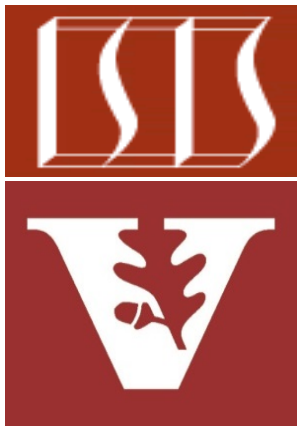


# Infrastructure Middleware (Part 3): Android Runtime Core & Native Libraries



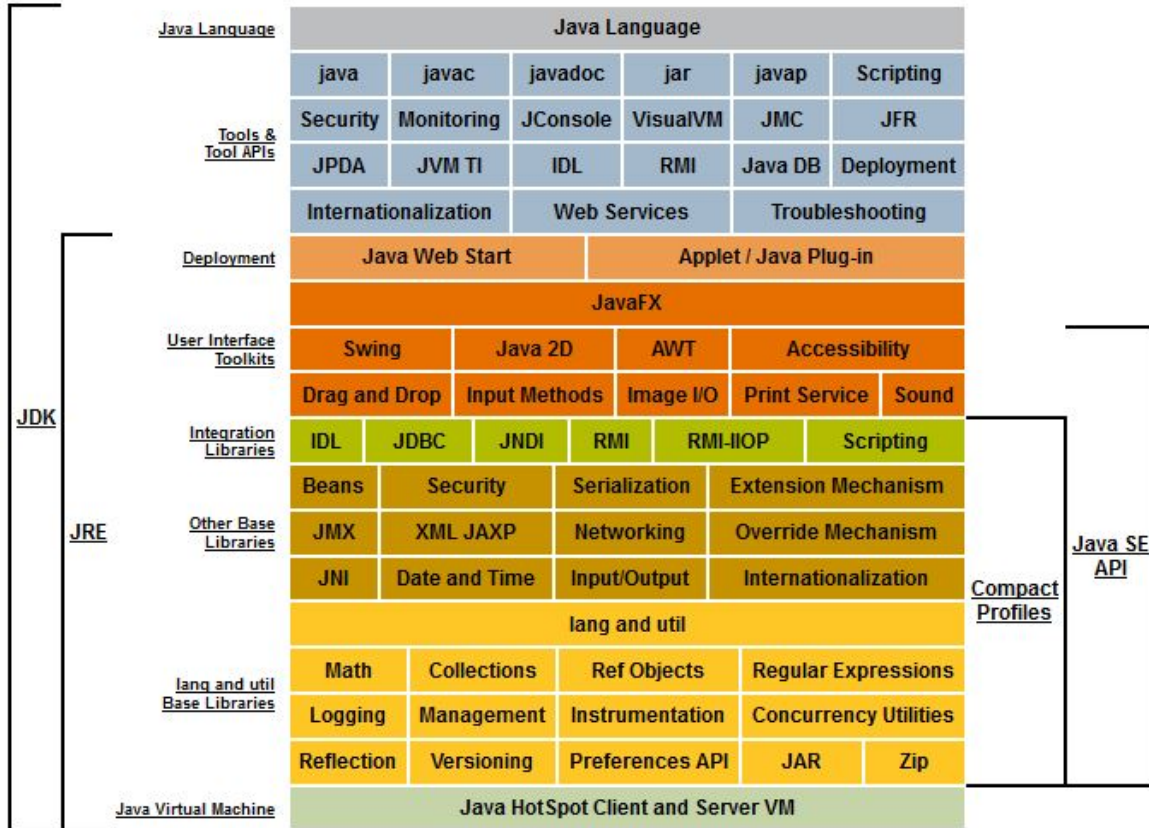
Douglas C. Schmidt  
[d.schmidt@vanderbilt.edu](mailto:d.schmidt@vanderbilt.edu)  
[www.dre.vanderbilt.edu/~schmidt](http://www.dre.vanderbilt.edu/~schmidt)

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



# Learning Objectives in this Part of the Lesson

- Recognize key core Java libraries that are part of the Android platform





# Learning Objectives in this Part of the Lesson

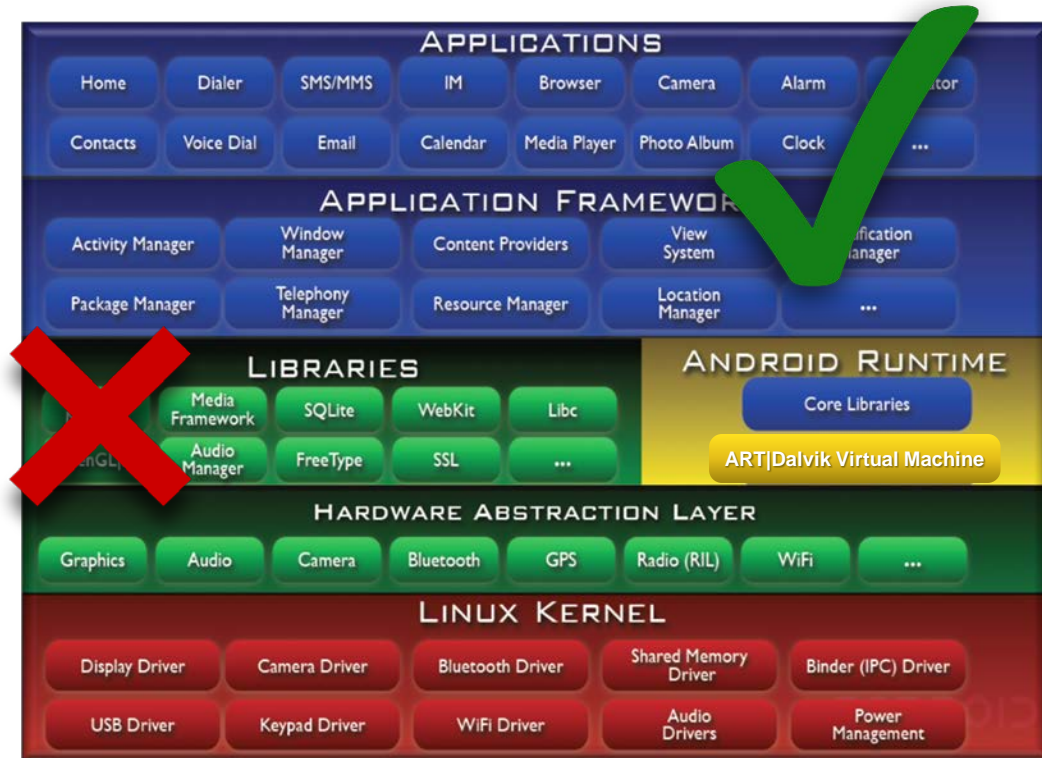
- Recognize key core Java libraries that are part of the Android platform
- Recognize key core Android libraries that are part of the Android platform
- Recognize key native libraries that are part of the Android platform

C++/C



# Learning Objectives in this Part of the Lesson

- Recognize key core Java libraries that are part of the Android platform
- Recognize key core Android libraries that are part of the Android platform
- Recognize key native libraries that are part of the Android platform



