AJAX

• Rich Internet Application (RIA)

- ✓ RIA is a web application designed to deliver the same features and functions normally associated with desktop applications
- ✓ Applications developed to provide exciting, rich interactivity features in software program are known as RIA
- ✓ Offers users a better visual experience and more interactivity than traditional browser applications that use only HTML and HTTP
- ✓ Built with powerful modern development tools that enable to include mixed media like several fonts, bit-map, vector graphic files, animations, online conferencing, audio and video
- ✓ RIAs can run faster and be more engaging

• Rich Internet Application (RIA)

- ✓ RIAs generally split the processing across the Internet/network by locating the user interface and related activity on the client side, and the data manipulation and operation on the application server side

- ✓ RIA normally runs inside a Web browser and usually does not require software installation on the client side to work. However, client machine operation requires installation of a platform - such as Adobe Flash, Java or Microsoft Silverlight

- ✓ For security purposes, most RIAs run their client portions within a special isolated area of the client desktop called a sandbox

- ✓ Sandbox limits visibility and access to the file and operating system on the client to the application server on the other side of the connection

- ✓ This approach allows the client system to handle local activities, calculations, reformatting and so forth, thereby lowering the amount and frequency of client-server traffic

- **Rich Internet Application (RIA)**

✓ One distinguishing feature of an RIA (in contrast to other Web-based applications) is the client engine that intermediates between the user and the application server

✓ Client engine downloads when the RIA launches

✓ The engine can be augmented during subsequent operation with additional downloads in which the engine acts as a browser extension to handle the user interface and server communications.

RIA TOOLS

- Ajax
- Dojo
- Adobe Flex
- MS Silverlight
- JavaFX
- Ruby on Rails
- Script.aculo.us
- JavaServer Faces
- ASP.NET Ajax

CHARACTERISTICS OF RIA

1. Direct interaction
2. Partial update of web pages
3. Better feedback to users
4. Consistency of look and feel
5. Offline use of application
6. Performance
7. Security

• Characteristics of RIA – Direct Interaction

- ✓ In a traditional page-based Web application, interaction is limited to a small group of standard controls: checkboxes, radio buttons and form fields.
- ✓ This severely hampers the creation of usable and engaging applications
- ✓ An RIA can use a wider range of controls that allow greater efficiency and enhance the user experience
- ✓ For example, users can interact directly with page elements through editing or drag-and-drop tools

• Characteristics of RIA – Partial page updating

✓ Standard HTML-based Web pages are loaded once

✓ If you update something on a page, the change must be sent back to the server, which makes the changes and then resends the entire page. There's no other way to do it with HTTP and HTML

✓ With traditional Web-based apps, network connectivity issues, processing limitations and other problems require users to wait while the entire page reloads

✓ RIA's overcomes the above issues by providing real time streaming, high performance virtual machines at client side, local caching mechanisms which helps to decrease latency and enhance responsiveness

• Characteristics of RIA – Better feedback

✓ Because of their ability to change parts of pages without reloading, RIAs can provide the user with the following:

- fast and accurate feedback
- real-time confirmation of actions and choices
- informative and detailed error messages

• Characteristics of RIA – Consistency of look and feel

✓ Internet is wide spread and used across various operating systems and web browsers

✓ Consistency of look and feel in the standard HTML applications has been an issue

✓ With RIA tools, the user interface and experience with different browsers, devices and operating systems can be more carefully controlled and made consistent

• Characteristics of RIA – Enhanced accessibility

- ✓ Earlier, Internet users used to communicate by exchanging text based emails
- ✓ With HTML and improvement in world wide web, users started to look for enhanced graphics based web pages
- ✓ These applications are static and the content on the web pages are formatted in advance
- ✓ Static applications failed when accessing features that include data manipulation, interacting with sophisticated business logic
- ✓ Enhanced accessibility was provided through specialized software applications called as RIA, which can run on user's computer and network

- **Characteristics of RIA – Improved features**

✓ RIA allows programmers to embed various functionalities in graphics based web pages that look more fascinating and engaging like desktop applications

