ANDROID DEVELOPMENT

THE ANDROID SYSTEM

Tan Cher Wah (isstcw@nus.edu.sg)

Agenda

- Dalvik Virtual Machine
- Android Runtime
- Application Framework
- Storages

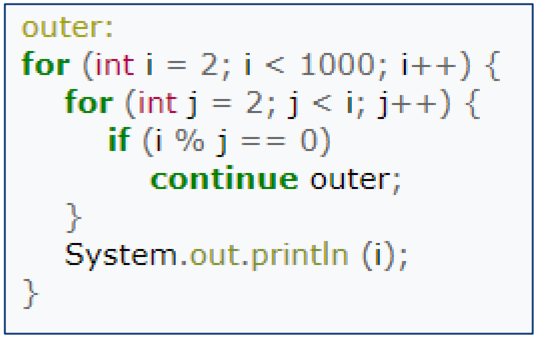
What is Android?

- Android is an operating system for mobile devices
  - Phones
  - Tablets
  - Wearables
- Based on Linux kernel
- Includes SDK, APIs and tools
- Recent versions
  - 5 **L**ollipop
  - 6 **M**arshmallow
  - 7 **N**ougart
  - 8 **O**reo
  - 9 **P**ie
  - 10 Android **Q**

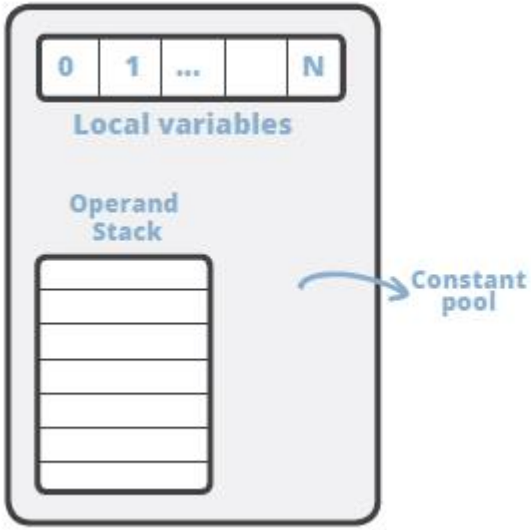
Java as Android's programming language

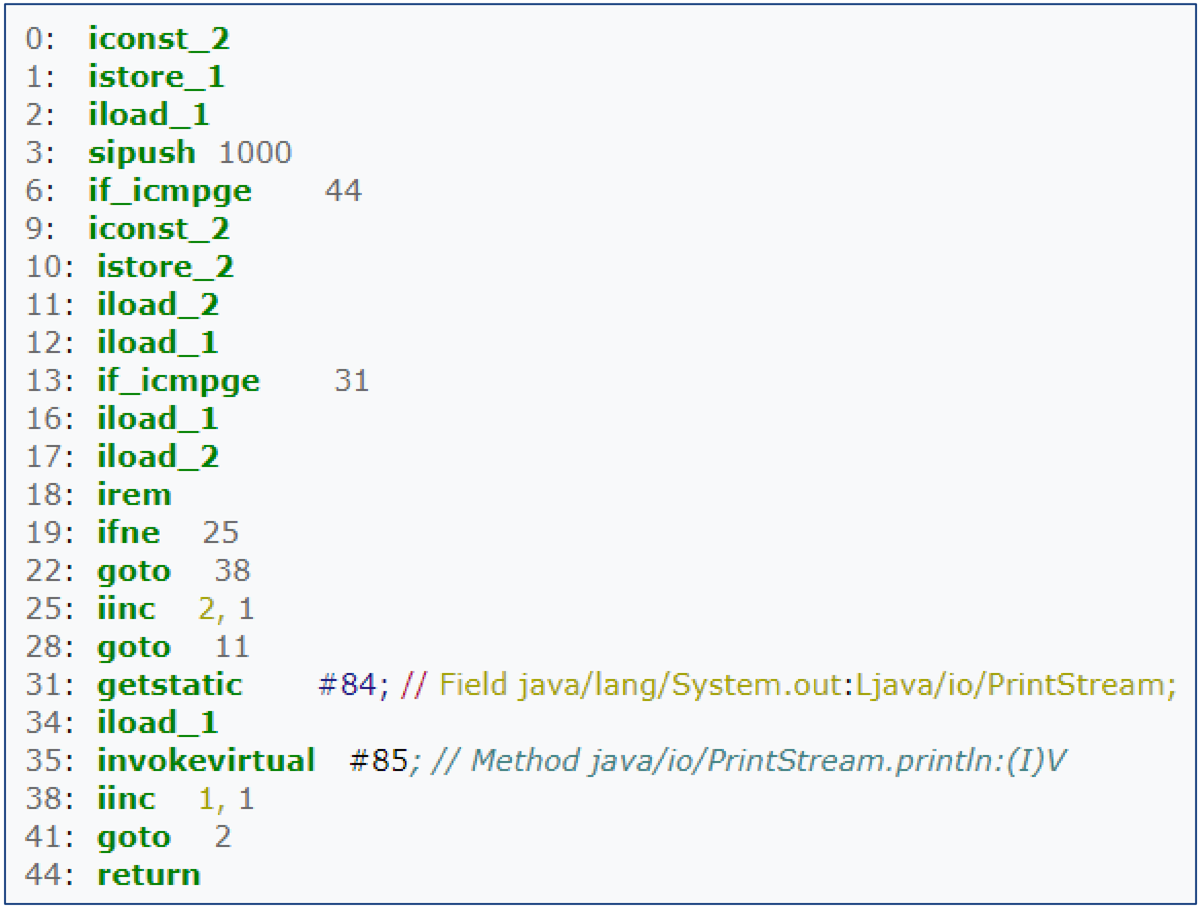
- Android chose Java language as its programming language (a new alternative is Kotlin)
- Java .classes run on Java Virtual Machine (JVM)
  - A .class file contains Java bytecodes
