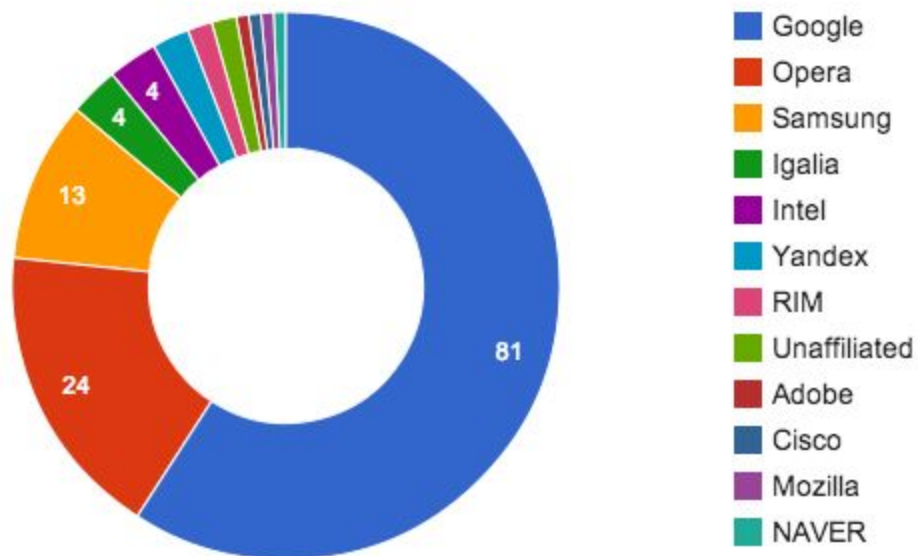


# **BlinkOn2**

eseidel & jochen

**Welcome**

## Affiliations

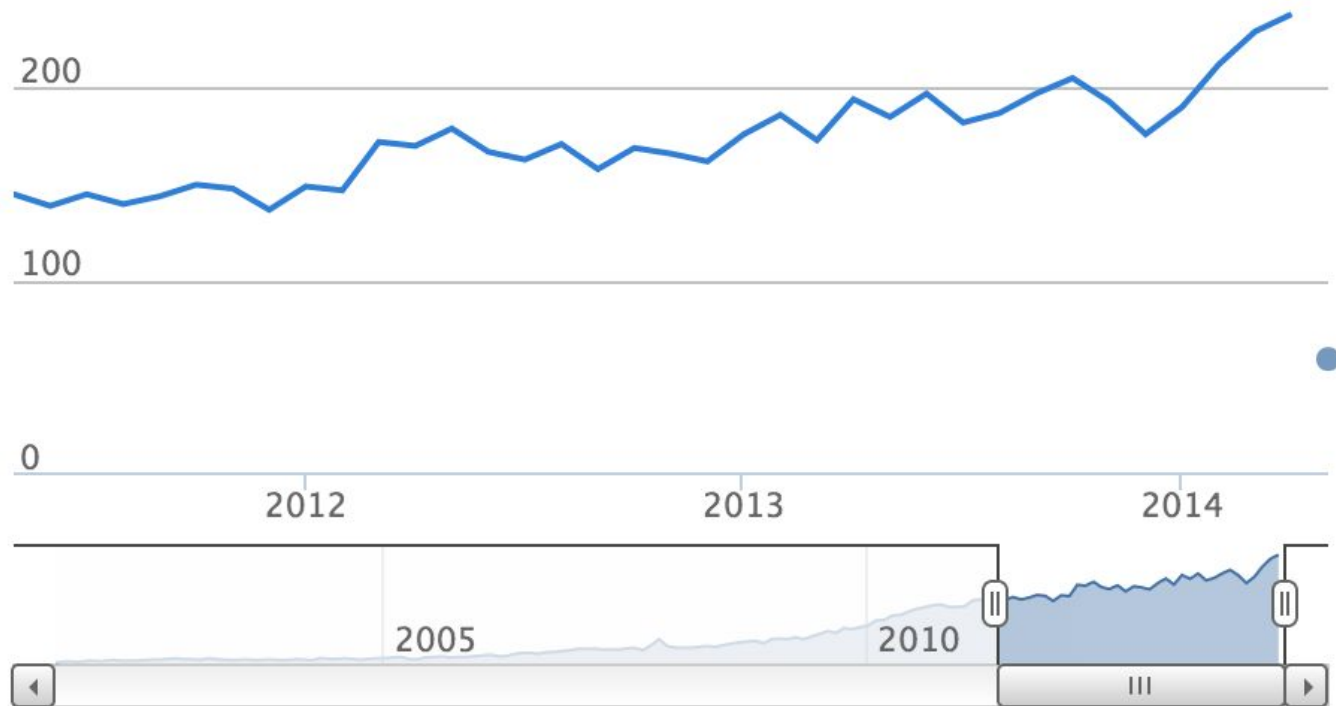


# Overview

**Last 6 months**

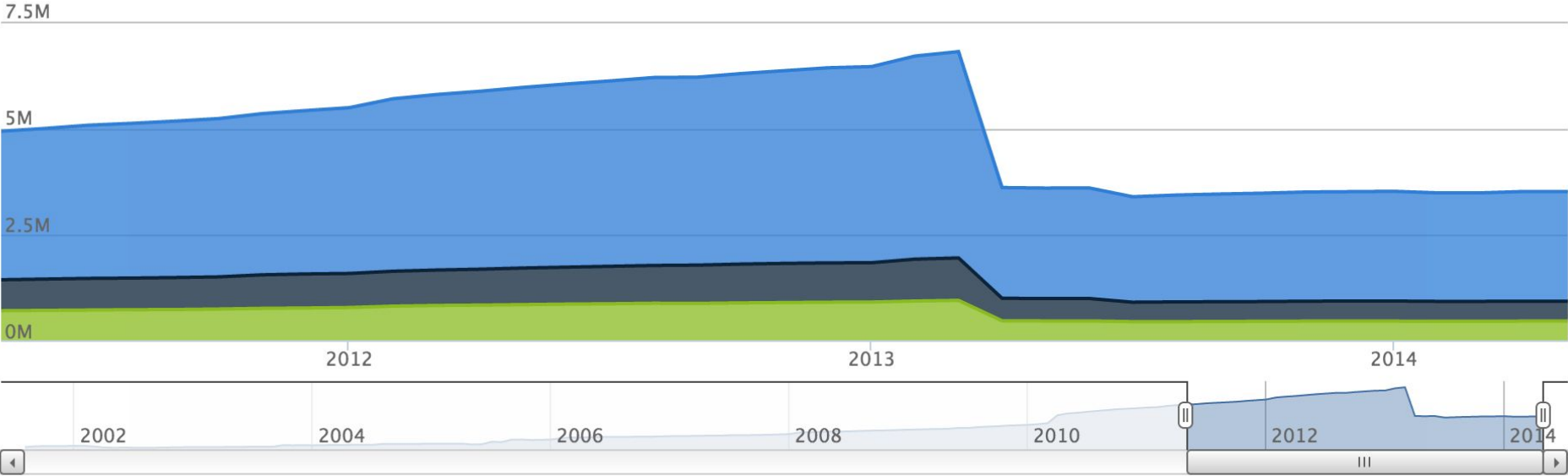
# Number of Contributors

Zoom 1yr 3yr 5yr 10yr All



# Code, Comments and Blank Lines

Zoom 1yr 3yr 5yr 10yr All



# Intents

113 intents

- 61% non-Google
- 37% “Remove”
- 24% “Ship”

[bit.ly/blinkintents](https://bit.ly/blinkintents)



# Shipped

Custom Elements

Gamepad

GPU-based Filters

FastText Autosizer

Layers Panel

Targeted StyleRecalc

SVG Paint Order

CSS touch-action

CSS src-set

Speech Synthesis

Device Vibration

Many more...

