Copyright Notice

This presentation is intended to be used only by the participants who attended the training session conducted by Prakash Badhe.

This presentation is for education purpose only. Sharing/selling of this presentation in any form is NOT permitted.

Others found using this presentation or violation of above terms is considered as legal offence.

AJAX Basics

Prakash Badhe

prakash.badhe@vishwasoft.in

Prerequisites…

- To understand Ajax you need to know
  - HTML
  - Java Script in html
  - Little bit CSS in html
  - XML
  - Server side programming with java/.net/php

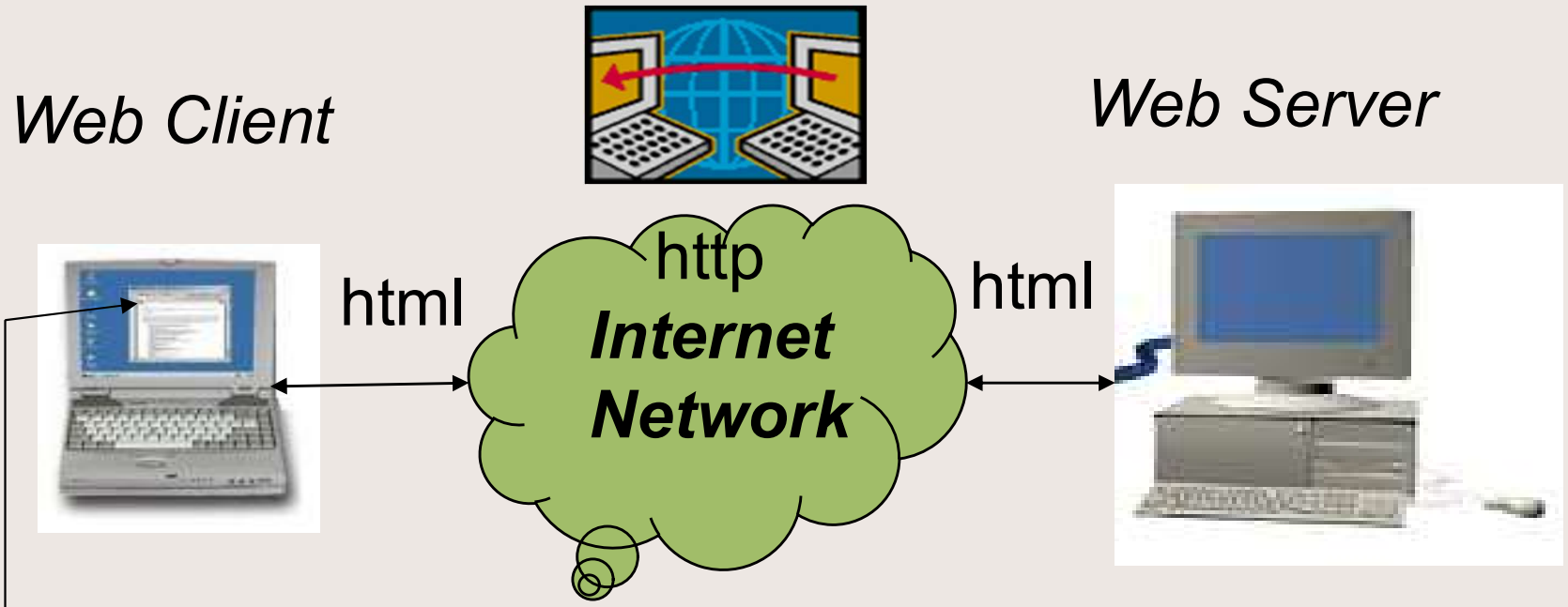
Java Script

- Java Script is the primary scripting language used within the html web pages executed by browser on client side.
- Java Script program contains variables, functions, standard objects and event handling functions to interact with the web user.
- Java Script was used primarily for parameter validation on client side like email id format, number format. length of characters in Login id etc.
- Java Script makes the user experience more interactive and entertaining on client side.

XML

- XML is extensible markup language like html
- XML is strongly typed language.
- XML contains application defined data elements which varies from application to application.
- Different users can define different types of markup tags/elements and their respective values.
- DTD or XML Schema files serves the reference structure to validate the xml elements and their values.

