Felix Gessert

Baqend

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

AMP, PWAs, HTTP/2 and Service Workers: A New Era of Web Performance?

Mobile Track

code.talks by ABOUT YOU

Who is talking today?

Felix Gessert

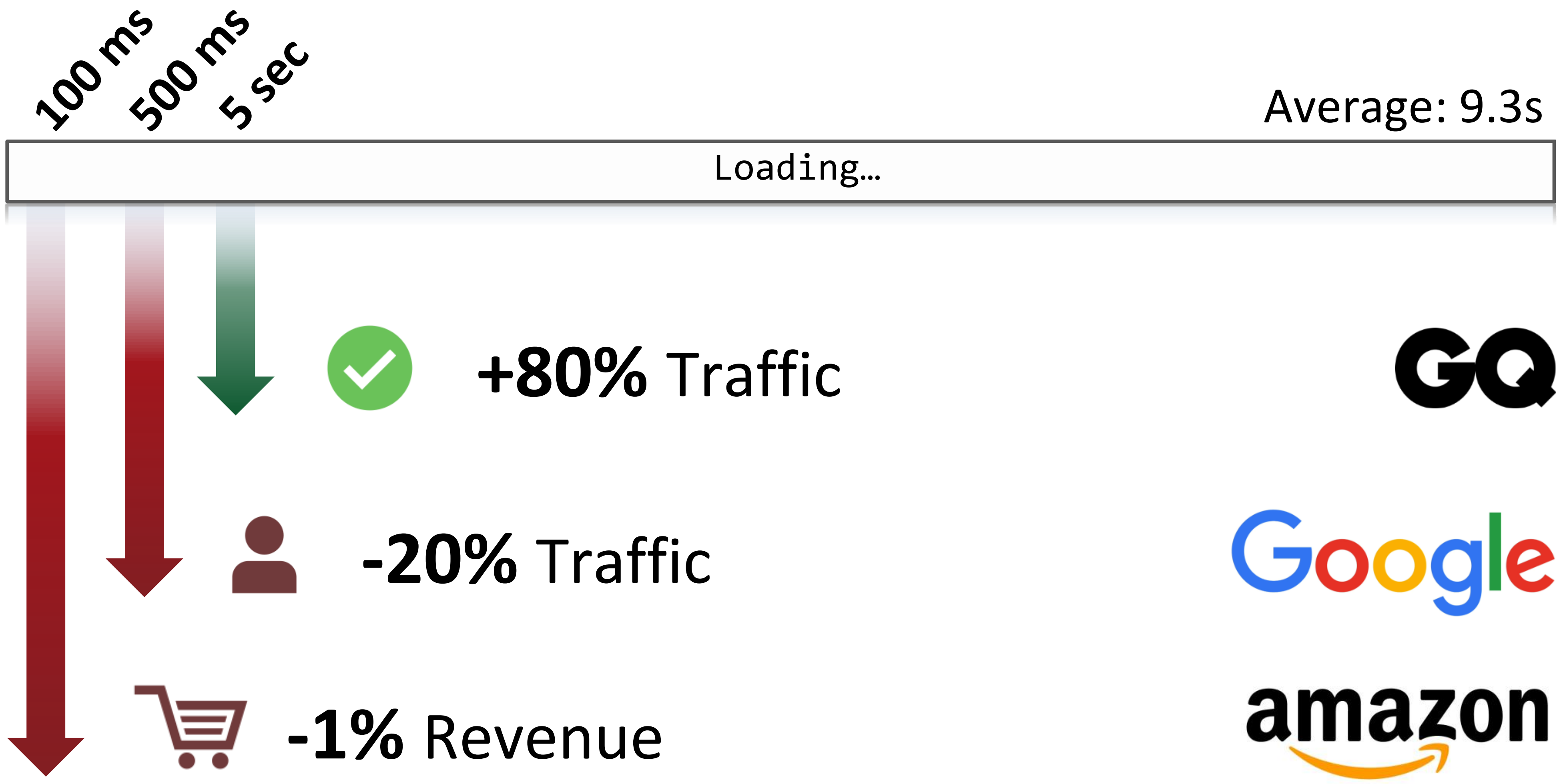
PhD Thesis
Web Performance
Cloud Data Management

Universität Hamburg

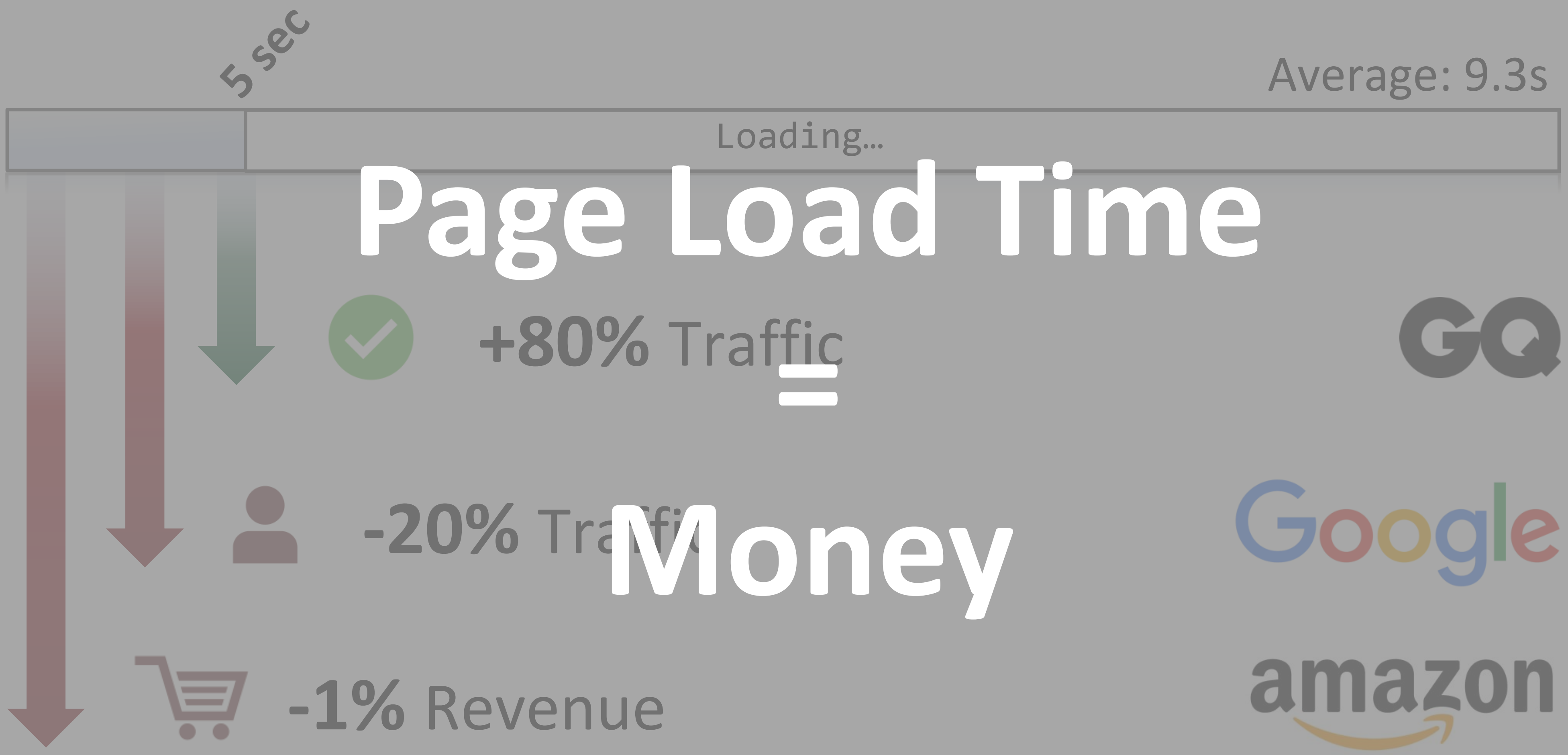
CEO & Co-Founder
Baqend Platform
Speed Kit Plugin

Baqend

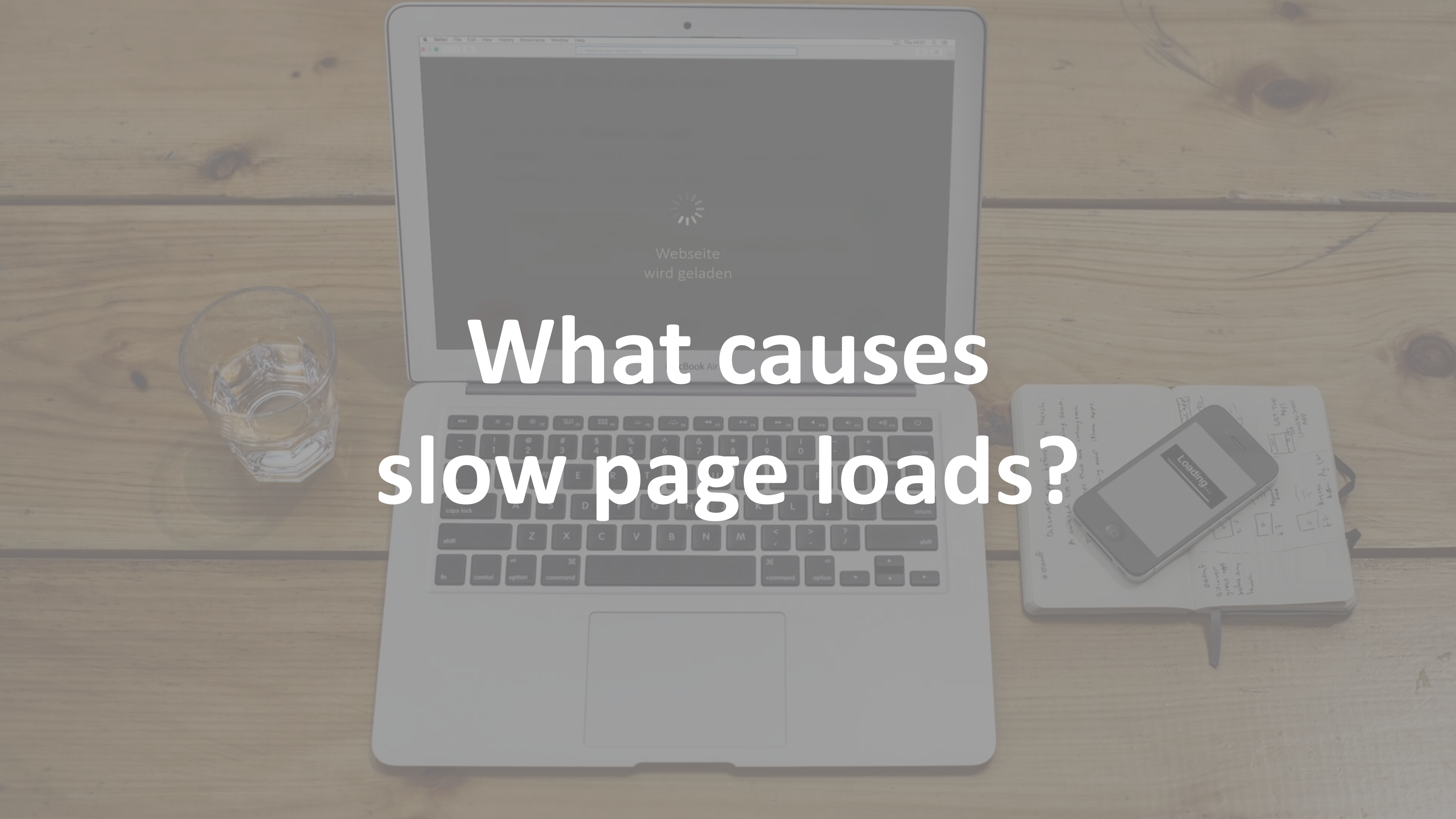
Presentation
is loading


100 ms
500 ms
5 sec
Average: 9.3s
Loading...
+80% Traffic
-20% Traffic
-1% Revenue

Page Load Time
=
Money

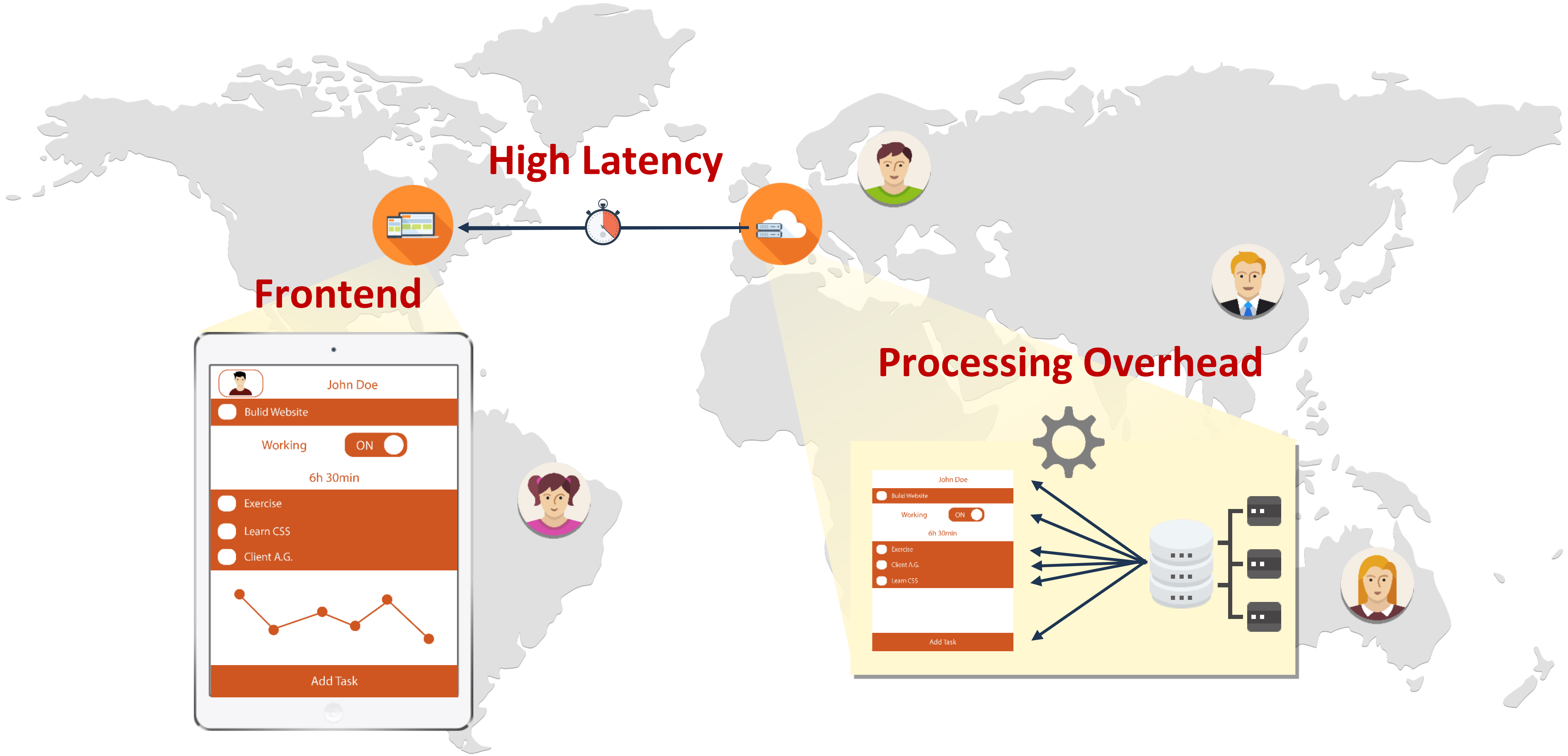


Webseite
wird geladen
MacBook Air
Loading...
Discover apps before they launch.
A curated list of apps coming soon.



What causes slow page loads?

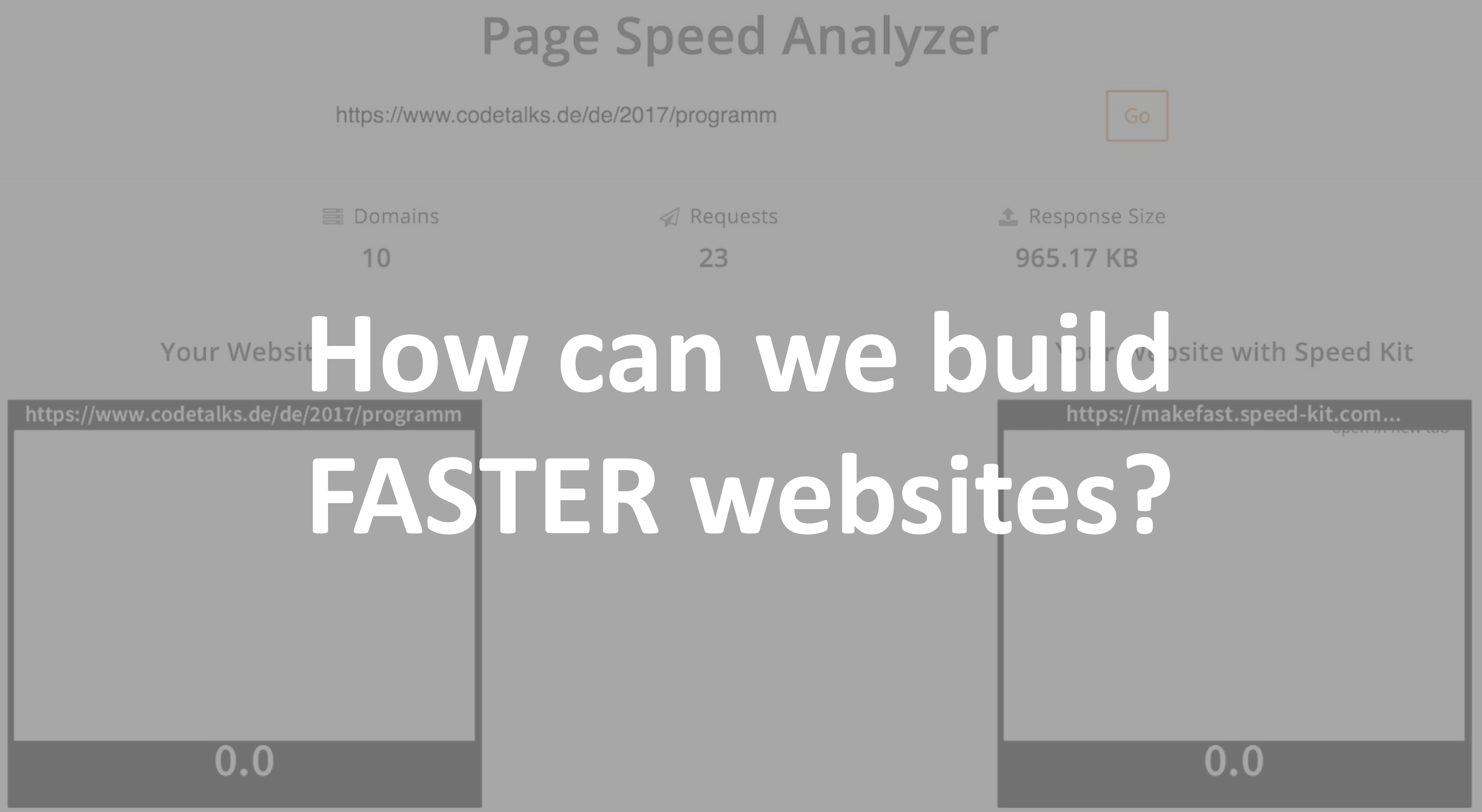
There are 3 performance problems.



There are 3 performance problems.

Is this really a problem in practice?

How can we build FASTER websites?



Page Speed Analyzer

https://www.codetalks.de/de/2017/programm

Domains	Requests	Response Size
10	23	965.17 KB

Your Websit

Your website with Speed Kit

https://www.codetalks.de/de/2017/programm

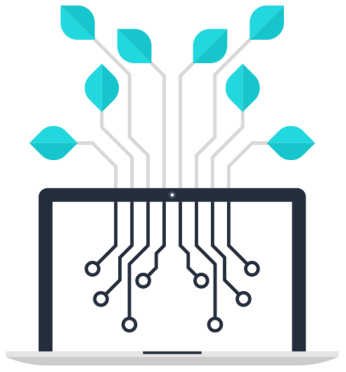
0.0

https://makefast.speed-kit.com...

0.0

What we are going to cover.

Frontend



Google AMP
Instant Articles
Progressive Web Apps

Network

HTTP/2
Service Workers

Backend

Cloud & NoSQL
Speed Kit

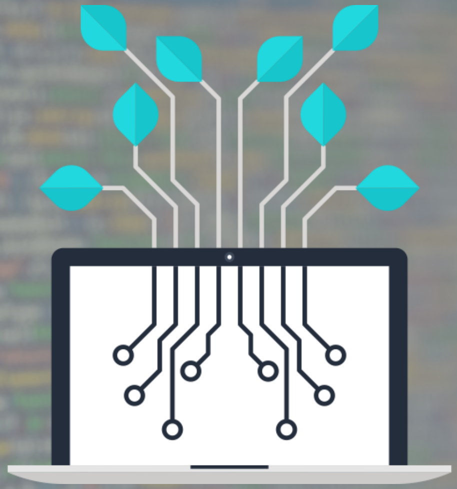
What is the goal?

Delay	Perception
0 – 100 ms	Instant
100 – 300 ms	Small perceptible delay
300 – 1000 ms	Machine is working
1+ s	Mental context switch
10+ s	Task abandoned

→

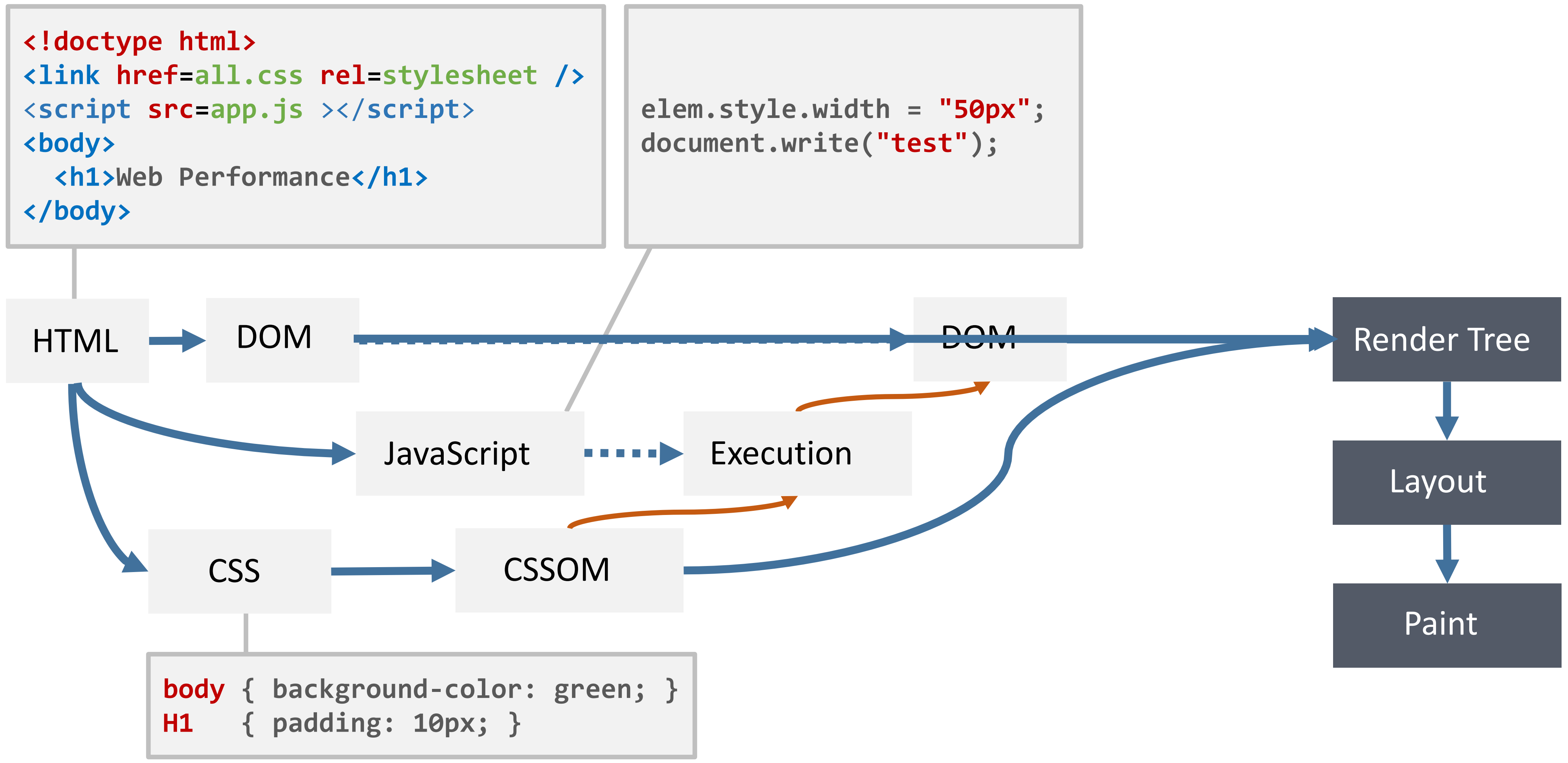
Load time < 1s

I. Grigorik, High performance browser networking. O'Reilly Media, 2013.



1. Frontend Performance

Frontend: Critical Rendering Path


```html
<!doctype html>
<link href=all.css rel=stylesheet />
<script src=app.js ></script>
<body>
  <h1>Web Performance</h1>
</body>
```

```javascript
elem.style.width = "50px";
document.write("test");
```

```css
body { background-color: green; }
H1   { padding: 10px; }
```

→ *Dependency* ⇢ *Delayed By Other Resource* → *Blocks*

Frontend: **Critical Rendering Path**

Best Practices

1. Minimize **Length** (*Round-Trips*)
2. Minimize **Size** (*Critical Resources*)
3. Minimize **Weight** (*Critical Bytes*)

Inlining cricital CSS and JS "above the fold"

Critical

concat nd JS

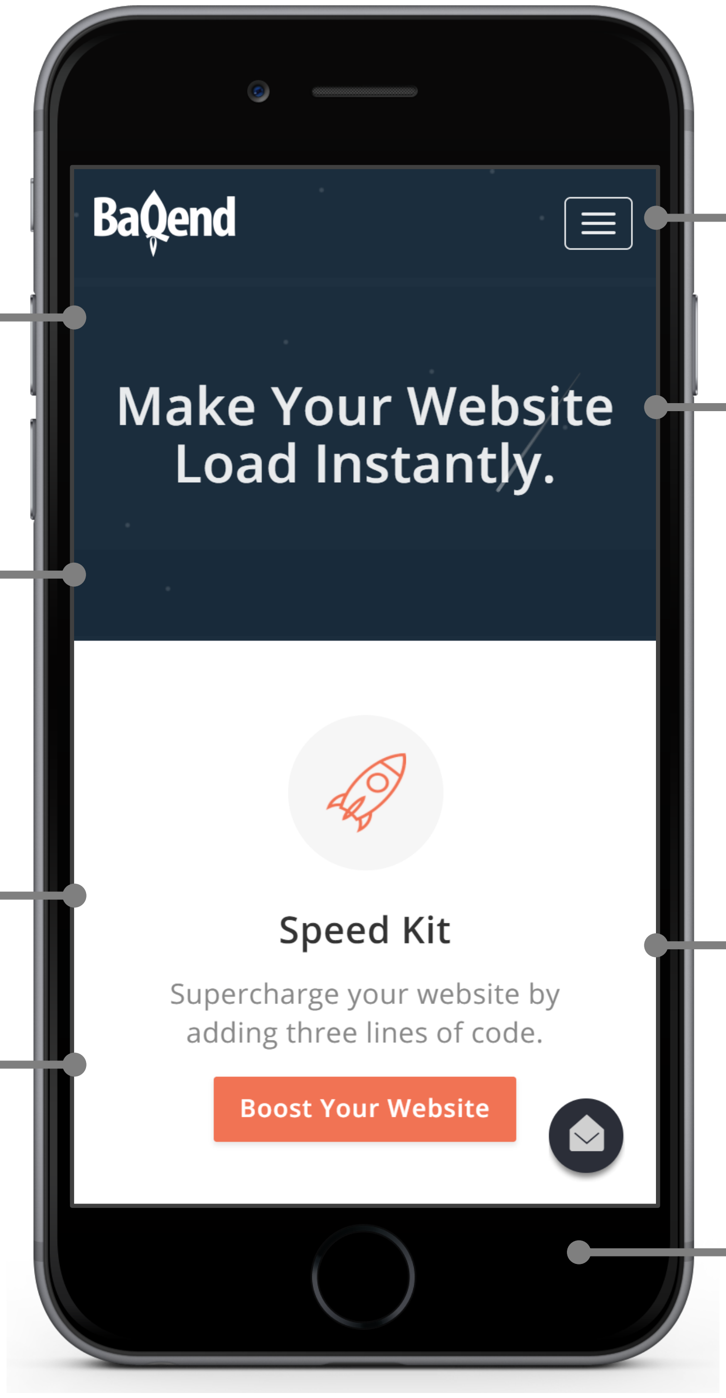
UglifyJs & cssmin

Single page application

handlebars

F ssive ering

processhtml


BaQend
Make Your Website Load Instantly.
Speed Kit
Supercharge your website by adding three lines of code.
Boost Your Website

