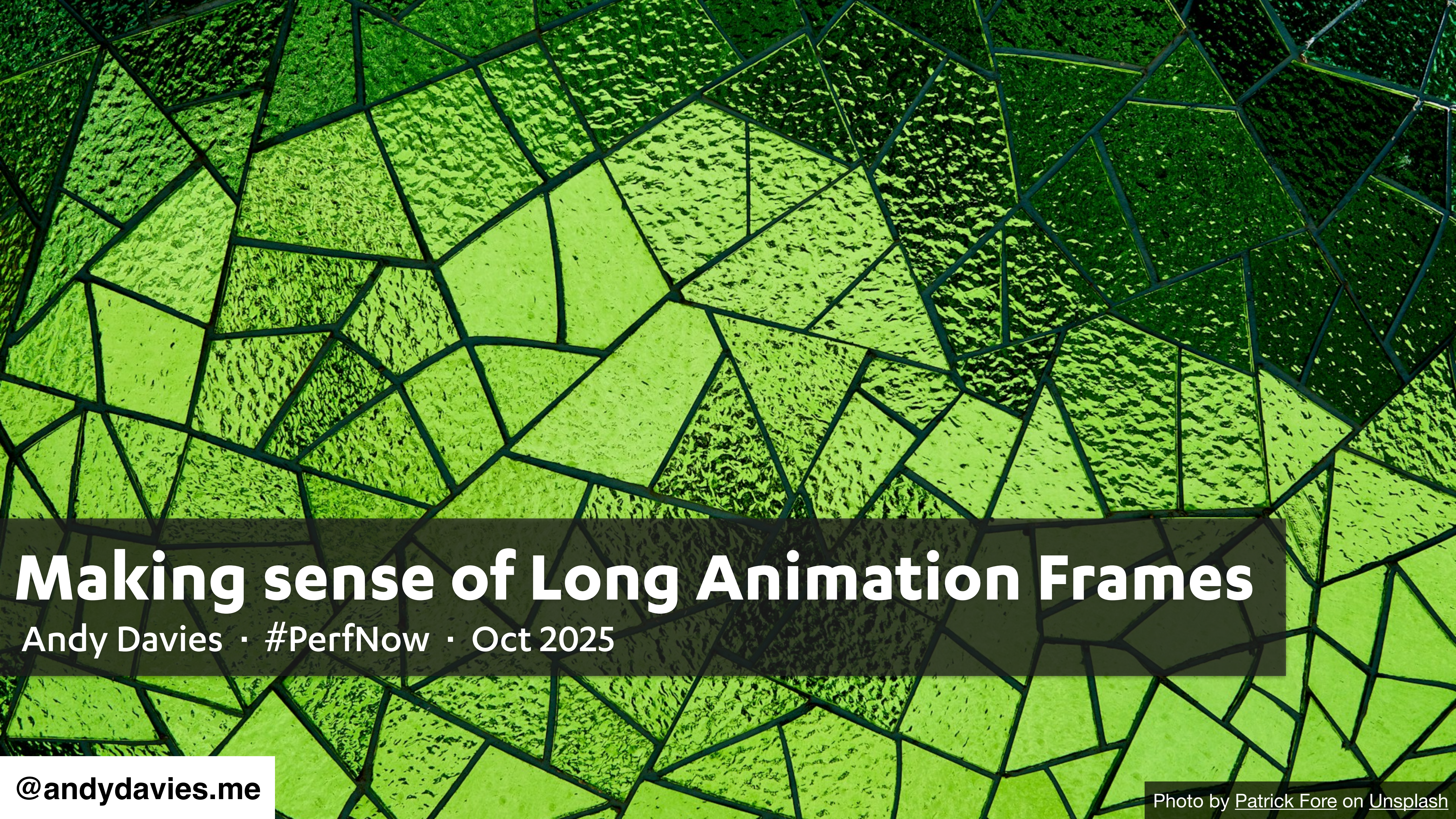




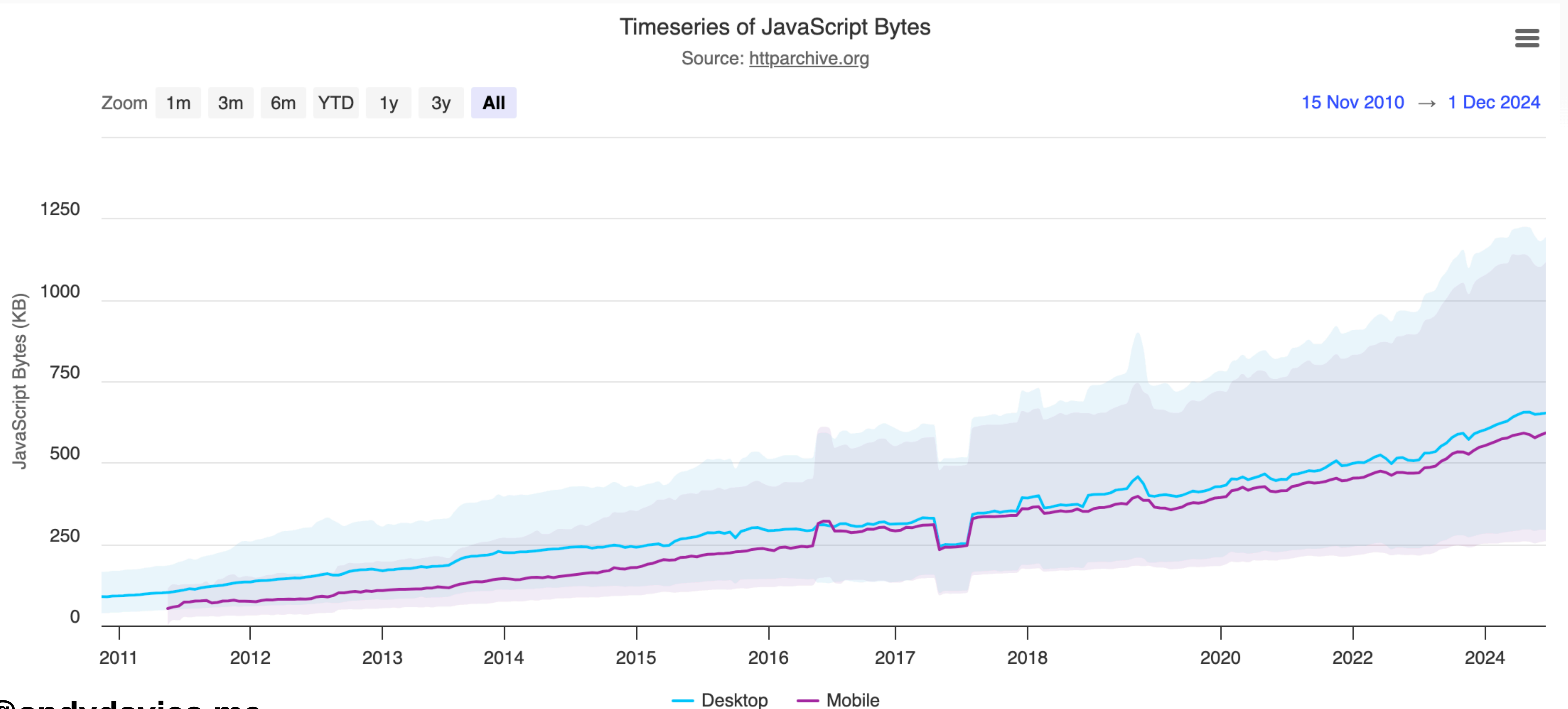
Making sense of Long Animation Frames

Andy Davies · #PerfNow · Oct 2025

@andydavies.me

Photo by [Patrick Fore](#) on [Unsplash](#)

We're shipping more and more JavaScript



Download Size != impact

```
document.addEventListener("mousemove", function() {  
    for(var a = Date.now() + 500; Date.now() < a;)  
        ;  
});
```

OK, so it's a bit of a silly example but you get the idea...



Runtime costs are a hidden danger

TABLE OF CONTENTS

- 1 Introduction
 - 1.1 Usage Example
- 2 Terminology
- 3 Long Task Timing
 - 3.1 PerformanceLongTaskTiming interface
 - 3.2 TaskAttributionTiming interface
 - 3.3 Pointing to the culprit
- 4 Processing model
 - 4.1 Report long tasks
- 5 Security & privacy considerations
 - 5.1 What is Exposed to Observers?
 - 5.2 Attack Scenarios Considered
- Conformance
 - Document conventions
 - Conformant Algorithms
- Index
 - Terms defined by this specification
 - Terms defined by reference
- Normative References
- IDL Index
- Issues Index

Long Tasks API

Editor's Draft, 24 May 2024



▼ More details about this document

- This version:**
<https://w3c.github.io/longtasks/>
- Test Suite:**
<http://w3c-test.org/longtask-timing/>
- Issue Tracking:**
[GitHub](#)
[Inline In Spec](#)
- Editor:**
[Noam Rosenthal](#) ([Google](#))
- Former Editors:**
[Shubhie Panicker](#) ([Google](#))
[Ilya Grigorik](#) ([Google](#))
[Domenic Denicola](#) ([Google](#))

Copyright © 2024 [World Wide Web Consortium](#). W3C® [liability](#), [trademark](#) and [permissive document license](#) rules apply.

Abstract

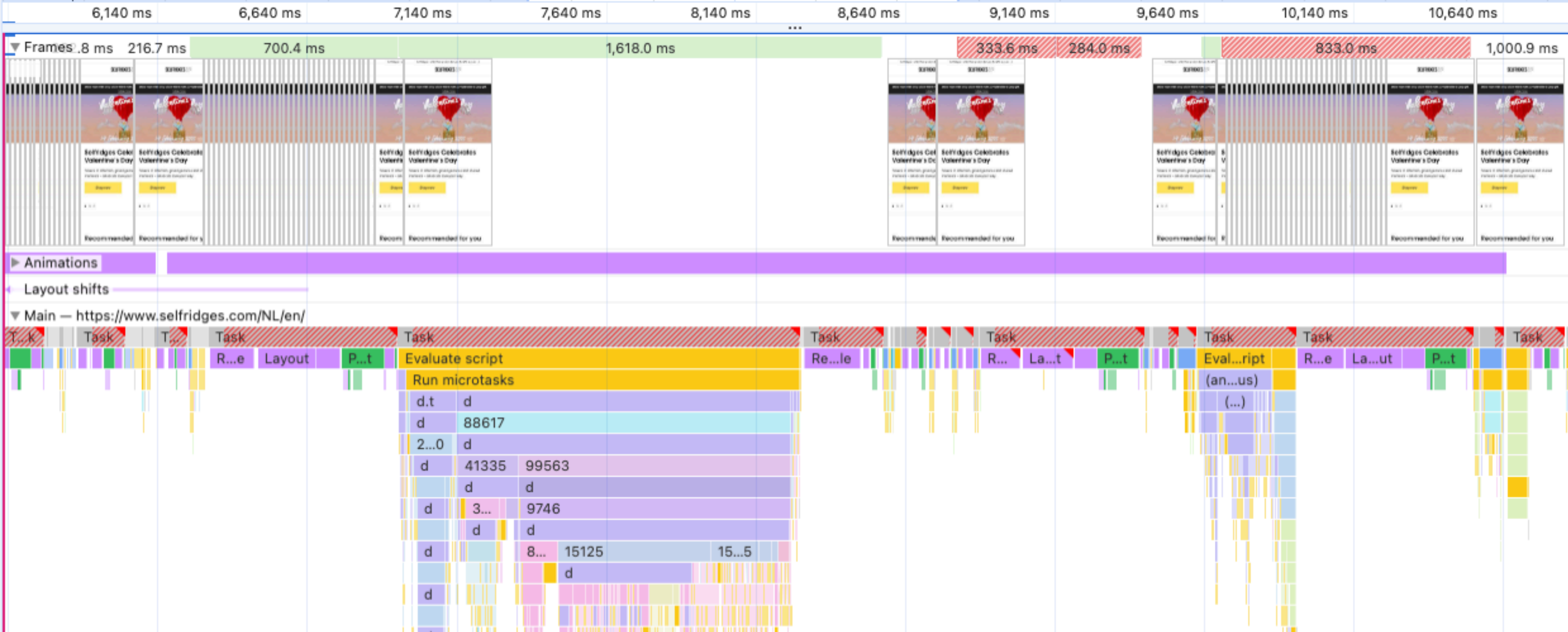
This document defines an API that lets web authors can schedule persistent "long tasks" that monopolize the UI thread for extended periods of time and block other critical tasks from being executed - e.g. reacting to user input.

Status of this document

This is a public copy of the editors' draft. It is provided for discussion purposes only. The publication here does not imply endorsement of its contents by W3C. It is a work in progress.

<https://w3c.github.io/longtasks/>

We've tried to measure them before...



Long Task = Main Thread Task > 50ms



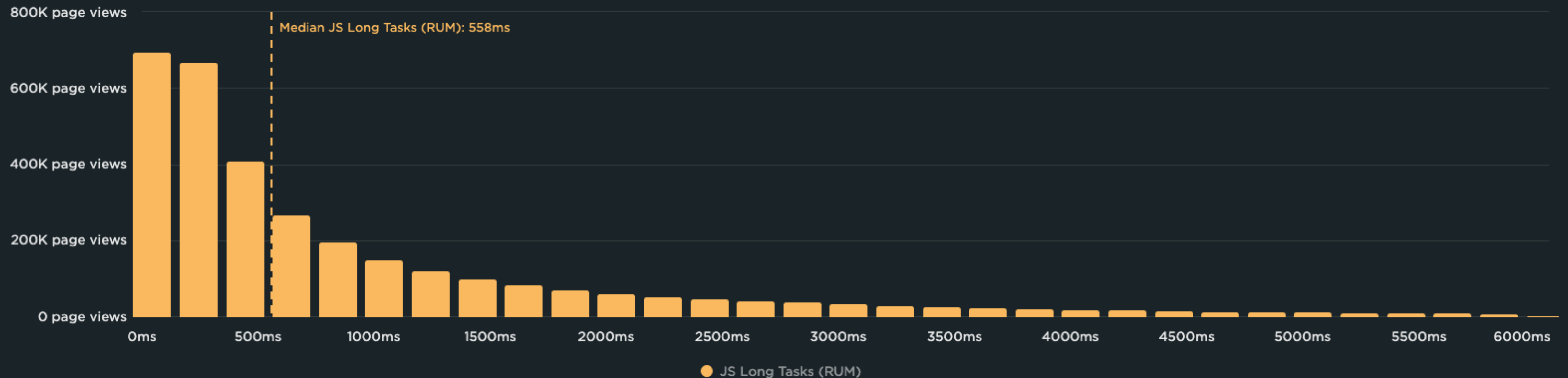
We can measure Long Tasks with RUM

JS LONG TASKS



JS Long Tasks (RUM)

558ms



But we get no detail on their cause

```
{
  "name": "self",
  "entryType": "longtask",
  "startTime": 48723.60000002384,
  "duration": 67,
  "navigationId": "3ca0c548-7618-4423-87b7-41ae43415040",
  "attribution": [
    {
      "name": "unknown",
      "entryType": "taskattribution",
      "startTime": 0,
      "duration": 0,
      "navigationId": "3ca0c548-7618-4423-87b7-41ae43415040",
      "containerType": "window",
      "containerSrc": "",
      "containerId": "",
      "containerName": ""
    }
  ]
}
```

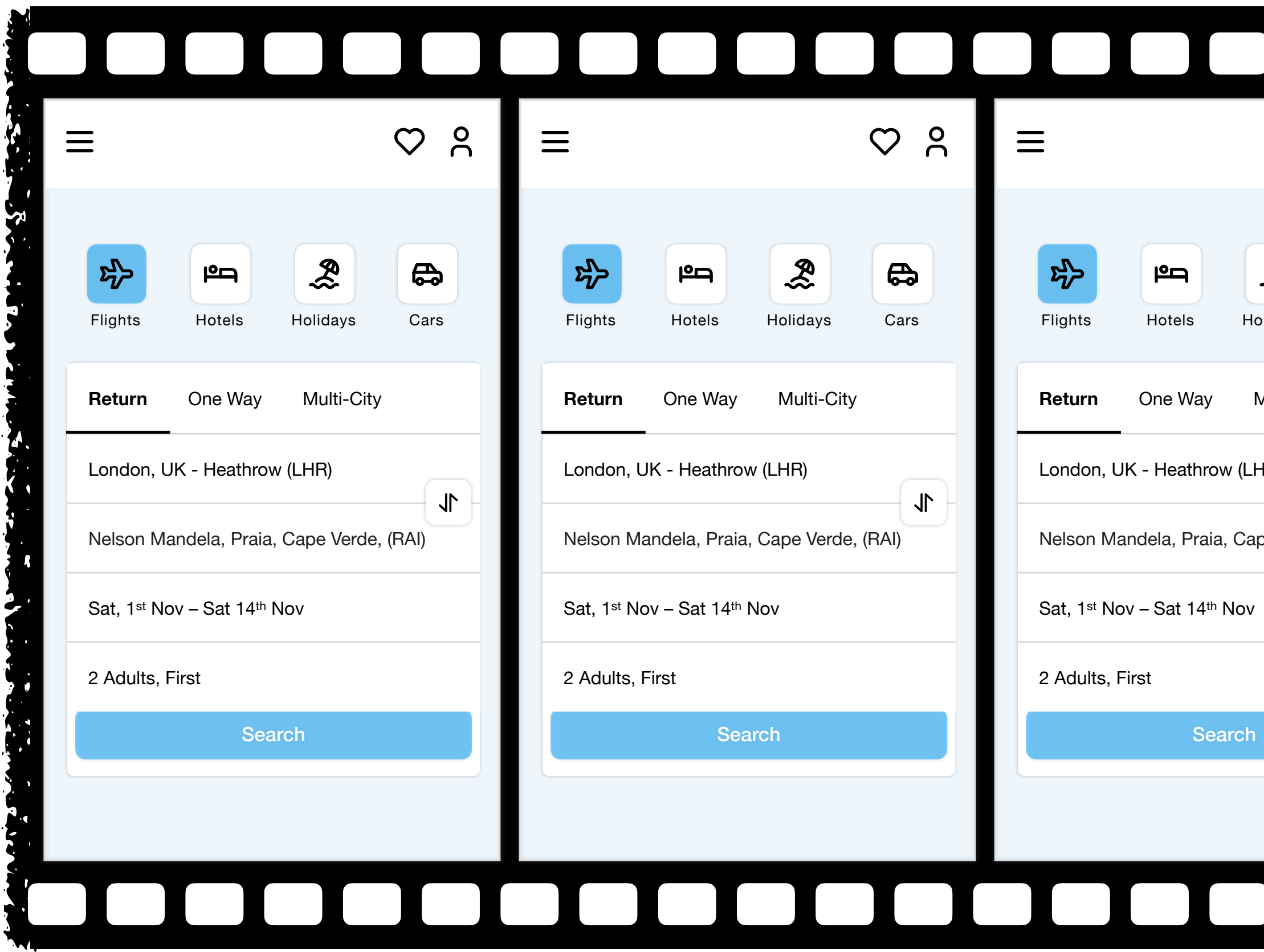
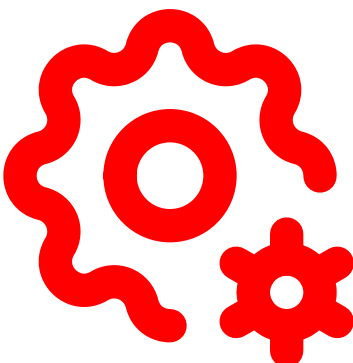
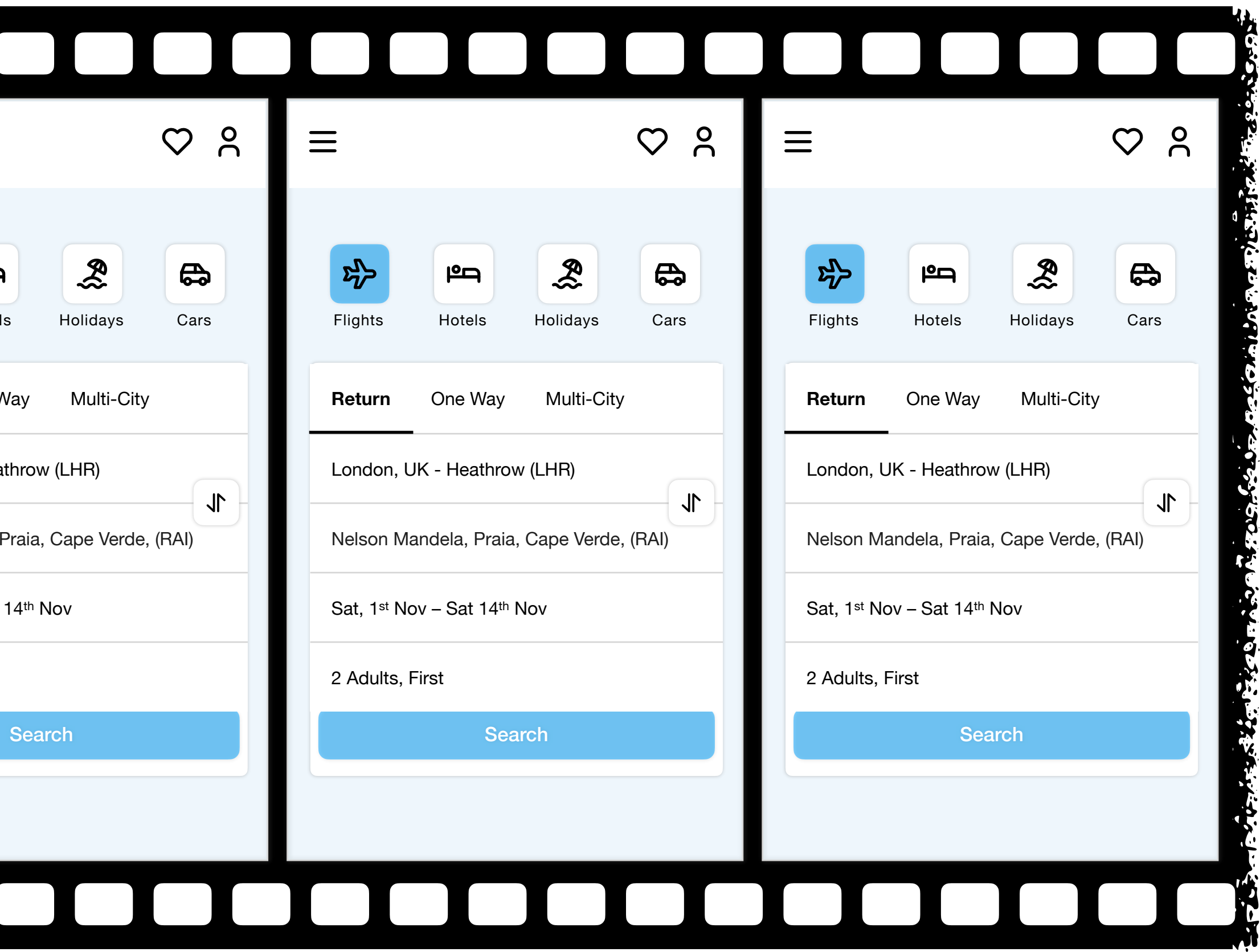




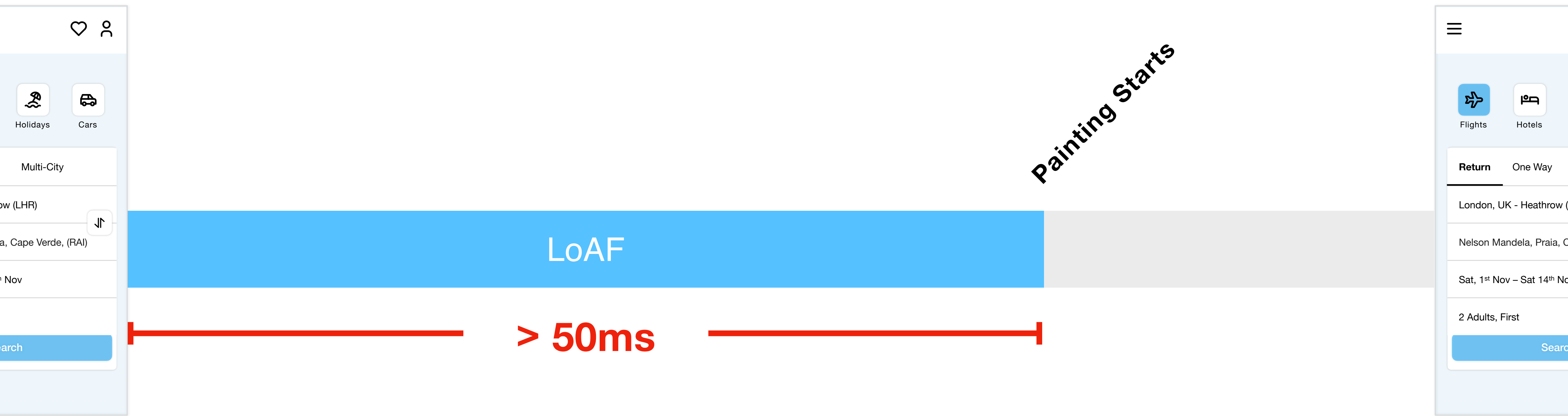

How can Long Animation Frames help us?

