

**Unit-5**

---

# **Introduction to Web 2.0**

# Content

---

- ✓ Introduction to Web 2.0: Overview of WWW 2.0, JSON, Web Services - SOAP & WSDL, RESTful.
- ✓ Introduction to AJAX: Basics of AJAX, Asynchronous and Synchronous message transformation.
- ✓ Future of Web: Overview of Semantic Web, Applications of Semantic Web, Virtual Reality, Web OS.

# **JSON with PHP**

---

# JSON

---

- Data interchange format that serves as a text-based alternative to XML.
- Humans find it simple to read and write, and machines find it easy to parse and generate.
- It is a textual way to represent objects by using two structures: collections of *name-value* pairs and arrays of values.
- JSON is a way to represent JavaScript objects as strings.
- Objects are unordered sets of *property-value* pairs.
- The *property-value* pairs in an object are separated by commas.
- It is an alternative to the XML for returning data from the server in response to an Ajax request.
- Why JSON?
  - Reason to use JSON instead of XML is to eliminate the complexity of parsing.

# JSON

---

## → Json Types

Type	Example
1. Strings	“SDMCET”, “Dharwad”, “S”
2. Numbers	22, 0.2, -100, 3.6e <sup>10</sup>
3. Boolean	True, False
4. null	null
5. Arrays	[11,22,33], [“SDM”, “Mahesh”]
6. Objects	{“Key/Name”: “Value”} {“Gender”: “Male”}

# JSON

---

→ Json file: Example – studentInfo.json

```
{  
    "name": "Mohan",  
    "USN" : "2SD20CS002",  
    "Branch": "CSE",  
    "Semester": 3,  
    "ClearedAllSubj": true,  
    "Programs_familiar": ["C", "Java", "Python", "C#"]  
}
```

# PHP with JSON

---

- A common use of JSON is to read data from a web server and display it on a web page.
- Objective:
  - Encoding and Decoding of JSON objects using PHP.
  - `json_decode`- Decodes a JSON string
  - `json_encode` - Returns the JSON representation of a value.
  - It uses an associative array

# PHP with JSON

---

## → JSON Encode

→ In PHP, `json_encode()` is used to convert PHP-supported data type into JSON formatted string to be returned as a result of JSON encode operation.

## → JSON Decoding

→ This is the reverse operation of JSON encoding, obviously used to convert JSON encoded data into its original PHP data type from where it is encoded initially.

→ It converts a JSON string into a PHP variable.

→ The resulting PHP variable can be an array or an object, depending on the JSON structure.

# Contd...

---

Example: json\_encode

```
<?php  
$arrayObj = array("College"=>"SDM", "Code" =>"2sd","location"  
                  =>"Dharwad");  
echo json_encode($arrayObj);  
?>
```

Json\_decode:

```
<?php  
$j_obj = '{"college":"sdm" , "Code" : "2sd", "place" : "Dharwad"}';  
  
var_dump(json_decode($j_obj));  
?>
```

The var\_dump() function is used to display structured information (type and value) about one or more variables.

# Example

---

```
{  
    "movies": [  
        {  
            "title": "abc",  
            "Year": "2003",  
            "genre": "Horror",  
            "director" : "asdfsdf"  
        },  
        {  
            "title": "adasd",  
            "Year": "2004",  
            "genre": "rtyret",  
            "director" : "jkghjhgj"  
        },  
        {  
            "title": "abc",  
            "Year": "2003",  
            "genre": "Horror",  
            "director" : "asdfsdf"  
        }  
    ]  
}
```

# Contd...

---

```
<?php
    $jsondata = file_get_contents("movies.json");
    $json = json_decode($jsondata, true);
    $output = "<ul>";
    foreach($json["movies"] as $movie){
        $output .= "<h4>" . $movie["title"] . "</h4>";
        $output .= "<li>Year:" . $movie["Year"] . "</li>";
        $output .= "<li>Genre:" . $movie["genre"] . "</li>";
        $output .= "<li>Director:" . $movie["director"] . "</li>
                    ";
    }
    $output .= "</ul>";
    echo $output;
?>
```

# **Web Services**

---

# Why Web Services?

---

## → Exposing the Existing Function on the network

-> A web service is a unit of managed code that can be remotely invoked using HTTP, that is, it can be activated using HTTP requests.

-> Web services allows you to expose the functionality of your existing code over the network.

-> Once it is exposed on the network, other application can use the functionality of your program.

## → Reusable application-components.

Example: Currency conversion,

Weather reports,

Language translation as services.

# Contd.

---

## → Interoperability

- > Web services allow various applications to talk to each other and share data and services among themselves.
- > Other applications can also use the web services.
- > Example:
  - > A VB or .NET application can talk to *Java web services* and vice versa.
  - > Web services are used to make the application platform and technology independent.

## → Standardized Protocol

- > Web services uses standard protocol for the communication.  
Example:  
A web service can use SOAP over HTTP protocol for communication.

# Definitions:

---

→ A web service is any piece of software that makes itself available over the internet and uses a standardized XML messaging system. XML is used to encode all communications to a web service.

Example: a client invokes a web service by sending an *XML message*, then waits for a corresponding XML response.

→ Web services are not tied to any one operating system or programming language.

Example: Java application service can talk with Perl application; Windows applications can talk with Unix applications.

## Cont'd..

---

➤ Web services are XML-based information exchange systems that use the Internet for direct application-to-application interaction. These systems can include programs, objects, messages, or documents.

[Or]

➤ Web services are client and server applications that communicate over the World Wide Web's (WWW) HyperText Transfer Protocol (HTTP).

[Or]

➤ *Web Services* refers to the technologies that allow for making connections.

# **API Vs. Web Service**

---

**API and Web service serve as a means of communication.**

# Cont'd..

API	Web Services
An API acts as an interface between two different applications so that they can communicate with each other.	Web service facilitates interaction between two machines over a network.
An API is a method by which the third-party vendors can write programs that interface easily with other programs.	A Web service is designed to have an interface that is depicted in a machine-processable format usually specified in Web Service Description Language (WSDL).
API may use any means of communication to initiate interaction between applications.  Ex: the system calls are invoked using interrupts by the Linux kernel API.	Web service also uses SOAP, REST, and XML-RPC as a means of communication.

## Cont'd..

---

- Desktop applications such as spreadsheets and word documents use VBA and COM-based APIs which don't involve Web service.
- The APIs can be exposed in a number of ways which include: COM objects, DLL and .H files in C/C++ programming language, JAR files or RMI in Java, XML over HTTP, JSON over HTTP, etc.
- The method used by Web service to expose the API is strictly through a network.

## Cont'd..

---

→ All Web services are APIs but all APIs are not Web services.

Ex:-

1. Weather report service, Google Map APIs, etc. are web-based APIs whereas system calls invoked by the APIs are not Web services.
2. MySQL connector acts like an API in Java applications.

→ A Web service uses only - SOAP, REST, and XML-RPC for communication

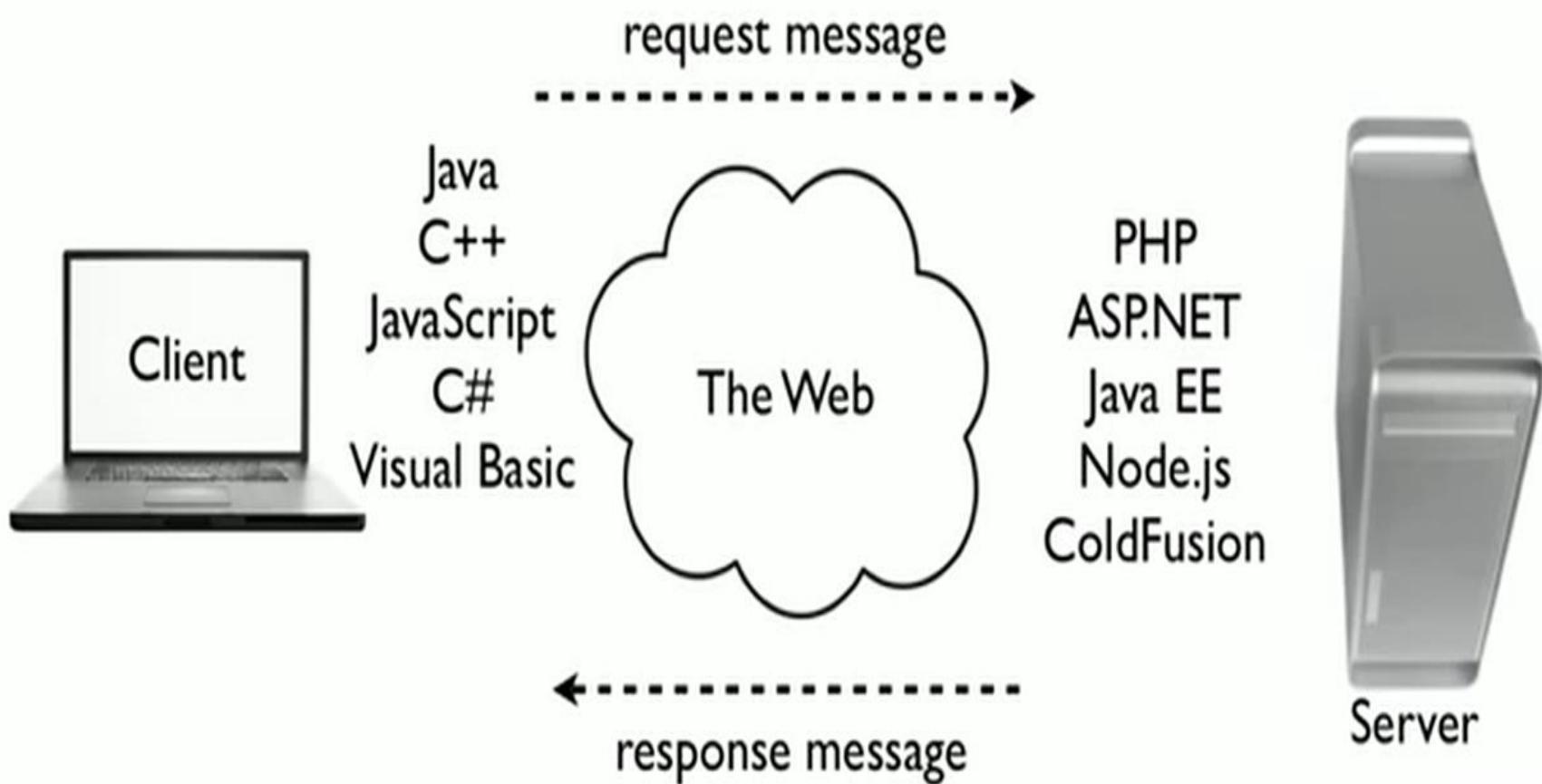
→ A Web service always needs a network for its operation whereas an API doesn't need a network for its operation.

# Cont'd..

---

Feature	APIs	Web Services
Scope and Usage	Broader, used in various contexts	Specific to web-based communication
Communication Protocols	Multiple protocols (HTTP, TCP/IP, etc.)	Primarily HTTP/HTTPS, SOAP, REST
Standards and Conventions	Flexible, no strict standards	Follows strict standards (SOAP) or principles (REST)

# Client and Server Languages



## Cont'd..

---

- Web services are application components
- Web services communicate using open protocols
- Web services can be used by other applications
- HTTP and XML is the basis for Web services
- Interoperability has Highest Priority in web service.

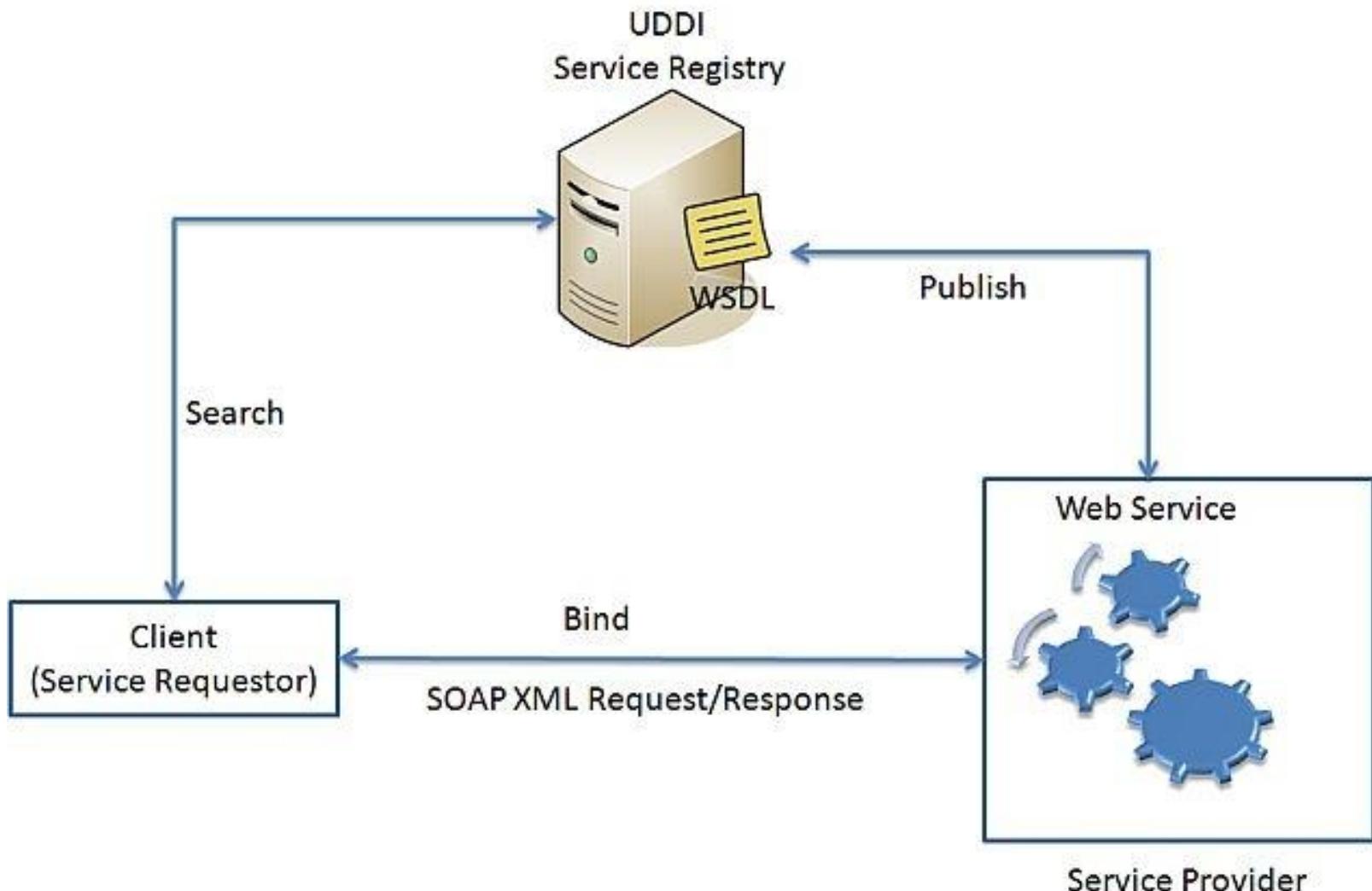
# Components of web services

---

- UDDI (Universal Description, Discovery and Integration)
- WSDL (Web Services Description Language)
- SOAP (Simple Object Access Protocol)

## Contd.

---



# Web Services Architecture

## Service Discovery (UDDI)

Service Discovery: This part of the architecture is responsible for centralizing services into a common registry and providing easy publish/search functionality.

## Service Description (WSDL)

Service Description: Web services are self-describing. This means that, once a Web Service is located, it will let us know what operations it supports and how to invoke it.

## Service Invocation (SOAP)

Service Invocation: Invoking a Web Service involves passing messages between the client and the server. SOAP specifies how we should format request messages to the server, and how the server should format its response messages.

## Service Transport (HTTP)

Transport: Finally, all these messages must be transmitted somehow between the server and the client. HTTP protocol used to access conventional web pages on the Internet. We could also use other protocols, but HTTP is currently the most used one.

# 1. UDDI

---

- ✓ UDDI (Universal Description, Discovery and Integration) is a platform-independent, XML based registry service where companies can register and search for Web services.
  - ✓ UDDI is a directory for storing information about web services
  - ✓ UDDI communicates via SOAP
  - ✓ UDDI is a directory of web service interfaces described by WSDL

## 2. WSDL

---

- ✓ WSDL (Web Services Description Language) is an XML-based language for locating and describing Web services.
- ✓ WSDL describes how to access a web service and what operations it will perform.
- ✓ WSDL is a W3C standard.

