

Android

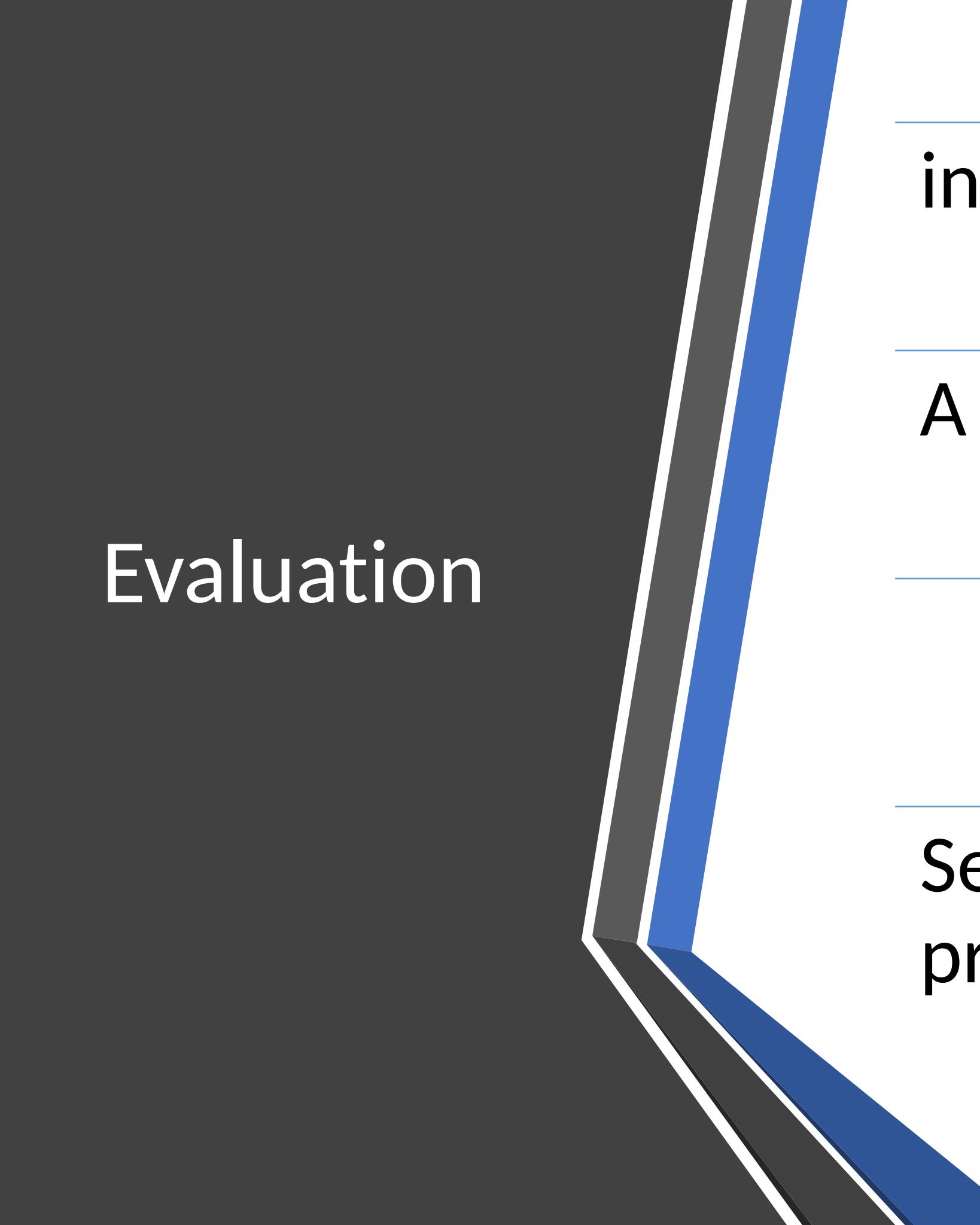
Mark Joselli

mark.joselli@gmail.com

Objective of the Course

installing Android
development
environment

creating interface and
debug it



Evaluation

in-class exercises

A final project

Send me the zip/report with the
project



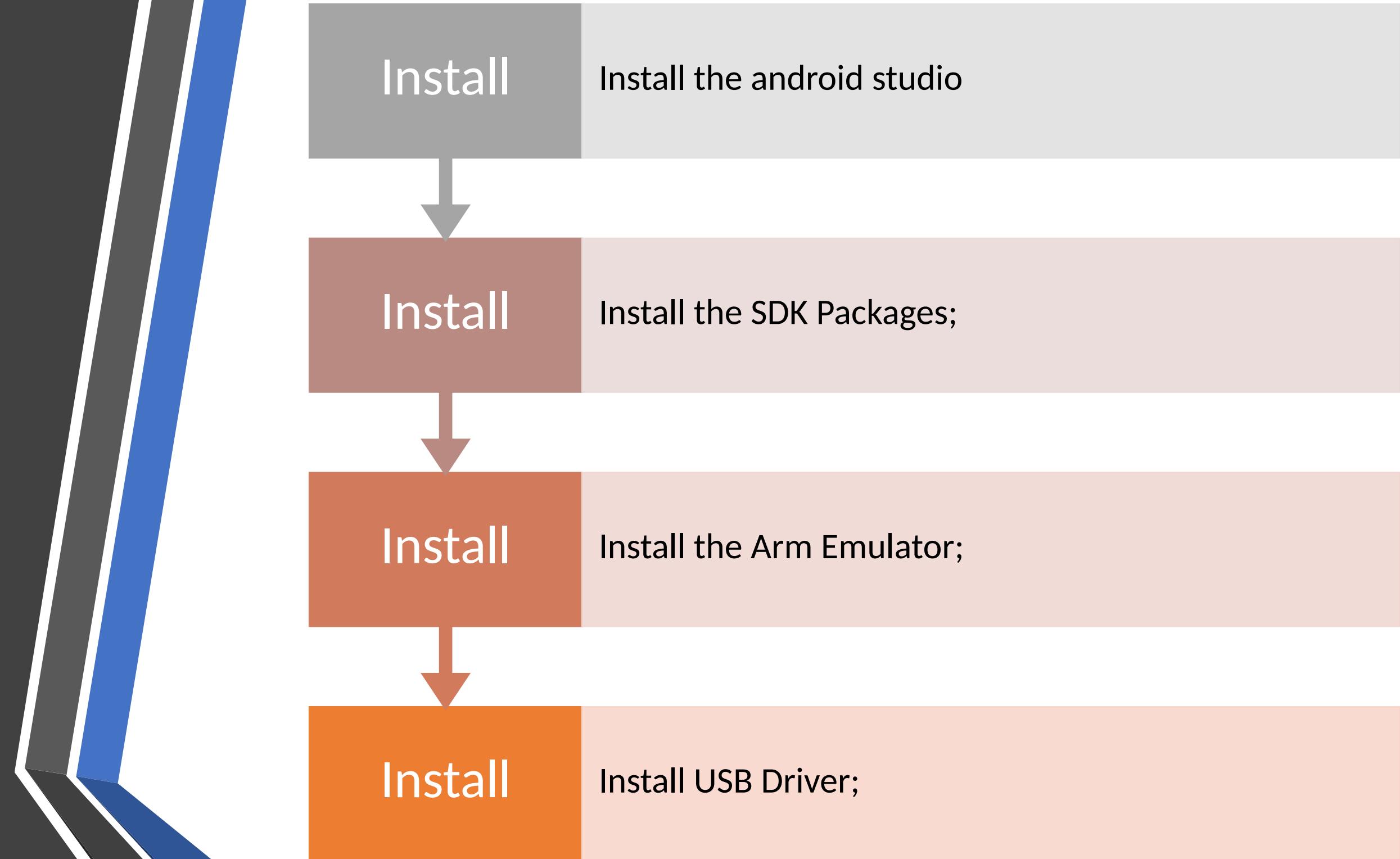
Todays
Objective

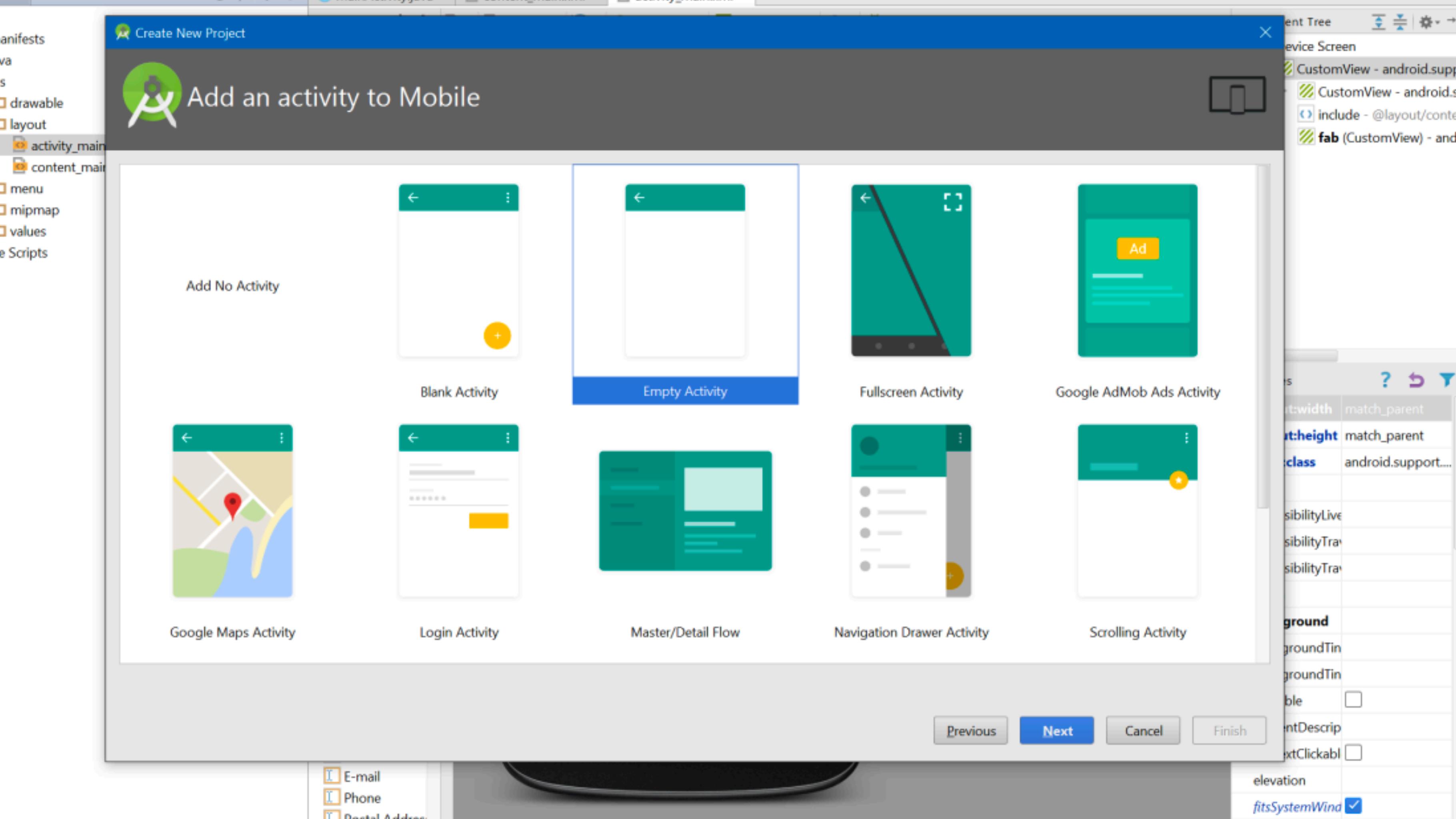
Prototyping;

First contact with
android;

Review of technologies;

Install Android





Android Development

- Need the knowledge of two technologies:
 - XML: for layout, definitions and configurations;
 - Java/Kotlin: for the behavior.

XML

- Used to define some of the resources
 - Layouts (UI)
 - Strings
- Manifest file
- Preferred way of creating UIs
 - Separates the description of the layout from any actual code that controls it
 - Can easily take a UI from one platform to another



XML

- XML is a meta markup language for text documents / textual data;
- XML allows to define languages ("applications") to represent text documents / textual data



XML

```
<article>
<author>Gerhard Weikum</author>
  <title>The Web in 10 Years</title>
</article>
```

- Easy to understand for human users
- Very expressive (semantics along with the data)
- Well structured, easy to read and write from programs



Advantages of XML

- Truly Portable Data
- Easily readable by human users
- Very expressive (semantics near data)
- Very flexible and customizable (no finite tag set)
- Easy to use from programs (libs available)
- Easy to convert into other representations (XML transformation languages)
- Many additional standards and tools
- Widely used and supported

