

Optimizing Interaction to Next Paint

Jeremy Wagner – jlwagner.net – @malchata
BiznagaFest 2002 – Málaga, España

What is responsiveness?

gShoe product Q&A:

What is gShoe?

What technology does gShoe use?

How much does gShoe cost? 

Poor responsiveness

gShoe product Q&A:

What is gShoe?

What technology does gShoe use?

How much does gShoe cost? 

Good responsiveness

What causes poor responsiveness?

Main thread

Task

Evaluate script

Compile script

Compile code



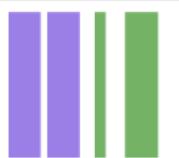
Task

Event: keyup

Function call

fetchWrapper

sendFetch





Task

Evaluate script

Compile script

Compile code

77.6 ms (total)

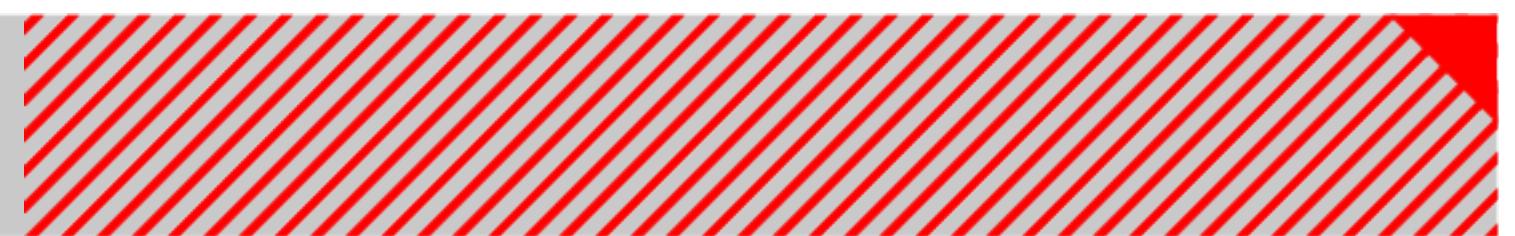
27.6 ms (blocking)