# Contd.

---

→ WSDL mainly contains two components:

Definitions: root element in WSDL

Types: define all datatypes, or import xml scheme,  
takes user name

Messages: return types, I/p msg, O/p Msg

Operations: methods in our programming

Porttypes: collection of operations

What ?

Binding name: links to abstract section, ie. Specify  
port types.

Service: Url , or how this web service can be  
consumed.

How?

### **3. SOAP**

---

- ✓ SOAP (Simple Object Access Protocol) is an XML-based communication protocol for exchanging structured information between applications over HTTP, SMTP or any other protocol.
- ✓ In other words, SOAP is a protocol for accessing a Web Service.

## Contd.

---

- ✓ A SOAP message is an ordinary XML document containing the following elements:
  - An Envelope (required) element that identifies the XML document as a SOAP message
  - An optional Header element that contains client application can authenticate web service using username and password.
  - A Body (required) element that contains call and response information
  - An optional Fault element containing errors and status information

# Contd.

---

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-
envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Header>
...
</soap:Header>

<soap:Body>
...
<soap:Fault>
...
</soap:Fault>
</soap:Body>

</soap:Envelope>
```

# Sample SOAP Request Message

POST /stockquote.asmx HTTP/1.1  
Content-type: text/xml; charset="utf-8"  
Soapaction: "<http://www.webserviceX.NET/GetQuote>"  
Accept: text/xml, multipart/related, text/html, image/gif, image/jpeg, \*; q=.2, \*/\*;  
q=.2  
User-Agent: JAX-WS RI 2.1.6 in JDK 6  
Host: localhost  
Connection: keep-alive  
Content-Length: 194

```
<?xml version="1.0" ?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <GetQuote xmlns="http://www.webserviceX.NET/">
            <symbol>GOOG</symbol>
        </GetQuote>
    </S:Body>
</S:Envelope>
```

# Sample SOAP Response message

---

```
HTTP/1.1 200 OK
Cache-Control: private, max-age=0
Content-Length: 975
Content-Type: text/xml; charset=utf-8
Server: Microsoft-IIS/7.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Wed, 30 May 2012
09:14:05 GMT
```

# Contd.

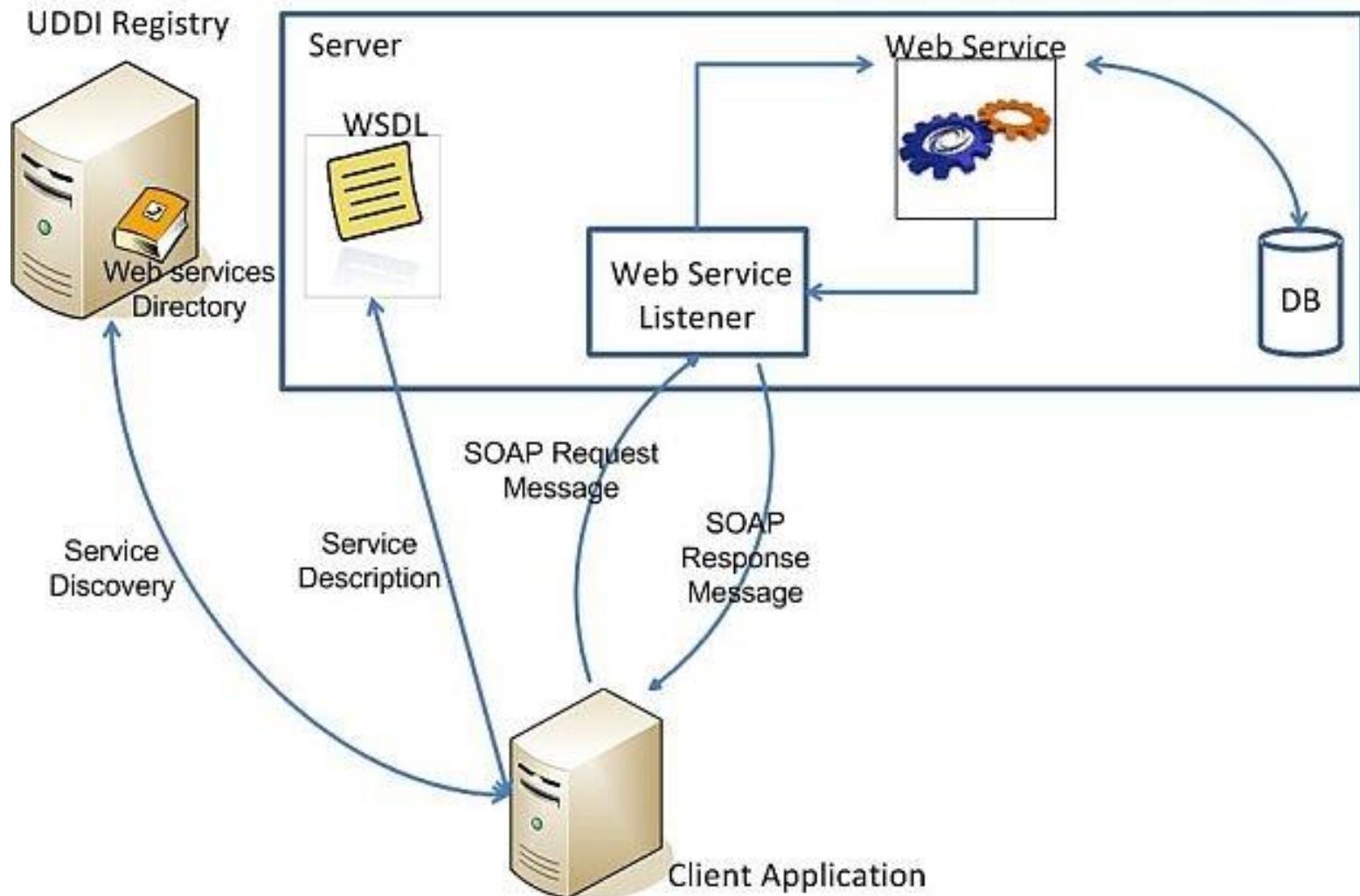
```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <soap:Body>
        <GetQuoteResponse xmlns="http://www.webserviceX.NET/">
            <GetQuoteResult>
                <StockQuotes>
                    <Stock>
                        <Symbol>GOOG</Symbol>
                        <Last>594.34</Last>
                        <Date>5/29/2012</Date>
                        <Time>4:00pm</Time>
                        <Change>0.00</Change>
                        <Open>N/A</Open>
                        <High>N/A</High>
                        <Low>N/A</Low>
                    </Stock>
                </StockQuotes>
            </GetQuoteResult>
        </GetQuoteResponse>
    </soap:Body>
</soap:Envelope>
```

# Contd.

---

