



WEB APPLICATION ENGINEERING II

Lecture 1

Umar Ibrahim Enesi



Objectives

- Basic understanding of the internet and web architecture.
- Review of HTTP, HTTP Request and HTTP Response
- Review of client side web application and the technologies behind it.
- Understand the technologies behind server side applications.
- Understand how all the pieces of web applications fit together.



Internet and the World Wide Web (WWW)

Internet and world wide web (or simply Web) are two separate but related things.

INTERNET

- Internet is the interconnection of network of networks.
- A network infrastructure owned by no one.
- The Internet links computers.

WWW

- WWW is a mechanism for accessing resources stored in computers that are connected to the Internet.
- Resources stored on a computer and accessed via the web are called web resources (eg web pages).
- WWW links web resources.



Internet and WWW (cont.)

- WWW is built on top of the Internet.
- WWW uses the Internet as a medium for accessing information (web resource).
- Other ways in which the Internet is of use:
 - E-mail – for sending electronic mails
 - Instant messaging – for sending short message in real time
 - File transfer
- Communications over Internet is governed by *protocols*.
- WWW uses HTTP protocol for disseminating information



Hypertext Transfer Protocol - HTTP

- HTTP is a protocol that specifies how hypertext documents and other web resources are transferred over the Internet.
- Hypertext are text with *hyperlinks*.
 - Often written in HTML.
- Hyperlinks:
 - Logical addresses that references other web resources over the web
 - Interconnects hypertext documents
 - Described by URI (URL)



URN, URL, URI, IRI

- **Uniform Resource Name (URN):** Identifies resources by name
- **Uniform Resource Locator (URL):** Identifies resources by location and specifies method of retrieval.
- **Uniform Resource Identifier (URI):** A recommended term that describes URN and URL.
- **Internationalised Resource Identifier (IRI):** An identifier that fully supports international characters.

HTTP relies upon the Uniform Resource Identifier (URI) to identify web resources.



Web Application

- A web application is a software that runs over the web.
- Web applications mostly run on hardware connected to a network (Internet or intranet).
- Delivers dynamic web content
- Incorporates databases and web services



Web Application Architecture (Logical)

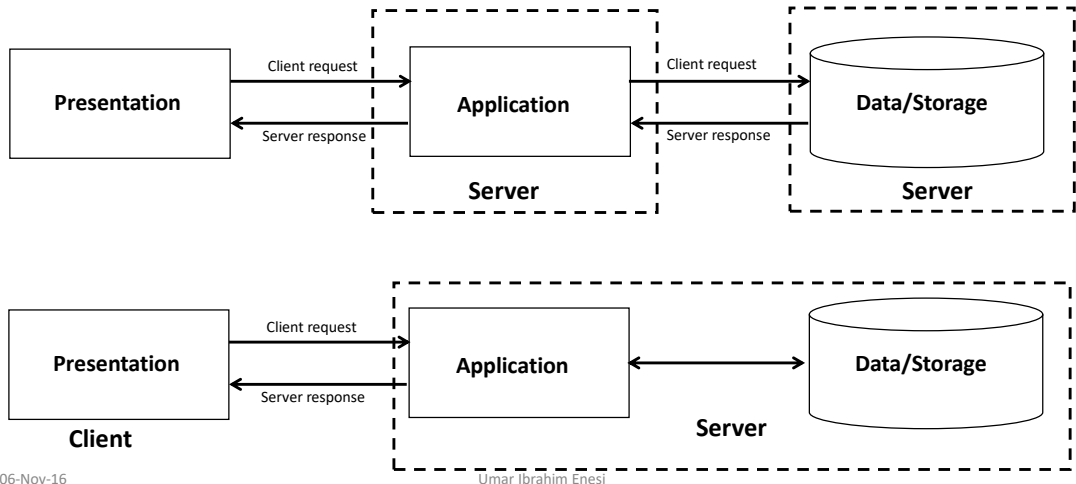
- The most common architecture for web application is the 3-tiered architecture.
- The 3-tiered architecture consists of *presentation*, *application* and *storage*.





Web Application Architecture (Interaction)

- With respect to interaction: Client-Server Architecture.