# Unshipped

CSS Variables

Old Web Notifications

CSS Regions

Lots of SVG 1.1

Attempted:

showModalDialog

Attr nodes

[chromestatus.com](https://chromestatus.com)

# Near-term (last BlinkOn)

- Web components performance (better)
  - Optimize for many inline style elements (shipped)
  - Reduce footprint of Document
- Web animations (Element.animate in M36)
- Better compositor integration (shipped)
- Platform-independent fonts (soon)
- Remove vendor prefixes (better)
- API encapsulation (style, rendering, etc.) (progress)

# Medium-term (last BlinkOn)

- Layers for SVG
- Remove widget tree (progress)
- Out of process iframes (progress)
  - Move history to embedder
  - Move CORS to embedder
- Unified C++/JS garbage collector (progress)
- Deprecate rarely-used features
  - XSLT? NPAPI? Attr nodes? (attempted, learned)
- Link modules into a separate dylib (progress)

# Speculative (last BlinkOn)

- Moar parallelism
  - Layout, style resolution (no progress)
- JIT style resolution (under investigation)
- Fast mode (investigated)
  - Disable slow features (margin collapsing, floats)
  - Optimized code path that goes fast
- Out-of-trunk modules
  - Apps/Extensions bindings in Chromium
- Swappable components
  - Editing and/or XSLT in JavaScript? (in progress)

