

**MOBA1**

**MOBILE APPLICATIONS 1**

**HS21**

# OVERVIEW

- Introduction to MOBA1
- Mobile Devices
- Mobile Platforms
- Mobile Applications
- Project: POIFinder

# OVERVIEW

- Introduction to MOBA1
- Mobile Devices
- Mobile Platforms
- Mobile Applications
- Project: POIFinder

# TOPICS MOBA1

1. Introduction, Mobile Platform
2. Design & Development of Mobile Apps (1)
3. Design & Development of Mobile Apps (2)
4. Kotlin
5. Android (1)
6. Android (2)
7. Android (3)
8. Android (4)
9. Beacons

# TOPICS MOBA1

10. Mobile Web: Overview, Device APIs
11. Mobile Web: Layout Options, Flexbox, Grid
12. Mobile Web: Components, APIs, Frameworks
13. Presentations: Projects and Labs
14. Presentations: Projects and Labs

# OUTLOOK: MOBA2 (PLANNED)

1. Mobile Web: React.js (1)
2. Mobile Web: React.js (2)
3. React Native: Introduction
4. React Native: Navigation
5. React Native: More Components and APIs
6. iOS Introduction
7. Swift
8. SwiftUI (1)
9. SwiftUI (2)
10. SwiftUI (3)
11. Presentations: Projects and Labs
12. Presentations: Projects and Labs

# LESSONS AND SELF-STUDY

- Lecture: two lessons per week
- Practical exercises: also two lessons
- Self-study

# PREVIOUS KNOWLEDGE

- HTML, CSS, JavaScript (WBE)
- Programming in C and Java

# ASSESSMENT

Weight	Type
--------	------

20%	Exercises and presentations during class time
-----	---

80%	Final exam
-----	------------

# EXERCISES AND PRESENTATIONS

- Small mobile project
- Exercises on most topics
- Presentation: project results

# EXERCISES AND PRESENTATIONS

- Small mobile project / presentation (teams of 2)  
up to 5 points (project: 3, presentation: 2)
- Solved exercises / presentation  
up to 5 points

Points acquired during class time count for the semester finals  
(weighted, 20%)

If the practical results are inferior to the semester finals, only the latter count for the course grade

# FINAL EXAM

Depends on Corona situation...

Alternative 1	Alternative 2
Written exam (*)	Mobile project
1.5h	5h
Open book	Open book

(\*) Different types of questions / tasks:  
Multiple choice, code examples, ...

# DOCUMENTS

In electronic form in Moodle:

- Slides
- Exercises
- Additional material

# MOODLE

<https://moodle.zhaw.ch/course/view.php?id=2155>

- Weekly schedule
- Documents
- Course details and conditions
- Forum (?)

# PRACTICAL LESSONS AND LECTURES

## PRACTICAL LESSONS

- Tasks explained
- You work with your notebooks
- Teamwork is appreciated

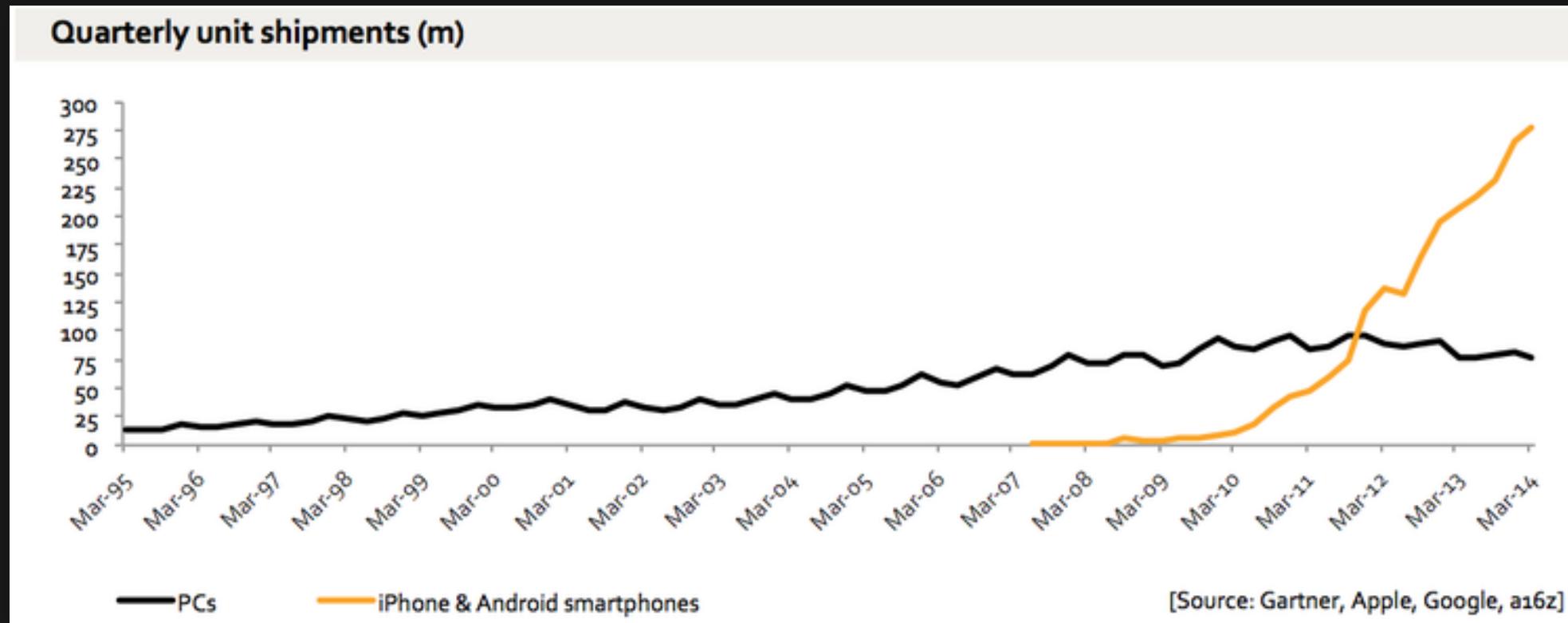
## LECTURES

- Private conversations disturbing
- Interaction and participation appreciated
- Take notes

# OVERVIEW

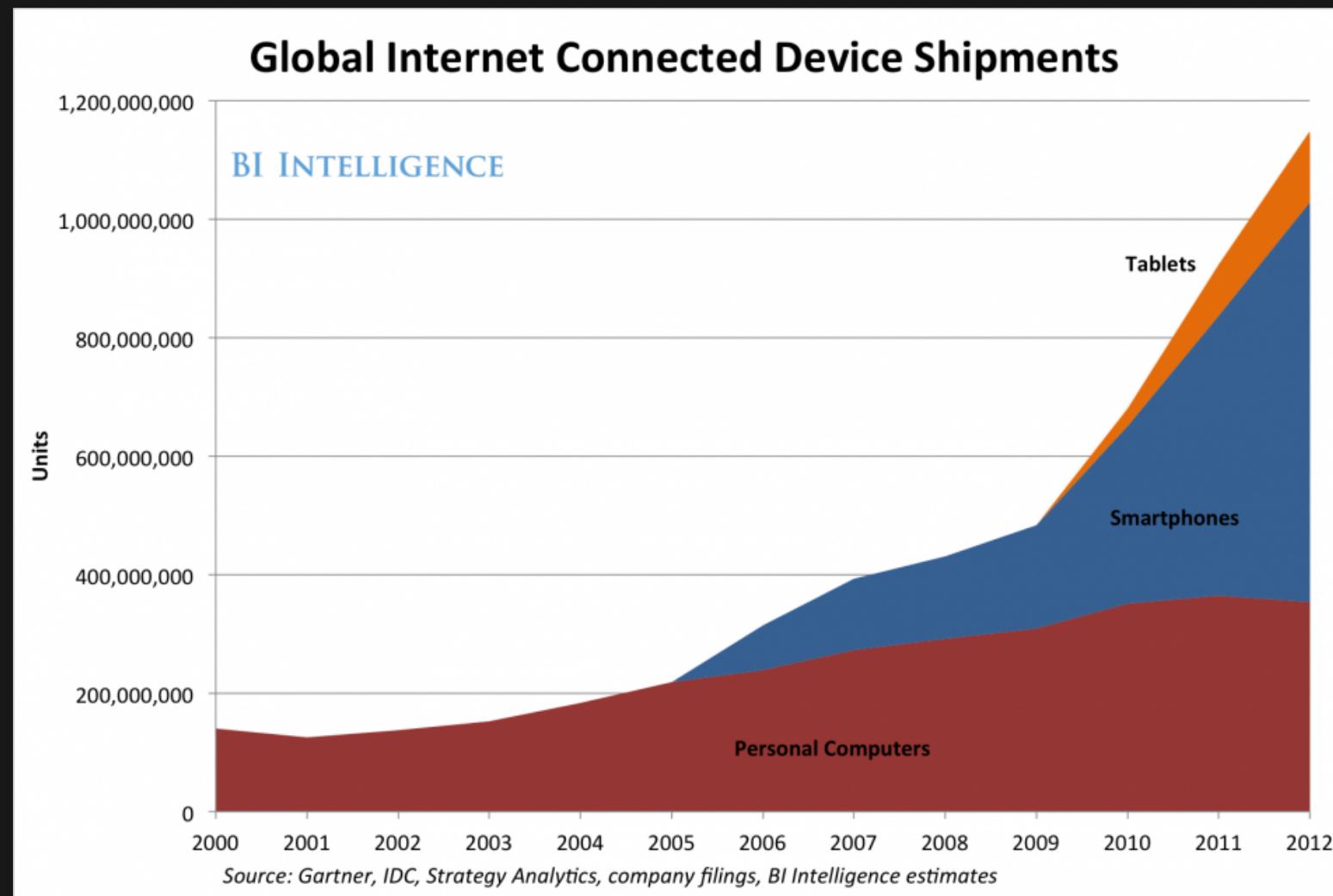
- Introduction to MOBA1
- Mobile Devices
- Mobile Platforms
- Mobile Applications
- Project: POIFinder

