How AJAX Works?

Agenda

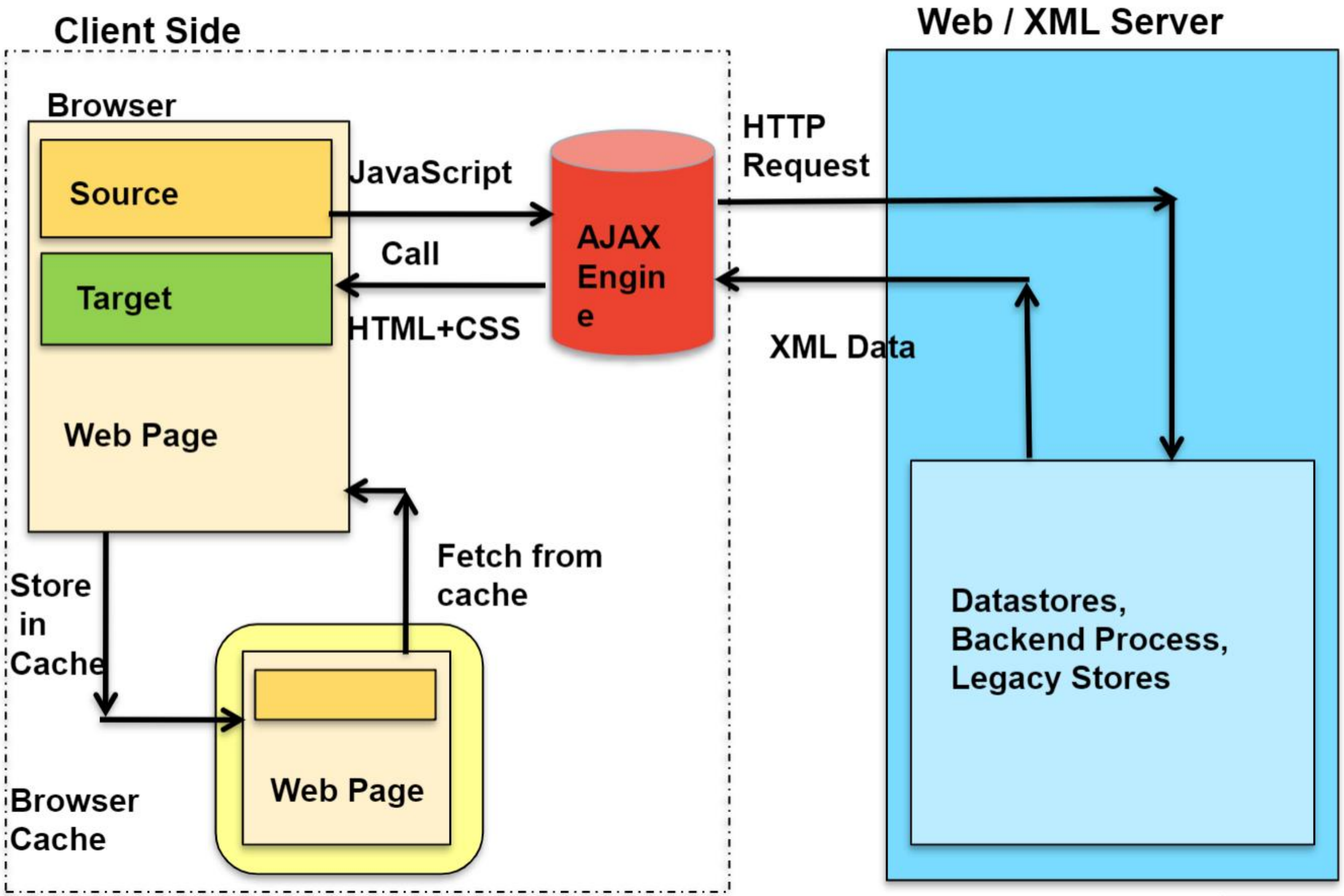
1 How Ajax Works!

Objectives

At the end of this module, you will be able to:

