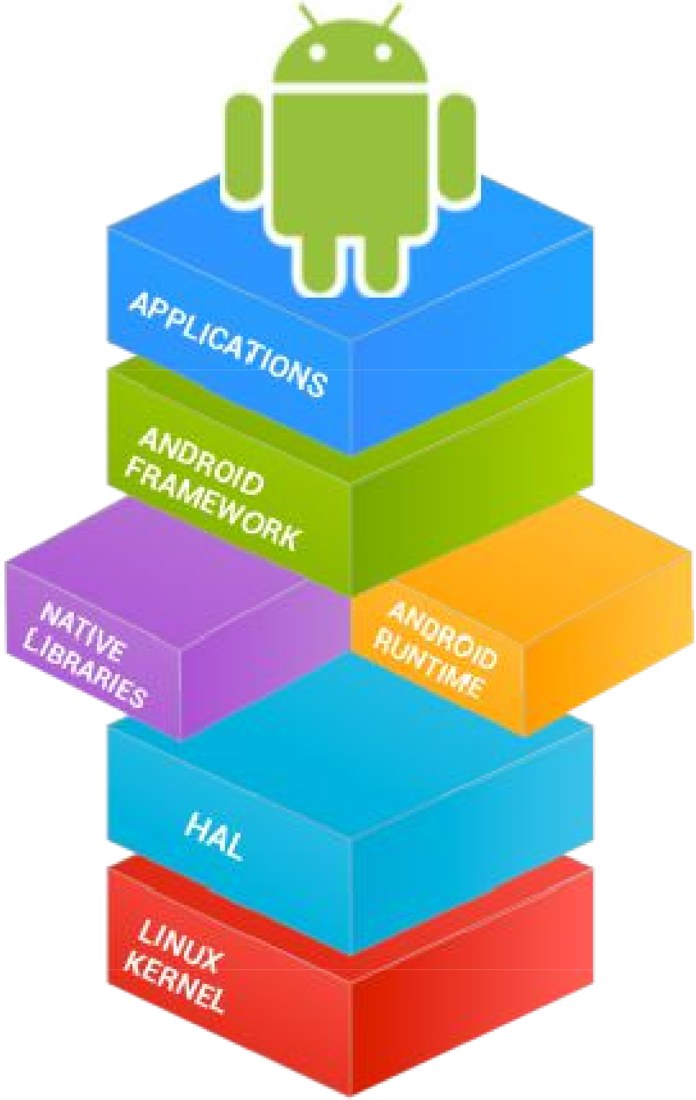
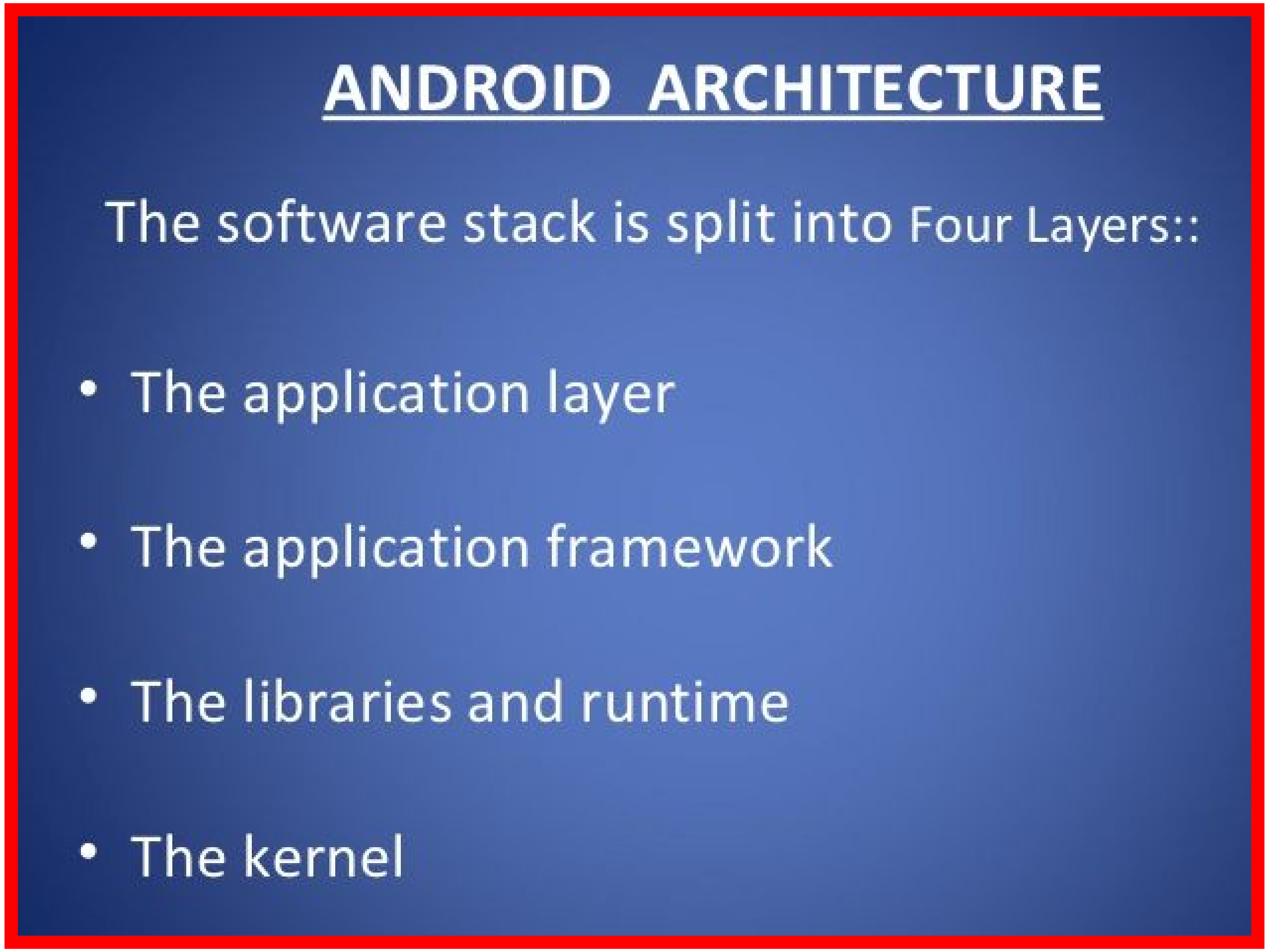
PLATFORM - THE ANDROID SOFTWARE STACK



