

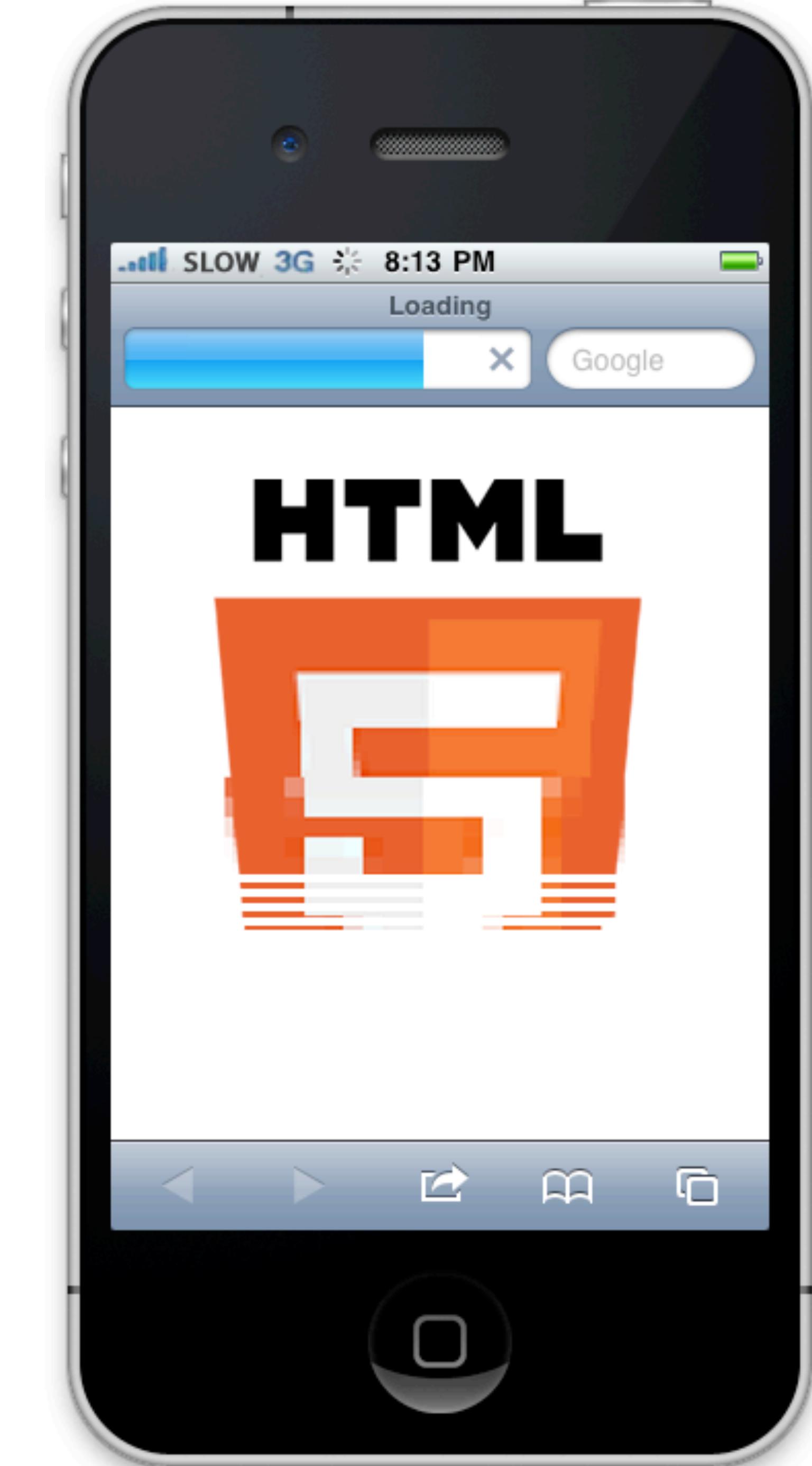
O'REILLY®

# Velocity

Web Performance & Operations  
CONFERENCE

June, 14, 2011  
Santa Clara, CA

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



# mobile web & html5 performance optimization

max firtman  
[@firt](https://twitter.com/firt)  
[firt.mobi](http://firt.mobi)

# who am I?

mobile+web developer

mobilexweb.com blog

wpo passionate



@firt



ITMaster  
PROFESSIONAL TRAINING

# where?

buenos aires ~ argentina

patagonia  
soccer  
tango



true lies  
tango scene



# books



*Reaching Users on iPhone, Android,  
BlackBerry, Symbian, and more*



*Programming the*

# Mobile Web

O'REILLY®

*Maximiliano Firtman*

*Using HTML5 to Design Web Apps for Tablets and Smartphones*



O'REILLY®

*Maximiliano Firtman*

# good news!

today ~ last day

save 50%  
[oreilly.com](http://oreilly.com)



# agenda

# agenda

- ▶ a little overview of mobile web
- ▶ measuring and debugging
- ▶ mobile html5
- ▶ optimization areas
- ▶ tango dance

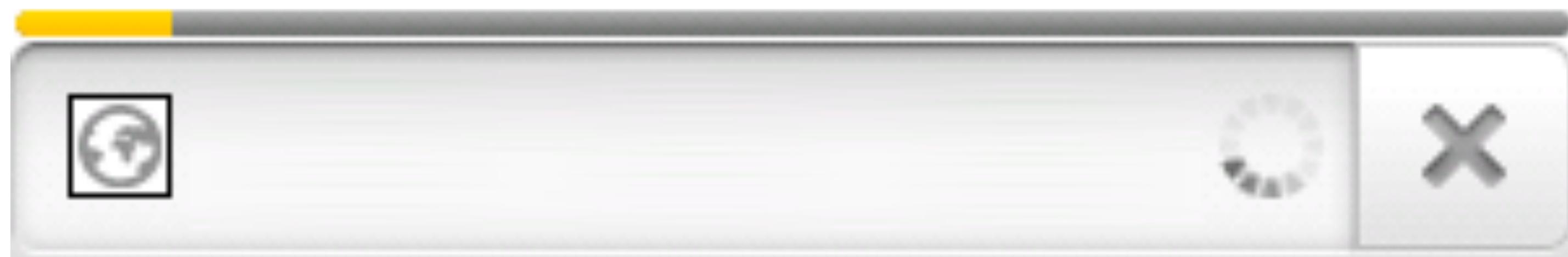
# facts

**you are guilty!**

users hate you



why?



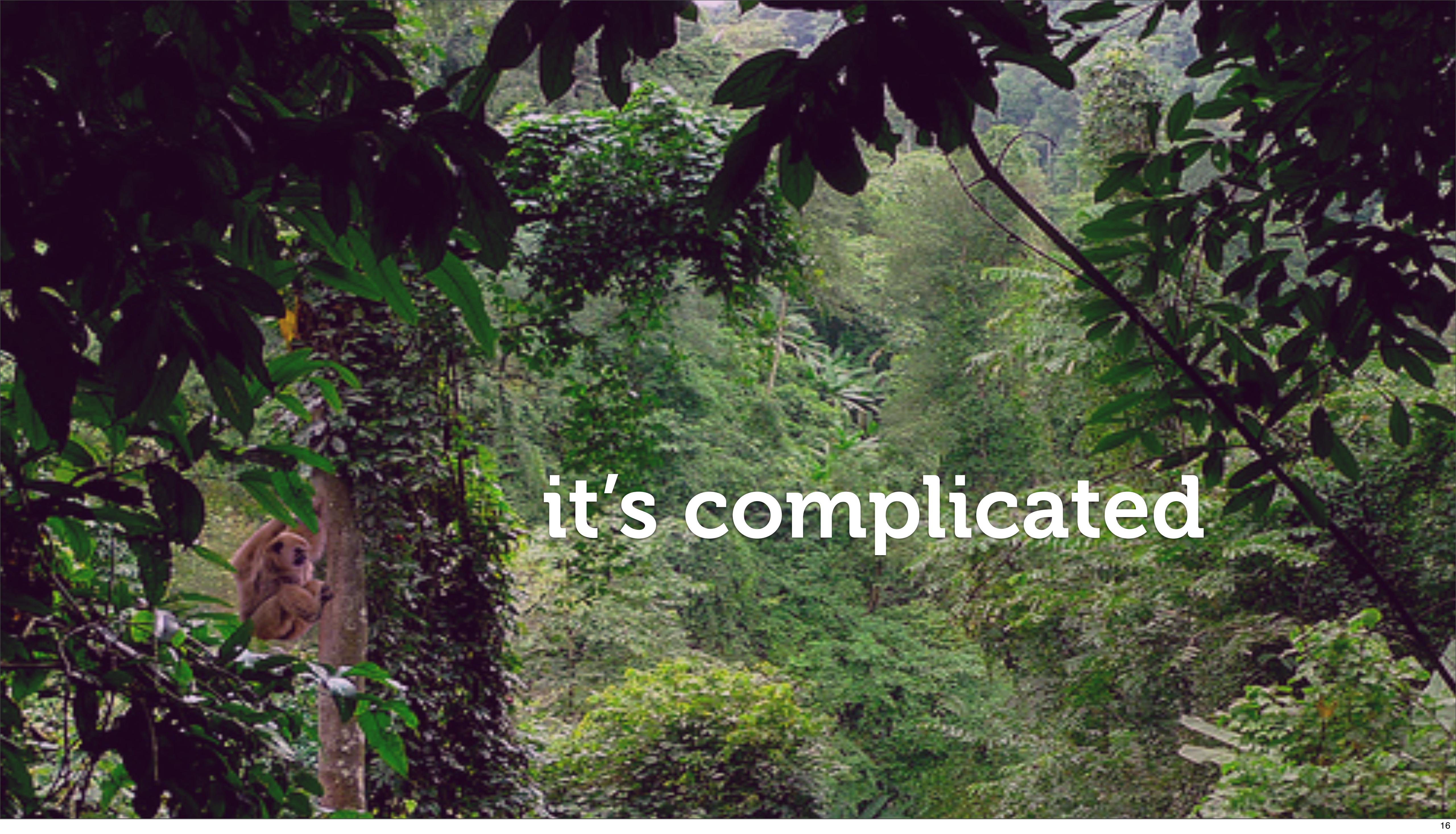
mobile web  
is slow



Picture from Simon Howden freedigitalphotos.net

mobile web is **slow** because...

we are doing it **wrong**

A photograph of a dense jungle scene. In the lower-left foreground, a monkey is clinging to a vertical tree trunk. The surrounding environment is filled with thick green foliage, including large leaves and various plants. The lighting suggests a bright day in a shaded forest area.

it's complicated

# differences

# differences

- ▶ slower networks
- ▶ higher latency
- ▶ slower hardware
- ▶ different browsing experience
- ▶ different context
- ▶ different behaviours
- ▶ different possible networks

