Android App Development

Saurabh Kumar

PhD Student, IIT Kanpur

Outline

- **Introduction to Android**
- Getting Started
- Android Programming

Introduction to Android

- Popular mobile device OS: 85% of global smartphone market
- Developed by Open Handset Alliance, led by Google

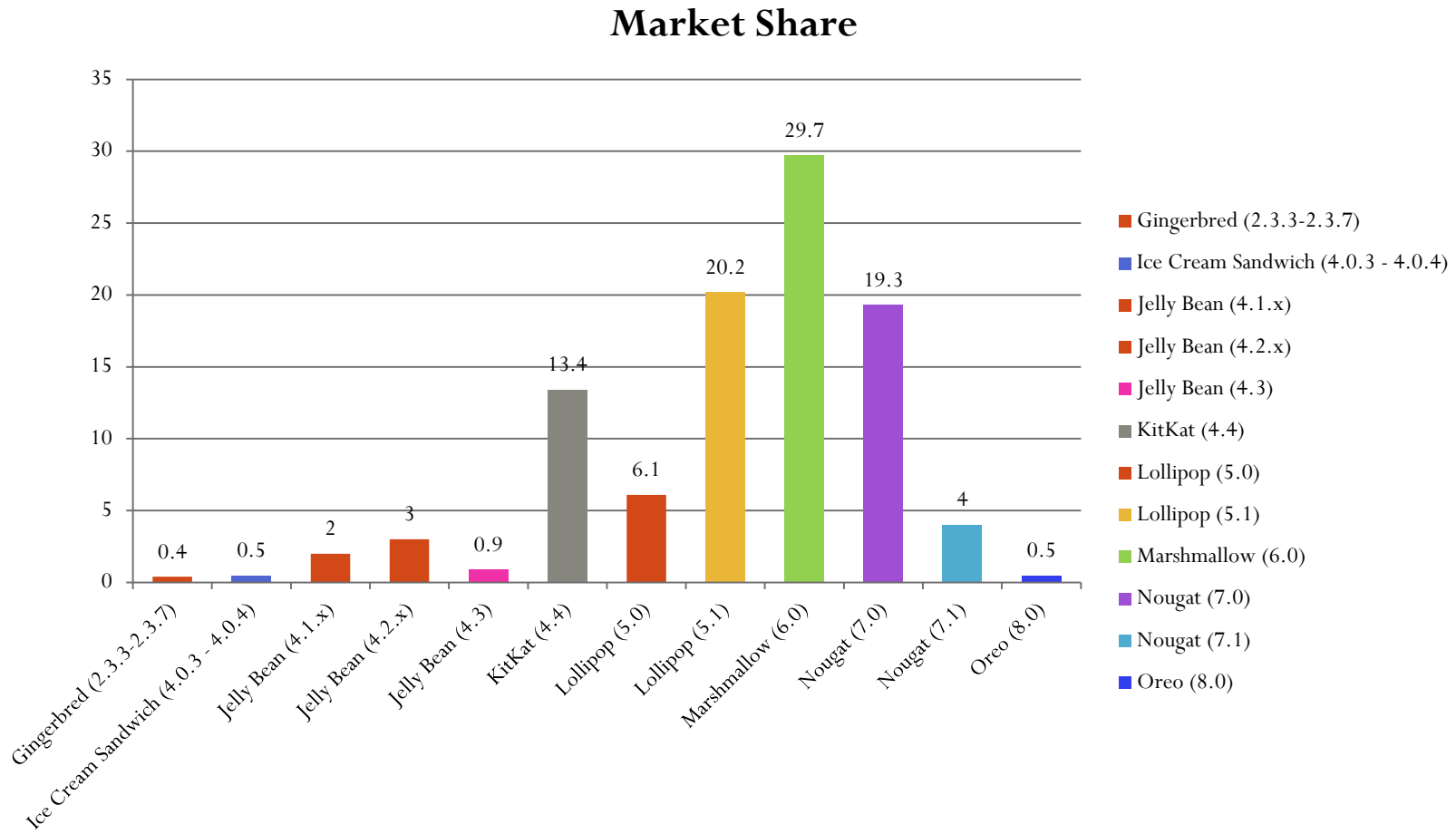
Period	Android	iOS	Windows	Others
Q1 2016	83.4%	15.4%	0.8%	0.4%
Q2 2016	87.6%	11.7%	0.4%	0.3%
Q3 2016	86.6%	12.5%	0.3%	0.4%
Q4 2016	81.4%	18.2%	0.2%	0.2%
Q1 2017	85%	14.7%	0.1%	0.1%

Source: IDC, May 2017 [1]

Introduction to Android

- Android is an operating system for mobile devices such as *smartphones* and *tablets* computers. It is developed by the Open Handset Alliance led by Google.
- Android has beaten Apple iOS, being the leading mobile operating system from first quarter of 2011.

Android Versions Market Share



Source: Android developer [2]

Actors in the Android Ecosystem

Tool chain
(Coredova, App generator, ...)

Use Tools

App Developer

Publish app

Publish app

Google Play

Sideloading

Alternate markets

Online services

Configure

Administrators

Advertisement
networks

Third Party app

Ad Libs

Application Framework

Native libs
(C / C++)

Android Runtime
(Dalvik / ART)

Linux Kernel (modified)

Platform vendors

Android Software Stack

Default apps

Contacts | SMS

Third party apps

paytm | linkedin

Application Framework

Native libs (C / C++)

Android Runtime (Dalvik / ART)

Linux Kernel (modified)

Outline

- Introduction to Android
- **Getting Started**
- Android Programming

Getting Started (1)