**Next 6 months**

# Near Term

Oilpan

Repaint After Layout

Layer Squashing

Web Animations

Blink Scheduling

Forced Compositing

Merge with Chromium

HTML Imports

APIs in Rendering

New SkRecord

Device APIs

Juice

# Medium Term

GPU Acceleration

RemoteFrame

Service Worker

New-Multicolumn

Tracing-Inspector

Page Transitions

Bindings Split

DirectWrite

Mojo

GN



# Speculative

JIT Style Resolution

Incremental Style Resolve & Layout

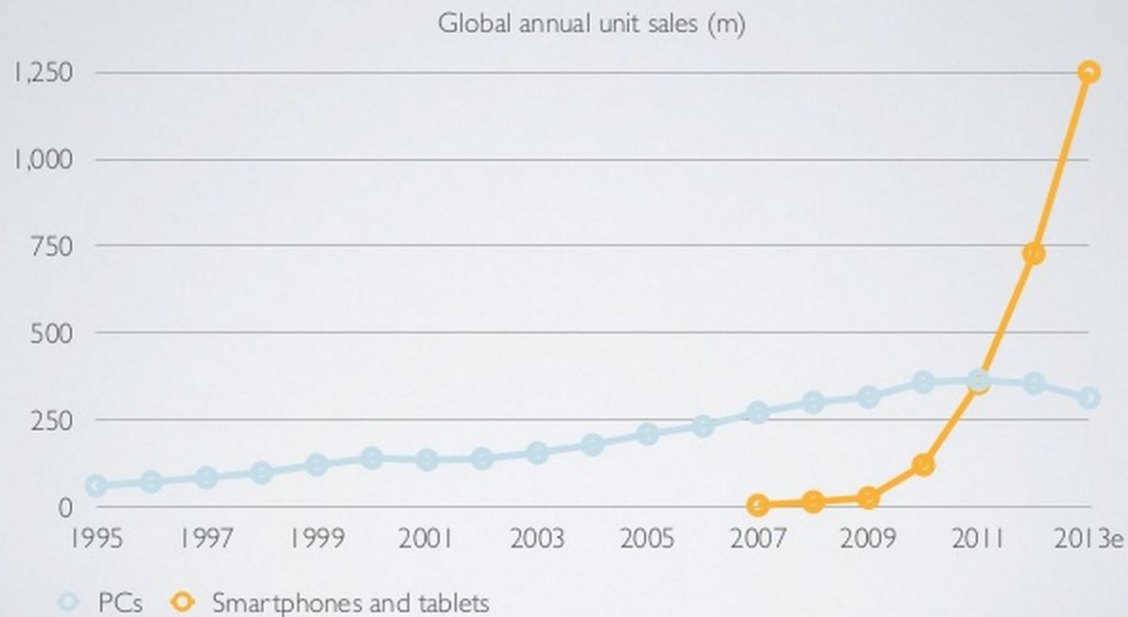
Blink in JavaScript

Moving XSLT etc. out of Blink

**Why?**

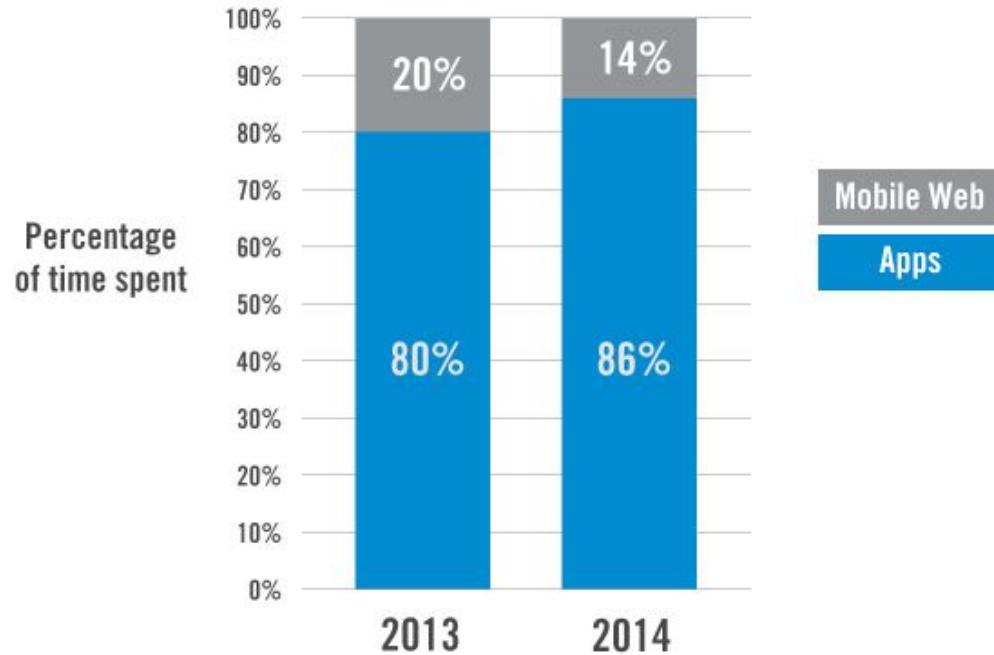
# Mobile

# The future is mobile



Source: “mobile is eating the world”, Benedict Evans Nov ‘13

## Apps Continue to Dominate the Mobile Web



Source: “the decline of the mobile web”, Chris Dixon April ‘14

“Mobile eats the world”

+

“The decline of the mobile web”

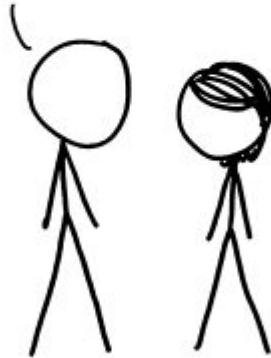
=

Web becomes irrelevant

**Must be relevant**

INSTALLING THINGS HAS  
GOTTEN SO FAST AND PAINLESS.

WHY NOT SKIP IT ENTIRELY,  
AND MAKE A PHONE THAT HAS  
EVERY APP "INSTALLED" ALREADY  
AND JUST DOWNLOADS AND RUNS  
THEM ON THE FLY?



I FELT PRETTY CLEVER UNTIL I  
REALIZED I'D INVENTED WEBPAGES.



