Android 的多媒体系统

Android 的多媒体系统

- 第一部分 多媒体系统的结构
- 第二部分 多媒体的各个层次
- 第三部分 多媒体实现的核心部分 OpenCore

第一部分 多媒体系统的结构

Android 的多媒体部分的框架涉及到应用层、 JAVA 框架、 C 语言框架、硬件抽象层等环节

多媒体主要包括两方面的内容:

- 输入输出环节（音频视频的输入输出）
- 中间处理环节（编解码环节）

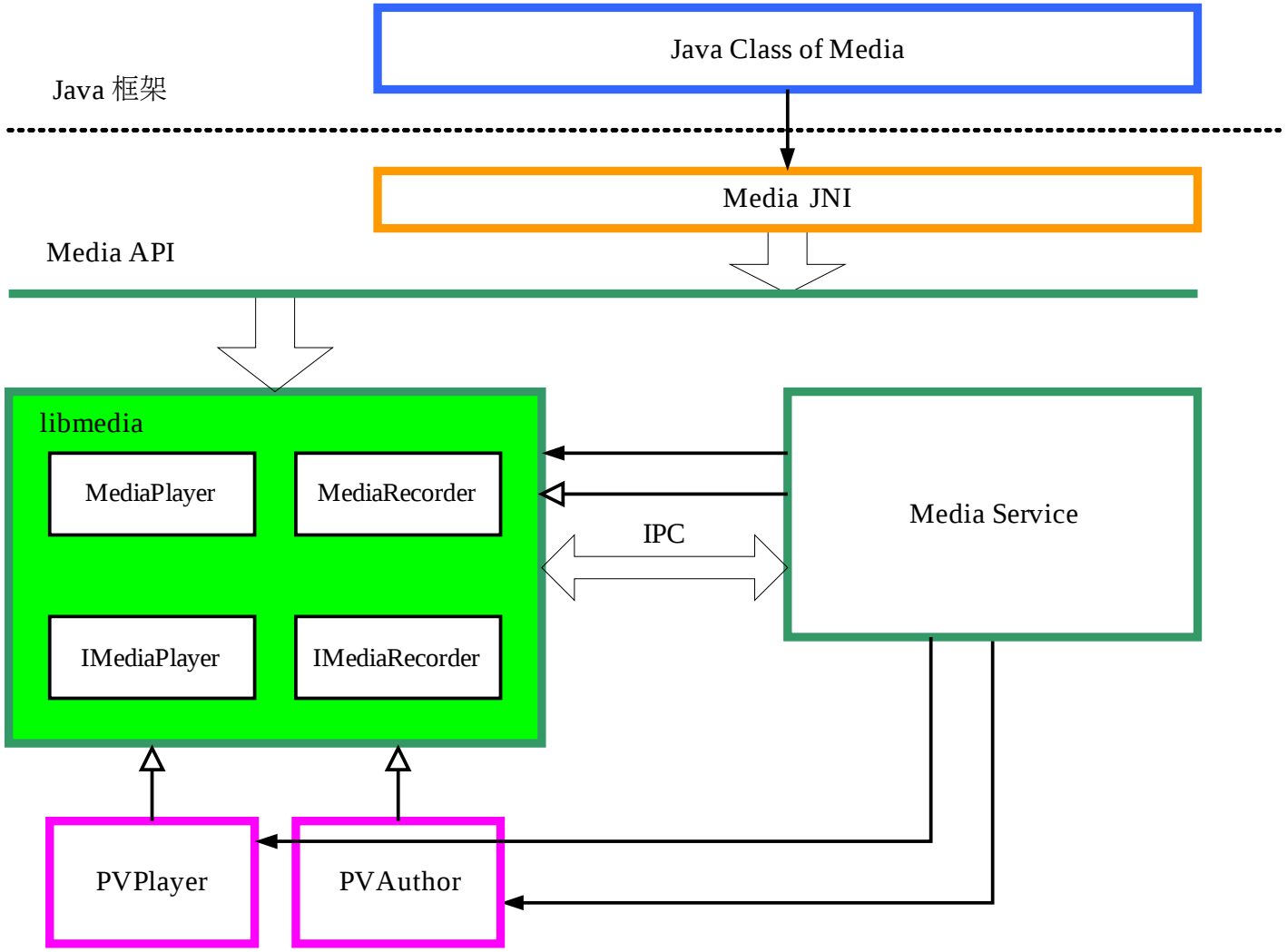
其中，输入输出环节由其他方面的硬件抽象层实现，中间处理环节主要由 PacketVideo 实现，可以使用硬件加速。

第一部分 多媒体系统的结构

Android的多媒体应用业务:

- Music Player
- Video Player
- Camera
- Sound Recorder
- Camcorder
- Video Telephone

第一部分 多媒体系统的结构



第一部分 多媒体系统的结构

JAVA 类：
frameworks/base/media/java/android/media/
类的名称为 android.media.*

JAVA 本地调用部分（JNI）：
frameworks/base/media/*
这部分内容编译成为目标是 libmedia_jni.so.

基于 OpenCore 的多媒体播放器和记录器
external/opencore/

第一部分 多媒体系统的结构

多媒体框架的 media 库：
frameworks/base/include/media/
frameworks/base/media/libmedia/
这部分的内容被编译成库 libmedia.so 。

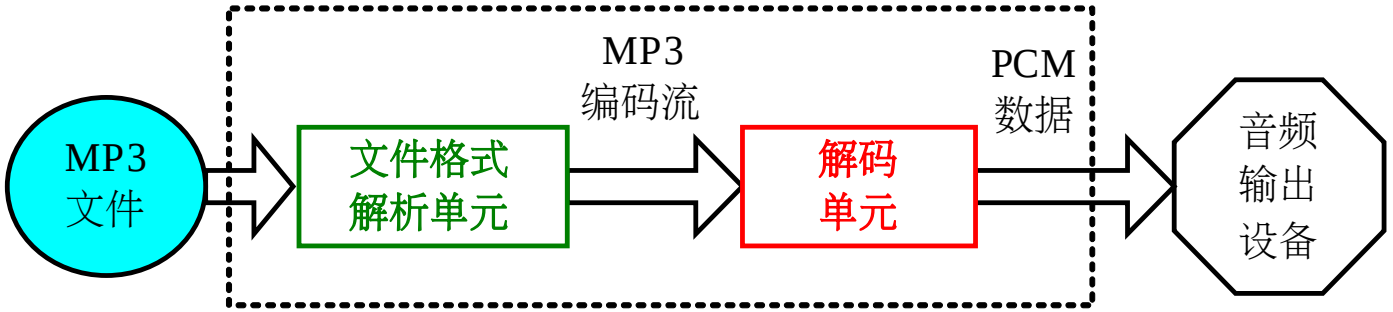
多媒体服务部分：
frameworks/base/media/libmediaplayerservice/
这部分内容被编译成库 libmediaplayerservice.so 。

第一部分 多媒体系统的结构

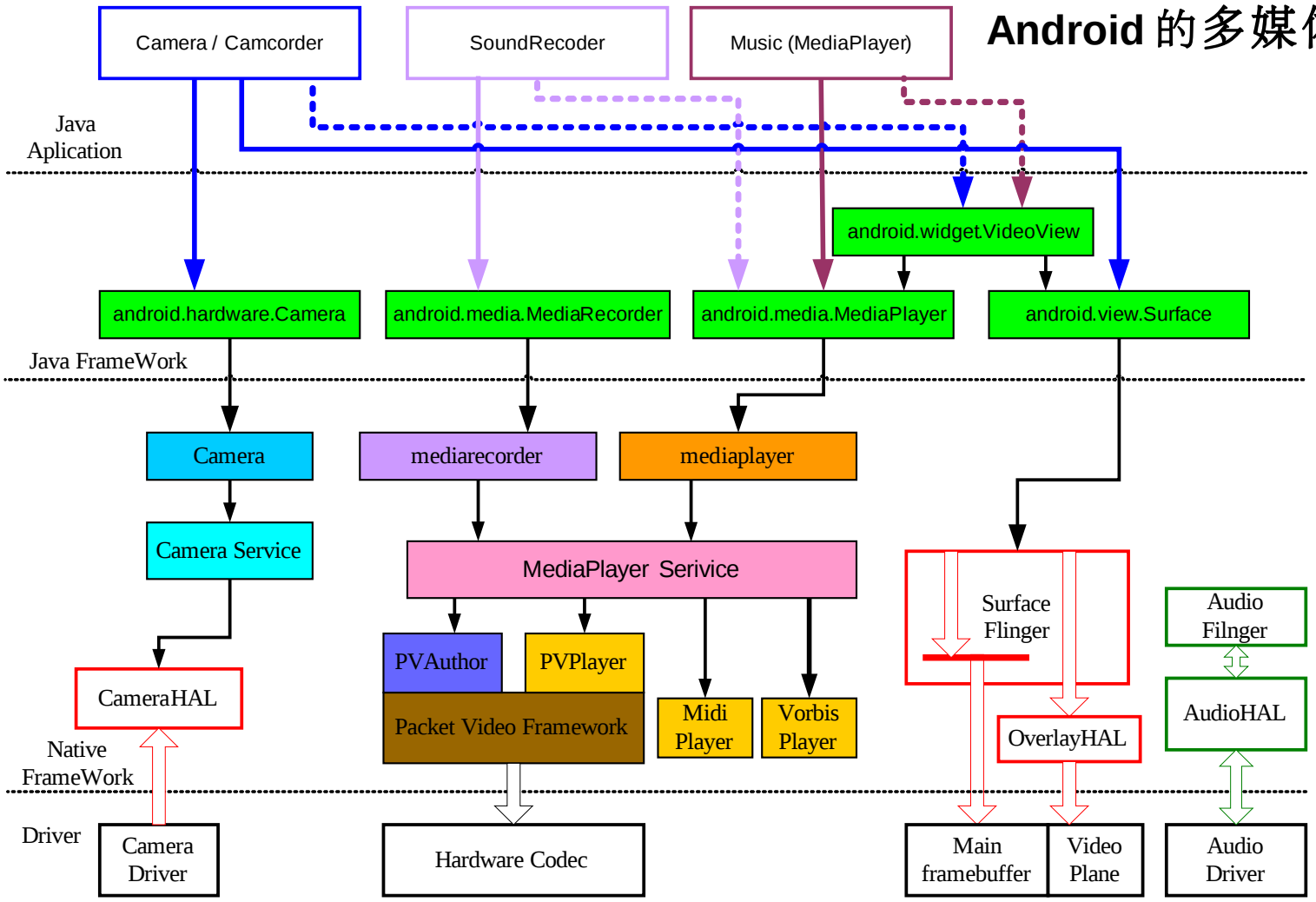
从多媒体应用实现的角度，主要包括两方面的内容：

- 输入输出环节（音频、视频纯数据流的输入输出系统）
- 中间处理环节（文件格式处理环节和编解码环节）

以一个 MP3 播放器为例，从功能的角度就是将一个 mp3 格式的文件作为播放器的输入，将声音从播放设备输出。从实现的角度， MP3 播放器经过了一下的阶段： MP3 格式的文件解析、 MP3 编码流的解码、 PCM 输出的播放。

