Overview of Android (Part 2): Middleware Infrastructure

Douglas C. Schmidt
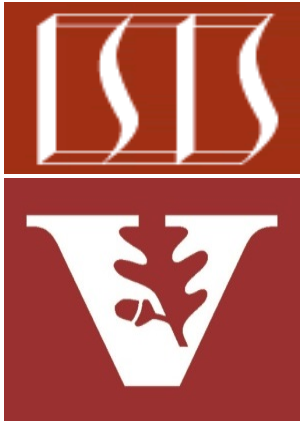
d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt

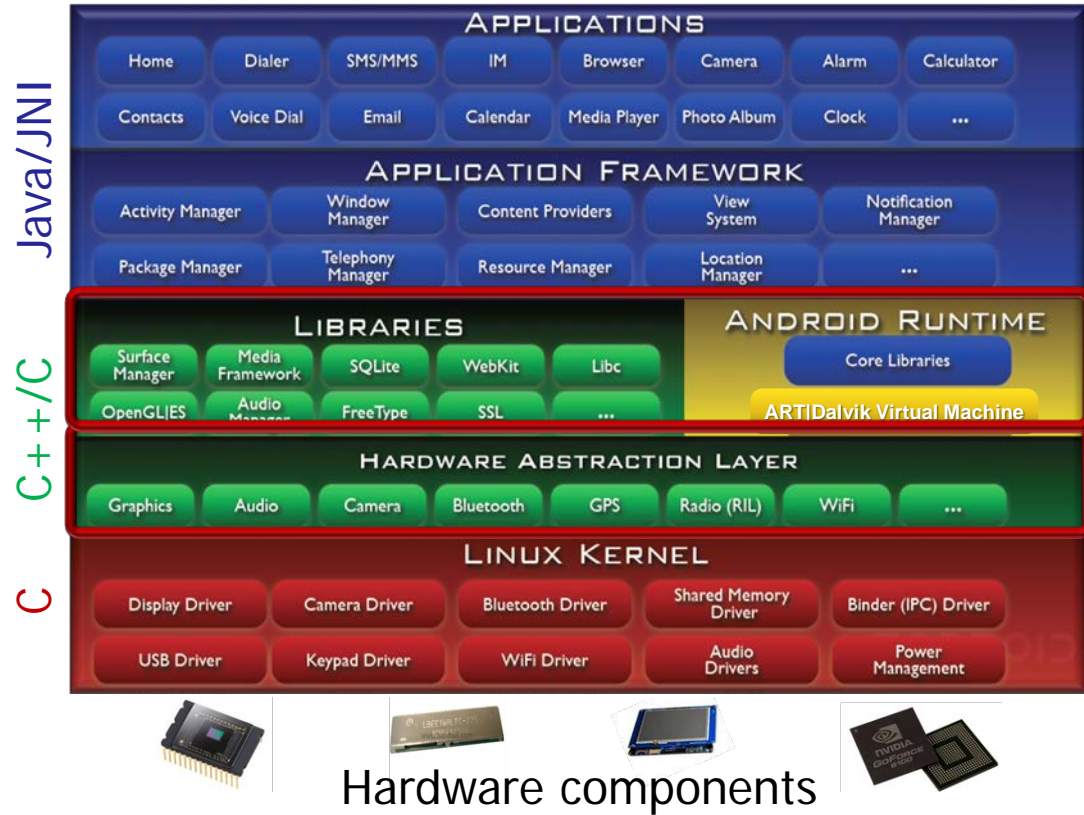**Professor of Computer Science**

Institute for Software Integrated Systems

Vanderbilt University Nashville, Tennessee, USA

Learning Objectives in this Part of the Lesson
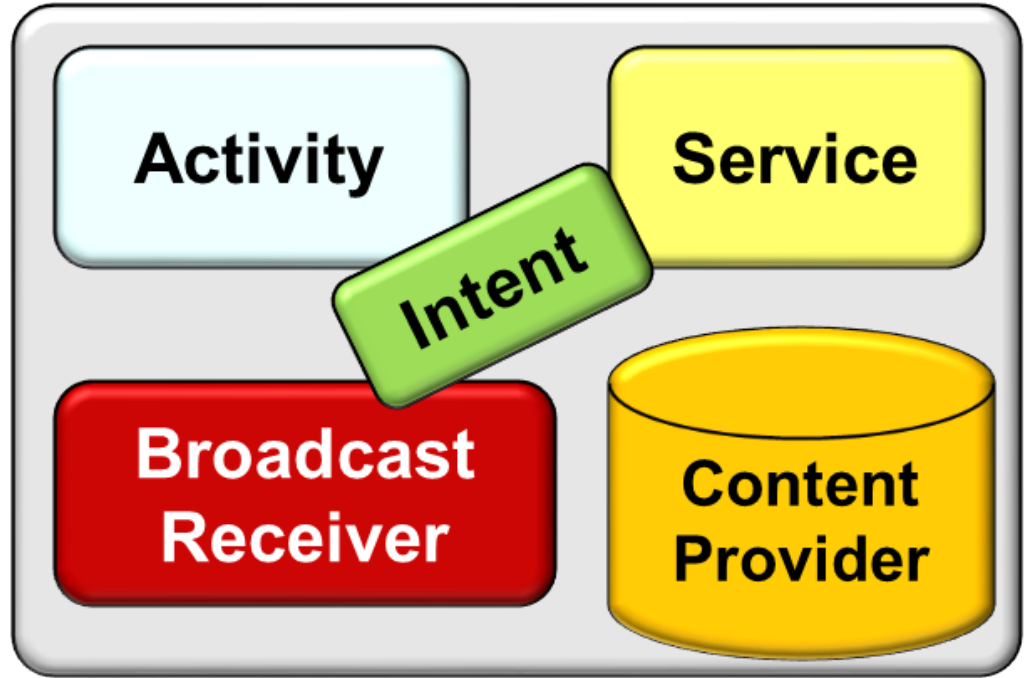
1. Understand key elements in Android's middleware infrastructure

   - e.g., hardware abstraction layer, Android runtime, & native libraries

Java/JNI

C++/C

C

Hardware components

Middleware infrastructure resides atop the OS & below the apps et al

Learning Objectives in this Part of the Lesson

1. Understand key elements in Android's middleware infrastructure

2. Name all the key app components in Android
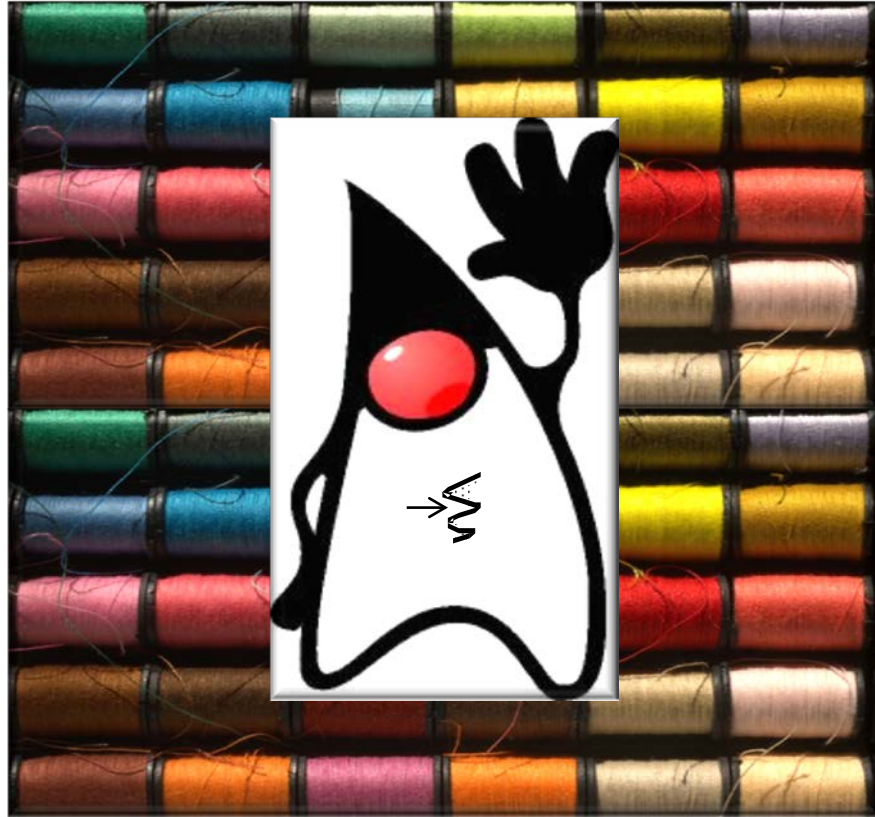
Learning Objectives in this Part of the Lesson

1. Understand key elements in Android's middleware infrastructure

2. Name all the key app components in Android

3. Know what Java threads are in the context of Android

Overview of Android's Middleware Infrastructure

Overview of Android's Middleware Infrastructure

- Android's *middleware infrastructure* provides reusable capabilities that extend hardware-centric OS kernel & protocol mechanisms
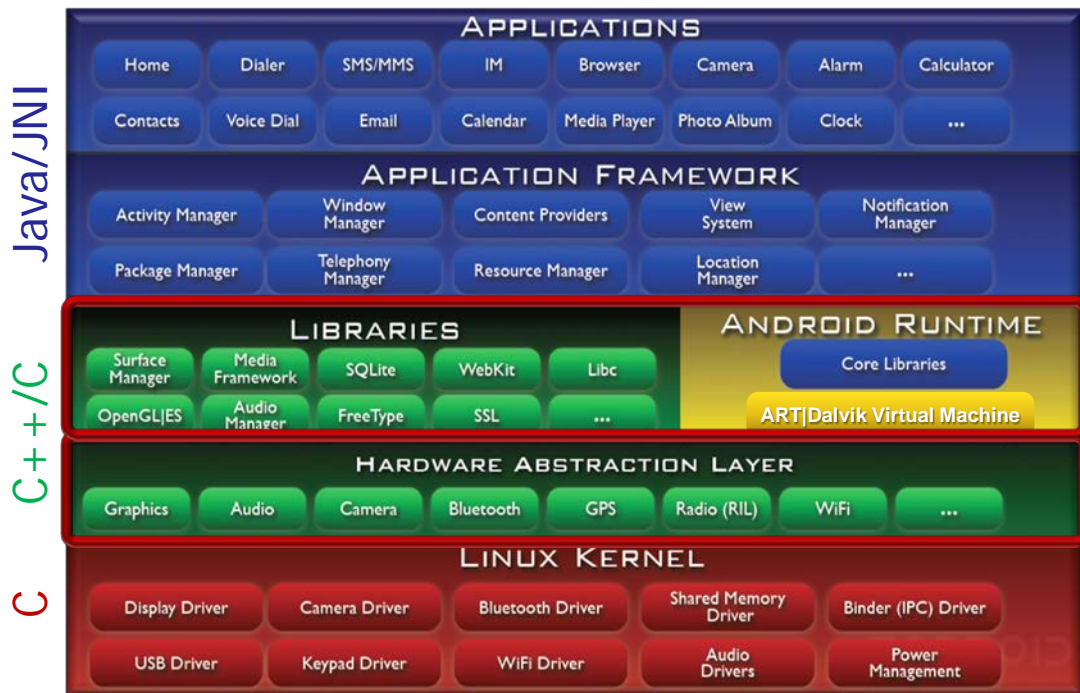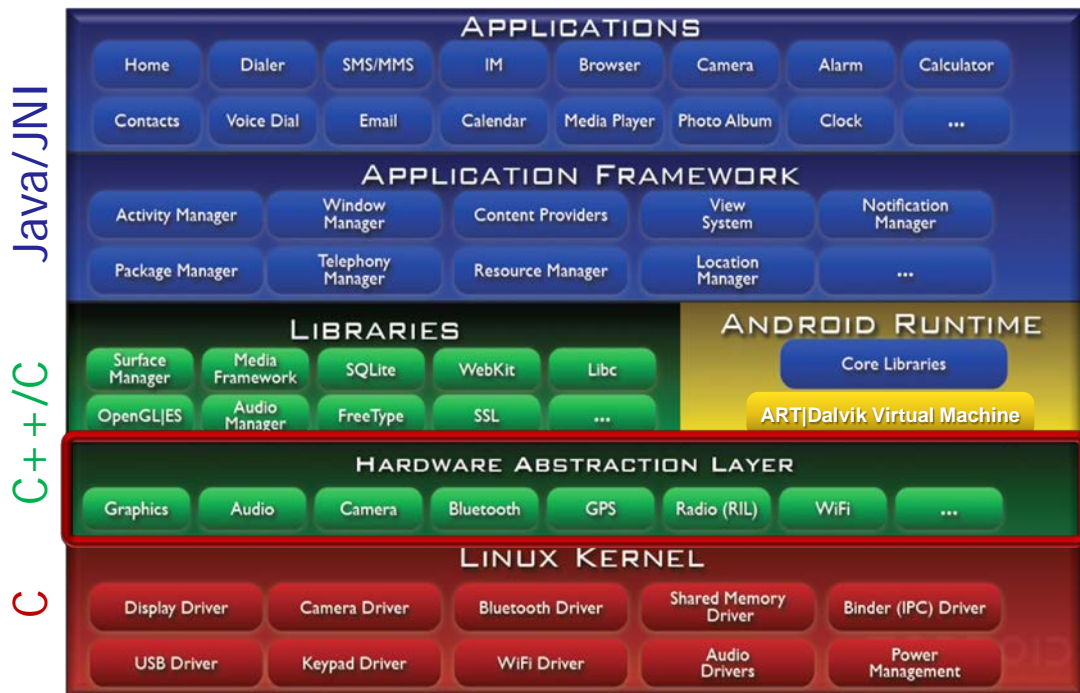
Overview of Android's Middleware Infrastructure

