Mobile & Embedded Computing Lab

An deep dive into

Android Open Source Project

Siji Sunny

Director, MELabs
siji@melabs.in
Mobile : +91 9820896398

History

- *October 2003* - Android, Inc founded in California by Andy Rubin, Rich Miner, Nick Sears and Chris White.
- The company's were to develop an advanced operating system for digital cameras
- *August 17, 2005* - Google acquired Android Inc.
- the team led by Rubin developed a mobile device platform powered by the Linux kernel.
- *October 22, 2008* - First Android device launched in the market.

Ref: Wikipedia

History

Cupcake

Donut

Eclair

Froyo

Gingerbread

Honeycomb

Ice Cream Sandwich

Jelly Bean

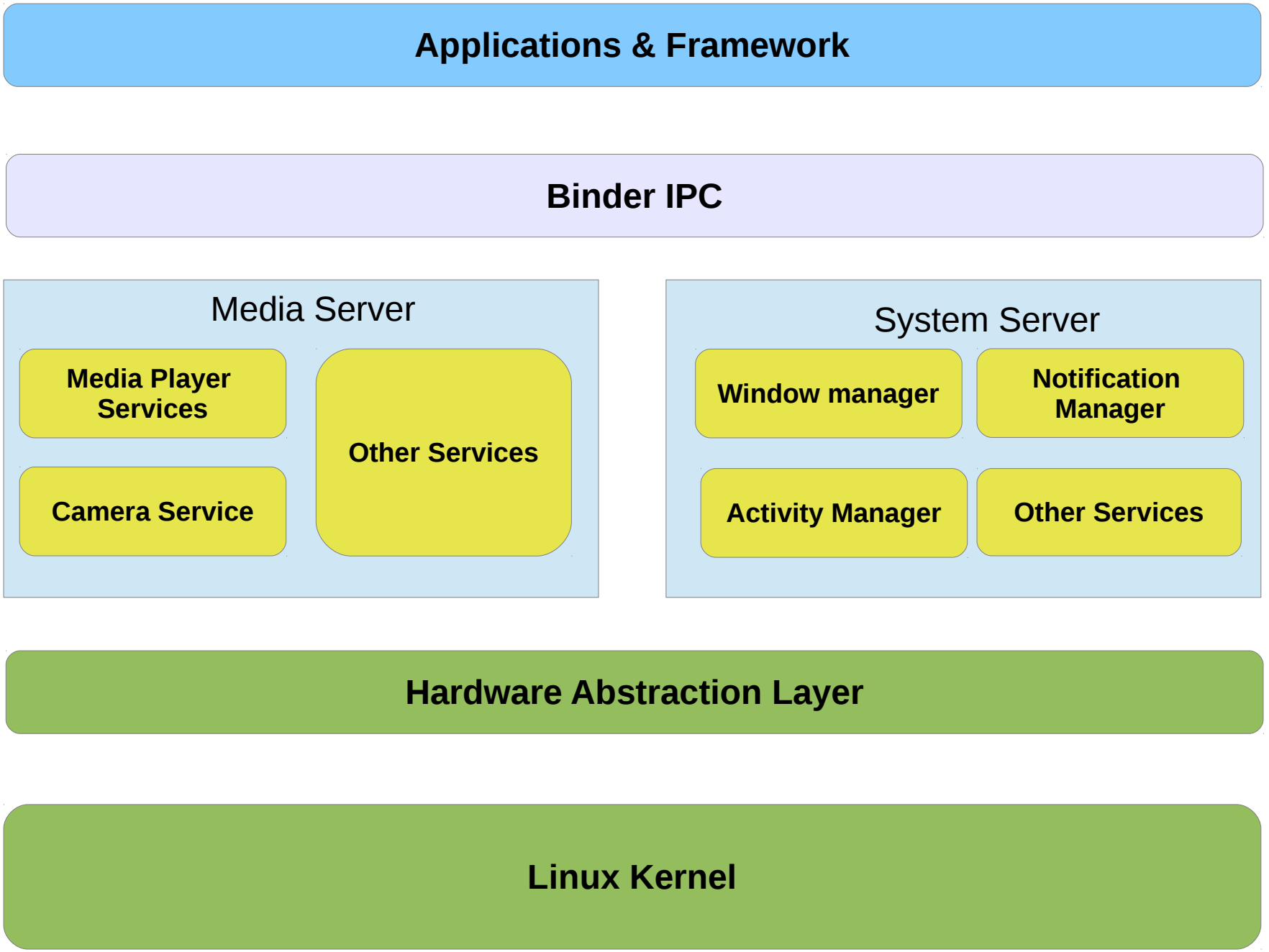
KitKat

Concept- AOSP

Android Open-Source Project

- Source code available for download-Contains
