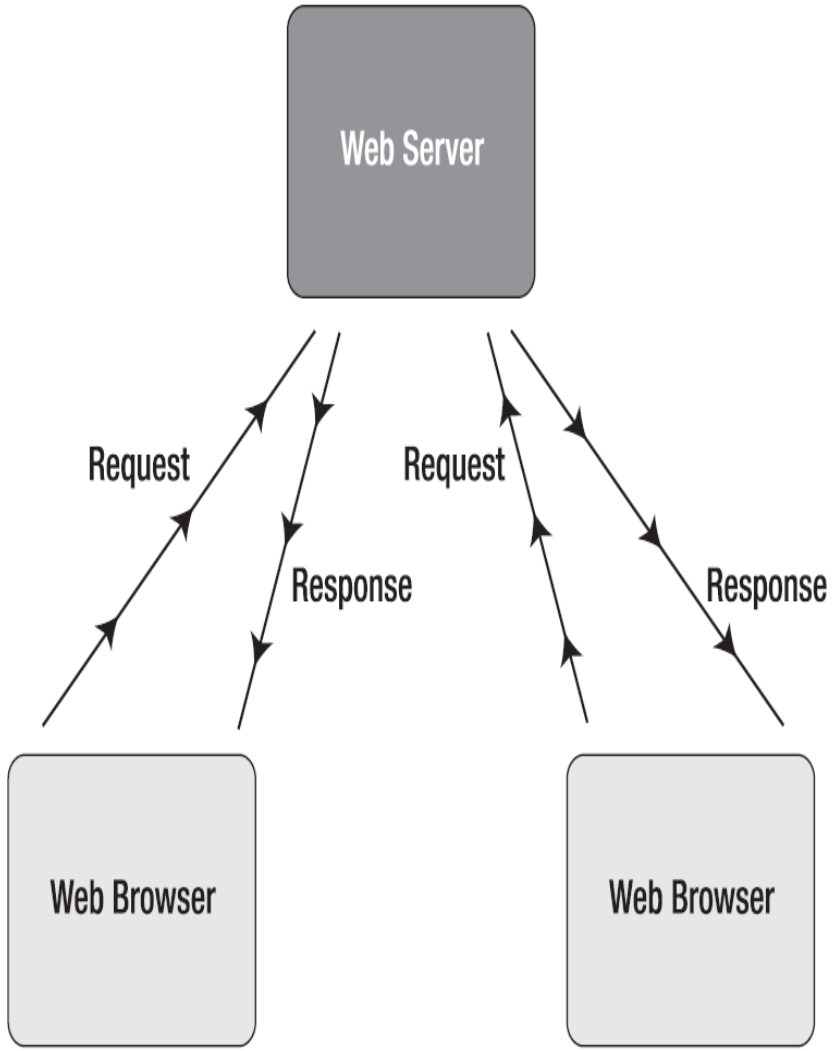
PHP and Ajax
Lecture-9

By: Ashiqullah Alizai
Herat University
Computer Science Faculty

Alizai.csf@hotmail.com

A Basic Web Request

- Client (Browser)
  - Submit single web request
  - Receive & render HTML
- Server
  - Receive request
  - Return HTML
- Static
  - One request,
  - no further interaction



HTTP Request and Response

- browser uses to send out a request
- a web site receive a response from the web server that is currently in charge of returning the response.
- when a request is sent, **certain headers** and **method** are passed along that allow the web server to know exactly what it is to be serving and how to handle the request.
- Once a request has been received, the server then decides what response to return.
  - Server return the response codes.

HTTP Request Methods

Method	Description
GET	The most common means of sending a request; simply requests a specific resource from the server
HEAD	Similar to a GET request, except that the response will come back without the response body; useful for retrieving headers
POST	Allows a request to send along user-submitted data (ideal for web-based forms)
PUT	Transfers a version of the file request in question
DELETE	Sends a request to remove the specified document
TRACE	Sends back a copy of the request in order to monitor its progress
OPTIONS	Returns a full list of available methods; useful for checking on what methods a server supports
CONNECT	A proxy-based request used for SSL tunneling

HTTP Response Codes

Code	Description
200 OK	This response code is returned if the document or file in question is found and served correctly.
304 Not Modified	This response code is returned if a browser has indicated that it has a local, cached copy, and the server's copy has not changed from this cached copy.
401 Unauthorized	This response code is generated if the request in question requires authorization to access the requested document.
403 Forbidden	This response code is returned if the requested document does not have proper permissions to be accessed by the requestor.
404 Not Found	This response code is sent back if the file that is attempting to be accessed could not be found (e.g., if it doesn't exist).
500 Internal Server Error	This code will be returned if the server that is being contacted has a problem.
503 Service Unavailable	This response code is generated if the server is too overwhelmed to handle the request.

