Nathaniel T. Schutta
@ntschutta

Back in the day…

It used to be so easy!

We would argue over server side frameworks.

Then ask a simple question.

What is the
default resolution?

And off we went.

Of course it wasn't quite that simple was it?

We did have browsers to deal with.

Or other client side issues.

But the target didn’t move.

We live in a different world today though.

An unprecedented number of client devices.

What do we do about that?

Mobile is kind of a big deal.

It used to be so simple…

What resolution?

What browser?

What widget library?

OK, we'd passionately discuss some things…

Today is different.

It isn't about the desktop anymore.

Desktop is “legacy”.

Now it is all about mobile.

Smartphones and tablets…

60 percent of digital media time spent in the U.S.

http://www.comscore.com/Insights/Presentations-and-Whitepapers/2014/The-US-Mobile-App-Report

Bit of growth there.

Android activations outpacing baby births.

http://www.telegraph.co.uk/technology/ces/9013487/CES-2012-Android-activations-outpacing-baby-births.html

Luke Wroblewski
@lukew

Follow

There are more iPhones sold per day (402k) than people born in the World per day (300k). twitter.com/#!/asymco/stat...

Reply Retweet Favorite ••• More

RETWEETS
396

FAVORITES
52

http://news.cnet.com/8301-17938_105-57365767-1/apple-makes-more-iphones-than-humans-make-babies/

https://twitter.com/#!/lukew/statuses/161943568024469504

That's soooo last year.

About 1.5 million Android devices activated daily.

http://news.yahoo.com/daily-android-activations-grow-1-5-million-google-0415524 61.html

Last holiday quarter, Apple sold 34,000 phones…

An hour.

http://www.dailymail.co.uk/sciencetech/article-2928981/Record-iPhone-sales-drive-blowout-quarter-Apple.html

Fourth quarter
showed record sales.

http://appleinsider.com/articles/15/10/27/apple-earns-record-515b-on-sales-of-48m-iphones-57m-macs

More than 300 million smart phones sold every quarter.

http://ben-evans.com/benedictevans/2014/5/28/beats-content-is-king

About 1.5 billion smart phones in the world today.

http://ben-evans.com/benedictevans/2014/5/28/beats-content-is-king

800 million iTunes accounts.

http://www.forbes.com/sites/nigamarora/2014/04/24/seeds-of-apples-new-growth-in-mobile-payments-800-million-itune-accounts/

More than 1 billion iOS devices in the world.

http://www.dailymail.co.uk/sciencetech/article-2928981/Record-iPhone-sales-drive-blowout-quarter-Apple.html

1.4 billion active Android users every month.

http://techcrunch.com/2015/09/29/android-now-has-1-4bn-30-day-active-devices-globally/

McDonalds? 23 years to sell 1 billion hamburgers.

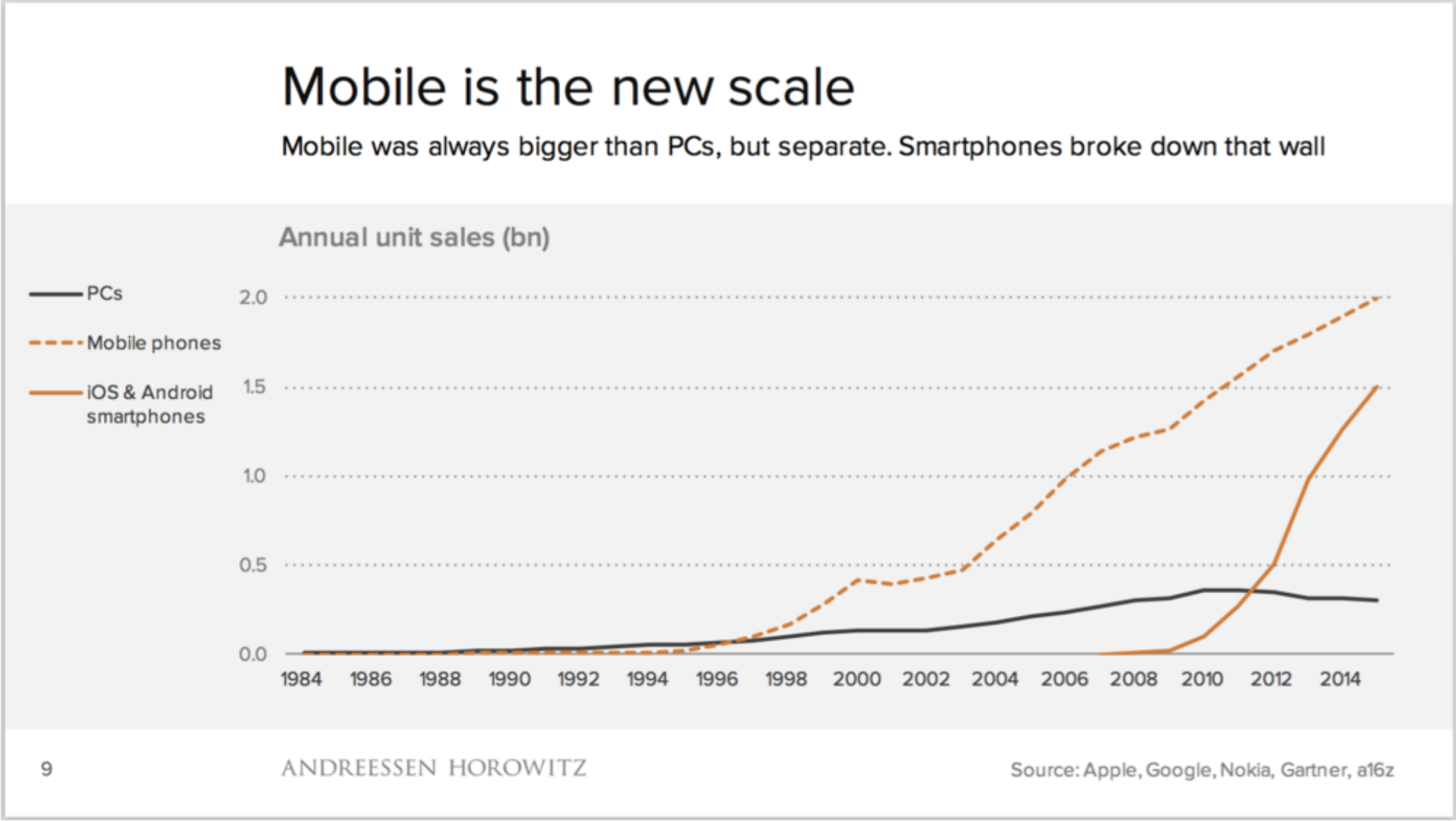
To put that in perspective...

1 billion Windows PCs.

2.2 billion Internet users.

http://www.lukew.com/ff/entry.asp?1626

The scale is hard to imagine.



http://ben-evans.com/benedictevans/2015/11/7/mobile-ecosystems-and-the-death-of-pcs

2008, Windows market share was 95%...

