libcamera: Unifying camera support on all Linux systems

LPC 2019
Lisbon, Portugal

Laurent Pinchart
laurent.pinchart@ideasonboard.com

libcamera

Cameras are complex devices that need heavy hardware image processing operations. Control of the processing is based on advanced algorithms that must run on a programmable processor. This has traditionally been implemented in a dedicated MCU in the camera, but in embedded devices algorithms have been moved to the main CPU to save cost. Blurring the boundary between camera devices and Linux often left the user with no other option than a vendor-specific closed-source solution.

To address this problem the Linux media community has very recently started collaboration with the industry to develop a camera stack that will be open-source-friendly while still protecting vendor core IP. libcamera was born out of that collaboration and will offer modern camera support to Linux-based systems, including traditional Linux distributions, ChromeOS and Android.

```
c s  /  +-----------------+
a e  |  |                 |
m n  |  |-----------------|
e s  |  |      ov5640     |
r o  |  |-----------------|
a r  |  |        0        |
     \  +-----------------+
                 |
                 v
     /  +-----------------+
     |  |        0        |
     |  |-----------------|
S i  |  |      CSI-2      |
o n  |  |-----------------|
C t  |  |        1        |
  e  |  +-----------------+
c r  |           |
a f  |           v
m a  |  +-----------------+
e c  |  |        0        |
r e  |  |-----------------|
a    |  |     scaler      |
     |  |-----------------|
     |  |        1        |
     \  +-----------------+
                 |
                 v
A    /  +-----------------+
P    |  |     capture     |
I    |  |  /dev/video0    |
     \  +-----------------+
```

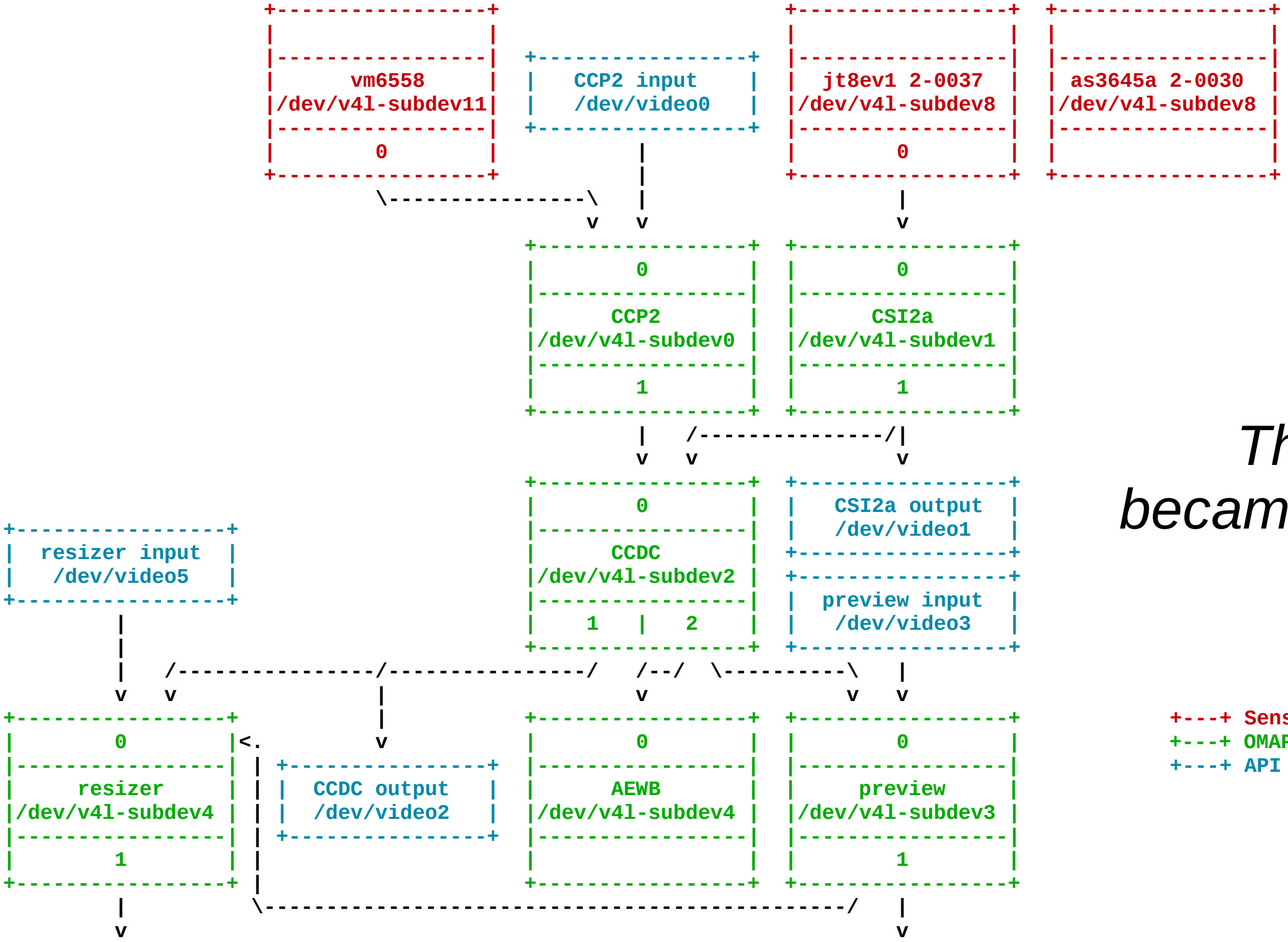
In the beginning were simple pipelines...

Why?

… and they were simple to control, with a single API.

```
+---------------+
|               |
|    Camera     |
|  Application  |
|               |
+---------------+
  ^
  |
V |
4 |
L |   +----+
2 |   |    |
  |   +----+
  |   +----+
  |   |    |
  |   +----+
  |   +----+
  \-->|    |
      +----+
```

Why?



```
                     +-----------------+                       +-----------------+  +-----------------+
                     |                 |                       |                 |  |                 |
                     |-----------------|  +-----------------+  |-----------------|  |-----------------|
                     |      vm6558     |  |   CCP2 input    |  |  jt8ev1 2-0037  |  | as3645a 2-0030  |
                     |/dev/v4l-subdev11|  |   /dev/video0   |  |/dev/v4l-subdev8 |  |/dev/v4l-subdev8 |
                     |-----------------|  +-----------------+  |-----------------|  |-----------------|
                     |        0        |           |           |        0        |  |                 |
                     +-----------------+           |           +-----------------+  +-----------------+
                              \----------------\   |                    |
                                               v   v                    v
                                          +-----------------+  +-----------------+
                                          |        0        |  |        0        |
                                          |-----------------|  |-----------------|
                                          |      CCP2       |  |      CSI2a      |
                                          |/dev/v4l-subdev0 |  |/dev/v4l-subdev1 |
                                          |-----------------|  |-----------------|
                                          |        1        |  |        1        |
                                          +-----------------+  +-----------------+
                                                   |   /---------------/|
                                                   v   v                v
                                          +-----------------+  +-----------------+
                                          |        0        |  |   CSI2a output  |
+-----------------+                       |-----------------|  |   /dev/video1   |
|  resizer input  |                       |      CCDC       |  +-----------------+
|   /dev/video5   |                       |/dev/v4l-subdev2 |  +-----------------+
+-----------------+                       |-----------------|  |  preview input  |
         |                                |    1   |   2    |  |   /dev/video3   |
         |                                +-----------------+  +-----------------+
         |   /----------------/----------------/   /--/  \----------\   |
         v   v                |                    v                v   v
+-----------------+           |           +-----------------+  +-----------------+
|        0        |<.         v           |        0        |  |        0        |
|-----------------| | +----------------+  |-----------------|  |-----------------|
|     resizer     | | |  CCDC output   |  |      AEWB       |  |     preview     |
|/dev/v4l-subdev4 | | |  /dev/video2   |  |/dev/v4l-subdev4 |  |/dev/v4l-subdev3 |
|-----------------| | +----------------+  |-----------------|  |-----------------|
|        1        | |                     |                 |  |        1        |
+-----------------+ |                     +-----------------+  +-----------------+
         |          \-----------------------------------------------/   |
         v                                                              v
```

