Fundamentals of Web Performance Optimization

Matthew Bennett
9 November 2013

About Me

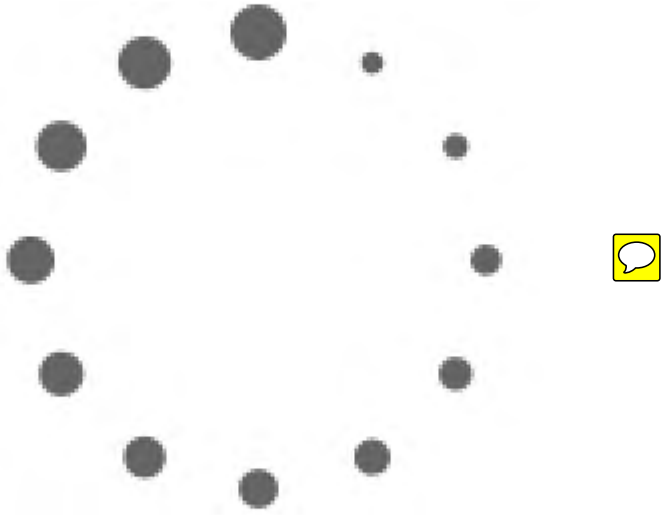
- @Lobstrosity
- C#/.NET since 2002
- ASP.NET MVC since 2008
- Off Madison Ave since 2010

About This Presentation

#dcc13 #wpo

1. Speed Is a Feature 
2. Causes of Slowness
3. How to Prevent Slowness

Slow is Notoriously Bad



Slow is Notoriously Bad

> "I distinctly remember switching [to Google] because it was blazing fast. To me, **performance is a feature**, and I simply like using fast websites more than slow websites, so naturally I'm going to build a site that I would want to use."

—Jeff Atwood

1. "Performance is a Feature", June 2011, Jeff Atwood

Slow is Notoriously Bad

Speed is a feature.

↓

Slow is a bug.

Slow is *Objectively* Bad

2001

> "If the load-time of a web page exceeds **eight seconds**, users are unlikely to wait for its completion."

2006

> "JupiterResearch recommends that retailers make every effort to keep page rendering to no longer than **four seconds**."

2009

> "Retailers should keep page rendering to under **two seconds**."

2. "The Need for Speed II", April 2001, Zona Research
3. "Retail Web Site Performance", June 2006, JupiterResearch
4. "eCommerce Web Site Performance Today", August 2009, Forrester Consulting

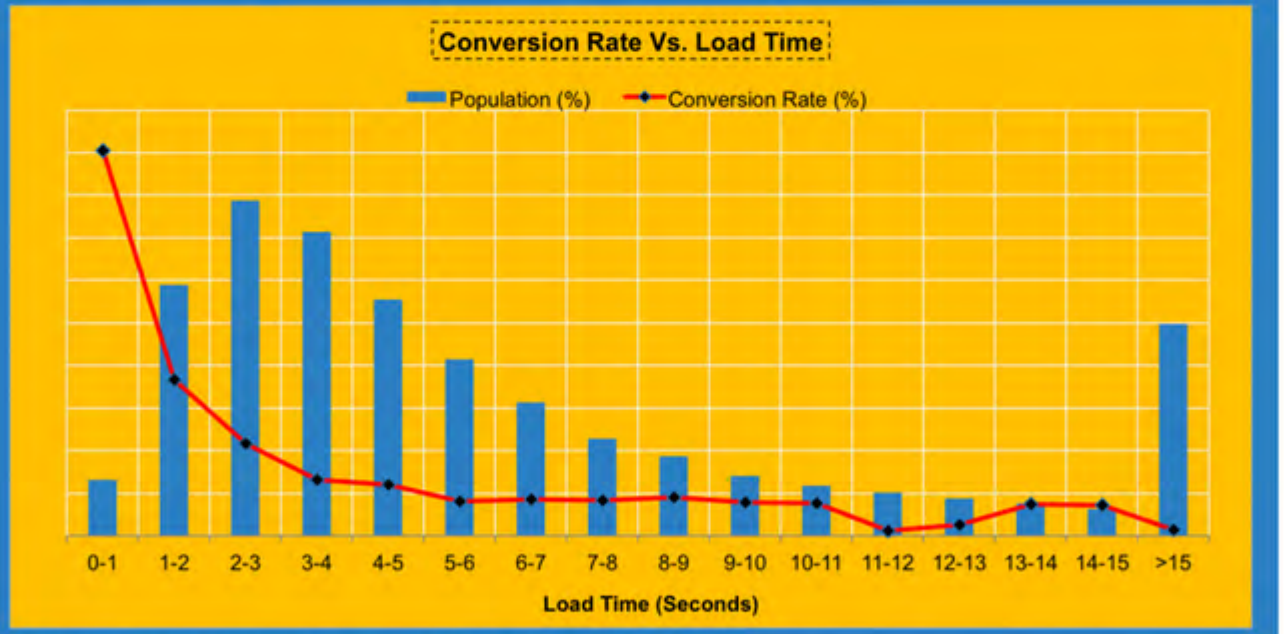
Slow is Objectively Bad

Impact of site performance on overall site conversion rate....

Baseline – 1 in 2 site visits had response time > 4 seconds

* Sharp decline in conversion rate as average site load time increases from 1 to 4 seconds

* Overall average site load time is lower for the converted population (3.22 Seconds) than the non-converted population (6.03 Seconds)



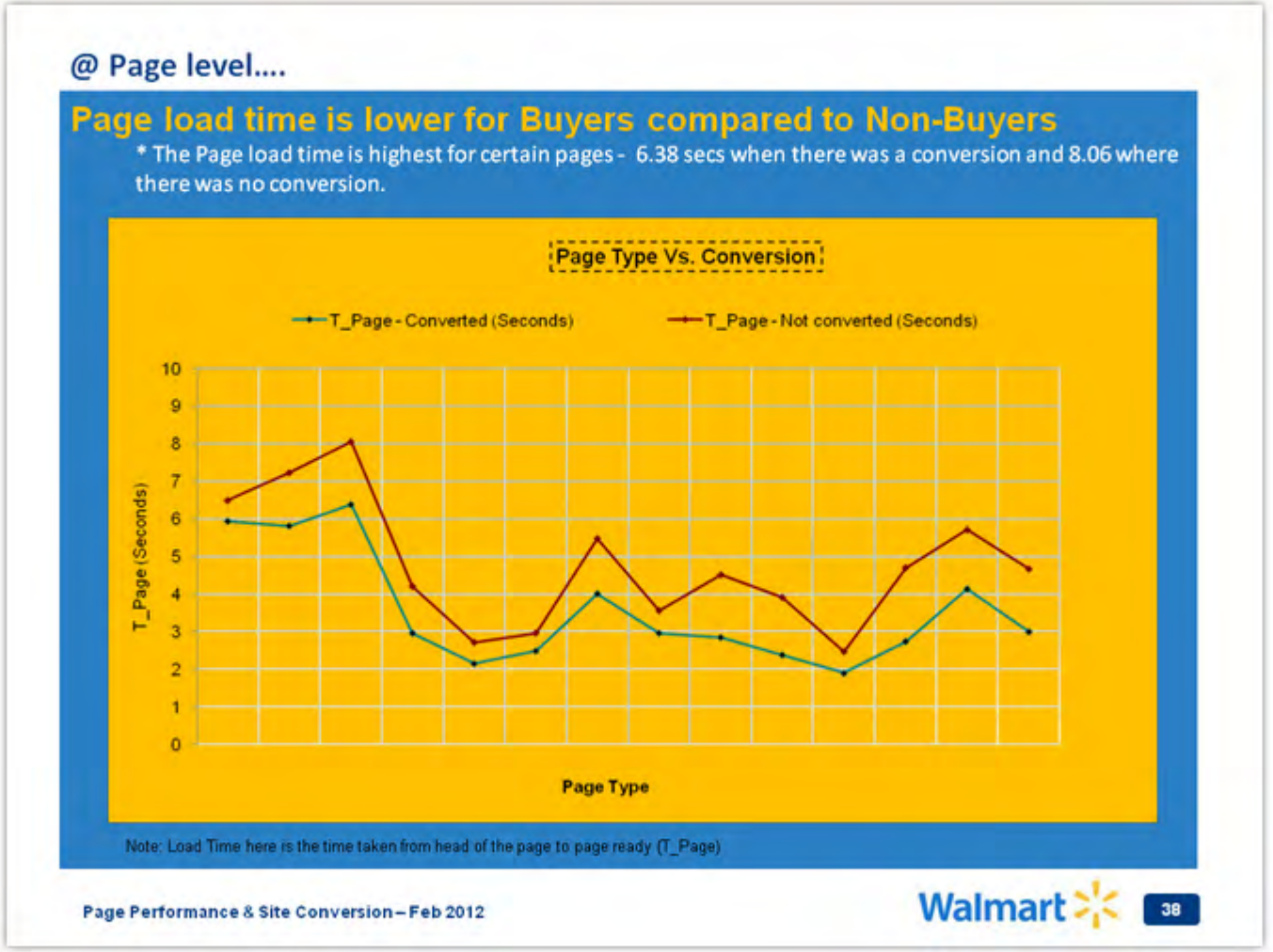
Note: Load Time here is the time taken from head of the page to page ready (T_Page)