Apps use core Java/Android libraries extensively; native libraries not as much

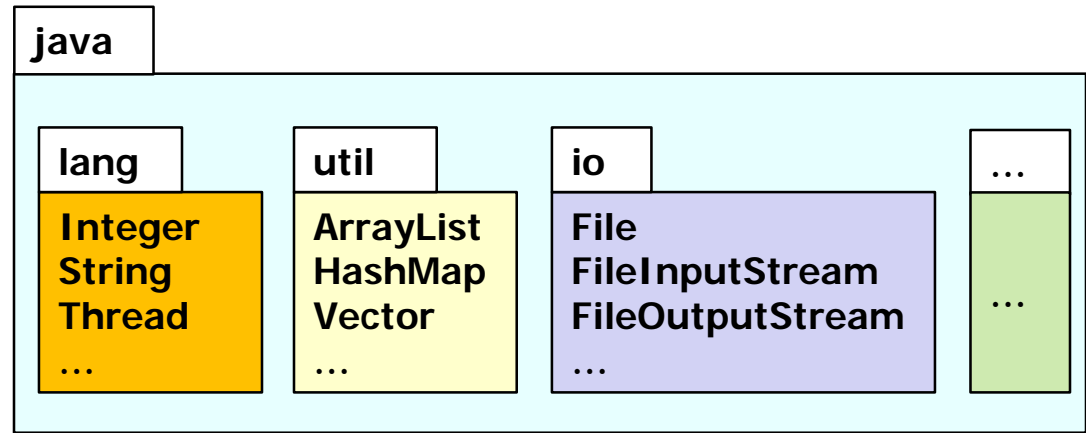
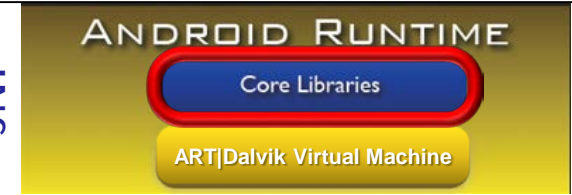
---

# Overview of the Android Runtime: Core Java Libraries

# Android Runtime: Core Java Libraries

- Android contains many (but not all) core Java libraries in the `java.*` & `javax.*` packages

C/Java/  
JNI



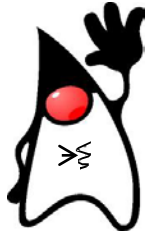
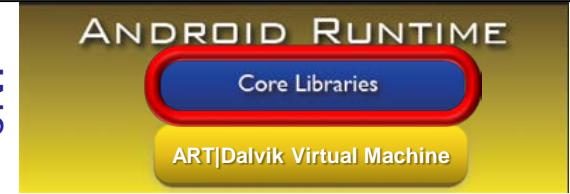
See [en.wikipedia.org/wiki/Comparison\\_of\\_Java\\_and\\_Android\\_API](https://en.wikipedia.org/wiki/Comparison_of_Java_and_Android_API)



# Android Runtime: Core Java Libraries

- Android contains many (but not all) core Java libraries in the java.\* & javax.\* packages, e.g.
  - Java Thread

C/Java/  
JNI



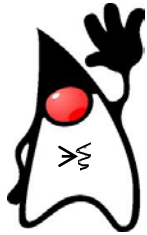
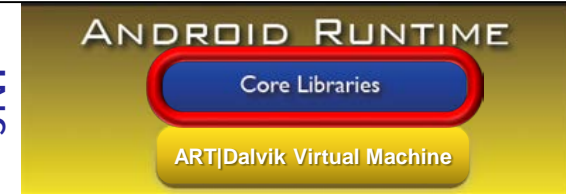
See [developer.android.com/reference/java/lang/Thread.html](https://developer.android.com/reference/java/lang/Thread.html)



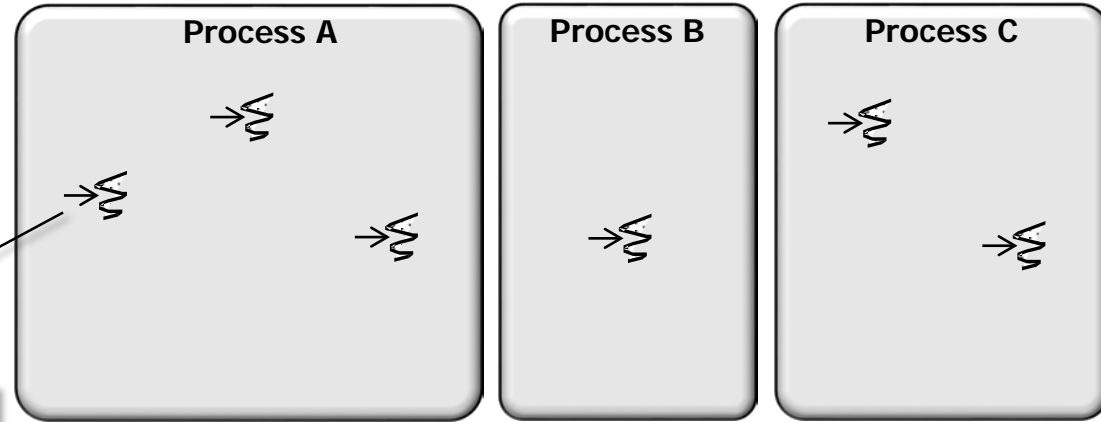
# Android Runtime: Core Java Libraries

- Android contains many (but not all) core Java libraries in the `java.*` & `javax.*` packages, e.g.
  - Java Thread

C/Java/  
JNI



*A Java thread is a unit of computation that runs in the context of a process*

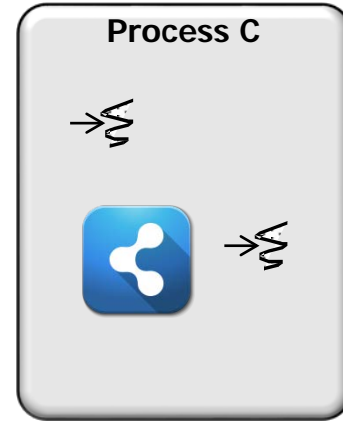
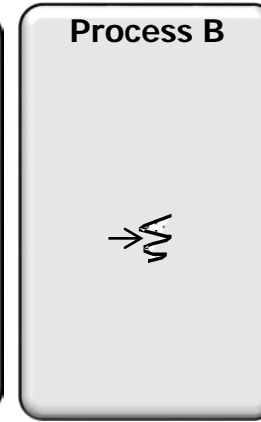
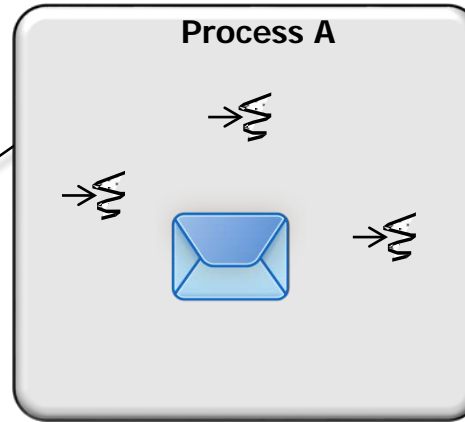
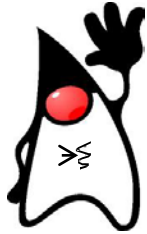
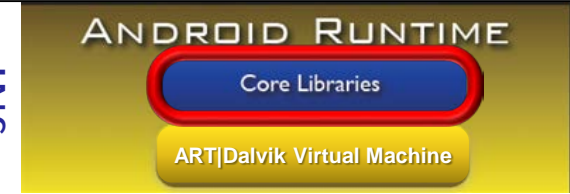


See [developer.android.com/reference/java/lang/Thread.html](https://developer.android.com/reference/java/lang/Thread.html)

# Android Runtime: Core Java Libraries

- Android contains many (but not all) core Java libraries in the `java.*` & `javax.*` packages, e.g.
  - Java Thread

C/Java/  
JNI



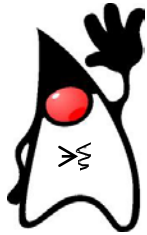
*Java threads running in a process can communicate with each other via shared objects or message passing*

See [en.wikipedia.org/wiki/Thread\\_\(computing\)](https://en.wikipedia.org/wiki/Thread_(computing))

