OVERVIEW OF ANDROID CONT.

Lecture 6

COMPSCI 702

Security for Smart-Devices

Muhammad **Rizwan** Asghar

March 11, 2021

ANDROID FRAGMENTATION

- Vendors customise the OS for their devices
  - Typically, vendors include their apps
  - Some of apps could compromise security/privacy
  - E.g., the Samsung app compromises on privileges
  - Link: http://randomthoughts.greyhats.it/2013/03/owning-samsung-phones-for-fun-but-with.html

- However, a vendor does not push updates frequently
  - Some devices could be some versions behind
  - Some vendors stop supporting their devices afterwards
  - Link: http://theunderstatement.com/post/11982112928/android-orphans-visualizing-a-sad-history-of-support

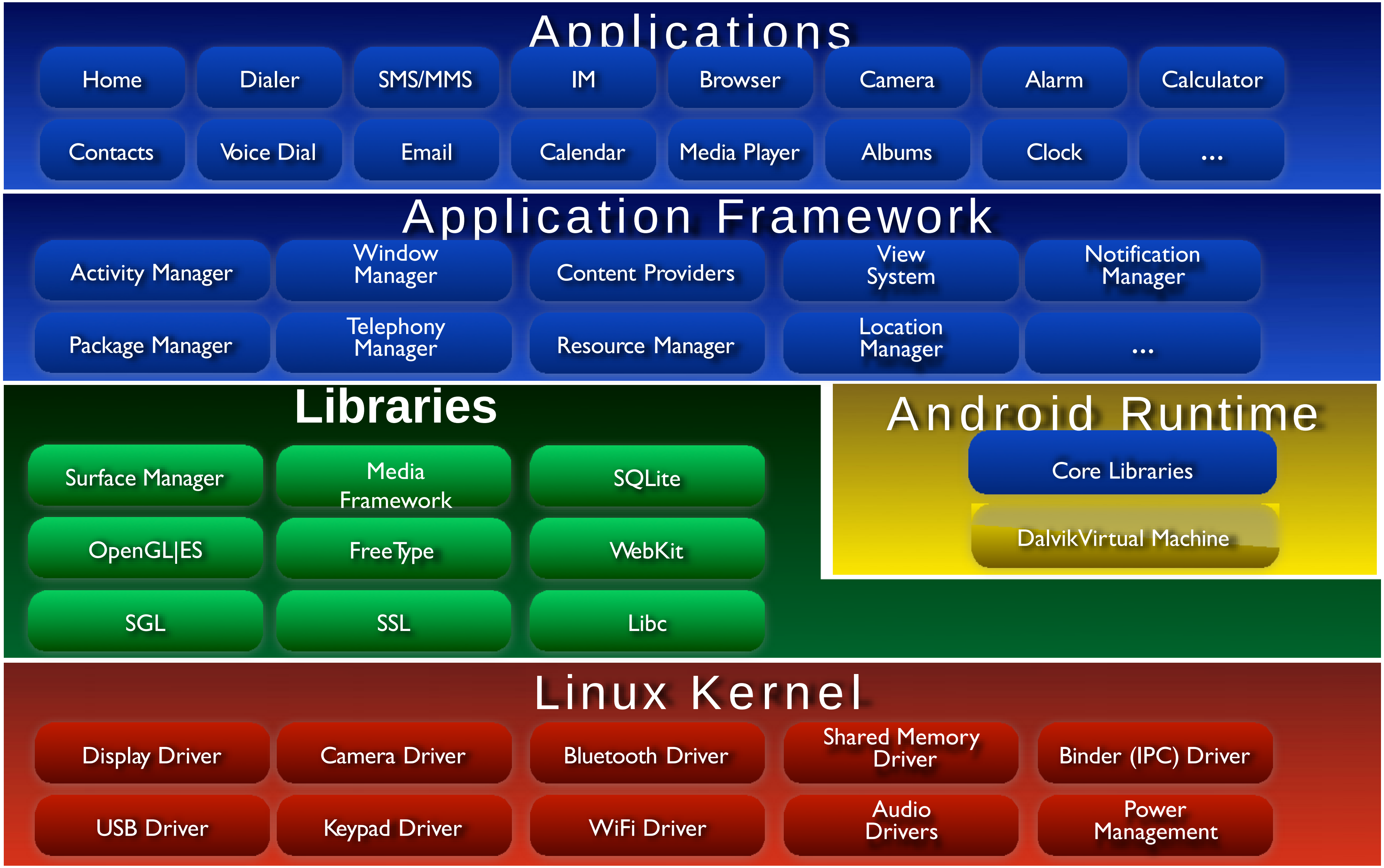
ANDROID FRAGMENTATION PROBLEM

- The lack of support can lead to vulnerabilities
- Often vendors just ignore vulnerabilities in their software
- Apple does a much better job
  - One single piece of hardware
  - One single software image

WHAT IS UNDER THE HOOD?

- Android is actually a middleware
- It sits between a Linux kernel and a set of APIs
- Android apps are mainly written in Java
  - Only Android apps can run on Android
- Through Android APIs, apps can access all the device components
  - It provides apps a rich set of information

ANDROID ANATOMY



Applications

- Home
- Dialer
- SMS/MMS
- IM
- Browser
- Camera
- Alarm
- Calculator
- Contacts
- Voice Dial
- Email
- Calendar
- Media Player
- Albums
- Clock
- ...

Application Framework

- Activity Manager
- Window Manager
- Content Providers
- View System
- Notification Manager
- Package Manager
- Telephony Manager
- Resource Manager
- Location Manager
- ...

