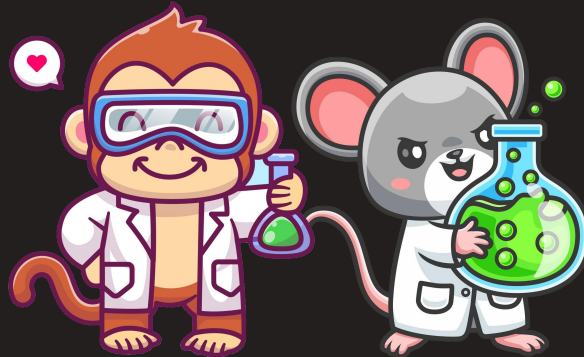


Synthetic monitoring

●● What it catches and what it misses



May 5, 2026

Front-end observability veterans Cliff Crocker and Tammy Everts deep dive into how synthetic monitoring catches issues before they hit your users – and where real user monitoring needs to take over.

Housekeeping items

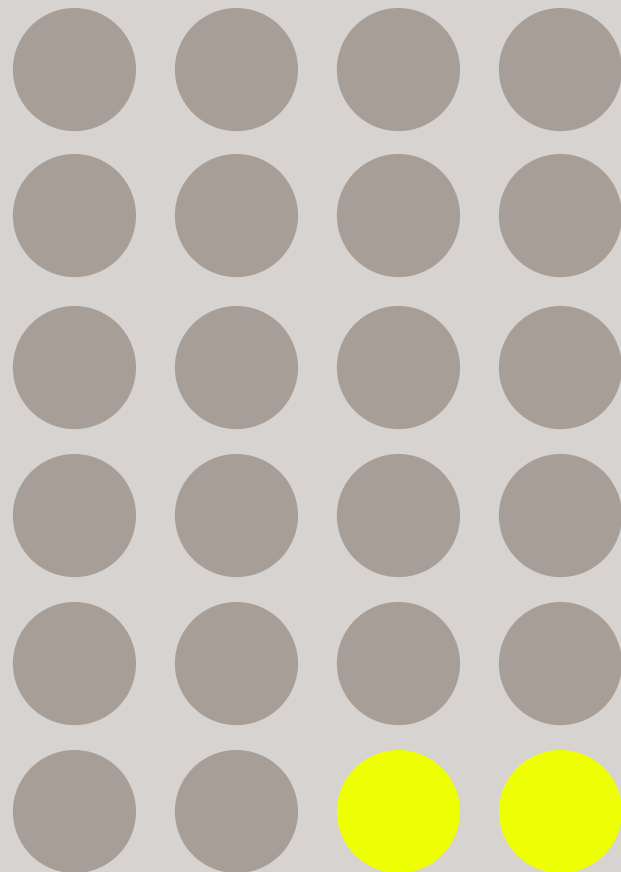
- The webinar is being recorded and will be available later for on-demand viewing.
- Please ask your questions in the chat box provided. We will get to the questions at the end of this webinar.



Cliff Crocker
VP of Product



Tammy Everts
Sr. Director of Community



HYATT



THE
NEW YORKER

TED

trivago

B B C

Forbes

BUSINESS
INSIDER

The
Guardian



F A I R E

User-focused observability for web starts here

RUM + synthetic monitoring,
fully connected in Embrace



Today we're talking about...

- ★ Synthetic monitoring
- ★ Core Web Vitals
- ★ Use cases & best practices
- ★ What's around the corner?
- ★ Q&A

