G53MDP
Mobile Device Programming

Introduction to Android OS

What is Android?

Learning Outcomes

- Understand Android platform architecture
- Knowledge about Android compilation
- Knowledge about Android bootup & run time

Android

- An operating system for mobile phones /tablets
- Purchased by Google in 2005
- Open (sort of)
  - Open source / Apache license eventually
  - Bootloaders / rooting
- Leverages existing technology
  - Linux (customised Linux kernel)
  - Java (but not really Java)
- A different programming model

Android Version Distribution (Sep/17)

Android Name	Android Version	Usage Share
Marshmallow	6.0	32.2%
Lollipop	5.0, 5.1	28.8%
Nougat	7.0, 7.1	15.8%
Kitkat	4.4	15.1%
Jelly Bean	4.1.x, 4.2.x, 4.3.x	6.9%
Ice-Cream Sandwich	4.0.3, 4.0.4	0.6%
Gingerbread	2.3.3 to 2.3.7	0.6%

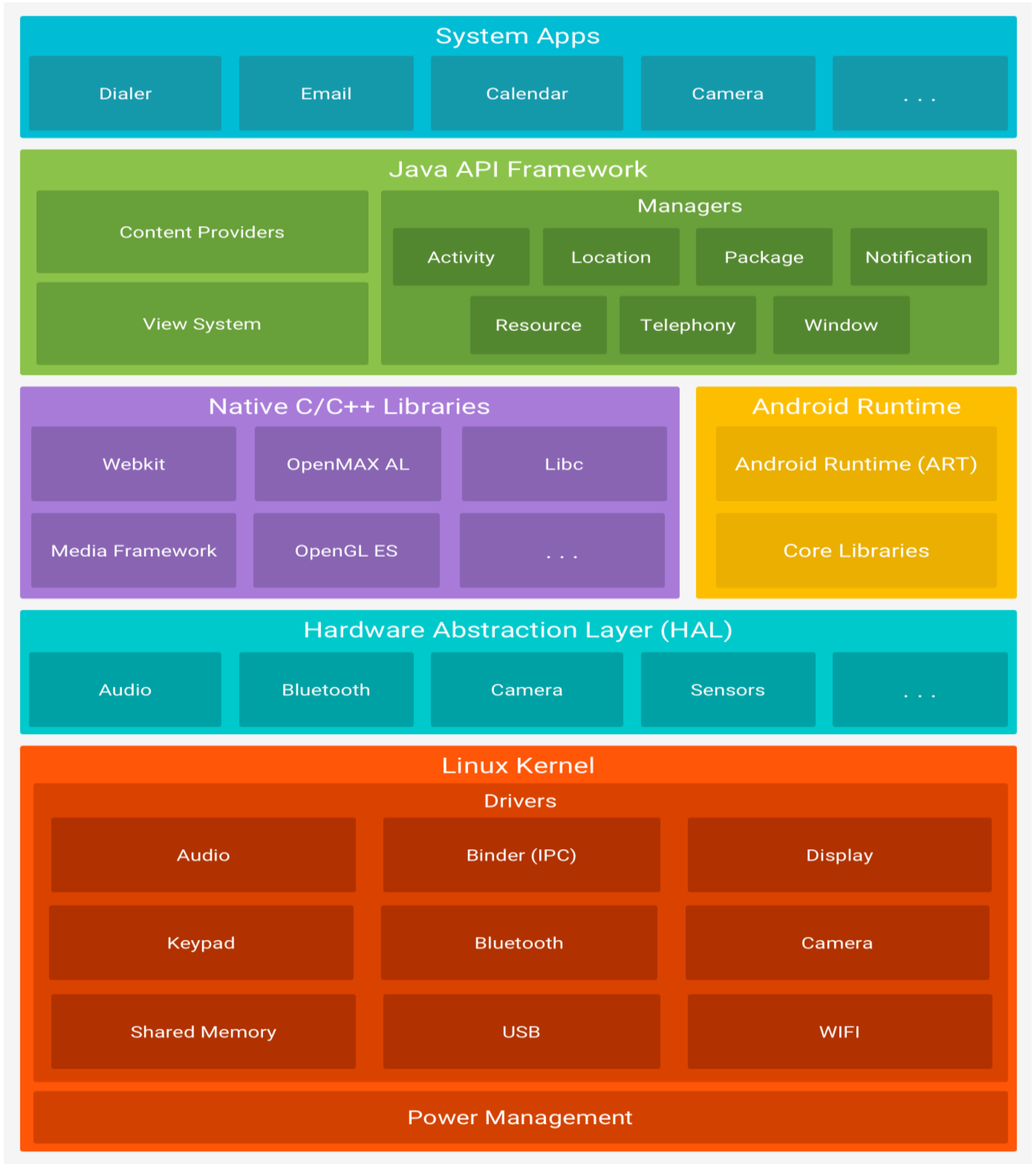
Android Compatibility

- Claims to be forwards / backwards compatible
  - An application built against 1.5 should work on the newest 7.* device
  - Some support for backwards compatibility
    - Cannot use an API that does not exist
    - Can restrict by specifying minimum API level
- The Android logo is CC licensed
- "Android phone" need to pass compatibility tests / supports the API
