

UNISOC
UDS710+UDX710
Android9.0 Camera
SW Customer Training

修改历史

版本号	日期	注释
V1.0	2020/05/15	初稿

Unisoc Confidential For hiar

文档信息



适用产品信息

UDS710_UDX710

适用版本信息

Android 9.0

关键字

Camera Hal, Structure, Software

Unisoc Confidential For hiar

Contents

1

Camera系统架构概览

2

Camera HAL & Drv子模块介绍

3

Camera Init/Preview/Video/Snapshot流程

4

Grab模块介绍

5

Dual Camera架构介绍

Contents

6

第三方图像后处理算法集成介绍

7

3A及ISP算法流程和逻辑

8

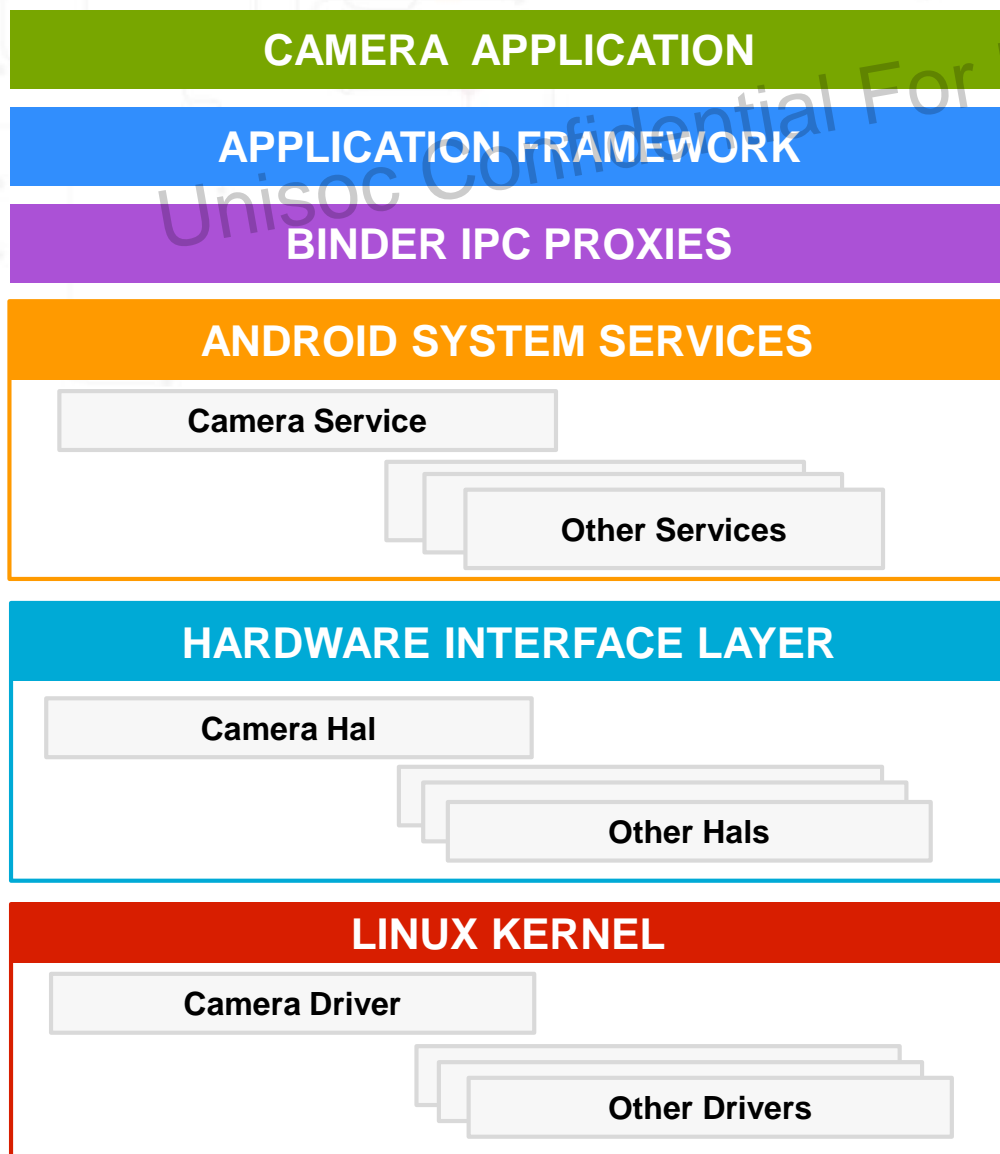
Camera Sensor驱动及参数配置

9

相机功耗和性能优化调试方法

10

常见问题分析方法



主要代码路径：

1)Camera Service:

frameworks/av/services/camera/libcameraservice /

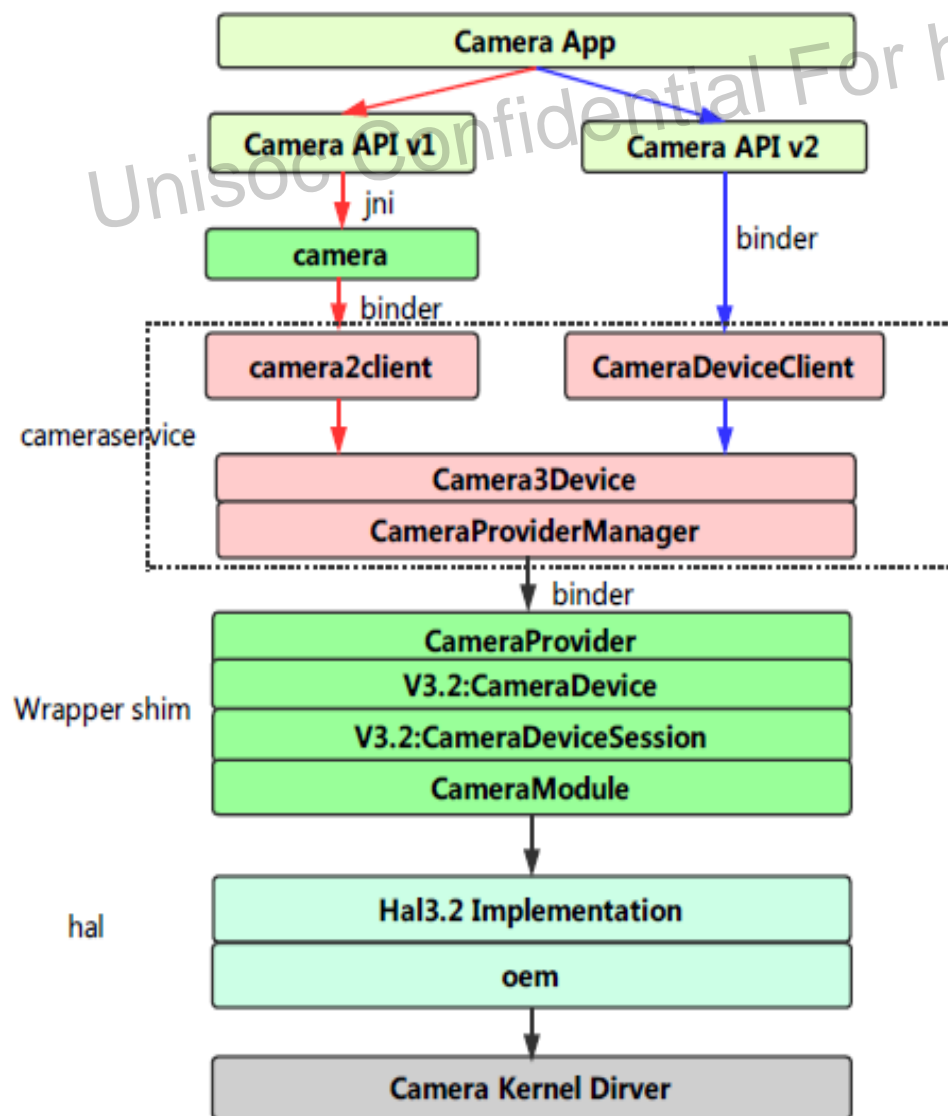
2)Camera Hal:

/vendor/sprd/modules/libcamera

3)Camera Driver:

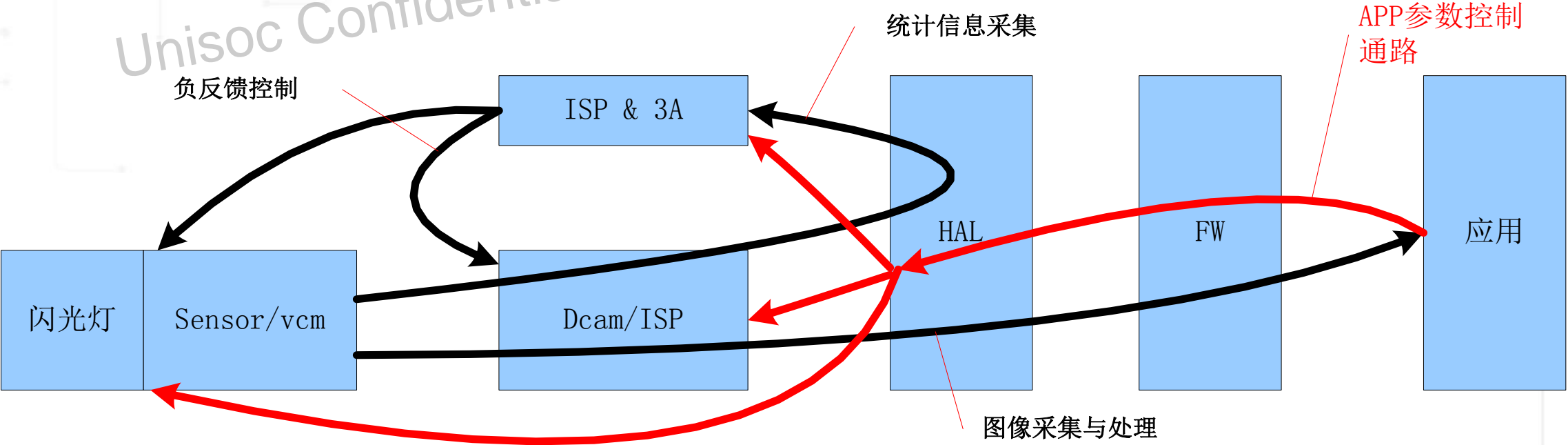
/vendor/sprd/modules/libcamera/kernel_module

Android7.0及之前Camera软件架构



Camera软件架构: Android8.0及之后 (HIDL)

