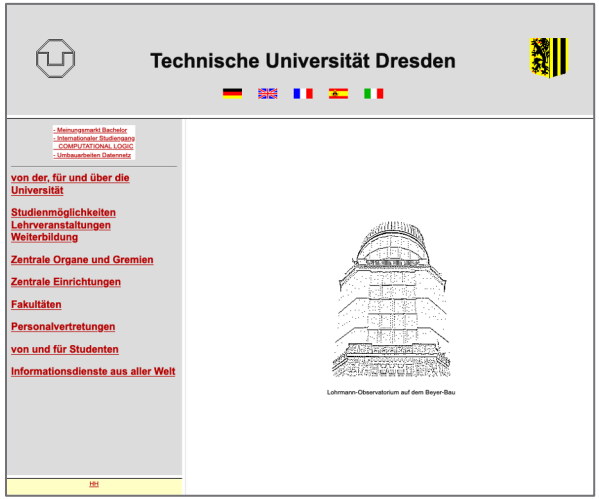
Lucas Vogel, Thomas Springer, Matthias Wählisch
Chair of Distributed and Networked Systems

From Files to Streams: A Critical Look at the Concept of the File-Based Web in the HTTP/3 Era

MIR^3, 23.11.2023

Web content changed drastically

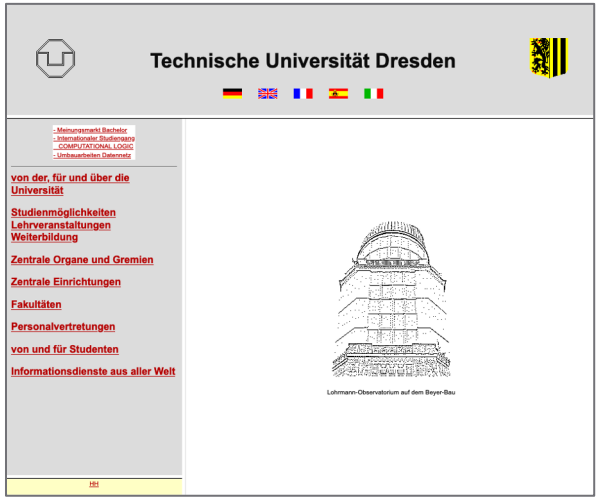
From mainly text

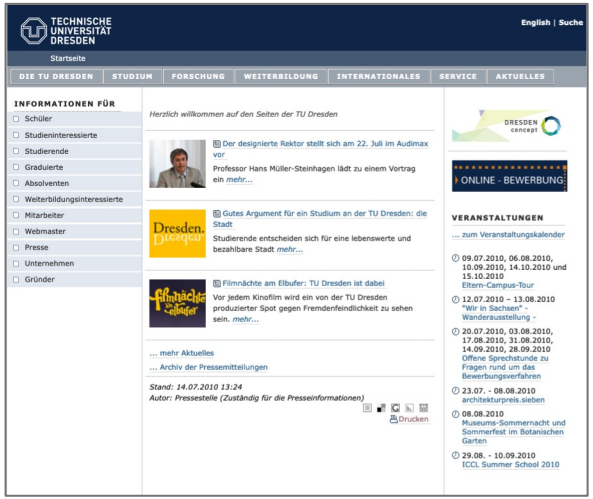


1997 2010 2023

Web content changed drastically

From mainly text to multimedia

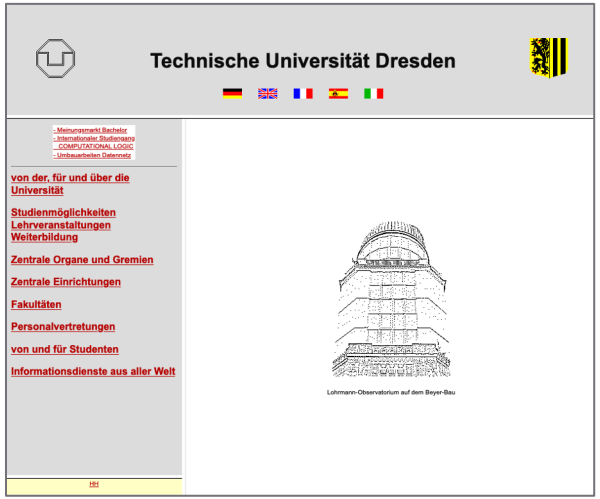


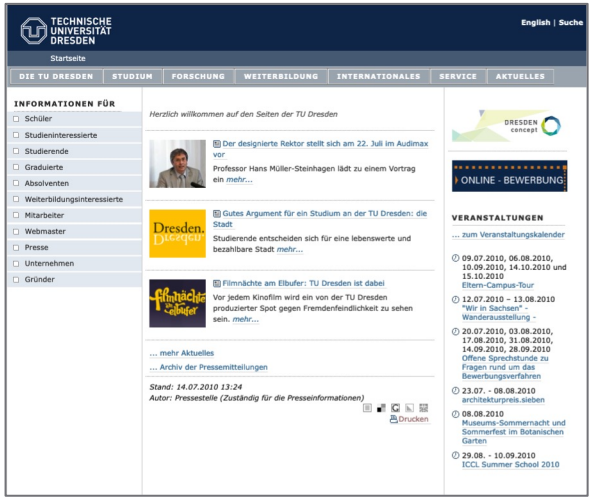


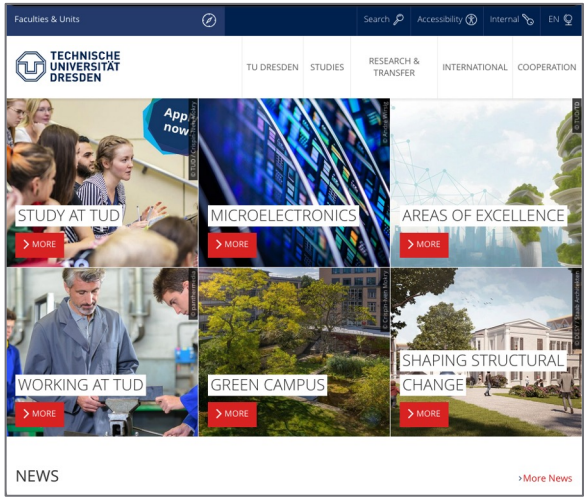
1997 2010 2023

Web content changed drastically

From mainly text to multimedia to dynamic content and many content pieces



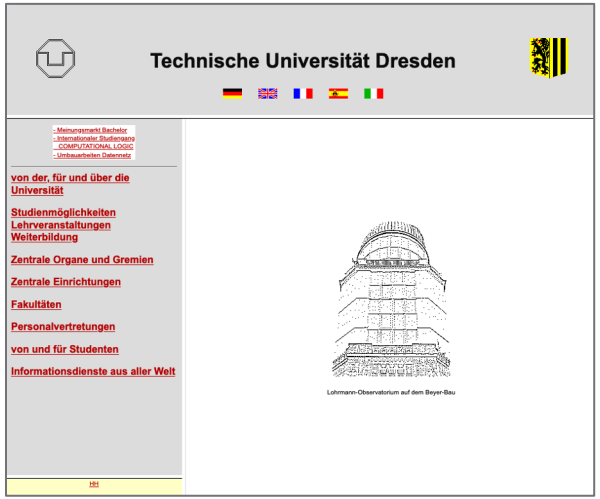




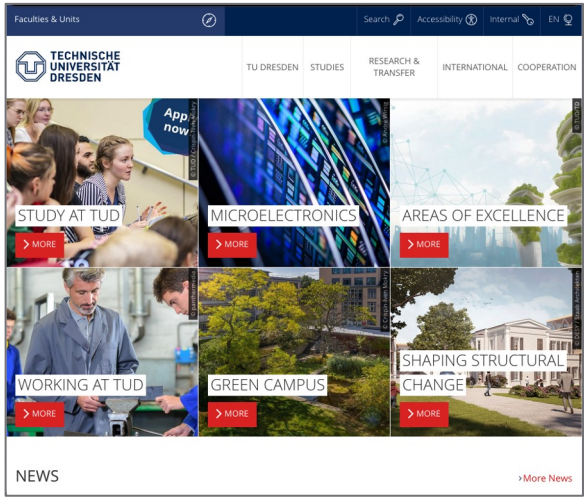
1997 2010 2023

Web content changed drastically

From mainly text to multimedia to dynamic content and many content pieces



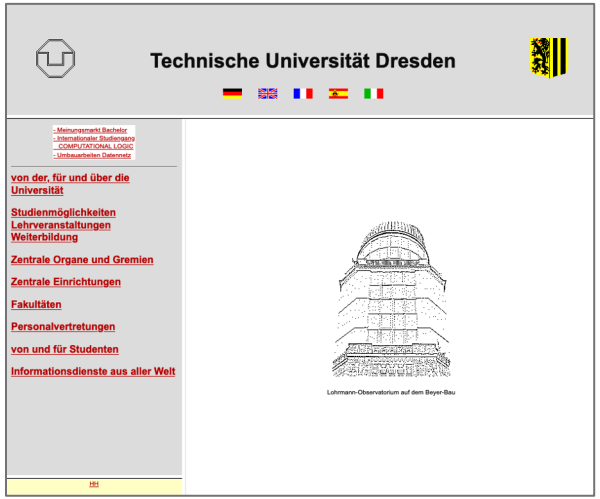
16236% more code



