

360开测视音频技术探索与应用

简介

- 团队：360QA中心效率平台
 - 专注于质量平台搭建、效率提升、创新技术与落地
- 产品：360开测
 - 技术开放的移动测试平台
- 个人：
 - 李珂
 - 资深测试开发专家&360开测产品负责人

视音频技术需求背景

- 视频流技术

- 如何快速支持Android Q上线
- 如何提高手机屏幕展示清晰度
- 如何提高IOS帧率
- 如何支持虚拟设备

- 音频流技术

- 除了看，我们还需要听

- 虚拟化

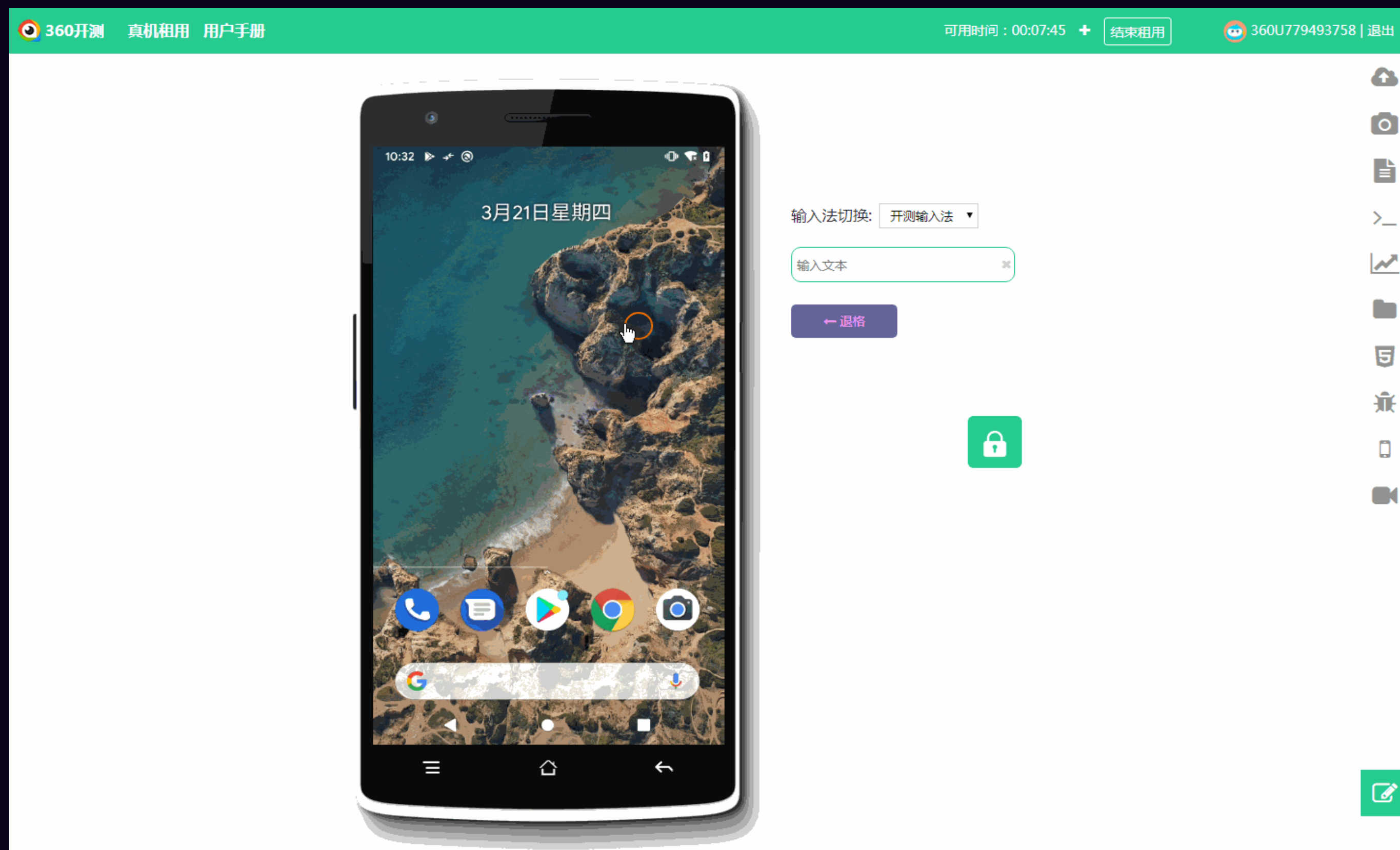
- 虚拟设备能否替代真机

Android Q 全网首发

2019.3.14 Beta 预览版发布



2019.3.15 360开测上线（免费）



Android视频流

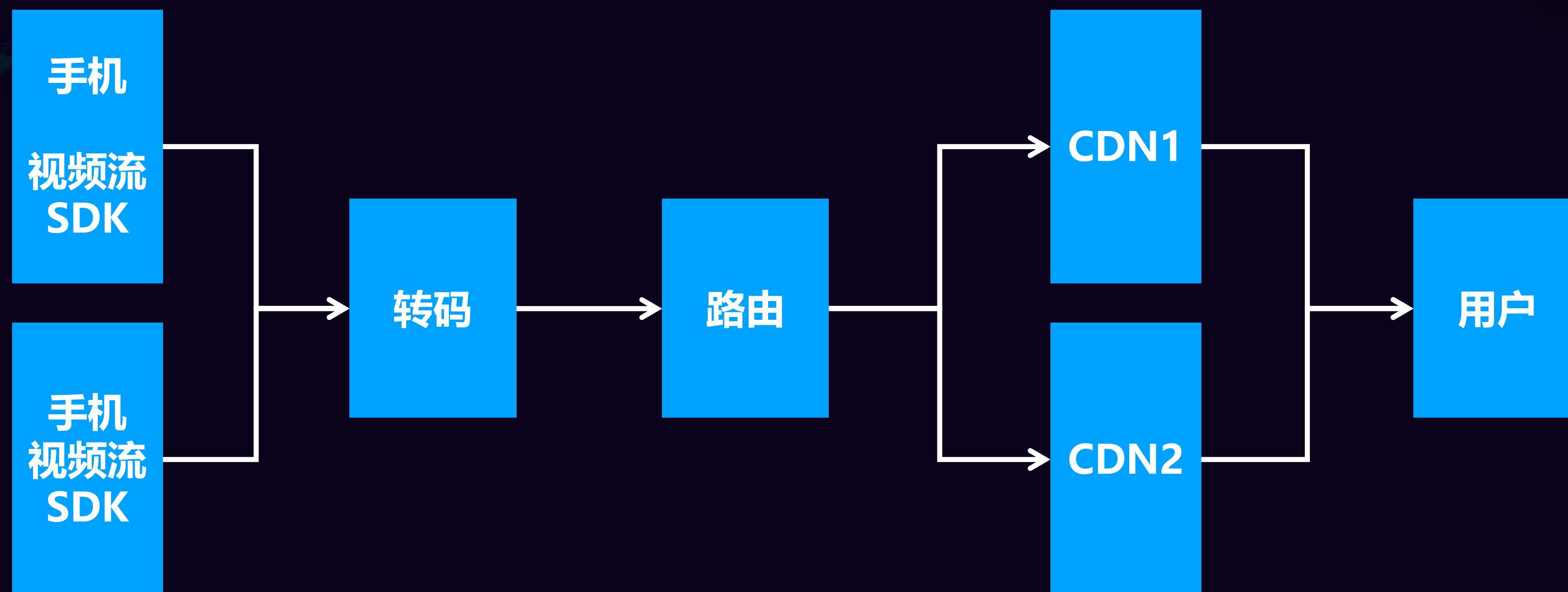
• minicap的不足

- minicap支持最新版Android系统有一定延迟
- 降低带宽&提升清晰度

• 技术背景

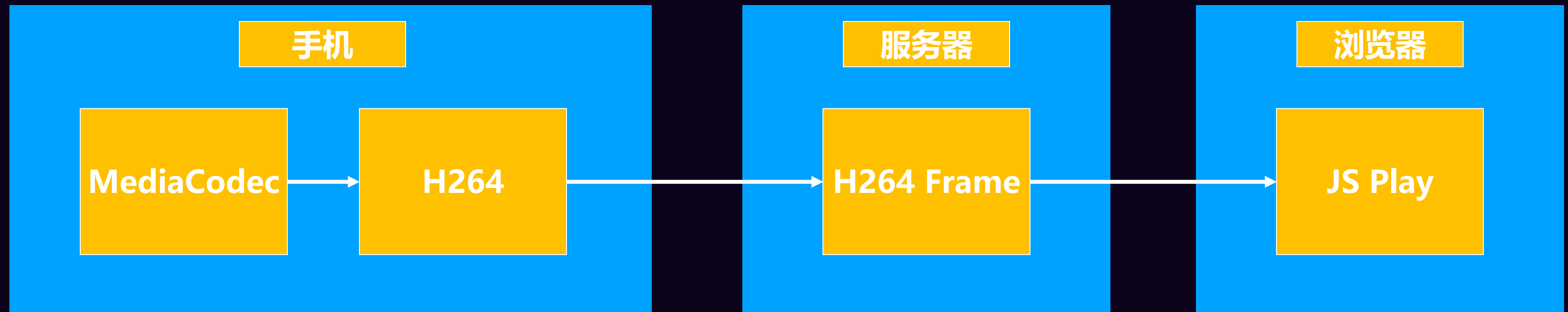
- 直播技术
- Genymotion的虚拟化设备 (<https://github.com/Genymobile/scrcpy>)
- Android MediaCodec

常用的直播架构



	RTMP	HLS	WebRTC
全称	Real Time Message Protocal	HTTP Living Streaming	Web Real-Time Communication
协议	TCP长连接	HTTP短连接	UDP
延时	1-3秒	5-20秒	毫秒级
兼容性	需要使用插件	支持H5	H5

视频流技术框架-毫秒级延迟



H264编码格式

NALU header	NALU body	NALU header	NALU body	NALU header	NALU body	NALU header	NALU body
\0x001	SPS	\x0001	PPS	\x0001	IDR	\0x001	SCLICE

