

微信H5视频播放器在海量业务下的实践

微信开放平台基础部

郭志伟 (raphealguo)

2017-04-16



业务背景

微信用户8亿+

* 来自腾讯2016年第三季度业绩报告

图文消息30亿+PV

* 2015年公众号图文消息日均PV

前端遇到什么问题

小bug容易引起投诉

访问量大，富资源的带宽成本巨高

快速的产品迭代，导致代码难以维护

某些特定环境下发生的用户侧问题难以排查

关于这次演讲

1.技术方案可控

通过灵活的技术方案使
得系统可控

2.监控和发现问题

能够通过各种手段发现
排查现网问题

Part1 通过合适的技术方案让系统可控

来自“微信H5播放器”的实践

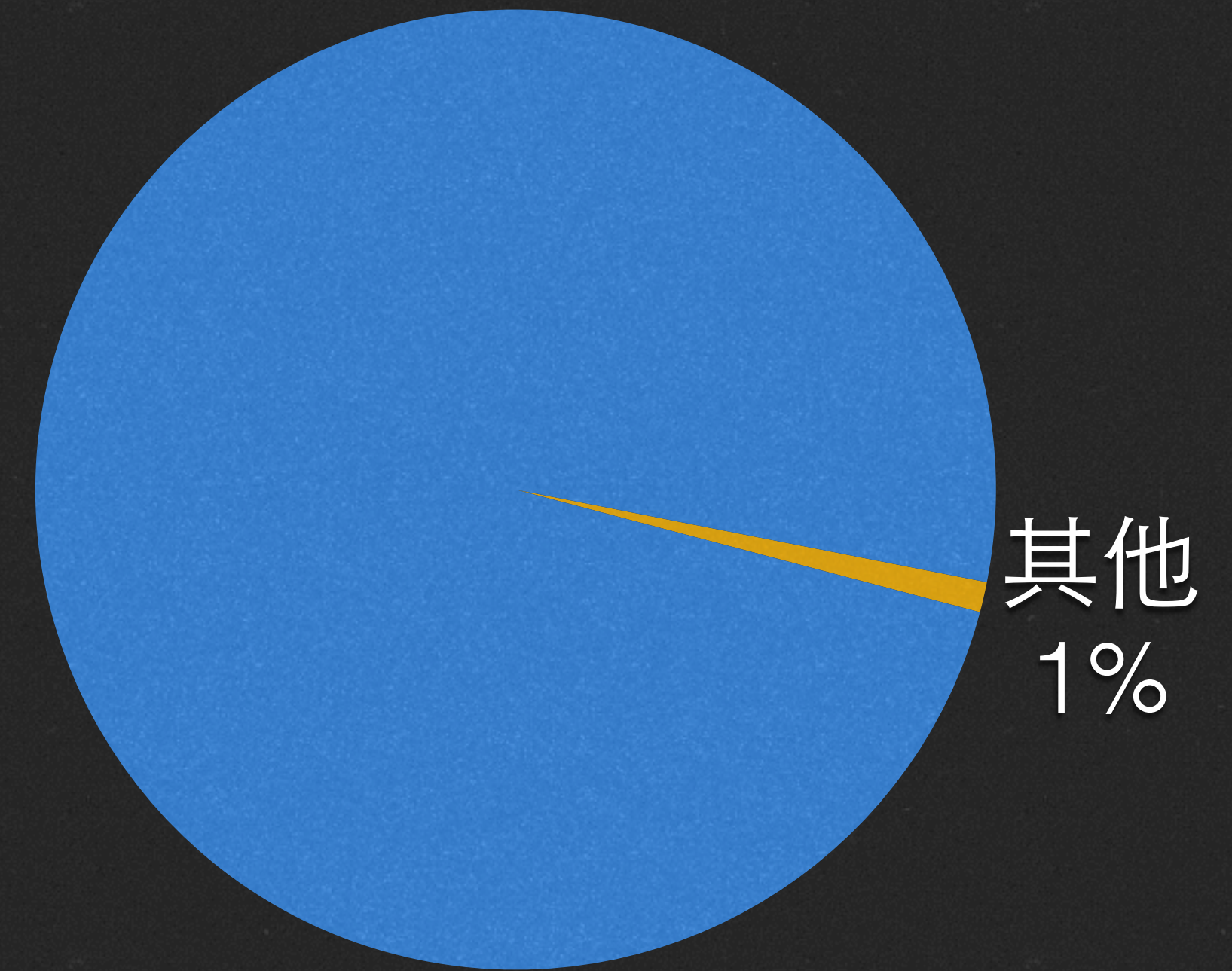
微信H5播放器



公众号文章访问情况

1. IOS & 安卓访问比例达99%
2. 安卓自带X5内核Webview

IOS &
99%



结论：只需开发HTML5版本Video播放器

朴素版播放器状态机

①

腾讯视频接口(拉取封面/视频url/时长等信息)

朴素版播放器状态机



①

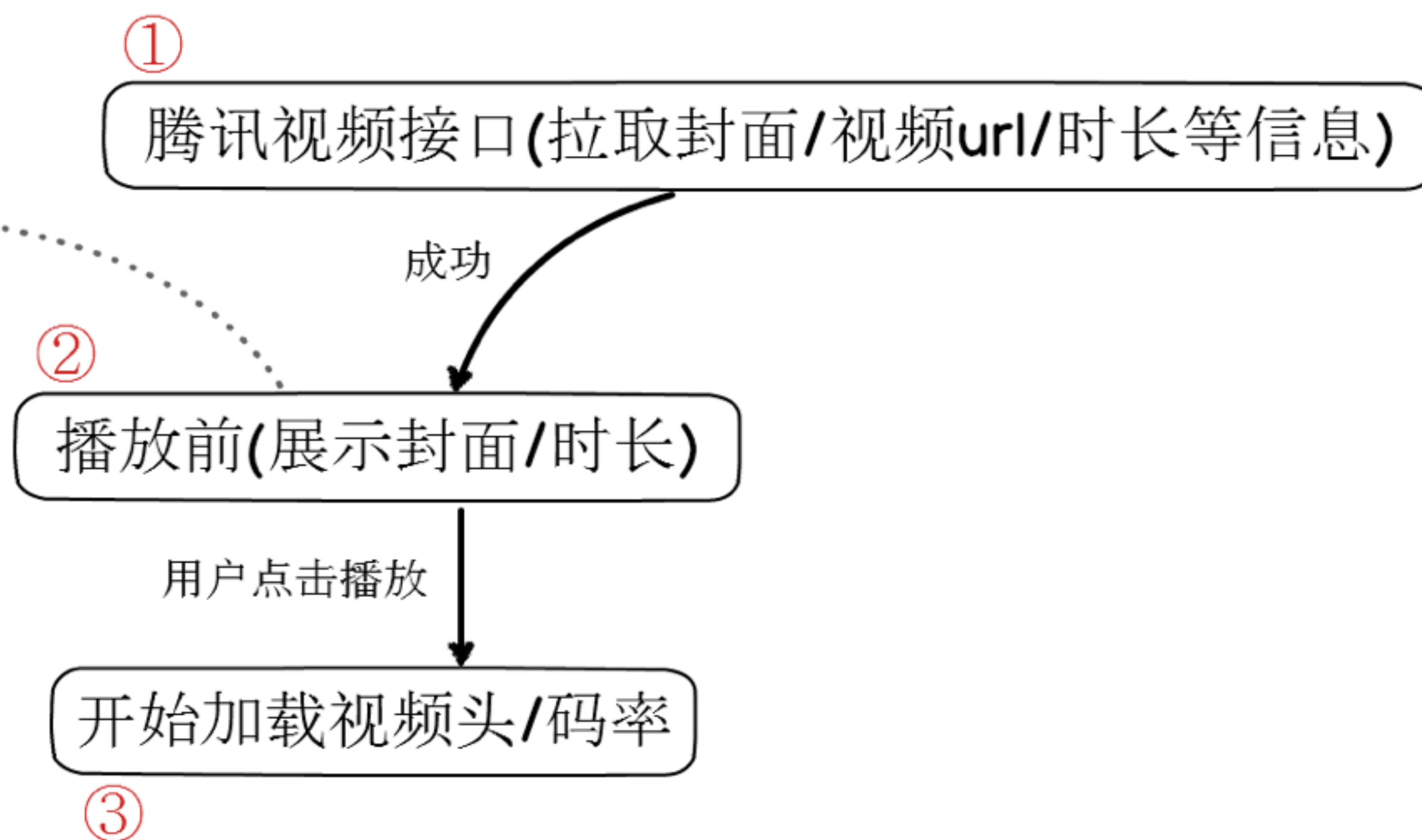
腾讯视频接口(拉取封面/视频url/时长等信息)

