



AJAX@localhost

PHP Conference UK 2006
10th Feb 2006, London, UK
Harry Fuecks



Where I've come from...

- JPSpan: <http://sourceforge.net/projects/jpspan>
 - Started before “AJAX”
 - XML-RPC / SOAP inspired:
 - RPC over HTTP
 - Mistakes and lessons
- AJAX: the Emperor isn't wearing much...



AJAX@localhost

about:ajax



What is AJAX?

- Acronym: Asynchronous Javascript + XML
- HTTP requests scripted with Javascript
- Working around web page reloads



AJAX Criticisms

- Things you hear...
 - “It's a stupid name”
 - “It's nothing new”
 - MS added XMLHttpRequest ActiveX object to IE ~ 1999
 - JSRS (~2000) <http://www.ashleyit.com/rs/main.htm>
 - “Think about the poor users!”
 - Usability / accessibility issues



AJAX Criticisms

- Things you hear less frequently...
 - “It's the latency stupid!”
 - “A great way to miss deadlines”
 - “Web designers specialize in HTML and CSS... not HTTP”



Nagging doubt of the day

- So we all agree Google Suggest rocks...
 - <http://www.google.com/webhp?complete=1&hl=en>



Nagging doubt of the day

- So we all agree Google Suggest rocks...
 - <http://www.google.com/webhp?complete=1&hl=en>
- Who's *actually* using Google Suggest in reality?