PostCSS

Co ess images

tiny png

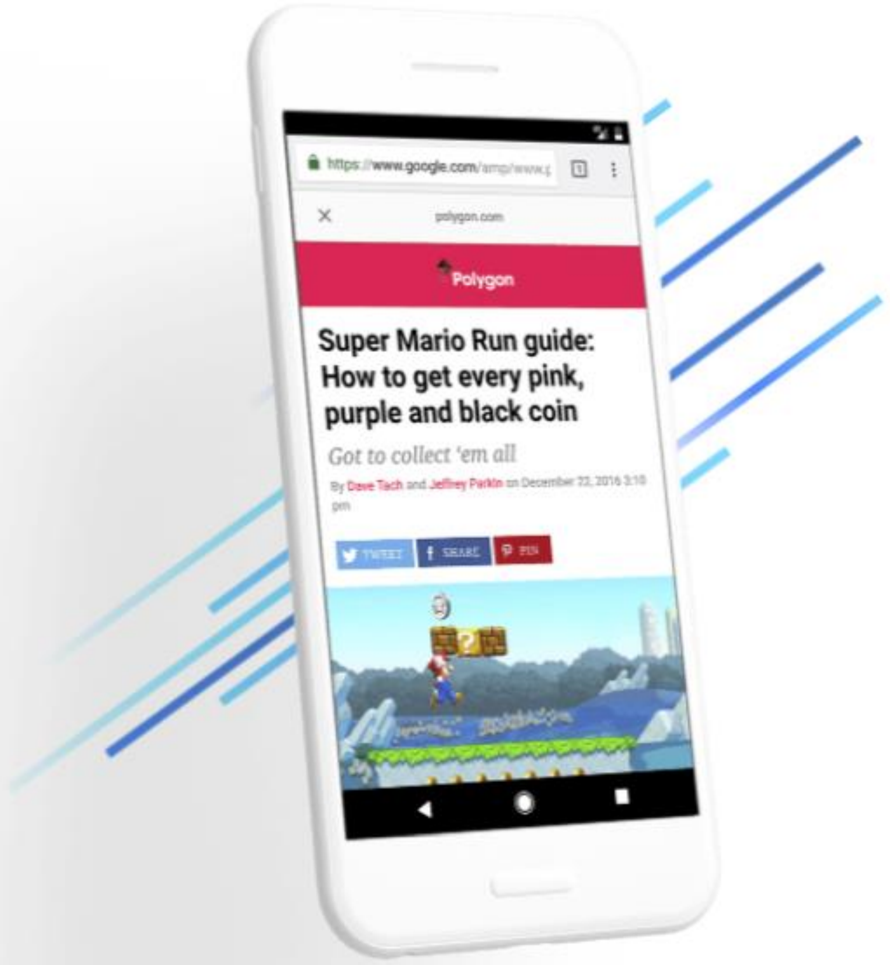
GTmetrix

Test your performance

WEBPAGETEST

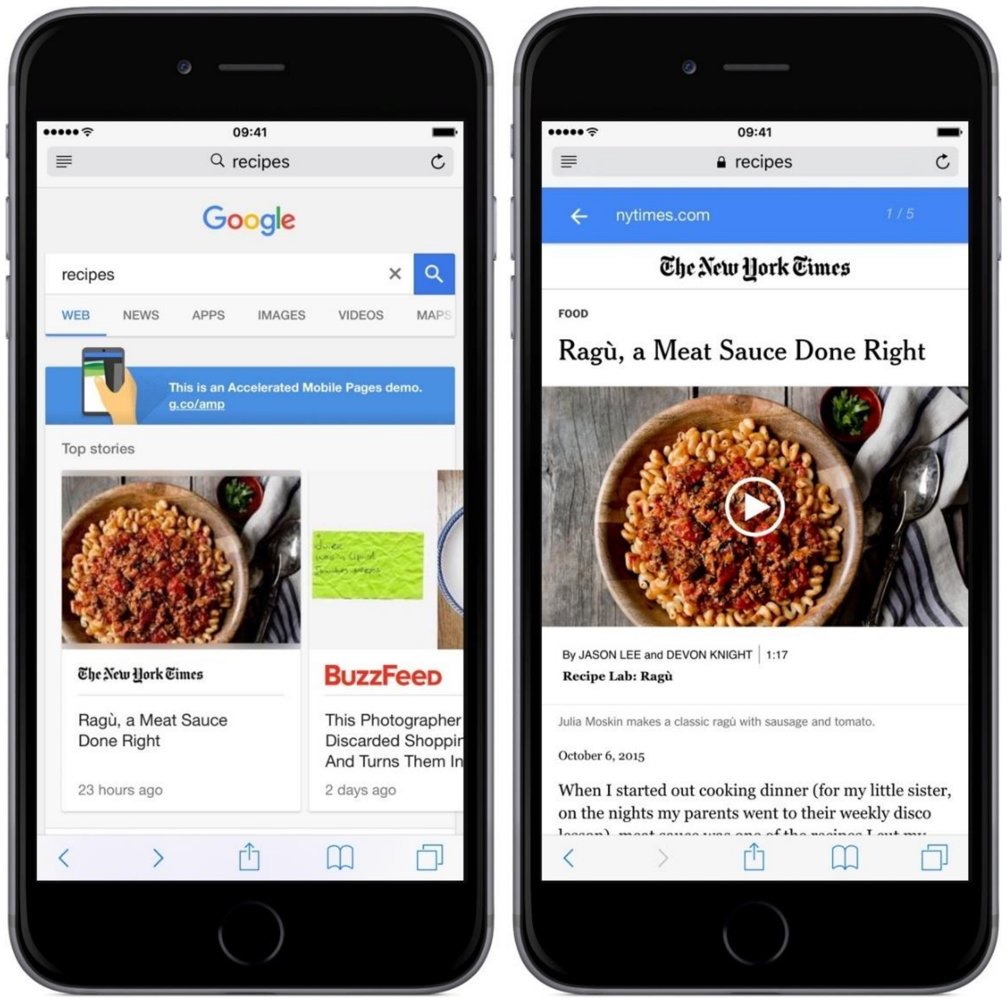
Load non-critical CSS and JS asynchronously

PageSpeed Insights


https://www.google.com/amp/www.
polygon.com
Polygon
Super Mario Run guide: How to get every pink, purple and black coin
Got to collect 'em all
By Dave Tach and Jeffrey Parkin on December 22, 2016 3:10 pm
TWEET
SHARE
PIN

Google‘s vision for a better web: AMP

Accelerated Mobile Pages (AMP)


09:41
recipes
Google
recipes
WEB
NEWS
APPS
IMAGES
VIDEOS
MAPS
This is an Accelerated Mobile Pages demo.
g.co/amp
Top stories
The New York Times
Ragù, a Meat Sauce Done Right
23 hours ago
BuzzFeed
This Photographer Discarded Shoppi
And Turns Them In
2 days ago
09:41
recipes
nytimes.com
1 / 5
The New York Times
FOOD
Ragù, a Meat Sauce Done Right
By JASON LEE and DEVON KNIGHT 1:17
Recipe Lab: Ragù
Julia Moskin makes a classic ragù with sausage and tomato.
October 6, 2015
When I started out cooking dinner (for my little sister, on the nights my parents went to their weekly disco

How AMP works:

- **Stripped down** HTML + AMP tags (e.g. img) → rendered asynchronously by runtime
- CSS must be **inlined + <50 KB**
- No **custom JS** (except in iframes)
- Only static sizes → **no repaints**
- **No Forms** & only for **mobile**
- Pre-Loaded in **Google Results**
- **Cached** in Google CDN, as long as it is crawled the next time

https://www.ampproject.org/docs/reference/spec.html

Implementing AMP for a website

1. Link to AMP Version:

```
<link rel="amphtml" href="full-url-to-amp-version">
```

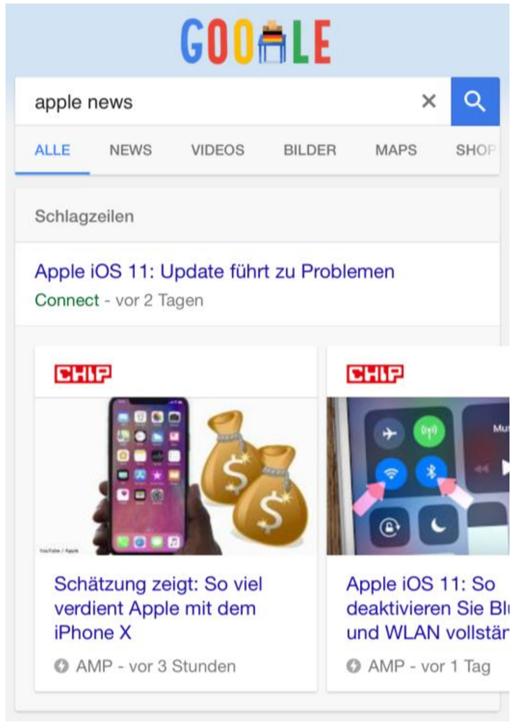
2. Use HTML Boilerplate:

```
<!doctype html>
<html ⚡ lang="en">
  <head> …
```

3. AMP Tags:

```
<amp-img src="logo.png" width="100" height="40">
```

AMP: the Good↑



Schlagzeilen

Apple iOS 11: Update führt zu Problemen
Connect - vor 2 Tagen

Schätzung zeigt: So viel verdient Apple mit dem iPhone X
AMP - vor 3 Stunden

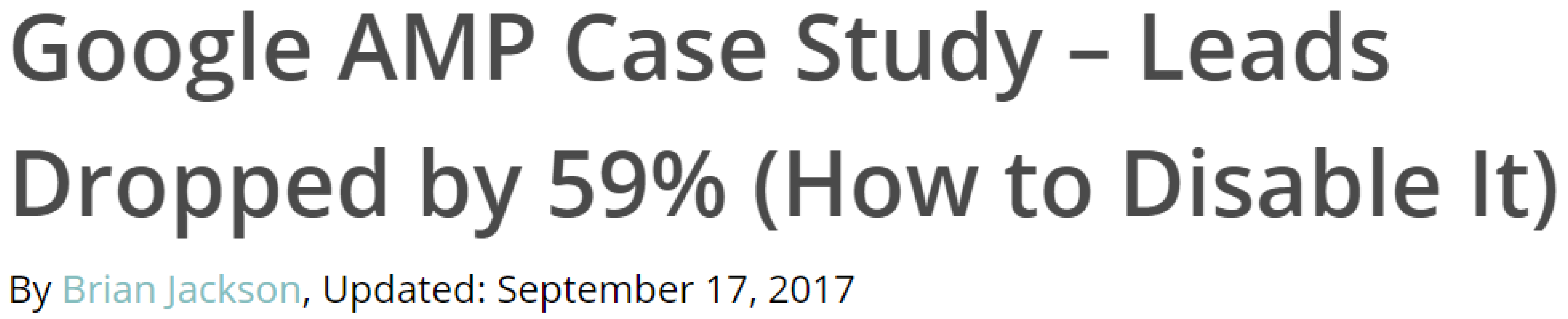
Apple iOS 11: So deaktivieren Sie Bl und WLAN vollstär
AMP - vor 1 Tag

Fast Mobile **Loads**

Google Result Carousel

Works Well for **Static Sites**

AMP: the Bad↓


Google AMP Case Study – Leads Dropped by 59% (How to Disable It)

By Brian Jackson, Updated: September 17, 2017

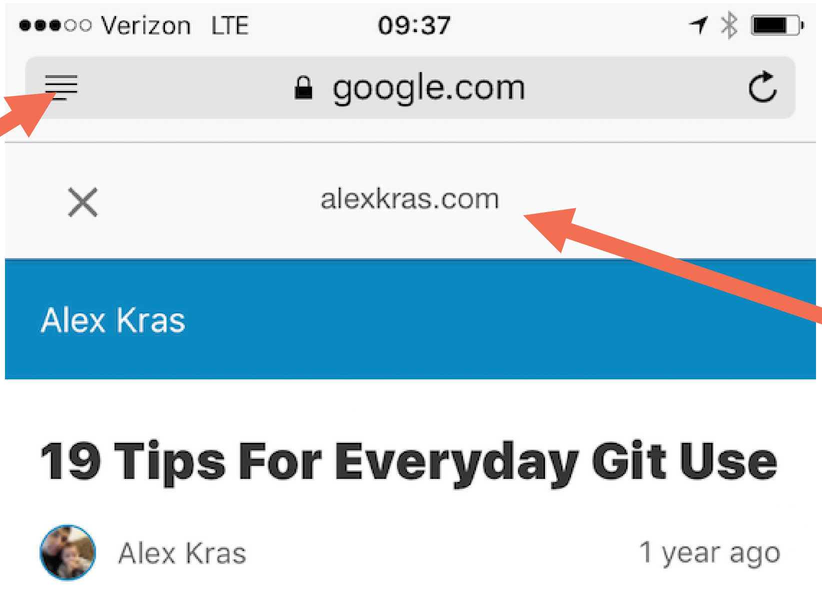
- 59% Leads

- 17% Signups

AMP: the Bad↓

Google May Be Stealing Your Mobile Traffic

October 15, 2016 by Alex Kras — 223 Comments



Google CDN URL

Injected Bar

AMP: the Ugly↓

Kill Google AMP before it KILLS the web

By Scott Gilbertson 19 May 2017 at 08:25 113

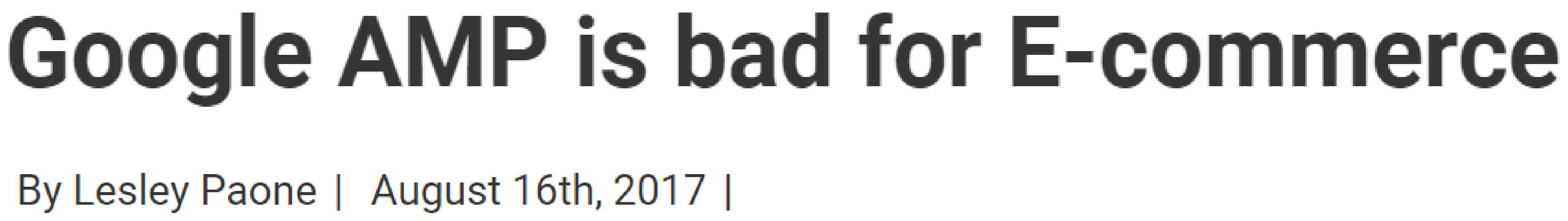
No JS: only Google‘s

Bad UX: iOS Scrolling

No Custom Analytics

Assumes Dumb Devs

AMP: the Ugly↓



Google AMP is bad for E-commerce

By Lesley Paone | August 16th, 2017 |

No **Chat** or **Payment**

No **Search** or **Login**

AMP: the Bad↓

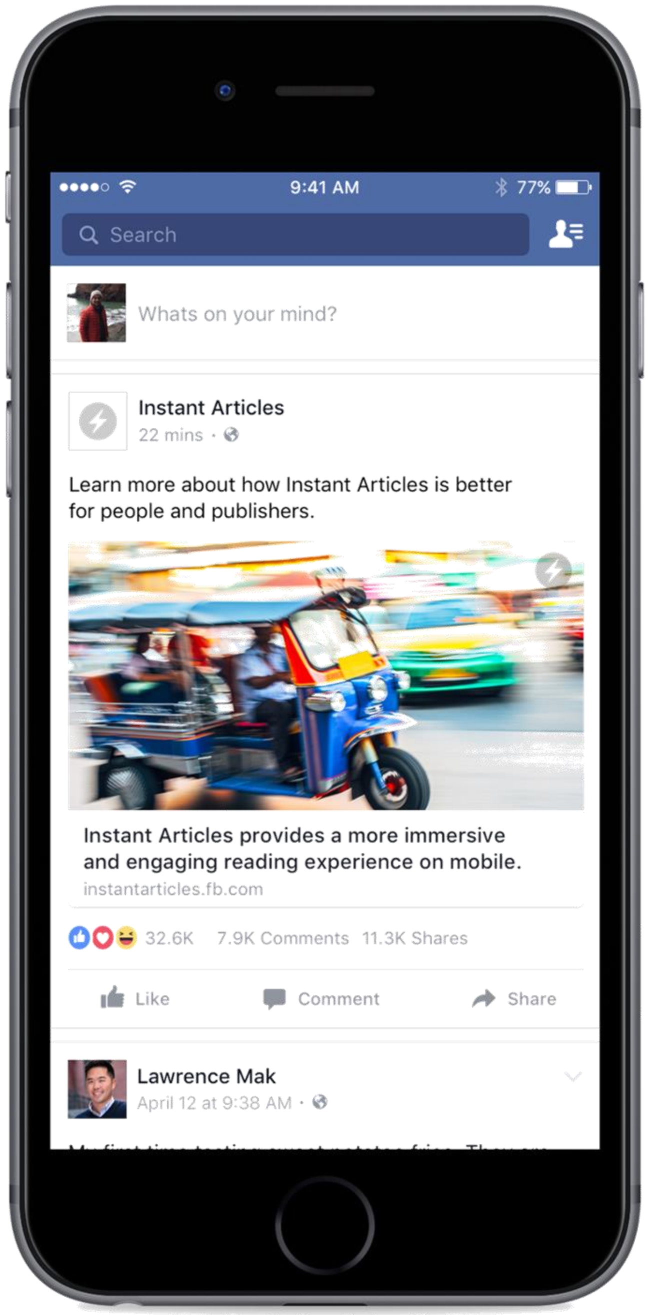
Google AMP is bad for E-commerce

By Lesley Pavone | August 10th, 2017 |

AMP started as good idea but it is too limiting.

No **Chat** or **Payment**

No **Search** or **Login**



●●●●○ 9:41 AM 77%

Search

Whats on your mind?

Instant Articles
22 mins ·

Learn more about how Instant Articles is better for people and publishers.

Instant Articles provides a more immersive and engaging reading experience on mobile.
instantarticles.fb.com

32.6K 7.9K Comments 11.3K Shares

Lawrence Mak
April 12 at 9:38 AM ·

Facebook‘s Alternative: Instant Articles

Facebook Instant Articles

- Single **HTML Document**
- **No CSS/JS**
- **Designed** in FB Editor
- Crawled from **RSS Feed**

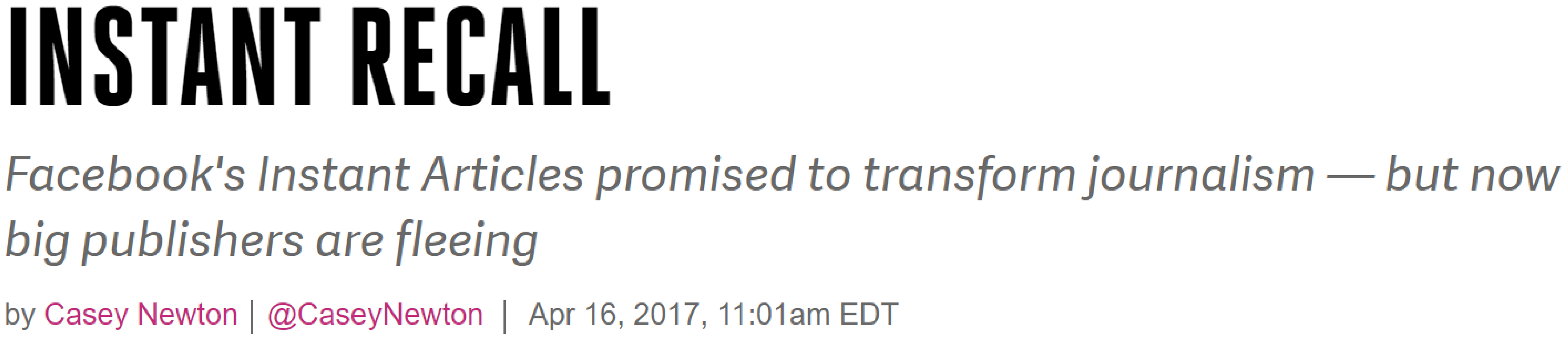
```
<head>
  <meta property="op:markup_version" content="v1.0">
  <!-- The URL of web version-->
  <link rel="canonical" href="http://example.com/article.html">
  <meta property="fb:article_style" content="myarticlestyle">
</head>
<body>
  <article>   ...  </article>
</body>
```

Instances Articles: the Good↑

Fast Mobile
Loads

Good UX for
Facebook Users

Instances Articles: the Bad↓



INSTANT RECALL

Facebook's Instant Articles promised to transform journalism — but now big publishers are fleeing

by Casey Newton | @CaseyNewton | Apr 16, 2017, 11:01am EDT

FB makes the **rules**

Users stop visit-ing **real site**

Why not apply the *good* ideas to any website?

Progressive Web Apps

What are Progessive Web Apps?

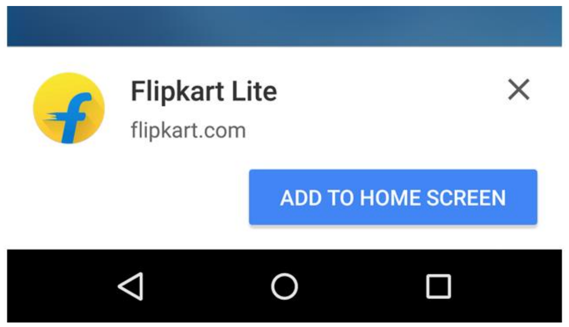
Try this:

codetalks.baqend.com

Progressive Web Apps (PWAs)

+

+



Fast Loads through Caching

Offline Mode (Service Workers)

Add-to-Homescreen and Push

Implementing PWAs

- PWAs are **best practices** not a technology
- **Progessively enhance** when supported

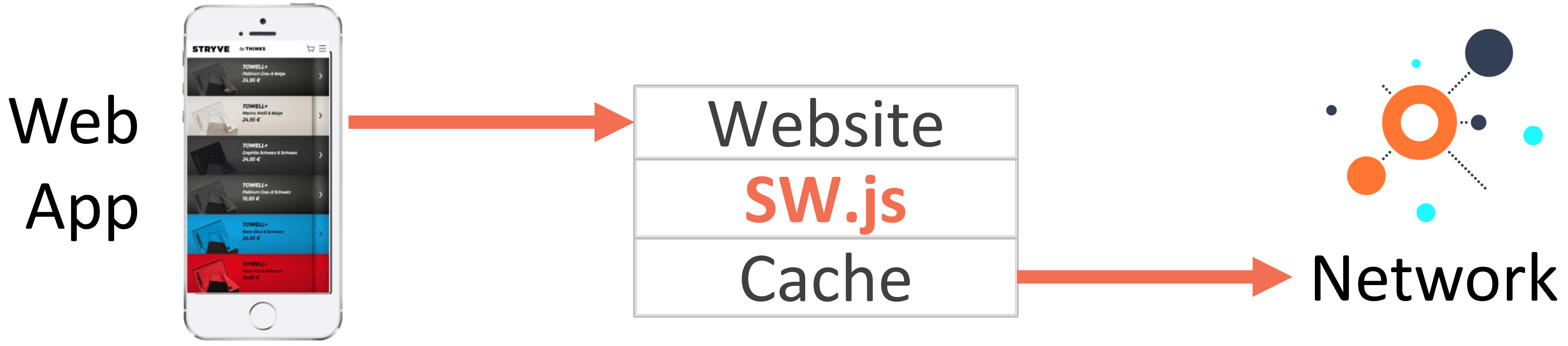
1. Manifest declares Add-to-Homescreen:

```
<link rel="manifest" href="/manifest.json">
{
  "short_name": "Codetalks PWA",
  "icons": [
    {"src": "icon-1x.png", "type": "image/png", "sizes": "48x48"}],
  "start_url": "index.html?launcher=true"
}
```

Implementing PWAs

- PWAs are **best practices** not a technology
- **Gracefully degrade** when not supported

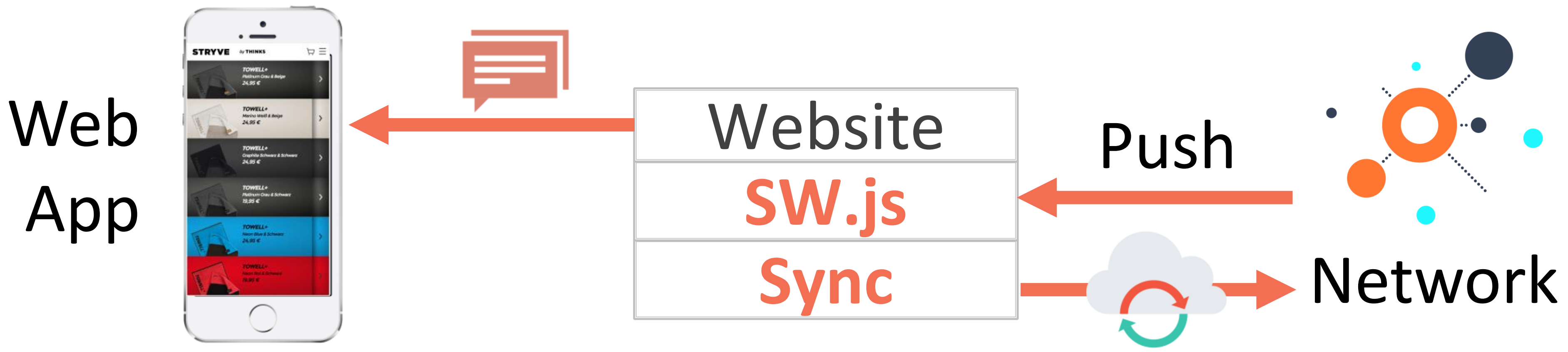
2. Service Workers for caching & offline mode:



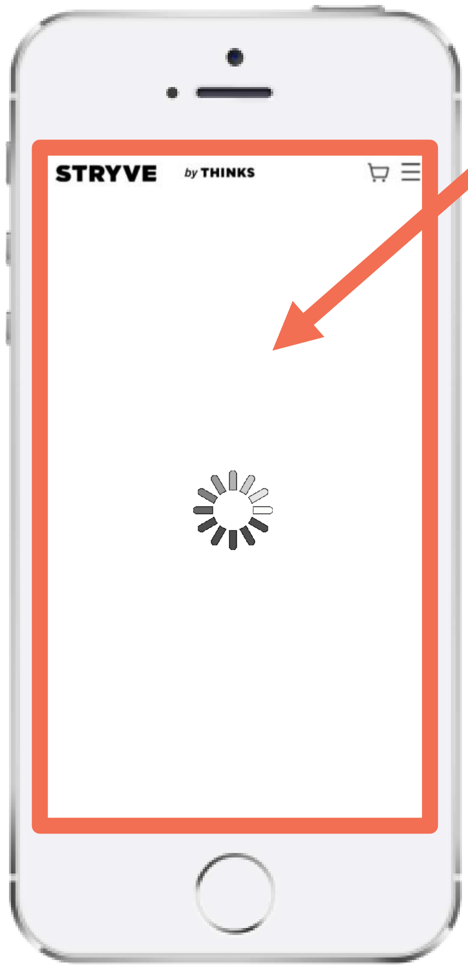
Implementing PWAs

- PWAs are **best practices** not a technology
- **Progressively enhance** the user experience

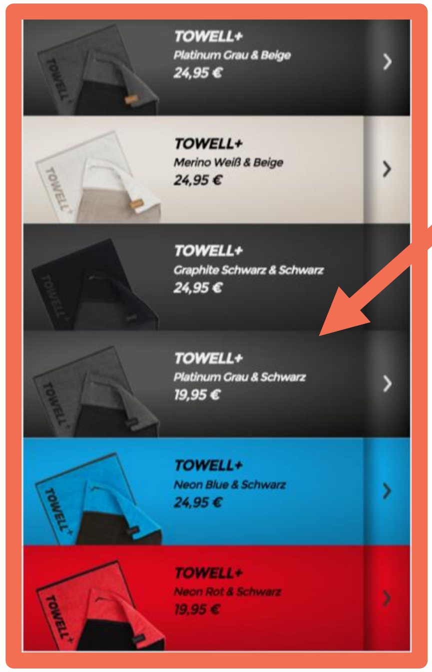
3. Add Web Push and Background Sync:



Typical Architecture: App Shell Model



App Shell: **HTML, JS, CSS, images** with app logic & layout



Content: Fetched on demand & may change more often

Why PWAs over AMP & Instant Articles?

