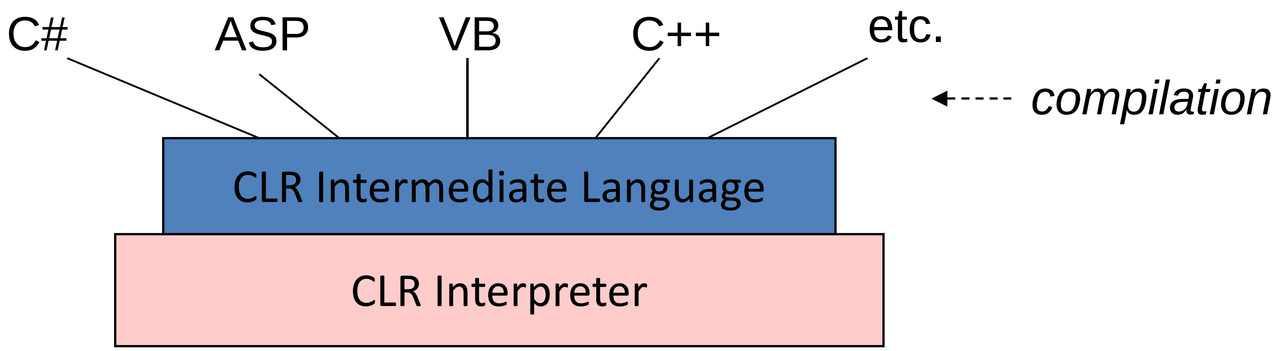
- A Component-based Framework abstracts the internals of the request handling and encapsulates the logic into reusable components, often independent from the web medium
- The state is automatically handled by the framework, based on the data that is present in each component instance.
- Together with some form of event handling, this development model is very similar to the features offered by desktop GUI toolkits.
  - Examples: JavaServerFaces, Tapestry, Wicket

Component-based Framework

- JavaServer Faces (JSF) is a server-side user interface component framework for Java-based Web applications.
- JSF contains an API for representing UI components and managing their state; **handling events**, **server-side validation**, and **data conversion**; **defining page navigation**; **supporting internationalization and accessibility**; and **providing extensibility for all these features**

.NET Framework

- .NET Framework is a Microsoft (licensed) product
- .NET supports many programming languages through its Common Language Runtime (CLR) system
- Feature of .NET Framework
  - Common Language Runtime,
  - ASP.NET, C#, ADO.NET, Enterprise Services,
  - Role-based security and user authentication mechanism,
  - Windows Forms,
  - .NET Compact Framework



C# ASP VB C++ etc.

compilation

CLR Intermediate Language

CLR Interpreter

Ruby On Rails

- Ruby: a dynamically typed object-oriented programming language
- Rails: a web application framework, featuring:
  - automatic code skeletons
  - built-in testing features
  - object-relation mapping
  - default implementation of common webapp functions
- Features of Ruby on Rails
  - Implementation of the model-view-controller design pattern,
  - Default implementation of many common we application operations,
  - Scaffolding,
  - Built-in testing features,
  - Separate development, test, and deployment environments,
  - Object-to-relation mapping Rake

Web Application Infrastructure

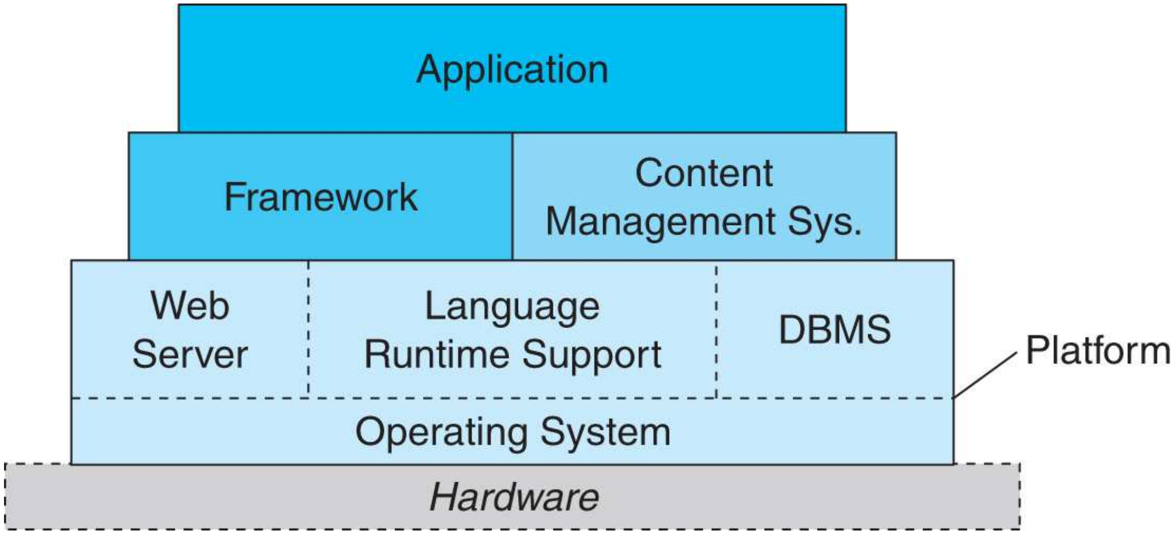
- Content Management System (CMS) is a tool for creating, cataloging and providing access to HTML pages, images, and other items that a web application might provide to users.
  - Allows use of workflow management process that to configure the needs and policies of an organization to allow integration of new content into the application by different levels of production and authorization
  - Responsible for version control if there are multiple versions of some documents through removal and archiving of obsolete content
  - Simplifies the process of content creation by separating semantic style from structure and content.

