

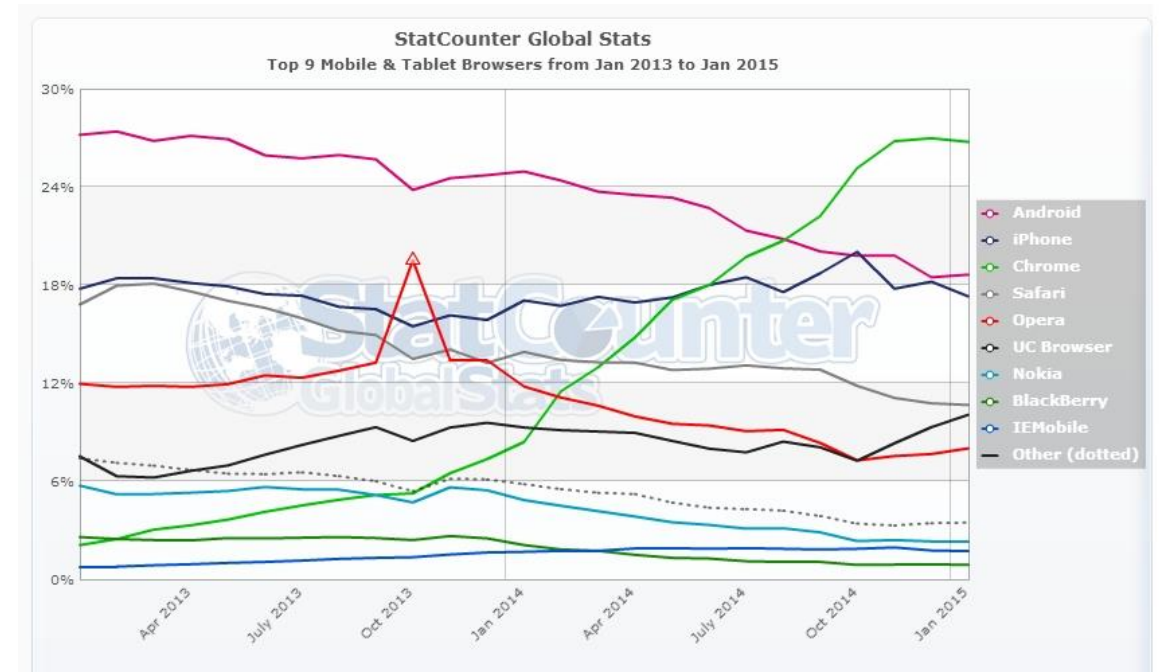
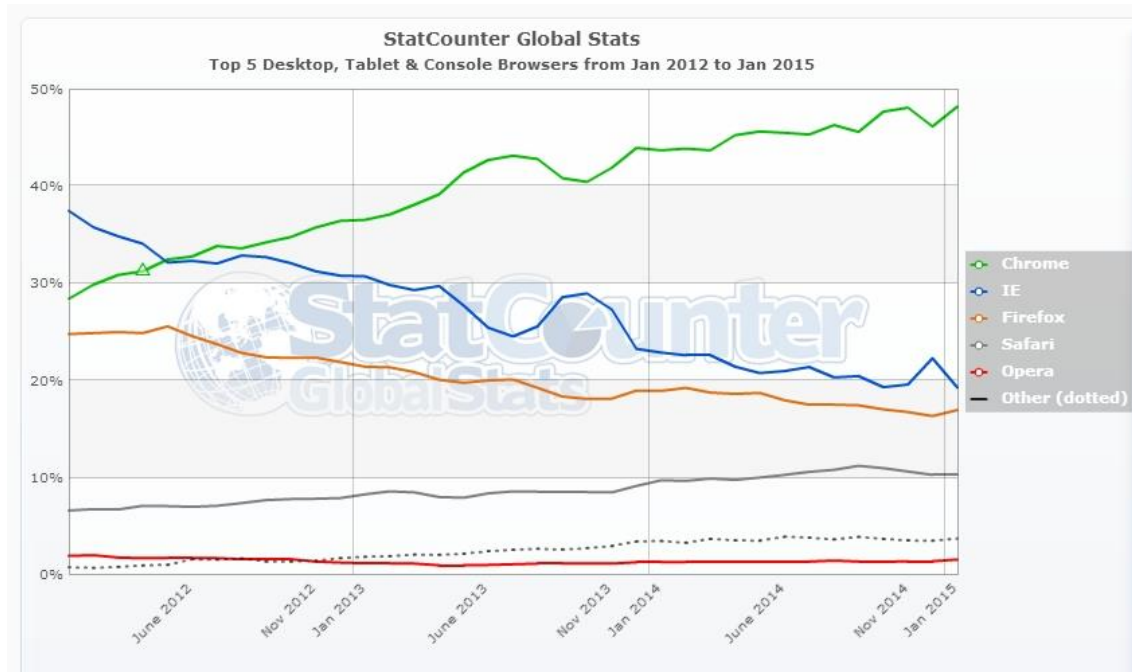


