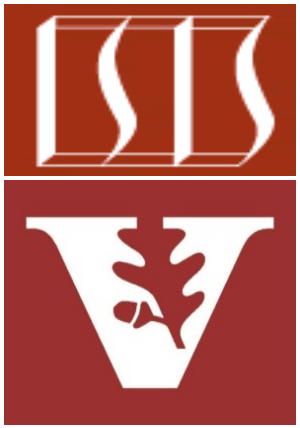
Infrastructure Middleware (Part 1): Hardware Abstraction Layer (HAL)

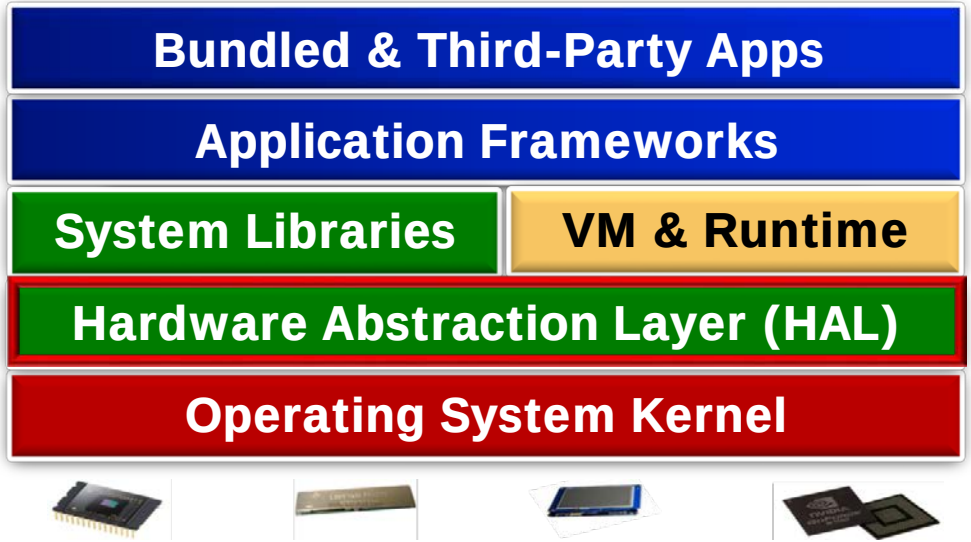


Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Institute for Software
Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Part of the Lesson

- Understand the role the HAL plays in Android's system architecture



The HAL mediates interactions between the OS kernel layer & higher layers

Learning Objectives in this Part of the Lesson

- Understand the role the HAL plays in Android's system architecture
- Know the technical & economic factors that motivated Google & its original equipment manufacturer (OEM) partners to add a HAL to the Android software stack

Learning Objectives in this Part of the Lesson

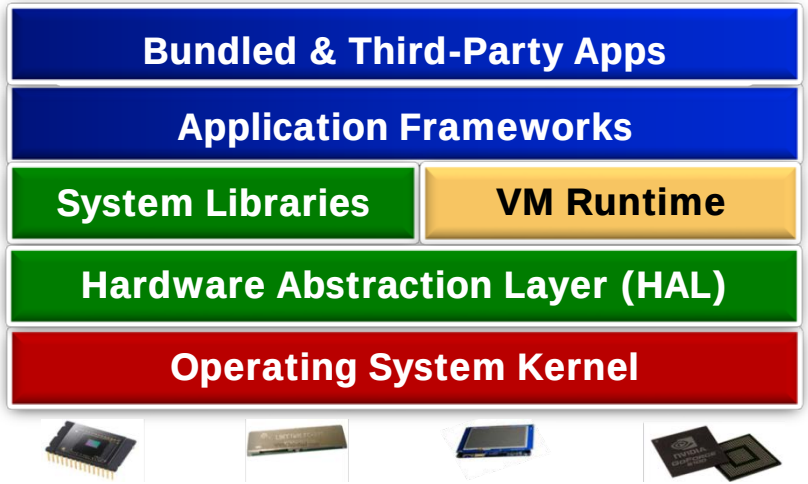
- Understand the role the HAL plays in Android's system architecture
- Know the technical & economic factors that motivated Google & its original equipment manufacturer (OEM) partners to add a HAL to the Android software stack

Android apps rarely access the HAL directly, so you needn't know all the details

Overview of Android Hardware Abstraction Layer

The Android Hardware Abstraction Layer (HAL)

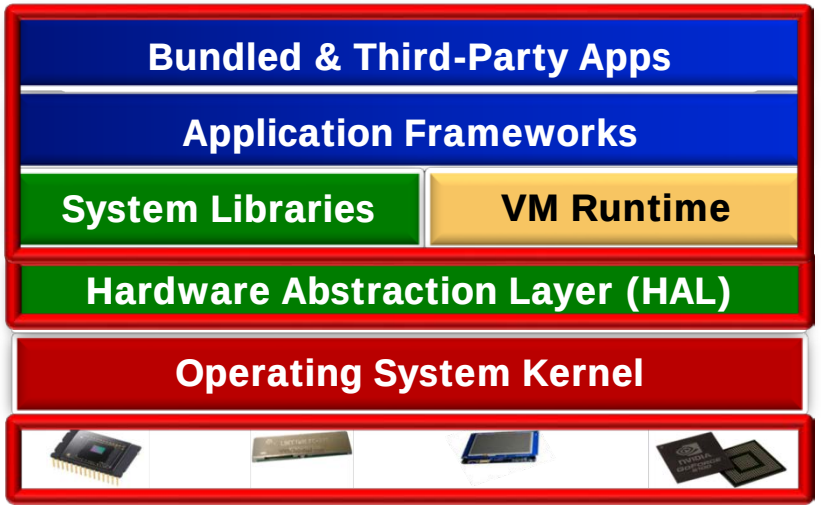
- The HAL helps to separate concerns in the Android system architecture



