

What HTTP/2 means to Java Developers?

JavaCRO16

May 2016

David Delabasse
@delabasse
Oracle

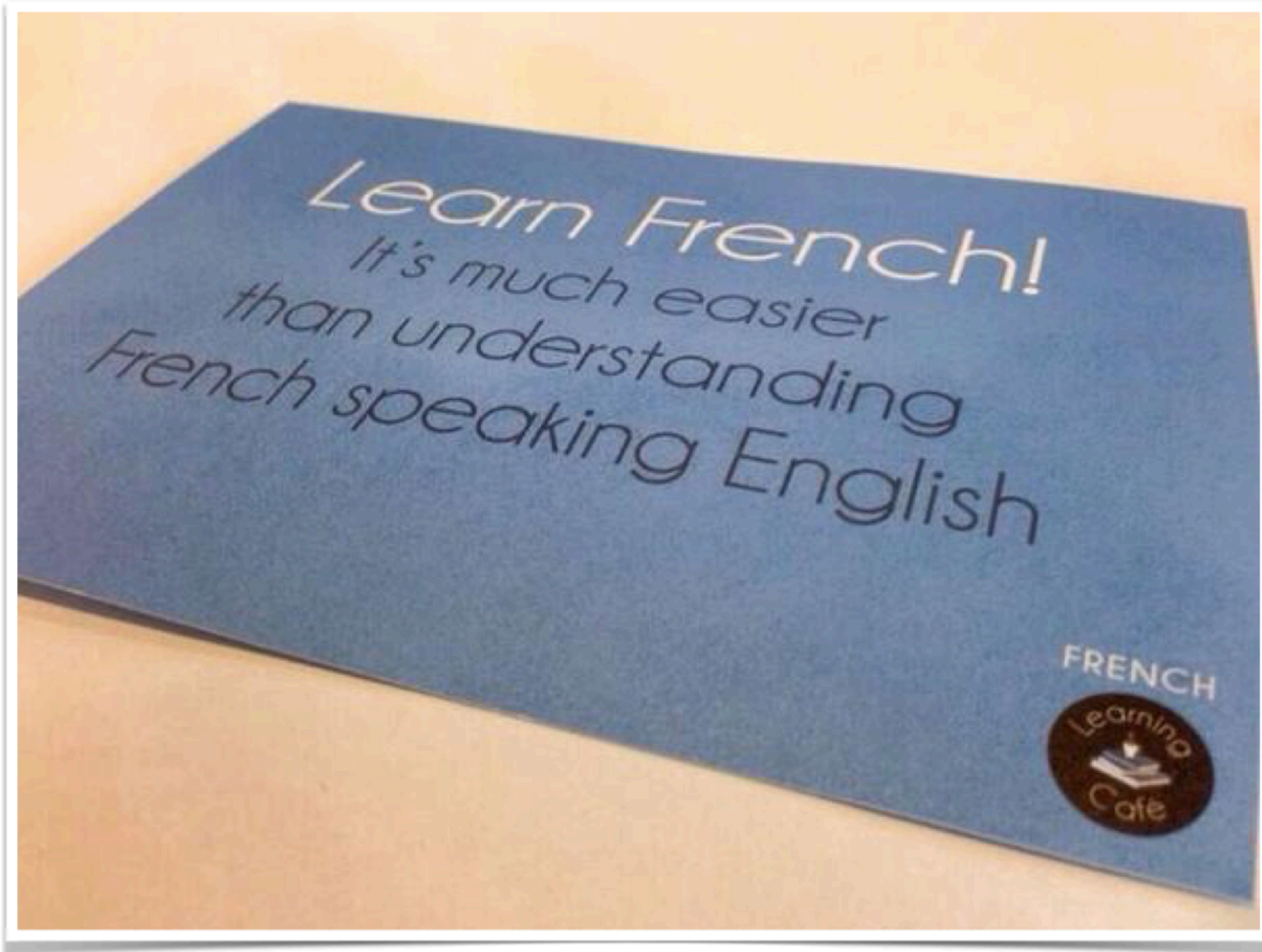


About me...



Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



Agenda

- Why HTTP/2?
- HTTP/2
- HTTP/2 and Java
- Summary

Agenda

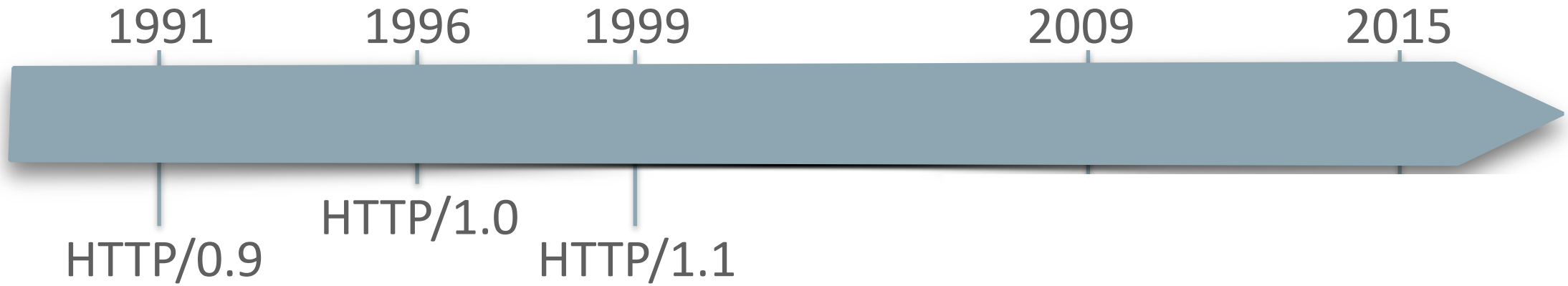
- ▶ Why HTTP/2?
- ▶ HTTP/2
- ▶ HTTP/2 and Java
- ▶ Summary

Data Never Sleeps 3.0



www.domo.com/blog/2015/08/data-never-sleeps-3-0/

HTTP 1.x



Google!

B E T A

Search the web using Google!