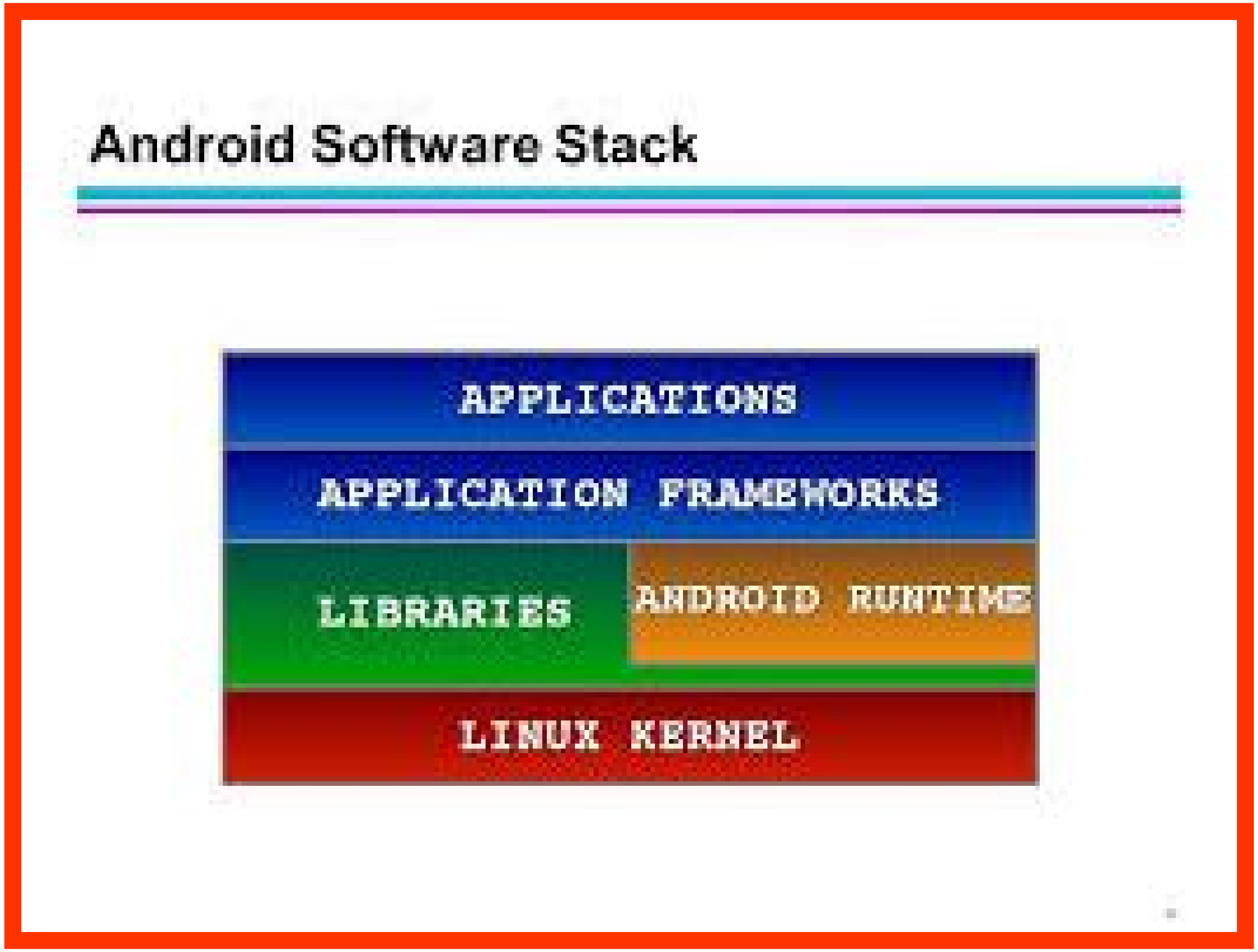
Prof.J.Rexy
Dept of Cs



ANDROID ARCHITECTURE

The software stack is split into Four Layers::

- The application layer
- The application framework
- The libraries and runtime
- The kernel



