# AJAX
## Asynchronous Design in Web Apps

IT 4403 Advanced Web and Mobile Applications

Jack G. Zheng
Fall 2019

KENNESAW STATE UNIVERSITY

## Topics

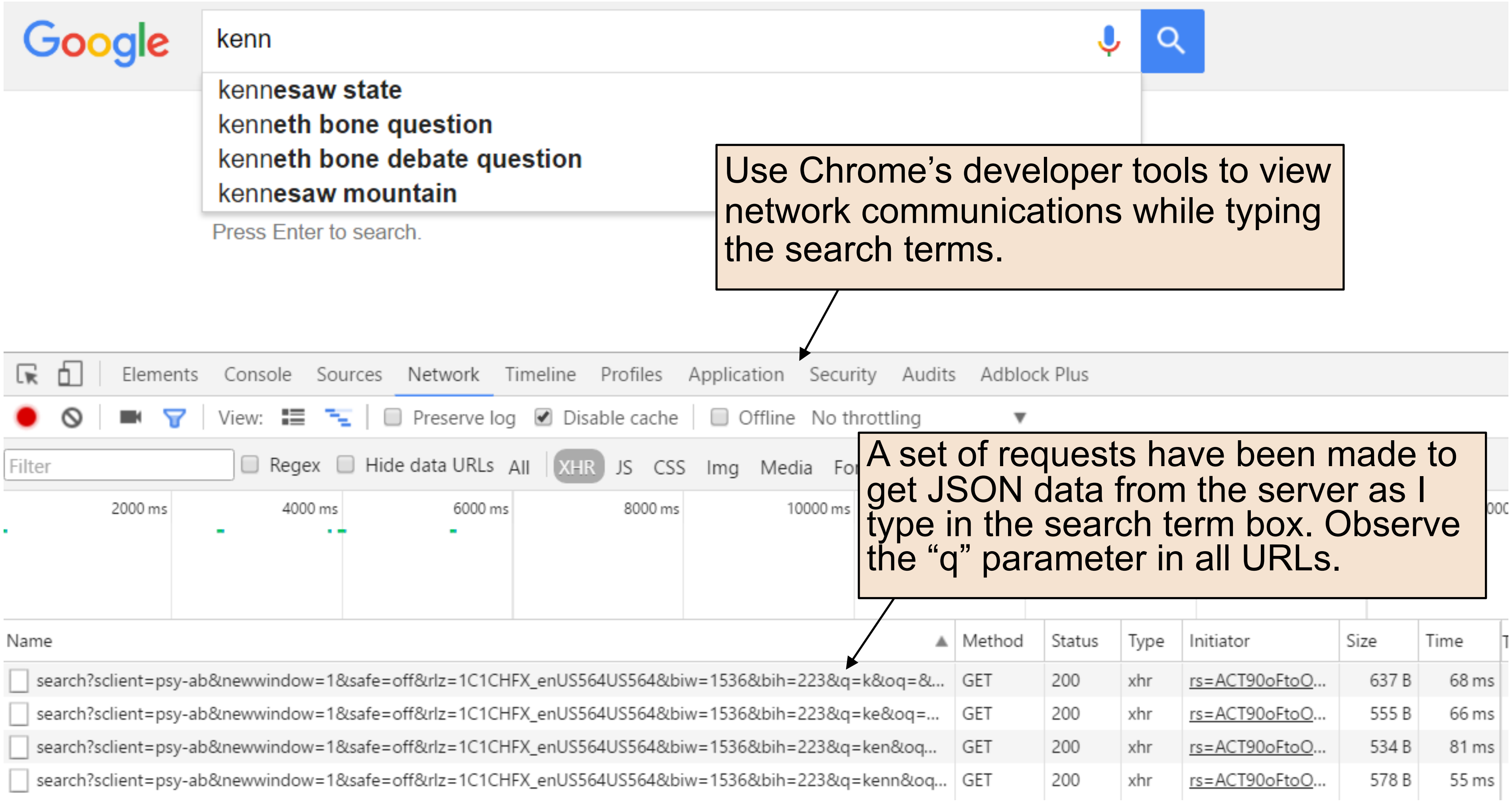
- AJAX concepts and technical elements
- AJAX implications and impacts
- jQuery AJAX
  - Basic and shorthand methods
  - Error handling

## AJAX

- AJAX (Asynchronous JavaScript and XML) is a group of interrelated web development techniques used on the client-side to create interactive web applications.
- Despite the name, the use of XML is not actually required, nor do the requests need to be asynchronous.

# First Impression

- https://www.google.com



kenn

kennesaw state
kenneth bone question
kenneth bone debate question
kennesaw mountain

Press Enter to search.

Use Chrome's developer tools to view network communications while typing the search terms.