# SMARTPHONE AND TABLET MARKET



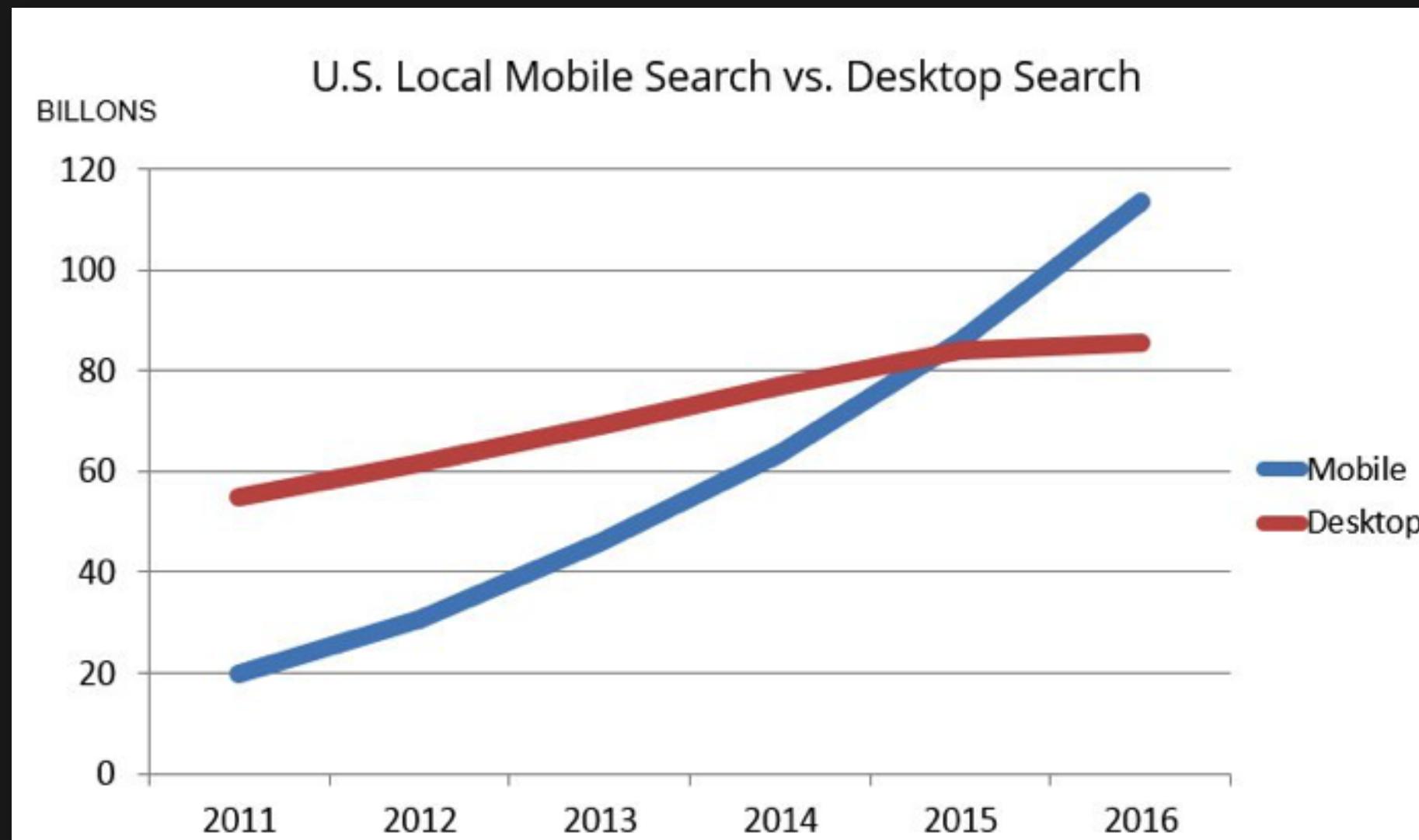
↓ more ↓

# SMARTPHONE AND TABLET MARKET



↓ more ↓

# US MOBILE SEARCH VS DESKTOP SEARCH



# FEATURE PHONE TO SMART PHONE



↓ more ↓



Quelle:  
<https://www.samsung.com>

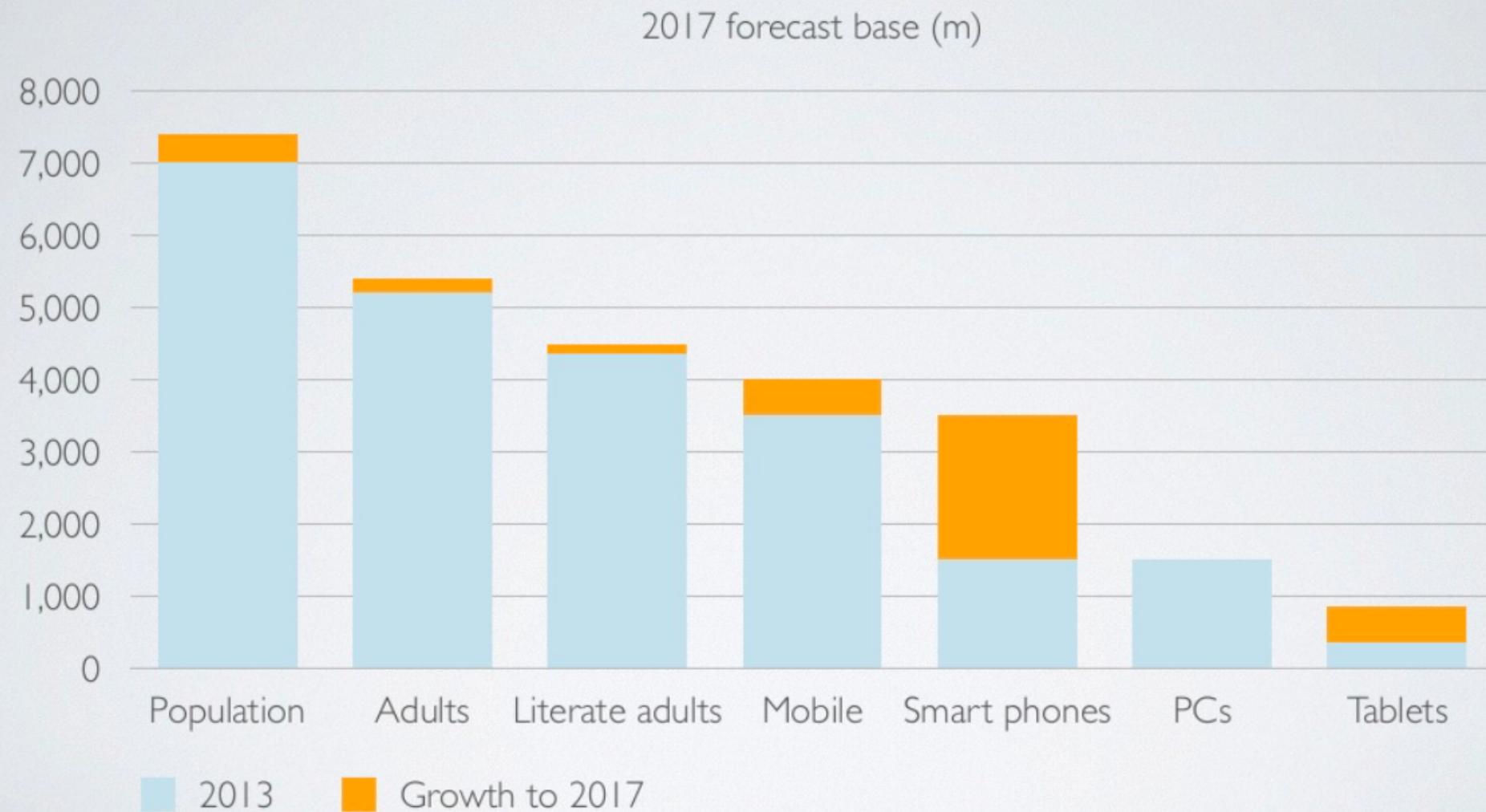
# MOBILE DEVICES



# MOBILE DEVICES ??

- Notebooks
- Netbooks
- Feature Phones
- Personal Digital Assistants
- Smartphones
- Tablets
- Smartwatches
- Calculator watches
- Handheld game consoles
- Head-mounted displays
- Wearable computers
- Portable media players
- Digital still cameras (DSC)
- Digital video cameras (DVC) or digital camcorders
- Personal navigation devices (PND)