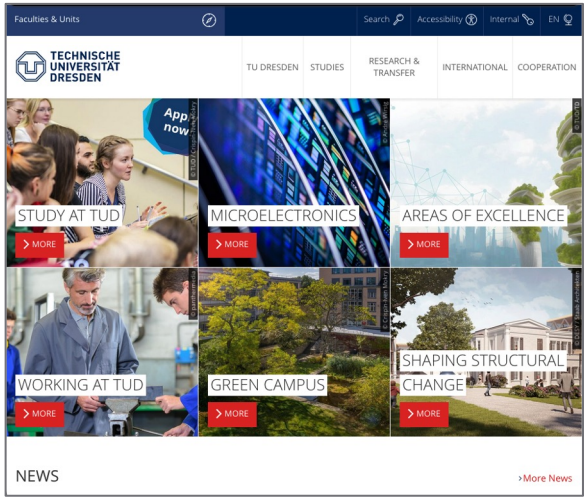
1997 — 2010 — 2023

Web content changed drastically

From mainly text to multimedia to dynamic content and many content pieces



16236% more code



1997 — 2010 — 2023

Why does speed matter?

57%

Ratio of users that leave a web page if the loading time is >3 seconds.

-20%

Loss of Google's ad revenue if loading the result page needs +0.5 seconds.

Starting point of this talk
Reducing latency is crucial in the Web

During the last 20 years, a lot changed to **deliver** Web content **fast**.

Physical network provides higher data rates

Protocols improved from HTTP/0.9 to HTTP/3

CDNs + better browser engines improved delivery and processing

Starting point of this talk
Reducing latency is crucial in the Web

During the last 20 years, a lot changed to **deliver** Web content **fast**.

Physical network provides higher data rates

QUIC

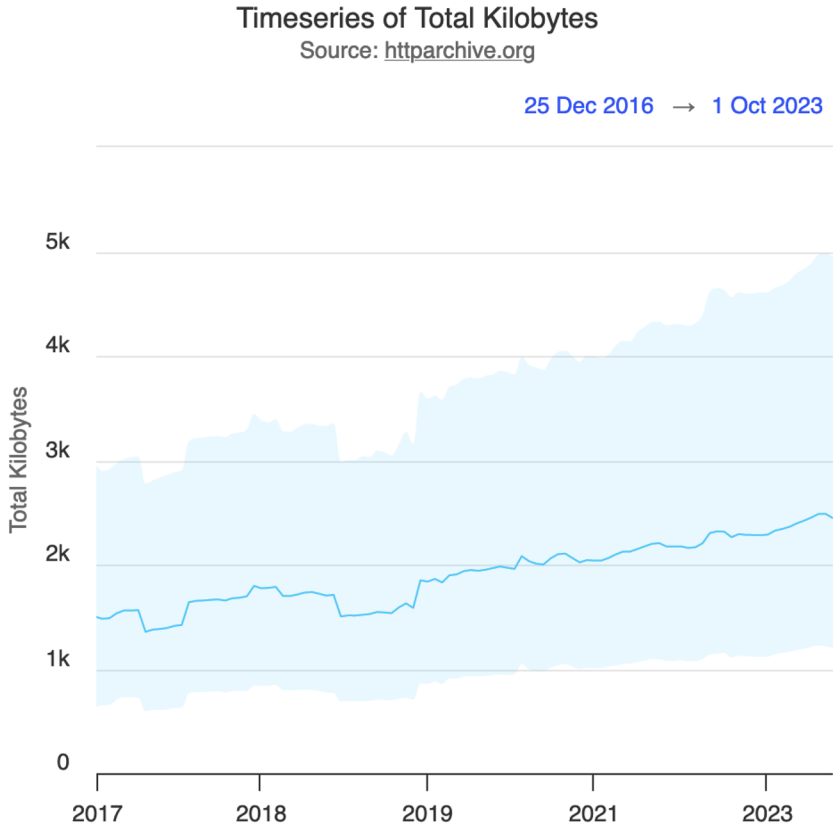
Protocols improved from HTTP/0.9 to HTTP/3