See source.android.com/devices/#Hardware Abstraction Layer

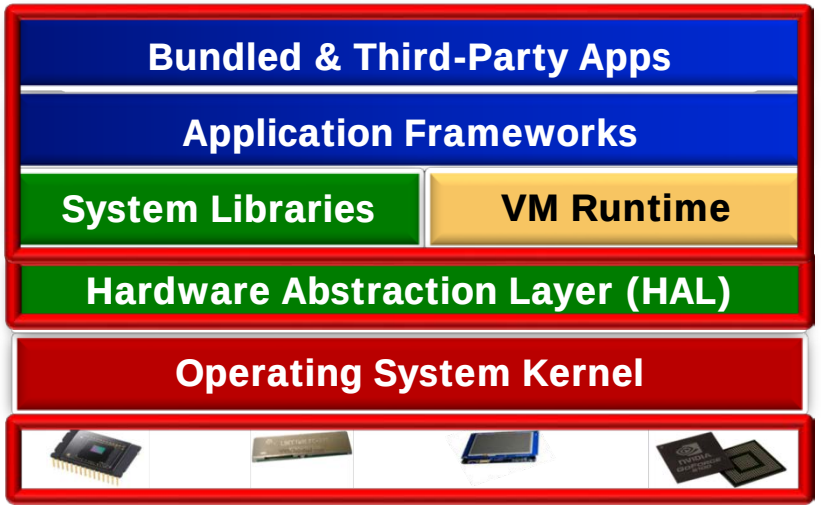
The Android Hardware Abstraction Layer (HAL)

- The HAL helps to separate concerns in the Android system architecture, e.g.
  - Decouples Android platform software from hardware



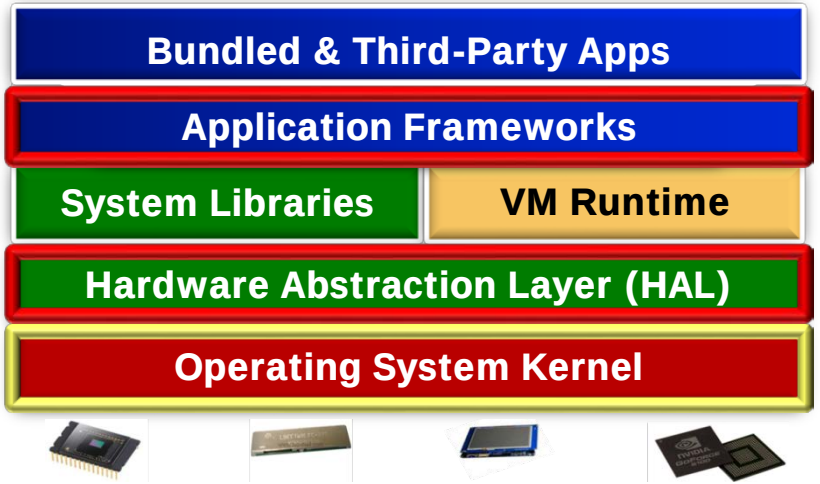
The Android Hardware Abstraction Layer (HAL)

- The HAL helps to separate concerns in the Android system architecture, e.g.
  - Decouples Android platform software from hardware
    - i.e., apps don't break when hardware changes



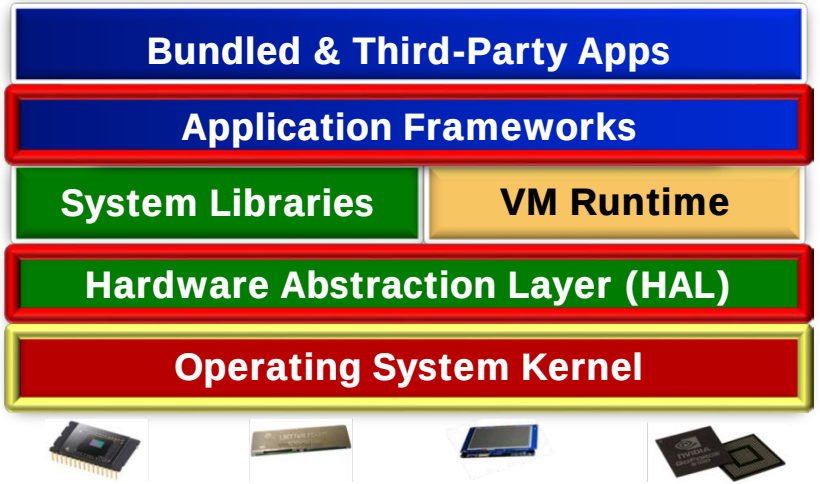
The Android Hardware Abstraction Layer (HAL)

- The HAL helps to separate concerns in the Android system architecture, e.g.
  - Decouples Android platform software from hardware
  - Decouples Android framework layer from the operating system layer



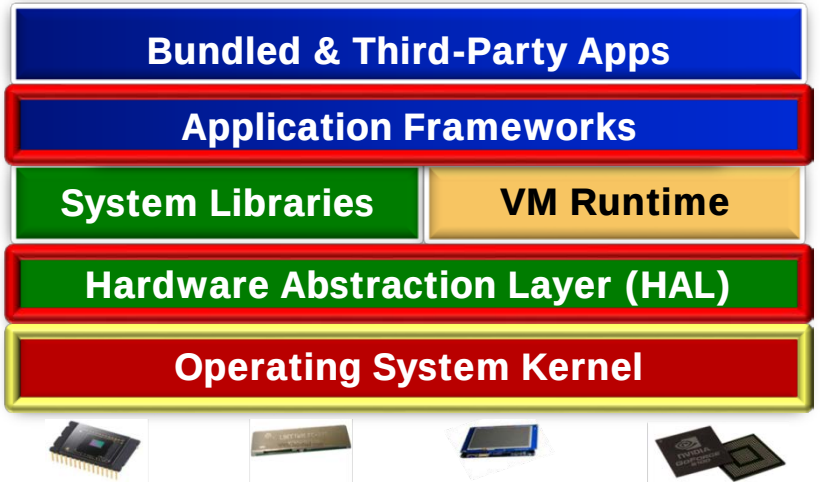
The Android Hardware Abstraction Layer (HAL)

- The HAL helps to separate concerns in the Android system architecture, e.g.
  - Decouples Android platform software from hardware
  - Decouples Android framework layer from the operating system layer
    - e.g., this layer can be programmed via method calls on Java objects, rather than C system function calls



The Android Hardware Abstraction Layer (HAL)

