



---

**Responsive cross-platform web apps**

*Deliver the best UX across all the platforms*

---

*Ilya Pukhalski*



Ilya Pukhalski,  
EPAM Mobile Competency Center



[@pukhalski](https://twitter.com/pukhalski)

**Why should I care about  
every device type?**

**0.122 Mb, 3 134 500 \$,  
~2200 pounds,  
computing power of modern microwave**



**128 Mb, 399 \$,  
135 g,  
computing power of NASA in 70s**



[iPhone, 1st gen](#)

2005

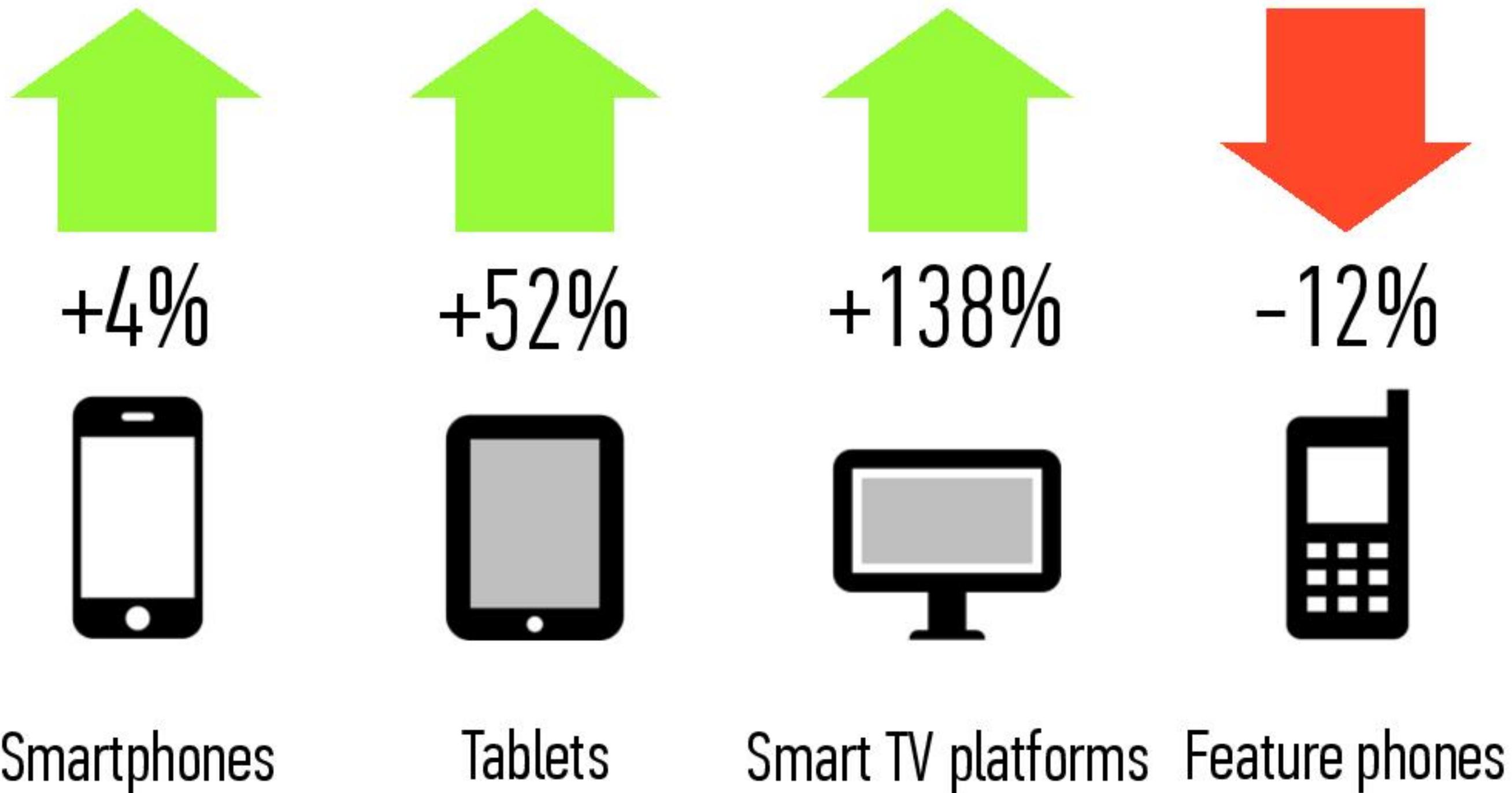
A photograph capturing a large, dense crowd of people from behind, all looking towards a bright, illuminated stage or building in the distance. The scene is filled with numerous small, glowing screens from mobile phones held by the spectators. In the lower-left foreground, the large, bold white text "2013" is overlaid.

2013

**Mobile is a king...  
but don't forget  
about desktop and Smart TV**

# Mobile Traffic

(2012)



**Yes, we have RWD...**

# Reflow problems

1. Images
2. Navigation
3. Tables
4. Audio and video
5. *Other*

Developer Tools - http://activetheory.net/

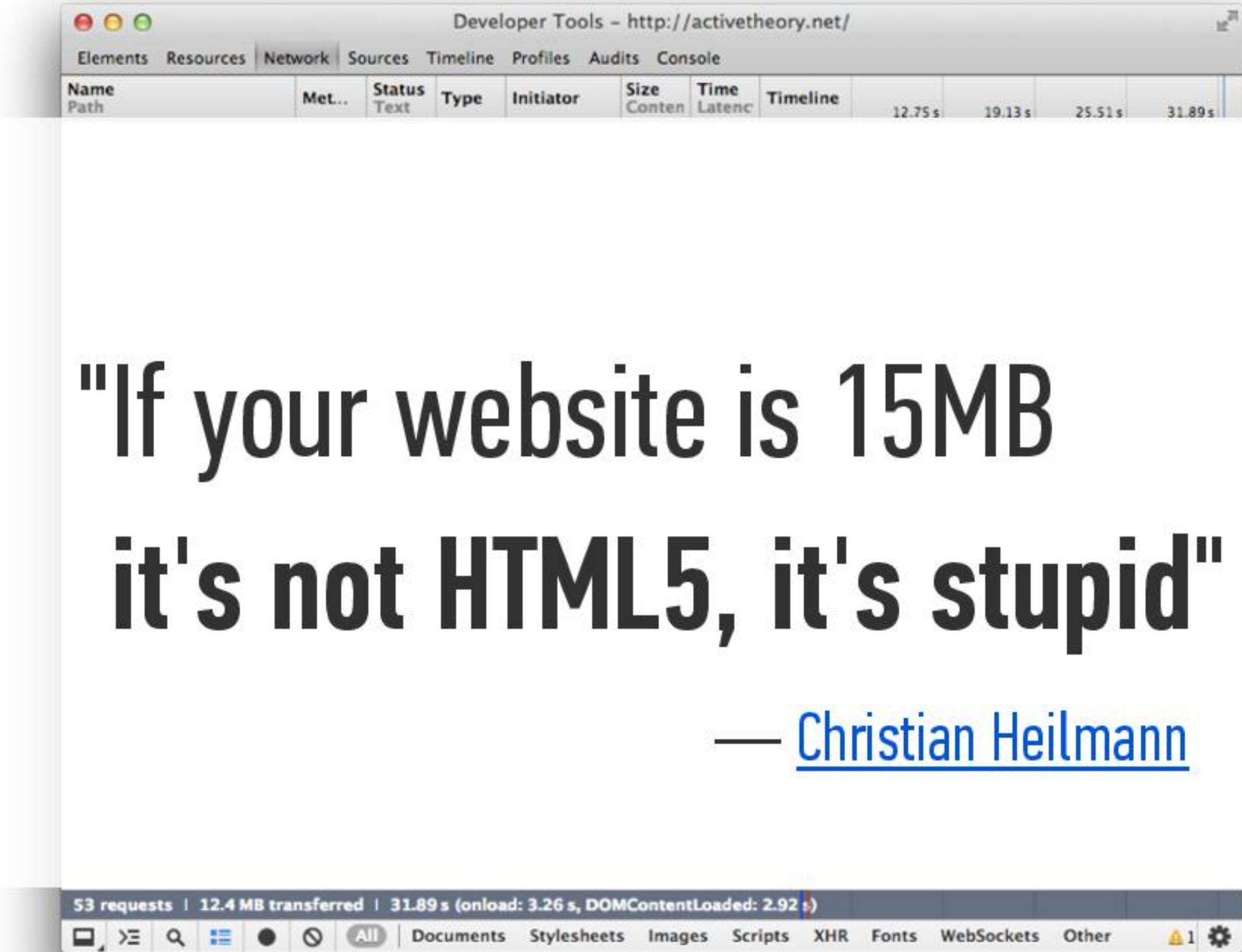
Elements Resources Network Sources Timeline Profiles Audits Console