Page Performance & Site Conversion – Feb 2012

Walmart

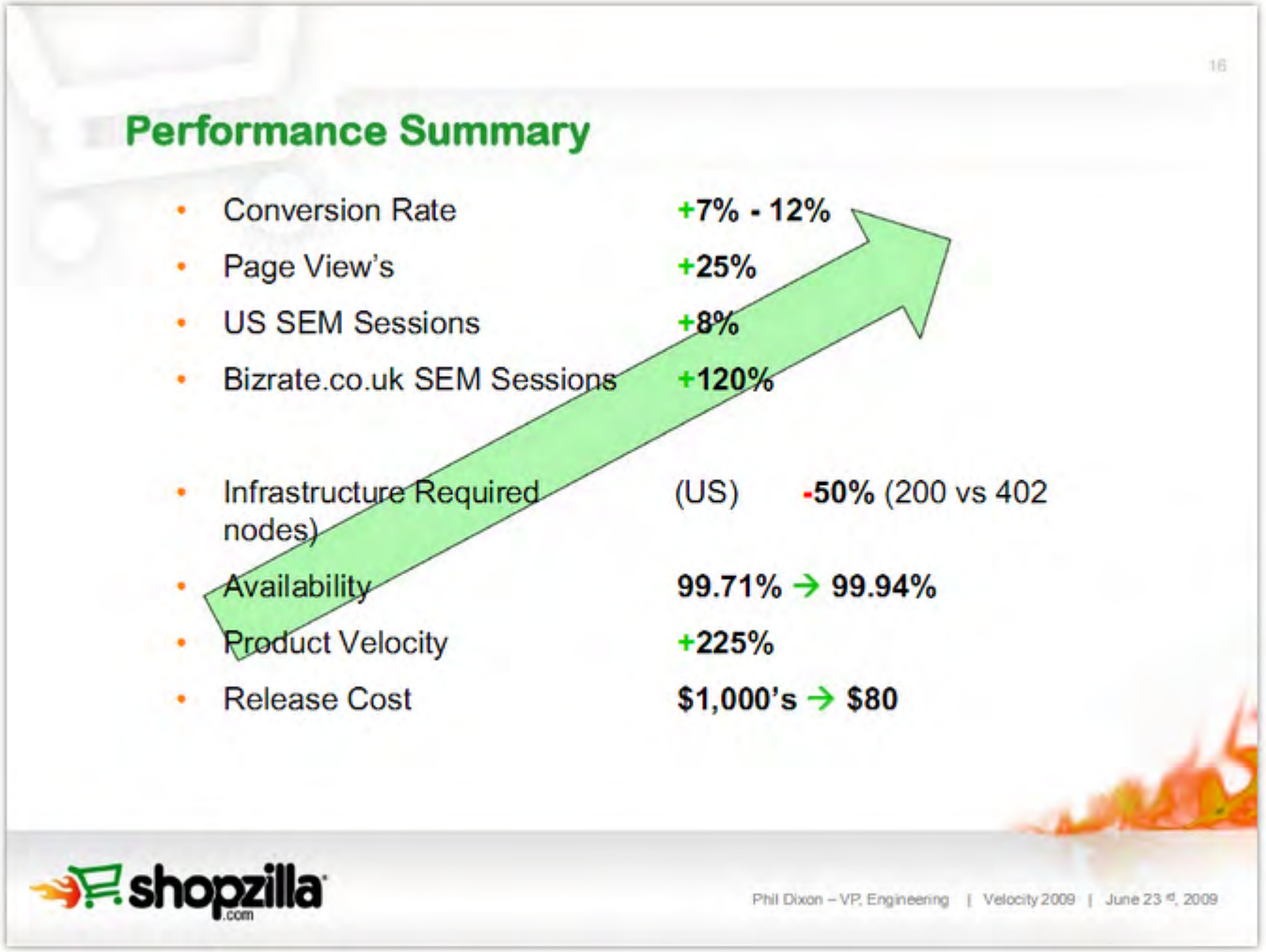
37

Slow is Objectively Bad



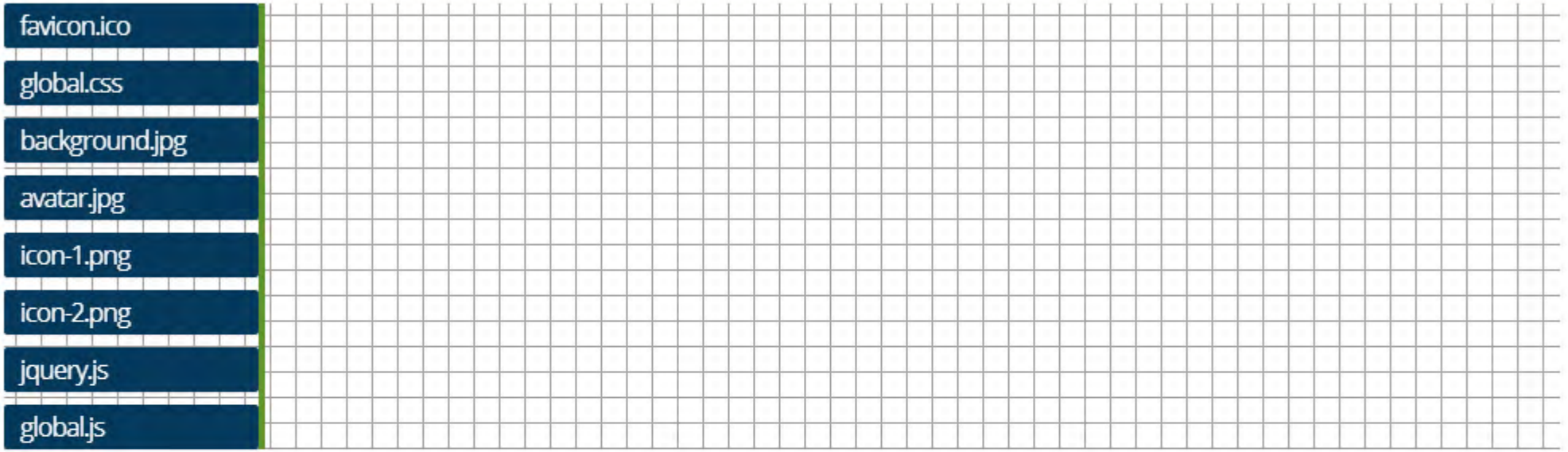
5. "Real User Monitoring", February 2012, Walmart Labs