Internet:The Rise of Web



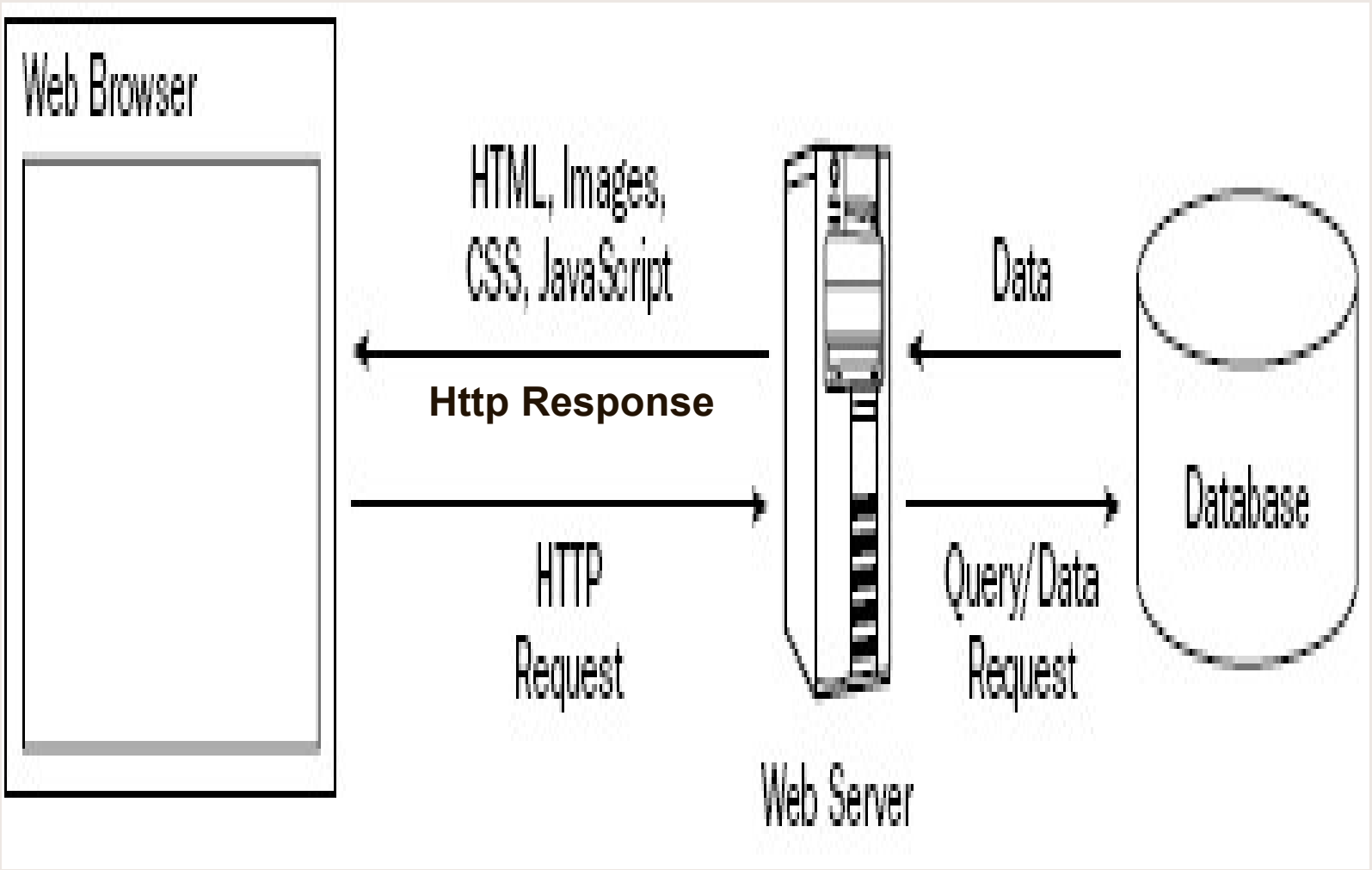
URL : <protocol>:<Host Address : Port Number>/filename

http://www.microsoft.com:80/index.html

Server side Programming

- In J2EE web applications Servlets and JSPs are server side dynamic components that generate dynamic html responses.
- In Microsoft technologies ASP.Net with VB.Net or C#.Net support dynamic pages.
- In php applications the php interpreter supports php scripting for dynamic web pages.
- We need a web server like Jakarta Tomcat /Apache Web server/Microsoft IIS web server to support web application components and web html pages.

Traditional Communication on Internet



Server to Client

- Internet is client-server architecture.
- The web client requests the content information and the web server sends these contents to web clients the browser.
- The contents can be defined as static html pages where the content is fixed and predefined.
- In case of dynamic contents, the content may vary from user to user e.g. email pages on mail server like Gmail, yahoo mail etc.
- These dynamic contents are generated on server by active programming like java, MS.Net,php and sent to web clients.

