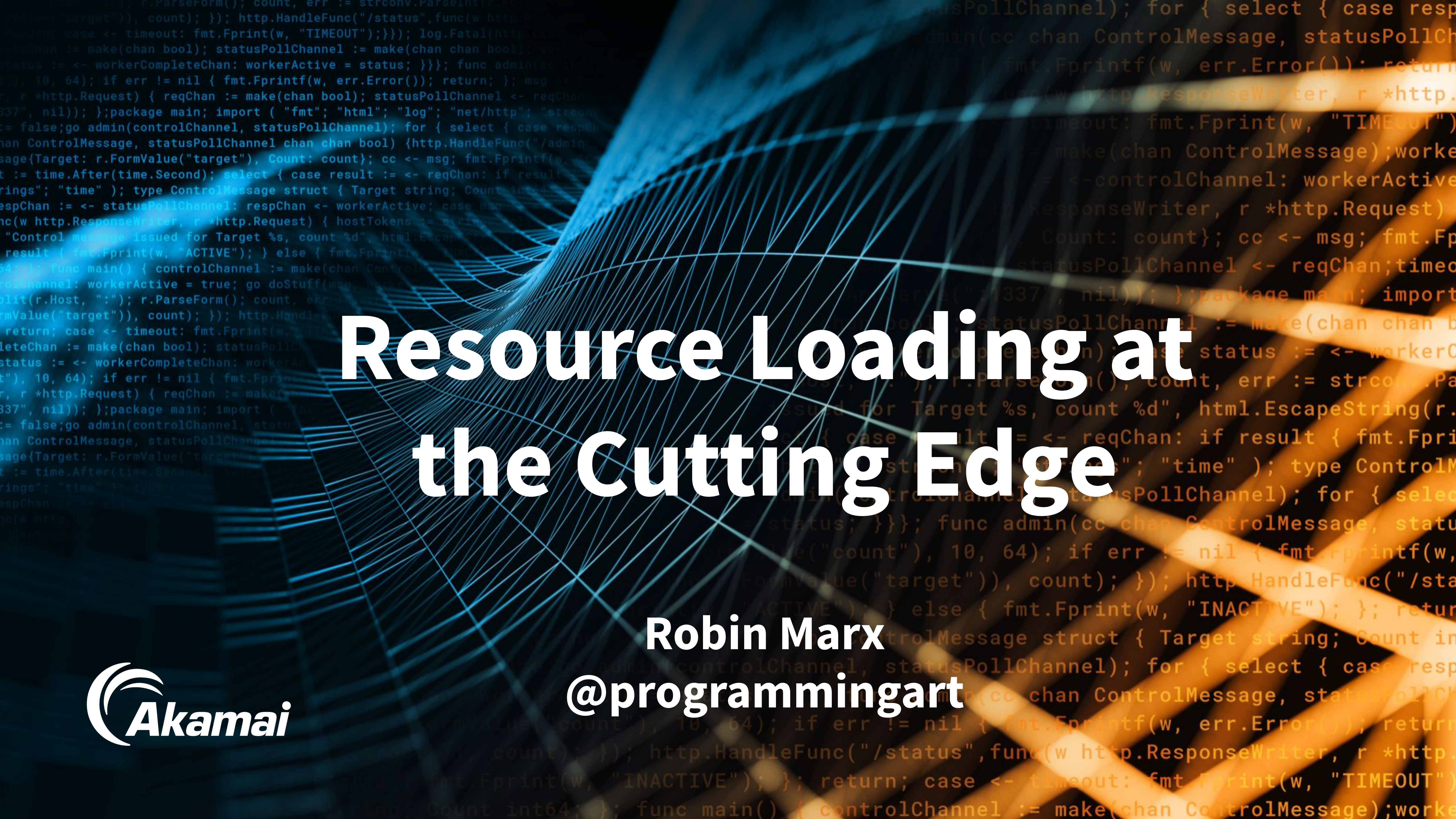


Resource Loading at the Cutting Edge

Robin Marx
@programmingart

Akamai

Resource Loading at *the Cutting Edge*

Robin Marx
@programmingart

Akamai

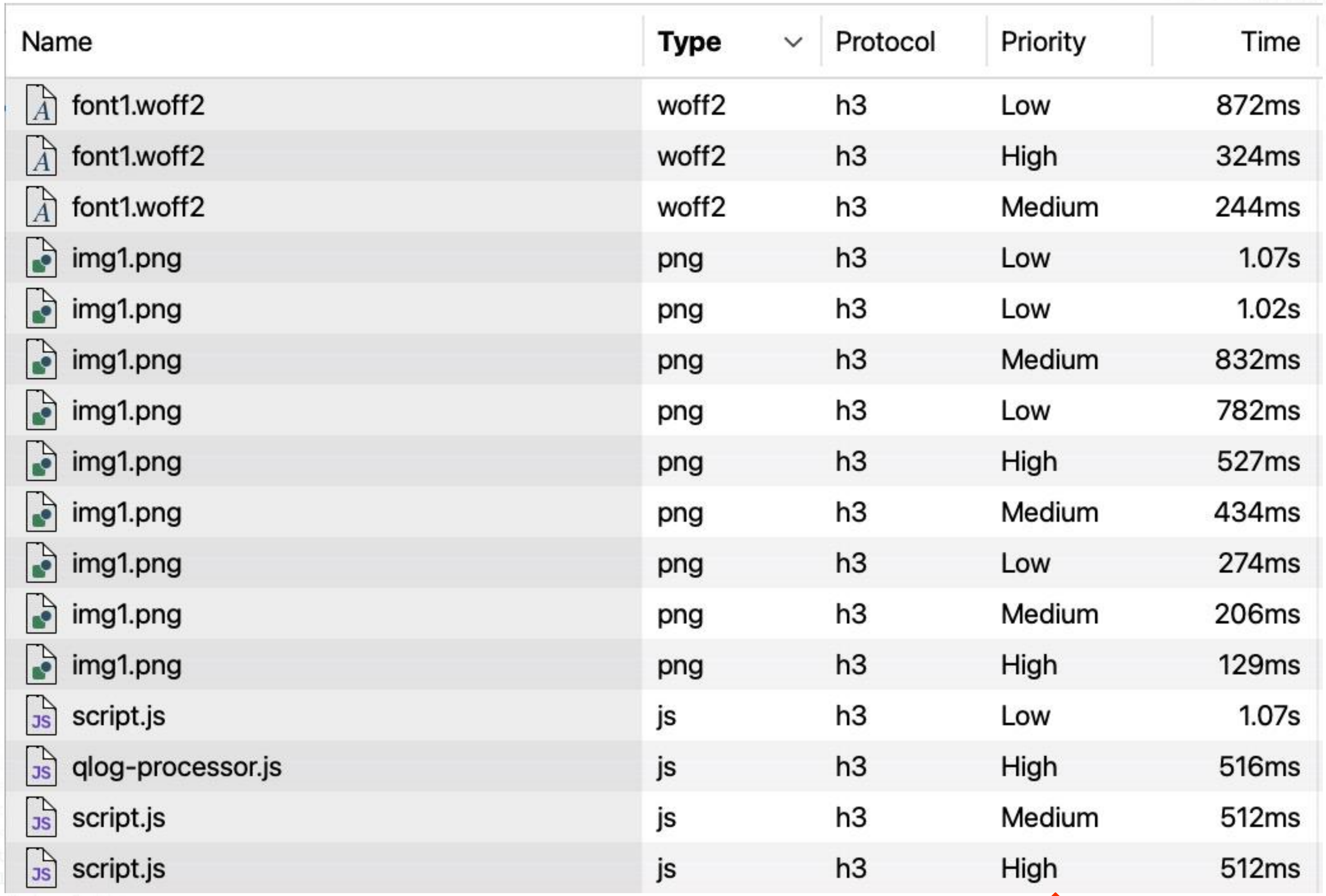
HISTORICAL EUROPEAN MARTIAL ARTS

HTTP/3
Fetchpriority
Preload
103 Early Hints
Async
Speculative Loading
Lazy Loading
0-RTT
TLS 1.3
Defer
BBRv2
Extensible Priotization

THIS IS FINE.

THIS IS FINE.

Priorities



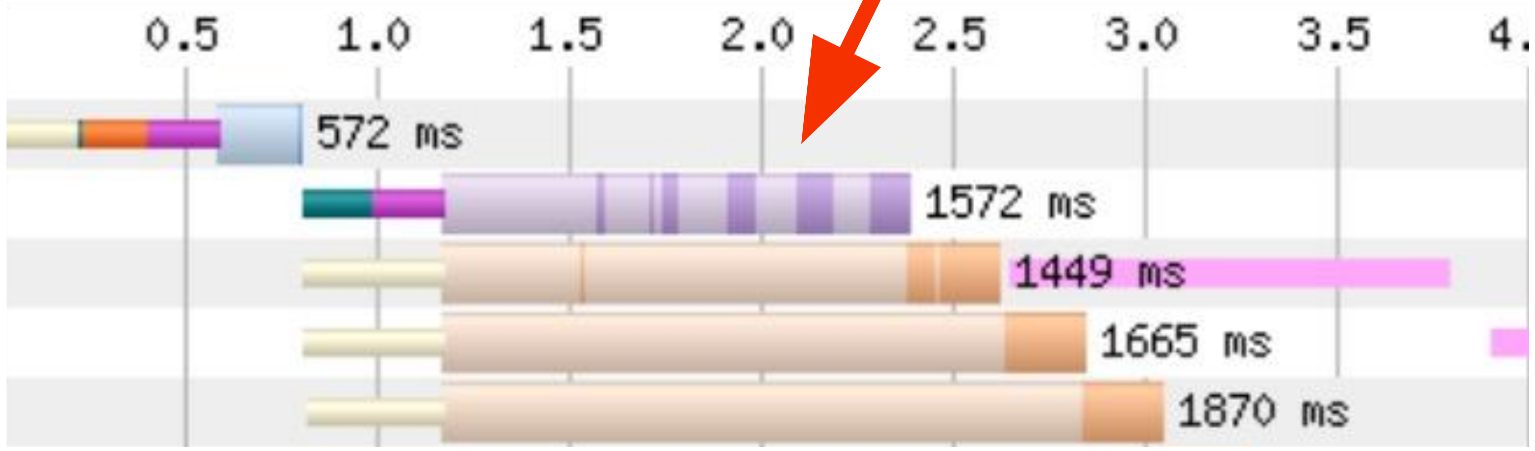
Name	Type	Protocol	Priority	Time
font1.woff2	woff2	h3	Low	872ms
font1.woff2	woff2	h3	High	324ms
font1.woff2	woff2	h3	Medium	244ms
img1.png	png	h3	Low	1.07s
img1.png	png	h3	Low	1.02s
img1.png	png	h3	Medium	832ms
img1.png	png	h3	Low	782ms
img1.png	png	h3	High	527ms
img1.png	png	h3	Medium	434ms
img1.png	png	h3	Low	274ms
img1.png	png	h3	Medium	206ms
img1.png	png	h3	High	129ms
script.js	js	h3	Low	1.07s
qlog-processor.js	js	h3	High	516ms
script.js	js	h3	Medium	512ms
script.js	js	h3	High	512ms

Akamai Experience the Edge

Bandwidth is not endless

New connections begin in **Slow Start** phase

1st Round Trip: 14 - 50 KB
2nd Round Trip: 28 - 100 KB
3rd Round Trip: 56 - 200 KB
...



https://www.tunetheweb.com/blog/critical-resources-and-the-first-14kb/

Loading order matters…

Server needs to fill 200 KB with one or more of:

- HTML remainder
- script.js
- styles.css
- image.jpg
- content.json
- font.woff2
- …

HTTP/2

HTTP/3

Single connection,
Multiple resources

…but the server doesn't have enough context

Server needs to fill 200 KB with one or more of:

- HTML remainder	is render-blocked?
- script.js	async/defer?
- styles.css	media = print?
- image.jpg	visible in viewport?
- content.json	Client Side Rendering?
- font.woff2	
- …	

HTTP/2

HTTP/3

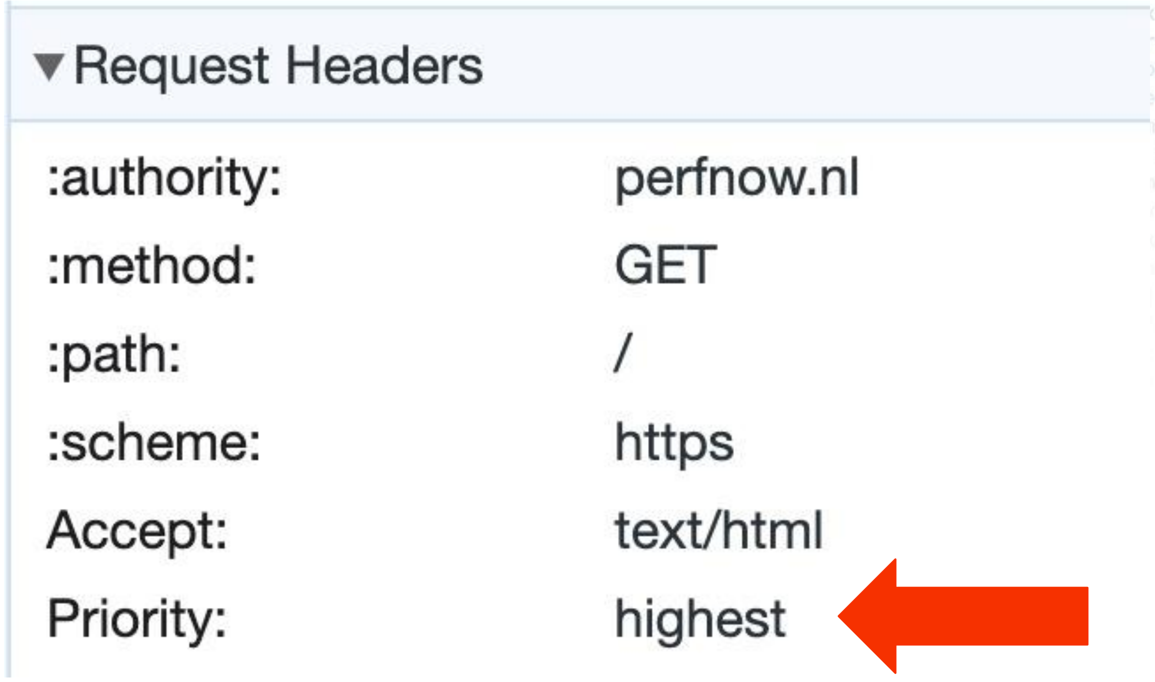
Single connection,
Multiple resources

The Browser tells the Server what to do

- HTML	: **highest** priority
- script.js	: **medium** priority
- styles.css	: **high** priority
- image.jpg	: **lowest** priority

How are Priorities communicated?

- HTML : **highest** priority
- script.js : **medium** priority
- styles.css : **high** priority
- image.jpg : **lowest** priority

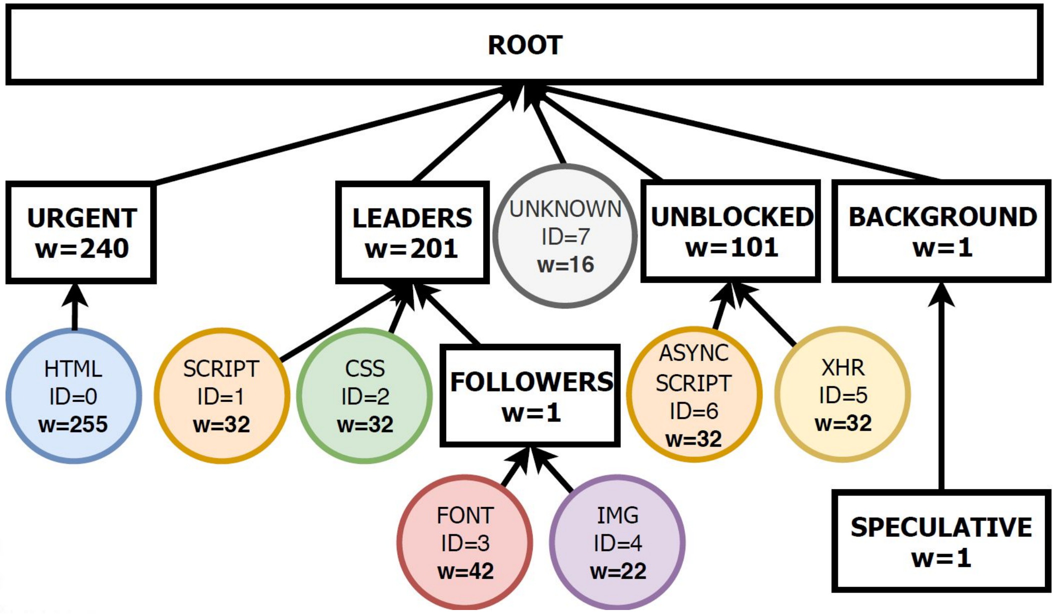


▼Request Headers

:authority:	perfnow.nl
:method:	GET
:path:	/
:scheme:	https
Accept:	text/html
Priority:	highest

Akamai Experience the Edge

How are Priorities communicated?

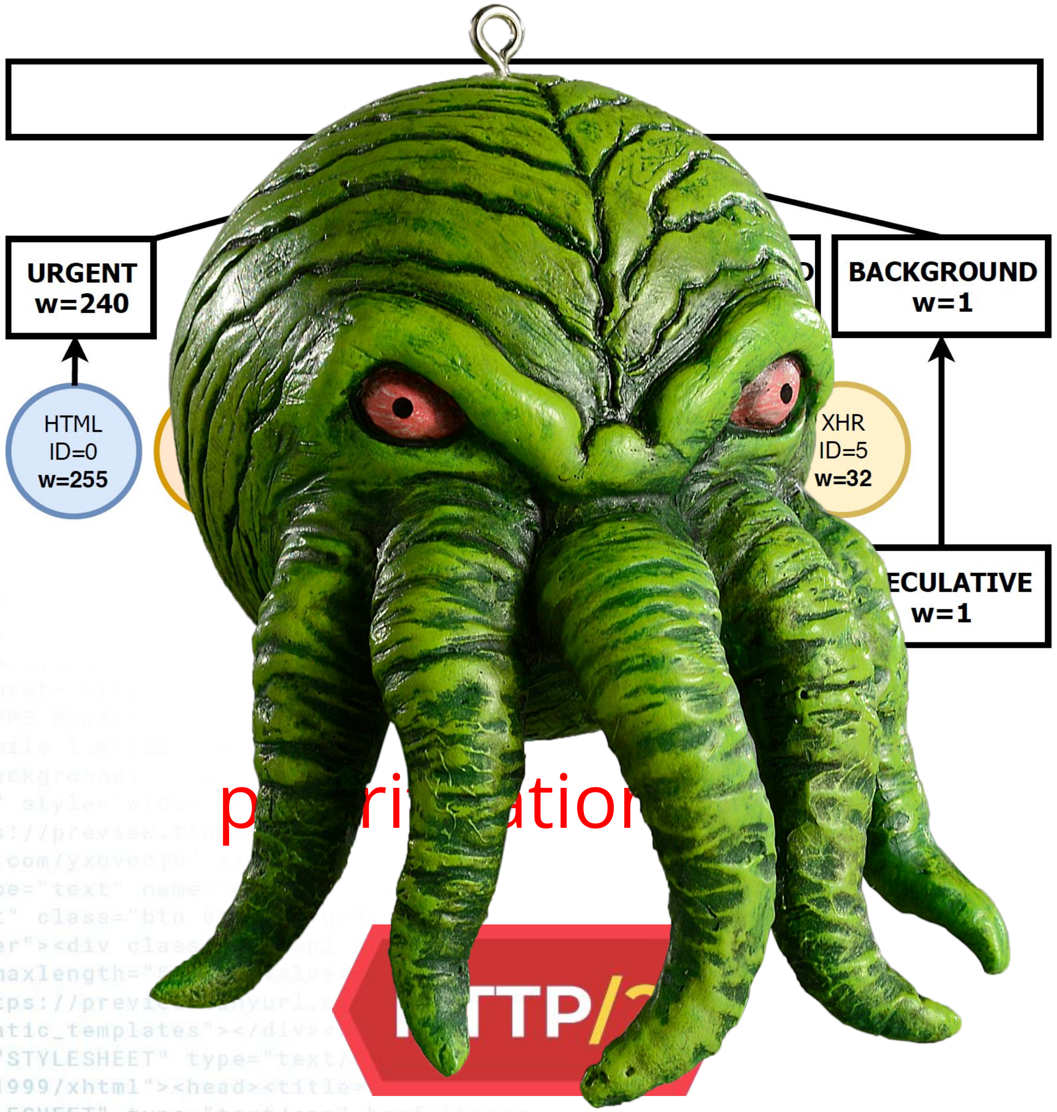


prioritization tree

HTTP/2

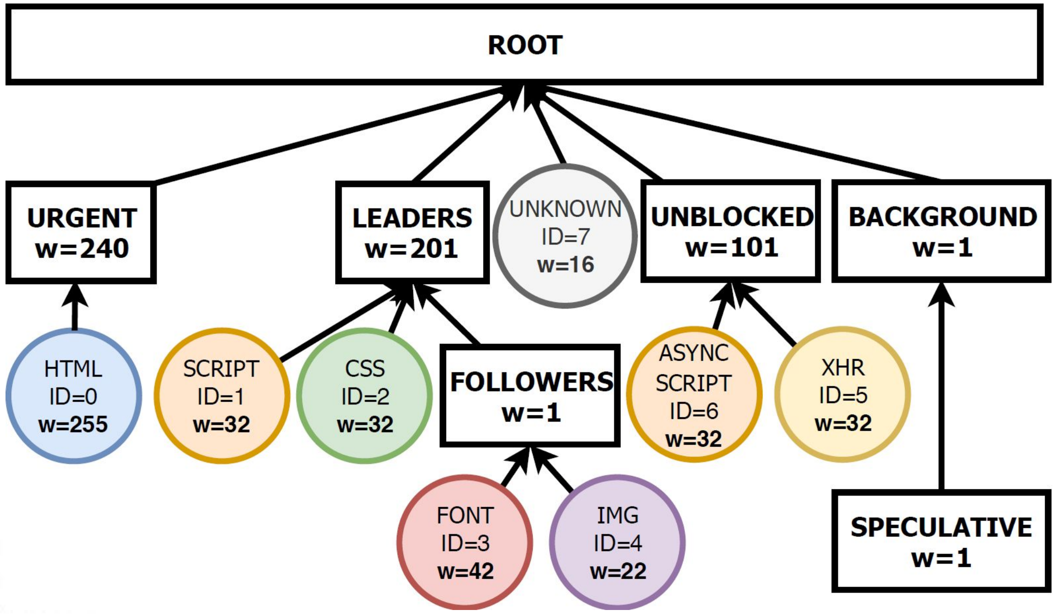
Akamai Experience the Edge

How are Priorities communicated?

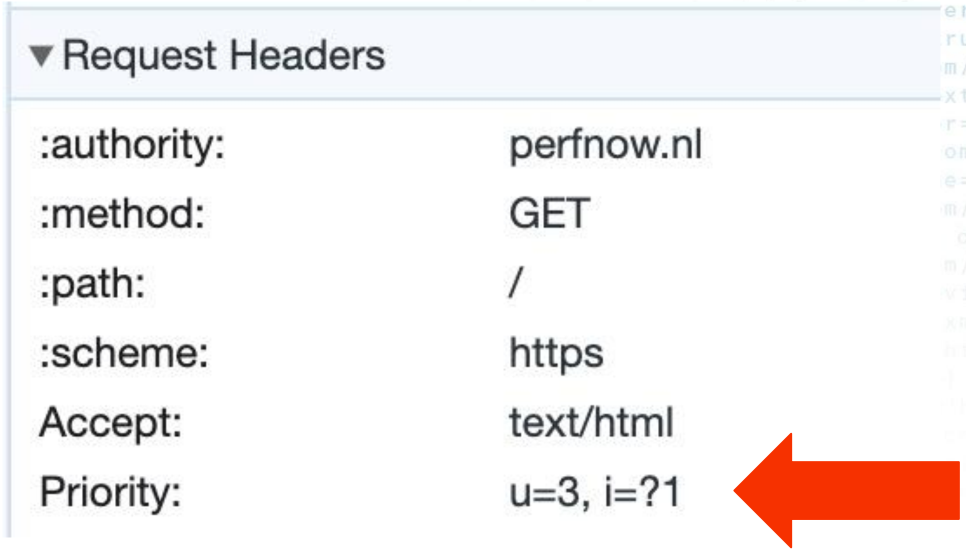


https://horrornaments.com/products/cthulhu

How are Priorities communicated?



prioritization tree



▼ Request Headers

:authority:	perfnow.nl
:method:	GET
:path:	/
:scheme:	https
Accept:	text/html
Priority:	u=3, i=?1

urgency:
from 0 (highest) to 7 (lowest)
incremental: 0 or 1

HTTP/2

HTTP/3

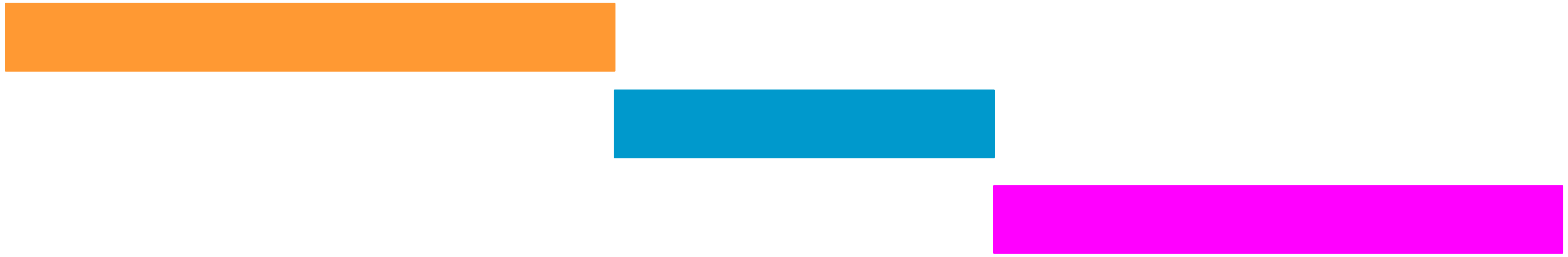
https://www.rfc-editor.org/rfc/rfc9218.html

Sequential vs Incremental

u=2, i=0

u=2, i=0

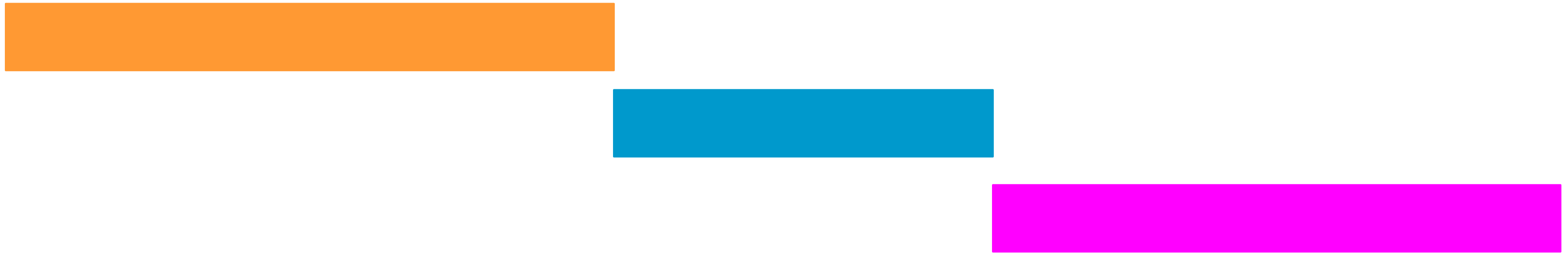
u=2, i=0



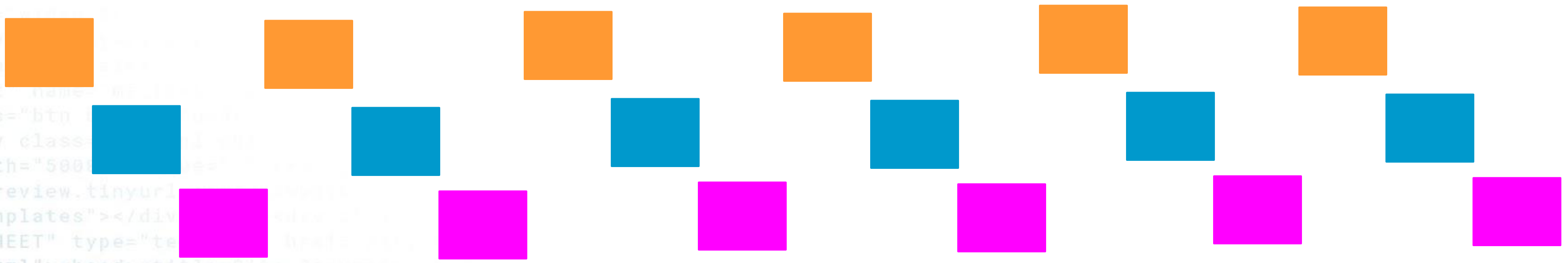
Sequential

Sequential vs Incremental

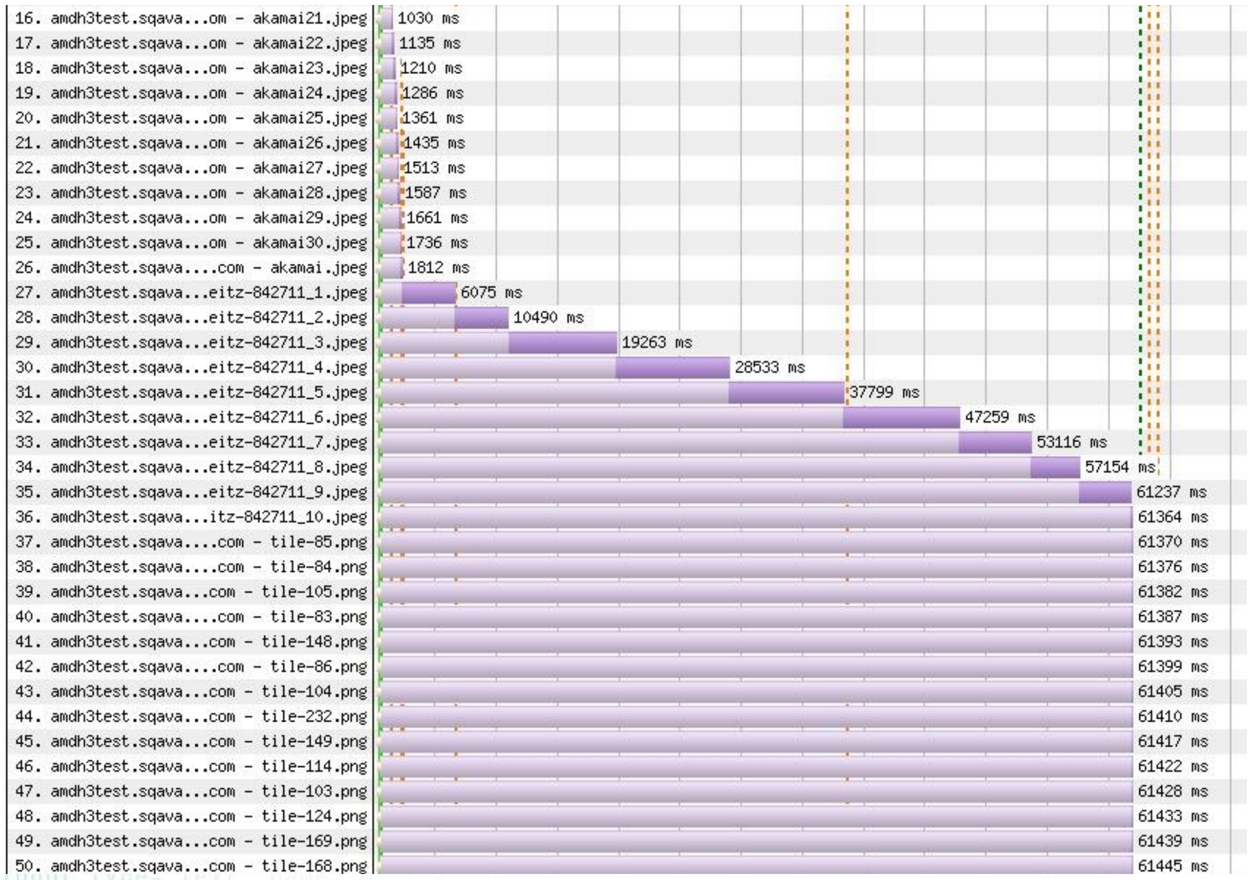
u=2, i=0
u=2, i=0
u=2, i=0



u=2, i=1
u=2, i=1
u=2, i=1



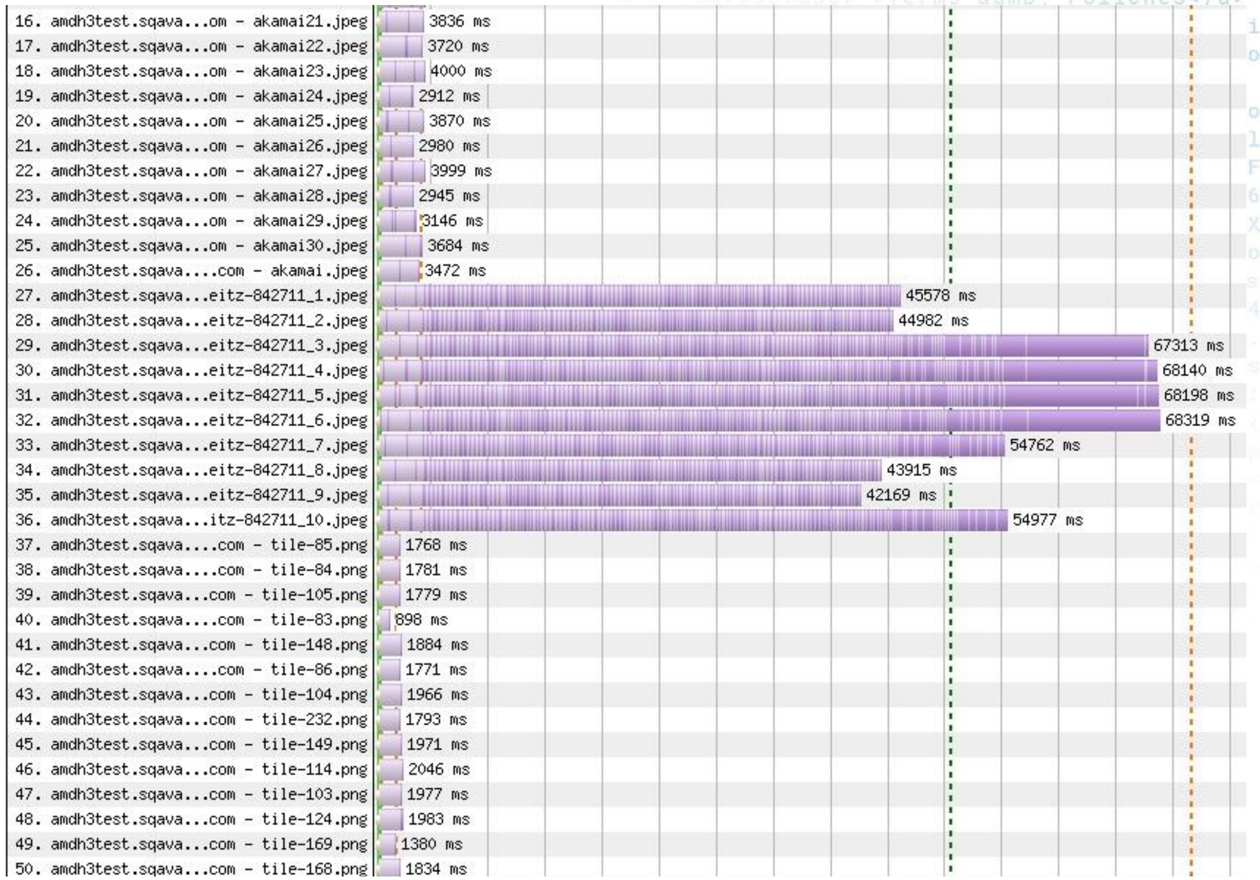
Sequential



16. amdh3test.sqava...om - akamai21.jpeg	1030 ms
17. amdh3test.sqava...om - akamai22.jpeg	1135 ms
18. amdh3test.sqava...om - akamai23.jpeg	1210 ms
19. amdh3test.sqava...om - akamai24.jpeg	1286 ms
20. amdh3test.sqava...om - akamai25.jpeg	1361 ms
21. amdh3test.sqava...om - akamai26.jpeg	1435 ms
22. amdh3test.sqava...om - akamai27.jpeg	1513 ms
23. amdh3test.sqava...om - akamai28.jpeg	1587 ms
24. amdh3test.sqava...om - akamai29.jpeg	1661 ms
25. amdh3test.sqava...om - akamai30.jpeg	1736 ms
26. amdh3test.sqava....com - akamai.jpeg	1812 ms
27. amdh3test.sqava...eitz-842711_1.jpeg	6075 ms
28. amdh3test.sqava...eitz-842711_2.jpeg	10490 ms
29. amdh3test.sqava...eitz-842711_3.jpeg	19263 ms
30. amdh3test.sqava...eitz-842711_4.jpeg	28533 ms
31. amdh3test.sqava...eitz-842711_5.jpeg	37799 ms
32. amdh3test.sqava...eitz-842711_6.jpeg	47259 ms
33. amdh3test.sqava...eitz-842711_7.jpeg	53116 ms
34. amdh3test.sqava...eitz-842711_8.jpeg	57154 ms
35. amdh3test.sqava...eitz-842711_9.jpeg	61237 ms
36. amdh3test.sqava...itz-842711_10.jpeg	61364 ms
37. amdh3test.sqava....com - tile-85.png	61370 ms
38. amdh3test.sqava....com - tile-84.png	61376 ms
39. amdh3test.sqava...com - tile-105.png	61382 ms
40. amdh3test.sqava....com - tile-83.png	61387 ms
41. amdh3test.sqava...com - tile-148.png	61393 ms
42. amdh3test.sqava....com - tile-86.png	61399 ms
43. amdh3test.sqava...com - tile-104.png	61405 ms
44. amdh3test.sqava...com - tile-232.png	61410 ms
45. amdh3test.sqava...com - tile-149.png	61417 ms
46. amdh3test.sqava...com - tile-114.png	61422 ms
47. amdh3test.sqava...com - tile-103.png	61428 ms
48. amdh3test.sqava...com - tile-124.png	61433 ms
49. amdh3test.sqava...com - tile-169.png	61439 ms
50. amdh3test.sqava...com - tile-168.png	61445 ms