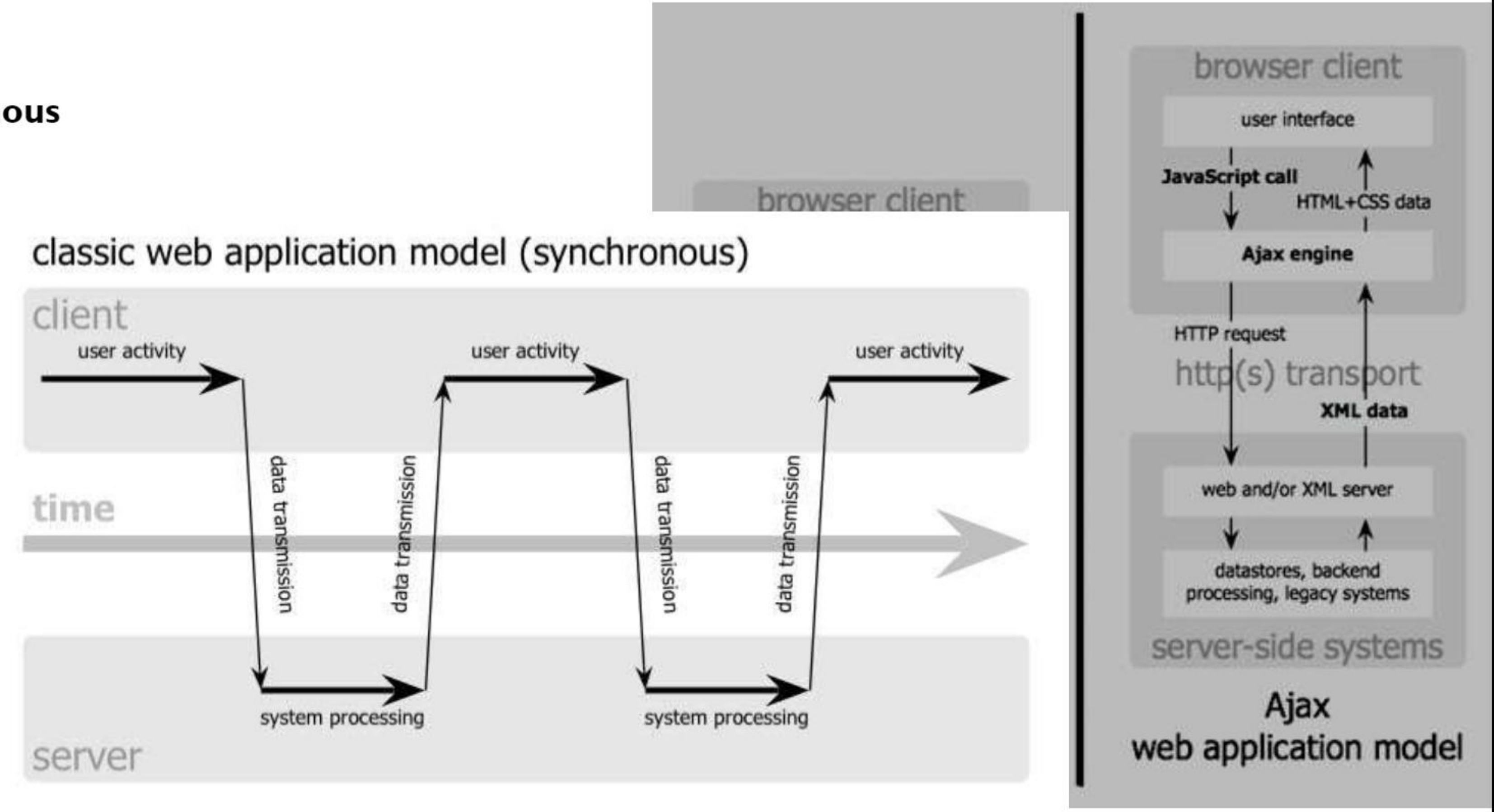
- Get to know how Ajax works
- Implement different Properties
- Send request to the server

AJAX – How it works?



