

Architecture

Android system architecture contains the following components:

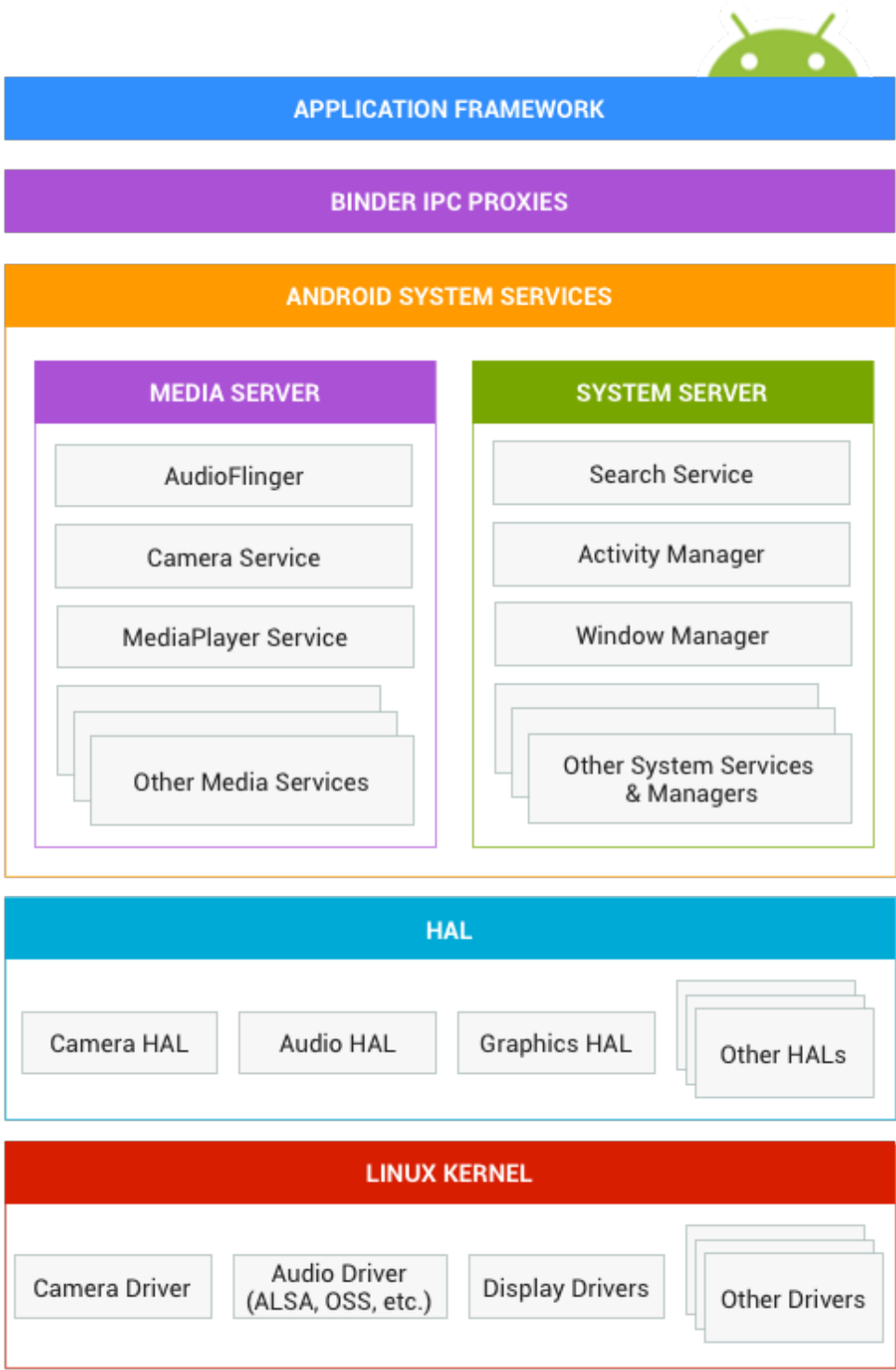


Figure 1. Android system architecture

Application framework

The application framework is used most often by application developers. As a hardware developer, you should be aware of developer APIs as many map directly to the underlying HAL interfaces and can provide helpful information about implementing drivers.

Binder IPC

The Binder Inter-Process Communication (IPC) mechanism allows the application framework to cross process boundaries and call into the Android system services code. This enables high level framework APIs to interact with Android system services. At the application framework level, this communication is hidden from the developer and things appear to "just work".

System services

System services are modular, focused components such as Window Manager, Search Service, or Notification Manager. Functionality exposed by application framework APIs communicates with system services to access the underlying hardware. Android includes two groups of services: *system* (such as Window Manager and Notification Manager) and *media* (services involved in playing and recording media).

Hardware abstraction layer (HAL)

A HAL defines a standard interface for hardware vendors to implement, which enables Android to be agnostic about lower-level driver implementations. Using a HAL allows you to implement functionality without affecting or modifying the higher level system. HAL implementations are packaged into modules and loaded by the Android system at the appropriate time. For details, see [Hardware Abstraction Layer \(HAL\)](https://source.android.com/devices/architecture/hal.html) (<https://source.android.com/devices/architecture/hal.html>).

Linux kernel

Developing your device drivers is similar to developing a typical Linux device driver. Android uses a version of the Linux kernel with a few special additions such as wake locks (a memory management system that is more aggressive in preserving memory), the Binder IPC driver, and other features important for a mobile embedded platform. These additions are primarily for system functionality and do not affect driver development.

You can use any version of the kernel as long as it supports the required features (such as the binder driver). However, we recommend using the latest version of the Android kernel. For details, see [Building Kernels](https://source.android.com/source/building-kernels.html) (<https://source.android.com/source/building-kernels.html>).

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](http://creativecommons.org/licenses/by/3.0/) (<http://creativecommons.org/licenses/by/3.0/>), and code samples are licensed under the [Apache 2.0 License](http://www.apache.org/licenses/LICENSE-2.0) (<http://www.apache.org/licenses/LICENSE-2.0>). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (<https://developers.google.com/terms/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated July 5, 2017.