By the end of 2012?
Try 30%.

http://www.computerworld.com/s/article/9232778/_2013_is_going_to_be_ugly_for_Microsoft_as_it_shifts_to_Windows_8_Forrester_predicts

Now?

Ben Thompson
@monkbent

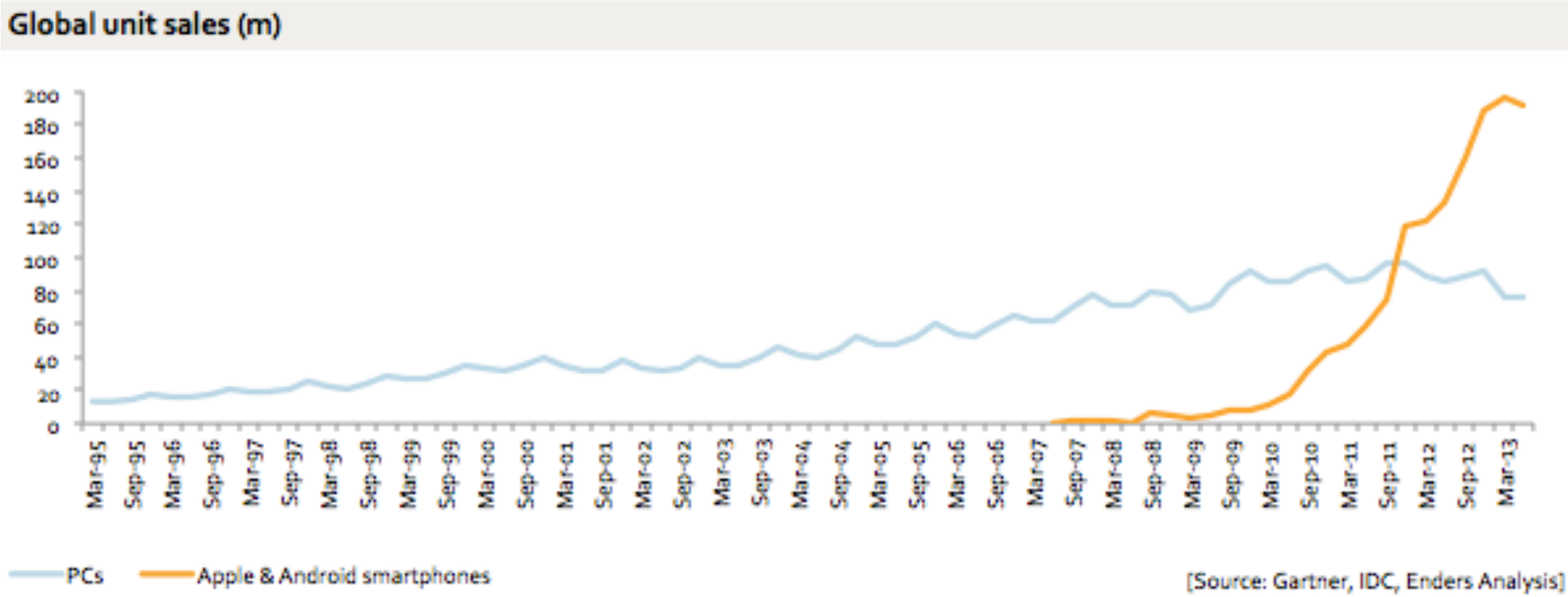
Follow

And people say nothing has changed at Microsoft. This is their own slide! pic.twitter.com/orH6xteUF4

Reply Retweet Favorite More

https://twitter.com/monkbent/statuses/488990398183116800

To put that into perspective...



http://ben-evans.com/benedictevans/2013/8/23/pcs-and-smartphones-the-long-view

App stores are booming.

Apple - more than 100 billion downloads.

http://www.theverge.com/2015/6/8/8739611/apple-wwdc-2015-stats-update

Somewhere north of 150 billion downloads total.

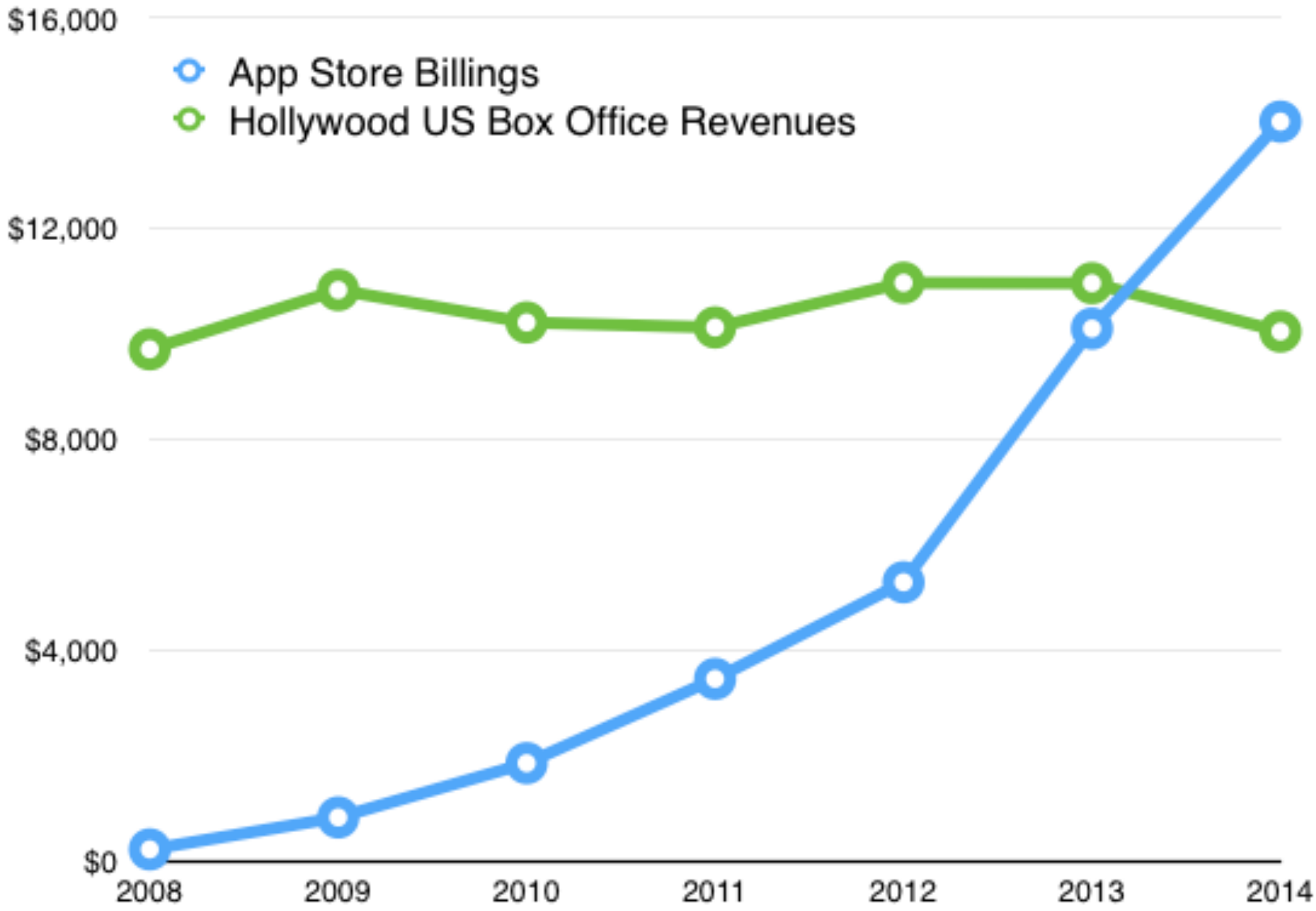
Monthly downloads? 9x McDonalds burger sales.