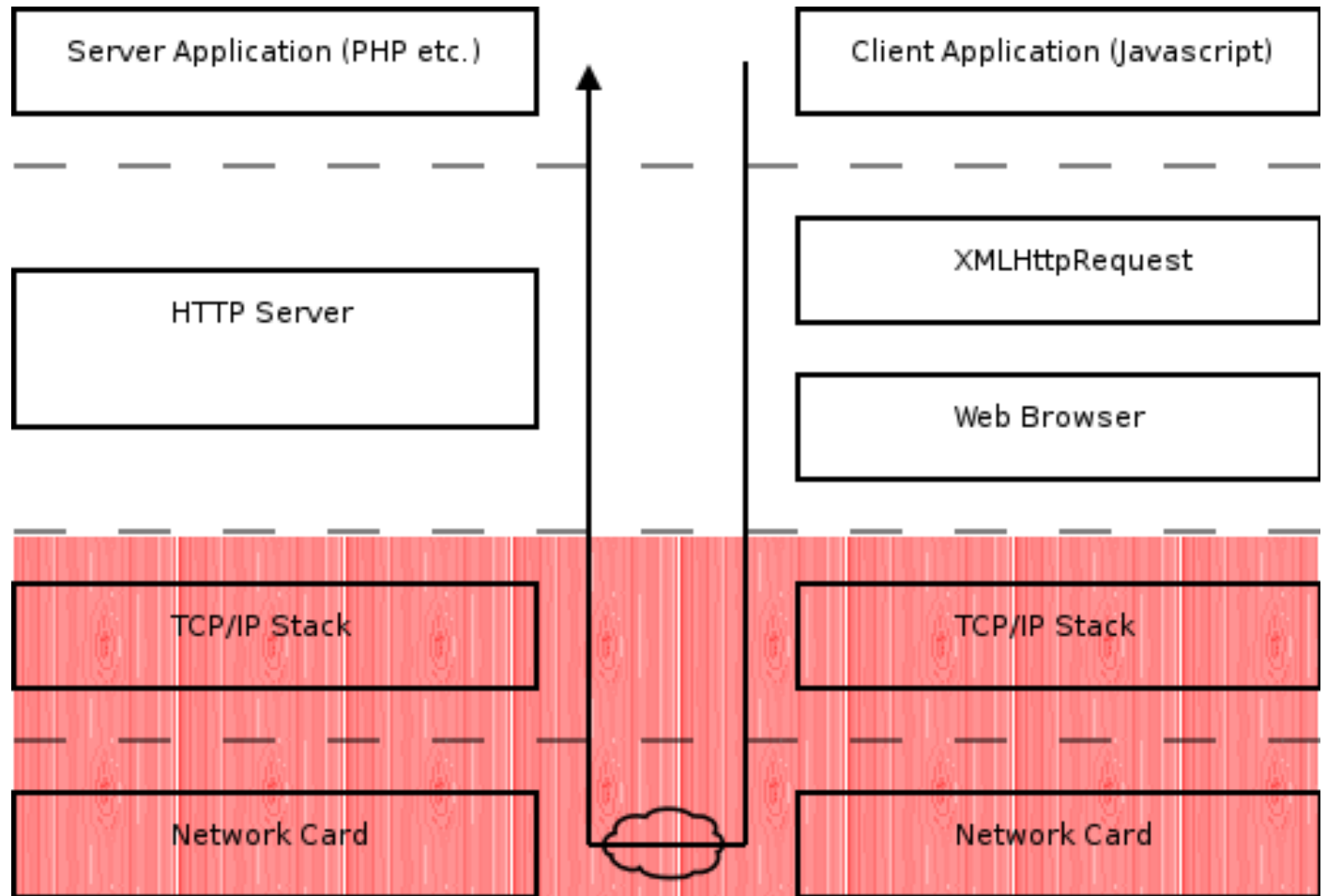


AJAX@localhost

bottom of the stack



AJAX “Stack”





Consumer Guide to Networks

- Two key dimensions:
 - Bandwidth
 - Volume of data at a single instance in time
 - Latency
 - Time delay from A to D (via B and C)
- Service providers don't sell latency
...and can only sell bandwidth they control
- “It's the latency stupid”

<http://rescomp.stanford.edu/~cheshire/rants/Latency.html>



Network Latency

- Latency not a constant - **varies over time!**

```
$ tcptraceroute www.php.net
Selected device wlan0, address 192.168.0.166, port 32776 for outgoing packets
Tracing the path to www.php.net (64.246.30.37) on TCP port 80 (www), 30 hops max
 1 192.168.0.1  9.650 ms  10.175 ms  1.950 ms
 2 10.190.96.1  7.959 ms  26.580 ms  10.146 ms
 3 * tengig-11-0.blxOTF001.gw.cablecom.net  9.314 ms  9.097 ms
 4 tengig-11-0.blxOTF001.gw.cablecom.net  8.434 ms  9.484 ms  9.629 ms
 5 pos-0-0.blxZH002.bb.cablecom.net  17.704 ms  10.388 ms  9.783 ms
 6 fast0-0.iix-igr01.nycl.twtelecom.net  99.681 ms  100.293 ms  99.039 ms
 7 core-01-so-1-0-0-0.nycl.twtelecom.net  100.207 ms  99.269 ms  98.446 ms
 8 dist-01-so-1-0-0-0.hsto.twtelecom.net  153.200 ms  145.780 ms  169.293 ms
 9 hagg-02-ge-2-3-0-506.hsto.twtelecom.net  144.844 ms  143.550 ms  143.198 ms
10 216-54-253-2.gen.twtelecom.net  177.272 ms  158.129 ms  159.765 ms
11 ivhou-207-218-245-29.ev1.net  152.826 ms  152.050 ms  152.501 ms
12 ivhou-207-218-223-119.ev1.net  151.677 ms  153.216 ms  153.481 ms
13 rsl.php.net  162.985 ms  154.419 ms  157.232 ms
```