Lo-AF

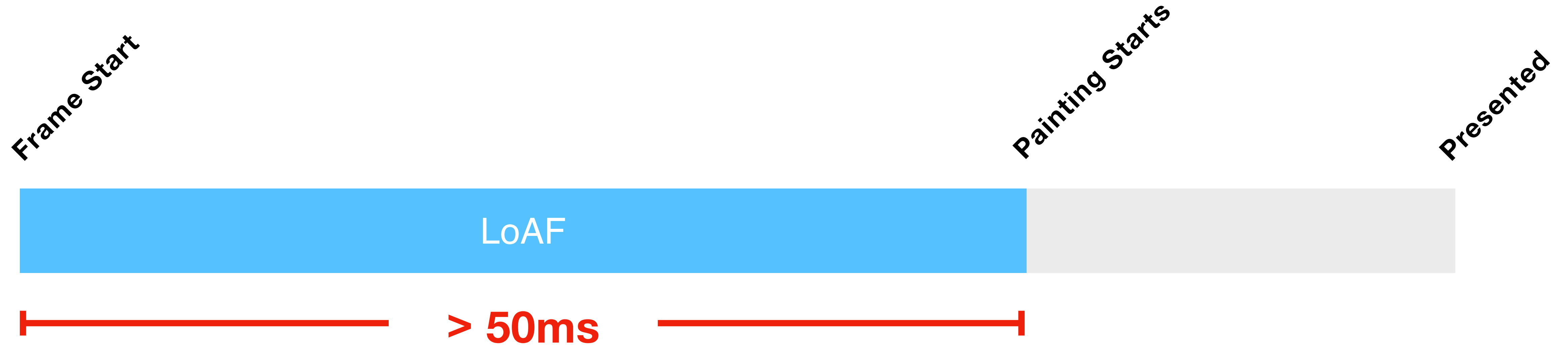
Long Animation Frames (LoAF)



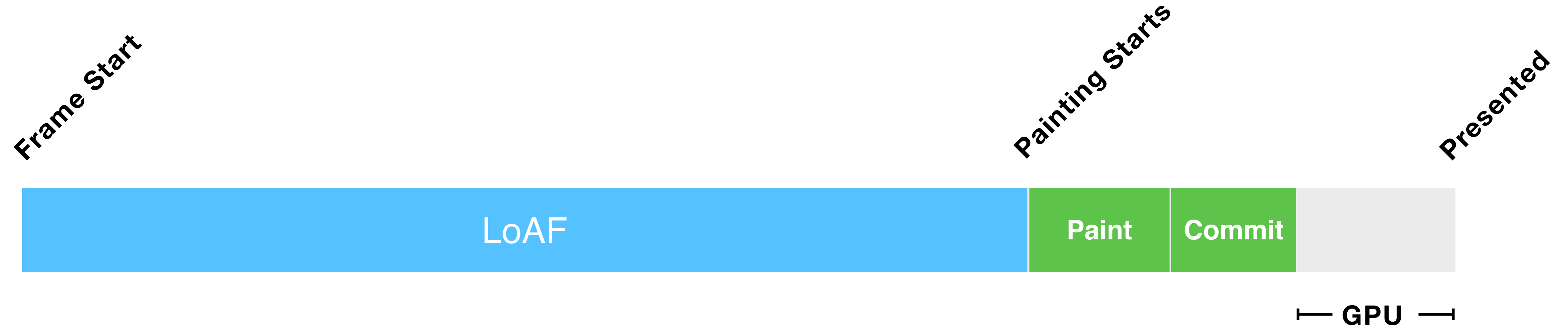
Partial measurement of frame duration



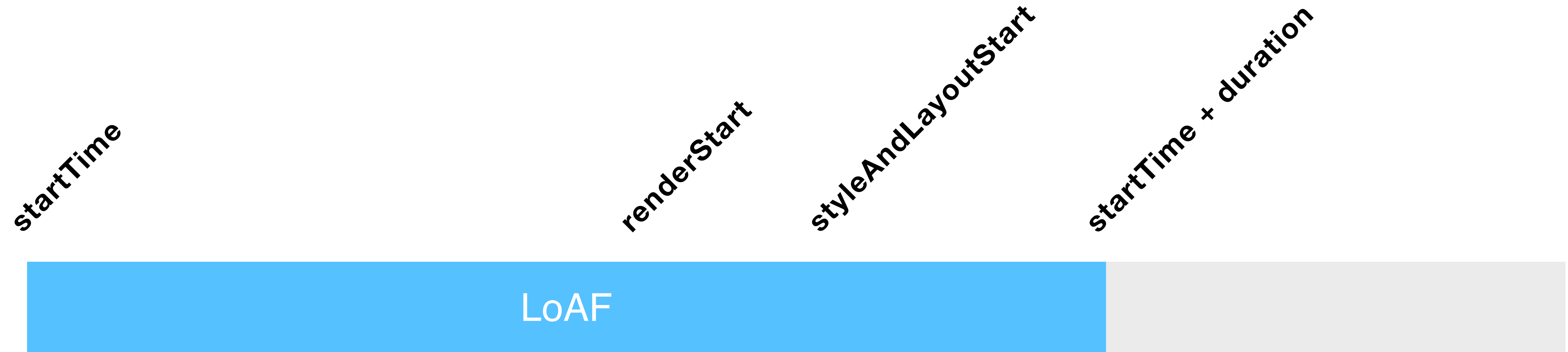
Start of frame to start of Painting



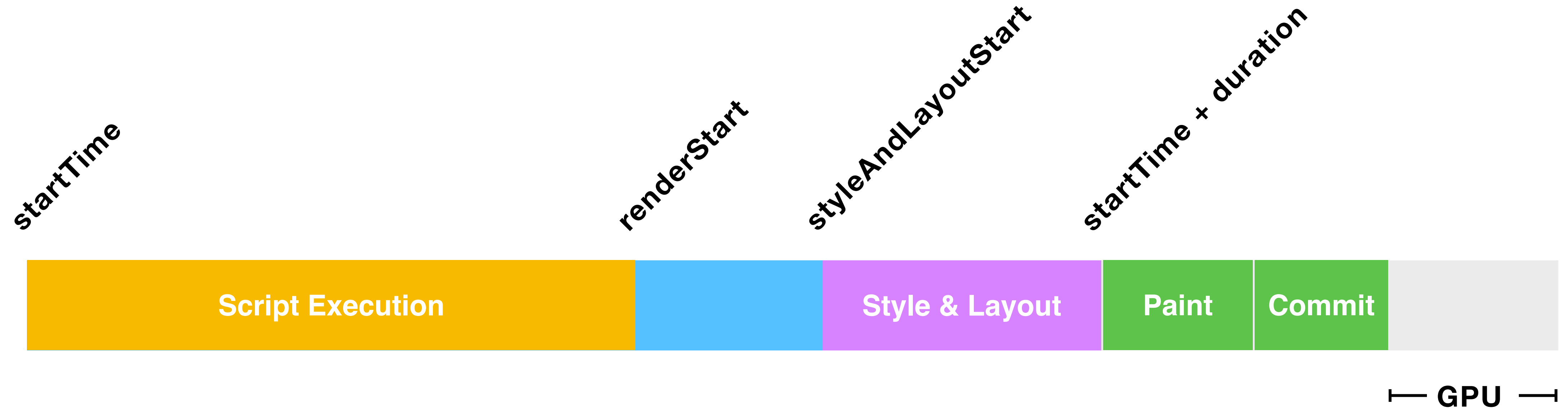
Start of frame to start of Painting



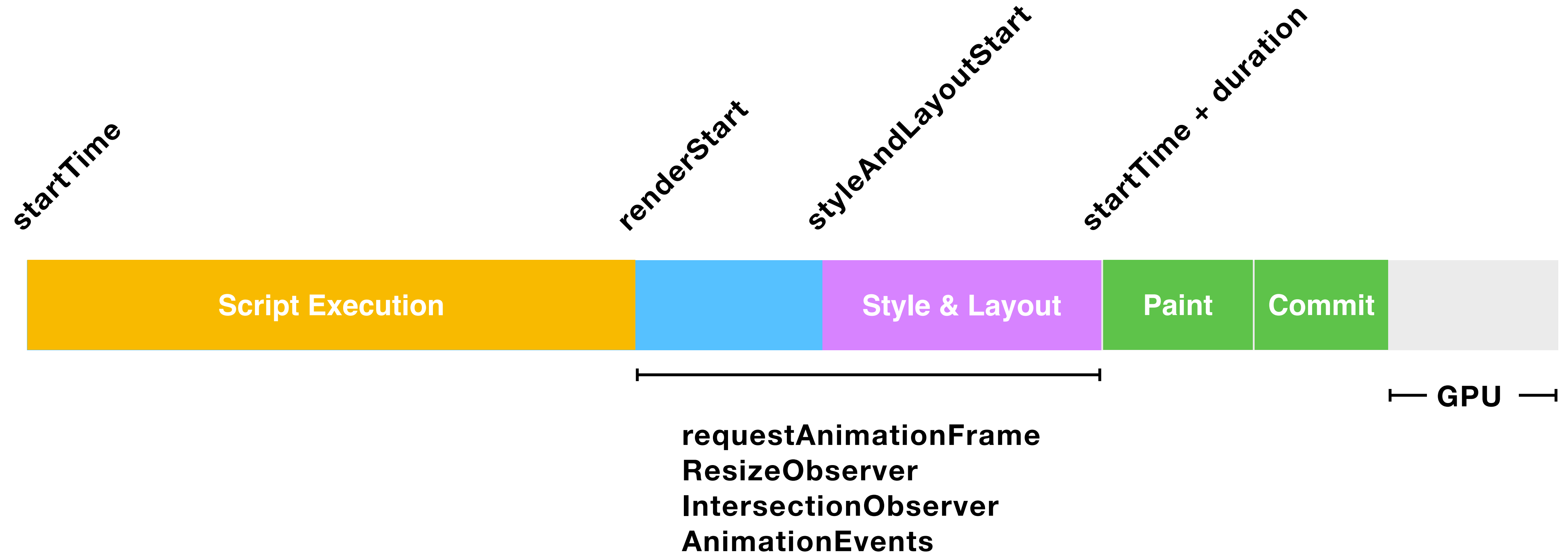
Timestamps for key milestones



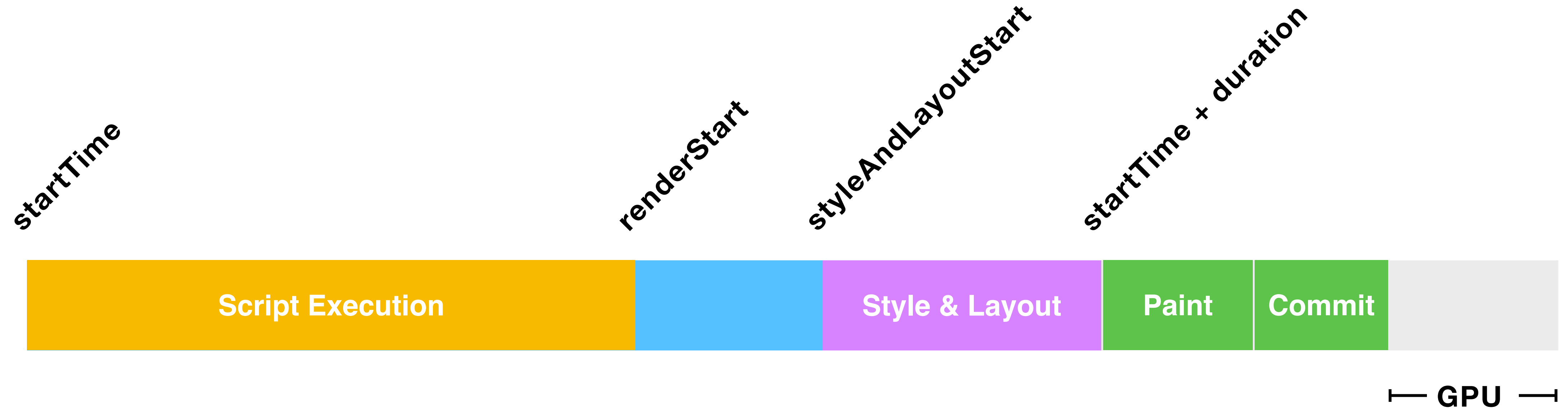
In simplified Main Thread terms...



Scripts execute during render phase too



Warning: This is a simplified view!



- scripts can force Style & Layout
- browsers sometimes perform Style and Layout mid-frame



Need a refresher on the Event Loop?