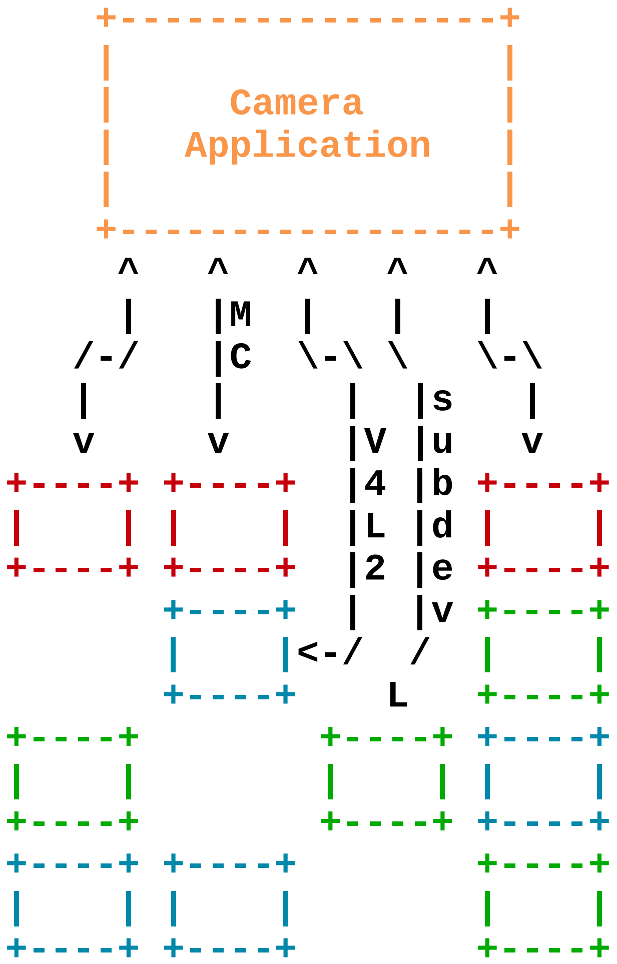
```
+---+ Sensors & flash
+---+ OMAP3 ISP
+---+ API
```

Then the world became complex ...

Why?

Why?

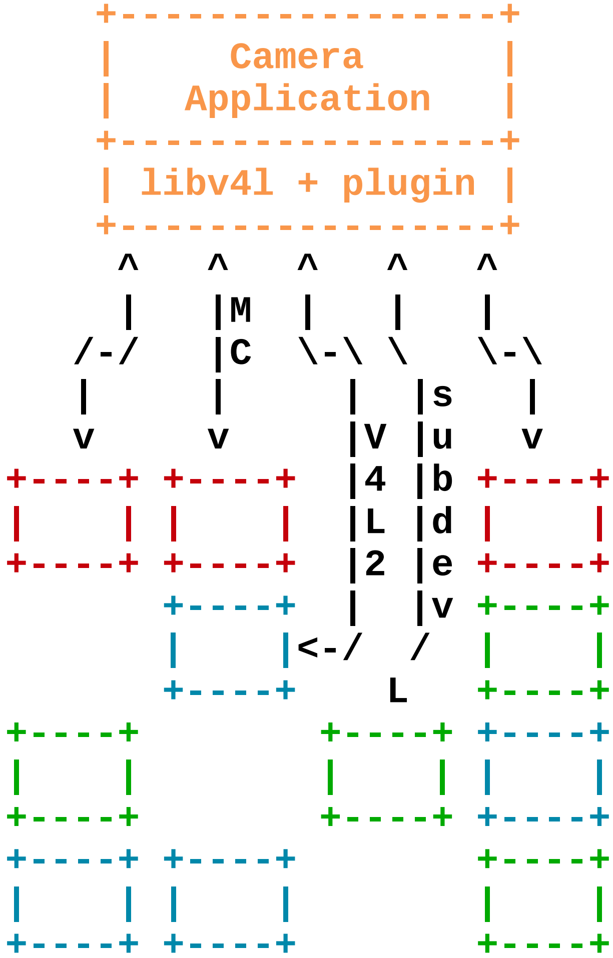
… and application developers were left suffering.



```
    +-----------------+
    |                 |
    |     Camera      |
    |   Application   |
    |                 |
    +-----------------+
     ^   ^   ^   ^   ^
     |   |M  |   |   |
   /-/   |C  \-\ \   \-\
   |     |     |  |s   |
   v     v     |V |u   v
+----+ +----+  |4 |b +----+
|    | |    |  |L |d |    |
+----+ +----+  |2 |e +----+
       +----+  |  |v +----+
       |    |<-/  /  |    |
       +----+    L   +----+
+----+        +----+ +----+
|    |        |    | |    |
+----+        +----+ +----+
+----+ +----+        +----+
|    | |    |        |    |
+----+ +----+        +----+
```

Why?

Solutions were proposed...



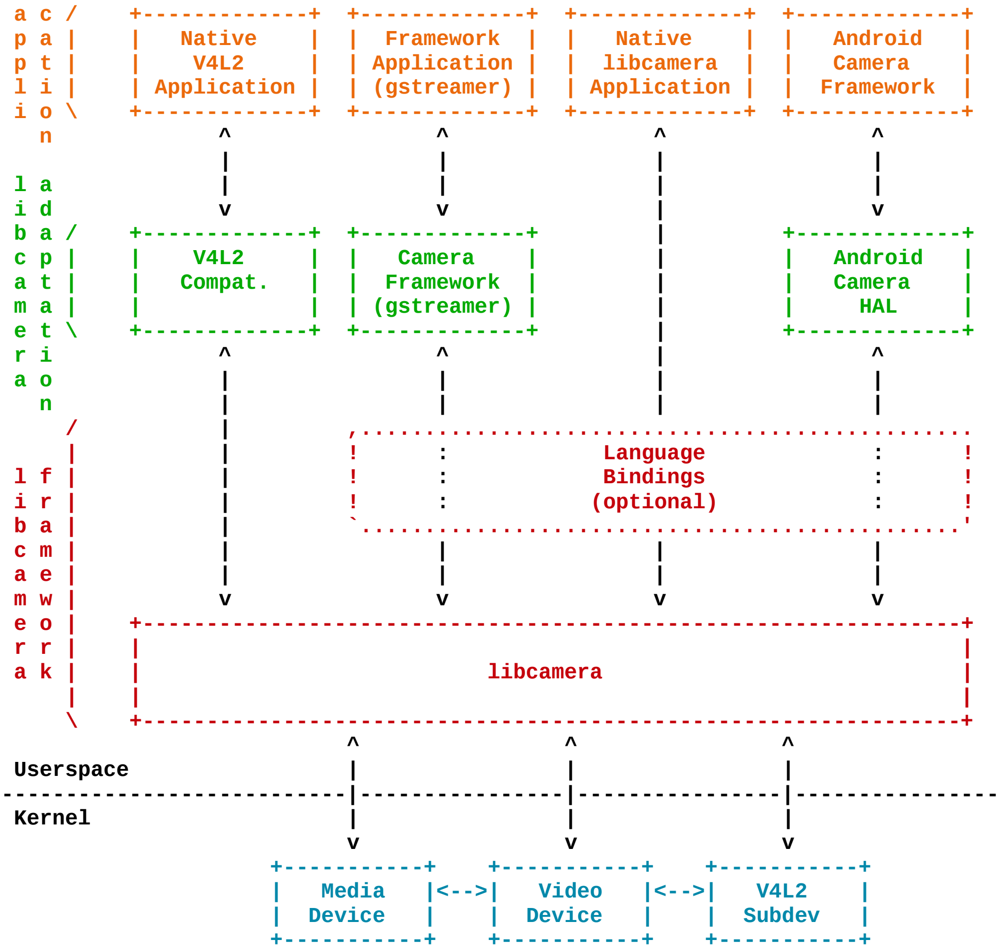
… but never implemented.

Then hope came back.

```
   _
+-/ \-+
| (o) |
+-----+
```

libcamera

libcamera provides a complete userspace camera stack.



```
 a c /    +-------------+  +-------------+  +-------------+  +-------------+
 p a |    |   Native    |  |  Framework  |  |   Native    |  |   Android   |
 p t |    |    V4L2     |  | Application |  |  libcamera  |  |   Camera    |
 l i |    | Application |  | (gstreamer) |  | Application |  |  Framework  |
 i o \    +-------------+  +-------------+  +-------------+  +-------------+
   n             ^                ^                ^                ^
                 |                |                |                |
 l a             |                |                |                |
 i d             v                v                |                v
 b a /    +-------------+  +-------------+         |         +-------------+
 c p |    |    V4L2     |  |   Camera    |         |         |   Android   |
 a t |    |   Compat.   |  |  Framework  |         |         |   Camera    |
 m a |    |             |  | (gstreamer) |         |         |     HAL     |
 e t \    +-------------+  +-------------+         |         +-------------+
 r i             ^                ^                |                ^
 a o             |                |                |                |
   n             |                |                |                |
     /           |         ,................................................
     |           |         !      :            Language             :      !
 l f |           |         !      :            Bindings             :      !
 i r |           |         !      :           (optional)            :      !
 b a |           |         `...............................................'
 c m |           |                |                |                |
 a e |           |                |                |                |
 m w |           v                v                v                v
 e o |    +----------------------------------------------------------------+
 r r |    |                                                                |
 a k |    |                           libcamera                            |
     |    |                                                                |
     \    +----------------------------------------------------------------+
                           ^                ^                ^
 Userspace                 |                |                |
---------------------------|----------------|----------------|----------------
 Kernel                    |                |                |
                           v                v                v
                     +-----------+    +-----------+    +-----------+
                     |   Media   |<-->|   Video   |<-->|   V4L2    |
                     |  Device   |    |  Device   |    |  Subdev   |
                     +-----------+    +-----------+    +-----------+
```

The ‘mesa’ of the camera world.