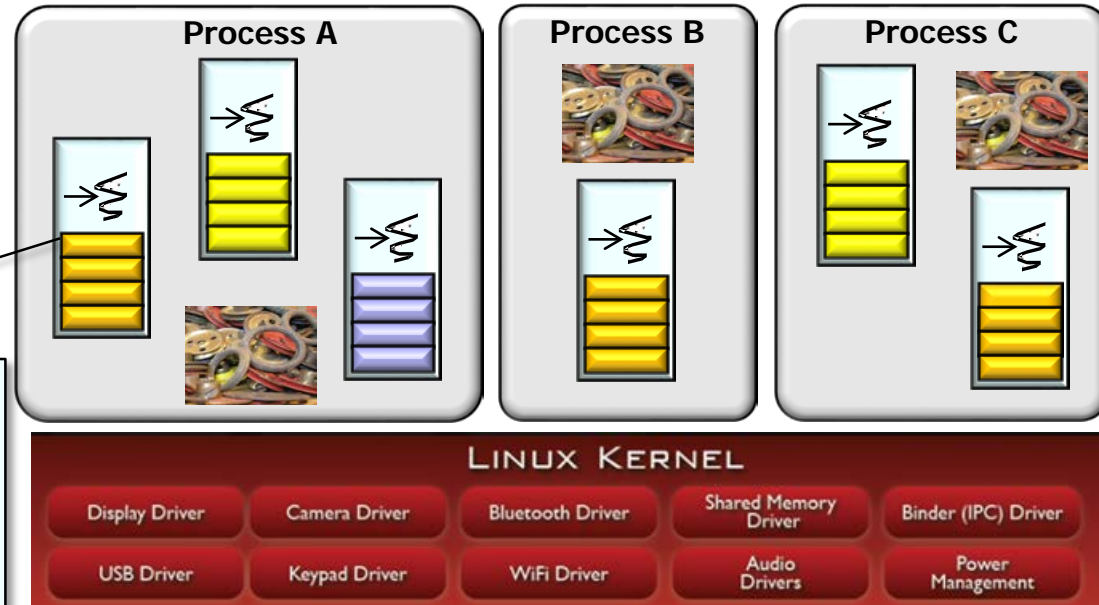
# Android Runtime: Core Java Libraries

- Android contains many (but not all) core Java libraries in the `java.*` & `javax.*` packages, e.g.
  - Java Thread

C/Java/  
JNI



*Each Java thread leverages unique "state" from the underlying Linux kernel thread, e.g., a stack, a program counter, & other registers*

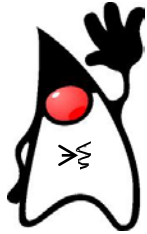
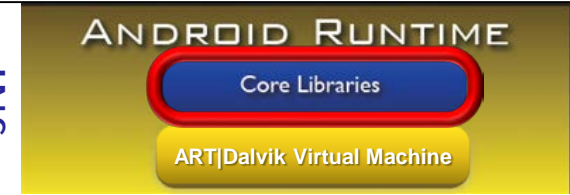


See [en.wikipedia.org/wiki/Thread\\_\(computing\)#Processes.2C\\_kernel\\_threads.2C\\_user\\_threads.2C\\_and\\_fibers](https://en.wikipedia.org/wiki/Thread_(computing)#Processes.2C_kernel_threads.2C_user_threads.2C_and_fibers)

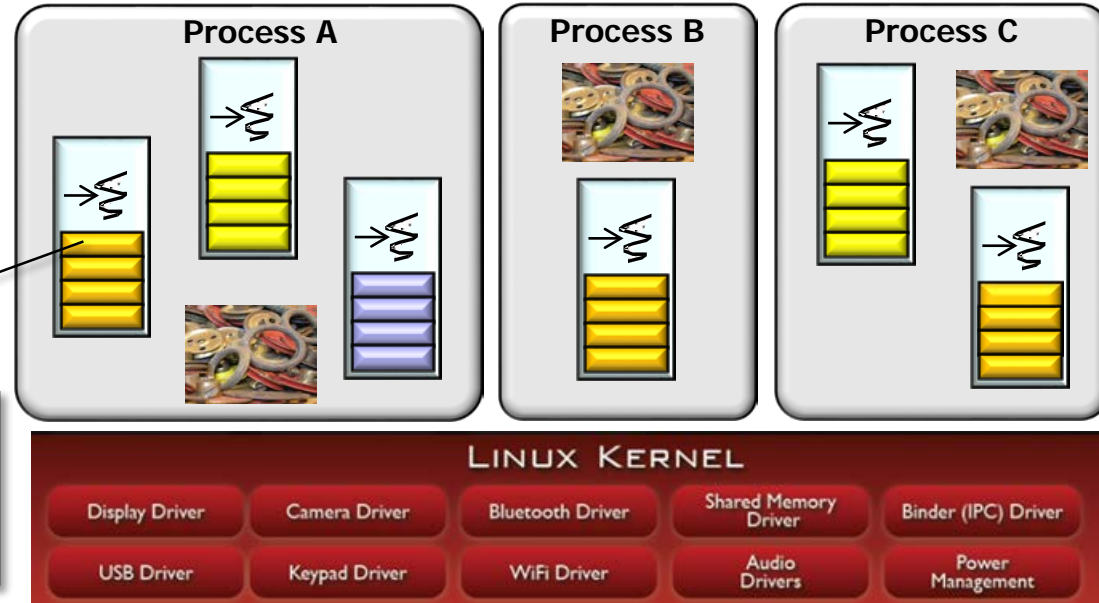
# Android Runtime: Core Java Libraries

- Android contains many (but not all) core Java libraries in the `java.*` & `javax.*` packages, e.g.
  - Java Thread

C/Java/  
JNI



*Java dynamic & static objects can be shared across Java threads (i.e., this "state" is common)*



See [en.wikipedia.org/wiki/Thread\\_\(computing\)#Processes.2C\\_kernel\\_threads.2C\\_user\\_threads.2C\\_and\\_fibers](https://en.wikipedia.org/wiki/Thread_(computing)#Processes.2C_kernel_threads.2C_user_threads.2C_and_fibers)

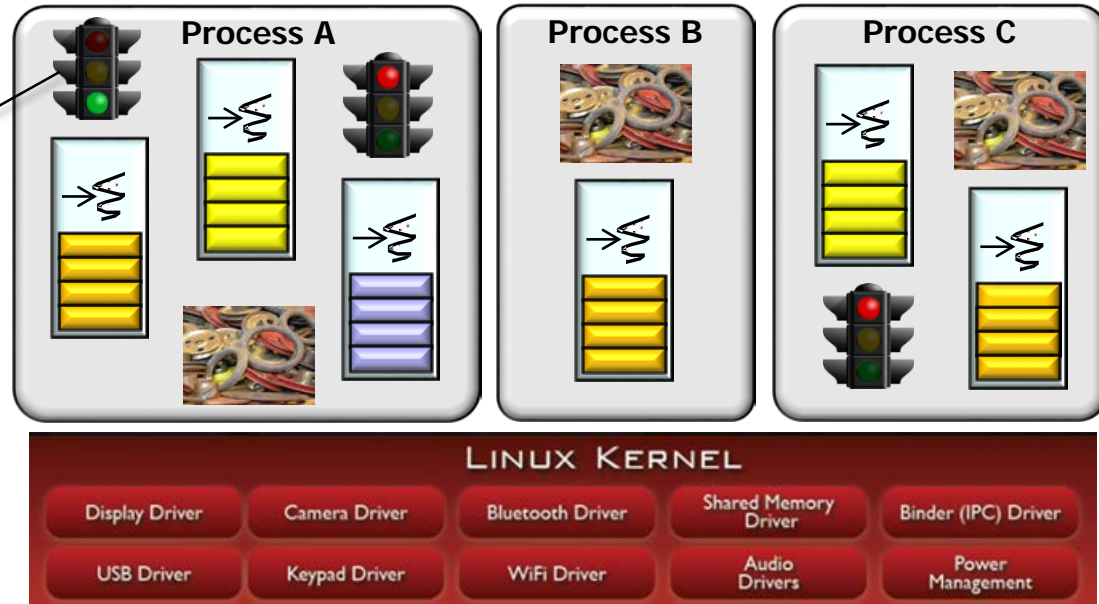
# Android Runtime: Core Java Libraries

- Android contains many (but not all) core Java libraries in the `java.*` & `javax.*` packages, e.g.