Slow is Objectively Bad

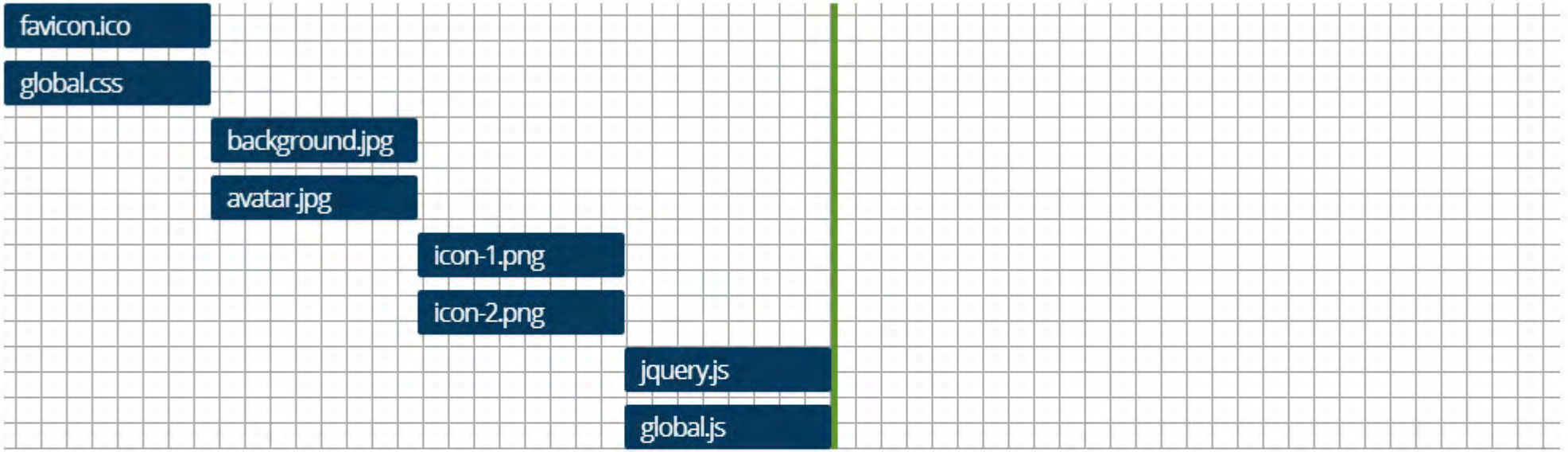


6. "Shopzilla Site Redesign—We Get What We Measure", June 2009, Phil Dixon

What Causes Slowness?



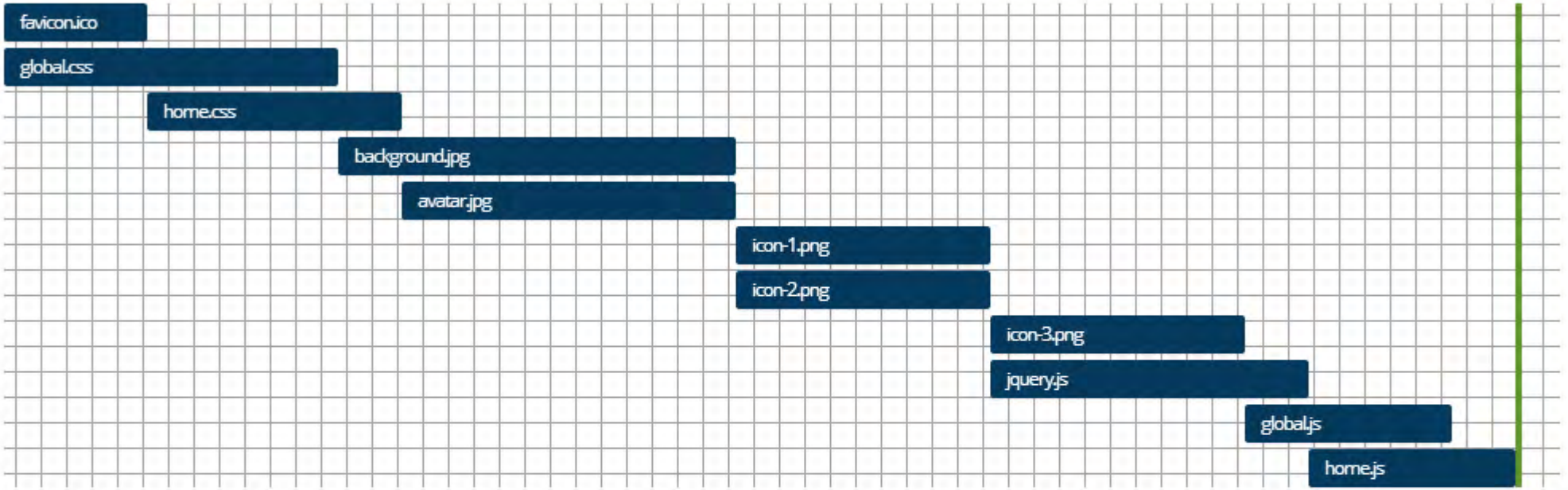
What Causes Slowness?



Section 8.1.4 of the HTTP 1.1 specification states, "A single-user client should not maintain more than 2 connections with any server or proxy."

7. "HTTP/1.1: Connections", September 2004, Fielding, et al.

What Causes Slowness?



The total size of web pages continues to rise. As of November 1st, the average is 1,614kb per page.

7. "HTTP/1.1: Connections", September 2004, Fielding, et al.
8. "Total Transfer Size & Total Requests", April 2013, HTTP Archive.

Three Basic Goals

1. Minimize Number of Requests
2. Minimize Size of Responses
3. Render Styled Content as Soon as Possible

How Do I Prevent Slowness?

Techniques You Can Use Right Now on the Front-End

1. Move Inline Styles and Scripts to External Files
2. Reference Scripts at the Bottom
3. Reference Stylesheets at the Top

Front-End Techniques That You'll Need a Tool For

4. Combine Stylesheets and Scripts
5. Minify Stylesheets and Scripts
6. Optimize Images

Techniques That Require Server-Side Functionality

7. Compress Text
8. Cache Everything

Move Inline Styles and Scripts to External Files


```
<!DOCTYPE html>
<html>
    <head>
        <style>
            body {
                font: 16px Arial;
                margin: 0;
            }
        </style>
    </head>
    <body>
        <!-- all your content here -->

        <script>
            $(function () {
                initialize();
            });
        </script>
    </body>
</html>
```

→

```
<!DOCTYPE html>
<html>
    <head>
        <link href="styles.css" />
    </head>
    <body>
        <!-- all your content here -->

        <script src="scripts.js"></script>
    </body>
</html>
```

Progressive Rendering

- Browsers render pages progressively, meaning that they parse and process elements from top to bottom in source order.
- Scripts and stylesheets block rendering until the resource is downloaded and processed.

```
<script src="i-get-processed-first.js"></script>
<p>I'm second.</p>
<p>And I'm third.</p>
```

Reference Scripts at the Bottom


```
<!DOCTYPE html>
<html>
    <head>
        <script src="scripts.js"></script>
    </head>
    <body>
        <!-- all your content here -->
    </body>
</html>
```

→

```
<!DOCTYPE html>
<html>
    <head>
    </head>
    <body>
        <!-- all your content here -->

        <script src="scripts.js"></script>
    </body>
</html>
```

Reference Stylesheets at the Top


```
<!DOCTYPE html>
<html>
    <head>
    </head>
    <body>
        <!-- all your content here -->

        <link href="styles.css" />
    </body>
</html>
```

→

```
<!DOCTYPE html>
<html>
    <head>
        <link href="styles.css" />
    </head>
    <body>
        <!-- all your content here -->
    </body>
</html>
```

Combine Stylesheets and Scripts


```
<!DOCTYPE html>
<html>
    <head>
        <link href="one.css" />
        <link href="two.css" />
    </head>
    <body>
        <!-- all your content here -->

        <script src="one.js"></script>
        <script src="two.js"></script>
        <script src="three.js"></script>
    </body>
</html>
```

→

```
<!DOCTYPE html>
<html>
    <head>
        <link href="one-and-two.css" />
    </head>
    <body>
        <!-- all your content here -->

        <script src="one-two-and-three.js">
        </script>
    </body>
</html>
```