Large files delay smaller ones behind

Incremental

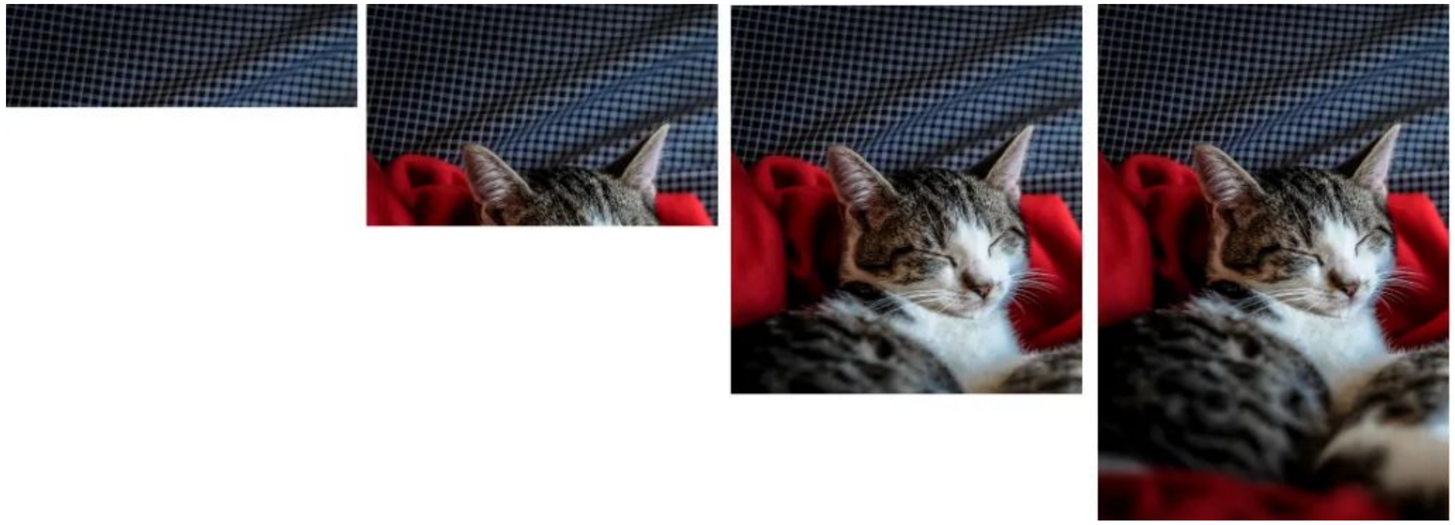


16. amdh3test.sqava...om - akamai21.jpeg	3836 ms
17. amdh3test.sqava...om - akamai22.jpeg	3720 ms
18. amdh3test.sqava...om - akamai23.jpeg	4000 ms
19. amdh3test.sqava...om - akamai24.jpeg	2912 ms
20. amdh3test.sqava...om - akamai25.jpeg	3870 ms
21. amdh3test.sqava...om - akamai26.jpeg	2980 ms
22. amdh3test.sqava...om - akamai27.jpeg	3999 ms
23. amdh3test.sqava...om - akamai28.jpeg	2945 ms
24. amdh3test.sqava...om - akamai29.jpeg	3146 ms
25. amdh3test.sqava...om - akamai30.jpeg	3684 ms
26. amdh3test.sqava....com - akamai.jpeg	3472 ms
27. amdh3test.sqava...eitz-842711_1.jpeg	45578 ms
28. amdh3test.sqava...eitz-842711_2.jpeg	44982 ms
29. amdh3test.sqava...eitz-842711_3.jpeg	67313 ms
30. amdh3test.sqava...eitz-842711_4.jpeg	68140 ms
31. amdh3test.sqava...eitz-842711_5.jpeg	68198 ms
32. amdh3test.sqava...eitz-842711_6.jpeg	68319 ms
33. amdh3test.sqava...eitz-842711_7.jpeg	54762 ms
34. amdh3test.sqava...eitz-842711_8.jpeg	43915 ms
35. amdh3test.sqava...eitz-842711_9.jpeg	42169 ms
36. amdh3test.sqava...itz-842711_10.jpeg	54977 ms
37. amdh3test.sqava....com - tile-85.png	1768 ms
38. amdh3test.sqava....com - tile-84.png	1781 ms
39. amdh3test.sqava...com - tile-105.png	1779 ms
40. amdh3test.sqava....com - tile-83.png	898 ms
41. amdh3test.sqava...com - tile-148.png	1884 ms
42. amdh3test.sqava....com - tile-86.png	1771 ms
43. amdh3test.sqava...com - tile-104.png	1966 ms
44. amdh3test.sqava...com - tile-232.png	1793 ms
45. amdh3test.sqava...com - tile-149.png	1971 ms
46. amdh3test.sqava...com - tile-114.png	2046 ms
47. amdh3test.sqava...com - tile-103.png	1977 ms
48. amdh3test.sqava...com - tile-124.png	1983 ms
49. amdh3test.sqava...com - tile-169.png	1380 ms
50. amdh3test.sqava...com - tile-168.png	1834 ms

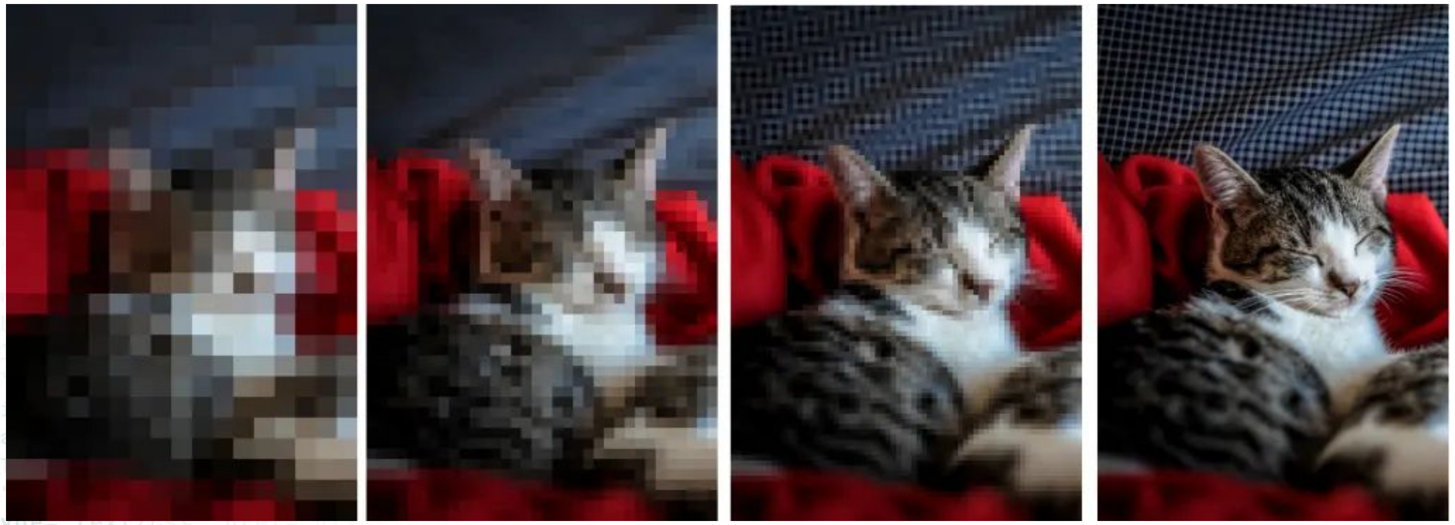
Small files come in early, barely delay larger ones

Akamai Experience the Edge

Partial downloads are sometimes useful



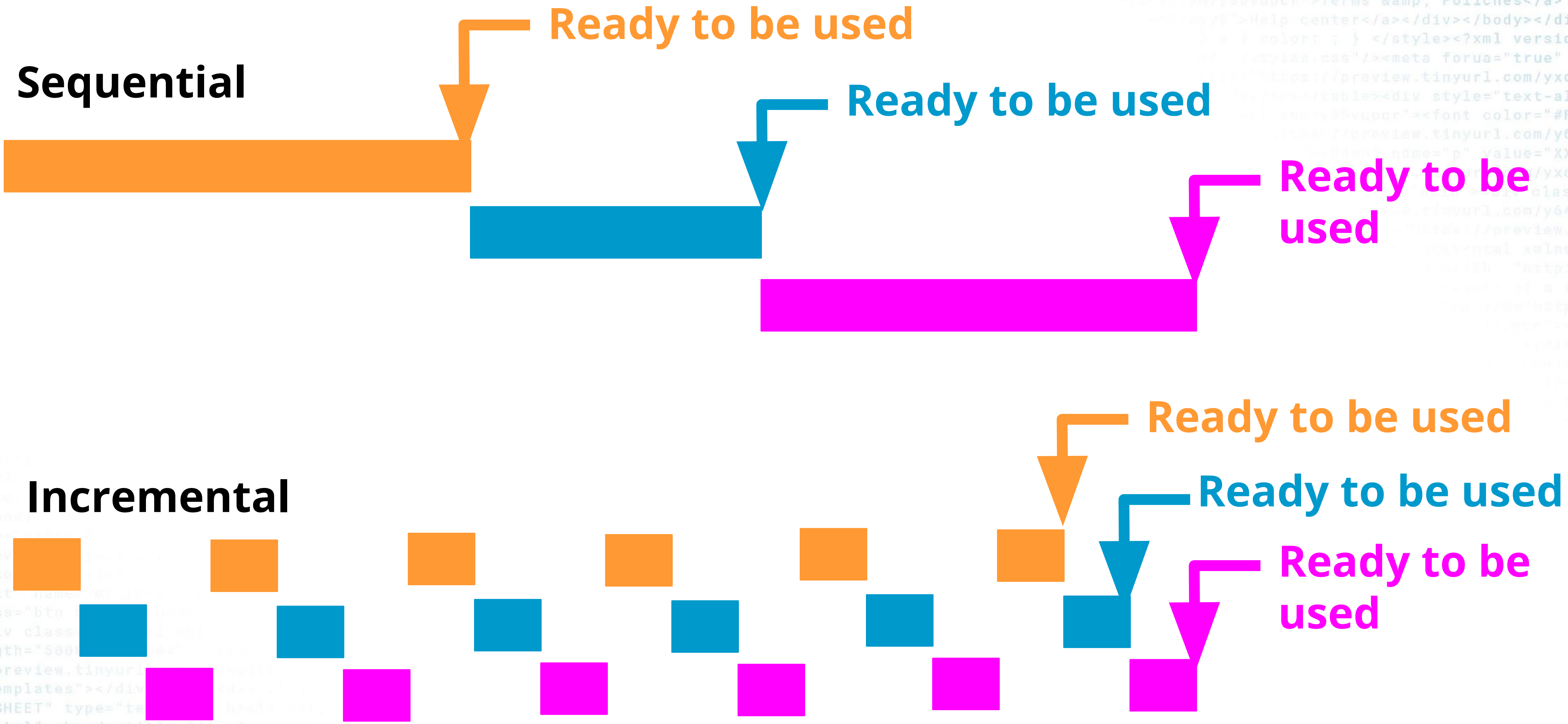
Top-down
incremental render



Increasing quality
incremental render

https://www.liquidweb.com/kb/what-is-a-progressive-jpeg

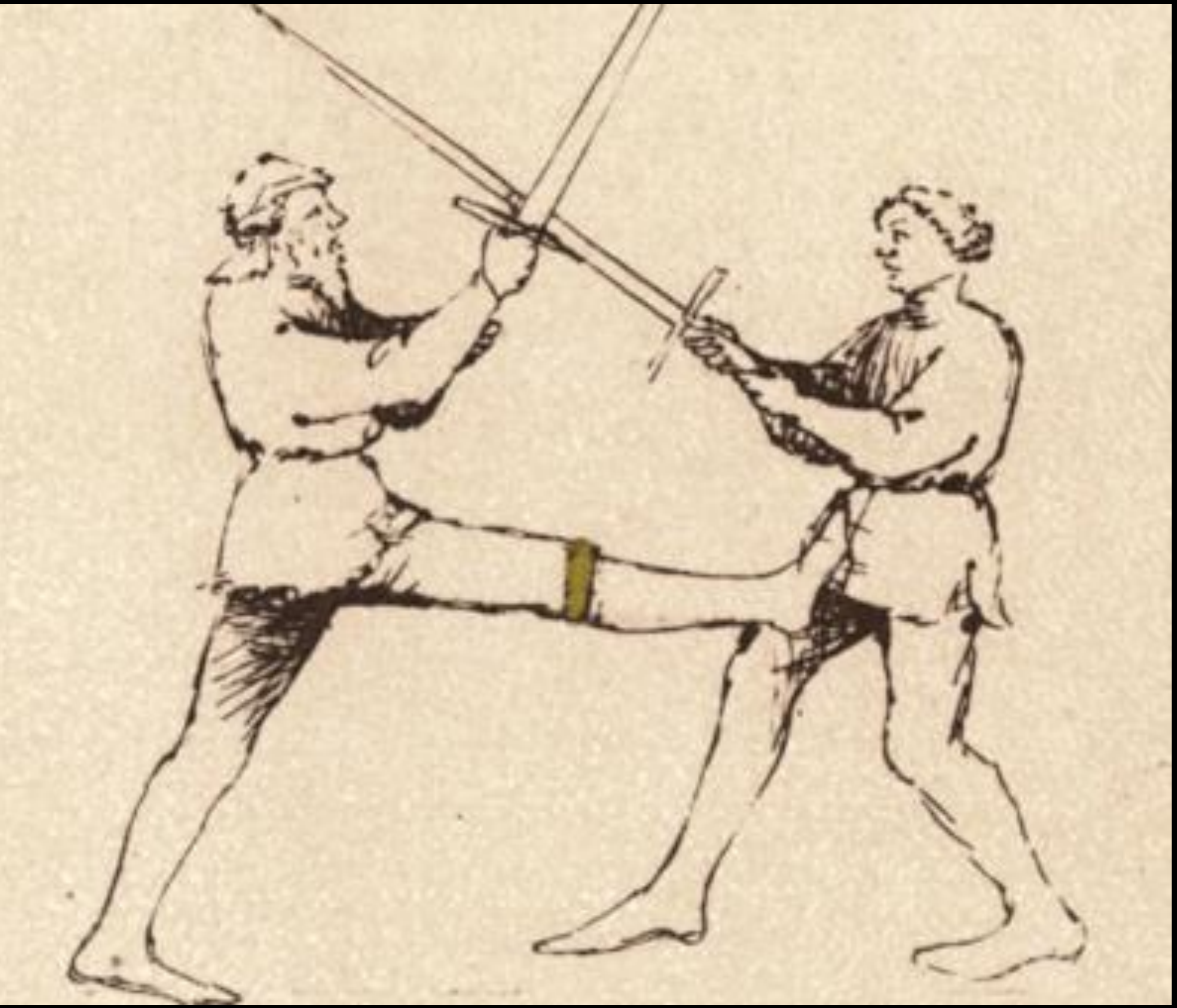
JS, CSS and Fonts need to be 100% loaded to be used!

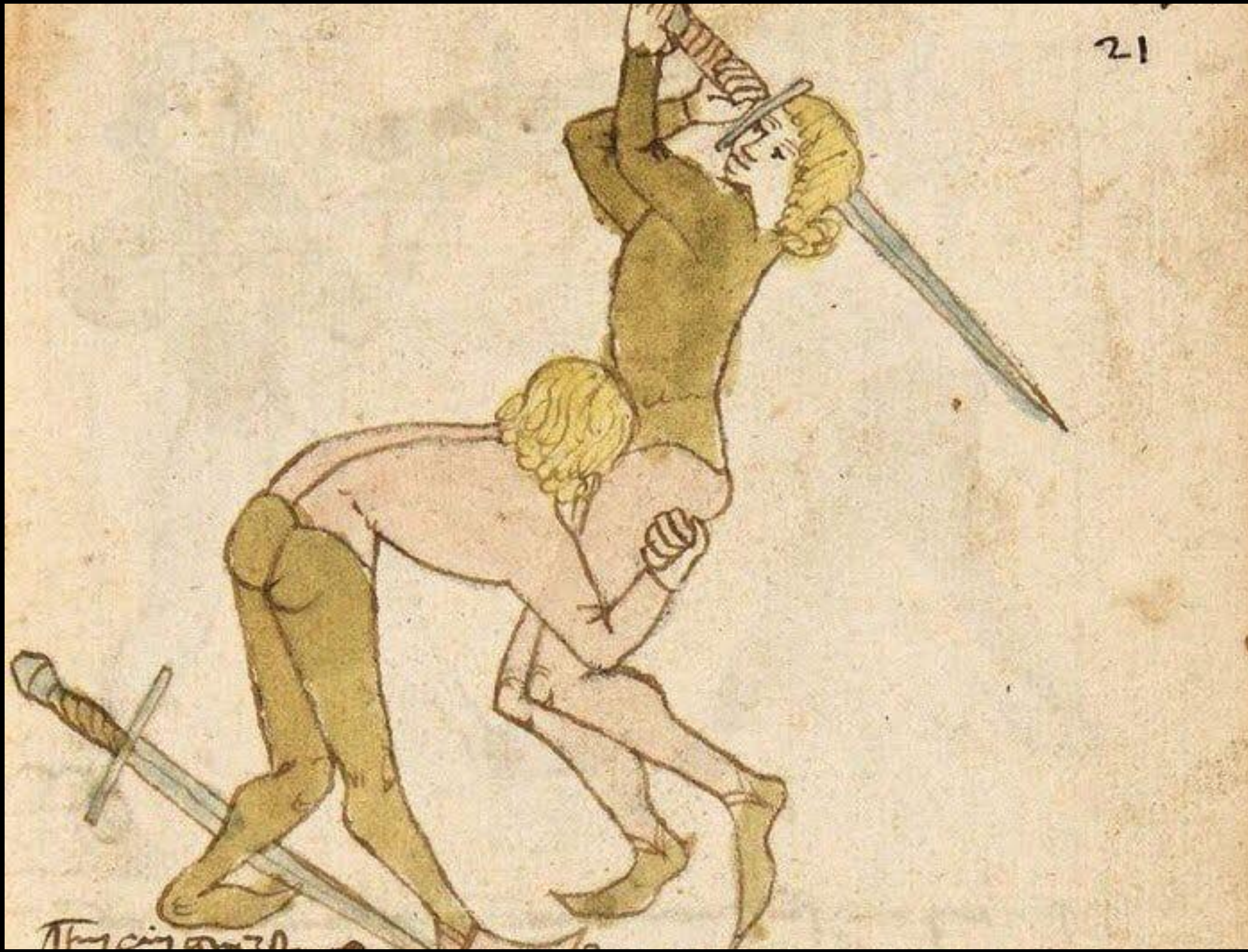


16th Century

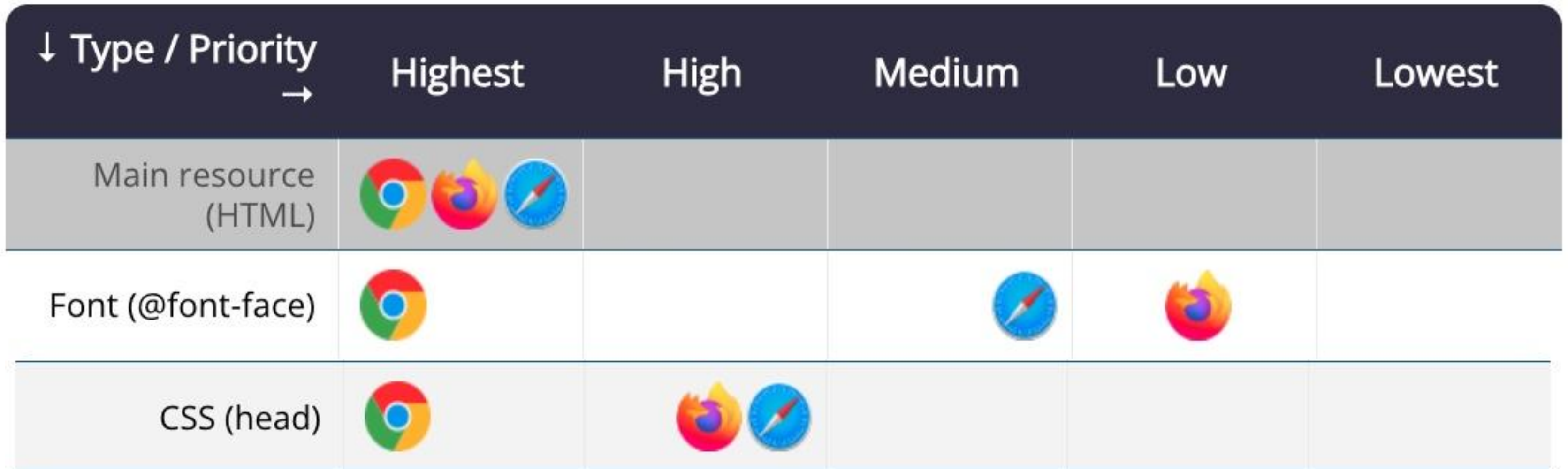
15th Century

14th Century




21

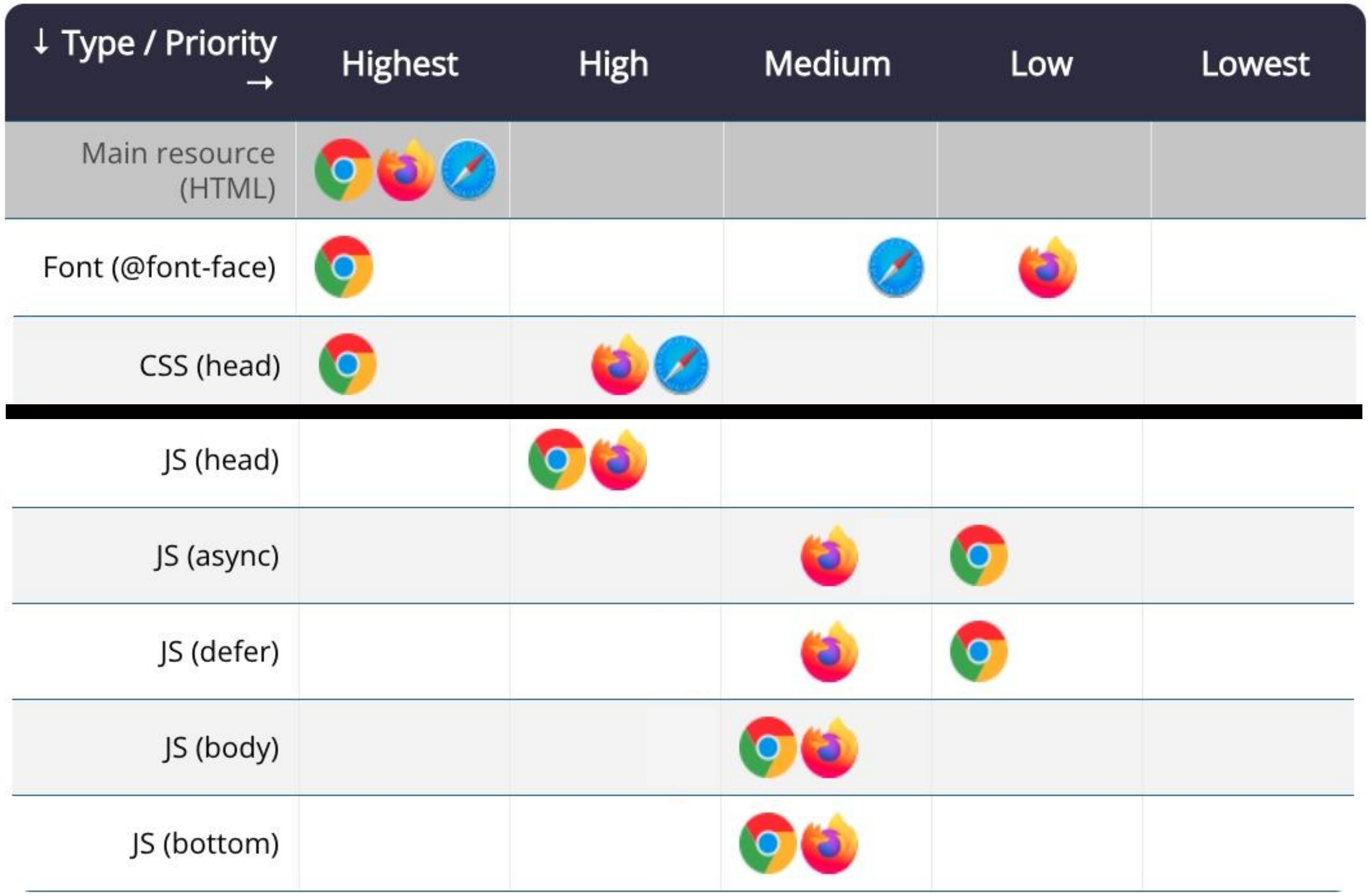
Browsers don’t agree on Urgency



↓ Type / Priority →	Highest	High	Medium	Low	Lowest
Main resource (HTML)	● ● ●				
Font (@font-face)	●		●	●	
CSS (head)	●	● ●			

https://calendar.perfplanet.com/2022/http-3-prioritization-demystified/

Browsers don’t agree on Urgency



↓ Type / Priority →	Highest	High	Medium	Low	Lowest
Main resource (HTML)	◉ 🦊 🧭				
Font (@font-face)	◉		🧭	🦊	
CSS (head)	◉	🦊 🧭			
JS (head)		◉ 🦊			
JS (async)			🦊	◉	
JS (defer)			🦊	◉	
JS (body)			◉ 🦊		
JS (bottom)			◉ 🦊		

https://calendar.perfplanet.com/2022/http-3-prioritization-demystified/

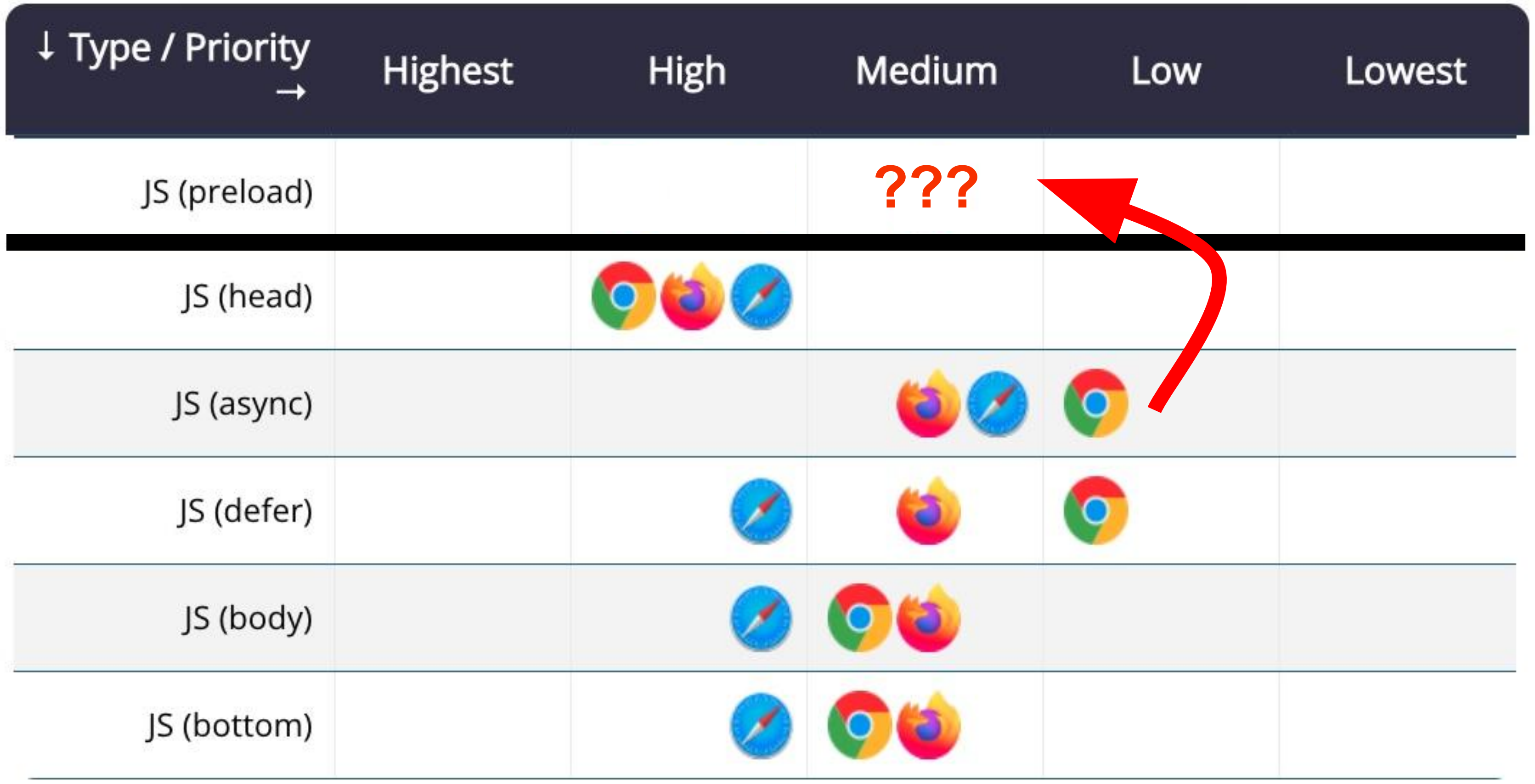
Browsers don’t agree on Urgency

↓ Type / Priority →	Highest	High	Medium	Low	Lowest
Main resource (HTML)	Chrome Firefox Safari				
Font (@font-face)	Chrome		Safari	Firefox	
CSS (head)	Chrome	Firefox Safari			
JS (head)		Chrome Firefox Safari			
JS (async)			Firefox Safari	Chrome	
JS (defer)		Safari	Firefox	Chrome	
JS (body)		Safari	Chrome Firefox		
JS (bottom)		Safari	Chrome Firefox		

https://calendar.perfplanet.com/2022/http-3-prioritization-demystified/

What about preload?