# differences

```
cookie2: $Version="1"
connection: Keep-Alive
x-wap-profile: " http://www-ccpp-mdp.alcatel.com/files/ALCATEL-
BG3_1.0.rdf"
accept-language: en
content-length: 0
drm-version: 2.0
host: www.mobilexweb.com
accept-encoding: gzip, deflate
accept: application/vnd.wap.wmlc;Type=4365,
application/vnd.wap.wmlc;Type=4360,
application/vnd.wap.wmlc;Type=1108, application/vnd.wap.wmlc;Level=1.3,
application/vnd.wap.wmlc, application/vnd.wap.wmlscriptc,
application/vnd.wap.multipart.related,
application/vnd.wap.multipart.mixed, application/x-up-device,
application/vnd.phonecom.mmc-wbxml;Type=4364,
application/vnd.phonecom.mmc-wbxml, application/vnd.phonecom.im,
application/octet-stream, application/vnd.openwave.pp,
application/vnd.wap.sic, application/vnd.wap.slc,
application/vnd.wap.coc,
application/vnd.uplanet.bearer-choice-wbxml,
application/vnd.uplanet.signal,
application/vnd.uplanet.cacheop-wbxml, application/vnd.uplanet.alert-
wbxml,
image/vnd.wap.wbmp;Type=0, application/*, audio/iMelody,
audio/midi, text/x-hdml,
image/mng, image/x-mng, video/mng, video/x-mng, image/bmp, text/vnd.wap.wml
accept-charset:ISO-8859-1, US-ASCII, UTF-8; Q=0.8, ISO-10646-UCS-2;
Q=0.6
user-agent: Alcatel-BG3/1.0 UP.Browser/5.0.3.1.2 UP.Link/1.1
```

# mobile browsers

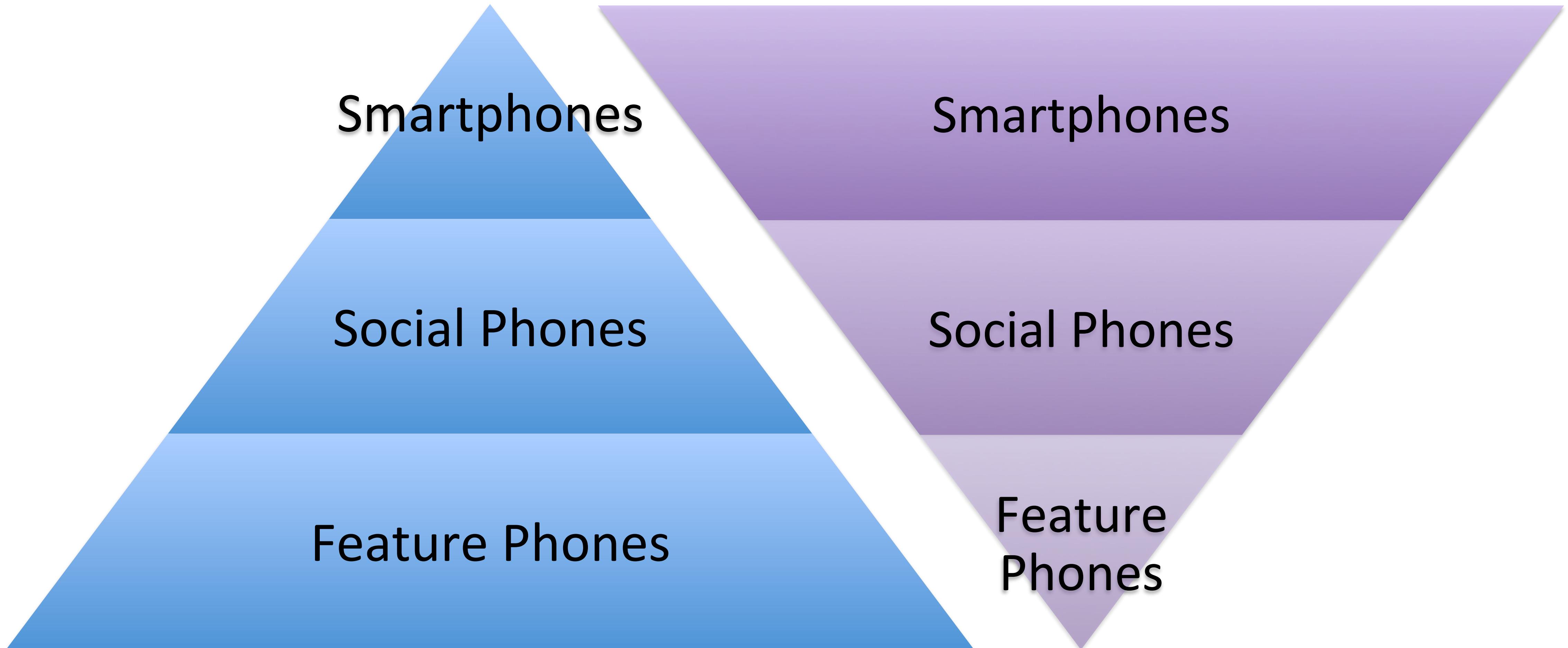
# differences

There are browsers supporting 1, 2,  
4, 6 parallel downloads

# mobile browsers

- ▶ too many
- ▶ (some) too limited
- ▶ (some) too innovative
- ▶ (some) proxied
- ▶ (most) without documentation
- ▶ (most) without a *name*
- ▶ (most) without debugging tools

# MOBILE WEB USAGE



**MARKET SHARE**

# a quick, dirty list

- ▶ Safari on iOS
- ▶ Android Browser
- ▶ Symbian Browser
- ▶ webOS Browser
- ▶ BlackBerry Browser
- ▶ Bada Browser
- ▶ Firefox
- ▶ Internet Explorer

# a quick, dirty list (cont.)

- ▶ NetFront
- ▶ Myriad
- ▶ Nokia Browser (Ovi)
- ▶ Phantom
- ▶ microB
- ▶ Opera Mobile
- ▶ Opera Mini
- ▶ Skyfire
- ▶ BOLT

# they are on tablets too!

- ▶ Safari on iOS
- ▶ Android Browser
- ▶ webOS Browser
- ▶ BlackBerry Browser
- ▶ Opera Mini

and with different versions...

# and there's more!

# pseudo-browsers

# pseudobrowsers

- ▶ mostly on iOS and Android
- ▶ uses the internal web control
- ▶ have different behavior than the native!



**we can do it!**

we need to forget  
and learn

# forget about

- ▶ pixels
- ▶ static designs
- ▶ desktop frameworks (ok, not always)
- ▶ always connected
- ▶ unlimited power

# learn about

- ▶ server-side detection
- ▶ progressive enhancement
- ▶ responsive design
- ▶ best experience for each context
- ▶ the top-model approach

**speed**

**speed ~ perception**



# measuring and debugging



# **two browsers with remote tools**

**Opera Mobile with DragonFly**

**BlackBerry Browser for PlayBook, for Smartphones 7.0**



and the  
others?

# emulators



- ▶ different types
- ▶ good for first testing
- ▶ free, +150
- ▶ [www.mobilexweb.com/emulators](http://www.mobilexweb.com/emulators)
- ▶ proxy: Charles Proxy, Fiddler

# real devices



- ▶ proxys on wifi? charles, fiddler
- ▶ proxys on the device: mobitest
- ▶ hybrids / pseudobrowsers: mobitest
- ▶ network sniffers: pcapperf
- ▶ server-side sniffers
- ▶ remote javascript tools: weinre, jdrop, mobileperf (p.ad.ag)

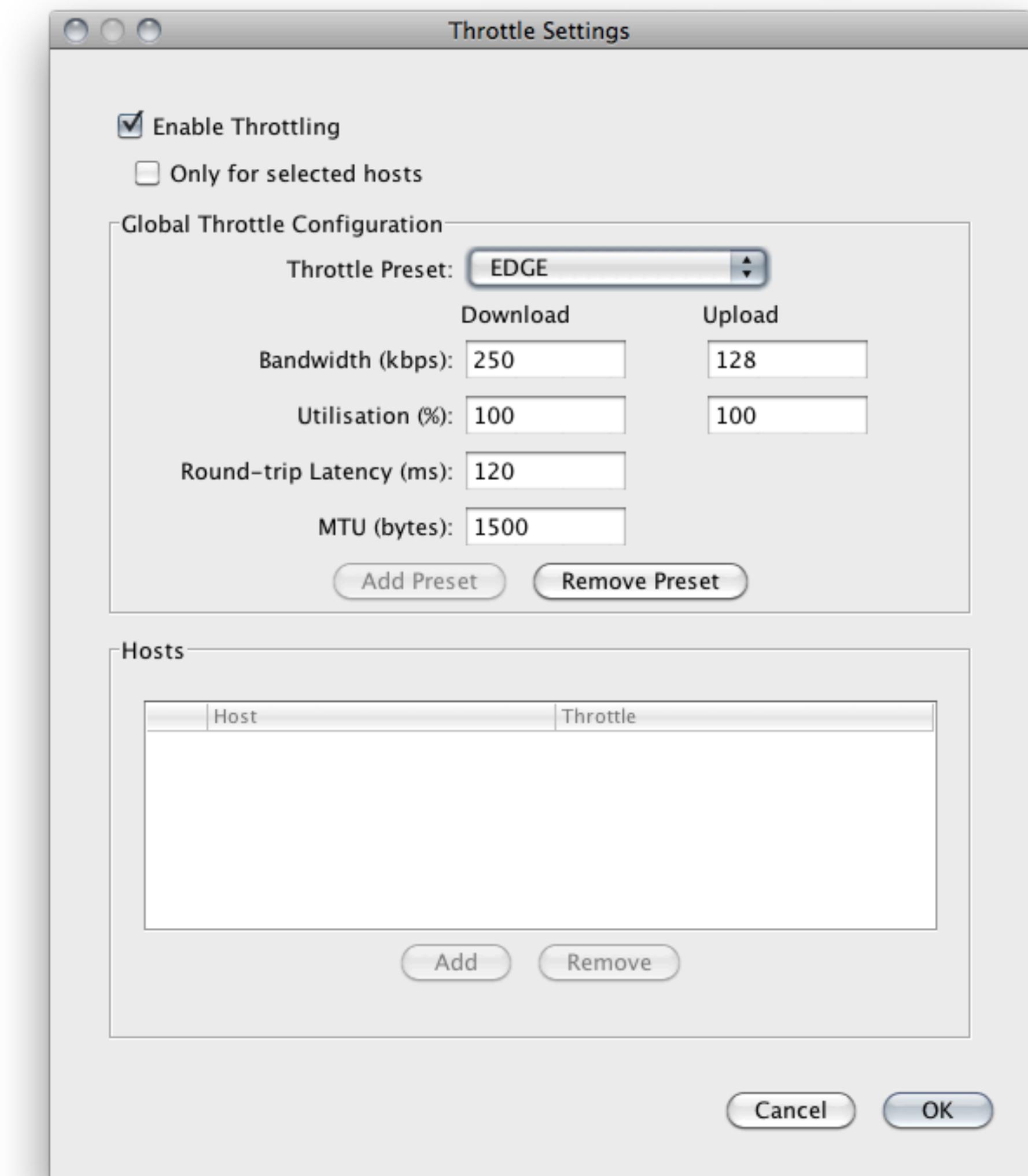
# remote labs



- ▶ deviceanywhere.com
- ▶ perfectomobile.com
- ▶ remote device access (nokia) - free
- ▶ lab.dev (samsung) - free

