# 共享音频方案调研汇总

## 修订记录

| 日期 | 修订版本 | 修改章节 | 修改描述 | 作者 |
| --- | --- | --- | --- | --- |
| 2022.7.20 | 1.00 |  | 初稿完成 |  |

## 调研结论

云视讯共享音频方案

**验证结论：** Maxhub大板3399调研确认，现有条件无法实现共享音频  
  
**原因：**  
 当前Androdid版本太低，需要升级到Androdid 10.0+或定制化系统才支持；

**已调研的方案：**

1. Android9以下版本

* Android无公用接口支持应用获取下行音频；
* 竞品(腾讯会议、小鱼、亿联、瞩目等)皆不支持共享音频，仅(华为大板welink)定制系统方案可行；

1. Android 10+或鸿蒙系统：

* 目前仅腾讯会议确认已支持；
* Android公用接口可支持；

**目前可尝试方案：**

***系统软件方向***  
1.系统升级Android 10以上，但升级系统适配工作量大、不确定性风险高(参见三．共享音频可行方案第1节)；

2.系统定制化增加接口，获取系统音频实现共享音频(参见三．共享音频可行方案第2节)；

***硬件方向***

3.MAXHUB修改硬件，需要额外增加采集电路，并验证可行性；  
4.设备外接声卡，需要考虑用户是否接受，及验证可行性；

**文档章节说明：**

二．竞品调研：针对各个竞品，确认是否支持共享音频，并对Android可能的实现方案开发验证可行性；

三．共享音频可行方案：针对不同Android版本及鸿蒙系统，实现共享音频方案有哪些可尝试方向；

## 竞品及方案调研

### 竞品调研(小鱼、welink、瞩目、腾讯会议、亿联)

#### 不支持共享竞品(小鱼、welink、瞩目、腾讯会议Android9以下)

##### 1.1.1 小鱼易联：不支持共享音频



仅共享白板，录屏，截图等

##### 1.1.2 Welink：不支持共享音频



移动端仅支持共享白板；

设备端华为大板Android9以下，为华为定制化系统可支持共享音频；

##### 1.1.3瞩目：不支持共享音频



不支持共享音频；

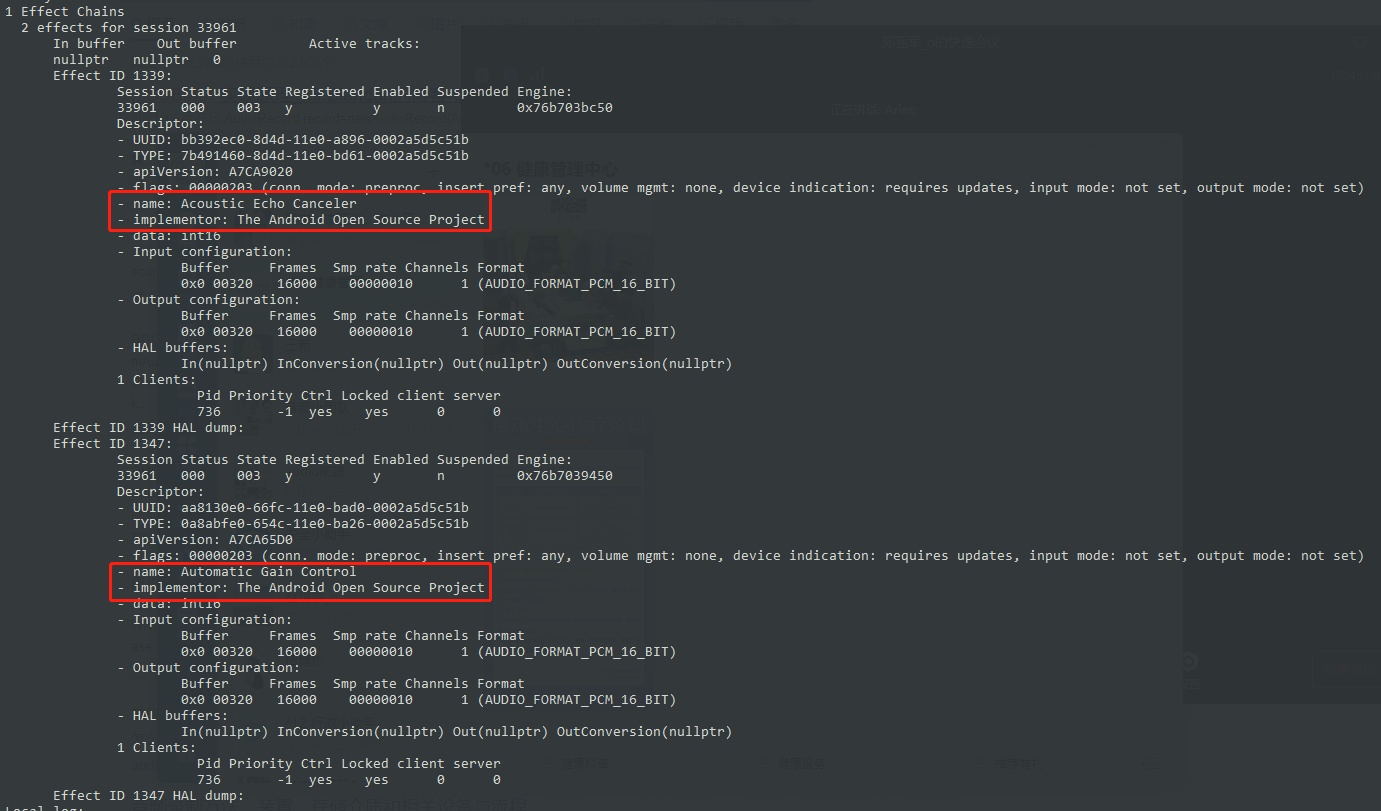
##### 1.1.4 腾讯会议：Android9及以下点击共享屏幕，不可共享音频



