Performance Techniques in 2017

Getting native performance with new Web APIs

Manu Martínez-Almeida

Core Developer at Ionic

Github: @manucorporat

METRICS

WHY PERFORMANCE MATTER TO US?

CAN WEB APP BE AS PERFORMANT AS NATIVE APPS?

OF COURSE!! NOT

BUT!

ABSOLUTE TERMS MEAN

NOTHING

Accepted minimums

- **16ms per frame** to have a 60FPS
- Under **80ms for user interactions** (clicking button)
- Startup-time
  - Native apps can be extremely slow too
  - Browsers are still being optimized for this
  - First time: **under 3-4 seconds** is acceptable

OF COURSE

YES!

Start-up time

Tooling

Runtime

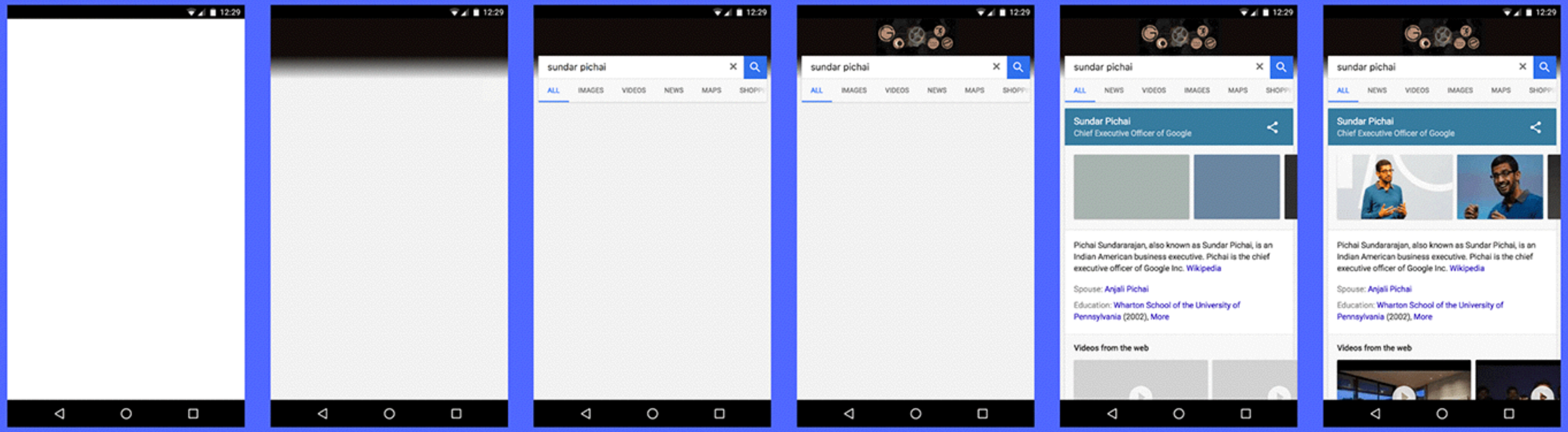
Rendering

Startup time

- First Paint (FP)
- First Contentfull Paint (FCP)
- First Meaningful Paint (FMP)
- Time to Interactive (TTI)

- Lazy loading
- Priorization
- Small bundles
- Web workers
- Server rendering

Startup time



First Paint (FP)

First Contentful Paint (FCP)

First Meaningful Paint (FMP)

