Extreme Web Caching

Jason Hunter

Principal Technologist
Mark Logic
http://marklogic.com

TS-4251

Real Life Web Caching

Make your site faster, for less money

How to serve more content faster by serving less content

Professional-Grade Caching Strategies

- How HTTP Works
- Using Caches and Proxies
- Keeping Personalized Content Private
- Serving Images and Static Content
- Managing CSS and Javascripttm Technology Changes
- Tracking Hit Statistics
- Based On “HTTP Caching and Cache-Busting for Content Publishers” By Michael Radwin, Yahoo!

Why Caching?

- Publishers have a lot of web content
  - HTML, images, movies, Flash
- Caching...
  - Makes your site faster and more responsive
  - Reduces your bandwidth bill
  - Smart caching still enables personalization...
  - Timely data (stock quotes, news stories)
  - Sensitive info (email, online shopping)

Agenda

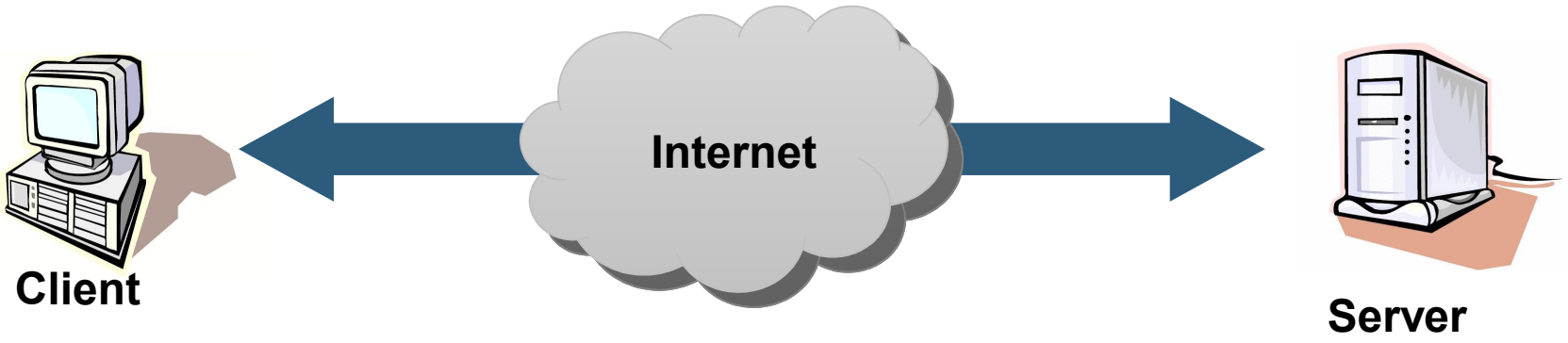
How HTTP Works

Top Five Techniques for Web Publishers

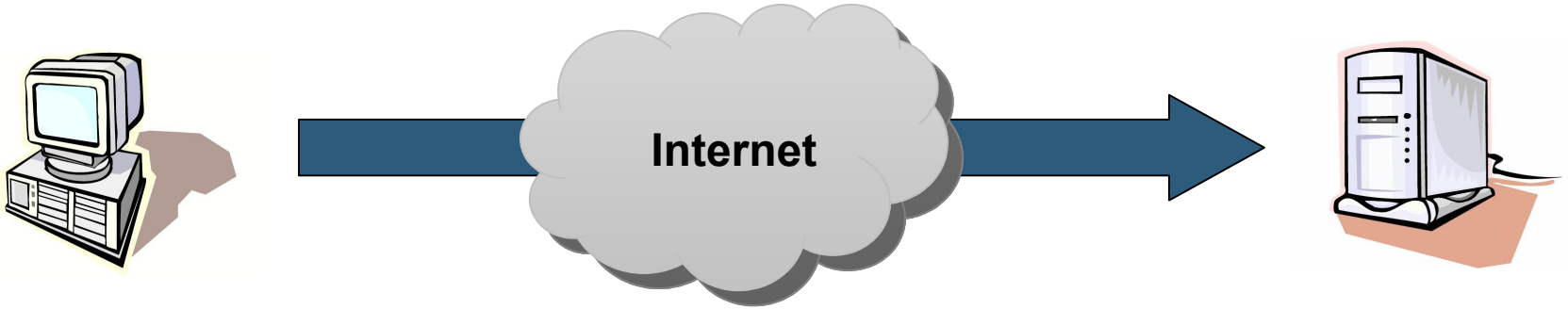
Review

HTTP: Simple and Elegant

1. Client connects to www.server.com port 80

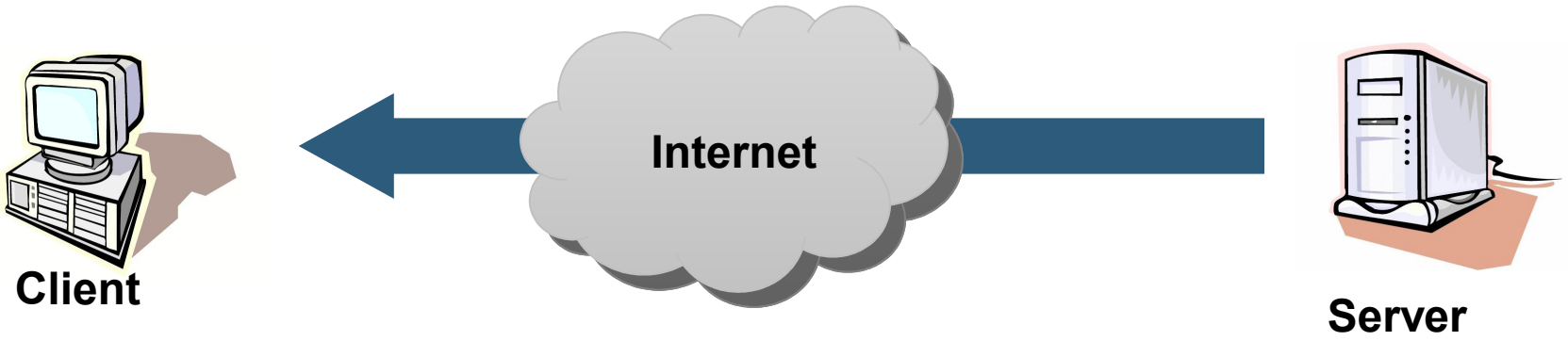


5. Client sends GET request

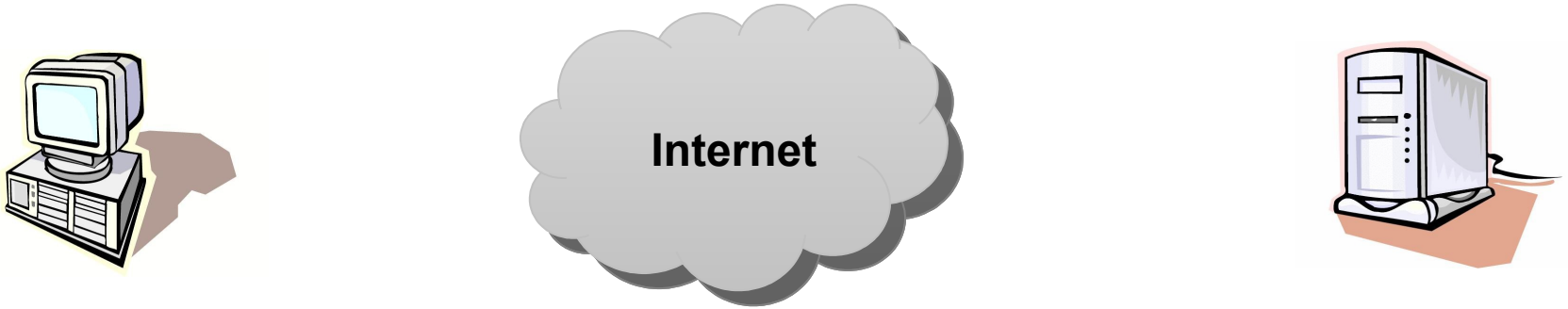


HTTP: Simple and Elegant

1. Server sends response



5. Client closes connection



HTTP Back and Forth

--Client--

```
$ telnet www.server.com 80
GET /index.html HTTP/1.1
Host: www.server.com
```

