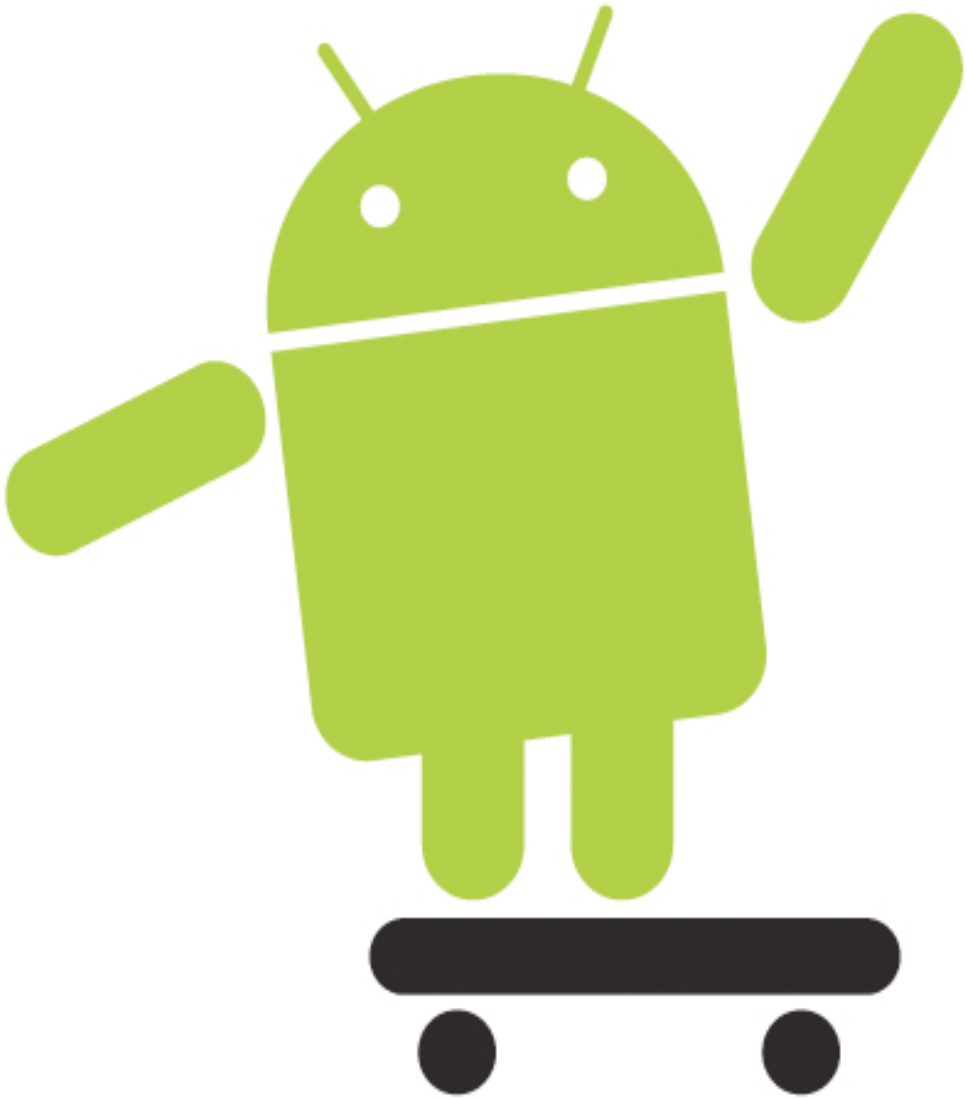


marakana

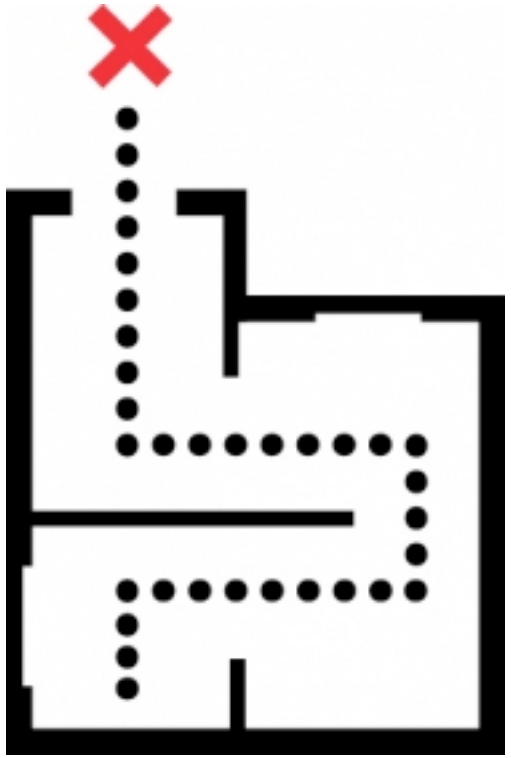
Android Internals Overview

Marko Gargenta
Marakana

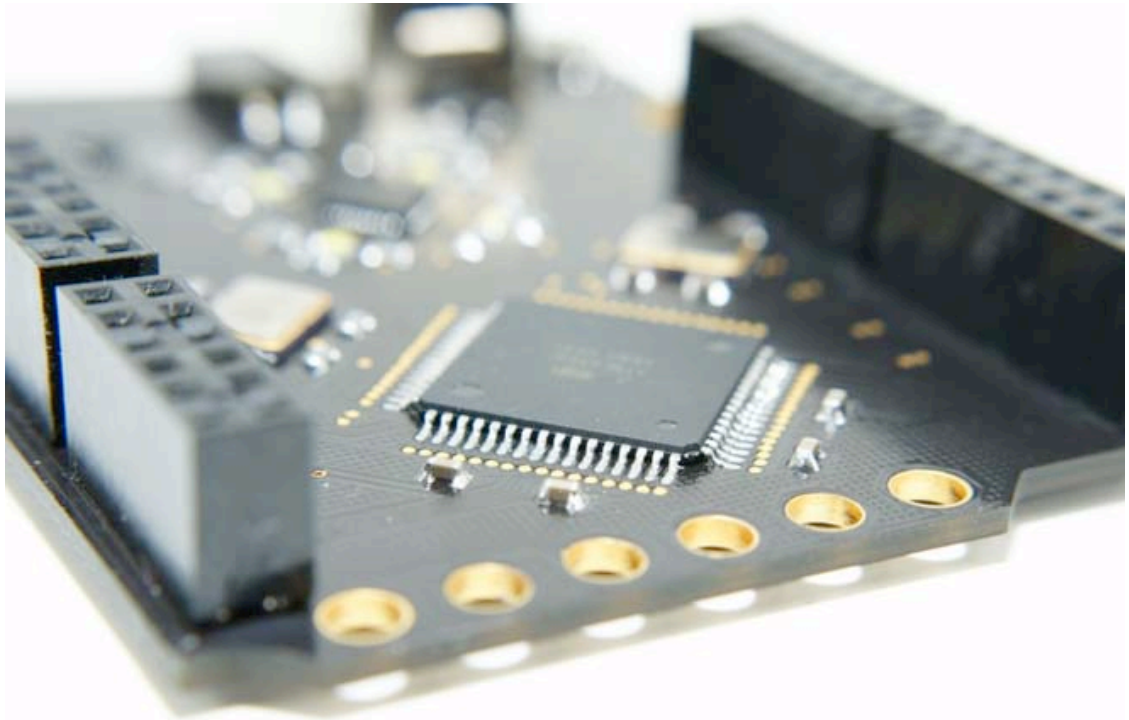
marakana

Agenda

- Working with Hardware
- Android Startup and Runtime
- Native Development Kit
- Summary