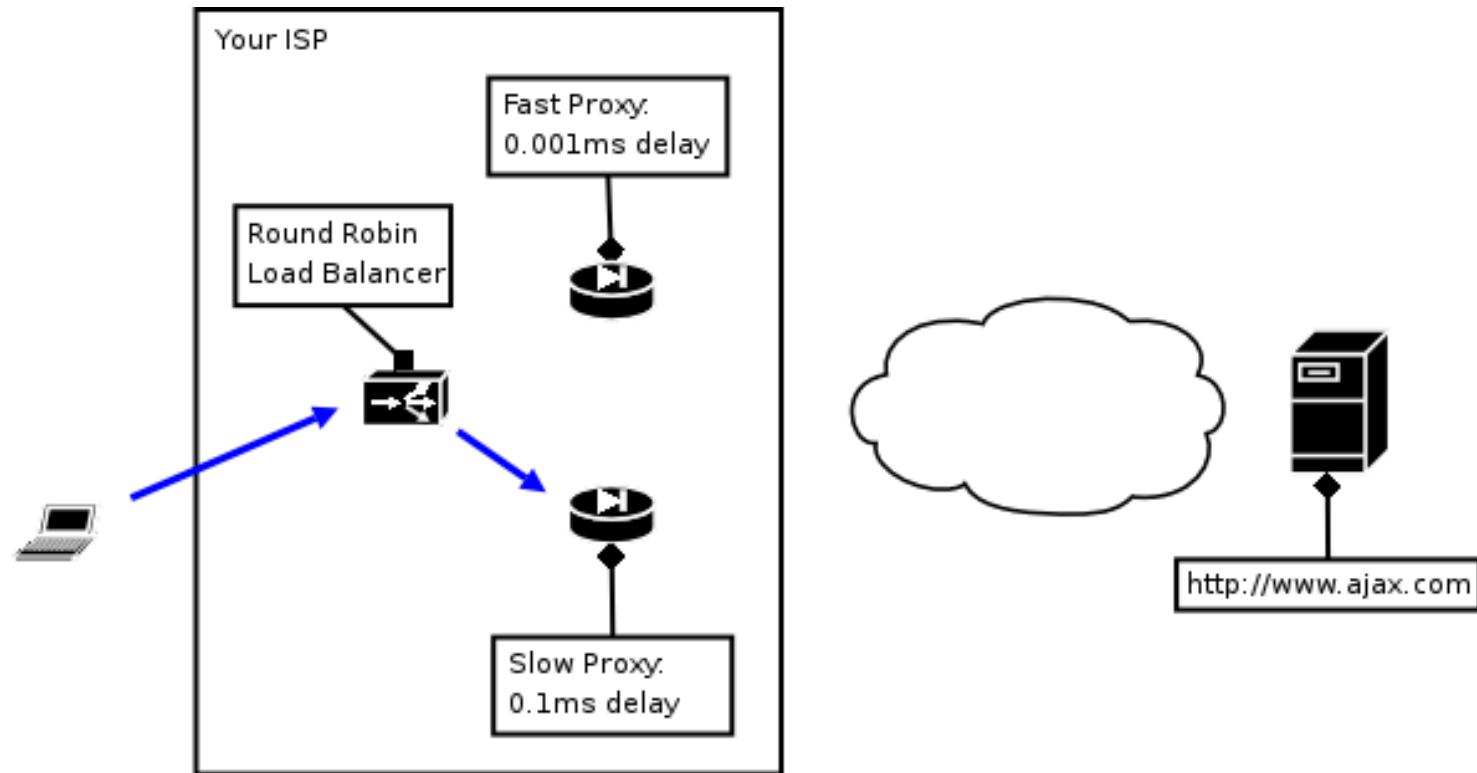


Server Latency

- Processor load, available child processes, disk I/O etc.
- Different “operations” in your code may be relatively faster or slow
- Typically bigger problem than network



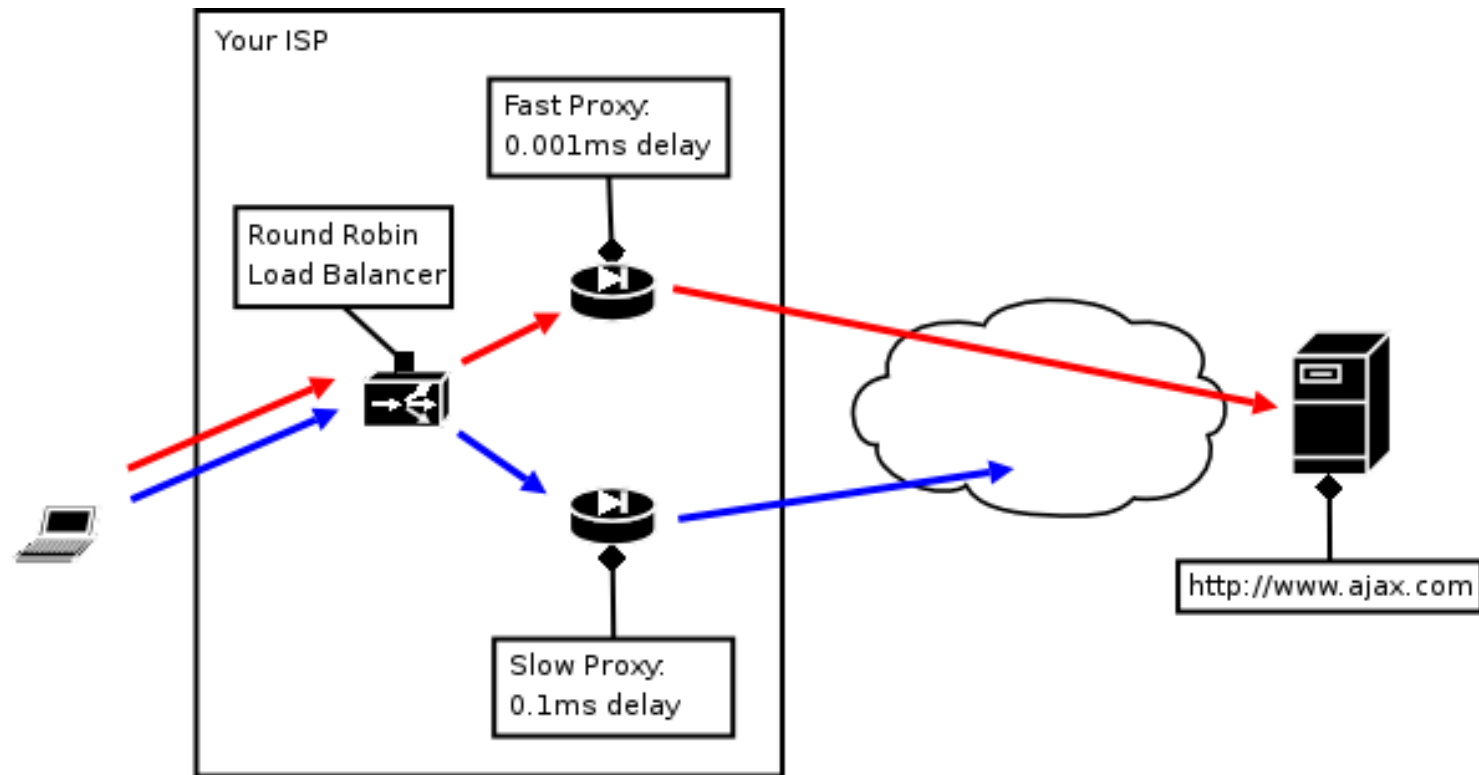
Just one of many delay scenarios...



