HTTP/2 for dummies

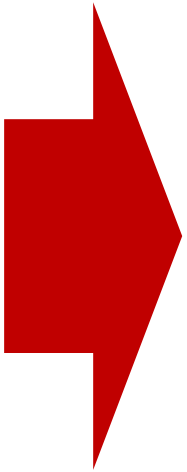
Patrizio Munzi

Senior Software Engineer
in Hotels.com Rome Tech Hub
@patrizio_munzi

Hotels.com

HTTP/2 ≧ HTTP/1

- https://blog.httpwatch.com/2015/01/16/a-simple-performance-comparison-of-https-spdy-and-http2/
- http://ieeexplore.ieee.org/document/7179400/
- https://www.tunetheweb.com/blog/http-versus-https-versus-http2/
- Just google for others ☺



100ms in delay results in 1% sales loss.
(potential hundreds of milions in lost revenues)

400ms delay results in 5-9% drop in full page traffic

500ms delay drops search traffic by 20%.

- https://www.slideshare.net/pob1970/mobile-first-lukew/41-100ms_delay_results_in_1
- https://news.ycombinator.com/item?id=273900
- http://glinden.blogspot.it/2006/11/marissa-mayer-at-web-20.html

Agenda

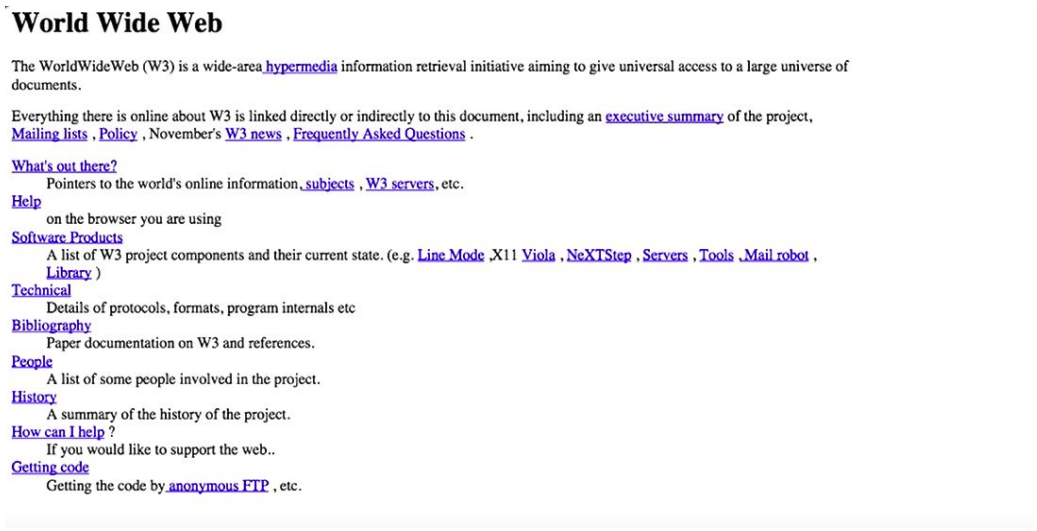
1. A bit of HTTP history
2. HTTP/2 new features
3. HTTP/2 in practice
4. HTTP/2 adoption recipes
5. Q/A

History HTTP

199 199 199 201 201

HTTP/0.9

GET
POST
HEAD



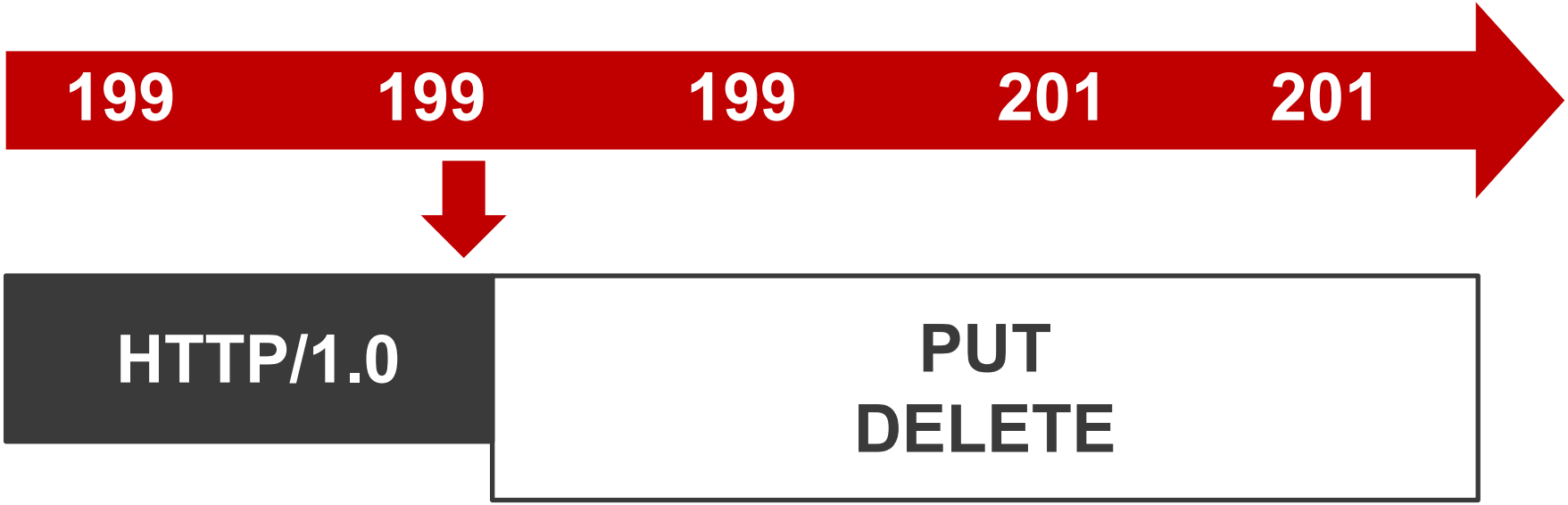
World Wide Web

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy , November's W3 news , Frequently Asked Questions .

