

# *iOS*中*VideoToolBox* 硬编码

---

课程研发:CC老师  
课程授课:CC老师

转载分享需备注出处,不得用于商业用途

# 直播/小视频核心技术分享 产品研发/面试方向

课程研发:CC老师  
课程授课:CC老师

转载分享需备注出处,不得用于商业用途

# 视频+资料源码获取:

## 小雁子老师QQ: 1900009930



流媒体知识核心概念



源码



直播APP架构.png



直播APP流程思维导图



H.264标准详细图解.pdf



GPUImage 官方文档翻译\_...老师.pdf



音视频学习从零到整(1).pdf



音视频学习从零到整(2).pdf

课程研发:CC老师  
课程授课:CC老师

转载分享需备注出处,不得用于商业用途

**封面！一美拍/秒拍/抖音/直播 录制选择一个封面！**

课程研发:CC老师  
课程授课:CC老师

转载分享需备注出处,不得用于商业用途

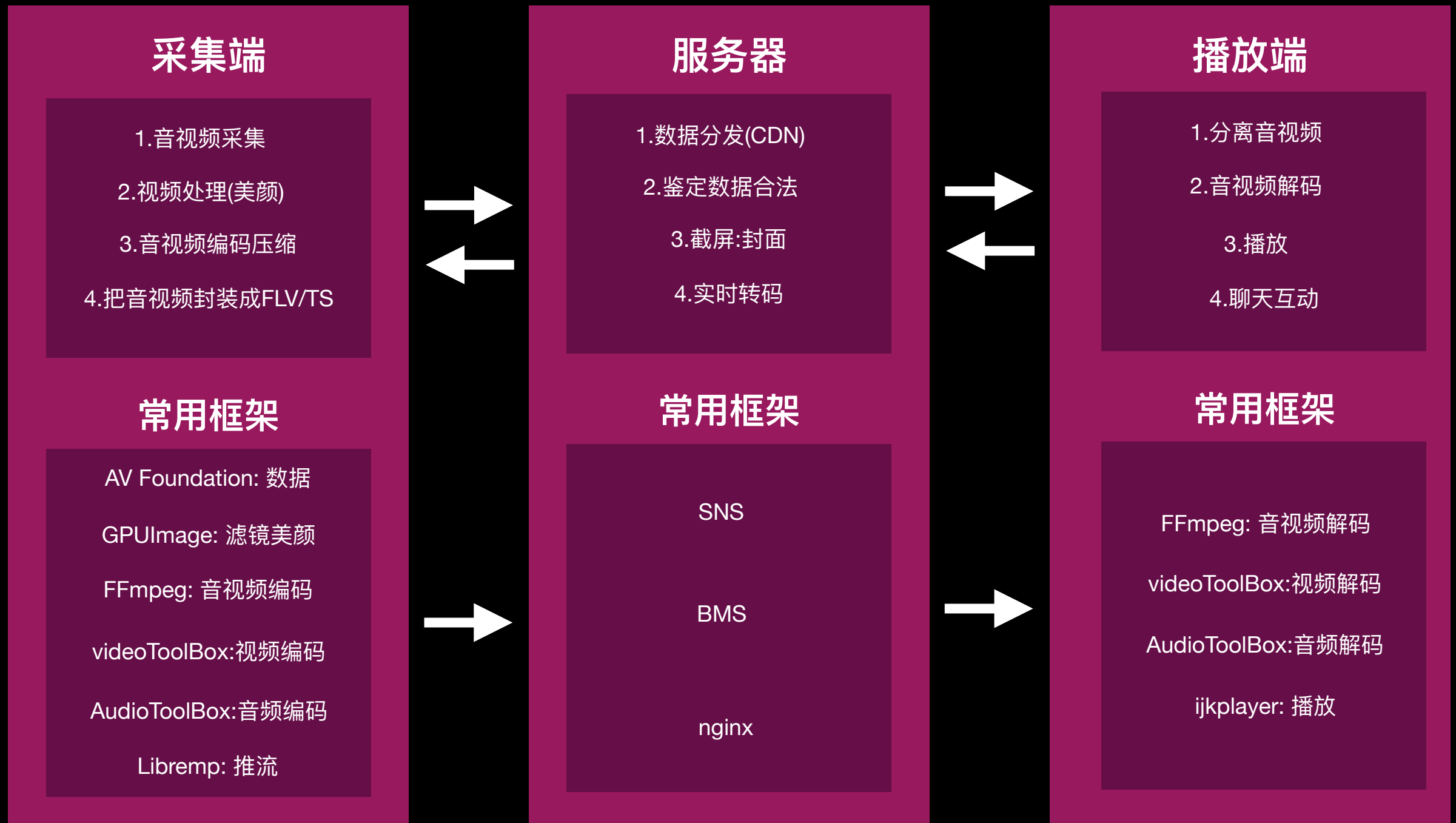
## 封面! —美拍/秒拍/抖音/直播 录制选择一个封面!