Apps are a big deal.

$25 billion paid out...

http://www.asymco.com/2015/01/22/bigger-than-hollywood/

iOS app developers earned more than Hollywood.



http://www.asymco.com/2015/01/22/bigger-than-hollywood/

78% of US Facebook users are mobile.

http://techcrunch.com/2013/08/13/facebook-mobile-user-count/

Half a billion Facebook users are mobile only.

http://techcrunch.com/2015/01/28/facebook-mobile-only-2/

Nearly 60% of Amazon's holiday shopping? Mobile.

http://techcrunch.com/2014/12/26/amazon-2014-holiday-sales/

When will the tablet market be bigger than PCs?

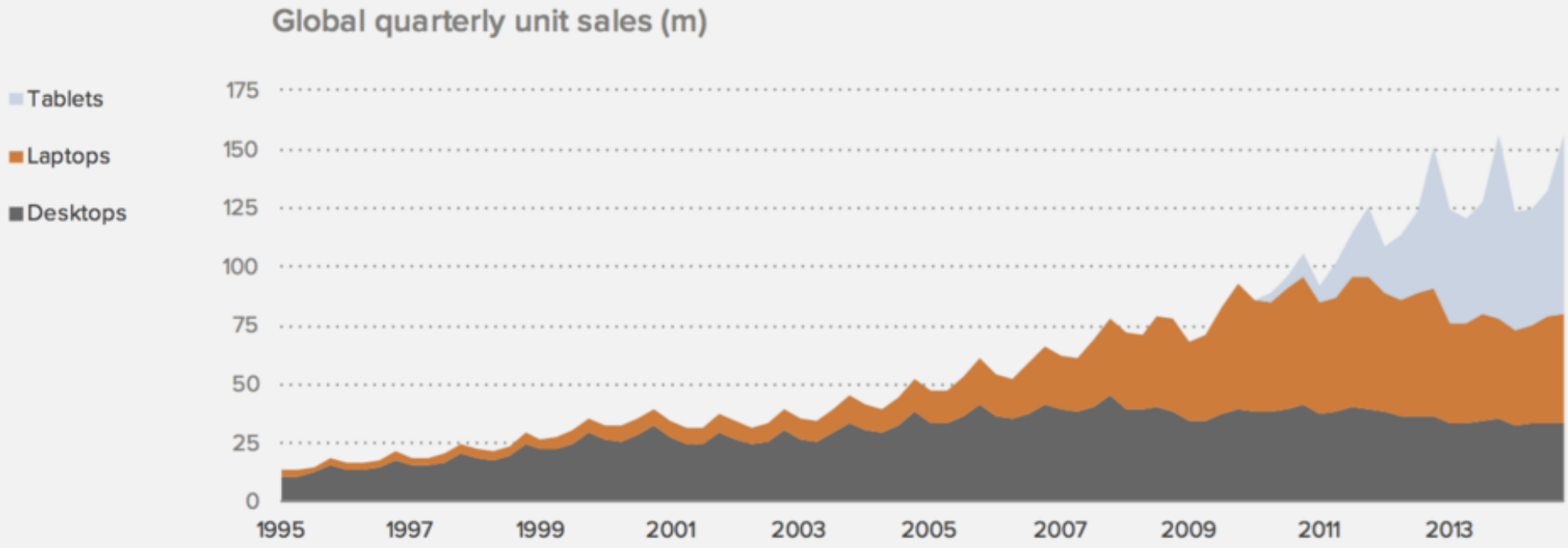
Fall of 2013...

http://www.asymco.com/2012/03/02/when-will-the-tablet-market-be-larger-than-the-pc-market/

Roughly a third of the market now.

Are tablets ‘mobile’, or are they ‘PCs’?

For your large screen, do you want a desktop, a laptop or a tablet? ‘Tablets’ have taken a third of that market



ANDREESSEN HOROWITZ

Source: IDC, Gartner, Barclays Capital

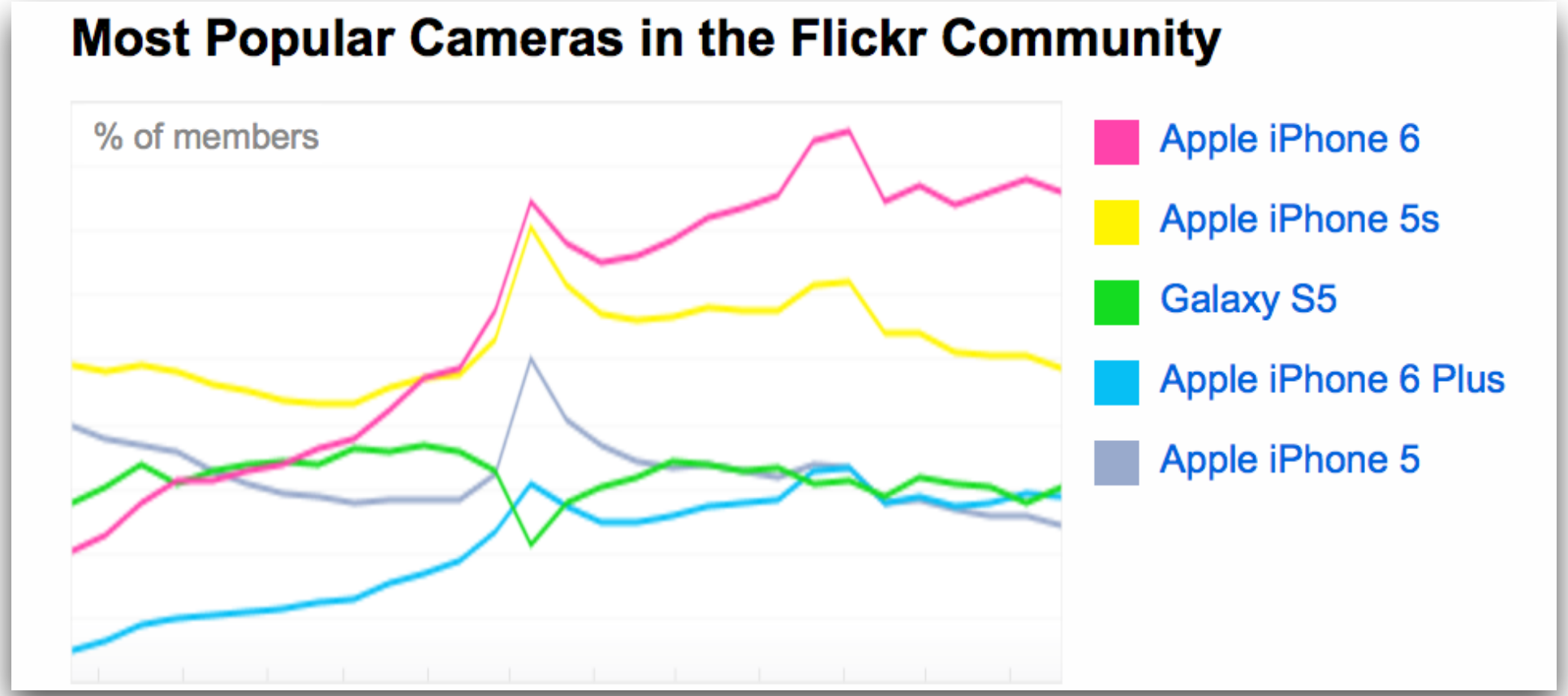
http://ben-evans.com/benedictevans/2015/11/7/mobile-ecosystems-and-the-death-of-pcs