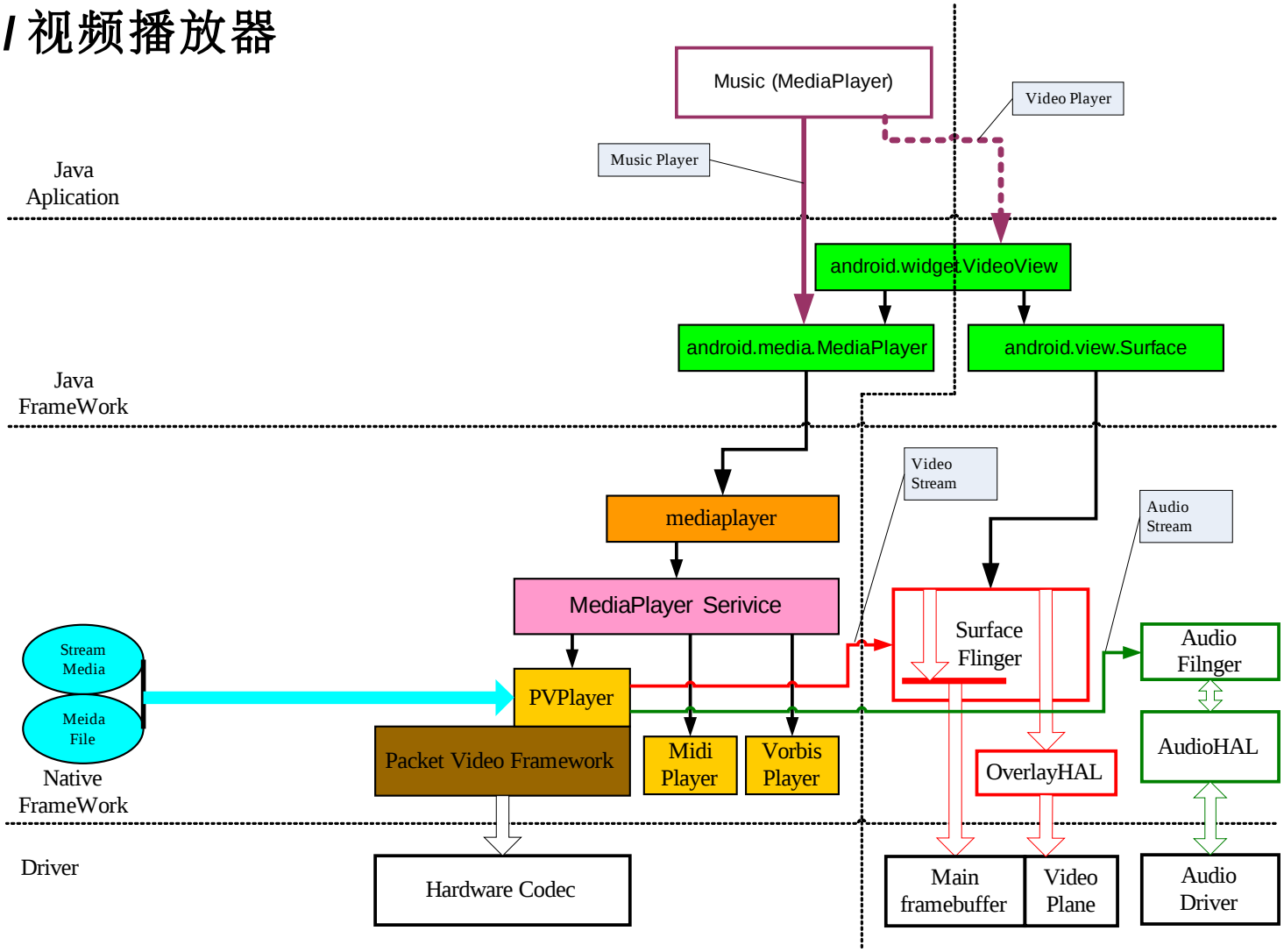


3.1 多媒体的各种业务

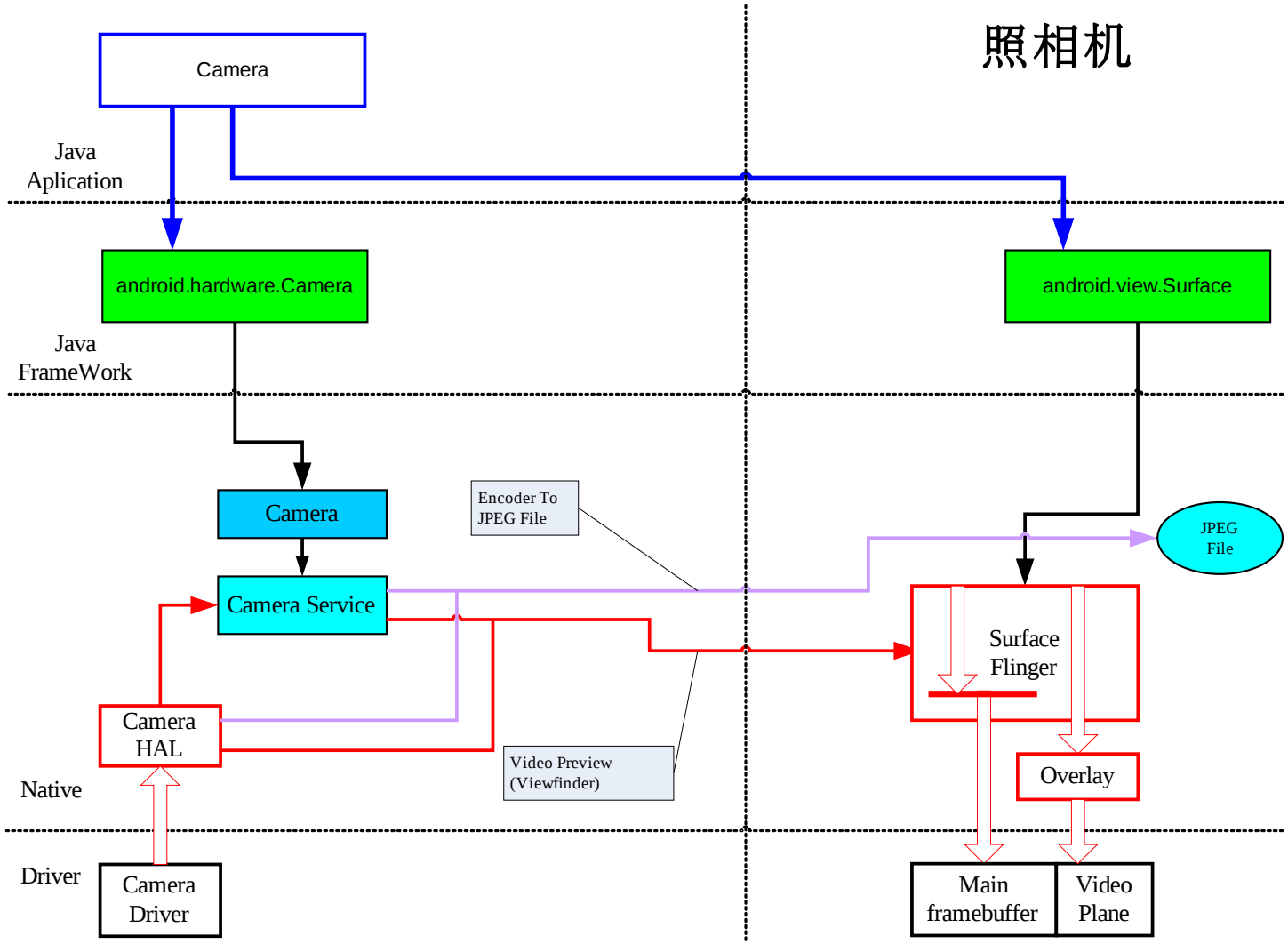


3.1 多媒体的各种业务

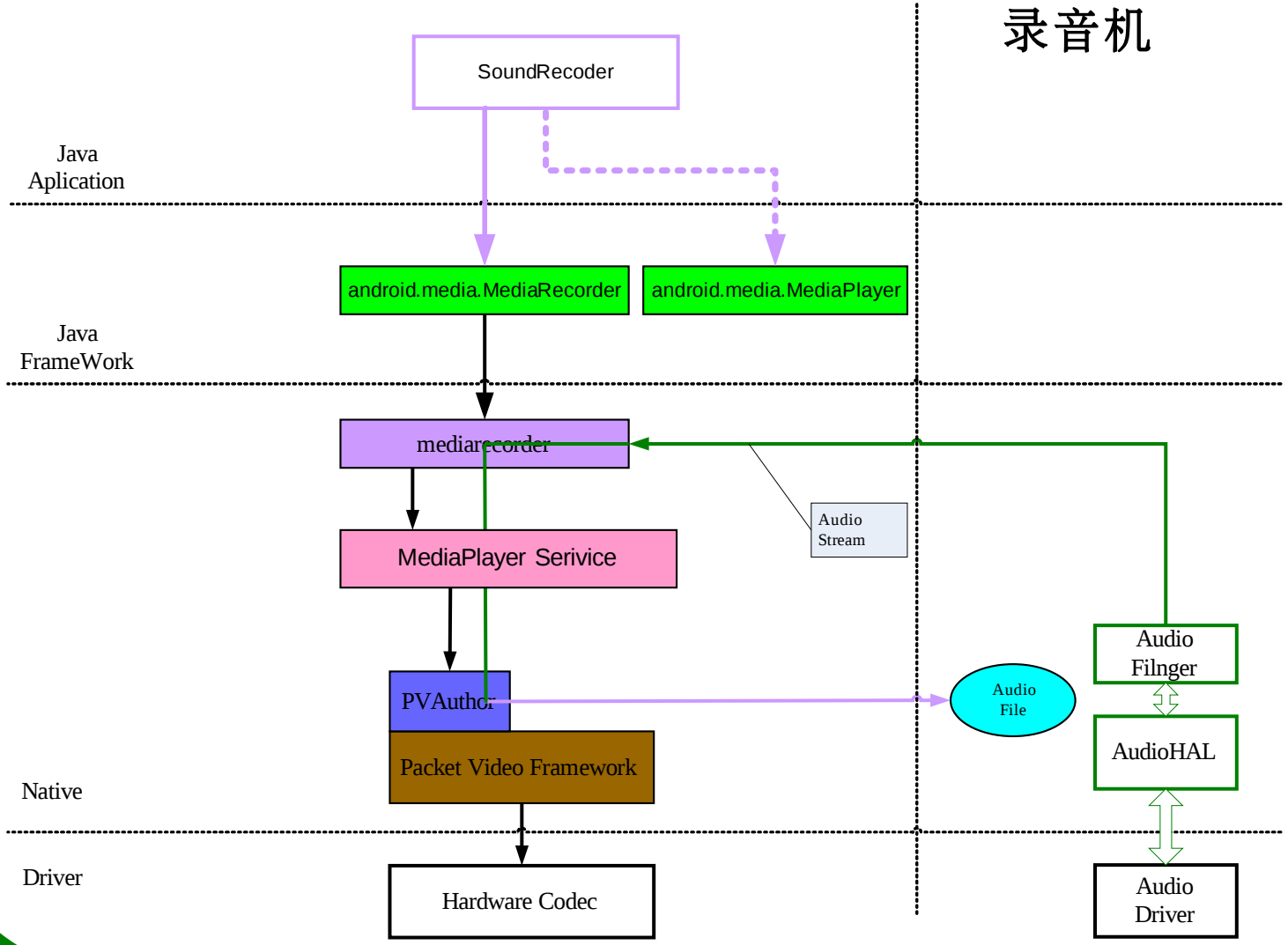
音频 / 视频播放器



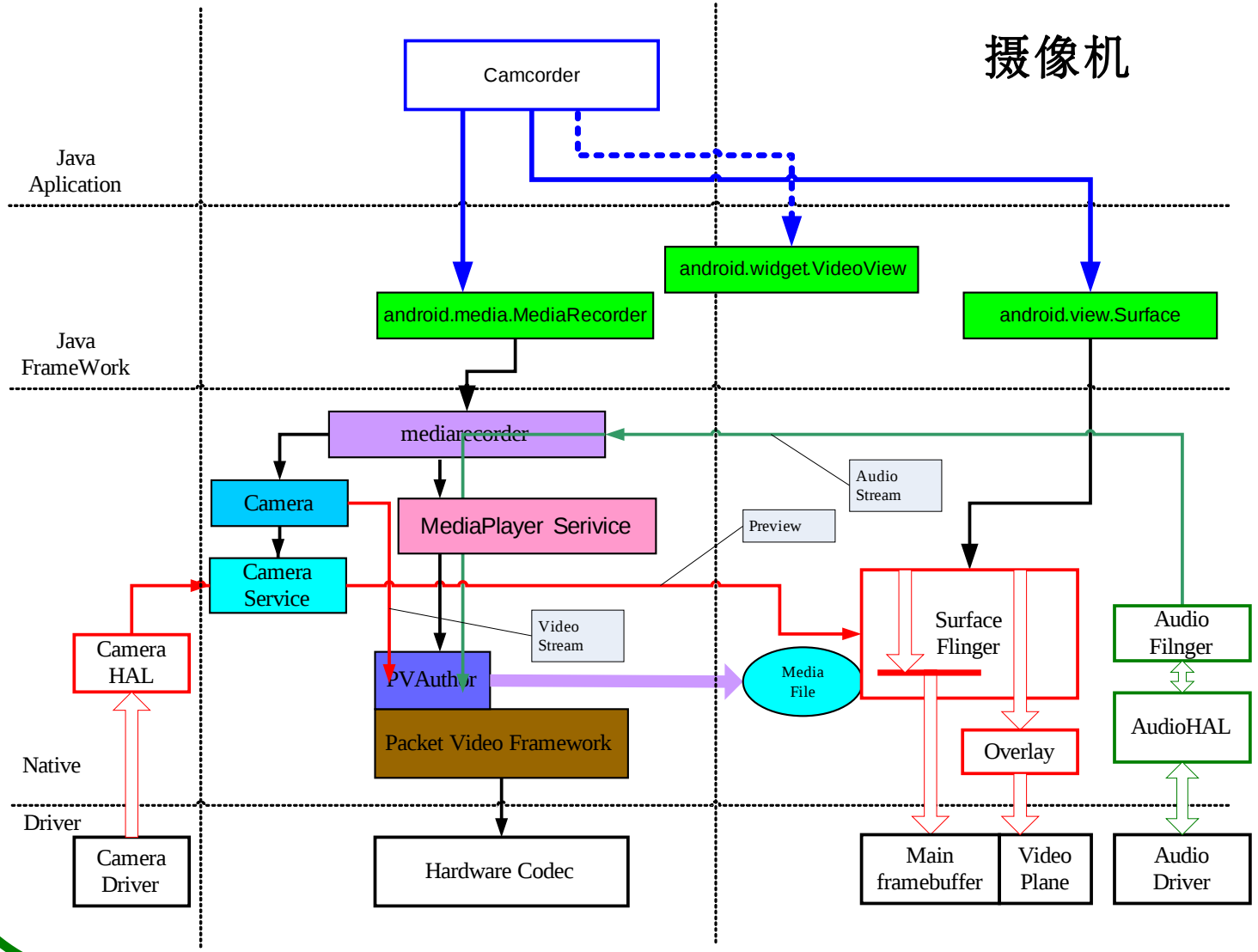
3.1 多媒体的各种业务



3.1 多媒体的各种业务



3.1 多媒体的各种业务



第二部分 多媒体的各个层次

2.1 libmedia 的框架部分

2.2 多媒体服务

2.3 多媒体部分的 JNI 代码

2.4 多媒体部分的 JAVA 框架代码

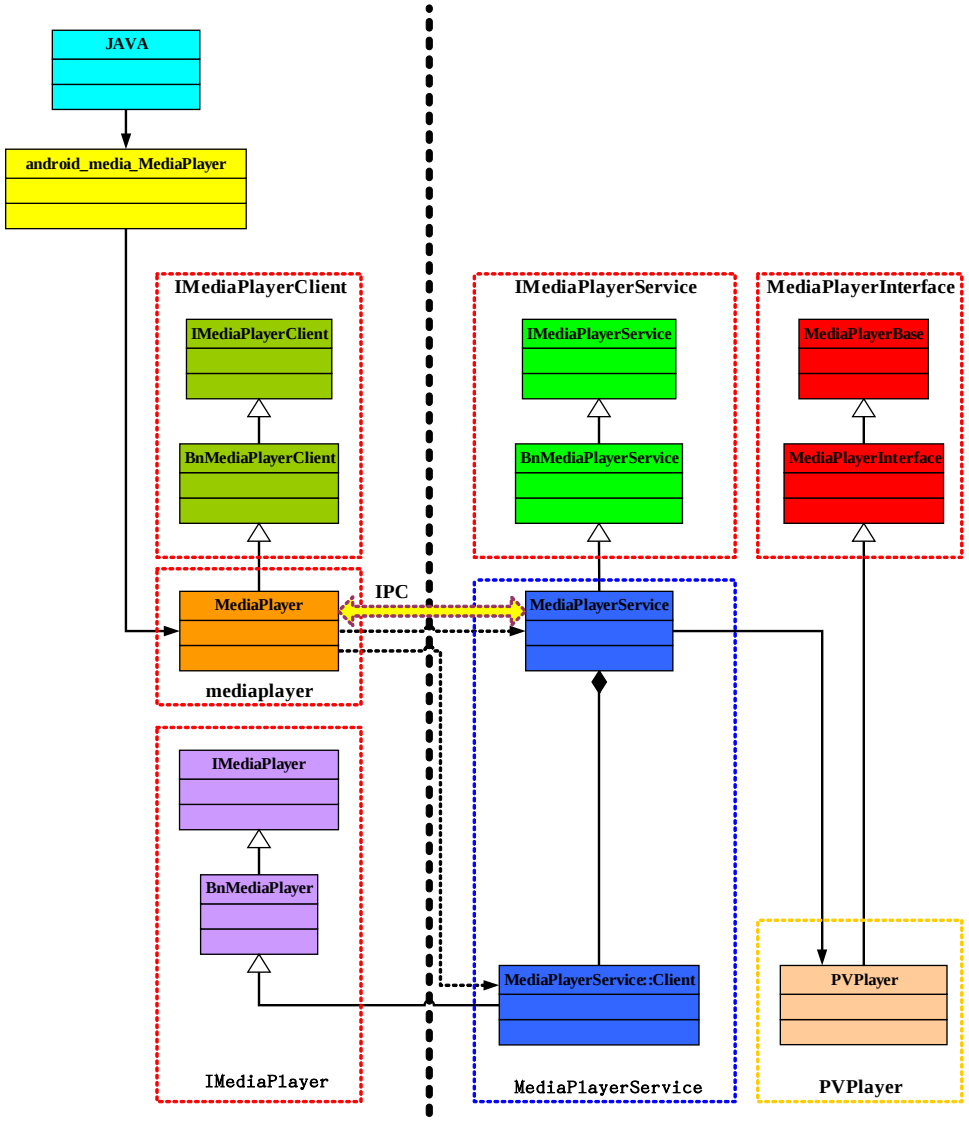
2.5 类 android.widget.VideoView

第二部分 多媒体的各个层次

Android 多媒体部分的 C 语言部分的核心是 media 库，它主要记录了媒体播放器和媒体记录器的框架。

media 库向上层通过 JNI 提供接口，下层通过 Packet Video 等实现。

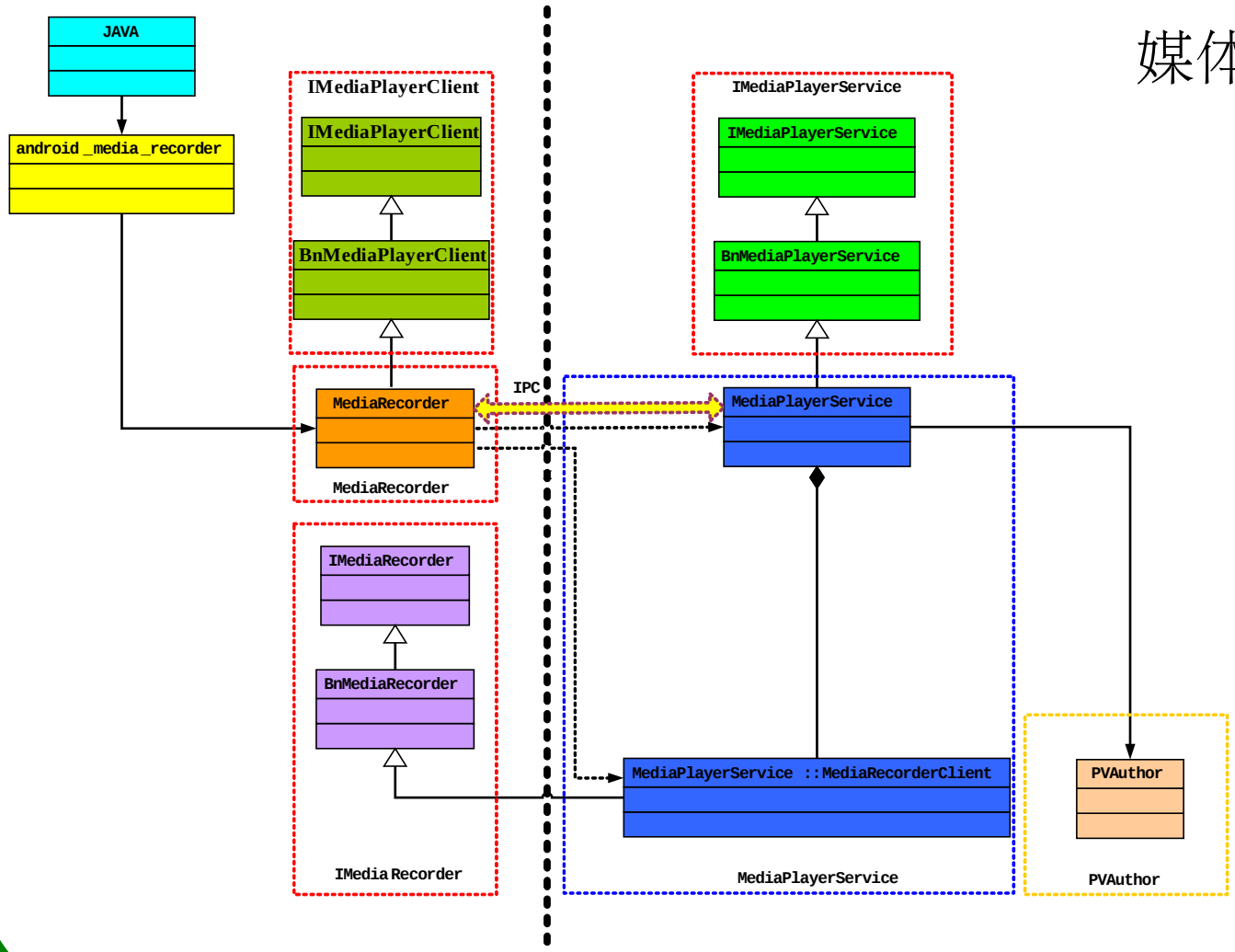
2.1 libmedia 的框架部分



媒体播放器

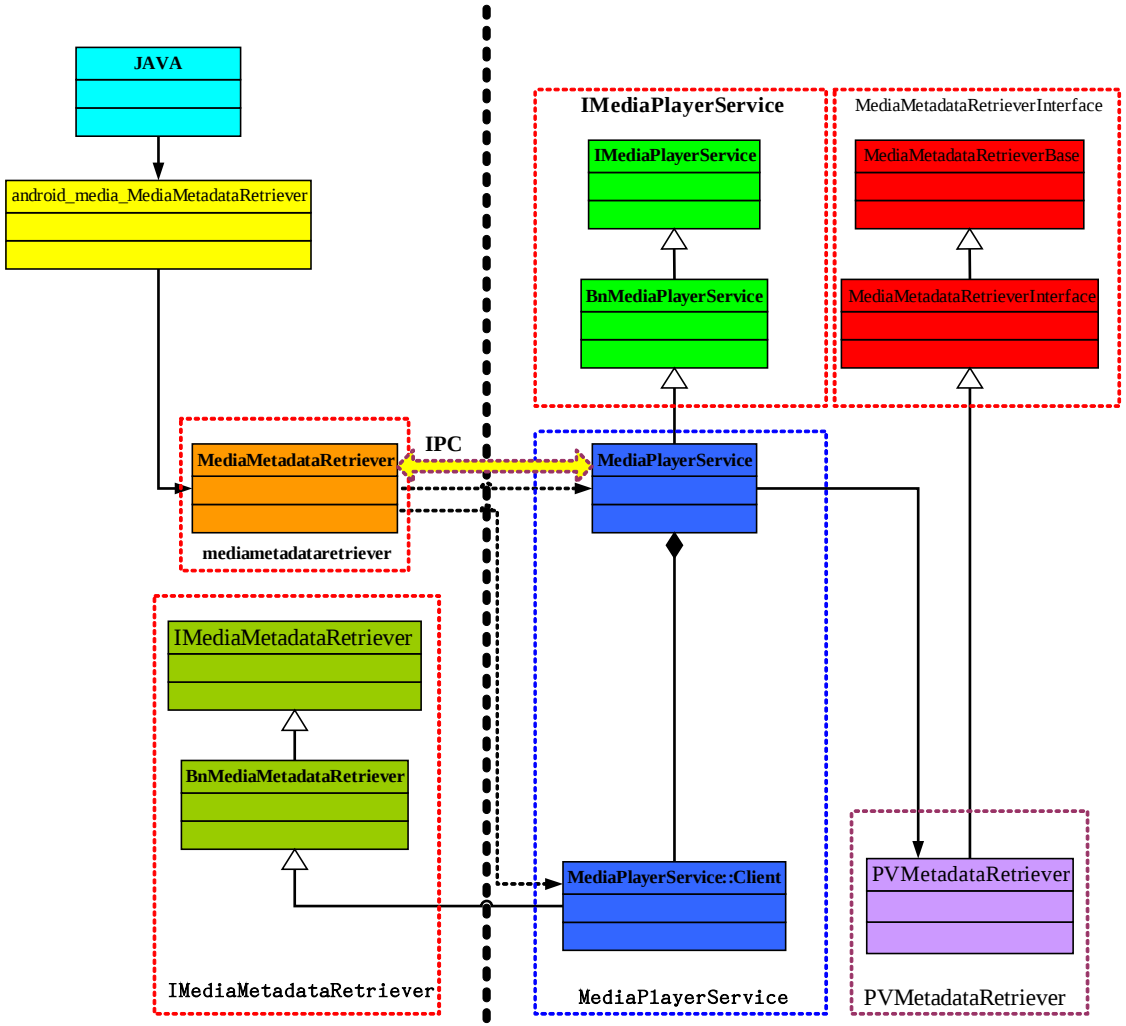
2.1 libmedia 的框架部分