- 1.服务器分发给你一直播(支持回放!)
- 2.客户端:默认第一帧视频就是封面—小视频
- 3.客户端:从视频中选择合适的一帧作为封面—小视频

课程研发:CC老师  
课程授课:CC老师

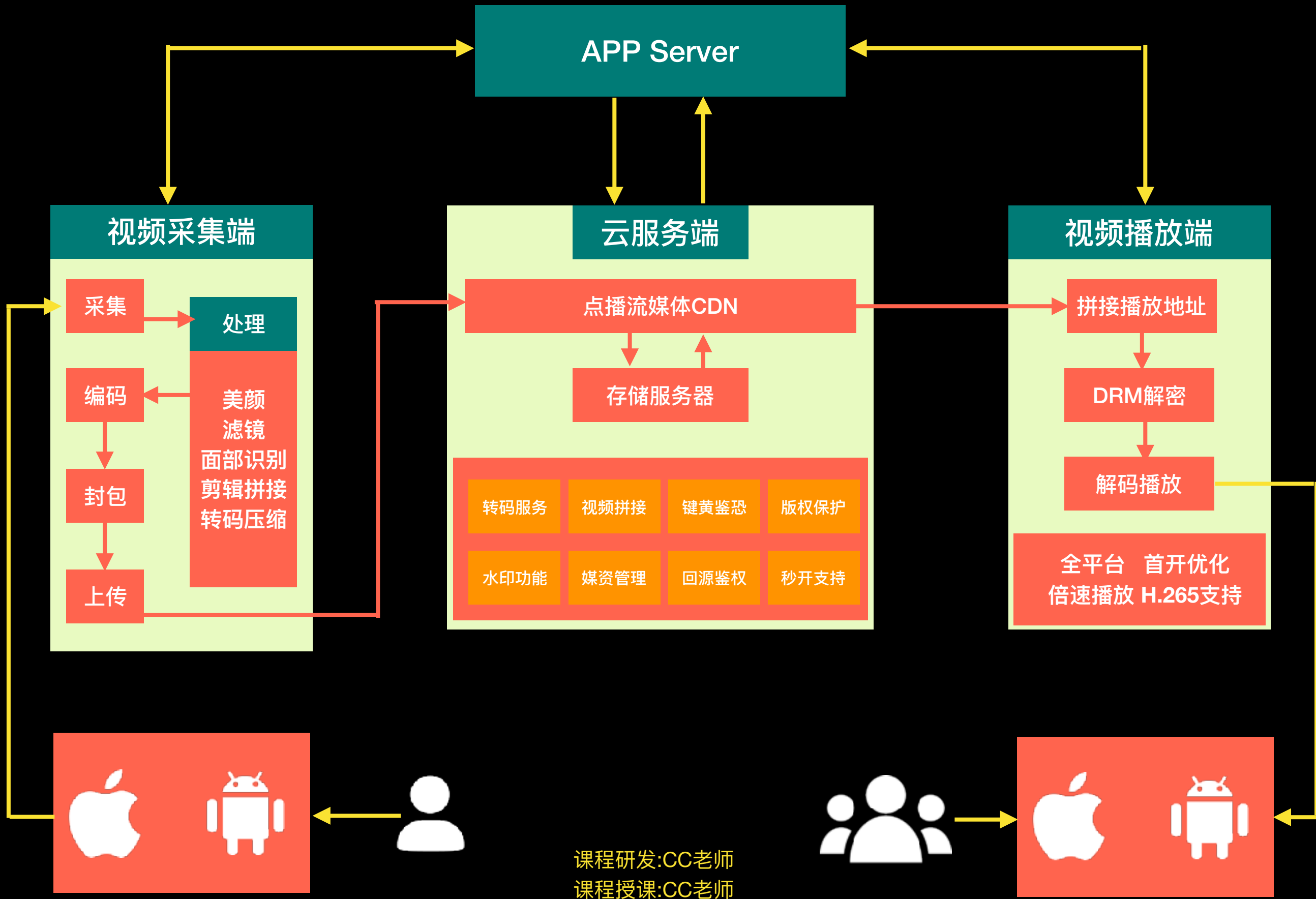
转载分享需备注出处,不得用于商业用途

# 直播APP架构



课程研发:CC老师

课程授课:CC老师



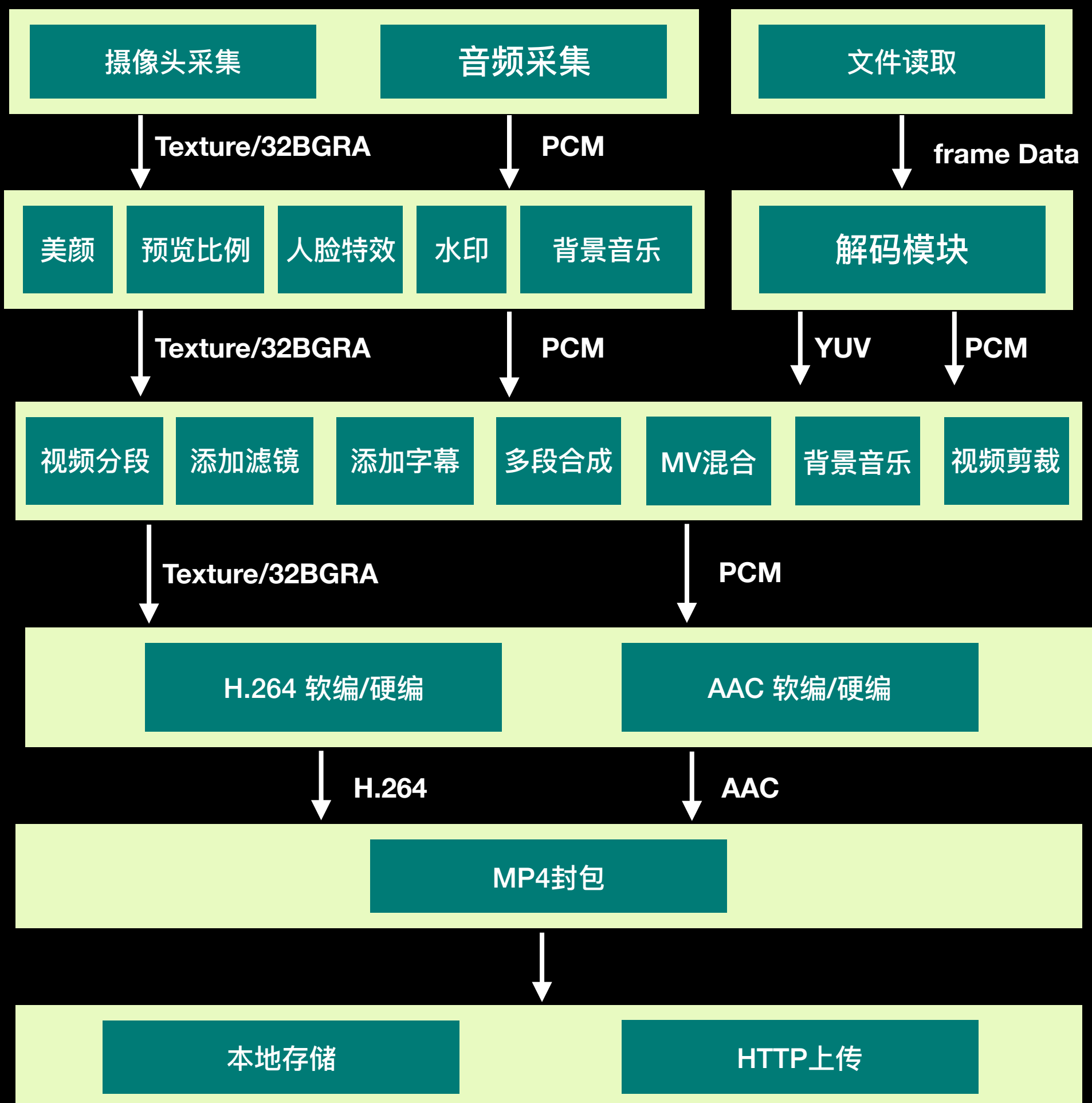
回源鉴权:回调地址作用当我们推流时服务商向我们的网站(服务器)发送请求,去验证此推流地址是否为我们自己的网站推出去的推流地址,如果是我们推出去的,要给服务商返回json数据code=200,证明是我们自己的网站或者我们自己的用户推出去的,这时服务商接收到json数据code=200,此刻鉴权成功,服务商允许我们推流和拉流;如果不是我们自己推出去的,给服务商返回json数据code=500,此刻鉴权失败,服务商不允许推流,更不可能拉流了。这样的防盗链相对来将更加安全。有效防止流量被他人倒用

第三个直播性能指标首屏耗时,指第一次点击播放后,肉眼看到画面所等待的时间。技术上指播放器解码第一帧渲染显示画面所花的耗时。通常说的“秒开”,指点击播放后,一秒内即可看到播放画面。首屏打开越快,说明用户体验越好

