

Android Development

Mark Joselli

mark.joselli@gmail.com

, ,

Who am I?

Presentation

- Professor at PUC-PR
- Chief-Research-Officer of Nullpointer
- Researcher in GPGPU, games and mobile
- Developer of mobile Apps for more than 17 years
- pHd and MD in Computer Science



a desaprovação. A cada seis semanas mil metros de Penstrução de Cefet. e Tecnog Cefet. emplada na ograma Na-

transformação econômica que virá com a implementação do Comperj. Todos esses setores dependerão de mão de obra qualificada, que vamos formar no novo instituto federal.

Ainda não há custo definido para a obra, que será bancada integralmente com recursos

viaremos as documentações necessárias.

Ano passado, durante visita de uma comitiva do Congresso Nacional às instalações da Petrobras na Ilha do Fundão, no Rio de Janeiro, um diretor pediu uma indicação para uma vaga técnica na

sil, o do Comperj, e temos que ter mão de obra capaz de atender à demanda do setor. Niterói é uma cidade polo e deve ter esse papel reafirmado — afirma o deputado federal Chico D'Angelo (PT), que negociou com Fernando Haddad, ex-ministro da Educa-

ção de mão de obra qualificada. Além disso, logística e construção civil devem também ser implantados — diz o presidente da Comissão de Educação da Câmara dos Vereadores, Vitor Junior (PT), responsável pela organização da grupo que discutirá os cursos oferecidos. ■

Pedro Teixeira

Incubadoras são o ensaio para a criação da primeira empresa

de ajuda empreendedores a ingressarem no mercado de trabalho

Schmitt
globo.com.br

uma boa ideia para criar um negócio, mas não têm dinheiro ou um espaço para trabalhar.

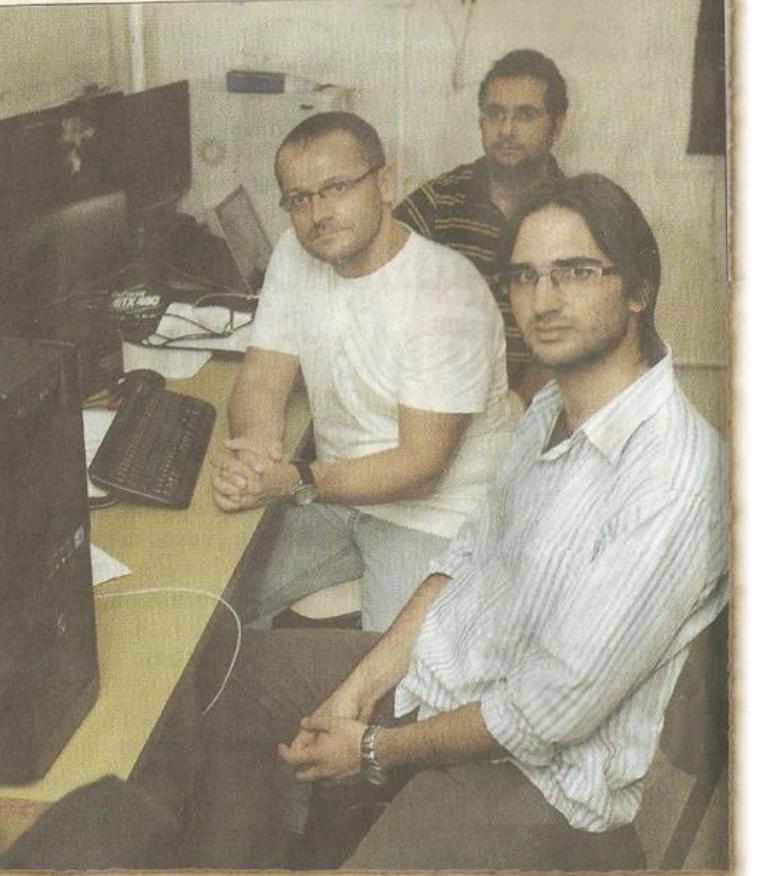
Marcelo Zamith, de 35 anos, Mark Joselli, de 30, e José Ricardo da Silva Junior, de 31, conheciam-se no doutorado em Computação. Desde 2008, o trio pesquisa Tecnologia de Computação em Alta Performance para identificar a topografia de poços de petróleo. O estudo rendeu frutos e, em 2009, eles foram contratados pela Petrobras.

— Aplicamos placas gráficas

óleo. Um processo que chegava a levar três dias passou a ser feito em horas — conta Zamith.

O projeto de pré-incubação precede a criação de uma empresa na estrutura da universidade. O coordenador acadêmico da Incubadora da UFF, Sérgio Mecena, explica que os empreendedores aproveitam a estrutura da instituição para consolidar seu modelo de negócio.

— Eles recebem todo o apoio necessário para que possam sobreviver no mercado. Passam a ter assessoria contábil, jurídica, marketing e



- Newspaper of my first startup.



MobileTv

I was part of the team that developed the first TV for mobile of Latin America



Got the Best pHD Thesis in Brazil of 2014



- One of my students received the Apple Design Awards for best game



SBGames

Teresina-Pi 2015

Festival de Jogos

Melhor Serious Game

Wyz

- Another has received best series game for a game that teach deaf children how to read.

And a group of
students were
hackaton
winners



Where do I
work?

PUCPR

- Pontifical Catholic University of Paraná
- Located in Curitiba in the State of Paraná
- More than 31,000 students
- Five academic units:
 - the Center for Biological and Health Sciences,
 - the Center for Exact Sciences and Technology,
 - the Center for Juridical and Social Sciences,
 - the Center for Humanities and Theology,
 - and the Business School.



PUCPR

- PUCPR offers graduation, MBA, Master and pHDs courses in different fields;
 - 63 graduation courses;
 - 175 pos graduation courses
 - and more than 2,000 R&D projects with more that 100 patents
- PUCPR has a central biblioteca, laboratories equipped with computers, projectors
- Focus on Teaching, Research, Innovation and Internationalization



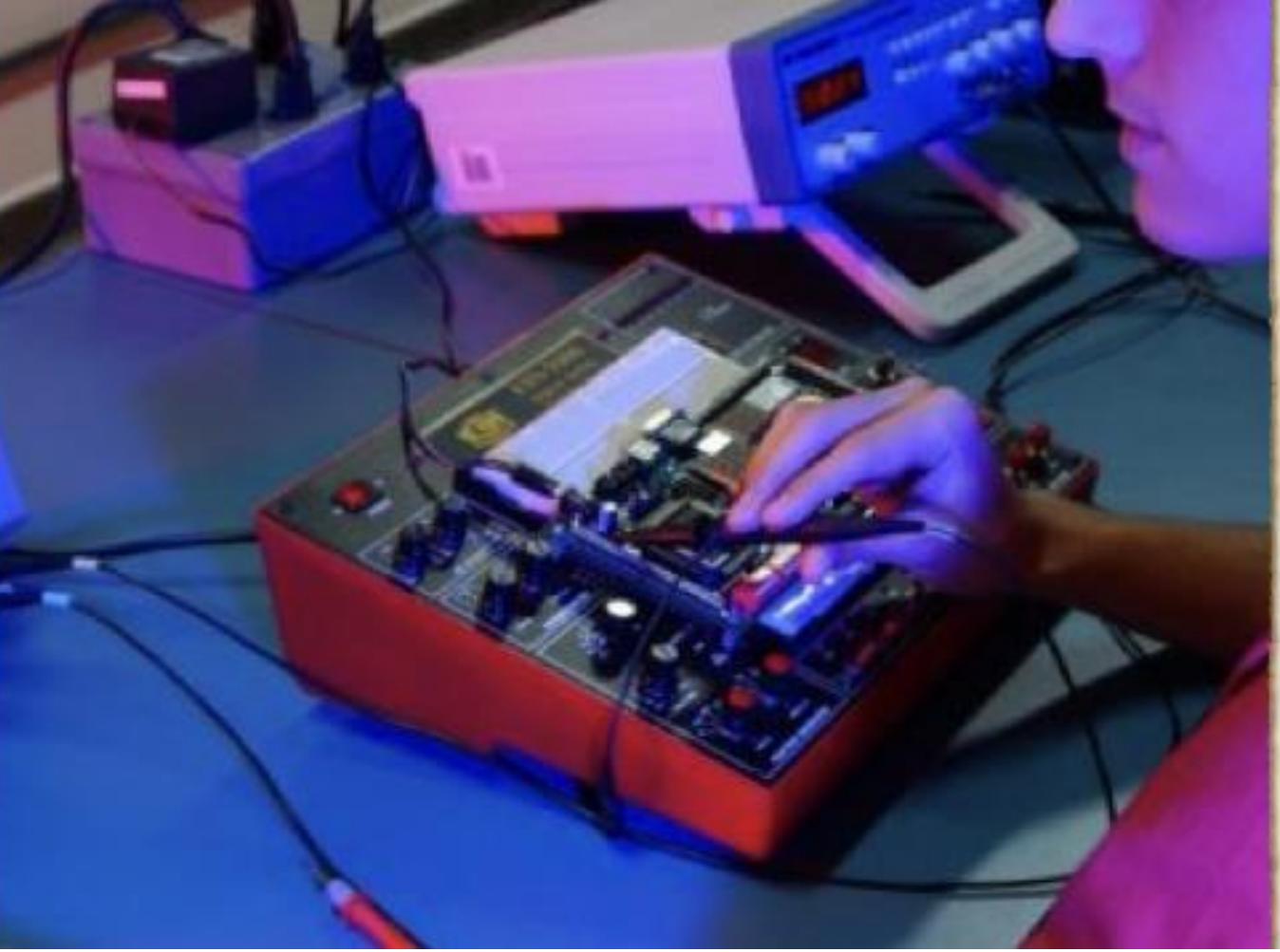
NOKIA

Equipment Room



MIC CWB

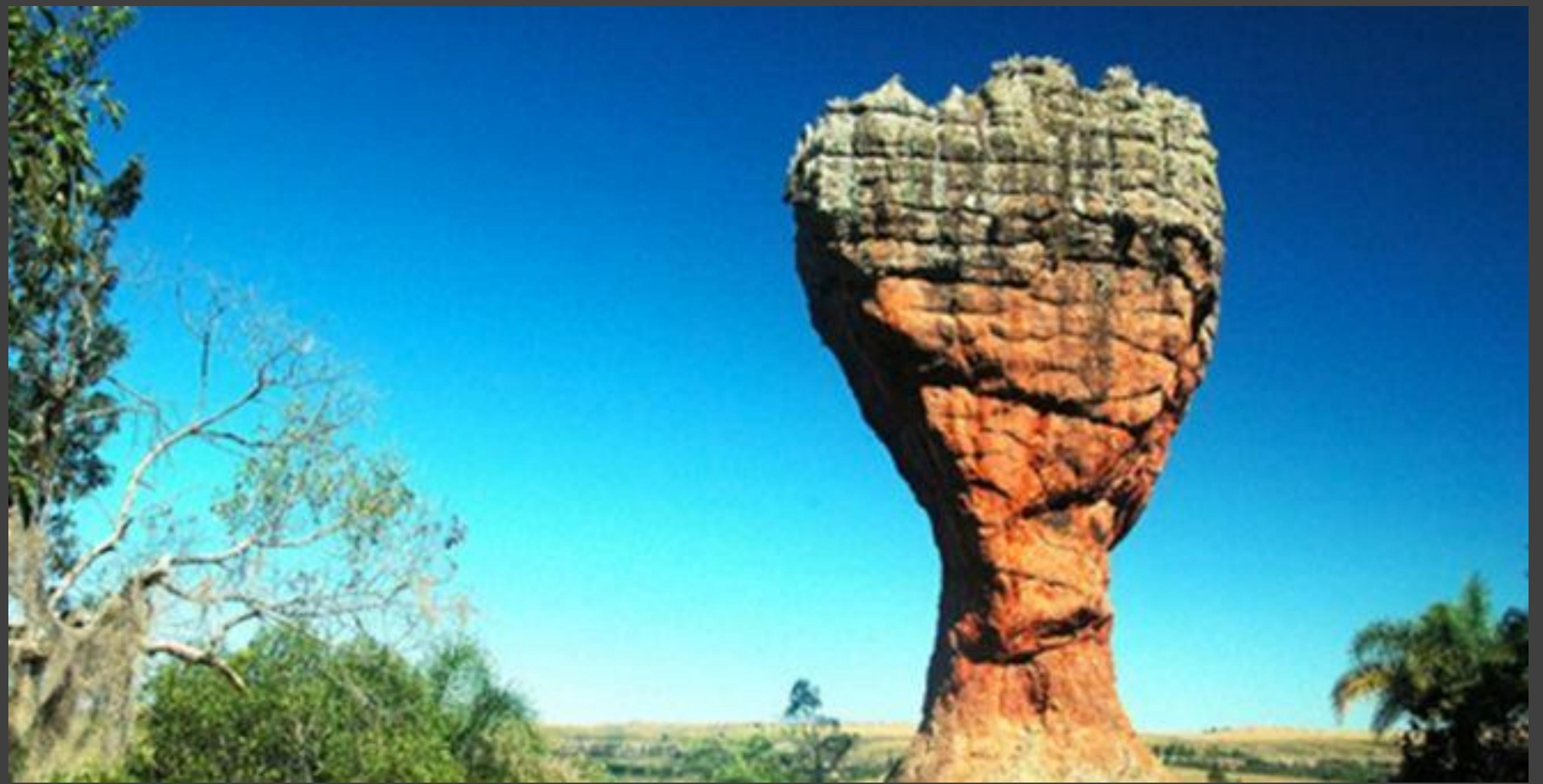


















What do I teach?

Games/Mobile Development

- I teach on the Graduation Program and Postgraduation Program
- Subject:
 - Programming
 - Mobile Development
 - Computer Graphics
 - Game Development
 - Server/Cloud Development
 - Databases
 - Blockchain development



AirPlay

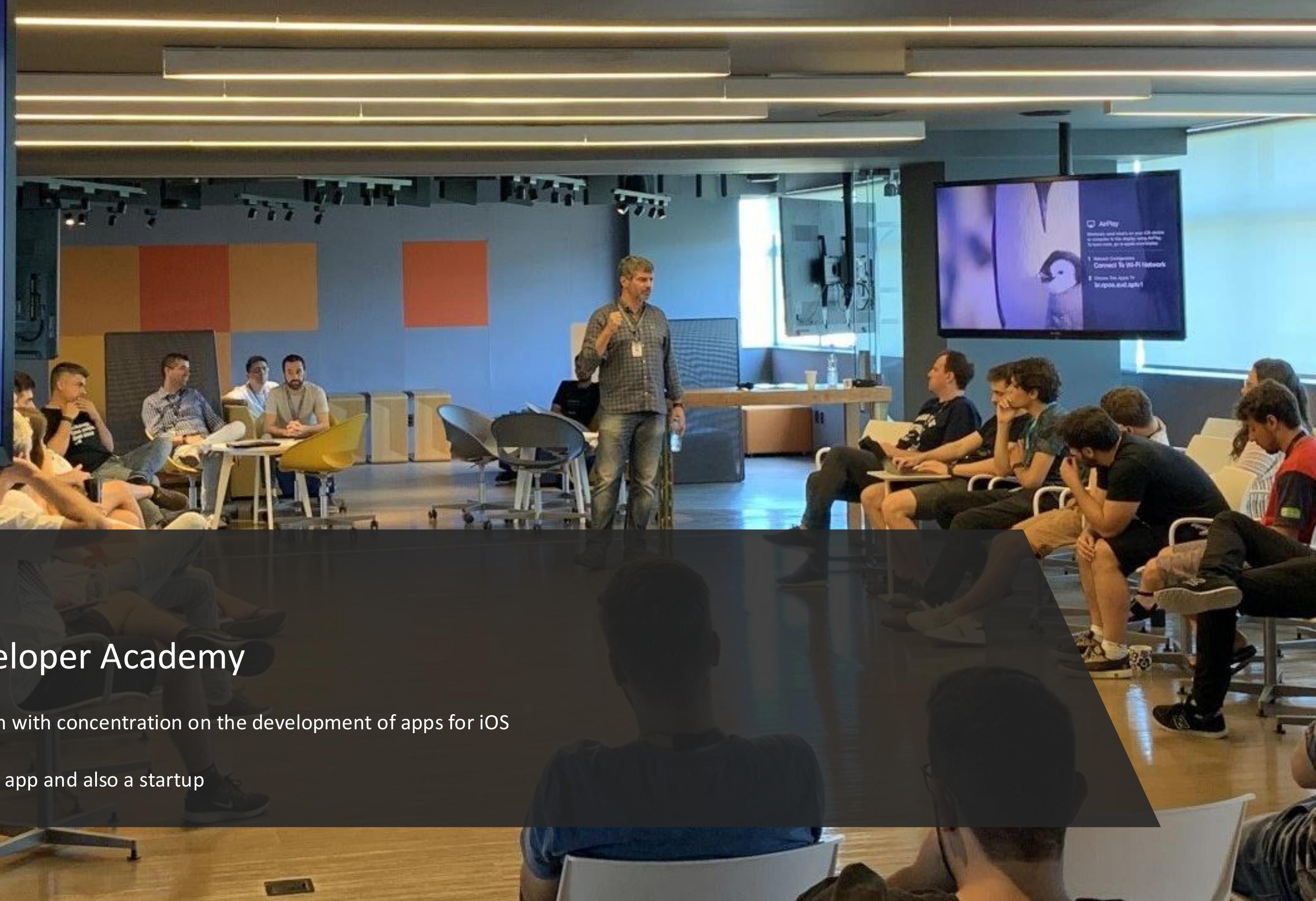
Wirelessly send what's on your iOS device or computer to this display using AirPlay. To learn more, go to apple.com/airplay.

1 Network Configuration

Connect To Wi-Fi Network

2 Choose This Apple TV

br.epoa.aud.aptv1



Apple Developer Academy

- A 2 year program with concentration on the development of apps for iOS
- They develop an app and also a startup

James Delivery

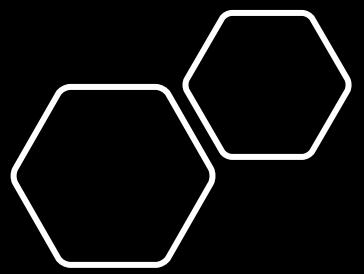
startup that deliveries
convenience to your home





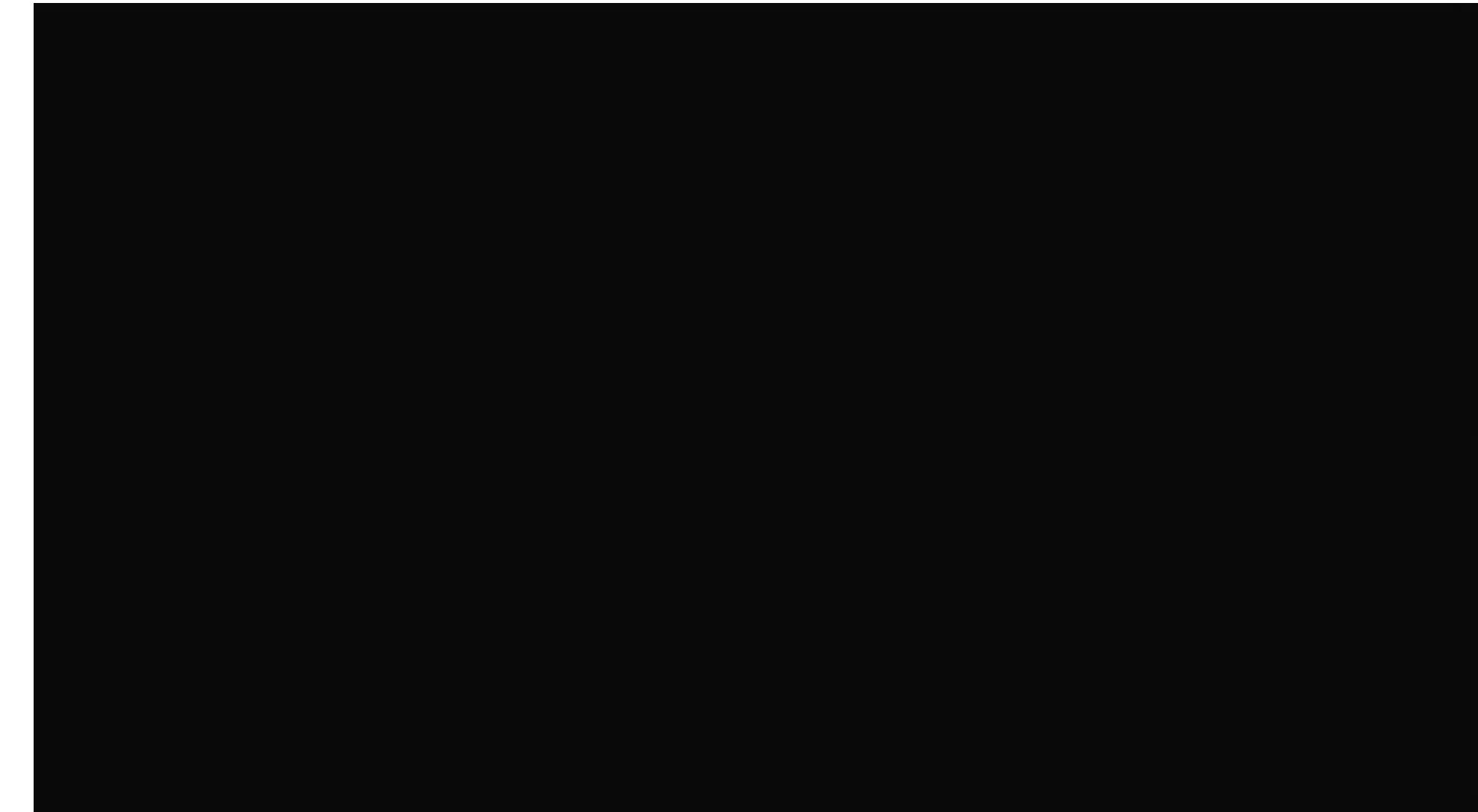
Super pads

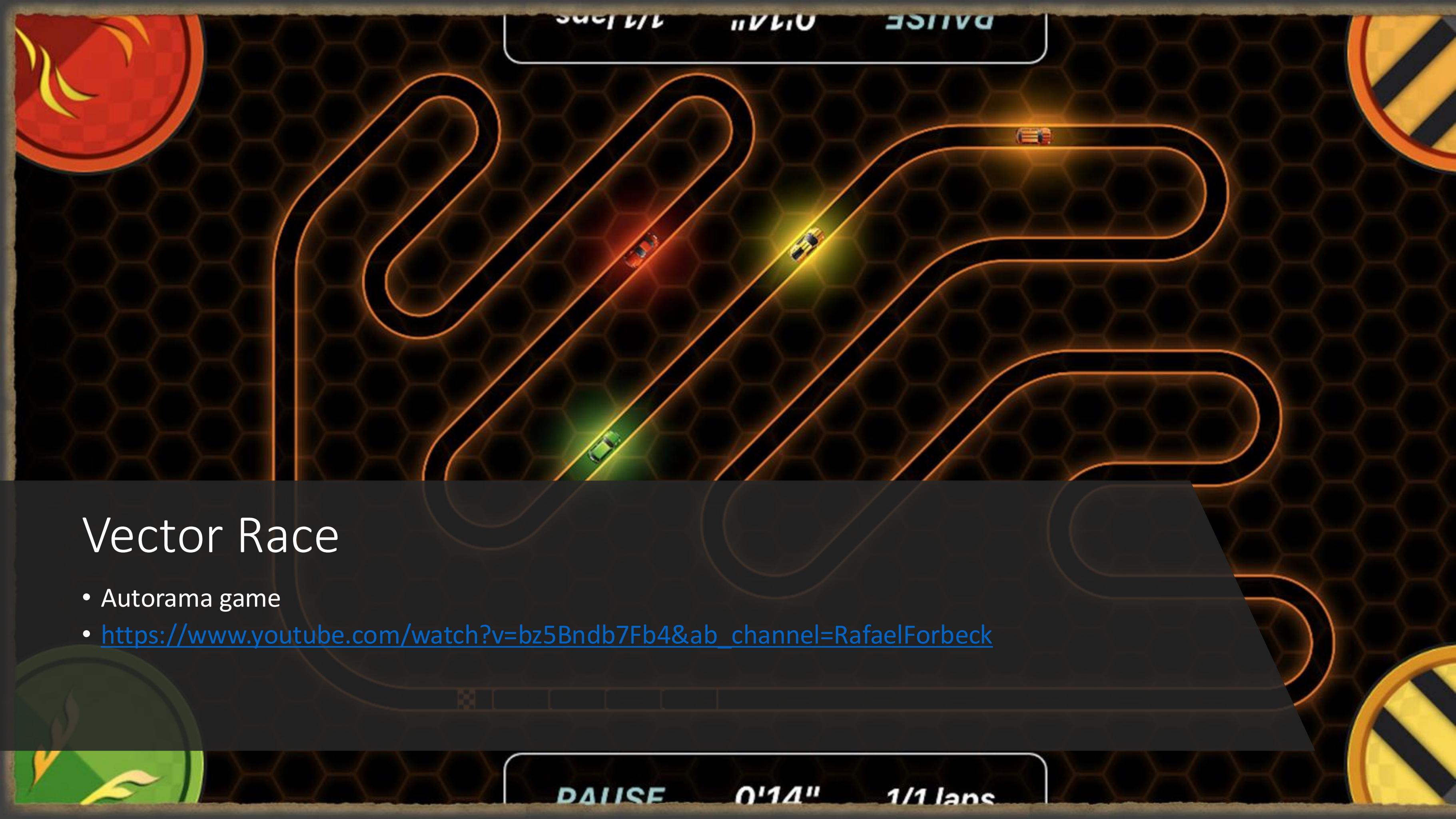
Create and Play top songs



Jump-o

Minimalist game





Vector Race

- Autorama game
- https://www.youtube.com/watch?v=bz5Bndb7Fb4&ab_channel=RafaelForbeck