# 2g, 3g, 4g

- ▶ Remember to test it
- ▶ Charles Proxy helps you with throttling



# html5



# html5

if the device is not html5-compatible

- ▶ it's fine...
- ▶ it will work without those enhancements
- ▶ even html5-compatible devices don't have the same compatibility level
- ▶ don't be fanatic, be multiplatform

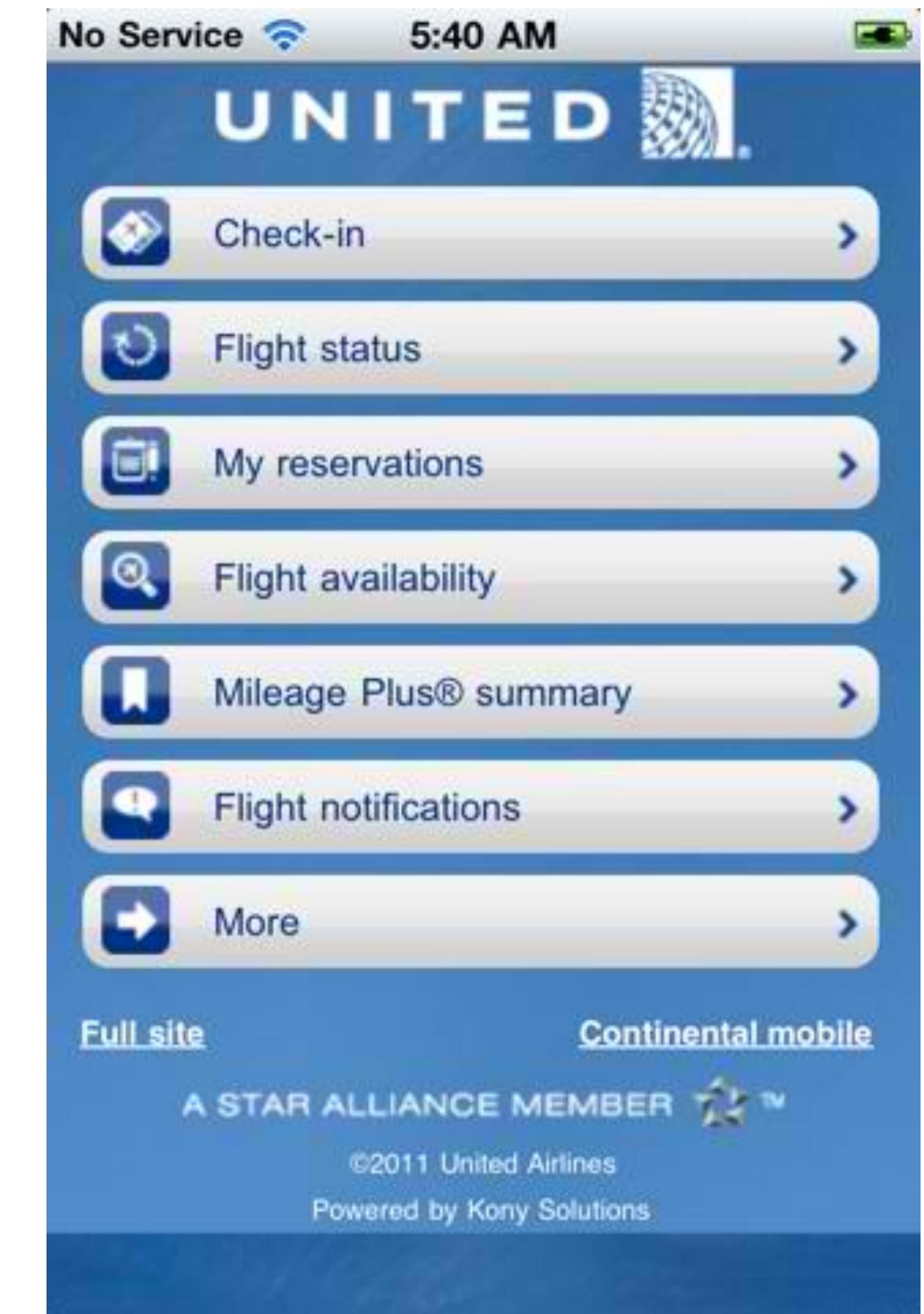
# html5 compatibility 2011

- ▶ iOS 3.0+
- ▶ android 2.0+
- ▶ blackberry smartphones 5.0+
- ▶ blackberry playbook 1.0+
- ▶ webOS 1.4+
- ▶ symbian anna+
- ▶ bada 2.0
- ▶ windows phone mango+
- ▶ ....

# real world sample

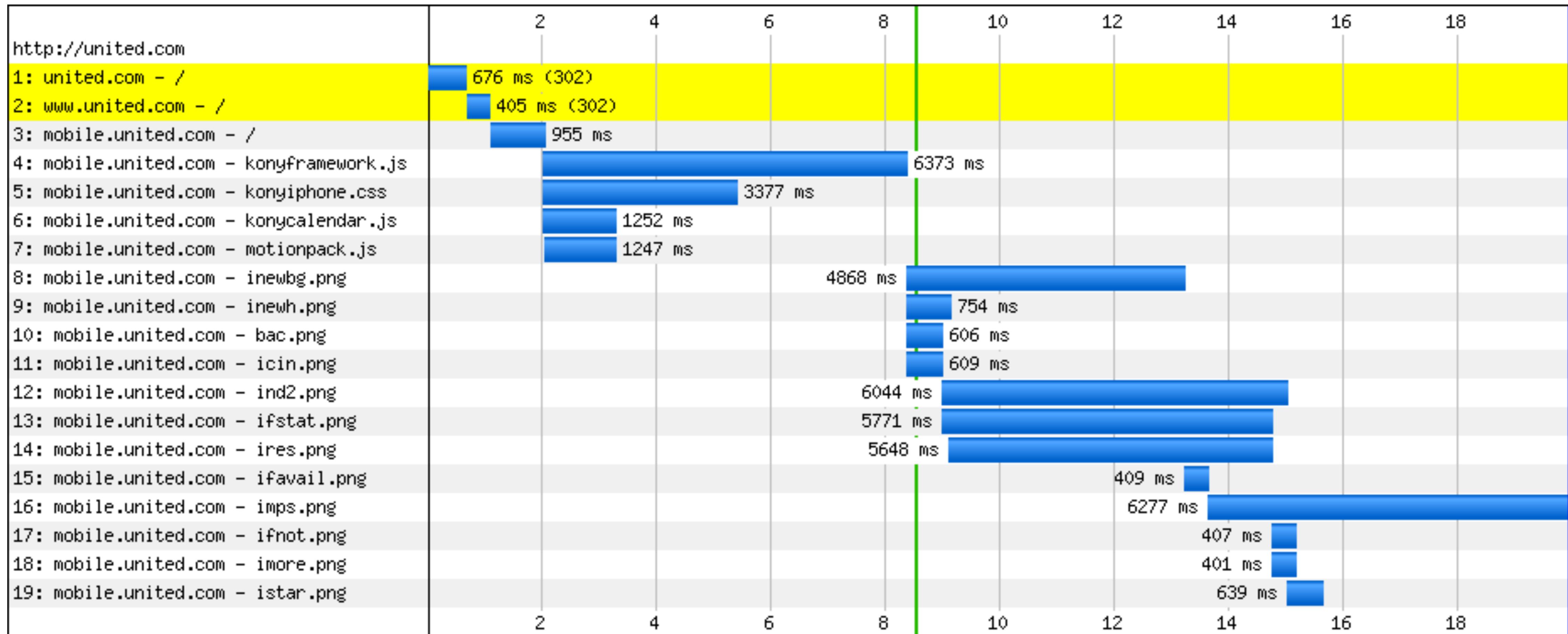
(nothing personal)

united.com



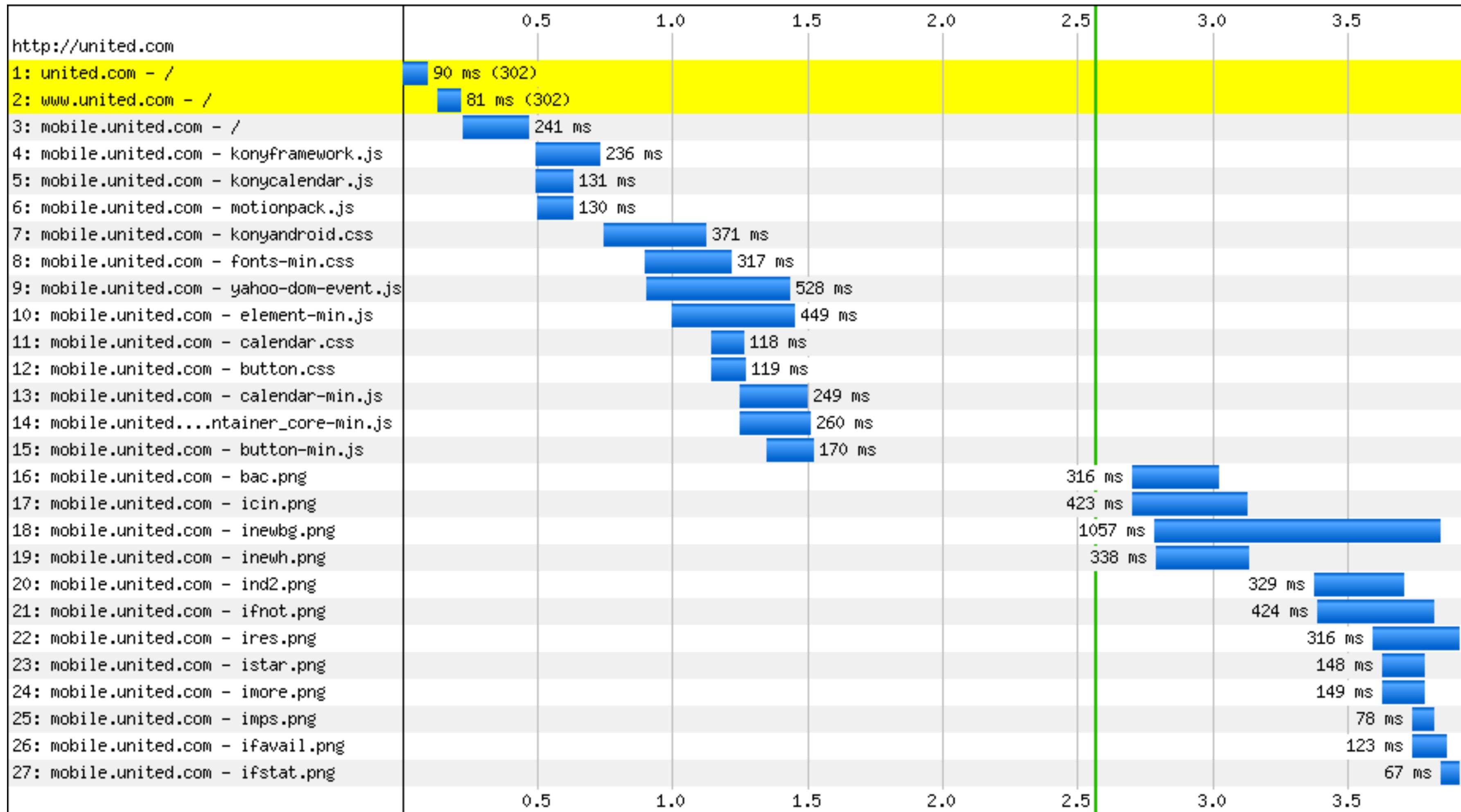
# original iPhone results from blaze.io/mobile