课程研发:CC老师  
课程授课:CC老师

转载分享需备注出处,不得用于商业用途





输入模块

视频采集

文件读取

编辑模块

编码模块

封包模块

输出模块

# 什么叫数据冗余？

你的妻子，Helen，将于明天晚上6点零5份在上海的虹桥机场接你

$$23 * 2 + 10 = 56 \text{个字符}$$

你的妻子将于明天晚上6点零5分在虹桥机场接你

$$20 * 2 + 2 = 42 \text{个字符}$$

Helen将于明天晚上6点在虹桥接你

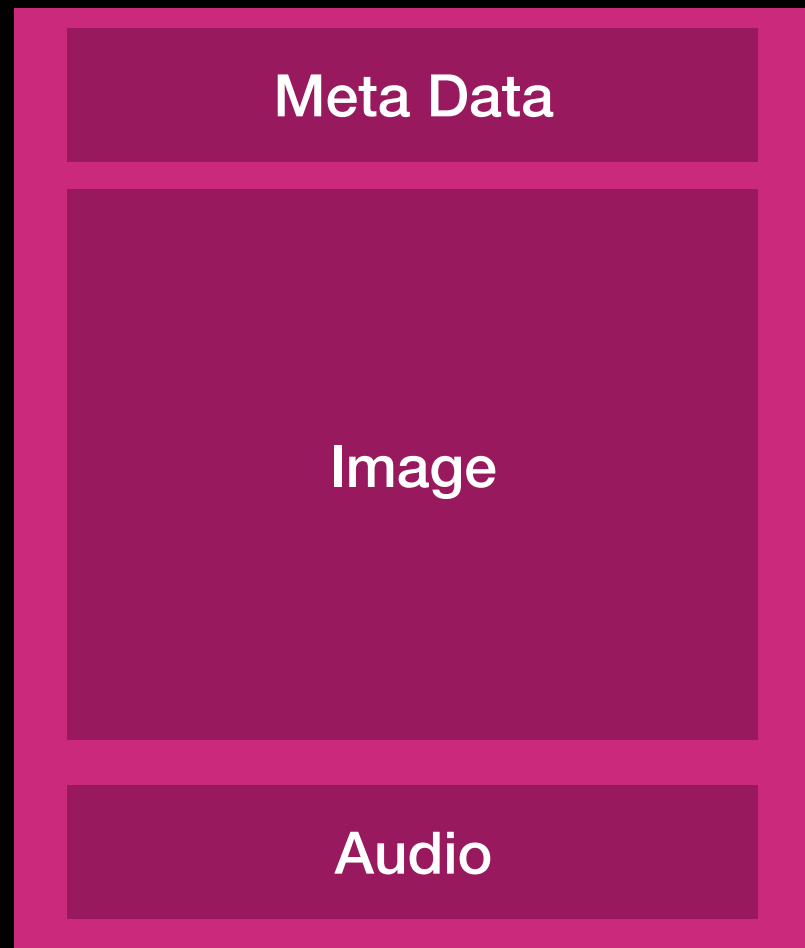
$$10 * 2 + 2 = 26 \text{个字符}$$

结论：只要接收端不会产生误解，就可以减少承载信息的数据量

课程研发:CC老师

课程授课:CC老师

# 视频到底是什么？



内容元素:

- 图像(Image)
- 音频(Audio)
- 元信息(Metadata)

编码格式:

- Video: H264
- Audio: AAC

容器封装:

- MP4/MOV/FLV/RM/RMVB/AVI

课程研发:CC老师

课程授课:CC老师

转载分享需备注出处,不得用于商业用途

# 视频帧

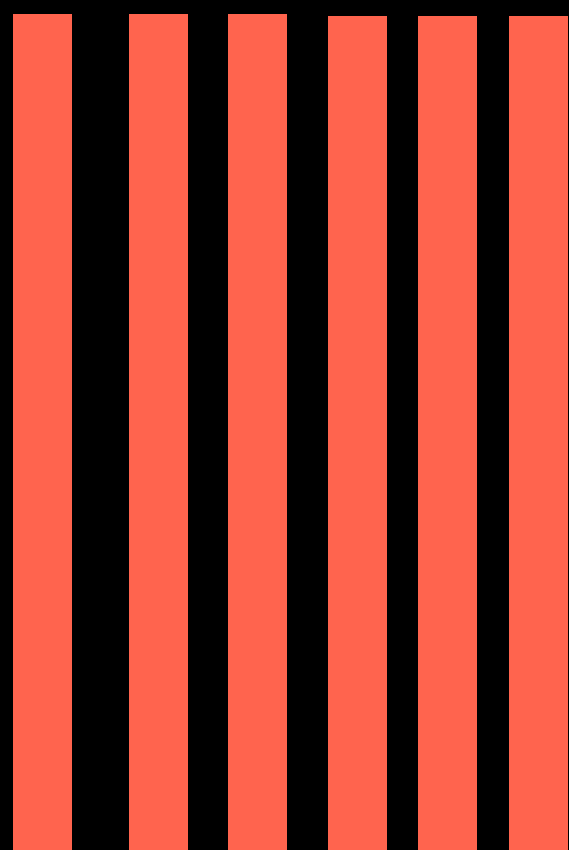


课程研发:CC老师

课程授课:CC老师

转载分享需备注出处,不得用于商业用途

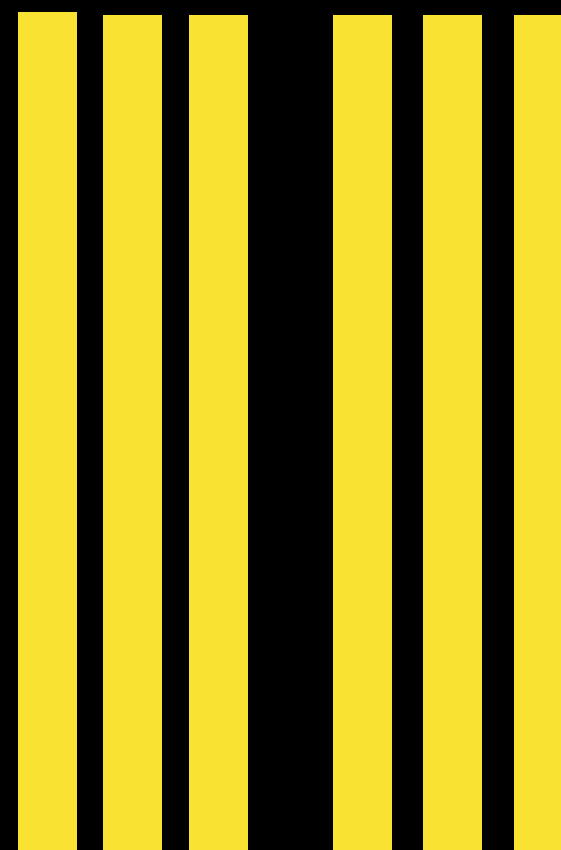
# 编码到底发生了什么？



帧率FPS:5



经过编码



帧率FPS:5  
一个GOP

课程研发:CC老师  
课程授课:CC老师

转载分享需备注出处,不得用于商业用途

# 编码发生了什么？



错误传递

完整图片,解码  
的参考

还原数据

参考帧 (B/P帧)