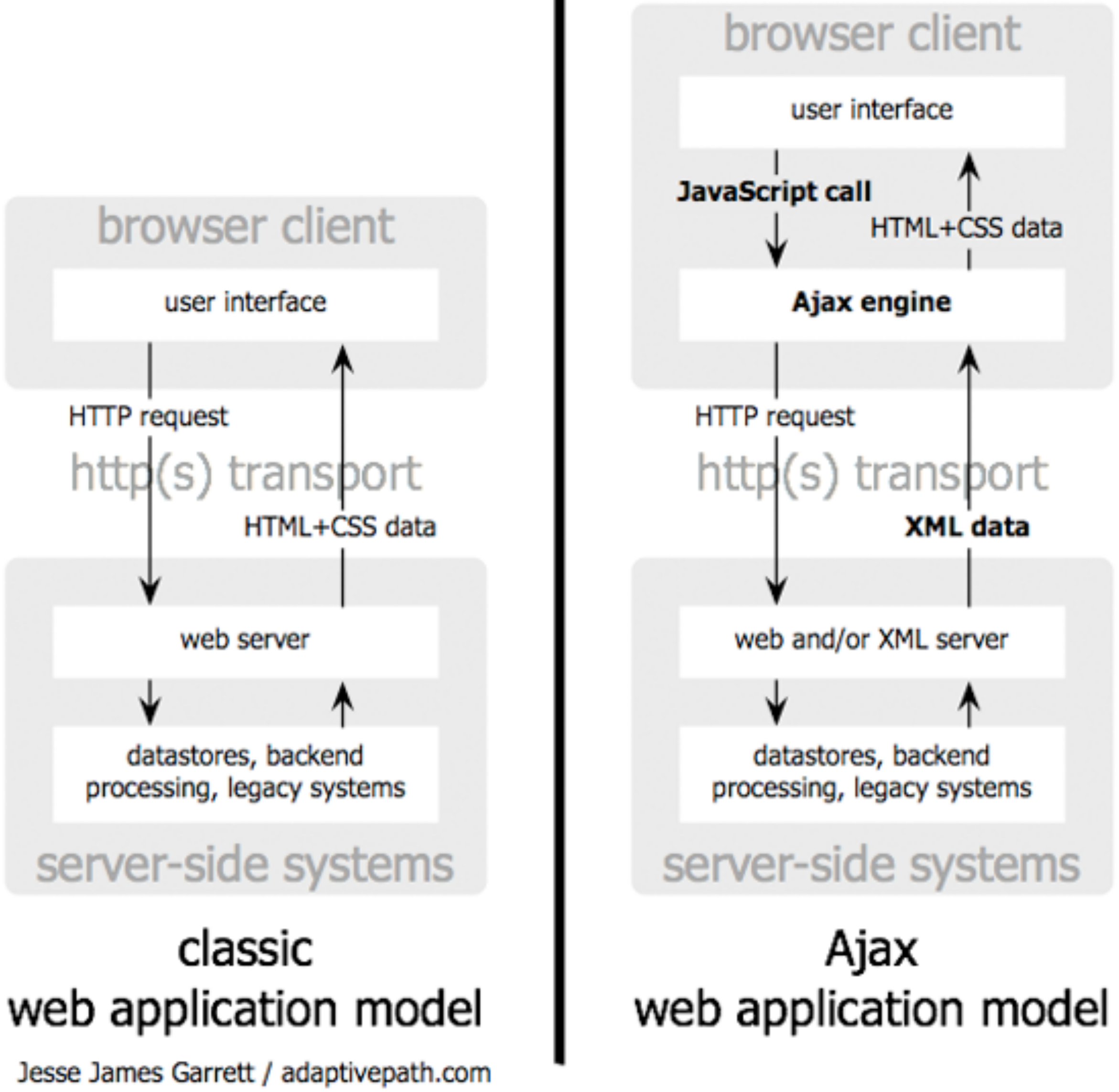
A set of requests have been made to get JSON data from the server as I type in the search term box. Observe the "q" parameter in all URLs.

| Name | Method | Status | Type | Initiator | Size | Time |
|---|---|---|---|---|---|---|
| search?sclient=psy-ab&newwindow=1&safe=off&rlz=1C1CHFX_enUS564US564&biw=1536&bih=223&q=k&oq=&... | GET | 200 | xhr | rs=ACT90oFtoO... | 637 B | 68 ms |
| search?sclient=psy-ab&newwindow=1&safe=off&rlz=1C1CHFX_enUS564US564&biw=1536&bih=223&q=ke&oq=... | GET | 200 | xhr | rs=ACT90oFtoO... | 555 B | 66 ms |
| search?sclient=psy-ab&newwindow=1&safe=off&rlz=1C1CHFX_enUS564US564&biw=1536&bih=223&q=ken&oq... | GET | 200 | xhr | rs=ACT90oFtoO... | 534 B | 81 ms |
| search?sclient=psy-ab&newwindow=1&safe=off&rlz=1C1CHFX_enUS564US564&biw=1536&bih=223&q=kenn&oq... | GET | 200 | xhr | rs=ACT90oFtoO... | 578 B | 55 ms |