Independent Technology

Work across **Devices**

No **Restrictions** on Development

Why PWAs over AMP & Instant Articles?

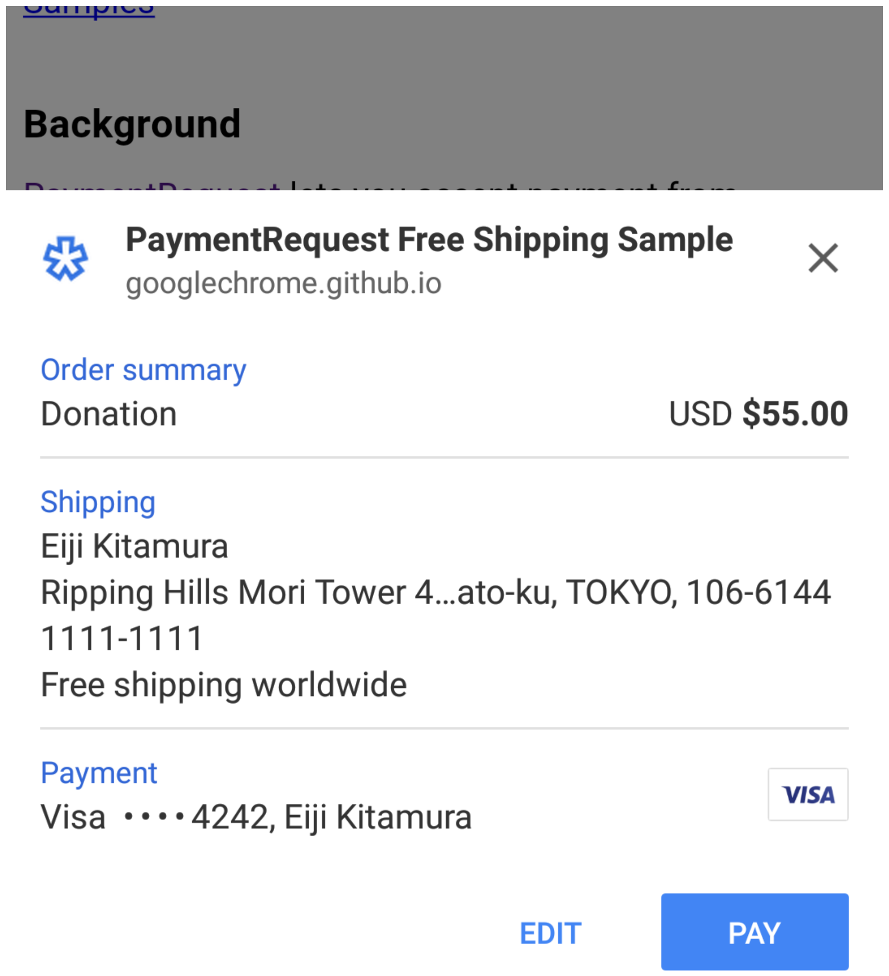
What is the future of Progessive Web Apps?

Independent Technology

Work across **Devices**

No **Restrictions** on Development

The Future of PWAs is bright.


Background

PaymentRequest Free Shipping Sample
googlechrome.github.io

Order summary
Donation USD **$55.00**

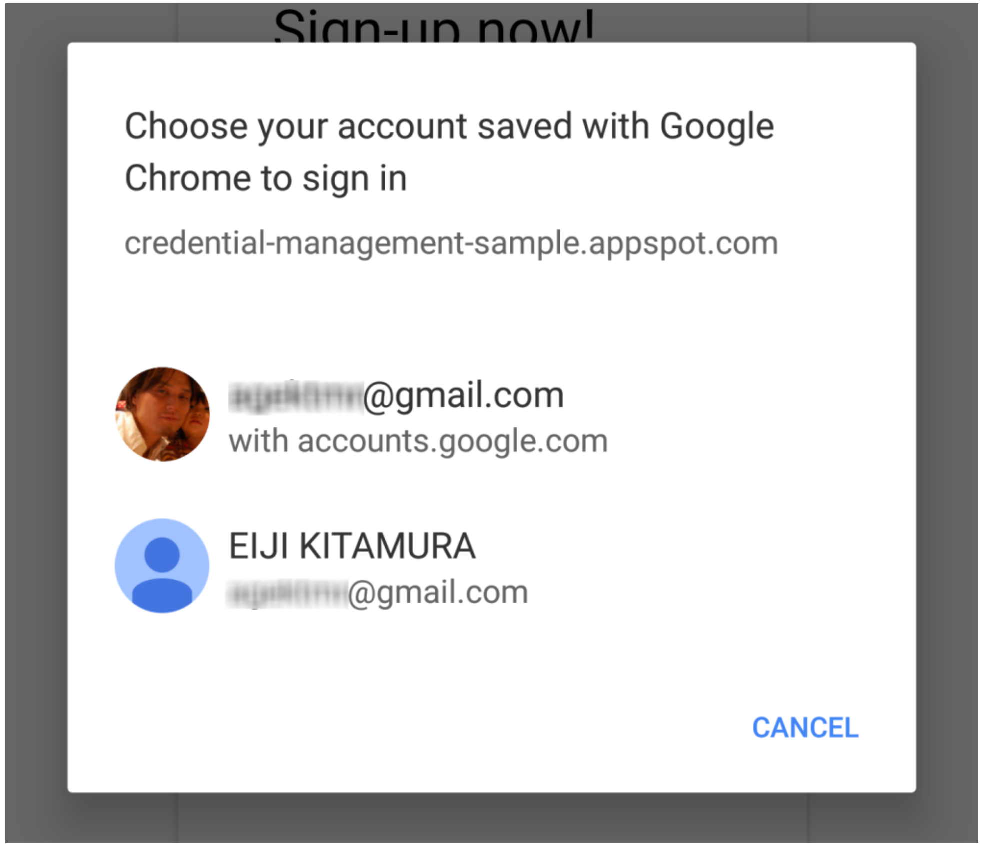
Shipping
Eiji Kitamura
Ripping Hills Mori Tower 4...ato-ku, TOKYO, 106-6144
1111-1111
Free shipping worldwide

Payment
Visa ••••4242, Eiji Kitamura

Payment Request API

- Goal: replace traditional **checkout** forms
- Just ~10 LOC for a **payment**
- Vendor- & Browser- **Agnostic**

The Future of PWAs is bright.


Sign-up now!

Choose your account saved with Google Chrome to sign in

credential-management-sample.appspot.com

[redacted]@gmail.com
with accounts.google.com

EIJI KITAMURA
[redacted]@gmail.com

Credentials Management API

1. Click **Sign-in** → Native Account Chooser
2. Credentials API **stores** information for future use
3. **Automatic** Sign-in afterwards

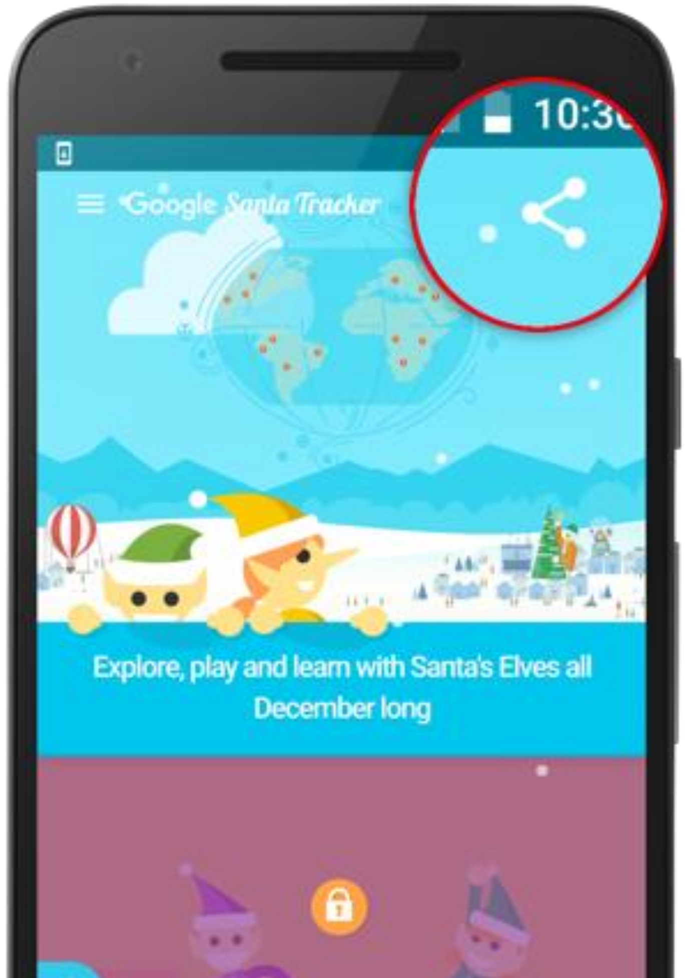
The Future of PWAs is bright.

Web Speech API

Native Speech Recognition in the Browser:

```
annyang.addCommands({
  'Hello Code.talks': () => {
    console.log('Hello you.');
  }
});
```

The Future of PWAs is bright.

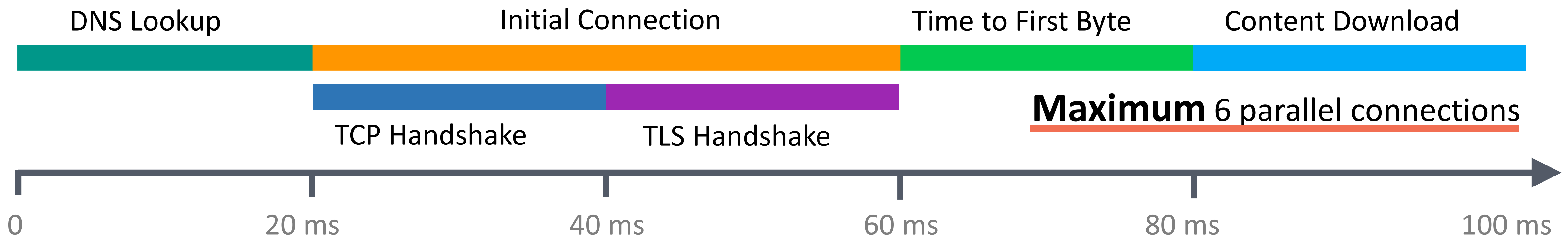

10:30
Google Santa Tracker
Explore, play and learn with Santa's Elves all December long

Web Share API

- **Share** site through native share sheet UI
- Service Worker can register as a **Share Target**

2. Network Performance

Network Performance in a Nutshell



DNS Lookup

- Every domain has its own **DNS lookup**

Initial connection

- TCP makes a **three-way handshake** → 2 roundtrips (1 with TCP Fast Open)
- **SSL** connections have a more complex handshake → +2 roundtrips (only 1 with TLS False Start or Session Resumption)

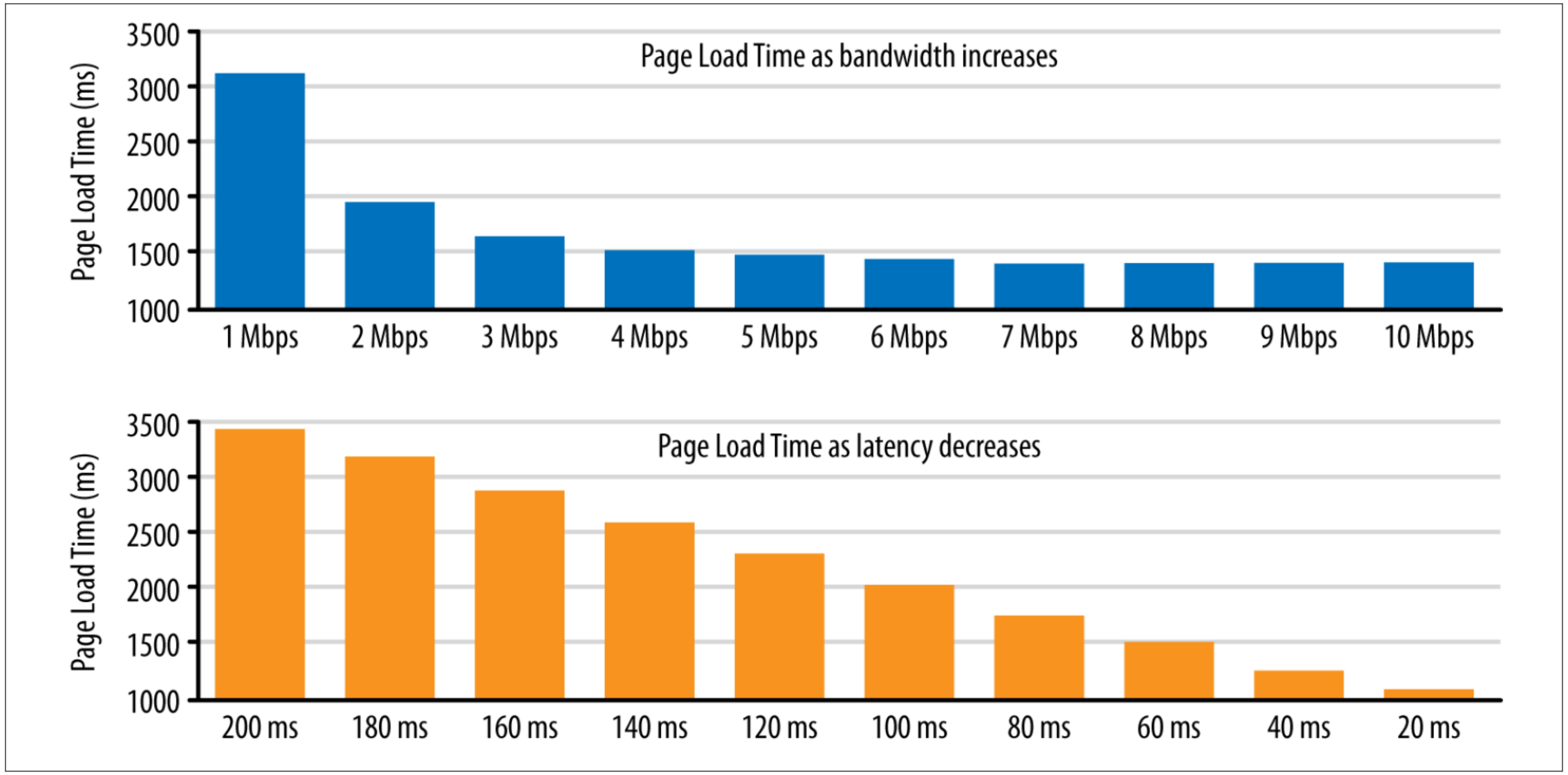
Time to First Byte

- Depends heavily on the **distance** between client and the backend
- Includes the time the backend needs to **render** → Session lookups, Database Queries, …

Content Download

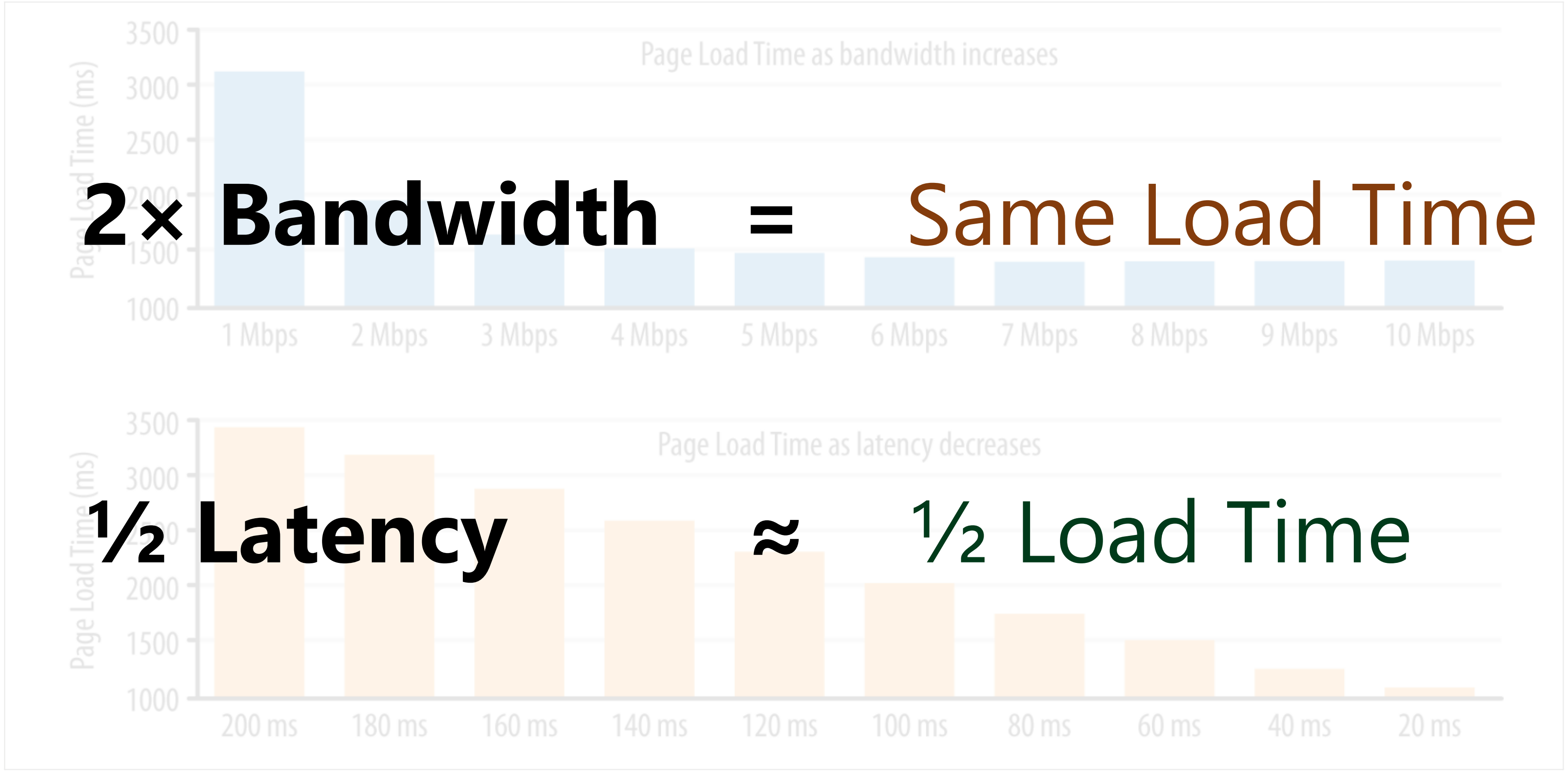
- New connections are slow (**initial congestion window** is small) → many roundtrips

Latency vs Bandwidth



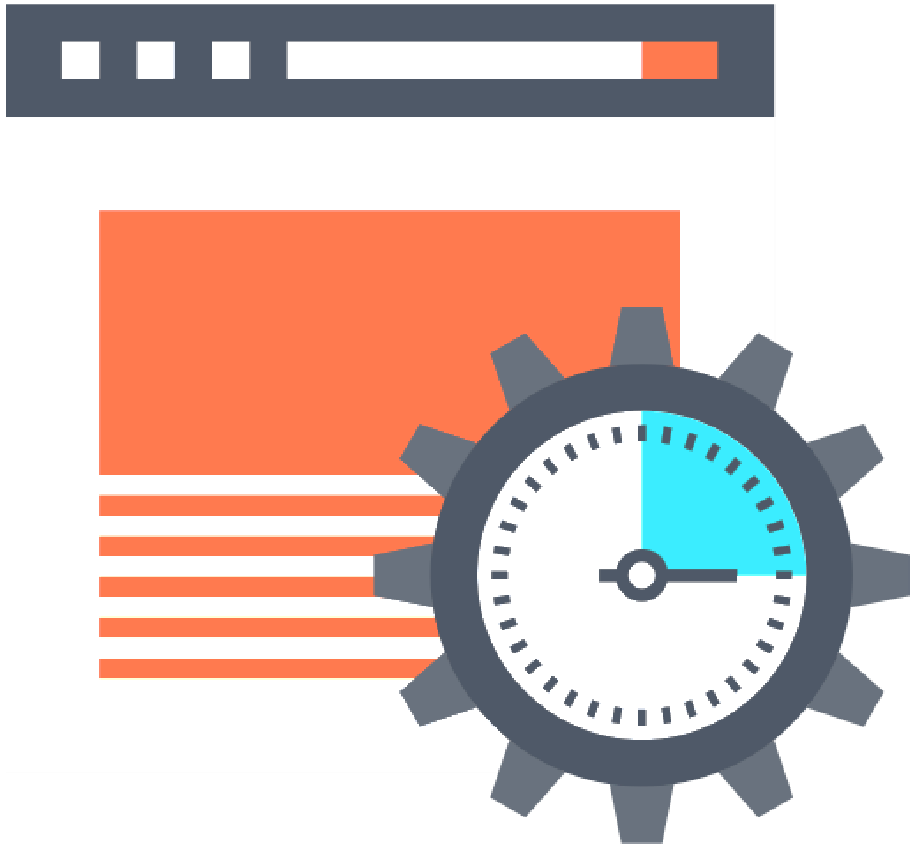
I. Grigorik, High performance browser networking. O'Reilly Media, 2013.

Latency vs Bandwidth



2× Bandwidth = Same Load Time

½ Latency ≈ ½ Load Time



How can network performance be tackled?

Common Tuning Knobs:

Avoid **redirects**, when necessary serve from CDN

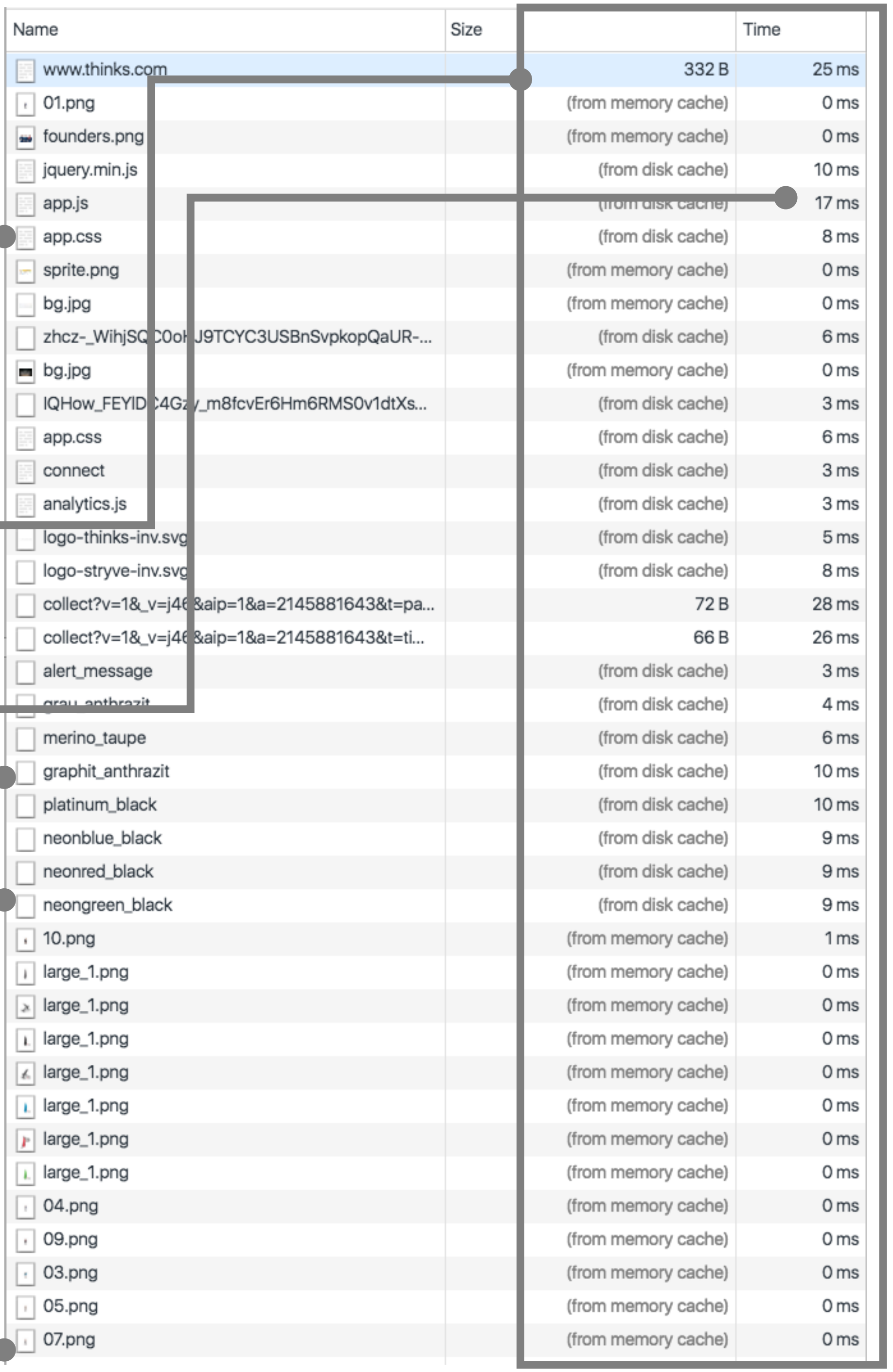
Heavy **browser and CDN caching**

Persistent backend connections and **IP anycasting**

HTTP/2 with optimized **SSL and TCP**

Gzip compression for text-based files

Minimize **DNS lookups**



Name	Size	Time
www.thinks.com	332 B	25 ms
01.png	(from memory cache)	0 ms
founders.png	(from memory cache)	0 ms
jquery.min.js	(from disk cache)	10 ms
app.js	(from disk cache)	17 ms
app.css	(from disk cache)	8 ms
sprite.png	(from memory cache)	0 ms
bg.jpg	(from memory cache)	0 ms
zhcz-_WihjSQC0oIJ9TCYC3USBnSvpkopQaUR-...	(from disk cache)	6 ms
bg.jpg	(from memory cache)	0 ms
IQHow_FEYIDC4Gzy_m8fcvEr6Hm6RMS0v1dtXs...	(from disk cache)	3 ms
app.css	(from disk cache)	6 ms
connect	(from disk cache)	3 ms
analytics.js	(from disk cache)	3 ms
logo-thinks-inv.svg	(from disk cache)	5 ms
logo-stryve-inv.svg	(from disk cache)	8 ms
collect?v=1&_v=j46&aip=1&a=2145881643&t=pa...	72 B	28 ms
collect?v=1&_v=j46&aip=1&a=2145881643&t=ti...	66 B	26 ms
alert_message	(from disk cache)	3 ms
grau_anthrazit	(from disk cache)	4 ms
merino_taupe	(from disk cache)	6 ms
graphit_anthrazit	(from disk cache)	10 ms
platinum_black	(from disk cache)	10 ms
neonblue_black	(from disk cache)	9 ms
neonred_black	(from disk cache)	9 ms
neongreen_black	(from disk cache)	9 ms
10.png	(from memory cache)	1 ms
large_1.png	(from memory cache)	0 ms
large_1.png	(from memory cache)	0 ms
large_1.png	(from memory cache)	0 ms
large_1.png	(from memory cache)	0 ms
large_1.png	(from memory cache)	0 ms
large_1.png	(from memory cache)	0 ms
large_1.png	(from memory cache)	0 ms
04.png	(from memory cache)	0 ms
09.png	(from memory cache)	0 ms
03.png	(from memory cache)	0 ms
05.png	(from memory cache)	0 ms
07.png	(from memory cache)	0 ms