关键帧 (I 帧)

课程研发:CC老师  
课程授课:CC老师

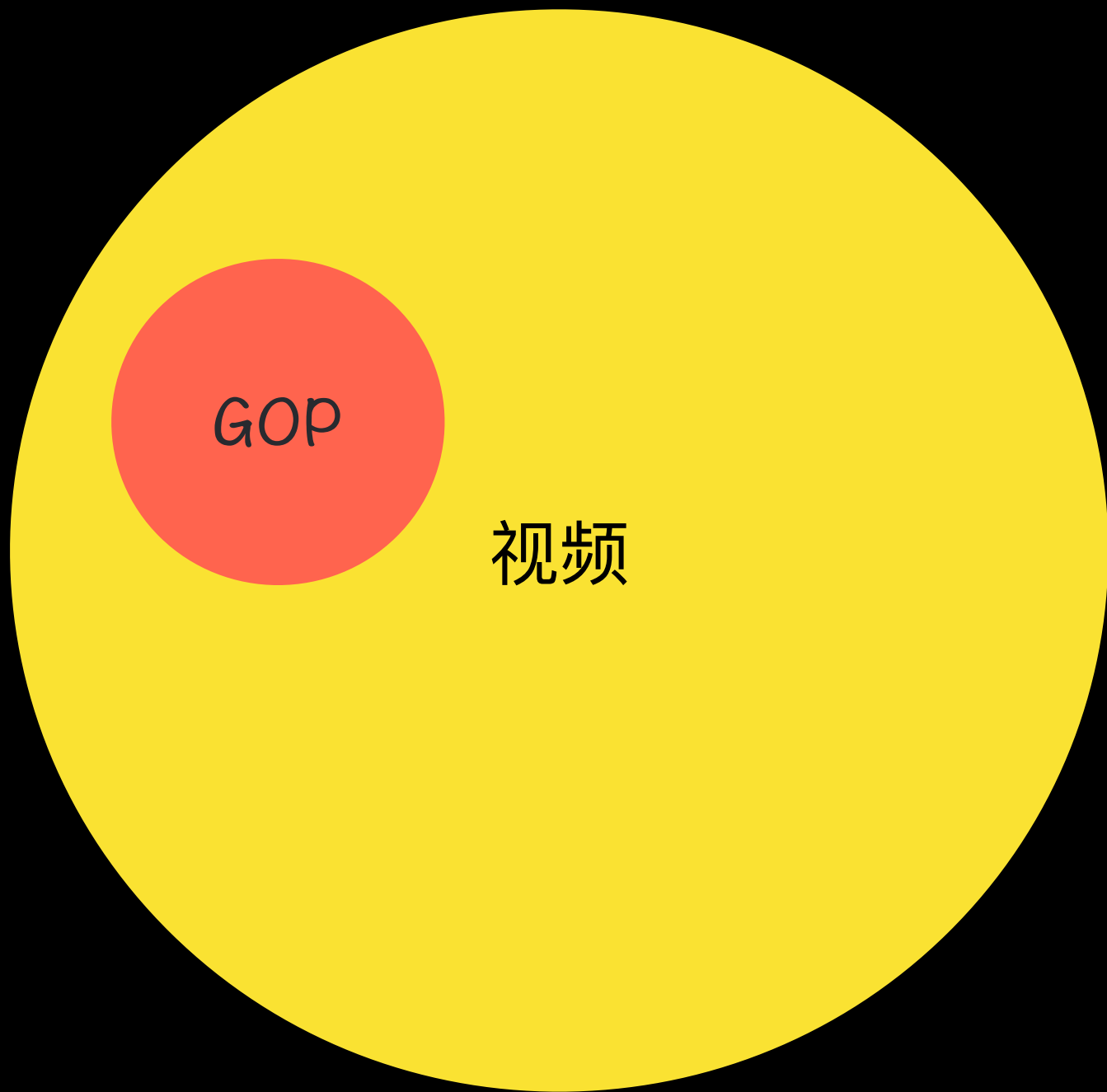
转载分享需备注出处,不得用于商业用途

# 小结

课程研发:CC老师

课程授课:CC老师

转载分享需备注出处,不得用于商业用途



- 1.直播流程
- 2.直播/音视频需要学习的框架?
- 3.视频的原理
- 4.编码/解码的本质

# videoToolbox工作流程

videoToolbox基于Core Foundation库函数, C语言

1.创建session -> 设置编码相关参数 -> 开始编码 ->循环输入源数据 (YUV类型的数据, 直接从摄像头获取) ->获取编码后的H264数据 ->结束编码