  - "Android" the brand licensed to Open Mobile Alliance members

Android Platform Architecture

- A software stack for mobile devices
- Operating system kernel
- Standard middleware
  - Android library support
- Key applications / user interfaces
  - Vendor specific modifications

Android Platform Architecture



- LK: threading, low-level memory management, driver
- HAL: libs for hardware module
- AR: virtual machine
- NCL: fundamental core functionalities
- API: programming interface
- APP: system apps can be customised.

Android Kernel

- Android specific modifications
  - wakelocks – keep the phone awake
  - binder – interprocess communication
  - ashmem – shared memory
  - oom – kills processes when memory is low
  - alarm manager – wakes up the phone when necessary

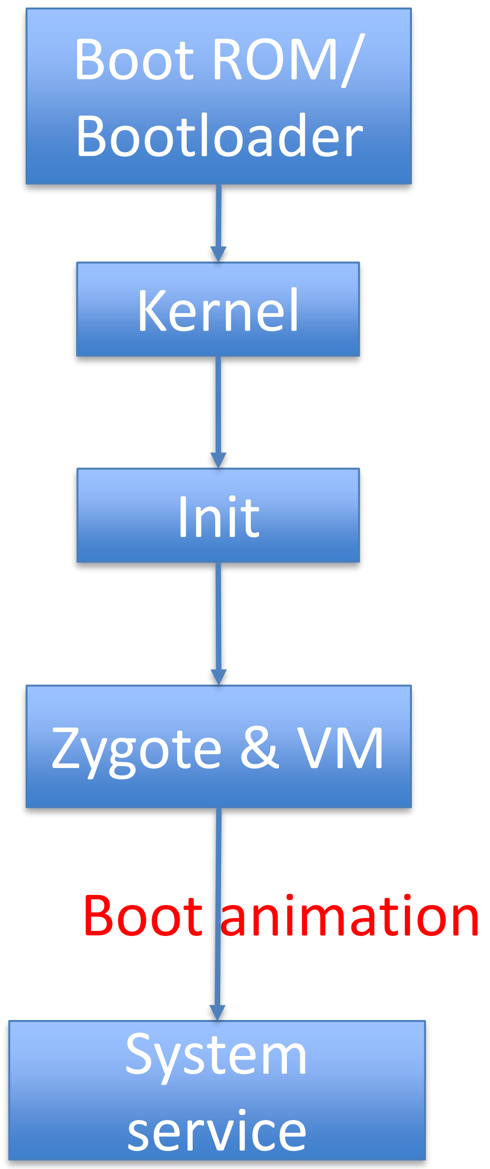
Android Hardware Support

- Bluetooth - BlueZ
- GPS – Manufacturer provided libgps.so
- Wifi – wpa_supplicant
- Display – Standard framebuffer driver
- Keyboard – Standard input event
- Lights – Manufacturer provided liblights.so
- Audio – Manufacturer provided libaudio.so
- Camera – Manufacturer provided libcamera.so
- Power Management – “wakelocks” kernel patch
- Sensors – Manufacturer provided libsensors.so
- Radio – Manufacturer provided libril.so