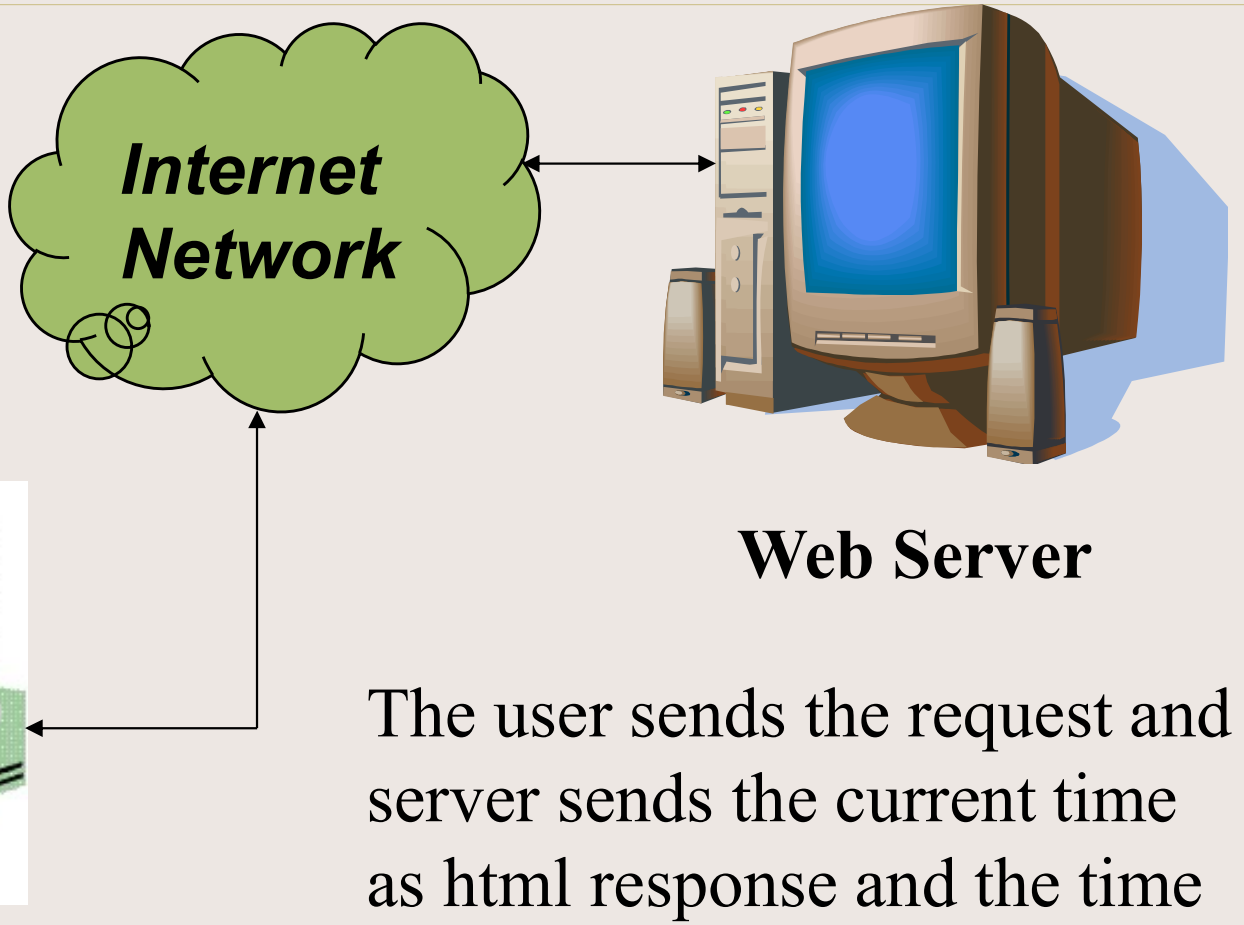
Internet Communication

- The user types the address of web server in browser address bar and hits GO!
- The browser sends http request to web server.
- The web server processes the request and sends response in html tag format to browser.
- The browser displays the graphics meant by html element tags.
- The browser request parameters and server response in html is termed as ‘Http Request’ and ‘Http Response’.

Request-Response Architecture

- The internet applications execution is a request-response process.
- The web client sends the request to web server URL and waits for the response from web server.
- The process is complete only when the client receives the complete response.
- It is a **synchronous** call from web client!
- If the server takes more time to process request, the client has to wait till it gets the response!!

Example : Get Current Time from Server…



The user sends the request and server sends the current time as html response and the time is displayed in browser.

Refresh your time now!

- Every time the user wants to update the current time displayed in browser, he/she has to send the request again by clicking ‘Refresh/Reload’ menu.
- How can be the time continuously updated and displayed in browser without the user refreshing or reloading it ?

Can server update the time to the client?

- Can the server program continuously send the updated time values to the web client ?
- How many of connections will have to be maintained by the web server on internet for millions of clients at same time ?
- HTTP is '*stateless*' i.e. for every new request it is a new connection, current request connection is closed when the response is complete for current request!

Client Pulls the data….

- Instead of server sending data, the client can fetch/pull the data but from server but it may get blocked if the server is busy and it requires user reloading the page continuously for every action.
  - The client gets delayed responses because of **Synchronous** nature of call
  - In delayed response user cannot do any other tasks, But just waiting for the response...
  - The user has to initiate the request every time..

AJAX

- The technique which will **automatically** pull the data from server to browser client in **asynchronous** manner-that is AJAX!
- Ajax doesn't involve reloading/refreshing the entire page.

What is AJAX ?

A Greek Hero ?