Task

Evaluate script

Compile script

Compile code



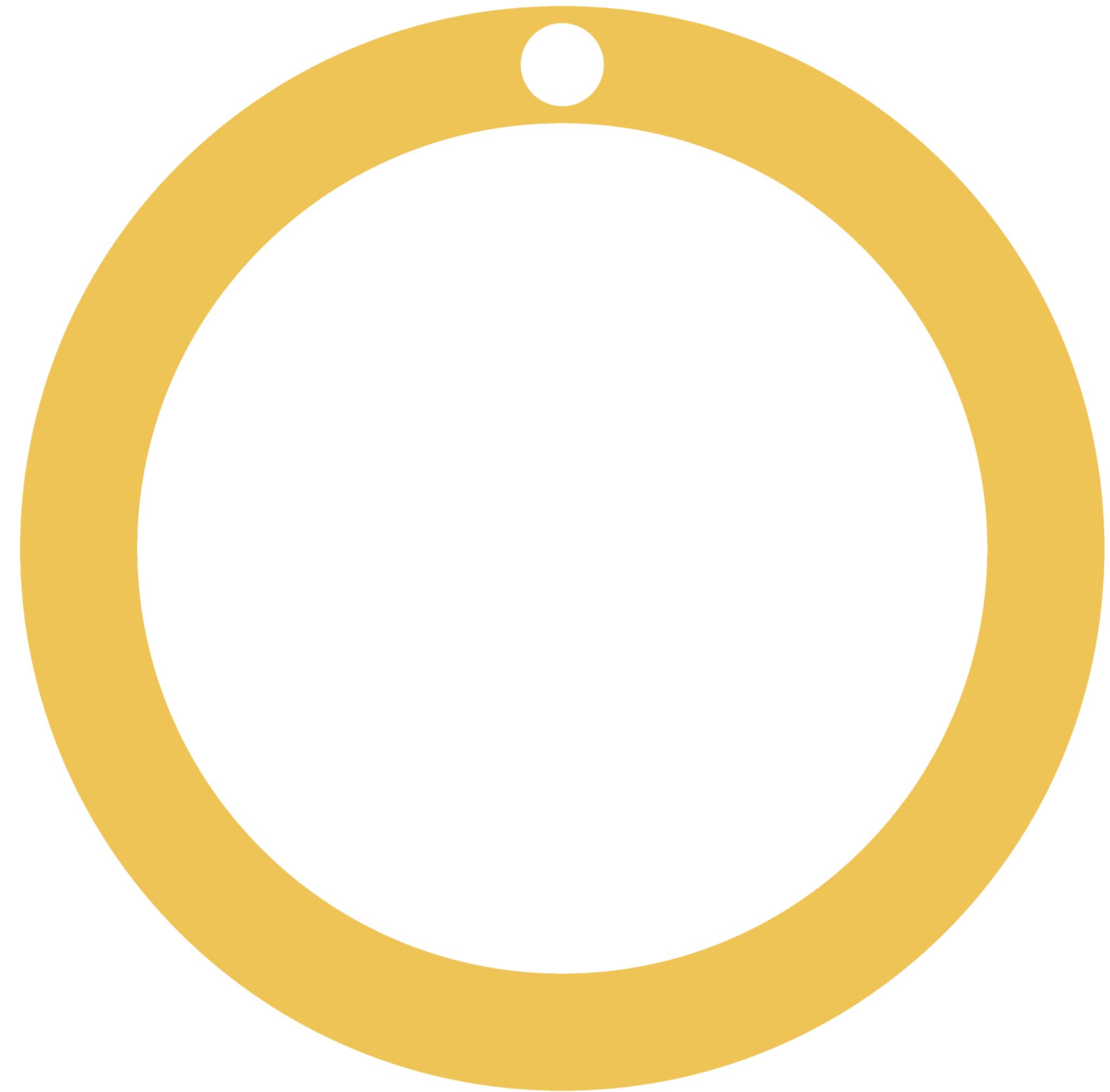
Task

Event: keydown

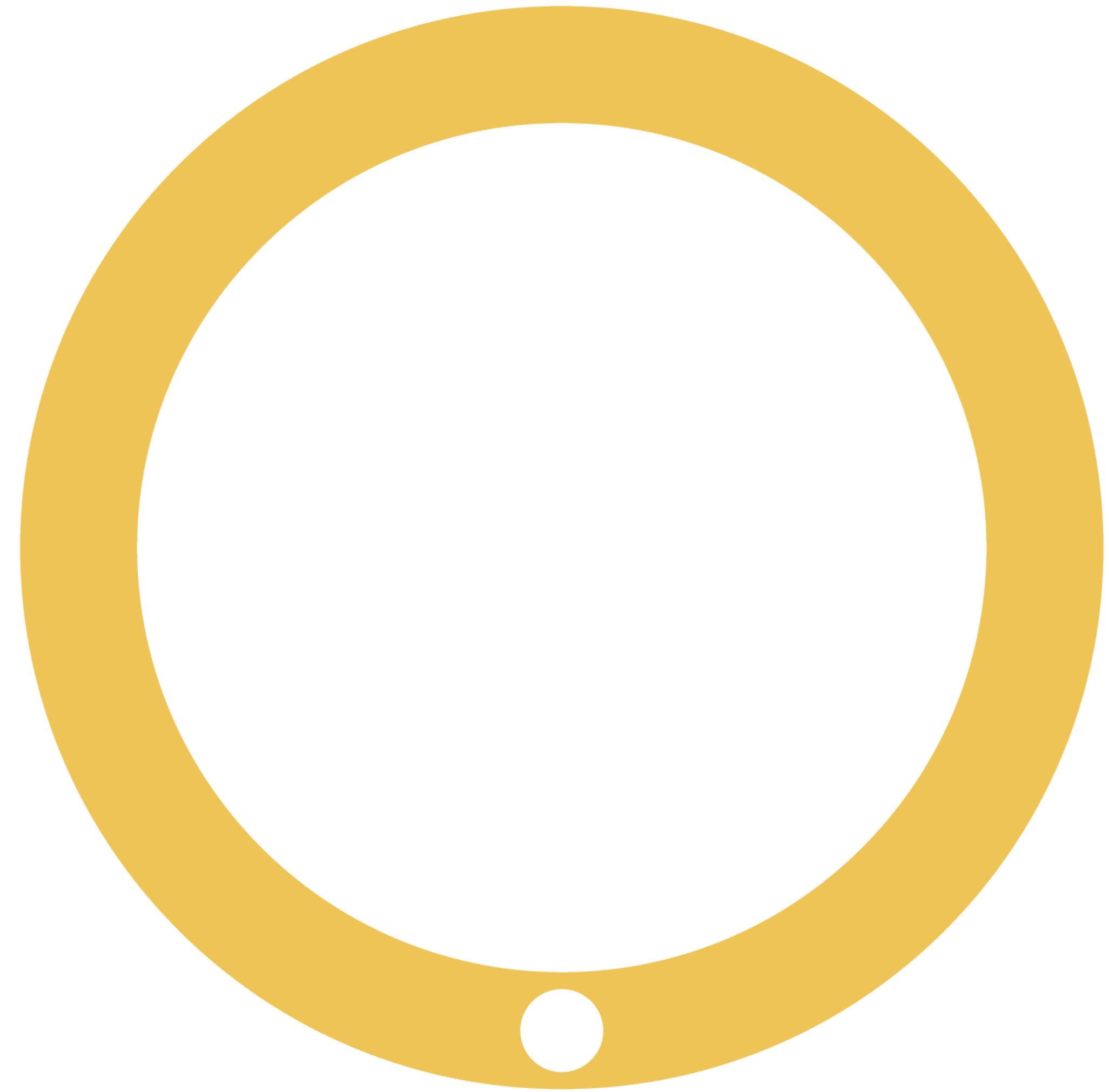
Function call

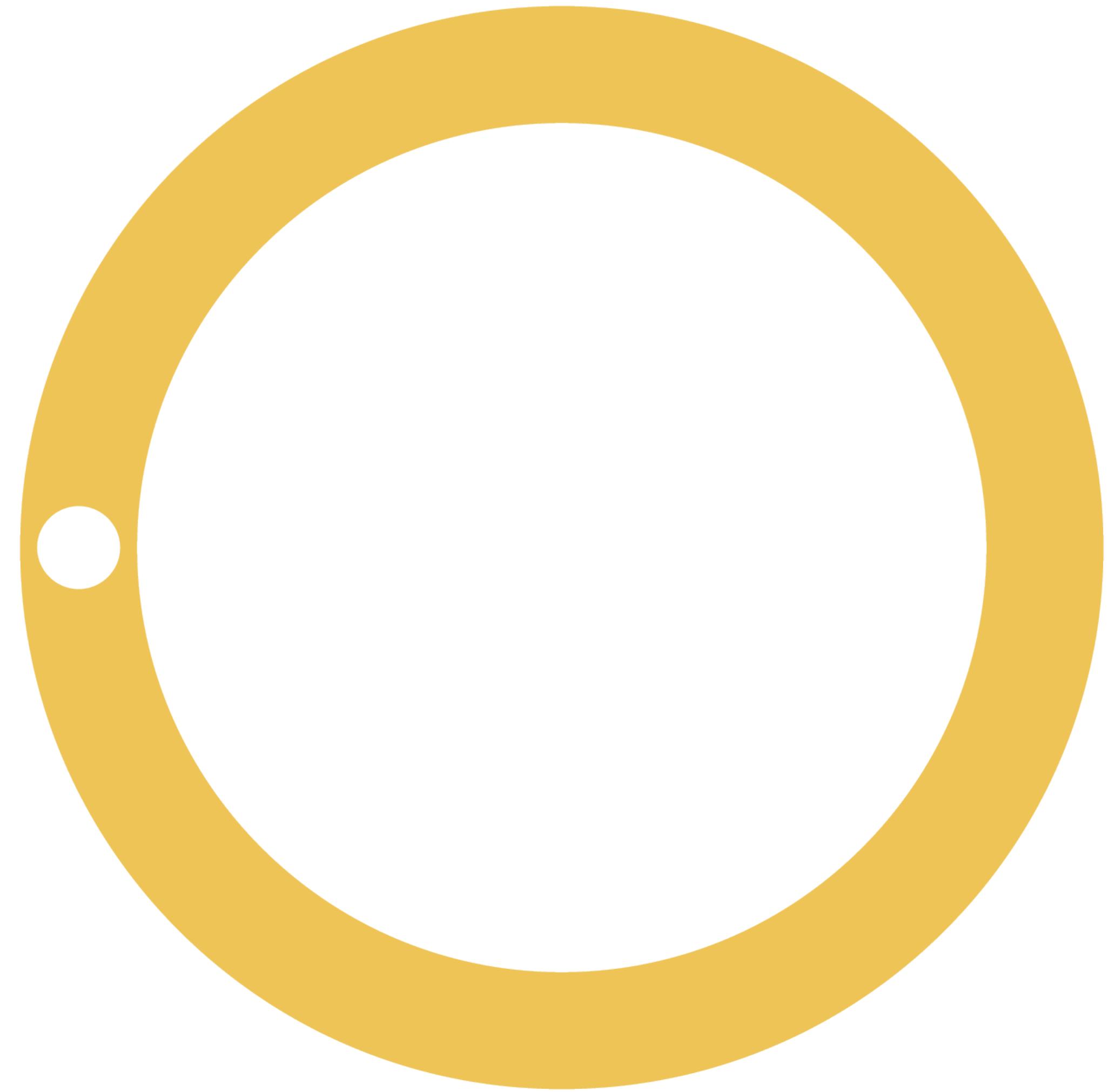
showSpinner

Why so much focus on tasks?