Unisoc Confidential For hiar



Contents

1

Camera系统架构概览

2

Camera HAL & Drv子模块介绍

3

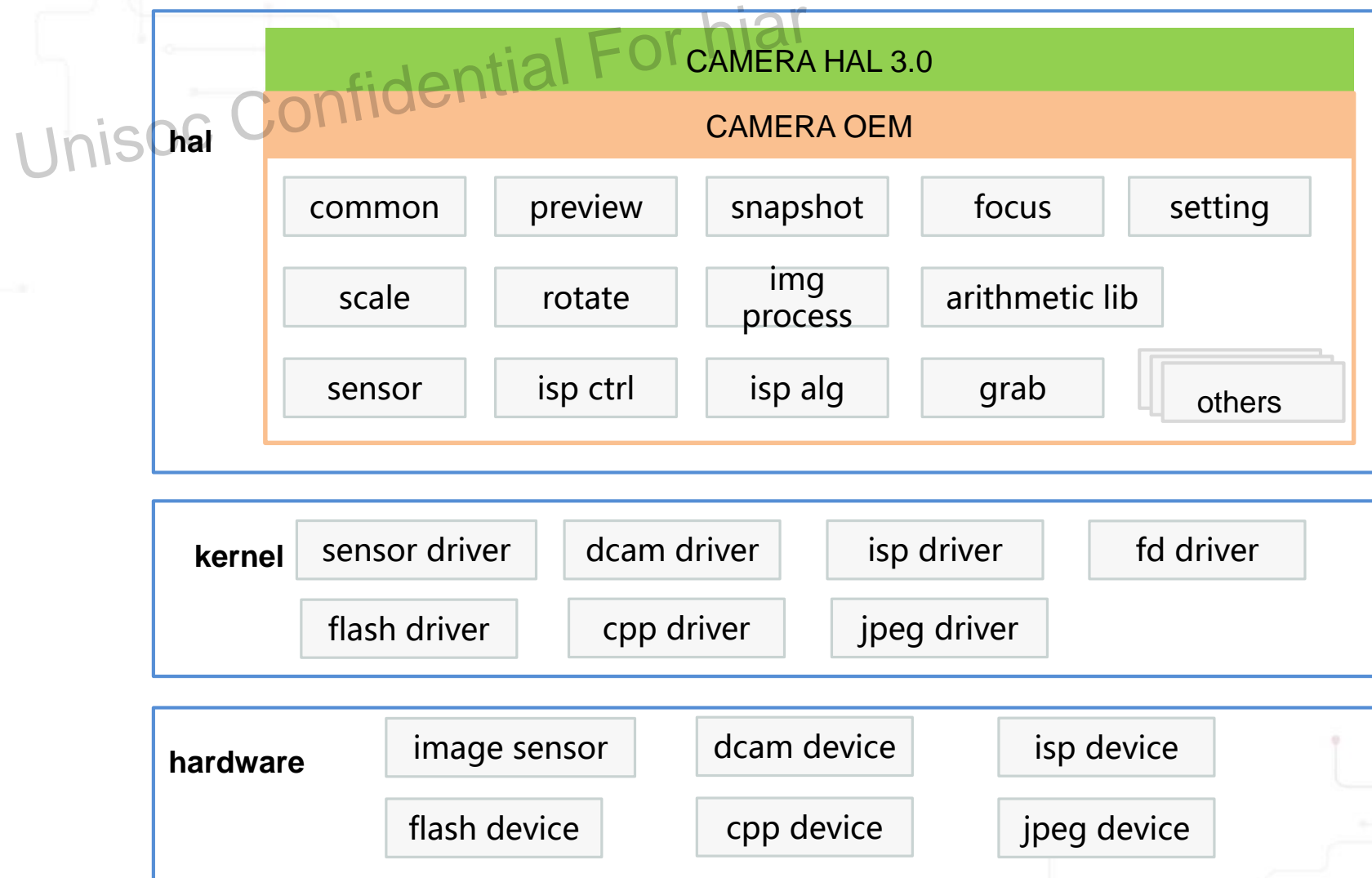
Camera Init/Preview/Video/Snapshot流程

4

Grab模块介绍

5

Dual Camera架构介绍

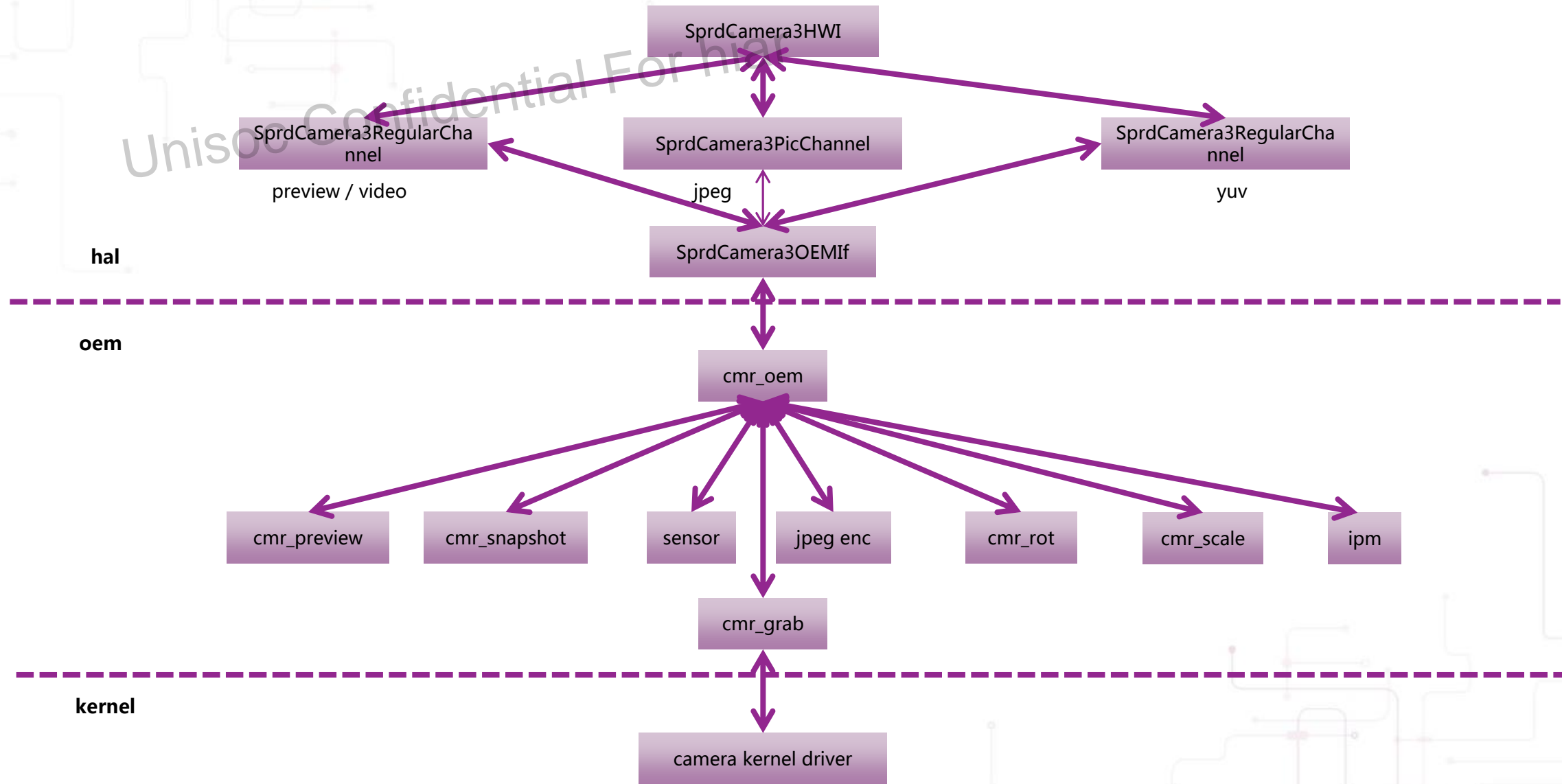


User: vendor/srpd/modules/libcamera

Kernel: /vendor/srpd/modules/libcamera/kernel_module

idx	子模块	文件路径
1	HAL	libcamera/hal3_2v6 libcamera/common
2	OEM	libcamera/oem2v6
3	图像算法	libcamera/arithmetic
4	ISP middle ware	libcamera/camdrv/isp2.6
5	3A	libcamera/ispalg/isp2.x
6	Sensor user drv	libcamera/sensor
7	ISP kernel drv	/libcamera/kernel_module/core/isp2.6
8	Sensor kernel drv	/vendor/srpd/modules/libcamera/kernel_module/sensor kernel4.14/arch/arm64/boot/dts/sprd

