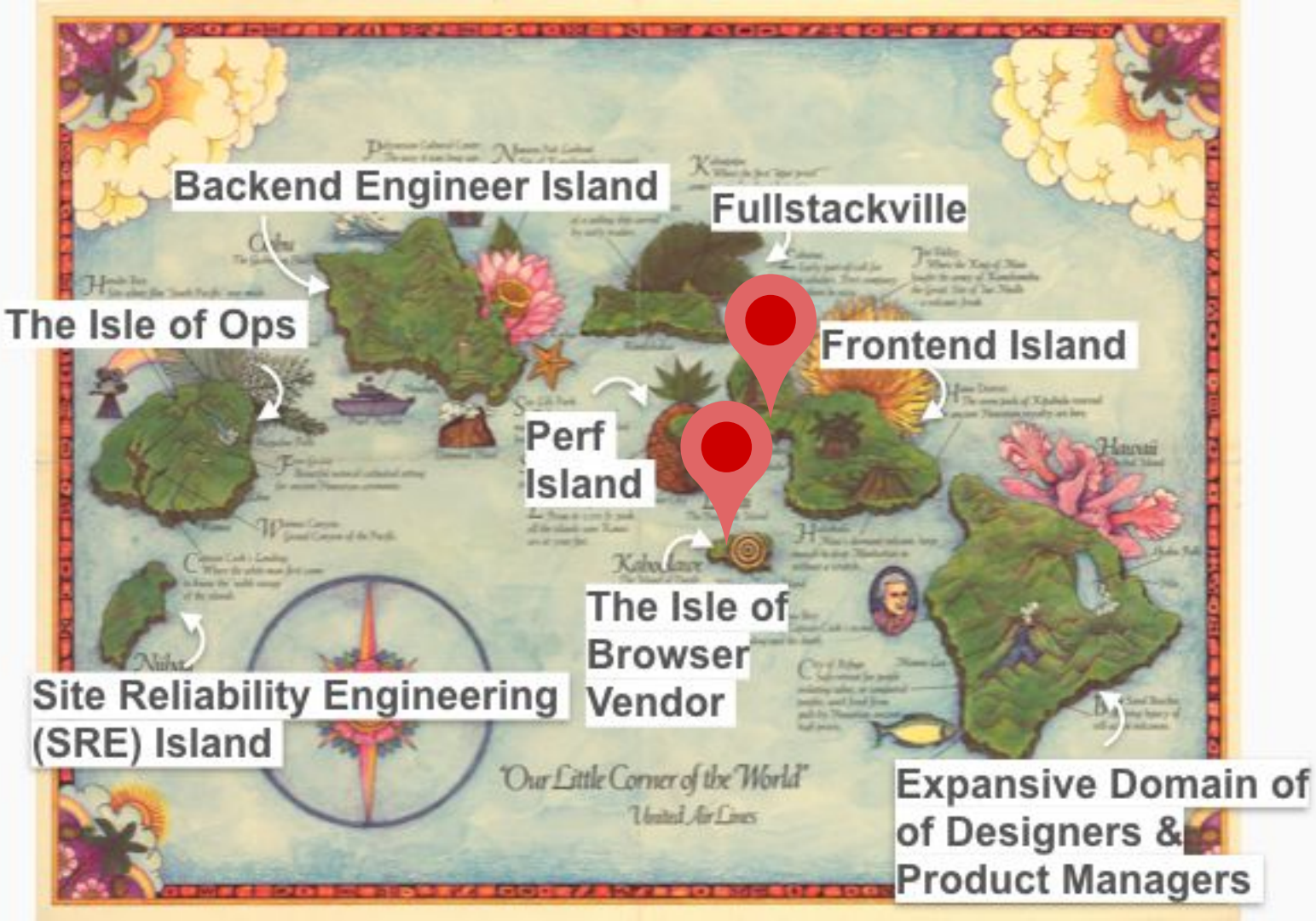
Metrics and Monitoring in Chrome

Lessons learned over the years

http://bit.ly/monitoring-and-metrics-in-chrome


Backend Engineer Island
Fullstackville
The Isle of Ops
Frontend Island
Perf Island
The Isle of Browser Vendor
Site Reliability Engineering (SRE) Island
Expansive Domain of of Designers & Product Managers
"Our Little Corner of the World"
United Air Lines