# AJAX Model Difference

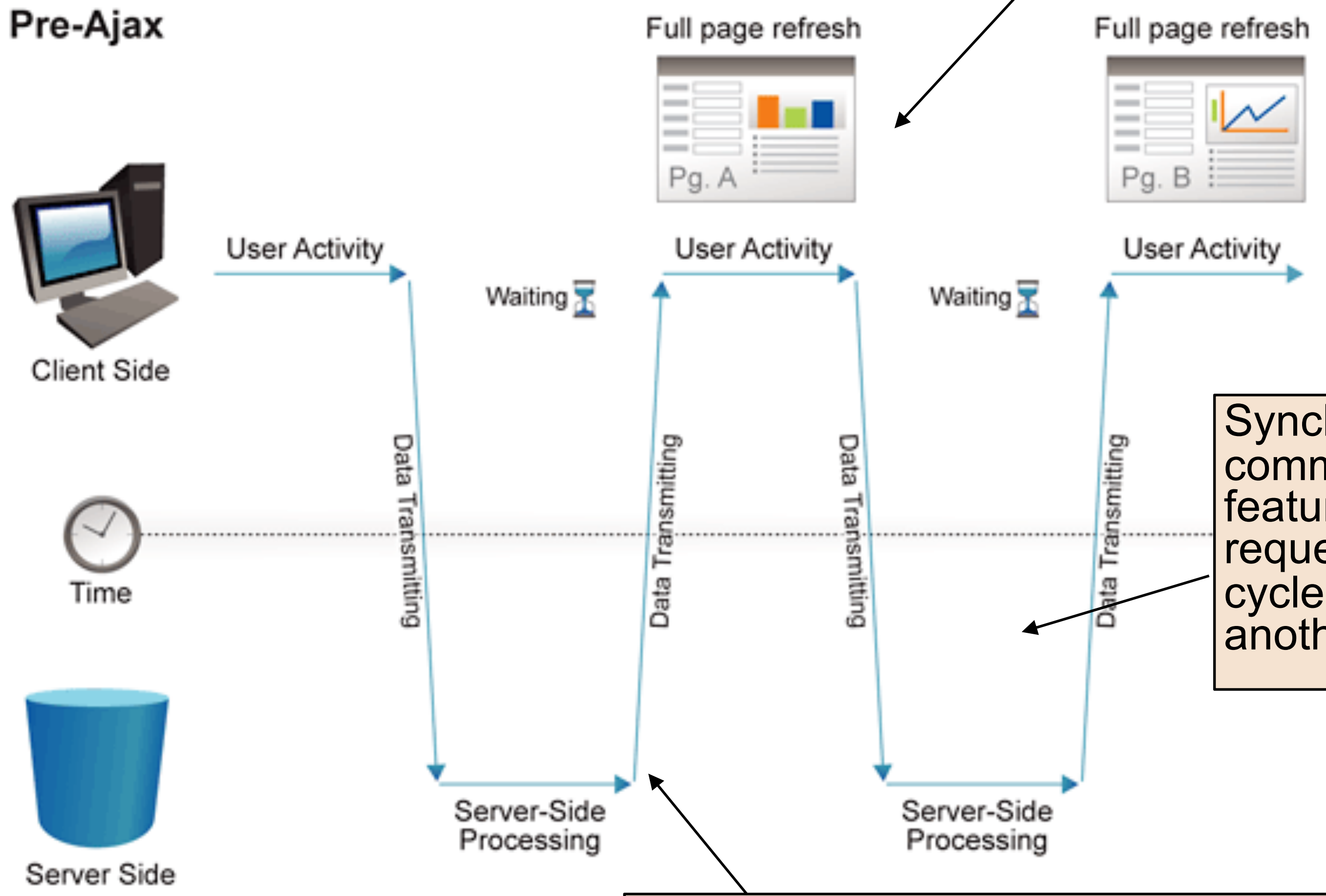
With Ajax, web applications can communicate with servers in the background without a complete page loading after every request/response cycle.

[http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications/](http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications/)



# Traditional Model

The client does not generate views/presentations (HTML/CSS).



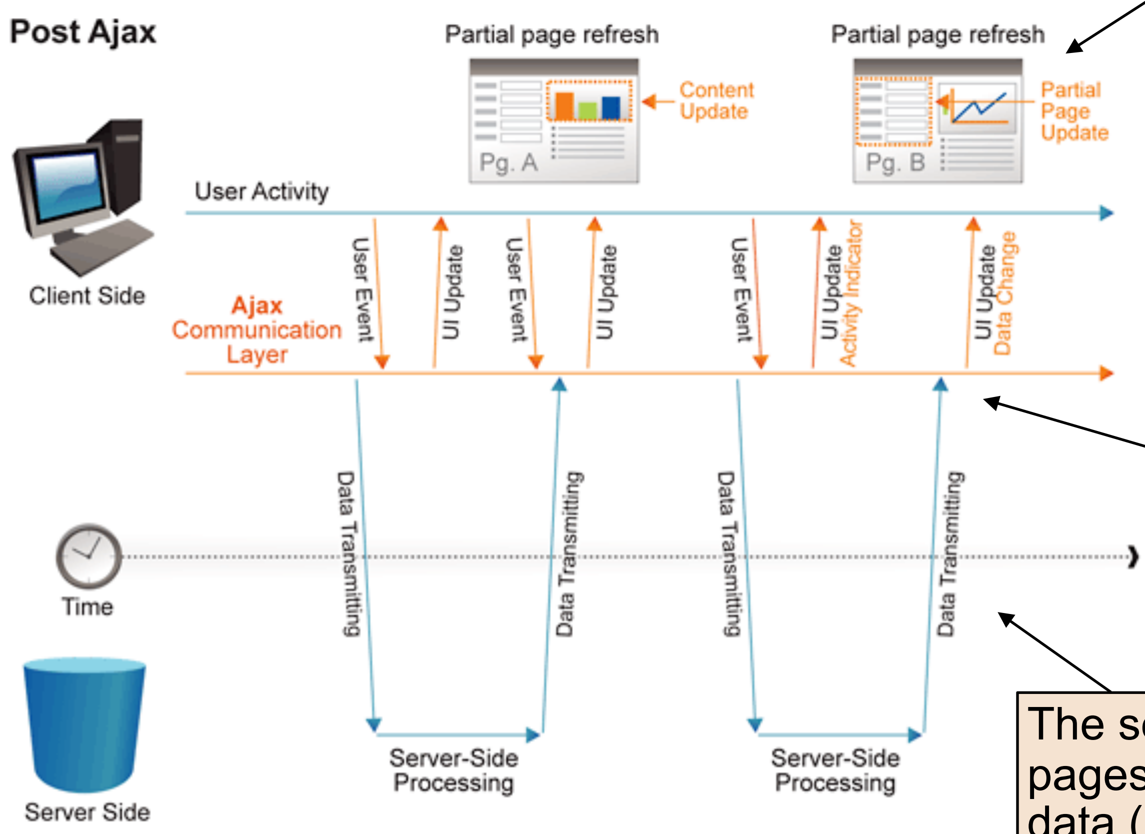
Synchronous communications feature sequential request/response cycles, one after another