From CGI to Flash to DHTML

- **CGI (Common Gateway Interface)**:the first scripting language to truly allow web applications the freedom.
- Properties:
  - dynamic image creation
  - database management,
  - complex calculation,
  - and dynamic web content creation
- **Still not supported:**
  - moving web content
  - Macromedia

From CGI to Flash to DHTML

- **Flash:** It allows a web developer to create visually impressive "movies" that can function as web sites, applications, and more.
- **Cons:**
  - in order for Flash to function, a plug-in must be installed into your browser.
- **DHTML(Dynamic HyperText Markup Language):** Combines HTML and CSS elements with JavaScript in an attempt to make things happen in your web browser dynamically.

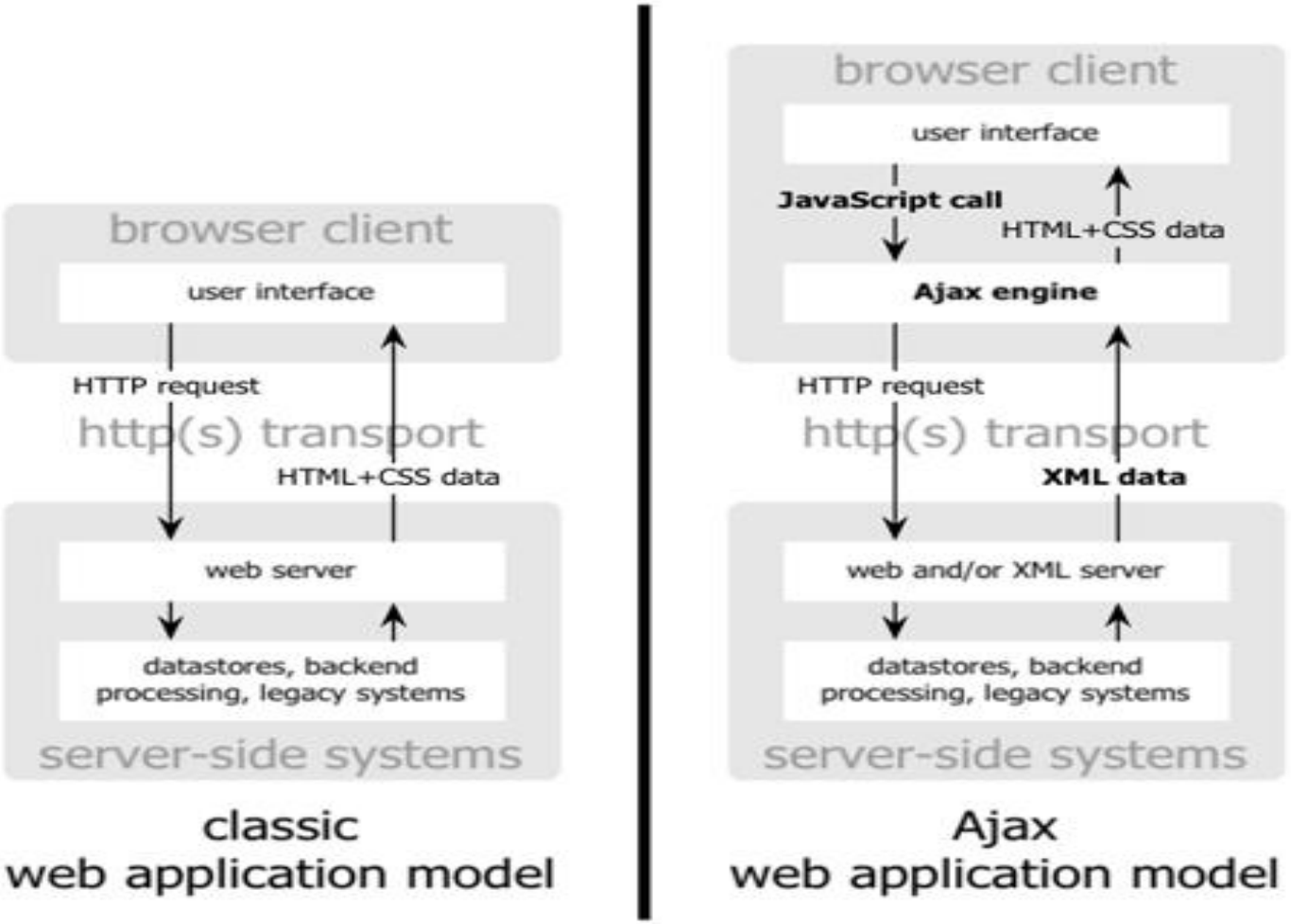
What is Ajax

- Asynchronous
- JavaScript
- And
- XmlHttpRequest (XHR)
  - Some use XML, but it is not exact
  - Examples of applications using AJAX: Google Maps, Gmail, Youtube, and Facebook tabs.

