IF3110 – Web-based Application Development

Internet Application Concept

Objective

- Students understand the differences Internet App vs Web App
- Students know various components composing a Web App
- Students understand typical processes done in a Web App

Main Concept

- Internet Programming → Programming Internet-based Application (Internet App)
- Application that is distributed across the place; using Internet as communication means
- Keywords:
  - Distributed System
  - Communication via Internet

Characteristics

- Concurrency
  - Handle many tasks at the same time (e.g., send/receive data, process user requests, handle many clients)
- Synchronize
  - activity coordination
  - time/timing handling
- Exception Handling
  - a fail in handling a user; won’t cause any problems to other users

Distributed System

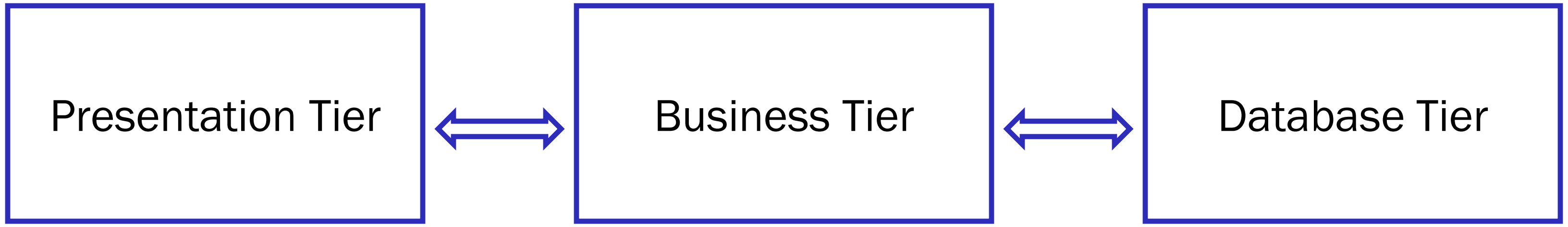
- Main Classes
  - Client Server: service requester (client) and provide services (server)
  - Peer-to-peer: each component is in the same level in providing and requesting services
- In a peer-to-peer system, each component plays a role as client and server

Client-Server Variant

- Stateless Client-Server
  - Each client/server doesn’t store each other’s status
  - Each client-server interaction is independent and stateless
- Stateful Client-Server
  - Store info about client-server interaction
  - Client can send message based on the existing interaction context (i.e., no need to provide such info at the message)

Other Flavour of Client-Server

- RPC: client-server interaction seen as procedure remote invocation
- N-tier systems: expand a server into various tiers of servers



Client-Server Trade-Off

- Advantageous
  - Distributing computation across several machines
  - Client can access services remotely
  - Client & Server can be designed and developed independently
  - Server can process many simultaneously requests
  - Data can be stored in either centrally in a server or distributed across clients
- Disadvantageous
  - communication delay
  - need to consider sync and parallel/concurrent process in the server

Protocol Communication

- Definition: a set of rules agreed by client-server in communication to each other
  - Application protocol
    - client-server can send messages to each other with following a particular format/syntax with a specific order
  - Transport protocol
    - a message is chunked into many packets
    - send the packets with various network routes
    - at the destination the packets are constructed to the original message
- This course emphasizes at application protocol

Application Protocol using Internet

- Web (protokol aplikasi: HTTP)
- Naming (DNS)
- E-mail (IMAP, POP, SMTP)
- Chatting
  - open standard: IRC
  - non standard: YM, ICQ, MSN chat, AOL, dll
- File transfer (FTP)
- Remote terminal (telnet)
- Directory service (LDAP)
- Network monitoring (SNMP)
- Web service (SOAP)
- Voice (SIP, XMPP, ASTERISX)
- etc.

Internet-based vs Web-based Application

Internet-based

- Using Internet Application Protocol or defining own protocol
- Application at the server communicates directly to client
- Application can be a standalone or a component for an existing application

Web-based

- Using HTTP
- Application communicate to Client via Web Server
- Application commonly runs in web browser

Technology in Web-based Application Development

- Web client (web browser)
- Web server
- URL : Uniform Resource Locator
- HTTP : HyperText Transfer Protocol
- HTML : HyperText Markup Language
- CSS : Cascading Style Sheet
- Web Programming
  - CGI (Common Gateway Interface)
  - server side scripting
  - client side scripting
  - plug-in

Web client (web browser)

web browser

- a software
- runs in client/user's compouter
- navigate through the web
- render/veiw the web page

Example:

- Chrome (Windows)
- Internet Explorer (Windows)
- Mozilla Firefox (Windows & Linux)
- Opera (Windows & Linux)
- Safari (Mac)
- lynx, berbasis teks (Linux)

Web server

- web server
  - a software
  - runs in a server
  - store web document so it can be accessed by users across the Internet