2.H264文件

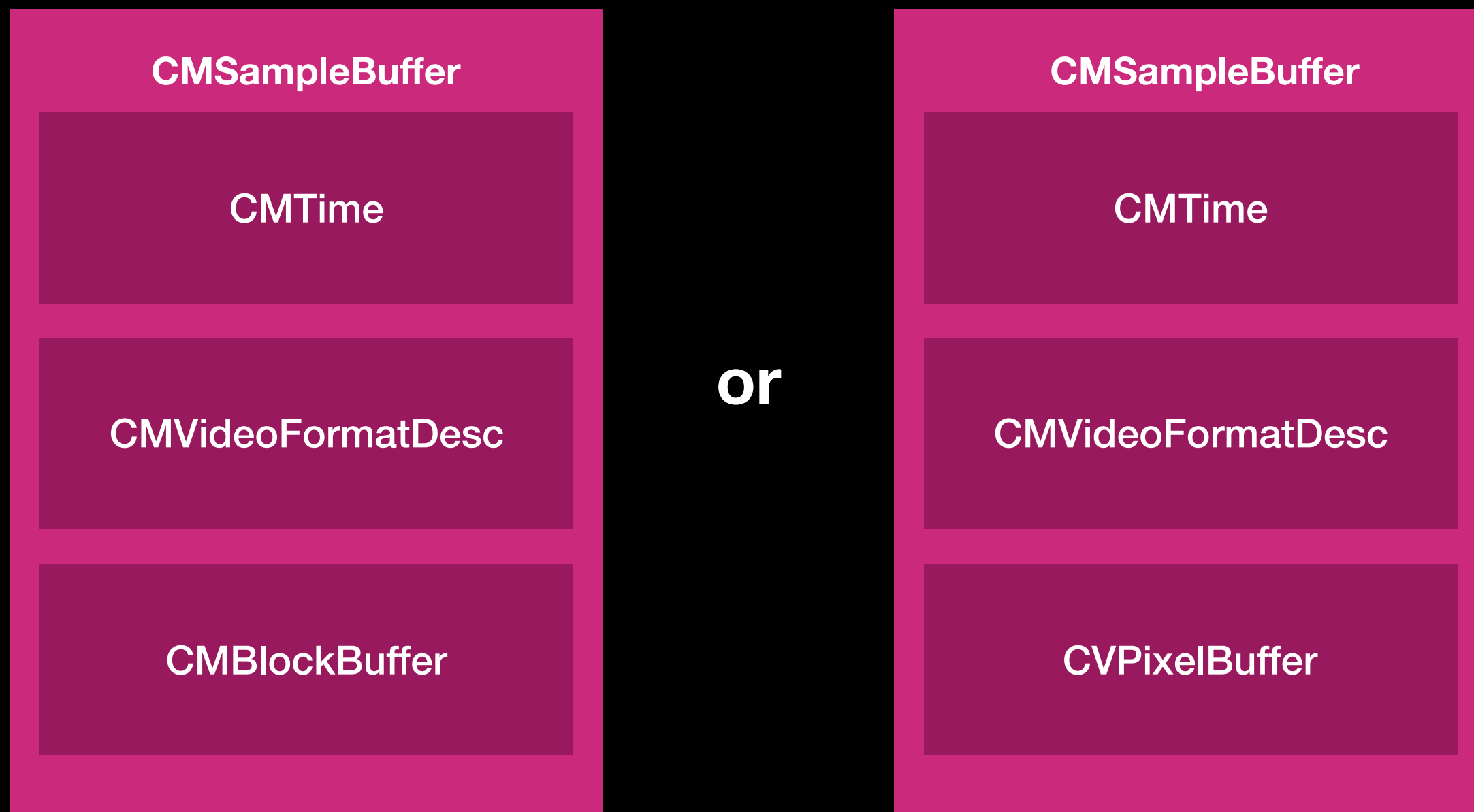
## 编码、解码

课程研发:CC老师  
课程授课:CC老师

转载分享需备注出处,不得用于商业用途



# CMSampleBuffer数据结构

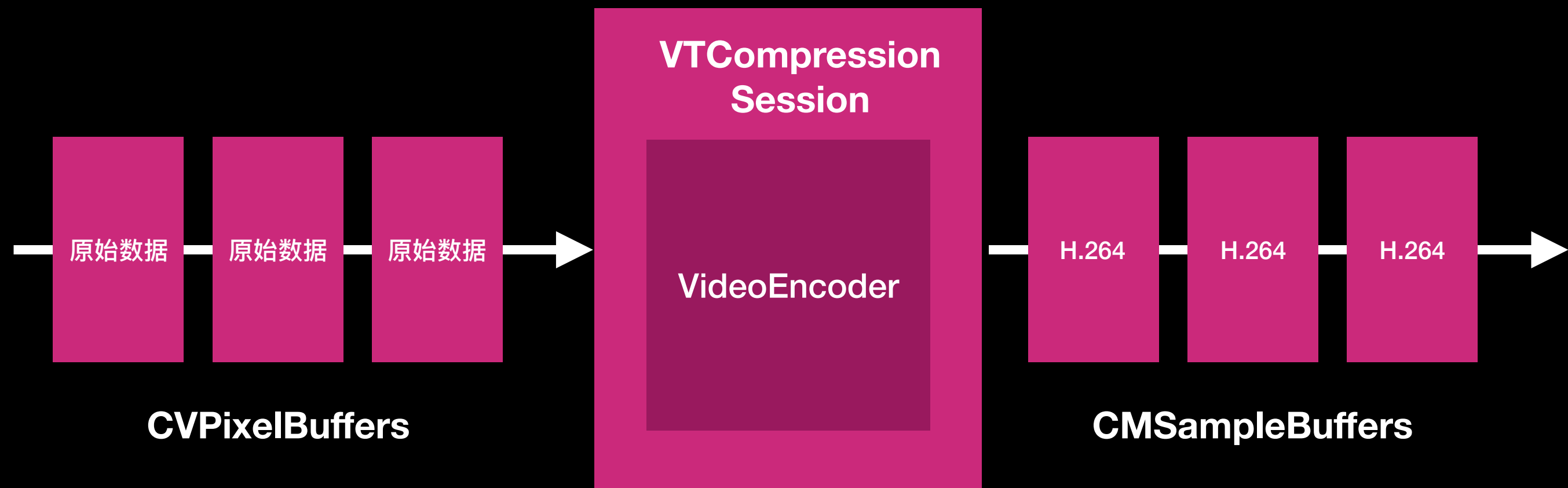


编码后的视频帧

课程研发:CC老师  
课程授课:CC老师

未编码的视频帧

# 编码的输入和输出

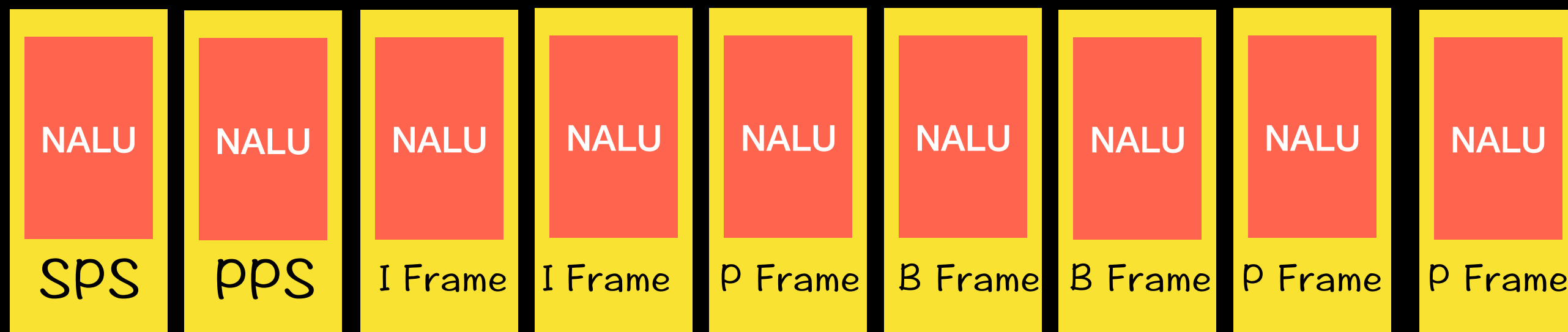


课程研发:CC老师  
课程授课:CC老师

转载分享需备注出处,不得用于商业用途

# H264文件？

$CMSampleBuffer = CMTime + FormatDesc + CMBlockBuffer$



FFmpeg、硬编码的方式，都是这样的格式

课程研发:CC老师  
课程授课:CC老师

转载分享需备注出处,不得用于商业用途

# 码率计算公式

项目	计算公式	192 * 144	320 * 240	480 * 360	640 * 480	1280 * 720	1920 * 1080
极低码率	(宽 * 高 * 3) / 4	30kb/s	60kb/s	120kb/s	250kb/s	500kb/s	1mbps
低码率	(宽 * 高 * 3) / 2	60kb/s	120kb/s	250kb/s	500kb/s	1mbps	2mbps
中码率	(宽 * 高 * 3)	120kb/s	250kb/s	500kb/s	1mbps	2mbps	4mbps
高码率	(宽 * 高 * 3) * 2	250kb/s	500kb/s	1mbps	2mbps	4mbps	8mbps
极高码率	(宽 * 高 * 3) * 4	500kb/s	1mb/s	2mbps	4mbps	8mbps	16mbps

# SPS和PPS是什么？

序列参数集SPS(Sequence Parameter Sets)

图像参数集PPS(Picture Parameter Sets )

课程研发:CC老师  
课程授课:CC老师

转载分享需备注出处,不得用于商业用途

