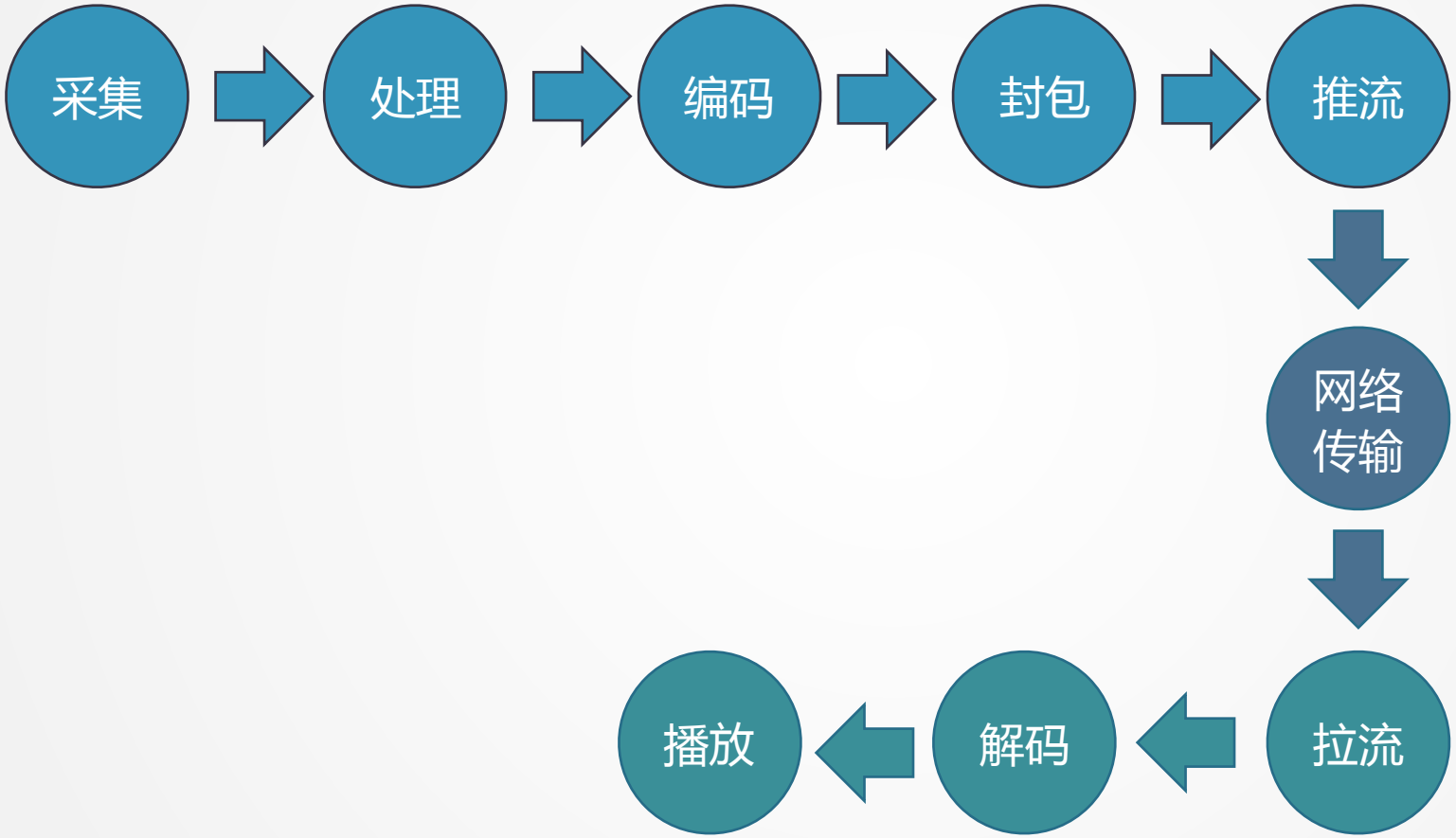
蘑菇街直播实践

蘑菇街-花荣

什么是直播

通俗点讲就是将主播端的视频信息以较低的延时传输到观众端。直播和人们常见的点播在线视频的区别，在于低延时地将主播端的视频信息传输到观众端。

直播简介 – 主要环节

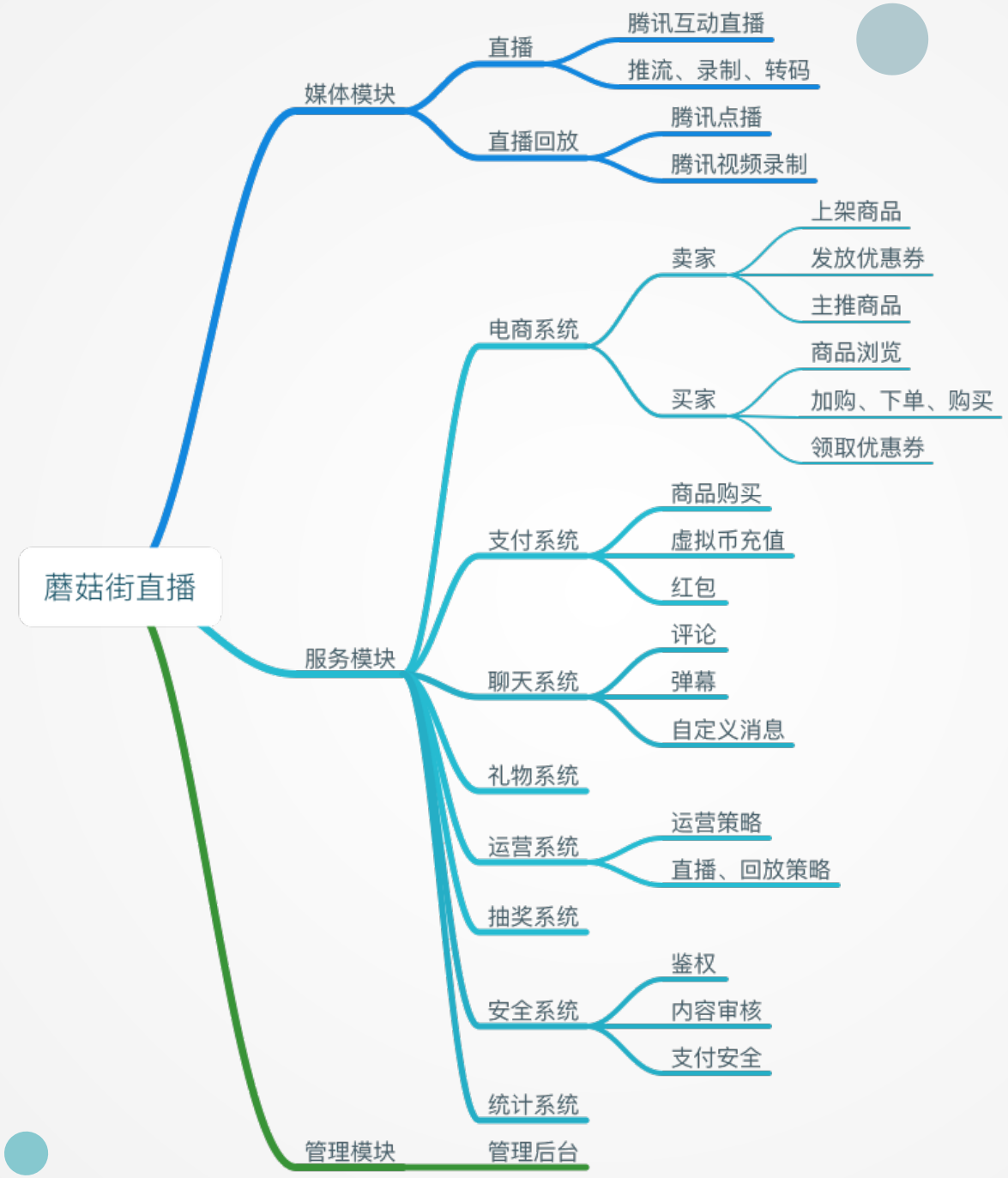


移动端直播形态

- 移动全民直播（映客、花椒）
- 社交软件直播（QQ空间直播，微博一直播、陌陌）
- 电商直播（蘑菇街、淘宝、聚美优品）
- 手游直播（斗鱼TV、触手TV）

蘑菇街直播形态

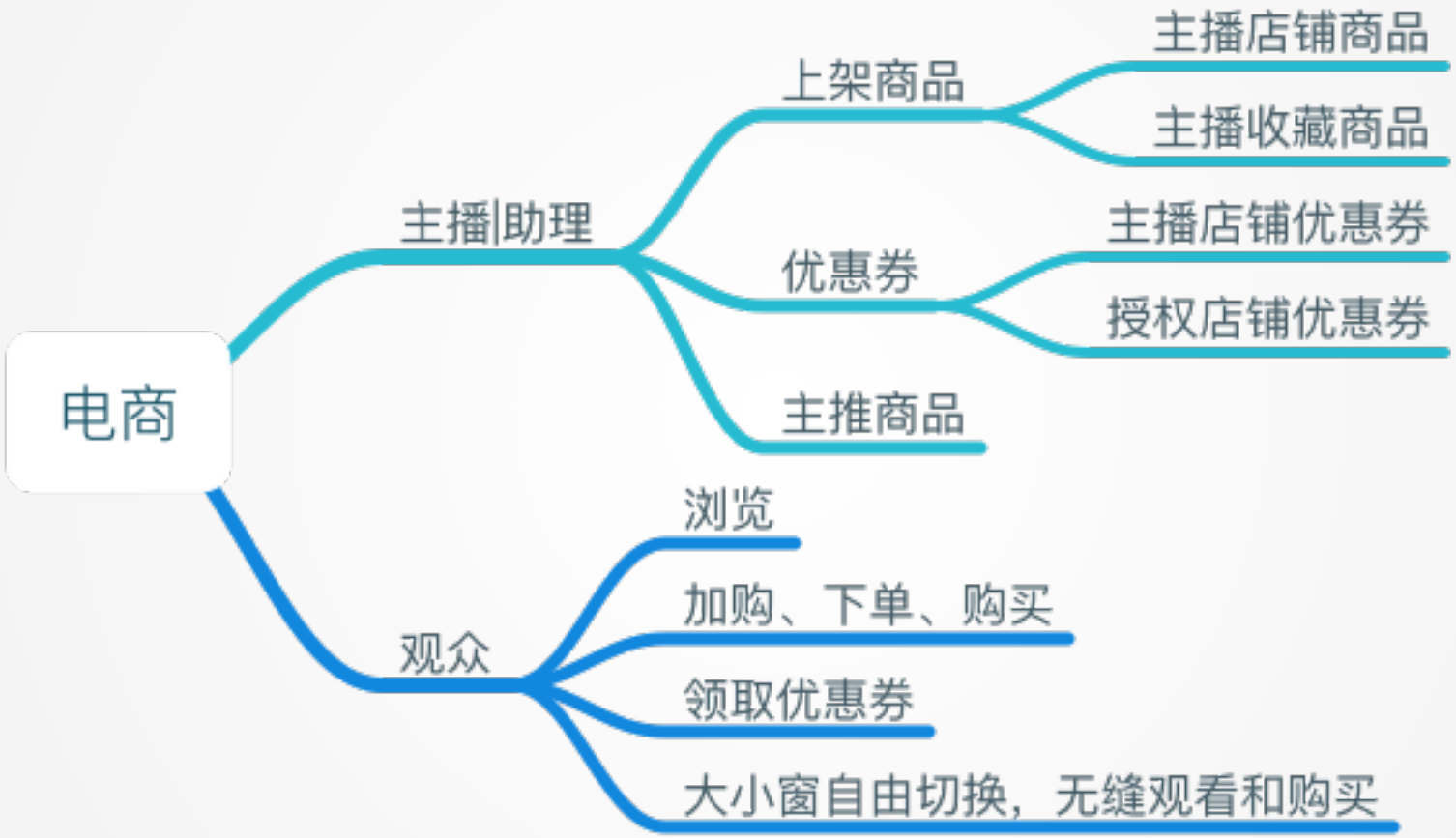
蘑菇街直播的组成


蘑菇街直播
媒体模块
直播
腾讯互动直播
推流、录制、转码
直播回放
腾讯点播
腾讯视频录制
服务模块
电商系统
卖家
上架商品
发放优惠券
主推商品
买家
商品浏览
加购、下单、购买
领取优惠券
支付系统
商品购买
虚拟币充值
红包
聊天系统
评论
弹幕
自定义消息
礼物系统
运营系统
运营策略
直播、回放策略
抽奖系统
安全系统
鉴权
内容审核
支付安全
统计系统
管理模块
管理后台

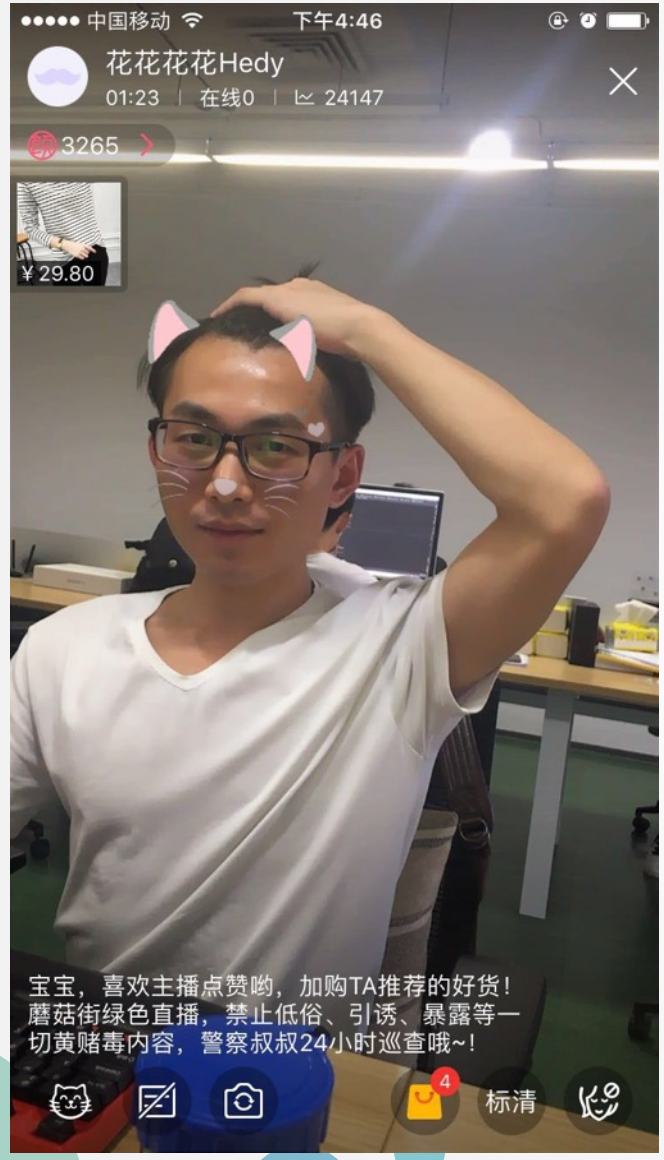
社交


社交
礼物
虚拟币系统
礼物转化颜值
红包
红包转化颜值
聊天
群聊
弹幕
颜值