Android Software Stack

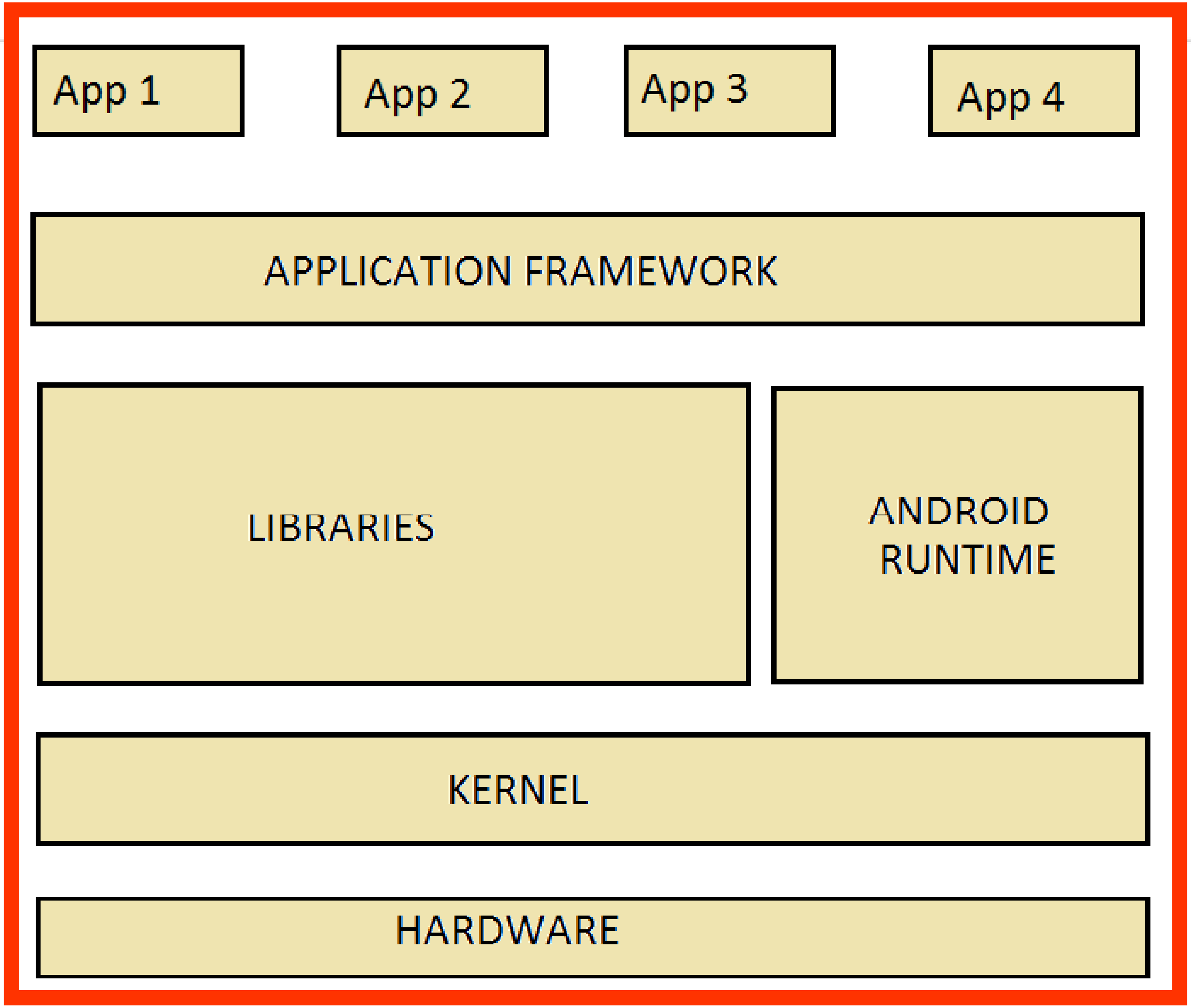
APPLICATIONS

APPLICATION FRAMEWORKS

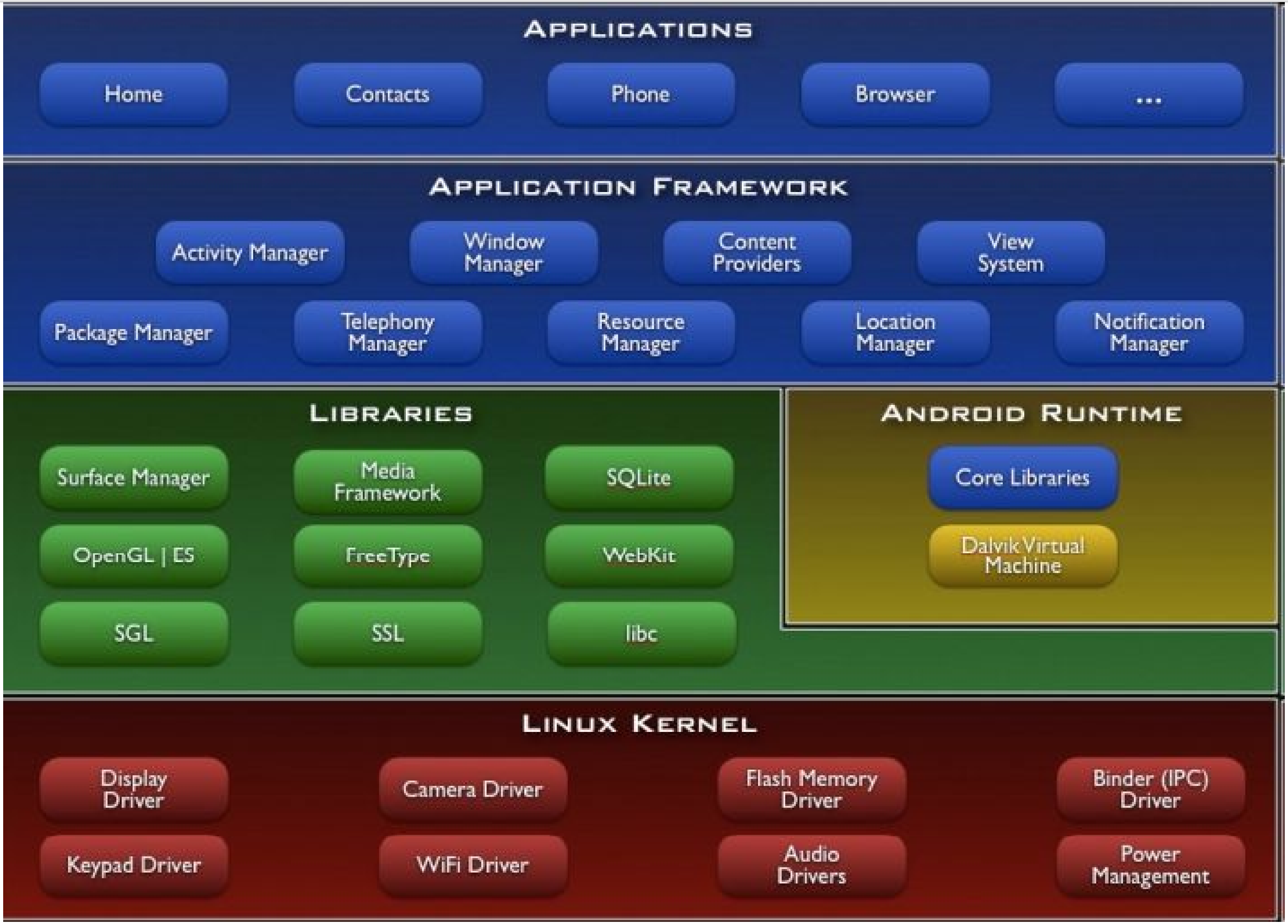
LIBRARIES

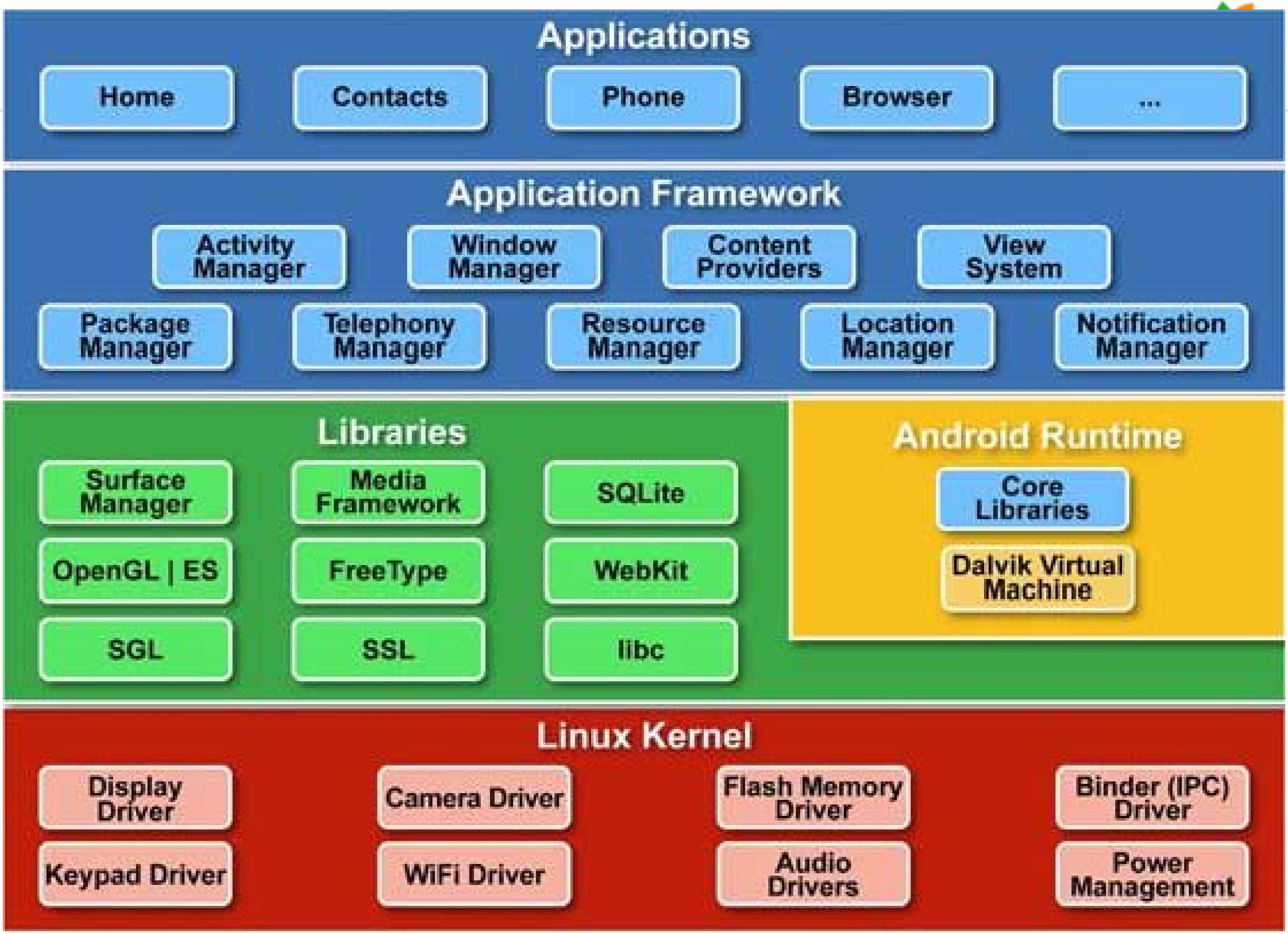
ANDROID RUNTIME

LINUX KERNEL

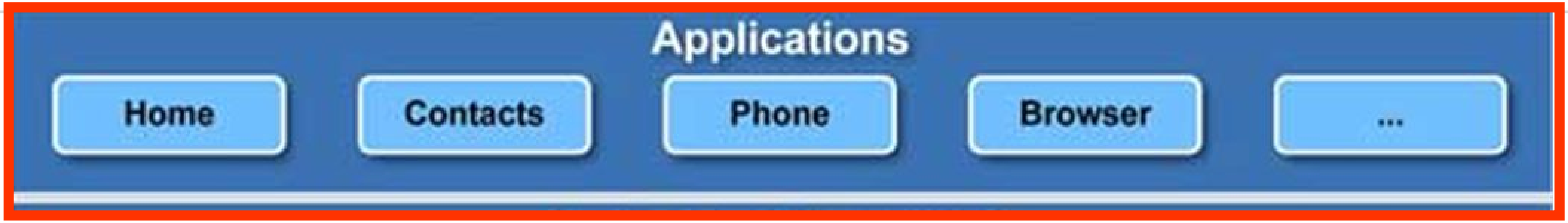

App 1
App 2
App 3
App 4
APPLICATION FRAMEWORK
LIBRARIES
ANDROID RUNTIME
KERNEL
HARDWARE

The Android Software Stack



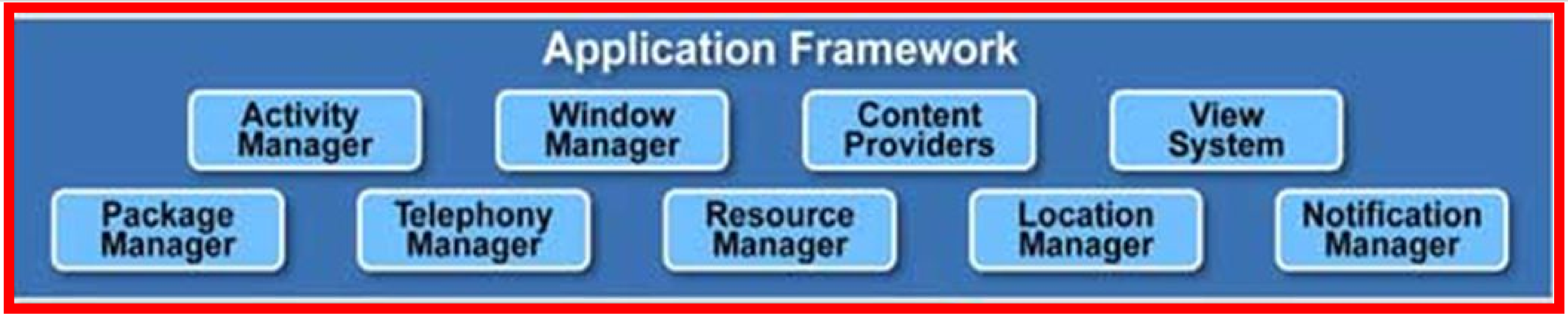

Applications
Home
Contacts
Phone
Browser
...
Application Framework
Activity Manager
Window Manager
Content Providers
View System
Package Manager
Telephony Manager
Resource Manager
Location Manager
Notification Manager
Libraries
Surface Manager
Media Framework
SQLite
OpenGL | ES
FreeType
WebKit
SGL
SSL
libc
Android Runtime
Core Libraries
Dalvik Virtual Machine
Linux Kernel
Display Driver
Camera Driver
Flash Memory Driver
Binder (IPC) Driver
Keypad Driver
WiFi Driver
Audio Drivers
Power Management

Android S/W Stack - Application



- Android provides a set of core applications:
  - ✓ Email Client
  - ✓ SMS Program
  - ✓ Calendar
  - ✓ Maps
  - ✓ Browser
  - ✓ Contacts
  - ✓ Etc
- All applications are written using the Java language.