Google Search

I'm feeling lucky

[Why use Google!](#)

[Press about Google!](#)

[Help!](#)

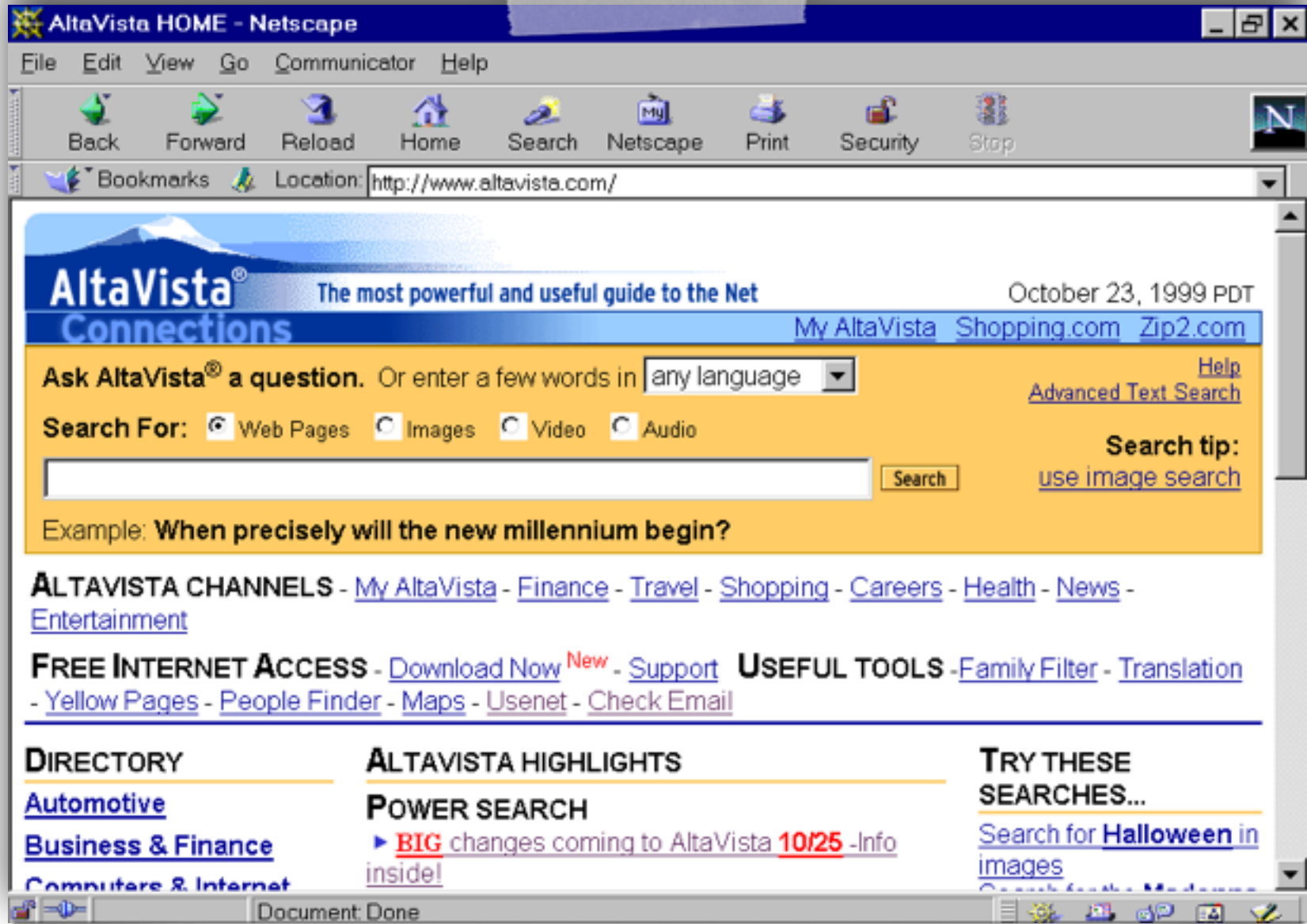
[Company Info](#)

[Jobs at Google](#)

[Google! Logos](#)

[Making Google! the Default](#)

Copyright ©1999 Google Inc.



SEARCH

All Products

GO

Search of the Day: [zaffron](#)



Hello! Shopping at Amazon.com is 100% secure--[guaranteed](#).
 Already a customer? [Sign in](#).

Vote in our [Millennium Poll](#)--you could win 300 CDs, books, and videos!

BROWSE

- [Books](#)
[Bestsellers](#), [Computers](#),
[Kids](#), [Business...](#)
- [Music](#)
[Top Sellers](#), [New Releases](#),
[Recommendation Center](#), [Soundtracks...](#)
- [Video](#)
[DVDs](#), [Top Sellers](#), [New Releases](#), [Kids & Family...](#)
- [Electronics](#)
[PalmPilots](#), [Sony Products](#), [Top Sellers](#),
[Computer Add-Ons...](#)
- [Toys & Games](#)
[Toys for Grownups](#),
[Games & Puzzles](#),
[Halloween Store](#), [Toy Quest Contest...](#)
- [Auctions](#)
[Tutorial](#), [Books](#),
[Collectibles](#), [Toys & Games...](#)
- [Free e-Cards](#)
[Daily Cards](#), [Hi, Love](#),
[Friendship...](#)
- [zShops](#)
[Buying](#), [Selling](#), [Our Guarantee](#), [Start Browsing...](#)
- [...And More](#)
[Women's Apparel](#),
[Garden & Patio](#),
[Software](#), [Sporting Goods](#)

In Books
Test Case



The postwar inventors of the Scholastic Aptitude Test hoped to produce a brainier brand of meritocracy in the United States. But as Nicholas Lemann reveals in *The Big Test*, the SAT hit a great many ideological potholes--and ended up creating yet another, pencil-pushing elite. Go to [Books](#)

*****In zShops*****
Many Merchants, Fabulous Finds

Earth's Biggest Selection just got bigger! You'll discover an amazing array of products from merchants large and small, including:

- A library of [literature](#)
 - Scads of [sports stuff](#)
 - A cornucopia of [costume jewelry](#)
- ...and so much more. Go to [zShops](#)



In Toys & Games
Halloween Headquarters

Dare to be a T-Rex--or a hippie or a pumpkin--with the costumes and accessories for kids and adults in our [Halloween Bootique](#). You'll also find decorations, party supplies, and more. Go to [Toys & Games](#)

In Auctions
Significant Signatures

On October 20, LiveBid will broadcast a live Profiles in History auction featuring hundreds of signed American treasures. [Register today](#) and you just might end up holding one of these:

- A card signed by Marilyn Monroe and Joe DiMaggio
- A letter written aboard the Titanic

Guarantee holiday gift bliss with [Amazon.com Wish Lists!](#)

Amazon.com 100 Hot Books

Updated Hourly

1. [The Carbohydrate Addict's Lifespan Program: A Personalized Plan for Becoming Slim, Fit and Healthy in Your 40s, 50s, 60s and Beyond](#)
 by Rachael F. Heller, Richard Ferdinand Heller
2. [The Carbohydrate Addict's Diet: The Lifelong Solution to Yo-Yo Dieting](#)
 by Rachael F. Heller, Richard Ferdinand Heller
3. [The Carbohydrate Addict's Gram Counter](#)
 by Richard Ferdinand Heller, Rachael F. Heller

[More Hot Books](#)

[Amazon.com Auctions](#)

Today in Music
[Free mp3 downloads](#)

kindle

Emportez votre bibliothèque avec vous



kindle
Dès 69,99€



kindle paperwhite
Dès 129,99€



kindle voyage
Dès 189,99€

Notre sélection de smartphones [Voir plus](#)



Notre sélection d'enceintes portables [Voir plus](#)




amazon famille
Économisez 15%
dès 60€ d'achats
avec Amazon Famille



Ventes Flash
Quantités limitées

Découvrez le style industriel [Voir plus](#)