--Server--

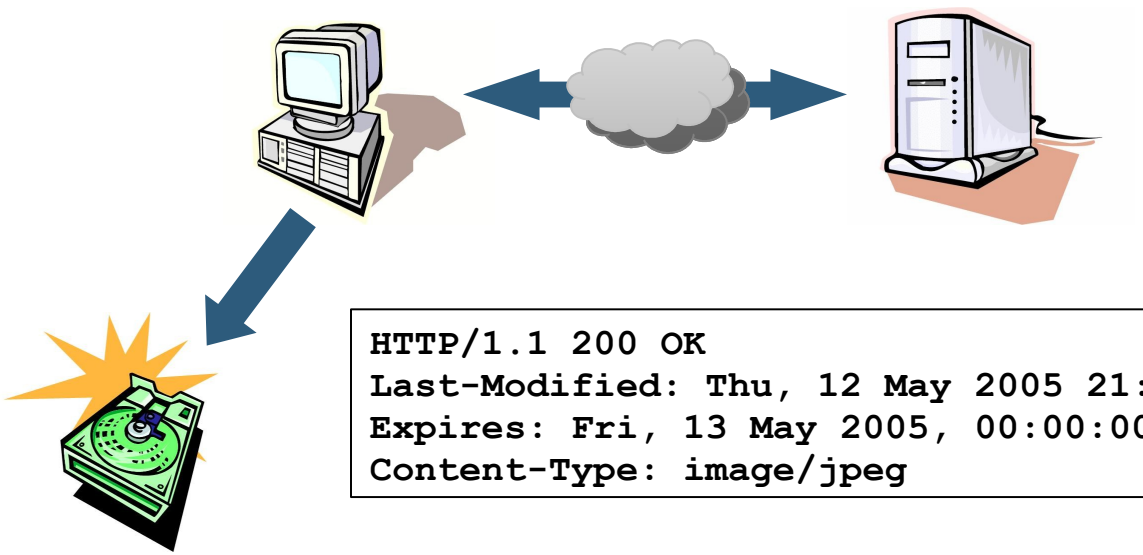
```
HTTP/1.1 200 OK
Date: Thu, 12 May 2005 22:59:02 GMT
Last-Modified: Thu, 12 May 2005 21:08:50 GMT
Content-Length: 2750
Connection: close
Content-Type: text/html

<html><head>
<title>
...
```

Private Caches

- Browsers save content in private caches

```
GET /logo.jpg HTTP/1.1
Host: www.server.com
```

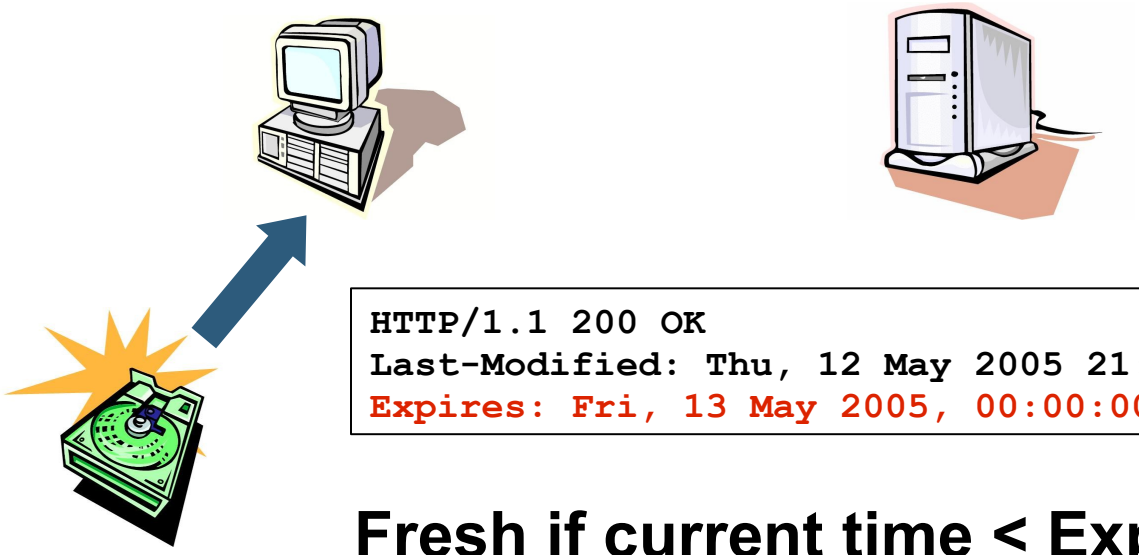


```
HTTP/1.1 200 OK
Last-Modified: Thu, 12 May 2005 21:08:50 GMT
Expires: Fri, 13 May 2005, 00:00:00 GMT
Content-Type: image/jpeg
```

Browser Cache

Fresh Content

- Reload before the Expires time? No net hit!
  - Content is considered fresh



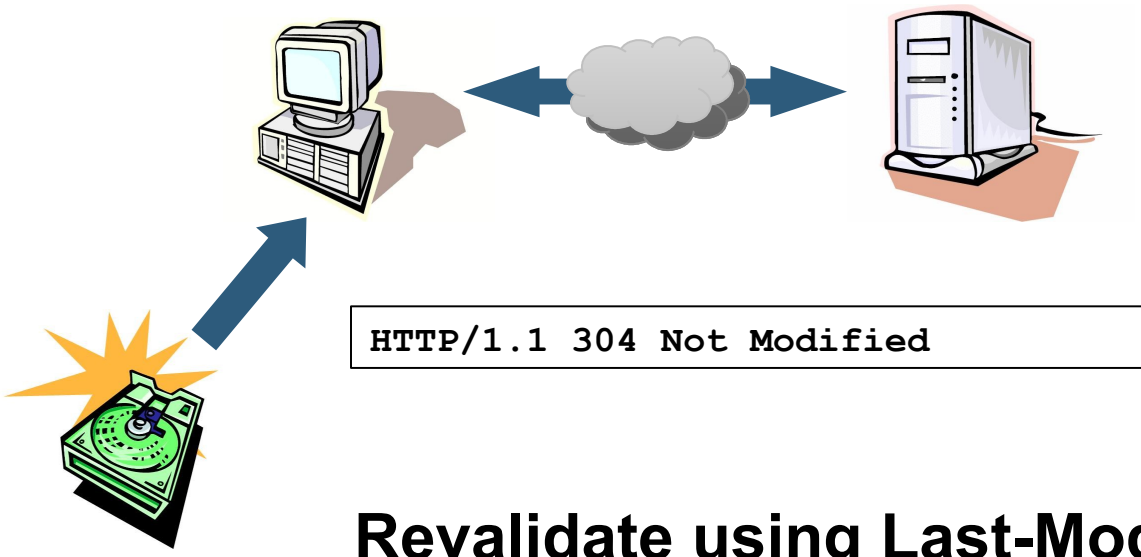
```
HTTP/1.1 200 OK
Last-Modified: Thu, 12 May 2005 21:08:50 GMT
Expires: Fri, 13 May 2005, 00:00:00 GMT
```

Fresh if current time < Expires time

Conditional GET

- Use conditional GET for revalidation

```
GET /logo.jpg HTTP/1.1
Host: www.server.com
If-Modified-Since: Thu, 12 May 2005 21:08:50 GMT
```