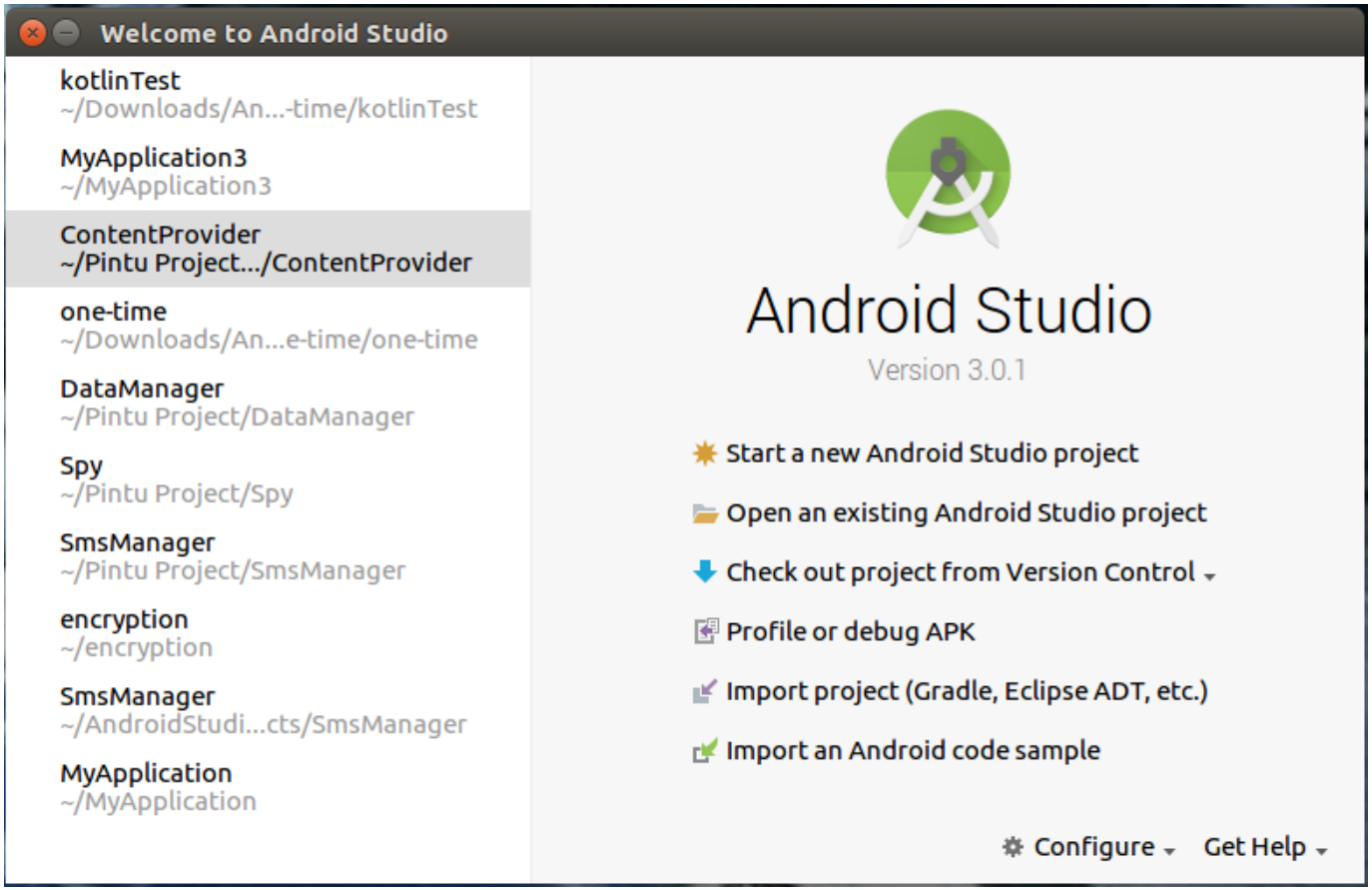
- Need to install Java Development Kit (JDK) to write Java (and Android) programs
  - **Do not** install Java Runtime Environment (JRE); JDK and JRE are different!
- Can download the JDK for your OS at http://www.oracle.com/technetwork/java/index.html
- Alternatively, for OS X, Linux:
  - OS X:
    - Open `/Applications/Utilities/Terminal.app`
    - Type `javac` at command line
    - Install Java when prompt appears
  - Linux:
    - Type `sudo apt-get install default-jdk` at command line (Debian, Ubuntu)
    - Other distributions: consult distribution's documentation

Getting Started (2)

- After installing JDK, download Android SDK from https://developer.android.com/studio/index.html
- Simplest: download and install Android Studio bundle (including Android SDK) for your OS

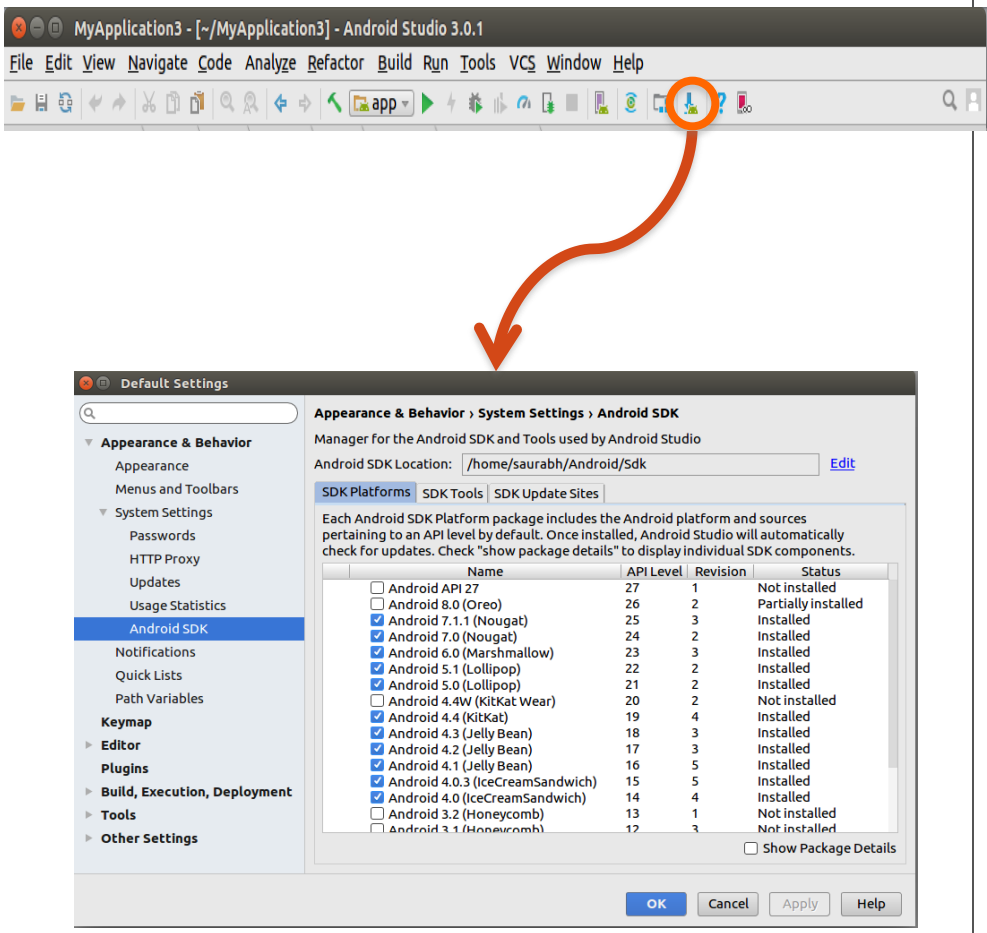
Getting Started (3)