Ajax overviews

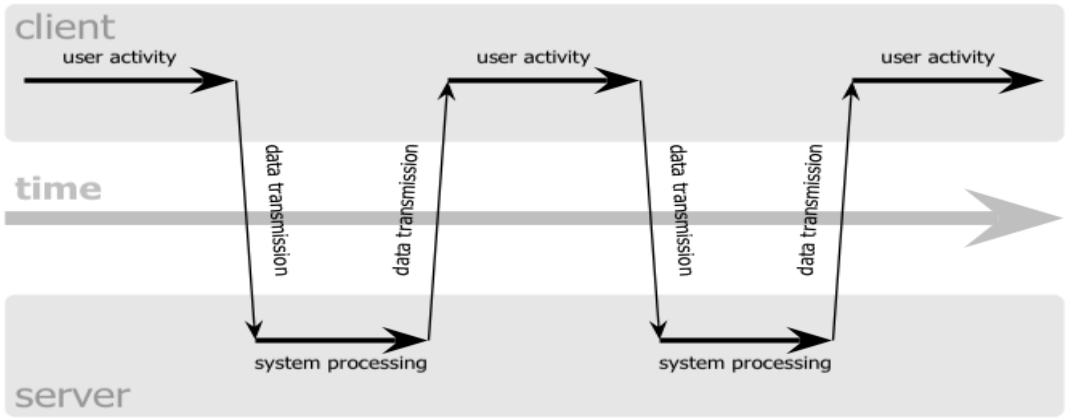
- It is not a new programming language, but a new way (technique) to use existing standards
  - To create better, faster, and more user-friendly and interactive web applications
- Based on JavaScript and HTTP requests
  - Uses JavaScript as its programming language
  - Uses the XMLHttpRequest object to communicate directly with the server
  - Trades data with a web server without reloading the page
- Uses asynchronous data transfer (via HTTP requests) between the browser and the web server
  - Allowing web pages to request small bits of information from the server instead of whole pages
- It is a browser technology
- It is independent of web server software