颜值计算
颜值排行
月排行
总排行
颜值提现
守护榜
直播间守护榜
主播守护榜
点赞榜
答谢
主播答谢送礼观众
用户
粉丝
助理

电商


电商
主播助理
上架商品
主播店铺商品
主播收藏商品
优惠券
主播店铺优惠券
授权店铺优惠券
主推商品
观众
浏览
加购、下单、购买
领取优惠券
大小窗自由切换，无缝观看和购买

电商 – 主播

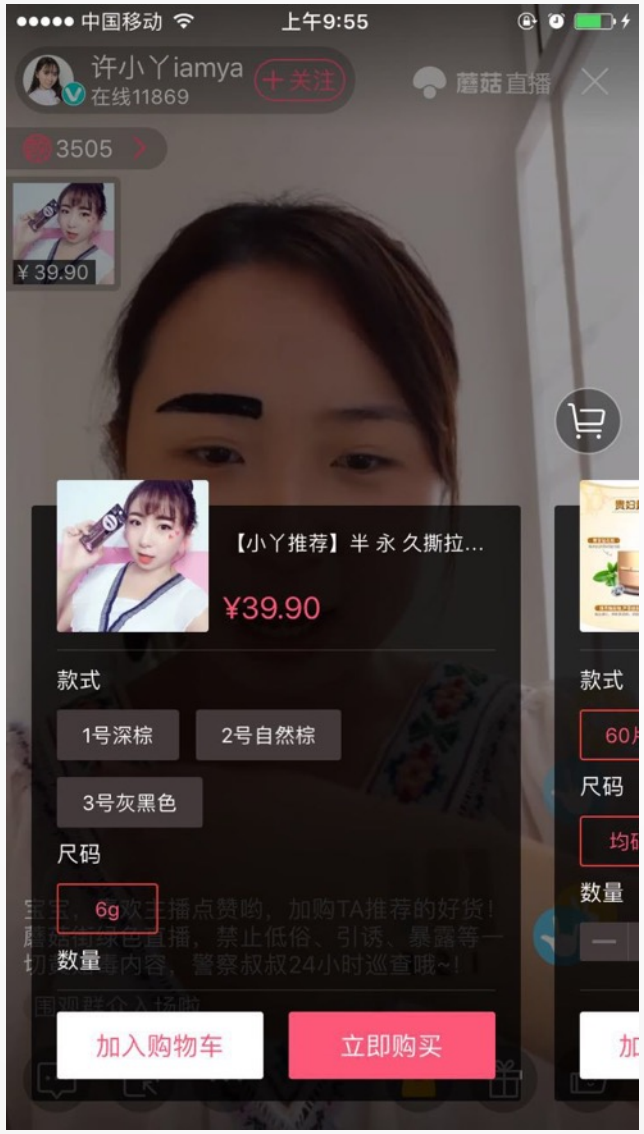

中国移动 下午4:46
花花花花Hedy
01:23 | 在线0 | 24147
3265
¥ 29.80
宝宝，喜欢主播点赞哟，加购TA推荐的好货！
蘑菇街绿色直播，禁止低俗、引诱、暴露等一
切黄赌毒内容，警察叔叔24小时巡查哦~！
4
标清


中国移动 上午10:20
小店商品 | 收藏商品 | 优惠券
初秋新品纯色半...
¥38.5
韩版春秋季条纹...
¥29.8
【么么哒】韩版...
¥29.8
韩版早秋棒球服...
¥87.5
秋季新款 韩版...
¥58
韩版棉质纯色长...
¥29.8
韩版秋季新款学...
¥74.55
秋季新款韩版原...
¥74.55
2016韩版新款秋...
¥62.93
加入售货袋(3)


中国移动 上午10:22
花花花花Hedy
01:55 | 在线0 | 24004
3265
¥ 29.80
韩版棉质纯色长袖T恤女修...
¥42.57
¥29.80
取消展示