Trends in HTML5

Matt Spencer
UI & Browser Marketing Manager

Where to focus?

Chrome is the worlds leading browser - by a large margin



Chrome or Chromium, what's the difference



- Chromium is an open source browser
- Open governance
- <http://www.chromium.org/Home>
- All key development happens here
- Chrome built on Chromium
- Google added 'magic sauce'
 - Single signon
 - Global history
 - ...

The flavours of Chrome



Chrome Canary

- Daily release
- Experimental features
- Latest developer tools



Chrome Beta

- Weekly release
- Maturing features



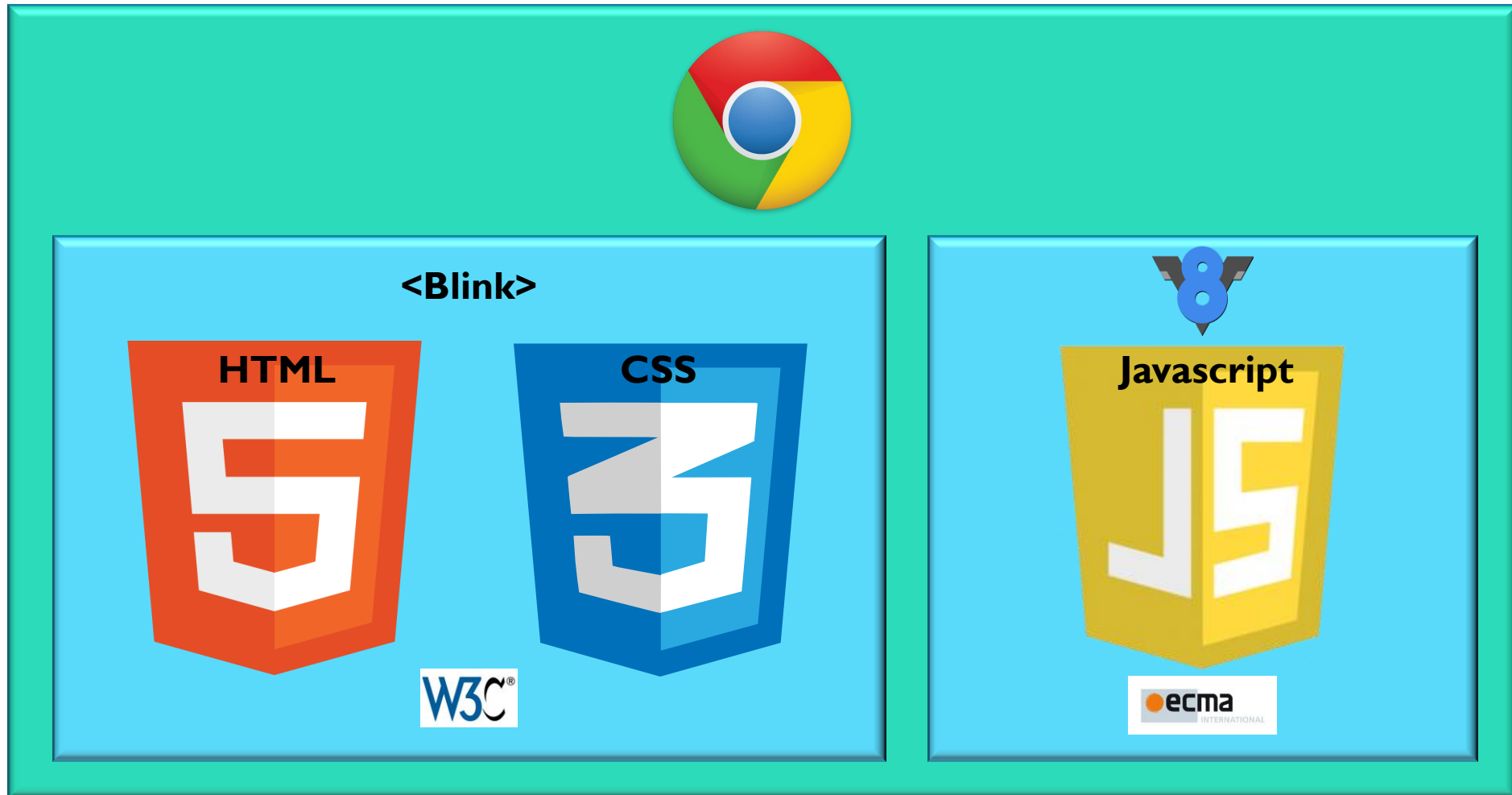
Chrome

- 6 weekly release
- Mature features



Chrome architecture

The 60,000ft view



HTML5 Ratified finally!

After 7 years of development, the HTML5 specification was ratified on 28th October '14

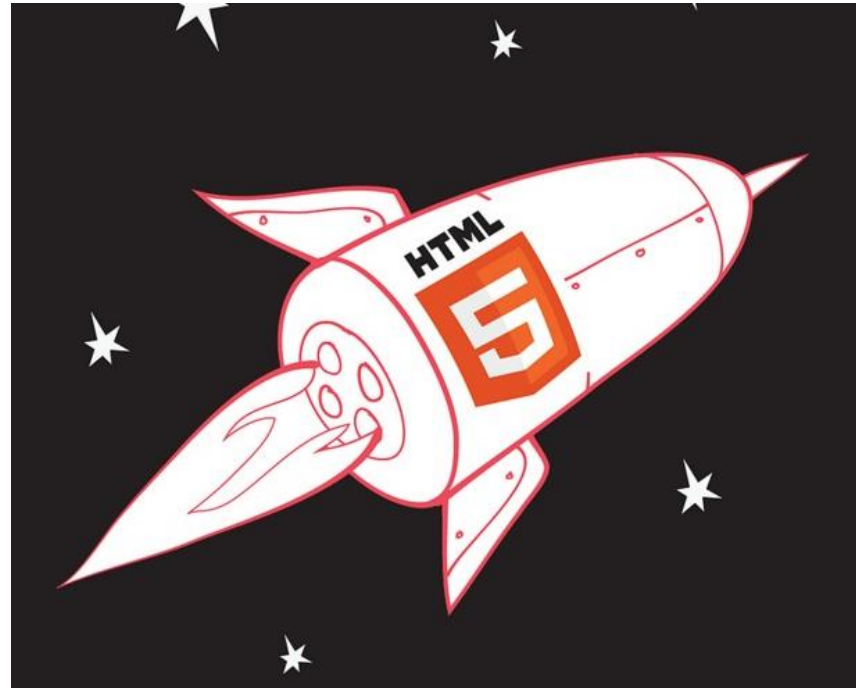
<source> **<video>**

<svg>

<video>

<article>

<progress> **<footer>**



<audio>

<track>

<nav>

<summary>

<canvas>

<embed> **<article>**

Over 22 new Semantic element, 3 new Form elements, 29 new input types
2 new Graphics Elements, 5 new Media elements, plus lots more!

