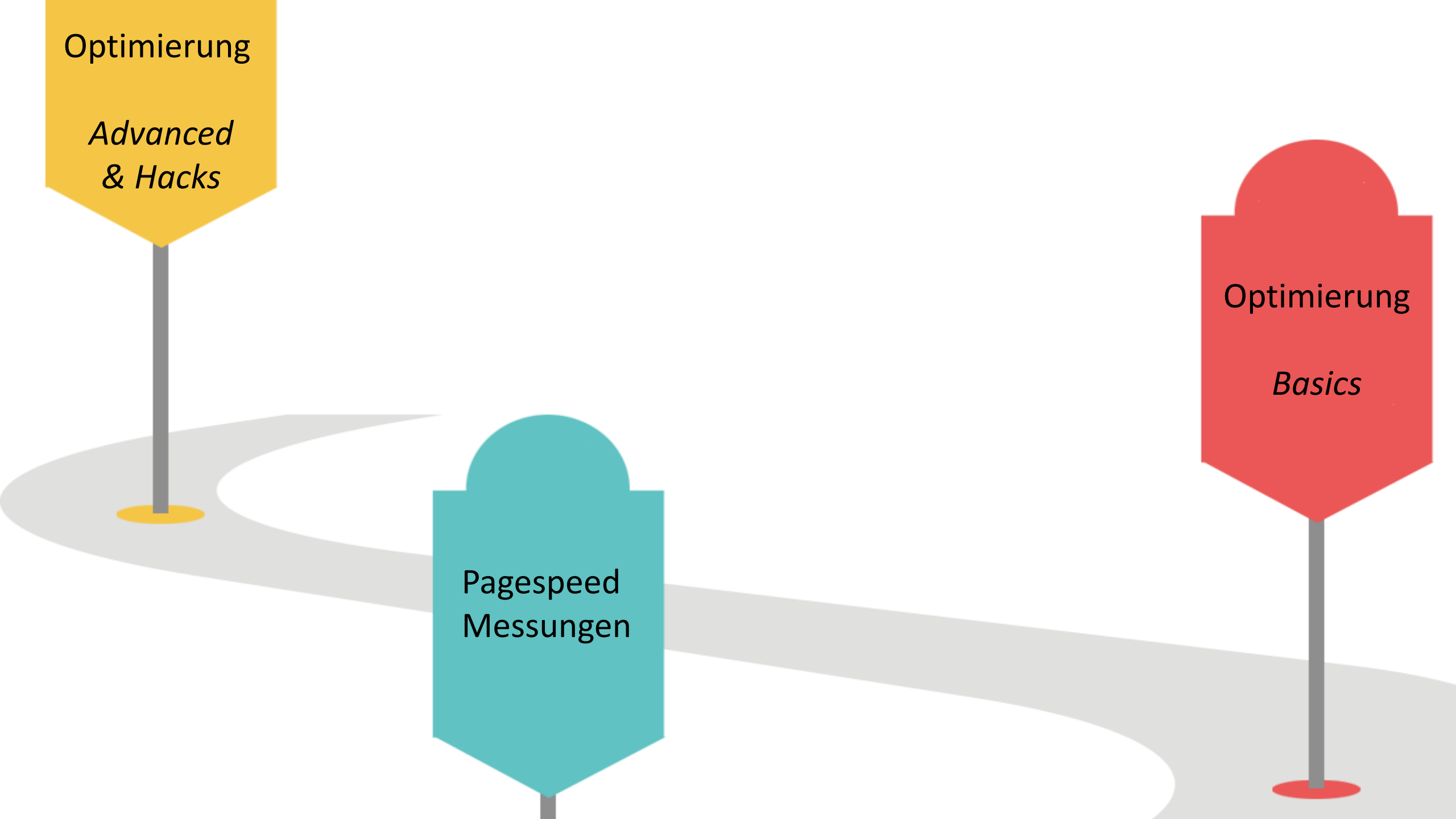
SEO-Atelier
SEO & Online Marketing

Pagespeed Hacks

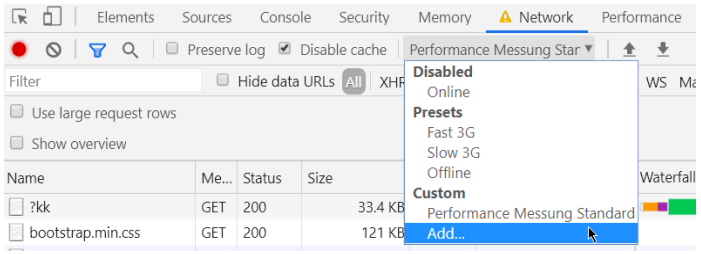

Optimierung
Advanced & Hacks
Pagespeed Messungen
Optimierung
Basics

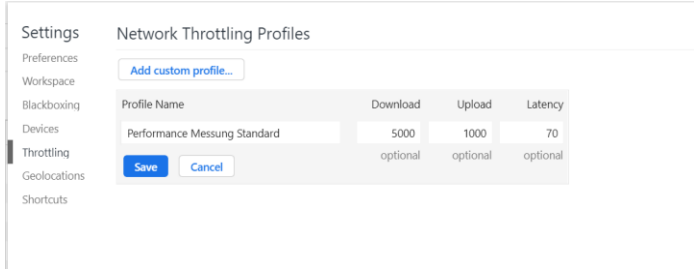
Pagespeed Messungen

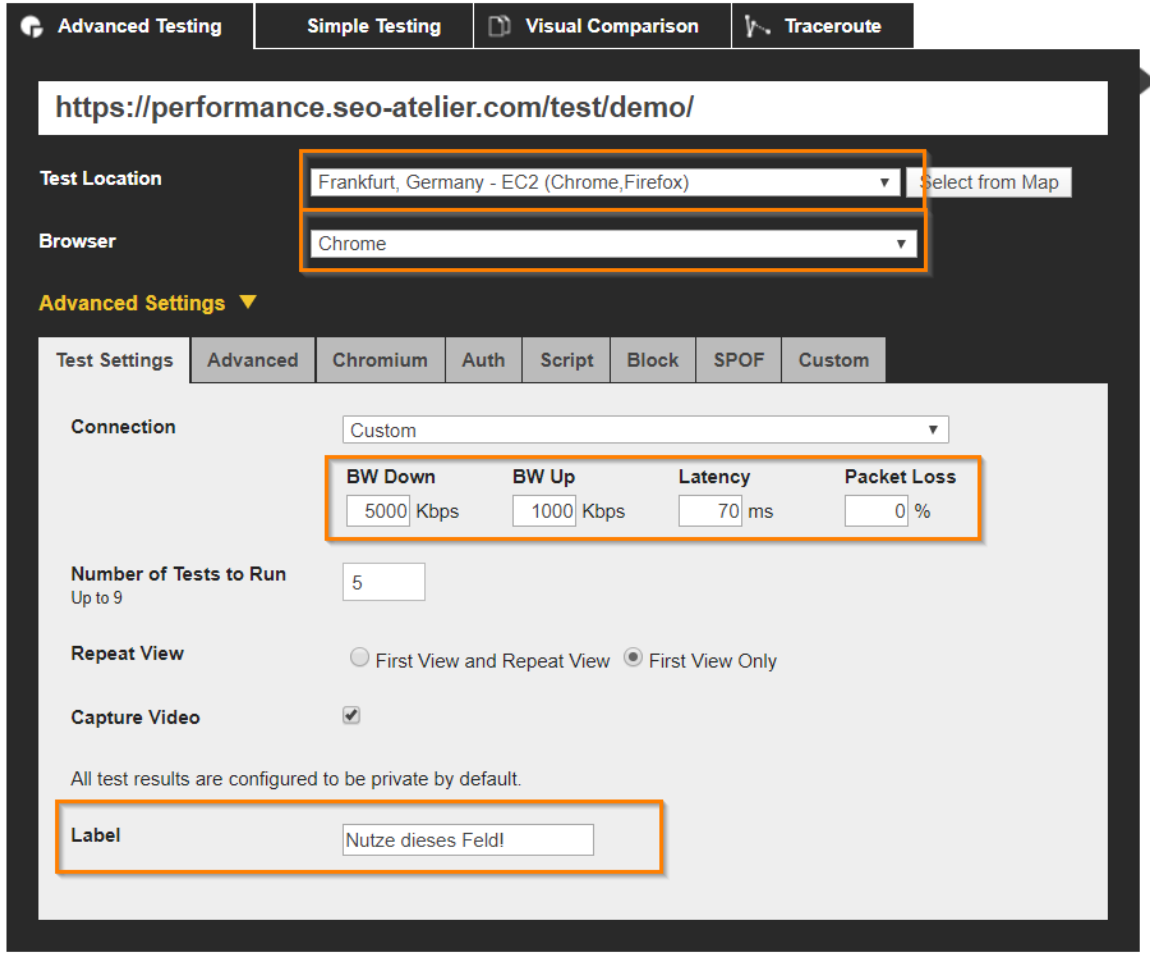
- Werkzeug definieren
- Messkriterien definieren
- KPIs definieren

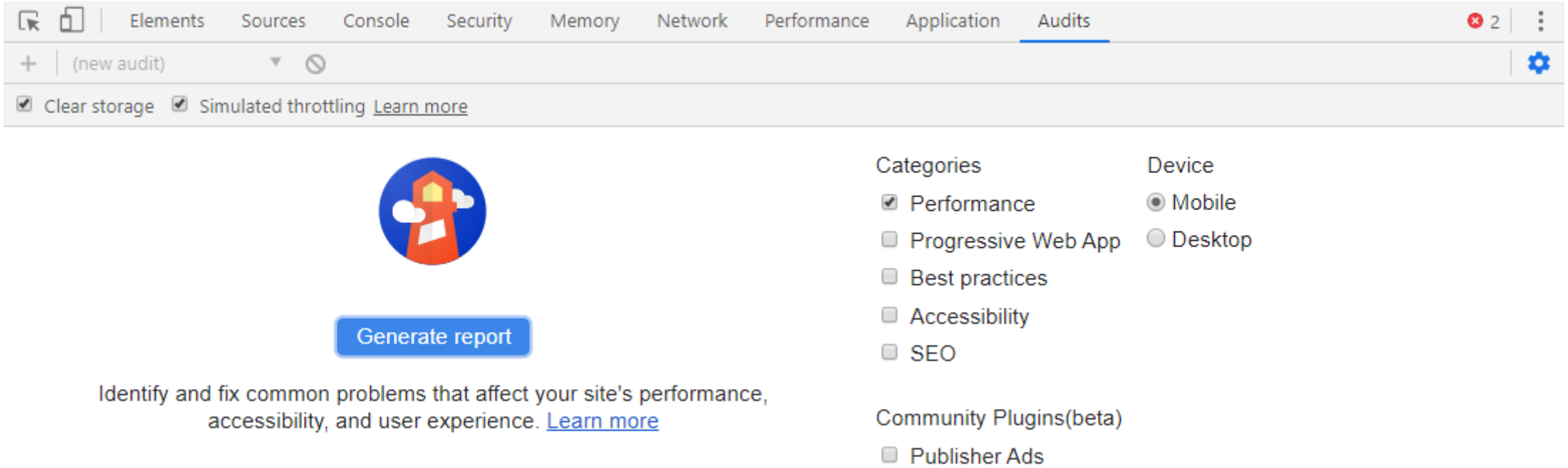
- Webpagetest.org
- Lighthouse
- Google Chrome

Einheitlich messen


Elements
Sources
Console
Security
Memory
Network
Performance
Preserve log
Disable cache
Performance Messung Star
Filter
Hide data URLs
All
XHR
WS
Use large request rows
Show overview
Name
Me...
Status
Size
Waterfall
?kk
GET
200
33.4 KB
bootstrap.min.css
GET
200
121 KB
Disabled
Online
Presets
Fast 3G
Slow 3G
Offline
Custom
Performance Messung Standard
Add...


Settings
Preferences
Workspace
Blackboxing
Devices
Throttling
Geolocations
Shortcuts
Network Throttling Profiles
Add custom profile...
Profile Name
Download
Upload
Latency
Performance Messung Standard
5000
1000
70
optional
optional
optional
Save
Cancel


Advanced Testing
Simple Testing
Visual Comparison
Traceroute
https://performance.seo-atelier.com/test/demo/
Test Location
Frankfurt, Germany - EC2 (Chrome,Firefox)
Select from Map
Browser
Chrome
Advanced Settings
Test Settings
Advanced
Chromium
Auth
Script
Block
SPOF
Custom
Connection
Custom
BW Down
5000 Kbps
BW Up
1000 Kbps
Latency
70 ms
Packet Loss
0 %
Number of Tests to Run
Up to 9
5
Repeat View
First View and Repeat View
First View Only
Capture Video
All test results are configured to be private by default.
Label
Nutze dieses Feld!


Elements
Sources
Console
Security
Memory
Network
Performance
Application
Audits
2
(new audit)
Clear storage
Simulated throttling
Learn more
Generate report
Identify and fix common problems that affect your site's performance, accessibility, and user experience. Learn more
Categories
Performance
Progressive Web App
Best practices
Accessibility
SEO
Device
Mobile
Desktop
Community Plugins(beta)
Publisher Ads

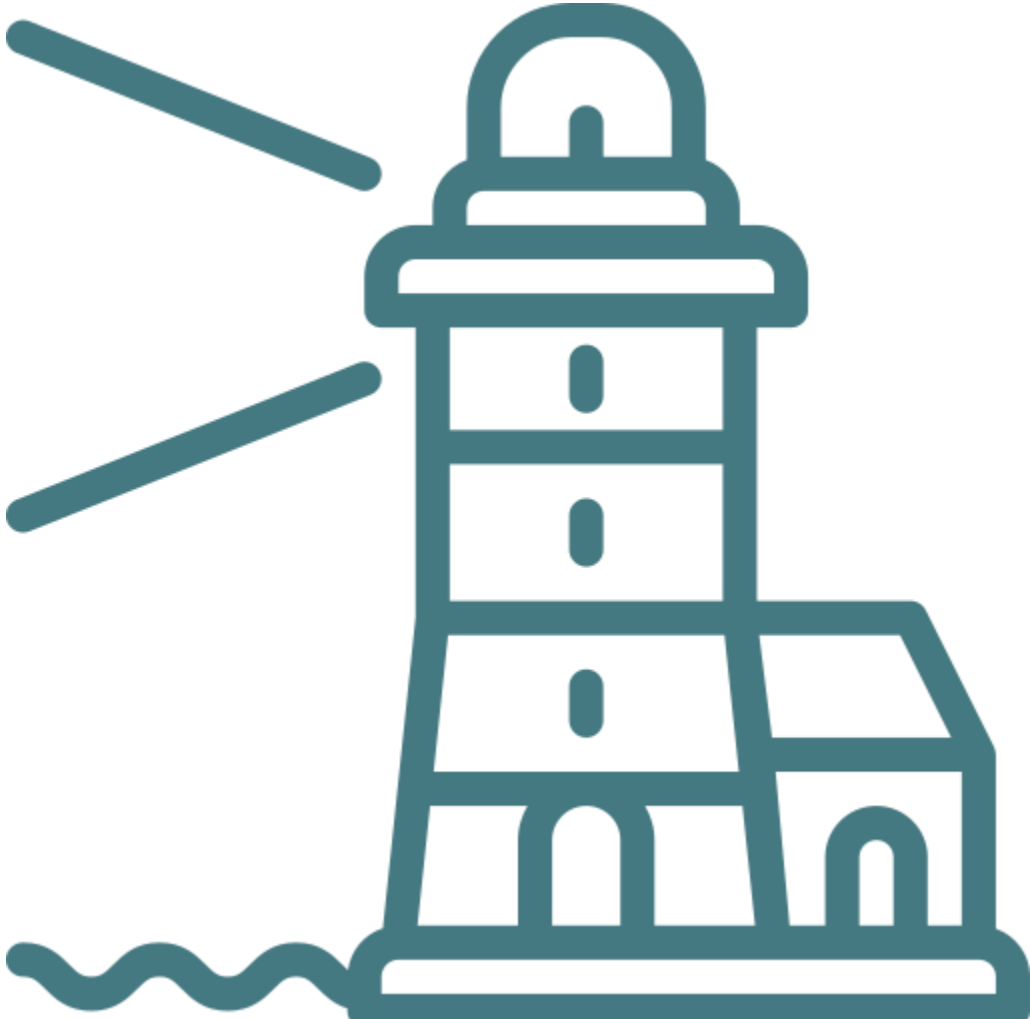
Lighthouse – Tipp: Lighthouse als CL-Tool

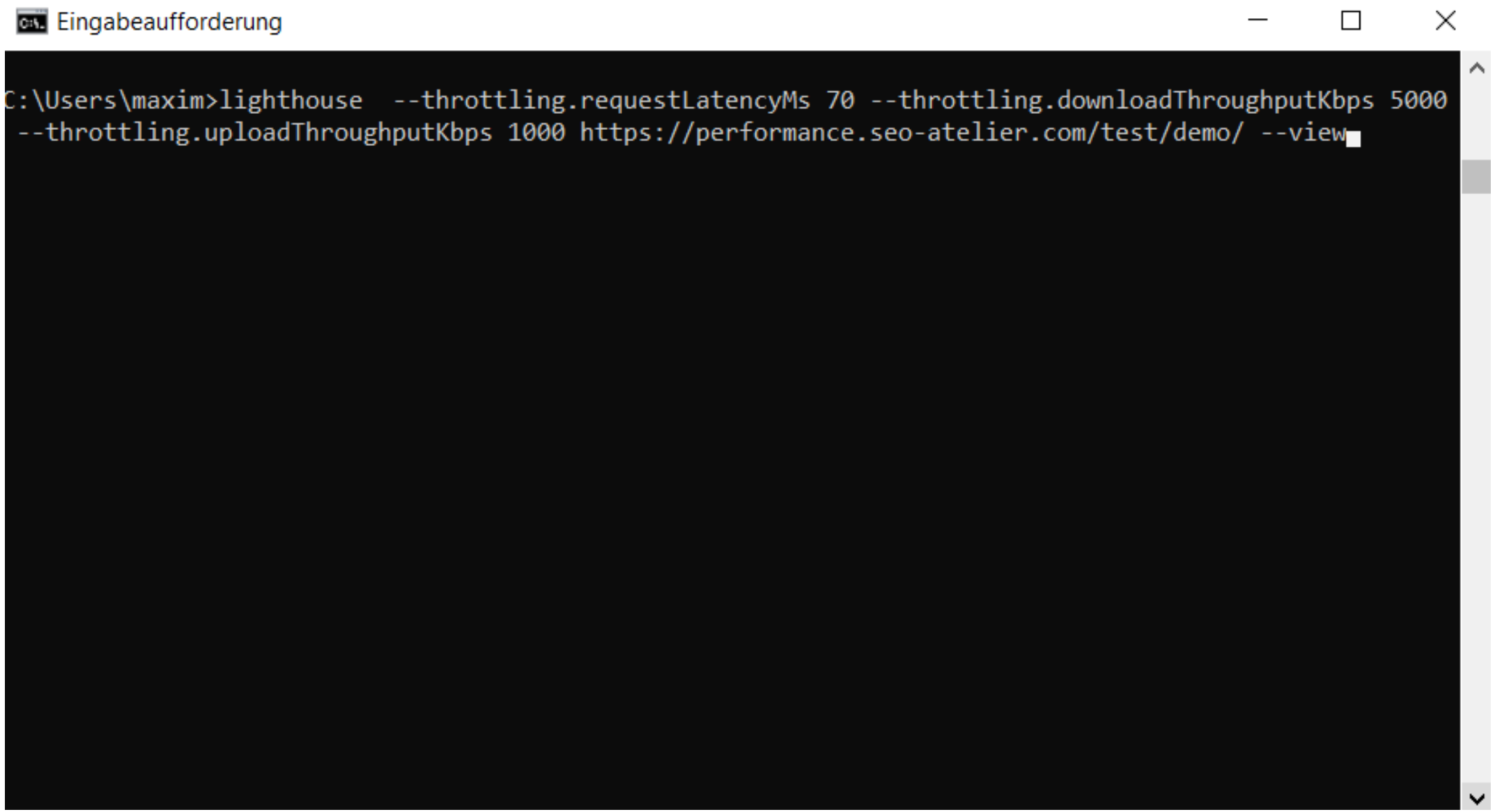
Lighthouse installieren

npm install -g lighthouse-ci

Run mit eigenen Messkriterien möglich:

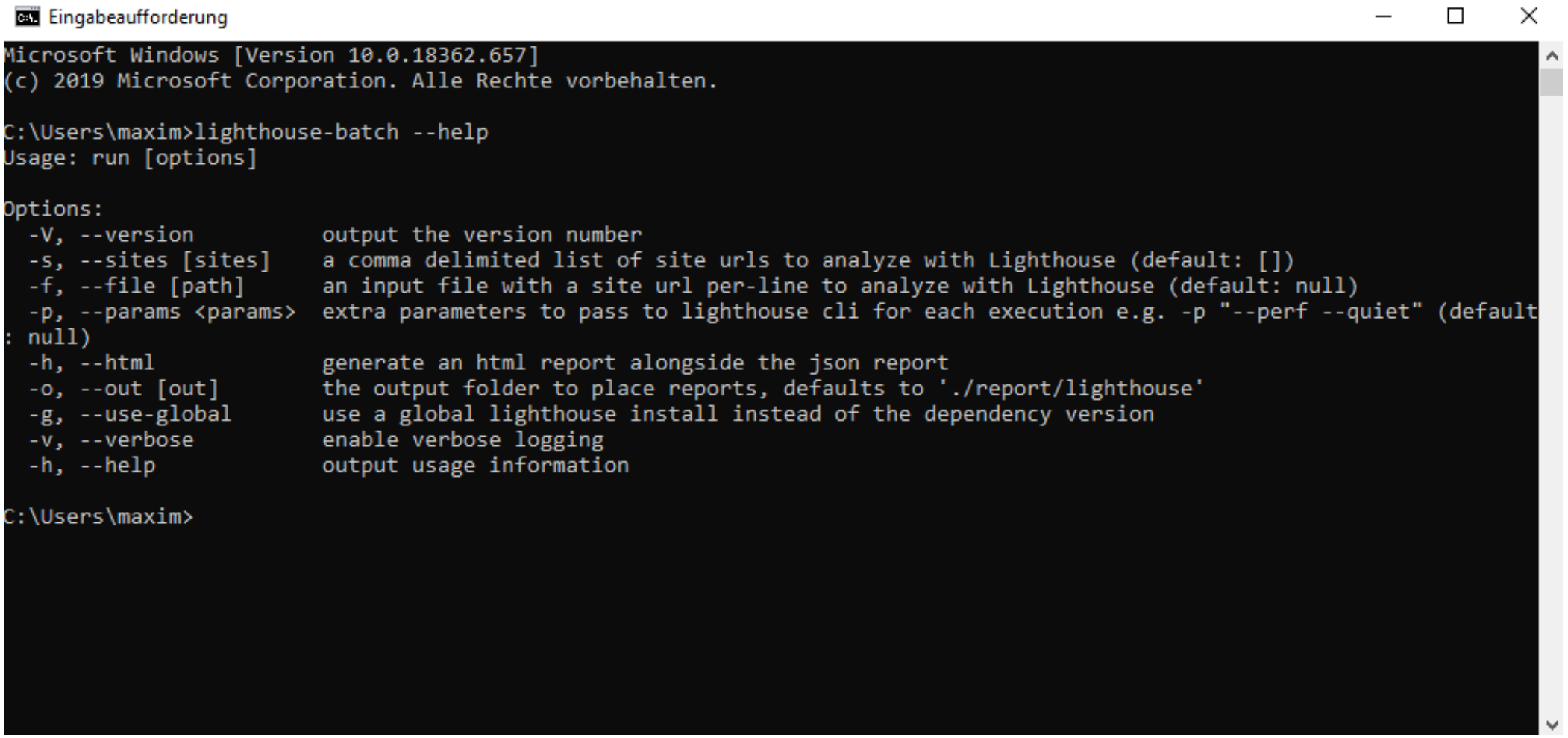
lighthouse --throttling.requestLatencyMs 70
--throttling.downloadThroughputKbps 5000
--throttling.uploadThroughputKbps 1000
<<url>>
--view




Eingabeaufforderung
C:\Users\maxim>lighthouse --throttling.requestLatencyMs 70 --throttling.downloadThroughputKbps 5000
 --throttling.uploadThroughputKbps 1000 https://performance.seo-atelier.com/test/demo/ --view

Lighthouse – Tipp: Lighthouse Batch-Variante

npm install lighthouse-batch -g



Performance Results (Median Run)

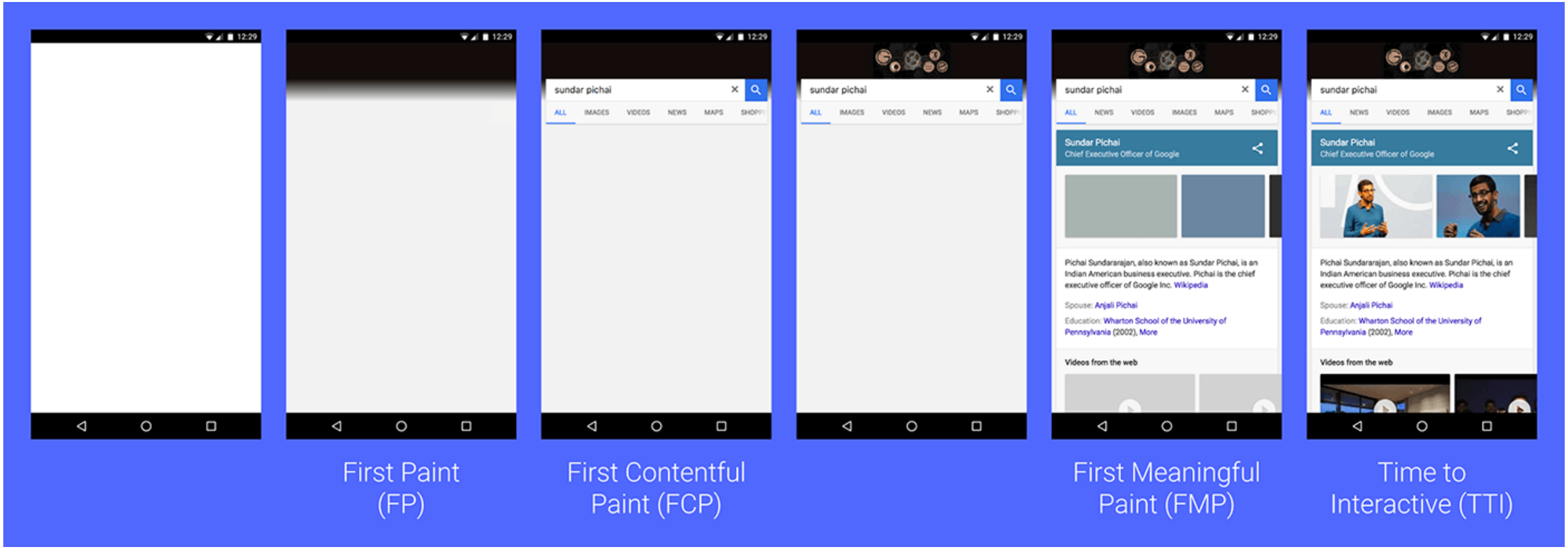
								Document Complete			Fully Loaded			
	Load Time	First Byte	Start Render	First Contentful Paint	Speed Index	Last Painted Hero	First CPU Idle	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View (Run 1)	4.644s	1.419s	3.000s	2.985s	3.265s	3.500s	> 3.000s	4.644s	12	1,620 KB	5.019s	13	1,620 KB	$$$$-
Repeat View (Run 2)	1.533s	1.381s	1.500s	1.521s	1.500s	1.500s	> 1.500s	1.533s	1	2 KB	1.533s	1	2 KB	

