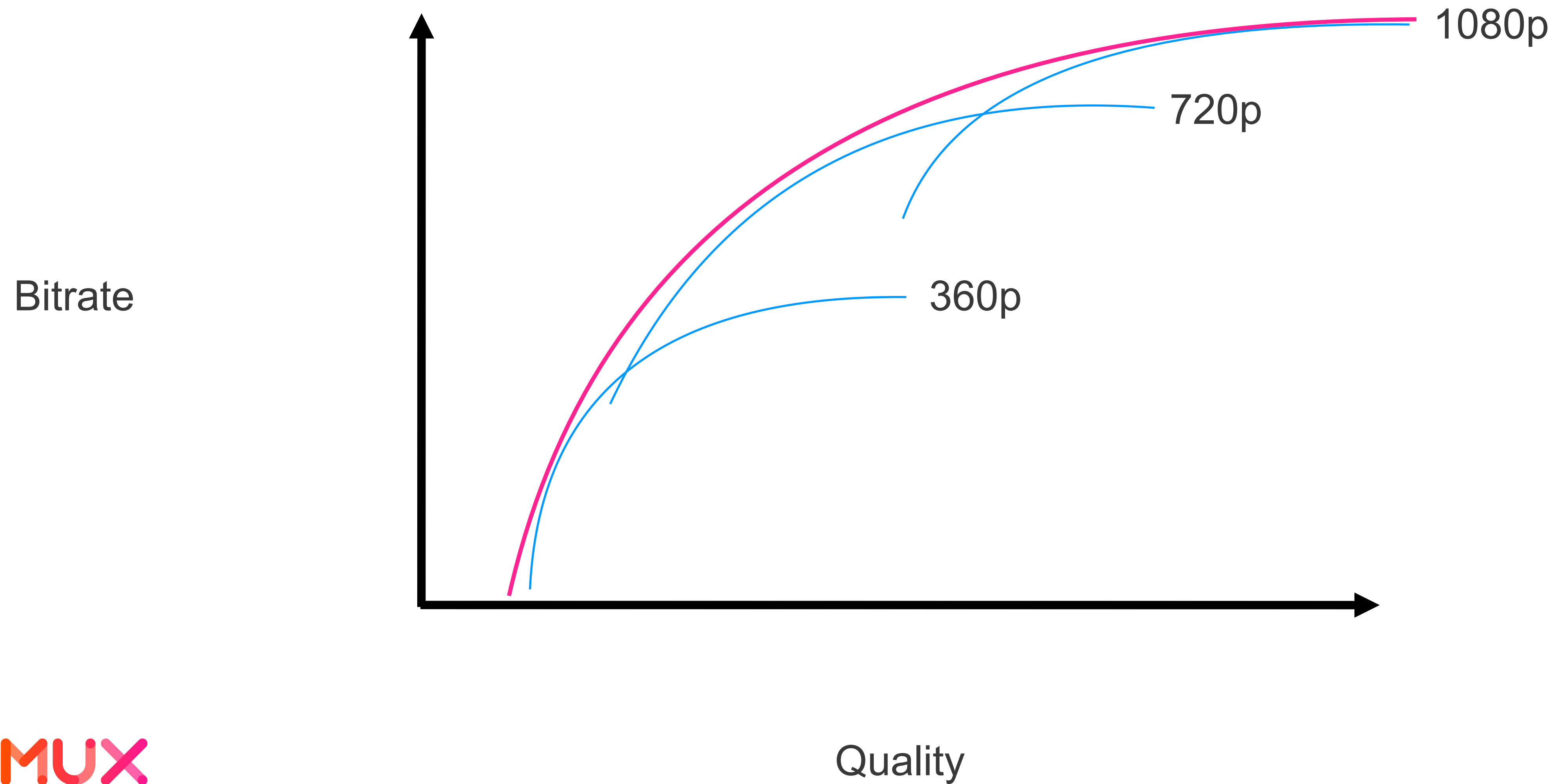
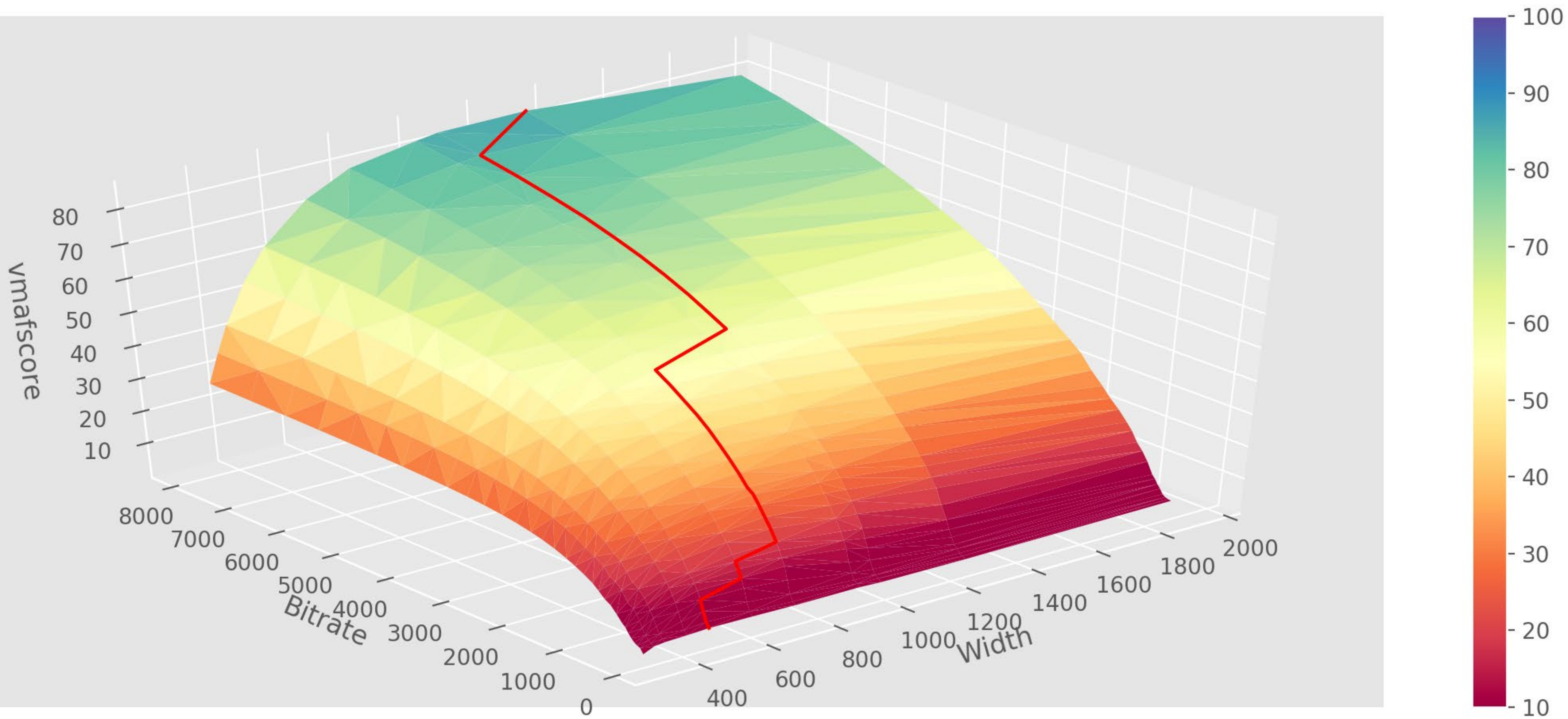




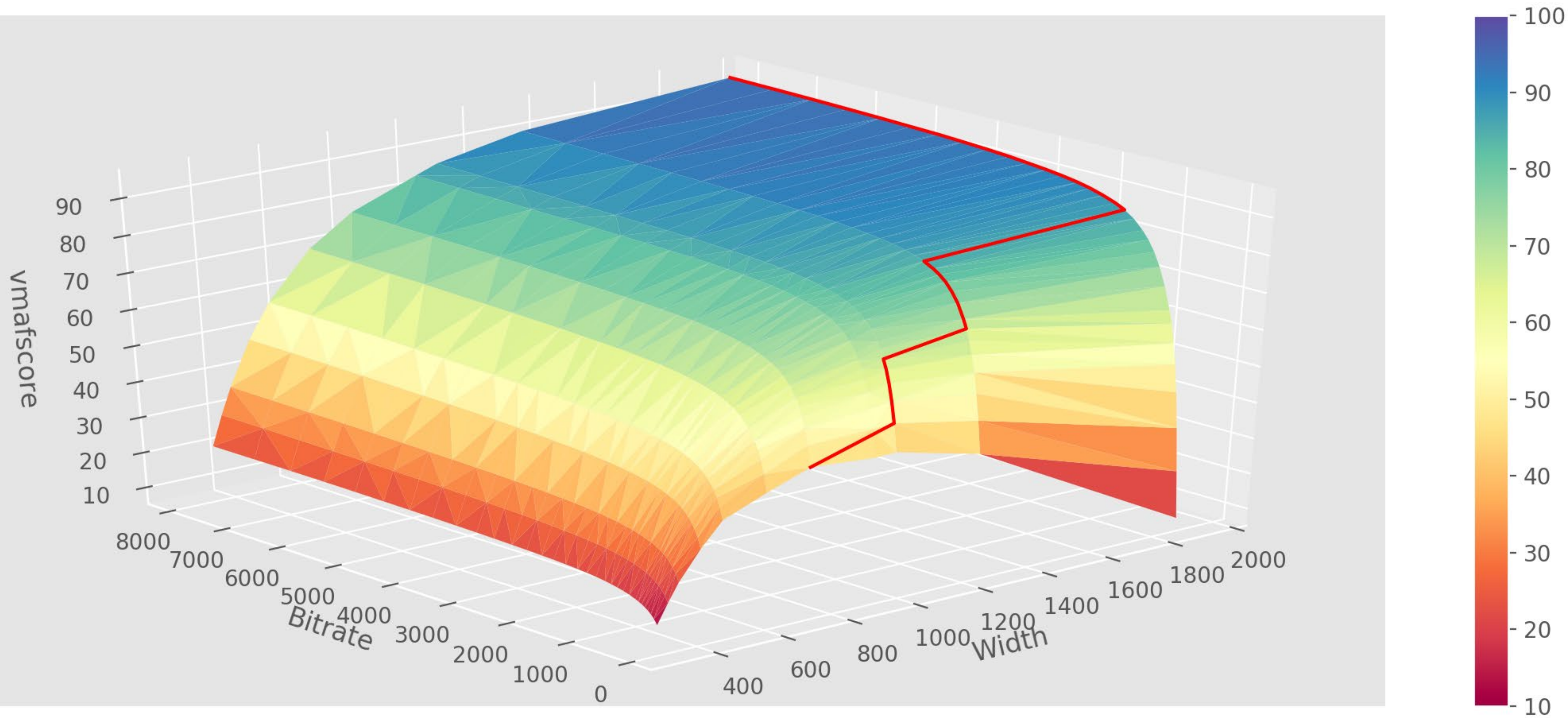
From per-title encoding to audience-adaptive encoding