Agenda for our trip

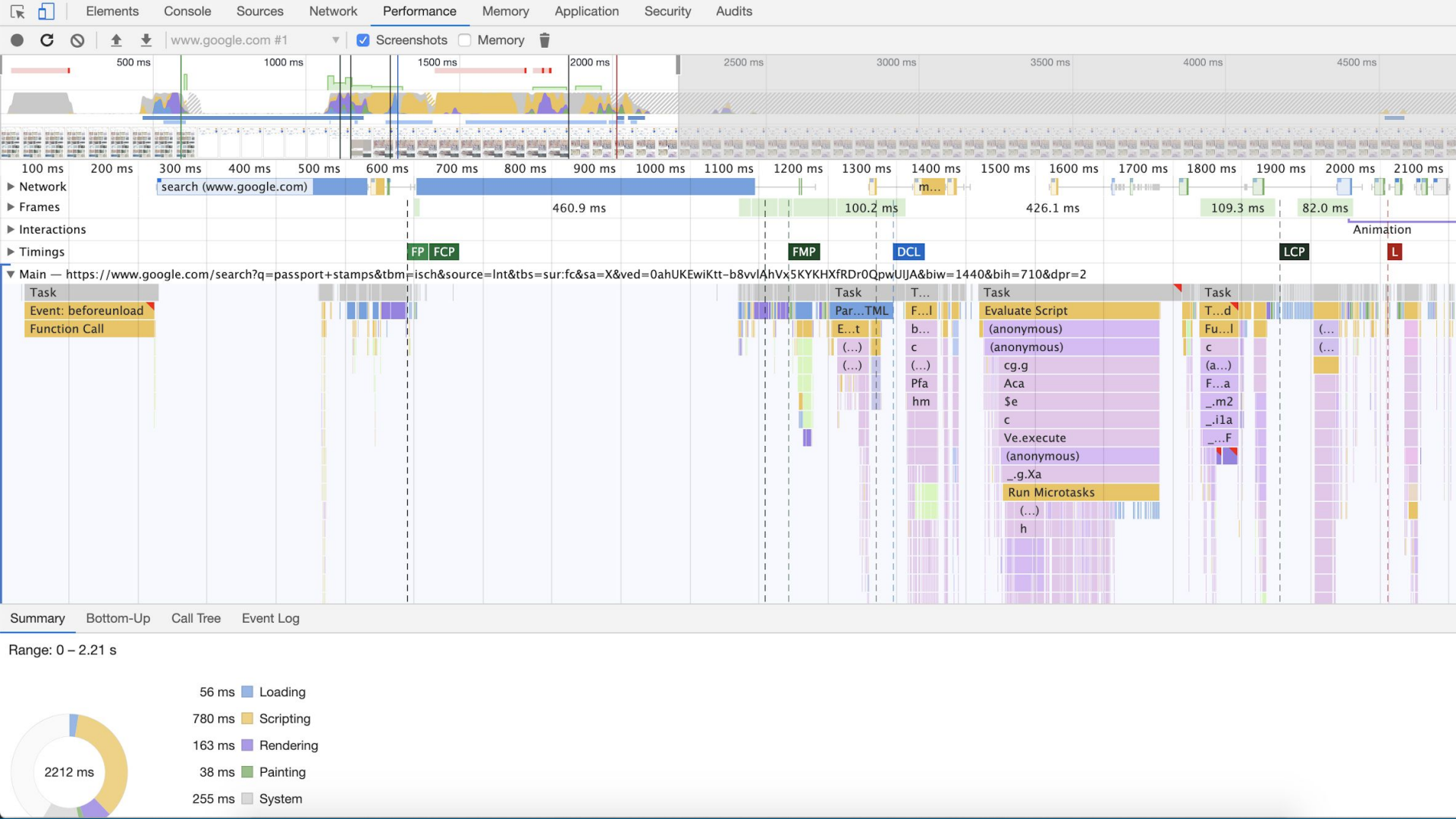
Metrics

- Overview
- Properties of good metrics
- Use cases for metrics
- Example

Monitoring

- Lab
- A/B Testing
- RUM

Metrics


Elements
Console
Sources
Network
Performance
Memory
Application
Security
Audits
www.google.com #1
Screenshots
Memory
500 ms
1000 ms
1500 ms
2000 ms
2500 ms
3000 ms
3500 ms
4000 ms
4500 ms
100 ms
200 ms
300 ms
400 ms
500 ms
600 ms
700 ms
800 ms
900 ms
1000 ms
1100 ms
1200 ms
1300 ms
1400 ms
1500 ms
1600 ms
1700 ms
1800 ms
1900 ms
2000 ms
2100 ms
Network
search (www.google.com)
m...
Frames
460.9 ms
100.2 ms
426.1 ms
109.3 ms
82.0 ms
Interactions
Animation
Timings
FP
FCP
FMP
DCL
LCP
L
Main — https://www.google.com/search?q=passport+stamps&tbm=isch&source=lnt&tbs=sur:fc&sa=X&ved=0ahUKEwiKtt-b8vvlAhVx5KYKHXfRDr0QpwUIJA&biw=1440&bih=710&dpr=2
Task
Event: beforeunload
Function Call
Task
Par...TML
E...t
(...)
(...)
T...
F...l
b...
c
(...)
Pfa
hm
Task
Evaluate Script
(anonymous)
(anonymous)
cg.g
Aca
$e
c
Ve.execute
(anonymous)
_.g.Xa
Run Microtasks
(...)
h
Task
T...d
Fu...l
c
(a...)
F...a
_.m2
_.i1a
_...F
(...
(...
Summary
Bottom-Up
Call Tree
Event Log
Range: 0 – 2.21 s
2212 ms
56 ms Loading
780 ms Scripting
163 ms Rendering
38 ms Painting
255 ms System

We need good **top level** metrics.

Properties of a Good Top Level Metric

Representative

Accurate

Interpretable

Simple

Stable

Elastic

Realtime

Orthogonal

Use Cases of Top Level Metrics

Lab	Web Performance API	Chrome RUM
		

Lab

Fewer data points than RUM

Lab metrics need to be **Stable** and **Elastic**.

This can put them at odds with being **Simple** and **Interpretable**.

They don’t require **Realtime**.

Care should be put into handling **user input**.

Web Perf API

Broad range of use cases

It is critical that metrics exposed to Web Perf APIs are **Representative** and **Accurate**.

They must be **Realtime**.

It's more important to be **Simple** and **Interpretable** than **Stable** or **Elastic**.

Clear definitions that can be **implemented across engines** are important.