  - Kernel
  - Native Libraries
  - Android runtime
  - Application Framework
- License -Android Open Source Project is the Apache Software License, Version 2.0 (For userspace).
- Android originated by a group of companies - Open Handset Alliance, led by Google.
- **Freedom to implement your own device ?**
  - Google Apps are not open-source
  - Vendor depended firmware/drivers

Architecture



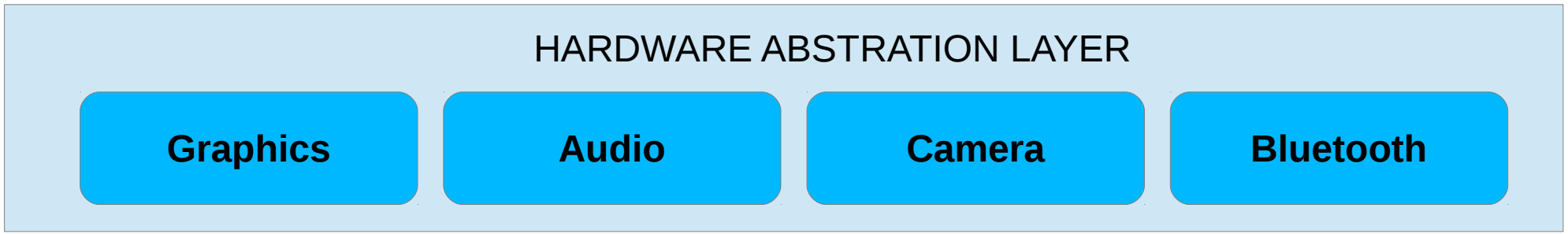
Ref :http://s.android.com/devices/index.html

Binder-IPC

- Driver to facilitate inter-process communication (IPC)
- Allows high level framework APIs to interact with Android's system services
- At the application framework level, all of this communication is hidden from the developer
- High performance through shared memory