- Android's *middleware infrastructure* provides reusable capabilities that extend hardware-centric OS kernel & protocol mechanisms

  - Hardware abstraction layer

See source.android.com/devices/#Hardware Abstraction Layer

Overview of Android's Middleware Infrastructure

- Android's *middleware infrastructure* provides reusable capabilities that extend hardware-centric OS kernel & protocol mechanisms

  - Hardware abstraction layer

    - Shields Android stack from low-level hardware details

Hardware components

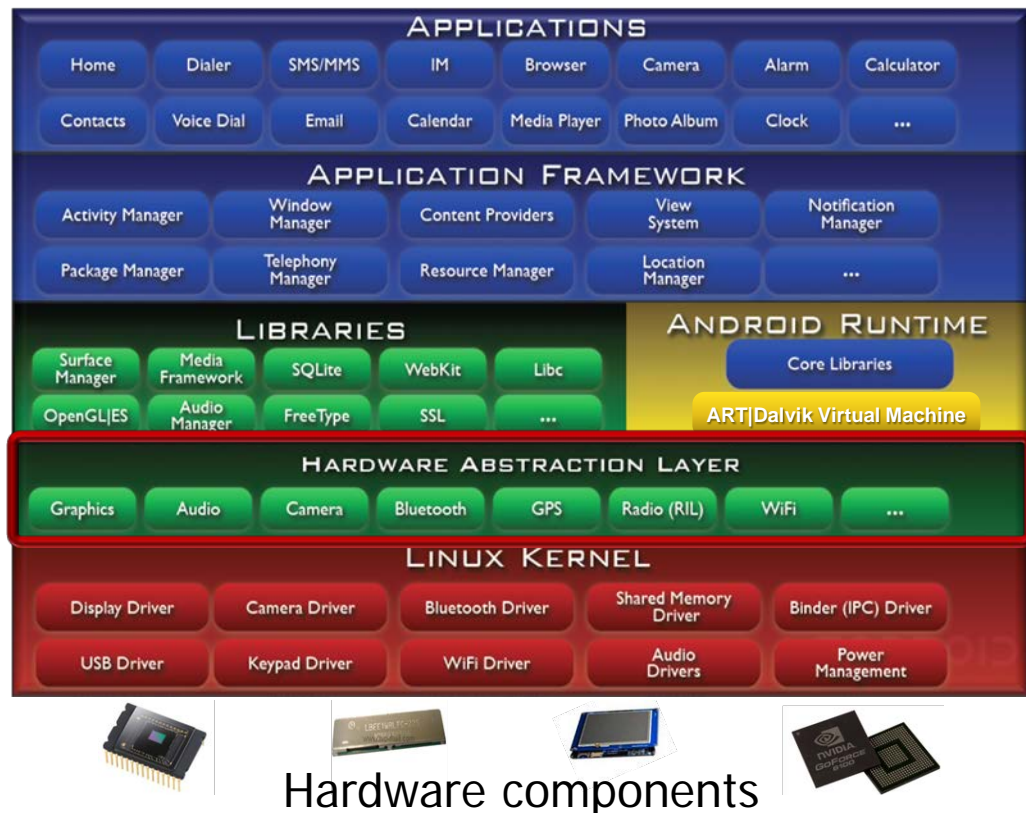# Overview of Android's Middleware Infrastructure

- Android's *middleware infrastructure* provides reusable capabilities that extend hardware-centric OS kernel & protocol mechanisms

  - Hardware abstraction layer
    - Shields Android stack from low-level hardware details
  - Shields OEMs from GNU Public License "virality"

Hardware components

See en.wikipedia.org/wiki/Linux_kernel#Licensing_terms
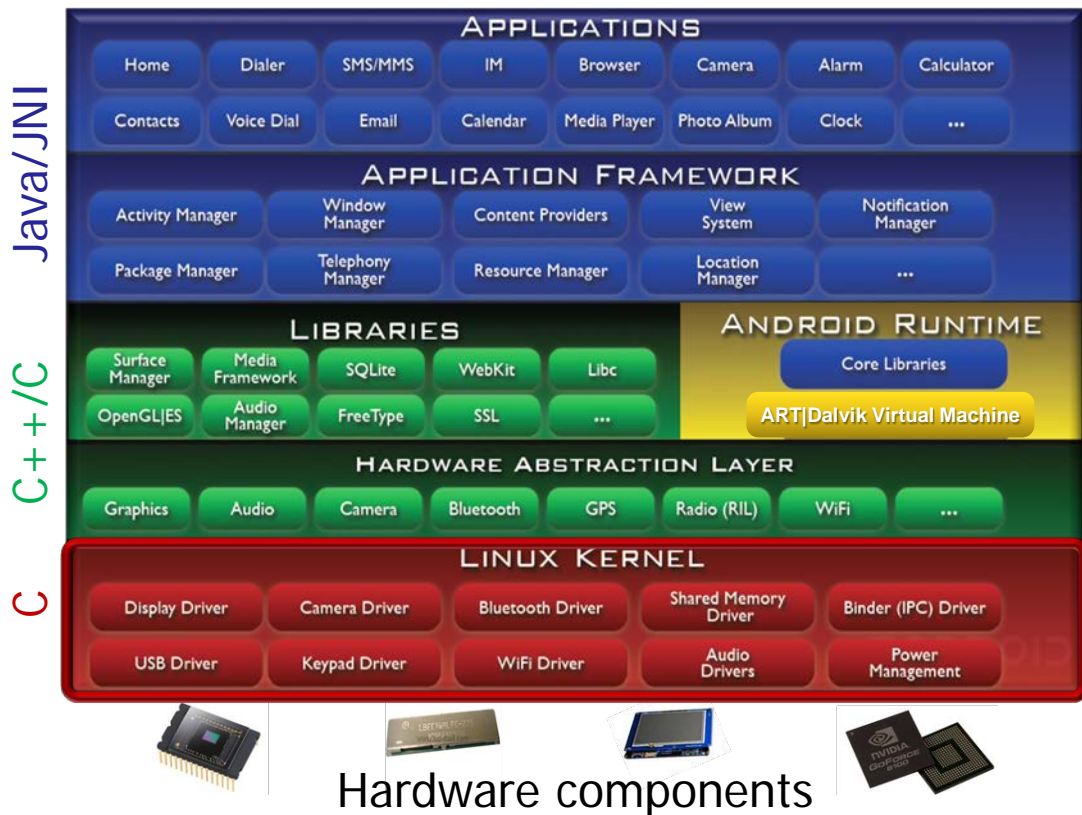
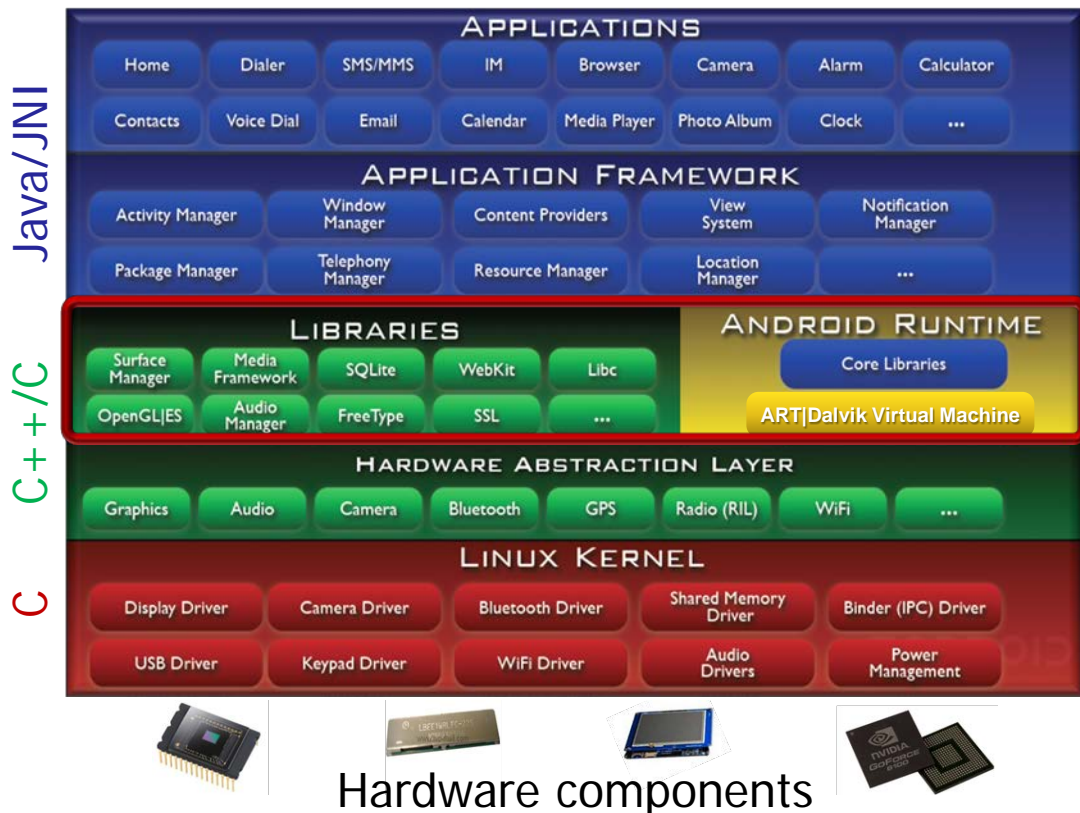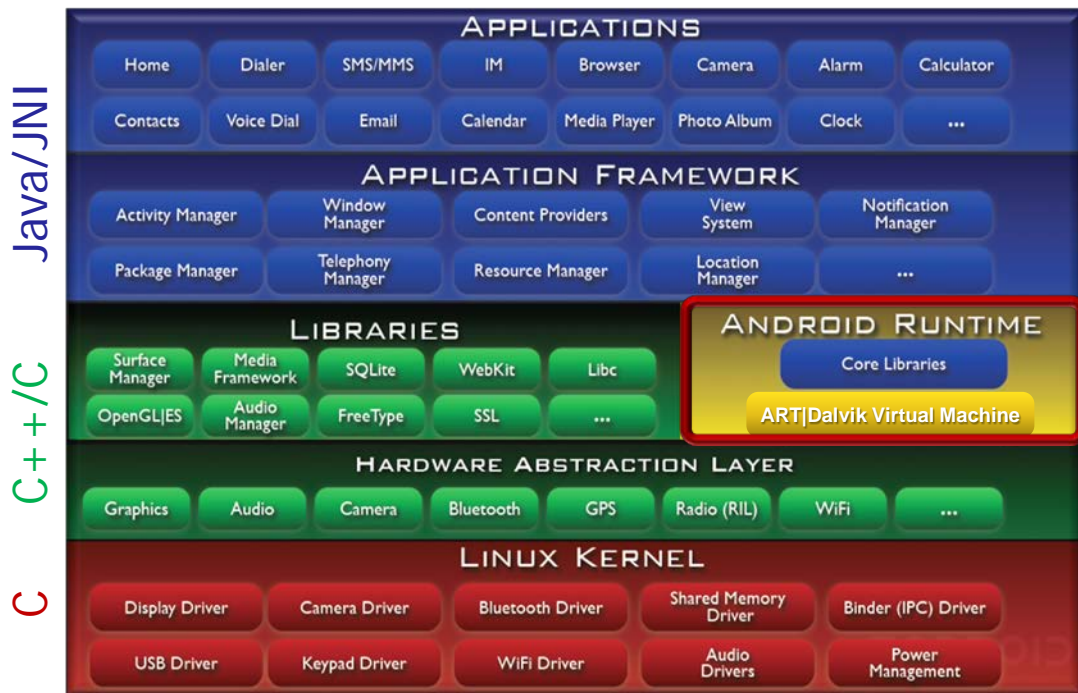# Overview of Android's Middleware Infrastructure

- Android's *middleware infrastructure* provides reusable capabilities that extend hardware-centric OS kernel & protocol mechanisms