A Simple XML Document

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```

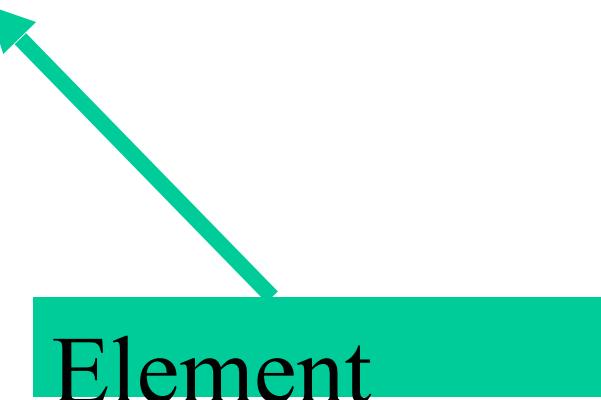
A Simple XML Document

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the
      universal...
    </section>
  </text>
</article>
```

tags

A Simple XML Document

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```



A Simple XML Document

Start Tag

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```

Start Tag

End Tag

Element

A Simple XML Document

Start Tag

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```

End Tag

Element

Content of the
Element
(Subelements
and/or Text)

A Simple XML Document

Start Tag

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```

Attributes with
name and value

End Tag

Element

Content of the
Element
(Subelements
and/or Text)

Elements in XML Documents

- (Freely definable) **tags**: **article**, **title**, **author**
 - with start tag: `<article>` etc.
 - and end tag: `</article>` etc.
- **Elements**: `<article> ... </article>`
- Elements have a **name (article)** and a **content (...)**
- Elements may be nested.
- Elements may be empty: `<this_is_empty/>`
`
`
- Each XML document has exactly one root element and forms a tree.

Elements vs. Attributes

- Elements may have **attributes** (in the start tag) that have a **name** and a **value**, e.g. `<section number="1">`.
- What is the difference between elements and attributes?
- Only one attribute with a given name per element (but an arbitrary number of subelements)
- Attributes have no structure, simply strings (while elements can have subelements)