Eternal

Social network that help plan a
marriage party



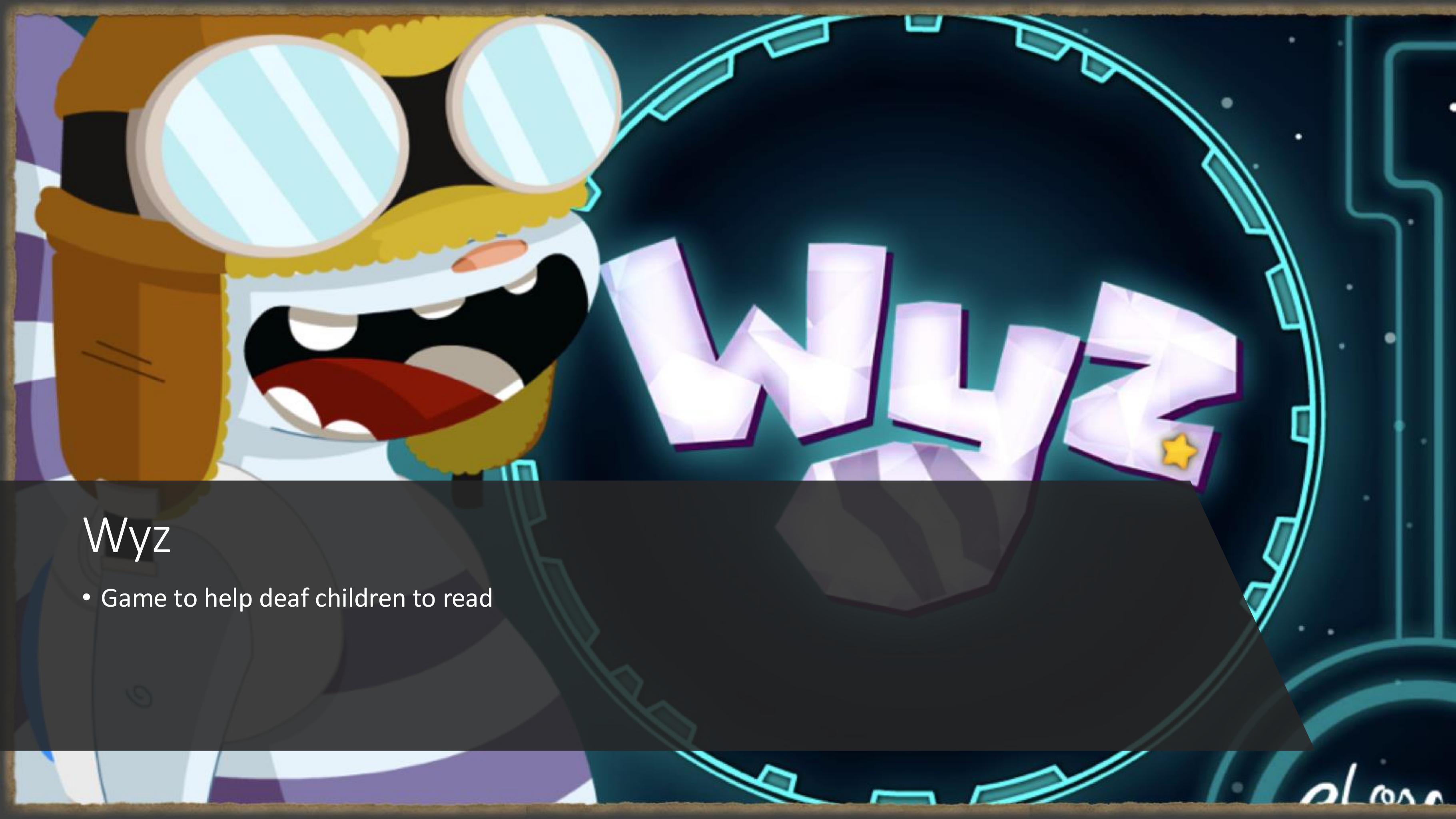
eternal

A rede social do seu Casamento



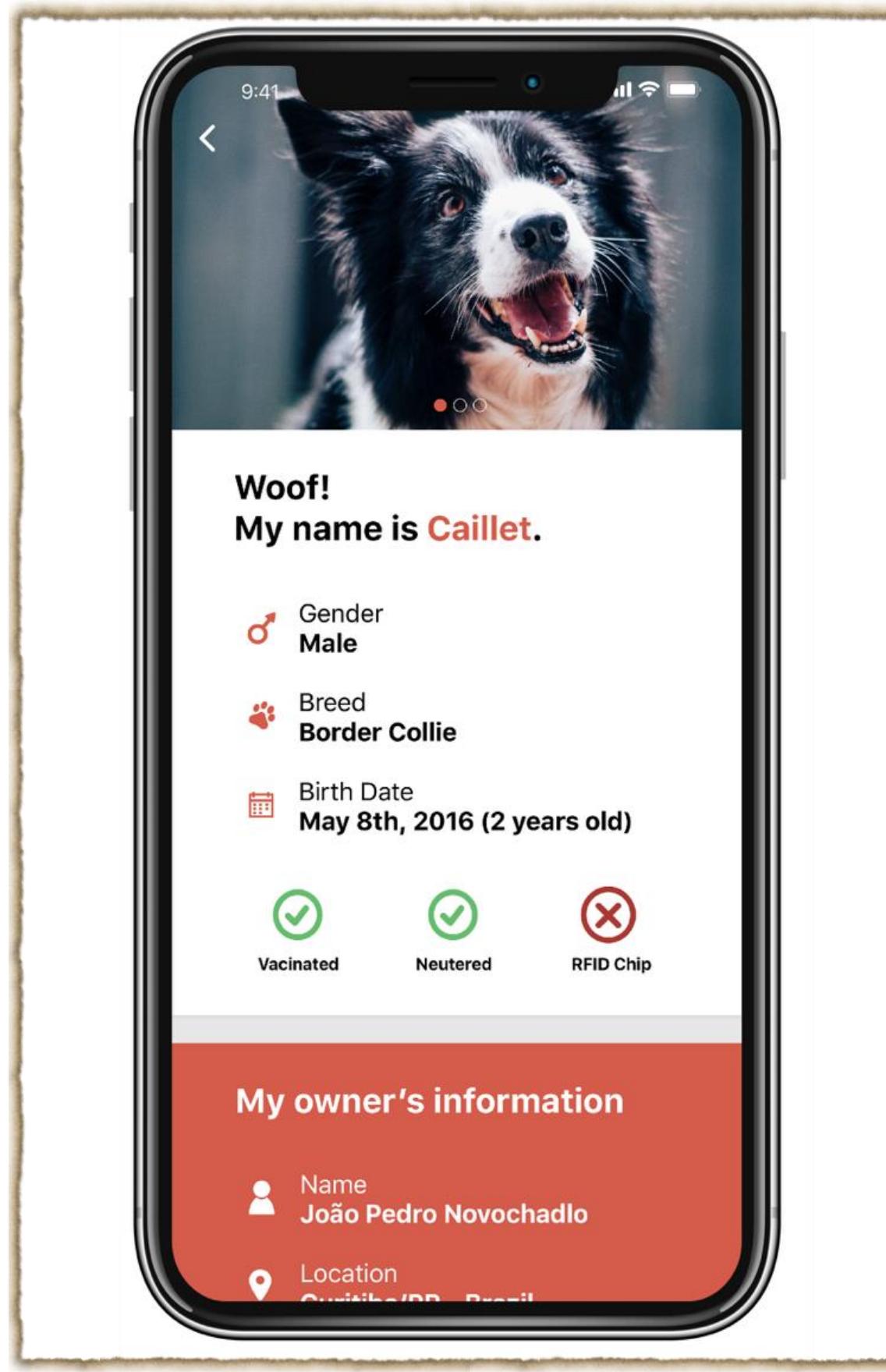
Acesse com o Facebook

Entrar



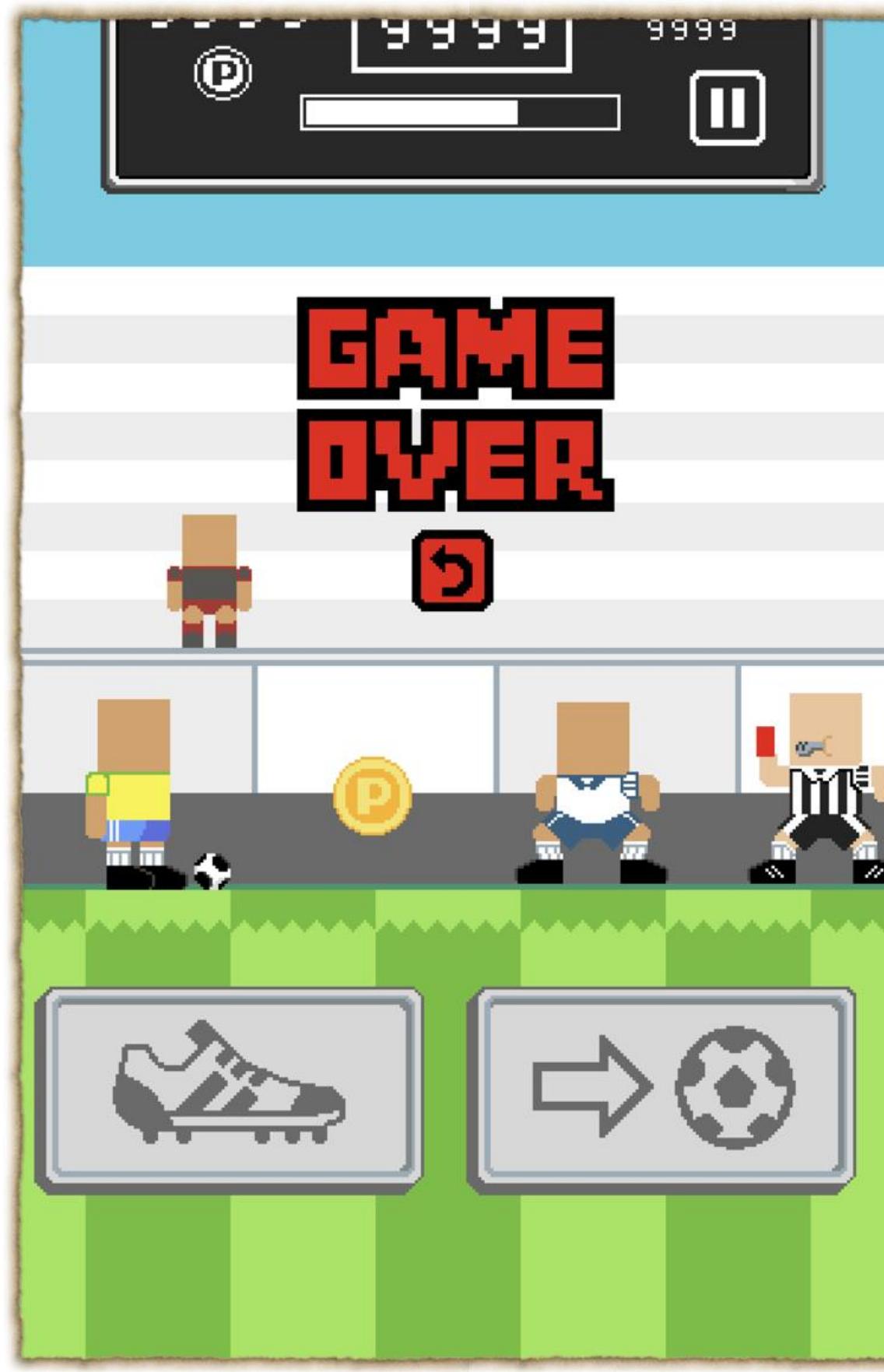
Wyz

- Game to help deaf children to read



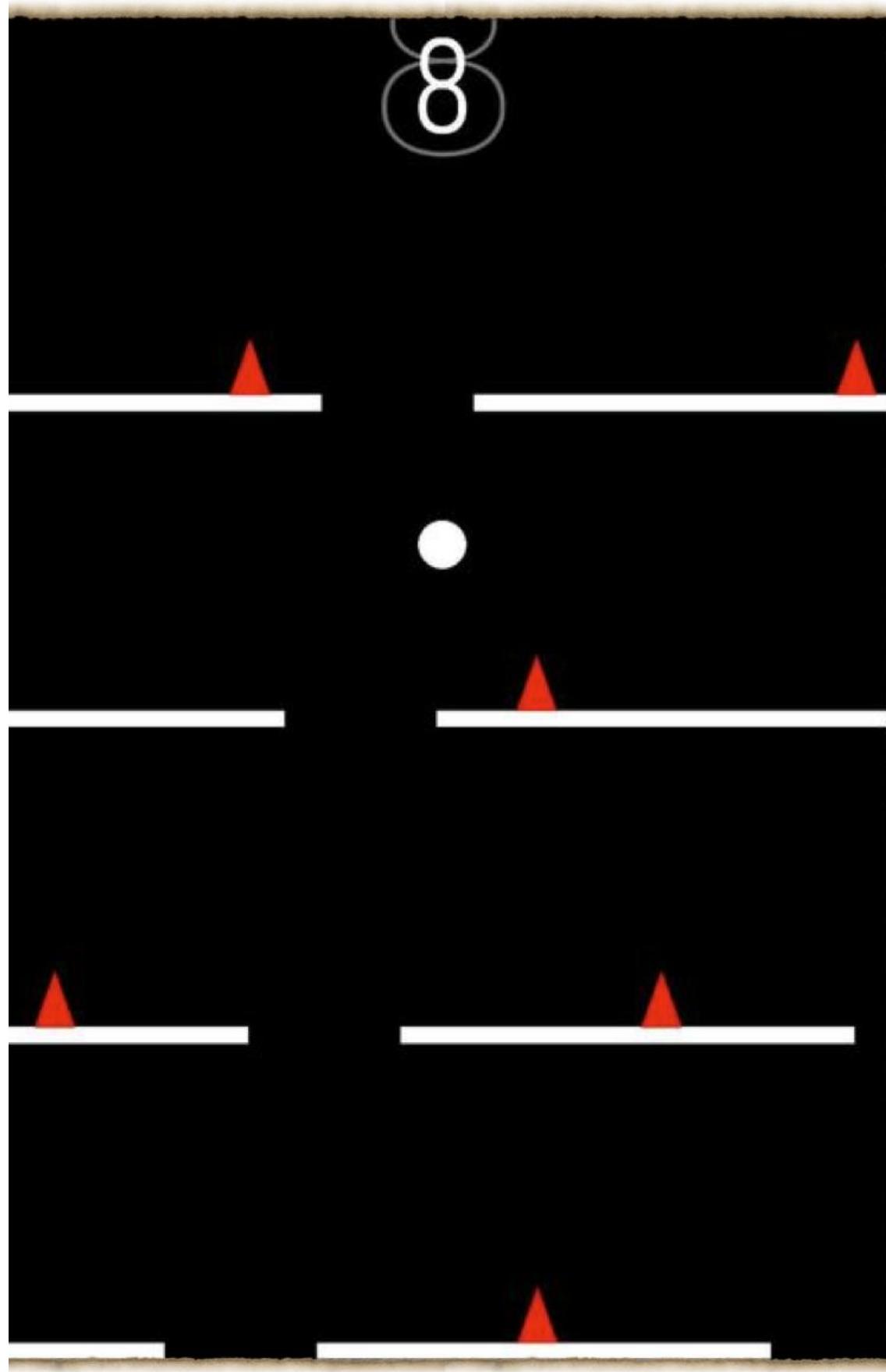
Kryptags

App with nfc technology to track pets



Soccer Runner

Hyper casual game



Linear Jump

Hyper casual game

Bottle Stack

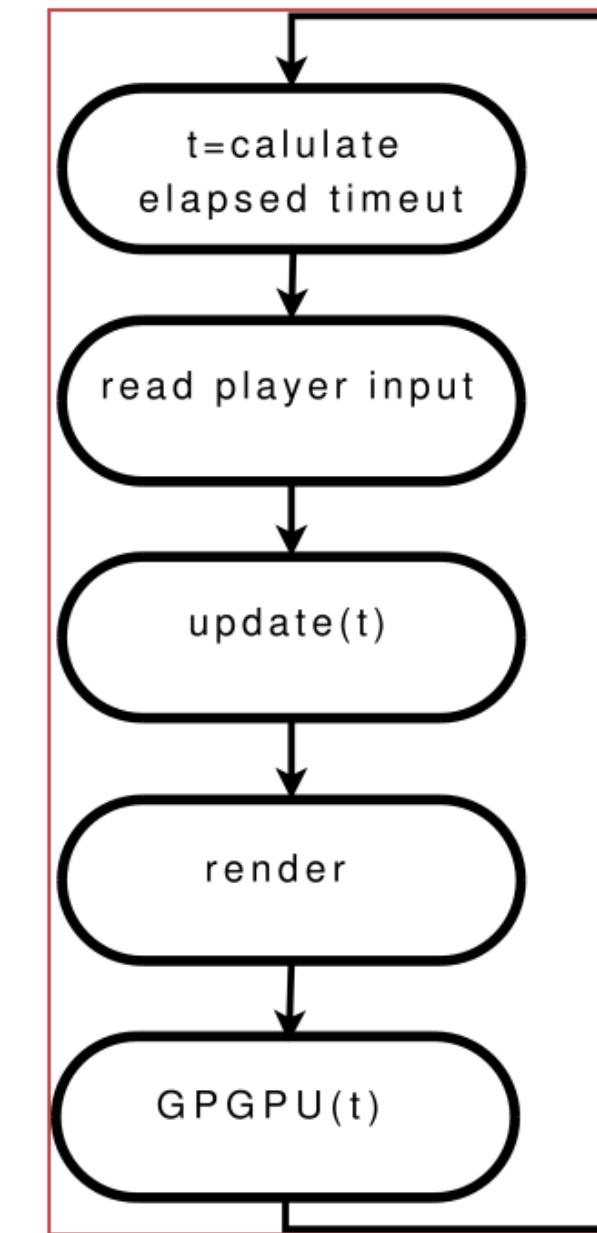
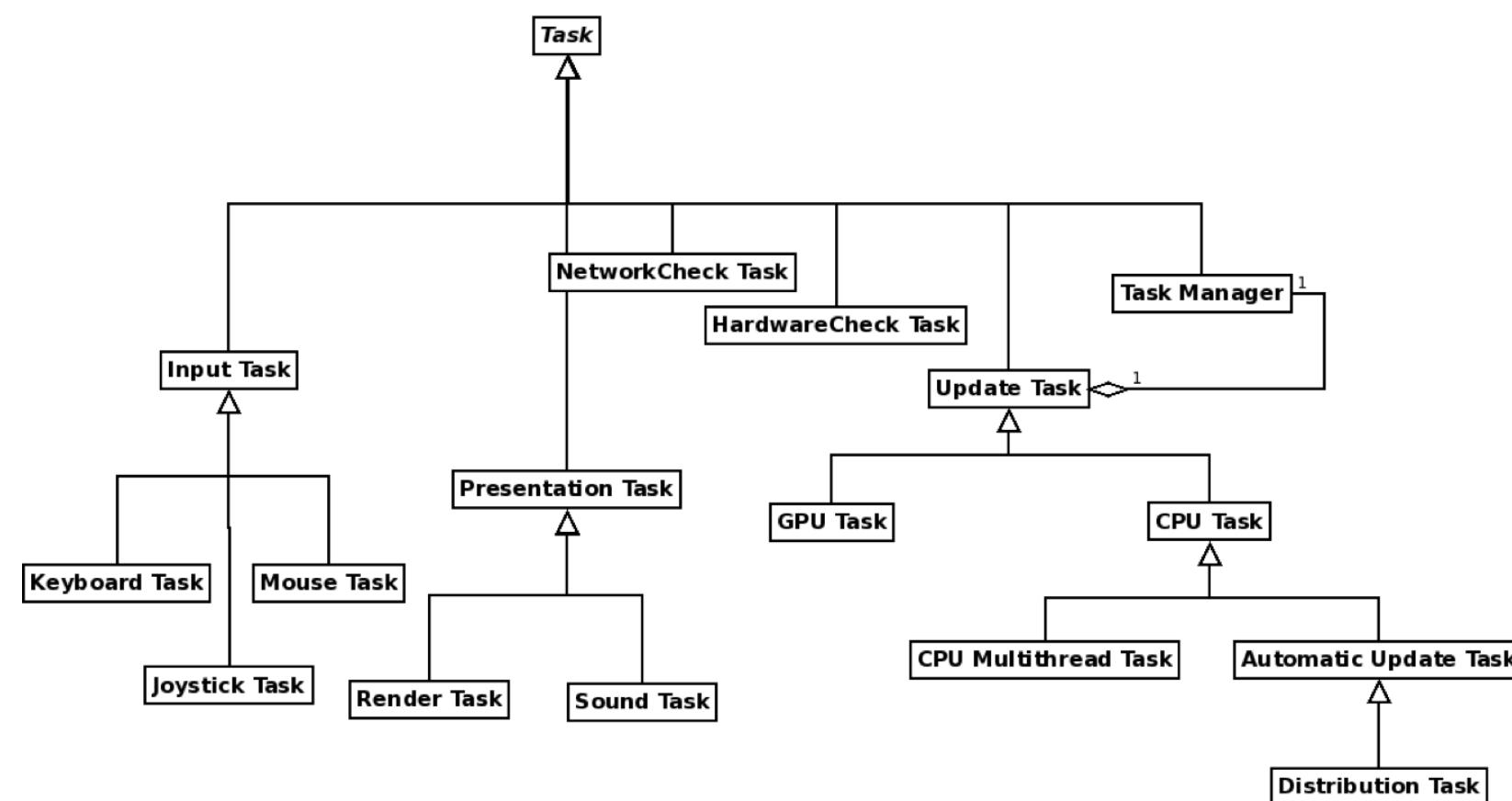


What do I
research?

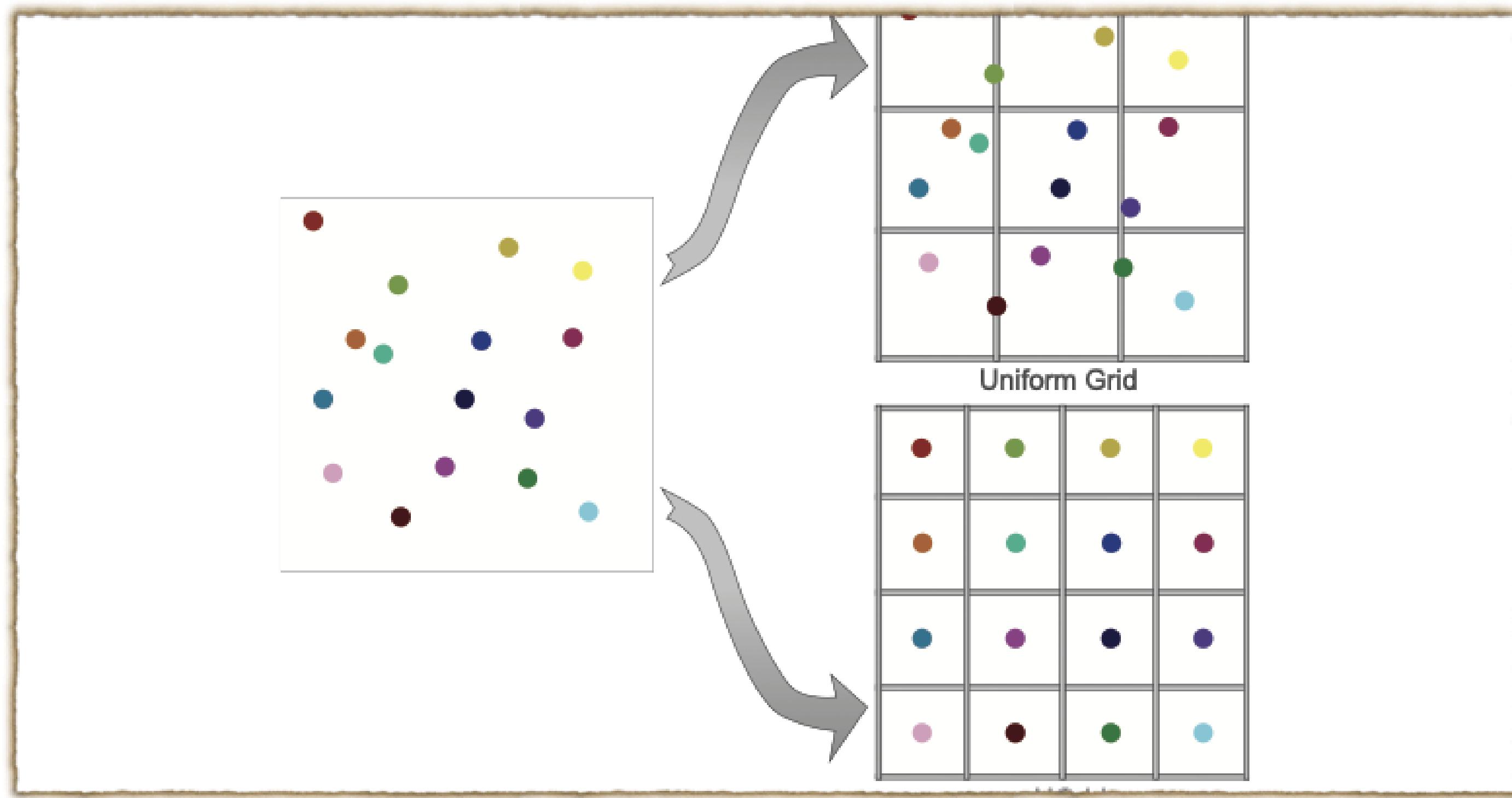


New Game Architectures

- ◆ How to develop new architectures and framework in games to get the best of the hardware.