marakana



WORKING WITH HARDWARE

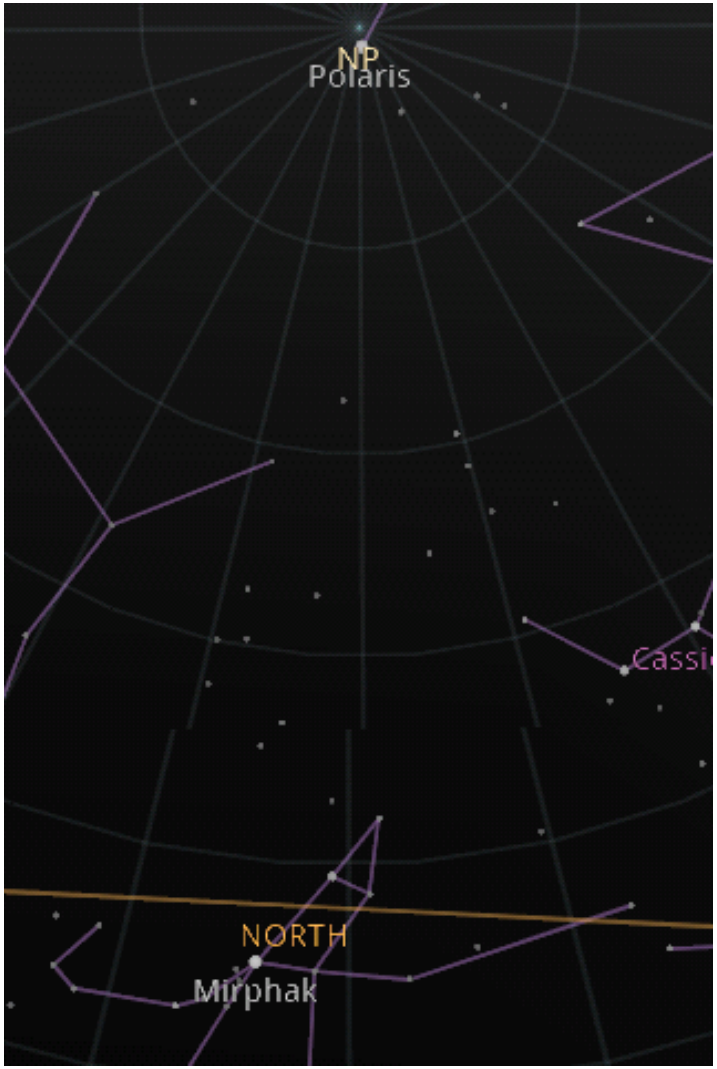
marakana

Sensors

Android supports many built-in sensors. You simply register with Sensor Manager to get notifications when sensor data changes.

Sensors are erratic and data comes in uneven intervals.

Emulator doesn't have good support for sensors.



marakana

Camera

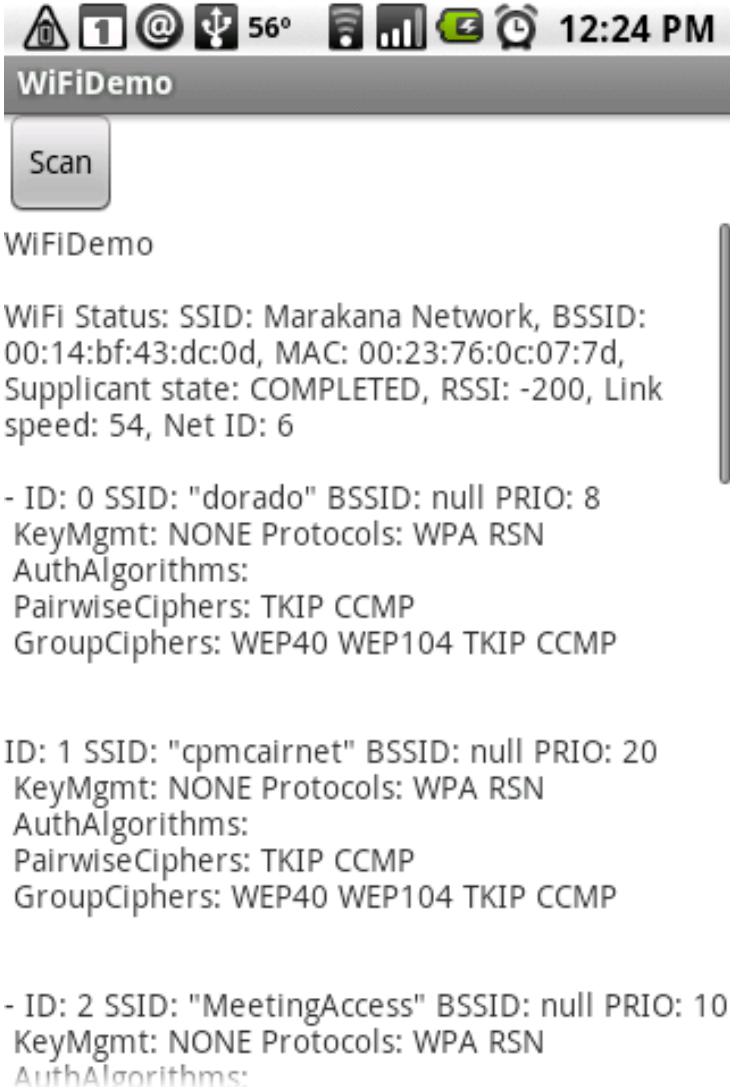
Android SDK supports access to built-in Camera and its preview.

You can access real-time frames, or get a callback when shutter is open. The photo data is passed back in either raw or jpeg format.

marakana

WiFi

WiFi API allows for managing your connection, scanning for active WiFi points and find out details about each.