What's out there?
Pointers to the world's online information, subjects , W3 servers, etc.
Help
on the browser you are using
Software Products
A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep , Servers , Tools , Mail robot , Library)
Technical
Details of protocols, formats, program internals etc
Bibliography
Paper documentation on W3 and references.
People
A list of some people involved in the project.
History
A summary of the history of the project.
How can I help ?
If you would like to support the web..
Getting code
Getting the code by anonymous FTP , etc.

History HTTP



History HTTP

199 199 199 201 201

HTTP/1.1

OPTIONS
HOST HEADER
KEEP ALIVE

HTTP/1 best practices

HTTP/1 ~~best practices~~ workarounds

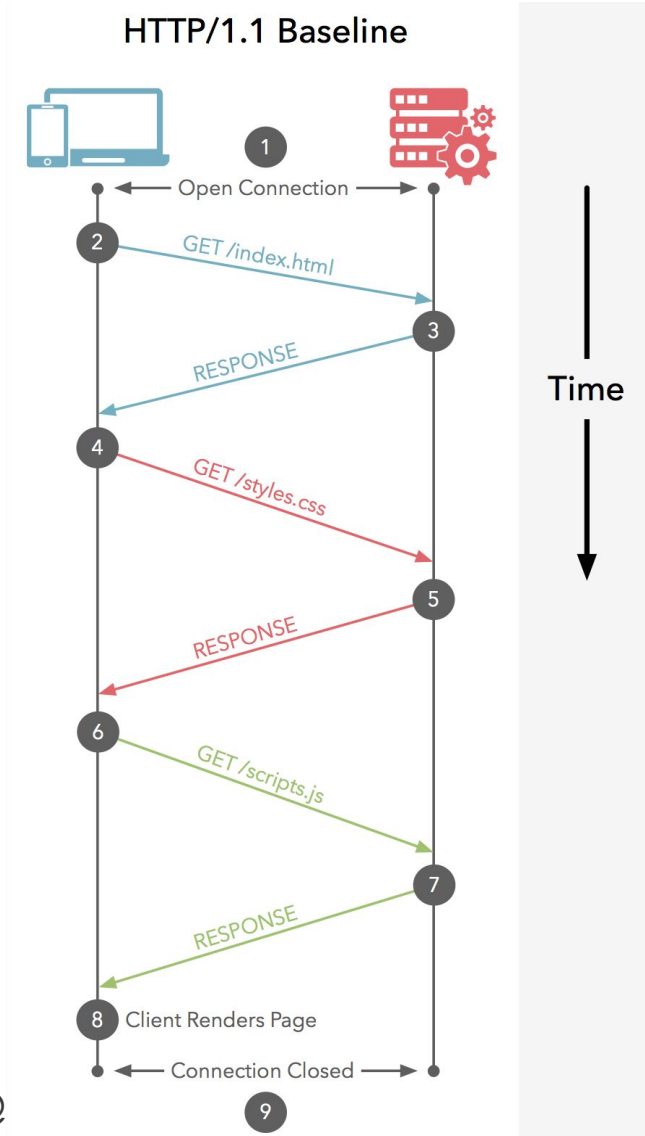
HTTP/1 ~~best practices~~ ~~workarounds~~ hacks

