

Android Overview



2



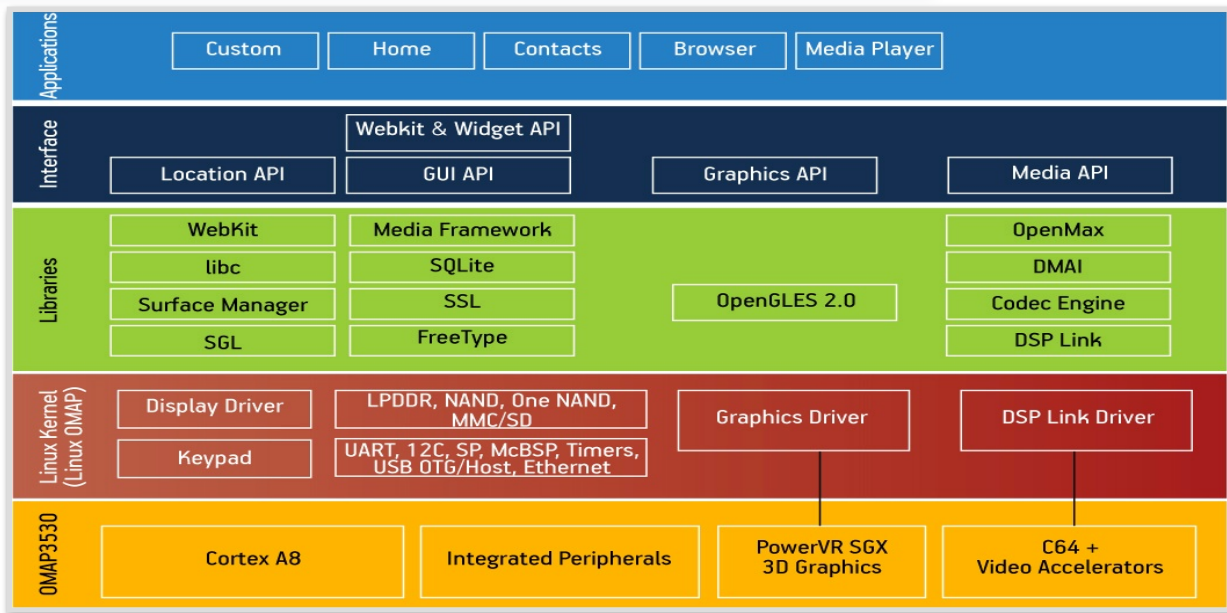
The platform structure and
components

Agenda

- The Platform
- Setting up the Environment
- Android Apps vs iPhone Apps

Comprehensive

- Android is a complete software stack for a mobile device..

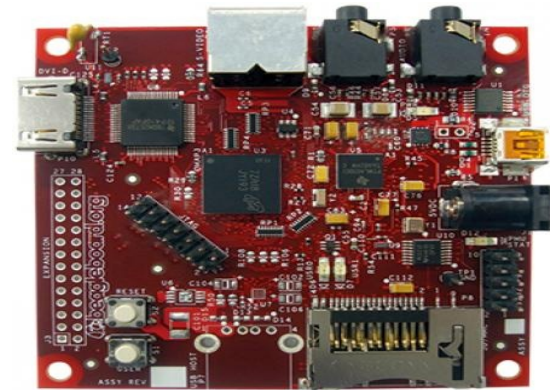


- Users can customize their phone experience substantially.
- Manufacturers can also customise the platform in substantial ways - even generating complete 'forks' of the original project (Amazon).