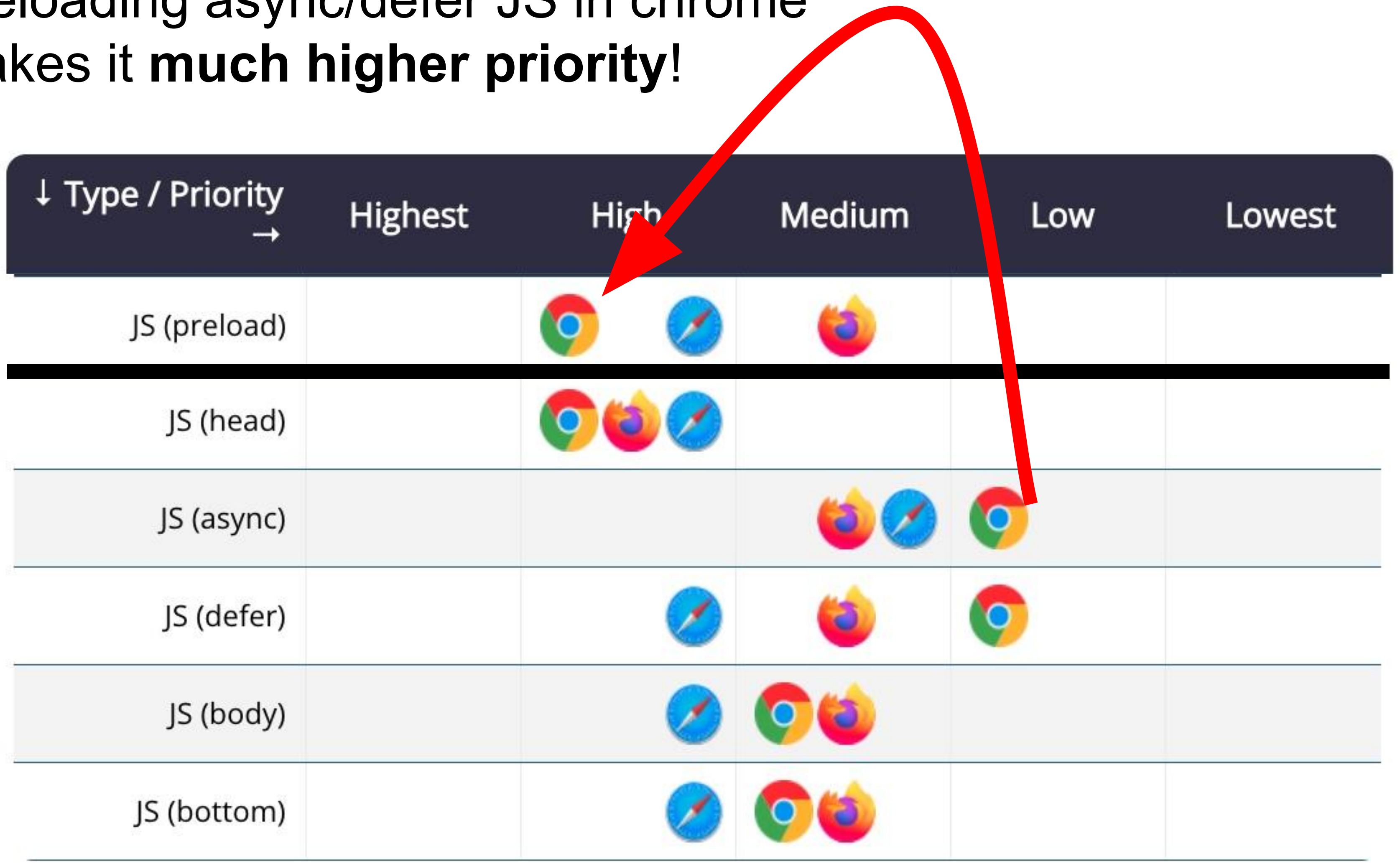
```html
<link rel="preload" href="submodule.js" as="script">
```


↓ Type / Priority →
Highest
High
Medium
Low
Lowest
JS (preload)
???
JS (head)
JS (async)
JS (defer)
JS (body)
JS (bottom)

https://calendar.perfplanet.com/2022/http-3-prioritization-demystified/

Browser doesn't have all the context yet…

Preloading async/defer JS in chrome makes it **much higher priority**!



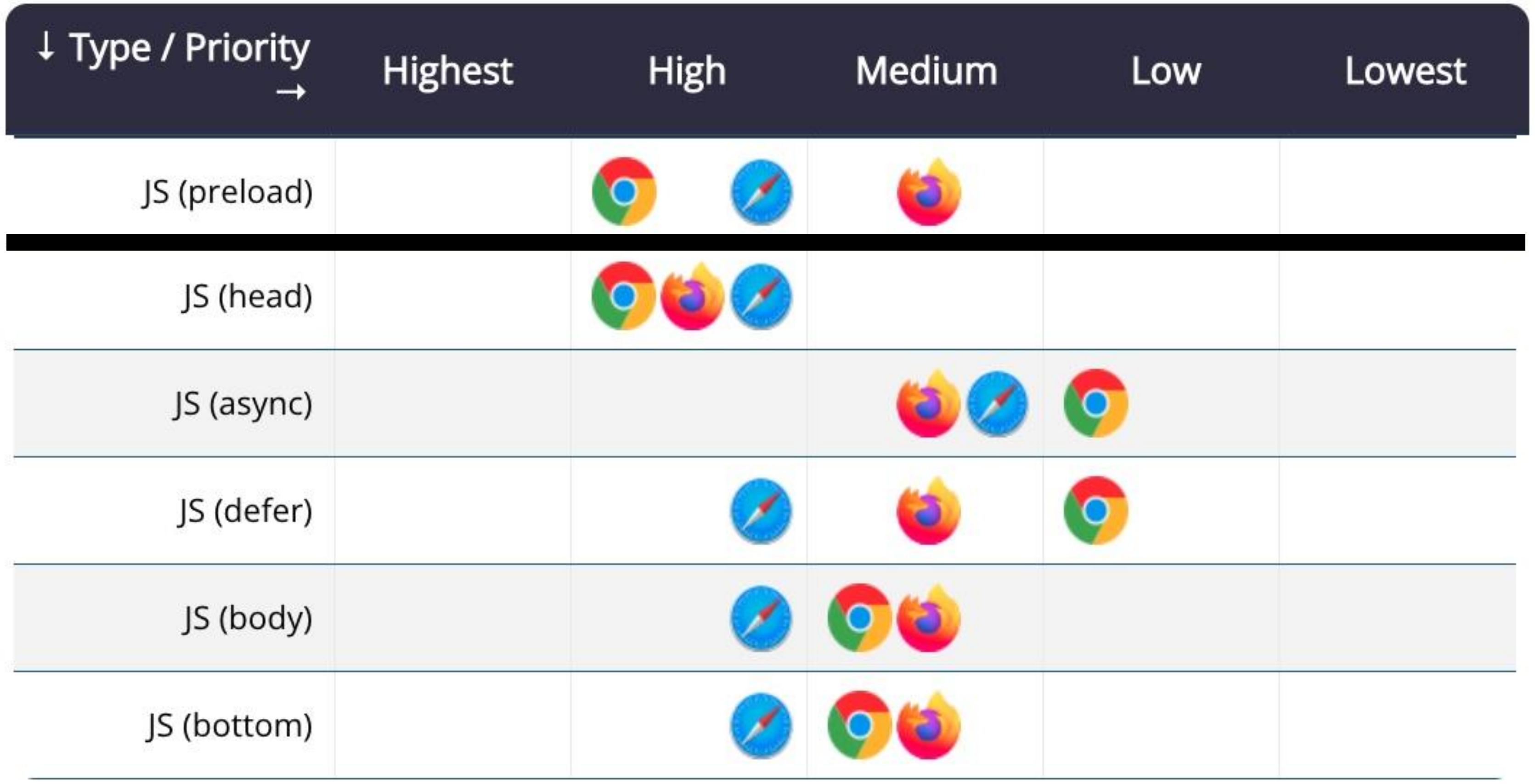
↓ Type / Priority →	Highest	High	Medium	Low	Lowest
JS (preload)		Chrome, Safari	Firefox		
JS (head)		Chrome, Firefox, Safari			
JS (async)			Firefox, Safari	Chrome	
JS (defer)		Safari	Firefox	Chrome	
JS (body)		Safari	Chrome, Firefox		
JS (bottom)		Safari	Chrome, Firefox		

https://calendar.perfplanet.com/2022/http-3-prioritization-demystified/

Browser contextless choices are inconsistent

Preloading images: **MEDIUM** → **LOW**

Preloading fonts: **HIGHEST** → **HIGH**

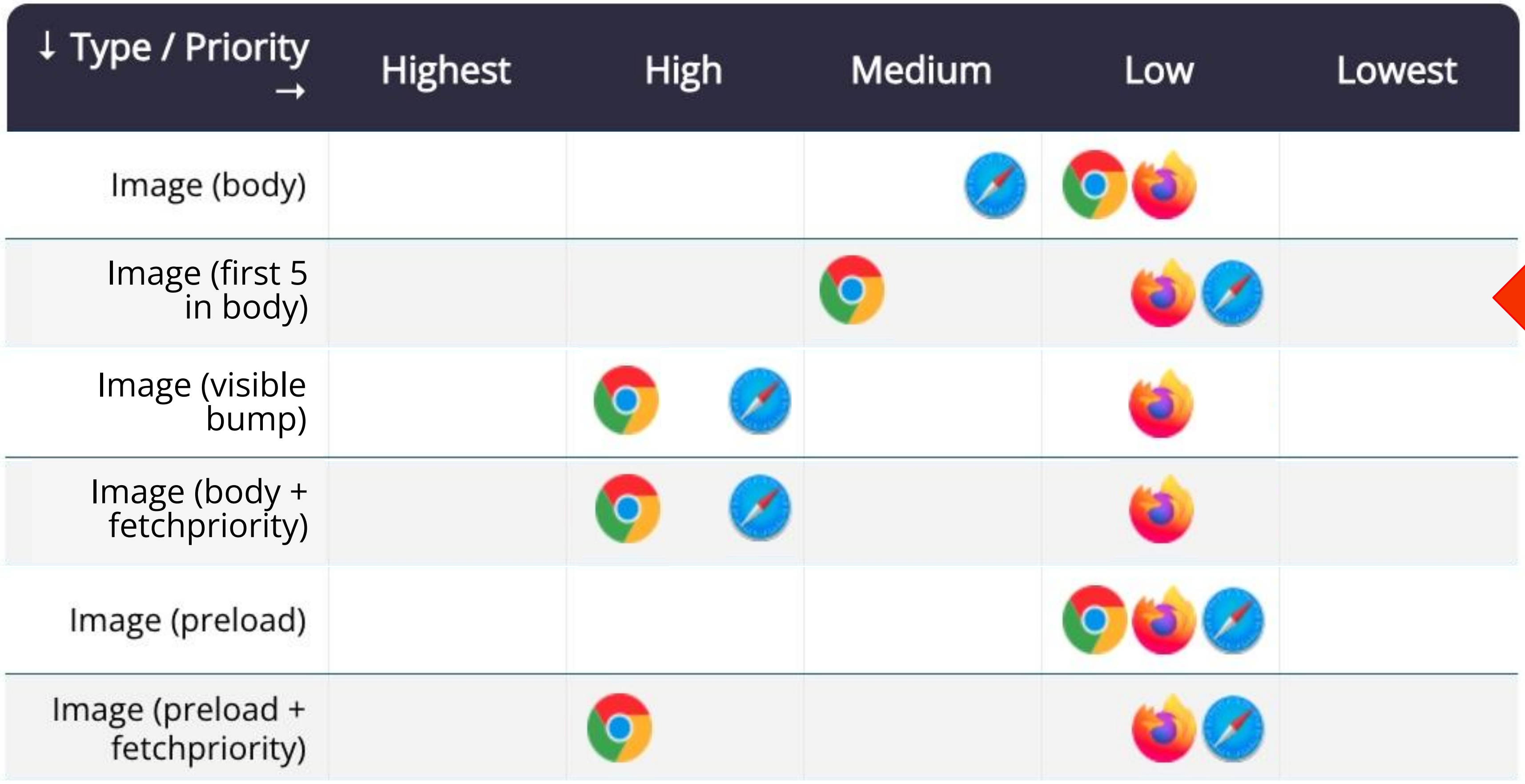


↓ Type / Priority →	Highest	High	Medium	Low	Lowest
JS (preload)		● ●	●		
JS (head)		● ● ●			
JS (async)			● ●	●	
JS (defer)		●	●	●	
JS (body)		●	● ●		
JS (bottom)		●	● ●		

https://calendar.perfplanet.com/2022/http-3-prioritization-demystified/

Akamai Experience the Edge

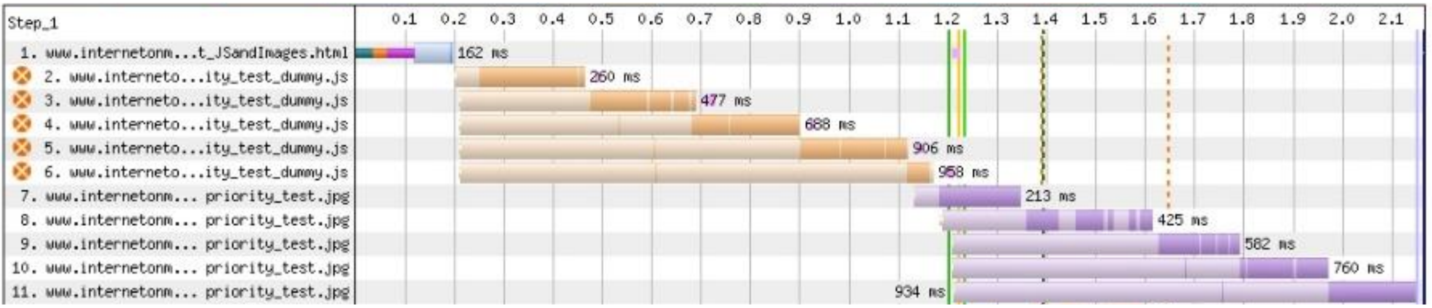
Let’s not talk about images…



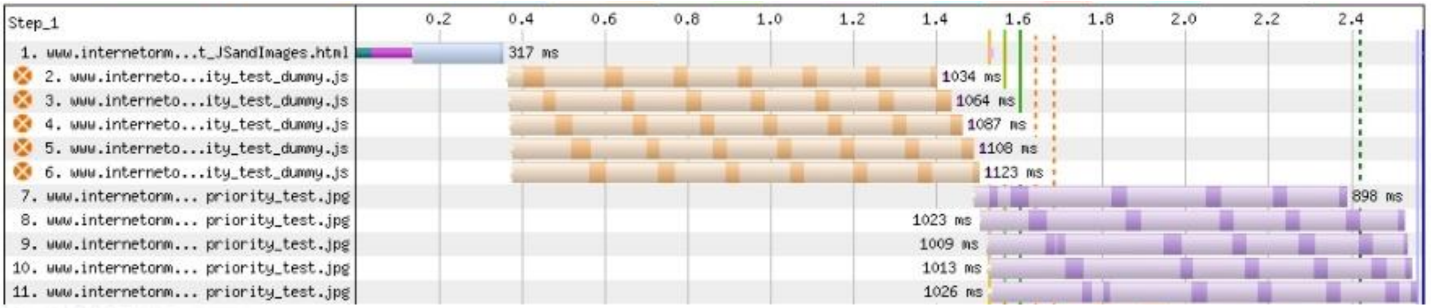
↓ Type / Priority →	Highest	High	Medium	Low	Lowest
Image (body)			Safari	Chrome Firefox	
Image (first 5 in body)			Chrome	Firefox Safari	
Image (visible bump)		Chrome Safari		Firefox	
Image (body + fetchpriority)		Chrome Safari		Firefox	
Image (preload)				Chrome Firefox Safari	
Image (preload + fetchpriority)		Chrome		Firefox Safari	

https://calendar.perfplanet.com/2022/http-3-prioritization-demystified/
https://bugs.chromium.org/p/chromium/issues/detail?id=1431169

Sequential vs Incremental

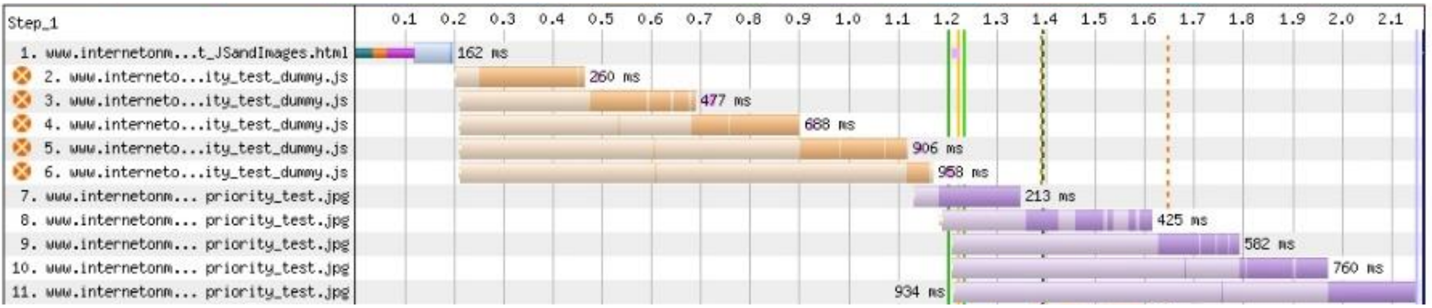


i = 0



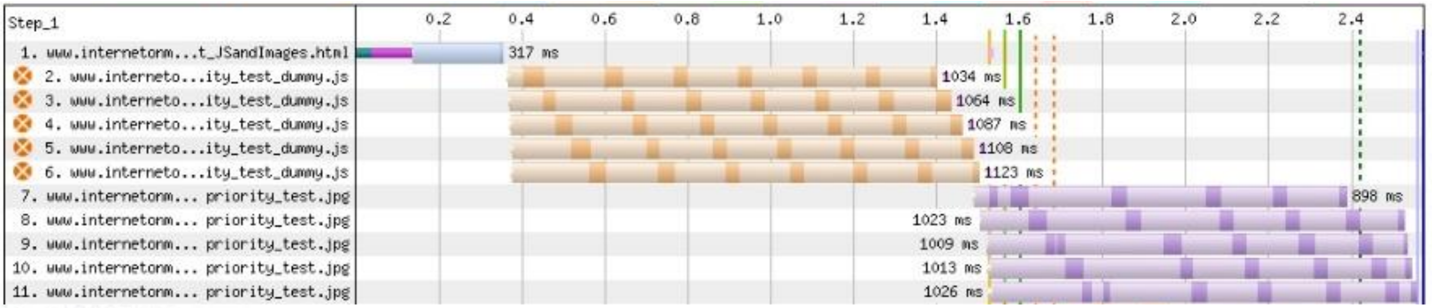
i = 1

Sequential vs Incremental vs Combined

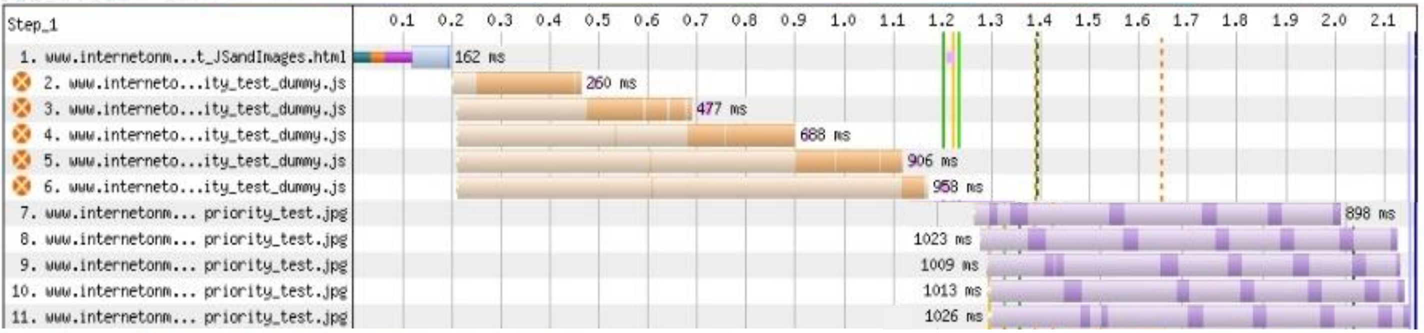


HTTP/3

i = 0



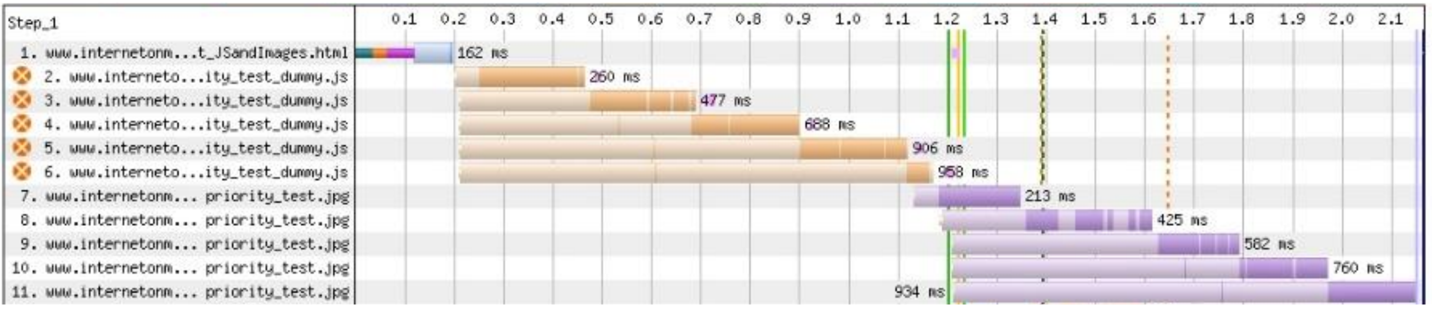
i = 1



HTTP/2

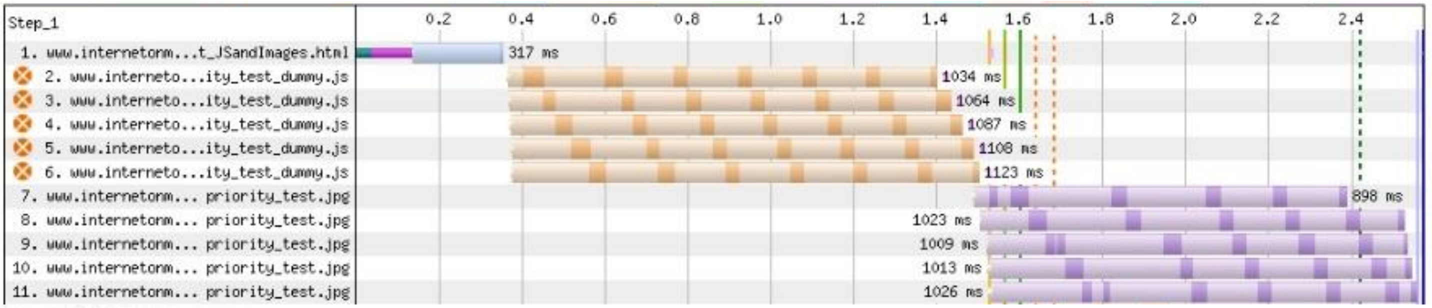
i = 0 OR
i = 1

Sequential vs Incremental vs Combined

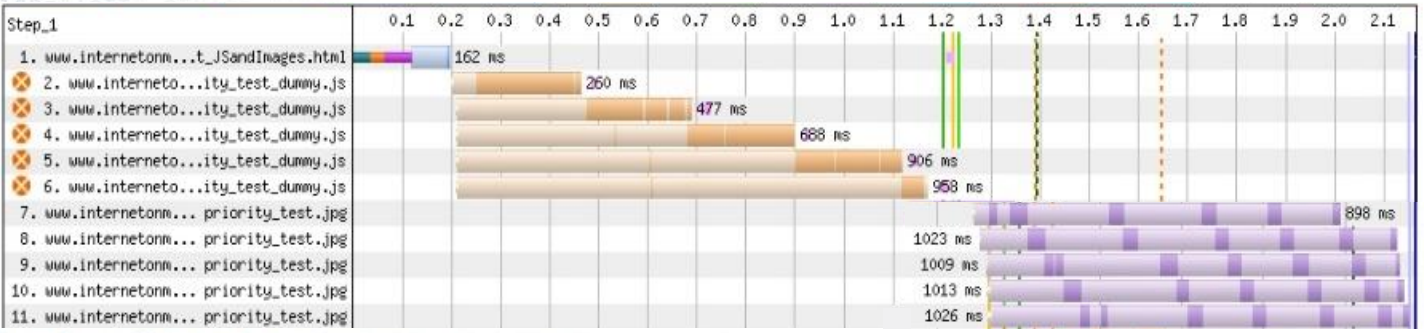


HTTP/3 HTTP/2

i = 0



i = 1

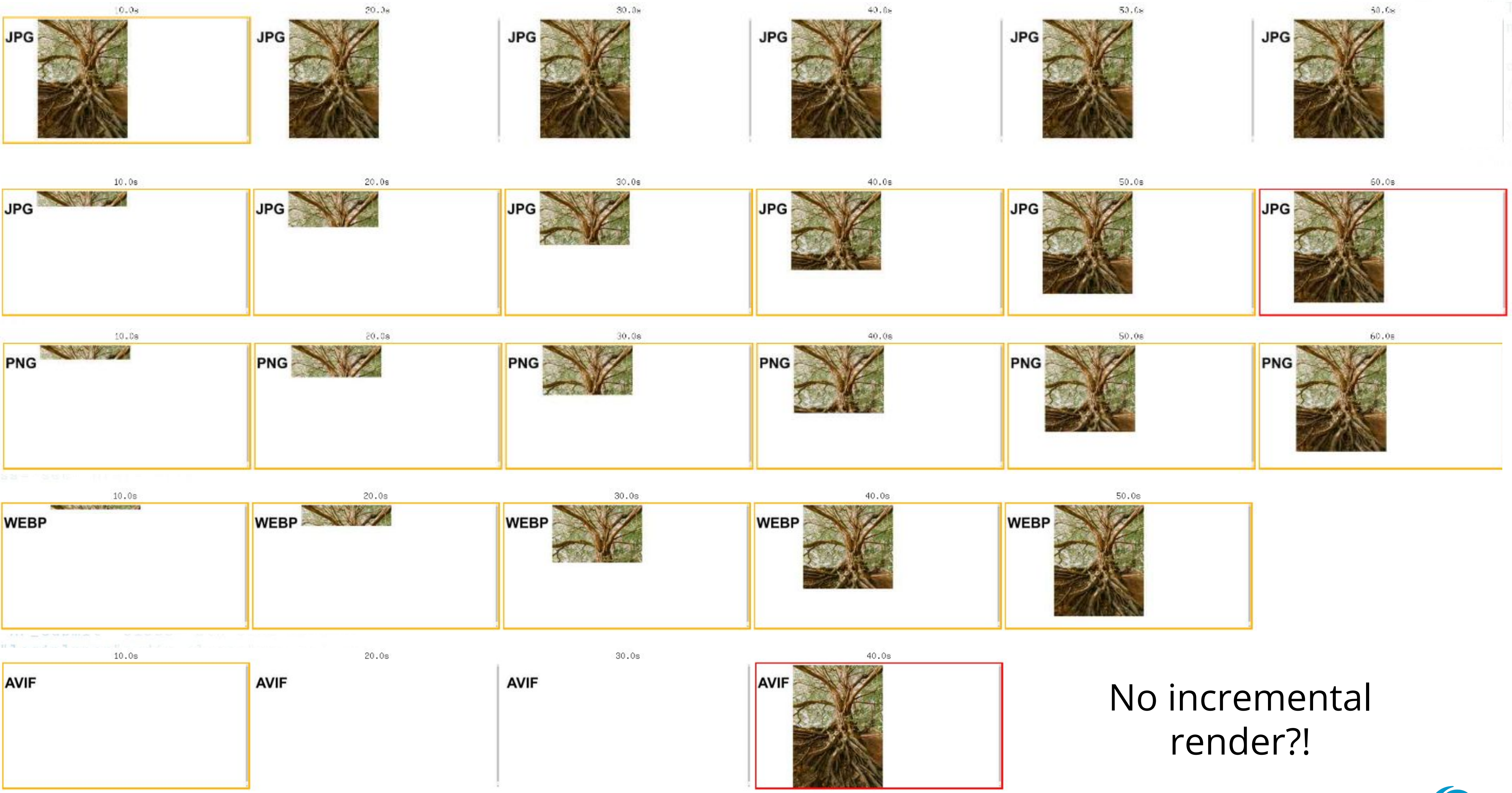


HTTP/2 HTTP/3

i = 0 OR
i = 1

https://bugs.chromium.org/p/chromium/issues/detail?id=1404785

10s 20s 30s 40s 50s 60s



Progressive JPEG

JPEG

PNG

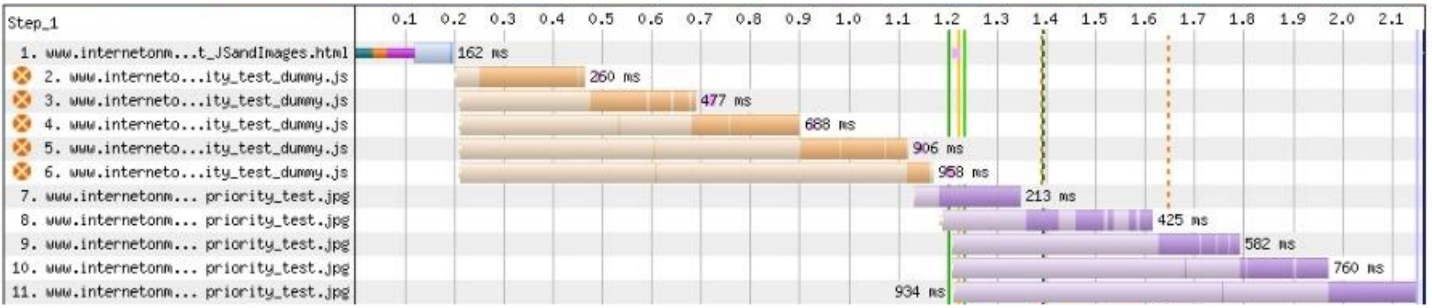
WEBP

No incremental render?!

AVIF

Experience the Edge

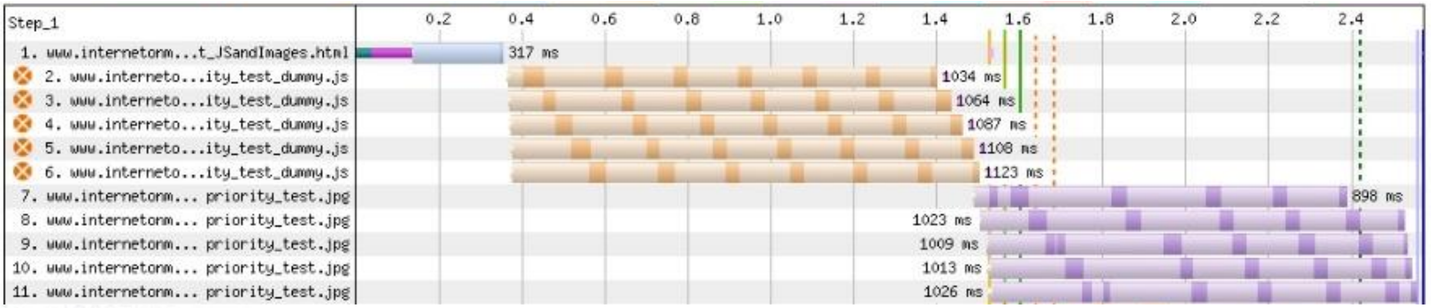
Browsers are inconsistent with THEMSELVES?!?