- Example: `<person born="1912-06-23" died="1954-06-07">Alan Turing</person>`

A Simple XML Document

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```



<https://www.youtube.com/watch?v=c8VF3nb8Z4M>

- Mobile Market
- Android x iOS

Mobile world



Moto Q

BlackBerry

Palm Treo

Nokia E62



Works like magic

No stylus

Far more accurate

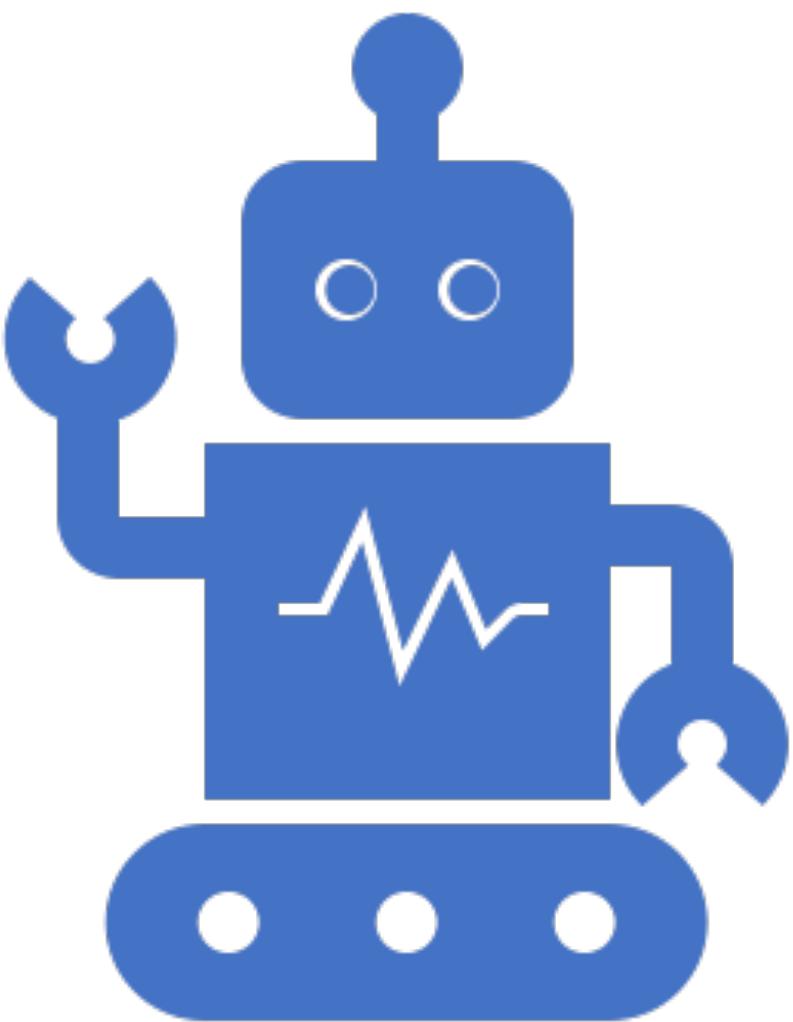
Ignores unintended touches

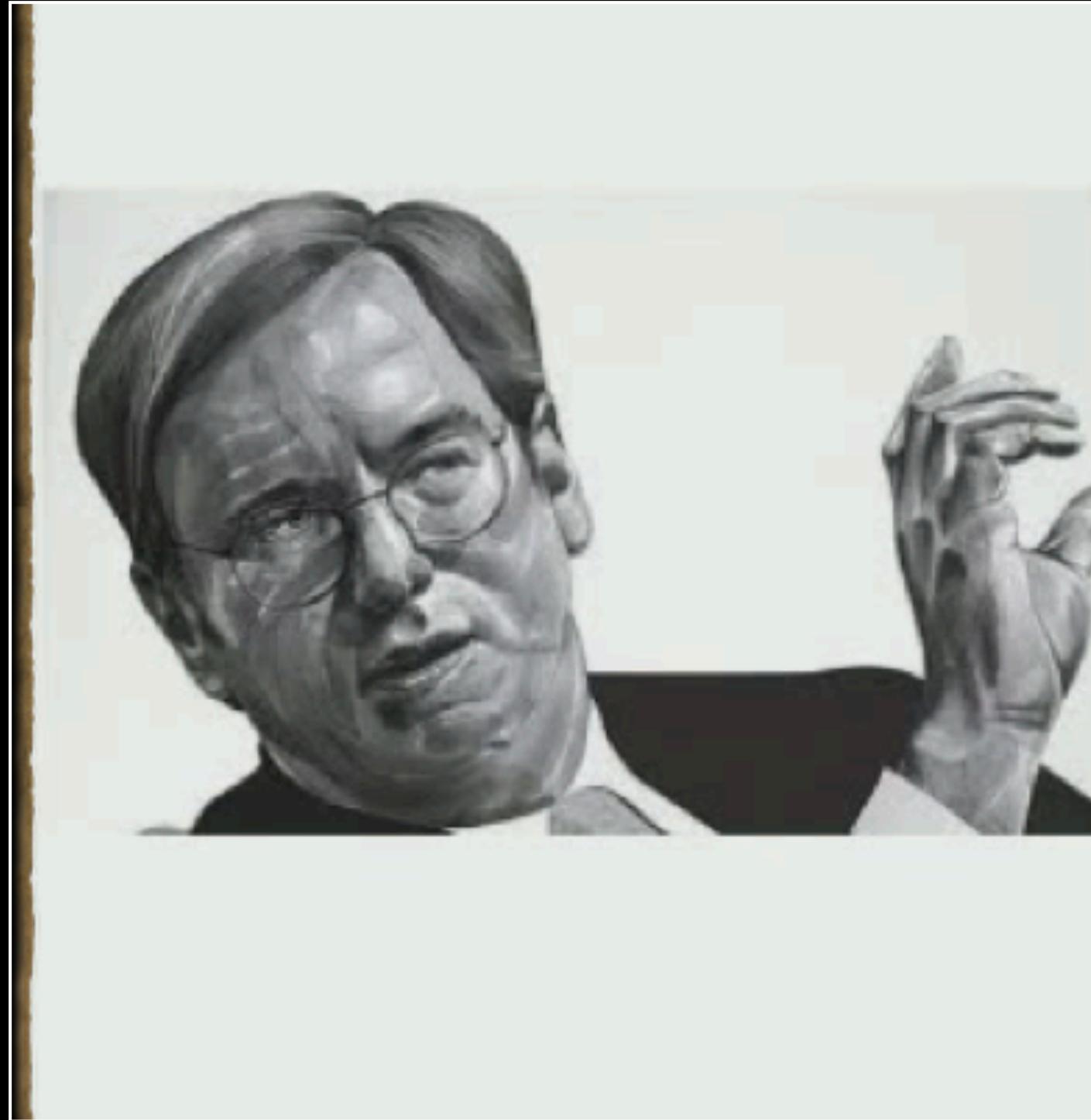
Multi-finger gestures

Patented !

Introduction to Android

- “Android is a software stack for mobile devices that includes an operating system, middleware and key applications.”
- “The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.”
- Stated goal: "accelerate innovation in mobile and offer consumers a richer, less expensive, and better mobile experience."





Our goal is not just a single device. Our vision is a mobile platform that runs on many many different devices.

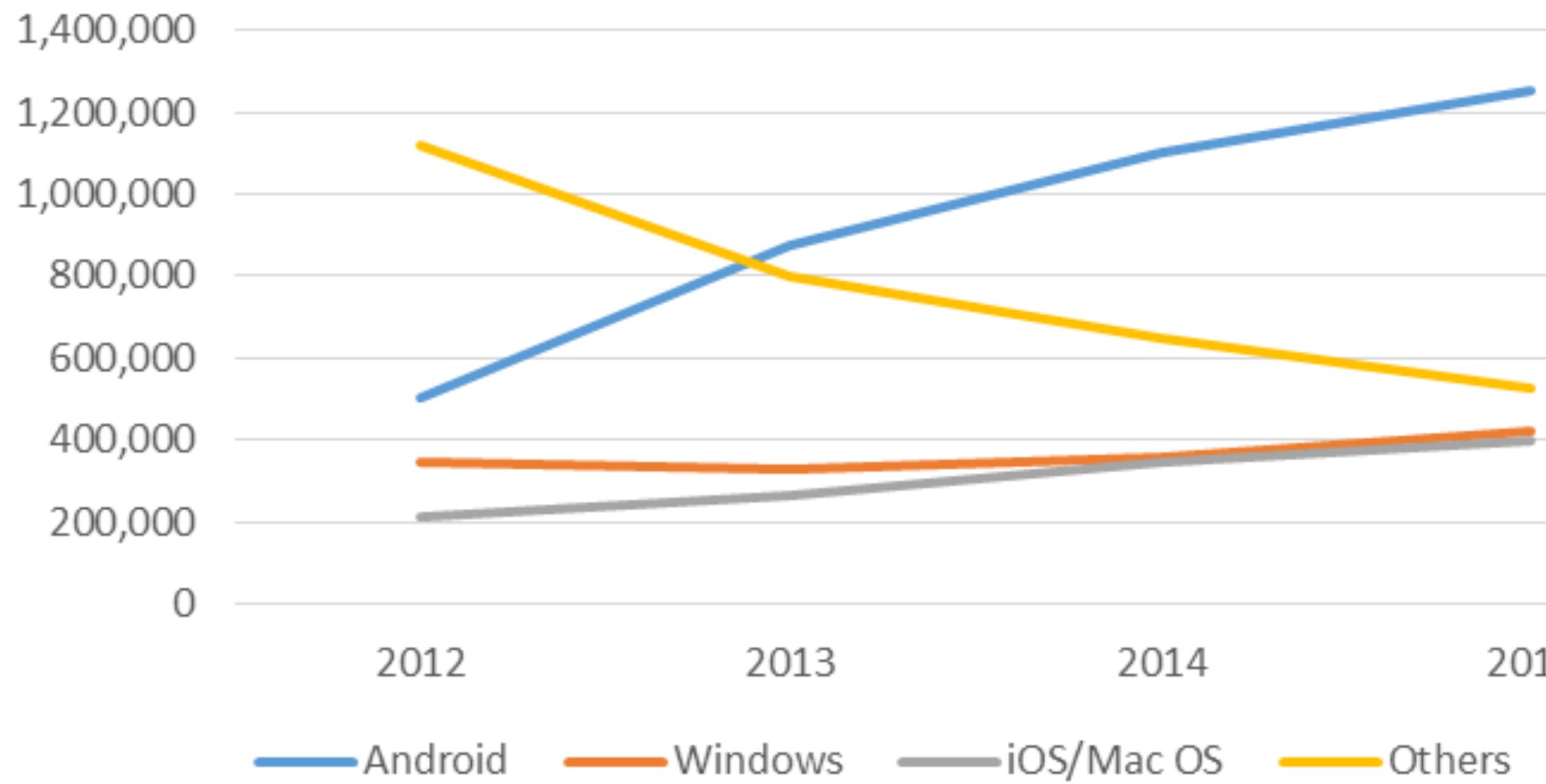
– Eric Schmidt

Vision (2011)

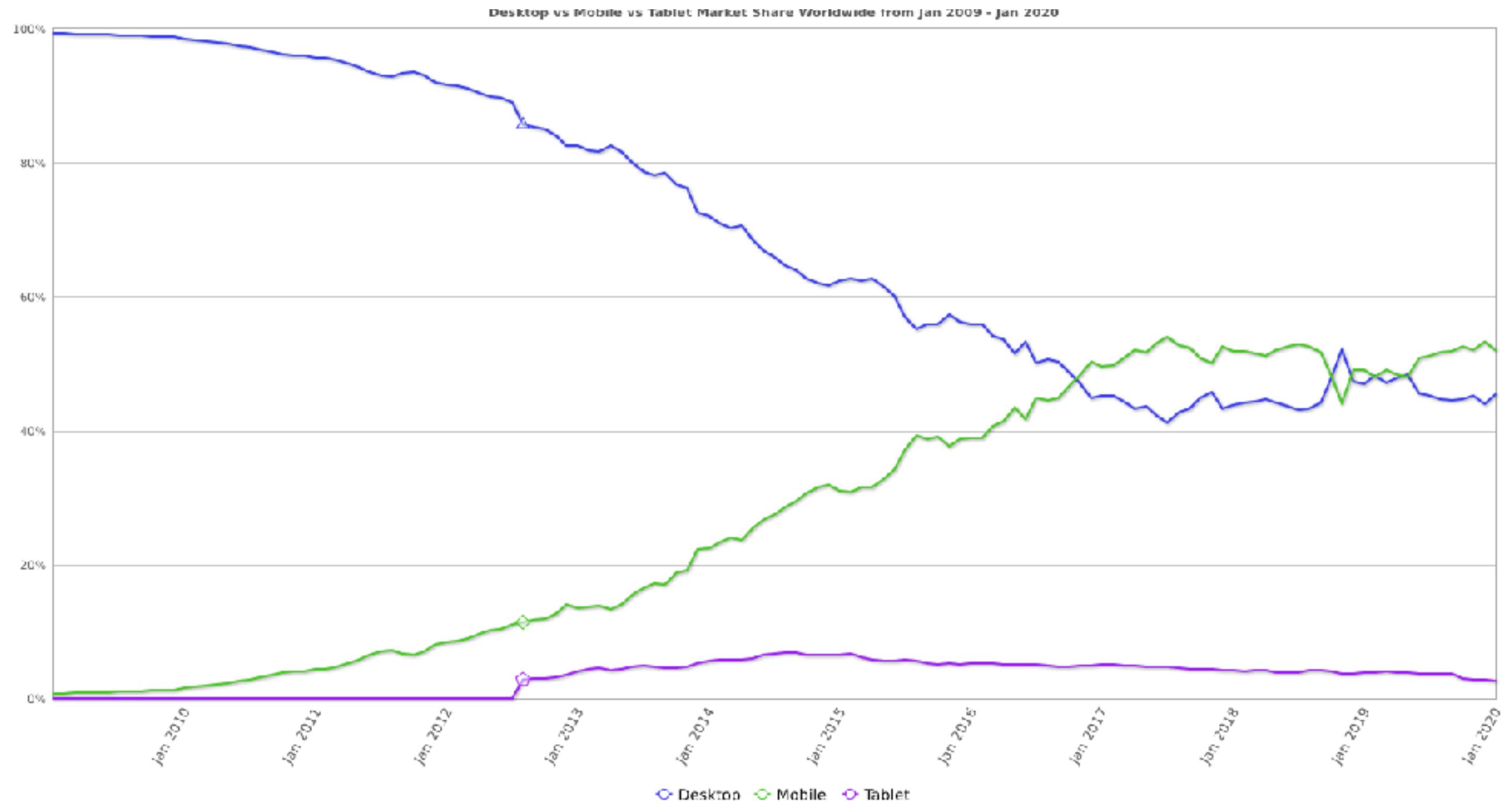
Top Five Smartphone Operating Systems, Worldwide Shipments, and Market Share, 2014Q2 (Units in Millions)

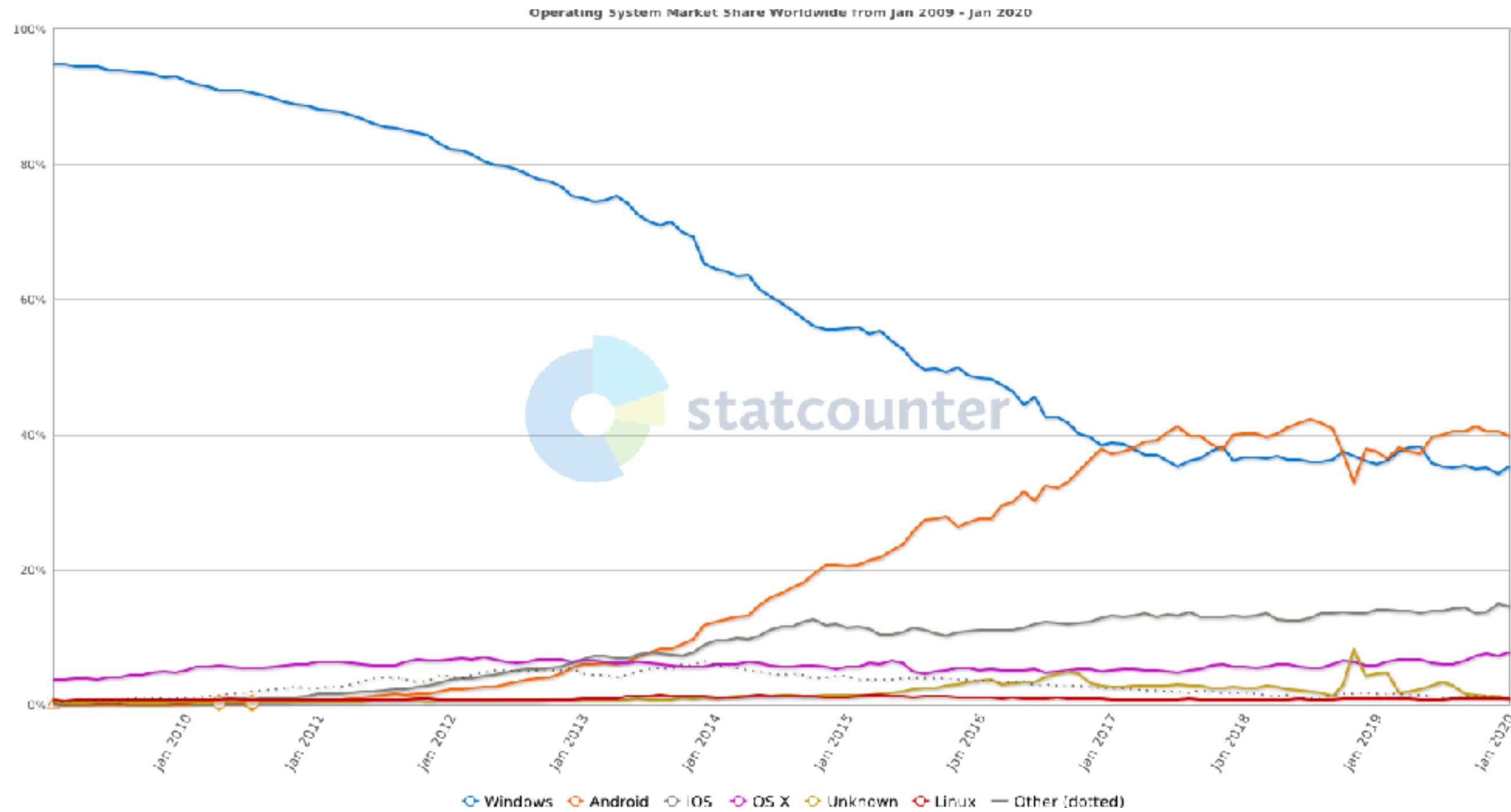
Operating System	2Q14 Shipment Volume	2Q14 Market Share	2Q13 Shipment Volume	2Q13 Market Share	2Q14/2Q13 Growth
Android	255.3	84.7%	191.5	79.6%	33.3%
iOS	35.2	11.7%	31.2	13.0%	12.7%
Windows	7.4	2.5%	8.2	3.4%	-9.4%
BlackBerry	1.5	0.5%	6.7	2.8%	-78.0%
Others	1.9	0.6%	2.9	1.2%	-32.2%
Total	301.3	100%	240.5	100%	25.3%