Chrome RUM

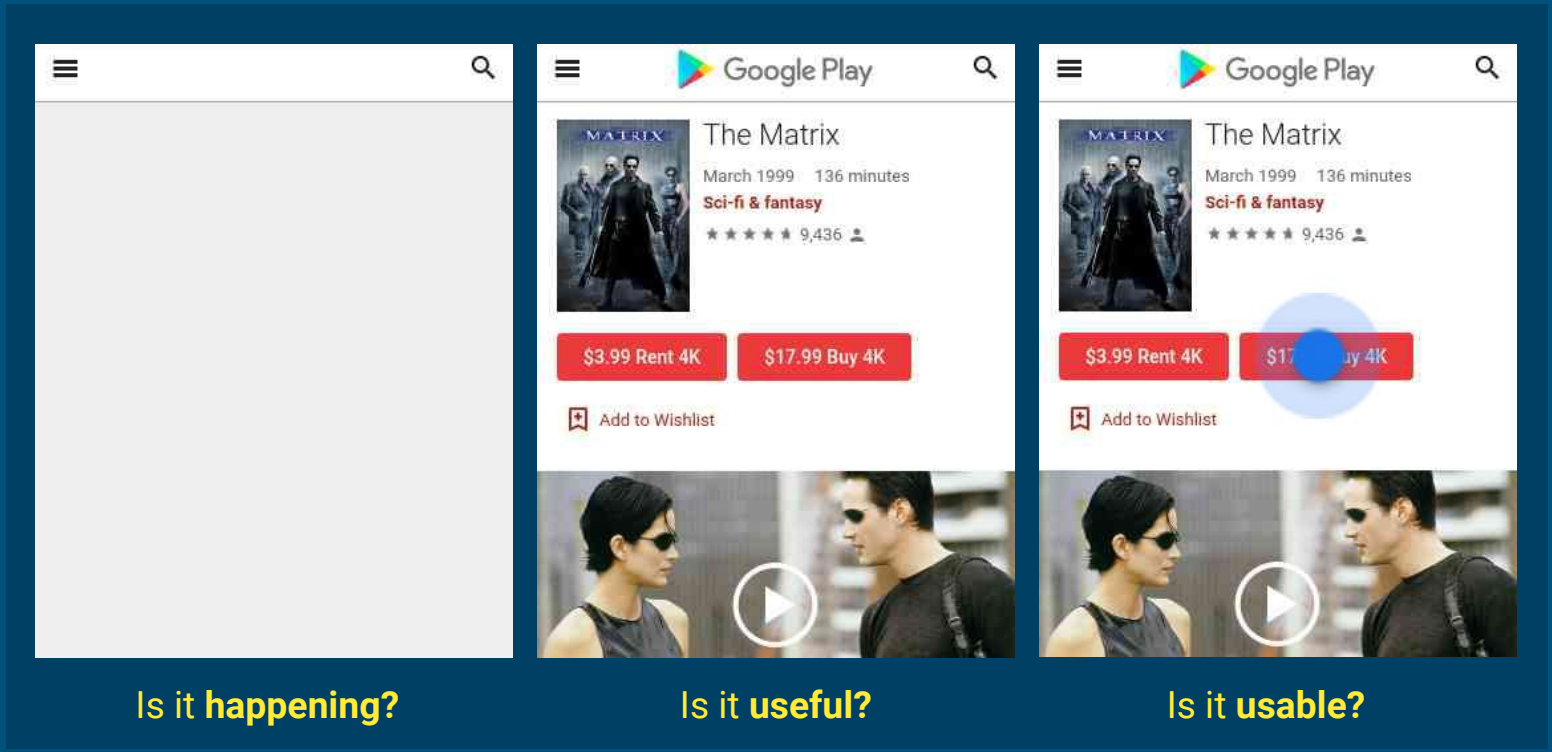
Understanding Chrome performance in the wild.

We care a lot that our internal metrics are **Representative** and **Accurate**, but we can iterate on them frequently.

We get a very large volume of data, so it's less important that they are **Stable** or **Elastic**.