电商 – 观众

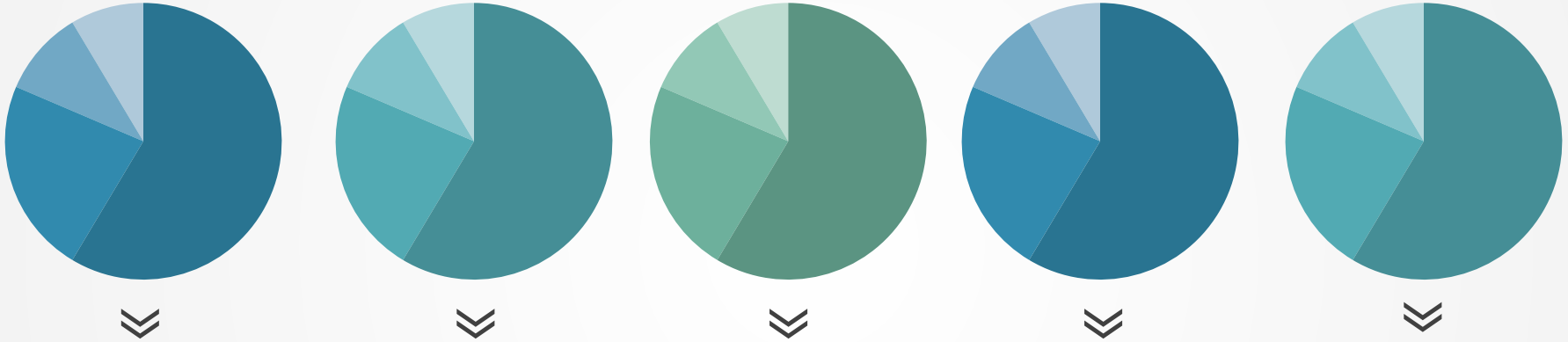

中国移动 上午9:59
叫我吕太太
在线17056
+关注
蘑菇直播
570
¥ 49.90
蘑菇街绿色直播，禁止低俗、引诱、暴露等一切黄赌毒内容，警察叔叔24小时巡查哦~！
时光像个旧美人92 等一群小可爱来了
围观群众入场啦
大傻小傻一起吧 等一群小可爱来了
m_laymi7lyqdu57
身高160体重120可以穿吗
可爱点点Ella 等一群小可爱来了


中国移动 上午9:55
许小丫iamya
在线11869
+关注
蘑菇直播
3505
¥ 39.90
【小丫推荐】半 永 久撕拉...
¥39.90
款式
1号深棕
2号自然棕
3号灰黑色
尺码
6g
数量
加入购物车
立即购买


中国移动 上午9:49
odbo
CC
Brightening
cc cushion
大大家 新品气垫带替换装
¥159.00
销量 67
默认快递
请选择 款式 尺码
私聊
收藏
小店
加入购物车
立即购买

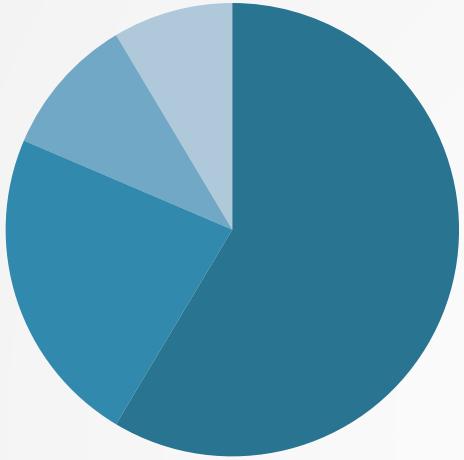
直播遇到的问题

蘑菇街快速接入直播遇到的问题：



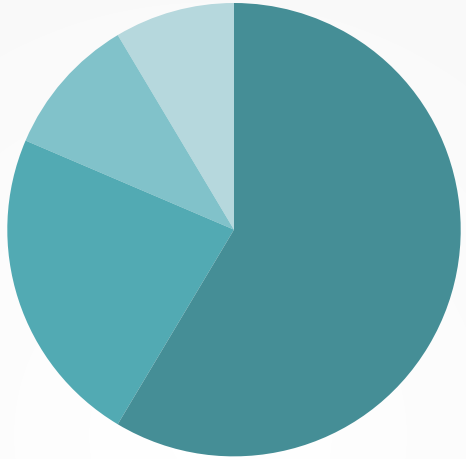
迭代　　云服务　　代码质量　　稳定性　　性能

迭代过快



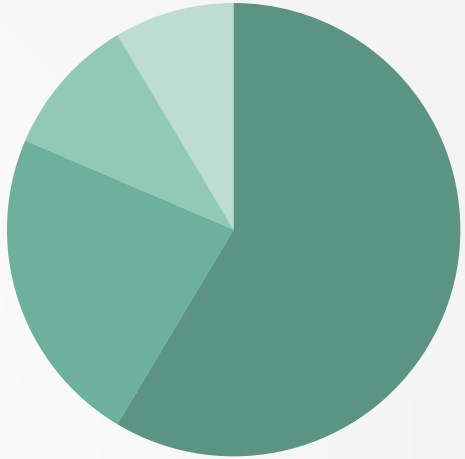
需求调研不充分

直播从立项到上线差不多三个星期



新需求

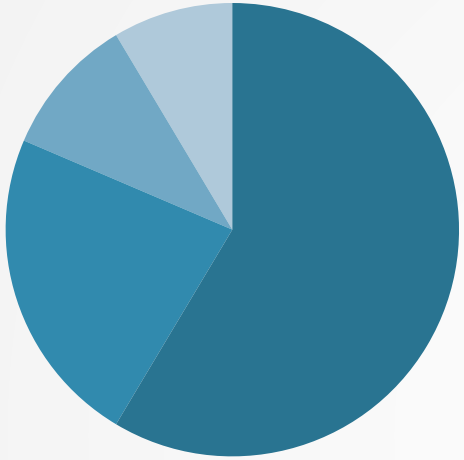
直播保持着两周一个版本上线的班车计划，实际开发周期一周



新老bug

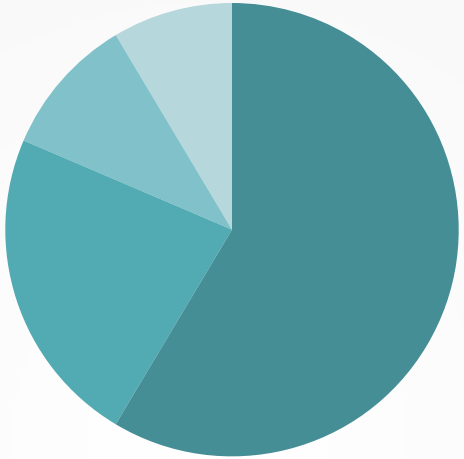
快速迭代上线的直播的之后几个版本，均是新老bug并存的局面

三方服务不稳定



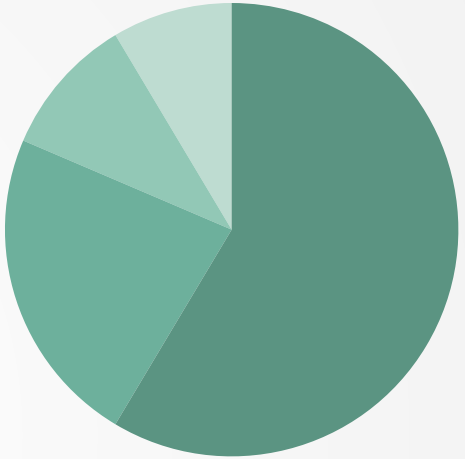
文档不全

接入文档不全或存在疑点，导致接入后因调用方式或者线程时序带来了不少问题



客户端SDK不稳定

SDK内部存在较多异常和crash问题



云服务不稳定

云服务不稳定，时常出现服务不可用情况

稳定性优化

稳定性存在的问题：

无法正常观看直播，闪退

- 内存泄漏
- 客户端SDK不稳定
- 存在硬件兼容性问题
- 多线程情况复杂

稳定性优化

Code Review & 代码规范

优化代码质量

统一编码风格

静态分析 & 内存泄漏检测

接入静态分析

Debug模式下接入MLeaksFinder

日志查看上报

CocoaLumberJack主流程打点

日志查看、发送

日志定向上报分析

稳定性优化 – 案例：block嵌套

多层block嵌套，大屏幕写起来还挺爽，但是伴随着较差的阅读性和可维护性，
例如回调丢失，循环引用等问题

```
// 获取房间号
[self fetchRoomID:^(NSString *roomID, MGJLiveIMType imType, NSError *error) {
    MGBenchStep(@"创建直播",@"MGJLive: 获取房间号成功");
    @strongify(self);
    if (error) errorBlock(error);
    else {
        switch (_imType) {
            case MGJLiveIMTypeTencent:
                _chatRoomManager = [MGJChatRoomManager tencentChatRoomManager];
                break;
            default:
                _chatRoomManager = [MGJChatRoomManager imChatRoomManager];
                break;
        }
        // 登录 IM
        [self loginIMWithUserID:[MGJUser shareInstance].userAuthenticate.uid
                        handle:^(NSError *error) {
                            MGBenchStep(@"创建直播",@"MGJLive: 登录IM成功");
                            if (error) errorBlock(error);
                            else {

                                // 初始化房间
                                [self initVideoRoom:roomID
                                               type:MGJLiveBroadcastTypeActor
                                             handle:^(NSError *error) {
                                                 MGBenchStep(@"创建直播",@"MGJLive: 初始化房间成功");
                                                 if (error) errorBlock(error);
                                                 else {

                                                     // 获取旁路 url
                                                     [self getLiveURLWithType:AV_ENCODE_HLS
                                                                       roomID:roomID
                                                                  channelName:[NSString stringWithFormat:@"%@:%@", [MGJUser shareInstance].userAuthenticate.uname, _roomID]
                                                            channelDescription:@""
                                                                       handle:^(NSString *url, NSError *error) {
                                                                           MGBenchStep(@"创建直播",@"MGJLive: 旁路开启成功");
                                                                           if (error) errorBlock(error);
                                                                           else {
                                                                               // 创建群聊
                                                                               [self createChatRoomWithRoomId:roomID
                                                                                                        block:^(NSString *chatRoomID, NSError *error) {
                                                                                                            MGBenchStep(@"创建直播",@"MGJLive: 创建群聊成功");
                                                                                                            if (error) errorBlock(error);
                                                                                                            else {
                                                                                                                successBlock(roomID, chatRoomID, url);
                                                                                                            }
                                                                                                        }];
                                                                           }
                                                                       }];
                                                 }
                                             }];
                            }
                        }];
    }
}];
```