# The world in 2017



# MOBILE DEVICES IN MOBA 1/2

- In MOBA1/2 we focus on Smartphones (and Tablets)
- With some comments on Smartwatches and Wearables
- Priority on devices with widespread mobile OSs

# FEATURES OF MOBILE DEVICES

- Smaller screens (here, smartphones and tablets differ)
- Different input concepts (touch, keyboard, stylus)
- Slow, unstable network connection
- Less powerful processors
- Batteries – minimize power consumption

# DEVICE SENSORS

- Camera
- Microphone
- Geolocation, GPS
- Accelerometer
- Gyroscope
- Magnetometer
- Battery state
- Proximity sensor

# INSIDE IPHONE XS AND XS MAX



Teardown  
↓ more ↓

# IPHONE 6S TAPTIC ENGINE



↓ more ↓

# IPHONE 6S TAPTIC ENGINE

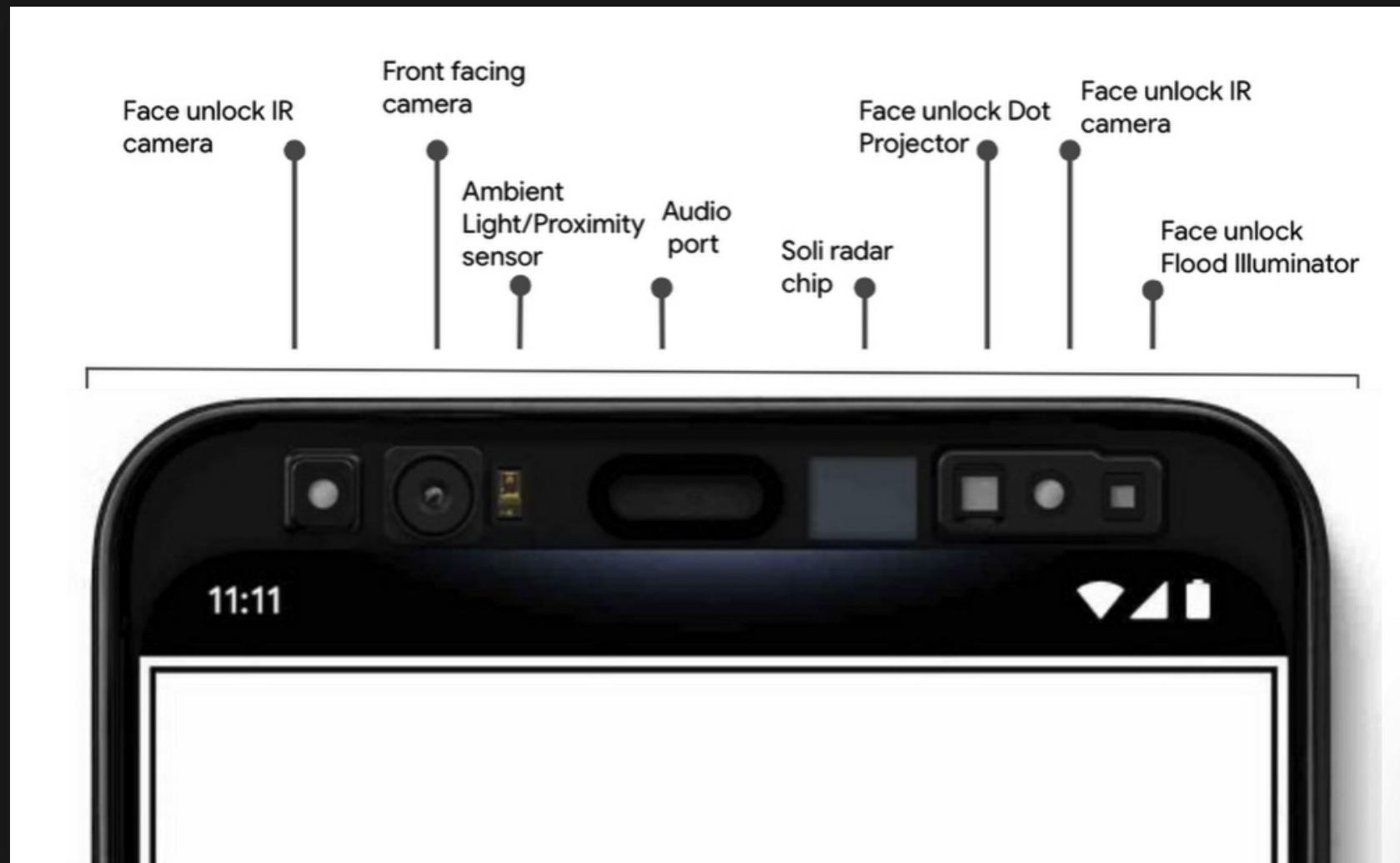


# INSIDE SAMSUNG GALAXY S10 AND S10E

Teardown



# GOOGLE PIXEL 4



↓ more ↓

# GOOGLE PIXEL 4 – PROJECT SOLI



# OVERVIEW

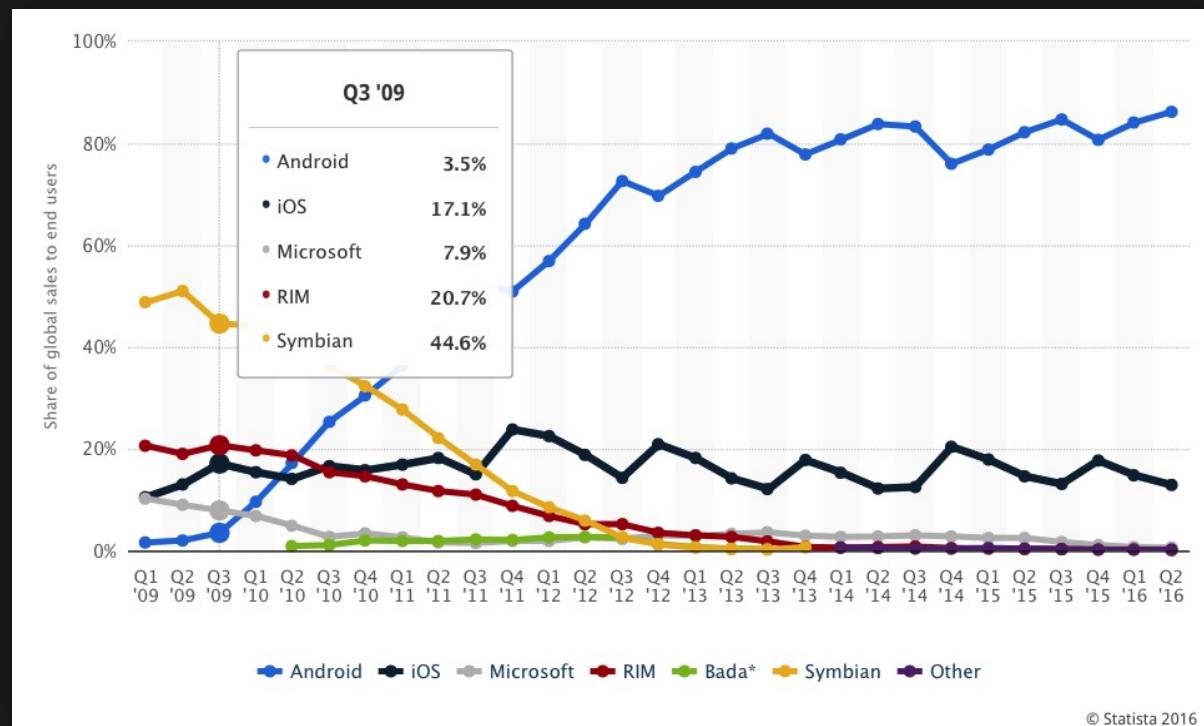
- Introduction to MOBA1
- Mobile Devices
- **Mobile Platforms**
- Mobile Applications
- Project: POIFinder