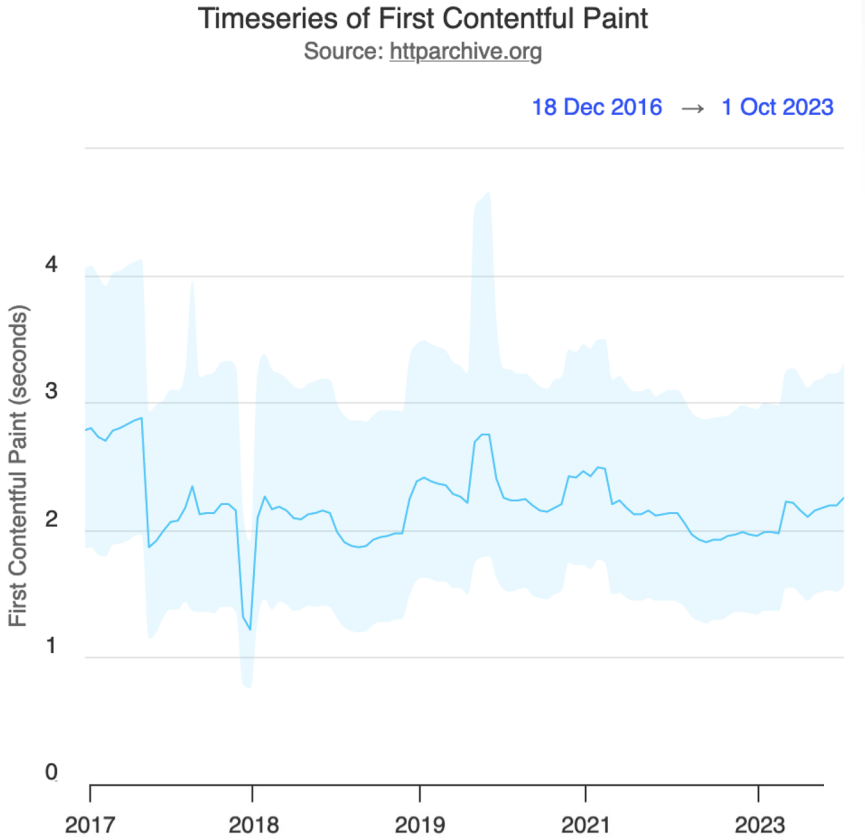
Akamai

CDNs + better browser engines improved delivery and processing

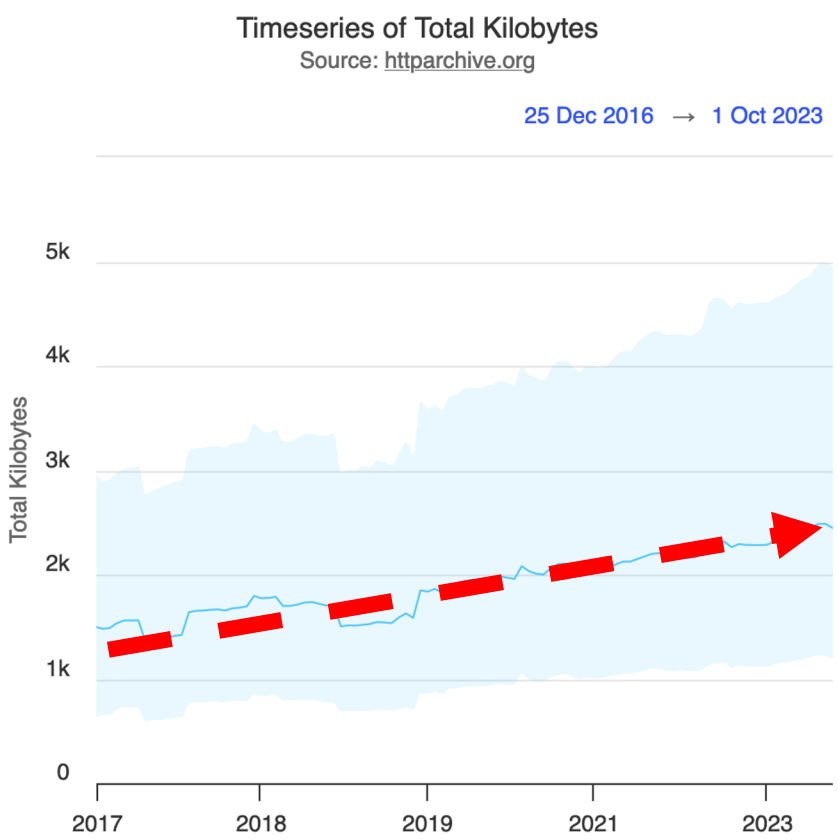
Unfortunately, we **didn't care** much about **presenting** content **timely.**

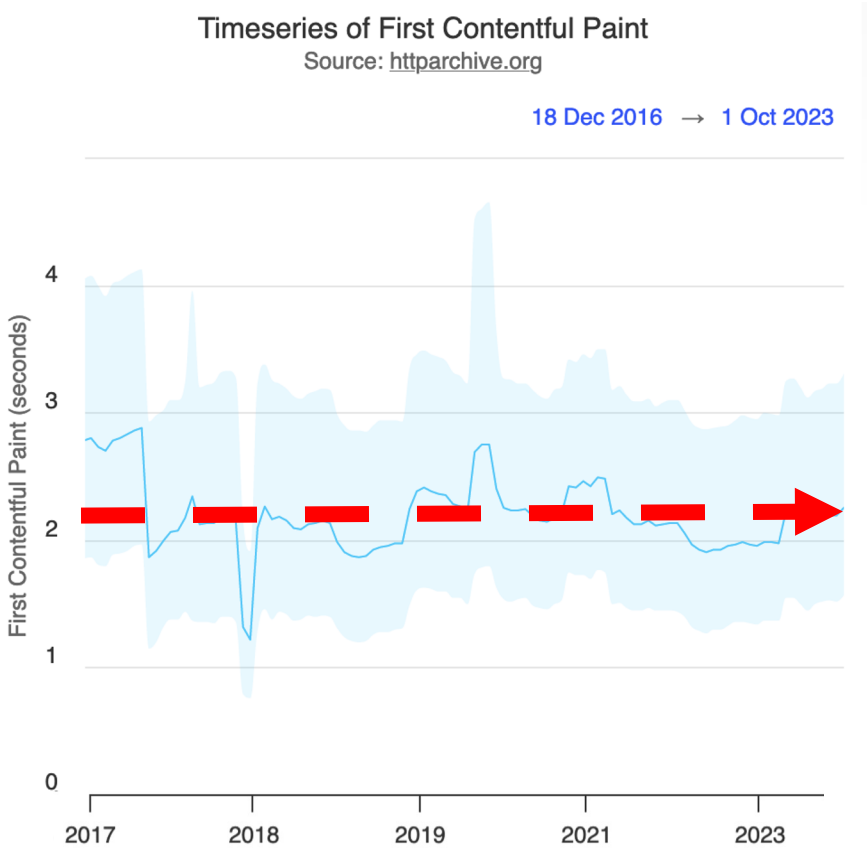
Web page size vs. first contentful paint on a desktop





Web page size vs. first contentful paint on a desktop





What is this talk about?

What slows down content presentation?
Beyond transmission delay, the problem of render blocking.

Why are we faced with the situation?
A brief historic recap.

How can we improve the situation?
Split bundles and stream content pieces.

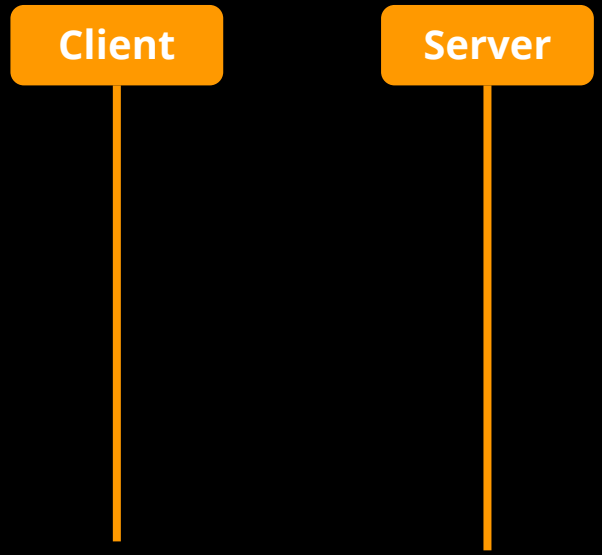
What is this talk about?

What slows down content presentation?
Beyond transmission delay, the problem of render blocking.

Why are we faced with the situation?
A brief historic recap.