Android S/W Stack – App Framework

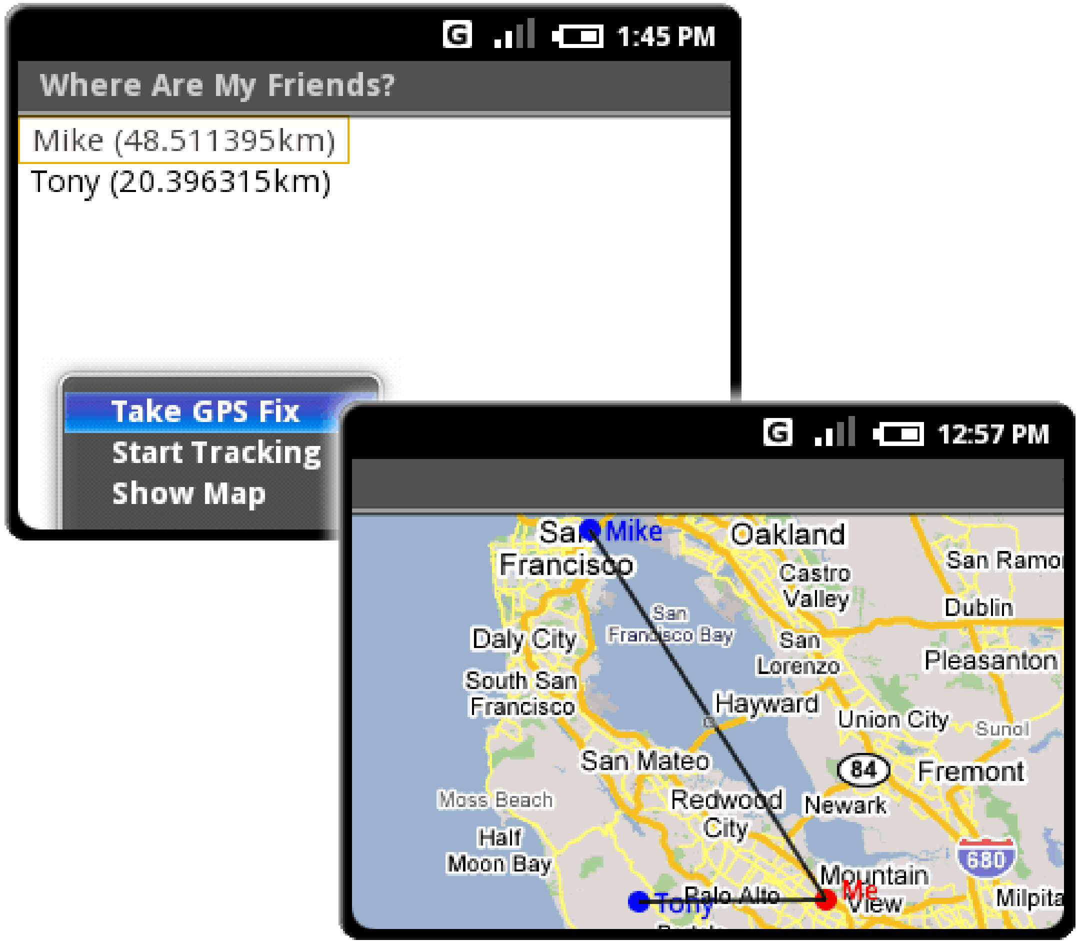


- Provides tools for development of Android Applications.
- Activity Manager: manages activity lifecycle of application.
- Content Provider: manages data sharing among applications.
- Telephony Manager: manages all voice calls
- Location Manager: Location update using GPS or cell tower.
- Resource Manager: manages various types of resources we use in our application.

Android S/W Stack –
App Framework (Cont)

Feature	Role
View System	Used to build an application, including lists, grids, text boxes, buttons, and embedded web browser
Content Provider	Enabling applications to access data from other applications or to share their own data
Resource Manager	Providing access to non-code resources (localized string , graphics, and layout files)
Notification Manager	Enabling all applications to display customer alerts in the status bar
Activity Manager	Managing the lifecycle of applications and providing a common navigation backstack

Location Manager

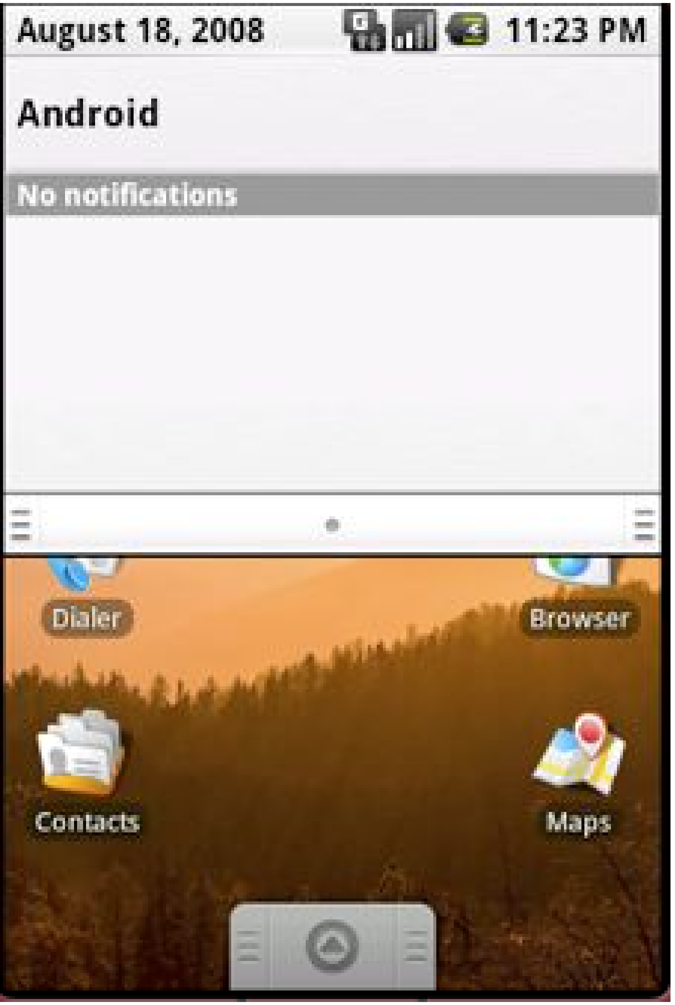


Where Are My Friends?

Mike (48.511395km)

Tony (20.396315km)