Name Path	Met...	Status Text	Type	Initiator	Size Content	Time Latency	Timeline	12.75 s	19.13 s	25.51 s	31.89 s
"If your website is 15MB it's not HTML5, it's stupid"											

— [Christian Heilmann](#)

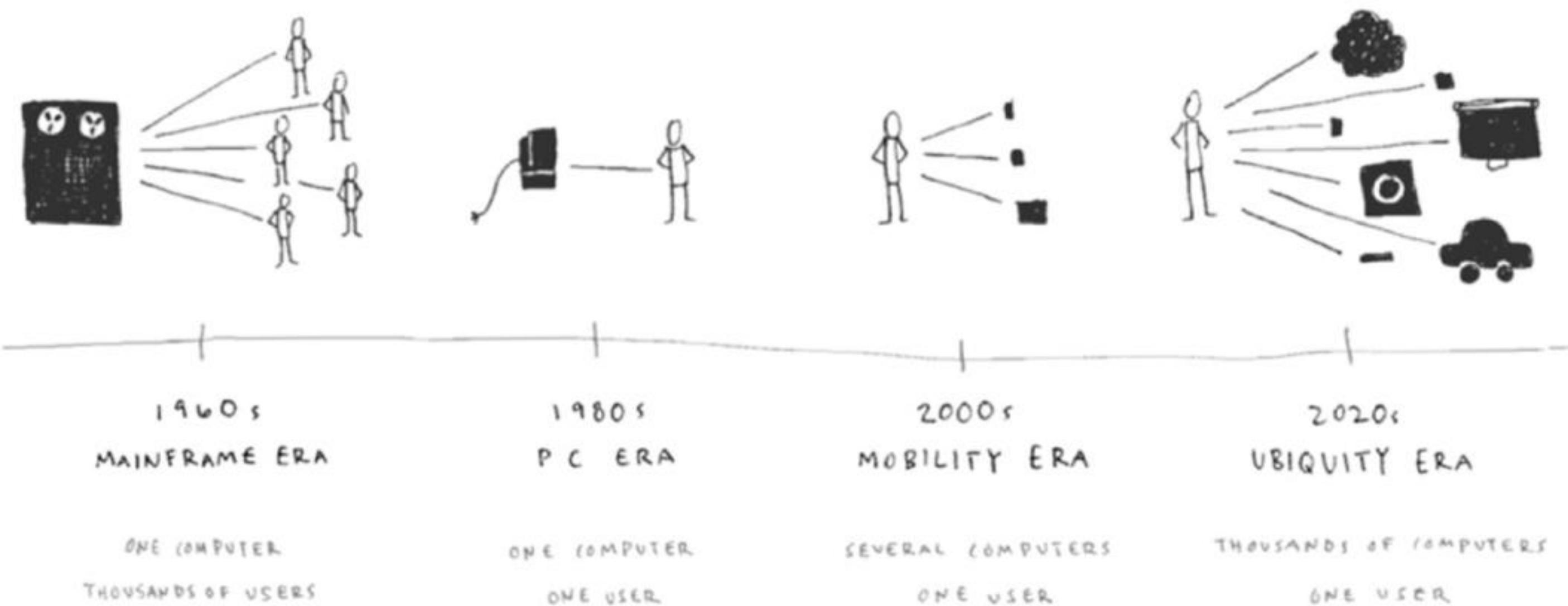
53 requests | 12.4 MB transferred | 31.89 s (onload: 3.26 s, DOMContentLoaded: 2.92 s)

Documents Stylesheets Images Scripts XHR Fonts WebSockets Other 1 🔍



# User interaction problems

## from context to personalization



# Content adaptation problems



**«A responsive website  
is not automatically  
a mobile-friendly website»**

— [Jon Arne Sæterås](#)

