

# Surviving the Zombie Apocalypse of Connected Devices

(HTML Hypermedia APIs and Adaptive Web Design)

Gustaf Nilsson Kotte

# About me



@gustaf\_nk

# Twitter mentions



**Jon Moore**

@jon\_moore

Awesome article on a combined mobile/desktop/API strategy using [#hypermedia](#) APIs and HTML: [jayway.com/2012/08/01/com...](http://jayway.com/2012/08/01/com...)

← Reply ↺ Retweet ★ Favorite

9

RETWEETS

9

FAVORITES



2:48 PM - 4 Aug 12 · Embed this Tweet



**Mike Amundsen**

@mamund

read this. let it sink in. then read it again.  
very smart stuff here. [bit.ly/NdxljV](http://bit.ly/NdxljV) #HTTP  
#API #Web #LOD #Hypermedia /via  
@gustaf\_nk

← Reply ↺ Retweet ★ Favorite

13

RETWEETS

10

FAVORITES



8:59 PM - 5 Aug 12 · Embed this Tweet



**Hypermedia APIs** @hypermediaapis

4 Aug

Combining HTML Hypermedia APIs and Adaptive Design:  
[jayway.com/2012/08/01/com...](http://jayway.com/2012/08/01/com...)

↺ Retweeted by Roy T. Fielding

Expand



# Two problems

How to scale development of apps?

How to make our web work on every browser?

# Two problems

How to scale development of apps? => HTML Hypermedia APIs

How to make our web work on every browser? => AWD

HTML Hypermedia APIs + AWD => Perfect combo!



[bradfrostweb.com](http://bradfrostweb.com)





[bradfrostweb.com](http://bradfrostweb.com)



# THIS IS THE WEB.



A collage of various electronic devices and a car, all displaying the word 'WEB' on their screens, surrounded by numerous large orange question marks, symbolizing the integration of the internet into everyday life. The devices include a small orange car, a desktop monitor, several laptops, a tablet, a smartphone, a digital camera, a printer, a game console, a refrigerator, and a small handheld device. Each device's screen shows the word 'WEB' in white capital letters on an orange background. The background is a light beige color, and the entire scene is framed by many large, stylized orange question marks of varying sizes, creating a sense of inquiry and uncertainty about the future of technology and the internet.

