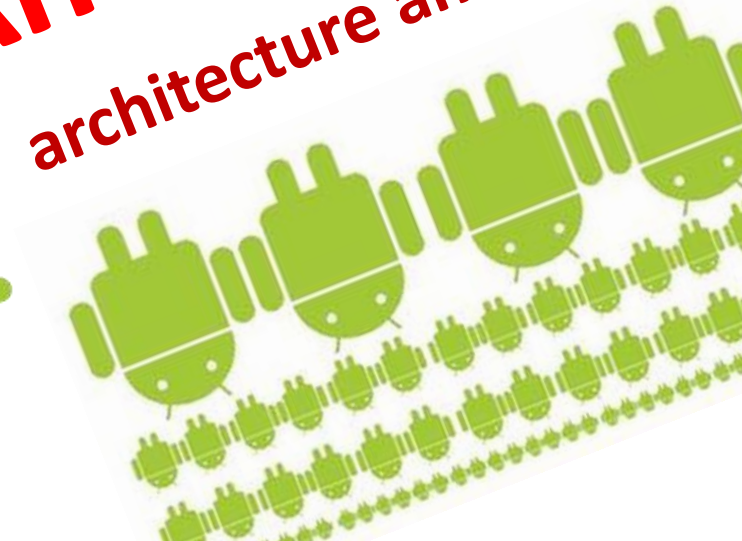


Mobile Application Development

Course 5



Android OS
architecture analysis

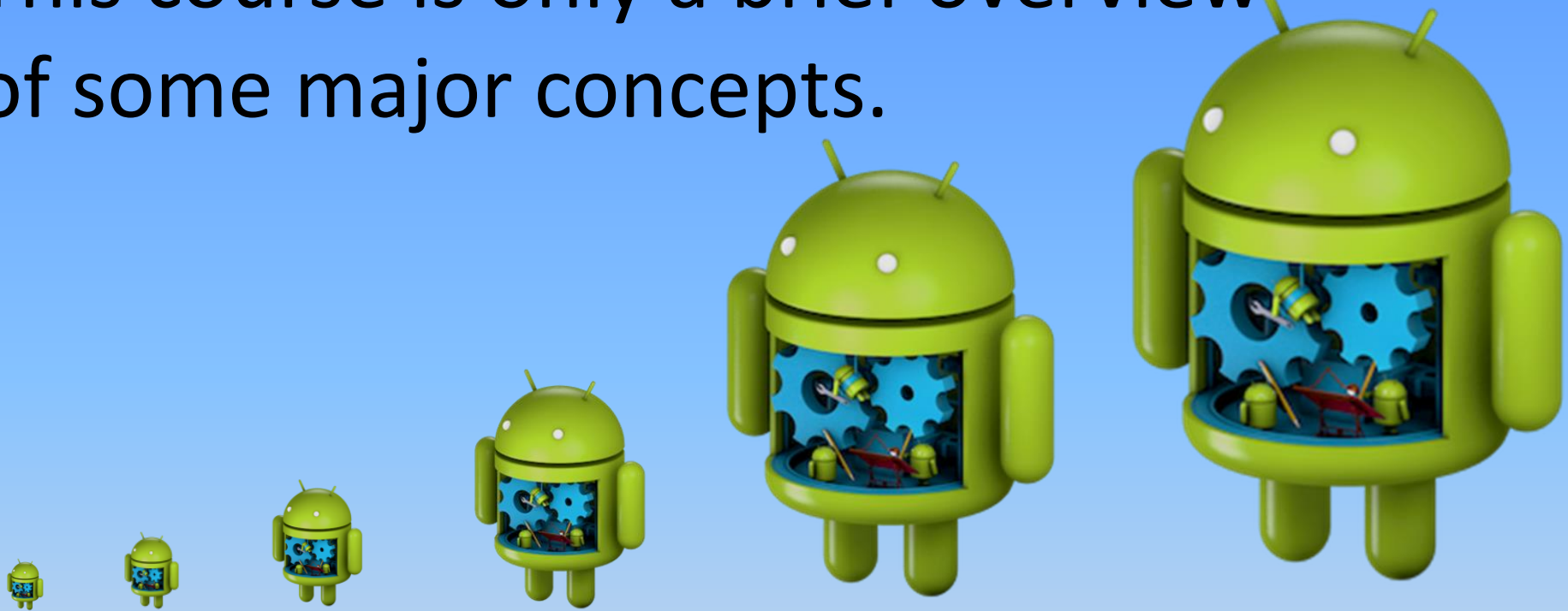


West University of Timisoara, Romania

Computer Science Department

IE3, Fall 2015, Dr. Liviu Octavian Maftciu-Scai

This course is only a brief overview of some major concepts.



Android is much richer and more complex!