The Platform- Open Source

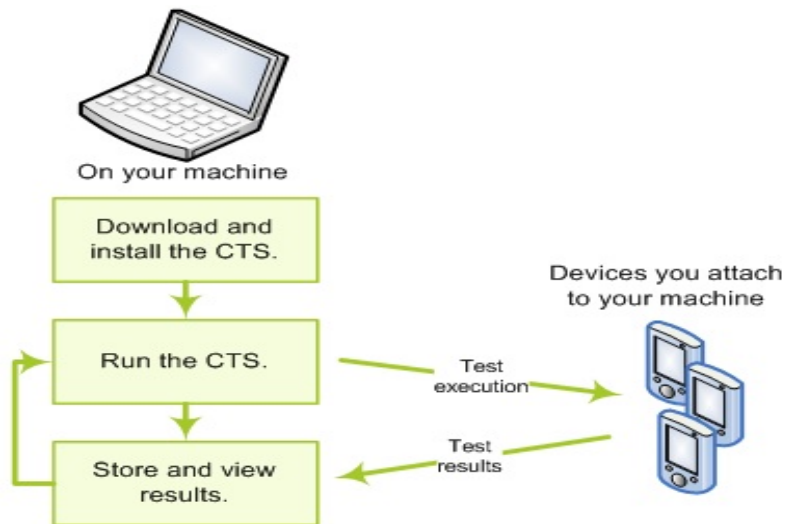
- Android is an open source platform.
- Aside from the Linux kernel itself, Android is licensed under business-friendly licenses (Apache/MIT/BSD) so that others can freely extend it and use it for variety of purposes.
- Manufacturers can port Android OS to specific hardware. with minimal legal issues.
- Android has many hooks at various levels of the platform, allowing anyone to extend it in unforeseen ways.

BeagleBoard, a low-power open-source hardware single-board computer produced by Texas Instruments



CTS

- The Compatibility Test Suite (CTS), defines what it means to be an Android-compatible device.
- CTS is a combination of automated tests as well as a document that specifies what an Android device must have, should have, or what features are simply optional.
- The goal of CTS is to ensure that, for a regular consumer, an average app from the market will run on an average Android device if that device claims to be supporting a certain version of Android.



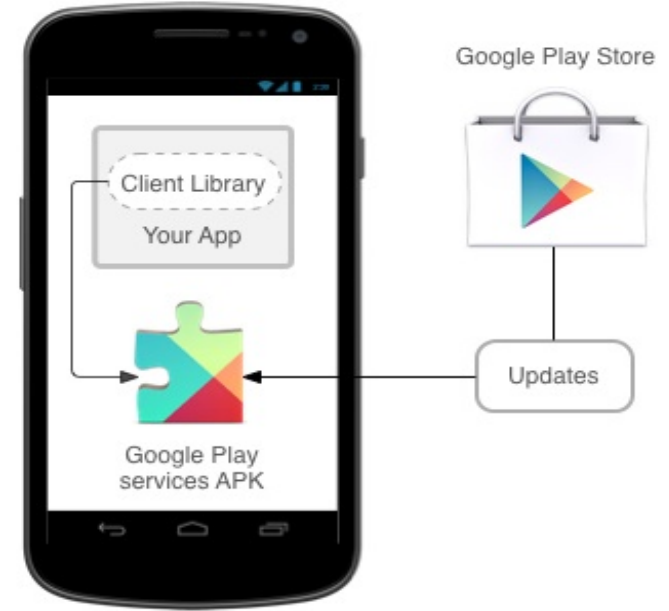
Compatibility is optional

- CTS is being completely avoided by Amazon with the Kindle Fire and phone series of devices, built on top of the Android OS.
- Anyone is welcome to download and “remix” Android in any way they see fit.
- Android has been customized for everything from cars to satellites, and from photocopiers to washing machines.