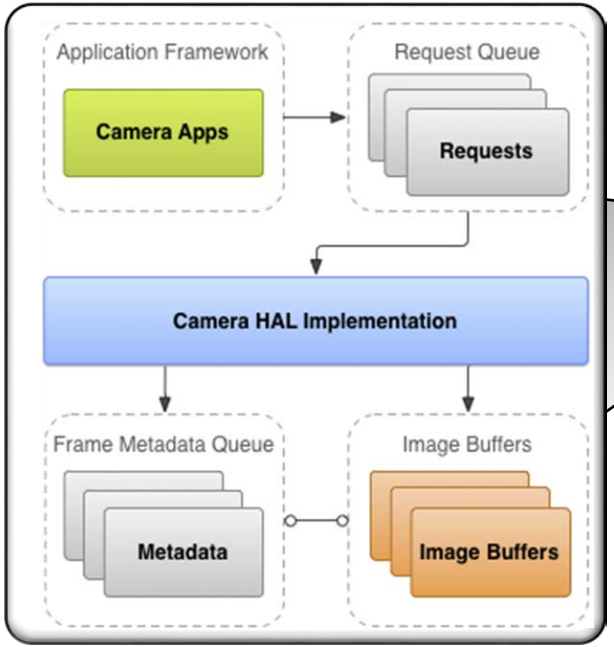
- The HAL helps to separate concerns in the Android system architecture, e.g.
  - Decouples Android platform software from hardware
  - Decouples Android framework layer from the operating system layer
    - e.g., this layer can be programmed via method calls on Java objects, rather than C system function calls

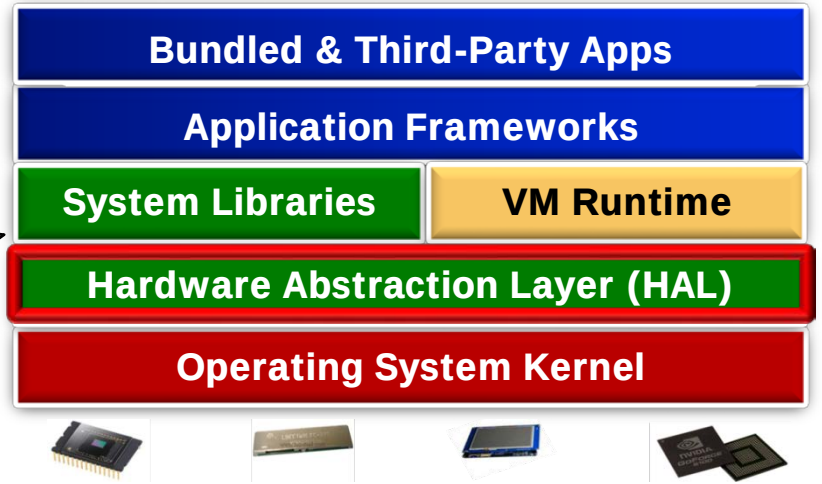


App developers can thus focus on business logic, rather than low-level details

The Android Hardware Abstraction Layer (HAL)

- Android's HAL defines interfaces that driver modules must implement

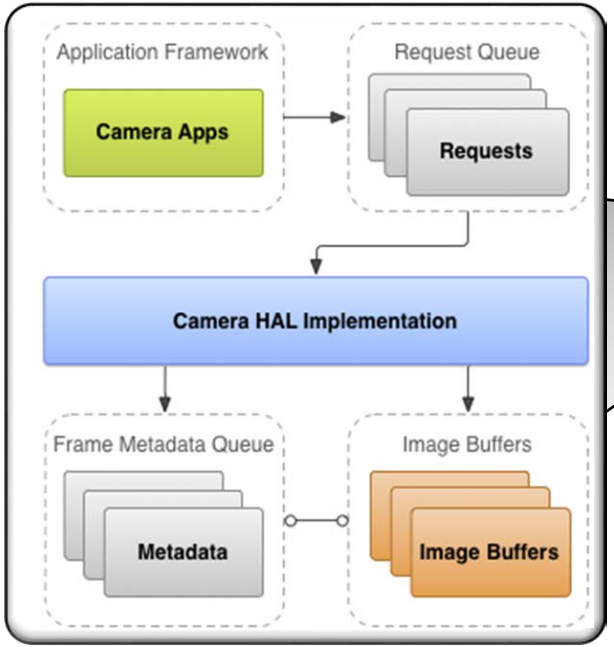


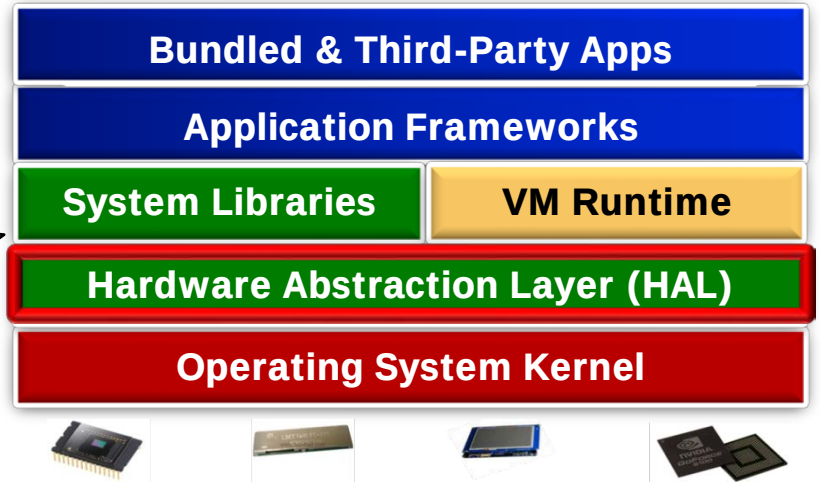


See source.android.com/devices/#structure

The Android Hardware Abstraction Layer (HAL)

- Android's HAL defines interfaces that driver modules must implement

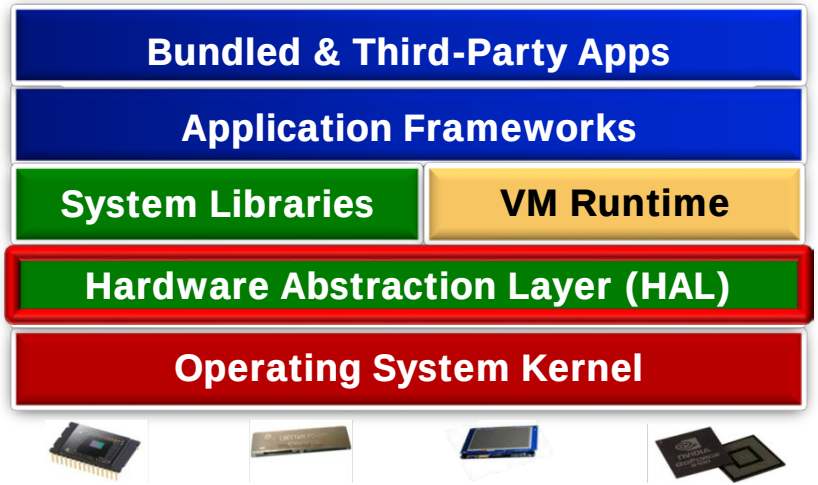




These driver modules do not run in the Android Linux kernel!