In the Loop, Jake Archibald

<https://vimeo.com/254947206>

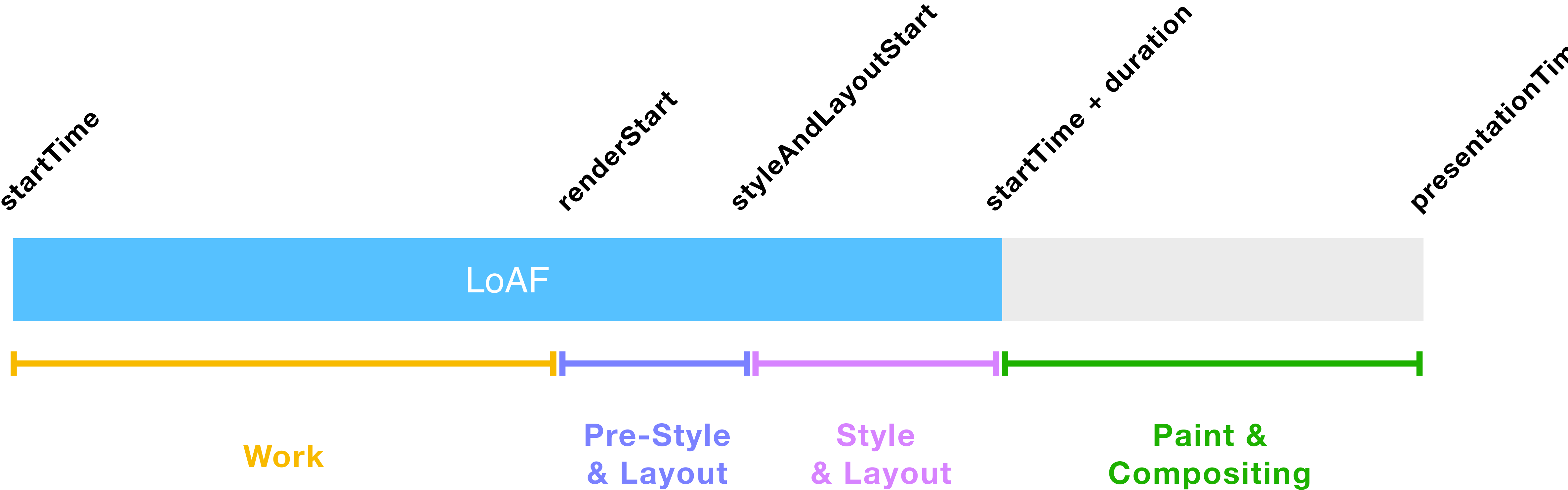
Paint Timing Mixin

Adds paintTime & presentationTime – Chrome Canary only ATM



<https://www.w3.org/TR/paint-timing/#the-paint-timing-mixin>

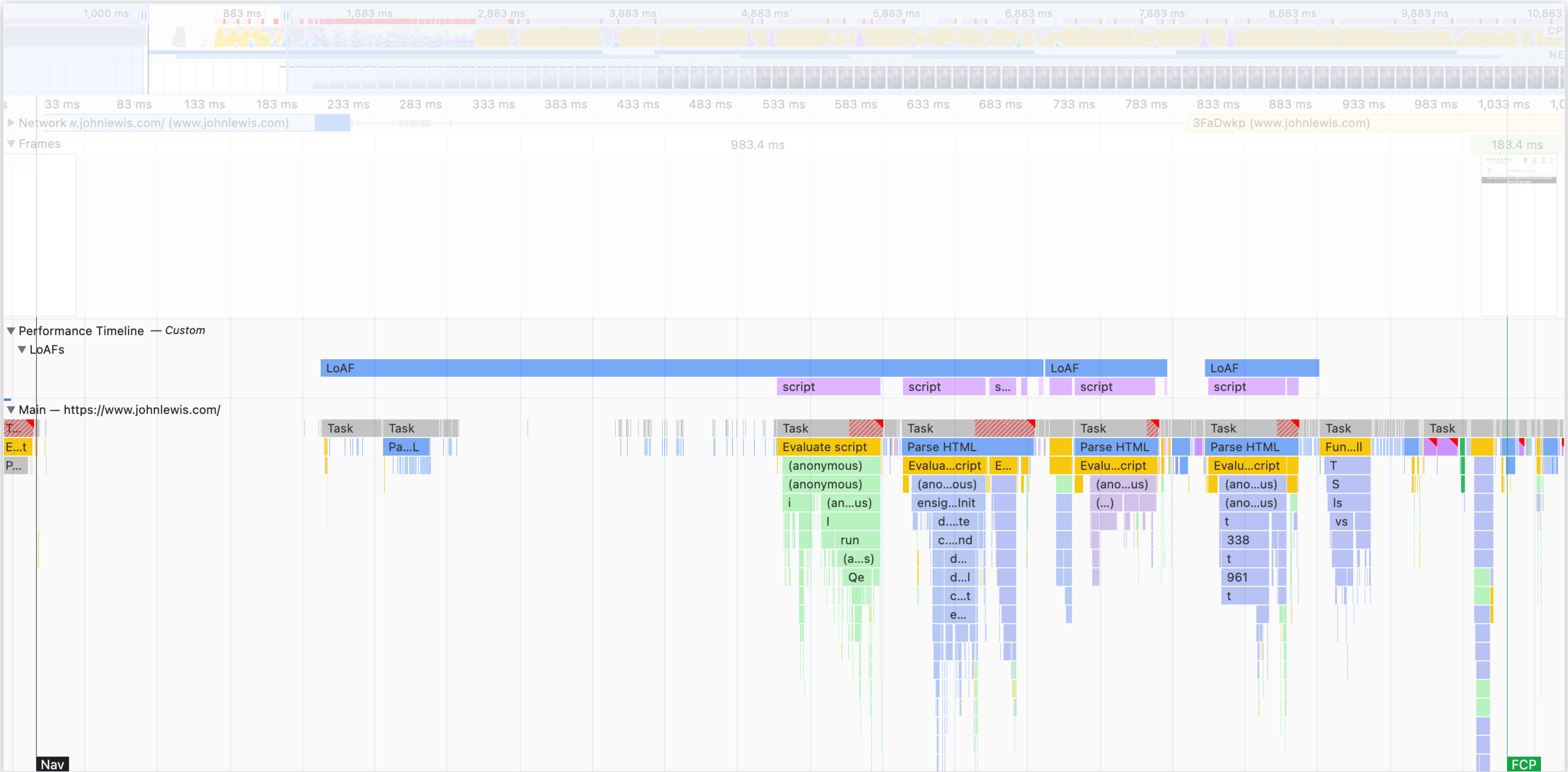
Calculate durations of internal phases



LoAFs can occur across the page lifecycle



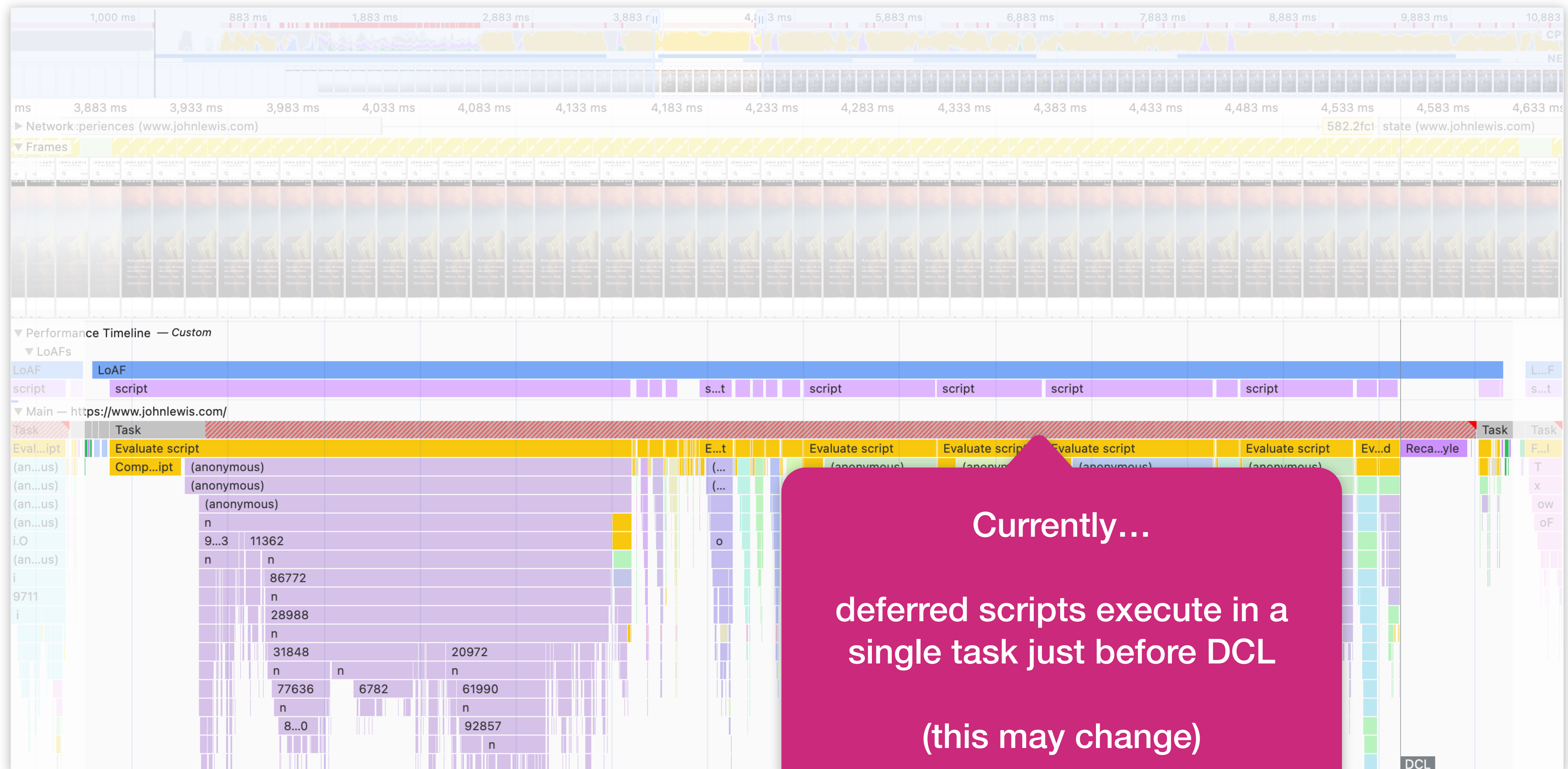
Normally at least one before FCP



Often a LoAF intersecting DCL



Often a LoAF intersecting DCL

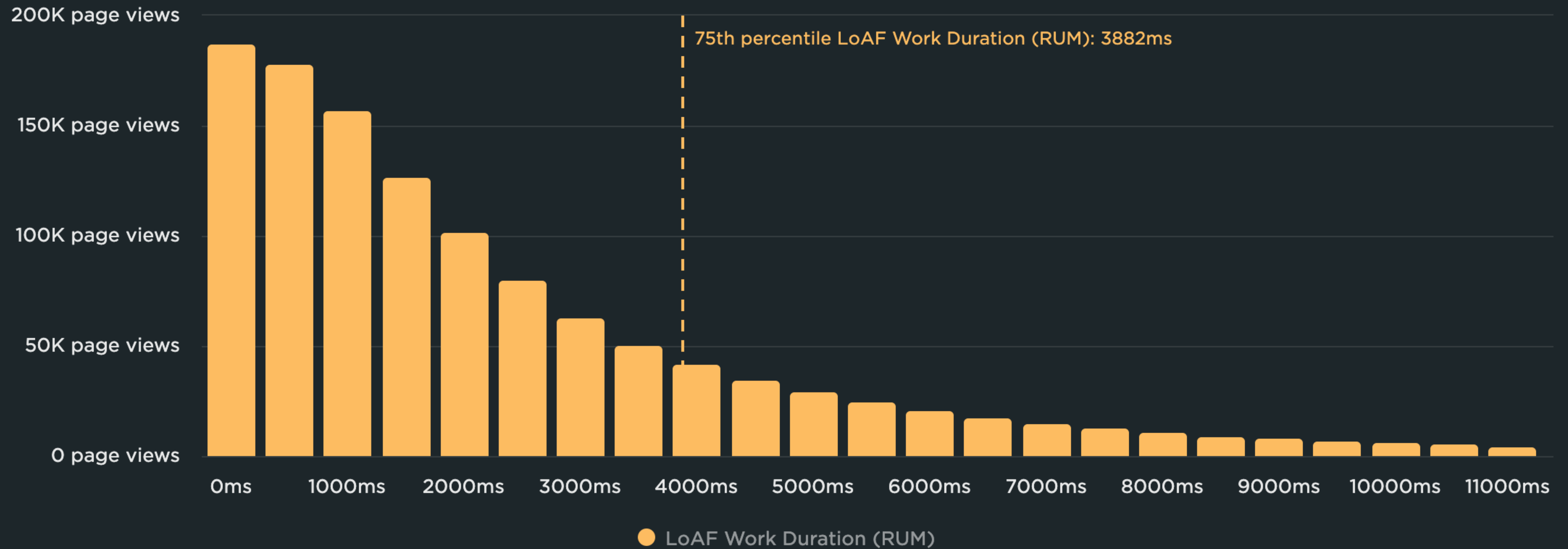


LOAF WORK DURATION



LoAF Work Duration (RUM)

3,882ms

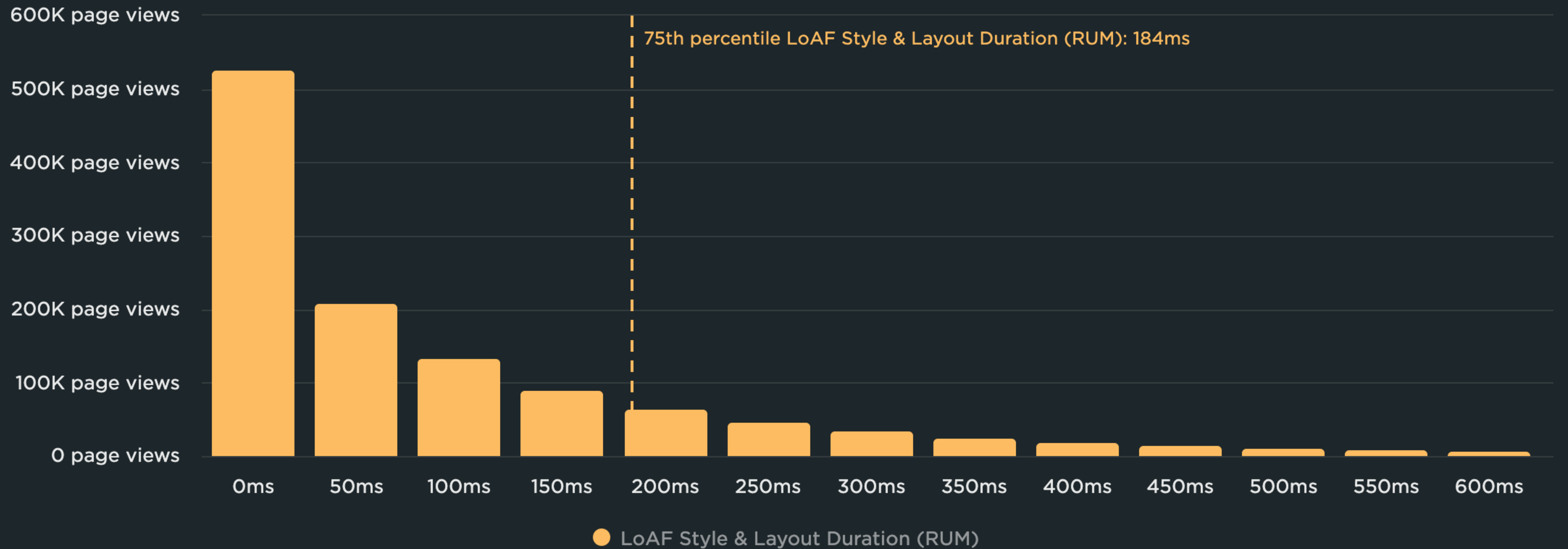


LOAF STYLE & LAYOUT DURATION



LoAF Style & Layout Duration (RUM)

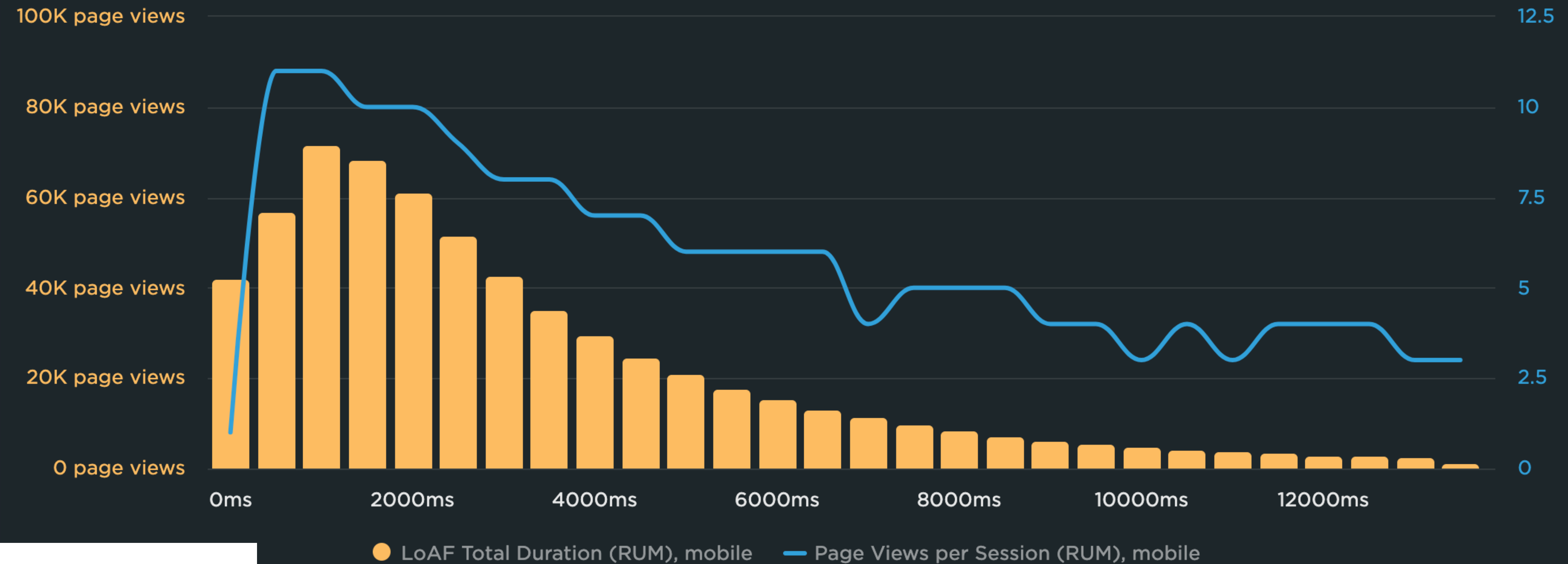
184ms



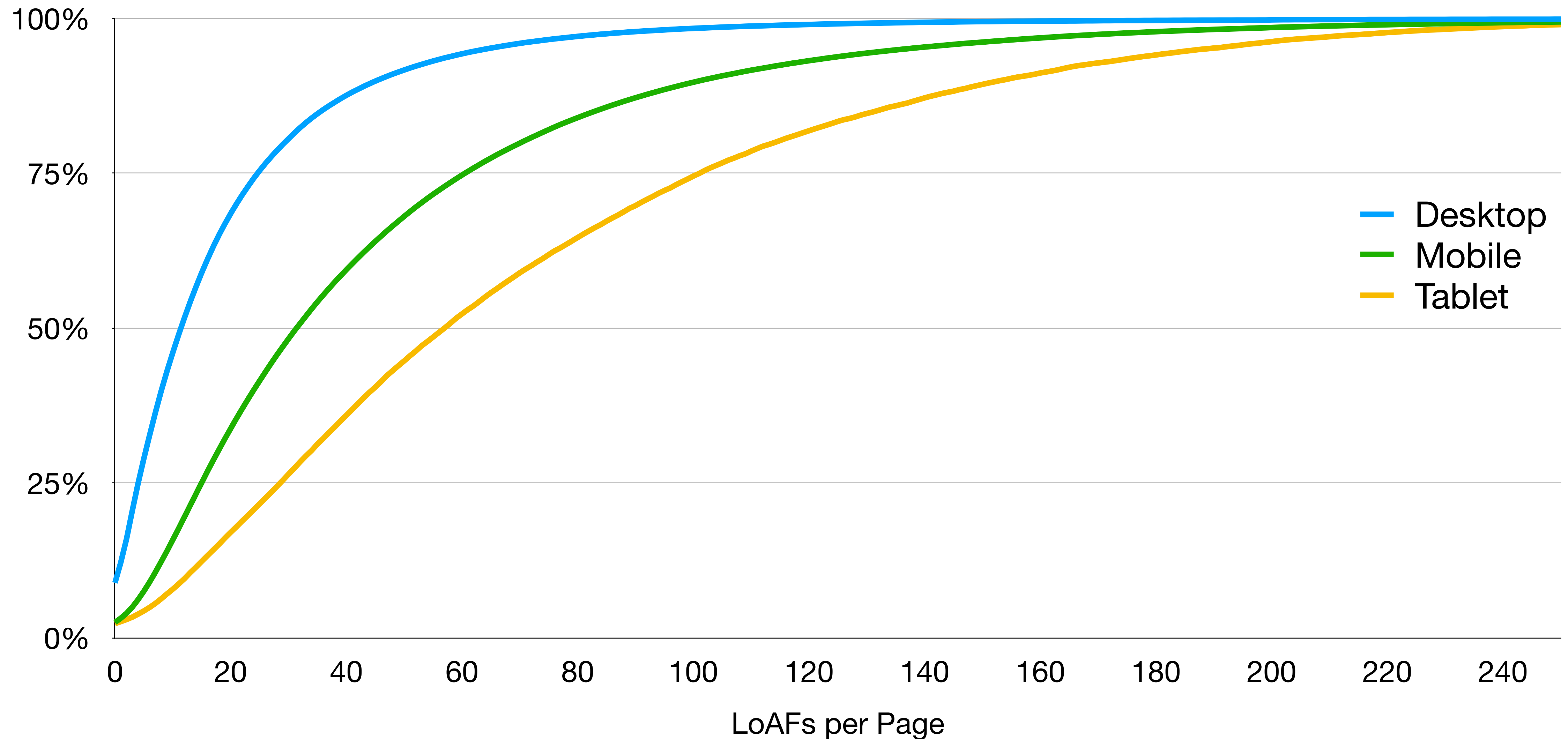
How do LoAFs affect visitor behaviour?

LoAF Total Duration (RUM), mobile

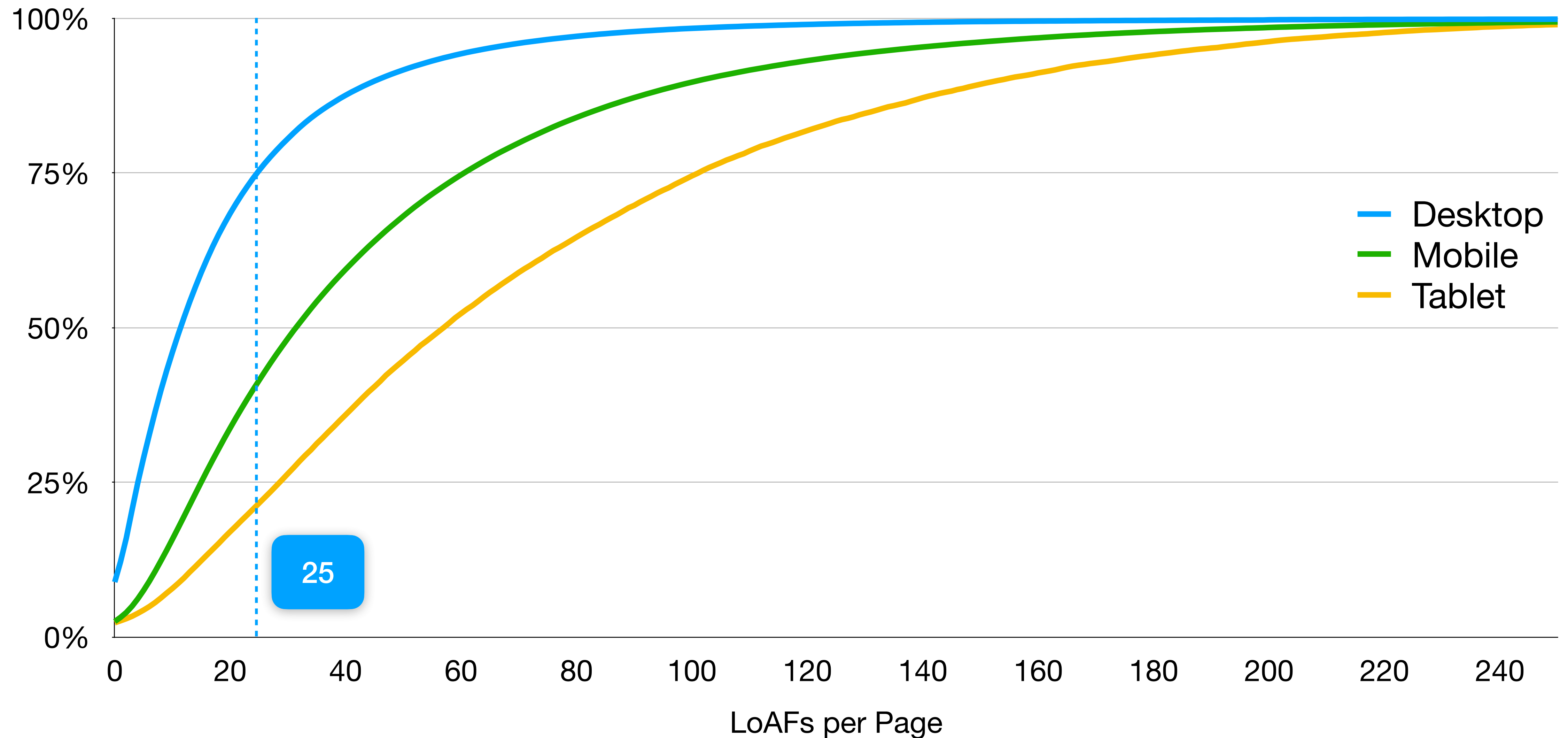
2,740ms



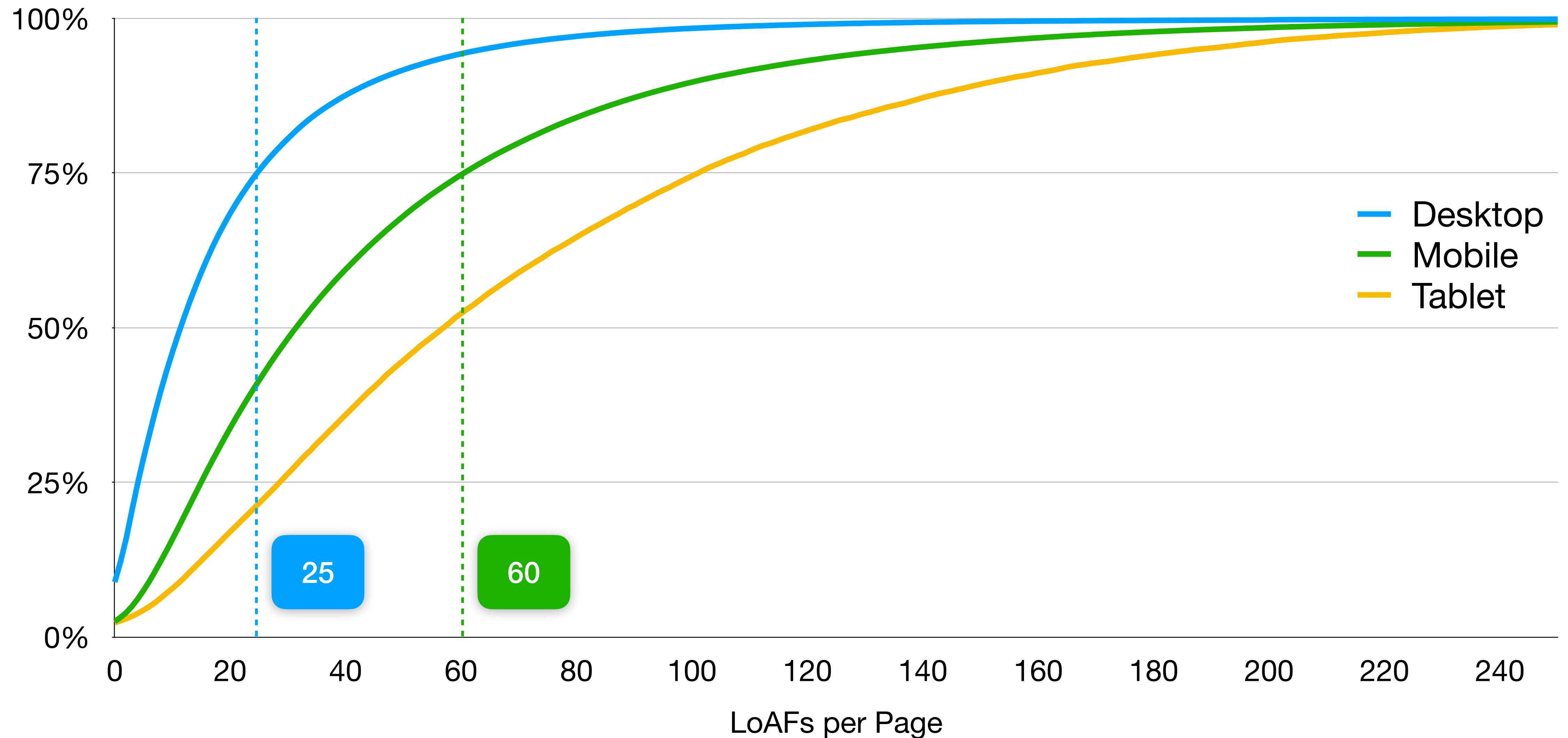
Pages can generate many LoAF entries...