成功

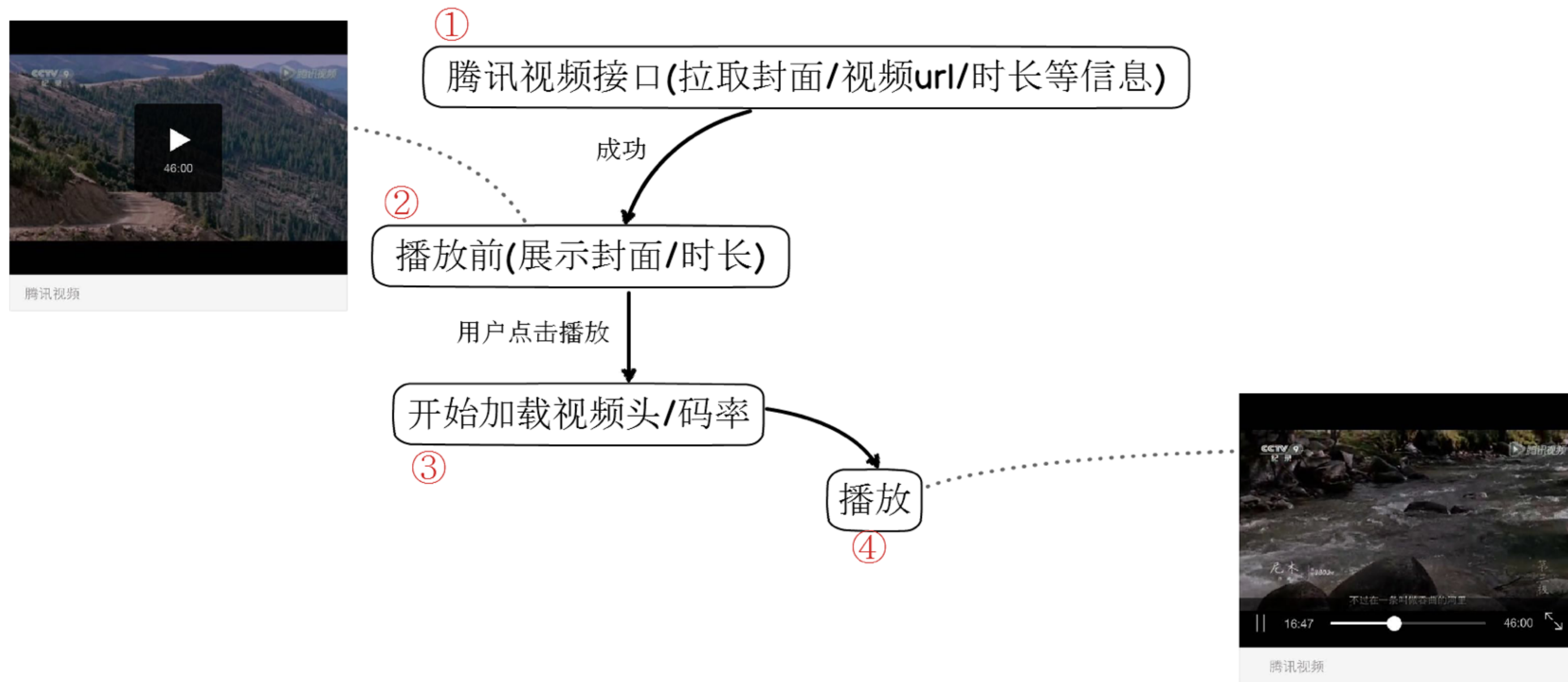
②

播放前(展示封面/时长)