The Android Hardware Abstraction Layer (HAL)

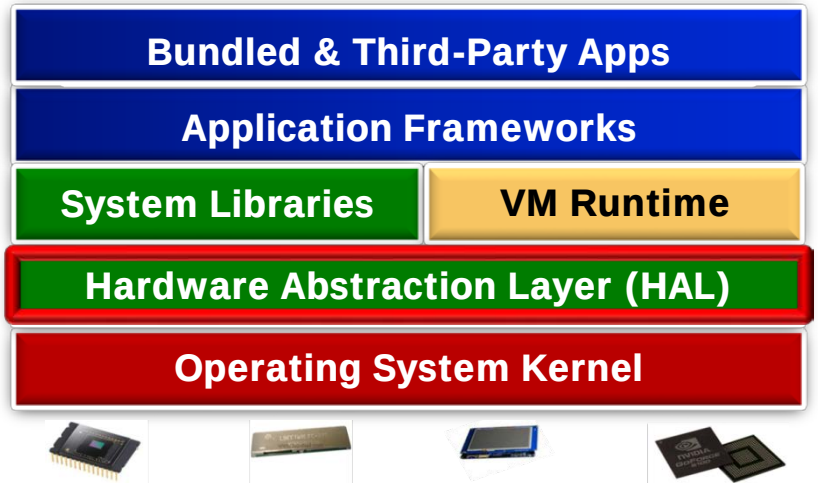
- The Android HAL is a *user space* C/C++ library layer, unlike device drivers residing in the Android Linux kernel



See earlier lesson on the “Android Linux Kernel”

The Android Hardware Abstraction Layer (HAL)

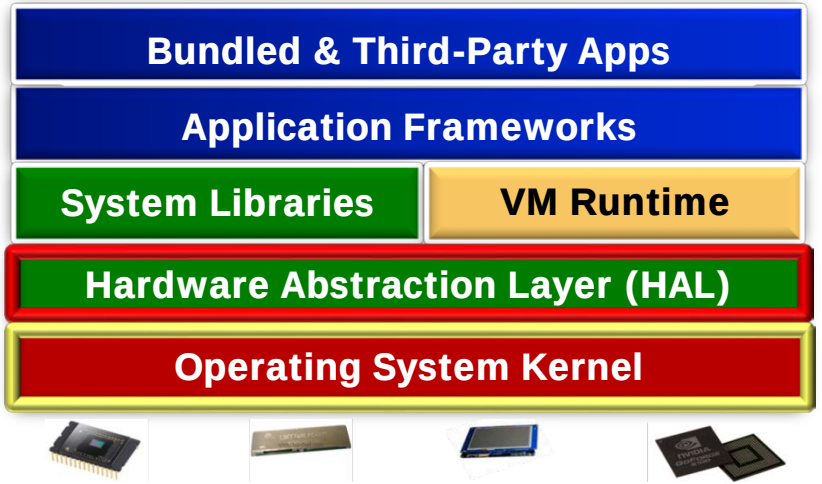
- The Android HAL is a *user space* C/C++ library layer, unlike device drivers residing in the Android Linux kernel
  - User space code can't directly access Android Linux kernel device drivers



See earlier lesson on the "Android Linux Kernel"

The Android Hardware Abstraction Layer (HAL)

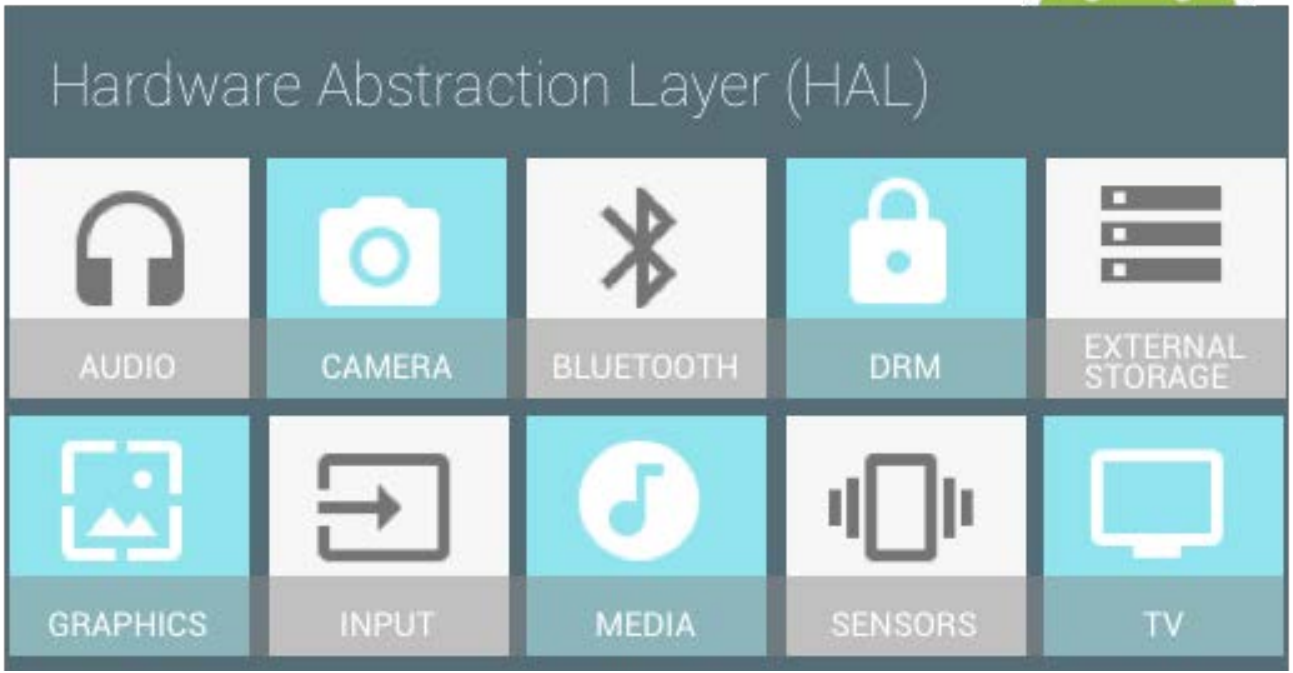
- The Android HAL is a *user space* C/C++ library layer, unlike device drivers residing in the Android Linux kernel
  - User space code can't directly access Android Linux kernel device drivers
  - Therefore apps & system services use the HAL to interact w/kernel drivers