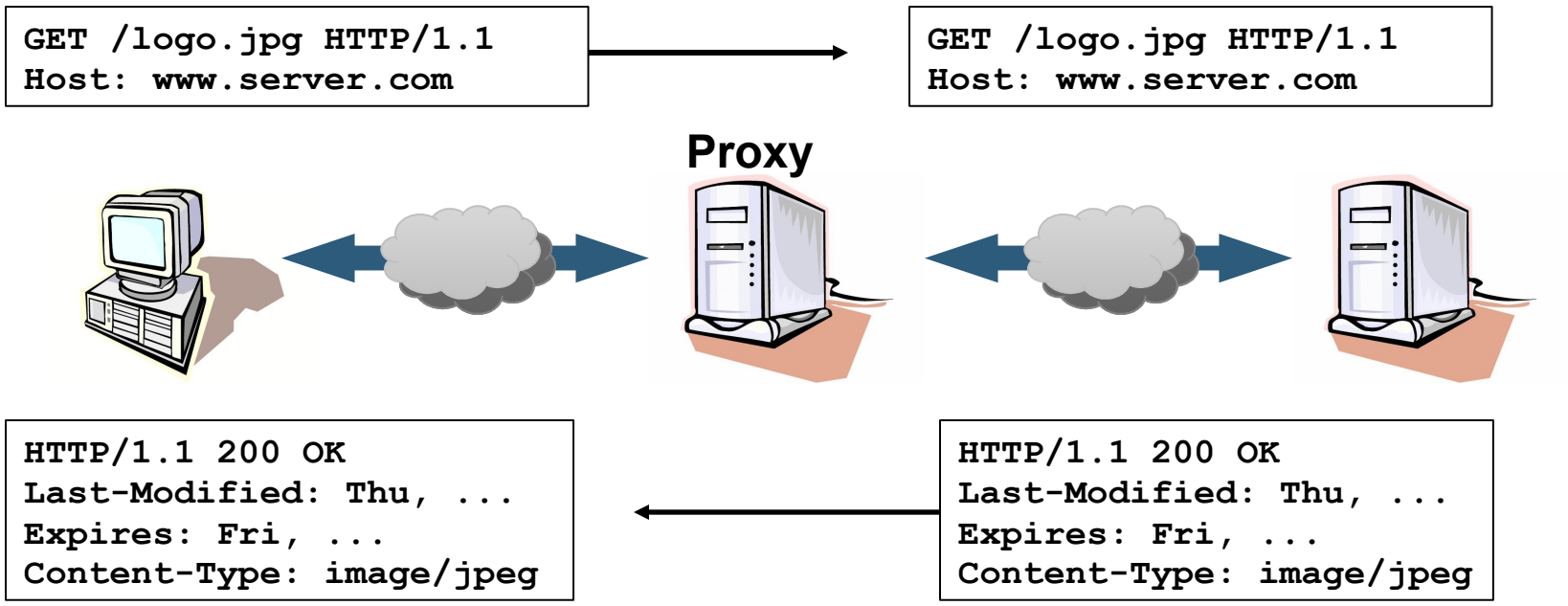
```
HTTP/1.1 304 Not Modified
```

Revalidate using Last-Modified time

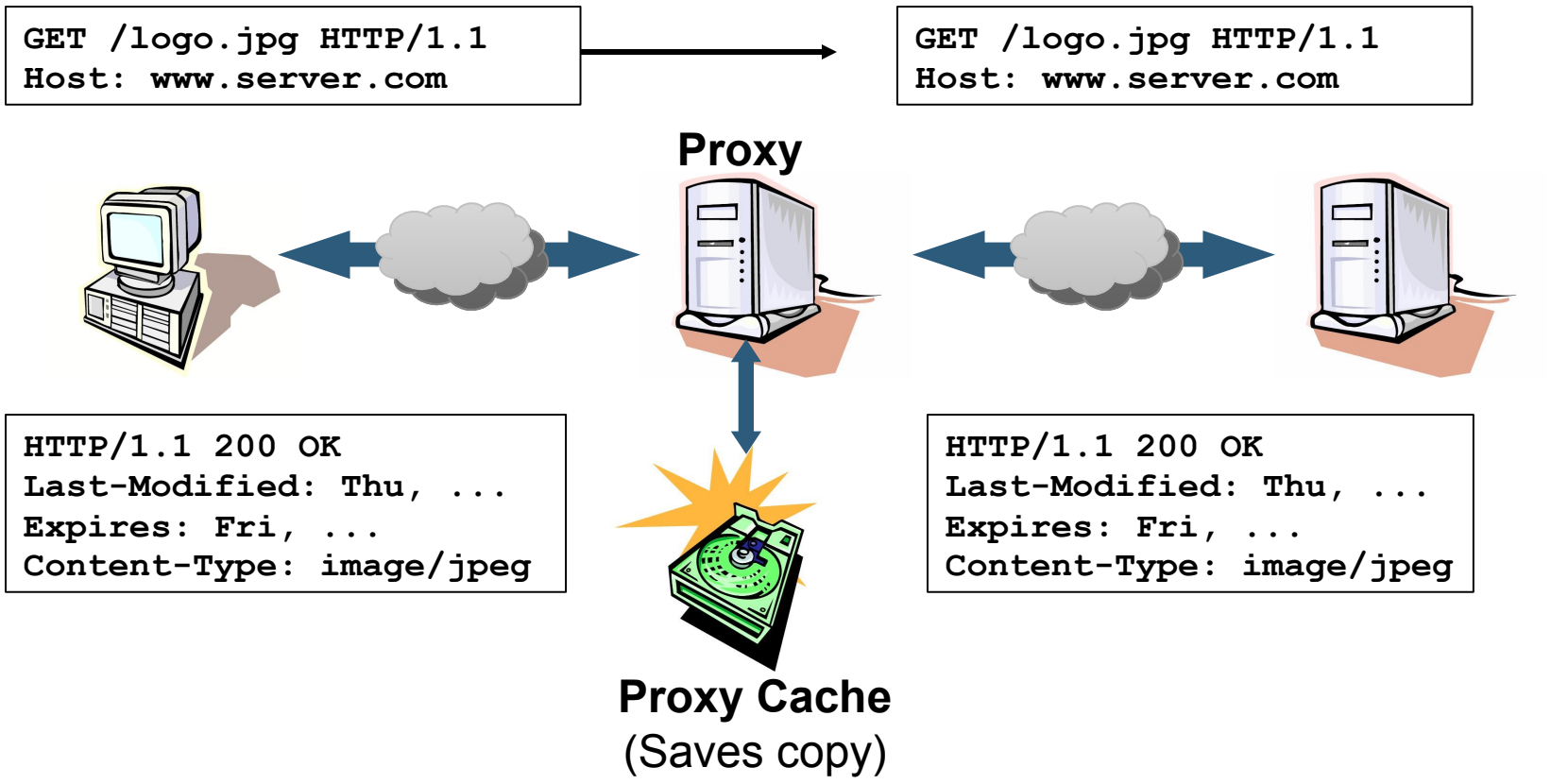
Proxies

- Proxies act as go-betweens in an HTTP request
  - Client request goes through the proxy
  - Proxies can be set in the browser, or handled by the underlying network
- Some proxies cache
  - Corporate, university, national
- Some proxies only pass through
  - Firewall, anonymizer, network gateway