媒体纪录器



2.1 libmedia 的框架部分

媒体元信息



3.2 多媒体的服务

多媒体服务的守护进程的代码：

frameworks/base/media/mediaserver/

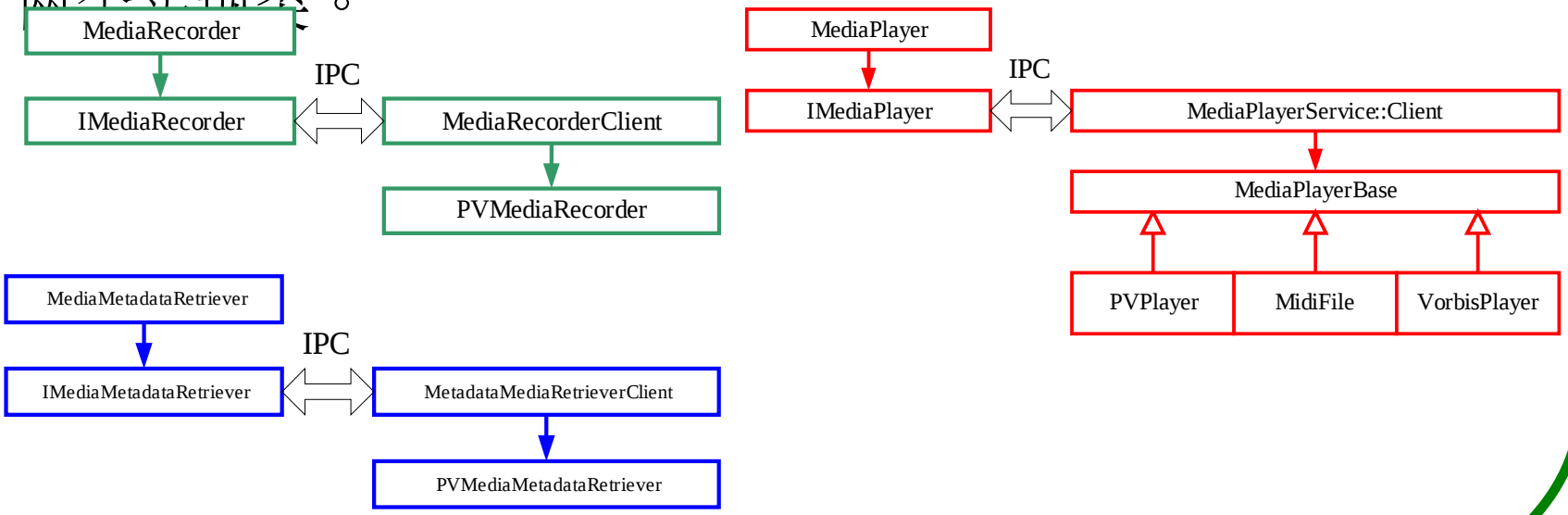
其中只有一个源文件 main_mediaserver.cpp，将被编译成为一个可执行程序 mediaserver。它负责启动了多媒体服务、照相机服务，音频服务这三个服务

```
service media /system/bin/mediaserver
    user media
    group system audio camera graphics inet net_bt net_bt_admin
```

3.2 多媒体的服务

多媒体服务提供多媒体的服务部分，多媒体服务通过 IPC 与 libmedia 库通信，多媒体服务其路径是：
frameworks/base/media/libmediaplayerservice/

其中，主要的类是 MediaPlayerService，从这个类中可以获取 IMediaPlayer 和 IMediaRecorder 两个实现类。



2.2 多媒体的服务

MediaPlayerService 的定义：

```
class MediaPlayerService : public BnMediaPlayerService
{
    virtual sp<IMediaRecorder> createMediaRecorder(pid_t pid);
    virtual sp<IMediaMetadataRetriever> createMetadataRetriever(pid_t pid);

    virtual sp<IMediaPlayer>  create(pid_t pid,
                                     const sp<IMediaPlayerClient>& client,
                                     const char* url);
    virtual sp<IMediaPlayer>  create(pid_t pid,
                                     const sp<IMediaPlayerClient>& client,
                                     int fd,
                                     int64_t offset, int64_t length);
}
```

2.3 多媒体部分的 JNI 代码

media 的 JAVA 本地调用部分（JNI）：
frameworks/base/media/*

这部分内容编译成为目标是 libmedia_jni.so.