Worldwide Device Shipments by Operating System (Thousands of Units)



Android vs.

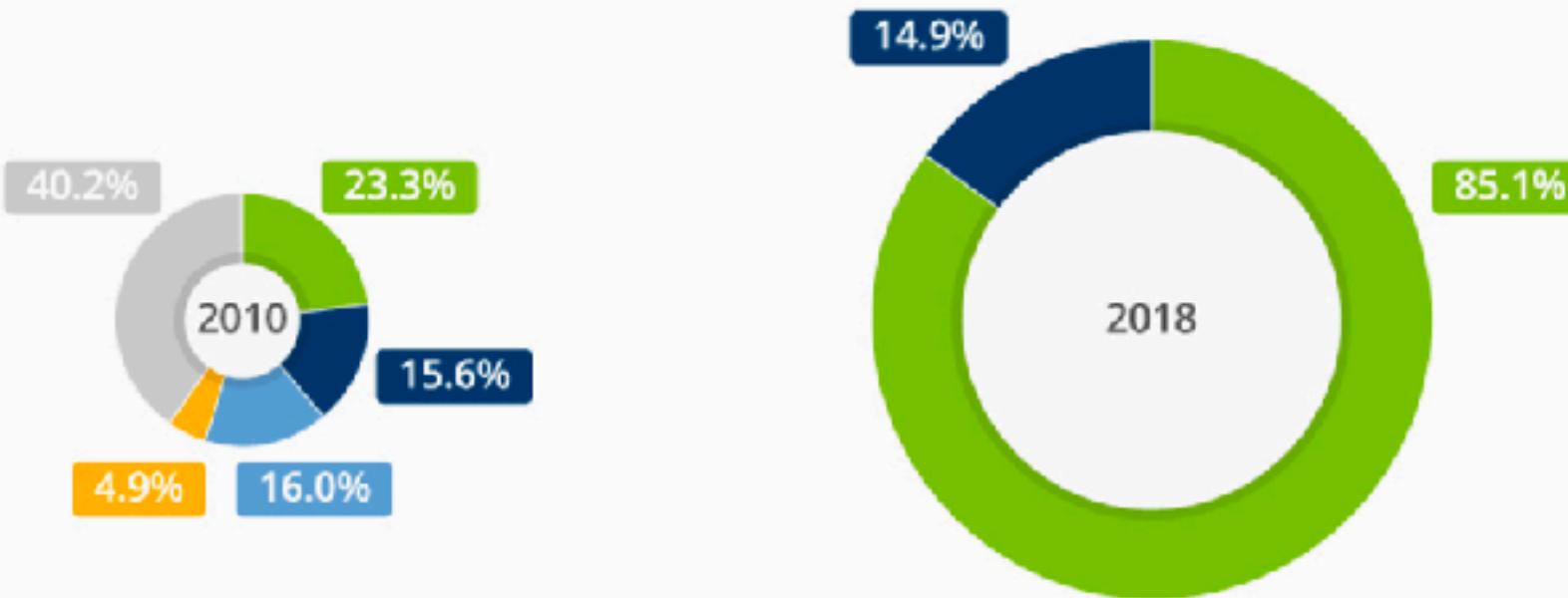




The Smartphone Duopoly

Worldwide smartphone market share by operating system (based on unit shipments)

● Android ● iOS ● BlackBerry ● Windows Phone ● Others



Total sales

305m

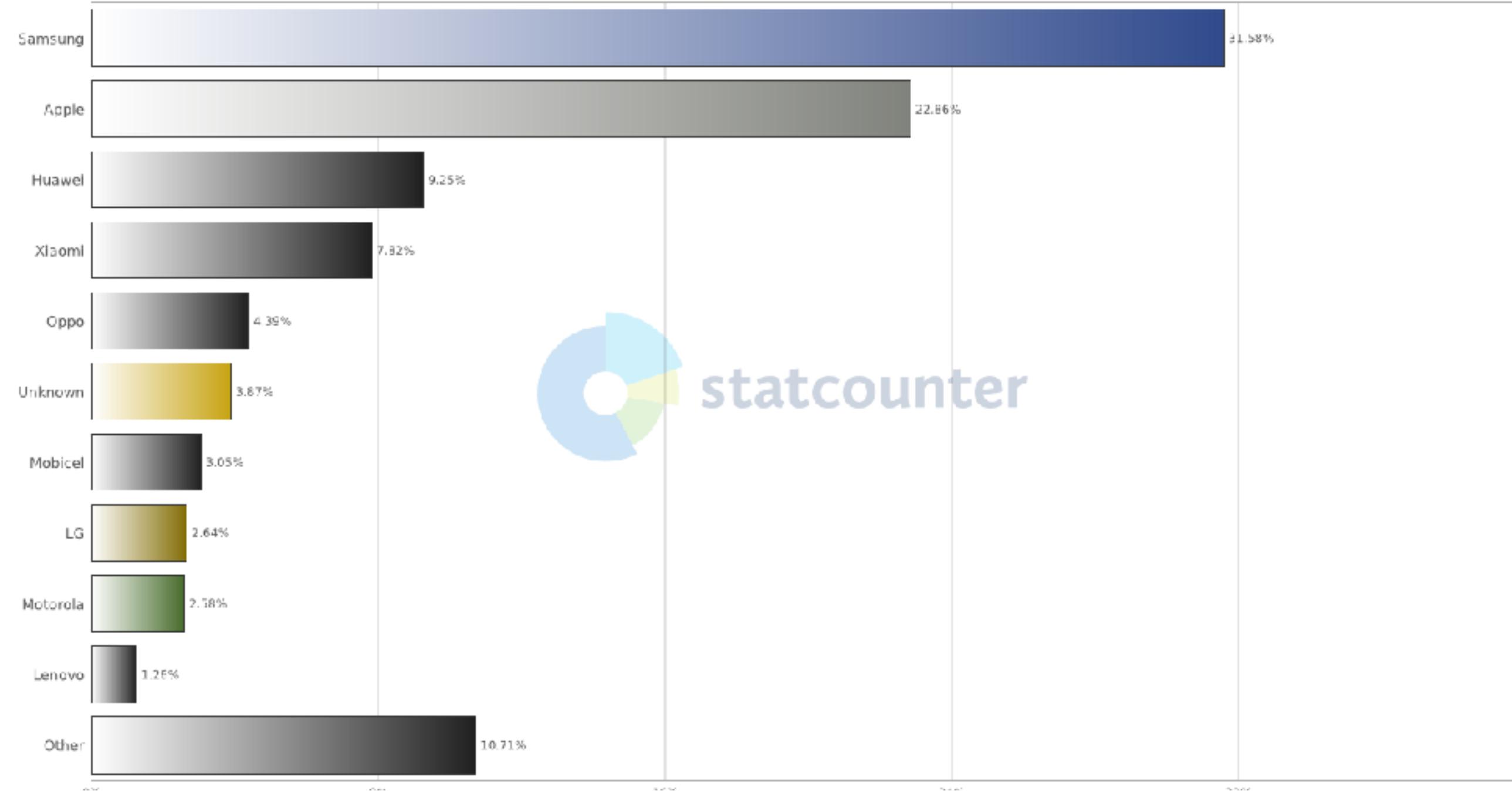
1,405m



@StatistaCharts Source: IDC

statista

Mobile Vendor Market Share Worldwide from Jan 2019 - Jan 2020



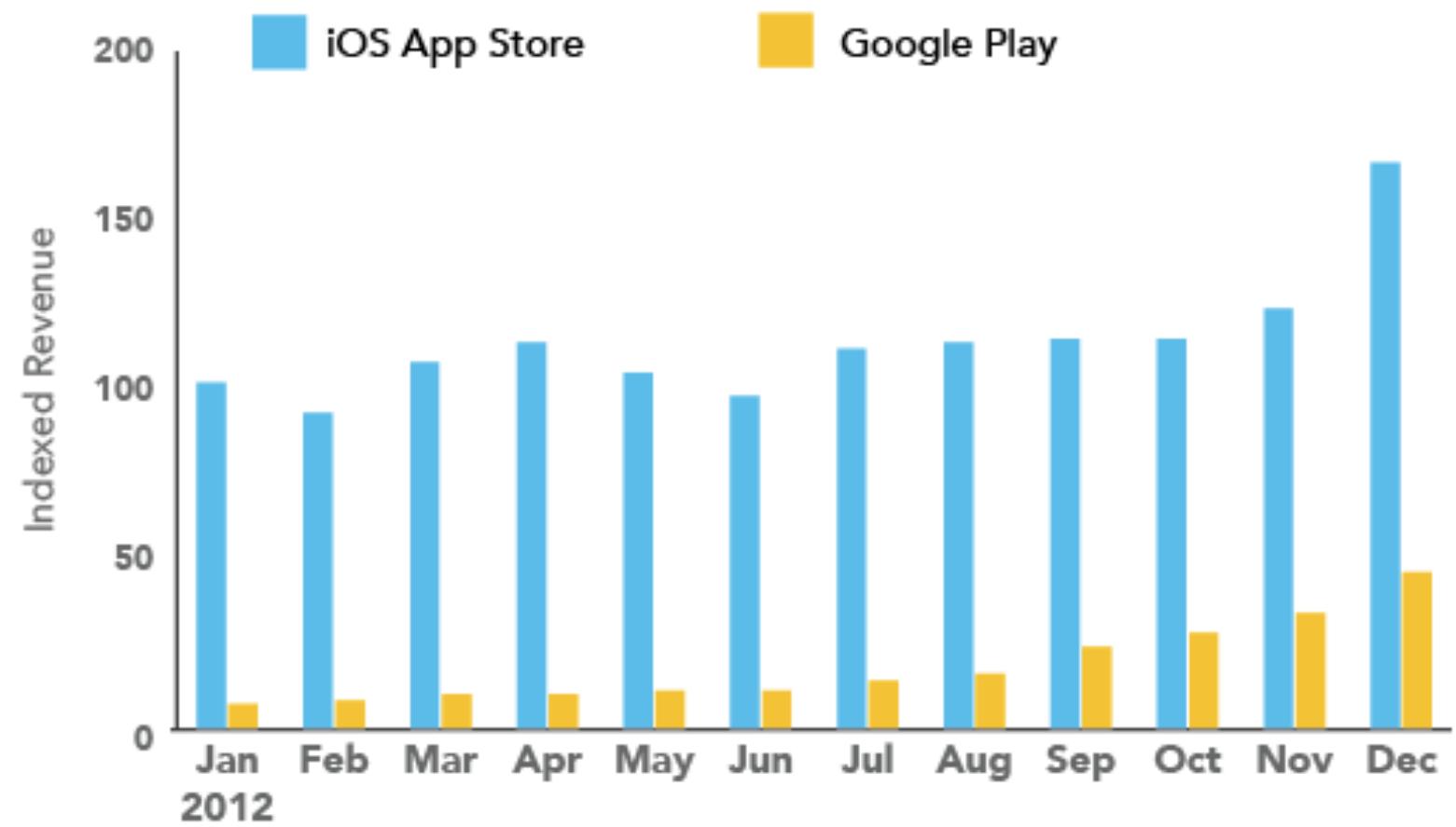
Premium Smartphone Segment Price Tier Split and OEM Share



Technology Market Research



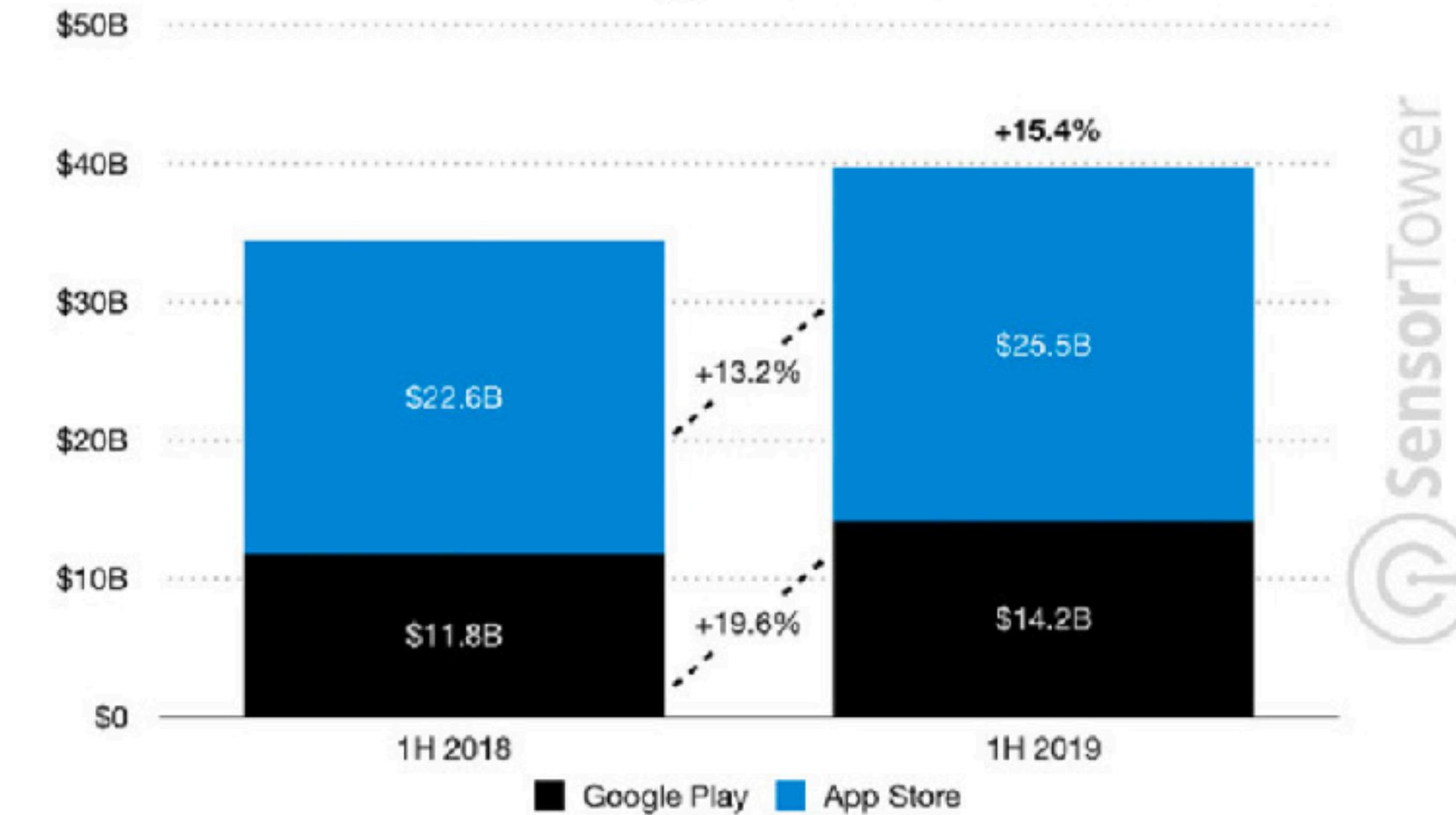
iOS App Store vs Google Play Revenue



App Annie

SOURCE: App Annie Intelligence
iOS App Store January 2012 Revenue Index set to 100

Worldwide Gross App Revenue - First Half 2019



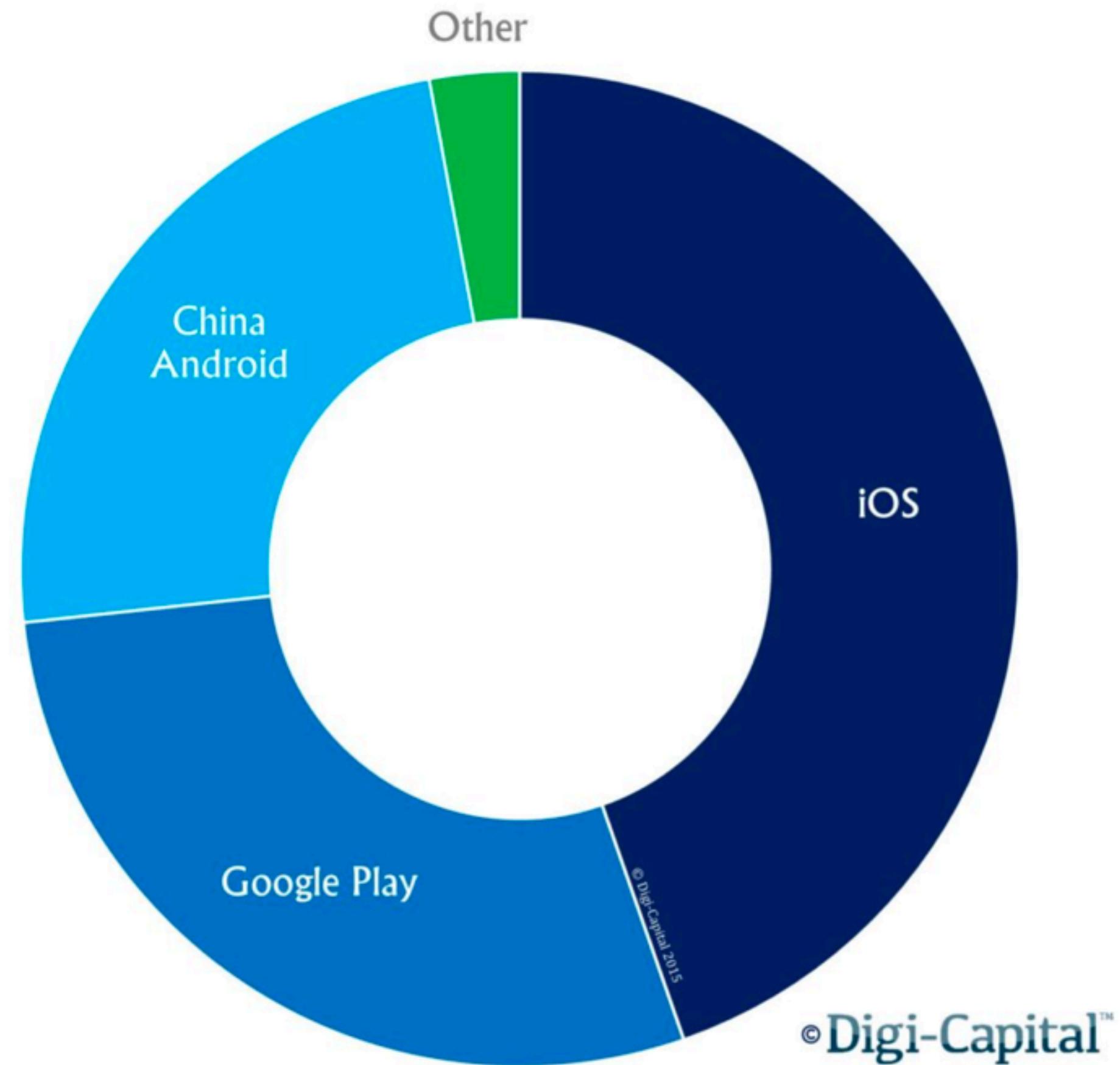
 SensorTower

Worldwide App Downloads and Revenue by Store

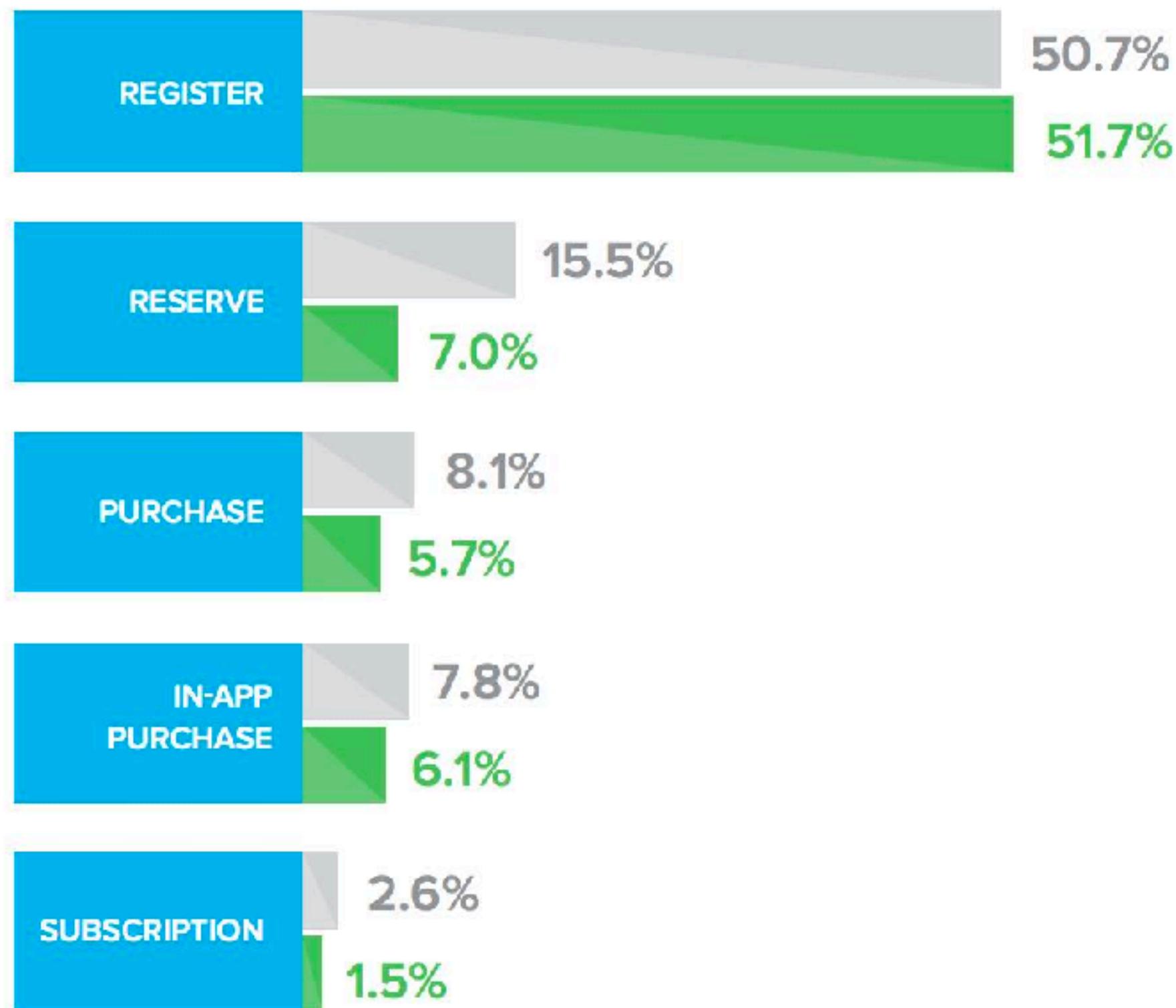


Compared to Q1 2016, the downloads gap remained the same, but iOS extended its lead over Google Play in revenue by 10 percentage points.

Global app stores revenue value share 2014



POST-INSTALL ACTIVITY ENGAGEMENT RATES



Released in

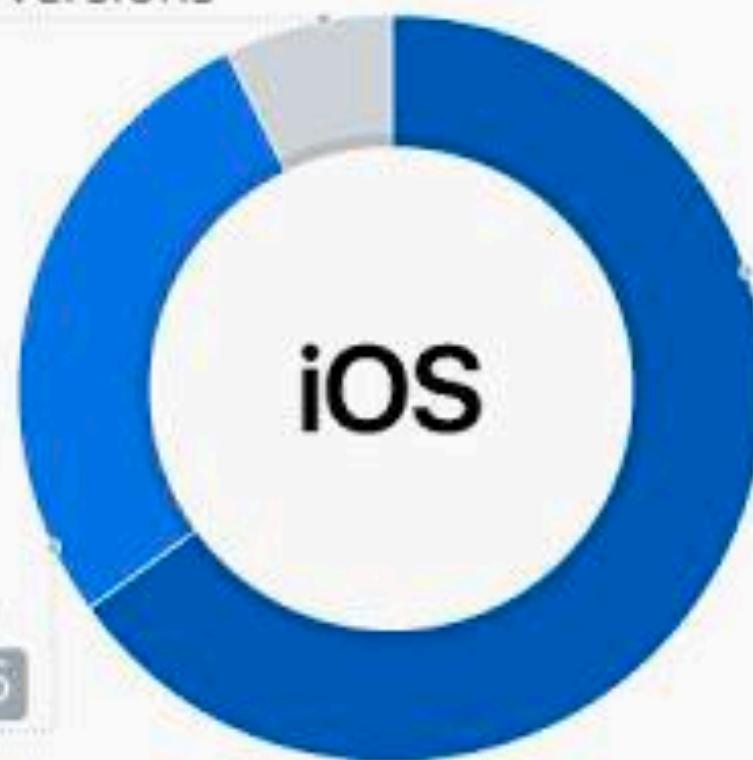
Earlier versions

7%

iOS 10

Sep 16

28%



Data collected by the App Store
on January 18, 2018

Earlier versions

17%

iOS 11

Sep 17

65%

Lollipop

Nov 14

25%



Data collected during a 7-day period ending
on February 5, 2018

Android: Pros and Cons

- Many different Android devices, more being developed all the time;
