



## Android application frameworks overview



---

## Contents

---

1. Android application frameworks overview .....	3
2. STM32MP15 Linux kernel overview .....	4

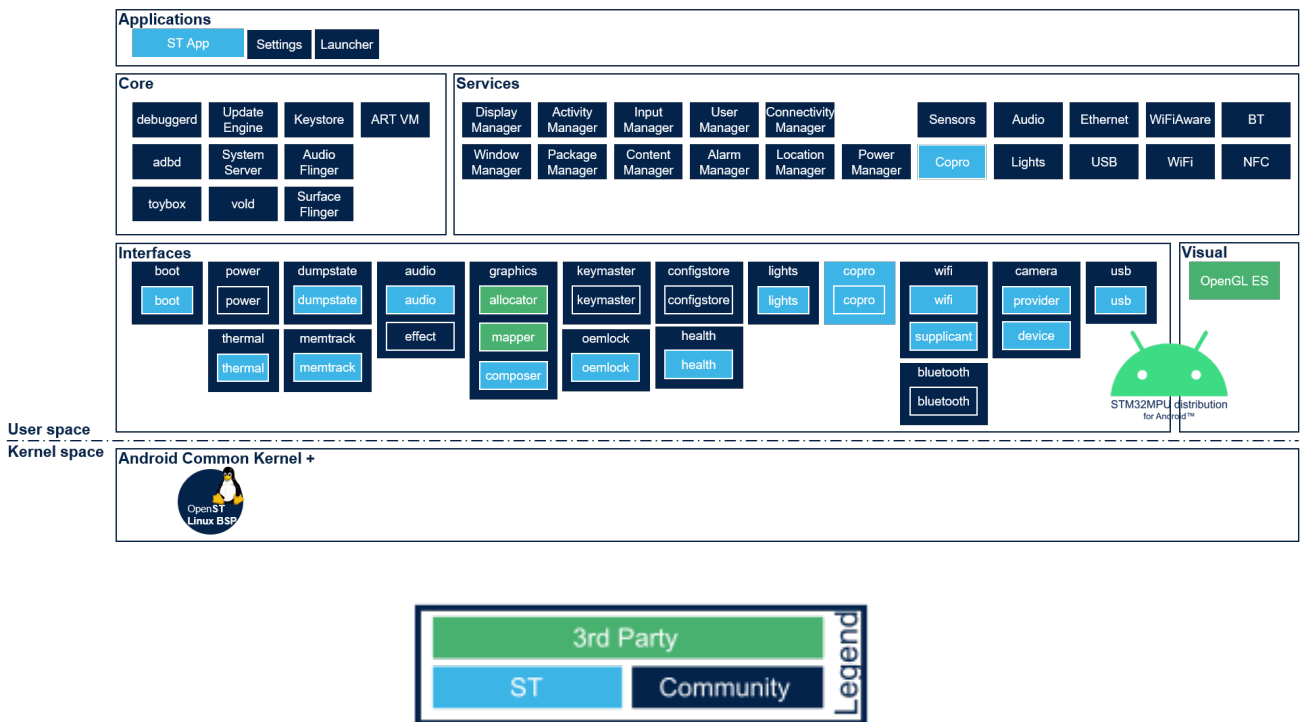


A quality version of this page, approved on 9 October 2019, was based off this revision.

The diagram below gives an overview of the Android application frameworks (Linux user space components) that rely on Linux kernel.

This Linux kernel is based on the Android Common Kernel<sup>[1]</sup> following Google recommendation for its configuration<sup>[2]</sup>.

It shows the main components grouped per functional domain, however it is not exhaustive.



The Android framework is structured in several layers:

- hardware interfaces providing a standard way to configure the underlying driver, based on the Android HIDL (Hardware Interface Definition Language) mechanism
- the Android core (ART virtual machine, useful daemons including vold, adbd, and debuggerd, and system services)
- Android services (providing interfaces to the application = SDK)
- Android applications (including the launcher)

The Android services are often implemented partly in native cpp, and partly in Java. The JNI (Java Native Interface) IPC mechanism is available to allow communication between the two worlds.

In addition to the standard Android services, a proprietary coprocessor service for Android has been introduced (for development purposes only). The CoproService is composed of two parts:

- firmware management (check running firmware, start/stop firmware, get/set the firmware name)
- TTY management (open/close and read/write the TTY interface). You must implement your own protocol on top of this.



## References

- AOSP: <https://android.googlesource.com/kernel/common/>
- AOSP: <https://android.googlesource.com/kernel/configs/>

Linux® is a registered trademark of Linus Torvalds.