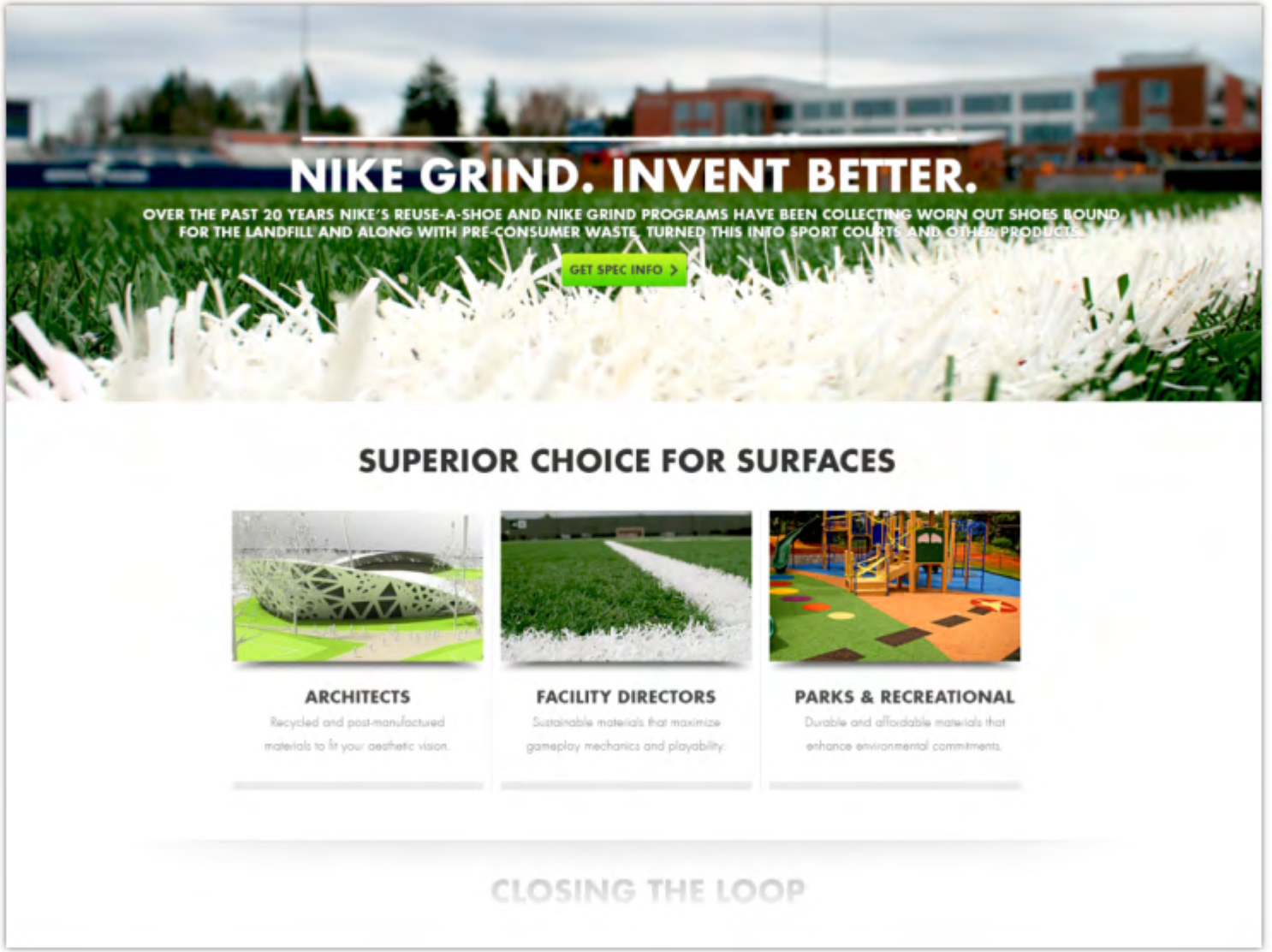
Minify Stylesheets and Scripts

- Minification is the act of removing unnecessary characters from a file.
- In stylesheets and scripts, you can get rid of comments and (most) white space and take advantage of various shortcuts.
- Much more significant reductions in scripts because variable and function names can be reduced.
- For example, jQuery 2.0.3 is 237kb. After minification, 82kb.

Optimize Images

- The average web page is 1,614kb.
- Images make up 987kb of that total.
- JPEGmini for JPEGs.
- PNGGauntlet for PNGs. 

Optimize Images



Optimize Images

File Format	Images	Total Size Before	Total Size After	Savings
PNG	11	168kb	148kb	12%
JPEG	12	983kb	306kb	69%
Total	23	1151kb	454kb	61%

Compress Text

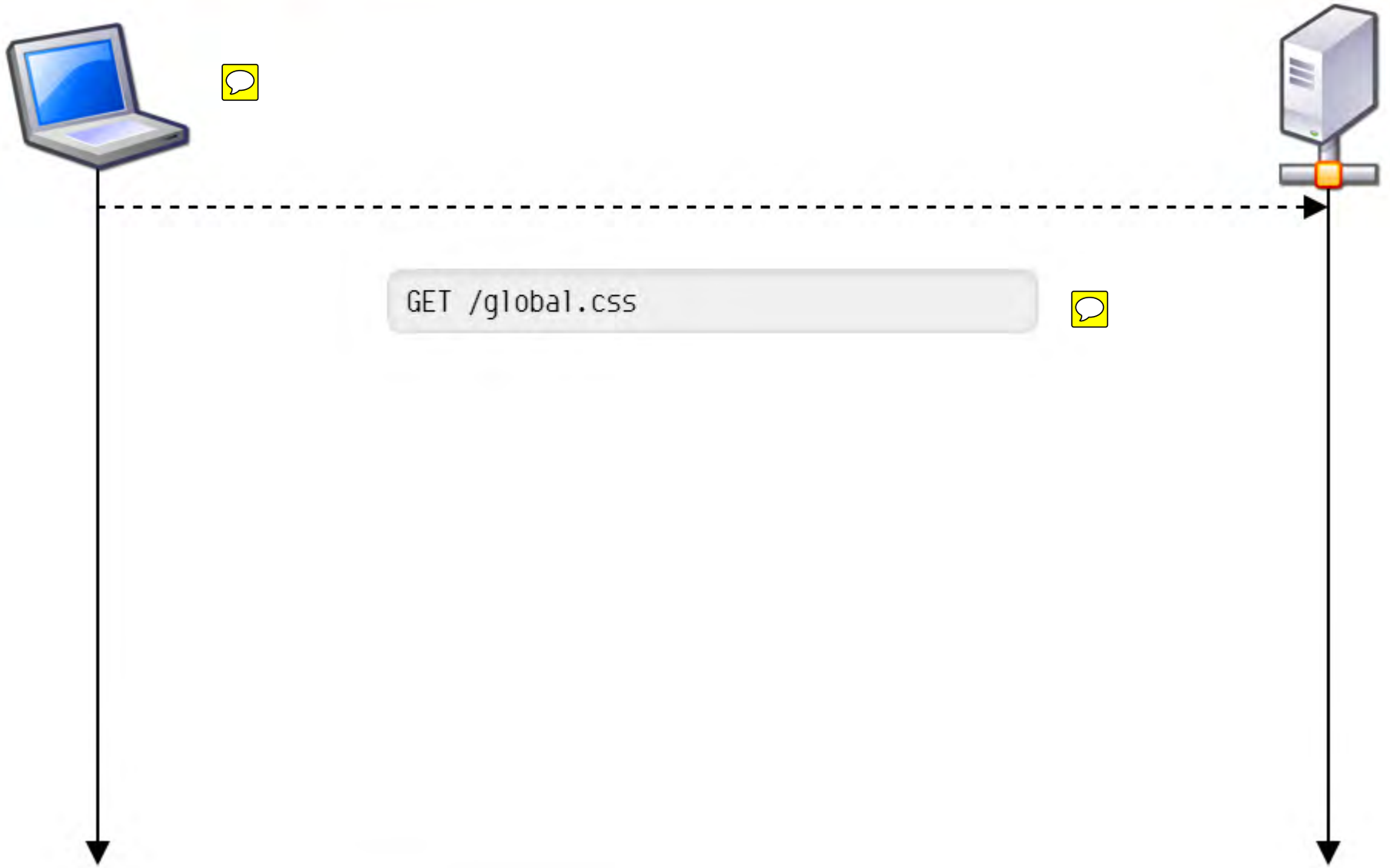
- Server compresses content before sending.
- Browser decompresses before processing.
- Reduces transfer size by roughly 65%.
- For example, jQuery 2.0.3 is 237kb. Minification reduces that to 82kb. Compression further reduces that to 29kb. 
- All major browsers and servers support either GZIP or DEFLATE.
- Chances are, your server is already doing this for you.

"There are only two hard problems in Computer Science: **cache invalidation**, naming things, and off-by-one errors."