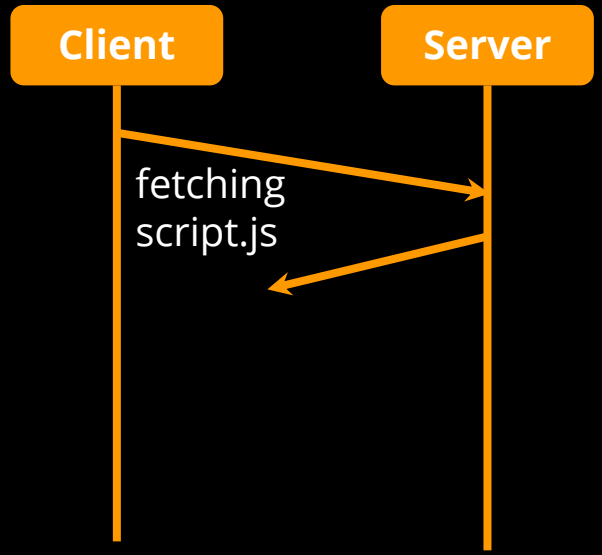
How can we improve the situation?
Split bundles and stream content pieces.

```
<head>
<script src="script.js"></script>
<link rel="stylesheet" type="text/css" href="style.css"/>
</head>
<body>Hello World!</body>
```



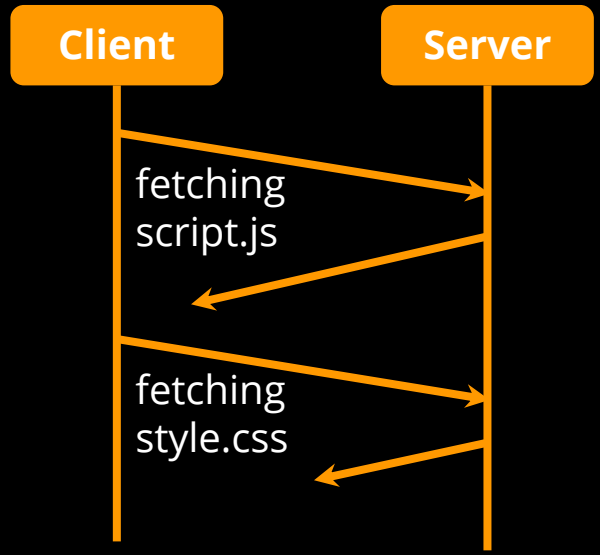
```
<head>
<script src="script.js"></script>
<link rel="stylesheet" type="text/css" href="style.css"/>
</head>
<body>Hello World!</body>
```

Rendering is blocked.



```
<head>
<script src="script.js"></script>
<link rel="stylesheet" type="text/css" href="style.css"/>
</head>
<body>Hello World!</body>
```

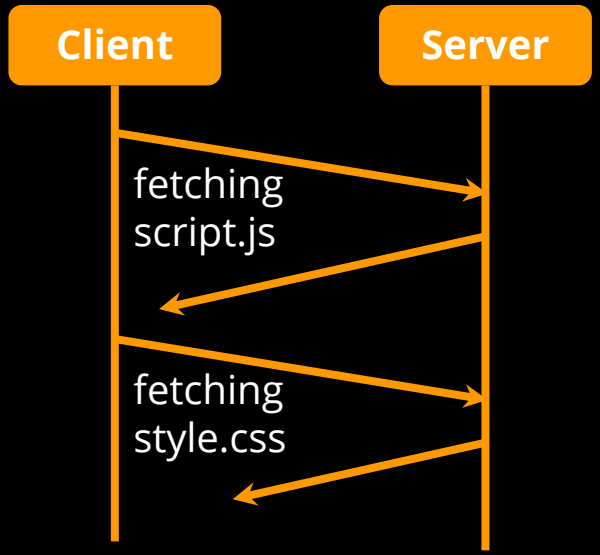
Rendering is blocked.



```
<head>
<script src="script.js"></script>
<link rel="stylesheet" type="text/css" href="style.css"/>
</head>
```

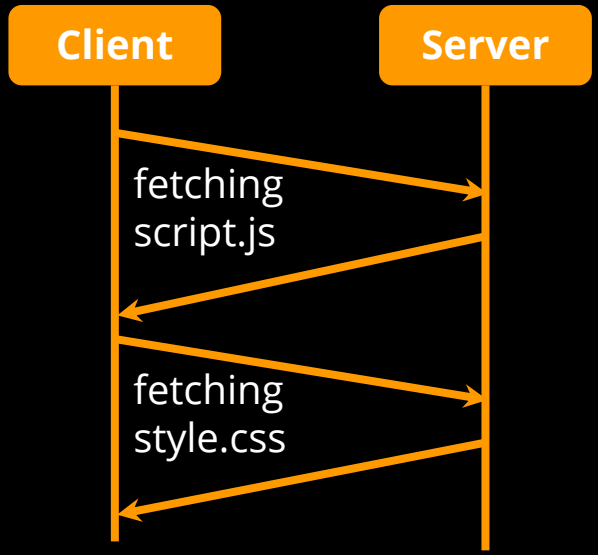
```
<body>Hello World!</body>
```

Rendering is blocked.



```
<head>
<script src="script.js"></script>
<link rel="stylesheet" type="text/css" href="style.css"/>
</head>
<body>Hello World!</body>
```

Rendering starts!



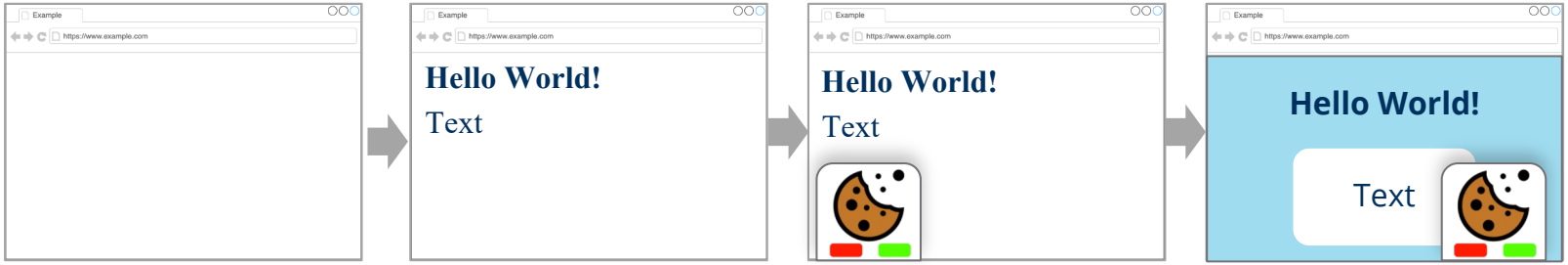
Render Blocking, a resource prevents a browser from rendering a Web page

Status quo
All or nothing

Render Blocking, a resource prevents a browser from rendering a Web page

Status quo
All or nothing

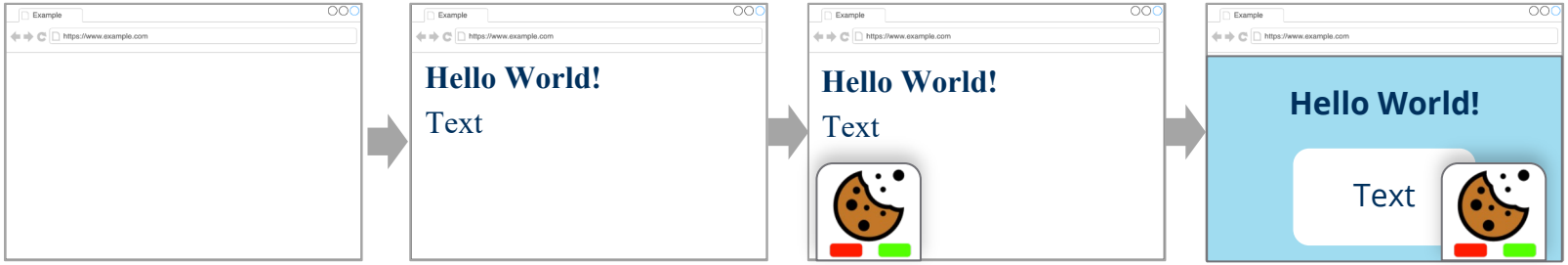
Naiv
Just don't block the render? (async/defer)



