

---

# Android Architecture

CS 4720 – Mobile Application Development

# The Basics

---

- In general, all apps are written in Java (this is not always the case as there are third-party conversion tools)
- A compiled Android app is an .apk
- Android apps must be digitally signed in some way to execute
- This digital signature can be a debug certificate that comes default with any installation

# The Basics

---

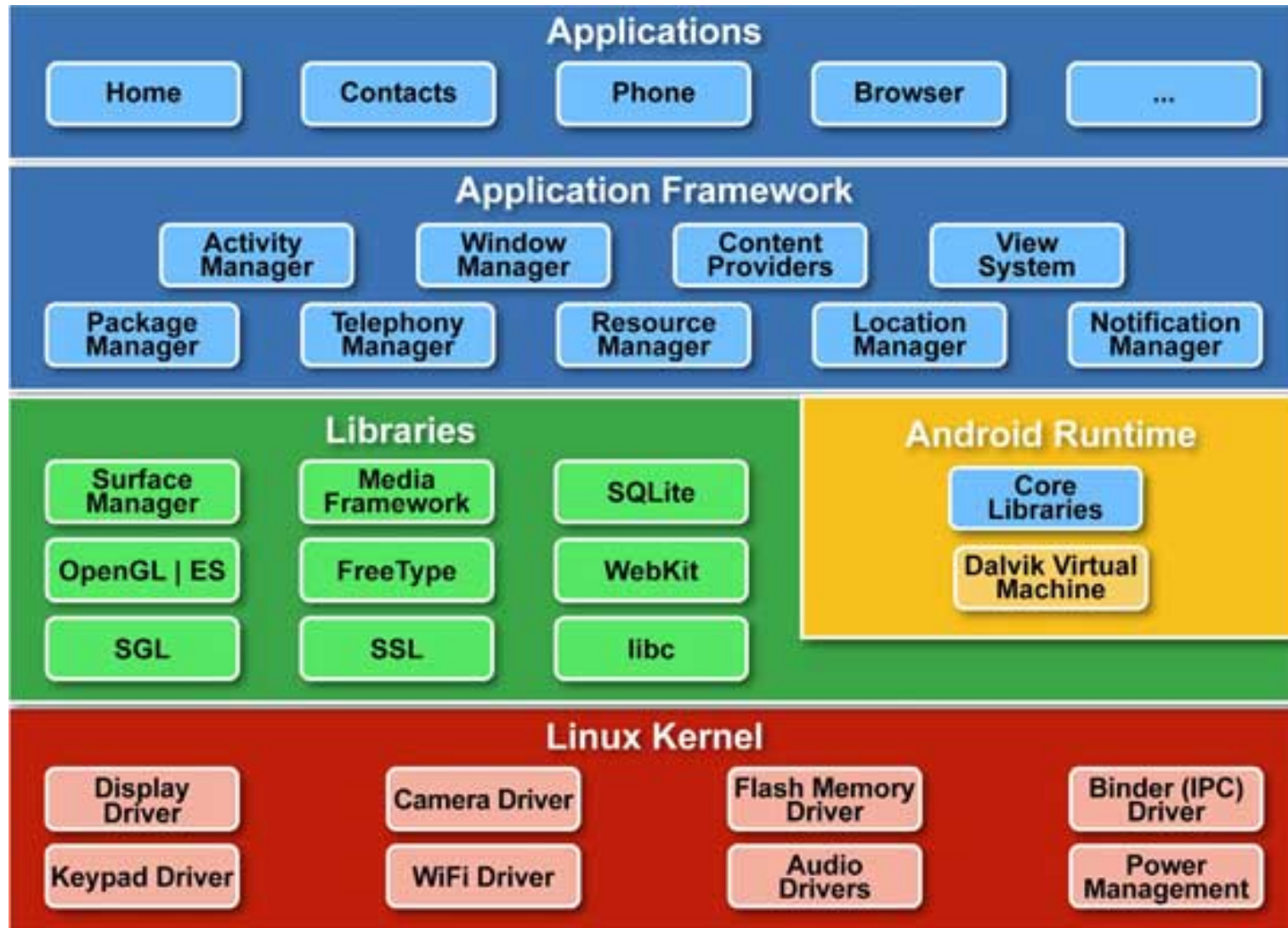
- Android is a multi-user Linux OS
- Each app is given it's own user id when it is installed
- Every app is given it's own "sandbox" where the files are set to permissions only for that app to read and write
- Every app runs in it's own VM (i.e. code runs in isolation from other apps)
- Every app is a separate Linux process

# The Basics

---

- It is possible for apps to share data with other apps
  - 1. Two apps can have the same Linux user id (and thus share resources) if and only if they are signed by the same digital certificate
  - 2. Apps can setup data sharing privileges through the permissions xml system in each manifest

# The Android Architecture



# Main Components

---

- Activities – represent a single screen with a UI
- Services – represents a process running in the background
- Content Provider – a link back to the data
- Broadcast Receiver – listens for system-wide messages to respond to
- Application – a set of Activities that make up a cohesive unit
- Intent – a message to be passed

# Activity

---

- Conceptually, an Activity is a single screen of your application
- In other words, an App really is a collection of related Activities
- Consider each Activity both a screen and a feature
- Apps can activate Activities in other Apps

