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

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

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

See 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

See 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

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
Android Runtime: Core Java Libraries

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

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

See 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

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
Android Runtime: Core Java Libraries

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

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
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

  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.
  - Java Thread
  - Java synchronizers

Java synchronizers are used to prevent race conditions

See 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

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

See developer.android.com/reference/java/net/HttpURLConnection.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

Java file mechanisms can store data persistently on Android devices.

See docs.oracle.com/javase/8/docs/api/java/nio/file/Files.html
Overview of the Android Runtime: Core Android Libraries
Android Runtime: Core Android Libraries

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

See [www.makelinuc.net/android/classes](http://www.makelinuc.net/android/classes)
Android Runtime: Core Android Libraries

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

See 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)*

  Operations run in one or more threads & publish their results to the UI thread

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

See developer.android.com/reference/android/os/AsyncTask.html
Android Runtime: Core Android Libraries

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

See 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
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

See elinux.org/Android_Binder
Android Runtime: Core Android Libraries

- The source code for all the core Java & Android libraries is available online. See source.android.com
Overview of Android Native C/C++ Libraries
Android Native C/C++ Libraries

- 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
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).

See [developer.android.com/training/articles/perf-jni.html](http://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’s Native Development Kit (NDK) allows the implementation of apps & services using native C/C++ code.
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'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

- **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

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

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