Nos meilleures ventes Compacts et bridges [Voir plus](#)



Amazon utilise des cookies. [En savoir plus.](#)

amazon Premium

Essayez gratuitement Amazon Premium pendant 30 jours

Livraison en 1 jour ouvré sur des millions d'articles

*base nationale



[Votre avis sur l'annonce](#)



Jusqu'à -60% sur la collection sports d'hiver



Barres de son
Toute notre sélection

Meilleures ventes Caméras de surveillance

- 1 
- 2 
- 3 
- 4 
- 5 

[Voir plus](#)







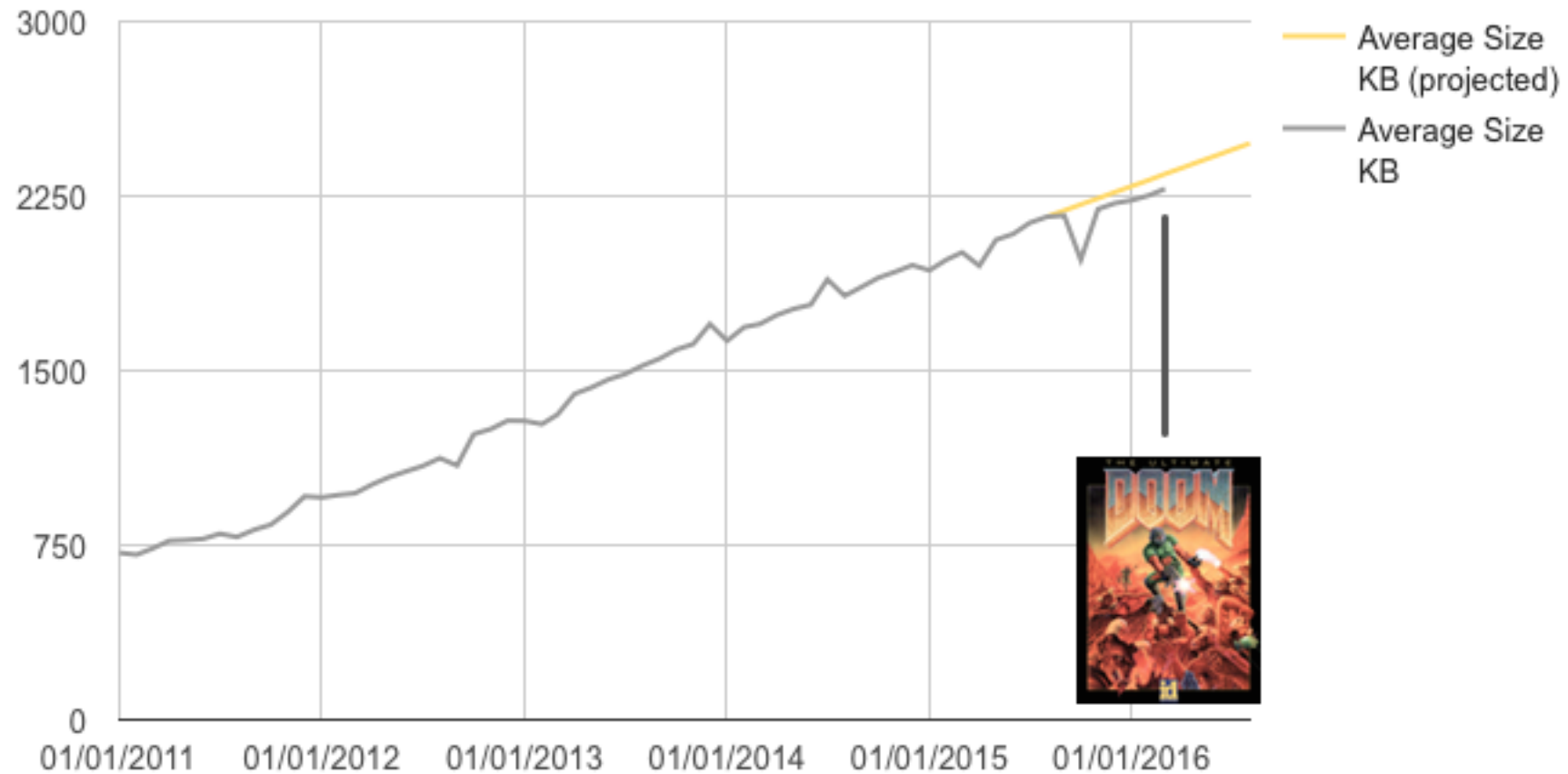
WEB TOUCH



JAVA
A SUN MICROSYSTEMS BRAND

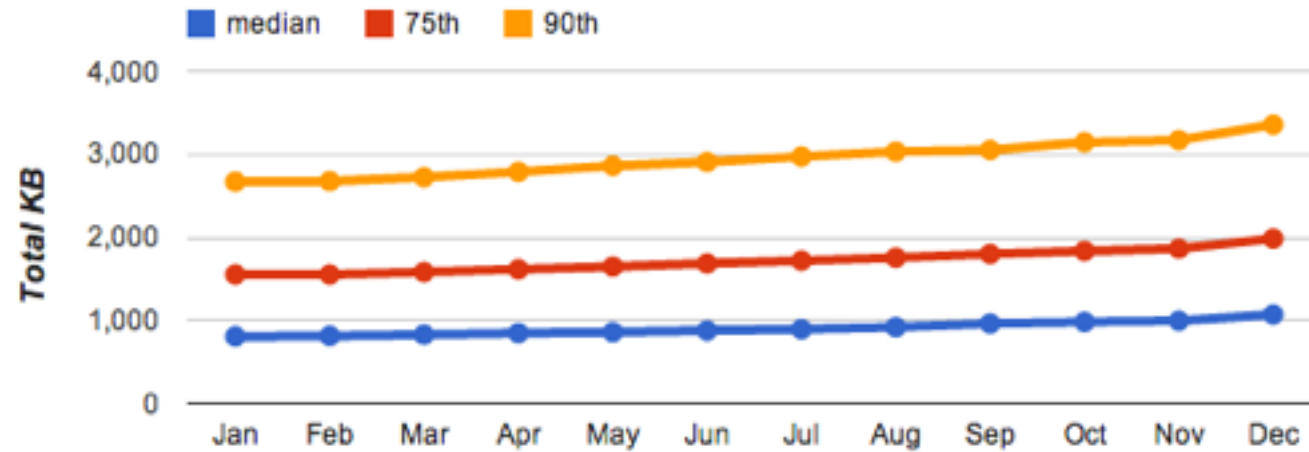


The Web is Doom



<https://mobiforge.com/research-analysis/the-web-is-doom>

Transfer size

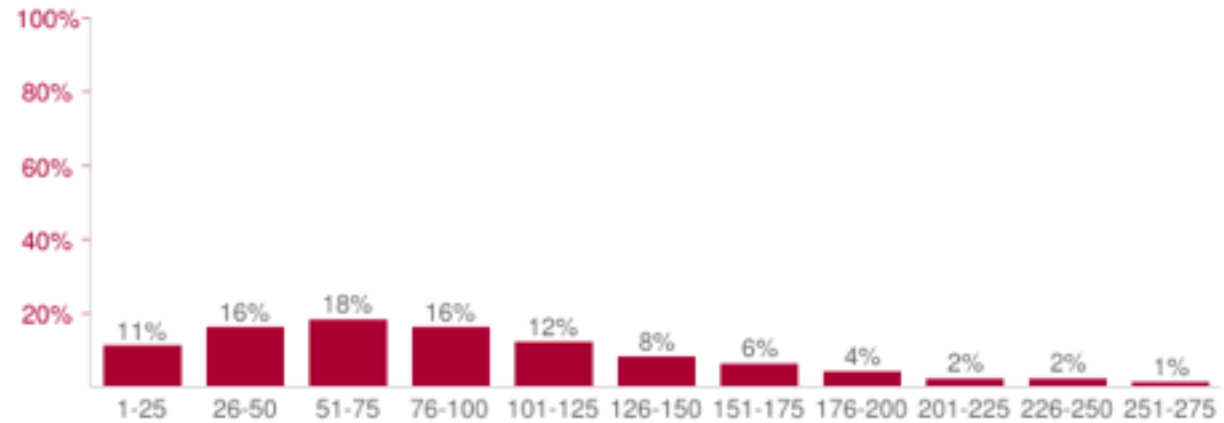


	50th percentile	75th percentile	90th percentile
HTML	13 KB	26 KB	54 KB
Images	528 KB	1213 KB	2384 KB
JavaScript	207 KB	385 KB	587 KB
CSS	24 KB	53 KB	108 KB
Other	282 KB	308 KB	353 KB
Total	1054 KB	1985 KB	3486 KB

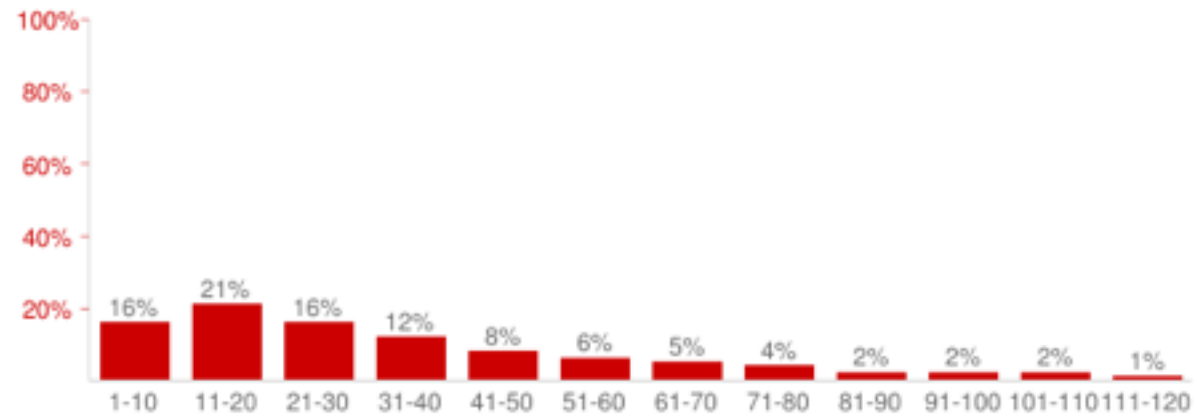