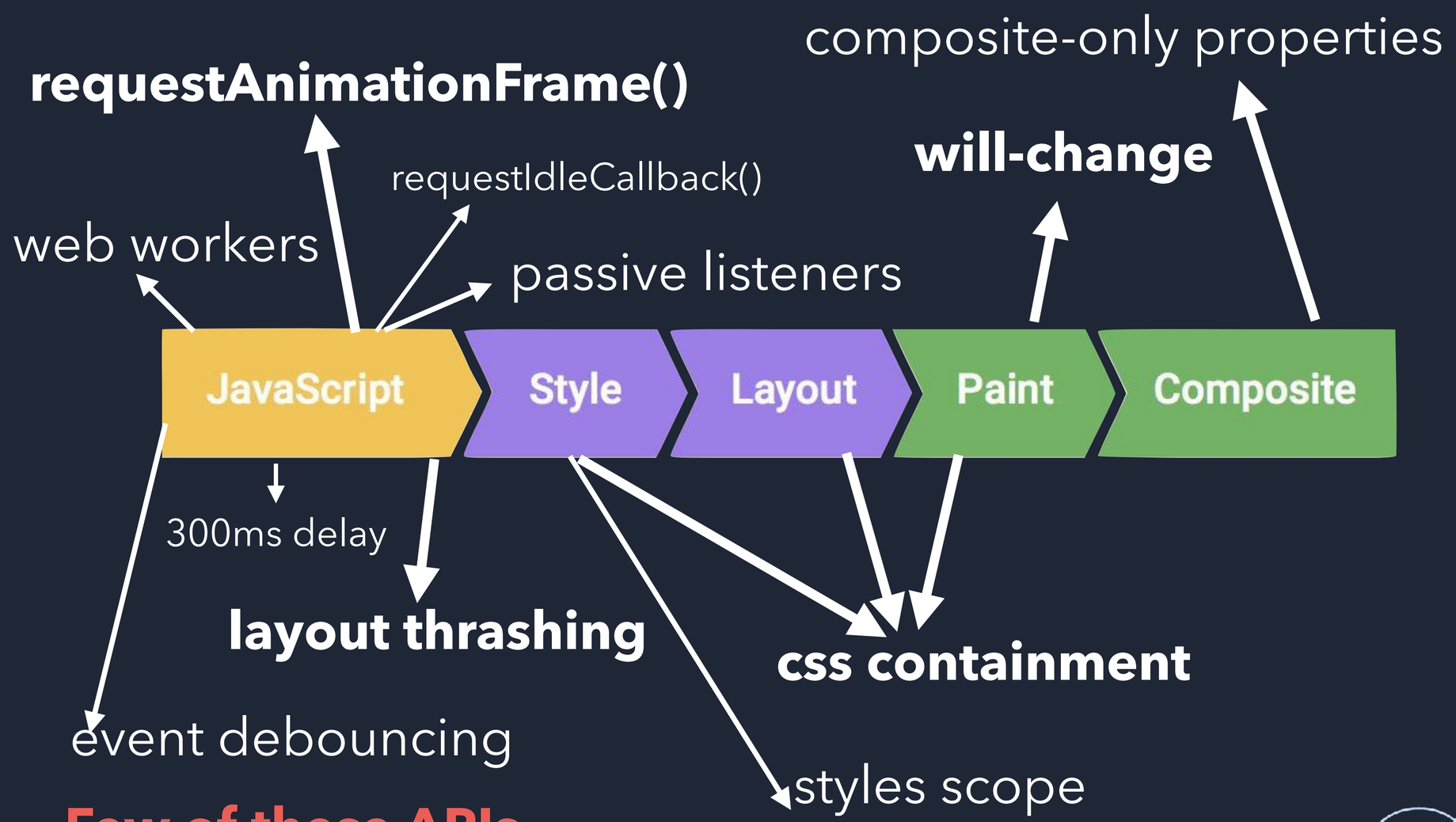
Time to Interactive (TTI)

Tooling

- Build-time per app
- Build-time per component
- NPM install time

- Incremental builds
- Ship compiled code
- Reduce complexity

Rendering



Few of these APIs available in 2013!

Event debouncing

Some events (touch/mouse) can be dispatched a higher rate than the refresh rate of the screen

Duplicated work per frame

Event debouncing

perf(item): improves performance of sliding item #9005

Merged brandyscarney merged 1 commit into `ionic-team:master` from `manucorporat:perf-sliding-item` on Nov 4, 2016

Conversation 2 | Commits 1 | Files changed 2

manucorporat commented on Nov 3, 2016 • edited — Member

Same technique used in this PR: driftyco#8986

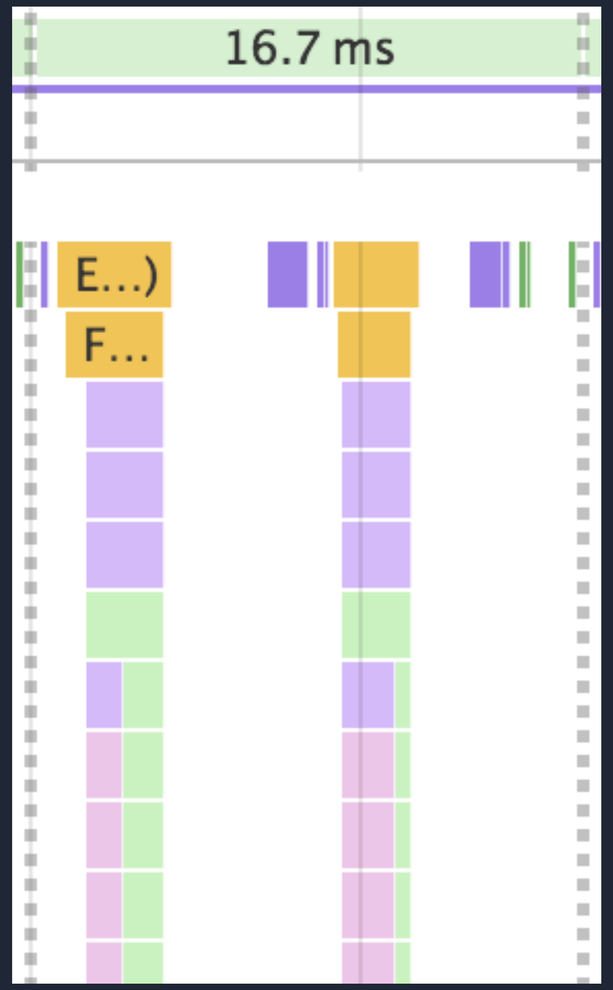
This PR should improves dramatically the performance in low end devices:

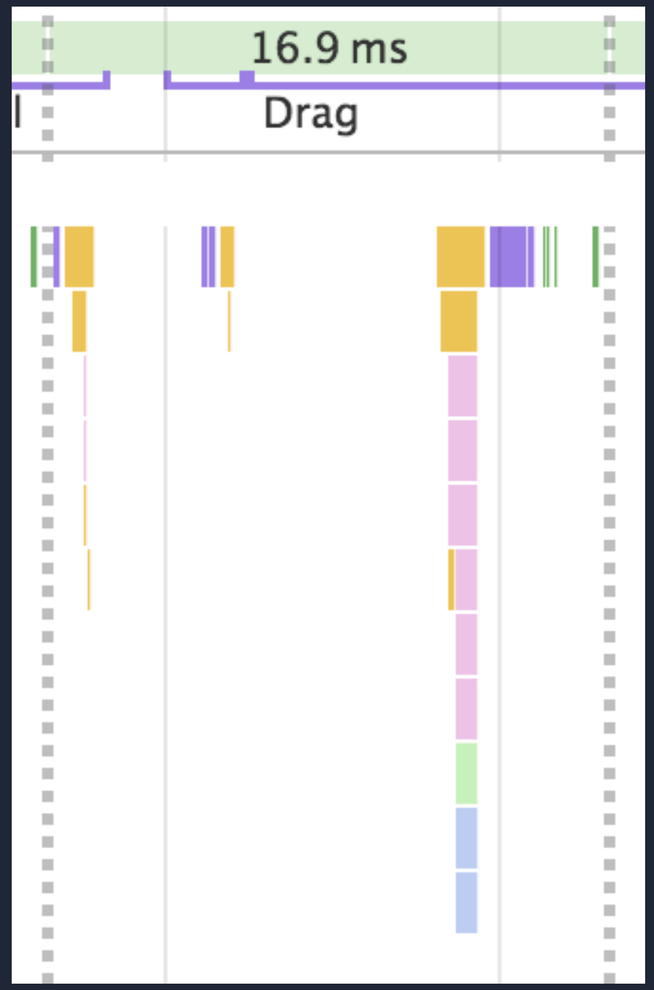
Review

No revi

Assign

Event debouncing





will-change

Indicates to the browser certain properties will **change frequently** (ex: scrolling, animations, gestures)

Browser promotes element to own layer

Smoother animations with less CPU usage (though possibly higher RAM usage)

will-change

```
will-change: auto;
will-change: scroll-position;
will-change: contents;
will-change: transform;
will-change: opacity;
will-change: left, top;
```

Fallback:

```
transform: translateZ(0)
```

requestAnimationFrame()

Syncs the execution of a function with the refresh rate of the screen

Updates the DOM elements once at the right moment

Smooth animation without jank

requestAnimationFrame()

```
function animate() {
  requestAnimationFrame(animate)

  myEl.style.transform = `translateX(${x}px)`;

  x++;
}

requestAnimationFrame(animate)
```