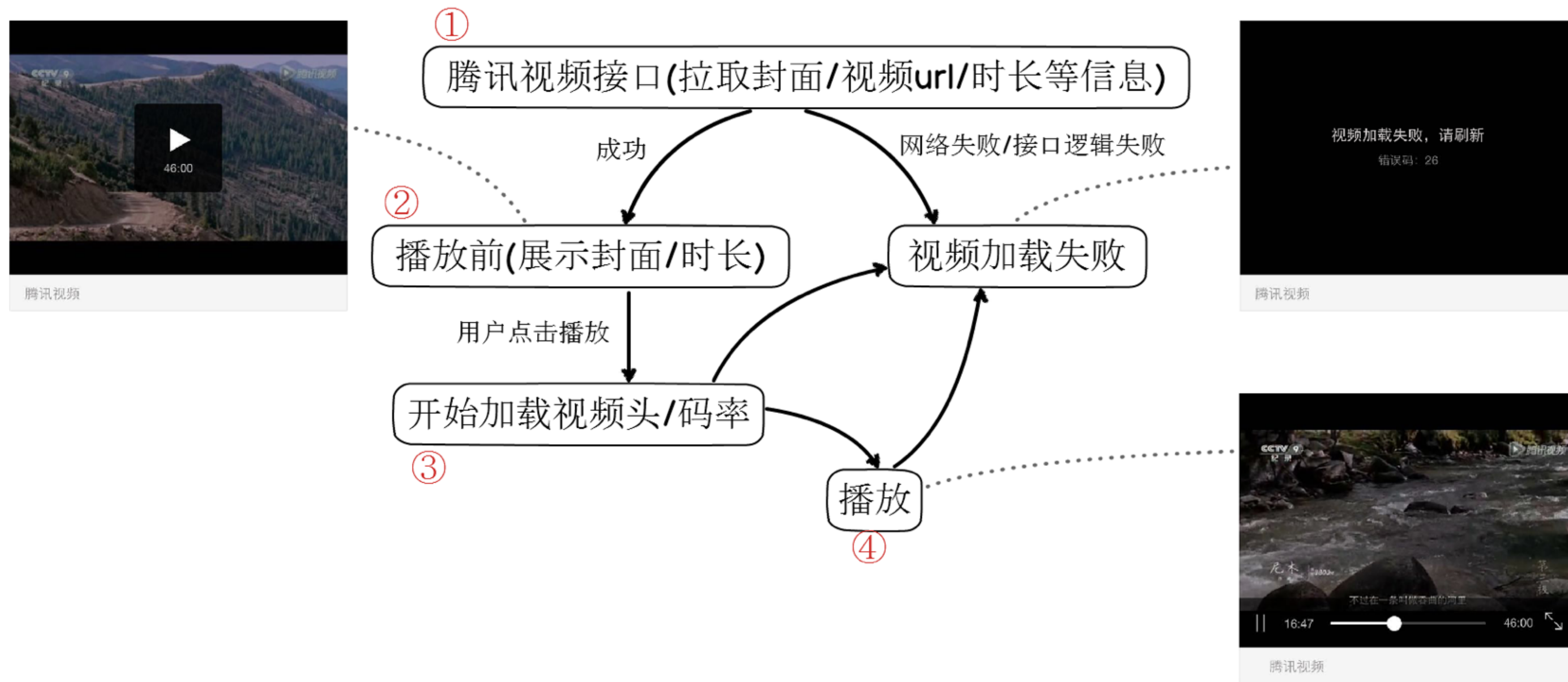
朴素版播放器状态机



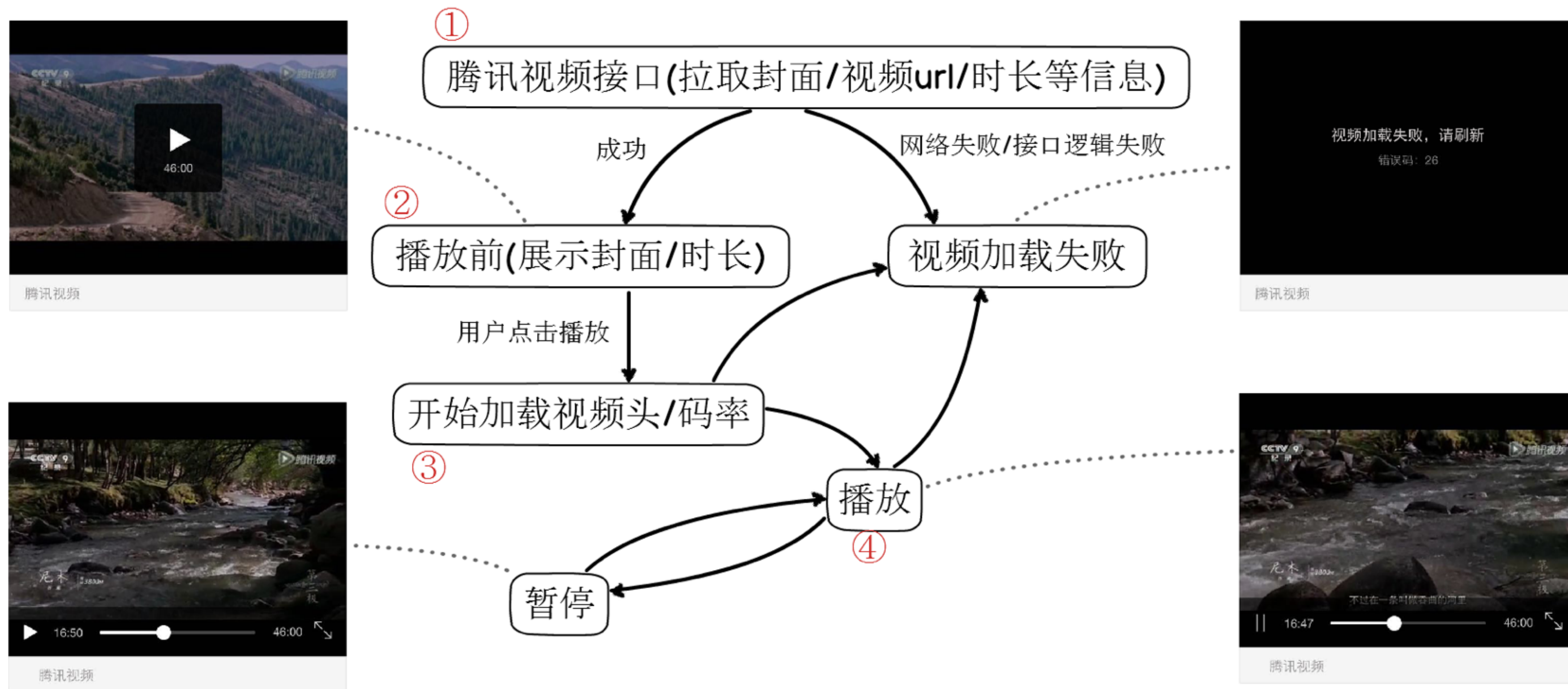
朴素版播放器状态机



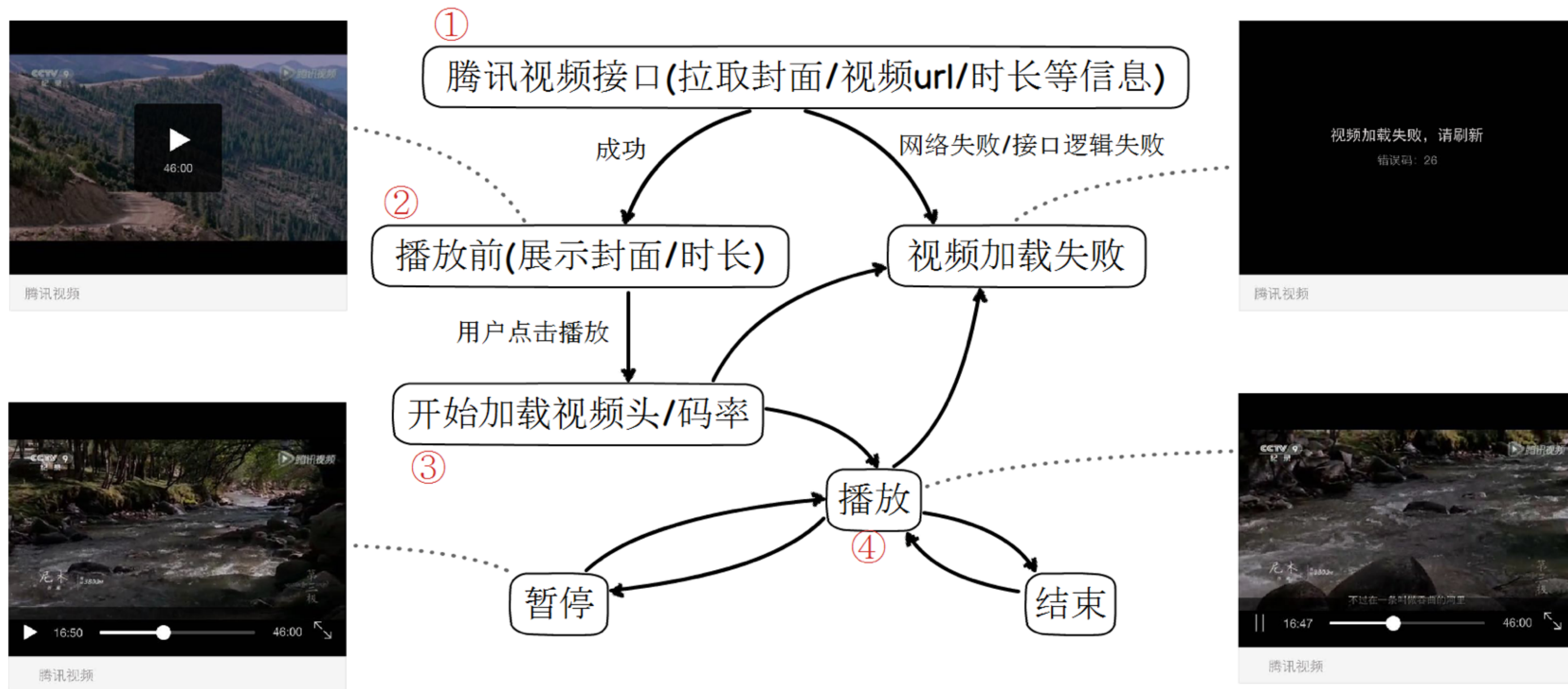
朴素版播放器状态机



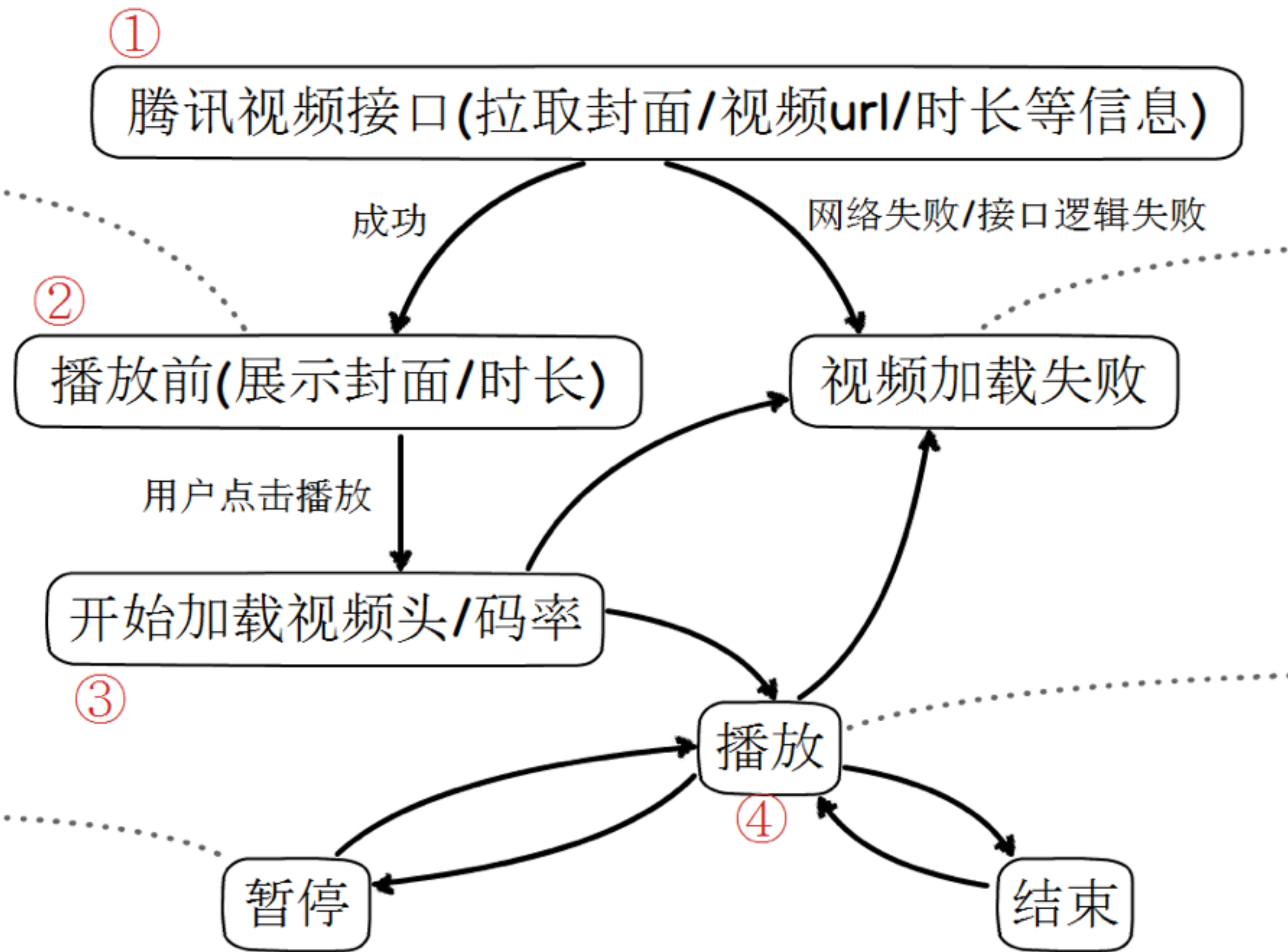
朴素版播放器状态机



朴素版播放器状态机



朴素版播放器状态机



容易实现：解决兼容性+解决UI表现

还要实现播放器的扩展功能

播放器需求——前贴片广告

美拍网友恶搞版花千骨，无节操三连发，笑死宝宝了
恶搞第一集

2 秒后可关闭广告



腾讯视频

恶搞第二集

婚姻不同时期的爱情保鲜剂。

纸婚篇

—— Paper wedding ——



腾讯视频

纸婚：结婚一周年，形容夫妻关系最初薄如一纸婚约，等待所有考验。

mp.weixin.qq.com | @微信公众平台

并且，再一次觉得自己是文盲。



他的风格并非明朝那些事儿一样，
到处闪耀着意淫的光辉，

(匿凉北叶画子回样(步垂转米年昭日)

到处闪耀着意淫的光辉，
他的风格并非明朝那些事儿一样，

播放器需求——前贴片广告

美拍网友恶搞版花千骨，无节操三连发，笑死宝宝了
恶搞第一集

2 秒后可关闭广告



腾讯视频

恶搞第二集

婚姻不同时期的爱情保鲜剂。

纸婚篇

—— Paper wedding ——



腾讯视频

纸婚：结婚一周年，形容夫妻关系最初薄如一纸婚约，等待所有考验。

mp.weixin.qq.com | @微信公众平台

并且，再一次觉得自己是文盲。



他的风格并非明朝那些事儿一样，
到处闪耀着意淫的光辉，

(匿凉北叶画子回样(步垂转米年昭日)