Concatenation
Inlining
Spriting
Domain Sharding

HTTP/1 worst problem

HOL Blocking

One resource at a time



@

HTTP/1 waterfall

Demo concept inspired by Golang's Gophertiles

https://http2.akamai.com/demo

History HTTP

SPDY

BINARY
HEADER COMPRESSION
MULTIPLEXED STREAMS
SERVER PUSH
PROTOCOL NEGOTIATION

History HTTP

HTTP/2

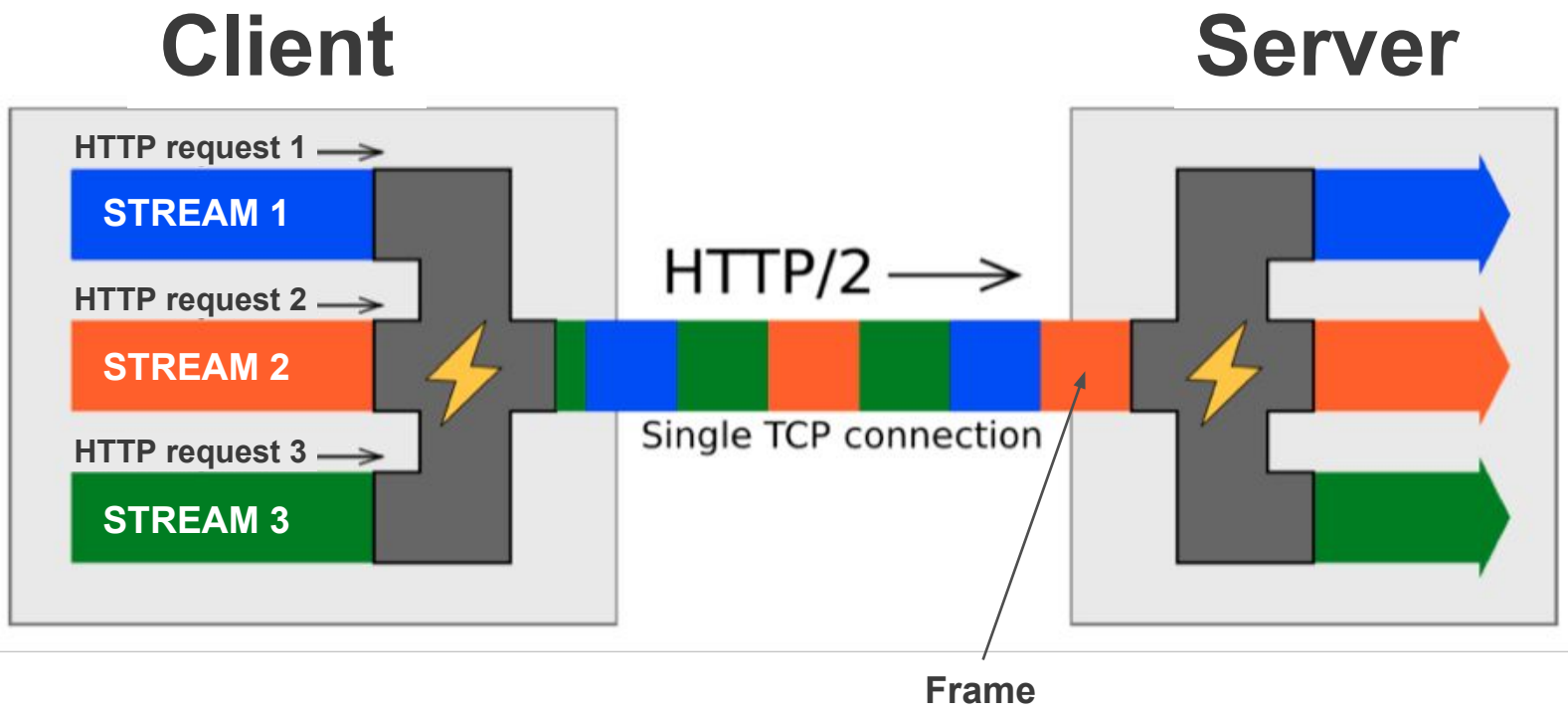
BINARY
HPACK
MULTIPLEXED STREAMS
SERVER PUSH
PROTOCOL NEGOTIATION

Why HTTP/2 is faster?

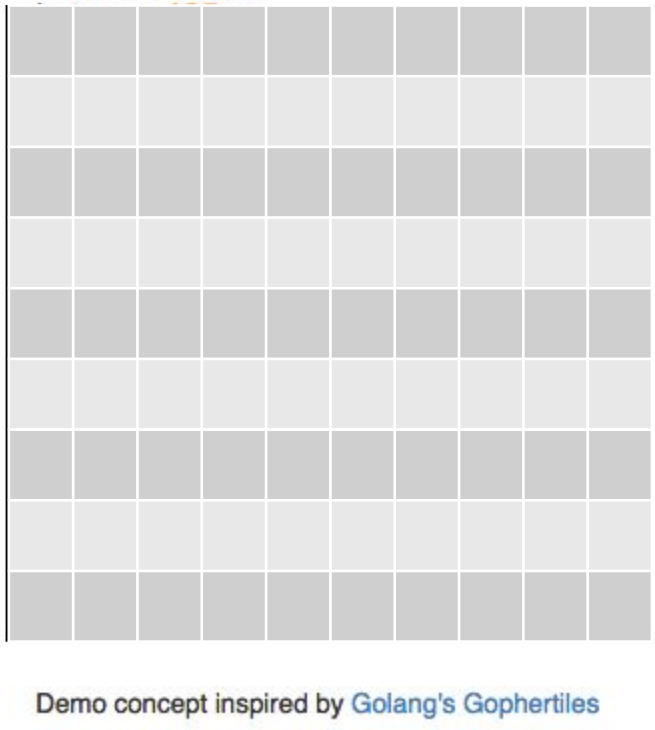
1 - MULTIPLEXING

Why HTTP/2 is faster?

