Mobile Programming

Introduction to Android

Definition of Smartphone

- Smartphone?
  - A mobile phone that has evolved from a general mobile phone, has performance similar to that of a PC, and supports openness by embedding a high-functioning general-purpose OS
  - Combines functions, such as phone calling, personal information management, data acquisition and processing, etc.

History (1/4)

- 1992 – Simon (IBM)
  - 3” touch screen ($899)
  - Calculator, address book, email, etc.
- 1996 – Nokia 9000 Communicator
  - Intel i386 CPU & 8MB RAM
  - GEOS 3.0 OS installed
- 2000 – Nokia 9210 Communicator
  - The first color TFT display (640 x 200)
  - ARM9 32bit CPU
  - Symbian OS
  - Third-party applications available

History (2/4)

- 1999 – SCH-M100/SPH-M100 (Samsung)
  - Touchscreen
  - Built-in web browser, email, schedule, memo, dictionary, game, etc.
- 2002 – iPAQ (HP)
  - PocketPC OS
  - Various applications available
- 2004 – LC8000 (LG)
  - XScale 400MHz CPU, 192MB RAM, 2.8” TFT
  - 1.1M pixel camera, full touch screen

History (3/4)

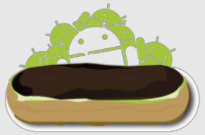
- 2002 – Blackberry 957 (RIM)
  - Blackberry OS
  - Services for enterprises
- 2007 – iPhone (Apple)
  - Removes qwerty keyboard and provides Multi-touch feature
- 2008 – Android-based phones
  - Installed Android OS
  - HTC(2008), Samsung Galaxy(2010), LG Andro-1 (2010)

History (4/4)

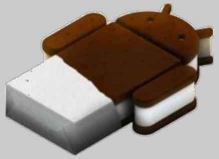
Android

- 2007 Nov: Android SDK release (OHA)
- 2008 Aug: Android market
- 2008 Sep: Android phone release (HTC G1 - T-Mobile)
- 2008 Sep: Android SDK v1.0 (API level 1) release
- 2008 Oct: Android source open
- . . .
- 2014 Jun: Android Wear OS for smartwatch
- 2014 Dec: Android studio 1.0 release
- 2015: Extended to various domains: TV, Auto, etc.
- 2015 ~ : OS & Android studio upgrade
- 2017: Kotlin as official development language

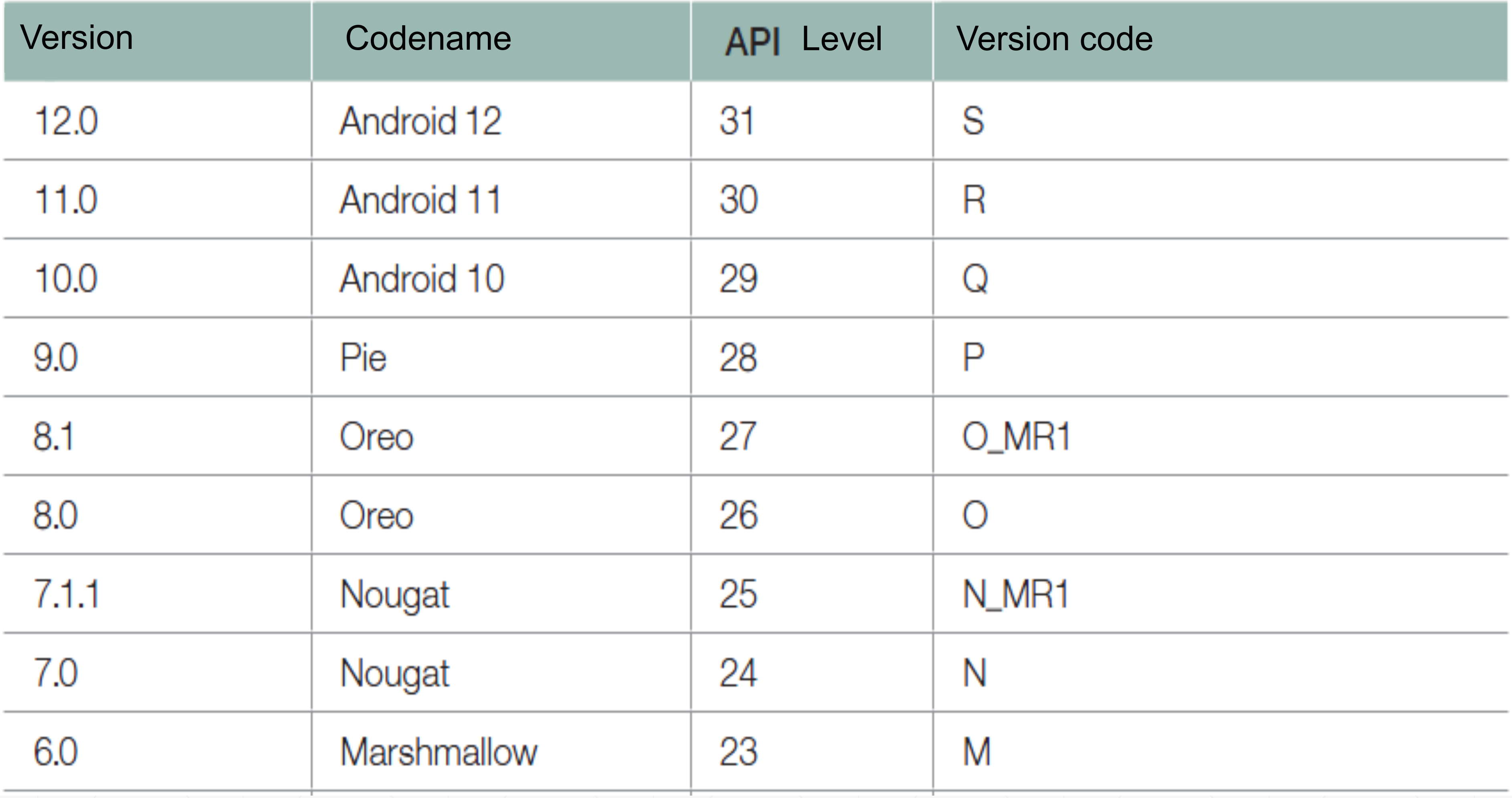
Android Versions (1/3)

