Camera HAL v3 overview

Android's camera Hardware Abstraction Layer (HAL) connects the higher level camera framework APIs in android.hardware.Camera (http://developer.android.com/reference/android/hardware/Camera.html) to your underlying camera driver and hardware. The latest version of Android introduces a new, underlying implementation of the camera stack. If you have previously developed a camera HAL module and driver for other versions of Android, be aware that there are significant changes in the camera pipeline.

IN THIS DOCUMENT

Version 1 of the camera HAL is still supported for future releases of Android because many devices still rely on it. Implementing both HALs is also supported by the Android camera service, which is useful when you want to support a less capable front-facing camera with version 1 of the HAL and a more advanced back-facing camera with version 3 of the HAL. Version 2 was a stepping stone to version 3 and is not supported.

There is only one camera HAL module (with its own version number, currently 1, 2, or 2.1), which lists multiple independent camera devices that each have their own version. Camera module v2 or newer is required to support devices v2 or newer, and such camera modules can have a mix of camera device versions. This is what we mean when we say Android supports implementing both HALs.

> **Note:** The new camera HAL is in active development and can change at any time. This document describes at a high level the design of the camera subsystem and omits many details. See Camera version support (versioning.html) for our plans.

Overview

Version 1 of the camera subsystem was designed as a black box with high-level controls. Roughly speaking, the old subsystem has three operating modes:

- Preview
- Video Record
- Still Capture

Each mode has slightly different and overlapping capabilities. This made it hard to implement new types of features, such as burst mode, since it would fall between two of these modes.

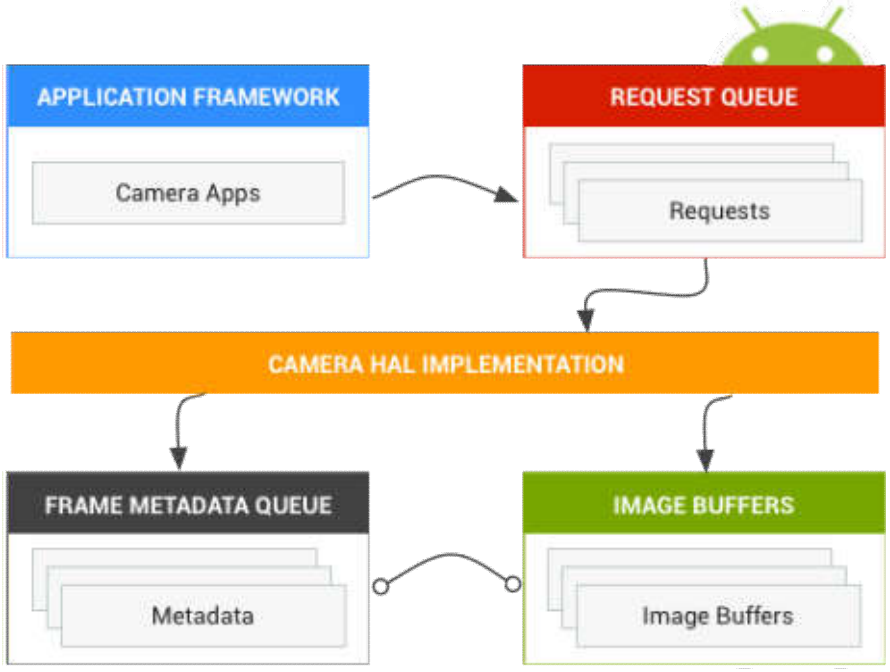

Figure 1. Camera components

Version 3 enhancements

The aim of the Android Camera API redesign is to substantially increase the ability of applications to control the camera subsystem on Android devices while reorganizing the API to make it more efficient and maintainable.

The additional control makes it easier to build high-quality camera applications on Android devices that can operate reliably across multiple products while still using device-specific algorithms whenever possible to maximize quality and performance.

Version 3 of the camera subsystem structures the operation modes into a single unified view, which can be used to implement any of the previous modes and several others, such as burst mode. This results in better user control for focus and exposure and more post-processing, such as noise reduction, contrast and sharpening. Further, this simplified view makes it easier for application developers to use the camera's various functions. The API models the camera subsystem as a pipeline that converts incoming requests for frame captures into frames, on a 1:1 basis. The requests encapsulate all configuration information about the capture and processing of a frame. This includes: resolution and pixel format; manual sensor, lens and flash control; 3A operating modes; RAW->YUV processing control; statistics generation; and so on.