1

Synthetic monitoring

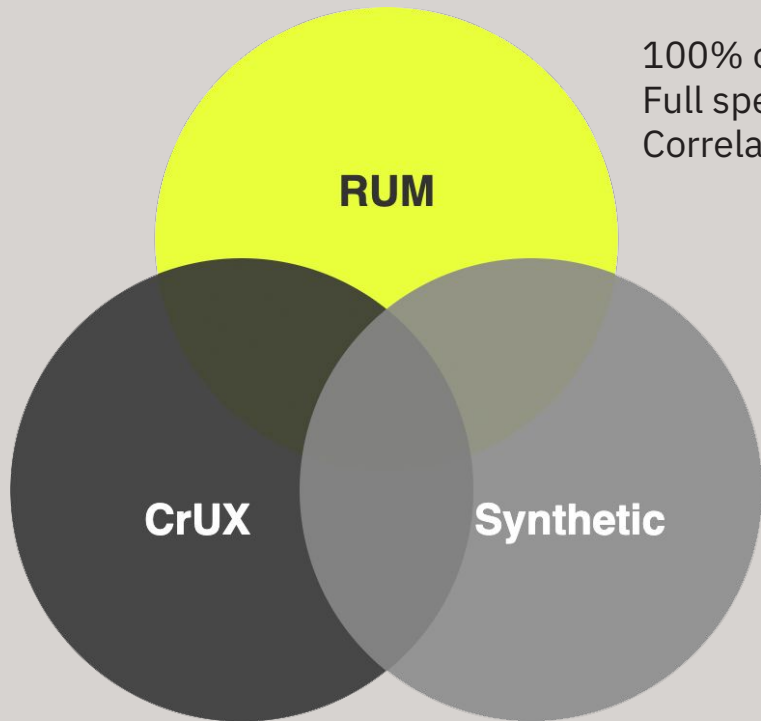


Synthetic monitoring

AKA lab data

Helps you understand **your pages** and what you need to fix to improve UX

Proactive – see problems **before** your users do



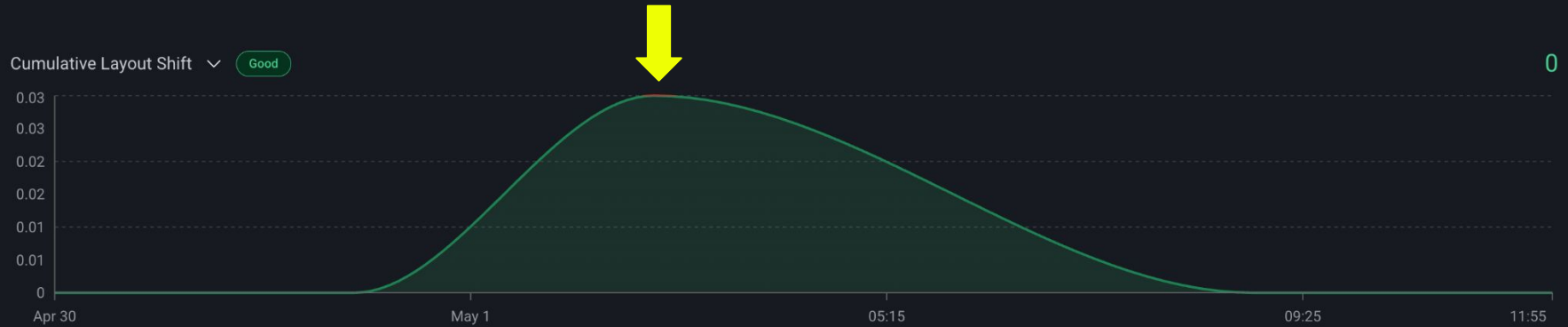
100% of your real users
Full spectrum of user experiences
Correlate to business and UX metrics

Chrome only
Limited dataset and metrics
Influences Google search

Test any URL
Competitive benchmarking
Detailed diagnostics

●● Establish a clean baseline

Spot when CLS threshold is violated



•• Measure **any** URLs

URLs

Add as many URLs as you like. Use labels to group URLs together and compare pages across different sites.

<https://www.ikea.com/us/en/>

<https://www.lowes.com/>

<https://www.apple.com/>

<https://www.nike.com>

<https://www.amazon.com>

<https://www.walmart.com>

+ Add URL

•• Script different scenarios

`https://www.ikea.com/us/en/`

</>

Script Templates ▲

Scripting Docs

Repeat View


Block Third Party



PWA: Repeat view while offline

htt

Don't add 'PTST' to user agent

•• Test pages in staging/dev environments

https://www.ikea.com/us/en/ IKEA 

 Auth Username Auth Password 



Add a **username** and **password** for basic HTTP authorization

•• Trigger a test on deploy

On-Demand Testing ×

What would you like to test?