Libraries

- Surface Manager
- Media Framework
- SQLite
- OpenGL|ES
- FreeType
- WebKit
- SGL
- SSL
- Libc

Android Runtime

- Core Libraries
- DalvikVirtual Machine

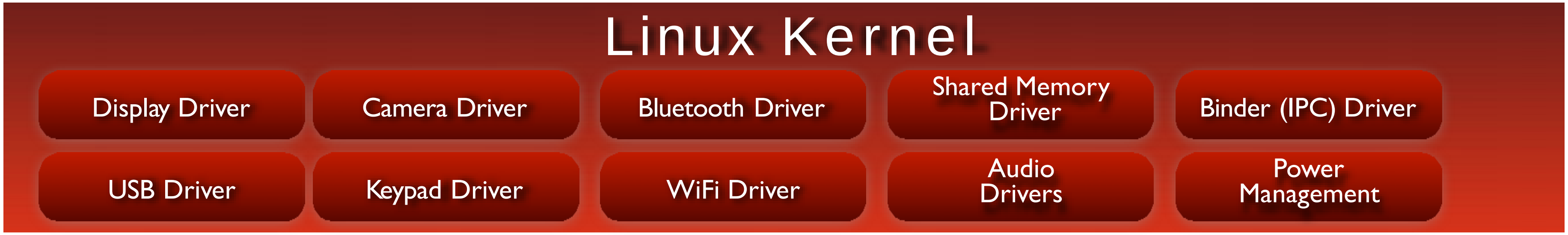
Linux Kernel

- Display Driver
- Camera Driver
- Bluetooth Driver
- Shared Memory Driver
- Binder (IPC) Driver
- USB Driver
- Keypad Driver
- WiFi Driver
- Audio Drivers
- Power Management

Source: Android Anatomy and Physiology, Patrick Brady

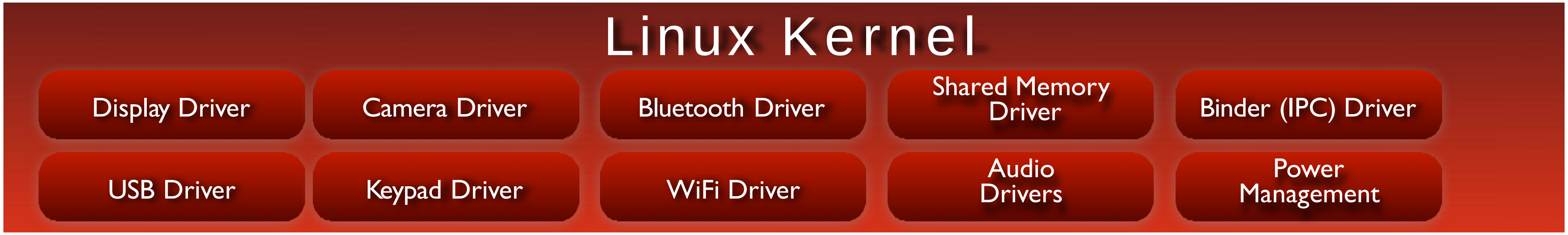
LINUX KERNEL

- Android is built on the Linux kernel
  - But it is not Linux
- No glibc support
- Does not include the full set of standard Linux utilities
- Kernel enhancements



WHY LINUX KERNEL?

- Great memory and process management
- Permissions-based security model
- Proven driver model
- Support for shared libraries
- It is already open-source!



Linux Kernel

Display Driver

Camera Driver

Bluetooth Driver

Shared Memory Driver

Binder (IPC) Driver

USB Driver

Keypad Driver

WiFi Driver

Audio Drivers

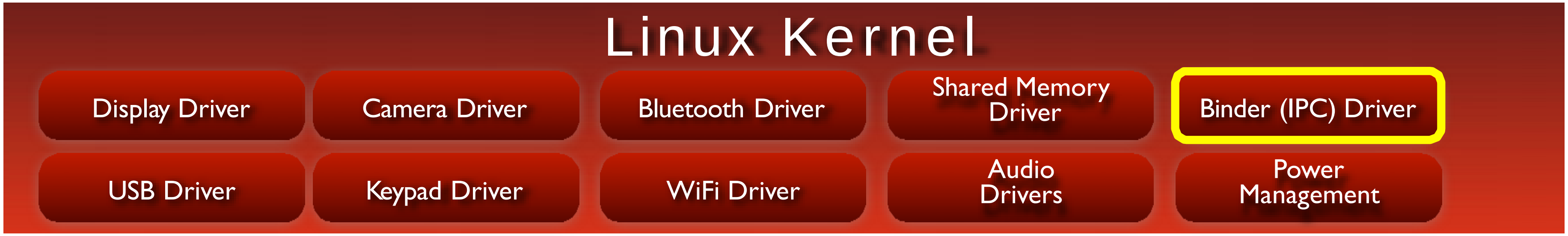
Power Management

BINDER

- Applications and Services may run in separate processes but must communicate and share data

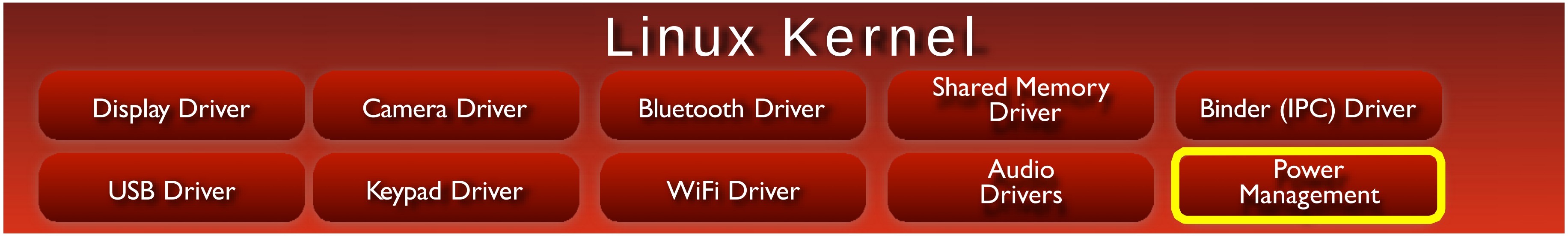
- **Issue:** Inter-Process Communication (IPC) can introduce significant processing overhead and security holes