Camera Hal & Driver子模块关系



Contents

1

Camera系统架构概览

2

Camera HAL & Drv子模块介绍

3

Camera Init/Preview/Video/Snapshot流程

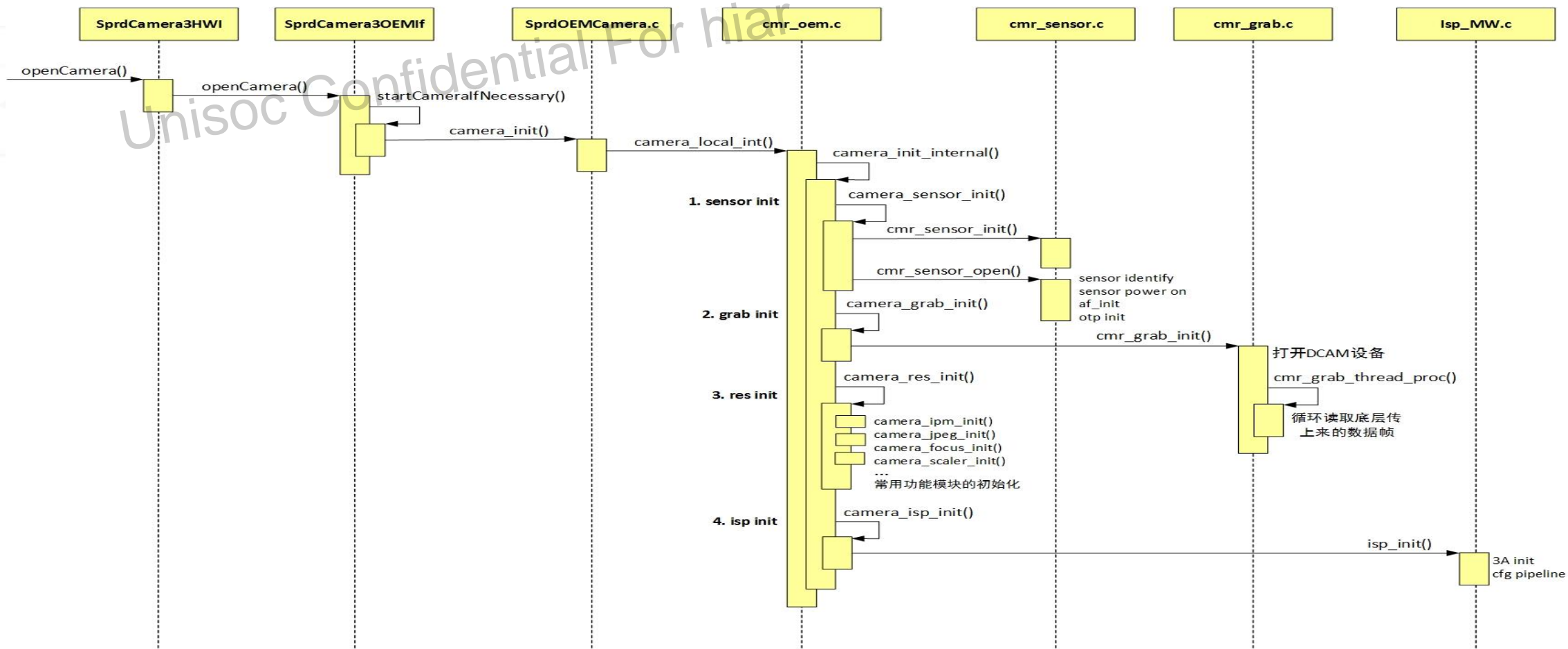
4

Grab模块介绍

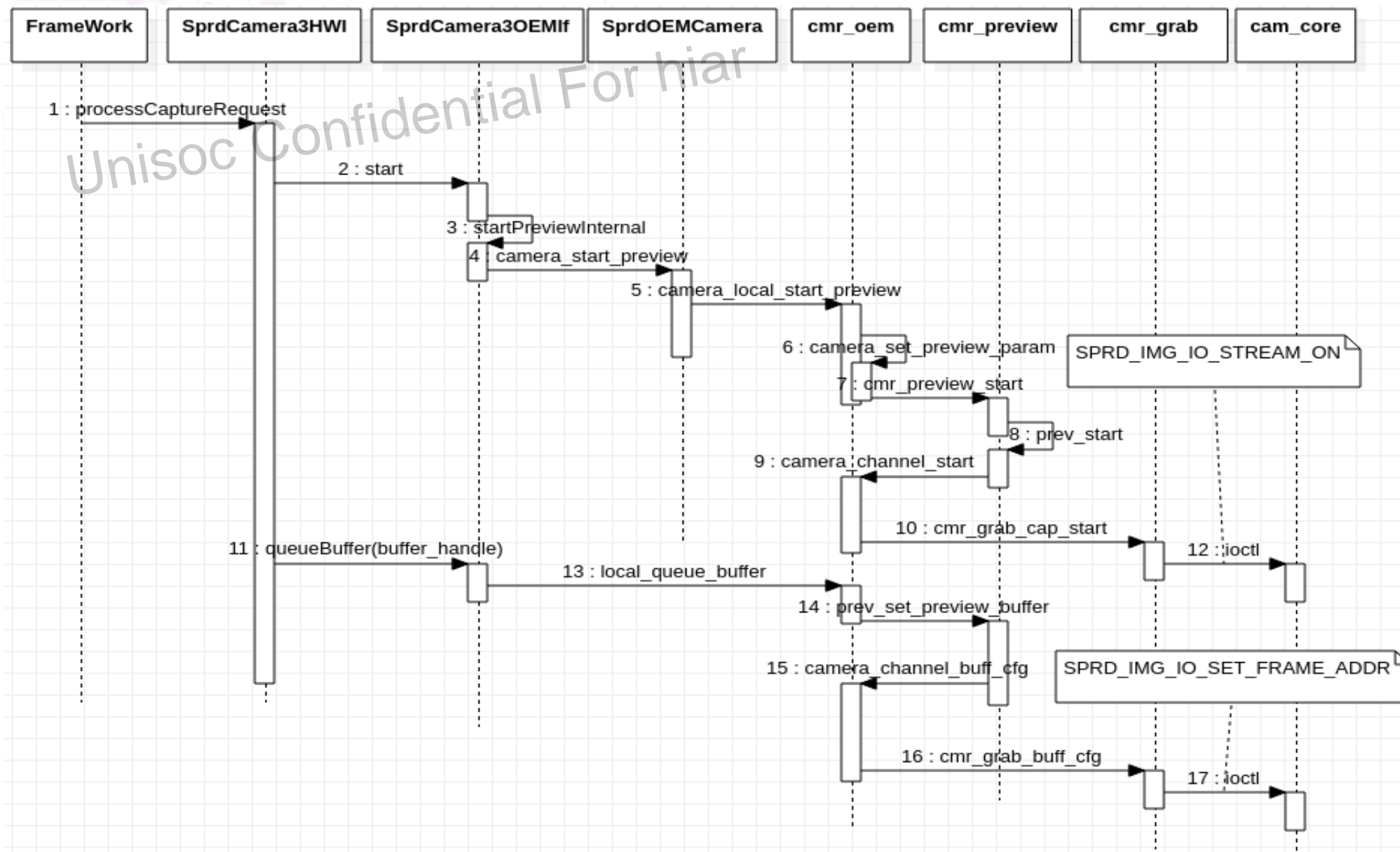
5

Dual Camera架构介绍

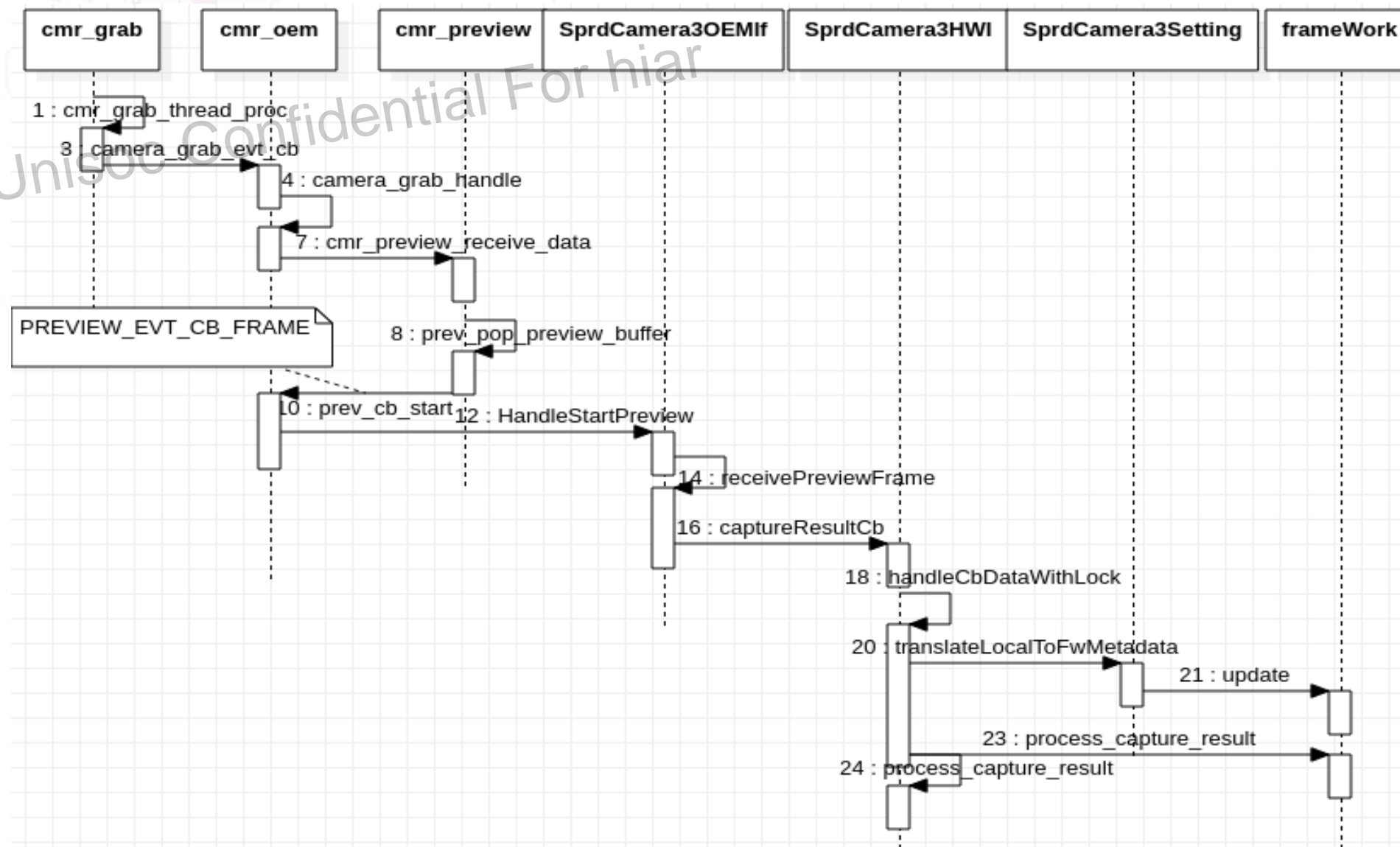
Camera Init流程

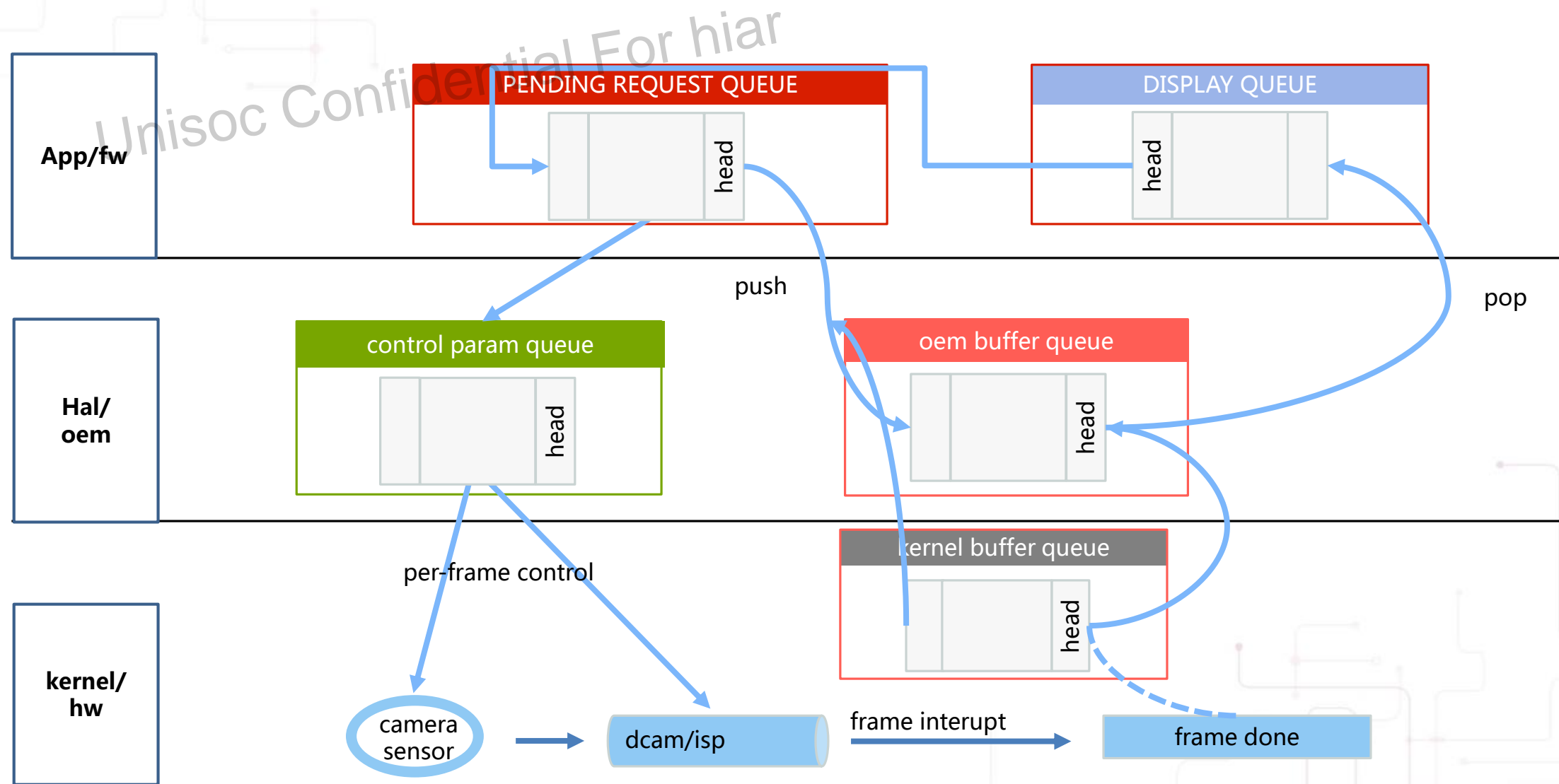