Camera Stack

```
       +------------------------+   +---------+
       |    _                   |   |    _    |
       | +-/ \-+                |   | +-/ \-+ |
       | | (o) |                |   | | (o) | |
       | +-----+                |   | +-----+ |
       +------------------------+   +---------+
            ^              ^             ^
 Userspace  |              |             |
------------|--------------|-------------|-------
 Kernel     |              |             |
            v              v             v
       +---------+    +---------+   +---------+
       |#########|<-->|#########|   |#########|
       |#########|    |#########|   |#########|
       +---------+    +---------+   +---------+
```

libcamera enumerates cameras...

Camera Devices & Enumeration

Streams

```
                     +-----+
   _         --->    | <>< |
+-/ \-+     /        +-----+
| (o) | ---/
+-----+    \      +----------+
            \      |   ,-,    |
             --->  |  ('_)<   |
                   |   `-`    |
                   +----------+
```

It supports multiple concurrent streams for the same camera...

… and per-frame controls.

```
   _        +----------+     +----------+
+-/ \-+     |     ,-,  |     |  ,-,     |
| (o) | --> |    ('_)< | --> | ('_)<    | --> ...
+-----+     |     `-`  |     |  `-`     |
            +----------+     +----------+
```

Per-Frame Controls

Image Processing Algorithm are loaded as external modules.

```
+---+      +-----+        +-----+
|(o)| ---> | ISP | --+--> | <>< | --->
+---+      +-----+   |    +-----+
              ^      |
              |      |   +--------+
              |      \-> | Stats  |
              |          +--------+
           +-----+            |
           | IPA | <----------/
           +-----+
```

Image Processing Algorithms (3A)

```
+-----------+
| V4L2 App. |
+-----------+

+-----------+
|           |
|   V4L2    |
|   API     |
|           |
|           |
|           |
|           |
+-----------+

+--------------------------------+
|           libcamera            |
```

Adaptation layers offer backward compatibility with existing APIs...

Adaptation

```
+-----------+        +-----------+
| V4L2 App. |        |  Android  |
+-----------+        +-----------+

+-----------+        +-----------+
|           |        |   \___/   |
|   V4L2    |        |   /. .\   |
|   API     |        |   -----   |
|           |        | !| Cam.|! |
|           |        | !| HAL |! |
|           |        |   -----   |
|           |        |    | |    |
+-----------+        +-----------+

+--------------------------------+
|           libcamera            |
```

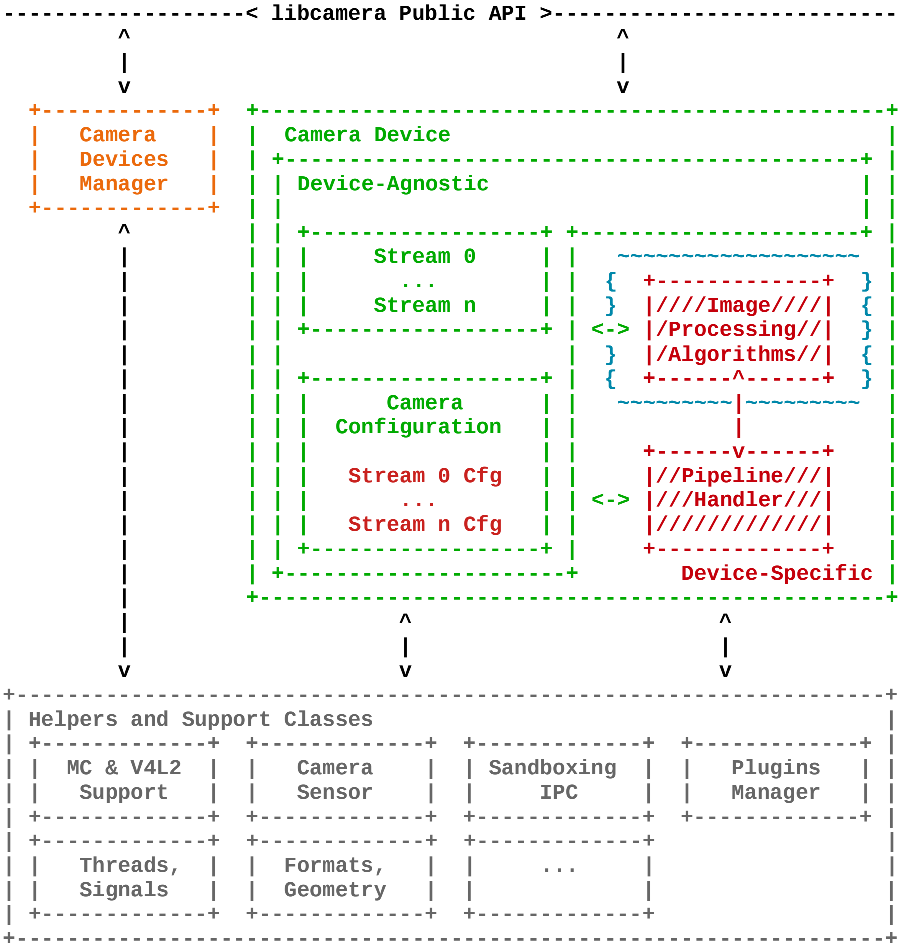
… and integrate libcamera with other operating systems.

Adaptation

```
   _
+-/ \-+
| (o) | libcamera
+-----+
```

IDEAS ON BOARD

libcamera architecture



```
-------------------< libcamera Public API >---------------------------
         ^                                      ^
         |                                      |
         v                                      v
  +-------------+  +-------------------------------------------------+
  |   Camera    |  |  Camera Device                                  |
  |   Devices   |  | +---------------------------------------------+ |
  |   Manager   |  | | Device-Agnostic                             | |
  +-------------+  | |                                             | |
         ^         | | +------------------+ +----------------------+ |
         |         | | |     Stream 0     | |   ~~~~~~~~~~~~~~~~~~~  |
         |         | | |       ...        | |  {  +-------------+  } |
         |         | | |     Stream n     | |  }  |////Image////|  { |
         |         | | +------------------+ | <-> |/Processing//|  } |
         |         | |                      |  }  |/Algorithms//|  { |
         |         | | +------------------+ |  {  +------^------+  } |
         |         | | |      Camera      | |   ~~~~~~~~~|~~~~~~~~~  |
         |         | | |  Configuration   | |            |           |
         |         | | |                  | |     +------v------+    |
         |         | | |   Stream 0 Cfg   | |     |//Pipeline///|    |
         |         | | |       ...        | | <-> |///Handler///|    |
         |         | | |   Stream n Cfg   | |     |/////////////|    |
         |         | | +------------------+ |     +-------------+    |
         |         | +----------------------+      Device-Specific |
         |         +-------------------------------------------------+
         |                     ^                        ^
         |                     |                        |
         v                     v                        v
+--------------------------------------------------------------------+
| Helpers and Support Classes                                        |
| +-------------+  +-------------+  +-------------+  +-------------+ |
| |  MC & V4L2  |  |   Camera    |  | Sandboxing  |  |   Plugins   | |
| |   Support   |  |   Sensor    |  |     IPC     |  |   Manager   | |
| +-------------+  +-------------+  +-------------+  +-------------+ |
| +-------------+  +-------------+  +-------------+                  |
| |   Threads,  |  |  Formats,   |  |     ...     |                  |
| |   Signals   |  |  Geometry   |  |             |                  |
| +-------------+  +-------------+  +-------------+                  |
+--------------------------------------------------------------------+
```

Central to the stack is the Camera object, interfacing to device-specific pipeline handlers.