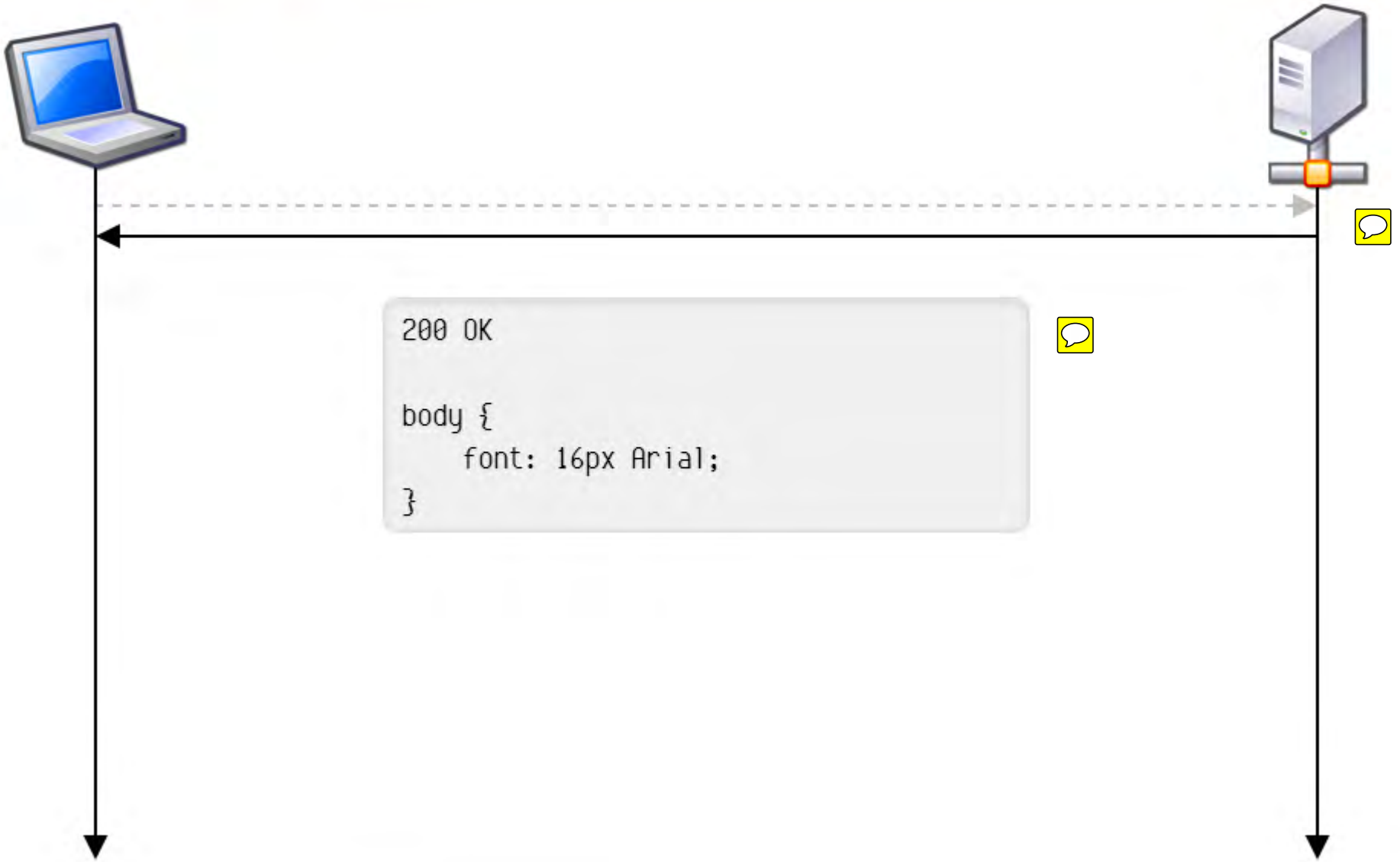
—Phil Karlton

Cache Everything

Cache Everything



Cache Everything



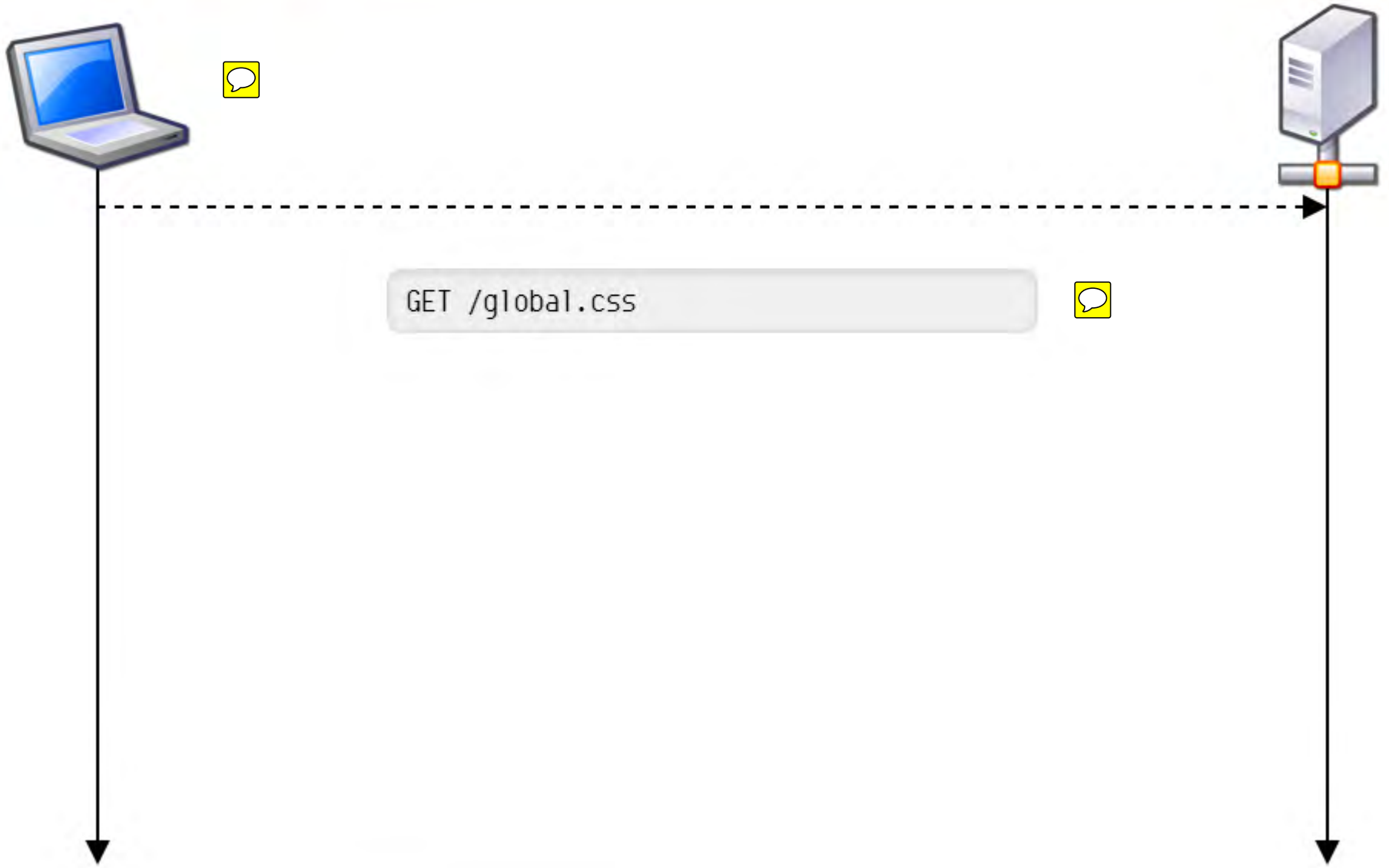
Cache Everything

Cache Everything

Cache Everything

```
200 OK