*15-20 seconds to load (over WiFi!) total 602Kb*

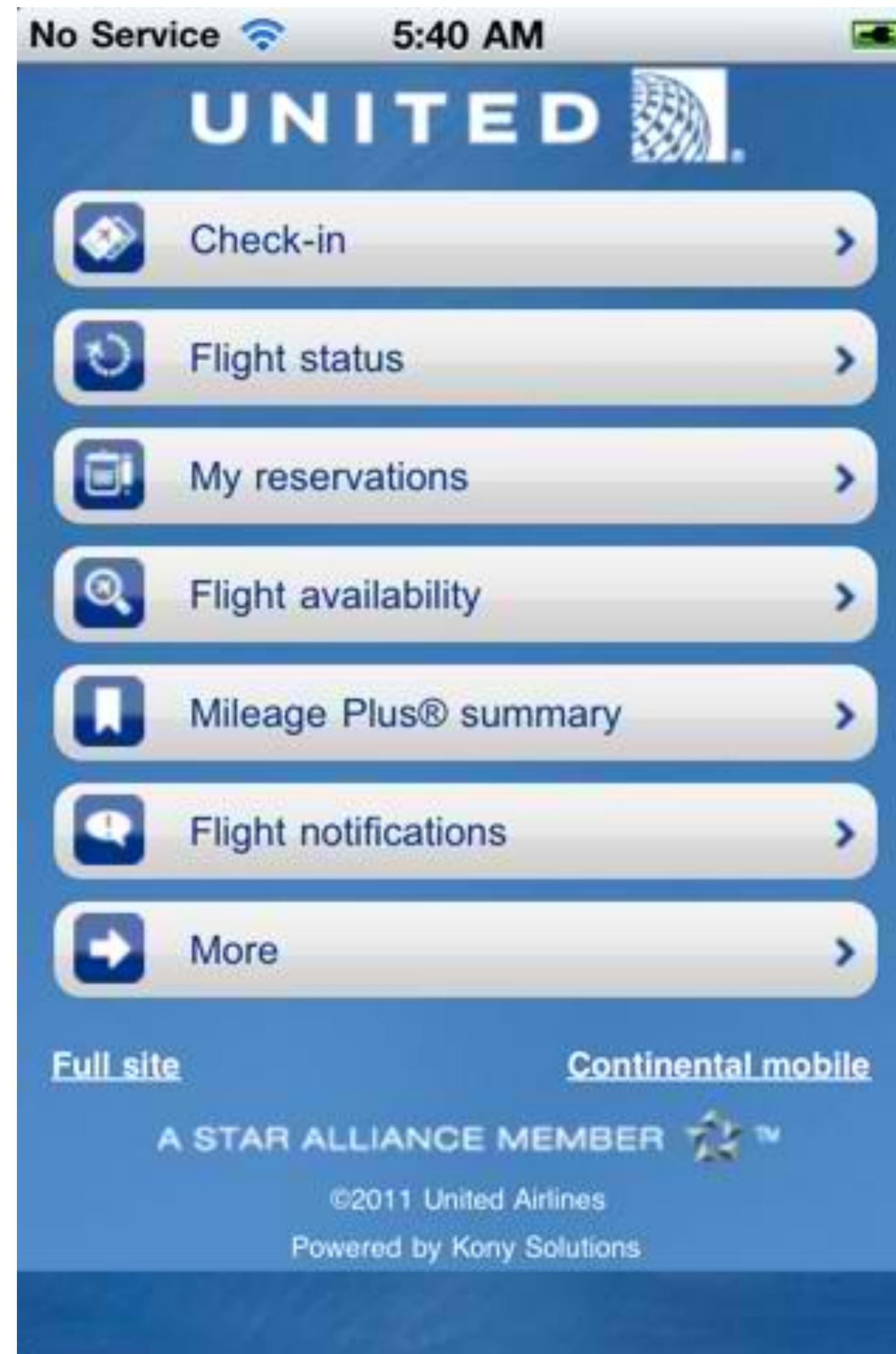


# original Android results from blaze.io/mobile

4-8 seconds to load (over WiFi!) total 442Kb



# 400 / 600Kb? Where are they?



let's start

# 1. be mobile

# 1. be mobile

- ▶ don't leave just a desktop website
- ▶ use mobile meta tags & viewport
- ▶ usability
- ▶ use server-side detection
- ▶ right experience to each context

# 1. be mobile

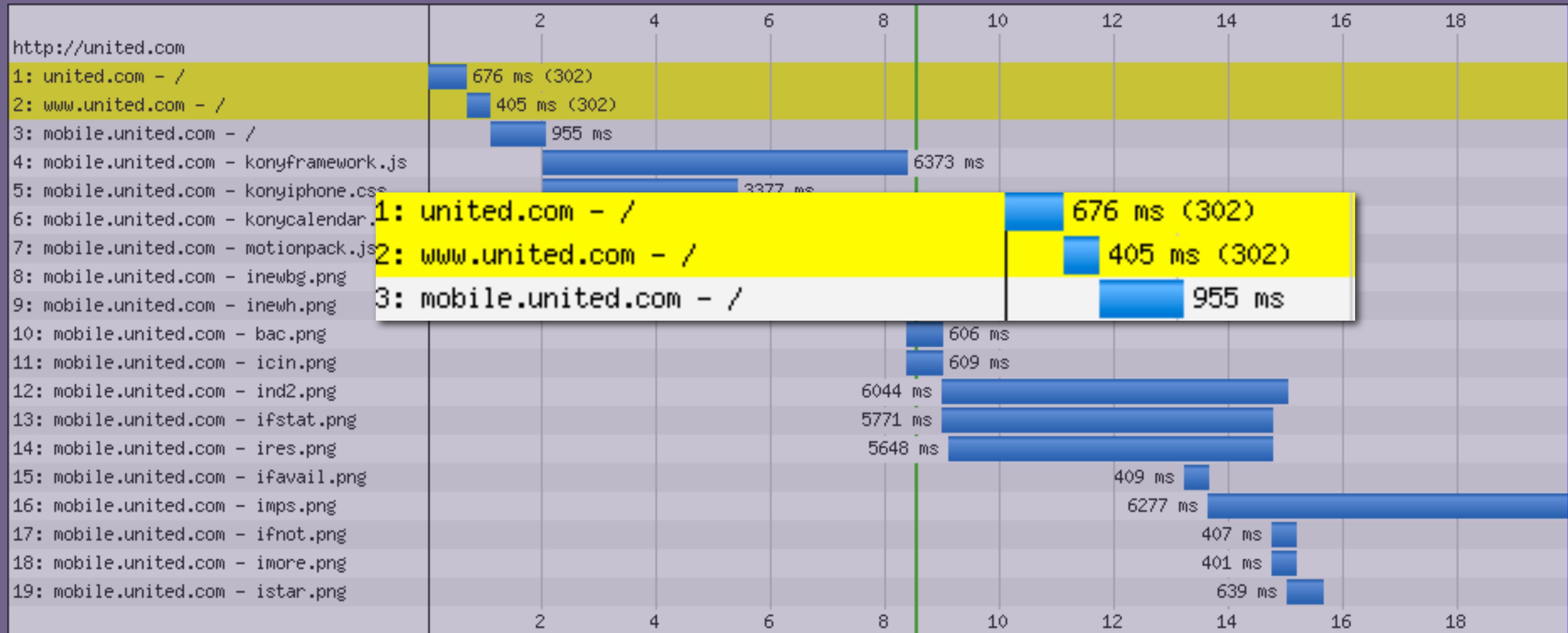


DeviceAtlas™  
MOBILE DEVICE INTELLIGENCE

**scientiamobile**

## 2. don't redirect

## 2. don't redirect



united.com > www.united.com > mobile.united.com

## 2. don't redirect

- ▶ deliver the mobile site directly
- ▶ it's seo-compatible
- ▶ if too difficult: only one redirect and cached

# 3. be simple

# 3. be simple

- ▶ simple, semantic DOM
- ▶ complexity == wrong
- ▶ remember, it's mobile
- ▶ use html5 rules and tags

# 3. be simple

```
</div>
```



# 3. be simple

```
<div class="ktable kbasemargin kwt100"  
konywidgettype = "Ksegment"  
>
```



# 3. be simple

```
<body  
class = "newBg"  
>
```



# 3. be simple

```
<a  
konywidgettype = "Ksegment"  
href="#" eventname="segNewHomeevent_"  
rowid="0">
```



# 3. be simple

Current XHTML:

- 18Kb
- 180 DOM elements
- 145 has class definition (!?)
- 80% has multiple class definition

# 3. be simple

Current CSS:

- 137Kb!
- 6700 lines
- 681 selectors

3. **bossimole**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"><head><meta name="viewport" content="width=device-width, initial-scale=1.0" /><script type="text/javascript" src="http://konycalendar.konycloud.com/konycalendar.js"></script><script type="text/javascript" src="http://iphone/motionpack.js"></script><title>Bossimole</title><input checked="" type="checkbox" name="formid" value="frmNewHome"/><input checked="" type="checkbox" name="cadid" value="frmNewHome"/>
```

The image shows a mobile web page from United Airlines. The header features the word "simple" in a large, bold, sans-serif font, with "be" in a smaller font to its left. Below the header is a navigation bar with links for "HOME", "FLIGHTS", "RESERVATIONS", "MILEAGE", "NOTIFICATIONS", and "MORE". The main content area displays flight information for United Airlines flight UA875 from San Francisco (SFO) to Honolulu (HNL) on a Boeing 777-200ER. It includes sections for "Flight Status", "Flight Availability", "Reservations", "Mileage Plus", and "Flight Notifications". The page uses a clean, modern design with white space and a light blue background.

# 3. be simple

Apply HTML5 semantic tips

- ▶ doctype: <!DOCTYPE html>
- ▶ no xml ns
- ▶ no insane attributes <script type?>
- ▶ no **divitis**: article, section, nav, ...
- ▶ no **classitis**: header, footer, ...

# 3. be simple

```
<!DOCTYPE html>
<html>
<head>...</head>

<body>
<header>
  
</header>
<nav>
  <ul>
    <li><a href='checkin.html'>Check in</a></li>
    <li><a href='flight.html'>Flight status</a></li>
    <li><a href='reservations.html'>My reservations</a></li>
    <li><a href='availability.html'>Flight availability</a></li>
    <li><a href='milleage.html'>Milleage Plus Summary</a></li>
    <li><a href='notifications.html'>Flight Notifications</a></li>
    <li><a href='more.html'>More</a></li>
  </ul>
</nav>