- Backed by Google, one of the world's biggest and most powerful IT companies;
- Google is deliberately “disruptive”;
- Free development environment for low cost of entry;
- Free OS for hardware developers;



Android: Pros and Cons

- Although Android can be used on many types of devices, they are not always supported by Google;
- Devices must support quite a rich mix of capabilities in order to be certified as Android compatible;
- Only “compatible” devices have access* to Google Play (the Android Market);

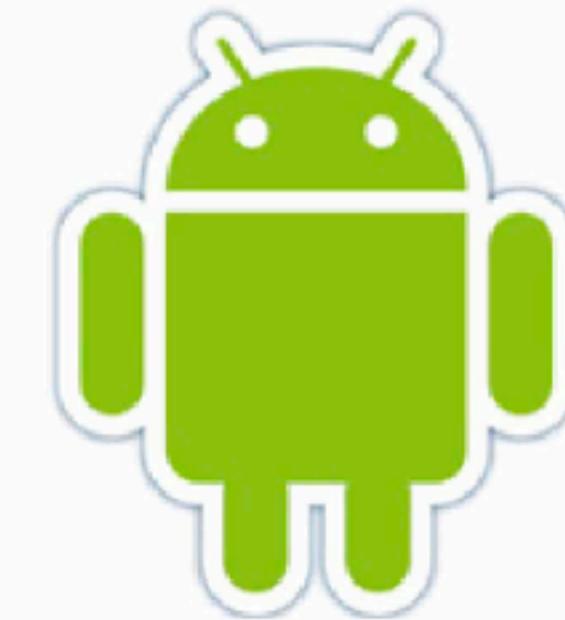
Android: Why develop?

- Google Play Apps is open to all applications
 - No screening except for illegal or malware content
 - No controls on applications which compete with Google
 - Android explicitly allows new components to replace old ones, even at the OS level
- The downside is that some apps are very poor quality, check user feedback scores
- But Google is implementing a review process for publication on the store.

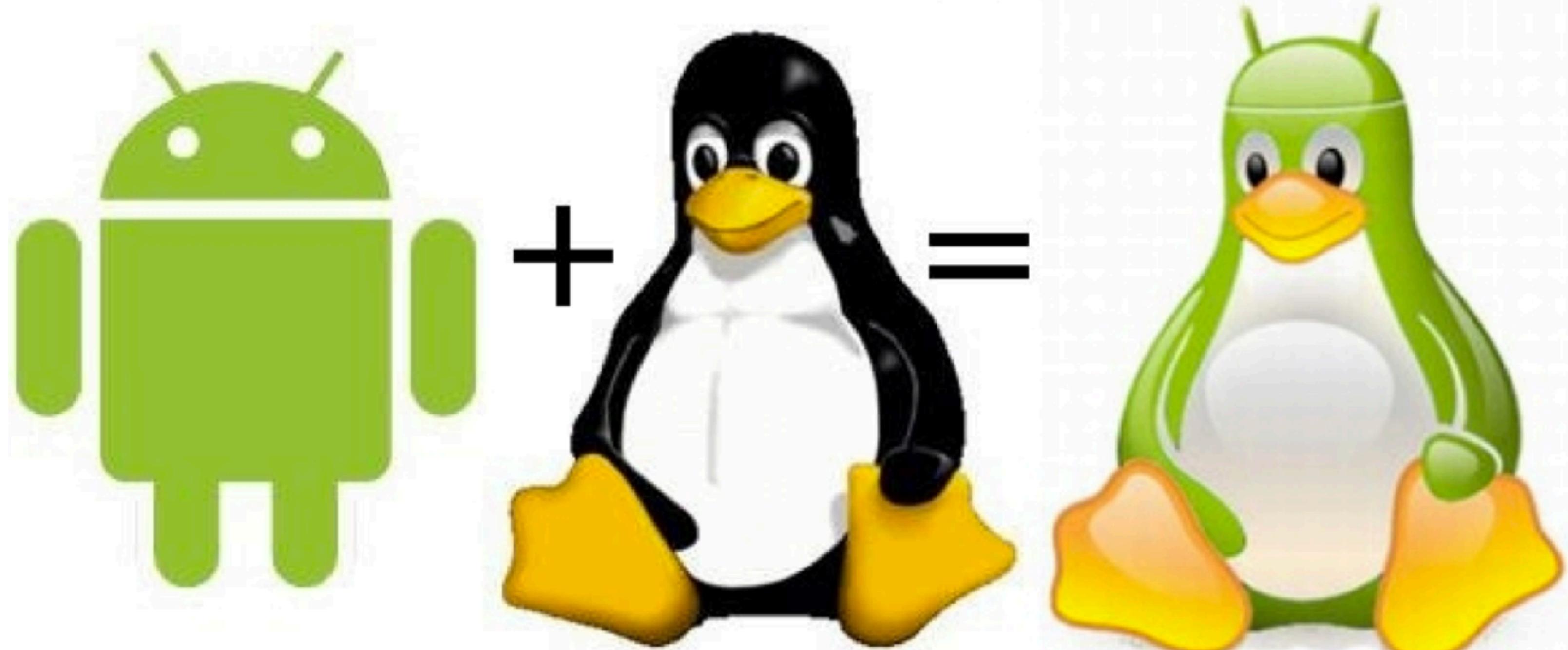
Android: Success

- Android is tightly regulated by Google at the hardware level
- But Android is an open system for developers
- The success of Android has surprised many
- Much of the success may be down to:
 - Apple's iPhone, opening a new market sector
 - Apple and its over-controlling nature and expensive price
 - Microsoft's inactivity and ball-dropping
 - Google's aggressive development





Architectures



Android: Programming Model

- Android is built on:
 - Linux kernel
 - The Dalvik Runtime (Java SE compatible) or the Android runtime (ART) on Lollipop
 - The Android APIs