稳定性优化 – 案例：block嵌套优化方案

修改为单一职责的方法调用，新增了代码的可阅读性和可维护性，同时不易造成回调丢失，内存泄漏等问题

```objc
/**
 *  登录IM
 */
- (void)loginIMWithSignature:(NSString *)signature
{
    /**
     *  建房流程2
     */
    self.createRoomStep = 2;
    LiveSDKLogInfo(@"建房流程2 loginIM");

    MGJLVWeakSelf;
    [MGJLVLoginManager login:self.userId userSign:signature succ:^(id data) {
        MGJLVStrongSelf
        [self startVideoContext];

    } fail:^(NSError *error) {
        MGJLVStrongSelf;
        NSError *err =  createMGJLVRoomErrorMaskMessageFrom(MGJLVRoom_Error_Create_Room_Fail, error);
        [self onCreateRoomFailedHandler:err];
        LiveSDKLogError(@"建房流程2 error:%@", err);
    }];
}

/**
 *  开启视频上下文
 */
- (void)startVideoContext
{
    /**
     *  建房流程3
     */
    self.createRoomStep = 3;
    LiveSDKLogInfo(@"建房流程3 startVideoContext");

    MGJLVWeakSelf
    [[MGJLVVideoContextManager sharedInstance] startVideoContext:self.userId complete:^(QAVResult result) {
        MGJLVStrongSelf
        if (result != QAV_OK) {
            NSError *error = createMGJLVVideoError(MGJLVVideo_Error_Start_Context_Fail, result, @"开启视频上下文失败");
            [self onCreateRoomFailedHandler:error];
            LiveSDKLogError(@"建房流程3 error:%@", error);
        }
        else {
            [self onLoginSuccessHandler];
        }
    }];
}

- (void)onLoginSuccessHandler
{
    LiveSDKLogInfo(@"建房流程onLoginSuccessHandler");

    [self createVideoRoom];
    [self createChatRoom];
}
```

稳定性优化 – 内存泄漏

- Instruments进行内存泄漏的检测
- iOS客户端使用MLeaksFinder针对直播组件进行了Debug下的内存泄漏检测，将内存泄漏扼杀在开发阶段。

稳定性优化 – 内存泄漏**MLeaksFinder**

```objc
#pragma mark - MLeaksFinder
#ifdef ENABLE_LEAK_FINDER
- (BOOL)willDealloc {
    if (![super willDealloc]) {
        return NO;
    }

    // components
    MLCheck(self.viewerRoomInfo);
    MLCheck(self.closeComponent);
    MLCheck(self.containerComponent);
    MLCheck(self.hostInfoComponent);
    MLCheck(self.toolbarComponent);
    MLCheck(self.praiseComponent);
    MLCheck(self.giftPlayerComponent);
    MLCheck(self.debugComponent);
    MLCheck(self.giftPickerComponent);
    MLCheck(self.barrageComponent);
    MLCheck(self.inputBarComponent);
    MLCheck(self.biglegUserComponent);
    MLCheck(self.shareComponent);
    MLCheck(self.leftContainerComponent);
    MLCheck(self.goodsComponent);
    MLCheck(self.commentTableComponent);
    MLCheck(self.luckBagComponent);
    MLCheck(self.goodsCouponRushComponent);
    MLCheck(self.hostBusyComponent);
    MLCheck(self.networkAlertComponent);

    // services
    MLCheck(self.paymentService);

    return YES;
}
#endif
```

Mute Assertion

If your class is designed as singleton or for some reason objects of your class should not be dealloced, override `-(BOOL)willDealloc` in your class by returning NO.

```objc
- (BOOL)willDealloc {
    return NO;
}
```

Find Leaks in Other Objects

MLeaksFinder finds leaks in UIView and UIViewController objects by default. However, you can extend it to find leaks in the whole object graph rooted at a UIViewController object.

```objc
- (BOOL)willDealloc {
    if (![super willDealloc]) {
        return NO;
    }

    MLCheck(self.viewModel);
    return YES;
}
```

稳定性优化 – 流程打点

日志内部使用CocoaLumberJack实现，自定义了log等级，

log文件大小，回滚频率，数量分别为2MB，24h，10个。

```
typedef NS_OPTIONS(NSUInteger, MGJLiveLogFlag) {
    MGJLiveLogFlagError   = (1 << 5), // 0...0000100000
    MGJLiveLogFlagWarning = (1 << 6), // 0...0001000000
    MGJLiveLogFlagInfo    = (1 << 7), // 0...0010000000
    MGJLiveLogFlagDebug   = (1 << 8), // 0...0100000000
    MGJLiveLogFlagVerbose = (1 << 9), // 0...1000000000
};

typedef NS_ENUM(NSUInteger, MGJLiveLogLevel) {
    MGJLiveLogLevelOff     = 0,
    MGJLiveLogLevelError   = (MGJLiveLogFlagError),
    MGJLiveLogLevelWarning = (MGJLiveLogLevelError   | MGJLiveLogFlagWarning),
    MGJLiveLogLevelInfo    = (MGJLiveLogLevelWarning | MGJLiveLogFlagInfo),
    MGJLiveLogLevelDebug   = (MGJLiveLogLevelInfo    | MGJLiveLogFlagDebug),
    MGJLiveLogLevelVerbose = (MGJLiveLogLevelDebug   | MGJLiveLogFlagVerbose),
    MGJLiveLogLevelAll     = MGJLiveLogLevelVerbose //只是LiveLog中的所有Level
};

typedef NS_ENUM(NSUInteger, MGJLiveLogContext) {
    MGJLiveLogContextDefault = 10,
    MGJLiveLogContextCoreSDK = 11,
    MGJLiveLogContextLiveSDK = 12,
    MGJLiveLogContextMWP     = 13,
};
```

稳定性优化 – 流程打点

使用相应的log等级，在需要的地方打印log即可

```
/**
 *  登录IM
 */
- (void)loginIMWithSignature:(NSString *)signature
{
    /**
     *  建房流程2
     */
    self.createRoomStep = 2;
    LiveSDKLogInfo(@"建房流程2 loginIM");

    MGJLVWeakSelf;
    [MGJLVLoginManager login:self.userId userSign:signature succ:^(id data) {
        MGJLVStrongSelf
        [self startVideoContext];

    } fail:^(NSError *error) {
        MGJLVStrongSelf;
        NSError *err =  createMGJLVRoomErrorMaskMessageFrom(MGJLVRoom_Error_Create_Room_Fail, error);
        [self onCreateRoomFailedHandler:err];
        LiveSDKLogError(@"建房流程2 error:%@", err);
    }];
}
```

稳定性优化 – 流程打点

解决发生异常或者crash
无法定位问题的痛点

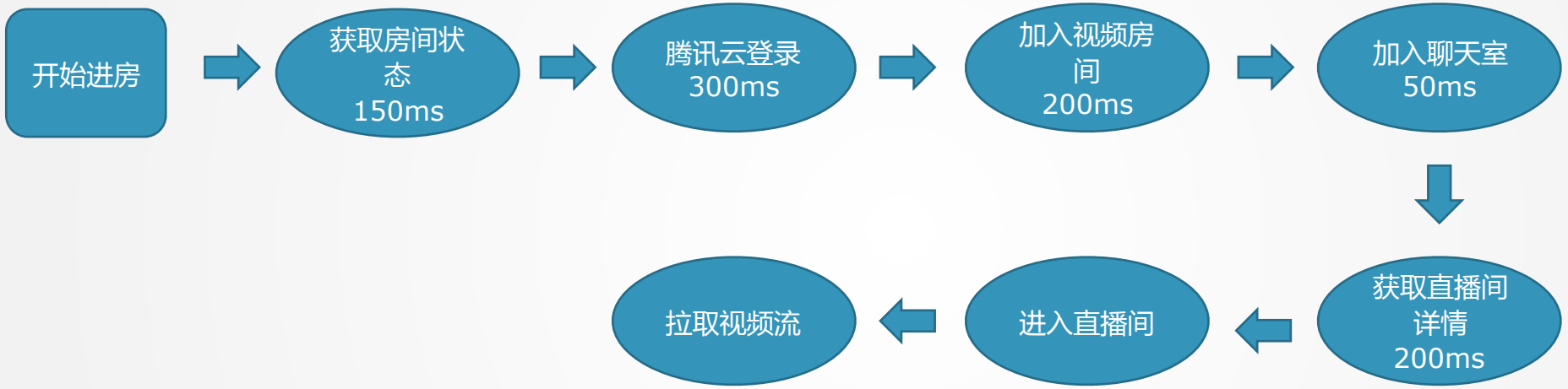
性能优化

问题五：性能

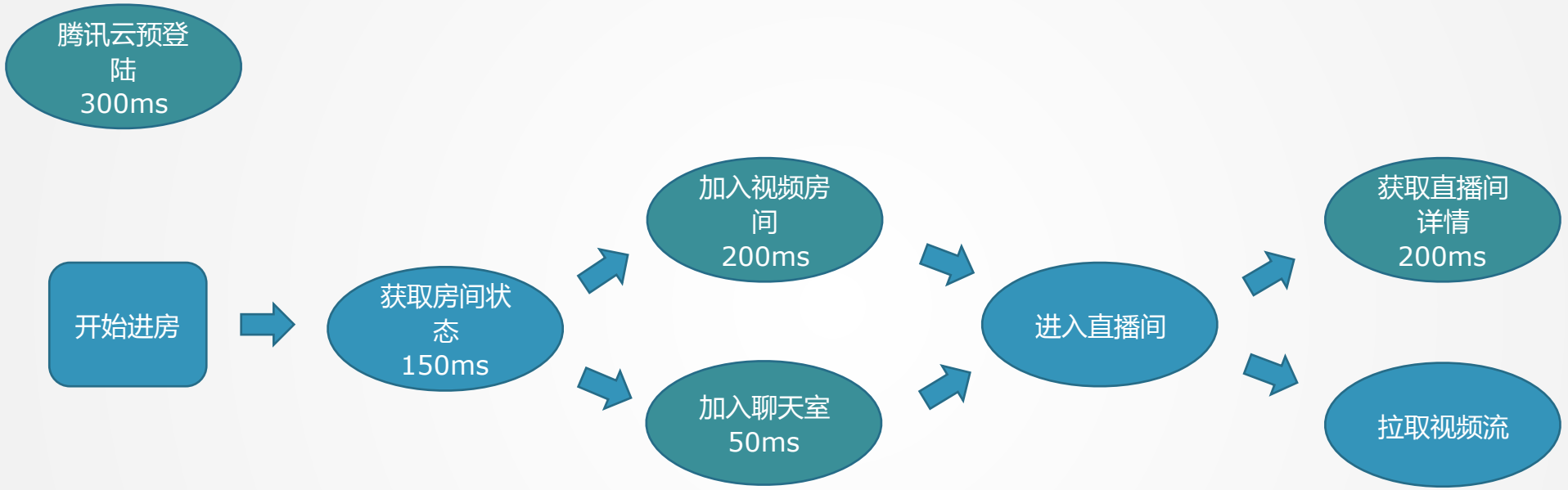
初期不支持硬件编解码	评论列表刷新过于频繁	点赞、礼物、弹幕渲染	高并发下打点
不支持硬件编解码导致CPU和内存占用过高，容易发烫、卡顿	评论列表在高并发情况下依然保持收到消息就立即处理刷新，导致很多资源的损耗	点赞、礼物、弹幕在高并发下存在无法复用、离屏渲染、图片缓存等问题	高并发下的基础库打点存在磁盘读写和网络请求频繁的问题，CPU负荷重