Data Structures

Created a new Data Structure for games and interactive applications.



Points: 202

FPS: 100

GPU Wars

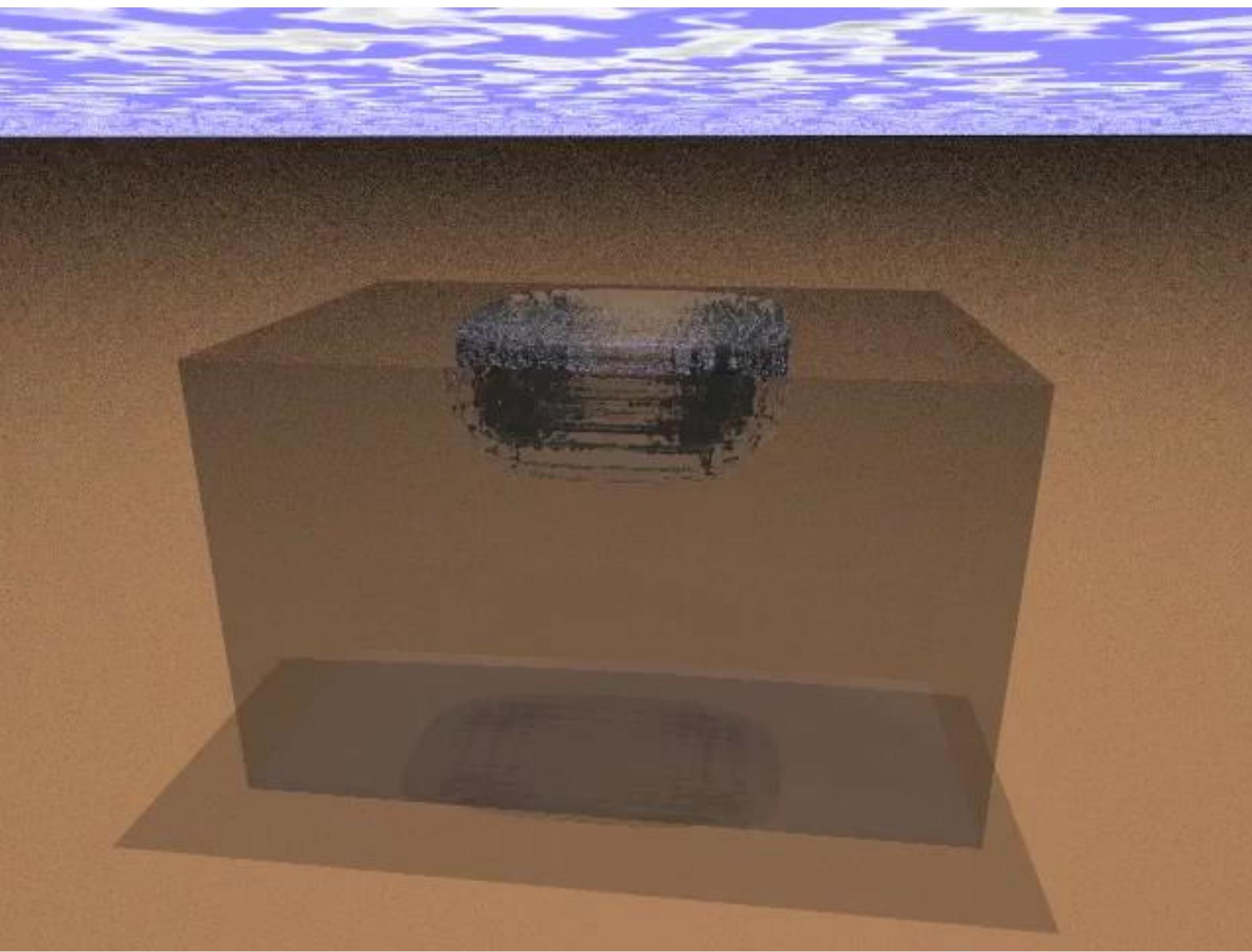




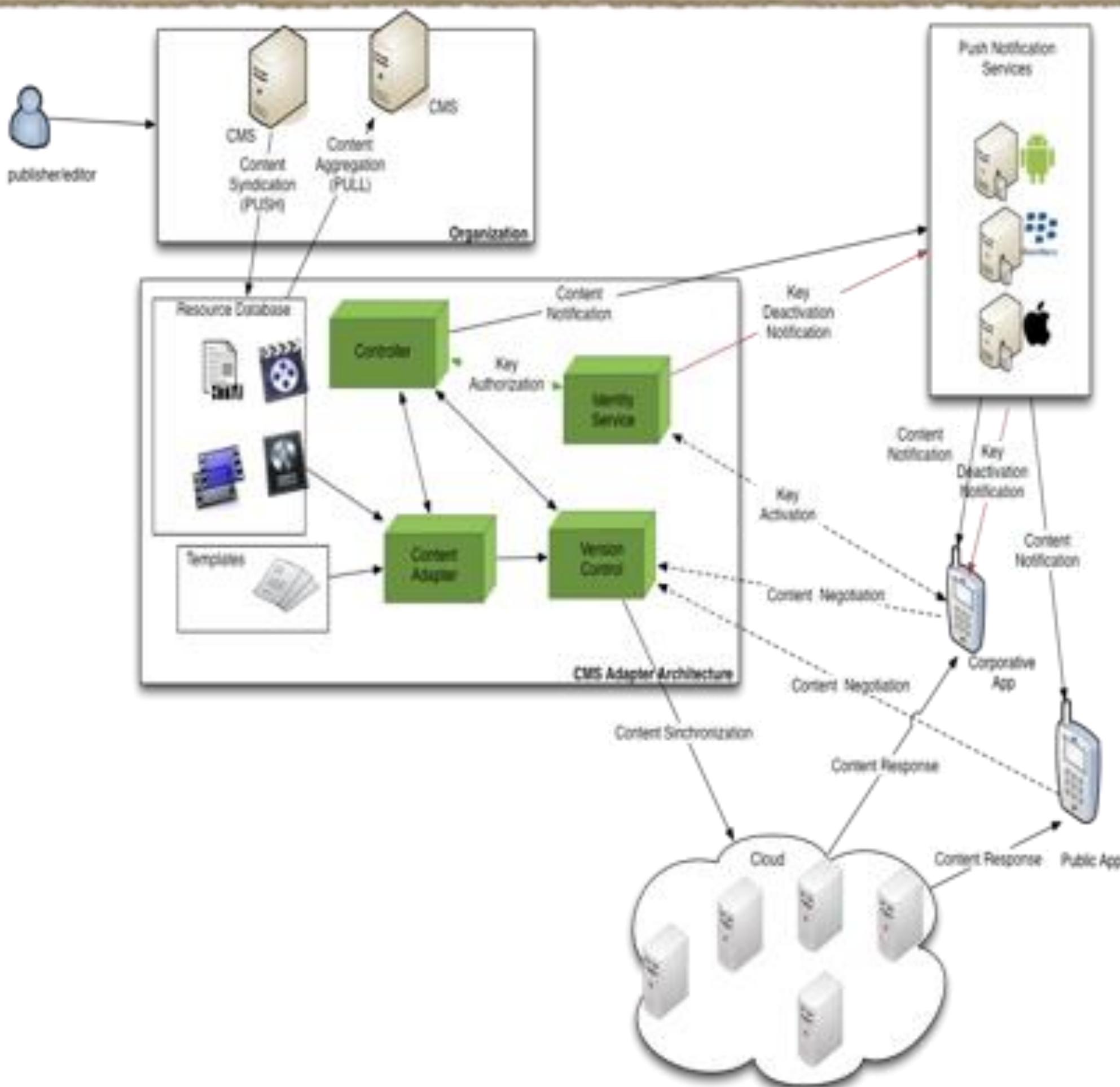
RsBalls

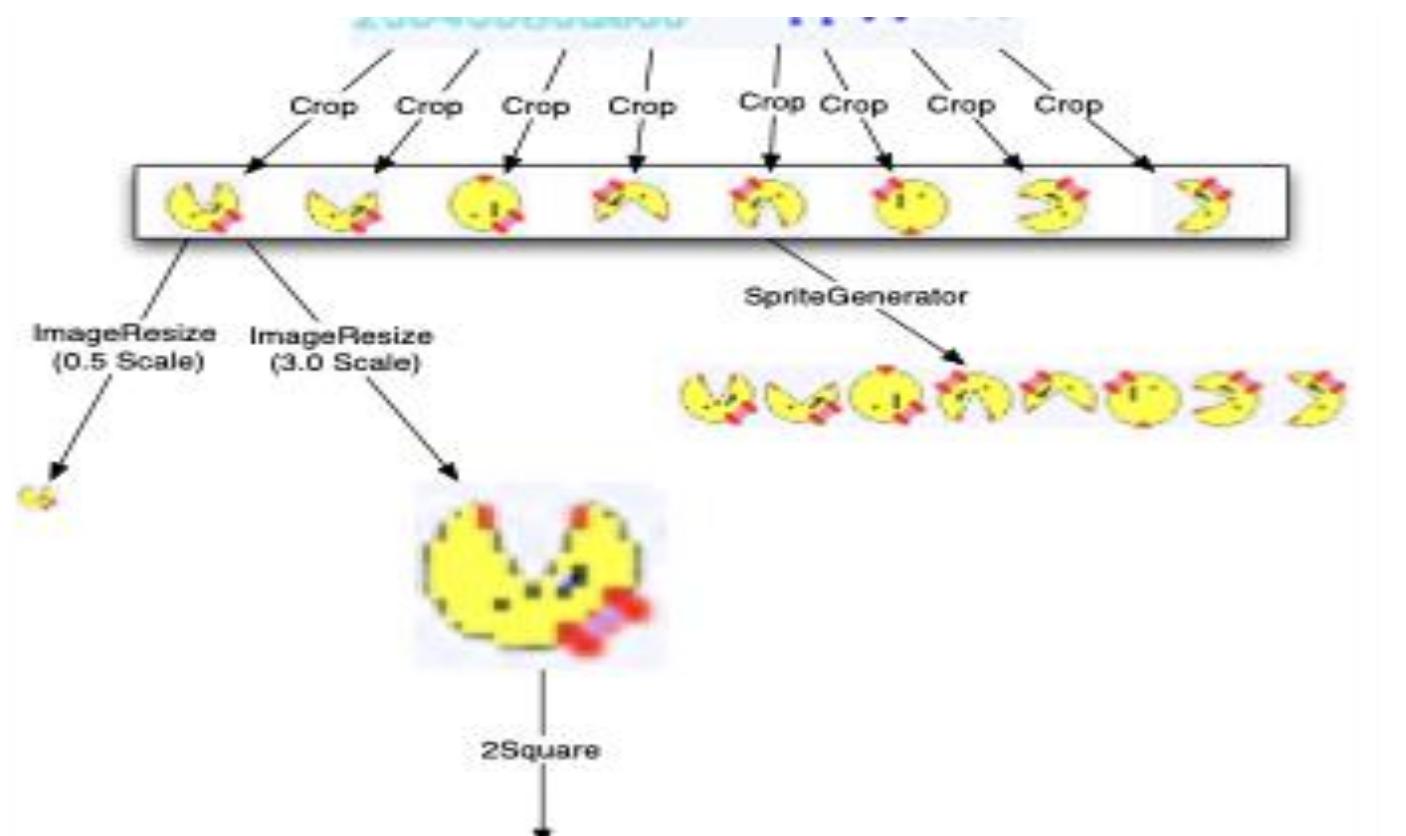
particles1024



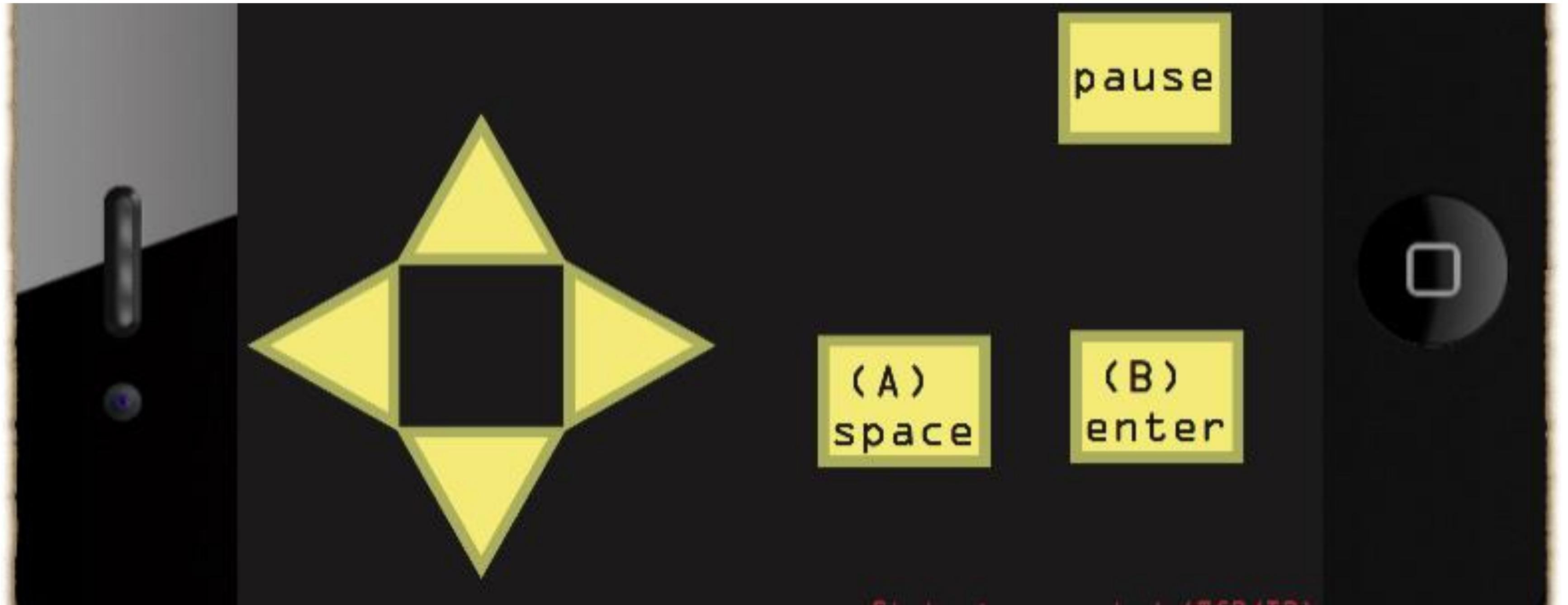


Cloud Architecture for mobile Resources









Adapt Control

Have a custom controller
that adapts to the user
input.

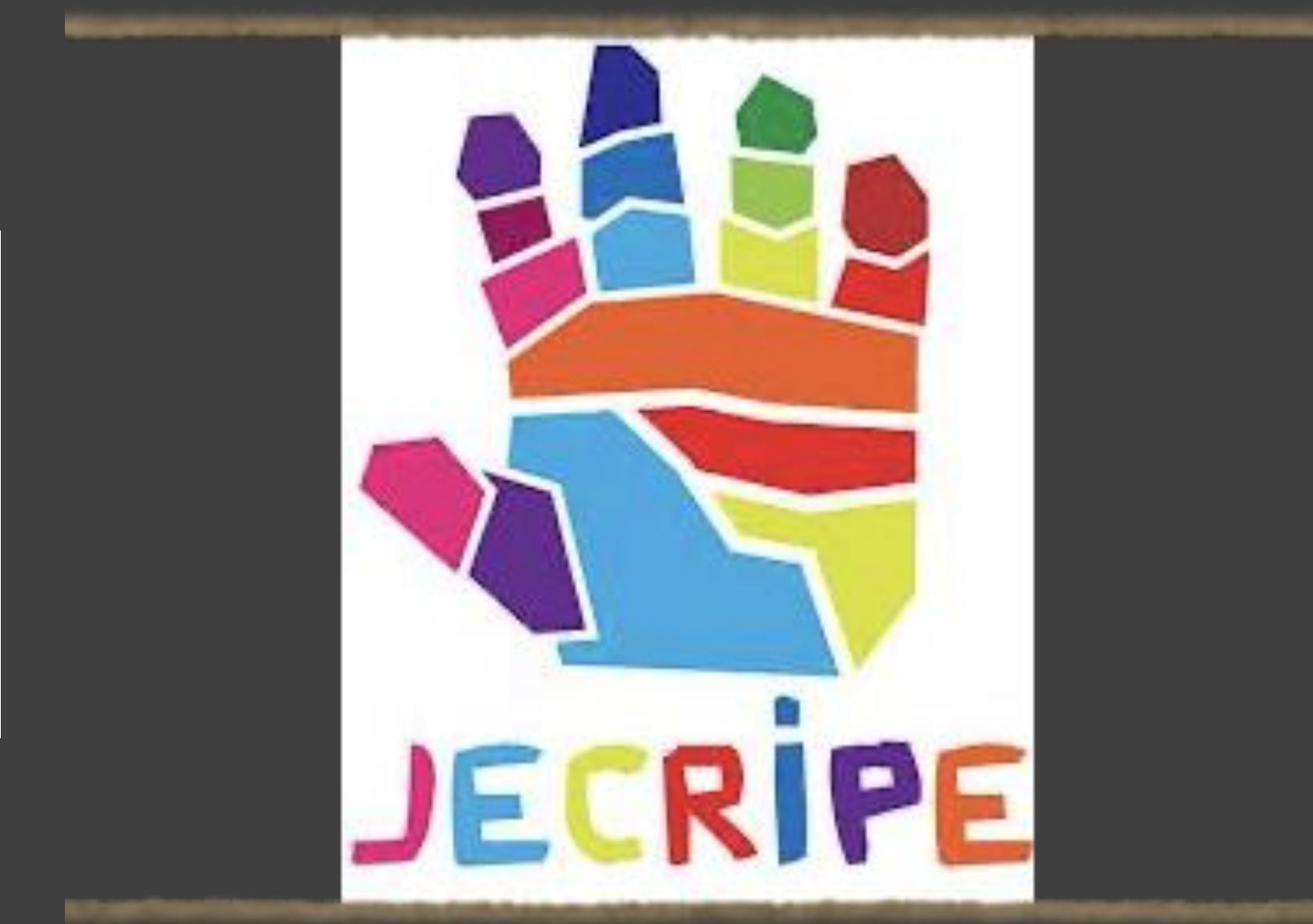


Neurofeedback games

Use the brainwaves as a input for games.

Jecripe

A game to teach children with Down syndrome.