Logo	Codename	version	API level	Release	Feature
	Alpha Beta	1.0 1.1	1 2	2008. 9 2009. 2	Official release Bug fix, API change
	Éclair	2.0	7	2010. 1	HW optimization, Bluetooth 2.1
	Froyo	2.2	8	2010. 5	USB tethering, update, flash 10.1
	Gingerbread	2.3	10	2010. 12	UI enhancement, NFC support
	Honeycomb	3.0 3.1 3.2	11 12 13	2011. 2 2011. 5 2011. 7	Tablet PC optimization

Android Versions (2/3)

Logo	Codename	Ver.	API level	Release	Features
	Ice cream Sandwich	4.0	14 15	2011. 10 2011. 11	Smartphone+tablet spuport
	Jelly Bean	4.1 4.2 4.3	16 17 18	2012. 07 2012. 12 2013. 06	Graphic enhancement
	KitKat	4.4	19 20	2013. 10 2014. 06	GPU acceleration, UI change Wearables support
	Lollipop	5.0 5.1	21 22	2014. 11 2015. 03	64bit support, new design, device sharing
	Marshmallow	6.0	23	2015. 10	App permission Scanning fingerprint
	Nougat	7.0 7.1	24 25	2016.06 2016.10	Multi-window, background optimization, 3D acceleration