- Install Android Studio directly (Windows, Mac); unzip to directory `android-studio`, then run `./android-studio/bin/studio.sh` (Linux)



Getting Started (4)

- Strongly recommend testing with real Android device
  - Android emulator: *very* slow
  - Faster emulator: Genymotion [7], [8]
  - Install USB drivers for your Android device!
- Bring up the Android SDK Manager



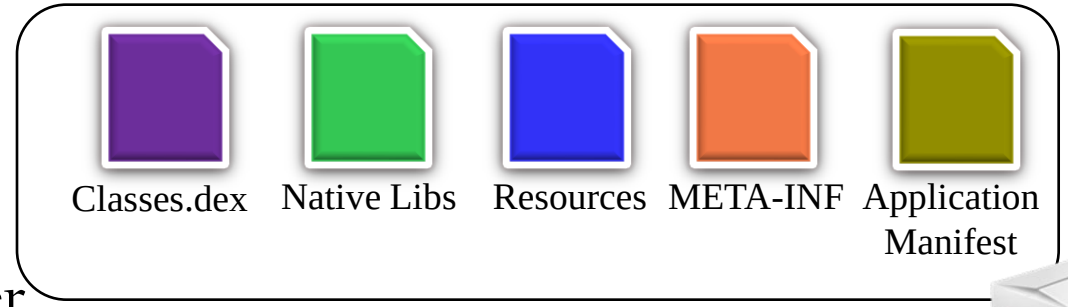
Now you're ready for Android development!