Classic Web Application Model

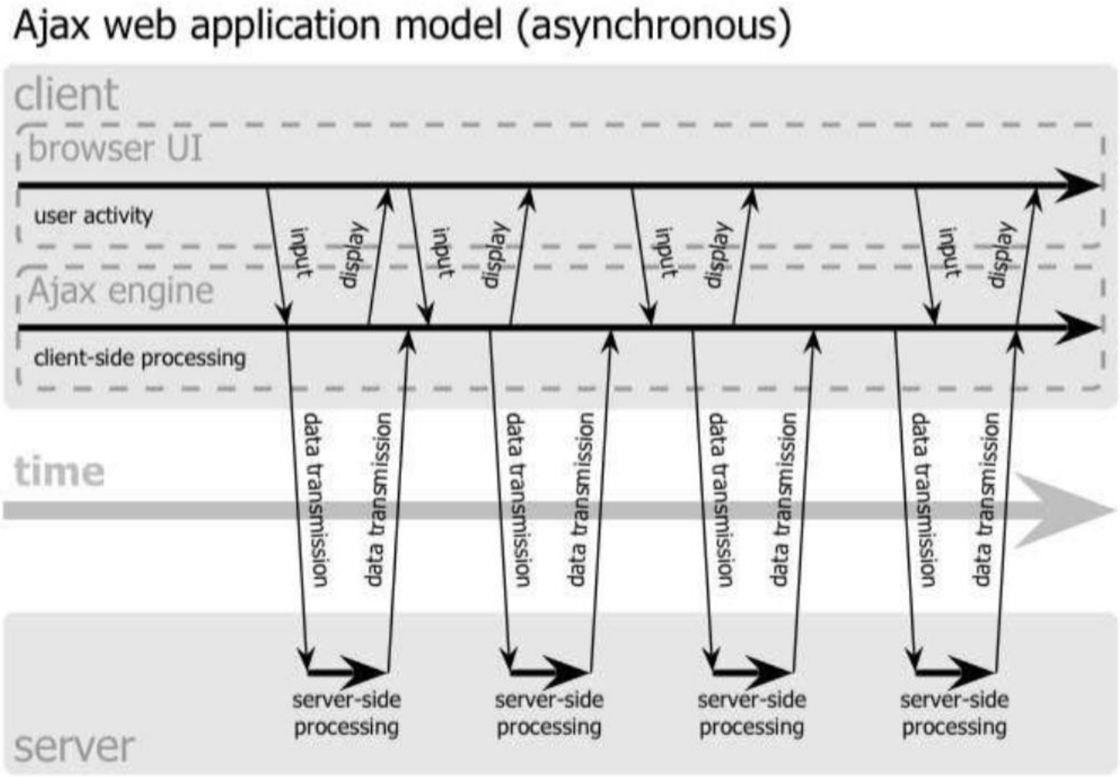
Synchronous



Source: James Garrett / adaptivepath.com

AJAX Web Application Model

Asynchronous



AJAX and HTTP requests

Contrary to the traditional JavaScript working,

- With AJAX, your JavaScript communicates directly with the server, through the JavaScript ***XMLHttpRequest object***.
- With an HTTP request, a web page can make a request to, and get a response from a web server - without reloading the page.
- The user will stay on the same page, and he or she will not notice that scripts request pages, or send data to a server in the background.

The XMLHttpRequest Object

- The main element of AJAX is the *XMLHttpRequest* Object.
- By using the XMLHttpRequest object, a web developer can update a page with data from the server after the page has loaded.
- The XMLHttpRequest Object is supported in various browsers like Internet Explorer 5.0+ , Safari 1.2 , Mozilla 1.0/Firefox, Opera 8+ and Netscape 7.
- Different browsers use different methods to create the XMLHttpRequest Object

  (Internet Explorer uses an *ActiveXObject*, while other browsers uses the built-in JavaScript object called *XMLHttpRequest*.

Example:

```
var xmlHttp;
xmlHttp=new XMLHttpRequest(); //firefox, safari , opera 8.0+
xmlHttp=new ActiveXObject("Msxml2.XMLHTTP");  //IE 6.0+
xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");  //IE 5.5+
```

The onreadystatechange Property

- Once a request is made to the server, the function receives the data returned by the server
- The ***onreadystatechange property*** stores the function that will process the response from a server
- The function that is stored in the onreadystatechange property is a function that is stored in the property to be called automatically

Example:

```
xmlHttp.onreadystatechange=function(){
    //some code for handling server response}
```

- The onreadystatechange property stores an empty function inside it

The readyState Property

- The ***readyState property*** holds the status of the server's response.
- The onreadystatechange function will be executed, when each time the readyState changes
- The possible values for the readyState property:

State	Description
0	Request not initialized
1	Request has been set up / Connection Established
2	Request sent / request received by server
3	Request under process / response is in process
4	Request completed / Server ready to generate response

The responseText Property

- The data sent back from the server can be retrieved with the *responseText property*.

Example:

```
xmlHttp.onreadystatechange=function()
{
    if (xmlHttp.readyState==4)
    {
    document.myFormname.myTextboxname.value=xmlHttp.responseText;
    }
  }
```

Sending a request to the server

- To send a request to the server, we use the ***open()*** and the ***send()*** methods
- The open() method:

  The open() method takes three arguments.

  1. The first argument defines which *method to use* when sending the request (GET or POST)
  2. The second argument specifies the *URL of the server-side script*
  3. The third argument takes a boolean which specifies that the *request should be handled asynchronously*
- The send() method - sends the request off to the server.

Example:

```
xmlHttp.open("GET",“server-side-component.jsp",true);
xmlHttp.send(null);
```

Summary

In this module, you were able to:

- Get to know how Ajax works
- Implement different Properties
- Send request to the server

References

- w3schools.com (2012). AJAX Introduction. Retrieved April 30, 2012, from, *http://www.w3schools.com/ajax/default.asp*

- Greg Murray (2005). Asynchronous JavaScript Technology and XML(Ajax) With the Java Platform. Retrieved April 30, 2012, from*, http://www.oracle.com/technetwork/articles/javaee/ajax-135201.html*

- Adaptive path (2012). Ajax: A New Approach to Web Applications. Retrieved May 2, 2012, from, *http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications*

Thank You