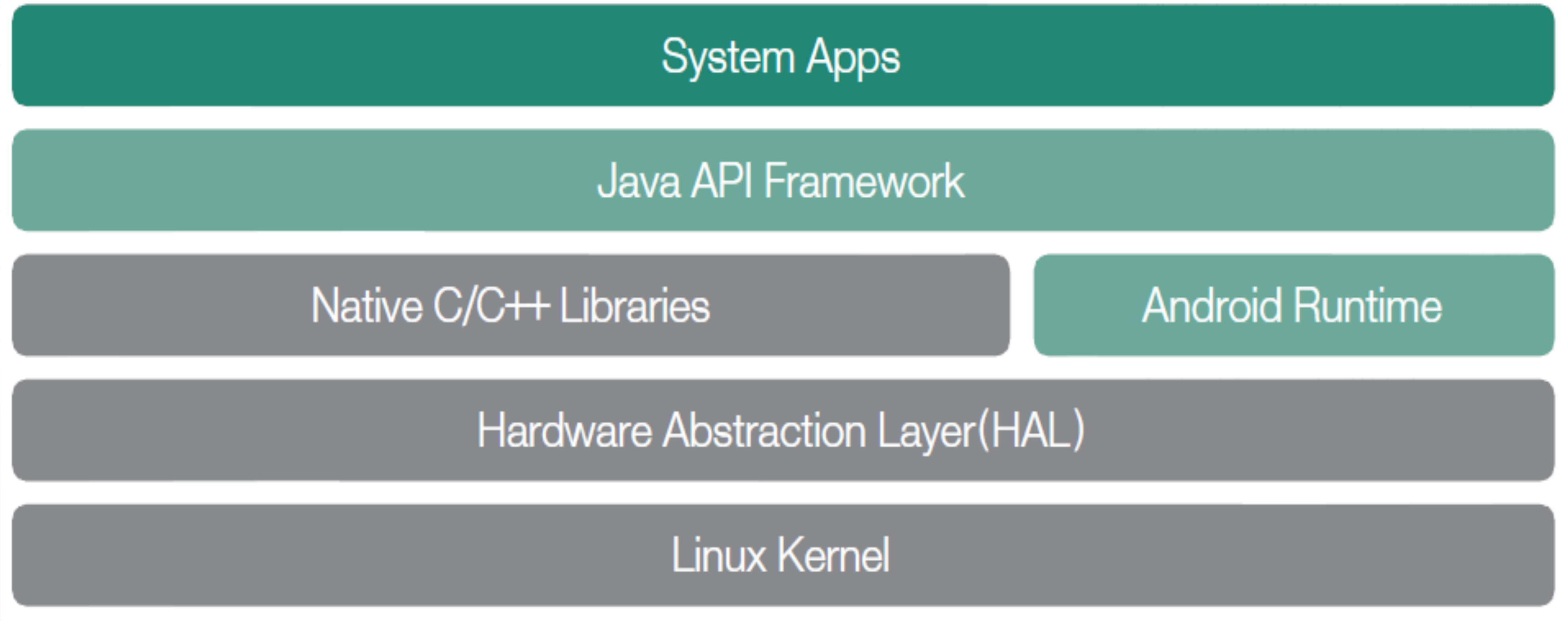
Android Versions (3/3)



Version	Codename	API Level	Version code
12.0	Android 12	31	S
11.0	Android 11	30	R
10.0	Android 10	29	Q
9.0	Pie	28	P
8.1	Oreo	27	O_MR1
8.0	Oreo	26	O
7.1.1	Nougat	25	N_MR1
7.0	Nougat	24	N
6.0	Marshmallow	23	M

Android OS & Platform (1/4)

- Android interacts with Linux OS



Android OS & Platform (2/4)

- Applications
  - General applications running on Android smartphones
  - Web-browser, contacts, game, etc.
  - Developed using Java/Kotlin languages

- Application (API) Framework
  - Provides Android APIs
  - Supports HW access

Android OS & Platform (3/4)

- Android Runtime
  - VM-based approach
  - Previous: Dalvik Virtual Machine (a kind of JVM) + Java core libraries
  - Current: ART (android runtime)

- Libraries
  - GUI, graphics, drawing, database management

- Linux Kernel
  - HW operation, low-level management
  - Memory management, device driver, security, etc.

Android OS & Platform (4/4)

- Development environment

Environment	Language	Toolkit
Application	Java/Kotlin	SDK (Software Development Kit)
System application	C, C++	NDK (Native Development Kit)
Hardware control / kernel	C, C++	PDK (Platform Development Kit)