# In general, growing number of...

- Connected physical devices
- Types of devices
- Platforms
- Device models

# Technology evolves in all directions





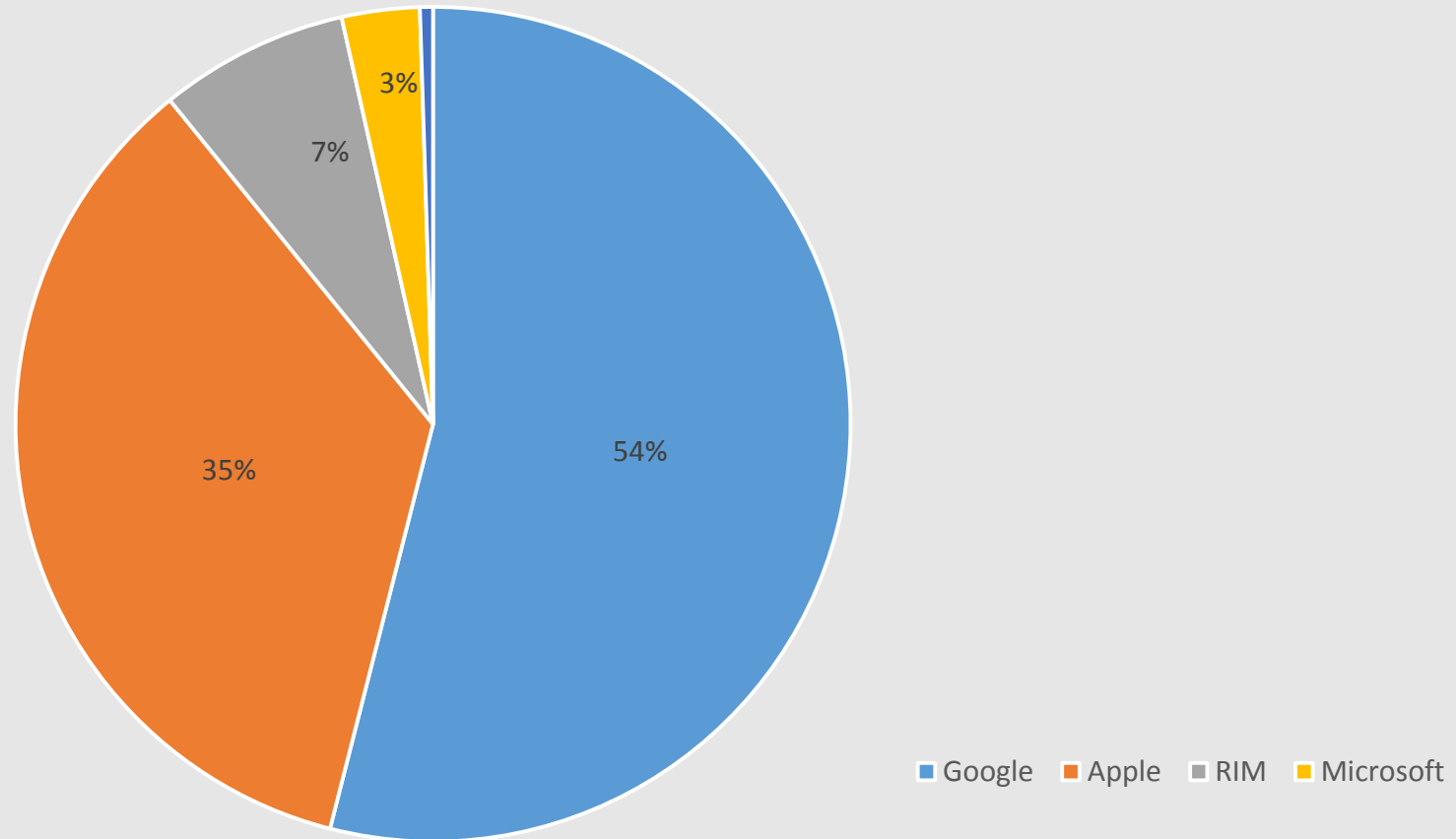


Raspberry Pi

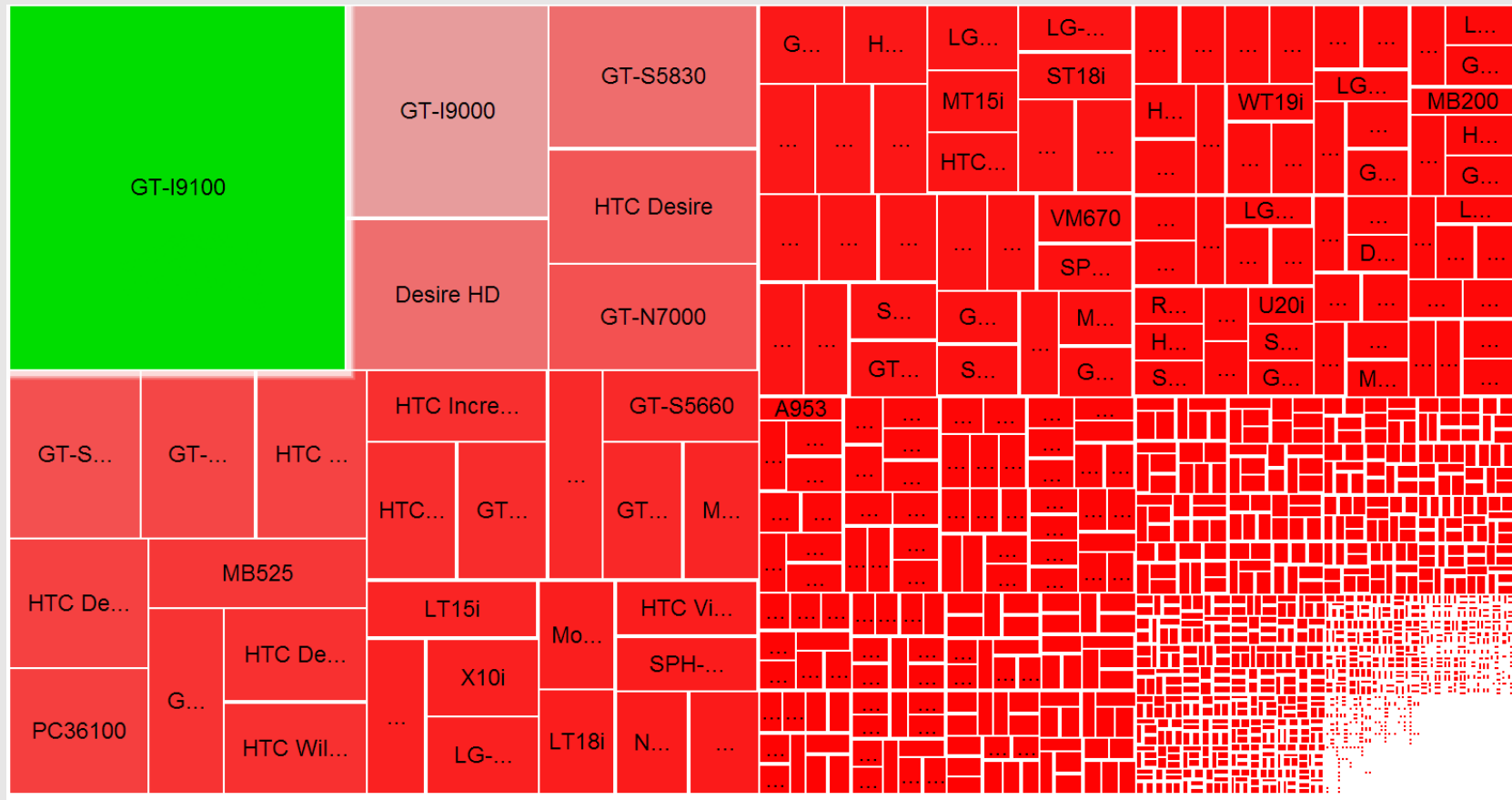




# Smartphone market share, January 2013



# Android fragmentation



**Zombies!**





# Zombie properties

A close-up photograph of a zombie's face, showing a greyish, decaying skin texture and a wide, toothy grin. A semi-transparent grey rectangular box is overlaid on the left side of the image, containing the title and a list of properties.