  - Hardware abstraction layer

  - Runtime & libraries layer

Java/JNI

C++/C

C

APPLICATIONS

| Home | Dialer | SMS/MMS | IM | Browser | Camera | Alarm | Calculator |
| Contacts | Voice Dial | Email | Calendar | Media Player | Photo Album | Clock | ... |

APPLICATION FRAMEWORK

| Activity Manager | Window Manager | Content Providers | View System | Notification Manager |
| Package Manager | Telephony Manager | Resource Manager | Location Manager | ... |

LIBRARIES

| Surface Manager | Media Framework | SQLite | WebKit | Libc |
| OpenGL|ES | Audio Manager | FreeType | SSL | ... |

ANDROID RUNTIME

Core Libraries

ART|Dalvik Virtual Machine

HARDWARE ABSTRACTION LAYER

| Graphics | Audio | Camera | Bluetooth | GPS | Radio (RIL) | WiFi | ... |

LINUX KERNEL

| Display Driver | Camera Driver | Bluetooth Driver | Shared Memory Driver | Binder (IPC) Driver |
| USB Driver | Keypad Driver | WiFi Driver | Audio Drivers | Power Management |

Hardware components

This layer is composed of several middleware elements

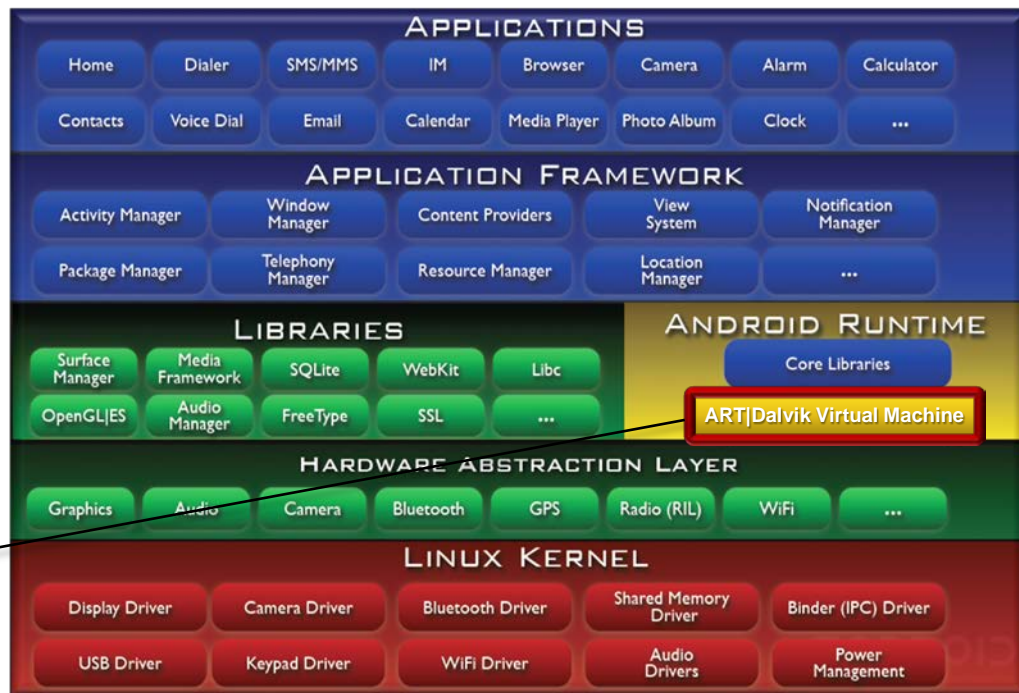# Overview of Android's Middleware Infrastructure

- Android's *middleware infrastructure* provides reusable capabilities that extend hardware-centric OS kernel & protocol mechanisms

  - Hardware abstraction layer

  - Runtime & libraries layer

    - Android runtime
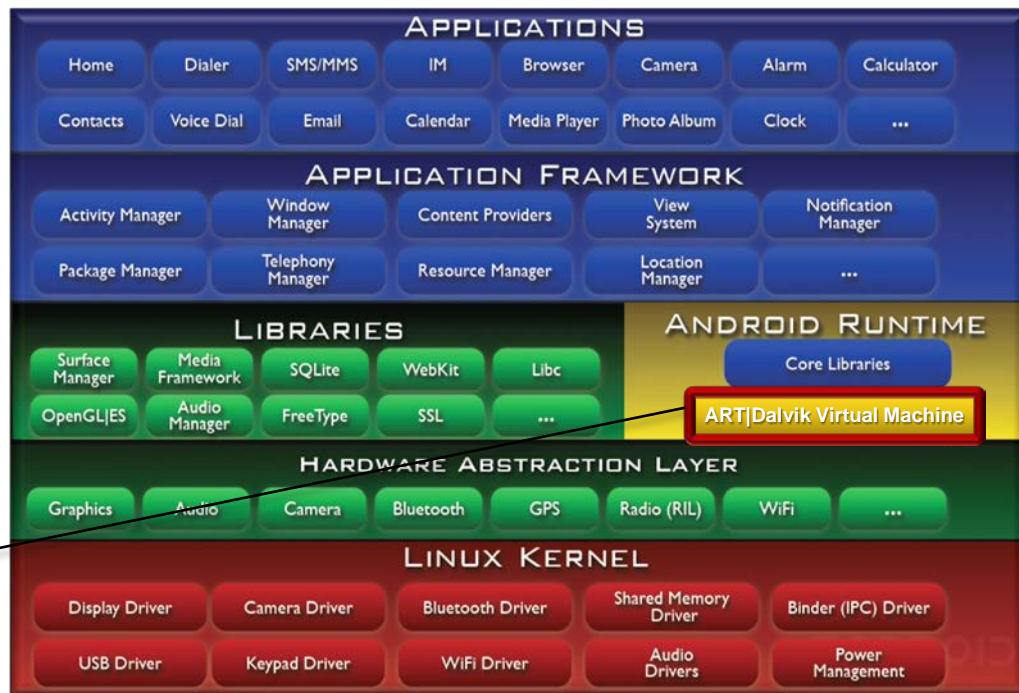

This element is composed of two parts

Overview of Android's Middleware Infrastructure

- Android's *middleware infrastructure* provides reusable capabilities that extend hardware-centric OS kernel & protocol mechanisms

  - Hardware abstraction layer

  - Runtime & libraries layer

    - Android runtime

A managed execution environment that efficiently runs Java-based apps & some Android system services

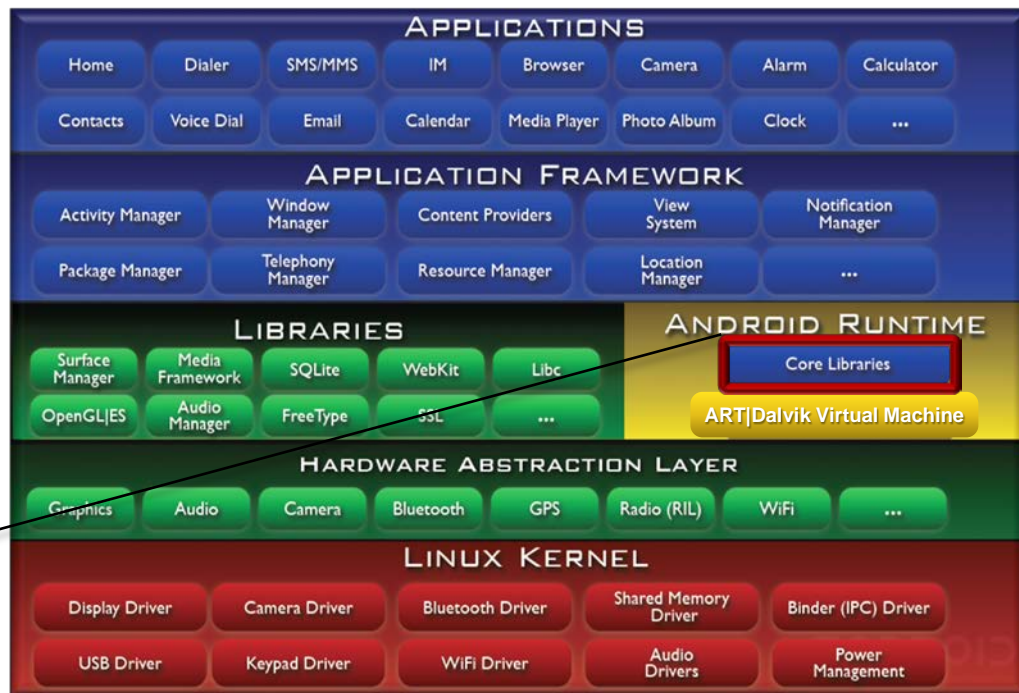See source.android.com/devices/tech/dalvik

Overview of Android's Middleware Infrastructure

- Android's *middleware infrastructure* provides reusable capabilities that extend hardware-centric OS kernel & protocol mechanisms

  - Hardware abstraction layer

  - Runtime & libraries layer

    - Android runtime

This managed execution environment is optimized for mobile device constraints

See sites.google.com/site/io/dalvik-vm-internals

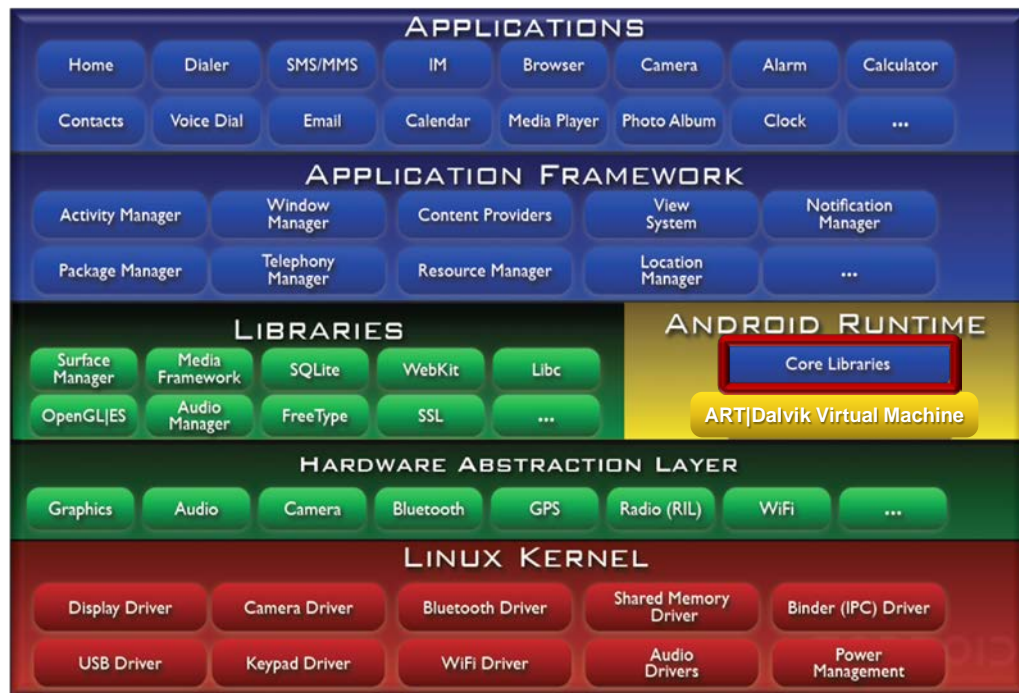# Overview of Android's Middleware Infrastructure

- Android's *middleware infrastructure* provides reusable capabilities that extend hardware-centric OS kernel & protocol mechanisms
  - Hardware abstraction layer
  - Runtime & libraries layer
    - Android runtime