  - JVM is based on a Stack-based Architecture
    - All operations are done via the Stack (load, push, pop)

Java bytecodes

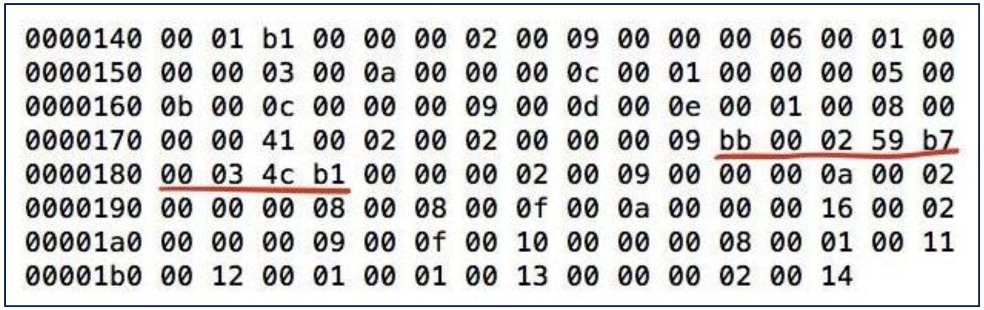


```
outer:
for (int i = 2; i < 1000; i++) {
    for (int j = 2; j < i; j++) {
        if (i % j == 0)
            continue outer;
    }
    System.out.println (i);
}
```





```
0:   iconst_2
1:   istore_1
2:   iload_1
3:   sipush  1000
6:   if_icmpge      44
9:   iconst_2
10:  istore_2
11:  iload_2
12:  iload_1
13:  if_icmpge      31
16:  iload_1
17:  iload_2
18:  irem
19:  ifne    25
22:  goto    38
25:  iinc    2, 1
28:  goto    11
31:  getstatic      #84; // Field java/lang/System.out:Ljava/io/PrintStream;
34:  iload_1
35:  invokevirtual  #85; // Method java/io/PrintStream.println:(I)V
38:  iinc    1, 1
41:  goto    2
44:  return
```