Android Apps

- Applications are sandboxed
  - A security mechanism for separating running applications and data
- Android application sandbox
  - Linux is a multi-user system
    - How many people use your phone at once?
  - Makes use of Linux permissions and security
    - Own process, own VM, own UID/AID for different app
    - Cannot access other application files / data / processes
      - Owner not generally given access to the root user
      - Root can access the entire system

System Bootup Process



Load bootloader into RAM, detect external RAM, setup network, memory, etc.

Setup cache, protected memory, scheduling and loads drivers.

Mounts directories like /sys , /dev or /proc
Runs init.rc script

Enables code sharing across the android VM for quick start of separate VM for different apps
preloadClasses(), preloadResources()

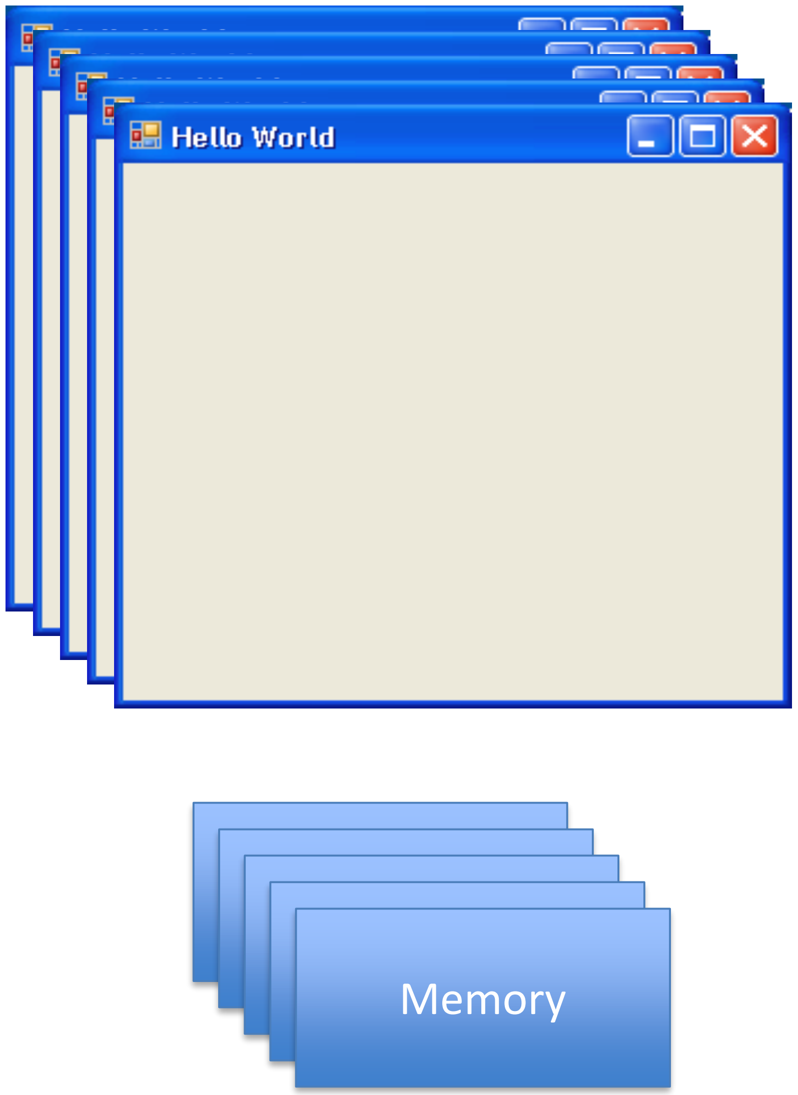
Power manager, activity manager, telephony registry, package manger, context manager, system contact providers, etc.