The server prepares the whole page.

http://www.websiteoptimization.com/secrets/ajax/8-1-ajax-pattern.html

# Ajax Model

- With Ajax, web applications can communicate with servers in the background without a complete page loading after every request/response cycle.



The client generates views/presentations and update content (partial page) by manipulating DOM.

Asynchronous communications feature independent request/response cycles

The server prepares partial pages (partial HTML) or just data (XML or JSON).

http://www.websiteoptimization.com/secrets/ajax/8-1-ajax-pattern.html

## Major AJAX Elements

1. Standards-based presentation using HTML and CSS;
2. Asynchronous operations and communications using XMLHttpRequest;
3. Data interchange and manipulation using XML, JSON, or preformatted HTML;
4. Dynamic display and interaction through the Document Object Model (DOM);
5. JavaScript binding everything together.

## XHR (XMLHttpRequest)

- XMLHttpRequest (XHR) is an API available to web browser scripting languages such as JavaScript.
- XHR is used to send HTTP or HTTPS requests to a web server and load the server response data back into the script, in an asynchronous way.
  - See the diagram on slide 6 and 7 to compare synchronous and asynchronous communication
- It is the AJAX “engine” or the AJAX communication layer mentioned previously.
- https://en.wikipedia.org/wiki/XMLHttpRequest

## Response Format

- The response from the asynchronous call does not have to be in XML.
- Two other popular formats used
  - HTML (AHAH or AJAH): Server returns the preformatted HTML/CSS as a partial page
  - JSON (AJAJ, or Asynchronous JavaScript and JSON)
    - JSON has gain popularity for the past years to be used as an exchange format
    - https://en.wikipedia.org/wiki/AJAJ

## HTML DOM

- The HTML DOM is a standard object model and programming interface for HTML.
- Web page is partially updated through DOM rather than the whole page.
  - Update/change content
  - Add/insert new content
  - Hide/remove existing content

## JavaScript

- JS plays a central role in AJAX and ties everything together
  - Manages the AJAX communication through XHR
    - Triggers AJAX requests in page loading or user actions (clicking, mouse moving, typing, etc.)
  - Update the page dynamically through DOM
    - No complete page refresh
    - Only designated section update

## AJAX Impacts

- Two major impacts of AJAX
  1. User interface design and rich user experience (RUE)
  2. Application architecture

# AJAX's Impact on RUE

- More efficient and faster communications between server and client.
  - Enables more frequent "thin" communications that are more efficient and faster over the Internet.
  - Applications that rely on frequent updates are made possible: automatic completion, auto update, etc.

- Smoother and continuous user interaction (user experience)
  - Reduces complete page refreshes and transitions
  - Partial page loading; reducing unnecessary page refreshes, less web page transitioning and less wait time
  - Multi-path navigation/action: less navigation interruptions.
  - Improves operational efficiency, enables multi-tasking and immediate action feedback

- More like applications, building rich interaction/interface: more responsive and interactive; brings desktop experience to the web
  - Advanced interaction actions, such as drag and drop, mouse scroll, etc.
  - Desktop GUI like dynamic interfaces are possible, such as MDI (multiple div(ision) interface) which is similar to MDI (multiple document interface) or TDI (tabbed document interface) in desktop applications.
  - In-page navigation: a web application becomes less dependent on navigations between web pages. A web page becomes more like an application holder that many tasks can be completed within the page. Page transitioning is significantly decreased and navigation path is less complex.
  - Multi tasking is possible in the same page.
  - Automatic content (partial page) update

# Impact on Application Architecture

- Occasional use of AJAX will greatly improve on some UI functions
- But it had also impacted web application architectures
- Thin communications drive the redesign of client/server roles, and more separation of front end from back end
  - Web app instead of web page – a page is the app holder
  - Server provides service (not page) and resources (service oriented or resource oriented): serves data and responds to actionable request.
  - Client MVC – UI logic and part of business logic move to client.
  - Presentation processing can just stay on the client side (except for the initial loading)
  - The client has the ability to communicate directly with other servers
  - The communication focuses on service call and data transmission
- The ultimate app: single-page app + serverless
- More at http://blog.octo.com/en/new-web-application-architectures-and-impacts-for-enterprises-1/