```
<Volume>100</Volume>
<MktCap>193.8B</MktCap>
<PreviousClose>594.34</PreviousClose>
<PercentageChange>0.00%</PercentageChange>
<AnnRange>473.02 - 670.25</AnnRange>
<Earns>32.998</Earns>
<P-E>18.01</P-E>
<Name>Google Inc.</Name>
</Stock>
</StockQuotes>
</GetQuoteResult>
</GetQuoteResponse>
</soap:Body>
</soap:Envelope>
```

# How Web Services work?



## Contd.

---

- The Service Provider generates the WSDL describing the application or service and registers the WDSL in UDDI directory or Service Registry.
- The Service Requestor or client application which is in need of web service contacts the UDDI and discovers the web service.
- The client based on the web service description specified in the WSDL sends a request for a particular service to the web service application listener in SOAP message format.
- The web service parses the SOAP message request and invokes a particular operation on the application to process that particular request. The result is packed in an appropriate SOAP response message format and sent to the client.
- The client parses the SOAP response message and retrieves the result or error messages if any.

# **REST & RESTful WEB SERVICES**

---

Representational State Transfer

# Applications

---

- ❖ **Twitter API:** Twitter provides a RESTful API to interact with their platform, allowing **operations** like **posting tweets**, **reading user timelines**, and more, all through standard HTTP methods and URLs.
- ❖ **OpenWeatherMap API:** Provides weather data and forecasts through RESTful endpoints, allowing developers to **integrate weather information into their applications**.
- ❖ **Facebook Graph API:** Enables developers to **retrieve** and **post data** to the **Facebook social graph**, including user profiles, photos, videos, and more.
- ❖ **Google Maps API:** Provides **RESTful services** to **access maps**, **geolocation data**, and other **location-based services**.
- ❖ **GitHub API:** GitHub's API allows **developers** to **interact with** the **GitHub platform**.

# Introduction

---

- ❖ REST is about **resources** and how to **represent resources** in different ways.
- ❖ REST is about **client-server** communication.
- ❖ REST is about **how to manipulate resources**.
- ❖ REST **offers** a **simple, interoperable** and **flexible** way of writing web services that can be very different from other techniques.
- ❖ Comes from Roy Fielding's Thesis study.

# **REST is NOT!**

---

- ❖ A protocol.
- ❖ A standard.
- ❖ A replacement for SOAP.

# REST

---

REST does not enforce any rule regarding how it should be implemented at lower level, it just put high level design guidelines and leave you to think of your own implementation.

- ❖ Architectural style
- ❖ Idea: a network of web pages where the client progresses through an application by selecting links
- ❖ When client traverses link, accesses new resource (i.e., transfers state)
- ❖ Uses existing standards, e.g., HTTP
- ❖ REST is an architecture all about the Client-Server communication.
- ❖ They also have some constraints or principles.
  - ❖ *Client-Server*
  - ❖ *Stateless*
  - ❖ *Cacheable*
  - ❖ *Uniform Interface*

# An Architectural Style

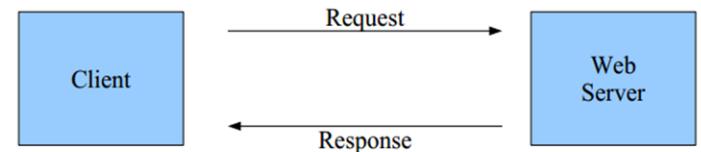
---

- An architectural style is constrained by a particular principle of how to build systems
- How components are organized, how data is manipulated, how components communicate, ...?
- E.g. a hierarchy of components, sequential processing of data, asynchronous communication, ...
- Each principle influences some quality attributes in a positive and some other in a negative way
- It is again balancing between advantages and disadvantages

# Client-Server

---

- ❖ Client and server are independent of each other.
- ❖ Client doesn't know anything about the resource which is kept on the server.
- ❖ Server responds as long as the right requests come in.
- ❖ Goal: Platform independency and to improve scalability.



# Stateless

---

- ❖ Each request is independent from other requests.
- ❖ No client session data or any context stored on the server.
- ❖ Every request from client stores the required information, so that the server can respond.
- ❖ If there is a need for session-specific data, it should be held and maintained by the client and transferred to the server with each request as needed.
- ❖ A service layer which doesn't have to maintain client sessions is much easier to scale.

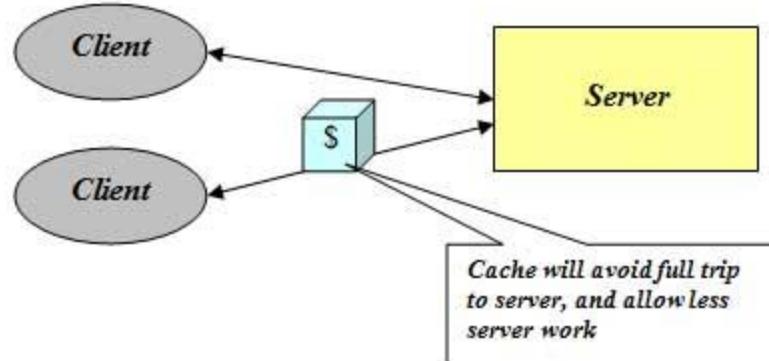
Scalability



# Cacheable

---

- ❖ HTTP responses must be cacheable by the clients.
- ❖ Important for performance.
- ❖ If a new request for the resources comes within a while, then the cached response will be returned.



# Uniform Interface

---

- ❖ All resources are accessed with a generic interface (HTTP-based).
- ❖ This makes it easier to manage the communication.
- ❖ By the help of a uniform interface, client and server evolve independently from each other.

# Uniform Interface

---

- Defines the interface between client and server
- Simplifies and decouples the architecture
- Fundamental to RESTful design
- For us this means:
  - HTTP verbs (GET, PUT, POST, DELETE)
  - URIs (resource name)
  - HTTP response (status, body)

# RESTful Web Services

---

- ❖ RESTful web services are web services which are REST based.
- ❖ Stateless & cacheable.
- ❖ Uses URI & HTTP methods.
- ❖ Quiet light, extensible and simple services.
- ❖ The reason behind the popularity of REST is that the applications we use are browser-based nowadays and top of it all, REST is built on HTTP.
- ❖ Main idea: Providing the communication between client and server over HTTP protocol rather than other complex architectures like SOAP and RPC etc.

# Contd.

---

- ❖ No severe restrictions on how the architectural model will be and what properties it will have.
- ❖ Models like SOAP have severe rules, REST does not.
- ❖ There are lots of frameworks to develop RESTful web services on platforms like C# and Java, but you can write one easily using some standard libraries.

# Contd.

---

- ❖ Platform independent.
- ❖ Language independent.
- ❖ Work on HTTP protocol.
- ❖ Flexible and easily extendible.
- ❖ They also have some constraints or principles.
  - ❖ *Client-Server*
  - ❖ *Stateless*
  - ❖ *Cacheable*
  - ❖ *Uniform Interface*

# REST Characteristics

---

- ❖ Resources: Application state and functionality are abstracted into resources.
  - ❑ URI: Every resource is uniquely addressable using URIs.
  - ❑ Uniform Interface: All resources share a uniform interface for the transfer of state between client and resource, consisting of