Site Custom URL

Site

Select site to be tested ▼

URLs

Select URLs to be tested ▼ [Edit URLs](#)

[Start Testing](#) [Cancel](#)

Plugins or API for testing in CI pipeline

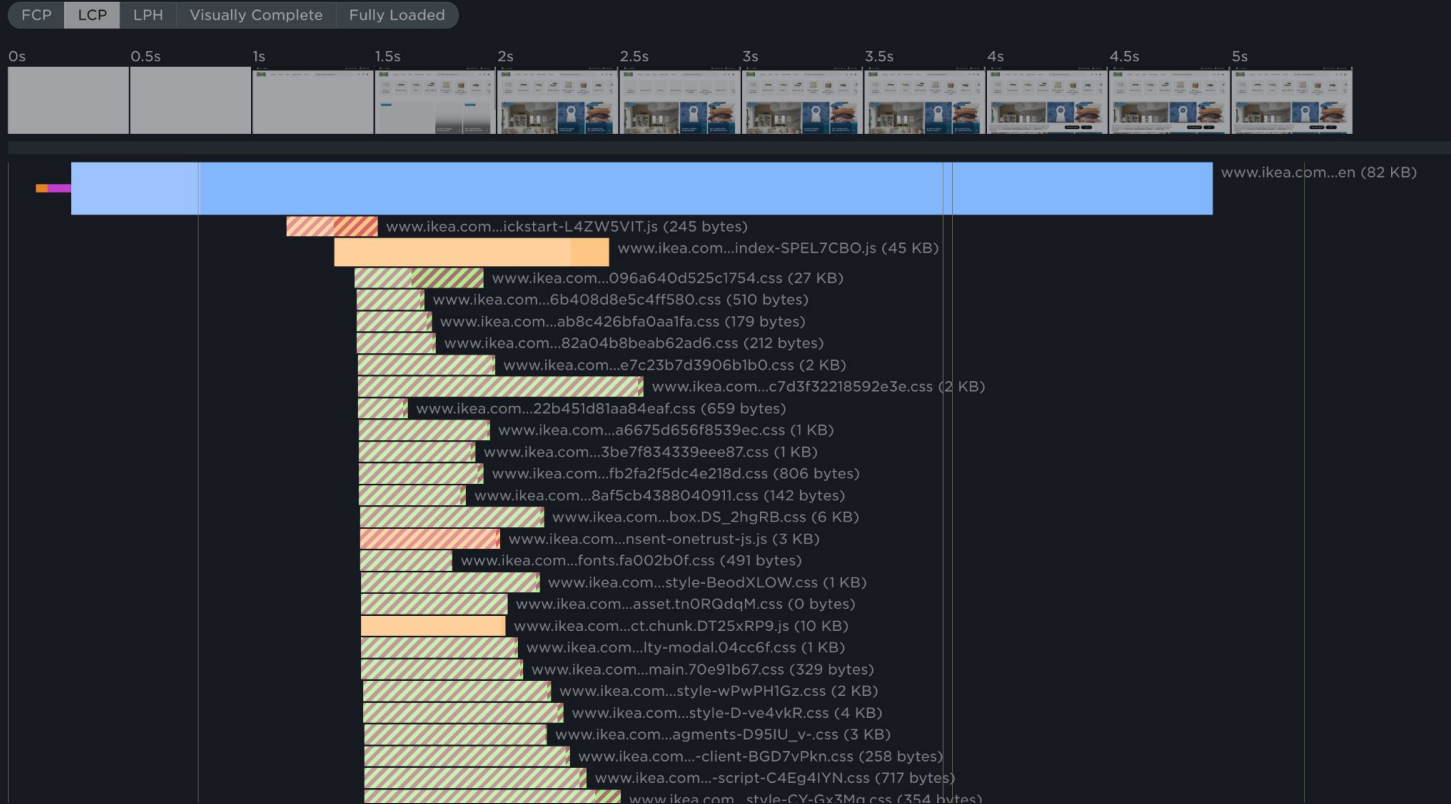
OR

GitHub integration – and even break the build!