See www.dre.vanderbilt.edu/~schmidt/PDF/Android-HAL.pdf

The Android Hardware Abstraction Layer (HAL)

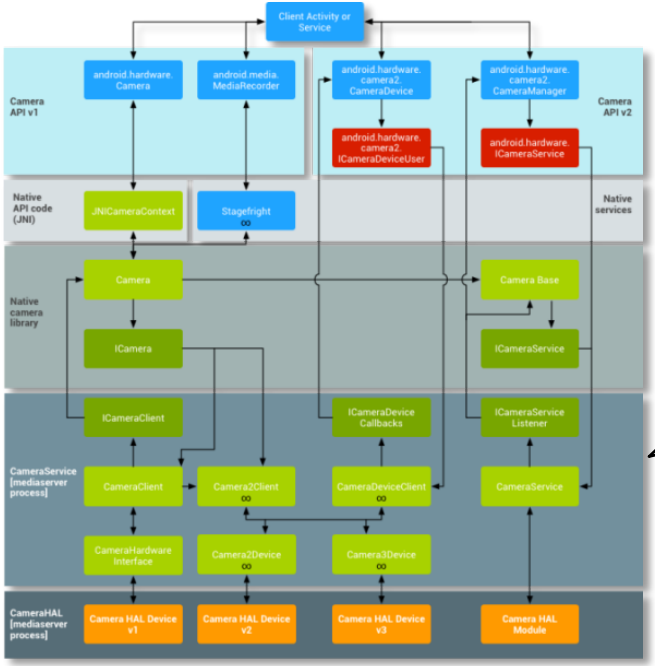
- OEMs implement HAL driver modules for their proprietary hardware

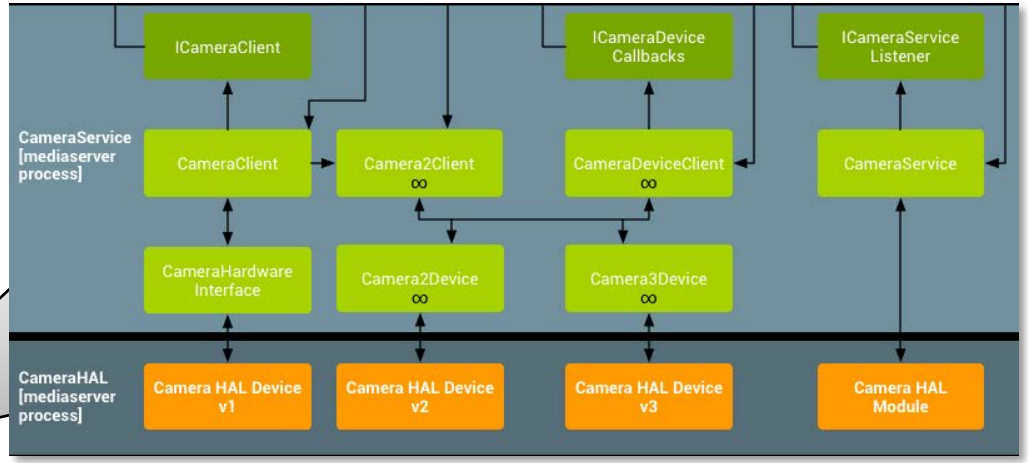


See source.android.com/devices/#Hardware Abstraction Layer

The Android Hardware Abstraction Layer (HAL)

- OEMs implement HAL driver modules for their proprietary hardware



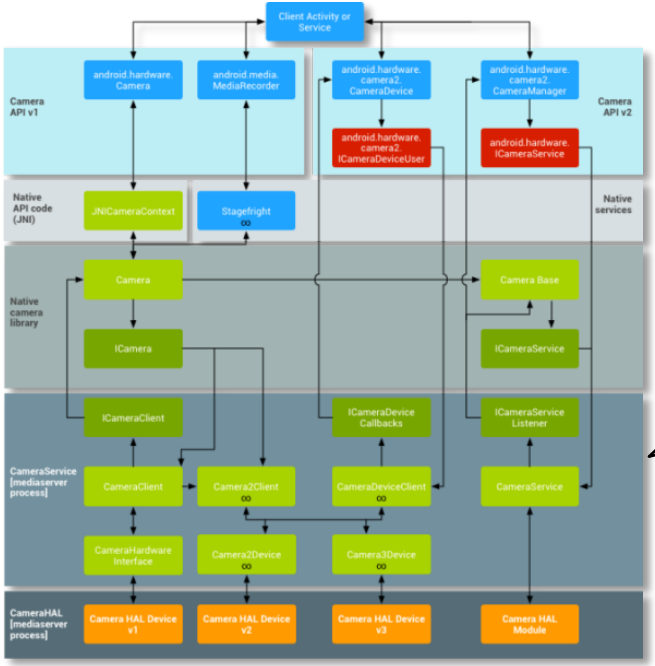


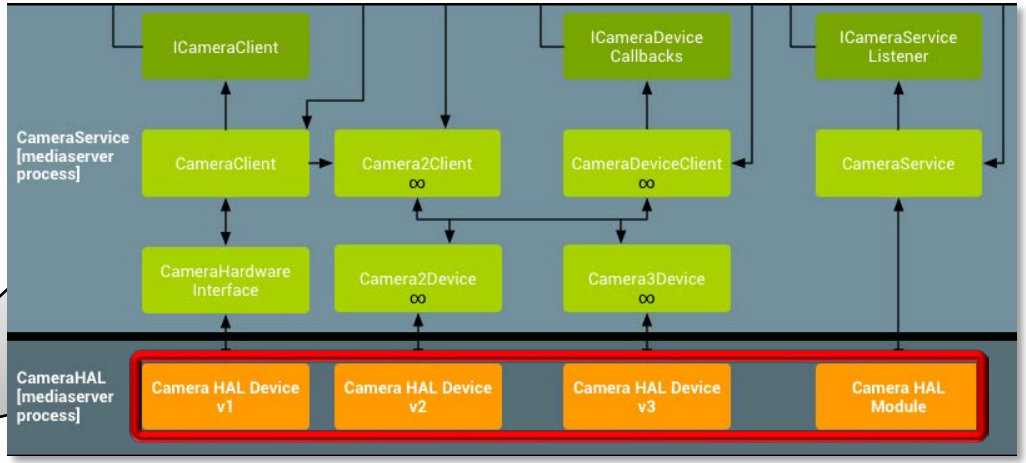
- e.g., camera hardware from an OEM has a camera system service & a camera HAL API