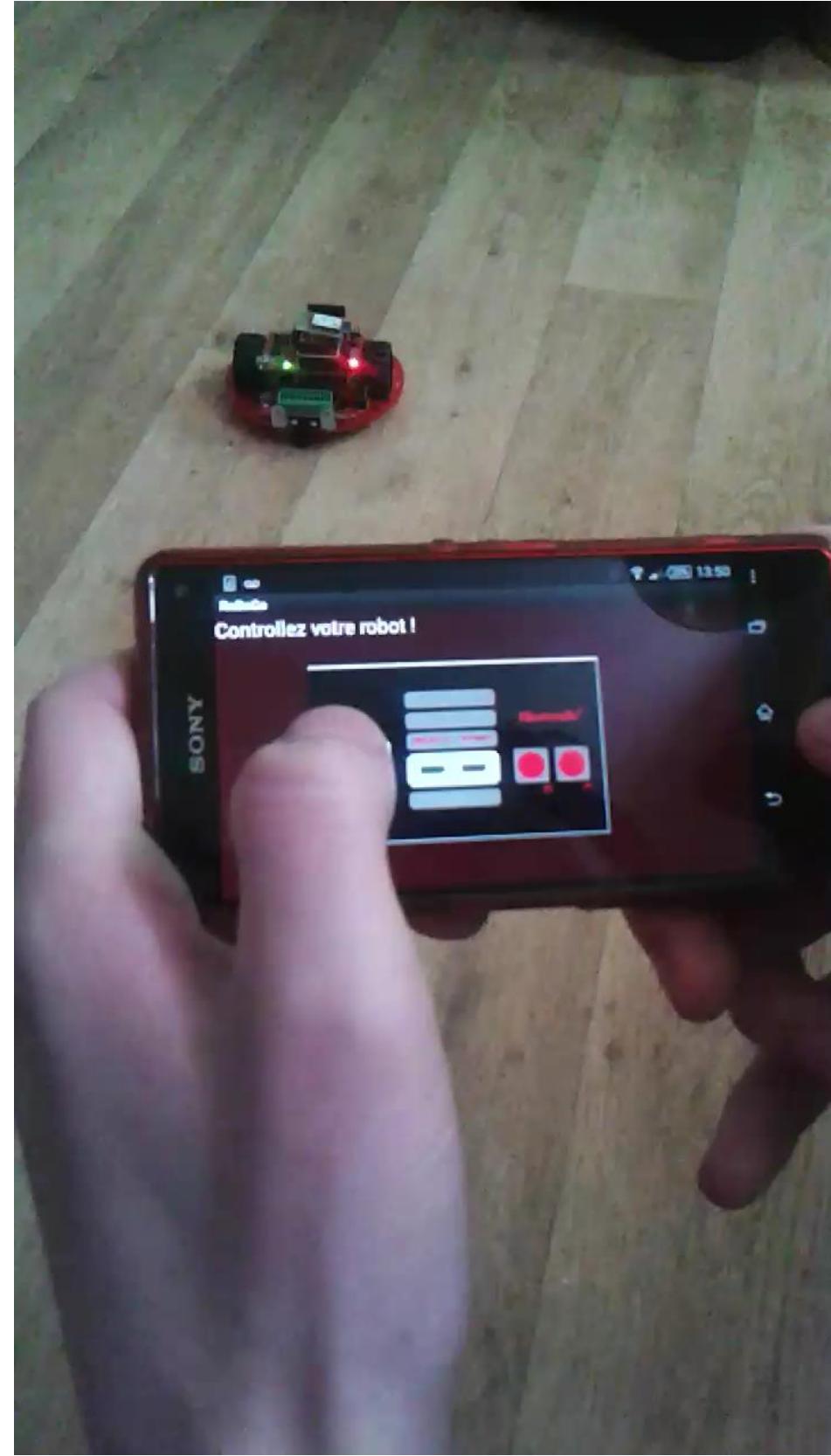
Carnaval Machine

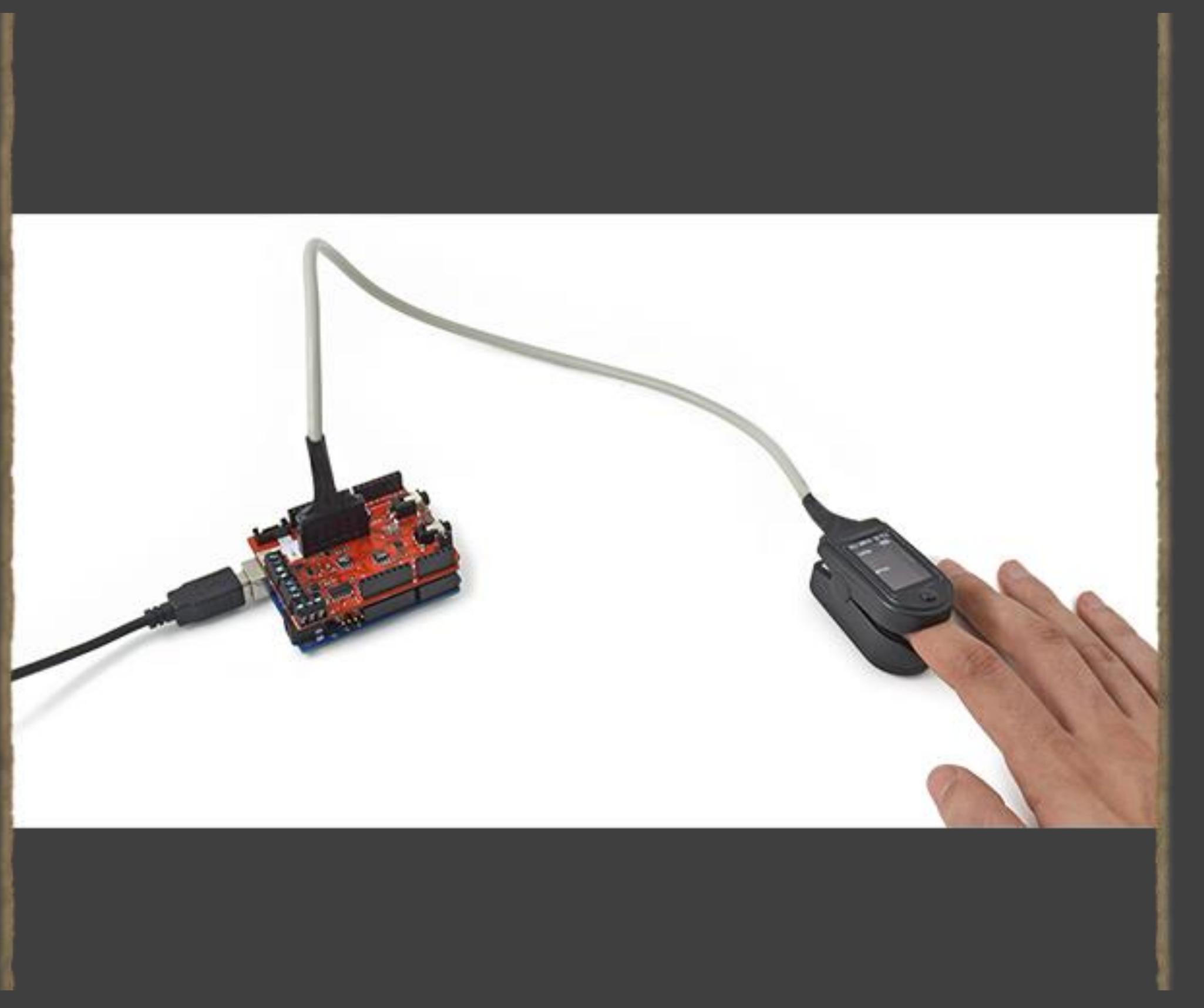
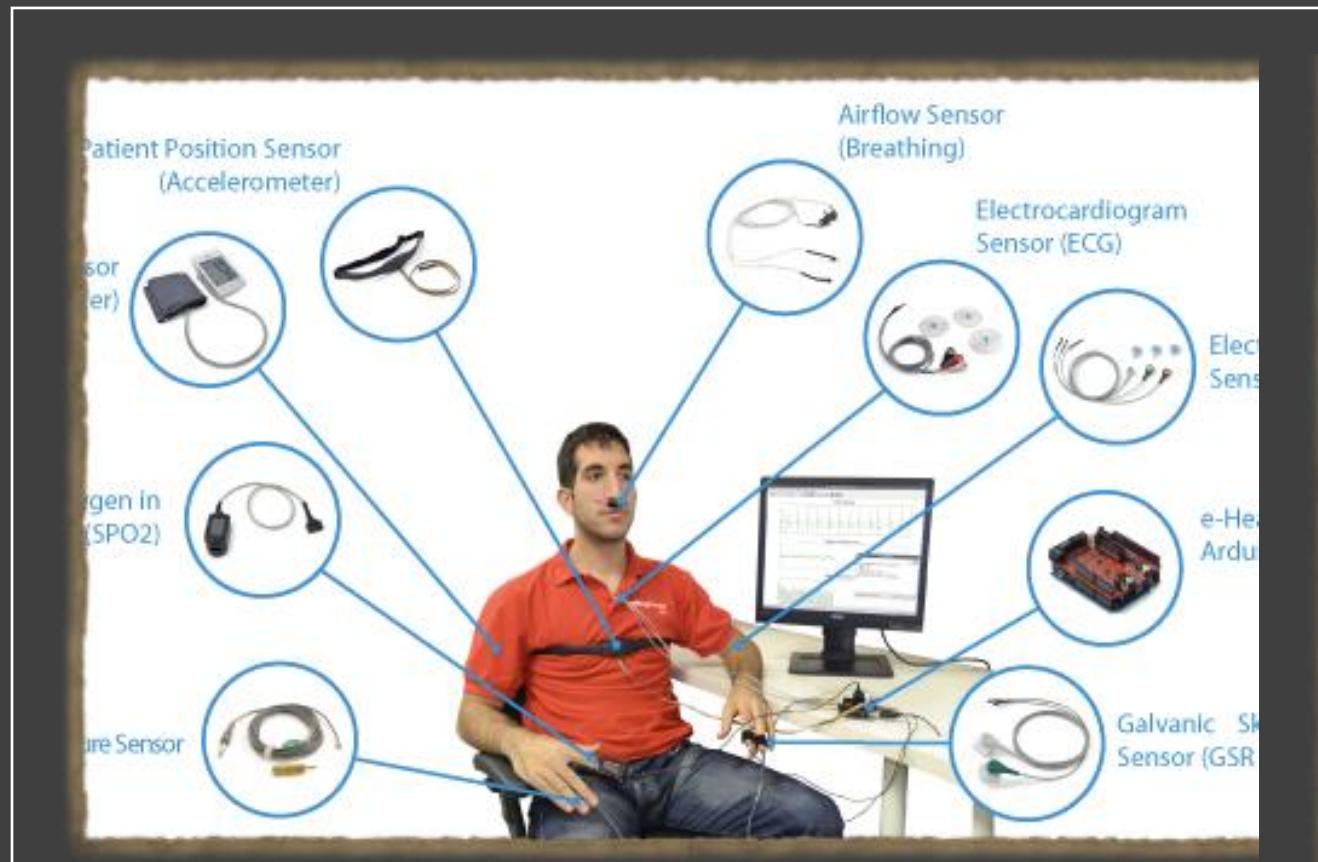
A game to present to the world the culture and world of the Carnaval.

What do I teach in
ESIGELEC?

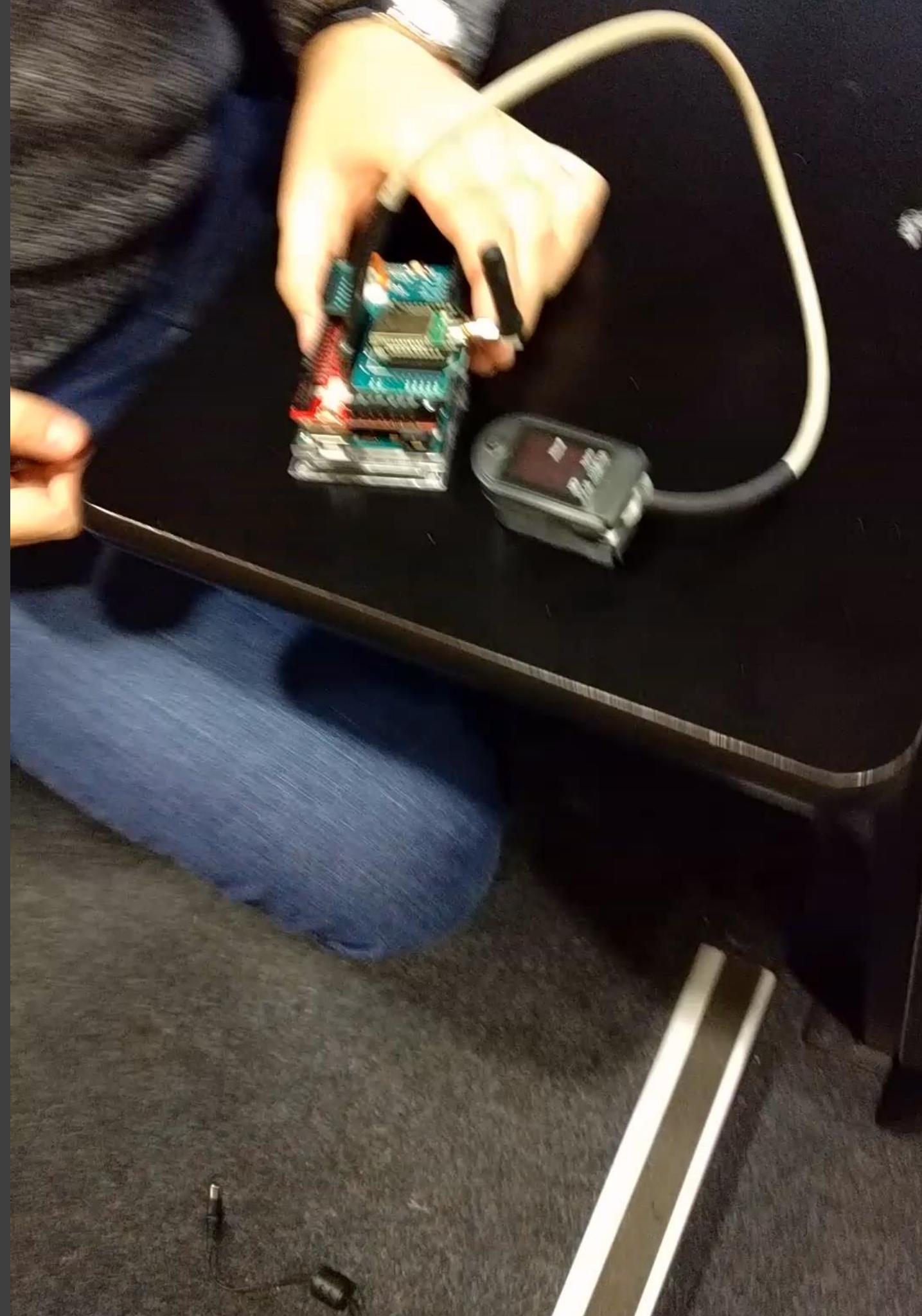
Android

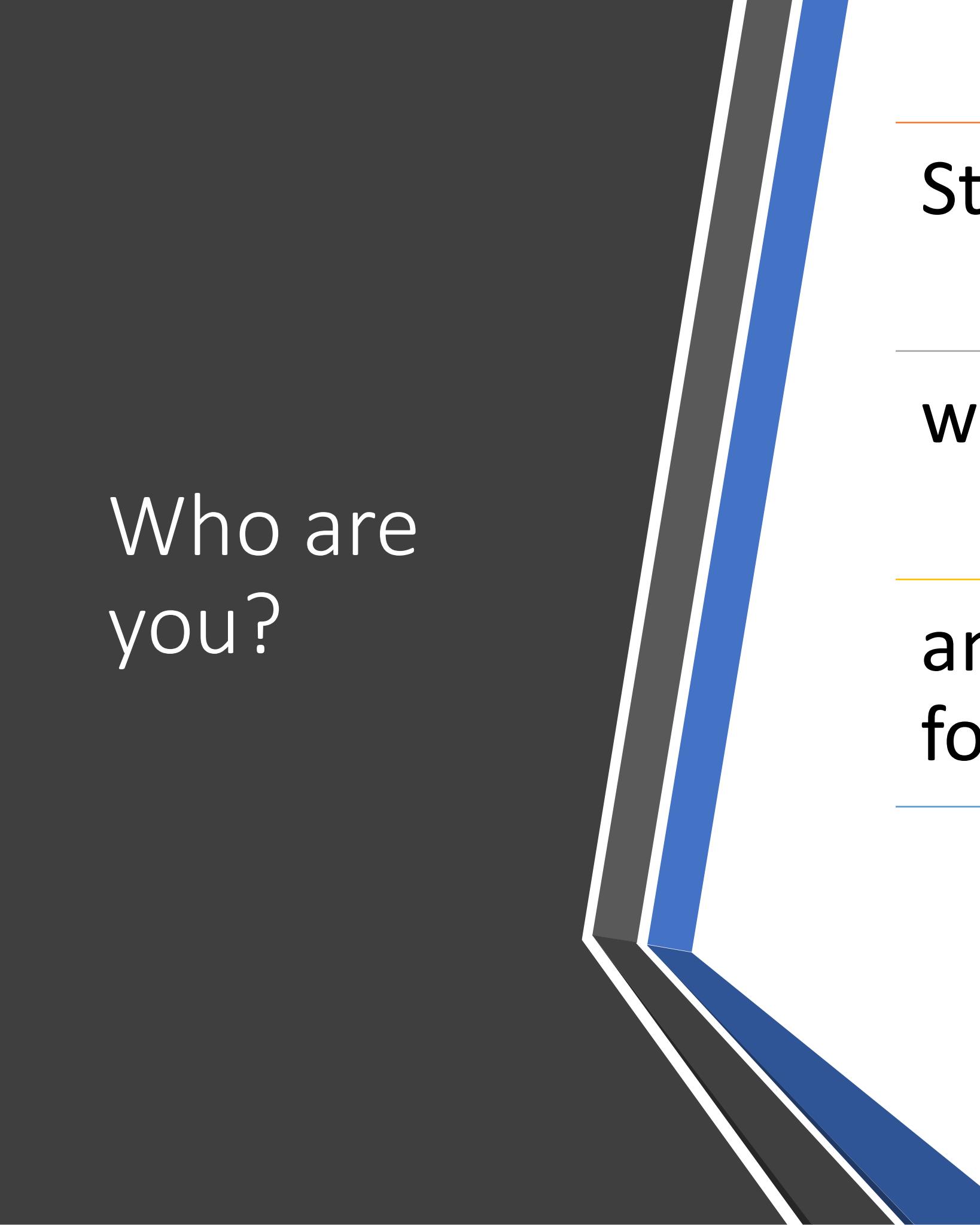
- I teach mobile development using Android;
- The goal is that after the course the student is able to develop Apps;
- I teach in three courses:
 - Android and Innovation - where the students build an App to control a robot
 - Android for Health Care and Well-being - where students build an health App using extra sensors;
 - App development for Master SEE





Project with Oximeter





Who are
you?

State your name,

where you were born

and your academic/professional
formation

Agenda (Can
be modified
during the
course)

07/02: Introduction to Android

08/02: Basic UI: Buttons, Text...

10/02: Activities and Intents

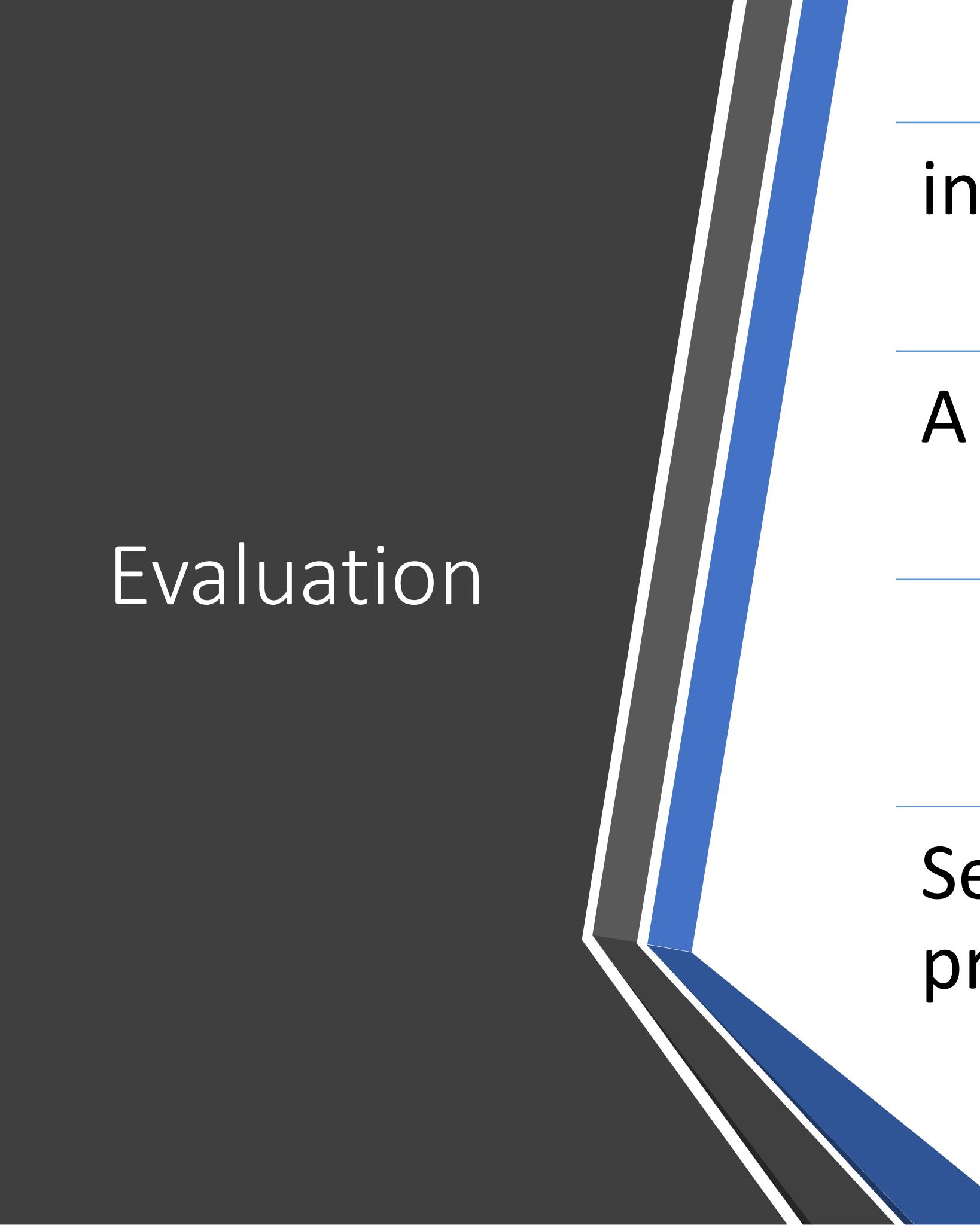
14/02: RecyclerView;

15/02: Bluetooth and Final Project

Objective of
the Course

installing Android
development
environment

creating interface and
debug it

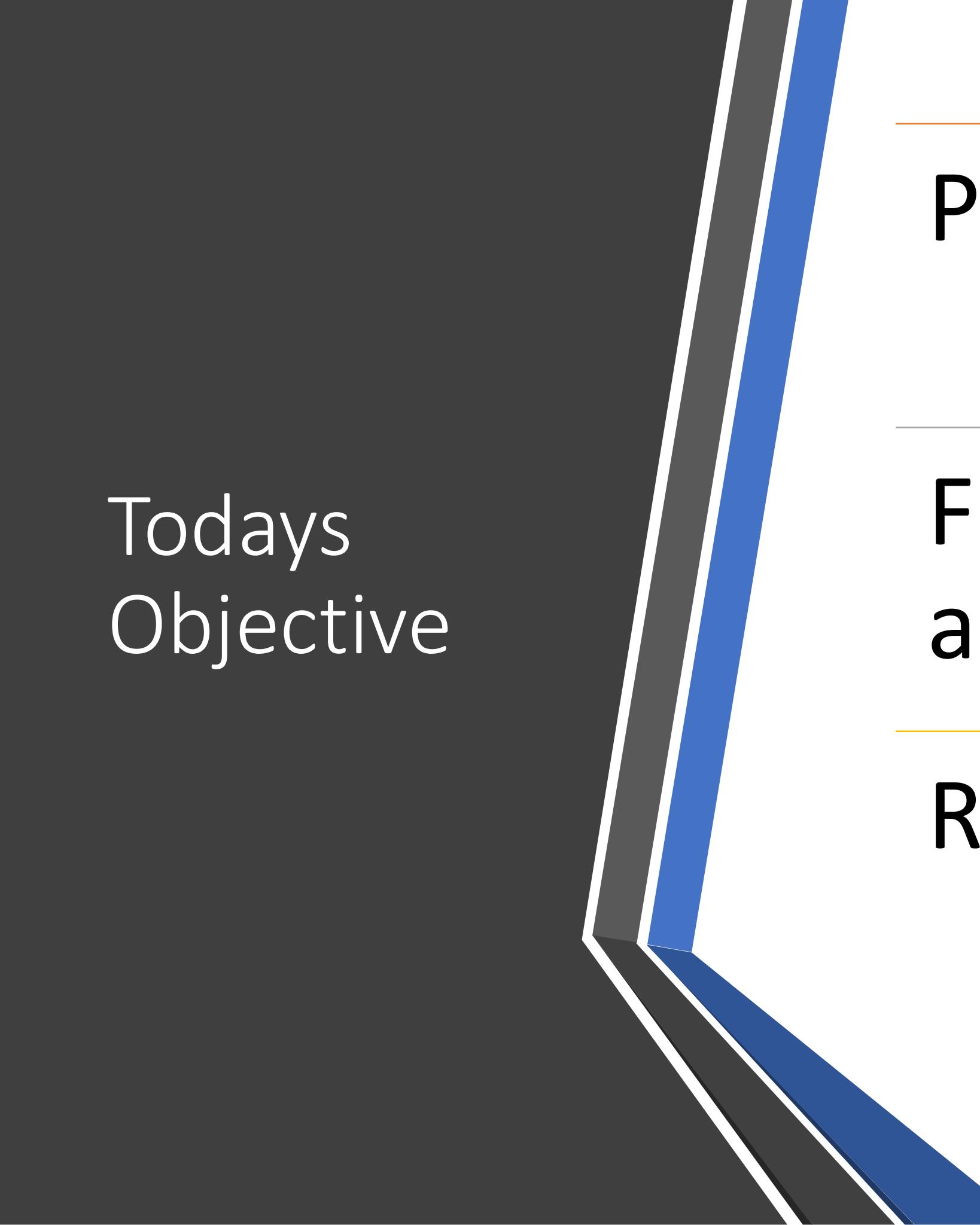


Evaluation

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

Install

Install the android studio

Install

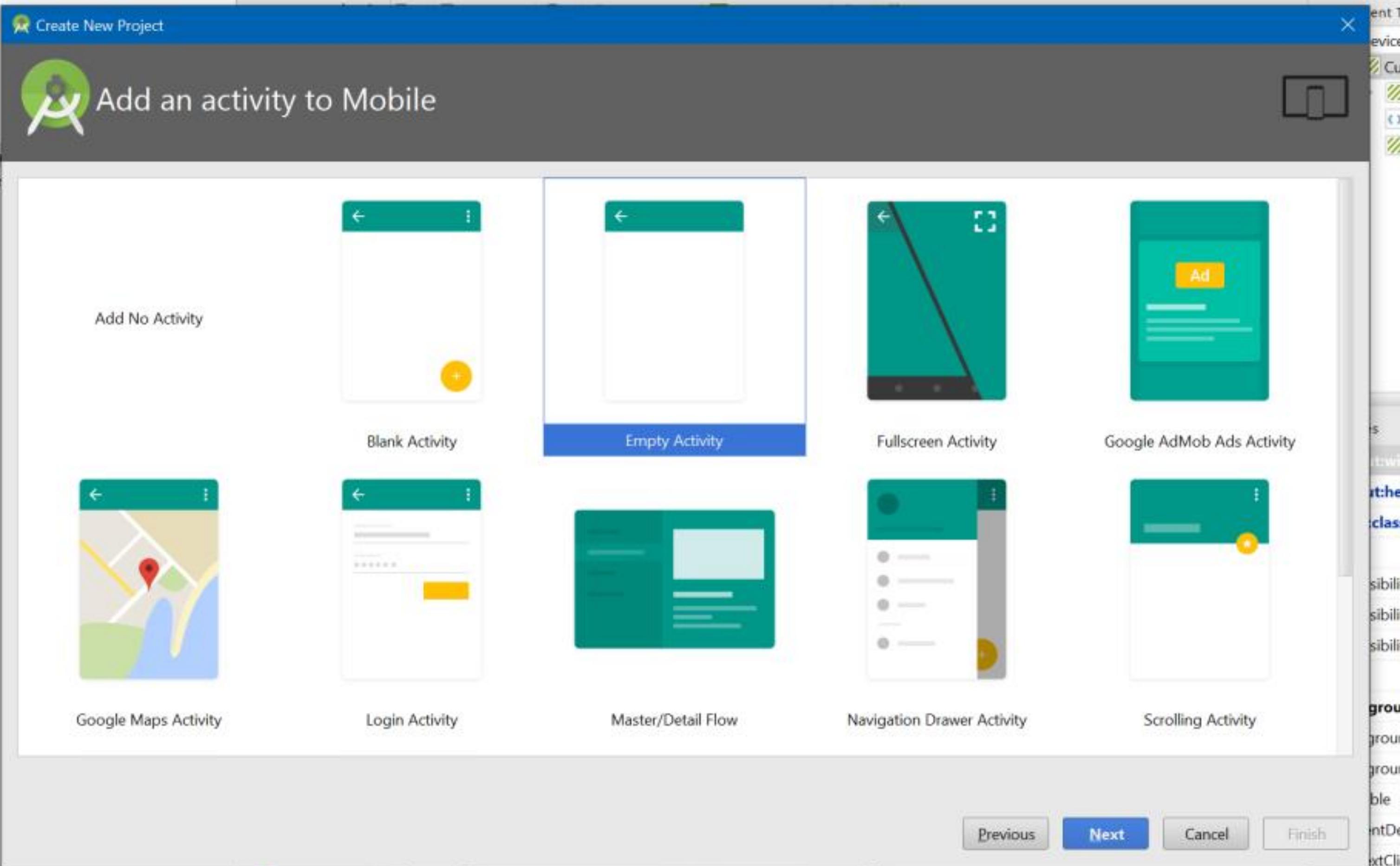
Install the SDK Packages;