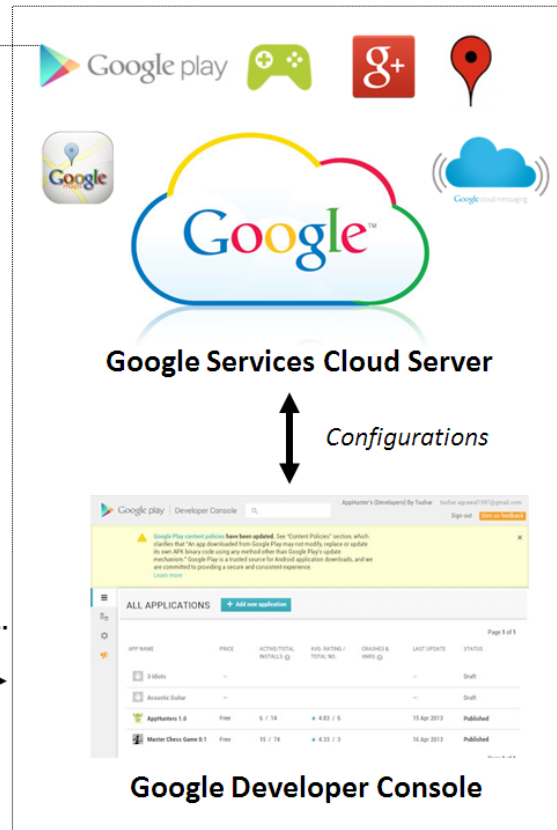
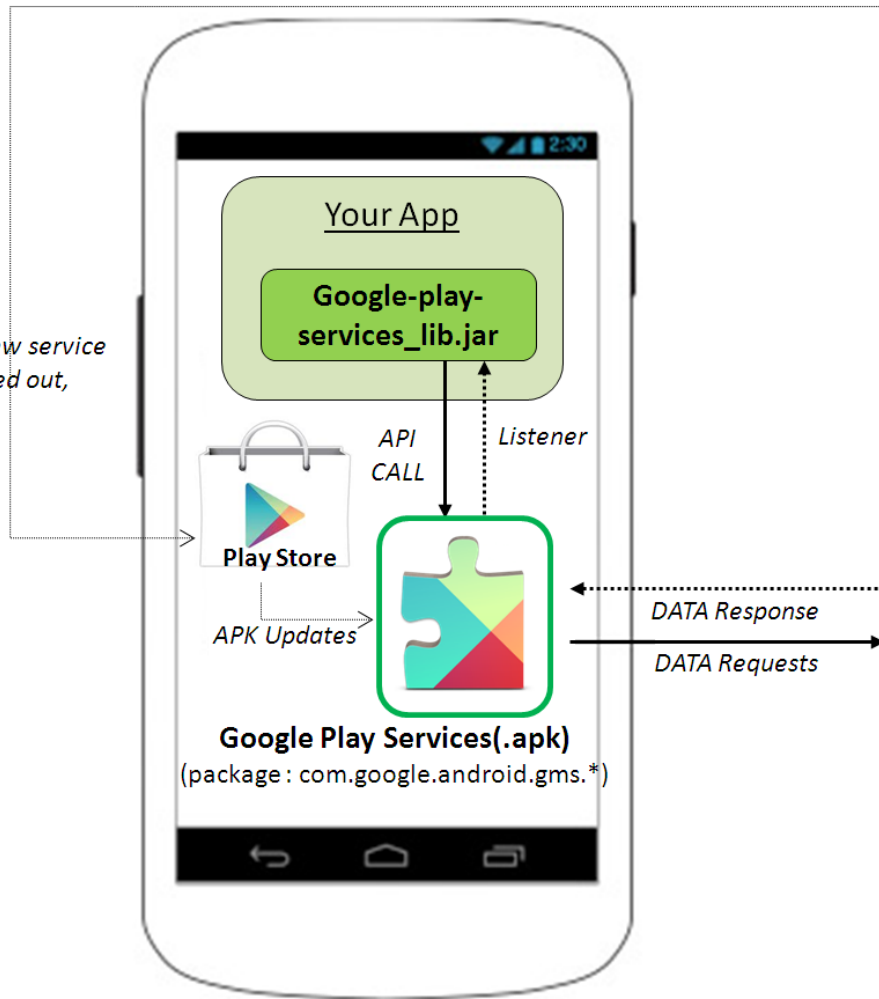
Area	Description
Signature tests	For each Android release, there are XML files describing the signatures of all public APIs contained in the release. The CTS contains a utility to check those API signatures against the APIs available on the device. The results from signature checking are recorded in the test result XML file.
Platform API Tests	Test the platform (core libraries and Android Application Framework) APIs as documented in the SDK Class Index to ensure API correctness, including correct class, attribute and method signatures, correct method behavior, and negative tests to ensure expected behavior for incorrect parameter handling.
Dalvik VM Tests	The tests focus on testing the Dalvik VM.
Platform Data Model	The CTS tests the core platform data model as exposed to application developers through content providers, as documented in the SDK android.provider package: contacts, browser, settings, etc.
Platform Intents	The CTS tests the core platform intents, as documented in the SDK Available Intents .
Platform Permissions	The CTS tests the core platform permissions, as documented in the SDK Available Permissions .
Platform Resources	The CTS tests for correct handling of the core platform resource types, as documented in the SDK Available Resource Types . This includes tests for: simple values, drawables, nine-patch, animations, layouts, styles and themes, and loading alternate resources.