Die »First Meaningful Paint« bezeichnet den Zeitpunkt, an dem der Benutzer den wichtigsten Inhalt einer Website als fertig geladen empfindet. *Im Grunde genommen, nachdem der above the fold Inhalt, sowie die Web Fonts geladen wurden.*

Quelle: https://kulturbanause.de/faq/was-ist-der-first-meaningful-paint-fmp/

KPIs

	Metrik
Tut sich was?	First Paint (FP) / First Contentful Paint (FCP)
Ist es nützlich?	First Meaningful Paint (FMP)
Ist es bedienbar?	Time to Interactive (TTI)



Quelle: https://developers.google.com/

WEBPAGETEST

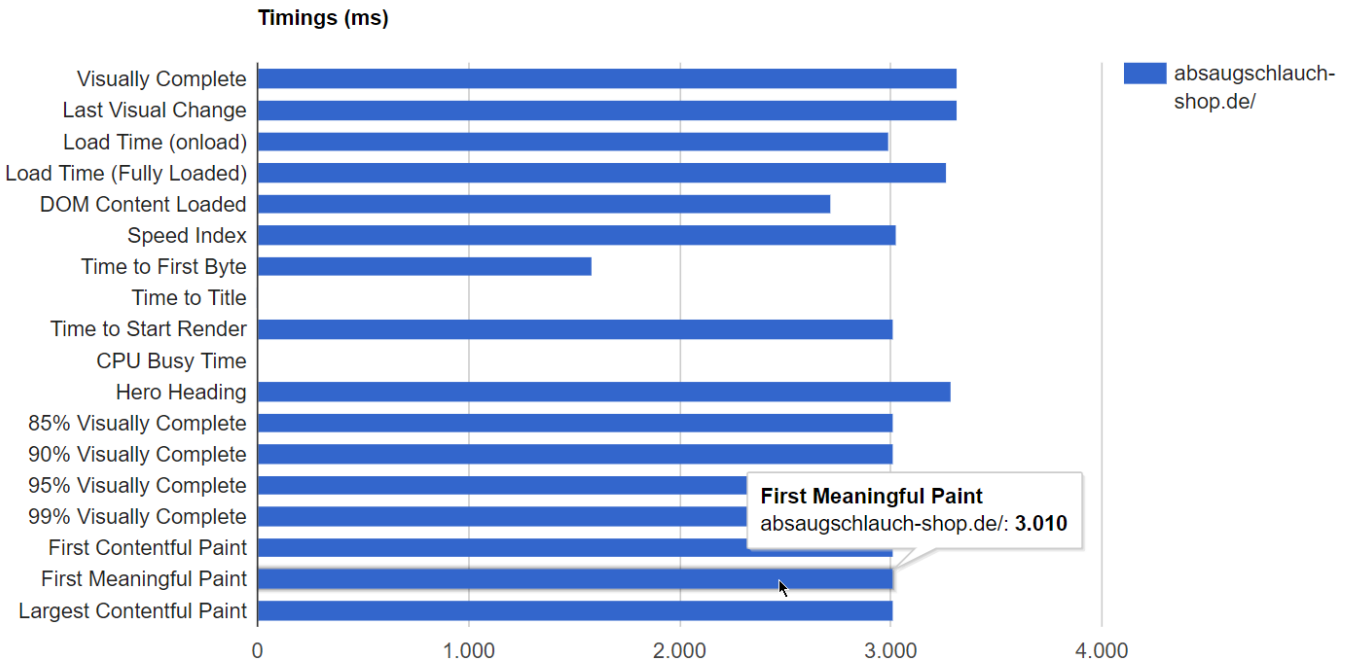
NEW ON-DEMAND WEBINAR:
High Performance Images - Beautiful Shouldn't Mean Slow
VIEW WEBINAR
Akamai FASTER FORWARD

View 1 Day test log for URLs containing [ ] Update List

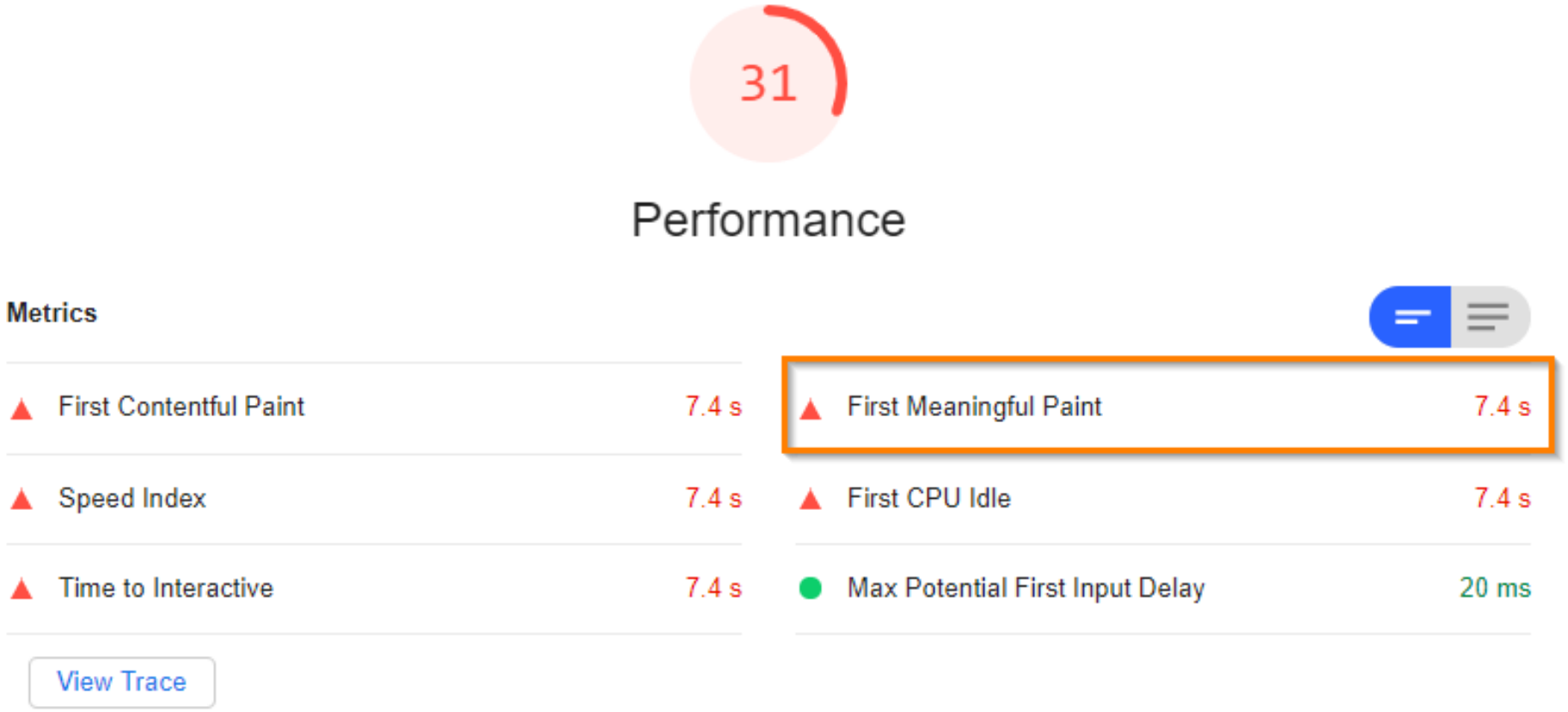
[ ] Only list tests which include video [ ] Show repeat view [ ] Do not limit the number of results (warning: WILL be slow)

Clicking on an URL will bring you to that test's results

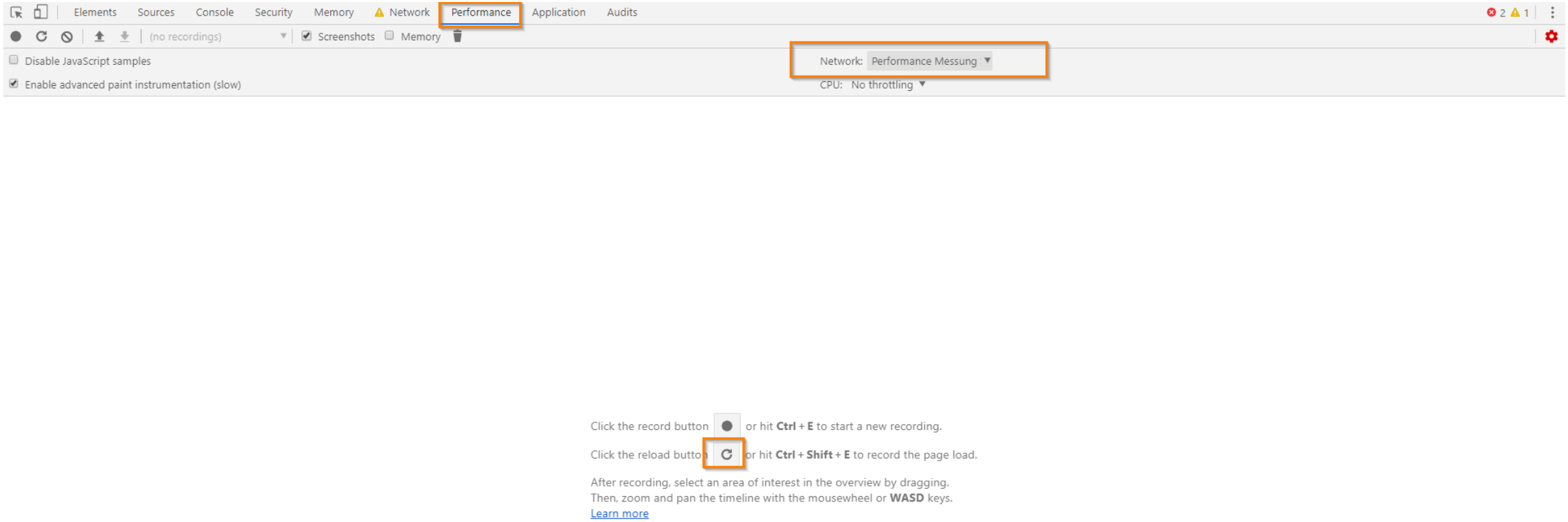
Compare	Date/Time	From	Label	URL
[x]	**02/13/20 23:04:08**	Frankfurt, Germany - EC2 - **Chrome - custom** (video)	GZIP u Min (Edit)	https://performance.seo-atelier.com/test/demo/
[ ]	**02/13/20 21:35:25**	Frankfurt, Germany - EC2 - **Chrome - custom** (video)	Opt. TTFB (Edit)	https://performance.seo-atelier.com/test/demo/
[ ]	**02/13/20 21:20:24**	Frankfurt, Germany - EC2 - **Chrome - custom** (video)	Urspr. Setu	
[ ]	**02/13/20 19:22:11**	Frankfurt, Germany - EC2 - **Chrome - custom** (video)	(Edit)	



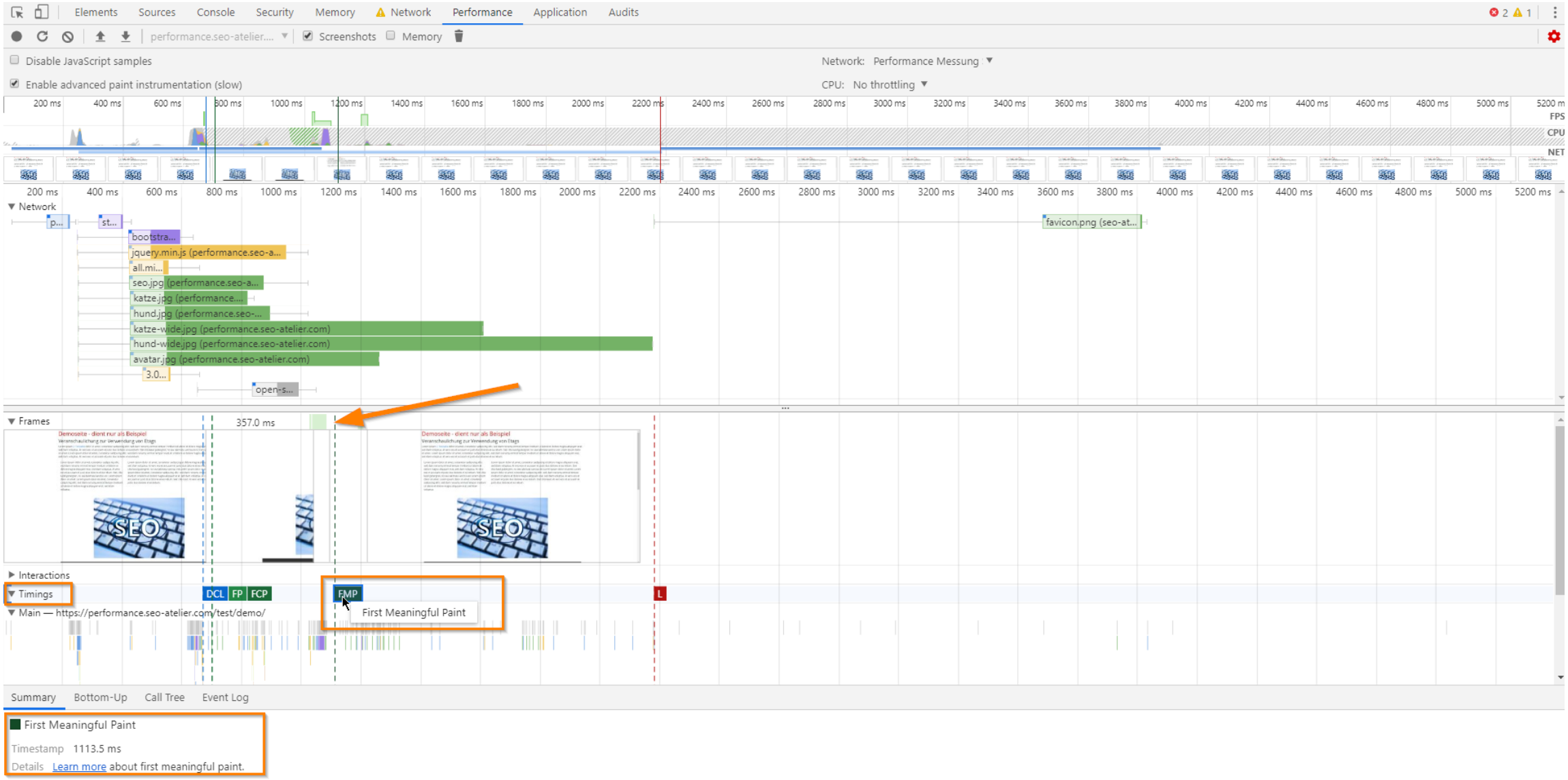
Lighthouse - FMP



Chrome - FMP


Elements
Sources
Console
Security
Memory
Network
Performance
Application
Audits
2 1
(no recordings)
Screenshots
Memory
Disable JavaScript samples
Enable advanced paint instrumentation (slow)
Network: Performance Messung
CPU: No throttling
Click the record button or hit Ctrl + E to start a new recording.
Click the reload button or hit Ctrl + Shift + E to record the page load.
After recording, select an area of interest in the overview by dragging.
Then, zoom and pan the timeline with the mousewheel or WASD keys.
Learn more

Chrome - FMP


Elements
Sources
Console
Security
Memory
Network
Performance
Application
Audits
Screenshots
Memory
Disable JavaScript samples
Enable advanced paint instrumentation (slow)
Network: Performance Messung
CPU: No throttling
FPS
CPU
NET
Network
favicon.png (seo-at...
bootstra...
jquery.min.js (performance.seo-a...
all.mi...
seo.jpg (performance.seo-a...
katze.jpg (performance....
hund.jpg (performance.seo-...
katze-wide.jpg (performance.seo-atelier.com)
hund-wide.jpg (performance.seo-atelier.com)
avatar.jpg (performance.seo-atelier.com)
open-s...
Frames
357.0 ms
Interactions
Timings
DCL
FP
FCP
FMP
First Meaningful Paint
L
Main — https://performance.seo-atelier.com/test/demo/
Summary
Bottom-Up
Call Tree
Event Log
First Meaningful Paint
Timestamp 1113.5 ms
Details Learn more about first meaningful paint.

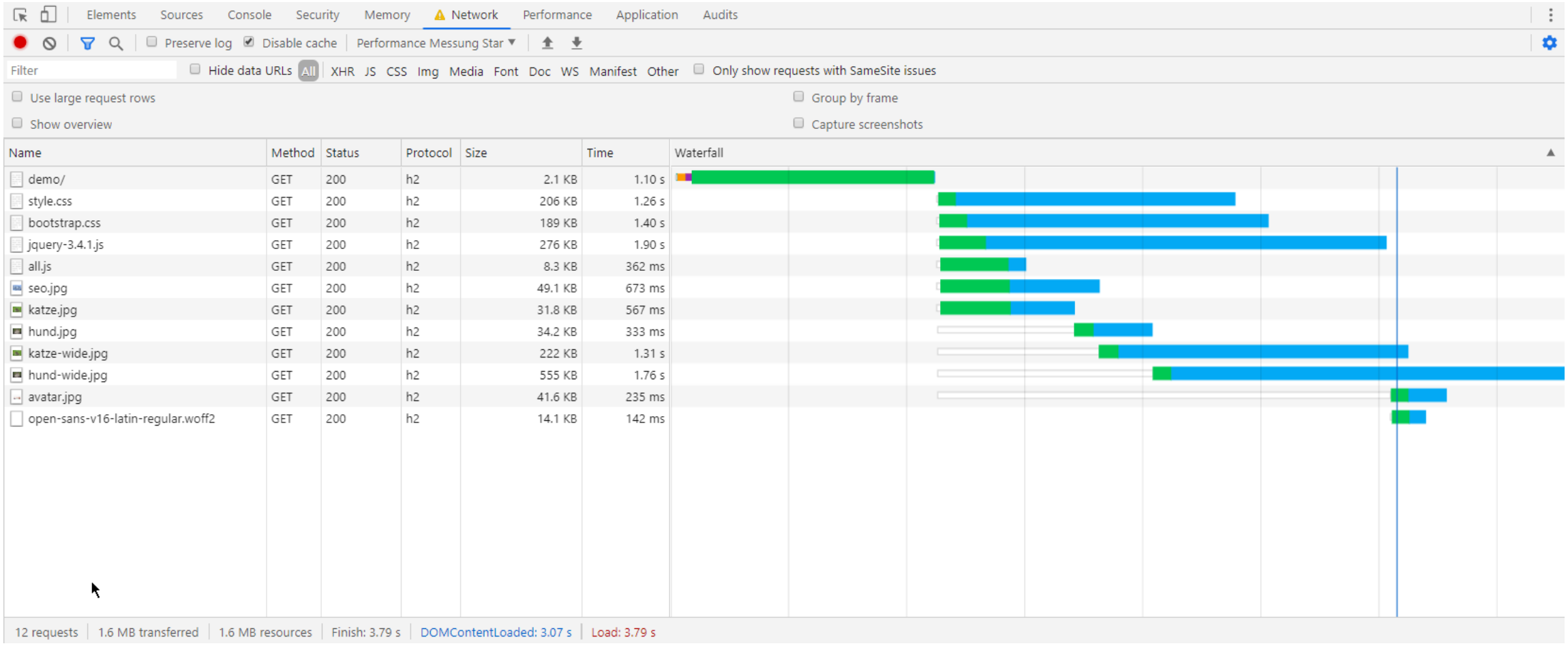
Optimierung

Basis

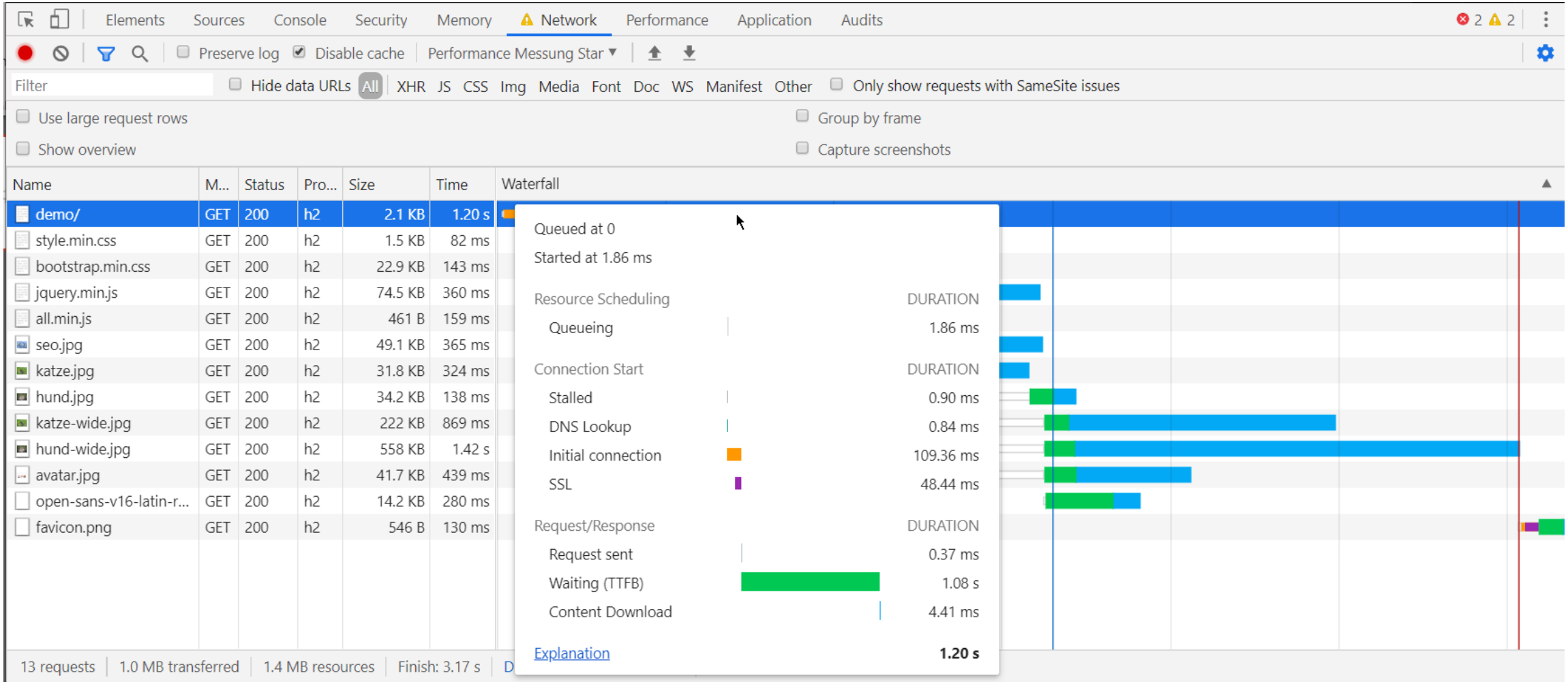
- Ordentliche Response-Time
- Browser-Caching
- Dateigröße

- **Ordentliche Response-Time**
  Browser-Caching
  Dateigröße

TTFB


Elements
Sources
Console
Security
Memory
Network
Performance
Application
Audits
Preserve log
Disable cache
Performance Messung Star
Filter
Hide data URLs
All
XHR
JS
CSS
Img
Media
Font
Doc
WS
Manifest
Other
Only show requests with SameSite issues
Use large request rows
Group by frame
Show overview
Capture screenshots
Name
Method
Status
Protocol
Size
Time
Waterfall
demo/ GET 200 h2 2.1 KB 1.10 s
style.css GET 200 h2 206 KB 1.26 s
bootstrap.css GET 200 h2 189 KB 1.40 s
jquery-3.4.1.js GET 200 h2 276 KB 1.90 s
all.js GET 200 h2 8.3 KB 362 ms
seo.jpg GET 200 h2 49.1 KB 673 ms
katze.jpg GET 200 h2 31.8 KB 567 ms
hund.jpg GET 200 h2 34.2 KB 333 ms
katze-wide.jpg GET 200 h2 222 KB 1.31 s
hund-wide.jpg GET 200 h2 555 KB 1.76 s
avatar.jpg GET 200 h2 41.6 KB 235 ms
open-sans-v16-latin-regular.woff2 GET 200 h2 14.1 KB 142 ms
12 requests
1.6 MB transferred
1.6 MB resources
Finish: 3.79 s
DOMContentLoaded: 3.07 s
Load: 3.79 s

TTFB


Elements
Sources
Console
Security
Memory
Network
Performance
Application
Audits
Preserve log
Disable cache
Performance Messung Star
Filter
Hide data URLs
All
XHR
JS
CSS
Img
Media
Font
Doc
WS
Manifest
Other
Only show requests with SameSite issues
Use large request rows
Group by frame
Show overview
Capture screenshots
Name
M...
Status
Pro...
Size
Time
Waterfall
demo/ GET 200 h2 2.1 KB 1.20 s
style.min.css GET 200 h2 1.5 KB 82 ms
bootstrap.min.css GET 200 h2 22.9 KB 143 ms
jquery.min.js GET 200 h2 74.5 KB 360 ms
all.min.js GET 200 h2 461 B 159 ms
seo.jpg GET 200 h2 49.1 KB 365 ms
katze.jpg GET 200 h2 31.8 KB 324 ms
hund.jpg GET 200 h2 34.2 KB 138 ms
katze-wide.jpg GET 200 h2 222 KB 869 ms
hund-wide.jpg GET 200 h2 558 KB 1.42 s
avatar.jpg GET 200 h2 41.7 KB 439 ms
open-sans-v16-latin-r... GET 200 h2 14.2 KB 280 ms
favicon.png GET 200 h2 546 B 130 ms
Queued at 0
Started at 1.86 ms
Resource Scheduling
DURATION
Queueing 1.86 ms
Connection Start
DURATION
Stalled 0.90 ms
DNS Lookup 0.84 ms
Initial connection 109.36 ms
SSL 48.44 ms
Request/Response
DURATION
Request sent 0.37 ms
Waiting (TTFB) 1.08 s
Content Download 4.41 ms
Explanation
1.20 s
13 requests
1.0 MB transferred
1.4 MB resources
Finish: 3.17 s