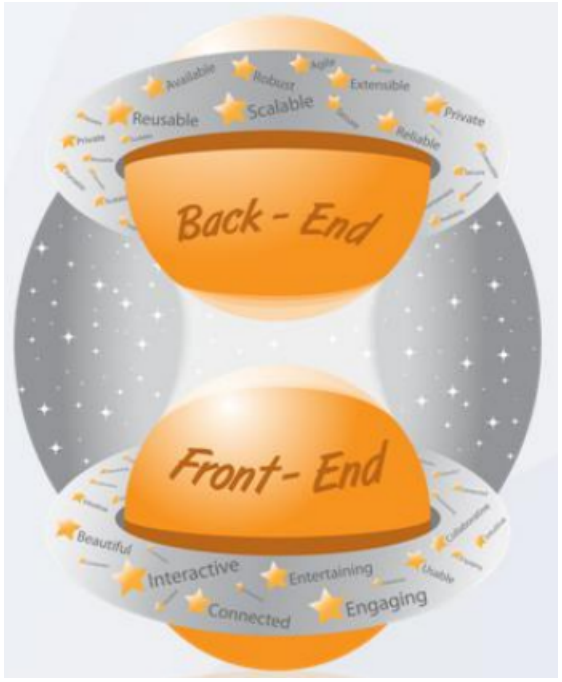
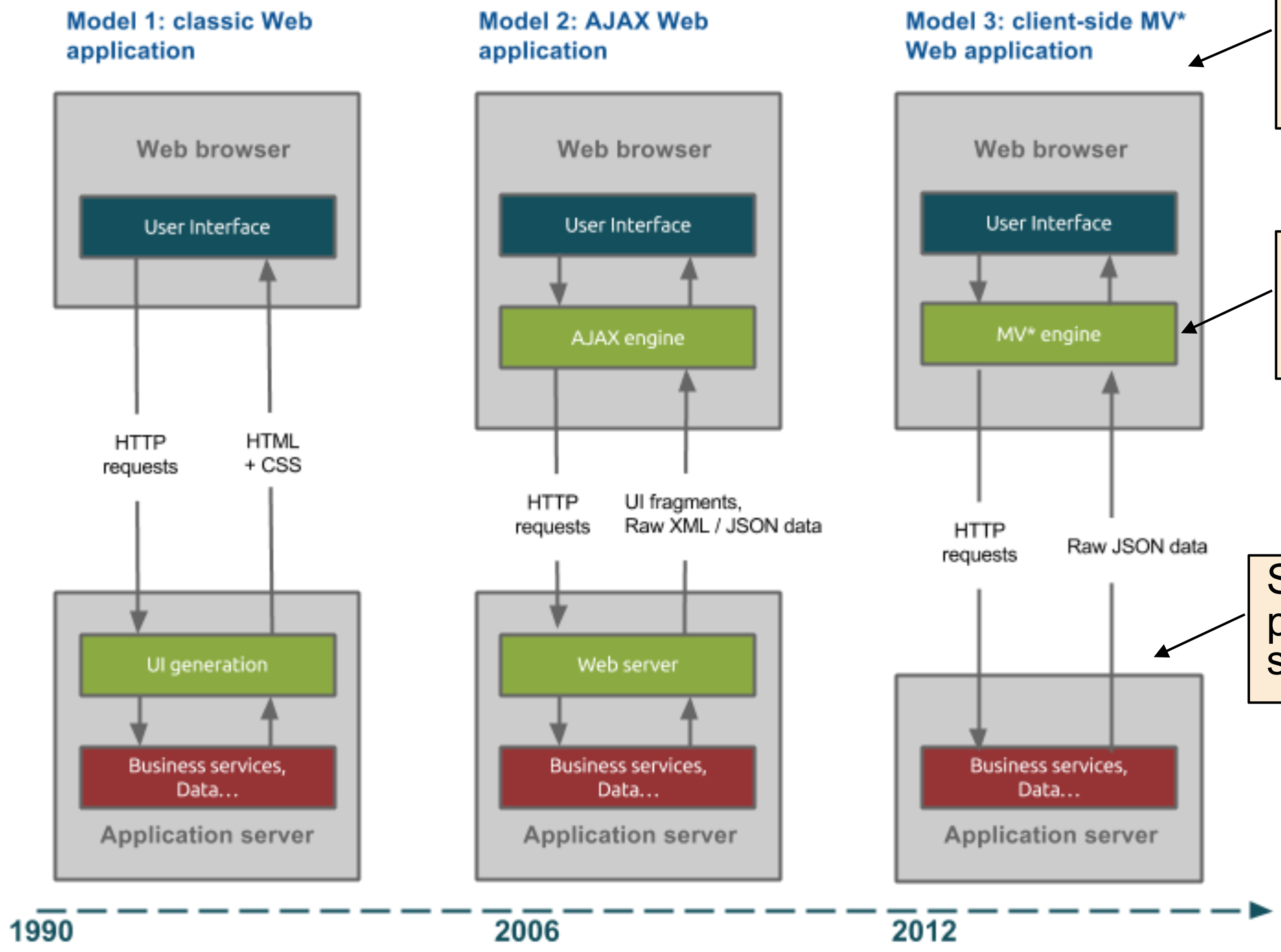


Figure from http://www.slideshare.net/Luristic/the-anatomy-of-rich-user-experience