C/Java/  
JNI

- Java Thread
- Java synchronizers

*e.g., reentrant locks, stamped locks, semaphores, condition objects, phasers, etc.*



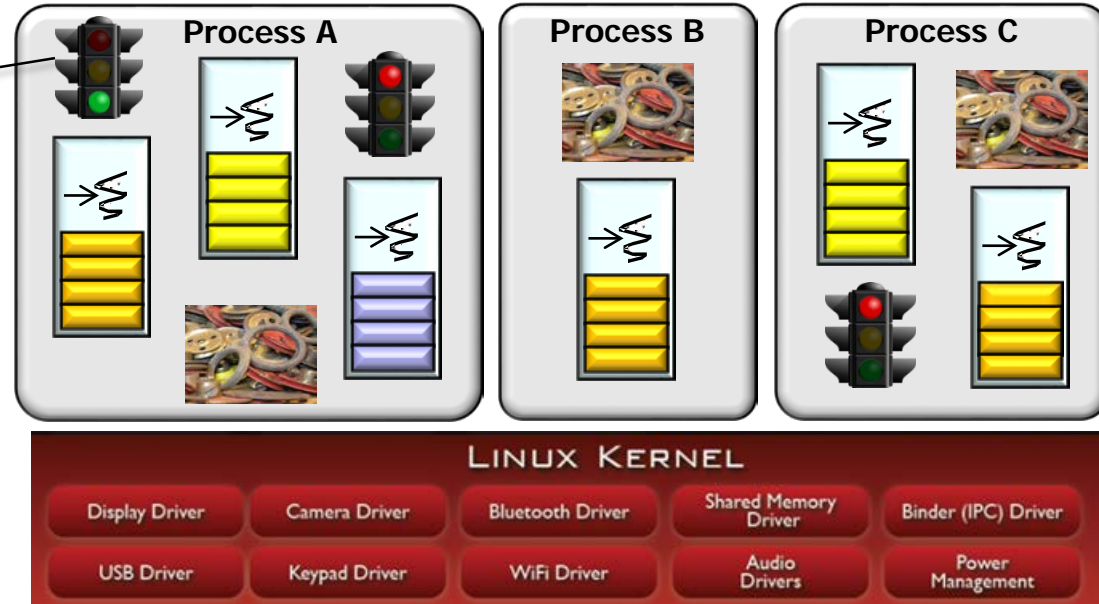
# Android Runtime: Core Java Libraries

- Android contains many (but not all) core Java libraries in the `java.*` & `javax.*` packages, e.g.

C/Java/  
JNI

- Java Thread
- Java synchronizers

*Java synchronizers are used to prevent race conditions*



See [en.wikipedia.org/wiki/Race\\_condition](https://en.wikipedia.org/wiki/Race_condition)

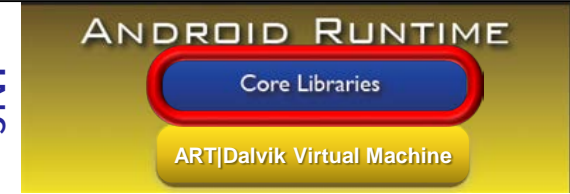


# Android Runtime: Core Java Libraries

- Android contains many (but not all) core Java libraries in the java.\* & javax.\* packages, e.g.

- Java Thread
- Java synchronizers
- Java networking

C/Java/  
JNI



*Java network programming mechanisms can exchange data between Android devices & remote servers*



See [developer.android.com/reference/java/net/URLConnection.html](http://developer.android.com/reference/java/net/URLConnection.html)



# Android Runtime: Core Java Libraries

- Android contains many (but not all) core Java libraries in the `java.*` & `javax.*` packages, e.g.

- Java Thread
- Java synchronizers
- Java networking
- Java I/O & files

C/Java/  
JNI



*Java file mechanisms can store data persistently on Android devices*

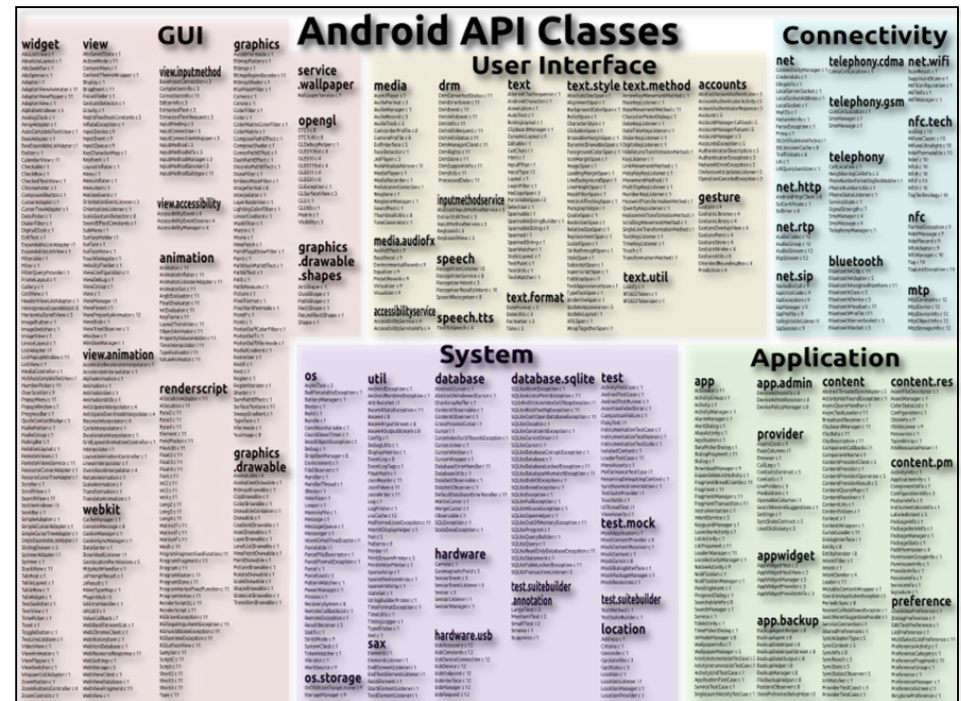
---

# Overview of the Android Runtime: Core Android Libraries

- Android contains thousands of classes in the `android.*` packages



## ART|Dalvik Virtual Machine



See [www.makelinux.net/android/classes](http://www.makelinux.net/android/classes)

# Android Runtime: Core Android Libraries

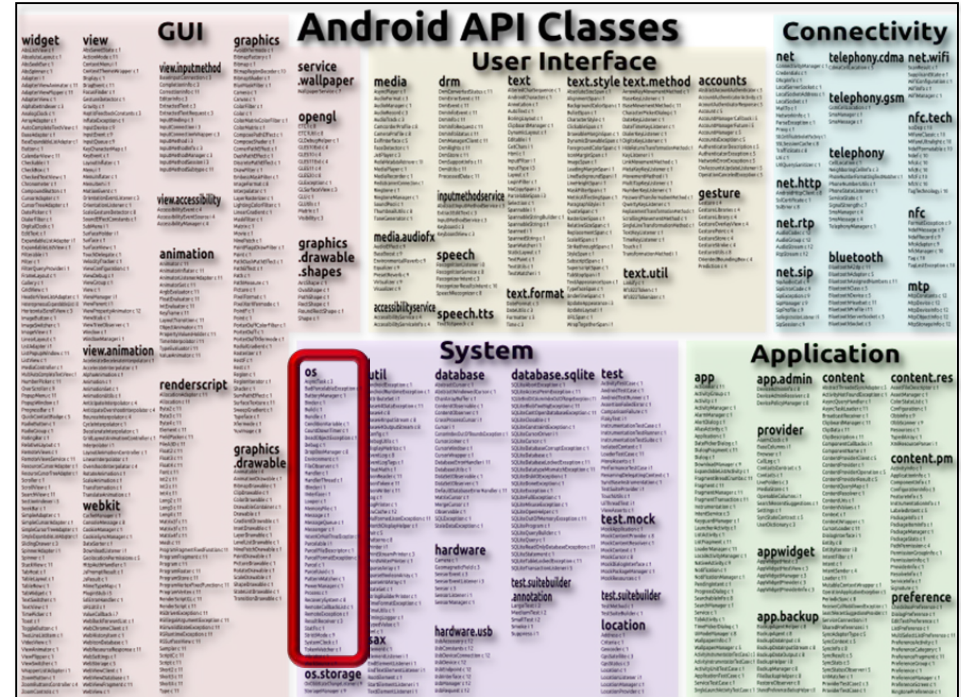
ANDROID RUNTIME

Core Libraries

ART/Dalvik Virtual Machine

C/Java/  
JNI

- Android contains thousands of classes in the android.\* packages, e.g.
- Concurrency