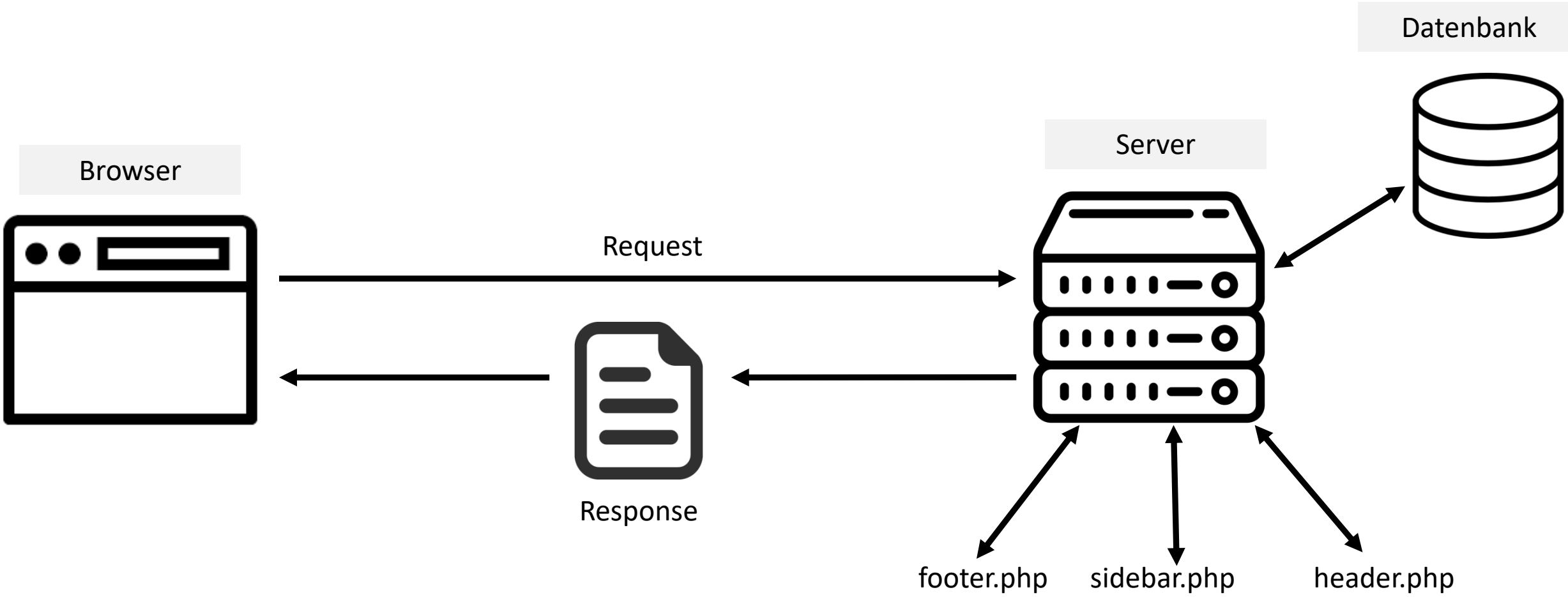
Performance Results (Median Run)

	Load Time	First Byte	Start Render	First Contentful Paint	Speed Index	Last Painted Hero	First CPU Idle	Document Complete			Fully Loaded			
								Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View (Run 1)	4.644s	1.419s	3.000s	2.985s	3.265s	3.500s	> 3.000s	4.644s	12	1,620 KB	5.019s	13	1,620 KB	$$$$-
Repeat View (Run 2)	1.533s	1.381s	1.500s	1.521s	1.500s	1.500s	> 1.500s	1.533s	1	2 KB	1.533s	1	2 KB	

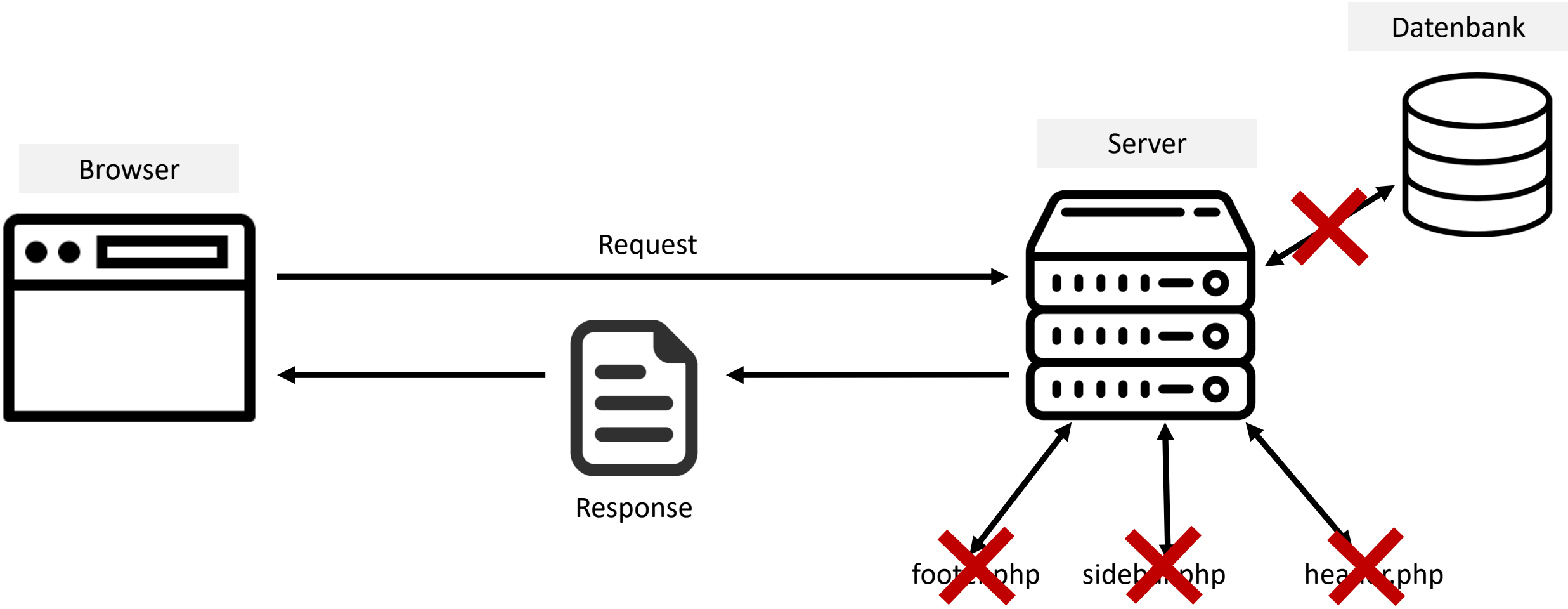
▲ Reduce server response times (TTFB) — 0.43 s

Time To First Byte identifies the time at which your server sends a response. Learn more.

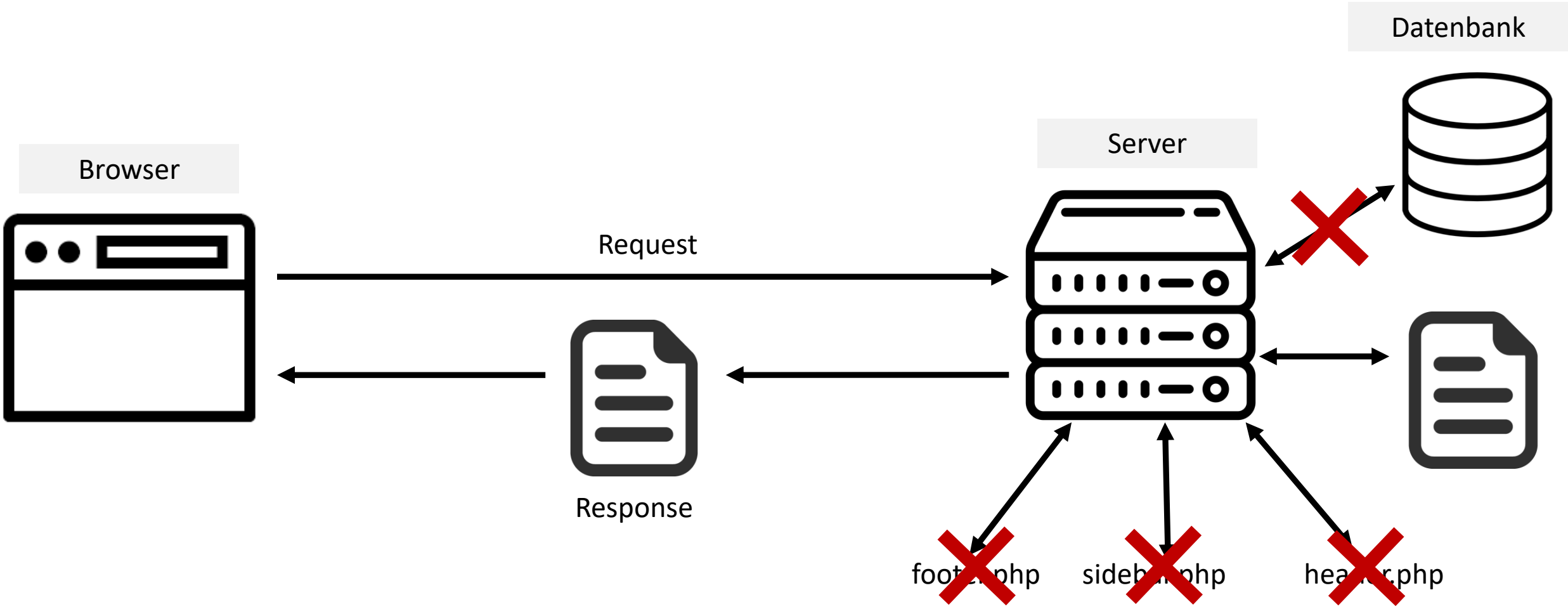
TTFB - Warten auf den Server…



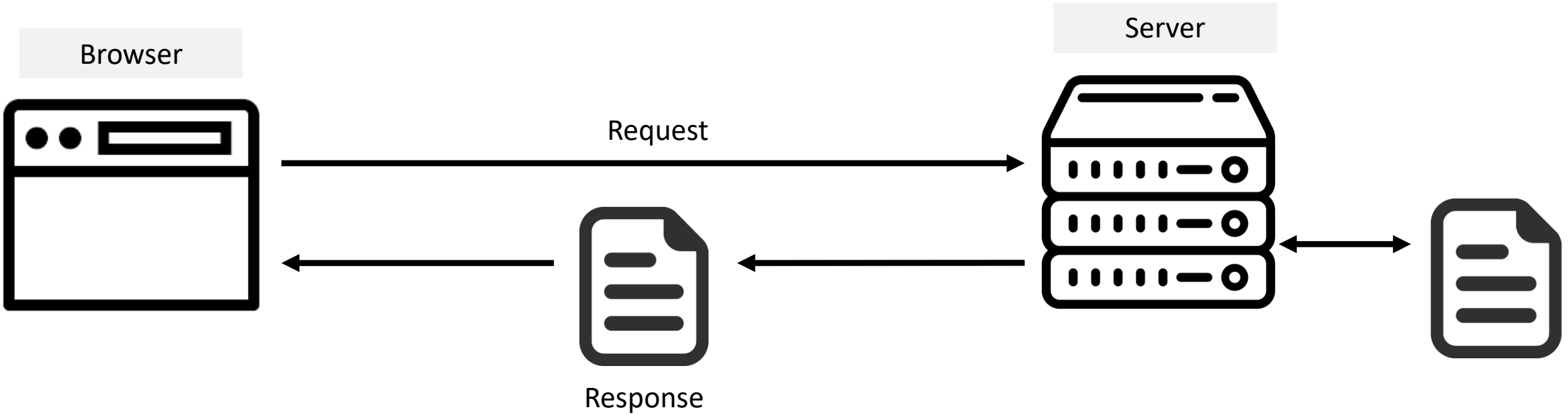
TTFB - Warten auf den Server…



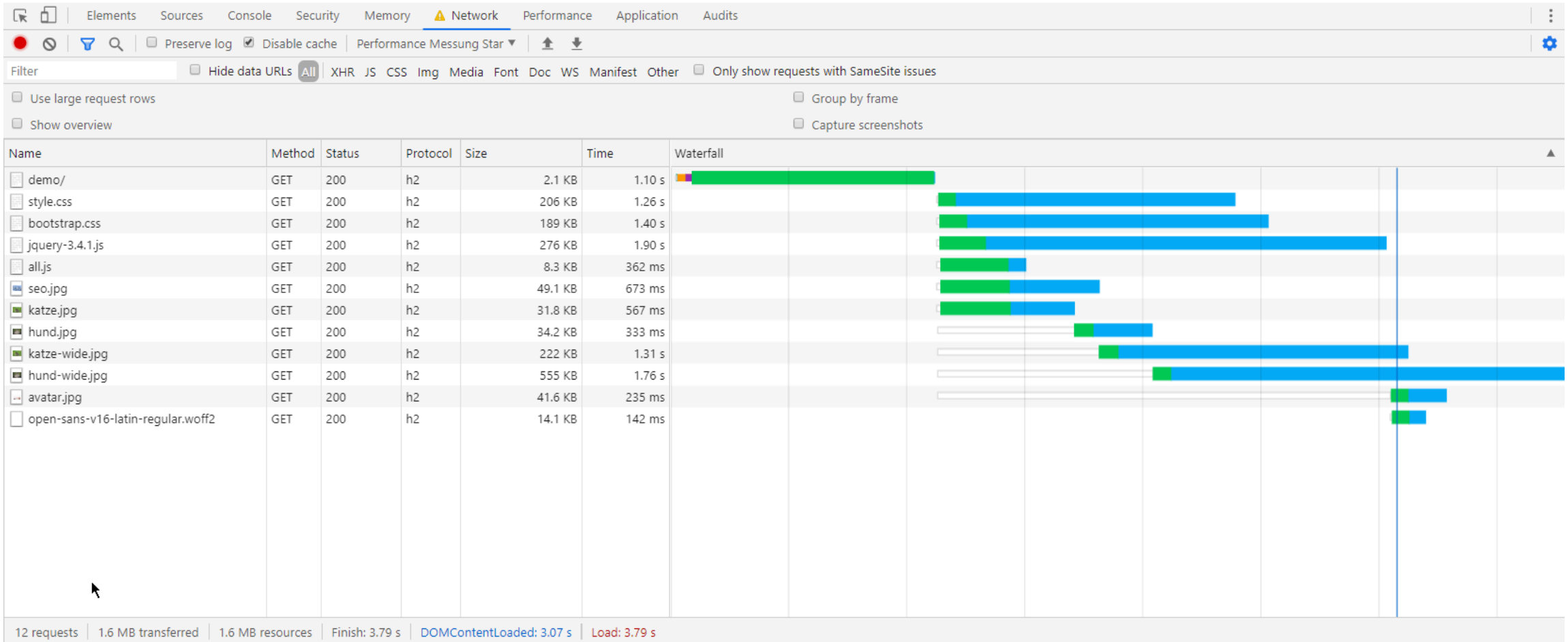
TTFB - Warten auf den Server…

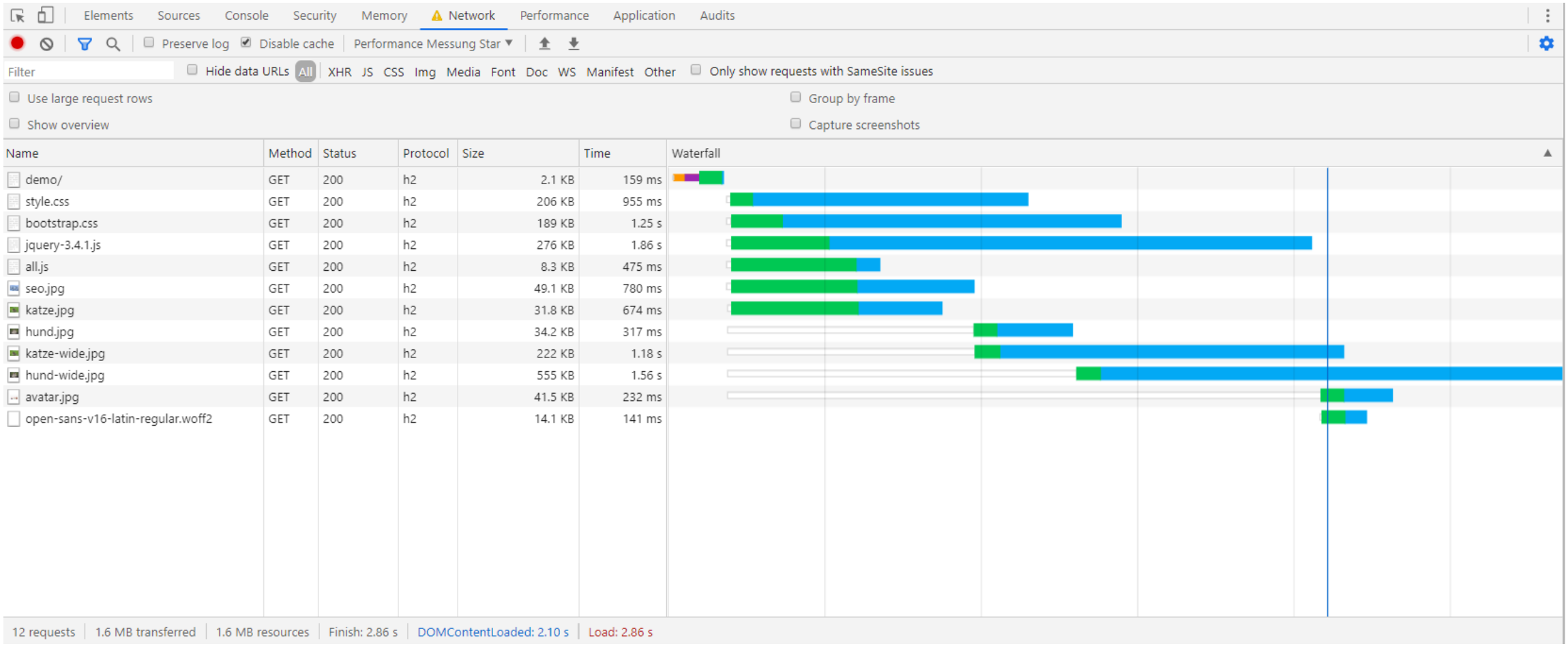


TTFB - Warten auf den Server…



- Plugin-Caches
- Serverseitige Caches
- CDN


Elements
Sources
Console
Security
Memory
Network
Performance
Application
Audits
Preserve log
Disable cache
Performance Messung Star
Filter
Hide data URLs
All
XHR
JS
CSS
Img
Media
Font
Doc
WS
Manifest
Other
Only show requests with SameSite issues
Use large request rows
Group by frame
Show overview
Capture screenshots
Name
Method
Status
Protocol
Size
Time
Waterfall
demo/ GET 200 h2 2.1 KB 1.10 s
style.css GET 200 h2 206 KB 1.26 s
bootstrap.css GET 200 h2 189 KB 1.40 s
jquery-3.4.1.js GET 200 h2 276 KB 1.90 s
all.js GET 200 h2 8.3 KB 362 ms
seo.jpg GET 200 h2 49.1 KB 673 ms
katze.jpg GET 200 h2 31.8 KB 567 ms
hund.jpg GET 200 h2 34.2 KB 333 ms
katze-wide.jpg GET 200 h2 222 KB 1.31 s
hund-wide.jpg GET 200 h2 555 KB 1.76 s
avatar.jpg GET 200 h2 41.6 KB 235 ms
open-sans-v16-latin-regular.woff2 GET 200 h2 14.1 KB 142 ms
12 requests
1.6 MB transferred
1.6 MB resources
Finish: 3.79 s
DOMContentLoaded: 3.07 s
Load: 3.79 s


Elements
Sources
Console
Security
Memory
Network
Performance
Application
Audits
Preserve log
Disable cache
Performance Messung Star
Filter
Hide data URLs
All
XHR
JS
CSS
Img
Media
Font
Doc
WS
Manifest
Other
Only show requests with SameSite issues
Use large request rows
Group by frame
Show overview
Capture screenshots
Name
Method
Status
Protocol
Size
Time
Waterfall
demo/ GET 200 h2 2.1 KB 159 ms
style.css GET 200 h2 206 KB 955 ms
bootstrap.css GET 200 h2 189 KB 1.25 s
jquery-3.4.1.js GET 200 h2 276 KB 1.86 s
all.js GET 200 h2 8.3 KB 475 ms
seo.jpg GET 200 h2 49.1 KB 780 ms
katze.jpg GET 200 h2 31.8 KB 674 ms
hund.jpg GET 200 h2 34.2 KB 317 ms
katze-wide.jpg GET 200 h2 222 KB 1.18 s
hund-wide.jpg GET 200 h2 555 KB 1.56 s
avatar.jpg GET 200 h2 41.5 KB 232 ms
open-sans-v16-latin-regular.woff2 GET 200 h2 14.1 KB 141 ms
12 requests
1.6 MB transferred
1.6 MB resources
Finish: 2.86 s
DOMContentLoaded: 2.10 s
Load: 2.86 s

Performance Results (Median Run)

								Document Complete			Fully Loaded			
	Load Time	First Byte	Start Render	First Contentful Paint	Speed Index	Last Painted Hero	First CPU Idle	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View (Run 1)	3.647s	0.409s	2.000s	1.984s	2.338s	2.600s	> 2.000s	3.647s	12	1,620 KB	4.000s	13	1,620 KB	$$$$-
Repeat View (Run 3)	0.541s	0.386s	0.600s	0.535s	0.600s	0.600s	> 0.600s	0.541s	1	2 KB	0.541s	1	2 KB	

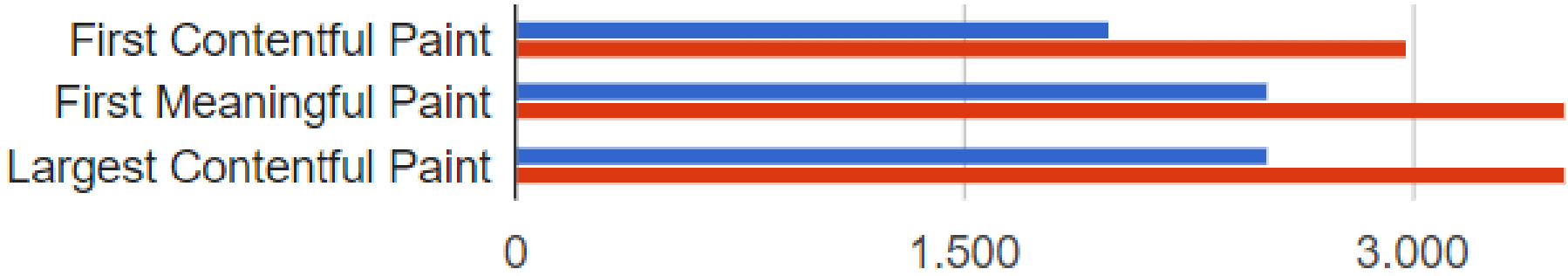
HOME TEST RESULT TEST HISTORY FORUMS DOCUMENTATION ABOUT

View 1 Day test log for URLs containing Update List

Only list tests which include video Show repeat view Do not limit the number of results (warning: WILL be slow)

Clicking on an URL will bring you to that test's results

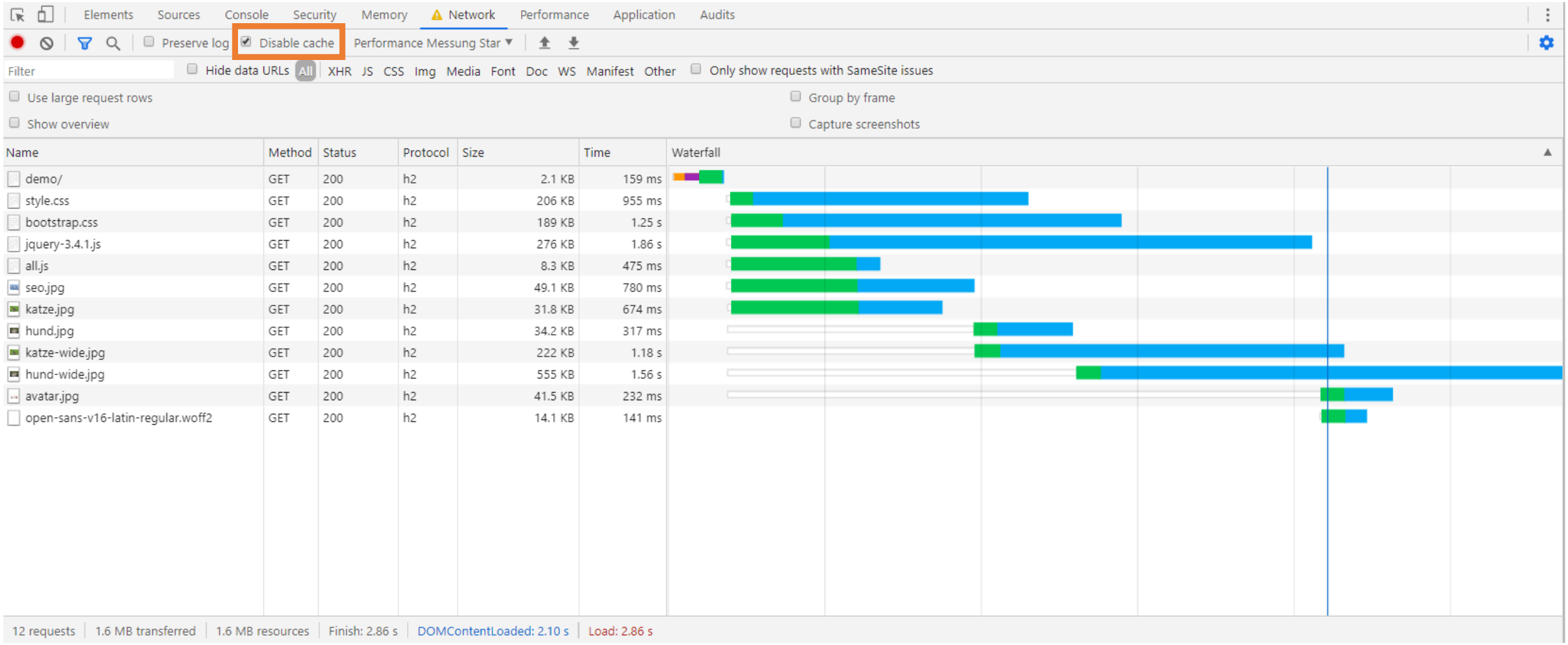
Compare	Date/Time	From	Label	URL
☑	**02/13/20 21:35:25**	Frankfurt, Germany - EC2 - **Chrome** - **custom** (video)	Opt. TTFB (Edit)	https://performance.seo-atelier.com/test/demo/
☑	**02/13/20 21:20:24**	Frankfurt, Germany - EC2 - **Chrome** - **custom** (video)	Urspr. Setup (Edit)	https://performance.seo-atelier.com/test/demo/