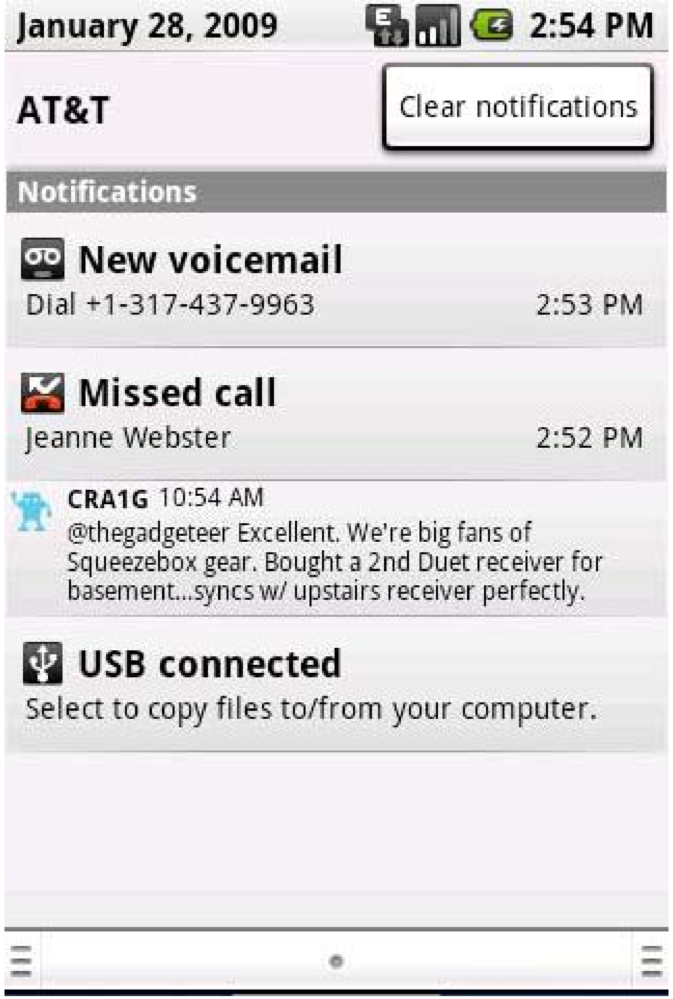
Notification Manager



August 18, 2008 11:23 PM

Android

No notifications



January 28, 2009 2:54 PM

AT&T

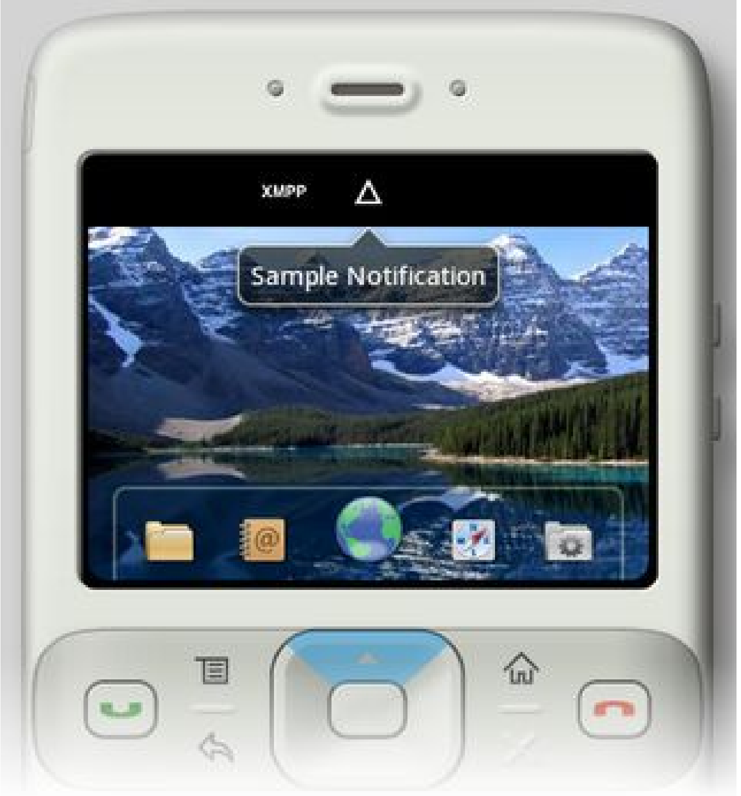
Notifications

New voicemail
Dial +1-317-437-9963 2:53 PM

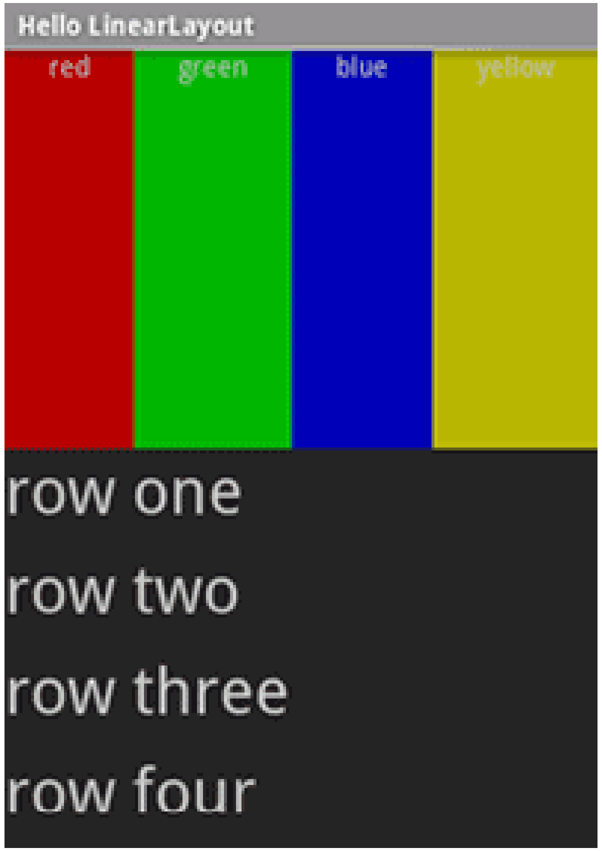
Missed call
Jeanne Webster 2:52 PM

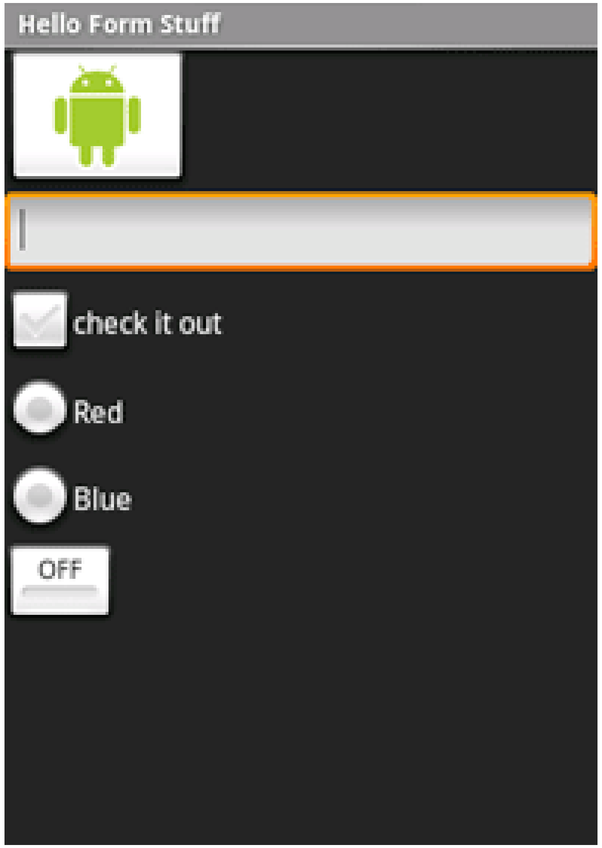
CRA1G 10:54 AM
@thegadgeteer Excellent. We're big fans of Squeezebox gear. Bought a 2nd Duet receiver for basement...syncs w/ upstairs receiver perfectly.

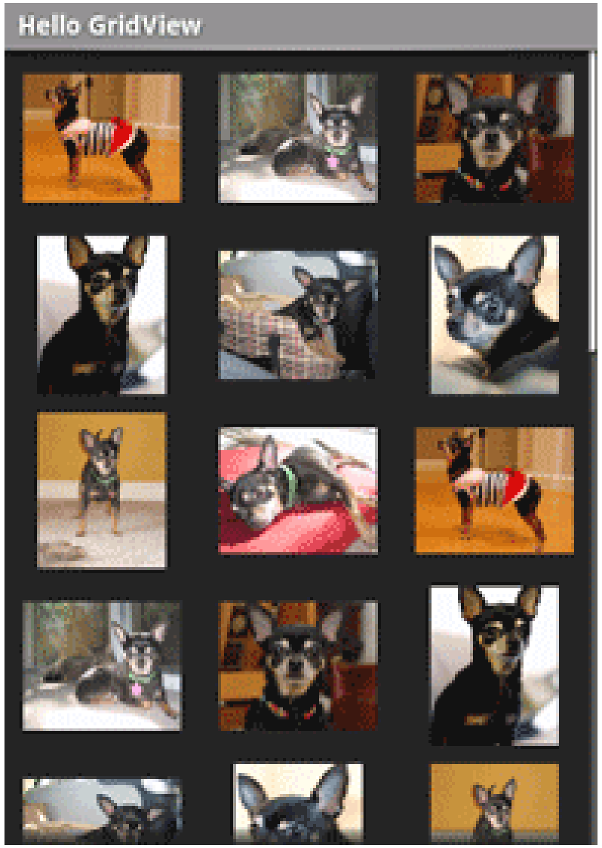
USB connected
Select to copy files to/from your computer.