First HTTP Request Issued
- routed through slow proxy



Just one of many delay scenarios...



Second Request Issued 0.01ms later

- routed through fast proxy
- reaches web server *before* first



AJAX Proxy

Download: <http://sourceforge.net/projects/jpspan>

- HTTP proxy for [AJAX@localhost](#)
 - Debugging - see HTTP request / response
 - Simulate network latency / failure
 - Simple mini language for scripting problems
- Hacked together in python
 - Some day: re-write using twisted
- Alternatives
 - Fiddler Proxy <https://www.fiddlertool.com/fiddler/> - Win32, scriptable
 - XMLHttpRequest Add Latency <http://blog.monstuff.com/archives/000264.html>
 - Greasemonkey – constant latency only – not a proxy



script.aculo.us reverse autocomplete

<http://demo.script.aculo.us/ajax/autocomplete/>

Demo:<http://www.phppatterns.com/stuff/latency.html>

```
# AJAX Proxy instructions
#Request      Response
wait:10       wait:0
wait:8        wait:0
wait:6        wait:0
wait:4        wait:0
wait:2        wait:0
wait:0        wait:0
```

Response to first request arrives last

Interesting security implication:

control over latency (e.g. DOS) == control over client

Same mistake: <http://jpspan.sourceforge.net/examples/autocomplete.php>



The Starbucks Factor

- “Starbucks Does Not Use Two-Phase Commit”

http://www.eaipatterns.com/ramblings/18_starbucks.html

“When you place your order the cashier marks a coffee cup with your order and places it into the queue.

[...]

Drinks are delivered out of sequence and need to be matched up to the correct customer.”



More on latency...

- “AJAX Latency problems: myth or reality?”
<http://richui.blogspot.com/2005/09/ajax-latency-problems-myth-or-reality.html>
- “AJAX: reducing latency with a CDN”
<http://richui.blogspot.com/2005/09/ajax-reducing-latency-with-cdn.html>
 - but remember **latency varies over time!**
- “Rapid Ajax requests out of order”
<http://wrath.rubyonrails.org/pipermail/rails-spinoffs/2005-November/001334.html>
 - “I'm trying to write an ajax chat. I'm noticing that when ajax requests are fired off in rapid succession, they often will arrive out of order to the server.”



Other minor details...

- Mobile phone service providers love AJAX!
- If connection dropped, XMLHttpRequest acts like nothing happened...
 - no exceptions
 - no nothing
- Connection timeout not included
 - roll your own
- HTTP pub / sub == “The Joy of Polling”

<http://ajaxblog.com/archives/2005/06/01/async-requests-over-an-unreliable-network>