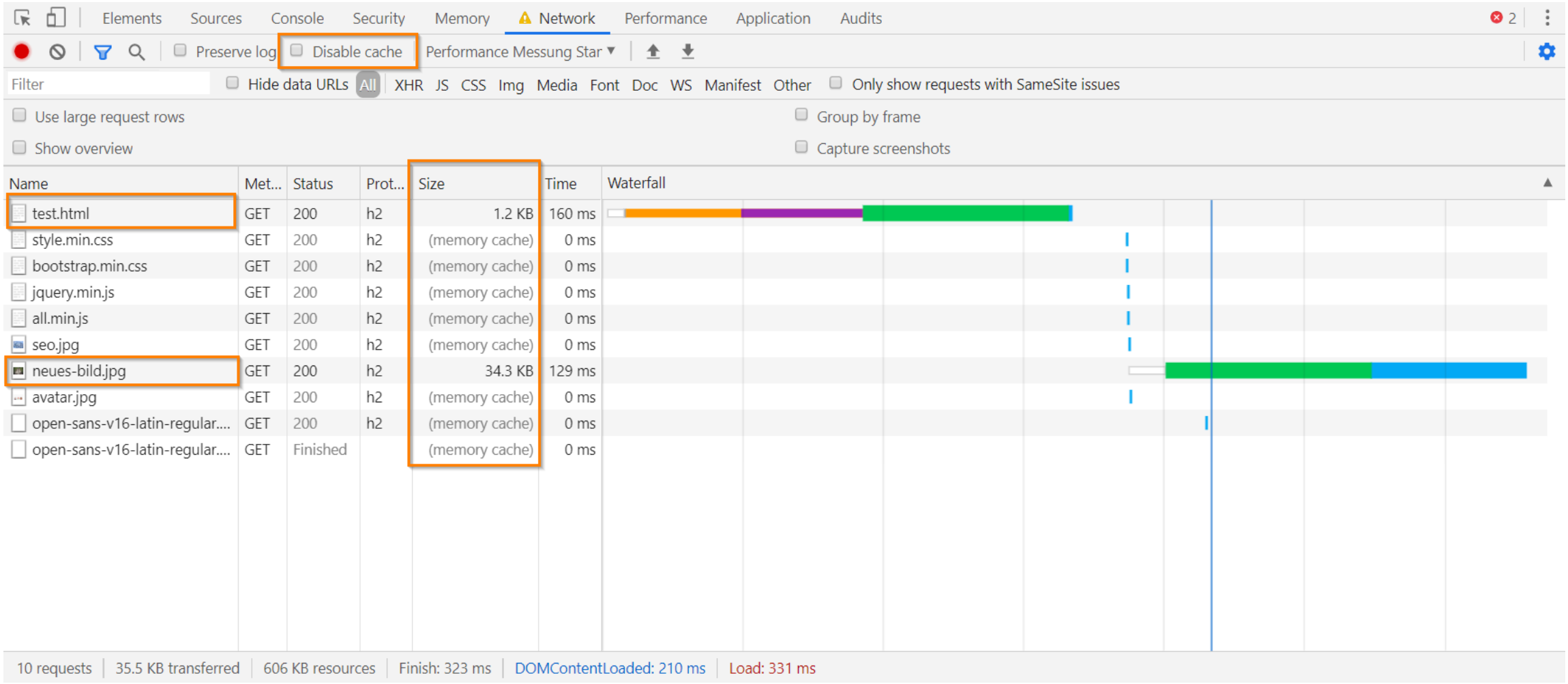
- Ordentliche Response-Time
- Browser-Caching
- Dateigröße

- Ordentliche Response-Time
- **Browser-Caching**
- Dateigröße

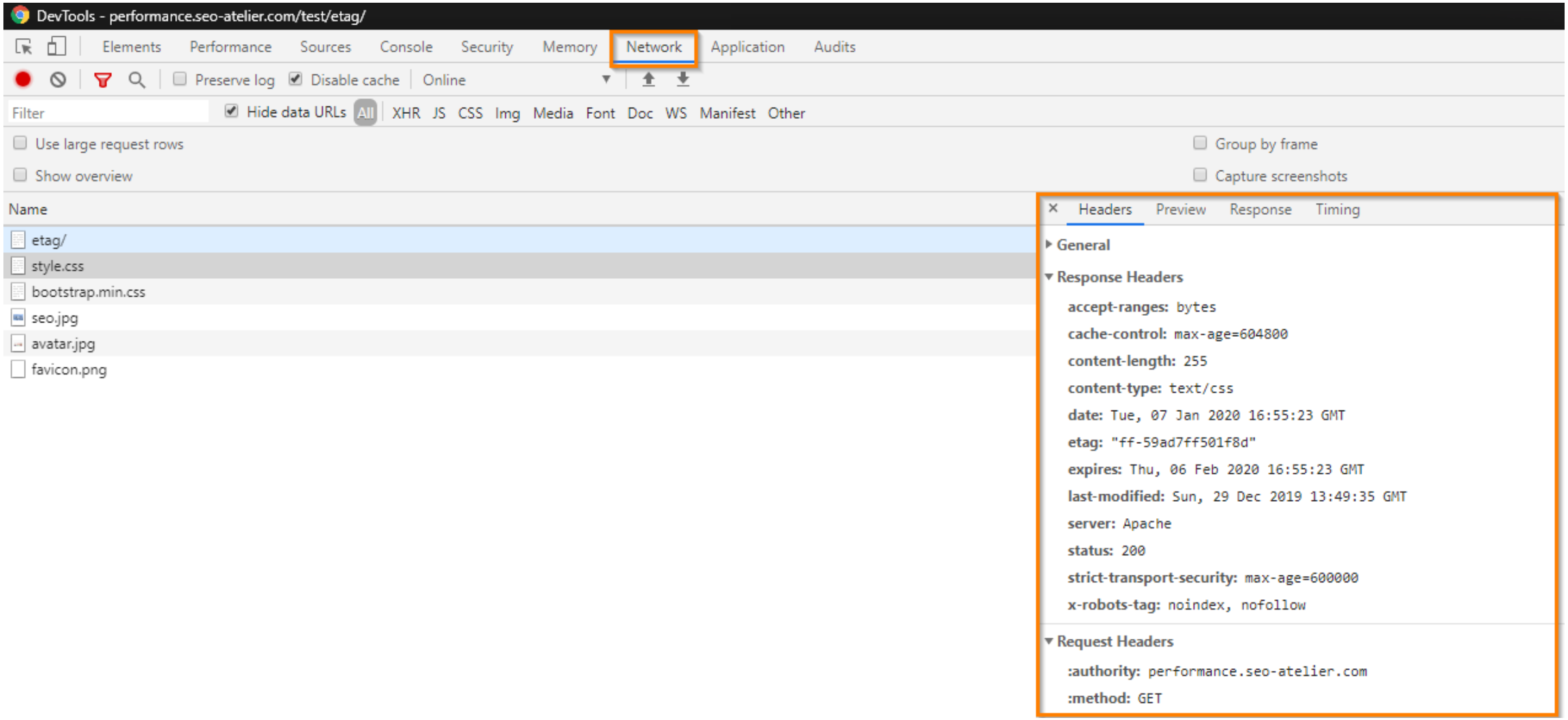
Warum das wichtig ist



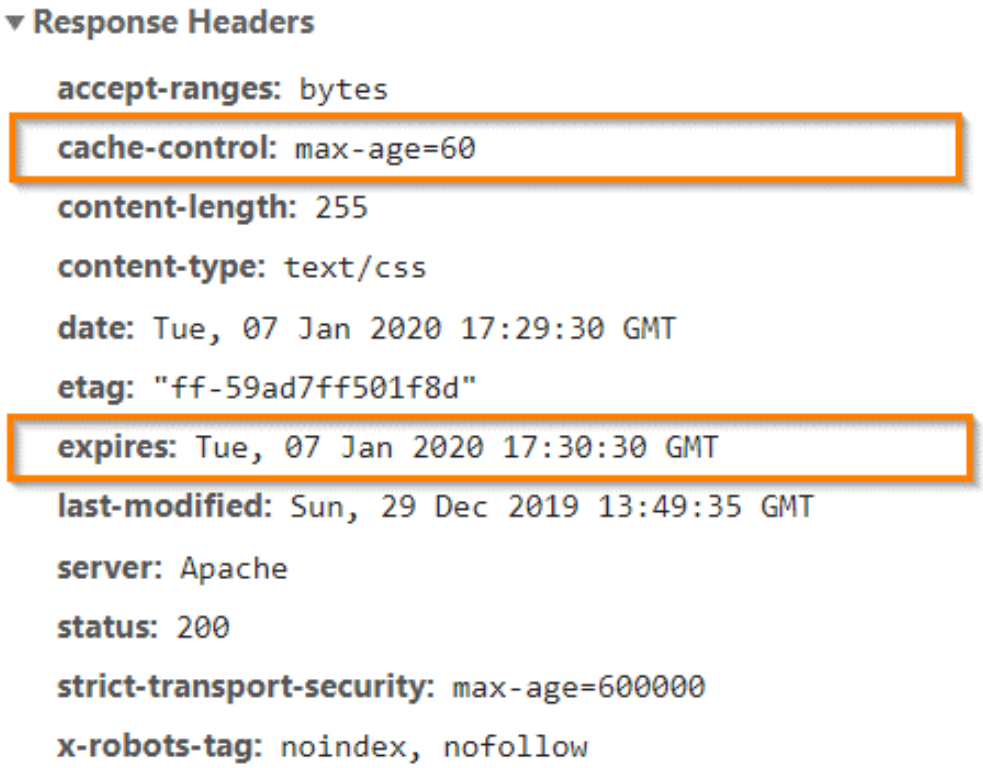
Warum das wichtig ist



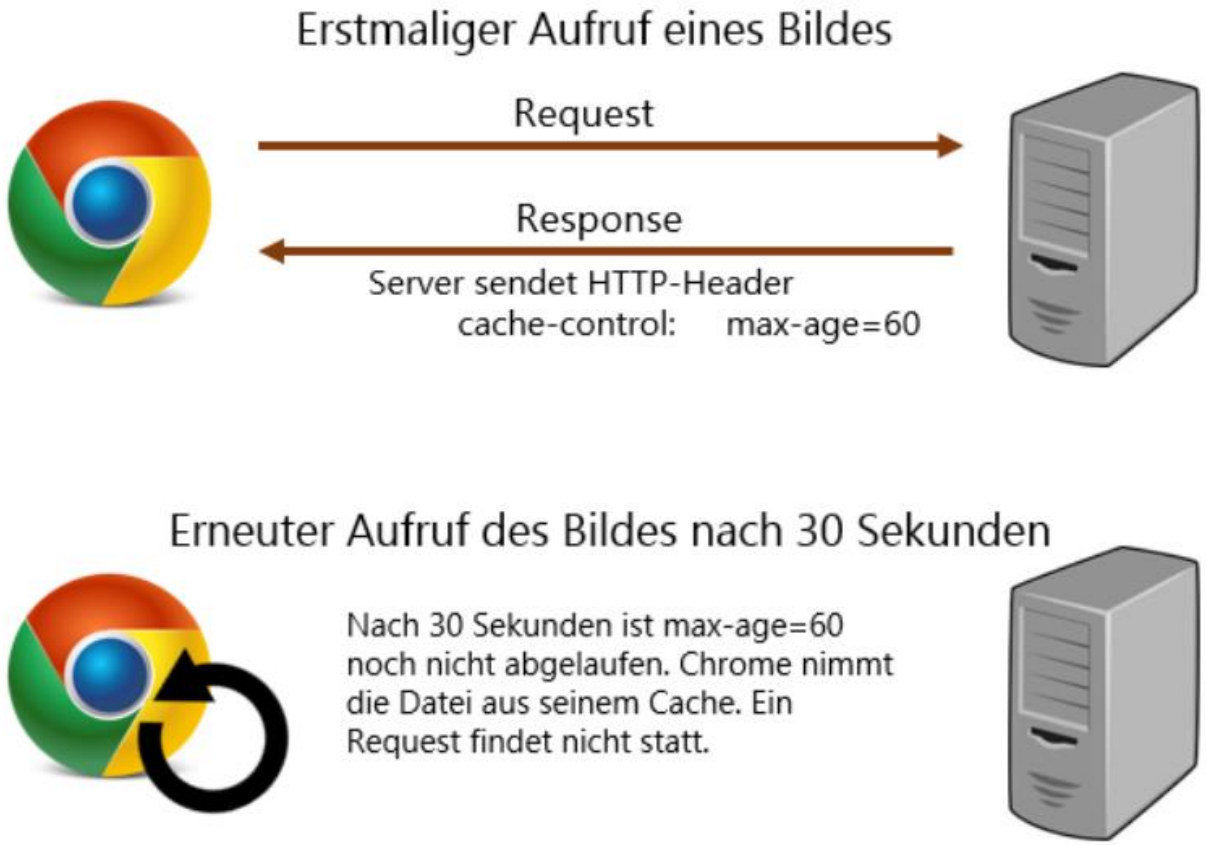
Caching & Header



Caching & Header

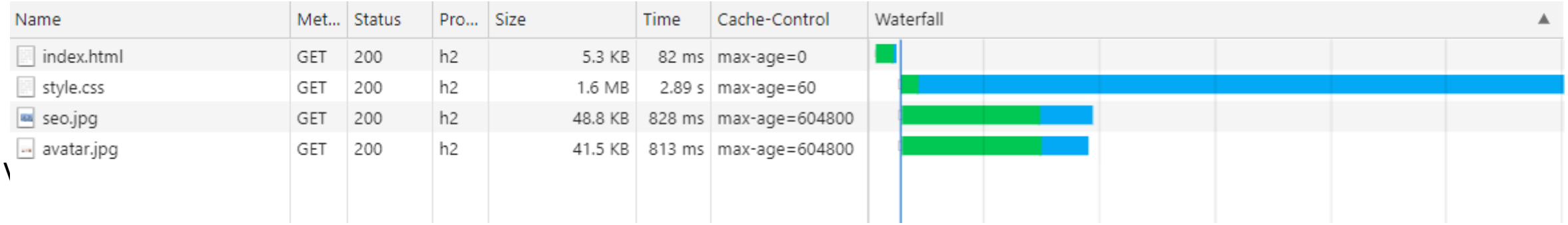


Innerhalb der TTL



Caching-Verhalten & TTL-Ablauf

Erster Aufruf



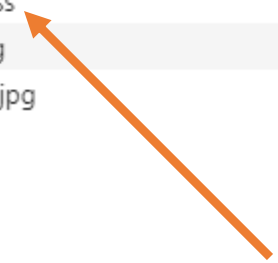
Name	Met...	Status	Pro...	Size	Time	Cache-Control	Waterfall
index.html	GET	200	h2	5.3 KB	82 ms	max-age=0	
style.css	GET	200	h2	1.6 MB	2.89 s	max-age=60	
seo.jpg	GET	200	h2	48.8 KB	828 ms	max-age=604800	
avatar.jpg	GET	200	h2	41.5 KB	813 ms	max-age=604800	

\

Caching-Verhalten & TTL-Ablauf

Verhalten innerhalb v. 60 Sekunden

Name	Met...	Status	Pro...	Size	Time	Cache-Control	Waterfall
test.html	GET	304	h2	72 B	76 ms	max-age=0	
style.css	GET	200	h2	(memory cache)	0 ms	max-age=60	
seo.jpg	GET	200	h2	(memory cache)	0 ms	max-age=604800	
avatar.jpg	GET	200	h2	(memory cache)	0 ms	max-age=604800	



Verhalten nach 60 Sekunden

Name	Met...	Status	Pro...	Size	Time	Cache-Control	Waterfall
index.html	GET	304	h2	80 B	134 ms	max-age=0	
style.css	GET	304	h2	75 B	76 ms	max-age=60	
seo.jpg	GET	200	h2	(memory cache)	0 ms	max-age=604800	
avatar.jpg	GET	200	h2	(memory cache)	0 ms	max-age=604800	

▼ Response Headers

accept-ranges: bytes

cache-control: max-age=60

content-length: 255

content-type: text/css

date: Tue, 07 Jan 2020 17:24:27 GMT

etag: "ff-59ad7ff501f8d"

expires: Tue, 07 Jan 2020 17:25:27 GMT

last-modified: Sun, 29 Dec 2019 13:49:35 GMT

server: Apache

status: 304

x-robots-tag: noindex, nofollow

▼ Request Headers

:authority: performance.seo-atelier.com

:method: GET

:path: /test/etag/css/style.css

:scheme: https

accept: text/css,*/*;q=0.1

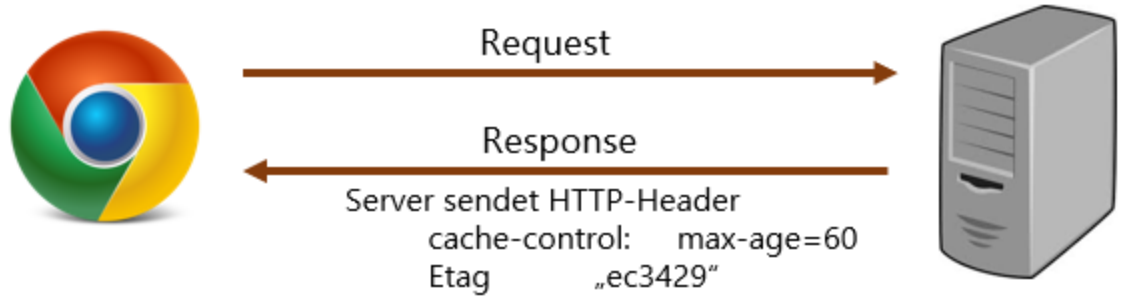
accept-encoding: gzip, deflate, br

accept-language: de-DE,de;q=0.9,en-US;q=0.8,en;q=0.7,lb;q=0.6,es;q=0.5,ru;q=0.4

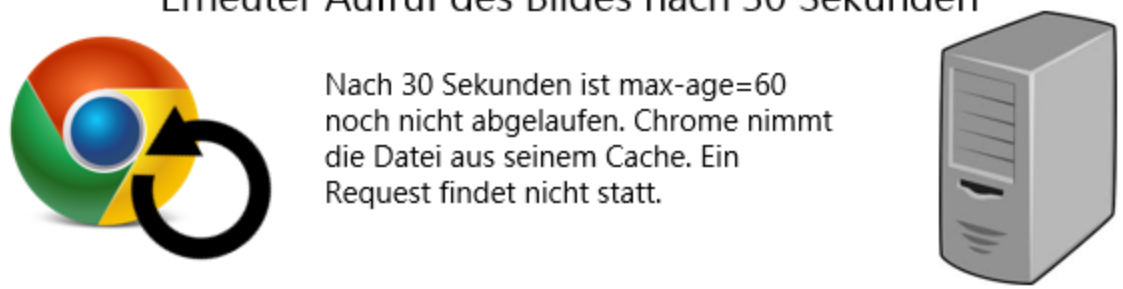
if-modified-since: Sun, 29 Dec 2019 13:49:35 GMT

if-none-match: "ff-59ad7ff501f8d"

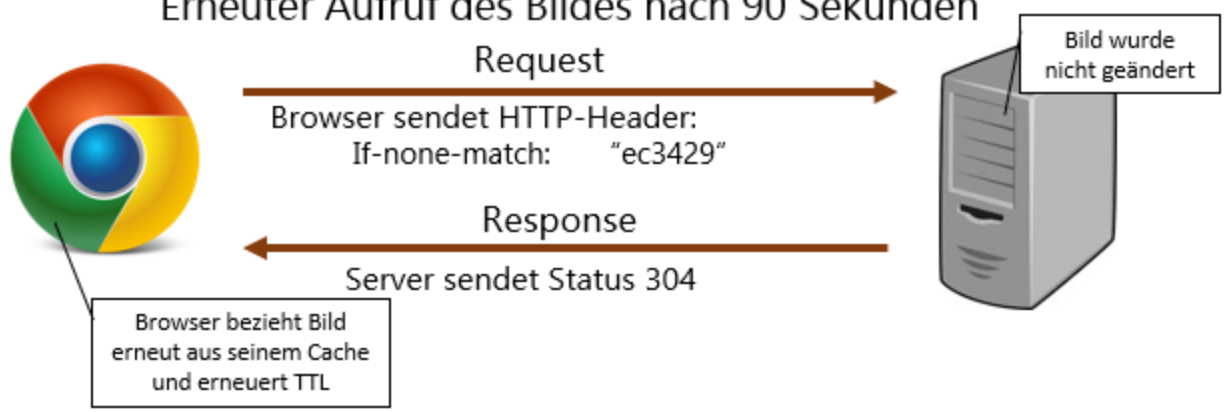
Erstmaliger Aufruf eines Bildes



Erneuter Aufruf des Bildes nach 30 Sekunden

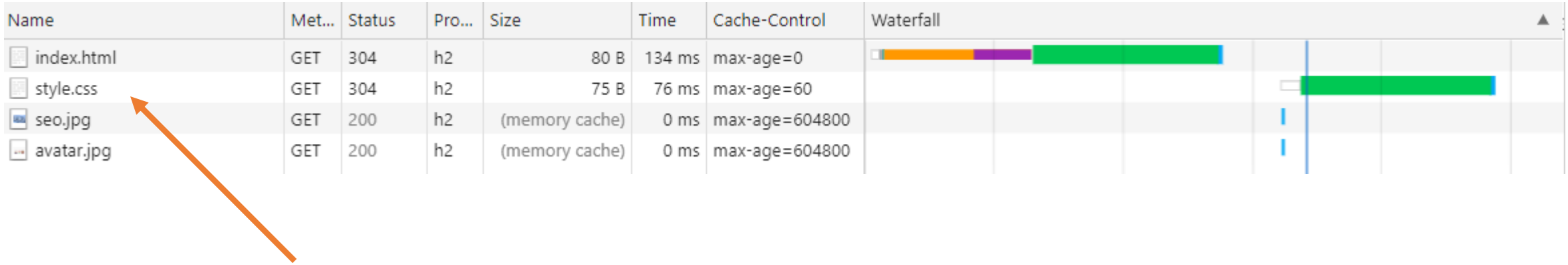


Erneuter Aufruf des Bildes nach 90 Sekunden



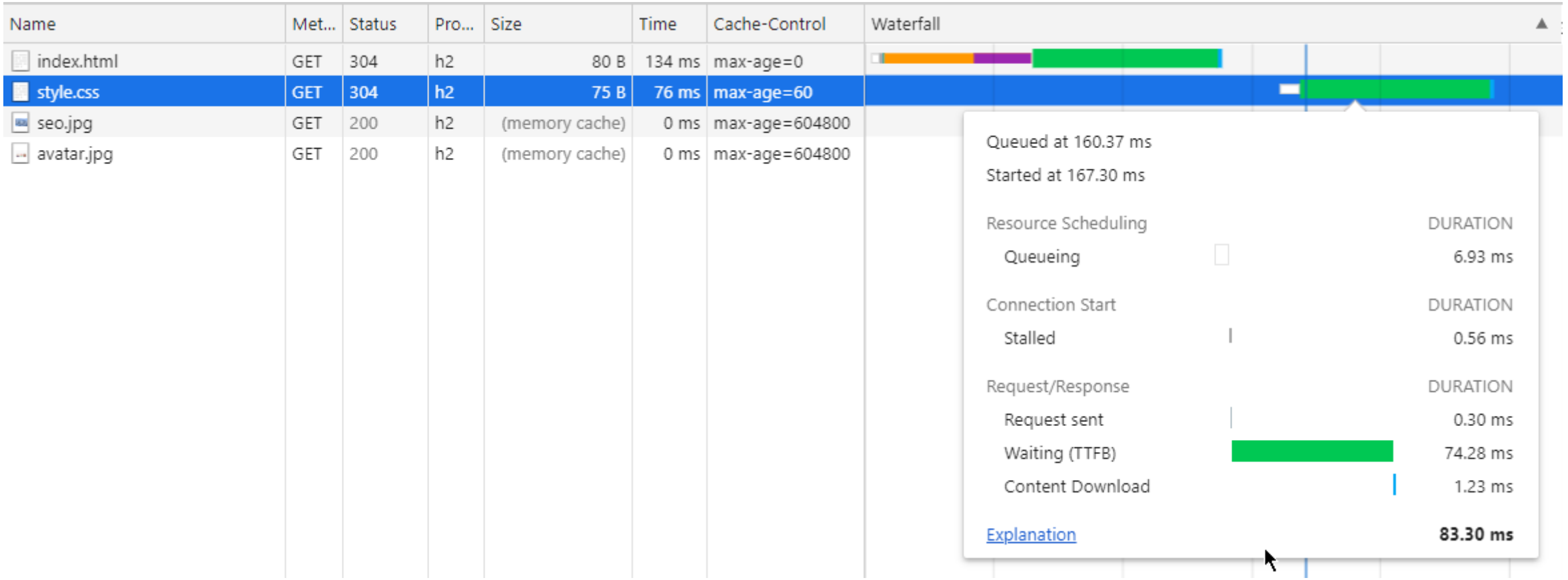
Caching-Verhalten & TTL-Ablauf

Verhalten nach 60 Sekunden



Name	Met...	Status	Pro...	Size	Time	Cache-Control	Waterfall
index.html	GET	304	h2	80 B	134 ms	max-age=0	
style.css	GET	304	h2	75 B	76 ms	max-age=60	
seo.jpg	GET	200	h2	(memory cache)	0 ms	max-age=604800	
avatar.jpg	GET	200	h2	(memory cache)	0 ms	max-age=604800	

Auch die Validierung kostet Zeit



Browser Caching – Aktualität vs. TTL

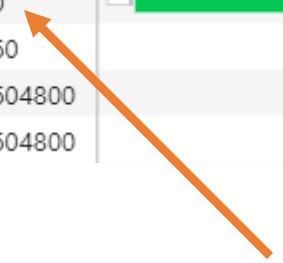
```
<img src="img/seo.jpg" alt="Etags und Validierung" title="Validierung mittels Etag"/>
```

```
<img src="img/72932348239 46/seo.jpg" alt="Etags und Validierung" title="Validierung mittels Etag"/>
```

Caching & HTML

Deswegen macht das Sinn:

Name	Met...	Status	Pro...	Size	Time	Cache-Control	Waterfall
test.html	GET	304	h2	72 B	76 ms	max-age=0	
style.css	GET	200	h2	(memory cache)	0 ms	max-age=60	
seo.jpg	GET	200	h2	(memory cache)	0 ms	max-age=604800	
avatar.jpg	GET	200	h2	(memory cache)	0 ms	max-age=604800	



Beispiel f. d. .htaccess

```
#######################################################
# Beispiel: Cache-Control-Angabe für Bilder           #
#######################################################

<IfModule mod_headers.c>
    <FilesMatch "\.(gif|ico|jpeg|jpg|png)$">
        Header set Cache-Control "max-age=604800"
    </FilesMatch>
</IfModule>
#######################################
# Setzen v. Expires-Header            #
#######################################
<IfModule mod_expires.c>
    ExpiresActive On
    ExpiresByType image/jpg "access plus 1 month"
    ExpiresByType image/jpeg "access plus 1 month"
    ExpiresByType image/gif "access plus 1 month"
    ExpiresByType image/png "access plus 1 month"
    ExpiresByType text/css "access plus 60 seconds"
    ExpiresByType application/pdf "access plus 1 month"
    ExpiresByType text/x-javascript "access plus 1 month"
    ExpiresByType application/x-shockwave-flash "access plus 1 month"
    ExpiresByType image/x-icon "access plus 1 year"
    ExpiresDefault "access plus 2 days"
</IfModule>

#Etag aktivieren
FileETag INode MTime Size

#Etag deaktivieren
Header unset ETag
```

Tipp: Achtung beim Testen

Reload != Navigation

Name	Method	Status	Protocol	Type	Size	Time	Cache-Control	ETag
etag/	GET	304	h2	document	63 B	25 ms	max-age=0	"13c7-5
style.css	GET	200	h2	stylesheet	(memory cache)	0 ms	max-age=60	"ff-59a
bootstrap.min.css	GET	200	h2	stylesheet	(memory cache)	0 ms	max-age=60	"235ed
seo.jpg	GET	200	h2	jpeg	(memory cache)	0 ms	max-age=604800	"c26f-5
avatar.jpg	GET	200	h2	jpeg	(memory cache)	0 ms	max-age=604800	"a5bc-5

Entwicklerwerkzeuge - Caching & Co. Testseite - SEO-Atelier - https://performance.seo-atelier.com/test/etag/

Inspektor | Konsole | Debugger | Stilbearbeitung | Speicher | Netzwerkanalyse | Web-Speicher | Barrierefreiheit | DOM | JS-Umgebung

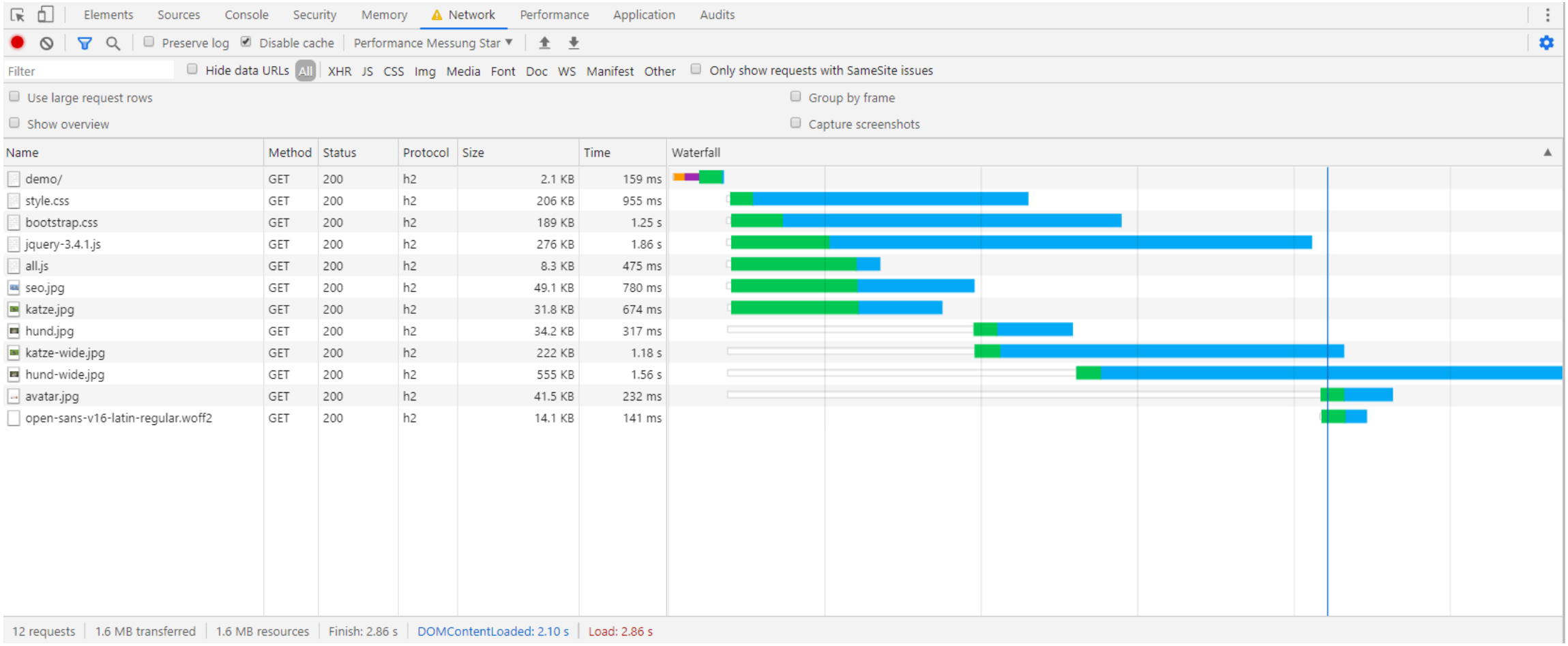
Adressen durchsuchen | Alles

Status	Methode	Datei	Ursprung	Typ	Übertragen	Größe	Start	Dauer	Cache-Control	ETag
304	GET	/test/etag/	document	html	Aus Cache	4,94 KB	0 ms	163 ms	max-age=0	"13c7-59b906cec50c0"
304	GET	style.css	stylesheet	css	Aus Cache	255 B	158 ms	23 ms	max-age=60	"ff-59ad7ff501f8d"
304	GET	bootstrap.min.css	stylesheet	css	Aus Cache	141,48 KB	158 ms	23 ms	max-age=60	"235ed-59a4be94f3c5e"
304	GET	seo.jpg	img	jpeg	Aus Cache	48,61 KB	204 ms	23 ms	max-age=604800	"c26f-59a4c17c8dc02"
304	GET	avatar.jpg	img	jpeg	Aus Cache	41,43 KB	204 ms	23 ms	max-age=604800	"a5bc-59a4c1e5ab4fd"

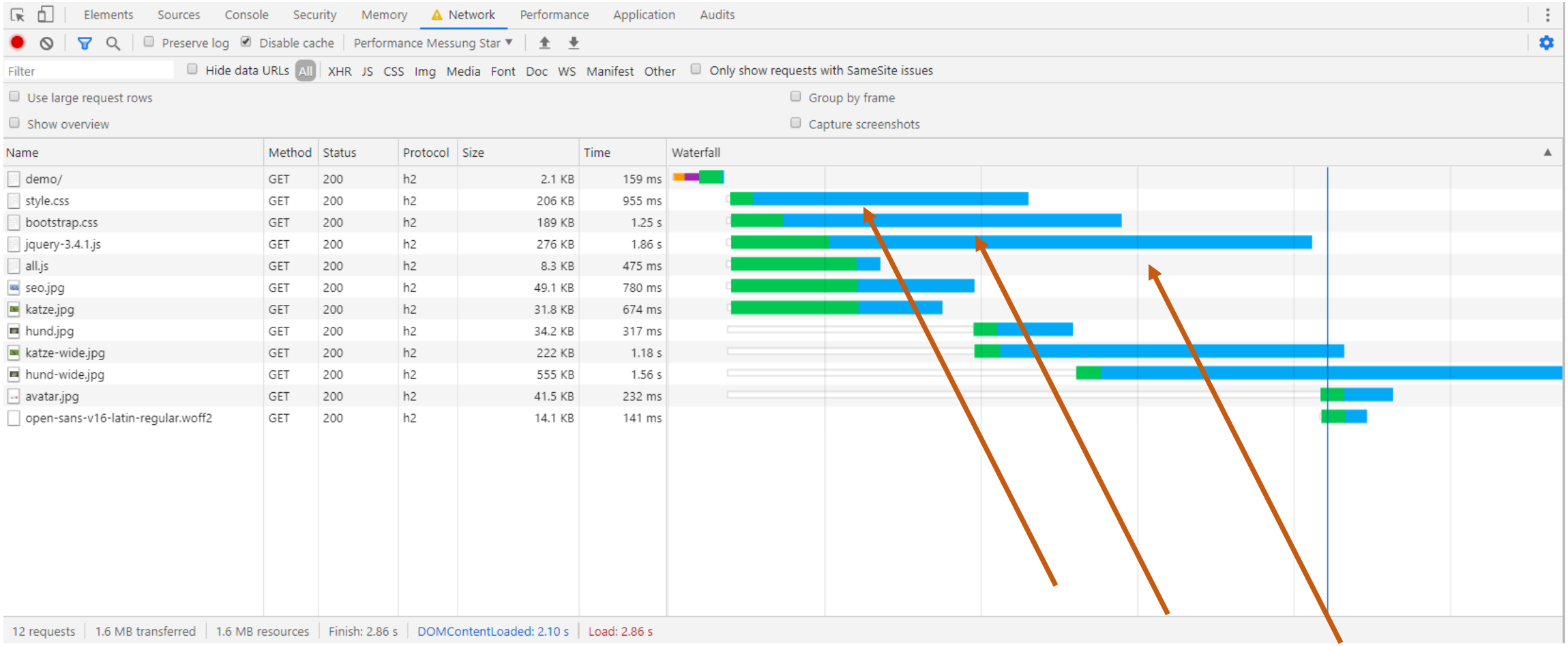
- Ordentliche Response-Time
- Browser-Caching
- Dateigröße

- Ordentliche Response-Time
- Browser-Caching
- **Dateigröße**

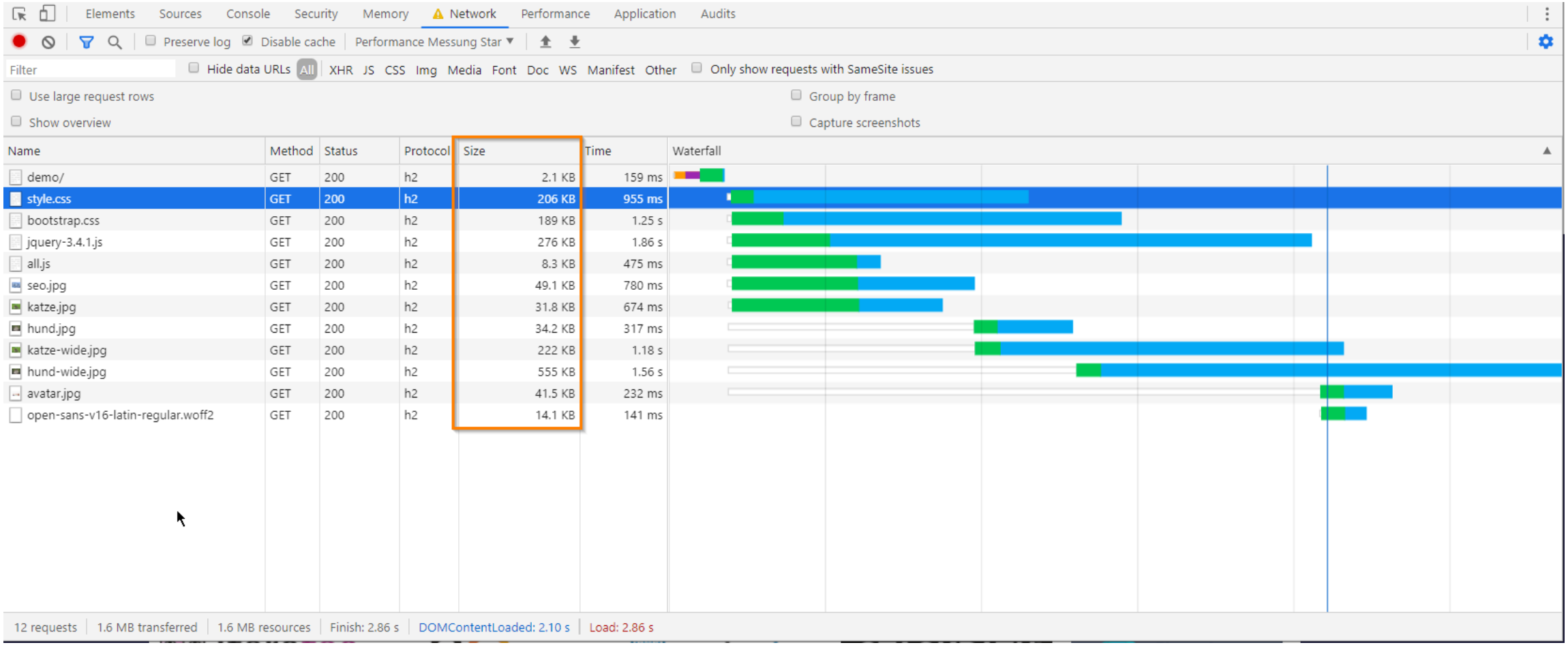
Große Dateien brauchen länger



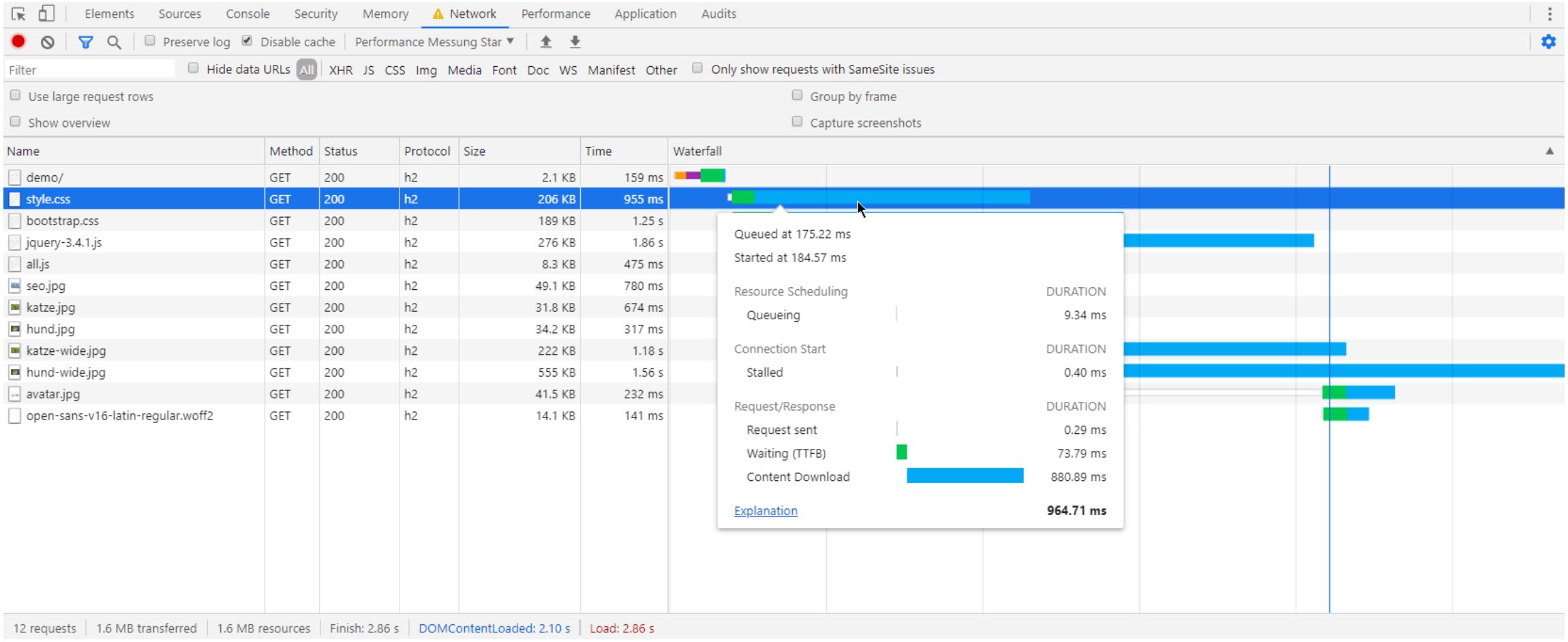
Große Dateien brauchen länger



Große Dateien brauchen länger



Große Dateien brauchen länger



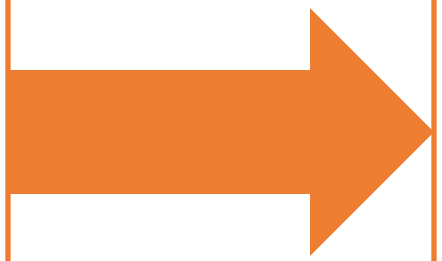
Textbasierte-Files

Minimierung

Minimierung

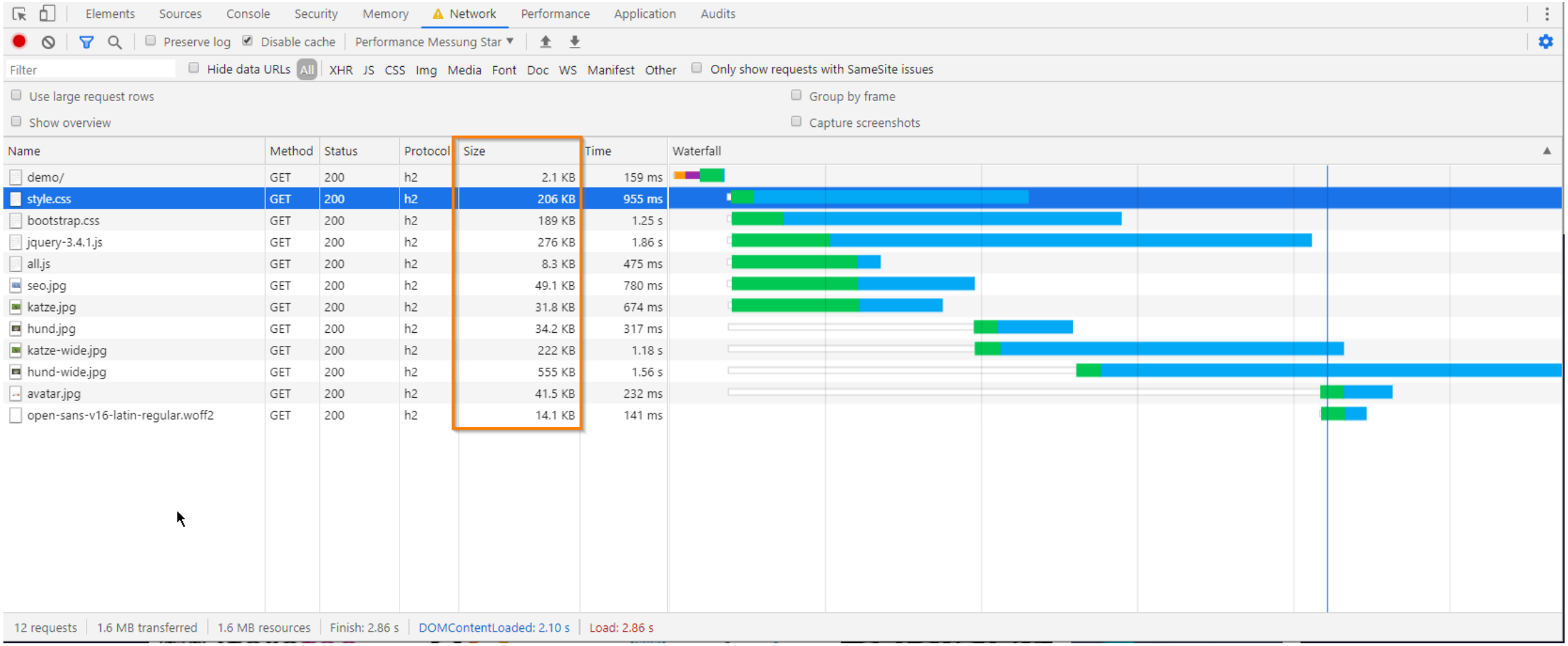
```
/* The Modal (background) */
.modal {
  display: none;
  position: fixed;
  z-index: 1;
  padding-top: 100px;
  left: 0;
  top: 0;
  width: 100%;
  height: 100%;
  overflow: auto;
  background-color: black;
}

/* Modal Content */
.modal-content {
  position: relative;
  background-color: #fefefe;
  margin: auto;
  padding: 0;
  width: 90%;
  max-width: 1200px;
}
```

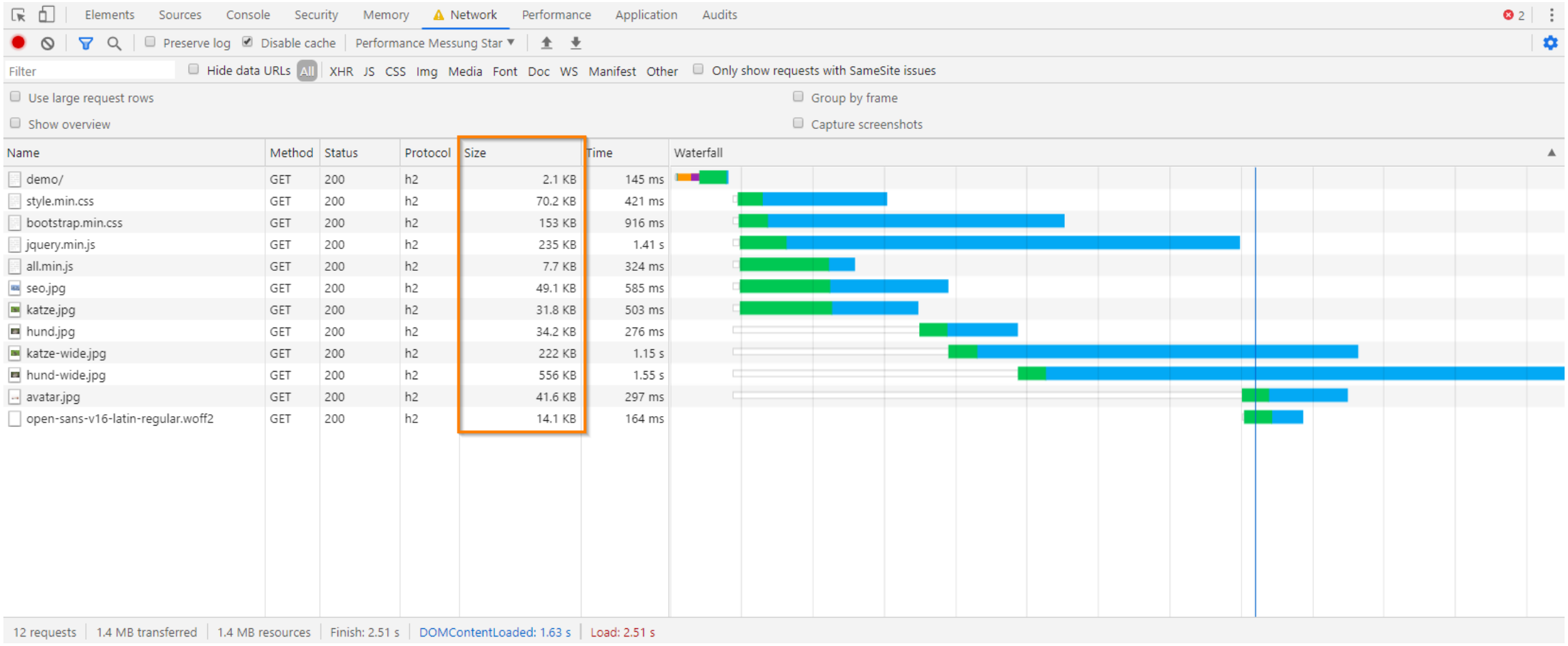


```
.modal{display:none;position:fixed;z-index:1;padding-top:100px;left:0;top:0;width:100%;height:100%;overflow:auto;background-color:#000}.modal-content{position:relative;background-color:#fefefe;margin:auto;padding:0;width:90%;max-width:1200px}
```