# Changing of C/S Interaction



Can also be called the AJAX service model

UI and data logic move to the client

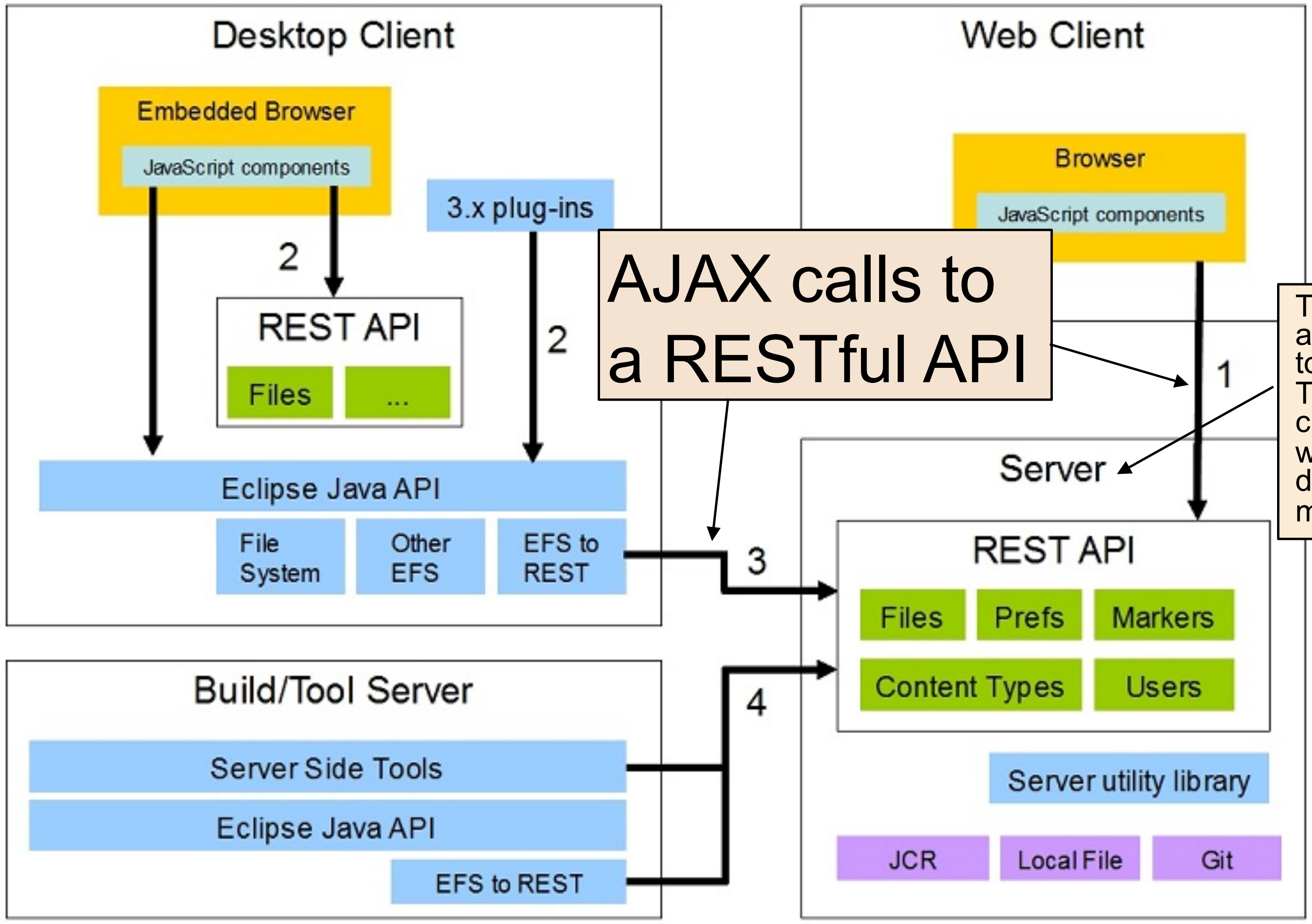
Server provides service

http://blog.octo.com/en/new-web-application-architectures-and-impacts-for-enterprises-1/

## AJAX and Services

- AJAX changes the meaning of the “client”
  - The web app is separated into the app holder (the client) and app content
  - The initial app holder is downloaded from the server and consumes services (which provide app content) from the server
- With AJAX, the presentation logic and business logic can completely move from server to client; data access logic remain on the server
  - Business logic is expose to the client; if not desired, business logic should remain on the server
- In many ways, AJAX applications follow the REST design principles. Each XMLHttpRequest can be viewed as a REST service request, sent using GET. And the response is often in JSON, a popular response format for REST.
- http://rest.elkstein.org/2008/02/ajax-and-rest.html

## AJAX + (RESTful) Services



AJAX calls to a RESTful API

The server can be any server; not tied to the initial server. There will be some cross-domain issues which will be discussed in the next module.