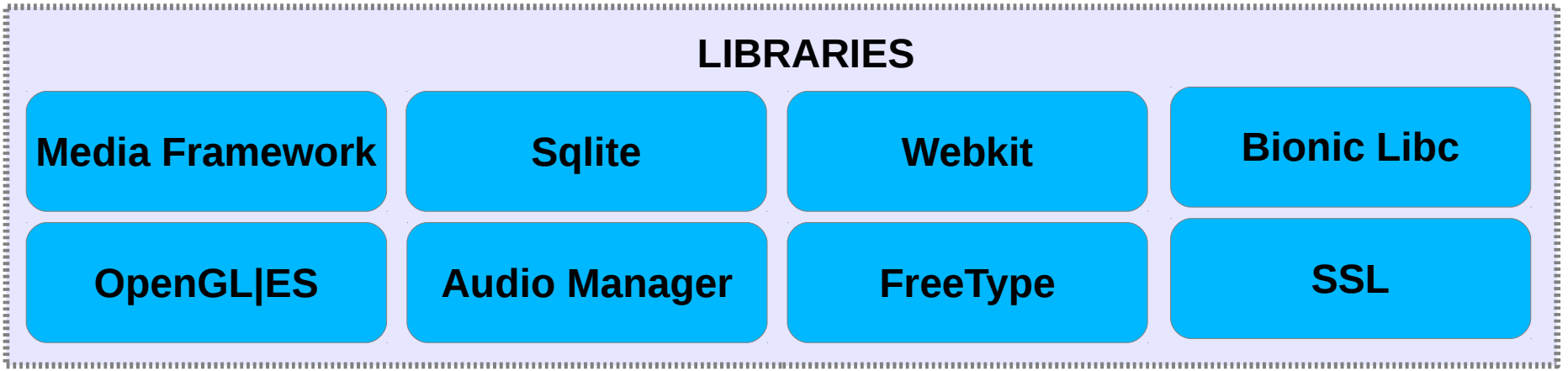
HAL

- User space C/C++ library layer
- Defines the interface that Android requires hardware “drivers” to implement
- Separates the Android platform logic from the hardware interface



Native Libraries

- Bionic Libc
- Function Libraries (For standard calls)
- Native Servers (for UI/Video/Audios)
- Hardware Abstraction Libraries



What is Bionic

- Custom libc implementation, optimized for embedded use.

Why build a custom libc library?

- License: **BSD License** (Objective was to keep GPL out of user-space)
- Size: will load in each process, so it needs to be small
- Fast: limited CPU power means we need to be fast
- Small size and fast code paths
- Very fast and small custom pthread implementation

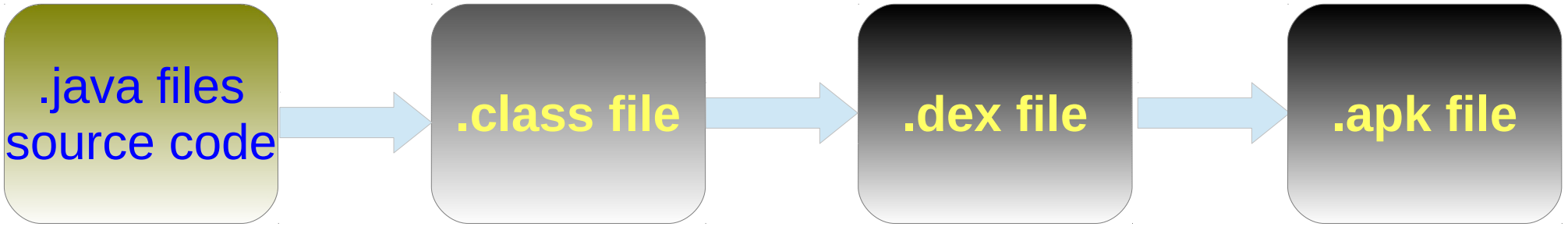
Bionic -Disadvantage

- Doesn't support certain POSIX features
- Not compatible with Gnu Libc (glibc)
- All native code must be compiled against bionic

Dalvik Vm

- The software that runs the apps on Android devices
- It's fast, even on weak CPUs
- it will run on systems with little memory
- it will run in an energy-efficient way
- Provides application portability and runtime consistency
- Runs optimized file format (.dex) and Dalvik bytecode
- Java .class / .jar files converted to .dex at build time