# Should I care about web?



User situation

# Facebook DAU vs Mobile DAU

## Daily Active Users (DAUs)

In Millions

- Rest of World
- Asia
- Europe
- US & Canada



DAUs / MAUs

Q2'11	Q3'11	Q4'11	Q1'12	Q2'12	Q3'12	Q4'12	Q1'13	Q2'13
56%	57%	57%	58%	58%	58%	59%	60%	61%

# Facebook DAU vs Mobile DAU

## Mobile Daily Active Users (Mobile DAUs)

In Millions



**78% of US Facebook users are mobile**

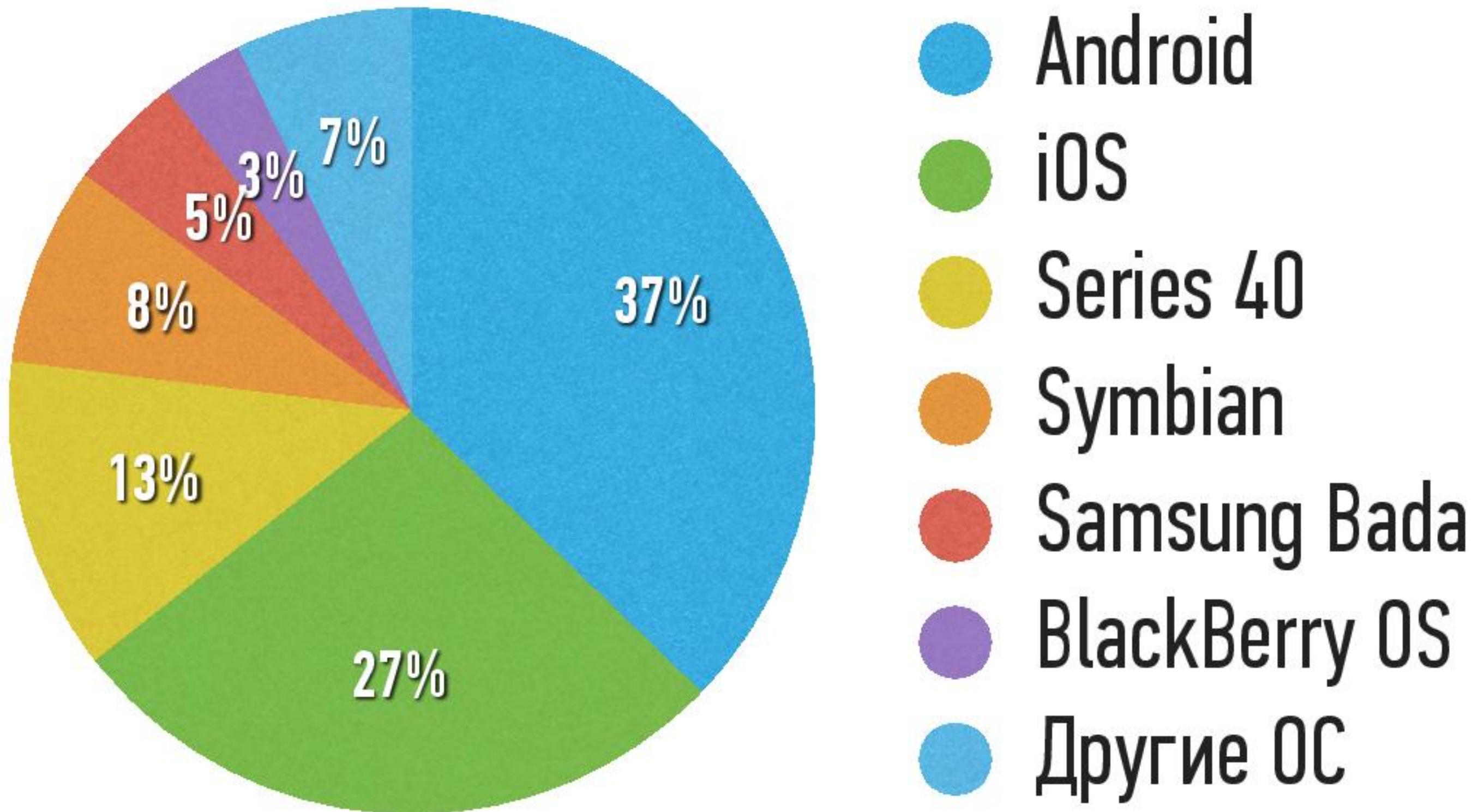
# Wunderlist



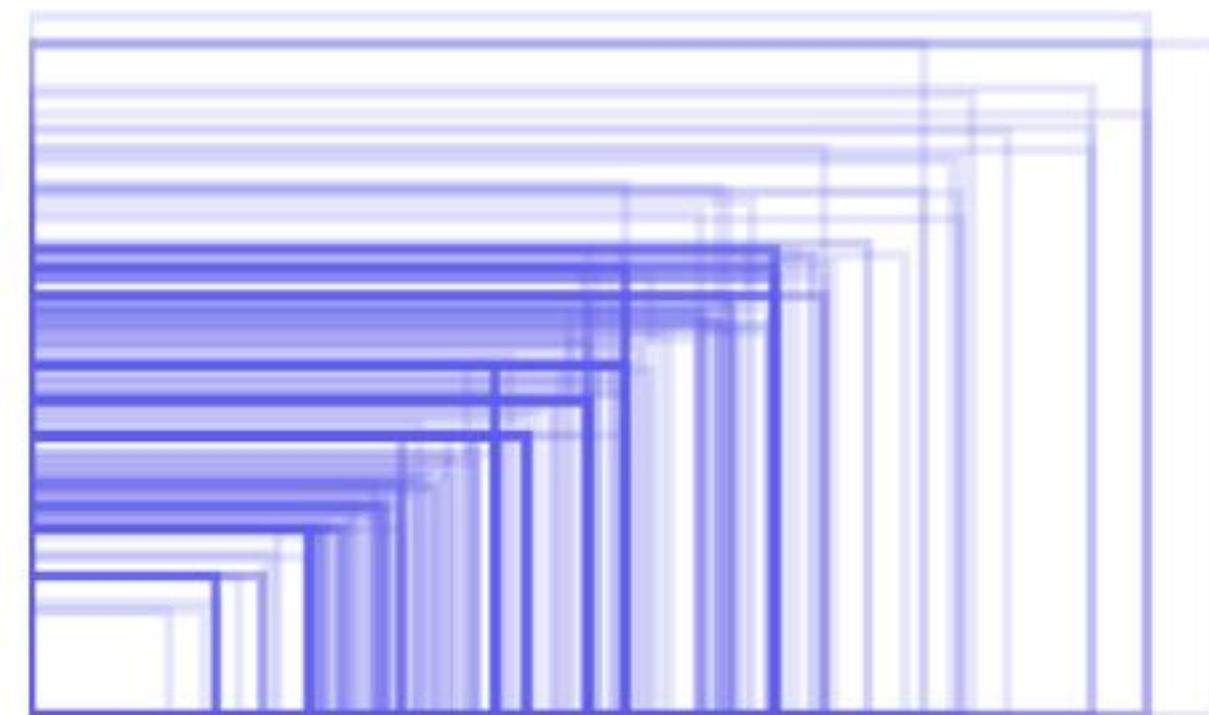
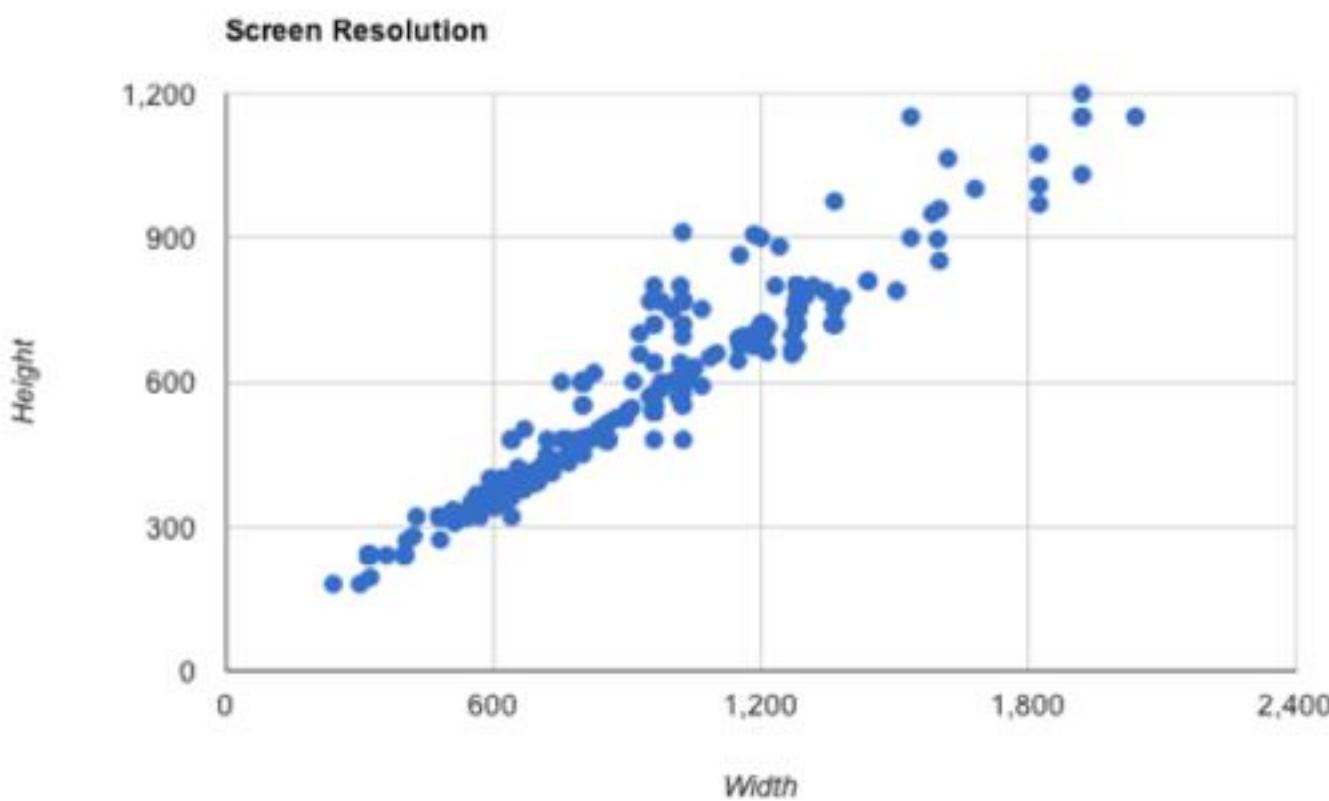
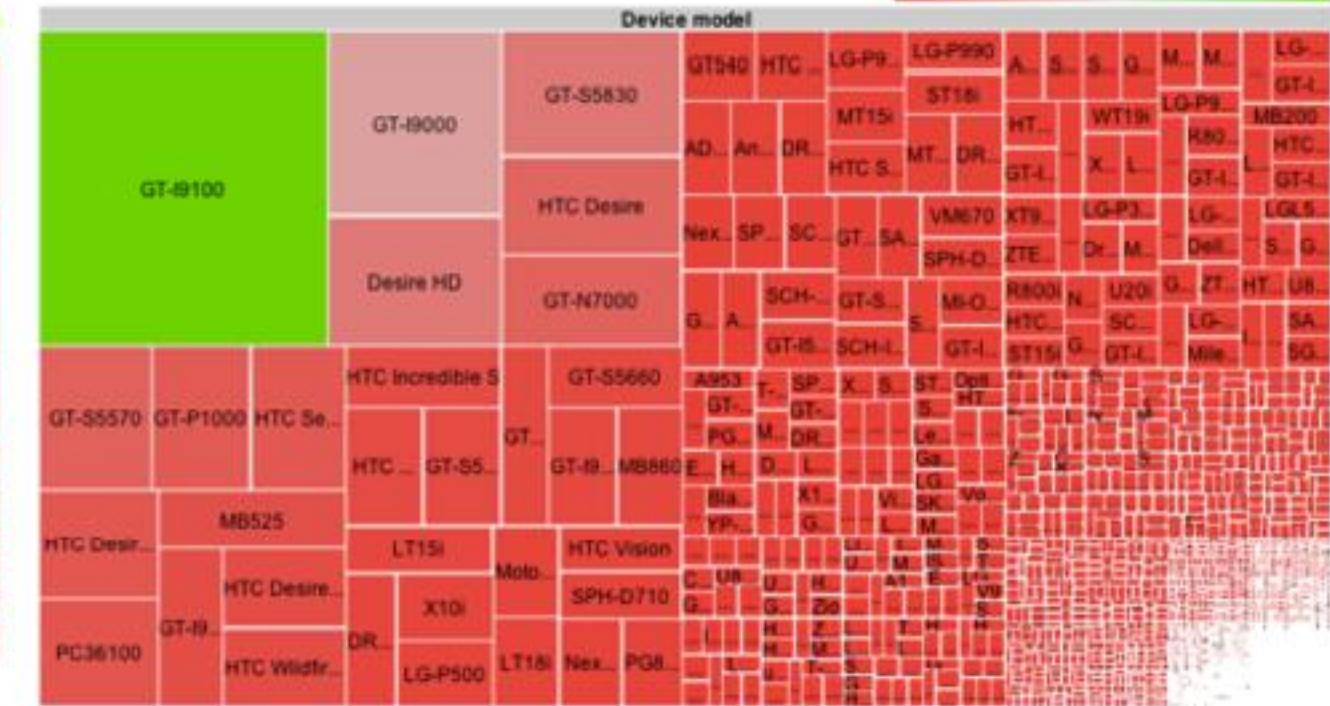
# Problems of web apps