Classic web application Vs Ajax web Application

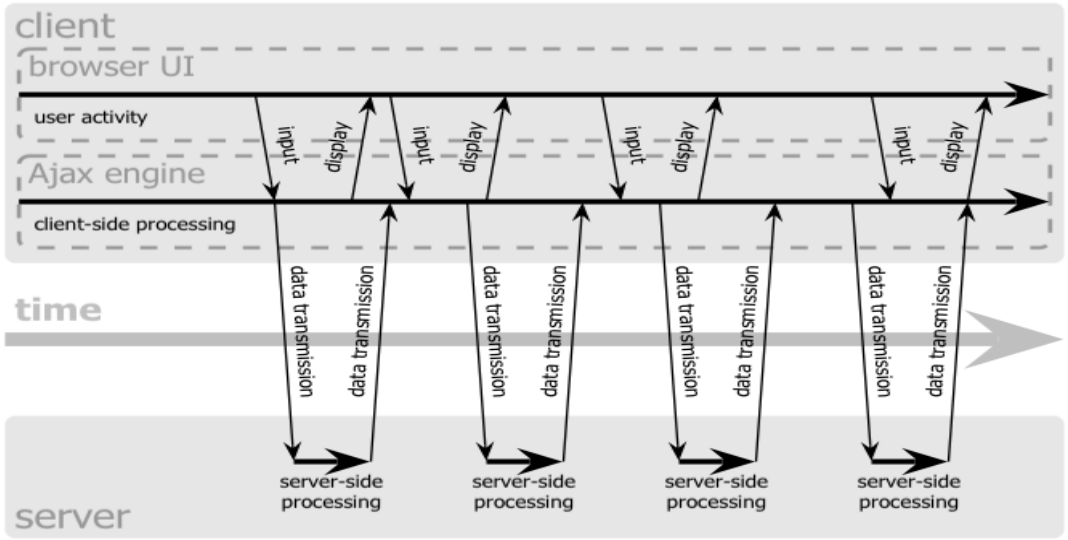


Synchronous Vs Asynchronous

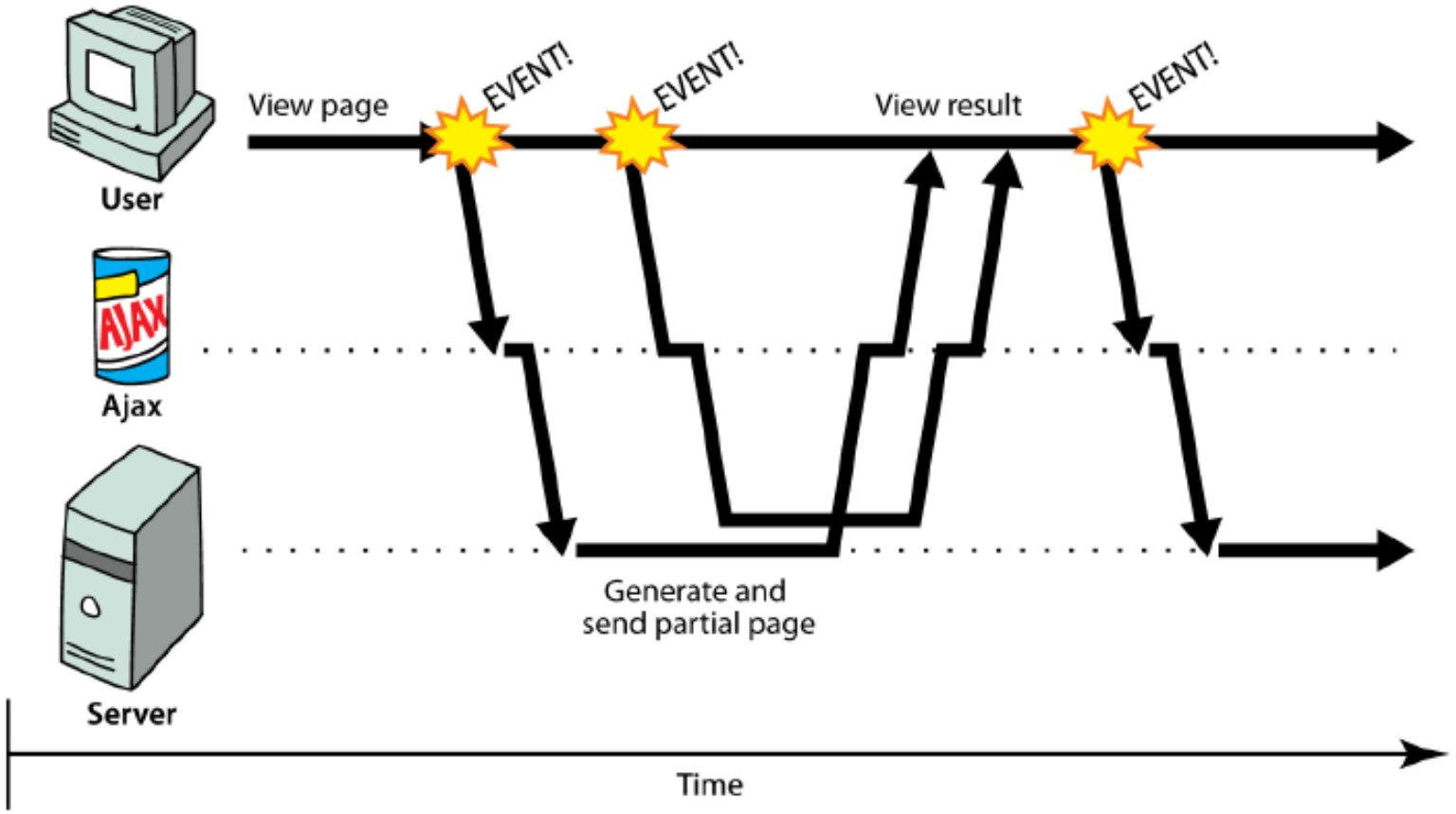
classic web application model (synchronous)



Ajax web application model (asynchronous)



Using Ajax engine



Implementing AJAX

- To implement AJAX we need to answer three questions:
  - What triggers the AJAX request?
    - Usually a JavaScript event (onblur, onclick, etc.)
  - What is the server process that handles the AJAX request and issues the response?
    - Some kind of URL (use a Service Locator)
  - What processes the response from the server(what is the callback method)?
    - A JavaScript function that gets the response and manipulates the DOM, based on the text returned.