性能优化：进房速度慢

串行的进房流程



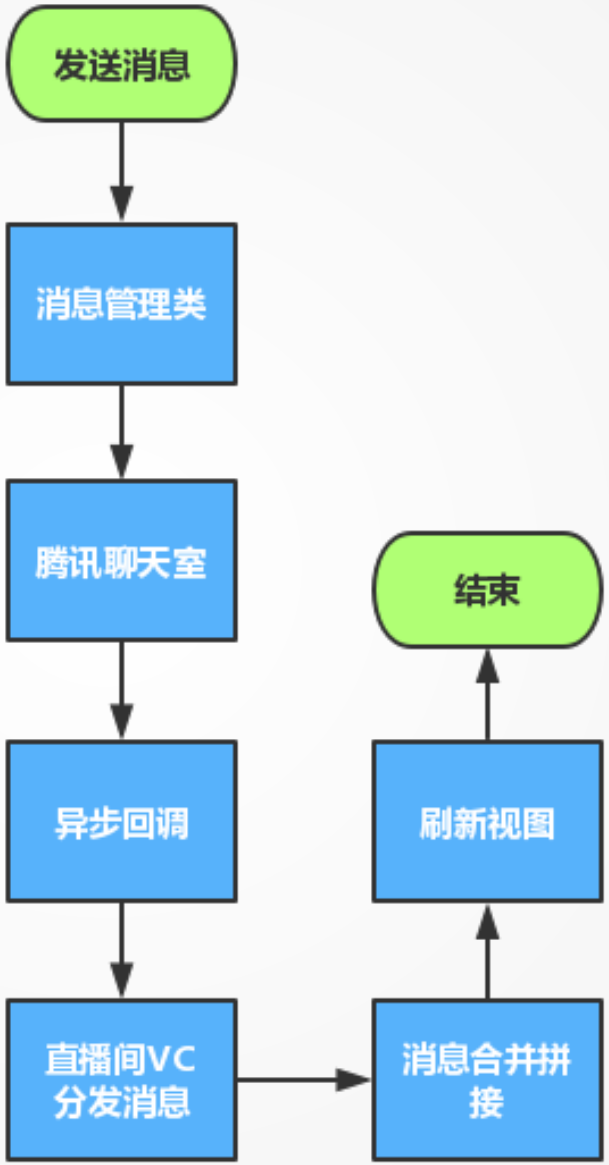
性能优化：进房速度慢优化方案



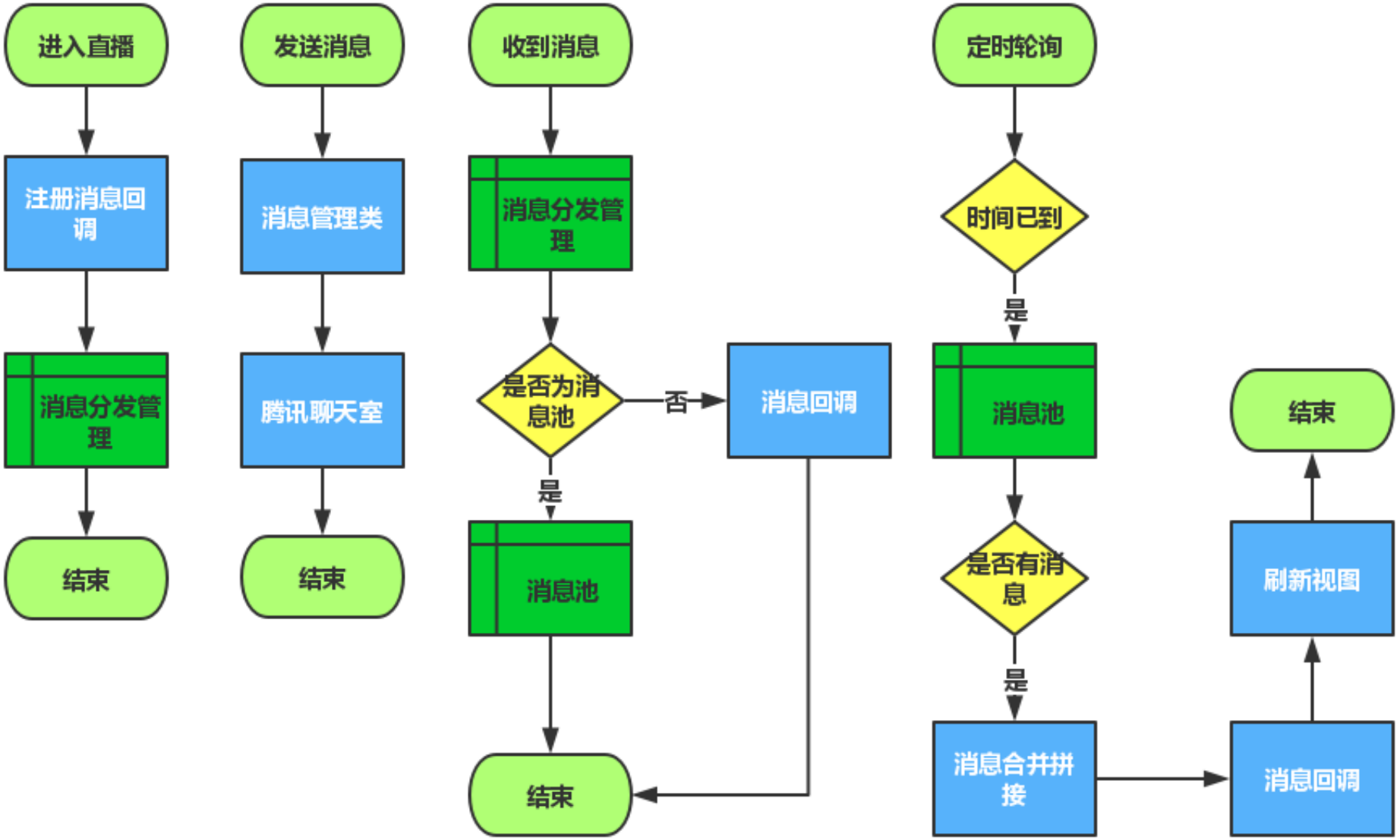
性能优化：进房速度慢优化结果

存在的三个问题

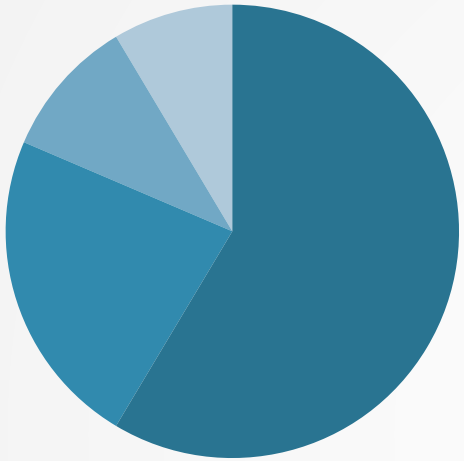
- 触发时机过于频繁
- 消息缺少缓存池
- 没有缓存列表的高度计算



性能优化 – 消息优化方案

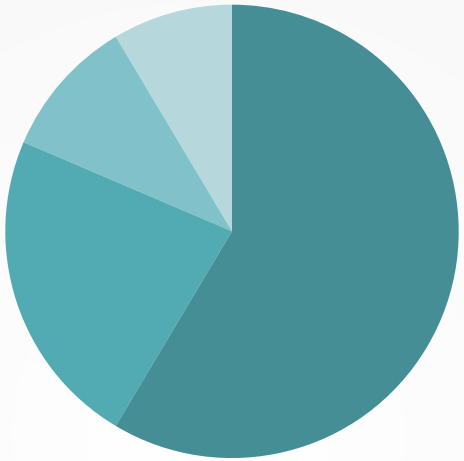

进入直播
注册消息回调
消息分发管理
结束
发送消息
消息管理类
腾讯聊天室
结束
收到消息
消息分发管理
是否为消息池
否
消息回调
是
消息池
结束
定时轮询
时间已到
是
消息池
是否有消息
是
消息合并拼接
消息回调
刷新视图
结束

