



SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

COURSE NAME :19CS603- MOBILE APPLICATION DEVELOPMENT

III YEAR /VI SEMESTER

Unit 4- Introduction to I-Android

Topic : Android Architecture





What is Android?



- » Android is a software stack for mobile devices that includes an operating system, middleware and key applications.

Android is an open source operating system, created by Google specifically for use on mobile devices (cell phones and tablets)



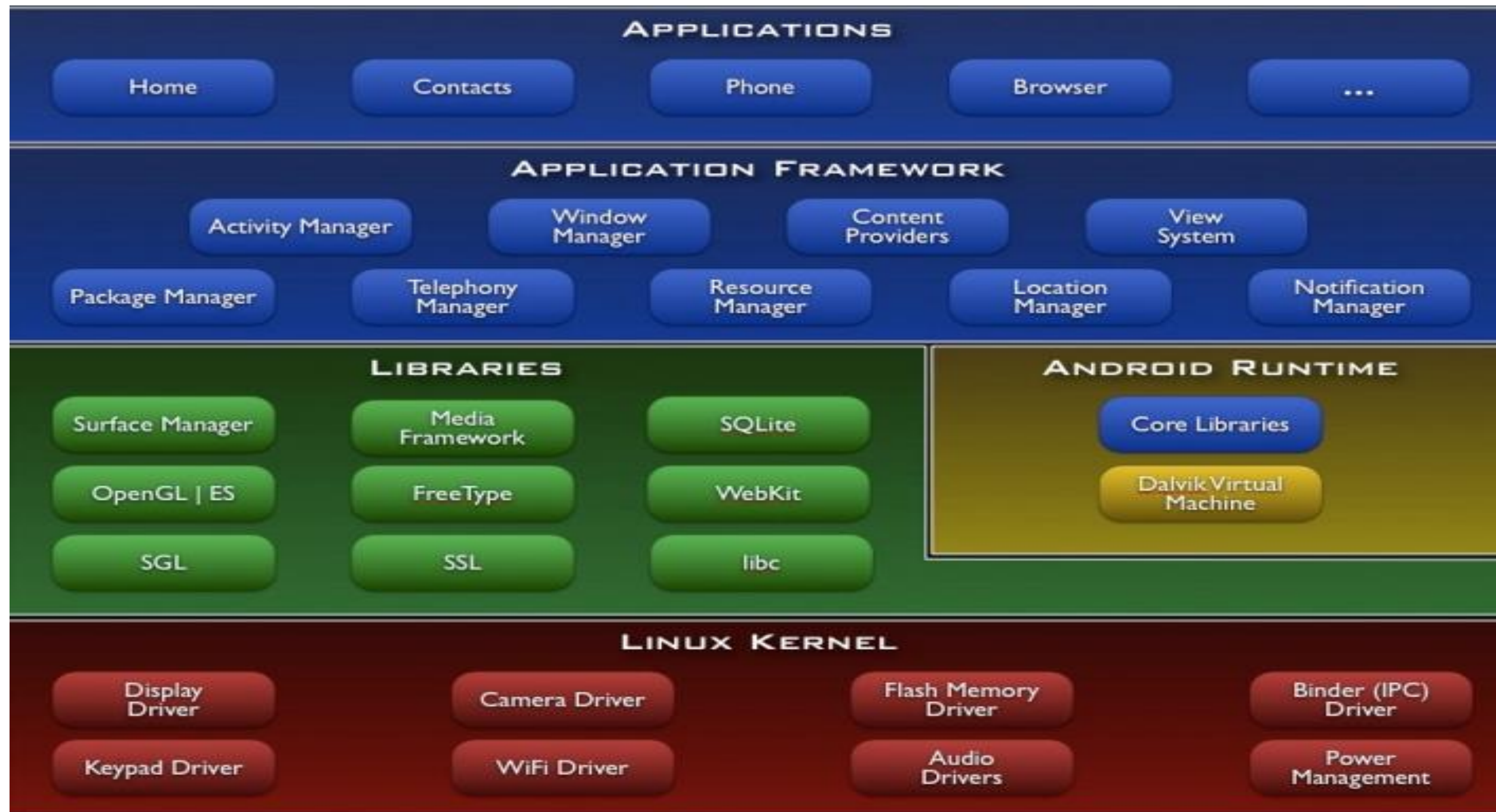
Platform Versions



Code name	Version number(s)	Initial release date	API level
No codename	1.0	September 23, 2008	1
	1.1	February 9, 2009	2
Cupcake	1.5	April 27, 2009	3
Donut	1.6	September 15, 2009	4
Eclair	2.0 – 2.1	October 26, 2009	5 – 7
Froyo	2.2 – 2.2.3	May 20, 2010	8
Gingerbread	2.3 – 2.3.7	December 6, 2010	9 – 10
Honeycomb	3.0 – 3.2.6	February 22, 2011	11 – 13
Ice Cream Sandwich	4.0 – 4.0.4	October 18, 2011	14 – 15
Jelly Bean	4.1 – 4.3.1	July 9, 2012	16 – 18
KitKat	4.4 – 4.4.4	October 31, 2013	19 – 20
Lollipop	5.0 – 5.1.1	November 12, 2014	21 – 22
Marshmallow	6.0 – 6.0.1	October 5, 2015	23
Nougat	7.0 – 7.1.2	August 22, 2016	24 – 25
Oreo	8.0 – 8.1	August 21, 2017	26 – 27
Pie	9.0	August 6, 2018	28
Android 10	10.0	September 3, 2019	29



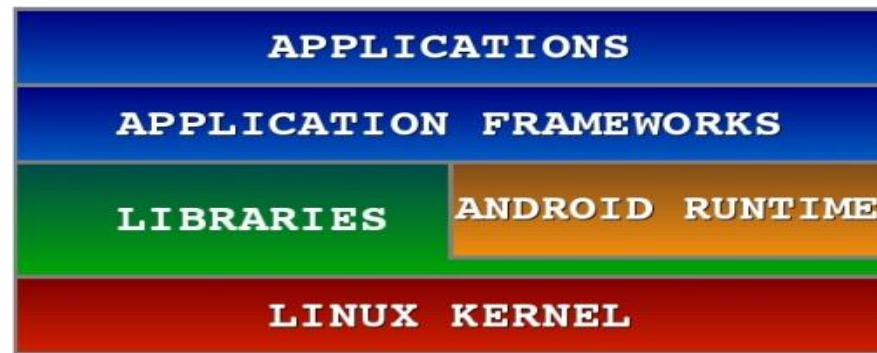
Architecture