Install

Install the Arm Emulator;

Install

Install USB Driver;



Manifest Tree

Device Screen

CustomView - android.support.v7.widget.RecyclerView

CustomView - android.support.v7.widget.RecyclerView

include - @layout/content_main

fab (CustomView) - android.support.v7.widget.RecyclerView

Ad

match_parent

match_parent

android.support.v7.widget.RecyclerView

sibilityLive

sibilityTra

sibilityTra

ground

groundTin

groundTin

ble

intDescrip

extClickabl

elevation

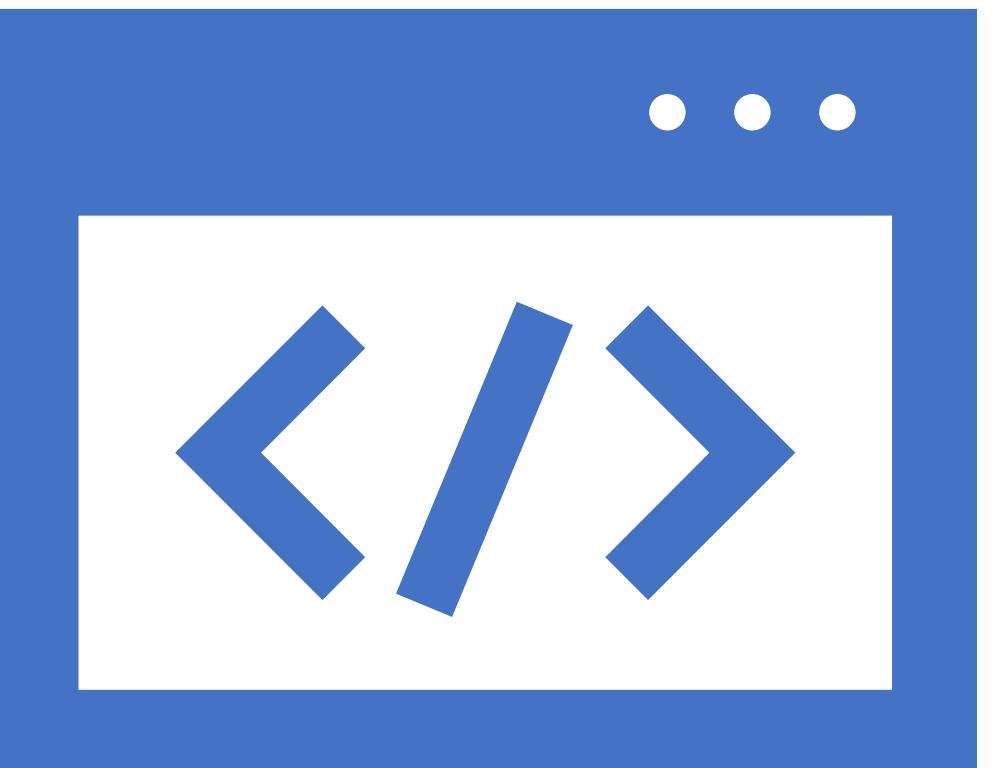
fitsSystemWind

Android Development

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

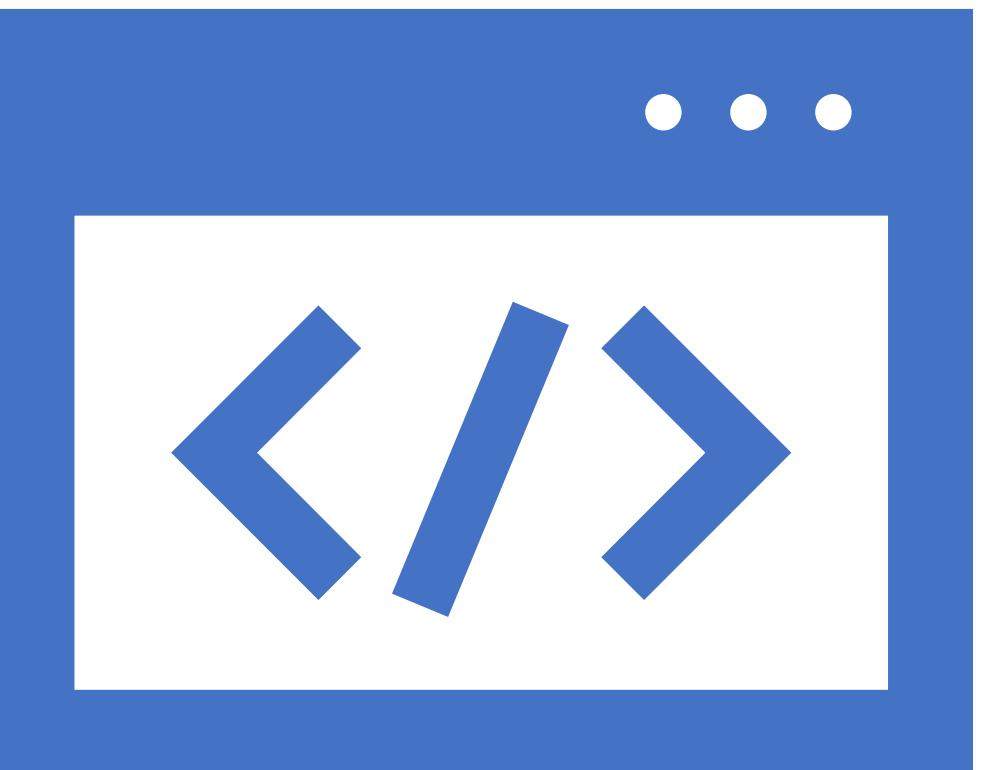
XML

- 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

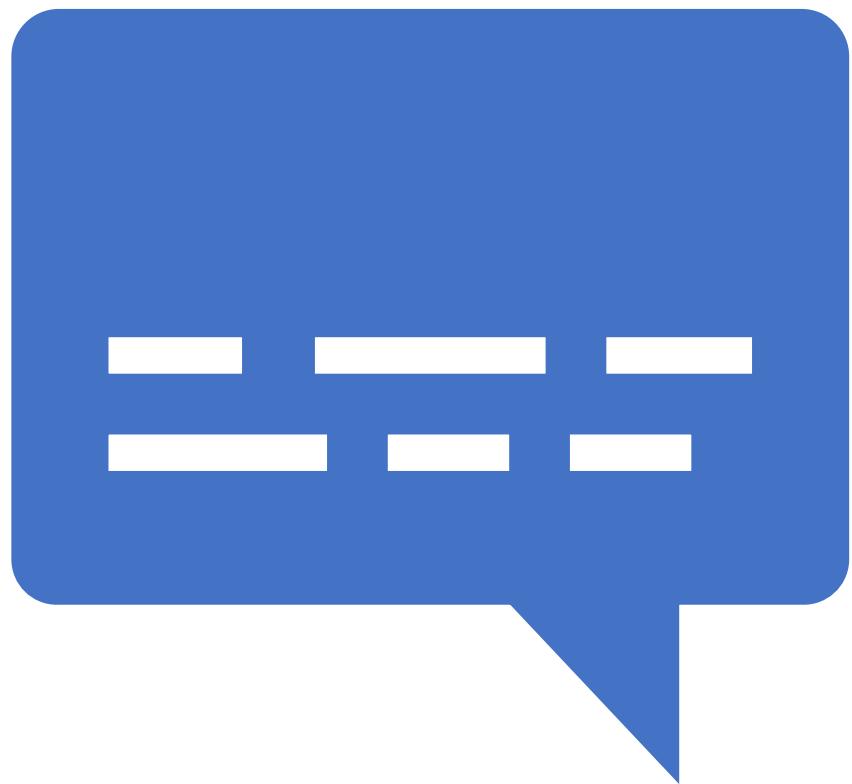
- XML is a meta markup language for text documents / textual data;
- XML allows to define languages ("applications") to represent text documents / textual data