性能优化 – 动画



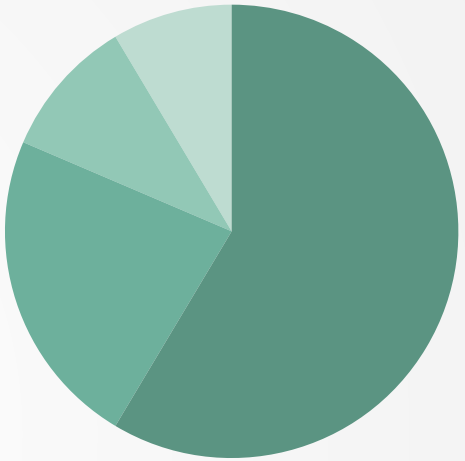
动画控件复用

优化方案：复用动画控件，减少内存的分配以及防止内存爆发式增长



离屏渲染严重

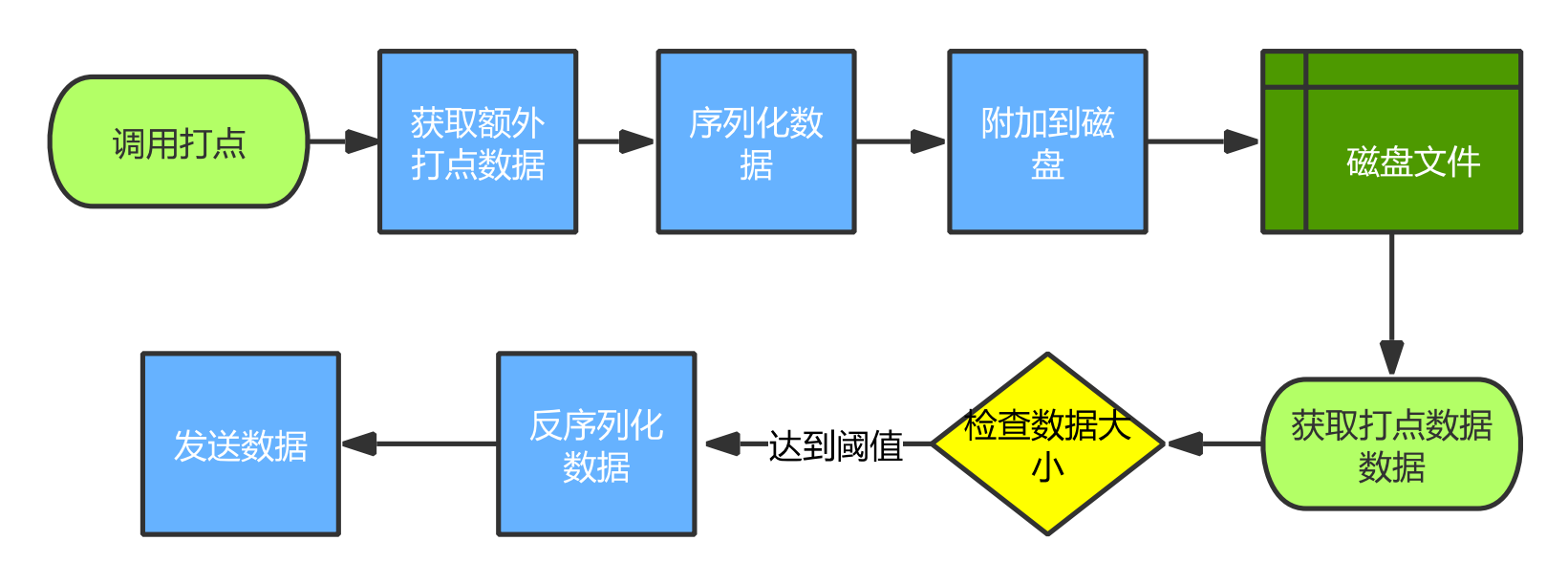
优化方案：使用instruments检测离屏渲染严重的视图，例如去除圆角，栅格化等损耗CPU性能的方式



序列帧图片缓存

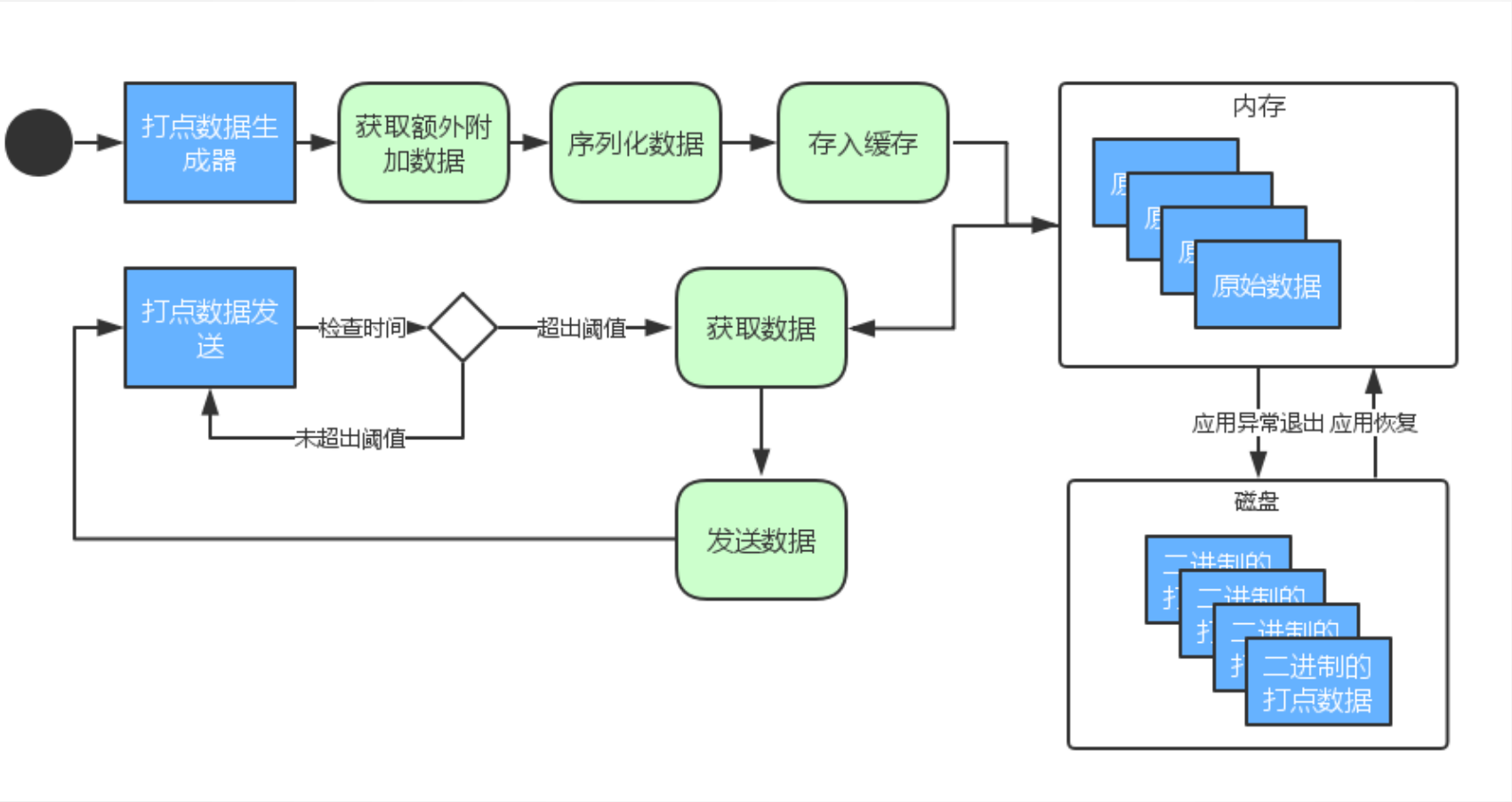
解决方案：使用YYImage三方库处理了直播内部的大量序列帧动画，及时释放动画的图片缓存

性能优化 – 打点



- 每次打点都要读写磁盘，IO和CPU负荷重
- 打点数据超过1K就要发送，短时间频繁打点导致网络频繁发送请求

性能优化 – 打点

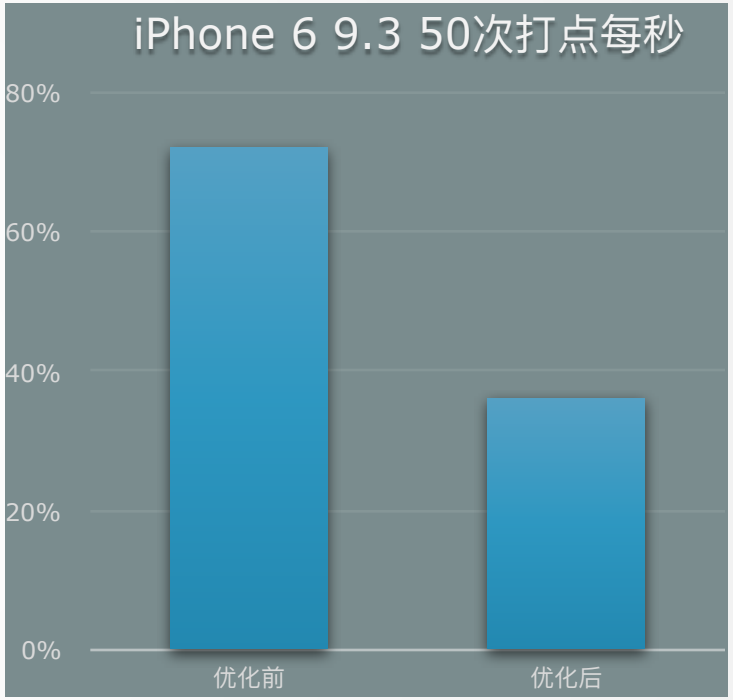

打点数据生成器
获取额外附加数据
序列化数据
存入缓存
内存
原始数据
打点数据发送
检查时间
超出阈值
获取数据
未超出阈值
应用异常退出
应用恢复
发送数据
磁盘
二进制的打点数据