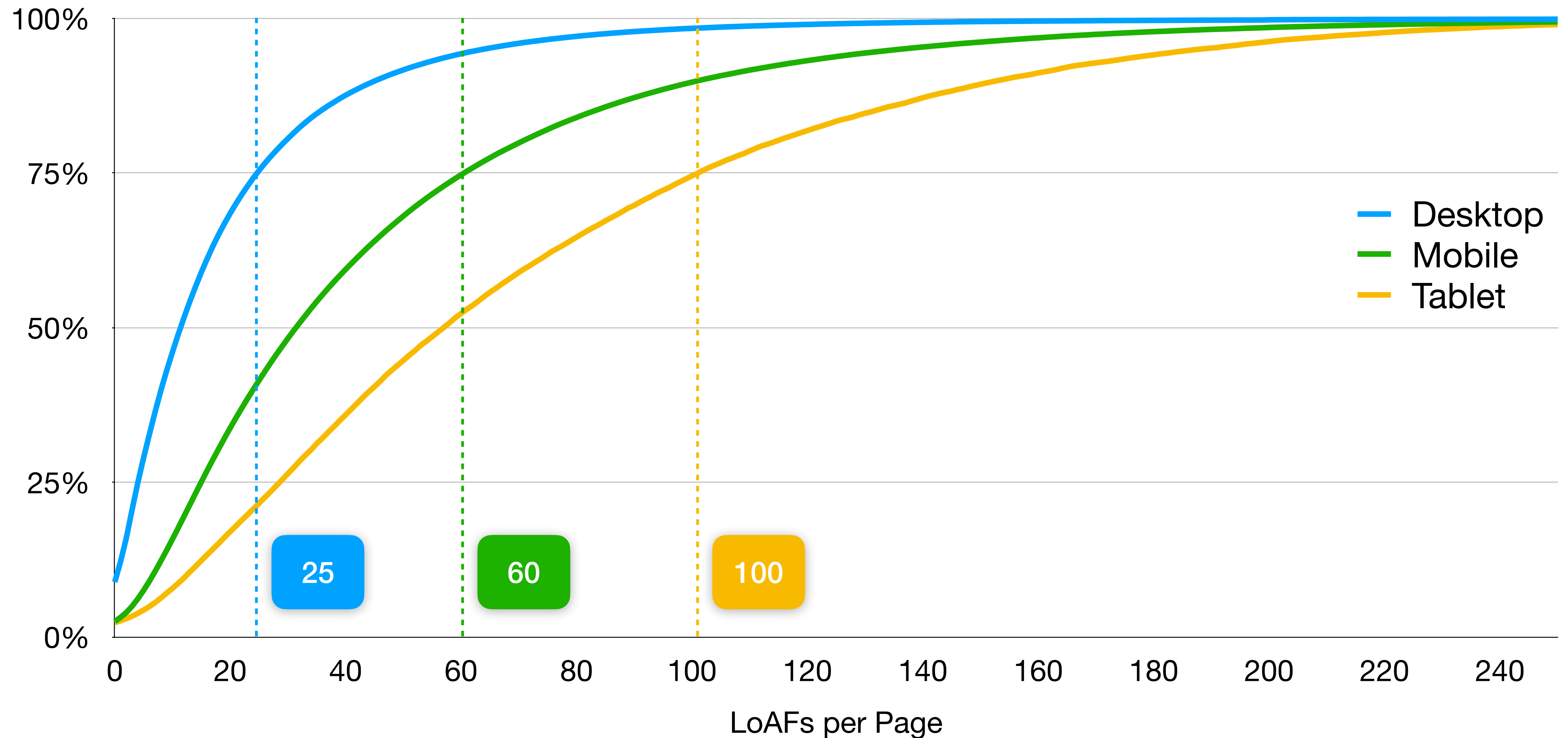
Pages can generate many LoAF entries...



Pages can generate many LoAF entries...



Pages can generate many LoAF entries...



Query entries from Performance Timeline

```
performance.getEntriesByType("long-animation-frame");
```


Observe entries via PerformanceObserver

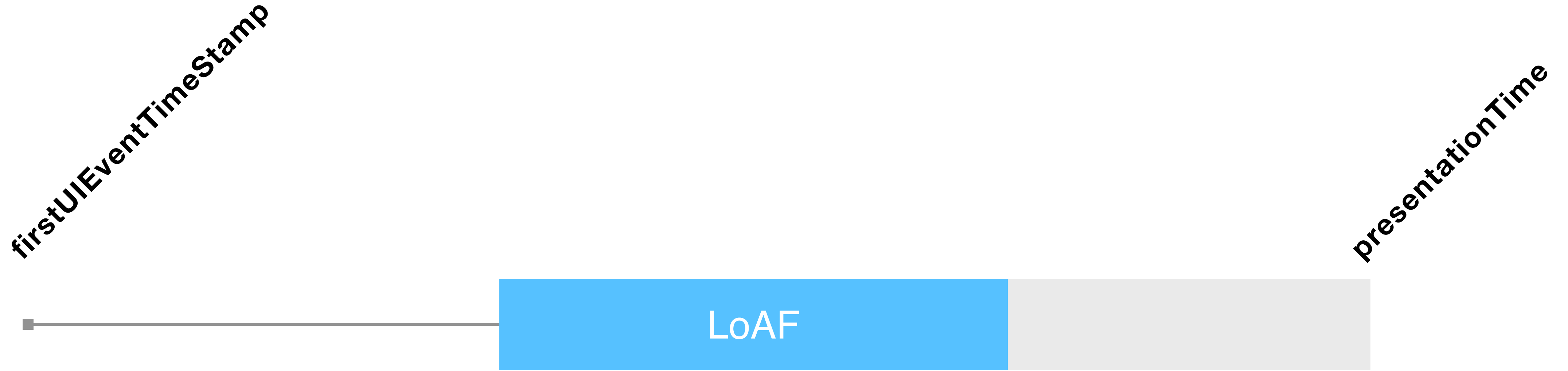
```
const observer = new PerformanceObserver((list) => {  
  for (const entry of list.getEntries()) {  
    console.log(entry)  
  }  
});  
  
observer.observe({ type: 'long-animation-frame', buffered: true });
```



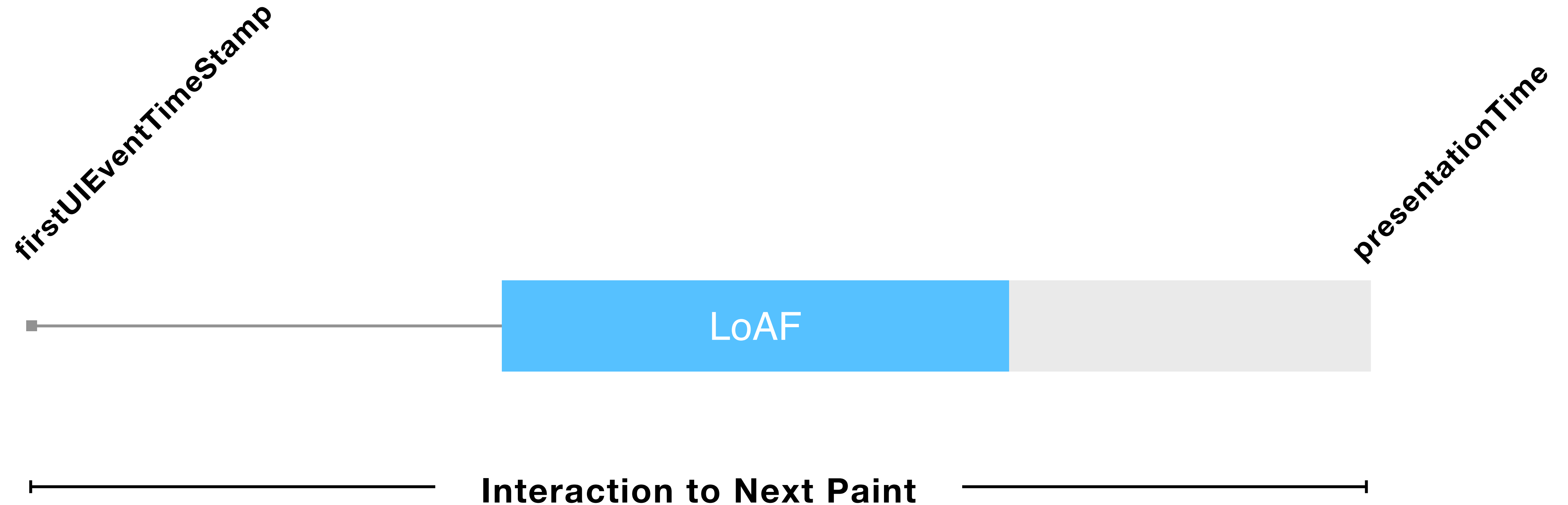

Entries are only created for visible tabs



Did the frame handle an interaction?

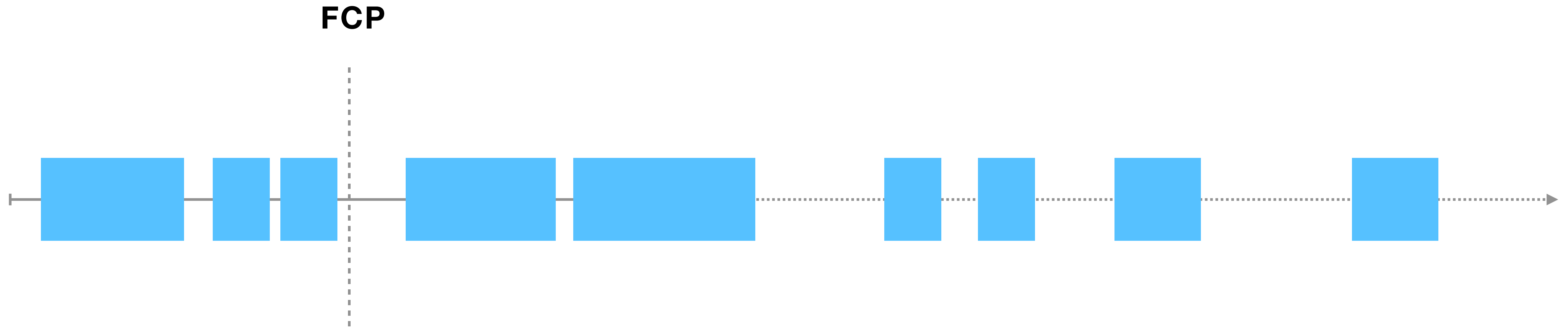


Did the frame handle an interaction?

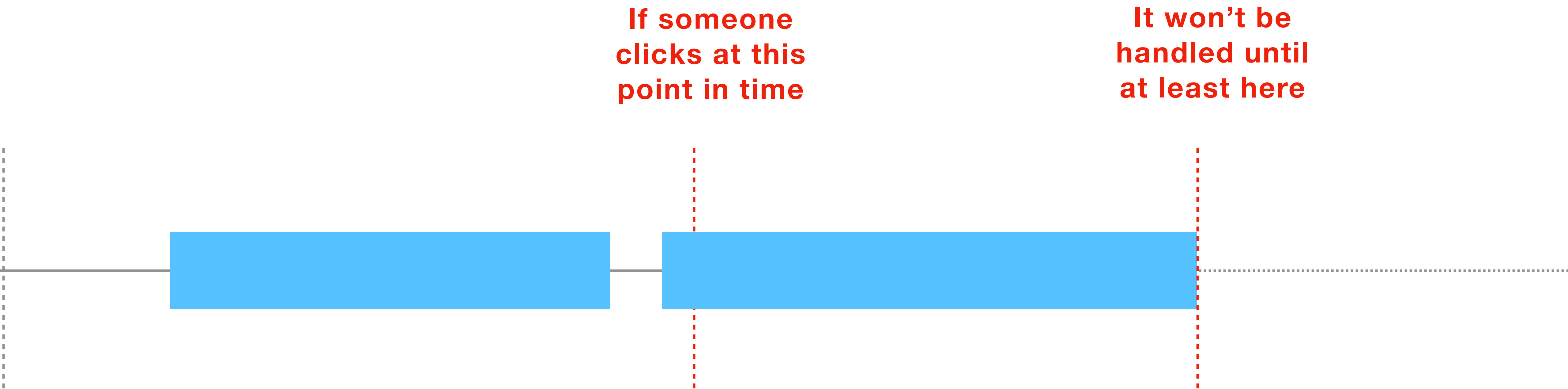


Measures a wider range of interactions than INP, but also has less detail

How large could Input Delay be?

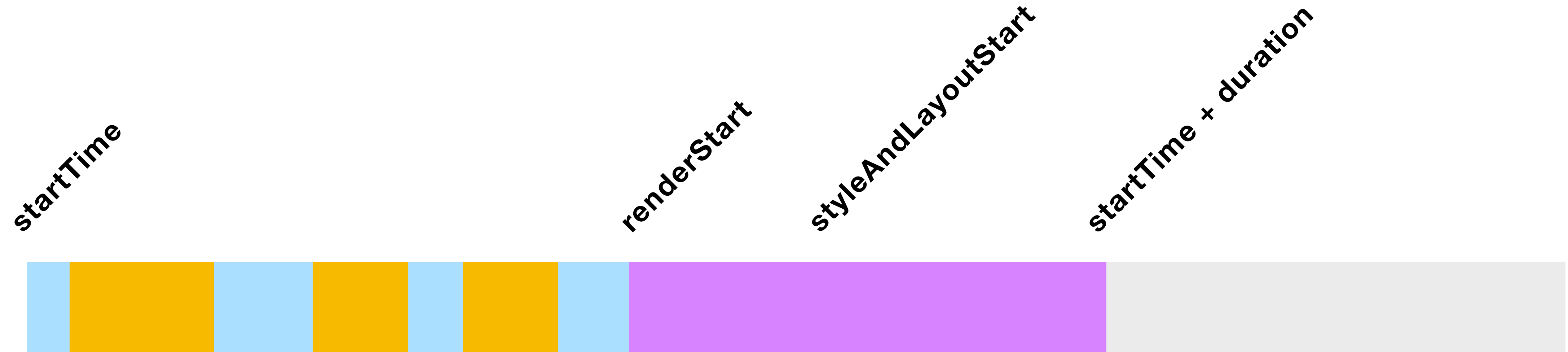


How large could Input Delay be?

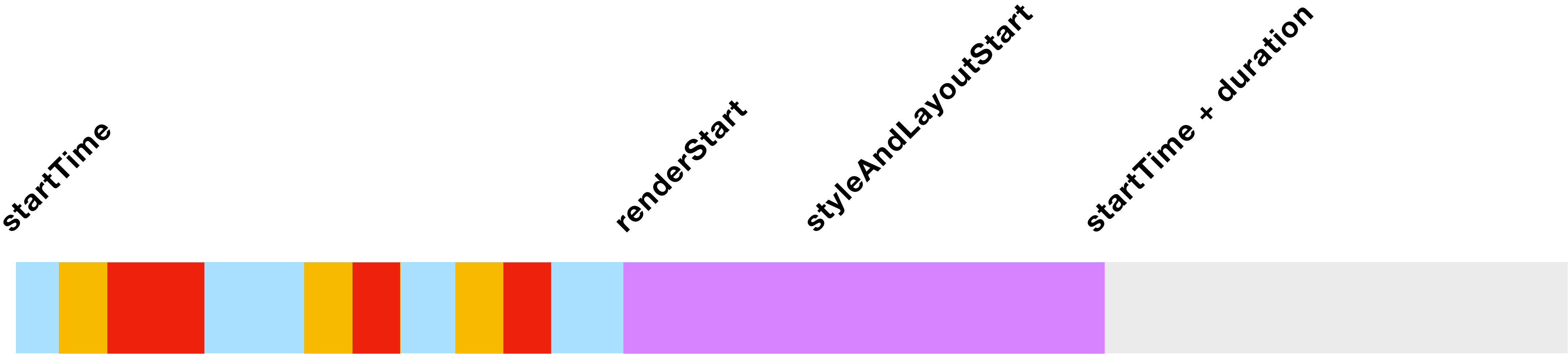


Longest LoAF = Maximum Potential Input Delay

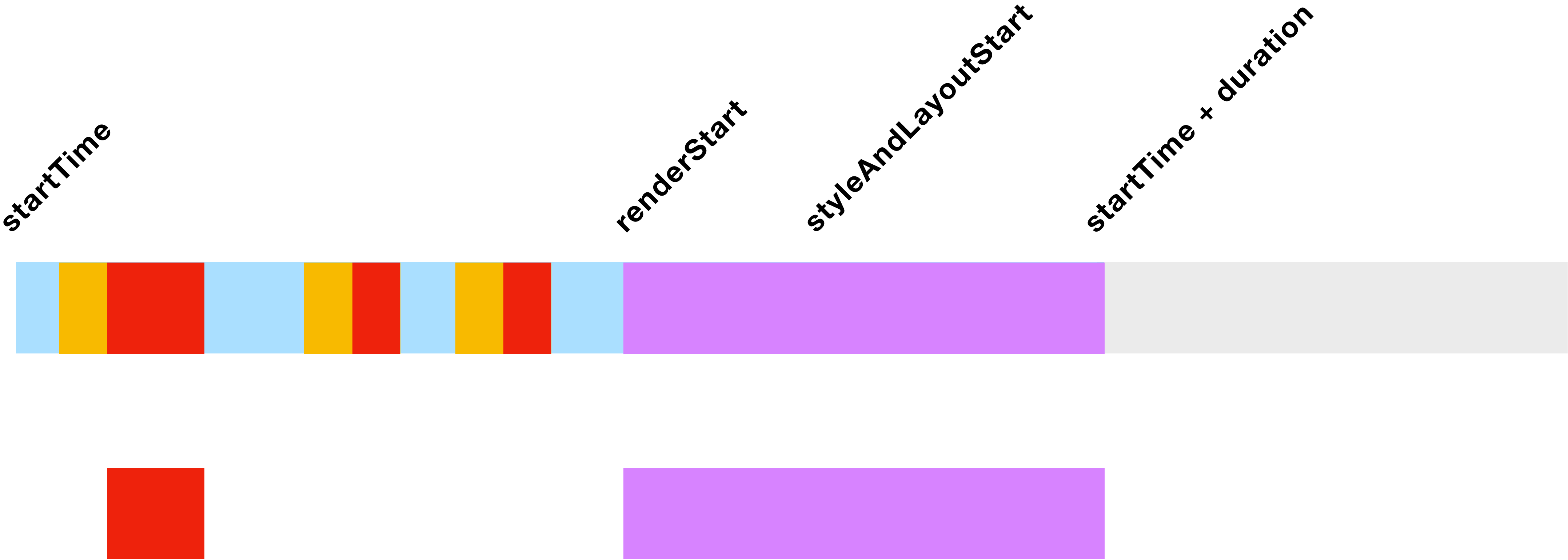
Estimate responsiveness to User Input



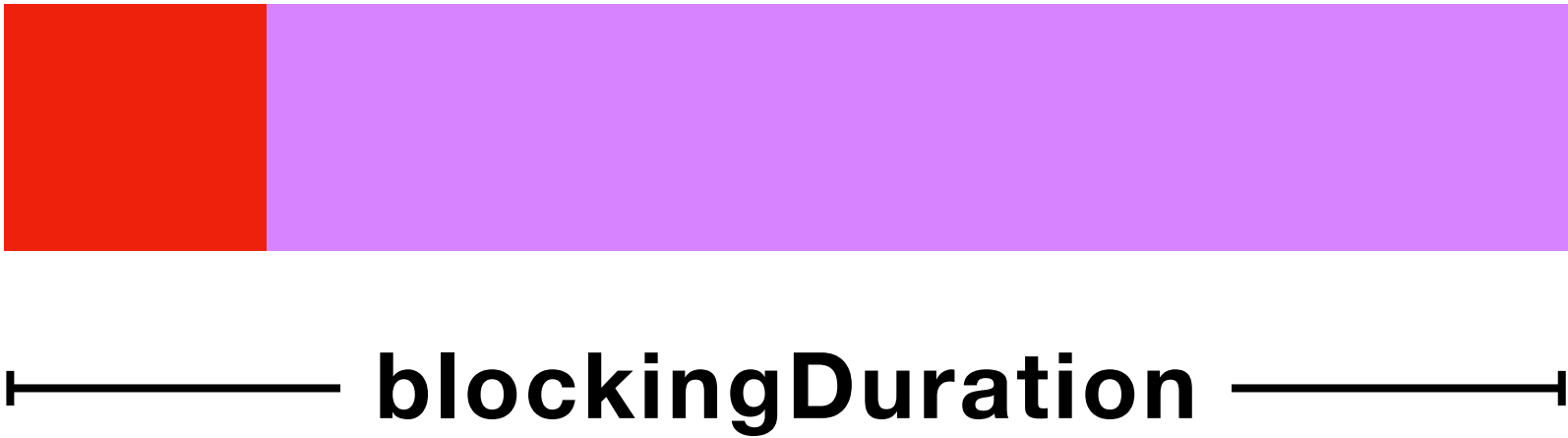
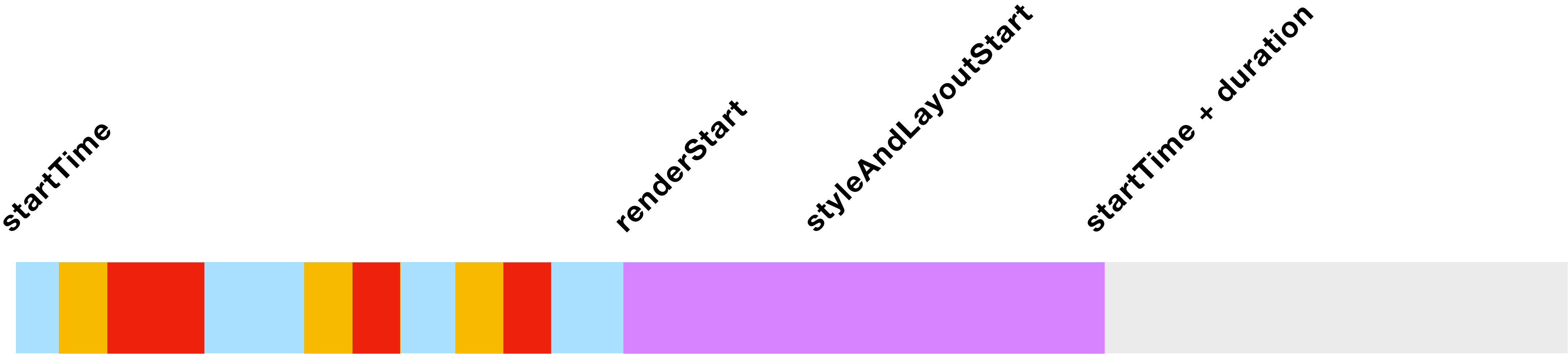
Estimate responsiveness to User Input



Estimate responsiveness to User Input



Estimate responsiveness to User Input



Total Blocking Duration (TBD)

```
totalBlockingDuration = 0;  
LoAFs  
    .forEach(entry => {  
        totalBlockingDuration += entry.blockingDuration  
    });
```

It's a slightly naive approach but we launched TBD in SpeedCurve as a experiment with the aiming of learning and refining in public

