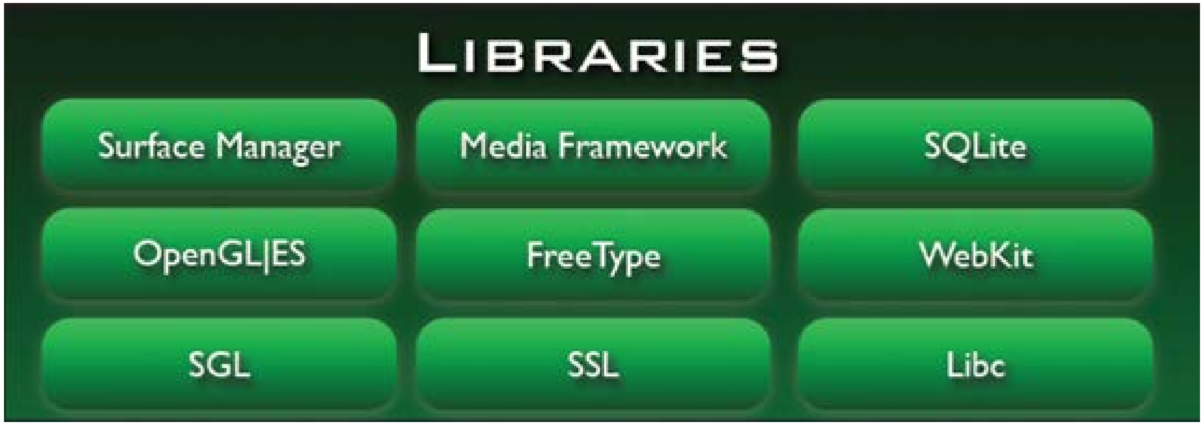
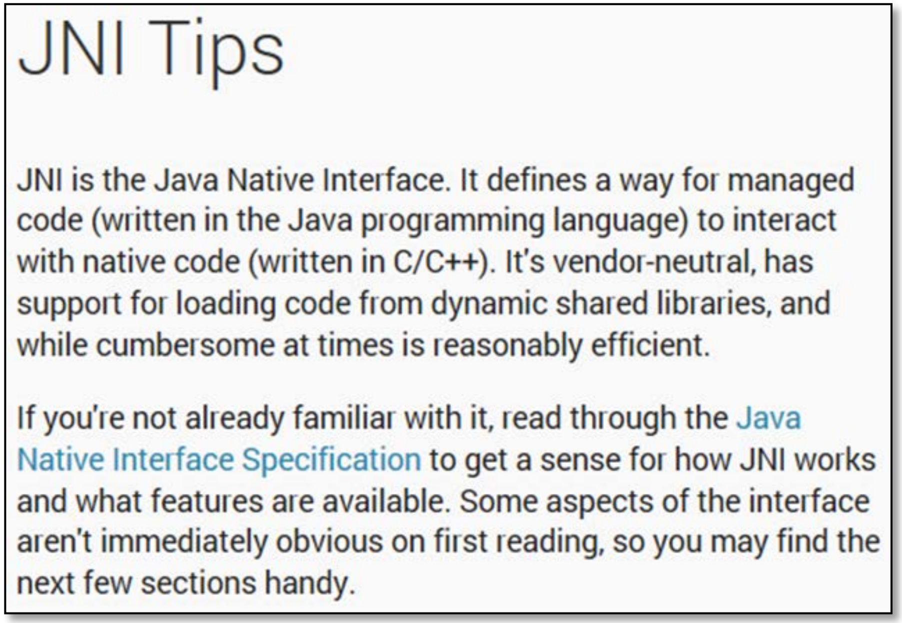
Overview of Android Native C/C++ Libraries

Android Native C/C++ Libraries

C++/C



- Although Android apps are written using Java APIs, the *implementations* of these APIs are often written using native C & C++ available via the JNI