A copy of core Java class libraries & core Android class libraries

See en.wikipedia.org/wiki/Comparison_of_Java_and_Android_API

Overview of Android's Middleware Infrastructure

- Android's *middleware infrastructure* provides reusable capabilities that extend hardware-centric OS kernel & protocol mechanisms
  - Hardware abstraction layer
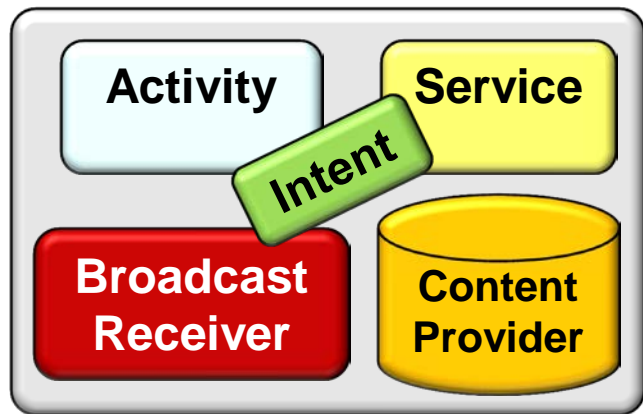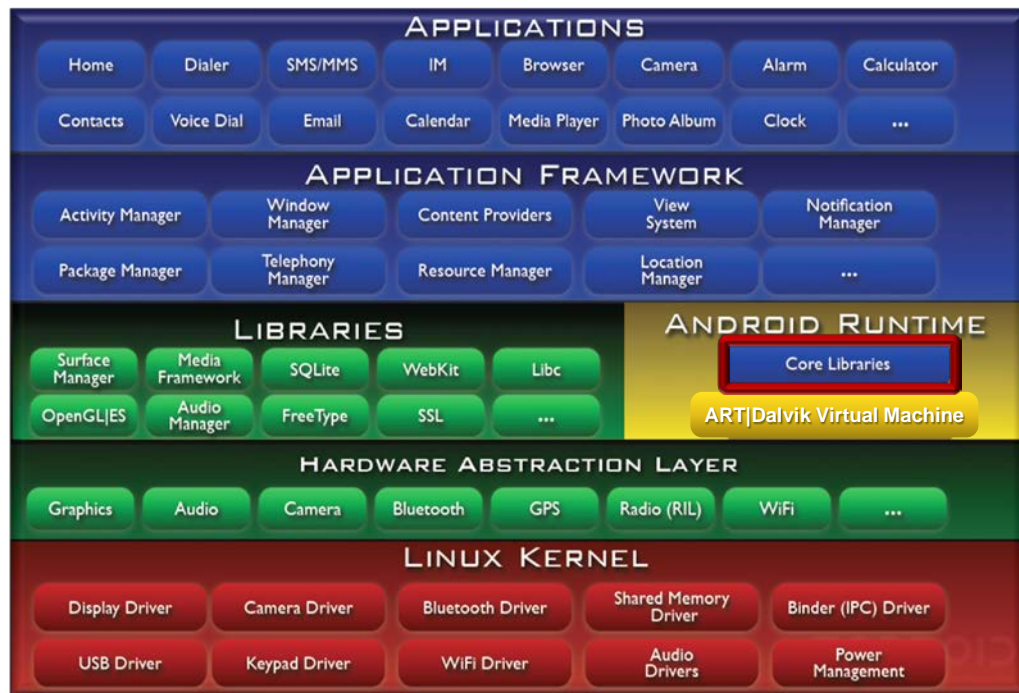 - Runtime & libraries layer
    - Android runtime

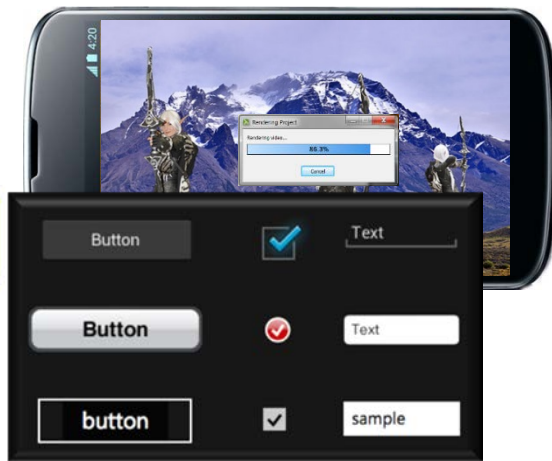
We'll discuss Java threading shortly

Overview of Android's Middleware Infrastructure

- Android's *middleware infrastructure* provides reusable capabilities that extend hardware-centric OS kernel & protocol mechanisms
  - Hardware abstraction layer
  - Runtime & libraries layer
    - Android runtime

Android's core libraries provide key components that we'll also cover shortly
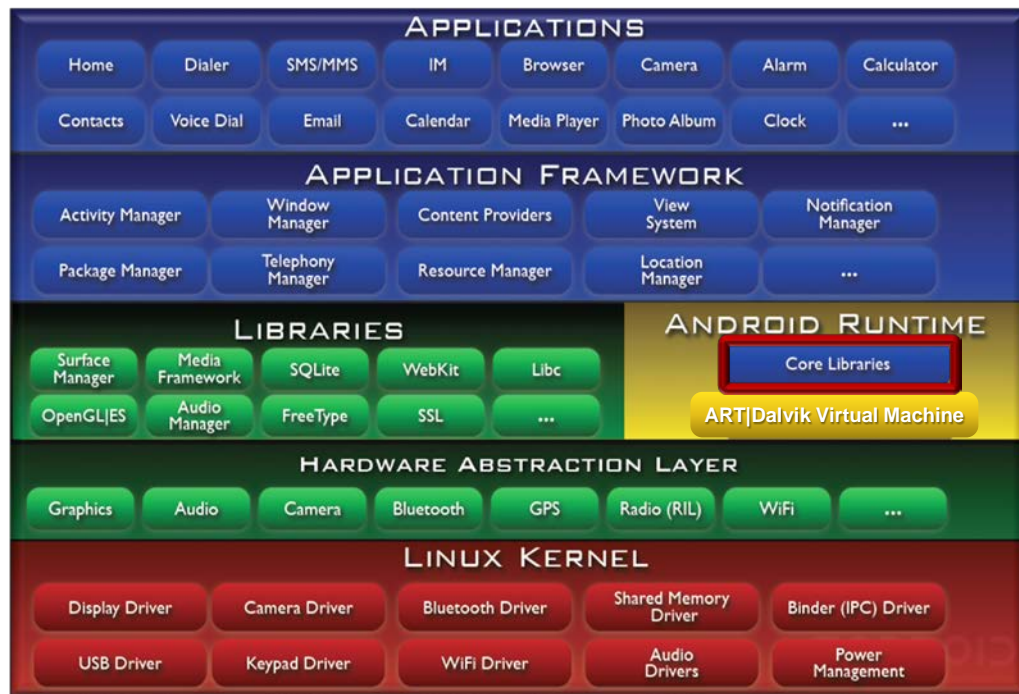
Overview of Android's Middleware Infrastructure

- Android's *middleware infrastructure* provides reusable capabilities that extend hardware-centric OS kernel & protocol mechanisms
  - Hardware abstraction layer
  - Runtime & libraries layer
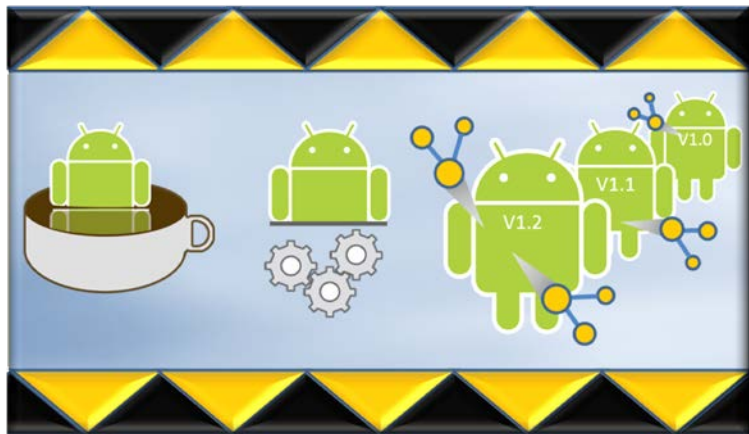   - Android runtime

Android's core libraries provide many other UI & persistence components

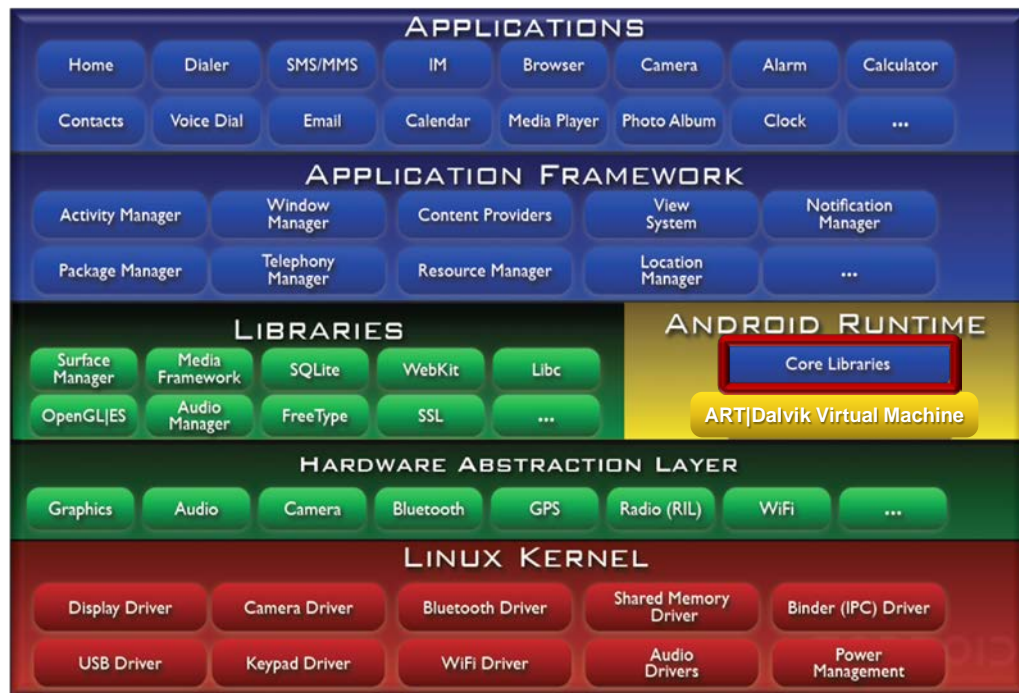# Overview of Android's Middleware Infrastructure

- Android's *middleware infrastructure* provides reusable capabilities that extend hardware-centric OS kernel & protocol mechanisms

  - Hardware abstraction layer

  - Runtime & libraries layer

    - Android runtime

We cover Android's core libraries in the *Android App Development* Specialization
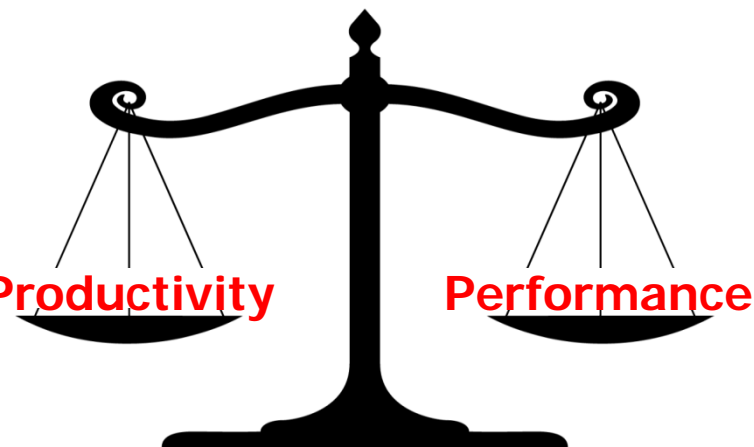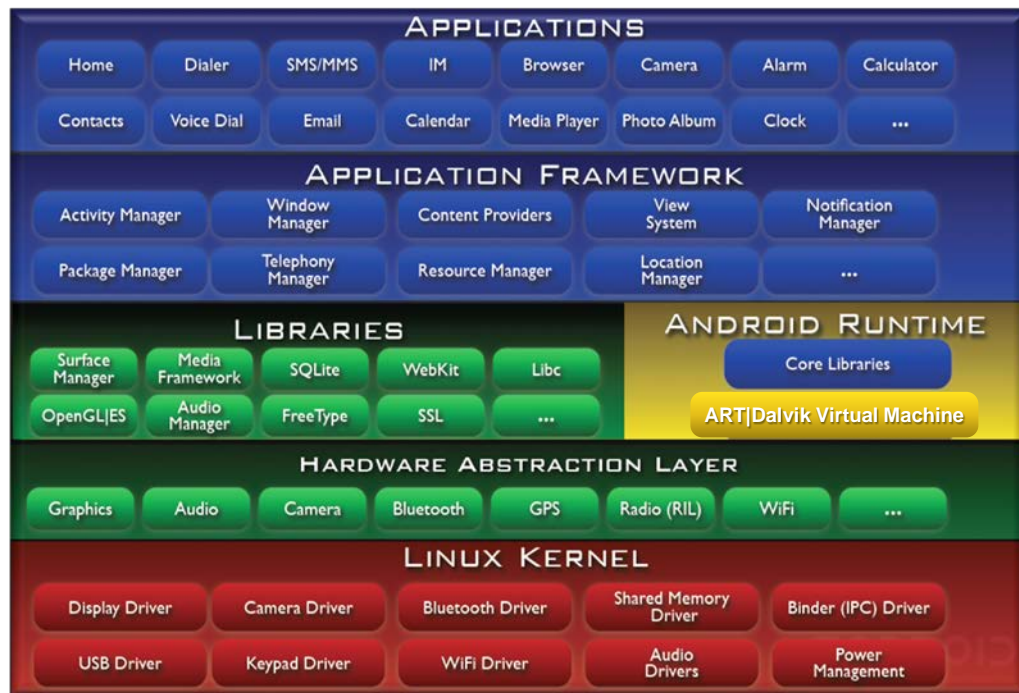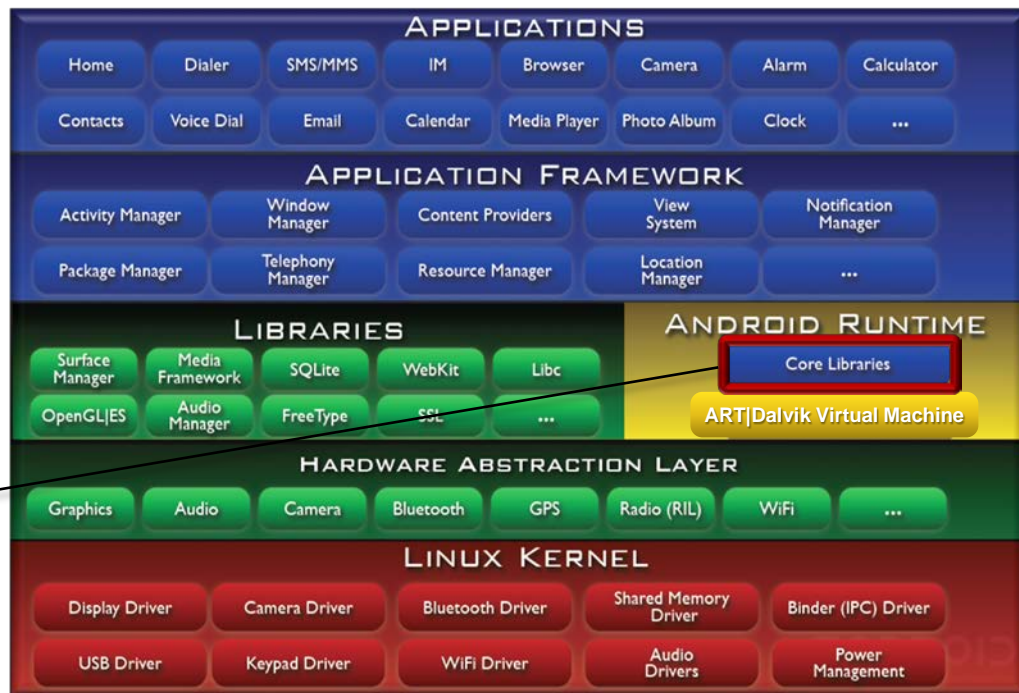
Overview of Android's Middleware Infrastructure