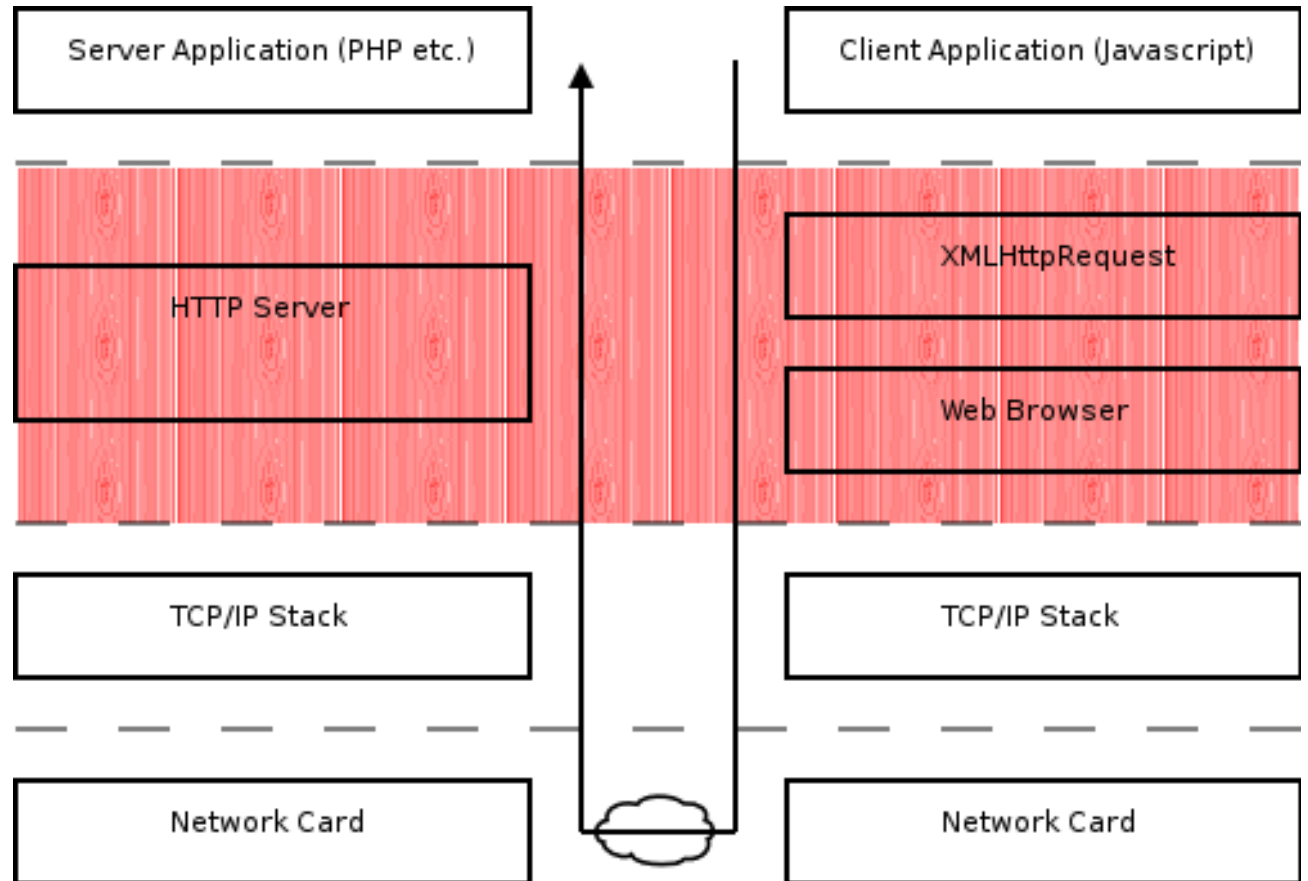


AJAX@localhost

around the middle



AJAX “Stack”





Choices

- XMLHttpRequest vs. “Remote Scripting”
- Wire formats
 - JSON (or just eval()able Javascript)
 - Micro-content
 - Plain text
 - Some flavour of XML (unpopular)
 - etc.



XMLHttpRequest Limitations

- Browser implementations vary
- Browser has final say
 - no cross domain requests (security)
 - limited control over caching (security)
 - concurrent connection restrictions (RFC2616)
<http://kb.mozillazine.org/Network.http.max-connections>
<http://blogs.msdn.com/ie/archive/2005/04/11/407189.aspx>
- Doesn't offer enough
 - e.g. no timeouts – roll your own
<http://ajaxblog.com/archives/2005/06/01/async-requests-over-an-unreliable-network>
- Varying degrees of HTTP support
 - XMLHttpRequest Tests: <http://www.mnot.net/javascript/xmlhttprequest/>



AJAX in 30 Seconds

The fun starts with a synchronous request...

```
function createRequestObject() {
    var ro;
    var browser = navigator.appName;
    if(browser == "Microsoft Internet Explorer"){
        ro = new ActiveXObject("Microsoft.XMLHTTP");
    }else{
        ro = new XMLHttpRequest();
    }
    return ro;
}

var http = createRequestObject();

function sndReq(action) {
    http.open('get', 'rpc.php?action='+action);
    http.onreadystatechange = handleResponse;
    http.send(null);
}

function handleResponse() {
    if(http.readyState == 4){
        var response = http.responseText;
        var update = new Array();

        if(response.indexOf('|' != -1)) {
            update = response.split('|');
            document.getElementById(update[0]).innerHTML = update[1];
        }
    }
}
```