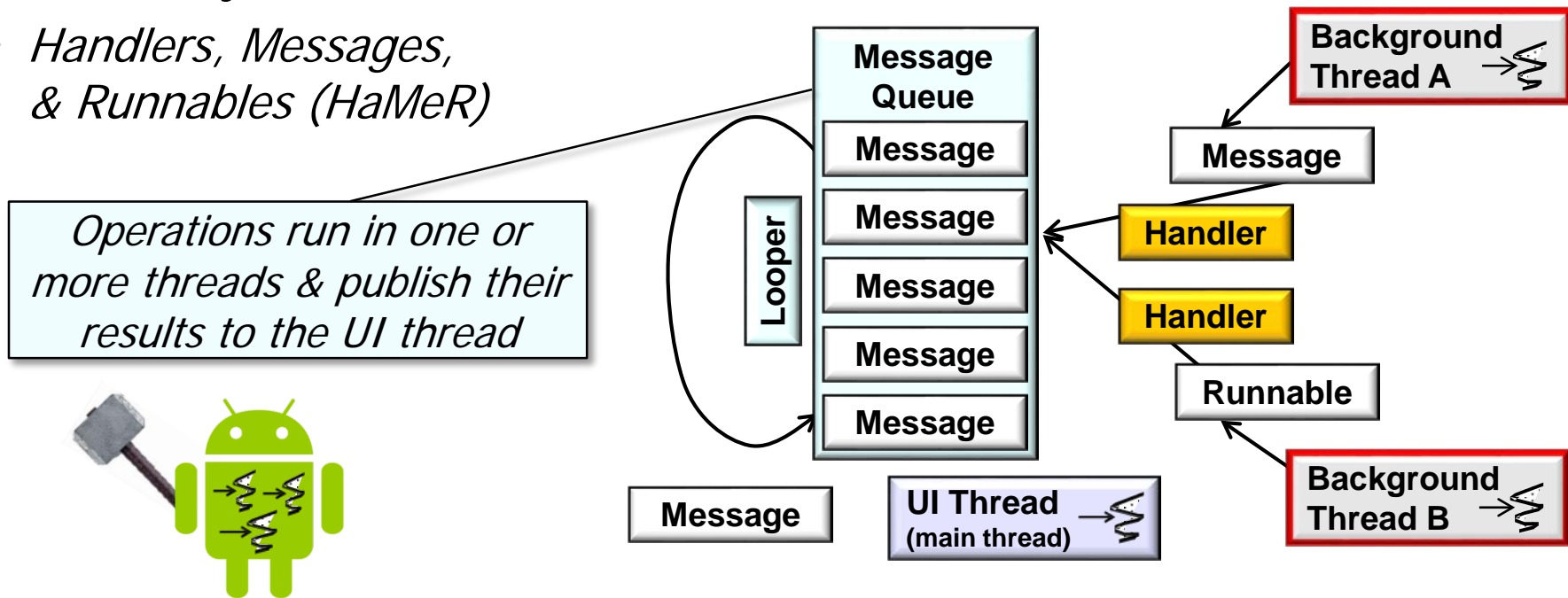
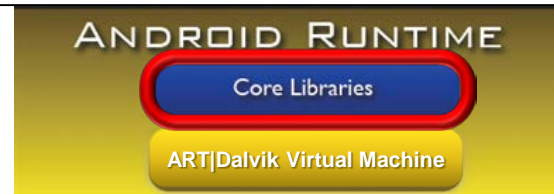


See [www.vogella.com/tutorials/AndroidBackgroundProcessing/article.html](http://www.vogella.com/tutorials/AndroidBackgroundProcessing/article.html)

# Android Runtime: Core Android Libraries

- Android contains thousands of classes in the android.\* packages, e.g.
- Concurrency
  - *Handlers, Messages, & Runnables (HaMeR)*

C/Java/  
JNI

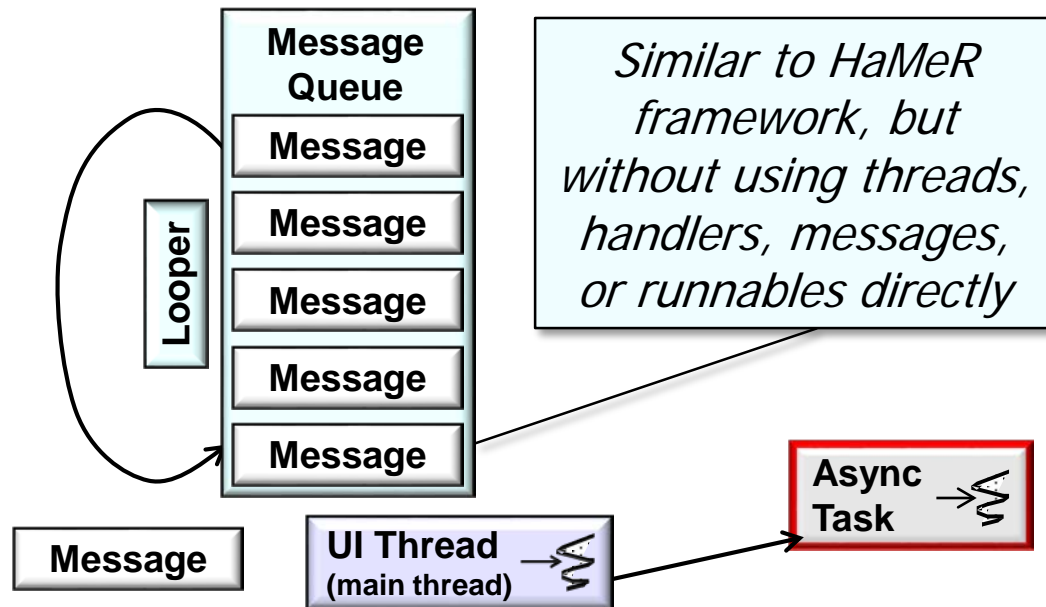
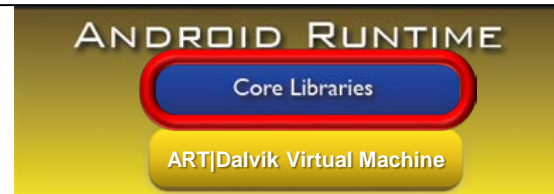


See [developer.android.com/training/multiple-threads/communicate-ui.html](http://developer.android.com/training/multiple-threads/communicate-ui.html)

# Android Runtime: Core Android Libraries

- Android contains thousands of classes in the android.\* packages, e.g.
- Concurrency
  - *Handlers, Messages, & Runnables (HaMeR)*
- *AsyncTask*