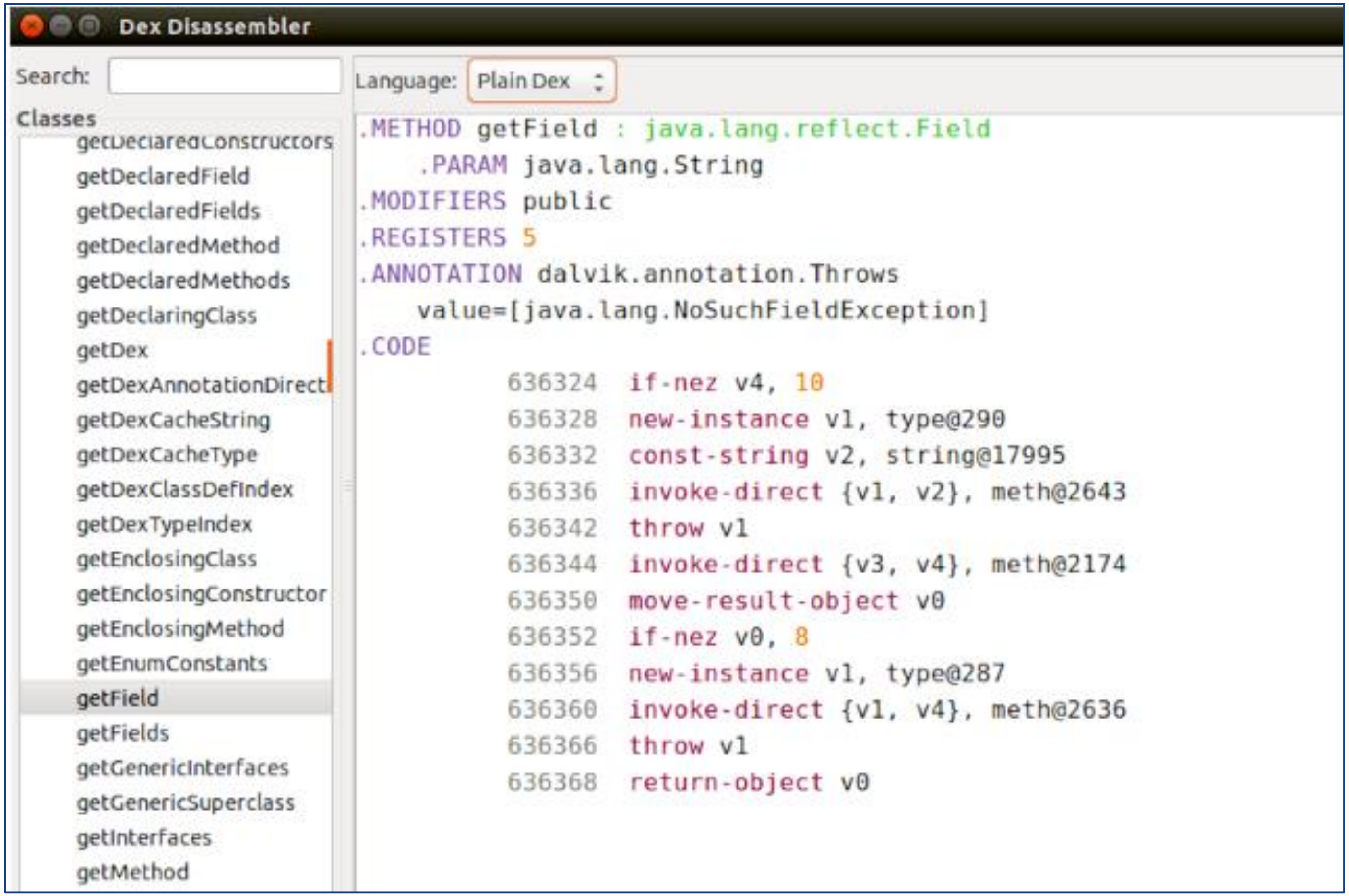
```
0000140 00 01 b1 00 00 00 02 00 09 00 00 00 06 00 01 00
0000150 00 00 03 00 0a 00 00 00 0c 00 01 00 00 00 05 00
0000160 0b 00 0c 00 00 00 09 00 0d 00 0e 00 01 00 08 00
0000170 00 00 41 00 02 00 02 00 00 00 09 bb 00 02 59 b7
0000180 00 03 4c b1 00 00 00 02 00 09 00 00 00 0a 00 02
0000190 00 00 00 08 00 08 00 0f 00 0a 00 00 00 16 00 02
00001a0 00 00 00 09 00 0f 00 10 00 00 00 08 00 01 00 11
00001b0 00 12 00 01 00 01 00 13 00 00 00 02 00 14
```