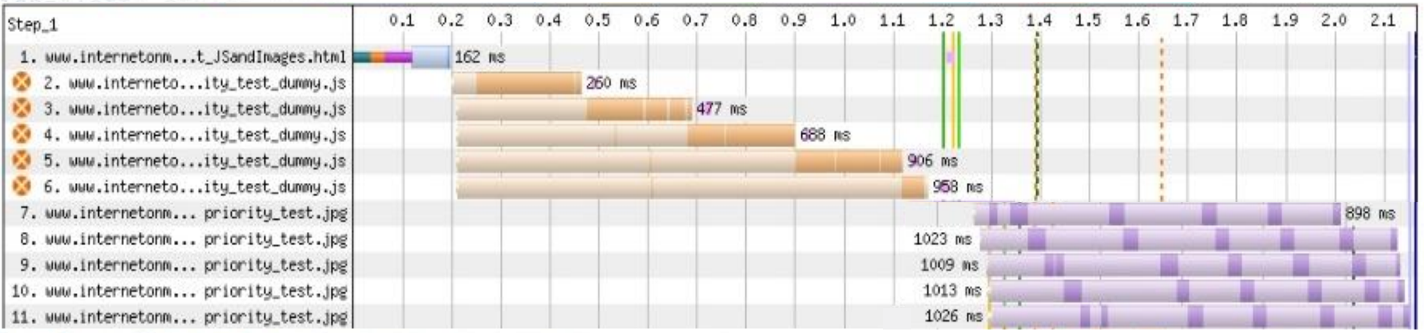
HTTP/3

HTTP/2

i = 0



i = 1



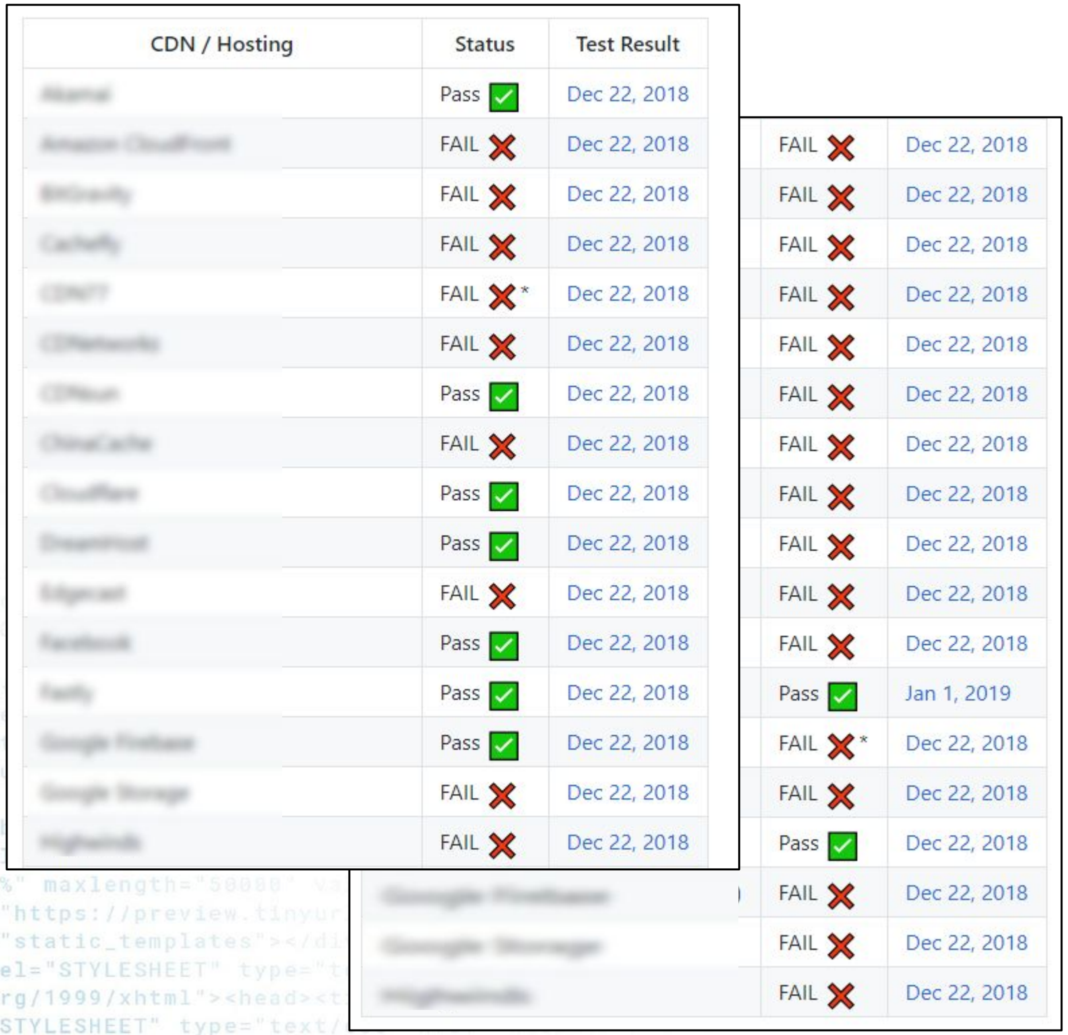
HTTP/2

HTTP/3

i = 0 OR
i = 1

https://bugs.chromium.org/p/chromium/issues/detail?id=1404785

HTTP/2 prioritization was (is?) broken



CDN / Hosting	Status	Test Result
[redacted]	Pass ✓	Dec 22, 2018
[redacted]	FAIL ✗	Dec 22, 2018
[redacted]	FAIL ✗	Dec 22, 2018
[redacted]	FAIL ✗	Dec 22, 2018
[redacted]	FAIL ✗ *	Dec 22, 2018
[redacted]	FAIL ✗	Dec 22, 2018
[redacted]	Pass ✓	Dec 22, 2018
[redacted]	FAIL ✗	Dec 22, 2018
[redacted]	Pass ✓	Dec 22, 2018
[redacted]	Pass ✓	Dec 22, 2018
[redacted]	FAIL ✗	Dec 22, 2018
[redacted]	Pass ✓	Dec 22, 2018
[redacted]	Pass ✓	Dec 22, 2018
[redacted]	Pass ✓	Dec 22, 2018
[redacted]	FAIL ✗	Dec 22, 2018
[redacted]	FAIL ✗	Dec 22, 2018

	FAIL ✗	Dec 22, 2018
	FAIL ✗	Dec 22, 2018
	FAIL ✗	Dec 22, 2018
	FAIL ✗	Dec 22, 2018
	FAIL ✗	Dec 22, 2018
	FAIL ✗	Dec 22, 2018
	FAIL ✗	Dec 22, 2018
	FAIL ✗	Dec 22, 2018
	FAIL ✗	Dec 22, 2018
	FAIL ✗	Dec 22, 2018
	FAIL ✗	Dec 22, 2018
	Pass ✓	Jan 1, 2019
	FAIL ✗ *	Dec 22, 2018
	FAIL ✗	Dec 22, 2018
	Pass ✓	Dec 22, 2018
[redacted]	FAIL ✗	Dec 22, 2018
[redacted]	FAIL ✗	Dec 22, 2018
[redacted]	FAIL ✗	Dec 22, 2018

2018 - 2019:

only **9 / 34** deployments pass

https://github.com/andydavies/http2-prioritization-issues

Looked at these companies

Akamai

shopify

CLOUDFLARE®

fastly®

HTTP/3

HTTP/2

JSDELIVR

amazon cloudfront

Google Cloud

Caddy®

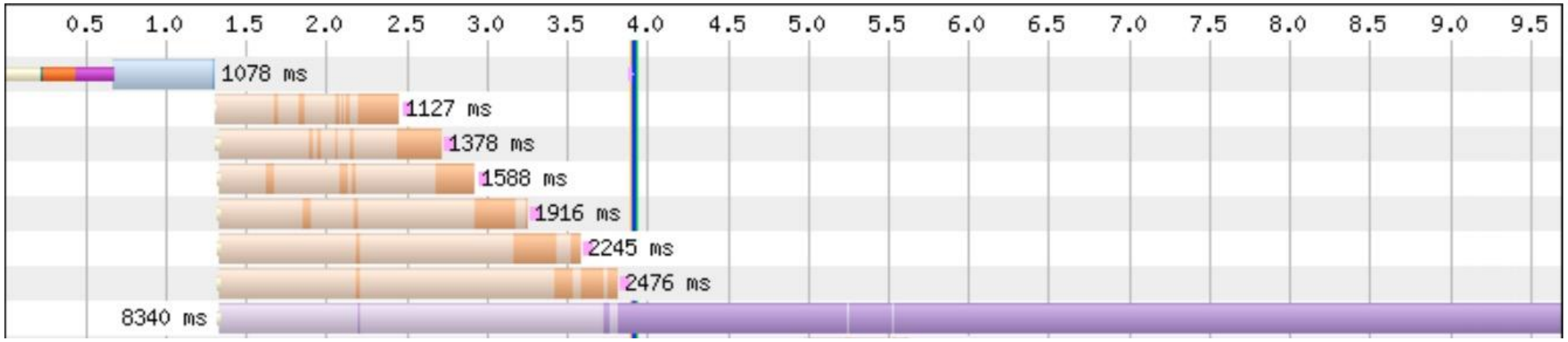
node JS

Vercel

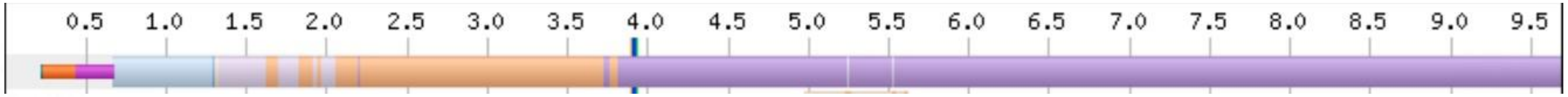
QUIC.cloud

netlify

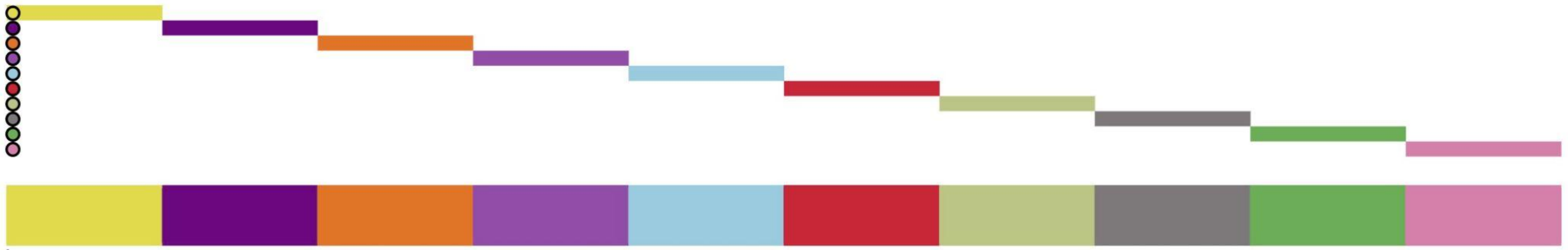
How to interpret the next few slides:



waterfall



connection view

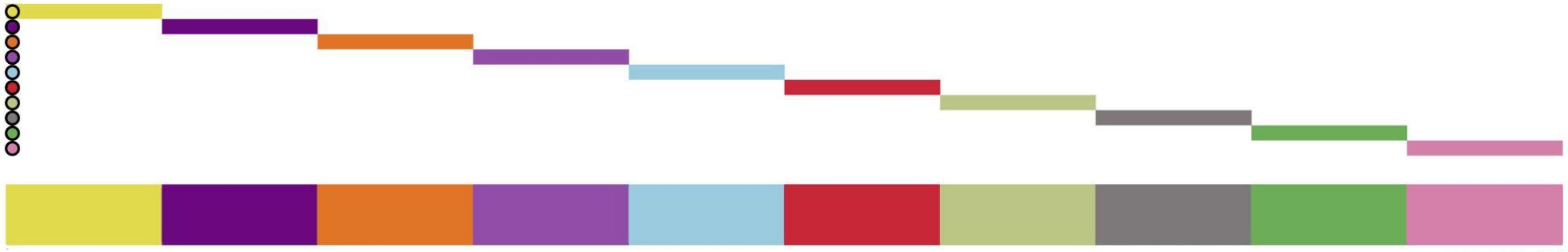


My **inferior** imitations

HTTP/3

Same Urgency, Sequential

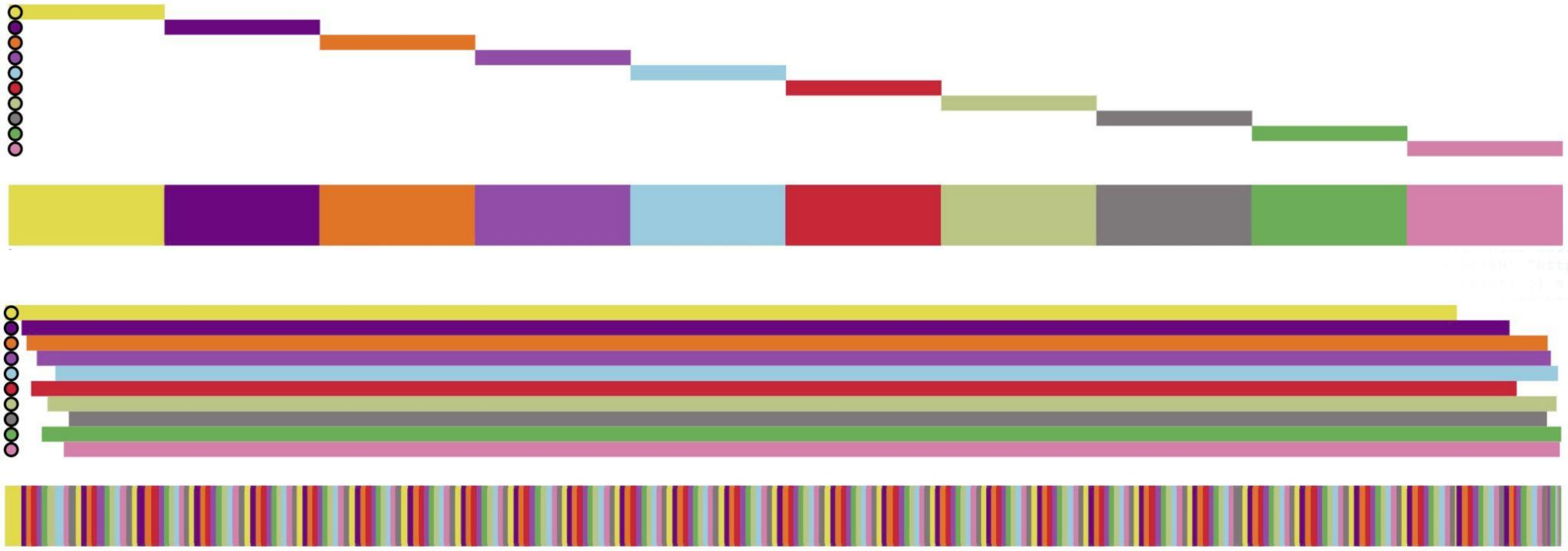
All: u=3, i=0



Akamai Experience the Edge

Same Urgency, Sequential

All: u=3, i=0

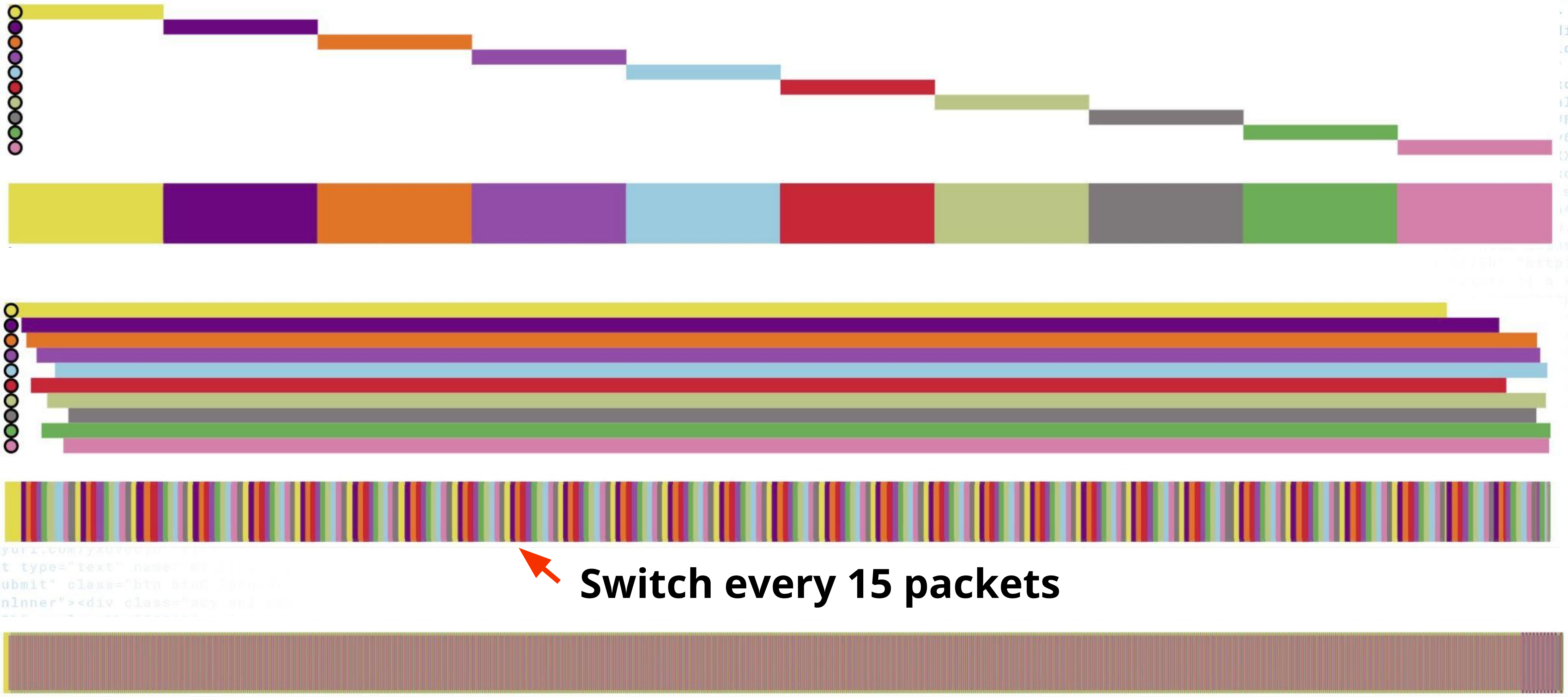


Pretends i=1 for all...

HTTP/3

Same Urgency, Sequential

All: u=3, i=0



Switch every 15 packets

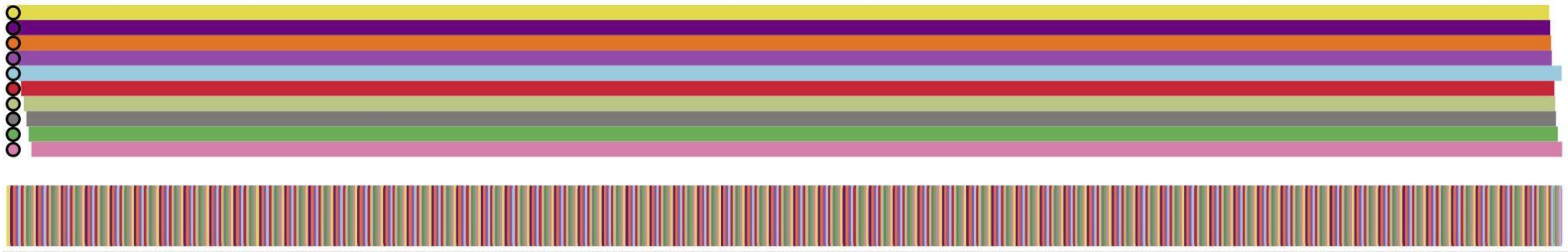
Switch every packet

Akamai Experience the Edge

HTTP/3

Same Urgency, Incremental

All: u=3, i=1

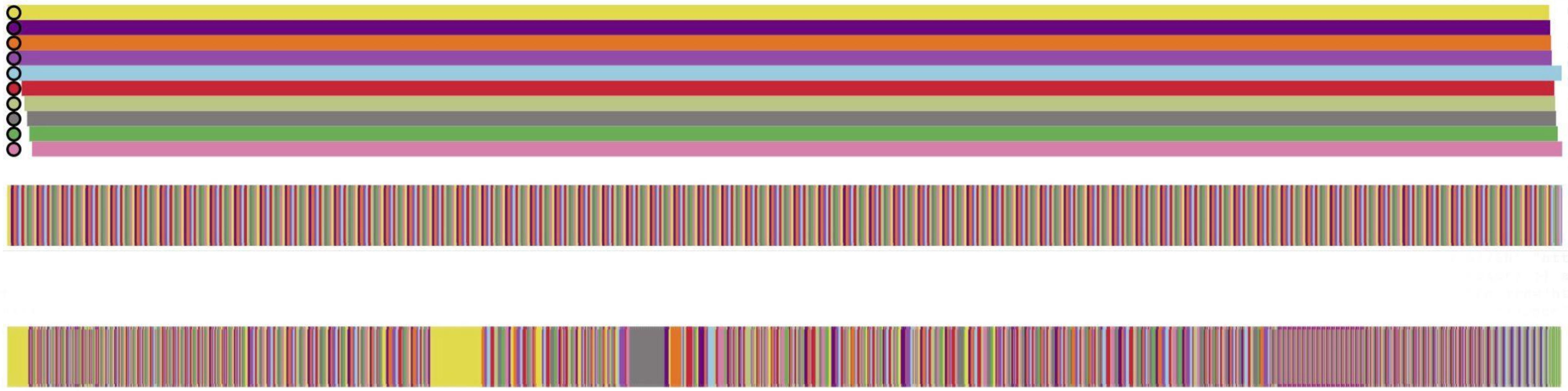


Akamai Experience the Edge

HTTP/3

Same Urgency, Incremental

All: u=3, i=1

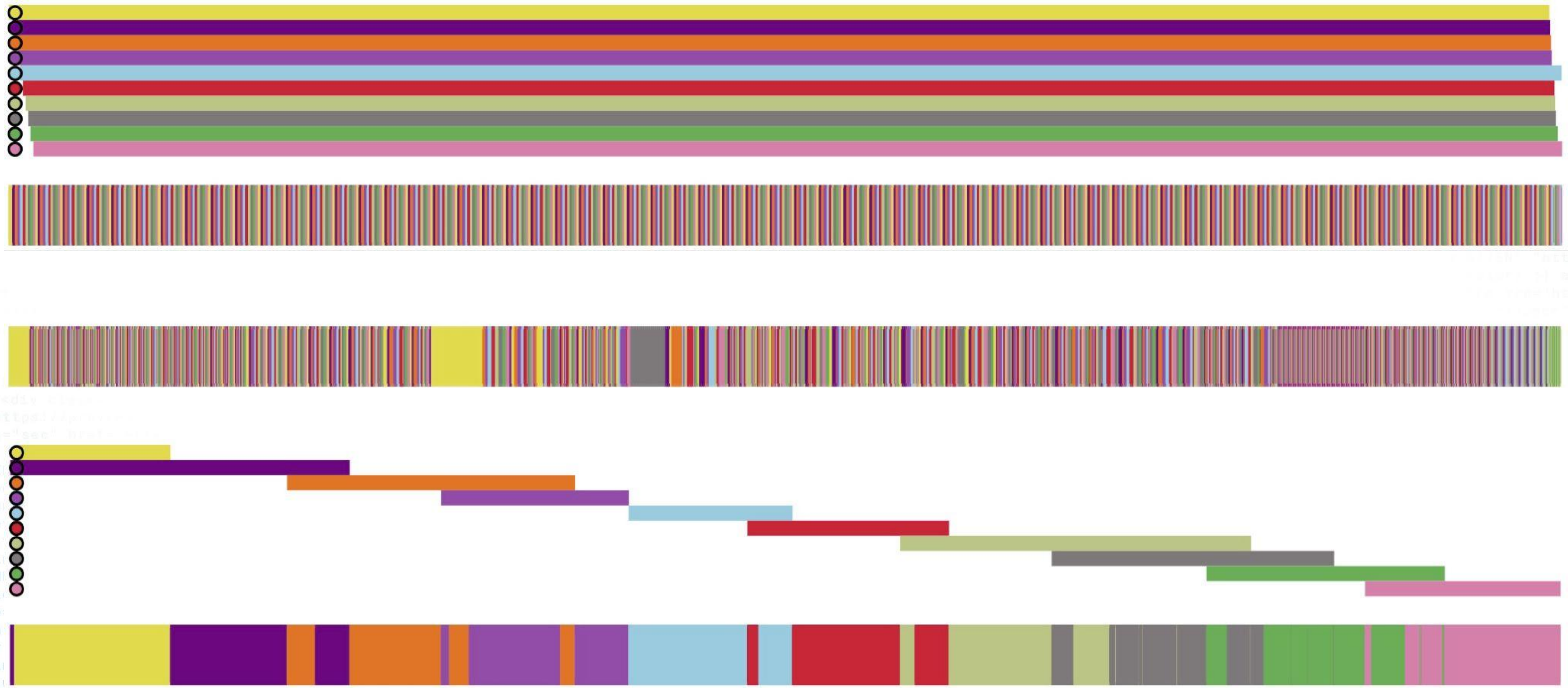


Akamai Experience the Edge

HTTP/3

Same Urgency, Incremental

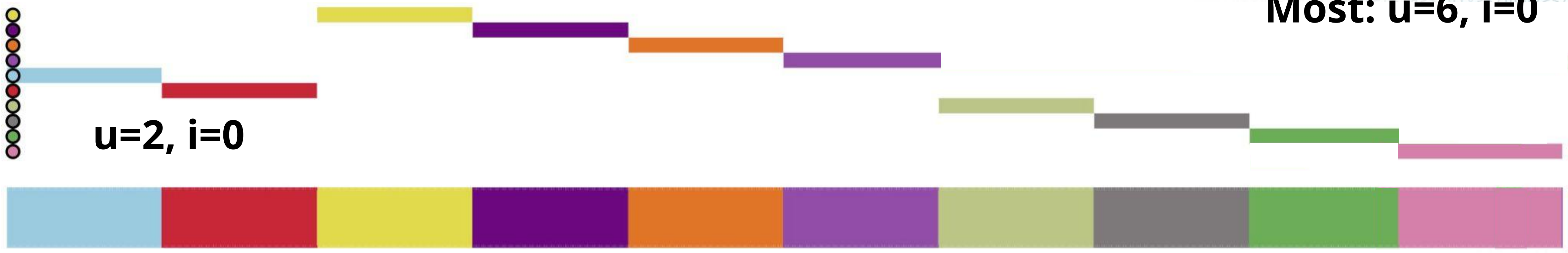
All: u=3, i=1



Akamai Experience the Edge

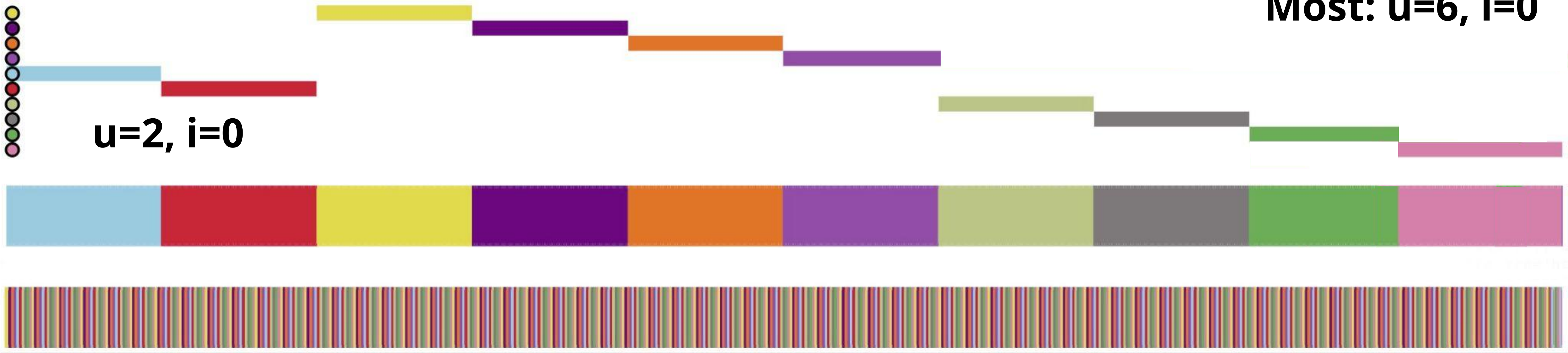
HTTP/3

5 and 6 with higher priority, sequential



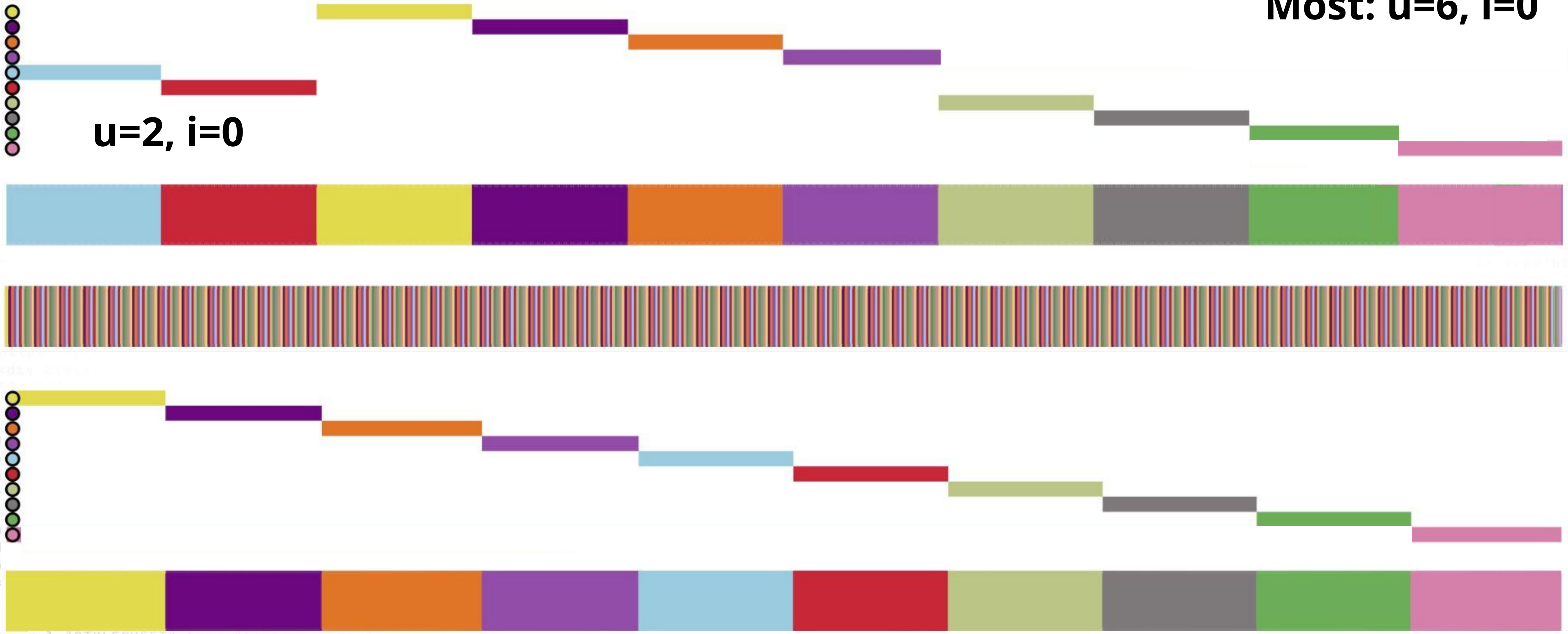
Akamai Experience the Edge

5 and 6 with higher priority, sequential



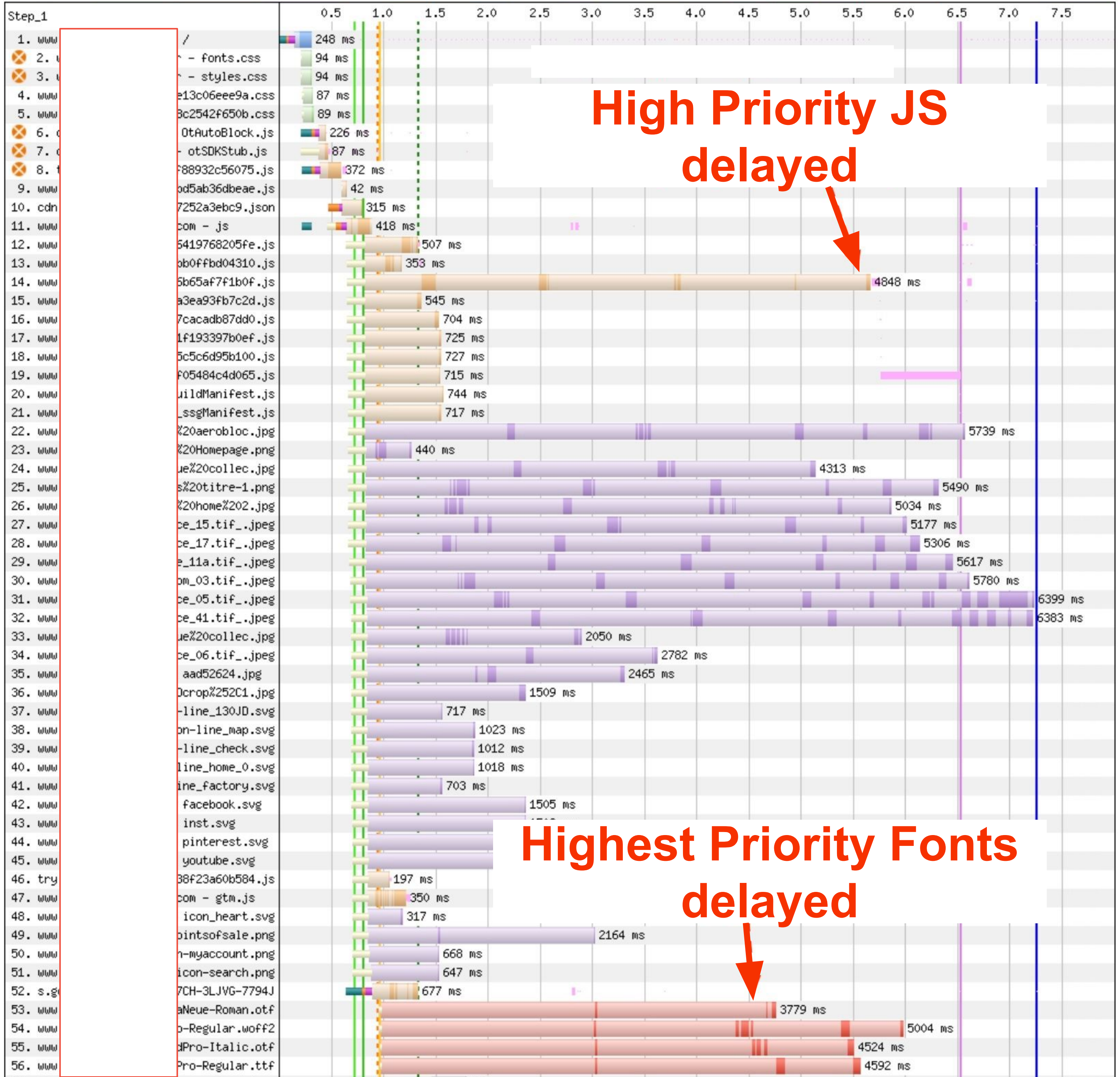
HTTP/3

5 and 6 with higher priority, sequential



Ignore urgency, only look at request order

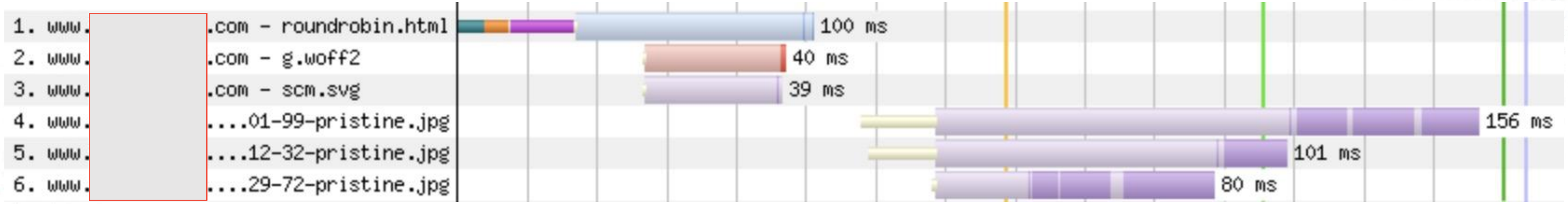
Akamai Experience the Edge



HTTP/2

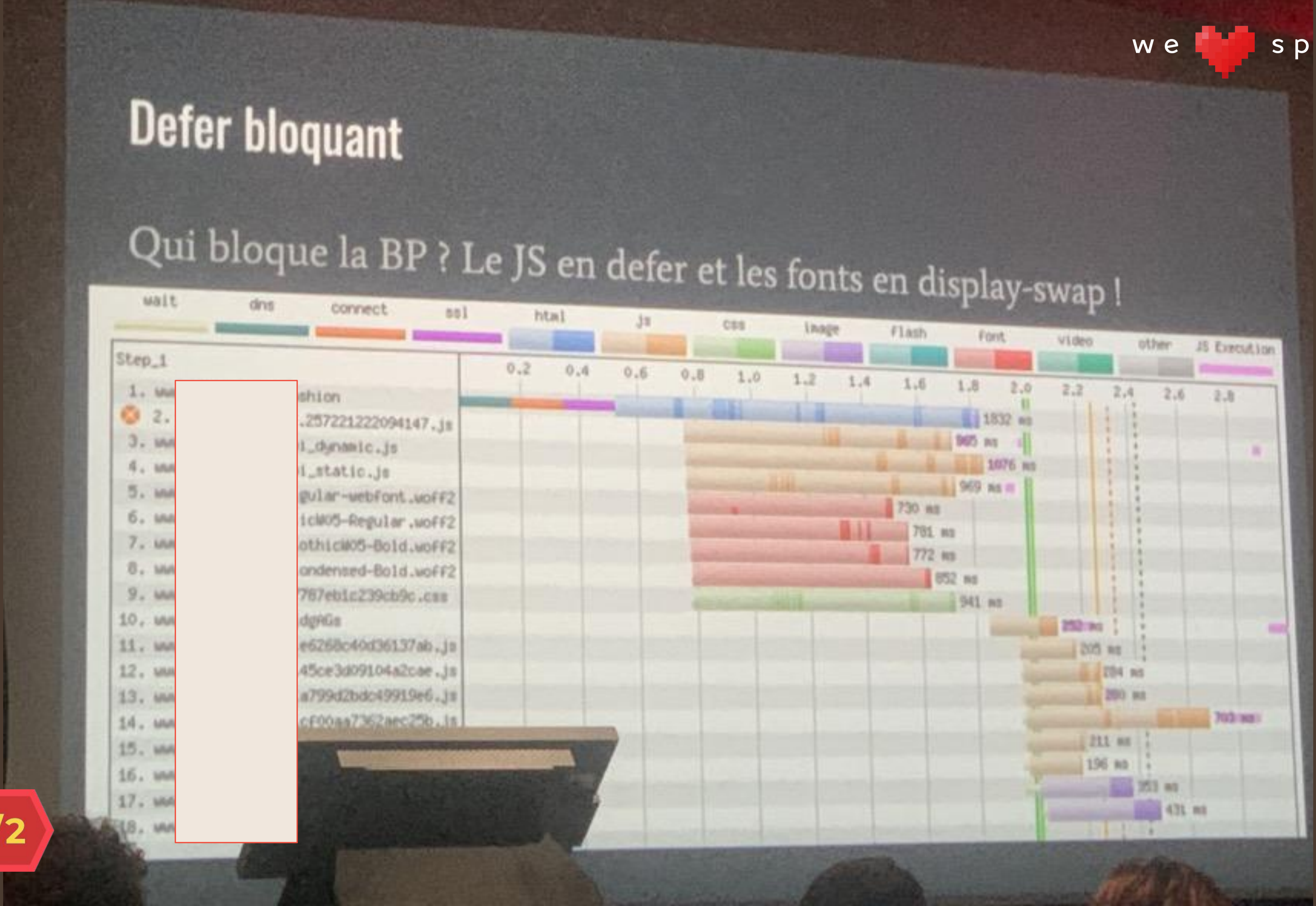
Akamai Experience the Edge

Order inversion!