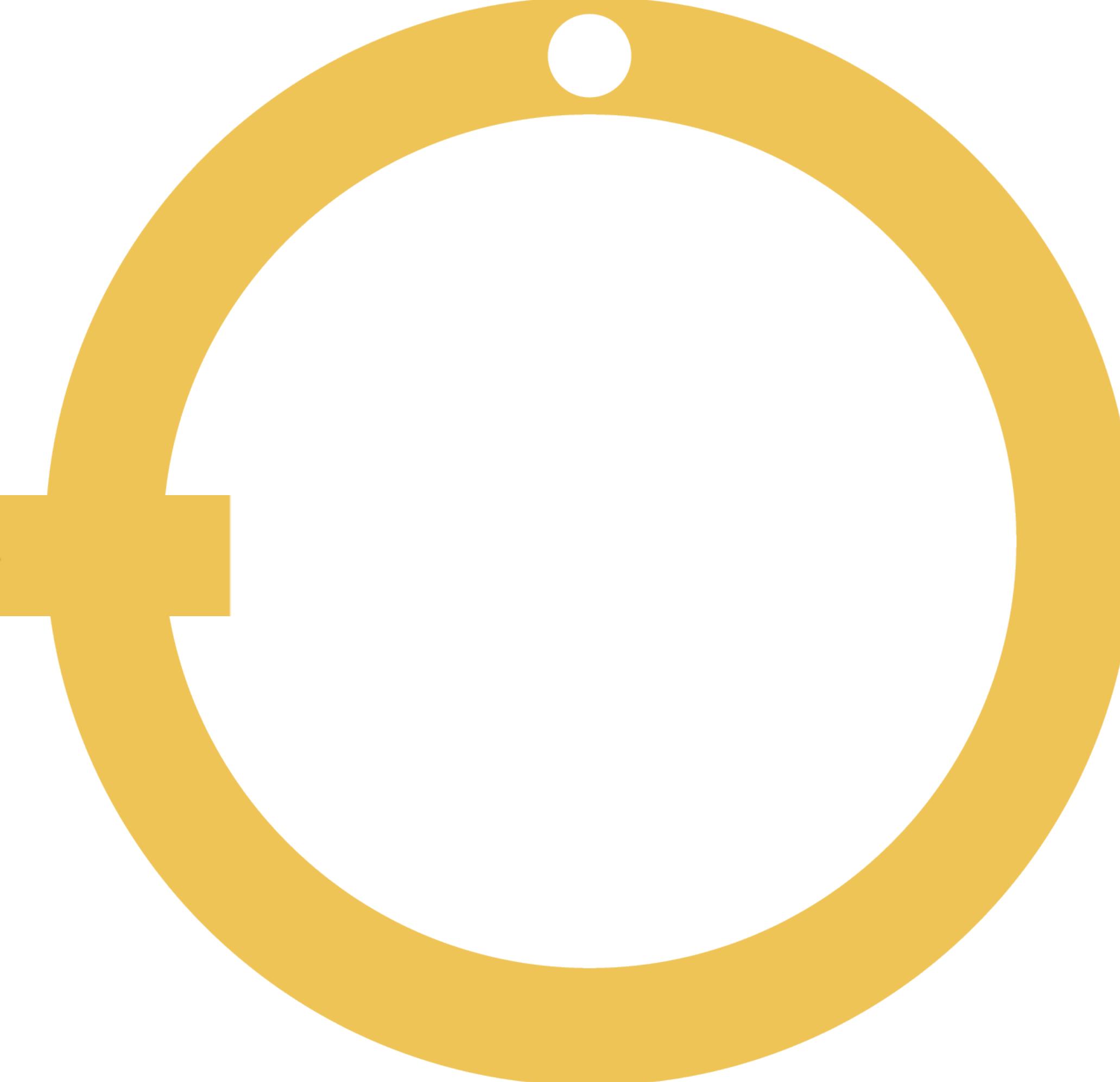








4GIFS.com



Event: keyup

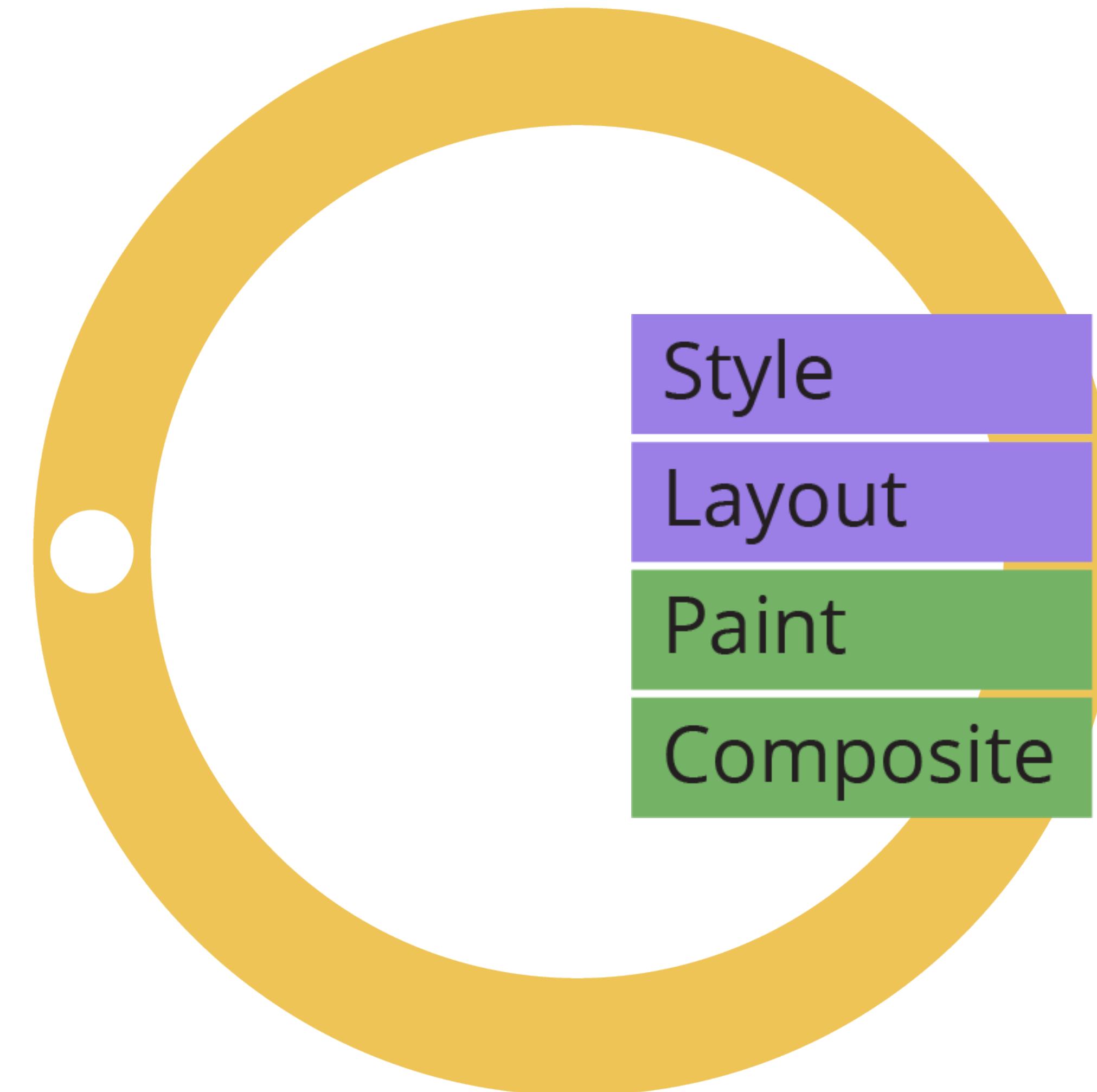
Event: keyup

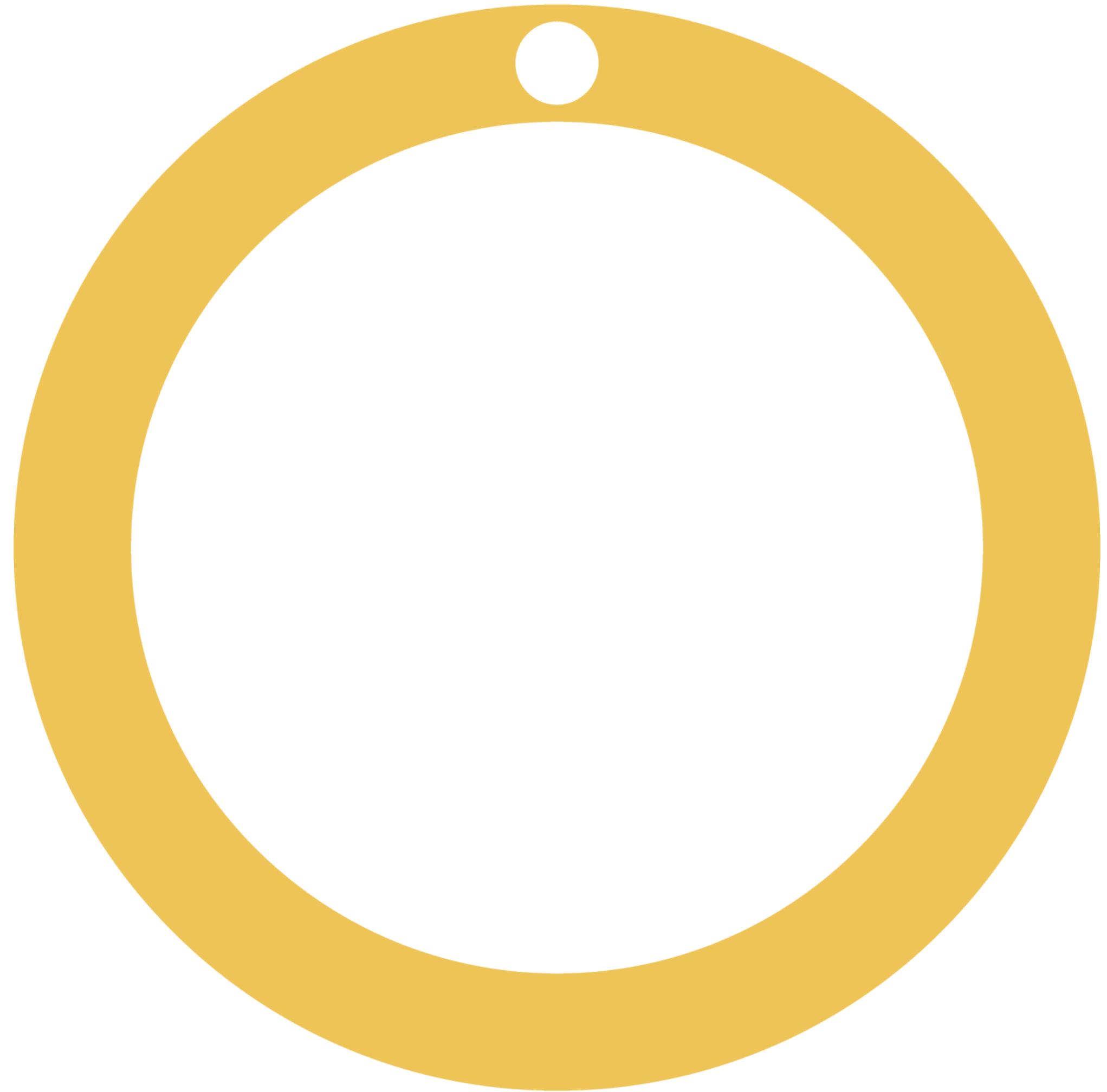


Event: keyup



Event: keyup





Style

Layout

Paint

Composite



Style

Layout

Paint

Composite

Page Unresponsive

You can wait for it to become responsive or exit the page.



Welcome — jlwagner.net

[Exit Page](#)

[Wait](#)

The PWA community is coming together for #PWASummit22. Have a great story about developing a web app? [Submit your talk today](#)

Interaction to Next Paint (INP)

May 6, 2022 — Updated Jul 18, 2022