- **Solution:** Driver to facilitate IPC



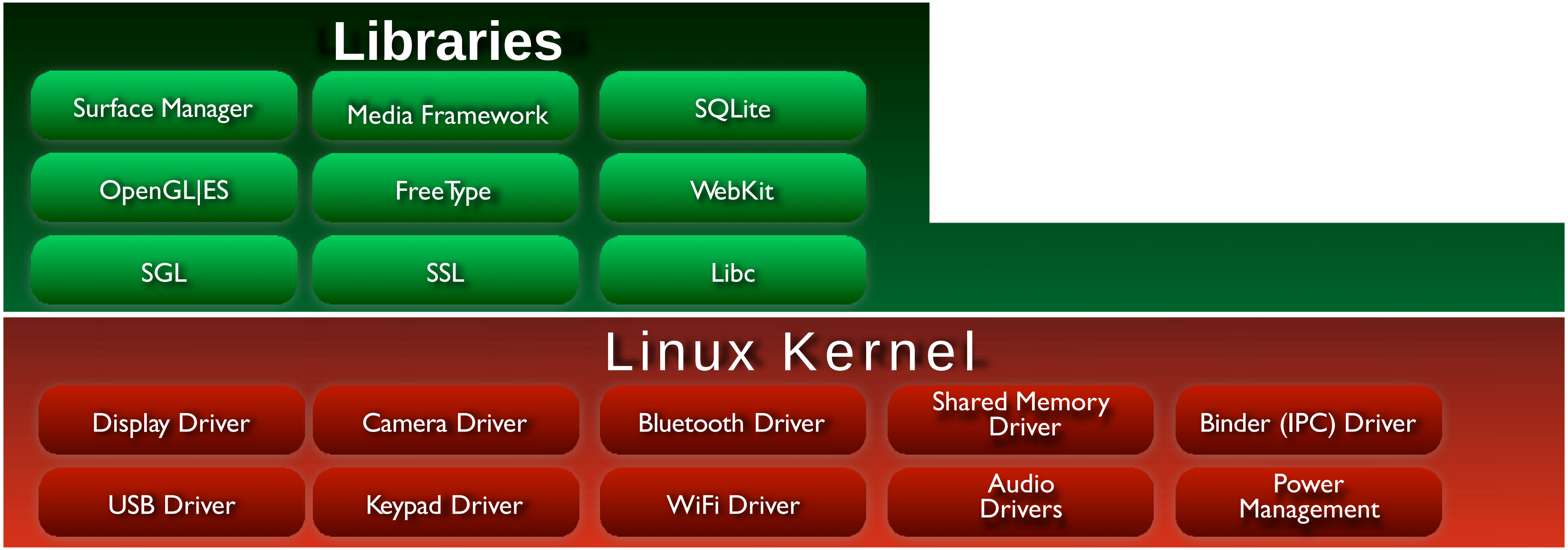
POWER MANAGEMENT

- Mobile devices run on battery power
- Batteries have limited capacity
- Built on top of standard Linux Power Management
- More aggressive power management policy



NATIVE LIBRARIES

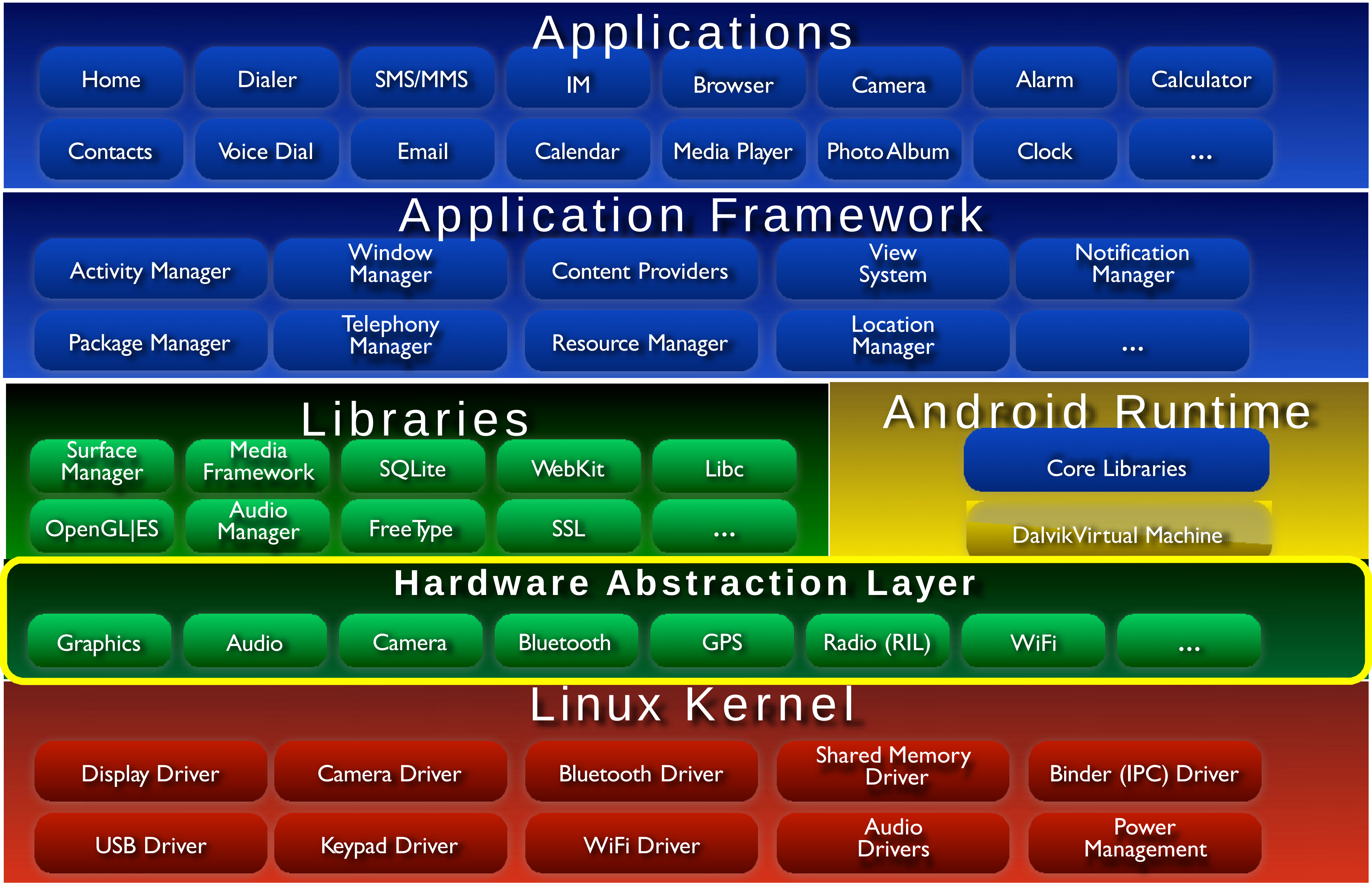
- Bionic libc is a custom libc implementation
- Why not glibc?