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>
```



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

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

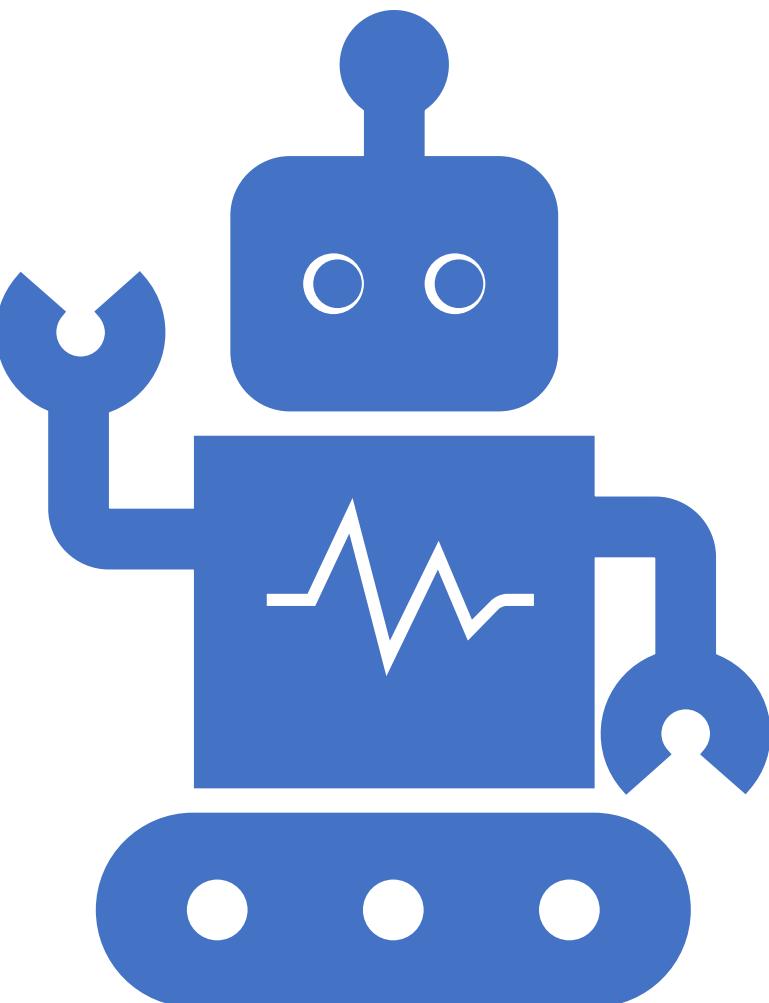
Ignores unintended touches

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.

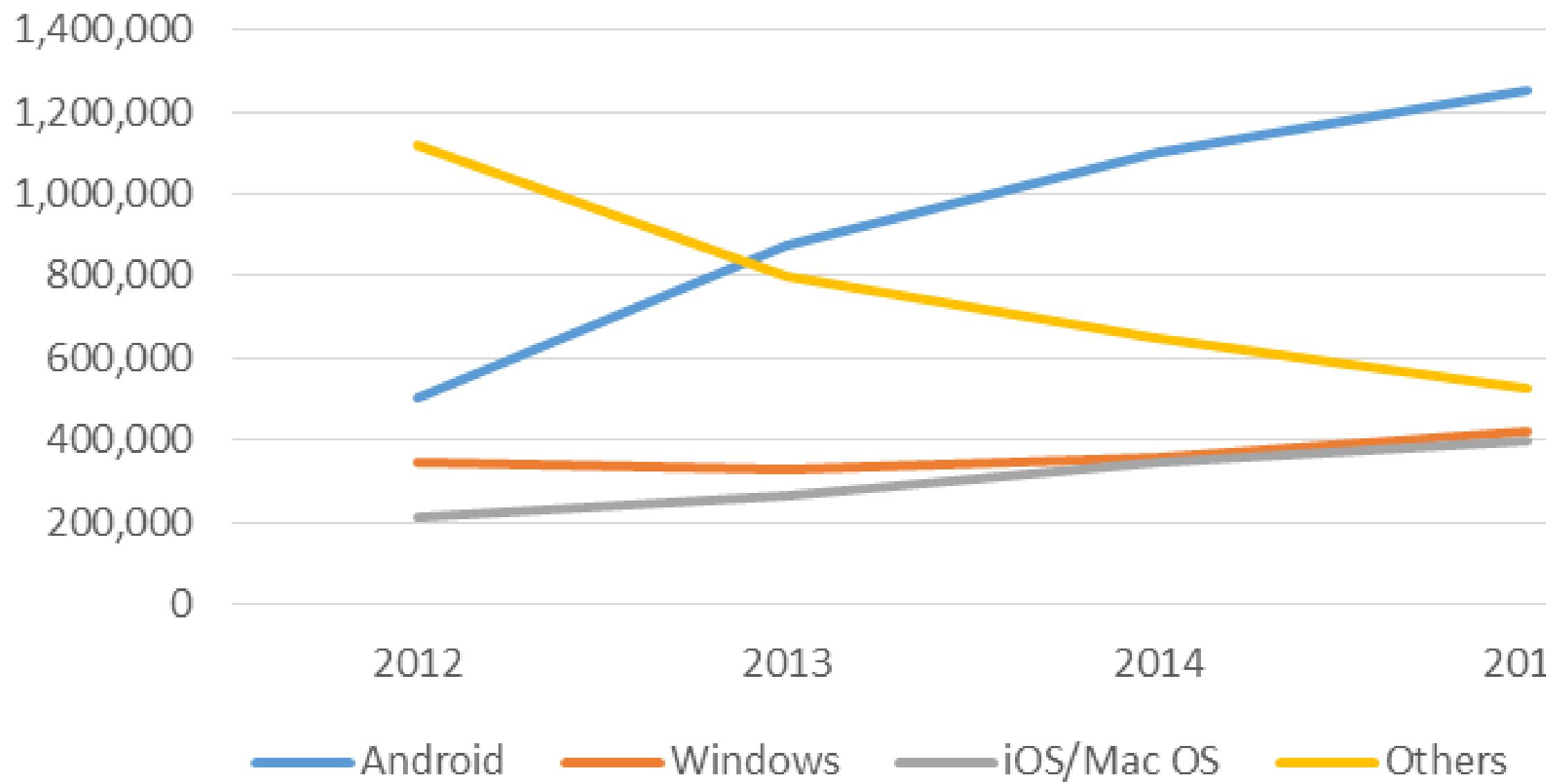
– Eric Schmidt

Vision (2011)

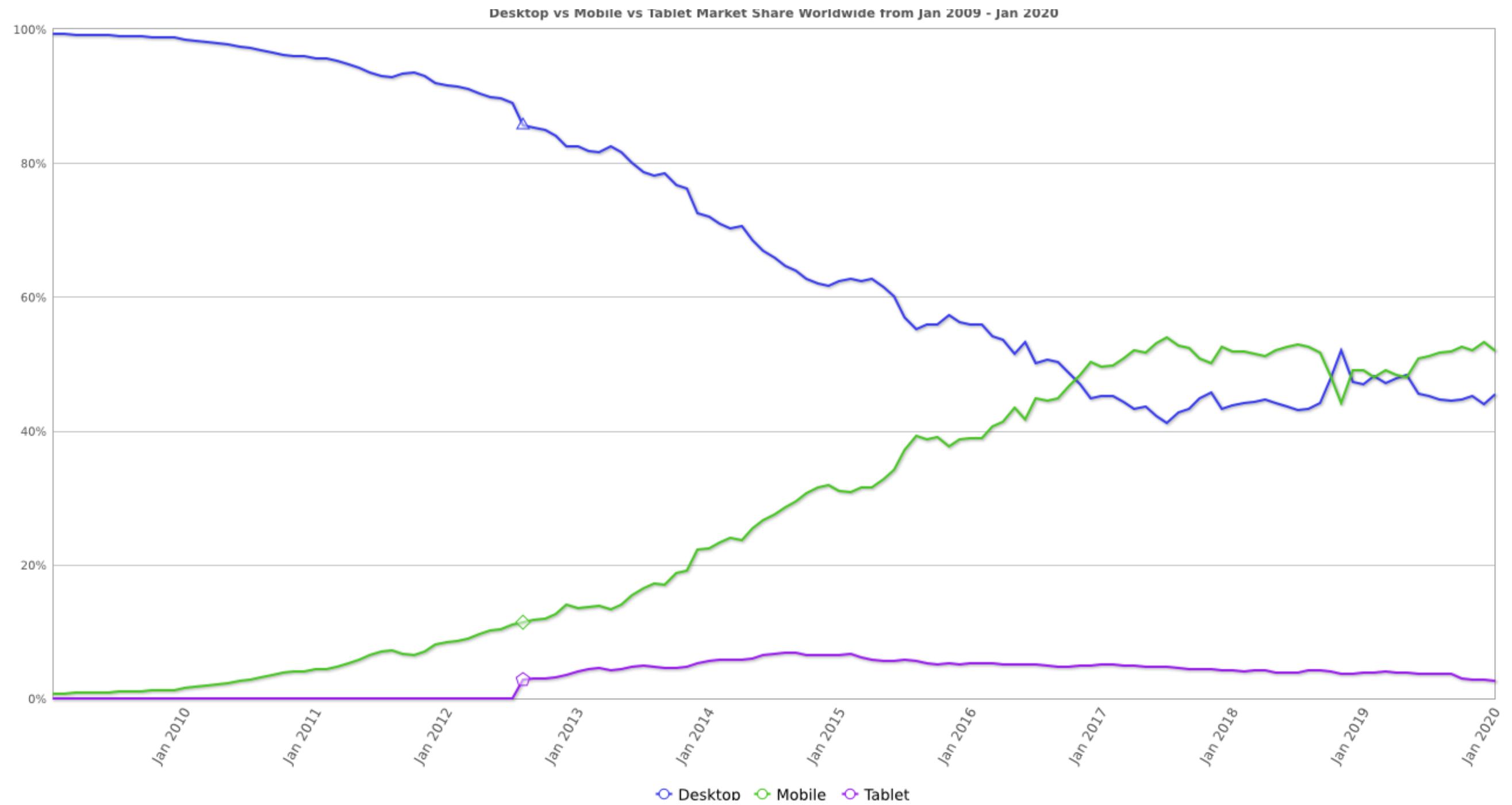
Top Five Smartphone Operating Systems, Worldwide Shipments, and Market Share, 2014Q2 (Units in Millions)

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%

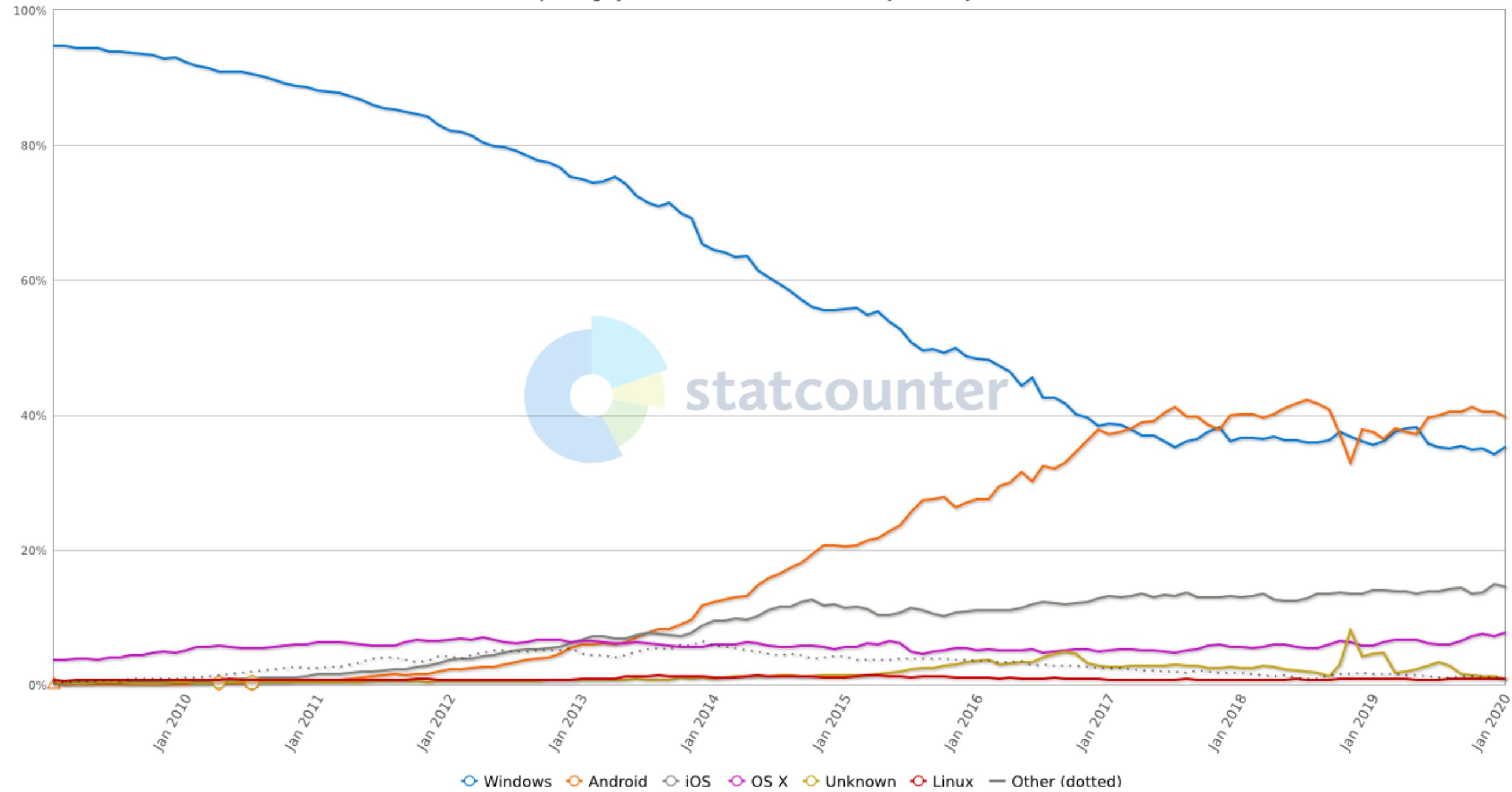
Worldwide Device Shipments by Operating System (Thousands of Units)



Android vs.



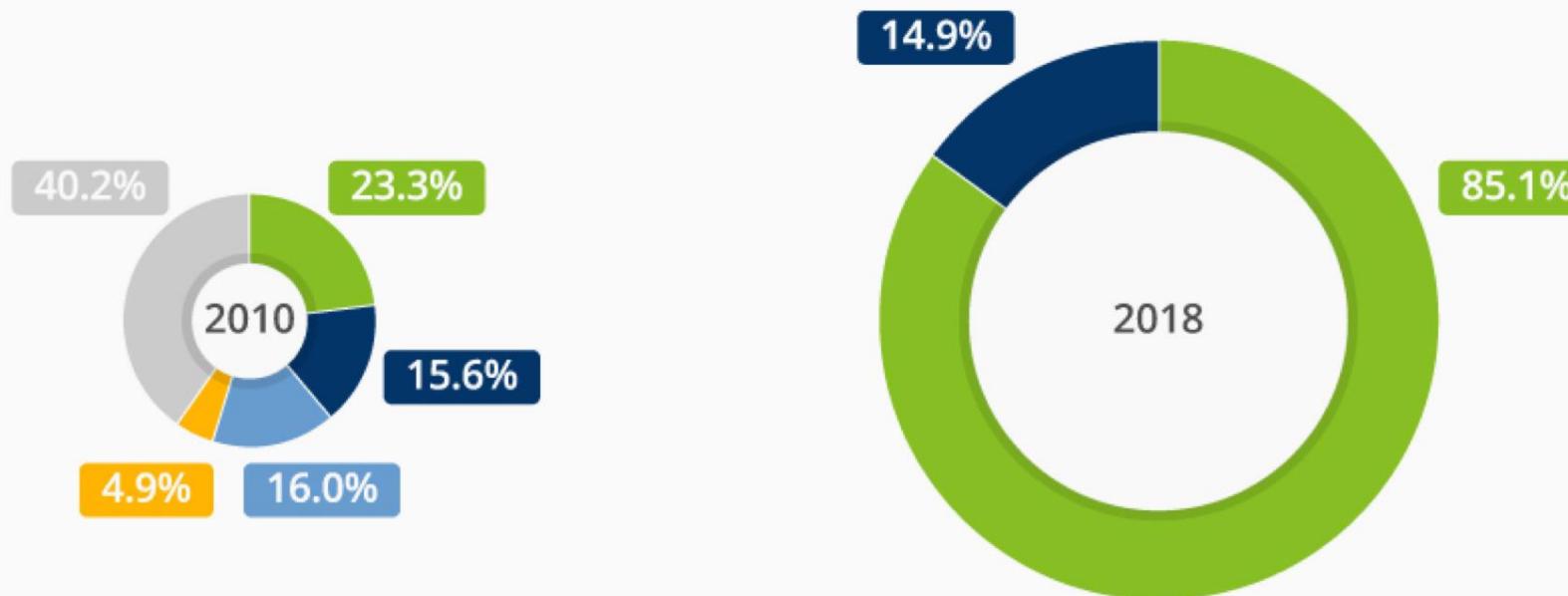
Operating System Market Share Worldwide from Jan 2009 - Jan 2020



The Smartphone Duopoly

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

● Android ● iOS ● BlackBerry ● Windows Phone ● Others



Total sales

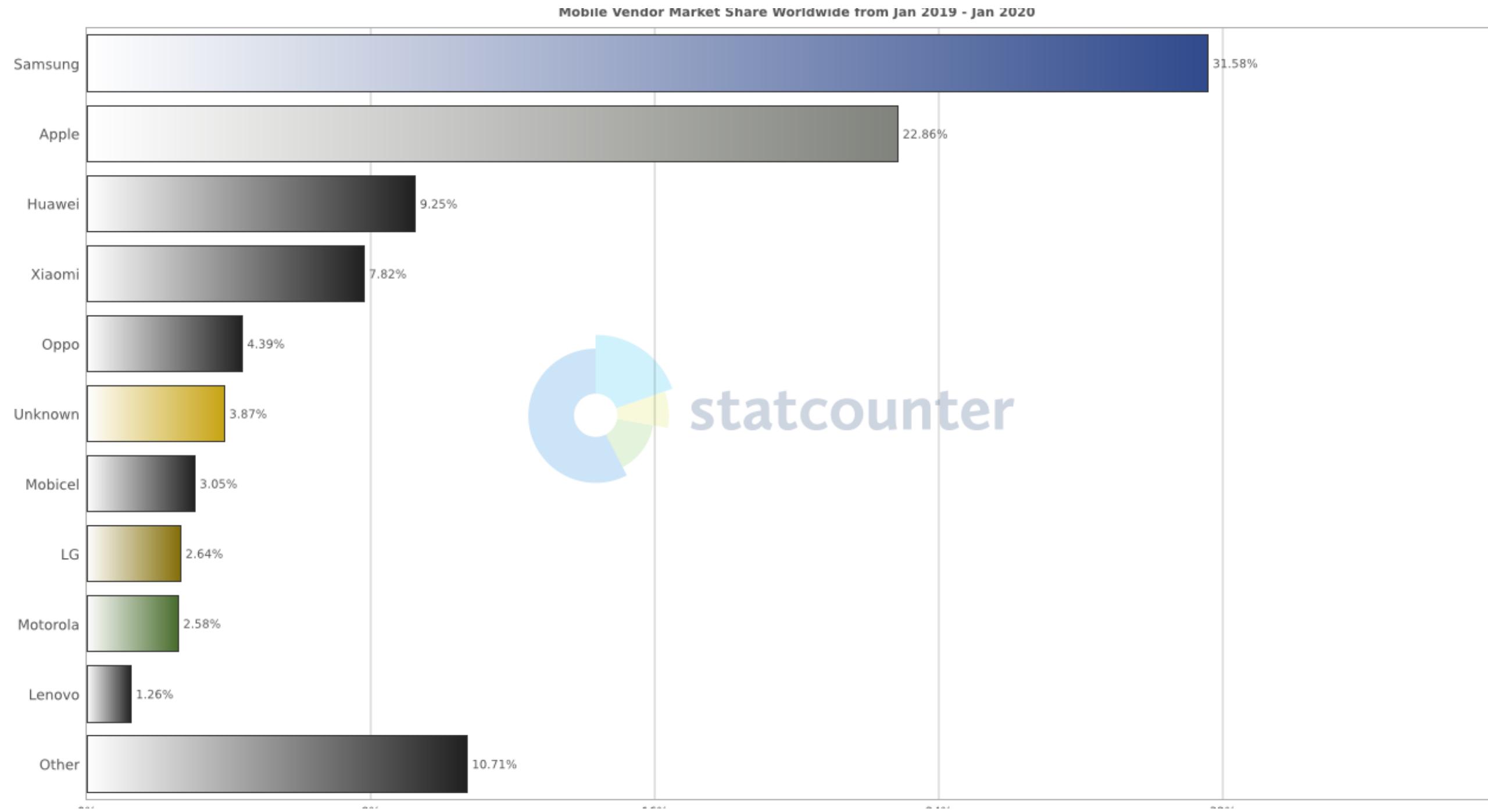
305m

1,405m



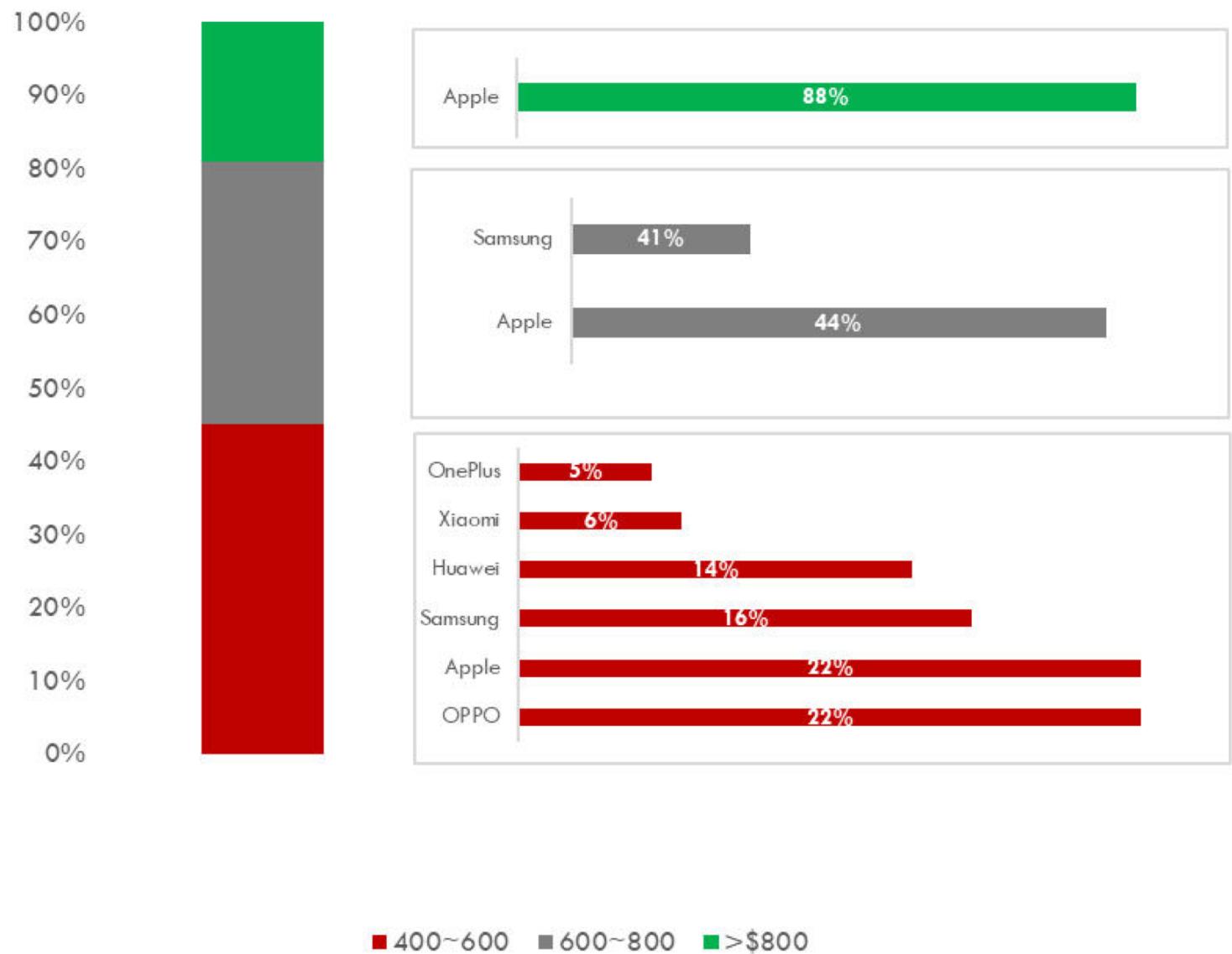
@StatistaCharts Source: IDC

statista

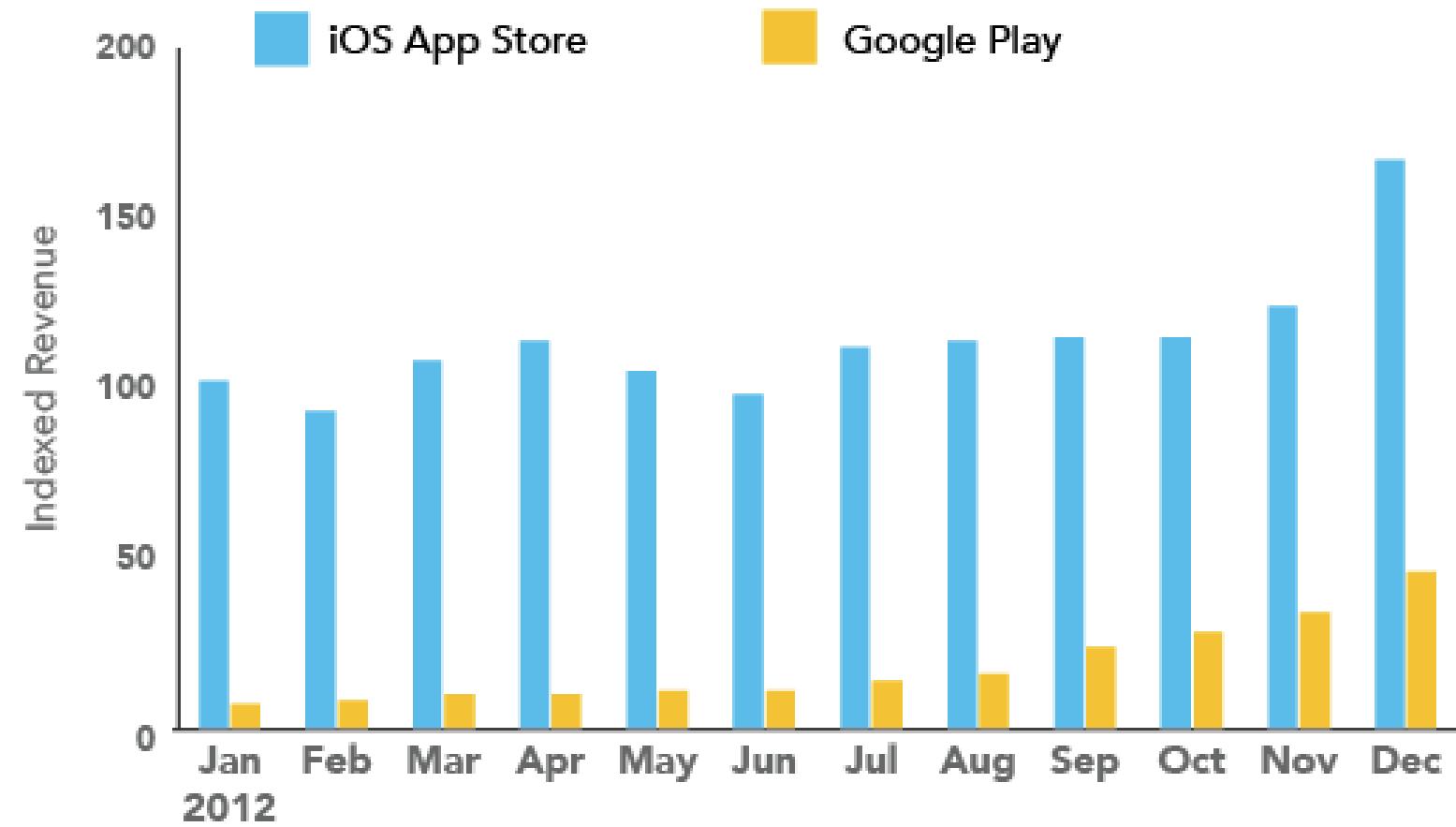


statcounter

Premium Smartphone Segment Price Tier Split and OEM Share



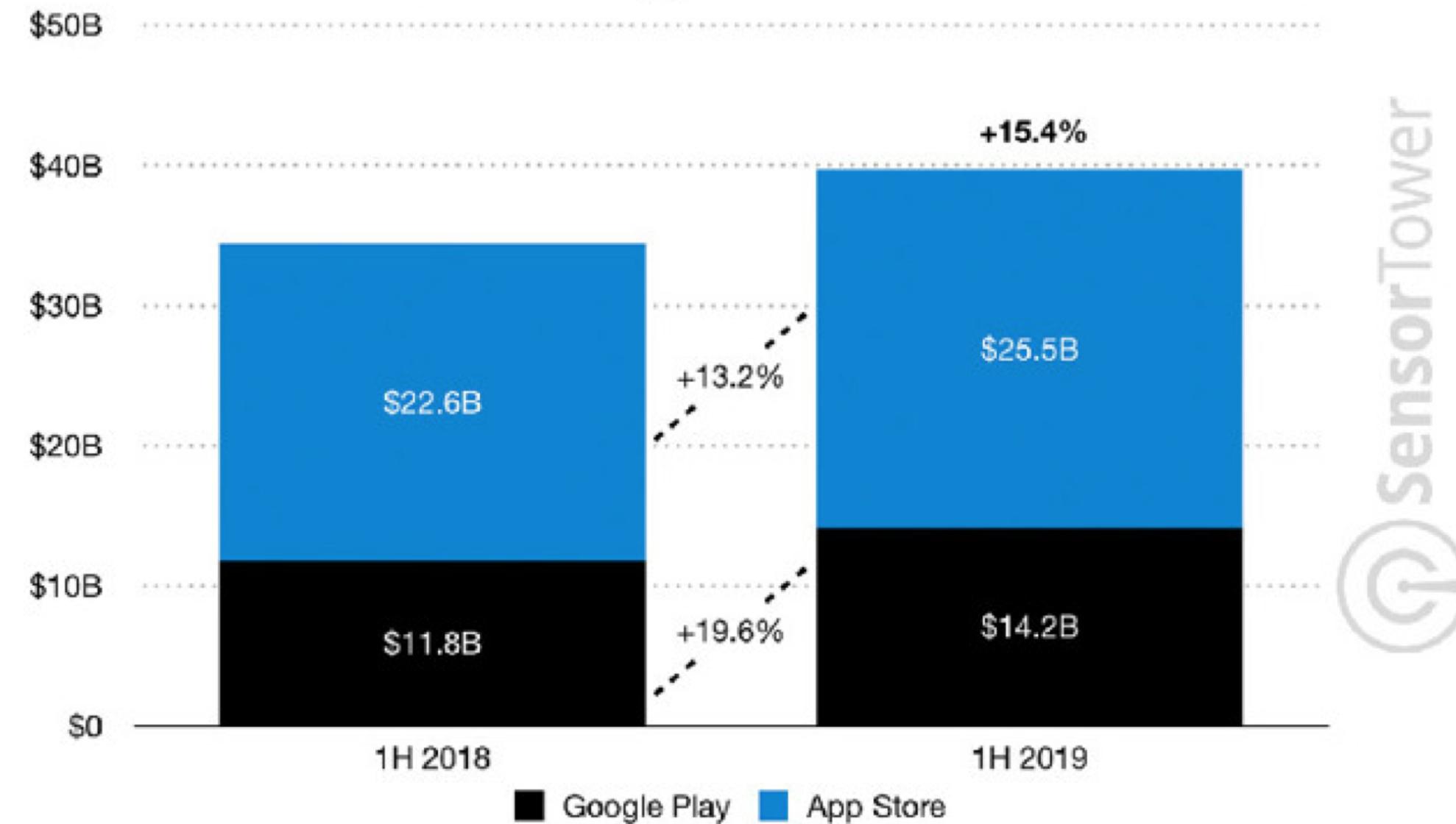
iOS App Store vs Google Play Revenue



App Annie

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

Worldwide Gross App Revenue - First Half 2019



SensorTower