- Android's *middleware infrastructure* provides reusable capabilities that extend hardware-centric OS kernel & protocol mechanisms

  - Hardware abstraction layer

  - Runtime & libraries layer

    - Android runtime

Productivity   **Performance**

Java/JNI

C++/C

C

APPLICATIONS

| Home | Dialer | SMS/MMS | IM | Browser | Camera | Alarm | Calculator |
| Contacts | Voice Dial | Email | Calendar | Media Player | Photo Album | Clock | ... |

APPLICATION FRAMEWORK

| Activity Manager | Window Manager | Content Providers | View System | Notification Manager |
| Package Manager | Telephony Manager | Resource Manager | Location Manager | ... |

LIBRARIES | **ANDROID RUNTIME**

| Surface Manager | Media Framework | SQLite | WebKit | Libc | Core Libraries |
| OpenGL|ES | Audio Manager | FreeType | SSL | ... | ART|Dalvik Virtual Machine |

HARDWARE ABSTRACTION LAYER

| Graphics | Audio | Camera | Bluetooth | GPS | Radio (RIL) | WiFi | ... |

LINUX KERNEL

| Display Driver | Camera Driver | Bluetooth Driver | Shared Memory Driver | Binder (IPC) Driver |
| USB Driver | Keypad Driver | WiFi Driver | Audio Drivers | Power Management |

Android—like Java—balances run-time performance & developer productivity

Overview of Android's Middleware Infrastructure

- Android's *middleware infrastructure* provides reusable capabilities that extend hardware-centric OS kernel & protocol mechanisms

  - Hardware abstraction layer

  - Runtime & libraries layer

    - Android runtime

Android's core libraries are often implemented as wrapper facades

See www.dre.vanderbilt.edu/~schmidt/PDF/wrapper-facade.pdf

Overview of Android's Middleware Infrastructure

- Android's *middleware infrastructure* provides reusable capabilities that extend hardware-centric OS kernel & protocol mechanisms

  - Hardware abstraction layer

  - Runtime & libraries layer

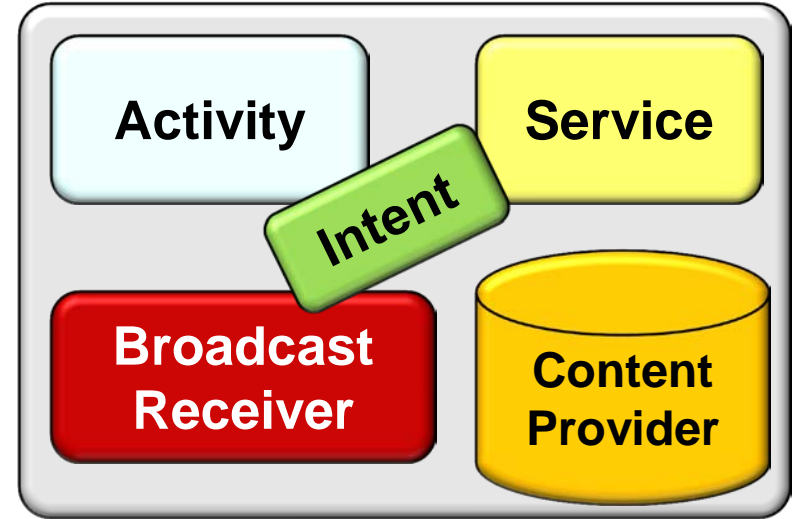   - Android runtime

      - Native C/C++ libraries
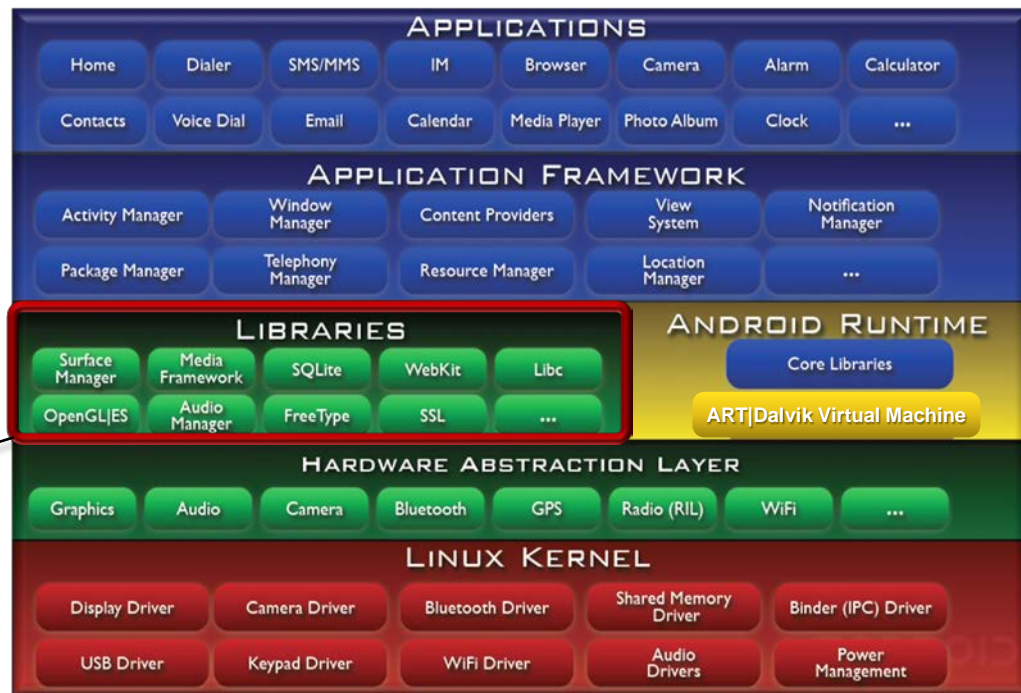
These Java wrapper façade are implemented via native C/C++ code

See developer.android.com/tools/sdk/ndk

Overview of Key Android App Components

Overview of Key Android App Components

- App components are essential building blocks of mobile apps that provide various hooks via which Android can effect an app's lifecycle
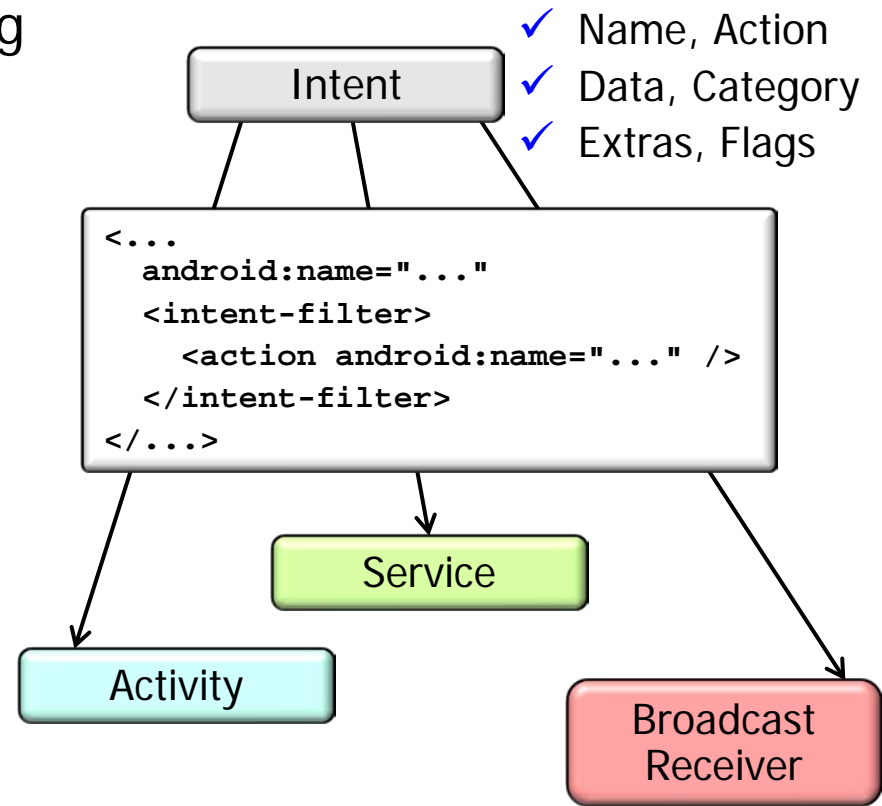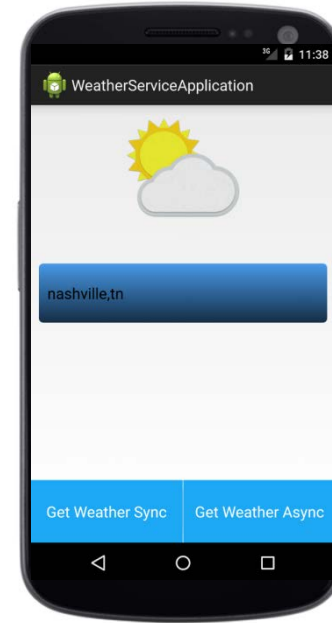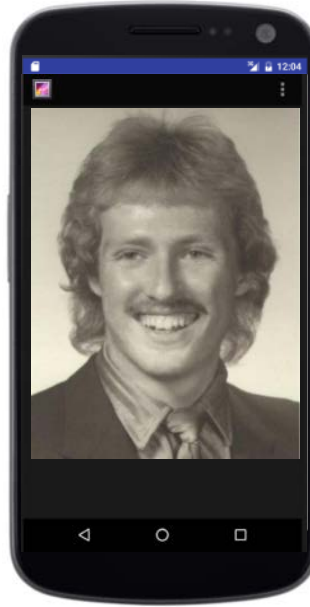
Overview of Key Android App Components

- App components are essential building blocks of mobile apps that provide various hooks via which Android can effect an app's lifecycle, e.g.