1. www.[redacted].com - roundrobin.html 100 ms
2. www.[redacted].com - g.woff2 40 ms
3. www.[redacted].com - scm.svg 39 ms
4. www.[redacted]....01-99-pristine.jpg 156 ms
5. www.[redacted]....12-32-pristine.jpg 101 ms
6. www.[redacted]....29-72-pristine.jpg 80 ms

we speed



Defer bloquant

Qui bloque la BP ? Le JS en defer et les fonts en display-swap !

```
Step_1
 1. ww[redacted]shion                      1832 ms
 2. [redacted].257221222094147.js           905 ms
 3. ww[redacted]i_dynamic.js               1076 ms
 4. ww[redacted]i_static.js                 969 ms
 5. ww[redacted]gular-webfont.woff2         730 ms
 6. ww[redacted]icW05-Regular.woff2         781 ms
 7. ww[redacted]othicW05-Bold.woff2         772 ms
 8. ww[redacted]ondensed-Bold.woff2         852 ms
 9. ww[redacted]787eb1c239cb9c.css          941 ms
10. ww[redacted]dgAGs                       252 ms
11. ww[redacted]e6268c40d36137ab.js         205 ms
12. ww[redacted]45ce3d09104a2cae.js         284 ms
13. ww[redacted]a799d2bdc49919e6.js         280 ms
14. ww[redacted]cf00aa7362aec25b.js         703 ms
15. ww[redacted]                            211 ms
16. ww[redacted]                            196 ms
17. ww[redacted]                            353 ms
18. ww[redacted]                            431 ms
```

HTTP/2

Only 2 of these companies do it (100%) correctly…

Akamai

shopify

CLOUDFLARE

fastly

HTTP/3

HTTP/2

JSDELIVR

amazon cloudfront

Google Cloud

Caddy

node JS

Vercel

QUIC.cloud

netlify

THIS IS FINE.

CDNs

fetchpriority

tight mode

Network Performance isn't the *most impactful* thing

“If you have an optimized site on a CDN, the network is *usually* fast enough.

If you're loading 5MB of JavaScript without a CDN, you have **bigger problems than the network!**”

Robin Marx, *PerfNow 2023*

FetchPriority

```html
<img src="lcp-image.jpg" fetchpriority="high">
```

```html
<link rel="preload" href="/defer.js" as="script" fetchpriority="low">
```

https://web.dev/fetch-priority

FetchPriority doesn't control Urgency *directly*

```
<img src="lcp-image.jpg" fetchpriority="high">

<img src="lcp-image.jpg" fetchpriority="low">
```

✅

```
<img src="lcp-image.jpg" fetchpriority="highest">

<img src="lcp-image.jpg" fetchpriority="lowest">
```

❌

https://web.dev/fetch-priority

FetchPriority doesn't control Urgency *directly*

```html
<img src="lcp-image.jpg" fetchpriority="high">
<img src="lcp-image.jpg" fetchpriority="low">
```

LOW → HIGH

LOW → LOW

```html
<link rel="stylesheet" href="main.css" fetchpriority="high" />
<link rel="stylesheet" href="main.css" fetchpriority="low" />
```

HIGHEST → HIGHEST

HIGHEST → HIGH

https://web.dev/fetch-priority

FetchPriority doesn't control Urgency *directly*

```
<img src="lcp-image.jpg" fetchpriority="high">
<img src="lcp-image.jpg" fetchpriority="low">
```

LOW → HIGH

LOW → LOW

```
<link rel="stylesheet" href="main.css" fetchpriority="high" />
<link rel="stylesheet" href="main.css" fetchpriority="low" />
```

HIGHEST → HIGHEST

HIGHEST → HIGH

```
<script src="late.js" fetchpriority="high"></script>
<script src="late.js" fetchpriority="low"></script>
```

MEDIUM → HIGH

MEDIUM → LOW

https://web.dev/fetch-priority

Akamai Experience the Edge

FetchPriority is Urgency only!

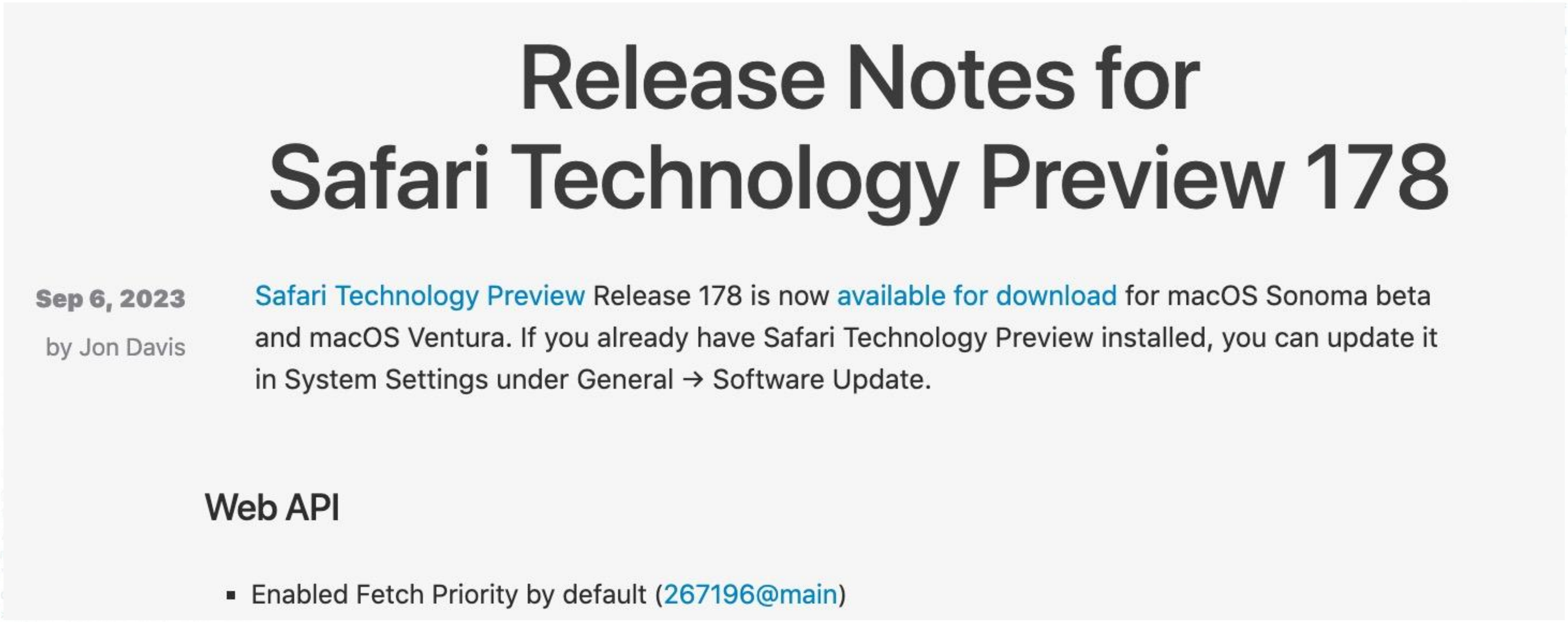
```html
<img src="lcp-image.jpg" fetchpriority="high, incremental">

<img src="lcp-image.jpg" fetchpriority="low, sequential">
```

❌

😭

FetchPriority: coming to a Browser near you!



Release Notes for Safari Technology Preview 178

Sep 6, 2023

by Jon Davis

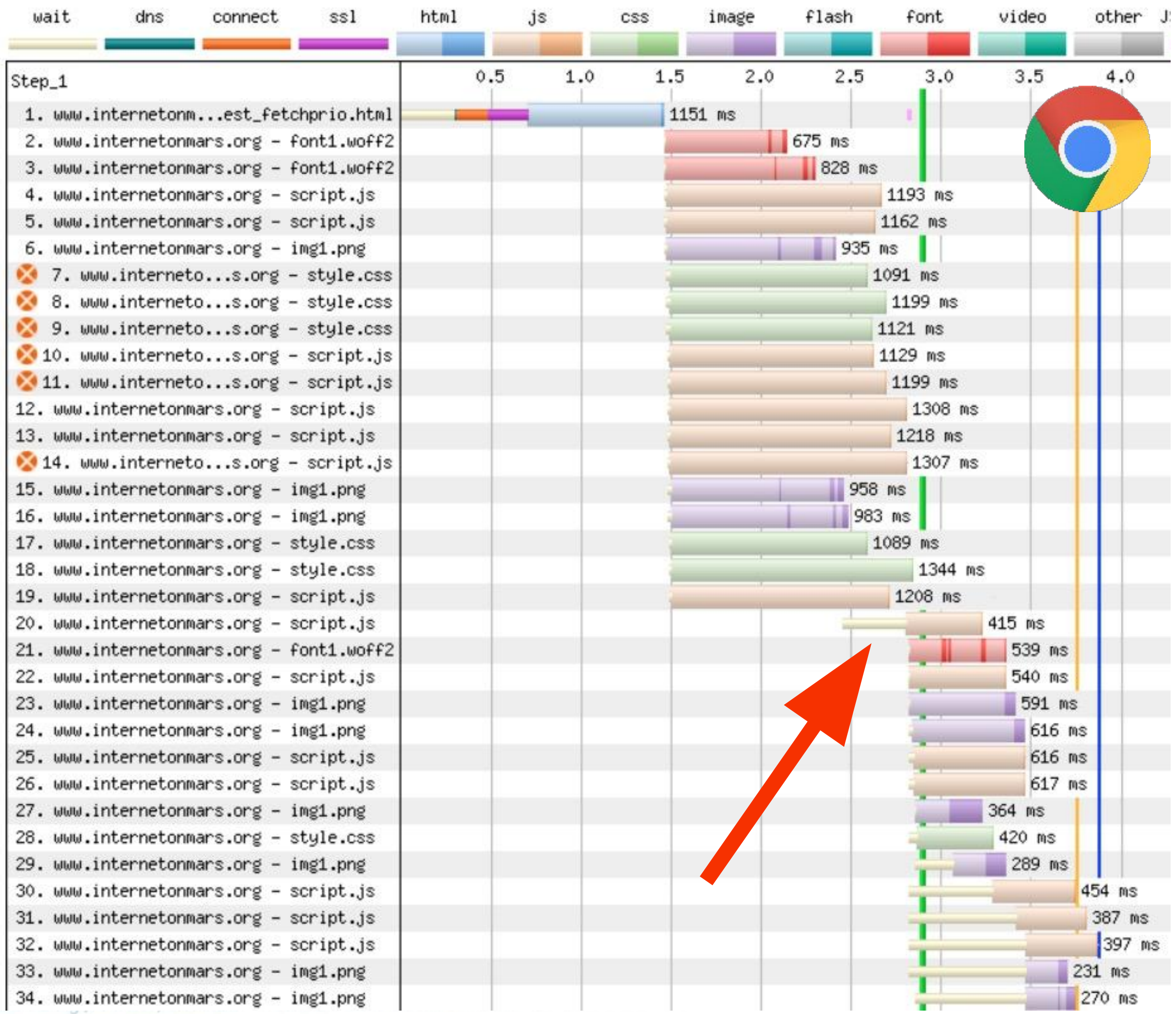
Safari Technology Preview Release 178 is now available for download for macOS Sonoma beta and macOS Ventura. If you already have Safari Technology Preview installed, you can update it in System Settings under General → Software Update.

Web API

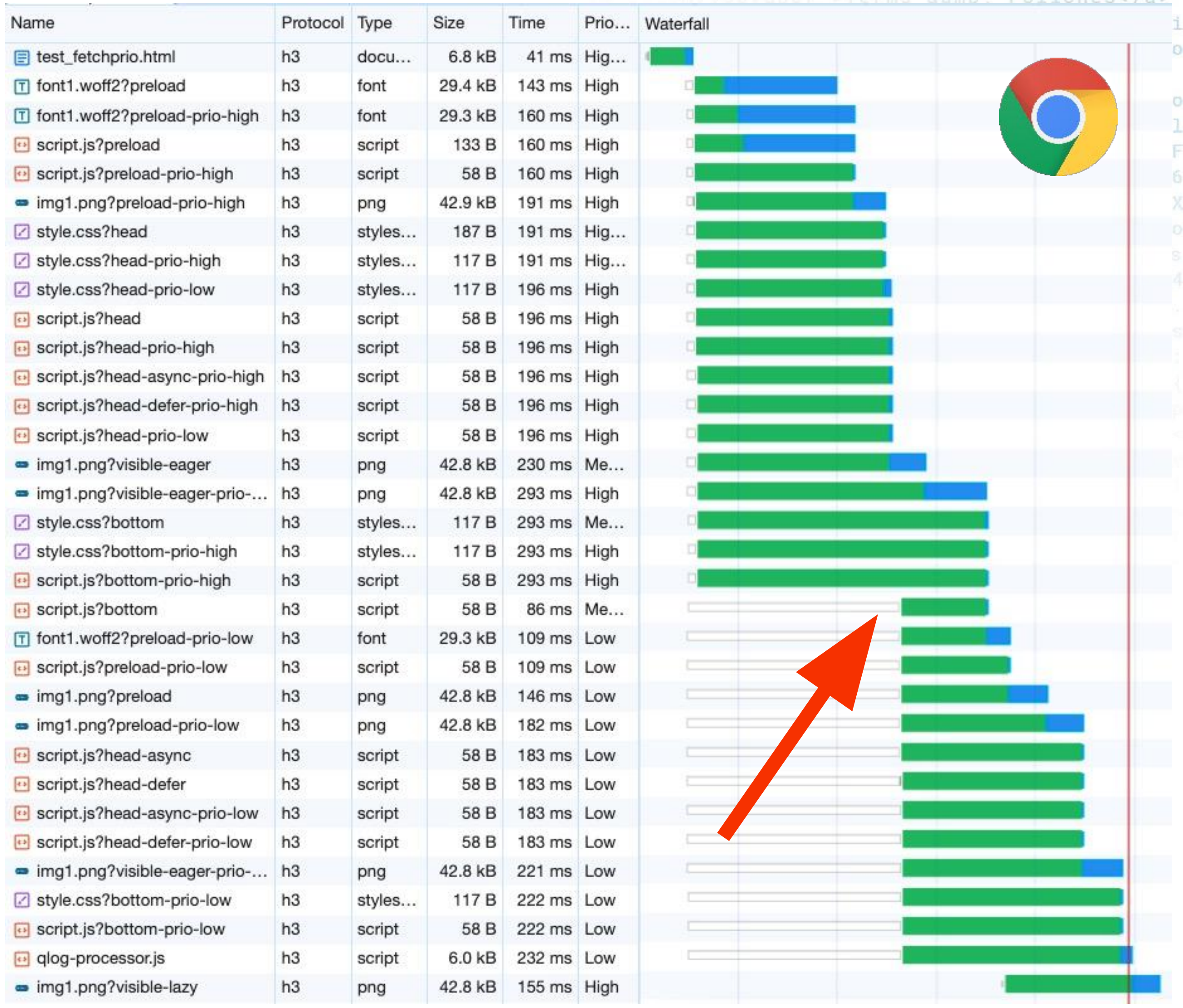
- Enabled Fetch Priority by default (267196@main)

https://bugzilla.mozilla.org/show_bug.cgi?id=1797715
https://webkit.org/blog/14435/release-notes-for-safari-technology-preview-178/

tight mode / two-phase loading

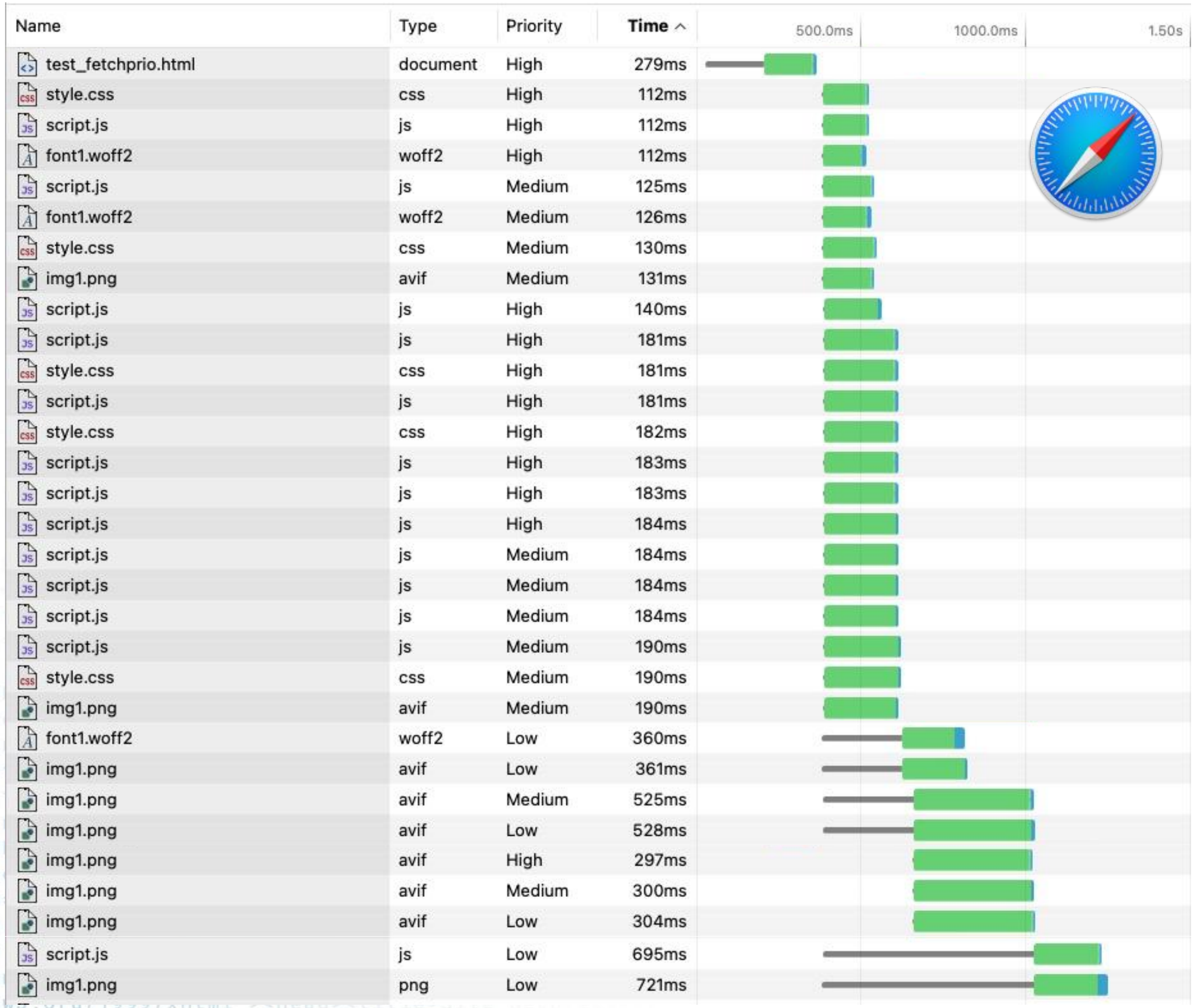


	Request	Time
1.	www.internetom...est_fetchprio.html	1151 ms
2.	www.internetonmars.org - font1.woff2	675 ms
3.	www.internetonmars.org - font1.woff2	828 ms
4.	www.internetonmars.org - script.js	1193 ms
5.	www.internetonmars.org - script.js	1162 ms
6.	www.internetonmars.org - img1.png	935 ms
7.	www.interneto...s.org - style.css	1091 ms
8.	www.interneto...s.org - style.css	1199 ms
9.	www.interneto...s.org - style.css	1121 ms
10.	www.interneto...s.org - script.js	1129 ms
11.	www.interneto...s.org - script.js	1199 ms
12.	www.internetonmars.org - script.js	1308 ms
13.	www.internetonmars.org - script.js	1218 ms
14.	www.interneto...s.org - script.js	1307 ms
15.	www.internetonmars.org - img1.png	958 ms
16.	www.internetonmars.org - img1.png	983 ms
17.	www.internetonmars.org - style.css	1089 ms
18.	www.internetonmars.org - style.css	1344 ms
19.	www.internetonmars.org - script.js	1208 ms
20.	www.internetonmars.org - script.js	415 ms
21.	www.internetonmars.org - font1.woff2	539 ms
22.	www.internetonmars.org - script.js	540 ms
23.	www.internetonmars.org - img1.png	591 ms
24.	www.internetonmars.org - img1.png	616 ms
25.	www.internetonmars.org - script.js	616 ms
26.	www.internetonmars.org - script.js	617 ms
27.	www.internetonmars.org - img1.png	364 ms
28.	www.internetonmars.org - style.css	420 ms
29.	www.internetonmars.org - img1.png	289 ms
30.	www.internetonmars.org - script.js	454 ms
31.	www.internetonmars.org - script.js	387 ms
32.	www.internetonmars.org - script.js	397 ms
33.	www.internetonmars.org - img1.png	231 ms
34.	www.internetonmars.org - img1.png	270 ms

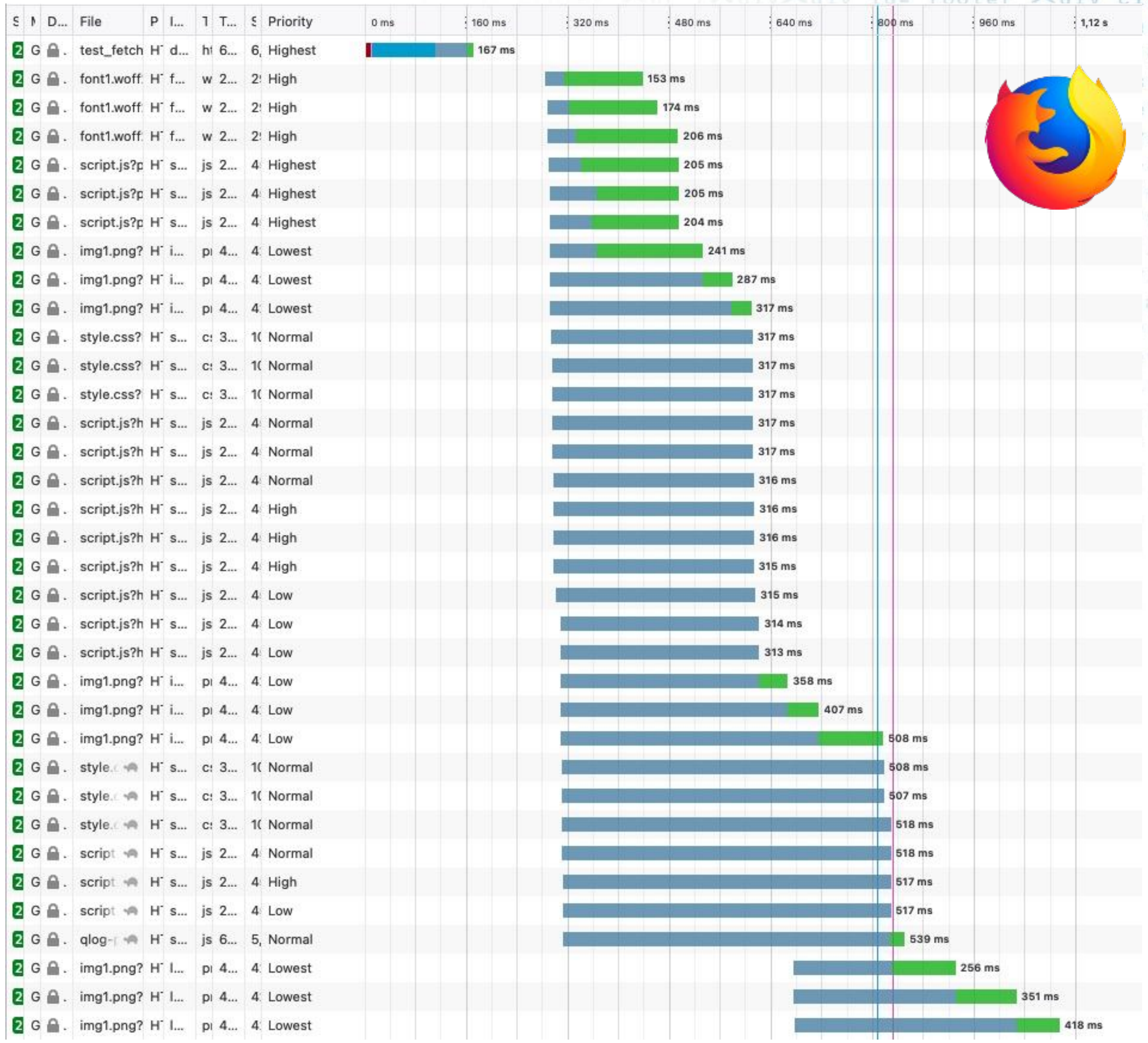


Name	Protocol	Type	Size	Time	Prio...	Waterfall
test_fetchprio.html	h3	docu...	6.8 kB	41 ms	Hig...	
font1.woff2?preload	h3	font	29.4 kB	143 ms	High	
font1.woff2?preload-prio-high	h3	font	29.3 kB	160 ms	High	
script.js?preload	h3	script	133 B	160 ms	High	
script.js?preload-prio-high	h3	script	58 B	160 ms	High	
img1.png?preload-prio-high	h3	png	42.9 kB	191 ms	High	
style.css?head	h3	styles...	187 B	191 ms	Hig...	
style.css?head-prio-high	h3	styles...	117 B	191 ms	Hig...	
style.css?head-prio-low	h3	styles...	117 B	196 ms	High	
script.js?head	h3	script	58 B	196 ms	High	
script.js?head-prio-high	h3	script	58 B	196 ms	High	
script.js?head-async-prio-high	h3	script	58 B	196 ms	High	
script.js?head-defer-prio-high	h3	script	58 B	196 ms	High	
script.js?head-prio-low	h3	script	58 B	196 ms	High	
img1.png?visible-eager	h3	png	42.8 kB	230 ms	Me...	
img1.png?visible-eager-prio-...	h3	png	42.8 kB	293 ms	High	
style.css?bottom	h3	styles...	117 B	293 ms	Me...	
style.css?bottom-prio-high	h3	styles...	117 B	293 ms	High	
script.js?bottom-prio-high	h3	script	58 B	293 ms	High	
script.js?bottom	h3	script	58 B	86 ms	Me...	
font1.woff2?preload-prio-low	h3	font	29.3 kB	109 ms	Low	
script.js?preload-prio-low	h3	script	58 B	109 ms	Low	
img1.png?preload	h3	png	42.8 kB	146 ms	Low	
img1.png?preload-prio-low	h3	png	42.8 kB	182 ms	Low	
script.js?head-async	h3	script	58 B	183 ms	Low	
script.js?head-defer	h3	script	58 B	183 ms	Low	
script.js?head-async-prio-low	h3	script	58 B	183 ms	Low	
script.js?head-defer-prio-low	h3	script	58 B	183 ms	Low	
img1.png?visible-eager-prio-...	h3	png	42.8 kB	221 ms	Low	
style.css?bottom-prio-low	h3	styles...	117 B	222 ms	Low	
script.js?bottom-prio-low	h3	script	58 B	222 ms	Low	
qlog-processor.js	h3	script	6.0 kB	232 ms	Low	
img1.png?visible-lazy	h3	png	42.8 kB	155 ms	High	

Also happens in Safari / Firefox



Name	Type	Priority	Time
test_fetchprio.html	document	High	279ms
style.css	css	High	112ms
script.js	js	High	112ms
font1.woff2	woff2	High	112ms
script.js	js	Medium	125ms
font1.woff2	woff2	Medium	126ms
style.css	css	Medium	130ms
img1.png	avif	Medium	131ms
script.js	js	High	140ms
script.js	js	High	181ms
style.css	css	High	181ms
script.js	js	High	181ms
style.css	css	High	182ms
script.js	js	High	183ms
script.js	js	High	183ms
script.js	js	High	184ms
script.js	js	Medium	184ms
script.js	js	Medium	184ms
script.js	js	Medium	184ms
script.js	js	Medium	190ms
style.css	css	Medium	190ms
img1.png	avif	Medium	190ms
font1.woff2	woff2	Low	360ms
img1.png	avif	Low	361ms
img1.png	avif	Medium	525ms
img1.png	avif	Low	528ms
img1.png	avif	High	297ms
img1.png	avif	Medium	300ms
img1.png	avif	Low	304ms
script.js	js	Low	695ms
img1.png	png	Low	721ms