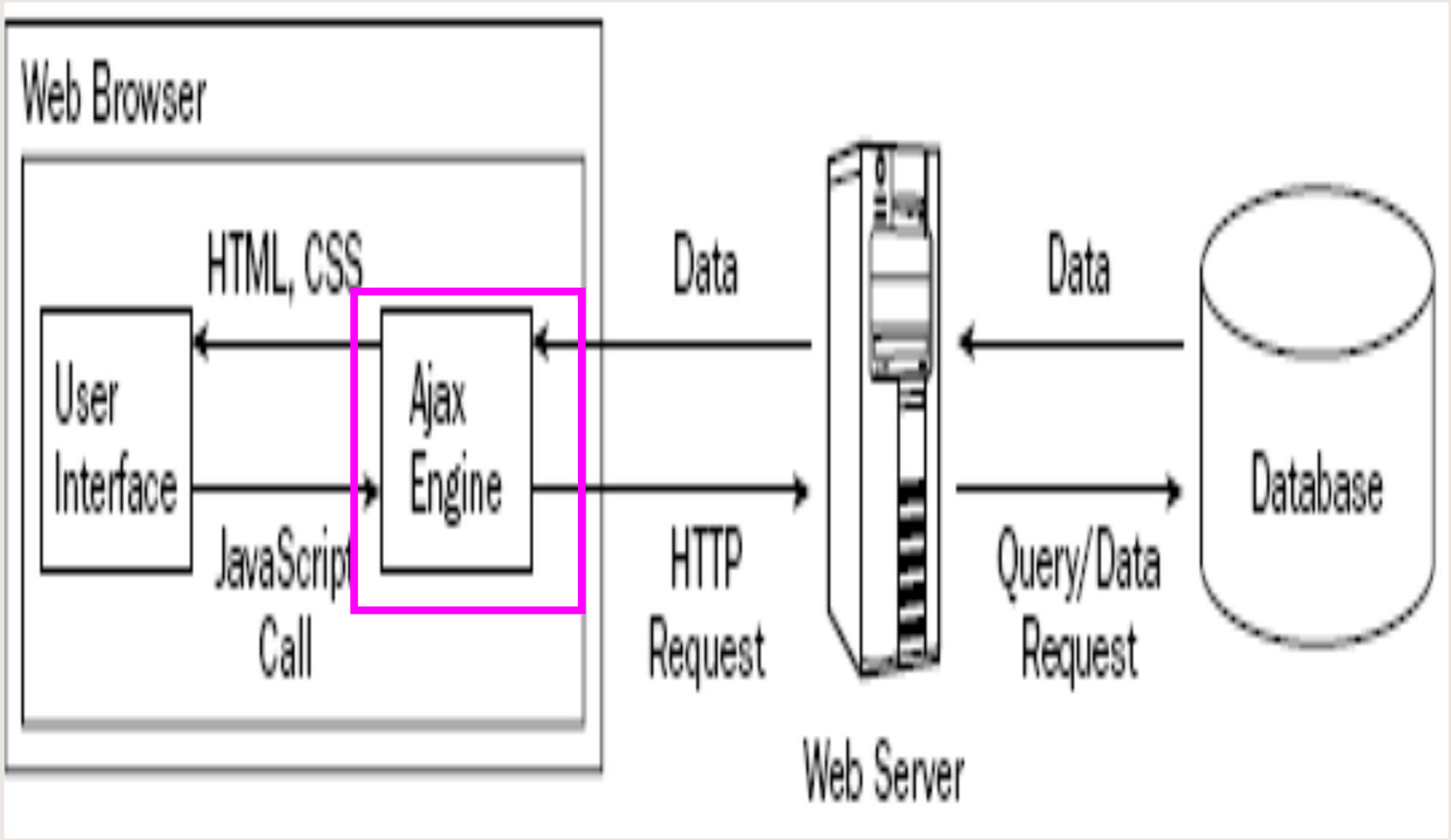
Ajax: the technique…

- Ajax stands for *Asynchronous JavaScript with XML.*
- *Ajax is based on existing technologies and languages like html,javascript and xml.*
- *Ajax is used in client side web pages in web applications i.e. the Ajax in html pages with java script is to be supported by the web browsers.*

Ajax is for the user on Internet

- Ajax is nothing more than an approach for web user interaction with web server.
- This approach involves transmitting only a small amount of information to and from the server in order to give the user the most interactive and rich experience.