/// Device-Specific Components
~~~ Sandboxing
~~~

```
   _
+-/ \-+
| (o) | libcamera
+-----+
```

IDEAS ON BOARD

Goals

- Provide camera support for all Linux systems

  Linux distros, Android, Chrome OS, …

- Create an environment that fosters camera innovation

  vendor blobs *and* open-source

- Feedback from the Android community on the overall architecture
- Feedback from SoC vendors on the device-specific interfaces and device support in general
- Next development steps for libcamera to support the LEVEL 3 profile
- Contribution of libcamera to Project Treble and integration in AOSP
- Future of the Android Camera HAL API and feedback from libcamera team
- Future of the Linux kernel camera APIs

Discussions

```
   _
+-/ \-+
| (o) | libcamera
+-----+
---------< libcamera Public API >------------------
                                      ^
                                      |
                                      v
+-------------------------------------------------+
|  Camera Device                                  |
| +---------------------------------------------+ |
| | Device-Agnostic                             | |
| |                                             | |
| | +------------------+ +----------------------+ |
| | |     Stream 0     | |   ~~~~~~~~~~~~~~~~~~~  |
| | |      ...         | |  {  +-------------+  } |
| | |     Stream n     | |  }  |////Image////|  { |
| | +------------------+ | <-> |/Processing//|  } |
| |                      |  }  |/Algorithms//|  { |
| | +------------------+ |  {  +------^------+  } |
| | |      Camera      | |   ~~~~~~~~~|~~~~~~~~~  |
| | |   Configuration  | |            |           |
| | |                  | |     +------v------+    |
| | |   Stream 0 Cfg   | |     |//Pipeline///|    |
| | |                  | | <-> |///Handler///|    |
| | |   Stream n Cfg   | |     |/////////////|    |
| | +------------------+ |     +-------------+    |
| +----------------------+        Device-Specific |
+-------------------------------------------------+
             ^                        ^
             |                        |
             v                        v
```

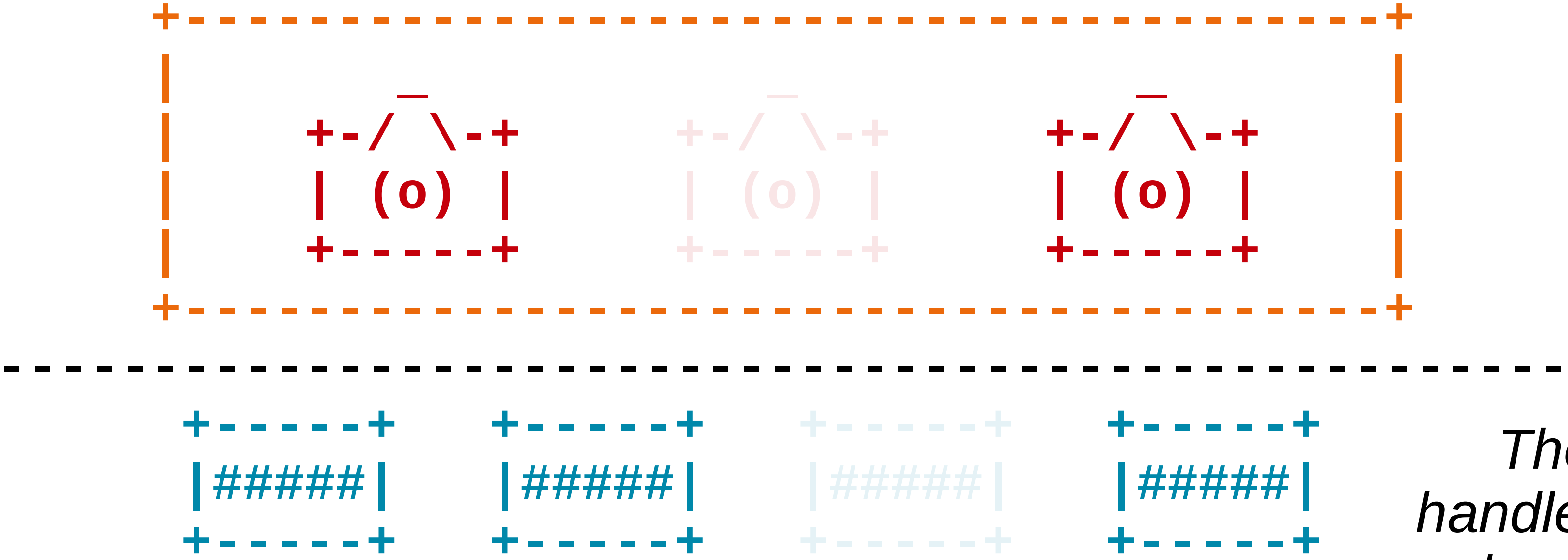
The camera configuration is backed by device-specific validation from the pipeline handler.

Camera Device

```
     +---------------------------------------+
     |                                       |
     |       _           _                   |
     |    +-/ \-+     +-/ \-+                |
     |    | (o) |     | (o) |                |
     |    +-----+     +-----+                |
     |                                       |
     +---------------------------------------+
--------------------------------------------------
      +------+    +------+    +------+
      |######|    |######|    |######|
      +------+    +------+    +------+
```

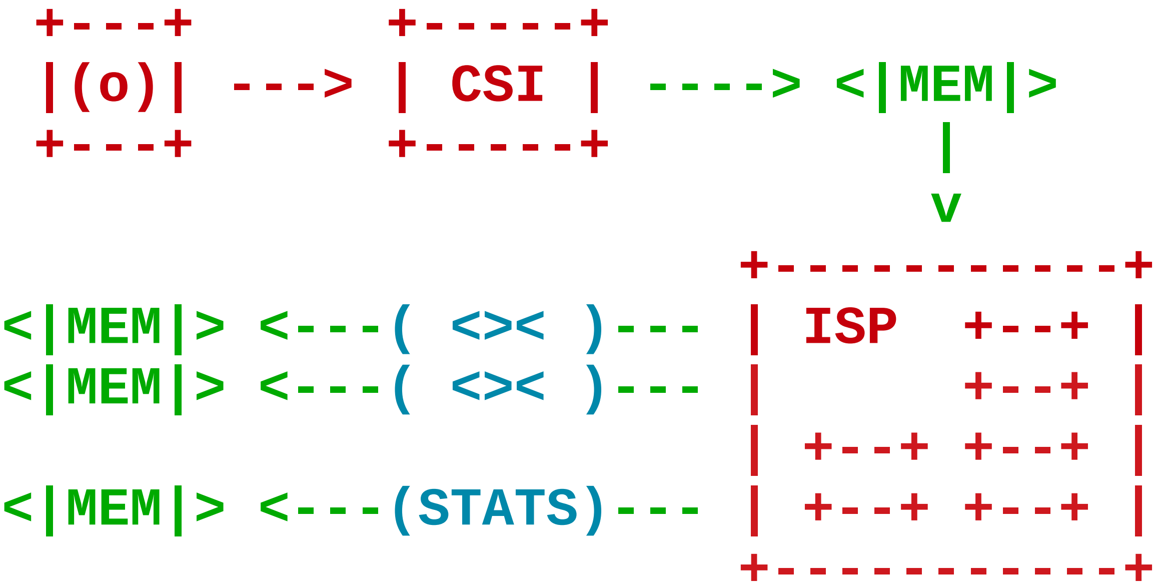
The Camera Manager enumerates media devices and instantiates corresponding pipeline handlers.

Camera Devices Manager



The pipeline handlers create and register one or more cameras.

Camera Devices Manager



```
 +---+      +-----+
 |(o)| ---> | CSI | ----> <|MEM|>
 +---+      +-----+          |
                             v
                       +-----------+
<|MEM|> <---( <>< )--- | ISP  +--+ |
<|MEM|> <---( <>< )--- |      +--+ |
                       | +--+ +--+ |
<|MEM|> <---(STATS)--- | +--+ +--+ |
                       +-----------+
```

The pipeline handler interfaces with all kernel devices. It abstracts them and exposes video streams to upper layers.

