CS 213 SOFTWARE METHODOLOGY

Lily Chang

CS Department @ Rutgers New Brunswick

FALL 2023

Introduction to Android App Development

Lecture Note #17

Mobile App Development

- Smartphones and portable devices are increasing powerful and becoming the essential devices people use to complete their routine tasks with the help of the apps
- The two most well-known operating systems are the Android operating system, from Google, and iOS, from Apple.
- Other popular operating systems for smartphones include BlackBerry, Windows, and Symbian.
- The annual number of smartphones sold worldwide is now over one billion.

The Android Market

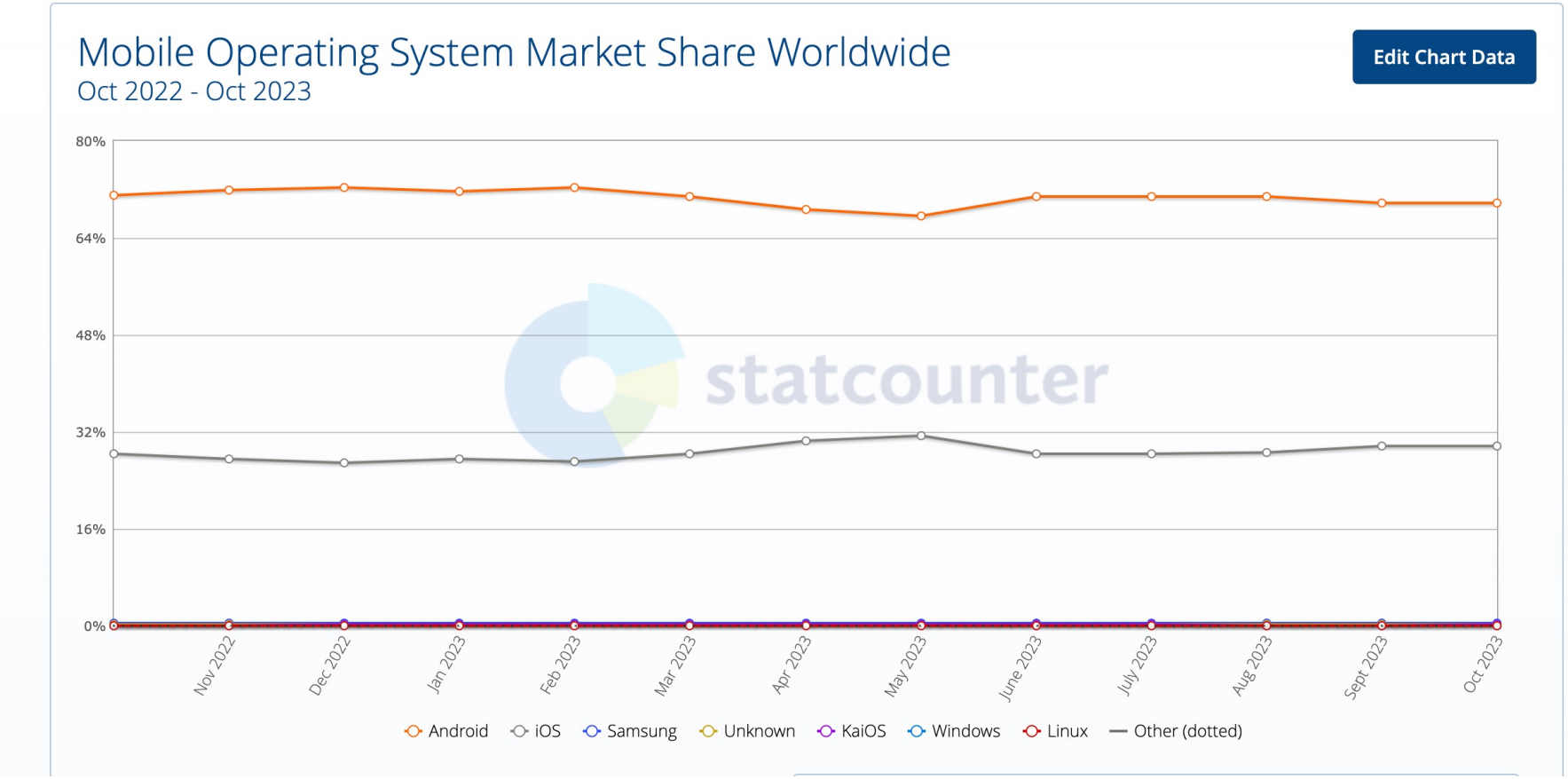
One of the main factors determining the success of a smartphone platform is the applications that support it.

It is clear from the success of the iPhone that applications play a very vital role in determining whether a new platform swims or sinks. Also, making these applications accessible to the general users is extremely important.

Users can simply use the Google Play application that is preinstalled on their Android devices to directly download third-party applications to their devices. Both paid and free applications are available in the Google Play Store