# Challenges

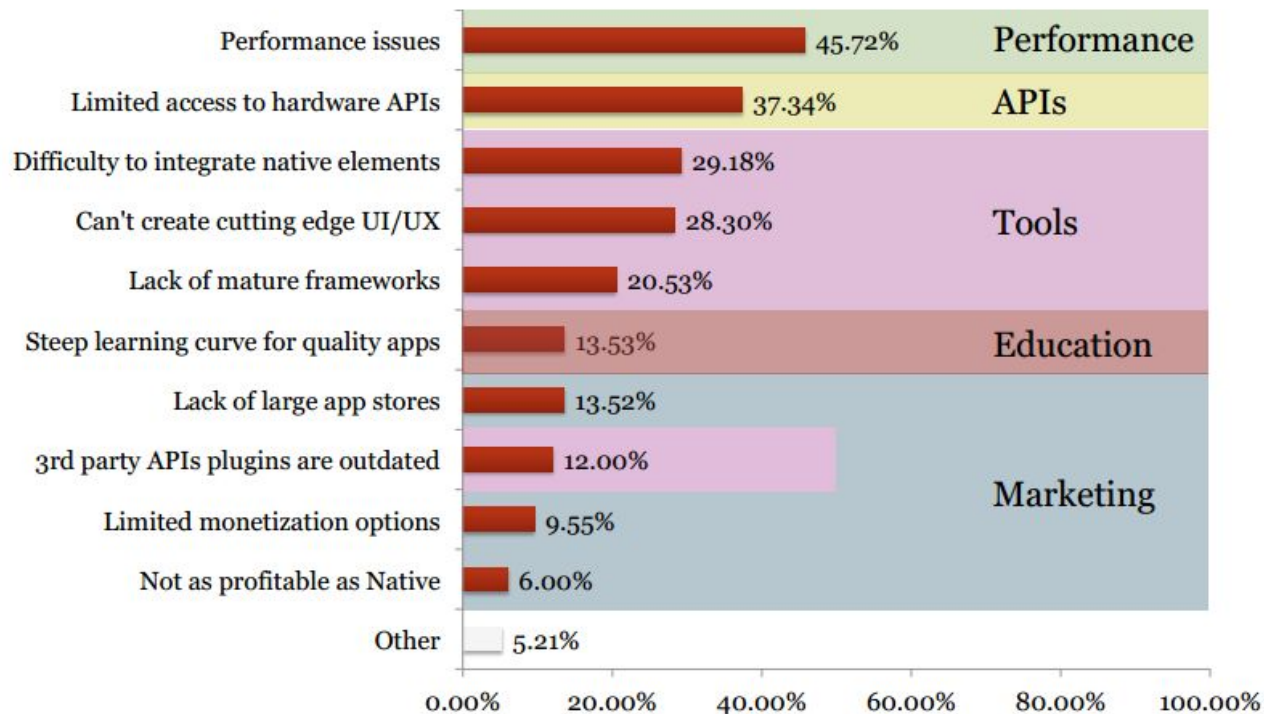
# Mobile

How does the Web stay relevant?

How does Blink?

Are we doing enough?

# WHAT STOPS MOBILE DEVELOPERS FROM USING HTML5?



60hz, Speed Team, Juice  
Device APIs, ServiceWorker

WebComponents  
WebAnimation  
[polymer-project.org](http://polymer-project.org)

[developer.google.com/web](http://developer.google.com/web)

Source: Vision Mobile Blog

# **Mobile web is in its infancy**

How can we push the mobile web forward?

# **Move so fast**

Are we leading the web?

How do we not steer off a cliff?

# Leaving the past

Enterprise?

If we don't move fast, someone else will.

# Complexity

How can we make it easier to hack on blink?  
What does Blink look like as part of  
Chromium?