Ajax Communication Model



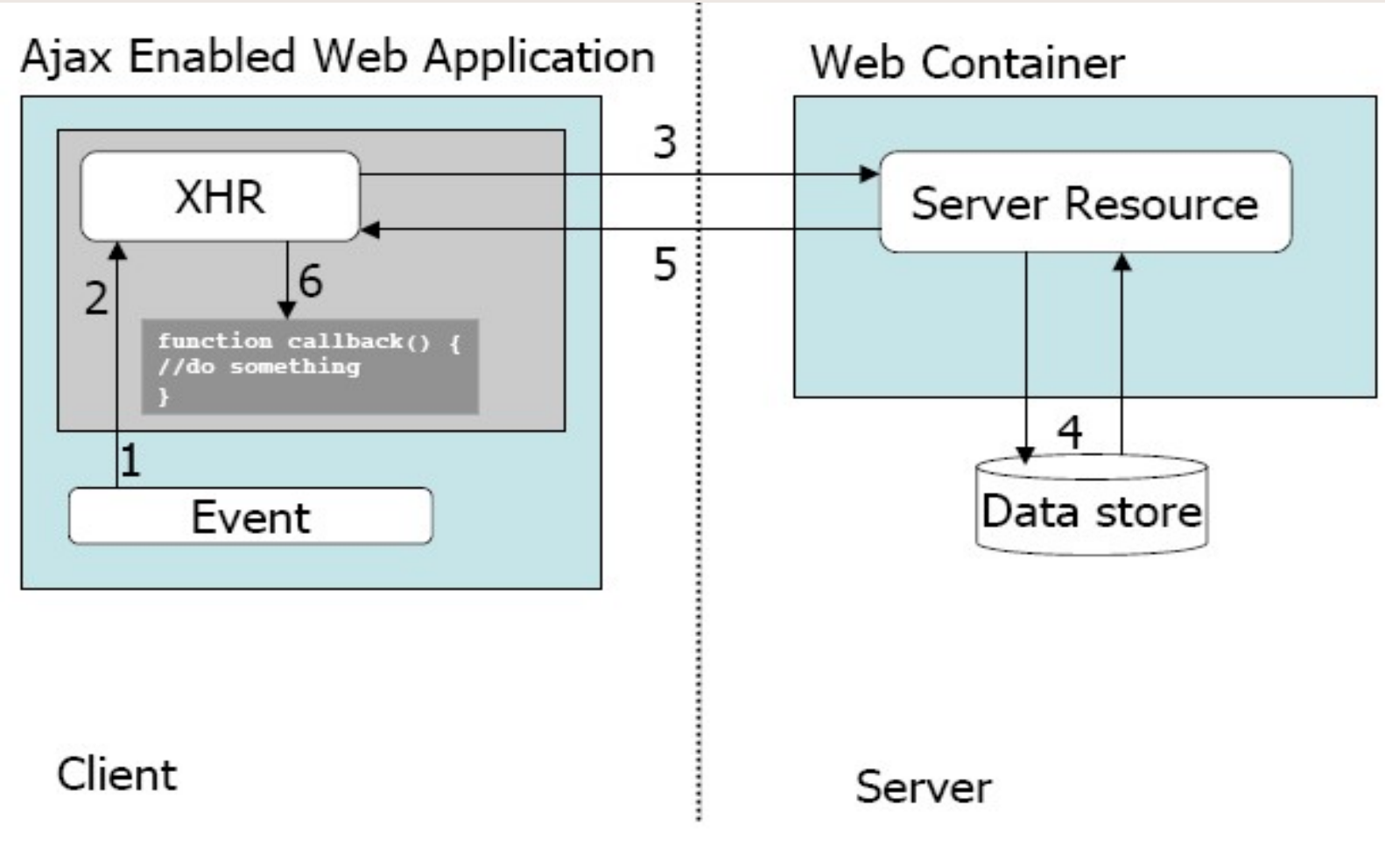
Ajax Engine…

- Instead of the traditional web application model where the browser itself is responsible for initiating requests to, and processing responses from the web server, the Ajax provides an intermediate layer called as '*Ajax Engine*' to handle the communication to server.
- An Ajax engine is really just the JavaScript object functions that is called whenever information needs to be requested from the server.
- Ajax sends the request to web server and process the server response *asynchronously*, meaning that code execution in the browser doesn't wait for a response before continuing.
- The server which could serve up HTML, images, CSS, or JavaScript, is configured to return data that the Ajax engine can use.

Heart of AJAX

- At the heart of AJAX is the JavaScript object called **XMLHttpRequest**, introduced by Microsoft Internet Explorer.
- This **XMLHttpRequest** object is like an HTTP request that could be controlled from JavaScript functions on client side.
- Programmers have access to HTTP status codes and headers, as well as any data returned from the server. The response data might be structured XML, HTML or serialized JavaScript objects, or data in any other format desired by the application.
- Ajax makes it possible to load the server response using pure JavaScript code from the browser, independent of the page load/refresh cycle by the users.

Ajax Interaction to Server



XmlHttpRequest methods

- Open (HttpMethod, ServerURL, asyncMode, username, password);
  - Prepares the request for the call to server– (note the use of '**asynchMode**' flag, last parameter.)
- send(content) : Sends the request to the server.
- abort() : Stops the current request.
- getAllResponseHeaders() :Returns all the response headers for the HTTP request as key/value pairs.
- getResponseHeader("header"): Returns the string value of the specified header.
- setRequestHeader("header", "value"):Sets the specified header to the supplied value.

XmlHttpRequest Properties

Property	Description/Values
readyState (The state of the XHR object)	0 = uninitialized, 1 = loading, 2 = loaded, 3 = interactive, 4 =complete
onreadystatechange	The event handler function that gets called every time the ‘readyState’ value is changed

XMLHttpRequest Response properties

- responseText :The response from the server as a string.
- responseXML :The response from the server as XML.
- Status : The HTTP status code from the server.
- statusText : The text version of the HTTP status code.

XHR in Browsers

- For old version of browsers there is a variation in the way of creating the XHR instances.
- Hence we have to detect the specific type/support of the browser and then apply the browser specific way to create the instances.
- For new versions of browsers there is a common way of creating the instance and is compatible across newer versions.