Non-Caching Proxy

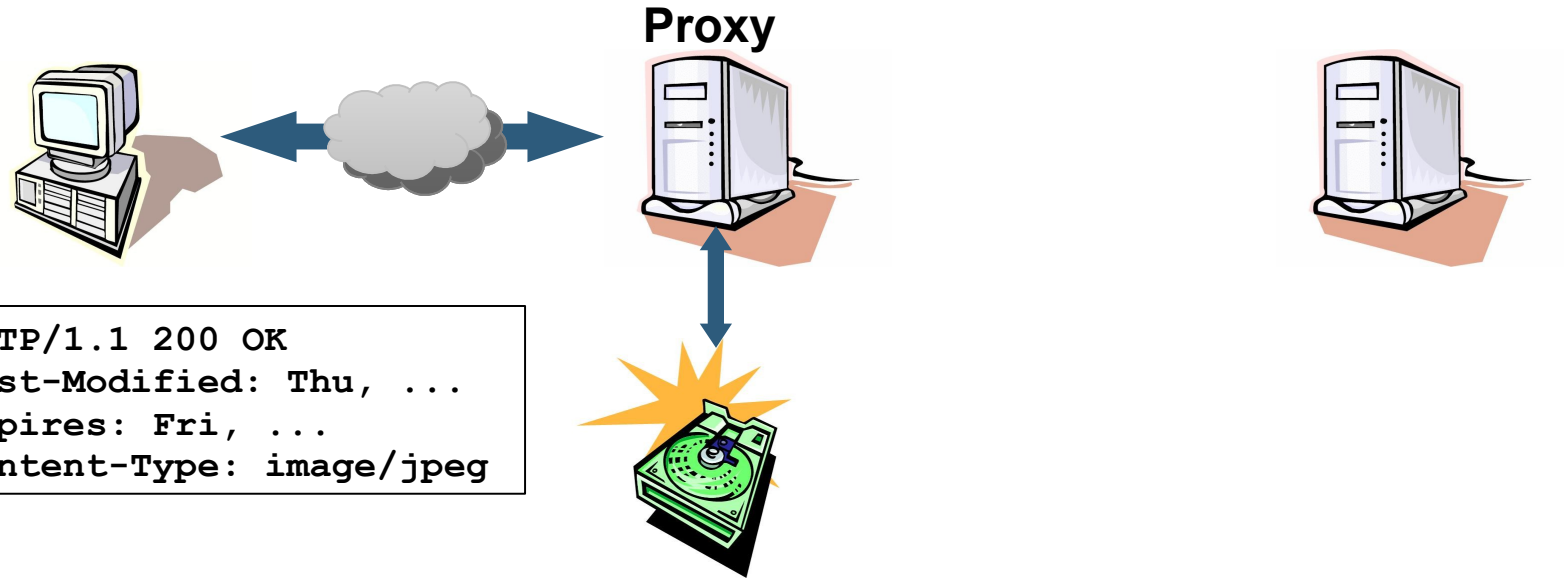


Caching Proxy: Miss



Caching Proxy: Hit

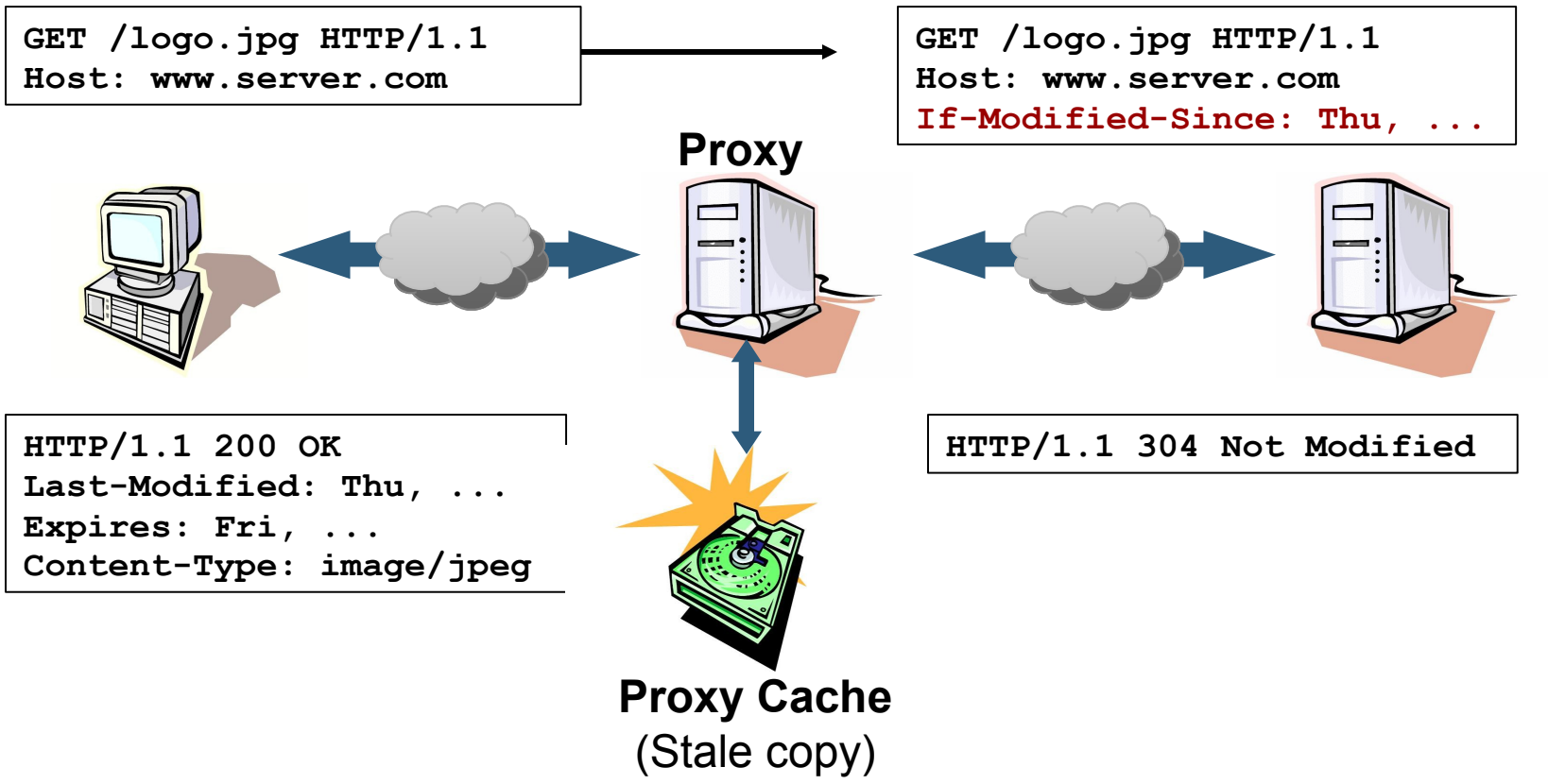
```
GET /logo.jpg HTTP/1.1
Host: www.server.com
```




```
HTTP/1.1 200 OK
Last-Modified: Thu, ...
Expires: Fri, ...
Content-Type: image/jpeg
```

Proxy Cache
(Fresh copy!)

Caching Proxy: Revalidation



Agenda

How HTTP Works

Top Five Techniques for Web Publishers

Review

Content Assumptions

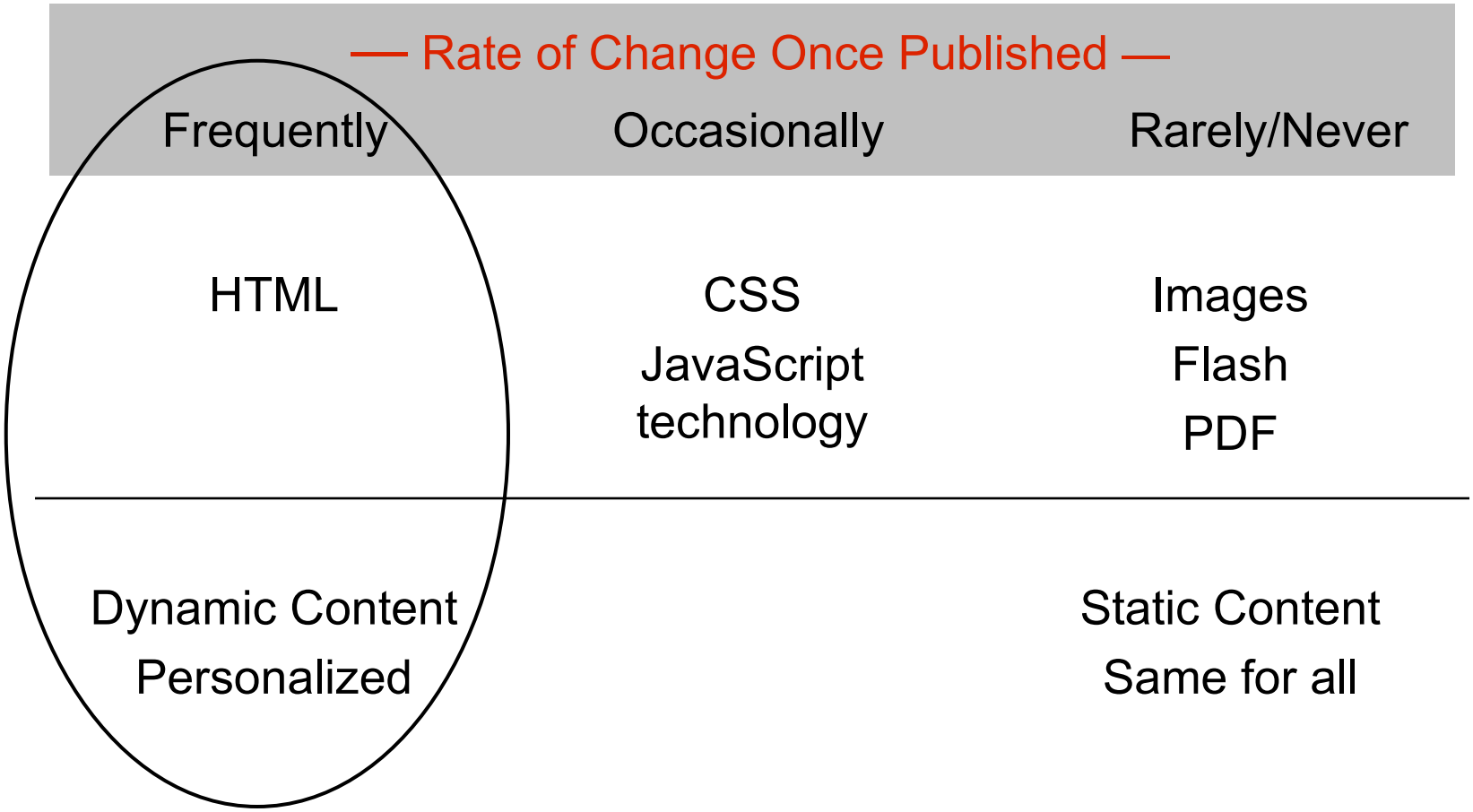
— Rate of Change Once Published —		
Frequently	Occasionally	Rarely/Never
HTML	CSS JavaScript technology	Images Flash PDF

