Introduction to Android: "Hello, Android!"

26 Mar 2010
CMPT166
Dr. Sean Ho
Trinity Western University

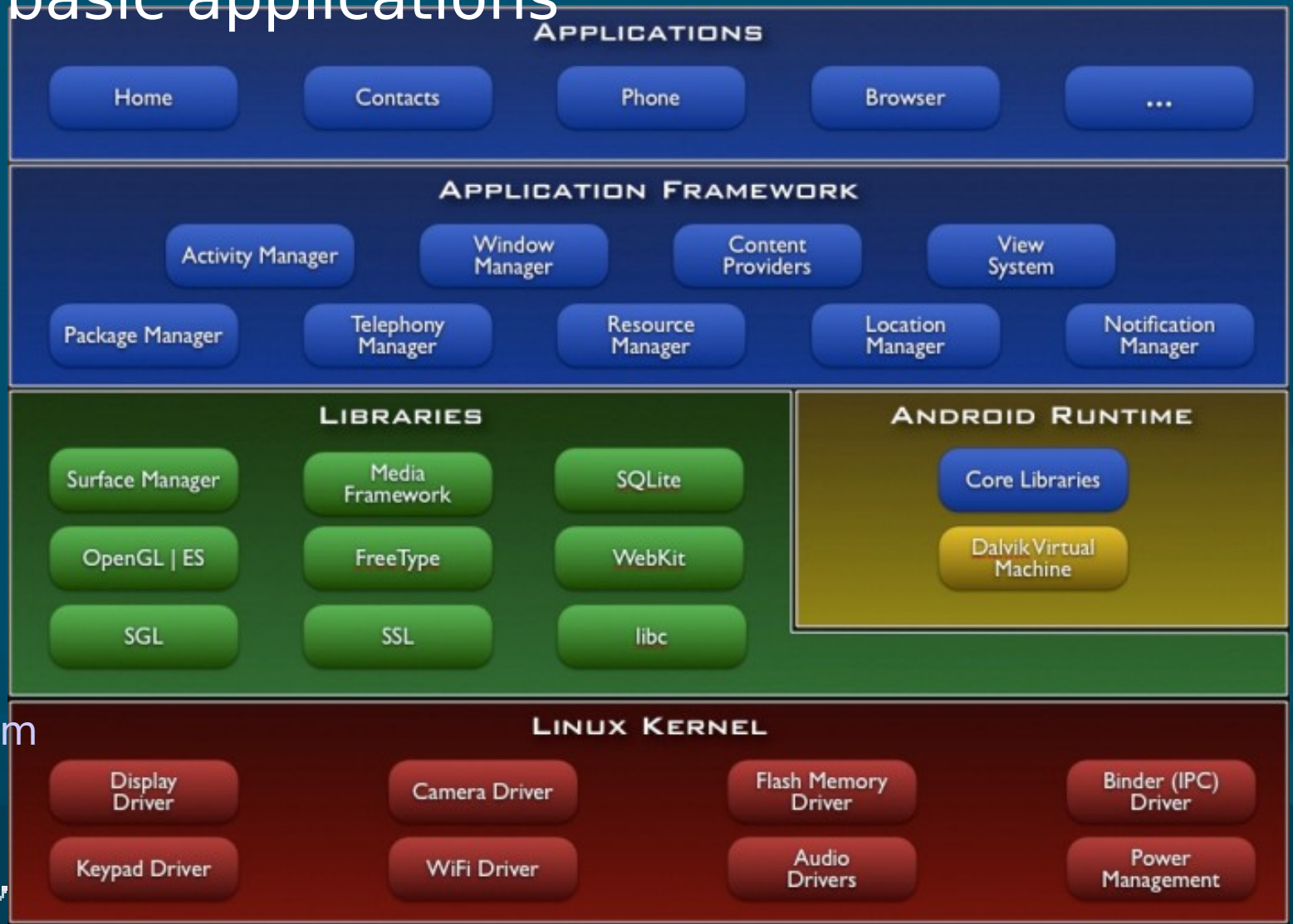
TRINITY WESTERN UNIVERSITY

Android OS

- Open-source mobile OS (mostly Apache licence)
- Developed by Google + Open Handset Alliance
- Linux kernel
- Most apps written in Java, using Android SDK
- Apps run on Dalvik: custom Java VM
- Android Open Source Project: fully open-source
- "Google Experience":
  adds closed-source apps (Maps, Gmail, etc.)
- Hardware drivers are also often closed-source

Android architecture

- Android is: OS, core libraries, “middleware”, plus basic applications



Source: d.android.com

What can you do with Android?