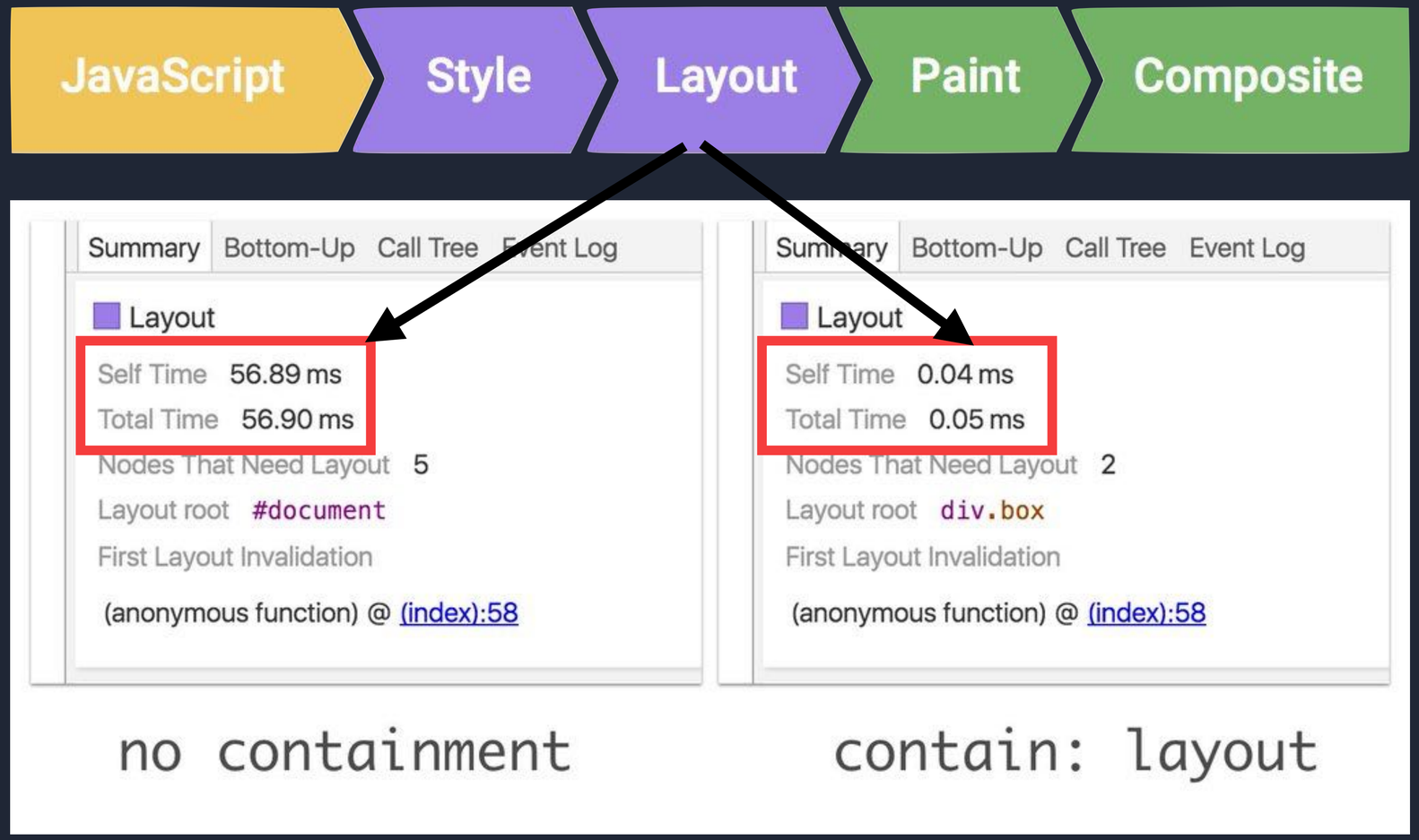
CSS containment

Isolate elements from the rest of the app

Browser optimizes, limiting recalc paint/layout/size/style to sub-tree

Fast component updates

CSS containment



layout x1425 faster!!