OR

Just do a manual test after deploy

•• Waterfall charts



•• Detailed audits and recommendations

Performance 45 Accessibility 100 Best Practices 48 SEO 92

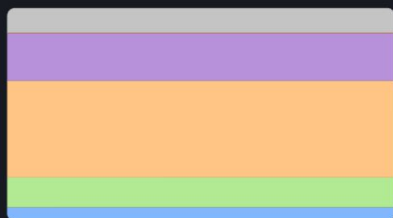
Lighthouse Performance Audit	
>	Page prevented back/forward cache restoration (1 failure reason)
>	Avoid large layout shifts (5 layout shifts found)
>	Reduce the impact of third-party code (Third-party code blocked the main thread for 5,500 ms) TBT
>	Minimize main-thread work (35.6 s)
>	Reduce JavaScript execution time (18.8 s) TBT
>	Avoid an excessive DOM size (3,442 elements) TBT
>	Reduce unused JavaScript (Est savings of 1,150 KiB) FCP LCP
>	Reduce unused CSS (Est savings of 94 KiB) FCP LCP
>	Image elements do not have explicit `width` and `height` CLS
>	Avoid `document.write()`
>	Serve static assets with an efficient cache policy (127 resources found)
>	Avoid enormous network payloads (Total size was 8,498 KiB)
>	Avoid serving legacy JavaScript to modern browsers (Est savings of 42 KiB) FCP LCP
>	Enable text compression (Est savings of 2 KiB) FCP LCP
>	Defer offscreen images (Est savings of 35 KiB) FCP LCP
>	Properly size images (Est savings of 424 KiB) FCP LCP
20 passed audits ▾	