C/Java/  
JNI

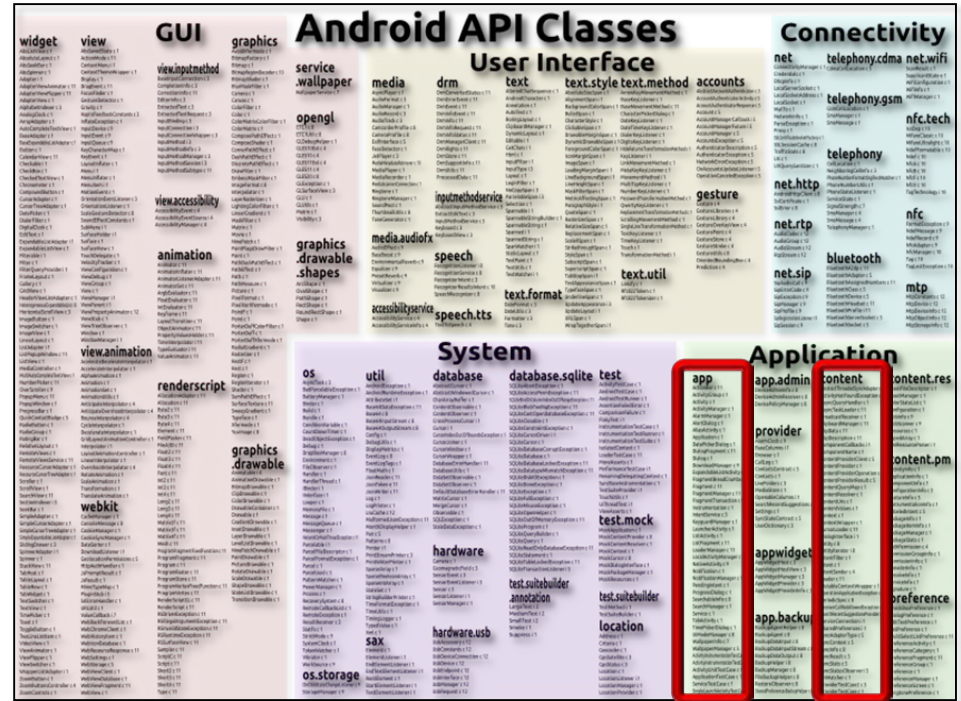


See [developer.android.com/reference/android/os/AsyncTask.html](http://developer.android.com/reference/android/os/AsyncTask.html)



- Android contains thousands of classes in the `android.*` packages, e.g.

- ## C/Java/JNI

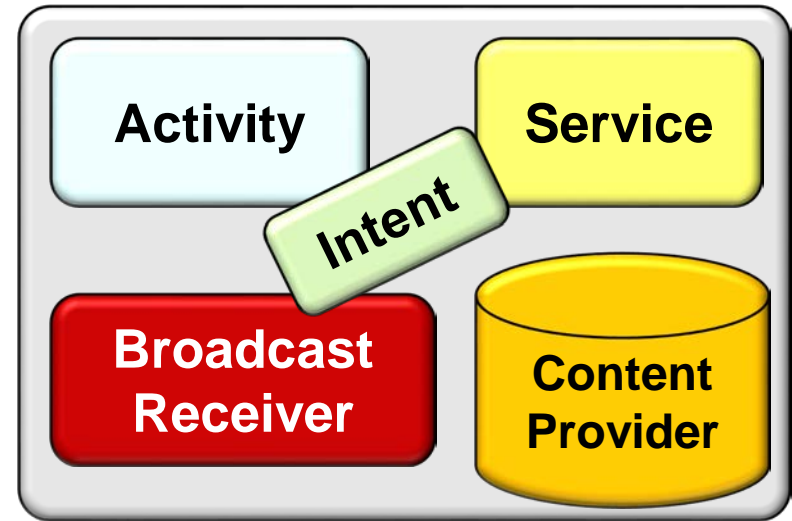
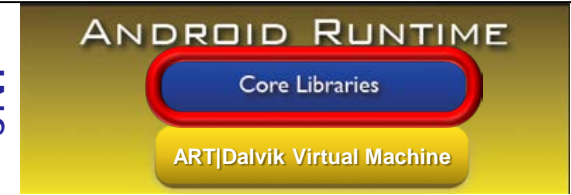




# Android Runtime: Core Android Libraries

- Android contains thousands of classes in the android.\* packages, e.g.
  - Concurrency
  - App components
    - Building blocks of mobile apps that provide hooks that Android uses to control an app's lifecycle

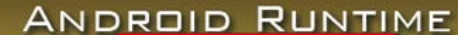
C/Java/  
JNI



See [developer.android.com/guide/components/fundamentals.html#Components](https://developer.android.com/guide/components/fundamentals.html#Components)

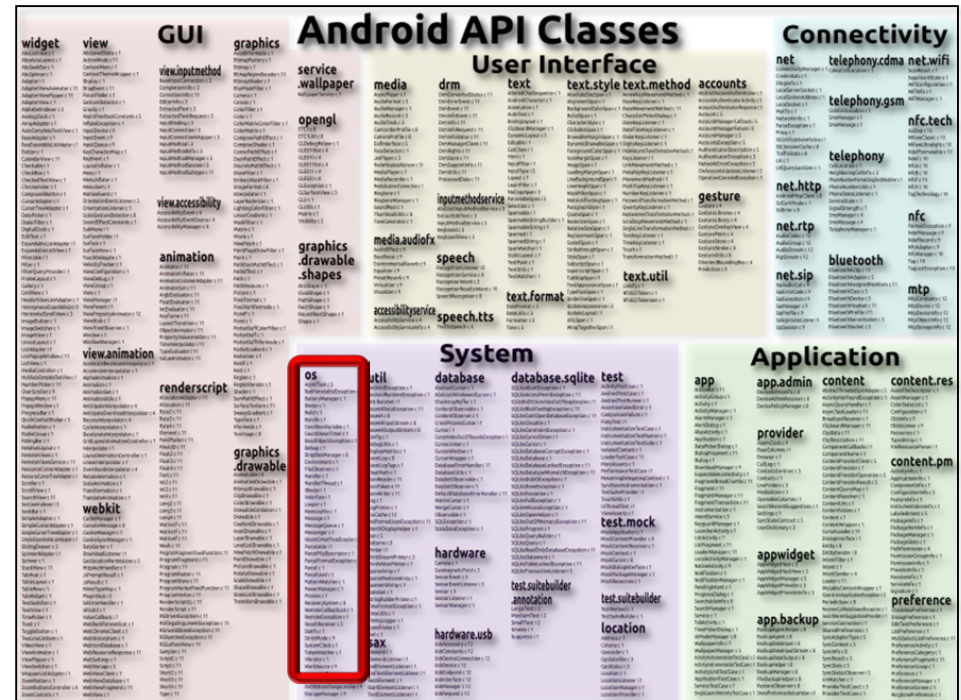
# Android Runtime: Core Android Libraries

- Android contains thousands of classes in the `android.*` packages, e.g.
  - Concurrency
  - App components
  - Binder IPC framework



## Core Libraries

## ART|Dalvik Virtual Machine



# Android Runtime: Core Android Libraries

- Android contains thousands of classes in the android.\* packages, e.g.

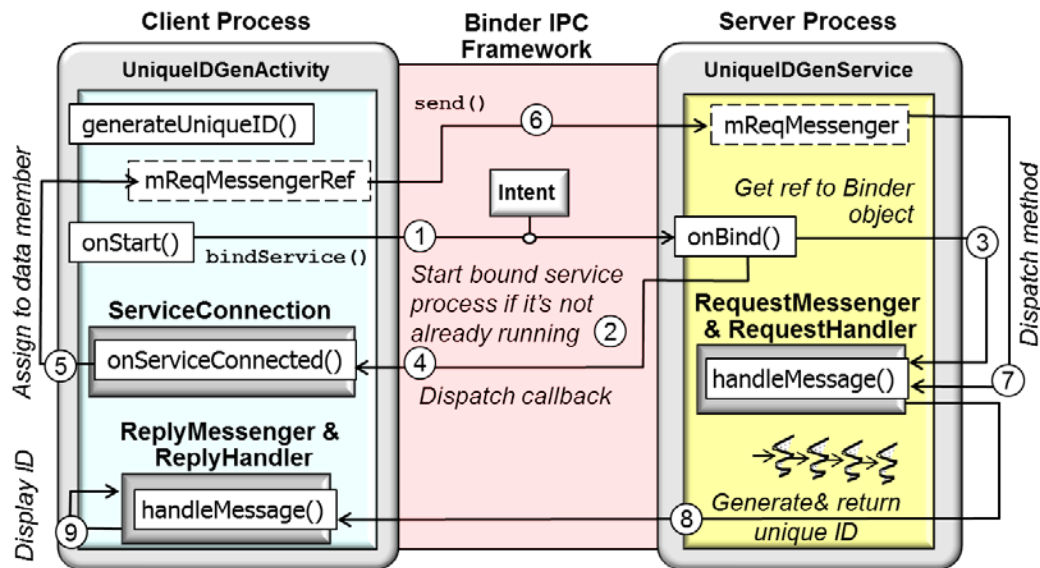
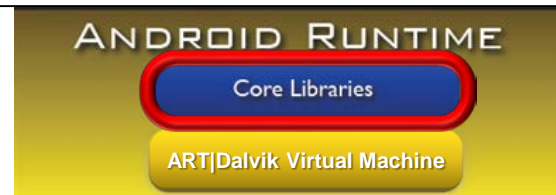
- Concurrency