Dynamic Content Personalized	Static Content Same for all

Top 5 Techniques

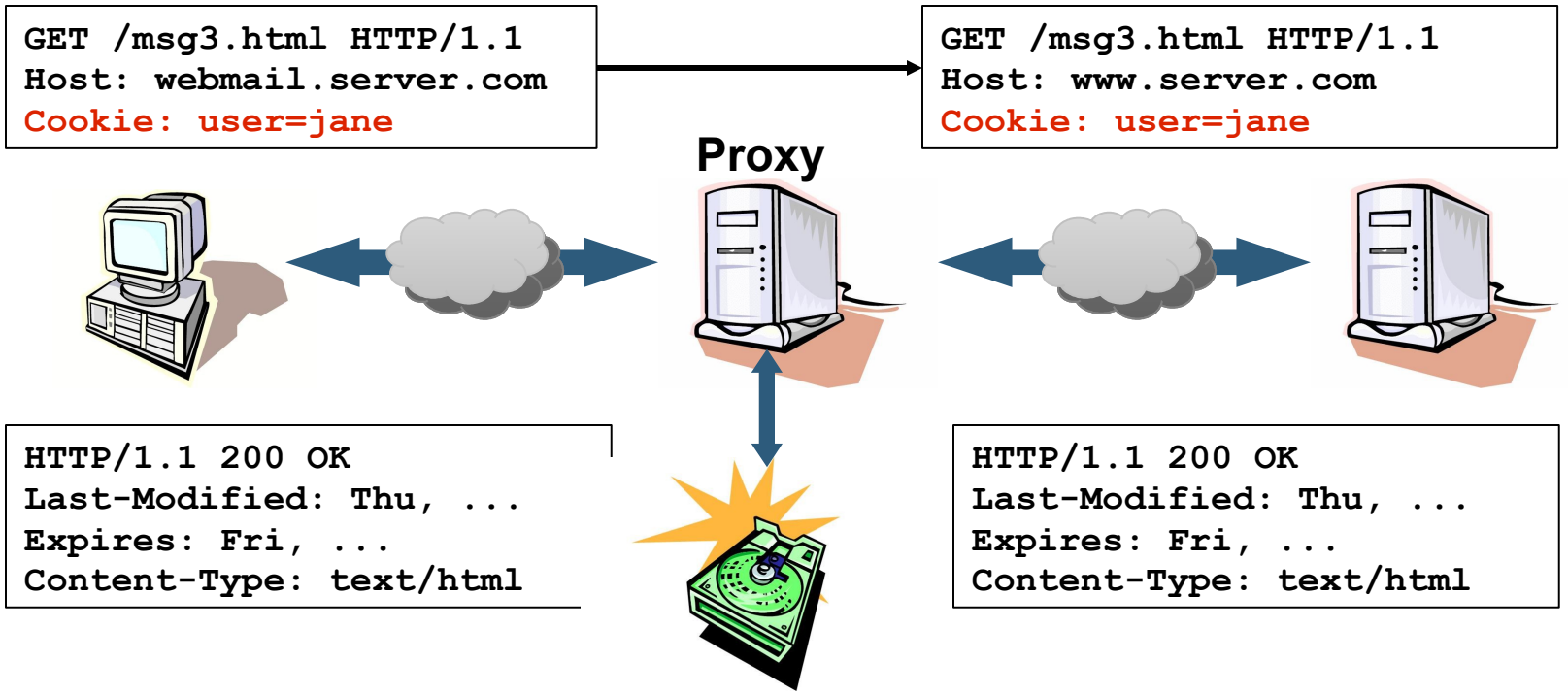
- Michael Radwin's **Top 5** Techniques for Publishers
  1. Use "Cache-Control: private" for personalized content
  2. Implement "Images Never Expire" policy
  3. Use a cookie-free TLD for static content
  4. Use Apache defaults for CSS and JavaScript technology
  5. Use random strings in URLs for accurate hit metering or very sensitive information

1. Cache-Control: private



Bad Caching (1)

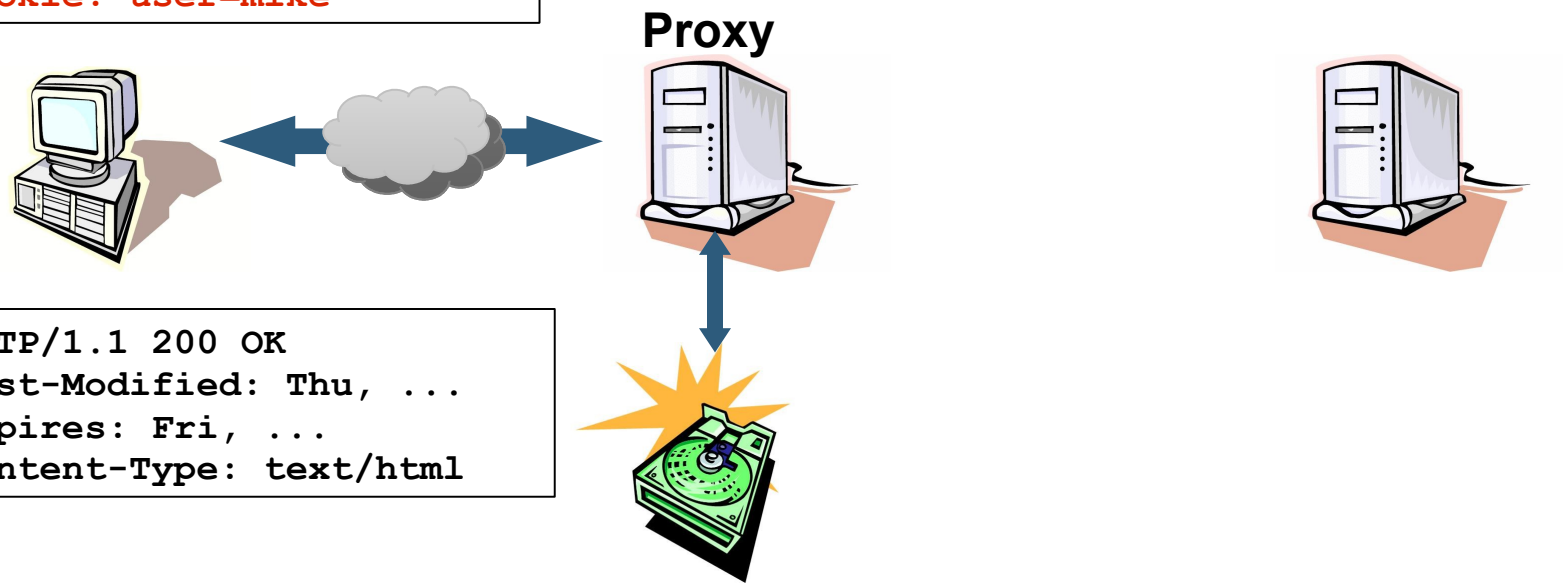
- The URL isn't all that matters



Bad Caching (2)

- Witness a false positive cache hit

```
GET /msg3.html HTTP/1.1
Host: webmail.server.com
Cookie: user=mike
```




```
HTTP/1.1 200 OK
Last-Modified: Thu, ...
Expires: Fri, ...
Content-Type: text/html
```

Proxy Cache
(Fresh copy of Jane's)

What's Cacheable?