Example: Largest Contentful Paint

Key Moments in Page Load

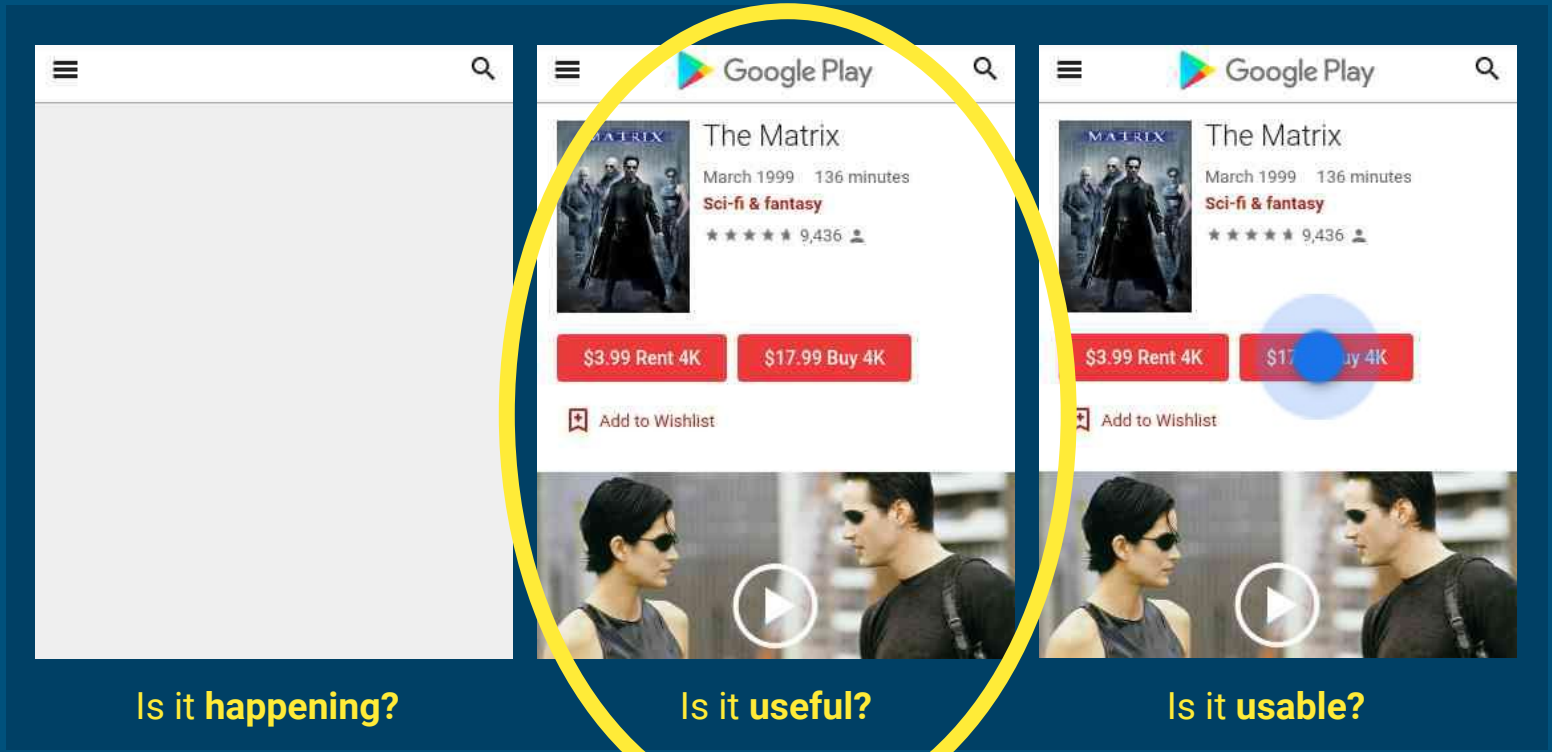


Is it **happening?**

Is it **useful?**

Is it **usable?**

Goal: When is Main Content Loaded?



Prior work: Speed Index

Average time at which visible parts of the page are displayed

Speed index is **Representative, Accurate, Interpretable, Stable**, and **Elastic**.

But it's not **Realtime**, which is a requirement for web perf APIs and RUM.

And it's not **Simple**.

Prior work: First Meaningful Paint (deprecated)

Heuristic: first paint after biggest layout change

First Meaningful Paint is **Representative, Interpretable,** and **Realtime**.

But it produces strange, outlier results in 20% of cases, making it not **Accurate**. It's not **Simple,** or **Stable,** or **Elastic.**

Main Content Painted Metric Priorities

1. Representative
2. Accurate
3. Realtime
4. Interpretable
5. Simple

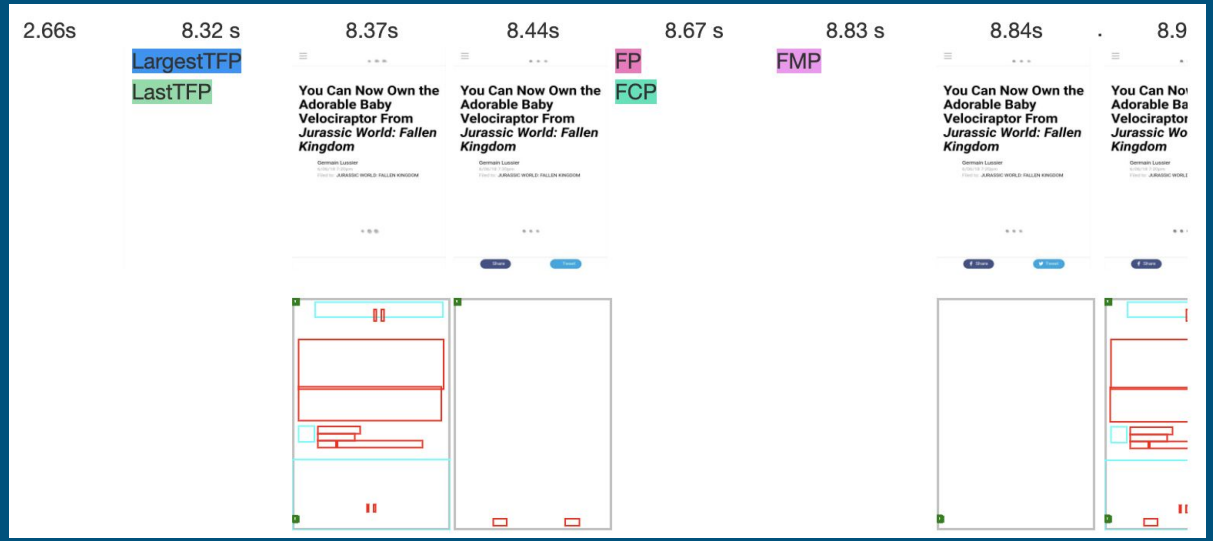
We can get paint events in **Realtime**. Can we make a metric that is **Simple** and **Accurate**?

Brainstorm Simple Metrics

- Largest image paint in viewport
- Largest text paint in viewport
- Largest image **or** text paint in viewport
- Last image paint in viewport
- Last text paint in viewport
- Last image **or** text paint in viewport...

And Test Them for Accuracy

- Generate filmstrips for 1000 sites.
- Look at the filmstrips and the layouts, and what each metric reports for them.
- Pick the best.