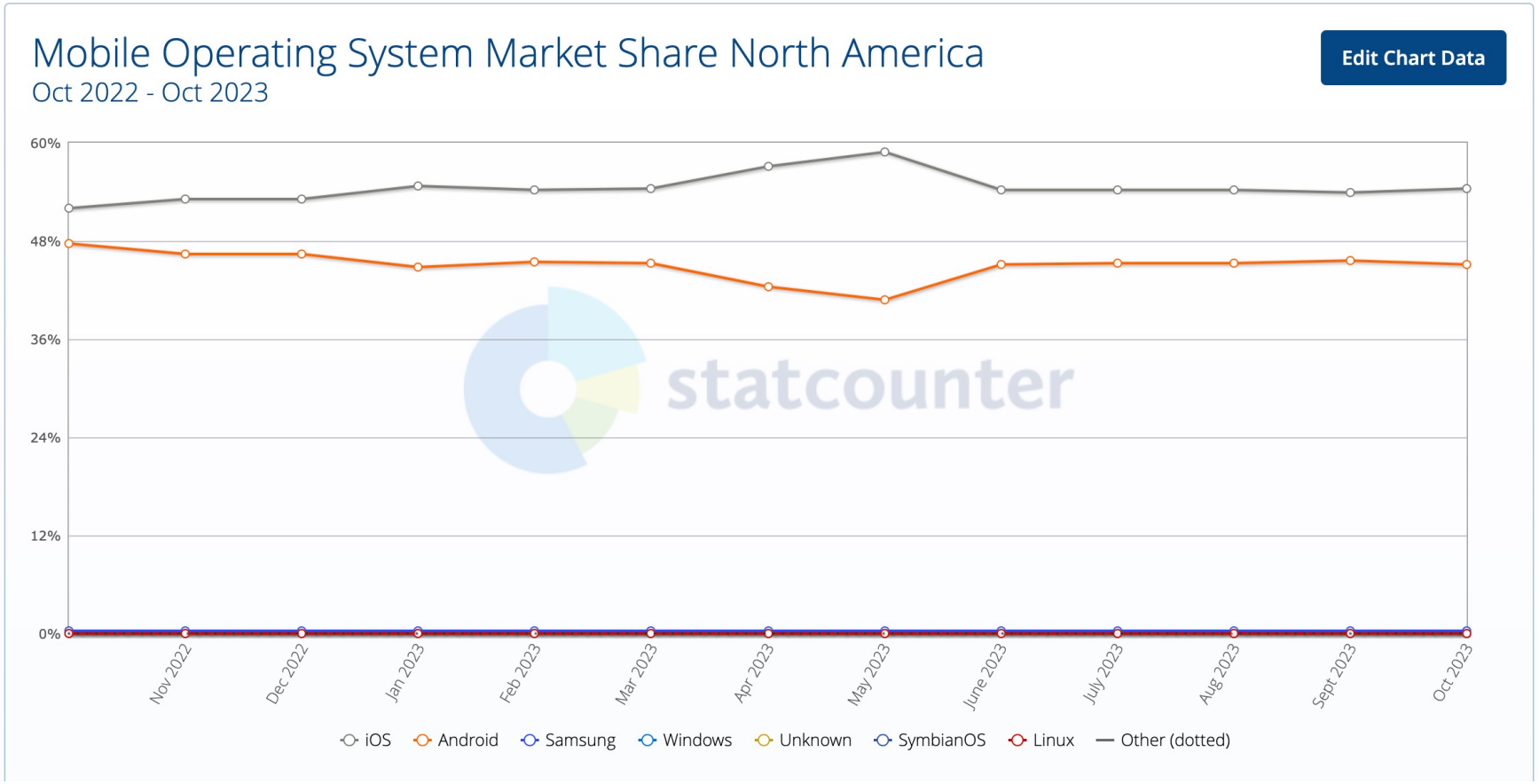
Android	iOS	Samsung	Unknown	KaiOS	Windows
69.64%	29.67%	0.37%	0.14%	0.11%	0.02%

Mobile Operating System Market Share Worldwide - October 2023



iOS	Android	Samsung	Windows	Unknown	Linux
54.4%	45.11%	0.3%	0.07%	0.04%	0.03%

Mobile Operating System Market Share in North America - October 2023



What is Android?

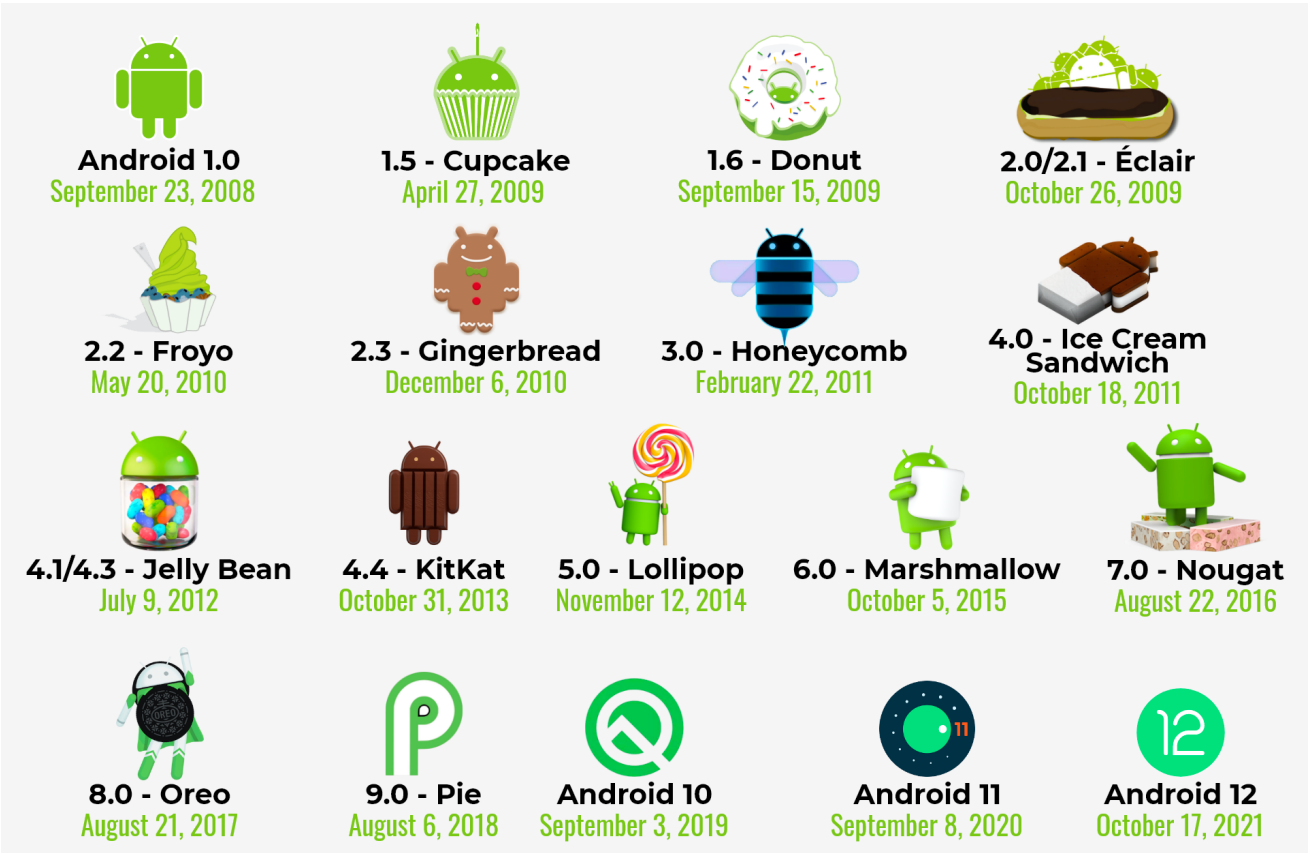
A mobile operating system that is based on a modified version of Linux.