- Android is an operating system for small devices (smartphones, tablets etc), that run on battery and have a lot of hardware like GPS, cameras, light and orientation sensors, WiFi and a touchscreen.
- The *Android platform* was created by *Android Inc.* (California US) which was bought by *Google* in 2005 and released as *Android Open Source Project* (AOSP) in 2007.
- Android applications are written in Java and run in a virtual machine: the **Dalvik VM** (virtual machine) which executes it's own byte code.
- In Nov. 2007 a group of 78 companies (Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, Nvidia, Sony Ericsson, Vodafone Group, ARM Holding , Asustek Computer Inc., Toshiba Corp.,....) formed the *Open Handset Alliance* (OHA) that is dedicated to develop and distribute Android OS.

Mobile App Development 5, **Android OS – the layered architecture**



Android applications run in their own sandboxed Dalvik VM and can consist of multiple components: activities, services, broadcast receivers and content providers. Components can interact with other components of the same or a different application via *Intents*.

Frameworks in the Application Framework layer are written in Java and provide abstractions of the underlying native libraries and Dalvik capabilities to applications.

The kernel is a modified Linux 2.6

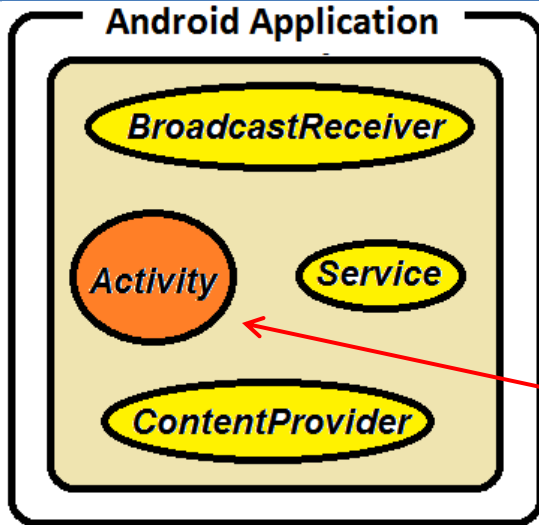
The most libraries for CPU and GPU are developed for low memory consumption.

The Android Runtime consists of the Dalvik virtual machine and the Java core libraries. The Dalvik VM is an interpreter for bytecode that has been transformed from Java bytecode to Dalvik bytecode. The successor of Dalvik is [Android Runtime](#) (ART), which uses the same bytecode and .dex files. The new runtime environment was included for the first time in [Android 4.4 "KitKat"](#) and replaced Dalvik entirely in [Android 5.0 "Lollipop"](#).

Terms *task* and *application* appear to be similar **but**:

Tasks are a series of activities of possibly multiple applications. So, tasks can be considered a logical history of user actions.

Example: An user opens an email application (Yahoo mail) in which he opens a personal mail. In this mail is included a link (www.emag.ro) which is opened in a browser (Chrome). Thus, the task would include two applications (mail and browser). The advantage of the task concept is the opportunity to allow the user to go forth or back, step by step, like a push or pop operation on a stack.



- Applications can start other applications (or components of these) by sending an **Intent**.
- The **IntentManager** solves incoming Intents and starts the proper application or component.

Note: at least an Activity
(to present a GUI)

- Services and Broadcast receivers allow applications to perform jobs in the background and provide additional functionality to other components.
- Broadcast receivers can only run a short time whereas a Service may run a long time.

The compiled code of the application components and additional resources, like libraries, images and other necessary data is packed into a single .apk file that forms the executable Android application.

AndroidManifest.xml

- All Android app. need to have a XML document in the application's root directory called AndroidManifest.xml.
- AndroidManifest.xml presents important information to the Android OS, before OS run any of the app's code (app's name, components, permissions, needed libraries, declares the minimum level of the Android API that the application requires etc).

Activities

- An Activity is a single screen of an Android application.
- An application can have multiple activities.
- An Activity is a subclass of class android.app.Activity. The main functions (methods) are onCreate(), onDestroy(), onResume(), onPause() and onRestart() (*details in next course*).

Content Provider

- Main functions: data storage and retrieval in Android apps
- More, a provider can be used to share data between multiple applications.
- Android OS has default providers for e.g. images, videos, contacts and settings (in the android.provider package).

Intents

- An *Intent* is an asynchronously sent message object including the message that should be transported.
- Intents can be filtered* by an application to specify which intents can be processed by the application's components. The list of filters is set in Android.Manifest.xml, thus Android can determine the intents before starting an application.

Background activities

perform some supporting operations in the background, in general without a graphical interface.

- BroadcastReceiver* is preferred for short operations.
- Service* is preferred for long running jobs.
- Both need to be announced in the application manifest file.