NAL列表

☒ 最多输出5000条

Number	nal_reference_idc	NAL Type(nal_u...	NAL Size(len)
0	96	SPS	18
1	96	PPS	4
2	96	IDR_SLICE	32548
3	64	SLICE	871
4	64	SLICE	344
5	64	SLICE	1733
6	64	SLICE	4409
7	64	SLICE	11845
8	64	SLICE	18744
9	64	SLICE	4257
10	64	SLICE	24766
11	64	SLICE	5888
12	64	SLICE	30210
13	64	SLICE	23746
14	64	SLICE	7765
15	64	SLICE	25184
16	64	SLICE	30098
17	64	SLICE	9479
18	64	SLICE	4170
19	64	SLICE	29437

单击NAL显示详细信息

```
===== NAL =====
forbidden_zero_bit : 0
nal_ref_idc : 3
nal_unit_type : 7 ( Sequence parameter set )
===== SPS =====
profile_idc : 66
constraint_set0_flag : 1
constraint_set1_flag : 0
constraint_set2_flag : 0
constraint_set3_flag : 0
constraint_set4_flag : 0
constraint_set5_flag : 0
reserved_zero_2bits : 0
level_idc : 42
seq_parameter_set_id : 0
chroma_format_idc : 1
residual_colour_transform_flag : 0
bit_depth_luma_minus8 : 0
bit_depth_chroma_minus8 : 0
qp_prime_y_zero_transform_bypass_flag : 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
```


手机端获取视频流数据

- 设置H264编码参数

- createFormat
- 指定编码方式、帧率、码率、关键帧频率等参数

- 采集屏幕原始数据并转码

- streamScreen

- 发送视频流

- encode
- 将H264数据写入到网络连接的文件句柄中

- <https://github.com/Genymobile/scrcpy/blob/master/server/src/main/java/com/genymobile/scrcpy/ScreenEncoder.java>

视频格式设置

```
private static MediaFormat createFormat(int bitRate, int frameRate, int
iFrameInterval) throws IOException {
    MediaFormat format = new MediaFormat();
    format.setString(MediaFormat.KEY_MIME, "video/avc");
    format.setInteger(MediaFormat.KEY_BIT_RATE, bitRate);
    format.setInteger(MediaFormat.KEY_FRAME_RATE, frameRate);
    format.setInteger(MediaFormat.KEY_I_FRAME_INTERVAL, iFrameInterval);
    format.setInteger("level", MediaCodecInfo.CodecProfileLevel.AVCLevel3);
    format.setInteger(MediaFormat.KEY_PROFILE, MediaCodecInfo.CodecProfileLe
vel.AVCProfileBaseline);
    return format;
}
```