    - Methods: Use only HTTP methods such as GET, PUT, POST, DELETE, HEAD
    - Representation
- ❖ Protocol (The constraints and the principles)
  - ❑ Client-Server
  - ❑ Stateless
  - ❑ Cacheable

# Representations

---

- How resources get manipulated
- Part of the resource state
  - Transferred between client and server
- Typically JSON or XML
- Example:
  - Resource: person (Todd)
  - Service: contact information (GET)
  - Representation:
    - name, address, phone number
    - JSON or XML format

# HTTP Methods

---

- ❖ GET - *safe, idempotent, cacheable*
- ❖ PUT - *idempotent*
- ❖ POST
- ❖ DELETE - *idempotent*
- ❖ HEAD
- ❖ OPTIONS

# CRUD Operations Mapped to HTTP Methods in RESTful Web Services

---

OPERATION	HTTP METHOD
Create	POST
Read	GET
Update	PUT or POST
Delete	DELETE

# REST vs SOAP

---

- ❖ SOAP is a XML-based message protocol, while REST is an architectural style.
- ❖ SOAP uses WSDL for communication between consumer and provider, whereas REST uses XML or JSON to send and receive data.
- ❖ SOAP invokes services by calling RPC method, REST simply calls services via URL path.
- ❖ SOAP doesn't return human readable result, whilst REST result is readable which is just plain XML or JSON.
- ❖ SOAP is not just over HTTP, it also uses other protocols such as SMTP, FTP, etc, REST is over only HTTP.

# Introduction to AJAX

---

Asynchronous JavaScript and XML

# What is AJAX?

---

- ✓ AJAX is an acronym for Asynchronous JavaScript and XML.
- ✓ It is a group of inter-related technologies like javascript, dom, xml, html, css etc.
- ✓ AJAX allows you to send and receive data **asynchronously without reloading** the **web page**.
- ✓ AJAX is the **art** of **exchanging data with a server**, and **updating parts of a web page - without reloading the whole page**.
- ✓ It makes your application interactive and faster.

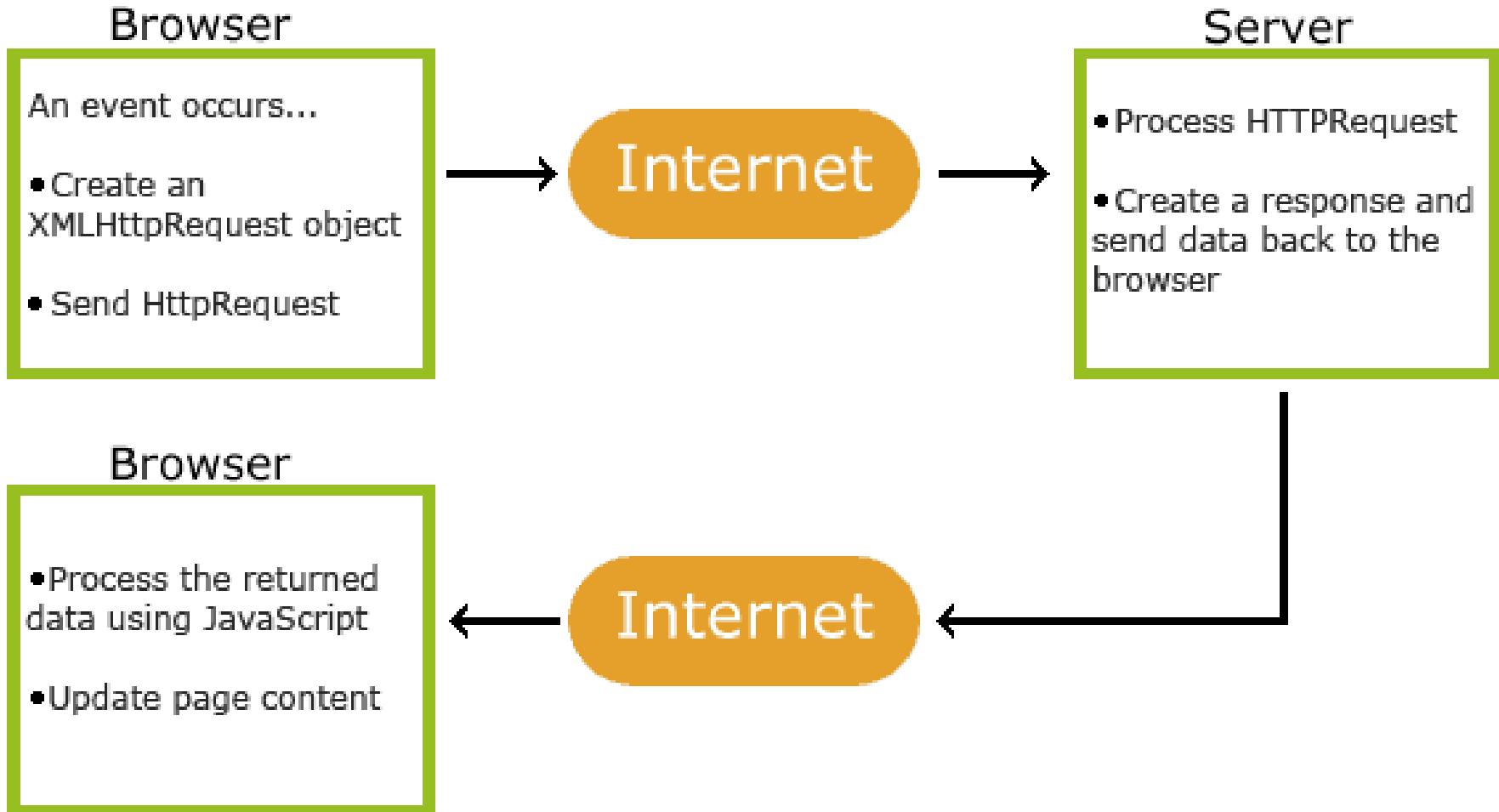
# Where it is used?

---

- ✓ There are too many web applications running on the web that are using ajax technology like
  - Gmail
  - Facebook
  - Twitter
  - Google map
  - Youtube
  - Crickinfo

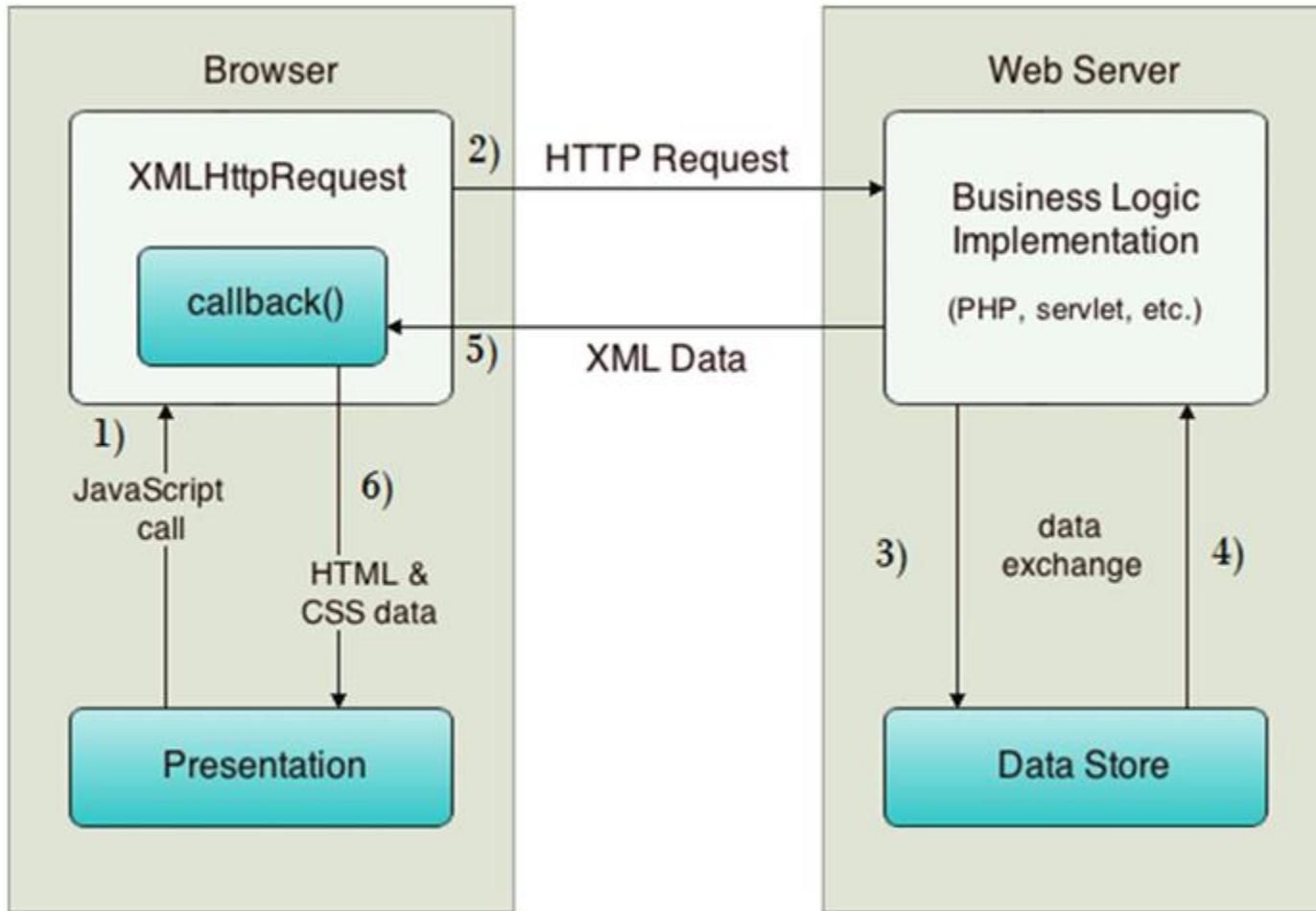
# How AJAX Works

---



# Contd.

→ AJAX communicates with the server using an XMLHttpRequest object.



# Contd.

---

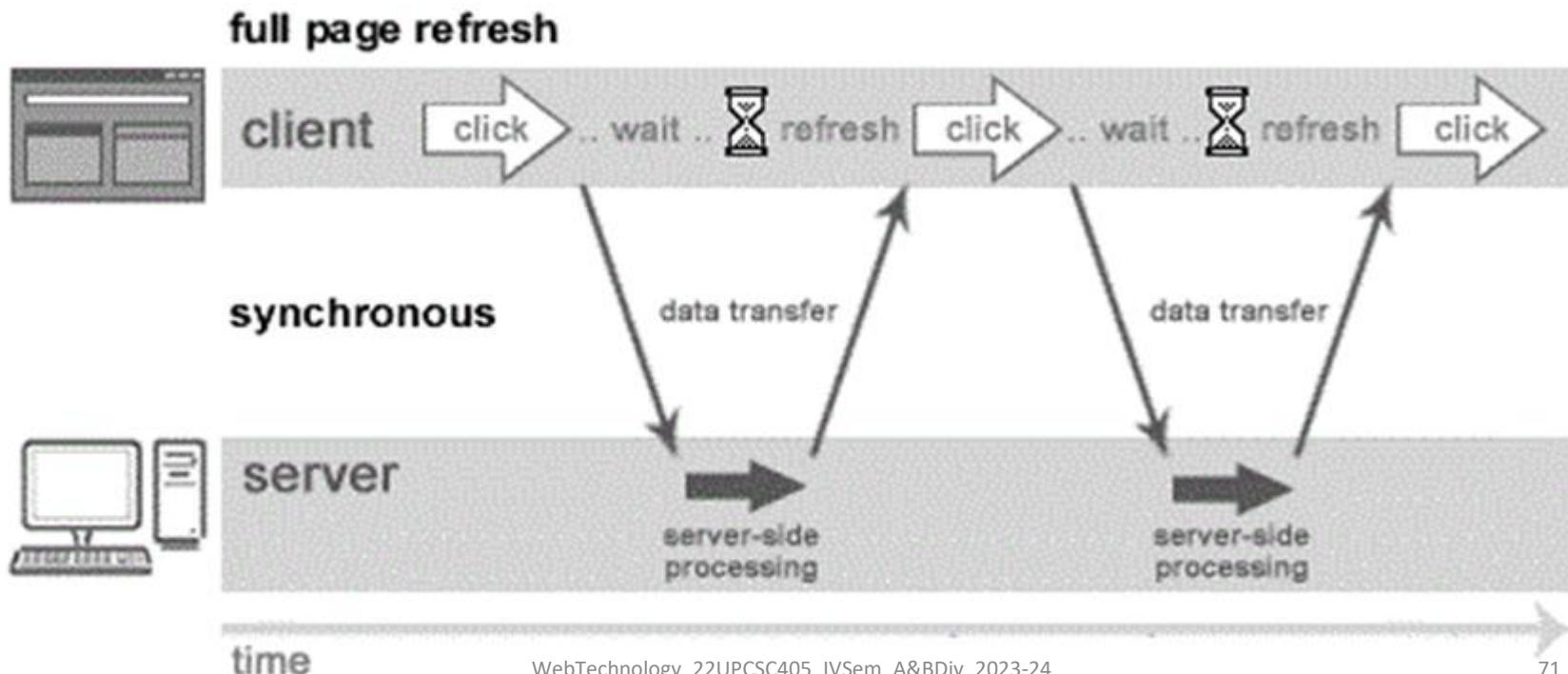
- ✓ User sends a request from the UI and a JavaScript call goes to the XMLHttpRequest object.
- ✓ HTTP Request is sent to the server by XMLHttpRequest object.
- ✓ Server interacts with the database using JSP, PHP, Servlet, ASP.net etc.
- ✓ Data is retrieved.
- ✓ Server sends XML data or JSON data to the XMLHttpRequest callback function.
- ✓ HTML and CSS data is displayed on the browser.

# Synchronous Vs Asynchronous

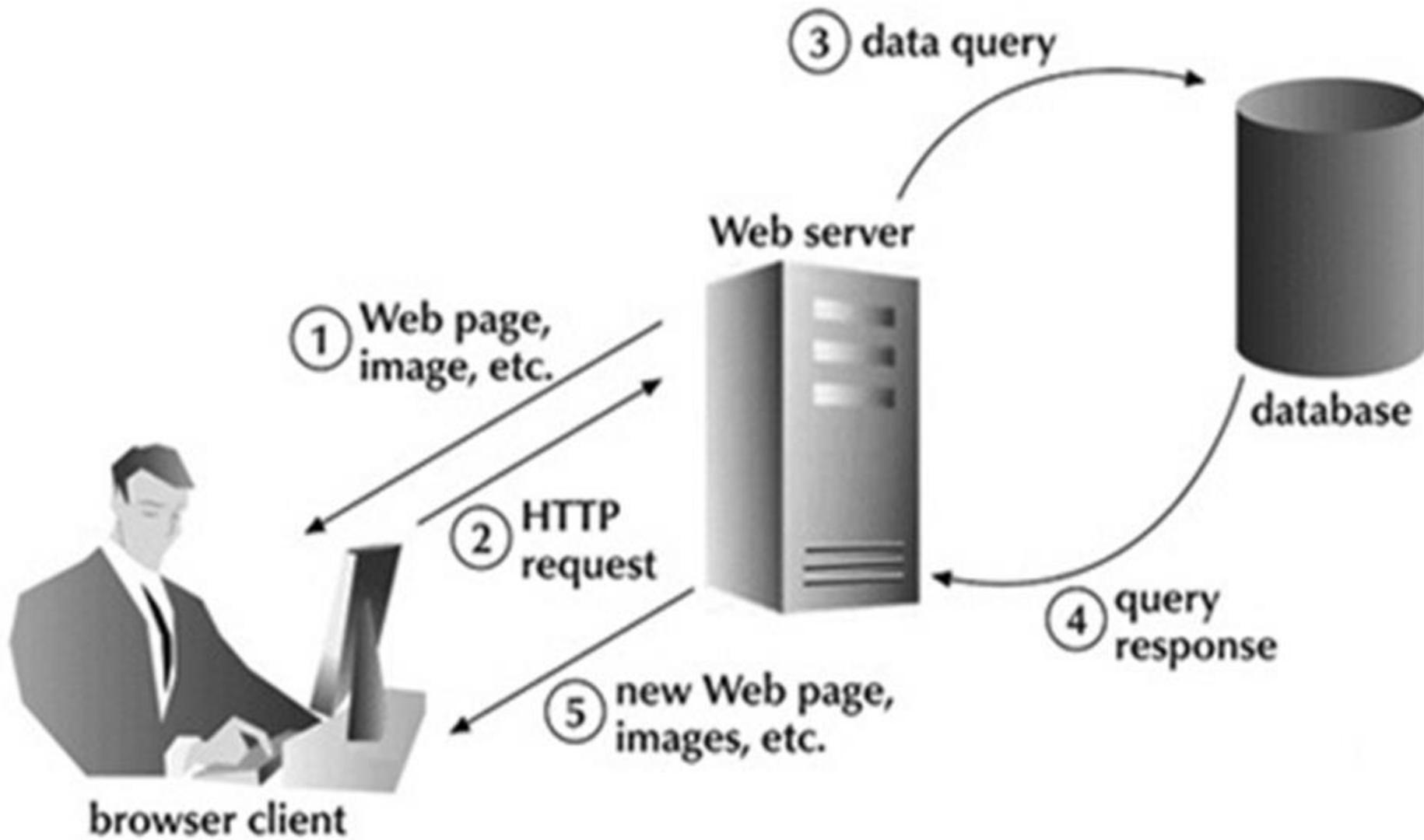
---

→ Synchronous (Classic Web-Application Model)

- ✓ A synchronous request blocks the client until operation completes.
- ✓ Browser is unresponsive
- ✓ Full page is refreshed at request time and user is blocked until request completes.



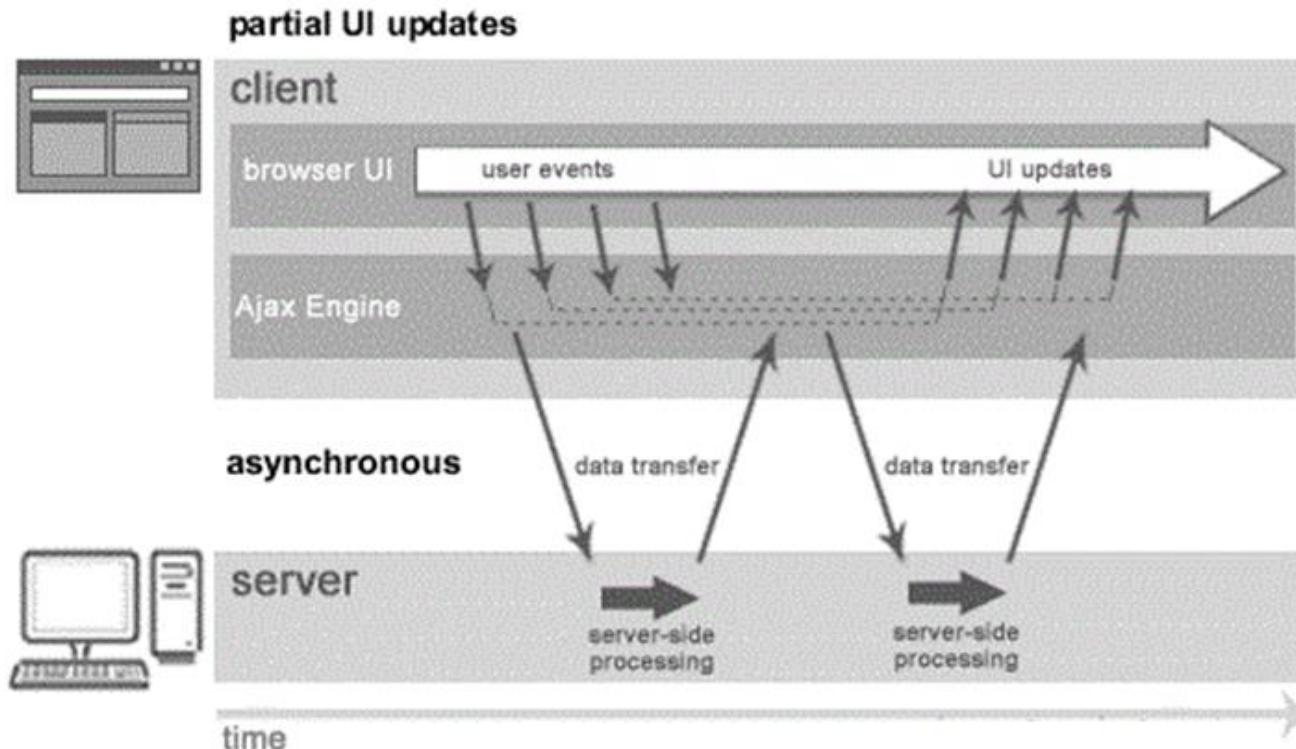
# Synchronous-Classic Web-Application Model



# Contd.

## → Asynchronous (AJAX Web-Application Model)

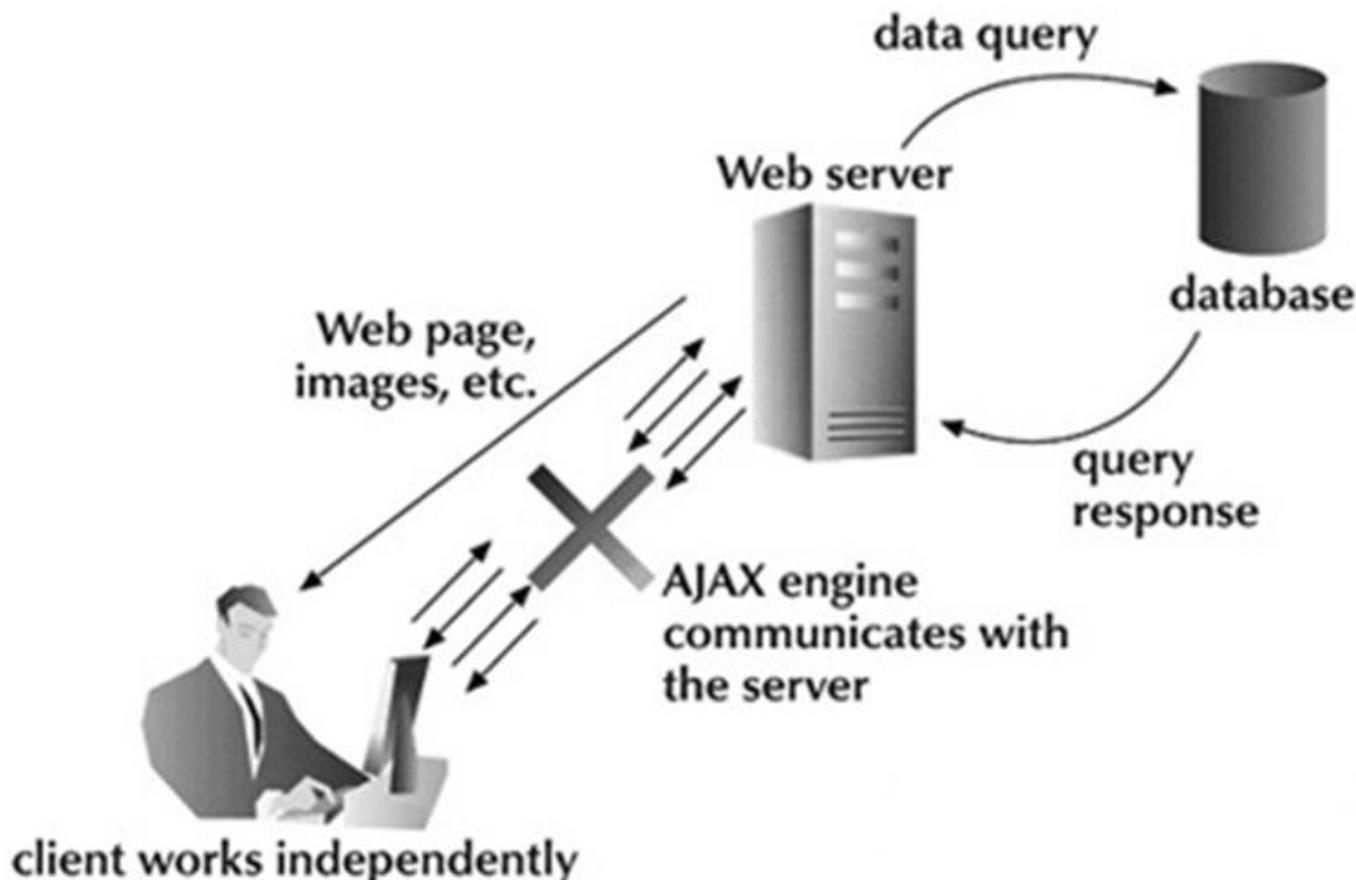
→ An asynchronous request doesn't block the client i.e. browser is responsive. At that time, the user can perform other operations also.



## Contd.

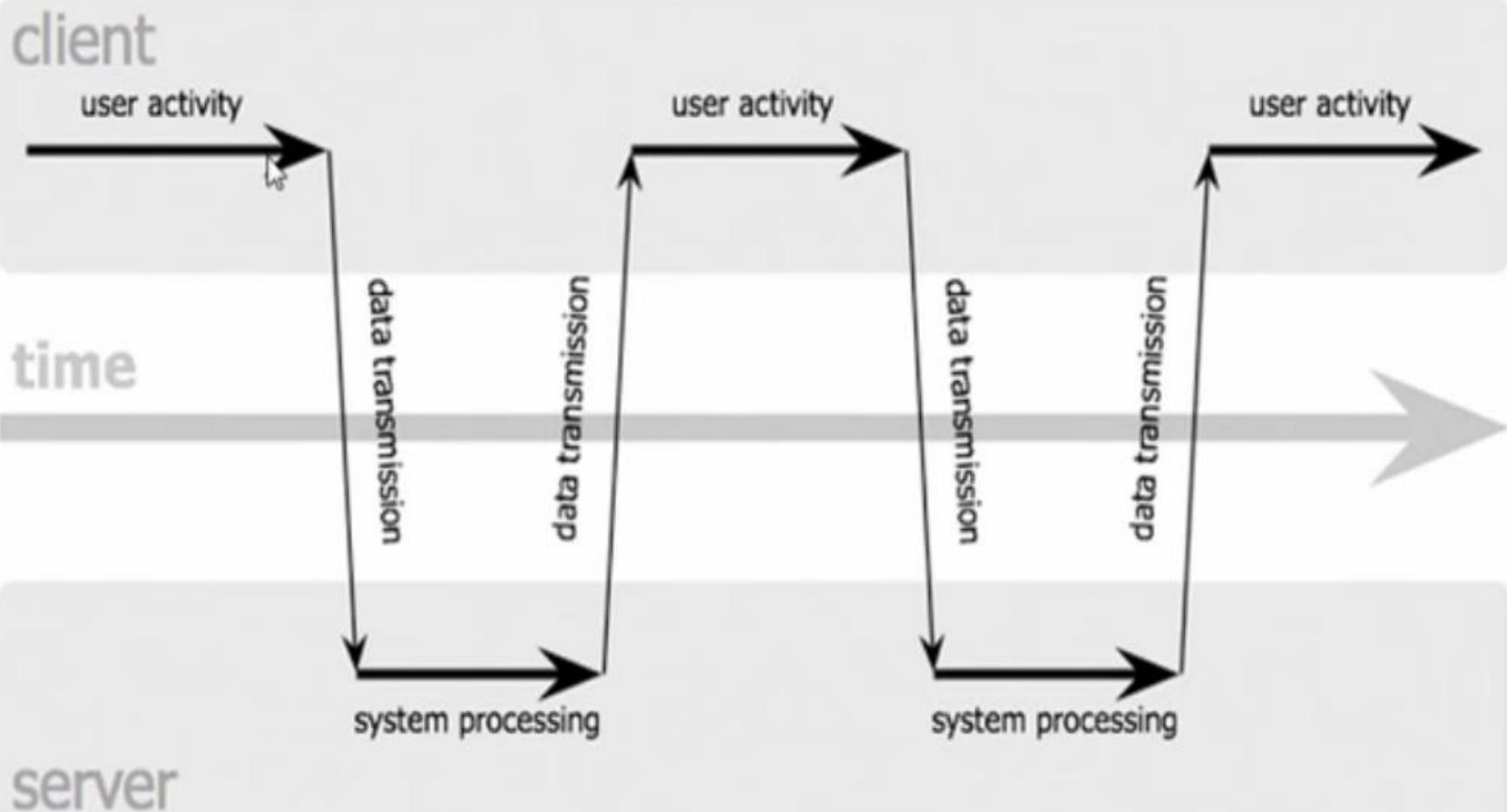
---

- Full page is not refreshed at request time and user gets response from the Ajax engine.



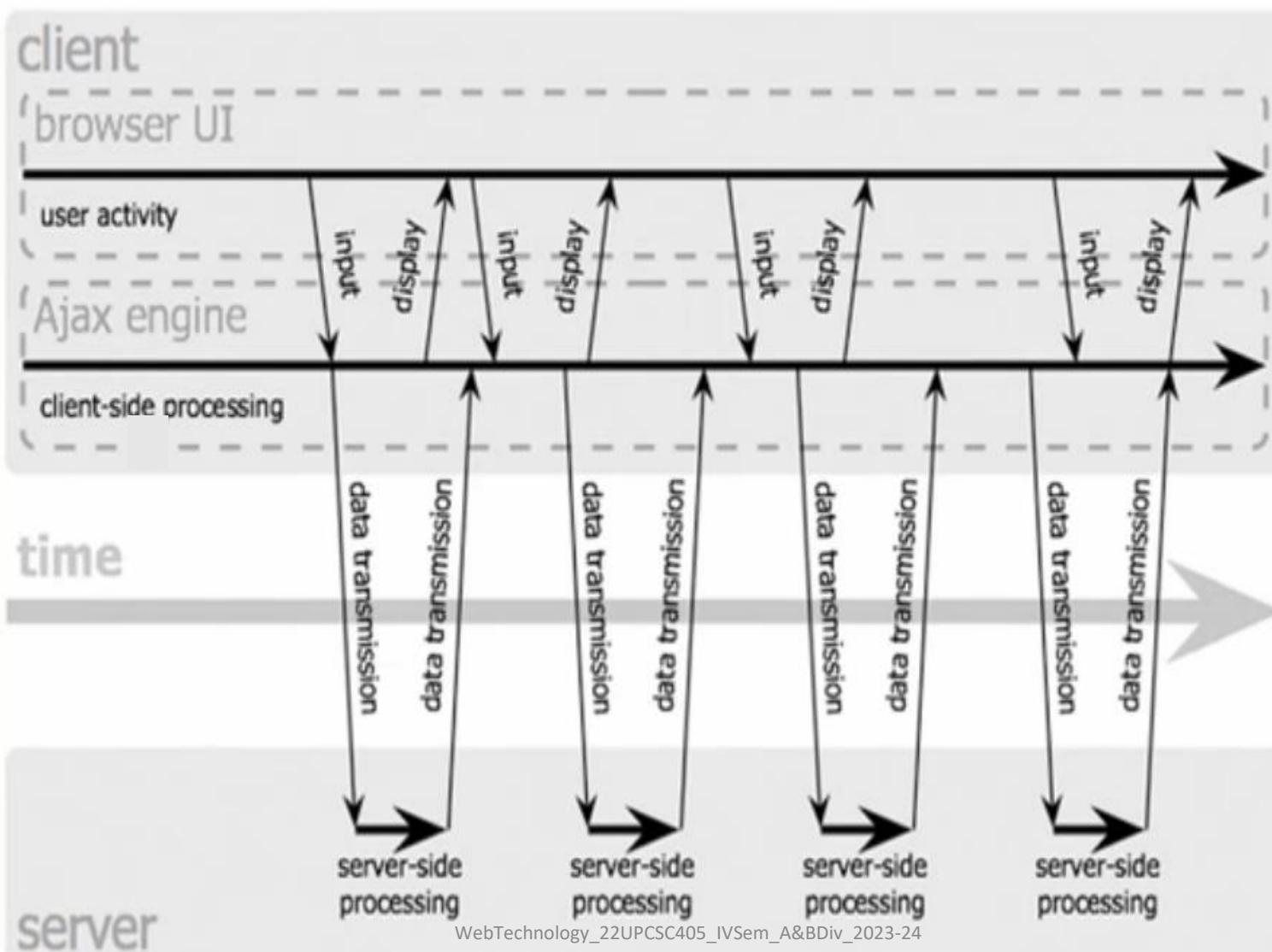
# BEFORE AJAX

classic web application model (synchronous)



# AFTER AJAX (Async Javascript & XML)

## Ajax web application model (asynchronous)

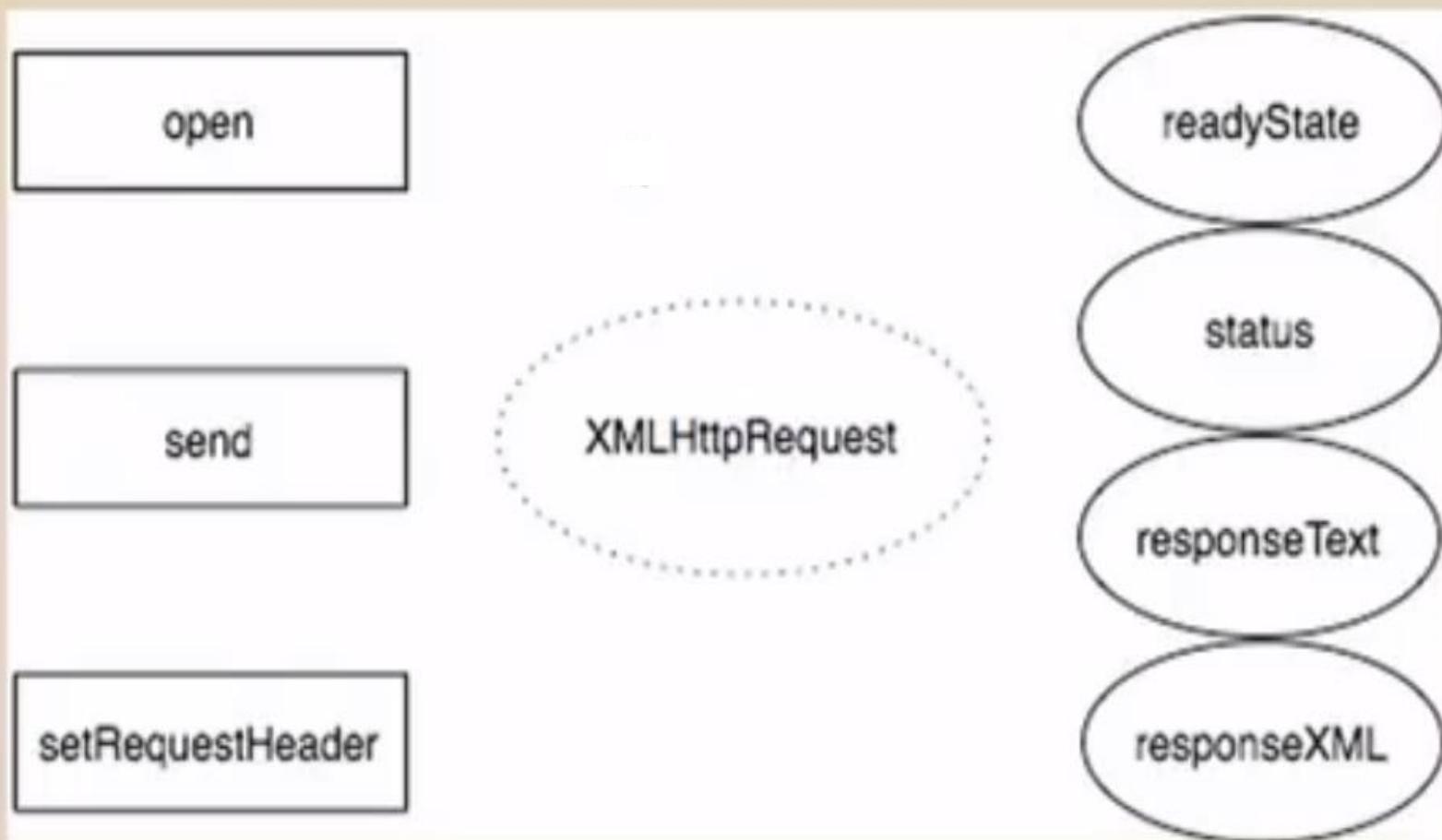


# XMLHttpRequest object

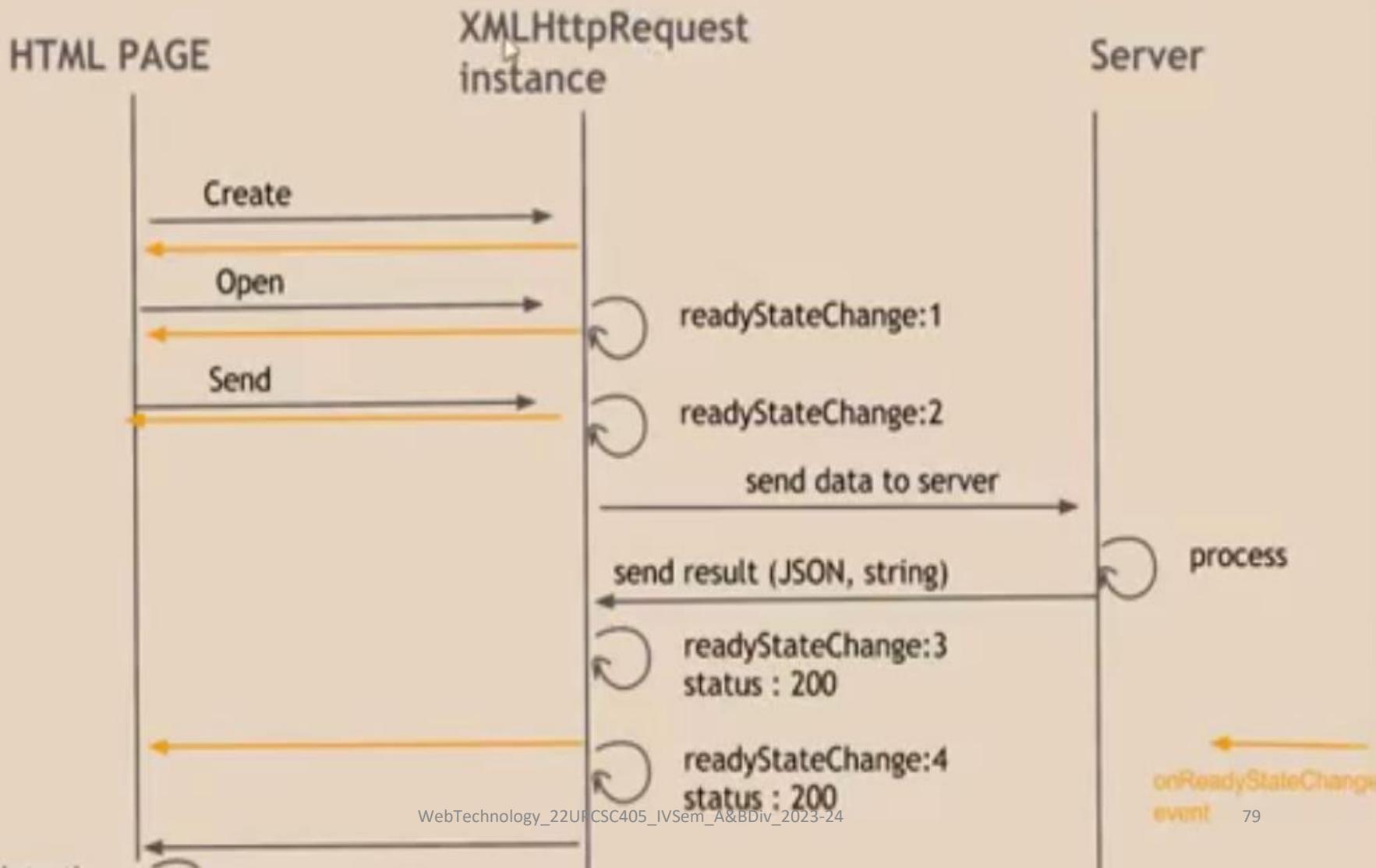
Provides the ability to

- 1- Send asynchronous requests to the server side.
- 2- Be noticed in each step of the flow (request sent, headers received, loading, done, error occurred)

# XMLHttpRequest object



# SUCCESSFUL AJAX FLOW



# AJAX Technologies

---

- AJAX is a group of inter-related technologies.
  - ❖ HTML/XHTML and CSS
  - ❖ DOM
  - ❖ XML or JSON
  - ❖ XMLHttpRequest
  - ❖ JavaScript

# Contd.

---

- ❖ HTML/XHTML and CSS
  - These technologies are used for displaying content and style. It is mainly used for presentation.
- ❖ DOM
  - It is used for dynamic display and interaction with data.
- ❖ XML or JSON
  - For carrying data to and from server. JSON (JavaScript Object Notation) is like XML but short and faster than XML.
- ❖ XMLHttpRequest
  - For asynchronous communication between client and server.
- ❖ JavaScript
  - It is used mainly for client-side validation.

# Understanding XMLHttpRequest

---

- The XMLHttpRequest object is used to exchange data with a server behind the scenes.
- It is possible to update parts of a web page, without reloading the whole page.
- It performs the following operations:
  - Sends data from the client in the background
  - Receives the data from the server
  - Updates the webpage without reloading it.

# Contd.

---

→ Syntax for creating an XMLHttpRequest object:

*variable=new XMLHttpRequest();*

Old versions of Internet Explorer (IE5 and IE6) uses an ActiveX Object:

*variable=new ActiveXObject("Microsoft.XMLHTTP");*

# Properties of XMLHttpRequest object

→ The common properties of **XMLHttpRequest** object are as follows:

Property	Description
onreadystatechange	It is called whenever readystate attribute changes. It must not be used with synchronous requests.
readyState	<p>represents the state of the request. It ranges from 0 to 4.</p> <p><b>0</b> UNOPENED open() is not called.</p> <p><b>1</b> OPENED open is called but send() is not called.</p> <p><b>2</b> HEADERS_RECEIVED send() is called, and headers and status are available.</p> <p><b>3</b> LOADING Downloading data; responseText holds the data.</p> <p><b>4</b> DONE The operation is completed fully.</p>
reponseText	returns response as text.
responseXML	returns response as XML

# Methods of XMLHttpRequest object

→ The important methods of XMLHttpRequest object are as follows:

Method	Description
void open(method, URL)	opens the request specifying get or post method and url.
void open(method, URL, async)	same as above but specifies asynchronous or not.
void open(method, URL, async, username, password)	same as above but specifies username and password.
void send()	sends get request.
void send(string)	send post request.
setRequestHeader(header,value)	it adds request headers.

# AJAX - Send a Request to a Server

→ The XMLHttpRequest object is used to exchange data with a server.