 - A minimum-spec hardware platform
- Android software is:
 - Multi-threaded
 - Interruptible
 - Written in standard Java/Kotlin

Pre-requisites for development

- Need a java JDK (Java SDK (7):
www.oracle.com/technetwork/java/javase/downloads/index.html)
 - Contains all the java commands, compiler, and more
- Need the Android SDK
 - Unique java functionality for mobile apps
- Android Studio
(developer.android.com/sdk/installing/studio.html)
 - Ide for development



Android Garden

ANDROID VERSIONS LIST: A COMPLETE HISTORY & FEATURES



Cupcake
1.5



Donut
1.6



Eclair
2.0/2.1



Froyo
2.2



Gingerbread
2.3



Honeycomb
3.0/3.1



Ice Cream Sandwich
4.0



Jelly Bean
4.1/4.2/4.3



KitKat
4.4



Lollipop
5.0



Marshmallow
6.0



Nougat
7.0



Oreo
8.0



Pie
9.0

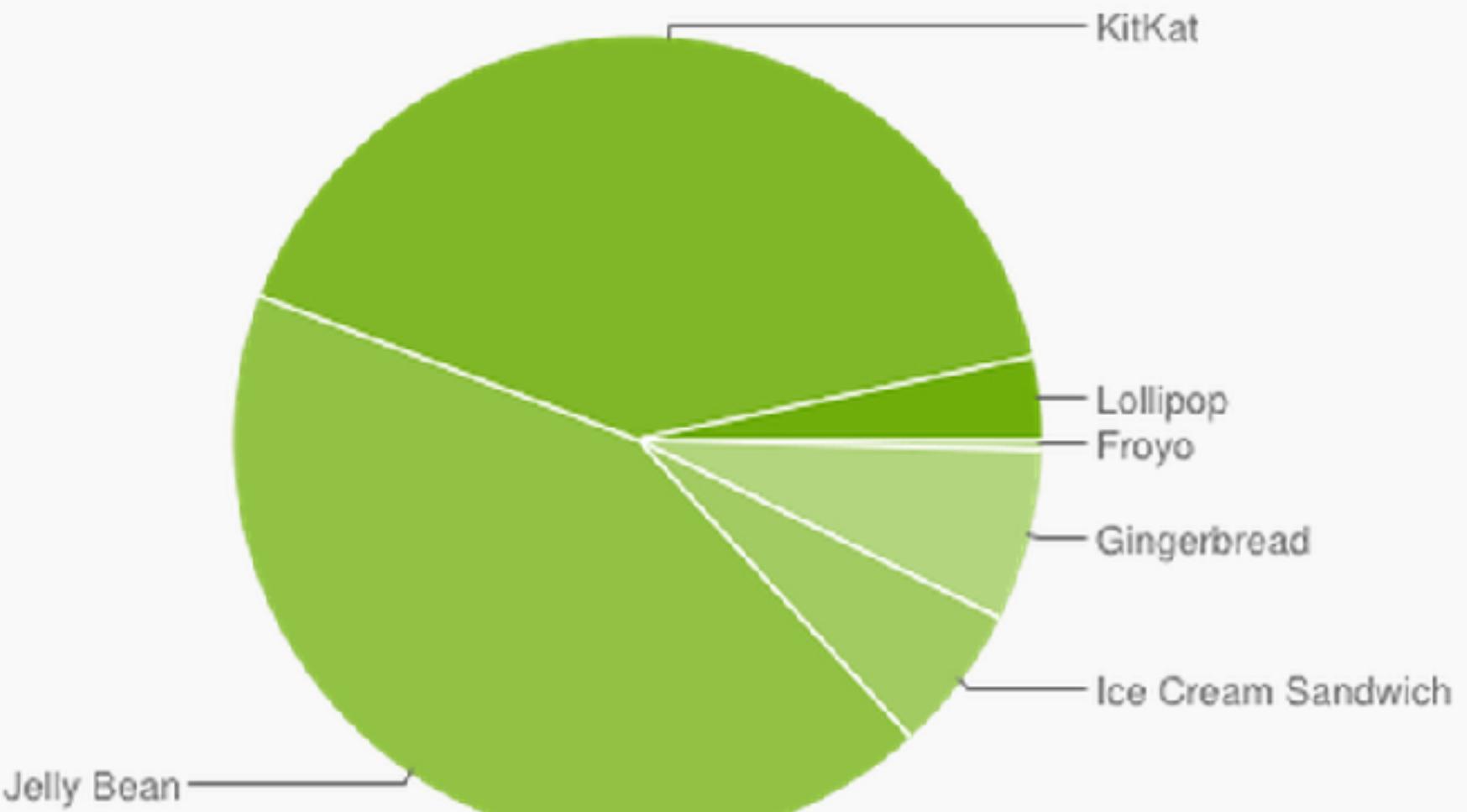


10



android

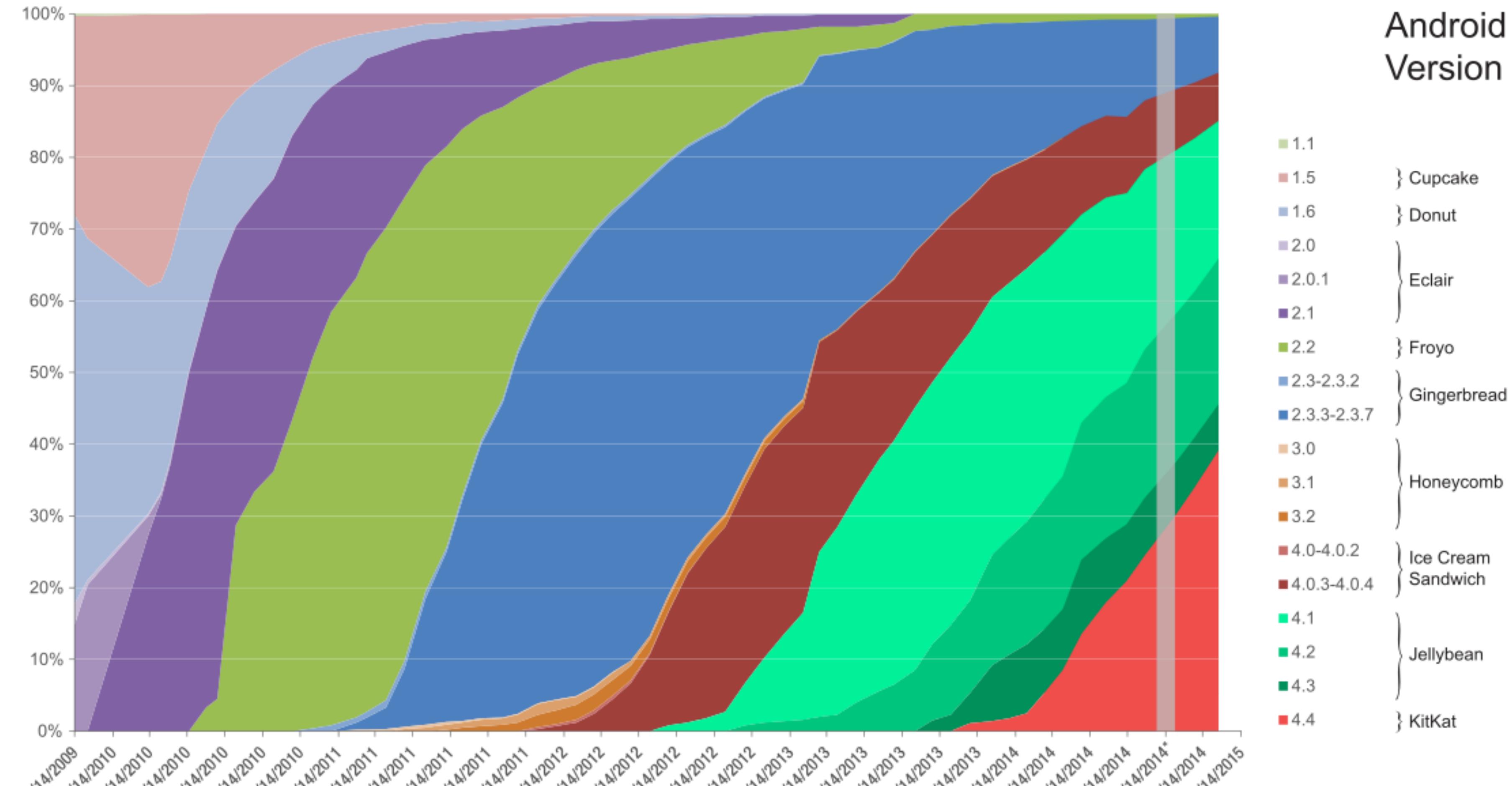
Version	Codename	API	Distribution
2.2	Froyo	8	0.4%
2.3.3 - 2.3.7	Gingerbread	10	6.9%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	5.9%
4.1.x	Jelly Bean	16	17.3%
4.2.x		17	19.4%
4.3		18	5.9%
4.4	KitKat	19	40.9%
5.0	Lollipop	21	3.3%



*Data collected during a 7-day period ending on March 2, 2015.
Any versions with less than 0.1% distribution are not shown.*

Using the compatibility (support) libraries, you can develop in 5.0 and min-build-

Android Version

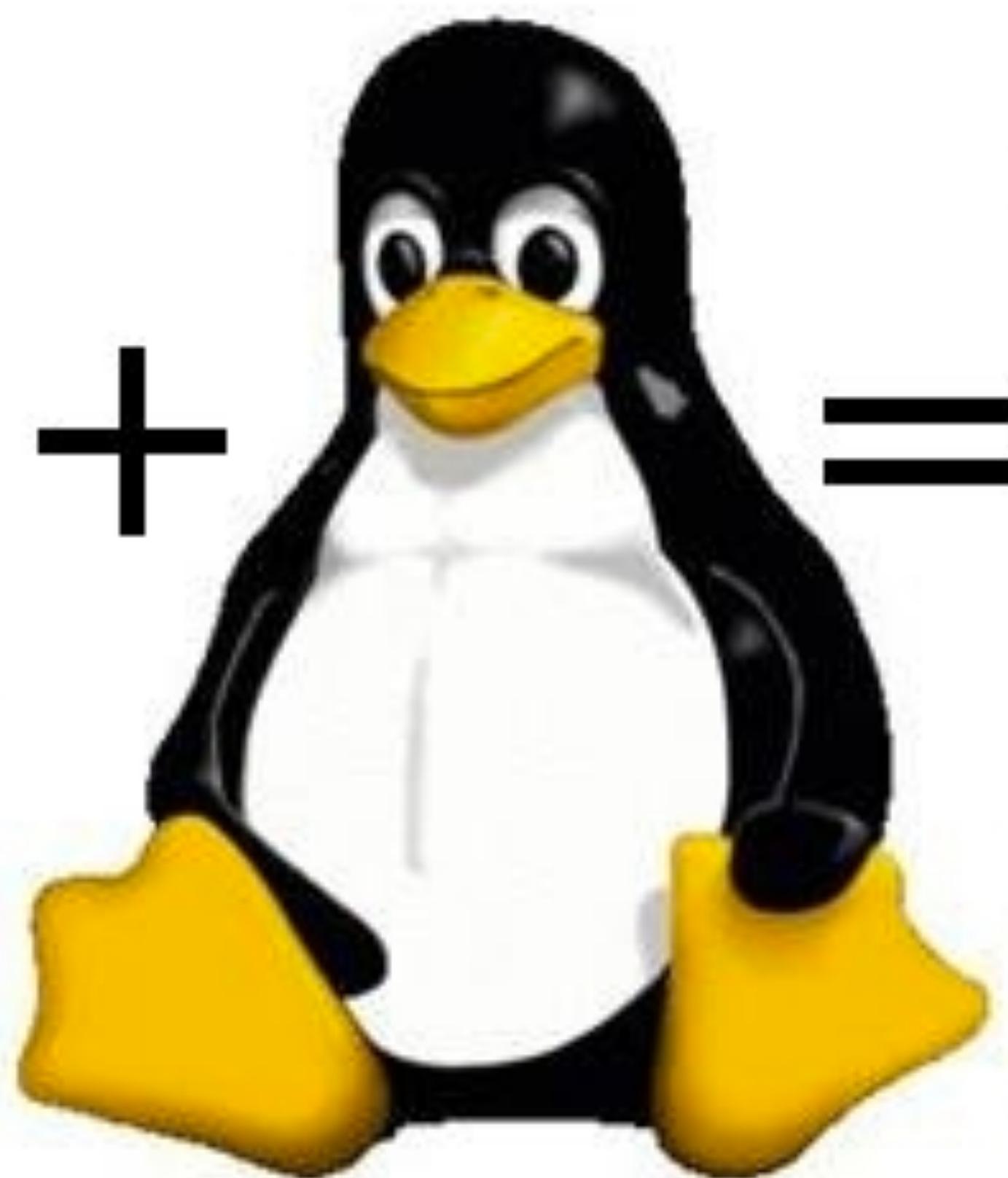


Code/Version/API map

Code name	Version	API level
Lollipop	5.1	API level 22
Lollipop	5.0	API level 21
KitKat	4.4 - 4.4.4	API level 19
Jelly Bean	4.3.x	API level 18
Jelly Bean	4.2.x	API level 17