<https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency>

Requests vs. Connections

Total Requests per Page



Connections per Page



<http://httparchive.org/>

Latency Vs. Conversion Rate

Time is Money!

- If a page takes longer than 4 sec to load, 1/4 people abandons that page ⁽¹⁾
- 57% of consumers will abandon a page that takes longer than 3 sec to load ⁽²⁾
- Page load slowdown of 1 sec could cost Amazon \$1.6 billion in sales a year ⁽¹⁾
- Slowing search results by 0.4 sec, Google could lose 8 million searches per day ⁽¹⁾

(1) <http://www.fastcompany.com/1825005/how-one-second-could-cost-amazon-16-billion-sales>

(2) RadView Spring 2015 State of the Union: Ecommerce Page Speed & Web Performance

HTTP 1.1

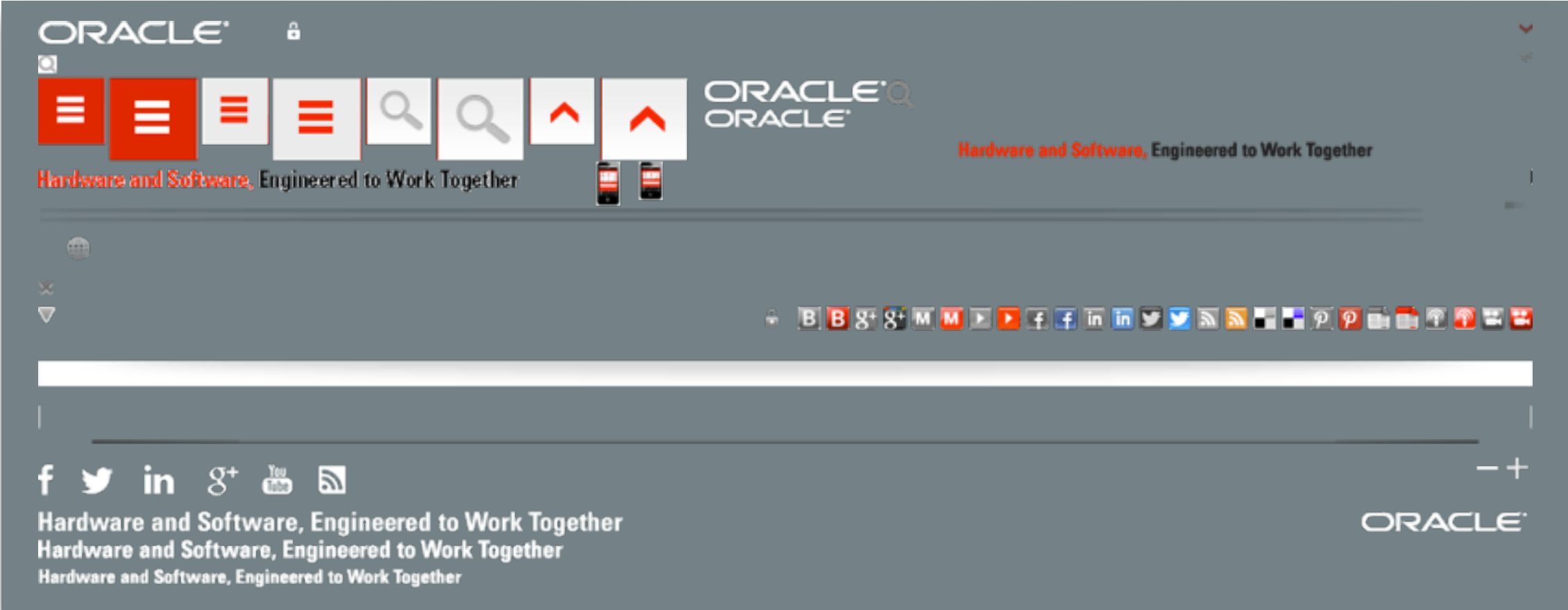
File Concatenation and Image Sprites

- Modern web page consists of +90 resources fetched from 15 distinct hosts (*)
- TCP Efficiency Improves with Larger Files
- Shoving more than one logical file into one physical file