Originally developed by a startup of the same name, Android, Inc, which was acquired by Google in 2005

Most of the Android code was released under the open-source Apache License

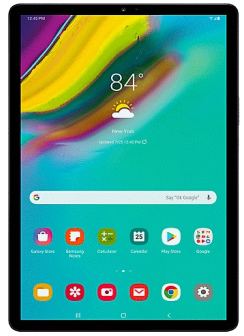
The main advantage to adopting Android is that it offers a unified approach to mobile application development

Android Releases



Android Devices in the Market

- Wear OS
- Android TV
- Android for Cars
- Android Things
- Chrome OS devices
- **Other portable devices** – Smartphones, Tablets, E-reader devices, Smartwatches, etc.

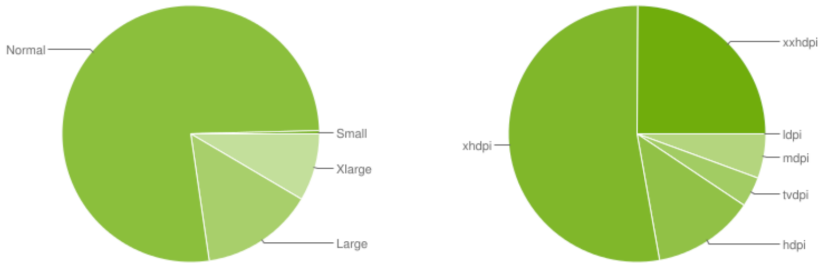


Platform Driving Android Things

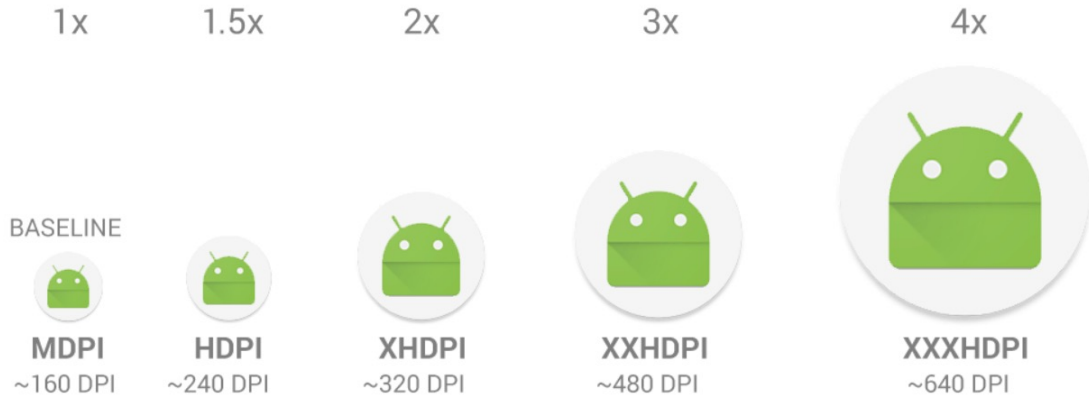
Screen Sizes and Densities

- Simplify the user interfaces design for different screen configurations
- Divides the range of actual screen sizes and densities into several buckets

	ldpi	mdpi	tvdpi	hdpi	xhdpi	xxhdpi	Total
Small					0.4%		0.4%
Normal		0.1%	0.3%	7.9%	45.4%	23.2%	76.9%
Large		1.2%	3.4%	1.1%	6.8%	1.7%	14.2%
Xlarge		4.3%	0.1%	3.8%	0.3%		8.5%
Total	0.0%	5.6%	3.8%	12.8%	52.9%	24.9%	



Data collected during a 7-day period ending on January 6, 2023.
Any screen configurations with less than 0.1% distribution are not shown.



Android Releases

Android 14 - Per-app language preferences for Internationalization

Android 13 – Build for user privacy with photo picker and notification permission