- Component architecture: reuse parts of apps
- Integrate web browser into your app (WebKit)
- Audio, video, images (MPEG4, MP3, PNG, etc.)
- 2D and 3D graphics (OpenGL-ES 1.0/1.1)
- SQLite on-board database
- Telephony (calls, SMS, etc.)
- EDGE/3G, WiFi data, Bluetooth
- Camera, GPS, compass, accelerometer, etc.
- Develop in Eclipse, debug on phone

Getting started with Android

- Eclipse IDE for Java
- Android SDK starter package
- ADT plugin for Eclipse
- From plugin, add Android 1.6 platform
  - Could also develop for 1.5, 2.0, 2.1, etc.
  - Setup an emulator instance(virtual phone)
- Try the “Hello World!” tutorial
  - Run/debug on the emulator
  - Run/debug on actual phone via USB

Android application design

- No single entry point (i.e., main())
  - Instead, subclass an Android class and override certain methods ("hooks")
- Other apps can use parts of your app
  - e.g., use Browser to request a web page
  - e.g., search in Contacts for a phone number
- Android can resume your app if crashed
  - It can also kill your app if out of memory
  - So save/load state and be prepared to die at (nearly) any time

Kinds of Android applications

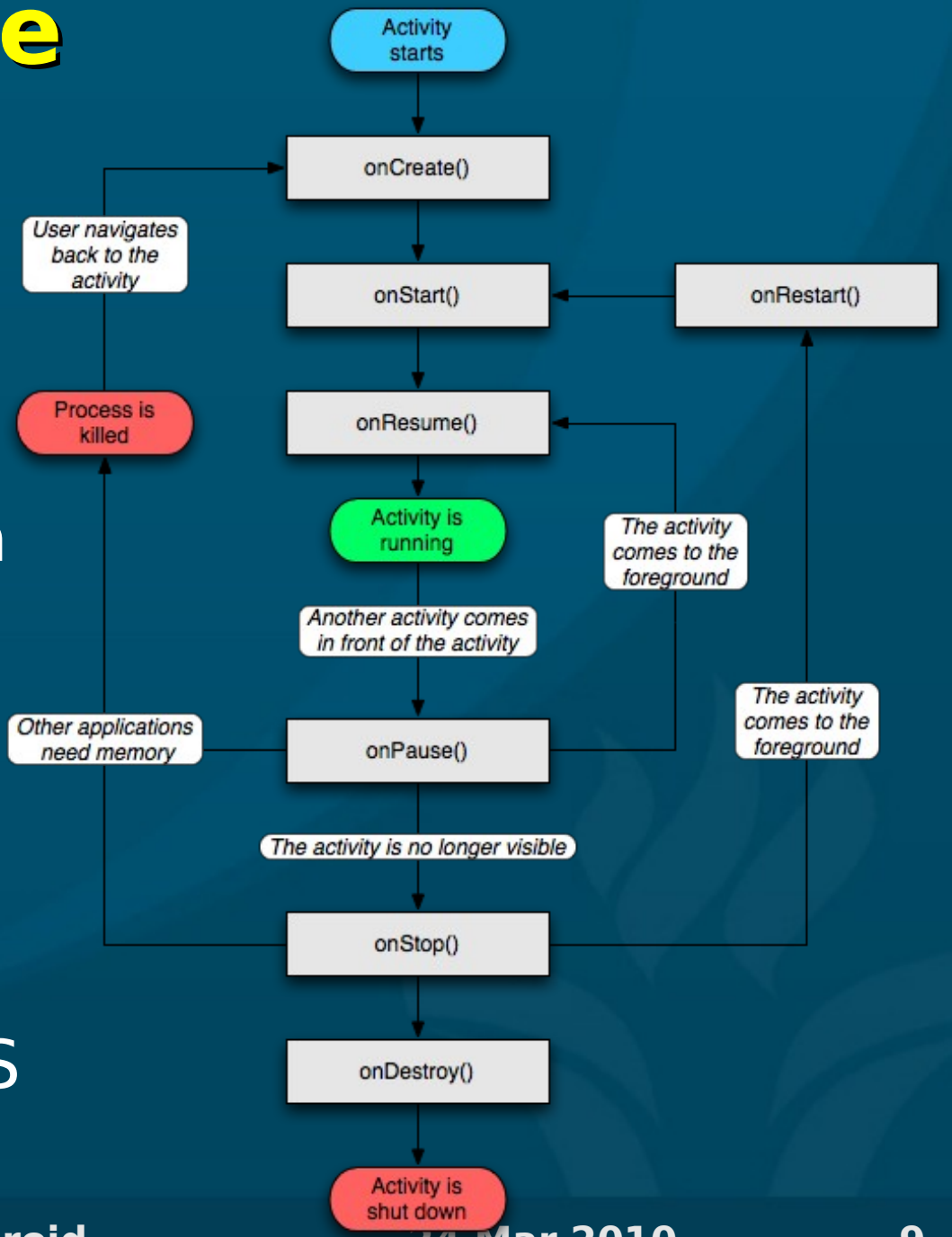
- Activity: present UI for one interactive task
  - e.g.: get username+password, display map
- Service: background task, often w/o UI
  - e.g.: play music, fetch file over network
- Broadcast Receiver: respond to announcements
  - e.g.: if timezone changes, battery low, etc.
- Content Provider: access/query a datastore
  - e.g.,: music library, student database, etc.
- We will focus on Activities, as the simplest kind