✓ Provides more complex application screens on which various mixed media including different fonts, vector graphics and bitmap files, online conferencing are presented by using different modern development tools

• Characteristics of RIA – Offline use

- ✓ RIA's are able to retain their current state on the client side
- ✓ Helps the users to use RIAs even if Internet connection is lost
- ✓ Nowadays, it is possible for some standard Web/HTML applications also to retain the current state in offline condition

- **AJAX**

✓ AJAX stands for **A**synchronous **J**avaScript **A**nd **X**ML

✓ AJAX is a technique for creating **better**, **faster**, **dynamic** and more **interactive** web applications

✓ Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display

✓ XMLHttpRequest is a JavaScript object that performs asynchronous interaction with the server.

✓ Conventional web applications transmit information to and from the sever using synchronous requests. It means you fill out a form, hit submit, and get directed to a new page with new information from the server

✓ With AJAX, when you hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen

✓ XML is commonly used as the format for receiving server data, although any format, including plain text, can be used

- **AJAX**

✓ AJAX is the most viable Rich Internet Application (RIA) technology

✓ It is a web browser technology independent of web server software

✓ User can continue to use the application while the client program requests information from the server in the background

✓ Intuitive and natural user interaction. Clicking is not required, mouse movement is a sufficient event trigger

✓ Data-driven as opposed to page-driven

- **AJAX**

✓ Update a web page without reloading the page

✓ Request data from a server - after the page has loaded

✓ Receive data from a server - after the page has loaded

✓ Send data to a server - in the background

AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

• AJAX - XMLHttpRequest Object

✓ Used to exchange data with a server making it possible to update parts of a web page, without reloading the whole page

✓ Syntax for creating XMLHttpRequest object

***variable* = new XMLHttpRequest();**

• AJAX - XMLHttpRequest Object Methods

Method	Description
abort()	Cancels the current request
getAllResponseHeaders()	Returns header information
getResponseHeader()	Returns specific header information
open(*method,url,async,user,psw*)	Specifies the request *method*: the request type GET or POST *url*: the file location *async*: true (asynchronous) or false (synchronous) *user*: optional user name *psw*: optional password **Note: Synchronous XMLHttpRequest (async = false) is not recommended because the JavaScript will stop executing until the server response is ready. If the server is busy or slow, the application will hang or stop.**
send()	Sends the request to the server. Used for GET requests.
send(*string*)	Sends the request to the server. Used for POST requests.
setRequestHeader()	Adds a value pair to the header to be sent