http://stage.vambenepe.com/archives/1801

## Typical AJAX UI and Apps

- Partial page update
  - Display data/items in a template
  - In-page navigation: multi tab/div for content: (Yahoo mail, Gmail, Google Docs, etc.)
  - Content preview: http://www.bing.com
- Application that need intensive server/client communications
  - Auto completion: http://www.google.com/, http://directory.kennesaw.edu/dir/people/, http://www.apmex.com/
  - Dynamic forms
  - Map navigation: Google maps
  - Live spelling correction
- Quick/micro user actions that should not interrupt current task
  - Quick help information, Content preview: https://zh.wikipedia.org/wiki/%E4%BA%9A%E7%89%B9%E5%85%B0%E5%A4%A7
  - Voting, server side action confirmation
  - Login
  - Auto saving: Google Docs; Quick edit, input validation, spell checking, etc.
  - Error notification
  - Form item dynamic population
  - Configurations and settings
- Push type applications
  - Auto notifications of news, message, status change, or updates
  - Auto refresh: http://www.google.com/finance
- Applications that take long processing time
  - File upload (slideshare.com),
- More
  - http://www.noupe.com/design/how-ajax-works.html
  - http://code.tutsplus.com/tutorials/20-excellent-ajax-effects-you-should-know--net-1090
  - http://www.noupe.com/development/most-wanted-ajax-techniques-50-ajax-examples-and-tutorials.html
  - http://voidcanvas.com/should-i-use-ajax-if-yes-when-where-and-why/
  - http://ajaxian.com/archives/6-places-you-must-use-ajax

# AJAX Programming Examples

- Live demo of all examples at http://it4203.azurewebsites.net/demo/ajax
- **Examples can be downloaded on the content page. Please put all files to your server and see its effects. These AJAX examples need sever side support.**

| Files | Example |
|---|---|
| 1.raw-js-ajax.html<br>server-partial-file.html | A simple example of AJAX with regular JavaScript, loading a static HTML partial page. “server-partial-file.html” is a static partial HTML page. |
| 2.jquery-ajax-loadhtml.html<br>server-data-html.php | Use three jQuery AJAX methods (load, get, ajax) to load a static or dynamic HTML partial page. “server-data-html.php” is a dynamic HTML page with PHP code. |
| 3.jquery-ajax-loadjson.html<br>server-data-json.php | jQuery AJAX methods (.getJSON, .get, .ajax) to load JSON content returned by the server. “server-data-json.php” is a dynamic JSON file generated in PHP. The example also demonstrate three ways of triggering the click event. |
| 4.jquery-ajax-post.html | Use the POST method and work with an external service http://jsonplaceholder.typicode.com |
| 5.jquery-ajax-errors.html<br>server-response.php | Demonstrates error handling; the “server-response.php” page will simulate various situations by passing a parameter in URL. |
| 6.spa-event-paramter.html<br>itbook.store-proxy.php | This example shows how to add the click event dynamically to a set of clickable elements and how to pass parameter value to the event. |

# The Simplest Classic AJAX Call

Also see http://www.w3schools.com/ajax/tryit.asp?filename=tryajax_first

```
<html><head>
<script type="text/javascript">
function UpdateDivAjax()
{
    xmlHttp = new XMLHttpRequest();
    xmlHttp.onreadystatechange = function () {
      if (xmlHttp.readyState == 4 && xmlHttp.status == 200)
        document.getElementById('ajaxDiv').innerHTML = xmlHttp.responseText;
   };
    xmlHttp.open("GET", 'ajax-partialpage.html', true);
    xmlHttp.send();
}
</script></head>
<body>
  <input type="button" value="Click me” onclick="UpdateDivAjax()" />
  <div id="ajaxDiv"></div>
</body></html>
```

1. Create a XMLHttpRequest object.

2. Define a call back function to complete tasks when the response is returned. See the next slide.

3. Send the request, with request method and target defined.

The target here is an partial HTML file which will be returned and inserted to the ajaxDiv division.

“onclick” event triggers the UpdateDivAjax function, which will update the “ajaxDiv” section.

## Call Back Function

- Call back function is an event handler to complete tasks when an asynchronous response is received. This usually defines the partial page updates.

This is an inline call back function, which is defined immediately. “function” is the default name for the inline call back function – don’t rename it.

```
xmlHttp.onreadystatechange = function ()
{
     if (xmlHttp.readyState == 4 && xmlHttp.status == 200)
      {
               document.getElementById('ajaxDiv').innerHTML=
                         xmlHttp.responseText;
      }
};
```

If the result is successfully returned and loaded.

Partial page update through DOM.

“responseText” contains raw text returned from the Ajax call target.

# AJAX Call Targets

- The AJAX call target can be set to any files or applications on the server. This can be:
  - A static file, including HTML, XML, JSON, CSV, or just pure text.
  - A dynamic web page (either HTML, XML, or JSON) generated by ASP.Net (.aspx), PHP, Servlet, etc.
  - URL parameters can be included
  - Third party Web API service point

- Examples

```
xmlHttp.open("GET", 'ajax-partialpage.html', true);
xmlHttp.open("GET", 'ajax-partialpage.aspx', true);
xmlHttp.open("GET", 'ajax-partialpage.php?bookid=k123', true);
xmlHttp.open("GET", 'ajax-partialpage.xml', true);
```

- Which server? Normally the target has to be on the same server (domain) as the requesting page.
  - The *same origin policy* prevents some AJAX techniques from being used across domains https://developer.mozilla.org/En/Same_origin_policy_for_JavaScript

## AJAX and jQuery

- Raw AJAX handling in JS is very tedious
- jQuery provides flexible and strong support to handle AJAX interactions through a set of jQuery functions.
- Main low-level function
  - $.ajax()
- Useful higher level shorthand methods
  - .load()
  - $.get()
  - $.getJSON()
  - $.getScript()
  - $.post()
- See jQuery AJAX API reference
  - http://api.jquery.com/category/ajax/