```
xmlhttp.open("GET","ajax_info.txt",true);
xmlhttp.send();
```

Method	Description
<code>open(<i>method,url,async</i>)</code>	Specifies the type of request, the URL, and if the request should be handled asynchronously or not.  <i>method</i> : the type of request: GET or POST <i>url</i> : the location of the file on the server <i>async</i> : true (asynchronous) or false (synchronous)
<code>send(<i>string</i>)</code>	Sends the request off to the server.  <i>string</i> : Only used for POST requests

# Future of Web

---

Web 3.0/A Semantic Web

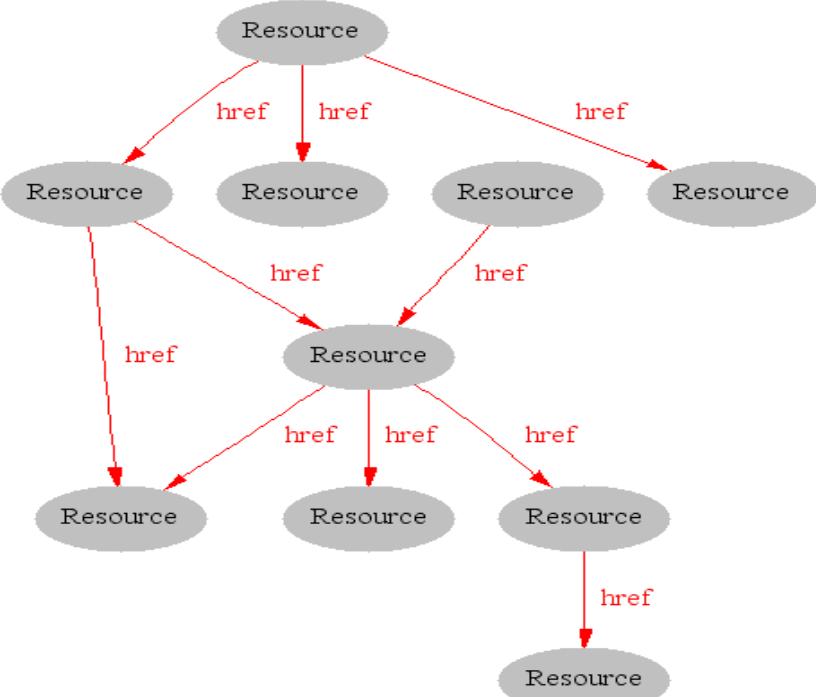
# Before the Web: Document

---

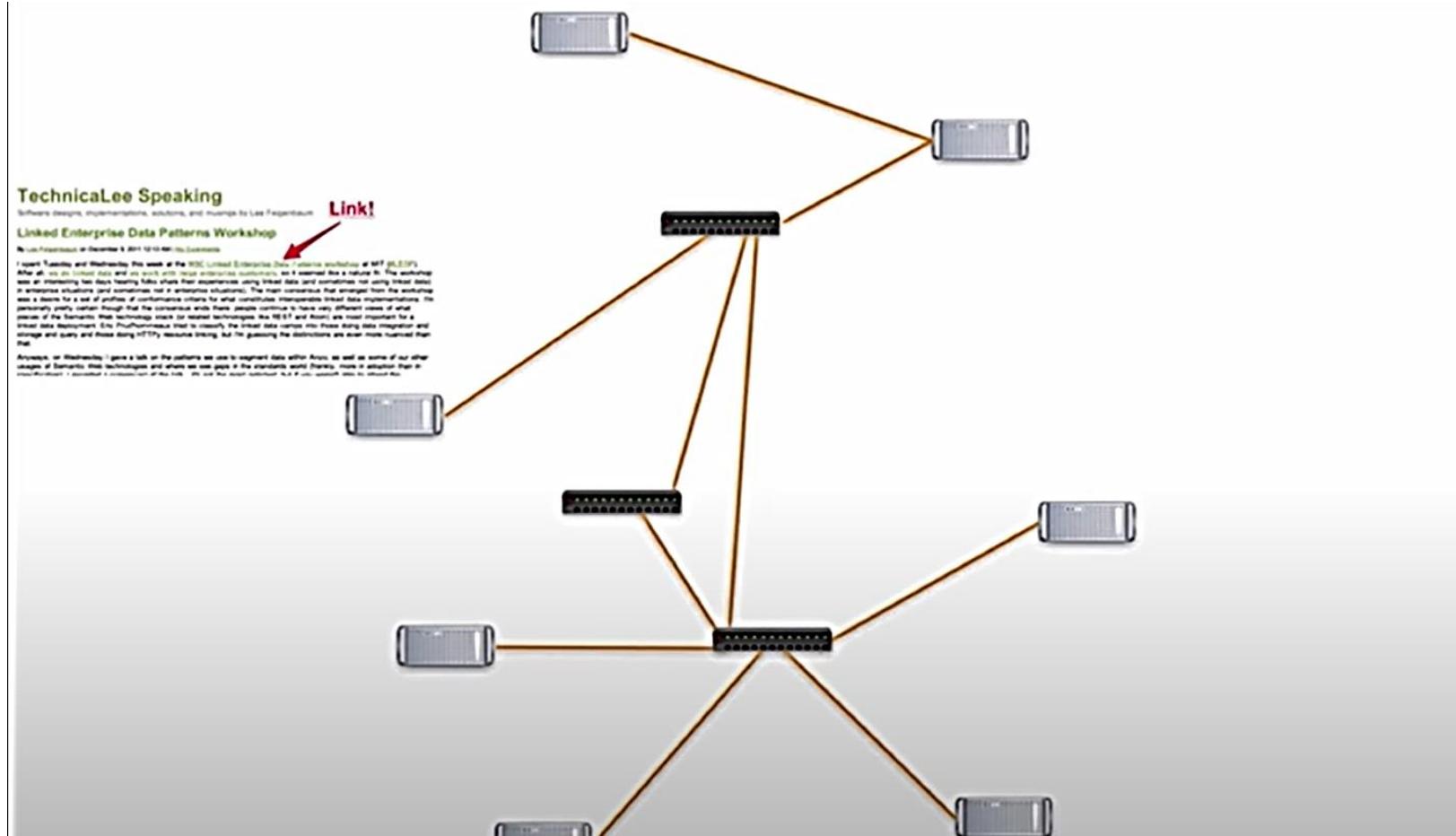


# Web 1.0 : Web of Documents

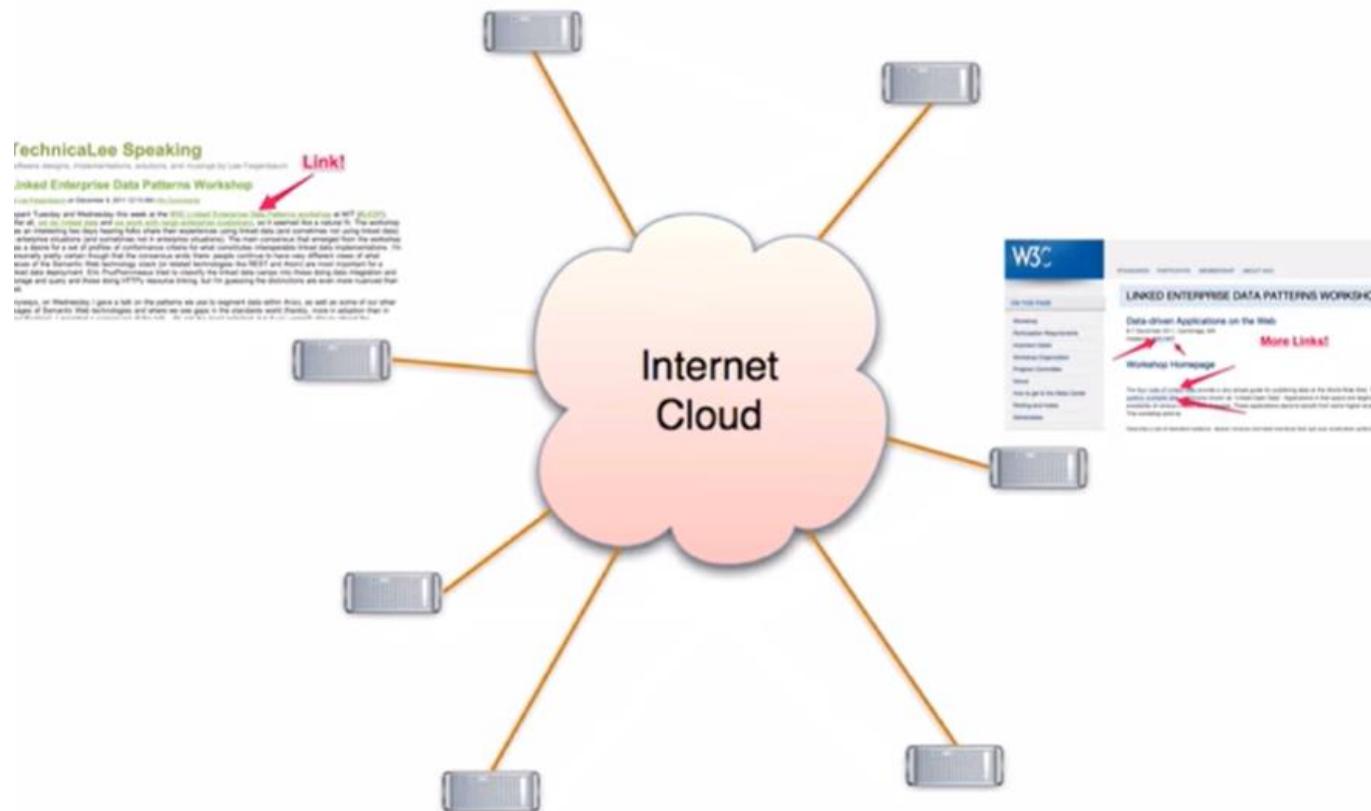
The image shows two screenshots illustrating the Web 1.0 era. The top screenshot is the homepage of the Eleventh International World Wide Web Conference (WWW 2002) held in Honolulu, Hawaii, from May 7-11, 2002. It features a banner for the conference, navigation links like 'Conference Proceedings', 'Call for Participation', and 'Registration Information', and a section for 'FEATURED SPEAKERS (CONFIRMED)' featuring Tim Berners-Lee and Richard A. DeMillo. The bottom screenshot is a screenshot of Tim Berners-Lee's profile page on the 3Com website, showing his bio, publications, and other details.



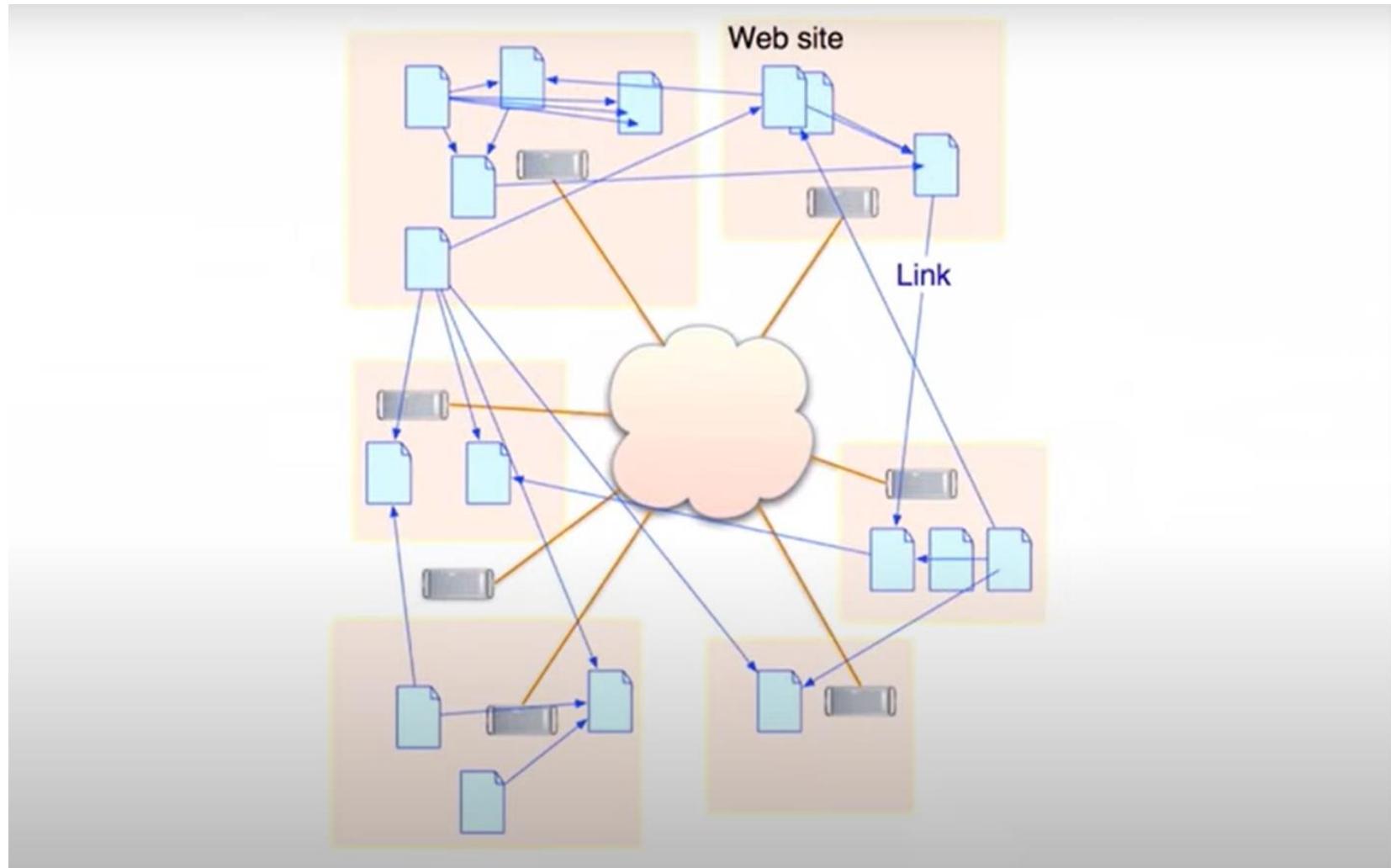
# Web 1.0 : Web of Documents



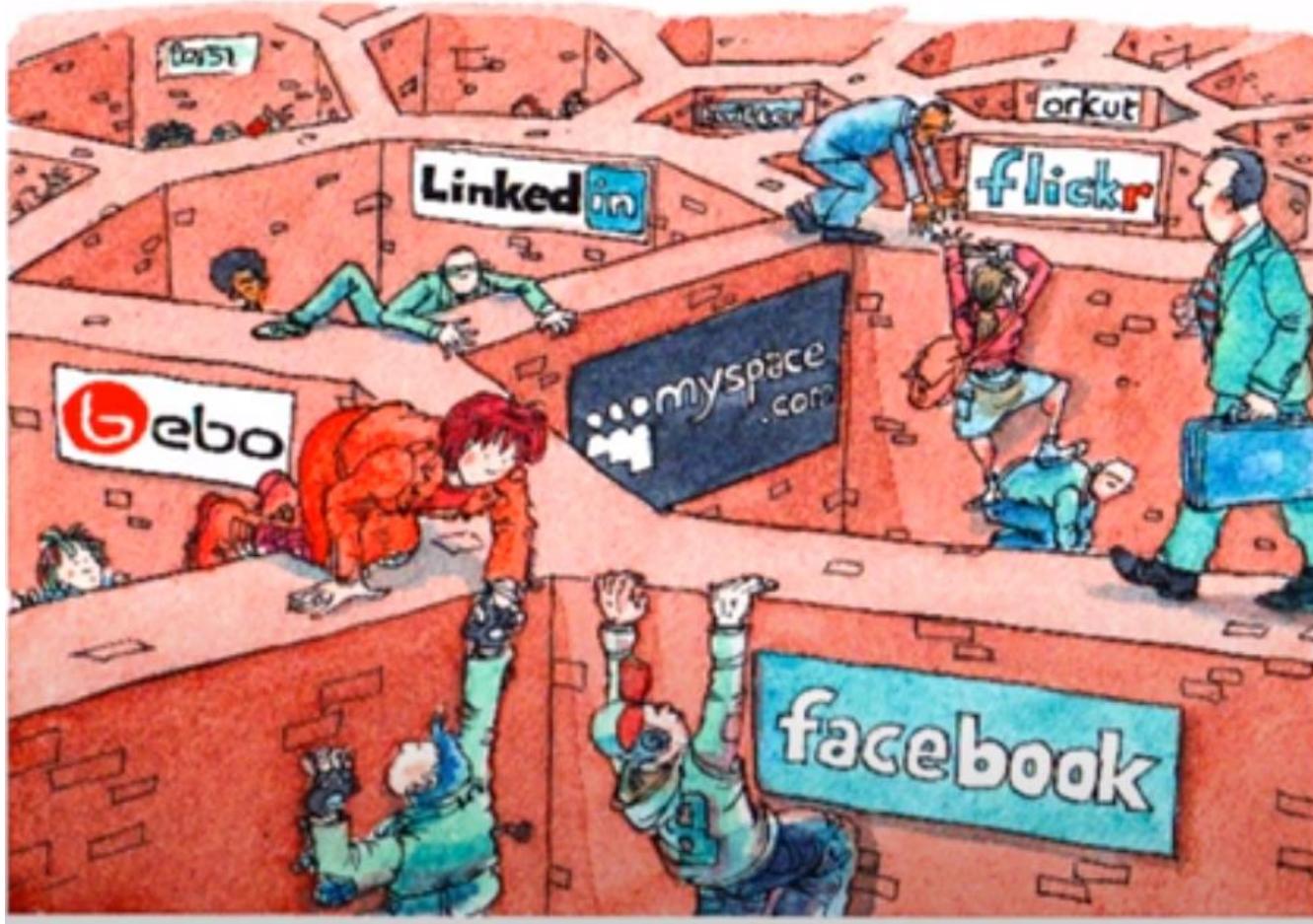
# Web 1.0 : Web of Documents



# Web 1.0 : Web of Documents

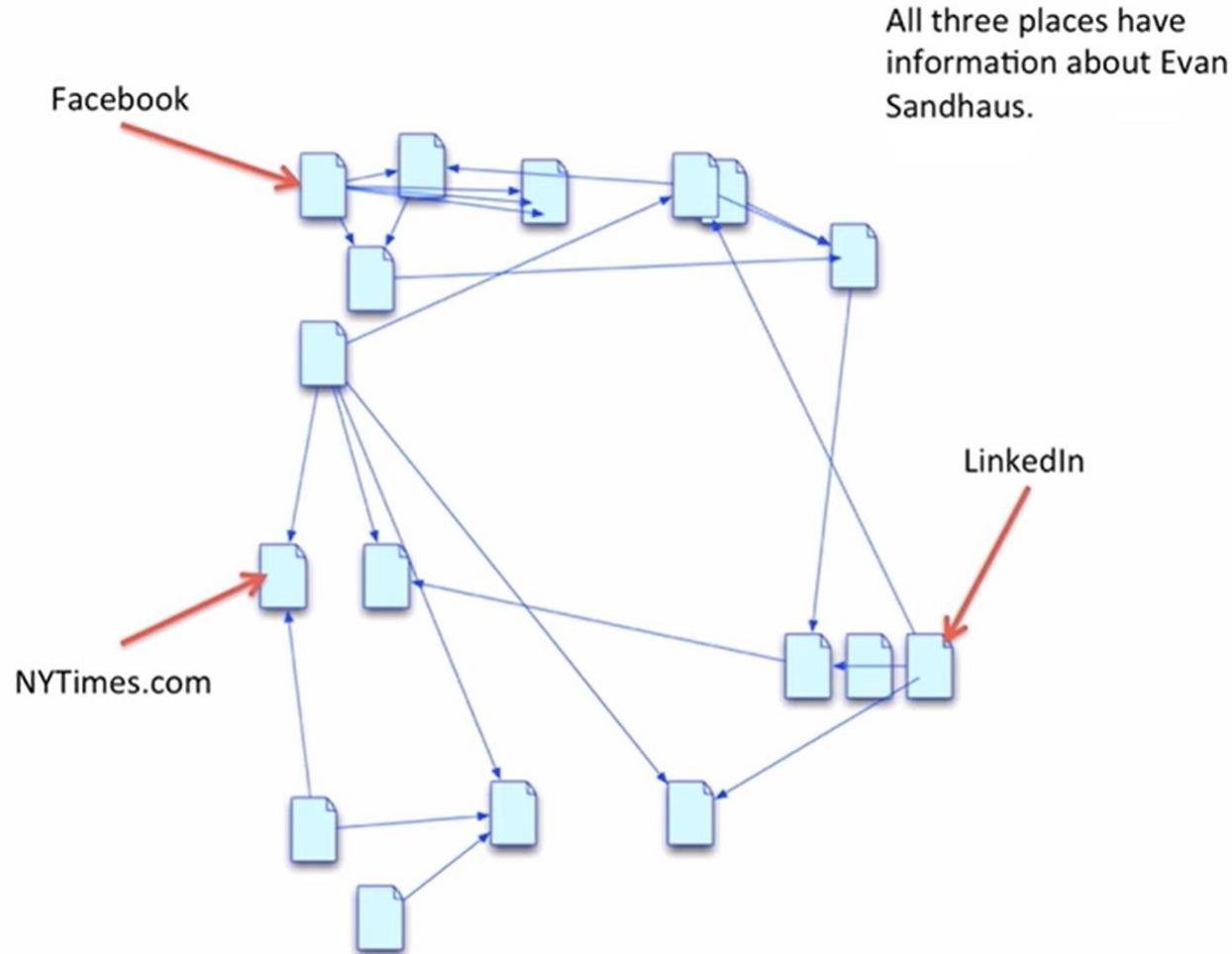


# Web 2.0 : Applications

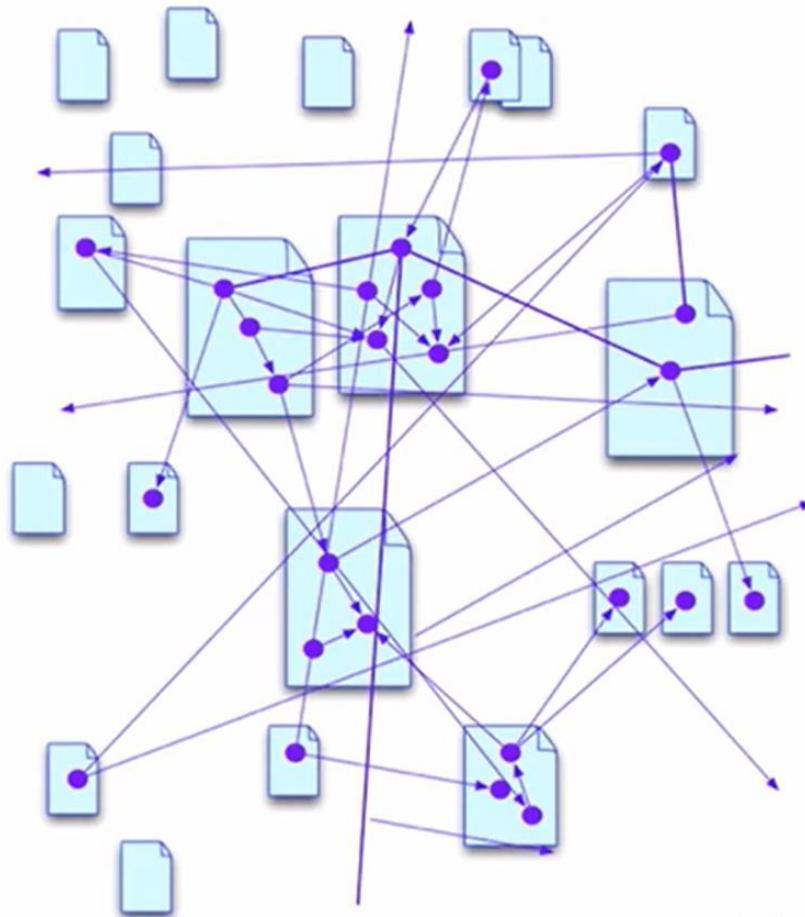


# Web 3.0 : Connect Data

---



# Web 3.0 : Building the data web



Links between *facts!*

# Example: Hard Work using the Syntactic Web...

What is web - Google Search

https://www.google.com/search?q=What+is+web&sca\_esv=580877352&source=hp&ei=vgJNZfOgHc-l-QaGoligBQ&iflsig=A...

Google

What is web

All Images Books Videos News More Tools

About 20,77,00,00,000 results (0.32 seconds)

**Dictionary**  
Definitions from Oxford Languages · Learn more

English

**web**  
/wɛb/  
*noun*

1. a network of fine threads constructed by a spider from fluid secreted by its spinnerets, used to catch its prey.  
Similar: mesh, netting, net, lattice, latticework, lacework, interlacing

2. a complex system of interconnected elements.  
"he found himself caught up in a web of bureaucracy"  
Similar: network, nexus, tangle, knot, complex, mass, conglomeration

*verb*

cover with or as though with a web.  
"she noticed his tanned skin, webbed with fine creases"

Feedback

See more →



World Wide Web

The World Wide Web, commonly known as the Web, is an information system that enables content sharing over the Internet through user-friendly ways meant to appeal to users beyond IT specialists and hobbyists.

Wikipedia

# Why Semantic Web?

---

- Spend less time searching
- Spend less time looking at things that do not matter
- Spend less time explaining what we want to computers

Bottomline – improve the online experience!!!

# What is the semantic web?

---

A binary matrix representing words as rows and letters as columns. The matrix consists of 8 rows and 10 columns. The words are: know, what, you, mean, and a blank row. The matrix is as follows:

0	0	1	0	0	0	0	1	1	
0	1	1	0	i	0	0	1	0	
0	0	k	n	o	w	0	0	0	
0	w	h	a	t	0	0	0	0	
0	0	0	y	o	u	1	0	1	
1	0	1	m	e	a	n	.	.	
1	0	1	1	0	0	1	0	0	



**cricket...**

**...or cricket?**

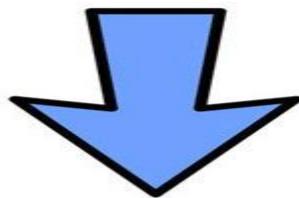


<http://www.fuelinteractive.com/media/images/cricket-semantic-web.jpg>

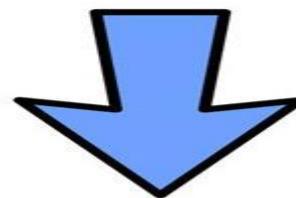
# Which apple?



or



the food



the brand

Copyright © 2012 StrategicRevenue.com

# Google

India

Dhoni



- dhoni **retires**
- dhoni **movie**
- dhoni **wiki**
- dhoni **news**
- dhoni **images**
- dhoni **house**
- dhoni **age**
- dhoni **retires from test**
- dhoni **twitter**
- dhoni **net worth**

Google Search

I'm Feeling Lucky

# Why not just Google?

---

- ❖ Currently, users search for data on the web by asking questions that are of the form: “Which documents contain these words and phrases?”
- ❖ The Semantic web will involve more involved questions, relationships and trust.
- ❖ Instead of word matching, the web will be able to show related items showing new relationships.

# What exactly is the Semantic web?

---

- ❖ Semantics means (from Ancient Greek), is the study of meaning.
- ❖ A set of **standard** and *best practices* for sharing data and the semantics of the data over the web *for use by applications*

## Standard

- *The RDF data model*
- *The SPARQL query language*
- *The RDF Schema and OWL standards for storing vocabularies and ontologies.*

## Best practices:

- The use of URIs to represent the names of things
- The use of standards like RDF and SPARQL

## Example:

uH982312d8e556?Address

<https://www.epf.gov.in/employer/employee/Address>

# What exactly is the Semantic web?

---

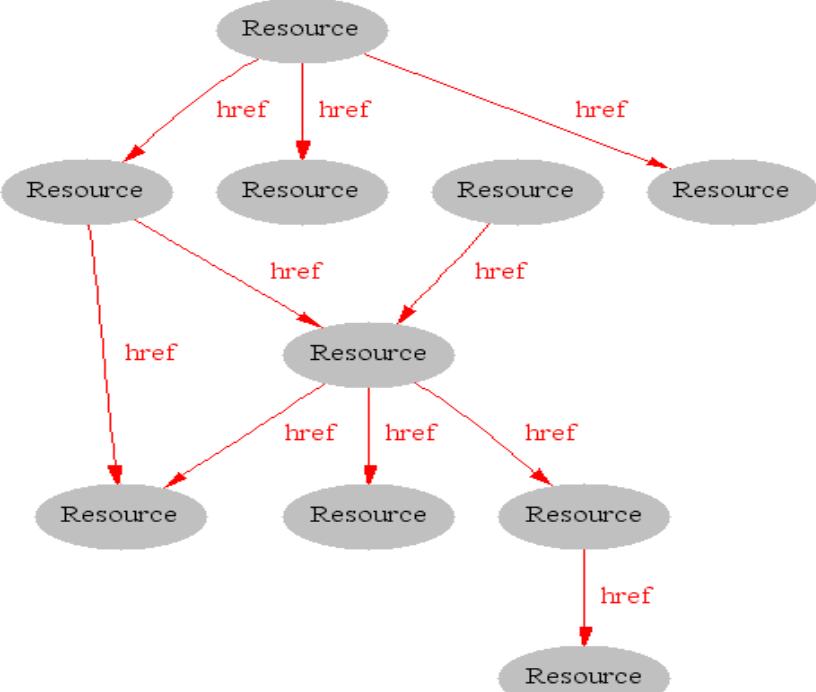
- ✓ The Semantic Web is a vision of an extension of the existing World Wide Web, which provides software programs with machine-interpretable metadata of the published information and data.

Or

- ✓ The Semantic Web is an extension of the current web in which information is given well-defined meaning, better-enabling computers and people to work in cooperation.

# Where we are today: The Syntactic web

The image shows two screenshots. The top screenshot is the homepage of the WWW 2002 conference, held in Honolulu, Hawaii, from May 7-11, 2002. It features a logo with a globe and the text "WWW 2002 THE ELEVENTH INTERNATIONAL WORLD WIDE WEB CONFERENCE". The bottom screenshot is a web browser displaying Tim Berners-Lee's profile page. The browser window title is "Tim Berners-Lee - Netscape". The profile includes a photo of Tim Berners-Lee, his name, and links to his bio, address, talks, speaking engagements, and press interviews. The browser interface shows various tabs and icons.