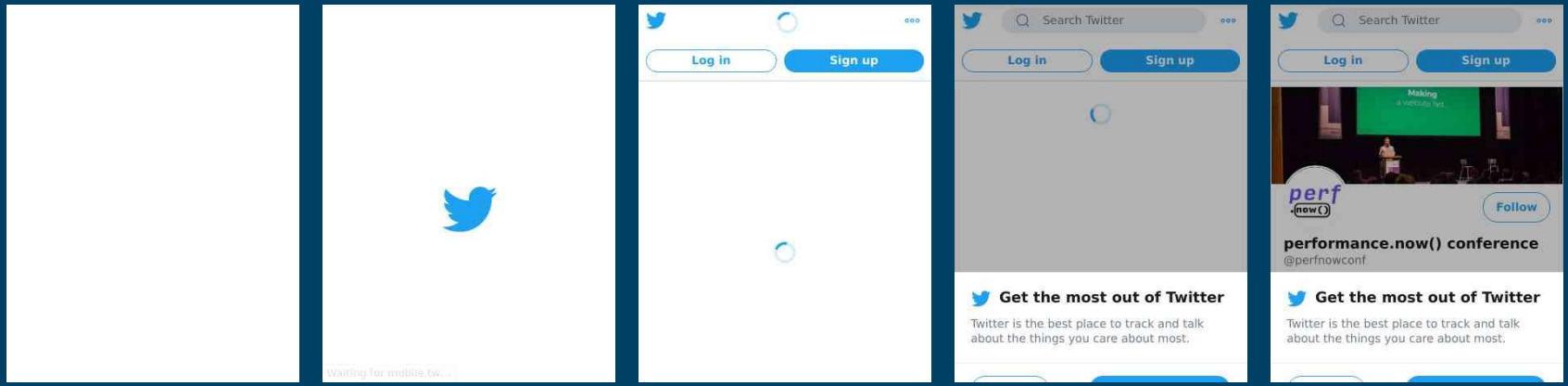
And the Winner Is...

Largest image or text paint in viewport: **Largest Contentful Paint**.

Unfortunately, it couldn’t be **quite** that simple…

What About Splash Screens?

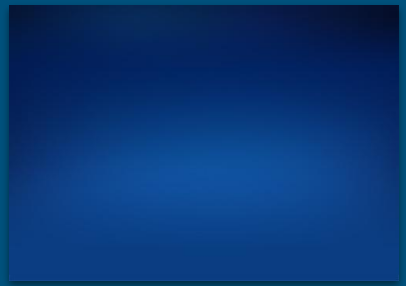
Elements that are removed from the DOM are invalidated as candidates for Largest Contentful Paint.



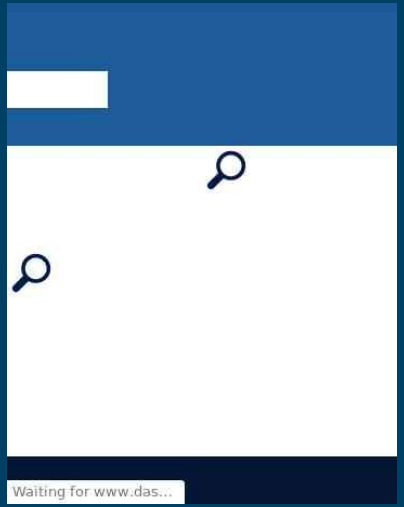
First contentful paint (removed)

Largest contentful paint

What About Background Images?



Page background image



Background image painted

Real largest contentful paint

What is a “Text Paint”?

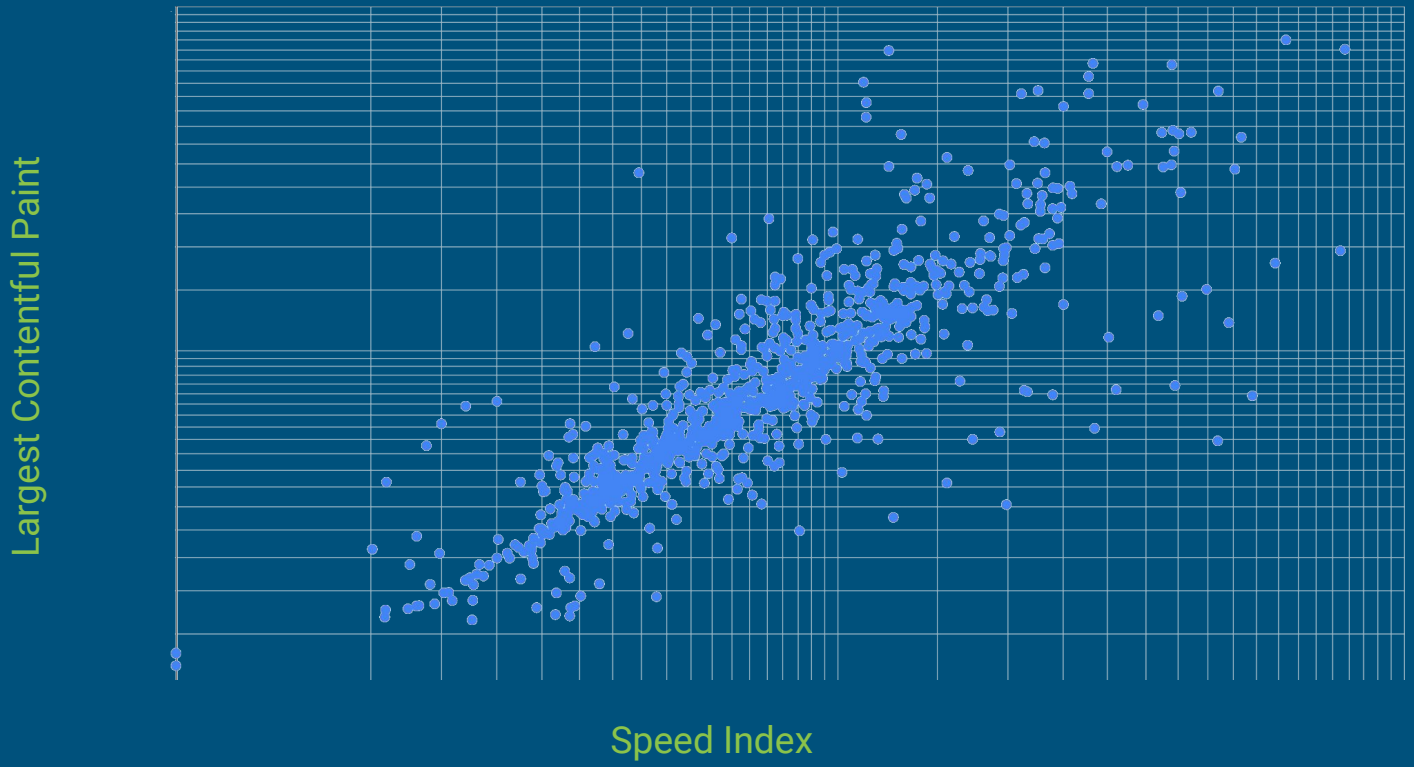
Aggregate to block-level elements containing text nodes or other inline-level text elements children.