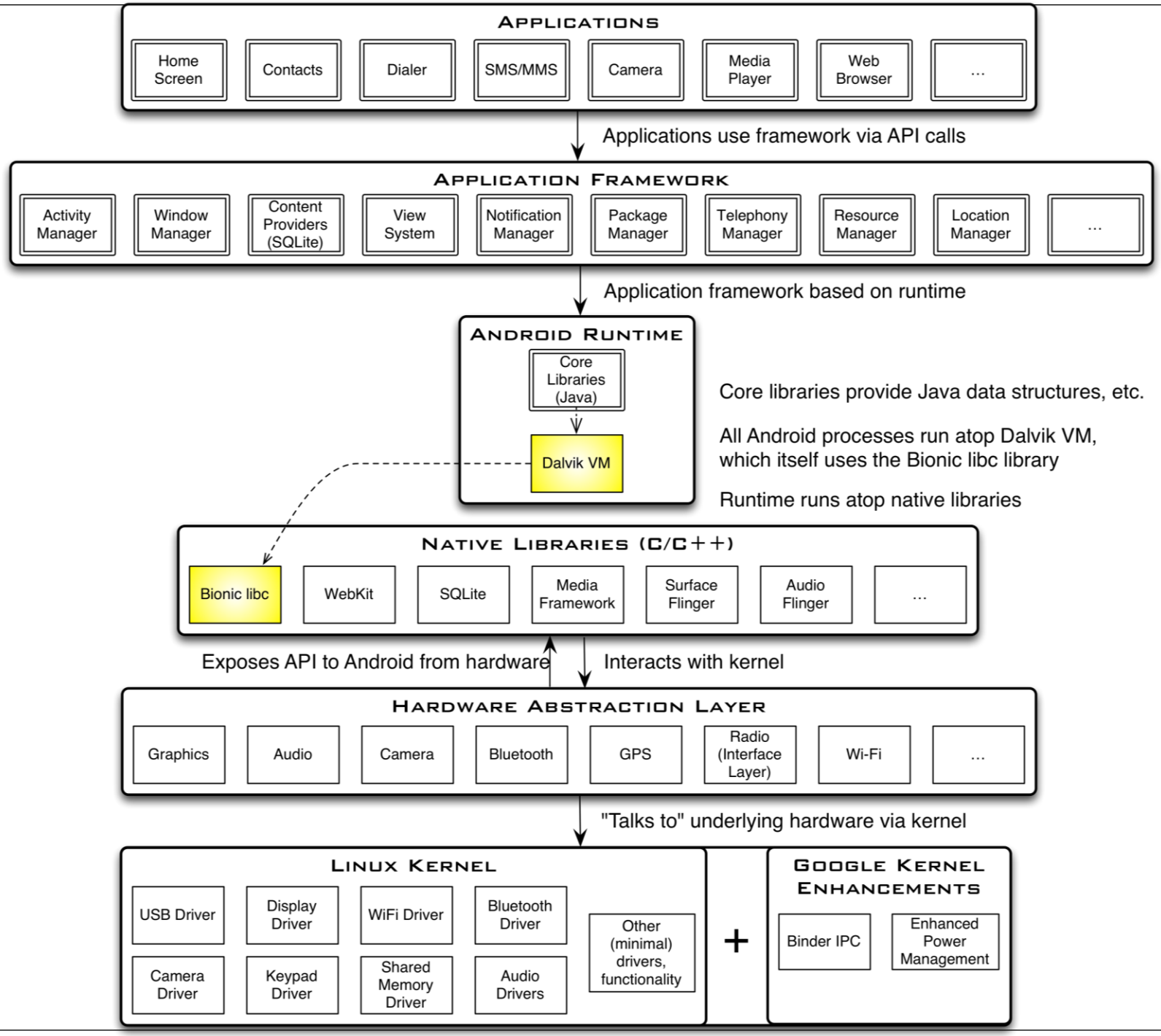
Outline

- Introduction to Android
- Getting Started
- **Android Programming**

Application Packages (APK)

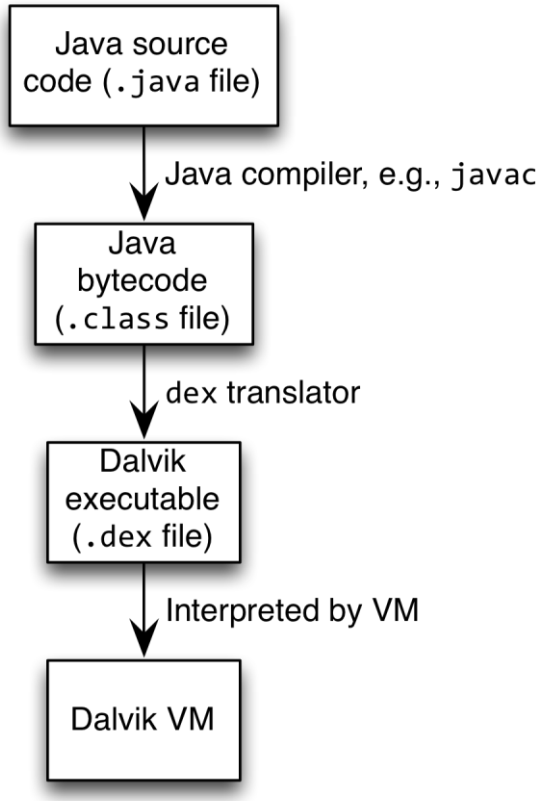
- APK is simply a packaging format like **JAR**, ZIP and TAR
- Component of Application
  - Activity
  - Content Provider
  - Services
  - Broadcast Receiver
- Native Code (C/C++ shared libraries)
- Resources
- META-INF
- Application Manifest




Applications
Home Screen
Contacts
Dialer
SMS/MMS
Camera
Media Player
Web Browser
...
Applications use framework via API calls
Application Framework
Activity Manager
Window Manager
Content Providers (SQLite)
View System
Notification Manager
Package Manager
Telephony Manager
Resource Manager
Location Manager
...
Application framework based on runtime
Android Runtime
Core Libraries (Java)
Dalvik VM
Core libraries provide Java data structures, etc.
All Android processes run atop Dalvik VM, which itself uses the Bionic libc library
Runtime runs atop native libraries
Native Libraries (C/C++)
Bionic libc
WebKit
SQLite
Media Framework
Surface Flinger
Audio Flinger
...
Exposes API to Android from hardware
Interacts with kernel
Hardware Abstraction Layer
Graphics
Audio
Camera
Bluetooth
GPS
Radio (Interface Layer)
Wi-Fi
...
"Talks to" underlying hardware via kernel
Linux Kernel
USB Driver
Display Driver
WiFi Driver
Bluetooth Driver
Camera Driver
Keypad Driver
Shared Memory Driver
Audio Drivers
Other (minimal) drivers, functionality
+
Google Kernel Enhancements
Binder IPC
Enhanced Power Management

Android Highlights (1)

- Android apps execute on Dalvik VM, a "clean-room" implementation of JVM
  - Dalvik optimized for efficient execution
  - Dalvik: register-based VM, unlike Oracle's stack-based JVM
  - Java .class bytecode translated to Dalvik EXecutable (DEX) bytecode, which Dalvik interprets