Move slowly

Hungry

Angry

Lots of them

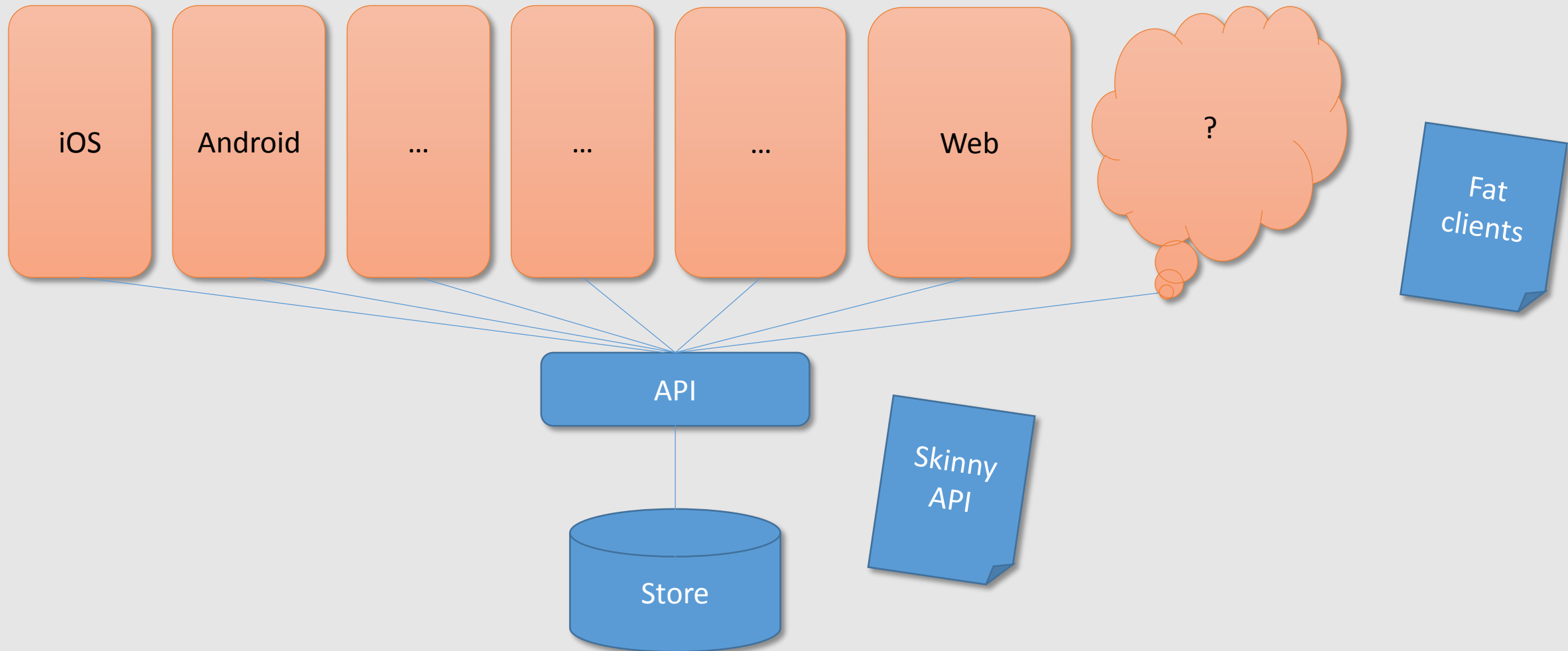
# Two problems

How to scale development of apps? => HTML Hypermedia APIs

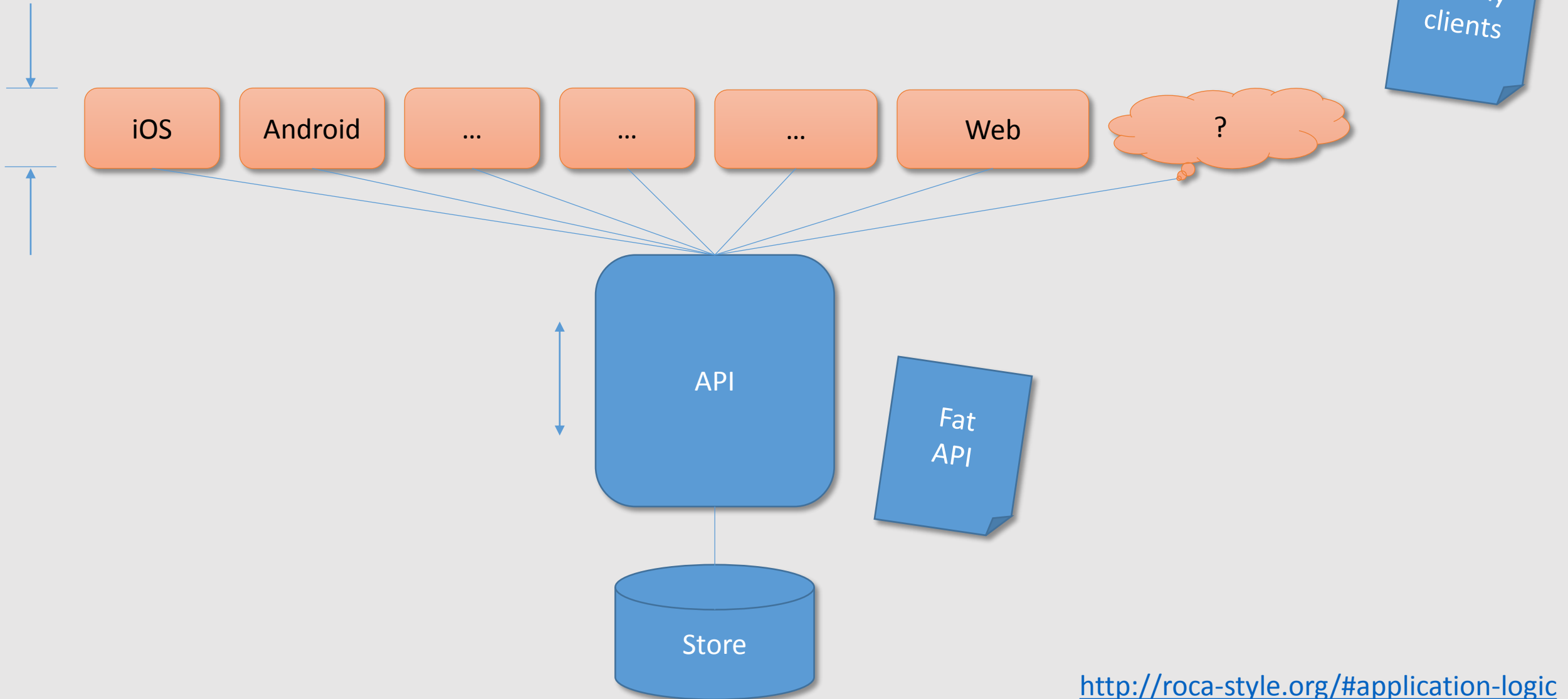
How to make our web work on every browser? => AWD