See source.android.com/devices/camera

The Android Hardware Abstraction Layer (HAL)

- Driver modules that implement HAL APIs are linked dynamically by Android

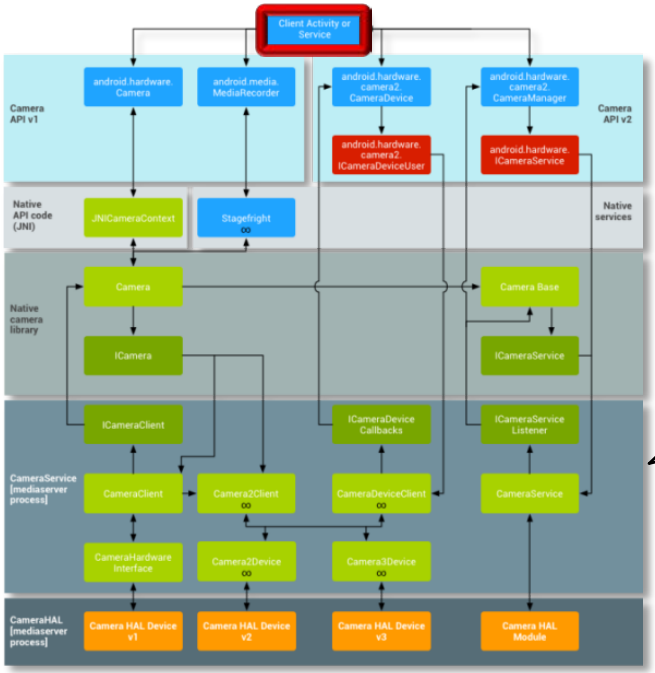


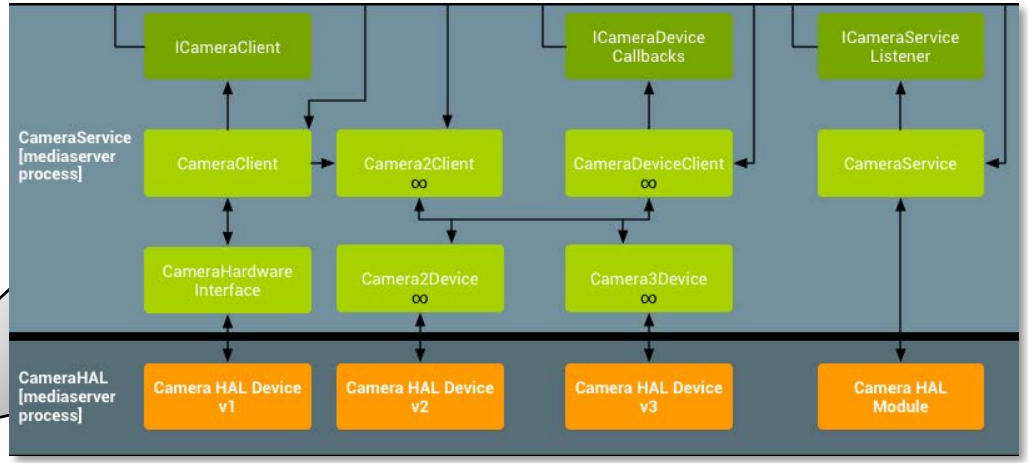


See source.android.com/devices/#modules

The Android Hardware Abstraction Layer (HAL)

- Driver modules that implement HAL APIs are linked dynamically by Android





- e.g., when an app first accesses the camera hardware encapsulated by the camera HAL API

See source.android.com/devices/#modules

Motivations for Android's Hardware Abstraction Layer

The Android Hardware Abstraction Layer (HAL)

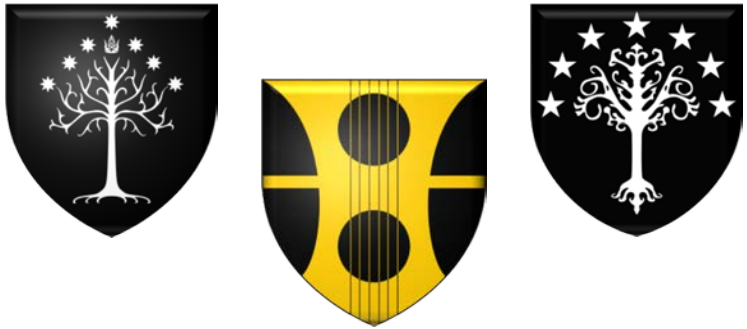
- There are two types of motivations for Android's user space HAL design

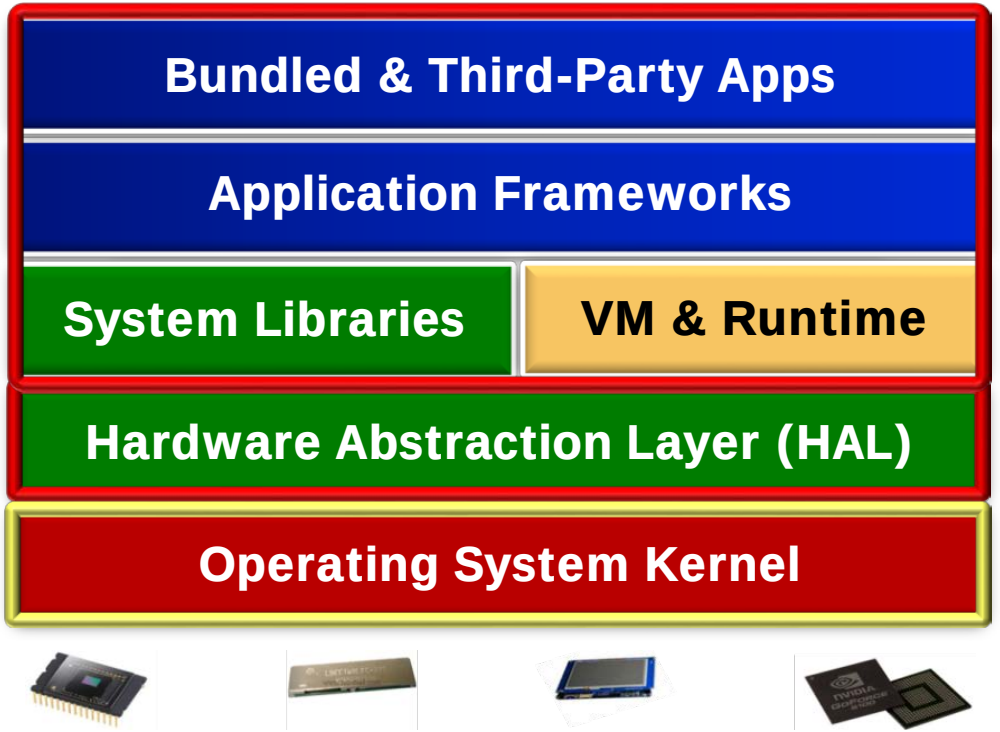
The Android Hardware Abstraction Layer (HAL)