<footer>
  <a href='http://united.com' id='lnkFull'>Full site</a>
  <a href='http://continental.com' id='lnkContinental'>Continental mobile</a>
  
  <h5>©2011 United Airlines</h5>
</footer>
</body></html>
```

# 3. be simple

## Semantic HTML5

- 1.3Kb (7% of original size)
- 31 DOM elements (17% of original qty)
- NO class
- <1Kb CSS
- Same design using CSS.
- If CSS is not there, no design!

# 4. known wpo techniques

# 4. known wpo techniques

- gzip components
- be friend of caching
- stylesheets at the top
- scripts to the bottom (*maybe*)
- ~~scripts and styles external~~
- reduce dns lookups
- minify javascript and css

# 4. known wpo techniques

```
<style type="text/css" media="screen">  
@import "iphone/iphone.css";  
</style>
```



# 4. known wpo techniques

check desktop-like wpo suggestions



Yslow!



Page Speed

# 5. resource != download

# 5. resource != download

Reduce http requests

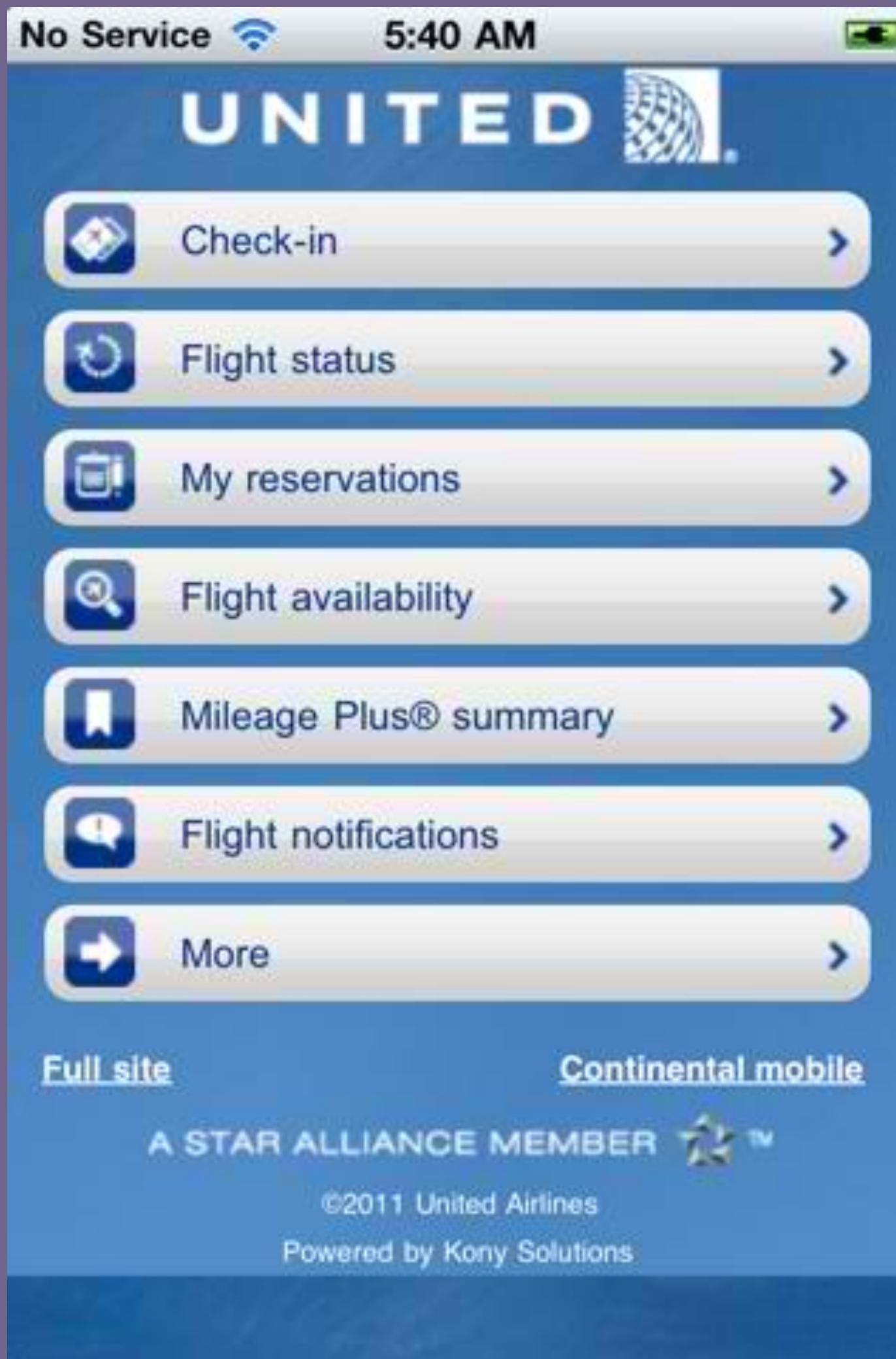
- › every request hurts
- › a lot...
- › more in the mobile space
- › ideal: 1 **only** request initial load
- › even no request!

# 6. images

# 6. images

- ▶ only **semantic** images
- ▶ no **effects**, no **ornaments**
- ▶ use CSS 2.1, CSS 3 or nothing
- ▶ compress images
- ▶ deliver the right image size for every device - context
- ▶ don't think on image=file

# 6. images



- 7 images on icons
- 2 images for logos
- 1 image for arrow
- 1 image for cell background
- 1 background image...
- ...350Kb and 854x854 pixels!!

# 6. images

- ▶ we have 2 **semantic** images: logos
- ▶ **remove** background, icons & other images
- ▶ don't worry! we'll try to provide **alternatives** if the context helps

# 6. images

for the background



a CSS3 gradient

```
background: #1e5799;  
  
background: -moz-linear-gradient(top,  
#1e5799 0%, #2989d8 50%, #7db9e8 100%);  
  
background: -webkit-gradient(linear,  
left top, left bottom, color-stop  
(0%,#1e5799), color-stop(50%,#2989d8),  
color-stop(100%,#7db9e8));
```

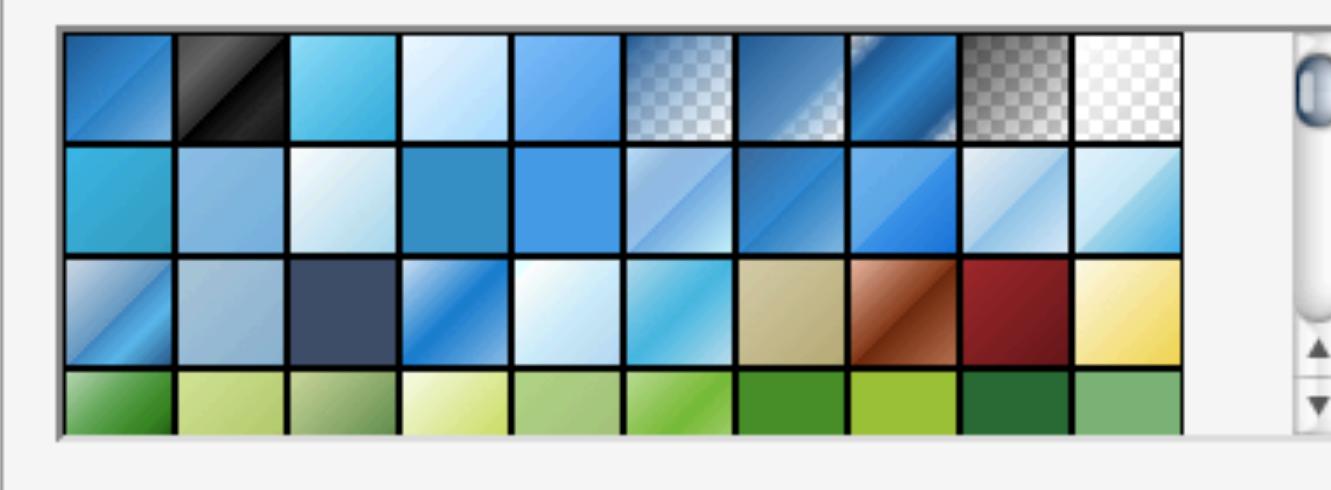
# 6. images

## CSS3 gradientes

<http://www.colorzilla.com/gradient-editor/>

**Ultimate CSS Gradient Generator**  
A powerful Photoshop-like CSS gradient editor from [ColorZilla](#).

Presets



Name:

Stops

Opacity:  Location:  %

Color:  Location:  %

Preview



Orientation:  Size:   IE

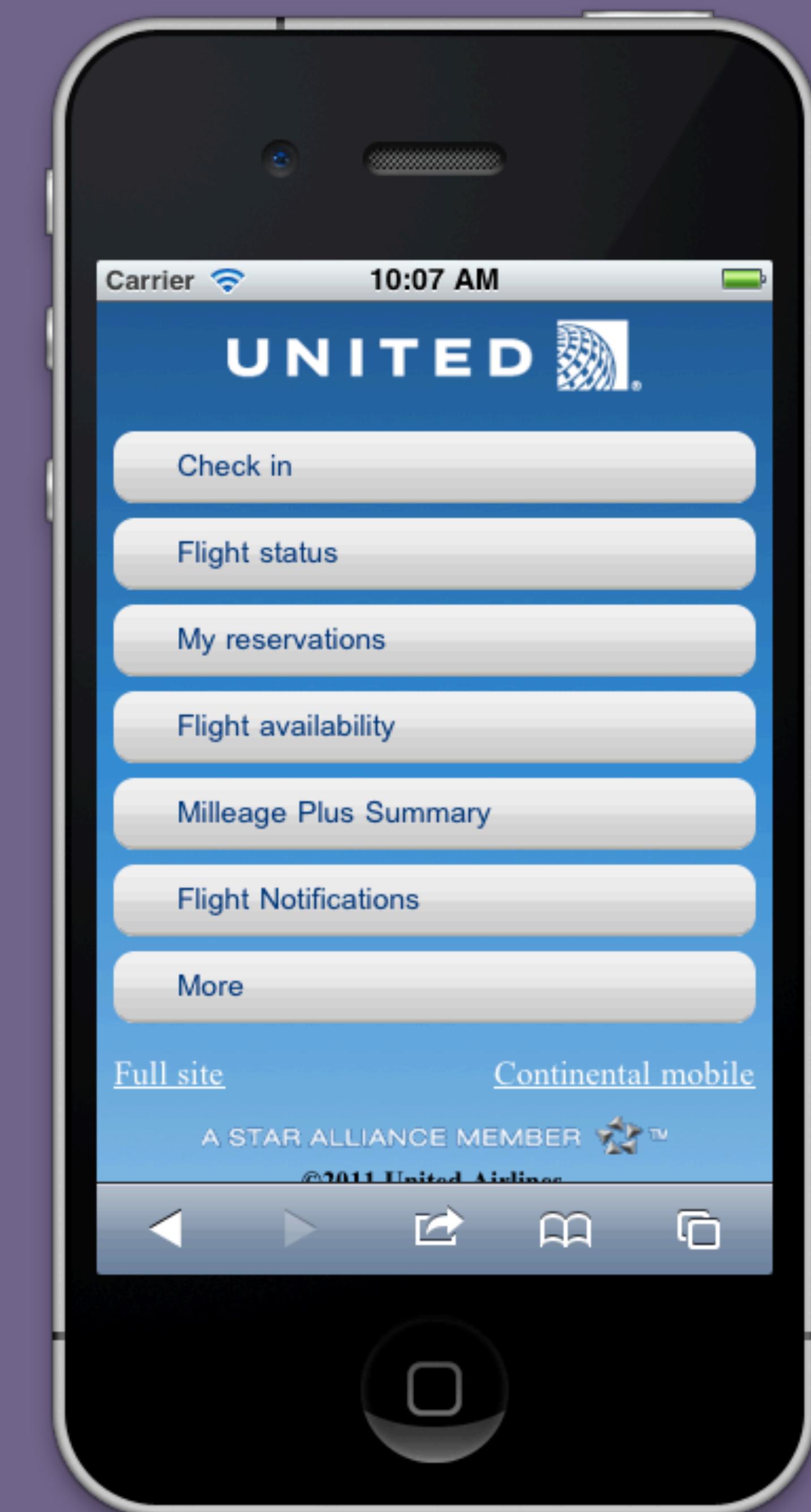
CSS