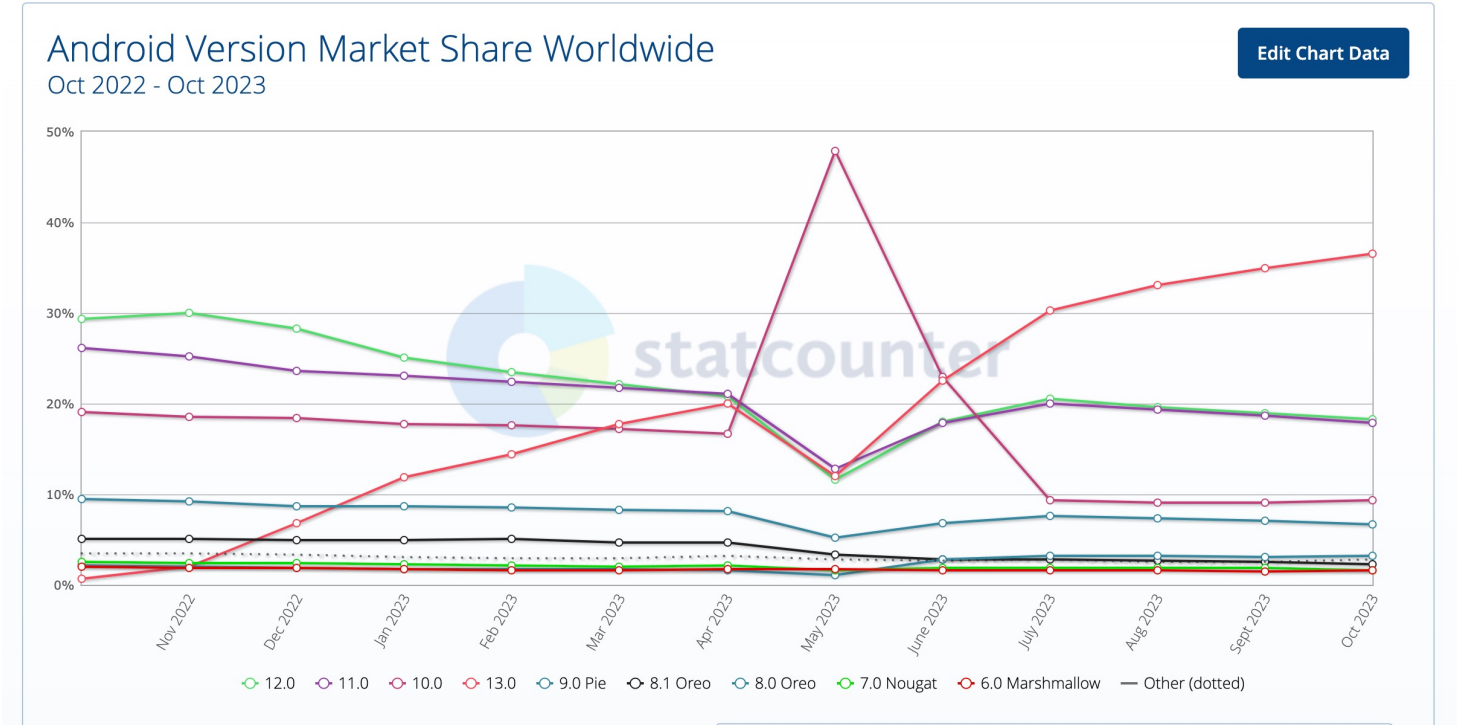
Android 12 – Material You

One important thing to keep in mind as you are looking at Android versions is that each version has its own features and APIs (application programming interfaces)

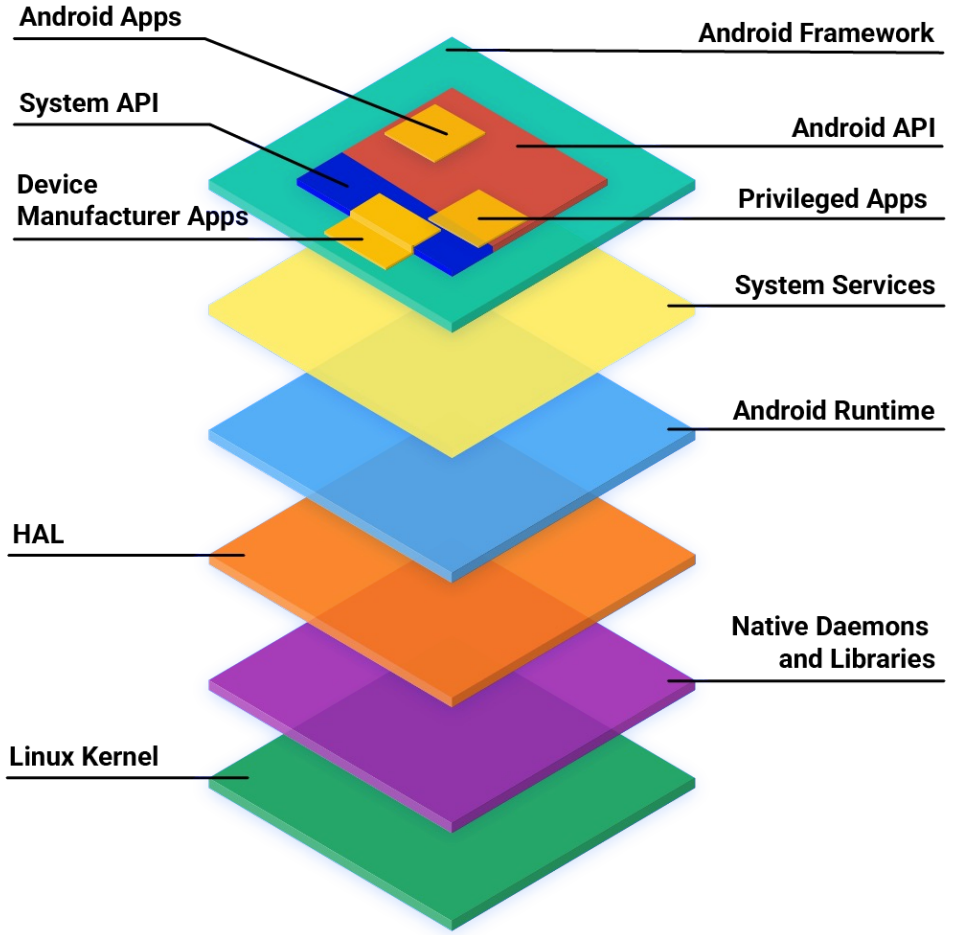
If your application is written for the newest version of Android, and it uses an API that was not present in an older version of Android, then only devices running that newer version of Android will be able to use your application