- One motivation is technical

The Android Hardware Abstraction Layer (HAL)

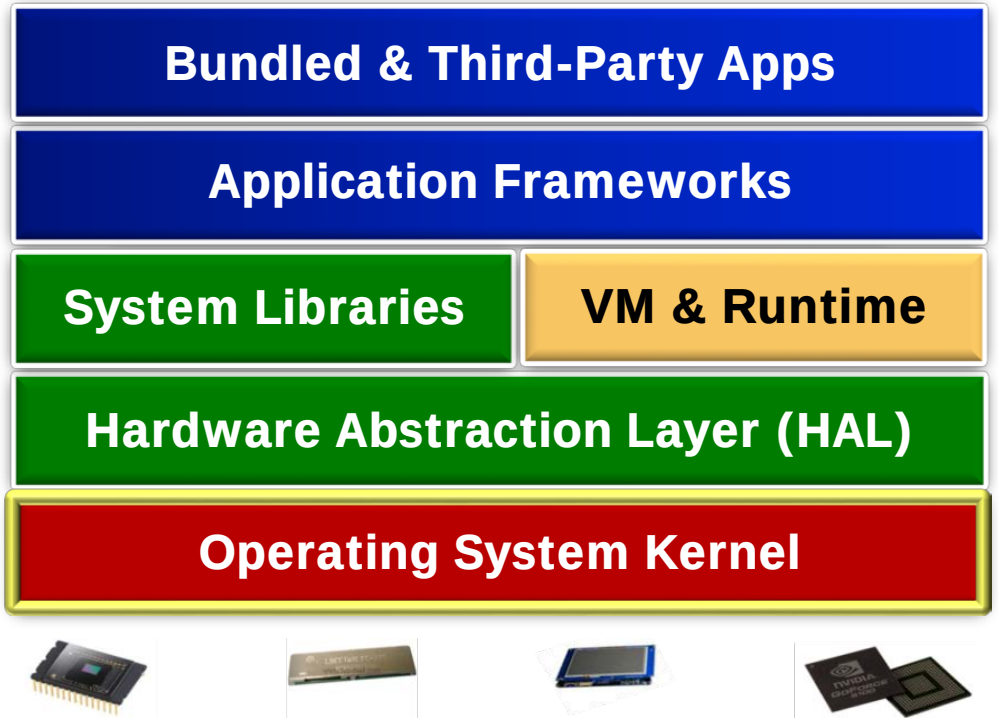
- One motivation is technical
  - Shielding higher layers of Android's software stack from Linux kernel idiosyncrasies





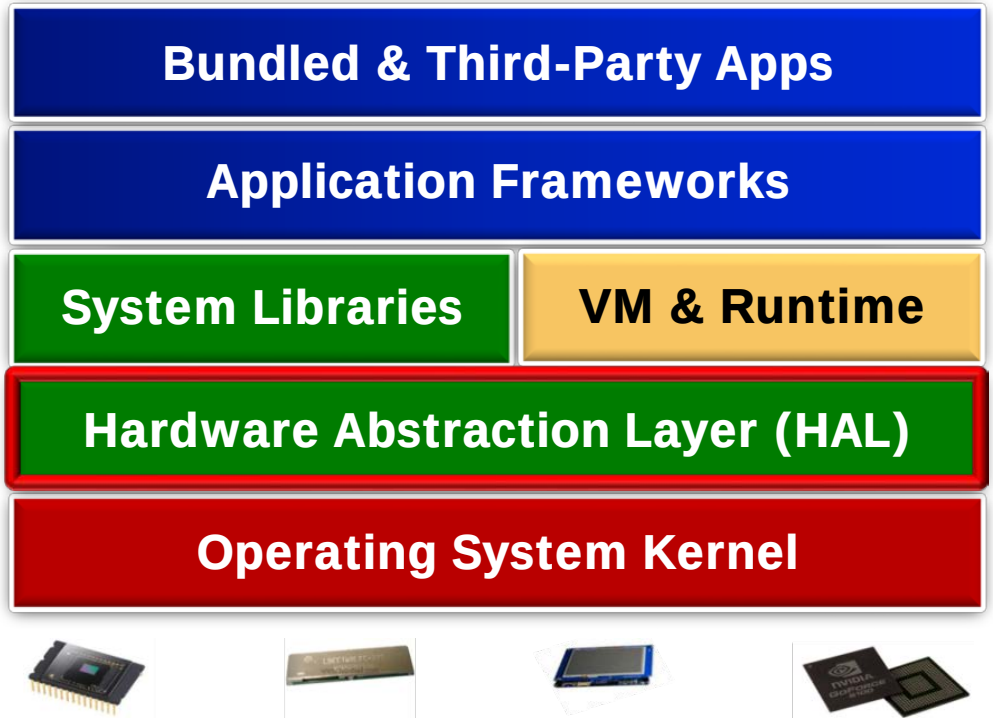
The Android Hardware Abstraction Layer (HAL)

- One motivation is technical
  - Shielding higher layers of Android's software stack from Linux kernel idiosyncrasies
    - GNU Linux device drivers don't provide consistent support for certain types of devices
      - It's historical focus is on laptops/desktops/servers



The Android Hardware Abstraction Layer (HAL)