Why HTTP/2 Matters

HTTP

HTTPS

3,22s

4,03s

HTTP

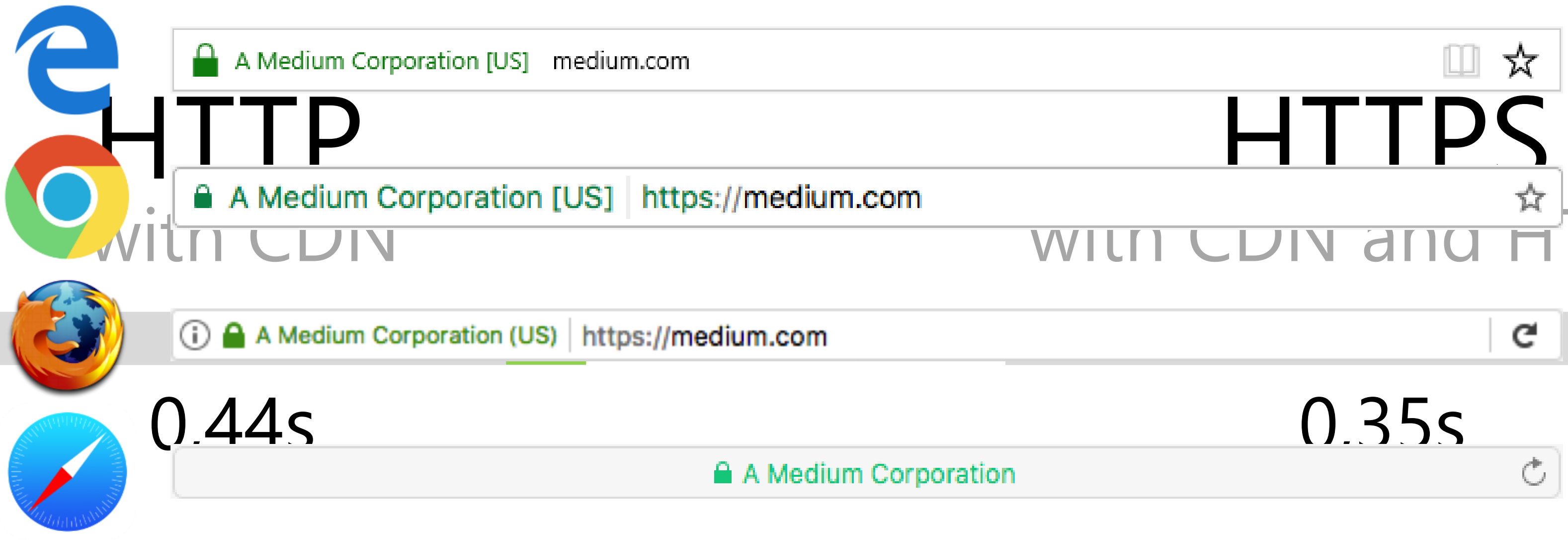
HTTPS

with CDN

with CDN and HTTP/2

0,44s

0,35s



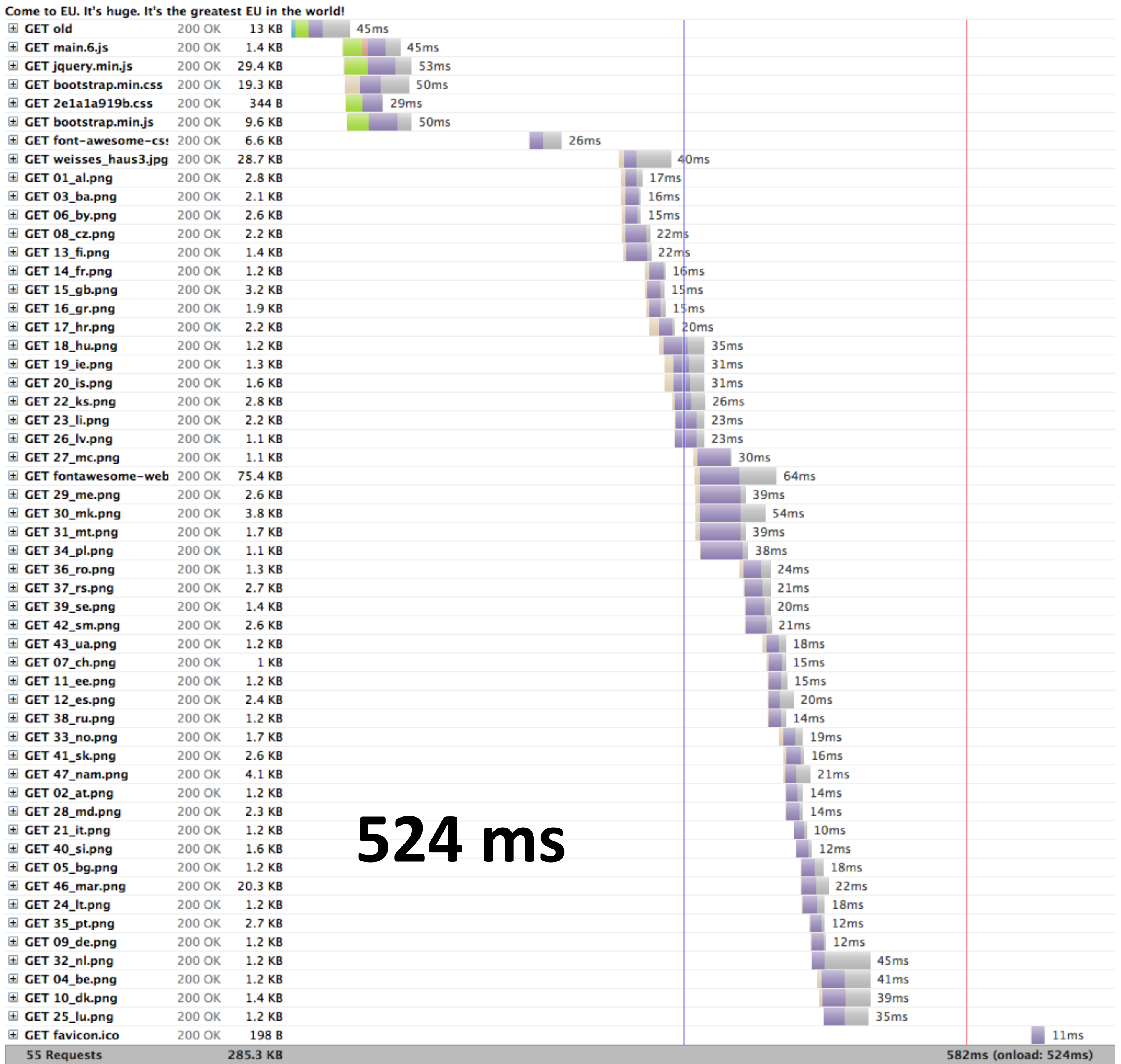
A Medium Corporation [US] medium.com

A Medium Corporation [US] https://medium.com

A Medium Corporation (US) https://medium.com

A Medium Corporation

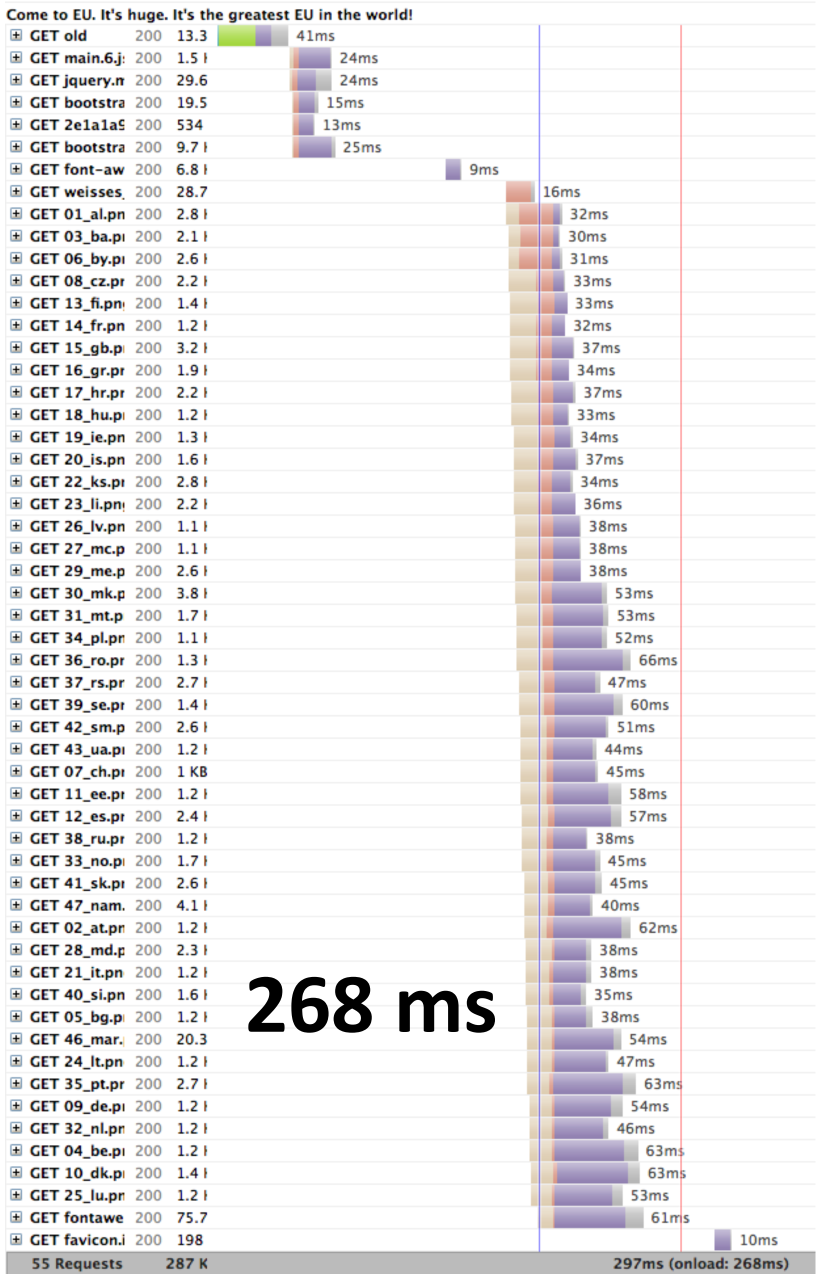
HTTP/1.1 vs HTTP/2



Come to EU. It's huge. It's the greatest EU in the world!

GET old	200 OK	13 KB	45ms
GET main.6.js	200 OK	1.4 KB	45ms
GET jquery.min.js	200 OK	29.4 KB	53ms
GET bootstrap.min.css	200 OK	19.3 KB	50ms
GET 2e1a1a919b.css	200 OK	344 B	29ms
GET bootstrap.min.js	200 OK	9.6 KB	50ms
GET font-awesome-cs	200 OK	6.6 KB	26ms
GET weisses_haus3.jpg	200 OK	28.7 KB	40ms
GET 01_al.png	200 OK	2.8 KB	17ms
GET 03_ba.png	200 OK	2.1 KB	16ms
GET 06_by.png	200 OK	2.6 KB	15ms
GET 08_cz.png	200 OK	2.2 KB	22ms
GET 13_fi.png	200 OK	1.4 KB	22ms
GET 14_fr.png	200 OK	1.2 KB	16ms
GET 15_gb.png	200 OK	3.2 KB	15ms
GET 16_gr.png	200 OK	1.9 KB	15ms
GET 17_hr.png	200 OK	2.2 KB	20ms
GET 18_hu.png	200 OK	1.2 KB	35ms
GET 19_ie.png	200 OK	1.3 KB	31ms
GET 20_is.png	200 OK	1.6 KB	31ms
GET 22_ks.png	200 OK	2.8 KB	26ms
GET 23_li.png	200 OK	2.2 KB	23ms
GET 26_lv.png	200 OK	1.1 KB	23ms
GET 27_mc.png	200 OK	1.1 KB	30ms
GET fontawesome-web	200 OK	75.4 KB	64ms
GET 29_me.png	200 OK	2.6 KB	39ms
GET 30_mk.png	200 OK	3.8 KB	54ms
GET 31_mt.png	200 OK	1.7 KB	39ms
GET 34_pl.png	200 OK	1.1 KB	38ms
GET 36_ro.png	200 OK	1.3 KB	24ms
GET 37_rs.png	200 OK	2.7 KB	21ms
GET 39_se.png	200 OK	1.4 KB	20ms
GET 42_sm.png	200 OK	2.6 KB	21ms
GET 43_ua.png	200 OK	1.2 KB	18ms
GET 07_ch.png	200 OK	1 KB	15ms
GET 11_ee.png	200 OK	1.2 KB	15ms
GET 12_es.png	200 OK	2.4 KB	20ms
GET 38_ru.png	200 OK	1.2 KB	14ms
GET 33_no.png	200 OK	1.7 KB	19ms
GET 41_sk.png	200 OK	2.6 KB	16ms
GET 47_nam.png	200 OK	4.1 KB	21ms
GET 02_at.png	200 OK	1.2 KB	14ms
GET 28_md.png	200 OK	2.3 KB	14ms
GET 21_it.png	200 OK	1.2 KB	10ms
GET 40_si.png	200 OK	1.6 KB	12ms
GET 05_bg.png	200 OK	1.2 KB	18ms
GET 46_mar.png	200 OK	20.3 KB	22ms
GET 24_lt.png	200 OK	1.2 KB	18ms
GET 35_pt.png	200 OK	2.7 KB	12ms
GET 09_de.png	200 OK	1.2 KB	12ms
GET 32_nl.png	200 OK	1.2 KB	45ms
GET 04_be.png	200 OK	1.2 KB	41ms
GET 10_dk.png	200 OK	1.4 KB	39ms
GET 25_lu.png	200 OK	1.2 KB	35ms
GET favicon.ico	200 OK	198 B	11ms
55 Requests		285.3 KB	582ms (onload: 524ms)

524 ms

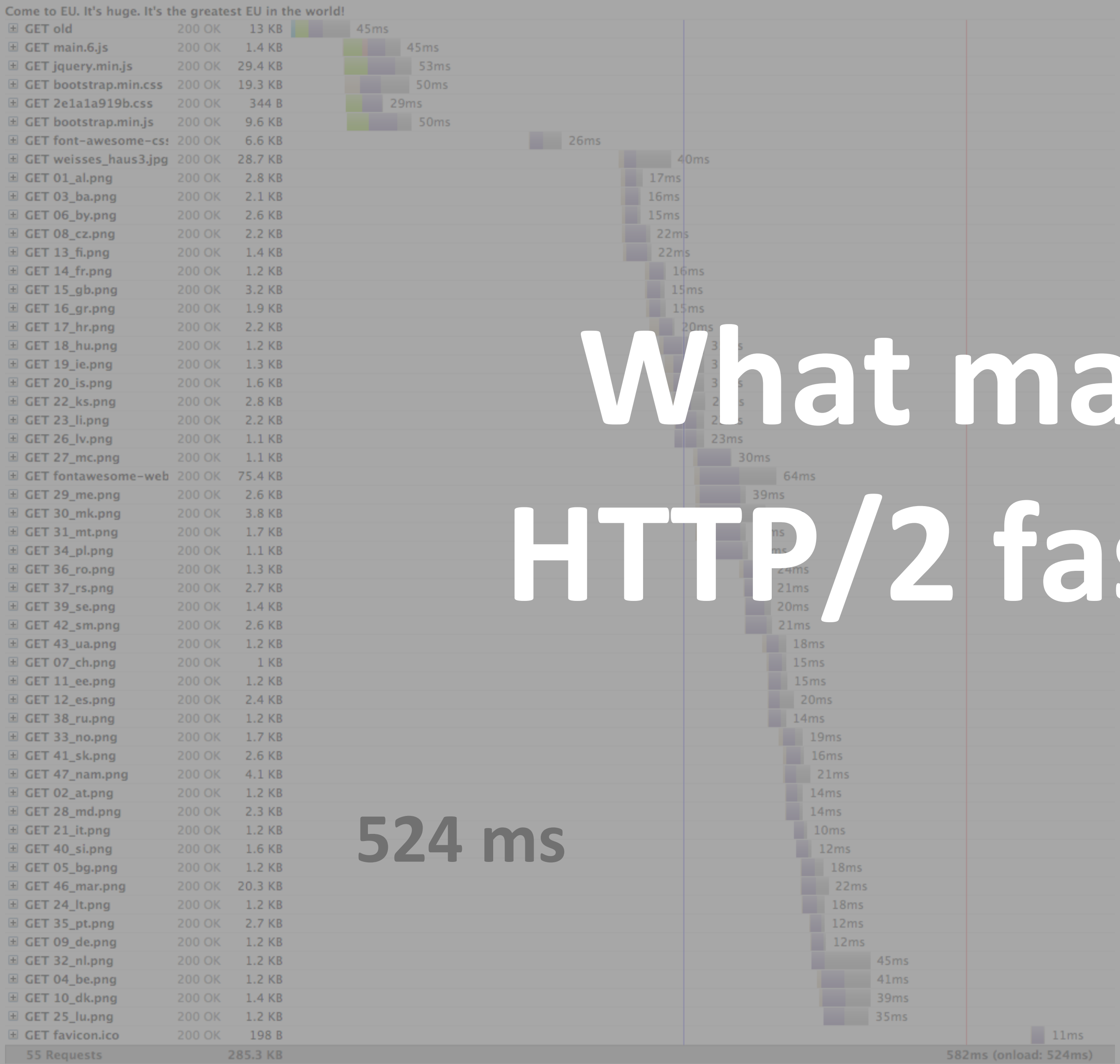


Come to EU. It's huge. It's the greatest EU in the world!

GET old	200	13.3	41ms
GET main.6.j	200	1.5	24ms
GET jquery.n	200	29.6	24ms
GET bootstra	200	19.5	15ms
GET 2e1a1a9	200	534	13ms
GET bootstra	200	9.7	25ms
GET font-aw	200	6.8	9ms
GET weisses_	200	28.7	16ms
GET 01_al.pn	200	2.8	32ms
GET 03_ba.p	200	2.1	30ms
GET 06_by.p	200	2.6	31ms
GET 08_cz.pr	200	2.2	33ms
GET 13_fi.pn	200	1.4	33ms
GET 14_fr.pn	200	1.2	32ms
GET 15_gb.p	200	3.2	37ms
GET 16_gr.pr	200	1.9	34ms
GET 17_hr.pr	200	2.2	37ms
GET 18_hu.p	200	1.2	33ms
GET 19_ie.pn	200	1.3	34ms
GET 20_is.pn	200	1.6	37ms
GET 22_ks.p	200	2.8	34ms
GET 23_li.pn	200	2.2	36ms
GET 26_lv.pn	200	1.1	38ms
GET 27_mc.p	200	1.1	38ms
GET 29_me.p	200	2.6	38ms
GET 30_mk.p	200	3.8	53ms
GET 31_mt.p	200	1.7	53ms
GET 34_pl.pr	200	1.1	52ms
GET 36_ro.pr	200	1.3	66ms
GET 37_rs.pr	200	2.7	47ms
GET 39_se.pr	200	1.4	60ms
GET 42_sm.p	200	2.6	51ms
GET 43_ua.p	200	1.2	44ms
GET 07_ch.p	200	1 KB	45ms
GET 11_ee.pr	200	1.2	58ms
GET 12_es.pr	200	2.4	57ms
GET 38_ru.pr	200	1.2	38ms
GET 33_no.p	200	1.7	45ms
GET 41_sk.p	200	2.6	45ms
GET 47_nam.	200	4.1	40ms
GET 02_at.pr	200	1.2	62ms
GET 28_md.p	200	2.3	38ms
GET 21_it.pn	200	1.2	38ms
GET 40_si.pn	200	1.6	35ms
GET 05_bg.p	200	1.2	38ms
GET 46_mar.	200	20.3	54ms
GET 24_lt.pn	200	1.2	47ms
GET 35_pt.pr	200	2.7	63ms
GET 09_de.p	200	1.2	54ms
GET 32_nl.pr	200	1.2	46ms
GET 04_be.p	200	1.2	63ms
GET 10_dk.p	200	1.4	63ms
GET 25_lu.pr	200	1.2	53ms
GET fontawe	200	75.7	61ms
GET favicon.i	200	198	10ms
55 Requests		287 K	297ms (onload: 268ms)

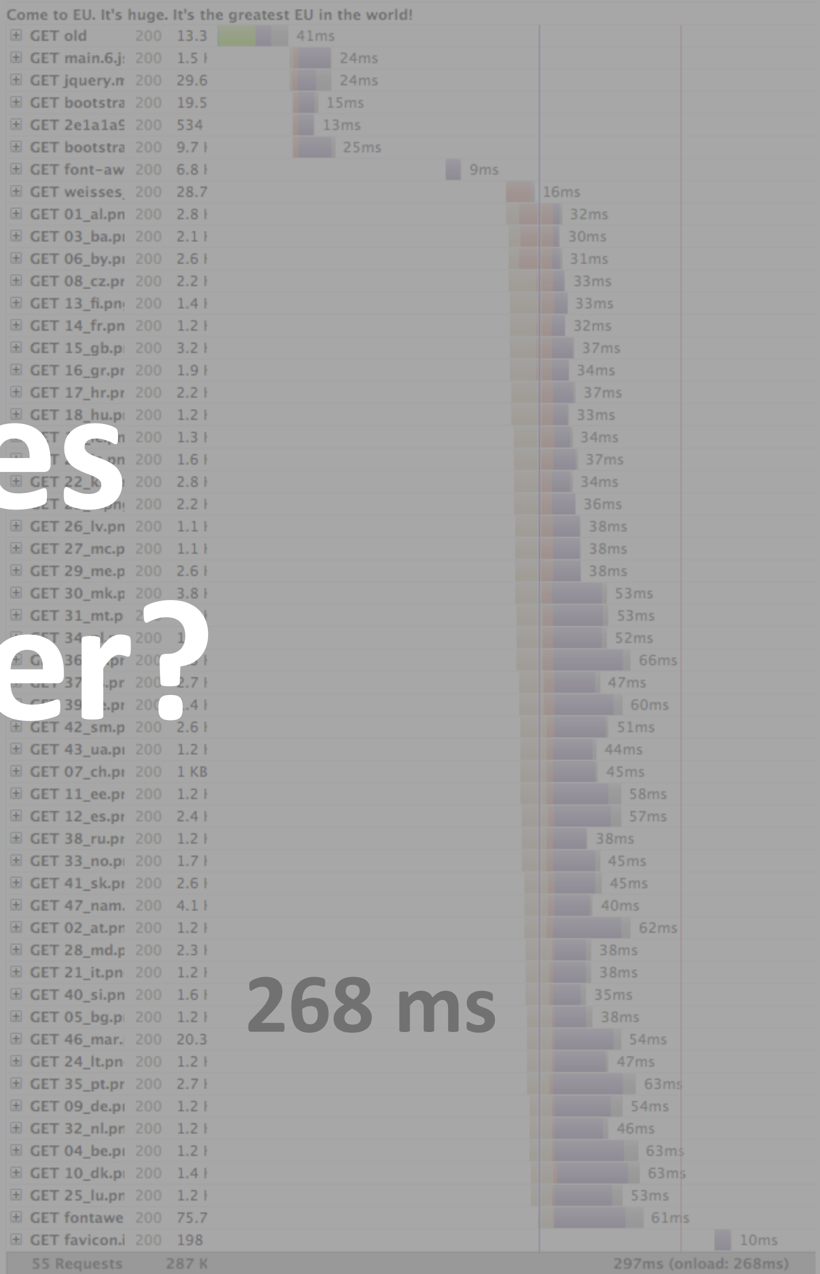
268 ms

HTTP/1.1 vs HTTP/2



```
Come to EU. It's huge. It's the greatest EU in the world!
GET old                 200 OK   13 KB     45ms
GET main.6.js           200 OK   1.4 KB    45ms
GET jquery.min.js       200 OK   29.4 KB   53ms
GET bootstrap.min.css   200 OK   19.3 KB   50ms
GET 2e1a1a919b.css      200 OK   344 B     29ms
GET bootstrap.min.js    200 OK   9.6 KB    50ms
GET font-awesome-css    200 OK   6.6 KB    26ms
GET weisses_haus3.jpg   200 OK   28.7 KB   40ms
GET 01_al.png           200 OK   2.8 KB    17ms
GET 03_ba.png           200 OK   2.1 KB    16ms
GET 06_by.png           200 OK   2.6 KB    15ms
GET 08_cz.png           200 OK   2.2 KB    22ms
GET 13_fi.png           200 OK   1.4 KB    22ms
GET 14_fr.png           200 OK   1.2 KB    16ms
GET 15_gb.png           200 OK   3.2 KB    15ms
GET 16_gr.png           200 OK   1.9 KB    15ms
GET 17_hr.png           200 OK   2.2 KB    20ms
GET 18_hu.png           200 OK   1.2 KB    3
GET 19_ie.png           200 OK   1.3 KB    3
GET 20_is.png           200 OK   1.6 KB    3
GET 22_ks.png           200 OK   2.8 KB    2
GET 23_li.png           200 OK   2.2 KB    2
GET 26_lv.png           200 OK   1.1 KB    23ms
GET 27_mc.png           200 OK   1.1 KB    30ms
GET fontawesome-web     200 OK   75.4 KB   64ms
GET 29_me.png           200 OK   2.6 KB    39ms
GET 30_mk.png           200 OK   3.8 KB
GET 31_mt.png           200 OK   1.7 KB    ms
GET 34_pl.png           200 OK   1.1 KB    ms
GET 36_ro.png           200 OK   1.3 KB    24ms
GET 37_rs.png           200 OK   2.7 KB    21ms
GET 39_se.png           200 OK   1.4 KB    20ms
GET 42_sm.png           200 OK   2.6 KB    21ms
GET 43_ua.png           200 OK   1.2 KB    18ms
GET 07_ch.png           200 OK   1 KB      15ms
GET 11_ee.png           200 OK   1.2 KB    15ms
GET 12_es.png           200 OK   2.4 KB    20ms
GET 38_ru.png           200 OK   1.2 KB    14ms
GET 33_no.png           200 OK   1.7 KB    19ms
GET 41_sk.png           200 OK   2.6 KB    16ms
GET 47_nam.png          200 OK   4.1 KB    21ms
GET 02_at.png           200 OK   1.2 KB    14ms
GET 28_md.png           200 OK   2.3 KB    14ms
GET 21_it.png           200 OK   1.2 KB    10ms
GET 40_si.png           200 OK   1.6 KB    12ms
GET 05_bg.png           200 OK   1.2 KB    18ms
GET 46_mar.png          200 OK   20.3 KB   22ms
GET 24_lt.png           200 OK   1.2 KB    18ms
GET 35_pt.png           200 OK   2.7 KB    12ms
GET 09_de.png           200 OK   1.2 KB    12ms
GET 32_nl.png           200 OK   1.2 KB    45ms
GET 04_be.png           200 OK   1.2 KB    41ms
GET 10_dk.png           200 OK   1.4 KB    39ms
GET 25_lu.png           200 OK   1.2 KB    35ms
GET favicon.ico         200 OK   198 B     11ms
55 Requests                      285.3 KB  582ms (onload: 524ms)
```

524 ms



```
Come to EU. It's huge. It's the greatest EU in the world!
GET old        200  13.3  41ms
GET main.6.j:  200  1.5   24ms
GET jquery.n   200  29.6  24ms
GET bootstra   200  19.5  15ms
GET 2e1a1a9    200  534   13ms
GET bootstra   200  9.7   25ms
GET font-aw    200  6.8   9ms
GET weisses_   200  28.7  16ms
GET 01_al.pn   200  2.8   32ms
GET 03_ba.p    200  2.1   30ms
GET 06_by.p    200  2.6   31ms
GET 08_cz.pn   200  2.2   33ms
GET 13_fi.pn   200  1.4   33ms
GET 14_fr.pn   200  1.2   32ms
GET 15_gb.p    200  3.2   37ms
GET 16_gr.p    200  1.9   34ms
GET 17_hr.p    200  2.2   37ms
GET 18_hu.p    200  1.2   33ms
               200  1.3   34ms
               200  1.6   37ms
GET 22_k       200  2.8   34ms
               200  2.2   36ms
GET 26_lv.pn   200  1.1   38ms
GET 27_mc.p    200  1.1   38ms
GET 29_me.p    200  2.6   38ms
GET 30_mk.p    200  3.8   53ms
GET 31_mt.p    2          53ms
               200  1     52ms
               200        66ms
         .pr   200  2.7   47ms
         .p    200  .4    60ms
GET 42_sm.p    200  2.6   51ms
GET 43_ua.p    200  1.2   44ms
GET 07_ch.p    200  1 KB  45ms
GET 11_ee.p    200  1.2   58ms
GET 12_es.p    200  2.4   57ms
GET 38_ru.p    200  1.2   38ms
GET 33_no.p    200  1.7   45ms
GET 41_sk.p    200  2.6   45ms
GET 47_nam.    200  4.1   40ms
GET 02_at.p    200  1.2   62ms
GET 28_md.p    200  2.3   38ms
GET 21_it.pn   200  1.2   38ms
GET 40_si.pn   200  1.6   35ms
GET 05_bg.p    200  1.2   38ms
GET 46_mar.    200  20.3  54ms
GET 24_lt.pn   200  1.2   47ms
GET 35_pt.pr   200  2.7   63ms
GET 09_de.p    200  1.2   54ms
GET 32_nl.pr   200  1.2   46ms
GET 04_be.p    200  1.2   63ms
GET 10_dk.p    200  1.4   63ms
GET 25_lu.pr   200  1.2   53ms
GET fontawe    200  75.7  61ms
GET favicon.i  200  198   10ms
55 Requests         287 K 297ms (onload: 268ms)
```

268 ms

What makes HTTP/2 faster?