到处闪耀着意淫的光辉，
他的风格并非明朝那些事儿一样，

播放器需求—优化带宽成本

使用HTML5的video能力播放视频

```
<video src="http://cdn.qq.com/123.mp4">
```

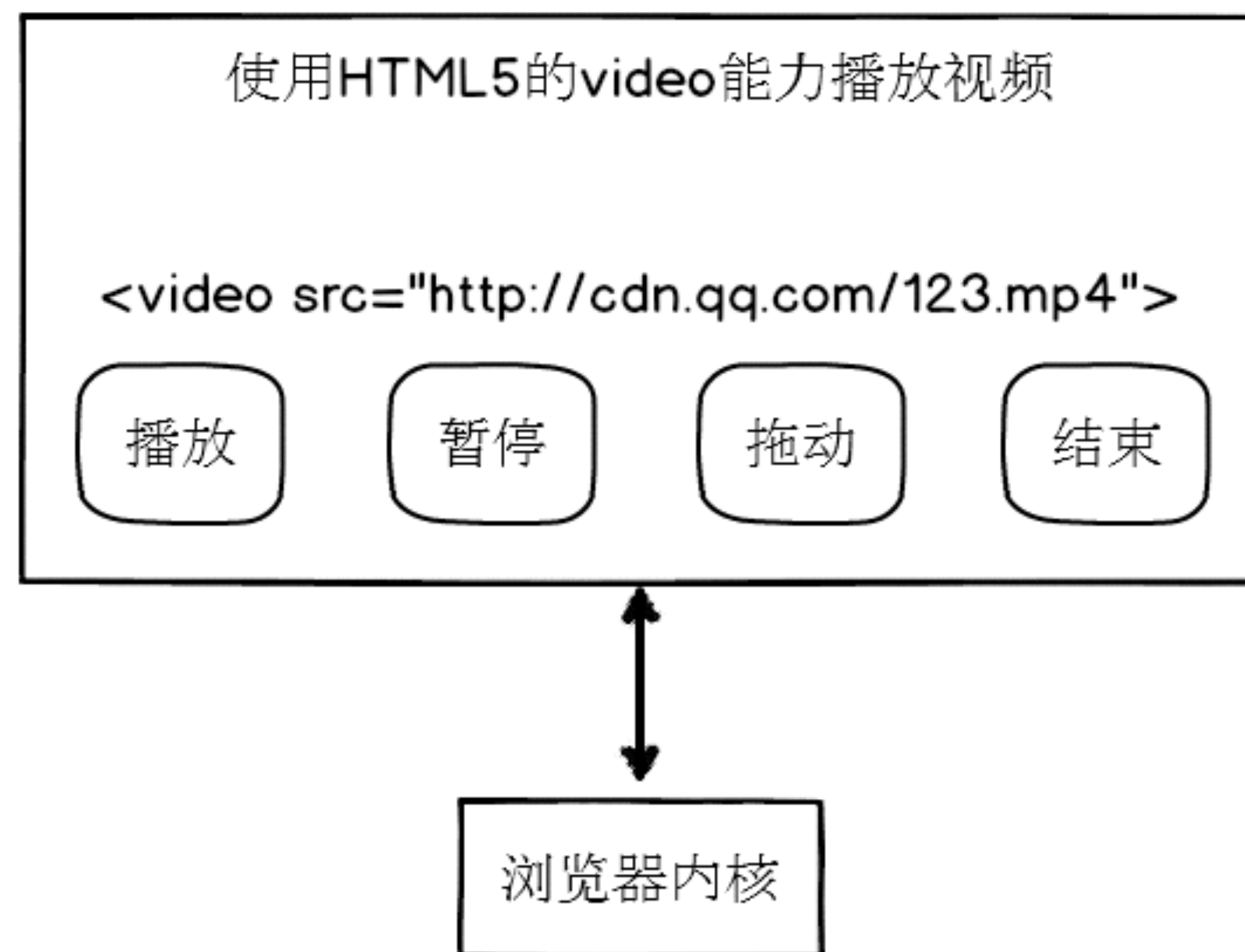
播放

暂停

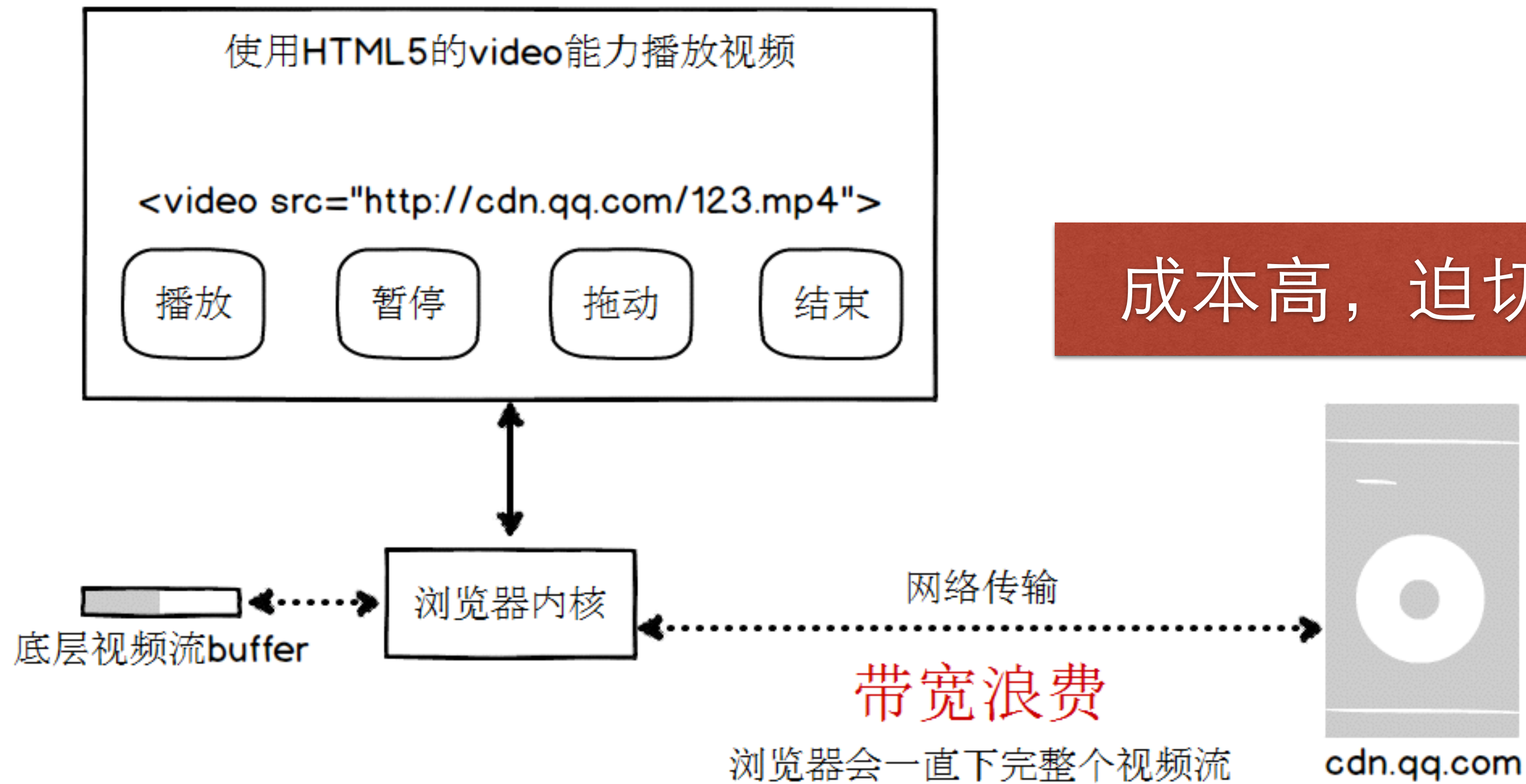
拖动

结束

播放器需求—优化带宽成本

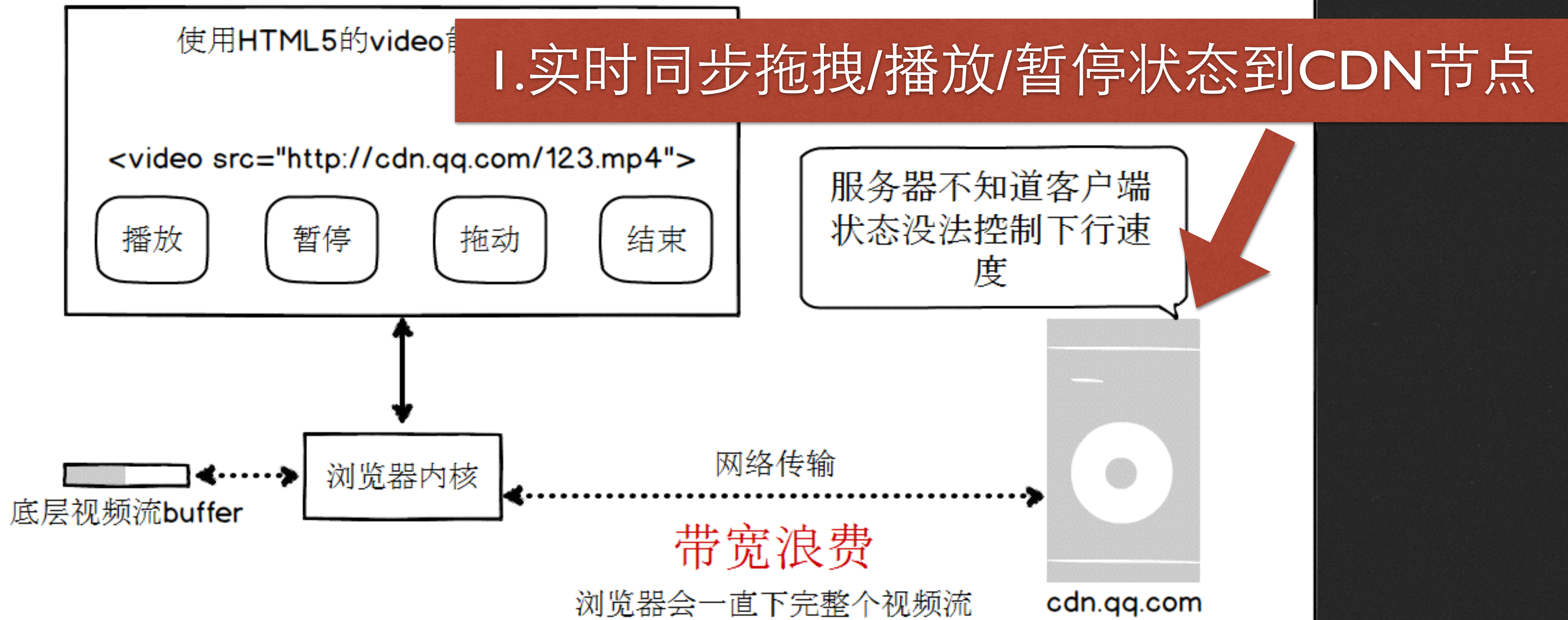


