

# Time To Interactive

## Metrics and calculation

bckenny@, paulirish@, tdresser@

Created: June, 2016

Last Updated: 19 July 2016

Status: draft

Time to interactive is a progressive web metric which seeks to report when a page first looks and feels interactive to the user.

Tracking bug [here](#).

[Time To Interactive](#)

[TTI At A Glance](#)

[TTI in Detail](#)

[Diagnostic Metrics](#)

[EQT between FMP and TTI](#)

[Online Computation](#)

[Alternatives](#)

[Not Just FMP](#)

[Main thread availability alternatives](#)

## Time To Interactive

A page load has a number of important timings to capture. (See: [Progressive Web Metrics](#)). While FCP and FMP mark important early delivery of content to the user, Time To Interactive (TTI) seeks to identify a more mature time where the page looks and feels interactive to the user.



■ **First Contentful Paint** “is it happening?”

Has my navigation started successfully (the server has responded, etc.)?

■ **First Meaningful Paint** “is it useful?”

Has the page painted enough critical content that I could engage with it?

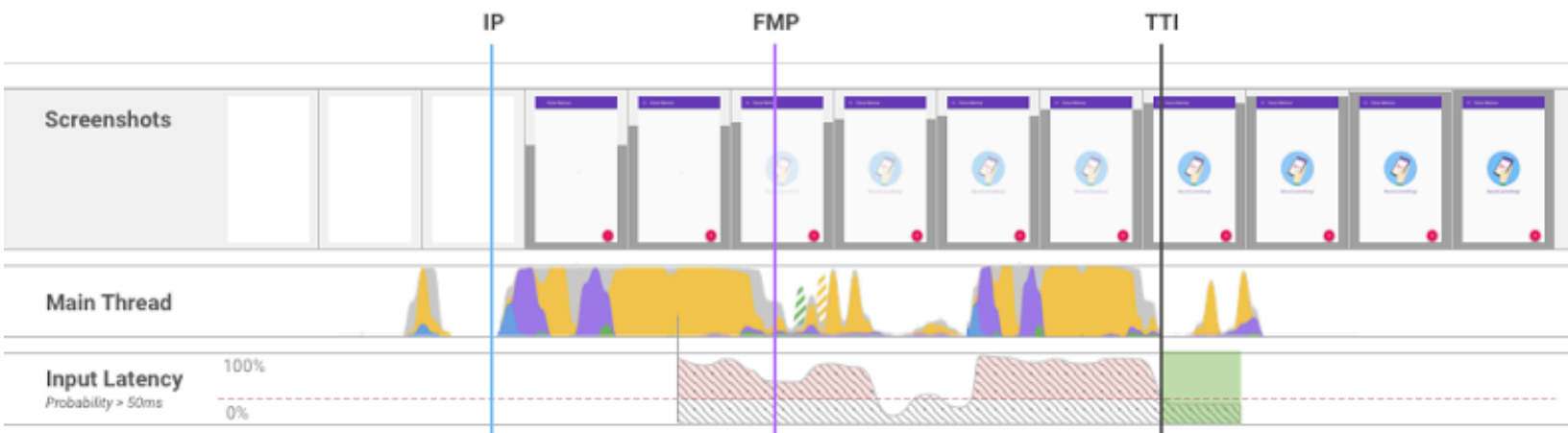
## ■ Time to Interactive “is it usable?”

Can I actually engage with the page or is it still busy?

## TTI At A Glance

TTI is marked when **Layout has stabilized & key webfonts are visible**, and *The main thread is available enough to handle user input*.

## TTI in Detail



### **Page is looking nearly done**

- First contentful paint and/or DOMContentLoaded has fired.

### **The main thread is available enough to handle user input**

- If the main thread isn't going to be responsive to the user, then the page shouldn't be considered interactive.
- We consider the page interactive during any time window of length > 10 seconds which contains 0 tasks longer than 50ms.
- Our first candidate window starts at FMP and ends 10 seconds later.
  - If satisfied, our TTI is the beginning of the window.
  - If not, then we reposition left edge of the window at the end of the window's last 50ms task and try again.
- If no such window exists, the page has an infinite TTI, which is okay.

## Diagnostic Metrics

[EQT](#) between FMP and TTI

We should expose the expected queueing time for the time window in which lack of main thread responsiveness is preventing TTI from firing:

This window is [FMP, TTI].

We should also expose the [diagnostic metrics for EQT](#) during this window.

## Online Computation

At the end of each time window that the page was interactive for, input code will send the page load logic the time window's start and end timestamps. Loading logic will store all window durations until it computes FMP. Since each window will be > 10 seconds long, There should be a fairly small number of windows before FMP. Once we've computed FMP, there are 3 cases:

- There's a window of interactivity which includes FMP, FMP == TTI
- We know about a window of interactivity after FMP, use its start timestamp
- We haven't seen the page become interactive since FMP, report the start time of the next reported window of interactivity

## Alternatives

### Not Just FMP

We could wait for additional signals before starting to check main thread availability. These signals may not be available everywhere we want to surface these metrics. We may end up surfacing additional metrics in some places - perhaps "Visual FMP" and "Visual TTI"?

The user believes the page is done enough to start interacting with

1. Layout has stabilized & key webfonts are visible (FMP)
2. Page looks nearly done

The page is actually ready for user:

1. domContentLoaded has fired
2. The main thread is available enough to handle user input

### Page looks nearly done

- Using [Perceptual Speed Index \(slide deck\)](#), we evaluate the visual completion of the page. While this requires a fairly expensive calculation of visual similarity across screenshots, it provides considerable confidence that the user considers the page near done and ready to interact with.
- Based on a manual evaluation of 50 page loads, 85% Visually Complete hits a sweet spot of feels ready enough even though I know it's not totally done. 80% starts hitting false positives.
- Caveat: This metric considers only above-the-fold.

*Requirements: Perceptual Visual Completion @ 85%*

### domContentLoaded has fired

- DCL fires when HTML parsing has finished, but we're interested in when all DCL event handlers have completed.
- Many event handlers are often bound to DCL, which leads to two concerns we want to

mitigate. 1) Any visual changes to the page made in DCL are complete. 2) No surprises of inactive buttons/actions as DOM element event handlers should be bound

Requirements: [domContentLoadedEventEnd](#) has fired.

## Main thread availability alternatives

There have been many approaches discussed to identify a timestamp of interest for TTI.

### 1. First time window of 10s where there are 0 tasks over 50ms

- (Described above)
- Left edge of window is our TTI timestamp

### 2. Compute from ILR's [RAIL Likelihood](#)

- An earlier approach wanted a 10 second window where the RAIL likelihood was 75%. Once satisfied, an inner window would work backwards to find the largest window where the RAIL Likelihood was 90%.
  - The left edge of that window would be the TTI timestamp.
- There are discussions around using EQT as a distribution, it's mean, or median. These approaches may have different implications for TTI. Currently we need more data to validate what matches reality best.