Jelly Bean	4.1.x	API level 16
Ice Cream Sandwich	4.0.3 - 4.0.4	API level 15, NDK 8
Ice Cream Sandwich	4.0.1 - 4.0.2	API level 14, NDK 7
Honeycomb	3.2.x	API level 13
Honeycomb	3.1	API level 12, NDK 6
Honeycomb	3.0	API level 11
Gingerbread	2.3.3 - 2.3.7	API level 10
Gingerbread	2.3 - 2.3.2	API level 9, NDK 5
Froyo	2.2.x	API level 8, NDK 4
Eclair	2.1	API level 7, NDK 3
Eclair	2.0.1	API level 5
Eclair	2.0	API level 5
Donut	1.6	API level 4, NDK 2
Cupcake	1.5	API level 3, NDK 1
(no code name)	1.1	API level 2
(no code name)	1.0	API level 1



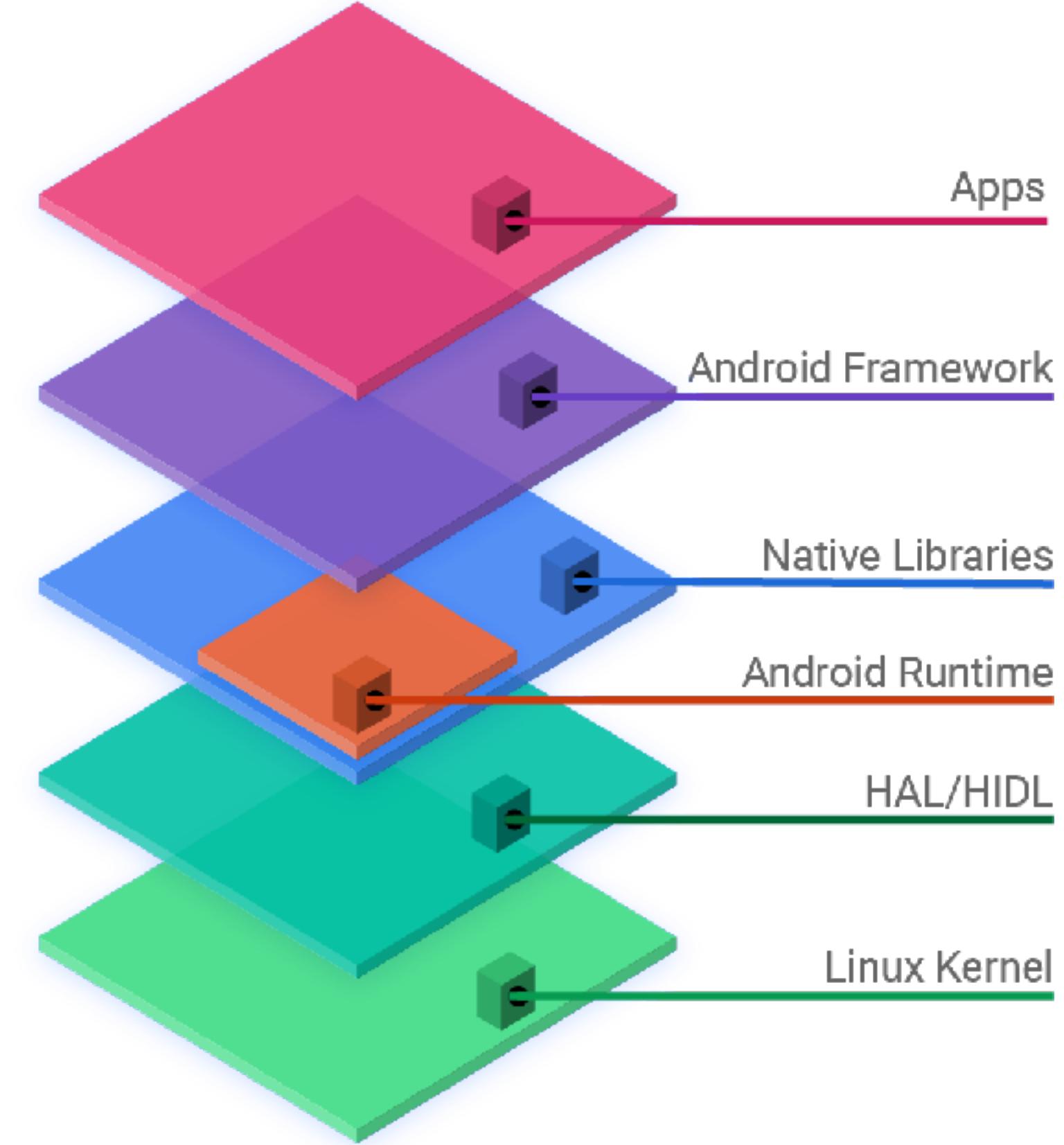
+

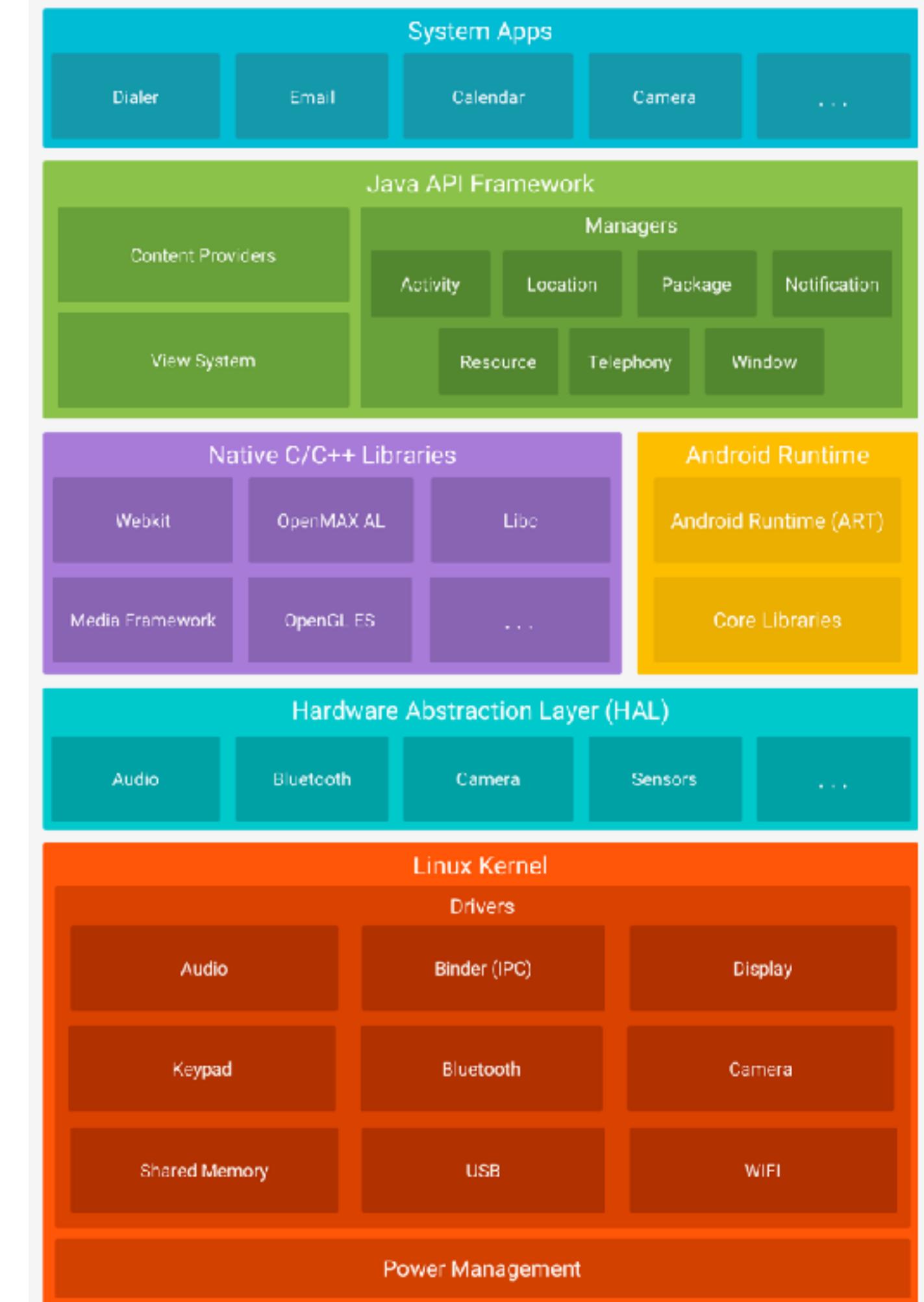


=



Android Architecture

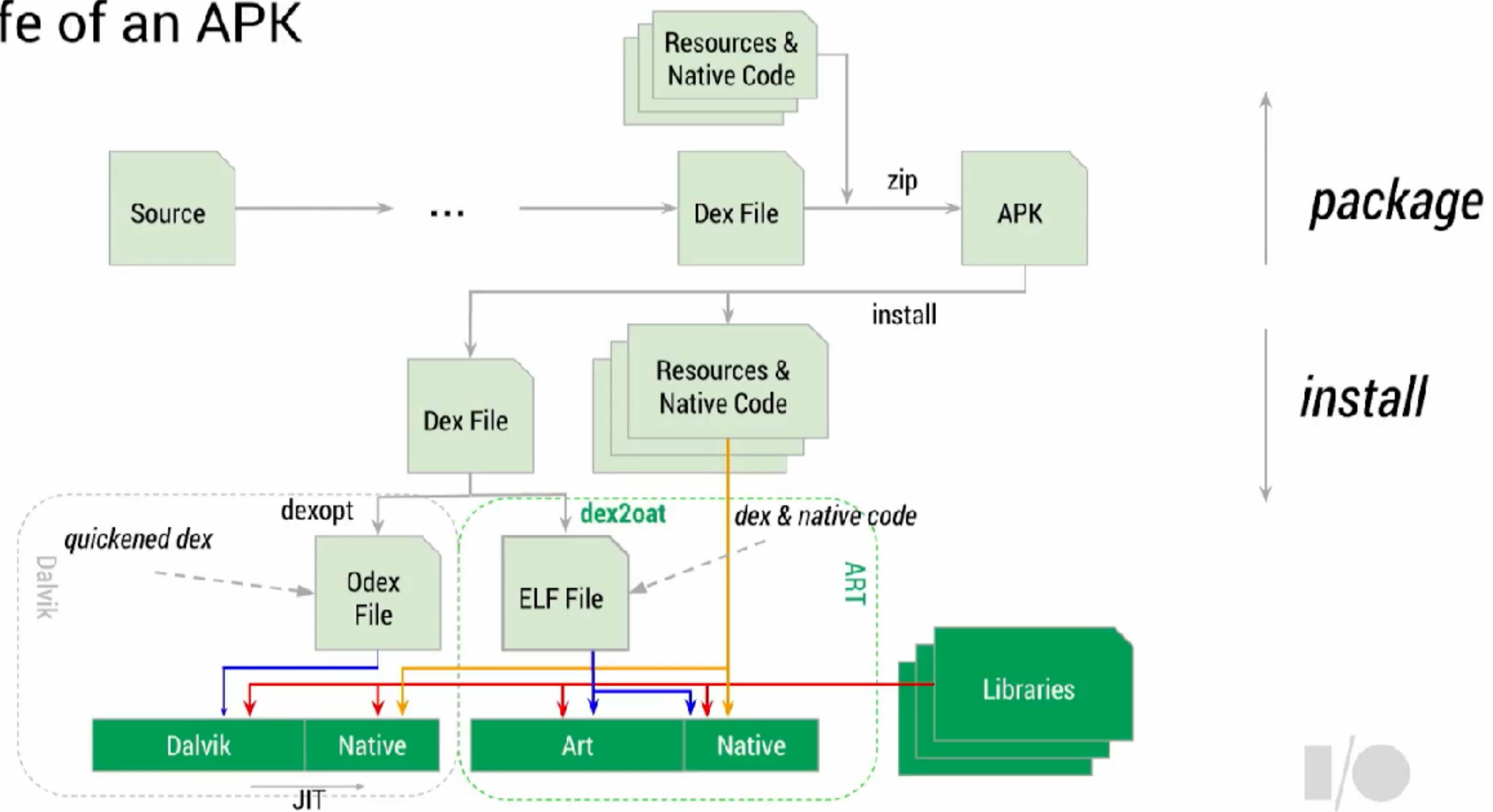




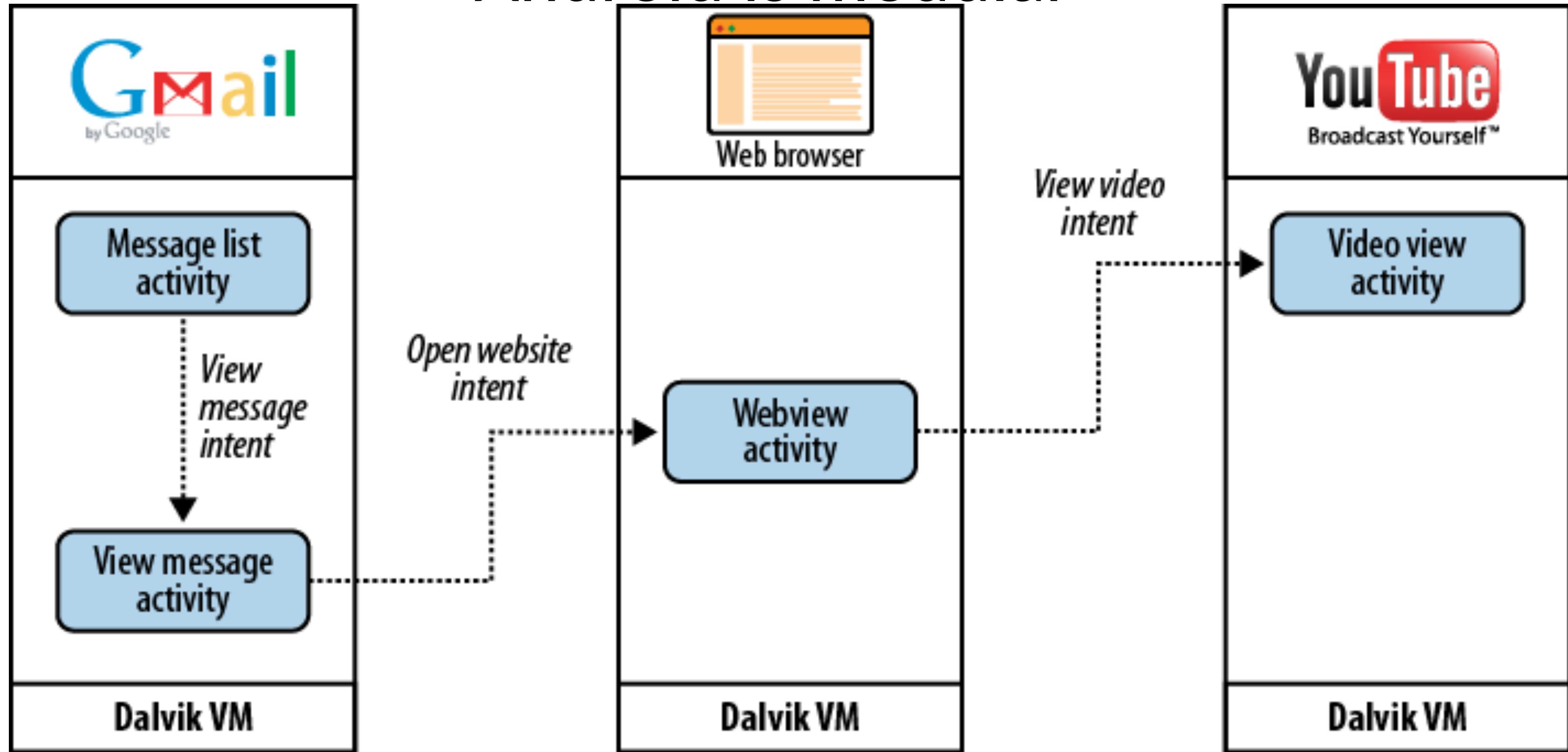
Architecture

- Android consists of a [kernel](#) based on the [Linux kernel](#),
 - with [middleware](#), [libraries](#) and [APIs](#) written in [C](#)
 - and [application software](#) running on an [application framework](#) which includes Java-compatible libraries based on [Apache Harmony](#).
- Android uses the [Dalvik virtual machine](#) with [just-in-time compilation](#) to run compiled [Java](#) code or the ART android runtime.

The life of an APK



Android is modular



Application Components



Activity

Present a visual user interface for one focused endeavor the user can undertake

Example: a list of menu items users can choose from



Services

Run in the background for an indefinite period of time

Example: music on background, calculate and provide the result to activities that need it



Broadcast Receivers