播放器需求—优化带宽成本



成本高，迫切需要优化带宽

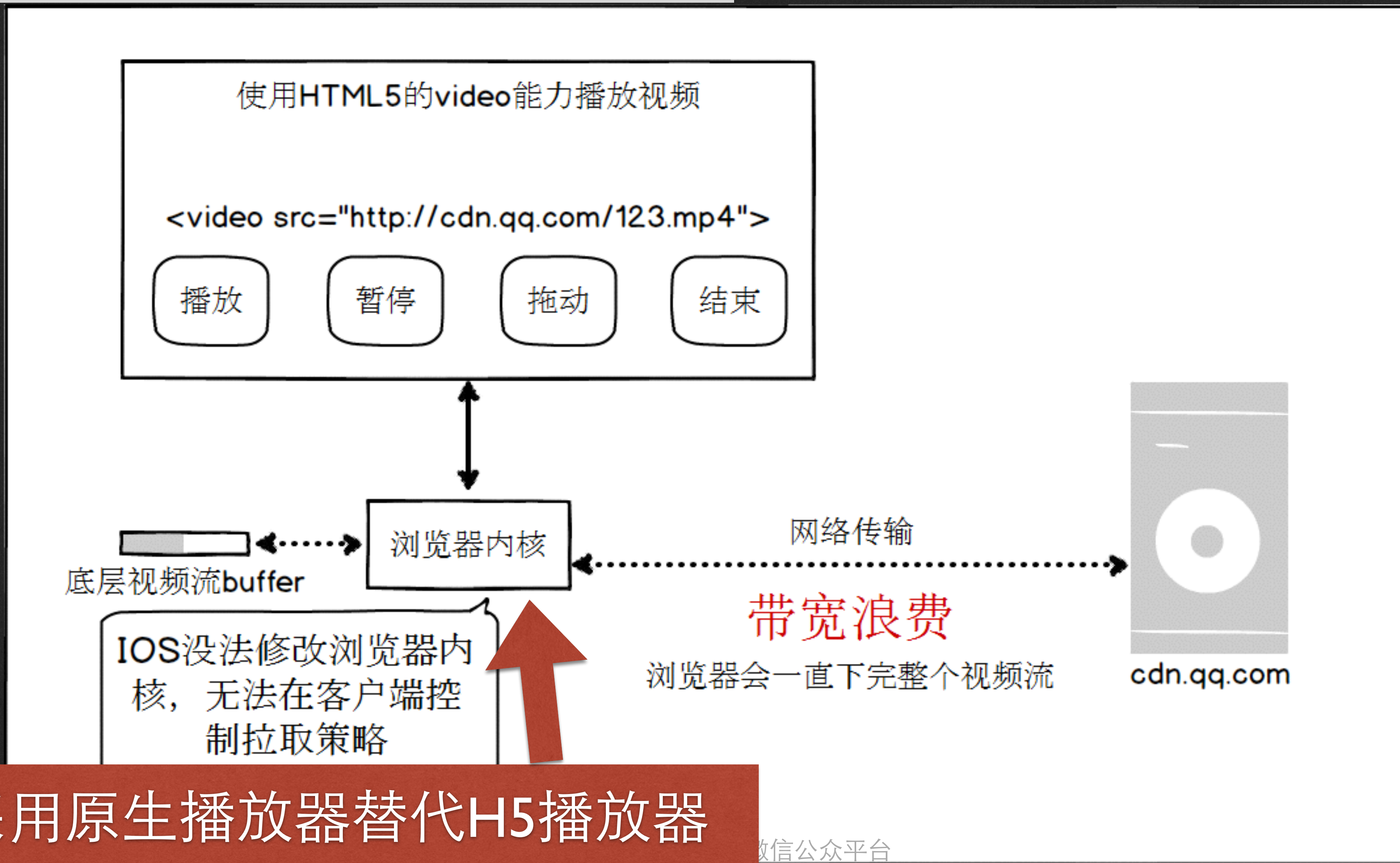
节省带宽的方案1



方案1的缺点

- CDN服务复杂化，不再是纯粹分发资源
- 客户端服务器同步操作状态有延时

节省带宽的方案2

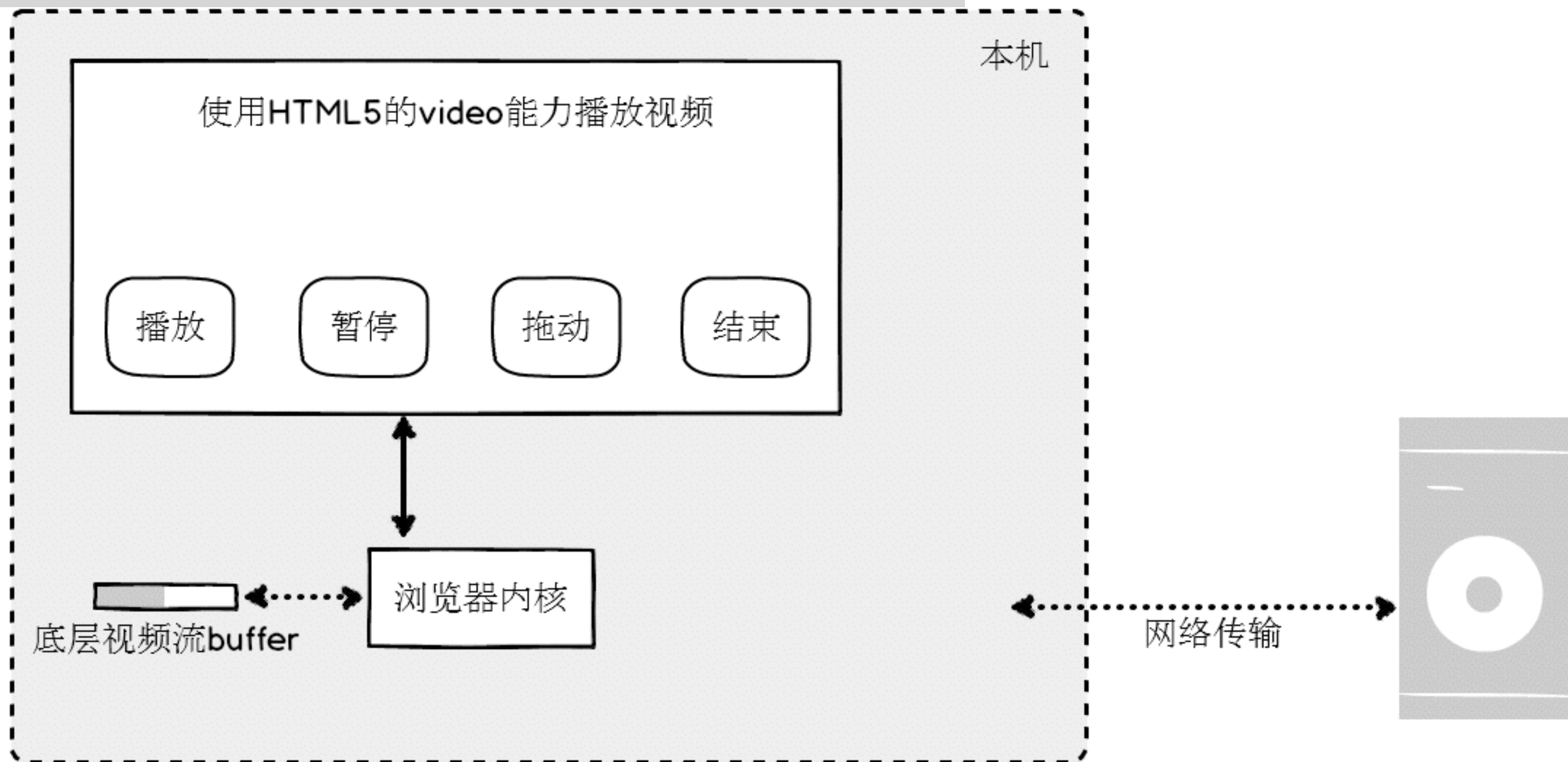


方案2的缺点

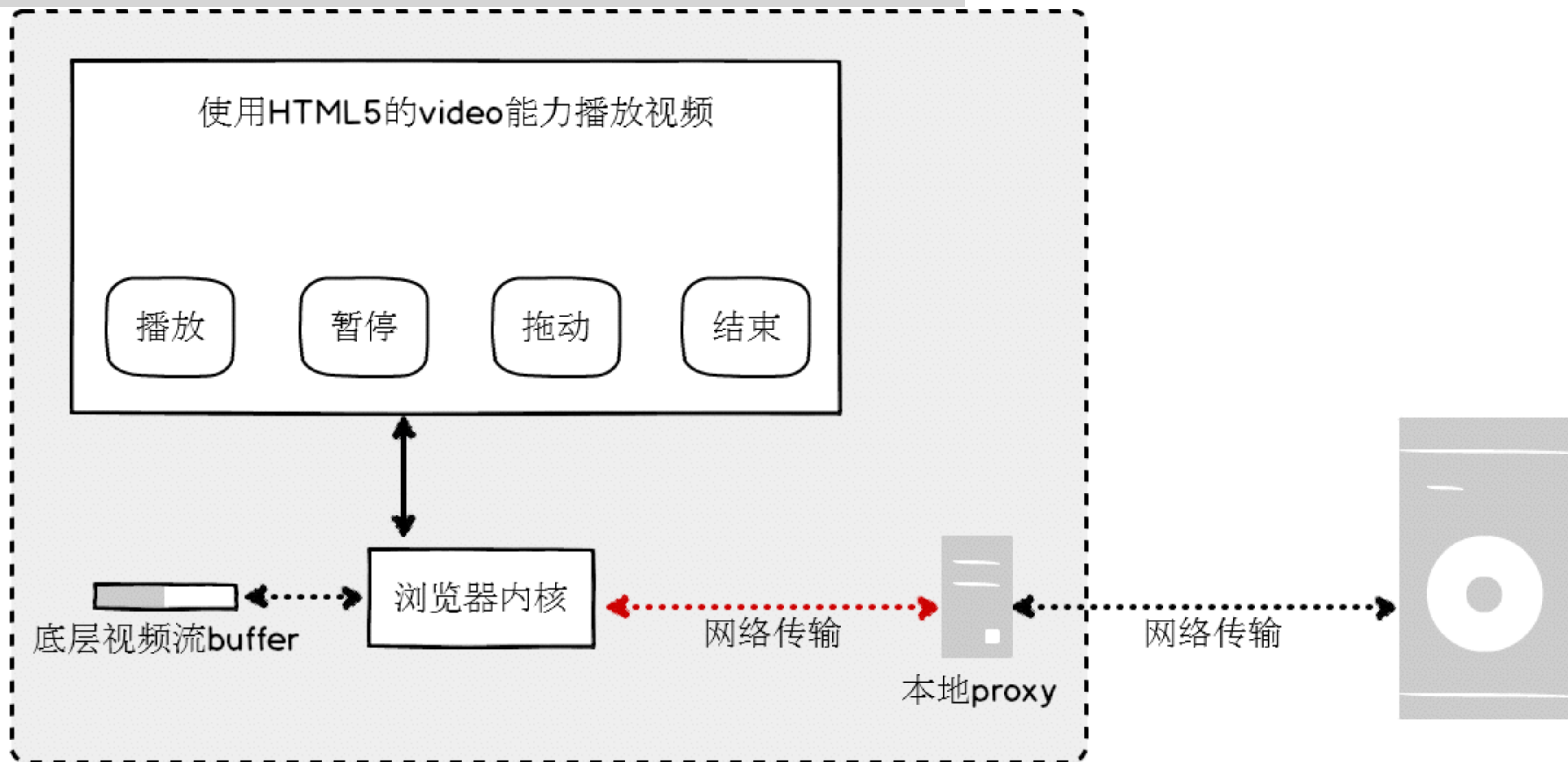
- 原生组件覆盖在Webview之上，需要解决滚动问题
- 迭代速度受到客户端发版影响