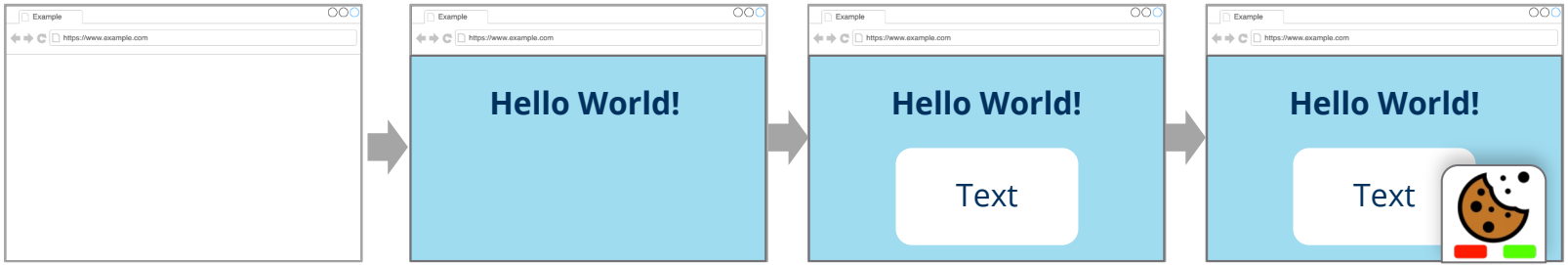
Render Blocking, a resource prevents a browser from rendering a Web page

Status quo
All or nothing

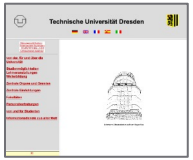
Naiv
Just don't block the render? (async/defer)



What we want
Progressive content



Real-world example of Render Blocking

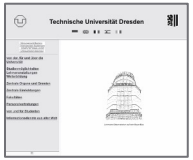


```
<body bgcolor="#e0e0e0" text="#0A0A0A" link="#bF0000" vlink="#ff0000" alink="#000000">
<center> <table border="0" width="100%" height="100%">
<frameset rows="23%,77%">
```

- No JavaScript
- CSS inline (technically render blocking)
- Fast due to focus on content

1997 2010 2023

Real-world example of Render Blocking



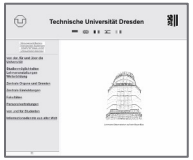
```
<link rel="stylesheet" type="text/css" href="http://tu-
dresden.de/portal_css/Site%20Default/ploneStyles9374.css"/>
<script type="text/javascript" src="http://tu-dresden.de/portal_javascripts/Site%20Default/ploneScripts7921.js">
```

- No JavaScript
- CSS inline (technically render blocking)
- Fast due to focus on content

- JavaScript from CMS
- Minimal user control over included code
- both JS and CSS fully render-blocking (missing async/defer)

1997 — 2010 — 2023

Real-world example of Render Blocking



```
<link rel="stylesheet" type="text/css" href="https://tu-dresden.de/portal_css/tud.theme.webcms2/resourceplone.formwidget.contenttreecontenttree-cachekey-7a[...]df2e.css"/>
<script type="text/javascript" src="https://tu-dresden.de/portal_javascripts/tud.theme.webcms2/popupforms-cachekey-1c4[...]112f.js"></script>
```

- No JavaScript
- CSS inline (technically render blocking)
- Fast due to focus on content

- JavaScript from CMS
- Minimal user control over included code
- both JS and CSS fully render-blocking (missing async/defer)

- JavaScript and CSS in bundles and collections
- Minimal user control over included code
- JS & CSS fully render-blocking (except tracking)

1997 — 2010 — 2023

Summary

Websites don’t load significantly faster despite protocol improvements

Render blocking is the core problem

Solutions such as async/defer exist but aren’t always easy to use

Code is combined into bundles

What is this talk about?

What slows down content presentation?
Beyond transmission delay, the problem of render blocking.

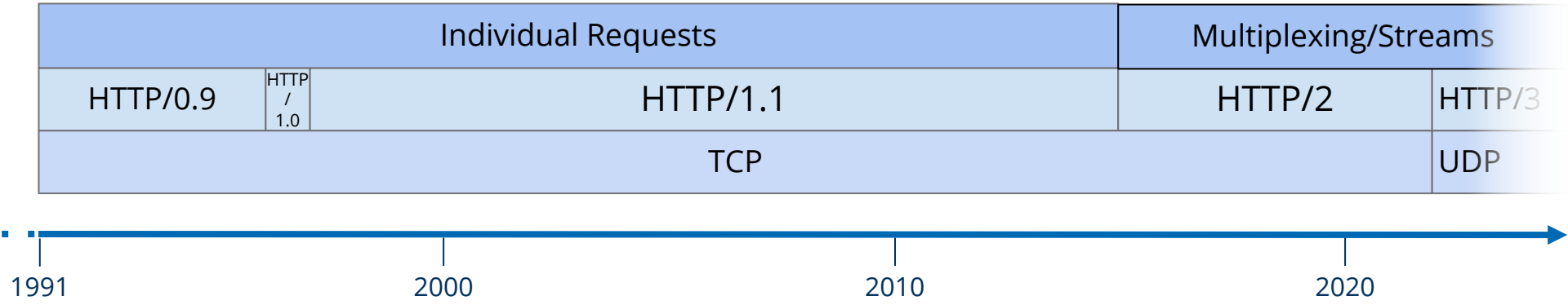
Why are we faced with the situation?
A brief historic recap.

How can we improve the situation?
Split bundles and stream content pieces.

The problem: protocol vs. content

Solution: **bundling**

Penalty for every request: needed full TCP handshake

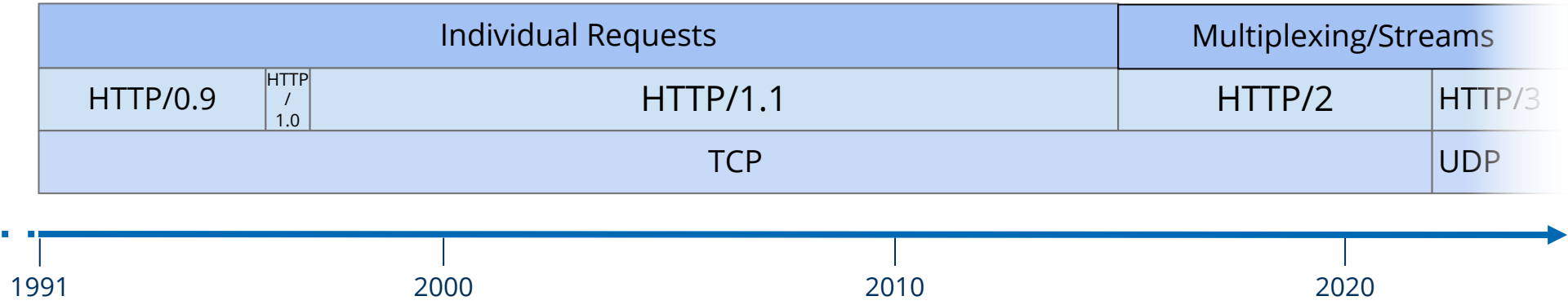


The problem: protocol vs. content

Solution: **bundling**

Penalty for every request: needed full TCP handshake

Less penalty for requests due to pipelining and keep-alive.
Still issues, due to buggy proxies and fixed request order