Client vs Server

- The terms "*client*" and "*server*" refer only to the roles that computers perform for a particular connection.
- Servers are computers that delivers web resources to clients
 - Servers run server software: Apache, Nginx etc.
- Clients (user-agents) make request for resources.
 - Clients run client software:
 - Browser based (eg Google Chrome, Opera...)
 - Client or custom software (desktop) based (eg Google drive, ...),
 - Mobile web app (Facebook, Twitter, SoundCloud...).
 - Spiders (web-based robot) and command line tools



HTTP Messaging

- HTTP is a protocol that operates by exchanging structured messages between client and server.
- HTTP specifies that a client initiates communication.
- A client sends an HTTP request to a server in the form of a request message
- A server responds to a client's request by sending one or more HTTP response messages



HTTP Request/Response Format

```

request-line
* (header-field CRLF)
CRLF
[message-body]
  
```

Request Message Format

```

status-line
* (header-field CRLF)
CRLF
[message-body]
  
```

Response Message Format

Key

* => 0 or more

CRLF => newline (carriage return or line feed)

[] implies optional



request-line

method **SP** *request-target* **SP** *HTTP-version* **CRLF**

method => GET, POST, PUT, HEAD, DELETE, OPTIONS, ...

request-target => resource being requested (or act upon)

HTTP-version => in this case "HTTP1.1"

SP => space

CRLF => newline

e.g. GET /dashboard/registration.php HTTP/1.1

Describe the status codes that are associated with an invalid request-line?



status-line

HTTP-version **SP** *status-code* **SP** *reason-phrase* **CRLF**

HTTP-version => HTTP/1.1

status-code => 200, 404, 400, ...

reason-phrase => OK, Not Found, Bad Request, ...

e.g. HTTP/1.1 200 OK

=====Classes of Response Codes=====

1xx = (informational) 4xx = (client error)

2xx = (successful) 5xx = (server error)

3xx = (redirection)



header-field

HTTP headers allow the client and the server to pass additional information with the request or the response.

field-name : **OWS** *field-value* **OWS**

field-name => case insensitive name of the header field

field-value => associated value

OWS => Optional White Space

Give examples of header fields?

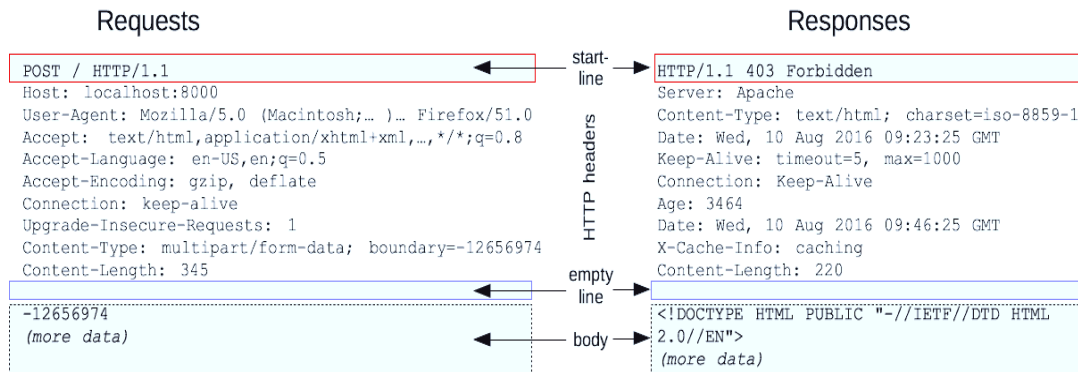


HTTP Message Body

- An optional body contains data associated with the request (like content of an HTML form), or the document associated with a response.



HTTP Message Example



Source: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Messages>

06-Nov-16

Umar Ibrahim Enesi

17



Client-Side Web Technologies (Presentation)

- Client-side technologies includes but are not limited to; (X)HTML, CSS & JavaScript
- These scripts are downloaded from the web server and executed on the client machine.
- On browser-based clients, software called web browsers are used to access information provided by web servers and identified by URLs.
- Browsers understands (X)HTML, CSS, JavaScript among others.
- Browsers incorporates rendering engine and JavaScript interpreter
- Rendering engines renders html and applies included stylesheets.
- JavaScript interpreter executes JavaScript codes.