Compiling & packaging



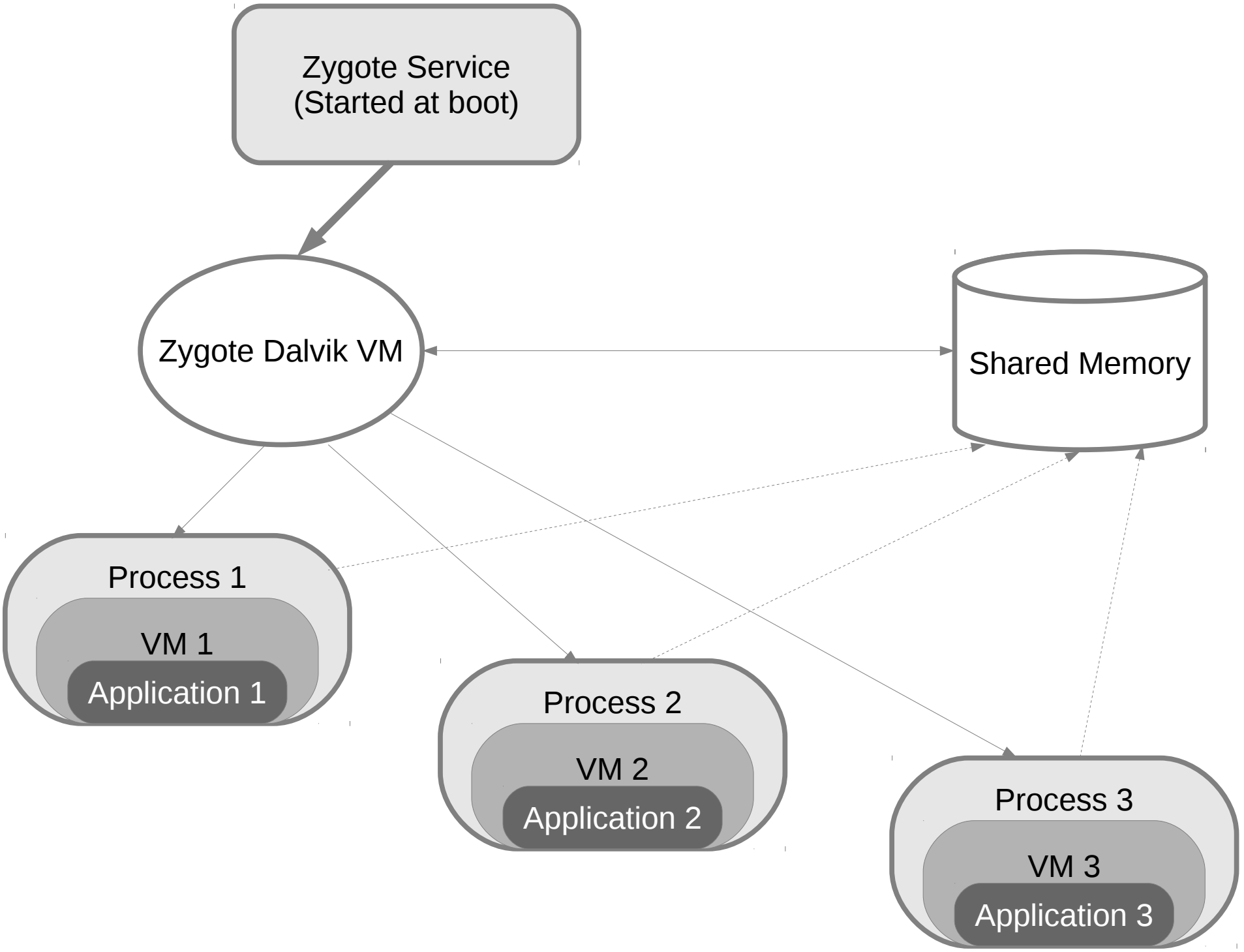
Java compiler
javac

Dex compiler
(dx)