Google Play Services

- The major reason manufacturers would want to ensure Android compatibility is access to Google Play Services, and its set of apps.
- It allow apps to take advantage of the latest, Google-powered features such as Maps, Places, Google+, and more, with automatic platform updates distributed as an APK through the Google Play store.
- Makes it faster for phone to receive updates and easier for developers to integrate the some new features into their apps.



When new service
is rolled out,



Android Developers 'Toolbox'

- Standard :
 - User Interface
 - IO widgets (buttons, textboxes, lists etc.)
 - Images
 - 2D/3D drawing
 - ActionBar/ActionMode
 - ViewPager
 - NavDrawer
 - Database (Sqlite/Rooms)

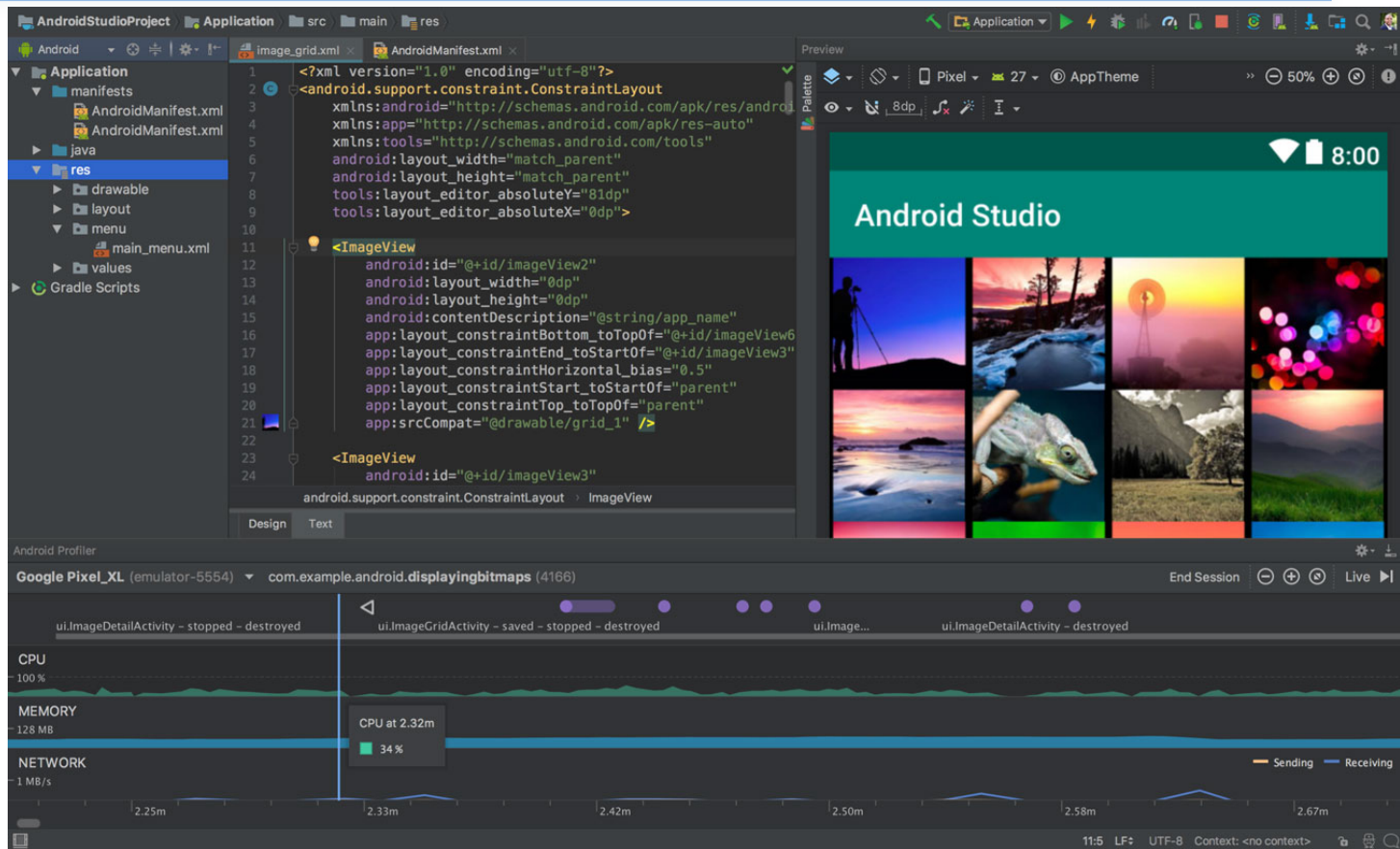
Android Developers 'Toolbox'