1 - MULTIPLEXING

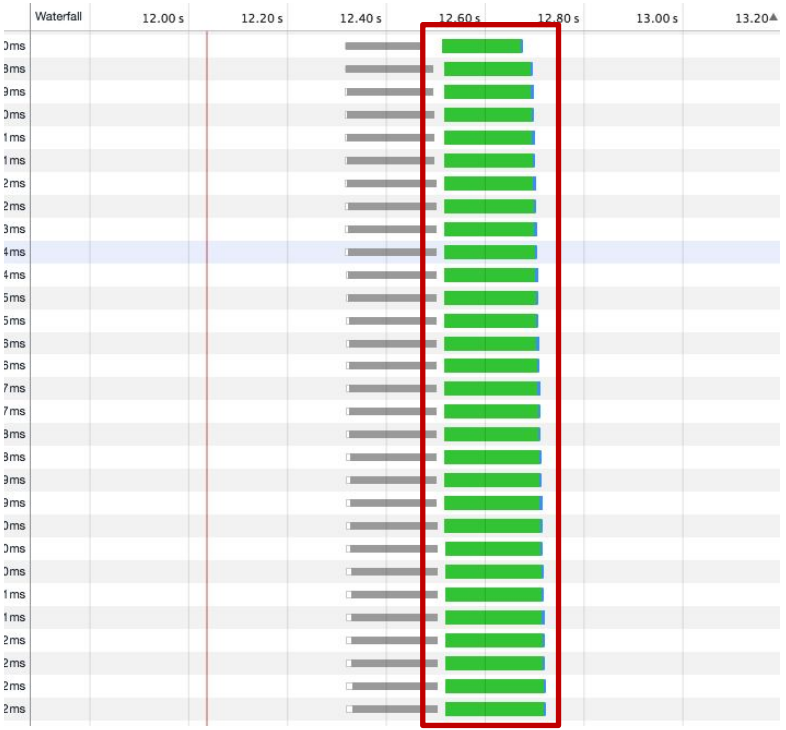


HTTP/2 waterfall



Demo concept inspired by Golang's Gophertiles

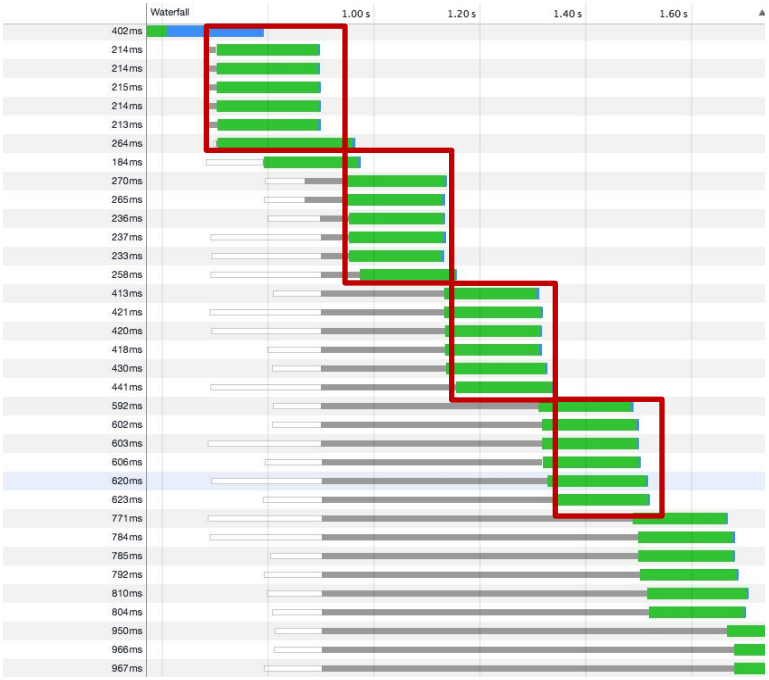
1 conn – multiple reqs



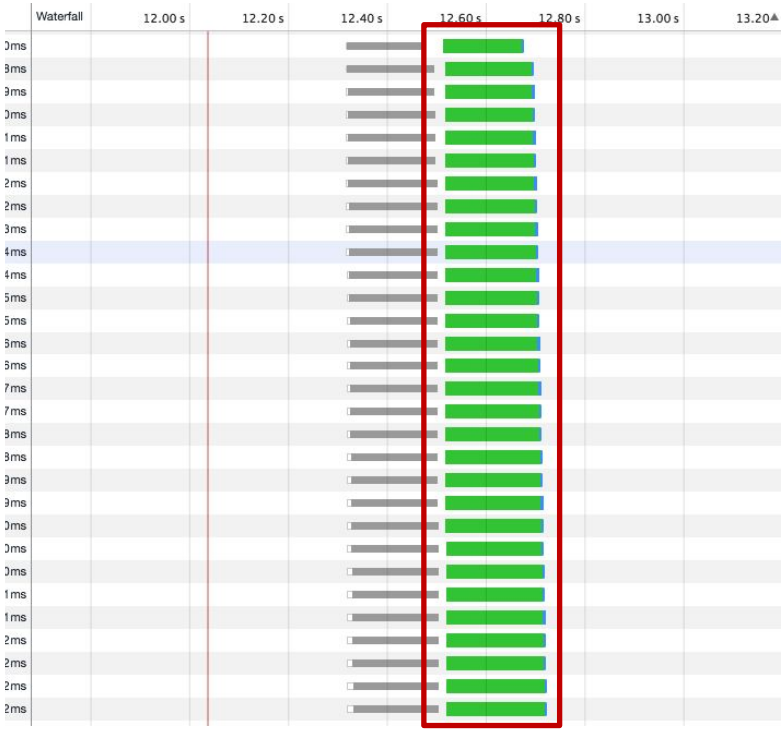
https://http2.akamai.com/demo

HTTP/1 vs HTTP/2 waterfall

HTTP/1: Max 6 conns a at time



HTTP/2: 1 conn – multiple reqs



No more HTTP/1 hacks

~~Concatenation~~