- App components

- Binder IPC framework

- Enables sync & async communication between components on a device

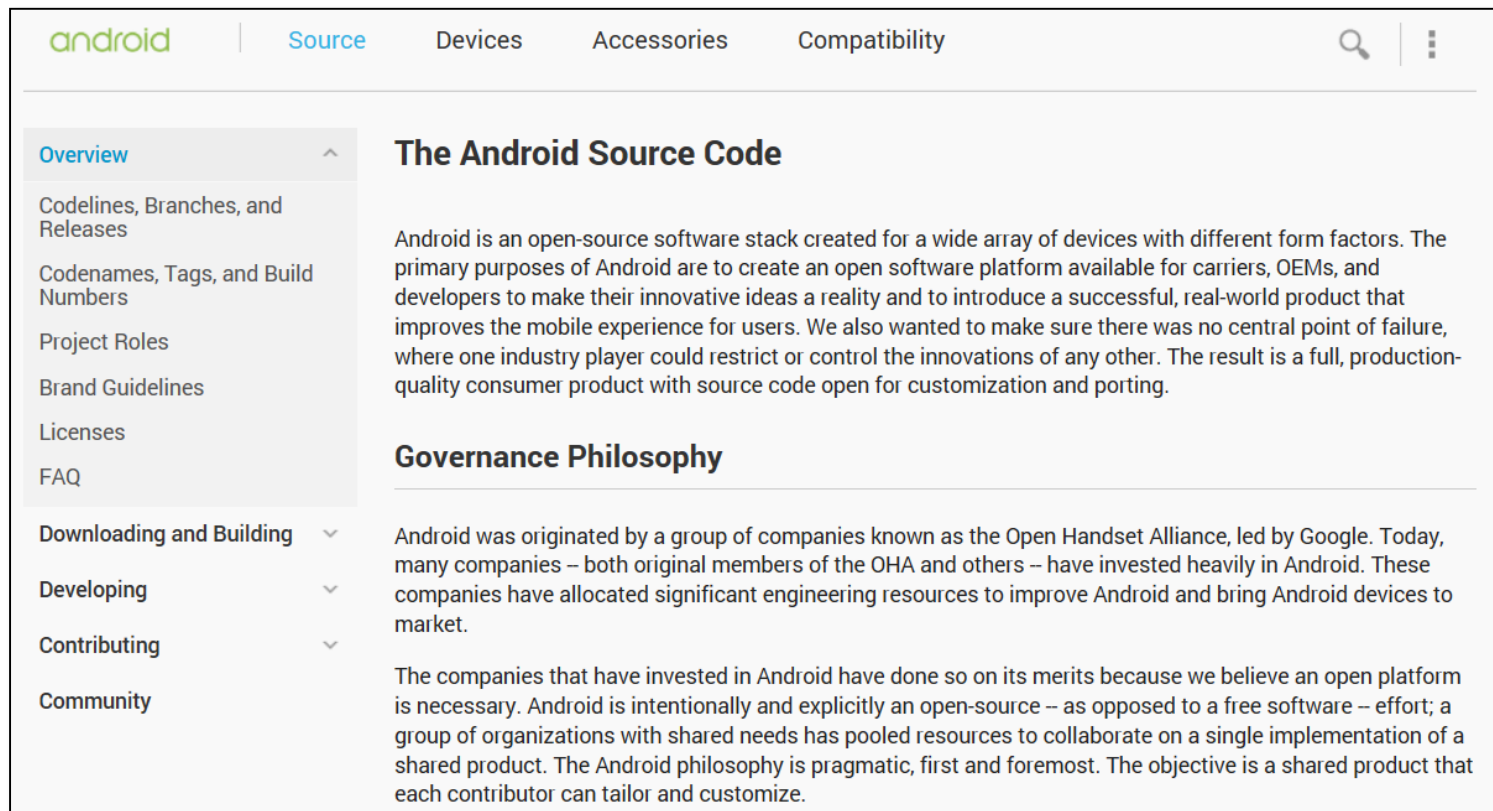
C/Java/  
JNI



See [elinux.org/Android\\_Binder](http://elinux.org/Android_Binder)

# Android Runtime: Core Android Libraries

- The source code for all the core Java & Android libraries is available online

A screenshot of the 'Source' page on the Android website. The page has a navigation bar at the top with the Android logo, 'Source' (highlighted in blue), 'Devices', 'Accessories', and 'Compatibility'. On the left is a sidebar menu with links: Overview (selected), Codelines, Branches, and Releases, Codenames, Tags, and Build Numbers, Project Roles, Brand Guidelines, Licenses, FAQ, Downloading and Building, Developing, Contributing, and Community. The main content area is titled 'The Android Source Code' and contains two paragraphs. The first paragraph describes Android as an open-source software stack created for various devices, emphasizing its primary purposes and the goal of a successful, real-world product. The second paragraph, under the heading 'Governance Philosophy', explains that Android was originated by the Open Handset Alliance, led by Google, and that many companies have invested heavily in it to improve and bring devices to market. It also states that the companies have invested in Android because they believe an open platform is necessary and that the Android philosophy is pragmatic, first and foremost, with the objective of a shared product that can be tailored and customized.

android | Source | Devices | Accessories | Compatibility

**Overview**

- Codelines, Branches, and Releases
- Codenames, Tags, and Build Numbers
- Project Roles
- Brand Guidelines
- Licenses
- FAQ
- Downloading and Building
- Developing
- Contributing
- Community

## The Android Source Code

Android is an open-source software stack created for a wide array of devices with different form factors. The primary purposes of Android are to create an open software platform available for carriers, OEMs, and developers to make their innovative ideas a reality and to introduce a successful, real-world product that improves the mobile experience for users. We also wanted to make sure there was no central point of failure, where one industry player could restrict or control the innovations of any other. The result is a full, production-quality consumer product with source code open for customization and porting.

### Governance Philosophy

Android was originated by a group of companies known as the Open Handset Alliance, led by Google. Today, many companies – both original members of the OHA and others – have invested heavily in Android. These companies have allocated significant engineering resources to improve Android and bring Android devices to market.

The companies that have invested in Android have done so on its merits because we believe an open platform is necessary. Android is intentionally and explicitly an open-source – as opposed to a free software – effort; a group of organizations with shared needs has pooled resources to collaborate on a single implementation of a shared product. The Android philosophy is pragmatic, first and foremost. The objective is a shared product that each contributor can tailor and customize.