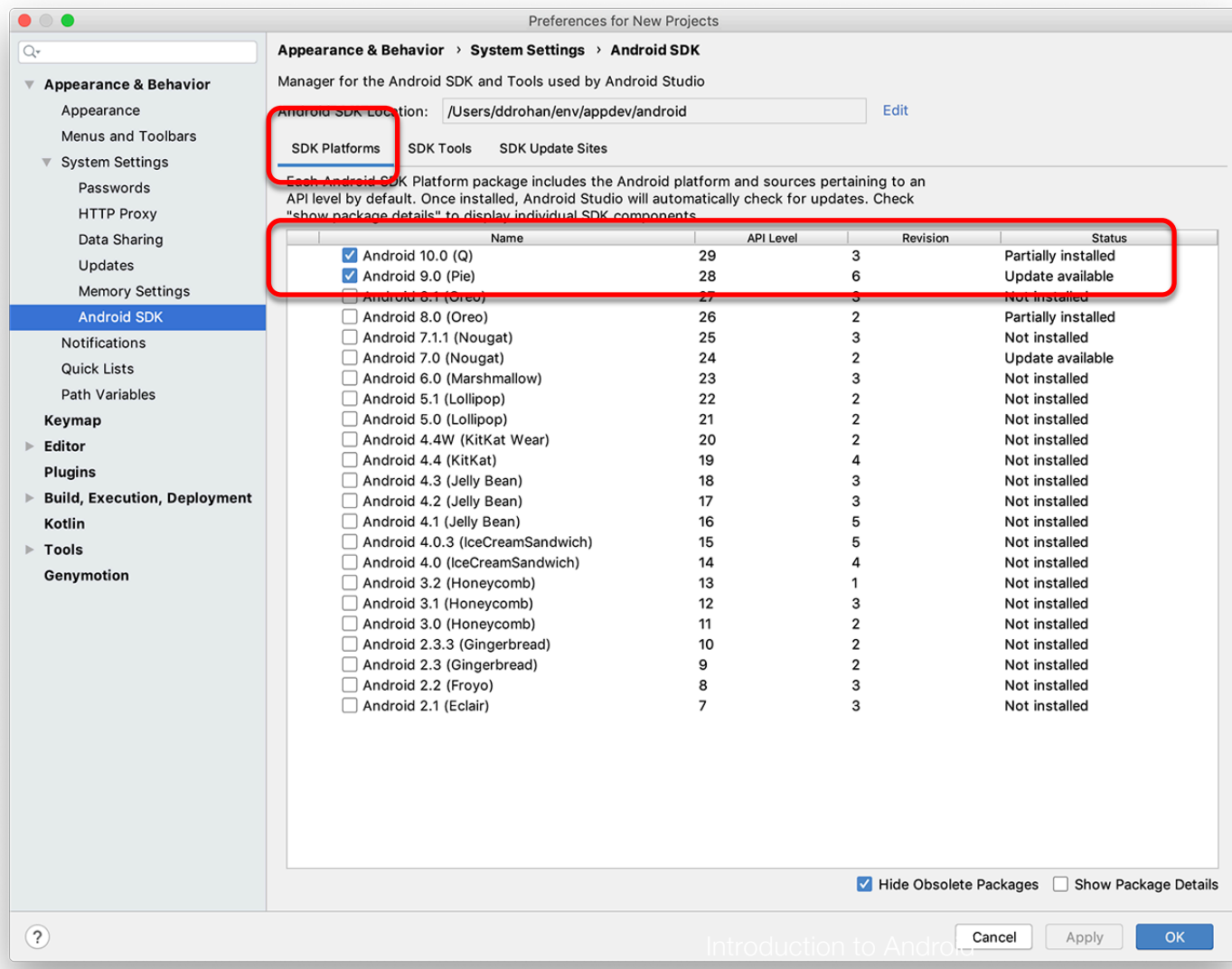
- Advanced :
 - Google Maps
 - Google Sign-In
 - FireBase
 - Hardware APIs
 - GPS/Geo-location, calls, accelerometer, compass, bluetooth, camera
 - Fused Location Provider (latest addition)
 - Multiple processes
 - Managed by ART (as of Kit Kat 4.4+)
 - Background Services (e.g. AsyncTasks/Volley)
 - Inter-process communications (e.g. Intents)
 - No difference between third-party and native apps (2 lines of code to launch Camera on Device)