Content Management System

- Content Management Systems (CMS) provide the infrastructure support required for creating and managing web content for large, dynamic web applications.
- CMS provides a variety of services, including:
  - Content Tracking,
  - Workflow and Collaboration Management,
  - Versioning,
  - Formatting,
  - Archiving,
  - Blogs,
  - Forums,
  - Wikis, etc.

Content Management System

- A CMS is a tool for creating, cataloging, and accessing documents (web pages)

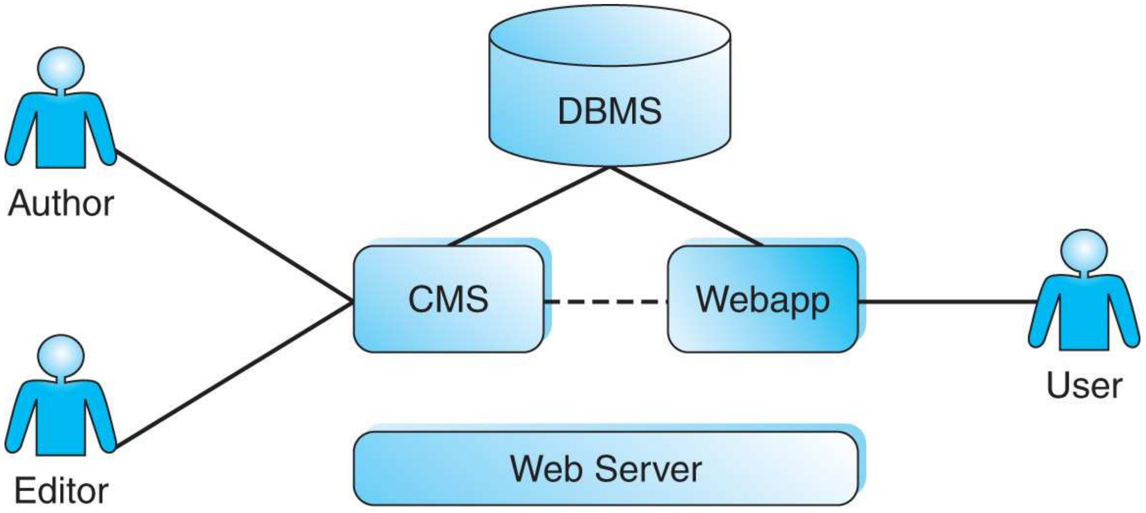


Content Management System

- Benefit of using a CMS to an organization
  - A CMS brings efficiency to the process of creating and deploying web content by reducing the number of steps required, reducing the degree of technical support required, an by automating the process.
  - A CMS also helps to manage the content development process, establish a brand image by enforcing standard configuration and appearance of web content, and enable the creation of content in popular formats without requiring custom application programming.