Thanks Rasmus: <http://marc.theaimsgroup.com/?l=php-general&m=112198633625636&w=2>



Synchronous Requests == BAD

- <http://ajaxblog.com/archives/2005/05/25/synchronous-requests-bad>
 - Demo: <http://www.phppatterns.com/stuff/sync.html>
- “Making callbacks synchronous, or how to shoot yourself in the foot”
<http://weblogs.asp.net/bleroy/archive/2005/12/15/433278.aspx>



AJAX in 30 Seconds

...and what about the RPC happening here?

```
function createRequestObject() {
    var ro;
    var browser = navigator.appName;
    if(browser == "Microsoft Internet Explorer"){
        ro = new ActiveXObject("Microsoft.XMLHTTP");
    }else{
        ro = new XMLHttpRequest();
    }
    return ro;
}

var http = createRequestObject();

function sndReq(action) {
    http.open('get', 'rpc.php?action='+action);
    http.onreadystatechange = handleResponse;
    http.send(null);
}

function handleResponse() {
    if(http.readyState == 4){
        var response = http.responseText;
        var update = new Array();

        if(response.indexOf('|' != -1)) {
            update = response.split('|');
            document.getElementById(update[0]).innerHTML = update[1];
        }
    }
}
```

Thanks Rasmus: <http://marc.theaimsgroup.com/?l=php-general&m=112198633625636&w=2>



The RPC Problem

- “First Law of Distributed Object Design: Dont distribute your objects!”
- Martin Fowler
- Scaling issues
- Performance issues
- Error handling
- Debugging joys



HTTP: the forgotten API

- HTTP defines an optimal RPC API
 - Nouns first please!
 - e.g. <http://www.php.net/manual/en/>
 - Verbs second...
 - GET, POST, DELETE etc.
- REST: the HTTP users manual
 - Start here: http://en.wikipedia.org/wiki/Representational_State_Transfer
 - For the braver: http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm



The Joy of REST

<http://greg.chiaraquartet.net/archives/49-PEAR-1.4.0,-meet-REST-1.0.html>

“Recently, PEAR has become a victim of its own success, experiencing tremendous load problems at pear.php.net. [...] the load introduced by dynamic processing of XML-RPC requests is substantial and significant.”

“REST is *Very* fast [...] marvel at how fast it is, especially with http caching :)”



HTTP don'ts

- Don't tunnel through HTTP GET
 - **ABUSE:** <http://ajax.com/widgets/1/update>
 - HTTP abuse
http://www.infoworld.com/article/05/04/20/17OPstrategic_1.html
 - **SILLY:** <http://ajax.com/getWidgetById?id=1>
- Try not to tunnel through POST
 - You can't cache POST
 - missed your chance with `getWidget(1)`
- Don't reinvent HTTP!
 - DIY RPC will pollute your entire design



More stuff to think about

- Impact of AJAX on server load?
 - Today's HTTP servers ready for AJAX?
- Client issues?
 - “Internet Explorer and Connection Limits”

<http://blogs.msdn.com/ie/archive/2005/04/11/407189.aspx>

“IE strictly follows the standards-- in this case, RFC2616, which covers HTTP1.1. As noted in the RFC:

Clients that use persistent connections SHOULD limit the number of simultaneous connections that they maintain to a given server. A single-user client SHOULD NOT maintain more than 2 connections with any server or proxy.”