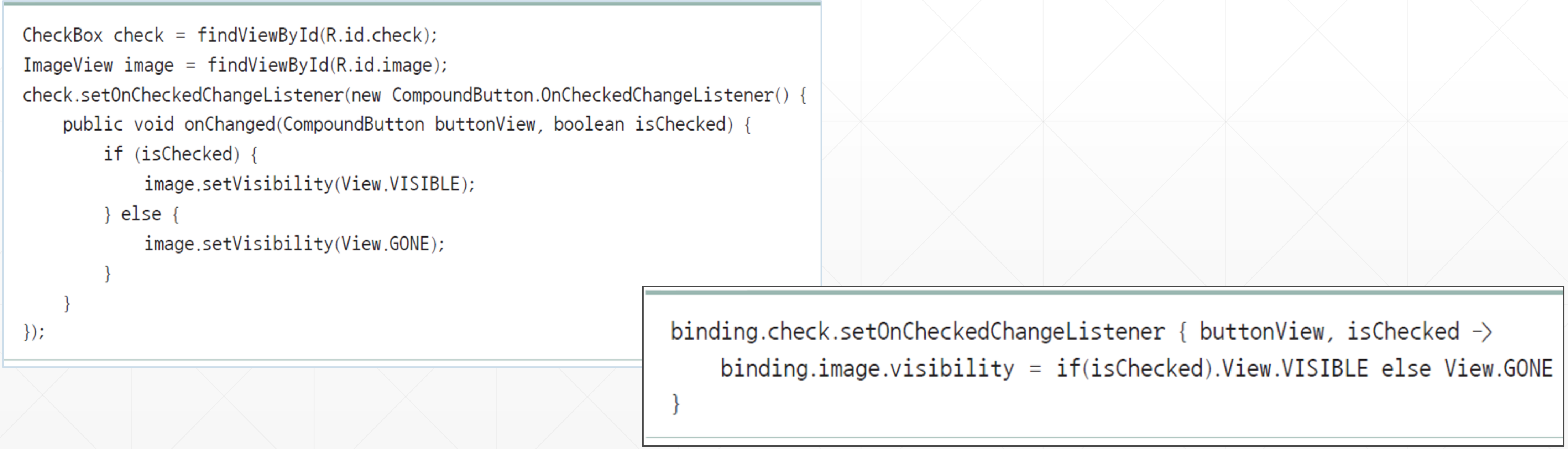
Kotlin (1/2)

- Kotlin
  - Default Programming Language for Android Development

- History
  - 2011: Birthdate of Kotlin
  - 2012: opensource
  - 2016: version 1.0 release
  - 2017: Default language for Android development, by Google
  - 2019: version 1.3.50, and say “Kotlin First!”
  - 2020: version 1.4.20

Kotlin (2/2)

- Kotlin
  - 100% Java compliant
  - Developed by JetBrains (Owner of IntelliJ)
  - Concise syntax

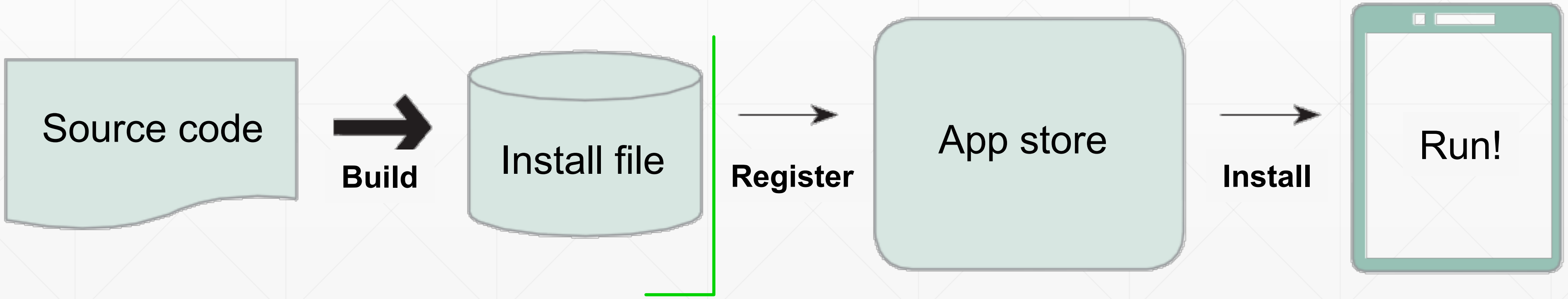


```
CheckBox check = findViewById(R.id.check);
ImageView image = findViewById(R.id.image);
check.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    public void onChanged(CompoundButton buttonView, boolean isChecked) {
        if (isChecked) {
            image.setVisibility(View.VISIBLE);
        } else {
            image.setVisibility(View.GONE);
        }
    }
});
```

```
binding.check.setOnCheckedChangeListener { buttonView, isChecked ->
    binding.image.visibility = if(isChecked).View.VISIBLE else View.GONE
}
```