思路：把远端的Server搬到本地——本地代理

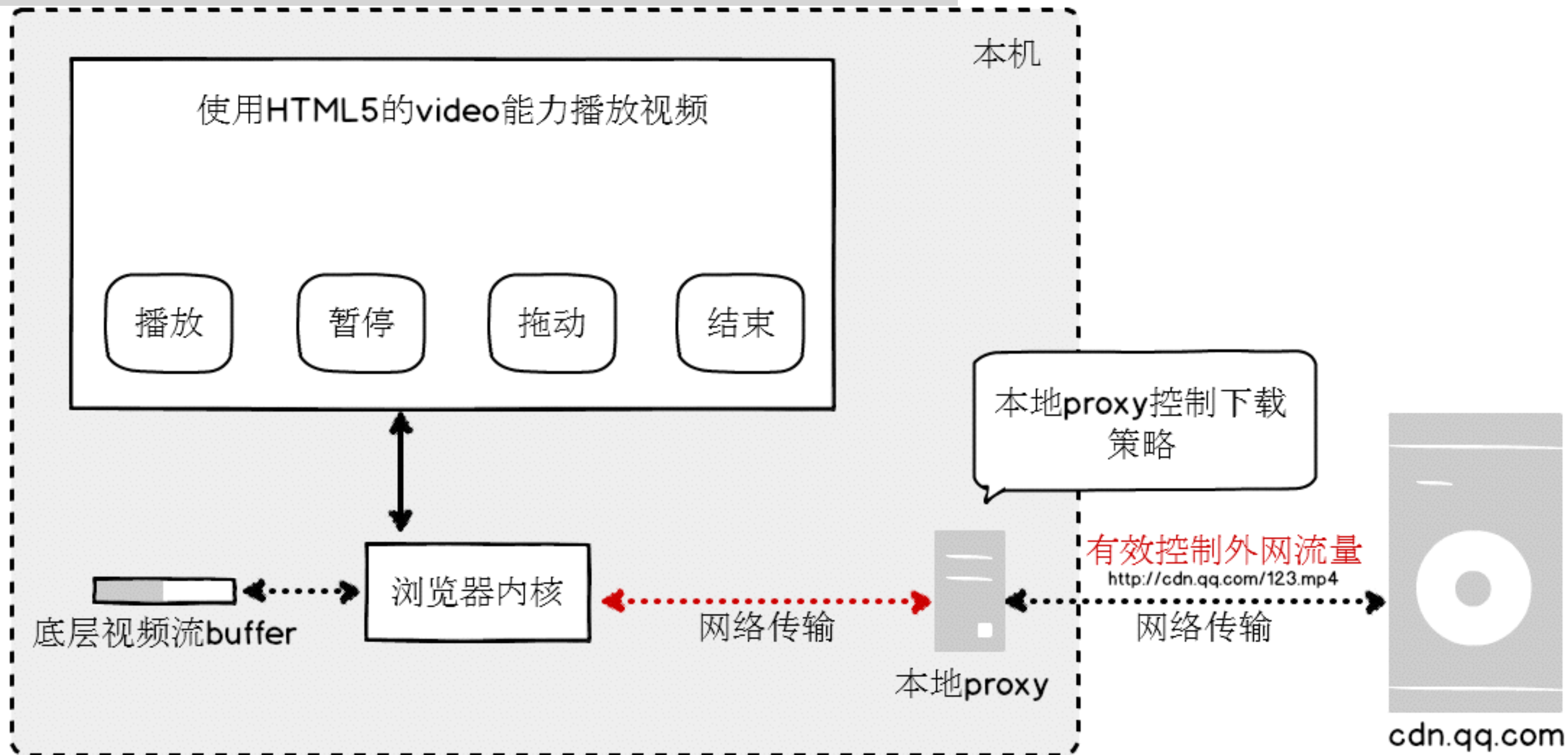
最终方案：本地代理proxy



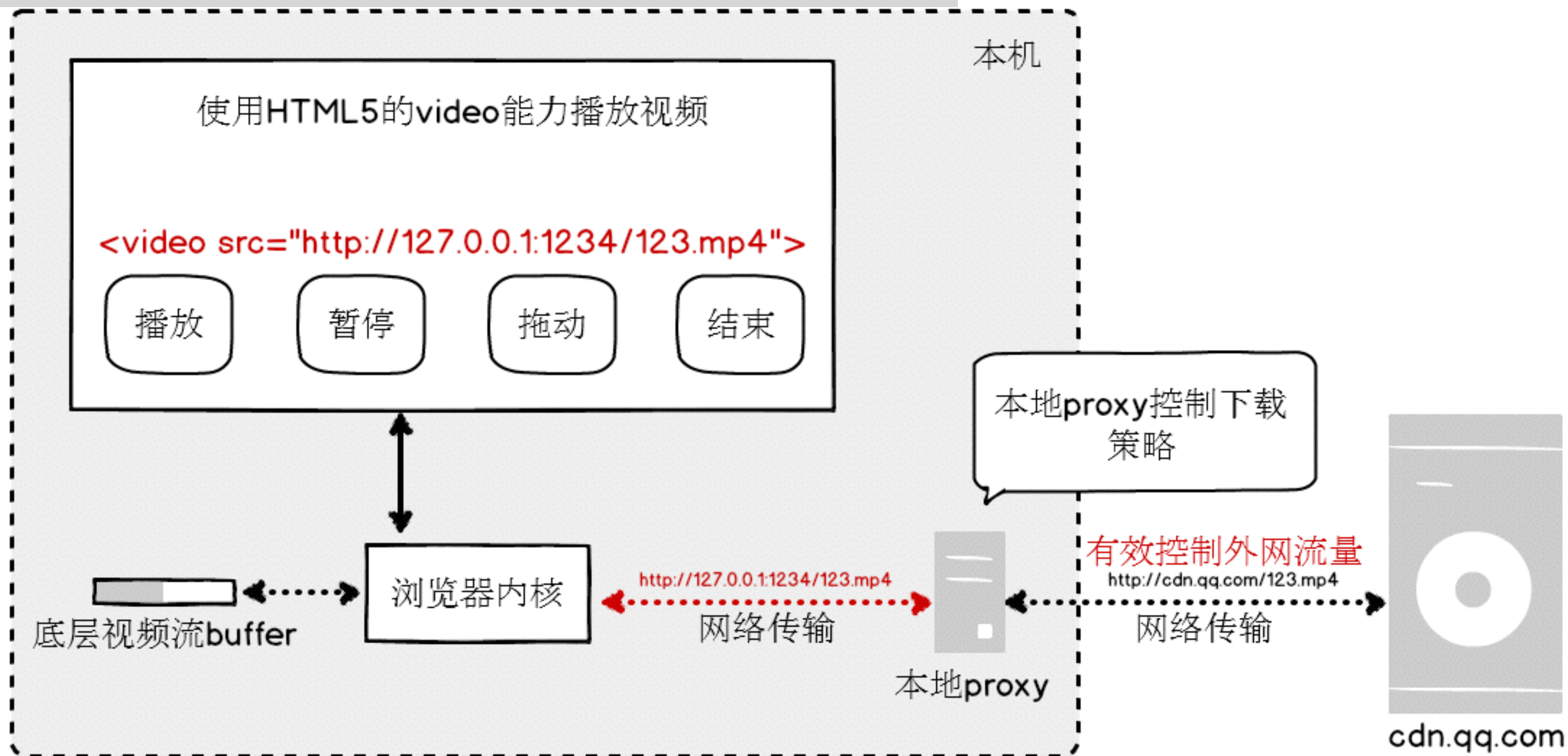
最终方案：本地代理proxy



最终方案：本地代理proxy



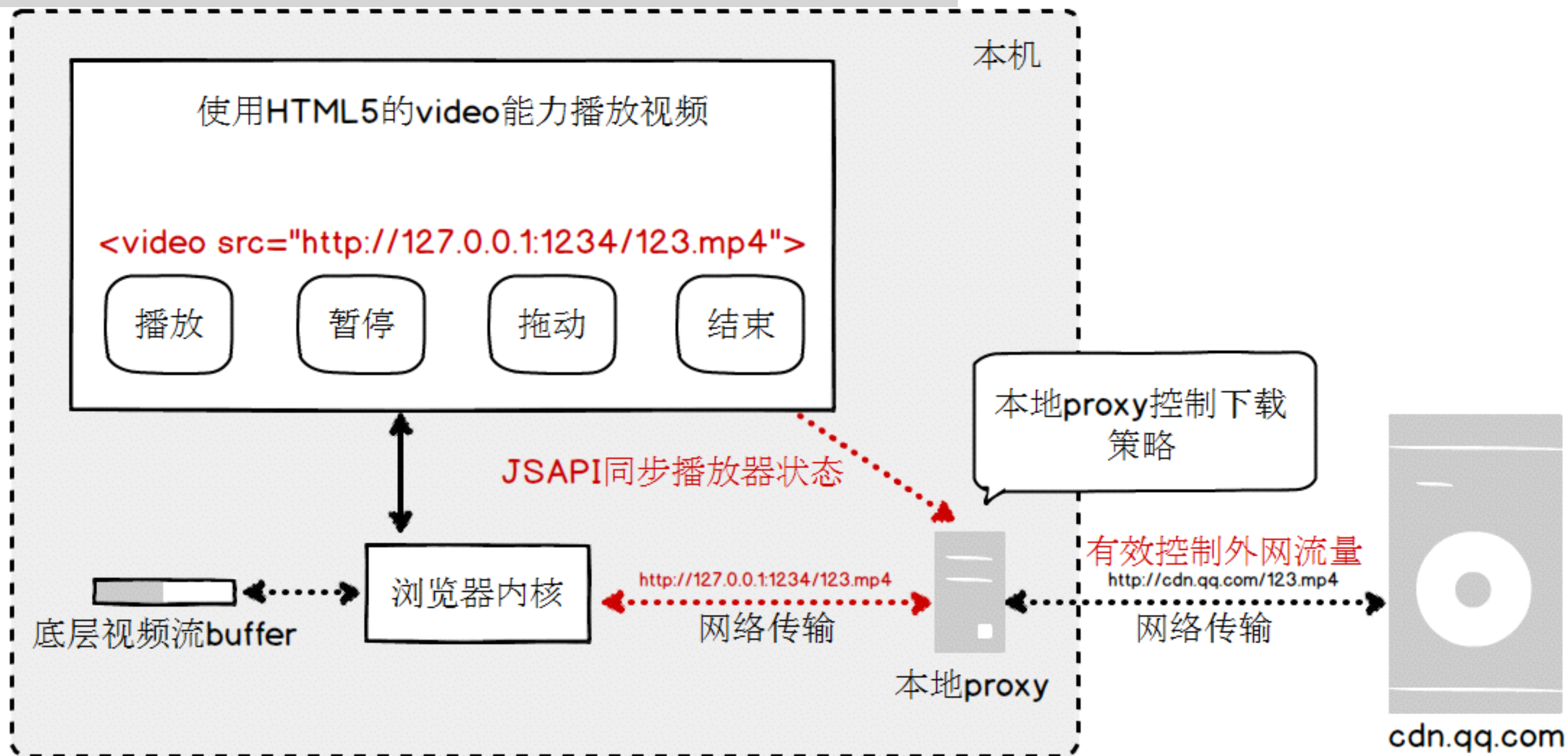
最终方案：本地代理proxy



proxy两个关键点：

1. 需要把一个外网地址换成本地地址

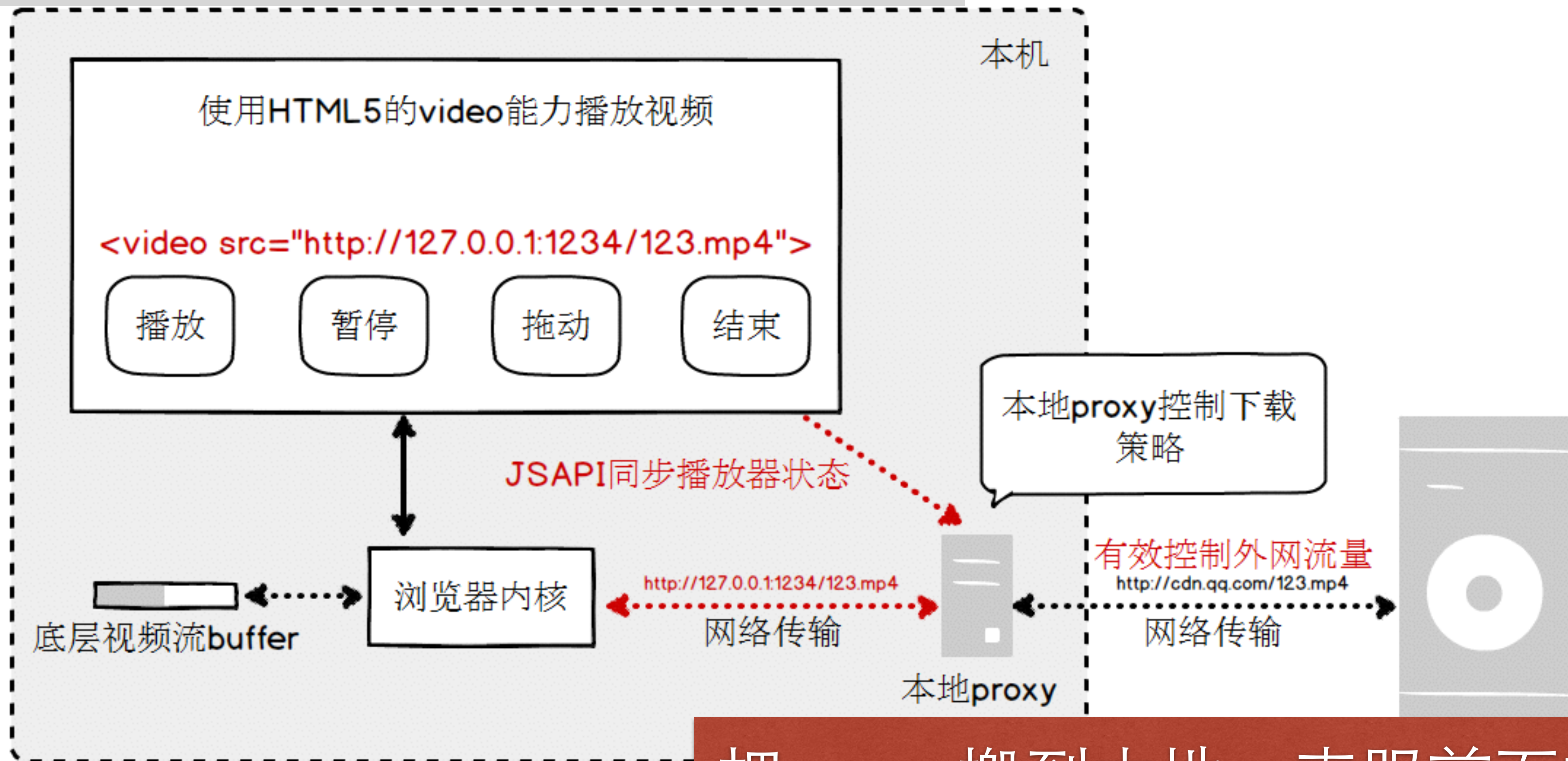
最终方案：本地代理proxy



proxy两个关键点：

1. 需要把一个外网地址换成本地地址
