

# 系统定义

## 系统定位

HarmonyOS 是一款“面向未来”、面向全场景（移动办公、运动健康、社交通信、媒体娱乐等）的分布式操作系统。在传统的单设备系统能力的基础上，HarmonyOS 提出了基于同一套系统能力、适配多种终端形态的分布式理念，能够支持手机、平板、智能穿戴、智慧屏、车机等多种终端设备。

- 对消费者而言，HarmonyOS 能够将生活场景中的各类终端进行能力整合，可以实现不同的终端设备之间的快速连接、能力互助、资源共享，匹配合适的设备、提供流畅的全场景体验。
- 对应用开发者而言，HarmonyOS 采用了多种分布式技术，使得应用程序的开发实现与不同终端设备的形态差异无关。这能够让开发者聚焦上层业务逻辑，更加便捷、高效地开发应用。
- 对设备开发者而言，HarmonyOS 采用了组件化的设计方案，可以根据设备的资源能力和业务特征进行灵活裁剪，满足不同形态的终端设备对于操作系统的要求。

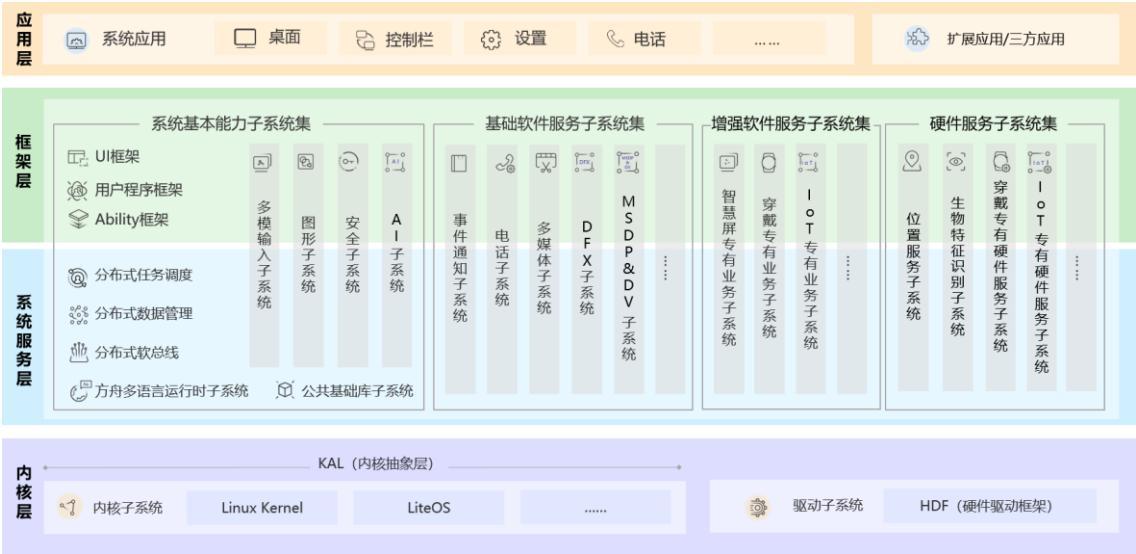
HarmonyOS 提供了支持多种开发语言的 API，供开发者进行应用开发。支持的开发语言包括 Java、XML (Extensible Markup Language)、C/C++、JS (JavaScript)、CSS (Cascading Style Sheets) 和 HML (HarmonyOS Markup Language)。

来源：HarmonyOS 官网

<https://developer.harmonyos.com/cn/docs/documentation/doc-guides/harmonyos-overview-000000000001190>

# 技术架构

HarmonyOS 整体遵从分层设计，从下向上依次为：内核层、系统服务层、框架层和应用层。系统功能按照“系统 > 子系统 > 功能/模块”逐级展开，在多设备部署场景下，支持根据实际需求裁剪某些非必要的子系统或功能/模块。HarmonyOS 技术架构如下所示。



## 内核层

内核子系统：HarmonyOS 采用多内核设计，支持针对不同资源受限设备选用适合的 OS 内核。内核抽象层（KAL，Kernel Abstract Layer）通过屏蔽多内核差异，对上层提供基础的内核能力，包括进程/线程管理、内存管理、文件系统、网络管理和外设管理等。

驱动子系统：硬件驱动框架（HDF）是 HarmonyOS 硬件生态开放的基础，提供统一外设访问能力和驱动开发、管理框架。

## 系统服务层

系统服务层是 HarmonyOS 的核心能力集合，通过框架层对应用程序提供服务。该层包含以下几个部分：

- **系统基本能力子系统集：**为分布式应用在 HarmonyOS 多设备上的运行、调度、迁移等操作提供了基础能力，由分布式软总线、分布式数据管理、分布式任务调度、方舟多语言运行时、公共基础库、多模输入、图形、安全、AI 等子系统组成。其中，方舟运行时提供了 C/C++/JS 多语言运行时和基础的系统类库，也为使用方舟编译器静态化的 Java 程序（即应用程序或框架层中使用 Java 语言开发的部分）提供运行时。
- **基础软件服务子系统集：**为 HarmonyOS 提供公共的、通用的软件服务，由事件通知、电话、多媒体、DFX（Design For X）、MSDP&DV 等子系统组成。
- **增强软件服务子系统集：**为 HarmonyOS 提供针对不同设备的、差异化的能力增强型软件服务，由智慧屏专有业务、穿戴专有业务、IoT 专有业务等子系统组成。
- **硬件服务子系统集：**为 HarmonyOS 提供硬件服务，由位置服务、生物特征识别、穿戴专有硬件服务、IoT 专有硬件服务等子系统组成。

根据不同设备形态的部署环境，基础软件服务子系统集、增强软件服务子系统集、硬件服务子系统集内部可以按子系统粒度裁剪，每个子系统内部又可以按功能粒度裁剪。

## 框架层

框架层为 HarmonyOS 应用开发提供了 Java/C/C++/JS 等多语言的用户程序框架和

Ability 框架，两种 UI 框架（包括适用于 Java 语言的 Java UI 框架、适用于 JS 语言的 JS UI 框架），以及各种软硬件服务对外开放的多语言框架 API。根据系统的组件化裁剪程度，HarmonyOS 设备支持的 API 也会有所不同。

## 应用层

应用层包括系统应用和第三方非系统应用。HarmonyOS 的应用由一个或多个 FA (Feature Ability) 或 PA (Particle Ability) 组成。其中，FA 有 UI 界面，提供与用户交互的能力；而 PA 无 UI 界面，提供后台运行任务的能力以及统一的数据访问抽象。FA 在进行用户交互时所需的后台数据访问也需要由对应的 PA 提供支撑。基于 FA/PA 开发的应用，能够实现特定的业务功能，支持跨设备调度与分发，为用户提供一致、高效的应用体验。