The problem: protocol vs. content

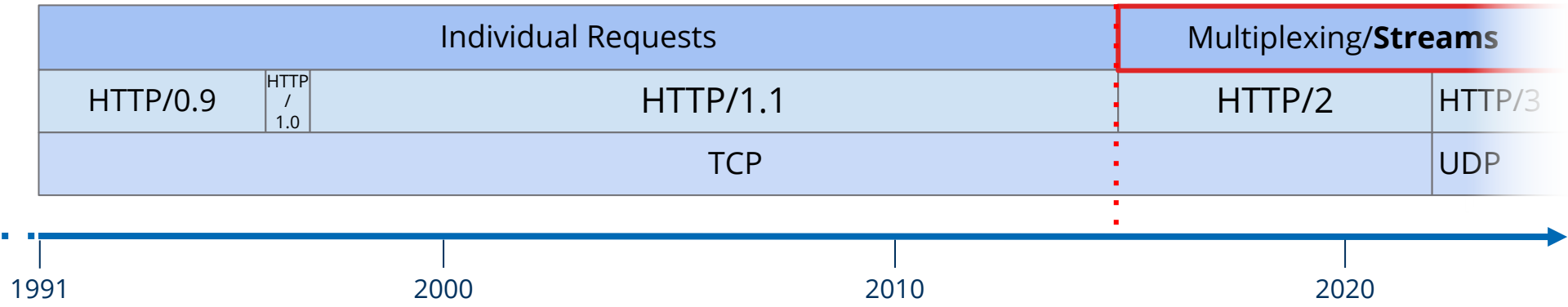
Solution: **bundling**

Solution: **splitting**

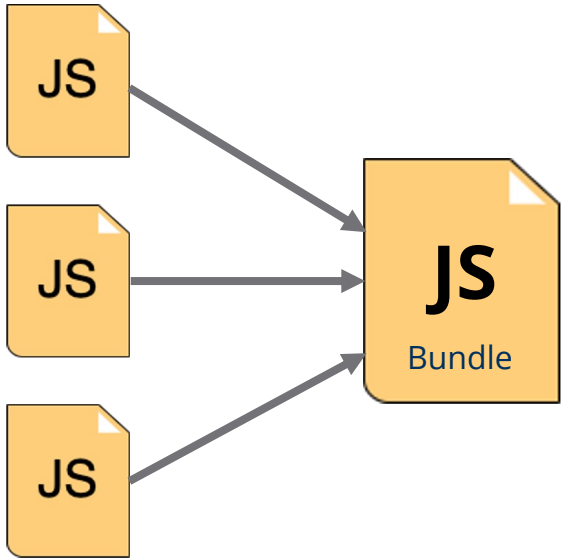
Penalty for every request: needed full TCP handshake

Less penalty for requests due to pipelining and keep-alive.
Still issues, due to buggy proxies and fixed request order

(almost) no penalty for requests due to 0-RTT, multiplexing and streams

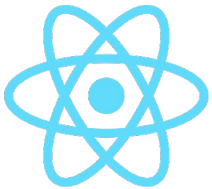


What are bundles?



- Started with **Browserify**, ca. 2013
- Today: most popular bundler: **Webpack**
- **By default: produces render-blocking bundles**

- **Webpack** is used by



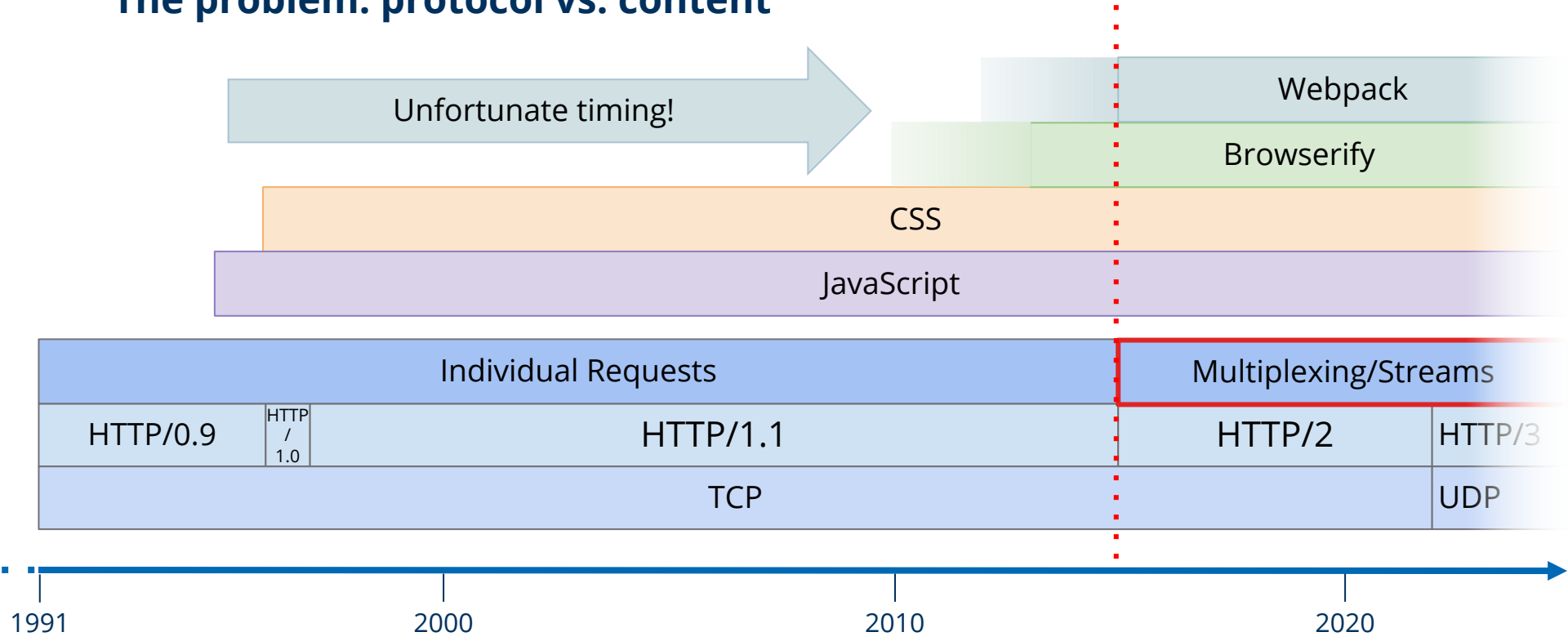
React

Angular

Vue.js

.. and many more

The problem: protocol vs. content



Bundling is an anti-pattern in HTTP/2

… but we still use bundling and render-blocking resources!

> “In HTTP/2, [bundling] will end up impacting the download-time of other resources as well, because of the way HTTP/2 works.”
>
> —Erwin Hofman, 2022

[https://www.erwinhofman.com/blog/two-main-performance-debts-of-http1/]

What is this talk about?

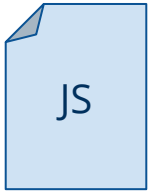
What slows down content presentation?
Beyond transmission delay, the problem of render blocking.

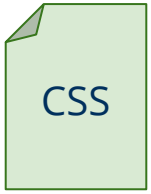
Why are we faced with the situation?
A brief historic recap.

How can we improve the situation?
Split bundles and stream content pieces.