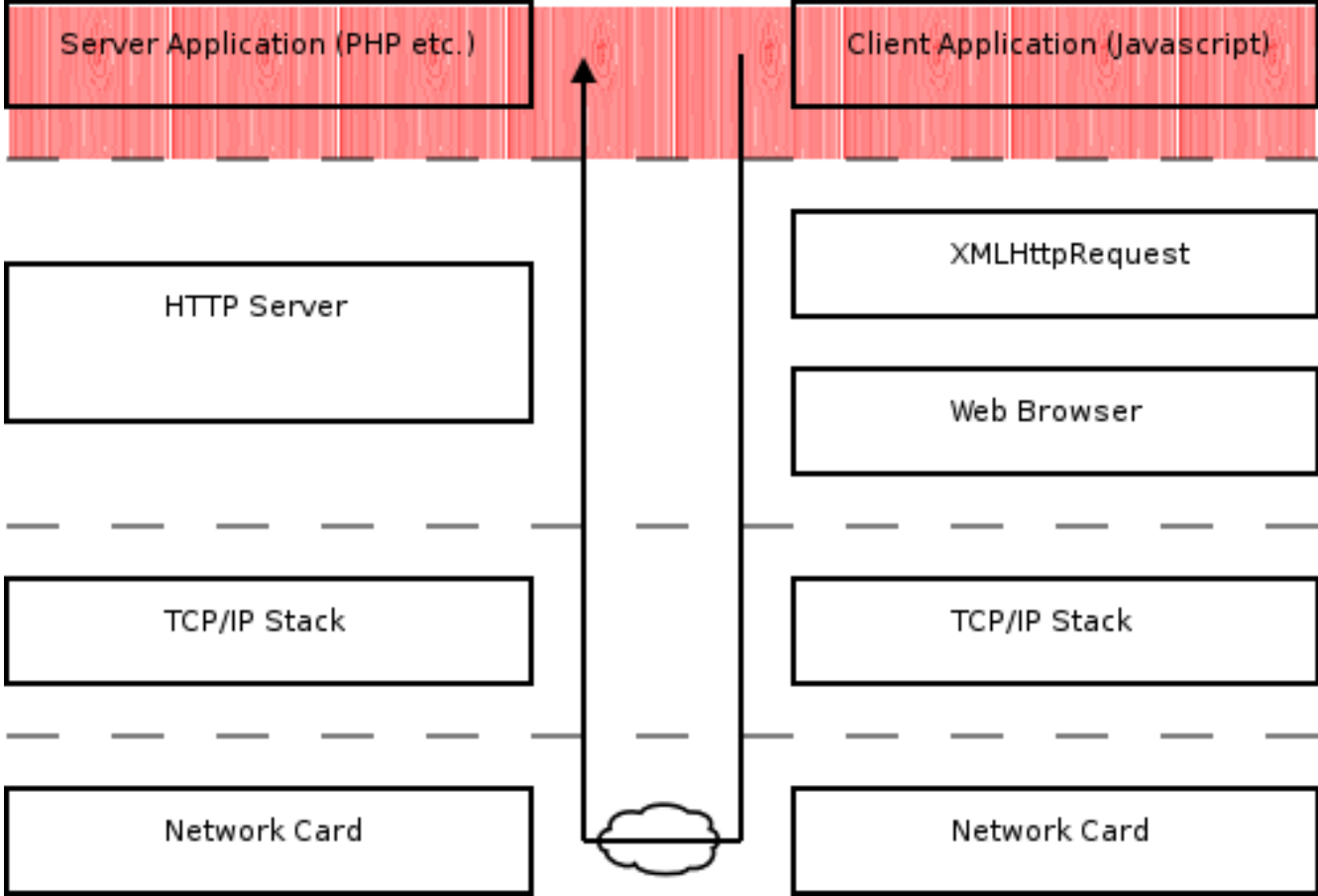


AJAX@localhost

top of the stack



AJAX Stack





Error Handling not included



The connection has timed out

The server at ch2.php.net is taking too long to respond.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

Try Again



Marcus says it best

- “Listen kids, AJAX is not cool”

<http://www.lastcraft.com/blog/index.php?p=19>

- “make sure it responds in 1/10th of a second”
- “When you write AJAX applications you drive a horse and cart through one of the most successful metaphors of all time.”
- “Perhaps the message is don't use a non-transactional medium for entering critical data.”

<http://www.sitepoint.com/forums/showpost.php?p=2198661&postcount=7>



Bottom Line...

- If you're coupling `onClick` to an HTTP request
 - think again
 - think some more



Bottom Line...

- If you're coupling `onclick` to an HTTP request
 - think again
 - think some more
- ...and really think before submitting forms with AJAX



Some angst with that doubt, Sir?

- Security issues?
 - Javascript injection? Cross domain requests **VERY** bad idea
 - “Exploiting the XMLHttpRequest object in IE”
http://www.modsecurity.org/archive/amit/exploiting_the_xmlhttprequest_object_in_ie.txt
- Client / Server character encoding mismatch?



Some angst with that doubt, Sir?

- What's a session with AJAX?