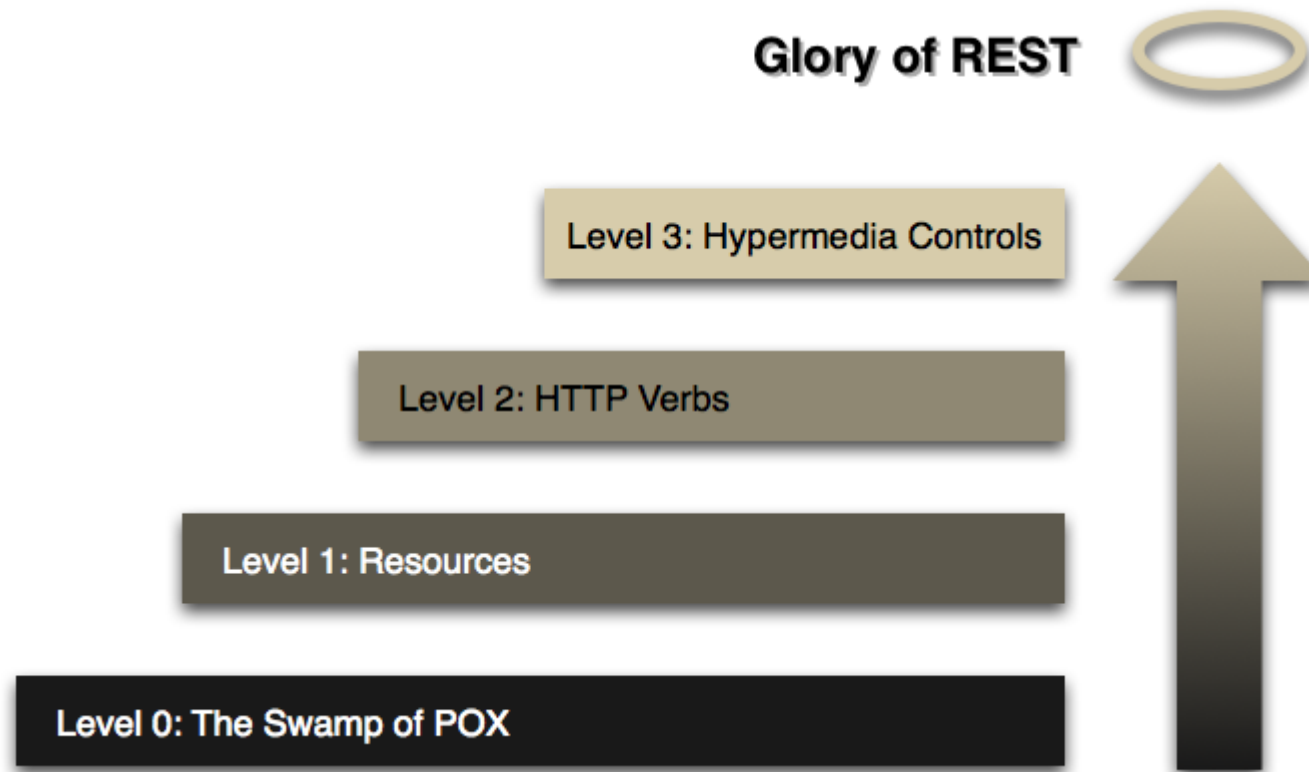
HTML Hypermedia APIs + AWD => Perfect combo!