06-Nov-16

Umar Ibrahim Enesi

18



(X)HTML

- (X)HTML is short for (eXtensible) HyperText Mark-up Language
- (X)HTML is a mark-up language for describing the structure and layout of a web page by using tags and attributes.
- (X)HTML documents are stored in files with .html or .htm extension.
- Browsers rendering engine uses (X)HTML tags to determine how to display its content.
- Versions: HTML, HTML2.0, HTML3.2, HTML4.01, XHTML, HTML5 (latest)

```
<!DOCTYPE html>
<html><head><title>Page Title</title></head><body>
  <h1>Web Application Engineering II</h1>
  <p>Welcome to HTML5.</p>
</body></html>
```



CSS

- CSS is short for Cascading StyleSheet.
- CSS describes the presentation of various tags in a mark-up document
- Browsers rendering engine uses CSS to apply styles to contents of a web page.
- Versions: CSS1, CSS2, CSS3 (latest)
- External stylesheets are CSS stored in files with .css extension. How is it applied?


```
<link rel="stylesheet" type="text/css" href="mystyle.css">
```
- Internal (Embedded) stylesheets are placed in the head element of a web page. How is it applied?

```
<head>
  <style>
    body { background-color: linen;}

    h1 {
      color: maroon;
      margin-left: 40px;
    }
  </style>
</head>
```



CSS (cont.)

- Inline stylesheets are placed as value of the style attribute of an element. How is it applied?

```
<h1 style="color:blue;margin-left:30px;">This is HTML5.</h1>
```

Inline VS Embedded VS External

- Inline:
 - Local effect, difficult to reuse, need to repeat for each tag
 - Difficult to maintain, style scattered around in file
- Embedded:
 - reusable within one HTML document
 - style grouped in HEAD, easier to maintain (than inline)
- External:
 - Clean separation (software engineering)
 - Reusable by different HTML files
 - Centralised maintenance
 - Link via URL possible
 - Cacheable



JavaScript

- JavaScript adds behaviour and interactivity to web pages.
- JavaScript extends the capabilities of web pages.
- JavaScript files/codes are executed by JavaScript interpreters.
- External JavaScript are stored in a file with .js extensions. How is it applied?

```
<script src="myScript.js"></script>
```

- Internal (Embedded) JavaScript are placed in the head or body of web pages. How is it applied?

```
<script>
    alert("Welcome to COSC405");
</script>
```



JavaScript (cont.)

- Inline JavaScript is placed as value of certain event handler attributes of html elements. How is it applied?

```
<button type="button" onclick="alert('Welcome to COSC405');">Greet</button>
```

- *inline VS embedded VS external*



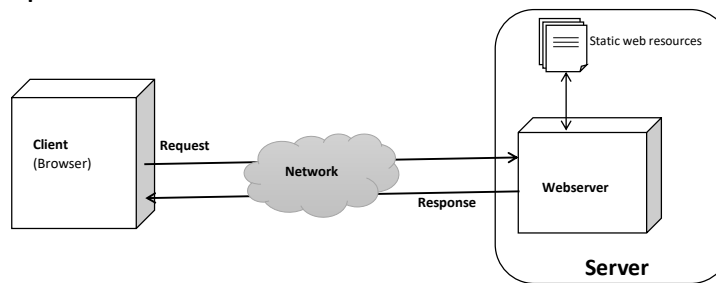
Other client-side technologies

- Adobe Flash
- Java Applets
- XML
- Microsoft Silverlight
- MathML
- KML
- CML
- SVG
- Etc.



Server-Side Web Technologies (Application)

- In a typical client-server architecture;
 - **Client:** this party is responsible for requesting for web resources (eg HTML files)
 - **Server:** this party is responsible for serving web resources (eg HTML files).
- Servers run special server program (web server) that responds to clients' request.



06-Nov-16

Umar Ibrahim Enesi

25



Limitations

- Dynamic content (content based on time or location, user specific information etc).
- Data processing (forms processing, persisting, retrieval and manipulation of data)
- Limited services/interaction (online payment, online booking/reservation, map, currency converter etc)

06-Nov-16

Umar Ibrahim Enesi

26



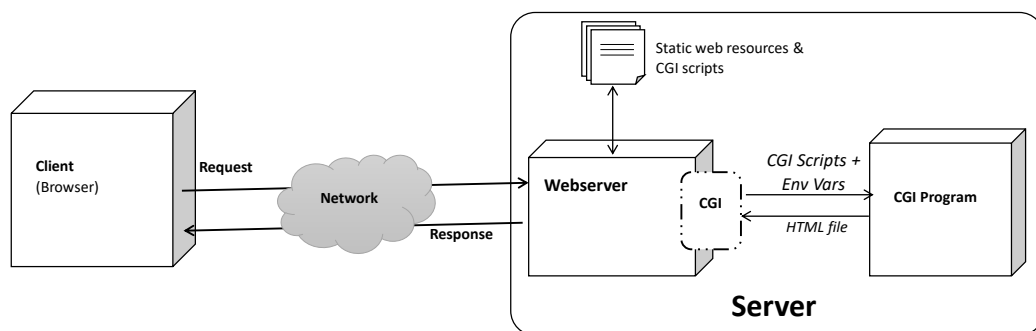
An Advancement- CGI

- CGI extends the capabilities of web servers
- CGI stands for Common Gateway Interface
- Instead of serving static HTML files, web servers can interface with executable programs called (**CGI programs**).
- CGI provides a consistent way for data to be passed from the client's request to CGI program and back to the client.
- CGI programs can generate web pages dynamically if supplied with **CGI script** and other ***environment variables***.
- The Webserver together with CGI programs constitute the Application tier of a 3-tier architecture