Dalvik

- Android code does not run on JVM, but on another virtual machine called Dalvik
- Dalvik uses a Register-based Architecture
- Code and data bundled into **.apk** Android package
- Strategy for Speed - Just in Time (JIT) compilation
  - Profiles DEX bytecode during runtime and dynamically compiles frequently executed codes into a device’s native instructions

DEX bytecode

- Sample DEX bytecode



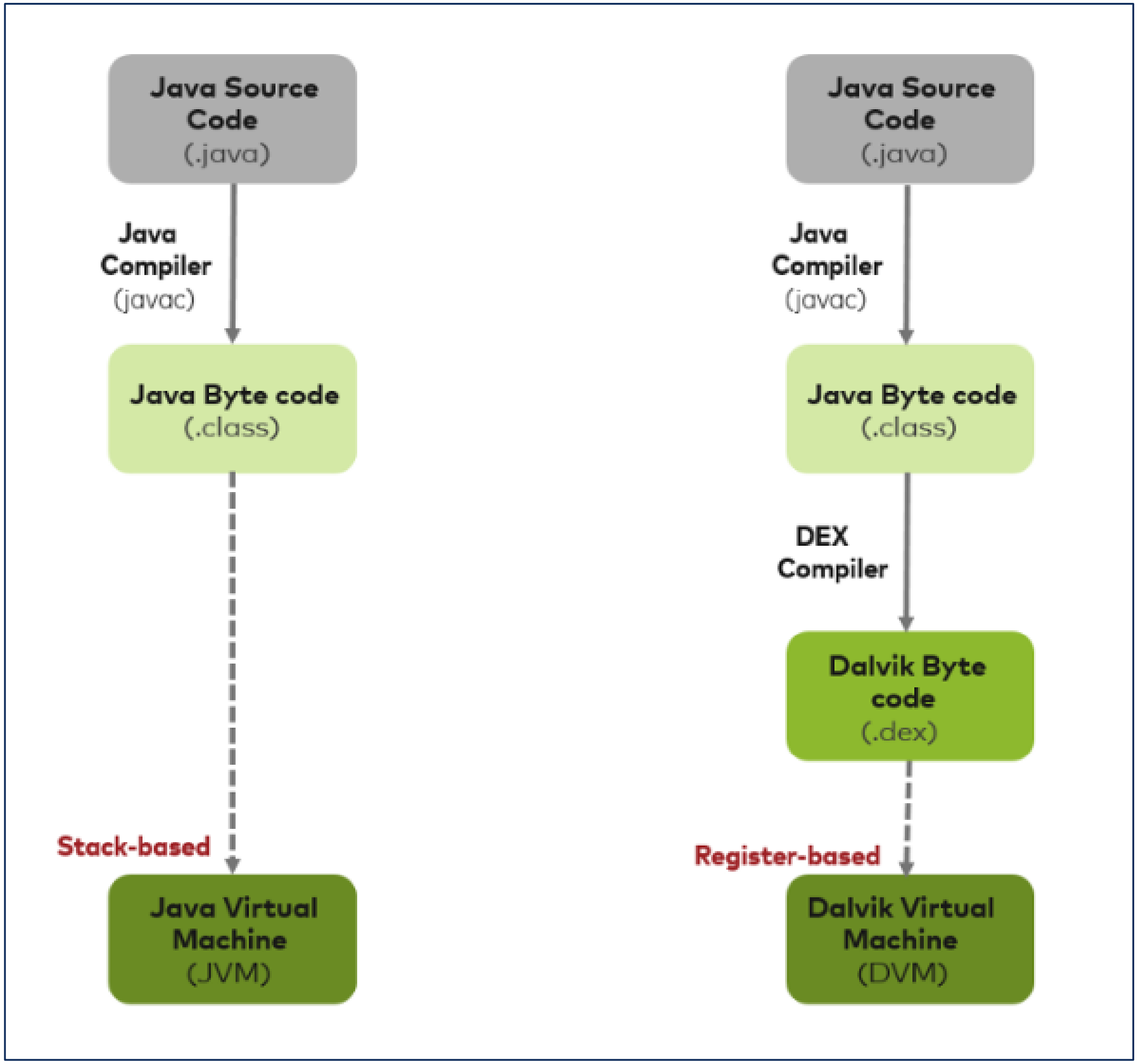
Language: Plain Dex

```
.METHOD getField : java.lang.reflect.Field
    .PARAM java.lang.String
.MODIFIERS public
.REGISTERS 5
.ANNOTATION dalvik.annotation.Throws
    value=[java.lang.NoSuchFieldException]
.CODE
          636324  if-nez v4, 10
          636328  new-instance v1, type@290
          636332  const-string v2, string@17995
          636336  invoke-direct {v1, v2}, meth@2643
          636342  throw v1
          636344  invoke-direct {v3, v4}, meth@2174
          636350  move-result-object v0
          636352  if-nez v0, 8
          636356  new-instance v1, type@287
          636360  invoke-direct {v1, v4}, meth@2636
          636366  throw v1
          636368  return-object v0
```