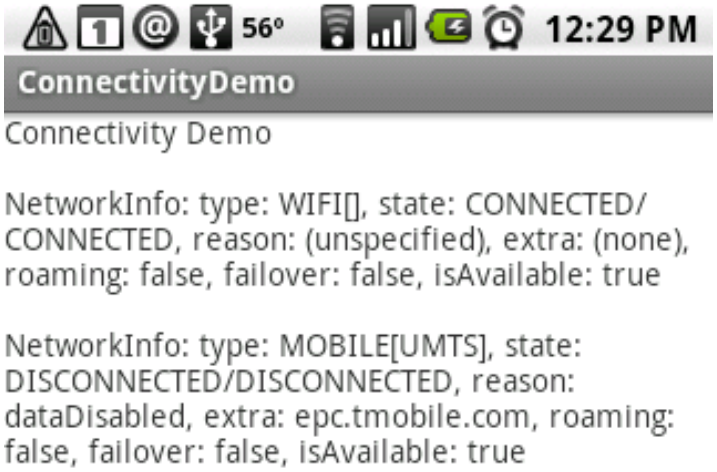
marakana

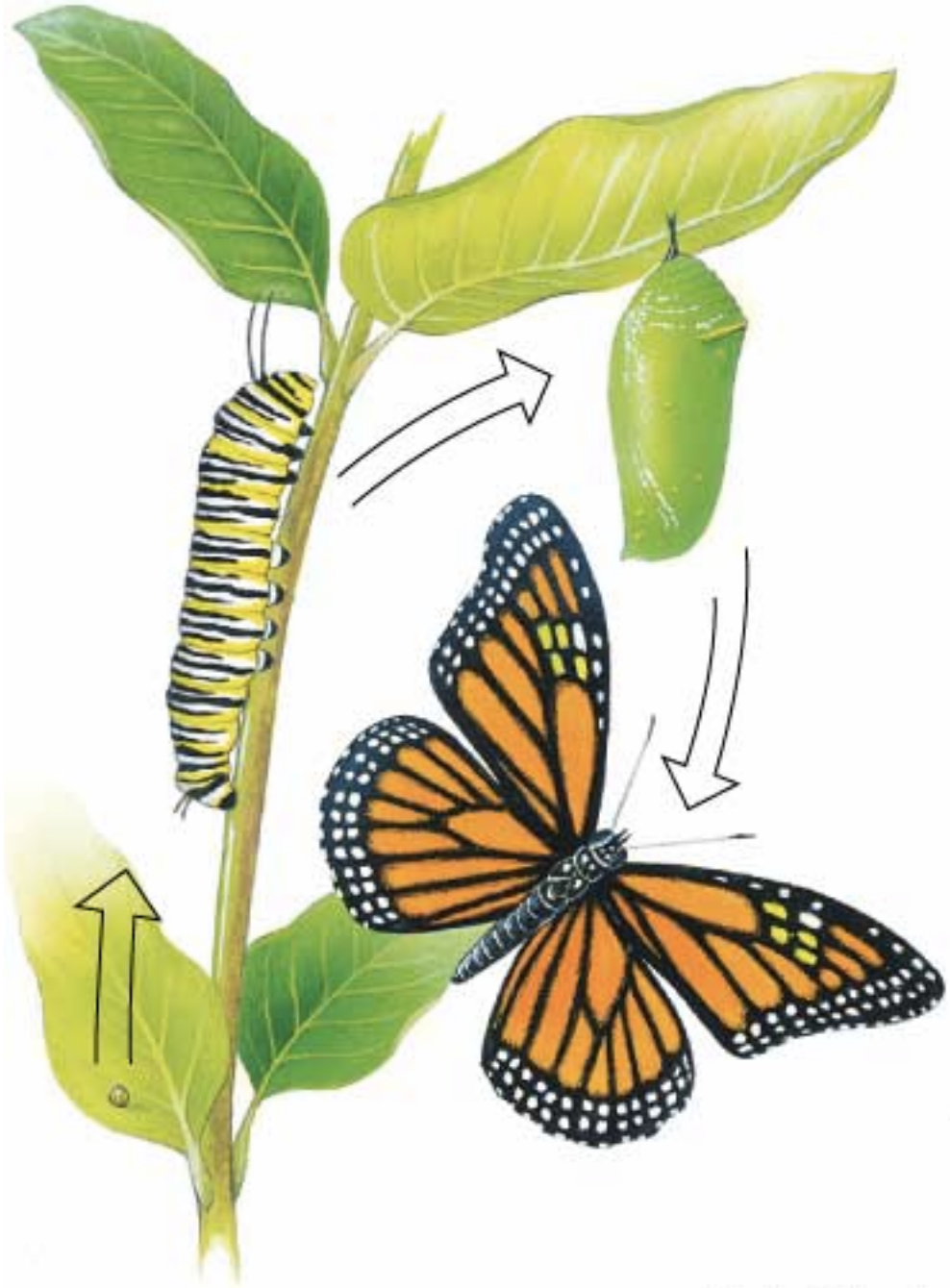
Telephony

With Telephony API, you can:

Make phone calls
Monitor phone state and activity
Access phone properties and status
Monitor data connectivity
Control the phone