BIONIC

- Why bionic?
  - **License:** glibc is LGPL, which prevents static linking of proprietary software
  - **Size:** Will load in each process, so it needs to be small
  - **Efficiency:** Limited CPU power requires efficient solutions
- Bionic libc
  - **License:** BSD license
  - **Size:** Small size and fast code paths
  - **Efficiency:** Very fast and small custom pthread implementation
  - Does not support certain POSIX features
  - Not compatible with glibc
  - All native code must be compiled against bionic

HARDWARE ABSTRACT LAYER (HAL)



Applications

Home, Dialer, SMS/MMS, IM, Browser, Camera, Alarm, Calculator

Contacts, Voice Dial, Email, Calendar, Media Player, Photo Album, Clock, ...

Application Framework

Activity Manager, Window Manager, Content Providers, View System, Notification Manager

Package Manager, Telephony Manager, Resource Manager, Location Manager, ...

Libraries

Surface Manager, Media Framework, SQLite, WebKit, Libc

OpenGL|ES, Audio Manager, FreeType, SSL, ...

Android Runtime

Core Libraries

Dalvik Virtual Machine

Hardware Abstraction Layer

Graphics, Audio, Camera, Bluetooth, GPS, Radio (RIL), WiFi, ...

Linux Kernel

Display Driver, Camera Driver, Bluetooth Driver, Shared Memory Driver, Binder (IPC) Driver

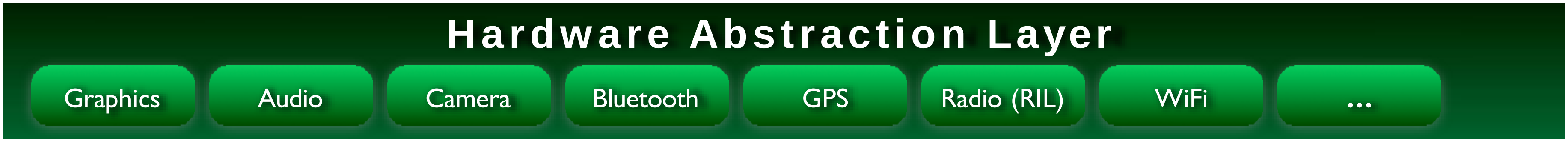
USB Driver, Keypad Driver, WiFi Driver, Audio Drivers, Power Management

HARDWARE ABSTRACTION LIBRARIES

- User space C/C++ library layer

- Separates the Android platform logic from the hardware interface

- Why do we need a user space HAL?
  - Not all components have standardised kernel driver interfaces
  - Kernel drivers are GPL, which exposes any proprietary IP
  - Android has specific requirements for hardware drivers