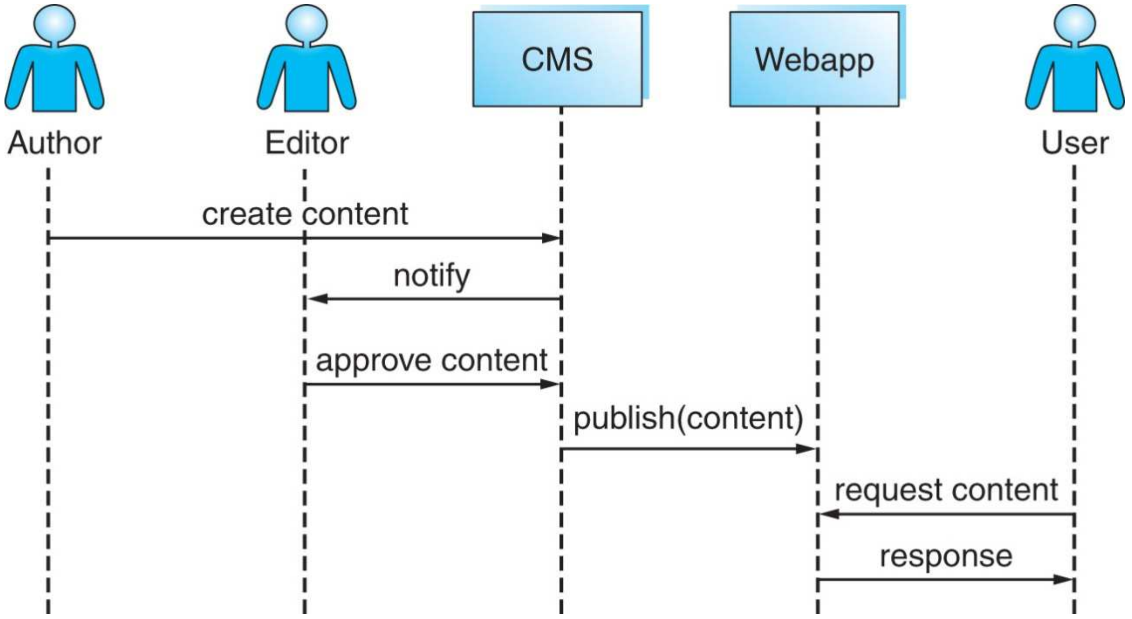
Content Management System

- A CMS can streamline and automate the process of creating, approving, deploying, and retiring web content



Workflow Management

- A CMS can coordinate the steps involved in creating, editing, and approving web content within an organization



Content Management Systems

- Joomla!
  - open source, free
  - based on MySQL and PHP
- Features
  - Supports a variety of content types, including basic web pages, RSS feeds, blogs, polls, forms, etc.
  - Supports content development and deployment with role-based workflow management and administration,
  - Provides website enhancement and analysis features such as traffic statistics, print capability, content rating, and search.

Content Management Systems

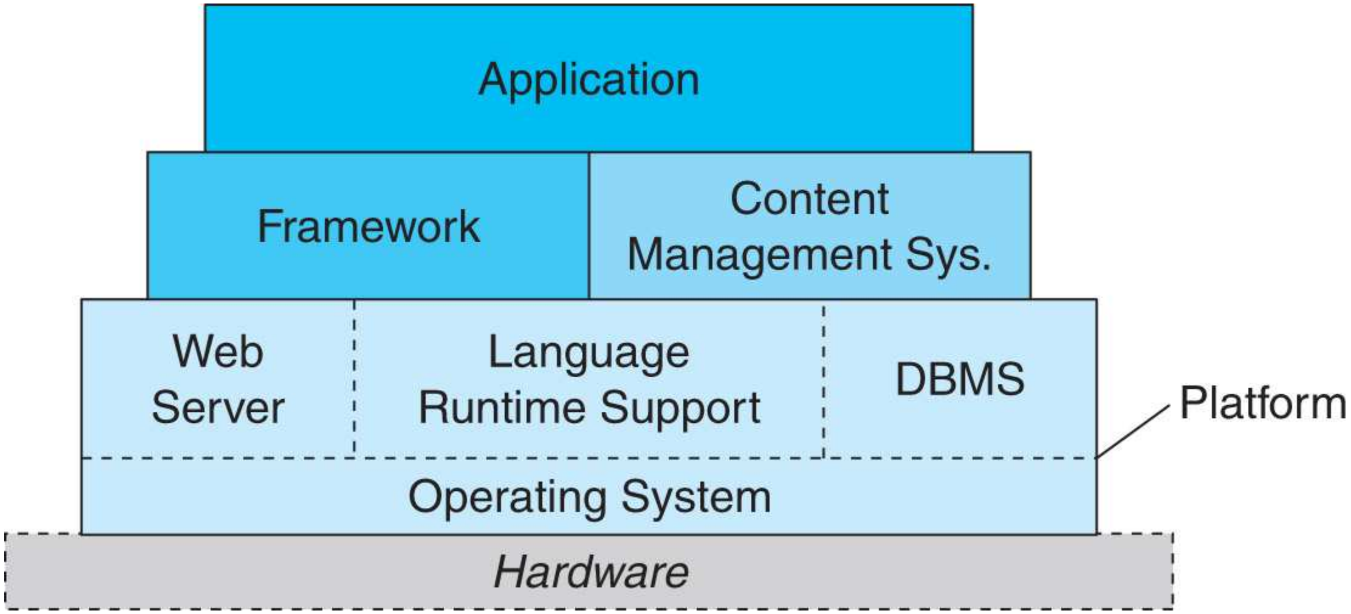
- Vignette
  - proprietary product
  - includes extensive workflow management support
  - integrates with related development tools
- Features
  - Support for workflow management, including library and version control and collection of workflow process metrics.
  - Enables content creation and acquisition from a variety of sources, including word processing, e-mail, and legacy systems, and it enables distributed work teams to collaborate on content creation.
  - Supports content publishing across a variety of platforms and also supports multi-target publishing
  - Supports standardization and common branding of information products across an organization

Web Application Software Stack

- The web application software stack is a layered model of web system architecture.
- It includes the platform, framework, application and the content management system.
- The platform serves as the system base and provides services to the framework and the CMS, which sit on the layer right on top of it.
- The application, which sits at the very top of the stack, extends and/or customizes the framework and CMS to create a unique web application.

Web Application Software Stack

- The web application software stack includes platform, framework, CMS, and application


Application
Framework
Content Management Sys.
Web Server
Language Runtime Support
DBMS
Platform
Operating System
Hardware