服务端解析H264帧数据

- H264是一种视频压缩技术，由NALU单元构成
- NALU Header: 00 00 00 01 (00 00 01) 0x67 (type)
- type= ord(nalu[4]) & 0x1f
 - 7 SPS (偶尔更新)
 - 8 PPS (偶尔更新)
 - 6 IDR (关键帧 手机的全屏数据)
 - 1 SLICE (局部补偿数据)
- 同一手机多开浏览器：发送已获取的SPS PPS IDR

浏览器渲染H264帧数据

- `var p = new Player({ <options> });`
- `p.canvas;` // the canvas – put it where you want it
- `p.decode(<h264 data>);`
- **Broadway目前只支持baseline**

```
format.setInteger("level", MediaCodecInfo.CodecProfileLevel.AVCLevel3);
```

```
format.setInteger(MediaFormat.KEY_PROFILE, MediaCodecInfo.CodecProfileLevel.AVCProfileBaseline);
```

- <https://github.com/mbebenita/Broadway>

视频流技术的兼容性

- 手机端获取数据 Android5.0以上
- Broadway
只支持baseline, 导致视频流使用受到限制
不同手机分辨率选用不同码率标准 (AVC.LEVEL 720P 3.1以下 1080P 4以下)
如何解决: <https://github.com/mbebenita/Broadway/issues/145>
- 虚拟化设备基本都兼容

baseline	最低profile	支持I/P 帧, 只支持无交错(Progressive)和 CAVLC
main	主流profile	支持I/P/B 帧, 无交错 (Progressive) 和交错 (Interlaced) CAVLC 和CABAC
high	高端profile	增加了8x8 内部预测、自定义量化、无损视频编码和更多的YUV 格式

IOS真机租用方案

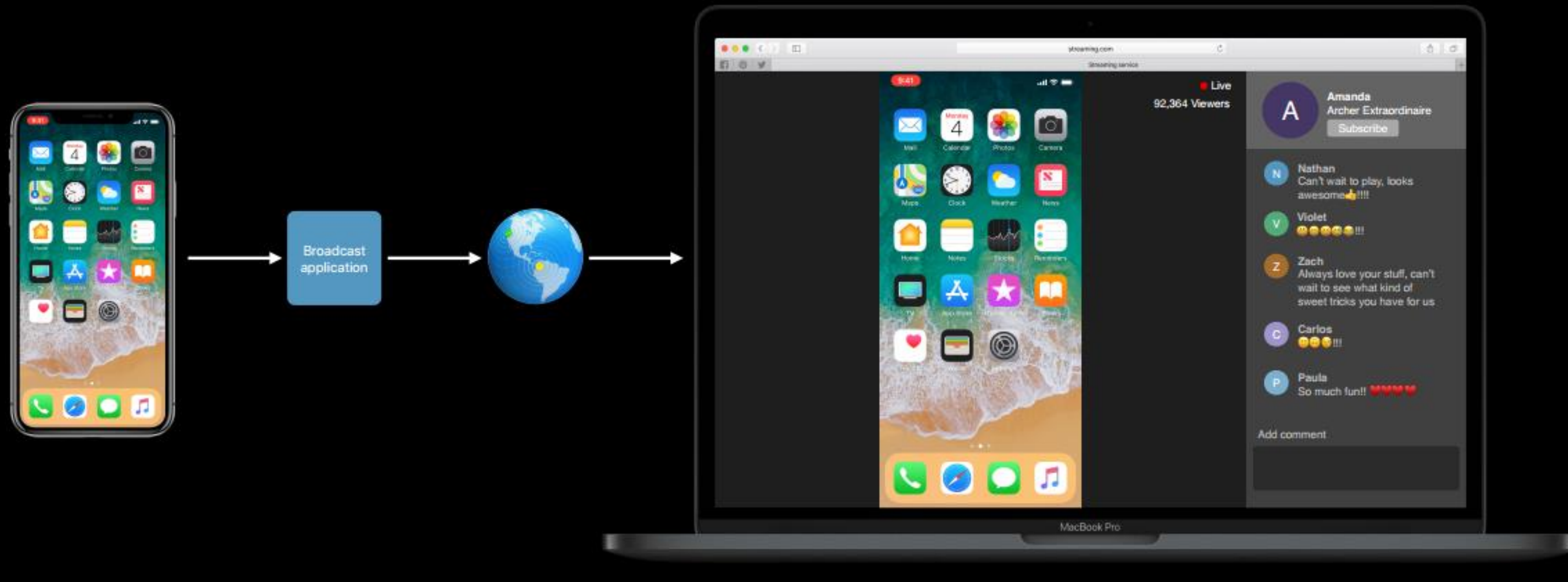
```
- (NSData *)screenCapture
{
    id xcuiScreen = NSStringFromClass(@"XCUIScreen");
    if (xcuiScreen) {
        return (NSData *)[xcuiScreen
valueForKeyPath:@"mainScreen.screenshot.PNGRepresentation"];
    }
    return [[XCAXClient_iOS sharedClient] screenshotData];
}
```