- HTTP/1.1 allows caching anything by default
  - Unless overridden with Cache-Control header
- In practice, most caches avoid anything with:
  - Restrictive **`Cache-Control/Pragma`** headers
  - Cookie/Set-Cookie headers
  - **`WWW-Authenticate/Authorization`** headers
  - POST/PUT methods
  - 302/307 status codes (redirects)
  - SSL content

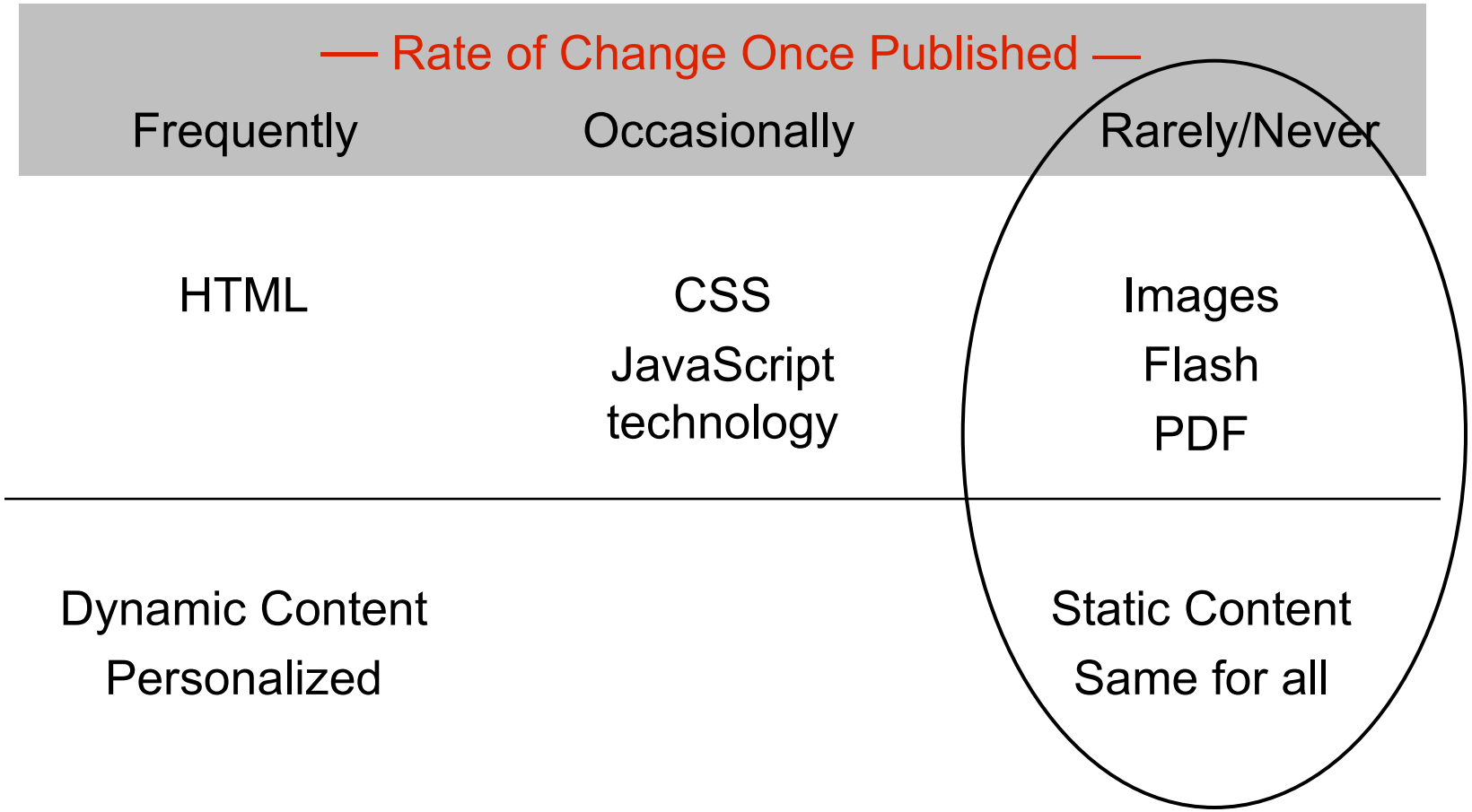
Cache-Control: private

- With personalized content, be explicit with your cache policy
  - **`Cache-Control: private`**
  - Disallows shared caching; allows private (browser) caching
- Avoid “personalization leakage” with a single link in httpd.conf or .htaccess
  - **`Header set Cache-Control private`**

Cache-Control

- To explicitly permit caching of cookied or authenticated content
  - **`Cache-Control: public`**
  - Explicitly allows shared caching
- To allow shared caching, but force revalidation
  - **`Cache-Control: public, no-cache`**
  - The **`no-cache`** forces revalidation on each request
  - Efficiently serve protected content!

2. Images Never Expire



The Policy

- Dictate that images (icons, logos) once published never change
  - Set Expires header 10 years in the future
- Use new names for new versions
  - `http://us.yimg.com/i/new.gif`
  - `http://us.yimg.com/i/new2.gif`

- Tradeoffs
  - More difficult for designers
  - Much faster experience, bandwidth savings

mod_expires

- Apache module **mod_expires** makes this policy easy

```
# This works with both HTTP/1.0 and HTTP/1.1
# 315360000 = 10*365*24*60*60
ExpiresActive On
ExpiresByType image/gif A315360000
ExpiresByType image/jpeg A315360000
ExpiresByType image/png A315360000
```

mod_headers

- Apache **mod_headers** can also manage Cache-Control

```
# Works with both HTTP/1.0 and HTTP/1.1
<FilesMatch "\.(gif|jpe?g|png)$">
Header set Expires "Mon, 28 Jul 2014 23:30:00 GMT"
</FilesMatch>

# Works with HTTP/1.1 only
<FilesMatch "\.(gif|jpe?g|png)$">
Header set Cache-Control "max-age=315360000"
</FilesMatch>
```