•• Detailed analysis of assets

Requests

540

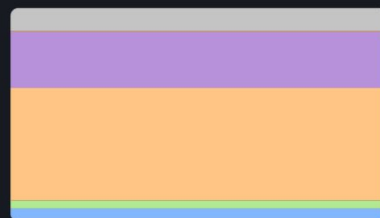
500
300
100



Size

5.77 MB

5 MB
3 MB
1 MB



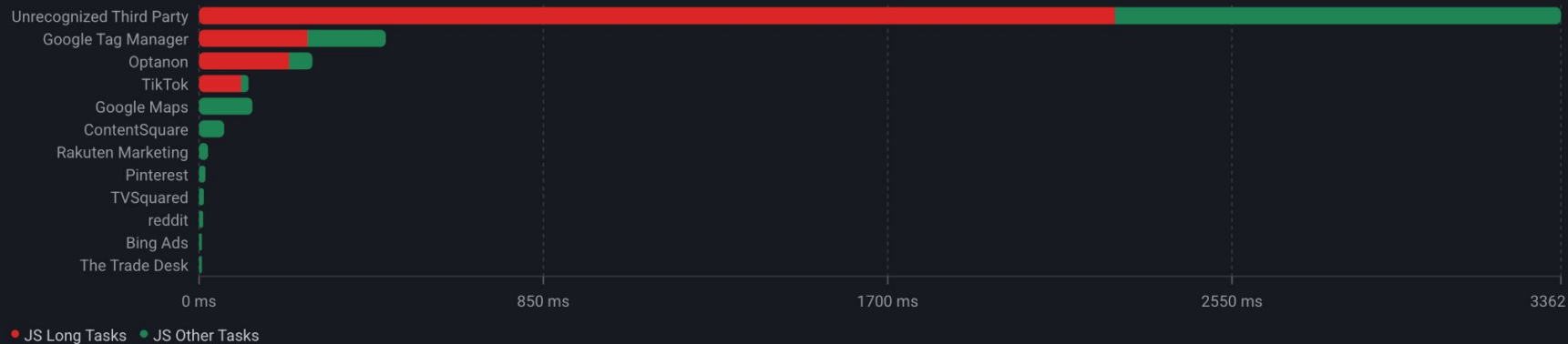
Content

Requests

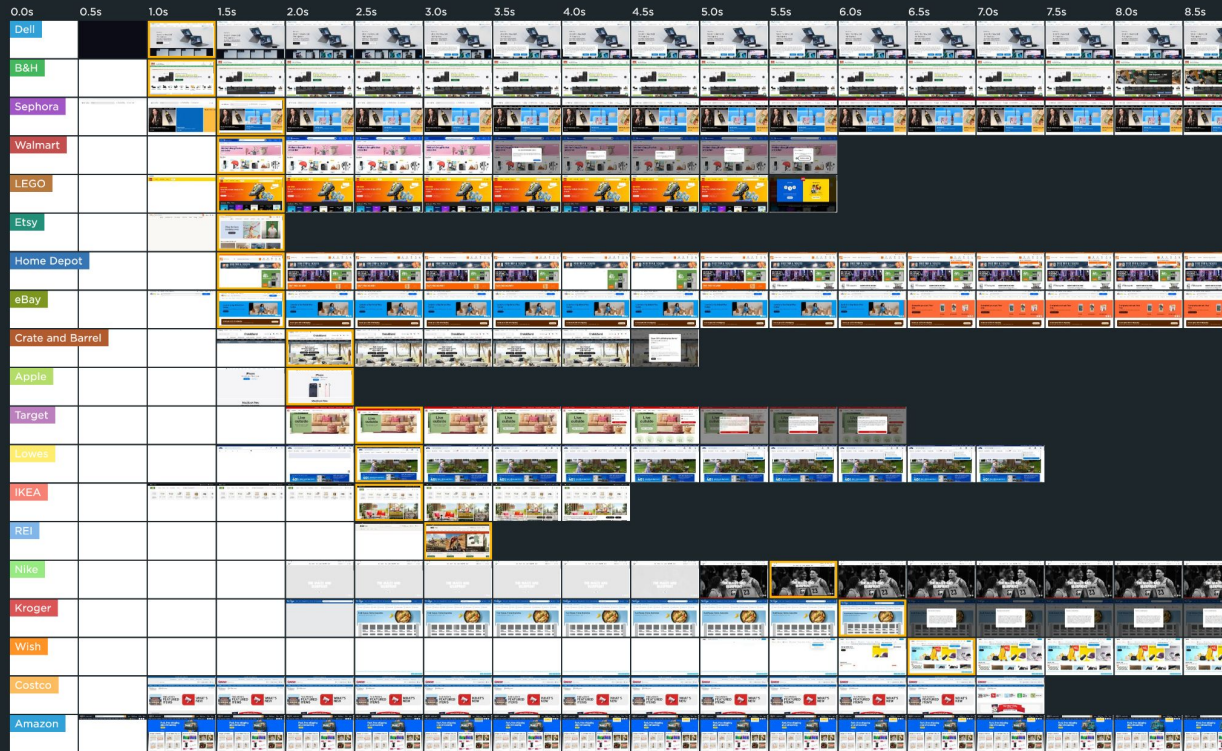
Size

• HTML	33	324.17 kB
• CSS	76	217.2 kB
• JS	246	3.06 MB
• Image	120	1.52 MB
• Font	2	22.38 kB
• Video	0	0 B
• Other	63	634.21 kB

•• Identify third parties & long tasks



RETAIL SITES IN USA ON A DESKTOP FAST CONNECTION



speedcurve.com/benchmarks/

2

Core Web Vitals

(Loading)

LCP

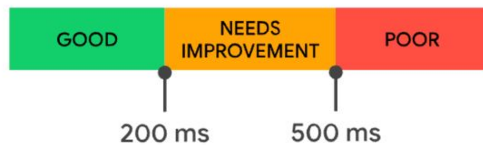
Largest Contentful Paint



(Interactivity)

INP

Interaction to Next Paint



(Visual Stability)

CLS

Cumulative Layout Shift



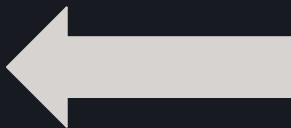
Is it loading?