13.0	12.0	11.0	10.0	9.0 Pie	8.0 Oreo
36.45%	18.27%	17.82%	9.35%	6.64%	3.14%

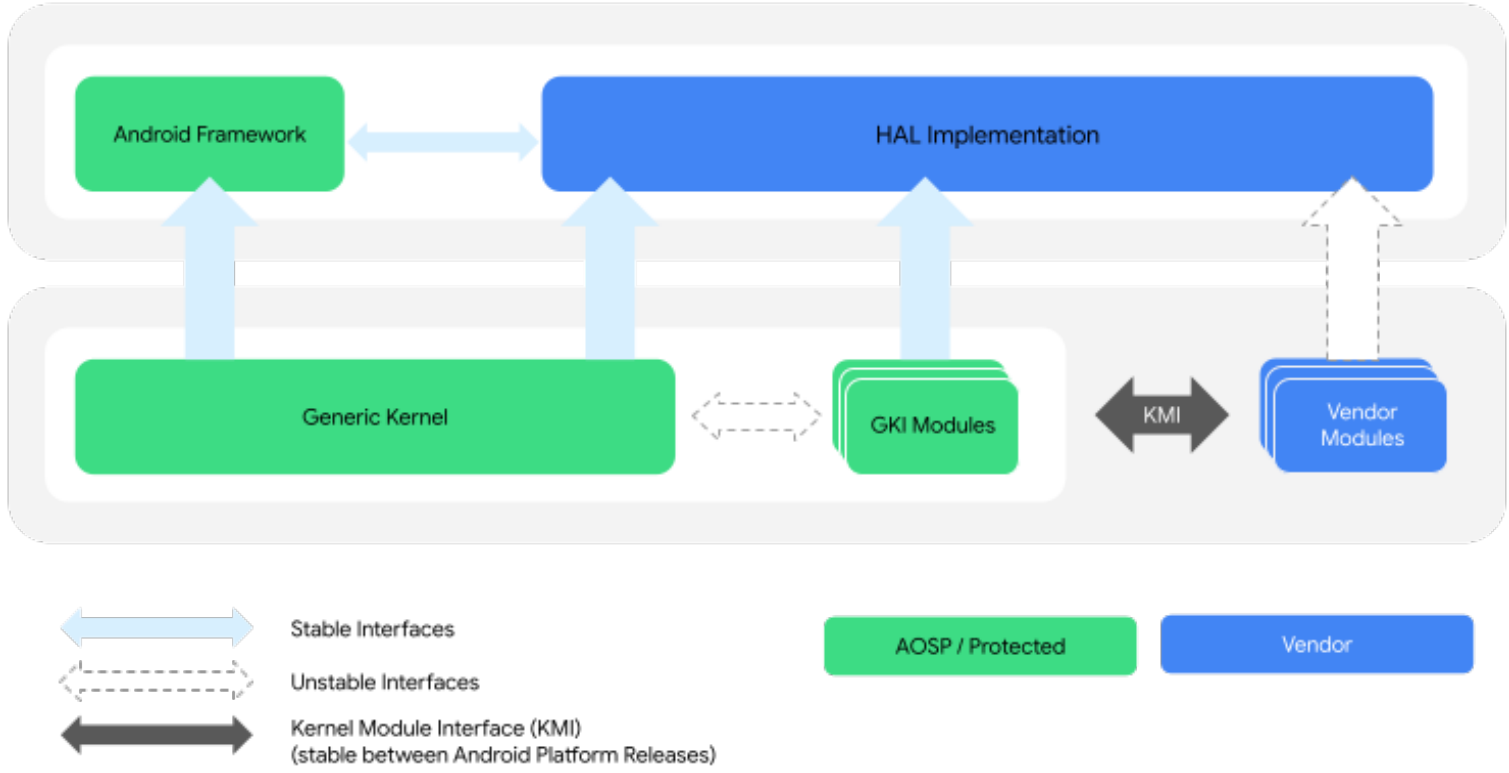
Android Version Market Share Worldwide - October 2023



Android Open-Source Project (AOSP) By Google



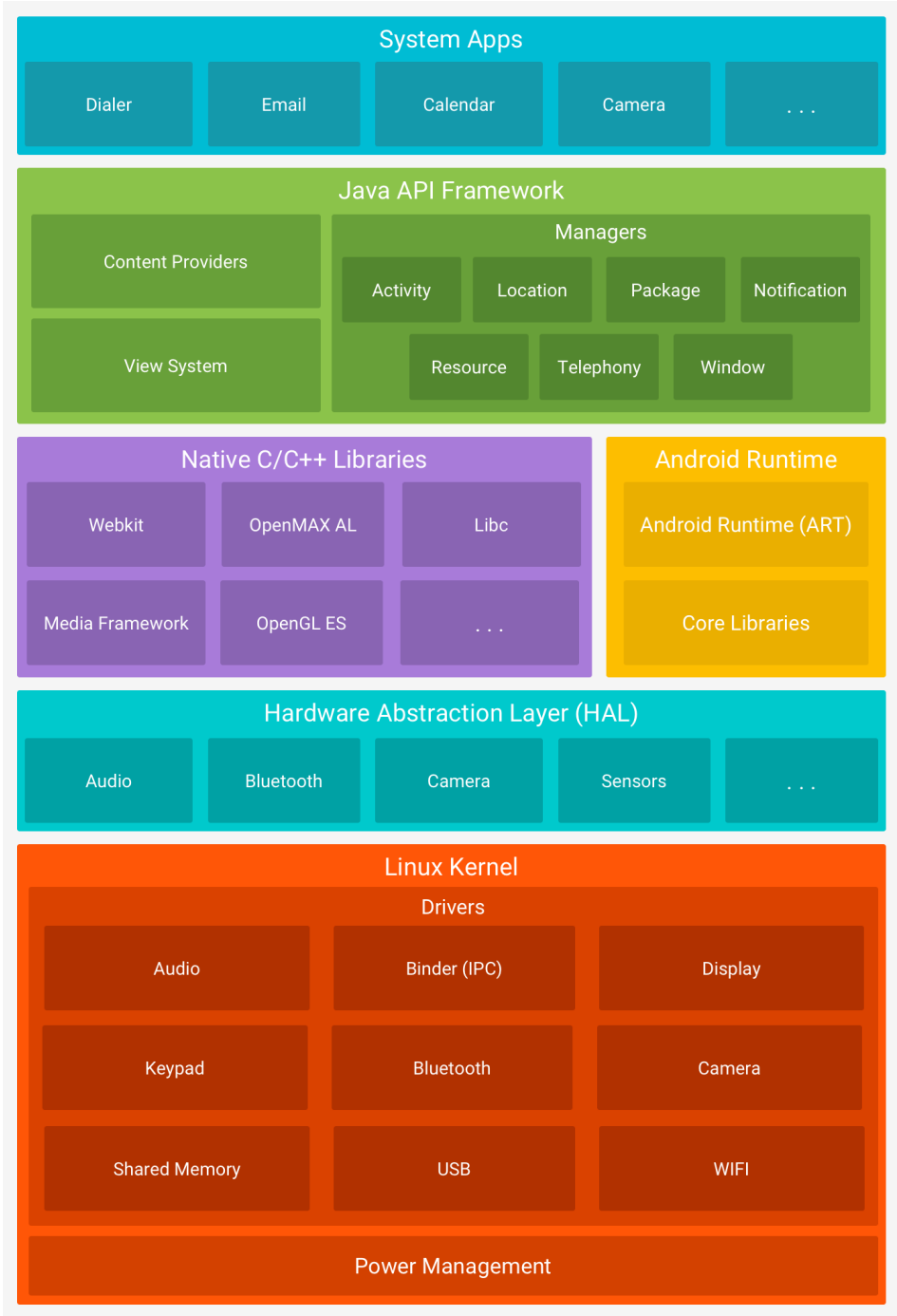
Android Common Kernels (ACKs)