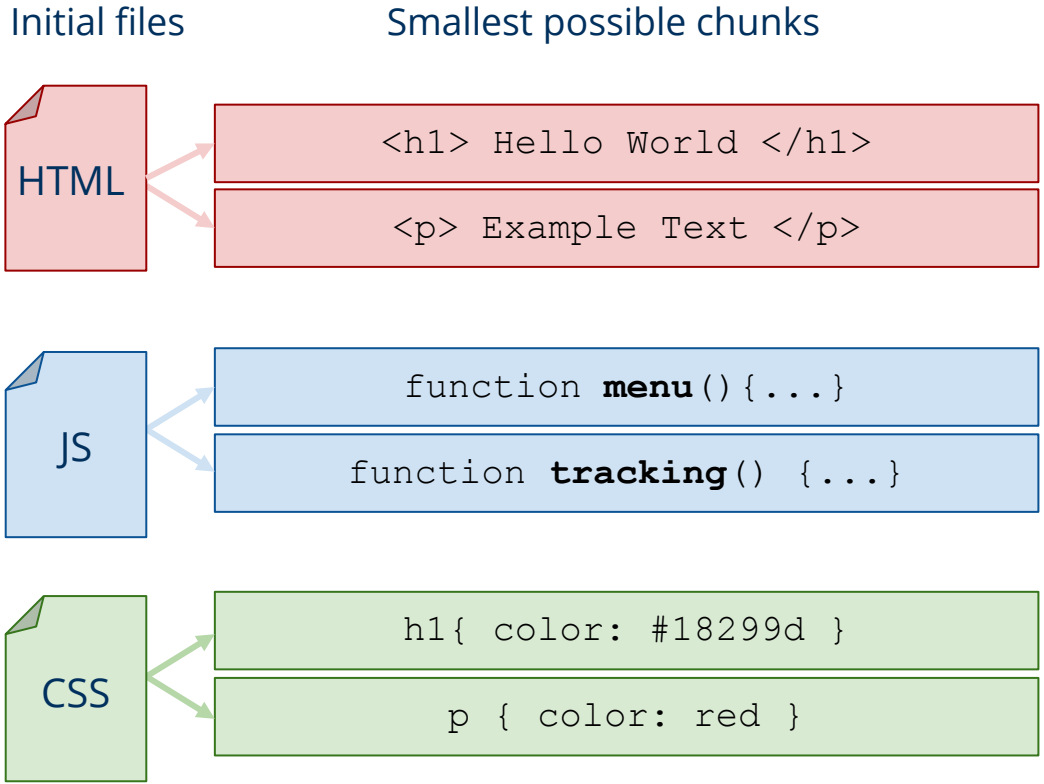
What if … we split as much as possible?

Initial files

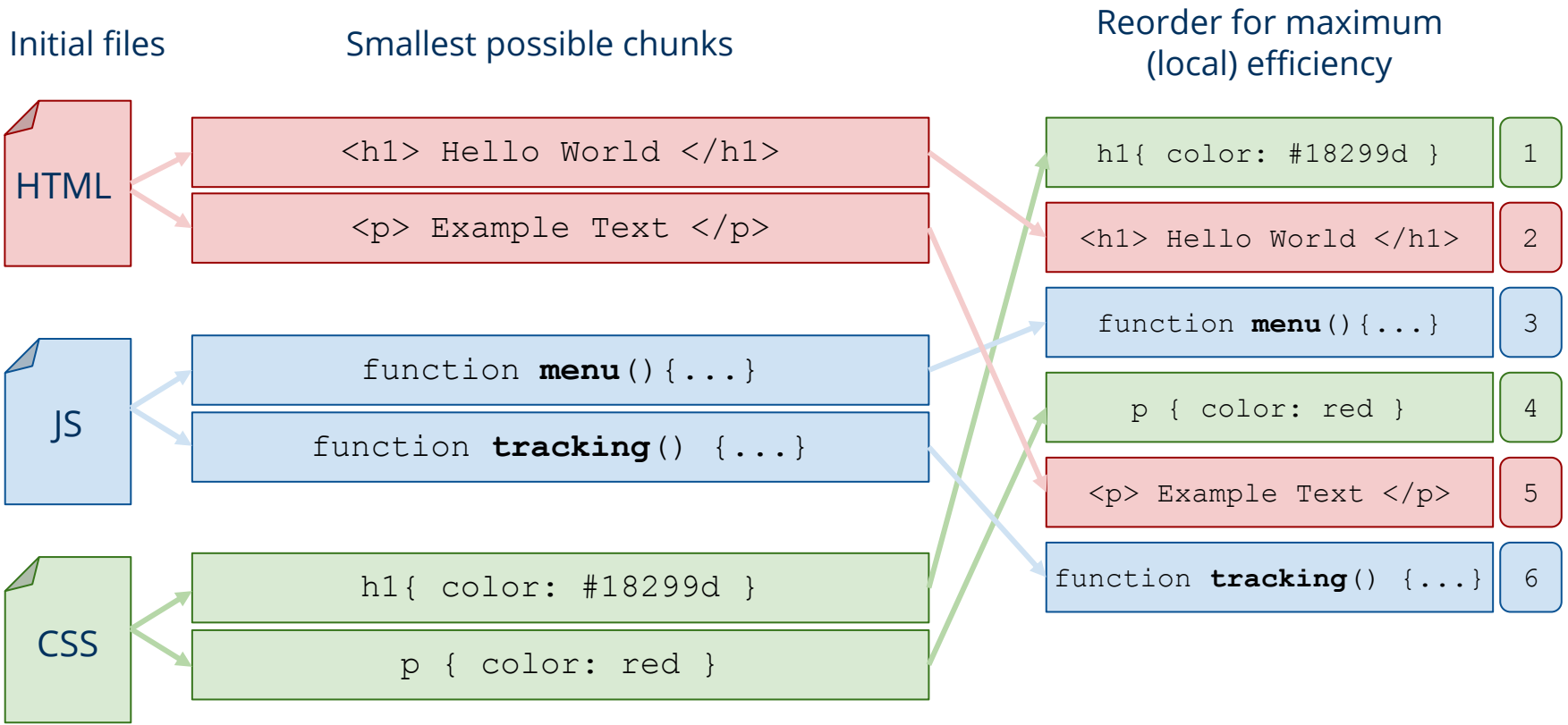




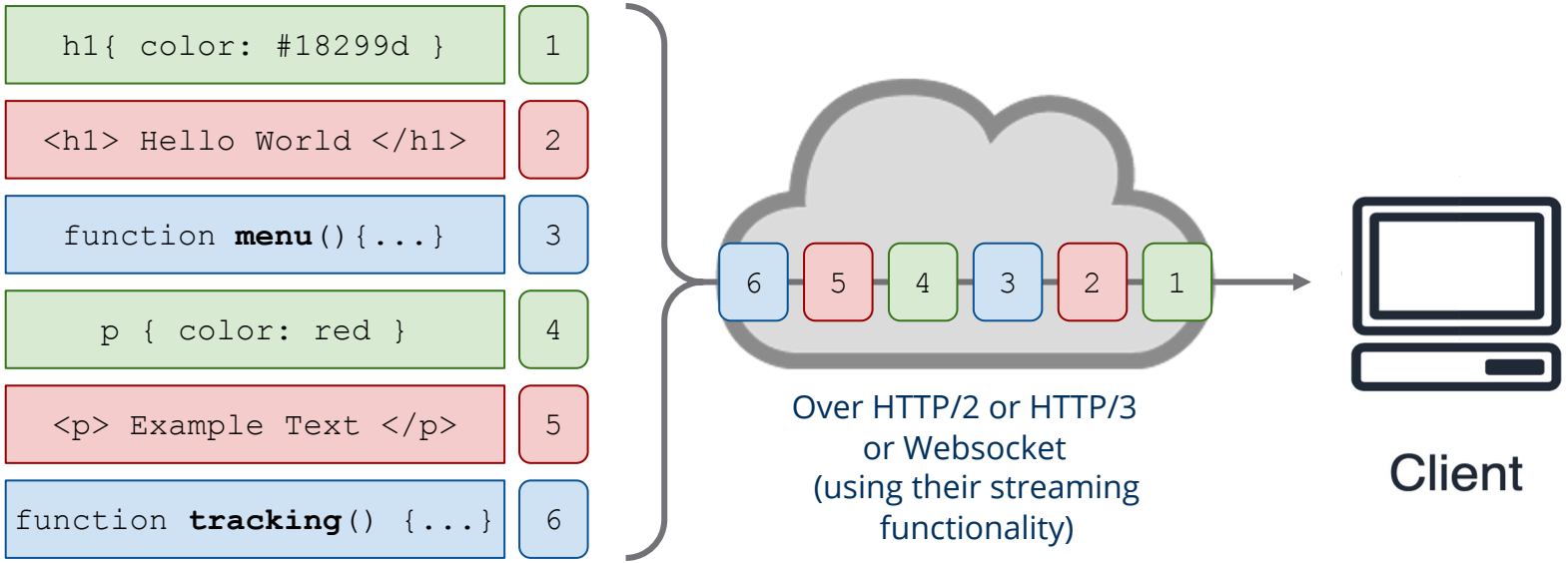
What if … we split as much as possible?