Let's assume you're familiar with this already





MUX High complexity content

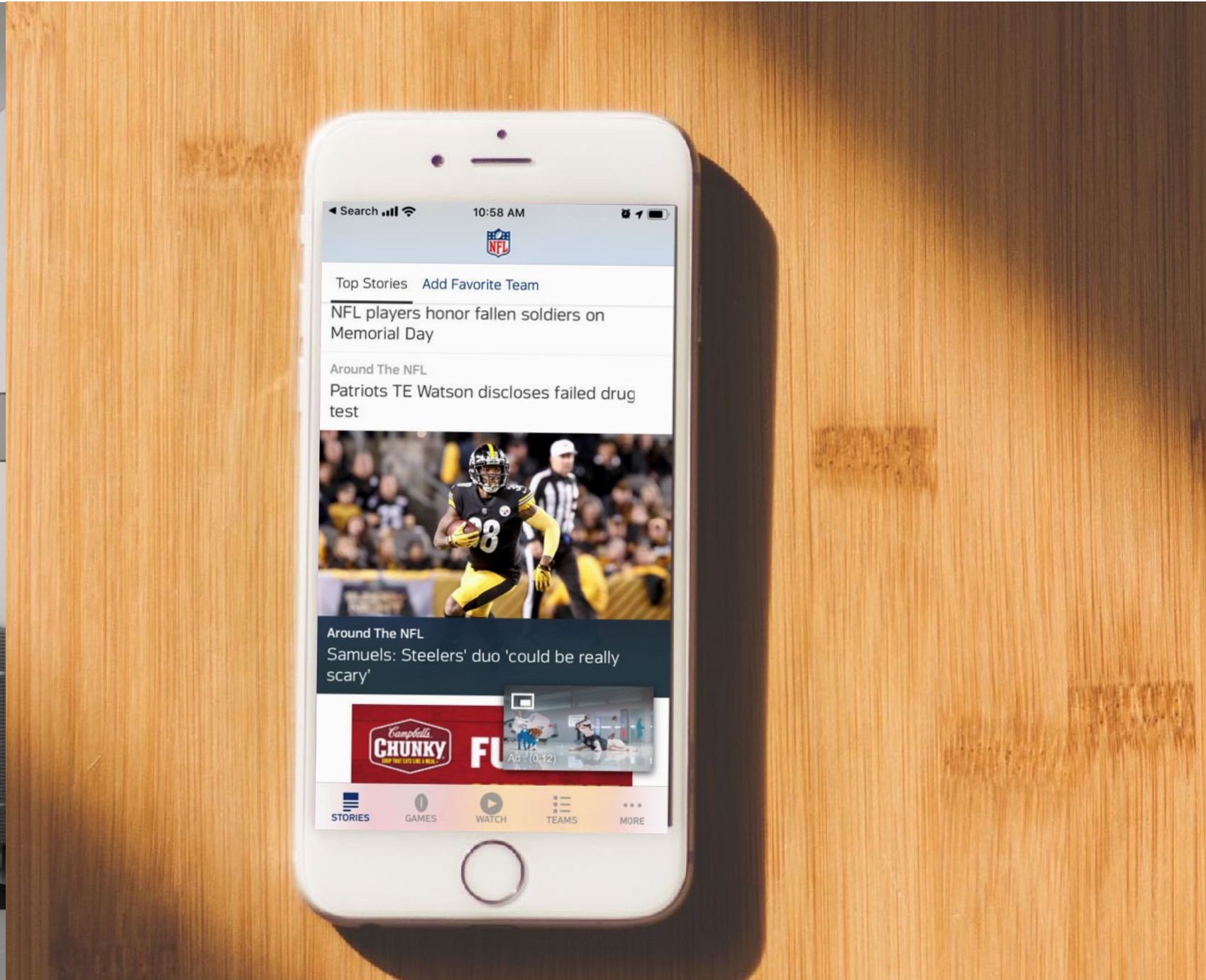


MUX Low complexity content

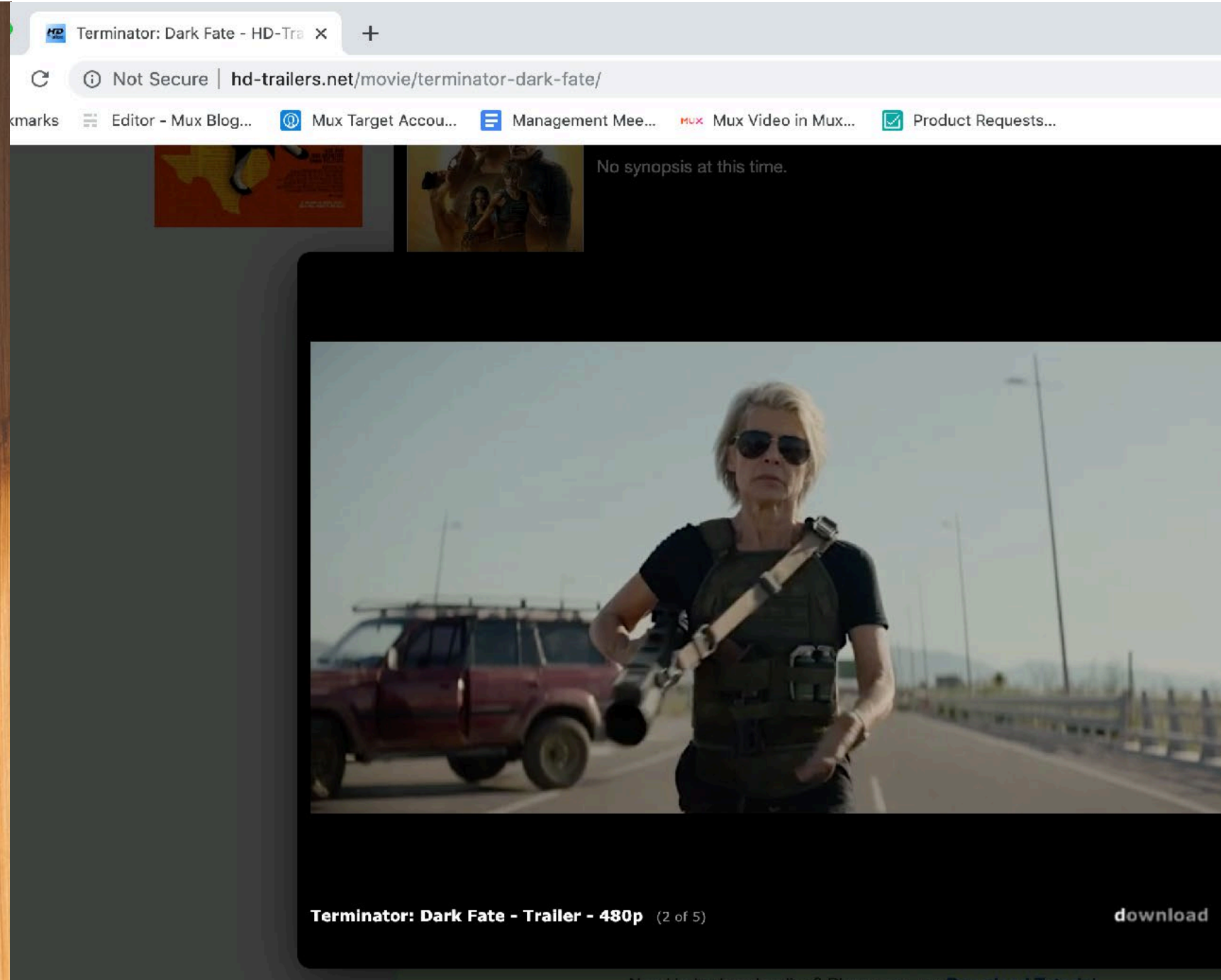
1080p display