Fundamental Elements

- System Architecture
- Connectivity
- Power Profiles
- Input and Sensors
- Custom Settings
- Font
- Media and Display
- System Health
- Data Storage and Access

Android Platform Architecture



App Architecture

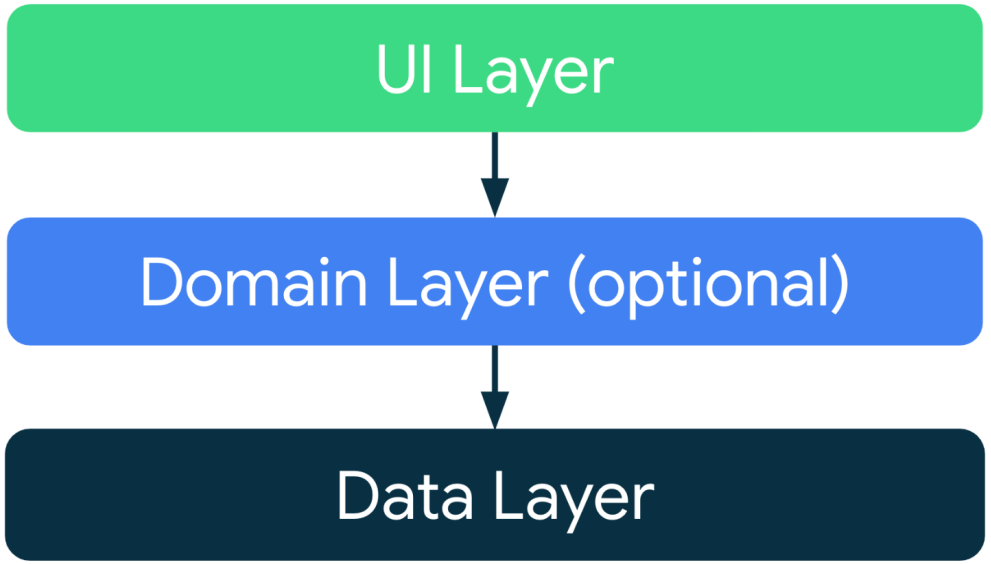
Mobile devices are resource-constrained – limited power and storage

Separation of concerns – scalability, drive UI from data model (persistent data)

UI Layer - displays application data on the screen

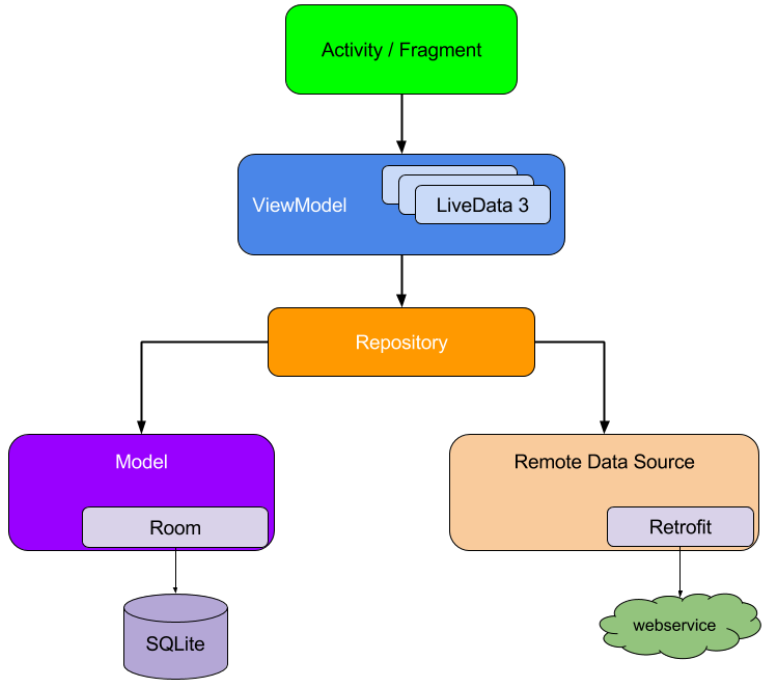
Domain Layer - simplify and reuse the interactions between the UI and data layers.

