AJAX

Prerequisites for AJAX

- Client side
  - Knowledge of HTML
  - JavaScript event handling
- Server side
  - Server side technology
    - E.g., Servlet/JSP programming
    - PHP
    - ASP.net
  - We will use Servlet/JSP

Rich Internet Applications

- Rich Internet Applications try to provide the characteristics of traditional desktop applications using browser
  - Responsive interface
  - Faster response time
  - Immediate feedback of user's interaction
- Rich Internet Applications differ from traditional web applications
  - While a user is interacting with a web-page, the web-page can retrieve information transparently based on events
  - User does not need to wait for response

Rich Internet Technologies

- Many technologies support Rich Internet Applications
  - Adobe Flash/Flex:
    - Client (browser) needs flash plugin
  - Silverlight
    - Client (browser) needs Microsoft Silverlight plugin
  - Applets
    - Client (browser) needs Java plugin (JRE)
  - DHTML (Dynamic HTML)
    - Uses JavaScritp. Support in all major browsers.
  - AJAX
    - JavaScript event
- Rich Internet Technologies are part of Web 2.0

Web 2.0

- Web 2.0 covers various ideas such as
  - Rich user experience applications
  - Radical (opinion based) content (Wikipedia)
  - Users' Participation in content
  - Users' input for PageRank, reviews, etc.
  - Decentralized content

What is AJAX?

- AJAX-Asynchronous JavaScript And XML
  - With user's perspective, AJAX provides asynchronous communication,
  - But in background it performs synchronous communication with server
  - When the response is received from the server, action to be done with retrieved data can be initiated.
- Update certain portion of a web-page dynamically, without refreshing whole page.
  - E.g., like/dislike a comment
  - Write a comment
- Can exchange small amount of data behind web-page transparently.
- It can increase application's interactivity, speed, and usability
- This term was first publicly used in 2005 by Jesse James Garret

Is AJAX dependent on server technology?

- AJAX is a web browser technology independent of web server software
- AJAX can be used with any server technology
  - PHP
  - JEE
  - ASP.net

Examples: uses of AJAX

- Google Search
  - Retrieve potential search matches as we type keywords
  - The web-page does not refresh, only suggestions are updated
  - While we are typing, the web page makes request behind the web-page using AJAX.
- Google Map
  - User can drag the map, no submit button
  - The web page transparently retrieves map(image tiles) data.
- Gmail

Use cases of AJAX

- Runtime (input form) data validation with server
  - E.g., check User IDs, serial numbers, postal codes, etc.
- Auto completion fields
  - Completing fields as user types
- Master detail operation
  - Based on user's input for a field, more detailed information can be retrieved for the detail field
  - E.g., Based on selection of state, possible cities of the state can be populated in City field.
- Advanced GUI controls
  - Provide controls without page refresh
  - Tree controls, menus, and progress bars