# Why so fat clients?



# Fat API is better!





# Hypermedia APIs

# Use HTML for Hypermedia APIs

Jon Moore



<http://codeartisan.blogspot.com/2012/07/using-html-as-media-type-for-your-api.html>

# Use HTML for Hypermedia APIs

Hypermedia controls

Jon Moore



<http://codeartisan.blogspot.com/2012/07/using-html-as-media-type-for-your-api.html>



# Hypermedia control, examples

## Support for out-bound navigational links (HTTP GET)

```
<a href="http://www.example.org/search" title="view search page">Search</a>
```

## Support for non-idempotent updates (HTTP POST)

```
<form method="post" action="http://www.example.org/my-keywords"/>  
  <label>Keywords:</label>  
  <input name="keywords" type="text" value="" />  
  <input type="submit" />  
</form>
```

# Use HTML for Hypermedia APIs

Rich semantics

`<ul>` => bag

`<ol>` => list

`<dl>` => map

Jon Moore



<http://codeartisan.blogspot.com/2012/07/using-html-as-media-type-for-your-api.html>

# Use HTML for Hypermedia APIs

Extend using micro format style

Jon Moore



<http://codeartisan.blogspot.com/2012/07/using-html-as-media-type-for-your-api.html>

# Example: Microformat style

```
<div id="bugs">
  <ul class="all">
    <li>
      <span class="title">JFokus bug</span>
      <span class="description">The description of the bug</span>

      <form class="move working next"
        action="/bugs/working" method="POST" title="Move to Working">
        <input name="id" type="hidden" value="...">
        <input name="submit" type="submit" value="Move to Working">
      </form>
    </li>
  </ul>
</div>
```

# Use HTML for Hypermedia APIs

Already standardized, everybody knows it

Jon Moore



<http://codeartisan.blogspot.com/2012/07/using-html-as-media-type-for-your-api.html>

# Use HTML for Hypermedia APIs

Good tooling support

Jon Moore



<http://codeartisan.blogspot.com/2012/07/using-html-as-media-type-for-your-api.html>

# Pareto principle (80/20 rule)

“Optimize where it matters”