## $.ajax()

- The $.ajax() function underlies all AJAX requests sent by jQuery.
- It is often unnecessary to directly call this function,
  - Use higher-level alternatives like $.get() and .load() which are easier to use.
  - If full customization is required, then use $.ajax() for flexibility
- Usage and reference
  - http://www.w3schools.com/jquery/ajax_ajax.asp
  - http://api.jquery.com/jquery.ajax/

# Shorthand Methods

- Five main shorthand methods
  - http://api.jquery.com/category/ajax/shorthand-methods/

| | |
|---|---|
| .load() | directly load a partial page to an HTML “div” |
| $.get() | send a request using HTTP GET |
| $.getJSON() | send a request using HTTP GET; the response is expected to be in JSON format and will be parsed into a JSON object |
| $.post() | send a request using HTTP POST |
| $.getScript() | get a piece of JavaScript and insert it to the current page dynamically |

## $.getJSON()

- Load JSON-encoded data from the server using a GET HTTP request
- Basic usage

Full or partial URL that returns JSON data.

The default callback function which is executed once the response is ready.

```
$.getJSON( “http://server/file", function( json )
{
   // processing JSON using JavaScript here.

});
```

A variable representing the JSON object that is built based on the JSON response from the server, assuming the JSON parsing is correct. An error will be thrown if the parsing fails.

- Reference
  - http://www.w3schools.com/jquery/ajax_getjson.asp
  - http://api.jquery.com/jQuery.getJSON/

## jQuery AJAJ Example

```
$(document).ready(function ()
   {
      $("#button1").click(function ()
      {
         $.getJSON("instructor.json", function (instructor)
         {
                  $("#Instructor").html(instructor.FirstName +
                        " - " + instructor.Title);
         });
      });
   });
```

jQuery function to work with AJAJ

The JSON object from the response will be referred to in the function using this variable

Dynamically updates the page through DOM.

## Trigger an AJAX Request

- Three basic techniques to initiate an AJAX request
  1. Use regular buttons without HTML form (you may style it like links or other clickable areas)
  2. Use the “a” tag but with # (<a href="#">) and handle the clicks using JS/jQuery
  3. Use any element and simulate it as a link/button or a clickable area using CSS style.

- See the example: 3.jquery-ajax-loadjson.html

- Clicking should be handled by JS (click event) to stay on the same page

- **Avoid** the following ways, because they will refresh the whole page.
  - using HTML form and submit button; instead, see #1 above
  - using normal links (“a” tag with “href” pointing to a real URL); instead, see #2 method above

## Passing Parameter Value in Events

- Normally we can pass parameter via URL for multipage applications.
- In SPA, we can pass parameters to the click event through the following two ways
  1. Use custom HTML tag attribute
  2. Use text content
- Two ways to read parameter values in the event handler
  - $(this) reference http://html-tuts.com/jquery-this-selector/

```
$(".booklistitem").on('click', function ()
{
    getBookDetails( $(this).attr("data-bookid") );
});
```

  - Event target parameter https://api.jquery.com/event.target/

```
$(".booklistitem2").on('click', function (event)
    {   getBookDetails( $(event.target).text());  });
```

- See example “6.spa-event-parameter.html”

## Error Handling

- Common error situations
  - Timeout: waiting too long
  - Server error: 404 not found, 500 internal server error, etc.
  - Incorrect/invalid JSON
- Use the fail() function to handle errors.
- See the example “5.jquery-ajax-errors.html” on error handling

# More Resources

- AJAX concepts
  - What is AJAX (short video from lynda.com) https://www.youtube.com/watch?v=RDo3hBL1rfA
  - AJAX: A New Approach to Web Applications (this is the original concept by Jesse Garrett) http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications/
  - AJAX Basics Explained By Working At A Fast Food Restaurant https://blog.codeanalogies.com/2018/01/15/ajax-basics-explained-by-working-at-a-fast-food-restaurant/
  - https://en.wikipedia.org/wiki/Ajax_(programming)

- jQuery AJAX
  - http://www.tutorialspoint.com/jquery/jquery-ajax.htm or http://www.w3schools.com/jquery/jquery_ajax_intro.asp
  - Video tutorial:
    - Part 1 https://www.youtube.com/watch?v=fEYx8dQr_cQ
    - Part 2 https://www.youtube.com/watch?v=5nL7X1UMWsc
  - Major references:
    - http://api.jquery.com/category/ajax/shorthand-methods/
    - http://api.jquery.com/category/ajax/
    - http://www.w3schools.com/jquery/jquery_ref_ajax.asp

- Others
  - https://www.tutorialspoint.com/ajax/ajax_quick_guide.htm
  - http://blog.octo.com/en/new-web-application-architectures-and-impacts-for-enterprises-1/
  - Ajax and PHP http://www.w3schools.com/PHP/php_ajax_database.asp
  - AJAX+REST as the latest architectural mirage: http://stage.vambenepe.com/archives/1801
  - https://www.safaribooksonline.com/library/view/ajax-design-patterns/0596101805/ch01.html
  - http://www.alexhopmann.com/story-of-xmlhttp/
  - http://www.openajax.org