# Service

---

- A Service is a component that runs in the background to perform long-running operations
- A Service has no UI
- Examples of Services:
  - Playing music in background
  - Gathering GPS data
  - Downloading a data set from the server



# Content Provider

---

- A Content Provider manages a shared set of app data
- This shared set of data could be a file, an SQLite DB, a remote link to a web service, etc.
- Apps can query a Content Provider for data if they have permission
- For example, your App could query the Contacts DB for a set of email addresses
- Content Providers can also be private

# Broadcast Receivers

---

- A Broadcast Receiver responds to system-wide announcements (which are manifested as Intents)
- System status information is delivered this way (i.e. device turned on side, screen off, low battery, phone call incoming, etc.)
- Broadcast Receivers typically don't have a UI, but could have a status bar icon

# Connected Apps

---

- Due to the component nature of Apps (made up of Activities, Services, etc.), it is easy to build features of your App using existing system components
- For example, if your App needs to take a picture, you can query the Camera Activity to handle that request and return the resulting image
- This is handled through Intents

# Intent

---

- An Intent is a message that requests an action from another component of the system
- This includes the “please start up your App” Intent that the system sends when a user clicks on your App icon

# Tying it all Together

---

- If an App is made up of all these disparate parts, what holds them all together?
- The AndroidManifest.xml file!
  - Sets up all permissions the user has to agree to (i.e. Internet, GPS, contacts, etc.)
  - Declares the API level of the App
  - Requests hardware features needed
  - Needed libraries
  - Which Activities are part of this App

# What about the other stuff?

---

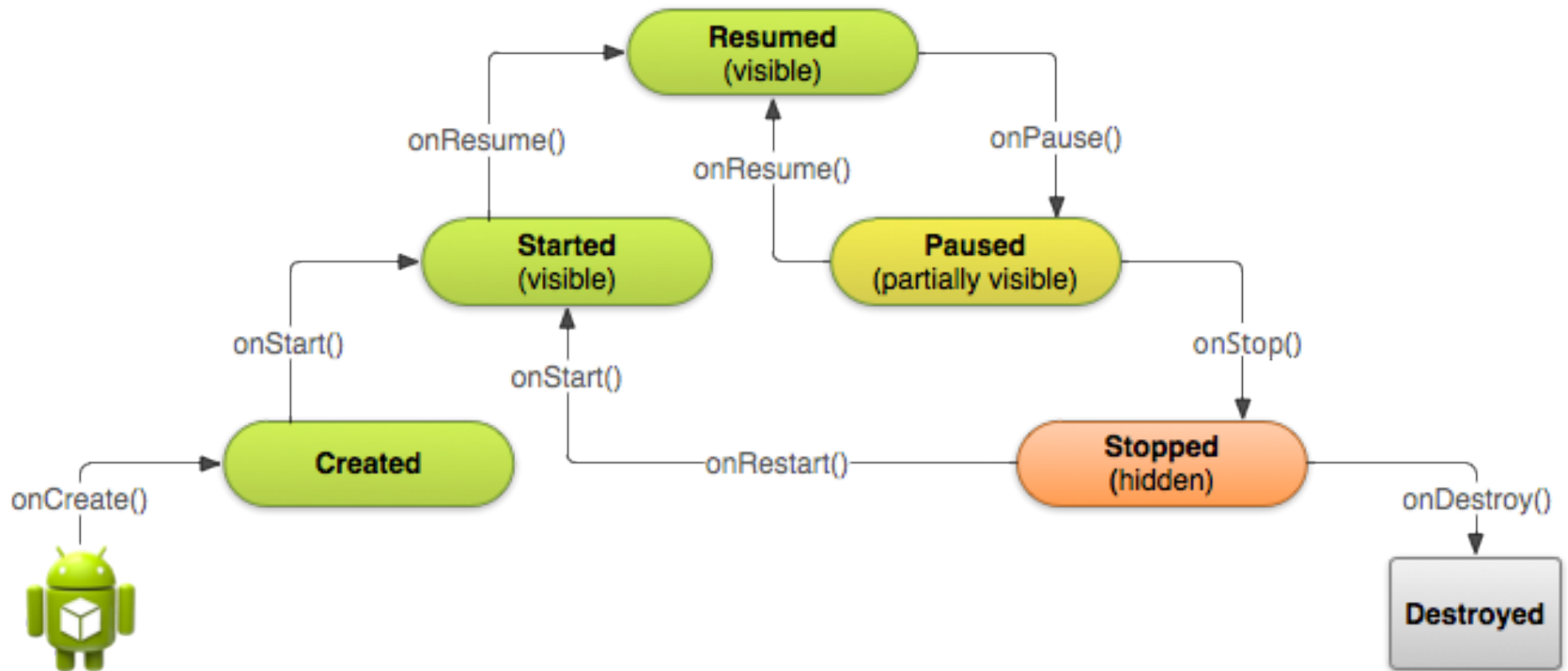
- Typically referred to as “assets,” anything that isn’t code is placed in the res/ folder
- Music
- Images
- Some static data files

# Where's the UI?

---

- The User Interface for an Android App is defined in the layout xml files
- Each layout xml file should correspond to an Activity

# The App Lifecycle





# Looking at Example Code

---

- Android examples are available in GitHub
- These examples are easily downloaded directly from Android Studio