**Stack
Architecture**



Android S/W Stack - Application



Android provides a set of core applications:

- ✓ Email Client
- ✓ SMS Program
- ✓ Calendar
- ✓ Maps
- ✓ Browser
- ✓ Contacts
- ✓ Etc

All applications are written using the Java language.

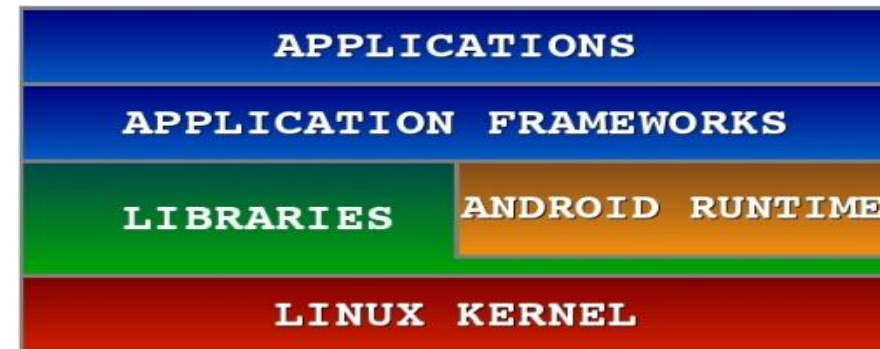
Applications

(Written in Java code)

- **Android Play Store**
- **Entertainment**
- **Productivity**
- **Personalization**
- **Education**
- **Geo-communication**
-



Android S/W Stack –App Framework



Enabling and simplifying the reuse of components

- ✓ Developers have full access to the same framework APIs (Application programming interface: set of routines, protocols and tools for building software application) used by the core applications.
- ✓ Users are allowed to replace components.

Feature	Role
View System	Used to build an application, including lists, grids, text boxes, buttons, and embedded web browser
Content Provider	Enabling applications to access data from other applications or to share their own data
Resource Manager	Providing access to non-code resources (localized strings, graphics, and layout files)
Notification Manager	Enabling all applications to display customer alerts in the status bar
Activity Manager	Managing the lifecycle of applications and providing a common navigation backstack