Create XHR Instance

Different browsers support the **xhr** object differently…

```
//Check browser support and ask to create XMLHTTP object
      function createXMLHttpRequest()
       {
          if (window.ActiveXObject)
              {
               //Microsoft IE Browser
                xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
              }
             else if(window.XMLHttpRequest)
              {
                //Non Microsoft Browsers such as Firefox, Safari and Opera
                 implement it as //a native JavaScript object.
                 xmlHttp = new XMLHttpRequest();
              }
}
```

Send the request

```
// prepare and send the request
function startRequest()
{
    //Ask the browser to create XHR object
    createXMLHttpRequest();
    //Assign to XHR Object the callBack Function
    xmlHttp.onreadystatechange = handleStateChange;
   //Prepare the call to send to ‘MyTime’ URL on server
    xmlHttp.open("GET", “/current“,true);
   //Send the request
    xmlHttp.send(null);
}
```

Response from Server

- The response needs to set the response headers so that the browser will not cache the results locally, defined by the following code:
  - response.setHeader("Cache-Control", "no-cache");
  - response.setHeader("Pragma", "no-cache");
- The response is returned to the browser. The Content-Type is set to “text/xml”—the XMLHttpRequest object can process results only of the “text/html” type.

Response callBack in Client


```
//Process the response from server
function handleStateChange()
{
      if(xmlHttp.readyState == 4) //is the response complete?
      {
         if(xmlHttp.status == 200) //Is the response correct ?
         {
          //Display the message sent by server
          alert("The server replied with: " + xmlHttp.responseText);
        }
      }
 }
```

Ajax Processing Steps

- An object of the XmlHttpRequest object is created.
- Using the open method on XmlHttpRequest object, the call is set up. The URL is set along with the desired HTTP method, typically GET or POST.
- The request is actually sent by a call to the send method on XmlHttpRequest object.
- A request is made to the server. This might be a call to a servlet, a CGI script, or any server-side script (PHP or ASP.NET).
- The server can return anything, including access a data store or even another system response (say, the billing system or the HR system) and return the proper response to client.

Response in Ajax

- Whenever a response is returned from web server ,the **XmlHttpRequest** callback invokes a JavaScript function to process the response asynchronously /synchronously
- The response processing in browser can modify/update the DOM structure
- The users are no longer captive to the request/response model on internet where browser sends the request and wait for the response!!!
- IFRAME components in html can achieve the results in similar way.

Set Up the call to server

- **void open (string httpMethod, string url, boolean async, string username, string password):**
- This method sets up your call to the server. It is meant to be the method of configuring a request. It has two required arguments and three optional arguments. You are required to supply the http method you are invoking (GET, POST, or PUT) and the URL of the resource you are calling.
- You may optionally pass a Boolean indicating whether this call is meant to be asynchronous or synchronous . The default is true, which means the request is asynchronous in nature.
- If the flag is false, processing waits until the response returns from the server. (The call becomes synchronous), .
- The last two parameters specify a username and password if required.

XHR:Send the request

- **void send(content):**
- This method actually makes the request to the server. If the request was declared as *asynchronous*, this method returns immediately, otherwise it waits until the response is received.
- The optional argument, the contents can be an instance of a DOM object, an input stream, or a string value to be sent to server.
- The content passed to this method is sent as part of the http request body and uploaded with http post method.

CallBack Function

- In asynchronous mode. the XmlHttpRequest object is configured to call the function defined by *handleStateChange* property when the request processing returns.
- This function checks the readyState property on the XmlHttpRequest object and then looks at the status code returned from the server. This becomes a callback function.
- Provided everything is as expected, the callback function processes the response on client side and displays the response whenever response is returned in an asynchronous manner.

XHR : Other methods

- **void abort():** stops the current request.
- **string getAllResponseHeaders():**
- It returns a string containing response headers from the HTTP request. Headers include Content-Length, Date, and URI.

Set and get current Request properties

- **void setRequestHeader(string header, string value):**
- This method sets a value for a given header value in the HTTP request. It takes a string representing the header to set and a string representing the value to place in the header.
- This method must be called after a call to open(…) method.
- **string getResponseHeader(string header):**This takes an argument representing the specific header name you want, returning its value as a string.