Appears in: [Metrics](#)



Jeremy Wagner

[Twitter](#) [GitHub](#) [Homepage](#)

On this page



 SHARE

 SUBSCRIBE

- Observes certain interactions using the Event Timing API.
- Unlike First Input Delay (FID), measures more than just the input delay of the first interaction.
- Chooses the single worst* interaction latency (in milliseconds), which is representative of the page's overall responsiveness.

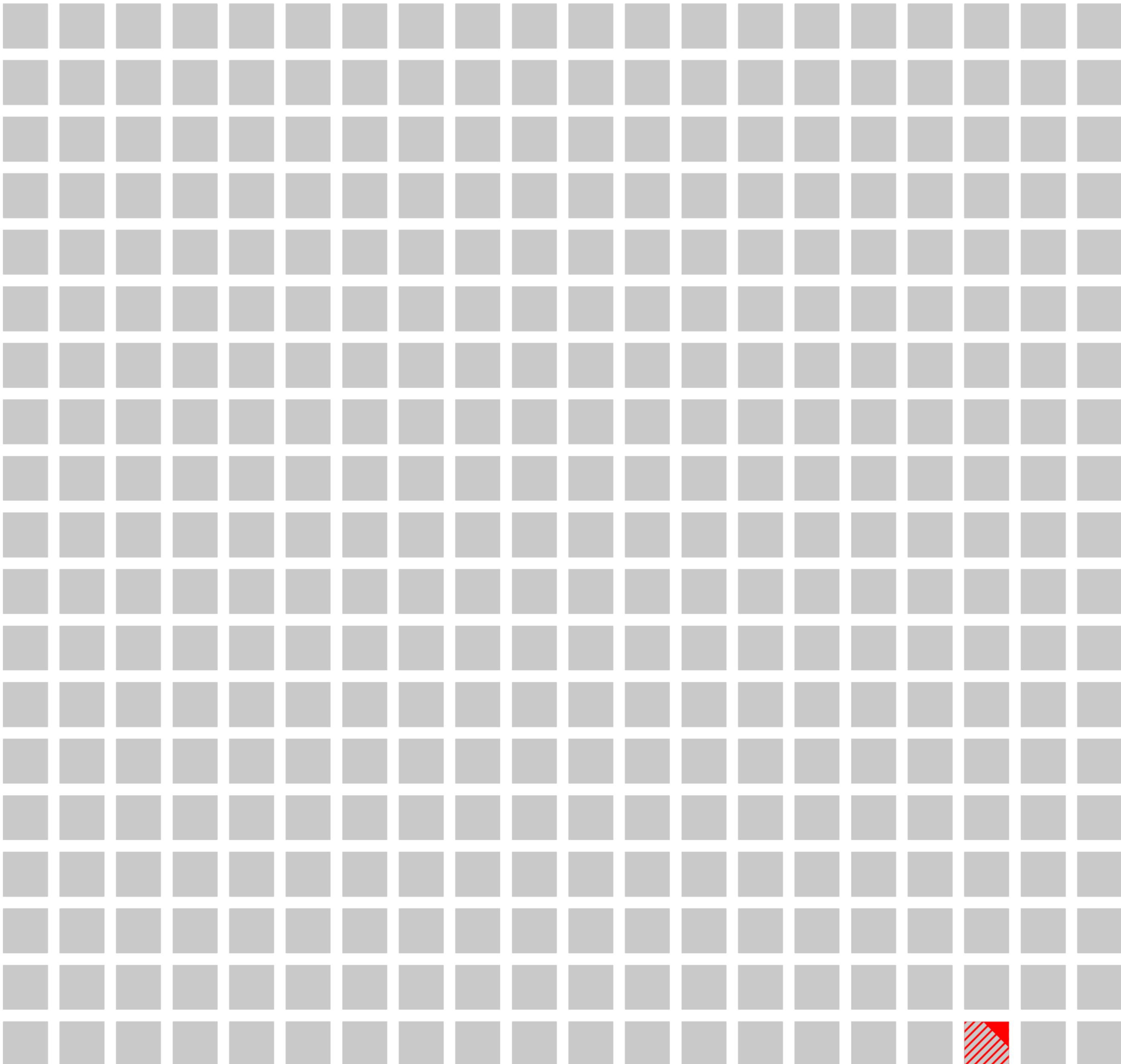
INP is all about the next frame.

- Keyboard interactions (including physical and onscreen).
- Mouse clicks.
- Taps on touchscreens.

- Hovering.
- Network activity.
- Scrolling*.