06-Nov-16

Umar Ibrahim Enesi

27



06-Nov-16

Umar Ibrahim Enesi

28



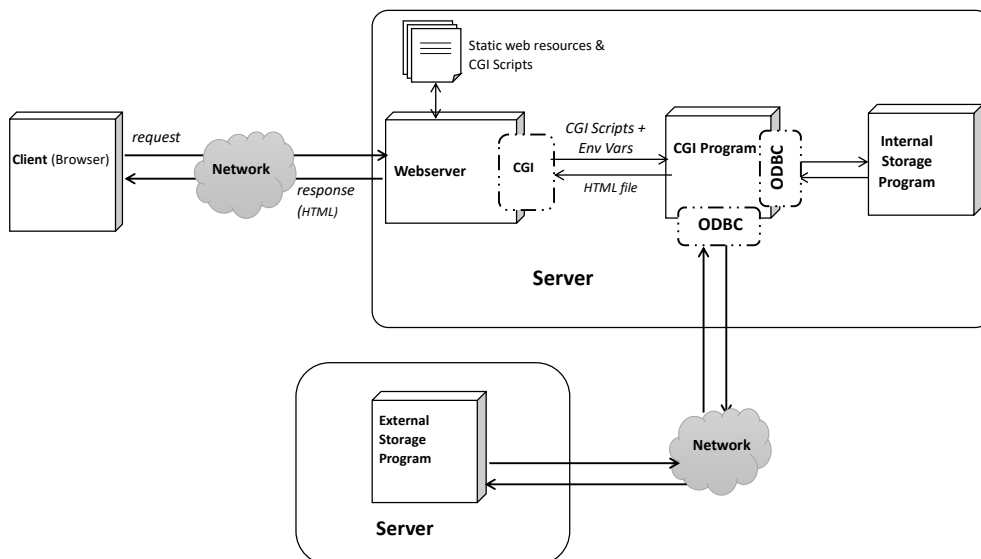
Server-Side Web Technologies (Storage)

- In order to retrieve and persist data into storage, CGI programs work in conjunction with Storage engines.
- CGI programs interact with Storage engines via standard interface called Open Database Connectivity (ODBC) driver.
- The Storage program constitutes the Storage/Data tier of 3-tier architecture.

06-Nov-16

Umar Ibrahim Enesi

29



06-Nov-16

Umar Ibrahim Enesi

30



Web Services

- Another way to extend the capabilities of web servers is by incorporating web services.
- According to W3C;
"web service is a software system designed to support interoperable machine-to-machine interaction over a network."
- Web service is a service that:
 - is available and discoverable over the internet (or intranet)
 - uses XML messaging system (exception JSON web service)
 - is not tied to any OS or programming language
 - Not targeted to end users.
 - has no GUI

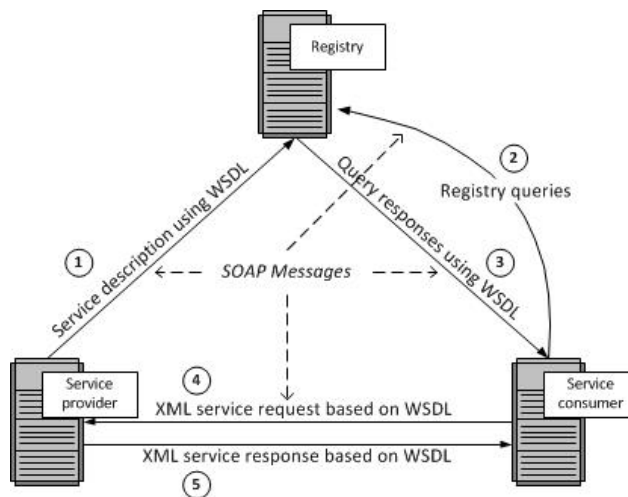


Web Services (cont.)