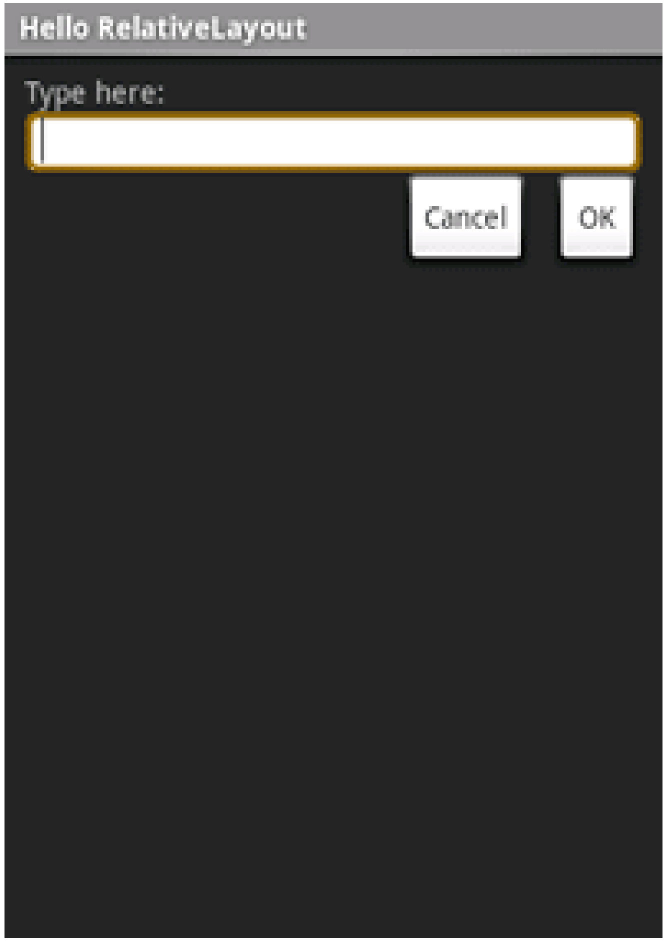
View System

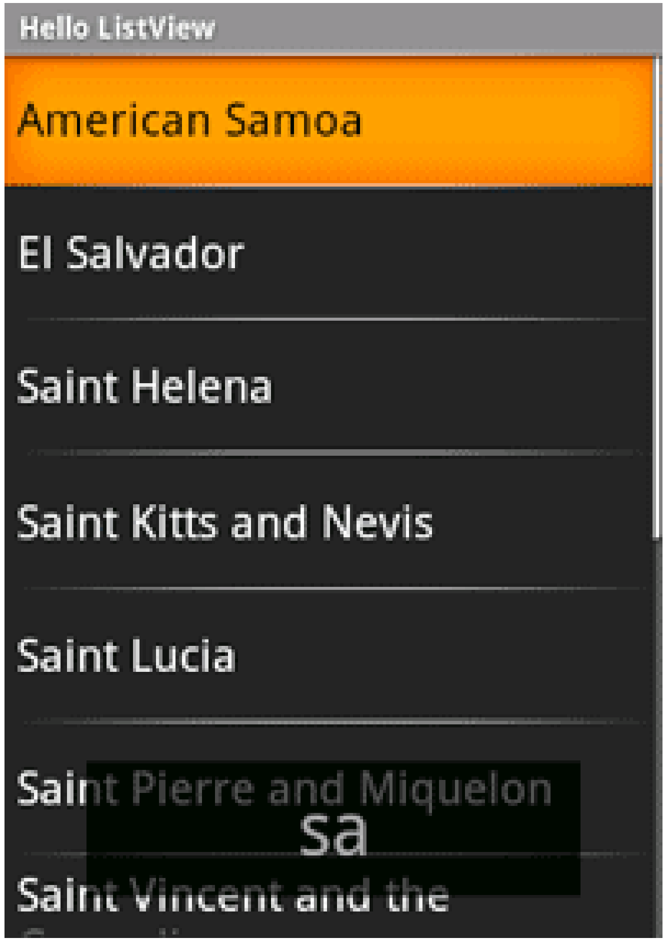

Hello LinearLayout
red
green
blue
yellow
row one
row two
row three
row four


Hello Form Stuff
☑ check it out
○ Red
○ Blue
OFF


Hello GridView

View System





American Samoa

El Salvador

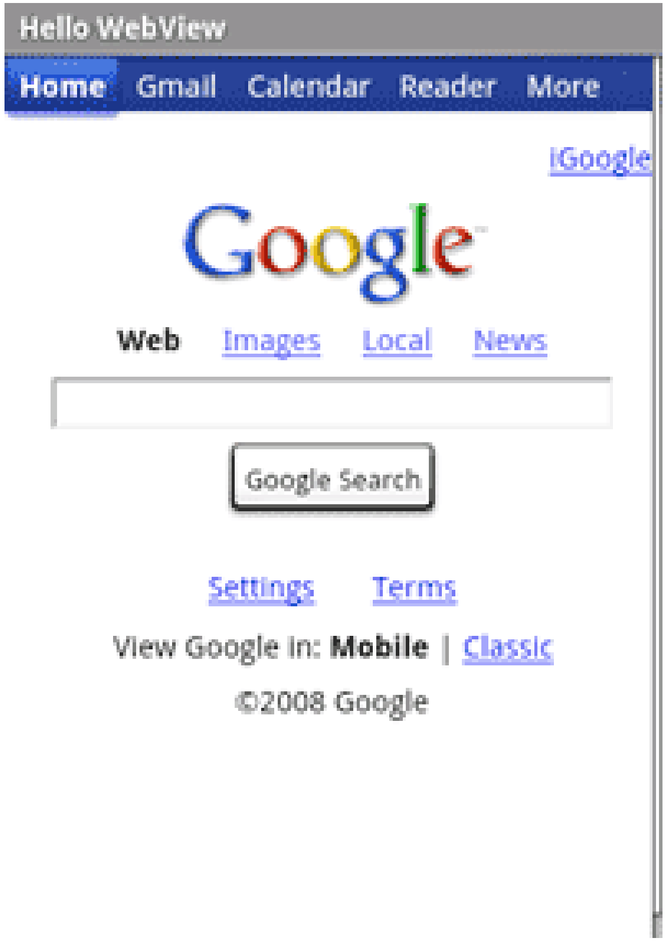
Saint Helena

Saint Kitts and Nevis

Saint Lucia

Saint Pierre and Miquelon

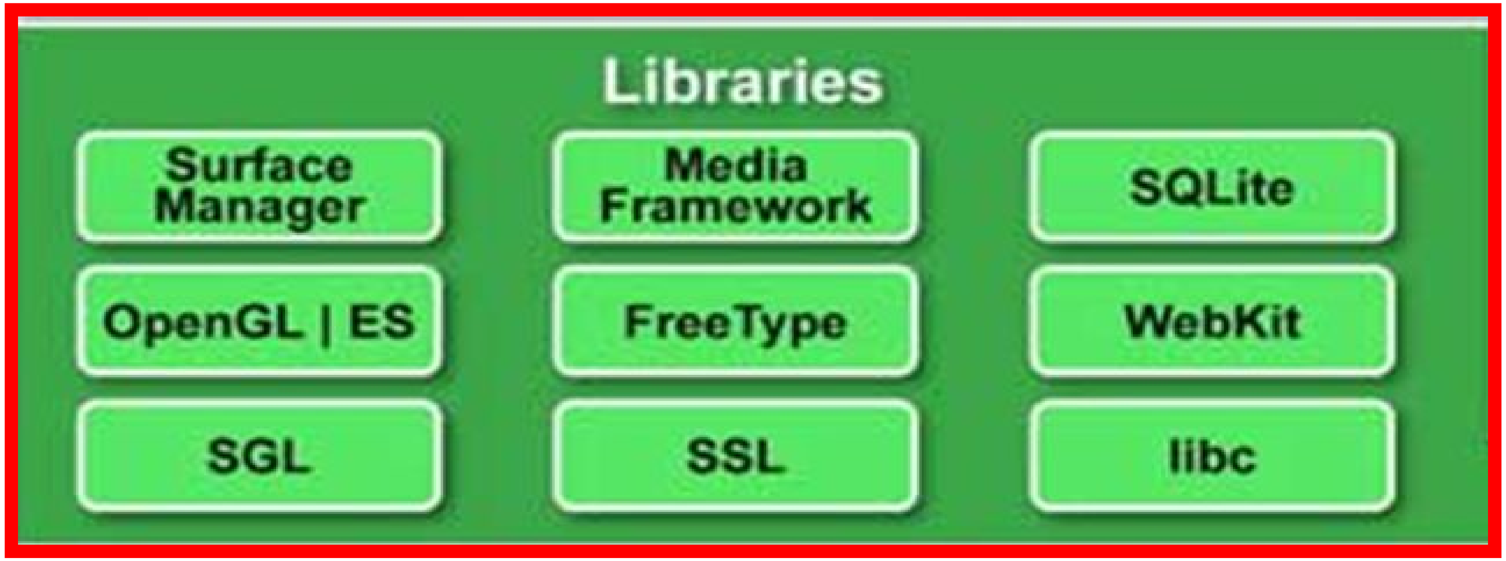
Saint Vincent and the



View Google in: **Mobile** | Classic

©2008 Google

Android S/W Stack - Libraries



Libraries

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

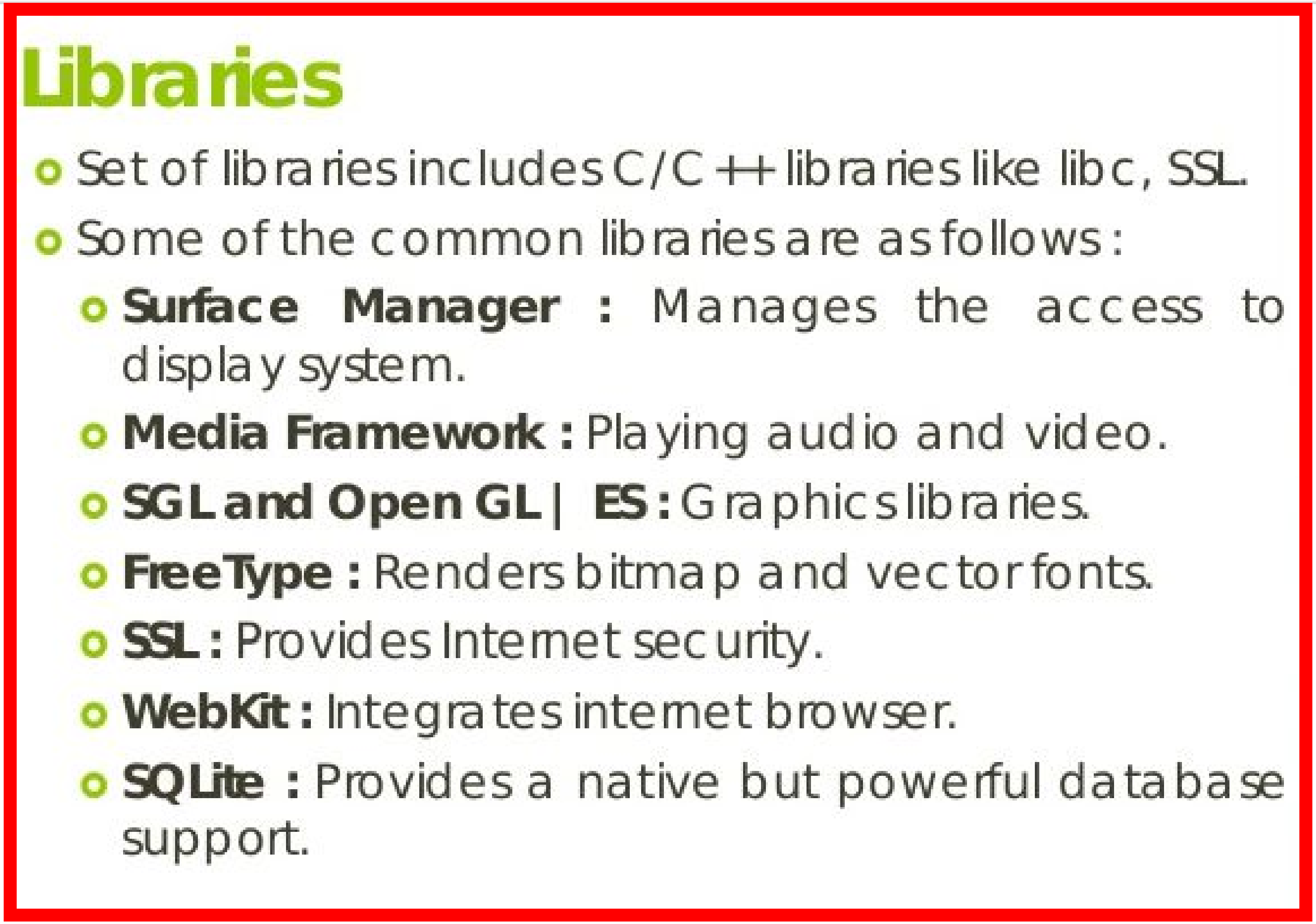
- ✓ System C Library (Bionic)
- ✓ Media Libraries
- ✓ Surface Manager (Surface Flinger)
- ✓ Audio Manager (Audio Flinger)
- ✓ LibWebCore (WebKit)
- ✓ SGL
- ✓ 3D Libraries
- ✓ FreeType
- ✓ SQLite