Optimizations in HTTP/2

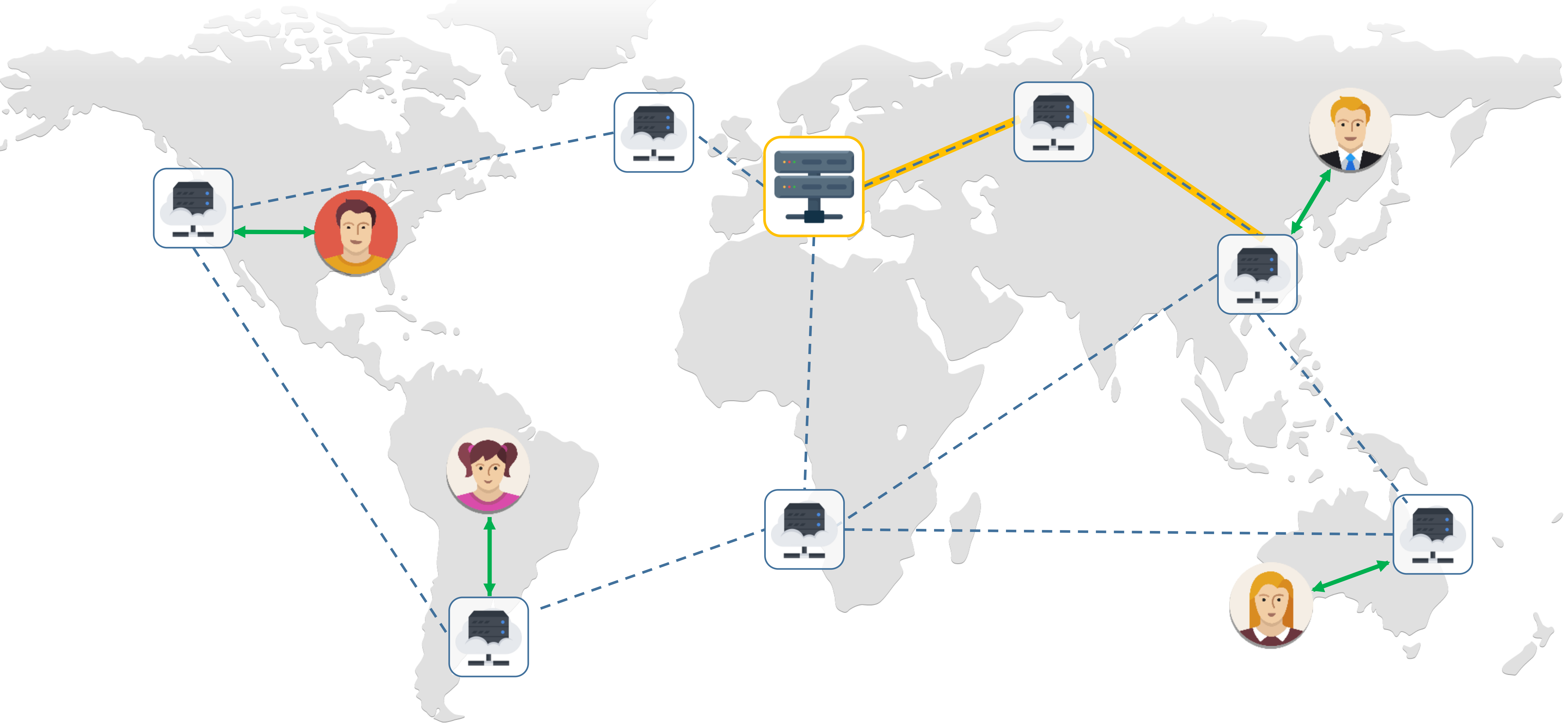
Multiplexing
(1 Connection)

Server
Push

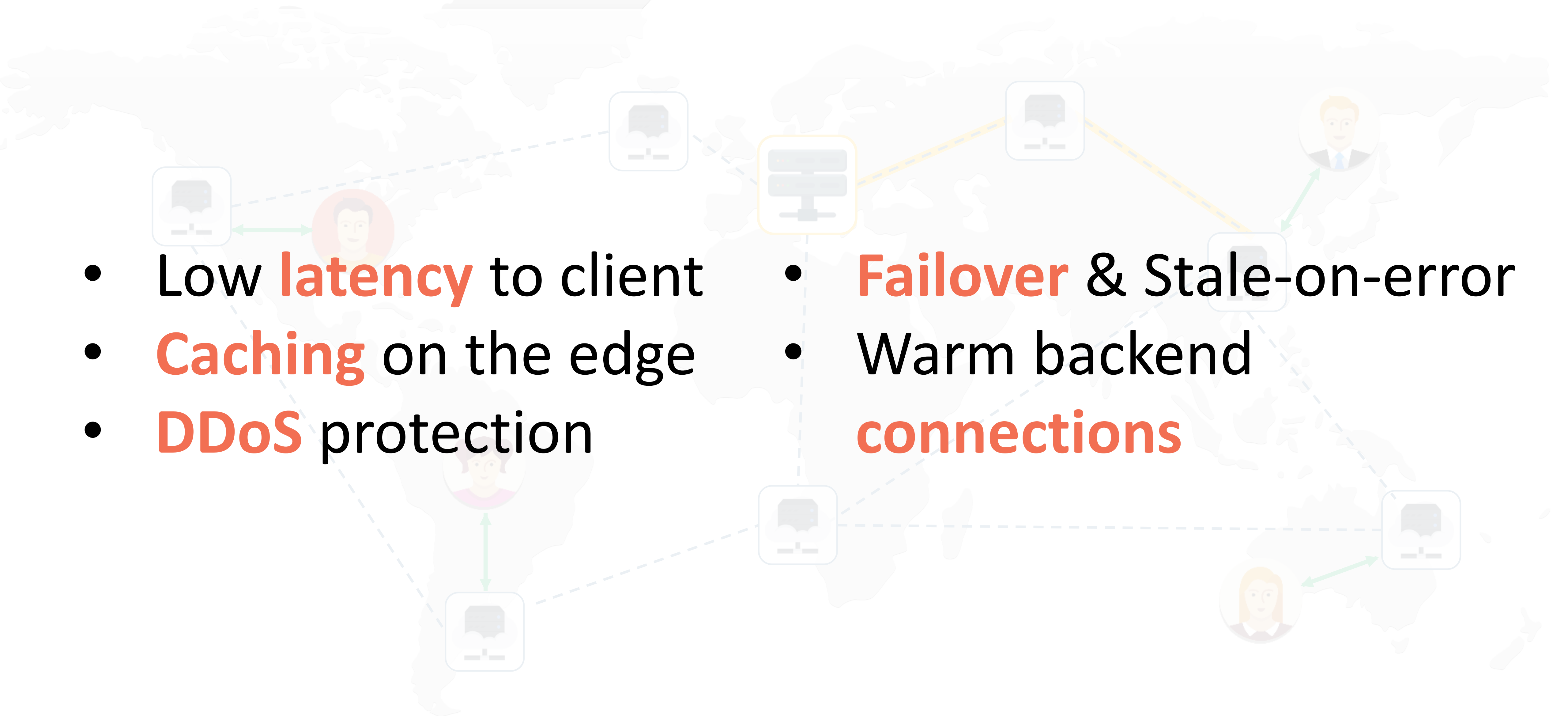
Resource
Prioritization

Header
Compression

Adding a Content Delivery Network (CDN)



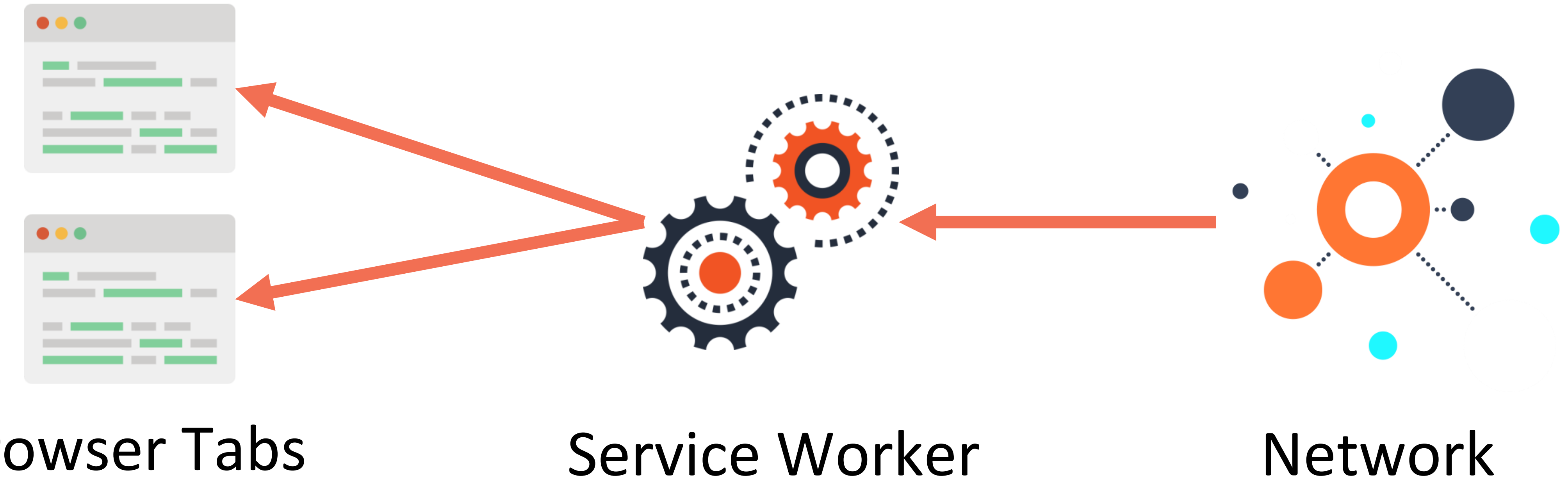
Adding a Content Delivery Network (CDN)



- Low latency to client
- Caching on the edge
- DDoS protection

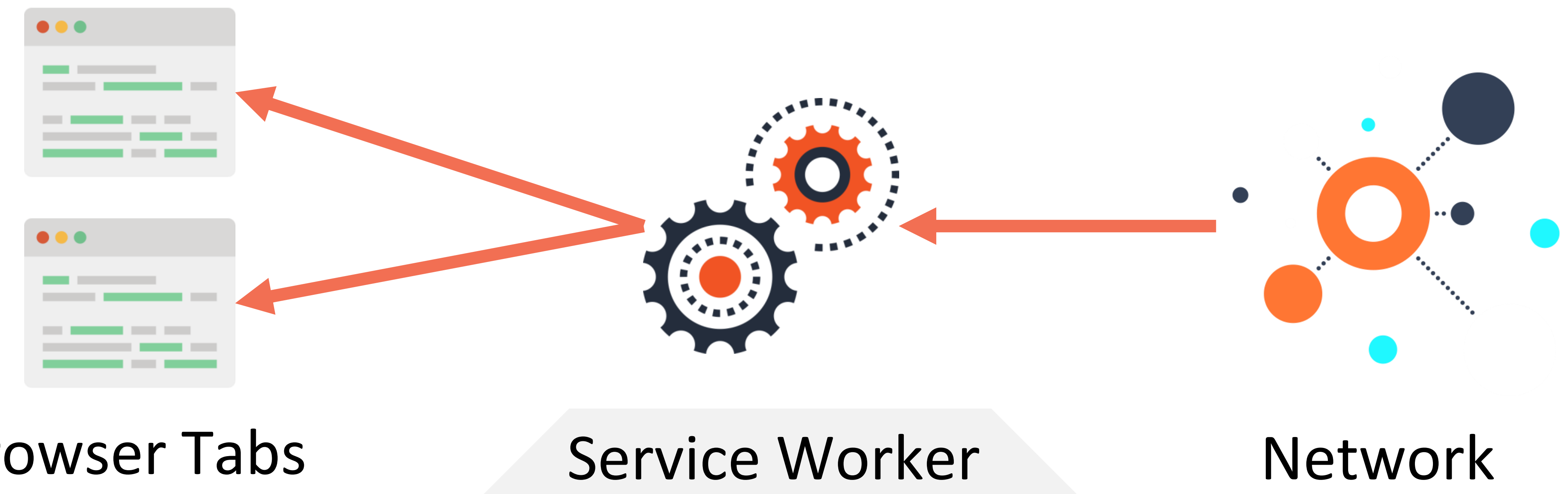
- Failover & Stale-on-error
- Warm backend connections

Hooking Into the Network: Service Workers



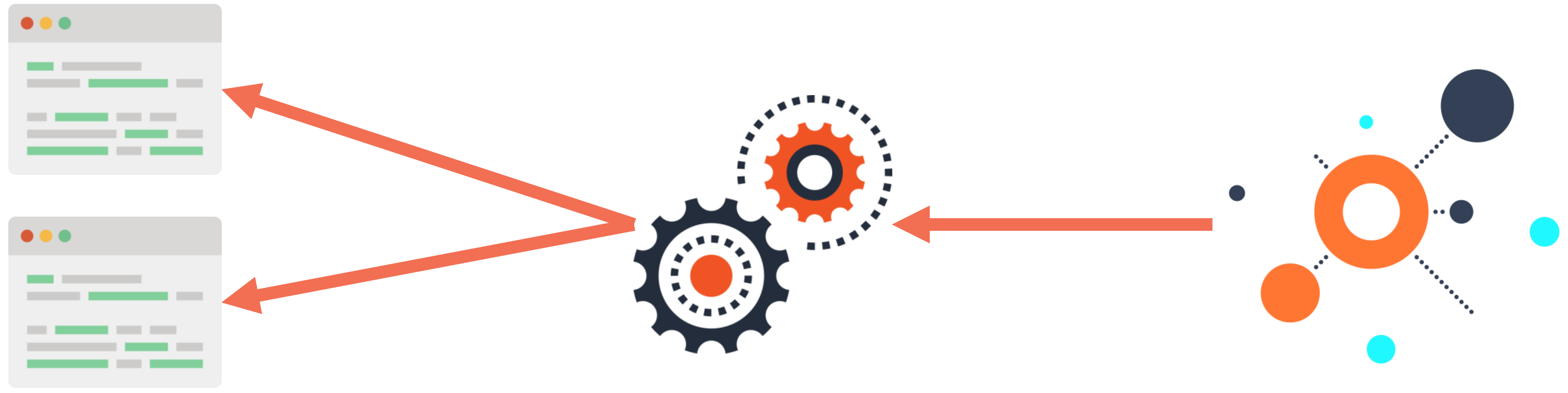
```
navigator.serviceWorker.register('/sw.js‘);
//In sw.js:
self.addEventListener('fetch', (event) => {}); //…
```

Hooking Into the Network: Service Workers



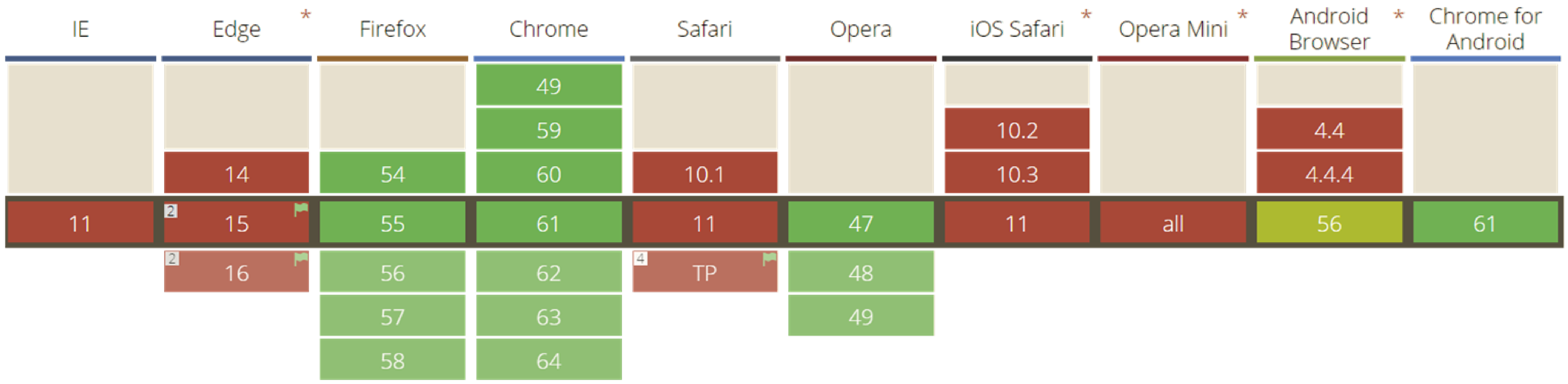
- **Cache** Data (CacheStorage)
- **Store** Data (IndexedDB)
- Receive **Push**
- Respond when **Offline**

Hooking Into the Network: Service Workers



- **Rewrite** HTTP Requests
- **Sync** Data in Background
- Hide **Flaky Connectivity** from the User

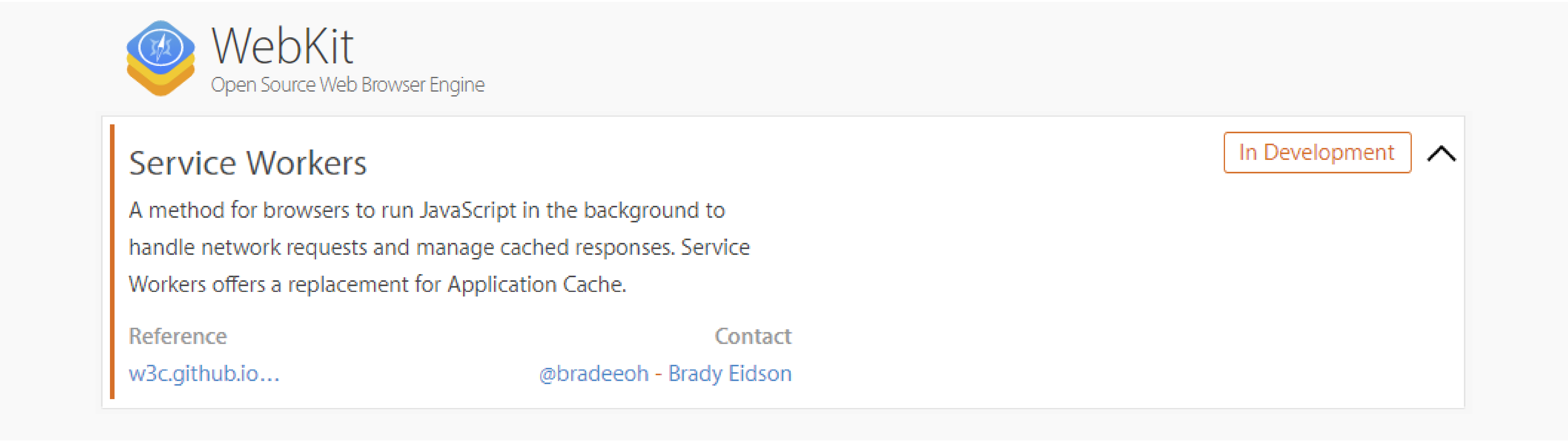
Browser Support for Service Workers



IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49						
			59			10.2		4.4	
	14	54	60	10.1		10.3		4.4.4	
11	[2] 15	55	61	11	47	11	all	56	61
	[2] 16	56	62	[4] TP	48				
		57	63		49				
		58	64						

Supported by **75%** of browsers.

Browser Support for Service Workers



Service Workers

In Development

A method for browsers to run JavaScript in the background to handle network requests and manage cached responses. Service Workers offers a replacement for Application Cache.

Reference: w3c.github.io...

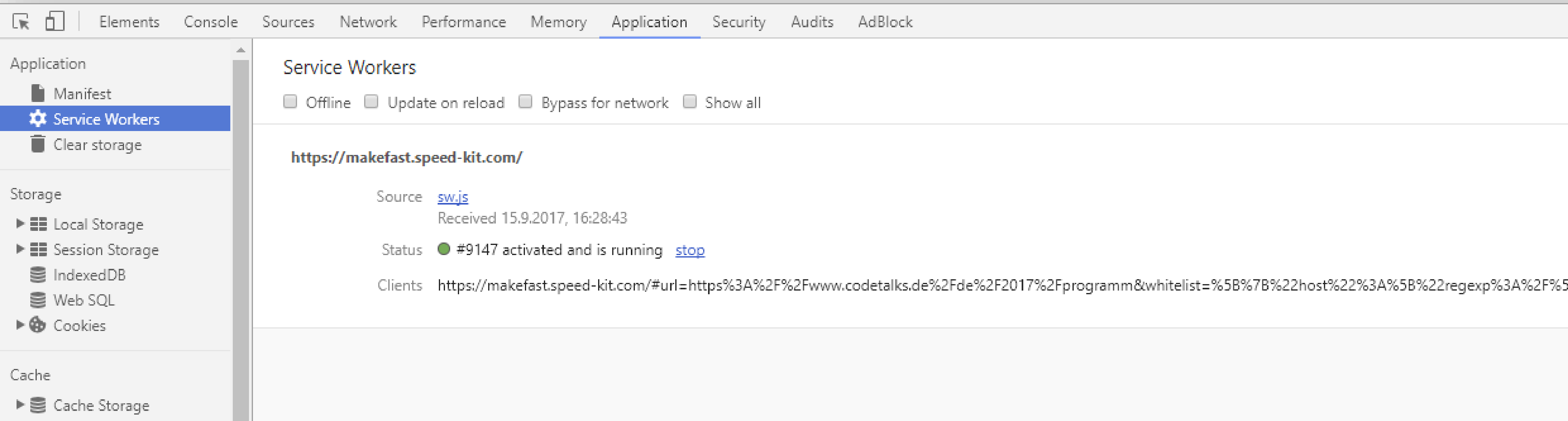
Contact: @bradeeoh - Brady Eidson

Safari: In Development

Edge: Implemented, but Toggled

Implementing Service Workers

- Requires **SSL**
- Hard to **debug**
- Sw.js must be served top-level (**root scope**)



Service Workers

https://makefast.speed-kit.com/

Source sw.js
Received 15.9.2017, 16:28:43

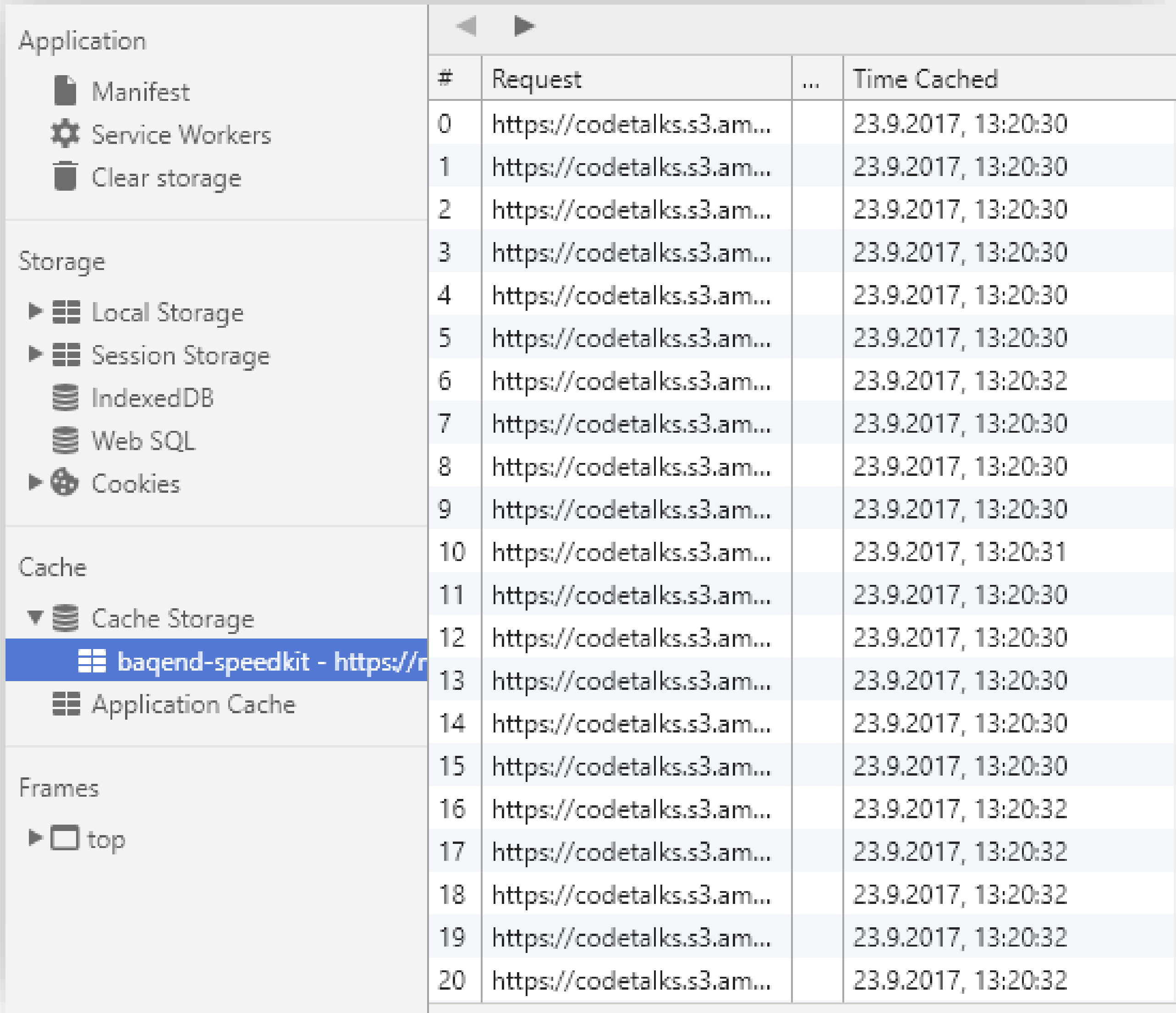
Status ● #9147 activated and is running

Clients https://makefast.speed-kit.com/#url=https%3A%2F%2Fwww.codetalks.de%2Fde%2F2017%2Fprogramm&whitelist=%5B%7B%22host%22%3A%5B%22regexp%3A%2F%5

Major Challenge: Cache Coherence

- Cache just stores **(Req, Res)-Pairs**
- HTTP **browser cache** always exists, too