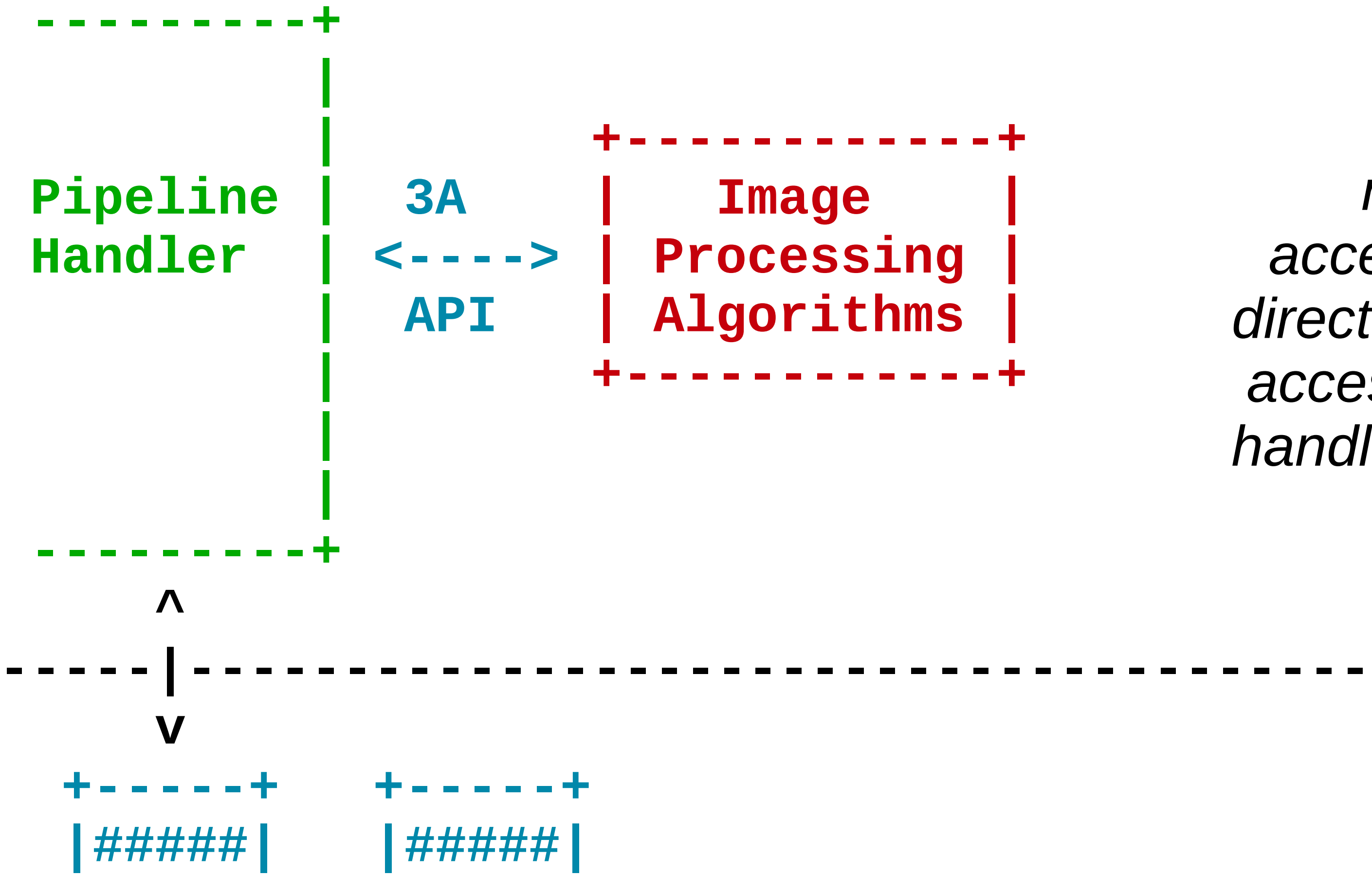
Pipeline Handler

```
+---+      +-----+        +-----+
|(o)| ---> | ISP | --+--> | <>< | --->
+---+      +-----+   |    +-----+
              ^      |
              |      |   +-------+
              |      \-> | Stats |
              |          +-------+
           +-----+           |
           | IPA | <---------/
           +-----+
```

Image Processing Algorithms (IPA) receive statistics from the hardware and compute optimal image parameters.

Image Processing Algorithms

Image Processing Algorithms



Pipeline Handler

3A API

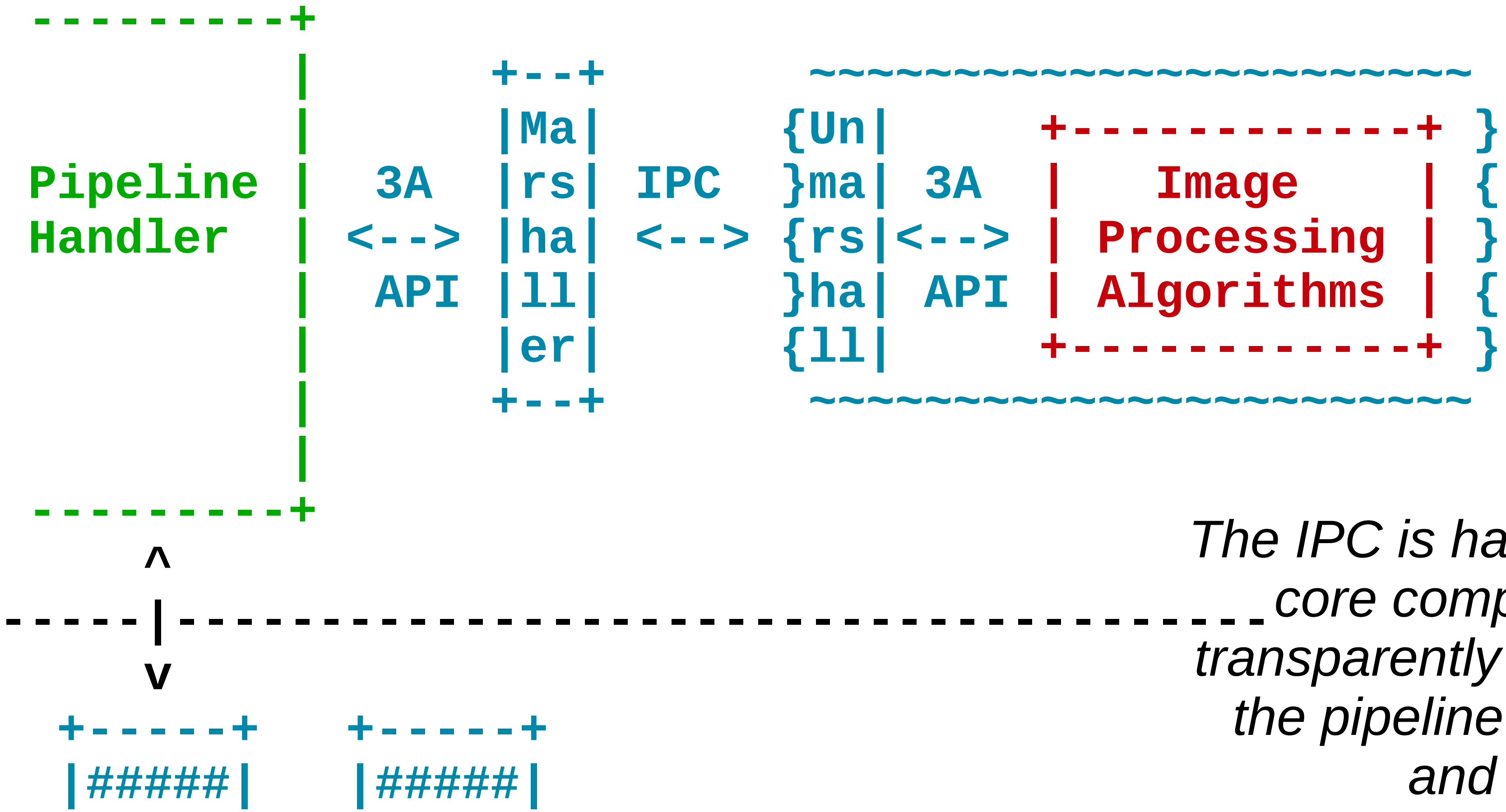
Image Processing Algorithms

IPAs are separate modules that don't access kernel devices directly. They only have access to their pipeline handler through the IPA API.

```
 ---------+
          |        ~~~~~~~~~~~~~~~~
          |      { +------------+ }
 Pipeline |  3A  } |   Image    | {
 Handler  | <----> | Processing | }
          |  API } | Algorithms | {
          |      { +------------+ }
          |        ~~~~~~~~~~~~~~~~
          |
 ---------+
     ^
-----|--------------------------------------
     v
  +------+  +------+
  |######|  |######|
```

Closed-source IPAs are sandboxed in a separate process. They communicate with the pipeline handler through IPC.

