

平台架构

本文内容

> Linux 内核

> 硬件抽象层 (HAL)

> Android Runtime

> 原生 C/C++ 库

> Java API 框架

> 系统应用

Android 是一种基于 Linux 的开放源代码软件栈，为广泛的设备和机型而创建。下图所示为 Android 平台的主要组件。

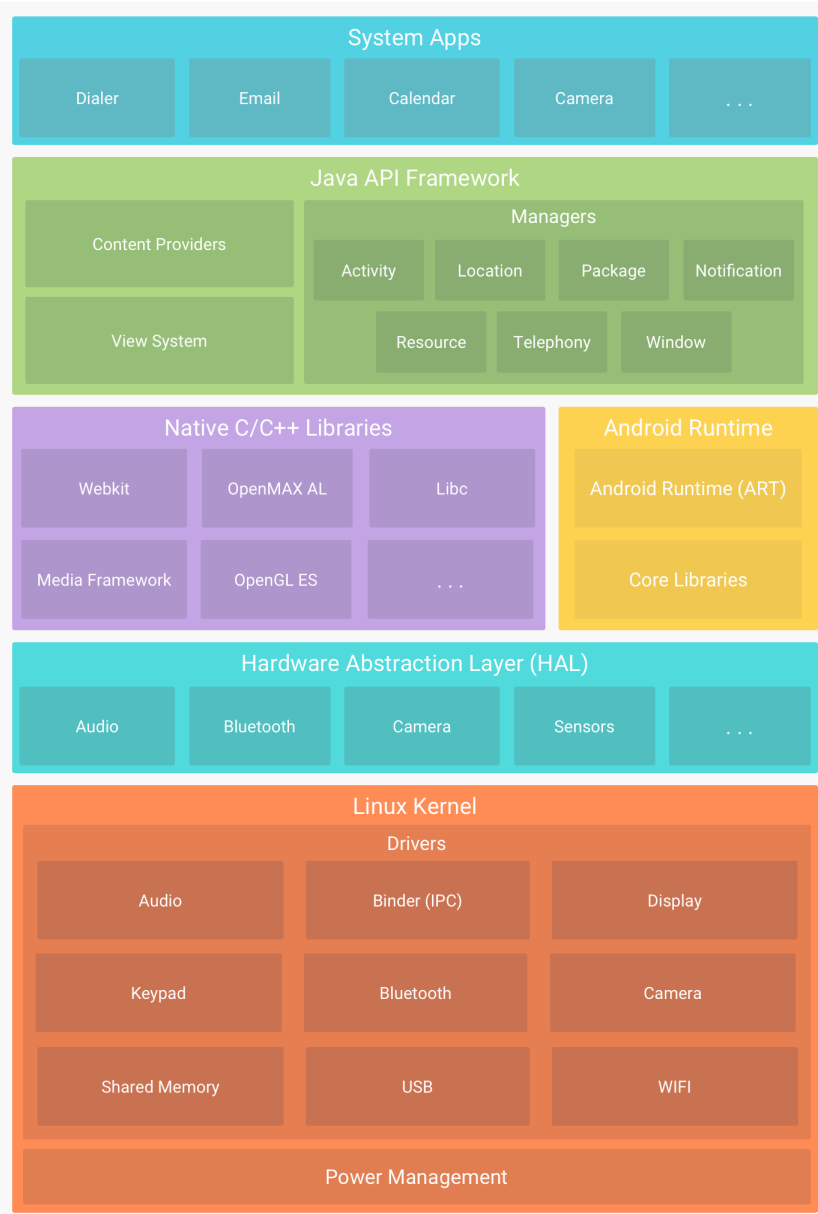


图 1. Android 软件栈。

Linux 内核

Android 平台的基础是 Linux 内核。例如，[Android Runtime \(ART\)](#) 依靠 Linux 内核来执行底层功能，例如线程和低层内存管理。

使用 Linux 内核可让 Android 利用[主要安全功能](#)，并且允许设备制造商为著名的内核开发硬件驱动程序。

硬件抽象层 (HAL)

[硬件抽象层 \(HAL\)](#) 提供标准界面，向更高级别的 [Java API 框架](#) 显示设备硬件功能。HAL 包含多个库模块，其中每个模块都为特定类型的硬件组件实现一个界面，例如[相机](#)或[蓝牙](#)模块。当框架 API 要求访问设备硬件时，Android 系统将为该硬件组件加载库模块。

Android Runtime

对于运行 Android 5.0（API 级别 21）或更高版本的设备，每个应用都在其自己的进程中运行，并且有其自己的 [Android Runtime \(ART\)](#) 实例。ART 编写为通过执行 DEX 文件在低内存设备上运行多个虚拟机，DEX 文件是一种专为 Android 设计的字节码格式，经过优化，使用的内存很少。编译工具链（例如 [Jack](#)）将 Java 源代码编译为 DEX 字节码，使其可在 Android 平台上运行。

ART 的部分主要功能包括：

- 预先 (AOT) 和即时 (JIT) 编译
- 优化的垃圾回收 (GC)
- 更好的调试支持，包括专用采样分析器、详细的诊断异常和崩溃报告，并且能够设置监视点以监控特定字段

在 Android 版本 5.0（API 级别 21）之前，Dalvik 是 Android Runtime。如果您的应用在 ART 上运行效果很好，那么它应该也可在 Dalvik 上运行，但[反过来不一定](#)。

Android 还包含一套核心运行时库，可提供 Java API 框架使用的 Java 编程语言大部分功能，包括一些 [Java 8 语言功能](#)。

原生 C/C++ 库

许多核心 Android 系统组件和服务（例如 ART 和 HAL）构建自原生代码，需要以 C 和 C++ 编写的原生库。Android 平台提供 Java 框架 API 以向应用显示其中部分原生库的功能。例如，您可以通过 Android 框架的 [Java OpenGL API](#) 访问 [OpenGL ES](#)，以支持在应用中绘制和操作 2D 和 3D 图形。

如果开发的是需要 C 或 C++ 代码的应用，可以使用 [Android NDK](#) 直接从原生代码访问某些[原生平台库](#)。

Java API 框架

您可通过以 Java 语言编写的 API 使用 Android OS 的整个功能集。这些 API 形成创建 Android 应用所需的构建块，它们可简化核心模块化系统组件和服务的重复使用，包括以下组件和服务：

- 丰富、可扩展的[视图系统](#)，可用于构建应用的 UI，包括列表、网格、文本框、按钮甚至可嵌入的网络浏览器
- [资源管理器](#)，用于访问非代码资源，例如本地化的字符串、图形和布局文件
- [通知管理器](#)，可让所有应用在状态栏中显示自定义提醒
- [Activity 管理器](#)，用于管理应用的生命周期，提供常见的[导航返回栈](#)
- [内容提供程序](#)，可让应用访问其他应用（例如“联系人”应用）中的数据或者共享其自己的数据

开发者可以完全访问 Android 系统应用使用的[框架 API](#)。

系统应用

Android 随附一套用于电子邮件、短信、日历、互联网浏览和联系人等的核心应用。平台随附的应用与用户可以选择安装的应用一样，没有特殊状态。因此第三方应用可成为用户的默认网络浏览器、短信 Messenger 甚至默认键盘（有一些例外，例如系统的“设置”应用）。

系统应用可用作用户的应用，以及提供开发者可从其自己的应用访问的主要功能。例如，如果您的应用要发短信，您无需自己构建该功能，可以改为调用已安装的短信应用向您指定的接收者发送消息。