  - **Intents**
    - Messages that describe an action to perform or an event that has occurred

Intent

- ✓ Name, Action
- ✓ Data, Category
- ✓ Extras, Flags

```
<...
  android:name="..."
  <intent-filter>
    <action android:name="..." />
  </intent-filter>
</...>
```

Service

Activity

Broadcast Receiver

See developer.android.com/reference/android/content/Intent.html

Overview of Key Android App Components

- App components are essential building blocks of mobile apps that provide various hooks via which Android can effect an app's lifecycle, e.g.
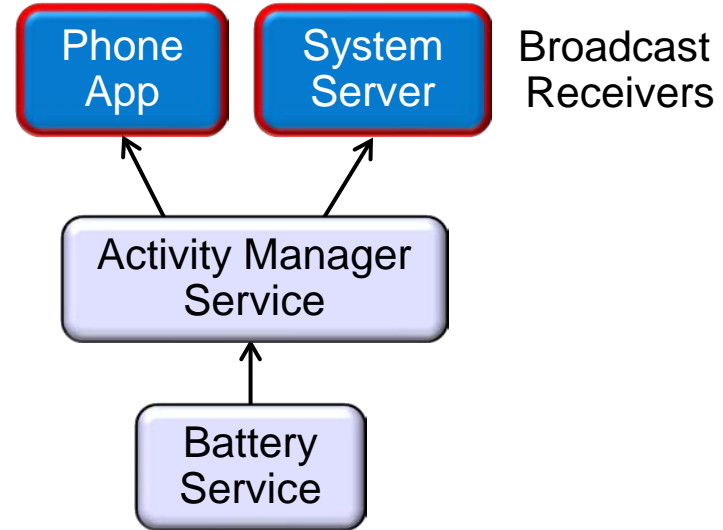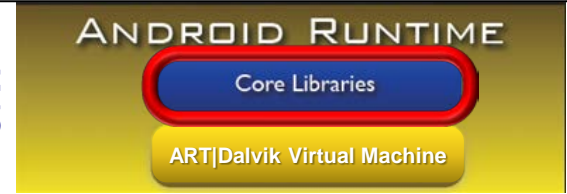
  - **Intents**

  - **Activities**

    - Provide a screen within which users can interact in order to do something

See developer.android.com/guide/components/activities.html

Overview of Key Android App Components

- App components are essential building blocks of mobile apps that provide various hooks via which Android can effect an app's lifecycle, e.g.

  - **Intents**
  - **Activities**

  - **Broadcast Receivers**
   - Event handlers that respond to broadcast announcements

C/Java/JNI

ANDROID RUNTIME
Core Libraries
ART|Dalvik Virtual Machine

Phone App   System Server   Broadcast Receivers

Activity Manager Service

Battery Service

See developer.android.com/reference/android/content/BroadcastReceiver.html

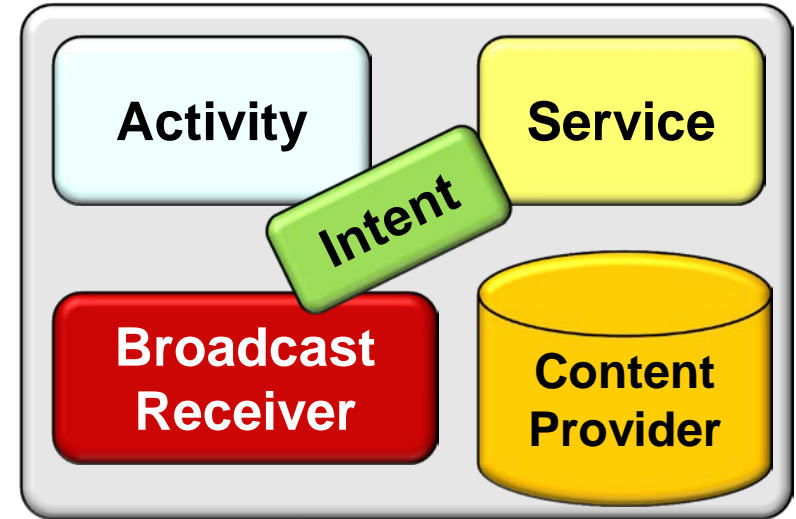# Overview of Key Android App Components

- App components are essential building blocks of mobile apps that provide various hooks via which Android can effect an app's lifecycle, e.g.

  - **Intents**
  - **Activities**
  - **Broadcast Receivers**

  - **Services**
    - Run in background to perform long-running operations or access remote resources
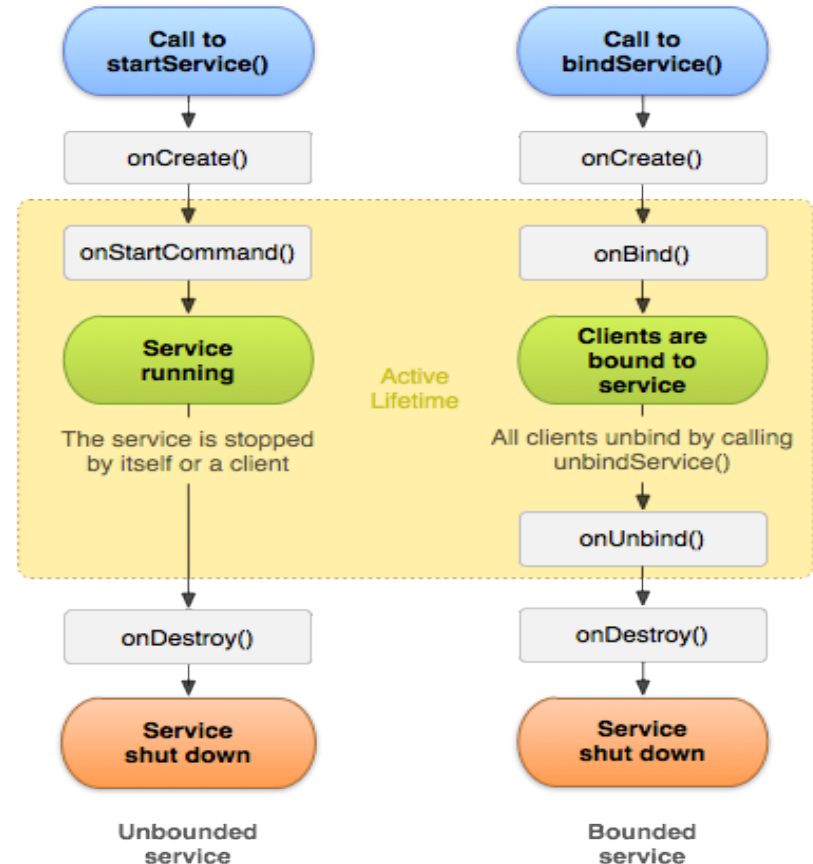
See developer.android.com/guide/components/services.html

Overview of Key Android App Components

- App components are essential building blocks of mobile apps that provide various hooks via which Android can effect an app's lifecycle, e.g.

  - **Intents**
  - **Activities**
  - **Broadcast Receivers**
  - **Services**

- **Content Providers**

  - Manage access to structured data & provide data security mechanisms
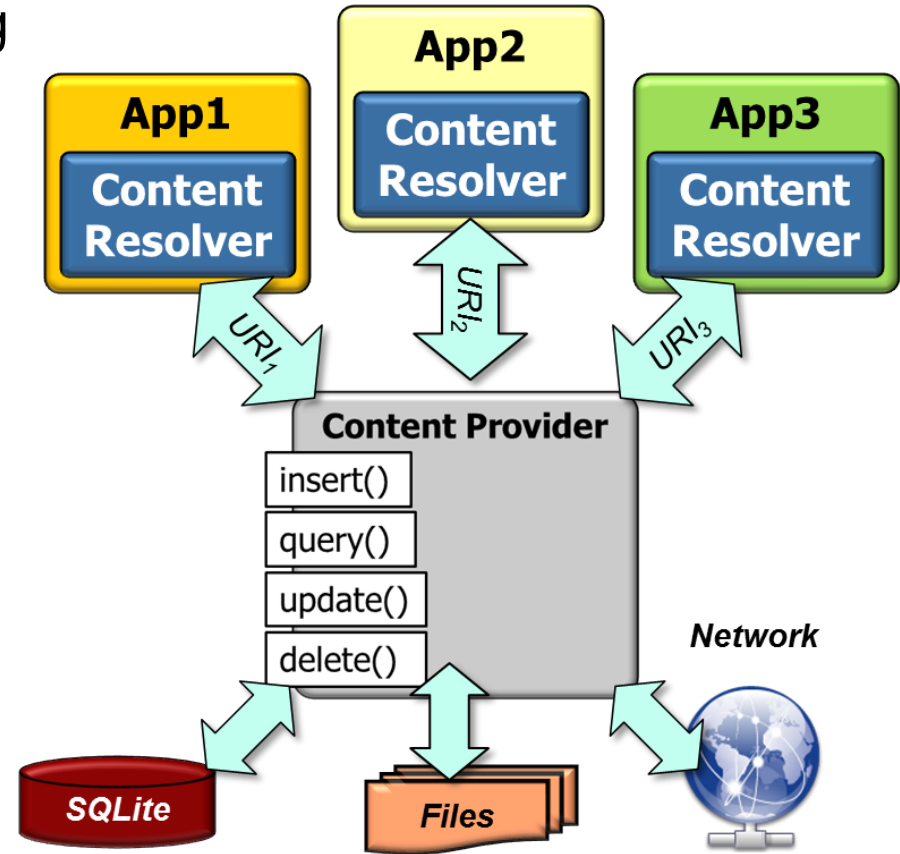
See developer.android.com/guide/topics/providers/content-providers.html

Overview of Key Android App Components

- We cover intents, activities, & broadcast receivers in MOOC 2 & services & content providers in MOOC 3

See developer.android.com/guide/components/fundamentals.html#Components
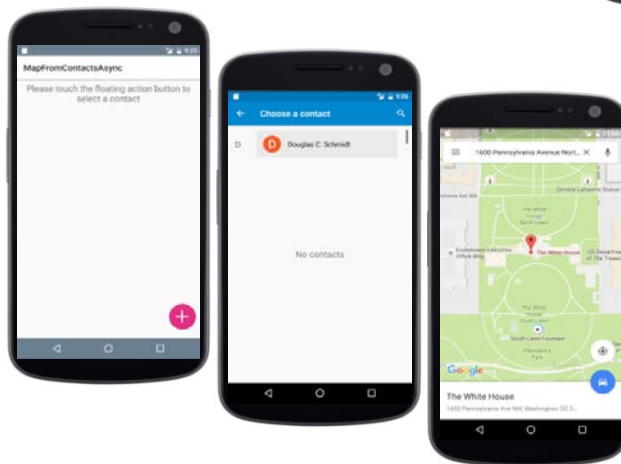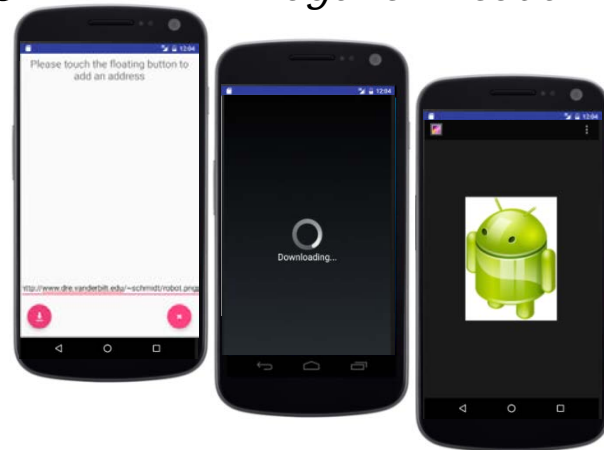
Overview of Java Threads in Android

Overview of Java Threads in Android

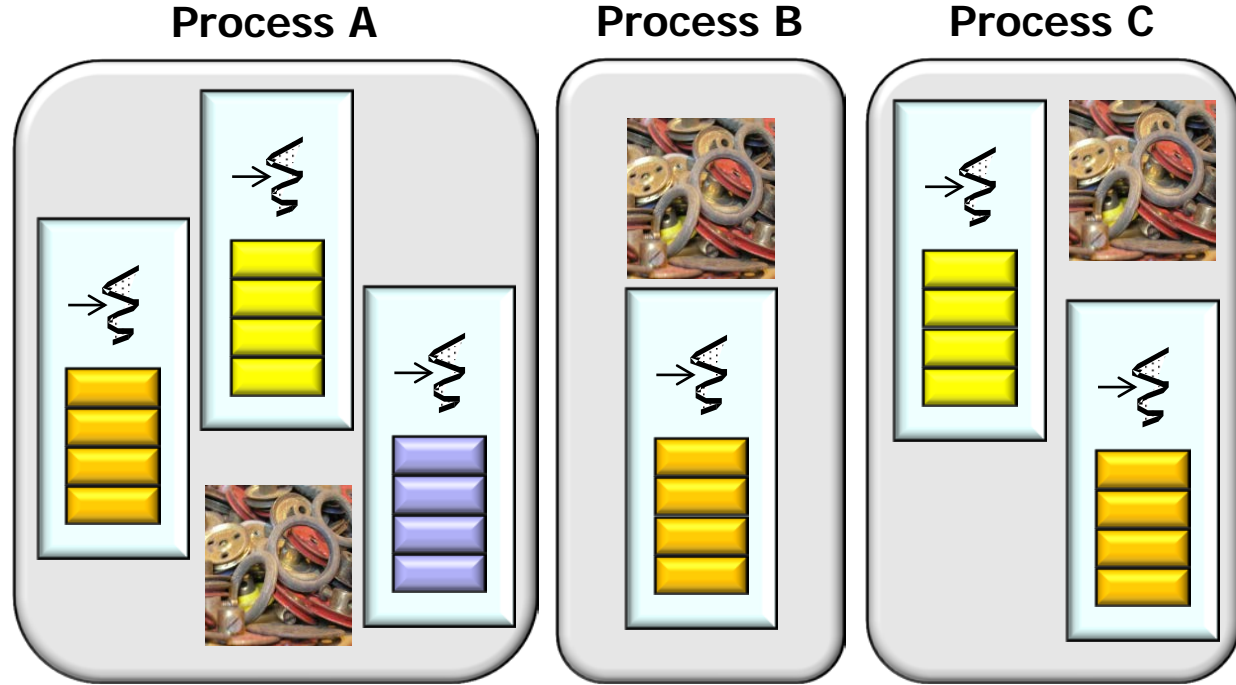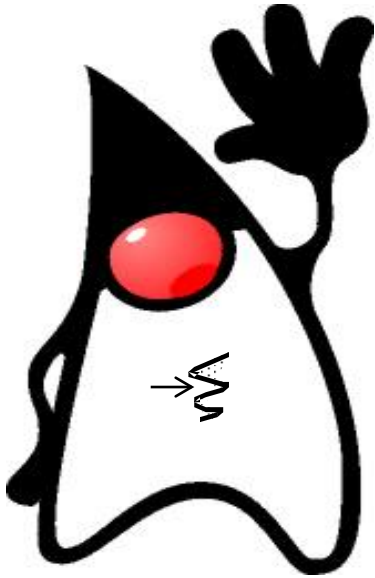- Many example apps in this course use Java threads