Is it stable?



Is it interactive?



synthetic proxy for INP

Is it **loading**?

LCP Good



Is the **server** responding?

TTFB Good



Is it **stable**?

CLS Good



Is **anything** happening on the page?

FCP Good



Is it **interactive**?

TBT Good



Page Label ↑↓	TTFB ⌚ ↑↓	FCP ⌚ ↑↓	LCP ⌚ ↑↓	TBT ⌚ ↑↓	CLS ⌚ ↑↓
Amazon	0.8 s	2 s	17.2 s	806 ms	0.13
Apple	0.7 s	3.3 s	3.3 s	0 ms	0
IKEA	0.8 s	1.7 s	2.6 s	1,688 ms	0.2
Lowes	1.2 s	1.6 s	8.7 s	10,150 ms	0
Nike	1 s	3.1 s	7.6 s	3,733 ms	0.01
Walmart	0.7 s	1.9 s	1.9 s	9,009 ms	0.08



(This is where INP lives if you have RUM)

3

Demo

AKA the good stuff!

4

Best practices

•• Four flavours of synthetic testing (and why you need them all)

1. **Scheduled tests** – Consistent baseline so you can detect performance changes and trigger alerts when something slows down
2. **Deployment tests** – Integrate testing into your CI/CD pipeline. Automatically test in staging and immediately after a deploy. Catch non-performant code before it hits your users.
3. **Ad-hoc testing** – Test any time just by hitting a button.
4. **Competitive benchmarking** – Most leading sites benchmark themselves against at least two competitors.

•• Best practices for synthetic testing

❑ **Test key page types**

Key pages will vary for different sites. A retailer will have Product, Category, Search, Checkout, and Landing pages. A news publisher will have Home, Section, and Article pages.

❑ **Test from multiple relevant geographic regions**

Align test locations to where your actual visitors are (which your RUM data will tell you) — not just default regions.

❑ **Use a minimum of 3 runs per test, taking the median result**

Reduces variability and catches outliers.

❑ **Test at least every 6 hours for baseline monitoring**

Increase to every 3 hours for high-revenue or critical pages. Test more when visitors are active.

❑ **Cover the right browser profiles**

Minimum: Desktop Slow + Mobile Fast + Mobile Medium

Also consider Mobile Slow (bad Android) and Desktop Fast

5

Roadmap

What's around the corner?

Future Capabilities for Synthetics

Larger synthetic initiatives actively being scoped

01

CI/CD Integration

Run Embrace Synthetics as a first-class step in CI/CD pipelines, gating deployments on performance budgets and regression detection.

Completes the shift-left observability story for front-end developers and SREs.

Catch performance regressions before they reach production, not after users are affected.

02

Competitive Benchmarks++

Synthetic performance benchmarks measured against competitor URLs, enriched with CrUX field data and predefined industry datasets.

Connects raw performance data to competitive context so teams can see where they stand in their market.

Provides a clear, data-backed case for performance investment that has the most competitive impact.

03

Functional Testing

Automated functional test execution tracked and correlated alongside real-user performance data.

Bridges the gap between synthetic test pass/fail and real-world quality signals.

This gives teams confidence that working code also performs well for real users in production.

6

Real user monitoring

Picks up where synthetic leaves off...

Synthetic

Test specific URLs, not entire sites

Mimics defined network & browser conditions

Limited test locations

Scheduled tests can miss sporadic outages & issues

No correlation to user behaviour or business KPIs

Real user monitoring

Large sample size → full fidelity

Real network and browser conditions

Full geographic spread

Always on

Conversion & KPI correlation to performance metrics



Thank you