**How can I help?**

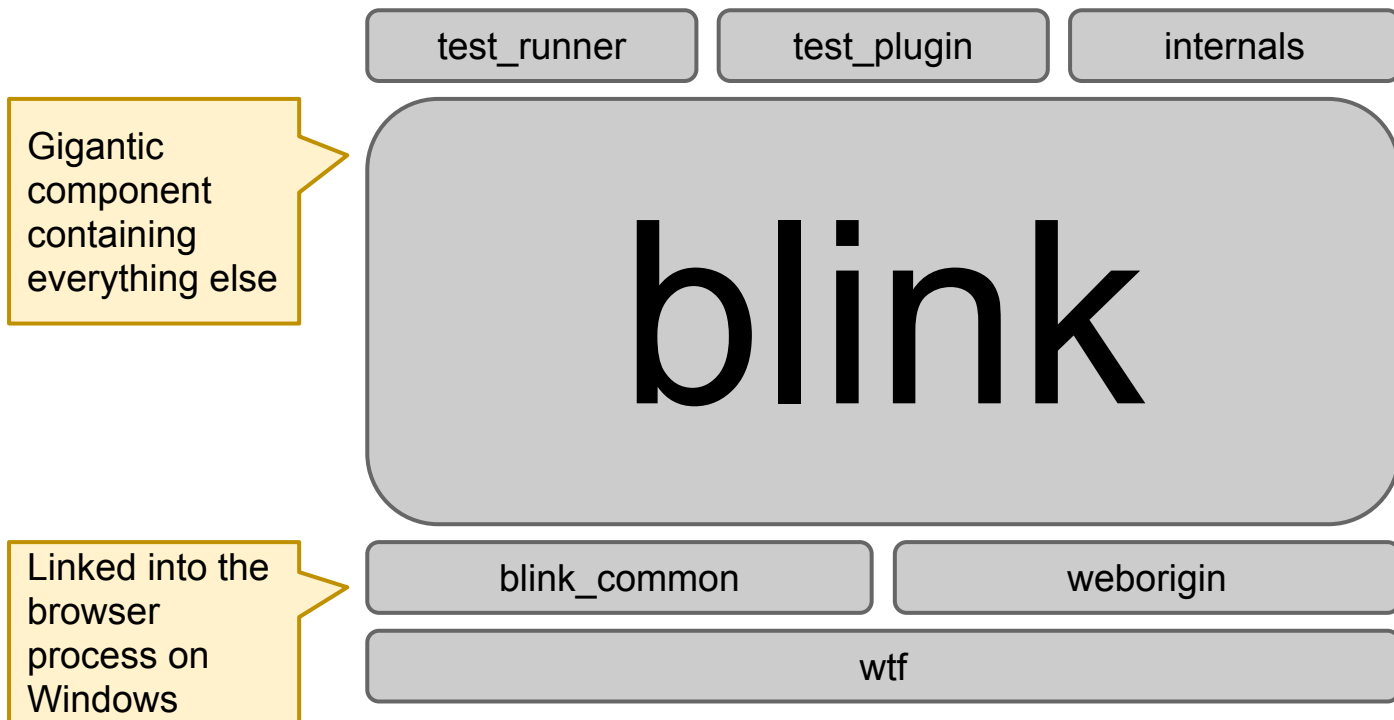


# Performance

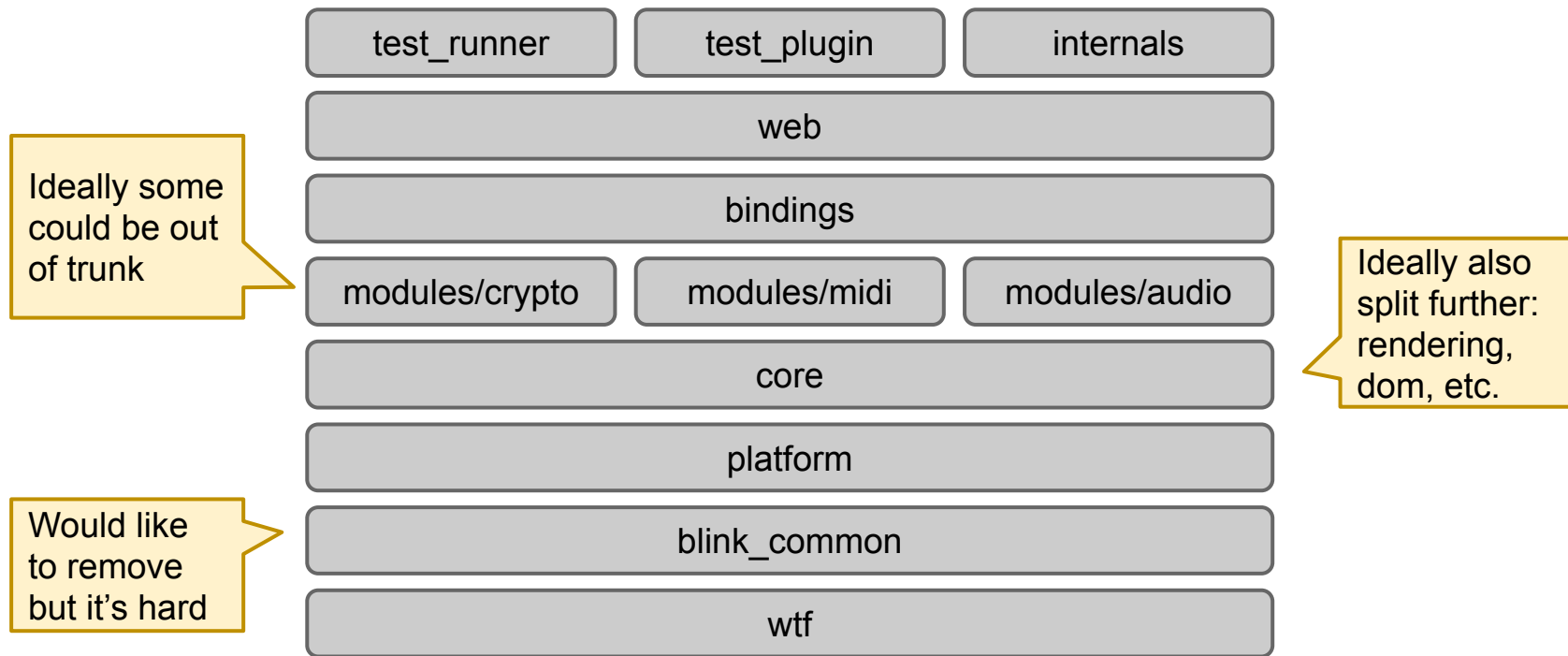
- Nat's talk
- about:tracing
