

[SCA] App取证视频拍摄—技术方案

•

Ownership

Product Manager	The PIC of Product manager
Project Manager	The PIC of Project Manager
Native Dev	The PIC of Android or iOS Dev
RN Dev	The PIC of React native dev
Server Dev	The PIC of non-native dev
Designer	The PIC of designer
QA	The PIC of QA

Resources

PRD	QC 支持视频上传
Figma	UI稿
Transify	The link of transify in https://transify.seagroup.com
Git Repo(Optional)	The git repo or branch for this project if necessary to indicate
Dependent Service Doc(Optional)	SDK integration doc, official guideline, etc.
Project Schedule Page (Optional)	Usually created by project manager, timeline of the document
Feasibility Study Doc (Optional)	Usually feasibility study is ahead of PRD.

Background

退货管理系统 QC 阶段通过视频的形式取证，较图片更高效，核心需求如下：

1. 支持通过相机拍摄或者本地相册选择视频上传。
2. App 和 Web 上传的视频支持互相查看。

App 视频拍摄属于通用的基础功能，需要沉淀到基础组件库中，方便后续其它 APP 接入，主要功能点如下：

1. 提供拍照和视频录制界面，视频录制最大时长支持配置。
2. 支持相机拍摄和本地相册视频的预览。
3. 支持对上传视频进行压缩。
4. 支持查看已上传的在线视频。
5. 对接图片选择控件。

Overall Design (Optional)

Improvement(revamp/refactor) for existing feature

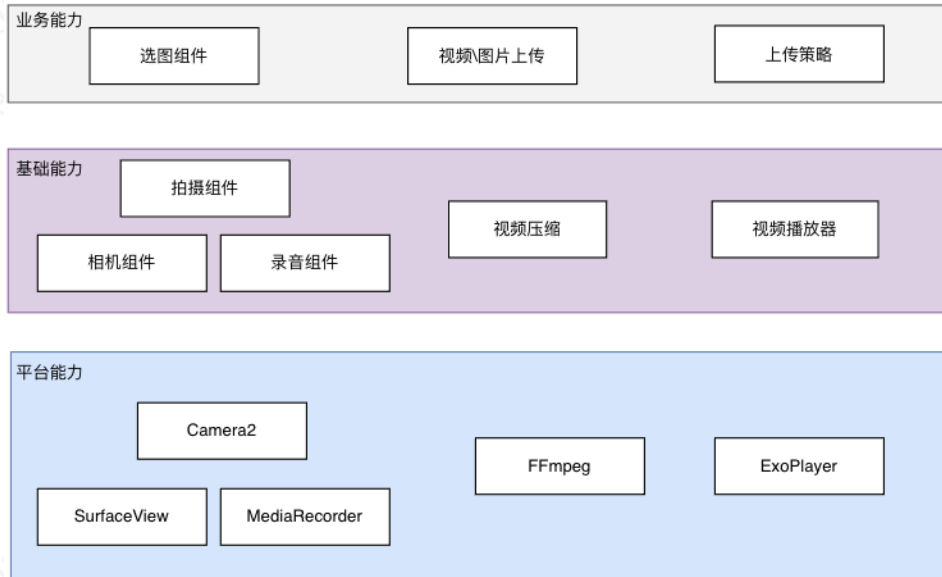
- Current

只支持拍摄图片进行取证。

- Improvement

支持拍摄图片和录制视频进行取证。

Brand new



各个层级介绍如下：

1. 平台能力：提供视频拍摄、压缩编码和预览播放核心 API，主要是利用 Android 原生和 FFmpeg 的能力。
2. 基础能力：相机拍摄、视频压缩库和视频播放器。
3. 业务能力：选图组件对接基础能力层支持视频拍摄功能，视频上传也是由业务实现。