# Mobile operating systems

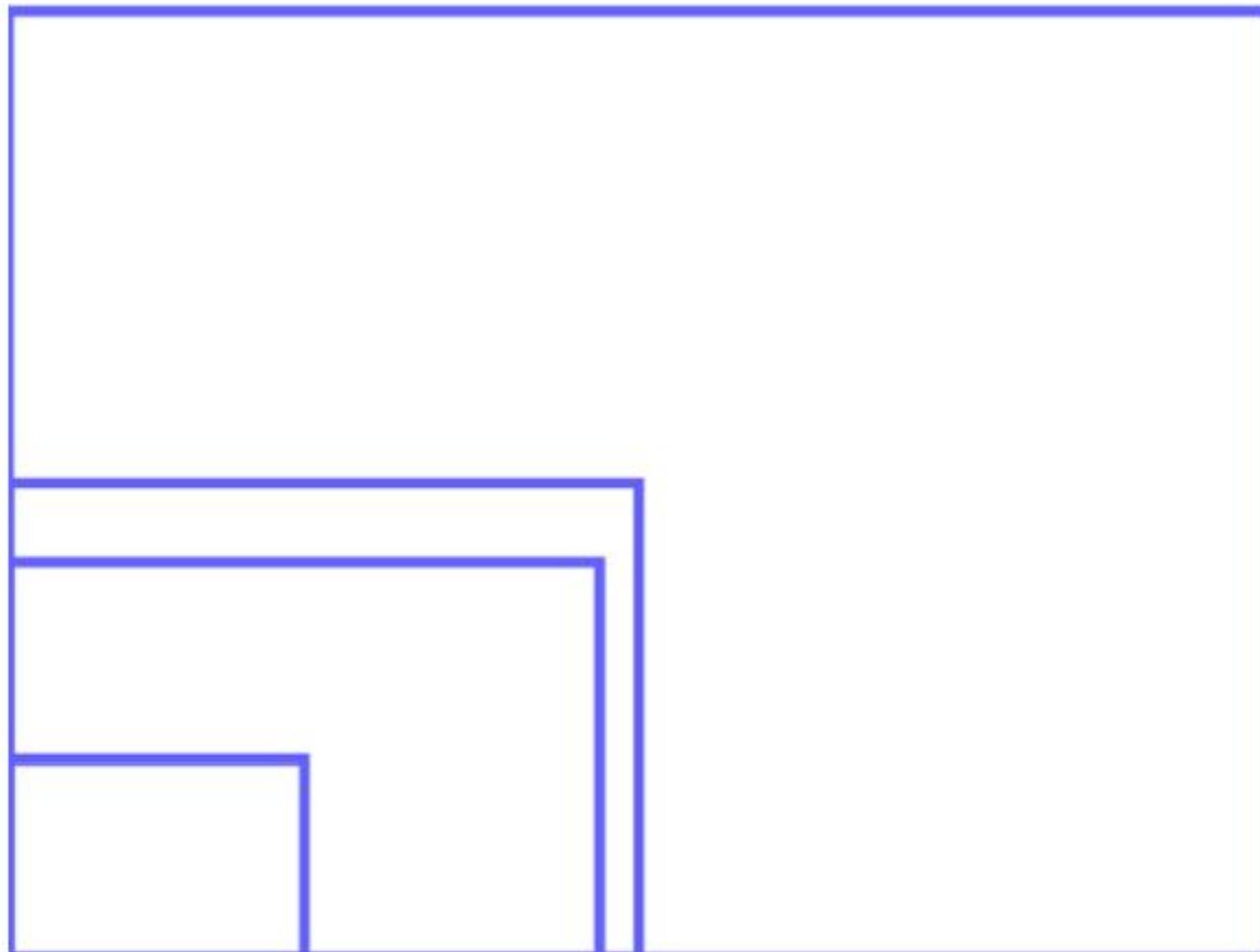
(MARCH 2013)