What if … we split as much as possible?



What if … we stream splitted content?



Proposed streaming in practice
An example, www.solarenergie.de

Splitting, reordering, and streaming improve Web experience

Faster First Contentful Paint

Independent of actual length of web page
Improvements especially at slow network speeds

Faster total transfer

Content is ordered by priority leading to improved efficiency, as necessary code is transferred first
“Forces” efficiency

Improved UX

Better user experience due to reduced/removed layout shift
Faster interaction with web page

On-going research

Render & split **CSS** into render critical & uncritical parts

Vogel, Lucas, and Thomas Springer. "Speed Up the Web with Universal CSS Rendering." *International Conference on Web Engineering*. Cham: Springer Nature Switzerland, 2023.

split **JS** into multiple parts delay execution

Vogel, Lucas, and Thomas Springer. "Waiter and AUTRATAC: Don't Throw It Away, Just Delay!." *International Conference on Web Engineering*. Cham: Springer Nature Switzerland, 2023.

Render **HTML** (with inline CSS and JS) **Reorder to be streamable** and render-able on the fly

Vogel, Lucas, and Thomas Springer. "How Streaming Can Improve the World (Wide Web)." *Companion Proceedings of the ACM Web Conference 2023*. 2023.

Conclusion

The Web is unnecessarily slow

Key takeaways

- Presenting web content fast is more than reducing transmission delays
- Bundles and render-blocking files slow down the Web

Proposed solution

- Consider loading behavior of web pages as a process over time and not file-based fetching
- Split content and stream pieces to send what is needed

Outlook

Next steps

- Fully automatic splitting of JavaScript
- Integration into current developer processes
- Streaming other media such as images

Open challenges

- How to leverage HTTP/3 features fully to support splitting and streaming best?
- Which websites would benefit from streaming most?
- What is the impact of current Web caches on streaming of content chunks?
- How to support multi-source, multi-path deployments?

Questions?

Sources

Slides: Web content changed drastically

- [left] http://web.archive.org/web/19970716153203/http://www.tu-dresden.de/ [online, Nov. 19th, 2023]
- [middle] http://web.archive.org/web/20100717183543/http://tu-dresden.de/ [online, Nov. 19th, 2023]
- [right] http://web.archive.org/web/20230716104002/https://tu-dresden.de/ [online, Nov. 19th, 2023]

Slide: Why does speed matter?

- [1] afaco GmbH. Breitbandmessung Jahresbericht 2018/19. 2020. url: https://download.breitbandmessung.de/bbm/Breitbandmessung_Jahresbericht_2018_2019.pdf (besucht am 03. 04. 2020).
- [2] Fiona Fui-Hoon Nah. „A study on tolerable waiting time: how long are web users willing to wait?“ In: Behaviour & Information Technology 23.3 (2004), S. 153– 163.
- [3] Greg Linden. Marissa Mayer at Web 2.0. url: http://glinden.blogspot.de/2006/11/marissa-mayer-at-web-20.html (besucht am 12.01.2021).

Slide: What improved?

- [left] https://blog.cloudflare.com/content/images/2018/07/QUIC-Badge-Dark-RGB-Horiz.png [online, Nov. 19th, 2023]
- [top right] https://www.akamai.com/content/dam/site/en/images/logo/akamai-logo-og-default.png [online, Nov. 19th, 2023]
- [bottom right] Image by Ditaucis from Pixabay [online, Nov. 19th, 2023]

Slide: Web page size vs. loading time (Desktop)

- [left] https://httparchive.org/reports/loading-speed [online, Nov. 19th, 2023]
- [right] https://httparchive.org/reports/state-of-the-web [online, Nov. 19th, 2023]

Slide: What are bundles?

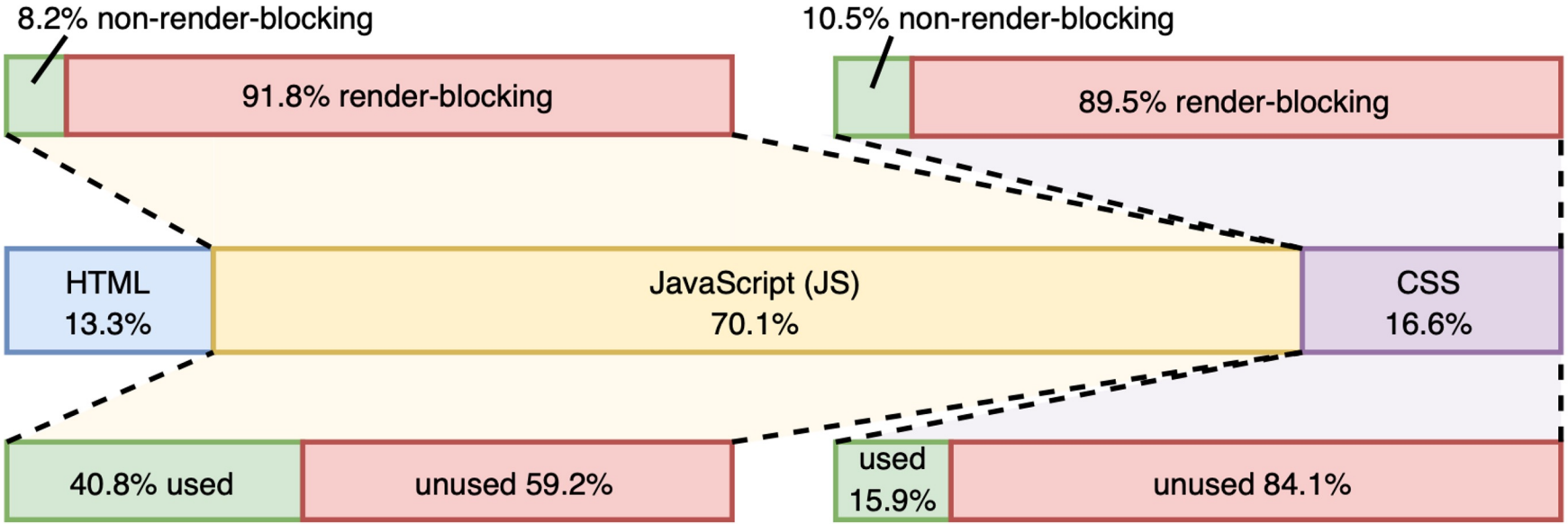
- [left, react logo] react.dev [online, Nov. 19th, 2023]
- [middle, angular logo] angular.io [online, Nov. 19th, 2023]
- [right, vue.js logo] vuejs.org [online, Nov. 19th, 2023]

Slide: How web pages can be streamed

- Vogel, Lucas, and Thomas Springer. "How Streaming Can Improve the World (Wide Web)." Companion Proceedings of the ACM Web Conference 2023. 2023.

Backup

Render blocking code. Average of top-10k Websites (Tranco).



Source:
Vogel, Lucas, and Thomas Springer. "User Acceptance of Modified Web Page Loading Based on Progressive Streaming." Web Engineering: 22nd International Conference, ICWE 2022, Bari, Italy, July 5–8, 2022, Proceedings. Cham: Springer International Publishing, 2022.