One interaction

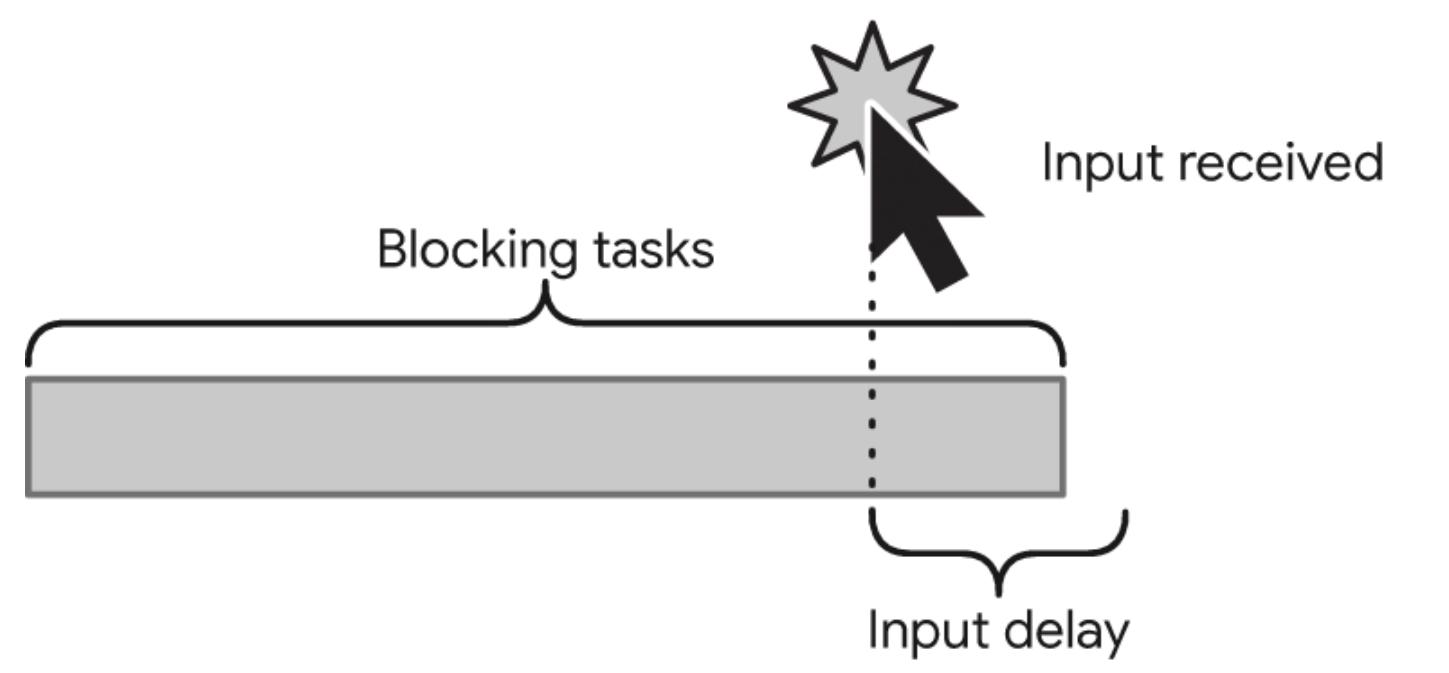


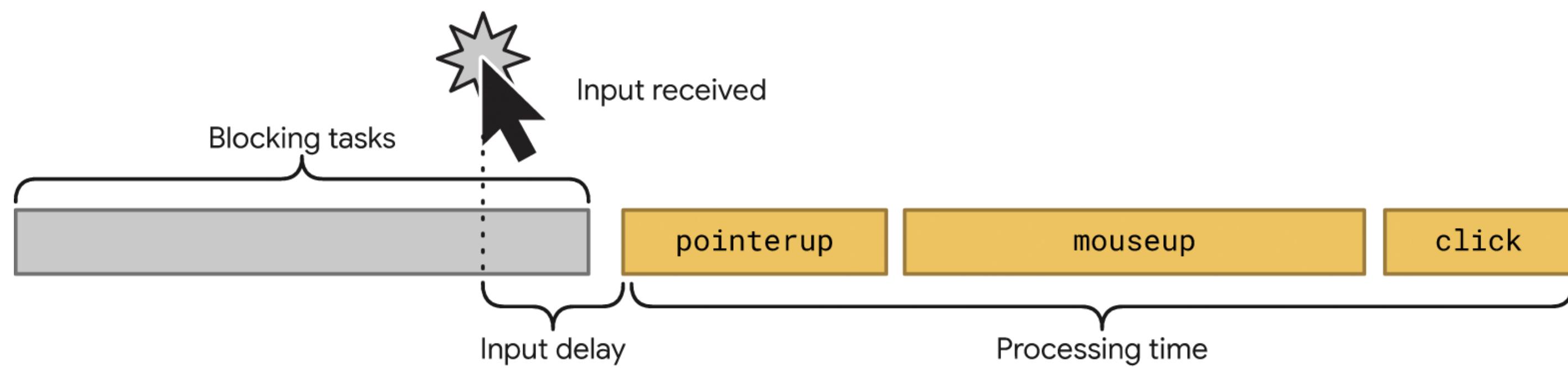


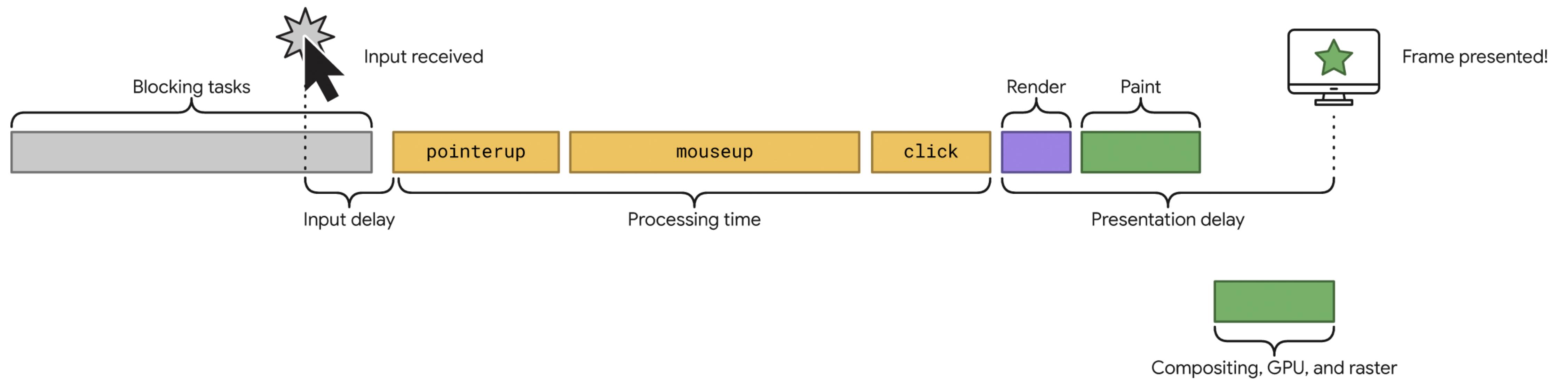
INP

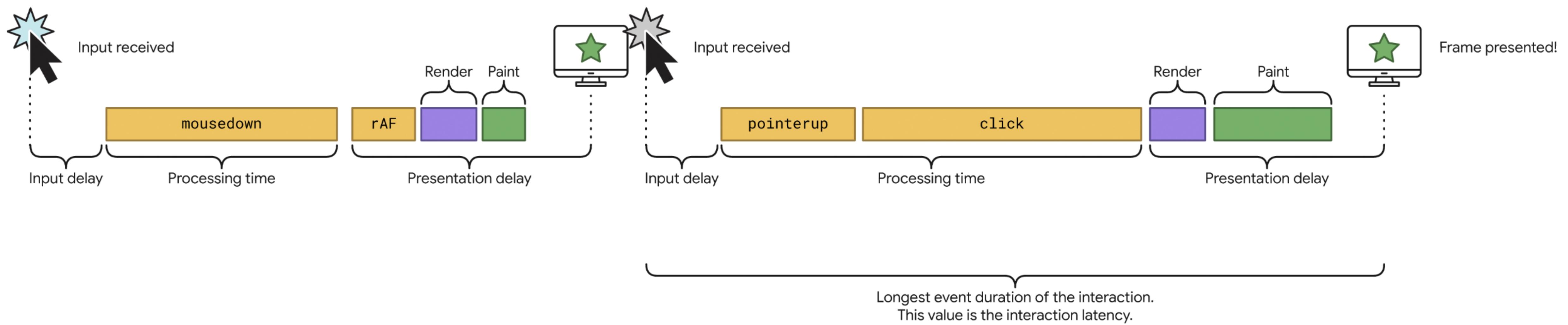
Interaction to Next Paint











Optimizing INP

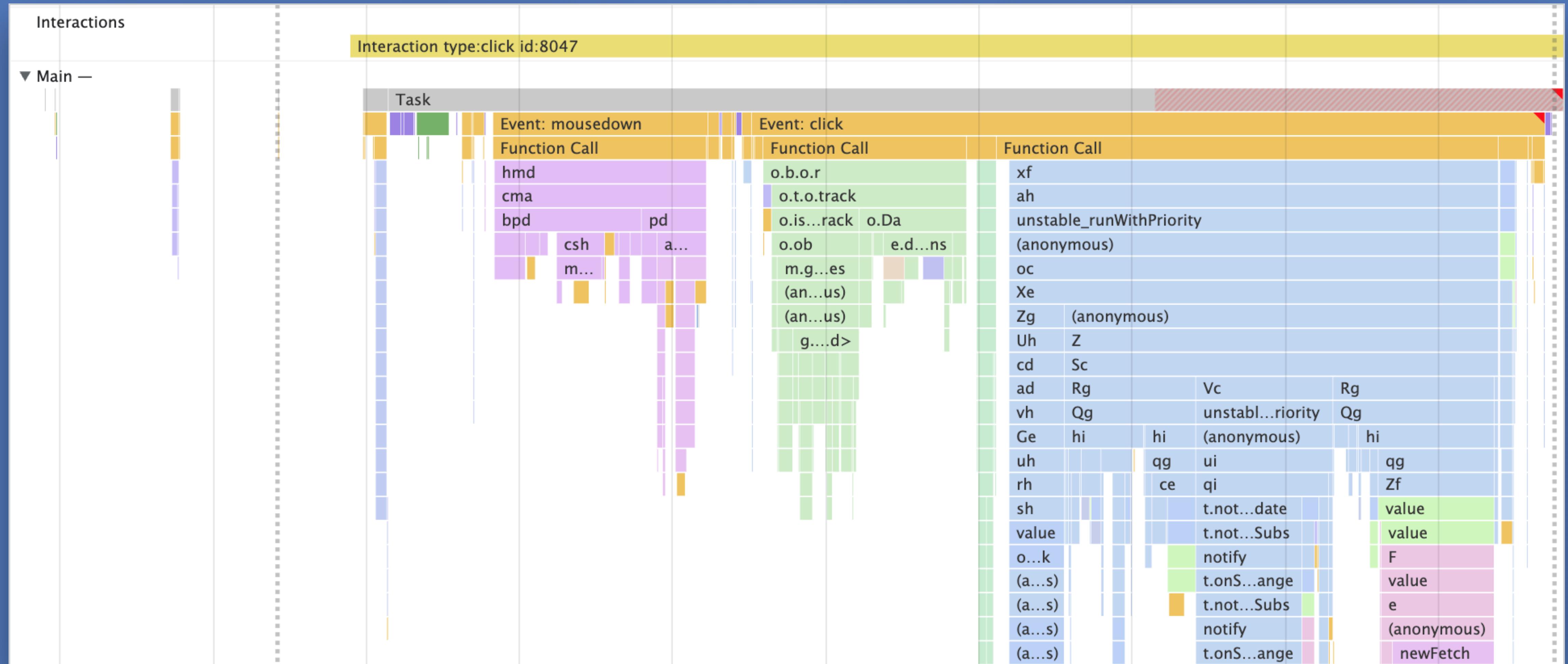
```
// Vendors
import { onINP } from "web-vitals";

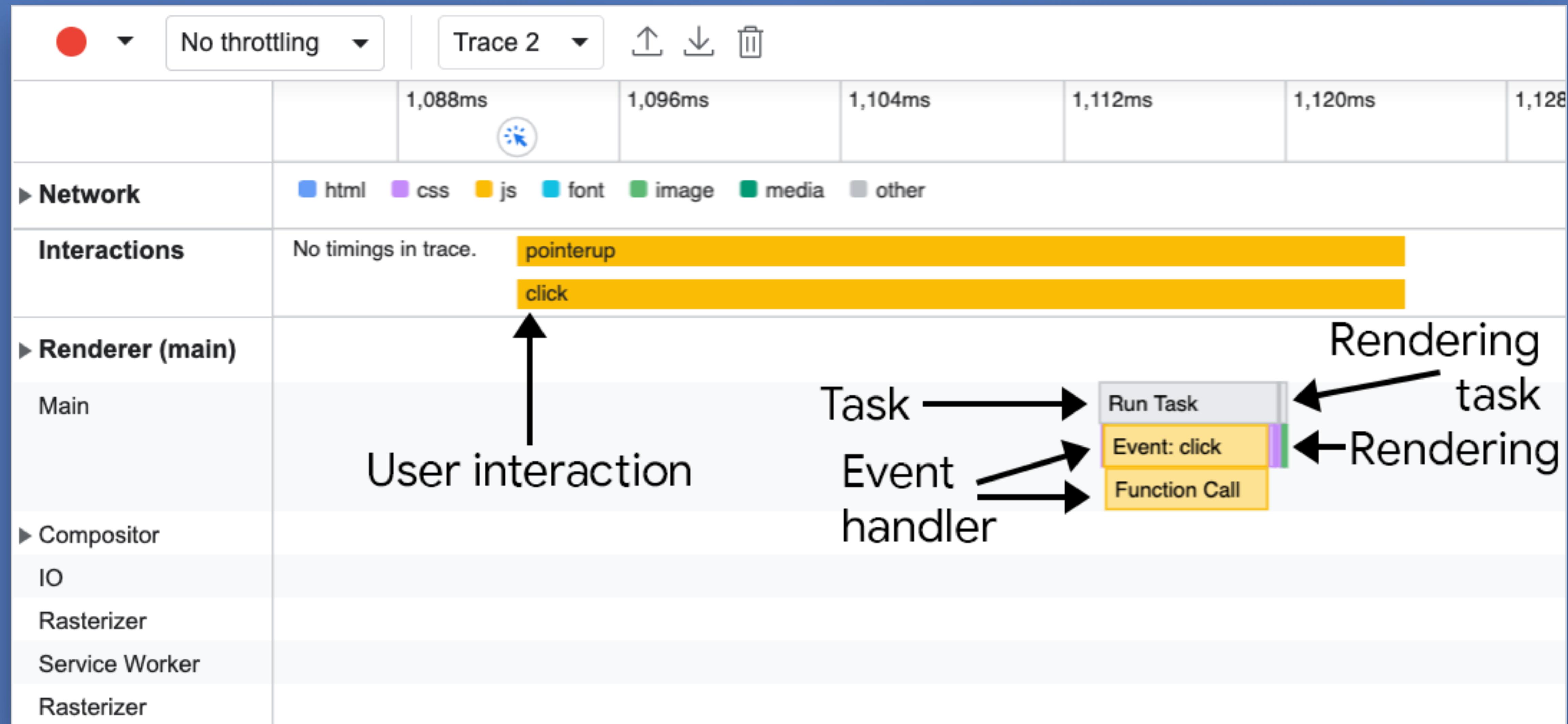
// https://gist.github.com/malchata/56e92306acb0c5b9fd3541ed16ce2d23
import { getSelector } from "./utils.js";

function sendMetrics({ metric, value, entries }) {
  fetch("https://mycoolwebsite.net/collect", {
    body: JSON.stringify({
      metric: value,
      element: getSelector(entries[0].target)
    }),
    method: "POST",
    keepalive: true,
    headers: {
      "Content-Type": "application/json",
    }
  });
}

onINP(sendMetrics);
```

Lab tools





Optimize!



[Home](#) > [All articles](#)

Optimize long tasks

You've been told: "don't block the main thread" and "break up your long tasks", but what does it mean to do those things? Find out in this article.

Sep 30, 2022



Jeremy Wagner

[Twitter](#) [GitHub](#) [Homepage](#)

On this page

[What is a task?](#)

[What is the main thread?](#)

[Task management strategies](#)

[Manually defer code execution](#)

[Use async/await to create yield points](#)

[Yield only when necessary](#)

[Gaps in current APIs](#)

[A dedicated scheduler API](#)

[Built-in yield with continuation](#)

[Conclusion](#)