Image Processing Algorithms



```
 ---------+
          |      +--+      ~~~~~~~~~~~~~~~~~~~~~~~
          |      |Ma|      {Un|     +------------+ }
 Pipeline |  3A  |rs| IPC  }ma| 3A  |    Image   | {
 Handler  | <--> |ha| <--> {rs|<--> | Processing | }
          |  API |ll|      }ha| API | Algorithms | {
          |      |er|      {ll|     +------------+ }
          |      +--+      ~~~~~~~~~~~~~~~~~~~~~~~
          |
 ---------+
     ^
-----|--------------------------------------
     v
  +-----+   +-----+
  |#####|   |#####|
```

The IPC is handled in core components, transparently for both the pipeline handler and the IPA.

Image Processing Algorithms

```
         |                     |                        |
         v                     v                        v

+--------------------------------------------------------------------+
| Helpers and Support Classes                                        |
| +-------------+  +-------------+  +-------------+  +-------------+ |
| |  MC & V4L2  |  |   Camera    |  | Sandboxing  |  |   Plugins   | |
| |   Support   |  |   Sensor    |  |     IPC     |  |   Manager   | |
| +-------------+  +-------------+  +-------------+  +-------------+ |
| +-------------+  +-------------+  +-------------+                  |
| |   Threads,  |  |  Formats,   |  |     ...     |                  |
| |   Signals   |  |  Geometry   |  |             |                  |
| +-------------+  +-------------+  +-------------+                  |
+--------------------------------------------------------------------+
```

Many helper classes ease the implementation of pipeline handlers for device vendors.

Helpers and Support Classes

```
|                                                    |
|             Native V4L2 Application                |
+---------------------------------------------------+
         | open()      | ioctl()       | mmap()
         v             v               v
+----------------------------------------------------+
| LD_PRELOAD=libcamera-v4l2.so                       |
|                                                    |
|    open() {      ioctl() {       mmap() {          |
|        ...           ...             ...           |
|    }             }               }                 |
+----------------------------------------------------+
                       | libcamera
                       v API
+----------------------------------------------------+
|                   libcamera                        |
|                                                    |
```

Native V4L2 applications are supported through a transparent compatibility layer.

V4L2 Compatibility

```
+--------------------------+
| Android Camera Framework |
+--------------------------+


+--------------------------+
|    \___/    +---------+  | HW level    |
|    /. .\    |  JPEG   |  |             |
|    -----    | Encoder |  | - EXTERNAL  | t
|  !| Cam.|!  +---------+  | - LEGACY    | i
|  !| HAL |!               | - LIMITED   | m
|    -----        ...      | - FULL      | e
|     | |                  | - LEVEL_3   |
+--------------------------+             v


+--------------------------+
|        libcamera         |
```

A single Android camera HAL module implementation for all devices supported by libcamera.

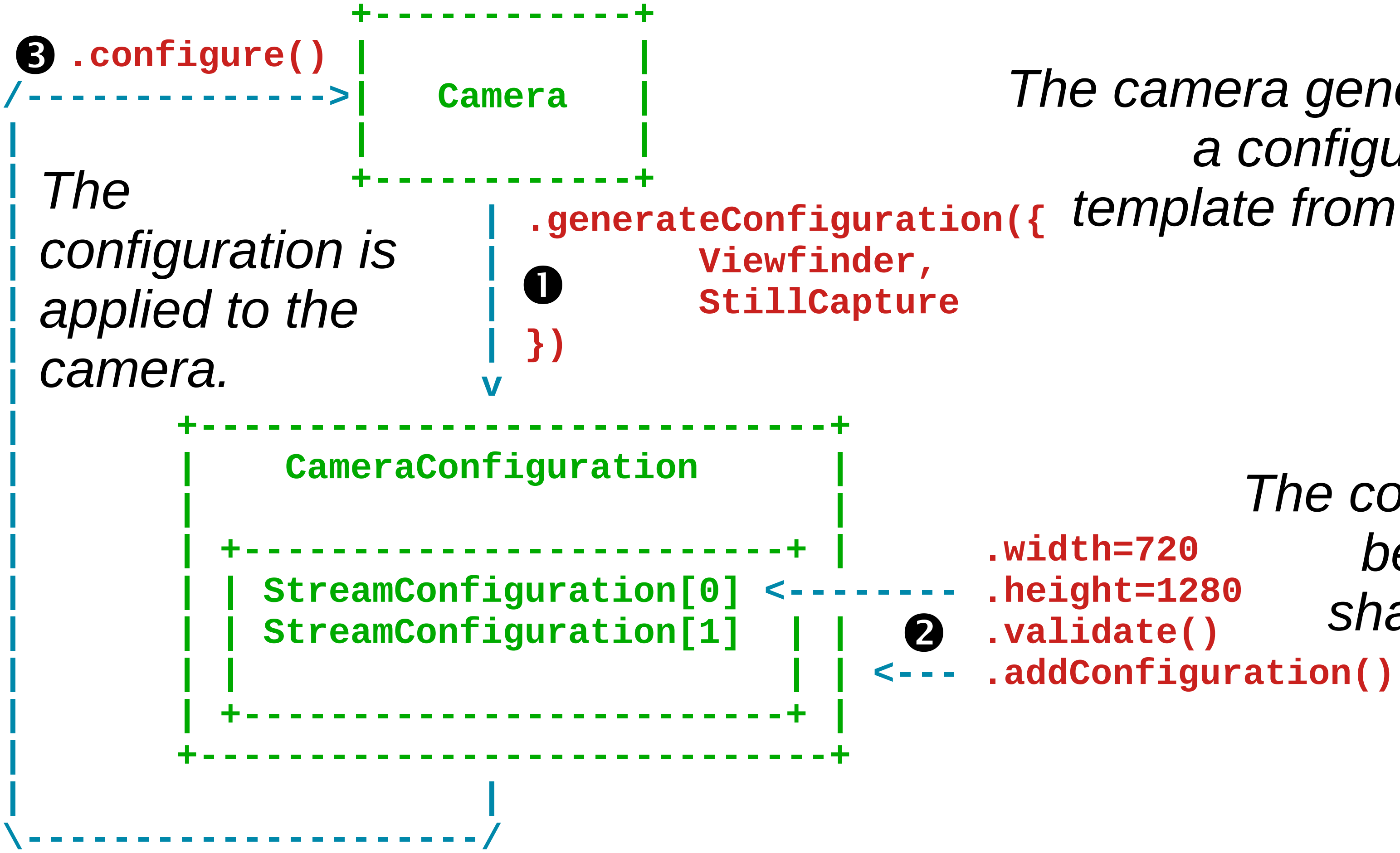
Android Camera HAL

```
   _
+-/ \-+
| (o) | libcamera
+-----+
```

Camera State Machine

```
   release()
    /-----\
    v     |
  ___________                ____________               ____________
//           \\  acquire()  /            \ configure() /            \
|| Available ||------------>|  Acquired  |------------>| Configured |
\\___________//             \____________/             \____________/
       ^         release()         |                       |  |  ^
       \---------------------------/-----------------------/  |  |
                                                              |  | freeBuffers()
                                            allocateBuffers() |  |
                                                              v  |
                                             start()
                             ____________               ____________
                            /            \<------------/            \
                            |  Running   |             |  Prepared  |
                            \____________/------------>\____________/
                               |      ^      stop()       |      ^
                               \------/                   \------/
                            createRequest()            createRequest()
                             queueRequest()
```

The camera implements a simple state machine.