性能优化 – 打点

移除每次都要进行的磁盘读写操作

将打点数据放入内存中作为缓存

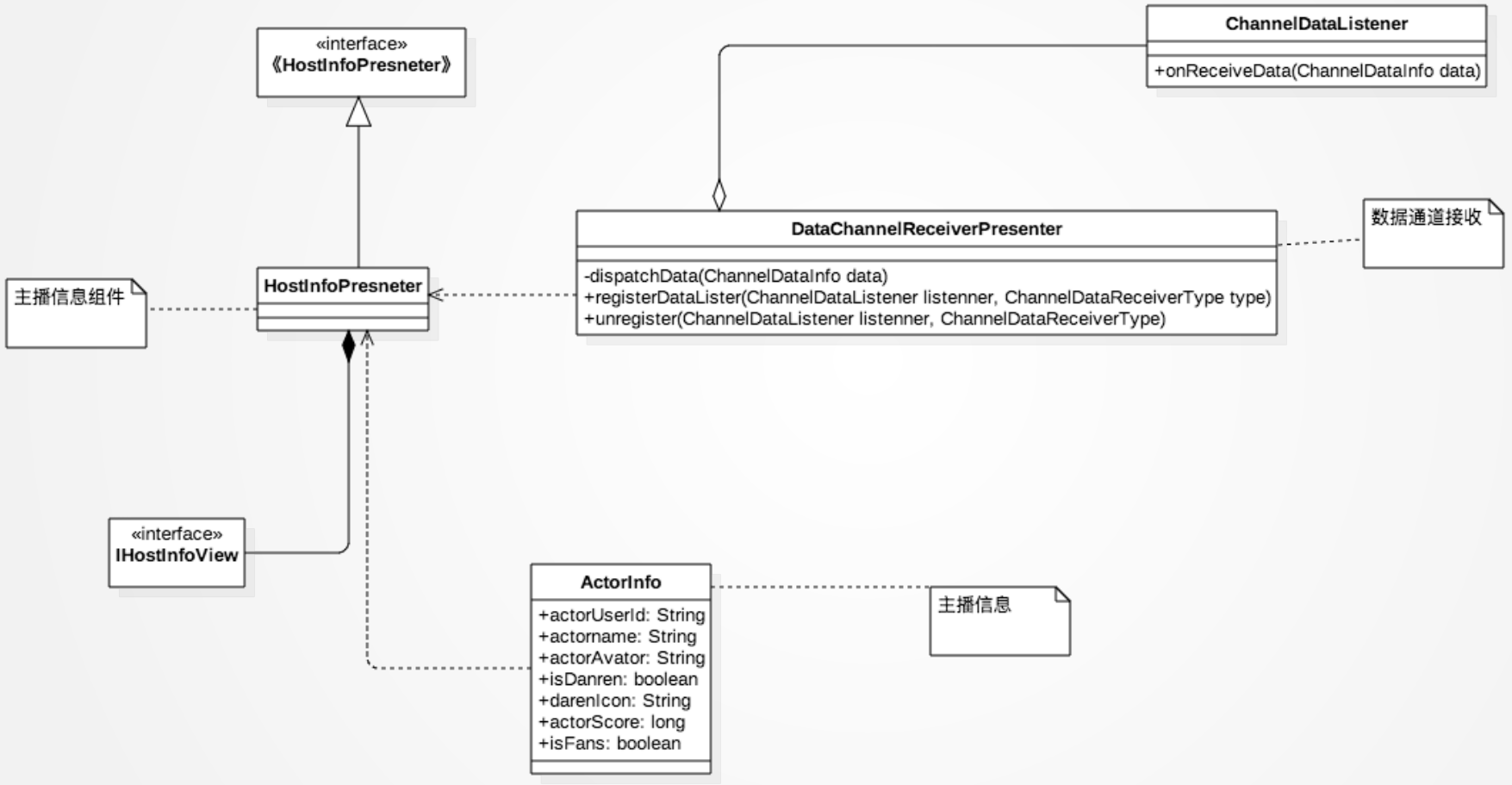
使用一个队列从缓存中读取打点数据发送，避免频繁请求网络的情况



组件化

直播组件化

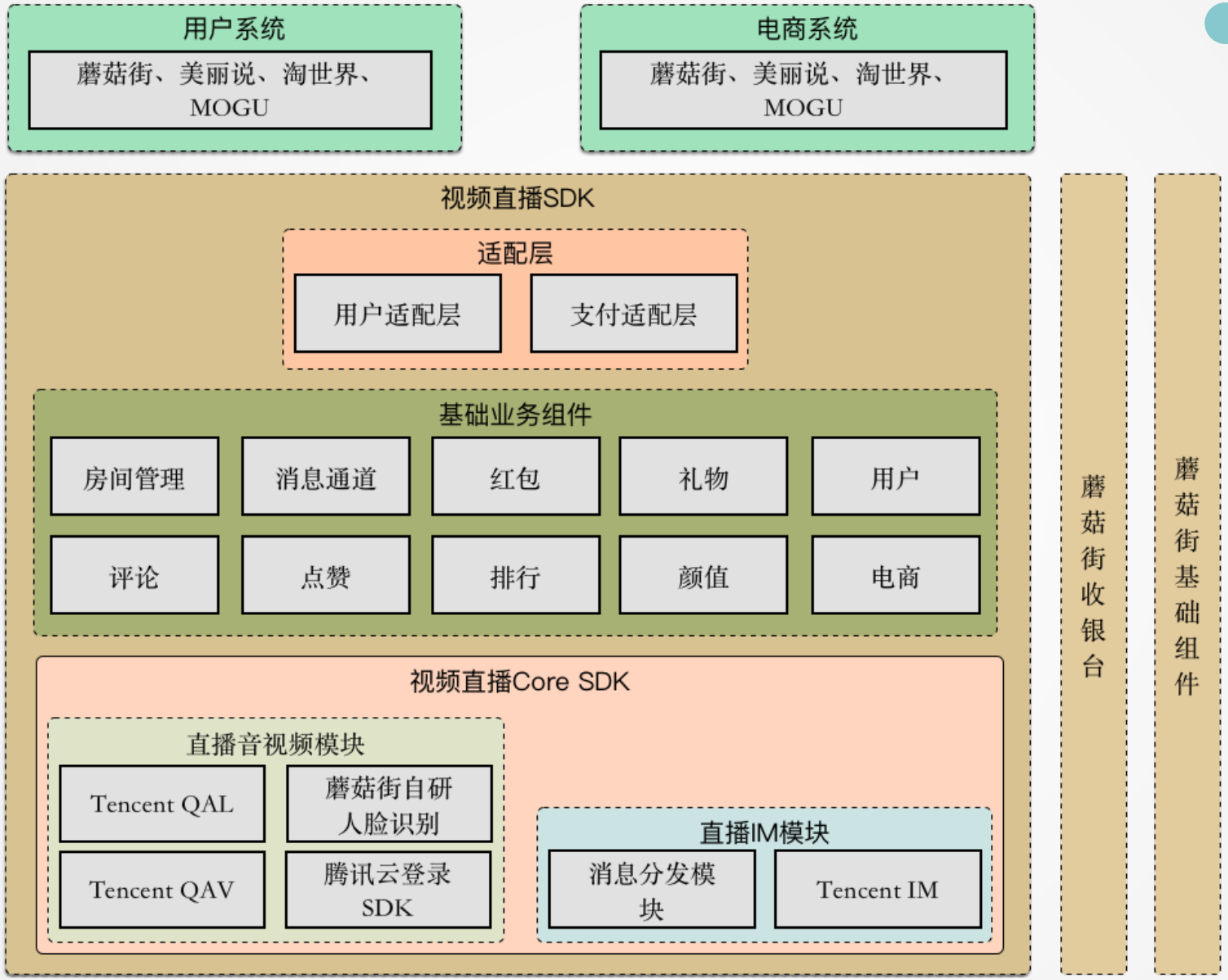
直播组件化 – 案例直播信息



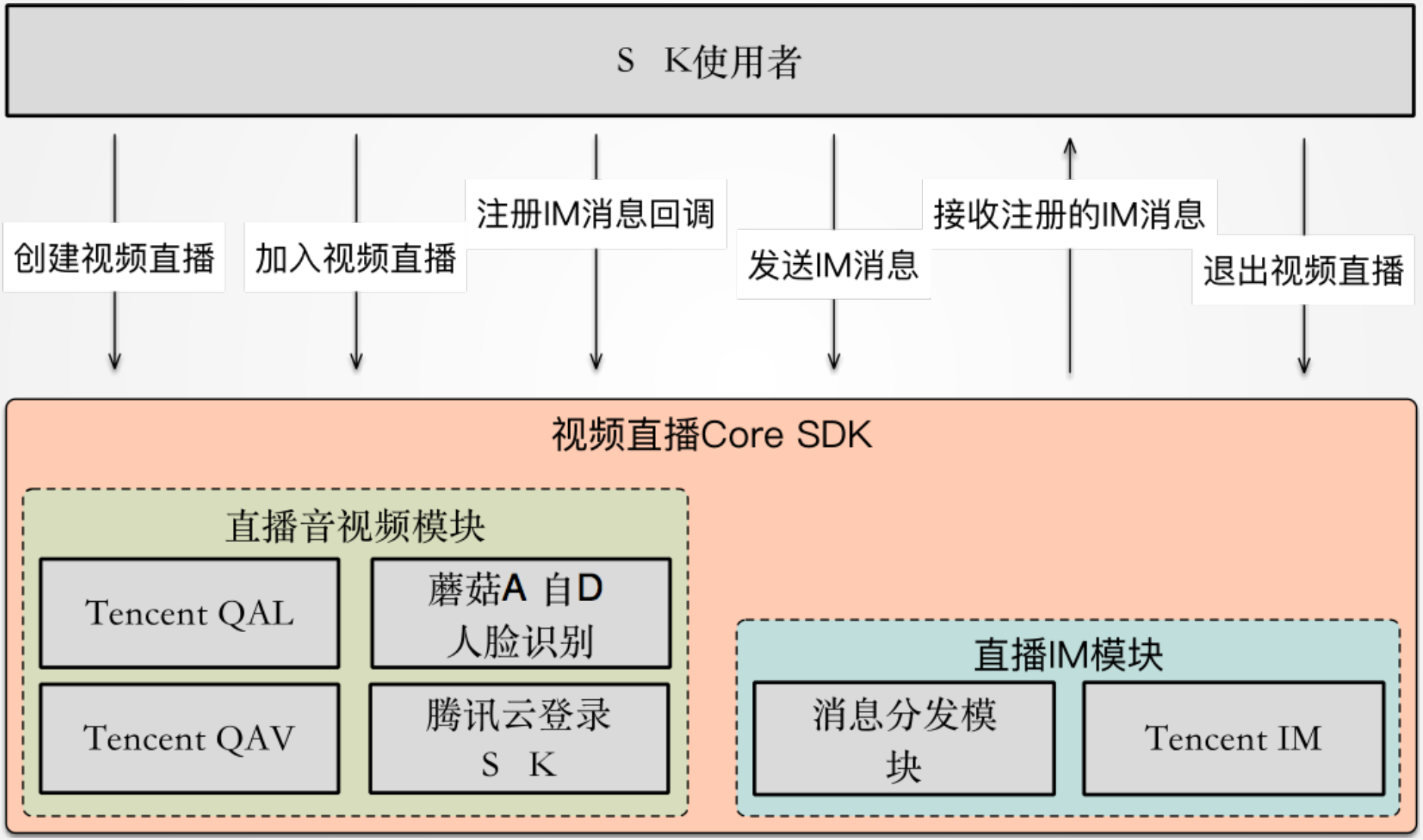
SDK化

- 降低集团内其他App接入成本
- 统一的接口和文档
- 良好的框架设计和扩展性
- 业务功能可配置化和UI定制化

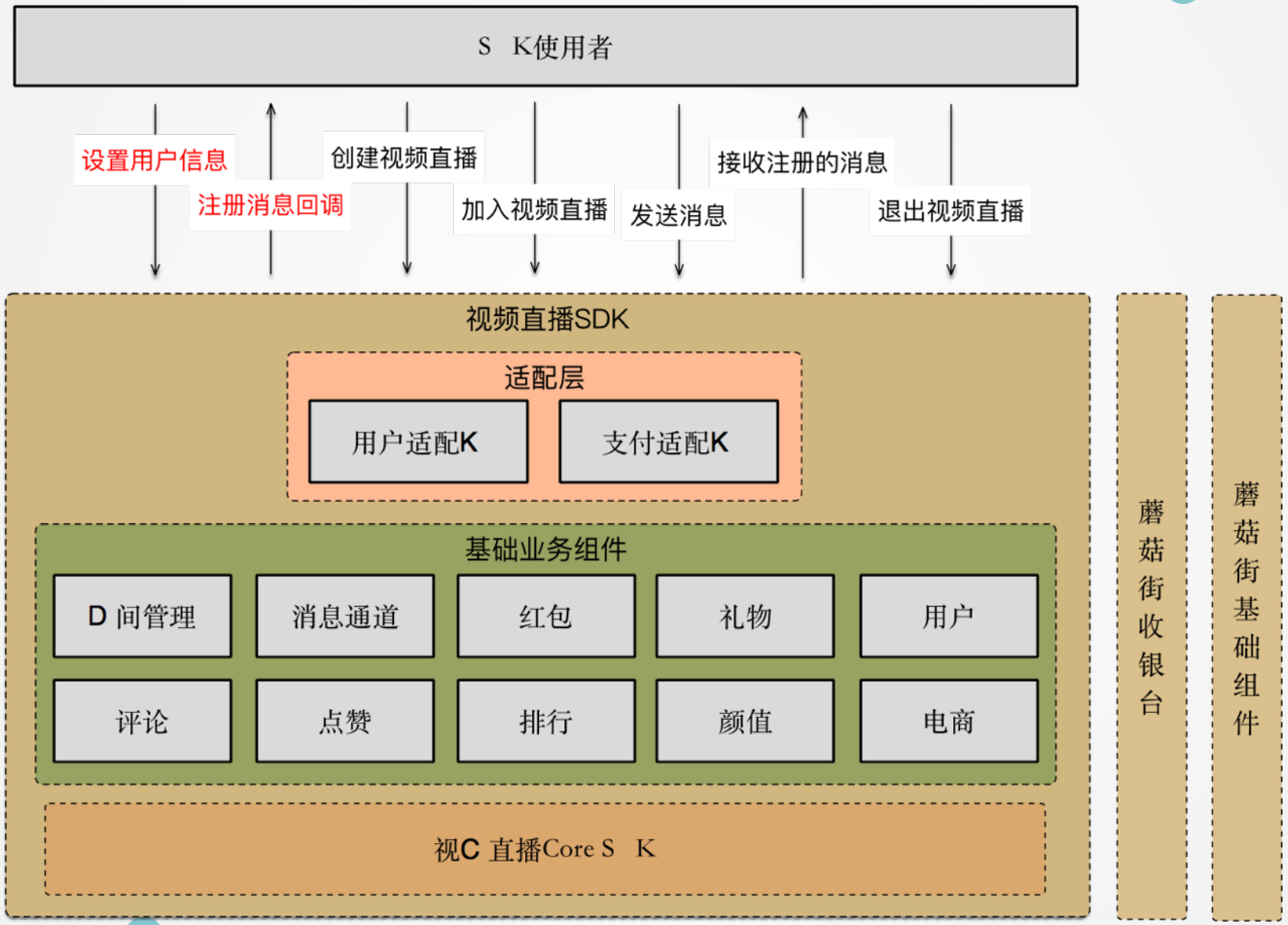
直播SDK化


用户系统
蘑菇街、美丽说、淘世界、MOGU
电商系统
蘑菇街、美丽说、淘世界、MOGU
视频直播SDK
适配层
用户适配层
支付适配层
基础业务组件
房间管理
消息通道
红包
礼物
用户
评论
点赞
排行
颜值
电商
视频直播Core SDK
直播音视频模块
Tencent QAL
蘑菇街自研人脸识别
Tencent QAV
腾讯云登录SDK
直播IM模块
消息分发模块
Tencent IM
蘑菇街收银台
蘑菇街基础组件

直播SDK化 - Core