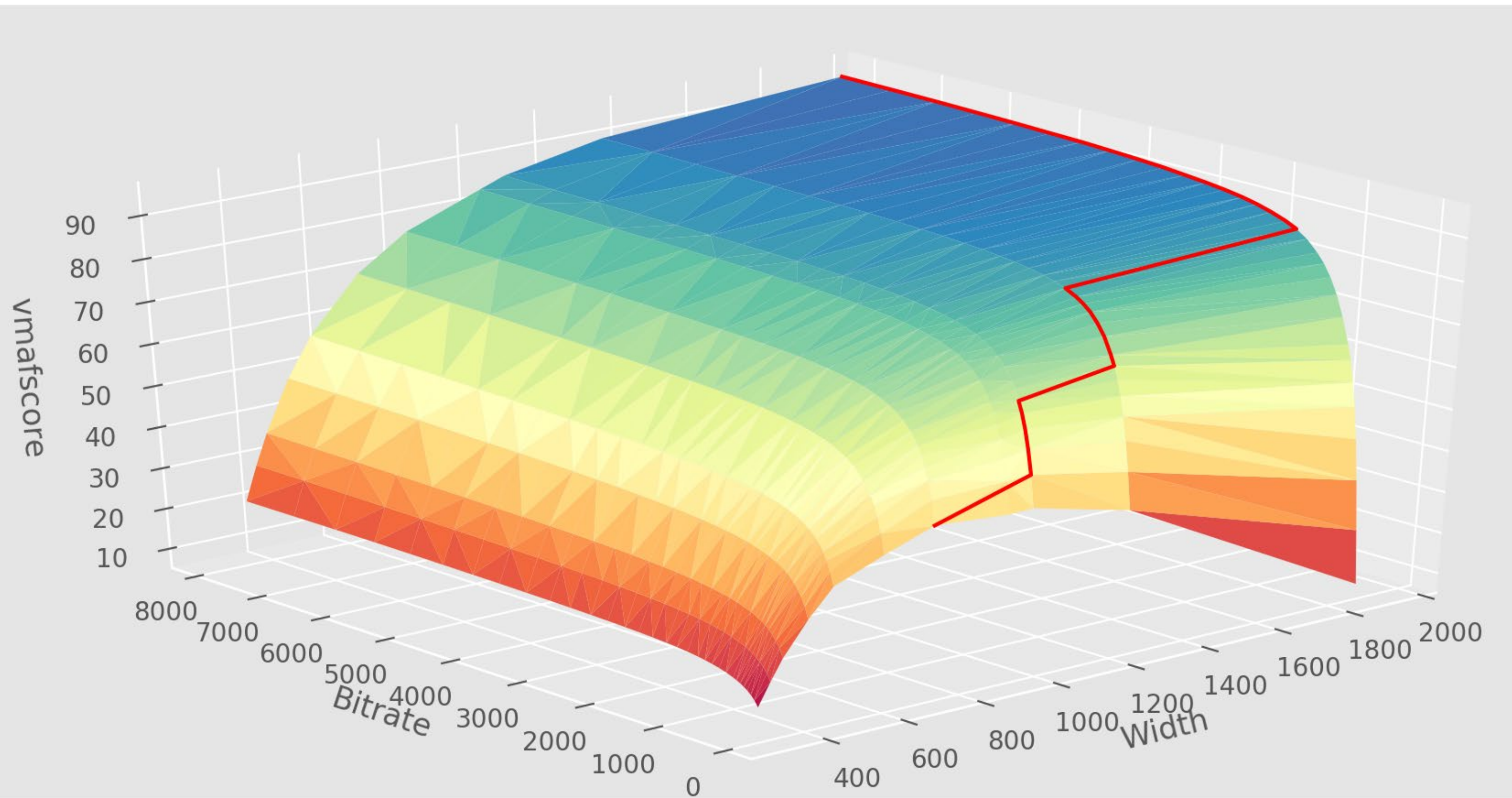
420p display



480p display



Limitation

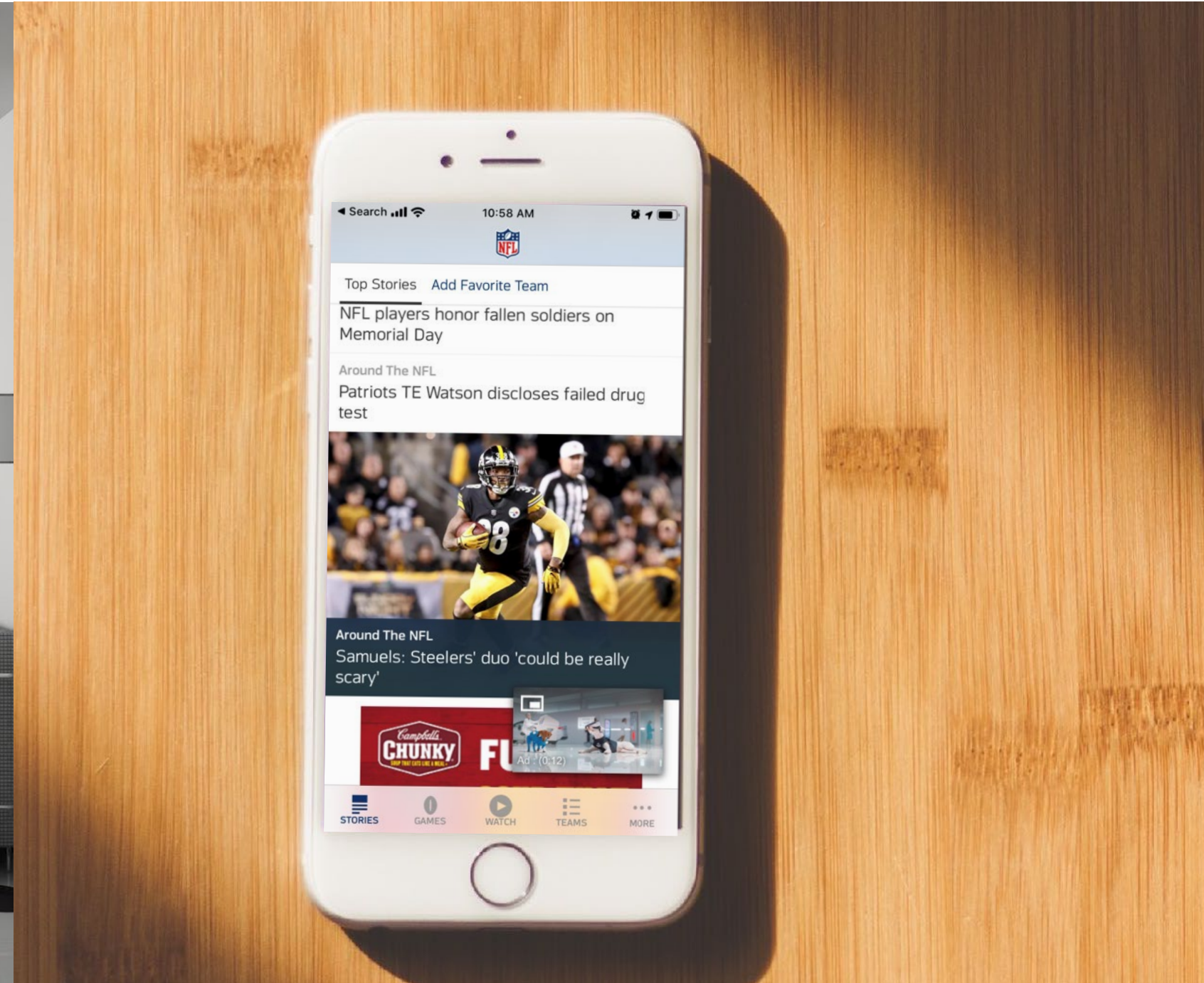


6000	1920x1080
4500	1920x1080
3000	1920x1080
2000	1920x1080
1100	1920x1080
730	1280x720
365	960x540
145	640x360