## 序列参数集SPS

课程研发:CC老师

课程授课:CC老师

```
profile_idc = 66
constrained_set0_flag = 1
constrained_set1_flag = 1
constrained_set2_flag = 1
constrained_set3_flag = 0
level_idc = 20
seq_parameter_set_id = 0
chroma_format_idc = 1
bit_depth_luma_minus8 = 0
bit_depth_chroma_minus8 = 0
seq_scaling_matrix_present_flag = 0
log2_max_frame_num_minus4 = 0
pic_order_cnt_type = 2
log2_max_pic_order_cnt_lsb_minus4 = 0
delta_pic_order_always_zero_flag = 0
offset_for_non_ref_pic = 0
offset_for_top_to_bottom_field = 0
num_ref_frames_in_pic_order_cnt_cycle = 0
num_ref_frames = 1
gaps_in_frame_num_value_allowed_flag = 0
pic_width_in_mbs_minus1 = 21
pic_height_in_mbs_minus1 = 17
frame_mbs_only_flag = 1
mb_adaptive_frame_field_flag = 0
direct_8x8_interference_flag = 0
frame_cropping_flag = 0
frame_cropping_rect_left_offset = 0
frame_cropping_rect_right_offset = 0
frame_cropping_rect_top_offset = 0
frame_cropping_rect_bottom_offset = 0
vui_parameters_present_flag = 0
```