主要文件是:
android_media_MediaPlayer.cpp
android_media_MediaRecorder.cpp

2.4 多媒体部分的 JAVA 代码

JAVA 类：
frameworks/base/media/java/android/media/

主要的文件是：
MediaPlayer.java
MediaRecorder.java

分别表示两个类：
android.media.mediaPlayer
android.media.mediaRecorder

这两个类的接口和 JNI 层次乃至本地代码是非常类似的。

2.5 VideoView 类

VideoView 是一个 media 集成的高层的 JAVA 类，这个类的文件是：：
frameworks/base/core/java/android/widget/
文件的类型是 VideoView.java
类的路径是 android.widget.java

VideoView 是一个集成了 MediaPlayer 和 SurfaceView 的类，可以作为一个 UI 元素（View）直接放置在 JAVA 应用的界面中，用于视频的播放。

第三部分 多媒体实现的核心部分 OpenCore

3.1 OpenCore 概述

3.2 OpenCore 的层次结构

3.3 OpenCore 的 OSCL 部分

3.4 OpenCore 的文件格式和编解码部分

3.5 OpenCore Node 介绍

3.6 OpenCore 的移植和扩展

3.7 OpenCore Player 介绍

3.8 OpenCore Author 介绍

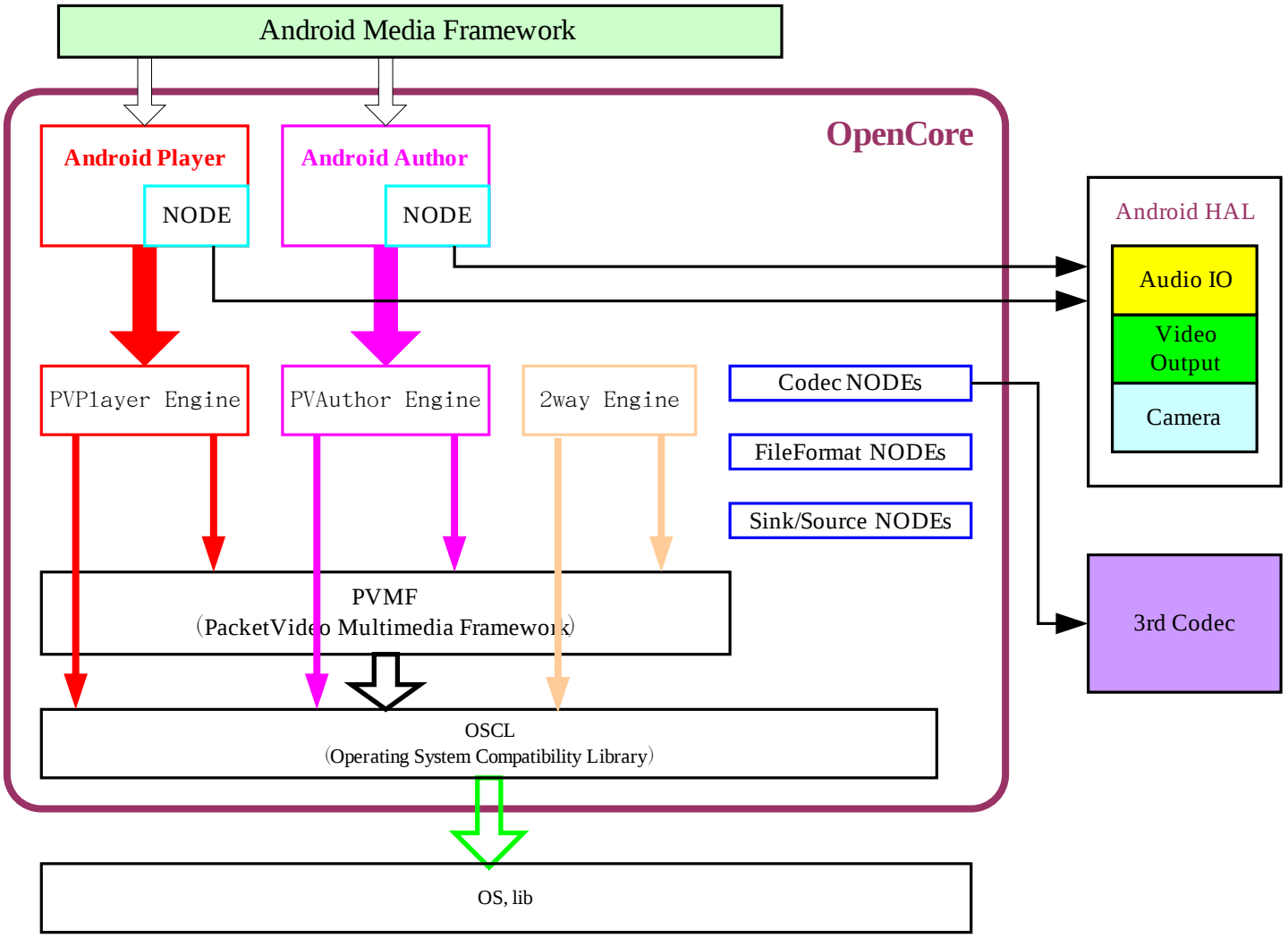
3.1 OpenCore 概述

OpenCore 是一个多媒体的框架，从宏观上来看，它主要包含了两大方面的内容：

- PVPlayer：
  提供媒体播放器的功能，完成各种音频（Audio）、视频（Video）流的回放（Playback）功能
- PVAuthor：
  提供媒体流记录的功能，完成各种音频（Audio）、视频（Video）流的捕获（Recording）功能

PVPlayer 和 PVAuthor 以 SDK 的形式提供给开发者，可以在这个 SDK 之上构建多种应用程序和服务。在移动终端中常常使用的多媒体应用程序，例如媒体播放器、照相机、录像机、录音机等等。

3.2 OpenCore 的层次结构



3.2 OpenCore 的层次结构

OpenCore 是一个多媒体的框架，从宏观上来看，它主要包含了两大方面的内容：

- OSCL（Operating System Compatibility Library，操作系统兼容层）
- PVMF（PacketVideo Multimedia Framework， PV 多媒体框架）
- 文件解析（parser）和组成（composer）两个部分
- 编解码部分
- NODEs （节点）
- Player Engine （播放器引擎）
- Author Engine （作者引擎）

3.2 OpenCore 的 OSCL 部分

OSCL，全称为 Operating System Compatibility Library (操作系统兼容库)，它包含了一些在不同操作系统中移植层的功能，它的在 OpenCore oscl/oscl 目录中，一般每一个目录表示一个模块。

OSCL 对应的功能是非常细致的，几乎对 C 语言中每一个细节的功能都进行封装，并使用了 C++ 的接口提供给上层使用。事实上， OperCore 中的 PVMF 、 Engine 部分都在使用 OSCL ， 而整个 OperCore 的调用者也需要使用 OSCL 。

3.4 OpenCore 的文件格式和编解码部分

OpenCore 有关文件格式处理和编解码部分两部分的内容，分别在目录 fileformats 和 codecs_v2 当中。这两部分都属于基础性的功能，不涉及具体的逻辑，因此它们被别的模块调用来使用。

文件格式处理有两种类型，一种是 parser （解析器），另一种是 composer （组成器）。

编解码部分的子目录 omx 实现了一个 khronos OpenMAX 的功能。 OpenMAX 是一个多媒体应用程序的框架标准

3.4 OpenCore 的文件格式和编解码部分

文件格式的处理部分：

由于同时涉及播放文件和记录文件两种功能，因此 OpenCore 中的文件格式处理有两种类型，一种是 parser （解析器），另一种是 composer （组成器）。其代码的目录为 fileformats 的目录，其中包含 mp3， mp4， wav 等子目录。

其中包含了 AVI， mp3， mp4， wav 等多种文件的解析器和组成器，各个目录中对应的是不同的文件格式等。

3.4 OpenCore 的文件格式和编解码部分

编解码部分:

编解码部分主要针对 Audio 和 Video 的软件编解码，其目录为 codecs_v2，其中包含了 audio 、 omx 、 utilities 、 video 等几个目录。

在 audio 和 video 目录中，对应了针对各种流的子目录，其中可能包含 dec 和 enc 两个目录，分别对应解码和编码。

```
video
 |-- avc_h264
 |   |-- common
 |   |-- dec
 |   |-- enc
 |   `-- patent_disclaimer.txt
 `-- m4v_h263
     |-- dec
     |-- enc
     `-- patent_disclaimer.txt
```

3.4 OpenCore 的文件格式和编解码部分

codecs_v2 目录的子目录 omx 实现了一个 khronos OpenMAX 的功能。

OpenMAX 是一个多媒体应用程序的框架标准，由 NVIDIA 公司和 Khronos 在 2006 年推出。OpenMAX IL 1.0（集成层）技术规格定义了媒体组件接口，以便在嵌入式器件的流媒体框架中快速集成加速式编解码器。

OpenMAX 的设计实现可以让具有硬件编辑码功能的平台提供统一的接口和框架，在 OpenMAX 中可以直接使用硬件加速的进行编解码乃至输出的功能，对外保持统一的接口。
此处的 OpenMAX 则是基于一个纯软件的实现，其实现的代码即是调用这里的 video 和 audio 目录中的软件编辑码的代码。

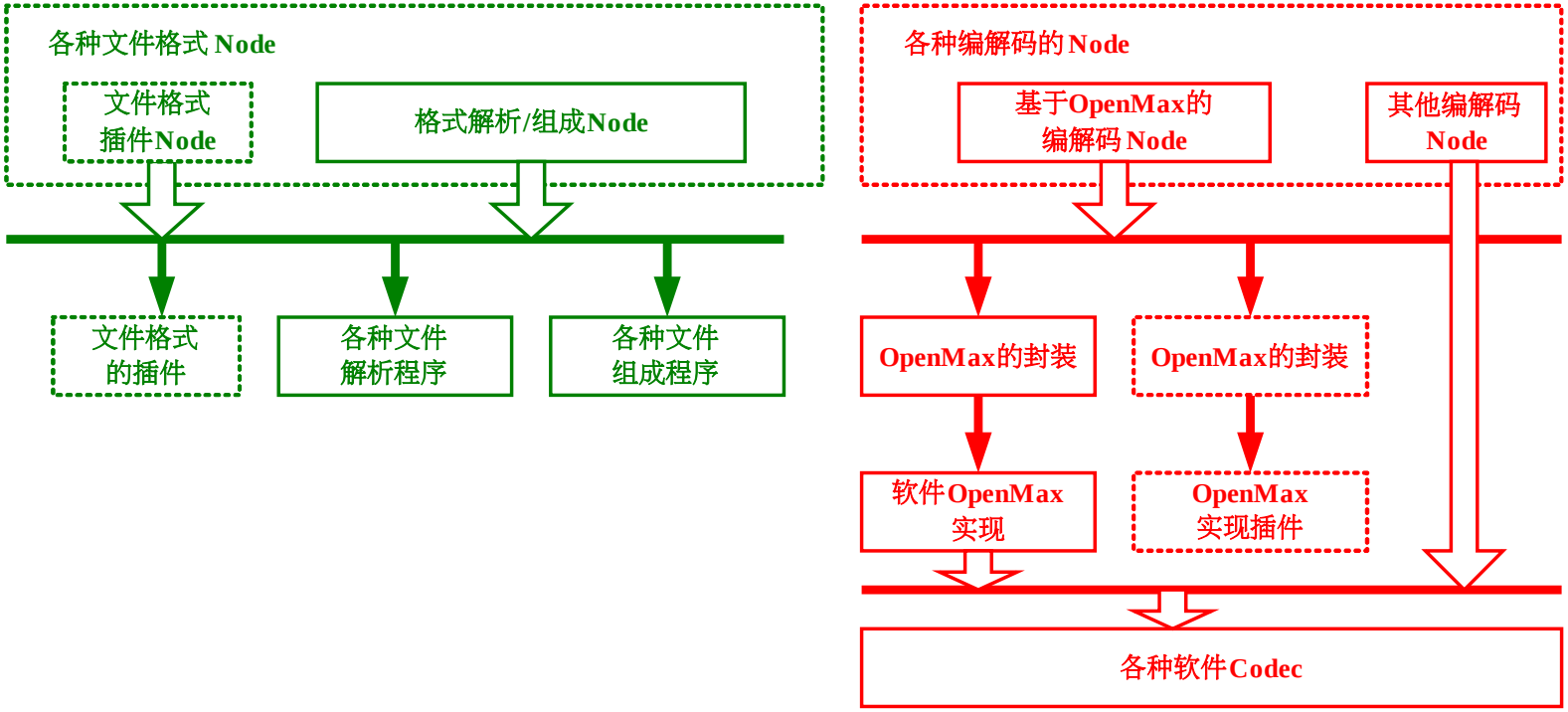
3.5 OpenCore Node 介绍

Node 是 OpenCore 中基本的功能模块，OpenCore 本身提供了一些 Node，也可以由上层软件来实现。本身提供的 Node 在 OpenCore 的 nodes 目录中。