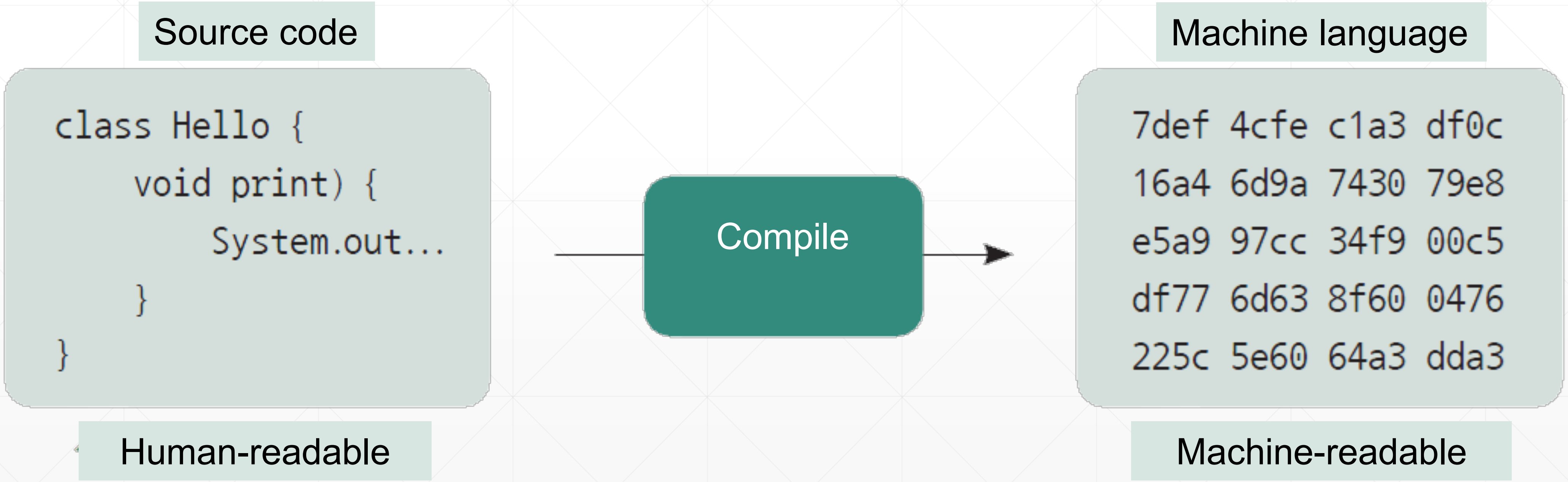
From Code to App

- Write source codes using Kotlin
- Convert your codes into the install file via build procedure
- Upload your app to Google playstore
- Register your app to the store
- Choose and install your app through Google playstore application
- Run your app by touching the icon



From Code to App (1/4)

- Build: convert your source codes into Android executable called APK file
  - Compile
    - Convert your source codes into machine-readable language

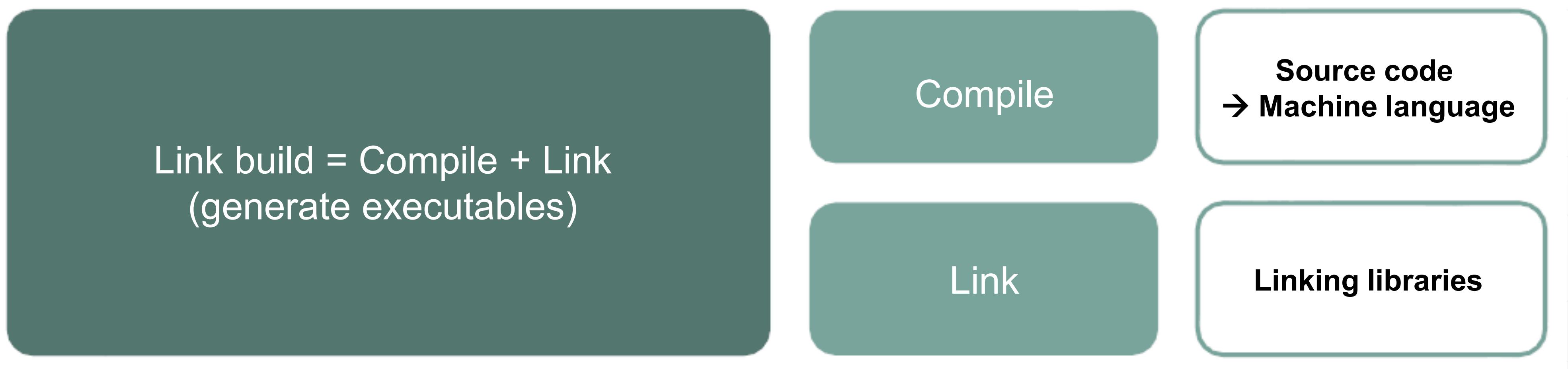


```
class Hello {
    void print) {
        System.out...
    }
}
```

```
7def 4cfe c1a3 df0c
16a4 6d9a 7430 79e8
e5a9 97cc 34f9 00c5
df77 6d63 8f60 0476
225c 5e60 64a3 dda3
```