Half of all American adults own a tablet or e-reader.

http://www.pewinternet.org/2014/01/16/e-reading-rises-as-device-ownership-jumps/

Most popular camera on Flickr?

iPhone.



http://www.flickr.com/cameras/

Photo sharing across all social networks…

North of a trillion.

http://ben-evans.com/benedictevans/2014/6/24/imaging

What are the implications of "that"?

The average person looks at their phone 220 times...

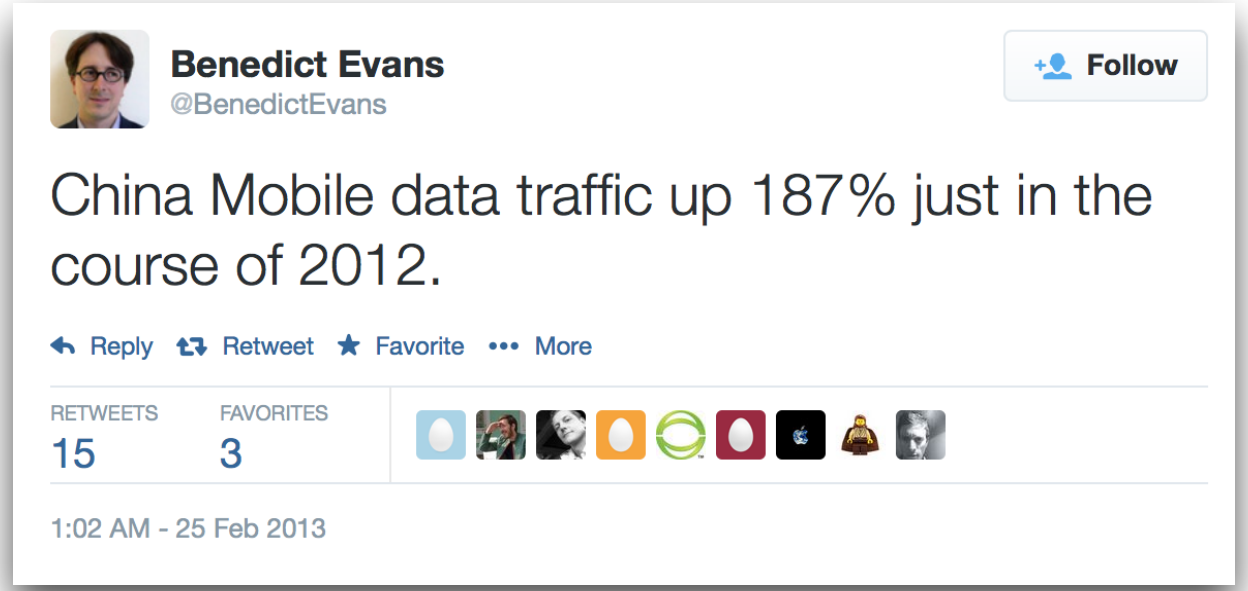
A DAY!

http://www.dailymail.co.uk/sciencetech/article-2783677/How-YOU-look-phone-The-average-user-picks-device-1-500-times-day.html

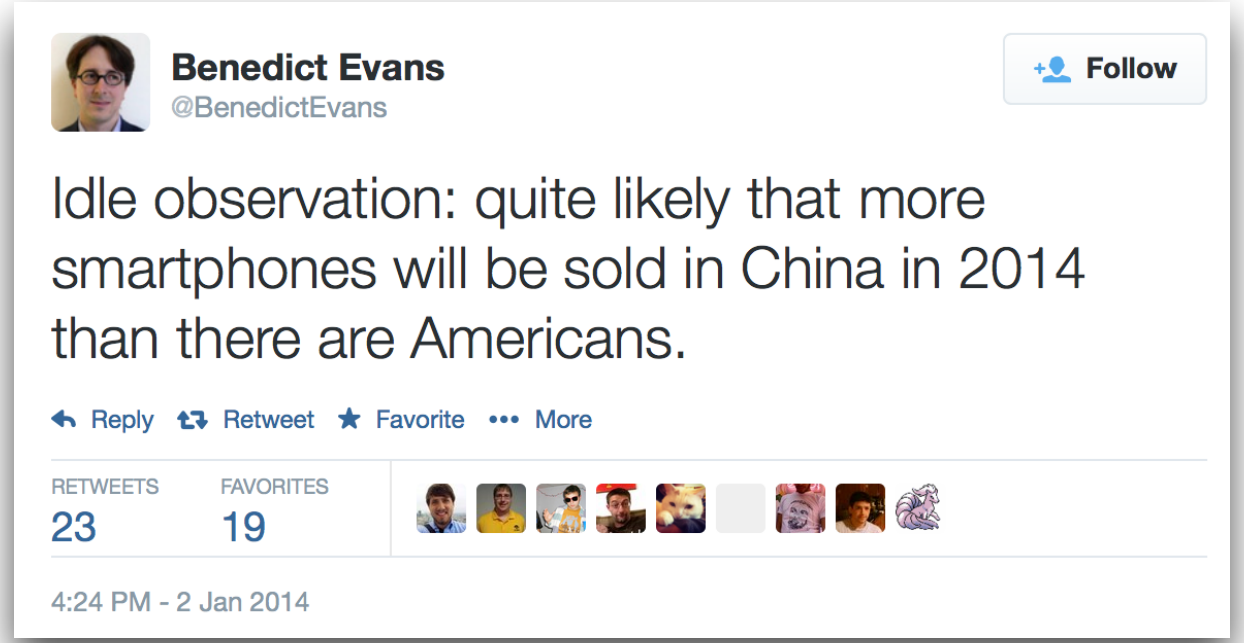
Average 3+ hours a day.