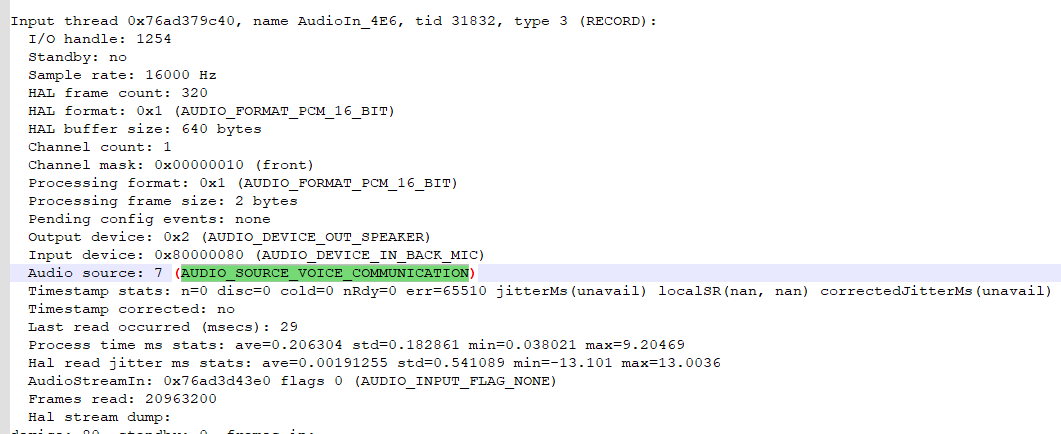
虽然UI显示共享屏幕，但实际上只共享了屏幕，并没有共享音频内容

#### 1.2支持共享(腾讯会议Android10+，鸿蒙2.0.0+）

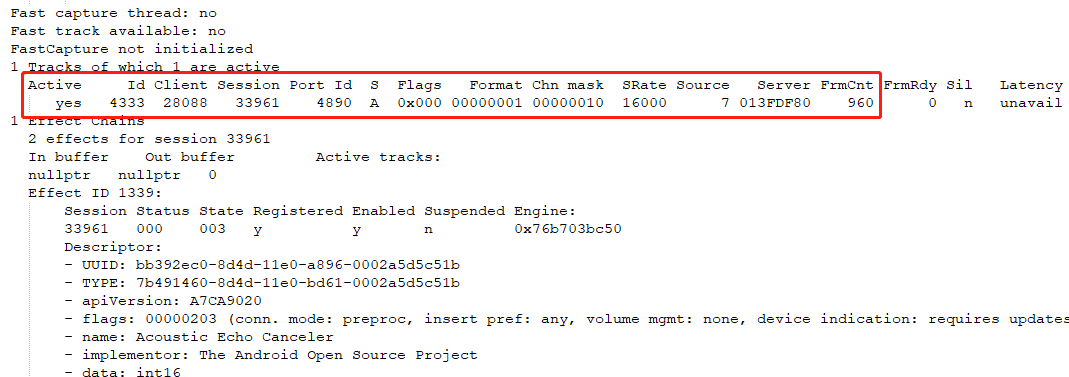
1. 腾讯会议开启会议后，可以看到如下系统信息，开启了AEC和AGC音效



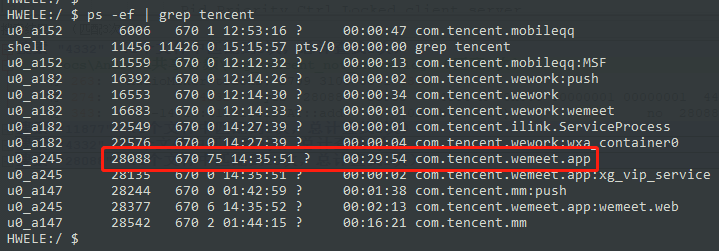
1. 有一路上行mic数据(16k、单声道、16bit)进行VOIP uplink