Seems to match Main Thread busyness ok

TOTAL BLOCKING DURATION, JS LONG TASKS | 75TH

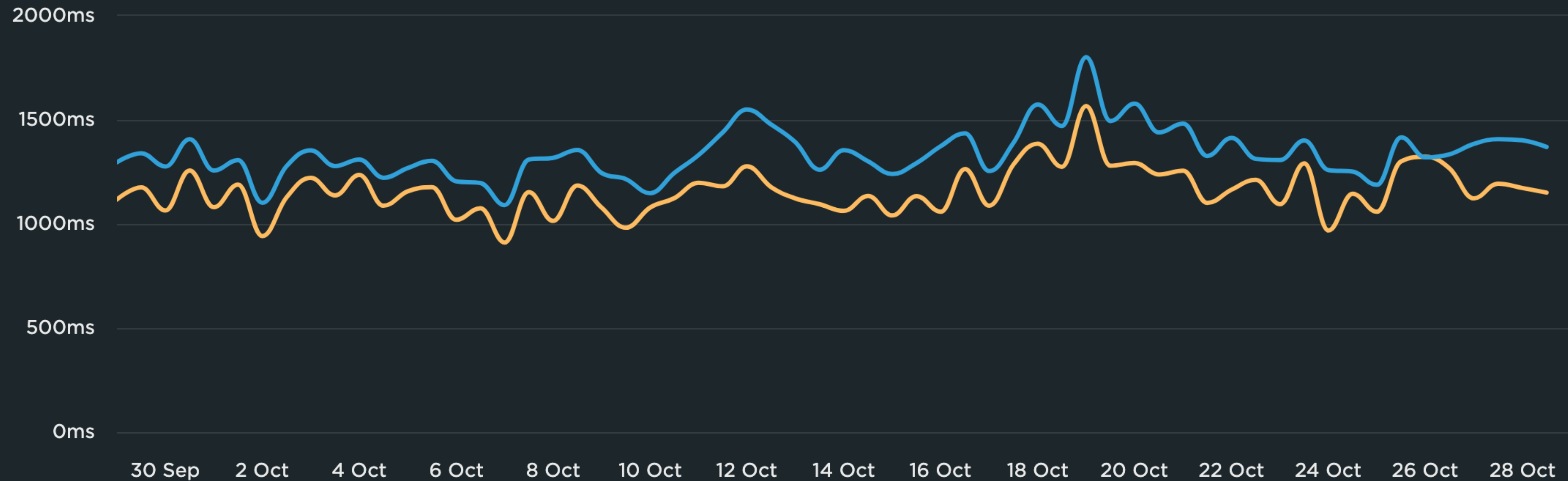


Total Blocking Duration

1,151ms

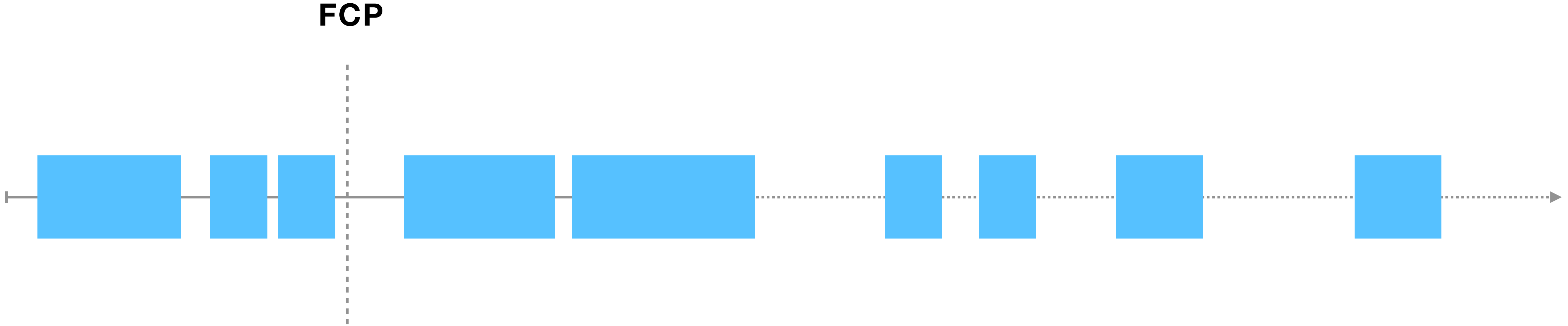
JS Long Tasks

1,318ms

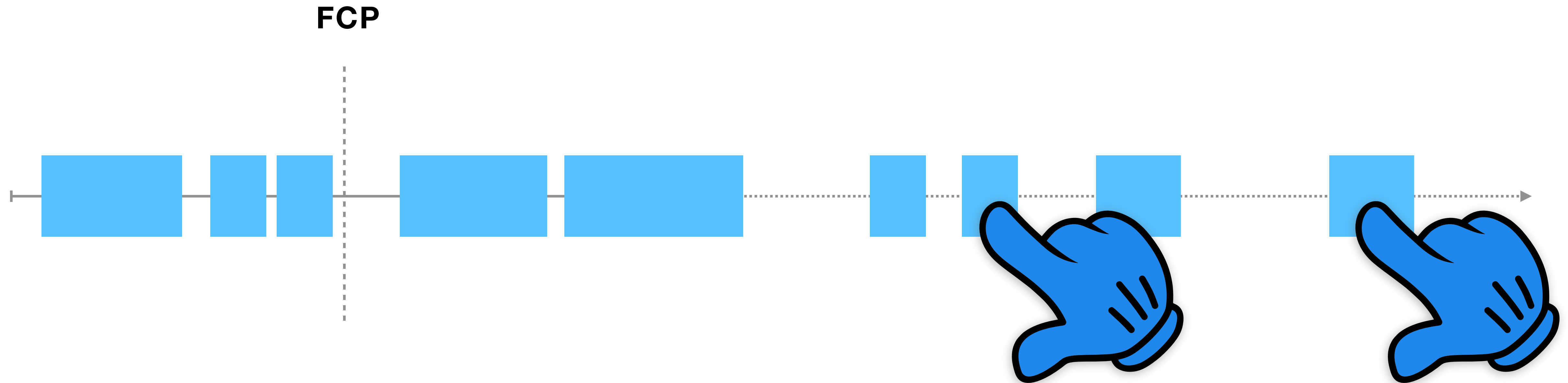


— Total Blocking Duration (RUM) — JS Long Tasks (RUM)

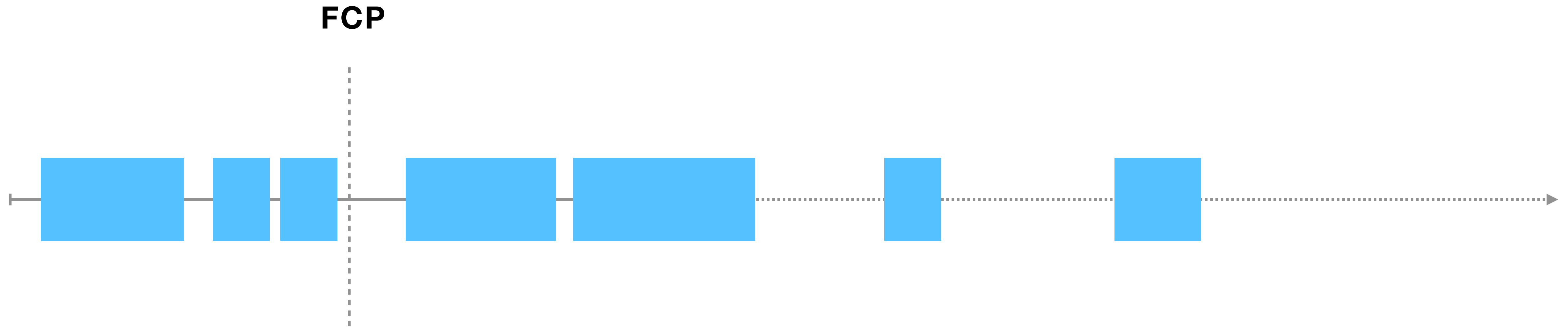
Refining Total Blocking Duration



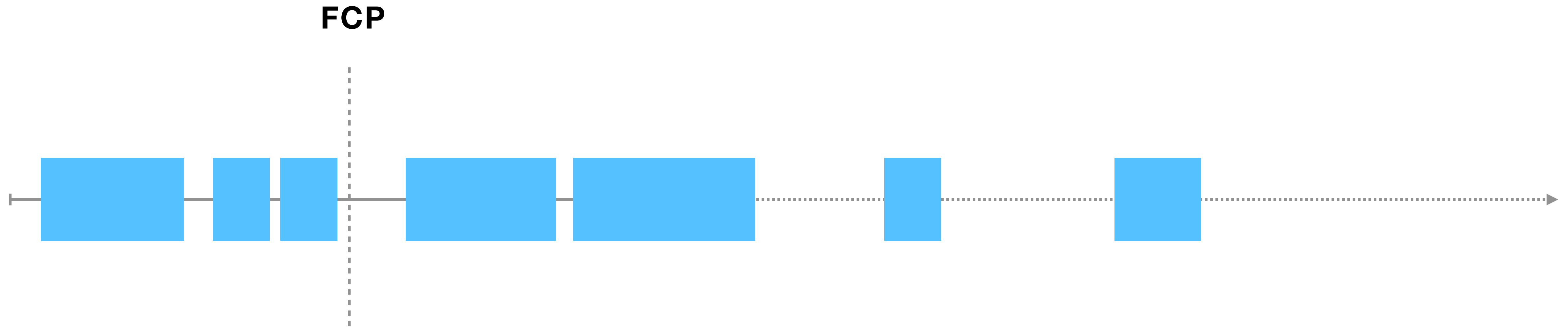
Exclude LoAFs linked to interactions



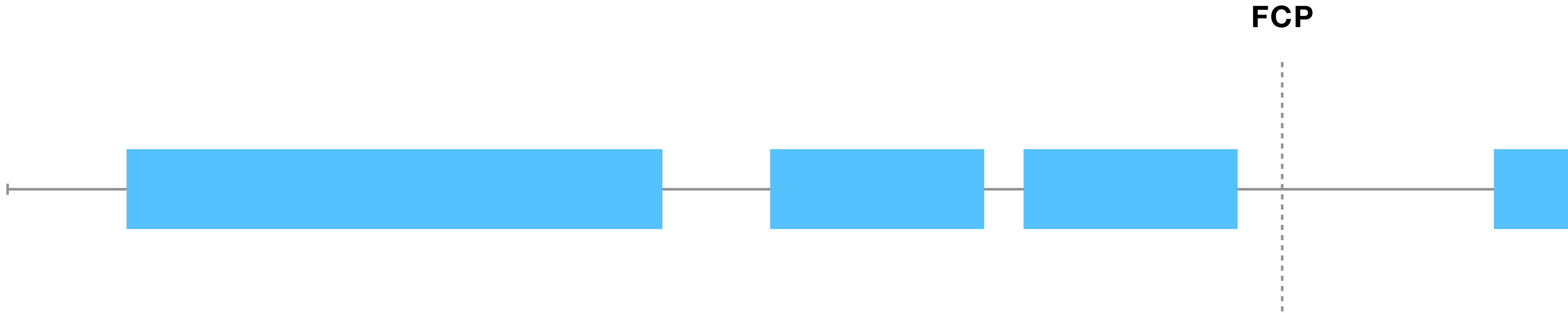
What about LoAFs before FCP?



What about LoAFs before FCP?



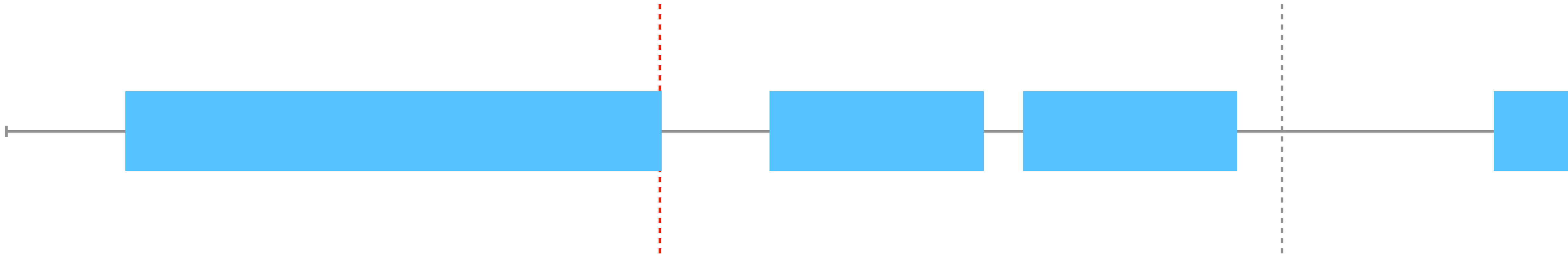
What about LoAFs before FCP?



What about LoAFs before FCP?

Could have
rendered at
this point

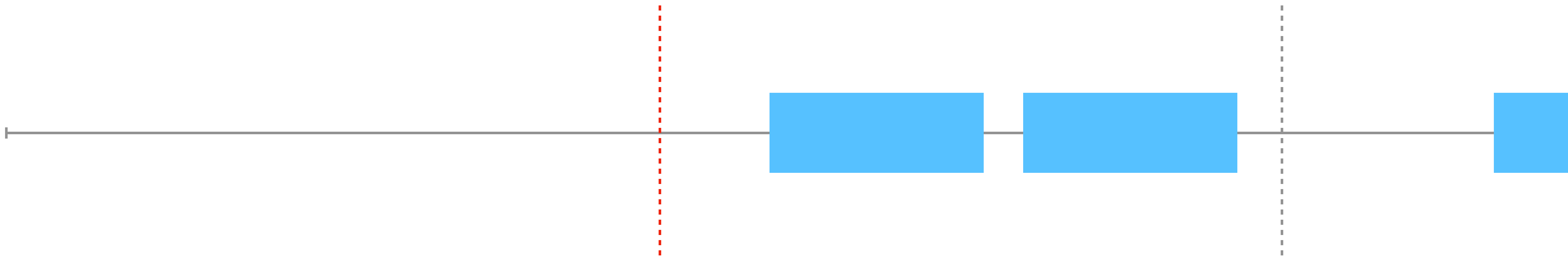
FCP



What about LoAFs before FCP?

Could have
rendered at
this point

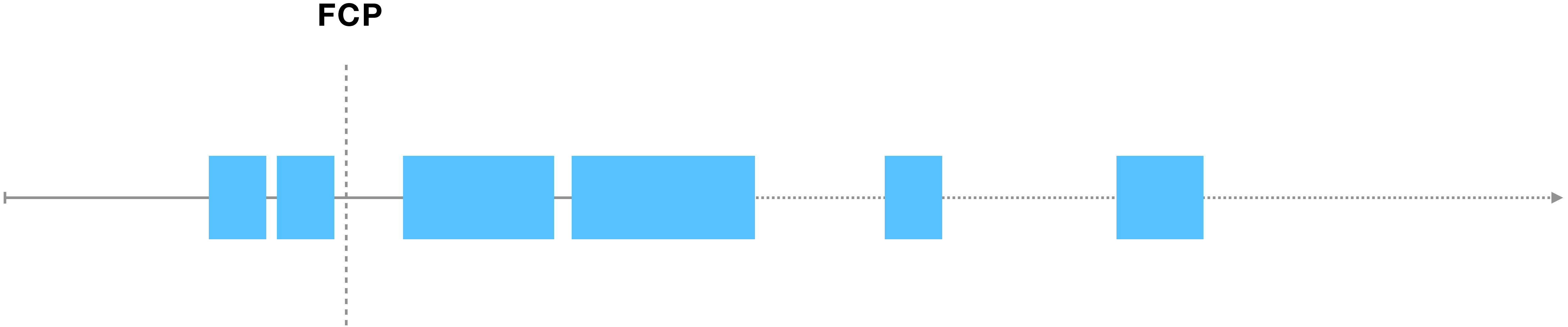
FCP



Currently thinking of just removing the first LoAF but...

...really interested in what every one else thinks too

Updated Total Block Duration calculation



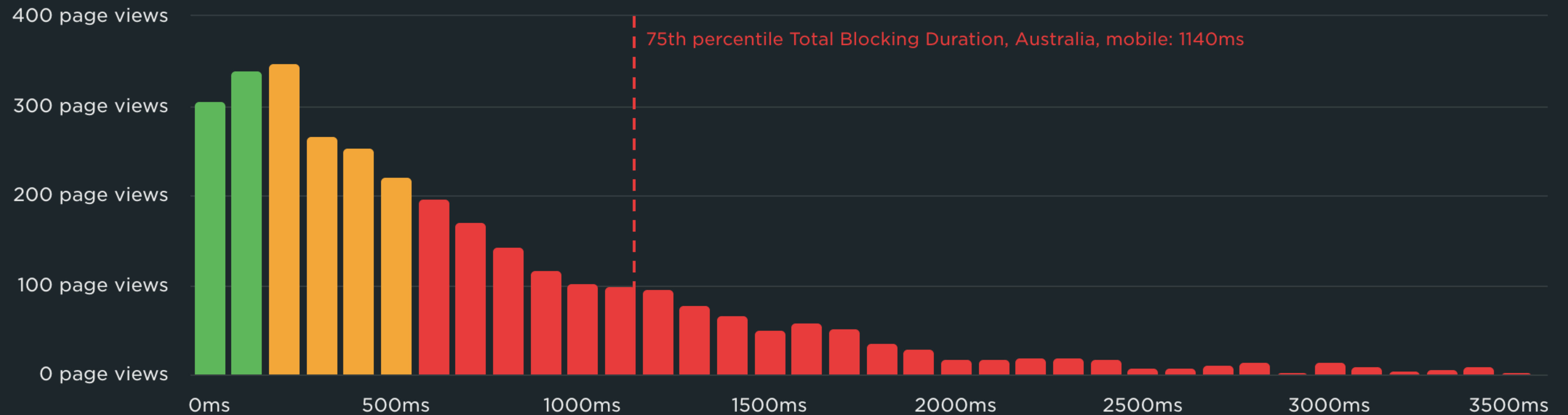
Updated Total Block Duration calculation

```
totalBlockingDuration = 0;
LoAFs
    .slice(1)
    .filter(entry => entry.firstUIEventTimestamp == 0)
    .forEach(entry => {
        totalBlockingDuration += entry.blockingDuration
    });
```


Default thresholds need reviewing too

TBD DISTRIBUTION

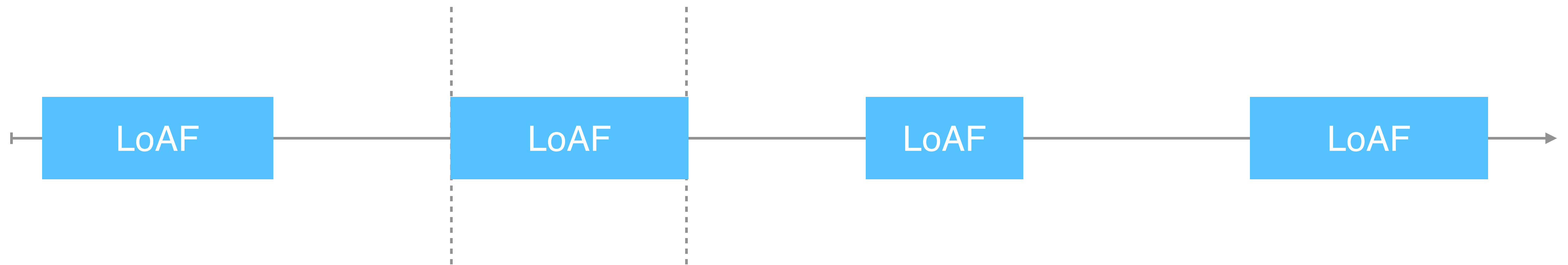
The distribution of your users Total Blocking Duration (TBD) time.



LoAF has some novel uses...

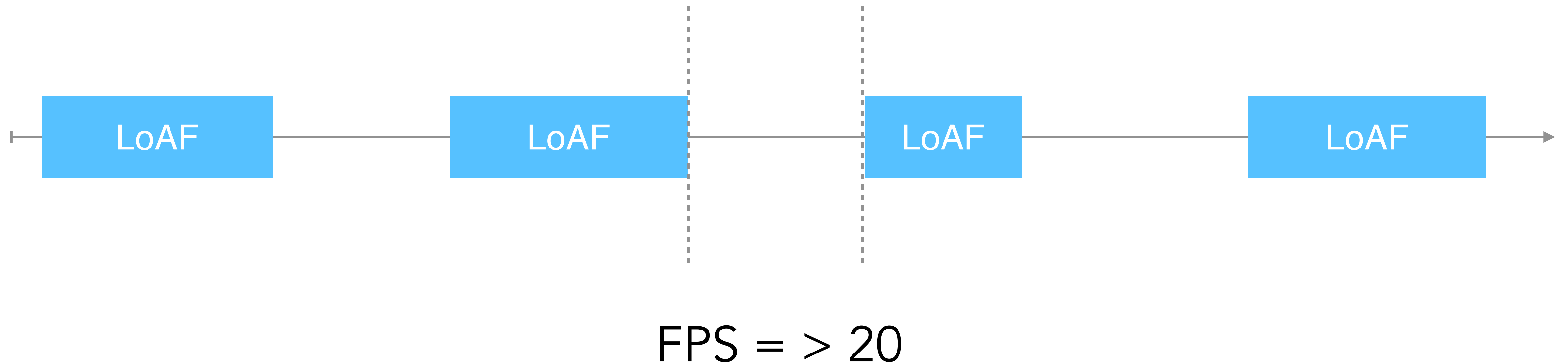


Approximate Frame Rate Calculations



$$\text{FPS} = 1,000\text{ms} / \text{LoAF.duration}$$

Approximate Frame Rate Calculations



LoAF gives us useful data

- How many frames were delayed and how long for?
- Where did the frames spend their time?
- What was the longest time a visitor might wait for a response?
- How long was the main thread unable to respond to user input?
- What was the approximate frame rate?
- And more...