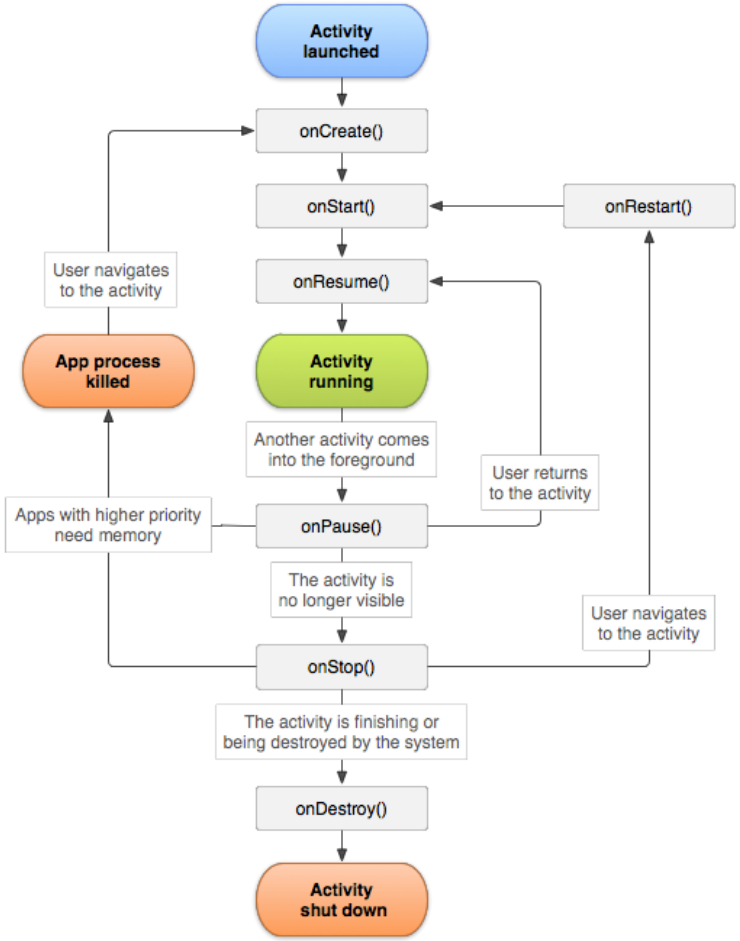
Code Flow from source to execution

App Manifest

- Every Android app must include an `AndroidManifest.xml` file describing functionality
- The manifest specifies:
  - App's Activities, Services, etc.
  - Permissions requested by app
  - Minimum API required
  - Hardware features required, e.g., camera with autofocus
  - External libraries to which app is linked, e.g., Google Maps library

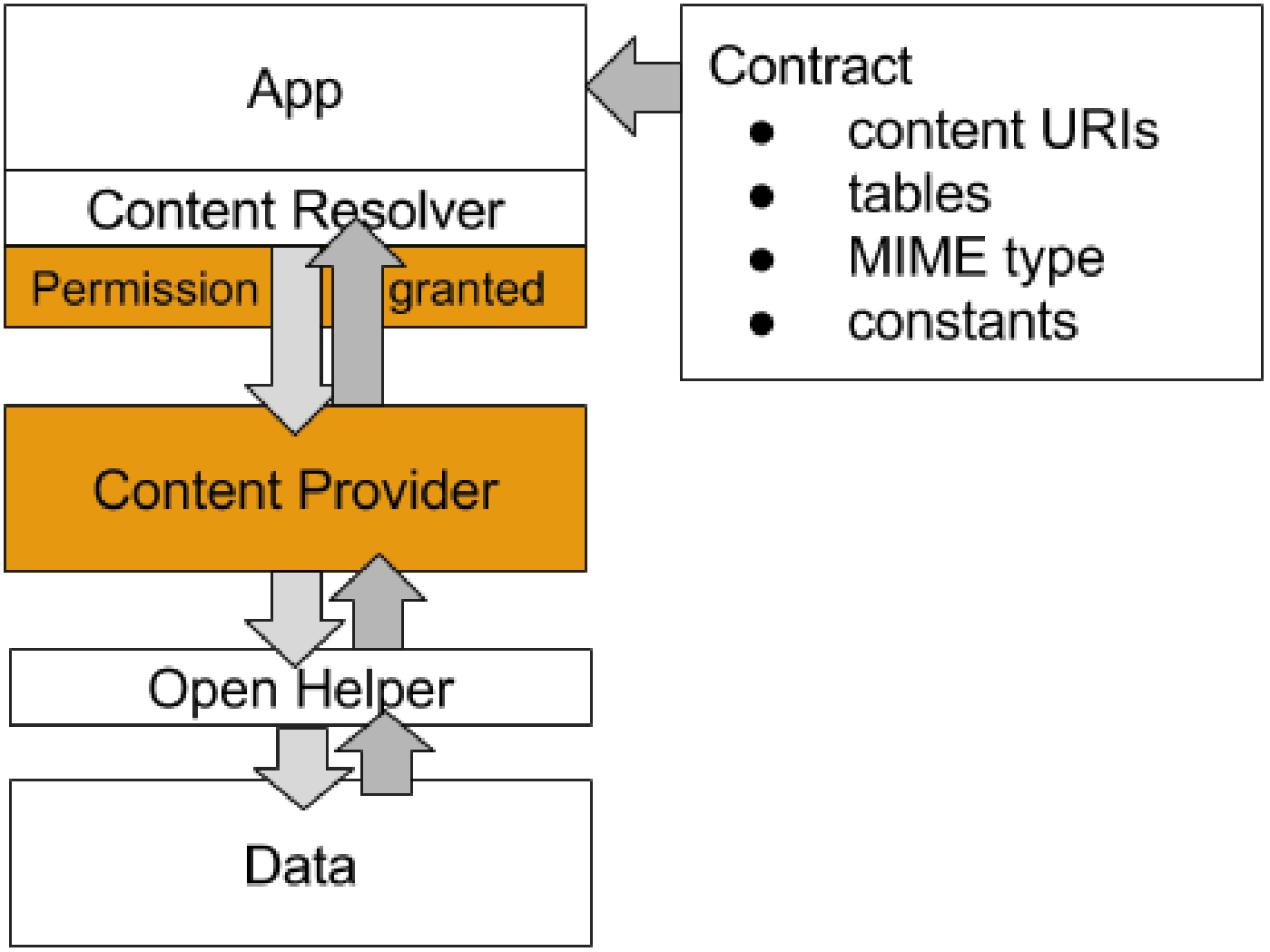
Activity Lifecycle

- `Activity`: key building block of Android apps
- Extend `Activity` class, override `onCreate()`, `onPause()`, `onResume()` methods
- Dalvik VM can stop any `Activity` without warning, so saving state is important!
- Activities need to be "responsive", otherwise Android shows user "App Not Responsive" warning:
  - Place lengthy operations in `Runnable` `Threads`, `AsyncTasks`!