JNI Tips

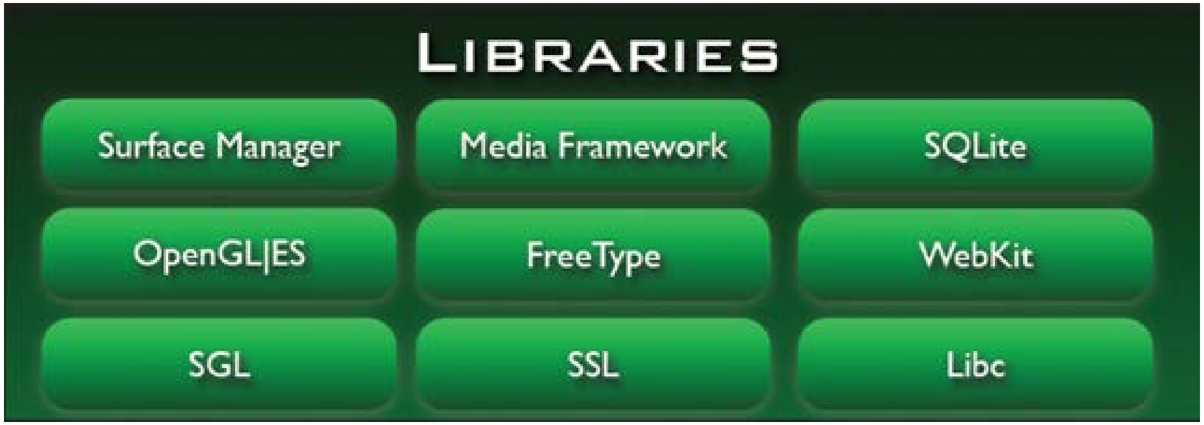
JNI is the Java Native Interface. It defines a way for managed code (written in the Java programming language) to interact with native code (written in C/C++). It's vendor-neutral, has support for loading code from dynamic shared libraries, and while cumbersome at times is reasonably efficient.

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.

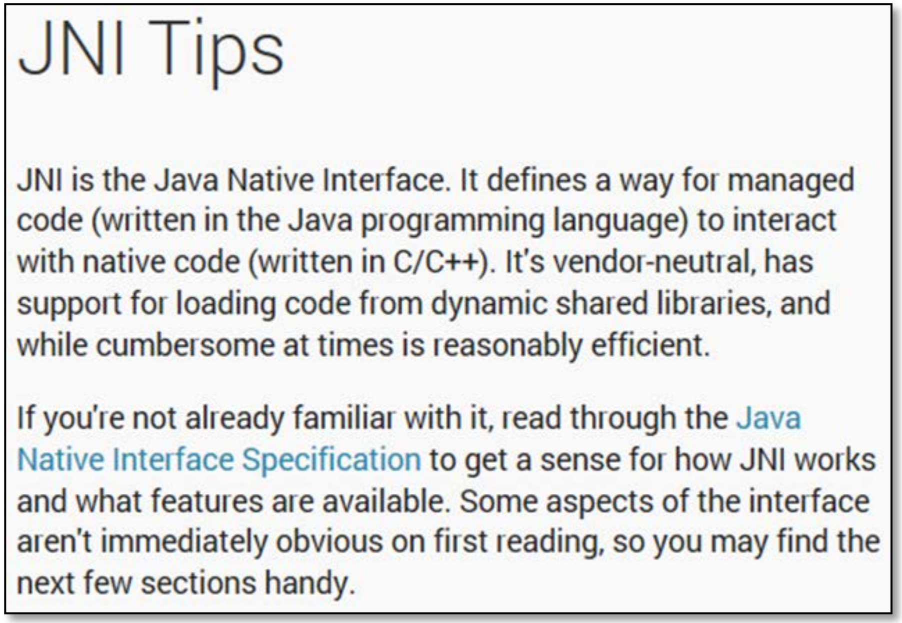
See developer.android.com/training/articles/perf-jni.html