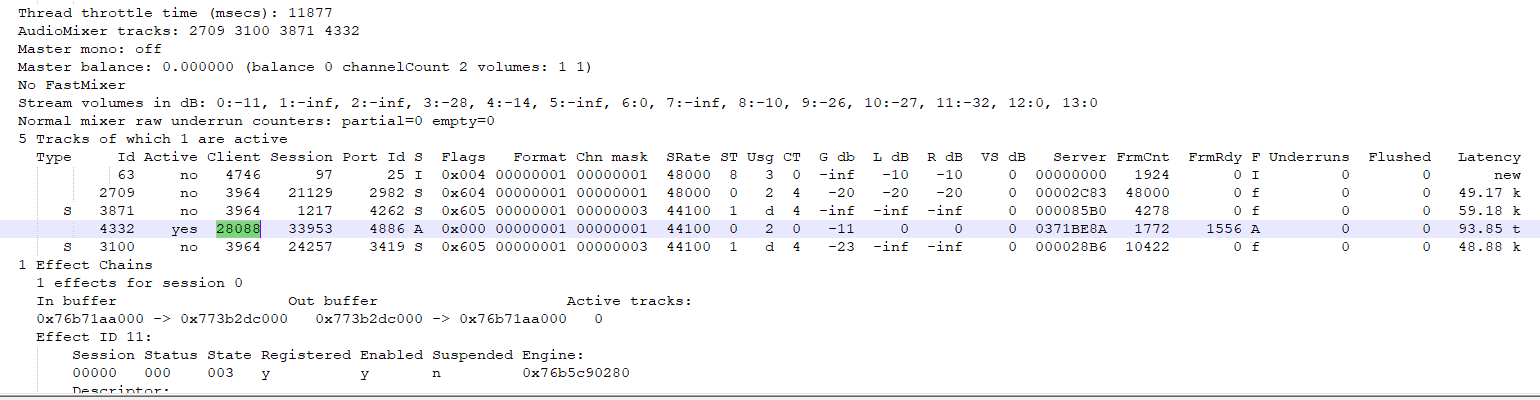


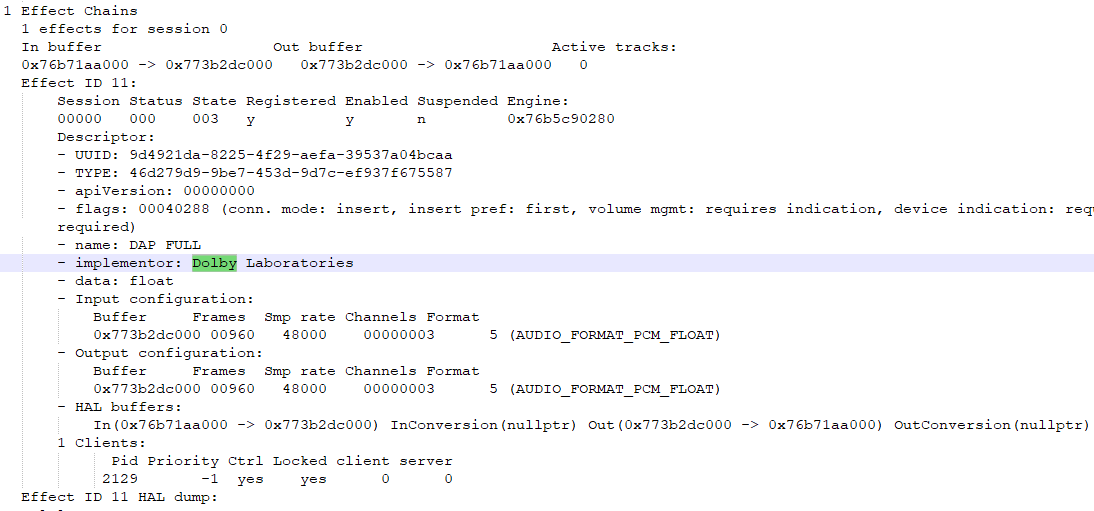
其中上行只有一路数据，同时上行含有AEC和AGC算法，是否全部使能未知



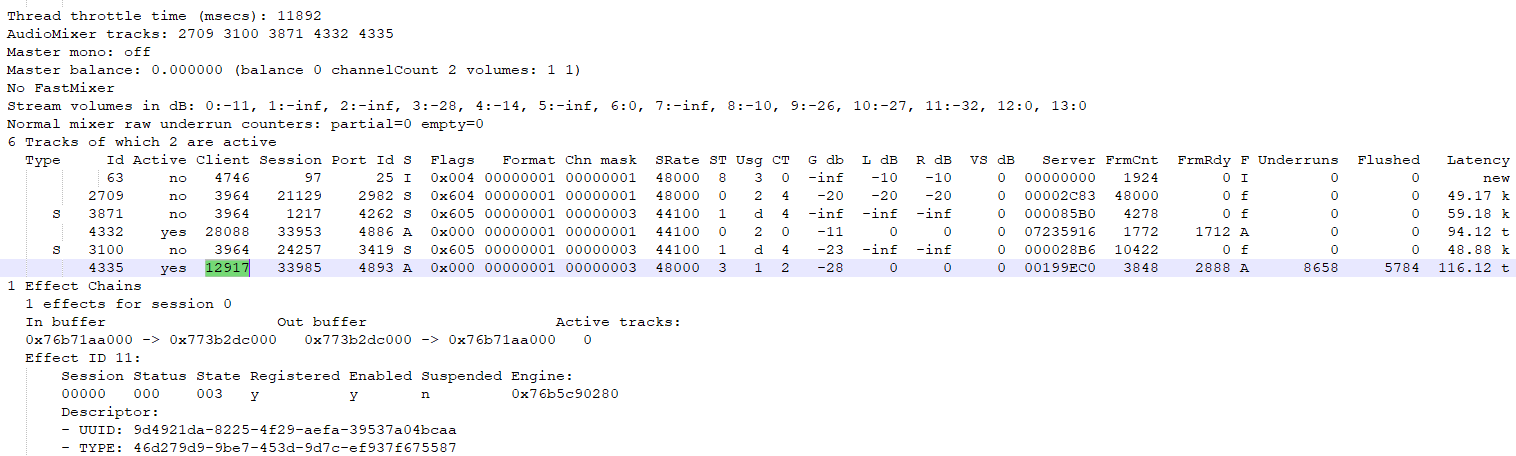
1. 一路下行数据对应VOIP downlink

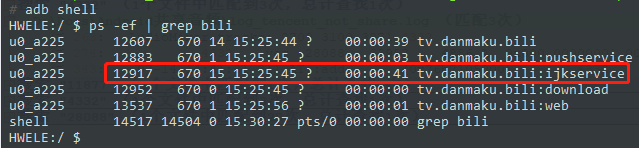




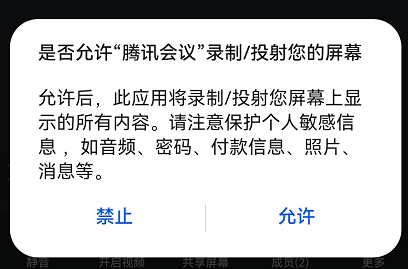
下行杜比音效

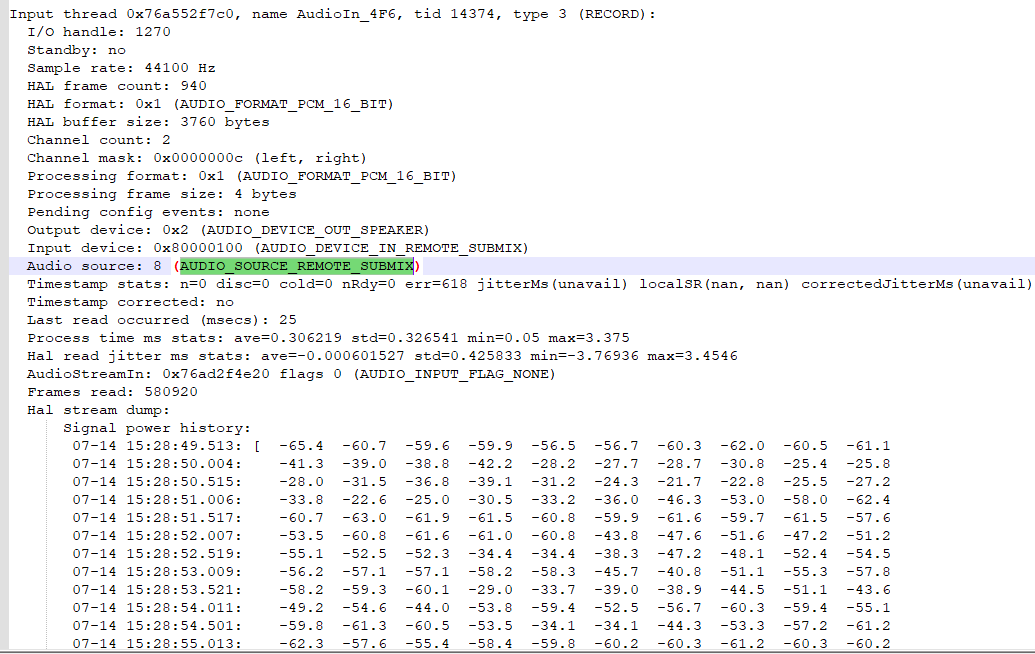
1. 开启哔哩哔哩播放

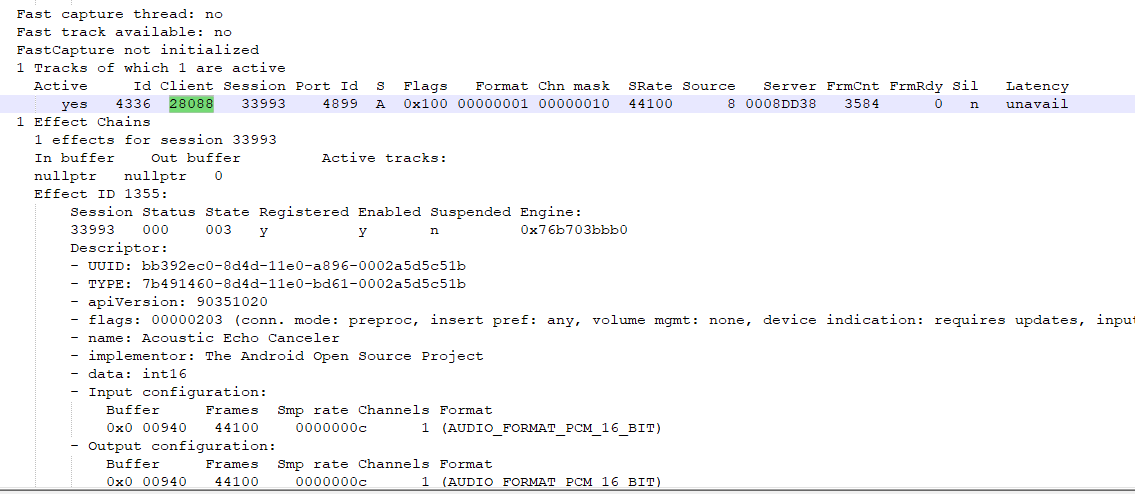




1. 开启共享音频后





腾讯对应进程多开启了remote\_submix线程，可能是通过录屏实现的功能

### 技术方案调研(Android 9以下版本)

#### 2.1 录音Remote submix方案

##### 2.1.1关于audiosource说明

|  |  |  |  |
| --- | --- | --- | --- |
| **int值** | **AudioSource** | **用途场景说明** | **其他** |
| -1 | AUDIO\_SOURCE\_INVALID | 不可使用，仅用于系统检测source等 |  |
| 0 | DEFAULT | 默认audiosource值，不知道设置哪个值时使用，系统底层一般情况下会将应用设置的source改为mic |  |
| 1 | MIC | 录音场景，一般想录取外界正常声音时使用，比如录音机应用等等； |  |