Or an entire day per week!

Global phenomena...



https://twitter.com/BenedictEvans/status/305966089458823168



https://twitter.com/BenedictEvans/status/418900526672388096

40% of UK online sales are from a mobile device.

https://www.uk.capgemini.com/news/uk-news/mobile-accounts-for-40-of-all-online-retail-sales

BBC iPlayer? More traffic on tablets than PCs.

http://www.bbc.co.uk/blogs/internet/posts/Christmas-2013-on-BBC-iPlayer

60% of Kenyans use mobile for money.

http://www.fsdkenya.org/finaccess/documents/13-10-31_FinAccess_2013_Report.pdf

Mobile is 20% of global Internet traffic.

http://marketingland.com/mobile-close-to-20-percent-of-internet-traffic-globally-58015/

We can’t ignore mobile.

We need a mobile solution.

OK, so let's build a mobile application then!

Hold on.

Now things get…interesting.

Bit of diversity out there.

A deluge of devices.

Which handsets?

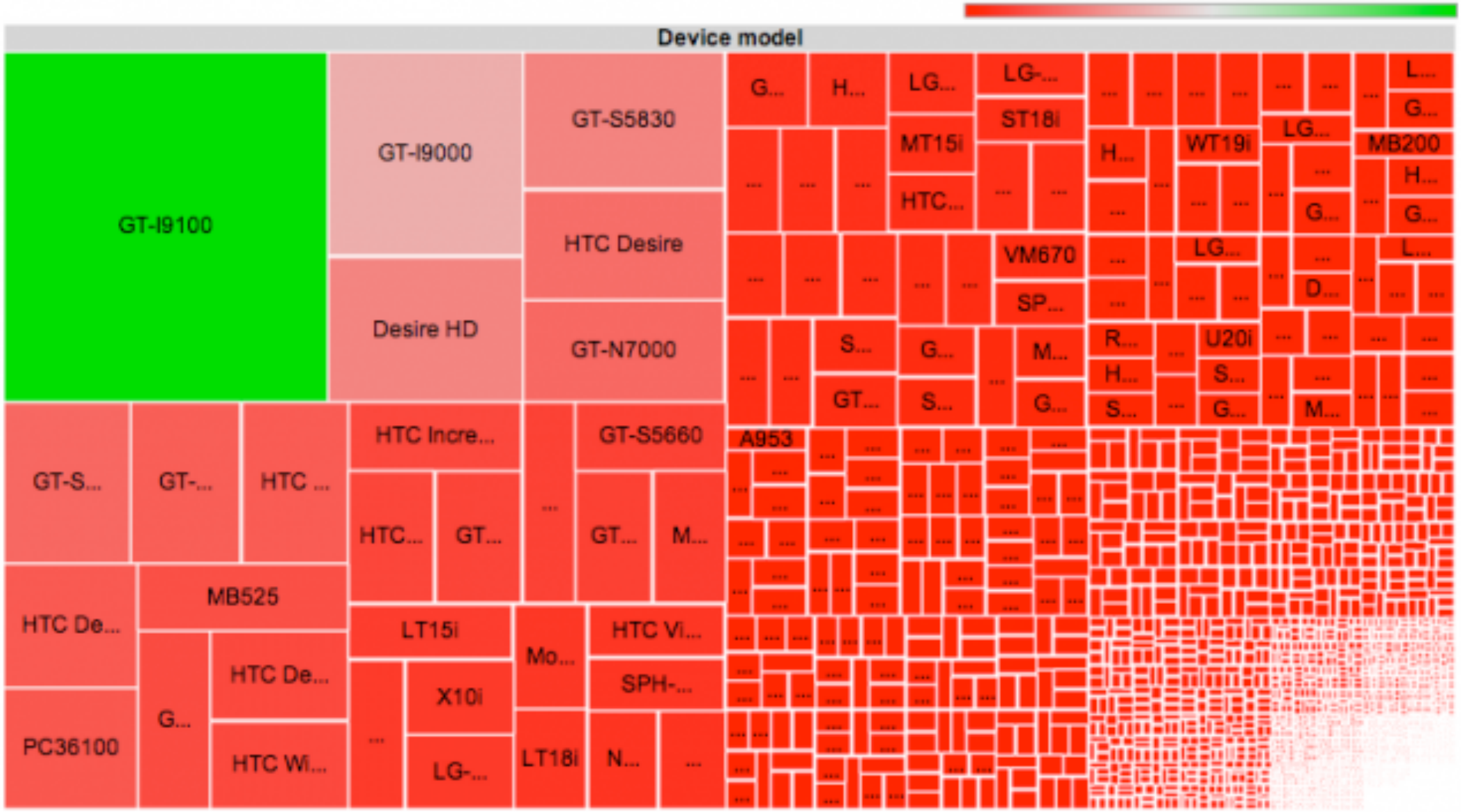
Except where otherwise noted, this work is licensed u
http://creativecommons.org/licenses/by-nd/3.

This Is What Developing For Android Looks Like.