- One motivation is technical
  - Shielding higher layers of Android's software stack from Linux kernel idiosyncrasies
    - GNU Linux device drivers don't provide consistent support for certain types of devices
    - The HAL therefore defines a consistent object-oriented API for these hardware elements



The Android Hardware Abstraction Layer (HAL)

- Another motivation is economic

The Android Hardware Abstraction Layer (HAL)

- Another motivation is economic
  - Kernel device drivers are “GPL’d”

See en.wikipedia.org/wiki/GNU_General_Public_License

The Android Hardware Abstraction Layer (HAL)

- Another motivation is economic
  - Kernel device drivers are “GPL’d”
    - The GPL requires OEMs give out source code for their drivers

See en.wikipedia.org/wiki/GNU_General_Public_License

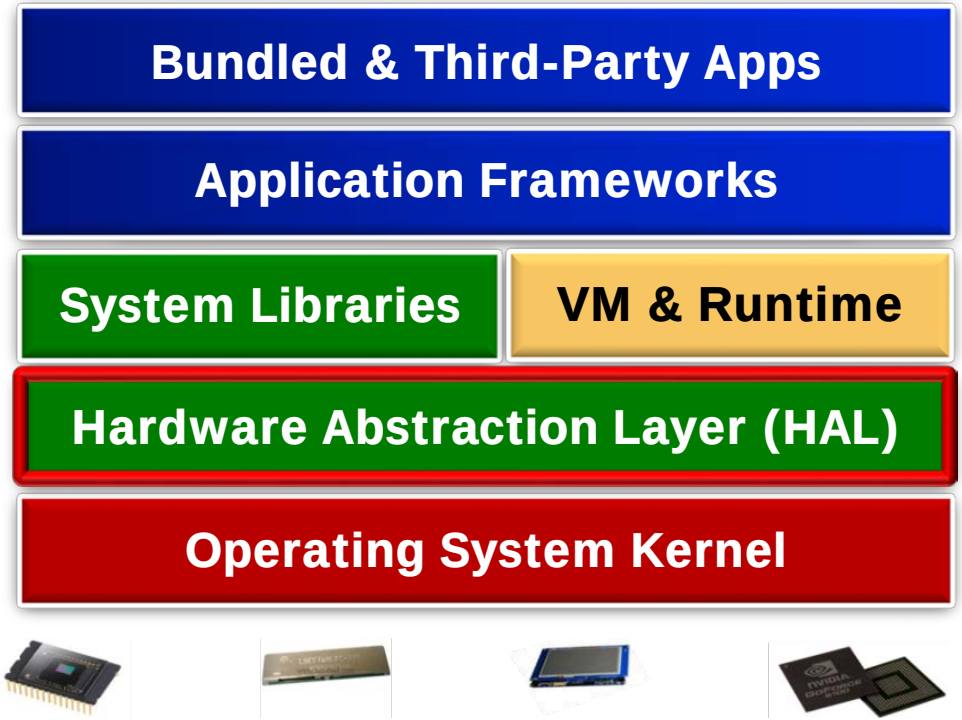
The Android Hardware Abstraction Layer (HAL)

- Another motivation is economic
  - Kernel device drivers are “GPL’d”
    - The GPL requires OEMs give out source code for their drivers
  - However, Android OEMs often use proprietary drivers
    - i.e., they don’t want to give out the source code for them

See higes.com/android-linux-kernel-drivers

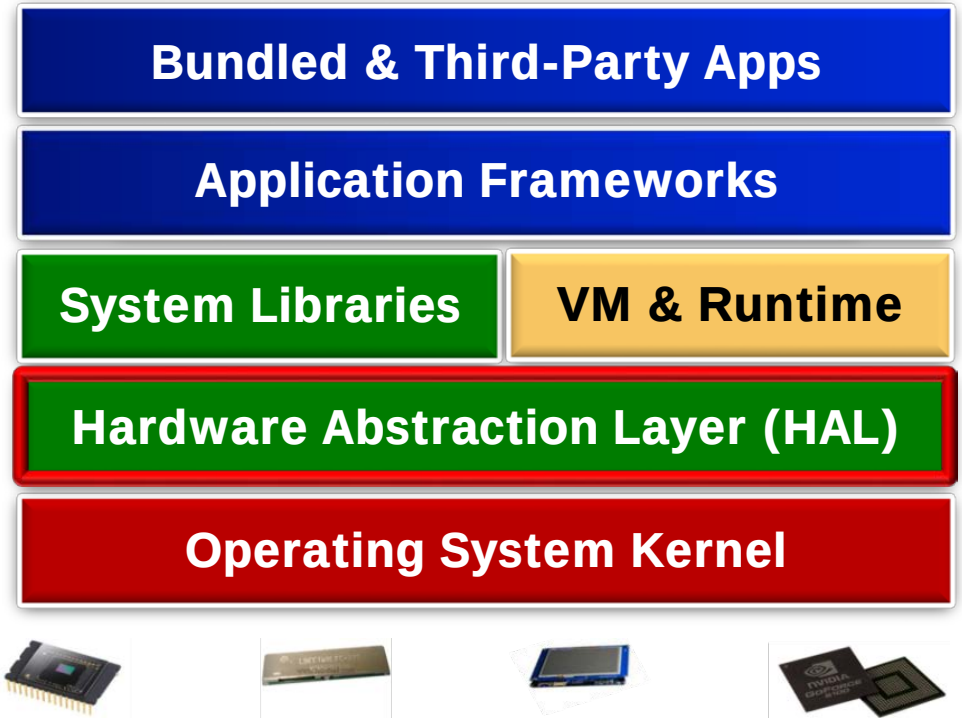
The Android Hardware Abstraction Layer (HAL)

- Another motivation is economic
  - Kernel device drivers are “GPL’d”
    - The GPL requires OEMs give out source code for their drivers
    - However, Android OEMs often use proprietary drivers
    - Putting driver implementations in user-space HAL enables them *not* to release the source code



The Android Hardware Abstraction Layer (HAL)

- Another motivation is economic
  - Kernel device drivers are “GPL’d”
    - The GPL requires OEMs give out source code for their drivers
    - However, Android OEMs often use proprietary drivers
    - Putting driver implementations in user-space HAL enables them *not* to release the source code
  - See Android licenses for details



See source.android.com/source/licenses.html

End of Infrastructure Middleware (Part 1): Hardware Abstraction Layer