# Mobile platform wars over?

- Apple and Google's Android have won: unlikely that other platforms will be relevant
- Apple camped out at the high end, Android taking the rest
- But Google's control of Android is partial
- Facebook and Amazon trying to extract value
- Samsung's position uncertain

# ANDROID AND IOS

- Dominate the smartphone market
- Easily 90% in key markets
- Also at the top in terms of developer mindshare
- But: Mobile space changes continuously



# Different focus for innovation

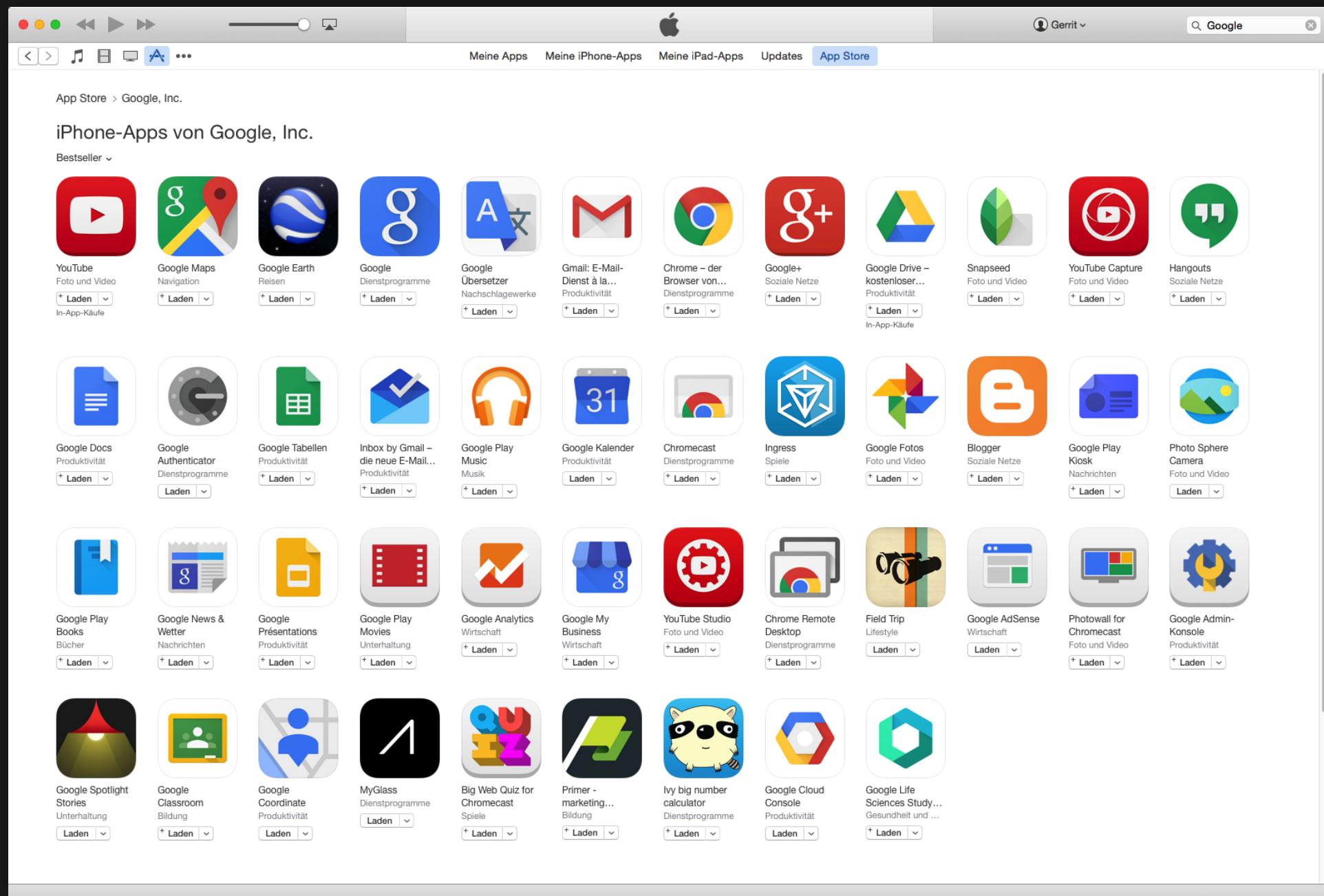
## Apple

- Integrated hardware & software
- Fingerprints, Bluetooth, Airdrop, 64 bit etc
- Move innovation down the stack (hard for Google)
- Directed discovery (iBeacon)

## Google

- Systemic fragmentation, little hardware control
- Google Now, Maps, Plus, semantic search etc
- Move innovation up the stack (hard for Apple)
- Predictive discovery (Now)

# GOOGLE APPS FOR IOS

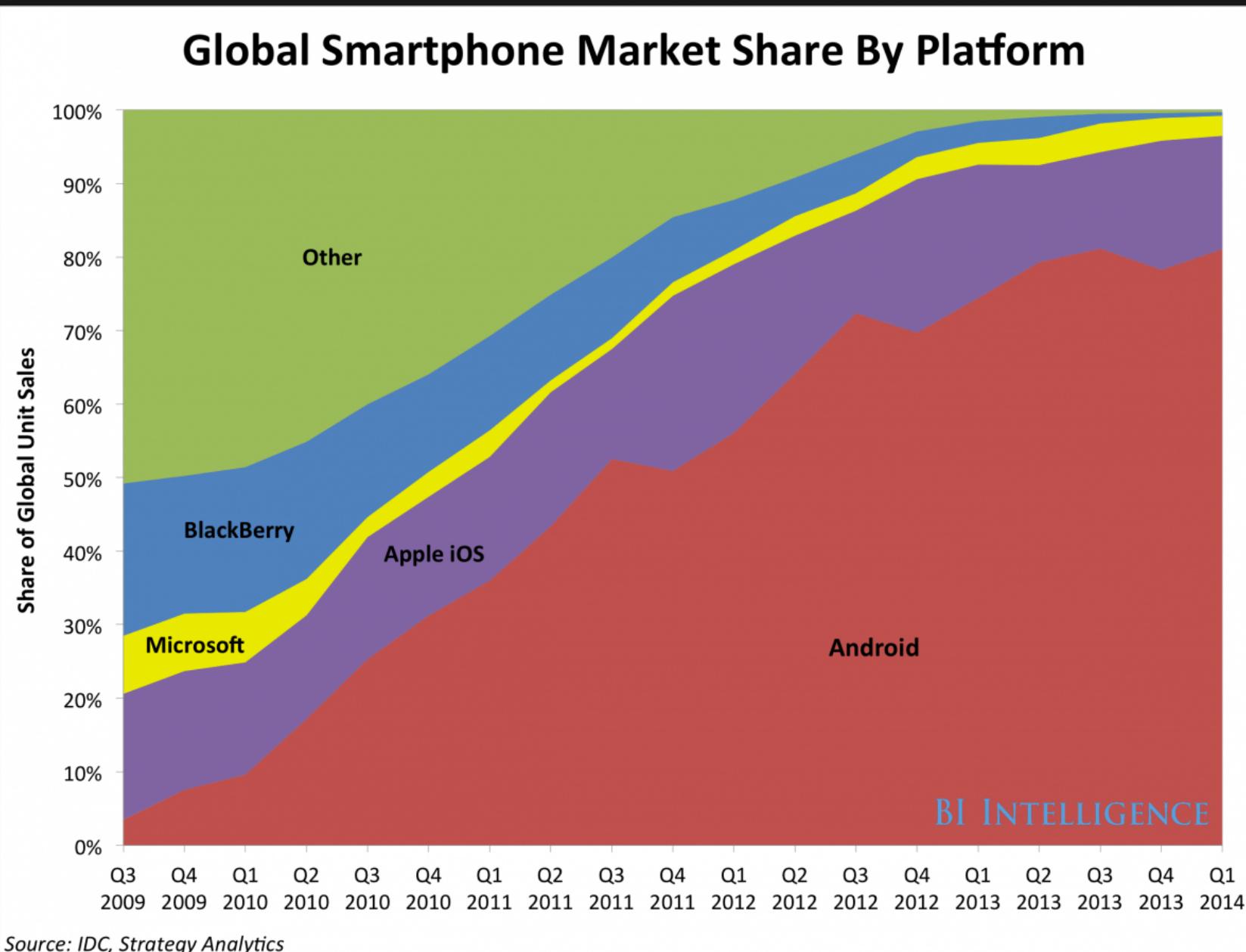


# ANDROID'S PURPOSE

- Spreading the use of Google services across the mobile web
- Makes economic sense to offer Android for free
- In 2009 device vendors needed an answer to the iPhone
- Android was available (and free...)
- Most device vendors jumped on the Android bandwagon
- Except for Apple, Nokia, and BlackBerry
- Nokia and BlackBerry failed