Ajax advantage

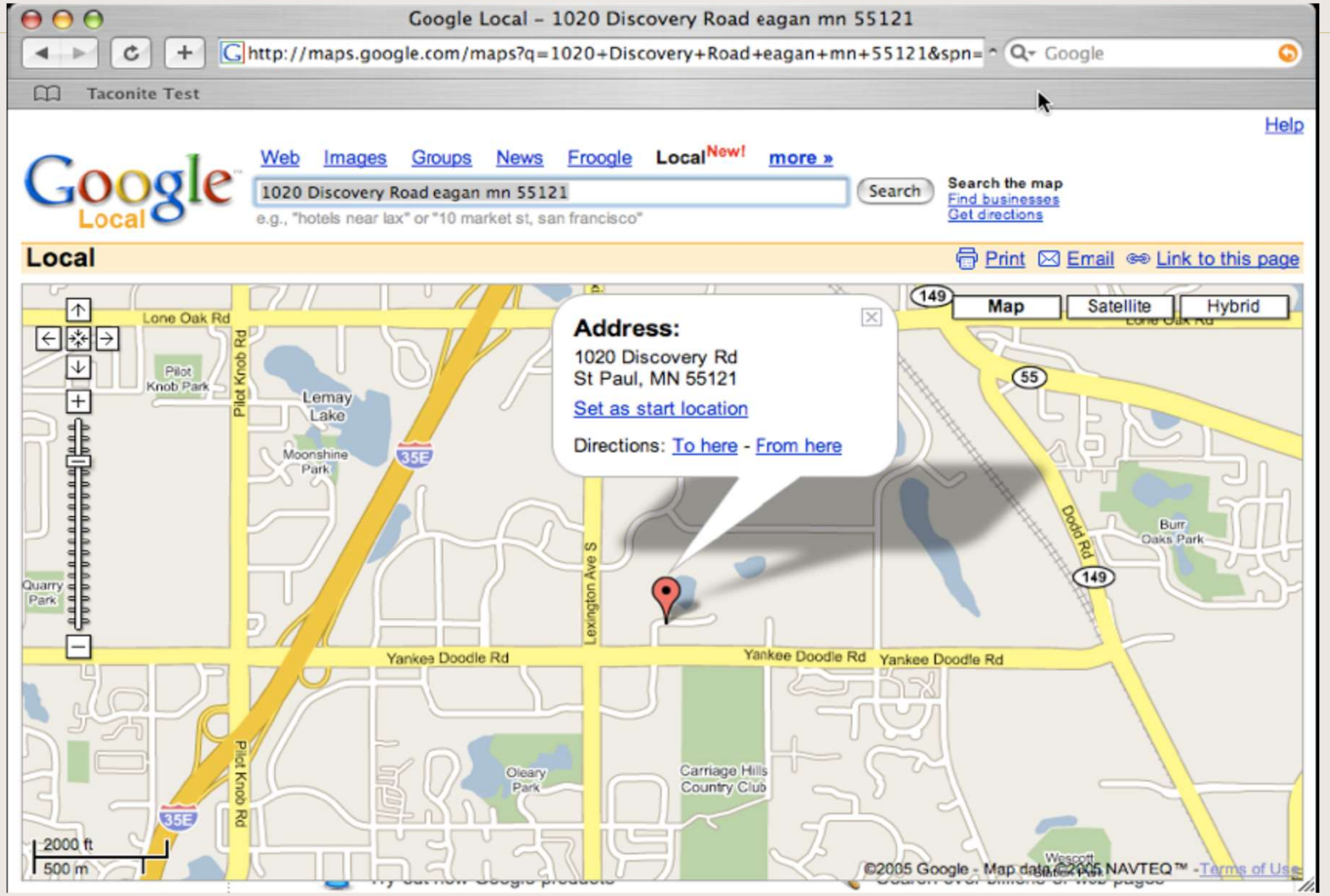
- Ajax changes the request/response model in the web applications.
- When the Ajax engine in browser receives the server response data, it starts into action, parsing the data and making changes in the user interface graphics based on the information returned, *asynchronously*.
- Because this process involves transferring less information than the traditional web applications, user graphics updates are faster, and the user is able to do the work more quickly.

Who Is Using Ajax?

- ***Google Search(***(www.google.com/).
- ***Gmail***
- ***Google Maps***
- ***The Amazon search engine (***(www.a9.com)
- ***Yahoo! News*** (http://news.yahoo.com)
- E-Mail attachments process in mail servers

Google Search Auto-complete

Google Maps


Google Local – 1020 Discovery Road eagan mn 55121
http://maps.google.com/maps?q=1020+Discovery+Road+eagan+mn+55121&spn=
Google
Taconite Test
Help
Google Local
Web Images Groups News Froogle Local New! more »
1020 Discovery Road eagan mn 55121
Search
e.g., "hotels near lax" or "10 market st, san francisco"
Search the map
Find businesses
Get directions
Local
Print Email Link to this page
Map Satellite Hybrid
Address:
1020 Discovery Rd
St Paul, MN 55121
Set as start location
Directions: To here - From here
Lone Oak Rd
Pilot Knob Park
Pilot Knob Rd
Lemay Lake
Moonshine Park
35E
Quarry Park
Lexington Ave S
Dodd Rd
Burr Oaks Park
149
55
Yankee Doodle Rd
Yankee Doodle Rd
Yankee Doodle Rd
Oleary Park
Carriage Hills Country Club
2000 ft
500 m
©2005 Google - Map data ©2005 NAVTEQ™ - Terms of Use

Ajax Usage!

- Start small code and test with users!
- Client side validation is a good first step
- Support Auto complete on client side
- More dynamic tool tips in browser
- Partial page updates in browser
- Recalculate the processing
- **It's all about the User HONEY!**

Be ware of Ajax!