What About User Actions During Load?

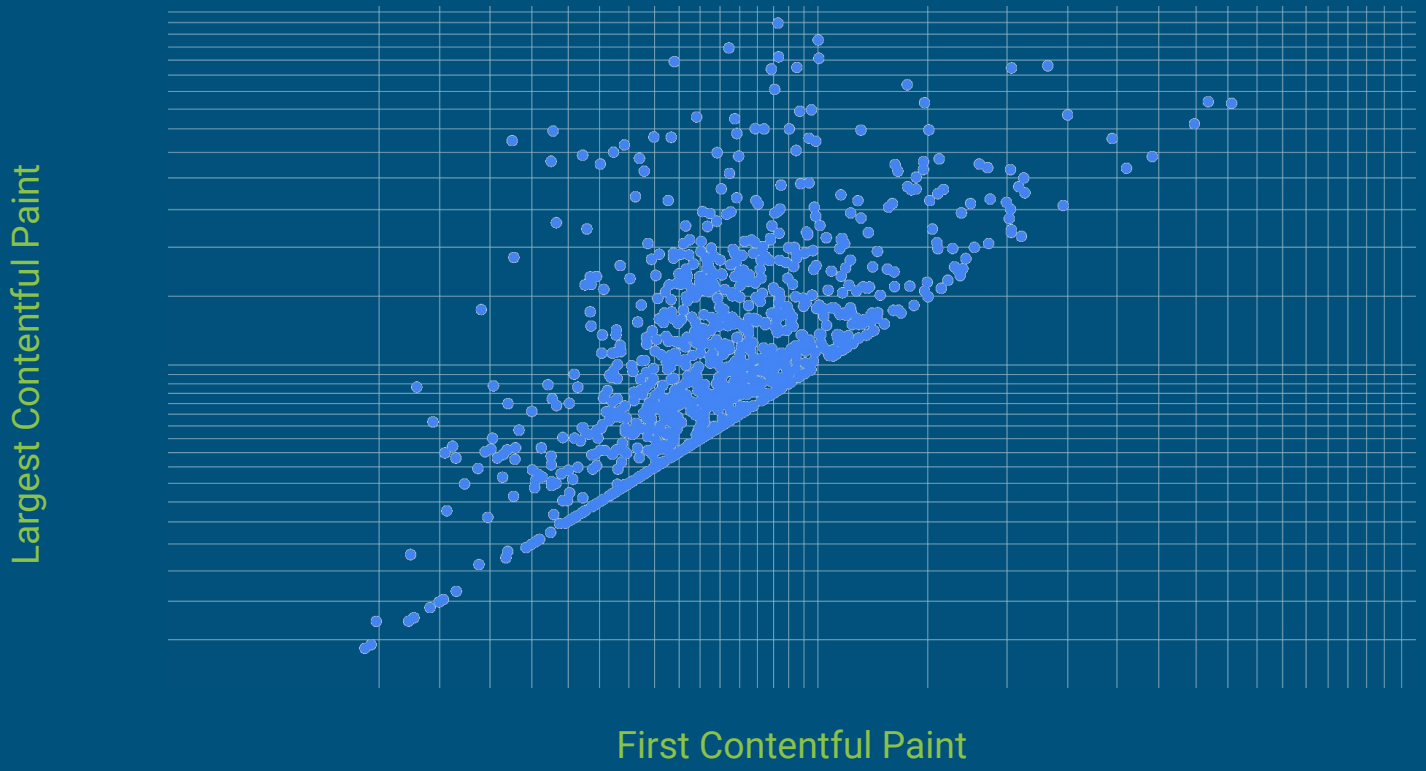
Largest Contentful Paint cannot be measured if there is user input.

Example: infinite scroll

Validating using HttpArchive



Validating using HttpArchive



We’d like to work with you!