Implementing AJAX

- **AJAX is Based on Internet Standards**
- **AJAX is based on internet standards, and uses a combination of:**
  - XMLHttpRequest object (to exchange data asynchronously with a server)
  - JavaScript/DOM (to display/interact with the information)
  - CSS (to style the data)
  - XML (often used as the format for transferring data)
  - AJAX applications are browser- and platform-independent!

XmlHttpRequest Object (XHR)

- The Heart of AJAX
- XMLHttpRequest object (to exchange data asynchronously with a server)
- <First implemented in IE in 1997 as part of the new DHTML standard
- Response comes in one of two properties:
  - responseXML – Returns a DOM document (can use functions such as, `getElementById()`)
  - responseText – A text string (can be HTML, or even JavaScript code)

XMLHttpRequest Methods

Method	Description
abort()	Cancels the current request
getAllResponseHeaders()	Returns all HTTP headers as a String type variable
getResponseHeader()	Returns the value of the HTTP header specified in the method
open()	Specifies the different attributes necessary to make a connection to the server; allows you to make selections such as GET or POST (more on that later), whether to connect asynchronously, and which URL to connect to
setRequestHeader()	Adds a label/value pair to the header when sent
send()	Sends the current request

XMLHttpRequest Properties

- Receiving data (handle response)
  - Onreadystatechange
  - readyState
    - responseText
    - responseXML

onreadystatechange

- Defines a function to receive data returned by the server after a request is sent
- Must be set before sending request
- The following code defines a function for this purpose (with an empty body for now)

```
var xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function()
{
// code for receiving response data
}
```

readyState

- This property holds the status of the server's response
- Each time the readyState changes, the onreadystatechange function will be executed

➢ State description

1. 0 The request is not initialized
2. 1 The request has been set up
3. 2 The request has been sent
4. 3 The request is in process
5. 4 The request is complete

Update the Function

```
xmlhttp.onreadystatechange=function()
{
if (xmlhttp.readyState==4)
{
// Get the data from the server's response
}
}
```

responseText

- Retrieve text data returned by the server
  - Type: DOMString (readonly)

```
xmlhttp.onreadystatechange=function()
{
if (xmlhttp.readyState==4)
{
document.getElementById('formentry').value =
xmlhttp.responseText;
}
}
```

XMLHttpRequest Methods

- Asking for data (send request)
  - open()
  - Two required arguments
    - method (GET, POST, PUT, DELETE, HEAD, OPTION)
    - server-side URI
  - send()
  - One argument
    - data to be sent (DOMString or Document)
    - **null for GET**
    - can be omitted

Execute the Ajax Function

- Want it to run "behind the scenes"

```
<script type="text/javascript">
function myajax()
{ . . . /* all of the code from before */ . . . }
</script>
<form>
<input type="text" onkeyup="myajax();"
name="userdata" />
<input type="text" id="formentry" />
</form>
```

The Basic Ajax Process

- JavaScript
  - **Define an object for sending HTTP requests**
  - **Initiate request**
    - **Get request object**
    - **Designate a request handler function**
  - **Supply as onreadystatechange attribute of request**
    - **Initiate a GET or POST request**
    - **Send data**
    - **Handle response**
  - **Wait for readyState of 4 and HTTP status of 200**
  - **Extract return text with responseText or responseXML**
  - **Do something with result**

The Basic Ajax Process (cont'd)

- HTML
  - Loads JavaScript
  - Designates control that initiates request
  - Gives ids to input elements that will be read by script

Define a Request Object

```
var request;
  function getRequestObject() {
      if (window.XMLHttpRequest) {
          return(new XMLHttpRequest());
      } else {
      return(null);
      }
  }
```

Initiate Request

```
function sendRequest() {
   request = getRequestObject();
   request.onreadystatechange =
   handleResponse;
        request.open("GET", "message-data.html“, true);
   request.send(null);
}
```

Handle Response

```
function handleResponse() {
    if (request.readyState == 4) {
        alert(request.responseText);
    }
}
```

Cross-Browser XMLHttpRequest

```
function getRequestObj() {
    var xmlHttp;
    try { // Firefox, Opera 8.0+, Safari
        xmlHttp=new XMLHttpRequest();
    } catch (e) { // Internet Explorer
        try {
            xmlHttp=new   ActiveXObject("Msxml2.XMLHTTP");
        } catch (e) {
            try {
            xmlHttp=new  ActiveXObject("Microsoft.XMLHTTP");
            } catch (e) {alert("Your browser does not support
AJAX!");            return null;}}}    return xmlHttp; }
```

PHP And Ajax

This is the end for this lecture