Zygote

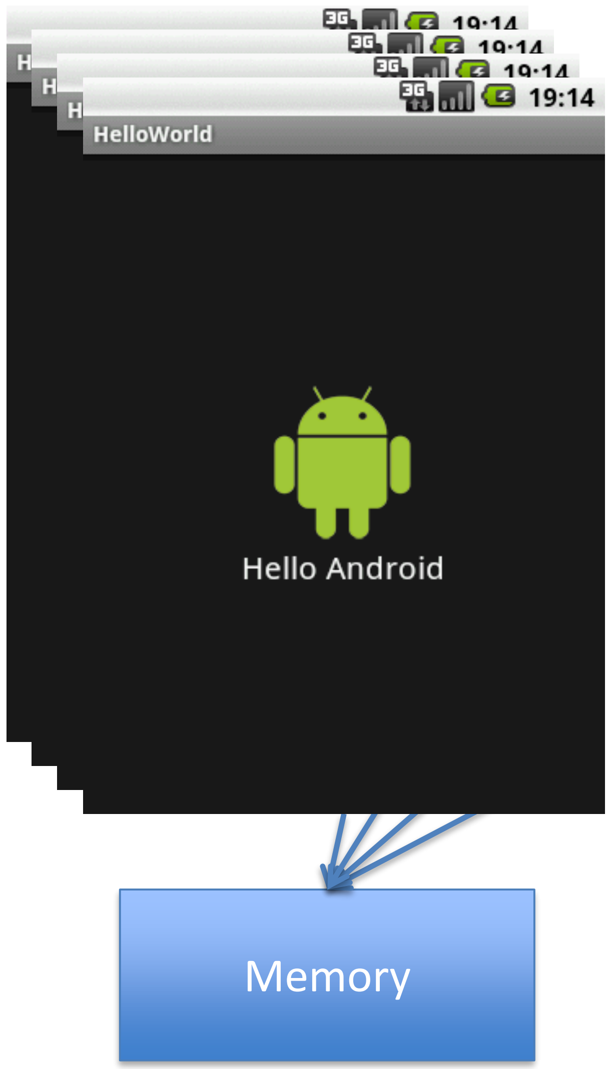
- Initialised process that has all core libraries linked in
- Load all java.*, android.* classes at boot time
- Initially create a single android VM process
  - Referencing classes loaded above
- When user runs an application
  - Creates a copy of itself in a separate address space
  - Does not copy memory, instead refers to original memory until modified