# Some Android facts



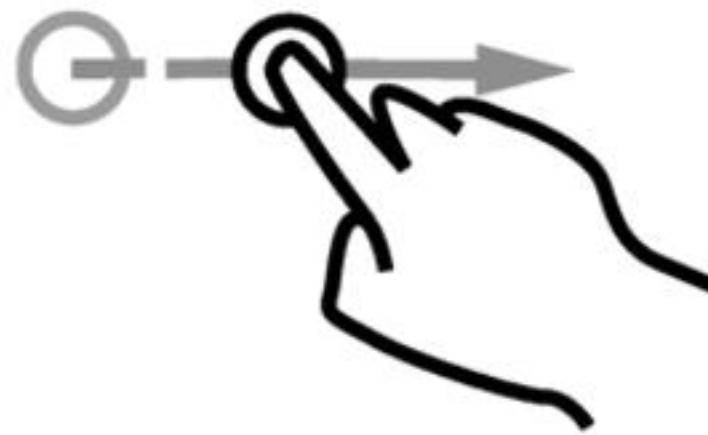
# Some iOS facts



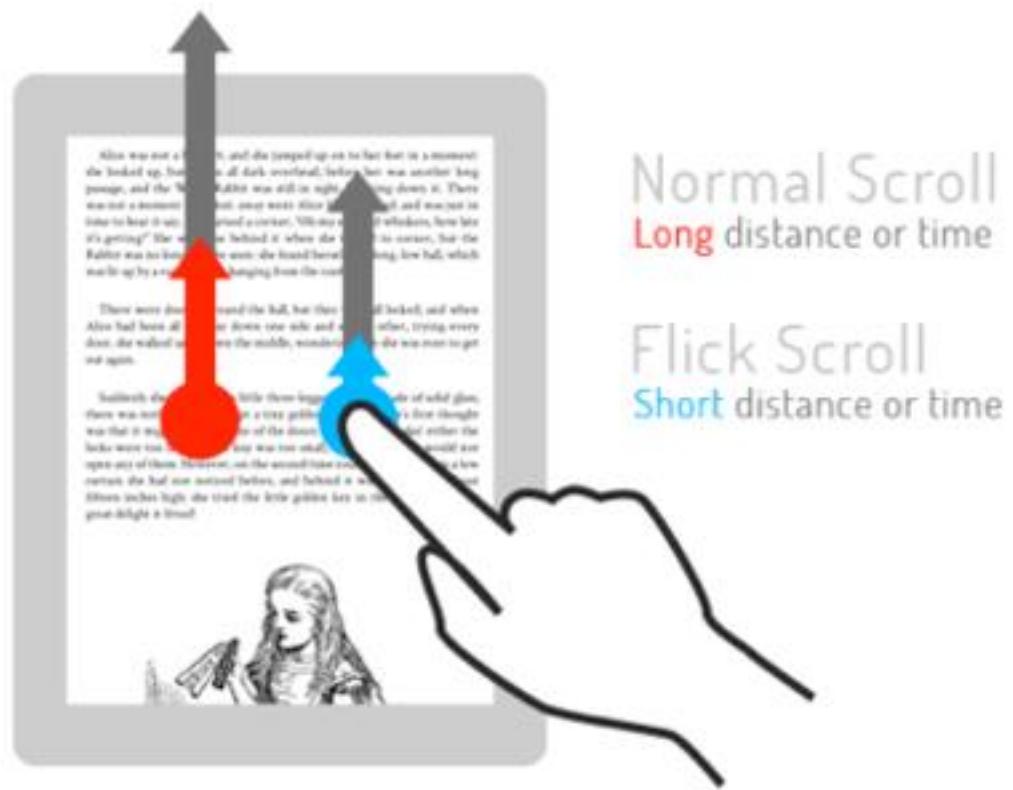


# **Access to device features**

# **Complex architecture of web app**



**Touch-events and gestures**  
together with the other input types

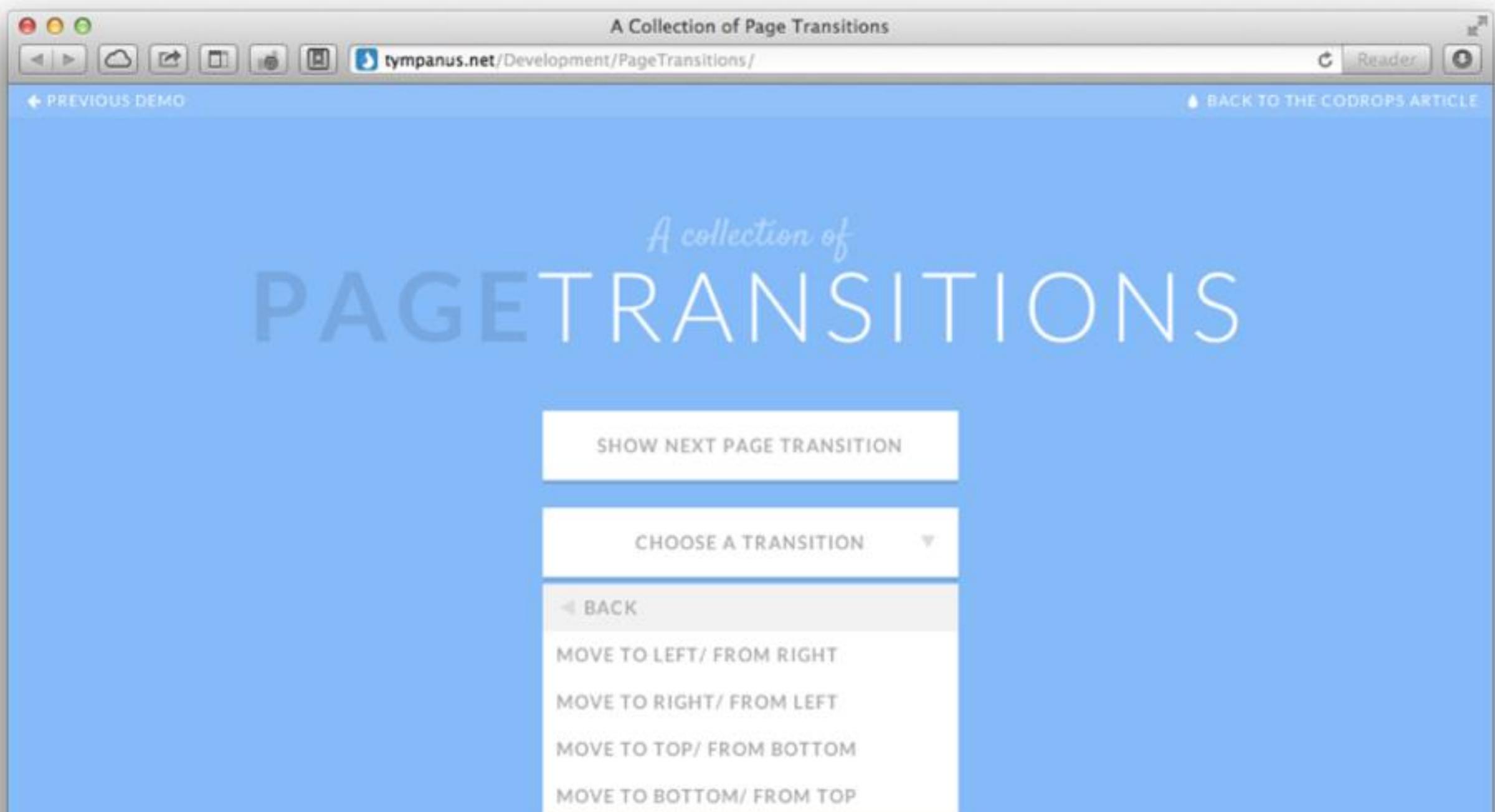


Normal Scroll  
**Long** distance or time

Flick Scroll  
**Short** distance or time

# Scrolling areas and sticky elements

# Animations



# Additional interface elements

The screenshot shows a web browser window with the title "Form elements - jQuery Mobile Demos". The address bar displays "view.jquerymobile.com/1.3.1/dist/demos/widgets/forms/". The main content area is titled "Sliders". It contains three examples of sliders:

- Slider:** A standard slider with a handle at position 50. To its left is a text input field containing the value "50".
- Slider with fill and step of 50:** A slider with a blue track and a white handle at position 60. To its left is a text input field containing the value "60".
- Slider with fill, mini, track theme:** A smaller slider with a blue track and a white handle.

Each slider example includes a "View Source" button below it.

**Native-like look and feel**

# Problems of current solutions

[Home](#) / [Products](#)[Products](#)[Support](#)[Tr](#)

## Sencha Touch

Build Mobile Web Apps with HTML5



jQuery Mobile

jquerymobile.com

jQuery

**jQuery mobile**

Demos API

JQUERY MOBILE 1.3.1 RELEASED!