OpenCore 的 Node 主要分成三个类型：编解码的 Node，文件格式的 Node，输入输出模块的 Node。

3.6 OpenCore 的功能扩展

OpenCore 本身提供了很强大的功能，在使用的过程中，还可以对 OpenCore 进行扩展。在扩展的使用，一般是基于 OpenCore 的框架为其增加固定的插件。插件主要一般可以做成 Node 的形式。



3.6 OpenCore 的功能扩展

编解码相关的 **Node** 包括：

pvomxbasedecnode， pvomxaudiodecnode， pvomxvideodecnode， pvomxencnode 等。

文件格式的 **Node** 包括：

pvwavffparsernode ，
pvaacffparsernode， pvamrffparsernode， pvmp3ffparsernode， pvmp4ffparsernode， pvvideoparsernode， pvmp4ffcomposernode 等。

输入输出的 **Node** 包括：

pvmediainputnode， pvmediaoutputnode， pvdummyinputnode， pvdummyoutputnode， pvfileoutputnode， pvdownloadmanagernode 等。

此外，还包括 pvsocketnode 、 pvdownloadmanagernode 等其他功能的 Node 。

3.6 OpenCore 的功能扩展

MediaIO 在 pvmi/pvmf/include 目录中的 PvmiMIOControl.h 和 pvmi_media_transfer.h 头文件中实现定义，在实现的过程中只需要继承和构建其中的接口，然后由框架最终实现成为 Node 在 OpenCore 系统中使用。事实上， MediaIO 是对 Node 的一种封装，封装成为多媒体的输入输出环节。

3.7 OpenCore Player 介绍

OpenCore 的 Player 的编译文件是 pvplayer/Android.mk，将生成动态库文件 libopencoreplayer.so。这个库包含了两方面的内容：一方是 Player 的 engine（引擎），一方面是为 Android 构件的 Player，这实际上是一个适配器（adapter）。engine 的路径是 engine/player； adapter 的路径是 android。

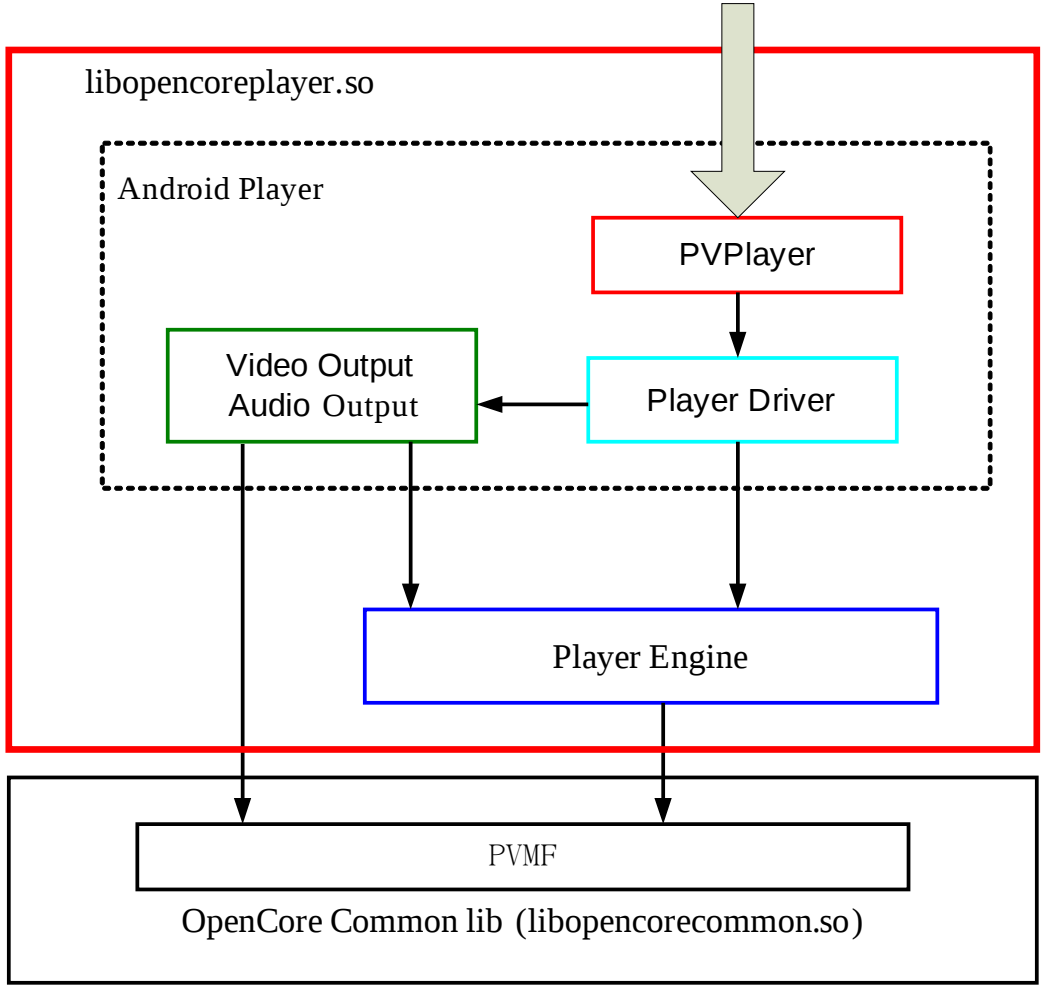
3.7 OpenCore Player 介绍

libopencoreplayer.so 中包含了以下内容：

- 一些解码工具；
- 文件的解析器（MP4）；
- 解码工具对应的 Node；
- Player 的引擎部分（编译文件： engines/player/Android.mk）；
- 为 Android 构建的 Player 适配器（编译文件： android/Android.mk）；
- 识别工具（pvmi/recognizer）；
- 编解码工具中的 OpenMAX 部分（codecs_v2/omx）；
- 对应插件 Node 的注册。

libopencoreplayer.so 中的内容较多，其中主要为各个文件解析器和解码器， PVPlayer 的核心功能在 engines/player/Android.mk 当中；而 android/Android.mk 的内容比较特殊，它是在 PVPlayer 之上构建的一个为 Android 使用的播放器。