Code organization

Where and how you wanna put your code, better a code dir graph

Interface Design

Interfaces of the most important part in your design

Detailed Design

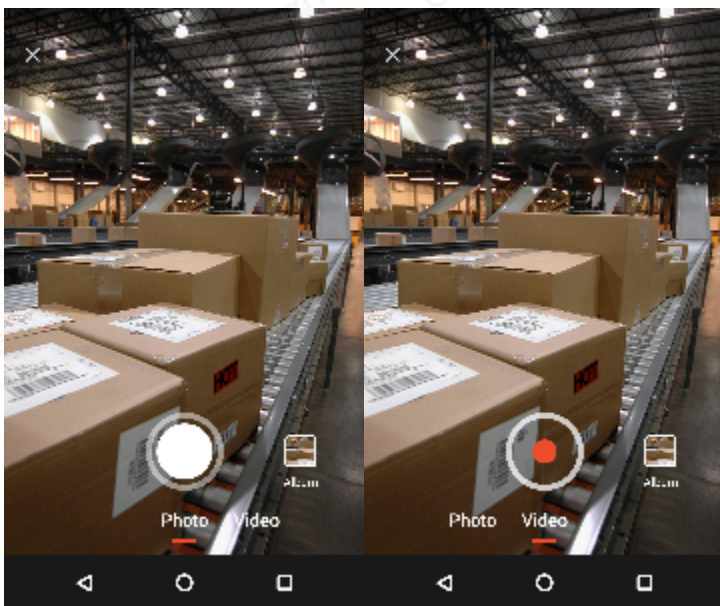
1、拍摄

开发 Android 相机拍摄界面，一般来说，有三个选择：

1. Camera1
2. Camera2
3. CameraX

Android 相机框架按照 Camera1 → Camera2 → CameraX 顺序迭代，Google 在 Android 5.0 之后已经废弃 Camera1 框架，CameraX 视频拍摄模块目前还没发布正式版本，所以选择使用 **Camera2** 进行自定义相机拍摄界面开发。

Camera2 是 Android 5.0 引入新的相机框架，可以搭配 MediaRecorder 进行视频录制。Android 相机各个框架使用差异很大，拍摄界面和相机框架实现逻辑需要分开，方便后期替换相机 API，以及拍摄界面自定义修改。



2、视频压缩

对于视频压缩的需求，一般来说，会有这么两个选择

1. MediaCodec
2. FFmpeg + x264

MediaCodec 是 API 16 之后 Google 推出的用于音视频编解码的一套偏底层的API，可以直接访问底层的媒体编解码器对视频进行编/解码，压缩速度较快，无需引入第三方库，但接入较复杂，可能存在设备兼容性**问题**。

FFmpeg 通过软件让 CPU 来对视频进行解码，具有更好的适应性，软解不考虑设备的硬件解码支持情况，有CPU就可以使用。FFmpeg 需要引入 so 库，只支持 arm64-v8a 一种 CPU 架构库大小是 7MB 作用，如果只需压缩指定格式视频，包大小可以进一步减小，预计在 3 MB 左右。

它们的使用方法：

1. MediaCodec 视频压缩方法可以查看官方文档：[MediaCodec](#)
2. FFmpeg 和 libx264先编译so库，通过下面命令行进行压缩：

```
"ffmpeg -y -i " + inPath + " -vf scale=540:960 -acodec copy -vcodec libx264 -r 25 -b:v 800k " + outPath
```

通过上面命令压缩视频 FFmpeg 的压缩速度比 MediaCodec 慢了三倍。查阅资料发现 FFmpeg 可以通过禁止自动旋转、指定快速双线性缩放算法和 superfast 预设等来提升压缩速度：

- -noautorotate 禁止自动旋转
- -sws_flags fast_bilinear 快速双线性缩放算法
- -preset superfast 预设是一系列参数的集合，这个集合能够在编码速度和压缩率之间做出一个权衡

```
"ffmpeg -y -noautorotate -i " + inPath + " -vf scale=540:960 -sws_flags fast_bilinear -acodec copy -vcodec libx264 -r 25 -b:v 800k -preset superfast " + outPath
```

下面对比 FFmpeg 和 MediaCodec 的压缩速度，视频分辨率从 1080 x 1920 压缩至 540 x 960，比特率=800k，FPS=25。

PDA ZEBRA TC20:

压缩方式	原始大小	压缩后大小	耗时
MediaCodec	38.4MB (15s)	1.7 MB	9.7 s
FFmpeg	38.4MB (15s)	1.7 MB	10.4 s

小米6:

压缩方式	原始大小	压缩后大小	耗时
MediaCodec	38.4MB (15s)	1.7 MB	5.1 s
FFmpeg	38.4MB (15s)	1.7 MB	5.5 s

通过对 FFmpeg 压缩命令进行优化，发现 FFmpeg 和 MediaCodec 压缩速度差异不大，设备的性能越好它们的压缩速度都会越快，FFmpeg 不依赖于具体硬件支持兼容性更佳，**所以使用 FFmpeg 进行压缩。**

3、上传

3.1、限制上传的视频格式

Android 相机拍摄的视频格式是 mp4，但用户可以通过手机相册和 Web 前端入口上传本地其它格式的视频，可能存在上传的视频不支持播放的问题。

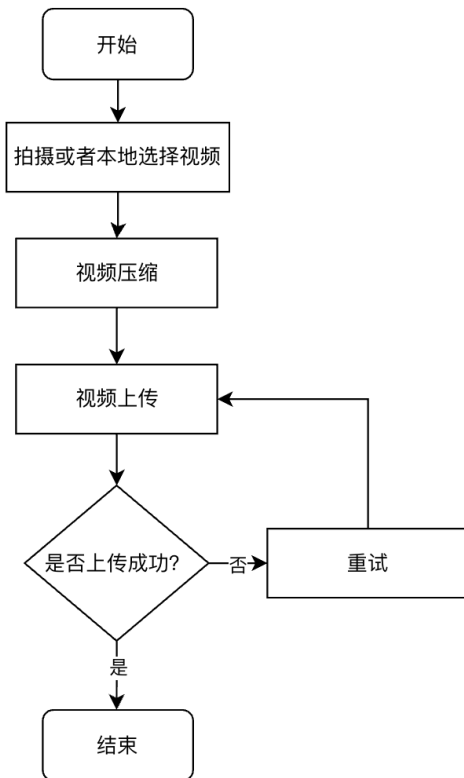
服务端视频存储服务没有接入视频格式转码，所以需要上传前选择的视频做格式做限制：

1. mp4
2. webm

如果后期需要支持更多的格式，可以考虑在服务端接入视频格式转码服务，转成统一格式的视频。

3.2、上传功能

上传功能由在业务方实现，需支持上传失败重试功能。目前WMS业务后端计划存储在公司统一的 USS 服务。



目前 SPX 项目如果丢失了图片地址，可以使用消息推送让 APP 再次上传图片的功能，如果项目中对视频上传需要防丢失功能可以复用图片这套逻辑。

4、播放

视频播放有两个场景：

1. 视频拍摄完成后的预览。
2. 上传后的在线视频。

其中相机拍摄的视频默认是 mp4 格式，上传的视频限制 mp4 和 webm 两种格式，所以播放器只需支持播放 mp4 和 webm 两种格式即可。

下面对比 MediaPlayer、ExoPlayer 和ijkplayer 在 PDA ZEBRA TC20 表现：

播放器	编/解码方式	首帧加载 60s 在线视频	增加的包大小	扩展性	接入复杂度	是否维护
MediaPlayer	仅支持 MediaCodec	5s	0	一般	简单	Android系统自带
ExoPlayer	默认 MediaCodec，支持切换为 FFmpeg	1.5s	2.3MB	好	简单	高频
ijkplayer	默认 FFmpeg，支持切换为 MediaCodec	0.9s	2.2MB	一般	一般	低频

1. MediaPlayer Android 系统自动，不会增加包大小，但是首帧加载较慢。
2. ijkplayer Bilibili开源的视频播放器，首帧加载较快，会增加包大小，维护低频，未解决的 issue 较多。
3. ExoPlayer 是Google 推出的第三方播放器，首帧加载较快，ExoPlayer 扩展性较好，维护团队强大，缺点是会增加包大小。

综上，选择 ExoPlayer。

二期功能

1. 异常监控
2. 压缩速度提升

参考资料

- [Camera2](#)
- [MediaRecorder](#)
- [MediaCodec](#)
- [ExoPlayer](#)
- [Android视频技术探索之旅：美团外卖商家端的实践](#)
- [微信 Android 视频编码爬过的那些坑](#)