It is a simple yet powerful API



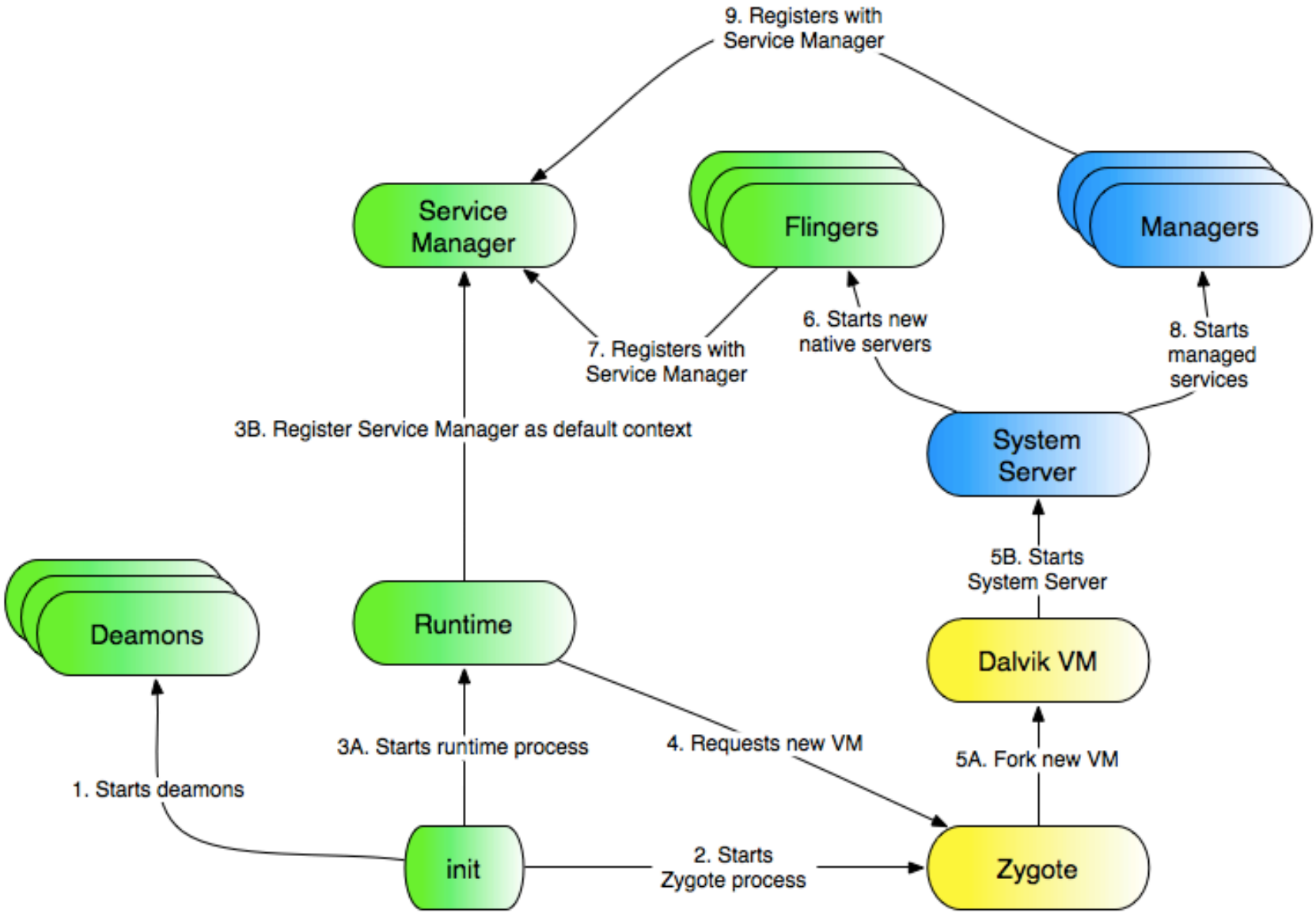


marakana

ANDROID STARTUP & RUNTIME

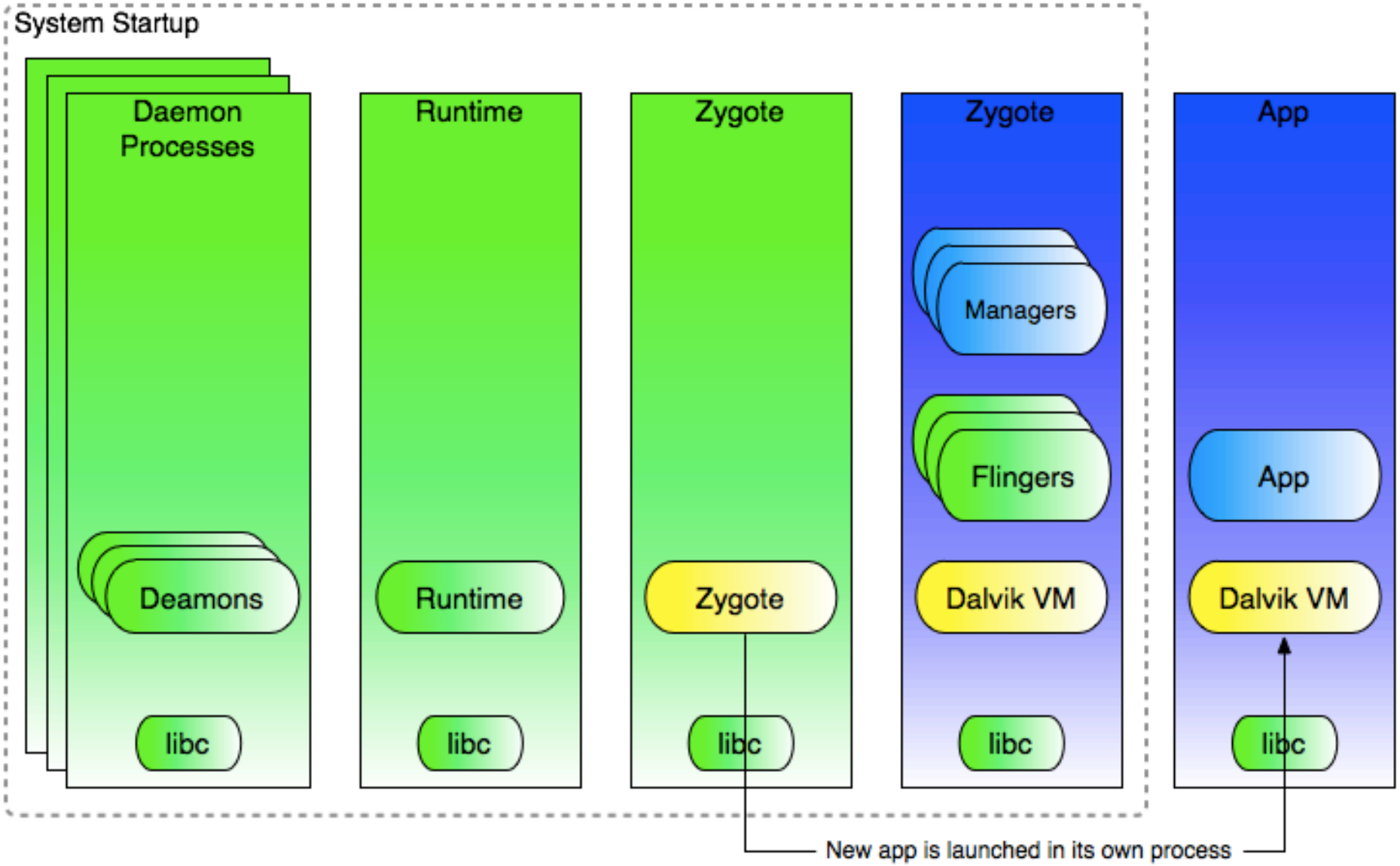
marakana

Startup Walkthrough



marakana

Runtime Overview



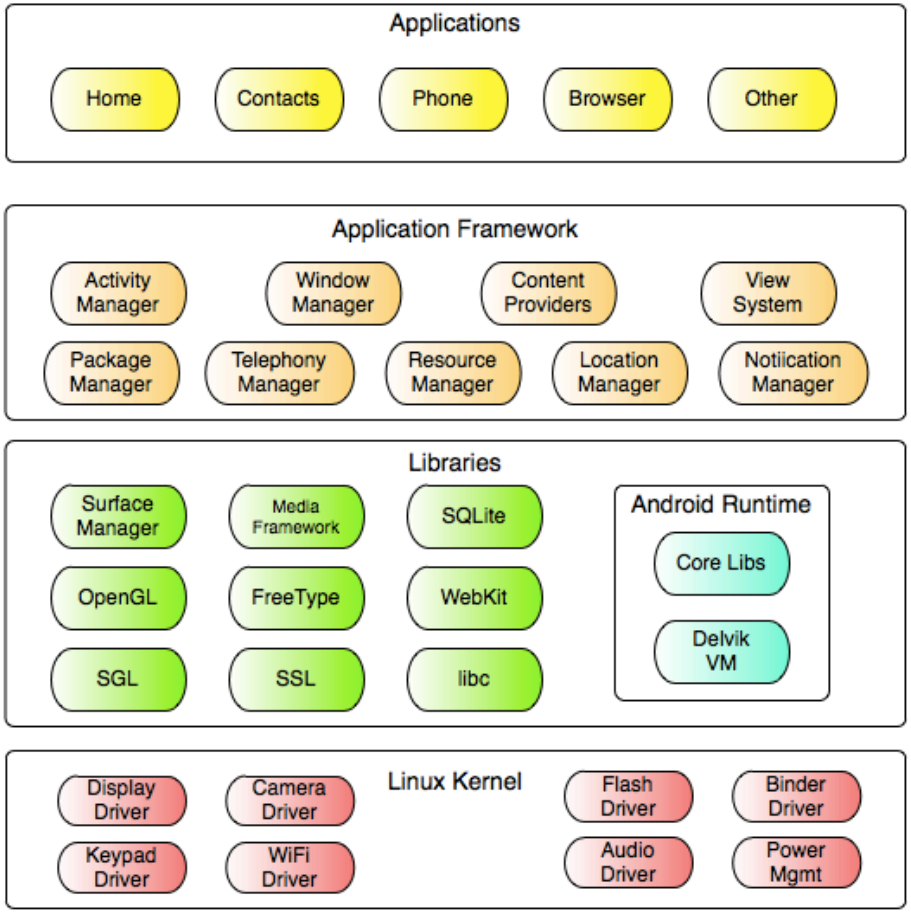
marakana