~~Inlining~~

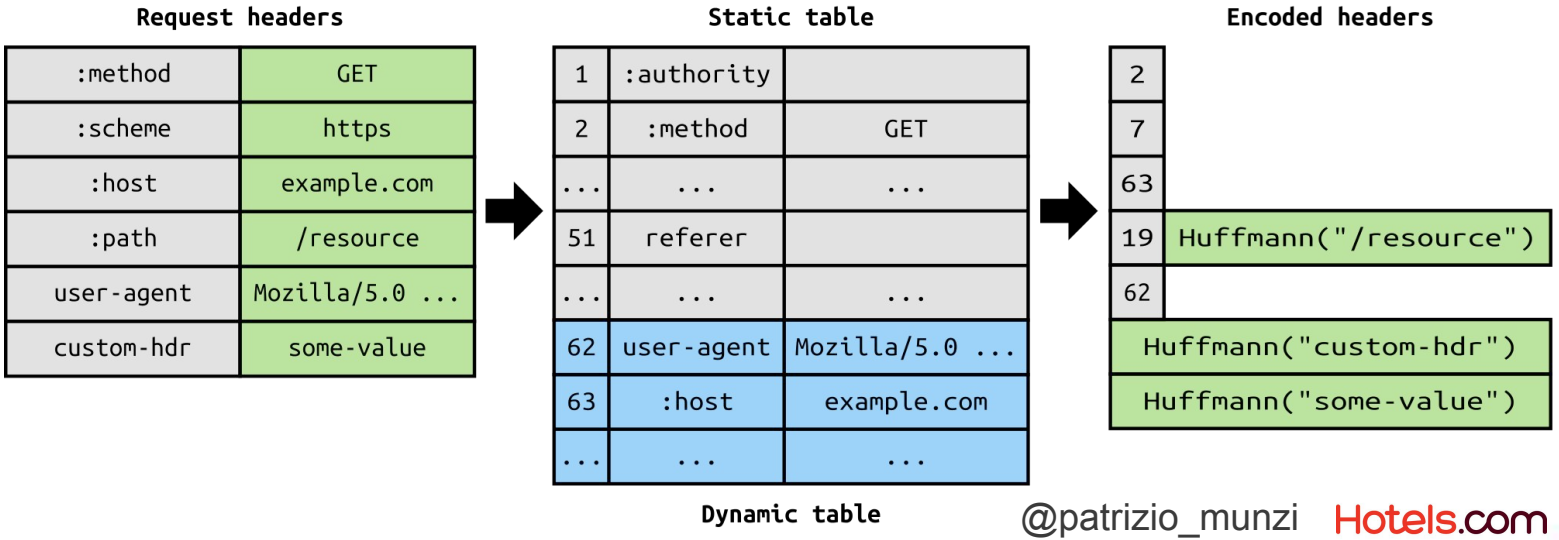
~~Spriting~~

~~Domain Sharding~~

Why HTTP/2 is faster?

2 - HPACK

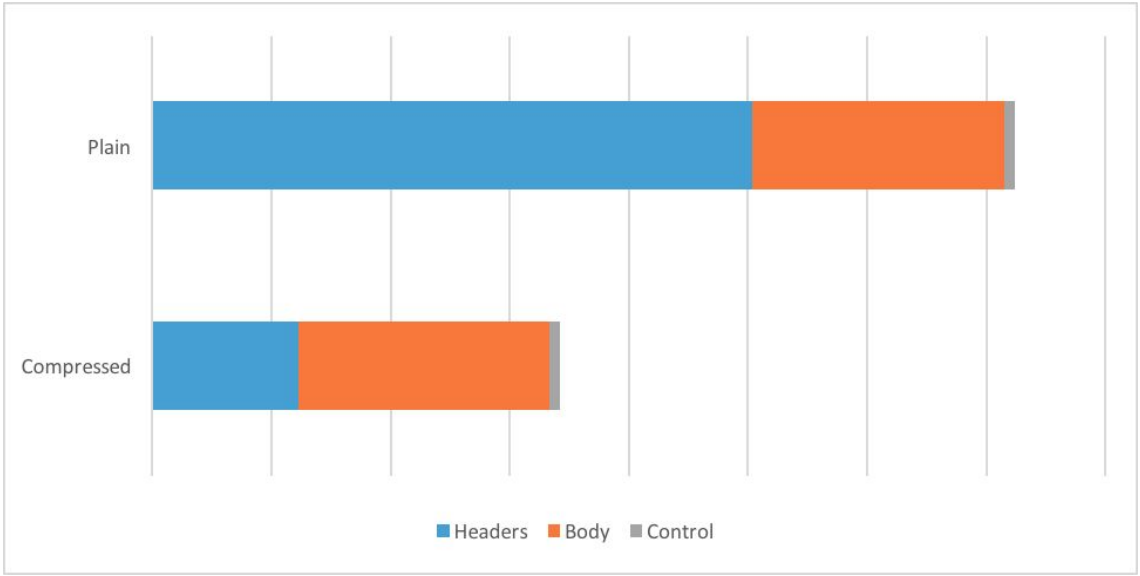
- **Static Dictionary**: 61 commonly used header fields
- **Dynamic Dictionary**: A list of actual headers that were encountered during the connection. Size limited.
- **Static Huffman Encoding**



Why HTTP/2 is faster?