http://techcrunch.com/2012/05/11/this-is-what-developing-for-android-looks-like/

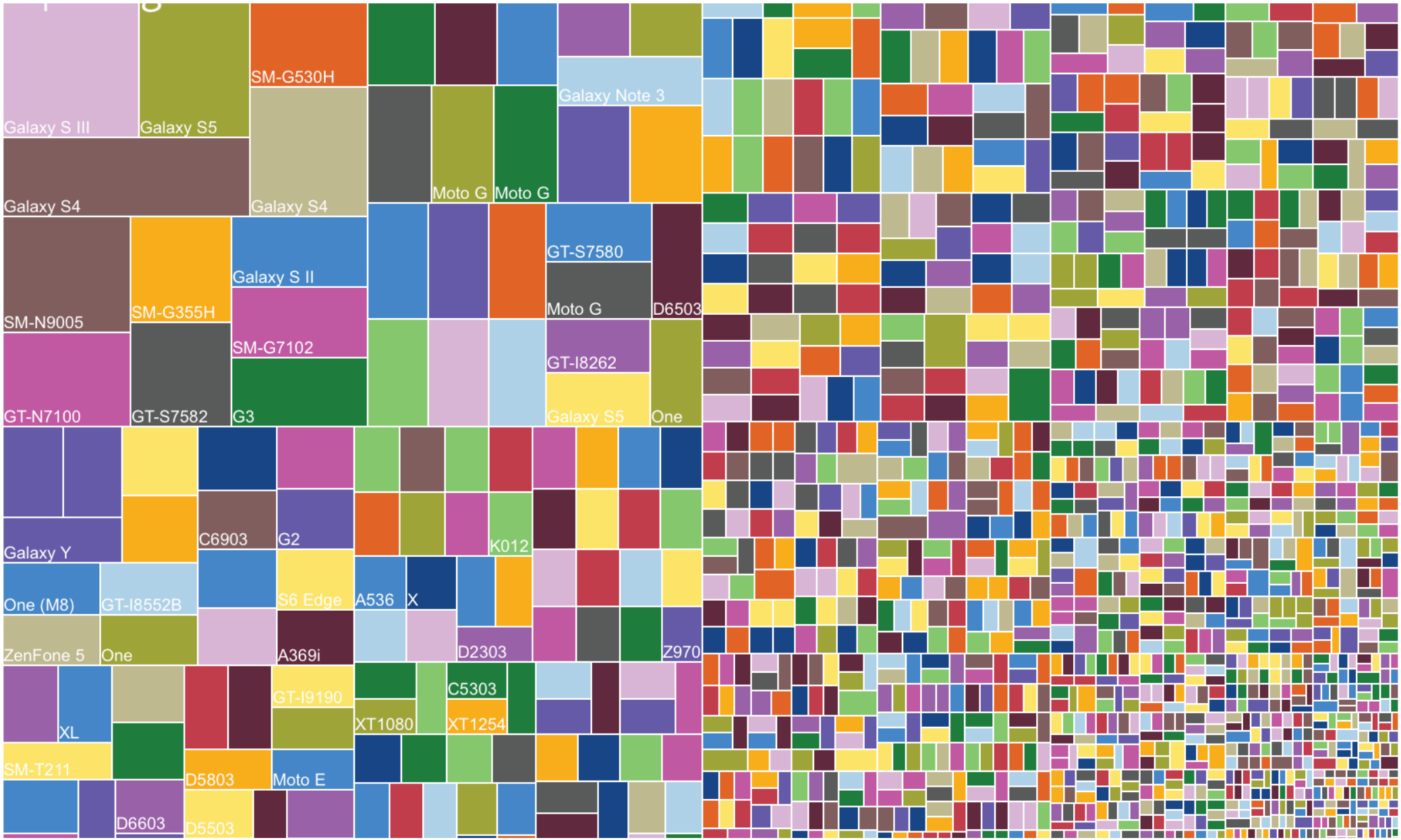
One developer,
3,997 devices.

http://arstechnica.com/gadgets/2012/05/android-fragmentation-one-developer-encounters-3997-devices/


Device model
GT-I9100
GT-I9000
Desire HD
GT-S5830
HTC Desire
GT-N7000
GT-S...
GT-...
HTC ...
HTC Incre...
HTC...
GT...
...
GT-S5660
GT...
M...
HTC De...
MB525
G...
HTC De...
HTC Wi...
PC36100
LT15i
...
X10i
LG-...
Mo...
LT18i
HTC Vi...
SPH-...
N...
...
G...
H...
LG...
LG-...
ST18i
MT15i
HTC...
VM670
SP...
S...
G...
GT...
S...
M...
G...
A953
WT19i
LG...
MB200
H...
G...
D...
L...
U20i
R...
H...
S...
G...
M...

OpenSignal's visualizations.

http://opensignal.com/reports/2015/08/android-fragmentation/



http://opensignal.com/reports/2015/08/android-fragmentation/

24,093 distinct
Android devices.

Some call it the
“matrix of pain”.

http://www.wired.com/business/2013/08/android-matrix-of-pain/all/1

Another way of visualizing the problem…

173°
024°
076°
188°
049°
358°
116°

What device size(s) do we need to worry about?

http://www.lukew.com/ff/entry.asp?1674

Everything from 3.5″ to 12.9″ with multiple resolutions.

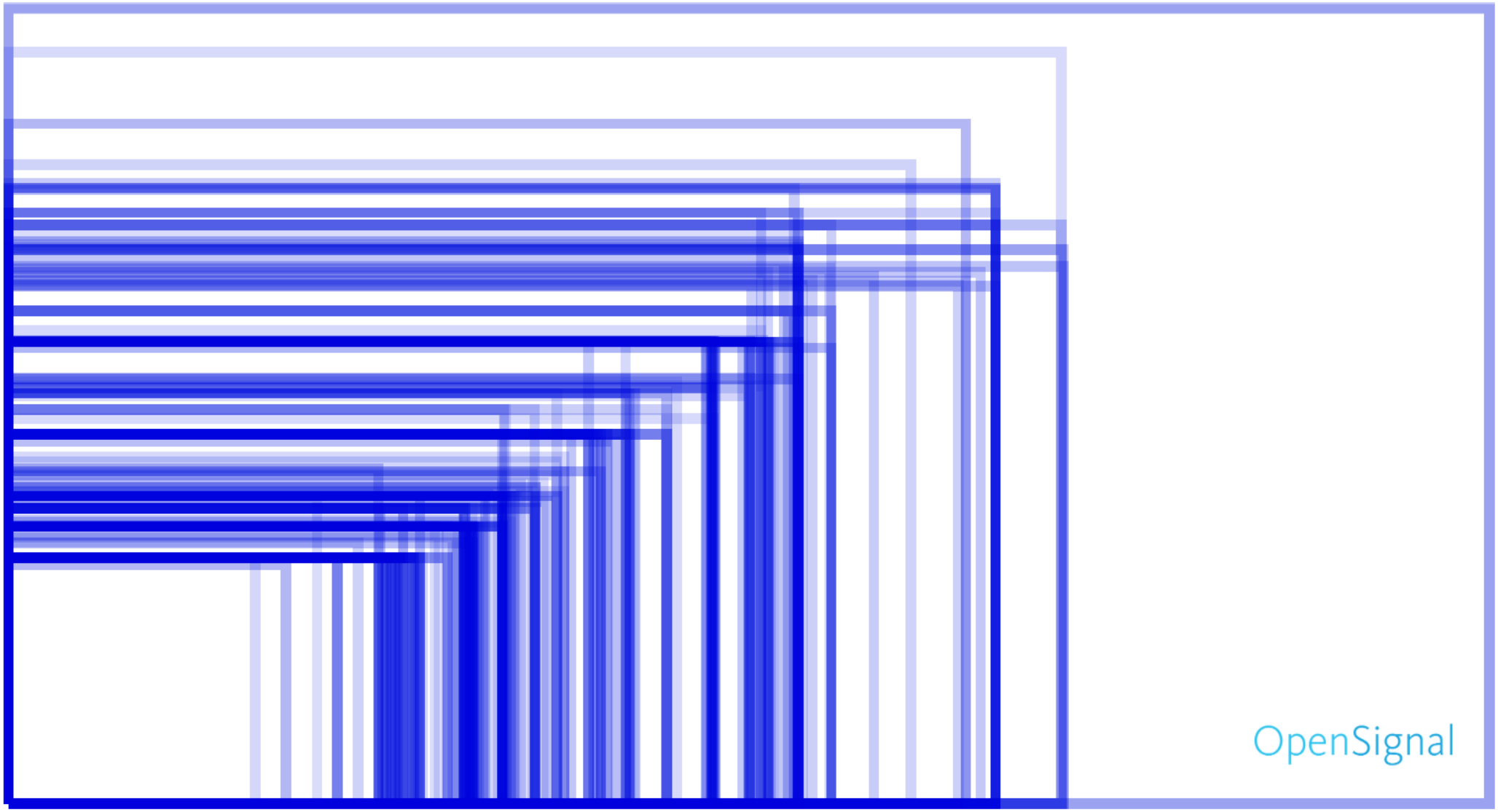
http://mobiledevicesize.com

LON 5:09
15:00
12PM LUNCH WITH KEN

And viewports aren't always the size you'd expect.