PingPongReceivers*

MapFromContacts*

ImageDownloader*

Overview of Java Threads in Android

- Java threads are the smallest unit of execution for sequences of programmed instructions

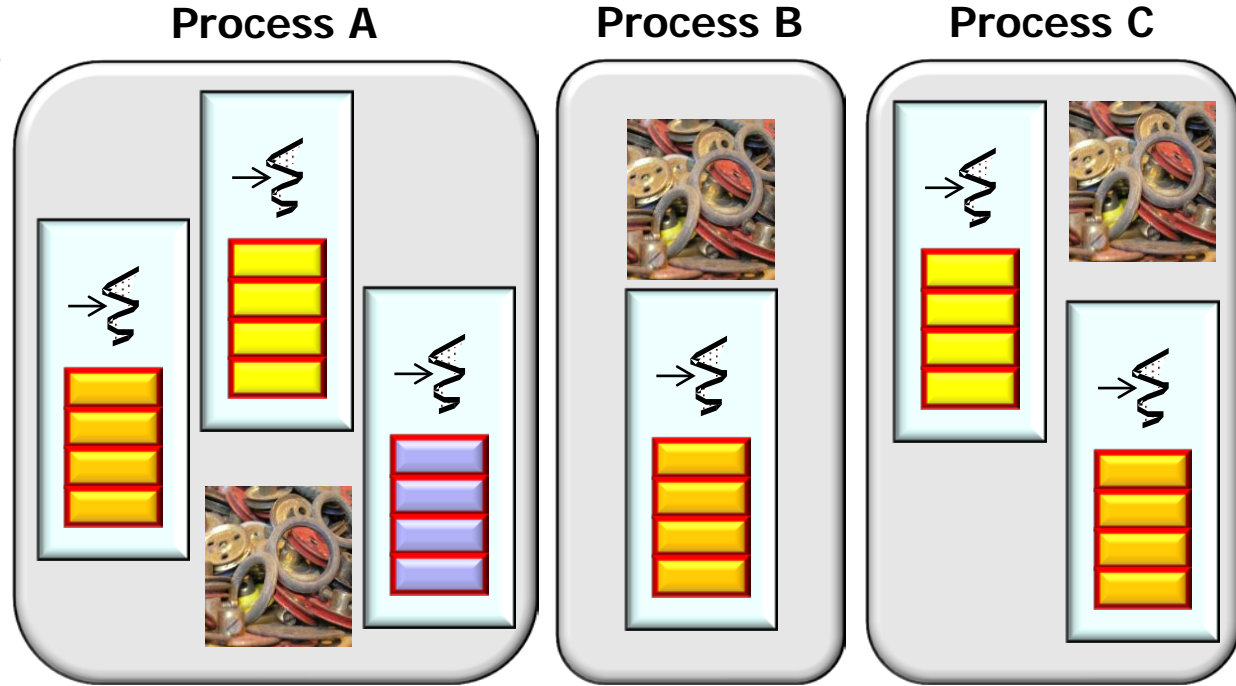**Process A**   **Process B**   **Process C**

Overview of Java Threads in Android

- Java threads are the smallest unit of execution for sequences of programmed instructions

  - Each process can have multiple threads that run concurrently

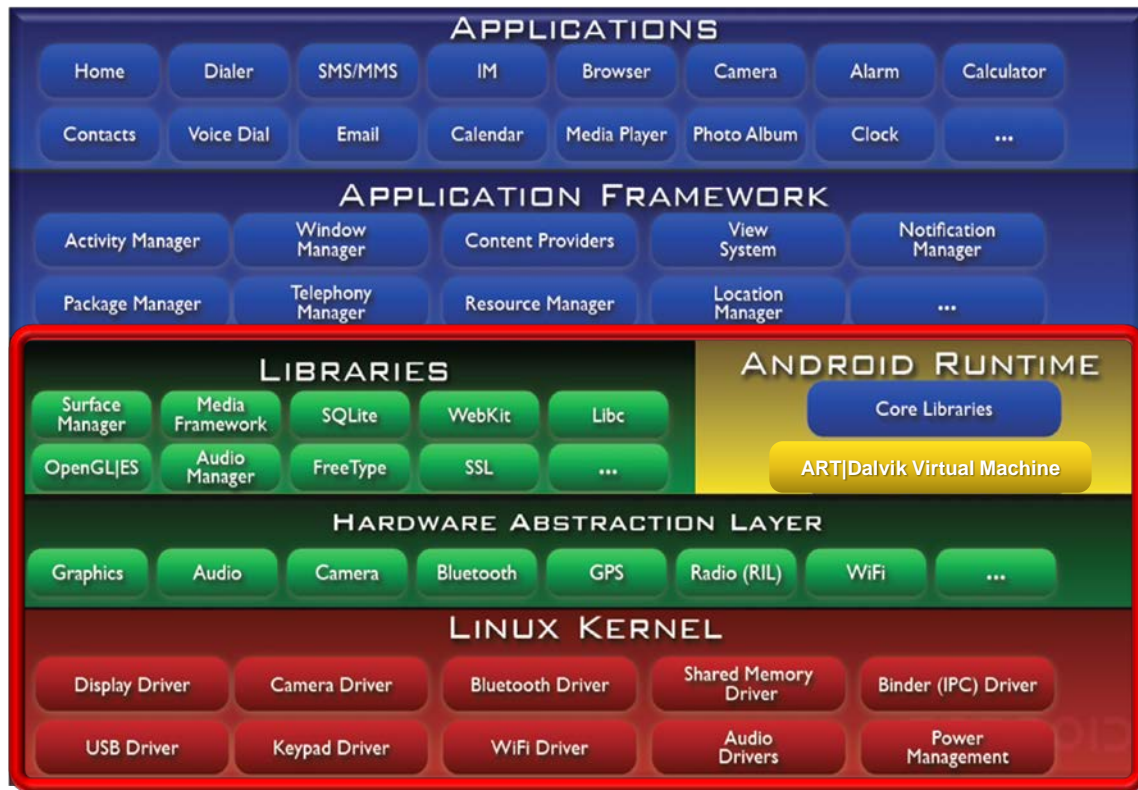**Process A** **Process B** **Process C**

Overview of Java Threads in Android

- Java threads are the smallest unit of execution for sequences of programmed instructions

  - Each process can have multiple threads that run concurrently

  - Each thread contains a call stack to keep track of method state

Process A **Process B** **Process C**

See en.wikipedia.org/wiki/Call_stack

Overview of Java Threads in Android

- Java threads are the smallest unit of execution for sequences of programmed instructions

  - Each process can have multiple threads that run concurrently

  - Each thread contains a call stack to keep track of method state

- Android implements Java threads using mechanisms in various layers

- Java threads are the smallest unit of execution for sequences of programmed instructions

  - Each process can have multiple threads that run concurrently

  - Each thread contains a call stack to keep track of method state

  - Android implements Java threads using mechanisms in various layers
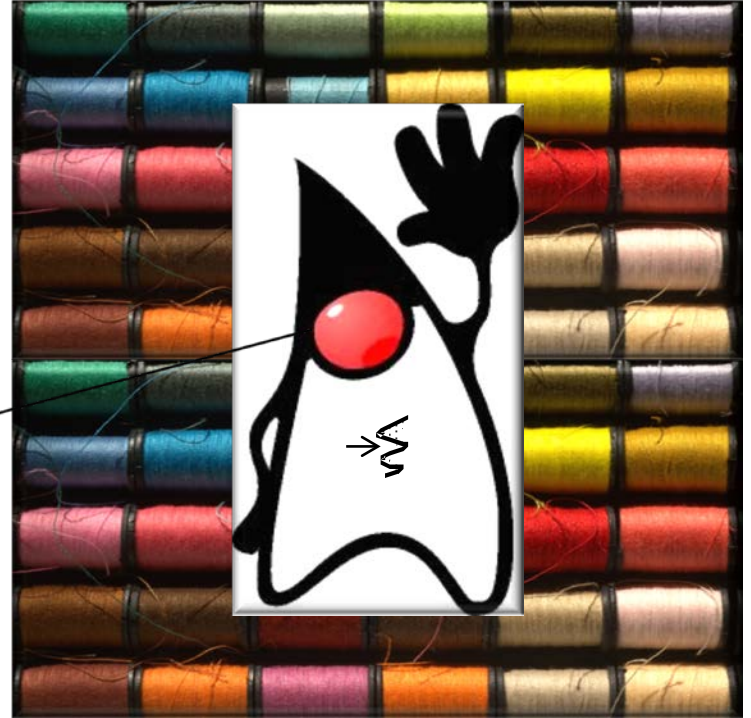
- Starting a Java thread takes a non-trivial amount of time & system resources

```
1. MyThread.start()

2. Thread.start()

3. VMThread.create()

4. Dalvik_java_lang_VMThread_create()

5. dvmCreateInterpThread()

6. pthread_create()

7. interpThreadStart()

8. dvmCallMethod()

9. MyThread.run()
```
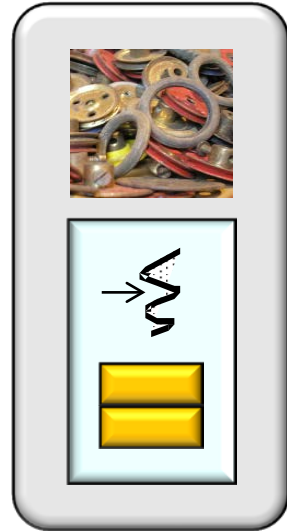
- Java threads must be given code to run

```
public void run() {
    // code to run goes here
}
```
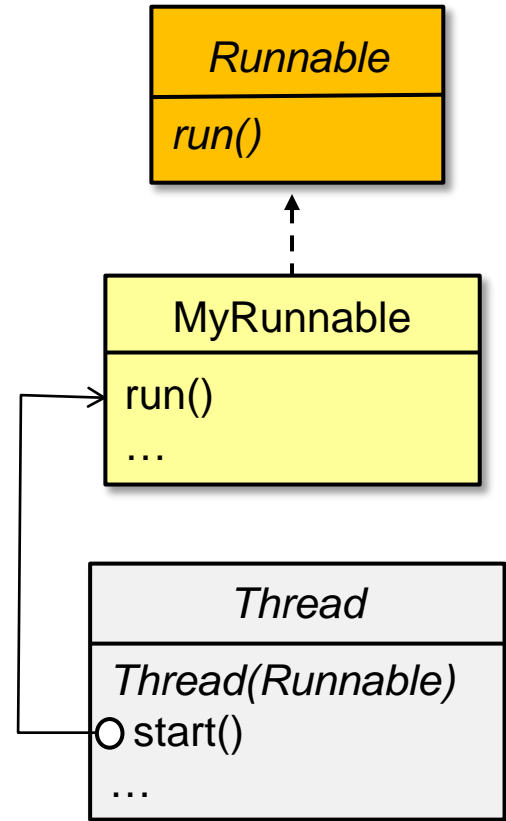
Overview of Java Threads in Android

- Java threads must be given code to run, e.g.

  - Implement the Runnable interface

```java
public class MyRunnable
        implements Runnable {
  public void run() {
    Log.d(TAG, "hello world"); ...
  }
}
final Runnable myRunnable =
  new MyRunnable();
new Thread(myRunnable).start();
```

*Create/start Thread using named class object as Runnable*

Runnable
run()

MyRunnable
run()
…

Thread
Thread(Runnable)
start()
…

See docs.oracle.com/javase/7/docs/api/java/lang/Runnable.html

Overview of Java Threads in Android

- Java threads must be given code to run, e.g.