Android Native C/C++ Libraries

C++/C



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

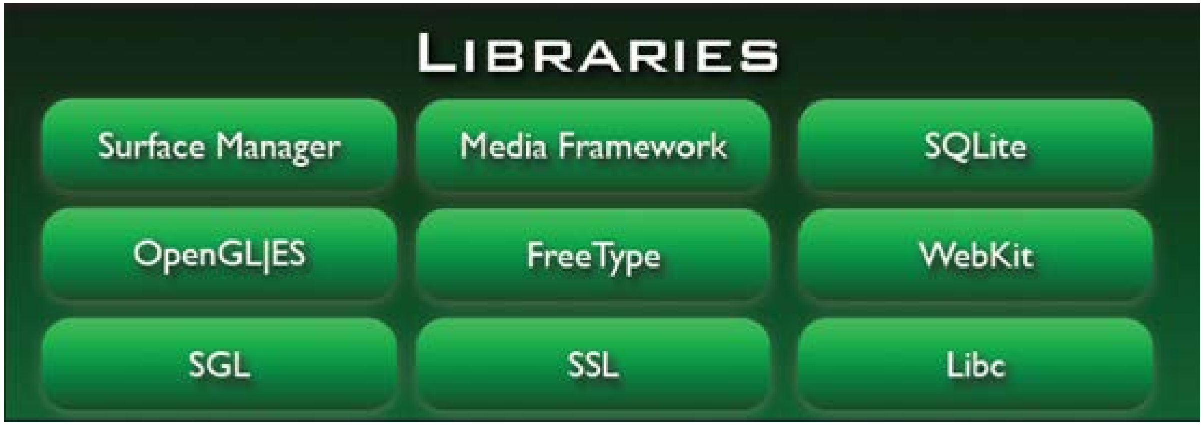


JNI Tips

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.

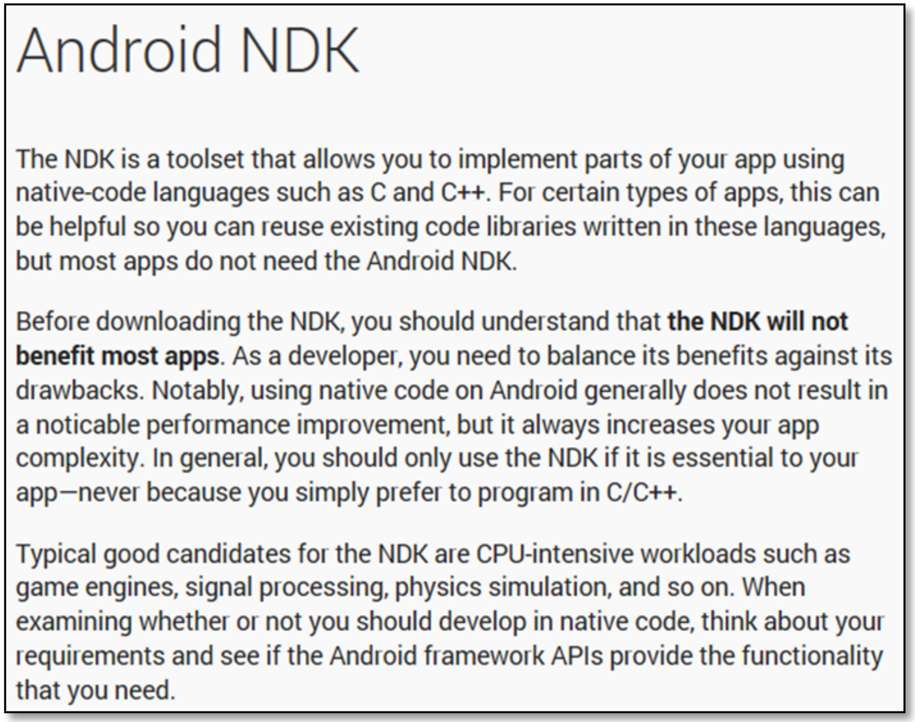
If you're not already familiar with it, read through the Java Native Interface Specification to get a sense for how JNI works and what features are available. Some aspects of the interface aren't immediately obvious on first reading, so you may find the next few sections handy.

Android Native C/C++ Libraries



C++/C

- Although Android apps are written using Java APIs, the *implementations* of these APIs are often written using native C & C++ available via the JNI
- The NDK allows the implementation of apps using native-code languages, such as C & C++