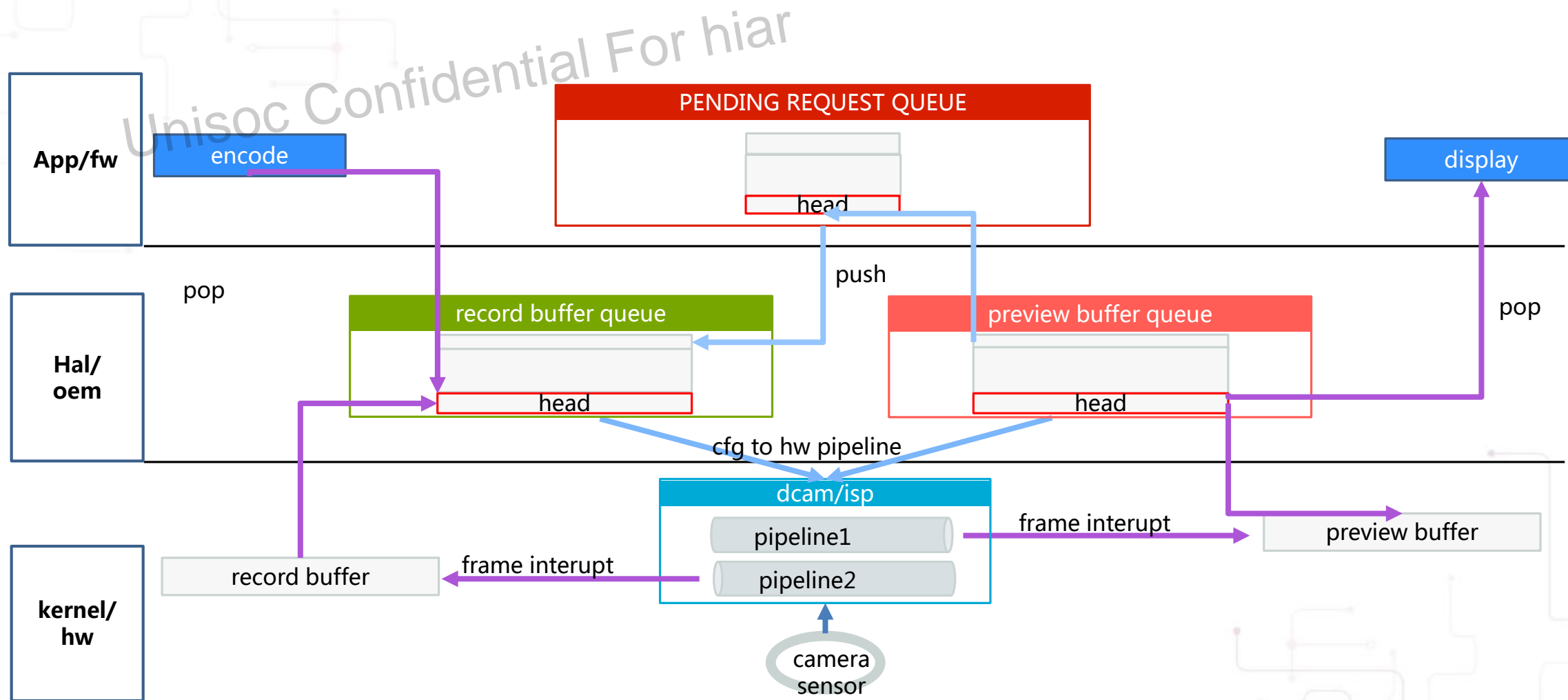
预览模式下第一帧数据的传递流程



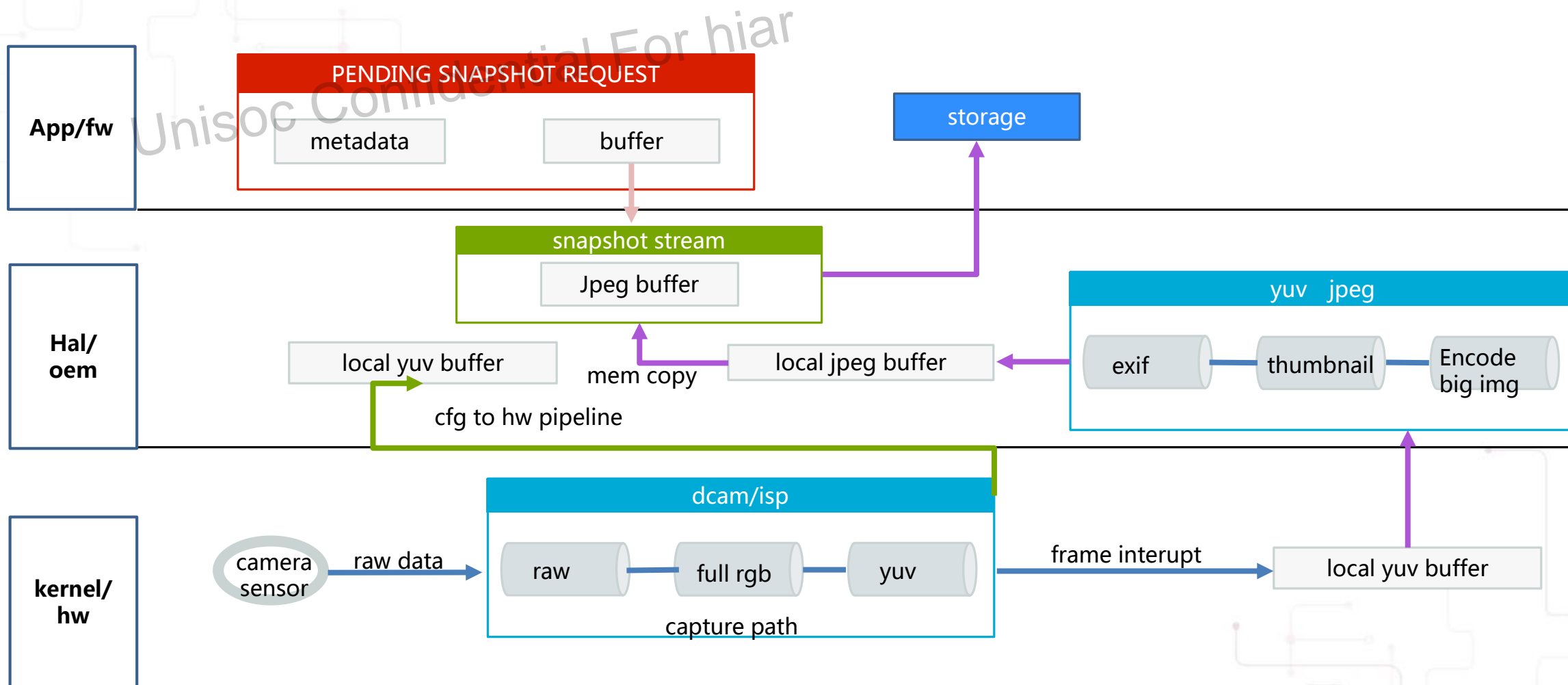
预览模式下第一帧数据的传递流程



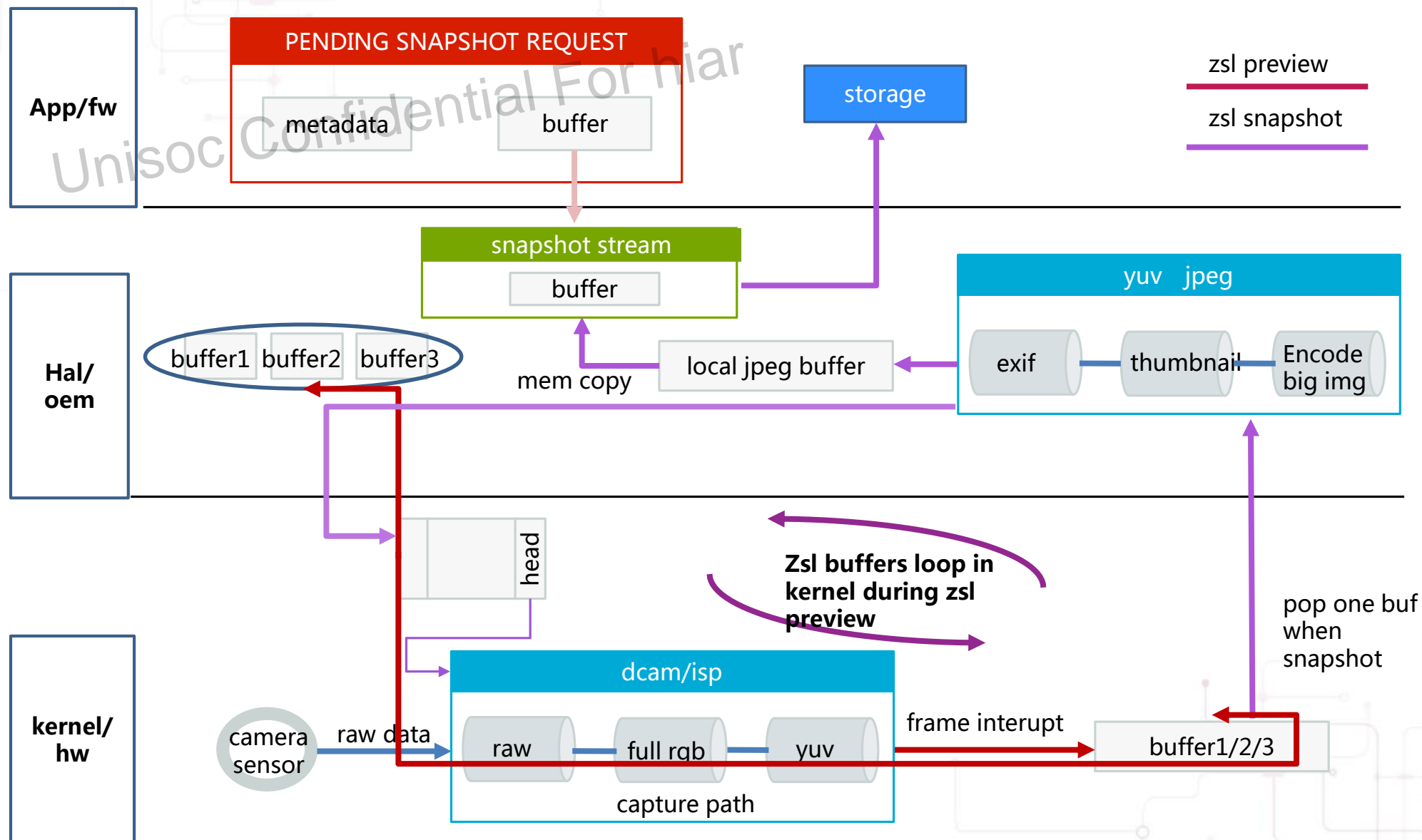


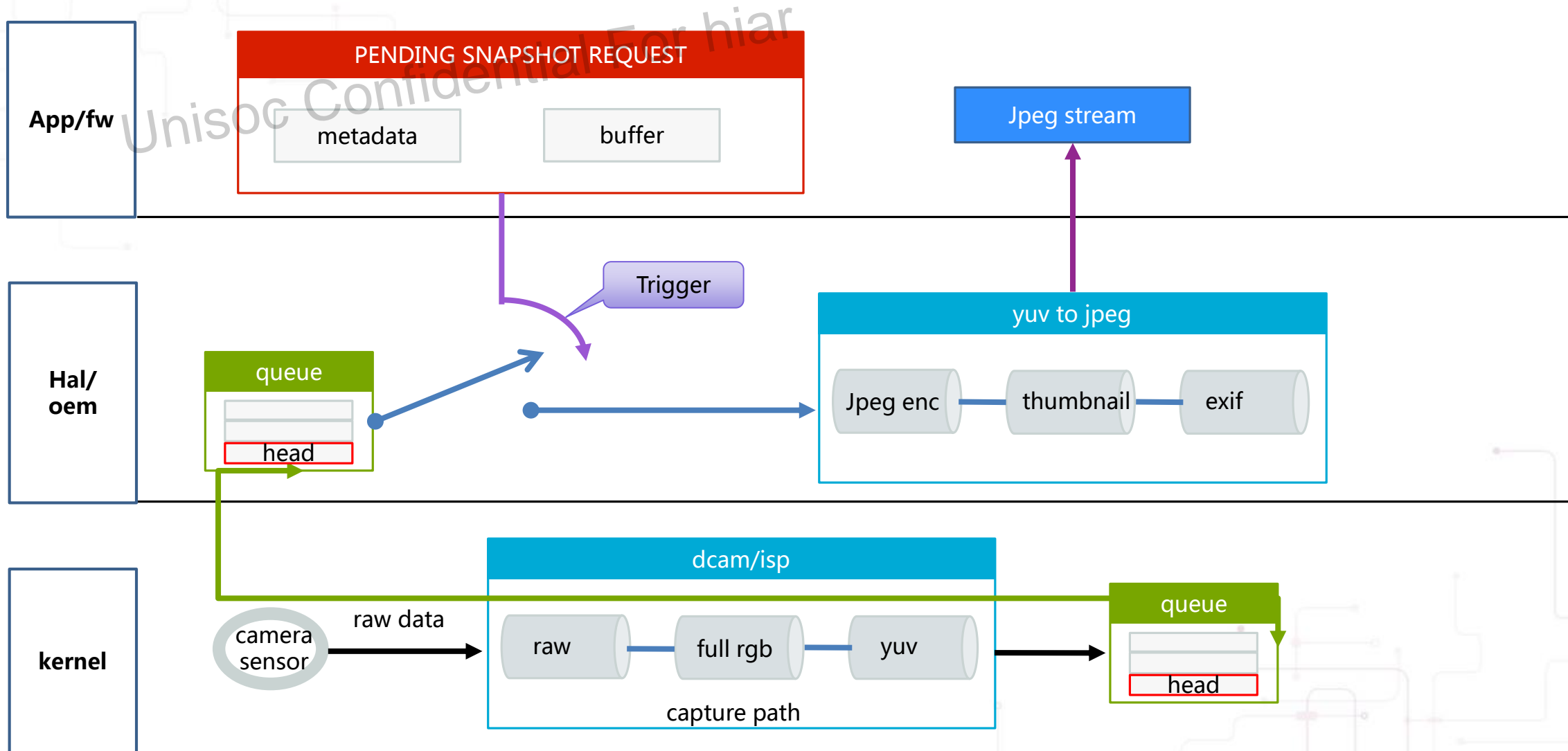


Capture数据流 (non-ZSL)



Capture数据流 (ZSL)





Contents

1

Camera系统架构概览

2

Camera HAL & Drv子模块介绍

3

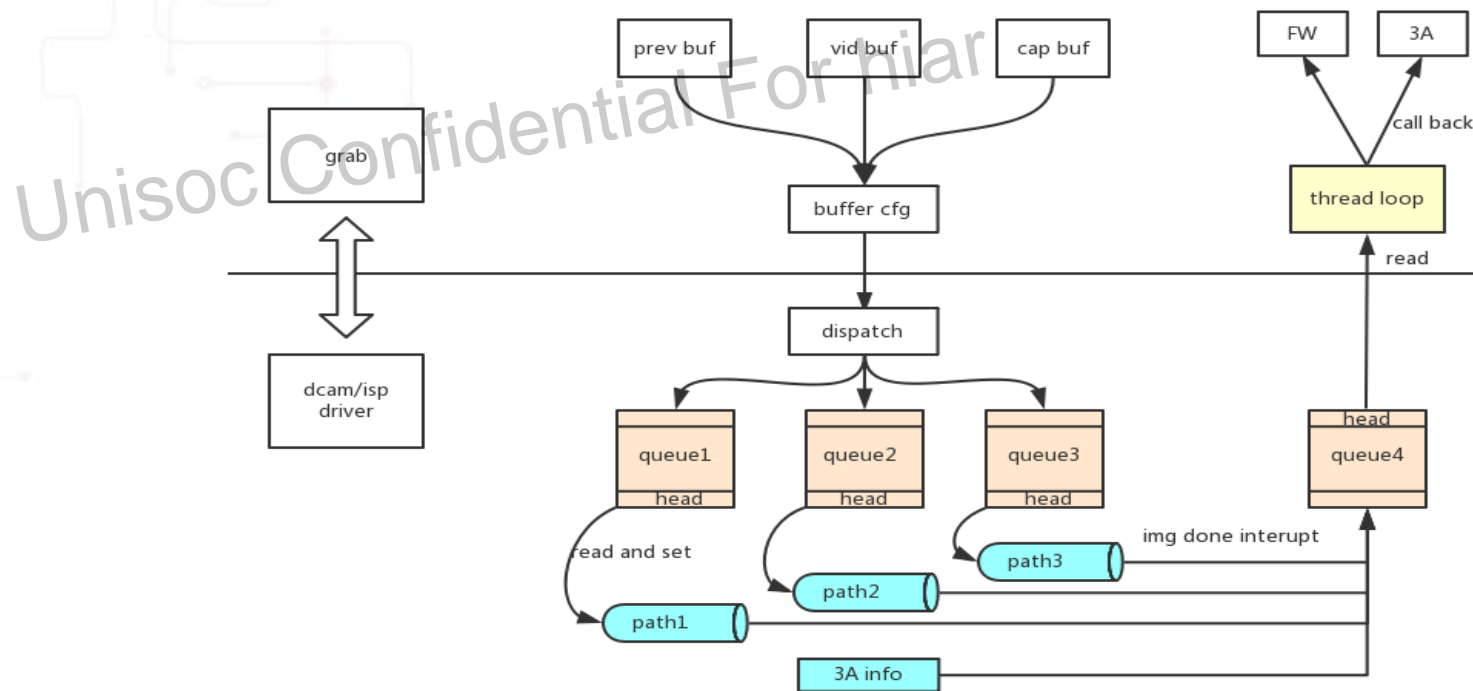
Camera Init/Preview/Video/Snapshot流程

4

Grab模块介绍

5

Dual Camera架构介绍



摘要：

Δ grab模块是oem层与dcam/isp设备的接口, 负责用户空间与设备驱动的交互。

Δ grab将上层下发的控制流以及buffer配置给驱动, 并单起一个线程监控驱动中统计信息和帧中断事件, 将读取到的统计信息回调给isp模块做3a运算, 将读取到的buffer上报oem/hal做进一步的处理。

Δ 不管是一路还是多路buffer, 上层都是通过grab的同一接口透传给驱动, 驱动层会根据每个buffer的channel id将其写入不同的队列中,并在帧中断处理中设置给对应的硬件path。驱动会将所有的帧和统计信息写入同一队列中,grab的监控线程会读取这些数据并分发处理。