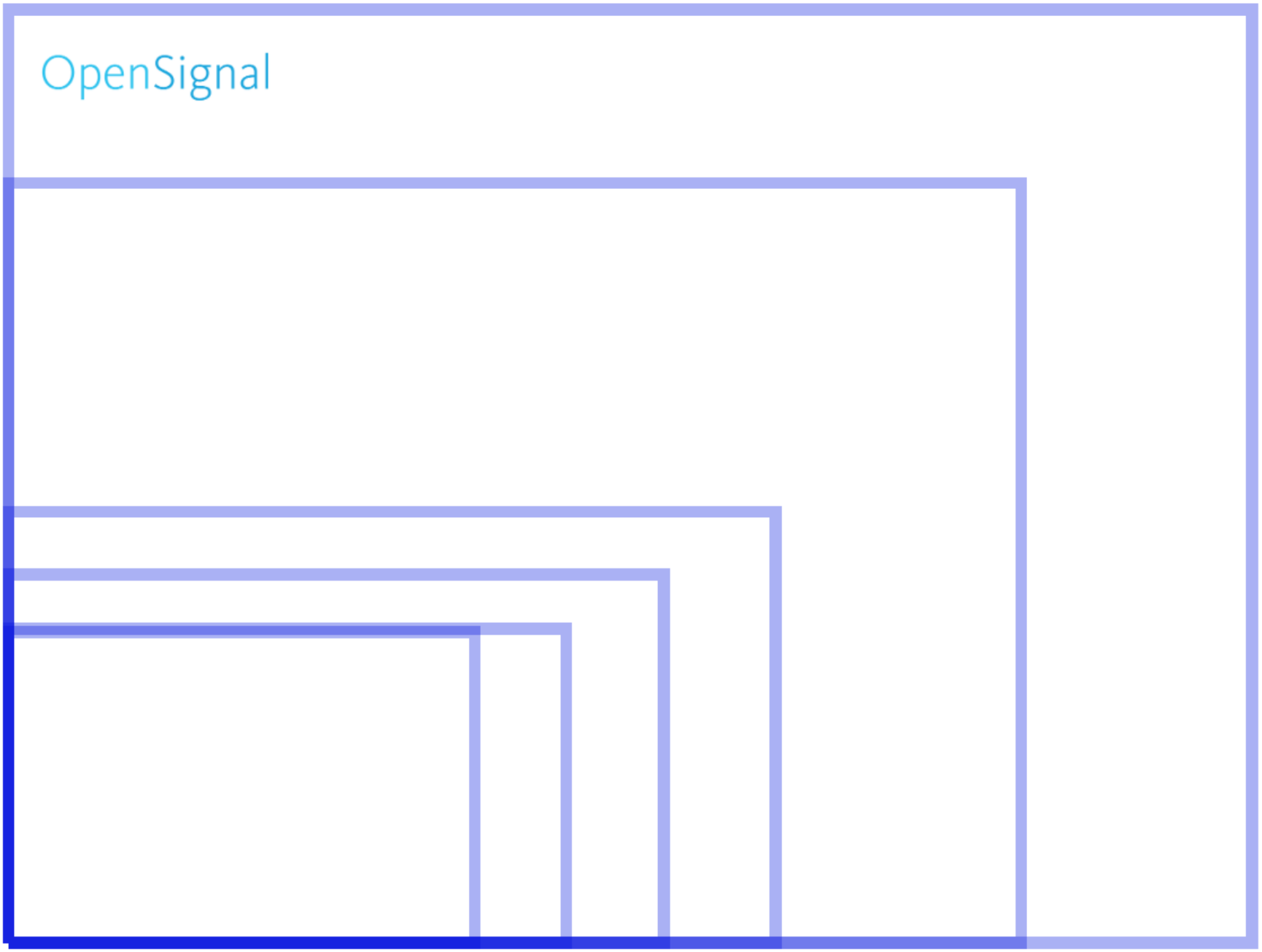
http://www.lukew.com/ff/entry.asp?1663

Another way to visualize the problem…



http://opensignal.com/reports/fragmentation-2013/

Here are the iOS screen sizes.



http://opensignal.com/reports/2015/08/android-fragmentation/

Android fragmentation, BlackBerry...Windows Phone?

We face a tremendous variety of devices...

The PC...isn't very personal.

Arguably it is legacy.

Horace Dediu
@asymco

Follow

Developing for desktop will be seen as legacy support in 5 years.

Reply Retweet Favorite ••• More

RETWEETS 140 FAVORITES 49

2:32 AM - 19 Nov 2012

https://twitter.com/asymco/status/270474579867230208

If it is external, it is mobile.

What they want,
whenever they want it...

On the device of their choice.

It is more and more likely that will be mobile.

Now what?

Is it too late to change careers?

Can’t just run away.

If it's external it's mobile.

Won't be long before your internal customers want it.

There are no silver bullets.

Responsive web design.

Client side rendering.

RESTful web services.

JSON…maybe edn.

https://github.com/edn-format/edn

Server === data.

Client === display.

Let the client figure out how to best display data.

Reuse is at the service layer.

New client is just
a new consumer.

UIs are more ephemeral.

Not like buying a couch…
won't last 10 years.

May only be a year or two before we have to replace it.

Or add another client.

We need to get comfortable with that.

In a year or two, there will be something new.

Either a new device or different dimensions.

Likely a new and improved UI framework...

Guarantee tomorrow will be different than today.

We need a flexible approach.

RWD.

Responsive web design.

Attempt to give a good experience across devices.

Minimizing pan and zoom.

Adapt the layout to the size of the device.

apple.com

Mac iPad iPhone Watch TV Music Support

WATCH

The Watch Reimagined Health & Fitness Design Gallery Apps Buy

To wear it is to love it.

Receive and respond to notifications in an instant. Track your daily activity. Control your music using only your voice. Pay for groceries just like that. With Apple Watch, important information and essential features are always just a raise of the wrist away.

Watch the keynote

Watch the TV ads

training.github.com

GitHub Training

Trainers Classes Schedule Kit Contact

Education from the experts

We love helping people just like you use GitHub and Git to collaborate and be more productive. No matter your level of experience, we've got something just for you.

Online Training

No matter where you are in the world, we have live web-based classes that you can access from the convenience of your own computer.

Learn from anywhere »

In-person Training

For face-to-face training, our highly experienced trainers can help increase your team's proficiency with a live, in-person class.

Schedule a session »

Training Resources

We've open sourced all of our class outlines, slides, and cheat sheets so you can easily leverage these high-quality educational materials.