Source: [5]

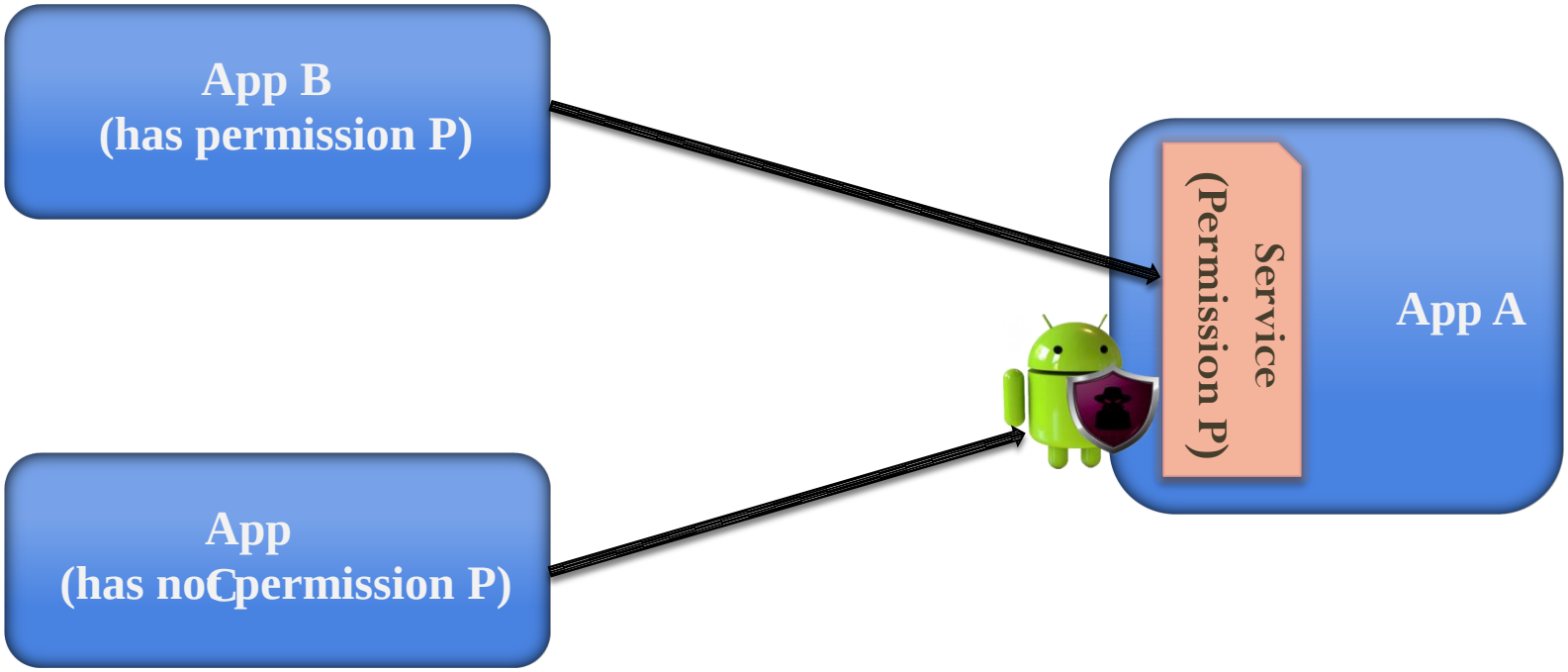
Content Provider



Android Permission System

- **Access rights** in Android's application framework
  - Permissions are required to **gain** access to
    - System interfaces (Internet, send SMS, etc.)
    - System resources (logs, battery, etc.)
    - Sensitive data (SMS, contacts, etc.)
  - Currently more than 140 default permissions defined in Android
- Permissions are **assigned** to sandbox
- Application developers can also **define** their **own** permissions