S	N	D...	File	P	I...	T	T...	S	Priority	
2	G	🔒.	test_fetch	H	d...	ht	6...	6,	Highest	167 ms
2	G	🔒.	font1.woff	H	f...	w	2...	2!	High	153 ms
2	G	🔒.	font1.woff	H	f...	w	2...	2!	High	174 ms
2	G	🔒.	font1.woff	H	f...	w	2...	2!	High	206 ms
2	G	🔒.	script.js?p	H	s...	js	2...	4	Highest	205 ms
2	G	🔒.	script.js?p	H	s...	js	2...	4	Highest	205 ms
2	G	🔒.	script.js?p	H	s...	js	2...	4	Highest	204 ms
2	G	🔒.	img1.png?	H	i...	pı	4...	4	Lowest	241 ms
2	G	🔒.	img1.png?	H	i...	pı	4...	4	Lowest	287 ms
2	G	🔒.	img1.png?	H	i...	pı	4...	4	Lowest	317 ms
2	G	🔒.	style.css?	H	s...	c:	3...	1(	Normal	317 ms
2	G	🔒.	style.css?	H	s...	c:	3...	1(	Normal	317 ms
2	G	🔒.	style.css?	H	s...	c:	3...	1(	Normal	317 ms
2	G	🔒.	script.js?h	H	s...	js	2...	4	Normal	317 ms
2	G	🔒.	script.js?h	H	s...	js	2...	4	Normal	317 ms
2	G	🔒.	script.js?h	H	s...	js	2...	4	Normal	316 ms
2	G	🔒.	script.js?h	H	s...	js	2...	4	High	316 ms
2	G	🔒.	script.js?h	H	s...	js	2...	4	High	316 ms
2	G	🔒.	script.js?h	H	s...	js	2...	4	High	315 ms
2	G	🔒.	script.js?h	H	s...	js	2...	4	Low	315 ms
2	G	🔒.	script.js?h	H	s...	js	2...	4	Low	314 ms
2	G	🔒.	script.js?h	H	s...	js	2...	4	Low	313 ms
2	G	🔒.	img1.png?	H	i...	pı	4...	4	Low	358 ms
2	G	🔒.	img1.png?	H	i...	pı	4...	4	Low	407 ms
2	G	🔒.	img1.png?	H	i...	pı	4...	4	Low	508 ms
2	G	🔒.	style.	H	s...	c:	3...	1(	Normal	508 ms
2	G	🔒.	style.	H	s...	c:	3...	1(	Normal	507 ms
2	G	🔒.	style.	H	s...	c:	3...	1(	Normal	518 ms
2	G	🔒.	script	H	s...	js	2...	4	Normal	518 ms
2	G	🔒.	script	H	s...	js	2...	4	High	517 ms
2	G	🔒.	script	H	s...	js	2...	4	Low	517 ms
2	G	🔒.	qlog-	H	s...	js	6...	5,	Normal	539 ms
2	G	🔒.	img1.png?	H	l...	pı	4...	4	Lowest	256 ms
2	G	🔒.	img1.png?	H	l...	pı	4...	4	Lowest	351 ms
2	G	🔒.	img1.png?	H	l...	pı	4...	4	Lowest	418 ms

Resource Fetch Prioritization and Scheduling in Chrome

Author: Patrick Meenan

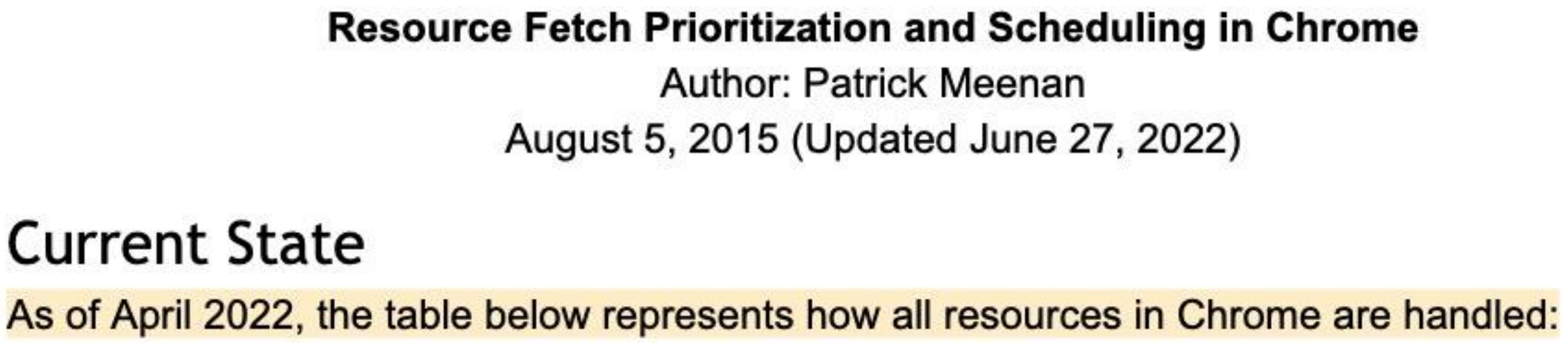
August 5, 2015 (Updated June 27, 2022)

Current State

As of April 2022, the table below represents how all resources in Chrome are handled:

	Load in "tight mode"		Conditionally load in "tight mode"		
Blink Priority	**VeryHigh**	**High**	**Medium**	**Low**	**VeryLow**
DevTools Priority	**Highest**	**High**	**Medium**	**Low**	**Lowest**
Main Resource	◉				
CSS*** (early**)	⬆◉	⬇			
CSS*** (late**)		⬆	◉	⬇	
Script (early** or not from preload scanner)		⬆◉		⬇	
Script (late**)		⬆	◉	⬇	
Script (async/defer)		⬆		◉⬇	

https://web.dev/articles/fetch-priority
https://imkev.dev/fetchpriority-opportunity
https://docs.google.com/document/d/1bCDuq9H1ih9iNjgzyAL0gpwNFiEP4TZS-YLRp_RuMlc



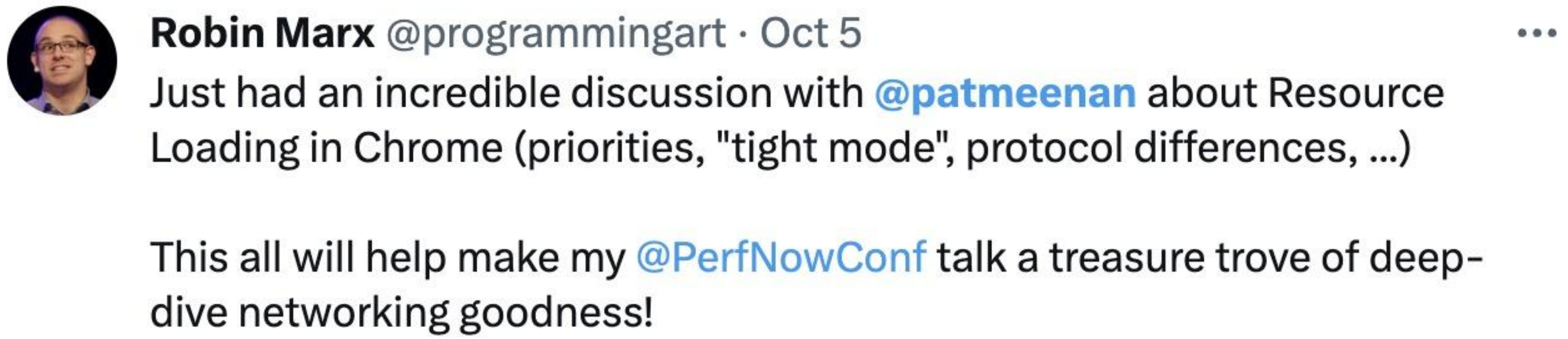
Resource Fetch Prioritization and Scheduling in Chrome

Author: Patrick Meenan

August 5, 2015 (Updated June 27, 2022)

Current State

As of April 2022, the table below represents how all resources in Chrome are handled:



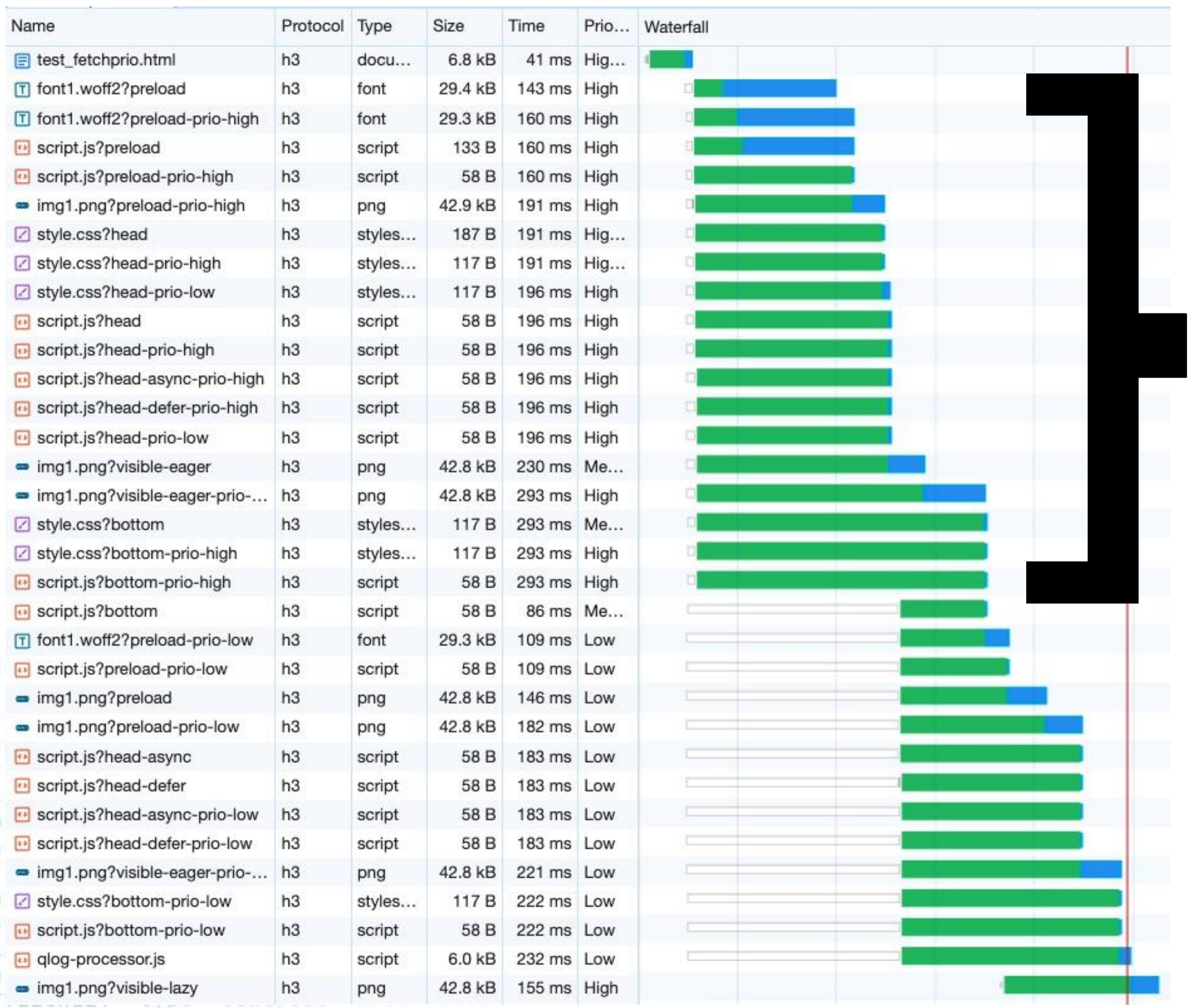
Robin Marx @programmingart · Oct 5

Just had an incredible discussion with @patmeenan about Resource Loading in Chrome (priorities, "tight mode", protocol differences, ...)

This all will help make my @PerfNowConf talk a treasure trove of deep-dive networking goodness!

https://web.dev/articles/fetch-priority
https://imkev.dev/fetchpriority-opportunity
https://docs.google.com/document/d/1bCDuq9H1ih9iNjgzyAL0gpwNFiEP4TZS-YLRp_RuMlc

Limit amount of critical in-flight resources



Name	Protocol	Type	Size	Time	Prio…	Waterfall
test_fetchprio.html	h3	docu…	6.8 kB	41 ms	Hig…	
font1.woff2?preload	h3	font	29.4 kB	143 ms	High	
font1.woff2?preload-prio-high	h3	font	29.3 kB	160 ms	High	
script.js?preload	h3	script	133 B	160 ms	High	
script.js?preload-prio-high	h3	script	58 B	160 ms	High	
img1.png?preload-prio-high	h3	png	42.9 kB	191 ms	High	
style.css?head	h3	styles…	187 B	191 ms	Hig…	
style.css?head-prio-high	h3	styles…	117 B	191 ms	Hig…	
style.css?head-prio-low	h3	styles…	117 B	196 ms	High	
script.js?head	h3	script	58 B	196 ms	High	
script.js?head-prio-high	h3	script	58 B	196 ms	High	
script.js?head-async-prio-high	h3	script	58 B	196 ms	High	
script.js?head-defer-prio-high	h3	script	58 B	196 ms	High	
script.js?head-prio-low	h3	script	58 B	196 ms	High	
img1.png?visible-eager	h3	png	42.8 kB	230 ms	Me…	
img1.png?visible-eager-prio-…	h3	png	42.8 kB	293 ms	High	
style.css?bottom	h3	styles…	117 B	293 ms	Me…	
style.css?bottom-prio-high	h3	styles…	117 B	293 ms	High	
script.js?bottom-prio-high	h3	script	58 B	293 ms	High	
script.js?bottom	h3	script	58 B	86 ms	Me…	
font1.woff2?preload-prio-low	h3	font	29.3 kB	109 ms	Low	
script.js?preload-prio-low	h3	script	58 B	109 ms	Low	
img1.png?preload	h3	png	42.8 kB	146 ms	Low	
img1.png?preload-prio-low	h3	png	42.8 kB	182 ms	Low	
script.js?head-async	h3	script	58 B	183 ms	Low	
script.js?head-defer	h3	script	58 B	183 ms	Low	
script.js?head-async-prio-low	h3	script	58 B	183 ms	Low	
script.js?head-defer-prio-low	h3	script	58 B	183 ms	Low	
img1.png?visible-eager-prio-…	h3	png	42.8 kB	221 ms	Low	
style.css?bottom-prio-low	h3	styles…	117 B	222 ms	Low	
script.js?bottom-prio-low	h3	script	58 B	222 ms	Low	
qlog-processor.js	h3	script	6.0 kB	232 ms	Low	
img1.png?visible-lazy	h3	png	42.8 kB	155 ms	High	

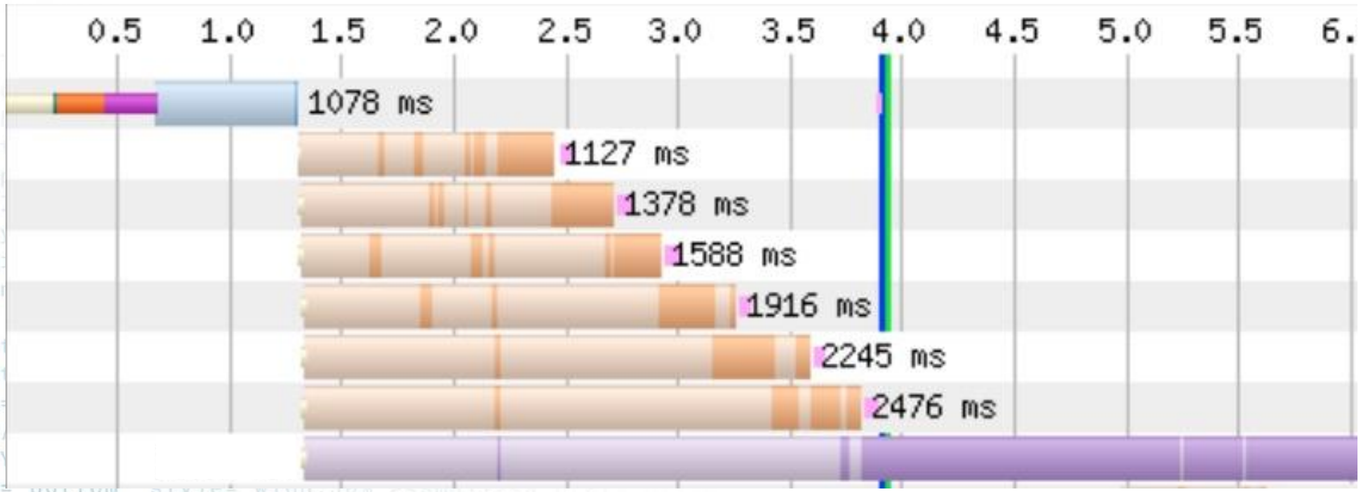
All High + Highest priority
and
2-4 Medium and Low priority

Rest of Medium, Low
and Lowest priority

Akamai Experience the Edge

Is this a good idea?

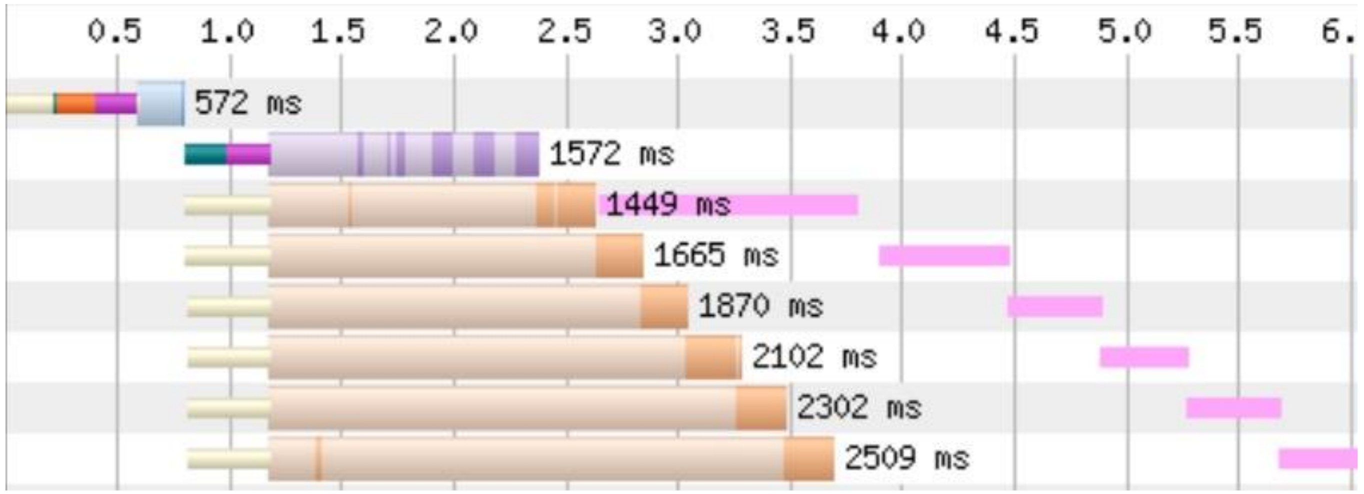
```
<link as="image" rel="preload" href="poster.jpg" fetchpriority="high">
```



preload on **bottom** of <head> + fetchpriority = high

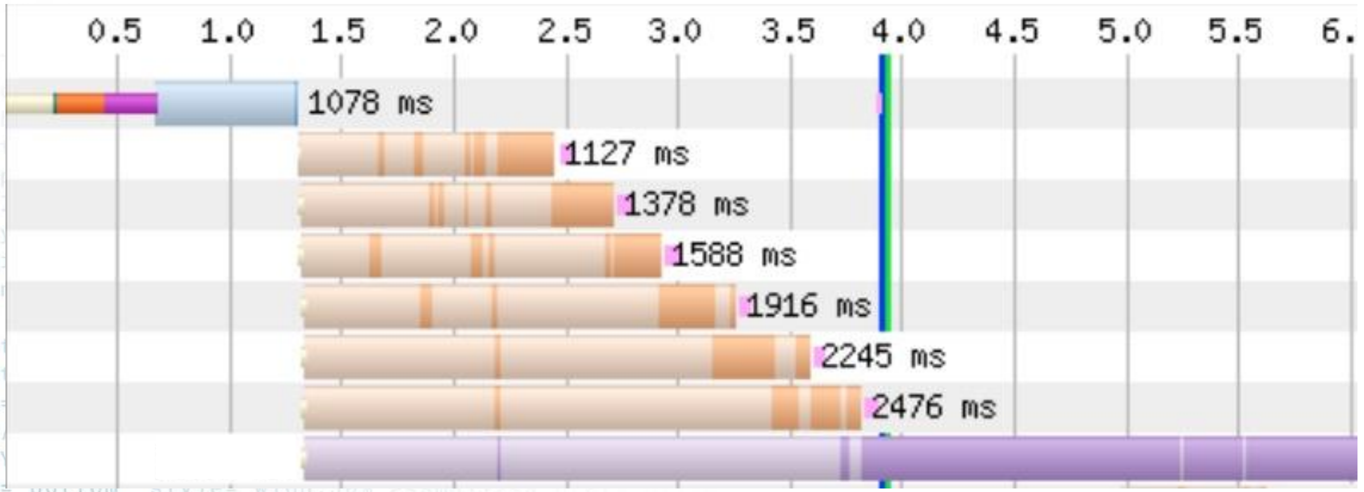
= fetched **after** parser-blocking JS

Preload + Fetchpriority LCP *can* delay JS!



preload on **top** of <head> + fetchpriority = high

= fetched **before** parser-blocking JS

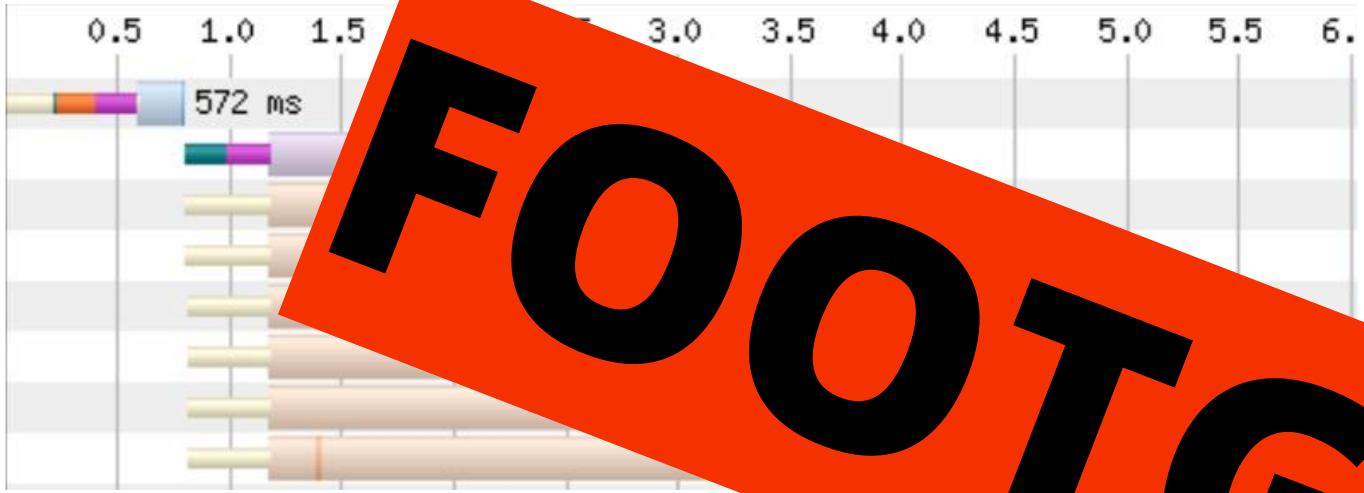


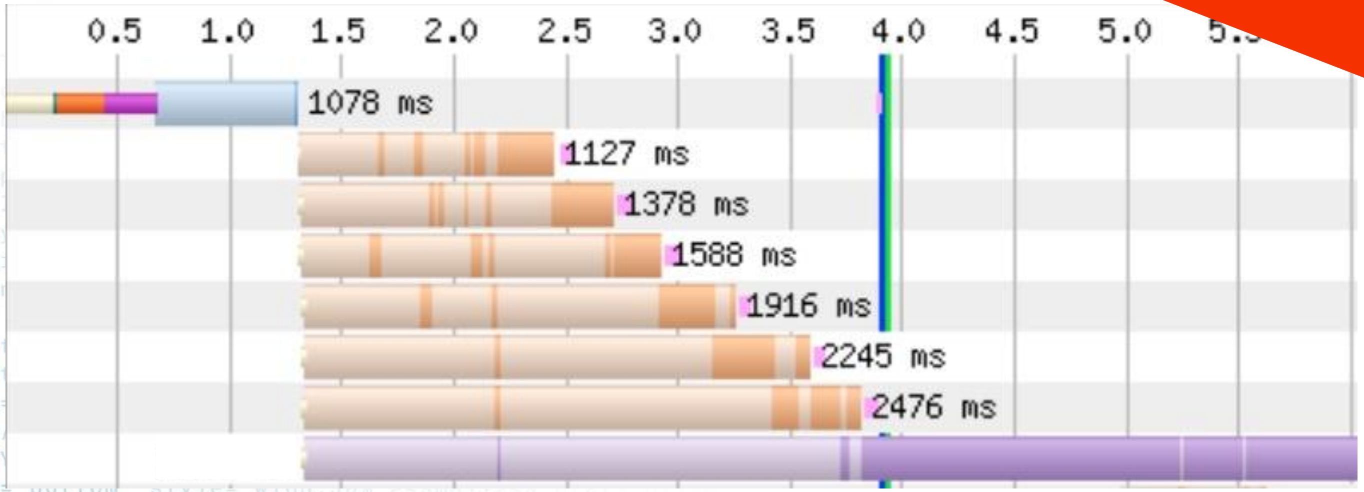
preload on **bottom** of <head> + fetchpriority = high

= fetched **after** parser-blocking JS

Akamai *Experience the Edge*

Preload + Fetchpriority LCP can delay JS!





FOOTGUN!?

preload on **top** of <head> +
fetchpriority = high

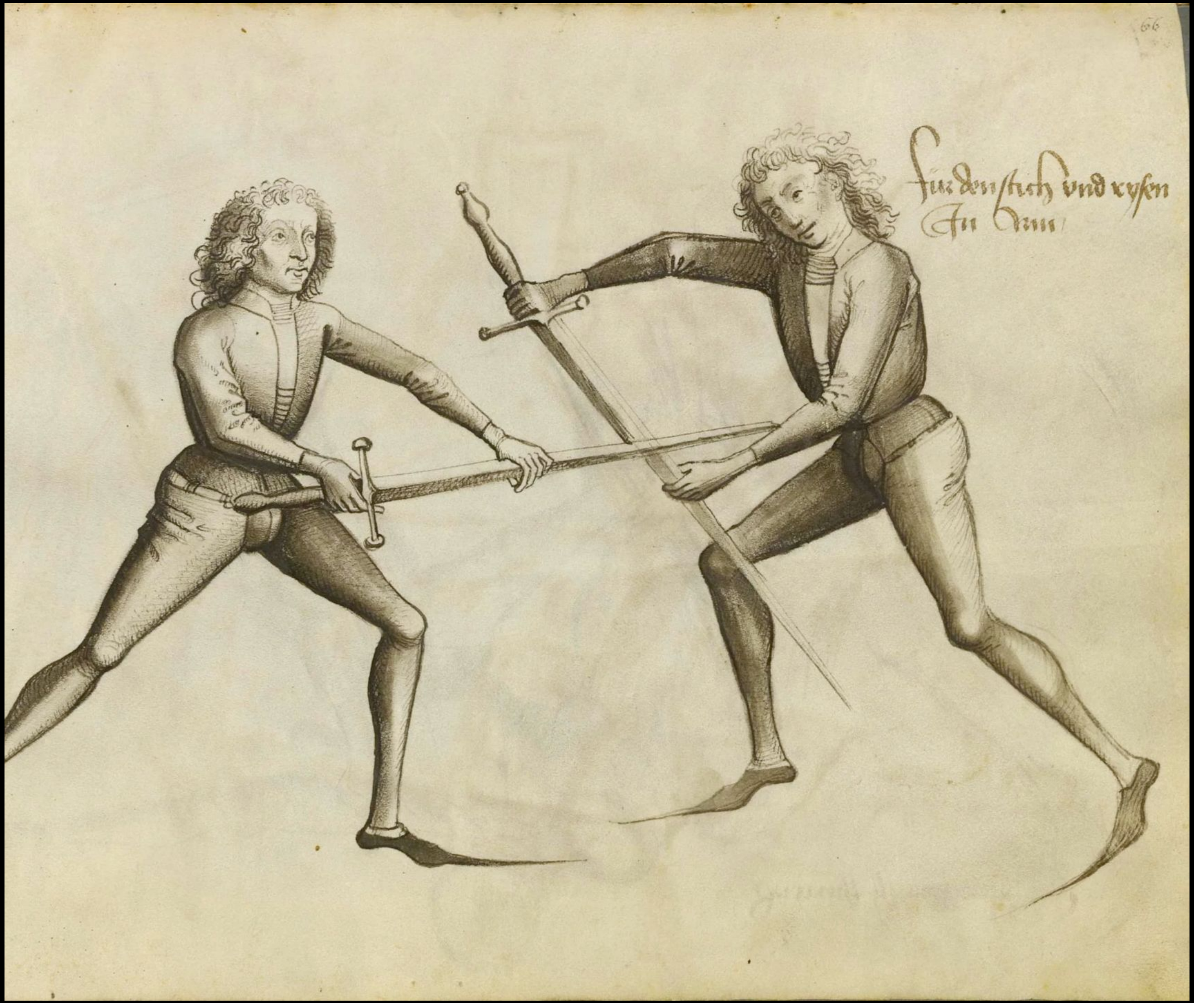
tched **before** parser-blocking JS

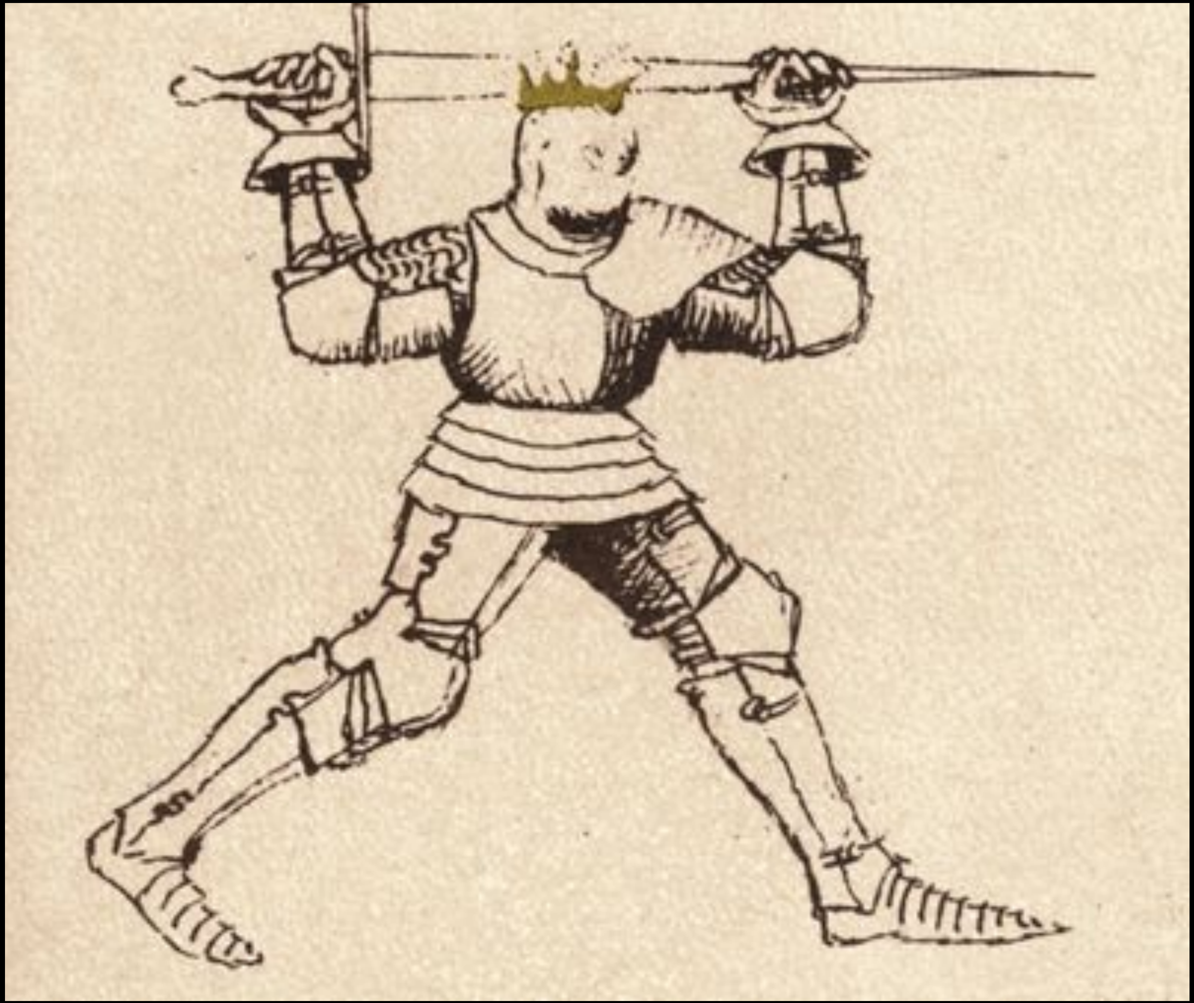
preloa <head> +
fetchpriority

= fetched **after** parser-blocking JS

Akamai Experience the Edge

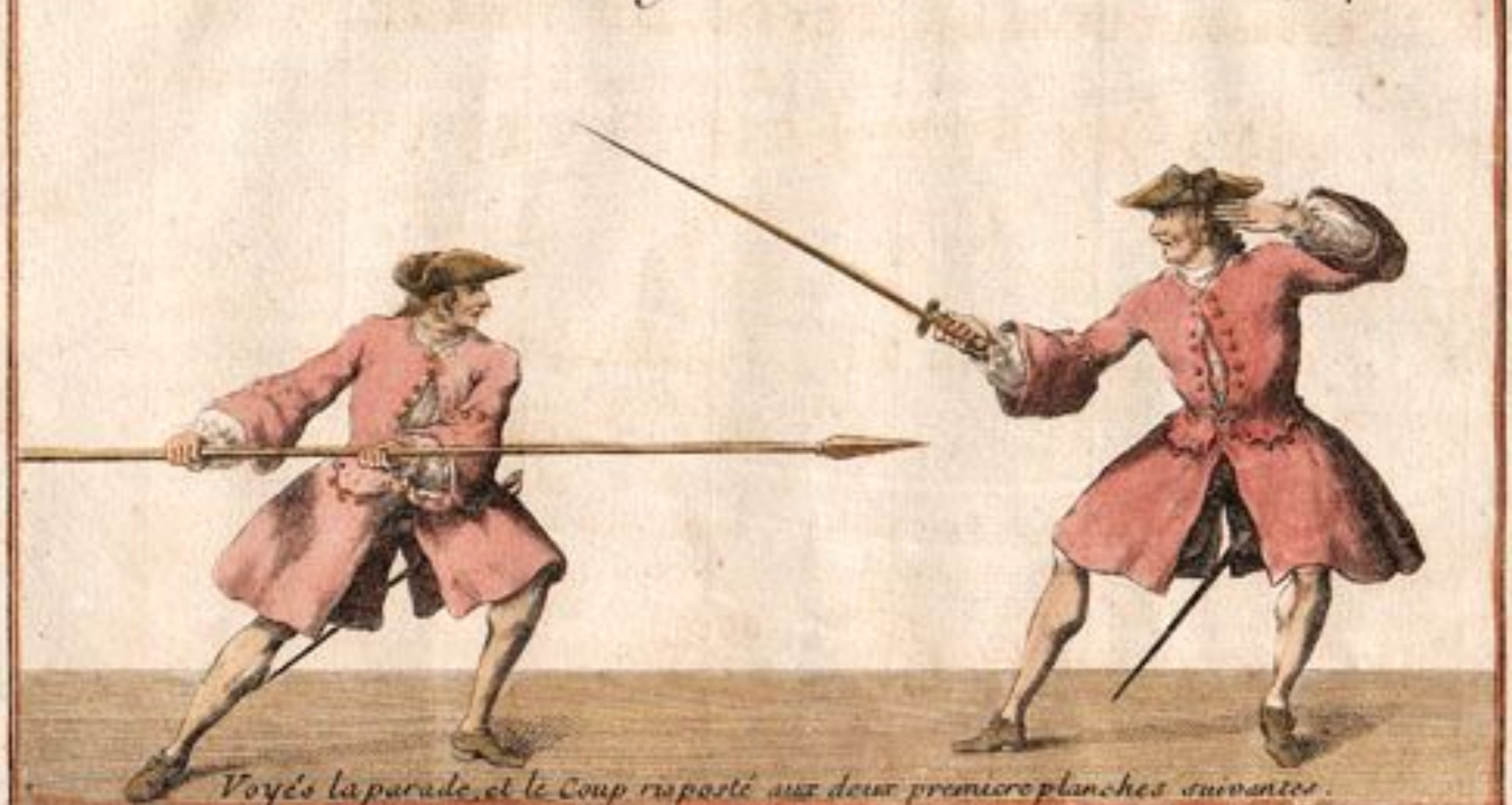
THIS IS FINE.