↓ more ↓

## Global Smartphone Market Share By Platform



# ANDROID DIFFERENTIATION

- Google allowed Android differentiation
- Purpose: Get device vendors to adopt it
- Vendors could create their own interface layer
- They could also change default apps, including the browser
- Samsung TouchWiz, HTC Sense, MotoBlur, ...
- Differentiation is the positive slant on fragmentation

# IOS: HISTORY (1)

- iPhone unveiled by Steve Jobs January 9th 2007
- OS named iPhone OS
- November 2010: launch of the fourth generation of the OS
  - renamed to iOS
  - launch of the original iPad
- New version each year

And Then Steve Said, ‘Let There Be an iPhone’

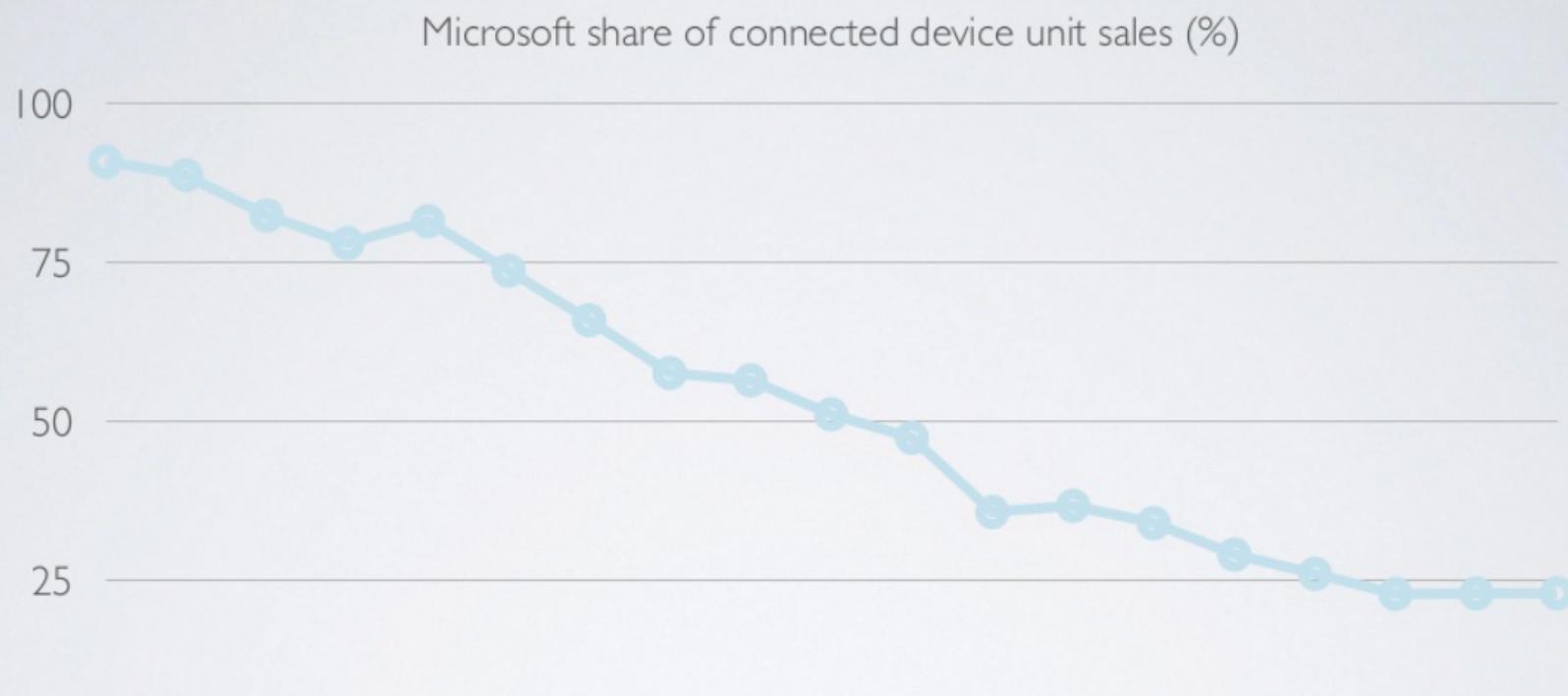
# IOS: HISTORY (2)

- iOS 7 was touted as a major UI refresh
- iOS 8 with new frameworks and services, Apple Watch
- iOS 9 with split view on iPad, better Siri, faster graphics
- iOS 10 with haptic feedback, better Siri, better search
- iOS 11 with Files App, Drag&Drop, ARKit, Metal 2
- iOS 12/13 with ?? ([various improvements](#))
- iOS 14 with Widgets on Home Screen ([more](#))

→ [Release notes for developers](#)

[High adoption rate of each iOS version soon after release](#)

# The irrelevance of Microsoft



# MARKET SHARE

Period	Android	iOS	Windows	BlackBerry	Others
2018	85.1%	14.9%	0.0%	0.0%	0.0%
2017	85.0%	14.7%	0.1%	0.0%	0.1%
2016	83.4%	15.4%	0.8%	0.0%	0.4%
2015	78.0%	18.3%	2.7%	0.3%	0.7%
2014	81.2%	15.2%	2.5%	0.5%	0.7%
2013	75.5%	16.9%	3.2%	2.9%	1.5%
2012	59.2%	22.9%	2.0%	6.3%	9.5%

(Source: IDC)

# THE UNDERDOGS



# FIREFOX OS (DISCONTINUED)

- Mozilla Foundation
- Linux based, open source
- Aimed at lower end smartphones
- Browser engine Gecko as a runtime environment for apps
- First release published in February 2013
- September 2016: end of development announced

# BLACKBERRY OS (DISCONTINUED)

- Ten years ago: number 2 in smartphone business
- Now a niche system
- Although most drawbacks were fixed
- Devices with and without real keyboards

Blackberry

# UBUNTU PHONE (DISCONTINUED)

- Also known as *Ubuntu Touch*
- Mobile version of the Ubuntu operating system
- Designed primarily for touchscreen mobile devices
- Qt5-based touch user interface
- April 2017: project terminated

Ubuntu Phone

# TIZEN

- Open source, Linux based
- Intel, Samsung, brand of the Linux Foundation
- Contributing: Fujitsu, NTT Docomo, Huawei, Vodafone, Orange
- Designed for various smart devices:
  - Smartphones, tablets, netbooks, watches, tvs, car multimedia
- Source code mostly open source
- Based on work of Nokia (Maemo), Intel (Moblin, MeeGo), LiMo Foundation (LiMo) and Samsung (Bada OS)

[www.tizen.org](http://www.tizen.org)