### jQuery Mobile: Touch-Optimized Web Framework for Smartphones & Tablets

A unified, HTML5-based user interface system for all popular mobile device platforms, built on the rock-solid jQuery and jQuery UI foundation. Its lightweight code is built with progressive enhancement, and has a flexible, easily themeable design.

[Latest stable version - 1.3.1](#)

Legacy versions: 1.2.1 - 1.1.2 - 1.0.1



# Kendo UI

THE WAY OF HTMILS

[Blogs](#) [Docum](#)[Products](#) ▾Mobile  
with HTBuild apps that  
WinPhone8, iOS  
and tablet platf  
and JavaScript.[Launch Dem](#)[Overview](#)[Widgets](#)[What's New](#)[Roadmap](#)

Native UI on every

Build apps and sites for mobile devices that offer native-like experience with no extra coding. Currently supporting: (NEW!) WinPhone8, iOS, Android, Blackberry 10 and adapting UI in the [Kendo UI Mobile demos](#).

**telerik**

# Kendo UI

THE WAY OF HTML5

**PRODUCT LIST**

Add new record

Drag a column header and drop it here to group by that column

Product Name	Category	Unit Price
Chef	Show items with value that	\$11.00
Chef	Is equal to	\$11.00
Chef	Is not equal to	\$11.00
Chef	Starts with	\$11.00
Chef	Contains	\$11.00
Chef	Does not contain	\$11.00
Chef	Ends with	\$11.00
Chang		\$11.00
Annika Syrup		\$11.00
Chef Anton's Cajun Seasoning		\$21.35
Chef Anton's Gumbo Mix		\$21.35
Grandma's Boysenberry Sauce		\$21.35
Uncle Bob's Organic Dried Pears		\$31.00
Nuts	Is equal to	\$31.00
Nuts	Starts with	\$31.00
Nuts	Contains	\$31.00
Nuts	Does not contain	\$31.00
Nuts	Ends with	\$31.00

Overview    Widgets    What's New

Mo NE

September 2012

Su Mo Tu We Th Fr Sa

**Sencha**

Products | Support | Training | Company

Home / Products / Sencha Ext JS

# Sencha Ext JS

JavaScript Framework for Rich Desktop Apps



Ext JS includes over 100 right out of the box.

View some of our favorites below.

[View Ext JS 4.2 Examples](#)

Overview | Features | Samples & Demos | Testimonials | Road Map

### Combination Examples

- Combination Examples
- Enterprise
- Accessibility
- Grids
- Charts
- Tabs
- Windows
- Trees
- Layout Managers
- Drawing
- Drag and Drop

**Kitchen Sink**

Showcase of Ext JS components using a preview release of the new Neptune theme



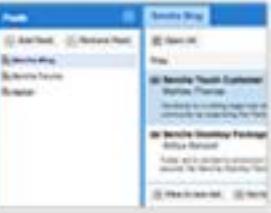
**Ext JS Calendar**

Example Calendar application. Demonstrates the new Day, Week and Month views and how to combine them.



**Sencha Touch Examples**

Sencha Touch Examples





**Kendo UI**

THE ART OF WEB DEVELOPMENT



*Knockout.*



**Sencha**



+



# Requirements

1. Scalable and maintainable architecture
2. Modularity
3. Reusability (even across the apps)
4. Async loading and UI
5. Good support range

6. Scalable and maintainable architecture
7. Modularity
8. Reusability (even across the apps)
9. Async loading and UI
10. Different view layer and level of UX  
for different device types