- hotlist:Jank
- Performance=Power
- Performance=Memory

# Modularity

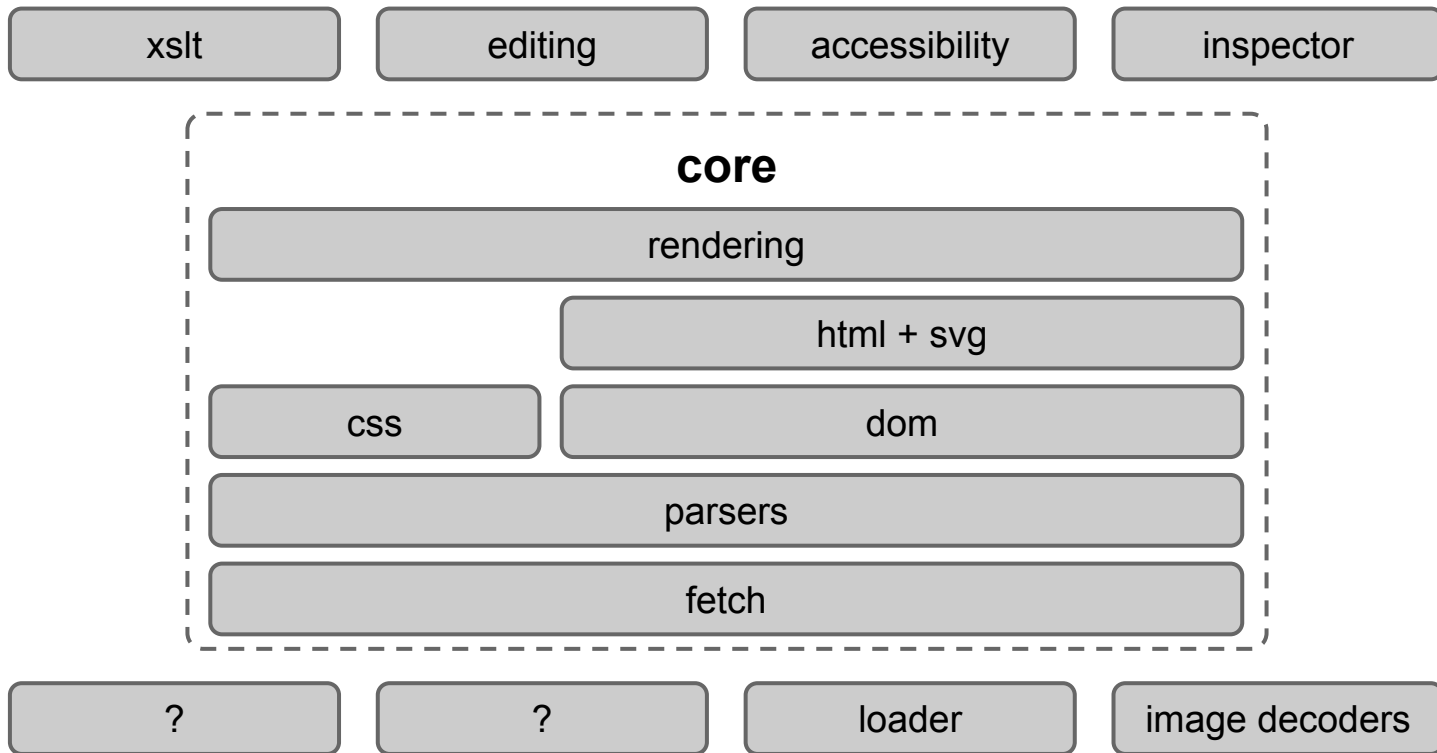
# Dependencies (Today - last BlinkOn)