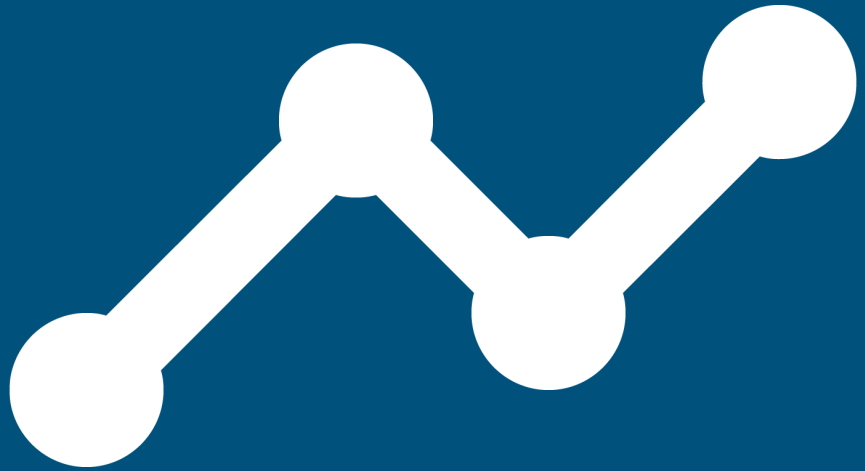
speed-metrics-dev@chromium.org

Monitoring

Monitoring stages

Lab → A/B Testing → RUM

Lab Testing



Lab Testing Pros and Cons

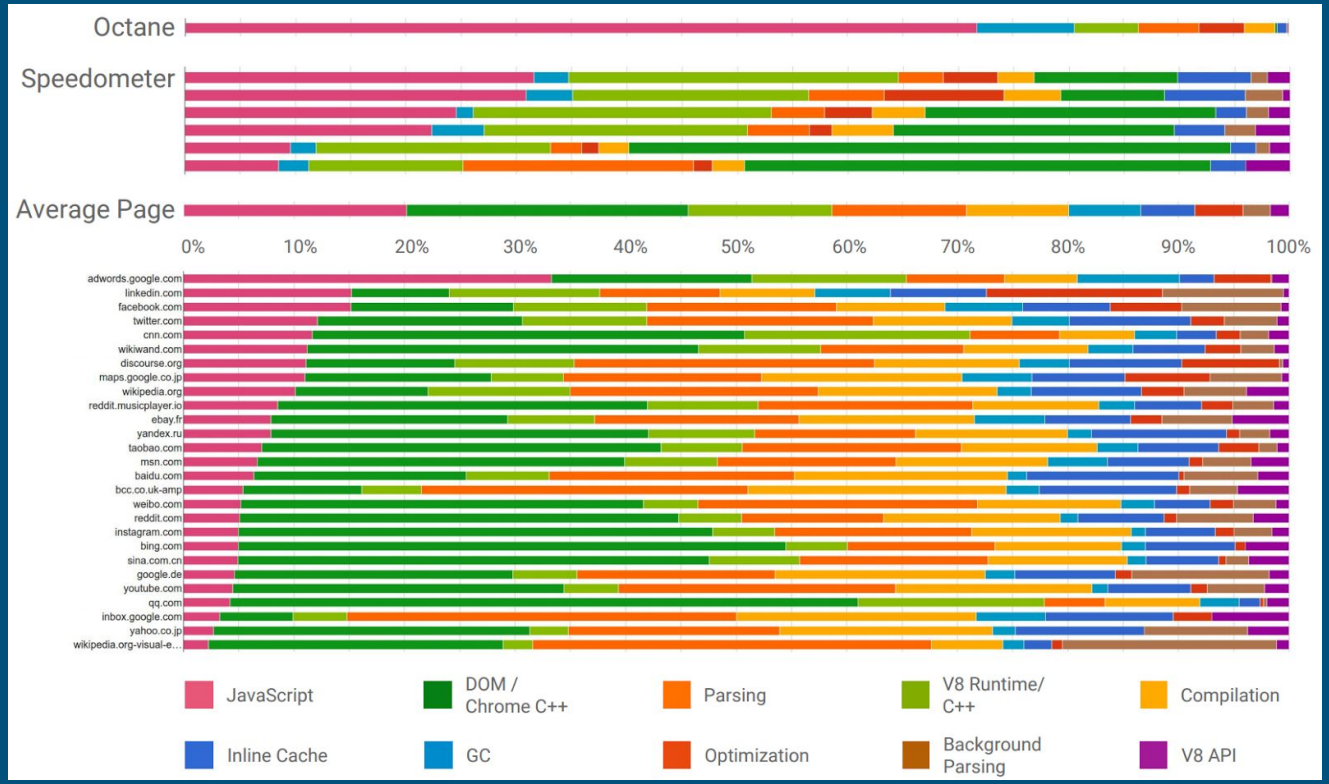
Good For	Limitations
Fast iteration	Can’t model every user
Repeatable results	Can’t predict magnitude of changes
Debugging	
Trying things you can’t launch	

Competing Goals

Reproducibility

Realism

Benchmark Realism: Real-world Stories



Better Reproducibility: Test Environment

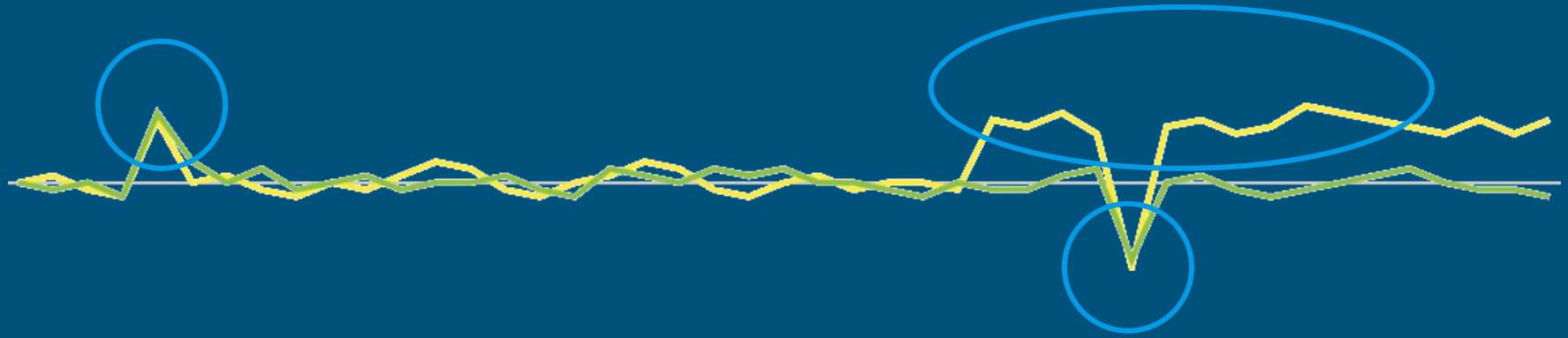
- Consider **real hardware** instead of VMs
- Use **exact same** model + configuration, or even same device
- On mobile, ensure devices are **cool** and **charged**
- Turn off all **background tasks**
- Reduce randomization
  - Simulate network
  - Mock Math.random(), Date.now(), etc
  - Freeze 3rd party scripts