HTML5 Ratified

But the platform has not finished evolving



Web Facing changes

Web Components

<http://webcomponents.org/>

- A core problem of HTML5 is Modularity
 - I can't implement a 'tile' and import the functionality into my page
 - I can't control how the main page styling affects my 'tile'
- Luckily web-components comes to save the day
- HTML Imports (<http://www.w3.org/TR/html-imports/>)
 - Allows me to import HTML into my document
- Templates (<http://www.w3.org/TR/html-templates/>)
 - Allows definition of template HTML content
- Custom Elements (<http://www.w3.org/TR/custom-elements/>)
 - Allows me to define a custom <my-element> style tag
- Shadow DOM (<http://www.w3.org/TR/shadow-dom/>)
 - Allows hiding content and styling from the rest of the page



Web Components - in action

Using the Polymer Polyfill for Web Components

index.html

... snip boilerplate ...

```
<link rel="import" href="my-comp.html">
```

```
<my-comp></my-comp>
```

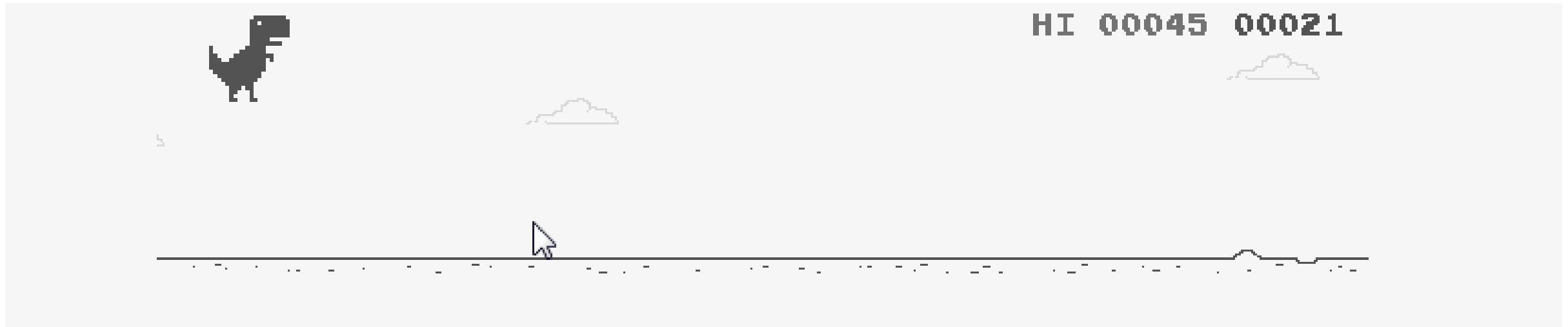
my-comp.html

```
<polymer-element name='my-comp'>  
  <style>  
    <!-- component specific styling -->  
  </style>  
  <template>  
    <!-- component template -->  
  </template>  
  <script>  
    <!-- components scripts (optional) -->  
  </script>  
</polymer-element>
```

Service Workers

<https://github.com/slightlyoff/ServiceWorker>

- Creating HTML apps has an inherent problem
 - “How can I function when I am offline?”



- This does not create a compelling offline offering - it does not act like a native app