<http://ajaxblog.com/archives/2005/05/27/ajax-whats-a-session>

- Javascript as a runtime

- memory “leaks”: simplistic garbage collection (excepting Spidermonkey)

- it just sucks:

<http://bob.pythonmac.org/archives/2005/06/02/javascript-sucks-volume-1/>

- “Thin to My Chagrin”

<http://blogs.msdn.com/ericlippert/archive/2003/11/18/53388.aspx>

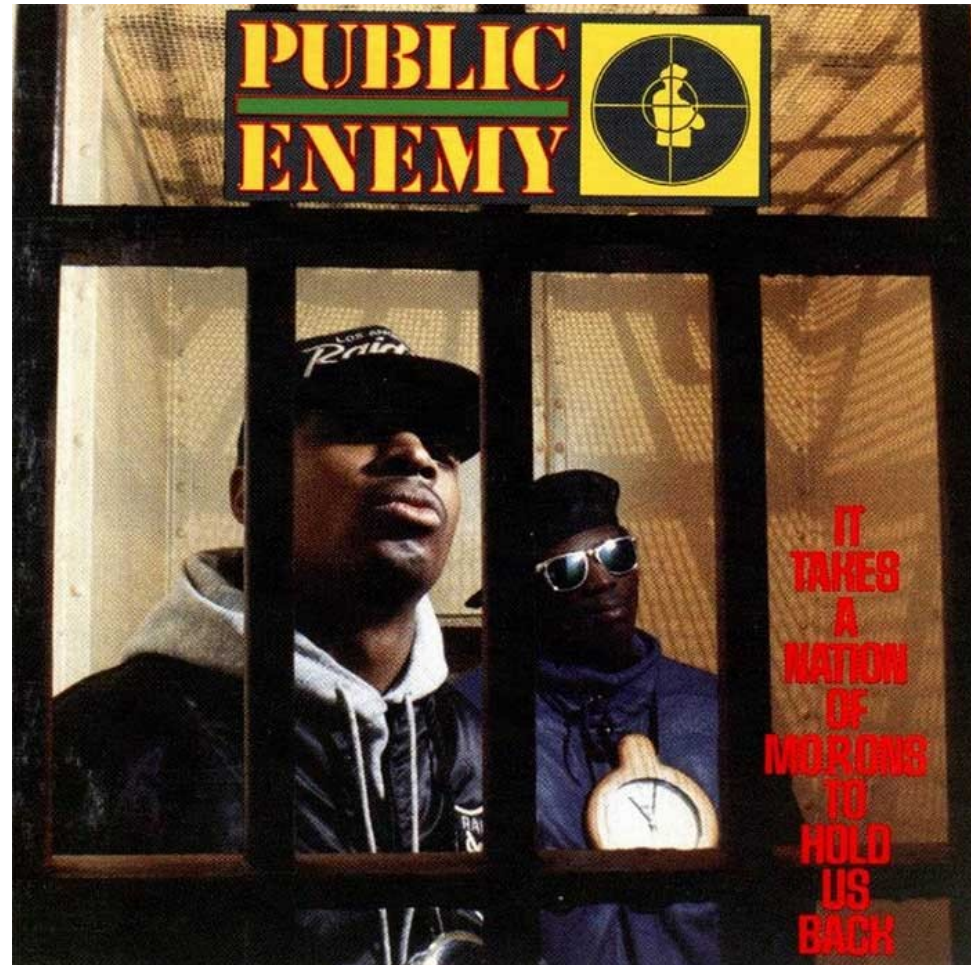
“JScript was designed for simple scripts on simple web pages, not large-scale software”



AJAX@localhost

buyers guide to ajax

Don't believe the hype





The 5 second AJAX test

- Pull out your network cable
 - Warnings?
 - Data loss?
 - Recovery?



The 5 minute AJAX test

- Simulate with AJAX proxy (or similar)
 - Insert delays
 - Insert varying delays
 - Drop connections
 - Still smiling?



AJAX@localhost

state of the art



AJAX Today

- Current frameworks and toolkits...
 - making the common problems easier
 - but ignoring the hard problems
- Technique may work better than framework
 - <http://particletree.com/features/the-hows-and-whys-of-degradable-ajax/>
- #AJAX frameworks > #PHP frameworks?
 - <http://blog.joshuaeichorn.com/ajax-resources/>
 - <http://wiki.osafoundation.org/bin/view/Projects/AjaxLibraries>
 - <http://edevil.wordpress.com/2005/11/14/javascript-libraries-roundup/>



AJAX Tomorrow

- Open AJAX? (IBM, Oracle, Google etc.)

<http://www.internetnews.com/dev-news/article.php/3582156>



AJAX Tomorrow

- Tibet (?)

<http://www.technicalpursuit.com/>

- Pre-dates “AJAX” by many years
- Work flows to help with asynchrony
- Be patient (unusual release strategy)...

http://www.technicalpursuit.com/ajax_indepth.htm

“[...] we've yet to see a single AJAX toolkit that offers you any kind of multiple-request coordination

-- so you'll end up writing the synchronization logic yourself.”



AJAX@localhost