Minimierung

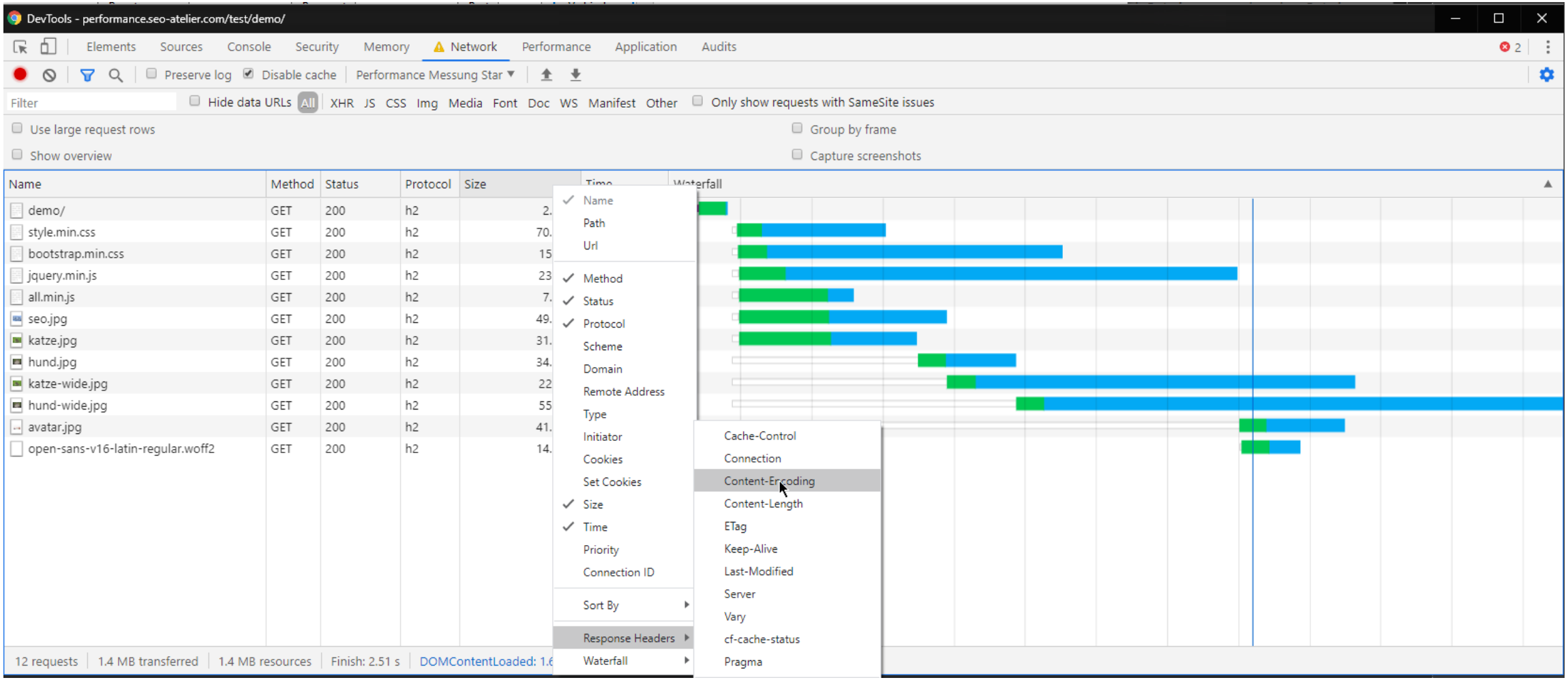


Minimierung

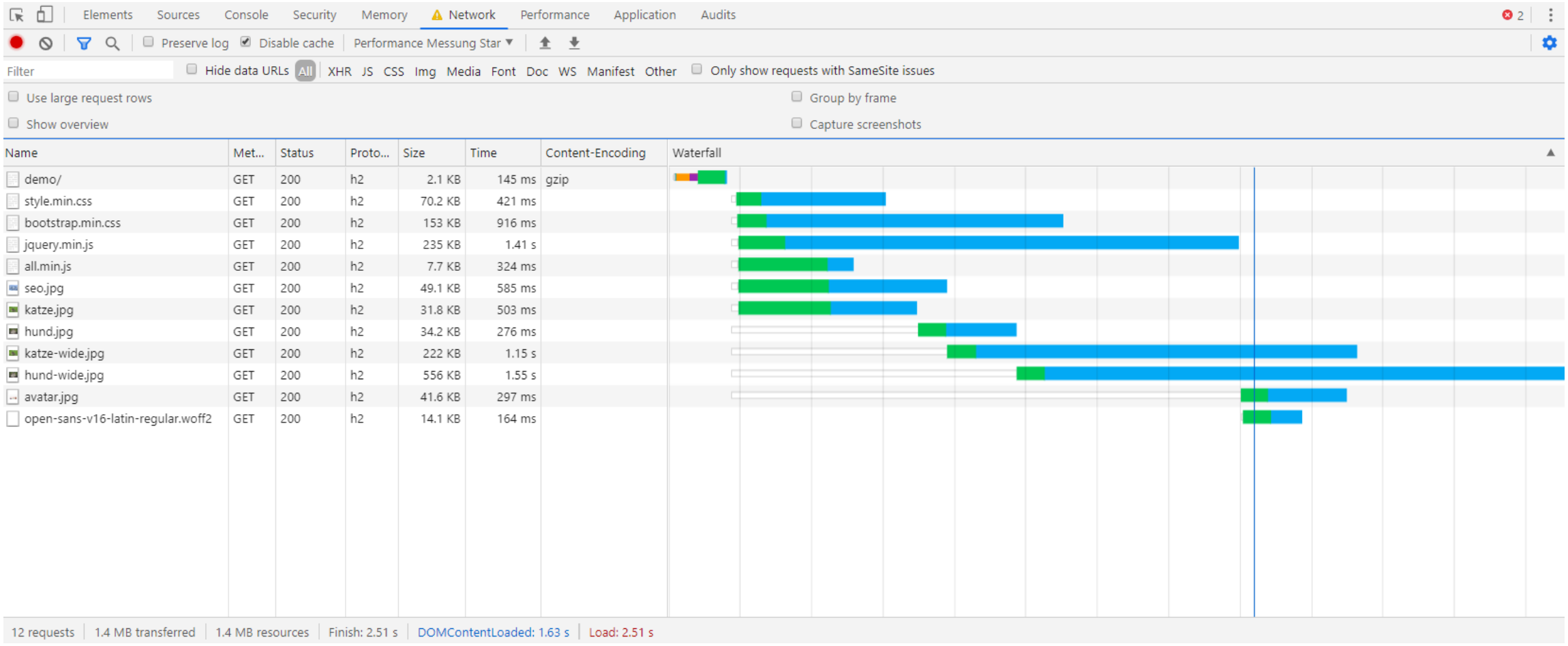


Text Compression

Komprimierung



Komprimierung

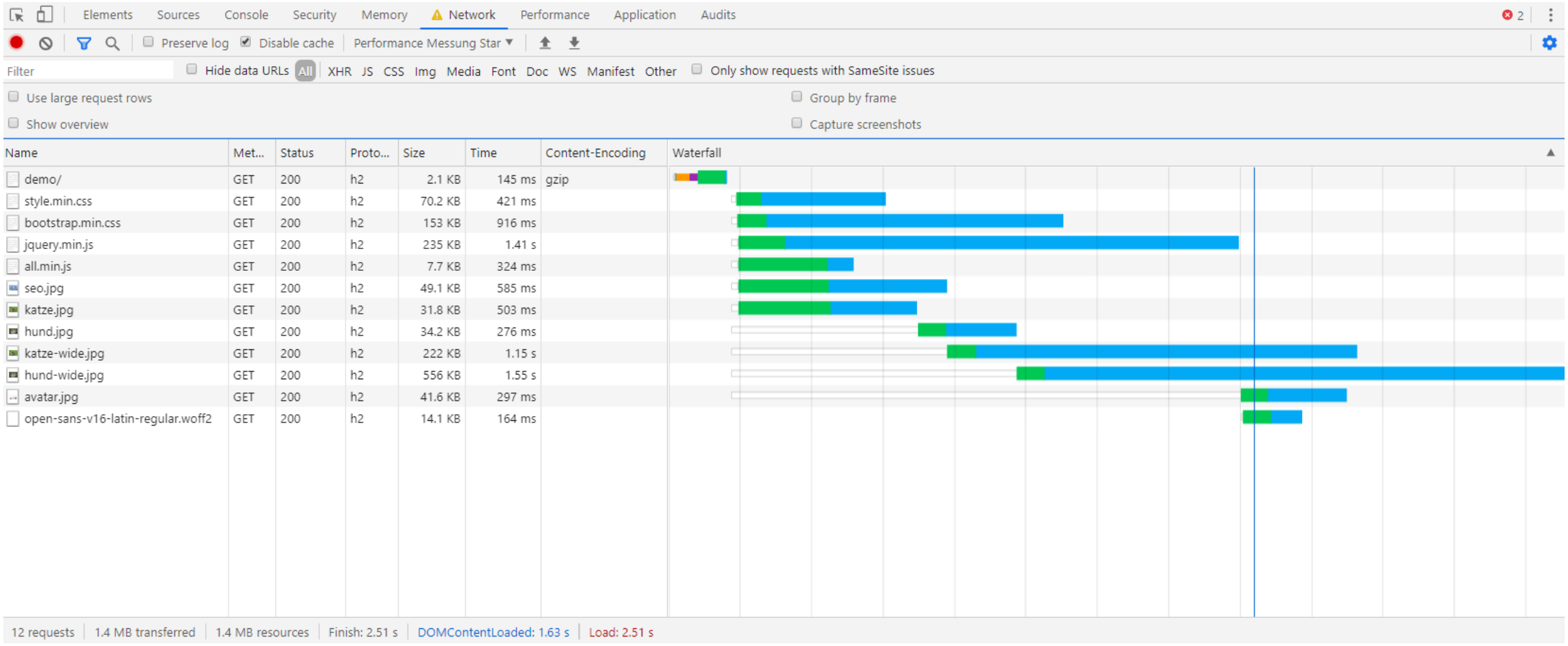


Beispiel f. d. .htaccess

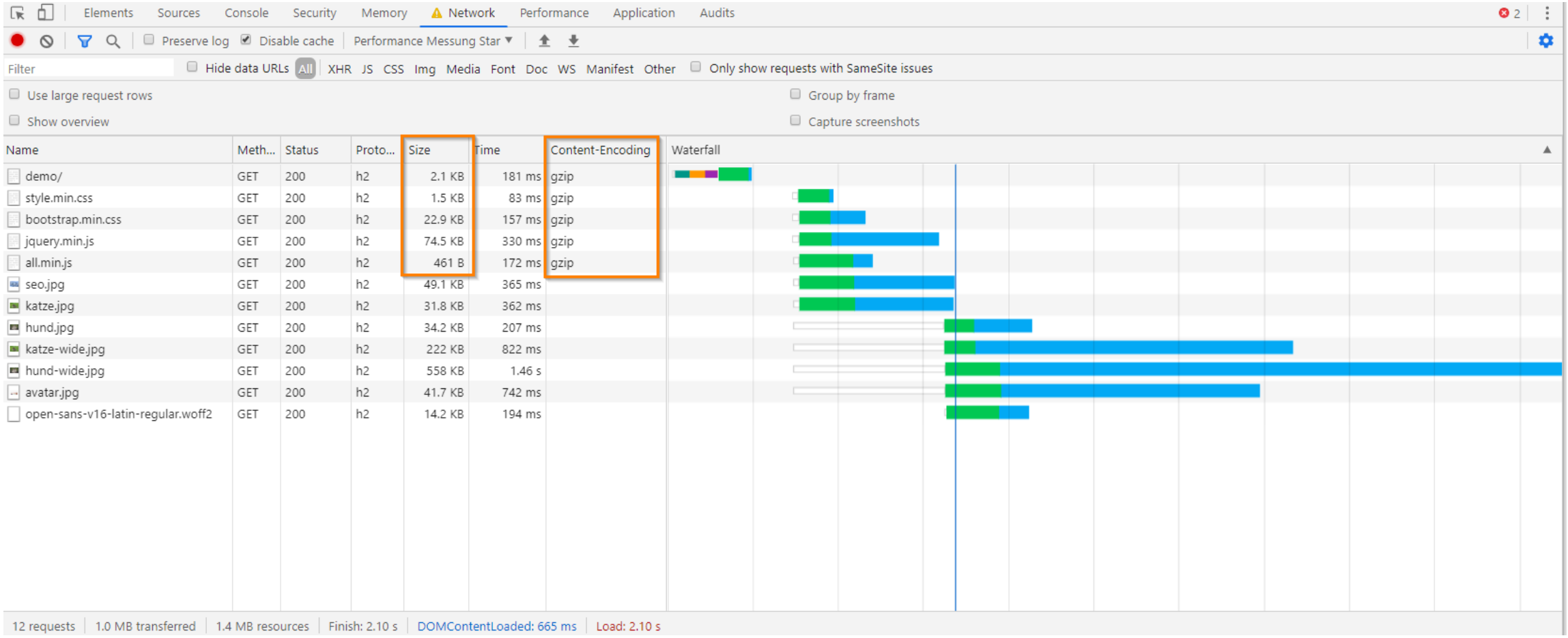
```
<ifModule mod_headers.c>
    Header set X-Robots-Tag "noindex, follow"
</ifModule>

<IfModule mod_deflate.c>
    AddOutputFilterByType DEFLATE text/plain
    AddOutputFilterByType DEFLATE text/html
    AddOutputFilterByType DEFLATE text/xml
    AddOutputFilterByType DEFLATE text/shtml
    AddOutputFilterByType DEFLATE text/css
    AddOutputFilterByType DEFLATE application/xml
    AddOutputFilterByType DEFLATE application/xhtml+xml
    AddOutputFilterByType DEFLATE application/rss+xml
    AddOutputFilterByType DEFLATE application/javascript
    AddOutputFilterByType DEFLATE application/x-javascript
</IfModule>
```

GZIP & Chrome


Elements
Sources
Console
Security
Memory
Network
Performance
Application
Audits
Preserve log
Disable cache
Performance Messung Star
Filter
Hide data URLs
All
XHR
JS
CSS
Img
Media
Font
Doc
WS
Manifest
Other
Only show requests with SameSite issues
Use large request rows
Group by frame
Show overview
Capture screenshots
Name
Met...
Status
Proto...
Size
Time
Content-Encoding
Waterfall
demo/ GET 200 h2 2.1 KB 145 ms gzip
style.min.css GET 200 h2 70.2 KB 421 ms
bootstrap.min.css GET 200 h2 153 KB 916 ms
jquery.min.js GET 200 h2 235 KB 1.41 s
all.min.js GET 200 h2 7.7 KB 324 ms
seo.jpg GET 200 h2 49.1 KB 585 ms
katze.jpg GET 200 h2 31.8 KB 503 ms
hund.jpg GET 200 h2 34.2 KB 276 ms
katze-wide.jpg GET 200 h2 222 KB 1.15 s
hund-wide.jpg GET 200 h2 556 KB 1.55 s
avatar.jpg GET 200 h2 41.6 KB 297 ms
open-sans-v16-latin-regular.woff2 GET 200 h2 14.1 KB 164 ms
12 requests
1.4 MB transferred
1.4 MB resources
Finish: 2.51 s
DOMContentLoaded: 1.63 s
Load: 2.51 s

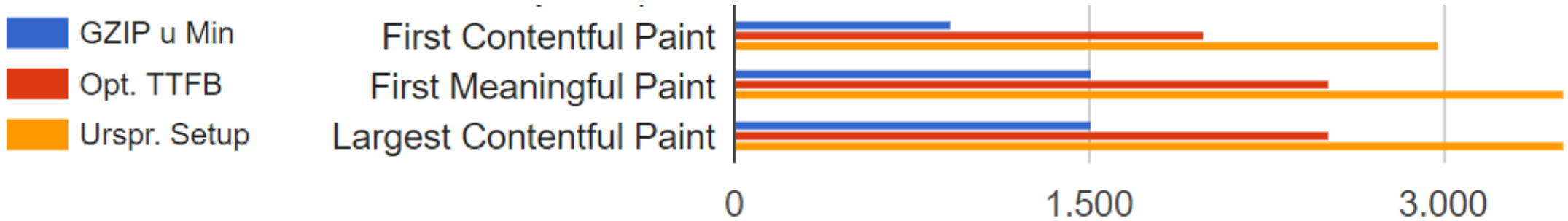
GZIP & Chrome


Elements
Sources
Console
Security
Memory
Network
Performance
Application
Audits
Preserve log
Disable cache
Performance Messung Star
Filter
Hide data URLs
All
XHR
JS
CSS
Img
Media
Font
Doc
WS
Manifest
Other
Only show requests with SameSite issues
Use large request rows
Group by frame
Show overview
Capture screenshots
Name
Meth...
Status
Proto...
Size
Time
Content-Encoding
Waterfall
demo/ GET 200 h2 2.1 KB 181 ms gzip
style.min.css GET 200 h2 1.5 KB 83 ms gzip
bootstrap.min.css GET 200 h2 22.9 KB 157 ms gzip
jquery.min.js GET 200 h2 74.5 KB 330 ms gzip
all.min.js GET 200 h2 461 B 172 ms gzip
seo.jpg GET 200 h2 49.1 KB 365 ms
katze.jpg GET 200 h2 31.8 KB 362 ms
hund.jpg GET 200 h2 34.2 KB 207 ms
katze-wide.jpg GET 200 h2 222 KB 822 ms
hund-wide.jpg GET 200 h2 558 KB 1.46 s
avatar.jpg GET 200 h2 41.7 KB 742 ms
open-sans-v16-latin-regular.woff2 GET 200 h2 14.2 KB 194 ms
12 requests
1.0 MB transferred
1.4 MB resources
Finish: 2.10 s
DOMContentLoaded: 665 ms
Load: 2.10 s

Performance Results (Median Run)

								Document Complete			Fully Loaded		
	Load Time	First Byte	Start Render	First Contentful Paint	Speed Index	Last Painted Hero	First CPU Idle	Time	Requests	Bytes In	Time	Requests	Bytes In
First View (Run 1)	3.647s	0.409s	2.000s	1.984s	2.338s	2.600s	> 2.000s	3.647s	12	1,620 KB	4.000s	13	1,620 KB

								Document Complete			Fully Loaded		
	Load Time	First Byte	Start Render	First Contentful Paint	Speed Index	Last Painted Hero	First CPU Idle	Time	Requests	Bytes In	Time	Requests	Bytes In
First View (Run 1)	2.590s	0.419s	0.900s	0.918s	1.217s	1.500s	> 0.900s	2.590s	12	1,044 KB	2.963s	13	1,044 KB

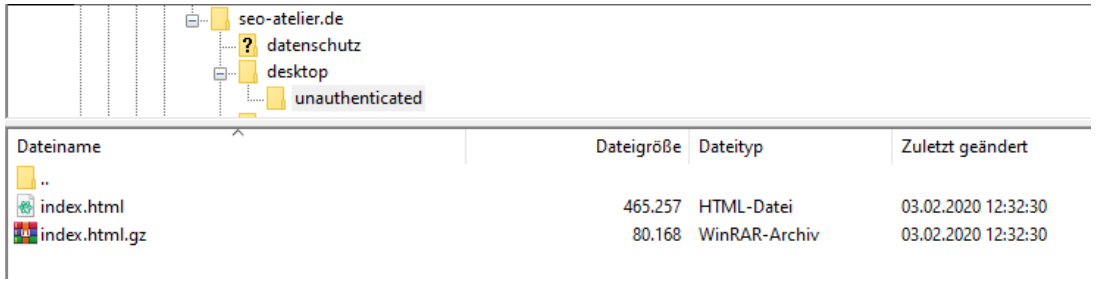


WP-Plugin Tipps

Tipp: WP-Plugin

Swift-Performance

- Cached Seiten als statische Dateien
- Legt als GZIP ab
- Text-Compression
- Image-Optimization (inkl. webp)



WP Fastest Cache

- Für ein schnelles & einfaches Setup

Tipp: Browser-Caching

- Selber machen ;)

WP Rocket

- Ähnlich umfangreich wie Swift-Performance

W3 Total Cache

- Sehr umfangreich
- Benötigt tiefere Kenntnisse
- Ideal, sofern weiterführende Funktionalität benötigt wird.

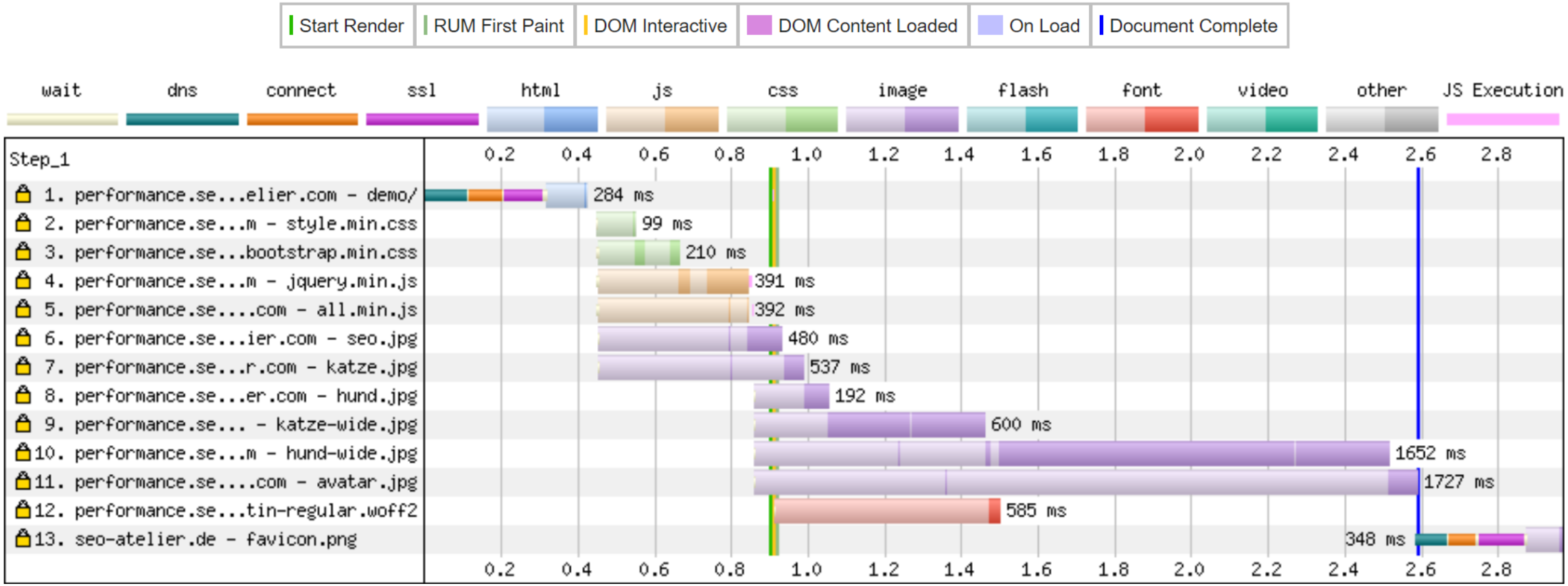
Pagespeed Hacks

(Hack 1: ASYNC/DEFER)

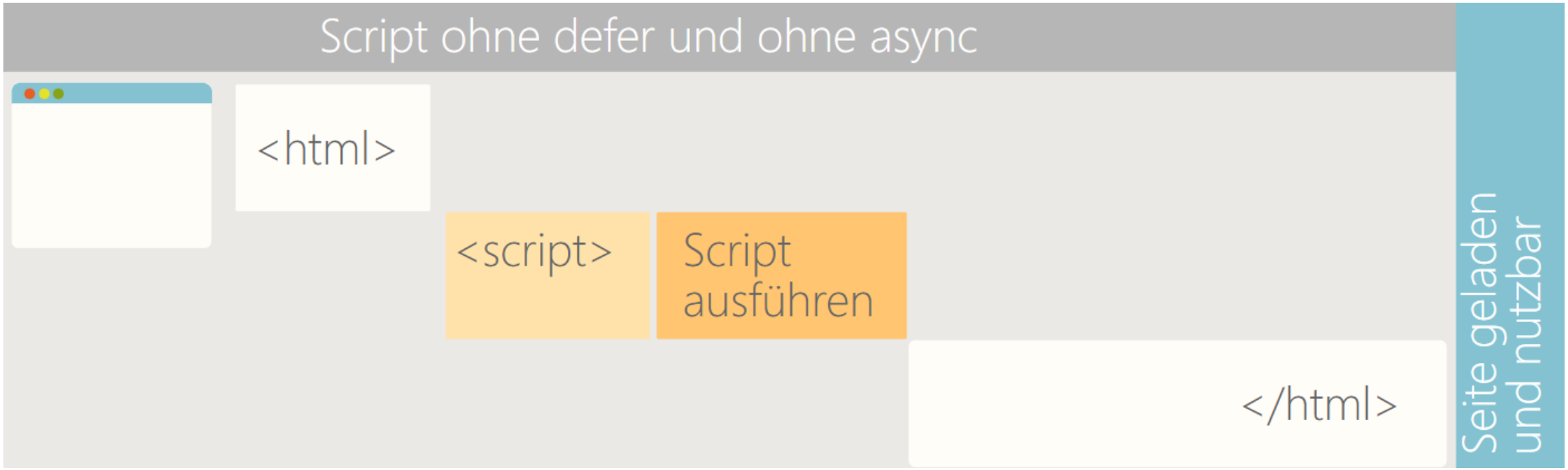
Synchrones JS blockiert das Parsen des DOMs

→JS fragt möglicherweise das CSSOM ab
→Browser muss auf CSS warten müssen, bevor er das JavaScript ausführen kann. (Auch bei Inlining v. JS)

Synchrones Laden von JS

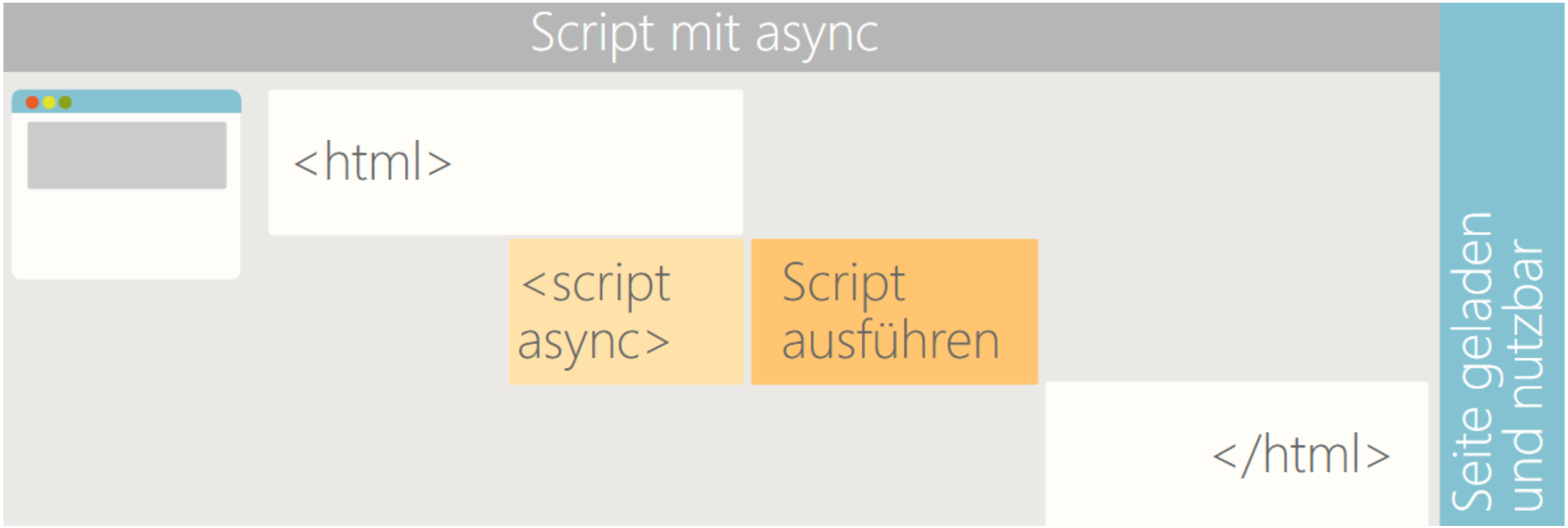


Synchrones Laden von JS



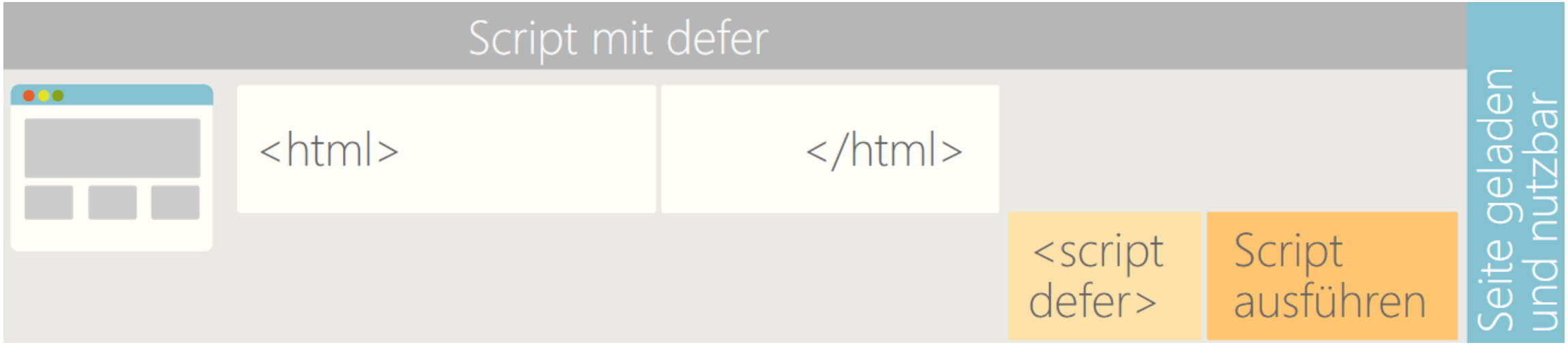
Quelle: https://www.mediaevent.de/javascript/programm-struktur.html