(*) <http://httparchive.org>

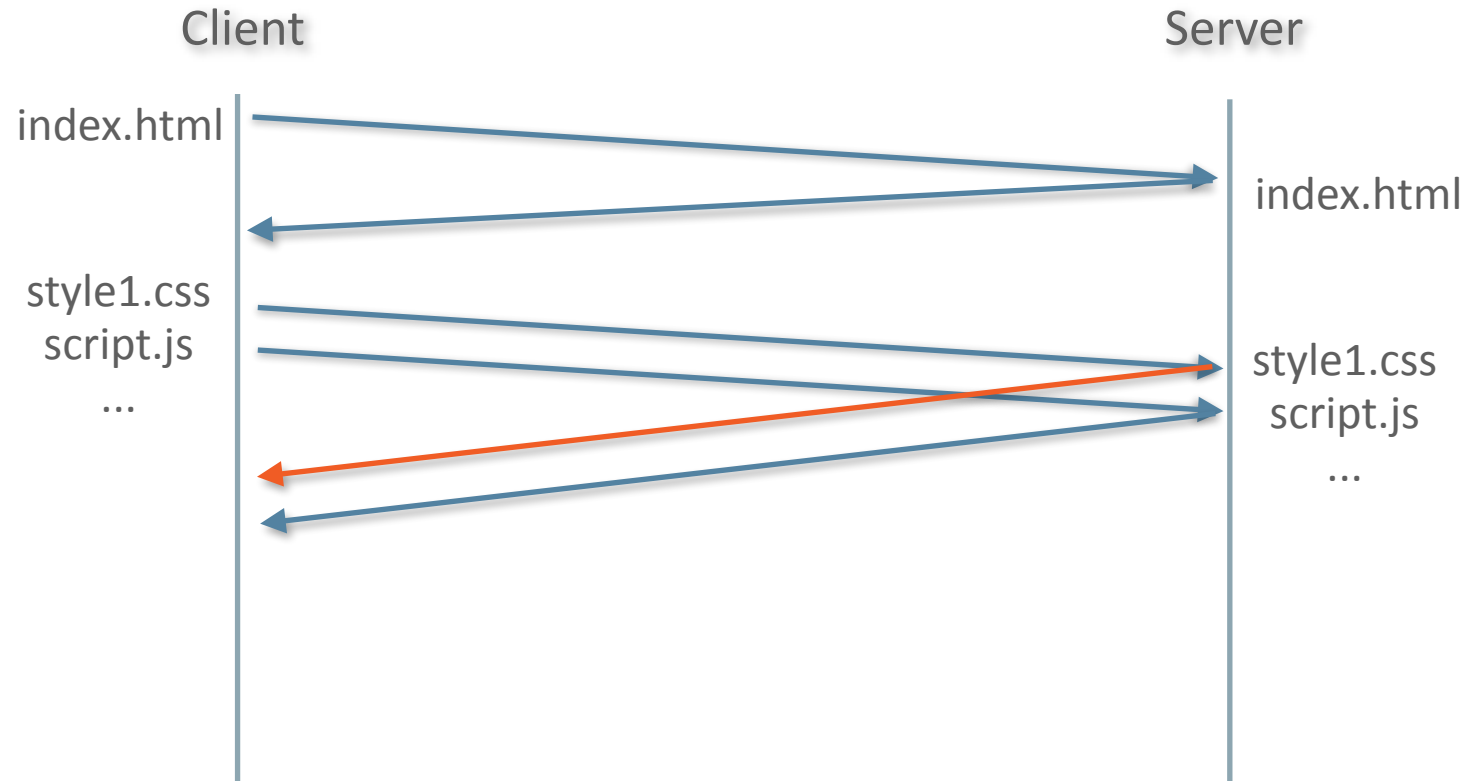
HTTP 1.1

File Concatenation and Image Sprites



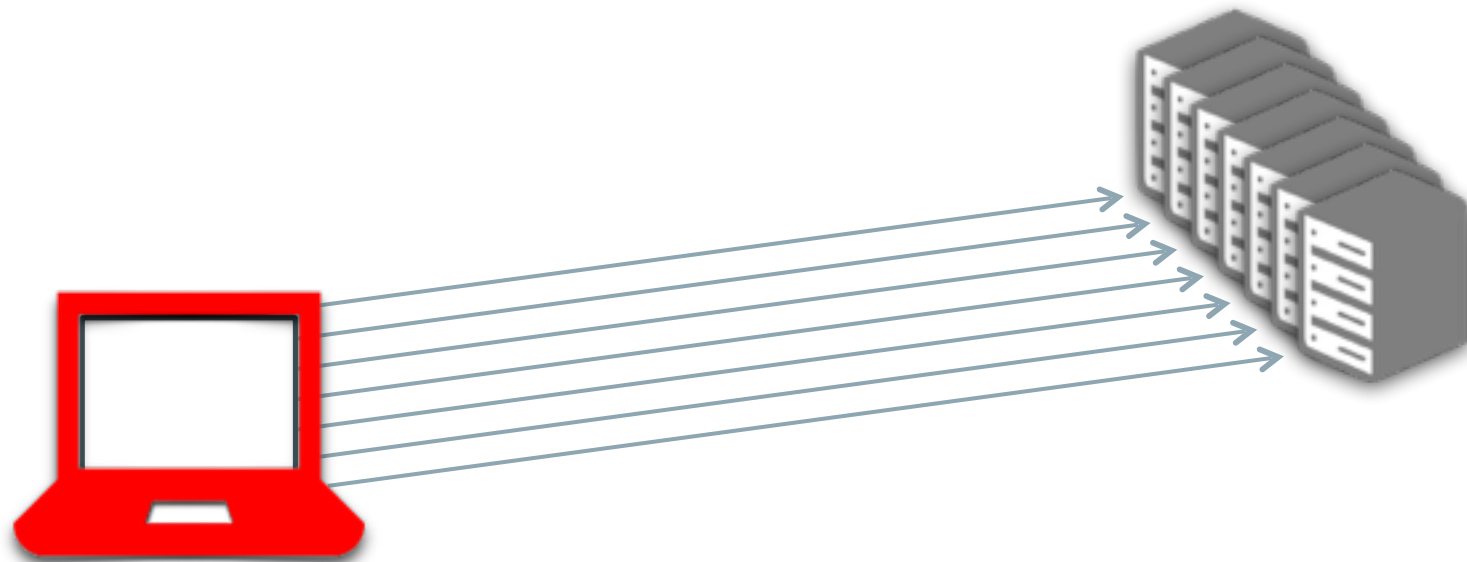
HTTP 1.1

Head-of-Line blocking



HTTP 1.1

Domain Sharding



HTTP 1.1

Asset Inlining

...

```

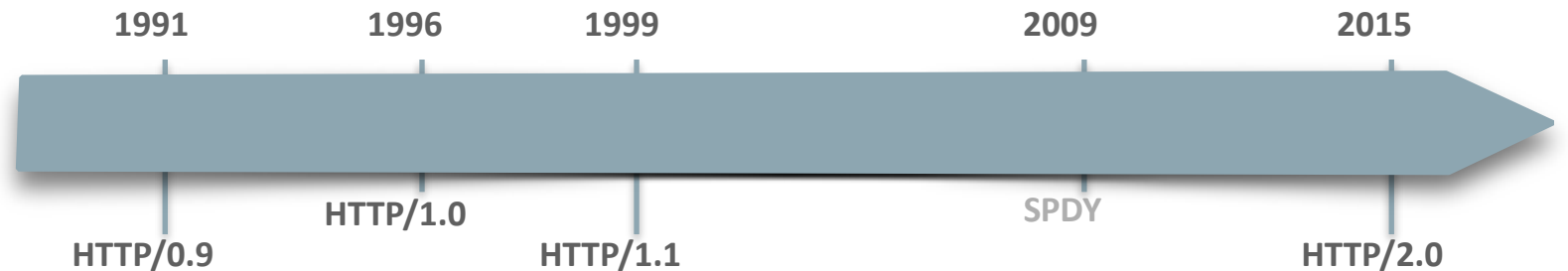
```

...

HTTP 1.1 circa 1999

Problems Vs Solutions

- HTTP uses TCP poorly
 - HTTP: short and bursty flows Vs. TCP: optimized for long-lived flows
- Solutions
 - Sprites
 - Domain sharding
 - Assets Inlining
 - File concatenations
 - ...



Agenda

- ▶ Why HTTP/2?
- ▶ HTTP/2
- ▶ HTTP/2 and Java SE
- ▶ HTTP/2 and Java EE
- ▶ Summary

HTTP/2

- Jan 2015 Enabled by default in FireFox (35) and Chrome (40)
- Feb 2015 IESG approved HTTP/2
- May 2015 HTTP/2 in 10% of all HTTP responses (FireFox)