  - Implement the Runnable interface

```java
public class MyRunnable
        implements Runnable {
  public void run() {
    Log.d(TAG, "hello world"); ...
  }
}
final Runnable myRunnable =
  new MyRunnable();
new Thread(myRunnable).start();
```
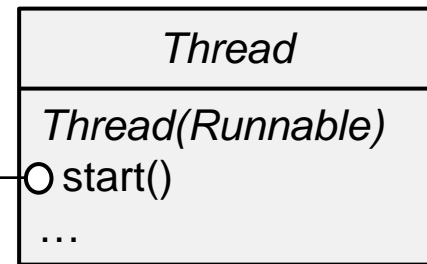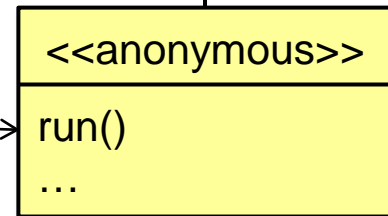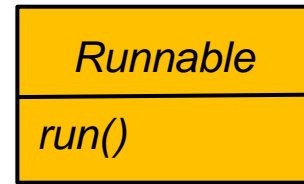
This hook method is called back at runtime
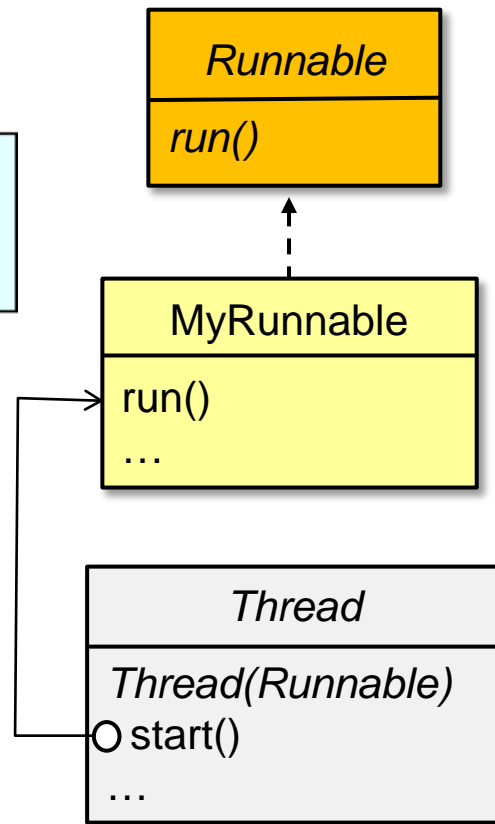
Runnable
run()

MyRunnable
run()
…

Thread
Thread(Runnable)
○ start()
…

Overview of Java Threads in Android

- Java threads must be given code to run, e.g.

  - Implement the Runnable interface

```java
public interface Runnable {
    public void run();
}


new Thread(new Runnable() {
  public void run(){
    Log.d(TAG, "hello world"); ...
  }
}).start();
```

Create/start a Thread using anonymous inner class as Runnable

Runnable
run()

<<anonymous>>
run()
…

Thread
Thread(Runnable)
start()
…

See docs.oracle.com/javase/tutorial/java/javaOO/anonymousclasses.html
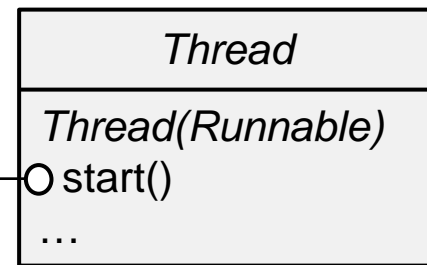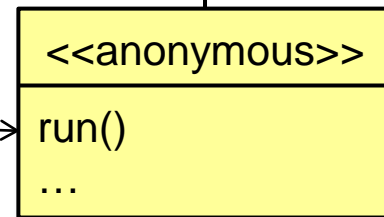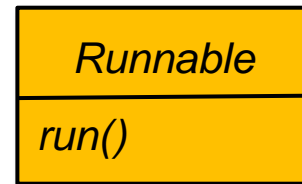
Overview of Java Threads in Android

- Java threads must be given code to run, e.g.

  - Implement the Runnable interface

```java
public interface Runnable {
    public void run();
}


new Thread(new Runnable() {
  public void run() {
    Log.d(TAG, "hello world"); ...
  }
}).start();
```

This hook method is called back at runtime

Runnable

run()

<<anonymous>>

run()

…

Thread

Thread(Runnable)

start()

…

Overview of Java Threads in Android
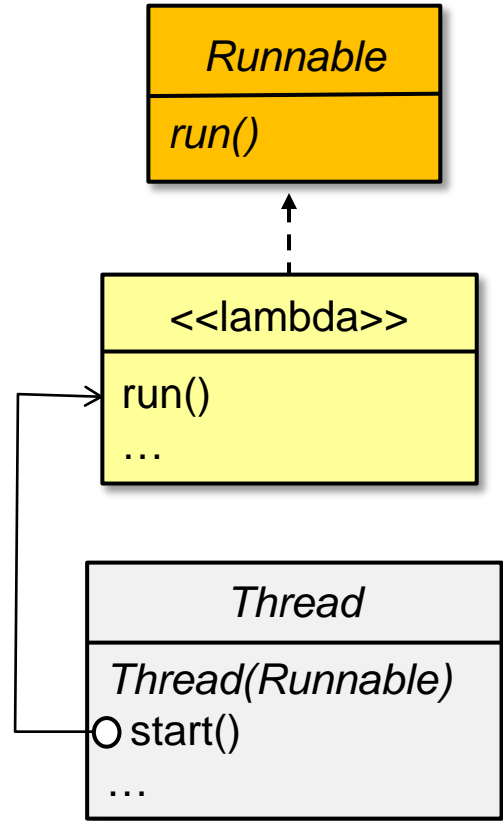
- Java threads must be given code to run, e.g.
  - Implement the Runnable interface

  - Use Java 8 lambda expressions

    ```java
    public interface Runnable {
        public void run();
    }

    new Thread(() -> {
      Log.d(TAG, "hello world"); ...
    }).start();
    ```

Create/start a Thread using a lambda expression as Runnable

Runnable

run()

<<lambda>>

run()

…

Thread

Thread(Runnable)
○ start()

…

Overview of Java Threads in Android
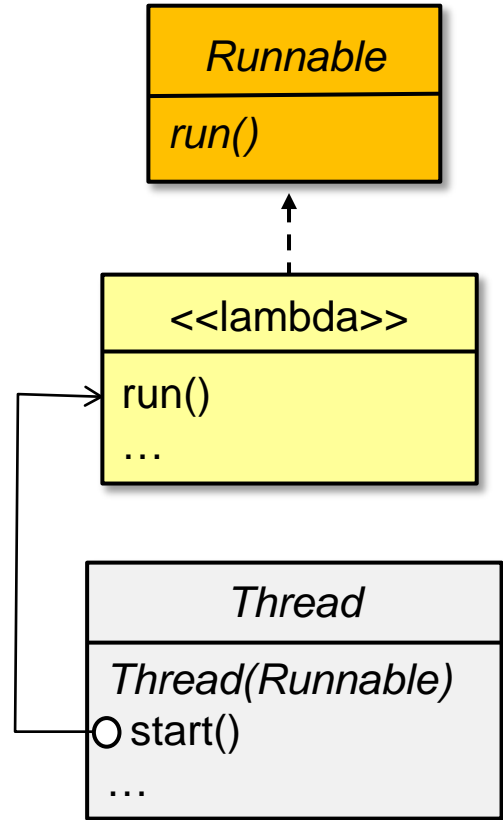
- Java threads must be given code to run, e.g.
  - Implement the Runnable interface

  - Use Java 8 lambda expressions

```java
public interface Runnable {
    public void run();
}


new Thread(() -> {
  Log.d(TAG, "hello world"); ...
}).start();
```

A lambda expression is an unnamed block of code that can be passed around & executed later

Runnable

run()

<<lambda>>

run()

…

Thread

Thread(Runnable)
start()

…

Overview of Java Threads in Android

- Java threads must be given code to run, e.g.
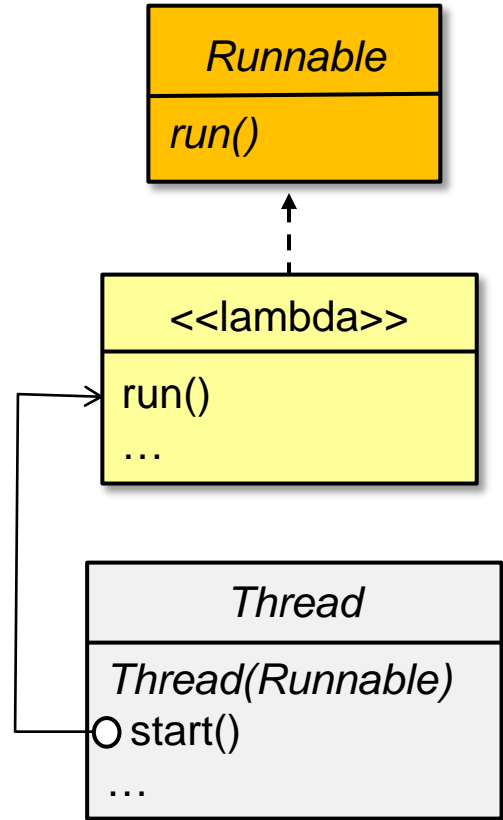 - Implement the Runnable interface
  - Use Java 8 lambda expressions

```java
public interface Runnable {
    public void run();
}

new Thread(() -> {
  Log.d(TAG, "hello world"); ...
}).start();
```

This lambda is called back at runtime

Runnable

run()

<<lambda>>

run()

…

Thread

Thread(Runnable)

start()

…

Java 8 lambda expressions are supported in Android API level 24 & beyond

Overview of Java Threads in Android

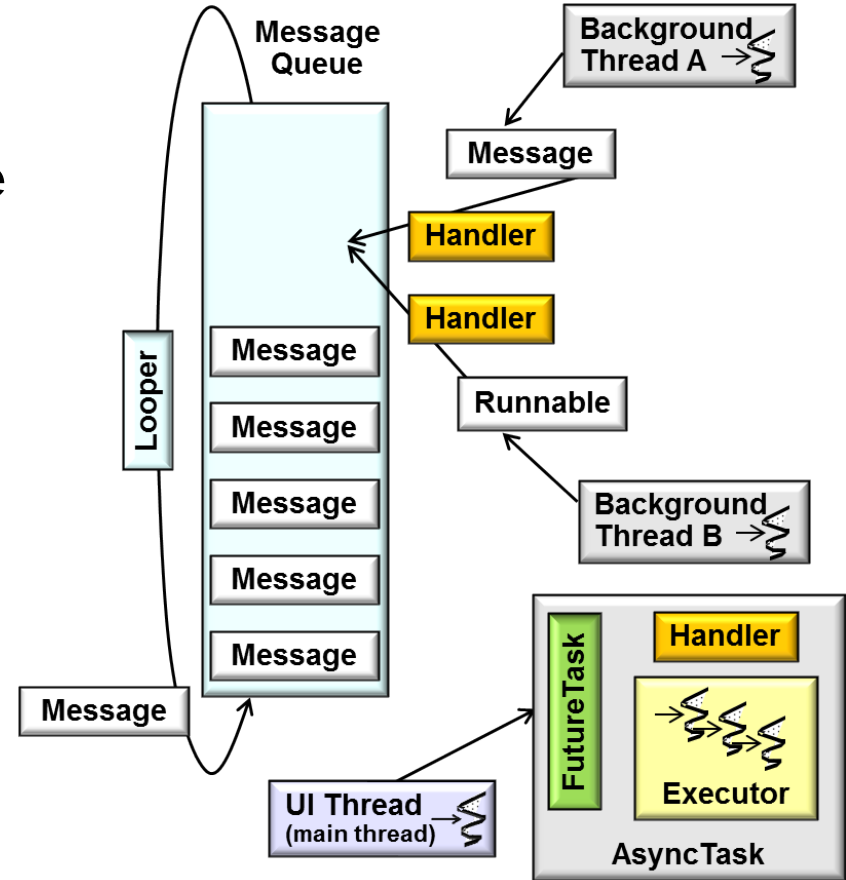- Android contains dozens of classes related to Java threads

- Android contains dozens of classes related to Java threads

  - Fortunately, Android encapsulates the bulk of these Java threads classes within its concurrency frameworks

See upcoming module on Android Activities for more on its concurrency frameworks

- More information on Java threads is available online

Android Concurrency: Overview of Java Threads

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Institute for Software
Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA

See www.youtube.com/watch?v=1YwVH-nhDtc

End of Overview of Android (Part 2): Middleware Infrastructure