- 截图与压缩导致一个CPU 100%，长时间会被系统回收
- 帧率低而且不稳定，与手机的硬件配置和系统类型有关

IOS视频流方案-ReplayKit

360技术嘉年华

iOS System Broadcast



真机租用-如何听到手机播放的声音?

- 3.5mm耳机插口转USB
- AOA协议 (Genymotion audio feature)

3.5mm耳机插口转USB框架图



- 优势
 - 手机均可适配，通用性强
- 劣势
 - 转接线路的电流干扰，音质受到影响
 - 连接线提高维护成本

3.5mm耳机插口-工程示例

- `// golang`
- `import "github.com/google/gousb"`
- `vid, pid := gousb.ID(0x0d8c), gousb.ID(0x0014)`
- `context := gousb.NewContext()`
- `device, _ := context.OpenDeviceWithVIDPID(vid, pid)`
- `device.SetAutoDetach(true)`
- `configNum, _ := device.ActiveConfigNum()`
- `config, _ := device.Config(configNum)`
- `intf, _ := config.Interface(2, 1)`
- `endPoint, _ := intf.InEndpoint(2)`
- `endPoint.Read(readBuffer)`

AOA协议框架



- AOA_SET_AUDIO_MODE -> AOA_START_ACCESSORY
<https://source.android.com/devices/accessories/aoa2>
- SDL
SDL_AudioCallback (获取音频数据)
SDL_GetNumAudioDevices
SDL_OpenAudioDevice
SDL_PauseAudioDevice(device, 0) (开始读取数据)
- 浏览器播放
<https://github.com/samirkumardas/pcm-player>