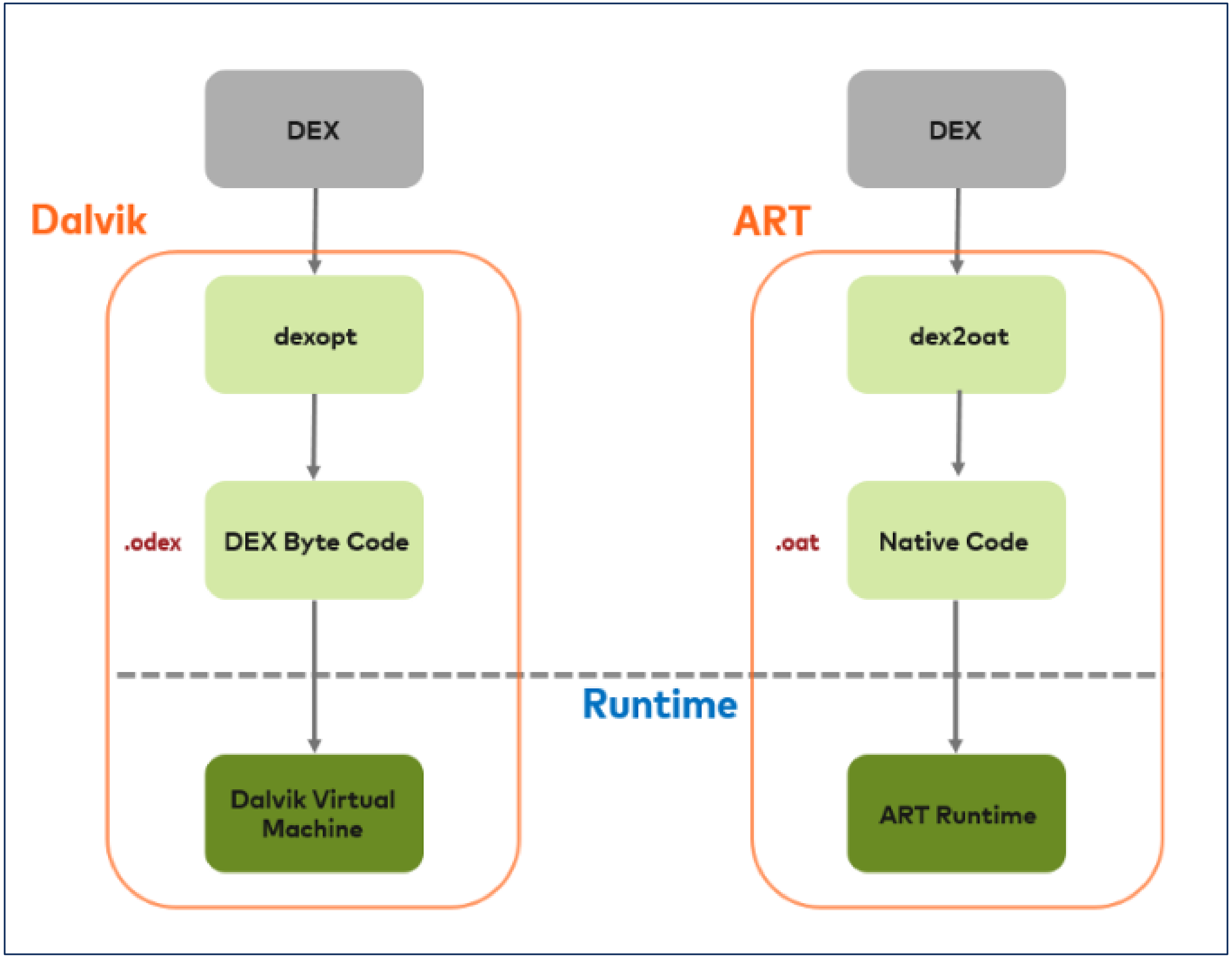
Android Runtime

- Android Runtime (ART)
  - Precedes Dalvik
  - Ahead of Time (AOT) compilation
    - Translates DEX bytecode into an Android device’s **native** instructions
  - Distribution
    - During installation, Android only downloads the binary that uses instruction sets of the target ARM processor