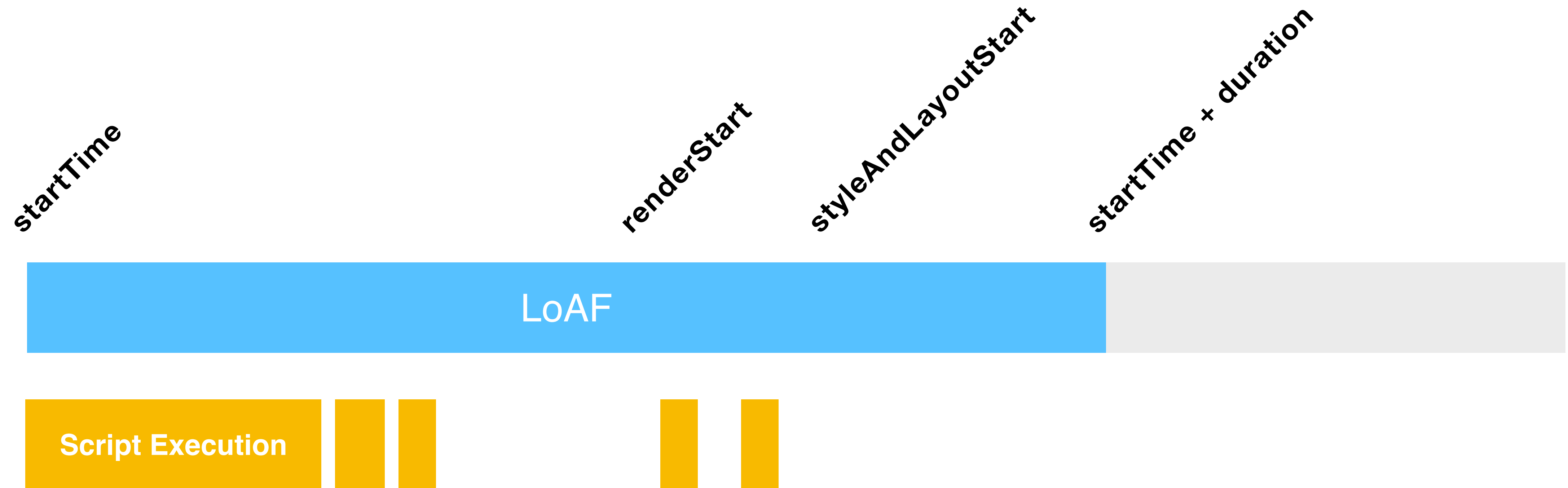


It can help understand JS runtime costs

Example LoAF entry

```
{  
  "name": "long-animation-frame",  
  "entryType": "long-animation-frame",  
  "navigationId": "95558f7f-38d7-4a06-b15e-67ccf1dec62e",  
  "startTime": 300.399999997615814,  
  "duration": 1573.40000000357628,  
  "blockingDuration": 605.986,  
  "renderStart": 1540,  
  "styleAndLayoutStart": 1540.10000000238419,  
  "paintTime": 1873.8000000011921,  
  "firstUIEventTimestamp": 0,  
  "scripts": []  
},
```


Entry for each script execution > 5ms

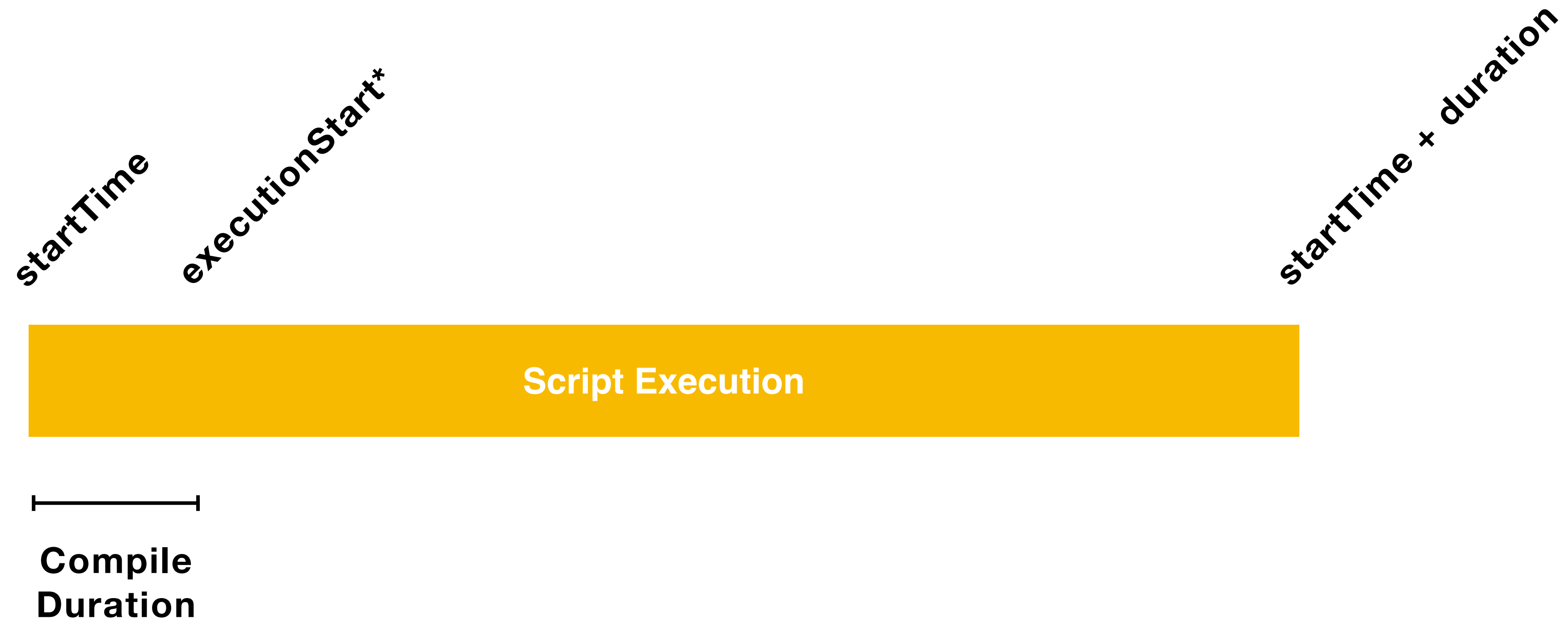


Entry for each script execution > 5ms



*executionStart is 'muted' for Cross Origin scripts

Entry for each script execution > 5ms



*executionStart is 'muted' for Cross Origin scripts

Other timing information

forcedStyleAndLayoutDuration

getComputedStyle & getBoundingClientRect for the browser to recalculate Style and Layout

pauseDuration

Synchronous XHR can force the browser to pause

What script executed?

```
"sourceURL": "https://www.example.com/script.js",  
"sourceFunctionName": "M",  
"sourceCharPosition": 98556
```


What script executed?

```
"sourceURL": "https://www.example.com/script.js",  
"sourceFunctionName": "M",  
"sourceCharPosition": 9855
```



Top level entry point

Why did it execute?

```
"invoker": "DOMWindow.onclick",  
"invokerType": "event-listener",
```

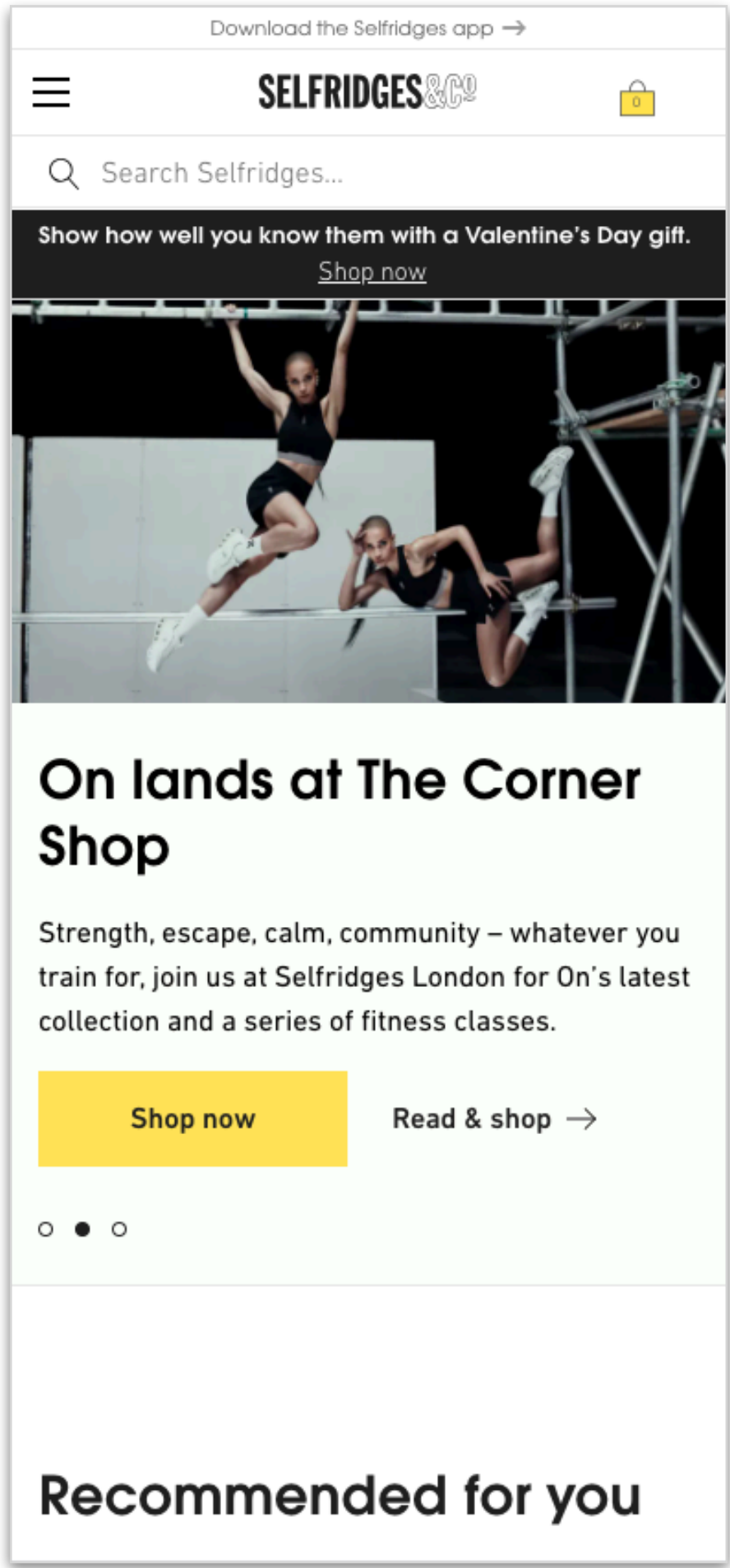

Aggregate the Data... Roll up the Data...

sourceURL	Total Duration (ms)	ForcedStyleAndLayout Total Duration (ms)	Occurrences
https://www.selfridges.com/static-mfe-clp/_next/static/chunks/914-ff512dbd6d857625.js	4,555	139	30
https://js-cdn.dynatrace.com/jstag/164ae1b51de/bf67380nlf/fb4fc3f4b31ef9b_complete.js	1,328	8	18
https://www.googletagmanager.com/gtag/js?id=AW-989335448&l=gDataLayer&cx=c>m=4e5am0	1,021	0	2
https://www.selfridges.com/static-mfe-clp/_next/static/chunks/937-c796c646a443351b.js	938	0	1
https://cdn.quantummetric.com/qscripts/quantum-selfridges.js	842	0	10
https://www.selfridges.com/static-mfe-clp/_next/static/chunks/d8ec93b9-980f704a97712473.js	767	0	3
https://www.googletagmanager.com/gtag/js?id=DC-5921516&l=gDataLayer&cx=c>m=4e5am0	618	0	1
https://tags.tiqcdn.com/utag/selfridges/main/prod/utag.js	543	119	5
https://tags.pw.adn.cloud/SMVEJB/activation.js	226	0	2
https://f.monetate.net/trk/4/s/a-26b02505/p/selfridges.com/1584236768-0?	204	0	1
https://www.googletagmanager.com/gtag/js?id=G-R05V82D63H&l=gDataLayer	156	0	2
https://www.selfridges.com/static-mfe-clp/_next/static/chunks/1dd3208c-191771bcb3be71a1.js	131	0	6
https://tags.tiqcdn.com/utag/selfridges/main/prod/utag.416.js?utv=ut4.51.202210200808	60	0	1
https://se.monetate.net/js/2/a-26b02505/p/selfridges.com/custom.js	57	0	2
https://analytics.tiktok.com/i18n/pixel/static/main.MWI5ZTBkMTQ4MA.js	38	0	3
https://www.selfridges.com/GB/en/	29	0	1
	21	0	1
https://cdn.attraqt.io/xo.all-2.min.js	7	0	1
https://sb.monetate.net/img/1/p/1581/5347147.js/monetate.c.cr.js	6	0	1
https://static.queue-it.net/script/queueconfigloader.min.js	5	0	1
https://www.selfridges.com/static-mfe-clp/_next/static/chunks/61721e05-4efc862ff71175ec.js	5	0	1

Aggregate the Data... Roll up the Data...

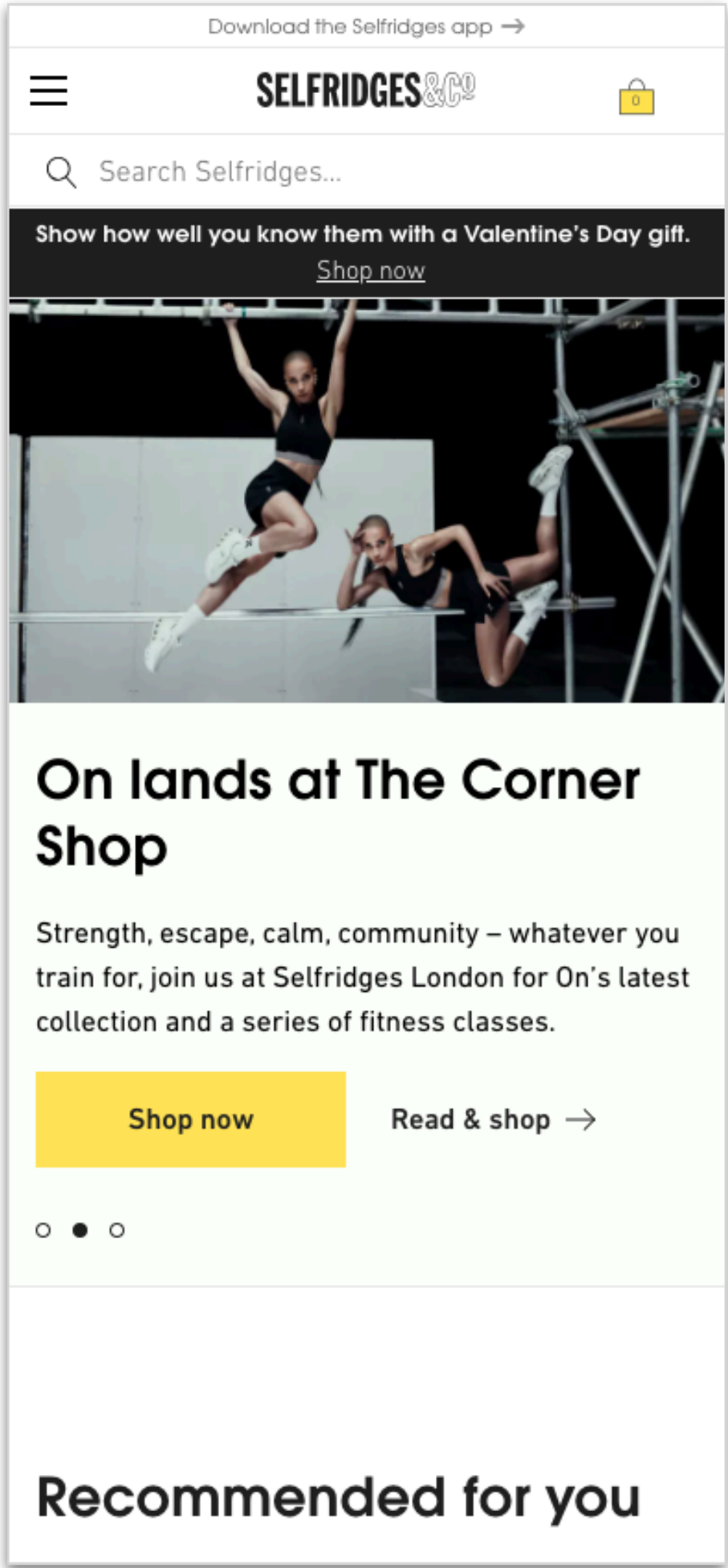
sourceURL	Total Duration (ms)	ForcedStyleAndLayout Total Duration (ms)	Occurrences
https://www.selfridges.com/static-mfe-clp/_next/static/chunks/914-ff512dbd6d857625.js	4,555	139	30
https://js-cdn.dynatrace.com/jstag/164ae1b51de/bf67380nlf/fb4fc3f4b31ef9b_complete.js	1,328	8	18
https://www.googletagmanager.com/gtag/js?id=AW-989335448&l=gDataLayer&cx=c>m=4e5am0	1,021	0	2
https://www.selfridges.com/static-mfe-clp/_next/static/chunks/937-c796c646a443351b.js	938	0	1
https://cdn.quantummetric.com/qscripts/quantum-selfridges.js	842	0	10
https://www.selfridges.com/static-mfe-clp/_next/static/chunks/d8ec93b9-980f704a97712473.js	767	0	3
https://www.googletagmanager.com/gtag/js?id=DC-5921516&l=gDataLayer&cx=c>m=4e5am0	618	0	1
https://tags.tiqcdn.com/utag/selfridges/main/prod/utag.js	543	119	5
https://tags.pw.adn.cloud/SMVEJB/activation.js	226	0	2
https://f.monetate.net/trk/4/s/a-26b02505/p/selfridges.com/1584236768-0?	204	0	1
https://www.googletagmanager.com/gtag/js?id=G-R05V82D63H&l=gDataLayer	156	0	2
https://www.selfridges.com/static-mfe-clp/_next/static/chunks/1dd3208c-191771bcb3be71a1.js	131	0	6
https://tags.tiqcdn.com/utag/selfridges/main/prod/utag.416.js?utv=ut4.51.202210200808	60	0	1
https://se.monetate.net/js/2/a-26b02505/p/selfridges.com/custom.js	57	0	2
https://analytics.tiktok.com/i18n/pixel/static/main.MWI5ZTBkMTQ4MA.js	38	0	3
https://www.selfridges.com/GB/en/	29	0	1
	21	0	1
https://cdn.attraqt.io/xo.all-2.min.js	7	0	1
https://sb.monetate.net/img/1/p/1581/5347147.js/monetate.c.cr.js	6	0	1
https://static.queue-it.net/script/queueconfigloader.min.js	5	0	1
https://www.selfridges.com/static-mfe-clp/_next/static/chunks/61721e05-4efc862ff71175ec.js	5	0	1

Are 3rd-party tags really the problem?



	1st Party	3rd Party
Total Script Duration (ms)	6,446	5,111

May generate many entries per page



CPU Slowdown	0x	4x	6x
LOAF Entries	21	32	37
ScriptTiming Entries	31	39	47

Example only:

Real world values will depend on device CPU, network connectivity and duration of observation

Helps answer questions such as

- Which scripts have the most impact on visitors experience?
- Which scripts are forcing style and layout operations?
- Which scripts delay FCP or LCP or affect a visitor completing a task?
- Are my 1st Party or 3rd-party scripts a problem?

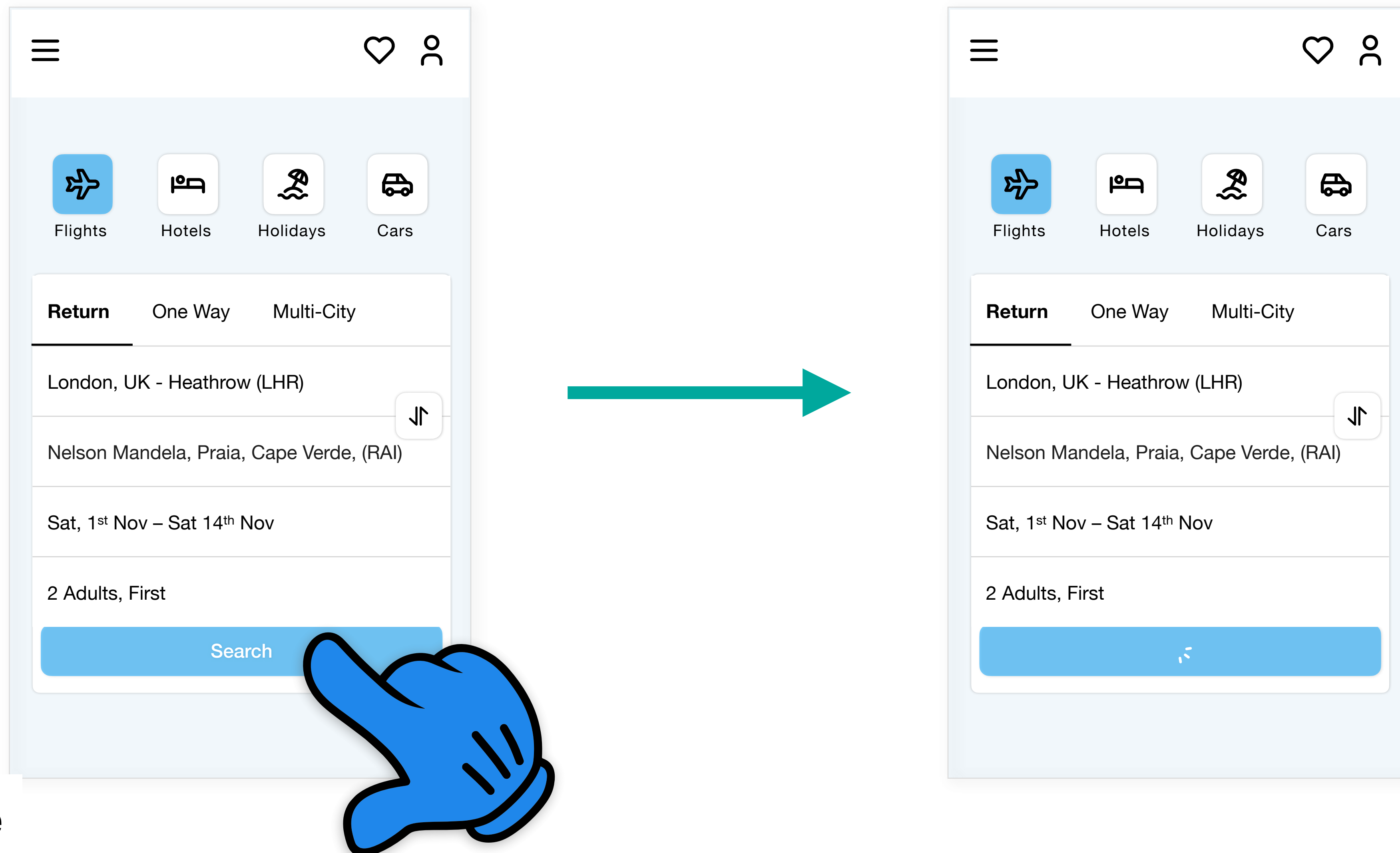


PRESS

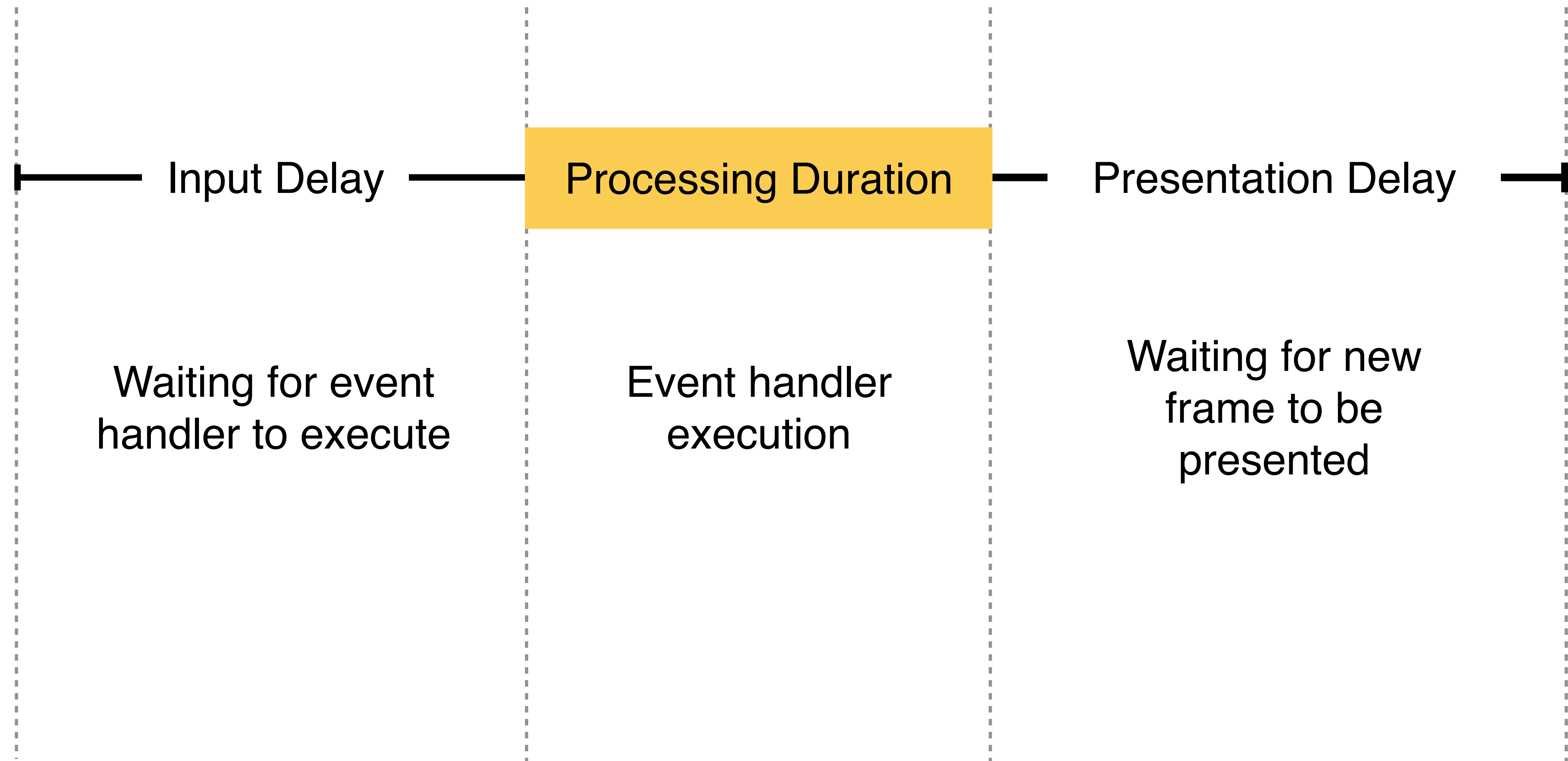
LoAF can help diagnose slow interactions