Layer Interactions

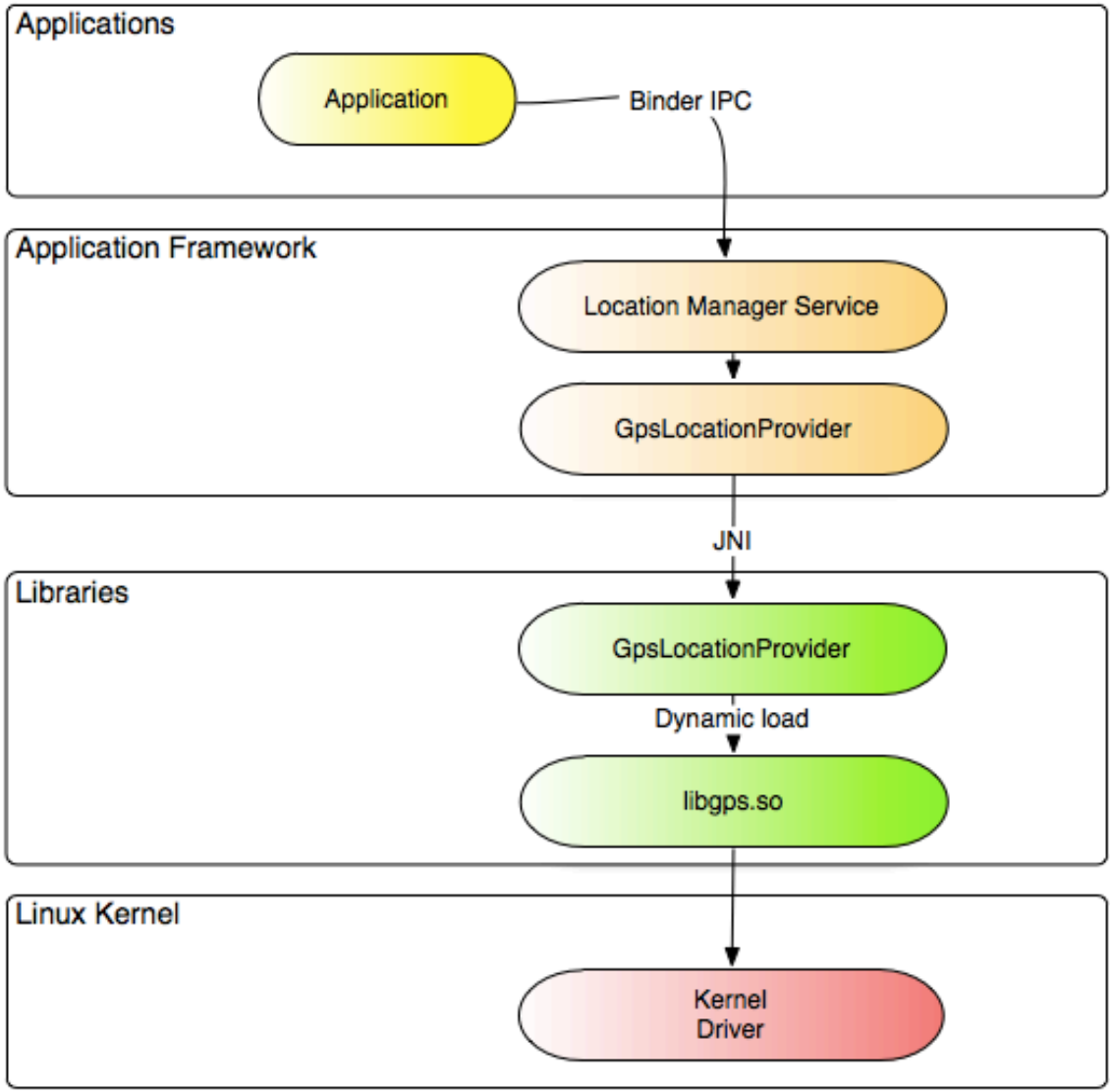
There are three main scenarios for your app to talk to native library:

- Directly
- Via native service
- Via native daemon

It will depend on the type of app and type of native library which method works best.