# Pareto principle (80/20 rule)

Primary use-cases => Native

Secondary use-cases => Render HTML response

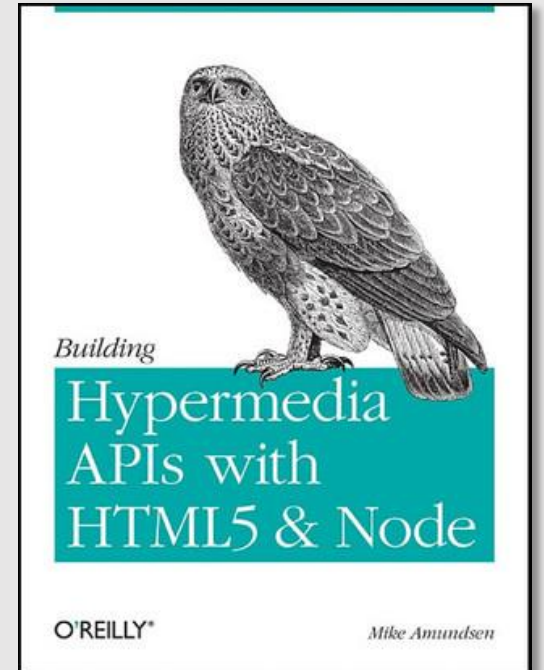


# Other media types?

- Hypermedia Controls (read/write)?
- Rich semantics?
- Extendable?
- Standardized?
- Everybody knows it?
- Good tooling support (server/clients)?

# Further exploration

”Building Hypermedia APIs with HTML5 and Node”, Mike Amundsen



<http://www.amazon.com/Building-Hypermedia-APIs-HTML5-Node/dp/1449306578>

# Two problems (recap)

How to scale development of apps? => HTML Hypermedia APIs

How to make our web work on every browser? => AWD

HTML Hypermedia APIs + AWD => Perfect combo!

# Reduce the complexity!

”Mobile first”

Simple HTML

~3kB CSS

Avoid JavaScript!



# Demo

RestBugs: Mobile web

# Another problem...

How to make a “desktop web app” with better UX?



# Progressive Enhancement



*“An escalator can never break: it can only become stairs.”*

– Mitch Hedberg

# Adaptive Web Design



<http://easy-readers.net/books/adaptive-web-design>



Aaron Gustafson



# Adaptive Web Design

Responsive Web Design

Unobtrusive JavaScript

Delayed content loading

Hijax

Bundling AJAX requests

Conditionally loaded content

Feature detection

Performance

HTML5 Navigation Timing

# Further exploration

“Progressive Enhancement 2.0”, Nicolas Zakas

“Resource-Oriented Client Architecture”

<http://www.webmonkey.com/2012/03/video-progressive-enhancement-2-0>

<http://roca-style.org>

HTML Hypermedia APIs

Mobile first

“Minimalistic HTML”



**The API and the web can be the same thing!**

# HTML Hypermedia APIs + AWD

# Highlevel view

Native apps (80/20)

Web browsers

Enhanced web (80/20)