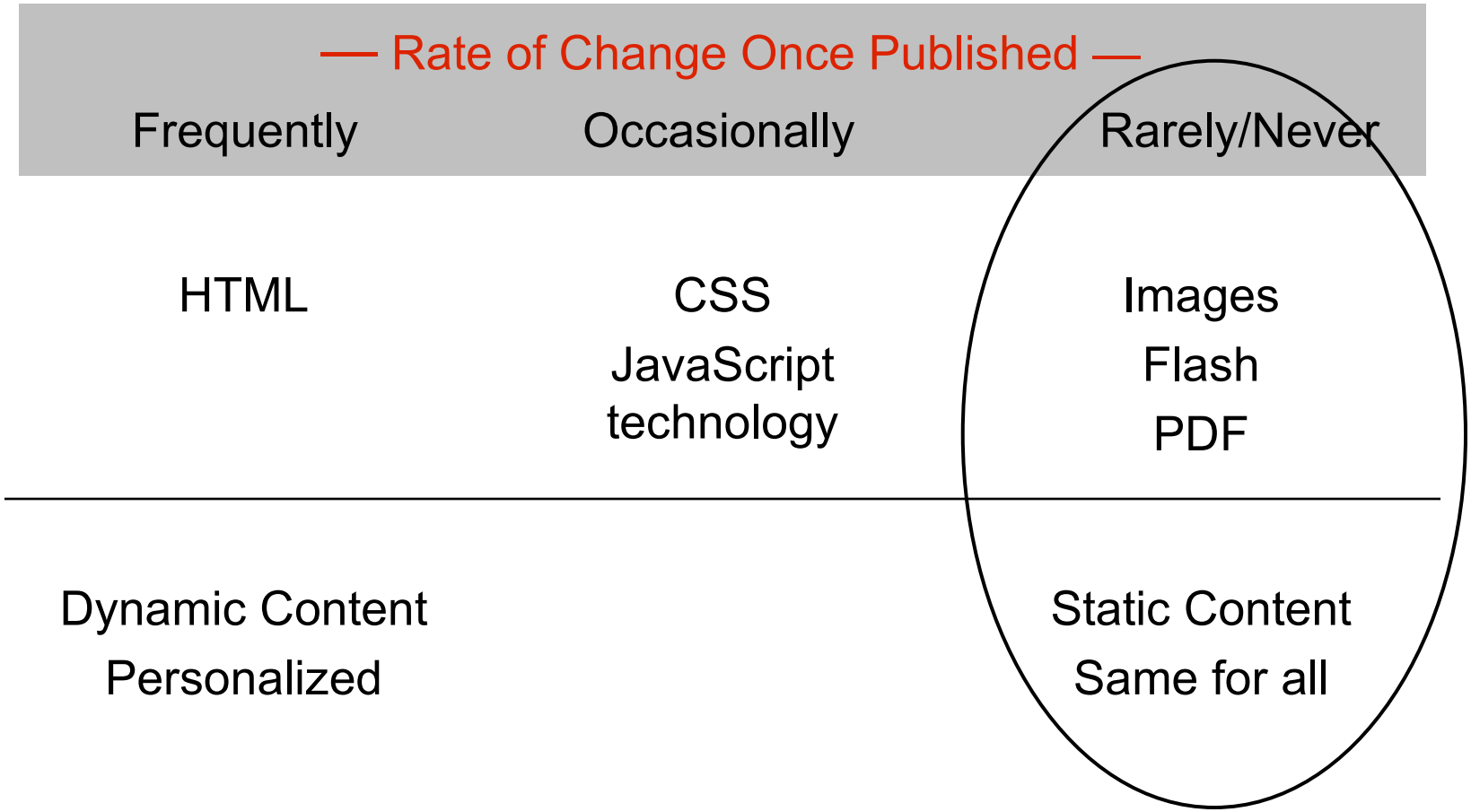
mod_images_never_expire

- Answer 304 to all image revalidation
  - This module runs at URI translation hook

```
static int translate_imgexpire(request_rec *r) {
  const char *ext;
  if ((ext = strrchr(r->uri, '.')) != NULL) {
    if (strcasecmp(ext,".gif") == 0 || strcasecmp(ext,".jpg") == 0 ||
        strcasecmp(ext,".png") == 0 || strcasecmp(ext,".jpeg") == 0) {
      if (ap_table_get(r->headers_in,"If-Modified-Since") != NULL ||
          ap_table_get(r->headers_in,"If-None-Match") != NULL) {
        /* Don't bother checking filesystem, just hand back a 304 */
        return HTTP_NOT_MODIFIED;
      }
    }
  }
  return DECLINED;
}
```

Also http://use.perl.org/~geoff/journal/22049

3. Cookie-free TLD for Static



The Policy

- Use a separate domain for static content
  - A domain without cookies
  - **`www.server.com`** for HTML
  - **`static.server.net`** for images, PDFs, etc
- Many proxies won't cache Cookie requests
  - Static pages aren't personalized, cookies aren't needed
- Necessary?
  - Cookie path settings are only inclusive
  - **`Cache-Control`**: public relies on proxy

A Typical Request

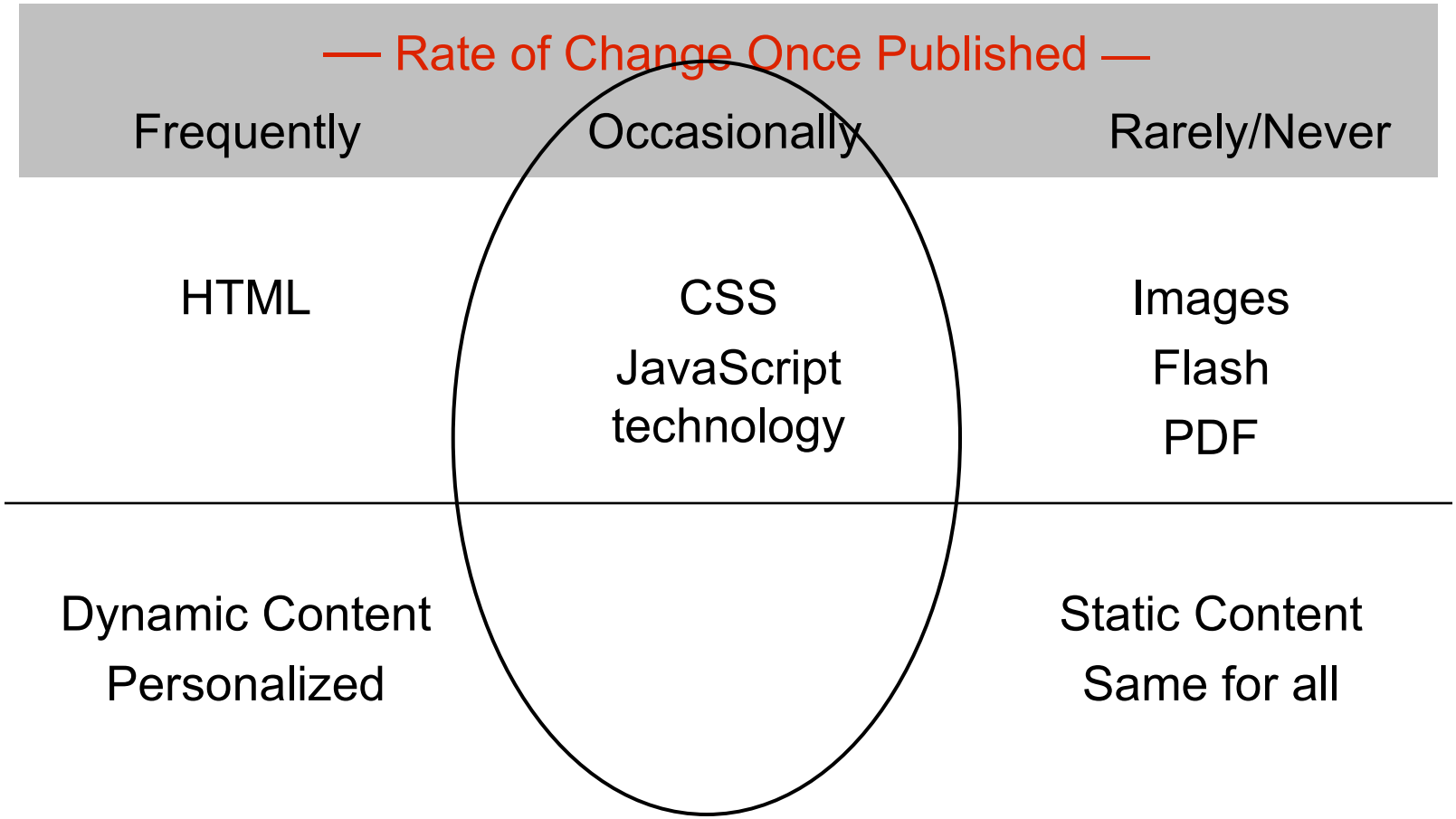
```
GET /i/app/logo.gif HTTP/1.1
Host: www.server.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.7) Gecko/20040707
  Firefox/0.8
Accept: application/x-shockwave-flash,text/xml,application/xml,application/xhtml+xml,
  text/html;q=0.9,text/plain;q=0.8,video/x-mng,image/png,image/jpeg,image/gif;q=0.2,*/*;q=0.1
Cookie: U=mt=vtC1tp2MhYv9RL5BlpxYRFN_P8DpMJoamllEcA--&ux=IIr.AB&un=42vnticvufc8v;
  brandflash=1; B=amfco1503sgp8&b=2; F=a=NC184LcsvfX96G.JR27qSjCHu7bII3s.tXa44psML
  liFtVoJB_m5wecWY_.7&b=K1It; LYC=l_v=2&l_lv=7&l_l=h03m8d50c8bo&l_s=3yu2qxz5zvwquww
  uzv22wrwr5t3w1zsr&l_lid=14rsb76&l_r=a8&l_um=1_0_1_0_0; GTSessionID835990899023=
  83599089902340645635; Y=v=1&n=6eecgejj7012f&l=h03m8d50c8bo/o&p=m012o33013000007&
  jb=16|47|&r=a8&lg=us&intl=us&np=1; PROMO=SOURCE=fp5; YGCV=d=;T=z=iTu.ABiZD/AB6dPWoqXi
  bIcTzc0BjY3TzI3NTY0MzQ&a=YAE&sk=DAAwRz5HlDUN2T&d=c2wBT0RBekFURXdPRFV3TWpFek5ETS0BY
  QFZQUUBb2sBWlcwLQF0aXABWUhaTVBBAXp6AWlUdS5BQmdXQQ--&af=QUFBQ0FDQURCOUFIQUJBQ0FEQUtBTE
  FNSDAmdHM9MTA5MDE4NDQxOCZwcz1lOG83MUVYcTYxOVouT2Ftc1ZFZUhBLS0-; LYS=l_fh=0&l_vo=myla;
  PA=p0=dg13DX4Ndgk-&p1=6L5qmg--&e=xMv.AB;YP.us=v=2&m=addr&d=1525+S+Robertson+Blvd%01
  Los+Angeles%01CA%0190035-4231%014480%0134.051590%01-118.384342%019%01a%0190035
Referer: http://www.server.com/app/page.jsp?abc=123&def=456
Accept-Language: en-us,en;q=0.7,he;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
```