SHARE

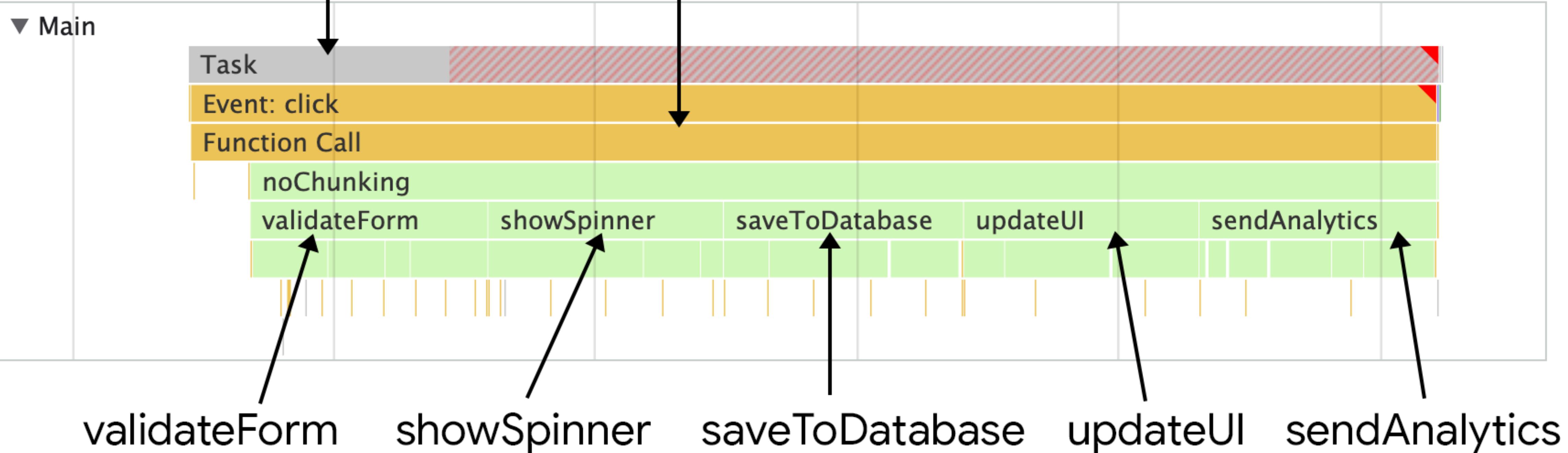
SUBSCRIBE

[web.dev/optimize-long-tasks](#)

If you read lots of stuff about web performance, the advice for keeping your JavaScript

```
function saveSettings () {  
    validateForm();  
    showSpinner();  
    saveToDatabase();  
    updateUI();  
    sendAnalytics();  
}
```

One long task saveSettings



```
function yieldToMain () {  
    return new Promise(resolve => {  
        resolve(setTimeout, 0);  
    });  
}
```

```
async function saveSettings () {
  // Do critical work that results in a
  // rendering update before anything else:
  validateForm();
  showSpinner();

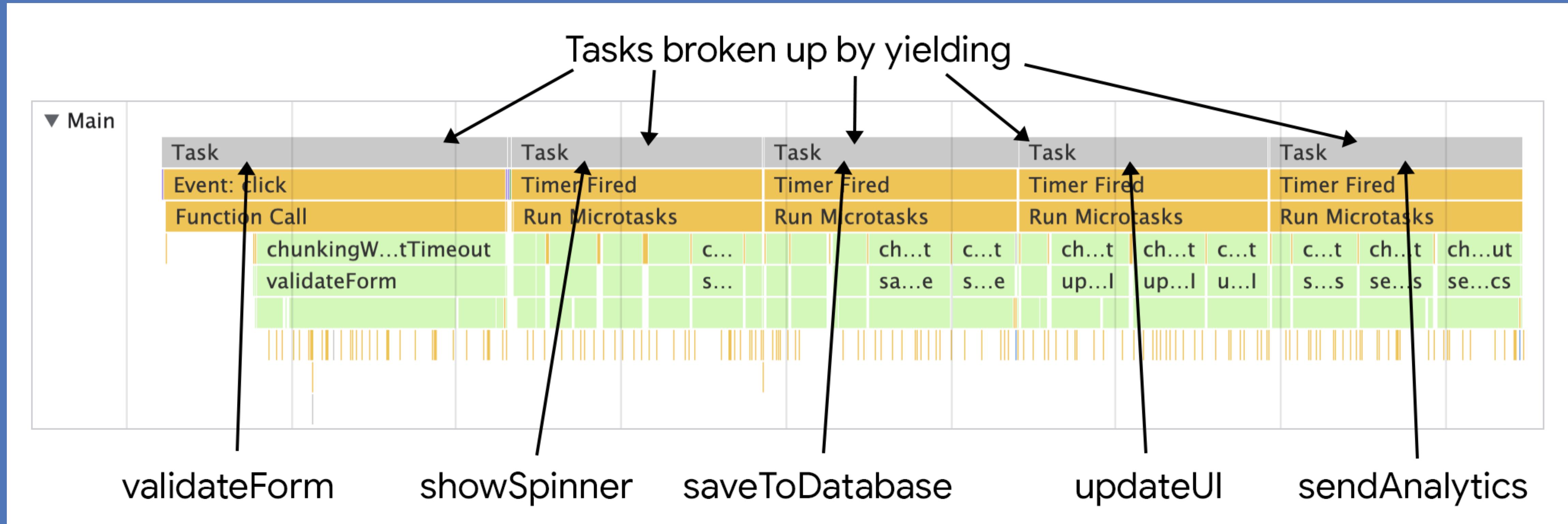
  // Create an array of functions to run:
  const tasks = [
    saveToDatabase,
    updateUI ,
    sendAnalytics
  ];

  // Loop over the tasks:
  while (tasks.length > 0) {
    // Shift the first task off the tasks array:
    const task = tasks.shift();

    // Run the task:
    task();

    // Yield to the main thread:
    await yieldToMain();
  }
}
```

Tasks broken up by yielding



You still need to do as
little work as possible.