AOA协议优劣势

•优势

- 音频数据质量较好

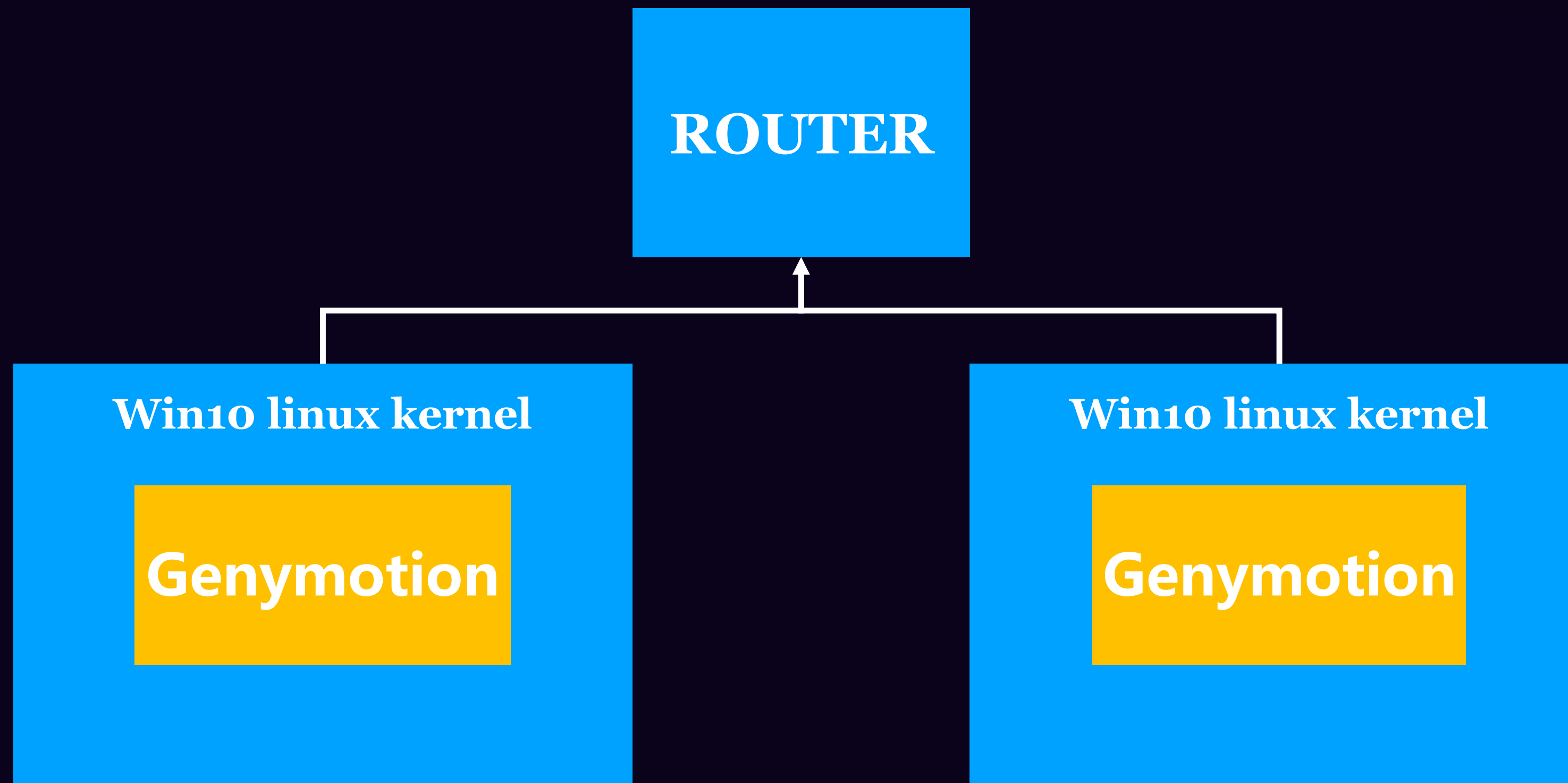
•劣势

- 目前发现少部分手机支持（兼容性有待深入研究）
- AOA协议的启动和停止可能会导致手机死机

虚拟化设备能否替代真机，如何实现

- 部分场景可以替换
 - 测试任务不关心测试设备
 - 降低真机的采购成本和运维成本
- 实现方案
 - 免费Android模拟器
 - 现有技术方案与模拟器的兼容性

虚拟化技术架构



Genymotion在windows系统上的资源消耗更低
win10的linux kernel使服务端程序更加稳定

虚拟化的核心技术与问题

- 核心技术

- 屏幕展示：视频流
- 屏幕操作：Reflect InputManager

- 问题

- 视频流显示有延迟 ($100\text{ms} < \text{delay} < 1000\text{ms}$)
 - 降低分辨率，提高关键帧间隔时间
- 模拟器资源消耗高
 - 16G内存 8核CPU SSD硬盘 独立显卡 启动6个模拟器
- APP兼容性
 - 产品支持x86指令集、安装指令转换插件（不适用所有APP）

虚拟化真机租用演示

360开测 真机租用 用户手册

可用时间：00:06:52 + 结束租用 测试账号 | 退出

优化完成

恭喜你,体检已完成

试试下面精彩推荐吧!

抖音短视频

新鲜原创音乐短视频分享社区

下载

百度

文件大小: 74.5MB

冲马桶的这些常识你中招了吗?

下载

58同城

文件大小: 56.0MB

全套家具配置到家能省一半多钱, 还是最...

下载

火山小视频

文件大小: 27.9MB

下载

输入法切换: 开测输入法

输入文本

← 撤销

🔒

手机基本信息

系统信息:

Android版本: Android 8.0

SDK版本: 26

ABI: x86

内存大小: 994.4MB

是否ROOT: 未ROOT

硬件信息:

厂商: VirtualPhone

Model: Google Pixel 2

CPU类型:

屏幕信息:

宽度: 1080

长度: 1920

360

360开测