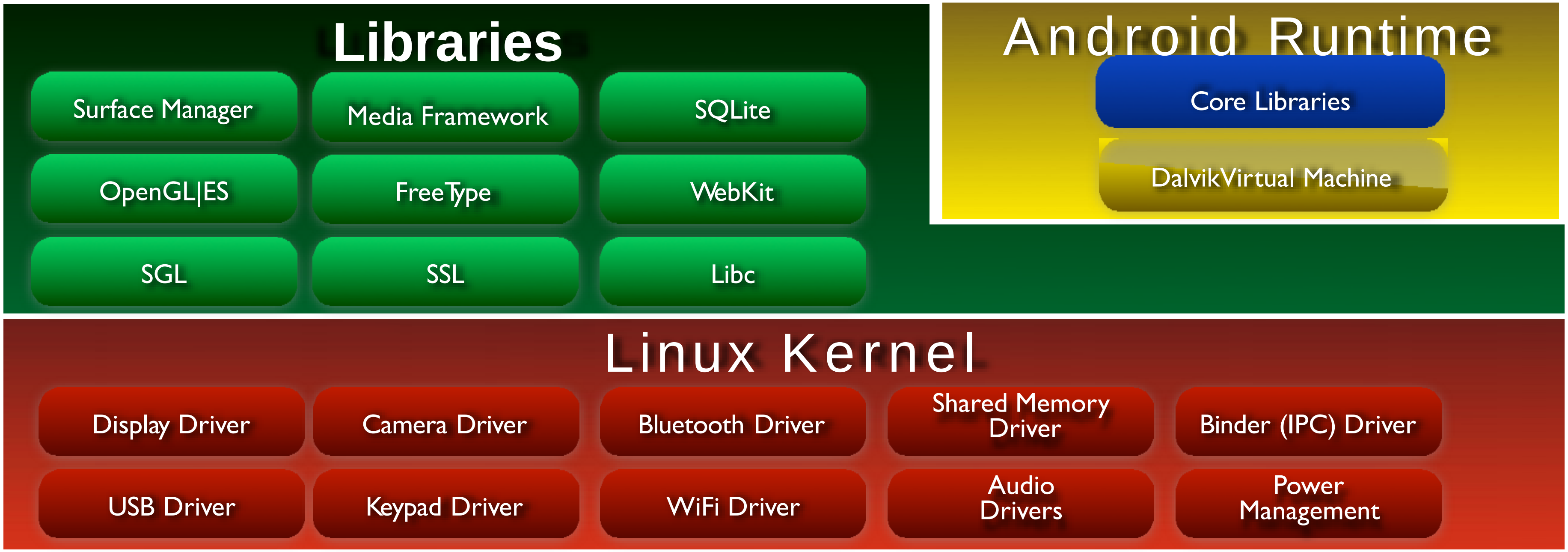


Hardware Abstraction Layer

Graphics | Audio | Camera | Bluetooth | GPS | Radio (RIL) | WiFi | ...

ANDROID RUNTIME

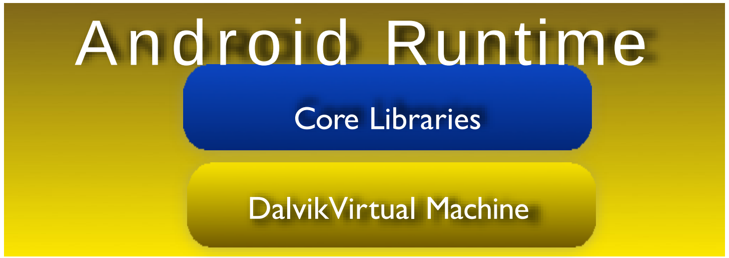
- Dalvik Virtual Machine
- Core libraries



DALVIK VIRTUAL MACHINE

- Android's custom clean room implementation
  - Provides application portability and runtime consistency
  - Runs Dalvik bytecode - optimised file format (.dex)
  - Java .class / .jar files converted to .dex at build time

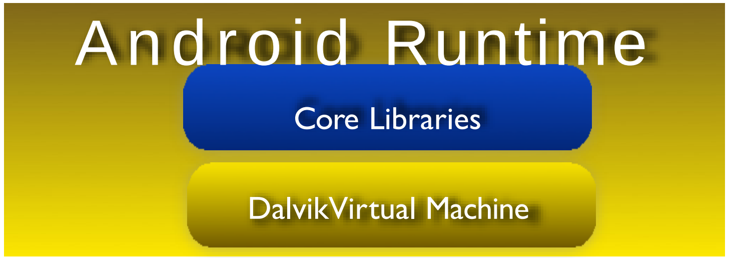
- Designed for embedded environment
  - Supports multiple virtual machine processes per device
  - Highly CPU-optimised bytecode interpreter
  - Uses runtime memory very efficiently



CORE LIBRARIES

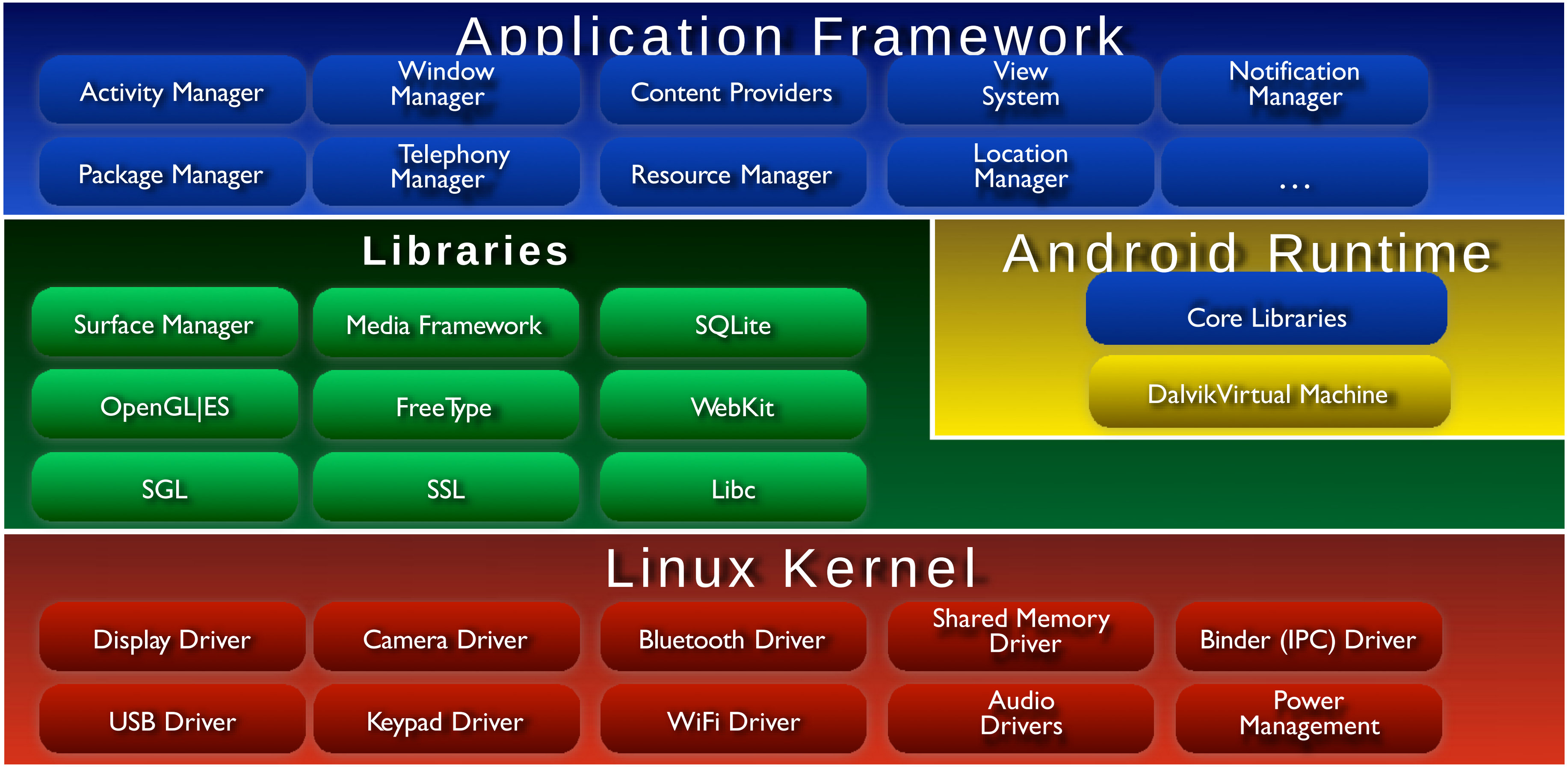
- Core APIs for Java language provide a powerful, yet simple and familiar development platform

