Towards improving FID

npm

What's FID?

First Input Delay is the delta between:

- timeStamp
- Time right before handlers are about to begin running

For the first of the following:

- keydown
- mousedown
- pointerdown followed by pointerup
- click

Some successes of FID

- Encourage people to break up their longtasks
- Tap delay issue surfaced
  - Use mobile viewports
  - Chrome is planning to get rid of the delay for some of the existing cases

How do we want to improve FID?

- Consider more than just the first input
- Potentially include scroll begin
- Evaluate a larger chunk of end-to-end latency

Evaluating more than just the first

We want to evaluate the following interactions, with multiple associated events:

- Keyboard
  - keydown
  - keyup
- Taps or drags
  - pointerdown and pointerup
  - touchstart and touchend
  - mousedown and mouseup
  - click
- Scrolls: not necessarily blocked on any event

Need to associate Event to interaction

- If we want to compute a metric value that associates the events of a single interaction, we want developers to have this association too. For example, the keydown-keyup match, etc.
- This would be solved by the proposed interactionID which I've presented before.
- Other ideas?

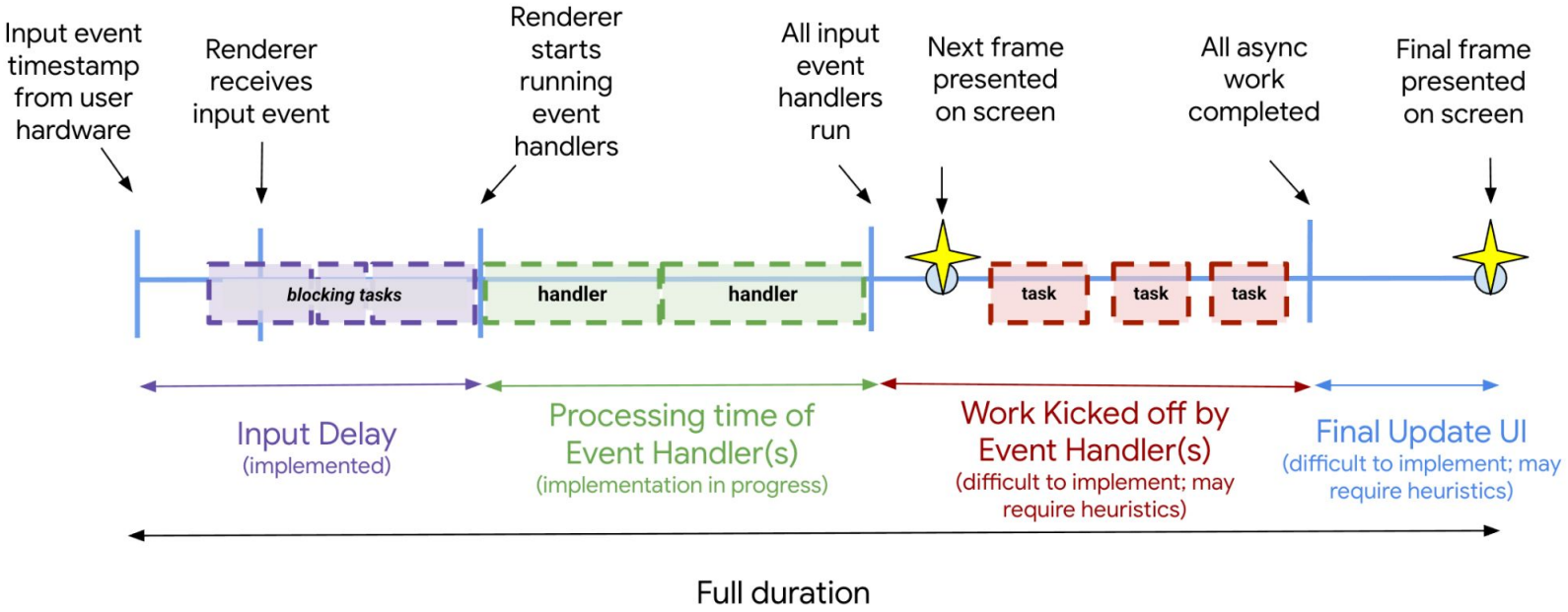
Enable measuring scrolling

- We need a new API to do this as there is no way for developers to do this right now.
- Scroll performance can be impacted by developers, for instance if they force scrolling to happen in the JS thread.
- There is a need to measure initial scroll reaction (scroll begin).
- Use cases for subsequent scroll reactions (scroll updates) would probably be better addressed by Frame Timing.

How can we expose initial scroll reactions?

- New PerformanceEventTiming entry
  - Name (usually reserved for event type):
    - scrollbegin
    - “scroll” would be confusing with the scroll Event
- New (Performance)ScrollTiming entry
- Other?

Evaluating end-to-end of interaction



Heuristics are hard

- Asynchronous work is hard to track
- We can at least improve over FID by looking at the frame produced as a result of event handlers running.
  - “Next frame presented on screen” in the diagram

Questions or thoughts?