- Example:
  - Apache (Linux & Windows)
  - NginX
  - MS Internet Information Server / IIS (Windows)
  - Tomcat, untuk Java (Windows & Linux)

URL (Uniform Resource Locator)

URL is locating a resource (file) in the internet

URL format is defined in RFC 1738 (http://www.ietf.org/rfc/rfc1738.txt)

URL has protocol type

Example

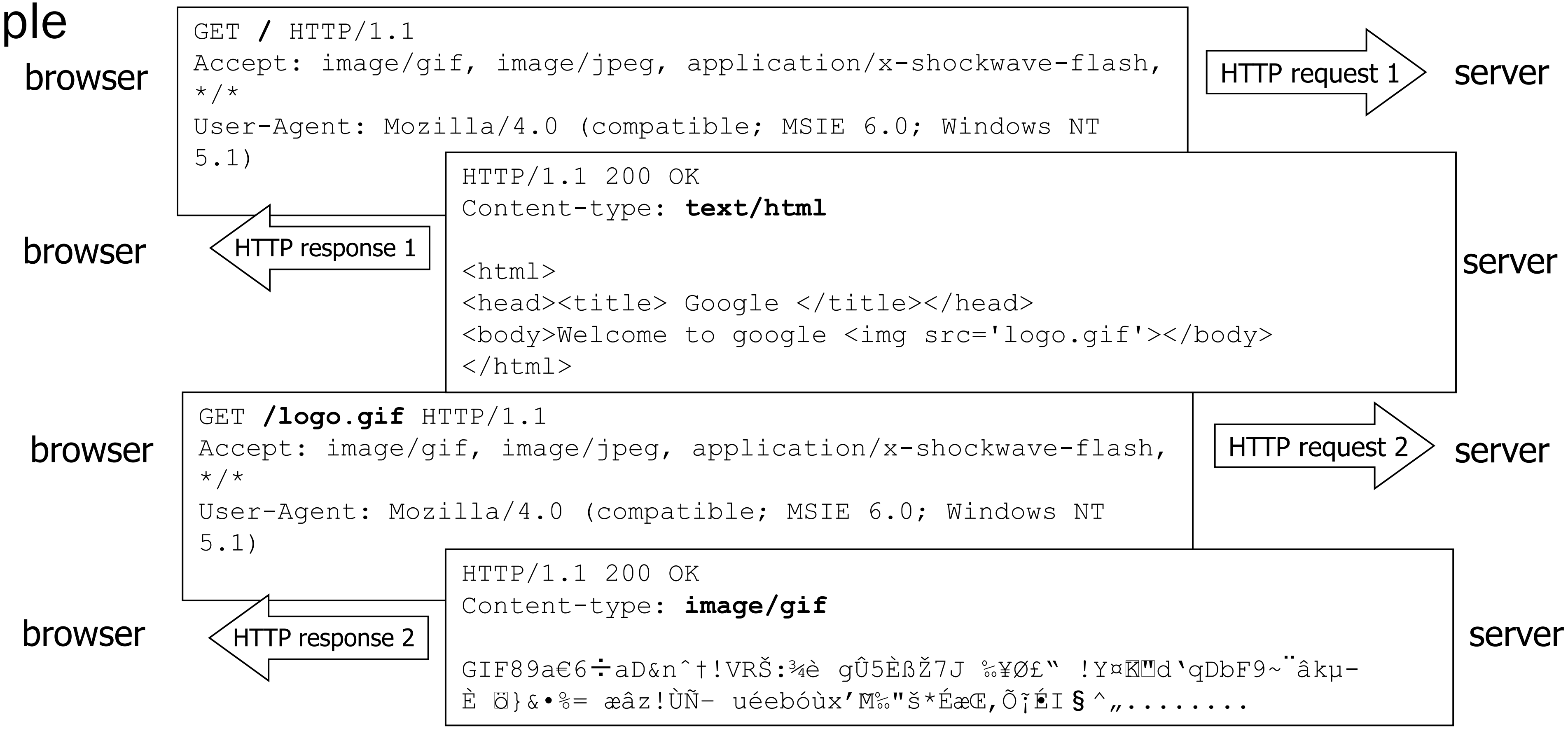
- http://www.if.itb.ac.id/
- mailto:fulan@informatika.org
- ftp://ftp.itb.ac.id/

Example of URL in the web:

- http://www.itb.ac.id/campus-life/index.html
- http://www.google.com/search?hl=en&q=URL+RFC
- http://www.indymedia.org:8081/

HTTP (HyperText Transfer Protocol)

- HTTP is a standard web protocol communication
- HTTP standard (HTTP 1.1) defined in RFC 2616 (http://www.ietf.org/rfc/rfc2616.txt)
- Example



browser

```
GET / HTTP/1.1
Accept: image/gif, image/jpeg, application/x-shockwave-flash,
*/*
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT
5.1)
```