body {
    font: 16px Arial;
}
```

Cache Everything



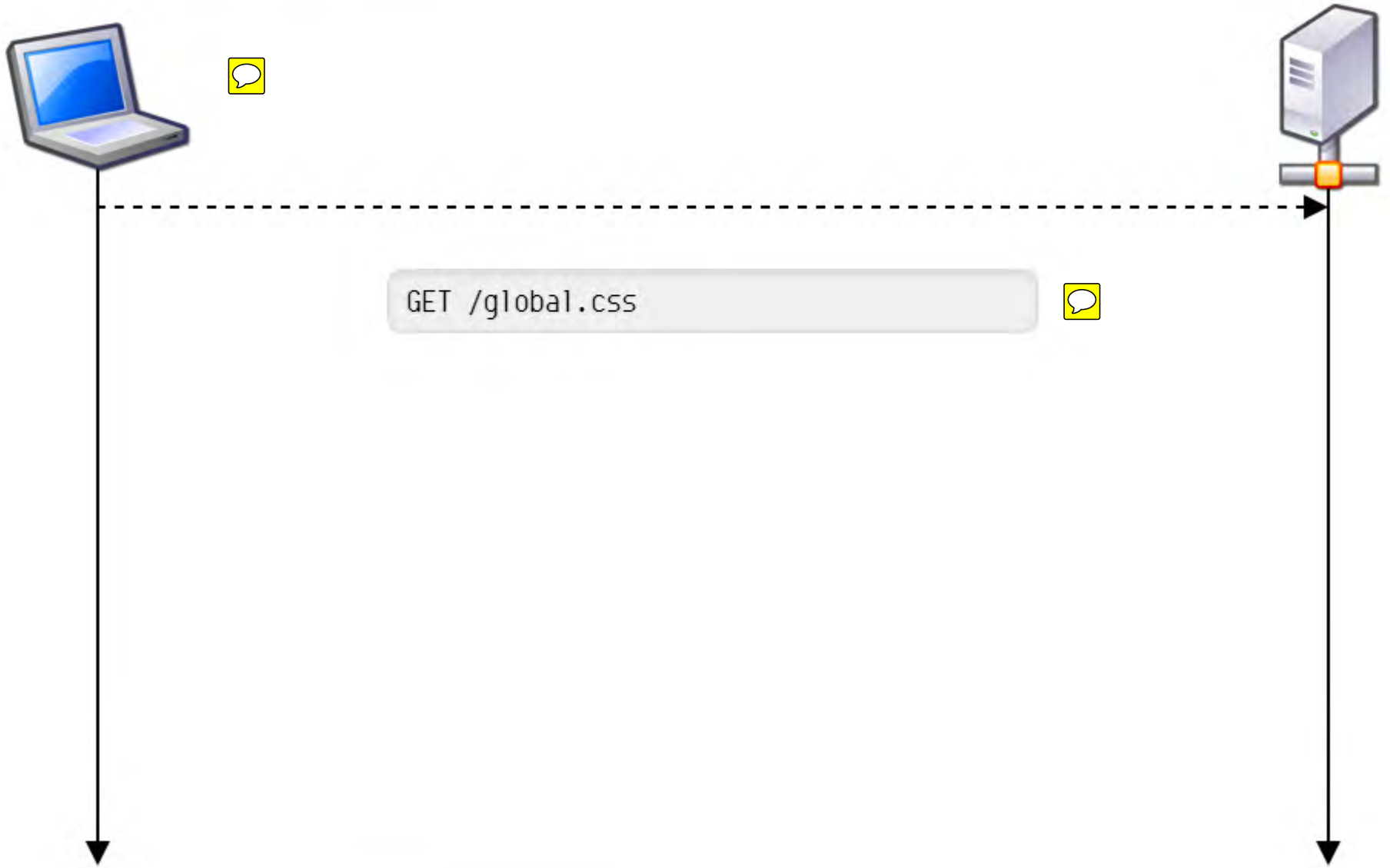
Cache Everything

Cache Everything

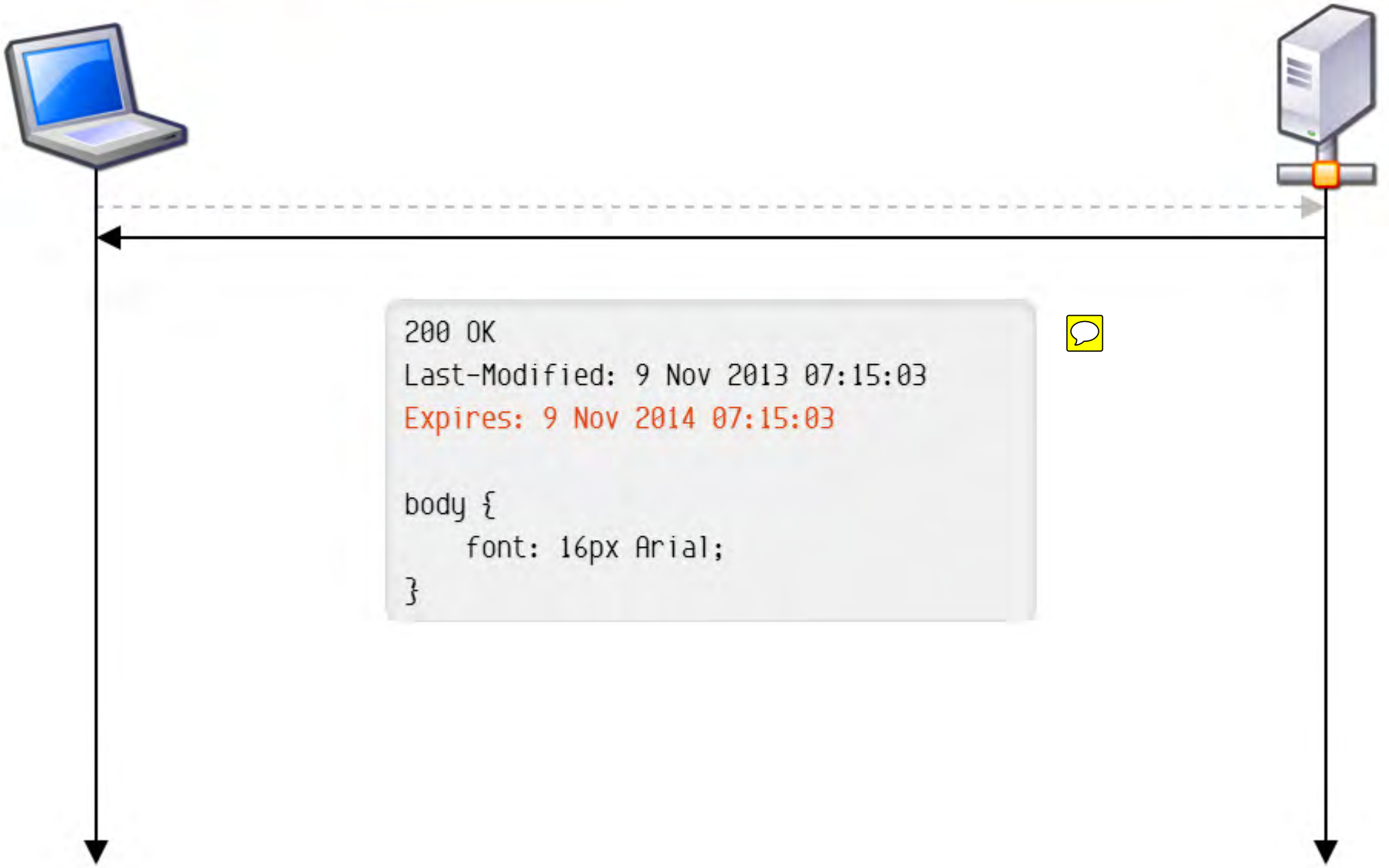
Cache Everything

Cache Everything

Cache Everything