ASYNC JS



Quelle: https://www.mediaevent.de/javascript/programm-struktur.html

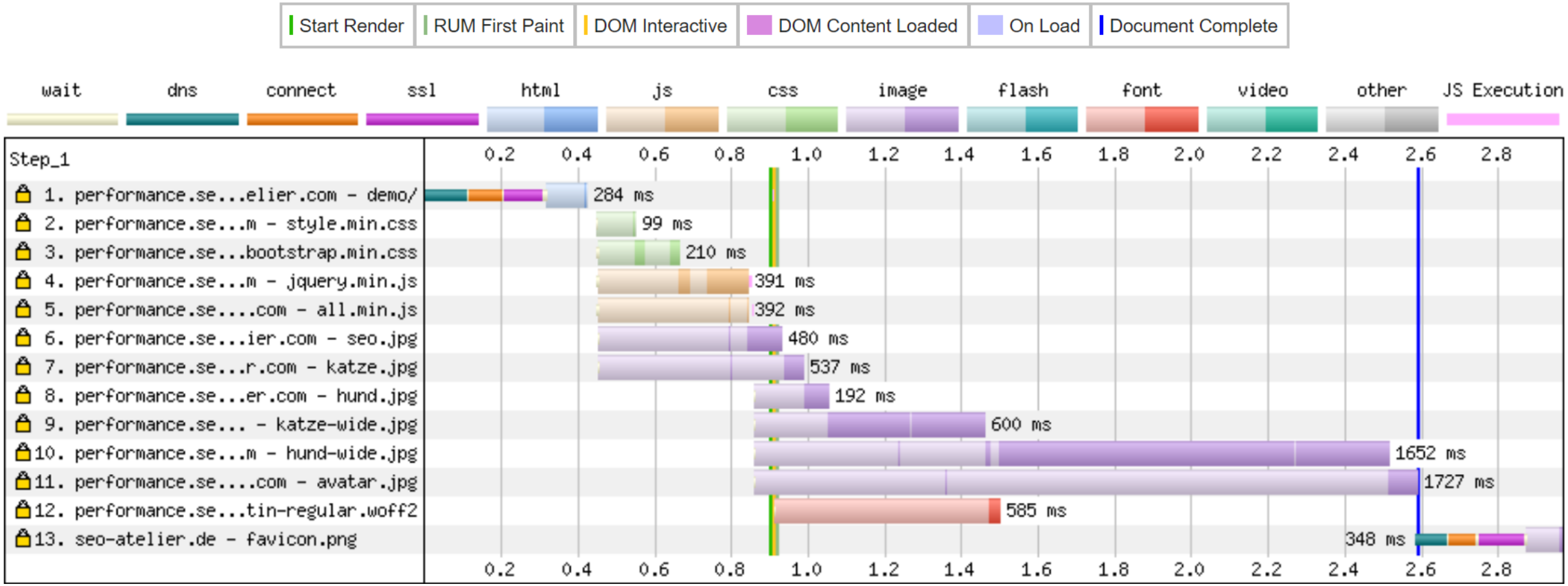
DEFER JS



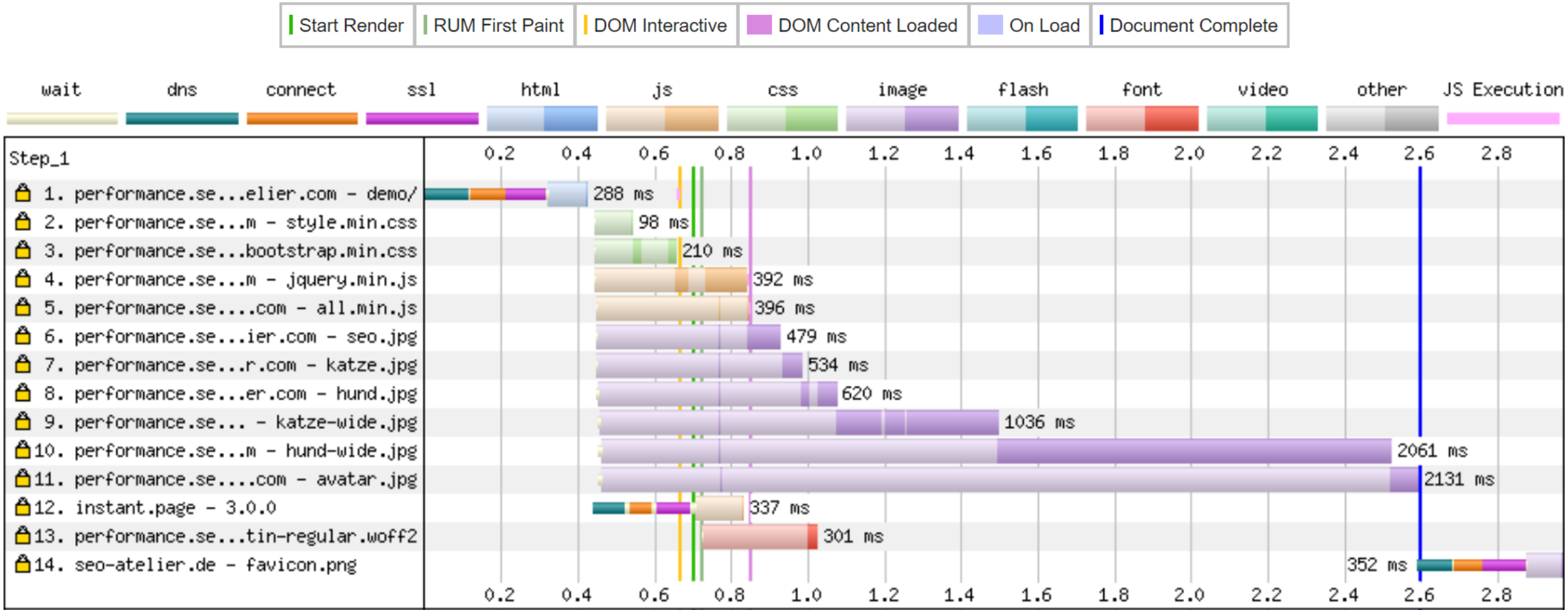
Wichtig: Defer behält die Ladereihenfolge der Dateien wie im Quellcode angegeben.

Quelle: https://www.mediaevent.de/javascript/programm-struktur.html

Synchrones Laden von JS



Mit Defer JS


Start Render
RUM First Paint
DOM Interactive
DOM Content Loaded
On Load
Document Complete
wait
dns
connect
ssl
html
js
css
image
flash
font
video
other
JS Execution
Step_1
1. performance.se...elier.com - demo/
2. performance.se...m - style.min.css
3. performance.se...bootstrap.min.css
4. performance.se...m - jquery.min.js
5. performance.se....com - all.min.js
6. performance.se...ier.com - seo.jpg
7. performance.se...r.com - katze.jpg
8. performance.se...er.com - hund.jpg
9. performance.se... - katze-wide.jpg
10. performance.se...m - hund-wide.jpg
11. performance.se....com - avatar.jpg
12. instant.page - 3.0.0
13. performance.se...tin-regular.woff2
14. seo-atelier.de - favicon.png
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
2.2
2.4
2.6
2.8
288 ms
98 ms
210 ms
392 ms
396 ms
479 ms
534 ms
620 ms
1036 ms
2061 ms
2131 ms
337 ms
301 ms
352 ms

(Hack 2: Inlining CSS)

```
1  <!DOCTYPE html>
2  <html lang="de-DE" prefix="og: http://ogp.me/ns#">
3    <head>
4      <meta charset="UTF-8">
5      <link href="css/style.css" rel="stylesheet" />
6      <title>Testseite | SEO-Atelier.de</title>
7      <script async type='text/javascript' src='js/all.min.js'></script>
8    </head>
9    <body>
```

(Hack 2: Inlining CSS)

```
<!DOCTYPE html>
<html lang="de-DE" prefix="og: http://ogp.me/ns#">
  <head>
    <meta charset="UTF-8">
    <style>
      body {
        background-color: #eee;
      }
      h1 {
        color: black;
        margin-left: 40px;
      }
    </style>
    <title>Testseite | SEO-Atelier.de</title>
    <script async type='text/javascript' src='js/all.min.js'></script>
  </head>
  <body>
```

(Hack 2: Inlining CSS)

Aber Achtung: Browser-Caching

(Hack 2: Inlining CSS)

Inlining Above-the-fold + Rest mittels JS nachladen

```
<!DOCTYPE html>
<html lang="de-DE" prefix="og: http://ogp.me/ns#">
  <head>
    <meta charset="UTF-8">
    <style>
      body {
        background-color: #eee;
      }
      h1 {
        color: black;
        margin-left: 40px;
      }
    </style>
    <link rel="preload" href="styles-below.css" as="style" onload="this.onload=null;this.rel='stylesheet'">
    <noscript><link rel="stylesheet" href="styles.css-below.css"></noscript>
    <title>Testseite | SEO-Atelier.de</title>
    <script async type='text/javascript' src='js/all.min.js'></script>
  </head>
  <body>
```

Führt leider meist zu FOUC

Demoseite - dient nur als Beispiel

Veranschaulichung zur Verwendung von Etags

Lorem ipsum 2. Testseite dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Lorem ipsum dolor sit amet, consetetur sadipscing et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd. At vero eos et accusam et justo duo dolores et ea rebum.

Hier steht eingeklappter Text

Ich bin eingeklappt

Lorem ipsum dolor sit amet

Lightbox

Demoseite - dient nur als Beispiel

Veranschaulichung zur Verwendung von Etags

Lorem ipsum 2. Testseite dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Lorem ipsum dolor sit amet, consetetur sadipscing et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd. At vero eos et accusam et justo duo dolores et ea rebum.

Hier steht eingeklappter Text

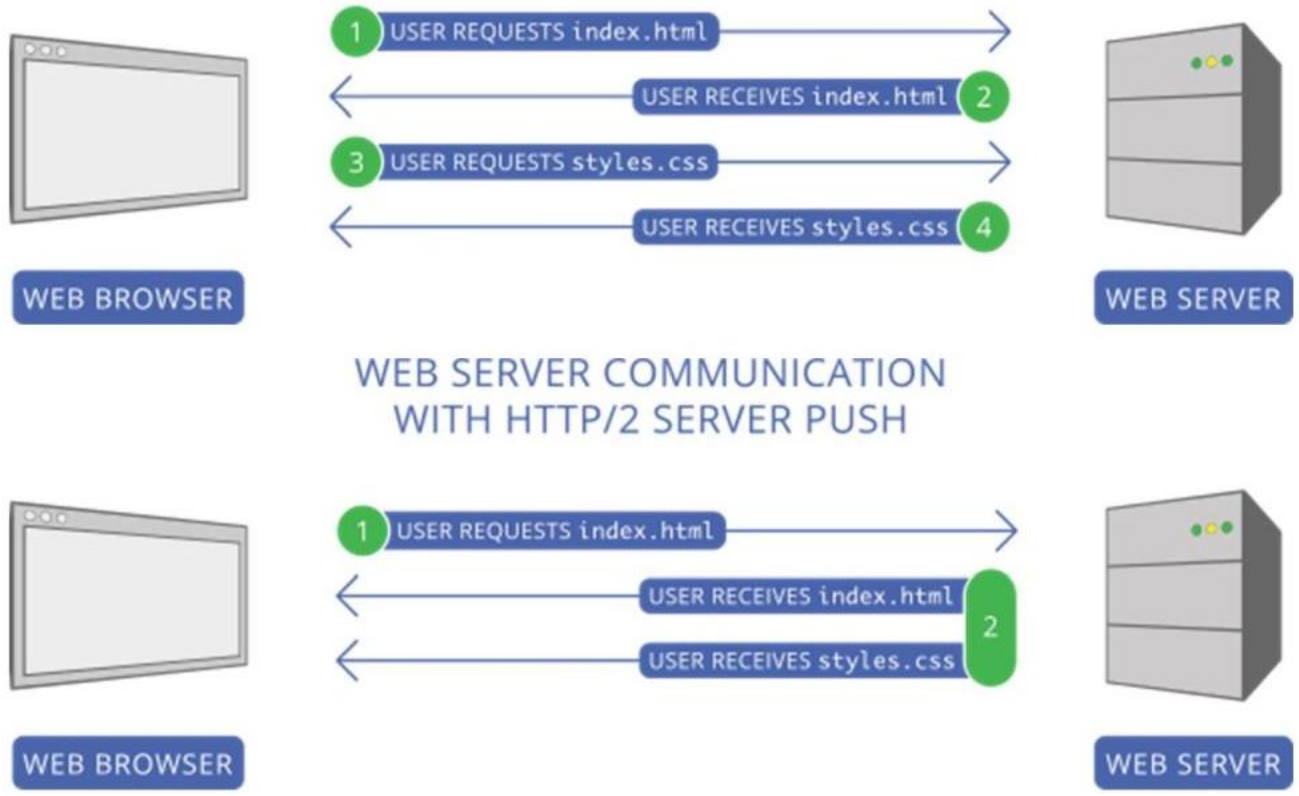
Ich bin eingeklappt

Lorem ipsum dolor sit amet

Lightbox

Hack 3: HTTP2 Serverpush

Nutzung v. Link-Element m. rel=„preload“ im HTTP-Header



Quelle: https://www.web-netz.de/blog/http2-die-grossen-vorteile-fuer-seo/

Hack 3: HTTP2 Serverpush

```
<FilesMatch "\.html$">
    Header add Link "<https://performance.seo-atelier.com/test/demo/css/bootstrap.min.css>; rel=preload; as=style"
    Header add Link "<https://performance.seo-atelier.com/test/demo/css/style.min.css>; rel=preload; as=style"
</FilesMatch>
```

HTTP Response-Header

▼ **Response Headers**

accept-ranges: bytes

cache-control: max-age=5

content-encoding: gzip

content-length: 2024

content-type: text/html

date: Mon, 17 Feb 2020 12:25:21 GMT

etag: "30bb-59ec4a50e9af8-gzip"

expires: Wed, 19 Feb 2020 12:25:21 GMT

last-modified: Mon, 17 Feb 2020 12:25:03 GMT

link: <https://performance.seo-atelier.com/test/demo/bootstrap/css/bootstrap.min.css>; rel=preload; as=style

link: <https://performance.seo-atelier.com/test/demo/css/style.min.css>; rel=preload; as=style

push-policy: default

server: Apache

status: 200

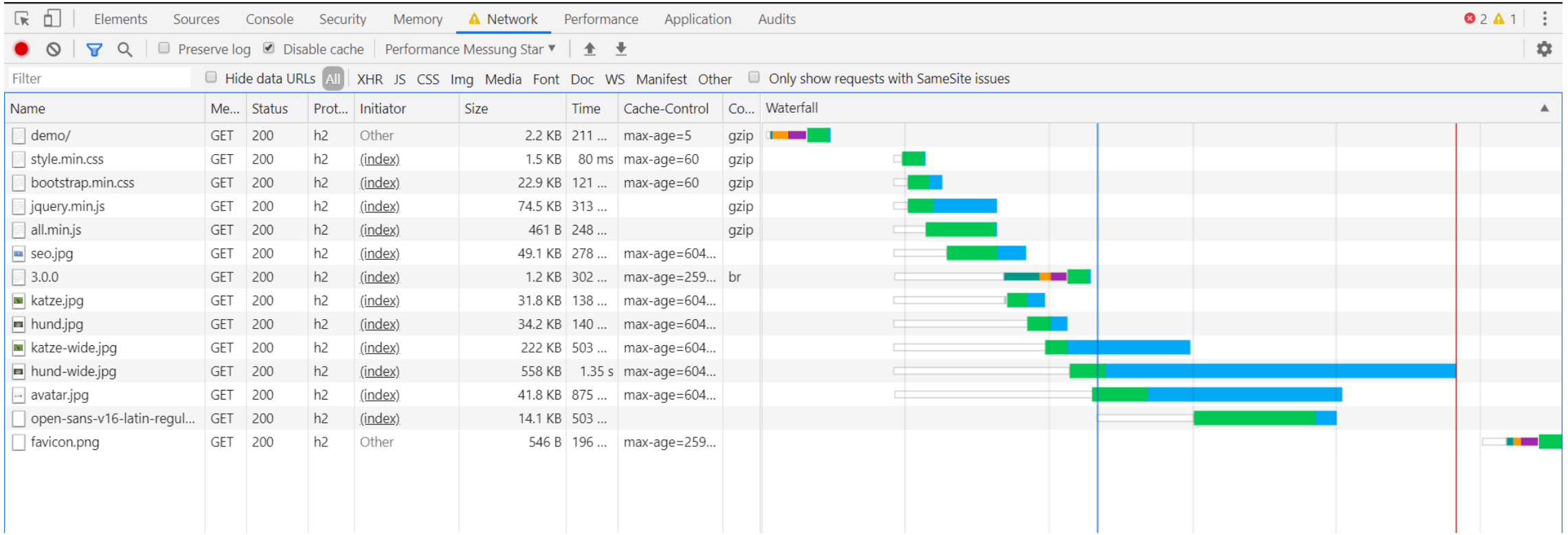
strict-transport-security: max-age=600000