- They do not actually perform much of the actual work and are, in fact, essentially Java “wrappers” around a set of C/C++ based libraries

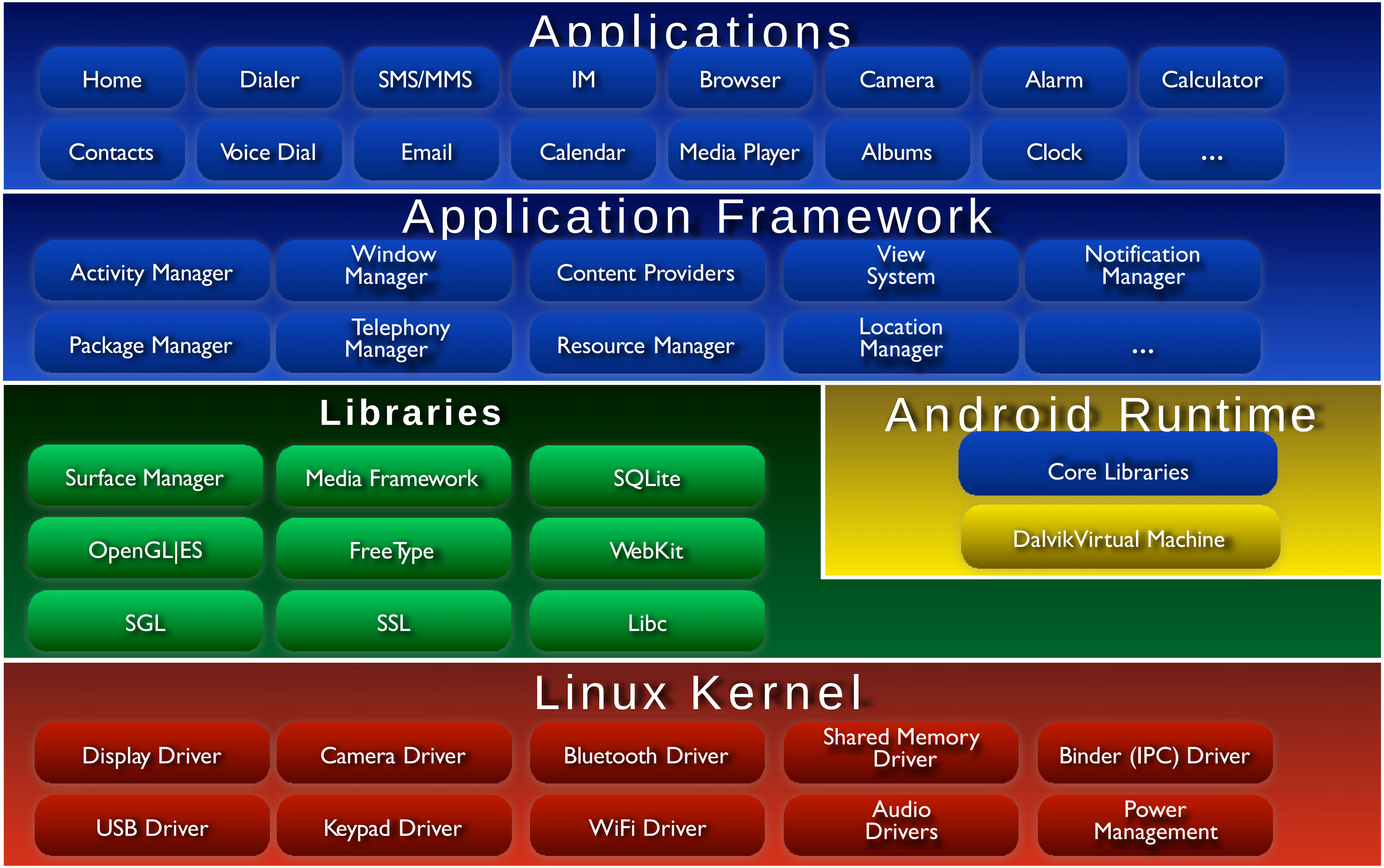


APPLICATION FRAMEWORK

- Services that are essential to the Android platform
- Apps typically do not access them directly