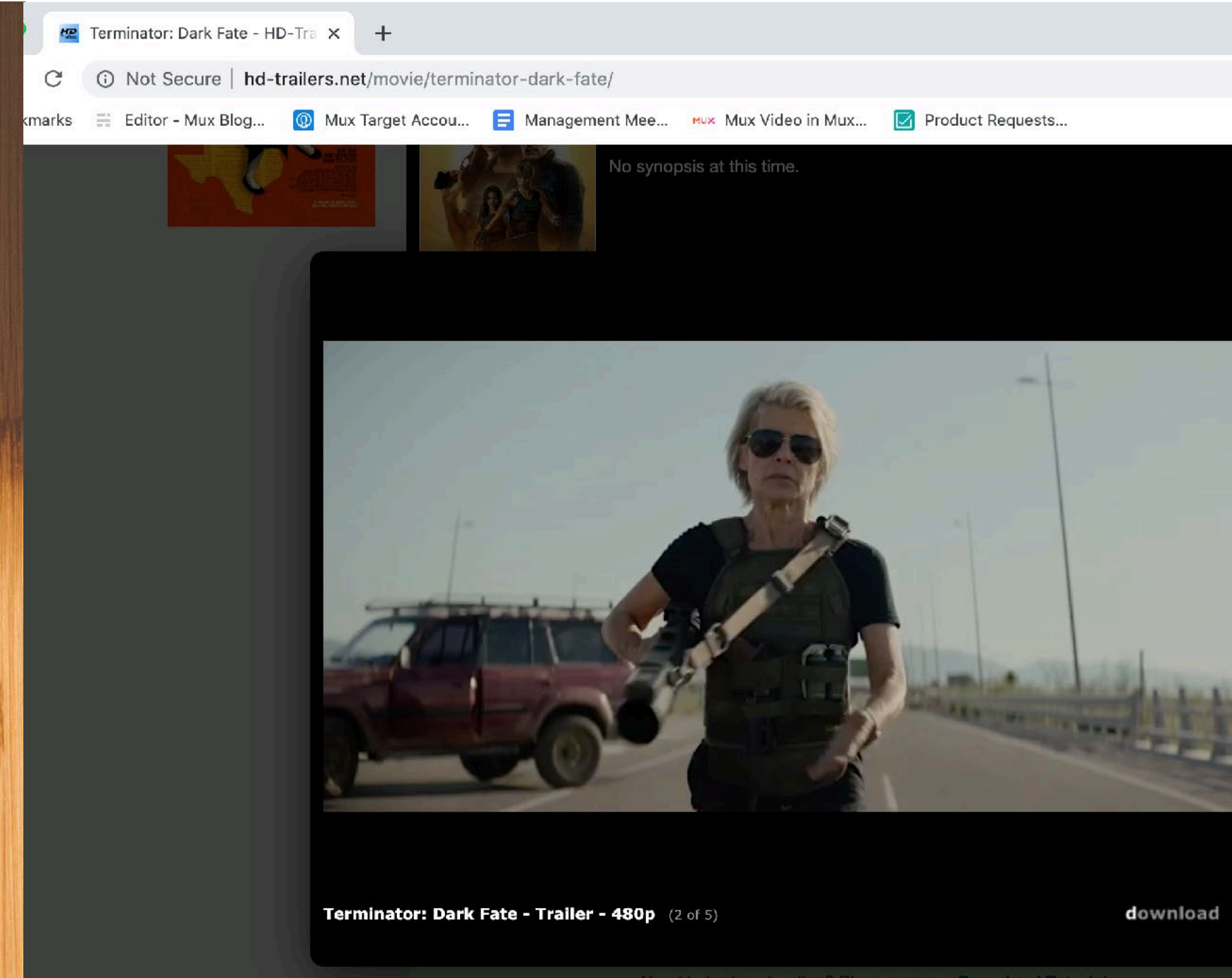
1920x1080 @ 6 Mbps
1080p display



420p display



480p display



VMAF: ~90

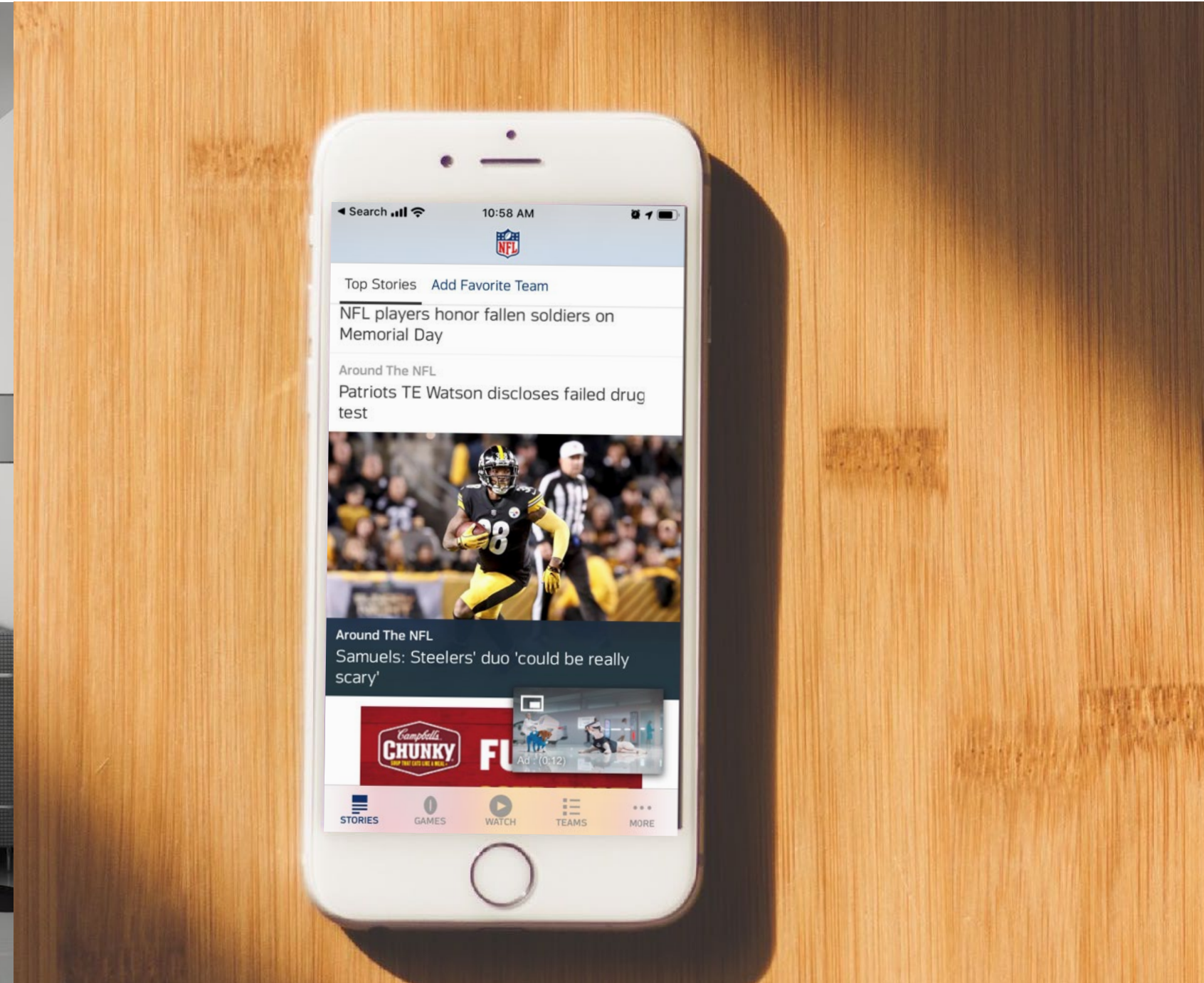


1920x1080 @ 6 Mbps
1080p display



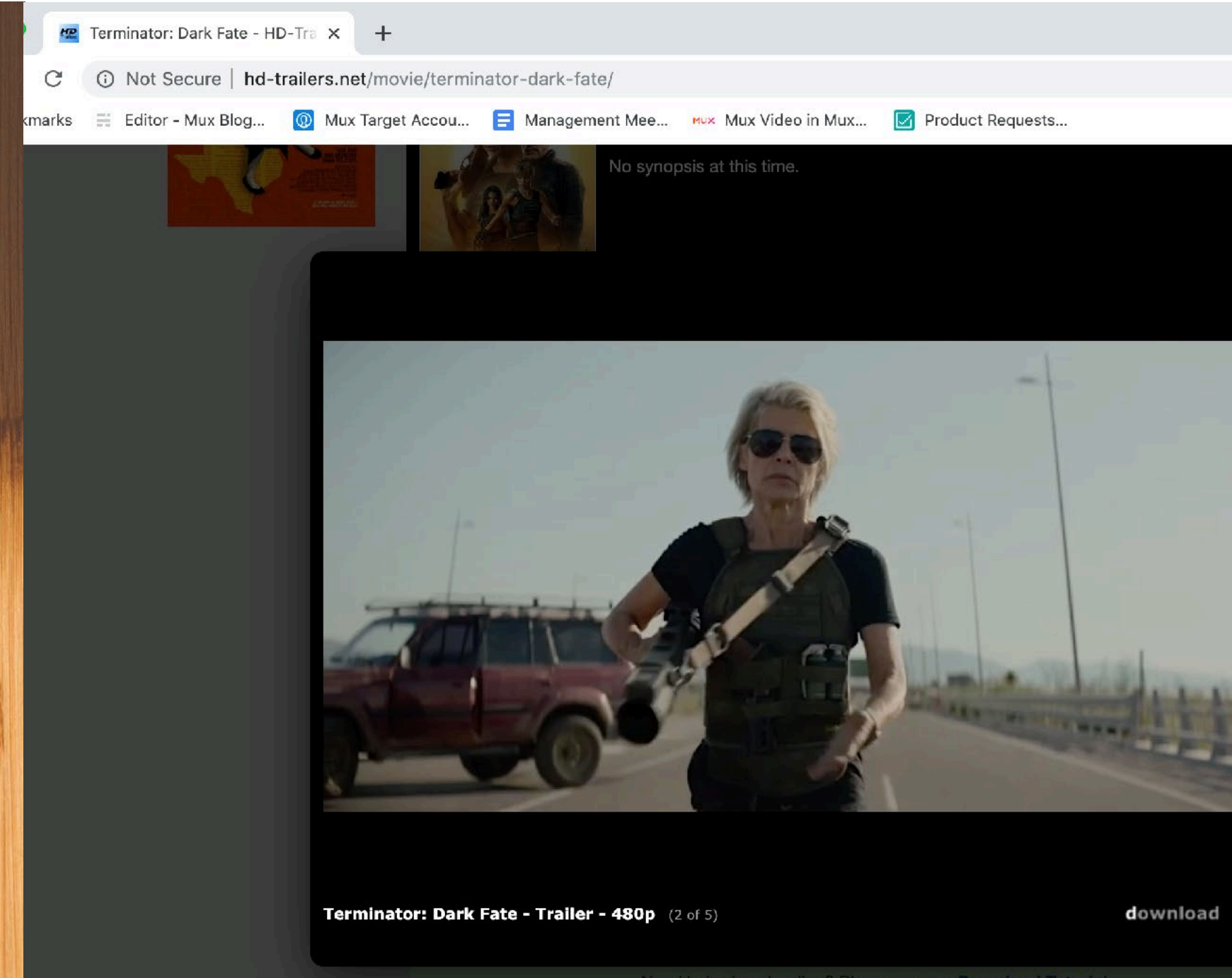
VMAF: ~90

960x540 @ 365 Kbps
420p display



VMAF: ~50

960x540 @ 365 Kbps
480p display



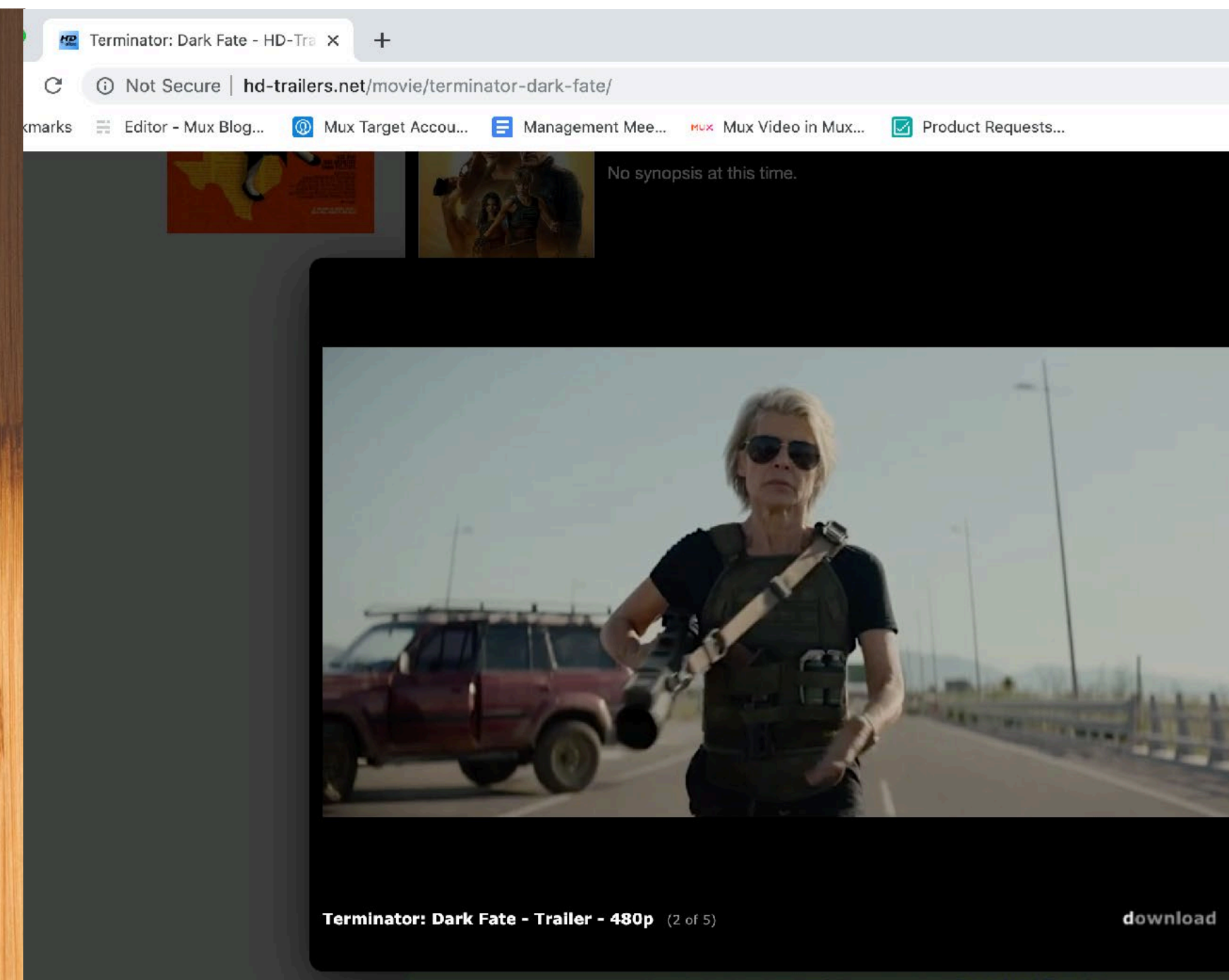
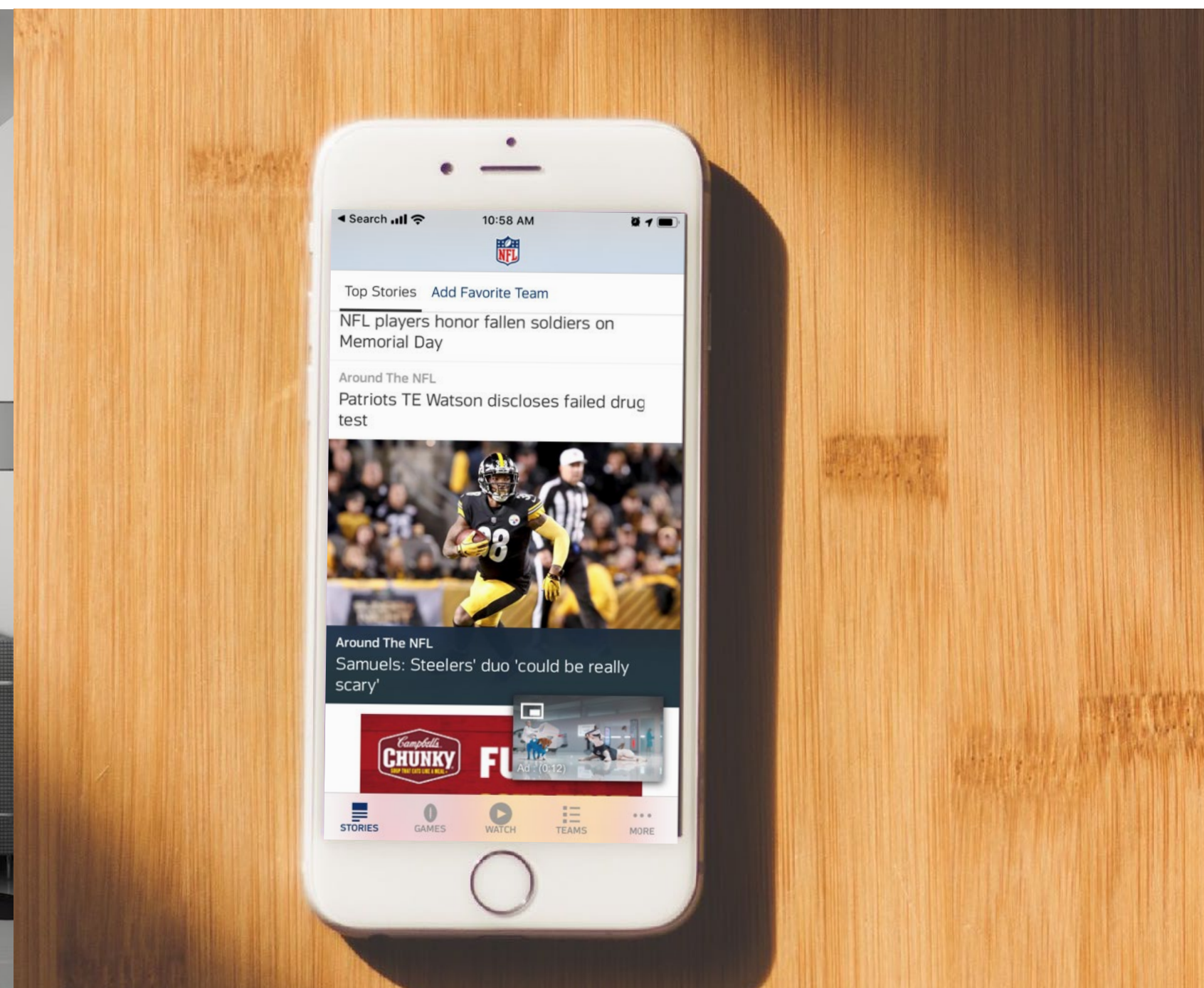
VMAF: ~50

Two options:

1. One ladder per device

2. Hybrid ladder