- Android Libraries are develped in C/C++
- We cannot access these libraries directly, to access these libraries we need application framework

Surface Manager – used for display management

Open GL/ES(Open Graphics Library/Extended system) **and SGL(Scalable Graphics Library) –** used for 2D/3D Graphics mainly for game development

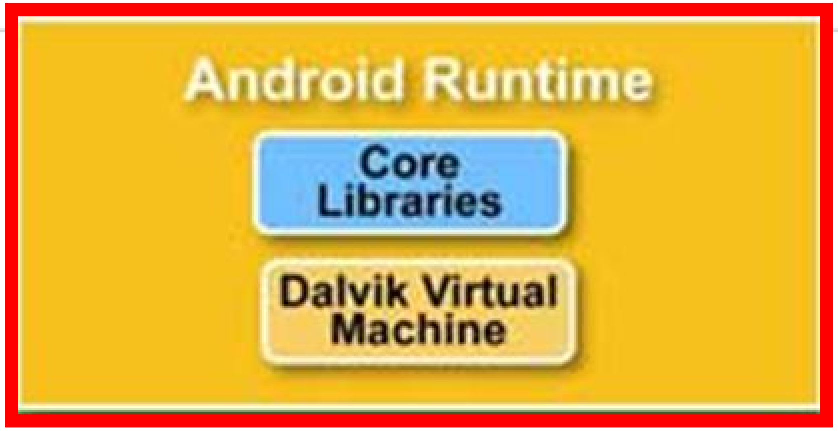
Media Framework – (PacketVideo)Audio/Video codes(mp3, mpg4)



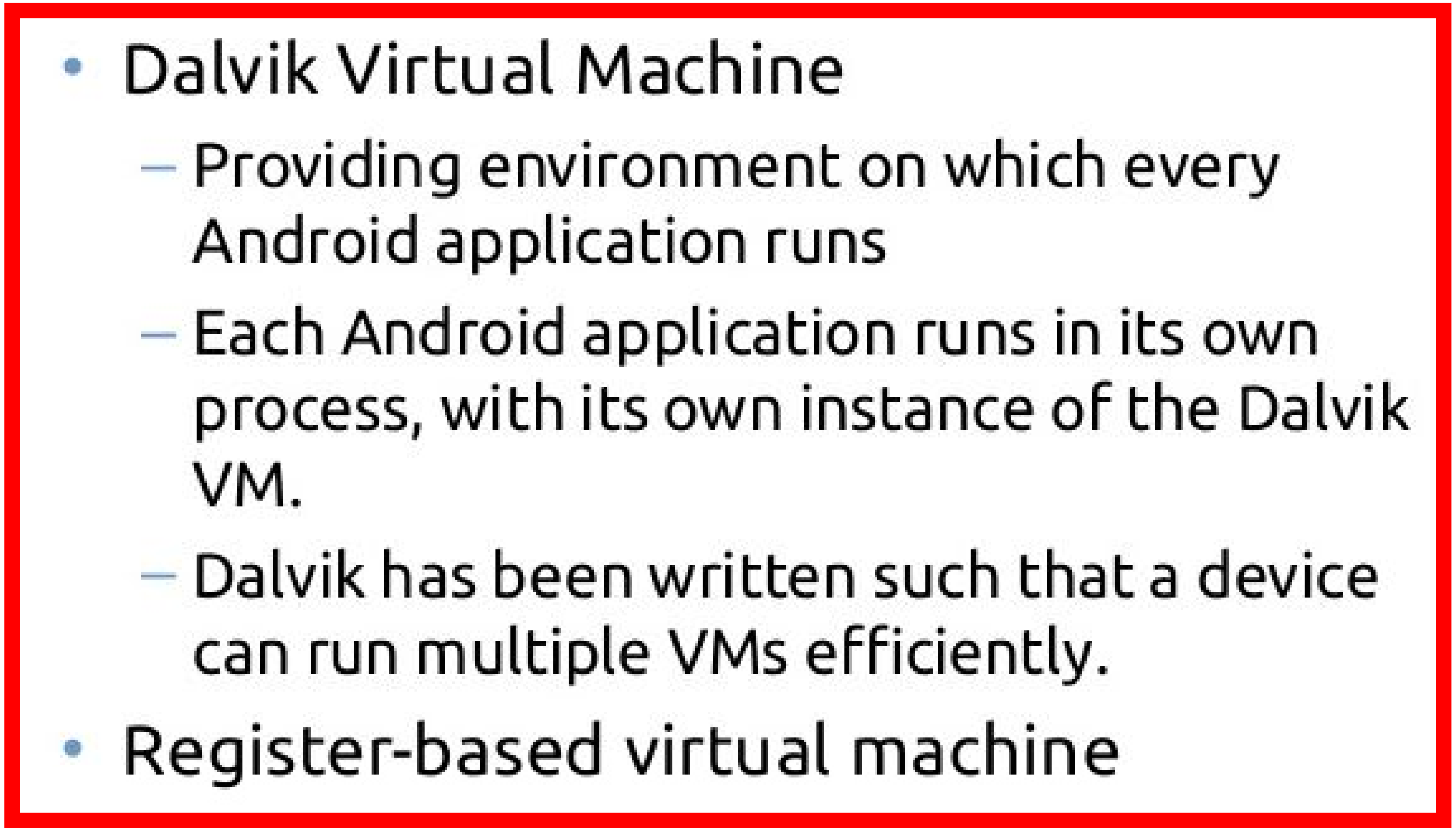
Libraries

- Set of libraries includes C/C ++ libraries like libc, SSL.
- Some of the common libraries are as follows :
  - **Surface Manager :** Manages the access to display system.
  - **Media Framework :** Playing audio and video.
  - **SGL and Open GL | ES :** Graphics libraries.
  - **FreeType :** Renders bitmap and vector fonts.
  - **SSL :** Provides Internet security.
  - **WebKit :** Integrates internet browser.
  - **SQLite :** Provides a native but powerful database support.