HAL interface definition language (see <https://source.android.com/devices/architecture/hidl>)

Android Runtime (see <https://source.android.com/devices/tech/dalvik>)

Software development kit (A programming package that enables a programmer to develop applications for a specific platform.)

Java Native Interface (for Android)

Inter-Processor Communication

TeleTYpewriter

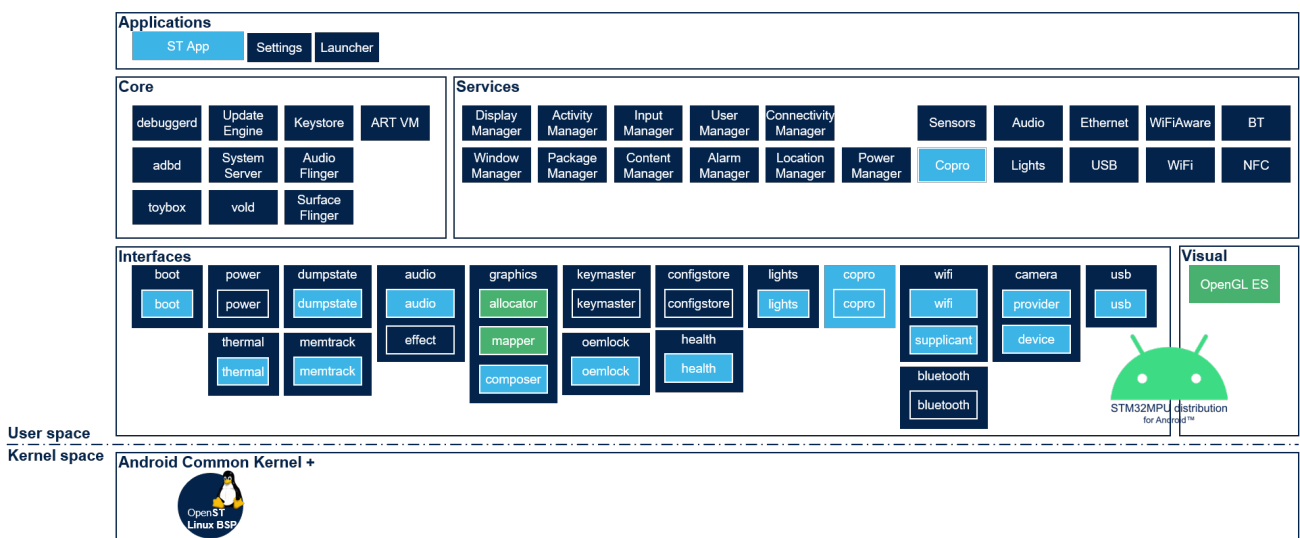
Android Open Source Project

Stable: 05.11.2021 - 08:48 / Revision: 21.10.2021 - 09:59

The diagram below gives an overview of the Android application frameworks (Linux user space components) that rely on Linux kernel.

This Linux kernel is based on the Android Common Kernel<sup>[1]</sup> following Google recommendation for its configuration<sup>[2]</sup>.

It shows the main components grouped per functional domain, however it is not exhaustive.





The Android framework is structured in several layers:

- hardware interfaces providing a standard way to configure the underlying driver, based on the Android HIDL (Hardware Interface Definition Language) mechanism
- the Android core (ART virtual machine, useful daemons including vold, adbd, and debuggerd, and system services)
- Android services (providing interfaces to the application = SDK)
- Android applications (including the launcher)

The Android services are often implemented partly in native cpp, and partly in Java. The JNI (Java Native Interface) IPC mechanism is available to allow communication between the two worlds.

In addition to the standard Android services, a proprietary coprocessor service for Android has been introduced (for development purposes only). The CoproService is composed of two parts:

- firmware management (check running firmware, start/stop firmware, get/set the firmware name)
- TTY management (open/close and read/write the TTY interface). You must implement your own protocol on top of this.



---

## References

---

- AOSP: <https://android.googlesource.com/kernel/common/>
- AOSP: <https://android.googlesource.com/kernel/configs/>

Linux<sup>®</sup> is a registered trademark of Linus Torvalds.

HAL interface definition language (see <https://source.android.com/devices/architecture/hidl>)

Android Runtime (see <https://source.android.com/devices/tech/dalvik>)

Software development kit (A programming package that enables a programmer to develop applications for a specific platform.)

Java Native Interface (for Android)

Inter-Processor Communication

TeleTYpewriter

Android Open Source Project