What if you want to apply PTE techniques to a mixed set of viewing conditions?



You have three decisions to make.



1. Max target quality

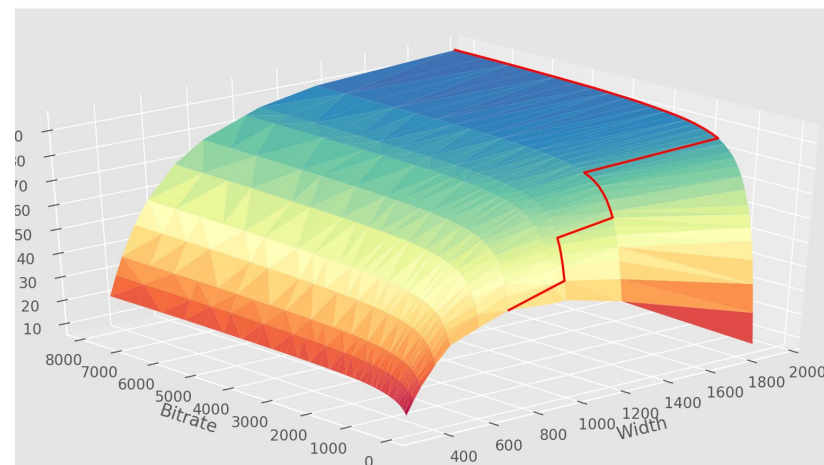


2. Min target size

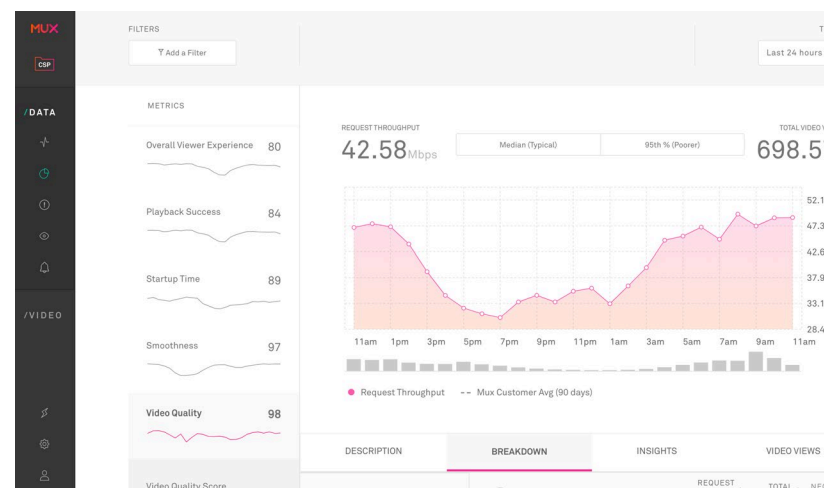


3. Number of renditions

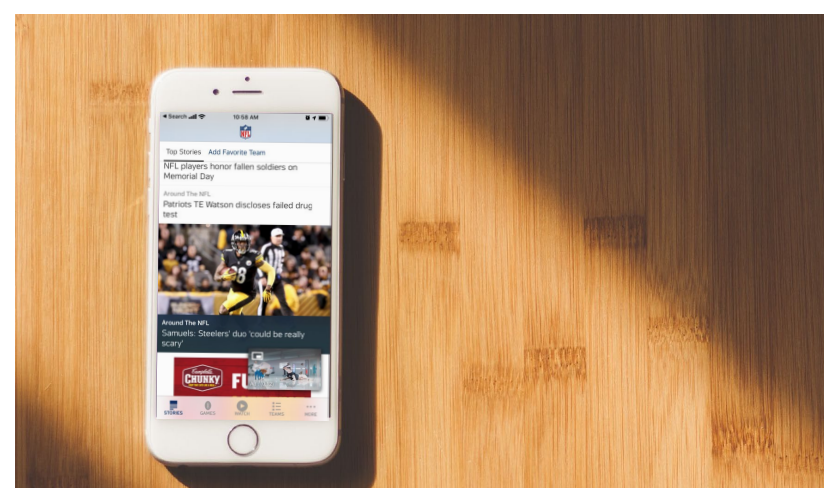
And you need three things.



4. Per-title encoding data



5. Audience bandwidth data

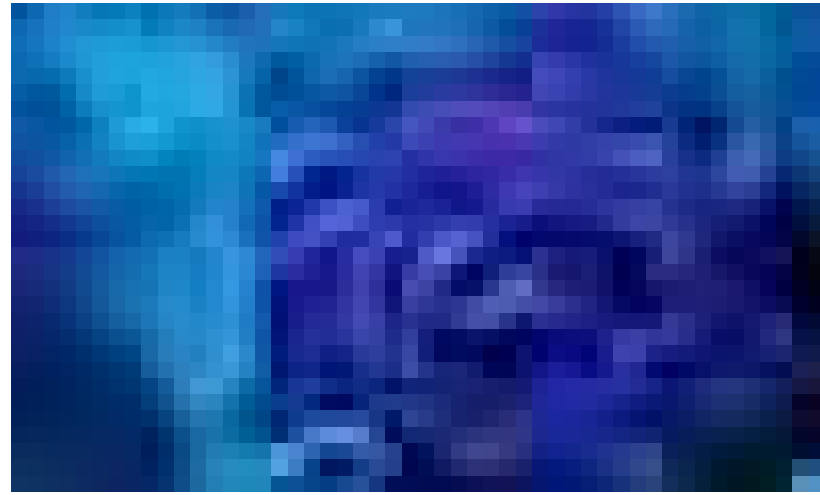


6. Audience resolution data



1. Max target quality

[VMAF 94]	[1080p]