Android NDK

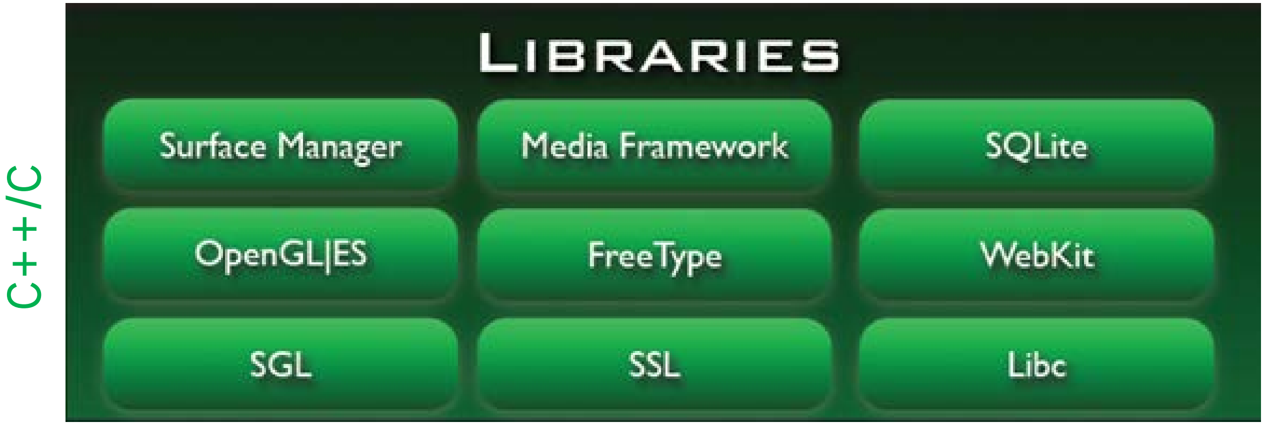
The NDK is a toolset that allows you to implement parts of your app using native-code languages such as C and C++. For certain types of apps, this can be helpful so you can reuse existing code libraries written in these languages, but most apps do not need the Android NDK.

Before downloading the NDK, you should understand that **the NDK will not benefit most apps**. As a developer, you need to balance its benefits against its drawbacks. Notably, using native code on Android generally does not result in a noticable performance improvement, but it always increases your app complexity. In general, you should only use the NDK if it is essential to your app—never because you simply prefer to program in C/C++.

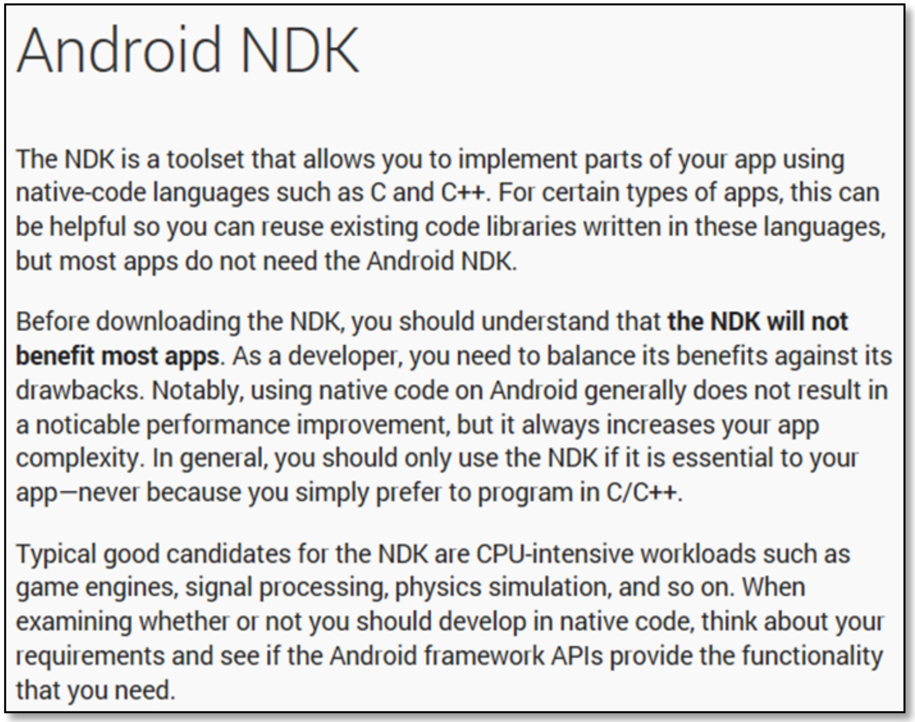
Typical good candidates for the NDK are CPU-intensive workloads such as game engines, signal processing, physics simulation, and so on. When examining whether or not you should develop in native code, think about your requirements and see if the Android framework APIs provide the functionality that you need.