Android Applications (re : the installation stuff)

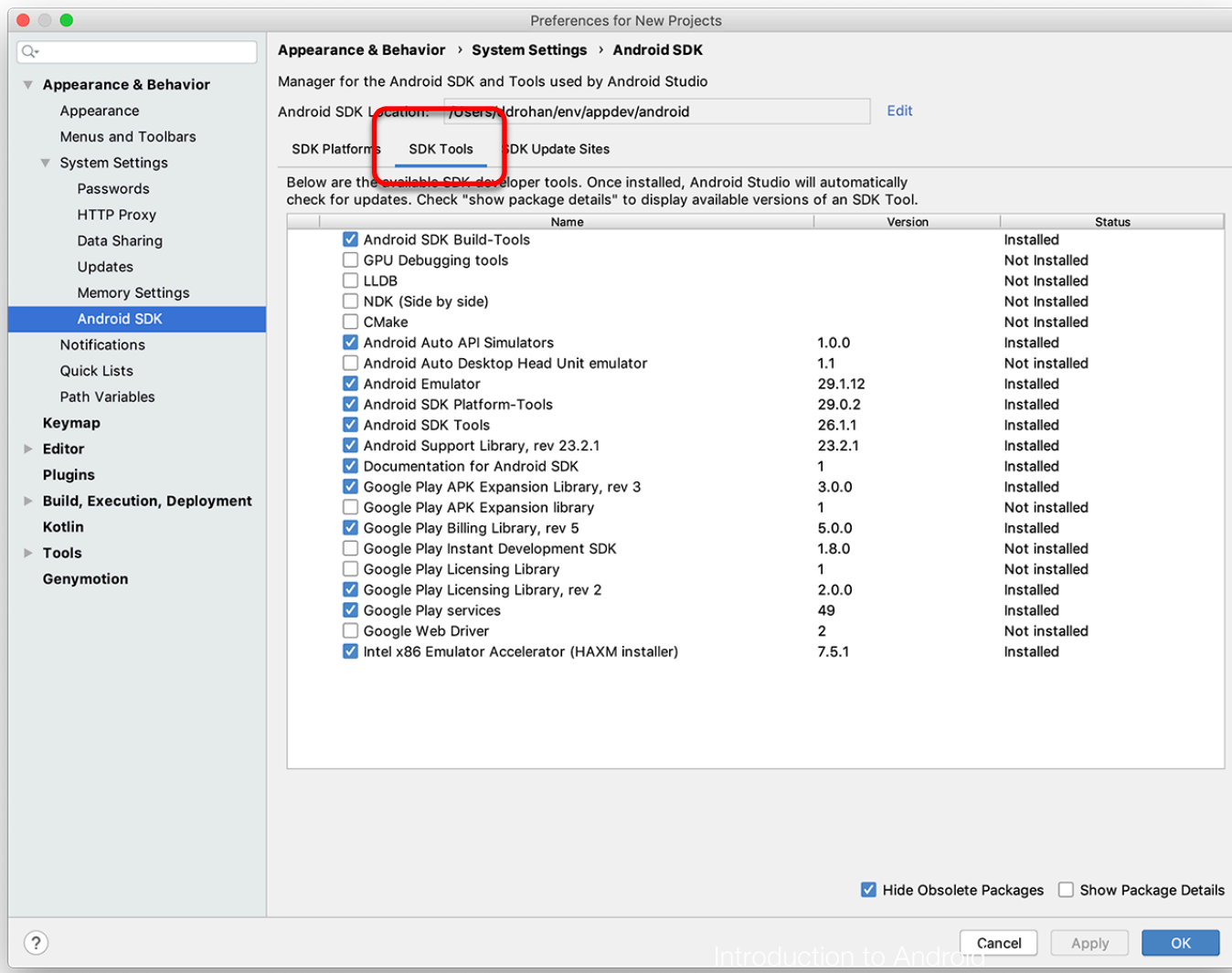
- Android applications get distributed in a .apk file
- APK == “Android Package”
 - It is simply an archive/zip file that has a particular file structure (similar to JAR files that take snapshots of the file system)
 - An APK contains
 - The **Android Manifest** file (an XML file with lots of metadata)
 - A **Resource bundle** containing sounds, graphics, etc.
 - The **ART bytecodes** that make up your application