See [source.android.com](https://source.android.com)

---

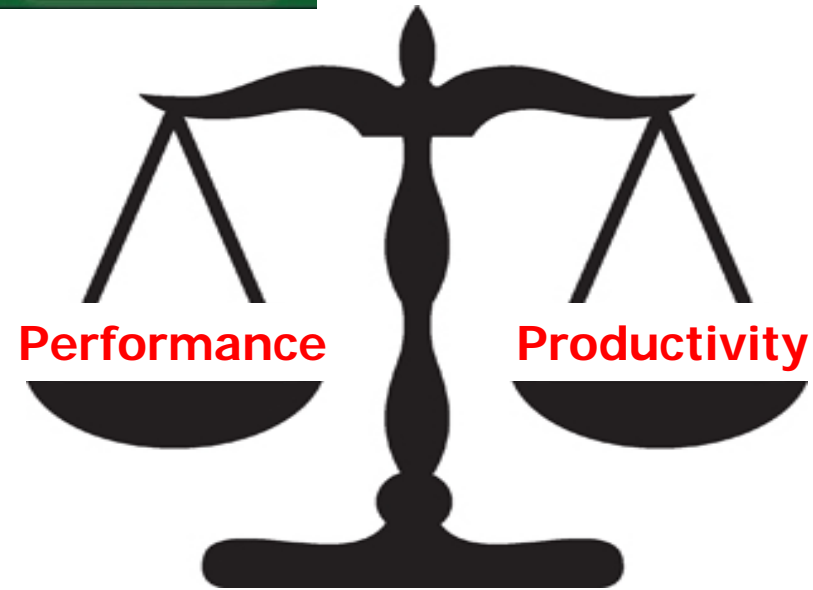
# Overview of Android Native C/C++ Libraries

# Android Native C/C++ Libraries

C++/C



- Although Android apps are written using Java APIs, *implementations* of these APIs are often written in C/C++



Goal is to enhance system performance w/out sacrificing developer productivity

# Android Native C/C++ Libraries



- Although Android apps are written using Java APIs, *implementations* of these APIs are often written in C/C++
- Java & C/C++ are combined via the Java Native Interface (JNI)

## JNI Tips

JNI is the Java Native Interface. It defines a way for managed code (written in the Java programming language) to interact with native code (written in C/C++). It's vendor-neutral, has support for loading code from dynamic shared libraries, and while cumbersome at times is reasonably efficient.

If you're not already familiar with it, read through the [Java Native Interface Specification](#) to get a sense for how JNI works and what features are available. Some aspects of the interface aren't immediately obvious on first reading, so you may find the next few sections handy.

See [developer.android.com/training/articles/perf-jni.html](https://developer.android.com/training/articles/perf-jni.html)



# Android Native C/C++ Libraries



- Although Android apps are written using Java APIs, *implementations* of these APIs are often written in C/C++
  - Java & C/C++ are combined via the Java Native Interface (JNI)
  - JNI defines a standard way for managed Java code to interact with native code written in C/C++

## JNI Tips

JNI is the Java Native Interface. It defines a way for managed code (written in the Java programming language) to interact with native code (written in C/C++). It's vendor-neutral, has support for loading code from dynamic shared libraries, and while cumbersome at times is reasonably efficient.

If you're not already familiar with it, read through the [Java Native Interface Specification](#) to get a sense for how JNI works and what features are available. Some aspects of the interface aren't immediately obvious on first reading, so you may find the next few sections handy.

# Android Native C/C++ Libraries

C++/C



- Android's Native Development Kit (NDK) allows the implementation of apps & services using native C/C++ code

See [developer.android.com/ndk](https://developer.android.com/ndk)

# Android Native C/C++ Libraries

C++/C



- Android's Native Development Kit (NDK) allows the implementation of apps & services using native C/C++ code
- Using the NDK on *portions* of code can help enhance performance by minimizing latency, maximizing throughput, & conserving key system resources



Resist the urge to develop all of your apps using the NDK!

# Android Native C/C++ Libraries



- Android's Native Development Kit (NDK) allows the implementation of apps & services using native C/C++ code
  - Using the NDK on *portions* of code can help enhance performance by minimizing latency, maximizing throughput, & conserving key system resources
- It can also be used to integrate existing C/C++ libraries into Android apps



# Android Native C/C++ Libraries



- **System C library**
  - bionic libc
- **Surface Manager**
  - display management
- **Media Framework**
  - audio/video streaming
- **FreeType**
  - library for rendering fonts
- **Webkit**
  - web browser engine
- **OpenGL ES, SGL**
  - graphics engines
- **SQLite**
  - relational database engine
- **SSL**
  - secure sockets layer

Android native libraries are open source & often have Java wrapper facades

# Android Native C/C++ Libraries



- **System C library**

- bionic libc

- **Surface Manager**

- display management

- **Media Framework**

- audio/video streaming

- **FreeType**

- library for rendering fonts

- **Webkit**

- web browser engine

- **OpenGL ES, SGL**

- graphics engines

- **SQLite**

- relational database engine

- **SSL**

- secure sockets layer

Enables developers to write native system services for Android

# Android Native C/C++ Libraries



- **System C library**
  - bionic libc
- **Surface Manager**
  - display management
- **Media Framework**
  - audio/video streaming
- **FreeType**
  - library for rendering fonts
- **Webkit**
  - web browser engine
- **OpenGL ES, SGL**
  - graphics engines
- **SQLite**
  - relational database engine
- **SSL**
  - secure sockets layer

Composites 2D & 3D graphic layers from multiple apps



# Android Native C/C++ Libraries

C++/C



- **System C library**
  - bionic libc
- **Surface Manager**
  - display management
- **Media Framework**
  - audio/video streaming
- **FreeType**
  - library for rendering fonts
- **Webkit**
  - web browser engine
- **OpenGL ES, SGL**
  - graphics engines
- **SQLite**
  - relational database engine
- **SSL**
  - secure sockets layer

Supports audio-video streaming in the background

# Android Native C/C++ Libraries



- **System C library**

- bionic libc

- **Surface Manager**

- display management

- **Media Framework**

- audio/video streaming

- **FreeType**

- library for rendering fonts

- **Webkit**

- web browser engine

- **OpenGL ES, SGL**

- graphics engines

- **SQLite**

- relational database engine

- **SSL**

- secure sockets layer

Renders bitmap & vector fonts

# Android Native C/C++ Libraries

C++/C



- **System C library**
  - bionic libc
- **Surface Manager**
  - display management
- **Media Framework**
  - audio/video streaming
- **FreeType**
  - library for rendering fonts
- **Webkit**
  - web browser engine
- **OpenGL ES, SGL**
  - graphics engines
- **SQLite**
  - relational database engine
- **SSL**
  - secure sockets layer

Framework used on mobile & non-mobile platforms for web browsing

# Android Native C/C++ Libraries



- **System C library**
  - bionic libc
- **Surface Manager**
  - display management
- **Media Framework**
  - audio/video streaming
- **FreeType**
  - library for rendering fonts
- **Webkit**
  - web browser engine
- **OpenGL ES, SGL**
  - graphics engines
- **SQLite**
  - relational database engine
- **SSL**
  - secure sockets layer

Supports 2D & 3D vector graphics, e.g., often used for gaming apps

# Android Native C/C++ Libraries



- **System C library**
  - bionic libc
- **Surface Manager**
  - display management
- **Media Framework**
  - audio/video streaming
- **FreeType**
  - library for rendering fonts
- **Webkit**
  - web browser engine
- **OpenGL ES, SGL**
  - graphics engines
- **SQLite**
  - relational database engine
- **SSL**
  - secure sockets layer

Relational database engine that performs CRUD operations on persistent data

# Android Native C/C++ Libraries



- **System C library**

- bionic libc

- **Surface Manager**

- display management

- **Media Framework**

- audio/video streaming

- **FreeType**

- library for rendering fonts

- **Webkit**

- web browser engine

- **OpenGL ES, SGL**

- graphics engines

- **SQLite**

- relational database engine

- **SSL**

- secure sockets layer

Ensures confidentiality & integrity for web interactions (e.g., e-commerce)



# Android Native C/C++ Libraries



- **System C library**
  - bionic libc
- **Surface Manager**
  - display management
- **Media Framework**
  - audio/video streaming
- **FreeType**
  - library for rendering fonts
- **Webkit**
  - web browser engine
- **OpenGL ES, SGL**
  - graphics engines
- **SQLite**
  - relational database engine
- **SSL**
  - secure sockets layer

Native C/C++ libraries use non-Java concurrency libraries, e.g., POSIX pthreads

---

# End of Infrastructure Middleware (Part 3): the Android Runtime Core & Native Libraries