Contents

1

Camera系统架构概览

2

Camera HAL & Drv子模块介绍

3

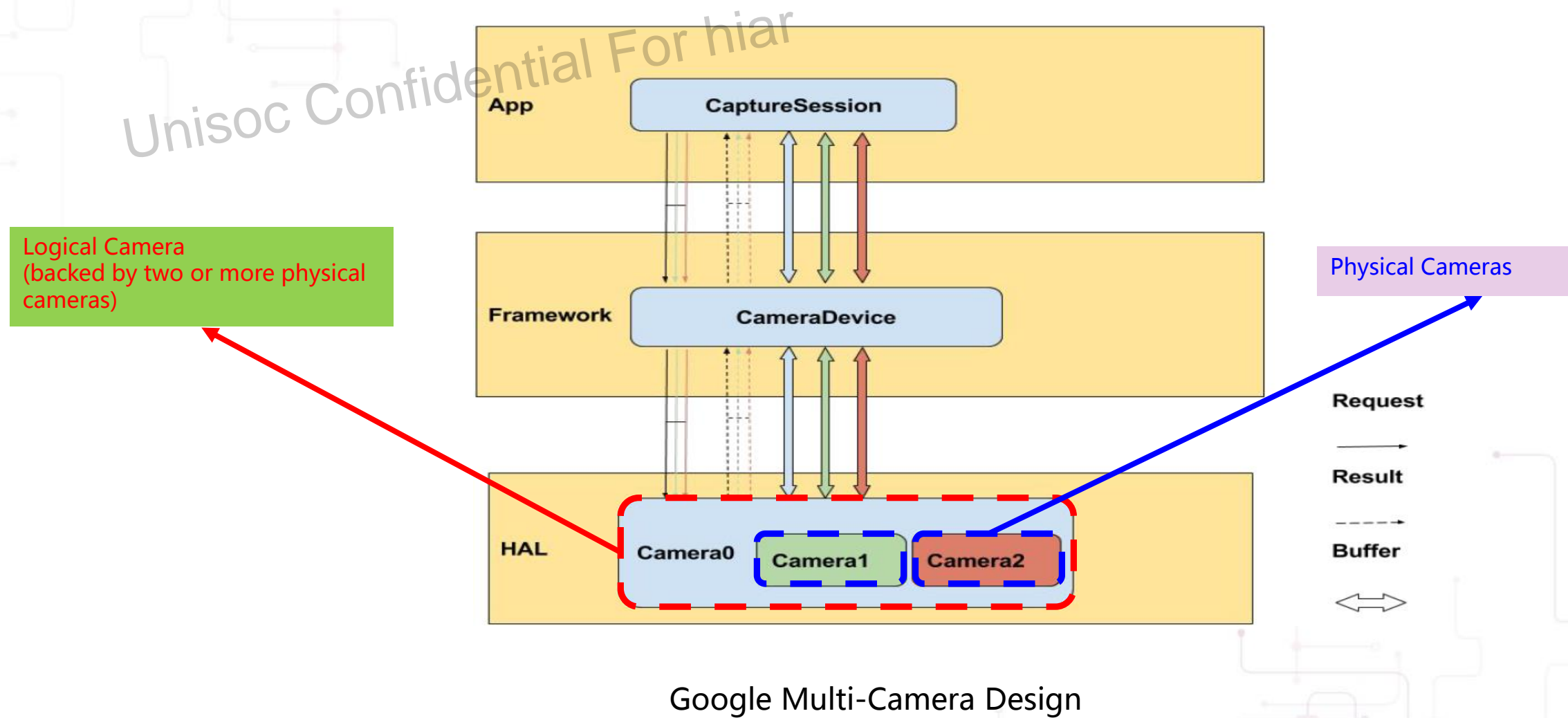
Camera Init/Preview/Video/Snapshot流程

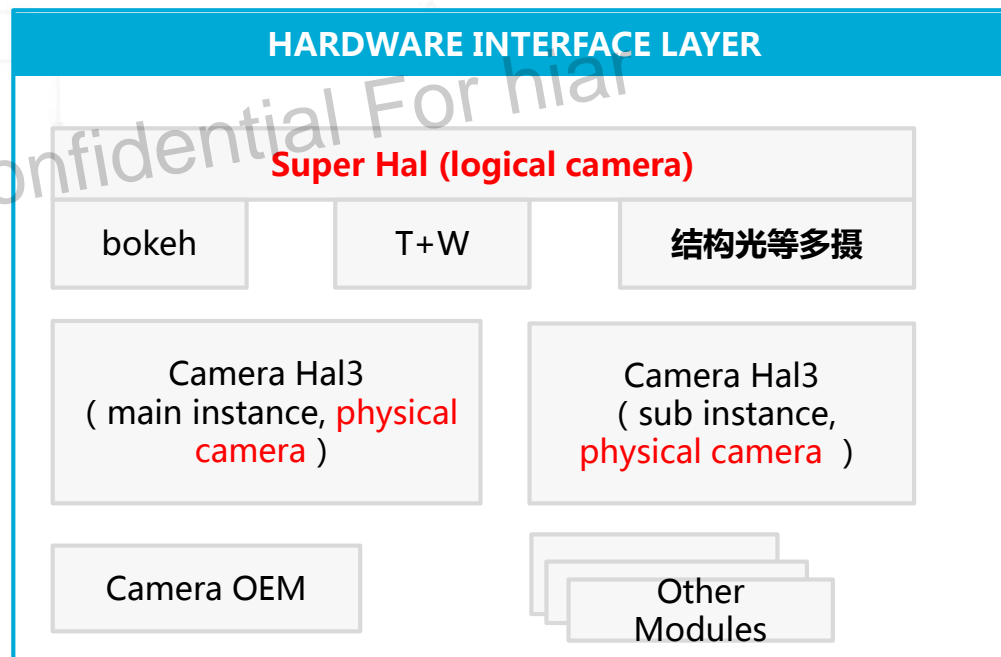
4

Grab模块介绍

5

Dual Camera架构介绍





Unisoc双摄软件架构 (Super HAL)

Super HAL代码路径：

9.0版本: /vendor/sprd/modules/libcamera/hal_common

双摄同步：

- 1.硬件级同步，通过Sensor Frame Sync信号实现；
- 2.软件级同步，通过时间戳实现图像数据同步和3A同步。

Contents

6

第三方图像后处理算法集成介绍

7

3A及ISP算法流程和逻辑

8

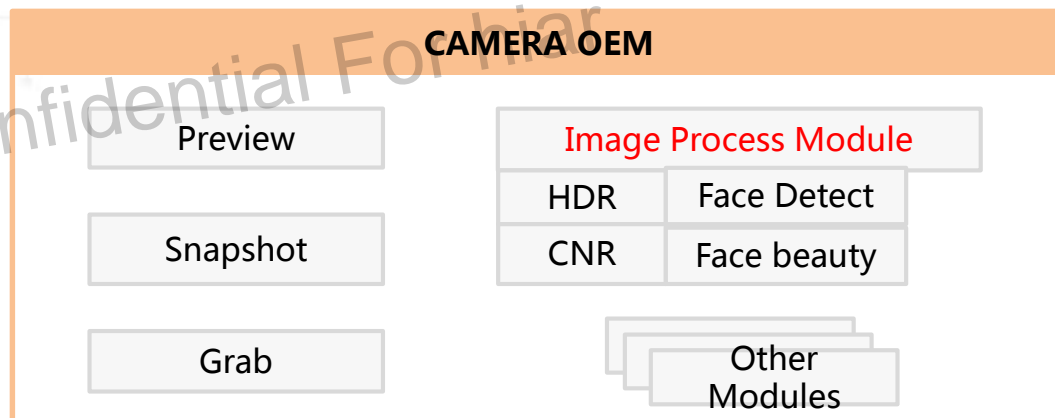
Camera Sensor驱动及参数配置

9

相机功耗和性能优化调试方法

10

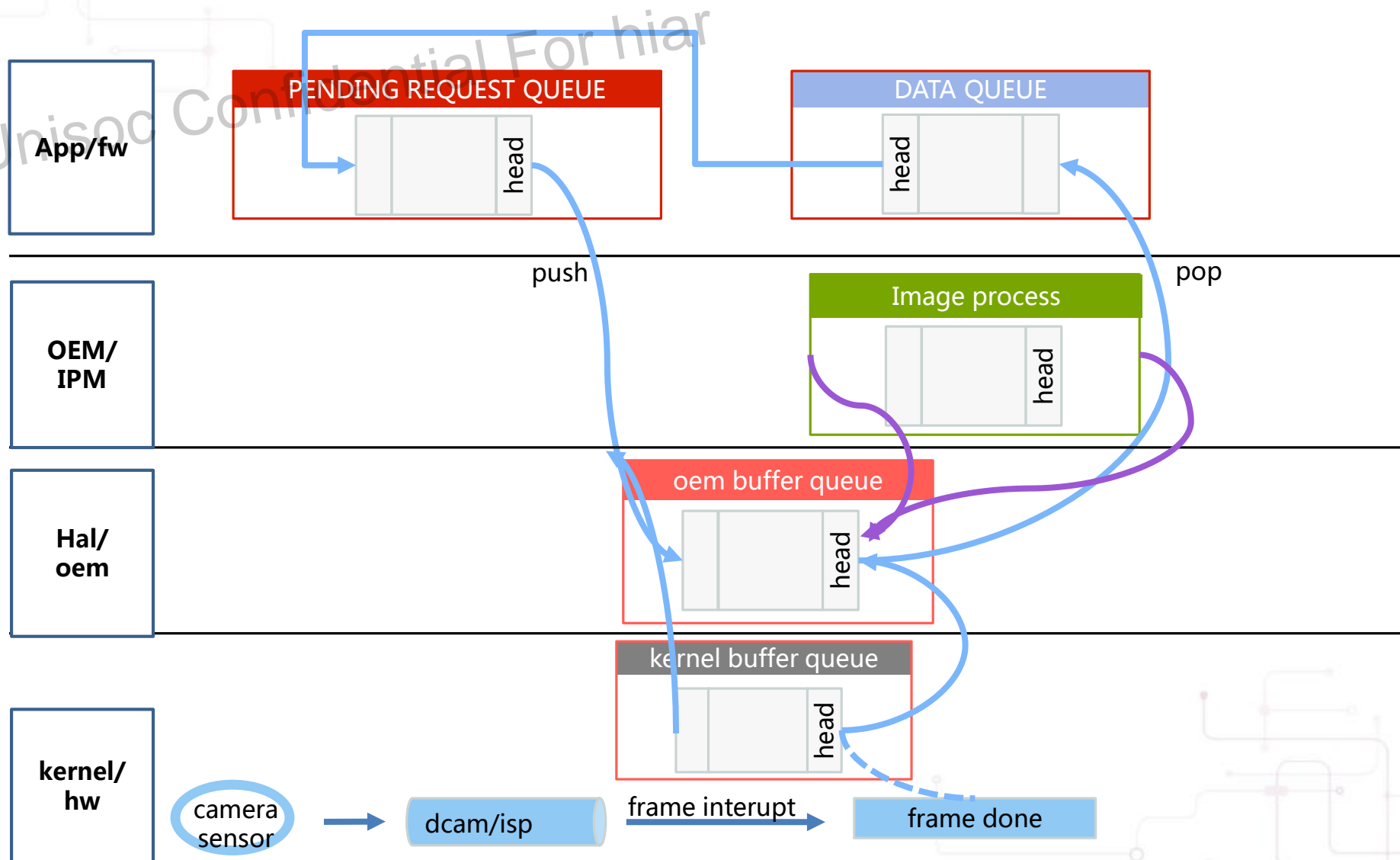
常见问题分析方法



图像后处理软件架构 (IPM)

IPM框架下集成第三方算法库步骤：

- 1 创建新文件（放入/vendor/sprd/modules/libcamera/oem2v6/src）
- 2 实现操作集：目前支持的操作有open, close, transfer_frame, pre_proc, post_proc，按需实现，不需要的置为空。
- 3 将操作集注册到IPM中：在/vendor/sprd/modules/libcamera/oem2v6/cmr_ipm.c中添加到class_type_tab[] 中。
- 4 在数据流的处理过程中添加调用新添加的处理操作。