Tasks

- Applications are formed from “tasks”: groups of related Activities that can call each other
- Each activity is independent of each other
- A “wizard”-style task might have a sequence of Activities: the “Next” button calls next Activity
- Activity stack tracks history of activities
  - Press hardware “back” button to go back
- The main entry point for your app is specified with an intent: android.intent.action.MAIN and category android.intent.category.LAUNCHER

Activity life cycle

- Four states:
- Active: running and in foreground
- Paused: running, but a dialog has popped-up on top of it
- Stopped: still running, but hidden by others
- Dead: terminated, perhaps by Android OS when low on memory



Life cycle methods

- Activity exists between onCreate/onDestroy():
  - Initial setup and final tear down of resources
- Activity is visible between onStart/onStop():
  - onRestart() also called when return to fore
- In foreground between onResume/onPause():
  - In foreground means accepting user input
  - onPause: commit unsaved changes, etc.
- A paused activity might be destroyed before it ever resumes!

Saving state

- Persistent state should be saved in onPause()
  - e.g. draft of a message being composed
  - Write to storage: preferences, SQL database, app-specific file, or SD card
- Transient state: use onSaveInstanceState()
  - e.g. how user filled out form before “submit”
  - Save in a Bundle, which is passed to both onCreate() and onRestoreInstanceState()
  - Use this, e.g., to fill out the form again when user goes “Back” to this activity

Views (widgets)

- View is Android's widget class (c.f. JComponent)
- Subclasses include: Button, TextView (label), EditText (text area), Spinner (pull-down list), ...
  - Or make your own subclass to customize!
- ViewGroups are layout managers: LinearLayout, GridView, TableLayout, TabHost,..
- In the activity's onCreate(), call setContentView to declare the activity's main View (panel)

“Hello, Android!” tutorial

- Only one activity: HelloAndroid
- onCreate() method for the activity
- Pass the saved state Bundle up to the superclass version of onCreate()
  - Use this to restore any saved state
- Set the main view to “R.layout.main”
  - This is defined in the auto-generated R class
  - Generated by XML layout