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

Android Native C/C++ Libraries



- Although Android apps are written using Java APIs, the *implementations* of these APIs are often written using native C & C++ available via the JNI
- The NDK allows the implementation of apps using native-code languages, such as C & C++
  - May optimize performance by minimizing latency, maximizing throughput, & conserving resources



Android NDK

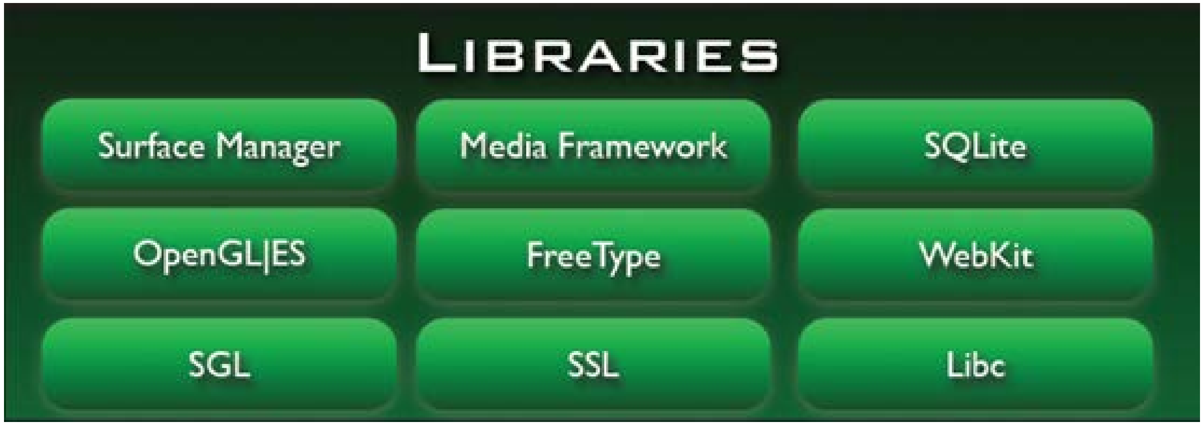
The NDK is a toolset that allows you to implement parts of your app using native-code languages such as C and C++. For certain types of apps, this can be helpful so you can reuse existing code libraries written in these languages, but most apps do not need the Android NDK.

Before downloading the NDK, you should understand that **the NDK will not benefit most apps**. As a developer, you need to balance its benefits against its drawbacks. Notably, using native code on Android generally does not result in a noticable performance improvement, but it always increases your app complexity. In general, you should only use the NDK if it is essential to your app—never because you simply prefer to program in C/C++.

Typical good candidates for the NDK are CPU-intensive workloads such as game engines, signal processing, physics simulation, and so on. When examining whether or not you should develop in native code, think about your requirements and see if the Android framework APIs provide the functionality that you need.