ANDROID PERMISSION: EXAMPLE

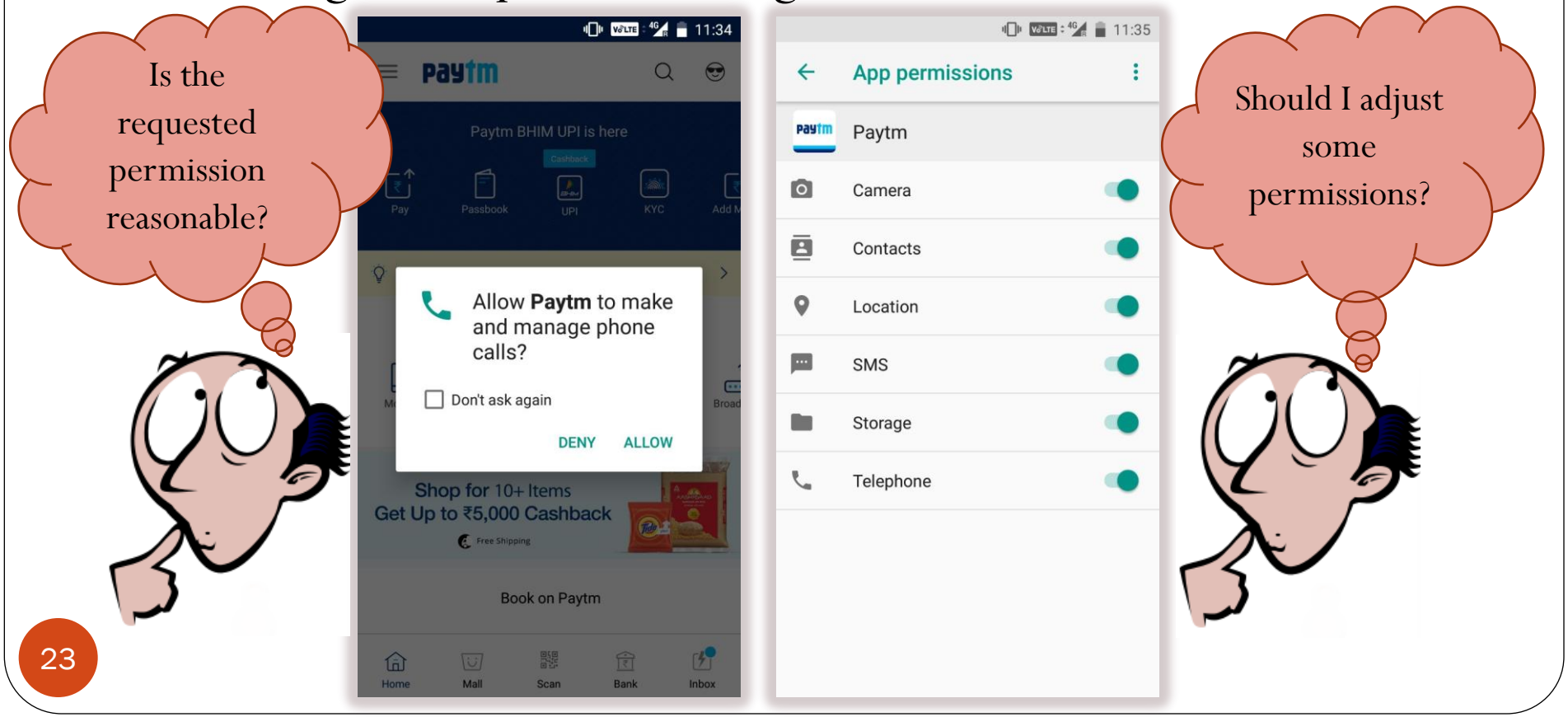


Permissions' Protection Level

- Normal
- Dangerous
- Signature
- SignatureOrSystem

Dynamic Permissions (≥ Android 6.0)

- App developers must **check** if their apps hold required **dangerous** permission, otherwise request them at runtime
- User can **grant** permissions at runtime and also **revoke** once granted permissions again

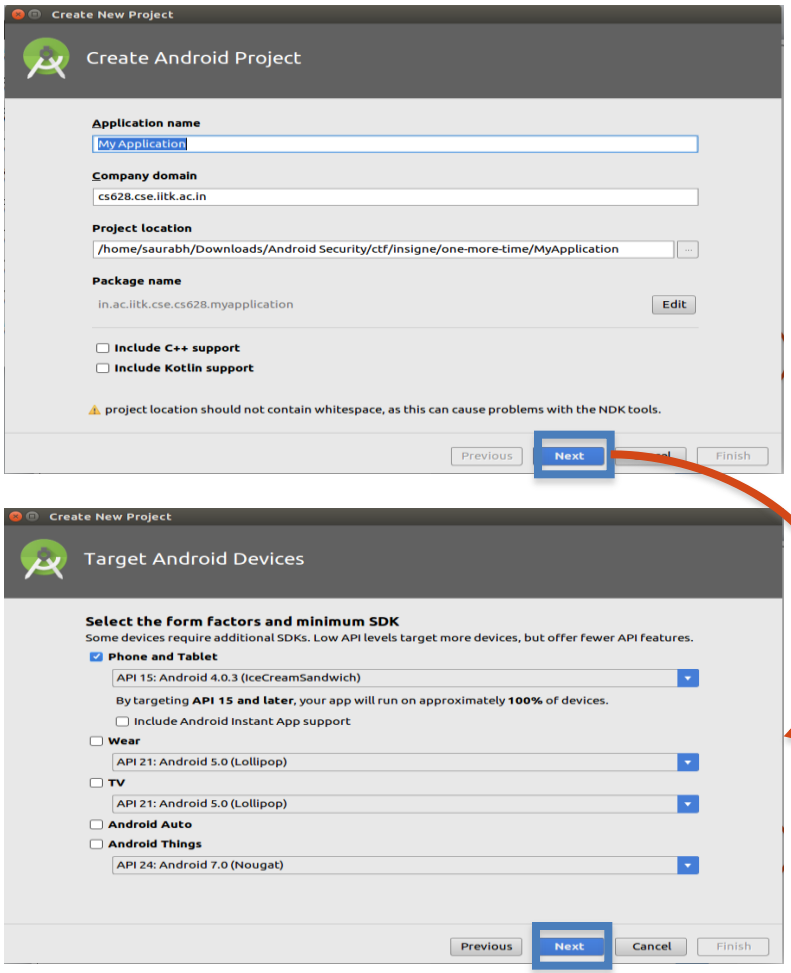


App Creation Checklist

- For Android device:
  - Ensure drivers are installed
  - Enable developer options on device under *Settings*, specifically *USB Debugging*
    - Android 4.2+: Go to *Settings*→*About phone*, press *Build number* 7 times to enable developer options
- For Android Studio:
  - Under File→*Settings*→*Appearance*, enable "Show tool window bars"; the *Android* view shows LogCat, devices
  - Programs should log states via `android.util.Log`'s `Log.d(APP_TAG_STR, "debug")`, where `APP_TAG_STR` is a `final String` tag denoting your app
  - Other commands: `Log.e()` (error); `Log.i()` (info); `Log.w()` (warning); `Log.v()` (verbose) – same parameters