Android Studio





Android SDK
Setup for the
Labs *



Android SDK
Setup for the
Labs *



Android Apps vs iPhone Apps

Installing Apps

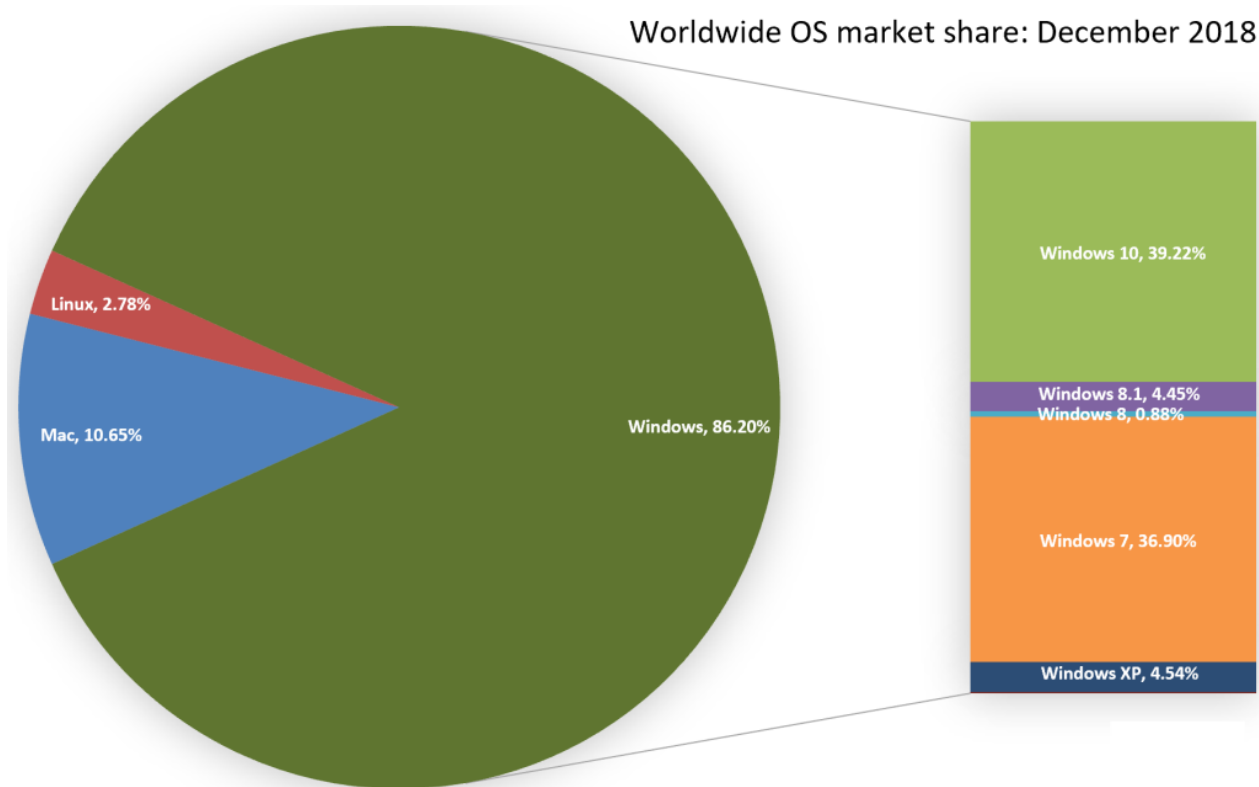
- For Generic (on the market) apps
 - iPhone has larger selection
 - Android catching up
- For In-house-developed corporate apps
 - iPhone apps can only (mostly) be installed via the App Store
 - iPhone requires you to submit app to the Apple App Store and get approval, even for apps from your own company
 - Unless you setup a Provisioning profile or
 - you use something like TestFlight or
 - jailbreak your phone of course.....
 - Android apps can be installed through
 - Google Play
 - Amazon App Store
 - USB connection from PC
 - Email
 - Corporate Web site

Languages for Apps

- iPhone
 - Objective-C
 - Similar to, but not exactly the same as, C++
 - Virtually no corporate presence for Objective-C, other than for mobile apps
 - However, **Swift** becoming ever more popular
- Android
 - Java
 - The single most widely used language inside corporations
 - C/C++
 - Can call native apps (with some difficulty) via an approach similar to JNI for desktop Java
 - Kotlin gaining a lot of traction now with 1st Class Support from Google

OS for Dev Apps

- iPhone
 - Macs
- Android
 - Anything with Java and Android Studio
 - Macs
 - PCs
 - Linux



Bottom Line: Android vs iPhone

- Which to use personally
 - iPhone has bigger app store (only slightly!), and more loyal users
 - Android more open and growing more rapidly
 - Bottom line: no clear winner, personal preferences prevail, but iPhone has a technical edge
- Which to use for in-house apps
 - iPhone apps very hard/restrictive to install, Android very simple
 - iPhone uses mainly Objective-C (Swift on the up), Android uses Java / Kotlin