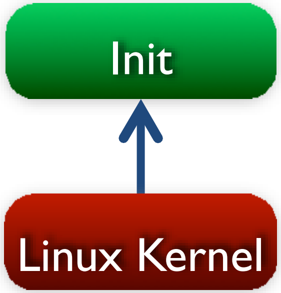


APPLICATIONS



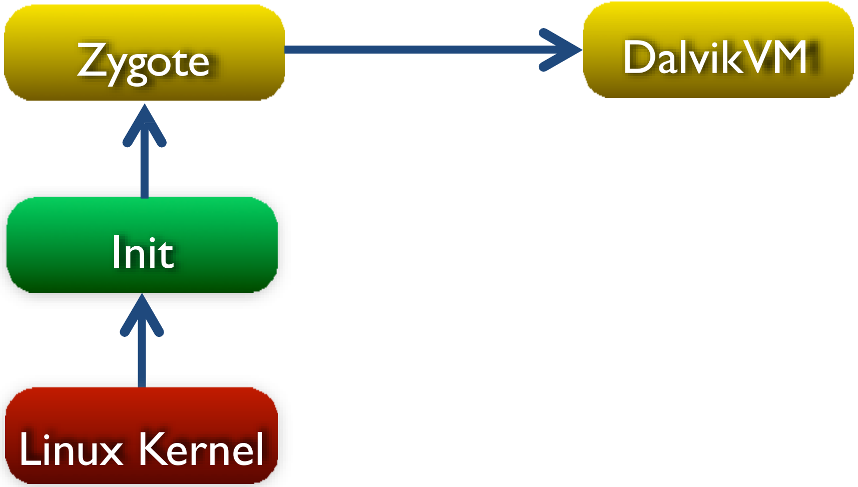
HIGH LEVEL VIEW: INIT

- Similar to most Linux-based systems, at startup, the bootloader loads the Linux kernel and starts the init process



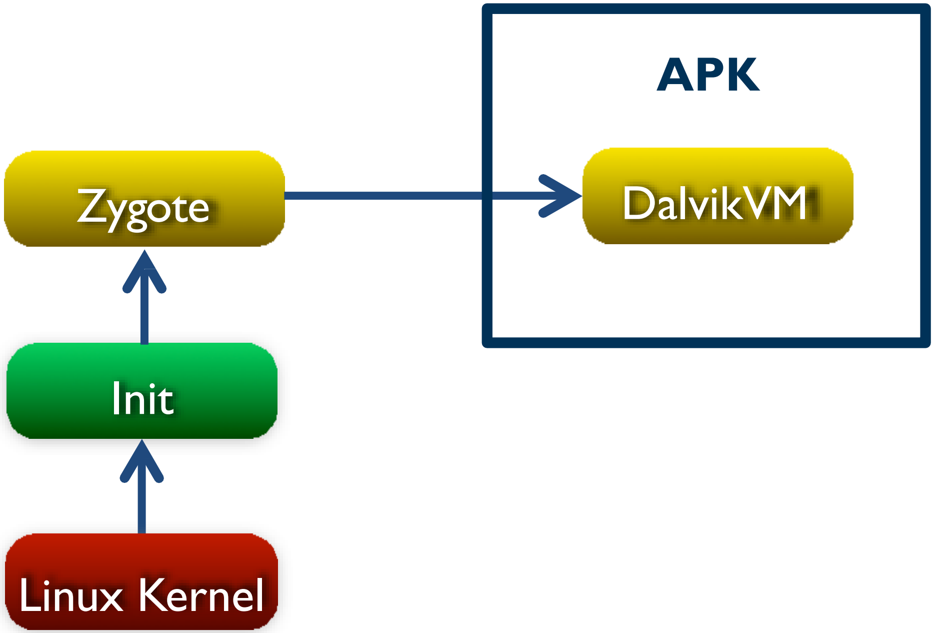
HIGH LEVEL VIEW: ZYGOTE

- Init process starts the zygote process
  - A nascent process that initialises a Dalvik VM instance
  - Forks on request VM instances for managed processes
  - Copy-on-write to maximise re-use and minimise footprint