From Code to App (2/4)

- Build: convert your source codes into Android executable called APK file
  - Build in Linux
    - Translating (compiling) the source code into machine language readable by a computer, and linking it with the library used in the source code to make it into a final executable file



Link build = Compile + Link (generate executables)

Compile

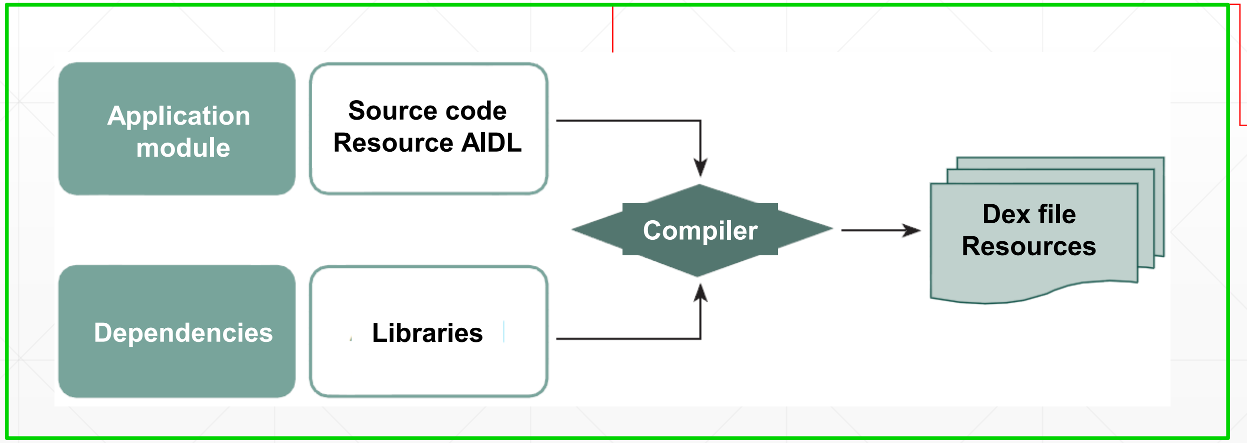
Source code → Machine language

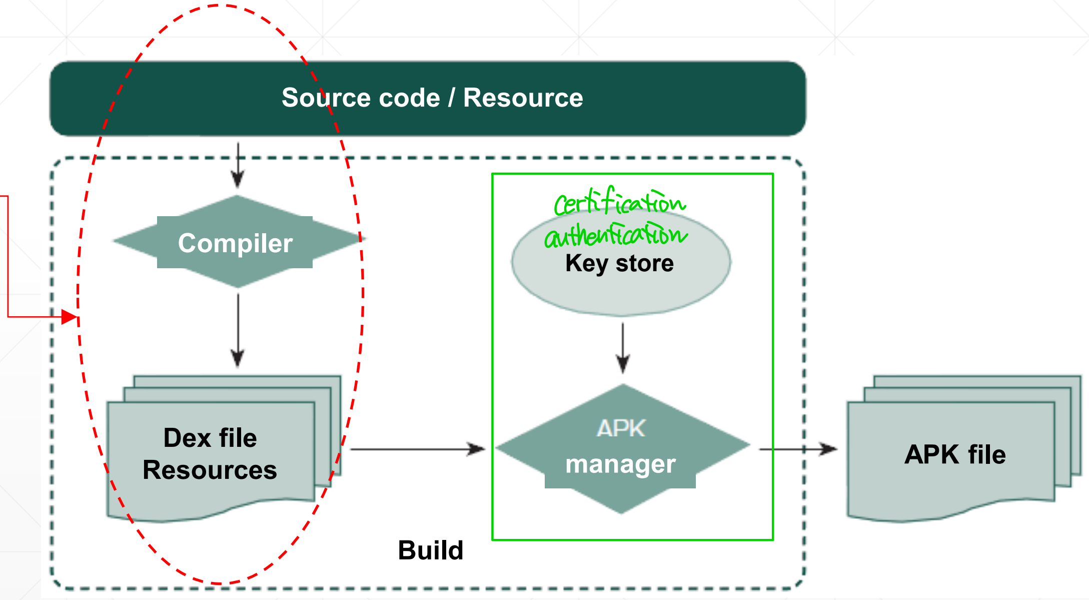
Link

Linking libraries

From Code to App (3/4)