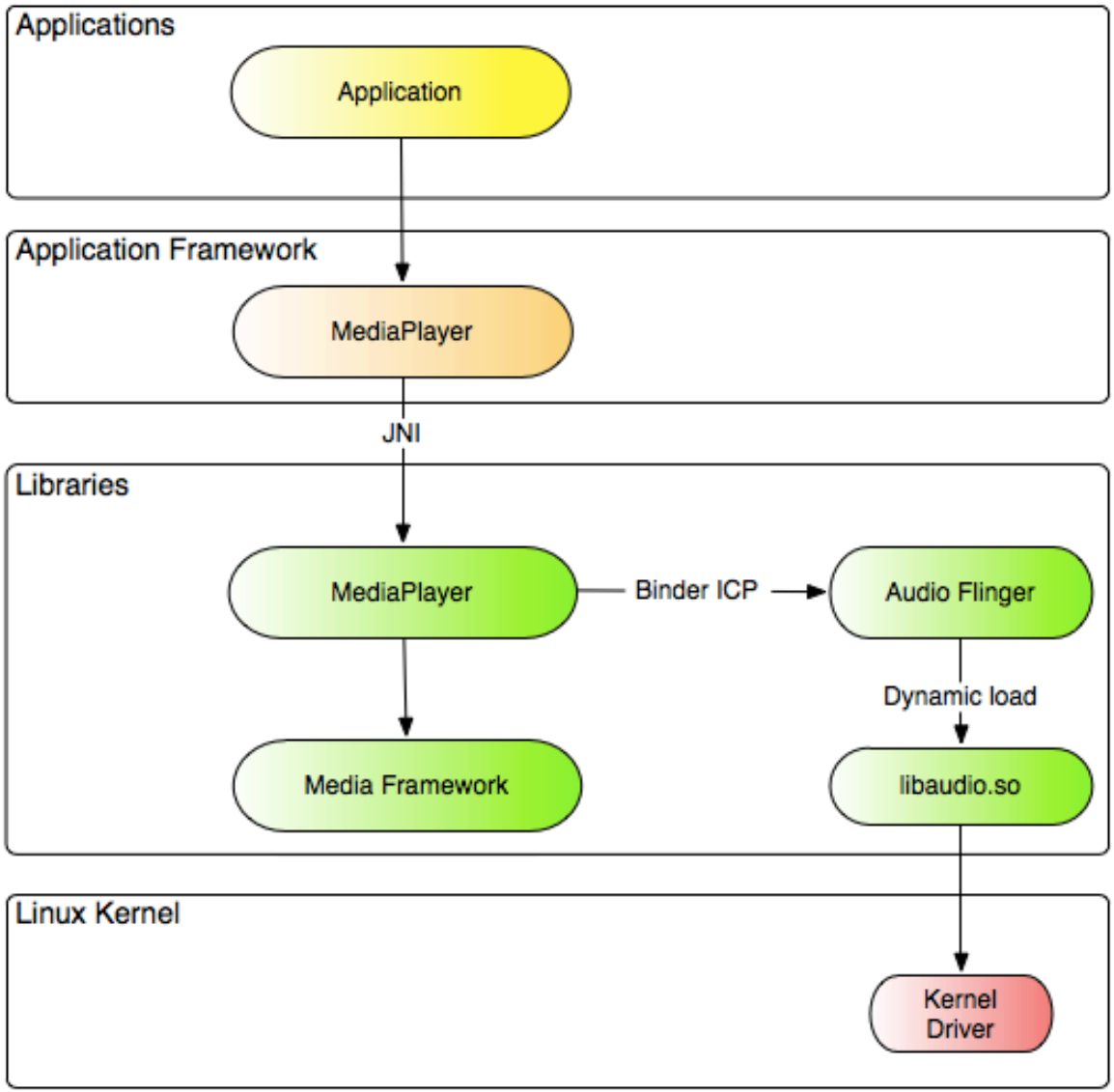


App – Runtime Service - Lib



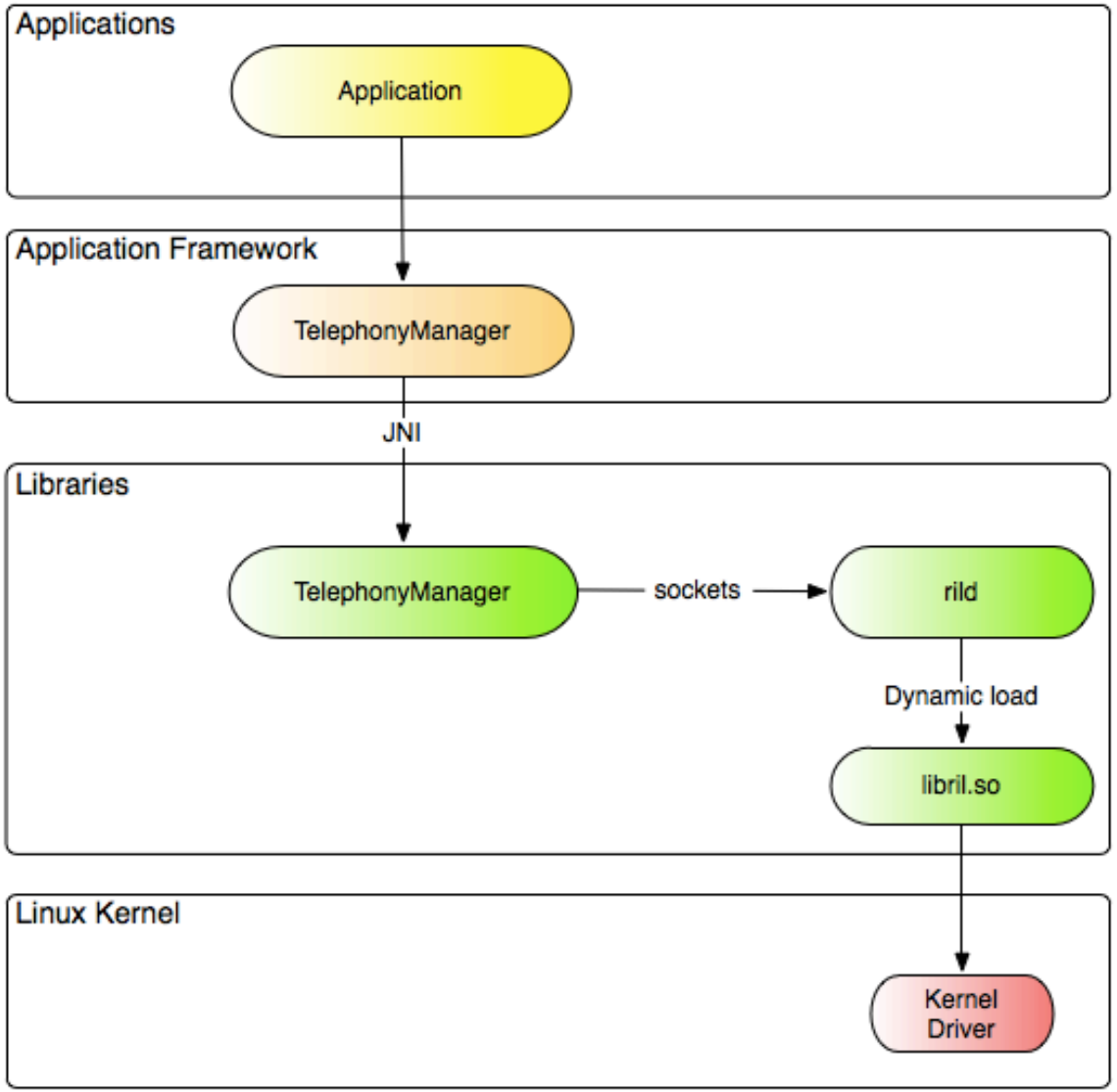
marakana

App – Runtime-Native Service-Lib



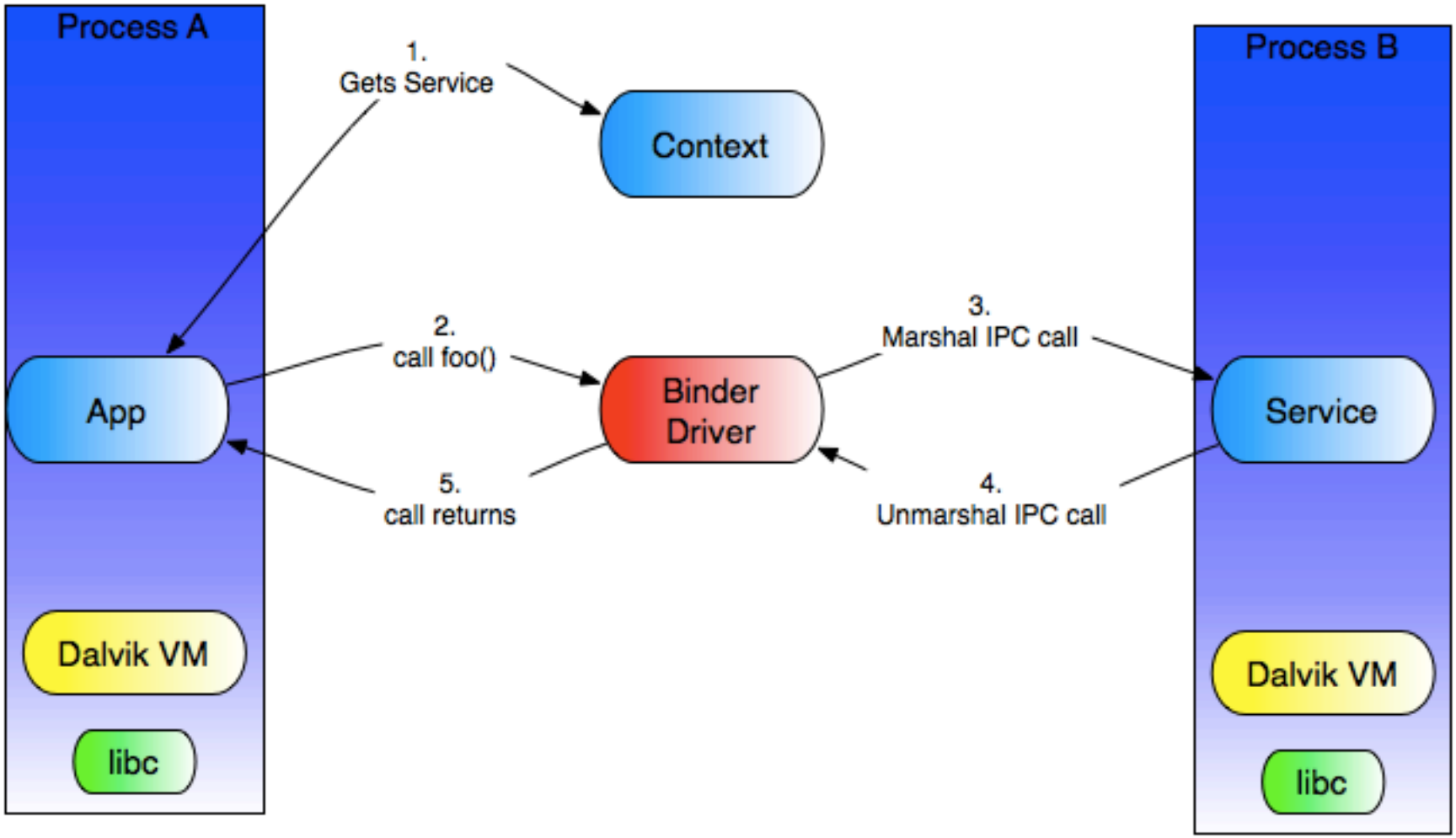
marakana

App–Runtime–Native Daemon-Lib



marakana

Binder IPC



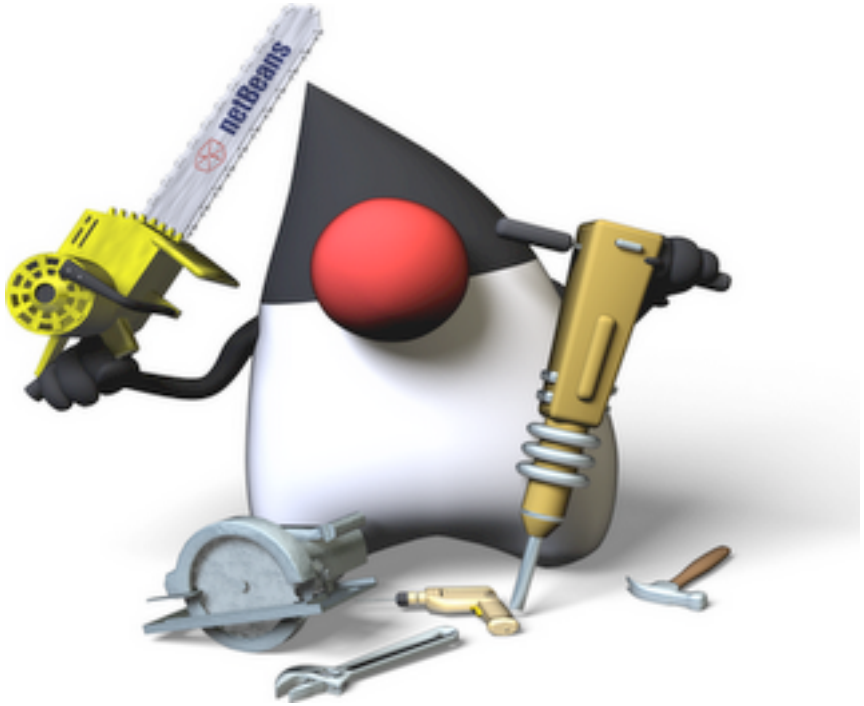
High-performance IPC: shared memory, per-process thread pool, synchronous

marakana

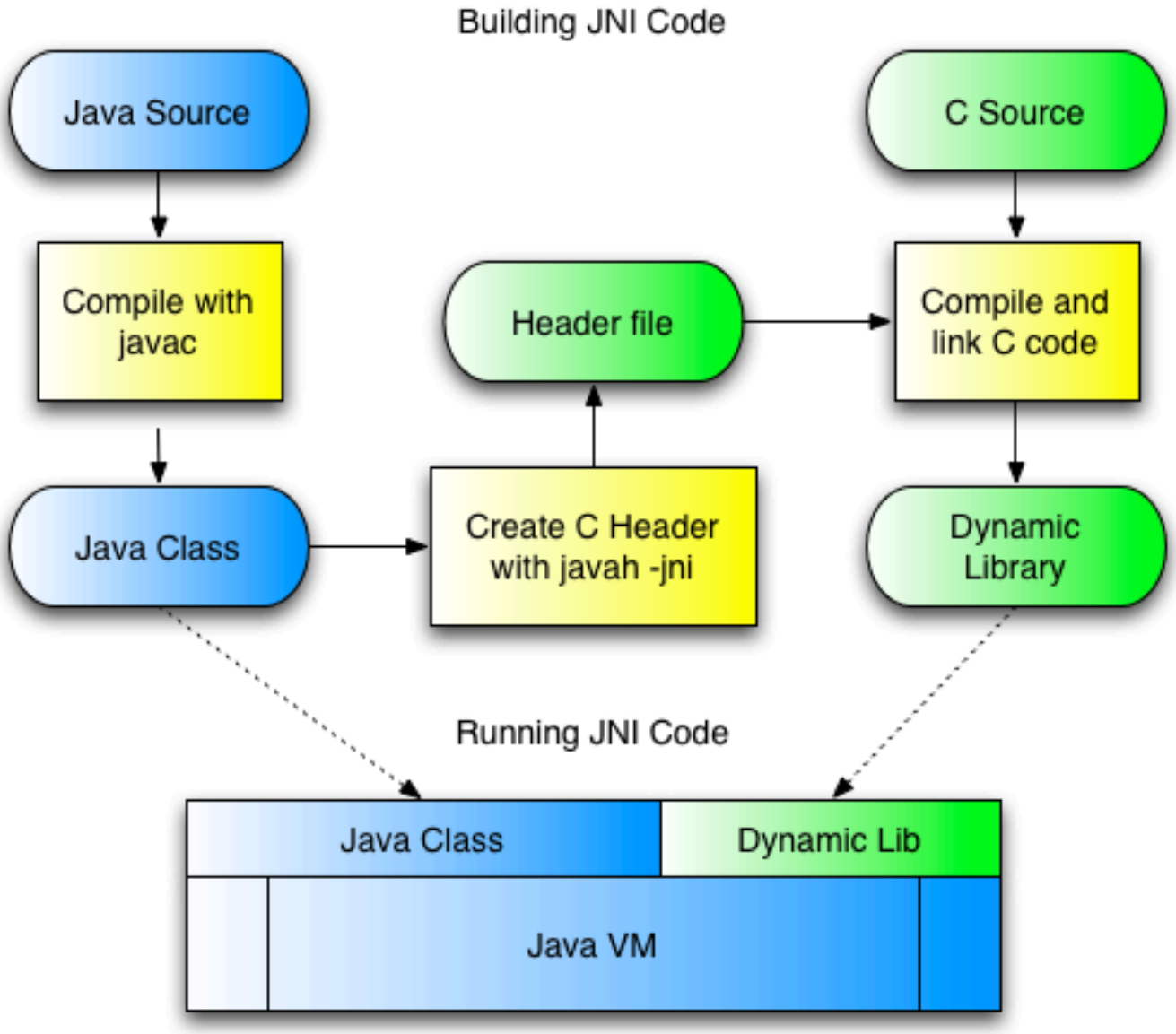
Java Native Interface

JNI defines naming and coding convention so that Java VM can find and call native code.

JNI is built into JVM to provide access to OS I/O and others.