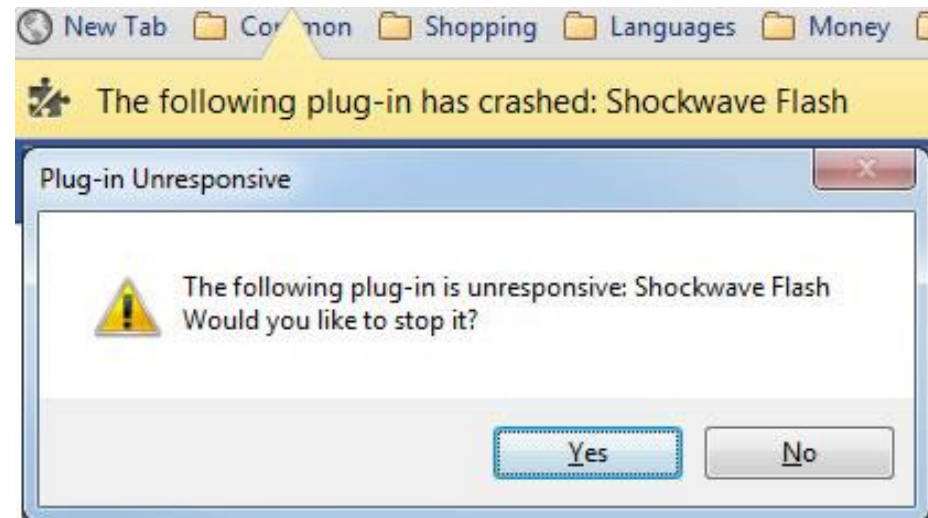
Service Workers - in action

<https://github.com/w3c-webmob/ServiceWorkersDemos>

- Service workers allow you to define a script that
 - Runs off the main browser thread
 - Has a lifecycle that is independent of the page that initiates it
 - Can intercept any HTTP request for the domain that served it
- Service Workers can make use of the following new technology
 - Caches (https://slightlyoff.github.io/ServiceWorker/spec/service_worker/#cache-objects)
 - Push Notifications (<http://w3c.github.io/push-api/>)
 - Background Sync (<https://github.com/slightlyoff/BackgroundSync>)
 - Geofencing (<https://github.com/slightlyoff/Geofencing>)
- By using these technologies, HTML app developers can create a truly native experience
<https://github.com/w3c-webmob/ServiceWorkersDemos>

No more Browser Plugins!

- Browser plugins - Java, Flash, Silverlight, ... have always been a weak attack surface
- All browser vendors have stated that they will deprecate support for plugins



Unbundled Android WebView

- WebView is no longer a part of the base Android Platform
- Can be updated independently of the OS
- Chrome, WebView and Blink codebases are being unified
 - WebView will be released on the same 6 week cadence as Chrome
 - WebView will pull in all the new API's
 - WebView will pull in all the performance enhancements
- The net result is a constantly improving, evolving mobile Web Application platform

Internal changes to Blink

Slimming Paint

<http://www.chromium.org/blink/slimming-paint>

- Goal

- Refactor and improve the rendering pipeline in Chrome

- Benefits

- Streamline Chrome -> Skia -> Ganesh -> GLES integration
- Move layer creation into the compositor, this allows GPU aware layer creation to be made
- Treat updates as a stream of 'delta's' to the scene graph, allows for selective updates

- Why of Interest

- The work is showing promising results so far
- Will allow for Mali GPU efficient paths to be created and optimised

OilPan

<http://www.chromium.org/blink/blink-gc>

- Goal

- Improve the memory management and garbage collection within Chrome

- Benefits

- Compile time enforcement of memory constraints
- Time sliced Garbage Collection - allow pre-empting garbage collection when needed
- Should improve responsiveness, especially on mobile platforms

- Why of Interest

- Will have a positive impact on the majority of real world use-cases
- Easier to control the usage of memory across the web platform

Cool tech on the horizon

WebVR

<https://github.com/MozVR/webvr-spec>

- Goal
 - Create a VR capable platform for the web
- Benefits
 - Target modern VR capable devices directly from the browser
 - Allows true cross platform VR development
- Why of interest
 - Streamlining of the web platform to reduce input latency
 - This will increase the browsers performance for all content - even non WebVR
 - Allows direct access to GPU acceleration, much like WebGL

WebGL 2

<https://www.khronos.org/registry/webgl/specs/latest/2.0/>

- Goal

- Bring modern Khronos API's to the web platform

- Benefits

- Allows OpenGL ES 3.0 content to be created for the web
- Allows Compute Shaders plus other aspects of GLES 3.1 to be exposed through extensions
- Will allow easier transition of premium game content to be translated over to the web

- Why of interest

- Should allow application engineers to create more efficient GLES based content for the web
- Should result in better performance and lower power consumption on modern GPUs

HTTP/2

<https://tools.ietf.org/html/rfc7540>

- Goal

- Increase the speed and security of website access

- Benefits

- Servers can 'push' content to clients, reducing the number of connection requests
- Allow multiplexing of requests and responses to minimise client blocking on requests
- Increased security - Firefox and Chrome will only support HTTP/2 over https:// connection

- Why of interest

- Reduced network chatter will increase mobile performance
- Increased security will be able to make use of ARM crypto engine
- Will help to make the web a more mobile friendly environment

Thank You

The trademarks featured in this presentation are registered and/or unregistered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Any other marks featured may be trademarks of their respective owners