SensorTower Data That Drives App Growth

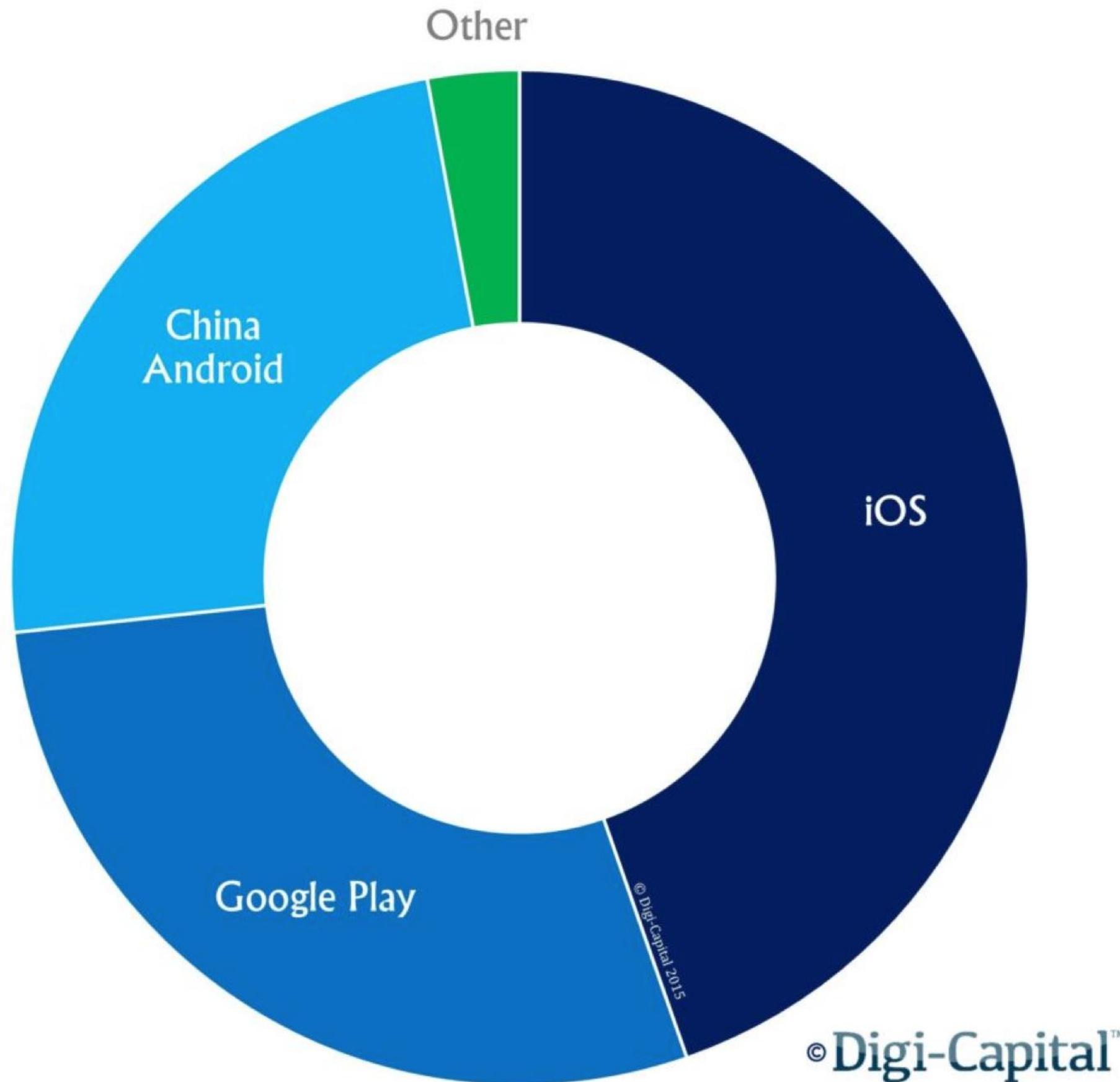
sensortower.com

Worldwide App Downloads and Revenue by Store

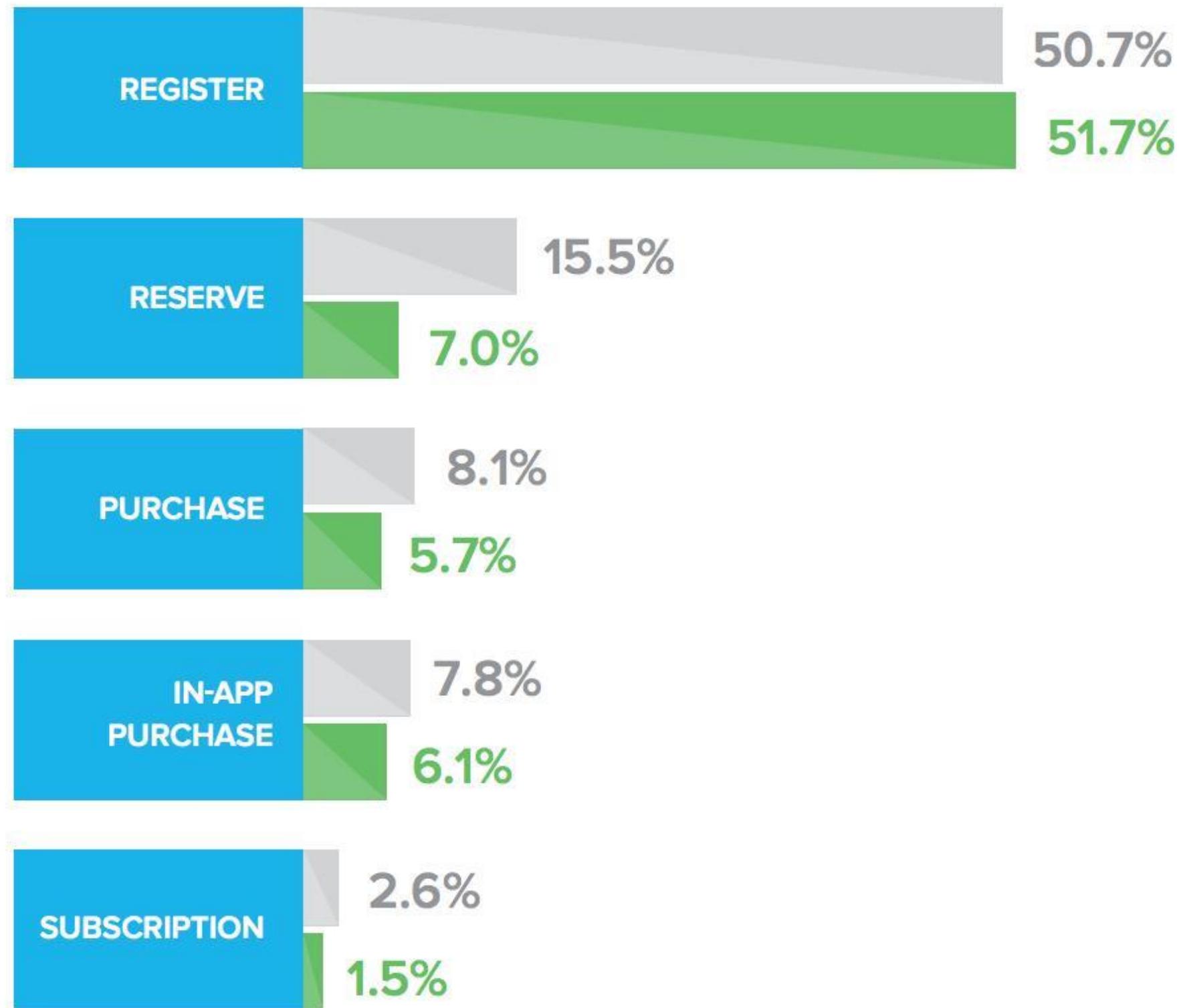


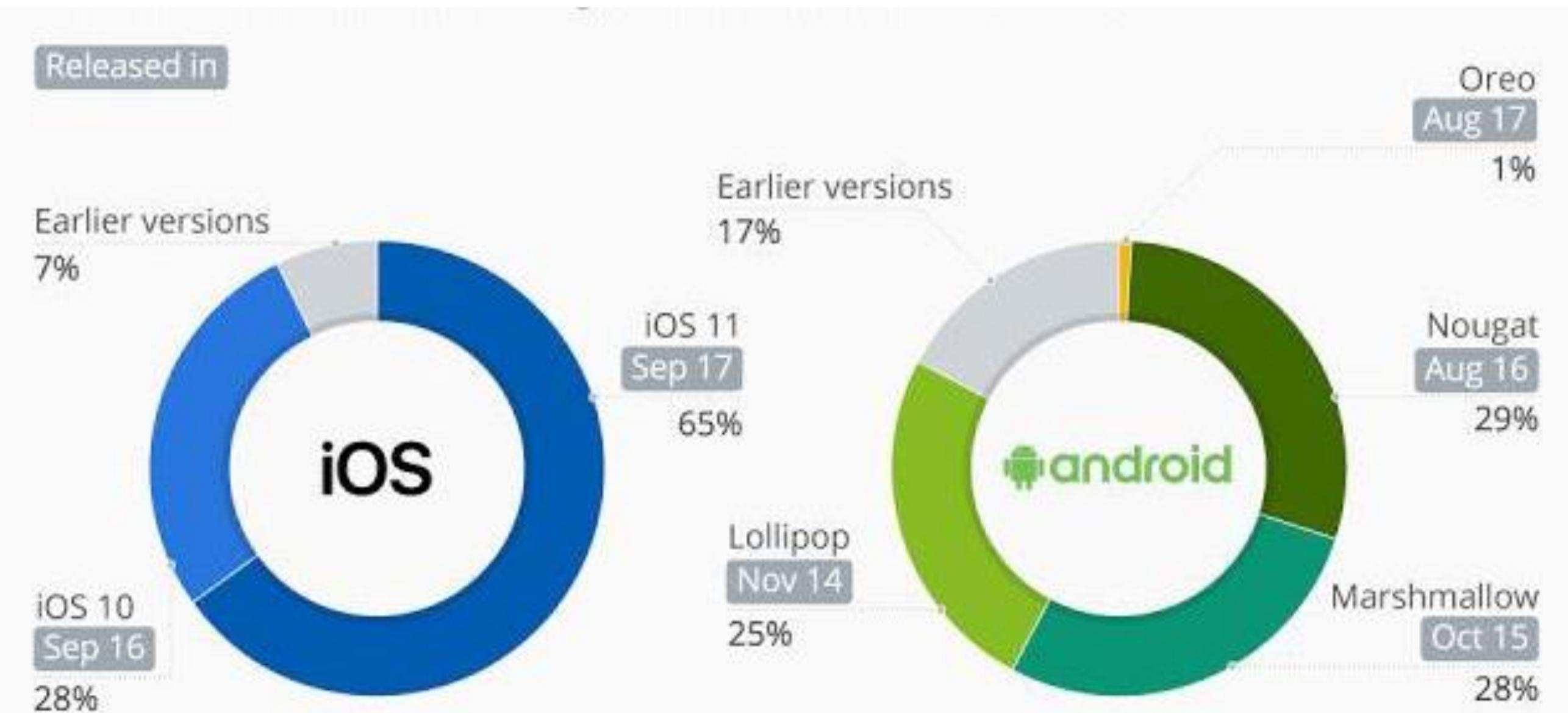
Compared to Q1 2016, the downloads gap remained the same, but iOS extended its lead over Google Play in revenue by 10 percentage points.

Global app stores revenue value share 2014



POST-INSTALL ACTIVITY ENGAGEMENT RATES





Data collected by the App Store
on January 18, 2018

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

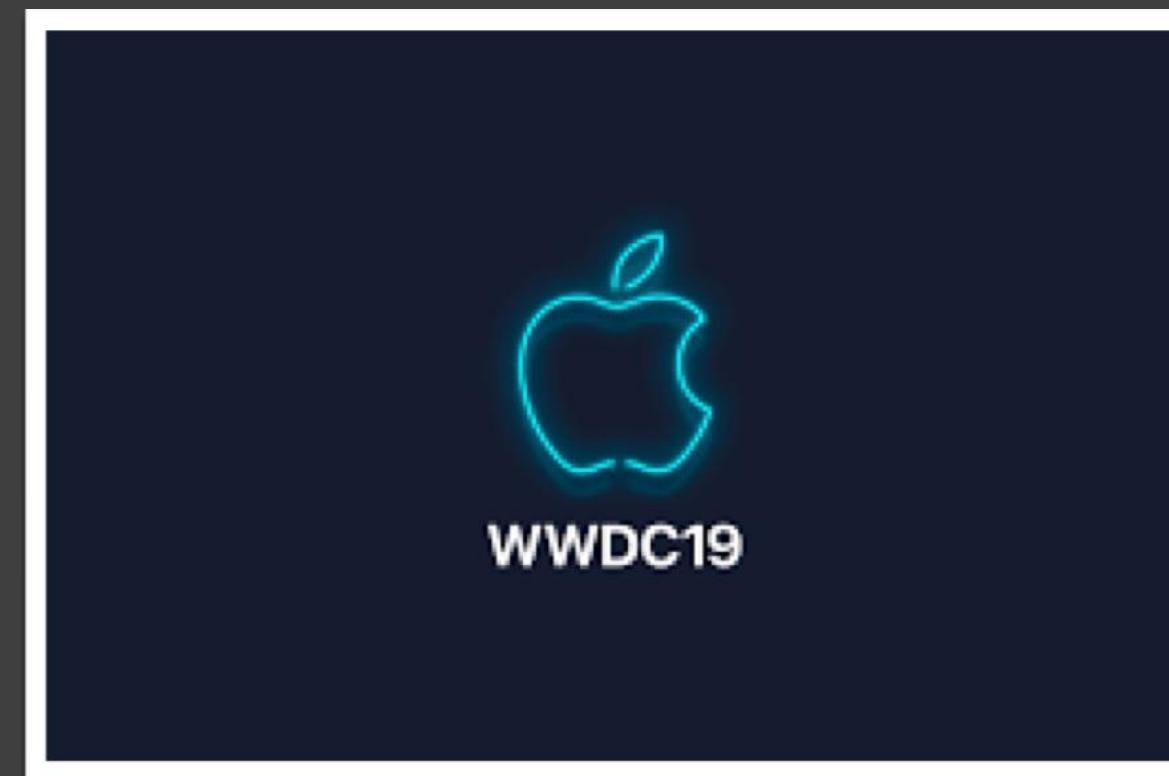
- Although Android can be used on many types of devices, they are not always supported by Google;
- Devices must support quite a rich mix of capabilities in order to be certified as Android compatible;
- Only “compatible” devices have access* to Google Play (the Android Market);

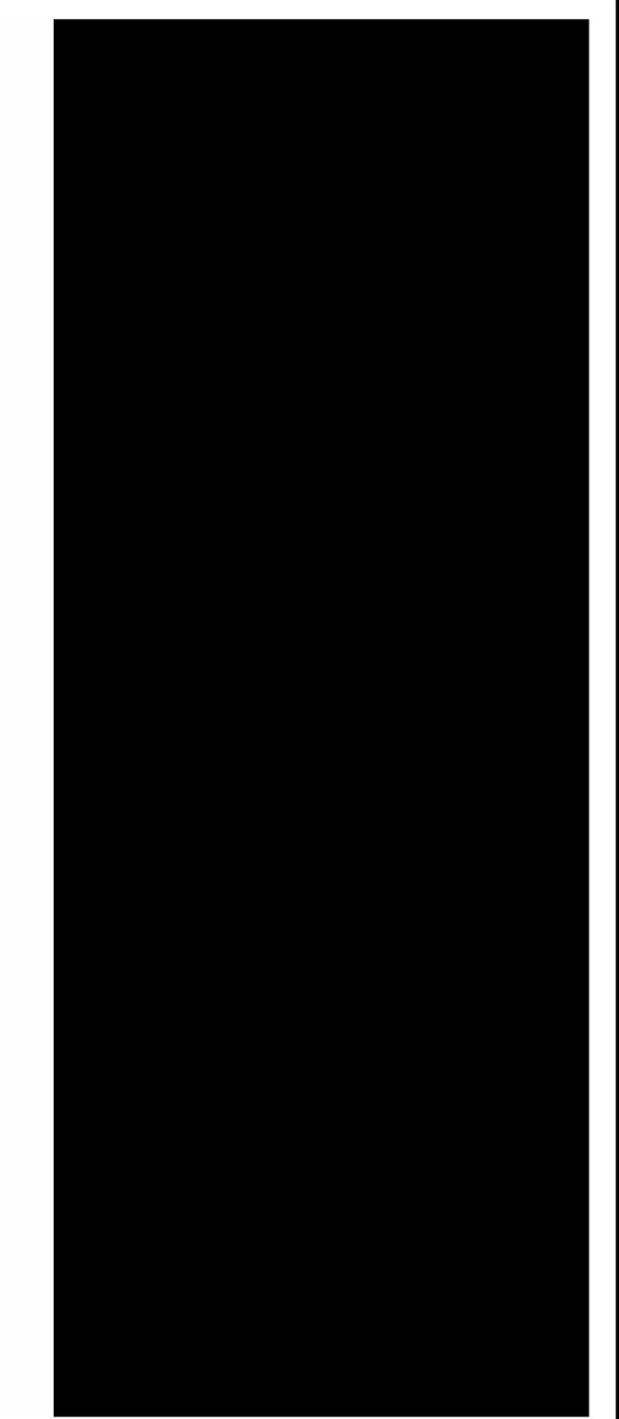
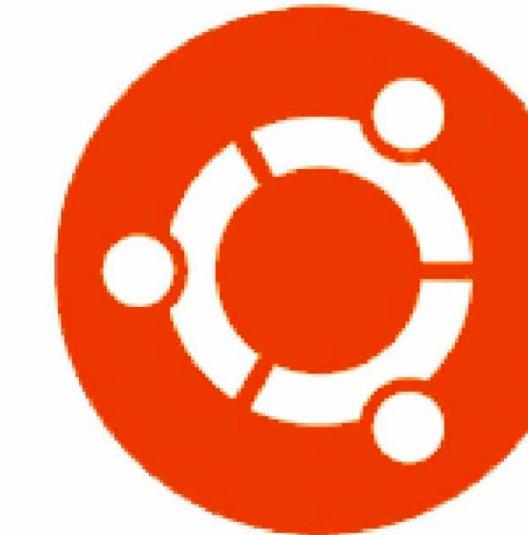
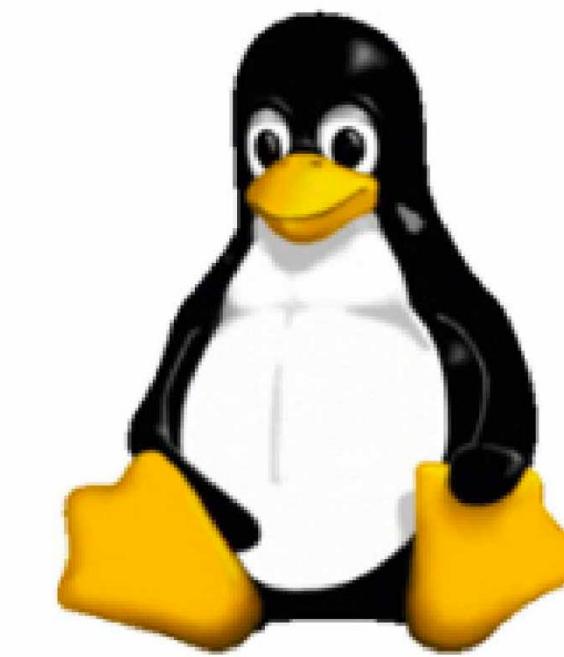
Android: Why develop?

- 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.

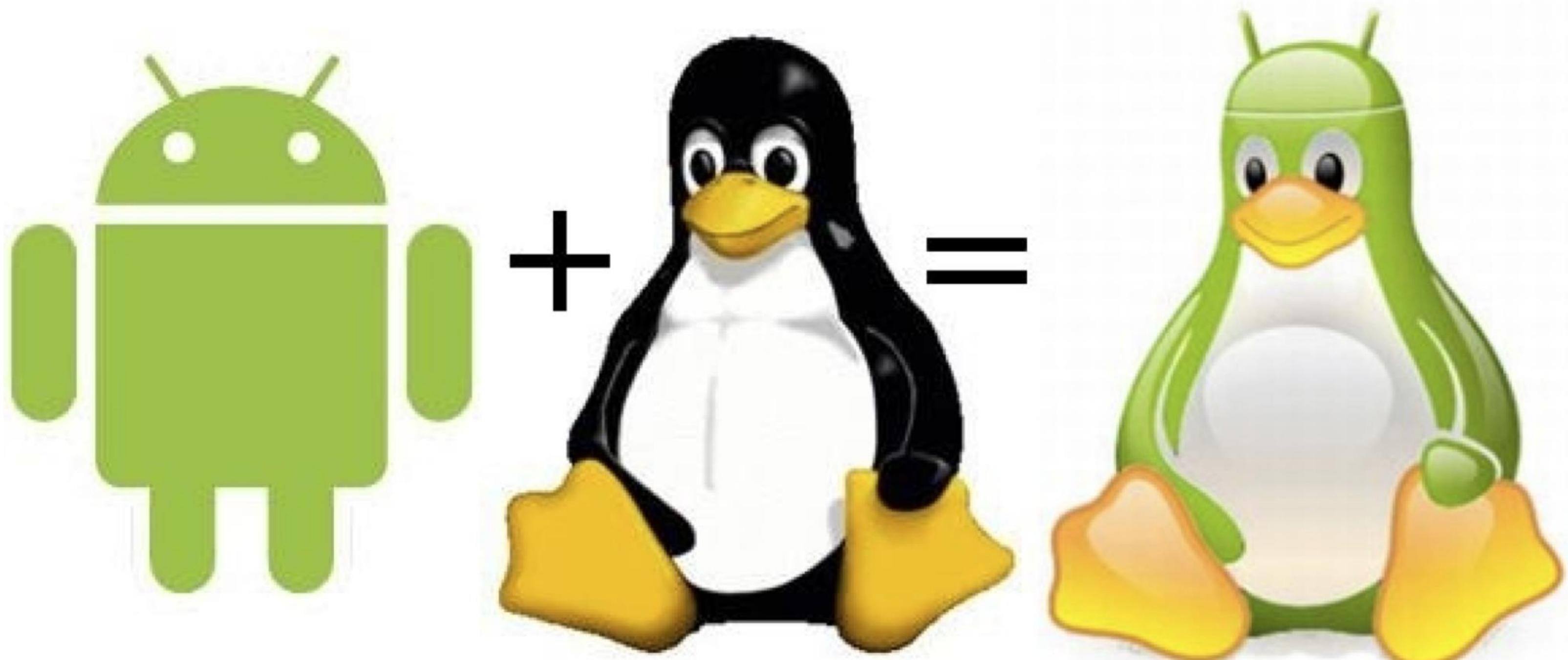
Android: Success

- 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

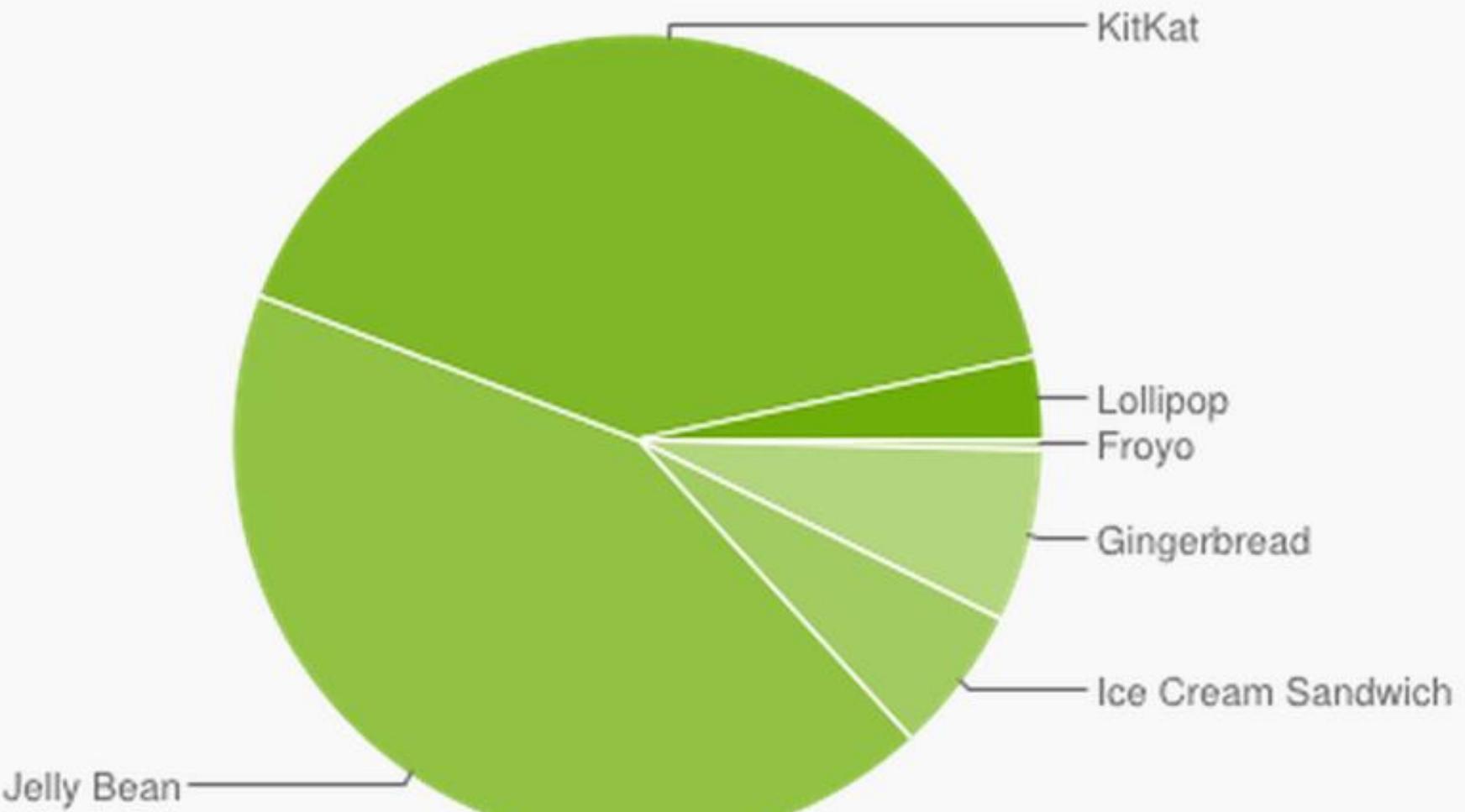


Pie
9.0



android

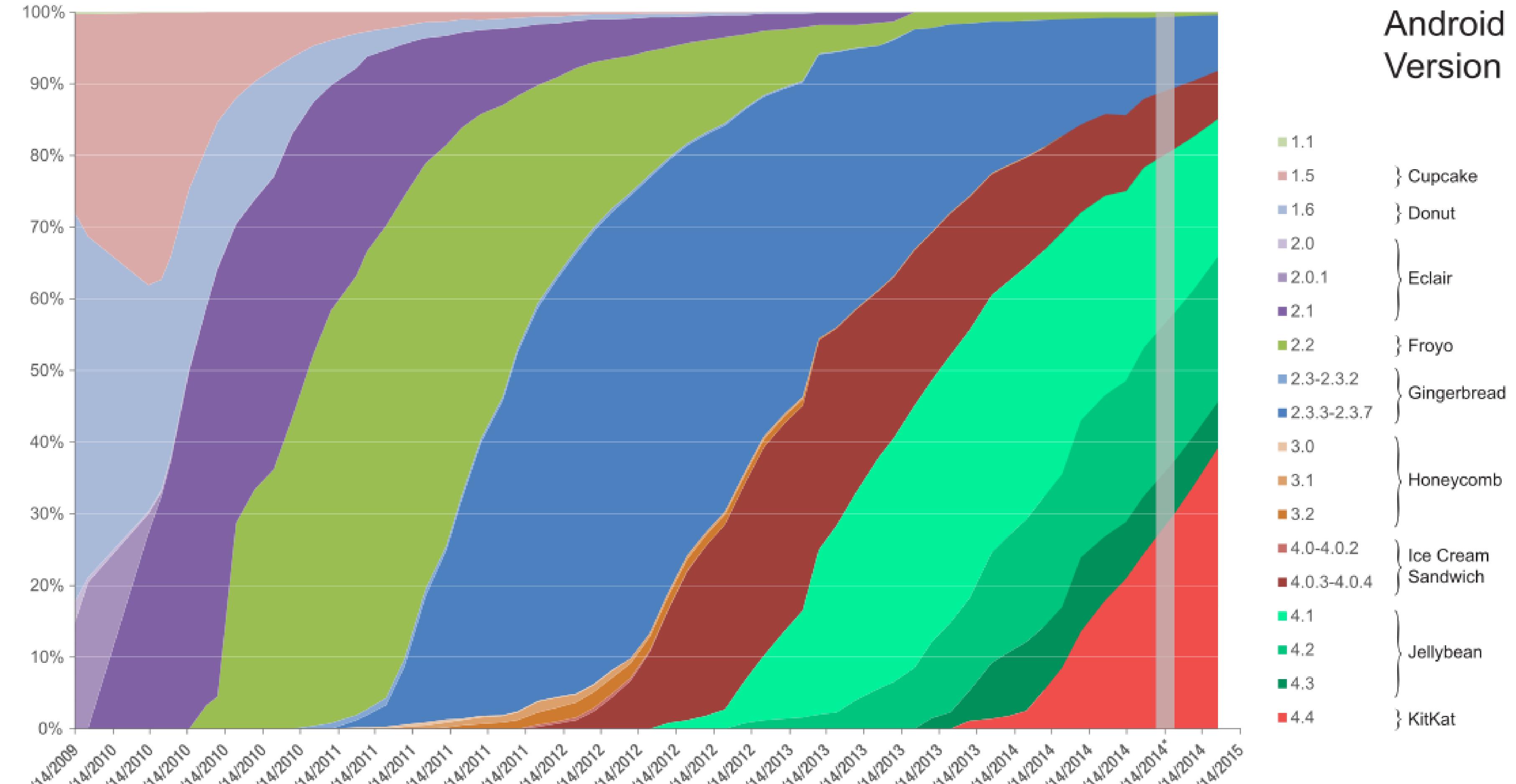
Version	Codename	API	Distribution
2.2	Froyo	8	0.4%
2.3.3 - 2.3.7	Gingerbread	10	6.9%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	5.9%