| 2 | VOICE\_UPLINK | 录取voice call对应的上行数据 |  |
| 3 | VOICE\_DOWNLINK | 录取voice call对应的下行数据 |  |
| 4 | VOICE\_CALL | 录取voice call上下行混音数据 |  |
| 5 | CAMCORDER | 录像场景使用的audiosource |  |
| 6 | VOICE\_RECOGNITION | 语音识别场景使用的audiosource值 |  |
| 7 | VOICE\_COMMUNICATION | 语音通话使用，一般用于VOIP通话场景 |  |
| **8** | **REMOTE\_SUBMIX** | **远程投屏等使用，使用后media声音无法通过系统播放而是传输到远程播放** |  |
| 9 | UNPROCESSED | 录取系统原始mic数据 |  |
| 1998 | RADIO\_TUNER | 录取radio音频数据，用于radio调试等场景 |  |
| 1999 | HOTWORD | 用于语音唤醒场景使用，比如你好小E、hi siri、OK google now | 仅用于系统语音唤醒 |

*/\*\*  
 \* Audio source for a submix of audio streams to be presented remotely.  
 \* <p>  
 \* An application can use this audio source to capture a mix of audio streams  
 \* that should be transmitted to a remote receiver such as a Wifi display.  
 \* While recording is active, these audio streams are redirected to the remote  