# The idea

# Desktop

Component #1

Component #2

Component #3

Component #4

# Tablet

Component #1\*

Component #2

Component #3\*

Component #4

Page #1

Page #2

# Phone

Component #1\*\*

Component #2\*

Component #3\*

Component #4

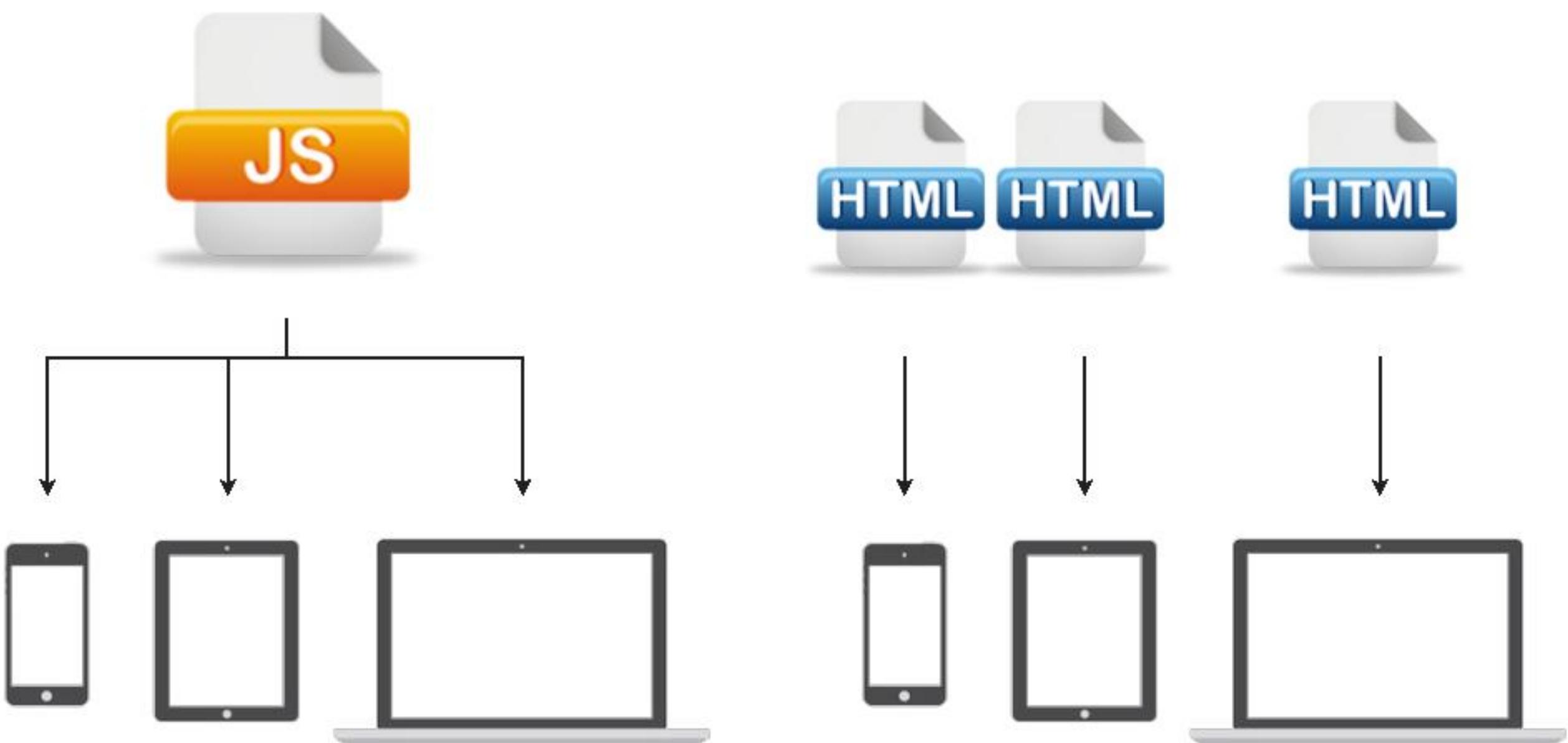
Page #1

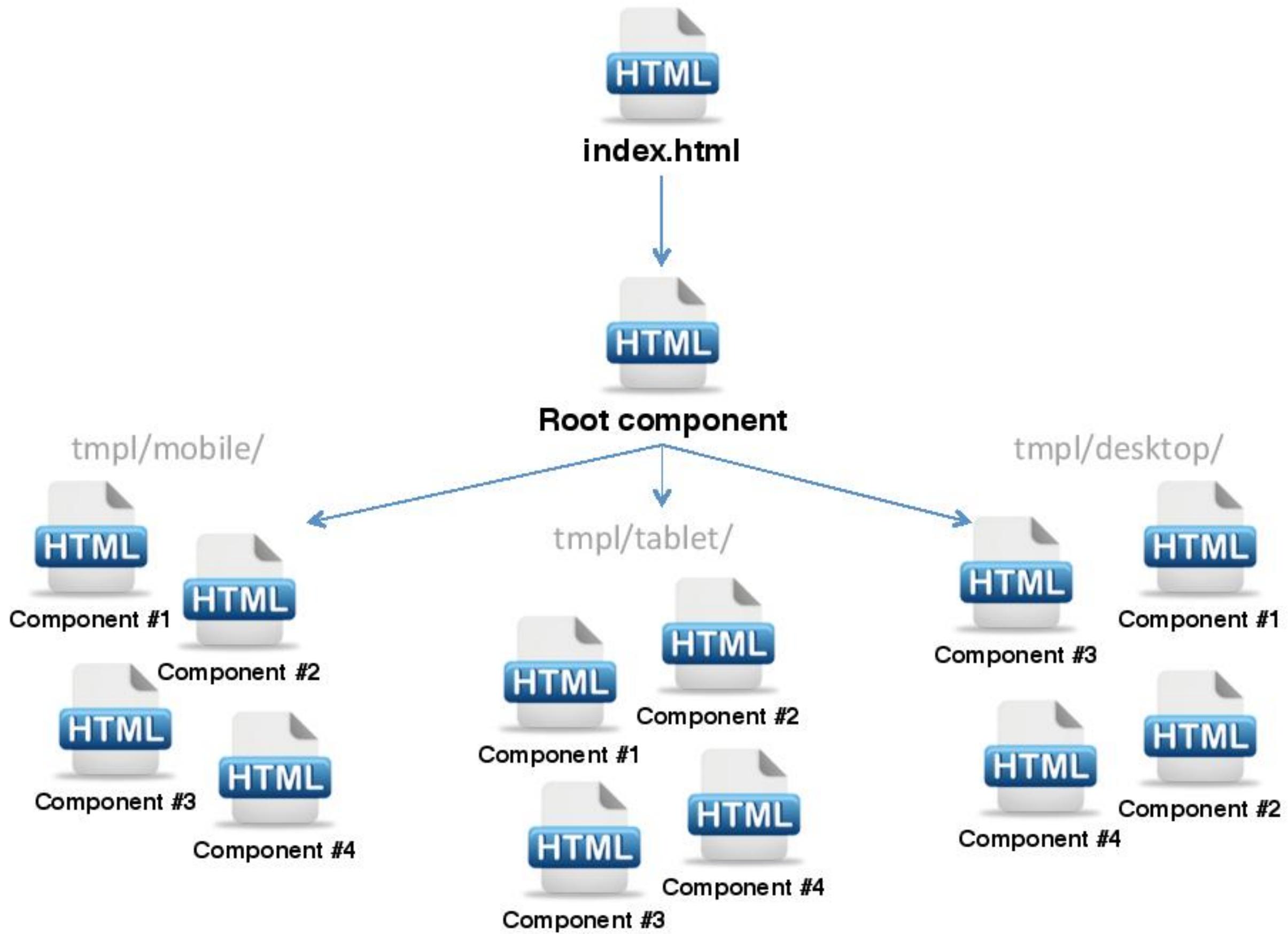
Page #2

Page #3

# The magic\*

\*the mechanics





# Core modules

+ helpers and utilities

# Components

+ library of high-level abstract components

# UI Elements

+ library of predefined UI elements

# Core modules

# Core modules

1. Script Loader
2. "Classes" for Component, View, Model and Collection
3. Router and history

# Core modules

4. Page Switcher
5. User input adapter
6. Device information object
7. Utilities and helpers

# Component placeholder

```
<div data-id="app-smart-component" data-
  component="smart-component">
  <script>
    APP.setOptionsByID(
      'app-smart-component',
      {
        prop: 'value'
      }
    );
  </script>
</div>
```

# jQuery / Zepto hooks

```
var _oldshow = $.fn.show;

$.fn.show = function () {
    var res = _oldshow.apply(this,arguments);
    APP.trigger('core:loadChildComponents', this);
    return res;
};
```

# Pages and routing

# Page switching by route

```
APP.start({
  settings: {
    // ...
  },
  router: {
    routes: {
      '' : 'home'
    },
    home: function () {
      // ...
    }
  }
});
```

```
<div data-page="home">
  <!-- HTML, components,
  UI elements -->
</div>
```

# Page switching animations

*...vary for different device types*

1. CSS animations
2. JavaScript animations as fallback
3. Tricks (fixed height, translateZ, backface-visibility)
4. Works together with routing automatically

# User input adapter

...one event to rule them all — "tap"