HTTP/2 used in 18% of global traffic (Google)
RFC 7540 “Hypertext Transfer Protocol Version 2”
RFC 7541 “HPACK: Header Compression for HTTP/2”

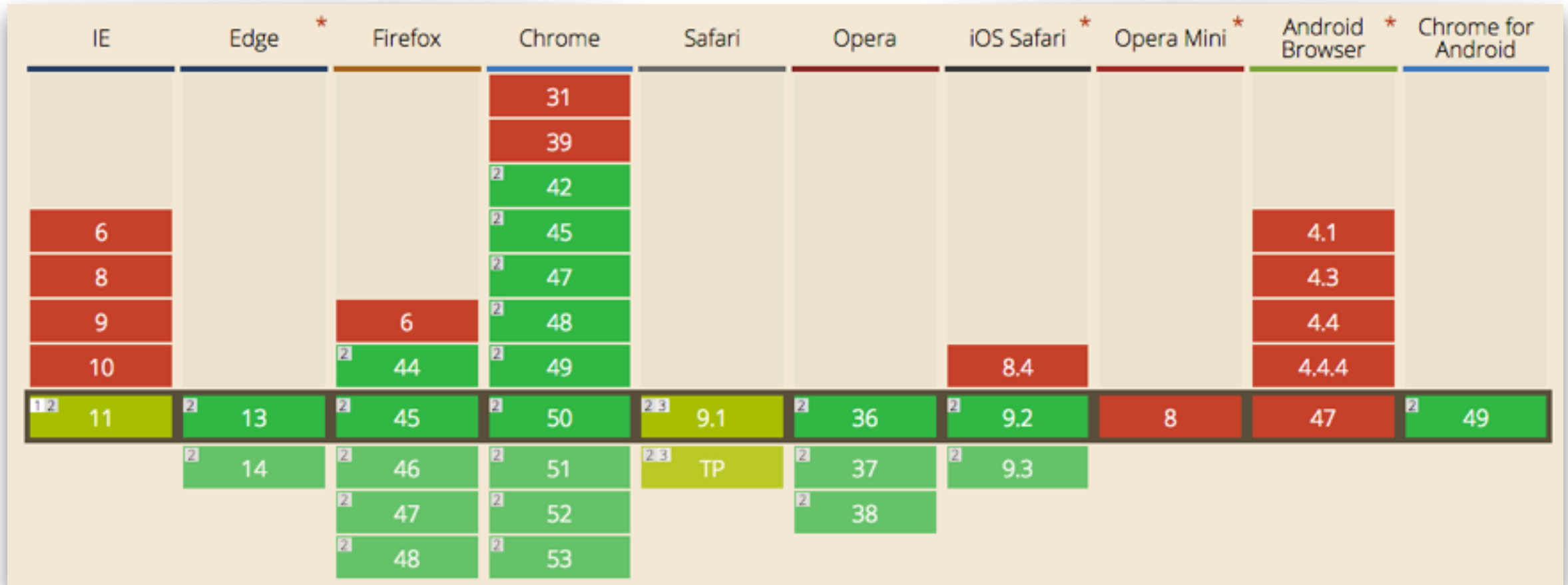
Adoption

Last year we announced our intent to end support for the experimental protocol SPDY in favor of the standardized version, HTTP/2. HTTP/2 is the next-generation protocol for transferring information on the web, improving upon HTTP/1.1 with more features leading to better performance.

Since then we've seen huge adoption of HTTP/2 from both web servers and browsers, with most now supporting HTTP/2. **Over 25% of resources in Chrome are currently served over HTTP/2**, compared to less than 5% over SPDY. Based on such strong adoption, starting on May 15th — the anniversary of the HTTP/2 RFC — Chrome will no longer support SPDY. ...