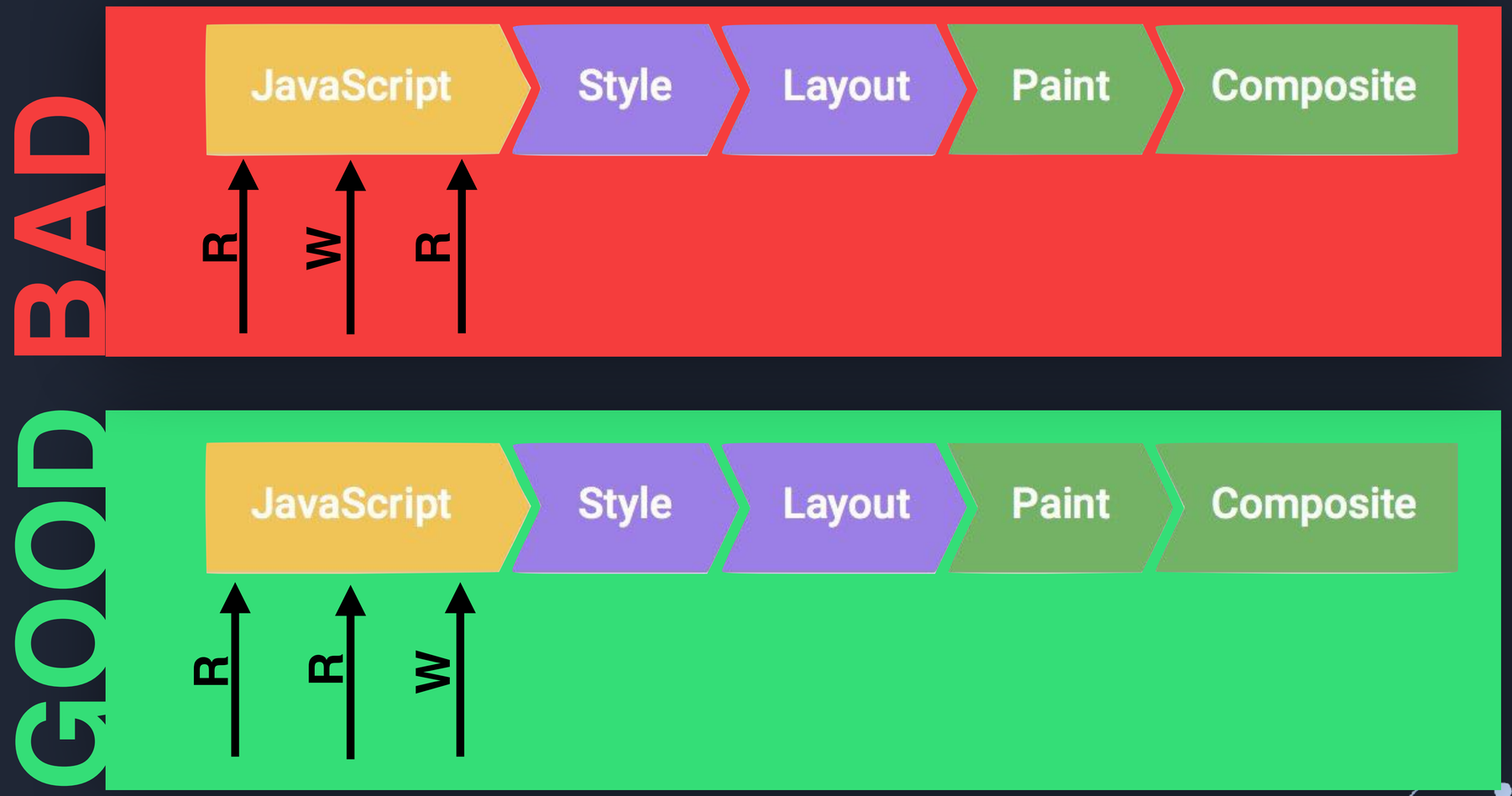
CSS containment

```
ion-modal {
  position: absolute;
  top: 0;
  left: 0;

  display: block;
  visibility: inherit !important;

  width: 100%;
  height: 100%;

  contain: strict;
}
```

```
contain: none | strict | content | [ size || layout || style || paint ]
```

Layout Thrashing

R: read **W: write**



Layout Thrashing

```
} else {
  // content does not have a view controller
  _dom.read(this._readDimensions.bind(this));
  _dom.write(this._writeDimensions.bind(this));
}
```

```
// at this point, this fn was called from within another
// requestAnimationFrame, so the next dom reads/writes within the next fr
// wait a frame before trying to read and calculate the dimensions

// ******** DOM READ ****************
this._dom.read(() => initReadNodes(this._plt, nodes, cells, data));

this._dom.write(() => {
  // update the bound context for each node
  updateNodeContext(nodes, cells, data);
```

Further reading

- **Will-change:** https://developer.mozilla.org/en/docs/Web/CSS/will-change
- **requestAnimationFrame():** https://developer.mozilla.org/en-US/docs/Web/API/window/requestAnimationFrame
- **CSS containment:** https://developers.google.com/web/updates/2016/06/css-containment
- **Layout Thrashing:** https://developers.google.com/web/fundamentals/performance/rendering/avoid-large-complex-layouts-and-layout-thrashing

ionic

Thanks!

Slides available online:

https://github.com/manucorporat/perf-apis2

Twitter: @manucorporat
Github: @manucorporat
Email: manuel@ionic.io