# Impossible (?) using the Syntactic Web...

---

- Complex queries involving **background knowledge**
  - Find information about “animals that use sonar but are neither bats nor dolphins”  
**e.g., Barn Owl**
- Locating information in **data repositories**
  - Travel enquiries
  - Prices of goods and services
  - Results of human genome experiments
- Finding and using “**web services**”
  - Visualise surface interactions between two proteins
- Delegating complex tasks to web “**agents**”
  - Book me a holiday next weekend somewhere warm, not too far away, and where they speak French or English



# Where we are today: the Syntactic web

---

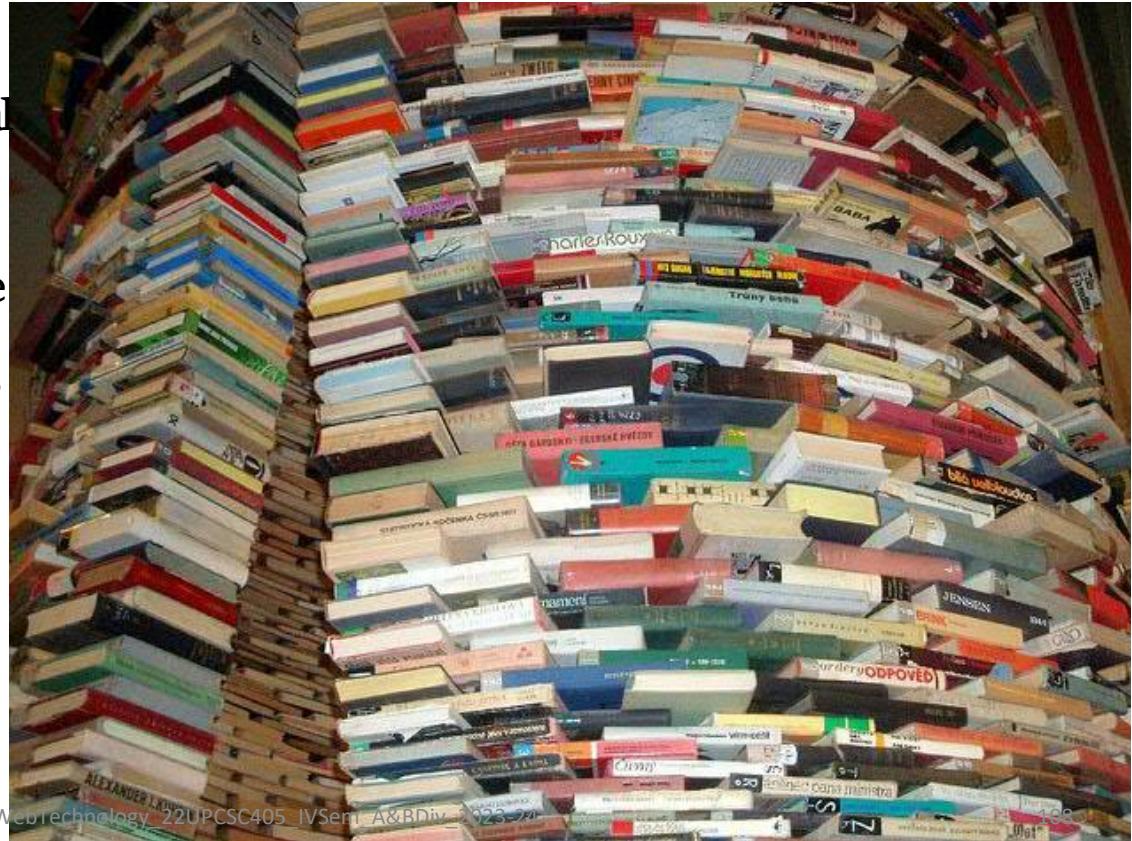
- Today's Web may be defined as the *Syntactic Web*.
  - Where information presentation is carried out by computers, and the **interpretation** and **identification** of relevant information is delegated to human beings.

# Information overload

---

- Identification & interpretation requires great effort to **evaluate, classify** and **select relevant information.**
- Because:→

The volume of available digital data is growing at an exponential rate, it is becoming virtually impossible for human beings to manage the complexity and volume of the available information.



# Why can't computers do this job for us?

---

## → Meta data

→ Web pages do not contain information about themselves  
i.e. about their contents and the subjects to which they refer.

Analogy: Imagine that we wanted to learn about the TCP/IP protocol. We would have to look for a book about networks. If we only used “network” as a keyword, we would retrieve *computer science books*, as well as books about *telephone and electrical networks*. We would then be responsible for filtering and selecting those books that are of genuine interest.

# Example: Hard Work using the Syntactic Web...

A screenshot of a Google search results page. The search query "TCP/IP protocol" is entered in the search bar. The results are categorized under "Web". The first result is a link to a TechTarget definition of TCP/IP, which includes a snippet about it being the basic communication language of the Internet. The second result is a link to the Wikipedia page for the Internet protocol suite, with a snippet about its three top layers: application, presentation, and link layers. The third result is a link to the Wikipedia page for TCP, with a snippet about it being one of the core protocols of the Internet protocol suite. The fourth result is a link to a reference page for various TCP/IP protocols, with a snippet about DARPA's development of TCP/IP.

TCP/IP protocol - Google

[https://www.google.co.in/?gfe\\_rd=cr&ei=T2u-VMWTNM2DoAORuYKICg&gws\\_rd=ssl#q=TCP%2FIP+protocol](https://www.google.co.in/?gfe_rd=cr&ei=T2u-VMWTNM2DoAORuYKICg&gws_rd=ssl#q=TCP%2FIP+protocol)

Google TCP/IP protocol

Web Images Books Videos Maps More ▾ Search tools

About 2,40,00,000 results (0.32 seconds)

[What is TCP/IP \(Transmission Control Protocol/Internet ...](#)  
searchnetworking.techtarget.com/definition/TCP-IP ▾  
TCP/IP (Transmission Control Protocol/Internet Protocol) is the basic communication language or protocol of the Internet. It can also be used as a ...  
What is gateway? - What is extranet? - What is intranet?

[Internet protocol suite - Wikipedia, the free encyclopedia](#)  
en.wikipedia.org/wiki/Internet\_protocol\_suite ▾  
Jump to Comparison of TCP/IP and OSI layering - [edit]. The three top layers in the OSI model—the application layer, the presentation layer and ...  
Link layer - Category:Link protocols - Lists of network protocols - Encapsulation

[TCP - Wikipedia](#)  
en.wikipedia.org/wiki/Transmission\_Control\_Proto... ▾  
The Transmission Control Protocol (TCP) is one of the core protocols of the Internet protocol suite (IP), and is so common that the entire suite is often called ...  
Octet - TCP - TCP congestion avoidance - Slow-start

[TCP / IP Protocols: ICMP UDP FTP HTTP Reference Page](#)  
www.protocols.com/pbook/tcpip1.htm ▾  
The Defense Advance Research Projects Agency (DARPA) originally developed Transmission Control Protocol/Internet Protocol (TCP/IP) to interconnect ...

# Example:

A screenshot of a Google search results page. The search query "TCP/IP protocol" is entered in the search bar. The results page shows approximately 2,40,00,000 results found in 0.32 seconds. The first result is a link to "What is TCP/IP (Transmission Control Protocol/Internet ...)" from searchnetworking.techtarget.com, with a snippet about TCP/IP being the basic communication language of the Internet. The second result is a link to "Internet protocol suite - Wikipedia, the free encyclopedia" from en.wikipedia.org, with a snippet about the three top layers of the OSI model. The third result is a link to "TCP - Wikipedia" from en.wikipedia.org, with a snippet about TCP being one of the core protocols of the Internet protocol suite.

TCP/IP protocol - Google

[https://www.google.co.in/?gfe\\_rd=cr&ei=T2u-VMWTNM2DoAORuYKICg&gws\\_rd=ssl#q=TCP%2FIP+protocol](https://www.google.co.in/?gfe_rd=cr&ei=T2u-VMWTNM2DoAORuYKICg&gws_rd=ssl#q=TCP%2FIP+protocol)

Google TCP/IP protocol

Web Images Books Videos Maps More ▾ Search tools

About 2,40,00,000 results (0.32 seconds)

[What is TCP/IP \(Transmission Control Protocol/Internet ...\)](#)  
searchnetworking.techtarget.com/definition/TCP-IP ▾  
TCP/IP (Transmission Control Protocol/Internet Protocol) is the basic communication language or protocol of the Internet. It can also be used as a ...  
[What is gateway? - What is extranet? - What is intranet?](#)

[Internet protocol suite - Wikipedia, the free encyclopedia](#)  
en.wikipedia.org/wiki/Internet\_protocol\_suite ▾  
Jump to Comparison of TCP/IP and OSI layering - [edit] The three top layers in the OSI model—the application layer, the presentation layer and ...  
[Link layer - Category:Link protocols - Lists of network protocols - Encapsulation](#)

[TCP - Wikipedia](#)  