3.7 OpenCore Player 介绍



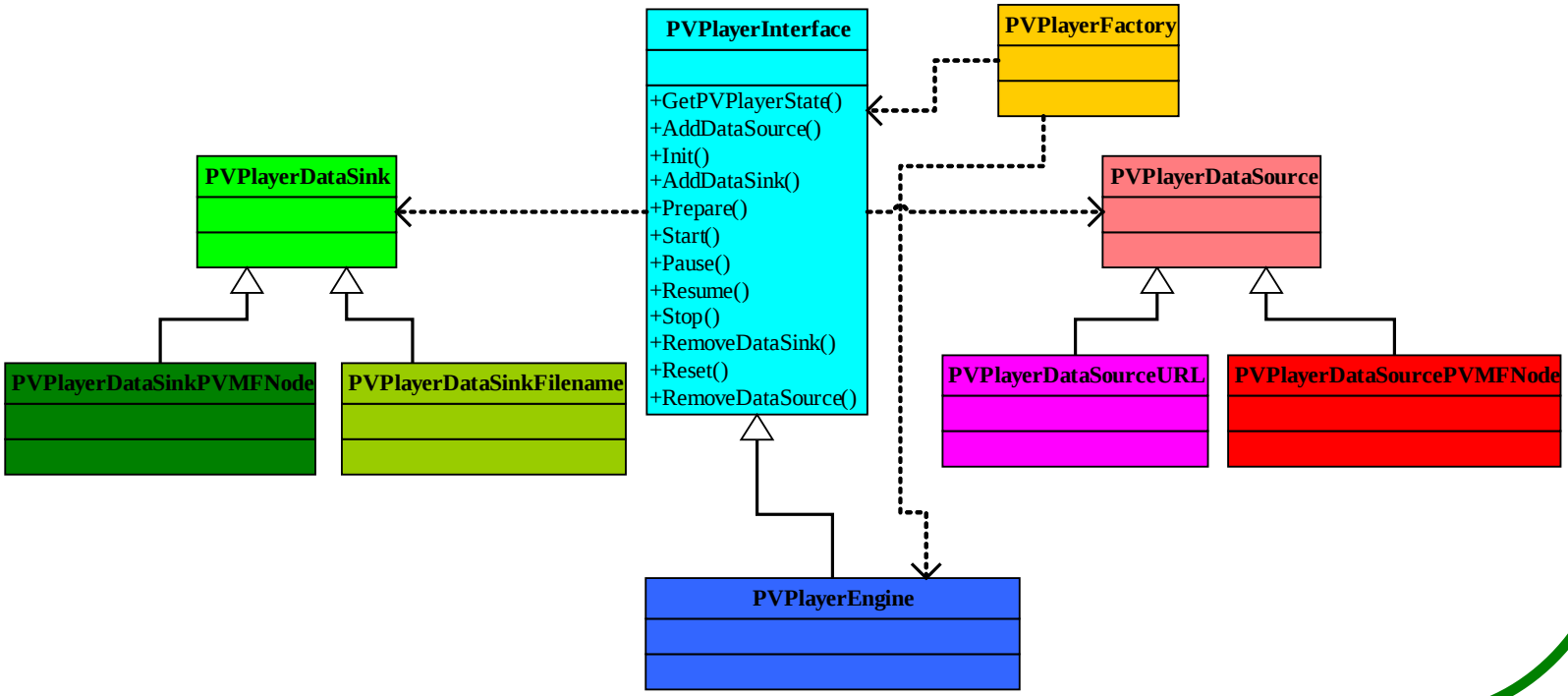
组成部分：

- 播放器核心引擎
- 适配器（PVPlayer）
- Video 输出环节
- Audio 输入环节

3.7 OpenCore Player 介绍

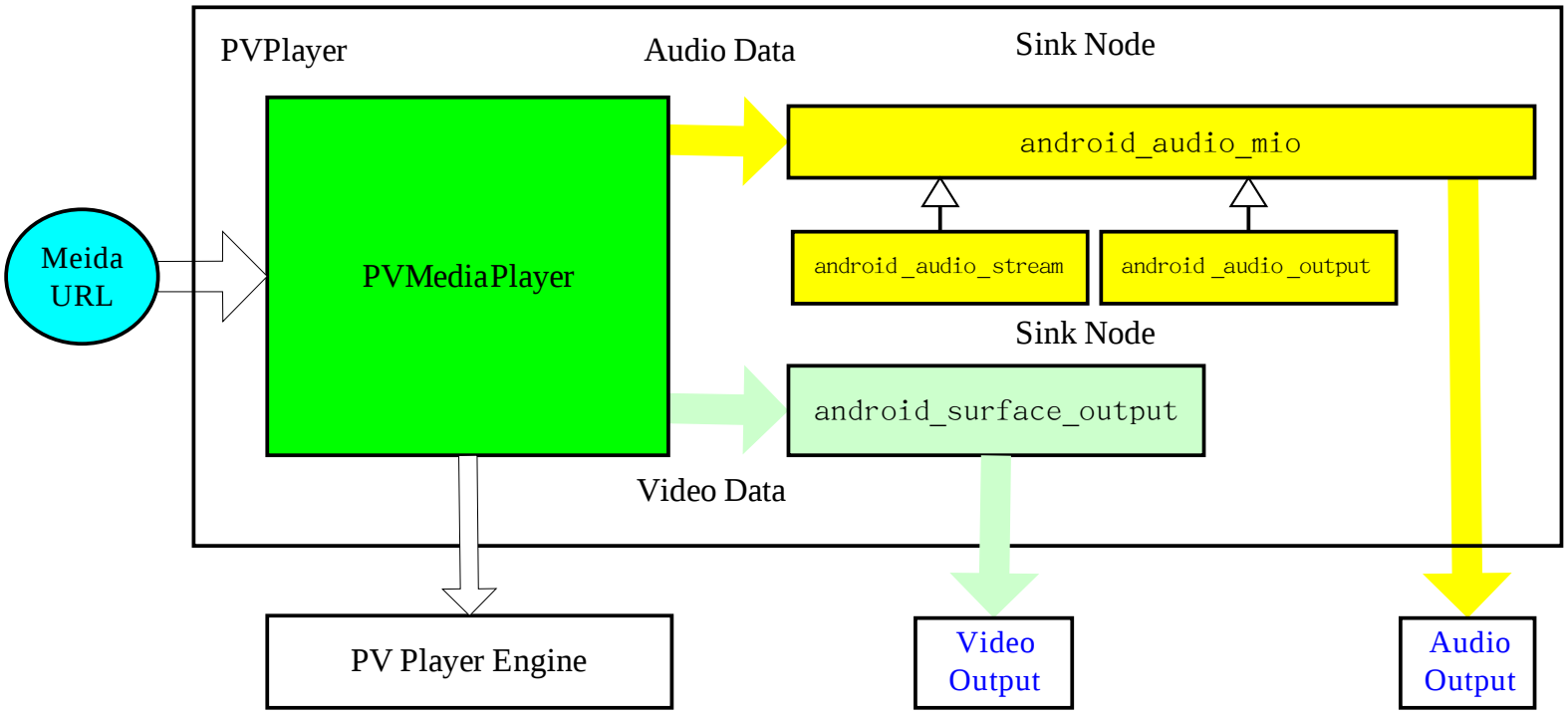
OpenCore 的 Player Engine 具有清晰明确的接口，在这个接口上，不同的系统可以根据情况实现不同的 Player 。位于 OpenCore 中的 engines/player/ 目录下，其中， engines/player/include 目录中保存的是接口头文件， engines/player/src 目录中保存是源文件和私有头文件 。

Player Engine 的类结构：



3.7 OpenCore Player 介绍

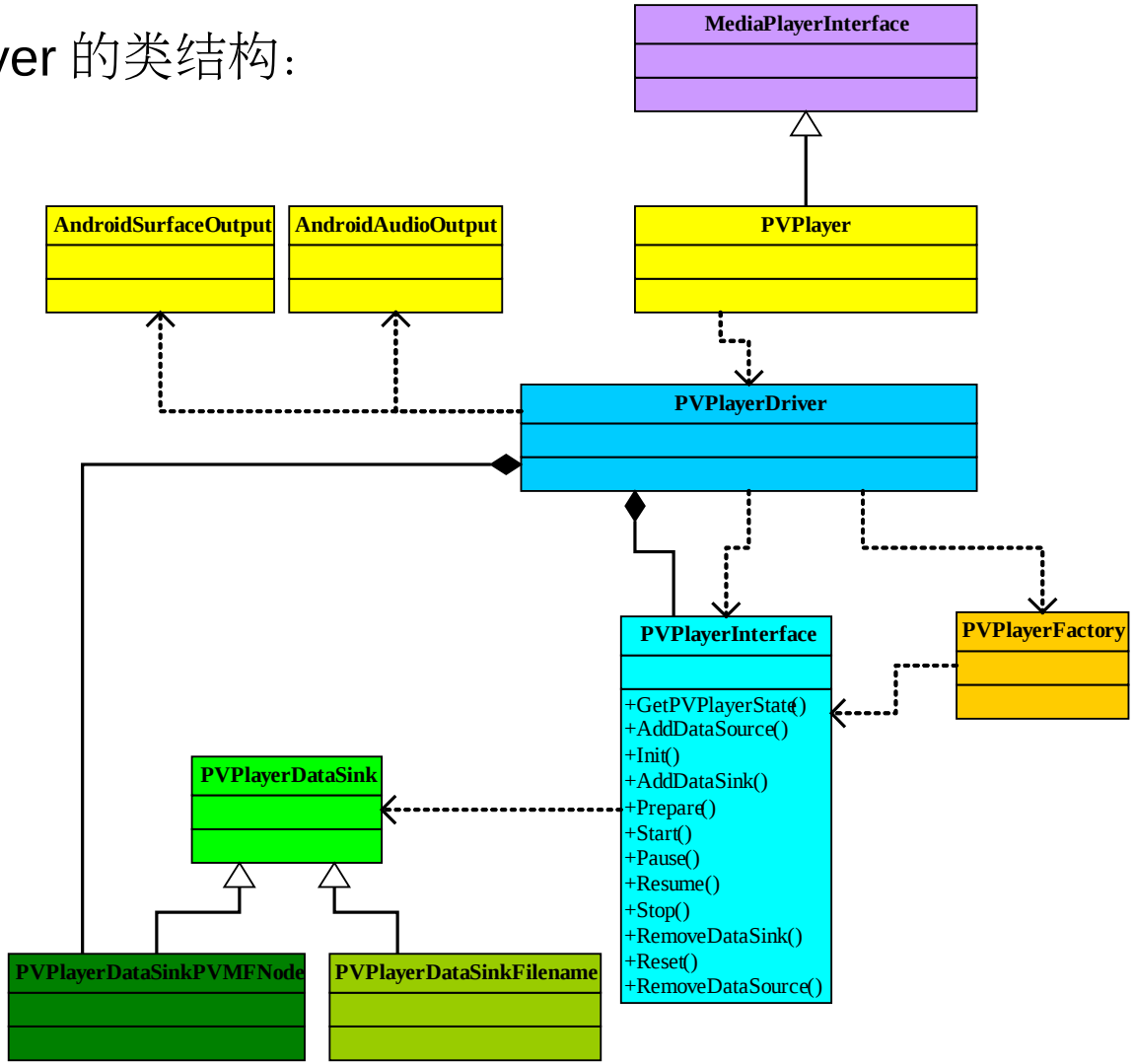
PVPlayer 的结构：



Sink Node 接受上一个 Node 写的动作

3.7 OpenCore Player 介绍

PVPlayer 的类结构：



3.7 OpenCore Player 介绍

OpenCore 的 Author 的编译文件是 pvauthor/Android.mk，将生成动态库文件 libopencoreauthor.so。与 Player 类似，这个库包含了两方面的内容：一方是 Author 的 engine（引擎），一方面是为 Android 构件的 Author，这实际上是一个适配器（adapter）。engine 的路径是 engine/author； adapter 的路径是 android/author。

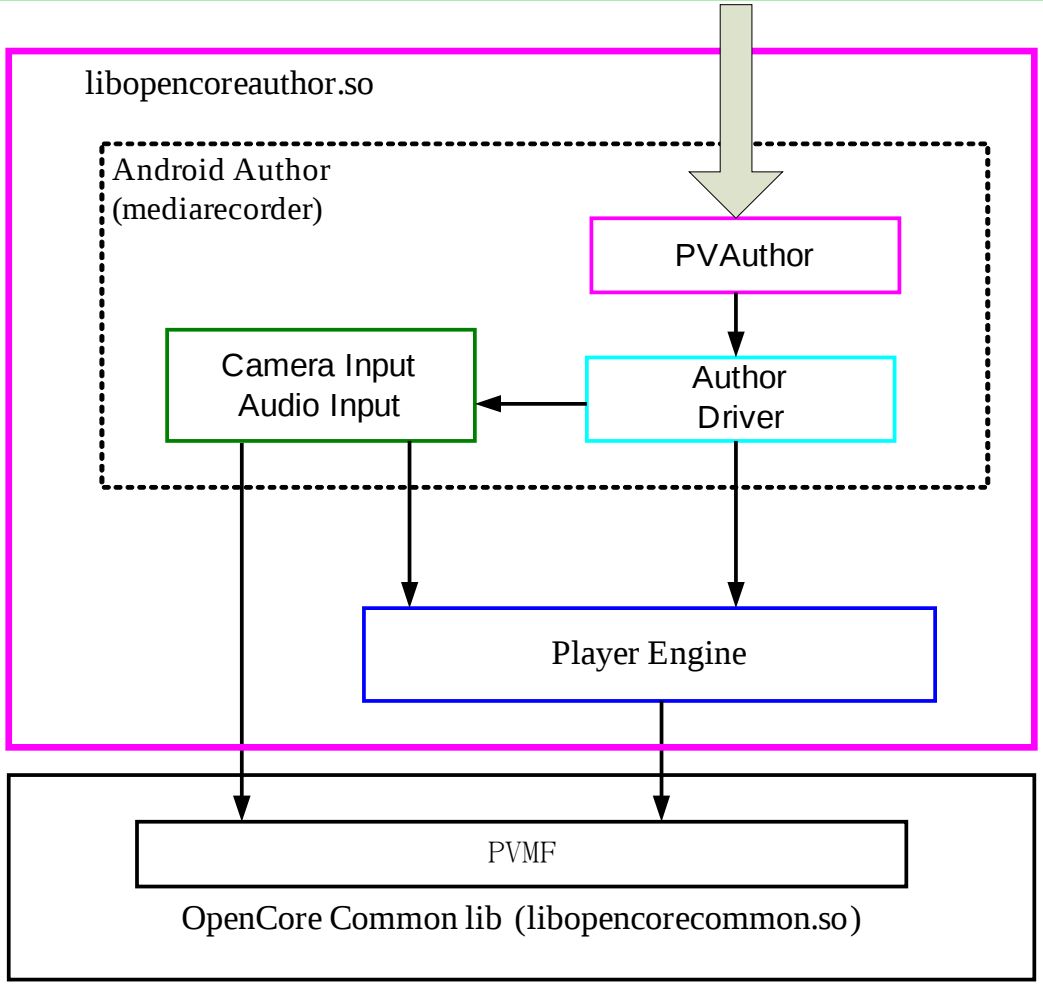
3.7 OpenCore Player 介绍

libopencoreauthor.so 中包含了以下内容:

- 一些编码工具（例如，视频流 H263 、 H264 和音频流 Amr ）；
- 文件的组成器（ MP4 ）；
- 编码工具对应的 Node ；
- 表示媒体输入的 Node
  (编译文件： nodes/pvmediainputnode/Android.mk ）；
- Author 的引擎部分
  (编译文件： engines/author/Android.mk ）；
- Android 的 Author 适配器
  (编译文件： android/author/Android.mk ）。

libopencoreauthor.so 中主要内容为各个文件编码器和文件组成器， PVAuthor 的核心功能在 engines/author/Android.mk 中，而 android/author/Android.mk 是在 PVAuthor 之上构建的一个为 Android 使用的媒体记录器。

3.8 OpenCore Author 介绍



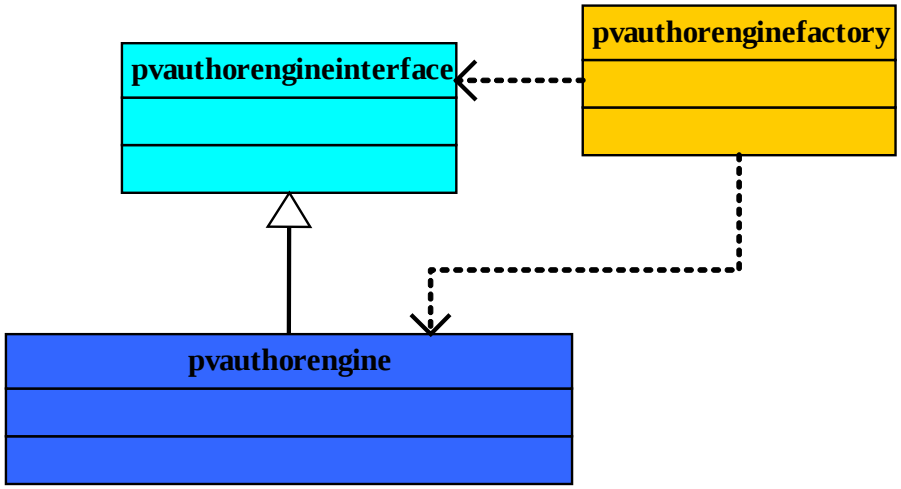
组成部分：

- 作者核心引擎
- 适配器（PVAuthor）
- Camera 输入环节
- Audio 输出环节

3.8 OpenCore Author 介绍

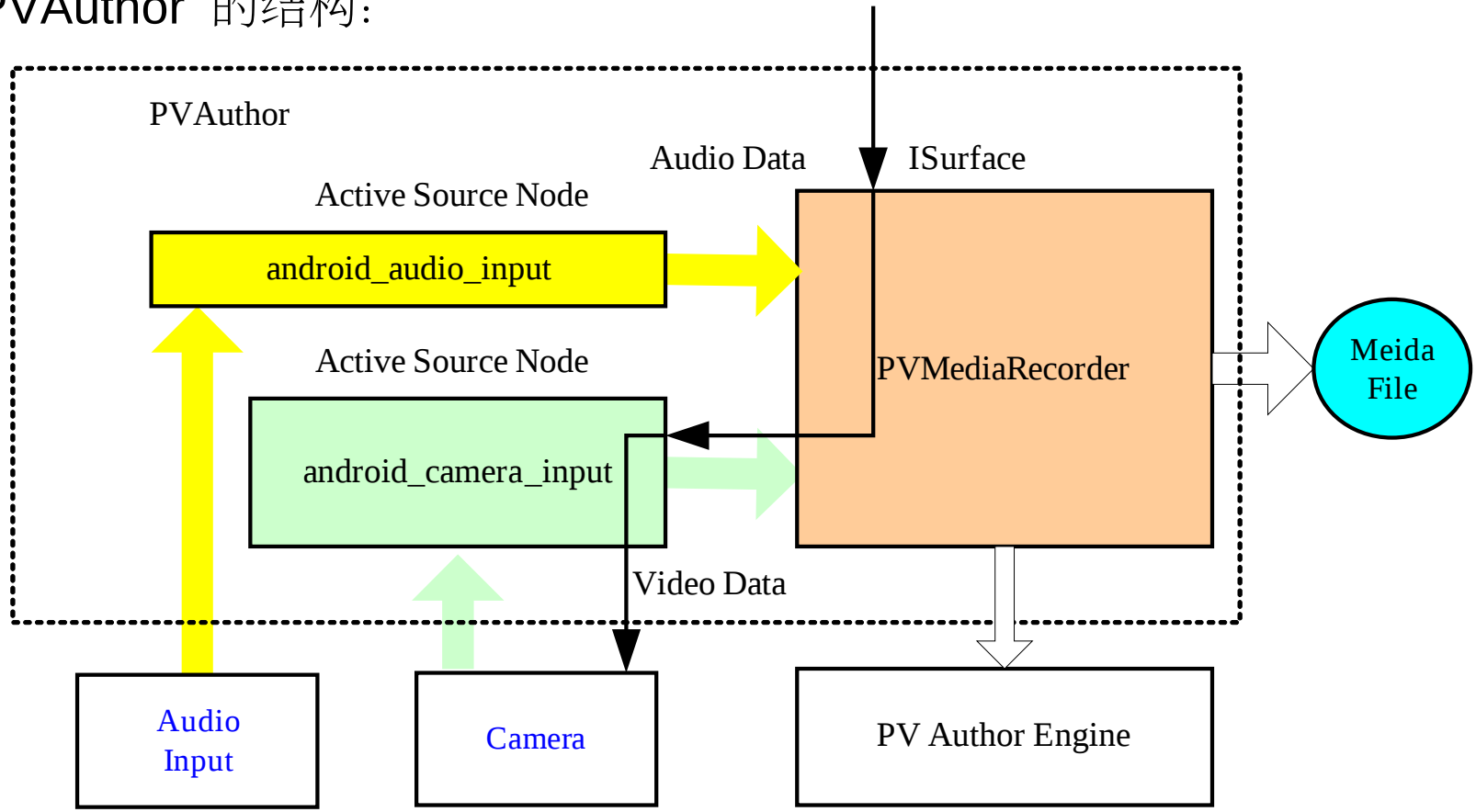
目录为 OpenCore 中的 engines/author/，是 Author 引擎目录，其主要包含 include 和 src 两个目录，头文件中的 pvauthorenginefactory.h 和 pvauthorengineinterface.h 两个文件为接口，源文件为主要的具体实现 pvauthorengine.cpp 。

Author Engine 的类结构：



3.8 OpenCore Author 介绍

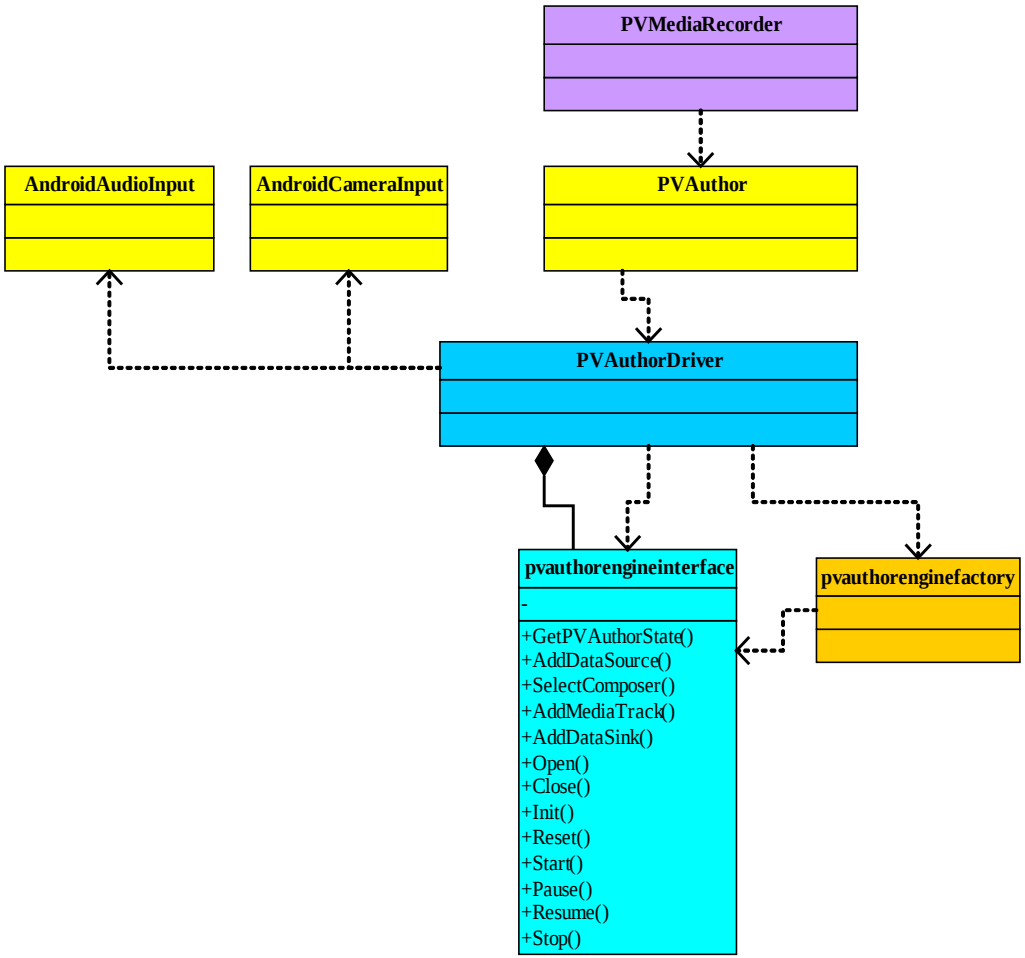
PVAuthor 的结构：



活动的数据源 Node 将主动写下一个环节。

3.8 OpenCore Author 介绍

PVAuthor 的类的结构：



附：Stagefright

Stagefright 是 Android Eclair 中新增的一个多媒体的实现。它是一个轻量级的多媒体实现。

Stagefright 头文件：
frameworks/base/media/libstagefright/include/
Stagefright 的实现：
frameworks/base/media/libstagefright/

附：Stagefright

Stagefright 是 Android Eclair 中新增的一个多媒体的实现。Stagefright 是一个轻量级的多媒体实现，主要功能基于 OpenMax 来实现，提供媒体播放等功能接口，为 Android 的框架层使用（如作为 mediaplayer 的实现）。

Stagefright 头文件：
frameworks/base/include/media/stagefright/
Stagefright 的实现：
frameworks/base/media/libstagefright/

谢谢！