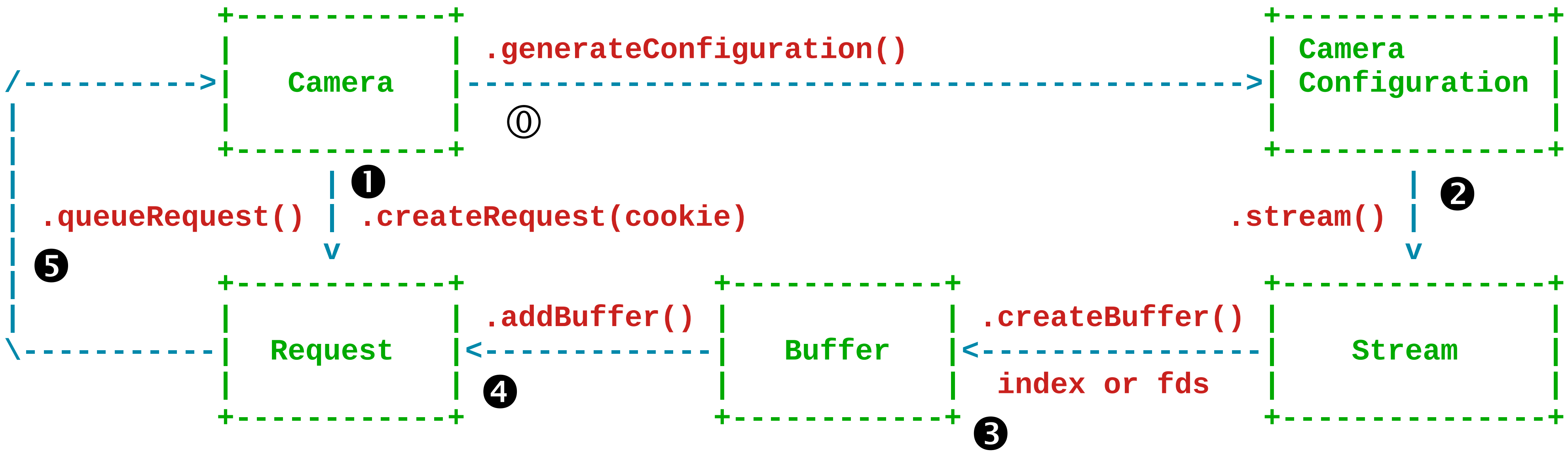
```
                +------------+
❸ .configure()  |            |
/-------------->|   Camera   |
|               |            |
|               +------------+
|                     |  .generateConfiguration({
|                     |          Viewfinder,
|                     |  ❶       StillCapture
|                     |  })
|                     v
|       +-----------------------------+
|       |    CameraConfiguration      |
|       |                             |
|       | +-------------------------+ |     .width=720
|       | | StreamConfiguration[0] <---------  .height=1280
|       | | StreamConfiguration[1]  | |   ❷  .validate()
|       | |                         | | <---  .addConfiguration()
|       | +-------------------------+ |
|       +-----------------------------+
|                     |
\---------------------/
```

The configuration is applied to the camera.

The camera generates a configuration template from roles.

The configuration can be modified, and shall be validated.

Camera Configuration



```
            +------------+                                              +---------------+
            |            | .generateConfiguration()                     | Camera        |
/---------->|   Camera   |--------------------------------------------->| Configuration |
|           |            |   ⓪                                          |               |
|           +------------+                                              +---------------+
|                 | ❶                                                          | ❷
| .queueRequest() | .createRequest(cookie)                           .stream() |
| ❺               v                                                            v
|           +------------+              +------------+                  +---------------+
|           |            | .addBuffer() |            | .createBuffer()  |               |
\-----------|  Request   |<-------------|   Buffer   |<-----------------|    Stream     |
            |            | ❹            |            |  index or fds    |               |
            +------------+              +------------+ ❸                +---------------+
```

A request is created on the Camera, populated with a Buffer for each Stream, and queued for capture.

Request Queuing

```
+-------------+ ❶.bufferCompleted()    +--------
|             |----------------------->|
|   Camera    | ❷.requestCompleted()   | Application
|             |----------------------->|
+-------------+                        +--------
      ^                                    |
      |  ❸ Create, prepare and queue       |
      \------------------------------------/
                 next request
```

Buffer and request completion are notified separately.

Applications submit new requests to keep the streams going.

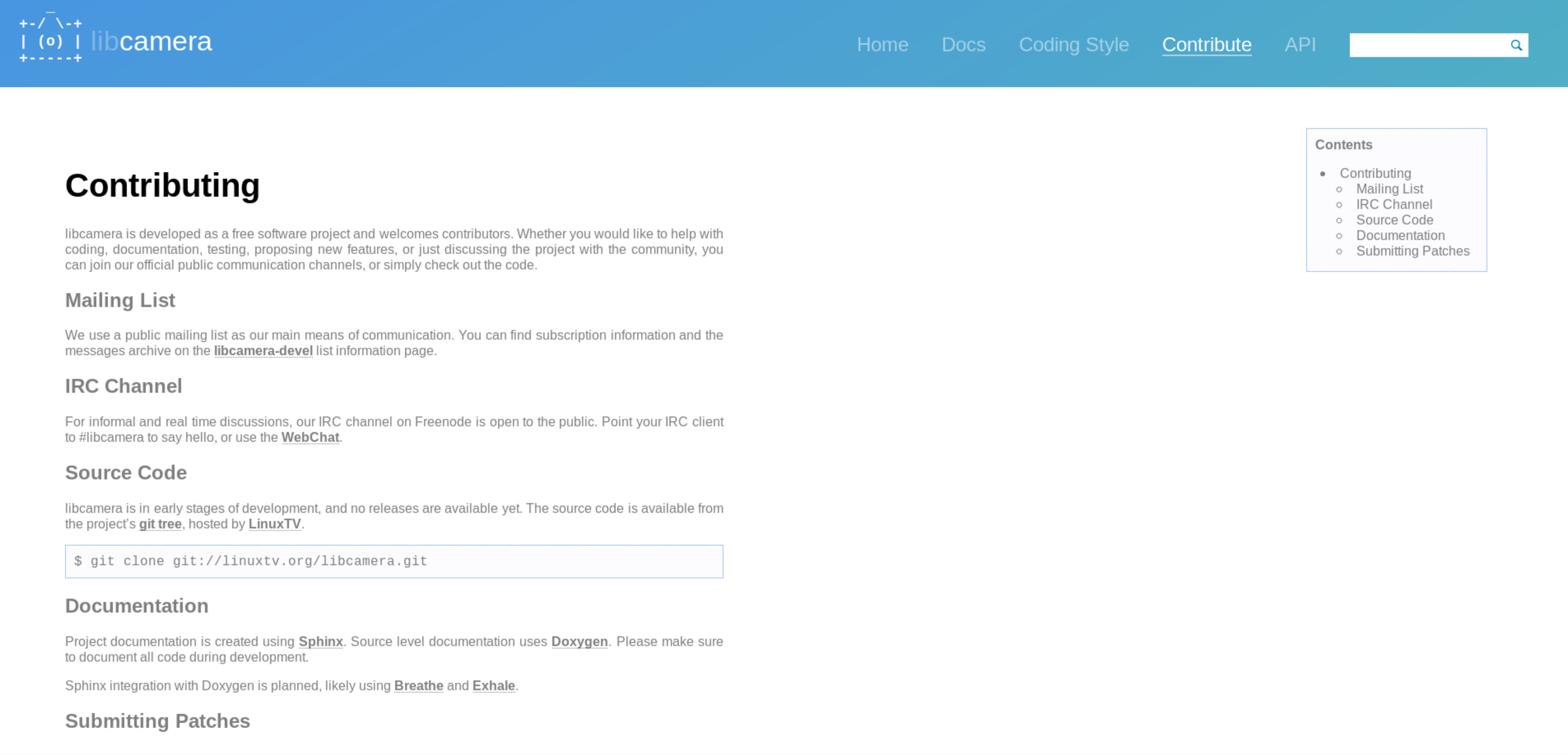
Request Completion

```
   _
+-/ \-+
| (o) |
+-----+
```

libcamera

IDEAS ON BOARD

Contribute



Contents

- Contributing
  - Mailing List
  - IRC Channel
  - Source Code
  - Documentation
  - Submitting Patches

Contributing

libcamera is developed as a free software project and welcomes contributors. Whether you would like to help with coding, documentation, testing, proposing new features, or just discussing the project with the community, you can join our official public communication channels, or simply check out the code.

Mailing List

We use a public mailing list as our main means of communication. You can find subscription information and the messages archive on the **libcamera-devel** list information page.

IRC Channel

For informal and real time discussions, our IRC channel on Freenode is open to the public. Point your IRC client to #libcamera to say hello, or use the **WebChat**.

Source Code

libcamera is in early stages of development, and no releases are available yet. The source code is available from the project's **git tree**, hosted by **LinuxTV**.