```
background: #f4f4f4; /* Old browsers */  
background: -moz-linear-gradient(top, #f4f4f4 0%,  
#e9e9e9 46%, #e6e6e6 49%, #dddddd 51%, #d1d1d1  
81%, #cdcdcd 97%, #bab6a9 100%); /* FF3.6+ */  
background: -webkit-gradient(linear, left top,  
left bottom, color-stop(0%,#f4f4f4), color-  
stop(46%,#e9e9e9), color-stop(49%,#e6e6e6),  
color-stop(51%,#dddddd), color-  
stop(81%,#d1d1d1), color-stop(97%,#cdcdcd),  
color-stop(100%,#bab6a9)); /* Chrome,Safari4+ */  
  
background: -webkit-linear-gradient(top, #f4f4f4  
0%,#e9e9e9 46%,#e6e6e6 49%,#dddddd 51%,#d1d1d1  
81%,#cdcdcd 97%,#bab6a9 100%); /*  
Chrome10+,Safari5.1+ */  
  
background: -o-linear-gradient(top, #f4f4f4  
0%,#e9e9e9 46%,#e6e6e6 49%,#dddddd 51%,#d1d1d1  
81%,#cdcdcd 97%,#bab6a9 100%); /* Opera11.10+ */
```

# 6. images

for the cell background

...a CSS3 gradient too



# 6. logos & icons

We can use CSS sprites

- › very good support on mobile
- › use a device library to detect
- › lot of free services: [spriteme.org](http://spriteme.org), [css-sprit.es](http://css-sprit.es)

# 6. logos & icons

We can use inline images (aka data URI)

- ▶ very good support on mobile
- ▶ use a device library to detect
- ▶ first, compress!
- ▶ lot of free online services
- ▶ just convert bytes into base64
- ▶ easy to convert from server-side

base64\_encode() in PHP

# 6. logos & icons

We can use inline images (aka data URI)

**data:<mime-type>;base64,<base64 data>**