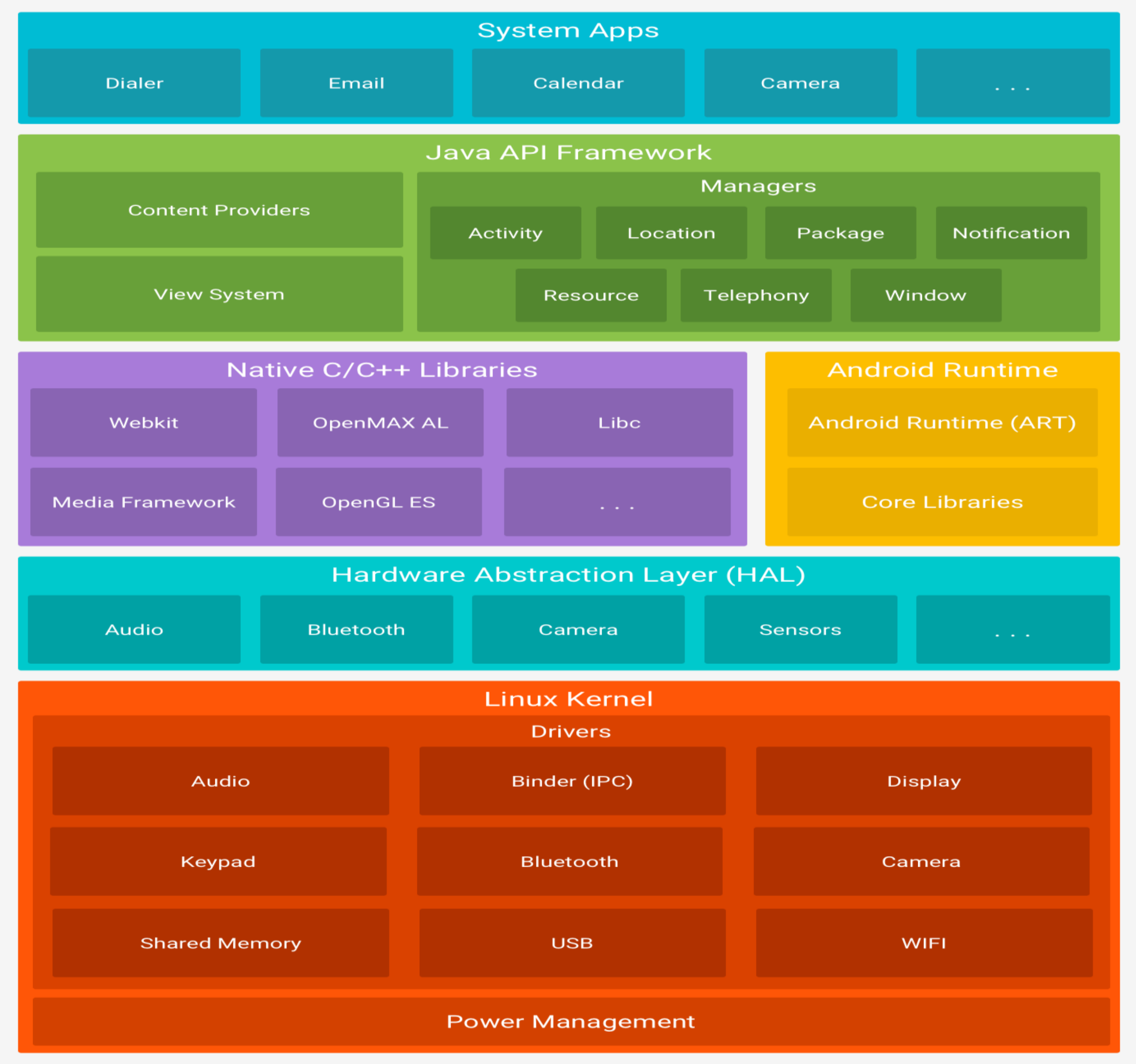
JVM vs DVM


Java Source Code (.java)
Java Compiler (javac)
Java Byte code (.class)
Stack-based
Java Virtual Machine (JVM)
Java Source Code (.java)
Java Compiler (javac)
Java Byte code (.class)
DEX Compiler
Dalvik Byte code (.dex)
Register-based
Dalvik Virtual Machine (DVM)

Dalvik vs ART



Platform Architecture diagram



System Apps

Dialer · Email · Calendar · Camera · . . .

Java API Framework

Content Providers

View System

Managers

Activity · Location · Package · Notification

Resource · Telephony · Window

Native C/C++ Libraries

Webkit · OpenMAX AL · Libc

Media Framework · OpenGL ES · . . .

Android Runtime

Android Runtime (ART)

Core Libraries

Hardware Abstraction Layer (HAL)

Audio · Bluetooth · Camera · Sensors · . . .

Linux Kernel

Drivers

Audio · Binder (IPC) · Display

Keypad · Bluetooth · Camera

Shared Memory · USB · WIFI

Power Management

Application Framework

- Android Components
  - **Activities** – screen with UI
  - **Services** – runs in background without UI
  - **Content Providers** – sharing of app data
  - **Broadcast Receivers** – responds to notifications

Application Framework

- Android apps written in Java
- Code and data bundled into **.apk** Android package
- Every app has manifest file called **AndroidManifest.xml**
  - The first file read by Android when installing an app
  - States **Permissions** for operations (e.g. WRITE_EXTERNAL_STORAGE)
  - Specify every **Activity** found in your app
  - Specify any other Android components used (e.g. Services, Providers)
  - **NOT** for defining **UI layouts**