A Request Without Cookies

```
GET /i/app/logo.gif HTTP/1.1
Host: static.server.net
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.7) Gecko/20040707
  Firefox/0.8
Accept: application/x-shockwave-flash,text/xml,application/xml,application/xhtml+xml,
  text/html;q=0.9,text/plain;q=0.8,video/x-mng,image/png,image/jpeg,image/gif;q=0.2,*/*;q=0.1
Referer: http://www.server.com/app/page.jsp?abc=123&def=456
Accept-Language: en-us,en;q=0.7,he;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
```

- Much more cacheable
- Bonus: smaller GET request
  - Dial-up MTU size 576 bytes, PPPoE 1492
  - 1450 bytes reduced to 550

4. Apache Defaults for CSS/JS



Revalidation Works Well

- Apache handles revalidation for static content
  - Browser sends If-Modified-Since request
  - Server replies with short 304 Not Modified
  - No special configuration needed
- Use if you can't predict when content will change
  - Page designers can change immediately
  - No renaming necessary
- Only cost: Extra HTTP transaction for 304

Revalidation with Servlets

- How can a server know when servlet content changes?
  - The .class timestamp is meaningless
  - Help the server with getLastModified()

```
public long getLastModified(HttpServletRequest req) {
   return dataModified.getTime();
 }
```

- Called before doGet(), possibly avoiding doGet()

More Caching Tips

- Avoid URLs that appear dynamic
  - “cgi-bin”, “.cgi”, “.jsp”, or “?”
  - Use extra path info instead
- Generate static content headers
  - Last-Modified, ETag
- Send explicit Cache-Control or Expires
  - Dictate how and how long to cache
- Don’t rely on META tags inside HTML content
  - Proxies don't usually peer into content

5. Random URL Strings

— Rate of Change Once Published —		
Frequently	Occasionally	Rarely/Never
HTML	CSS JavaScript technology	Images Flash PDF

Dynamic Content Personalized	Static Content Same for all

Accurate Ad Stats: Trusting

- How to guarantee an accurate count for advertisement impressions?
- If you trust proxies
  - Send **`Cache-Control`**: must-revalidate
  - Count 304 **`Not Modified`** log entries as hits

Accurate Ad Stats: Untrusting

- If you don’t trust proxies
  - Ask client to fetch uncacheable image URL
  - Return 302 to highly cacheable image file
  - Count 302 as hits
  - Don't bother to look at cacheable server log
- Uncacheable?
  - Add random URL string

Ad Stats Example (1)

- Adding random URL strings

```
<script type="text/javascript">
  var r = Math.random();
  var t = new Date();
  document.write("<img width='109' height='52'
    src='http://ads.server.com/ad/foo/bar.gif?t=" +
    t.getTime() + ";r=" + r + "'>");
</script>
<noscript>
  <img width="109" height="52" src=
   "http://ads.server.com/ad/foo/bar.gif?js=0">
</noscript>
```

Ad Stats Example (2)

- Redirect from uncacheable to highly cacheable (on static content server)

```
GET /ad/foo/bar.gif?t=1090538707;r=0.510772917234983 HTTP/1.1
Host: ads.server.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US;
  rv:1.7) Gecko/20040707 Firefox/0.8
Referer: http://www.server.com/foo/bar.jsp?abc=123&def=456
Cookie: uid=C50DF33E-E202-4206-B1F3-946AEDF9308B

HTTP/1.1 302 Moved Temporarily
Date: Wed, 28 Jul 2004 23:45:06 GMT
Cache-Control: max-age=0,no-cache,no-store
Expires: Tue, 11 Oct 1977, 01:23:45 GMT
Pragma: no-cache
Location: http://img.server.net/i/foo/bar.gif
Content-Type: text/html

<a href="http://img.server.net/i/foo/bar.gif">Moved</a>
```

Ad Stats Example (3)

- Serving the static image
  - Headers make it highly cacheable

```
GET /i/foo/bar.gif HTTP/1.1
Host: img.server.net
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US;
  rv:1.7) Gecko/20040707 Firefox/0.8
Referer: http://www.server.com/foo/bar.jsp?abc=123&def=456

HTTP/1.1 200 OK
Date: Wed, 28 Jul 2004 23:45:07 GMT
Last-Modified: Mon, 05 Oct 1998 18:32:51 GMT
ETag: "69079e-ad91-40212cc8"
Cache-Control: public,max-age=315360000
Expires: Mon, 28 Jul 2014 23:45:07 GMT
Content-Length: 6096
Content-Type: image/gif

GIF89a...
```

Turning Off All Caching

- To turn off all caching
  - A “Please Wait...” message
  - Any constantly changing content

```
// Set to expire far in the past
res.setHeader("Expires", "Sat, 6 May 1995 12:00:00 GMT");

// Set standard HTTP/1.1 no-cache headers
res.setHeader("Cache-Control",
              "no-store, no-cache, must-revalidate");

// Set IE extended HTTP/1.1 no-cache headers (addHeader)
res.addHeader("Cache-Control",
              "post-check=0, pre-check=0");

// Set standard HTTP/1.0 no-cache header
res.setHeader("Pragma", "no-cache");
```

Agenda

How HTTP Works

Top Five Techniquest for Web Publishers

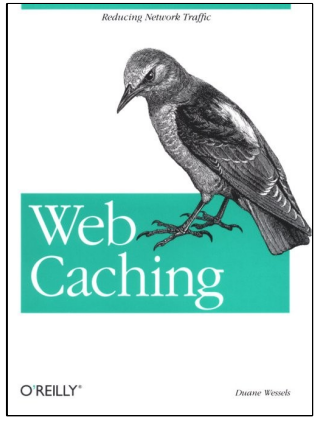
Review

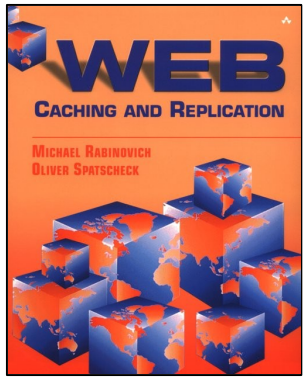
Review

1. Use “Cache-Control: private” for personalized content
2. Implement “Images Never Expire” policy
3. Use a cookie-free TLD for static content
4. Use Apache defaults for CSS and JavaScript
5. Use random strings in URLs for accurate hit metering or very sensitive information

Resources

- Cacheability Engine and Tutorial
  - http://www.mnot.net/cacheability/
  - http://www.mnot.net/cache_docs/
- Michael Radwin's blog
  - http://www.radwin.org/michael/blog/





Q&A

Jason Hunter

Extreme Web Caching

Jason Hunter

Principal Technologist
Mark Logic
http://marklogic.com

TS-4251