→ App decides when to **evict** cache



Application
Manifest
Service Workers
Clear storage

Storage
Local Storage
Session Storage
IndexedDB
Web SQL
Cookies

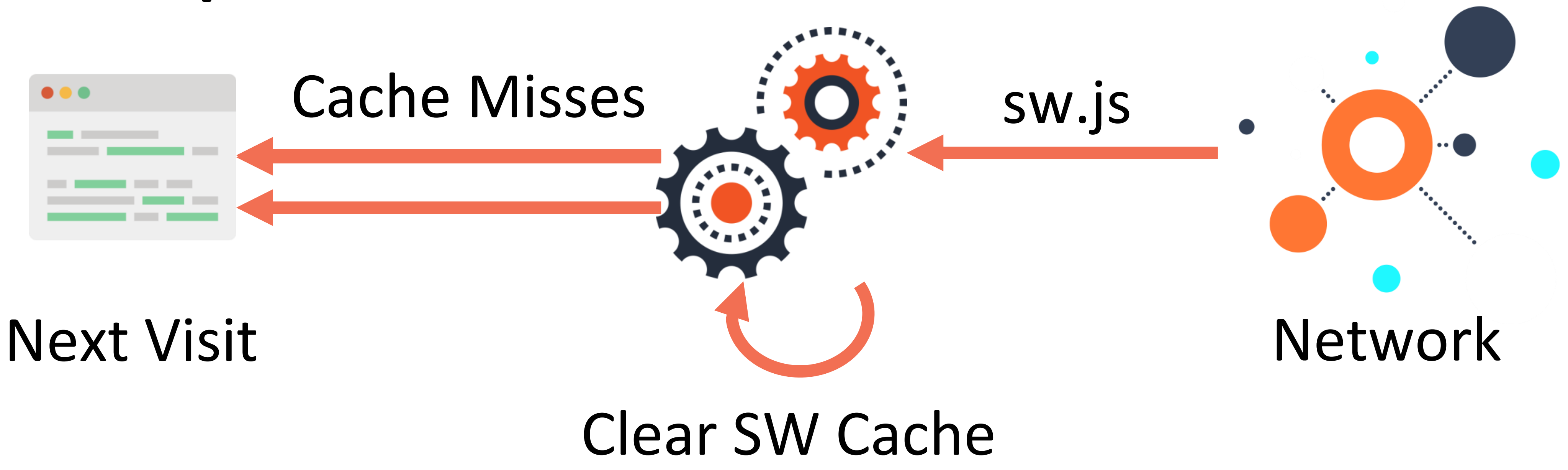
Cache
Cache Storage
baqend-speedkit - https://r
Application Cache

Frames
top

#	Request	...	Time Cached
0	https://codetalks.s3.am...		23.9.2017, 13:20:30
1	https://codetalks.s3.am...		23.9.2017, 13:20:30
2	https://codetalks.s3.am...		23.9.2017, 13:20:30
3	https://codetalks.s3.am...		23.9.2017, 13:20:30
4	https://codetalks.s3.am...		23.9.2017, 13:20:30
5	https://codetalks.s3.am...		23.9.2017, 13:20:30
6	https://codetalks.s3.am...		23.9.2017, 13:20:32
7	https://codetalks.s3.am...		23.9.2017, 13:20:30
8	https://codetalks.s3.am...		23.9.2017, 13:20:30
9	https://codetalks.s3.am...		23.9.2017, 13:20:30
10	https://codetalks.s3.am...		23.9.2017, 13:20:31
11	https://codetalks.s3.am...		23.9.2017, 13:20:30
12	https://codetalks.s3.am...		23.9.2017, 13:20:30
13	https://codetalks.s3.am...		23.9.2017, 13:20:30
14	https://codetalks.s3.am...		23.9.2017, 13:20:30
15	https://codetalks.s3.am...		23.9.2017, 13:20:30
16	https://codetalks.s3.am...		23.9.2017, 13:20:32
17	https://codetalks.s3.am...		23.9.2017, 13:20:32
18	https://codetalks.s3.am...		23.9.2017, 13:20:32
19	https://codetalks.s3.am...		23.9.2017, 13:20:32
20	https://codetalks.s3.am...		23.9.2017, 13:20:32

Major Challenge: Cache Coherence

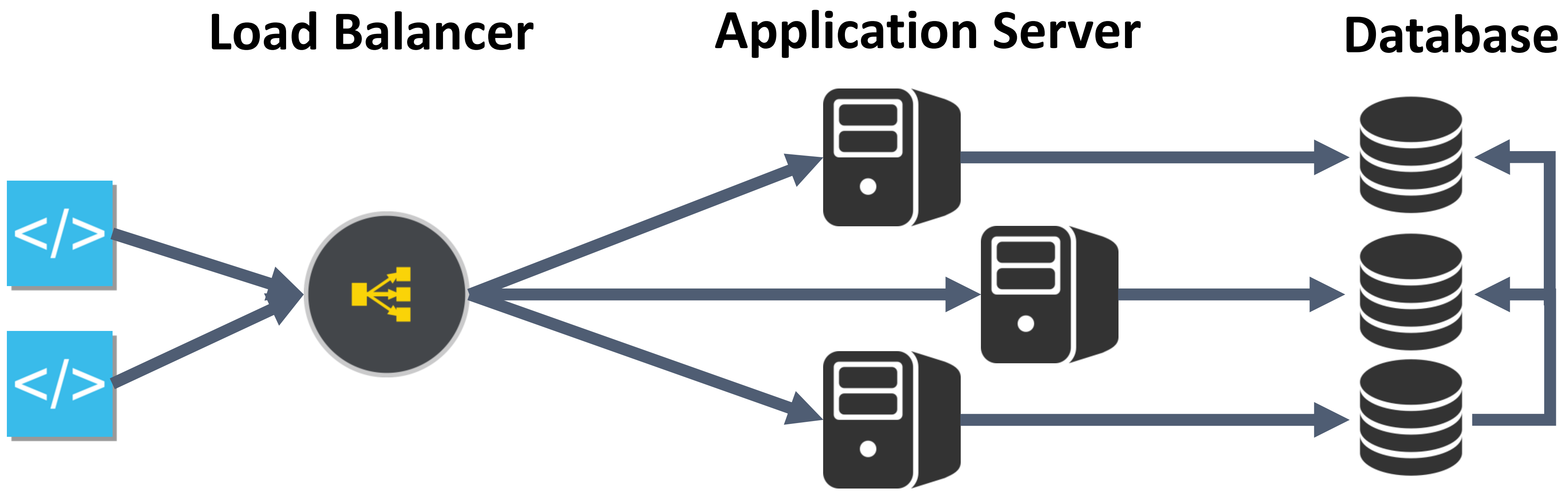
Usual pattern:



→ Does not improve inital **page load time**

3. Backend Performance

Backend Performance in a Nutshell

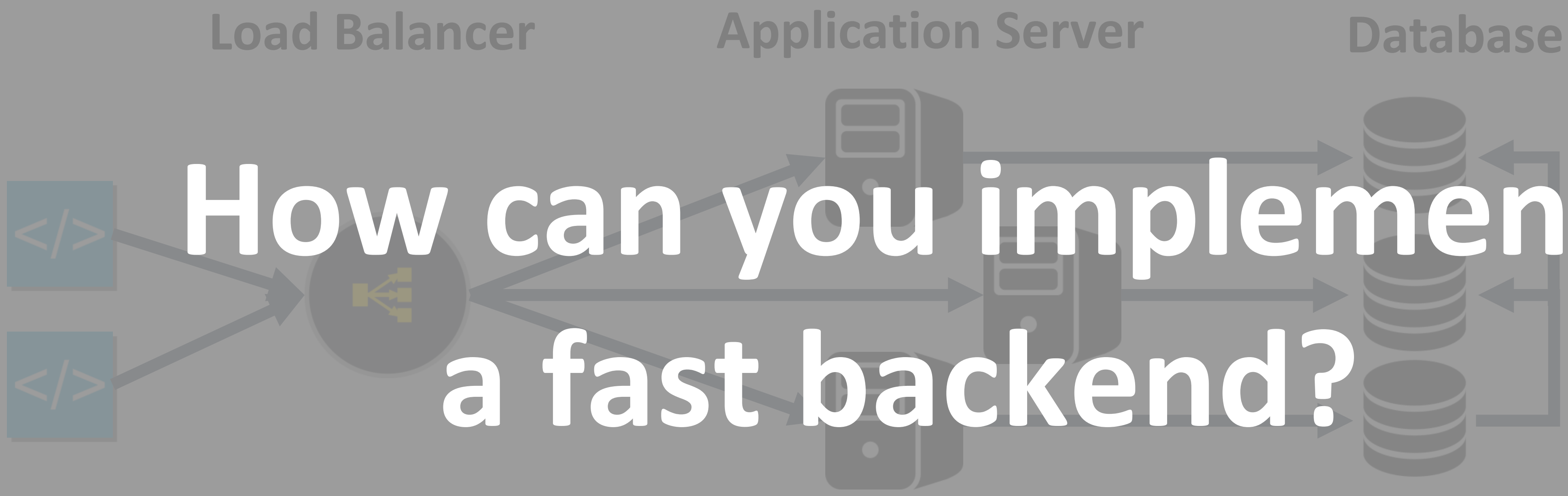


- Load Balancing
- Auto-scaling
- Failover

- Stateless Sessions
- Efficient Code & IO

- Horizontally scalable databases (e.g. “NoSQL”)
  - Replication
  - Sharding
  - Failover

Backend Performance in a Nutshell



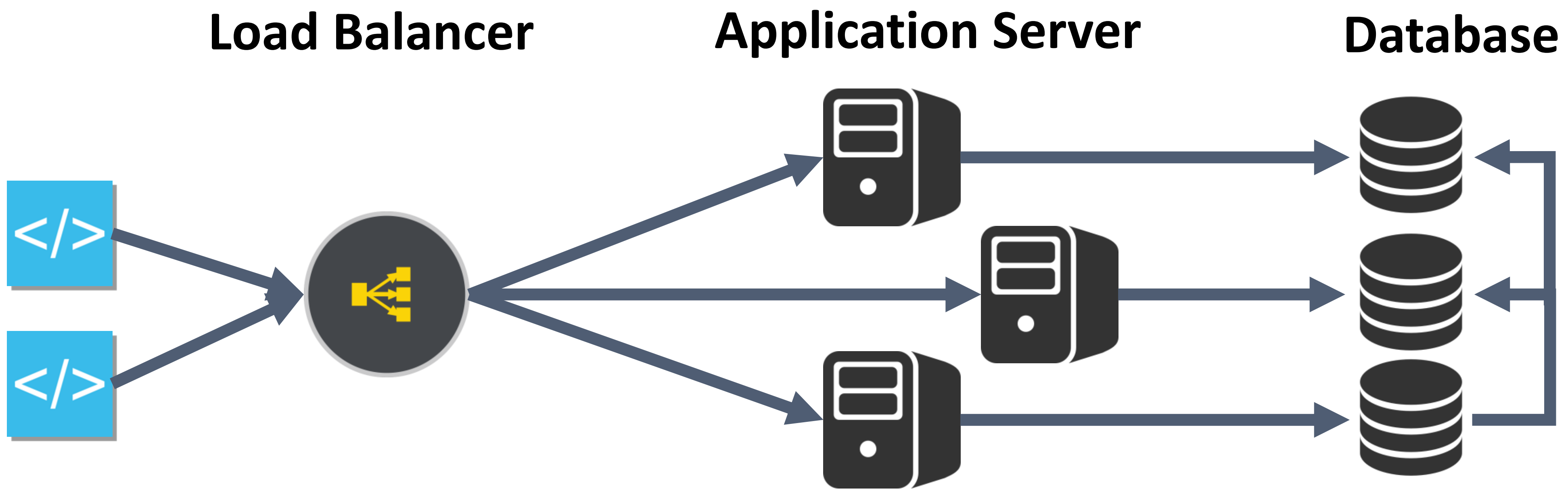
How can you implement a fast backend?

- Load Balancing
- Auto-scaling
- Failover

- Stateless Sessions
- Minimize shared state
- Efficient Code & IO

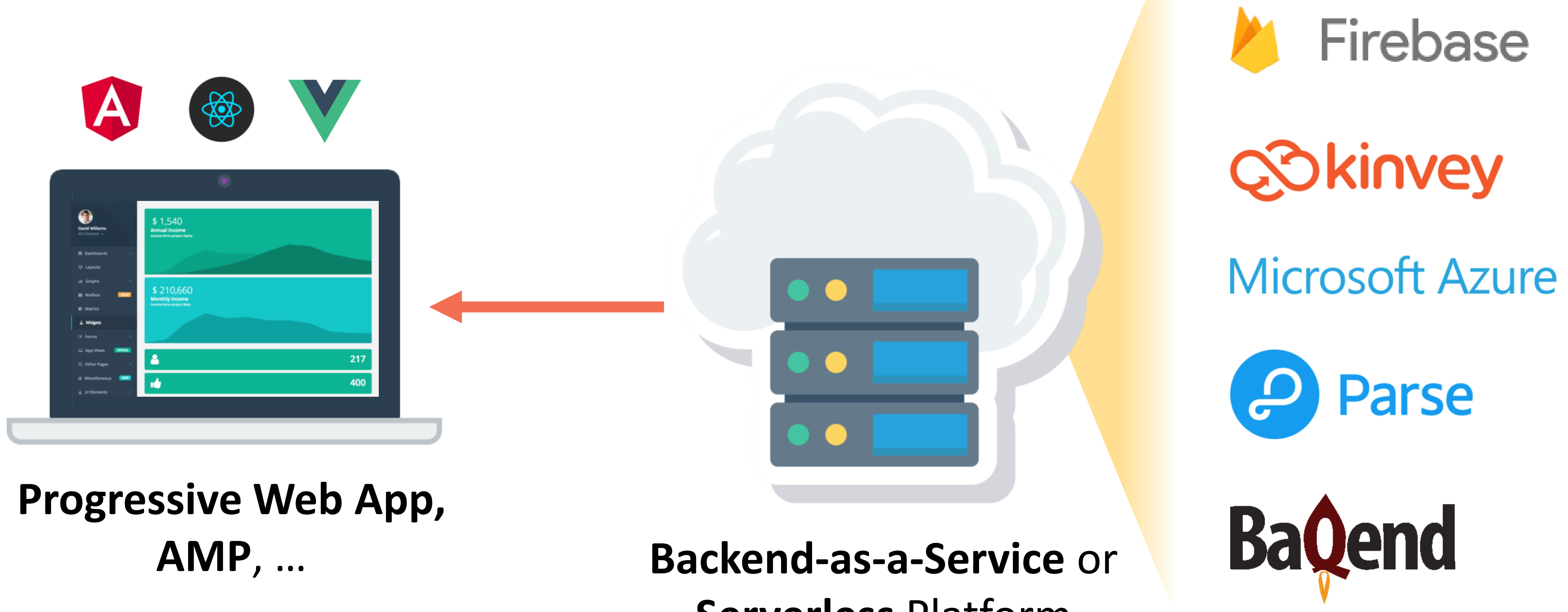
- Horizontally scalable databases (e.g. "NoSQL")
  - Replication
  - Sharding
  - Failover

Option 1: Build a Scalable Architecture



Windows Azure, amazon web services, VoltDB, mongoDB, Cassandra, redis, Open Telekom Cloud, Google Cloud, Amazon DynamoDB, riak, amazon web services S3, HBASE

Option 2: Use a Backend Platform



Progressive Web App, AMP, …

Backend-as-a-Service or **Serverless** Platform

Read More on Backend Performance

Articles on medium.baqend.com

Applause from you, Hannes Kuhlmann, and 112 others

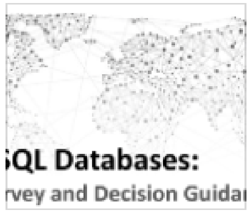
Scalable Stream Processing: A Survey of Storm, Samza, Spark and Flink

With this article, we would like to share our insights on real-time data processing we gained building Baqend. This is an updated version...

Applause from you, Hannes Kuhlmann, and 541 others

NoSQL Databases: a Survey and Decision Guidance

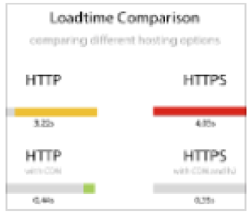
Together with our colleagues at the University of Hamburg, we — that is Felix Gessert, Wolfram Wingerath, Steffen Friedrich and Norbert...



Applause from you, Wolfram Wingerath, and 26 others

Web Performance in a Nutshell: HTTP/2, CDNs and Browser Caching

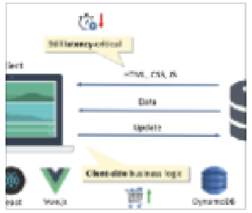
Successful websites need to be fast, scalable and secure. In this article we survey the state of the art of high-performance websites, in...



Applause from you, Hannes Kuhlmann, and 64 others

Lessons Learned Building a Backend-as-a-Service: A Technical Deep Dive

In this post we share our technical learnings from building a multi-tenant Backend-as-a-Service (BaaS). We cover how a BaaS works, how it...



Applause from you, DISTRIBOOTED, and 208 others

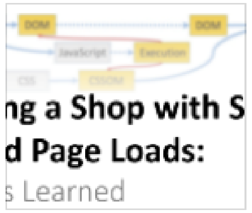
The AWS and MongoDB Infrastructure of Parse: Lessons Learned

This is the extended form of a comment that got some interest on Hackernews. After a grace period of one year, Parse is now offline. This...

Applause from you, Malte Lauenroth, and 312 others

Building a Shop with Sub-Second Page Loads: Lessons Learned

Here is the story of how we leveraged research on web-caching and NoSQL systems to prepare a webshop for hundreds of thousands of visitors...



Now, we have a PWA, HTTP/2, etc.

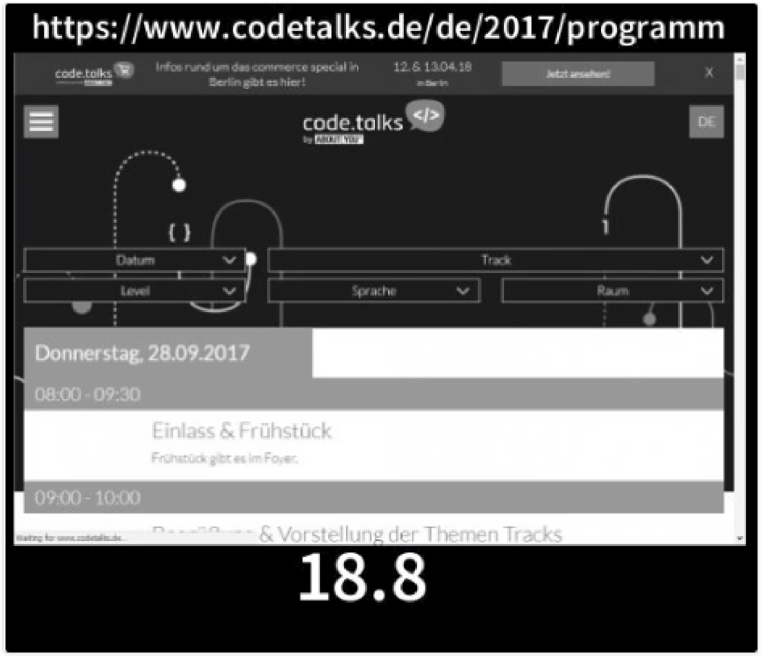
How do we measure web performance?

Page Speed Analyzer

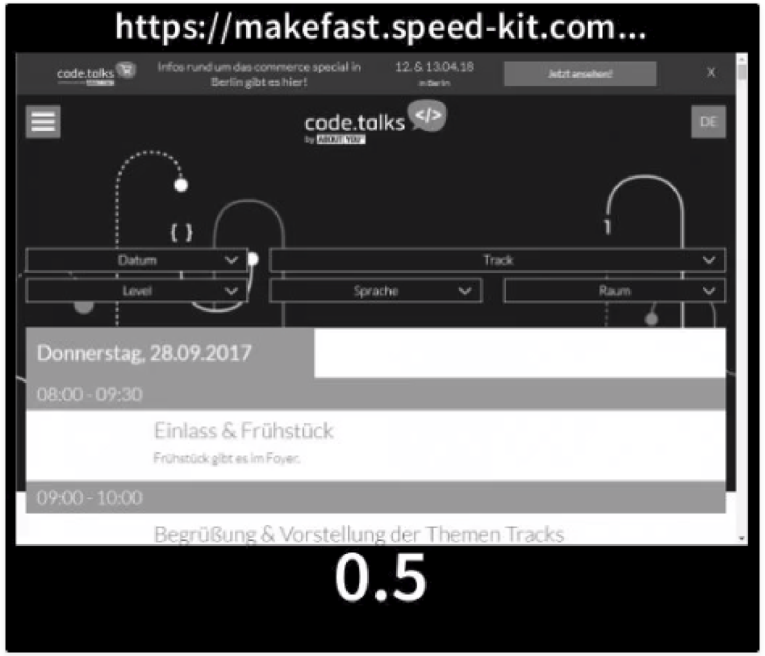
https://www.codetalks.de/de/2017/programm Go

Domains	Requests	Response Size
10	23	965.17 KB

Your Website

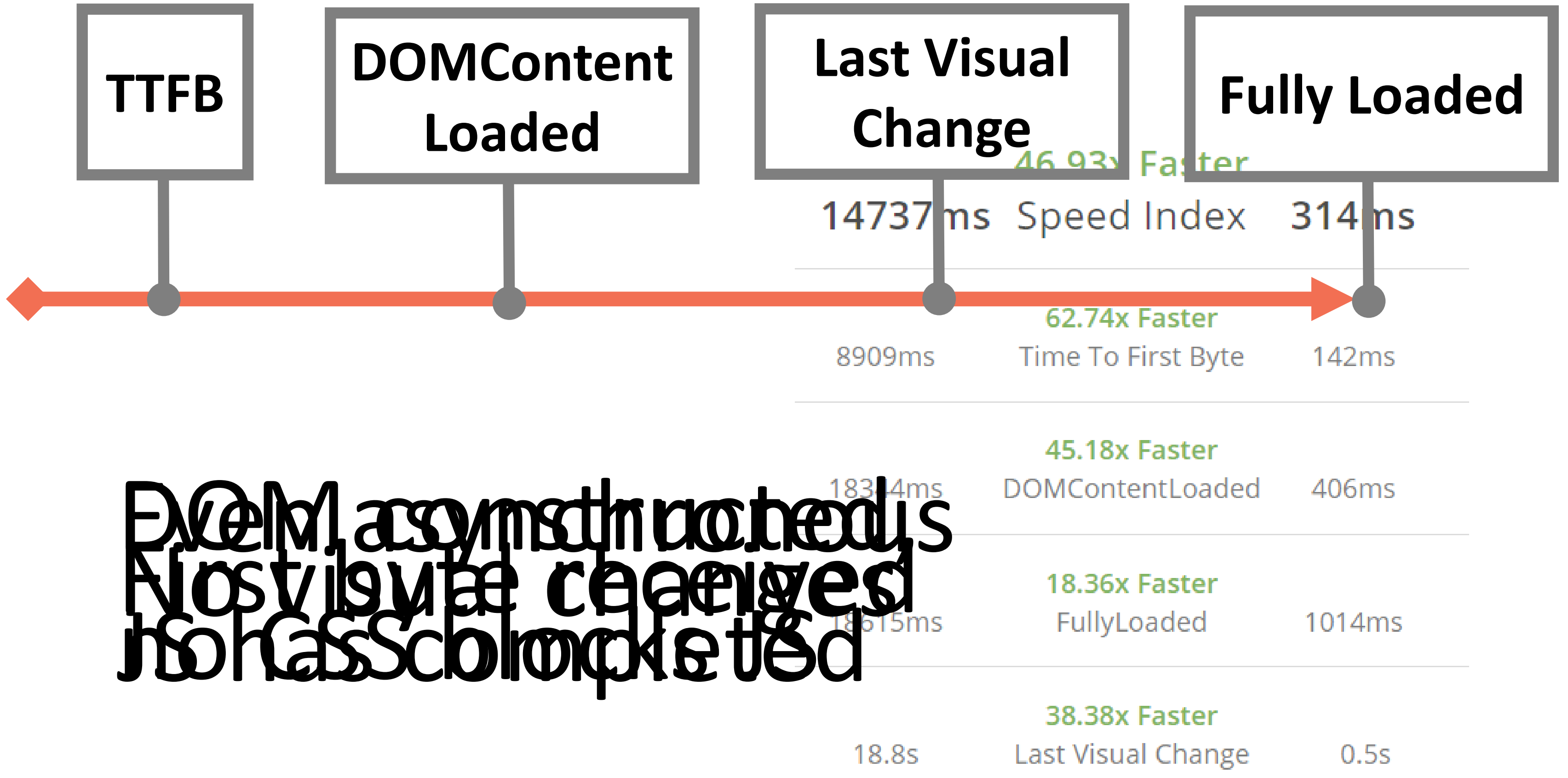


Your Website with Speed Kit



	46.93x Faster	
14737ms	Speed Index	314ms
	62.74x Faster	
8909ms	Time To First Byte	142ms
	45.18x Faster	
18344ms	DOMContentLoaded	406ms
	18.36x Faster	
18615ms	FullyLoaded	1014ms
	38.38x Faster	
18.8s	Last Visual Change	0.5s

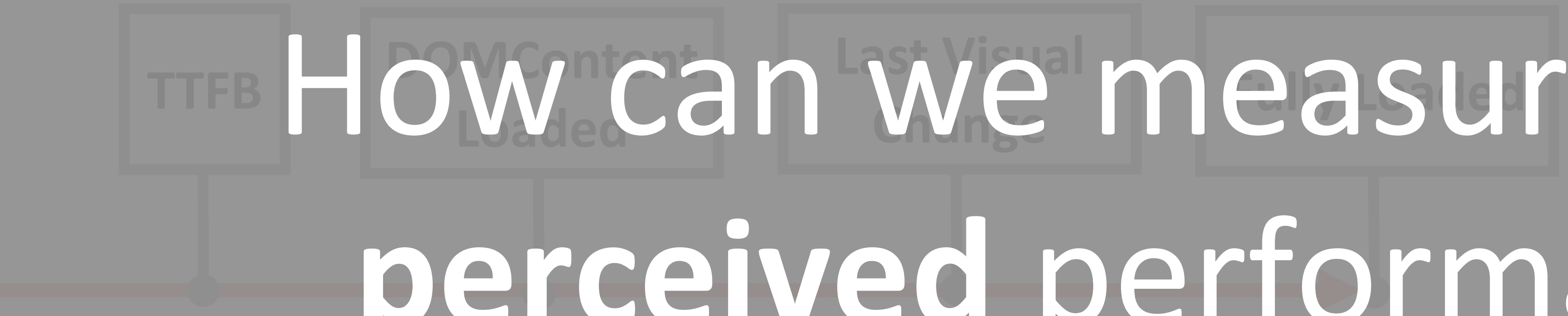
Measuring Web Performance



	46.93x Faster	
14737ms	Speed Index	314ms
	62.74x Faster	
8909ms	Time To First Byte	142ms
	45.18x Faster	
18344ms	DOMContentLoaded	406ms
	18.36x Faster	
18615ms	FullyLoaded	1014ms
	38.38x Faster	
18.8s	Last Visual Change	0.5s

DOM constructed
Even async loads
First byte received
No visual changes
JS & CSS blocked
onload completed