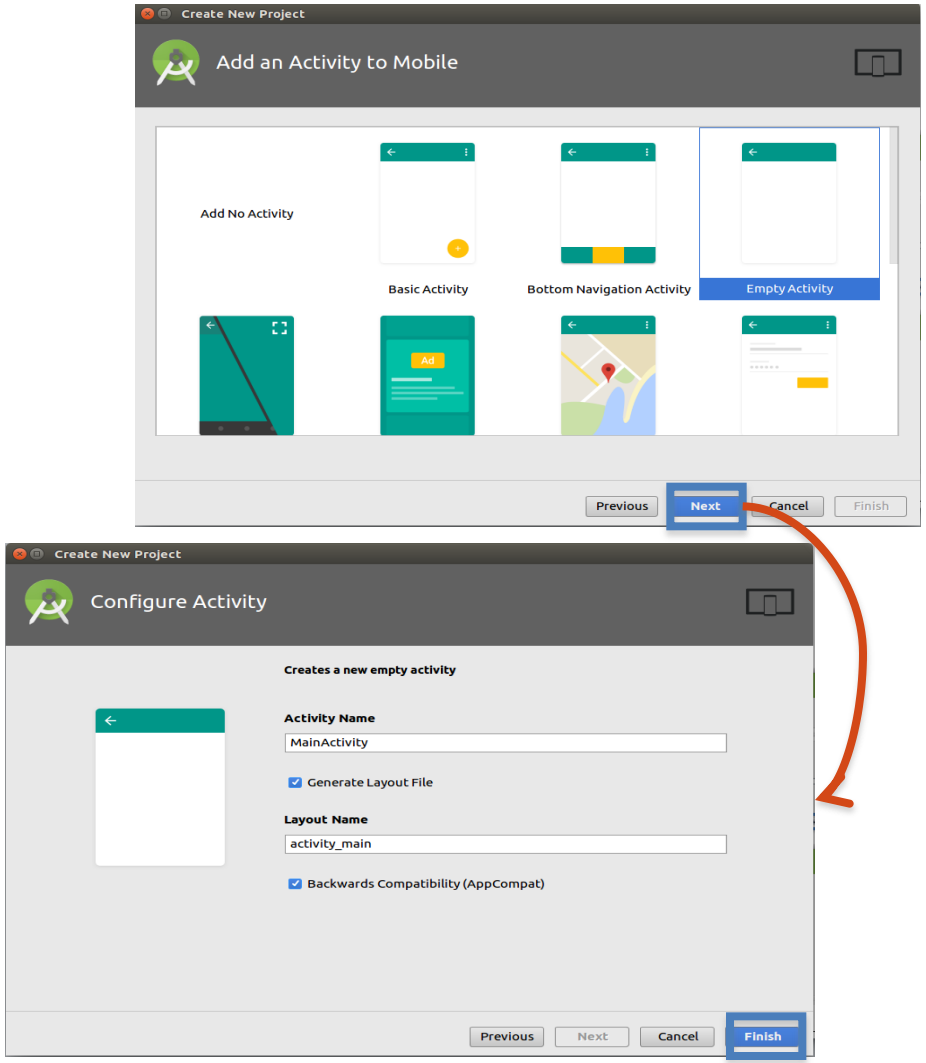
Creating Android App (1)

- Creating Android app project in Android Studio:
  - Go to *File⟶New Project*
  - Enter app, project name
  - Choose package name using "reverse URL" notation, e.g., **edu.osu.myapp,** then click Next
  - Select APIs for app, then click Next



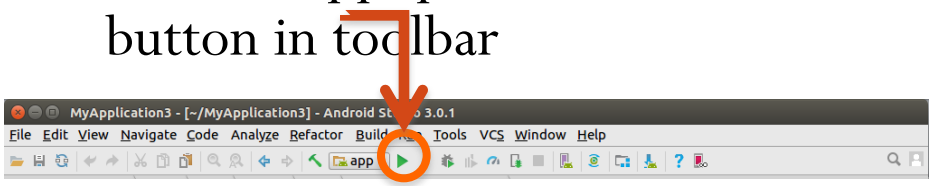
Creating Android App (2)

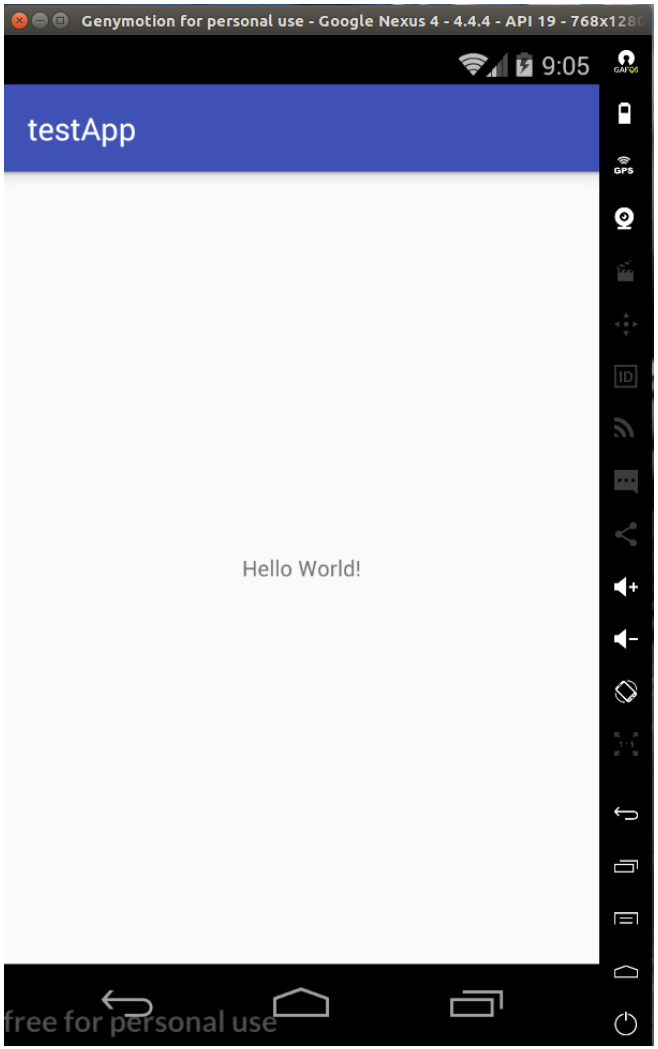
- Determine what kind of Activity to create; then click Next
  - Choose a Empty Activity for simplicity
- Enter information about your Activity, then click Finish
- This creates a "Hello World" app



Deploying the App

- Two choices for deployment:
  - Real Android device
  - Android virtual device
- Plug in your real device; otherwise, create an Android virtual device
- Emulator is slow. Try Intel accelerated version, or perhaps `http://www.genymotion.com/`
- Run the app: press "Run" button in toolbar






hands
on

References

1. IDC, May 2017. Online: https://www.idc.com/promo/smartphone-market-share/os
2. Android developer. Online: https://developer.android.com/about/dashboards/index.html
3. Android developer. Online: https://developer.android.com/guide/platform/index.html
4. Android Open Source Project, http://www.android.com
5. https://developer.android.com/guide/components/activities.html
6. https://developer.android.com/guide/topics/ui/declaring-layout.html#CommonLayouts
7. https://cloud.genymotion.com/page/doc/#collapse4
8. http://blog.zeezonline.com/2013/11/install-google-play-on-genymotion-2-0/

Thank You and Any Question?