en.wikipedia.org/wiki/Transmission\_Control\_Proto... ▾  
The Transmission Control Protocol (TCP) is one of the core protocols of the Internet protocol suite (IP), and is so common that the entire suite is often called ...  
[Octet - TCP - TCP congestion avoidance - Slow-start](#)

# How humans see a web page

The screenshot shows a Microsoft Internet Explorer window displaying a personal website. The address bar shows the URL as `file:///d:/karin/public_html/SP/index.html`. The page content includes:

- A large black vertical rectangle on the left side.
- A header section with the title "KARIN KOOGAN BREITMAN".
- A biography of Dr. Karin Breitman, mentioning her education at Pontifícia Universidade Católica do Rio de Janeiro, her MSc from COPPE-UFRJ, and her DSc from the Technion. It also notes her involvement in various grants and projects like ESSMA- 552068/2002-0 and her interests in Autonomic Computing, Requirements Engineering, Semantic Web, and Scenario-based software development. It mentions two books published in Portuguese.
- A "Courses" section listing "GRADUATE - INF 2391 - KNOWLEDGE REPRESENTATION".
- A "Recent Talks" section listing three events: "IFIP WORLD CONFERENCE - TUTORIAL - Semantic Web: Concepts, Technologies and Applications - AUGUST 20", "WEBCAST OF THE " SEMANTIC WEB TECHNOLOGIES: CHALLENGES AND OPPORTUNITIES" TALK - NASA", and "ODDARD SPACE CENTER - INFORMATION SCIENCE & TECHNOLOGY COLLOQUIUM - FEBRUARY 23".
- A "Publications" section featuring a QR code linking to the book "WEB SEMANTICA: A INTEIRO DE FATO" by Karin Koogan Breitman, published by Springer.

# How computers see the same web page



# Solution to the syntactic web

---

- Best practices for sharing data over the web
  - The use of URIs to name things
  - The use of standards such as RDF and SPARQL.
- To provide excellent guidelines for the creation of an infrastructure for the semantic web.

# Search engines started understanding natural languages

What was the age of nehru when he got married ?

What was the age of nehru when he died ?

https://www.google.co.in/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=what%20was%20the%20age%20of%20nehru%20when%20he%20died%20

Apps Welcome to Facebook...

Google what was the age of nehru when he died ?

Web News Images Videos More Search tools

About 7,54,000 results (0.66 seconds)

**74 (1889–1964)**

Jawaharlal Nehru, Age at death



Indira Gandhi Daughter 66 (1917–1984) Mahatma Gandhi 78 (1869–1948) Subhas Chandra Bose 48 (1897–1945)

Feedback

**India Mourning Nehru, 74, Dead of a Heart Attack: World ...**  
www.nytimes.com/learning/general/onthisday/big/0527.html ▾  
New Delhi, May 27 -- Jawaharlal Nehru, maker of modern India and her Prime Minister for the 17 years since independence, died today at the age of 74. Death was caused by coronary thrombosis, the formation of a clot in a coronary artery of the heart.

**Jawaharlal Nehru - Wikipedia, the free encyclopedia**  
en.wikipedia.org/wiki/Jawaharlal\_Nehru ▾

**Jawaharlal Nehru**  
Former Prime Minister of India  
Jawaharlal Nehru was the first Prime Minister of India and a central figure in Indian politics for much of the 20th century. Wikipedia  
Born: November 14, 1889, Allahabad  
Died: May 27, 1964, New Delhi  
Spouse: Kamala Nehru (m. 1916–1936)  
Children: Indira Gandhi  
Parents: Motilal Nehru, Swaruprani Thussu

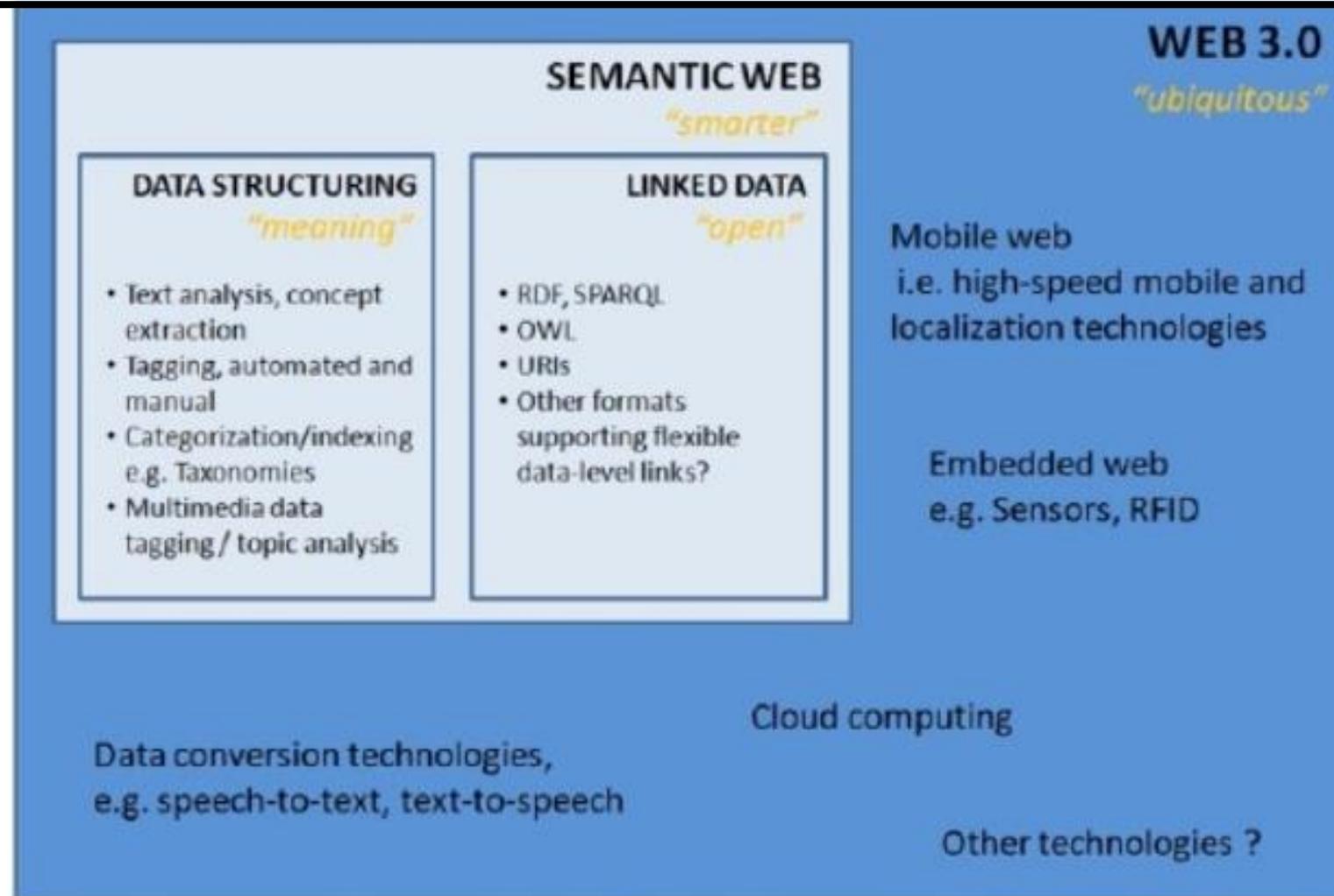
**Books**

      
1946 1936 1934 1929

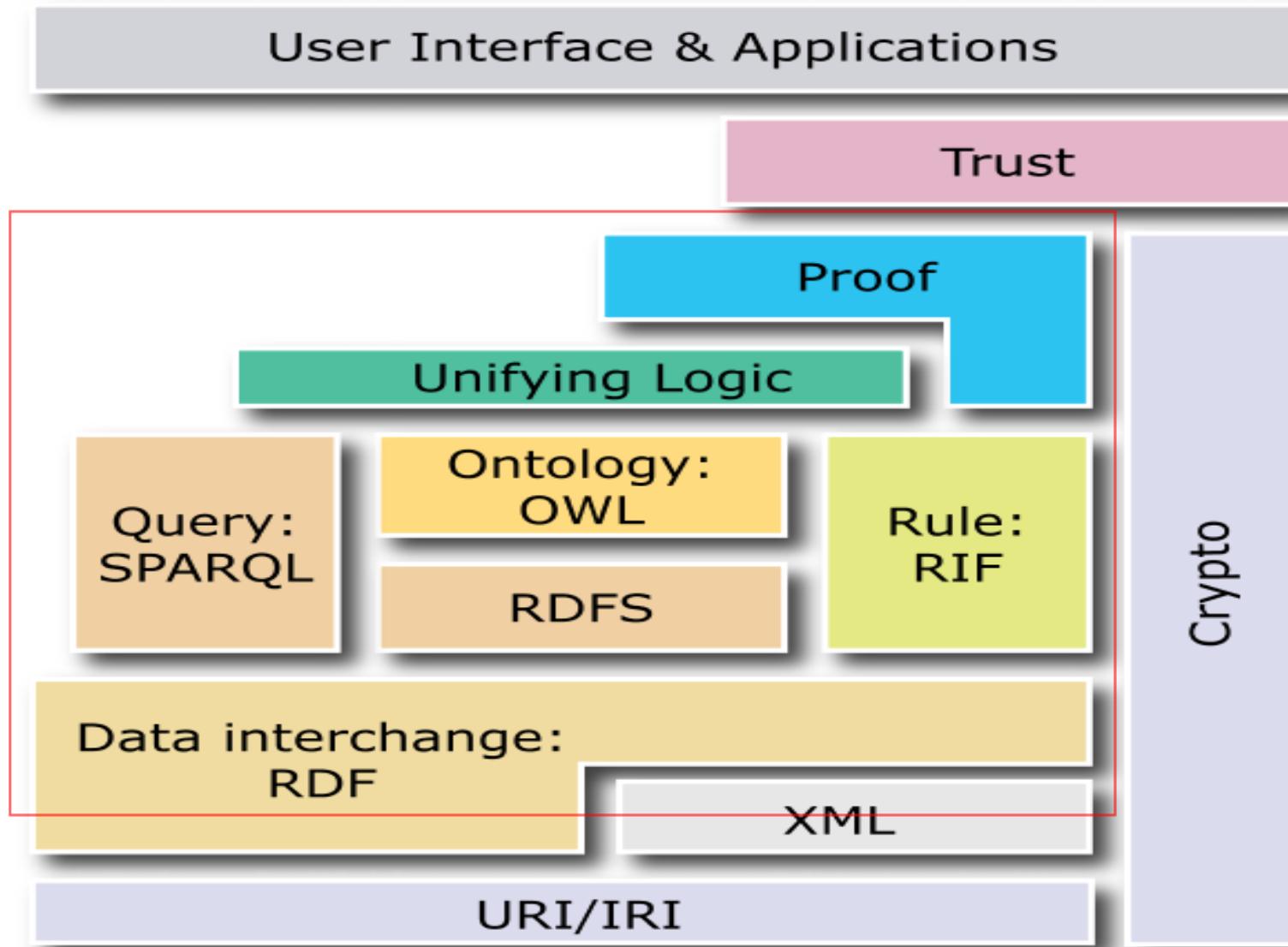
# WEB 3.0, SEMANTIC WEB, DATA STRUCTURING AND LINKED DATA

Birdseye View

[www.SemanticsIncorporated.com](http://www.SemanticsIncorporated.com)



# Semantic Web layered architecture



# Hypertext Web technologies

---

→ URI provides means for uniquely identifying semantic web resources.  
Unicode an international encoding standard for use with different languages and scripts, by which each letter, digit, or symbol is assigned a unique numeric value that applies across different platforms and programs.

XML is a markup language that enables creation of documents composed of structured data.

XML Namespaces provides a way to use markups from more sources.

# Standardized Semantic Web technologies

---

- Resource Description Framework (RDF) It represents information about resources in the form of a Giant Global Graph.
- RDF Schema (RDFS) provides a basic vocabulary for RDF. Using RDFS it is possible to create hierarchies of classes and properties.
- Web Ontology Language (OWL) extends RDFS by adding more advanced constructs to describe the semantics of RDF statements.

SPARQL is an RDF query language - it can be used to query any RDF-based data (i.e., including statements involving RDFS and OWL). Querying language is necessary to retrieve information for semantic web applications.

- RIF is a *rule interchange format*. It is important, for example, to allow describing relations that cannot be directly described using description logic used in OWL.

# What is RDF?

---

- ↳ RDF stands for Resource Description Framework
- ↳ RDF is a framework for describing resources on the web
- ↳ RDF is designed to be read and understood by computers
- ↳ RDF is not designed to be displayed to people
- ↳ RDF is written in XML
- ↳ RDF is a part of the W3C's Semantic Web Activity
- ↳ RDF is a W3C Recommendation

## Example:

- ✓ Describing properties for shopping items, such as price and availability
- ✓ Describing time schedules for web events
- ✓ Describing information about web pages (content, author, created and modified date)
- ✓ Describing content and rating for web pictures
- ✓ Describing content for search engines

# RDF Resource, Property, and Property Value

---

- A Resource is anything that can have a URI, such as "<http://www.w3schools.com/rdf>"
- A Property is a Resource that has a name, such as "author" or "homepage"
- A Property value is the value of a Property, such as "Jan Egil Refsnes" or "<http://www.w3schools.com>" (note that a property value can be another resource)

## Example:

```
<?xml version="1.0"?>
<RDF>
  <Description about="http://www.w3schools.com/rdf">
    <author>Jan Egil Refsnes</author>
    <homepage>http://www.w3schools.com</homepage>
  </Description>
</RDF>
```

- The combination of a **Resource**, a **Property**, and a **Property value** forms a Statement (known as the subject, predicate, and object of a Statement).
- Each triple consists of a subject (resource), a predicate (property), and an object (value or another resource).

# RDF Statements

---

- ✓ Statement: "The author of <http://www.w3schools.com/rdf> is Jan Egil Refsnes".

The **subject** of the statement above is:

<http://www.w3schools.com/rdf>

The **predicate** is: author

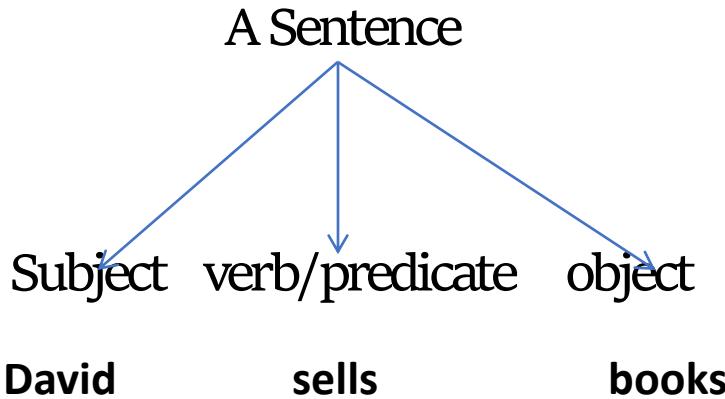
The **object** is: Jan Egil Refsnes

# RDF

---

- ❖ Also called as triplet.

Example: David sells books.



Resource: is anything that has identity

Ex:- David and books

These are identified by using URI

Description is a container holding several statements describing the resource.

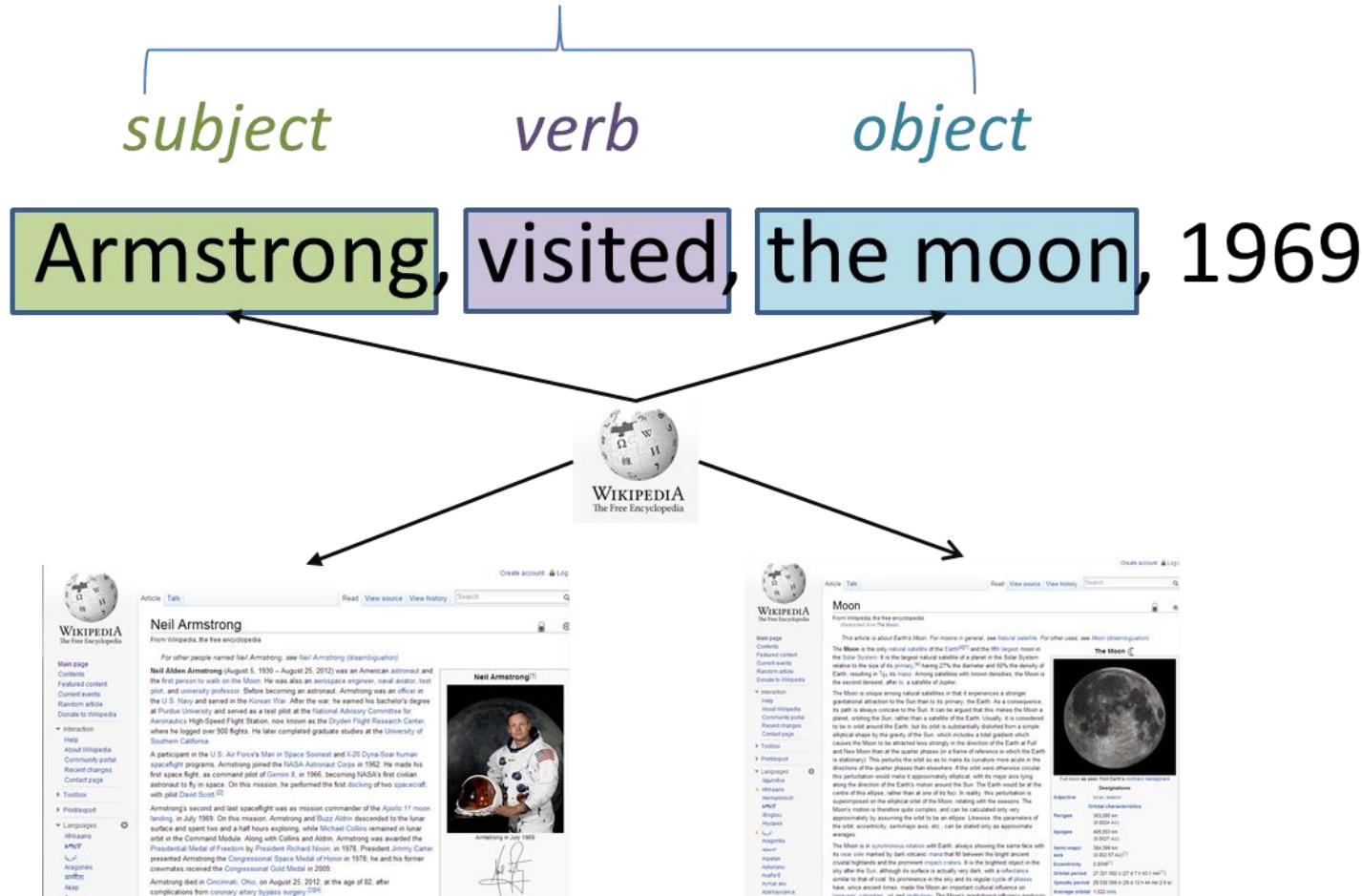
Ex: Specific identification of resource called David

A framework: A framework is needed to enable humans and machine to make and understand statements

Ex: When you type David, you will get complete sentence like David sells books

# Example:

## RDF Triple



# Ontology

---

❖ The word ontology comes from two Greek words:

Onto= existence, or being real

Logia = science, or study

Ontology is a term in philosophy and its meaning is “**theory of existence**”.

Ontology is an explicit specification of conceptualization.

Ontology is a **body of knowledge** describing some domain, typically a common sense knowledge domain.

# What is Ontology?

---

- ✓ Ontologies are **defined in a formal, machine-processable language** for the explicit purpose of **sharing semantic information** across **automated systems**.
- ✓ In simple terms, **ontology** seeks the **classification** and **explanation** of **entities**.
- ✓ Ontology is about the exact description of things and their **relationships**.
- ✓ For the web, ontology is about the **exact description of web information** and **relationships** between the **web**.

# What the Semantic web Is Not

---

- ✓ The semantic web is not **artificial intelligence**
- ✓ The semantic web is not a **separate web**
- ✓ The Semantic Web Will Not **Demand the Use of Complex Expressions**
- ✓ The Semantic Web Is Not a Rerun of a **Failed Experiment**

# Ten Myths...



- 1) Semantic Web is science fiction
- 2) Semantic Web is for tagging web sites
- 3) Semantic Web will put Google out of business
- 4) Semantic Web is too complex to succeed
- 5) Semantic Web is a catalog system
- 6) Semantic Web is an ivory tower design
- 7) Semantic Web is Description Logic (only)
- 8) Semantic Web is (just) Artificial Intelligence
- 9) Semantic Web is a \$20B industry
- 10) Semantic Web hasn't changed the world