<http://blog.chromium.org/2016/02/transitioning-from-spdy-to-http2.html>

Adoption



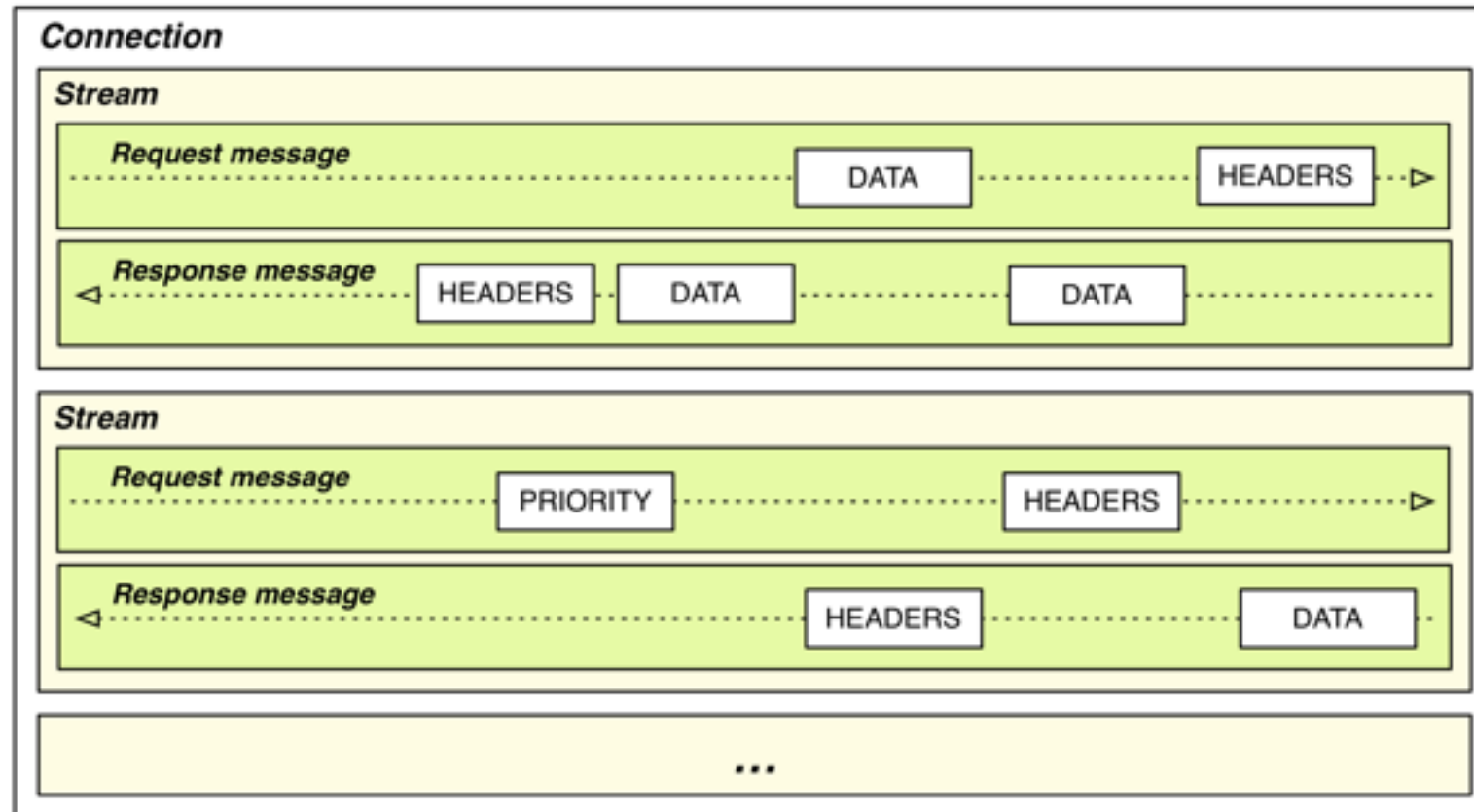
<http://caniuse.com/http2>

HTTP/2 Features

- Binary Framing over single TCP connection
- Request/Response multiplexing
- Stream Prioritization
- Server Push
- Upgrade from HTTP 1.1
- Header Compression
- Preserve HTTP semantic
- Flow Control

HTTP/2

Connections, Streams, Messages, Frames



Binary Frames

- Frames
 - HEADERS, DATA, PRIORITY, RST_STREAM, SETTINGS, PUSH_PROMISE, PING, GOAWAY, WINDOW_UPDATE, CONTINUATION
 - Prioritisation, Flow Control, Server Push, ...
- Single TCP Connection

Binary Frames

Example

```
GET /index.html HTTP/1.1  
Host: example.com  
Accept: text/html
```

Binary Frames

Example

```
HTTP/1.1 200 OK  
Content-Length: 84  
Content-Type: text/html
```

```
<!doctype html>  
<head>  
<meta charset=utf-8>  
</head>  
<body>Hello World</body>  
</html>
```

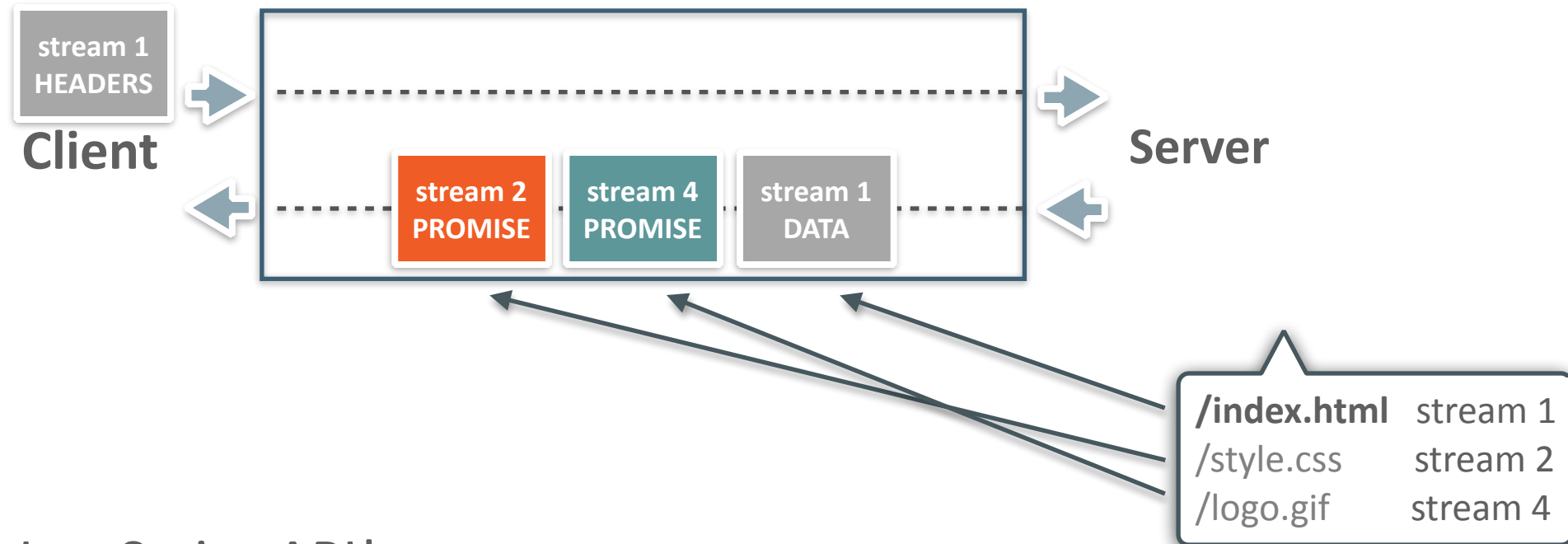
Multiplexing



Stream Prioritization

- Stream Dependency in HEADERS Frame
- PRIORITY frame type
- An additional 40 bytes
 - Stream id (31)
 - Weight (8): [1, 256]
 - Exclusive bit (1)
- Only an advice

Server Push



- No JavaScript API!