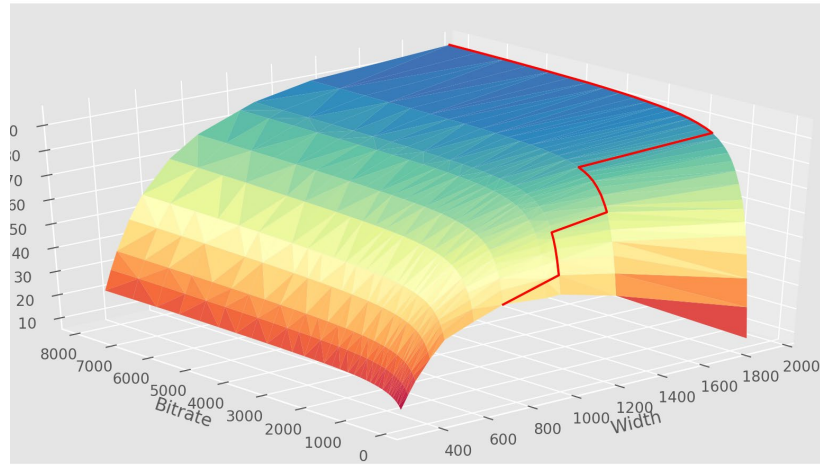
2. Min target size

[VMAF 94]	[1080p]
[2% of users]	[240p]



3. Number of renditions

1	[VMAF 94]	[1080p]
2		
3		
4		
5	[2% of users]	[240p]

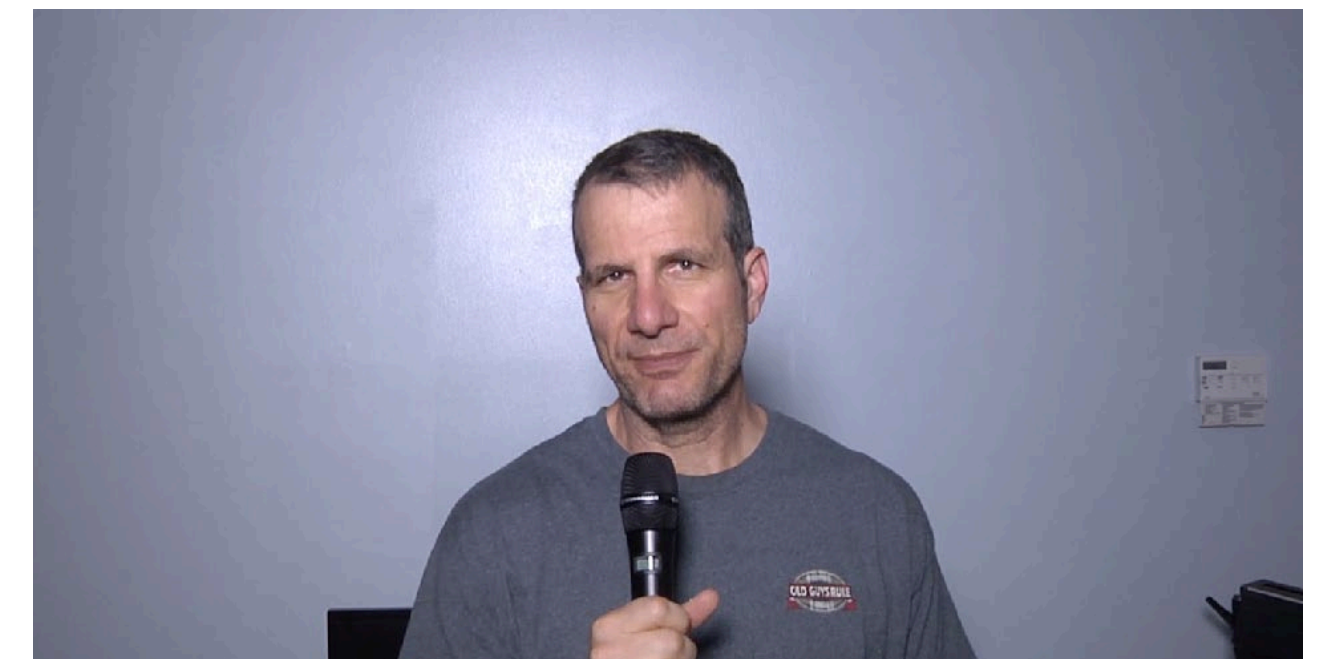


4. Per-title encoding data

1	7437	1280x720
2		
3		
4		
5	[2% of users]	[180p]

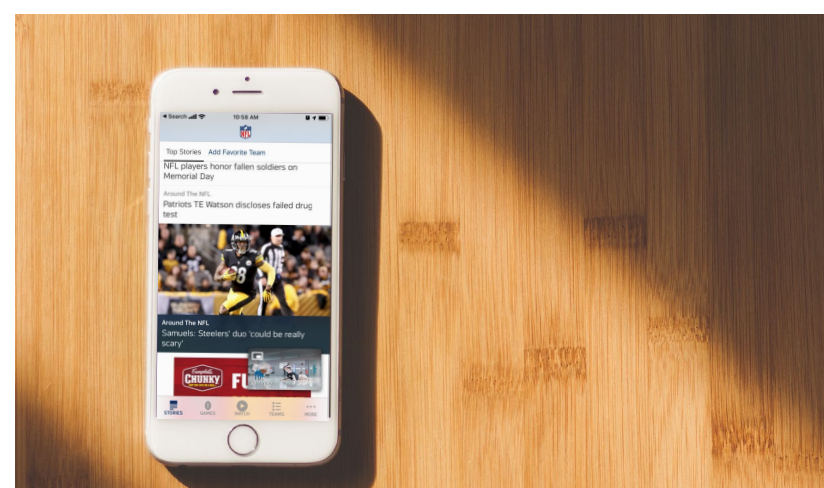
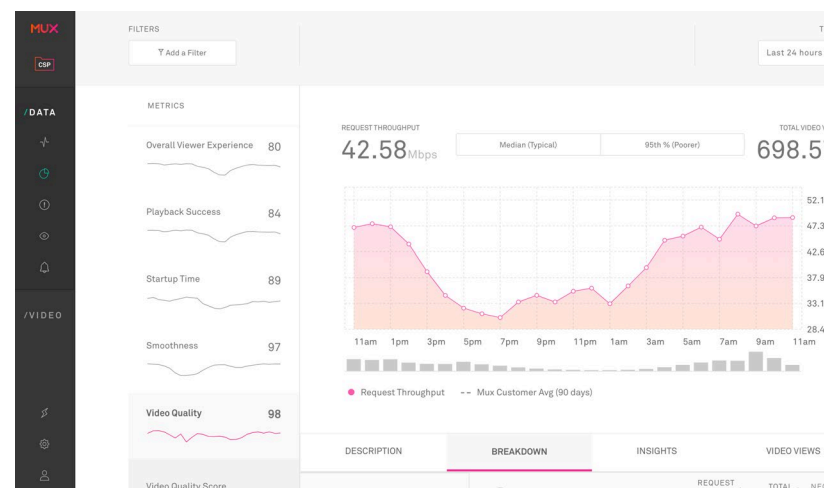
1	6500	1920x1080
2		
3		
4		
5	[2% of users]	[180p]

1	2000	1920x1080
2		
3		
4		
5	[2% of users]	[180p]



5. Audience bandwidth data

6. Audience resolution data



Add a Filter

Last 24 hours

METRICS

Overall Viewer Experience 80



Playback Success 84



Startup Time 89



Smoothness 97



Video Quality 98



REQUEST THROUGHPUT

42.58 Mbps

Median (Typical)

95th % (Poorer)

TOTAL VIDEO VIEWS

698.57



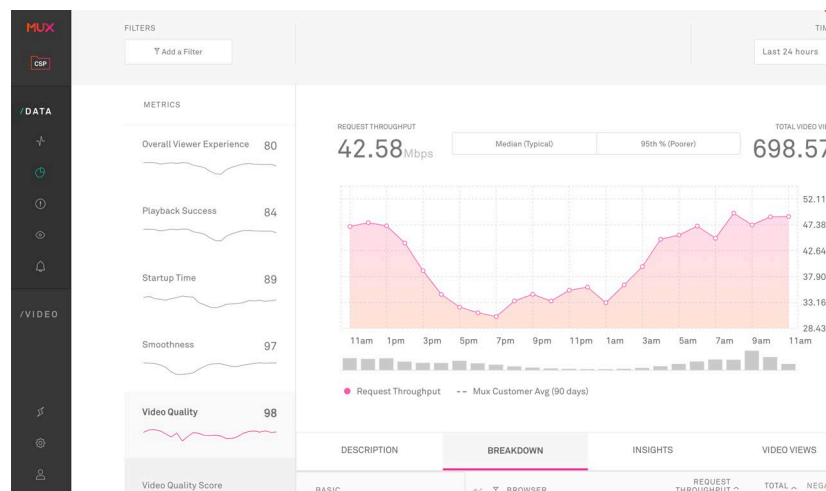
● Request Throughput - - Mux Customer Avg (90 days)