Contents

6

第三方图像后处理算法集成介绍

7

3A及ISP算法流程和逻辑

8

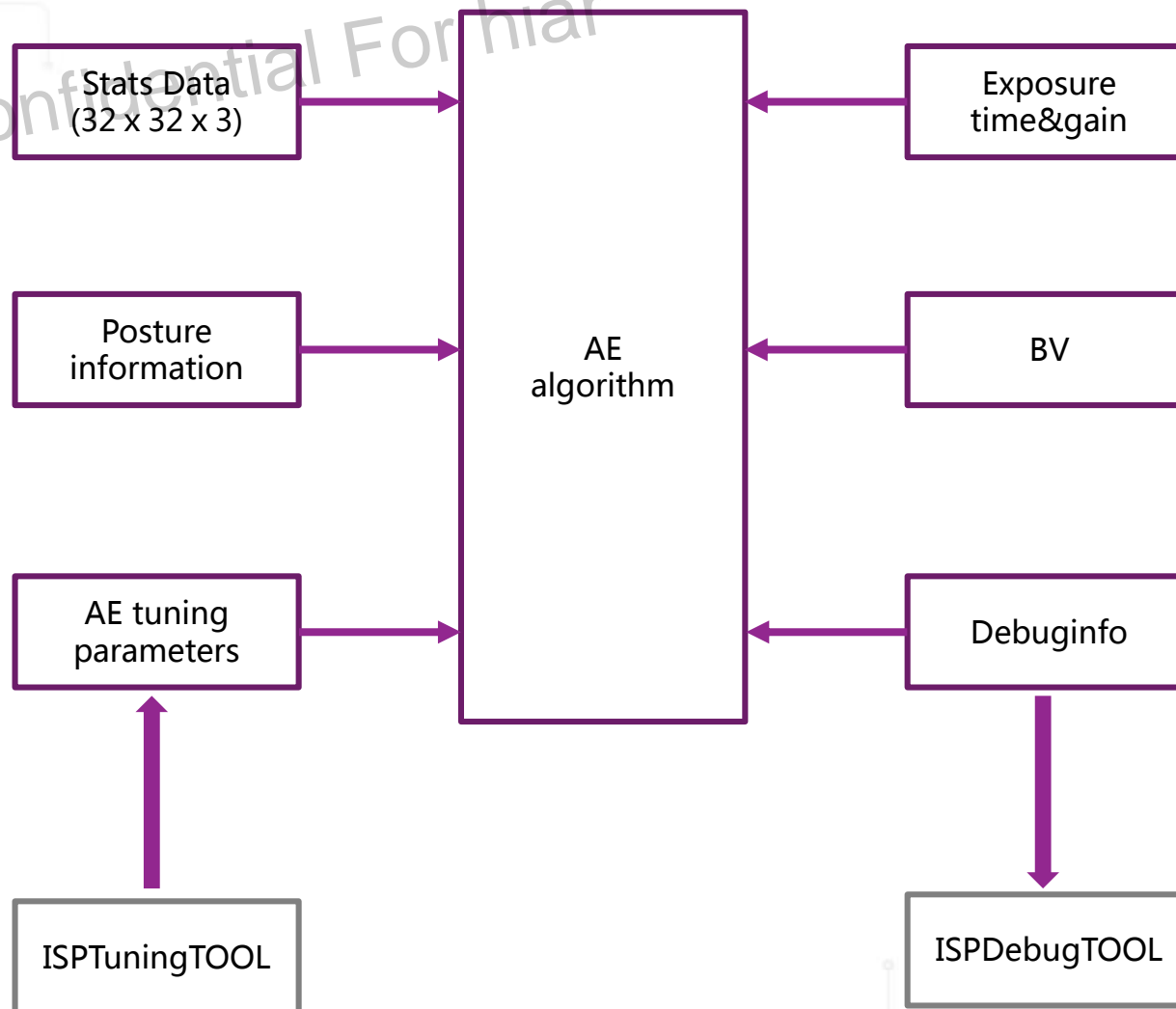
Camera Sensor驱动及参数配置

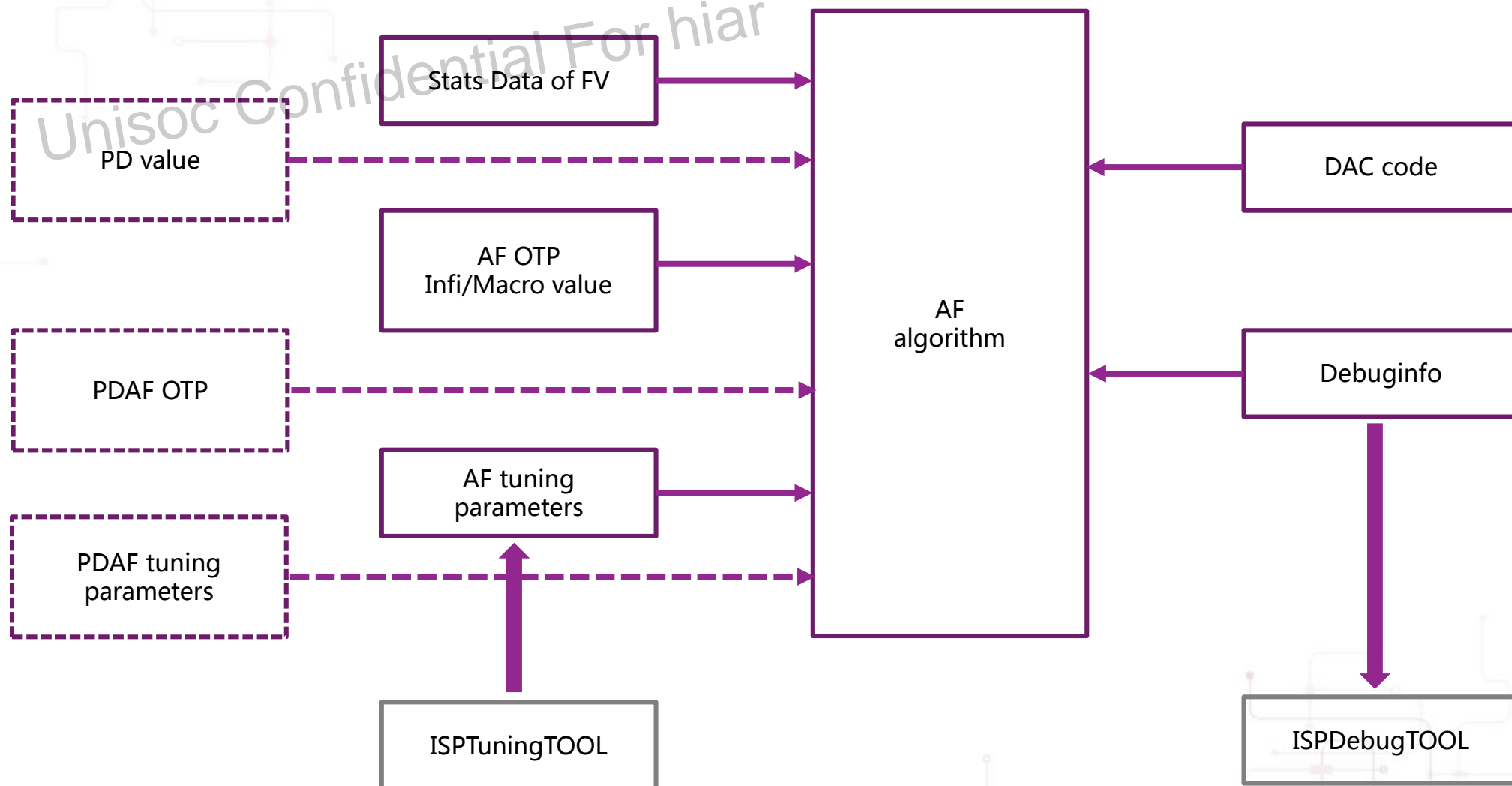
9

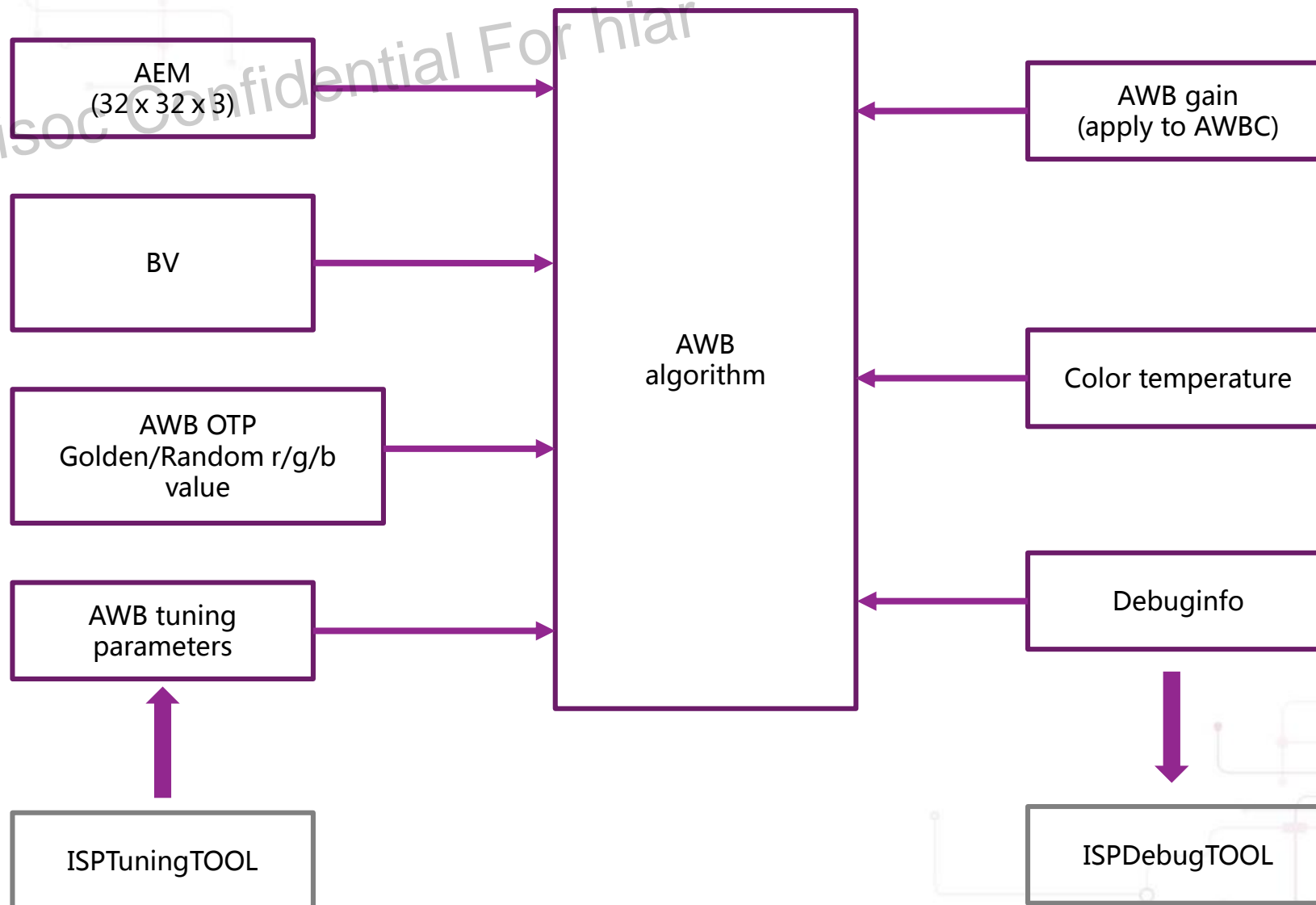
相机功耗和性能优化调试方法

10

常见问题分析方法







Contents

6

第三方图像后处理算法集成介绍

7

3A及ISP算法流程和逻辑

8

Camera Sensor驱动及参数配置

9

相机功耗和性能优化调试方法

10

常见问题分析方法

Sensor_drv相关文件

Unisoc Confidential For hiar

Sensor_drv_u.c

Sensor_cfg.c

Boardconfig.mk

xxx_mipi_raw.h
xxx_mipi_raw.c

Tuning bin

sprd_sensor_drv.c

dts

详细请参考相关平台Camera Driver Customization Guide

Contents

6

第三方图像后处理算法集成介绍

7

3A及ISP算法流程和逻辑

8

Camera Sensor驱动及参数配置

9

相机功耗和性能优化调试方法

10

常见问题分析方法

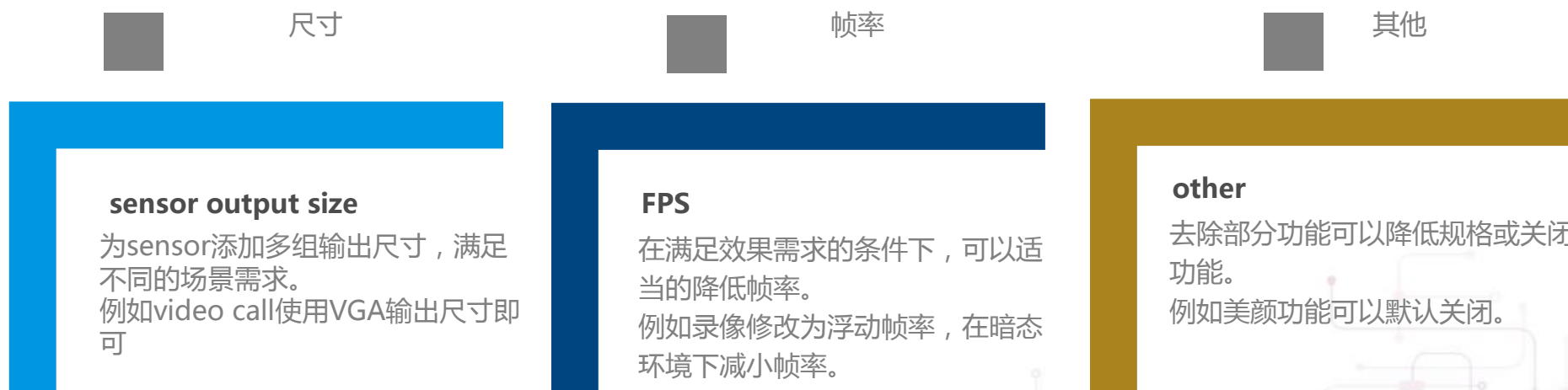
相机功耗的优化调试方法

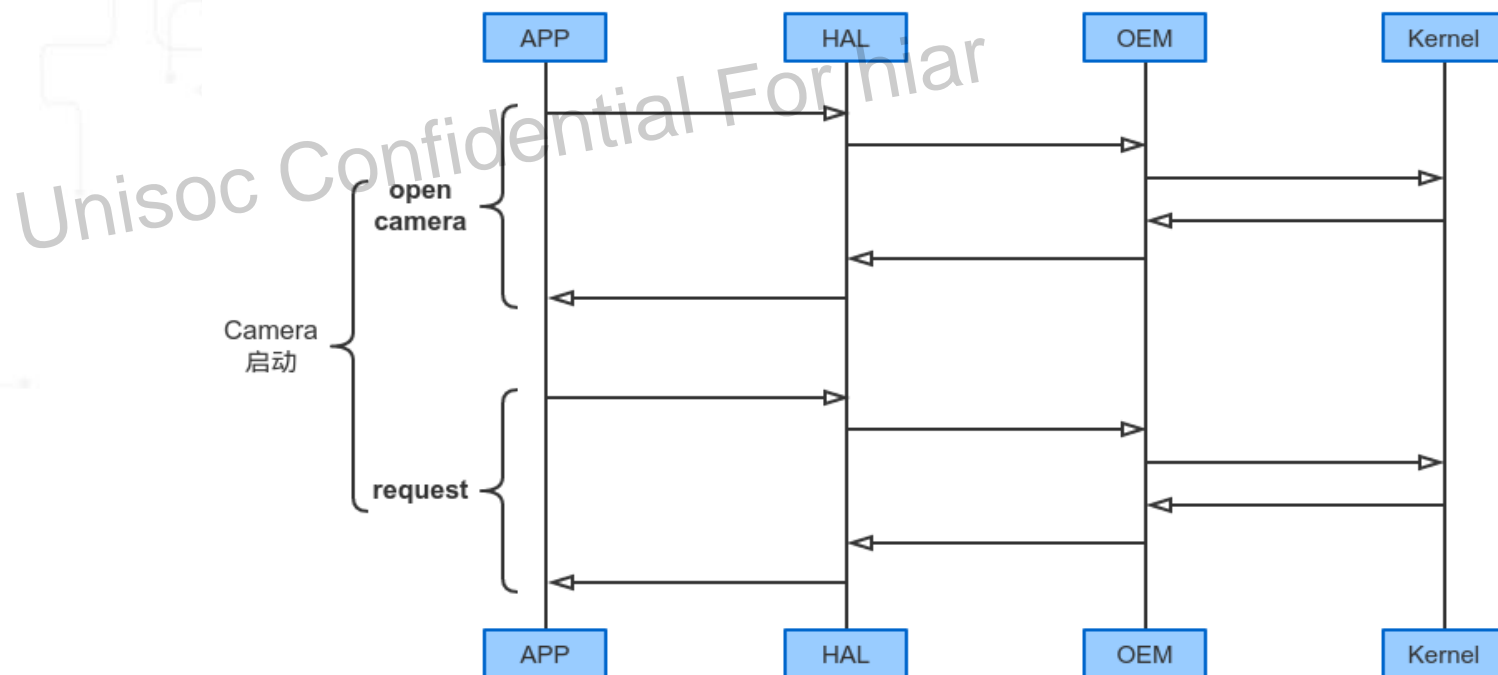
Δ DONE

一般release给客户的code是功耗最优方案，包含诸如CPU DVFS(Dynamic Voltage and Frequency Scaling)，DDR DFS(Dynamic Frequency Scaling)，系统Memory带宽优化，ISP clock auto gating 等。

Δ TODO

由于sensor型号不同，sensor输出尺寸和帧率可调功耗。





Camera启动流程：

- Δ 点触相机图标，上层初始准备完成后下发open camera命令，传到底层，底层进行创建线程、初始化参数、打开sensor硬件等操作，然后callback通知应用层；
- Δ 应用层收到通知，再往下发送request命令，请求帧数据用于预览，底层获取帧数据再上传给应用层，应用层进行处理然后显示。

Camera启动主要耗时点：

- Δ sensor启动的时序延时过长
- Δ IO读写操作（ioctl函数）
- Δ ISP初始化中的malloc/memset操作
- Δ dlopen加载动态库速度慢
- Δ ion内存分配

Contents

6

第三方图像后处理算法集成介绍

7

3A及ISP算法流程和逻辑

8

Camera Sensor驱动及参数配置

9

相机功耗和性能优化调试方法

10

常见问题分析方法

1、Log抓取方法(ylog, user & kernel)

插sdcard录制，ylog存放于\external_storage\ylog\ap\