Shared Memory

Java



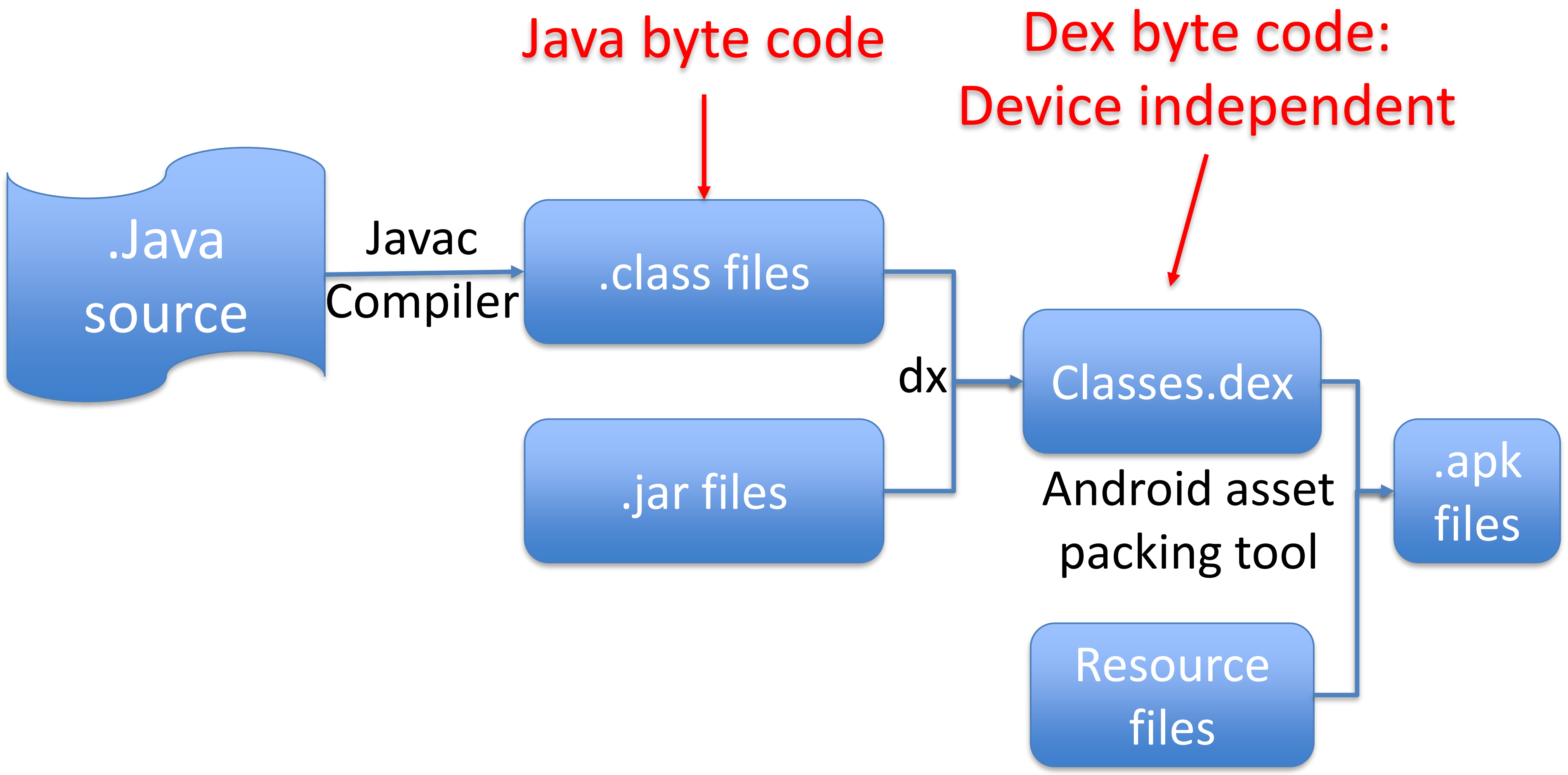
Android



Android Compilation

- Applications are written in Java
  - Run on Google’s own VM — Dalvik/ Android Run Time
  - Uses its own bytecode (DEX) format
- Code compiled using standard Java tools then convert to DEX format
  - Multiple class files in a single .dex file
- Code, data and resource files packed into a .apk file
  - Classes
  - Configuration
  - Resources

Android Compilation

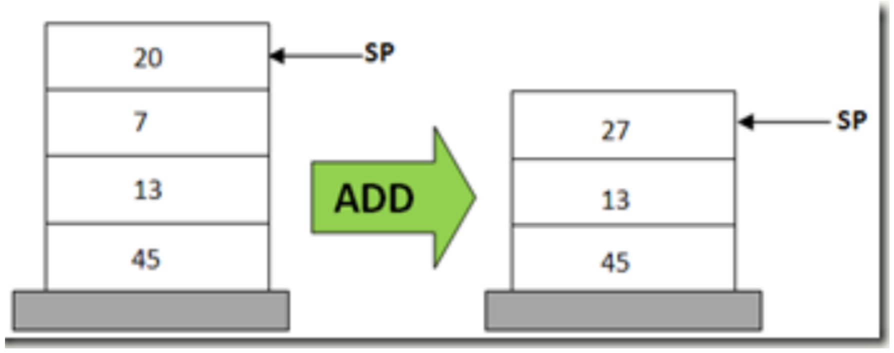


Dalvik

- Dalvik architecture is register based rather than stack based.
- Optimised to use less space
- Execute its own Dalvik byte code rather than Java bytecode
- Dalvik interprets .dex files
  - Post-processes .class files
  - Size reduction
  - JIT compilation to native ARM instructions
- Target slow cpu, no swap, low RAM, battery powered