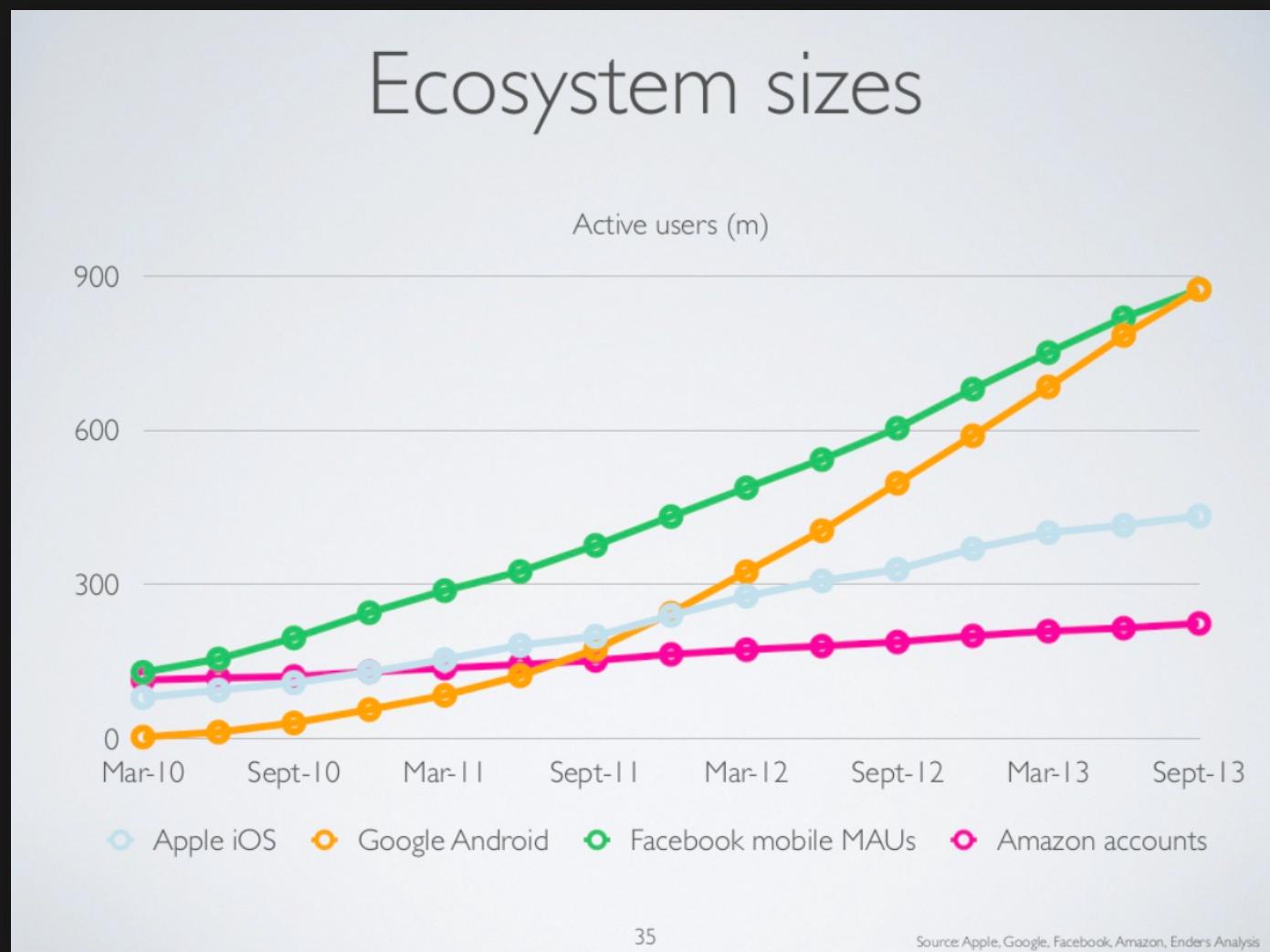
- Jolla (founded by former Nokia employees)
- Sailfish OS based on MeGo and Mer (community project)
- Homescreen with tiles
- Controlled with gestures

[jolla.com/jolla/](http://jolla.com/jolla/)  
[blog.jolla.com/](http://blog.jolla.com/)

# OLDER PLATFORMS

- Java ME (J2ME)
- Symbian (Nokia)
- Palm OS
- WebOS (Palm → HP → LG)
- Bada OS (Samsung)

# ECOSYSTEM SIZES

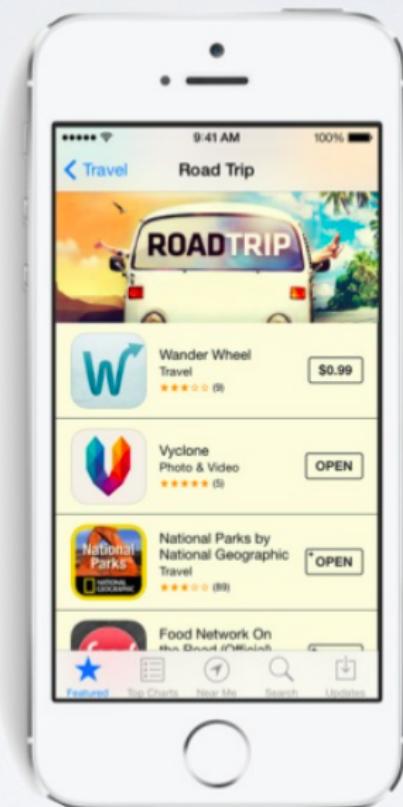


- netmarketshare
- kantarworldpanel
- StatCounter

# ECOSYSTEM DIFFERENCES

Ecosystem is the key leverage point

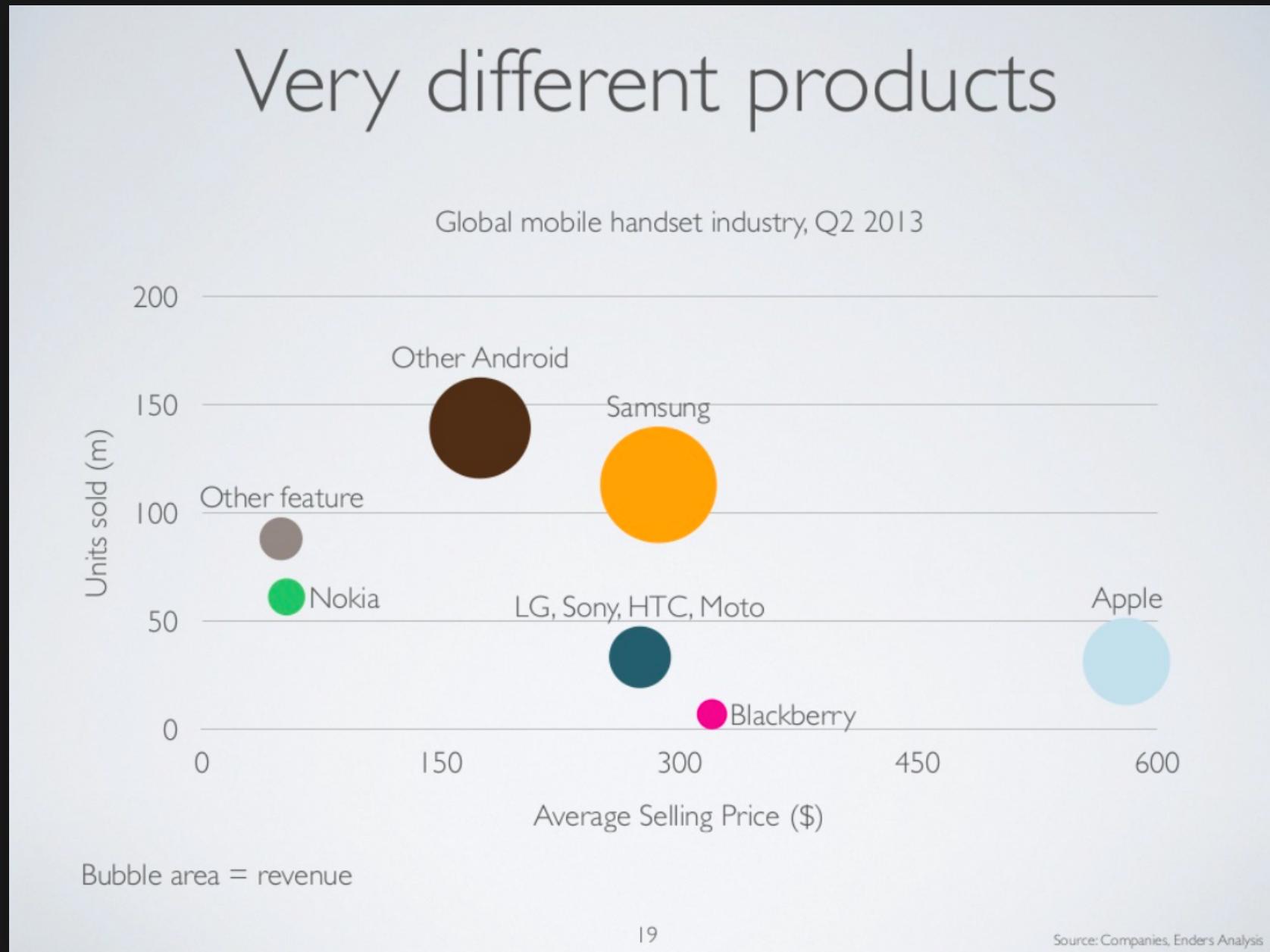
For Apple, the ecosystem is what sells hardware