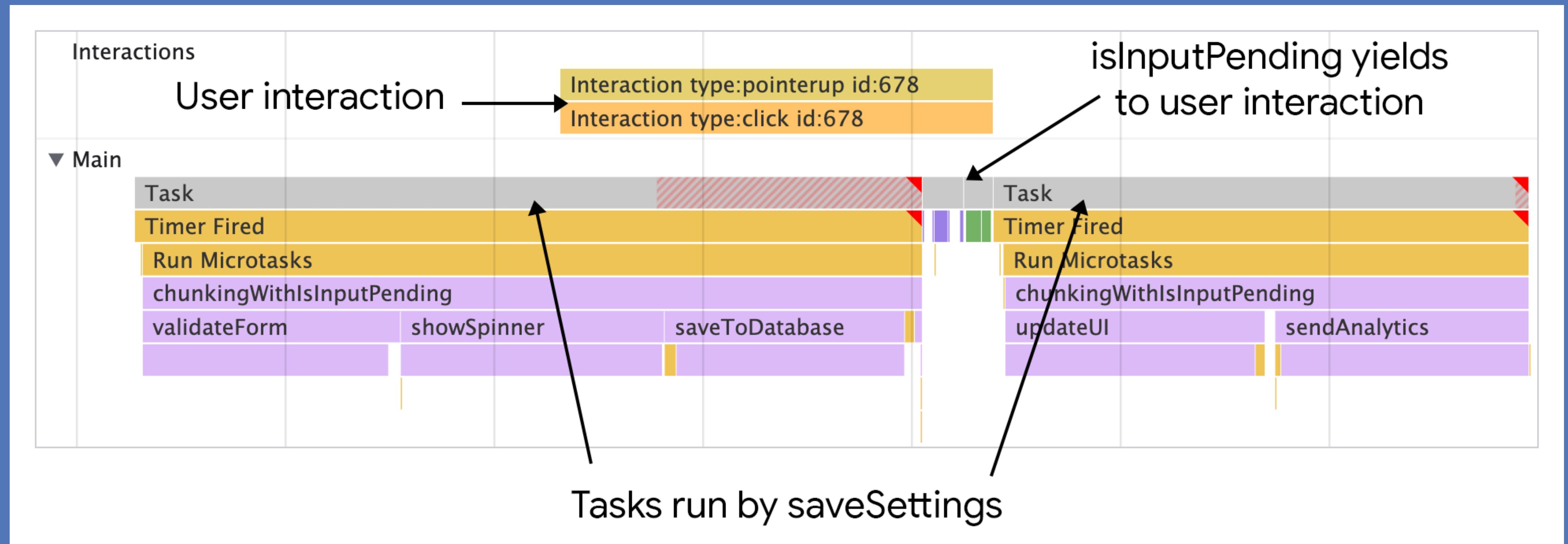
Yield to user input.

```
async function saveSettings () {
  // A task queue of functions
  const tasks = [
    validateForm,
    showSpinner,
    saveToDatabase,
    updateUI,
    sendAnalytics
  ];
  let deadline = performance.now() + 50;

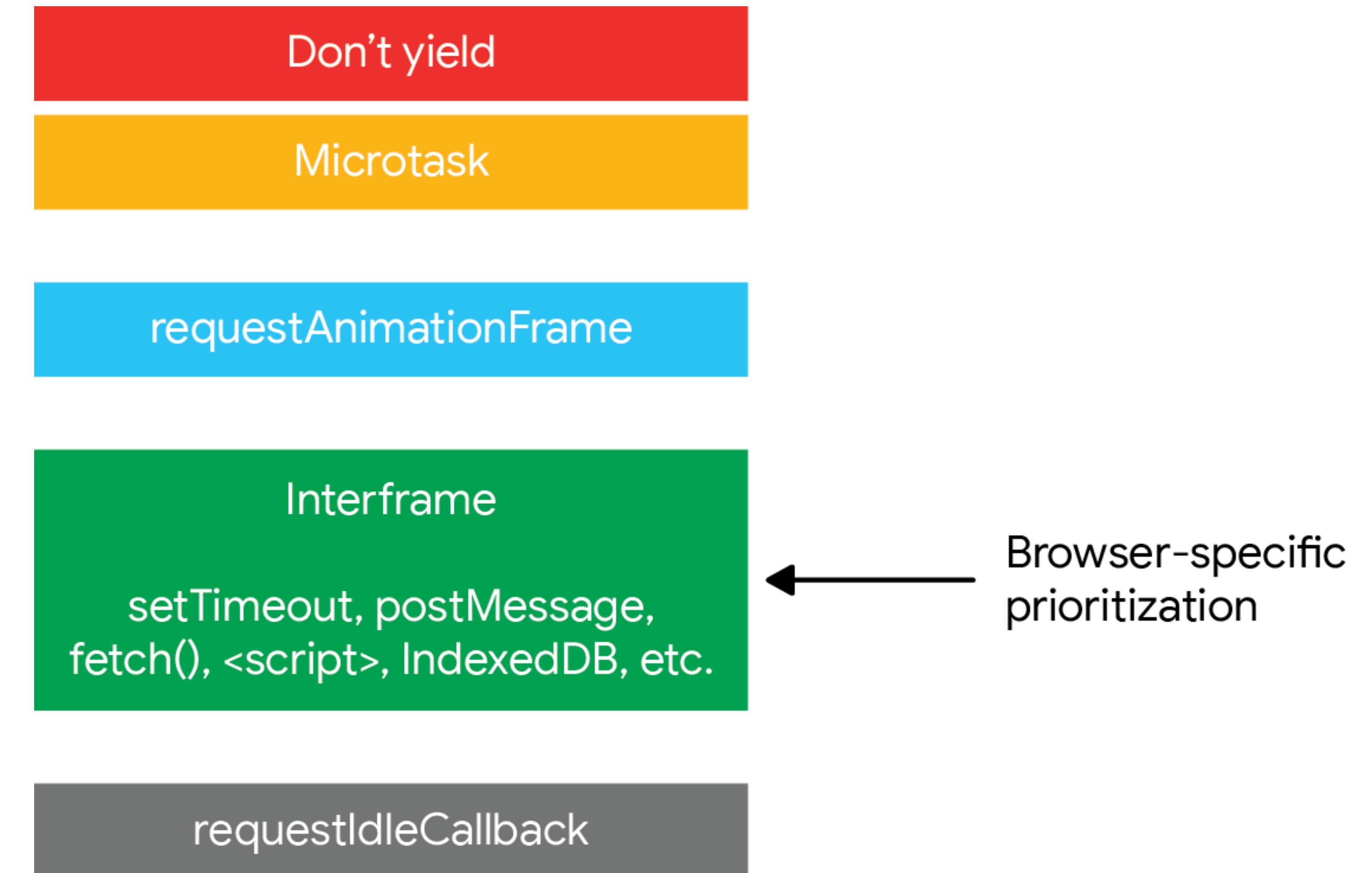
  while (tasks.length > 0) {
    // Optional chaining operator used here helps to avoid
    // errors in browsers that don't support `isInputPending`:
    if (navigator.scheduling?.isInputPending() || performance.now() >= deadline) {
      // There's a pending user input, or the
      // deadline has been reached. Yield here:
      await yieldToMain();

      // Extend the deadline:
      deadline = performance.now() + 50;
    } else {
      // Shift the the task out of the queue:
      const task = tasks.shift();

      // Run the task:
      task();
    }
  }
}
```



Prioritize work.



Don't yield

Microtask

`requestAnimationFrame`

Existing Task
Sources

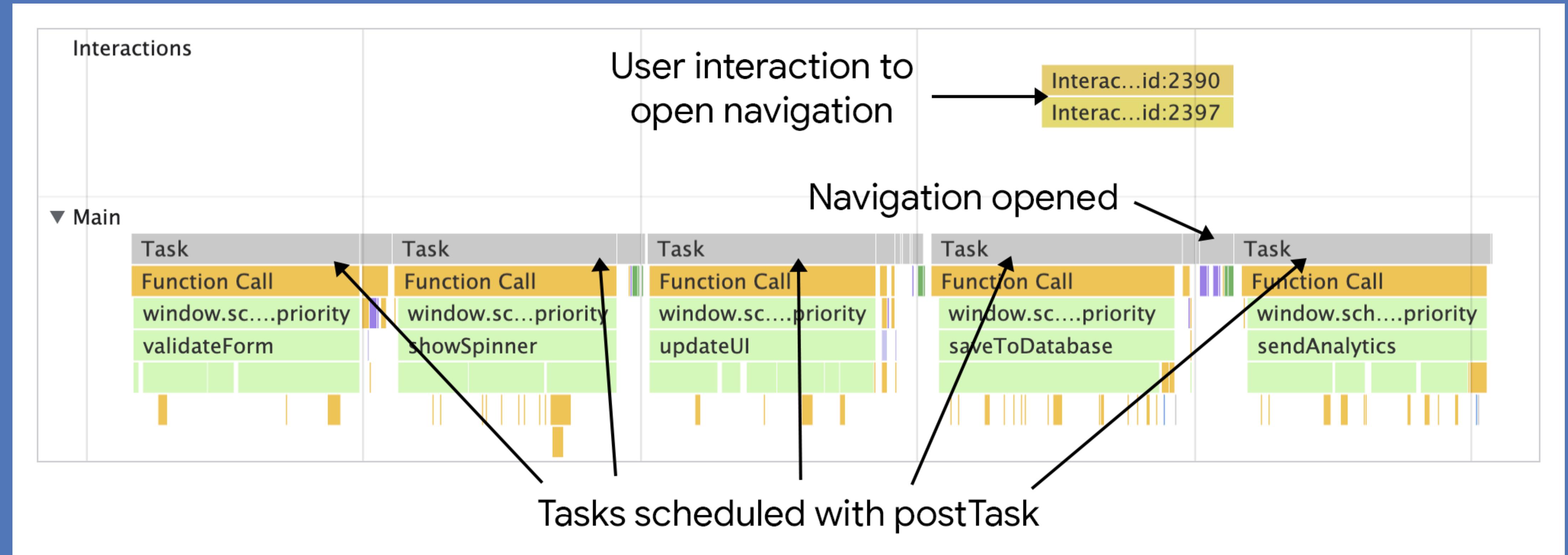
(UA-specific
prioritization)

Prioritized tasks

user-blocking
user-visible
background

`requestIdleCallback`

```
function saveSettings () {  
  // Validate the form at high priority  
  scheduler.postTask(validateForm, {priority: 'user-blocking'});  
  
  // Show the spinner at high priority:  
  scheduler.postTask(showSpinner, {priority: 'user-blocking'});  
  
  // Update the database in the background:  
  scheduler.postTask(saveToDatabase, {priority: 'background'});  
  
  // Update the user interface at high priority:  
  scheduler.postTask(updateUI, {priority: 'user-blocking'});  
  
  // Send analytics data in the background:  
  scheduler.postTask(sendAnalytics, {priority: 'background'});  
};
```



Optimize Interaction to Next Paint



[Home](#) > [All articles](#)

SHARE

SUBSCRIBE

Optimize Interaction to Next Paint

web.dev/optimize-inp

INP is evolving.



Thank you for your time.

Jeremy Wagner – jlwagner.net – @malchata
BiznagaFest 2002 – Málaga, España