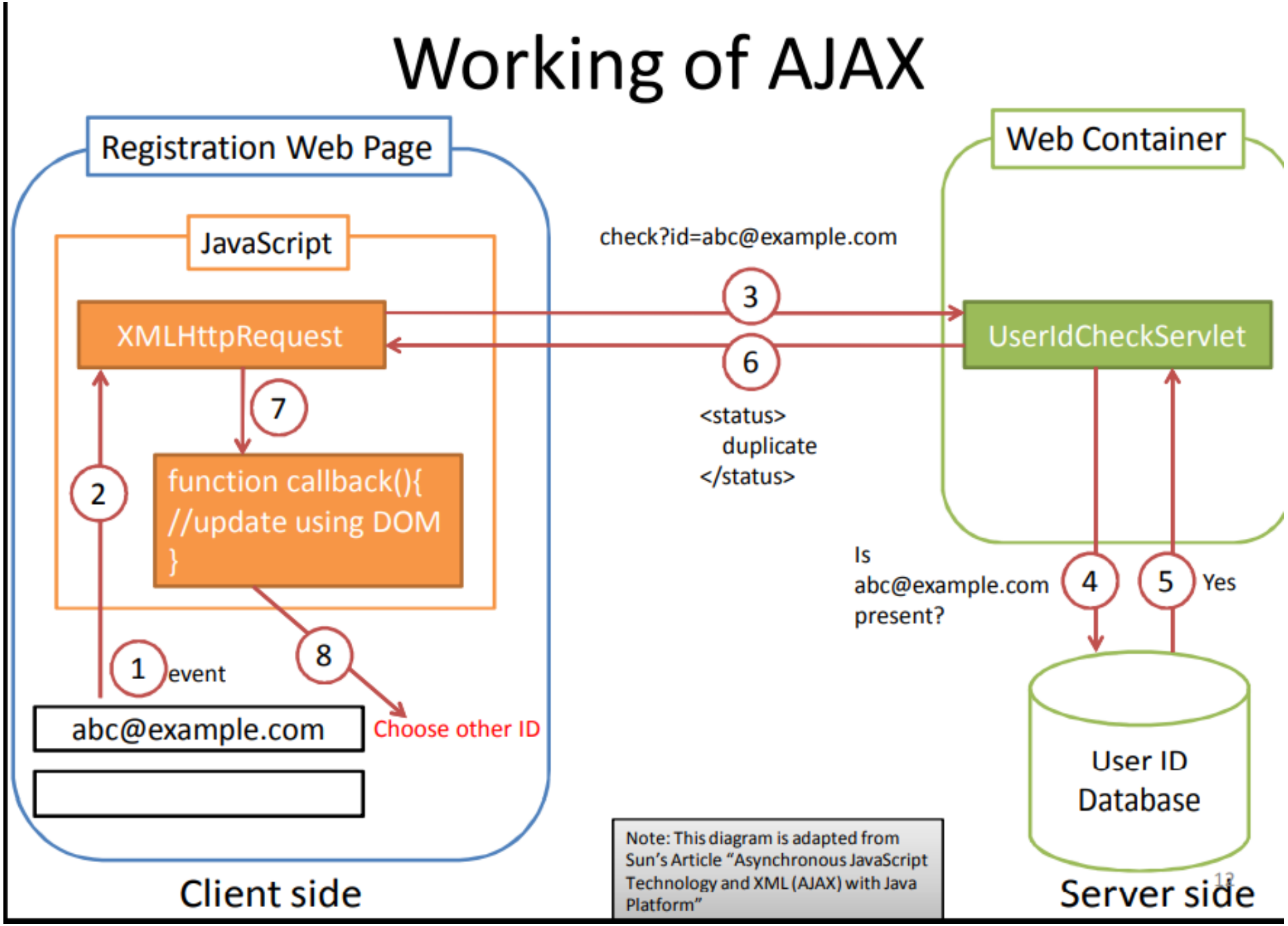
Use cases of AJAX

- Refreshing frequently changing data without page refresh
  - Cricket score
  - Stock prices
  - Weather information
  - Any such data
- Simulating server side notification
  - Simulate server side notification by polling the server in the background
  - E.g., gmail's email notification.

How to use AJAX?

- AJAX is not a standalone technology.
  - There is no specification or standard for it.
  - But, it relies on other standard based technologies
- Other technologies used with AJAX:
  - HTML: to structure data
  - CSS: to visualize data
  - JavaScript (for event handling)
  - XMLHttpRequest (JavaScript object to perform asynchronous communication)
  - Document Object Model: Allow access of structured content dynamically.
  - XML, JSON: Data format

Working of AJAX



Steps of AJAX working

1. A client event occurs.
2. An XMLHttpRequest object is created and the XMLHttpRequest object is configured.
3. The XMLHttpRequest object makes an asynchronous request to the Webserver (particular Servlet).
4. Web container (Servlet) delegates processing to service logic (e.g., database operation)
5. Web container (Servlet) receives result from the service
6. Servlet returns the result containing XML document.
7. The XMLHttpRequest object calls the callback() function and processes the result.
8. The callback() function updates the HTML DOM.

XMLHttpRequest

- It is JavaScript class supported by most browsers.
  - Short name XHR
- It allows to send HTTP request through JavaScript code
  - Communicates with a server via standard HTTP GET/POST
  - Does not interrupt user operation
- Received HTTP response is processed by JavaScript function (client side in browser)
  - Response could be of type
    - text/xml, text/plain, text/json, text/javascript
- Portability issue
  - Code to create XMLHttpRequest object varies among browsers.

Important properties of XMLHttpRequest

- onreadystatechange
  - An event handler for an event that fires at every state change
- readyState
  - It defines the current state of the XMLHttpRequest object
    - 0 = request is not yet initialized (uninitialized)
    - 1 = The request has been set up (loading)
    - 2 = request has been sent (loaded)
    - 3 = The request is in progress, responseText is partially set (interactive)
    - 4 = finished downloading response (complete)
- responseText
  - response as text; null if error occurs or ready state < 3

Important properties of XMLHttpRequest

- responseXML
  - response as DOM Document object; null if error occurs or ready state < 3
  - The XML HttpRequest object has an in-built XML parser. The responseXML contains the parsed XML.
- status
  - integer status code
    - 404 for page not found
    - 403 for Forbidden
    - 200 for OK
- statusText
  - Returns status as a string (e.g., “OK” or “Not Found”)

Important methods of XMLHttpRequest

- Main methods:
  - open(method, url[, async, userName, password]) – initializes a new HTTP request
    - method can be "GET", "POST", "PUT" or "DELETE“
    - url must be an HTTP URL (start with "http://")
    - async is a boolean indicating whether request should be sent asynchronously
      - defaults to true
    - userName and password: if web resource is password protected
  - send(body) – sends HTTP request to url (body could be null)
    - For GET request, it is null.
  - abort() – called after send() to cancel request

Important methods of XMLHttpRequest

- Header related methods:
  - void setRequestHeader(name, value)
  - String getResponseHeader(name)
  - String getAllResponseHeaders()
    - returns a string where “header: value” pairs are delimited by carriage returns

Understanding state changes when methods are called

- readyState = 0
  - After we have created the XMLHttpRequest object, but before we have called the open() method.
- readyState = 1
  - After we have called the open() method, but before we have called send().
- readyState = 2
  - After we have called send().
- readyState = 3
  - After the browser has established a communication with the server, but before the server has completed the response.
- readyState = 4
  - After the request has been completed, and the response data has been completely received from the server.