HTTP request 1

server

browser

HTTP response 1

```
HTTP/1.1 200 OK
Content-type: text/html

<html>
<head><title> Google </title></head>
<body>Welcome to google <img src='logo.gif'></body>
</html>
```

server

browser

```
GET /logo.gif HTTP/1.1
Accept: image/gif, image/jpeg, application/x-shockwave-flash,
*/*
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT
5.1)
```

HTTP request 2

server

browser

HTTP response 2

```
HTTP/1.1 200 OK
Content-type: image/gif

GIF89a€6÷aD&n^†!VRŠ:¾è gÛ5ÈßŽ7J ‰¥Ø£‟ !Y¤K"d‛qDbF9~¨âkµ-
È Ö}&•%= æâz!ÙÑ- uéebóùx’M‰"š*ÉæŒ,Õ¡ÉI§^„........
```

server

HTML (HyperText Markup Language)

- HTML is a standar document in the web
- HTML standard (HTML 4.01) accessible at http://www.w3.org/TR/html4/
- HTML standard (HTML 5) accessible at https://html.spec.whatwg.org/multipage/
- Example:

```
<html>
<head>
    <title>My first HTML document</title>
</head>
<body>
    <p>Hello world!<br>Welcome to my <b>first</b> HTML
page.
    </p>
</body>
</html>
```

- Output:

Hello world!
Welcome to my **first** HTML page.

CSS (Cascading Style Sheet)

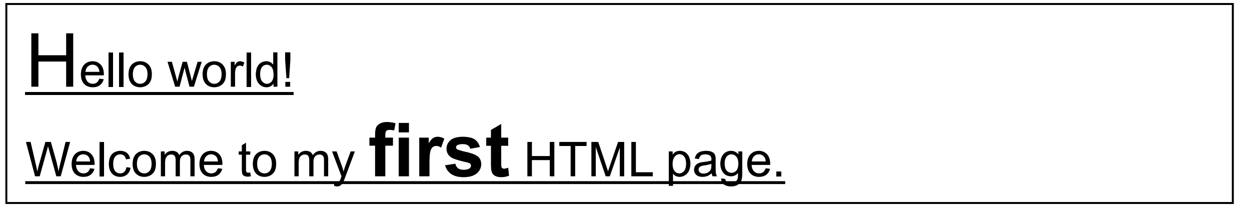
- CSS is a mechanism to define style (font, template, colour) in a web document
- CSS standard (CSS 2) is accessible at http://www.w3.org/TR/REC-CSS2/
- Example:

```
<html>
<head>
    <title>My first HTML document</title>
</head>




<body>
    <p>Hello world!<br>Welcome to my <b>first</b> HTML page.
    </p>
</body>
</html>
```

- Output:



Hello world!

Welcome to my **first** HTML page.

TCP/IP (Internet) Programming

- Web Programming often uses HTTP as transport protocol and HTML/XML as message format
  - Text oriented
  - Stateless, client server oriented
- More flexibility can be achieved by using transport or network protocol directly
- IP Network (layer transport):
  - TCP: connection oriented
  - UDP: datagram oriented
- Communication mode:
  - Unicast, Multicast, Broadcast

Web Programming

Web Programming

NGINX

- CGI, executing code in server-side (perl, C)
  - Web server executes a program and the output is piped to HTTP response
- server side scripting (PHP, ASP, JSP, Python)
  - Web server identifies and runs script/program and insert the outputs as a part of the web document
- client side scripting (JavaScript, JScript, VBScript)
  - Web browser identifies and runs the script/program and insert the outputs to the web document (received from the server) or modifies the web document or does what the script orders (e.g., XHR, Open Socket)
- plug-in, eksekusi program di sisi client (applet, ActiveX, Flash)
  - Web browser executes the program with the help of plugin and view the outputs at a defined place in the web document

Web Application Evolution

- Web site – a collection of a document accessible through internet
- Web app – a client/server app through a web as the client, and uses Internet to communicate

- Web Server → Web App Server

Typical web app processing

Source:

Shklar, L.
Web Application Architecture:
Principles, Protocols and
Practices
Wiley Publishing, Inc., 2003