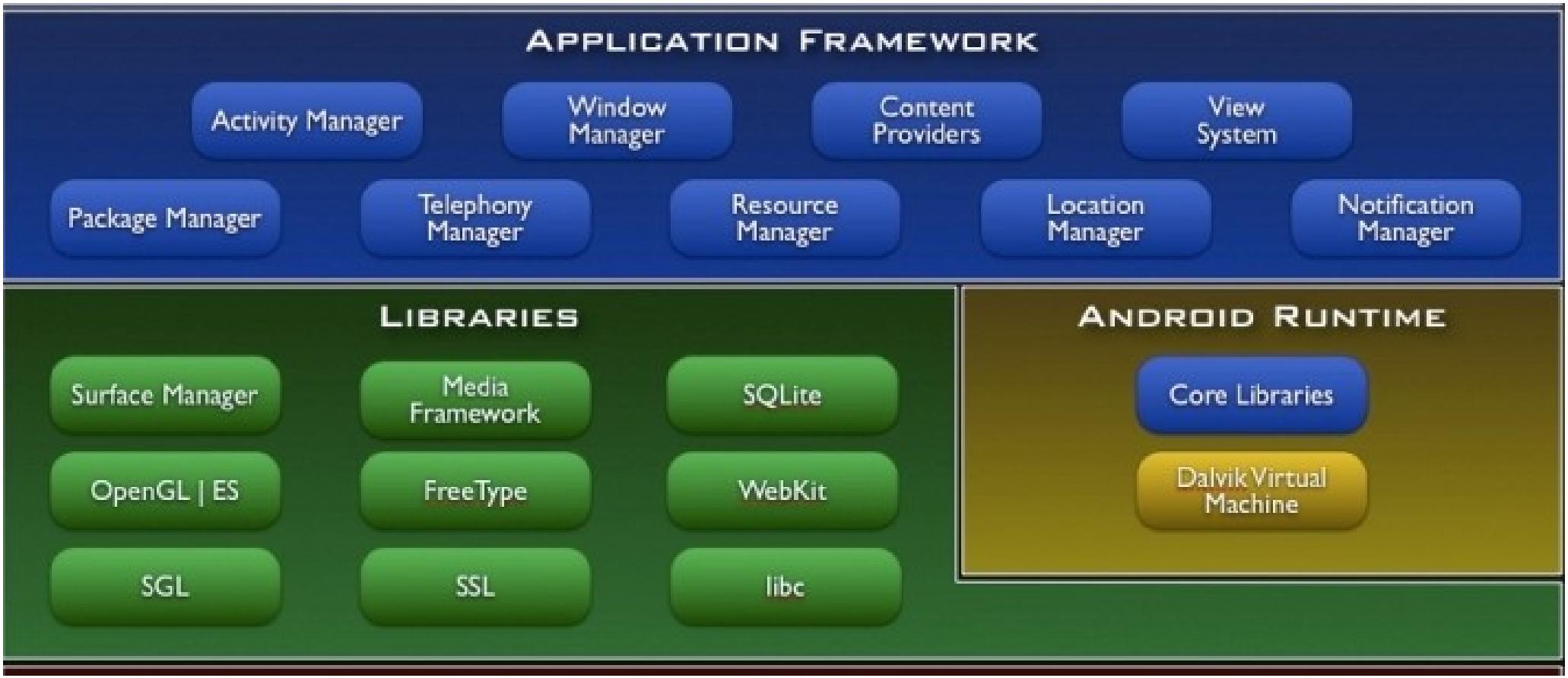
Packaging
(aapt)

Android Runtime

- Bootloader
- Kernel
- Init
- Android runtime Dalvik/Zygote
- System Server
- Activity Manager
- Launcher (Home)


Zygote Service
(Started at boot)
Zygote Dalvik VM
Shared Memory
Process 1
VM 1
Application 1
Process 2
VM 2
Application 2
Process 3
VM 3
Application 3

Application Framework



Core Platform Services

- Services that are essential to the Android platform
- Behind the scenes - applications typically don’t access them directly

Core Platform Services

- Activity Manager
- Package Manager
- Window Manager
- Resource Manager
- Content Providers
- View System

Hardware Services

- Telephony Service
- Location Service
- Bluetooth Service
- WiFi Service
- USB Service
- Sensor Service

Thank You

siji@melabs.in

Twitter : siji_sunny

sijisunny@gmail.com

Gtalk : sijisunny