Measuring Web Performance

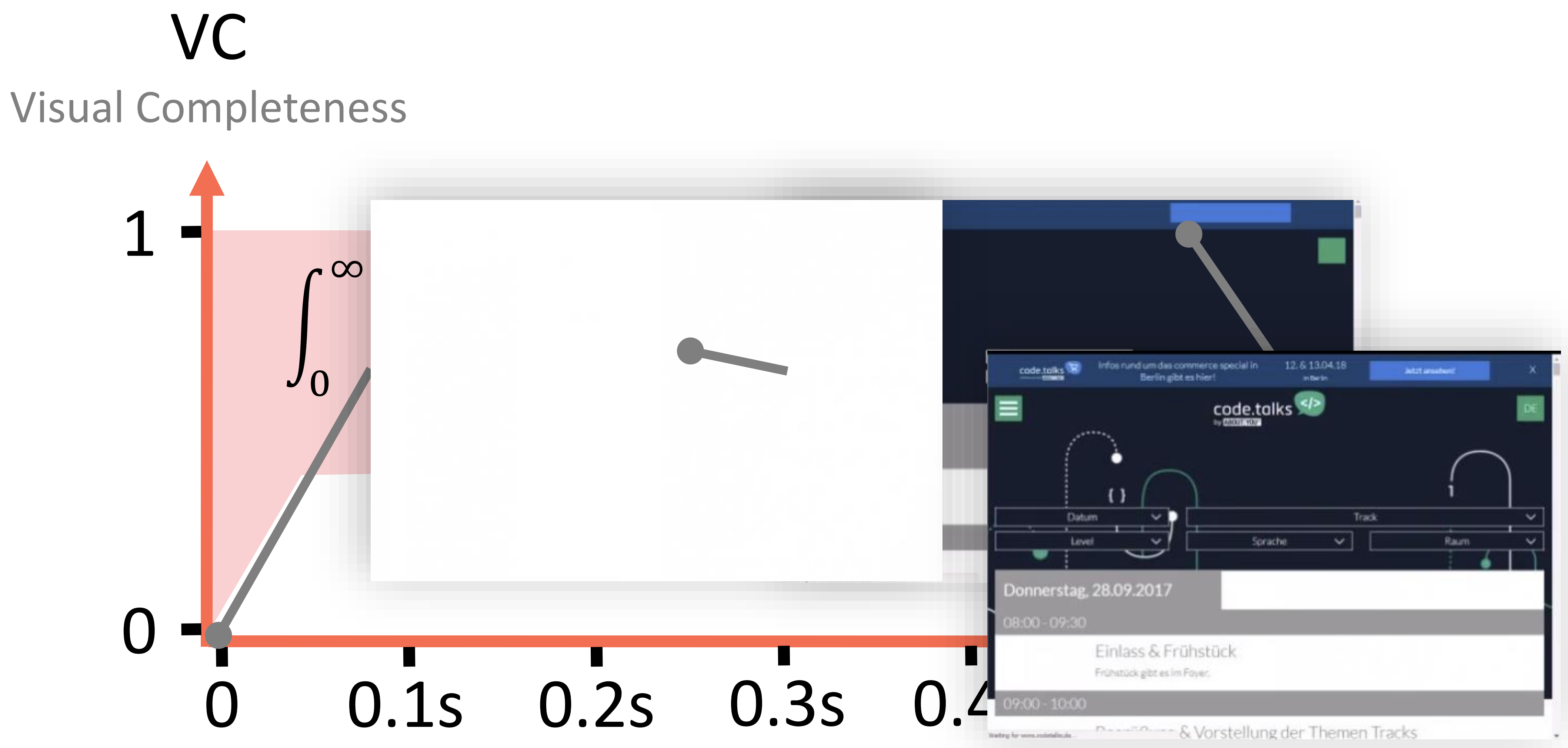


	46.93x Faster	
14737ms	Speed Index	314ms
	62.74x Faster	
8909ms	Time To First Byte	142ms
	45.18x Faster	
18344ms	DOMContentLoaded	406ms
	18.36x Faster	
18615ms	FullyLoaded	1014ms
	38.38x Faster	
18.8s	Last Visual Change	0.5s

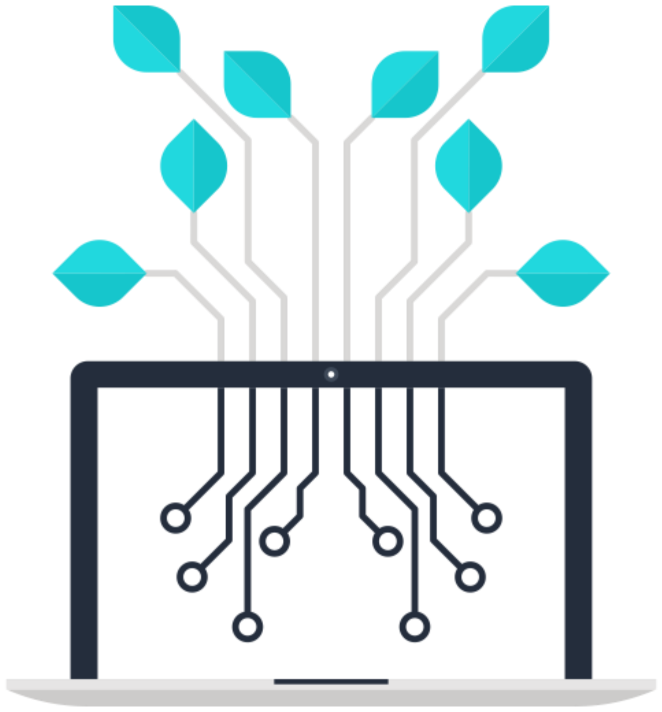
Even asynchronous JS has completed

How can we measure **user-perceived** performance?

The Speed Index



What we Learned: Wrap-up



Frontend

- AMP and Instant Articles: Fast but very limited
- PWAs bring native qualities to the web: offline, fast loads, push notifications

What we Learned: Wrap-up

Network

- HTTP/2 is much faster due to multiplexing and push
- CDNs tackle latency & caching
- Service Workers can modify the browser‘s requests

What we Learned: Wrap-up

Backend

- Cloud Providers make scaling out easier
- Servers and Database Systems need to support scalability and failover

How can we improve the performance of existing sites?

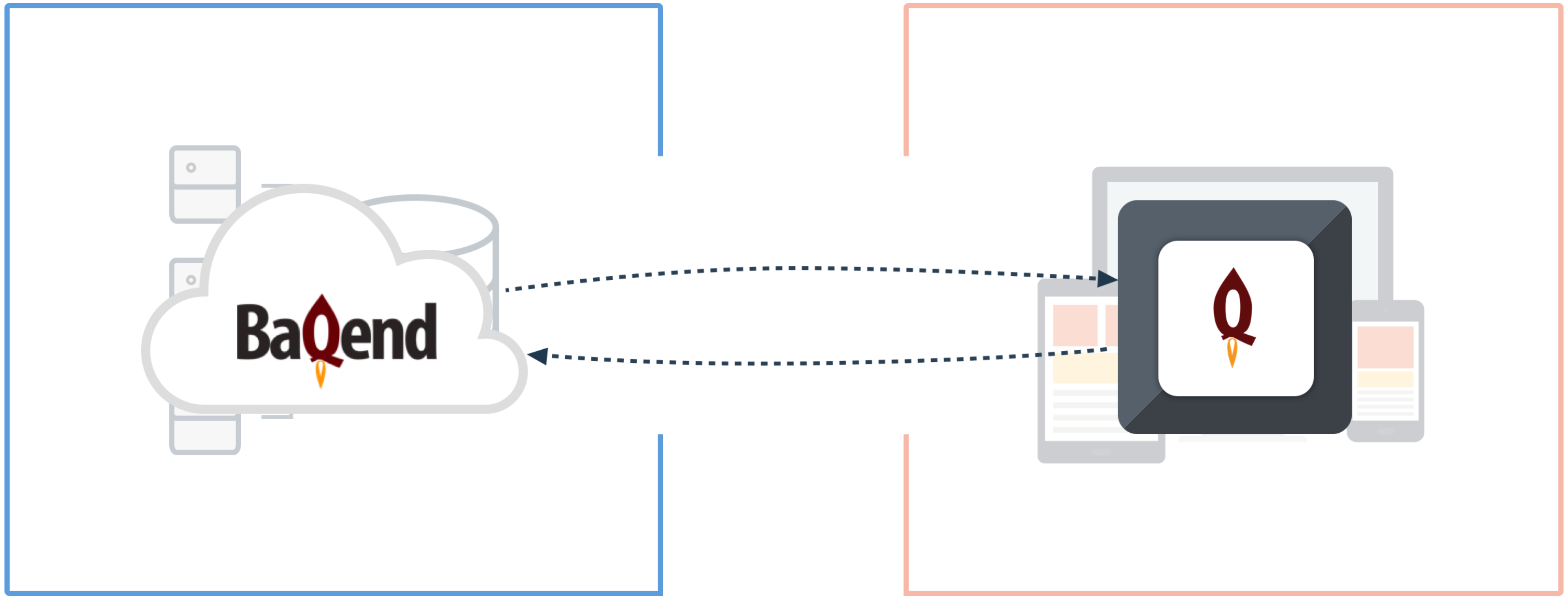
What we Learned: Wrap-up

Backend

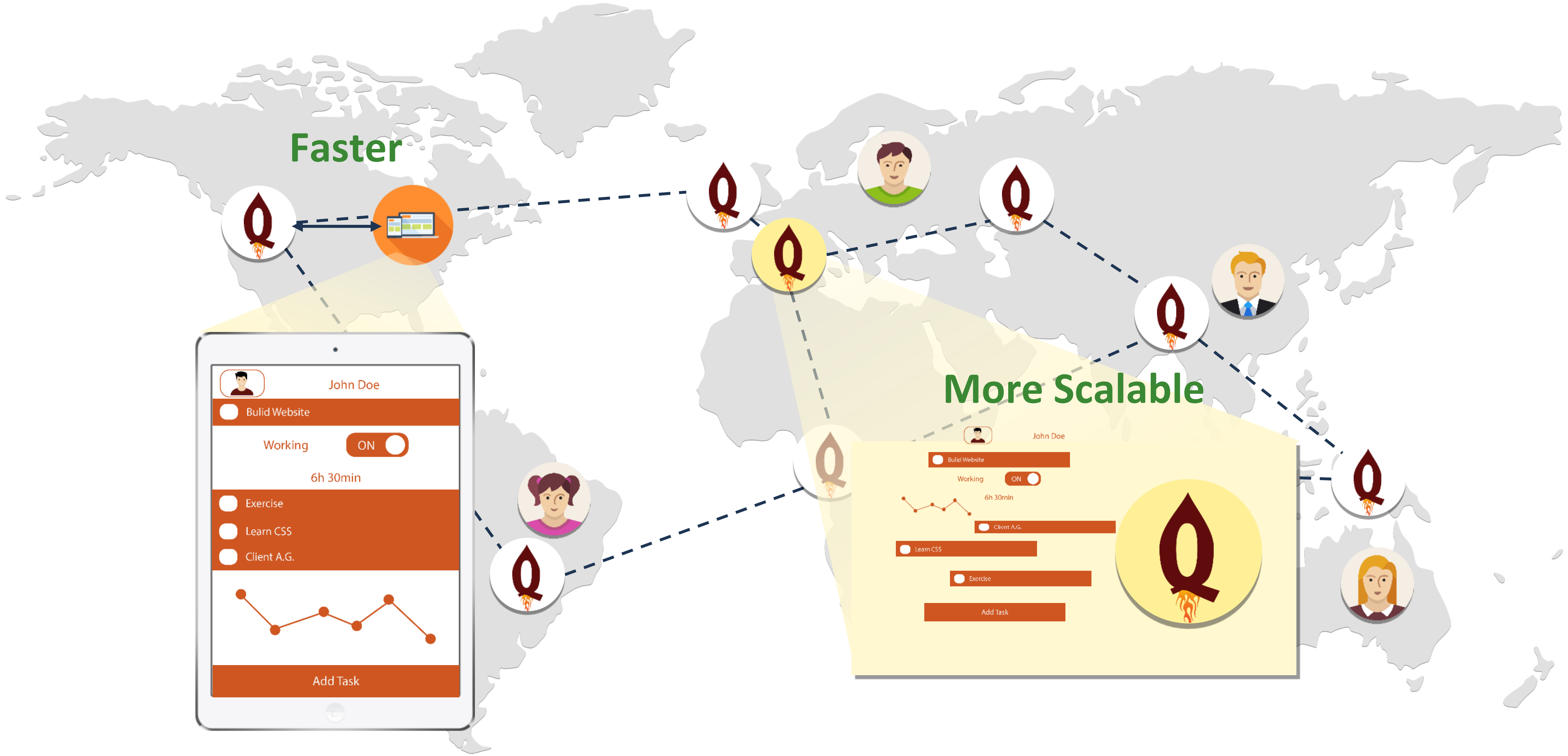
- Cloud Providers make scaling out easier
- [illegible] need to support scalability and failover

Speed Kit

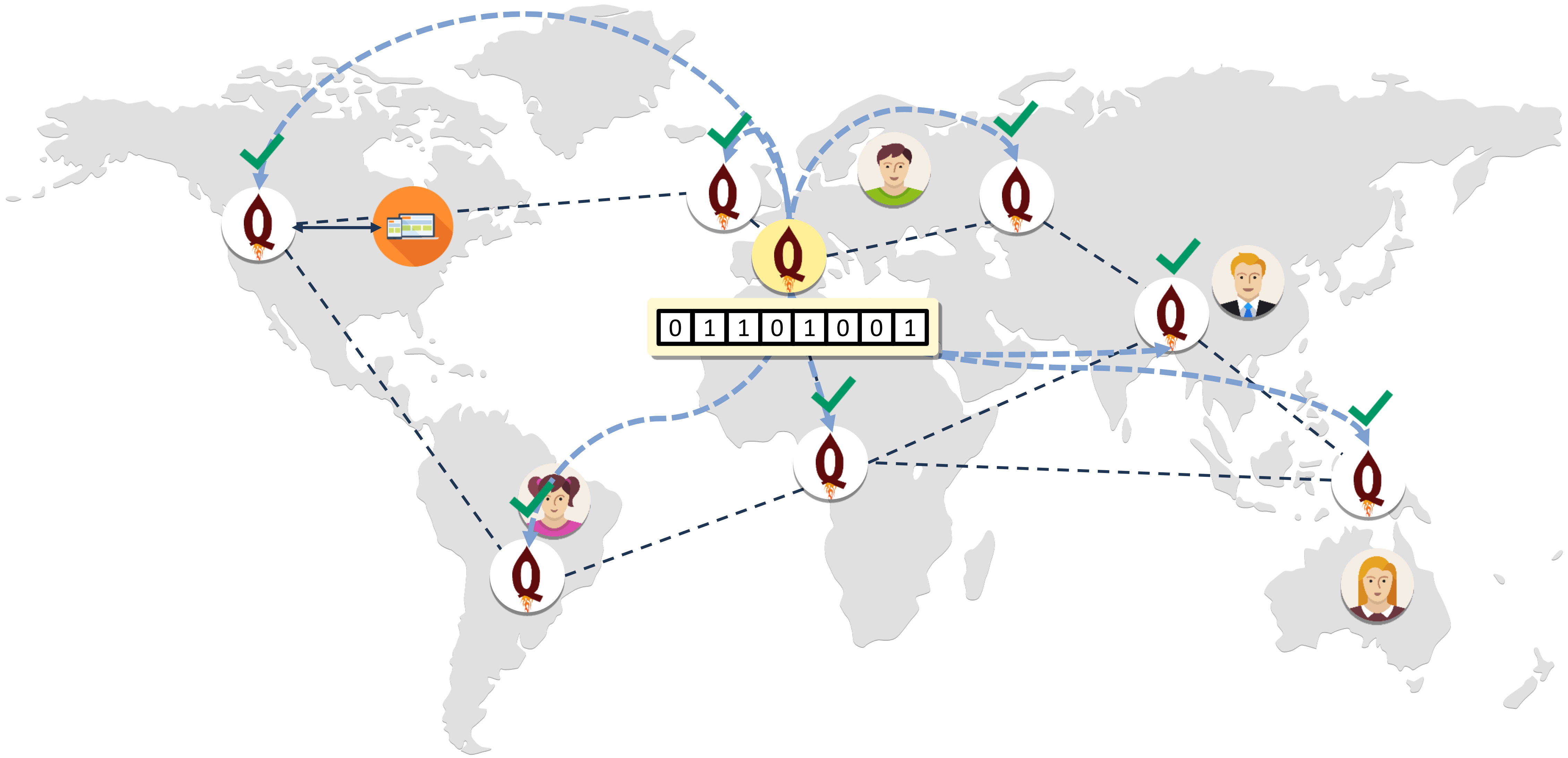
Turning Websites into Instantly-Loading Progressive Web Apps



What Speed Kit does.



What Speed Kit does.



What Speed Kit does.

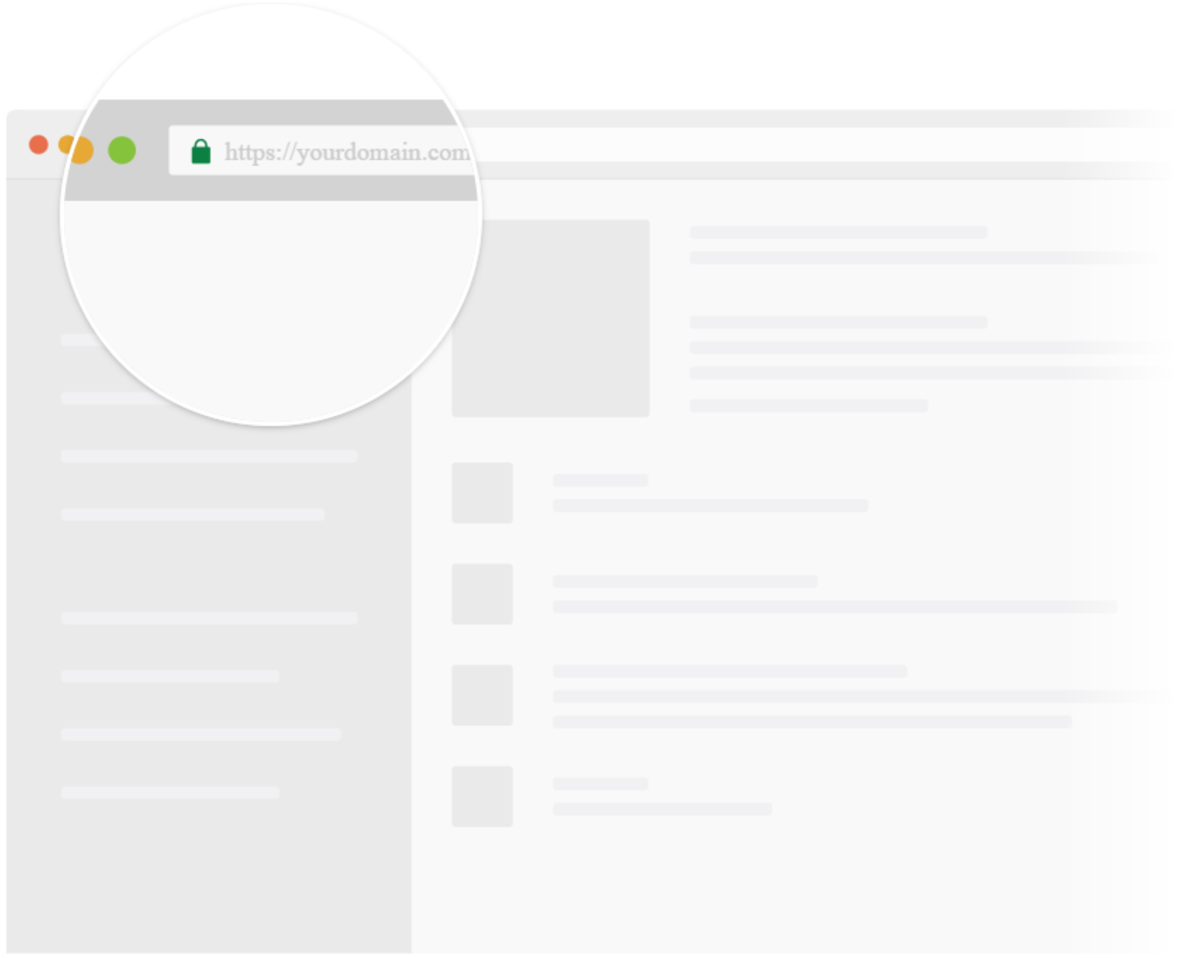
Backed by 30 Man-Years of Research

- F. Gessert, F. Bücklers, und N. Ritter, „ORESTES: a Scalable Database-as-a-Service Architecture for Low Latency“, in *CloudDB 2014*, 2014.
- F. Gessert und F. Bücklers, „ORESTES: ein System für horiz[illegible]kalierbaren Zugr[illegible]uf Cloud-Datenbanken“, in Informatiktage 2013, 2013.
- F. Gessert und F. Bücklers, *Performanz- und Reaktivitätssteigerung von OODBMS vermitt[illegible] Caching[illegible]erar[illegible]e*. Bachelorarbeit, 2010.
- M. Schaarschmidt, F. Gessert, und N. Ritter, „Towards Automated Polyglot Persistence“, in BTW 2015.
- S. Friedrich, W. Wingerath, F. Gessert, und N. Ritter, „NoSQL OLTP Benchmarking: A Survey“, in *44. Jahrestagung der Gesellschaft für Informatik*, 2014, Bd. 232, S. 693–704.
- W. Wingerath, F. Gessert, S. Friedrich, N. Ritter „Real-time stream processing for Big Data“, *Big Data Analytics it - Information Technology,* 2016
- F. Gessert, W. Wingerath, S. Friedrich, N. Ritter “NoSQL Database Systems: A Survey and Decision Guidance”, *Computer Science - Research and Development,* 2016
- F. Gessert, S. Friedrich, W. Wingerath, M. Schaarschmidt, und N. Ritter, „Towards a Scalable and Unified REST API for Cloud Data Stores“, in *44. Jahrestagung der GI*, Bd. 232, S. 723–734.
- F. Ges[illegible] M. Sch[illegible]arschmidt, W. Wingerath, S. Friedrich, und N. Ritter, „The Cache Sket[illegible]: [illegible]siting Expiration-based Caching in the Age of Cloud Data Management“, [illegible] BTW 2015.
- F. Gessert und F. Bücklers, *Kohärentes Web-Caching von Datenbankobjekten im Cloud [illegible]omputing*. Masterarbeit 2012.
- W. Wingerath, S. Friedrich, und F. Gessert, „Who Watches the Watchmen? On the Lack of Validation in NoSQL Benchmarking“, in BTW 2015.
- F. Gessert, „Skalierbare NoSQL- und Cloud-Datenbanken in Forschung und Praxis“, BTW 2015
- F. Gessert, N. Ritter „Scalable Data Management: NoSQL Data Stores in Research and Practice“, *32nd IEEE International Conference on Data Engineering, ICDE,* 2016
- F. Gessert, N. Ritter „Polyglot Persistence“, *Datenbank Spektrum*, 2016.

Adding Speed Kit to a Site

1. Configure Domain

Set which URLs Baqend should accelerate.

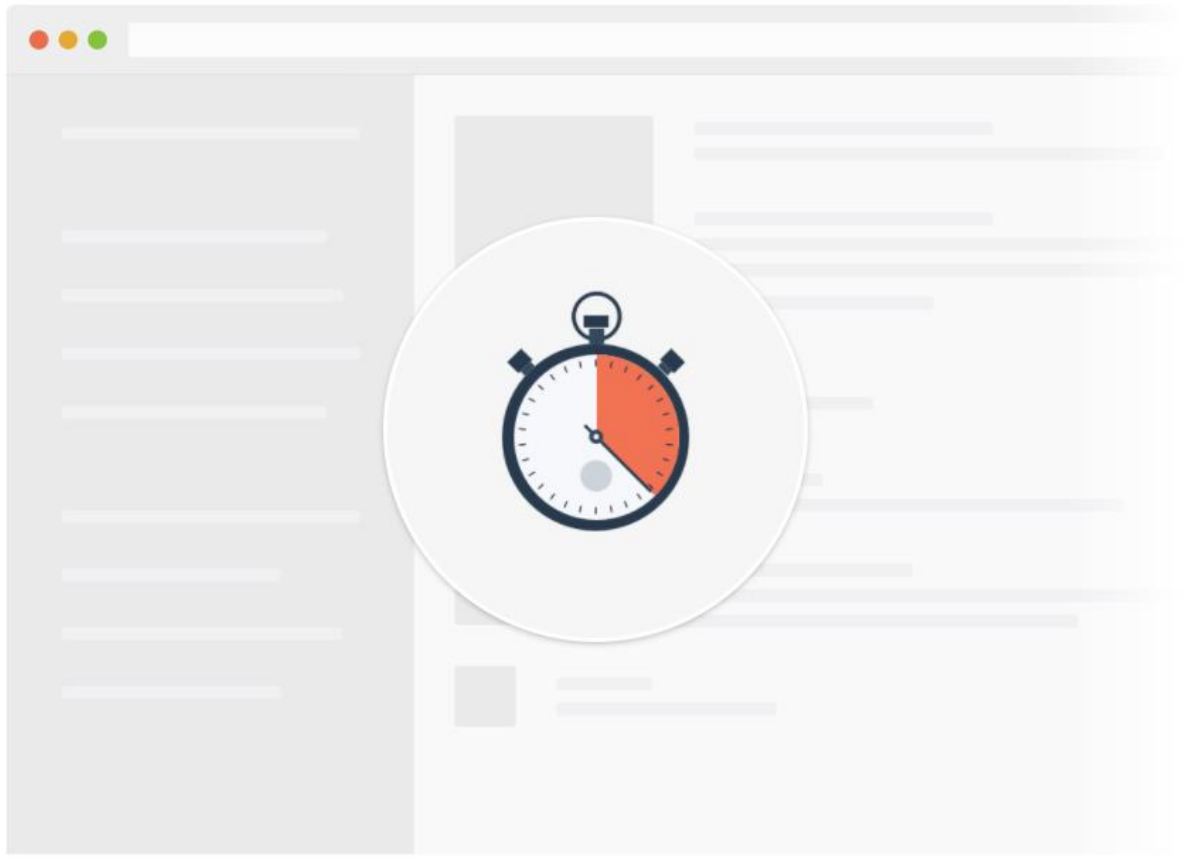


2. Include Code Snippet

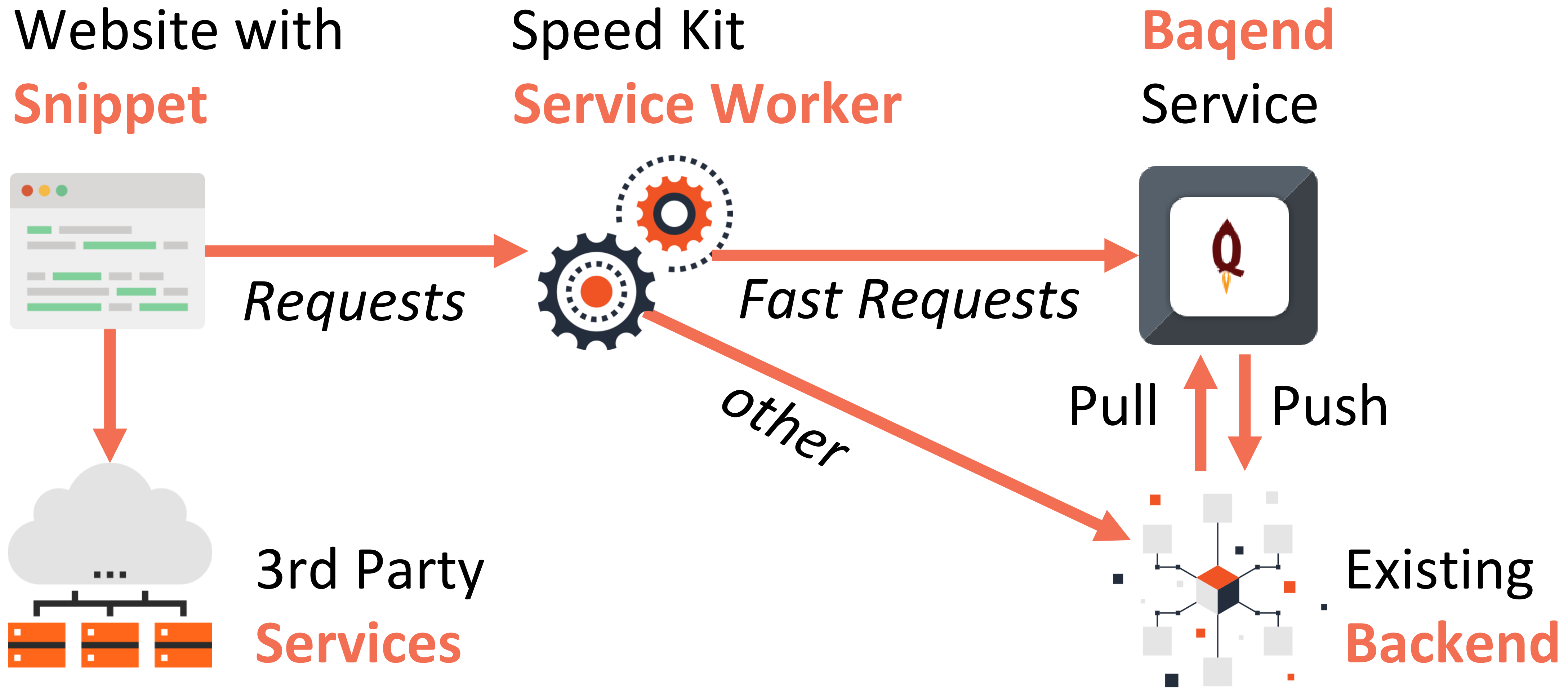
Add the Speed Kit Service Worker to the website.