```
data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAMUAAAAiAgMAAACSF/VZAAAAA3NCSVQICAjb4U/gAAAAACVBMVEX//////+ZmZIVZlogAAAAA3RSTIP//wDXygIBAAAAACXBWXMAAAsSAAALEgHS3X78AAAAHnRFWHRTb2Z0d2FyZQBBZG9iZSBGaXJld29ya3MgQIMILjGrH0jrAAAA9kIEQVQ4je3TOw6DMAwGYMsT4hSMKKf0UTKinLJ+JE AeLmmnDoIaELQf+e0ESHMD8sBE8Ce/SQhThJD4dPBJviQ/8U0eQf/qEfDJ6pHgEsxkacnqEkkGOJgFfRJklq0nUlgd5KY+XTJrsNiT/SLrnUS5zKU0ZGmlIgIoi0QWrSbwjnAG+dRL+UAGTX4gmmJrCHkkl8+xNjQCmcSWVEImd8Iiy0u8kt0sIVLeVr+Q5JNmW8aTkEd0WxIXg0YO2925C0OyGIkKSZbI9oqVw0m0DdsVjB8nEzNJNCahl/PvvnQK6ROCSnLQGaIASs9ng3IBIs/JbrW8AJuDnJ+8sV3sAAAAAEIFTkSuQmCC
```



# 6. logos & icons

inline images are just **GREAT**

- ▶ new way to think about images
- ▶ new way to transmit & store images
- ▶ they are just plain text
- ▶ can be used on HTML or CSS

# 6. logos & icons

you can create them **on the fly** with html5

- ▶ (check compatibility)
- ▶ use with html5 canvas drawing API
- ▶ use **toDataURL()** method from canvas
- ▶ voilá!
- ▶ even useful for converting image real files  
into data uri client-side

# 6. other tips

- ▶ 450 emoji characters on iOS

😊 &#xe415;	😊 &#xe056;	😊 &#xe057;	😊 &#xe414;
😔 &#xe417;	😔 &#xe40d;	😔 &#xe40a;	😔 &#xe404;
😉 &#xe402;	😉 &#xe108;	😉 &#xe403;	😉 &#xe058;
😢 &#xe40b;	😢 &#xe406;	😢 &#xe413;	😢 &#xe411;
😡 &#xe059;	😡 &#xe416;	😡 &#xe408;	😡 &#xe40c;
❤️ &#xe32a;	❤️ &#xe32d;	❤️ &#xe328;	❤️ &#xe32b;
💕 &#xe329;	✨ &#xe32e;	⭐ &#xe32f;	⭐ &#xe335;
❓ &#xe020;	❓ &#xe336;	ＺＺ &#xe13c;	👎 &#xe330;
🔥 &#xe11d;	💩 &#xe05a;	👍 &#xe00e;	👎 &#xe421;
✌️ &#xe011;	👉 &#xe41e;	👉 &#xe012;	👉 &#xe422;
👉 &#xe230;	👉 &#xe427;	👉 &#xe41d;	👉 &#xe00f;
🏃 &#xe115;	👫 &#xe428;	👳 &#xe51f;	👳 &#xe429;
🧞 &#xe426;	💏 &#xe111;	💏 &#xe425;	🧞 &#xe31e;
👲 &#xe002;	👲 &#xe005;	👲 &#xe004;	👲 &#xe51a;
👲 &#xe516;	👲 &#xe517;	👲 &#xe51b;	👲 &#xe152;
💀 &#xe11c;	👅 &#xe536;	👅 &#xe003;	👅 &#xe41c;

&#xe333;



<http://pukupi.com/post/1964>

# 6. other tips

- ▶ CSS3 multiple backgrounds
- ▶ CSS3 transformations
- ▶ CSS3 shadows, effects
- ▶ CSS3 rounded borders, border image
- ▶ SVG, inline SVG and canvas
- ▶ if not compatible, **accept it!**

7. be ready asap

# 7. be ready asap

- › defer most of your code after **onload**
- › defer content
- › don't even parse frameworks
- › did you say frameworks?
- › remove address bar **onload**

# 7. defer, defer

- great support for deferring content & code
- if you have lot of content, use your own loading
- use a server-side detection

# 7. don't parse yet!

- › parsing javascript takes time
- › from 1ms to 100ms per 1Kb
- › delays **onload**

# 7. don't parse yet!

- › divide your code into modules by needs
- › comment all your code  
`<script>/* ... */</script>`
- › when needed, remove comments and eval  
`eval(stripOutCommentBlock(script.innerHTML));`
- › created by Gmail Mobile team and Charles Jolley

# 7. forget about frameworks

- ▶ ok, not always
- ▶ think it 100 times
- ▶ do you really need it?
- ▶ don't use jQuery just for `$("")`
- ▶ jQuery can take up to 8s to parse on some phones

# 7. forget about frameworks

- ▶ W3C Selectors API [w3.org/TR/selectors-api](http://w3.org/TR/selectors-api)  
**document.querySelector!**
- ▶ create your own mini-library
- ▶ use mini-frameworks: XUI, zepto.js, microjs
- ▶ jQuery Mobile is not a mini-framework! it's a ui framework.
- ▶ if it's not compatible, do you **really** need it?

# 7. forget about frameworks

our sample uses three uncompressed  
javascript files:

framework: 60Kb

motionpack: 3Kb

calendar: 7Kb

# 7. remove address bar

- ▶ hack for some browsers
- ▶ scroll to 0, 1 after onload
- ▶ you need to have a **min-height** container of the available height - device library
- ▶ **window.scrollTo(0, 1);**
- ▶ the user will be happy to start using your web immediatly

# 8. application cache

# 8. application cache

- ▶ Defines an offline installation package  
`<html manifest='offline.appcache'>`
- ▶ Next time, it will be available **immediately** (even without connection)
- ▶ **Update process**
- ▶ Make the HTML and some basic resources cached and manage your own cache with **localStorage**

# 9. offline storage

# 9. offline storage

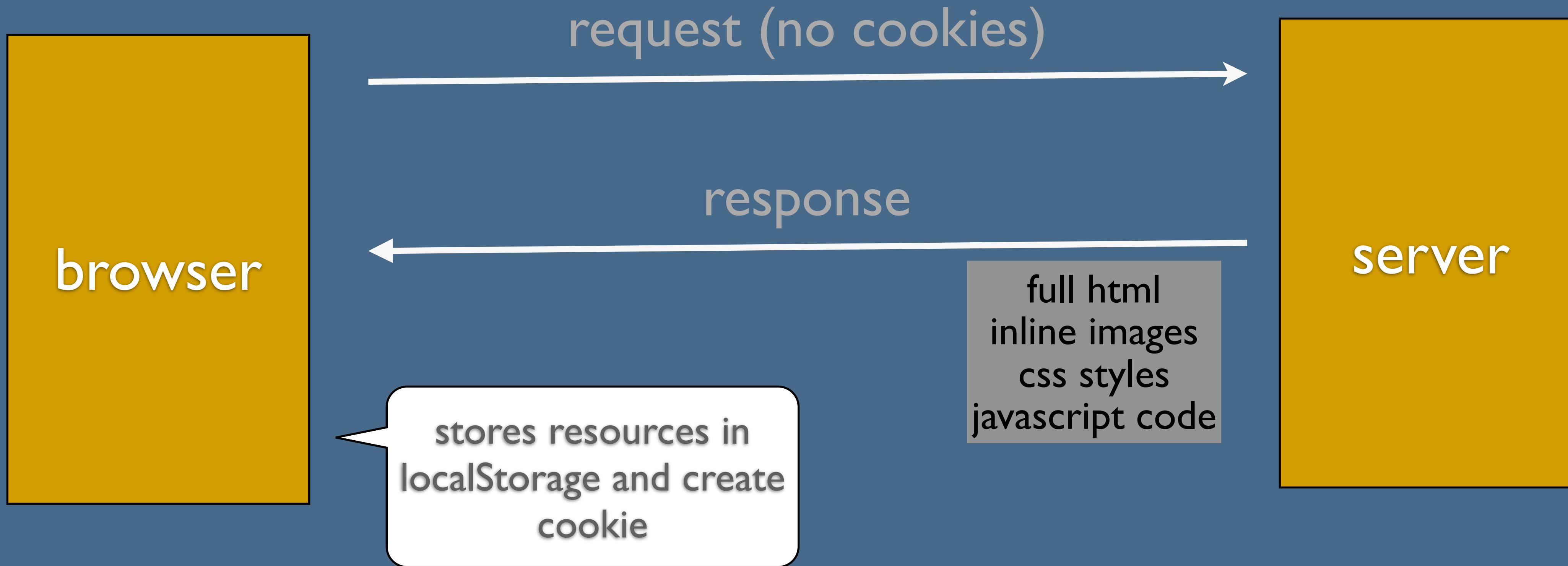
- ▶ localStorage and SQL storage
- ▶ localStorage easier and faster
- ▶ store strings (2x faster than objects)
- ▶ what to store?
  - ▶ images
  - ▶ code
  - ▶ styles
  - ▶ html elements
  - ▶ data

# 9. offline storage

- ▶ we can mix it with cookies
- ▶ resource storage library
- ▶ we store some resources client-side
- ▶ next time, we don't send them
- ▶ on mobile, up to 2Mb per domain is safe

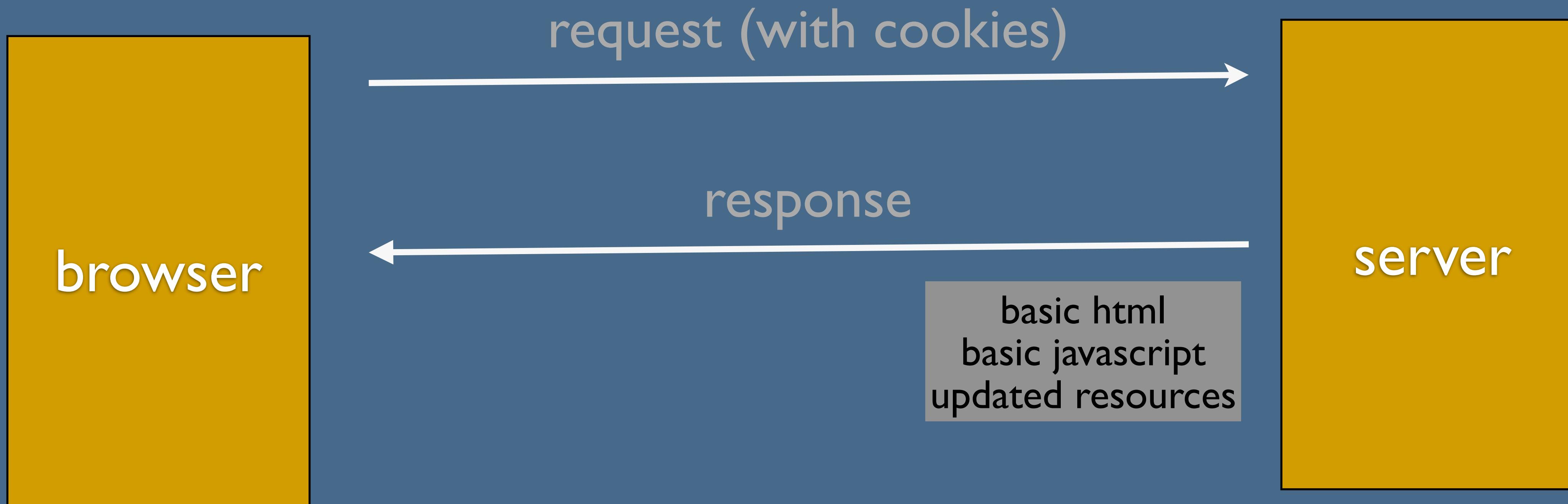
# 9. offline storage

first load



# 9. offline storage

second load



# 10. internal is better

# 10. internal is better

- ▶ code and resources used only once must be inlined in the HTML
- ▶ no external css, no external javascript, no external images
- ▶ if you use it on different pages, then:
  - ▶ external
  - ▶ localStorage

# 11. touch, no click

# 11. touch, no click

- ▶ on **touch** devices, click delays between 300 and 500 ms before executing.
- ▶ Too much!
- ▶ Use touch events: **ontouchend**
- ▶ We can use progressive enhancement and change links behavior

# 12. ajax please!

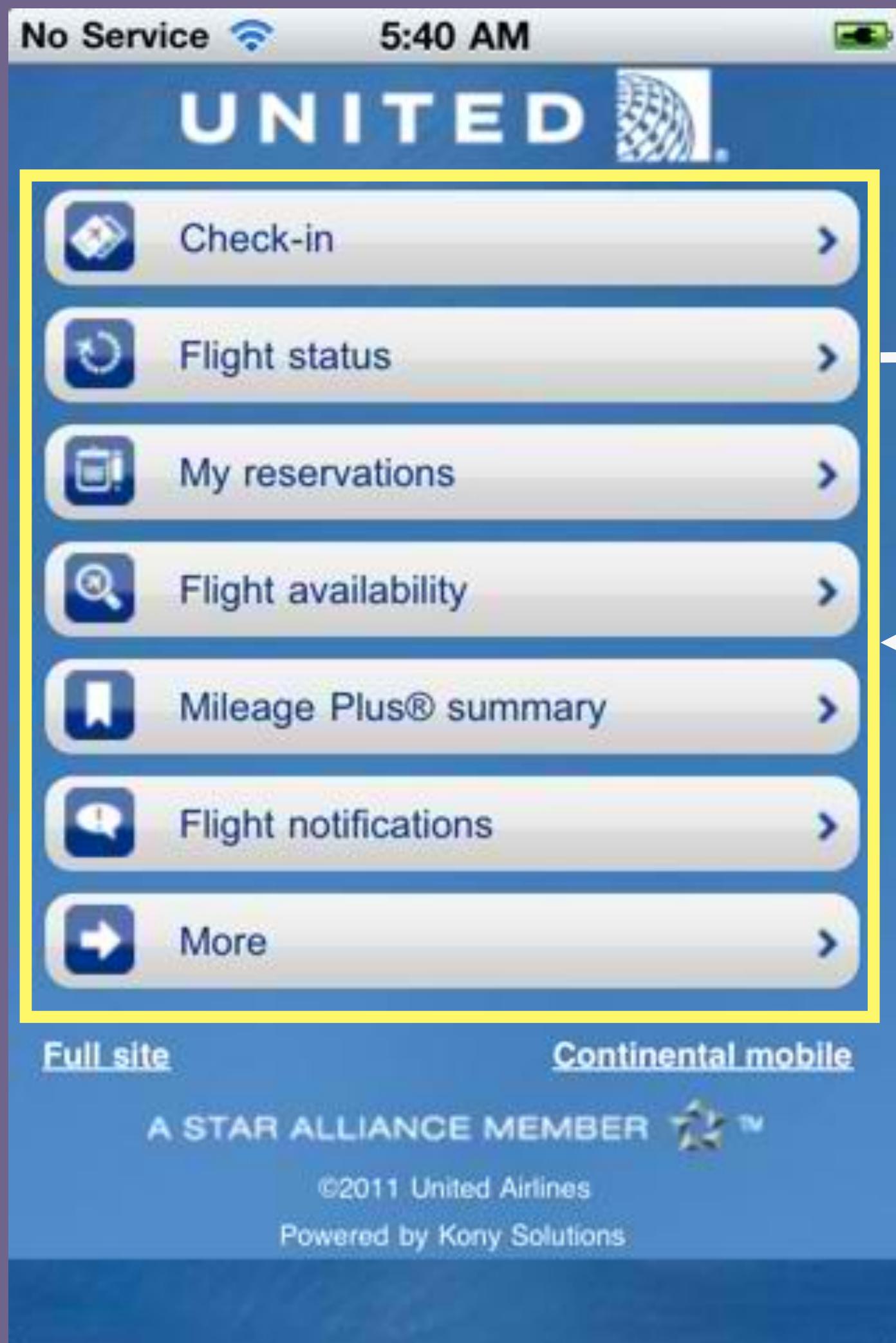
# 12. ajax please!

- ▶ progressive enhancement
- ▶ server-side framework
- ▶ it's seo-friendly too!
- ▶ most devices support XHR
- ▶ why don't use it!
- ▶ it's mobile-friendly!

# 12. ajax please!

- **onhashchange** for history management
- request only what needs to be changed
- **html > json > xml**
- with json, **JSON.parse** is up to 2x faster than eval

# 12. ajax please!



server

# 13. upgrading ajax

# 13. more than ajax

## html5 server-sent events

<http://dev.w3.org/html5/eventsource/>