Receive and react to broadcast announcements

Example: battery low, announcements that the time zone has changed



Content Providers

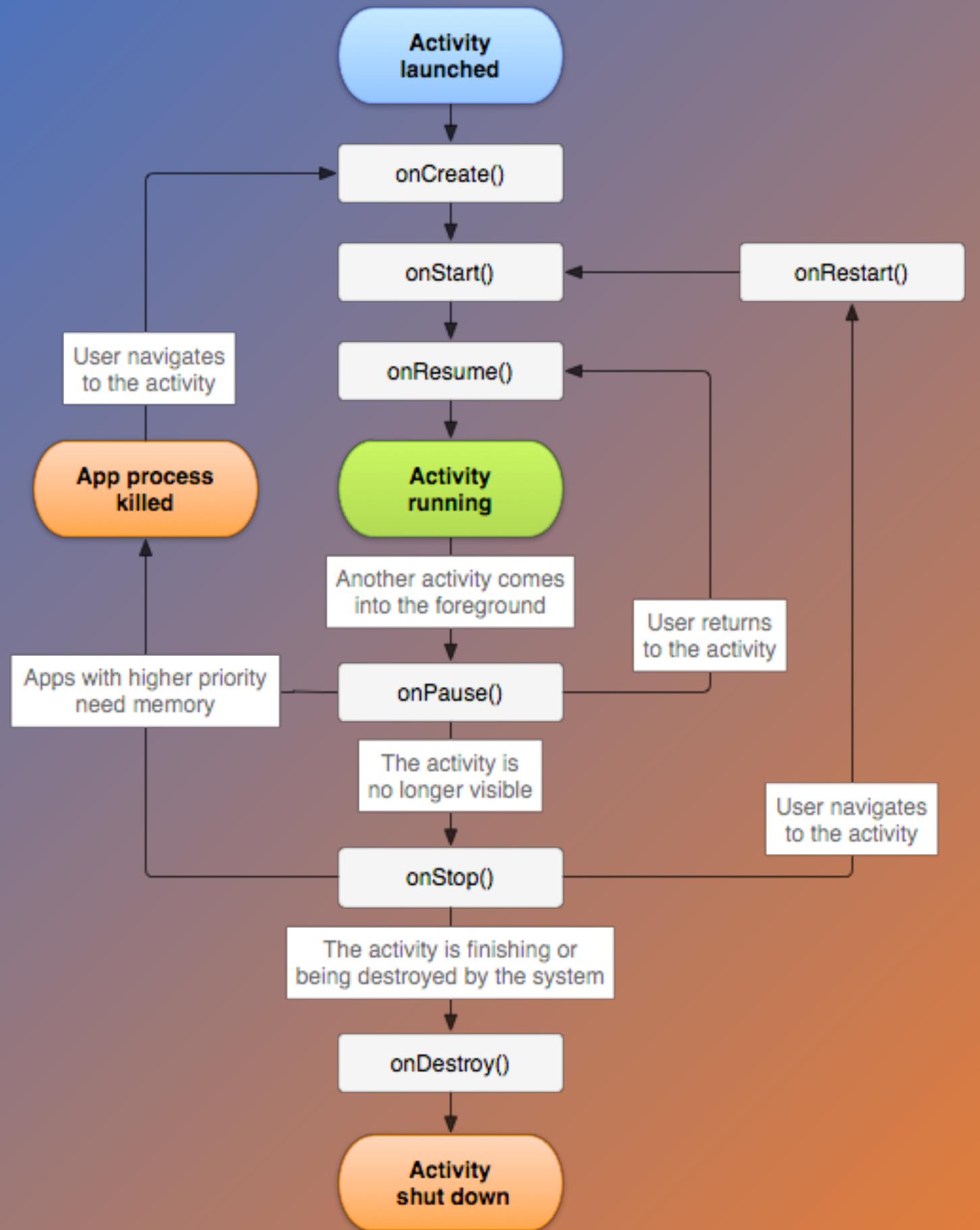
Store and retrieve data and make it accessible to all applications

Example: Android ships with a number of content providers for common data types (e.g., audio, video, images, personal contact information, etc.)



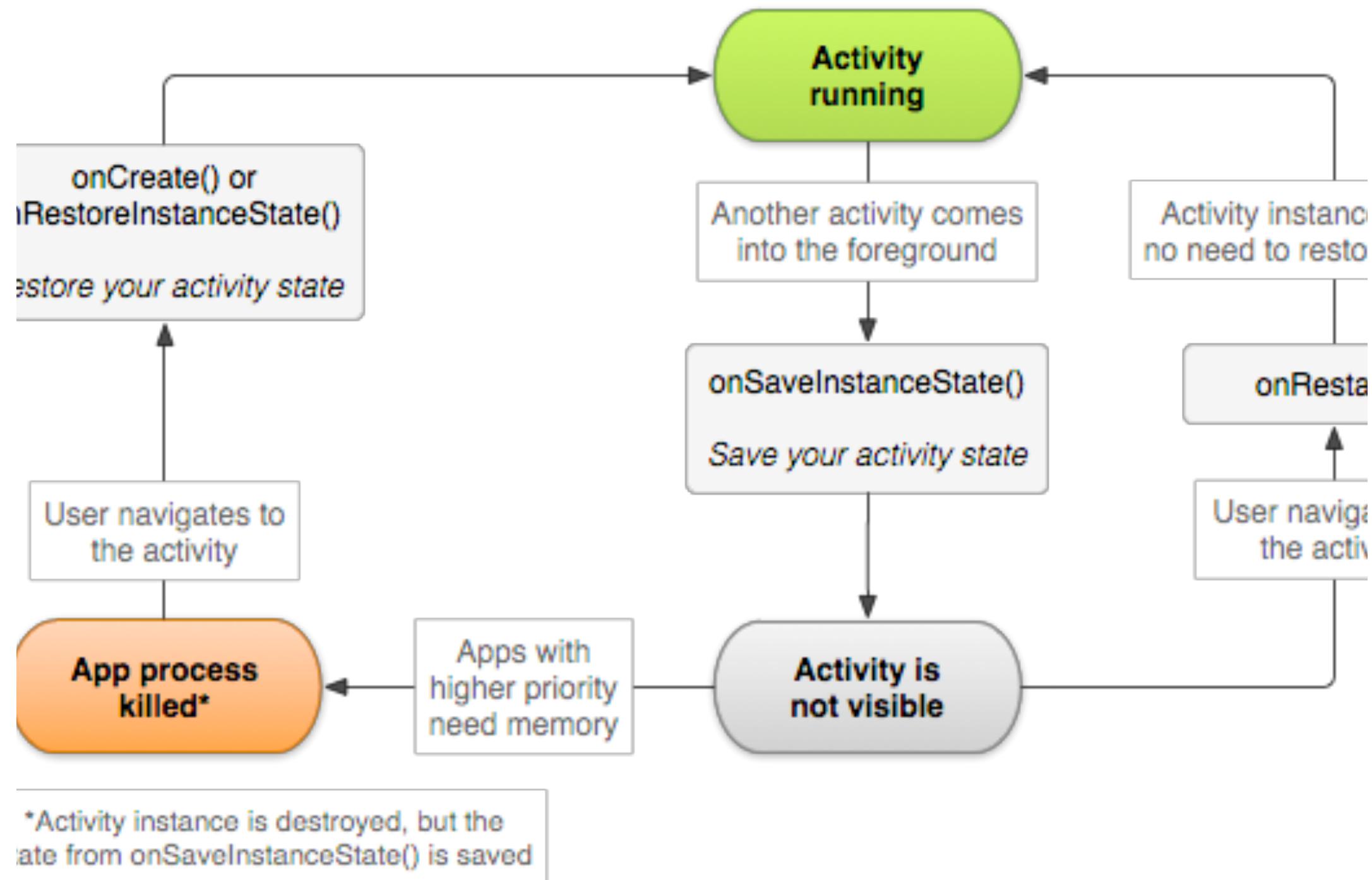
Intents

Hold the content of a message
Example: convey a request for an activity to present an image to the user or let the user edit some text



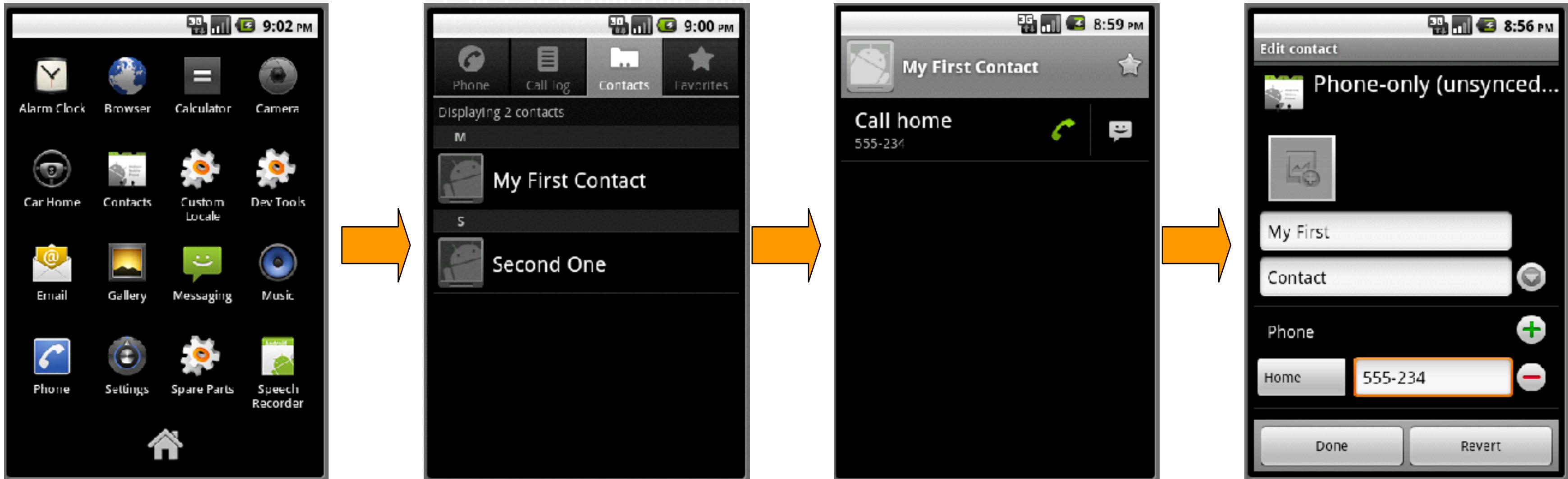
Activity

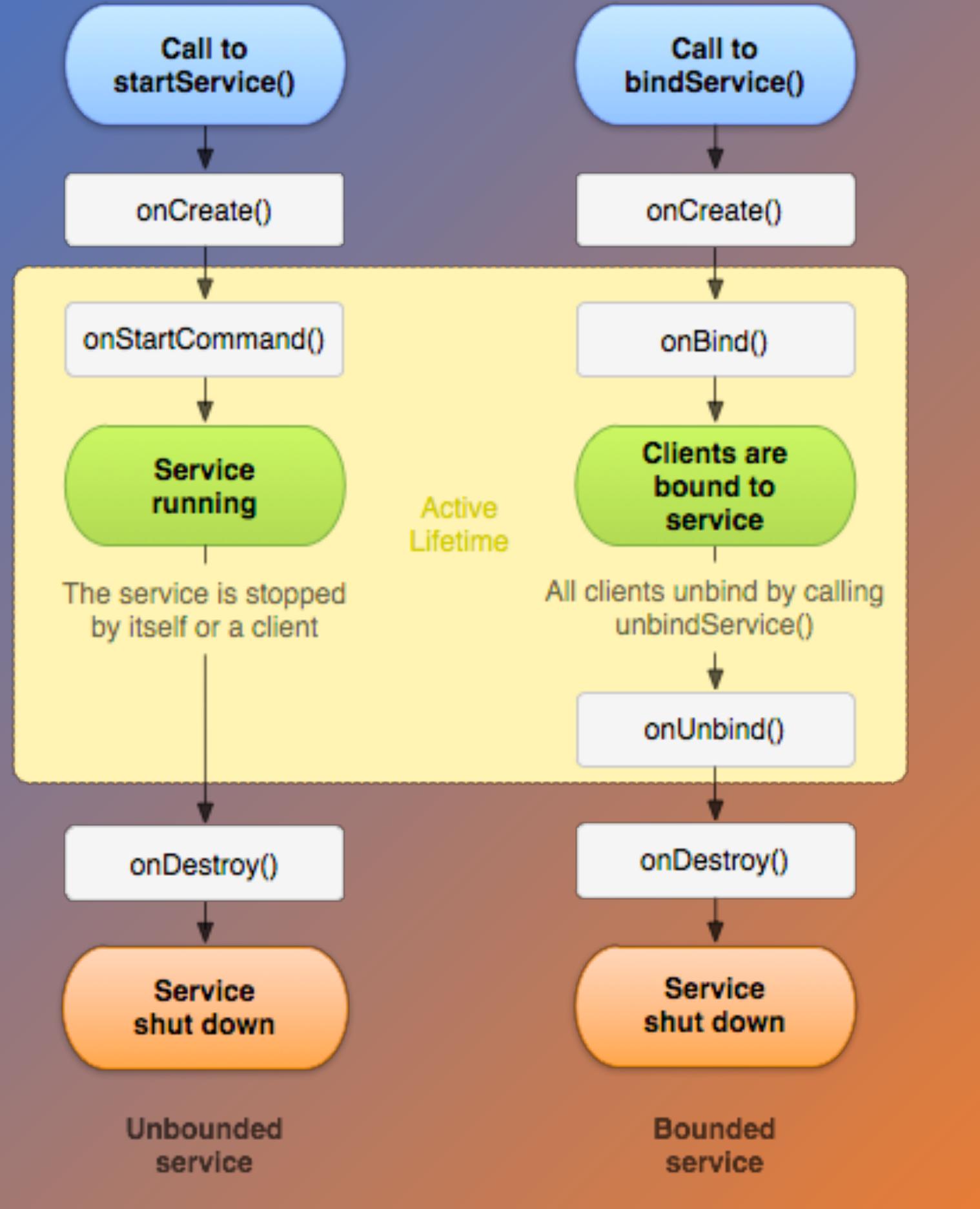
- System upcalls component as its state changes due to user actions.
- If another activity is started, the activity is paused.
- If a paused activity is not visible to the user, it is stopped.
- A stopped activity may be destroyed.
- And its app process may be killed.



Saving/restoring activity state

Activities start each other

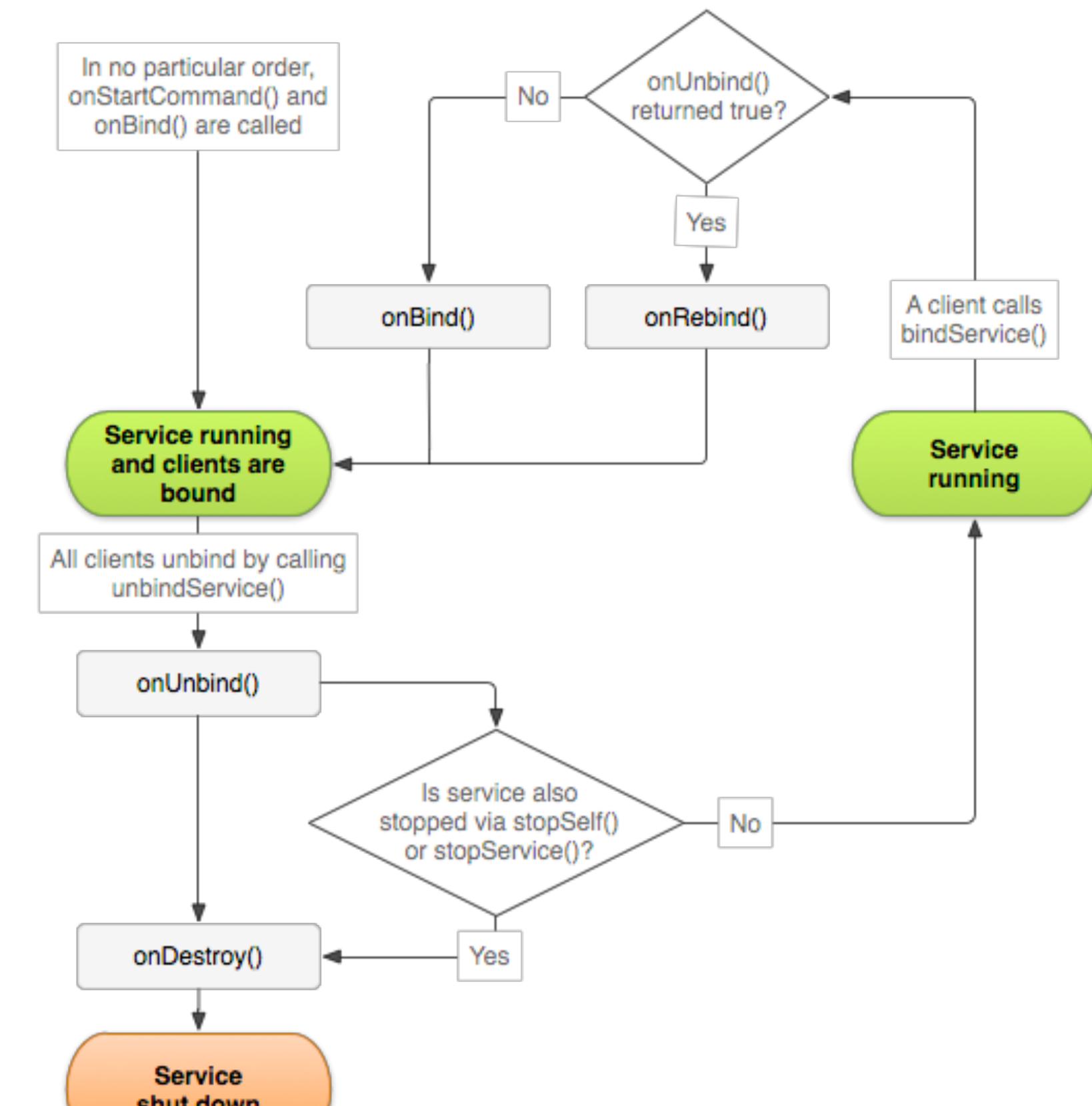




Service

- Services advertise one or more binder endpoints.
- Clients choose to bind/unbind (or unbind when stopped).
- A service with no bound clients may be shut down.

Service



In-class exercise 2: Prototyping

- ◆ Group of 1/2/3
- ◆ Think of an app that you would like to build.
- ◆ Build the screens of an App with Android Studio (minimum 3);
- ◆ Use components of android
- ◆ <https://www.figma.com/community/file/1101784689160904535/mobile-app-design-figma-tutorial-for-beginners> play around with this tutorial a little bit.