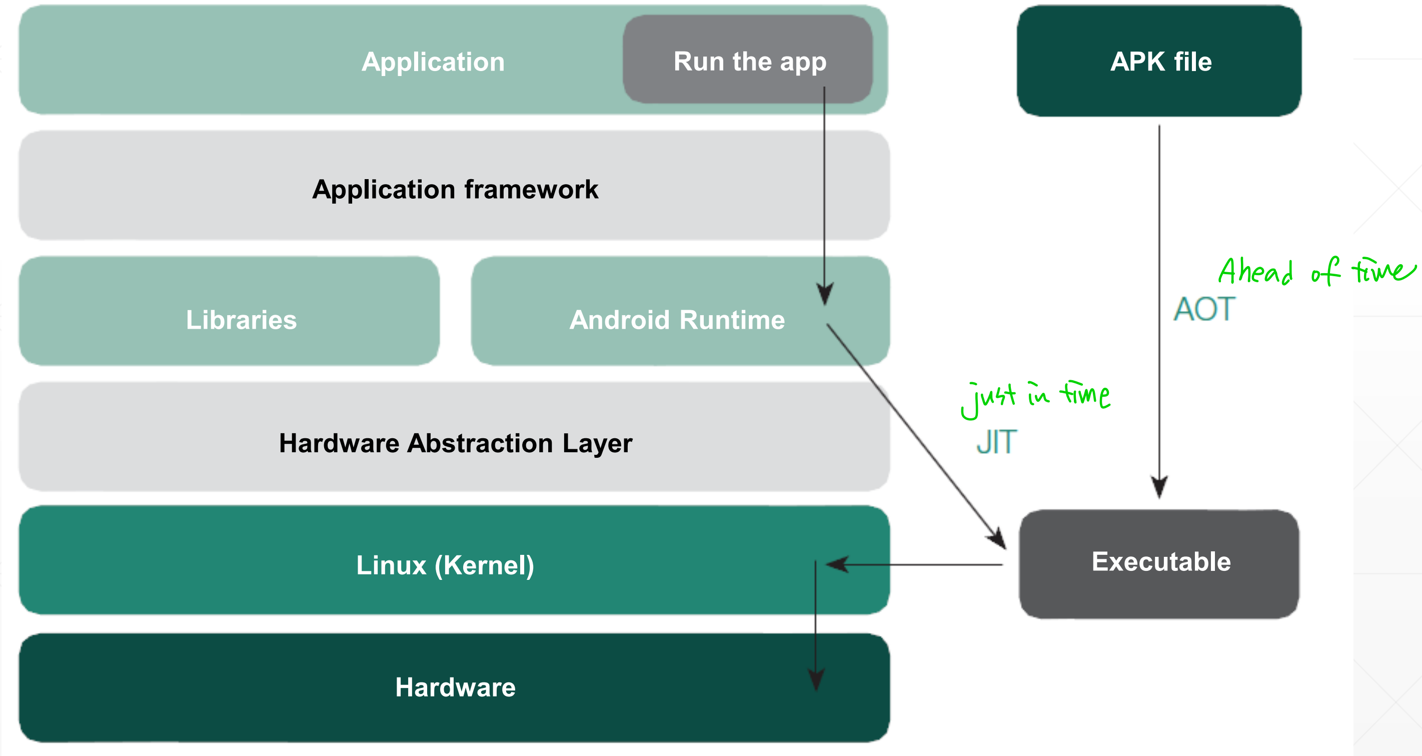
- Build: convert your source codes into Android executable called APK file
  - Build in Android
    - Android contains Resources
      - (Step 1) bytecode step
      - (Step 2) APK step





From Code to App (4/4)

- The installed app runs in application area (top)
- Android controls the developer-created app in the Android runtime area (middle)



Q&A

- Next week (eClass video)
  - Kotlin Basics (Part I&II)