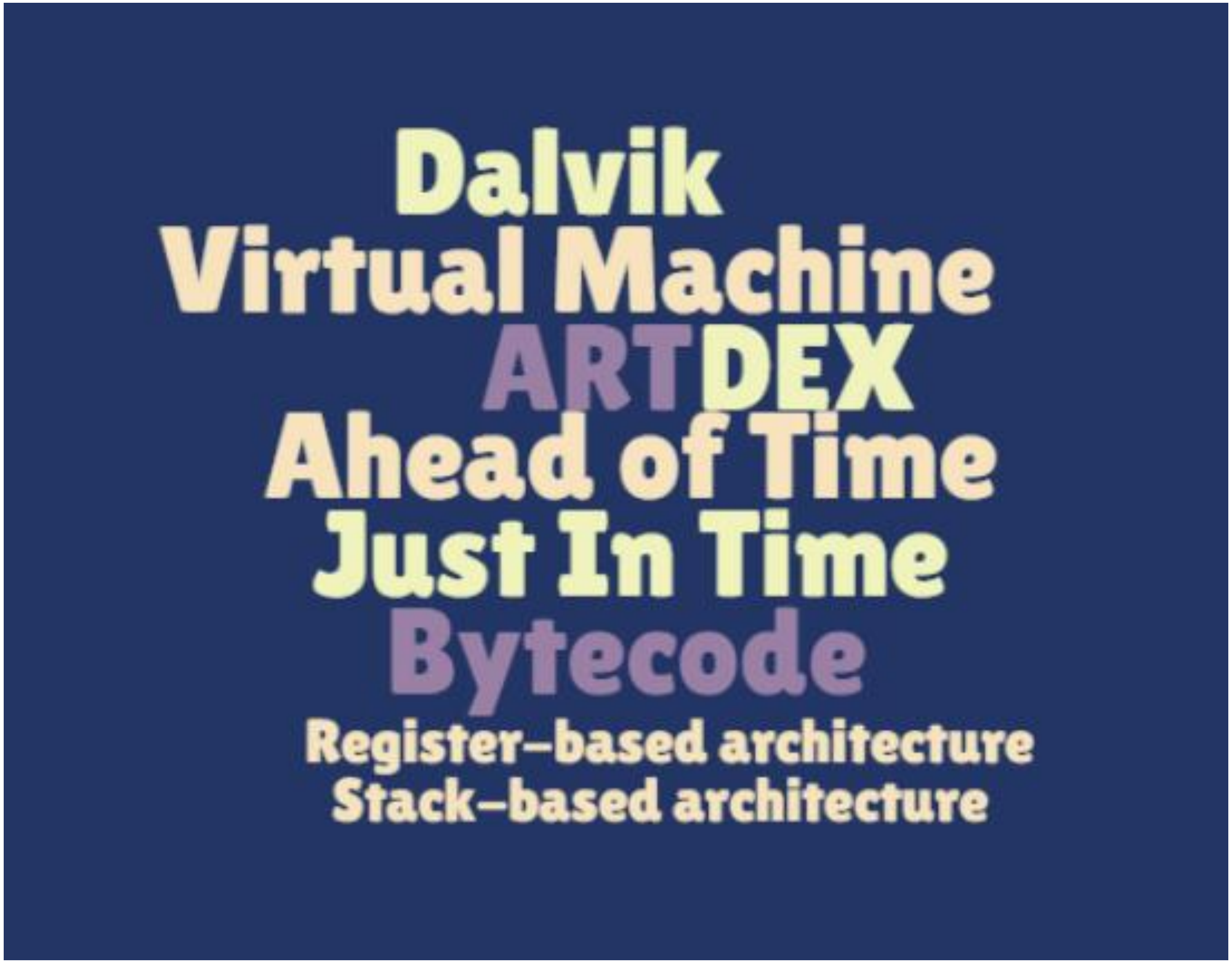
Storages

- Internal Storage
  - Folder path – “data/data/”
  - Only accessible by Android apps
    - “data/data/com.myapps.app1” is only accessible to the app called “com.myapps.app1”
  - Not visible to users
- External Storage
  - Includes SD Card (e.g. /mnt/sdcard)
  - Users can store data (e.g. music) in it
  - Accessible to Android apps too

References

- Java Virtual Machine - https://en.wikipedia.org/wiki/Java_virtual_machine
- Java bytecode - https://en.wikipedia.org/wiki/Java_bytecode
- The DEX File Format - https://www.bugsnag.com/blog/dex-and-d8
- Dalvik Virtual Machine - https://en.wikipedia.org/wiki/Dalvik_(software)

Active Recall


Dalvik
Virtual Machine
ART
DEX
Ahead of Time
Just In Time
Bytecode
Register–based architecture
Stack–based architecture