Interaction to Next Paint (INP)

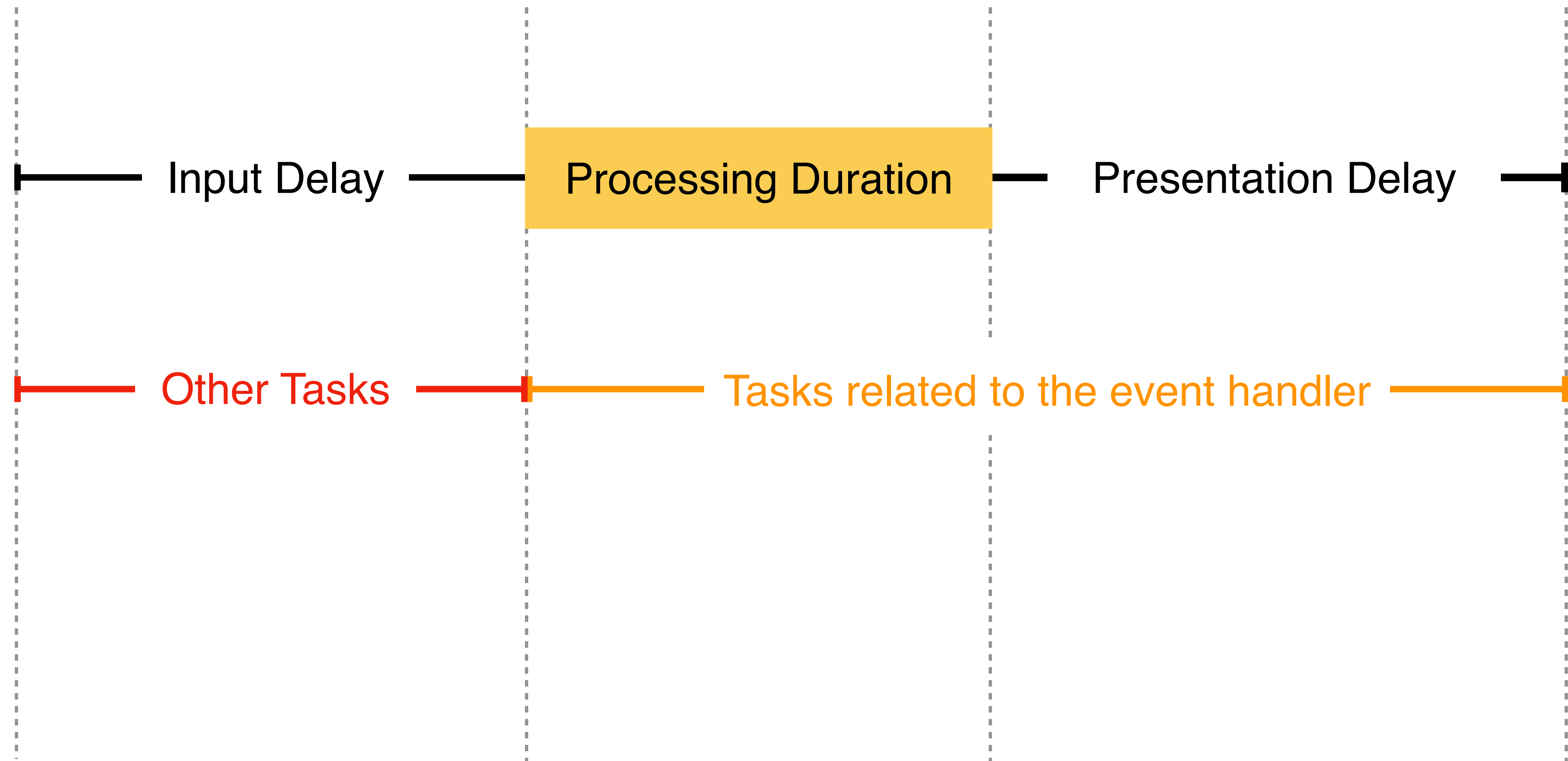
Time between a visitor interacting and the next frame being presented



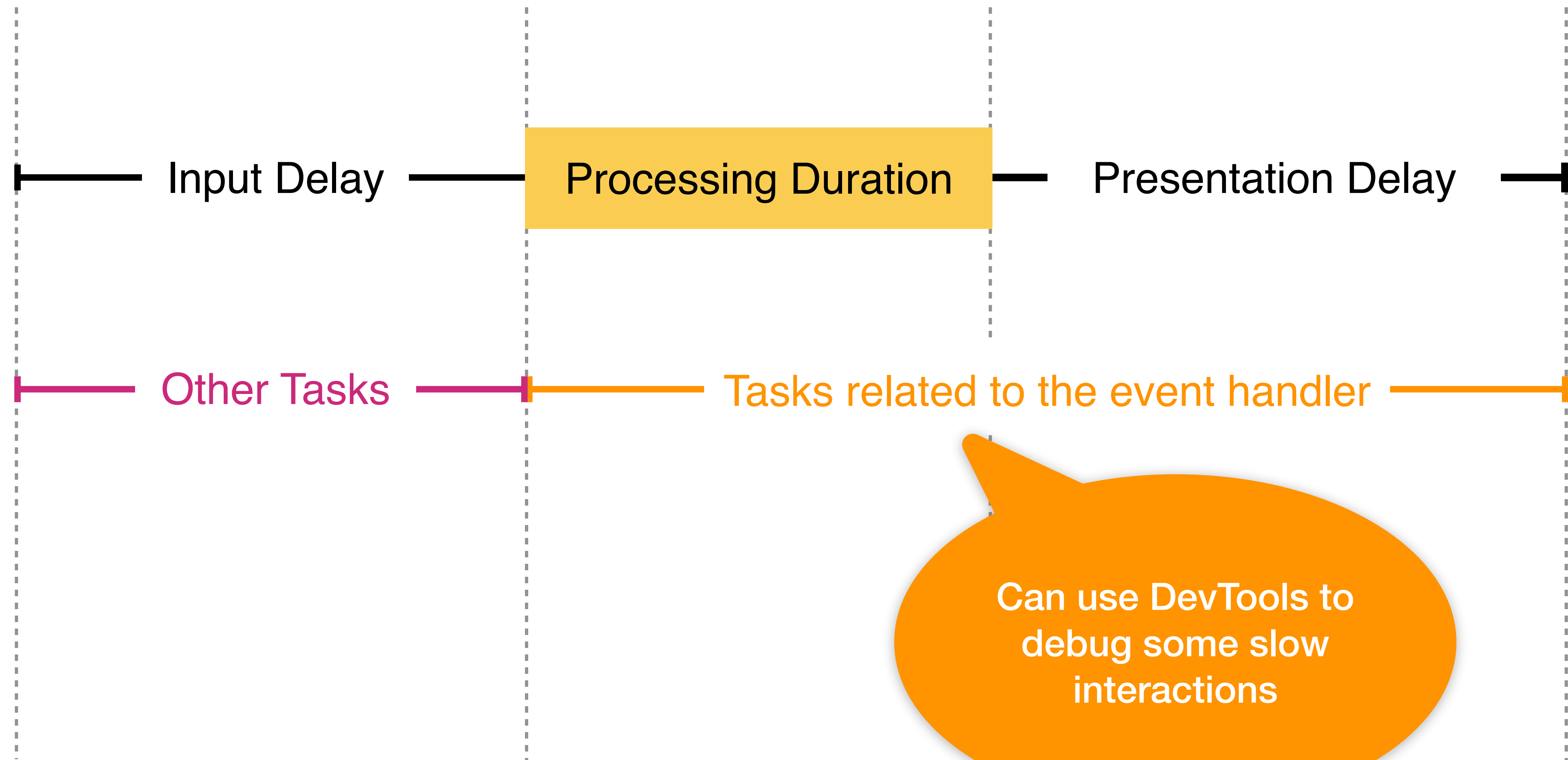
INP has Three Phases



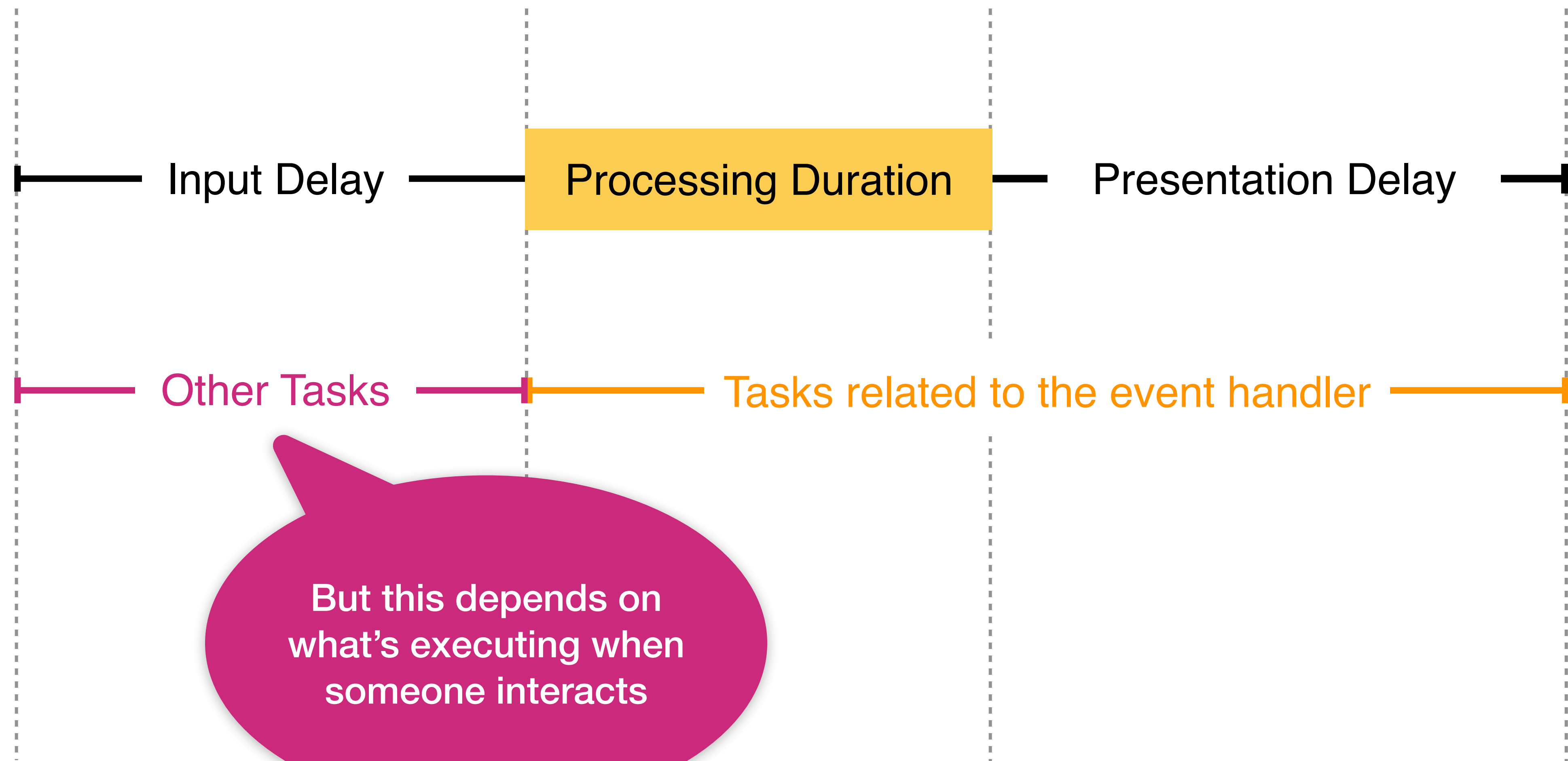
INP has Three Phases



INP has Three Phases



INP has Three Phases



INP BY PAGE LABELS

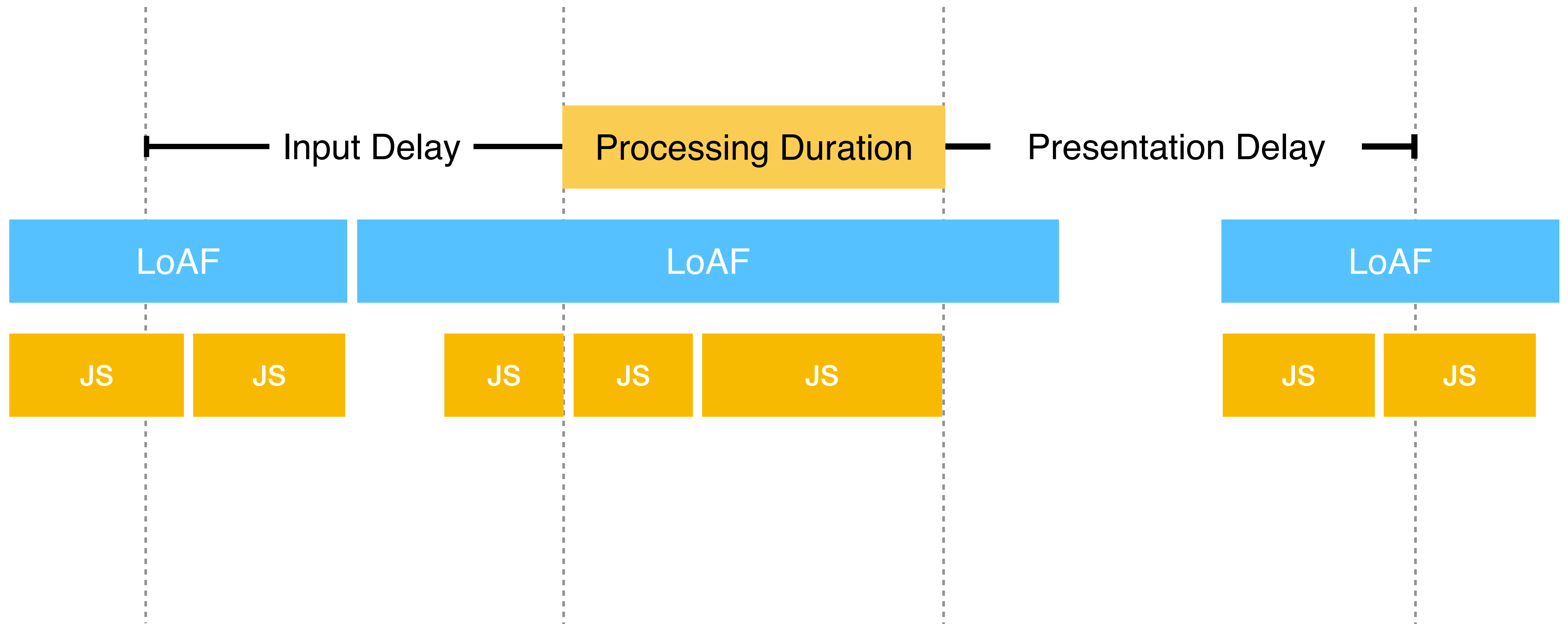
INP time split into subparts. Explore which dimensions affect INP the most.

PAGE LABELS	% OF PAGE VIEWS	INTERACTION TO NEXT PAINT	INPUT DELAY	PROCESSING DURATION	PRESENTATION DELAY
PDP	39.6%	136ms	41ms	23ms	48ms
PLP	27.3%	632ms	162ms	36ms	71ms
Blog	9.8%	304ms	47ms	65ms	52ms
Home	6.2%	160ms	60ms	35ms	55ms
My Account	4.1%	56ms	19ms	17ms	13ms
Basket	4%	136ms	33ms	51ms	43ms
Offers	3.9%	56ms	18ms	1ms	31ms
Checkout	3.8%	176ms	28ms	106ms	34ms
Locations	0.8%	120ms	33ms	16ms	44ms
Terms	0.4%	168ms	32ms	74ms	52ms

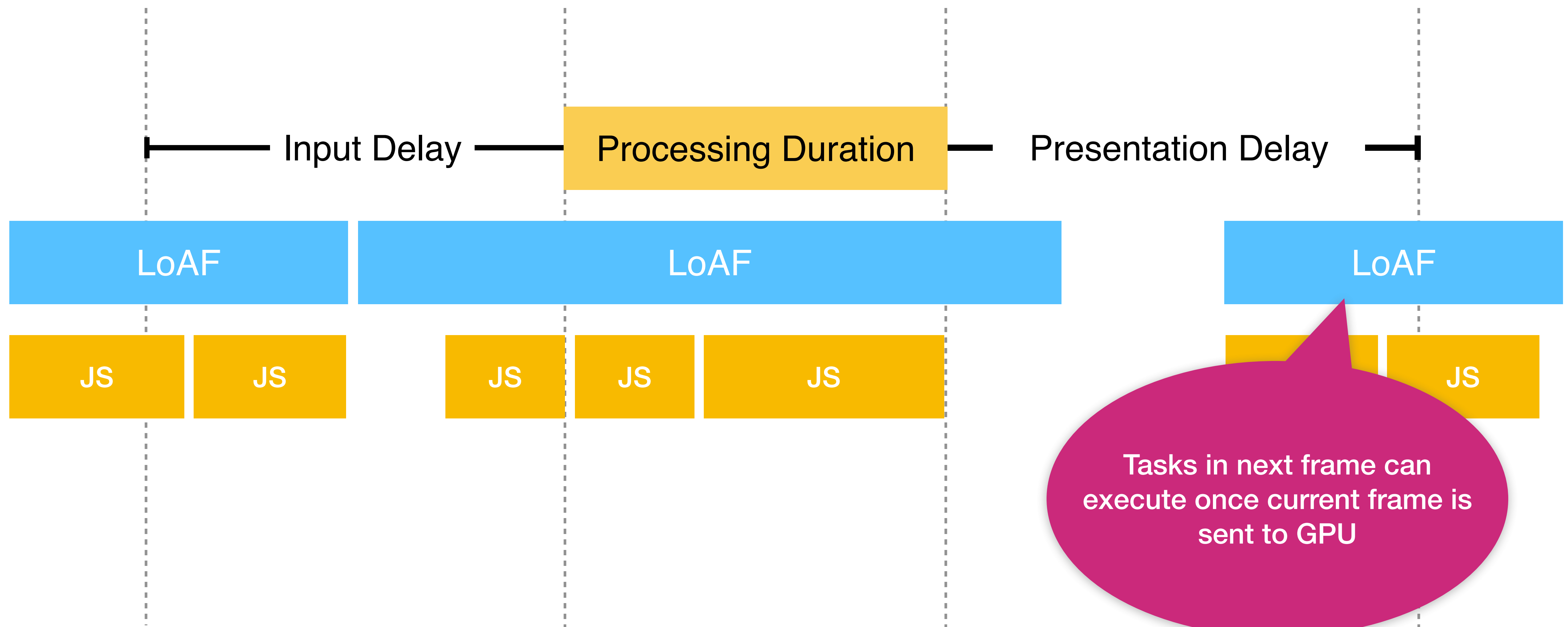
No direct way to link EventTiming & LoAF