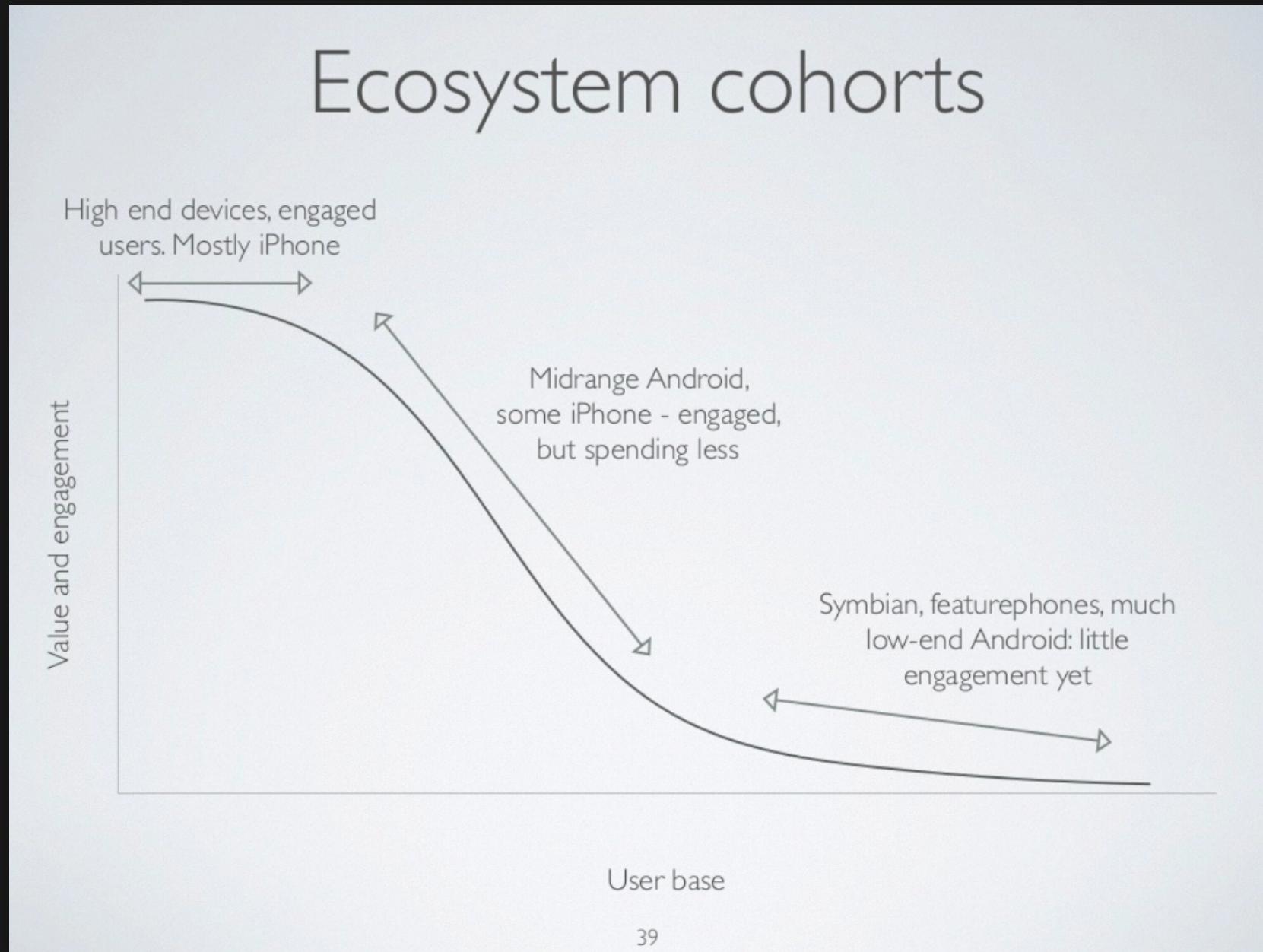
39

For Amazon, Google and Facebook, the experience on the phone is what drives engagement with all their services

# PRODUCT CATEGORIES



# ECOSYSTEM COHORTS



# OVERVIEW

- Introduction to MOBA1
- Mobile Devices
- Mobile Platforms
- **Mobile Applications**
- Project: POIFinder

# MOBILE APPLICATIONS

- Traditionally we targeted phones
- Then tablets followed
- Today we have a complete range of device types
  - smartwatch, phone, tablet, TV and automotive
  - Each one poses its own usability challenges

# MOBILE APPLICATIONS

- Native
- Web-based
- Hybrid

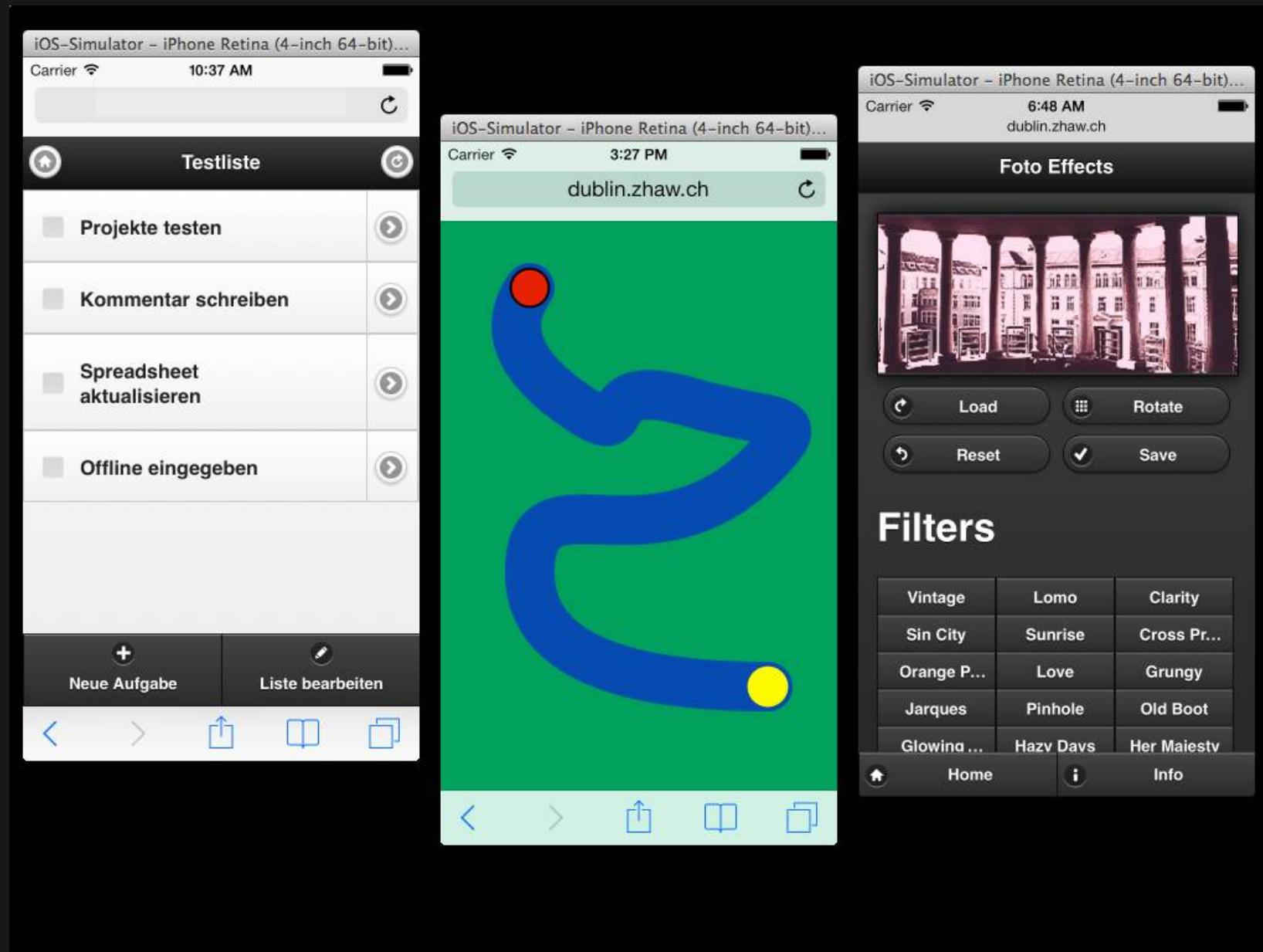
# NATIVE MOBILE APPLICATIONS

- Platform specific language
- Platform specific APIs
- Platform specific central app store
- Advantages
  - Usually offer the best performance
  - Deepest integration
  - Best overall user experience
- Disadvantage
  - Most complex development option