# Utilities and helpers

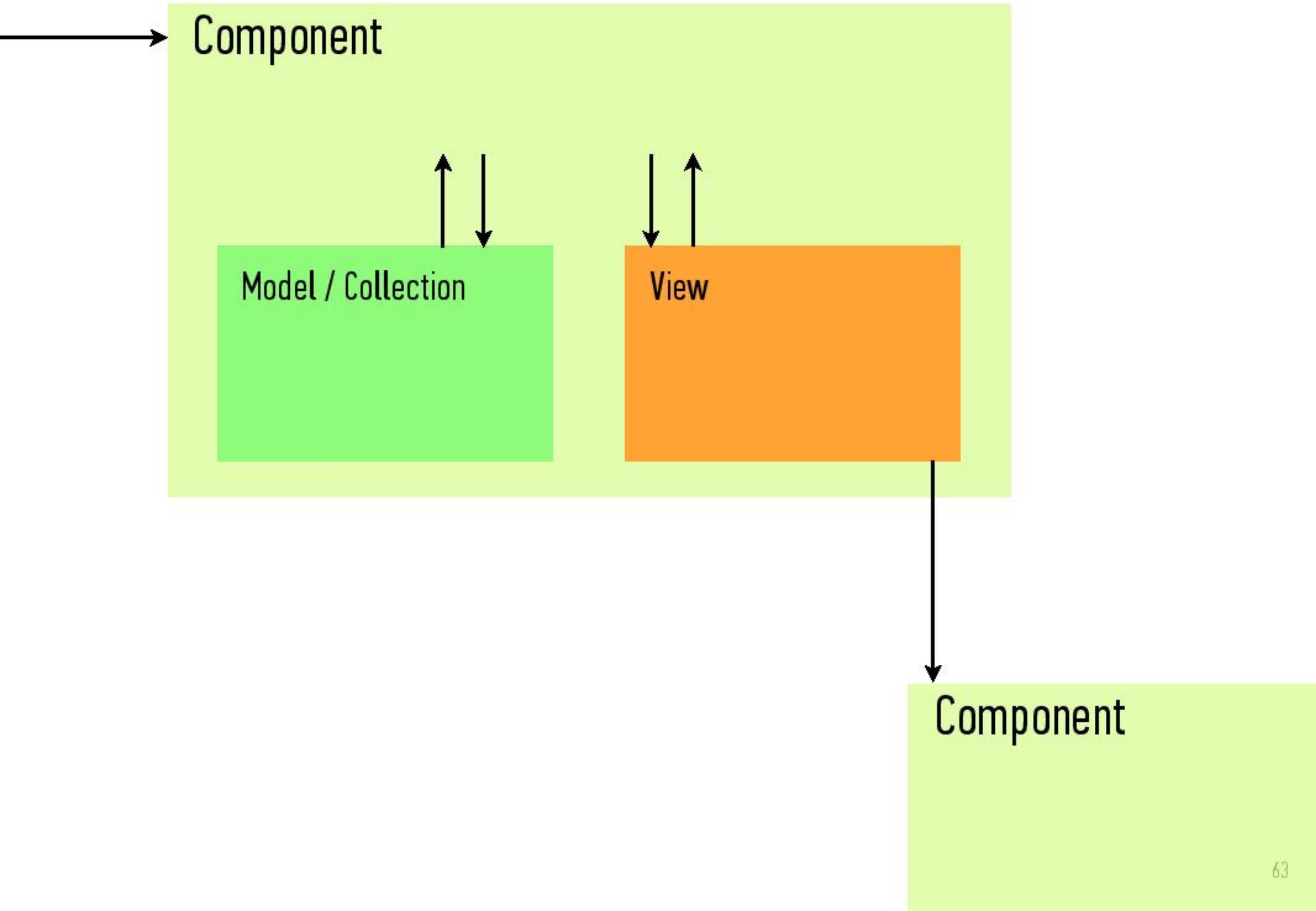
# Caching

1. Templates definition are stored in localStorage
2. Refreshing cache for new app version
3. Manifest for control the world!

# Dessert

1. Address bar hiding for smartphones
2. Device/browser information
3. Extended debugger

# Components



```
<div data-id="componentId" data-
  component="componentName">
  <script>
    App.setOptionsByID(
      'componentId',
      {
        prop: 'value'
      }
    );
  </script>
</div>
```

```
App.define(  
  'componentName',  
  App.Component.extend({  
    initialize: function () {  
      },  
      Collection: 'collectionName'  
      View: 'viewName'  
    } )  
);
```

Minsk

59°F Fair



Today  
65° 43°



Tomorrow  
65° 33°



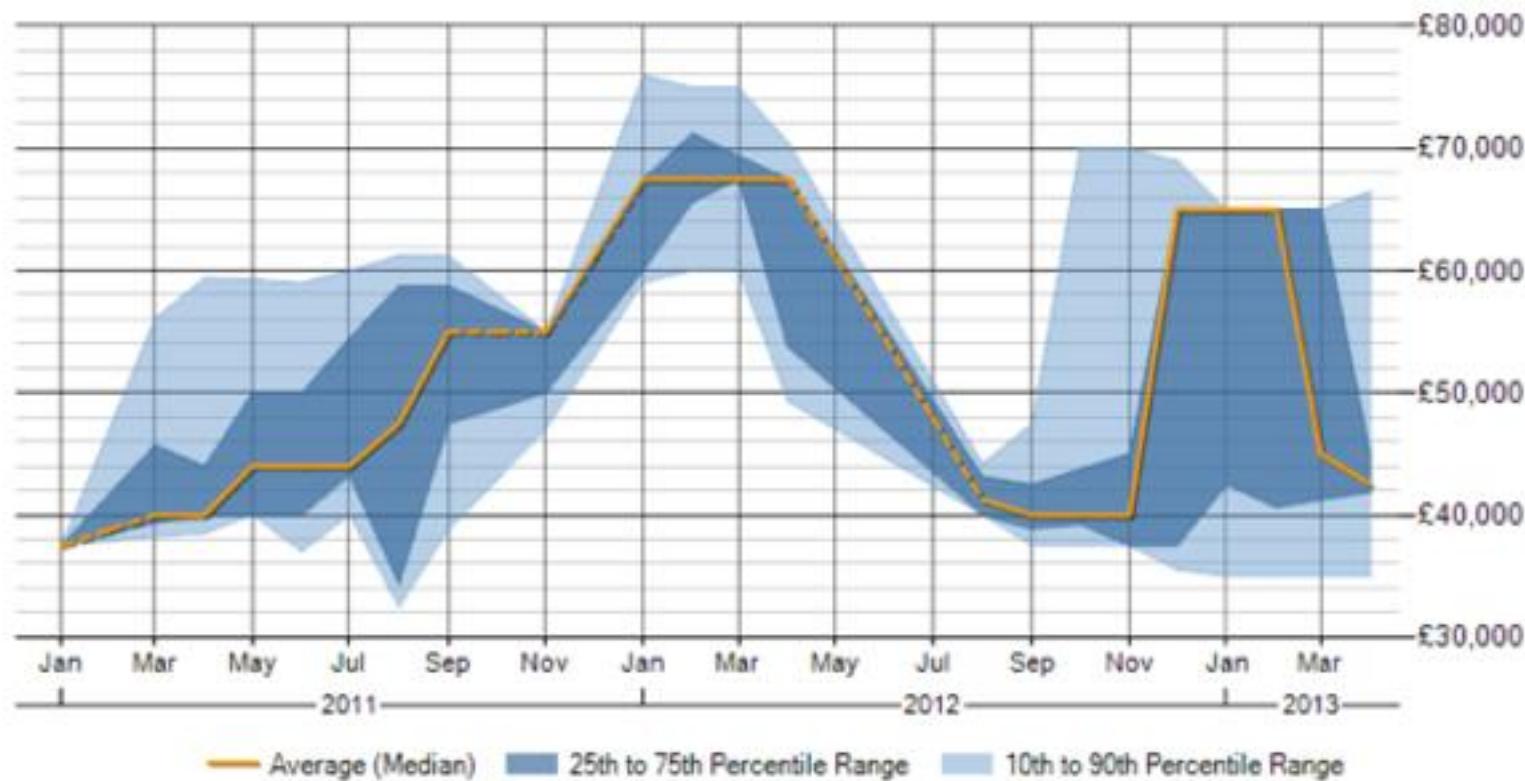
Sunday  
55° 33°



Component == Widget

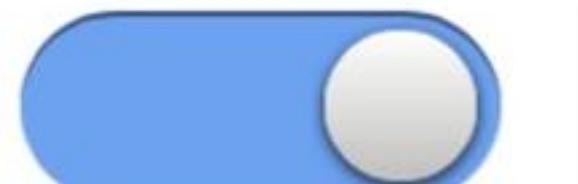
# UI Elements

# Components? UI Elements?



Component

Wi-Fi Switch



UI Element

# UI Elements: how it works

```
<label for="wifi-switch">Wi-Fi Switch</label>  
<input type="checkbox" data-role="switch"  
id="wifi-switch" />
```



# UI Elements: how it works

```
<div class="app-switch">
  <label class="app-switch-control" for="wifi-switch">
    <input type="radio" name="wifi-switch" id="wifi-switch">
    <span class="app-switch-track">
      <span class="app-switch-track-wrap">
        <span class="app-switch-thumb"></span>
      </span>
    </span>
  </label>
  <label class="app-switch-label" for="wifi-switch">Wi-Fi
    Switch</label>
</div>
```



# Communication mechanisms

# Why not events?

# Application level

- `app:started`
- *pages:show*
- *app:navigate*
- *component:id:refresh*
- `component:id:constructed`
- ...

# Component / View / Collection / Model level

- loaded / fetched
- *refresh*
- ...

# Going deeper

# Device and feature detection on the server side

# Your browser lies



Feature detection is good,  
browser/device detection is necessary evil

«The more experienced  
a mobile Web developer is,  
the more they rely  
on server-side browser detection»

— [PPK in "The Mobile Book"](#)

# Device detection libraries and services

1. [WURFL](#)
2. [WURFL Cloud by Scientia Mobile](#)
3. [OpenDDR\\*](#)
4. [Device Atlas](#)
5. [51 Degrees](#)

# Template rendering on the server side

1. The same templating engine for server and client side
2. Rendered HTML instead of JSON+TMPL
3. View of Component is only for DOM bindings

# Bonus



@xframeworkjs



@pukhalski