DESCRIPTION

BREAKDOWN

INSIGHTS

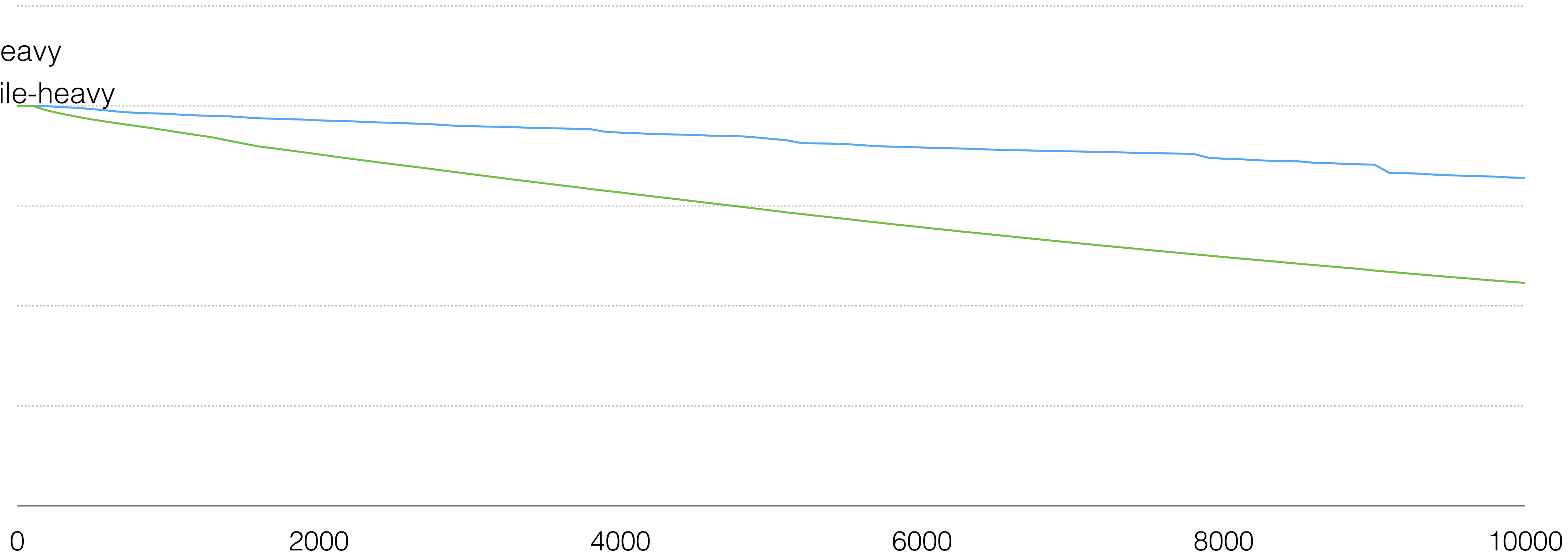
VIDEO VIEWS

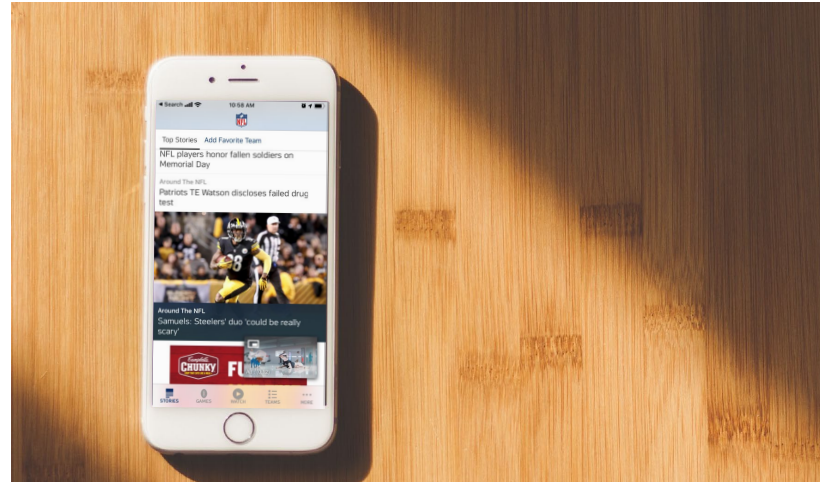


5. Audience bandwidth data

— TV-heavy

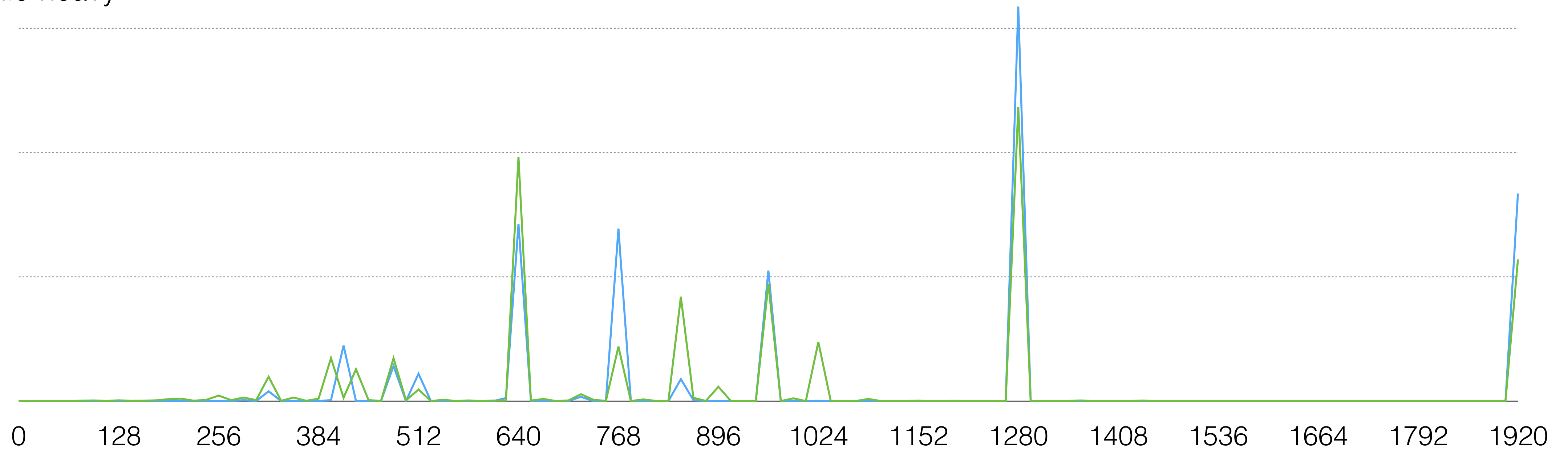
— Mobile-heavy





6. Audience resolution data

— TV-heavy
— Mobile-heavy





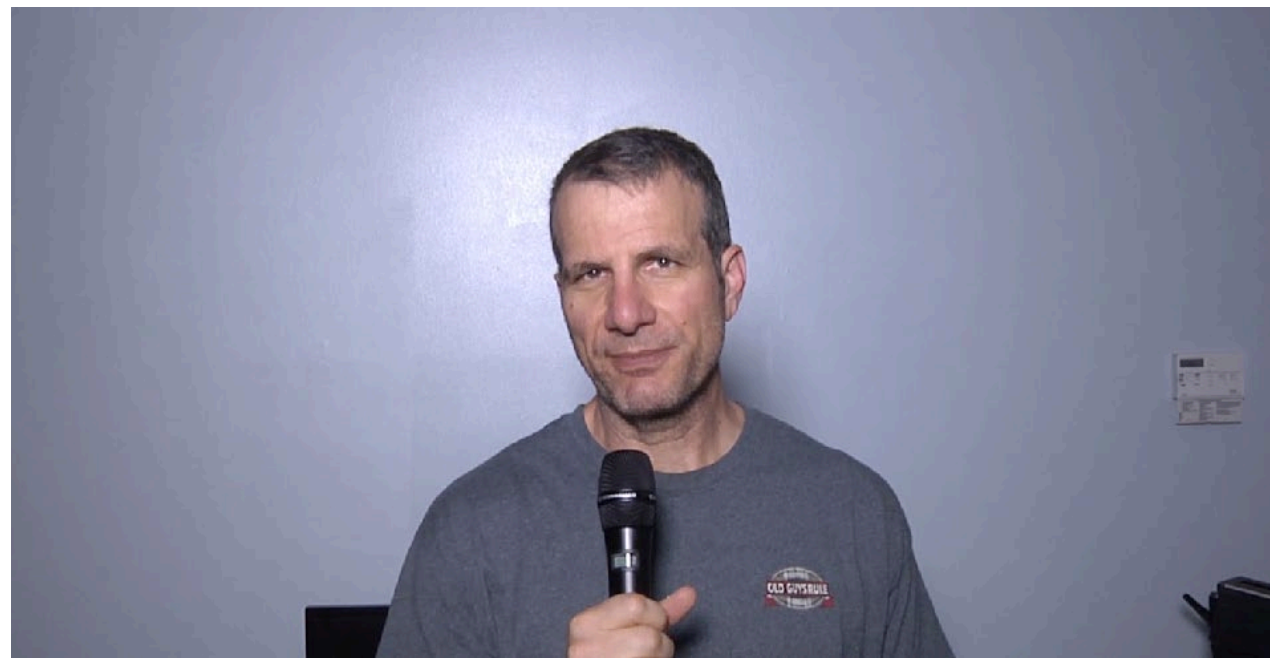
1	7437	1280x720
2		
3		
4		
5	502	426x240

1	7437	1280x720
2		
3		
4		
5	260	426x240



1	6500	1920x1080
2		
3		
4		
5	502	426x240

1	6500	1920x1080
2		
3		
4		
5	260	426x240



1	2000	1920x1080
2		
3		
4		
5	502	618x348

1	2000	1920x1080
2		
3		
4		
5	260	426x240

MUX

TV-heavy

Mobile-heavy



1	6500	1920x1080
2		
3		
4		
5	502	426x240

75% of audience

50% of audience

25% of audience



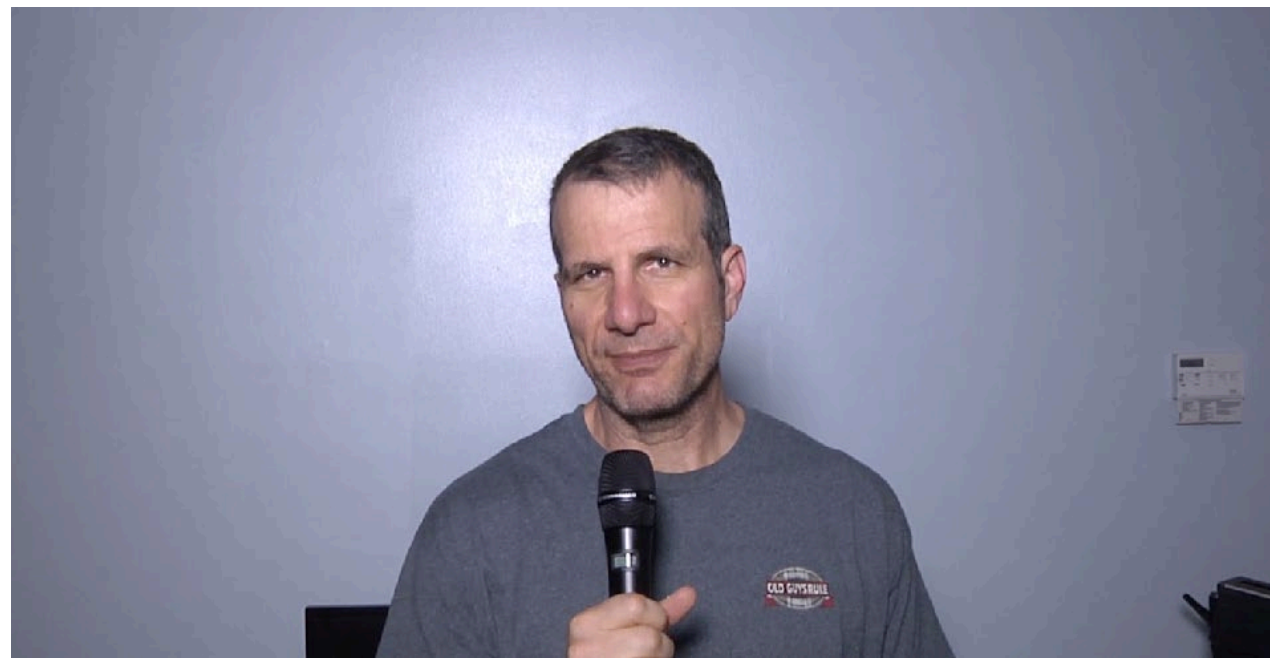
1	7437	1280x720
2	2962	960x540
3	1639	640x360
4	998	440x248
5	502	426x240