Building and Running JNI Code



marakana

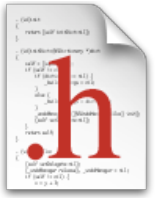
NATIVE DEVELOPMENT KIT

marakana

What's in NDK?

Tools to build and compile your native code for the device architecture (such as ARM)

A way to package your library into the APK file so you can distribute your application easily



A set of native system headers that will be supported for the future releases of Android platform (libc, libm, libz, liblog, JNI headers, some C++ headers, and OpenGL)

(some) documentation, sample code and examples

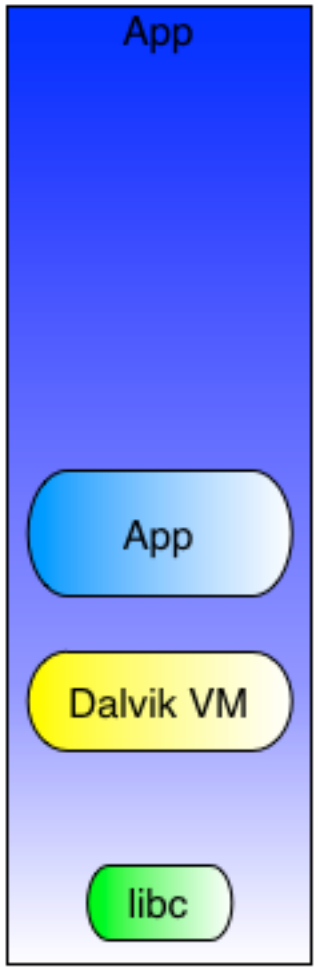
marakana

Why NDK?