Android S/W Stack - Libraries



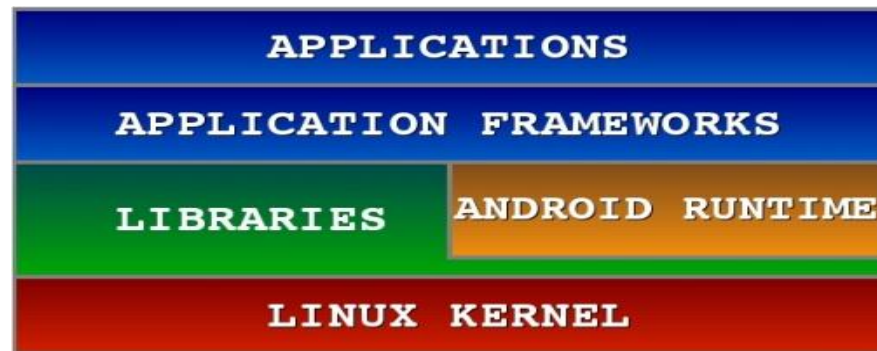
- Including a set of C/C++ libraries used by components of the Android system
- Exposed to developers through the Android application framework

Native Libraries (C/C++ code)

- **Graphics** (Surface Manager)
- **Multimedia** (Media Framework)
- **Database DBMS** (SQLite)
- **Font Management** (FreeType)
- **WebKit**
- **C libraries** (Bionic)
-



Android S/W Stack - Runtime



Core Libraries

- ✓ Providing most of the functionality available in the core libraries of the Java language
- ✓ APIs
 - Data Structures
 - Utilities
 - File Access
 - Network Access
 - Graphics
 - Etc

Dalvik Virtual Machine (VM)

- **Novel** Java Virtual Machine implementation (not using the Oracle JVM)
- Open **License** (Oracle JVM is not open!)
- **Optimized** for memory-constrained devices
- **Faster** than Oracle JVM
-

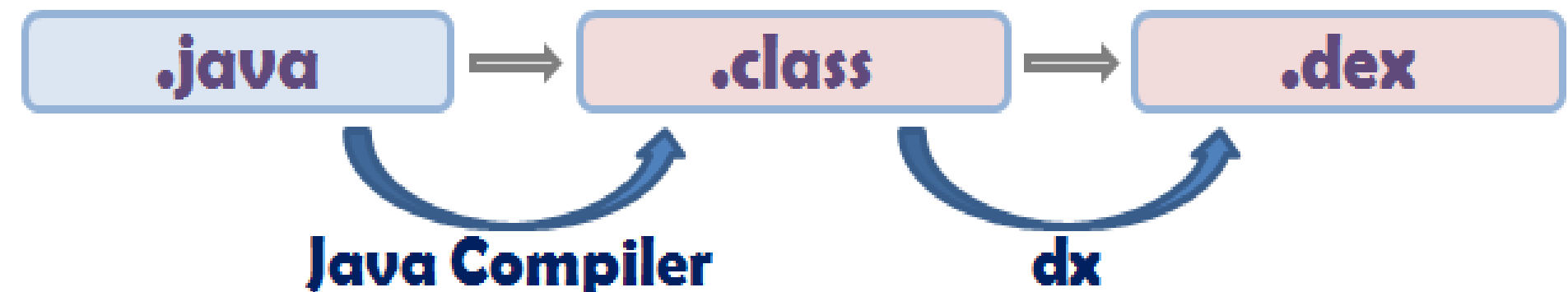


Dalvik Virtual Machine

- ✓ Providing environment on which every Android application runs
 - Each Android application runs in its own process, with its own instance of the Dalvik VM.
 - Dalvik has been written such that a device can run multiple VMs efficiently.
- ✓ Register-based virtual machine