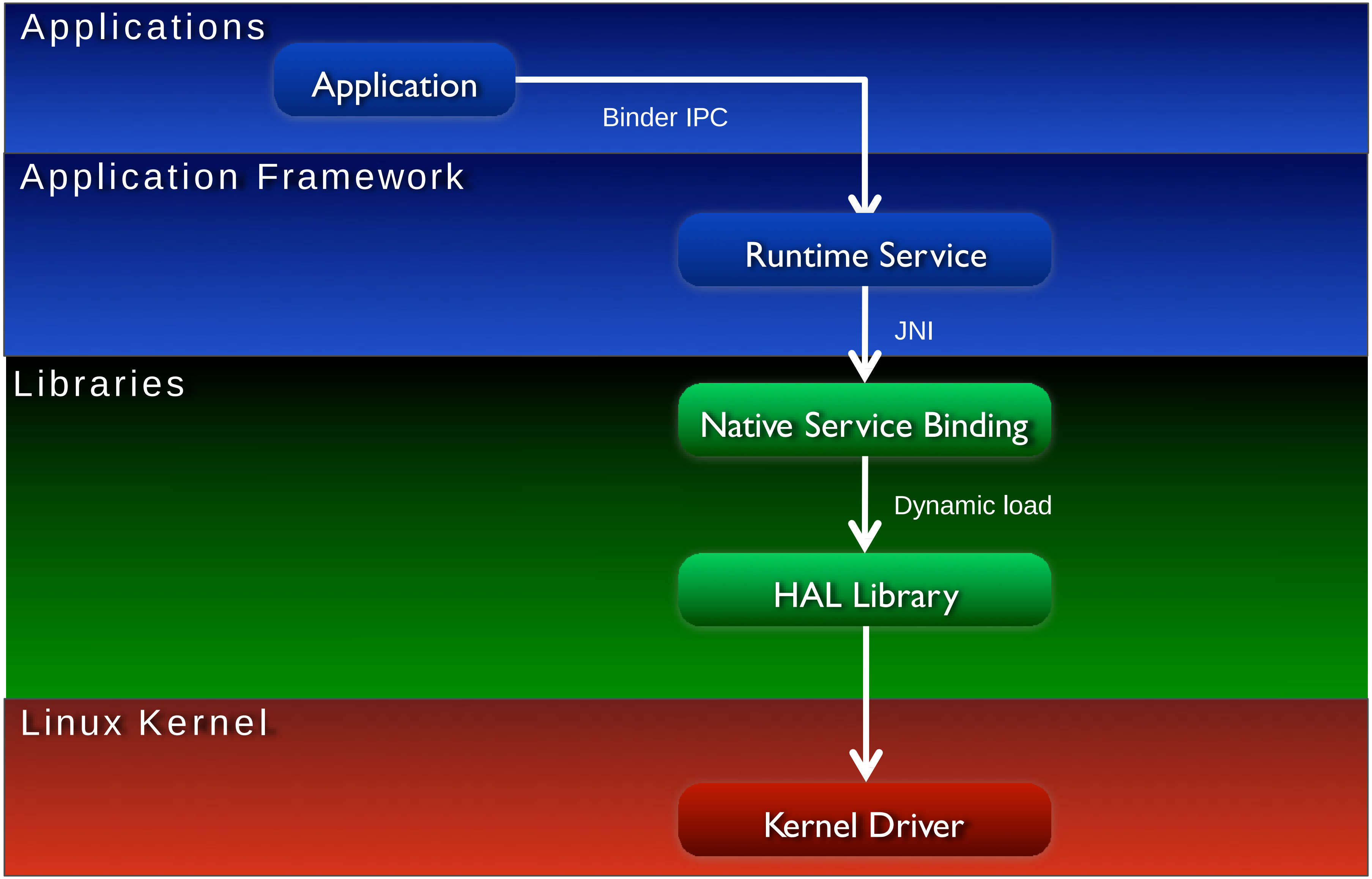


HIGH LEVEL VIEW: NEW APP

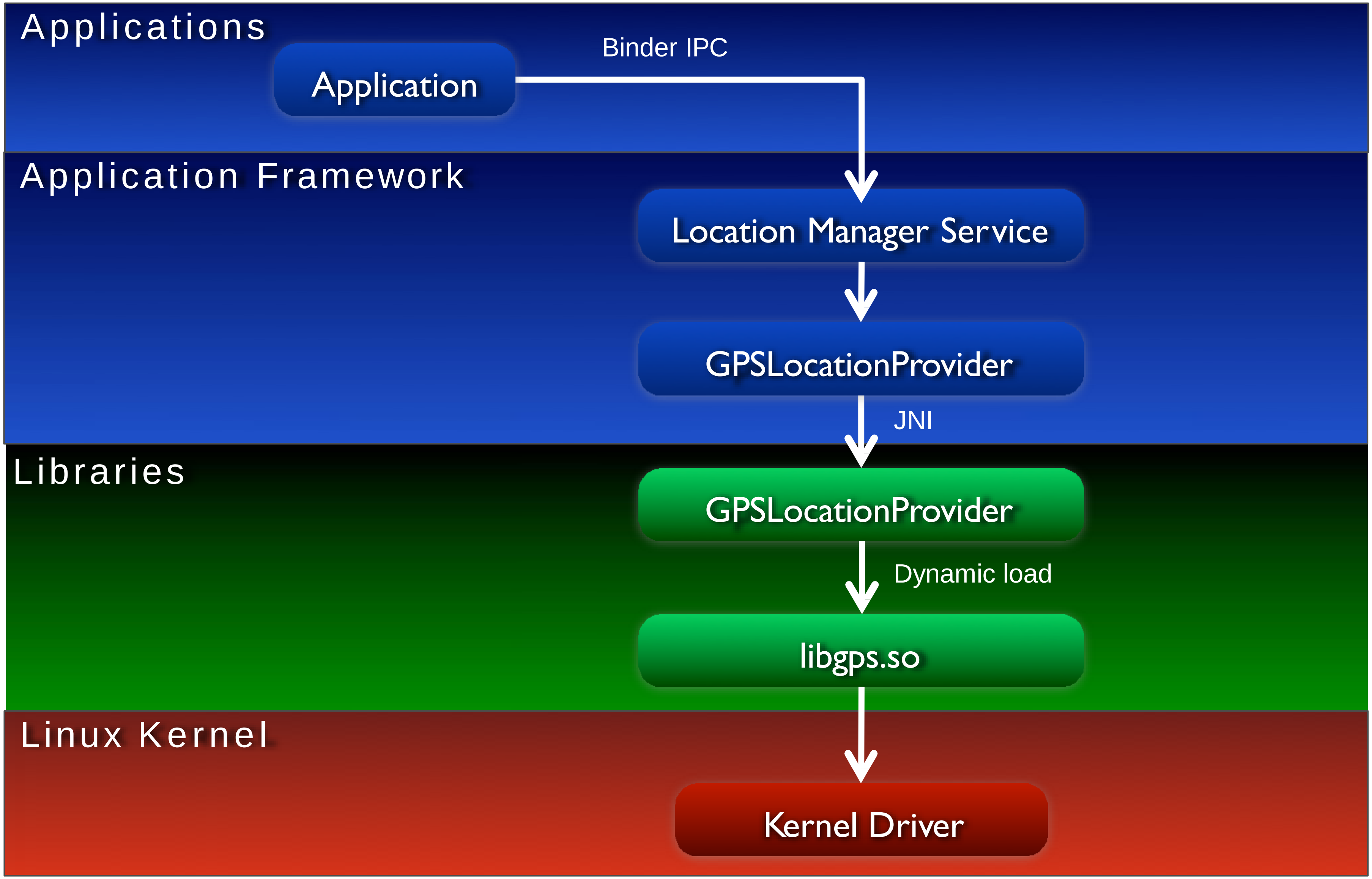
- A DalvikVM per APK



ANDROID RUNTIME SERVICES



EXAMPLE: LOCATION MANAGER



ANDROID SECURITY OBJECTIVES

- Protect user data
- Protect system resources
  - including the network
- Provide application isolation

ANDROID KEY SECURITY FEATURES

- Robust security at the OS level through the Linux kernel
- Mandatory application sandboxing for all applications
- Secure inter-process communication
- Application signing
- Application-defined and user-granted permissions

ACKNOWLEDGEMENT

- Some of the slides in this lecture are based on:
  Android Anatomy and Physiology
  By Patrick Brady

Questions?

Thanks for your attention!