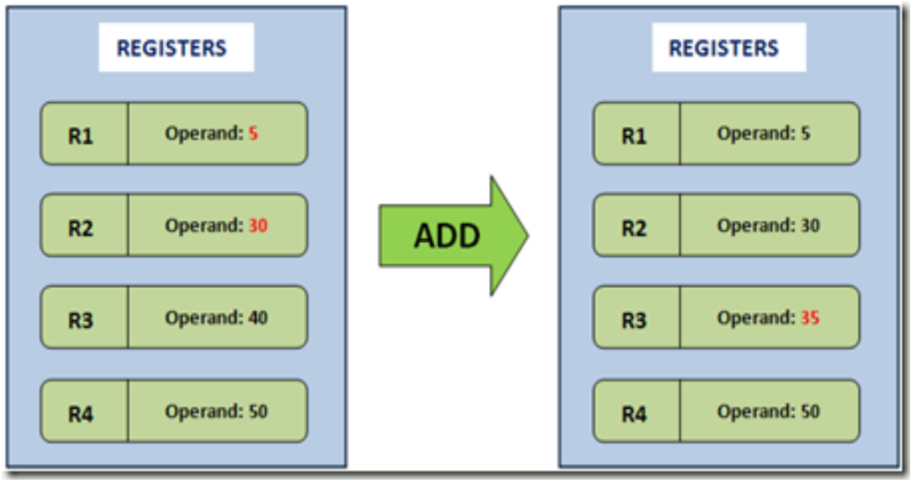
Stack Based VM vs Registered Based VM

Java VM



1. **POP 20**
2. **POP 7**
3. **ADD 20, 7, result**
4. **PUSH result**

Dalvik

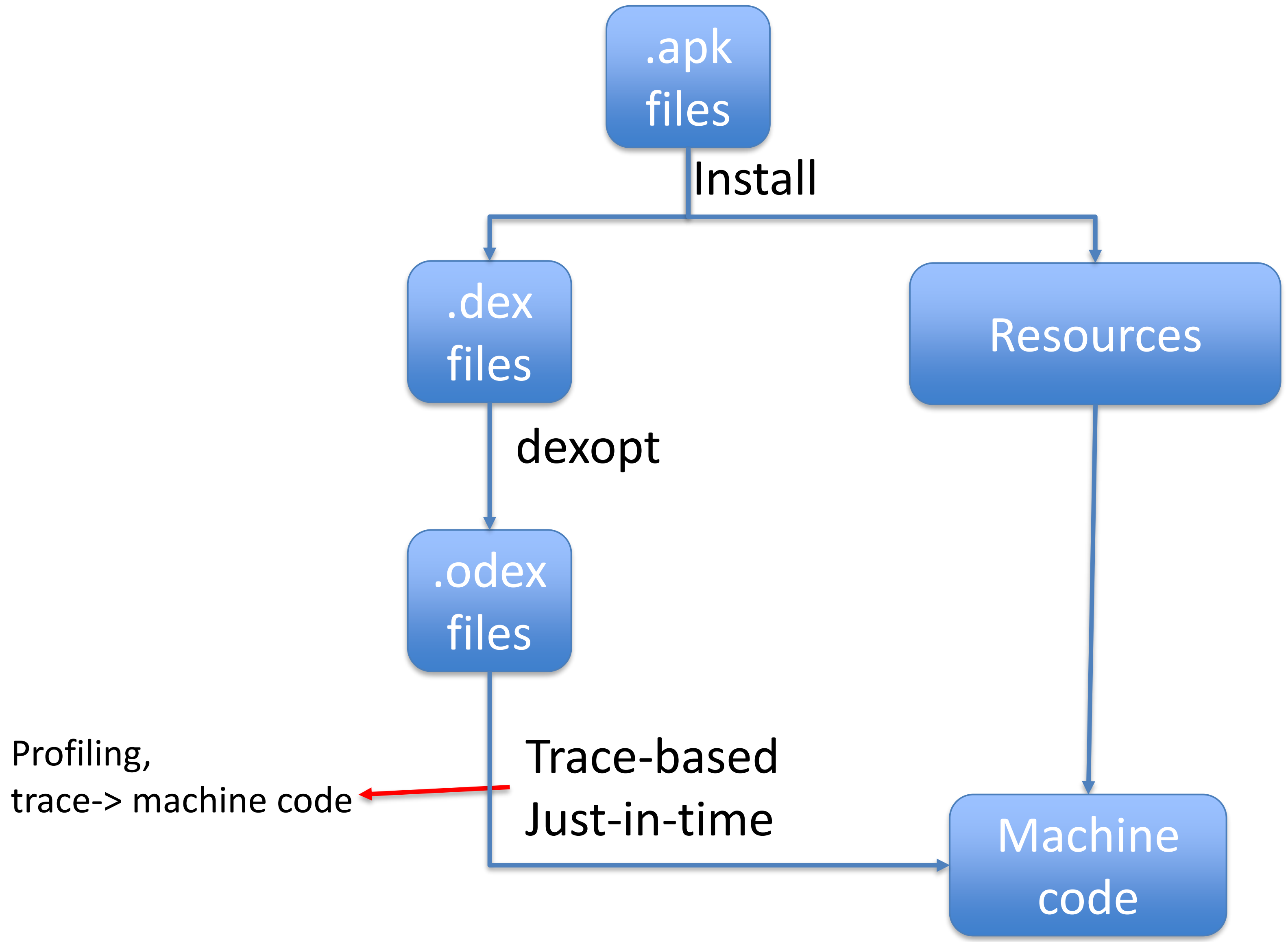


1. **ADD R1, R2, R3 ;** # Add contents of R1 and R2, store result in R3

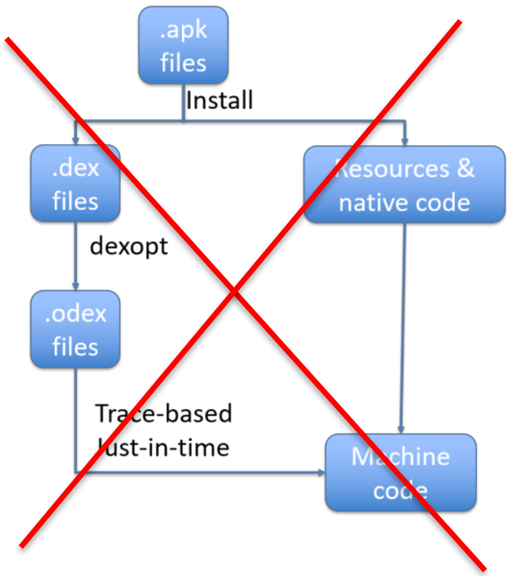
✓ Less executed instructions

X Instruction is larger than stack based

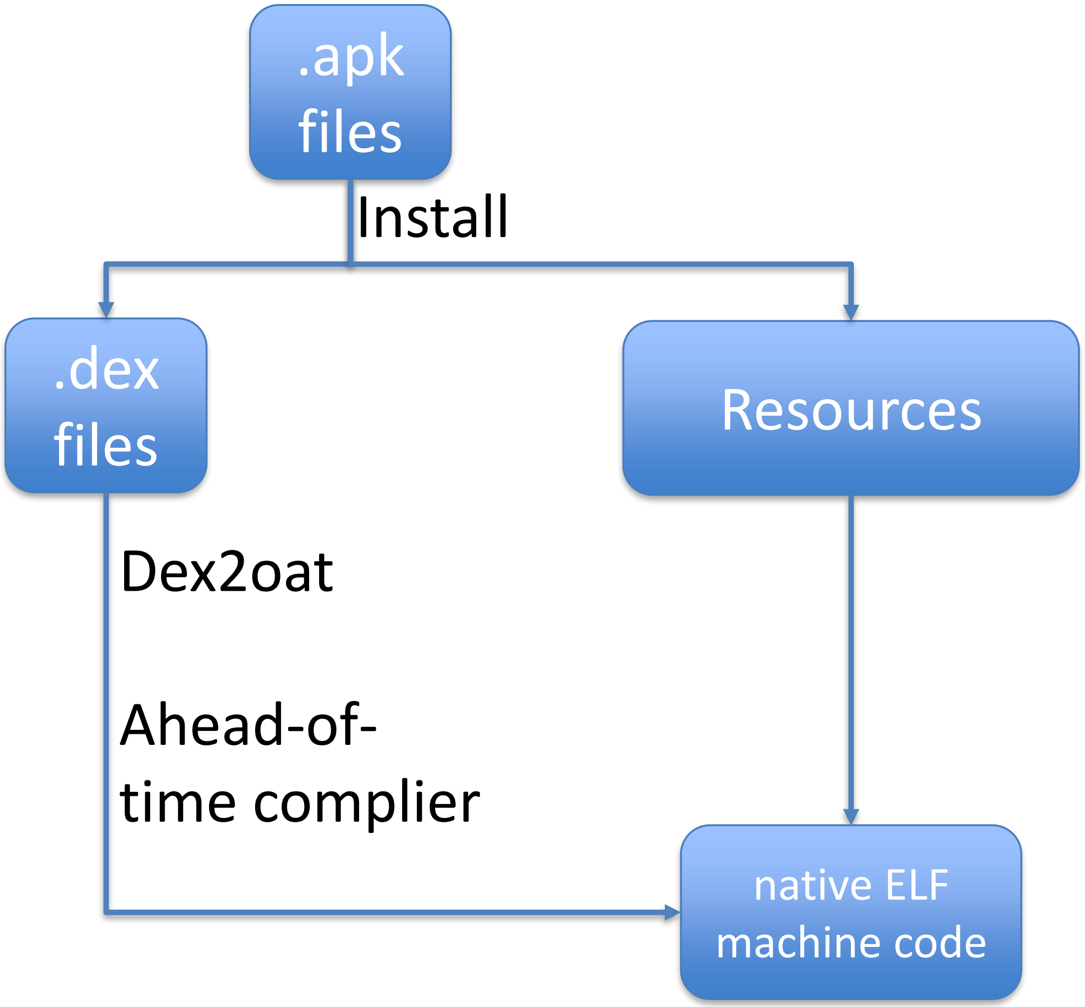
Runtime Environment with Dalvik



Runtime Environment with ART



Dalvik was replaced by Android Runtime (ART) after Adnroid 5.0 (Lollipop)



ART

- ✓ Apps run faster as DEX bytecode translation done during installation
- ✓ Reduces start-up time of applications as native code is directly executed
- ✓ Improves battery performance as power utilised to interpreted byte codes line by line is saved
- x App Installation takes more time because of DEX bytecodes conversion into machine code
- x More internal storage is required to store the fully converted machine code at installation

Android 7.0

- Android 7.0 adds a JIT compiler with code profiling to ART that constantly improves the performance of Android apps as they run.

Android Programming Model

- Traditional OS applications
  - A single entry point
    - Main
  - OS loads the program into a process and executes it
- Java applications
  - A Java VM is instantiated
    - Loads all classes used by the application
    - Executes main
- Component based model
  - Multiple application entry points
    - The point through which the system can “enter” the application

Android Components

- Activities
  - UI components
- Services
  - Mechanism for doing something long-running in the background
- Broadcast Receivers
  - Respond to broadcast messages from the OS / other apps
- Content Providers
  - Make data available to / make use of data from other apps

Summary

- Android platform architecture

  Android kernel, hardware layer, ART, java API, apps

- Knowledge about Android compilation

  dex bytecode

- Knowledge about Android bootup & run time

  Zygote, Dalvik/ ART