2 - HPACK

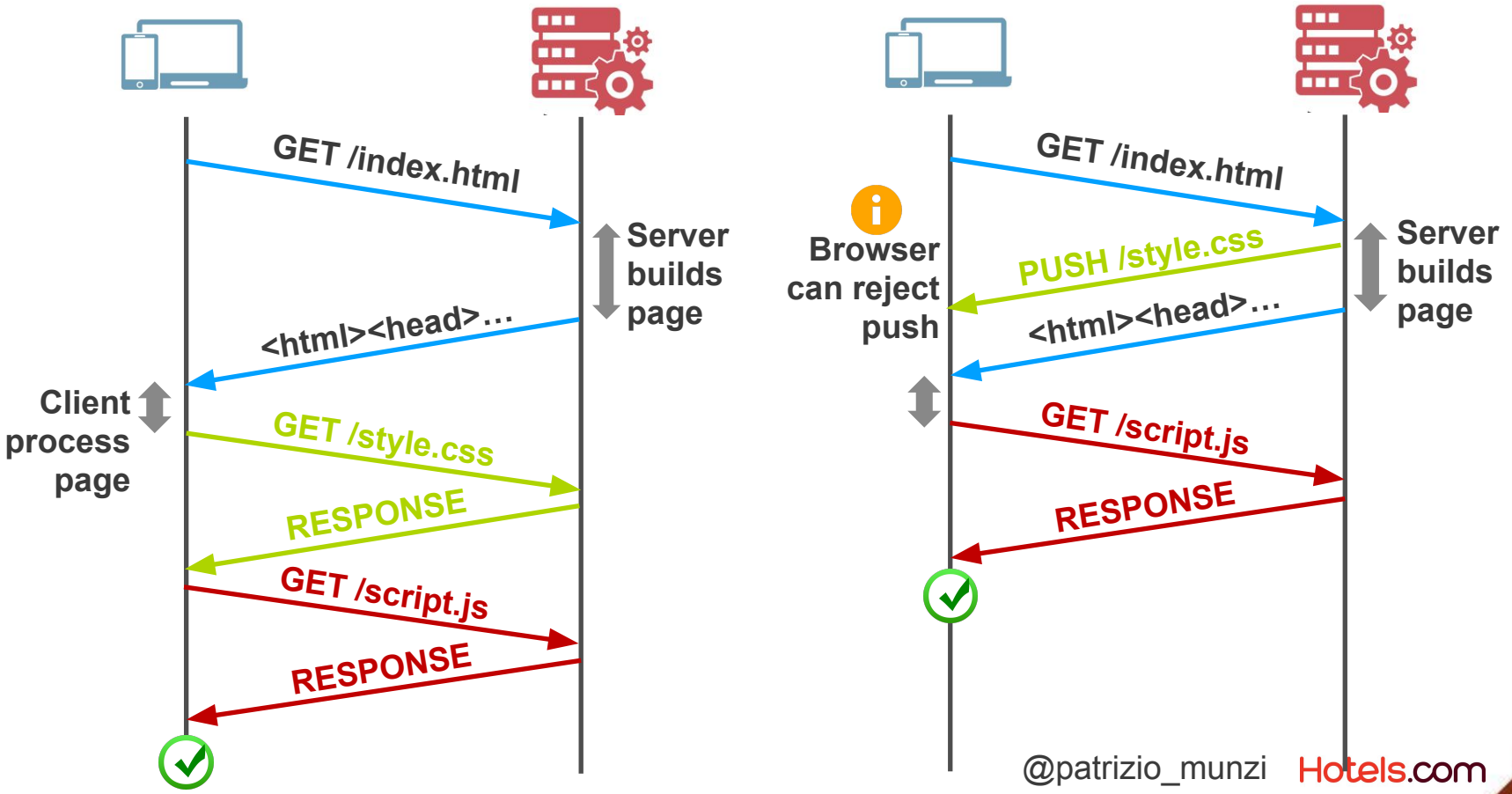
This is how ingress header traffic appears on the Cloudflare edge network during a six hour period



https://blog.cloudflare.com/hpack-the-silent-killer-feature-of-http-2/

Why HTTP/2 is faster?

3 - Server PUSH



HTTP/2 on jetty

```
$ java -jar $JETTY_HOME/start.jar --create-startd
--add-to-start=servlets,http,https,deploy,http2,http2c
$ java -jar $JETTY_HOME/start.jar
```

```
INFO:oejs.AbstractConnector:main: Started
ServerConnector@5f9b2141{SSL,[ssl, alpn, h2,
http/1.1]}{0.0.0.0:8443}
```

https://localhost:8443/http2/

HTTP/2 with Server Push

Add the push filter to your web.xml

web.xml

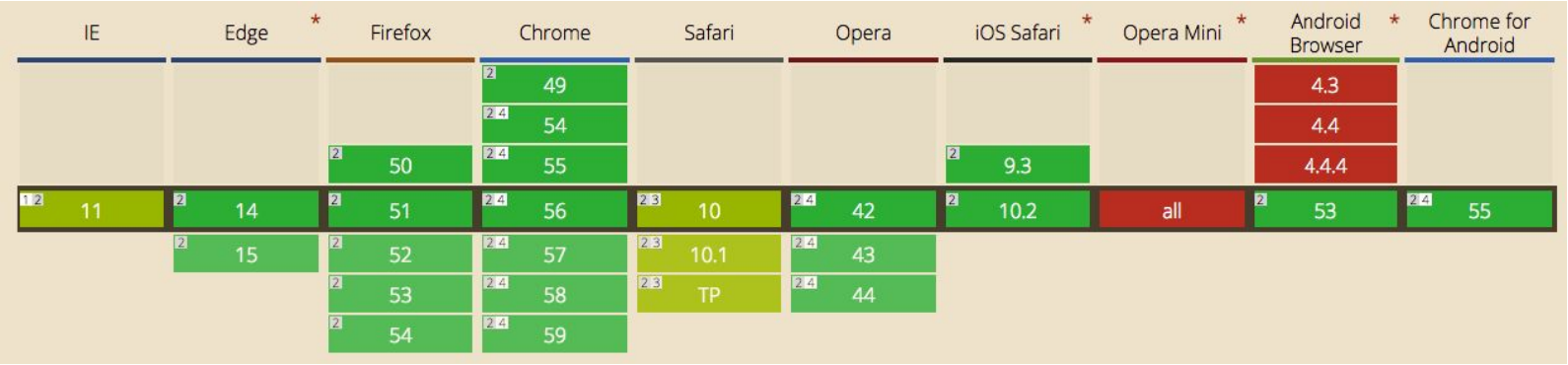
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app
    xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
    metadata-complete="true"
    version="3.1">