S K使用者
创建视频直播
加入视频直播
注册IM消息回调
发送IM消息
接收注册的IM消息
退出视频直播
视频直播Core SDK
直播音视频模块
Tencent QAL
蘑菇A 自D
人脸识别
Tencent QAV
腾讯云登录
S K
直播IM模块
消息分发模
块
Tencent IM

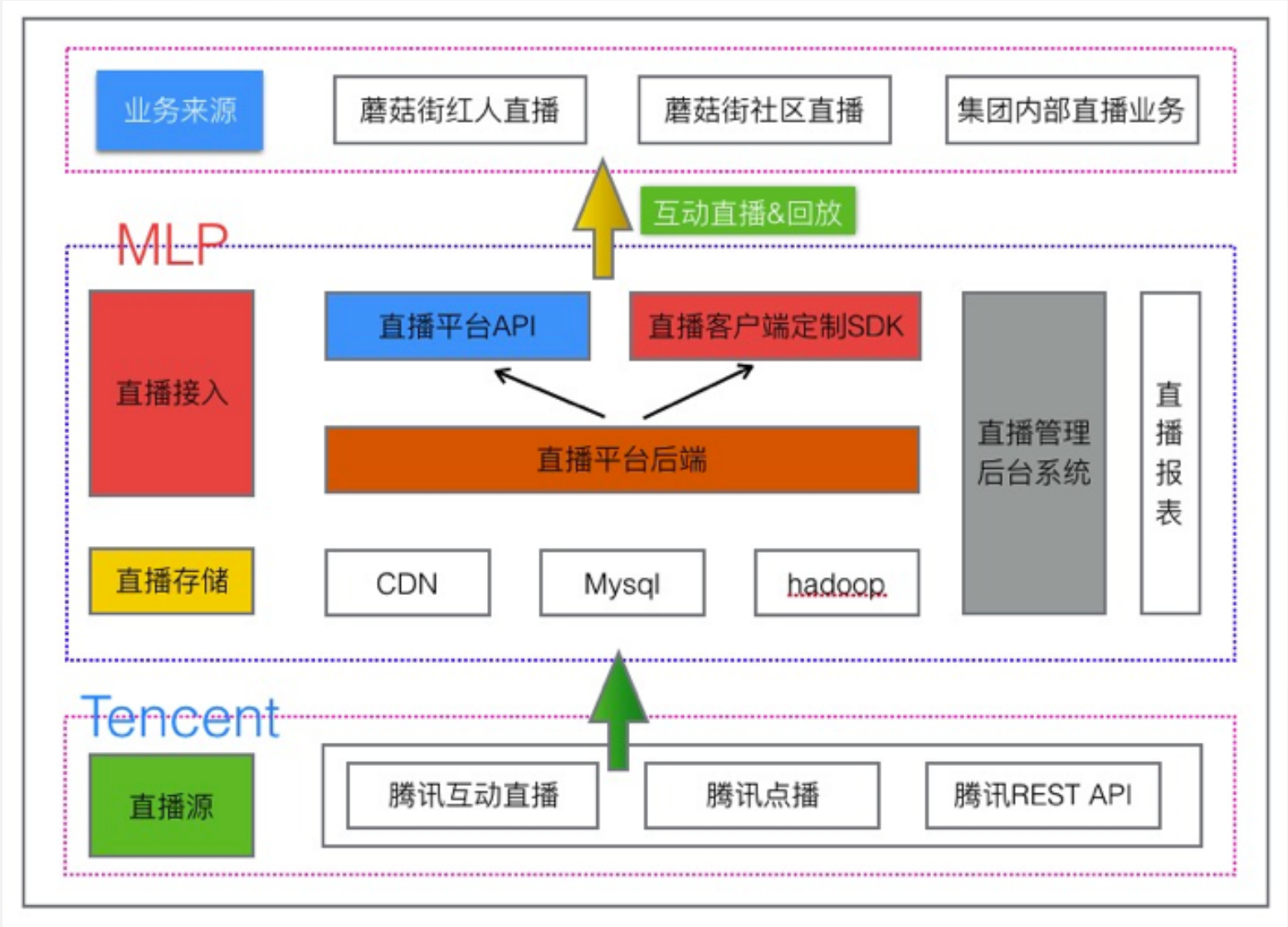
直播SDK化 – 业务


S K使用者
设置用户信息
注册消息回调
创建视频直播
加入视频直播
发送消息
接收注册的消息
退出视频直播
视频直播SDK
适配层
用户适配K
支付适配K
基础业务组件
D 间管理
消息通道
红包
礼物
用户
评论
点赞
排行
颜值
电商
视C 直播Core S K
蘑菇街收银台
蘑菇街基础组件

平台化

- 更好的业务方接入方式
- 更加便捷精准的数据平台

直播平台化 – 总体结构


业务来源
蘑菇街红人直播
蘑菇街社区直播
集团内部直播业务
互动直播&回放
MLP
直播接入
直播平台API
直播客户端定制SDK
直播平台后端
直播管理后台系统
直播报表
直播存储
CDN
Mysql
hadoop
Tencent
直播源
腾讯互动直播
腾讯点播
腾讯REST API

感谢直播团队的小伙伴们！

谢谢大家！

蘑菇街-花荣

Q & A