4.1.x	Jelly Bean	16	17.3%
4.2.x		17	19.4%
4.3		18	5.9%
4.4	KitKat	19	40.9%
5.0	Lollipop	21	3.3%



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

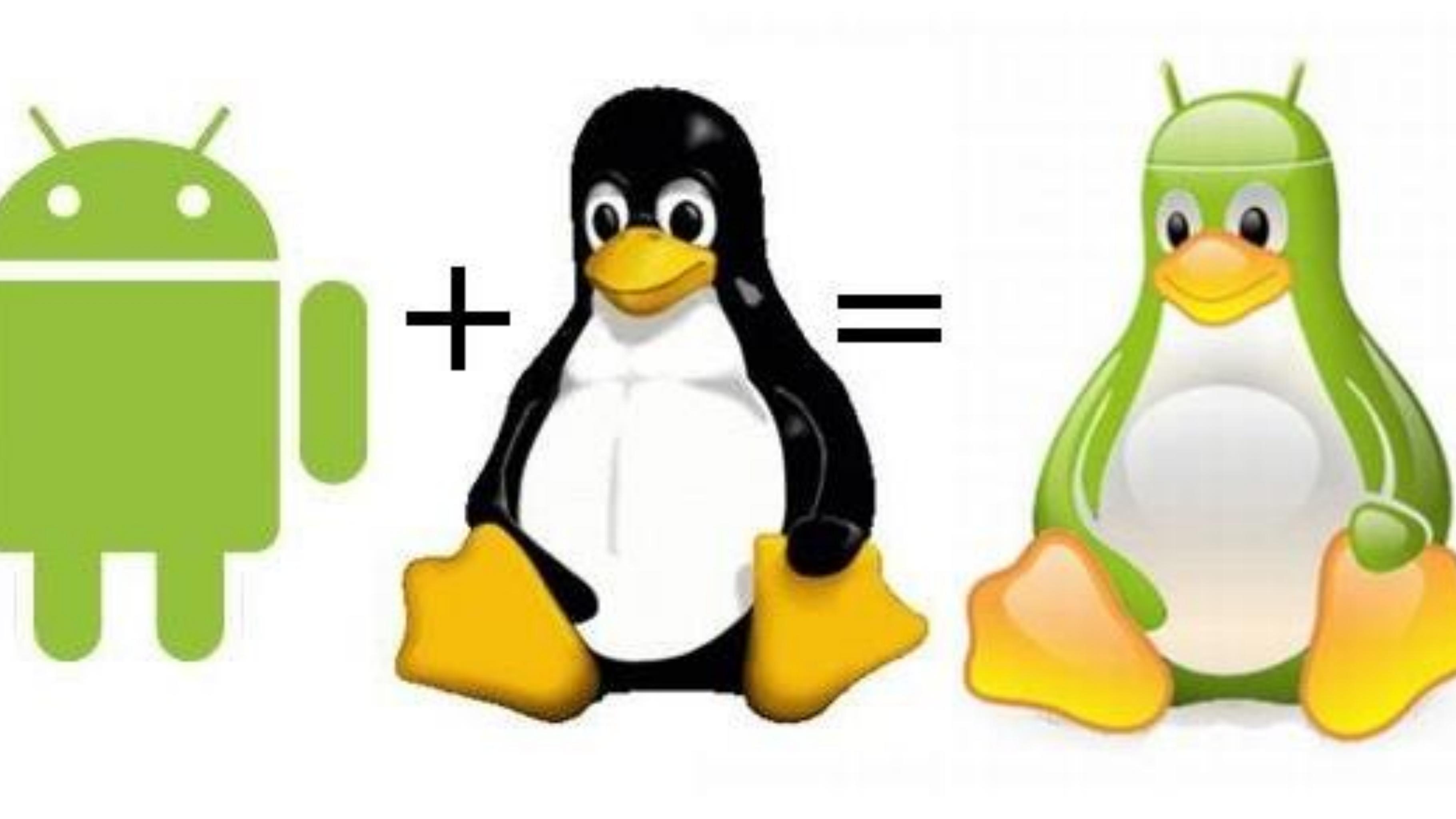
Using the compatibility (support) libraries, you can develop in 5.0 and min-build-

Android Version

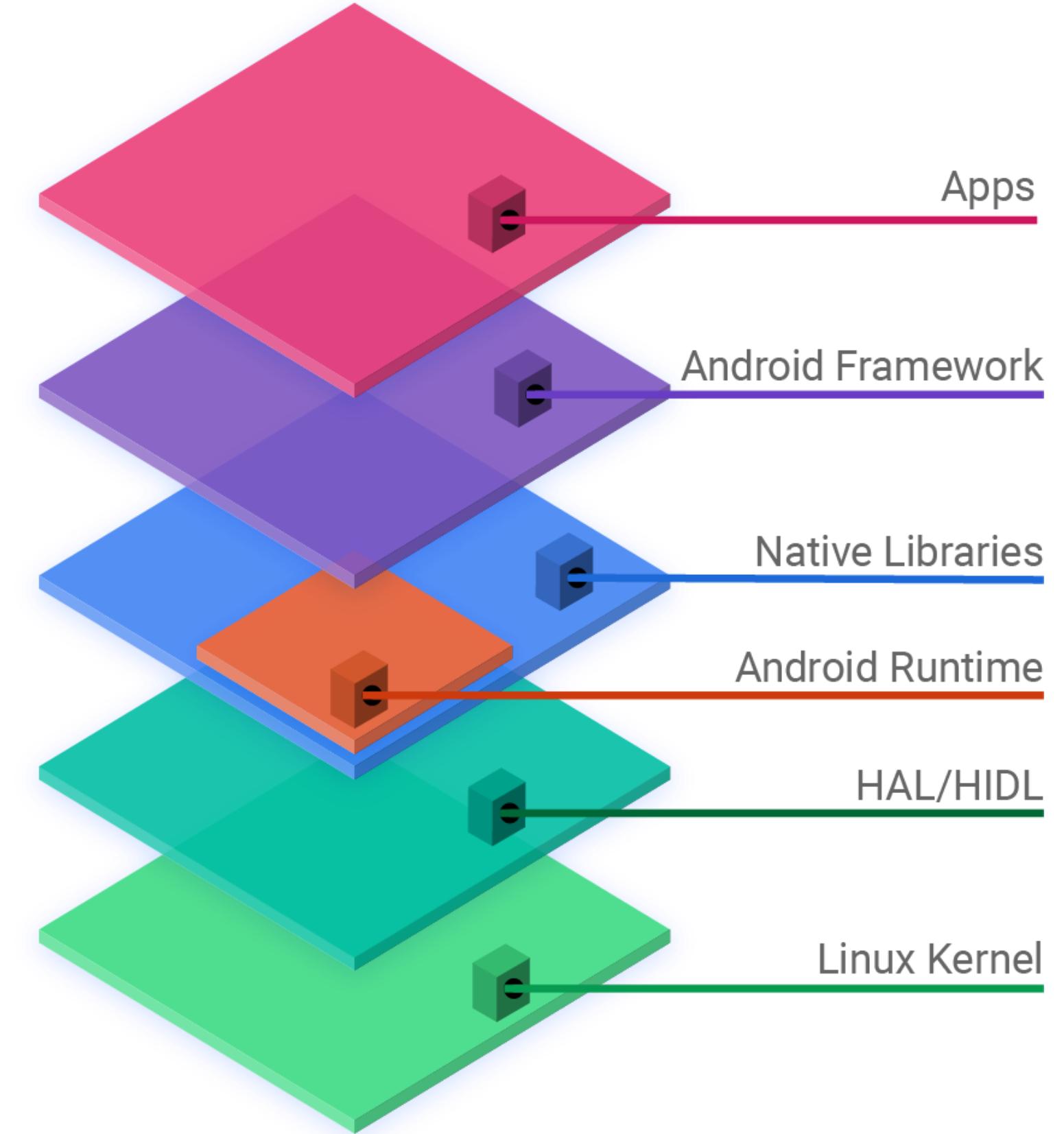


Code/Version/API map

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 6
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

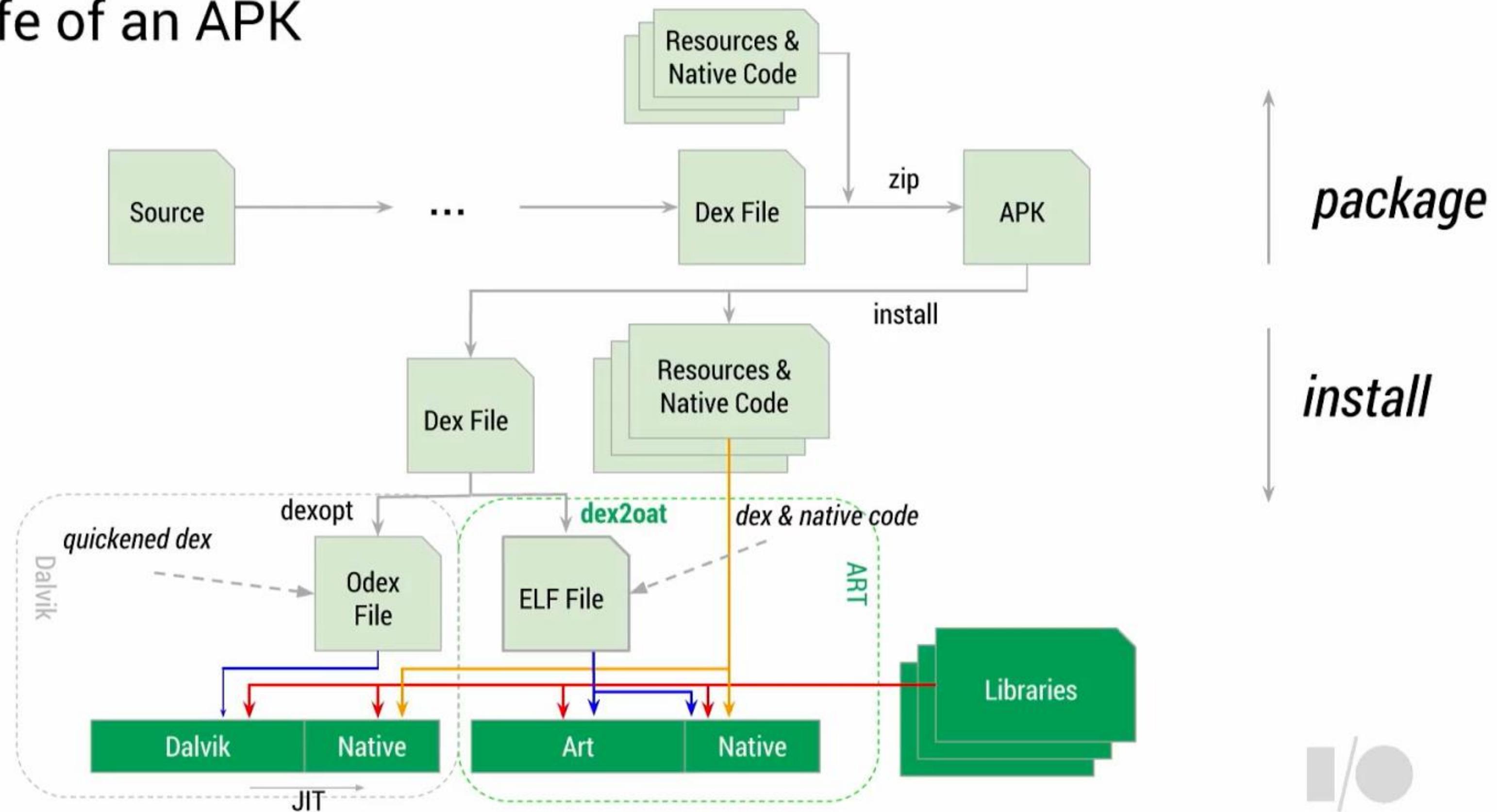




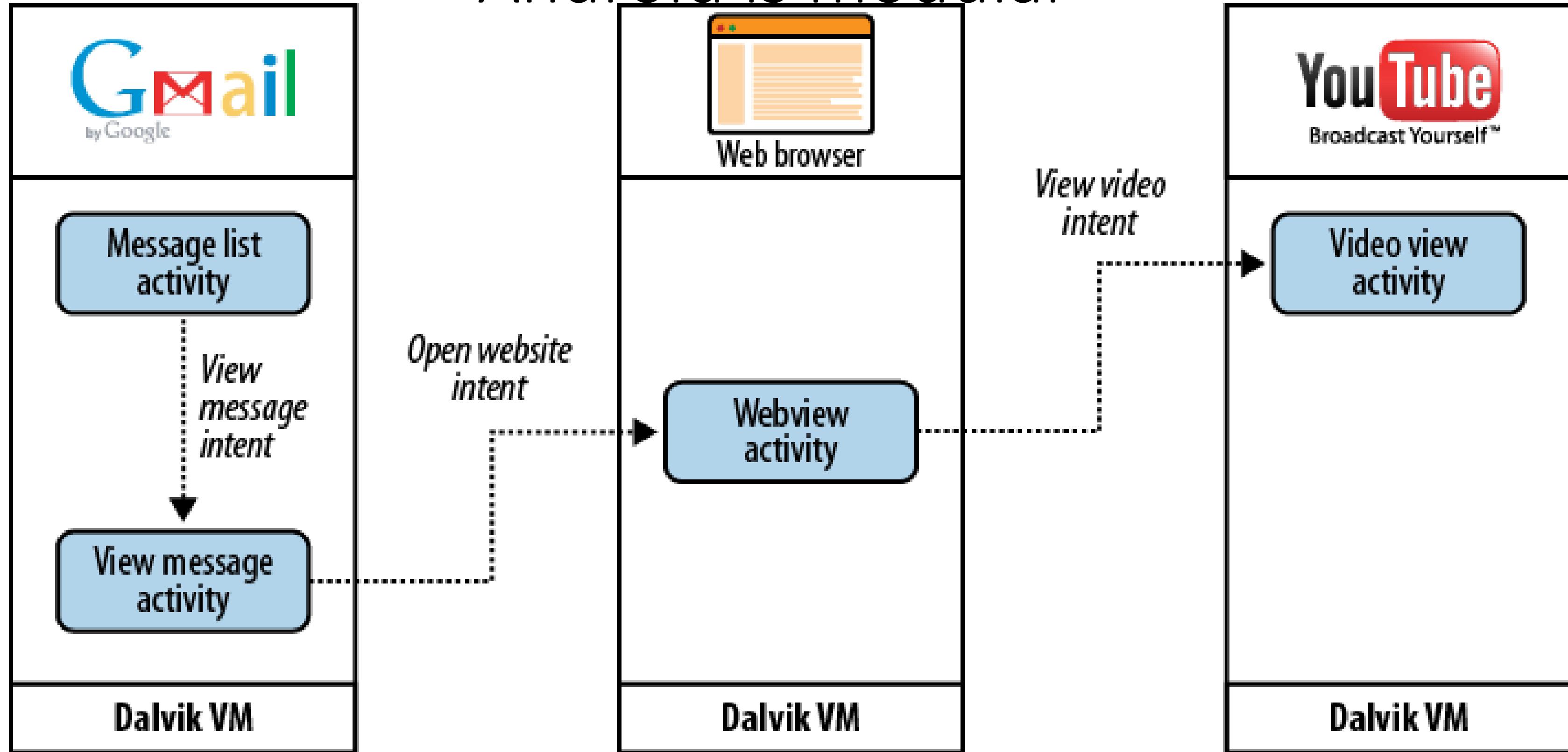
Architecture

- 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

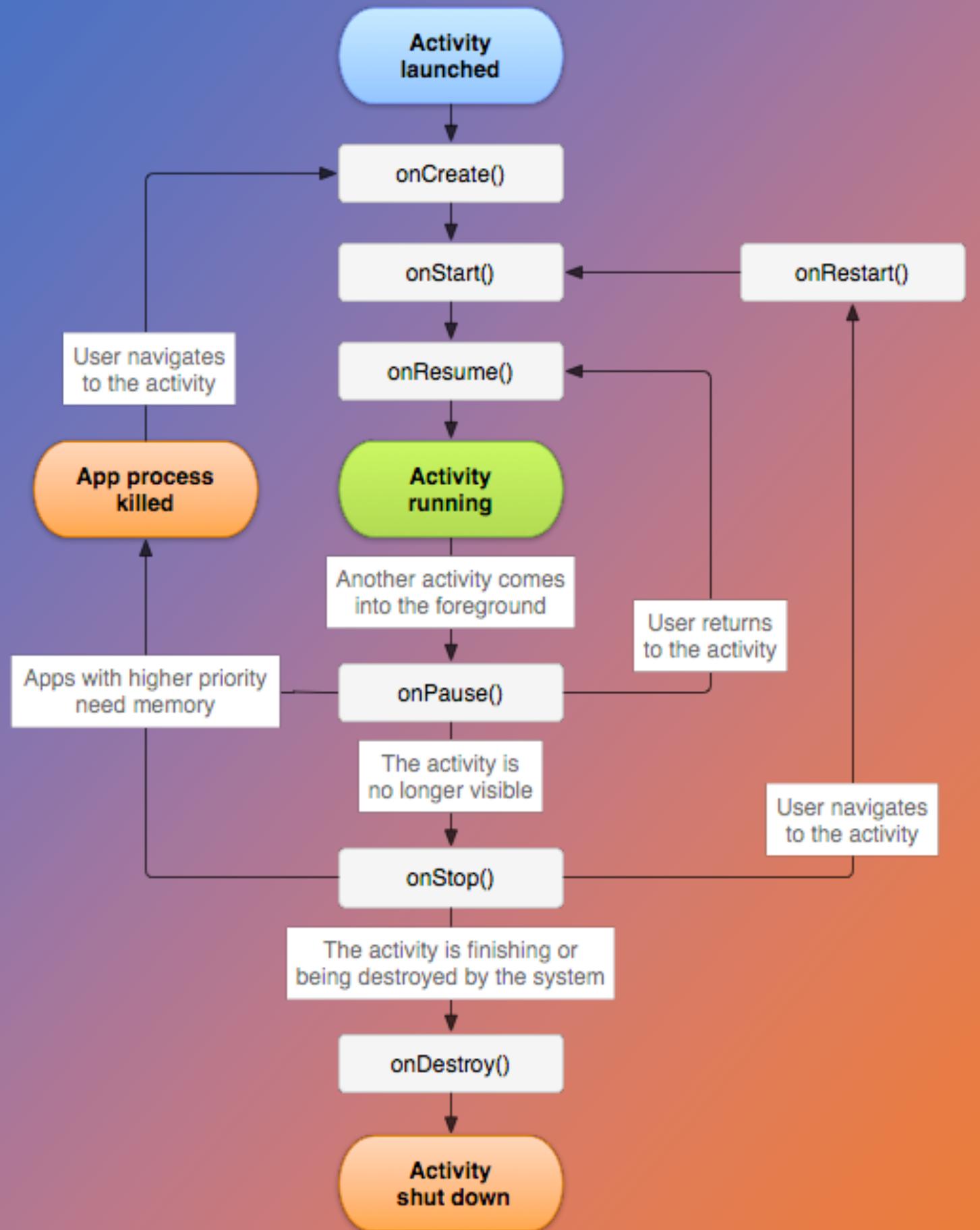
Store and retrieve data and make it accessible to all applications

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



Intents

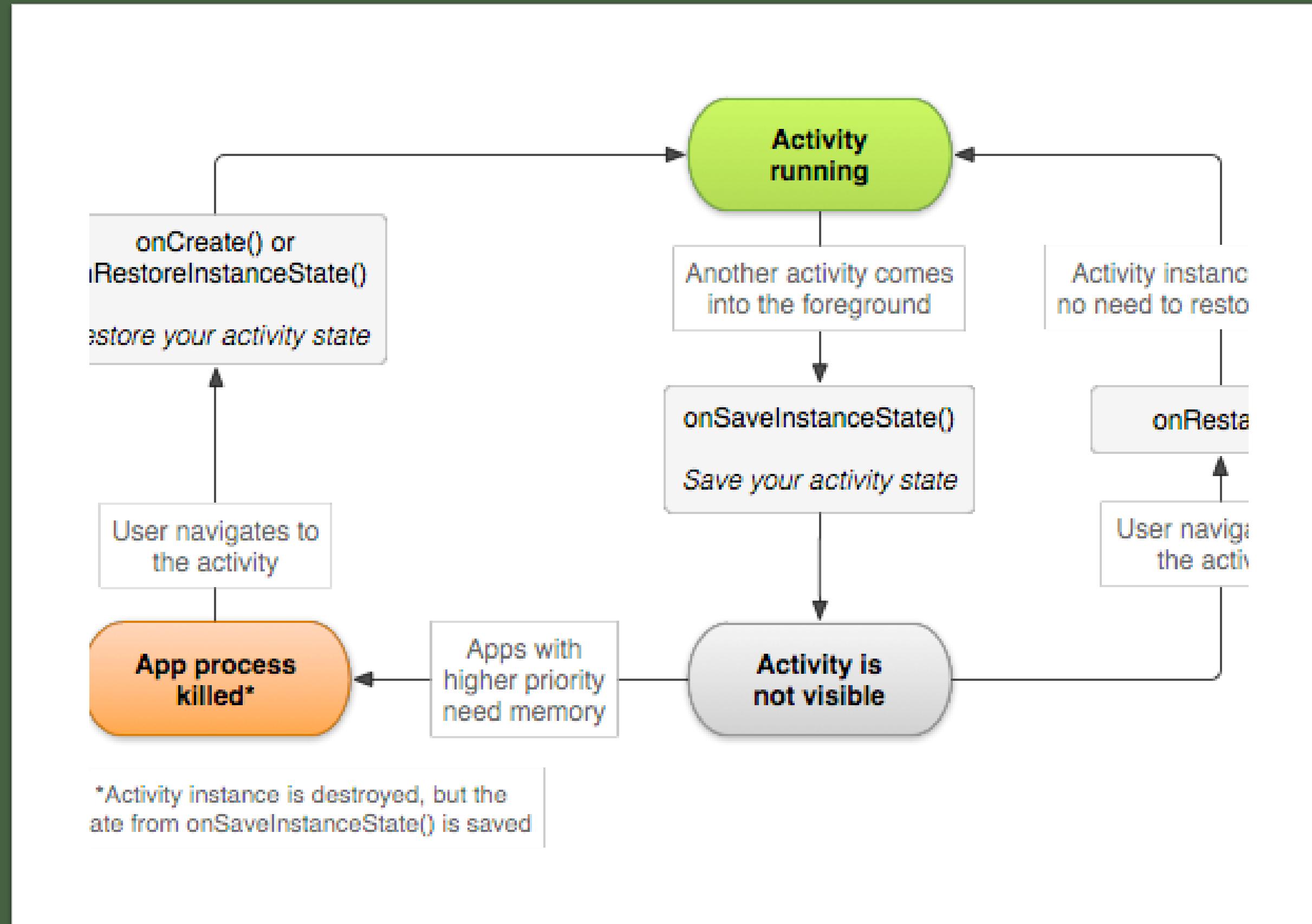
Hold the content of a message
Example: convey a request for an activity to present an image to the user or let the user edit some text



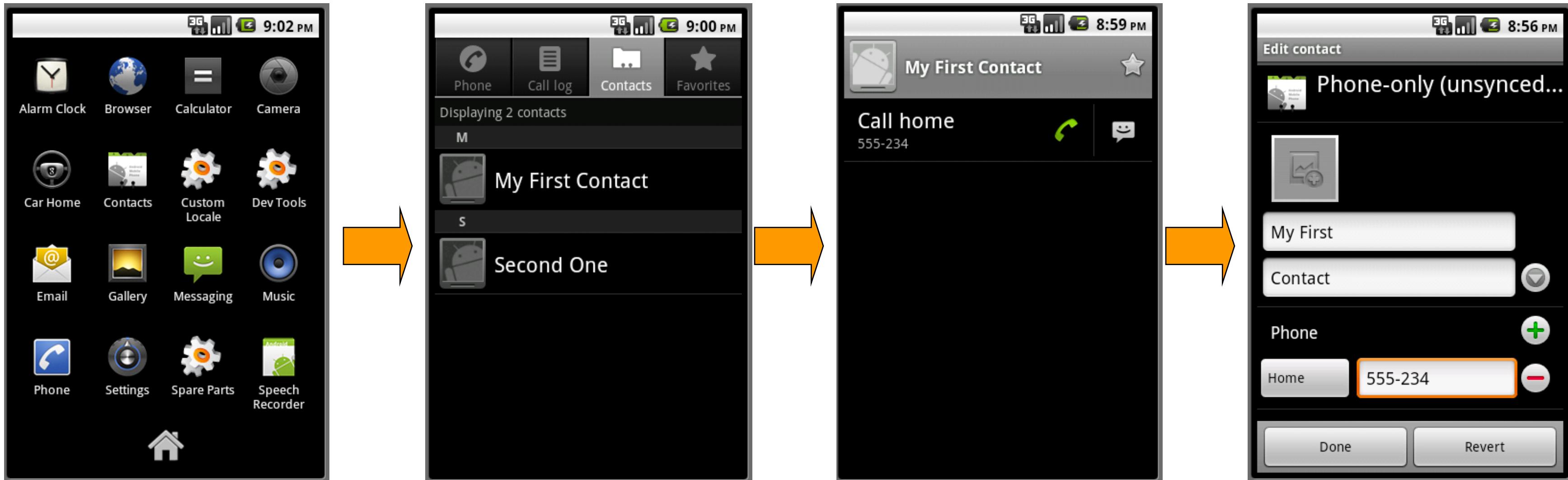
Activity

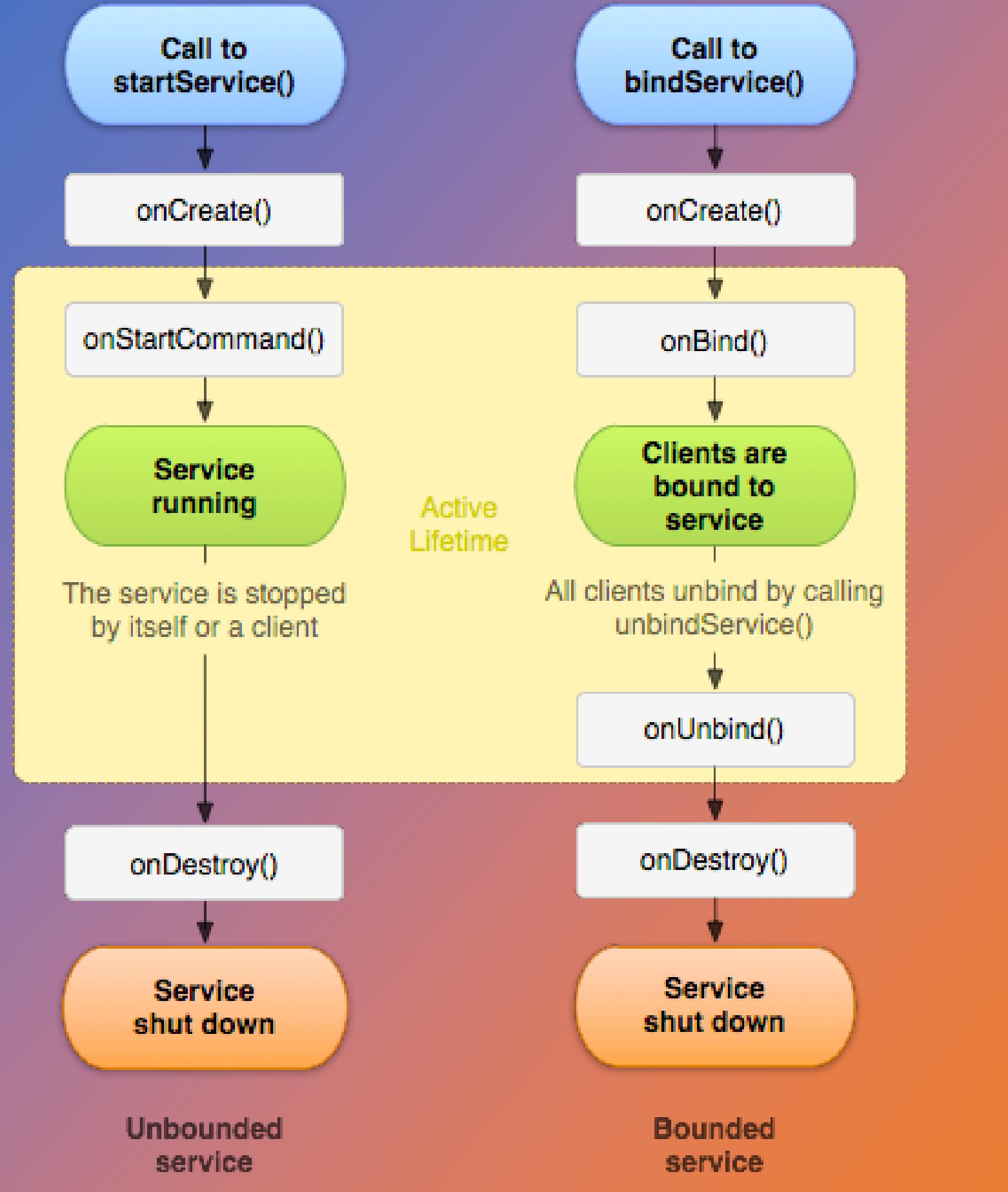
- System upcalls component as its state changes due to user actions.
- If another activity is started, the activity is paused.
- If a paused activity is not visible to the user, it is stopped.
- A stopped activity may be destroyed.
- And its app process may be killed.

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