In simple terms, the application framework requests a frame from the camera subsystem, and the camera subsystem returns results to an output stream. In addition, metadata that contains information such as color spaces and lens shading is generated for each set of results. The following sections and diagrams give you more detail about each component.
You can think of camera version 3 as a pipeline to camera version 1's one-way stream. It converts each capture request into one image captured by the sensor, which is processed into:

- A Result object with metadata about the capture.
- One to N buffers of image data, each into its own destination Surface.

The set of possible output Surfaces is preconfigured:

- Each Surface is a destination for a stream of image buffers of a fixed resolution.
- Only a small number of Surfaces can be configured as outputs at once (~3).

A request contains all desired capture settings and the list of output Surfaces to push image buffers into for this request (out of the total configured set). A request can be one-shot (with capture()), or it may be repeated indefinitely (with setRepeatingRequest()). Captures have priority over repeating requests.

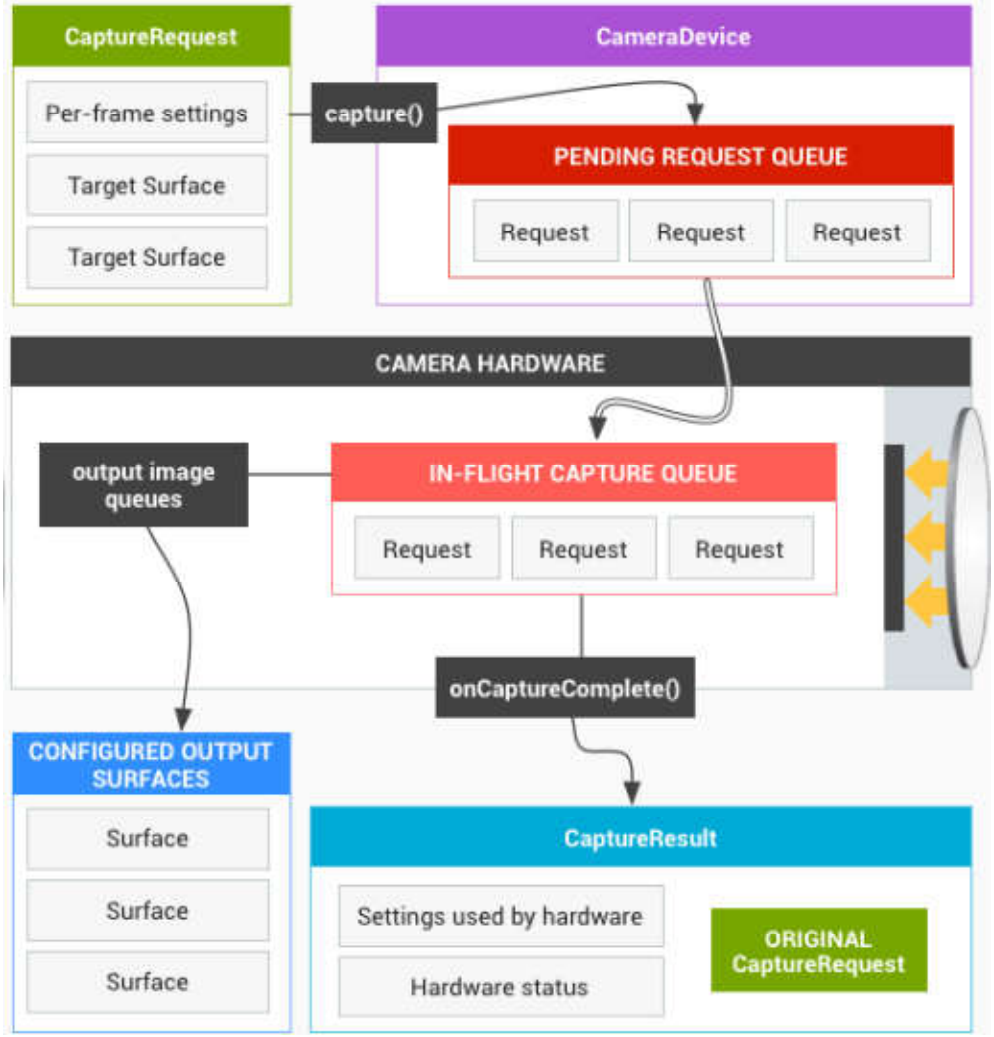

Figure 2. Camera core operation model

Supported version

Camera devices that support this version of the HAL must return CAMERA_DEVICE_API_VERSION_3_1 in camera_device_t.common.version and in camera_info_t.device_version (from camera_module_t.get_camera_info).
Camera modules that may contain version 3.1 devices must implement at least version 2.0 of the camera module interface (as defined by camera_module_t.common.module_api_version).
See camera_common.h for more versioning details.