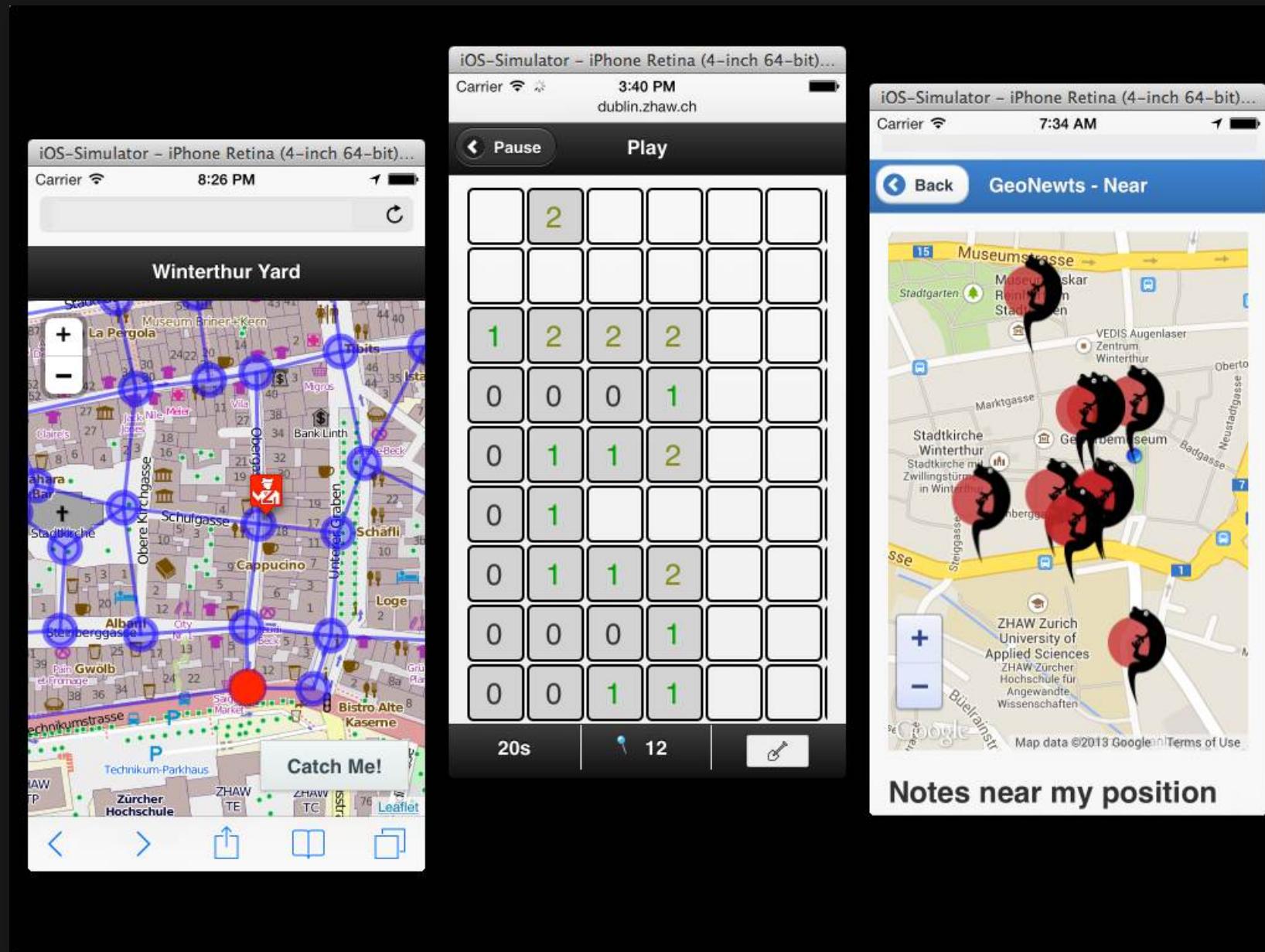
# WEB-BASED MOBILE APPLICATIONS

- Based on HTML, JavaScript and CSS
- Do not rely on an app store
- Essentially locally stored mobile sites
- Try to emulate the look-and-feel of an app
- Can be added to the home screen

# SAMPLE WBE PROJECTS



# SAMPLE WBE PROJECTS



# HYBRID MOBILE APPLICATIONS

- Frameworks can build a native wrapper around web apps (example: PhoneGap)
- Use native code for enhanced performance and integration
- Use a webview with HTML-based content for other parts
- Allow to revise content and features without using the app stores

# SOURCES

- Slides and other material from courses WEB1, WBE
- Mobile Developer's Guide To The Galaxy, 18th Edition, Open XChange,  
<https://www.open-xchange.com/resources/mobile-developers-guide-to-the-galaxy/>
- Mobile is eating the world, Benedict Evans, 2013,  
<http://de.slideshare.net/bge20/2013-11-mobile-eating-the-world>
- Mobile, context and discovery, Benedict Evans, keynote at the InContext 2014 conference,  
<http://de.slideshare.net/bge20/2014-02-incontext>,  
Youtube: <https://www.youtube.com/watch?v=VnhbvS0MBxE>

# OVERVIEW

- Introduction to MOBA1
- Mobile Devices
- Mobile Platforms
- Mobile Applications
- Project: POIFinder

# MOTIVATION

- POIFinder is one of our sample apps
- The POIFinder should be able to show interesting Points of Interest (POIs) nearby
- The base for POIFinder is OpenStreetMap (OSM)

# WHAT IS OSM?

- OpenStreetMap is a collaborative project (similar to wikipedia) to create and maintain a world map
- It was launched in 2004. Now, thousands of users are permanently updating the map
- Similar to wikipedia, OSM data can be used for free, all data is available with the Open Database License [Summary](#)
- Take a look at <https://www.openstreetmap.org> to get an idea
- OSM data contains much more (public transit, indoor mapping, hiking trails, ...) that the default map does not show

# MAP BUILDING BLOCKS

OSM data consists of elements. These are:

- Nodes (a point in space containing a latitude and longitude value)
- Ways (a linear list of nodes to define things like streets or area boundaries)
- Relations (group other elements to define a relation of these like a bus route or a multi polygon)

All elements can contain further subtags, interesting for us are "tag" and "nd".

# POIS IN OSM: NODE

```
<node id="29455086" visible="true" version="4" changeset="17291336"
  timestamp="2013-08-10T13:47:04Z" user="mdk" uid="178186"
  lat="47.4987786" lon="8.7261242">
  <tag k="addr:city" v="Winterthur"/>
  <tag k="addr:country" v="CH"/>
  <tag k="addr:housenumber" v="13"/>
  <tag k="addr:postcode" v="8400"/>
  <tag k="addr:street" v="Neumarkt"/>
  <tag k="amenity" v="cinema"/>
  <tag k="name" v="Kiwi"/>
  <tag k="wheelchair" v="limited"/>
</node>
```

# POIS IN OSM: WAY

```
<way id="138604454" visible="true" version="3" changeset="30341810"
  timestamp="2015-04-19T21:10:05Z" user="mdk" uid="178186">
  <nd ref="1519868939"/> <!-- references nodes via id -->
  ...
  <!-- other nd refs cutted out -->
  <tag k="addr:city" v="Winterthur"/>
  <tag k="addr:country" v="CH"/>
  <tag k="addr:housenumber" v="18"/>
  <tag k="addr:postcode" v="8400"/>
  <tag k="addr:street" v="Stadthausstrasse"/>
  <tag k="amenity" v="bank"/>
  <tag k="atm" v="yes"/>
  <tag k="building" v="yes"/>
  <tag k="name" v="UBS"/>
  <tag k="opening_hours" v="Mo-Fr 09:00-16:30; Th 09:00-17:00"/>
  <tag k="operator" v="UBS"/>
  <tag k="wheelchair" v="limited"/>
</way>
```

# OSM: API

- OSM offers a REST API return XML documents; his [API](#) can be used to send changes to the database; editing data requires an authentication
- Within this API, a simple read call to receive data by a bounding box is available:

```
GET /api/0.6/map?bbox=left,bottom,right,top
```

- The OSM call simply returns all elements within the bounding box ([example](#)); to prevent returning too much data, there might be cases where certain elements are returned (see API description)

# POI TAGGING

*Amenity: something, such as a swimming pool or shopping centre, that is intended to make life more pleasant or comfortable for the people in a town, hotel, or other place: The council has some spare cash, which it proposes to spend on public amenities.*

*CAMBRIDGE DICTIONARY*

We will concentrate on the tag "amenity" when showing POIs. Find a list of all possible OSM amenities [here](#).

# CONCLUSION

- Basically, the POIFinder app will always try to find the current location, retrieve all elements nearby and filter the answer to all nodes and ways that are amenities.
- To make life more easy for us, we will ignore the relations, although it might be possible that there is also an amenity tagged. We will also simply use the first node as a POI reference when dealing with ways.
- We will define the look and feel of the POIFinder during this course. Note that the look and feel as well as the features will vary between our different apps.