Data Layer - contains the business logic of your app and exposes application data



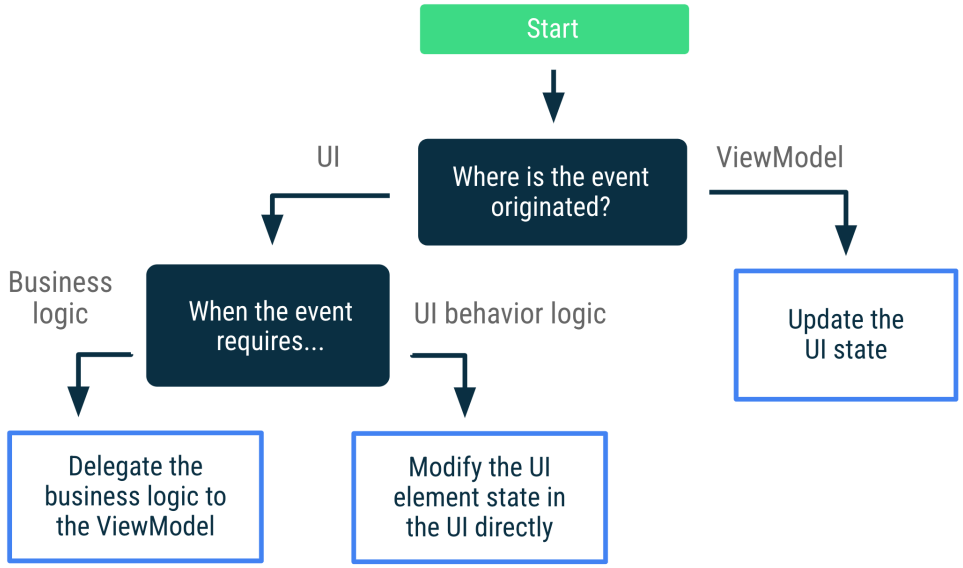
App Architecture

- Imagine you're building a UI that shows a user profile
- Typically, you use a private backend or a REST API to fetch the data for a given profile
- UI elements render the data on the screen (Activity/Fragment)
- ViewModel are state holders that hold data, expose it to the UI, and handle logic.
- Repository classes handle data sources.



UI Events

- ViewModel is responsible for handling the business logic of a particular user event—for example, the user clicking on a button to refresh some data
- Business logic refers to what to do with state changes—for example, making a payment or storing user preferences.
- UI behavior logic refers to how to display state changes



Android Studio – the IDE

For Android development, you can use a Mac, a Windows PC, or a Linux machine.

You can also use the iLab machines provided by the CS department: https://resources.cs.rutgers.edu/

You can freely download all the necessary tools

- Java JDK
- Android Studio: https://developer.android.com/studio ➔ IDE + Android SDK
- IntelliJ includes the Android Studio IDE
- **Android devices don't run .class and .jar files.** Instead, to improve speed and battery performance, Android devices use their own optimized formats for compiled code.

Resources for Android developers: https://developer.android.com/

Android app Development

- As a beginning Android programmer, there is a learning curve
- Android has a culture, which speaks Kotlin or Java (or a bit of both), but knowing Kotlin or Java is not enough
- Getting your head around Android requires learning many new ideas and techniques
- There are good software engineering practices involved!!

Kotlin vs. Java

- Official support for Kotlin for Android development was announced at Google I/O in 2017
- Since 2017, Kotlin has become widely adopted
- The Android framework team has released more and more Kotlin extensions for Android
- The Android framework was originally written in Java. This means most of the Android classes you interact with are Java
- Kotlin is interoperable with Java
- In May 2019, Google announced that the Kotlin programming language is now its preferred language for Android app developers

Deploy your Android Apps

Android apps are distributed via Google Play (https://play.google.com)

Google Play is a store for digital content, much like the Apple's app store, not just apps.

In order to distribute apps on Google Play, you need to become a registered developer

There are now over one million apps in Google Play, and the majority of those apps are free

One should also be aware that the Android operating system is open, so anyone can easily copy an app from one Android device to another; there is little protection for intellectual property

Creating Android Virtual Devices (AVDs)

Once you installed the Android Studio, the next step is to create an Android Virtual Device (AVD), which you use to test your Android applications

An AVD is an emulator instance that enables you to model an actual device

Each AVD consists of a hardware profile; a mapping to a system image; and emulated storage, such as a secure digital (SD) card

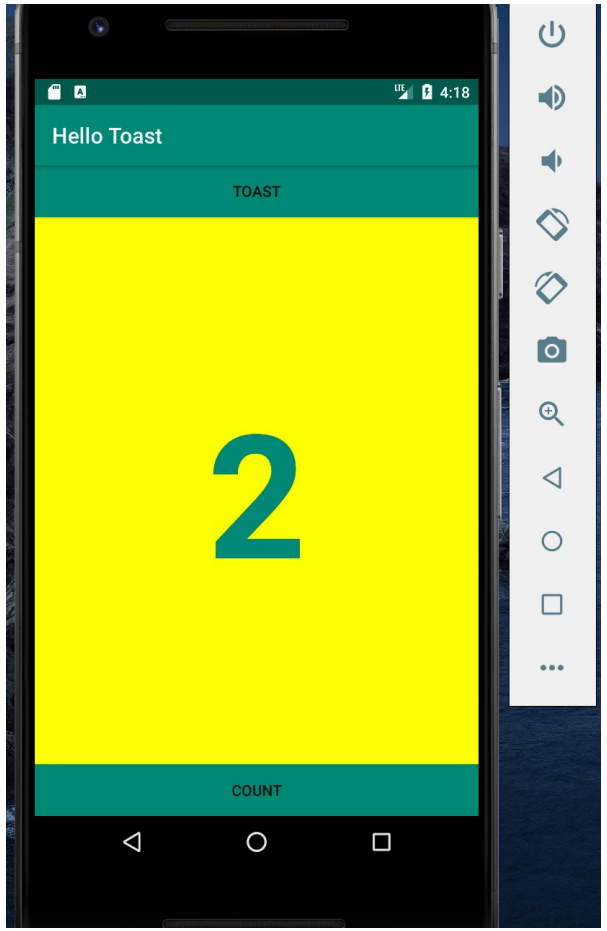
The emulator is good for doing some generalized testing of your applications