- Web services Terms:
 - **SOAP**- how to transfer data (protocol)
 - **WSDL**: description of available services (for each provider)
 - **UDDI** : what services are available (universally)
 - **JSON** : in what format is data serialized/transferred
 - **XML**: in what format is data serialized/transferred
 - **Service Provider**: The software system that provides data
 - **Service Requester**: The software system that requests data
 - **REST**: another way of publishing web services



Web Services (cont.)



06-Nov-16

Umar Ibrahim Enesi

33



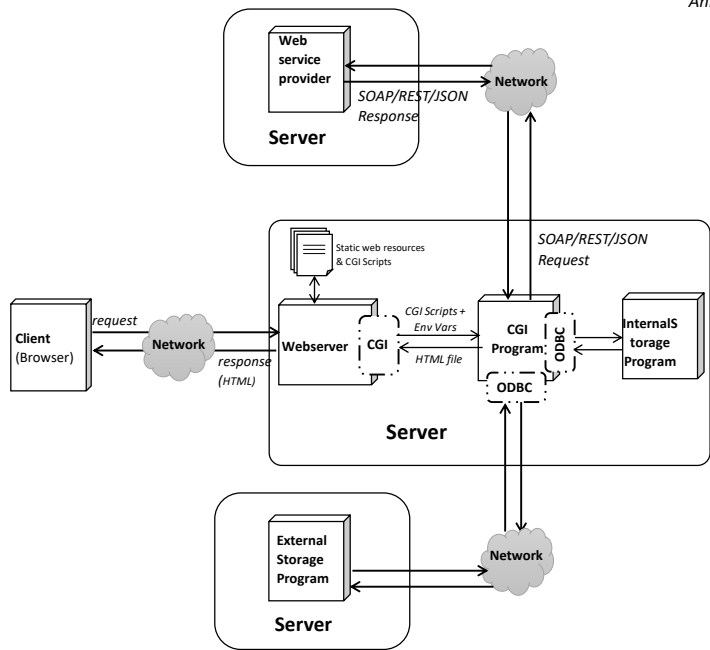
Web Service (cont.)

- A software system is said to **consume** a web service if it uses the web service.
- Examples of web services:
 - Microsoft Dynamics NAV Web services
 - Google Maps API
 - Amazon Elastic Compute Cloud
 - etc

06-Nov-16

Umar Ibrahim Enesi

34



06-Nov-16

35



Examples of CGI Programs

CGI Program	Scripting Language	File extension
Embedded PHP interpreter	PHP	.php
Embedded Perl interpreter	Perl	.pl
Java run-time interpreter	Java	.java
C CGI interpreter	C	.h

Examples of Webservers

Web Server	Creator
Apache HTTP server	Apache Software Foundation
Microsoft IIS	Microsoft
Nginx	NGINX
Oracle HTTP Server	Oracle Corporation
IBM HTTP Server	IBM

06-Nov-16

Umar Ibrahim Enesi

36

Examples of Database Programs

<u>Storage Program</u>	<u>Creator</u>
MySQL	Oracle Corporation
Microsoft Access	Microsoft
PosgreSQL	PostgreSQL Global Development Group
Oracle Database	Oracle Corporation
MongoDB	MongoDB Inc.

Technology Choices

- For this course:
 - **Webserver:** Apache HTTP webserver
 - **CGI Program:** PHP
 - **Data Storage Program:** MySQL

Why?



Installations

- The technology choices can be installed individually and configured to work together.
 - Experience needed
 - Good for production environment setup
- However, there are software that make it extremely easy for developers to create everything needed to make an out-and-out webserver.
 - Eg. XAMPP, WAMP, lighttpd
- For this course XAMPP or WAMP is recommended



Key Points

- www is one among the several ways to utilize the Internet.
- www uses HTTP protocol for communication.
- In logical perspective: 3-tiered architecture.
- In communication perspective: client-server architecture.
- Capabilities of web servers can be extended with technologies like CGI and web services.
- Web servers interact with CGI program via standard interface called CGI.
- CGI programs can interact with databases in order to persist and retrieve data
- Web service can be SOAP, REST, JSON based.
- HTTP is meant for human-machine interaction while Web service is meant for machine-machine interaction



References

- HTML, CSS and Javascript basic tutorials
 - www.w3schools.com
- HTTP, HTML, CSS Javascript and Web Service specifications
 - www.w3.org
- HTTP Request and Response Messages
 - <https://developer.mozilla.org/en-US/docs/Web/HTTP/Messages>
 - <https://tools.ietf.org/html/rfc7231#section-4>
 - <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers>
 - <https://tools.ietf.org/html/rfc7231#section-6>