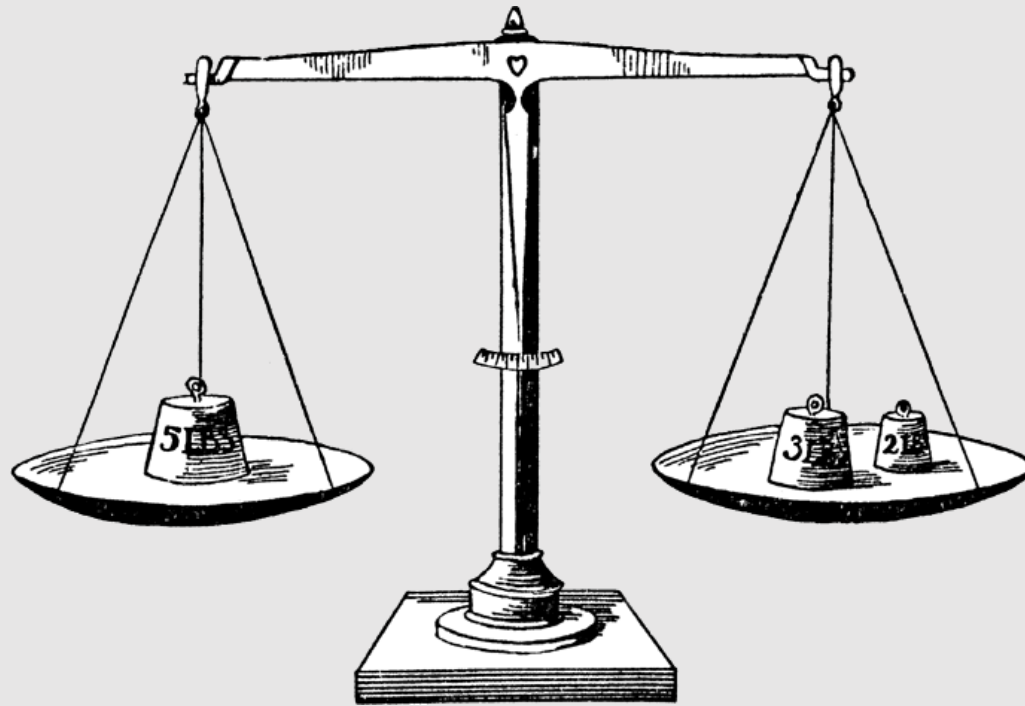
HTML Hypermedia API  
+  
AWD

*No application logic*

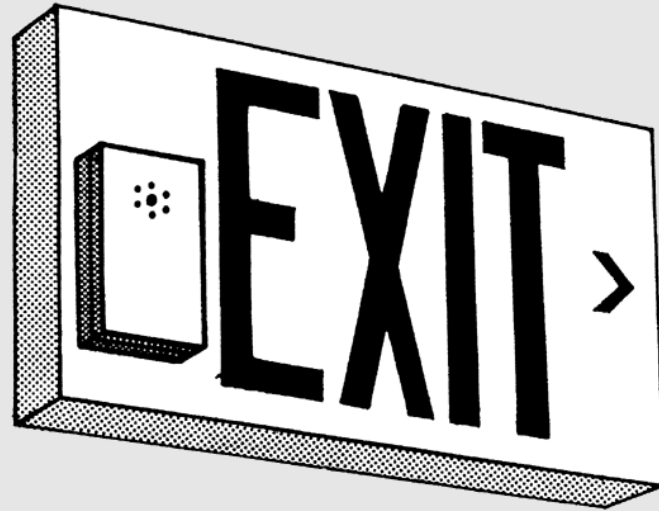
*Application logic*



# Costs < benefits?



# An exit strategy



[www.example.com](http://www.example.com)

[api.example.com](http://api.example.com)



# Demo

RestBugs: HTML Hypermedia API + AWD

# Client pseudo-code, initialization

(Inline javascript code in responses)

```
use harvey.js // polyfill for media queries
```

```
configuration "large screen":
```

```
  on:
```

```
    $.getScript("/js/client.min.js");
```

```
    $.get("/css/client.min.css", function(css) {...});
```

```
  off:
```

```
    location.reload()
```

# Client pseudo-code, initialization

```
(client.min.js)
```

```
links = $("#links a[rel!=index]");  
addForm = $("#form.new");  
load template containing angular.js markup  
clear <body>  
bootstrap angular
```

```
foreach link in links:  
    htmlResponse = GET link  
    parse htmlResponse  
    populate view model
```

```
Enable hijax for add and move forms
```

# Server pseudo-code, flow

```
(app.js)
renderWeb = function(){
  return host.contains(":9200")
};
```

```
if (renderWeb):
  redirect to index page
else:
  return 201 Created;
```

# Server pseudo-code, optimization

```
(app.js)
renderWeb = function(){
  return host.contains(":9200")
};
```

```
(server template)
<% if(renderWeb) { %>
  <script>...</script>
  <link href="/css/mobile.css" ... />
<% } %>
```

# Summary

# Summary: Growing number of...

- Connected physical devices
- Types of devices
- Platforms
- Device models

# Summary: Two problems

- How to scale development of apps? => HTML Hypermedia APIs
- How to make our web work on every browser? => AWD

HTML Hypermedia APIs + AWD => Perfect combo!



# Summary: Highlevel view

Native apps (80/20)

Web browsers

Enhanced web (80/20)

HTML Hypermedia API  
+  
AWD

*No application logic*

*Application logic*



# Questions?



# Thank you!

@gustaf\_nk

<https://github.com/gustafnk/RestBugs>