    <filter>
        <filter-name>PushFilter</filter-name>
        <filter-class>org.eclipse.jetty.servlets.PushCacheFilter</filter-class>
        <async-supported>true</async-supported>
    </filter>
    <filter-mapping>
        <filter-name>PushFilter</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>
</web-app>
```

https://localhost:8443/http2/

NOW?

(browsers)

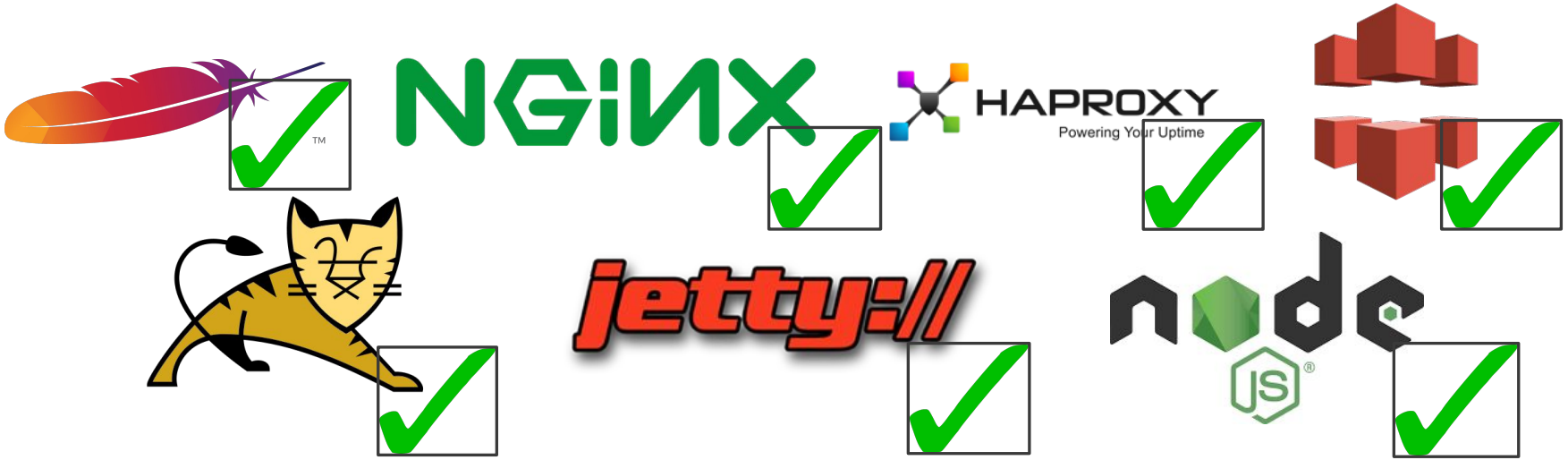


We're safe. HTTP/2 is an Upgrade Protocol

It kicks in only if the browser supports it

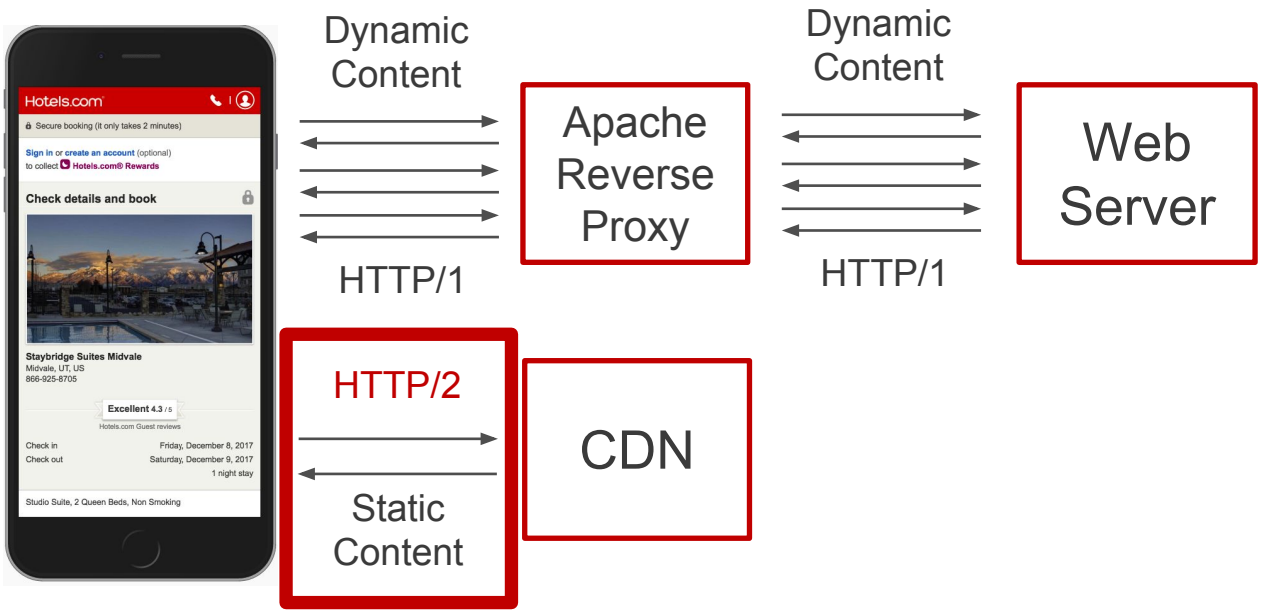
NOW?

(servers)



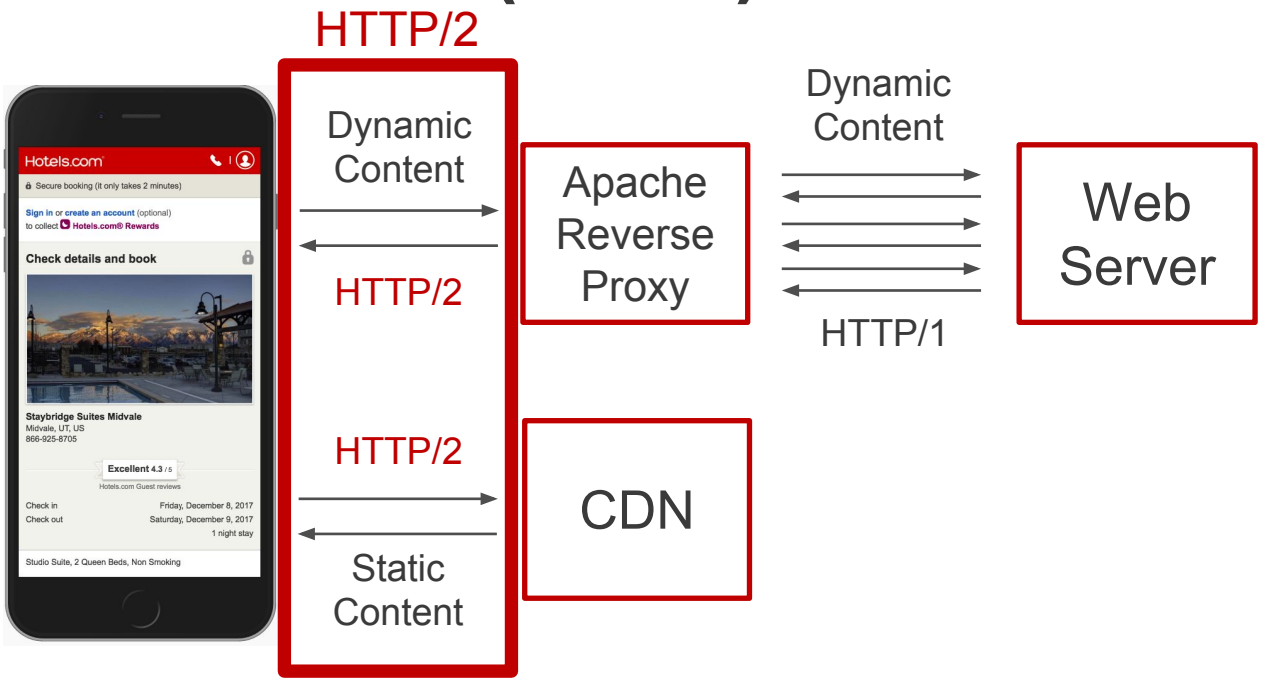
We're good

HTTP/2 adoption
(tier 1)



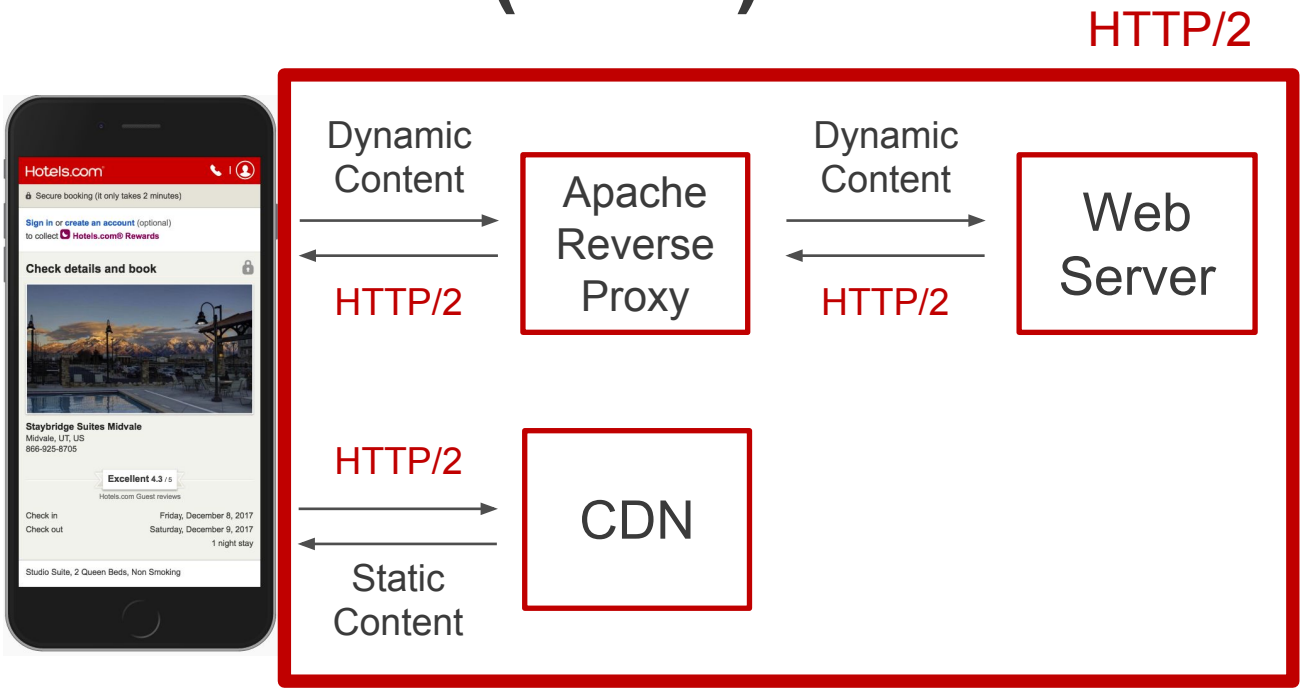
Put the static assets on a HTTP/2 CDN

HTTP/2 adoption
(tier 2)



Have a reverse proxy translating HTTP/2 calls to HTTP/1

HTTP/2 adoption
(tier 3)



The whole infrastructure over HTTP/2

References

- https://http2.github.io/
- https://http2.akamai.com/demo
- https://blog.cloudflare.com/hpack-the-silent-killer-feature-of-http-2/
- https://en.wikipedia.org/wiki/Huffman_coding
- https://webtide.com/introduction-to-http2-in-jetty/
- https://community.akamai.com/videos/1919-the-promise-of-http2-h2-server-push
- https://github.com/jetty-project/http2-demo
- https://www.webpagefx.com/blog/internet/website-page-load-time-conversions/
- https://blog.gigaspaces.com/amazon-found-every-100ms-of-latency-cost-them-1-in-sales/