Header Compression

HPack

Request #1

:method	GET
:scheme	https
:host	example.com
:path	/resource
accept	image/jpeg
user-agent	Mozilla/5.0 ...

Upgrade from HTTP 1.1

- HTTP
 - Port 80
 - HTTP Upgrade to “h2c” (101 Switching Protocol)
- HTTPS (*)
 - Application Layer Protocol Negotiation (ALPN)
 - ~~Next Protocol Negotiation (NPN)~~

(*) TLS is not mandatory

Demo?



Agenda

- Why HTTP/2?
- HTTP/2
- HTTP/2 and Java
- Summary

HTTP/2 Java implementations

<i>Implementation</i>	<i>Role(s)</i>	<i>Negotitation(s)</i>
<i>Apache Tomcat 8.5+</i>	<i>Server</i>	<i>ALPN, Upgrade</i>
<i>Jetty</i>	<i>Client, Intermediary, Server</i>	<i>ALPN, Upgrade, Direct</i>
<i>Netty</i>	<i>Client, Server</i>	<i>ALPN, NPN, Upgrade, Direct</i>
<i>OkHttp</i>	<i>Client, Mock Server</i>	<i>ALPN, NPN</i>
<i>Undertow</i>	<i>Server, Intermediary</i>	<i>ALPN, Upgrade</i>

<https://github.com/http2/http2-spec/wiki/Implementations>

Java 9 Support for HTTP/2

JEP 110

- Replace HttpURLConnection API
- Supports HTTP 1.1 and HTTP/2
- Easy to use API
- Covers the most common use cases
- Synchronous & Asynchronous
- ...
- See <http://openjdk.java.net/jeps/110>

Java 9

Client - Request - Response

Java 9

Reusable HTTP client container

```
HttpClient client = HttpClient.create()  
    .version(HTTP_2)  
    .sslContext(...)  
    .sslParameters(...)  
    .authenticator(...)  
    .cookieManager(...)  
    .followRedirects(...)  
    .executorService(...)  
    .priority(...)  
    .proxy(...)  
    ...  
    .build();
```

Java 9

HttpRequest

```
// using the default HttpClient
HttpResponse response = HttpRequest
    .create(new URI("http://..."))
    .headers("Foo", "aaa", "Bar", "bbb")
    .GET()
    .response();

int statusCode = response.statusCode();
String responseBody = response.body(asString());
```


Java 9

HttpRequest

```
HttpResponse response = HttpRequest  
    .create(new URI("http://www.foo.com"))  
    .body(fromString("par1=foo,par2=bar"))  
    .POST()  
    .response();
```

Java 9

HttpResponse

- Blocking
 - `HttpResponse response()`
- Asynchronous
 - `CompletableFuture<HttpResponse> responseAsync()`
- Asynchronous, expect multiple responses
 - `CompletableFuture<U> multiResponseAsync(MultiProcessor<U> rspproc)`

Java 9

BodyProcessor

- Handle responses bodies
- Use built-in `HttpResponse` body processors
 - `HttpResponse.asByteArray()`
 - `HttpResponse.asString()`
 - `HttpResponse.asFile(java.nio.file.Path)`
 - `HttpResponse.ignoreBody()`
- Or supply your own `BodyProcessor<T>` to `HttpResponse.body()` or `HttpResponse.bodyAsync()`

Java 9

MultiProcessor

- `HttpResponse.MultiProcessor<T>`
- Static interface
 - `onStart`
 - `onComplete` -> returns a BiFunction that is called for each Push Promise

Java 9

MultiProcessor

- `HttpResponse.multiFile()`
 - Returns a File MultiProcessor implementation
 - Writes the response bodies to files and returns `Map<URI,Path>`

```
CompletableFuture<Map<URI,Path>> cf =  
    HttpRequest.create(new URI("https://www.foo.com/"))  
        .version(Version.HTTP2)  
        .GET()  
        .sendAsyncMulti( HttpResponse.multiFile("/tmp") );
```

```
Map<URI,Path> results = cf.join();
```

Java 9

```
// HttpRequest builder from the default HttpClient  
HttpResponse response = HttpRequest.create(new URI("http://www.abc.be"))  
    .send(HttpRequest.noBody());  
  
String responseBody = response.body(HttpResponse.asString());
```

Java 9

```
HttpClient cl = HttpClient.create()  
    .proxy("http",proxAd).proxy("https", proxAd)  
    .sslContext(ctx)  
    .build();
```

Java 9 HttpClient API

- High Level API
- **HttpClient** - Reusable client container
 - Builder
- **HttpRequest** - One HTTP request which can be sent to a server
 - Builder
 - **HttpClient.request(...)**
- **HttpResponse** - Represents a response to a HttpRequest
 - **HttpRequest.response(...)**
 - BodyProcessor, MutliProcessor

Servlet 4

HTTP/2 features

- Request/Response multiplexing
- Binary Framing
- Stream Prioritization
- Server Push
- Header Compression
- Upgrade from HTTP 1.1
 - ALPN
 - 101 Switching Protocols

Servlet 4

Features to be exposed in the Servlet API

- Request/Response Multiplexing
- Binary Framing
- ~~Stream Prioritization~~
- Server Push
- Header Compression
- Upgrade from HTTP 1.1
 - ALPN
 - 101 Switching Protocols

Server Push

```
PushBuilder builder = baseRequest.getPushBuilder();  
  
builder.addHeader("X-Pusher", ...);  
  
builder.path(aResource)  
    .etag(associated._etag)  
    .lastModified(associated._lastModified)  
    .push();
```

Agenda

- Why HTTP/2?
- HTTP/2
- HTTP/2 and Java
- Summary

HTTP/2 Features

- Binary Framing over single TCP connection
- Request/Response multiplexing
- Stream Prioritization
- Server Push
- Upgrade from HTTP 1.1
- Header Compression
- Preserve HTTP semantic
- Flow Control

HTTP/2

Hypertext Transfer Protocol version 2 & HPACK

- Address the Limitations of HTTP 1.x
 - Improve resources utilization, performance, reduce latency
- *“Compatible”* with HTTP 1.1
 - Retain HTTP 1.1 semantics
 - Define interaction with HTTP 1.1
 - Undo your HTTP 1.1 tricks!!
- *“TLS not mandatory”*

HTTP/2 and Java

Plans

- JEP 110 brings HTTP/2 to Java SE
 - High level API
- Servlet 4.0 brings HTTP/2 to Java EE
 - Expose key features to the API
 - Server Push
 - HTTP 1.1 upgrade

Hvala ti

Resources (and credits)

- <http://http2.github.io>
- <http://chimera.labs.oreilly.com/books/12300000000545/ch12.html>
 - “High Performance Web Sites” ebook
- <http://openjdk.java.net/jeps/110>
- <http://download.java.net/java/jdk9/docs/api/java.httpclient-summary.html>
- <https://java.net/projects/servlet-spec/>

CREATE THE FUTURE

