

# Extensible Scrolling

11/5/14

Tim Dresser - [tdresser@chromium.org](mailto:tdresser@chromium.org)

[Rick Byers](#) - [rbyers@chromium.org](mailto:rbyers@chromium.org)

[BlinkOn 3 recording](#)

# Motivation

We needed this 6 years ago!

# Vision

Competing with native requires [extensibility](#)

# 1. Scroll synchronization

A) Main thread

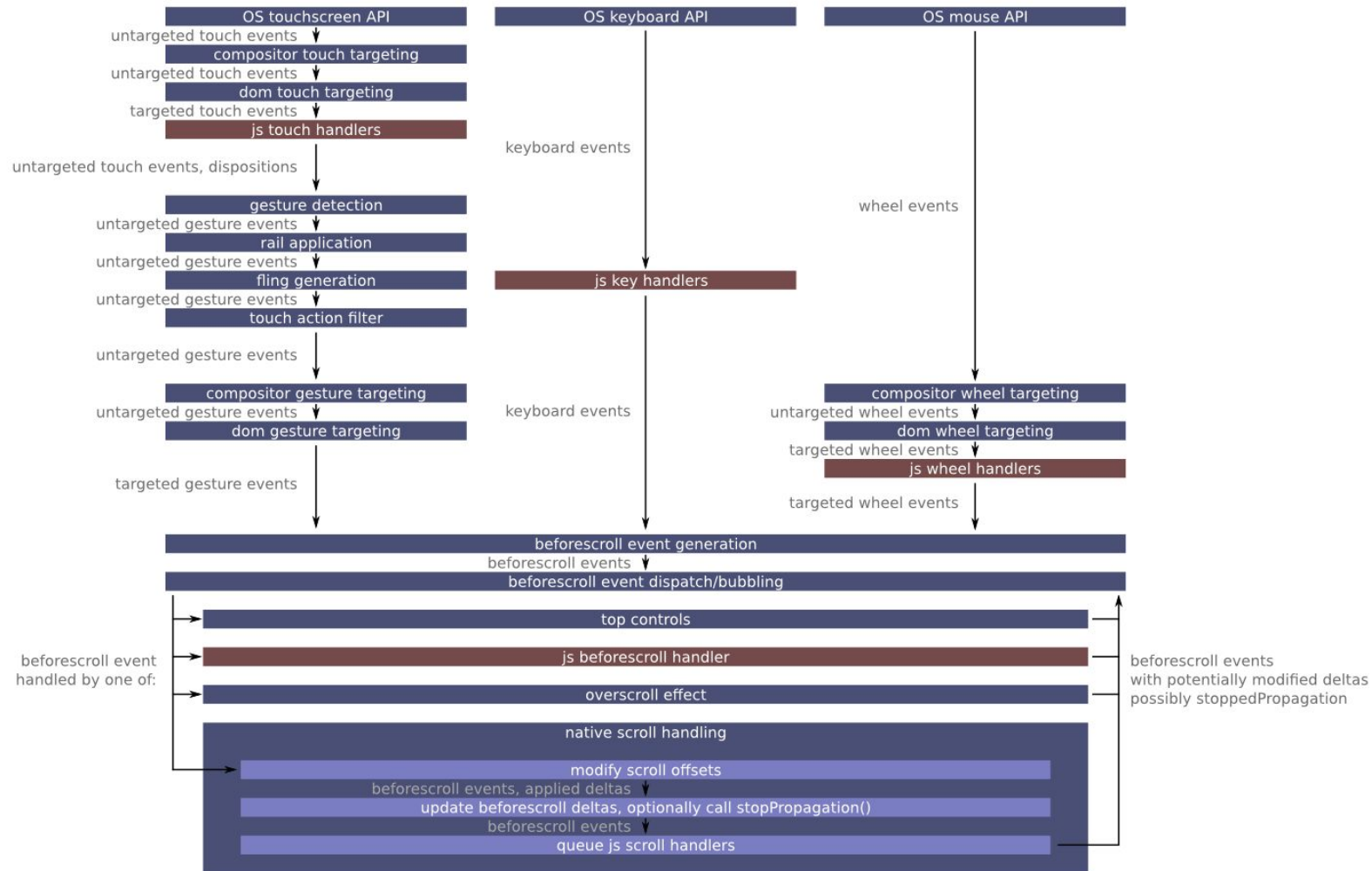
scroll-delay

B) Compositor thread

UIWorker

**A healthy web requires that we do both!**

## **2. Scroll customization**



# Design

Here's ~~exactly~~ what this means.  
sort of

# scroll-delay

Bug [347272](#)

‘scroll-delay’

Value: none |

| [ start-touch || wheel-event || scroll-event ]

Initial: none

Applies to: all elements

Inherited: no

UA style sheet:

html { scroll-delay: start-touch wheel-event }



# UIWorker

See Ian Vollick's talk  
(1:15 - Pacific Ocean)

# beforescroll

Bug [416862](#)

```
Element.prototype.scroll = function(scrollState) {  
  var dx, dy;  
  if (scrollState.deltaY && this.canScrollY) {  
    if (scrollState.deltaY > 0) {  
      dy = Math.min(scrollState.deltaY,  
                     this.scrollHeight - this.scrollTop);  
    } else {  
      dy = Math.max(scrollState.deltaY,  
                     -this.scrollTop);  
    }  
    this.scrollTop += dy;  
  }  
  if (scrollState.deltaX && this.canScrollX) { ... }  
  scrollState.consumeDelta(dx, dy);  
}
```

# beforescroll

```
Element.prototype.distributeScroll = function(scrollState) {  
  scrollState.popScrollChain().distributeScroll(scrollState);  
  this.scroll(scrollState);  
}
```

# beforescroll

```
document.scroll = function(scrollState) {  
  // Handle top controls  
  scrollState.consumeDelta(0, topControls.scroll(scrollState.deltaY));  
  if (scrollState.inInertialPhase || scrollState.isEnding)  
    topControls.snap();  
  
  Element.prototype.scroll(scrollState); // "super" in ES6  
  
  // Handle overscroll  
  if (scrollState.deltaX || scrollState.deltaY)  
    overscrollEffect.scroll(scrollState);  
}
```

# Demos

Require custom build

([blink side](#), [chromium side](#)) with beforescroll

[Snap Points](#)

[Pull To Refresh](#)

# Q&A

Discussion: [input-dev@chromium.org](mailto:input-dev@chromium.org)

# beforescroll - obsolete DOM event

```
Element.prototype.onbeforescroll = function(evt) {  
    var dx, dy;  
    if (evt.deltaY && this.scrollHeightY) {  
        if (evt.deltaY > 0) {  
            dy = Math.min(evt.deltaY, this.scrollHeight - this.scrollTop);  
        } else {  
            dy = Math.max(evt.deltaY, -this.scrollTop);  
        }  
        this.scrollTop += dy;  
    }  
    if (evt.deltaX && this.scrollOverflowX) { ... }  
    evt.consumeDelta(dx, dy);  
}
```