Examples:

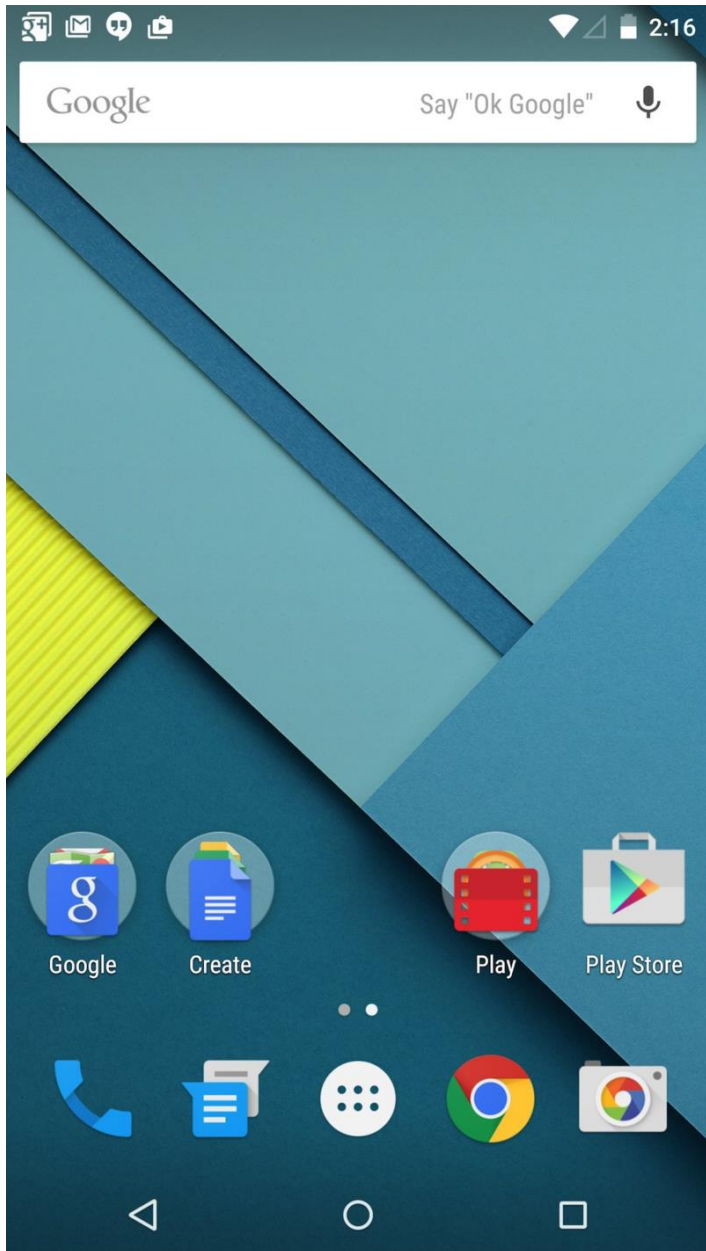
- a *Broadcast* announcing that the screen has turned off, the battery is low, or a picture was captured;
- a *Service* might play music or perform file I/O, all from the background (a *Service* is an application component that perform a longer-running operation while not interacting with the user)

Application security

The security model of Android depends on the multi-user capabilities of the underlying Linux. Each application runs with it's own unique user id and in it's own process. All Android applications run in a sandbox Dalvik VM so that communication with other processes or access to others are avoided.

- Dalvik** is a virtual machine (VM) used in Android OS (in fact part of this) that executes apps written for Android.
 - Dalvik VM** is not compatible to the standard Java VM.
 - Dalvik VM** is specialized and optimized for small systems
- In fact, Dalvik VM is a byte code interpreter. Dalvik has it's own byte code format which is adapted to the needs of Android target devices. The byte code generated is more compact and smaller than standard Java byte code.
- Every Android app runs in a sandboxed environment in it's own Dalvik VM instance.
 - To save storage space, .dex files only contain unique data. If multiple class files share the same string, this string would only exist once in the .dex file and the multiple occurrences are just pointers to this one string. Same for method names, constants and objects which leads to smaller files with much internal "pointing" (on average 50% reduction)

- JEB is an interactive dissambler and decompiler for Android apps.
- JEB decompiles Dalvik bytecode to Java source code (used in BitDefender Comp.-malware)
- JEB provides a full APK view (manifest, resources, certificates, etc.).
- Note: classes and methods are named using letters and numbers.



ago ^[4]	
Latest release	6.0 "Marshmallow" / October 5, 2015
Marketing target	Smartphones , tablet computers , TVs , cars and wearable devices
Available in	70 languages
Package manager	Google Play , APK
Platforms	32- and 64-bit : ARM architectures , x86 , x86-64 , MIPS and MIPS64
Kernel type	Monolithic (modified Linux kernel)
Userland	Bionic libc , mksh shell, native core utilities with a few from NetBSD
Default user interface	Graphical (Multi-touch)
License	Apache License 2.0 GNU GPL v2 for the Linux kernel modifications
Official website	www.android.com



Ta-Ta for now!