## 图像参数集pps

```
pic_parameter_set_id = 0
seq_parameter_set_id = 0
entropy_coding_mode_flag = 0
pic_order_present_flag = 0
num_slice_groups_minus1 = 0
slice_group_map_type = 0
num_ref_idx_l0_active_minus1 = 0
num_ref_idx_l1_active_minus1 = 0
weighted_pred_flag = 0
weighted_bipred_idc = 0
pic_init_qp_minus26 = 0
pic_init_qs_minus26 = 0
chroma_qp_index_offset = 10
deblocking_filter_control_present_flag = 1
constrained_intra_pred_flag = 0
redundant_pic_cnt_present_flag = 0
transform_8x8_mode_flag = 0
pic_scaling_matrix_present_flag = 0
second_chroma_qp_index_offset = 10
```

课程研发:CC老师

课程授课:CC老师

# SPS & PPS信息参数

编码所用的Profile、level、图像的宽和高、deblock录波器……

H264码流中第一个NALU是SPS & PPS

课程研发:CC老师  
课程授课:CC老师

转载分享需备注出处,不得用于商业用途



如下图：

Navigate to ☒ Hex Find

红色框为每个NAL的分割时，绿色框为PPS与SPS，它们也是以00000001分割的 接下为帧数据

0x00000000 :	00 00 00 01	67 42 00 1F	E9 05 89 C8	00 00 00 01
0x00000010 :	68 CE 06 F2	00 00 00 01	65 B8 40 56	38 8F FF 1F
0x00000020 :	10 00 02 0E	79 FD F7 C6	05 0E D3 4F	F8 8E 30 6C
0x00000030 :	2D 12 56 F7	C0 60 2A E0	26 60 06 D3	1B 07 34 8B
0x00000040 :	85 ED 37 04	8F 76 31 06	EB 5E 4A 02	F1 84 D3 7D
0x00000050 :	94 92 FB DE	1F FA 7D A0	D1 81 CB C2	48 94 1D 9C
0x00000060 :	0A 83 A5 02	07 A6 30 0F	F0 42 8A 87	C3 D5 CA 34
0x00000070 :	F5 38 D5 19	D2 A4 C9 F8	70 2A 59 3E	84 A5 04 68
0x00000080 :	74 FD 95 E8	BF FF 6E A8	E1 A1 CB A0	00 49 46 75
0x00000090 :	1F 54 E9 93	30 EE AA 6A	91 ED CB 4F	F2 6E 7F 24
0x000000A0 :	16 7F FE 05	6F BA 89 2C	4A C9 7F D4	46 0B AA 53
0x000000B0 :	60 04 D4 4B	4C 19 D6 73	FB BF EE FA	89 FF 48 56

课程研发:CC老师  
课程授课:CC老师

转载分享需备注出处,不得用于商业用途

## 2、如何判断帧类型（是图像参考帧还是I、P帧等）？

NALU类型是我们判断帧类型的利器，从官方文档中得出如下图：

nal_unit_type	NAL类型	C
0	未使用	
1	不分区、非 IDR 图像的片	2, 3, 4
2	片分区 A	2
3	片分区 B	3
4	片分区 C	4
5	IDR 图像中的片	2, 3
6	补充增强信息单元 (SEI)	5
7	序列参数集	0
8	图像参数集	1
9	分界符	6
10	序列结束	7
11	码流结束	8
12	填充	9
13..23	保留	
24..31	未使用	

课程研发:CC老师

课程授课:CC老师

转载分享需备注出处,不得用于商业用途

- (1) 第一位为禁位
- (2) 第2-3位为参考级别
- (3) 第4-8为nal单元类型



十六进制： 0x67

二进制： 0110 0111

十进制： (00111) 7

7对应序列参数集sps

课程研发:CC老师  
课程授课:CC老师

转载分享需备注出处,不得用于商业用途

十六进制： 0x68

二进制： 0110 1000

十进制： (01000) 8

8对应序列参数集pps

十六进制： 0x65

二进制： 0110 0101

十进制： (00101) 5

5对应序列 I 帧

课程研发:CC老师  
课程授课:CC老师

# H264解码思路

1.OpenGL ES

2.AVSampleBufferDisplayLayer

3.FFmpeg

4.videoToolbox硬解码

课程研发:CC老师  
课程授课:CC老师

转载分享需备注出处,不得用于商业用途

# FFmpeg命令

前提：Mac 上安装了homebrew  
并且下载安装好FFmpeg工具

```
ffmpeg -i H264文件 -i AAC文件 -vcodec copy -f MP4文件
```

```
ffmpeg -i /Users/liuyi/Desktop/CC_Video.h264 -i /Users/liuyi/Desktop/  
CC_Audio.aac -vcodec copy -f mp4 /Users/liuyi/Desktop/CC_Mp4.mp4
```

课程研发:CC老师  
课程授课:CC老师

转载分享需备注出处,不得用于商业用途

## 老师个人微信与技术博客地址



**CC老师\_MissCC** ♀

13	2960	75	70077	577
关注 >	粉丝 >	文章 >	字数	收获喜欢



课程研发:CC老师  
课程授课:CC老师

转载分享需备注出处,不得用于商业用途

想要学习更多 关注微信公众号 HelloCode开发者学习平台



课程研发:CC老师  
课程授课:CC老师

转载分享需备注出处,不得用于商业用途



# *Logic--Hello Coder*

---

学习,是一件开心的事

知识,是一个值得分享的东西

献给我可爱的程序员们!

课程研发:CC老师  
课程授课:CC老师

转载分享需备注出处,不得用于商业用途