Android Native C/C++ Libraries

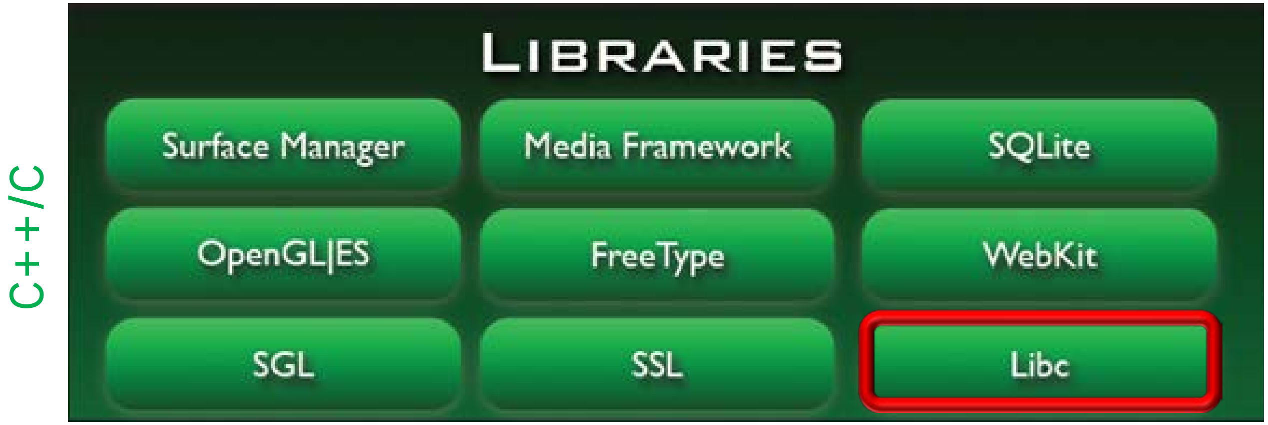
C++/C



- **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

Many native libraries are available on Android

Android Native C/C++ Libraries

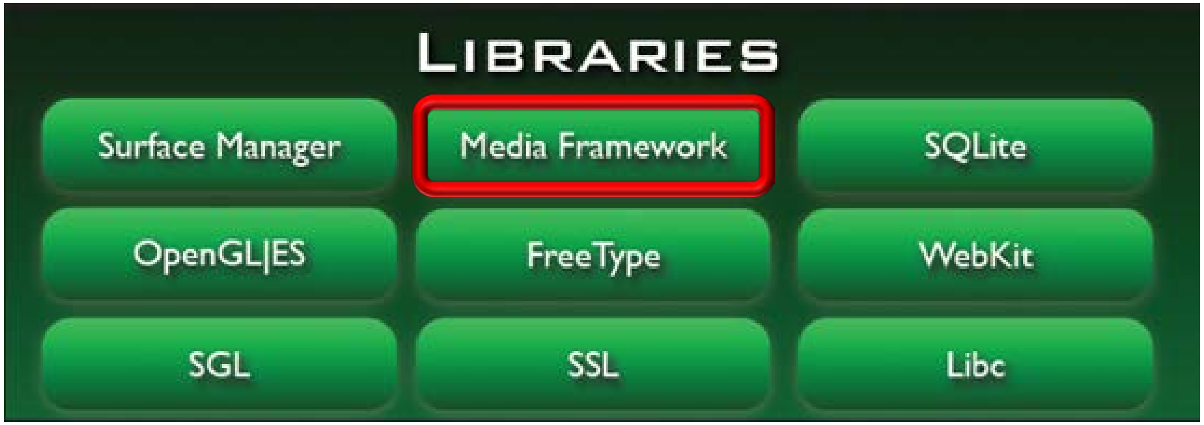


- **System C library**
  - bionic libc
- **Surface Manager**
  - display management
- **Media Framework**
  - audio/video streaming
- **FreeType**
  - library for rendering fonts

- **Webkit**
  - web browser engine
- **OpenGL ES, SGL**
  - graphics engines
- **SQLite**
  - relational database engine
- **SSL**
  - secure sockets layer