 \* submix instead of being played on the device speaker or headset.  
 \* </p><p>  
 \* Certain streams are excluded from the remote submix, including  
 \* {****@link*** *AudioManager#STREAM\_RING}, {****@link*** *AudioManager#STREAM\_ALARM},  
 \* and {****@link*** *AudioManager#STREAM\_NOTIFICATION}. These streams will continue  
 \* to be presented locally as usual.  
 \* </p><p>  
 \* Capturing the remote submix audio requires the  
 \* {****@link*** *android.Manifest.permission#CAPTURE\_AUDIO\_OUTPUT} permission.  
 \* This permission is reserved for use by system components and is not available to  
 \* third-party applications.  
 \* </p>  
 \*/*public static final int *REMOTE\_SUBMIX* = 8;

##### 2.1.2 Remote submix代码方案

检测录音权限，通过AudioRecord录取下行Remote submix数据，同时整个系统无媒体声

if (ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*RECORD\_AUDIO*) != PackageManager.*PERMISSION\_GRANTED*) {  
 // *TODO: do something while have no permission* return null;  
}  
  
int audioSource = MediaRecorder.AudioSource.*REMOTE\_SUBMIX*;  
int sampleRate = 16000;  
int format = AudioFormat.*CHANNEL\_IN\_MONO*;  
int encode = AudioFormat.*ENCODING\_PCM\_16BIT*;  
mBufferSizeInBytes = AudioRecord.*getMinBufferSize*(sampleRate, format, encode);  
AudioRecord audioRecorder = new AudioRecord(audioSource, sampleRate, format, encode, mBufferSizeInBytes);  
  
audioRecorder.startRecording();  
  
// read pcm buffer  
byte[] data = new byte[mBufferSizeInBytes];  
int readSize = audioRecord.read(data, 0, mBufferSizeInBytes);

// stop record  
audioRecorder.stop();  
audioRecorder.release();