66
Fur den ſtich vnd ryſen
Zu dem



Garde opposée aux piques, halebardes, bayonnettes au bout du fusil &c.

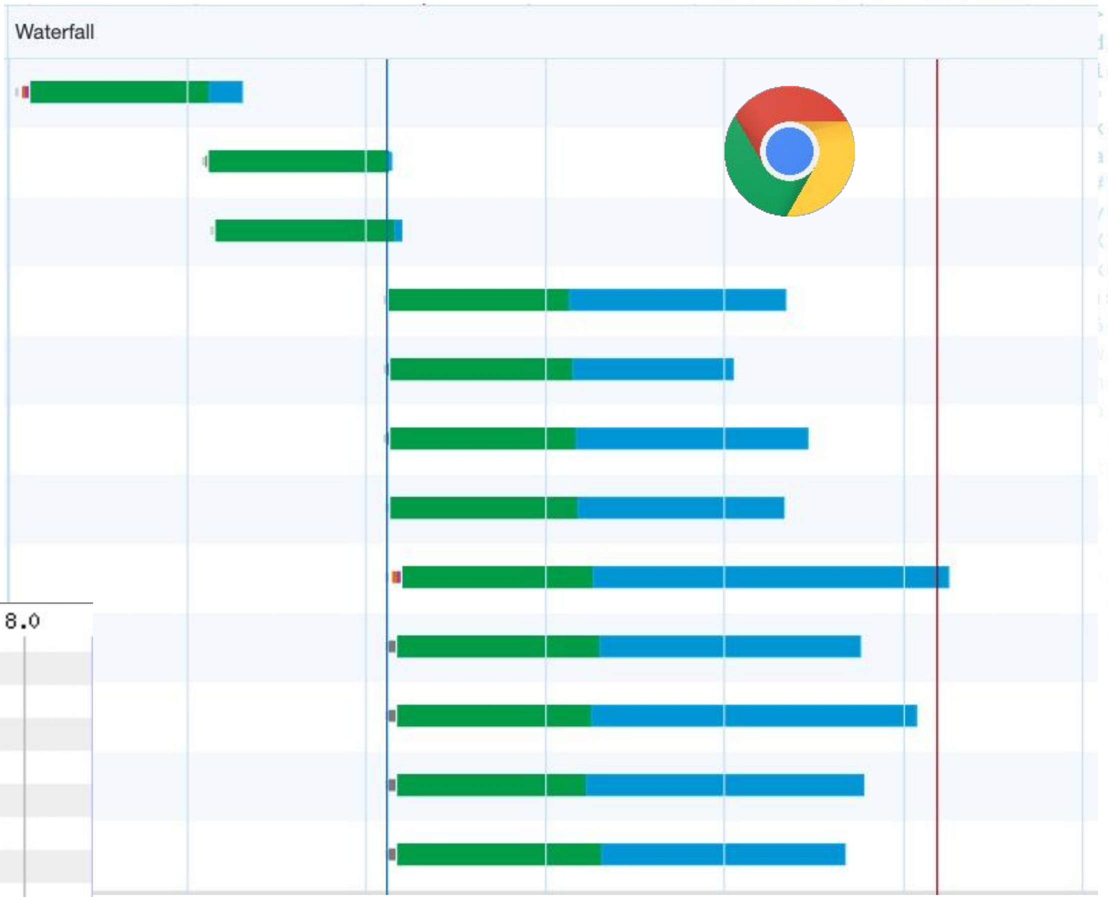
78e. planche

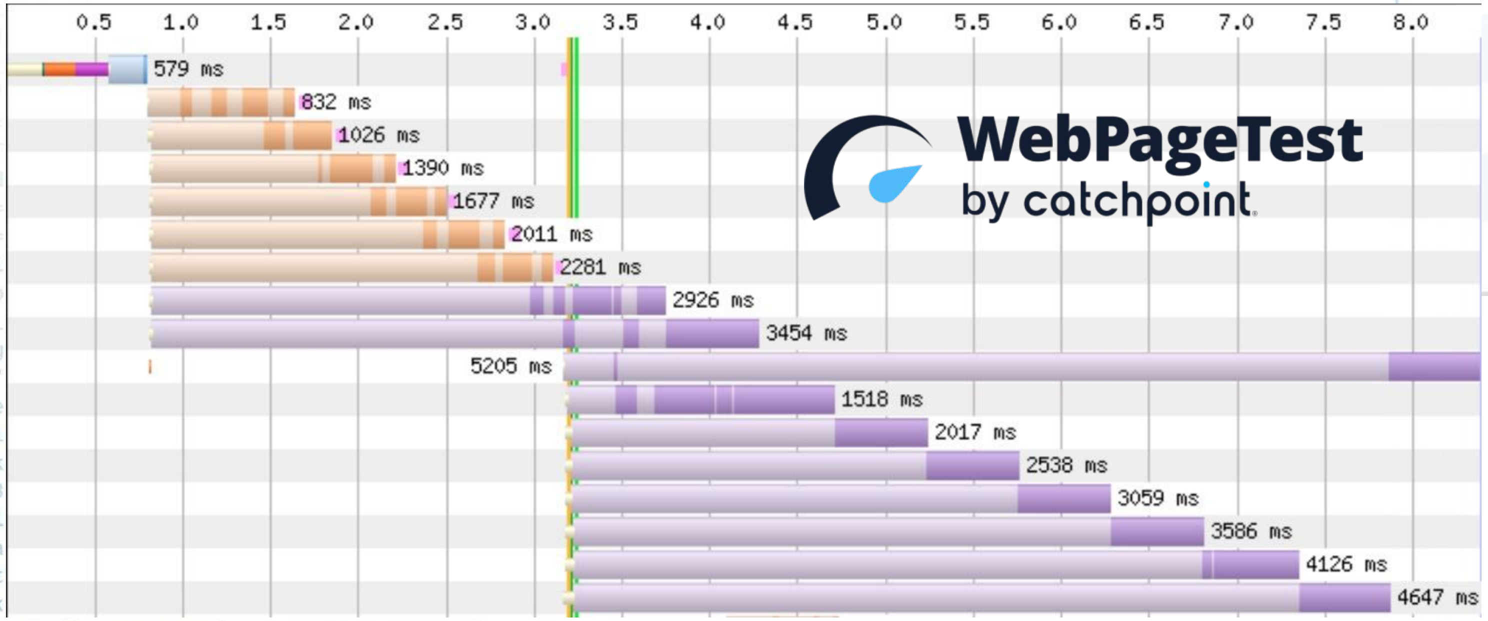


Voyés la parade, et le Coup riposté aux deux premiere planches suivantes.

NETWORKING FEATURES
WEBPERF TOOLS

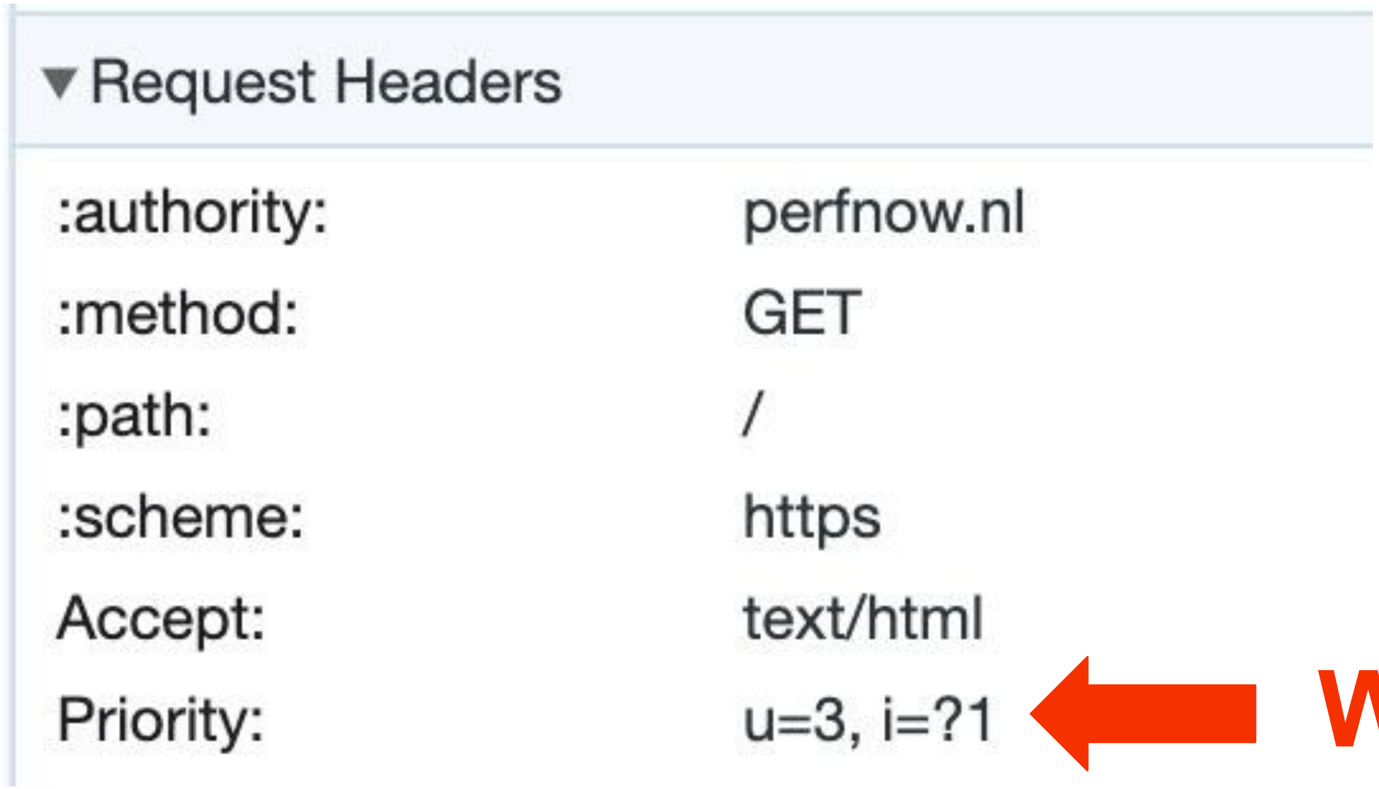
Chrome's Waterfall has to up its game





Akamai Experience the Edge

Priority Header not visible in browser devtools

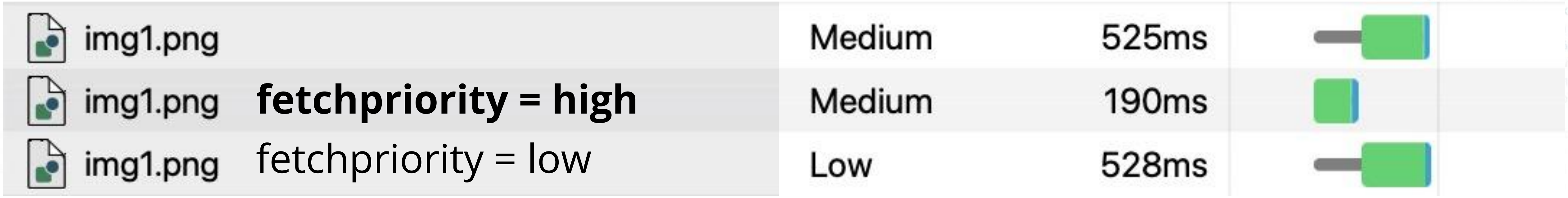


▼ Request Headers

:authority:	perfnow.nl
:method:	GET
:path:	/
:scheme:	https
Accept:	text/html
Priority:	u=3, i=?1

Where is it?!?

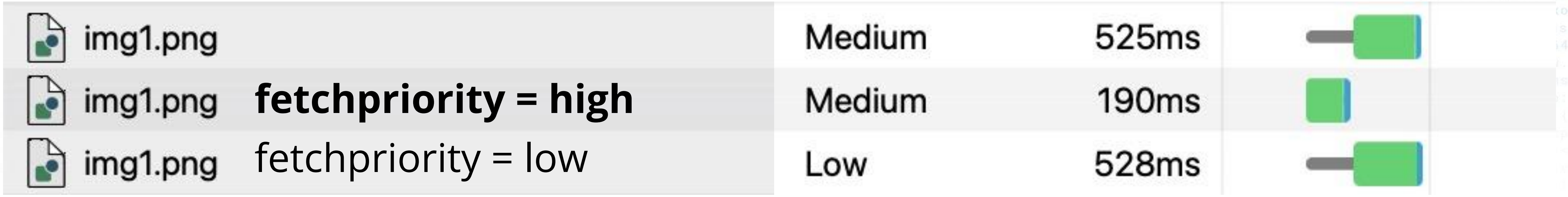
Safari devtools don’t (always) show final priority



img1.png		Medium	525ms	
img1.png	**fetchpriority = high**	Medium	190ms	
img1.png	fetchpriority = low	Low	528ms	

Akamai *Experience the Edge*

Safari devtools don’t (always) show final priority

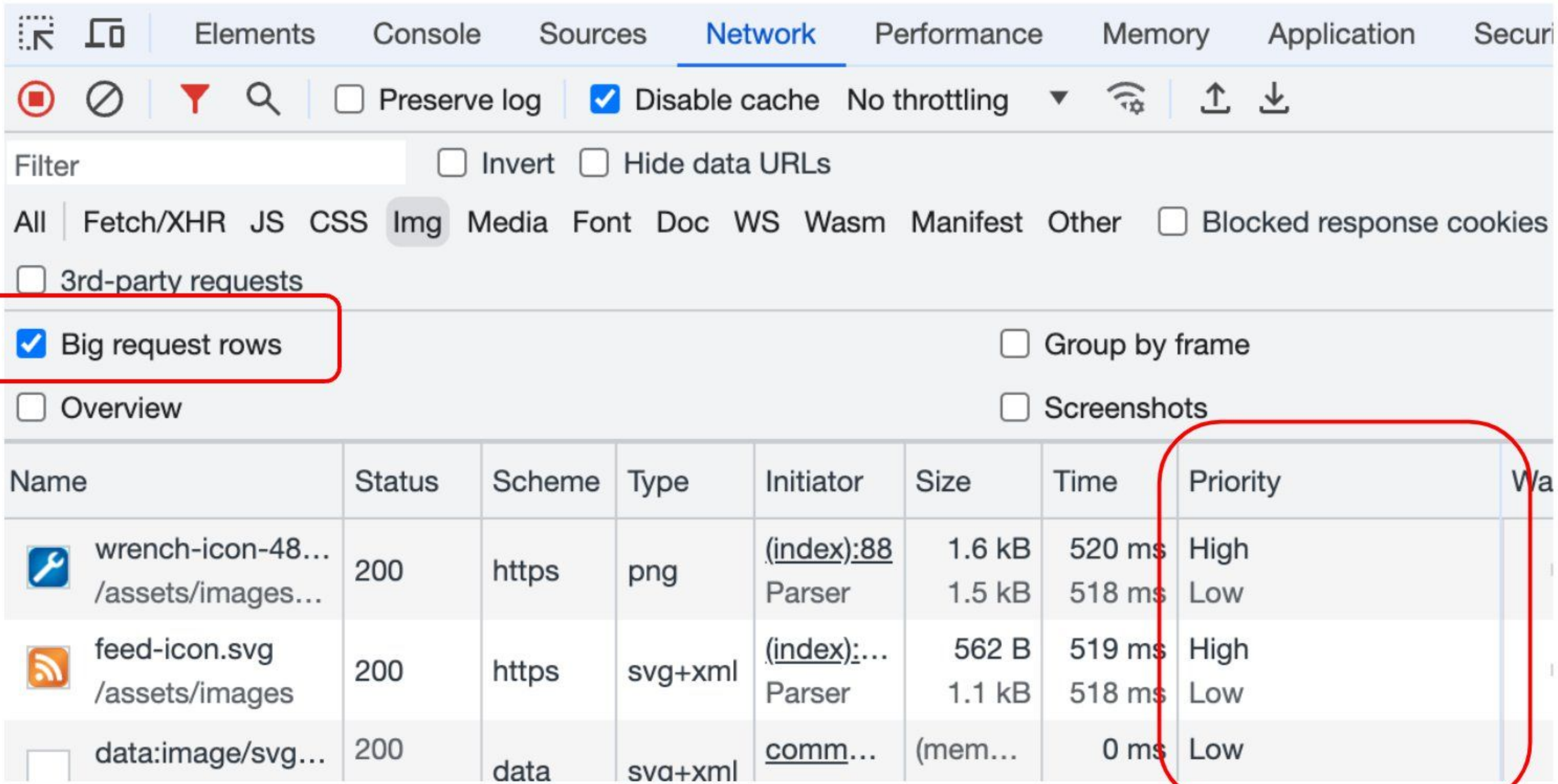


img1.png		Medium	525ms
img1.png	**fetchpriority = high**	Medium	190ms
img1.png	fetchpriority = low	Low	528ms

Requested as Medium,
then **immediately** updated to High

Akamai Experience the Edge

Chrome now does show this correctly



Elements Console Sources Network Performance Memory Application Securi

Preserve log | [✓] Disable cache | No throttling

Filter | Invert | Hide data URLs

All | Fetch/XHR | JS | CSS | Img | Media | Font | Doc | WS | Wasm | Manifest | Other | Blocked response cookies

3rd-party requests

[✓] Big request rows | Group by frame

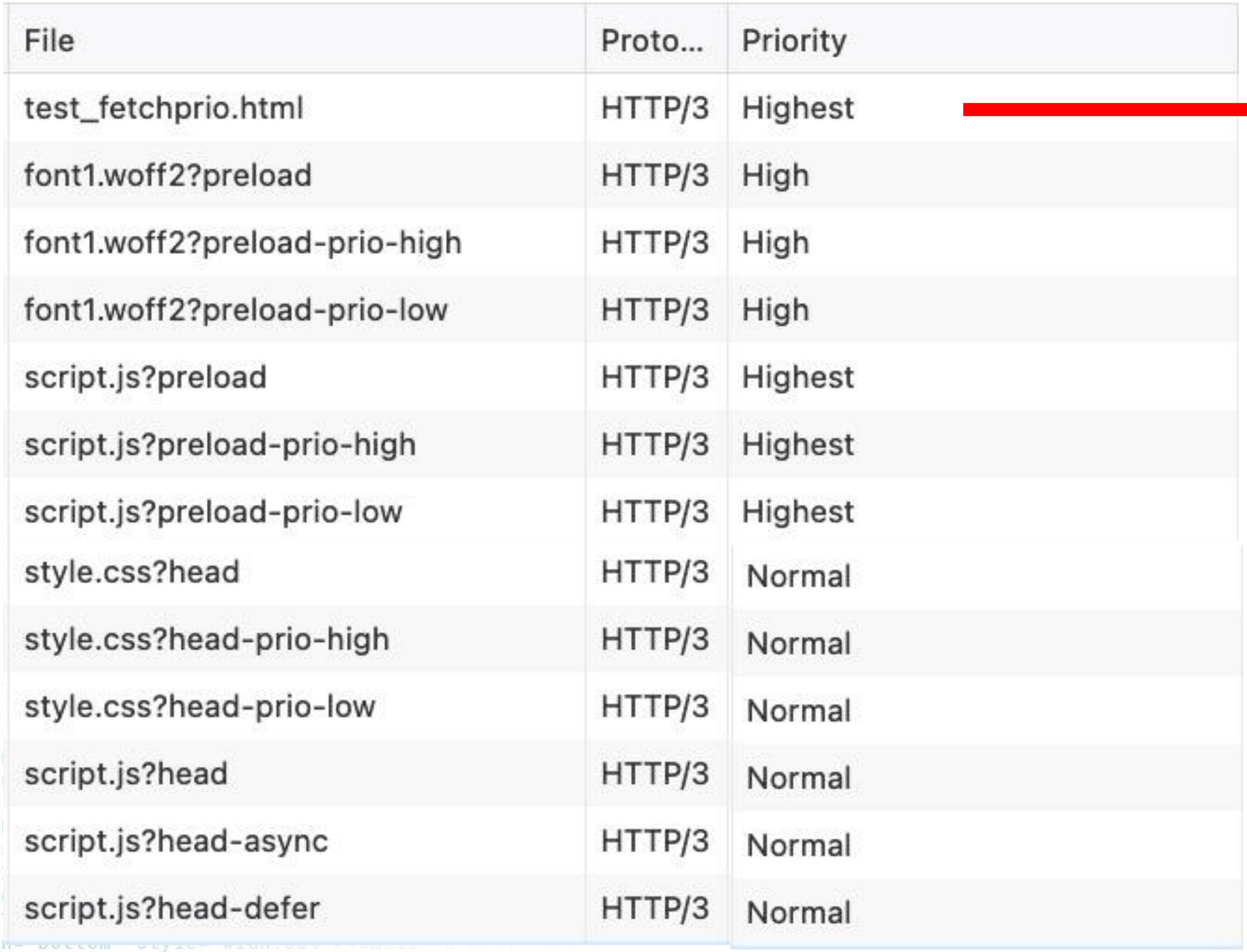
Overview | Screenshots

Name	Status	Scheme	Type	Initiator	Size	Time	Priority	Wa
wrench-icon-48… /assets/images…	200	https	png	(index):88 Parser	1.6 kB 1.5 kB	520 ms 518 ms	High Low	
feed-icon.svg /assets/images	200	https	svg+xml	(index):… Parser	562 B 1.1 kB	519 ms 518 ms	High Low	
data:image/svg…	200	data	svg+xml	comm…	(mem…	0 ms	Low	

https://x.com/tunetheweb/status/1681692340455391234

sky
Also on NOW
LIES
LIES
MORE LIES

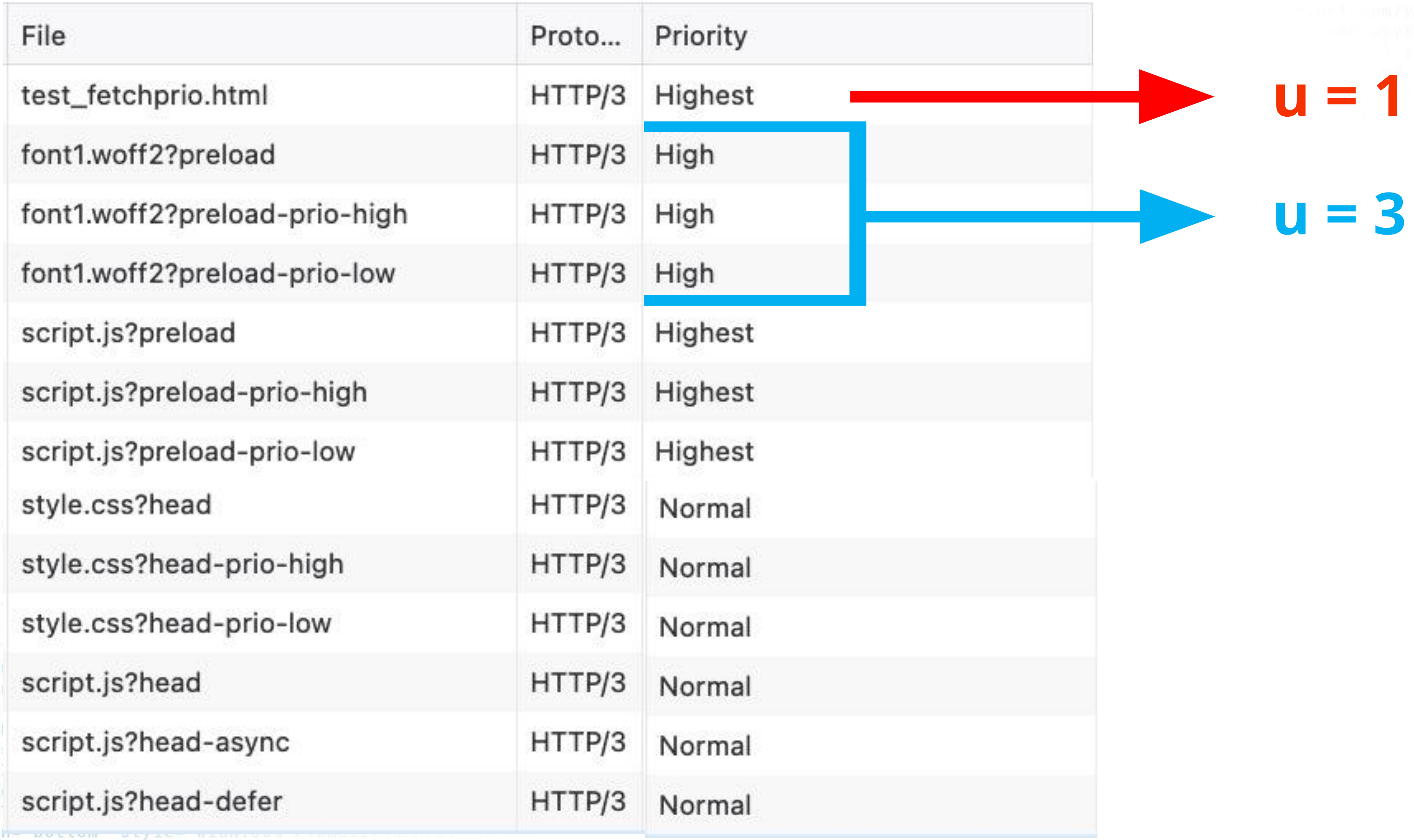
Firefox sits on a throne of lies



File	Proto...	Priority
test_fetchprio.html	HTTP/3	Highest
font1.woff2?preload	HTTP/3	High
font1.woff2?preload-prio-high	HTTP/3	High
font1.woff2?preload-prio-low	HTTP/3	High
script.js?preload	HTTP/3	Highest
script.js?preload-prio-high	HTTP/3	Highest
script.js?preload-prio-low	HTTP/3	Highest
style.css?head	HTTP/3	Normal
style.css?head-prio-high	HTTP/3	Normal
style.css?head-prio-low	HTTP/3	Normal
script.js?head	HTTP/3	Normal
script.js?head-async	HTTP/3	Normal
script.js?head-defer	HTTP/3	Normal

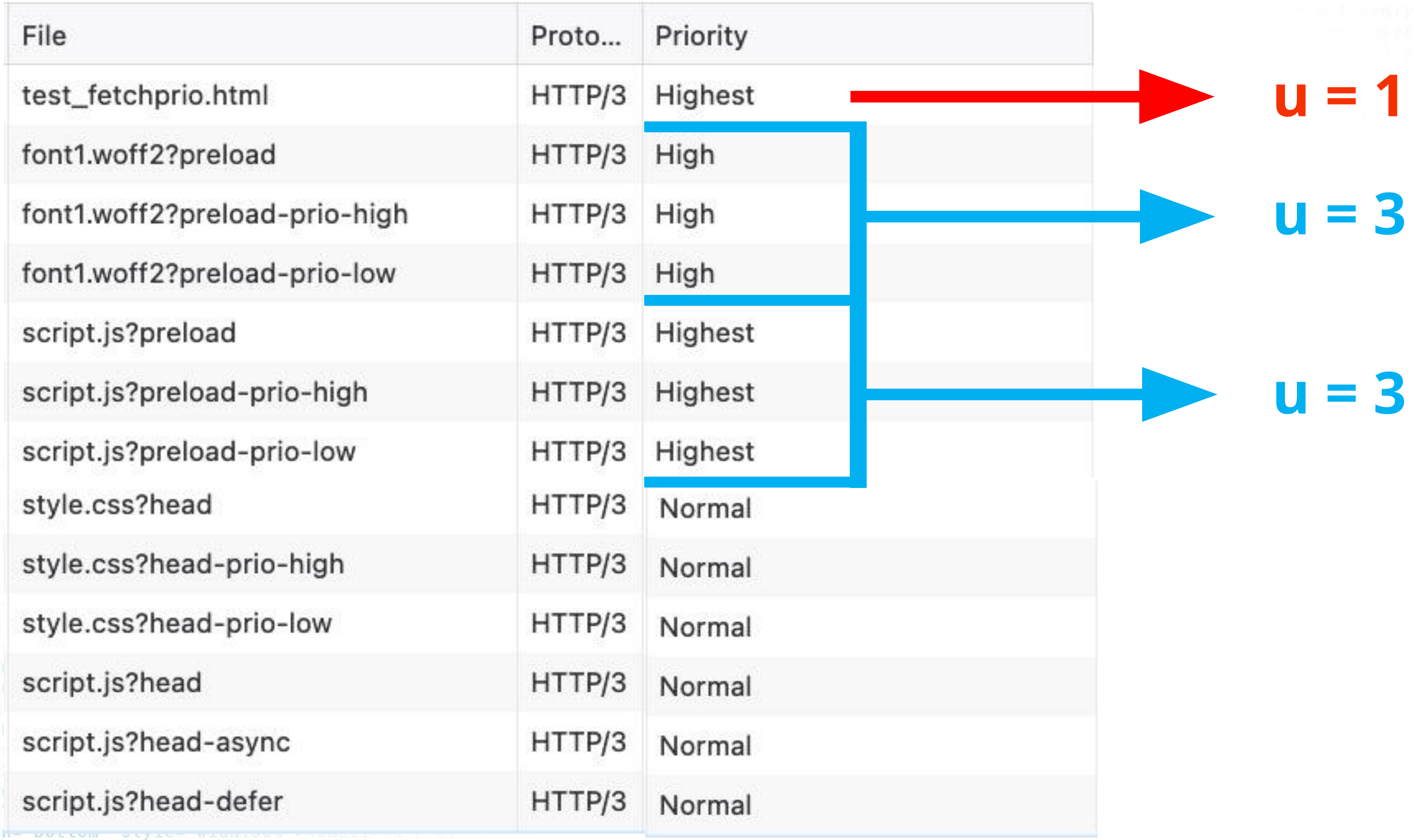
$\longrightarrow$ u = 1

Firefox sits on a throne of lies



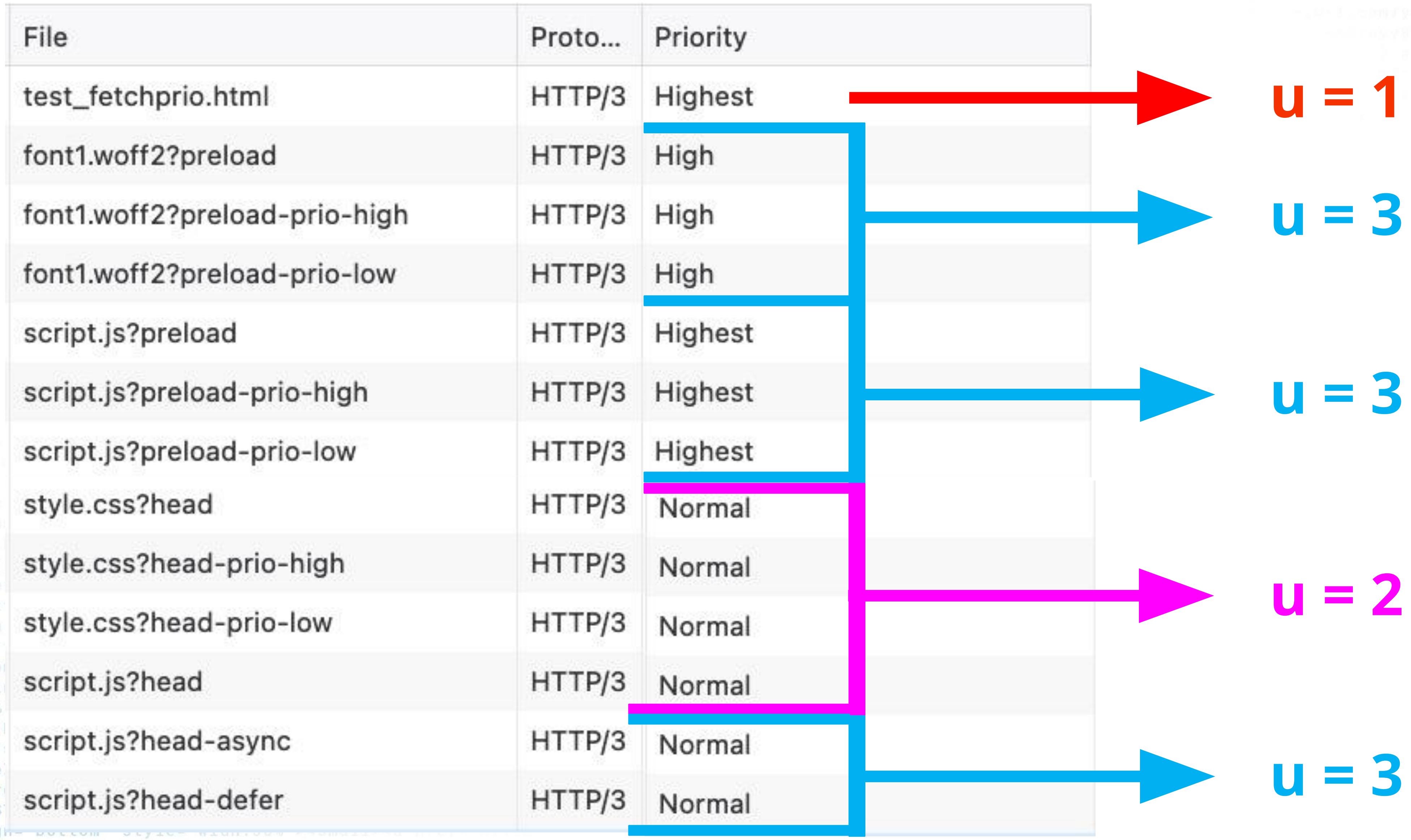
File	Proto...	Priority
test_fetchprio.html	HTTP/3	Highest
font1.woff2?preload	HTTP/3	High
font1.woff2?preload-prio-high	HTTP/3	High
font1.woff2?preload-prio-low	HTTP/3	High
script.js?preload	HTTP/3	Highest
script.js?preload-prio-high	HTTP/3	Highest
script.js?preload-prio-low	HTTP/3	Highest
style.css?head	HTTP/3	Normal
style.css?head-prio-high	HTTP/3	Normal
style.css?head-prio-low	HTTP/3	Normal
script.js?head	HTTP/3	Normal
script.js?head-async	HTTP/3	Normal
script.js?head-defer	HTTP/3	Normal