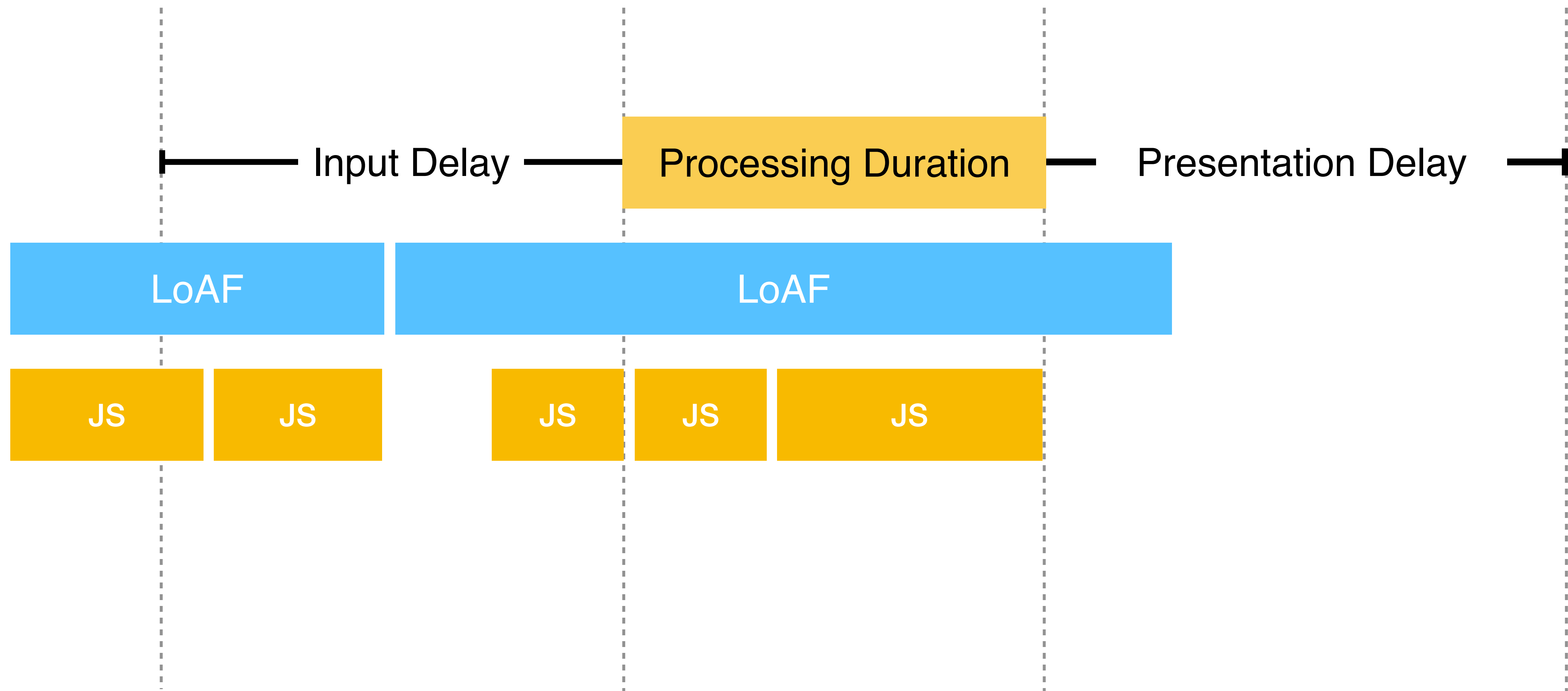
Use timestamps to map scripts to phases



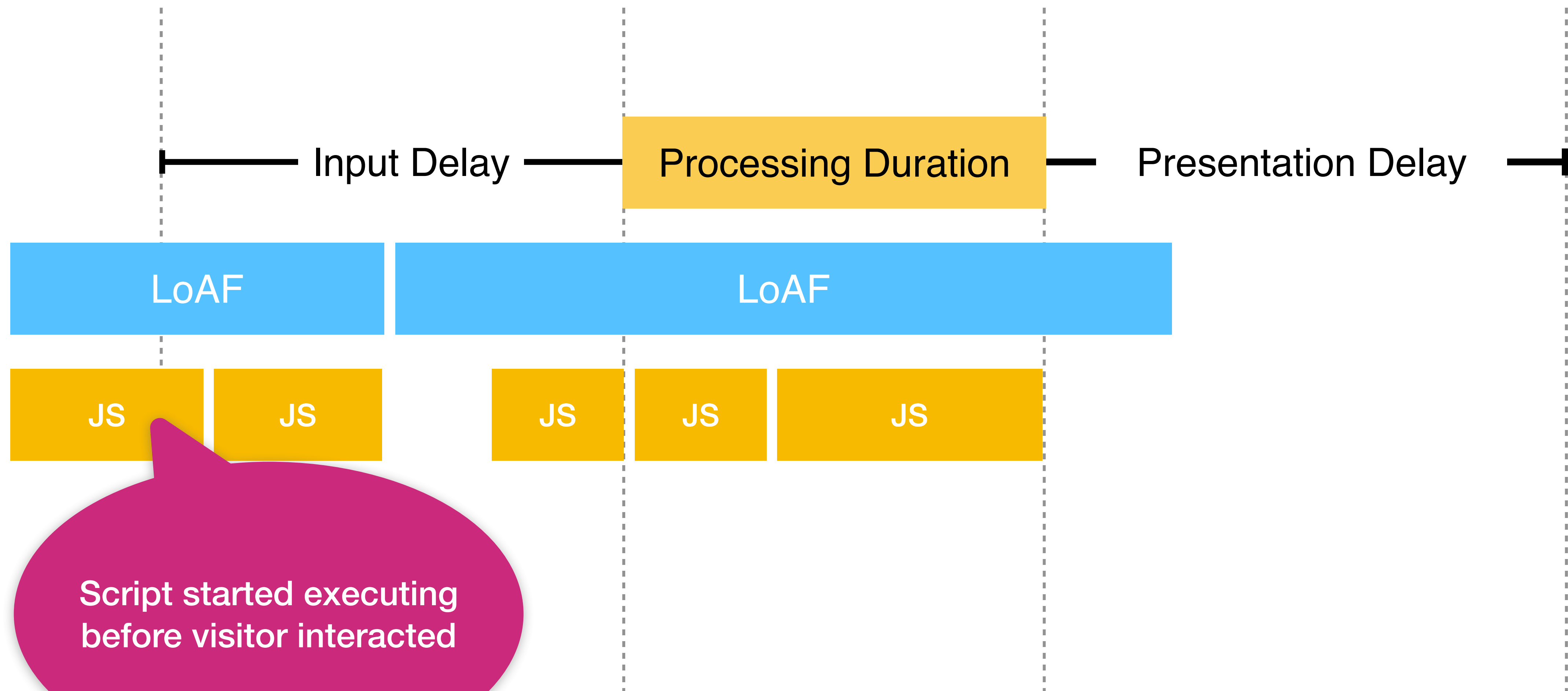
Discard LoAFs ending after Presentation



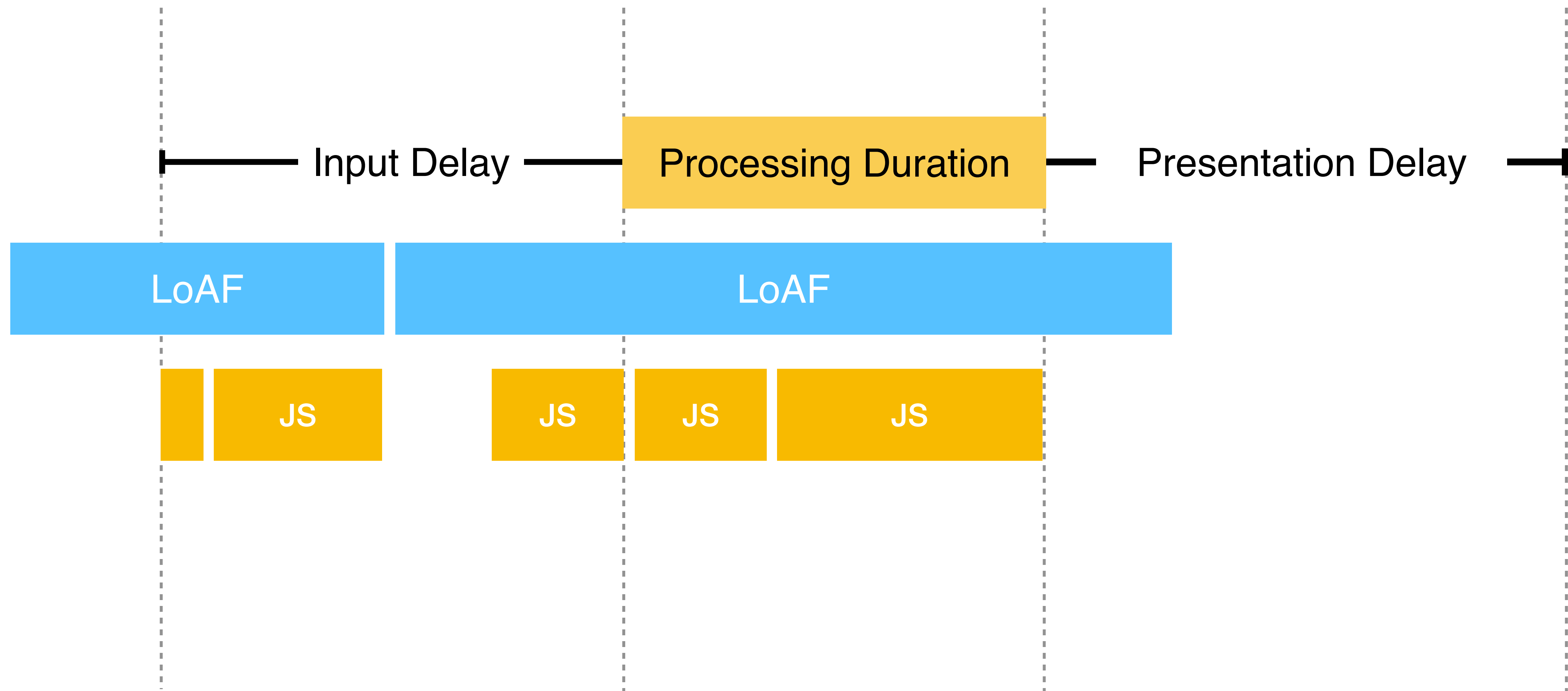
Discard LoAFs ending after Presentation



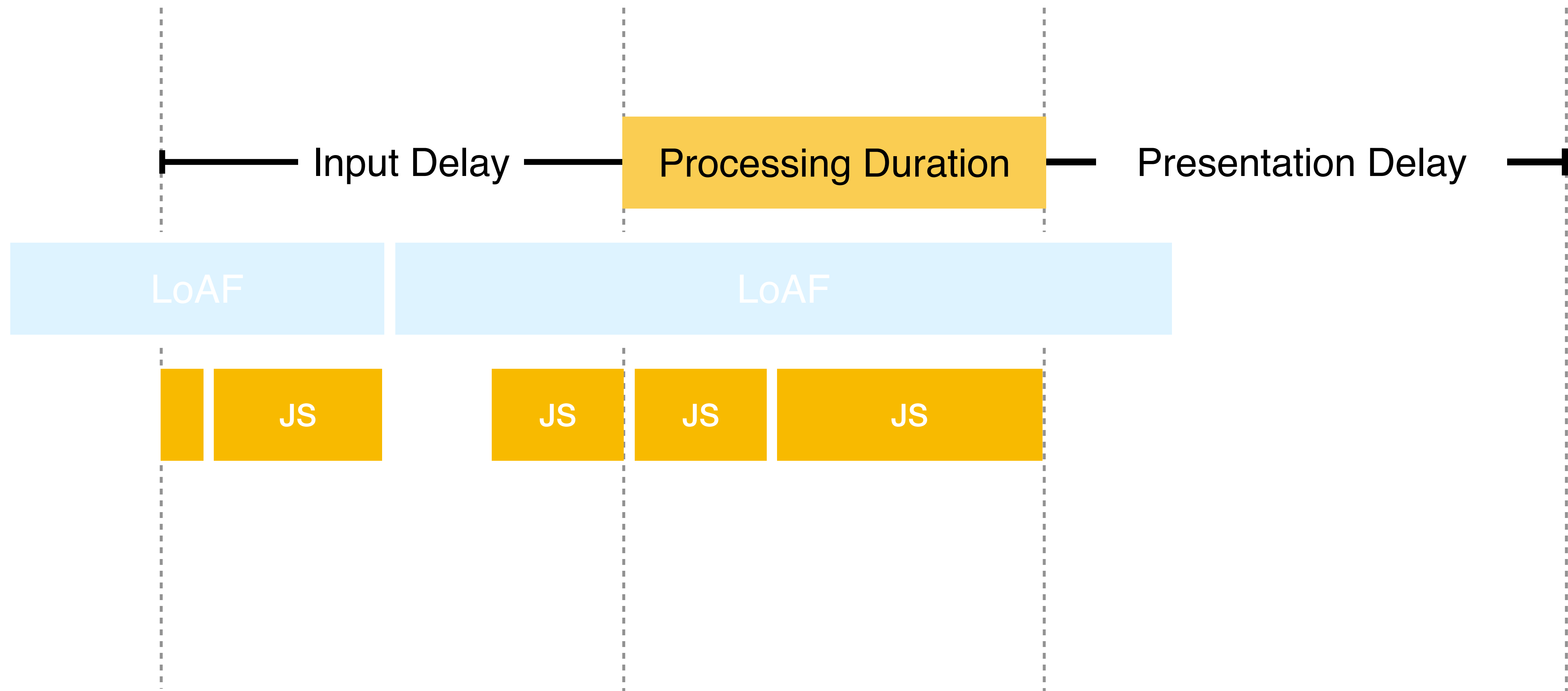
Exclude any script execution before Input



Exclude script execution before Input



Assign scripts that are left to INP phases



Assign scripts that are left to INP phases

phase	invokerType	duration (ms)	sourceURL
InputDelay	classic-script	119	https://tags.tiqcdn.com/utag/selfridges/main/prod/utag.js
InputDelay	event-listener	10	
InputDelay	event-listener	5	https://js-cdn.dynatrace.com/jstag/164ae1b51de/bf67380nlf/fb4fc3f4b31ef
InputDelay	event-listener	12	https://www.selfridges.com/NL/en/features/etc/designs/zg/selfridges-new/
ProcessingTime	event-listener	12	https://www.selfridges.com/NL/en/features/etc/designs/zg/selfridges-new/

Can help us answer questions such as:

- How are scripts affecting our visitors interactions?
- Which scripts have have the most impact across all interactions?
- Which are the slowest interaction handlers?
- Which scripts commonly delay interaction handlers

Attribution Issues: Empty sourceFunction

phase	invoker	invokerType	entryPoint	duration (ms)	sourceURL
ID	https://tags.tiqcdn.com/utag	classic-script		119	https://tags.tiqcdn.com/utag/selfridges/main/p
ID	SCRIPT[src=//tags.tiqcdn.cc	event-listener		10	
ID	#document.ontouchstart	event-listener		5	https://js-cdn.dynatrace.com/jstag/164ae1b51
ID	#document.ontouchstart	event-listener			https://www.selfridges.com/NL/en/features/etc
PT	BODY#selfridges-app.onclick	event-listener			www.selfridges.com/NL/en/features/etc

Entry points may be empty or minified function names

Attribution Issues: Empty sourceURLs

phase	invoker	invokerType	entryPoint	duration (ms)	sourceURL
ID	https://tags.tiqcdn.com/utag	classic-script		119	https://tags.tiqcdn.com/utag/selfridges/main/p
ID	SCRIPT[src=//tags.tiqcdn.cc	event-listener		10	
ID	#document.ontouchstart	event-listener		5	https://tags.tiqcdn.com/jstag/164ae1b51
ID	#document.ontouchstart	event-listener		12	https://tags.tiqcdn.com/jstag/164ae1b51/en/features/et
PT	BODY#selfridges-app.onclick	event-listener		12	https://tags.tiqcdn.com/jstag/164ae1b51/en/features/et

Sometimes sourceURLs
are empty

Attribution Issues: Wrappers

invoker	invokerType	sourceFunctionName	sourceURL	Total Duration	Occurences
#document.ontouchend	event-listener	nrWrapper	...com/browser-performance/newrelic.js	13	1
svg.onclick	event-listener	nrWrapper	...com/browser-performance/newrelic.js	16	2
DOMWindow.onmousemove	event-listener	nrWrapper	...com/browser-performance/newrelic.js	8	1
#document.onmousedown	event-listener	nrWrapper	...com/browser-performance/newrelic.js	10	1
#document.onpointerdown	event-listener	nrWrapper	...com/browser-performance/newrelic.js	28	2
DOMWindow.onclick	event-listener	nrWrapper	...com/browser-performance/newrelic.js	136	1
A.onclick	event-listener	nrWrapper	...com/browser-performance/newrelic.js	10	2
BODY.onclick	event-listener	nrWrapper	...com/browser-performance/newrelic.js	26	2
#document.ontouchstart	event-listener	nrWrapper	...com/browser-performance/newrelic.js	13	1
#document.onclick	event-listener	nrWrapper	...com/browser-performance/newrelic.js	14	2
DIV#root.onclick	event-listener	nrWrapper	...com/browser-performance/newrelic.js	167	1

Attribution Issues: **Versioned scripts**

Versioned / hashed resources are common in modern build processes

`.../_next/static/chunks/1dd3208c-191771bcb3be71a1.js`

`.../_next/static/chunks/914-ff512dbd6d857625.js`

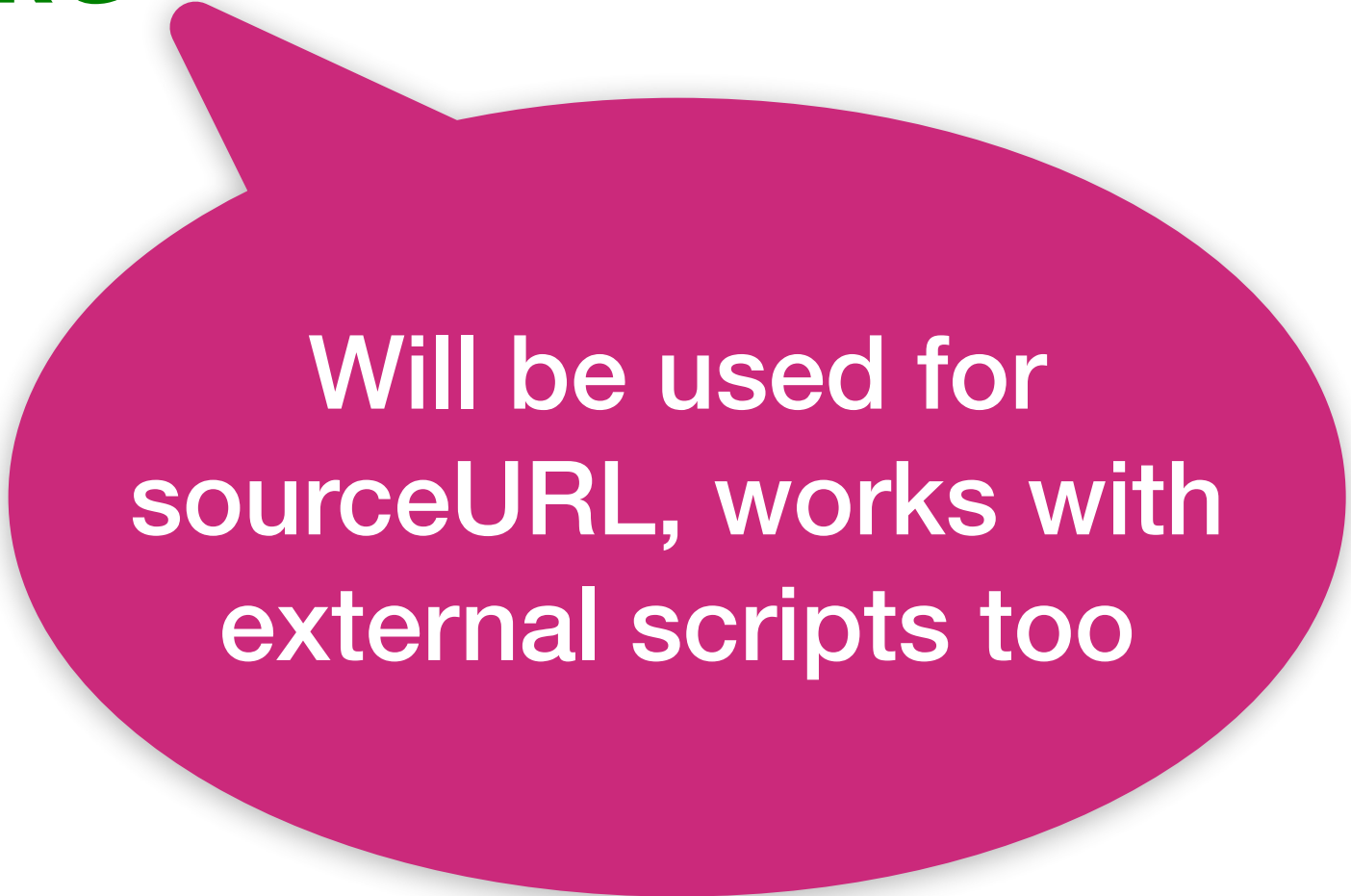
`.../_next/static/chunks/937-c796c646a443351b.js`

They can make it hard to track resource performance over time

(p r o g r a m : 2)

Name sources using `//# sourceMappingURL`

```
<script type="text/javascript">
  document.body.addEventListener("click", function(){
    for(var a=Date.now()+2E3;Date.now()<a;);
  });
  //# sourceMappingURL=Use-the-sourceURL-Luke
</script>
```



Will be used for
sourceURL, works with
external scripts too

Example: <https://andydavies.github.io/agent-tests/loaf/sourceurl.html>

No Script Entries for

- **Extensions**

For privacy reasons we can't know what extensions a visitor is using
Chrome on Android doesn't support extensions

- **Garbage Collection**

Tickets open for opaque attribution of these

WebPerf Snippets

Introduction

Core Web Vitals >

Loading v

TTFB

Content Visibility

Event Processing Time

Find above the Fold Lazy Loaded Images

Find Images with Lazy and Fetchpriority

Find Non Lazy Loaded Images Outside of the Viewport

Find Render Blocking Resources

First and Third Party Script Info

First and Third Party Script Timings

Fonts Preloaded and Used above the Fold

Interaction > Long Animation Frames Helpers

LoAF Helpers: Advanced Debugging in DevTools

While the [basic Long Animation Frames \(LoAF\)](#) snippet is great for capturing raw data, we often need more powerful tools to analyze and debug directly in the console without manual data processing.

This snippet installs a loafHelpers object on window, providing a suite of utility functions to filter, analyze, and export captured LoAF information in a much more convenient and efficient way.

Key Features

- Instant Summaries: Get an overview of the number of long frames, their durations, and severity.
- Culprit Identification: Quickly find the slowest scripts and those that are most significantly blocking rendering.
- Dynamic Filtering: Filter captured frames by minimum or maximum duration.
- Percentile Analysis: Calculate percentiles (P50, P75, P95, P99) for RUM reporting and performance monitoring.
- Data Export: Download the data in JSON or CSV format for later analysis or sharing.

How to Use

1. Copy the entire snippet code.

On This Page

[Key Features](#)

[How to Use](#)

[Usage Examples](#)

[Snippet](#)

[Snippet \(minified version\)](#)

Percentiles Analysis

[Why Use Percentiles?](#)

[Usage Examples](#)

[Understanding the Output](#)

[Integration with RUM](#)

andydavies / perf-timeline-to-devtools-profile

Q Type / to search

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

perf-timeline-to-devtools-profile

Public

Pin

Unwatch 2

Fork 0

Star 17

main

1 Branch

0 Tags

Go to file

Add file

<> Code

About

andydavies

Add optional chaining for paintTime and presentationTime

407c9d6 · last week

11 Commits

images	Update readme	11 months ago
LICENSE	Initial commit	11 months ago
README.md	Update readme	3 months ago
content-script.js	Add optional chaining for paintTime and presentationTime	last week
manifest.json	Fix INP measurement	3 months ago

README

MIT license

perf-timeline-to-devtools-profile

Chrome Extension that creates a custom track in the DevTools Performance Panel populated with entries from the Performance Timeline

Originally created to help visualise the LoA

Registers Performance Observers for the f

data for DevTools

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

JavaScript 100.0%

https://github.com/andydavies/perf-timeline-to-devtools-profile

I  **LoAF**

LoAF provides insights into the runtime costs of the code we ship

And allow us to identify our problem scripts

Within the context of our visitors environment

You're going to need RUM



Photo by [Ash Edmonds](#) on [Unsplash](#)

Feel like I've just scratched the surface...

... and there's more for LoAF to reveal



Finally... if you're flying out of Schiphol...

Thanks!

@andydavies.me

andy.davies@speedcurve.com