vary: Accept-Encoding

x-robots-tag: noindex, follow

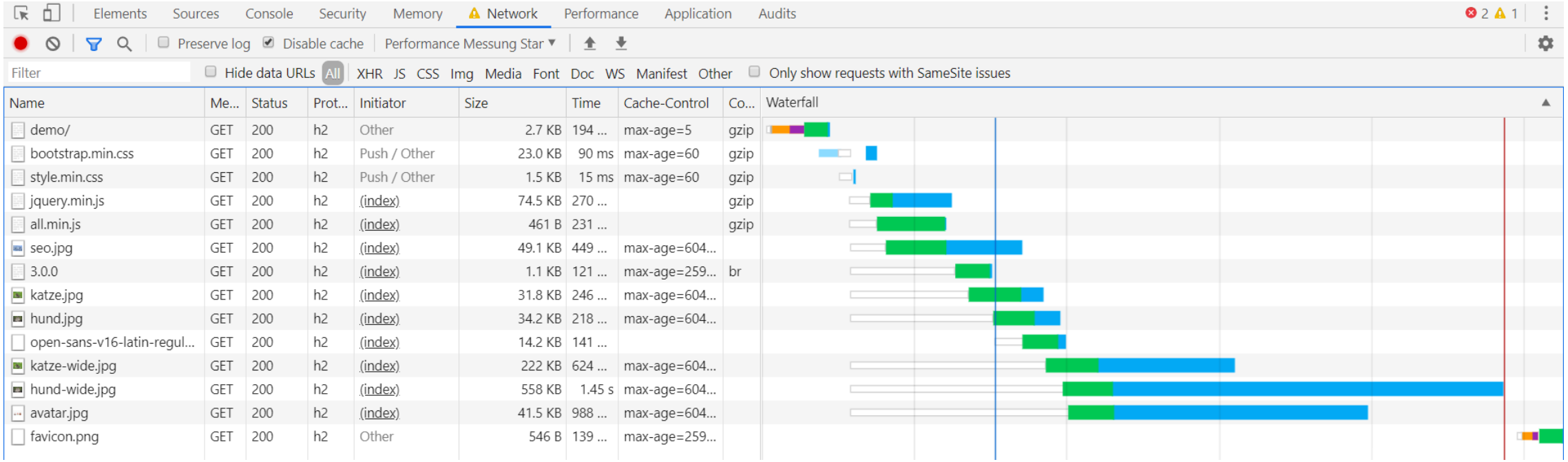
Hack 3: HTTP2 Serverpush

Ohne Preload



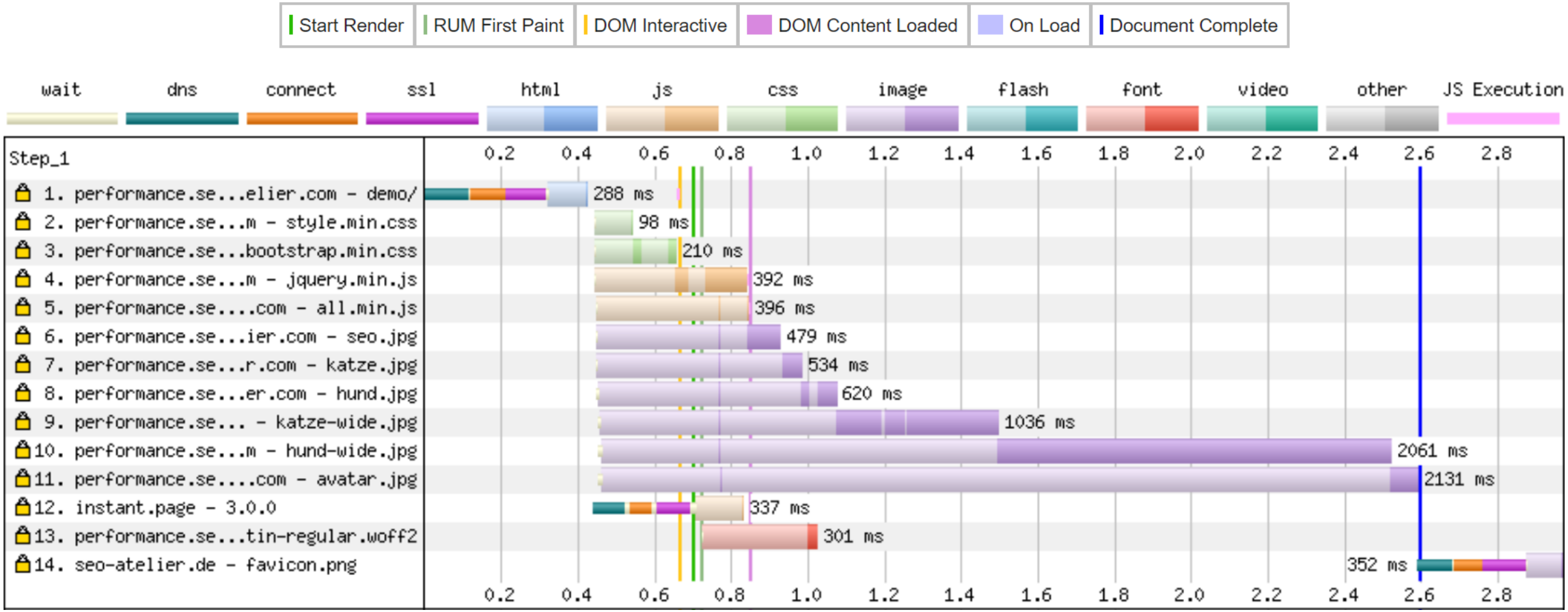
Hack 3: HTTP2 Serverpush

Mit Preload



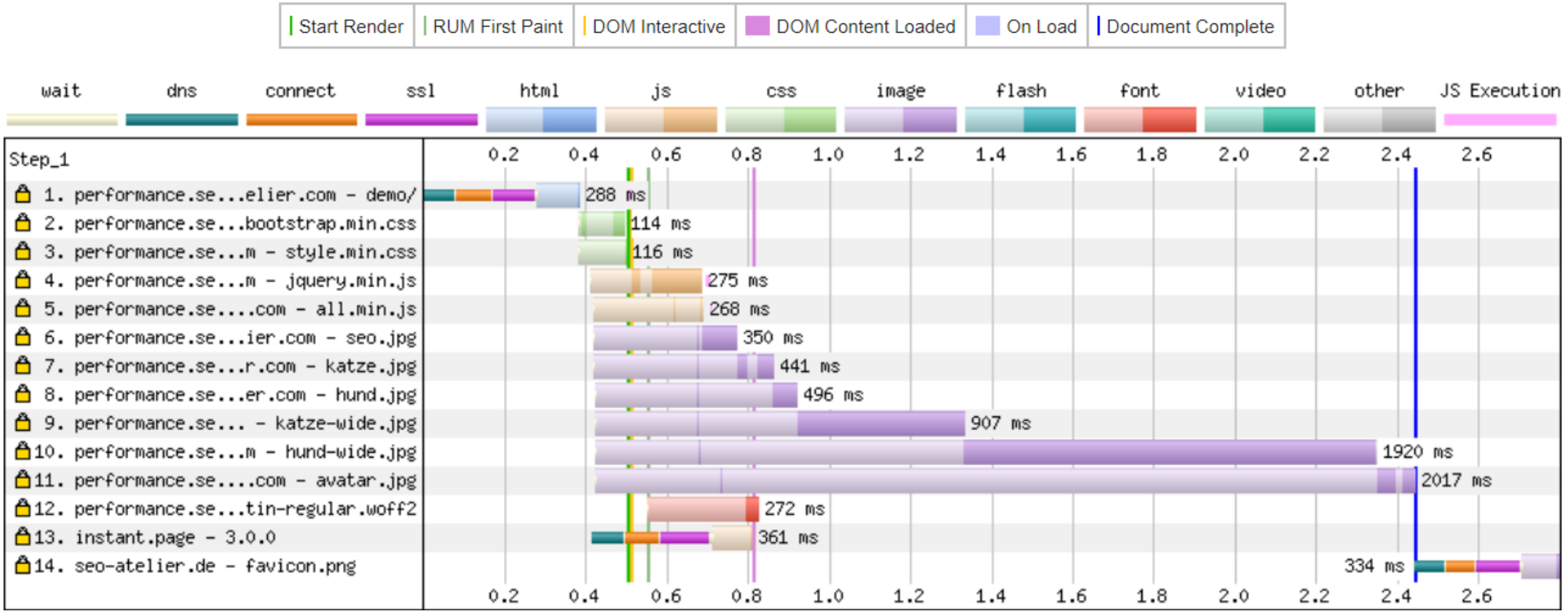
Hack 3: HTTP2 Serverpush

Ohne Preload



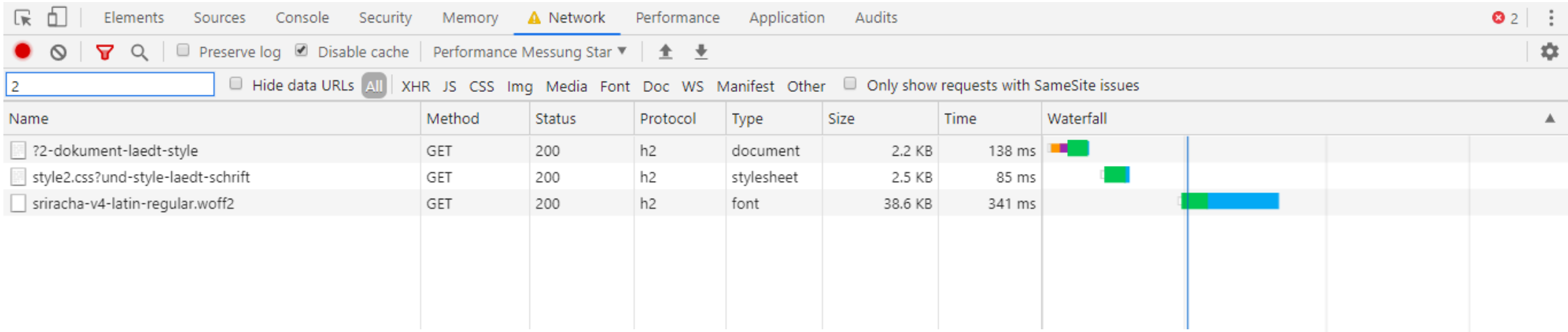
Hack 3: HTTP2 Serverpush

Mit Preload



Hack 4: Schriften

Das Problem mit den Schriften



Avoid chaining critical requests — 2 chains found

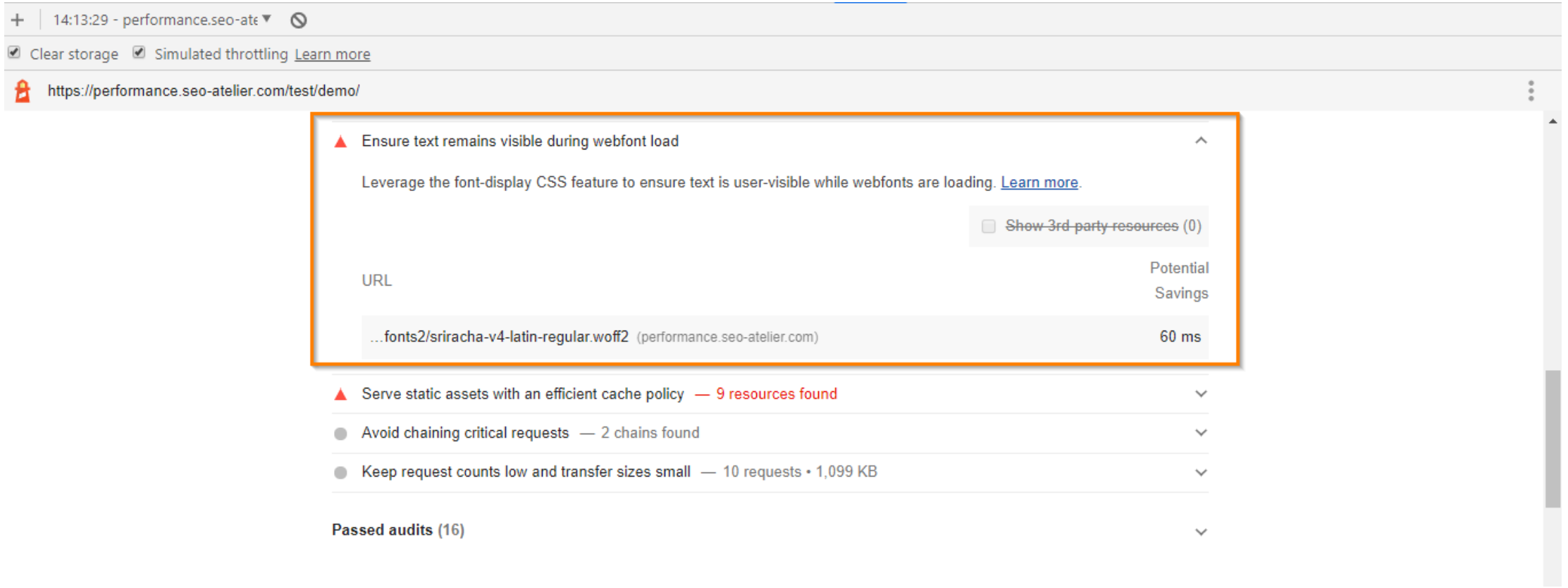
The Critical Request Chains below show you what resources are loaded with a high priority. Consider reducing the length of chains, reducing the download size of resources, or deferring the download of unnecessary resources to improve page load. Learn more.

Maximum critical path latency: **220 ms**

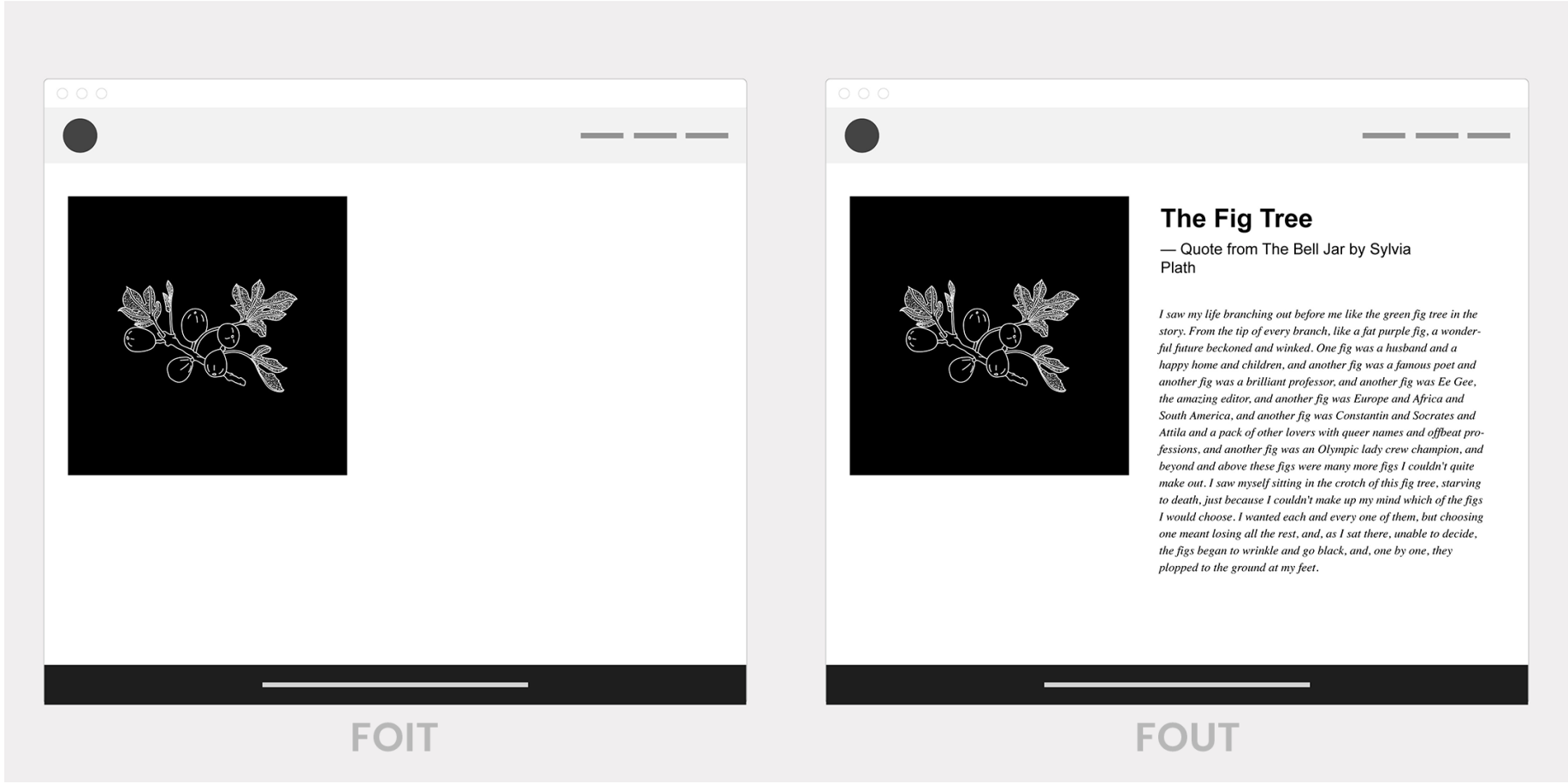
Initial Navigation

- /test/demo/ (performance.seo-atelier.com)
  - …css/style.css (performance.seo-atelier.com) - **30 ms, 2.54 KB**
  - …fonts2/sriracha-v4-latin-regular.woff2 (performance.seo-atelier.com) - **60 ms, 72.07 KB**

Hack 4: Schriften



Hack 4: Schriften - FOIT vs. FOUT WTF



Quelle: https://www.malthemilthers.com/font-loading-strategy-acceptable-flash-of-invisible-text/

Hack 4: Schriften

```
/* sriracha-regular - latin */
@font-face {
  font-family: 'Sriracha';
  font-style: normal;
  font-weight: 400;
  src: url('../fonts2/sriracha-v4-latin-regular.eot'); /* IE9 Compat Modes */
  src: local('Sriracha'), local('Sriracha-Regular'),
       url('../fonts2/sriracha-v4-latin-regular.eot?#iefix') format('embedded-opentype'), /* IE6-IE8 */
       url('../fonts2/sriracha-v4-latin-regular.woff2') format('woff2'), /* Super Modern Browsers */
       url('../fonts2/sriracha-v4-latin-regular.woff') format('woff'), /* Modern Browsers */
       url('../fonts2/sriracha-v4-latin-regular.ttf') format('truetype'), /* Safari, Android, iOS */
       url('../fonts2/sriracha-v4-latin-regular.svg#Sriracha') format('svg'); /* Legacy iOS */
        font-display: swap; /* Verhalten, wenn Schrift noch nicht geladen */
}
```

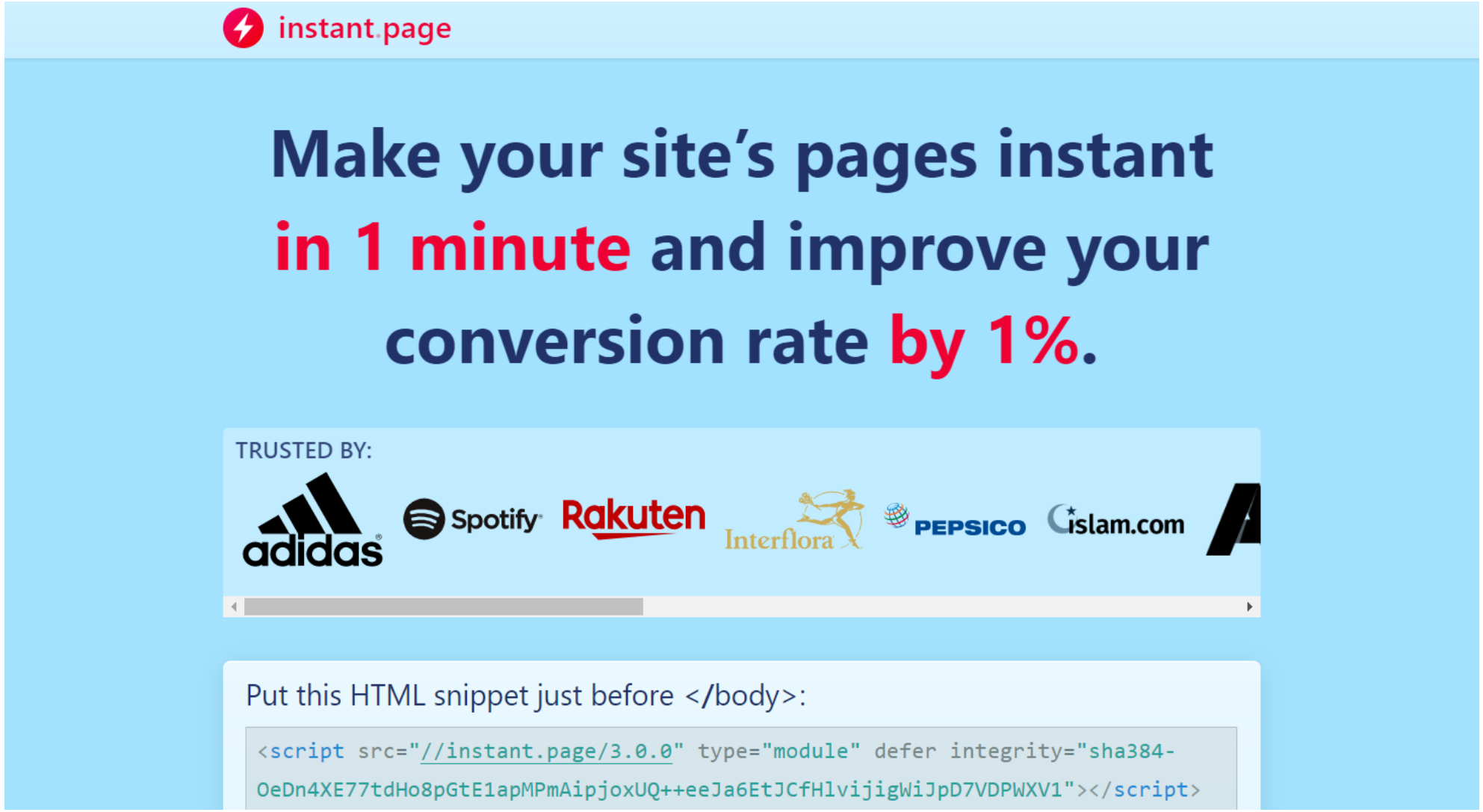
auto	Standardmethode des Browsers, meist wie/ähnlich „block“.
block	Text kurz auszublenden, bis die Schriftart vollständig heruntergeladen wurde. →FOIT
swap	Text mithilfe v Fallback-Schriftart anzeigen bis Schriftart heruntergeladen ist. → FOUT
fallback	Kompromiss zwischen „auto“ & „swap“. Browser blendet Text für ca. 100 ms aus & verwendet Fallback-Schrift. Nach Download wird auf neue Schriftart umgestellt, wenn innerhalb v. 3 Sekunden verfügbar.
optional	Text wird ausgeblendet, dann wird zur Fallback-Schrift übergegangen bis Schriftart verfügbar. Aber: Browser darf – je nach Verbindungsgeschwindigkeit - selbst entscheiden ob Schriftart überhaupt verwendet wird.

Hack 4: Schriften

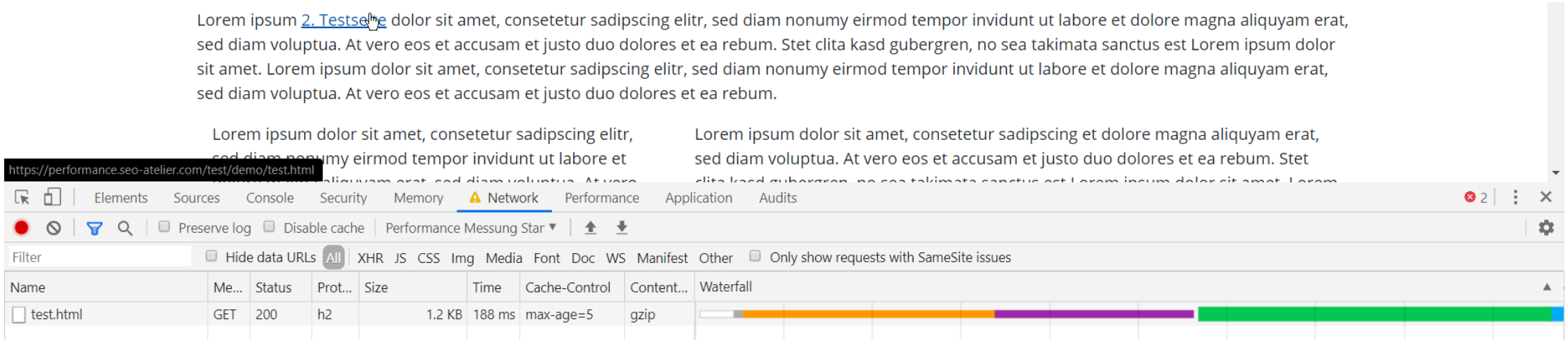
Tipp: Um FOUT nicht allzu grausam wirken zu lassen, empfiehlt sich ein Preload aus dem HTML-Head:

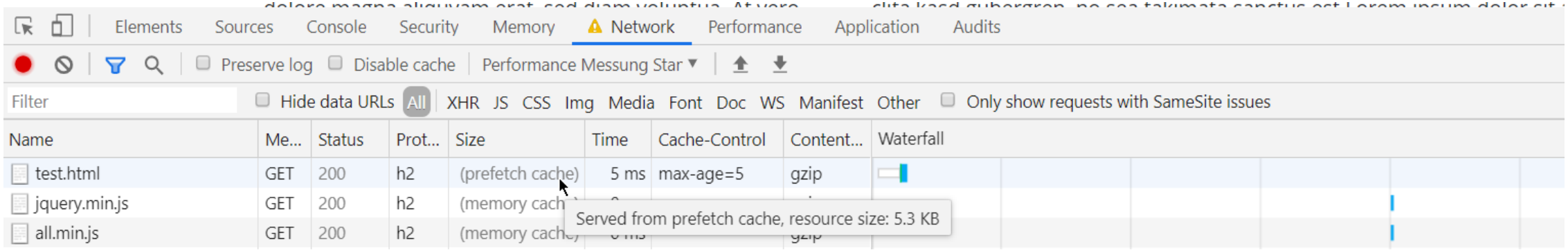
```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no" />
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <meta name="robots" content="none"/>
    <link href="css/style.css" rel="stylesheet" />
    <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet" />
    <link rel="preload" href="/fonts/open-sans-v16-latin-700.woff2" as="font" crossorigin="anonymous"/>
```

Hack 5: instant.page



Hack 5: instant.page

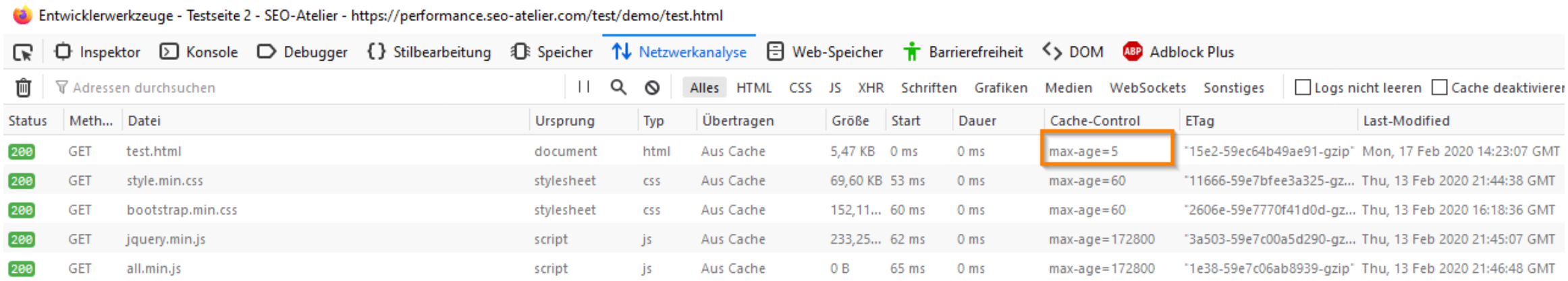




Hack 5: instant.page

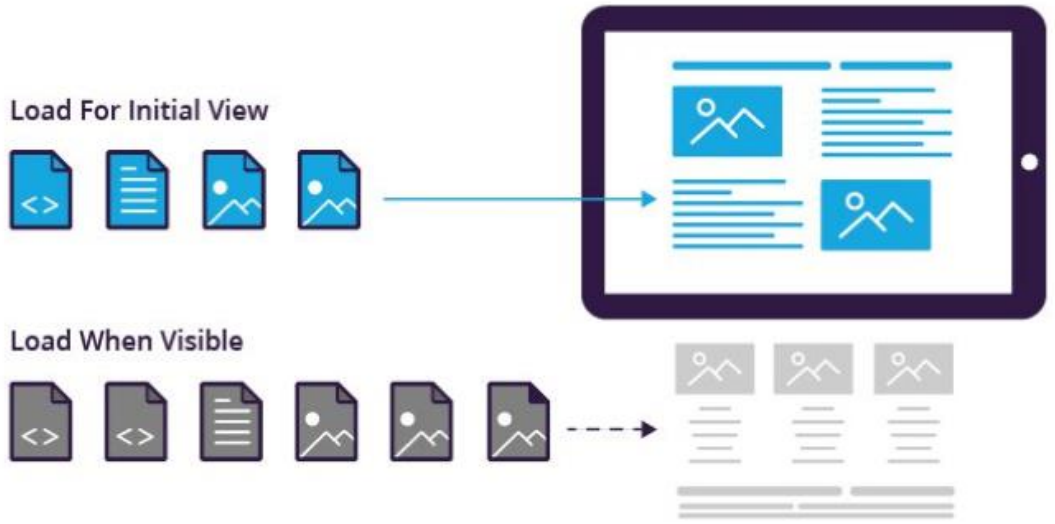
FF-Kompatibilität

```
############################################
<IfModule mod_headers.c>
    <FilesMatch "\.(html)$">
        Header set Cache-Control "max-age=5"
    </FilesMatch>
</IfModule>
```



Hack 6: Loading=„lazy“

Lazy Loading ohne JS, intersection-Observer, usw.




```
<img class="mx-auto d-block" src="img/7293234823946/seo.jpg" alt="SEO-Testbild 1" title="Bild 1" />
```

```
<img class="mx-auto d-block" src="img/7293234823946/seo.jpg" alt="SEO-Testbild 1" title="Bild 1" loading="lazy" />
```

Quelle: https://www.imperva.com/learn/performance/lazy-loading/

Hack 6: Stale-while-revalidate

Kein Fingerprint: Hohe Aktualität & effizientes Caching

Dateien wurden trotz abgelaufener TTL direkt aus dem Cache bezogen

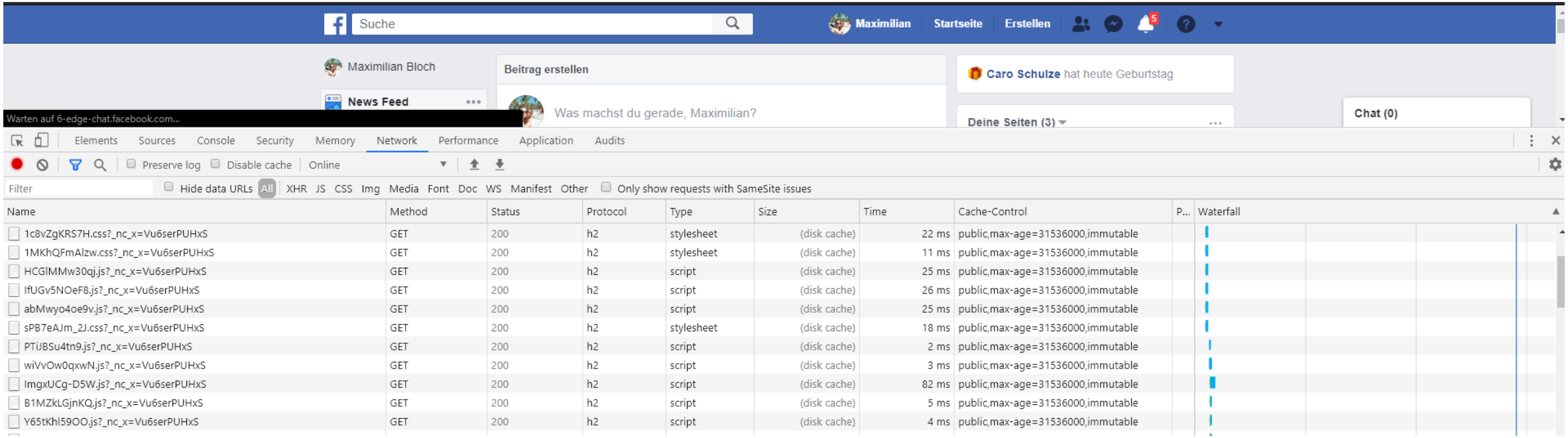
Name	Method	Status	Protocol	Domain	Type	Size	Time	Cache-Control	ETag	Last-Modified	Waterfall
etag/	GET	304	h2	performance.seo-atelier.c...	document	41 B	80 ms	max-age=20, immutable	"13b9-59a511c9e09b4"	Sun, 22 Dec 2019 20:54:06 GMT	
style.css	GET	200	h2	performance.seo-atelier.c...	stylesheet	(memory cache)	0 ms	max-age=20, stale-while-revalidate=30	"ff-59ad7ff501f8d"		
bootstrap.min.css	GET	200	h2	performance.seo-atelier.c...	stylesheet	(memory cache)	0 ms	max-age=20, stale-while-revalidate=30	"235ed-59a4be94f3c5e"		
seo.jpg	GET	200	h2	performance.seo-atelier.c...	jpeg	(memory cache)	0 ms	max-age=60, public	"c26f-59a4c17c8dc02"		
avatar.jpg	GET	200	h2	performance.seo-atelier.c...	jpeg	(memory cache)	0 ms	max-age=60, public	"a5bc-59a4c1e5ab4fd"		
style.css	GET	304	h2	performance.seo-atelier.c...	text/css	92 B	67 ms	max-age=20, stale-while-revalidate=30	"ff-59ad7ff501f8d"		
bootstrap.min.css	GET	304	h2	performance.seo-atelier.c...	text/css	42 B	69 ms	max-age=20, stale-while-revalidate=30	"235ed-59a4be94f3c5e"		
favicon.png	GET	200	h2	seo-atelier.de	png	438 B	61 ms	max-age=2592000	"184-57160bad6ed40"	Thu, 19 Jul 2018 21:19:09 GMT	

Erst im Anschluss erfolgte die Validierung. Beim nächsten Aufruf der Dateien würden die aktualisierte Variante verwendet werden.

Asynchroner Download - blockiert nicht das Rendering

Hack 7: immutable

Verhindert Validierung – auch bei F5



Danke

SEO-Atelier
SEO & Online Marketing

Kontakt

SEO-Atelier

Maximilian Bloch
Bachstraße 21
86420 Diedorf

https://seo-atelier.de
Tel: 0821 80735508
Mail: info@seo-atelier.de