NDK allows you to develop parts of your Android application in C/C++.

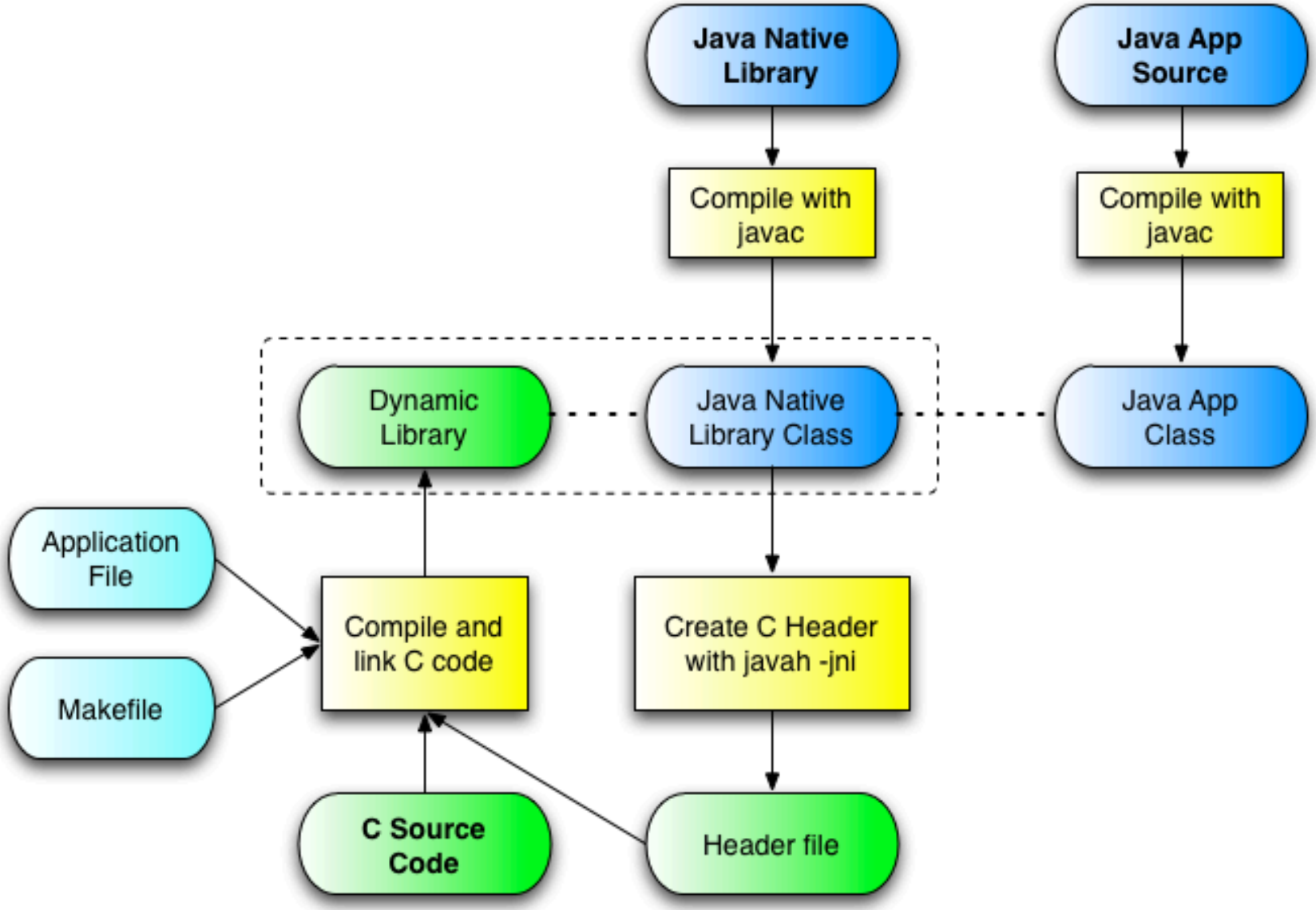
You cannot develop native-only apps in NDK – your app is still subject to security sandboxing.

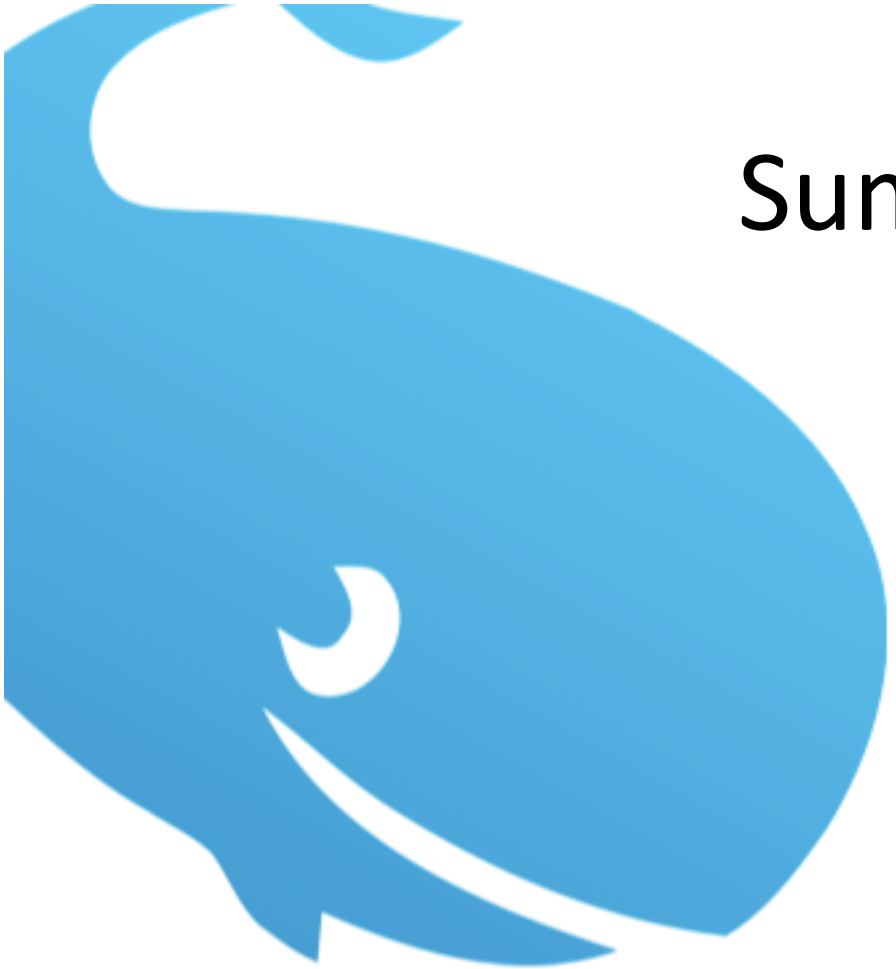
Main motivation for native code is performance.



marakana

Using NDK





marakana

Summary

Android SDK provides many APIs to allow you to get to the hardware.

In some cases, you may want to provide parts of your application in C. Main reason would be performance.

Android OS is based on Linux. You can modify the entire platform as well.

Marko Gargenta, Marakana.com
marko@marakana.com
+1-415-647-7000