User log: \external_storage\ylog\ap\录制时间\android\

Kernel log: \external_storage\ylog\ap\录制时间\kernel\

2、相机无法打开?

相机无法打开的原因多种多样，一般需要基于log具体分析。

3、ANR(相机无响应)?

首先分析log,从LOG可以看出ANR的类型，先看CPU的使用情况：

如果CPU使用量接近100%，说明当前设备很忙，有可能是CPU饥饿导致了ANR；

如果CPU使用量很少，说明主线程被BLOCK了；

如果IOwait很高，说明ANR有可能是主线程在进行I/O操作造成的。

从trace.txt文件查看各关联进程的调用stack, 结合代码仔细查看关键进程/线程的关系，是否有死锁的问题。找到造成死锁的原因。

4、Crash(相机报错)?

native crash问题需要从log中找到出错的backtrack，根据backtrace中的pc指针，使用address2line工具从符号表中解析出具体的代码出错位置，然后结合代码流程分析crash的原因。

1、死机/重启(panic)?

Get vmlinux of the kernel

Get ylog of the sysdump

- a) Prepare your sdcard
- b) After kernel crash, wait for ylog to be dumped into sdcard
- c) Reboot and pull ylog under data/ylog/, folder 1/ is the latest
- d) Combine sysdump files into one: `"cat sysdump.core.* > vmcore"`

Under vendor/sprd/tools/crash, run crash_arm for SharkLE: `"./crash_arm -m phys_base=0x80000000 vmcore vmlinux --cpus=2"`

Use gdb commands to analyze the log, e.g. :

- a)log : look at the kernel log before crash

2、预览/拍照无图像或图像异常？

dump camera HAL YUV图像数据，如果camera HAL层数据正常，需要从显示/编码端分析；如果camera HAL的YUV数据已经异常，需要从dcam/isp 驱动和sensor drv端分析。

THANKS



本文件所含数据和信息都属于紫光展锐所有的机密信息，紫光展锐保留所有相关权利。本文件仅为信息参考之目的提供，不包含任何明示或默示的知识产权许可，也不表示有任何明示或默示的保证，包括但不限于满足任何特殊目的、不侵权或性能。当您接受这份文件时，即表示您同意本文件中内容和信息属于紫光展锐机密信息，且同意在未获得紫光展锐书面同意前，不使用或复制本文件的整体或部分，也不向任何其他方披露本文件内容。紫光展锐有权在未经事先通知的情况下，在任何时候对本文件做任何修改。紫光展锐对本文件所含数据和信息不做任何保证，在任何情况下，紫光展锐均不负责任何与本文件相关的直接或间接的、任何伤害或损失。