3. Requests Accelerated

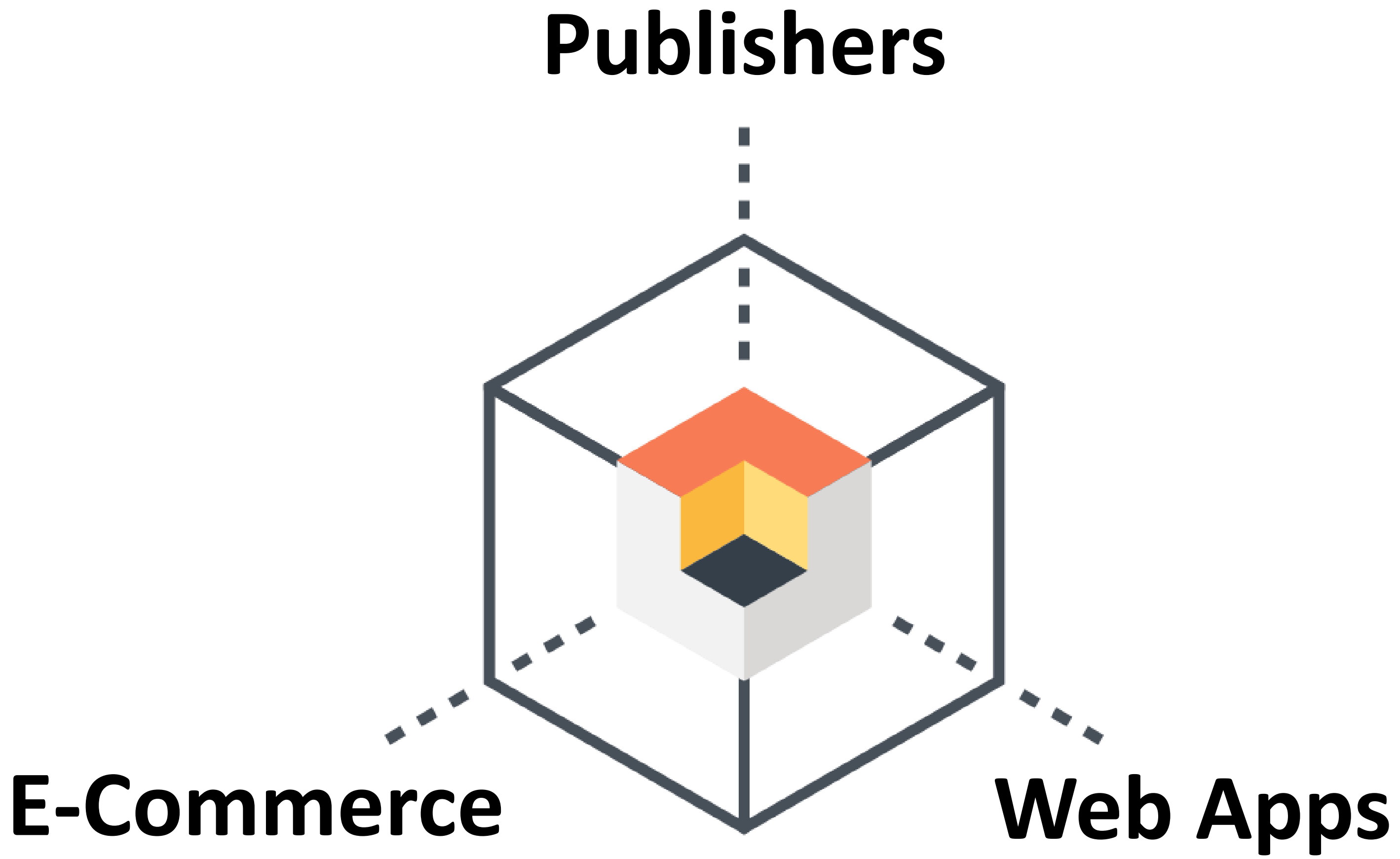
Speed Kit routes the requests through Baqend‘s CDN.



How it works under the hood

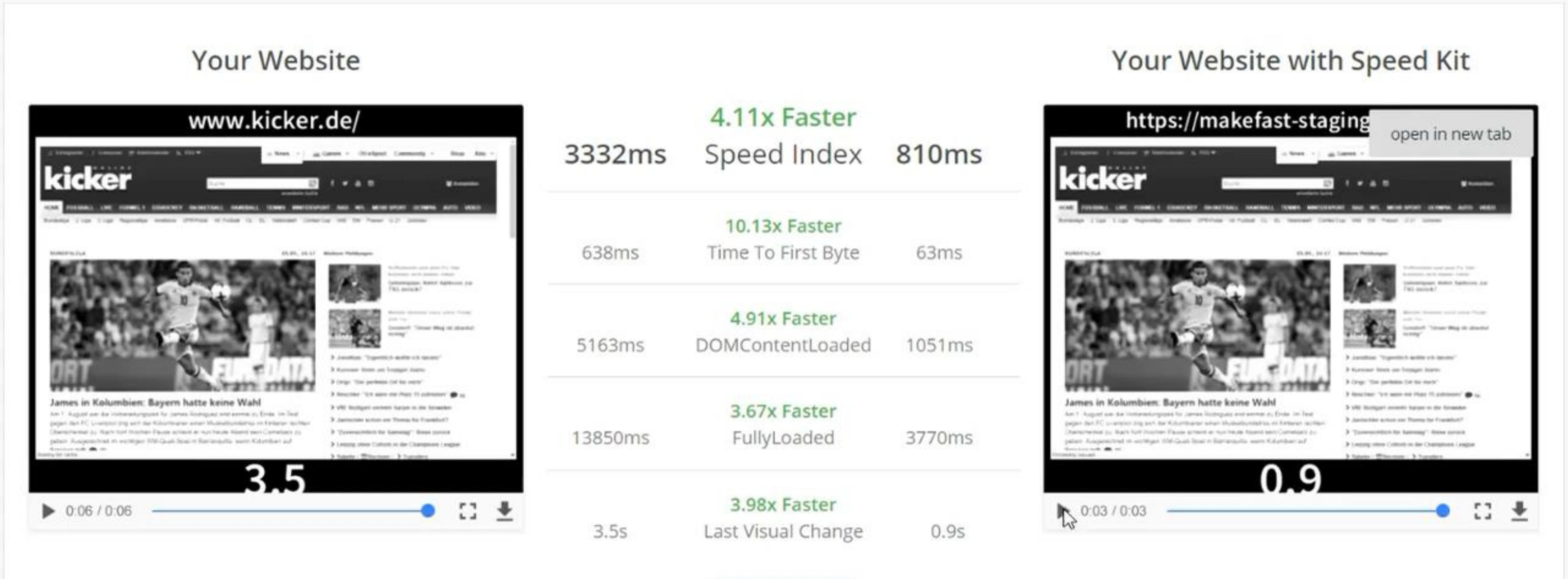


Speed Kit works across tech stacks.



Works for Publishers.

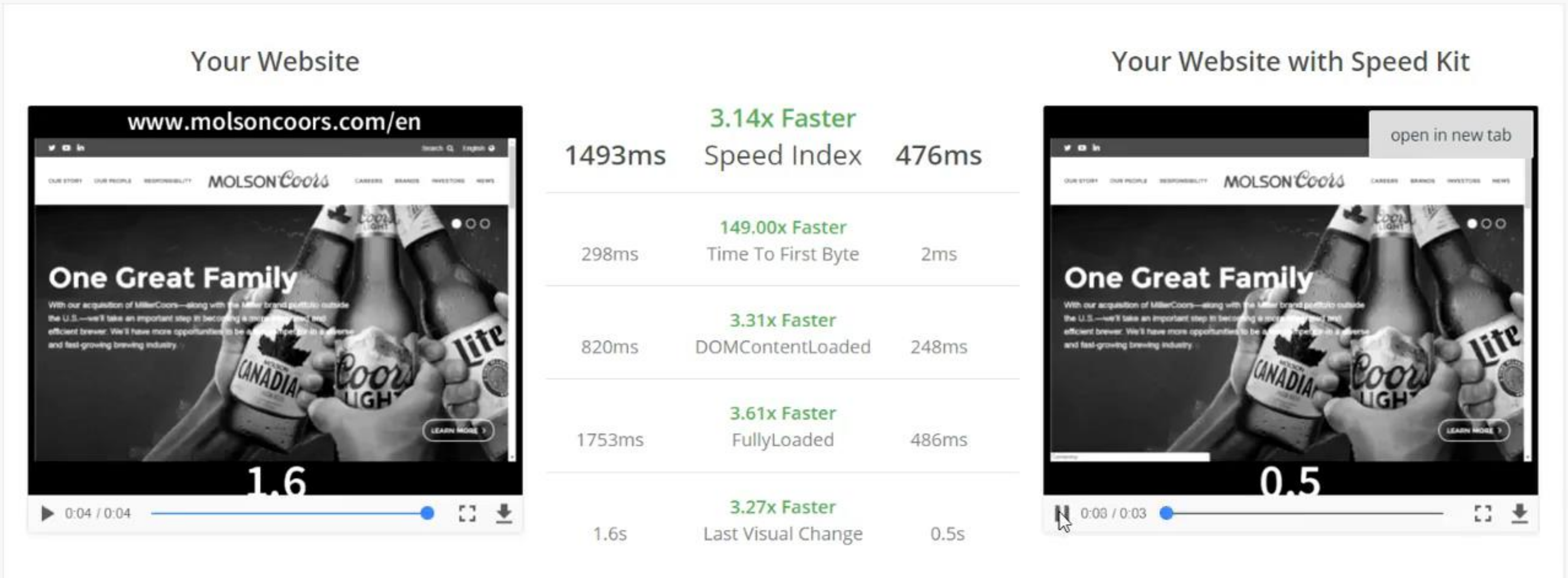
kicker.de



Your Website	Comparison	Your Website with Speed Kit
3332ms	4.11x Faster Speed Index	810ms
638ms	10.13x Faster Time To First Byte	63ms
5163ms	4.91x Faster DOMContentLoaded	1051ms
13850ms	3.67x Faster FullyLoaded	3770ms
3.5s	3.98x Faster Last Visual Change	0.9s

Works for Landing Pages.

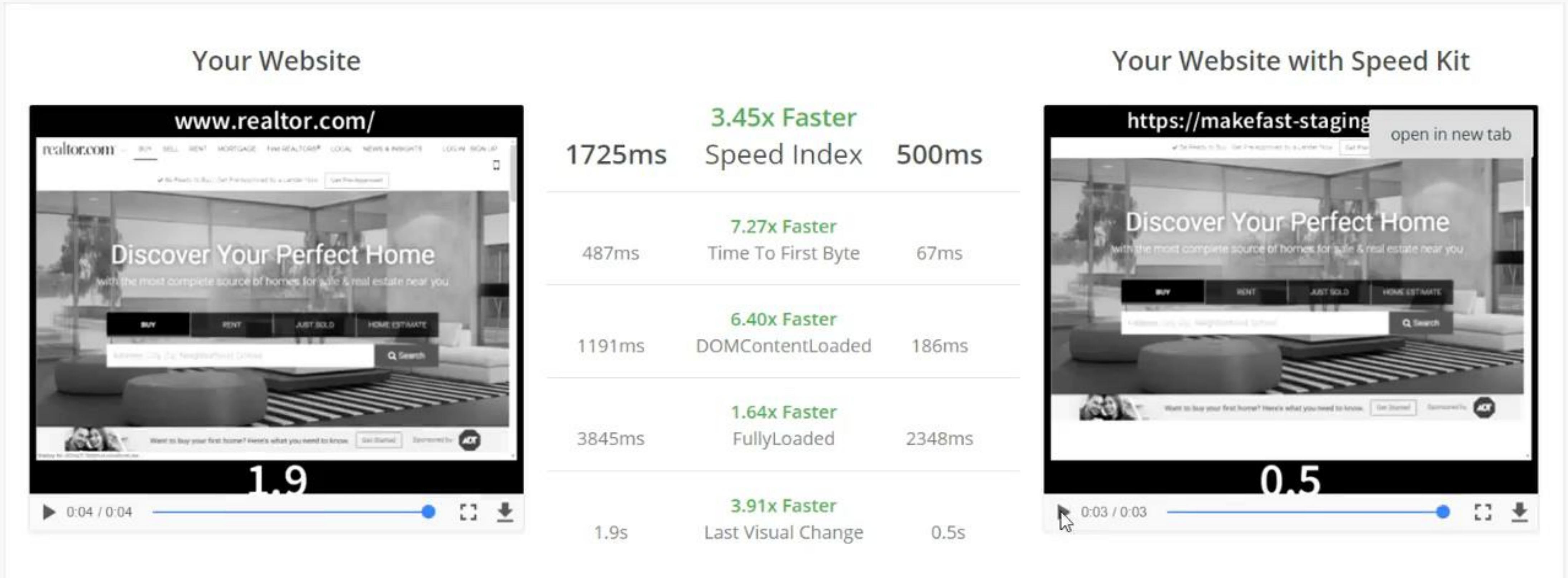
molsoncoors.com



Your Website		Your Website with Speed Kit
1493ms	3.14x Faster Speed Index	476ms
298ms	149.00x Faster Time To First Byte	2ms
820ms	3.31x Faster DOMContentLoaded	248ms
1753ms	3.61x Faster FullyLoaded	486ms
1.6s	3.27x Faster Last Visual Change	0.5s

Works for Portals.

realtor.com



Your Website

www.realtor.com/

Discover Your Perfect Home

with the most complete source of homes for sale & real estate near you

1.9

0:04 / 0:04

Your Website with Speed Kit

https://makefast-staging

open in new tab

Discover Your Perfect Home

with the most complete source of homes for sale & real estate near you

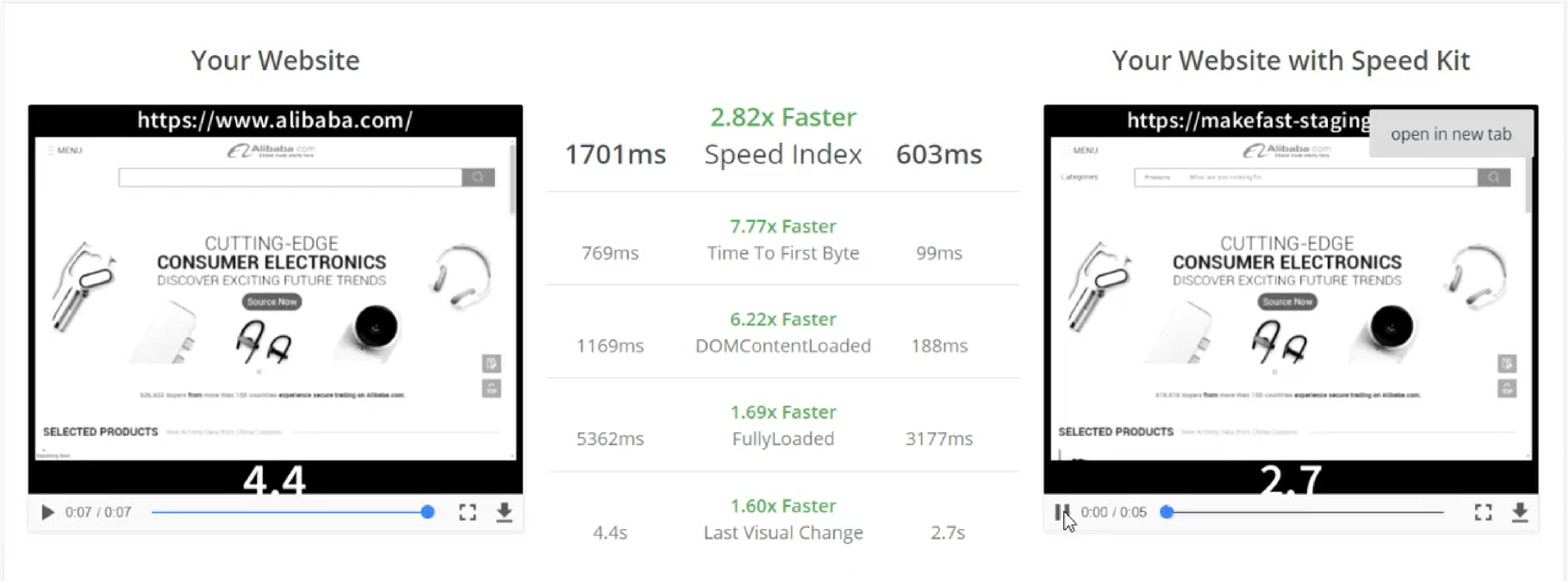
0.5

0:03 / 0:03

1725ms	3.45x Faster Speed Index	500ms
487ms	7.27x Faster Time To First Byte	67ms
1191ms	6.40x Faster DOMContentLoaded	186ms
3845ms	1.64x Faster FullyLoaded	2348ms
1.9s	3.91x Faster Last Visual Change	0.5s

Works for E-Commerce.

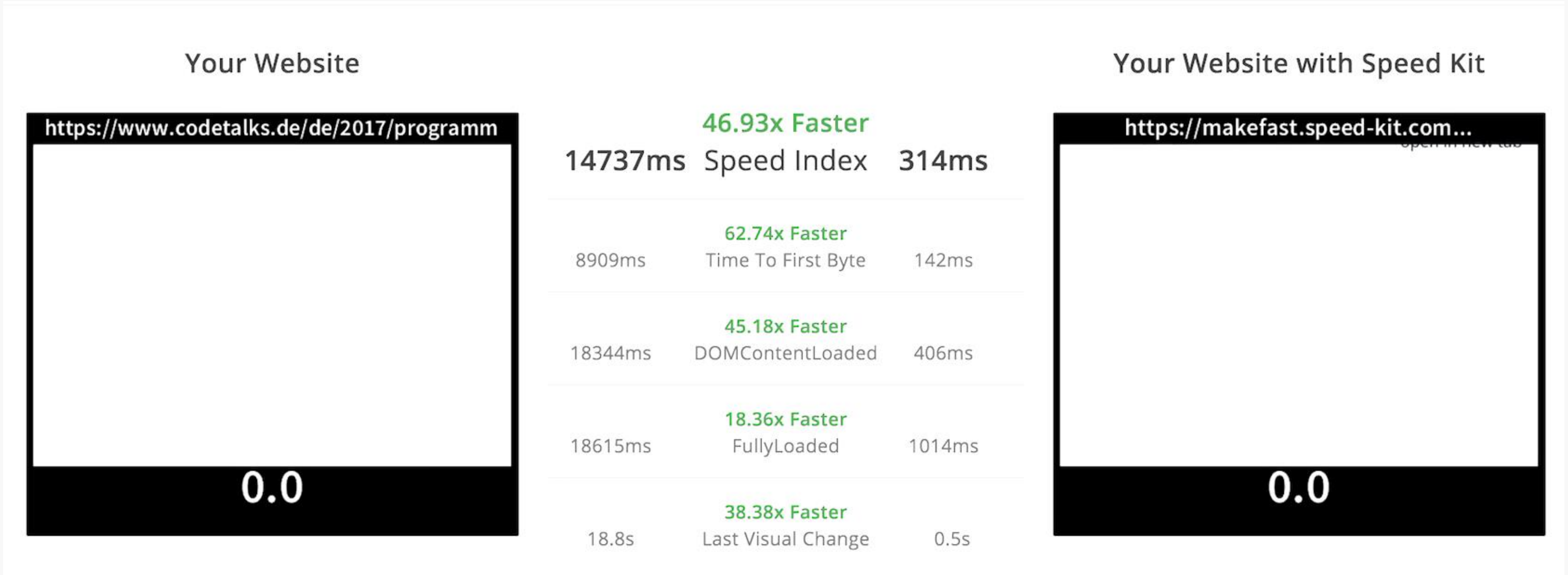
alibaba.com



Your Website		Your Website with Speed Kit
1701ms	2.82x Faster Speed Index	603ms
769ms	7.77x Faster Time To First Byte	99ms
1169ms	6.22x Faster DOMContentLoaded	188ms
5362ms	1.69x Faster FullyLoaded	3177ms
4.4s	1.60x Faster Last Visual Change	2.7s

Works for Conference Websites.

codetalks.de

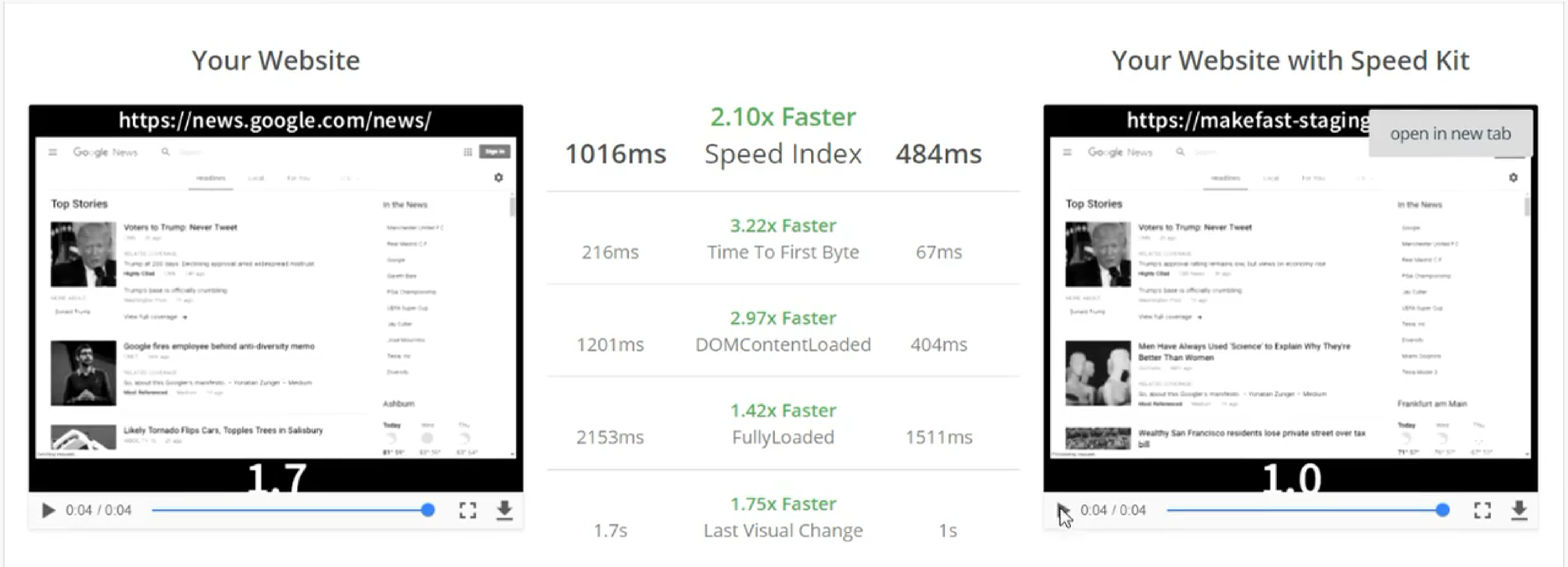


Your Website		Your Website with Speed Kit
https://www.codetalks.de/de/2017/programm		https://makefast.speed-kit.com...
0.0		0.0

	46.93x Faster	
14737ms	Speed Index	314ms
	62.74x Faster	
8909ms	Time To First Byte	142ms
	45.18x Faster	
18344ms	DOMContentLoaded	406ms
	18.36x Faster	
18615ms	FullyLoaded	1014ms
	38.38x Faster	
18.8s	Last Visual Change	0.5s

Works for Aggregators.

news.google.com



Your Website		Your Website with Speed Kit
1016ms	2.10x Faster Speed Index	484ms
216ms	3.22x Faster Time To First Byte	67ms
1201ms	2.97x Faster DOMContentLoaded	404ms
2153ms	1.42x Faster FullyLoaded	1511ms
1.7s	1.75x Faster Last Visual Change	1s

Does it work for Your Site?

test.speed-kit.com

What we develop at Baqend

Speed Kit

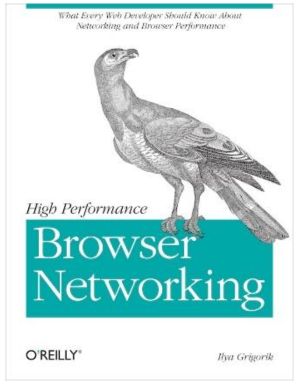
- Turns Existing Sites into **PWAs**
- **50-300% Faster** Loads
- **Offline** Mode

Platform

- Platform for building (Progressive) **Web Apps**
- **15x** Performance Edge
- Faster **Development**

Web Performance Literature

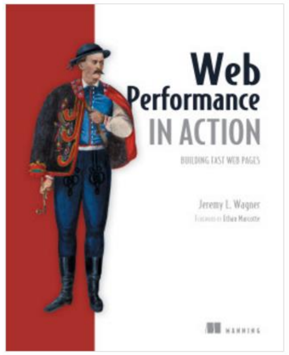
Good Resources


What Every Web Developer Should Know About Networking and Browser Performance

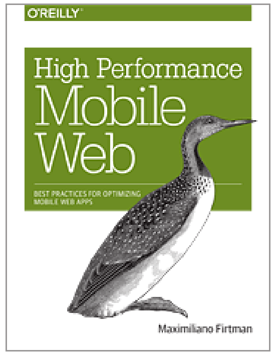
High Performance Browser Networking

O'REILLY

Ilya Grigorik


Web Performance IN ACTION

Jeremy L. Wagner


O'REILLY

High Performance Mobile Web

Maximiliano Firtman

O'REILLY

Building Progressive Web Apps

Tal Ater

https://hpbn.co/

Google Developers Performance

https://developers.google.com/web/fundamentals/performance/?hl=en

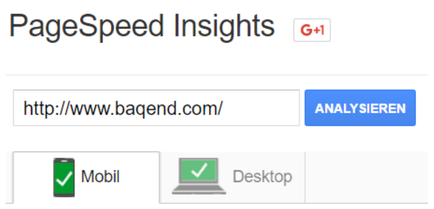
Website Performance Optimization
The Critical Rendering Path

https://www.udacity.com/course/website-performance-optimization--ud884

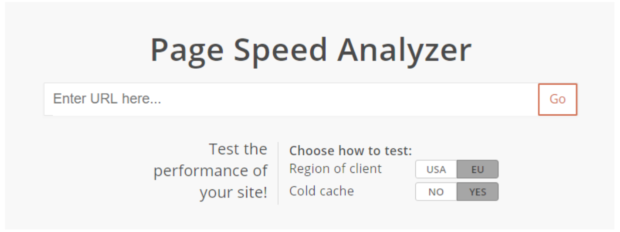
Baqend Blog
On Building a Faster Web

https://medium.baqend.com/

Performance Tools


PageSpeed Insights

http://www.baqend.com/


Page Speed Analyzer

Enter URL here...

Test the performance of your site!

Choose how to test:
Region of client: USA / EU
Cold cache: NO / YES

https://developers.google.com/speed/pagespeed/

https://test.speed-kit.com

Baqend

https://www.baqend.com/

WEBPAGETEST

http://www.webpagetest.org/

We are hiring.

Frontend Developers
Mobile Developers
Java Developers
Web Performance Engineers

Contact us.

BaQend Felix Gessert · fg@baqend.com · www.baqend.com

Questions?

Our other talks:

Th. 16:00 Real-Time Databases Explained: Why Meteor, RethinkDB, Parse and Firebase Don't Scale

Fr. 10:00 Real-Time Anwendungen mit React und React Native entwickeln

Fr. 17:00 Wie man ein Backend-as-a-Service entwickelt: Lessons Learned

BaQend Felix Gessert · fg@baqend.com · www.baqend.com