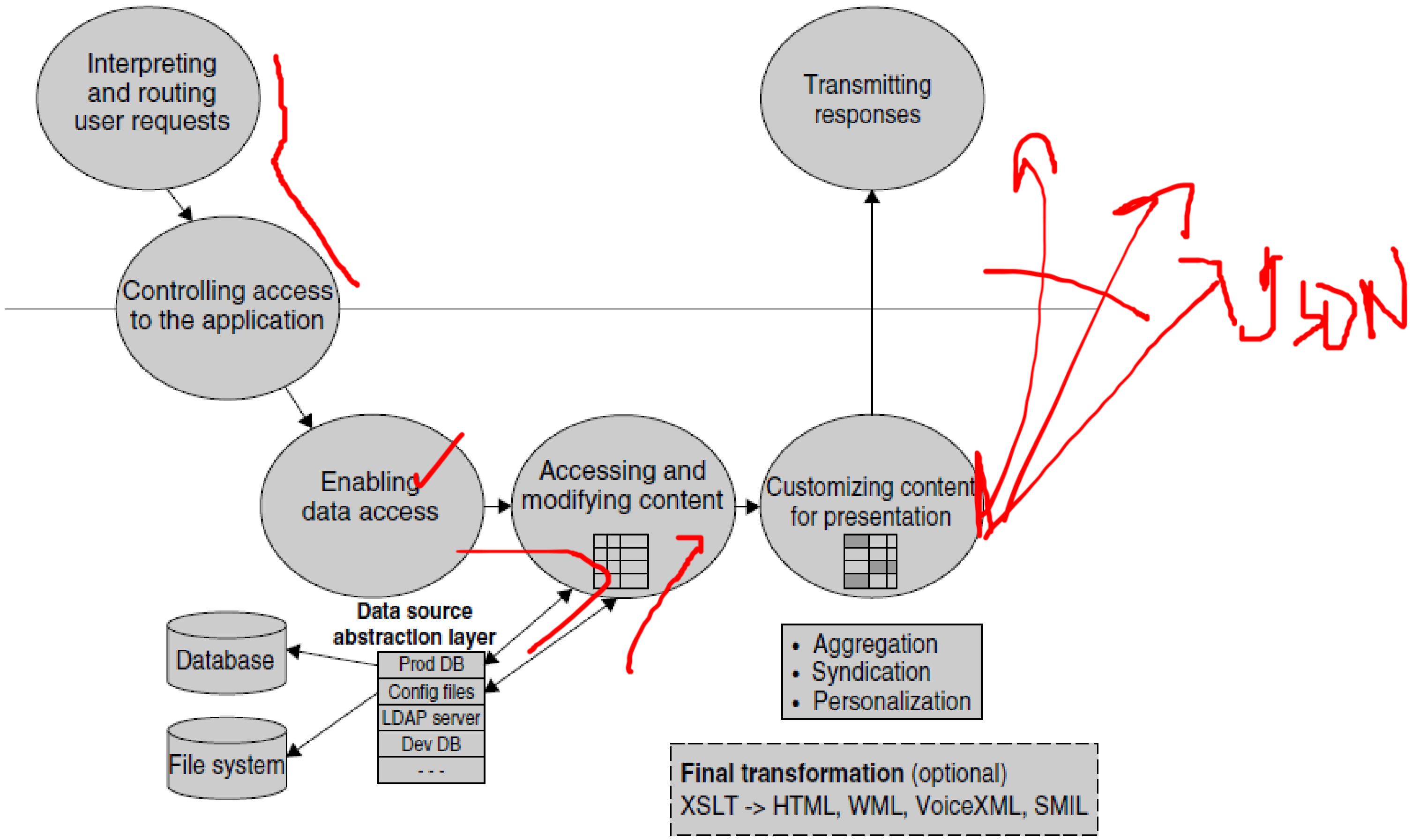

Interpreting and routing user requests
Controlling access to the application
Enabling data access
Accessing and modifying content
Customizing content for presentation
Transmitting responses
Data source abstraction layer
Prod DB
Config files
LDAP server
Dev DB
- - -
Database
File system
Aggregation
Syndication
Personalization
Final transformation (optional)
XSLT -> HTML, WML, VoiceXML, SMIL
CDN