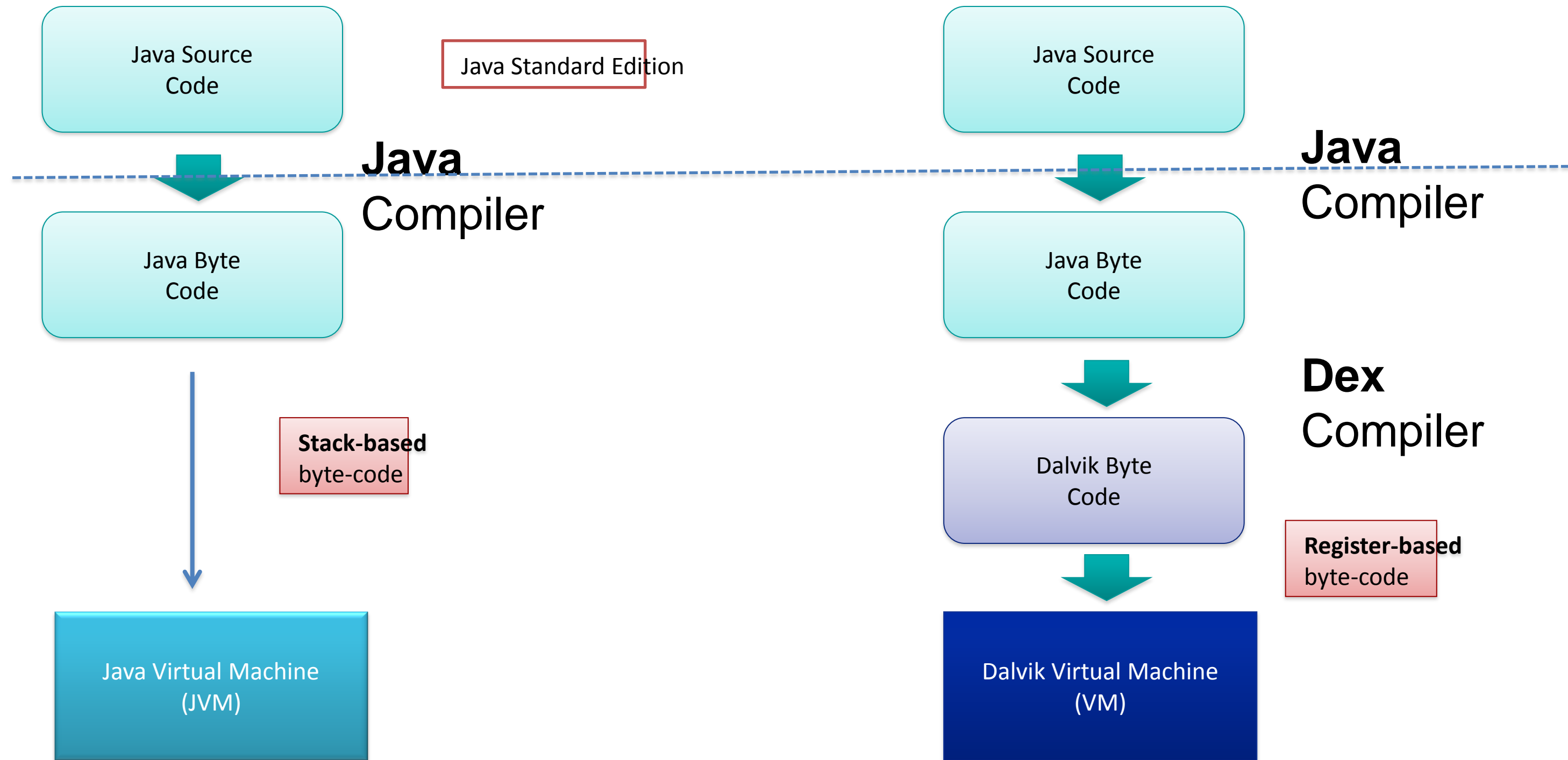
- ✓ Executing the Dalvik Executable (.dex) format
 - .dex format is optimized for minimal memory footprint.
 - Compilation

- ✓ Relying on the Linux Kernel for:
 - Threading
 - Low-level memory management



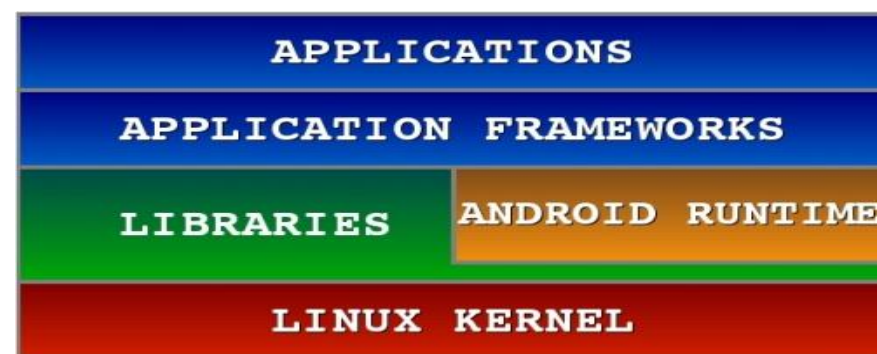


Dalvik Java Virtual Machine (JVM)





Android S/W Stack – Linux Kernel



- Relying on Linux Kernel 2.6 for core system services

- ✓ Memory and Process Management
- ✓ Network Stack
- ✓ Driver Model
- ✓ Security

- Providing an abstraction layer between the H/W and the rest of the S/W stack

Built on top of **Linux kernel**
(v. 2.6-3.0)

Advantages:

- Portability (i.e. easy to compile on different hardware architectures)
- Security (e.g. secure multi-process environment)
- Power Management