Android S/W Stack - Runtime



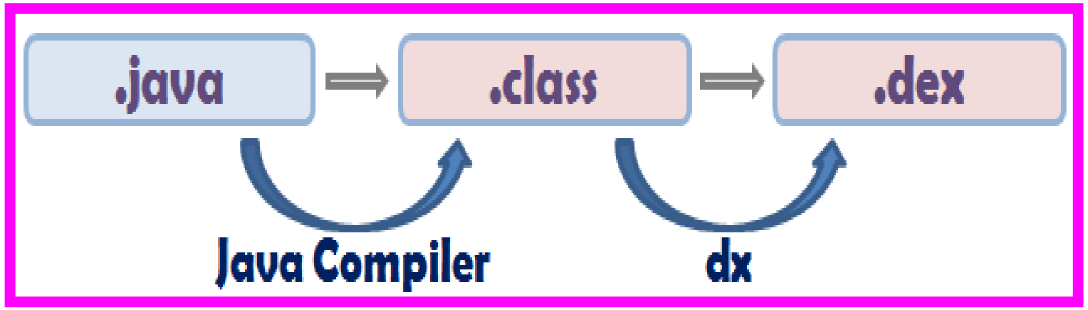
- Core Libraries
  - ✓Providing most of the functionality available in the core libraries of the Java language
  - ✓APIs
    - ➢Data Structures
    - ➢Utilities
    - ➢File Access
    - ➢Network Access
    - ➢Graphics
    - ➢Etc



- Dalvik Virtual Machine
  - Providing environment on which every Android application runs
  - Each Android application runs in its own process, with its own instance of the Dalvik VM.
  - Dalvik has been written such that a device can run multiple VMs efficiently.
- Register-based virtual machine

Android S/W Stack – Runtime (Cont)

- Dalvik Virtual Machine (Cont)
  - ✓Executing the Dalvik Executable (.dex) format
    - ➢.dex format is optimized for minimal memory footprint.
    - ➢Compilation



  - ✓Relying on the Linux Kernel for:
    - ➢Threading
    - ➢Low-level memory management

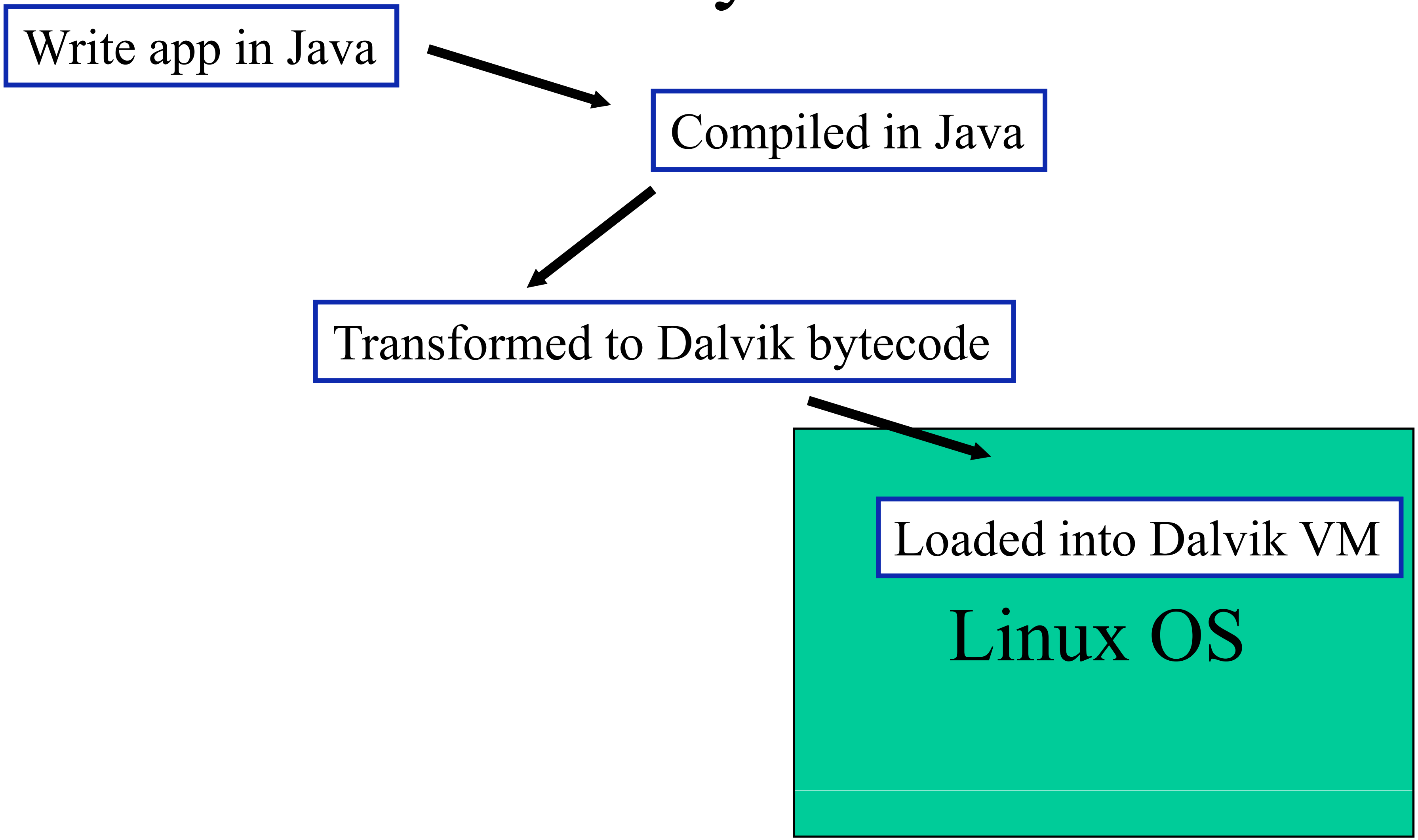
The Dalvik runtime is optimised for mobile applications

Run multiple VMs efficiently

Each app has its own VM

Minimal memory footprint

Android applications are compiled to Dalvik bytecode



Write app in Java

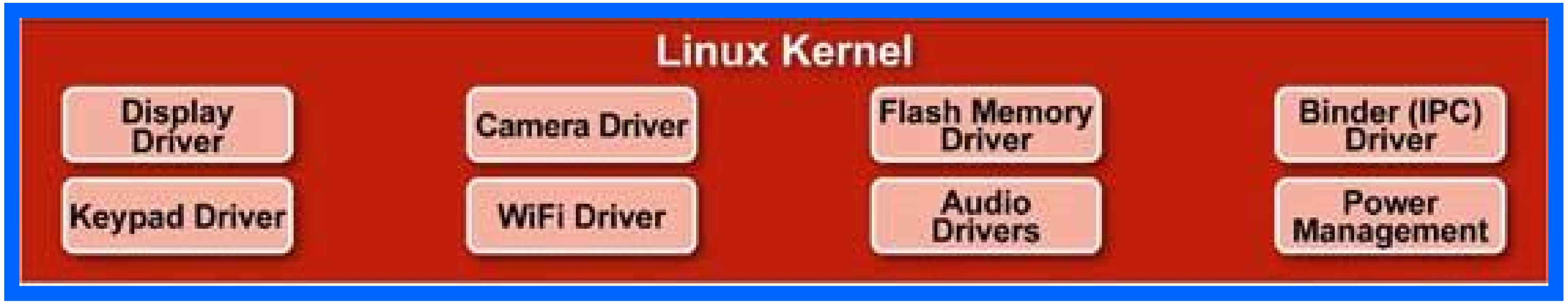
Compiled in Java

Transformed to Dalvik bytecode

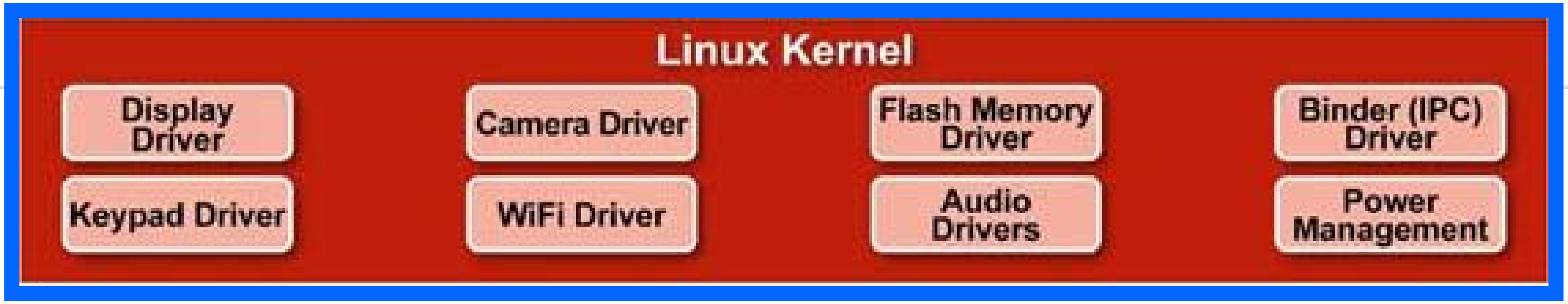
Loaded into Dalvik VM

Linux OS

Android S/W Stack – Linux Kernel



- Relying on Linux Kernel 2.6 for core system services
  - ✓ Memory and Process Management
  - ✓ Network Stack
  - ✓ Driver Model
  - ✓ Security
- The supplied device drivers include Display, Camera, Keypad, WiFi, Flash Memory, Audio, and IPC (interprocess communication).
- Providing an abstraction layer between the H/W and the rest of the S/W stack



Linux Kernel

Display Driver

Camera Driver

Flash Memory Driver

Binder (IPC) Driver

Keypad Driver

WiFi Driver

Audio Drivers

Power Management

- Works as a **Hardware Abstraction Layer (HAL)**
- Device drivers
- Memory management
- Process management
- Networking

Thank you

Applications

Android framework

Libraries | Android Runtime

Linux Kernal