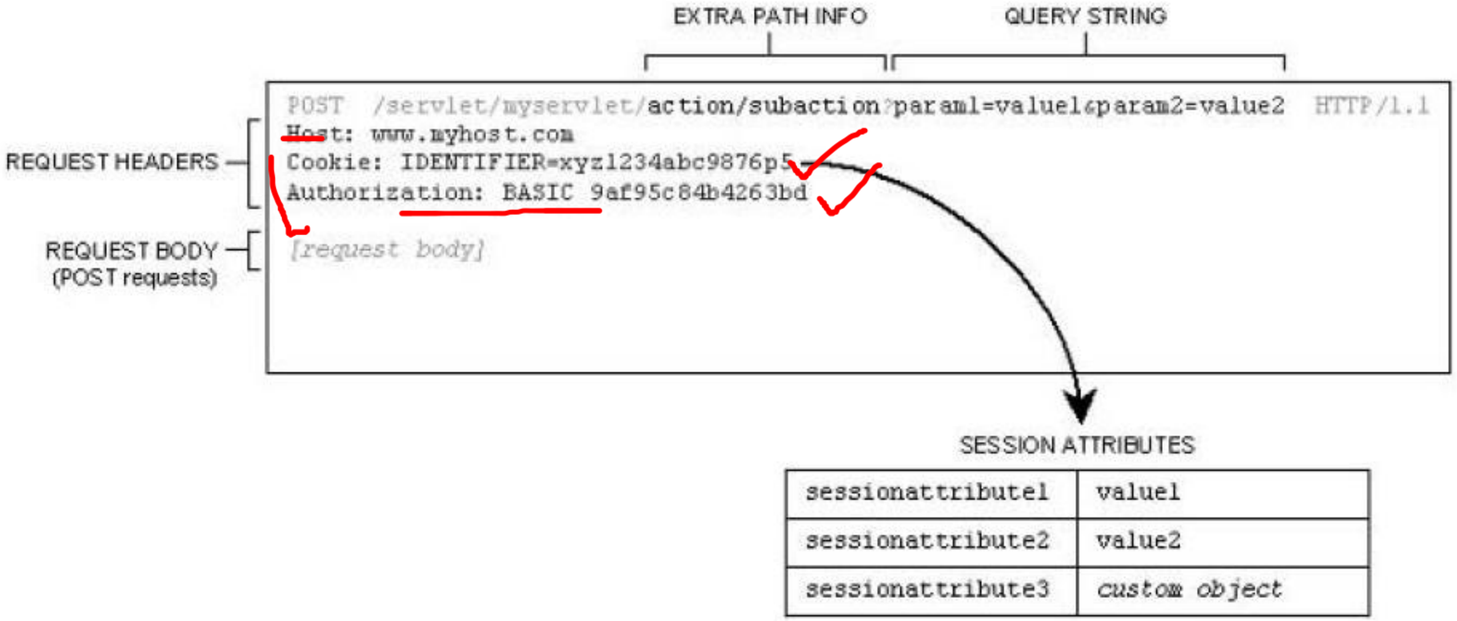
Figure 8.3 Processing flow in a typical Web application (Above the grey line—Web server; below the grey line—Web application)

Web Stack

- A collection of software required to develop and run web applications
- Contains all software layers
  - Operating System
  - Web Server
  - Programming Language (frontend & backend)
  - Web Framework (frontend & backend)
  - Database Server
- Examples: XAMP
  - LAMP, MEAN, ...
- It is **different** from Web Framework

Interpreting & processing request

- Browser communicate the request via HTTP (info sent by the browser called *request context*)
  - URL, query string, request headers, request body, session information



```
POST  /servlet/myservlet/action/subaction?param1=value1&param2=value2  HTTP/1.1
Host: www.myhost.com
Cookie: IDENTIFIER=xyz1234abc9876p5
Authorization: BASIC 9af95c84b4263bd

[request body]
```

SESSION ATTRIBUTES

sessionattribute1	value1
sessionattribute2	value2
sessionattribute3	*custom object*