1	7437	1280x720
2	2021	768x432
3	1515	576x324
4	1114	476x268
5	260	426x240



1	6500	1920x1080
2	3528	1280x720
3	2417	960x540
4	1232	576x324
5	502	426x240

1	6500	1920x1080
2	1764	768x432
3	1154	576x324
4	884	462x260
5	260	426x240



1	2000	1920x1080
2	1515	1280x720
3	1037	960x540
4	558	576x324
5	502	640x360

1	2000	1920x1080
2	812	768x432
3	597	576x324
4	356	426x240
5	260	426x240

MUX

TV-heavy

Mobile-heavy

What did we just do?

1	[VMAF 94]	[1080p]
2		
3		
4		
5	[2% of users]	[240p]

What did we just do?

Optimized top bitrate

1	[VMAF 94]	[1080p]
2		
3		
4		
5	[2% of users]	[240p]

Optimized top bitrate

PTE top resolution

1	[VMAF 94]	[1080p]
2		
3		
4		
5	[2% of users]	[240p]

Optimized top bitrate

PTE top resolution

1	[VMAF 94]	[1080p]
2		
3		
4		
5	[2% of users]	[240p]

Data-driven bottom rendition

Optimized top bitrate

PTE top resolution

1	[VMAF 94]	[1080p]
2		
3		
4		
5	[2% of users]	[240p]

} Intermediate renditions
tailored to audience

Data-driven bottom rendition



One last thing:

How do you measure this?

1. QoE

Resolution Fit

	Static Ladder	Audience Adaptive Ladder
Upscaling (Average)	6.56%	13.37%
Upscaling (95%)	35.36%	67.30%
Downscaling (Average)	25.81%	20.97%
Downscaling (95%)	72.21%	73.06%

162,135 video views, measured October 2019
Mid-sized OTT platform



Rebuffering

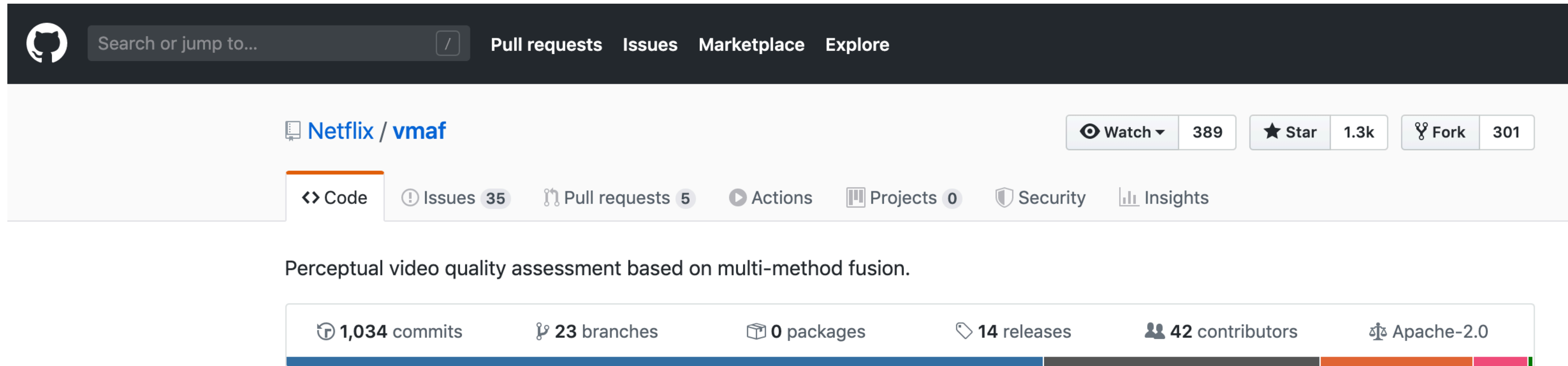
	Static Ladder	Audience Adaptive Ladder
Rebuffering Percentage	1.94%	3.21%
Rebuffering Frequency	0.14/minute	0.15/minute
Rebuffering Duration (50%)	1.53 seconds	1.66 seconds
Rebuffering Duration (95%)	19.96 seconds	22.53 seconds

162,135 video views, measured October 2019
Mid-sized OTT platform



2. Measure the actual ladder

Not like this



The screenshot shows the GitHub interface for the repository `Netflix/vmaf`. At the top, there is a dark navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. Below this, the repository name `Netflix/vmaf` is displayed, along with buttons for Watch (389), Star (1.3k), and Fork (301). A secondary navigation bar contains links for Code, Issues (35), Pull requests (5), Actions, Projects (0), Security, and Insights. The repository description reads: "Perceptual video quality assessment based on multi-method fusion." At the bottom, a statistics bar shows 1,034 commits, 23 branches, 0 packages, 14 releases, 42 contributors, and Apache-2.0 license.

Search or jump to... / Pull requests Issues Marketplace Explore

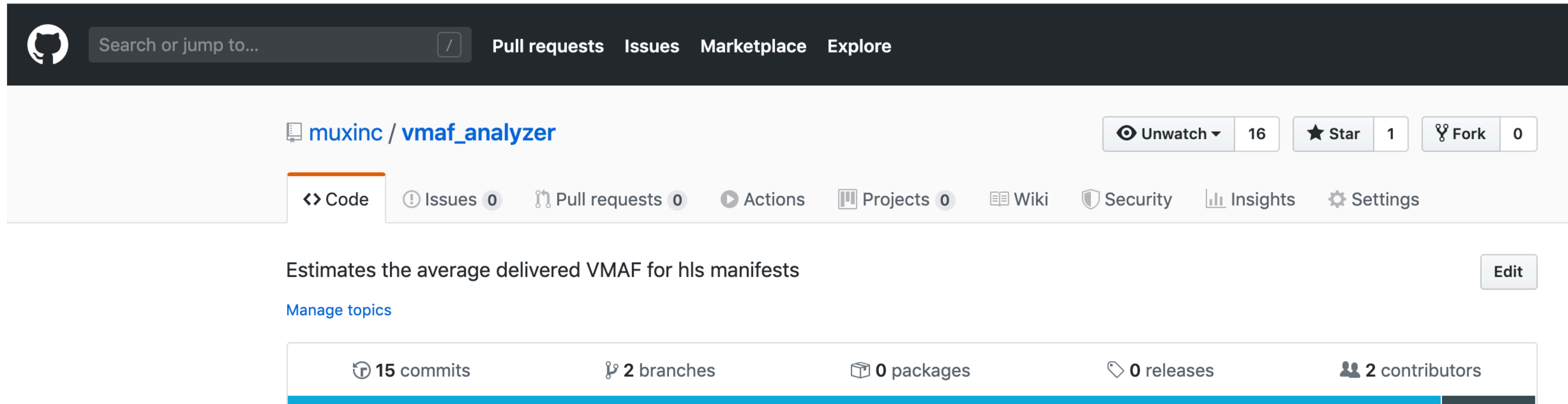
Netflix / vmaf Watch 389 Star 1.3k Fork 301

<> Code Issues 35 Pull requests 5 Actions Projects 0 Security Insights

Perceptual video quality assessment based on multi-method fusion.

1,034 commits 23 branches 0 packages 14 releases 42 contributors Apache-2.0

Like this



The screenshot shows the GitHub interface for the repository `muxinc / vmaf_analyzer`. At the top, there is a dark navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. Below this, the repository name is displayed along with interaction buttons: Unwatch (16), Star (1), and Fork (0). A secondary navigation bar contains links for Code, Issues (0), Pull requests (0), Actions, Projects (0), Wiki, Security, Insights, and Settings. The repository description is "Estimates the average delivered VMAF for hls manifests", with an "Edit" button to its right. Below the description is a "Manage topics" link. At the bottom, a summary bar shows: 15 commits, 2 branches, 0 packages, 0 releases, and 2 contributors.

Search or jump to... / Pull requests Issues Marketplace Explore

`muxinc / vmaf_analyzer` Unwatch 16 Star 1 Fork 0

<> Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

Estimates the average delivered VMAF for hls manifests Edit

[Manage topics](#)

15 commits 2 branches 0 packages 0 releases 2 contributors

The screenshot shows the GitHub interface for the repository `Netflix/vmaf`. At the top, there is a search bar and navigation links for Pull requests, Issues, Marketplace, and Explore. Below the repository name, there are buttons for Watch (389), Star (1.3k), and Fork (301). A secondary navigation bar includes Code, Issues (35), Pull requests (5), Actions, Projects (0), Security, and Insights. The repository description is "Perceptual video quality assessment based on multi-method fusion." At the bottom, a statistics bar displays: 1,034 commits, 23 branches, 0 packages, 14 releases, 42 contributors, and Apache-2.0 license.

For a **single encode**,
what is the **quality**?

The screenshot shows the GitHub interface for the repository 'muxinc / vmaf_analyzer'. At the top, there is a search bar and navigation links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the repository name, there are buttons for 'Unwatch' (16), 'Star' (1), and 'Fork' (0). A secondary navigation bar includes 'Code', 'Issues' (0), 'Pull requests' (0), 'Actions', 'Projects' (0), 'Wiki', 'Security', 'Insights', and 'Settings'. The repository description is 'Estimates the average delivered VMAF for hls manifests', with an 'Edit' button. Below this, a 'Manage topics' link is visible. A summary bar at the bottom shows: 15 commits, 2 branches, 0 packages, 0 releases, and 2 contributors.

For an **ABR ladder** and **multiple viewers**,
what is the **average quality** seen?