Firefox sits on a throne of lies



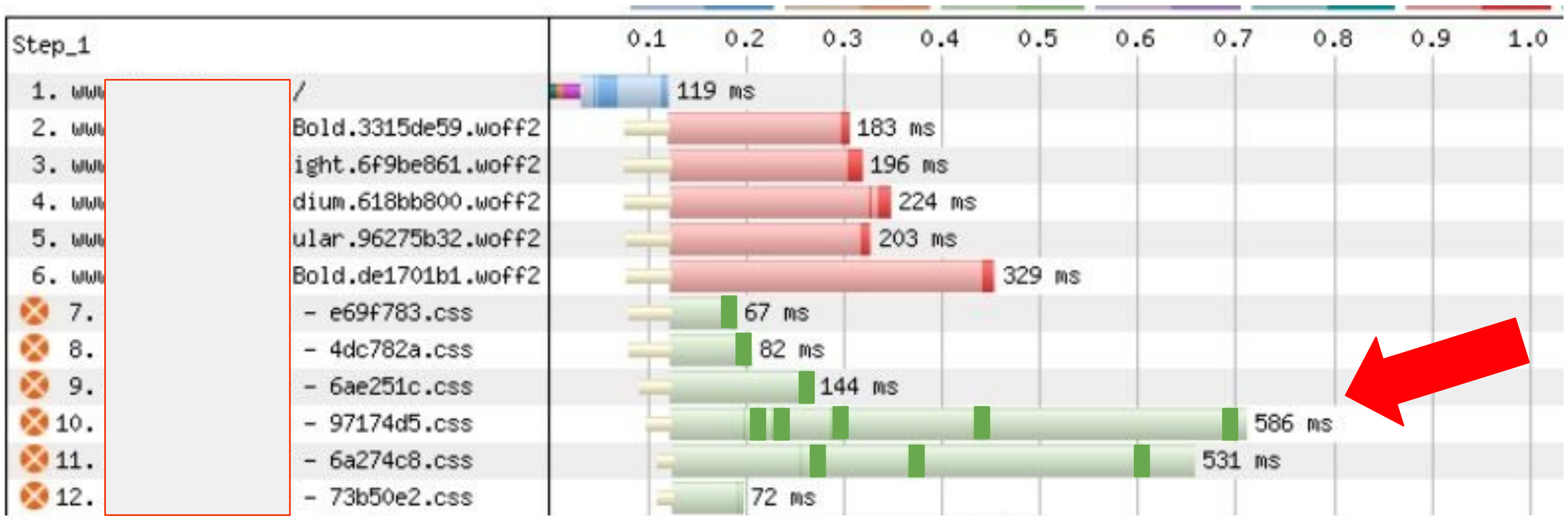
File	Proto...	Priority
test_fetchprio.html	HTTP/3	Highest
font1.woff2?preload	HTTP/3	High
font1.woff2?preload-prio-high	HTTP/3	High
font1.woff2?preload-prio-low	HTTP/3	High
script.js?preload	HTTP/3	Highest
script.js?preload-prio-high	HTTP/3	Highest
script.js?preload-prio-low	HTTP/3	Highest
style.css?head	HTTP/3	Normal
style.css?head-prio-high	HTTP/3	Normal
style.css?head-prio-low	HTTP/3	Normal
script.js?head	HTTP/3	Normal
script.js?head-async	HTTP/3	Normal
script.js?head-defer	HTTP/3	Normal

Firefox sits on a throne of lies



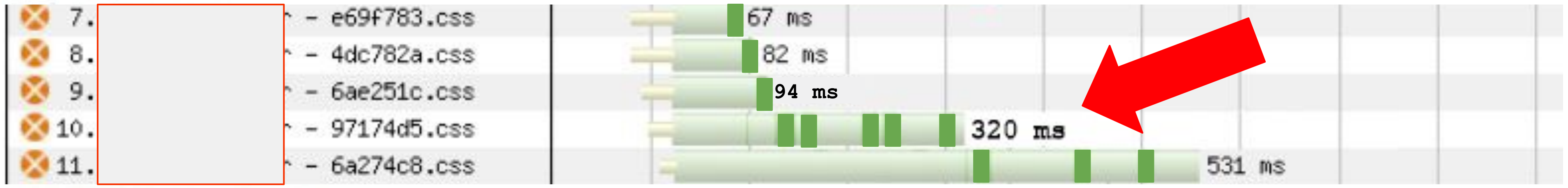
File	Proto...	Priority
test_fetchprio.html	HTTP/3	Highest
font1.woff2?preload	HTTP/3	High
font1.woff2?preload-prio-high	HTTP/3	High
font1.woff2?preload-prio-low	HTTP/3	High
script.js?preload	HTTP/3	Highest
script.js?preload-prio-high	HTTP/3	Highest
script.js?preload-prio-low	HTTP/3	Highest
style.css?head	HTTP/3	Normal
style.css?head-prio-high	HTTP/3	Normal
style.css?head-prio-low	HTTP/3	Normal
script.js?head	HTTP/3	Normal
script.js?head-async	HTTP/3	Normal
script.js?head-defer	HTTP/3	Normal

Tool 1: WebPageTest says: multiplexed CSS



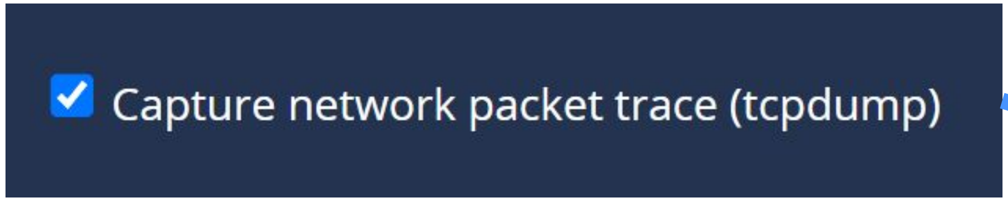
```
Step_1                         0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.0
 1. ww[redacted]/                   119 ms
 2. ww[redacted]Bold.3315de59.woff2  183 ms
 3. ww[redacted]ight.6f9be861.woff2  196 ms
 4. ww[redacted]dium.618bb800.woff2  224 ms
 5. ww[redacted]ular.96275b32.woff2  203 ms
 6. ww[redacted]Bold.de1701b1.woff2  329 ms
 7. [redacted] - e69f783.css         67 ms
 8. [redacted] - 4dc782a.css         82 ms
 9. [redacted] - 6ae251c.css         144 ms
10. [redacted] - 97174d5.css         586 ms
11. [redacted] - 6a274c8.css         531 ms
12. [redacted] - 73b50e2.css         72 ms
```

What we're expecting to see instead:

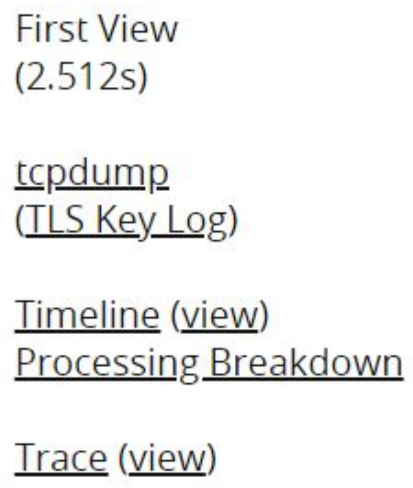


```
 7. [redacted] - e69f783.css   67 ms
 8. [redacted] - 4dc782a.css   82 ms
 9. [redacted] - 6ae251c.css   94 ms
10. [redacted] - 97174d5.css   320 ms
11. [redacted] - 6a274c8.css   531 ms
```

TCPDump / Wireshark to the Rescue!



✓ Capture network packet trace (tcpdump)



First View
(2.512s)

tcpdump
(TLS Key Log)

Timeline (view)
Processing Breakdown

Trace (view)

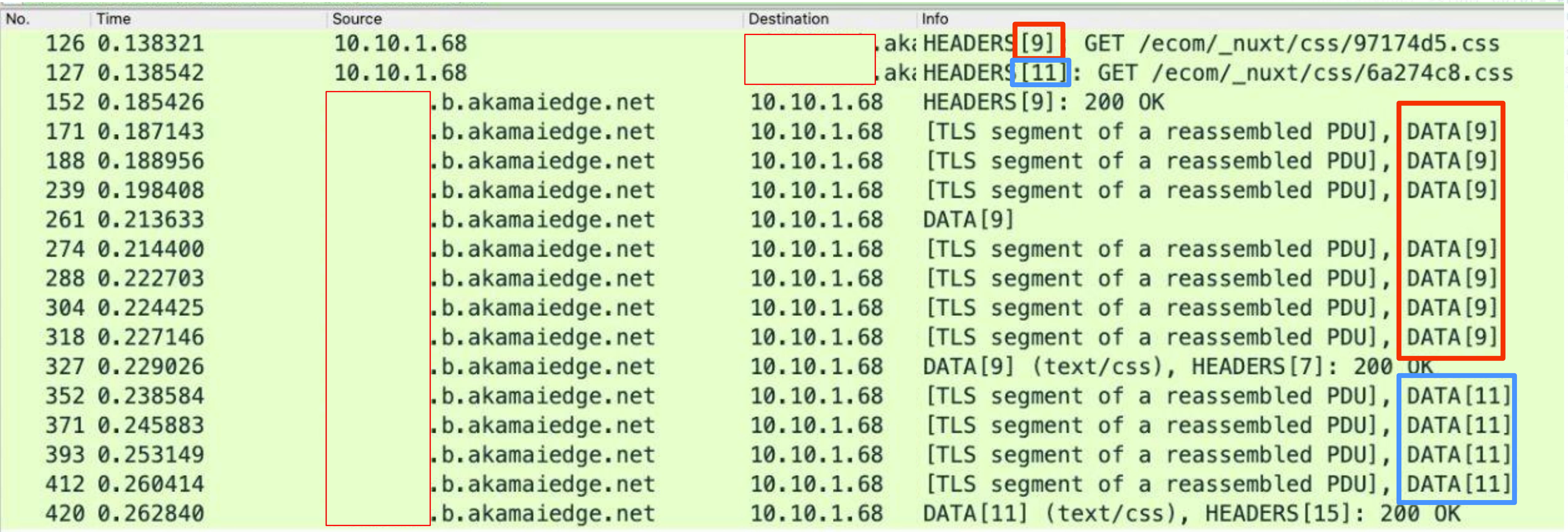
WIRESHARK

https://www.wireshark.org

Akamai

Tool 2: Wireshark says: sequential CSS

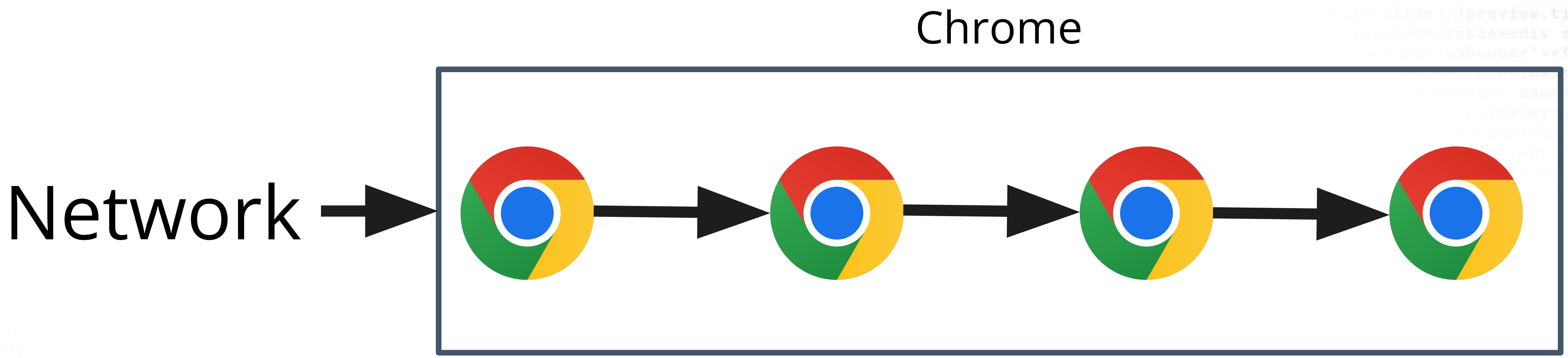
WIRESHARK



```
No.  Time      Source                       Destination       Info
126 0.138321   10.10.1.68                   [redacted].akamaiedge.net   HEADERS[9]: GET /ecom/_nuxt/css/97174d5.css
127 0.138542   10.10.1.68                   [redacted].akamaiedge.net   HEADERS[11]: GET /ecom/_nuxt/css/6a274c8.css
152 0.185426   [redacted].b.akamaiedge.net  10.10.1.68   HEADERS[9]: 200 OK
171 0.187143   [redacted].b.akamaiedge.net  10.10.1.68   [TLS segment of a reassembled PDU], DATA[9]
188 0.188956   [redacted].b.akamaiedge.net  10.10.1.68   [TLS segment of a reassembled PDU], DATA[9]
239 0.198408   [redacted].b.akamaiedge.net  10.10.1.68   [TLS segment of a reassembled PDU], DATA[9]
261 0.213633   [redacted].b.akamaiedge.net  10.10.1.68   DATA[9]
274 0.214400   [redacted].b.akamaiedge.net  10.10.1.68   [TLS segment of a reassembled PDU], DATA[9]
288 0.222703   [redacted].b.akamaiedge.net  10.10.1.68   [TLS segment of a reassembled PDU], DATA[9]
304 0.224425   [redacted].b.akamaiedge.net  10.10.1.68   [TLS segment of a reassembled PDU], DATA[9]
318 0.227146   [redacted].b.akamaiedge.net  10.10.1.68   [TLS segment of a reassembled PDU], DATA[9]
327 0.229026   [redacted].b.akamaiedge.net  10.10.1.68   DATA[9] (text/css), HEADERS[7]: 200 OK
352 0.238584   [redacted].b.akamaiedge.net  10.10.1.68   [TLS segment of a reassembled PDU], DATA[11]
371 0.245883   [redacted].b.akamaiedge.net  10.10.1.68   [TLS segment of a reassembled PDU], DATA[11]
393 0.253149   [redacted].b.akamaiedge.net  10.10.1.68   [TLS segment of a reassembled PDU], DATA[11]
412 0.260414   [redacted].b.akamaiedge.net  10.10.1.68   [TLS segment of a reassembled PDU], DATA[11]
420 0.262840   [redacted].b.akamaiedge.net  10.10.1.68   DATA[11] (text/css), HEADERS[15]: 200 OK
```

Akamai

Mismatch between Network and Renderer



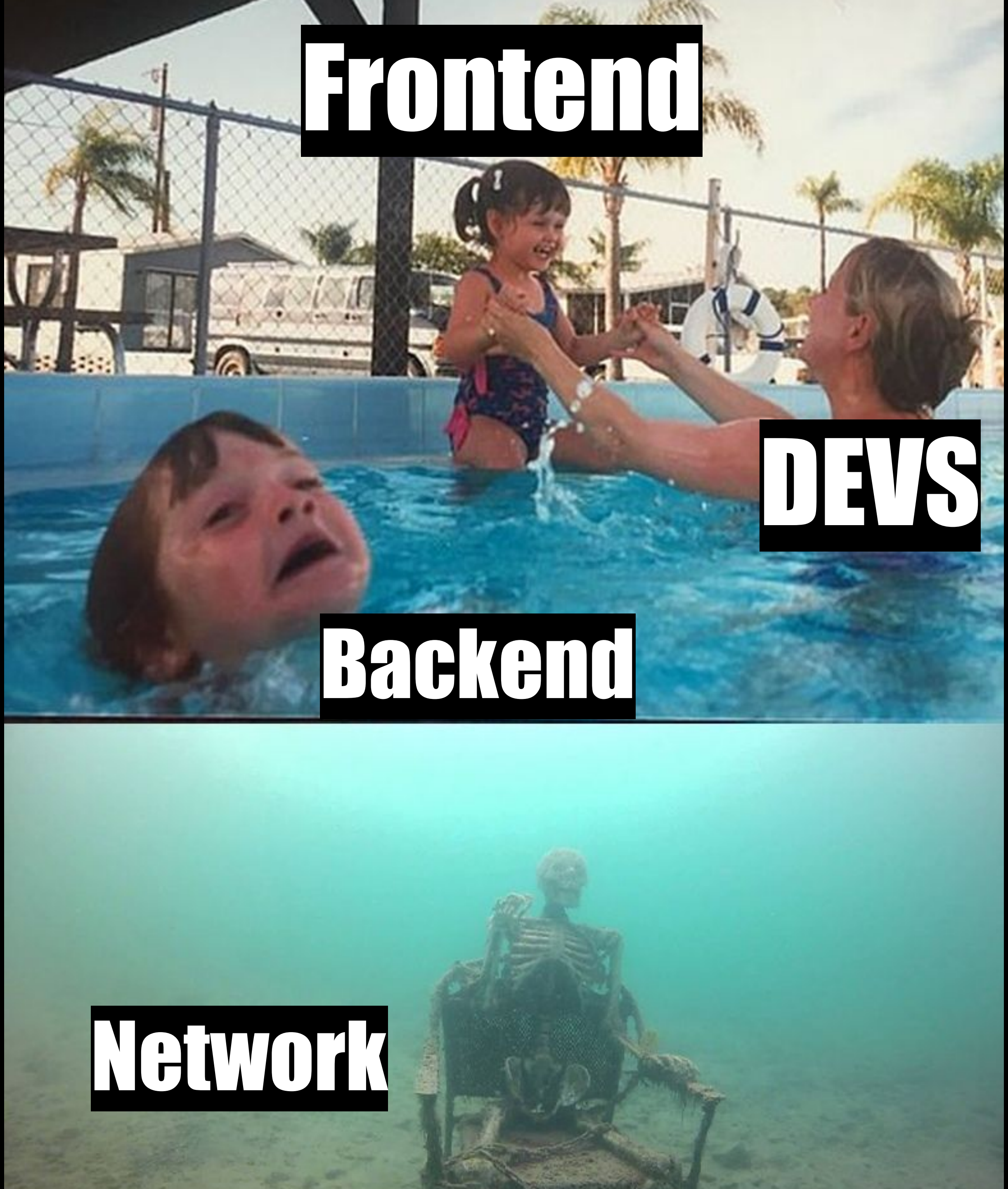
WIRESHARK

???

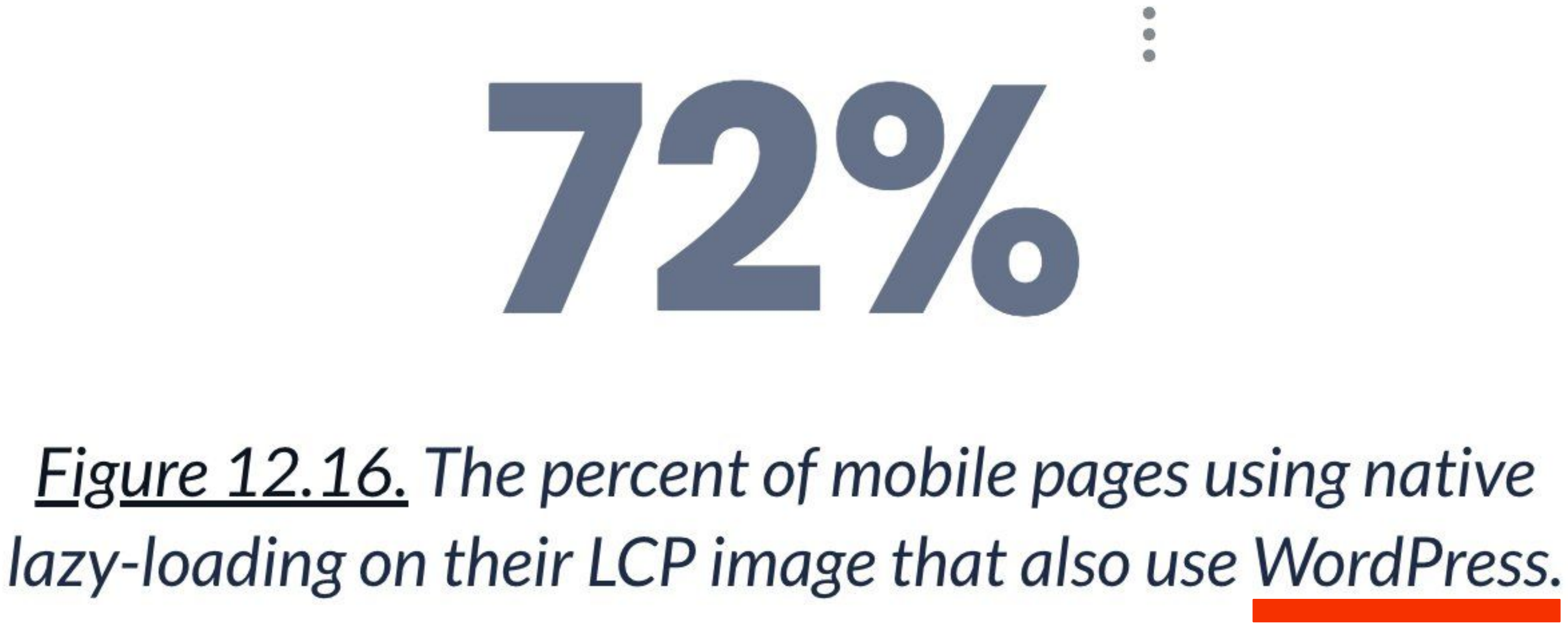
WebPageTest by catchpoint

THIS IS FINE.

The Edge cuts Both Ways


Frontend
DEVS
Backend
Network

The Edge can Cut Deep

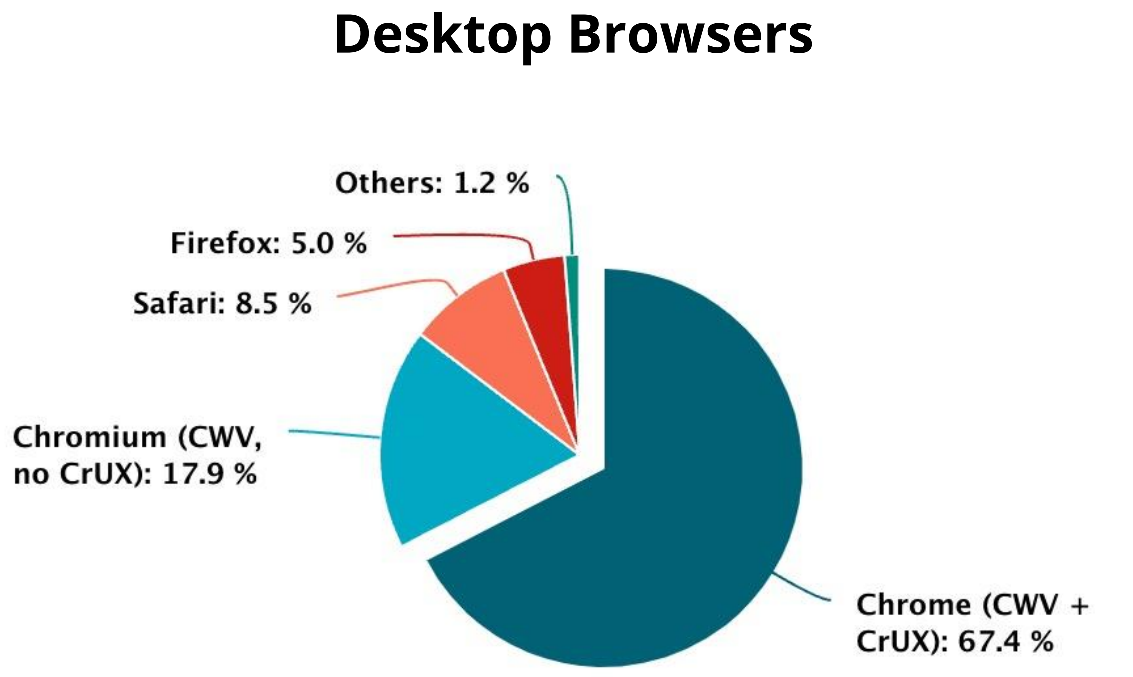


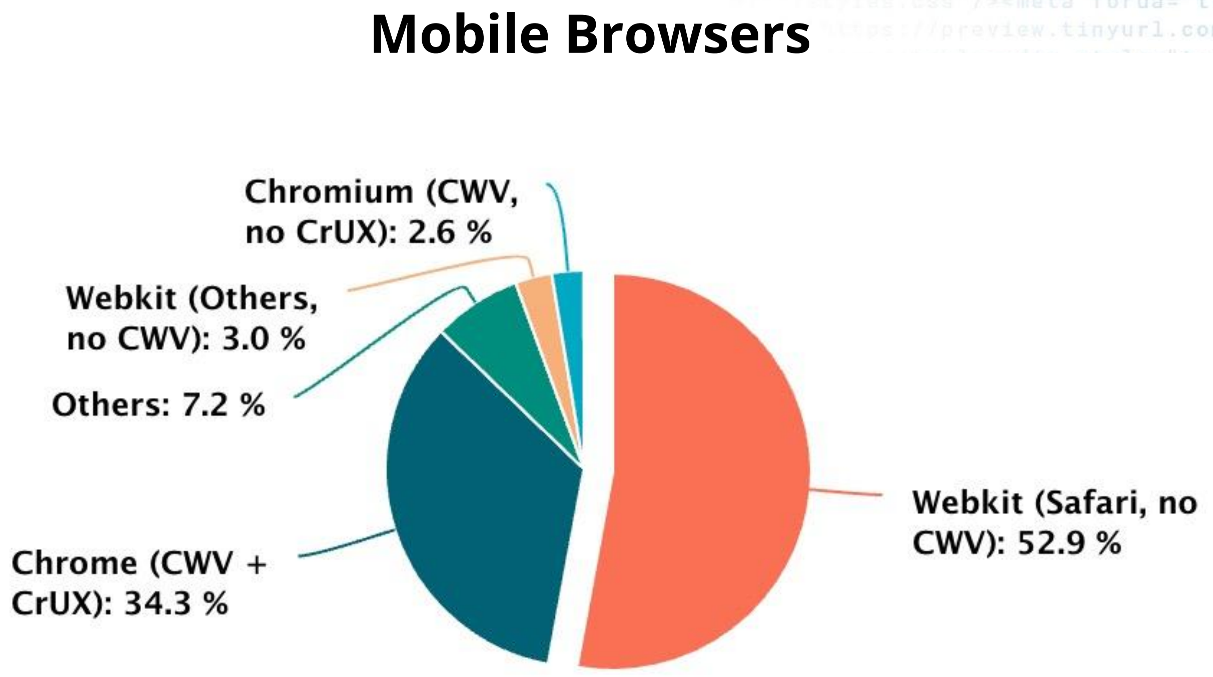
72%

Figure 12.16. The percent of mobile pages using native lazy-loading on their LCP image that also use WordPress.

https://web.dev/articles/lcp-lazy-loading
https://x.com/rick_viscomi/status/1585248419701874688?s=20
https://make.wordpress.org/core/2021/07/15/refining-wordpress-cores-lazy-loading-implementation/

Browser Marketshare (RUM Archive)





65% iOS
VS
35% Android

RUM Archive
www.rumarchive.com

Akamai

Time for a ~~Charge~~ Change

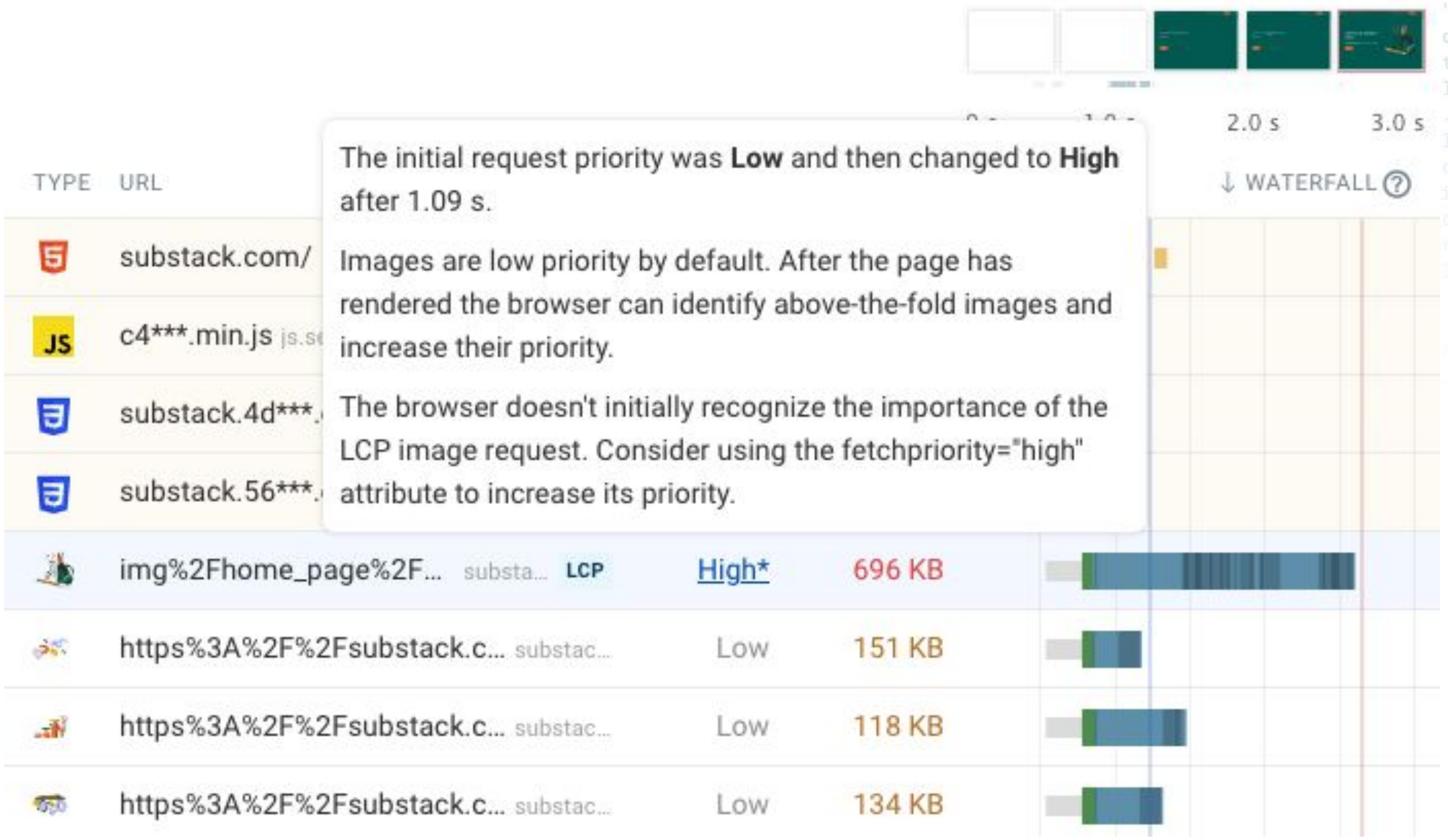
Learn to **recognize** issues

Hold companies **accountable**

Improve **tooling**

Knight in Shining Armour

DebugBear



The initial request priority was **Low** and then changed to **High** after 1.09 s.

Images are low priority by default. After the page has rendered the browser can identify above-the-fold images and increase their priority.

The browser doesn't initially recognize the importance of the LCP image request. Consider using the fetchpriority="high" attribute to increase its priority.

TYPE	URL			
HTML	substack.com/			
JS	c4***.min.js js.se			
CSS	substack.4d***.			
CSS	substack.56***.			
	img%2Fhome_page%2F... substa... LCP	High*	696 KB	
	https%3A%2F%2Fsubstack.c... substac...	Low	151 KB	
	https%3A%2F%2Fsubstack.c... substac...	Low	118 KB	
	https%3A%2F%2Fsubstack.c... substac...	Low	134 KB	

https://www.debugbear.com/blog/request-priorities

VERY EFFICIENT INDEED