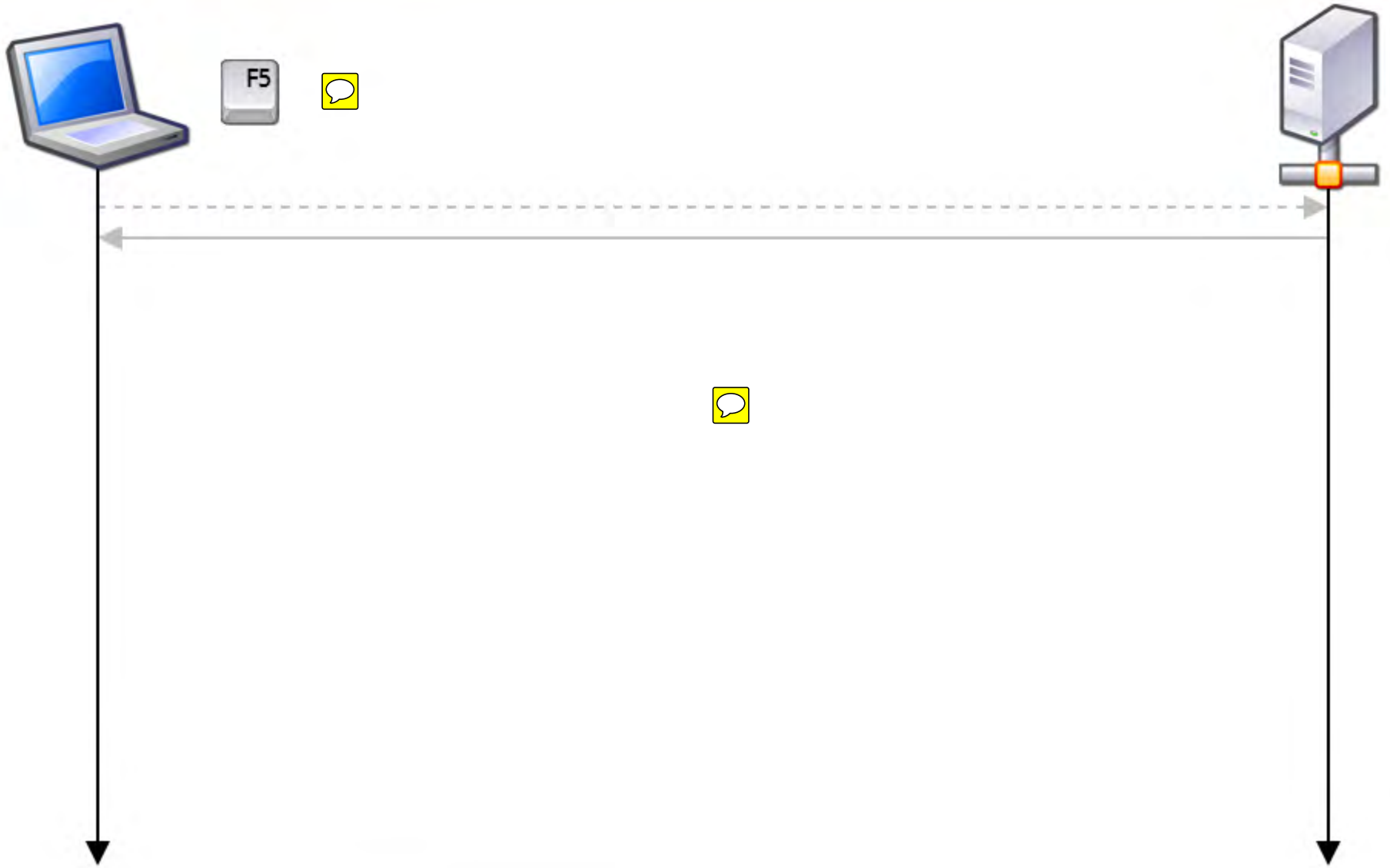


Cache Everything



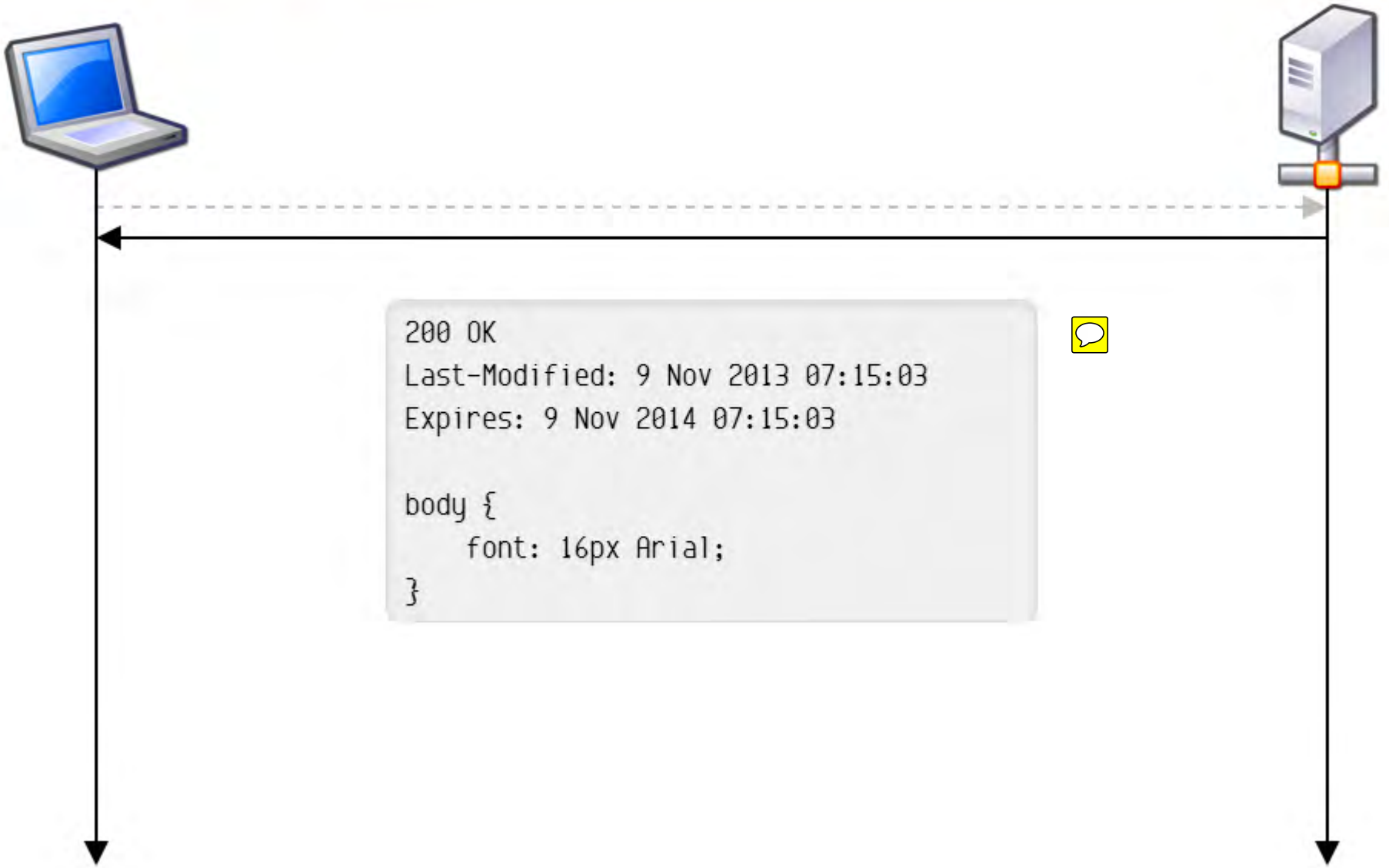
```
200 OK
Last-Modified: 9 Nov 2013 07:15:03
Expires: 9 Nov 2014 07:15:03

body {
    font: 16px Arial;
}
```