条件梳理如下：

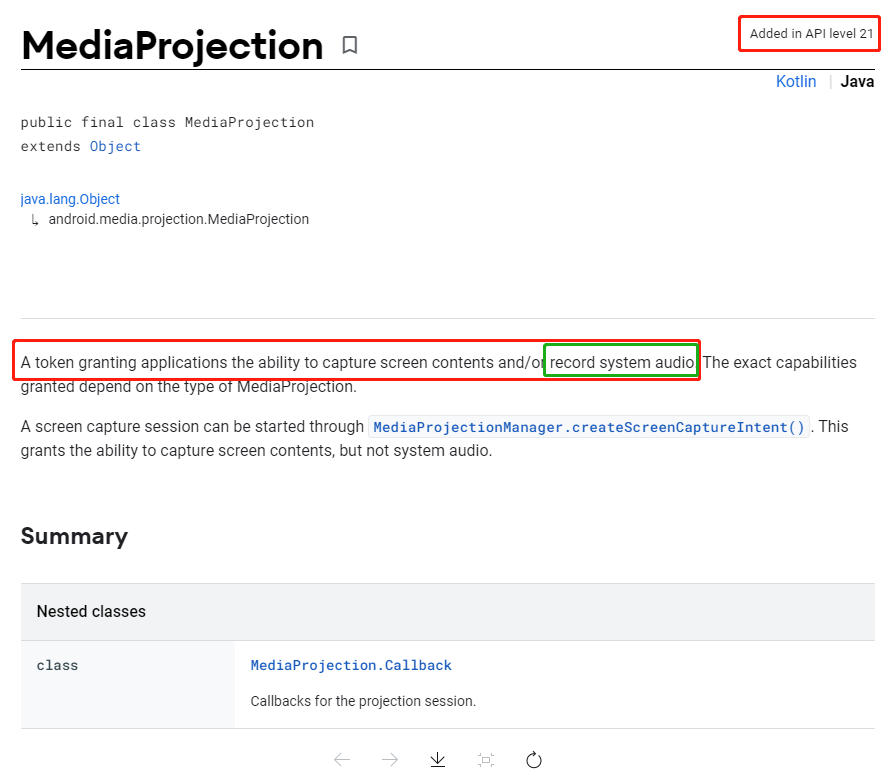
* 导致下行media声音不会播放到喇叭，而是被截取（不符合要求）；
* 系统自带的提示音等会正常播放；
* 需要有CAPTURE\_AUDIO\_OUTPUT权限；
* 只能系统组件等使用，对第三方应用无效（系统应用，设备端满足要求，移动端不可行）；
* 录取到的是纯pcm数据流；
* Android只允许一个录音进程，即使Android10+（只有一个audiorecord可以成功运行，具体参见https://developer.android.com/guide/topics/media/sharing-audio-input）；

##### 2.1.3 结论

直接通过Android的audiorecord，使用remote\_submix不满足要求，无法使用。

#### 2.2 录屏方案

Android 21(android 5.0)开始提供了录屏接口，第三方应用可通过申请临时令牌获得临时权限，进行录屏操作，可录取屏幕内容或系统声音



##### 2.2.1 录屏demo实现

录屏的实现方案参见Android studio工程



官方文档虽然说支持共享音频，但实际开发反复确认验证发现，Android9版本一下MediaProject根本没有接口支持录取下行音频数据。

具体录屏的方案可参考：<https://www.jianshu.com/p/3487d8291969>

用户可通过获取到用户授权的MediaProject进行录屏，然后通过MediaRecorder直接将屏幕的surface和音频编码成格式文件保存。

或者使用MediaCodec完成流推送等。

##### 2.2.2 结论

在Android 9及以下版本，确认MediaProject未提供录取系统音频的权限，与google官方文档不符，不满足共享音频要求。

## 共享音频调研可尝试方案(Android平台)

### Android10+版本playback-capture方案

参见官方文档

https://developer.android.com/guide/topics/media/playback-capture?hl=zh-cn#%E6%8D%95%E8%8E%B7%E9%9F%B3%E9%A2%91

### 2 系统定制方案(Android9以下版本)

根据remote\_submix的原理，对Android系统进行定制（已知华为在2016年已经实现录取下行音频的功能）



1. 修改Android framework audiopolicymanager，允许系统多个模块录取音频数据；

关于如何在Android9及以下版本，开启多应用录音，请参考

https://blog.csdn.net/qq\_33443989/article/details/84232061

1. 新增audiosource：record\_screen,record\_screen和remote\_submix一致，唯一的差异是不截断下行数据播放；

其中下图红色部分为修改方案，箭头线为buffer数据流

方案1：直接从framework层获取下行混音数据；

方案2：直接从Hal层获取下行混音数据；

方案3：直接从DAC前获取ecref数据；

拿到数据后给到Mindlinker即可

### **鸿蒙系统**

鸿蒙系统2.0.0.47版本支持录屏方案，腾讯会议已经专门支持鸿蒙系统，实现录屏共享音频方案

可以将鸿蒙系统2.0.0以上版本，对标为Android10+版本

目前鸿蒙如何实现录屏，还需要进一步确认