Enables developers to write native system services for Android

Android Native C/C++ Libraries

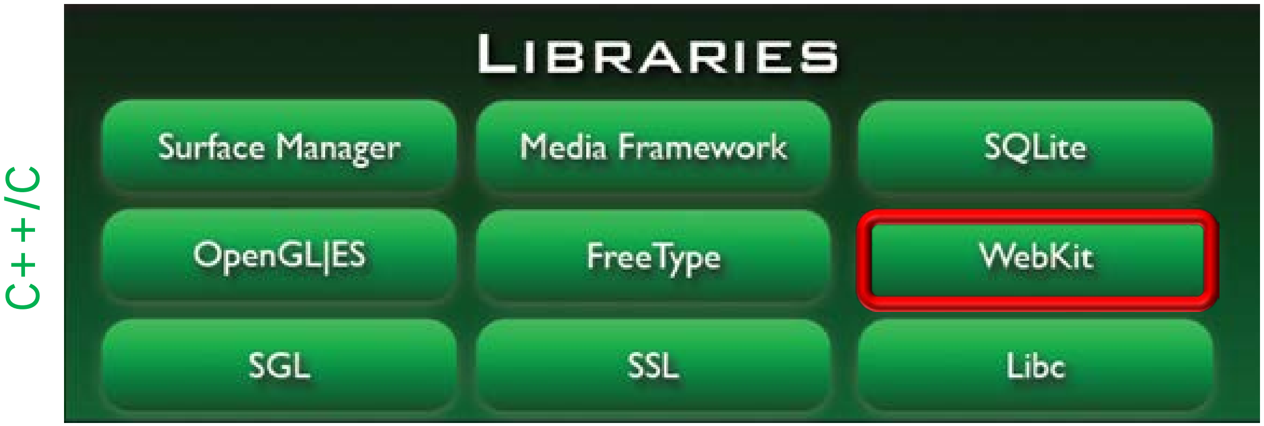


C++/C

- **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

Multi-media frameworks
for audio-video streaming

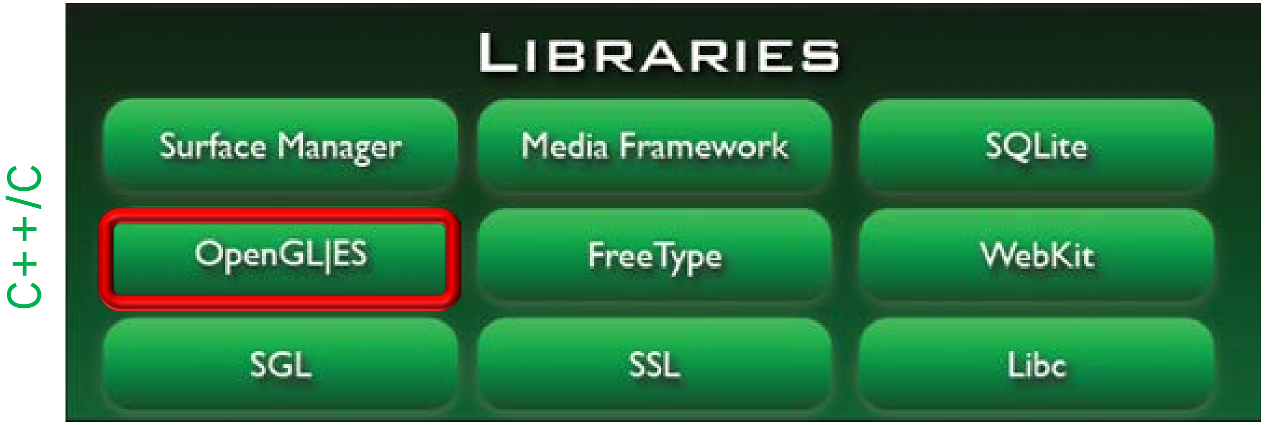
Android Native C/C++ Libraries



- **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

Open-source framework that's used on many mobile platforms for browser interfaces/interactions