- Un linkable pages - “Click To Load” options
- Broken back button
- Code bloat (Code redundancy)
- Graceful fallback - older browsers not supporting XHR objects.
- Ajax breaks the established GUI conventions
- Ajax doesn’t provide any “Loading” clues to user.

Un linkable page

- ***Unlinkable pages***: You have noticed that the address bar doesn't change even when the page does. When you use the XMLHttpRequest object to communicate with the server, user never need to modify the URL displayed in the address bar.
- It also means your users cannot bookmark your page or send the URL to their friends (think about maps or driving directions).
- If links are key to your application or site, be aware that Ajax makes this a bit of a challenge

No Visual Clues

- ***Lack of visual cues***: Since the entire page doesn't repaint, users may not perceive that anything has changed.
- That's why the Fade Anything Technique (FAT) was created.
- Gmail uses a "Loading" icon to indicate that it is doing some work.
- Depending on your application, you may have to add some sort of indication so your users know what is happening or going to happen…

Back to pavilion ?

- ***The broken Back button***: Some web applications deliberately disable the browser's '*Back*' button, In Ajax, clicking the Back button isn't going to do much of anything.
- If your users are expecting the Back button to work and you're using Ajax to manipulate parts of the page, you may have some problems to solve. (Ajax frameworks such as Dojo provide solution for this issue.)

Code Redundancy

- ***Code bloat***: The JavaScript that powers Ajax applications runs locally in the browser client.
- Although many have powerful machines with reams of random access memory (RAM), some users still have older machines that just don't offer this horsepower.
- If you put too much JavaScript into your application, you may find sluggish response times on the client side.
- Even if the JavaScript runs fine, more JavaScript means larger and larger pages, which means longer download times. try keeping JavaScript code to a minimum.

How many calls ?

- ***Death by a thousand cuts***: With the ease of making asynchronous calls, Ajax applications can get a bit chatty. You shouldn't add Ajax simply for the sake of adding Ajax.
- You need to think about each call you make to the server. Making a large number of fine-grain calls can have rather interesting impacts on your server architecture, so you'll want to spend some quality time with your favorite load-testing tool.
- Use Ajax in a complex page with caution.

Expose and Forget ?

- *Exposing the business layer*: The JavaScript code is transmitted to the client from server and (although you can certainly obfuscate the code,) if someone really wants to see what you did, they will. With that in mind, be very careful with what you show in your JavaScript. Don't expose the business layer functionality in Java Script code.
- *What about security*: Ajax doesn't present any new security vulnerabilities.All the standard security issues present in a regular web application still exist in an *Ajaxified* application.

Partial Update

- **Partial Page Update :**one of the real strengths of Ajax is that you no longer need to repaint the entire page; instead, you can just modify what has changed.
- This can be used in conjunction with FAT or Auto Refresh, and can be helpful for web applications.

How the user use it ?

- *Breaking established UI conventions*: Ajax lets developers create far richer web applications than they've created in the past.
- Just because you can do something doesn't mean you should.
- Your snappy new Ajax feature may not be obvious to your users.
- The user can simply drag and drop components to change their order, but after years of conditioning on web applications, many users might never have realized they could.

Ajax Conventions

- **Minimum traffic:** Ajax applications should send and receive as little information as possible to and from the server. In short, Ajax can minimize the amount of traffic between the client and the server. So the application becomes robust.
- **No surprises:** Ajax applications typically introduce different user interaction models than traditional web applications. As opposed to the web standard of click-and-wait, some Ajax applications use other user interface interactions such as drag-and-drop or double-clicking. Whatever the user interaction model you choose, be consistent so that the user knows what to do next.

JSON

Prakash Badhe

Mail : vishwasoft@yahoo.com

JavaScript To Java ?

- During communication between JavaScript code to java and from java back to JavaScript the code can exchange/transfer some standard i.e. character or number values.
- Same is the case for communication between JavaScript code and any other server side programming platform like C++, Pearl, VB.Net etc.
- If these OOP programming applications running on server side wants to exchange some objects with JavaScript on client side or JavaScript objects from client to server , how it can be done?
- Compatibility, object formats/structures are the issues.

JSON is the Answer..

- **JSON** (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate.
- It is based on a subset of the JavaScript Programming Language, Standard ECMA version
- JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others.
- These properties make JSON an ideal data-interchange language.

JSON Structure

- JSON is built on two structures:
- A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.
- These are universal data structures.
- In JSON an object is an unordered set of name/value pairs. An object begins with { (left brace) and ends with } (right brace). Each name is followed by : (colon) and the name/value pairs are separated by , (comma).

JSON Conversion

- Since JSON is the standard format , every object oriented or procedural language has some set of libraries to convert JSON objects (JavaScript) to the particular language or from language to Java script objects conversion.
- Java script to java and Java to Java script conversion is done by JSON-java libraries.

JSON-Java usage

- JSON-Java library provides conversion from JSON objects to Java language objects format.
- Java-JSON library supports onversion from Java objects to JSON objects format.
- Similarly there will be conversion between JavaScript and other languages.