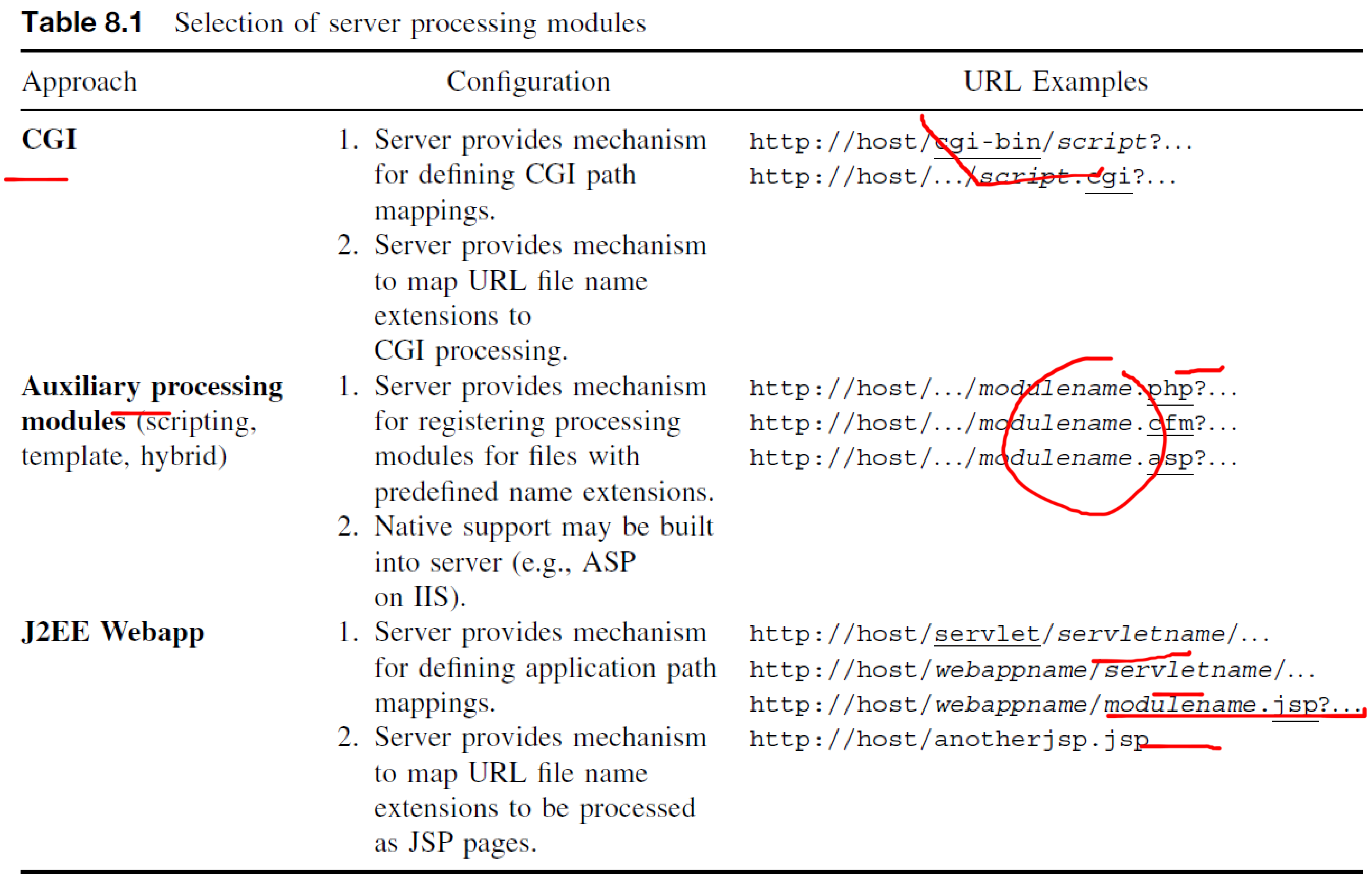


Table 8.1 Selection of server processing modules

Approach	Configuration	URL Examples
CGI	1. Server provides mechanism for defining CGI path mappings. 2. Server provides mechanism to map URL file name extensions to CGI processing.	http://host/cgi-bin/*script*?... http://host/.../*script*.cgi?...
Auxiliary processing modules (scripting, template, hybrid)	1. Server provides mechanism for registering processing modules for files with predefined name extensions. 2. Native support may be built into server (e.g., ASP on IIS).	http://host/.../*modulename*.php?... http://host/.../*modulename*.cfm?... http://host/.../*modulename*.asp?...
J2EE Webapp	1. Server provides mechanism for defining application path mappings. 2. Server provides mechanism to map URL file name extensions to be processed as JSP pages.	http://host/servlet/*servletname*/... http://host/*webappname*/*servletname*/... http://host/*webappname*/*modulename*.jsp?... http://host/anotherjsp.jsp

Controlling User Access

- HTTP provides an authentication mechanism
  - web server ask authentication info; when the request doesn’t have such info

Web Application Architecture

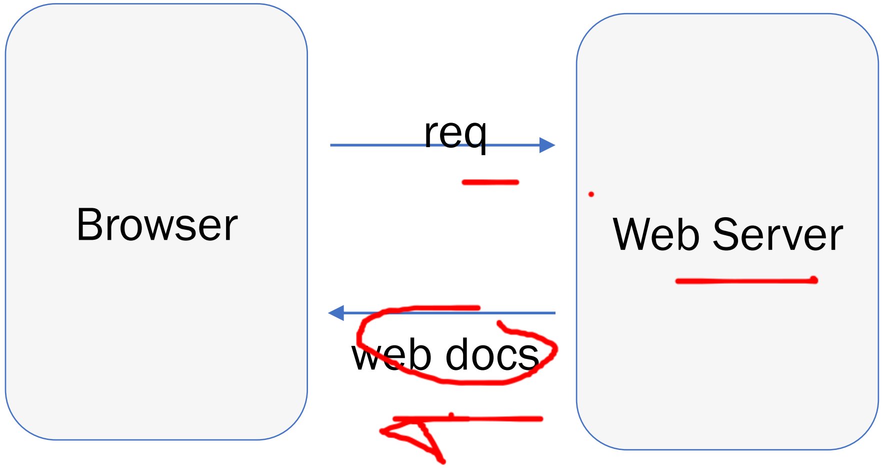
- presentation layer:
  - markup / HTML/XHTML
    - templating system
    - code segregation
  - CSS
- presentation/page logic
- business/application logic
- persistent store

N-tier

Web Site Architecture (Old)

Source“:

Microsoft(r) Application Architecture Guide, 2nd Edition (Patterns & Practices)
Microsoft Press, **2009**



Web Application Architecture

Source“:

Microsoft(r) Application Architecture Guide, 2nd Edition (Patterns & Practices)
Microsoft Press, **2009**