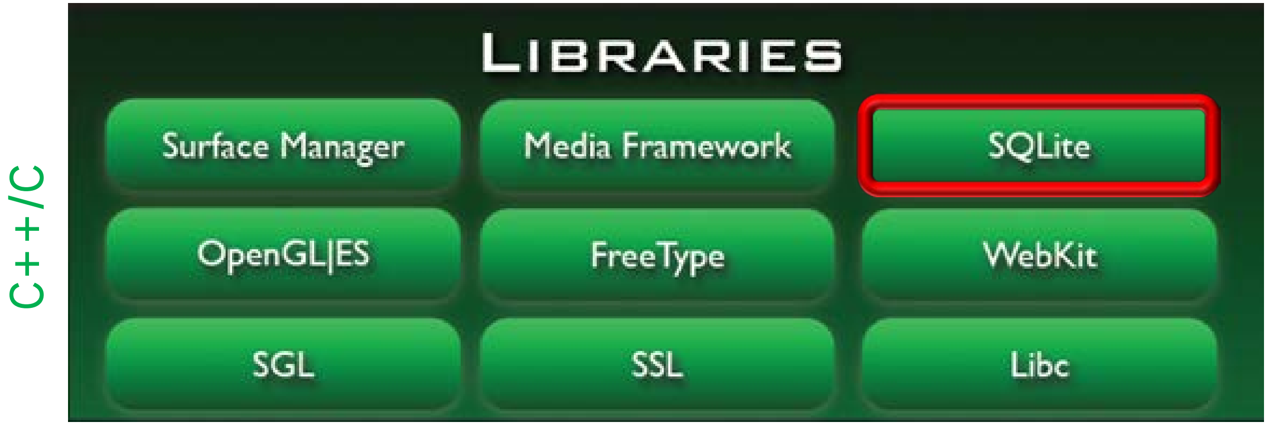
Android Native C/C++ Libraries



- **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

2D & 3D vector graphics
using Open GL

Android Native C/C++ Libraries

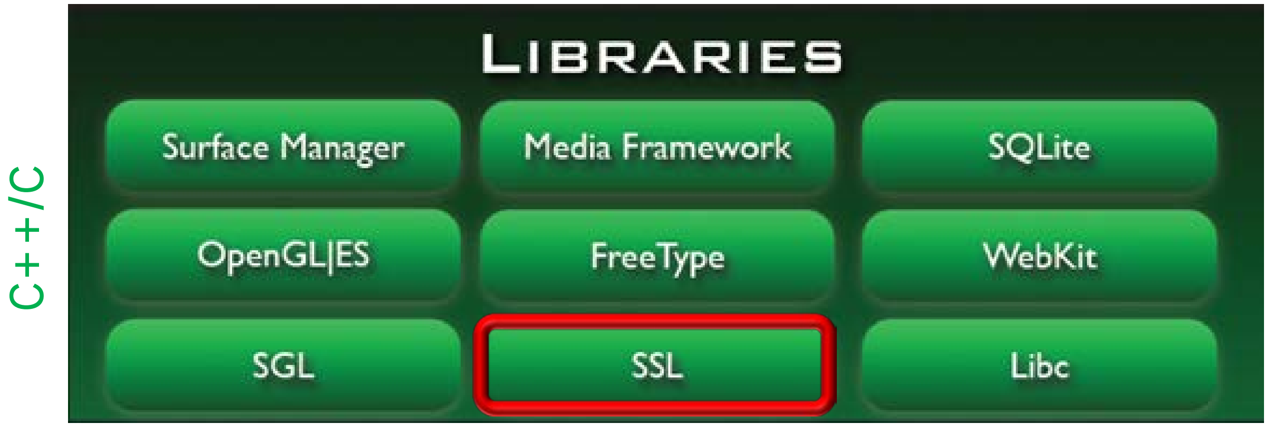


- **System C library**
  - bionic libc
- **Surface Manager**
  - display management
- **Media Framework**
  - audio/video streaming
- **FreeType**
  - library for rendering fonts

- **Webkit**
  - web browser engine
- **OpenGL ES, SGL**
  - graphics engines
- **SQLite**
  - relational database engine
- **SSL**
  - secure sockets layer

Relational database engine used to store/retrieve information persistently

Android Native C/C++ Libraries

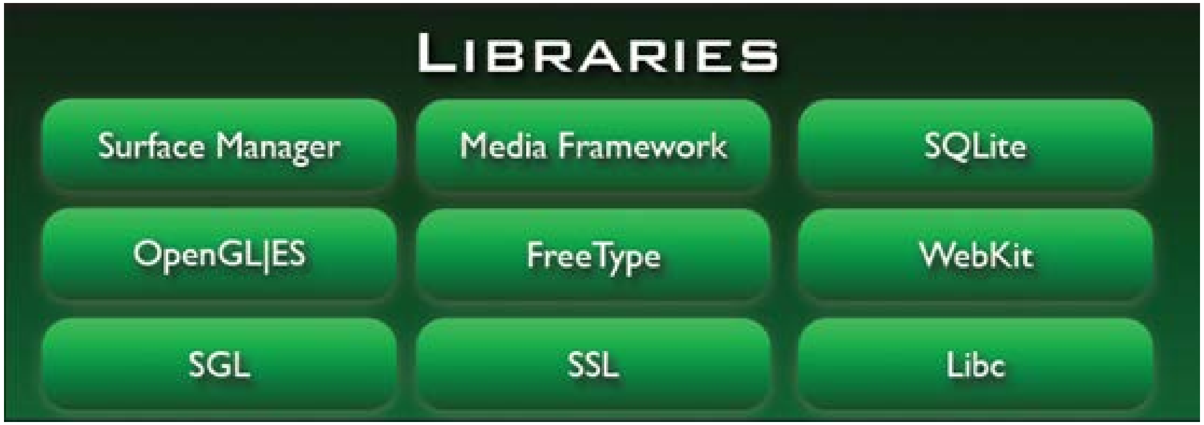


- **System C library**
  - bionic libc
- **Surface Manager**
  - display management
- **Media Framework**
  - audio/video streaming
- **FreeType**
  - library for rendering fonts

- **Webkit**
  - web browser engine
- **OpenGL ES, SGL**
  - graphics engines
- **SQLite**
  - relational database engine
- **SSL**
  - secure sockets layer

Ensures the confidentiality & integrity for various types of e-commerce interactions

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

Native C/C++ libraries use a range of (non-Java) concurrency mechanisms