Better Reproducibility: Diagnostic Metrics

- CPU time
- Breakdowns
- Sizes and counts (JS size, number of requests)
- Background noise

Better Reproducibility: Reference Runs

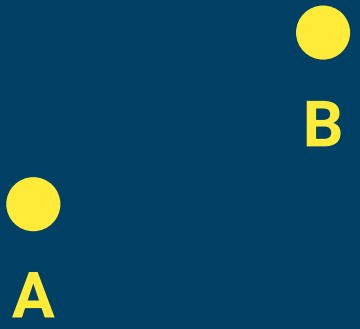
Consider a **reference** run with a pinned version, to find sources of noise outside the test case.



Better Reproducibility: Change Detection

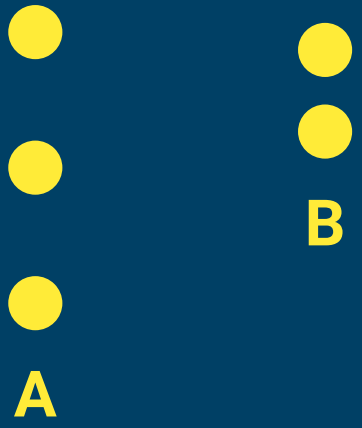
Comparing Two Versions

How do you know if it’s just noise?



Add more data points!

But how do you compare them?

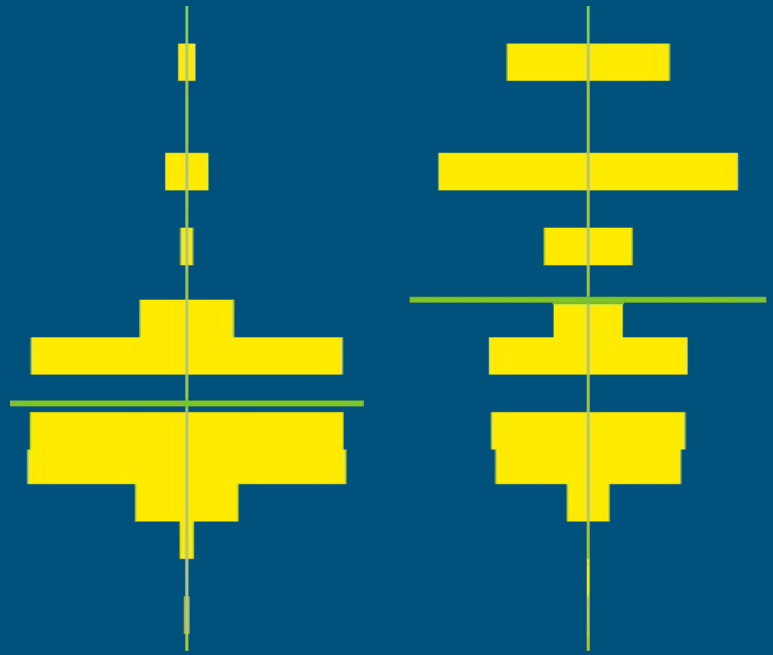


Each Group of Runs is a Set of Samples

Use a **hypothesis test** to see if the sets are from different.

Note that performance test results are usually **not normally distributed**.

Recommend looking into Mann-Whitney U, Kolmogorov-Smirnov, and Anderson-Darling.



A/B Testing

A/B Testing Pros and Cons

Good For	Limitations
Predicting real-world effect of performance optimizations	Can’t A/B test every change
Understanding potential regressions from new features	Many concurrent A/B tests can be hard to manage

Should be “Controlled Experimentation”

One control group

Any number of variations

User opt-in is **not** a controlled experiment

Best practices for experiment groups

- Use equal-sized control and experiment groups
- Compare groups on same version **before** experiment
- Options for getting more data
  - Run over a longer time period
  - Increase group size

Real User Monitoring

RUM Pros and Cons

Good For	Limitations
Ground truth about user experience	Hard to reproduce and debug By the time a regression is detected, real users are already experiencing it

Why is understanding RUM data so hard?

Reasons your RUM metric is doing something you don’t understand

- Diversity of user base
- Mix shifts
- Things out of your control--patch Tuesday type updates
- Changes in metric definition

What can you do about it?

What to monitor

- Use percentiles
- Monitor **median** and a **high** percentile

Checking for Mix Shift

- First check for volume change
- Try splitting
  - By county
  - By platform
  - By device type

Metric Breakdowns

Think in terms of **traces**, not **events.**

Make RUM more like A/B Testing

Launch new features via A/B test

Canary launches

Use a holdback

Takeaways

- Metrics are hard, but you can help! speed-metrics-dev@chromium.org
- Focus on user experience metrics
- Use lab for quick iteration, A/B testing for deeper understanding

Thanks!

@anniesullie

http://bit.ly/monitoring-and-metrics-in-chrome