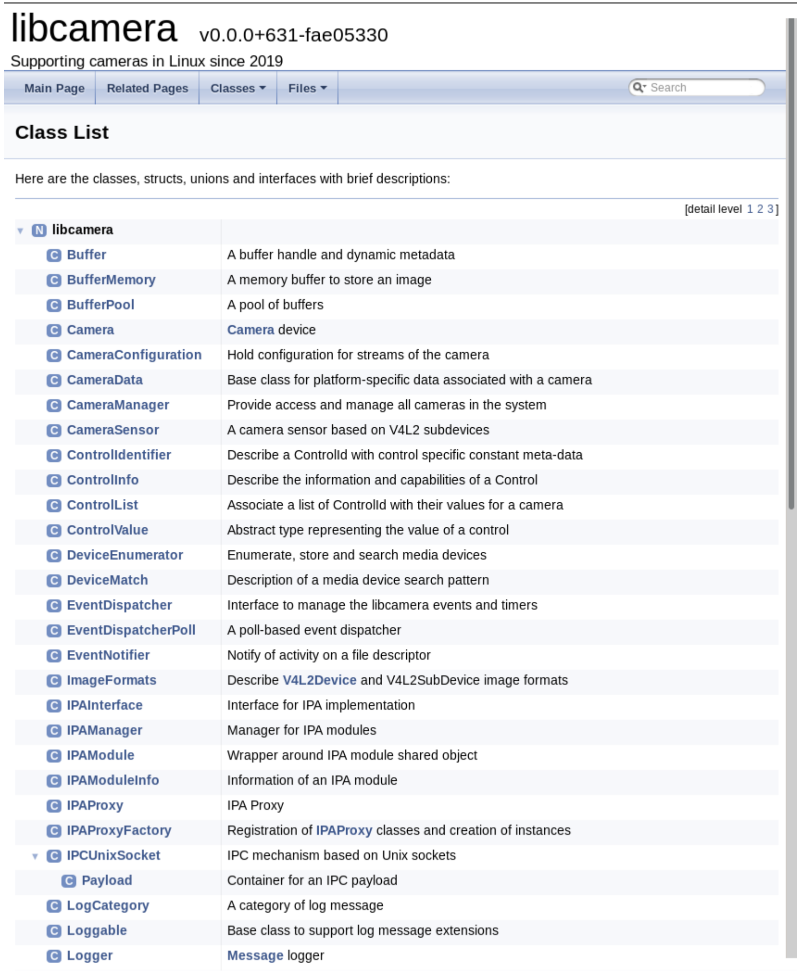
```
$ git clone git://linuxtv.org/libcamera.git
```

Documentation

Project documentation is created using **Sphinx**. Source level documentation uses **Doxygen**. Please make sure to document all code during development.

Sphinx integration with Doxygen is planned, likely using **Breathe** and **Exhale**.

Submitting Patches



libcamera v0.0.0+631-fae05330

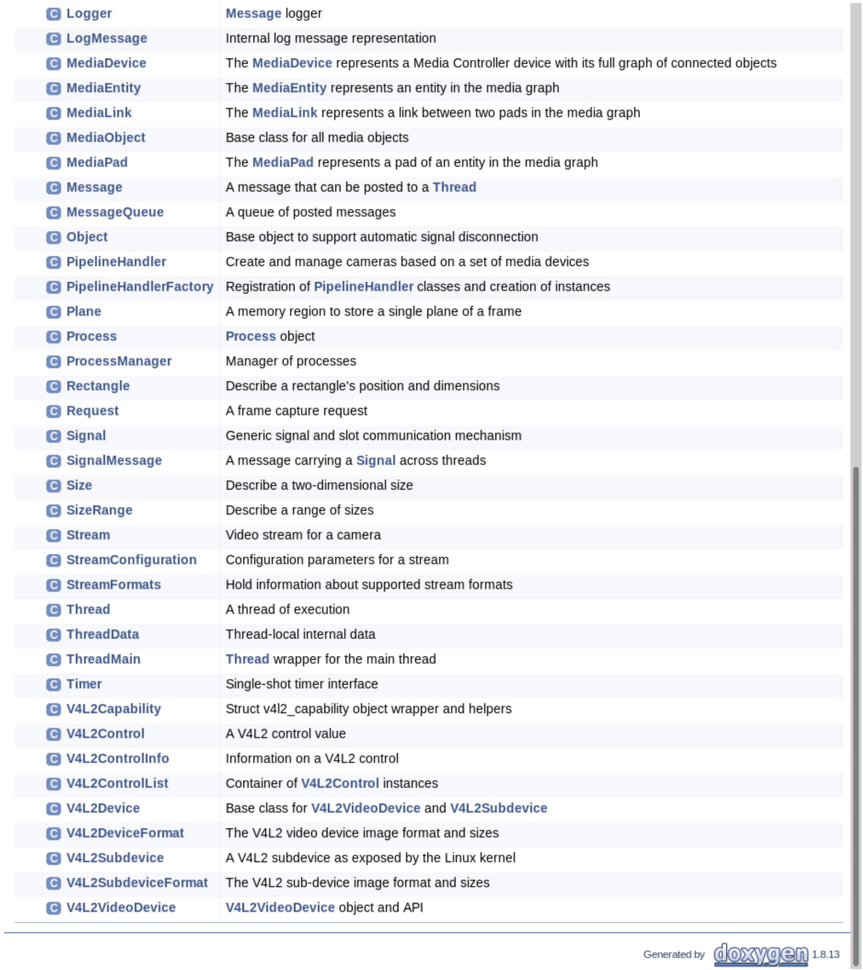
Supporting cameras in Linux since 2019

Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

[detail level 1 2 3]

▼ N libcamera	
C Buffer	A buffer handle and dynamic metadata
C BufferMemory	A memory buffer to store an image
C BufferPool	A pool of buffers
C Camera	Camera device
C CameraConfiguration	Hold configuration for streams of the camera
C CameraData	Base class for platform-specific data associated with a camera
C CameraManager	Provide access and manage all cameras in the system
C CameraSensor	A camera sensor based on V4L2 subdevices
C ControlIdentifier	Describe a ControlId with control specific constant meta-data
C ControlInfo	Describe the information and capabilities of a Control
C ControlList	Associate a list of ControlId with their values for a camera
C ControlValue	Abstract type representing the value of a control
C DeviceEnumerator	Enumerate, store and search media devices
C DeviceMatch	Description of a media device search pattern
C EventDispatcher	Interface to manage the libcamera events and timers
C EventDispatcherPoll	A poll-based event dispatcher
C EventNotifier	Notify of activity on a file descriptor
C ImageFormats	Describe V4L2Device and V4L2SubDevice image formats
C IPAInterface	Interface for IPA implementation
C IPAManager	Manager for IPA modules
C IPAModule	Wrapper around IPA module shared object
C IPAModuleInfo	Information of an IPA module
C IPAProxy	IPA Proxy
C IPAProxyFactory	Registration of IPAProxy classes and creation of instances
▼ C IPCUnixSocket	IPC mechanism based on Unix sockets
C Payload	Container for an IPC payload
C LogCategory	A category of log message
C Loggable	Base class to support log message extensions
C Logger	Message logger



C Logger	Message logger
C LogMessage	Internal log message representation
C MediaDevice	The MediaDevice represents a Media Controller device with its full graph of connected objects
C MediaEntity	The MediaEntity represents an entity in the media graph
C MediaLink	The MediaLink represents a link between two pads in the media graph
C MediaObject	Base class for all media objects
C MediaPad	The MediaPad represents a pad of an entity in the media graph
C Message	A message that can be posted to a Thread
C MessageQueue	A queue of posted messages
C Object	Base object to support automatic signal disconnection
C PipelineHandler	Create and manage cameras based on a set of media devices
C PipelineHandlerFactory	Registration of PipelineHandler classes and creation of instances
C Plane	A memory region to store a single plane of a frame
C Process	Process object
C ProcessManager	Manager of processes
C Rectangle	Describe a rectangle's position and dimensions
C Request	A frame capture request
C Signal	Generic signal and slot communication mechanism
C SignalMessage	A message carrying a Signal across threads
C Size	Describe a two-dimensional size
C SizeRange	Describe a range of sizes
C Stream	Video stream for a camera
C StreamConfiguration	Configuration parameters for a stream
C StreamFormats	Hold information about supported stream formats
C Thread	A thread of execution
C ThreadData	Thread-local internal data
C ThreadMain	Thread wrapper for the main thread
C Timer	Single-shot timer interface
C V4L2Capability	Struct v4l2_capability object wrapper and helpers
C V4L2Control	A V4L2 control value
C V4L2ControlInfo	Information on a V4L2 control
C V4L2ControlList	Container of V4L2Control instances
C V4L2Device	Base class for V4L2VideoDevice and V4L2Subdevice
C V4L2DeviceFormat	The V4L2 video device image format and sizes
C V4L2Subdevice	A V4L2 subdevice as exposed by the Linux kernel
C V4L2SubdeviceFormat	The V4L2 sub-device image format and sizes
C V4L2VideoDevice	V4L2VideoDevice object and API

Generated by doxygen 1.8.13

API Documentation

libcamera-devel@lists.libcamera.org
irc://chat.freenode.net/#libcamera

laurent.pinchart@ideasonboard.com

Contact

? !

Obrigado