```
var source = new EventSource('updates.cgi');
source.onmessage = function (event) {
    alert(event.data);
};
```

# 13. more than ajax

## html5 web sockets

<http://dev.w3.org/html5/websockets/>



# 14. context & user decision

forget about  
**DECIDING FOR THE USER**

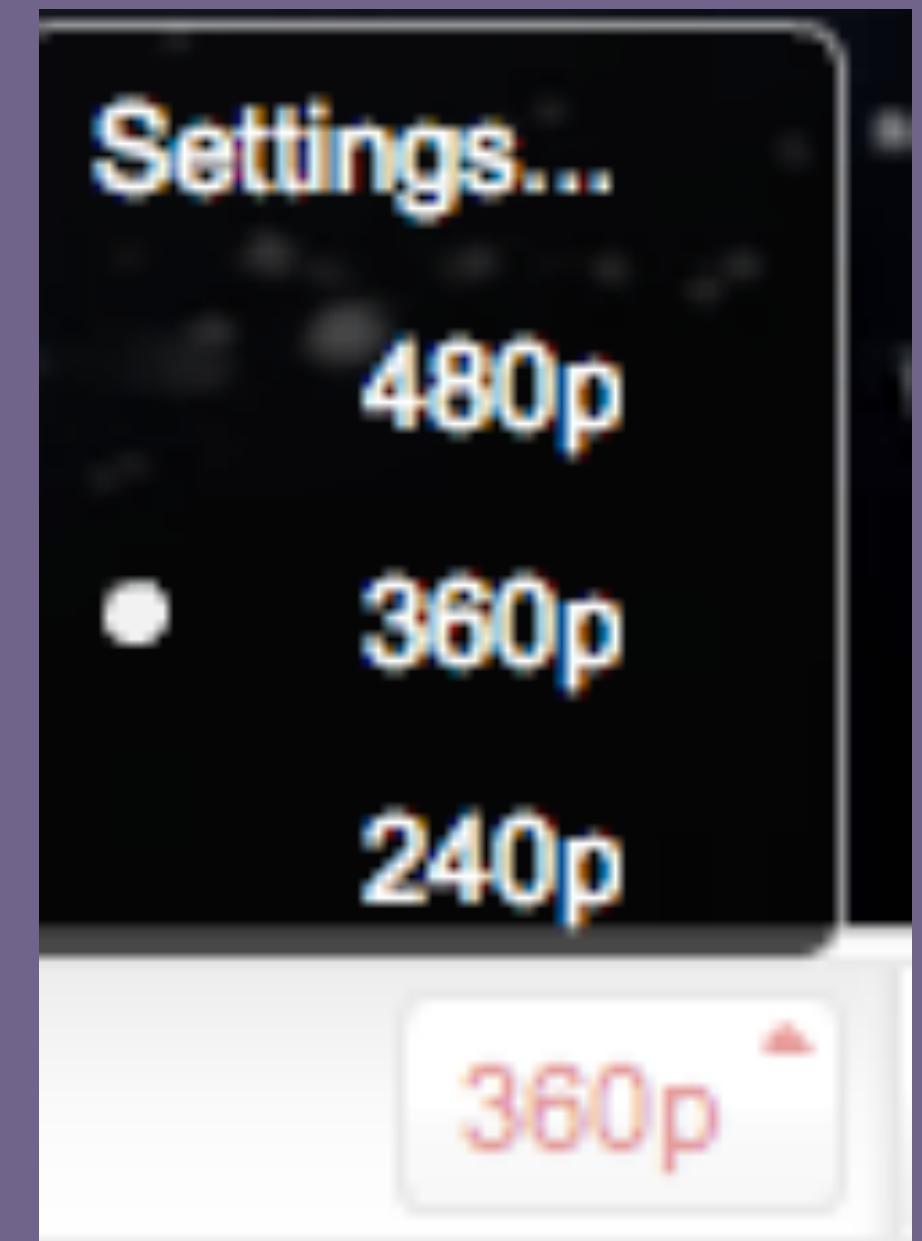
# 14. context & user decision

- › now, **you** decide the user experience based on the device or based on... I don't know.
- › if you have an iPhone or Android, this is the experience you want.
- › if you have a feature phone, then this is what you need

# WRONG

# 14. context & user decision

- ▶ provide different experiences
- ▶ YouTube resolution idea
- ▶ SD/HD version
- ▶ decide best version based on context
- ▶ let the user change the decision



how?

# 14. detecting connection

## W3C Network Information API

- ▶ it's client-side
- ▶ works on Android 2.2+
- ▶ `navigator.connection.type`
- ▶ can be `WIFI`, `CELL_2G`, `CELL_3G`, others

```
if (navigator.connection.type==navigator.connection.WIFI) {  
}
```

# 14. detecting connection

## BlackBerry client-side API

- ▶ works on BB smartphones 4.0+
- ▶ `blackberry.network==‘Wi-Fi’`

# 14. detecting connection

## Server-side detection

- ▶ Symbian header x-nokia-musicshop-bearer
- ▶ BlackBerry header via
- ▶ Trying to detect if it's a carrier network
- ▶ Massive's Operator Identification Platform  
[\(www.werwar.com\)](http://www.werwar.com)

# 14. detecting connection

## iOS detection

- ▶ No official way (there are some hacks)
- ▶ If you are an iOS developer, go to [bugrequest.apple.com](http://bugrequest.apple.com) and request this feature

# 14. detecting pixel density

don't download pixels that can not be seen

- ▶ iPhone 3 vs iPhone 4
- ▶ android
- ▶ on WebKit: `window.devicePixelRatio`
- ▶ can be used also in css3 media queries
- ▶ remember: let the user decide!

# 14. html5 audio & video

The standard only cares about codecs/formats

- › You should care about connectivity
- › iOS: Quicktime Reference Movies
- › A reference movie decides different sources regarding the current bandwidth
- › Doesn't update dynamically
- › Search for free MakeRefMovie tool

# 15. animations & timers

# 15. animations & timers

## Browsers animations or nothing

- ▶ ok, it was a bit extreme...
- ▶ don't use javascript based animation
- ▶ use css3 transitions & animations
- ▶ they can be hardware-accelerated
- ▶ move & scale using css3 transforms

# 15. animations & timers

## Timers

- ▶ be careful
- ▶ sluggish UI
- ▶  $\geq 1\text{s}$  frequency
- ▶ reduce DOM changes
- ▶ change the DOM once per cycle

# 15. animations & timers

## html5 web workers

- ▶ starting to appear on mobile
- ▶ allow different threads on javascript
- ▶ we can do things without affecting ui rendering (too much)
- ▶ use it if they are available!

# 16. mobilize

# 16. mobilize

- › speed is also about total time to do an action
- › reduce input, use html5 input types
- › use context and html5 to help the user reduce times
- › use w3c geolocation api
- › use localStorage for history and analytics

# GOVERNMENT CENTER

JND

TO B.C., CLEVELAND CIRCLE, RIVERSIDE, HEATH VIA PARK



NO SMOKING

DON'T BE  
AVERAGE.

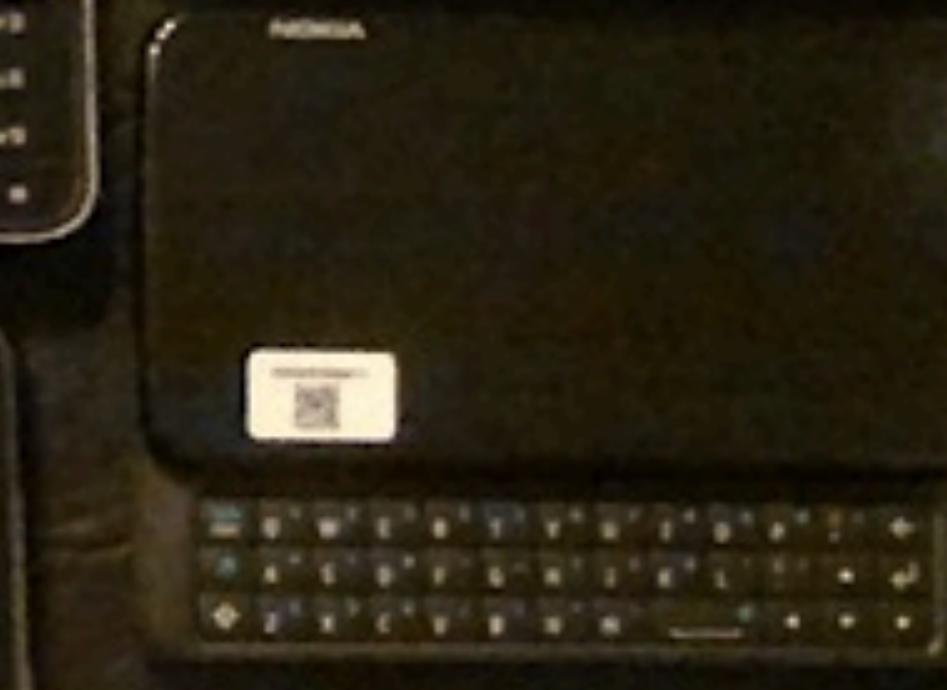
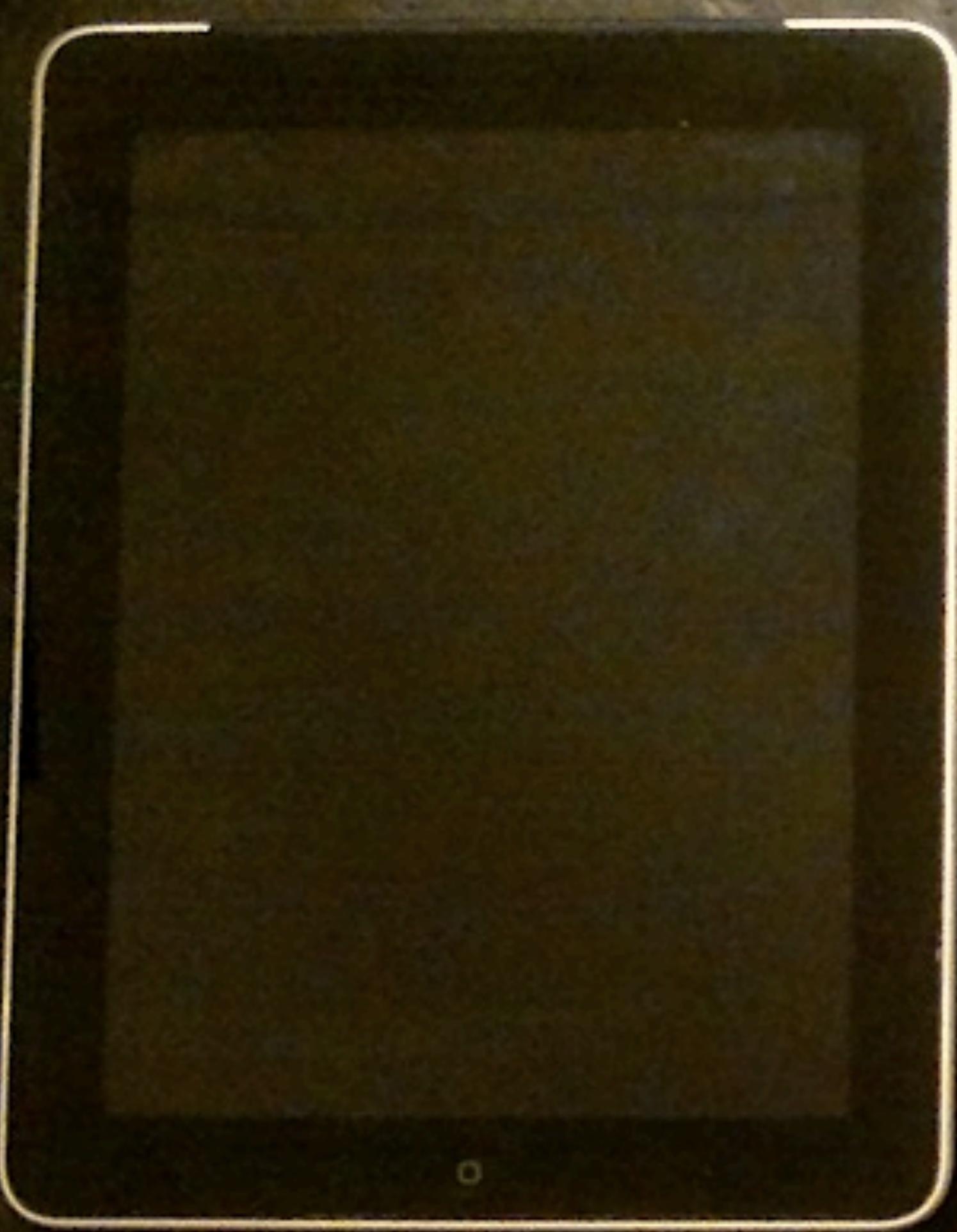


GOOD BANKING IS GOOD BUSINESS. Citizens Bank

use good practices



test, test & test



you can reach a good experience



# thank you!

firt.mobi  
twitter: @firt  
[www.mobilexweb.com](http://www.mobilexweb.com)