• AJAX - XMLHttpRequest Object Properties

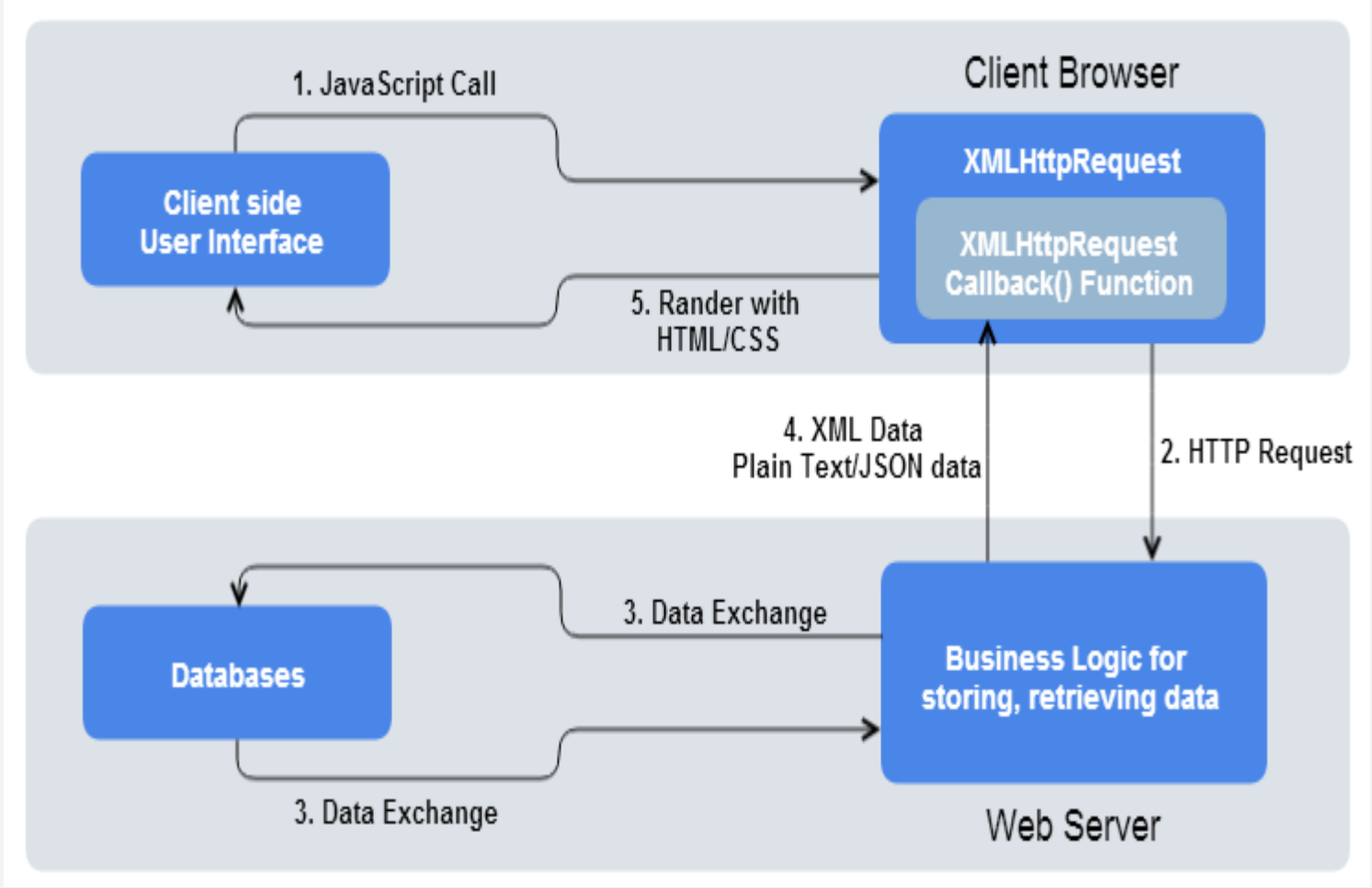
Properties	Description
onreadystatechange	Defines a function to be called when the readyState property changes
readyState	Holds the status of the XMLHttpRequest. 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
responseText	Returns the response data as a string
responseXML	Returns the response data as XML data
status	Returns the status-number of a request 200: "OK" 403: "Forbidden" 404: "Not Found"
statusText	Returns the status-text (e.g. "OK" or "Not Found")

- **Create a XMLHttpRequest and retrieve data from .txt file**

```
<html> <body>
<h1>XMLHttpRequest Object</h1>
<button type="button" onclick="loadDoc()">Change Content</button>
<script>
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML =
      this.responseText;
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
}
</script>
</body> </html>
```

- onreadystatechange function is called every time the readyState changes.
- When readyState is 4 and status is 200, the response is ready.

- **How AJAX reloads part of the web page without reloading the entire page? OR**
- **Working of AJAX.**



- **How AJAX reloads part of the web page without reloading the entire page? OR**
- **Working of AJAX.**

✓Client performs an action through the UI to trigger the event that generates a call to the JavaScript function. JavaScript function creates XMLHttpRequest object, which specifies the JavaScript callback function.

✓JavaScript XMLHttpRequest object sends a asynchronous HTTP request to the server.

✓Web Server processes the request by interacting with the database using JSP, PHP, Servlet, ASP.NET etc. **and returns an XML file containing data**.

✓Server sends XML data or JSON data to the XMLHttpRequest callback function.

✓Client browser updates the HTML DOM (HTML, CSS) representing the web page along with new data.