Browse our resources »

Got 15 minutes and want to learn the basics of Git? Try Git right in your web browser.

Fluid grid.

“Remove” columns as the size decreases.

Resize (and remove) images.

CSS3 media queries.

Various breakpoints.

Many applications rely on Bootstrap.

http://getbootstrap.com

It isn’t magic fairy dust.

Sorry.

Tool to help with consistency.

Consistent experience doesn't mean identical.

Screen size alone won't tell you everything.

http://www.lukew.com/ff/entry.asp?1816

How far away is the user?

320 pixels at 10 feet (TV) versus 10 inches (phone).

Input mechanisms
also change.

Be careful with assumptions…

http://www.lukew.com/ff/entry.asp?1817

Great for content.

Less wonderful for input.

Must design the experience for the device.

MADE IN CHINA
100% COTTON
ONE SIZE FITS ALL
HAND WASH ONLY
angusf

How does that usually work out?

That said…

RWD should be part of your strategy.

But it isn’t the answer to all your problems.

The hard part…

What capabilities does a user need based on context?

Where are they? What do the need to do?

Must deliver the right thing at the right time.

People transition from one device to another.

Also use multiple devices together!

Start thinking about the bigger picture.

JavaScript MV*.

App development
is like fashion.

We go back and forth.

We've gone from dumb terminals to thick clients...

To web pages.

Are web apps any different than green screens?

Forms with holes.

Request/response paradigm.

The web has images
and can scroll!

But it is the same concept.

The pendulum
keeps swinging.

Now browsers are the PC.

They're getting more powerful.

V8, Nitro, JägerMonkey.

And we're asking them to do more.

HTML5.

Web Workers, Web Sockets, Offline.

REST.

JavaScript.

First class citizen.

We're writing more of it too!

New possibilities.

And we have options.

We can break out of the request/response approach.

Click and wait?

http://alexmaccaw.co.uk/posts/async_ui

Enter asynchronous UIs.

Why?

Performance matters.

Well, perceived performance at least.

Milliseconds matter.

Amazon: 100ms delay reduces sales by 1%.

400ms on Yahoo!?

5-9% drop in traffic.

500ms extra on Google?

Drops traffic by 20%.

http://www.slideshare.net/stubbornella/designing-fast-websites-presentation

It matters.

Embraces what we're doing.

Provide structure to all that JavaScript.

Partial refreshes,
JSON, services.

Takes it a step further.

MVC.

Or MVVM.

Or MVP.

MV+, MV-whatever.

Same basic approach we've used on the server...

But now on the client.

Full separation of concerns.

Non-blocking UI.

Decouple requests from the UI.

Render view on client.

Push state to client.

Talk to server asynchronously.

Update the UI, then tell the server about it.

Wait. What?

Things will go wrong!

Yep.

What about validation?

Server could reject the change.

Client side validation.

Need to validate on the server too...

What if the server pukes?

Error handling.

Parallel requests?

Pipeline ajax requests.

Try to navigate off gmail with update pending...

It isn’t perfect!

But there are answers to many issues.

Why should we do this again?

JavaScript isn't just for interns anymore.

We need a way to manage our JavaScript.

In the mobile space, page weights matter a lot.

These libraries are generally lightweight.

Drives a clean separation of concerns.

Server basically JSON pump.

Variety of clients consume the same services.

Certain amount of future proofing.

What client technology will dominate in 2 years?

Probably won't be like today.

The client figures out the best way to present data.

To a certain extent,
this is our fault.

Web apps were simpler
“back in the day”.

We kept pushing boundaries.

Customers expect more from a web app today.

Better user experience.

Uh no.

Some things should be synchronous!

Need *some* feedback.

REST.

Representational State Transfer.

https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

Architectural style.

Why?

Supported across OSes and programming languages.

Performance.

Scalability.

Loosely coupled.

Well defined response codes for handling errors.

Uniformity - it is all just a representation.

Its simple!

Set of well known verbs.

GET, POST, PUT, DELETE.

Client-server:
separation of concerns.

Stateless communication.

Session state lives on the client, not server.

Response must indicate if the data is cacheable.

Uniformity.

Hierarchical layers.

Well defined boundaries.

Optionally, server can also send code.

Set of resources.

“Information that can be named.”

Can be pretty much anything.

Resources need an identifier.

On the web, that's a URI.

Uniform Resource Identifier.

Often called an address.

A resource can have multiple URIs…

But a URI can only identify one resource.

An identifier is associated with a representation.

Encoded in a transfer format like JSON or XML.

Returns an abstraction,
not direct access.

Resources can have multiple representations.

URIs can indicate a representation.

Aka terminating the URL with .json or .xml.

Rails apps for instance.

URIs can (and should) be more opaque than that.

Let the consumer negotiate the representation.

Ultimately the server decides what to return.

Expect some comments from the peanut gallery.

But what about SOAP?

Very little about SOAP that qualifies as simple…

Heavyweight.

Overkill in many cases.

Many SOAP APIs have been deprecated.

Often used for app to app web services.

Where WS* is seen as an advantage…

RESTful APIs could also be used for app to app calls.

What about security?

Yes, you can secure
a REST API.

See Amazon!

HTTP Basic (with SSL please!) and OAuth.

How do we handle cross domain requests?

JSONP - JSON with padding.

Wrap a function call around the payload.

Include the callback function with the data.

A script element points to the resource.

Vulnerable to cross site request forgery.

CORS - Cross Origin Resource Sharing.

A “modern”
alternative to JSONP.

Browser includes an origin header with the request.

Resource returns appropriate header.

Supported by most modern browsers.

Can also proxy requests.

May use any of the techniques depending.

Brave new world.

Device diversity is the norm…

No slow down in sight.

Need to be prepared.

If it is external, it is mobile.

We need to adapt.

Be flexible.

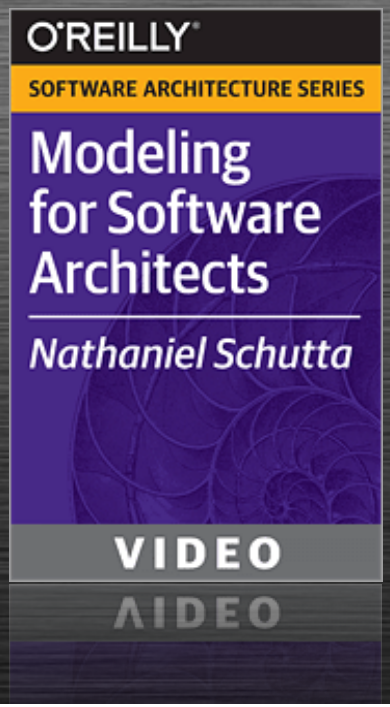
Embrace ephemerality.

Good luck!

Image Credits

- http://www.flickr.com/photos/rileyroxx/2969195685/
- http://www.flickr.com/photos/angusf/2270317189/sizes/o/

Thanks!



I'm a Software Architect, Now What?

Nathaniel T. Schutta
@ntschutta