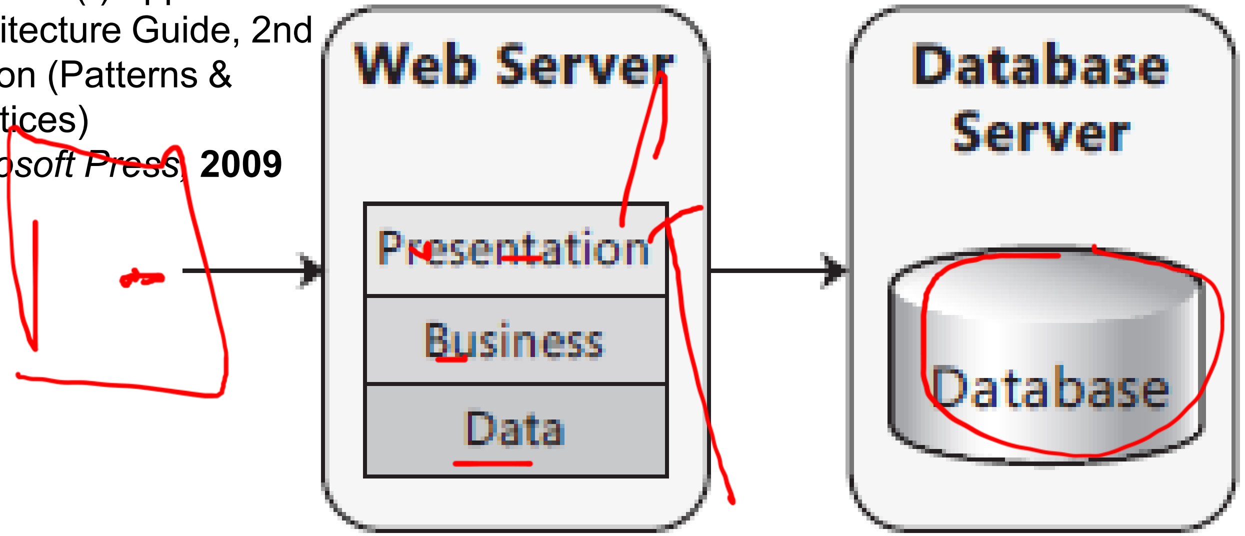


Figure 2
Nondistributed deployment of a Web application

Web Application Architecture

Source“:

Microsoft(r) Application Architecture Guide, 2nd Edition (Patterns & Practices)
Microsoft Press, **2009**

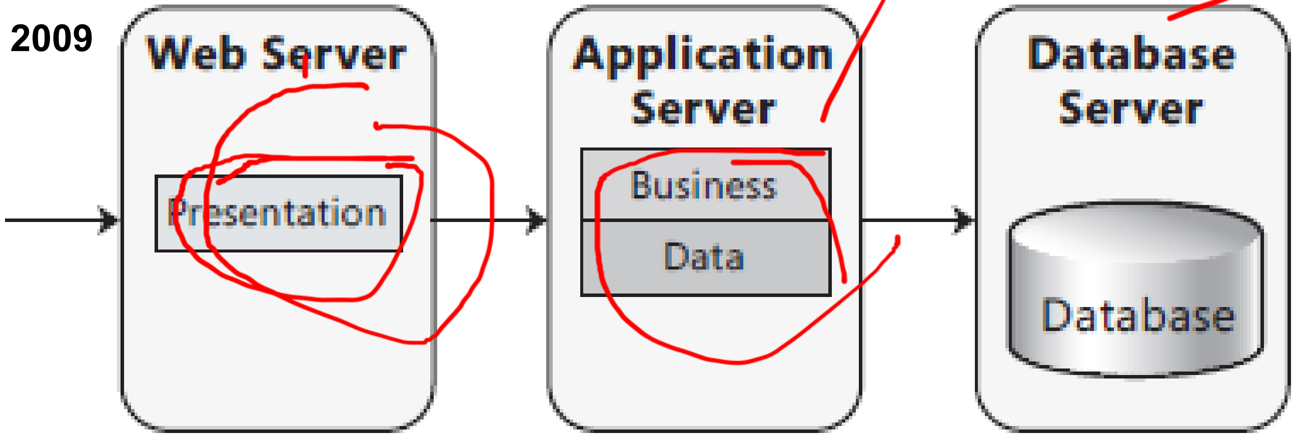


Figure 3
Distributed deployment of a Web application

Application Architecture

Source“:

Microsoft(r) Application Architecture Guide, 2nd Edition (Patterns & Practices)
Microsoft Press, **2009**