# Dependencies (Soon - last BlinkOn)



# Dependencies (Speculative)



# Dependency Work

- core/DEPS is free from modules
- bindings split in progress (TOK)
- 28 modules (up from 17 one year ago)

# Review

# **Growth**



# Mobile



# **BlinkOn3**

Nov '14, California

# **BlinkOn4**

Spring '15, Tokyo/Sydney?

# Thank you

eseidel@, jochen@

# Questions?

eseidel@, jochen@







# Extra Slides

Updated Shipping Guidelines  
Removal of non-mobile features

**Performance is a Blocker**

*Facebook for Android 2.0 ... replaces the hybrid native/webview code with an all-native infrastructure to answer years of complaints by making the app much faster.*

# Performance Lessons

- Mobile is simply 10x slower
- Threads are not the answer
- Frames

# Thinking in Frames

60hz = 16ms

Always Animating

Never miss a frame

# Frames

`about:tracing`

CrRendererMain

Chrome\_ChildIOThread

Mes

Thr

W

Mes

T...

We

Mes

Thr

W

R

Me

Thr

W

Mes

T...

W

Mes...

Thr...

Web

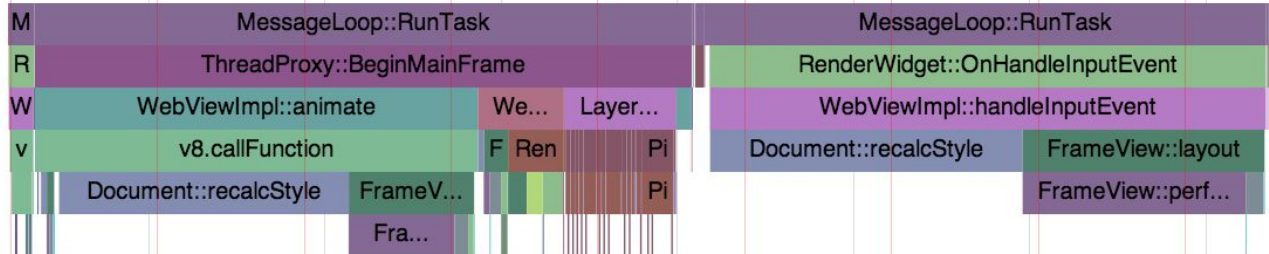
Do R

R

Do R



CrRendererMain



Chrome\_ChildIOThread