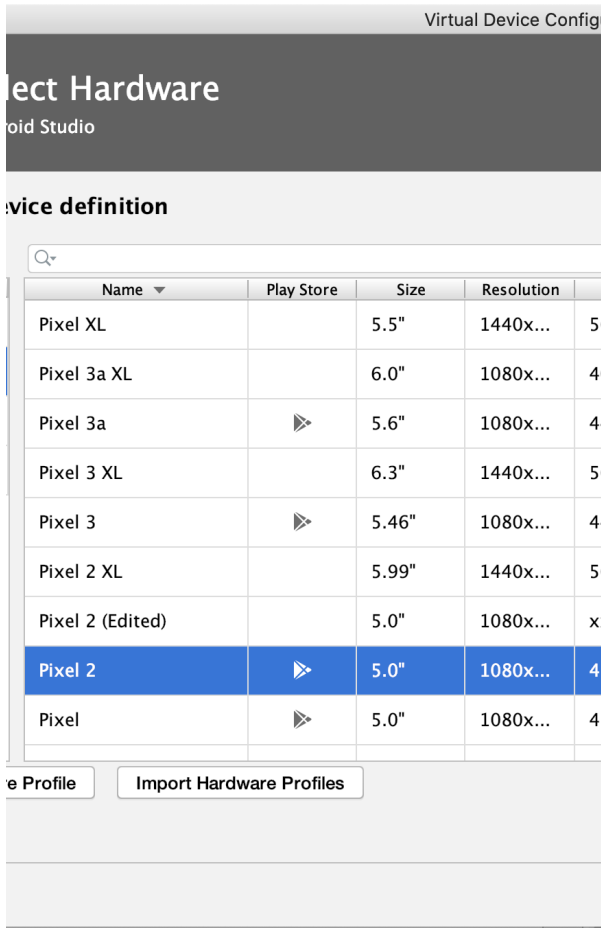
Games (GPU heavy) or applications using sensors (GPS) cannot be simulated with the same speed or consistency

You can create as many AVDs as you want to test your applications with different configurations

Android Device Manager

- When you create your first Android project, you will be asked to create an AVD, OR
- Select Tools/Device Manager to add other emulators



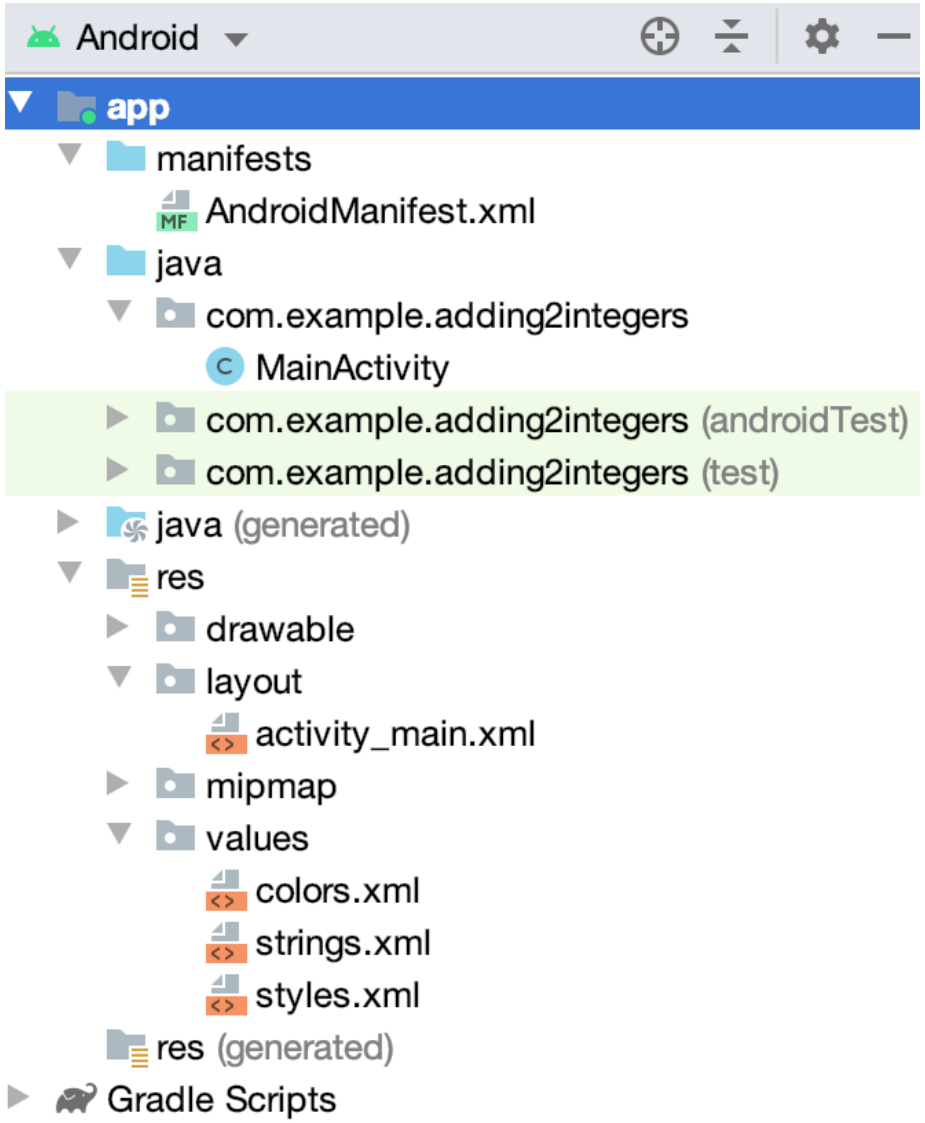


Run your app on a real device

- If you are using an Android smart phone, you can use it as an emulator to run your app
- Connect your device to your development machine with a USB cable
- Follow the steps on this webpage:
- https://developer.android.com/studio/run/device

Android Project Structure

- manifests
- source folder - MainActivity.java (the controller)
- res folder
  - layout folder - activity_main.xml (the view)
  - drawable folder – images
  - mipmap folder – app icons
  - values folder – constants
- Gradle Scripts – app configuration and dependency management
  - build.gradle (module)



Android SDK and Project Structure

- Make sure you select the right JDK version
  - Preference/Build Tools/Gradle → Gradle JDK version (e.g., 17)
- Make sure you install the Android SDK needed to run your app
  - Tools/SDK Manager → SDK Platform and SDK Tools
  - File/Project Structure
    - Gradle version
    - Android SDK location
    - SDK Compile and Build version
    - Dependencies
  - build.gradle (Module) → e.g., compileSdk 33