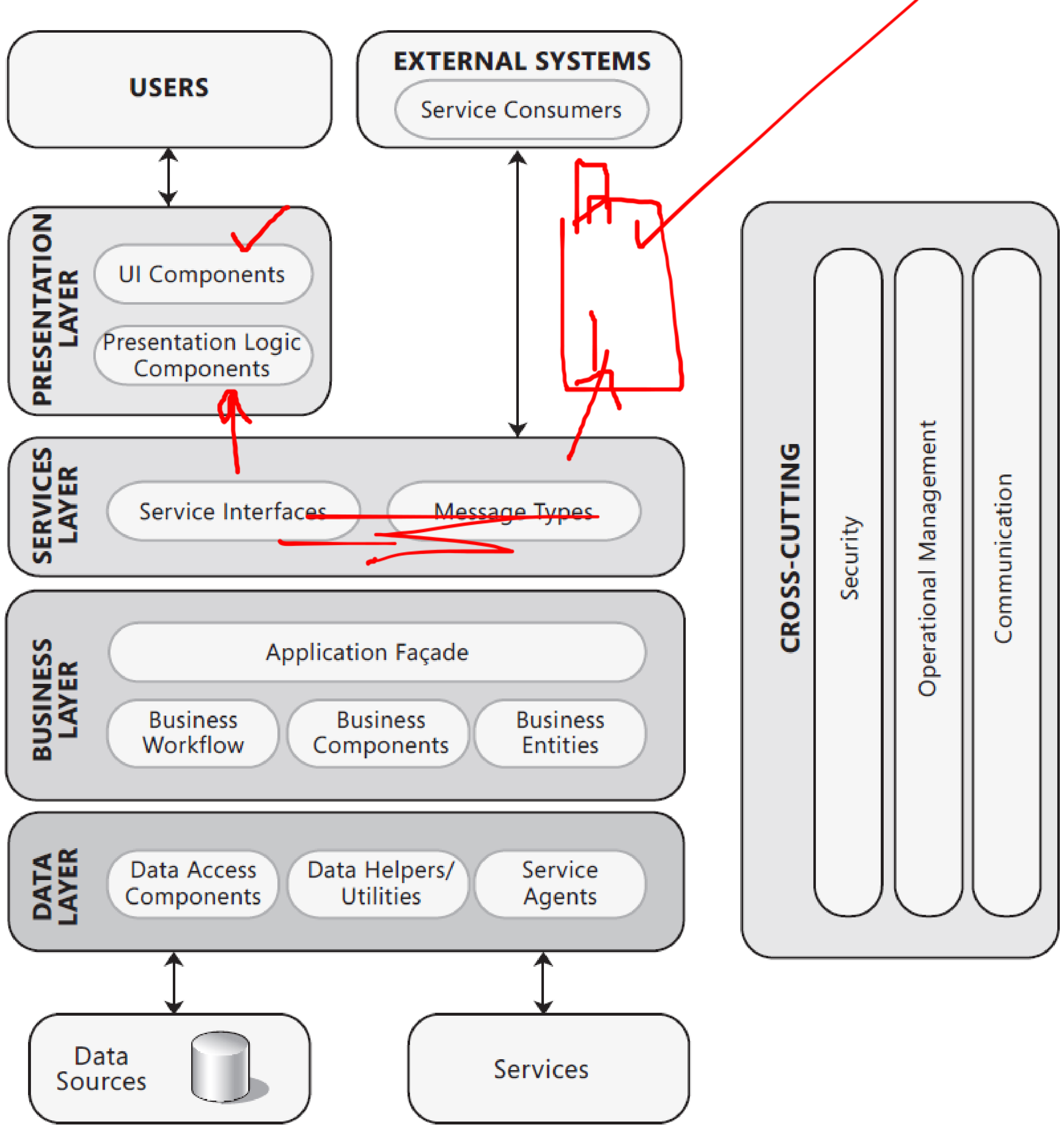


Figure 1
Common application architecture

Common Aspect in Web-App

- App request processing
- Authentication
- Authorization
- Caching
- Exception Management
- Logging & Instrumentation
- Navigation

- Page layout
- Page rendering
- Session management
- Validation
- Internationalization
- User Experience

Exercise: Under the Hood of Internet App (FTP and Web App)

- Objective
  - Students understand how an Internet App Client communicate with the Server
- Tools
  - FTP Client FileZilla https://wiki.filezilla-project.org/Using
  - Simple HTTP Server for Testing http://neverssl.com
  - Free FTP Server for Testing: https://dlptest.com/ftp-test/
    - FTP URL: ftp.dlptest.com or ftp://ftp.dlptest.com/
    - FTP User: dlpuser
    - Password: rNrKYTX9g7z3RgJRmxWuGHbeu
  - Capturing Live Network Data with Wireshark https://www.wireshark.org/docs/wsug_html/#ChapterCapture
  - Inspect Web Network Activity using Chrome DevTools https://developer.chrome.com/docs/devtools/network

Exercise: Under the Hood of Internet App (FTP and Web App) - contd

- Instruction
  - Access the service via a client (HTTP/FTP) that has been run at a HTTP/FTP server
  - Capture the traffic with Wireshark

Question:

- What has been done under the hood (in the network)?
- What are the differences between FTP and HTTP based on what you have discovered?
- What are the differences between stateful protocol and stateless protocol?