Cache Everything



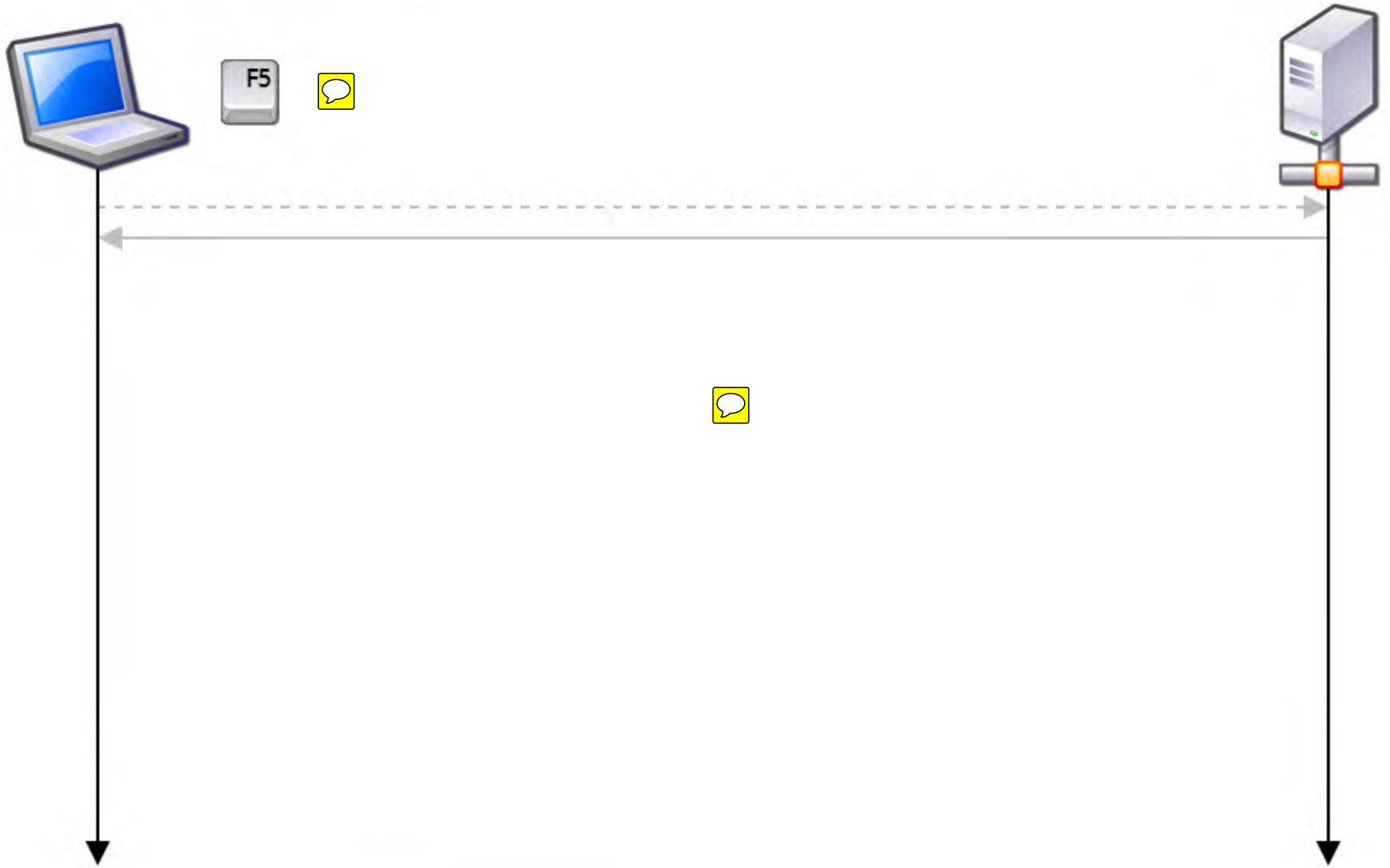
Cache Everything

Cache Everything

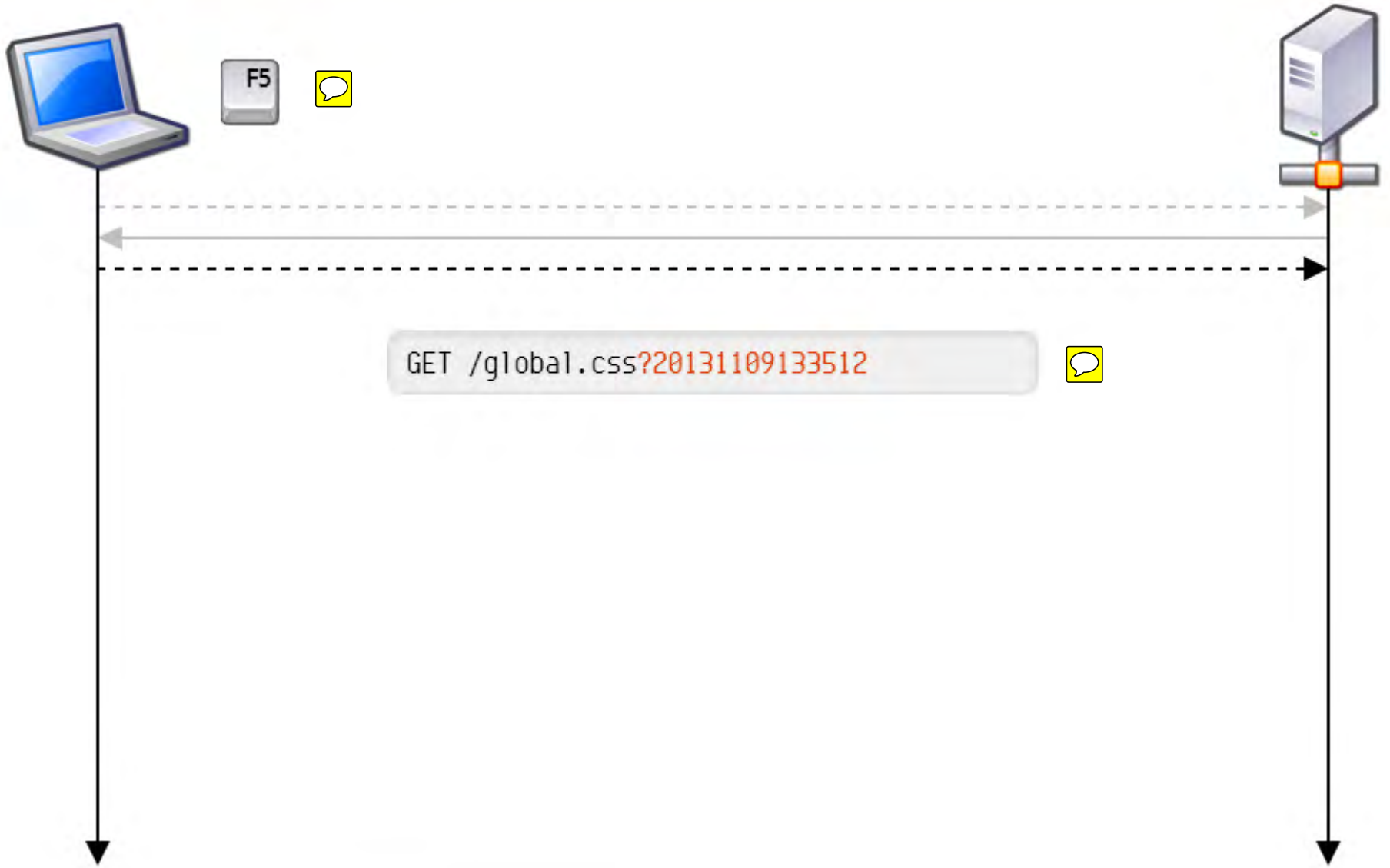


```
200 OK
Last-Modified: 9 Nov 2013 07:15:03
Expires: 9 Nov 2014 07:15:03

body {
    font: 16px Arial;
}
```

Cache Everything



Cache Everything



Cache Everything

```
200 OK
Last-Modified: 9 Nov 2013 13:35:12
Expires: 9 Nov 2014 13:35:12

body {
    font: 16px/24px Arial, sans-serif;
}
```

Cache Everything

Keys to aggressive caching:

- Version URLs somehow
- Include `Last-Modified` headers in your responses
- Handle `If-Modified-Since` request headers appropriately
- Include `Expires` headers in your responses, with far-future values

Combining vs. Caching

- These techniques are slightly at odds.
- If you combine too aggressively, you won't be able to reap the full benefits of caching.

```
<link href="A-B-C-J-and-K.css" />
```

→

```
<link href="A-B-and-C.css" />
<link href="J-and-K.css" />
```

```
<link href="A-B-C-X-Y-and-Z.css" />
```

→

```
<link href="A-B-and-C.css" />
<link href="X-Y-and-Z.css" />
```

Externalizing vs. Reducing Requests

- When you move styles and scripts into external files, you are naturally increasing the number of requests for a first-time visitor.
- In light of caching, this is OK, because you're moving cacheable resources out of inherently less-cacheable content.

In Closing

1. Speed is a feature.
2. Faster page loads lead to higher conversion rates.
3. Three key metrics to minimize: number of requests, size of responses, and rendering time.
4. Variety of techniques available to achieve these goals.

References

1. "Performance is a Feature", Jun 2011, Jeff Atwood
2. "The Need for Speed II", Apr 2001, Zona Research
3. "Retail Web Site Performance", Jun 2006, JupiterResearch
4. "eCommerce Web Site Performance Today", Aug 2009, Forrester Consulting
5. "Real User Monitoring", Feb 2012, Walmart Labs
6. "Shopzilla Site Redesign—We Get What We Measure", Jun 2009, Phil Dixon
7. "HTTP/1.1: Connections", Sep 2004, Fielding, et al.
8. "Total Transfer Size & Total Requests", Feb 2013, HTTP Archive.