2. 需要同步播放器状态给proxy

最终方案：本地代理proxy



proxy两个关键点：

把server搬到本地：克服前面方案的缺点

1. 需要把一个外网地址换成本地地址

2. 需要同步播放器状态给proxy

播放器的状态机

腾讯视频接口返回

网络失败/接口逻辑失败

原先播放器的状态

视频加载失败

解码失败/网络失败

开始加载视频头/码率

成功解析

解码失败/网络失败

播放

缓冲加载完毕/点击播放

视频播放完毕

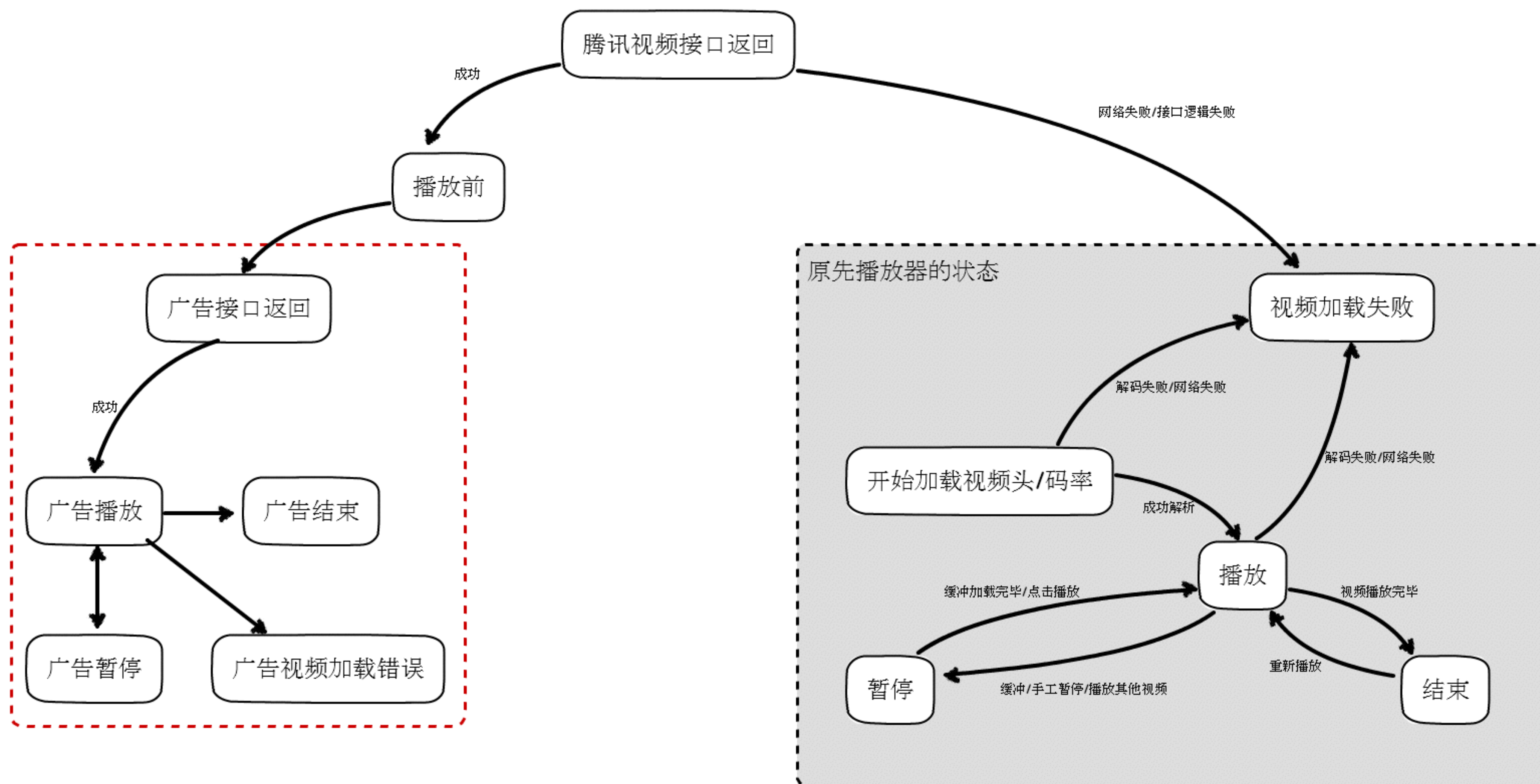
暂停

缓冲/手工暂停/播放其他视频

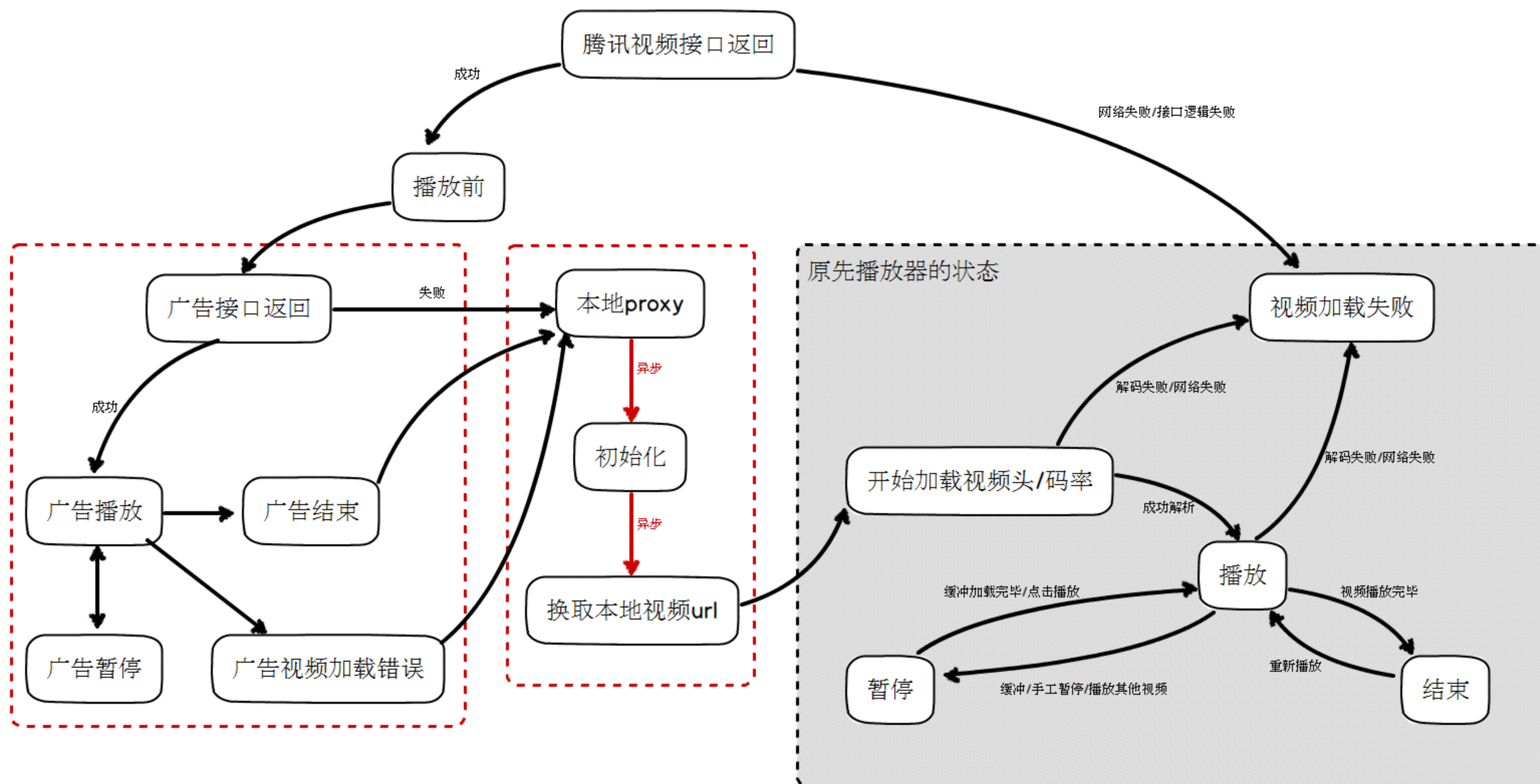
重新播放

结束

播放器的状态机

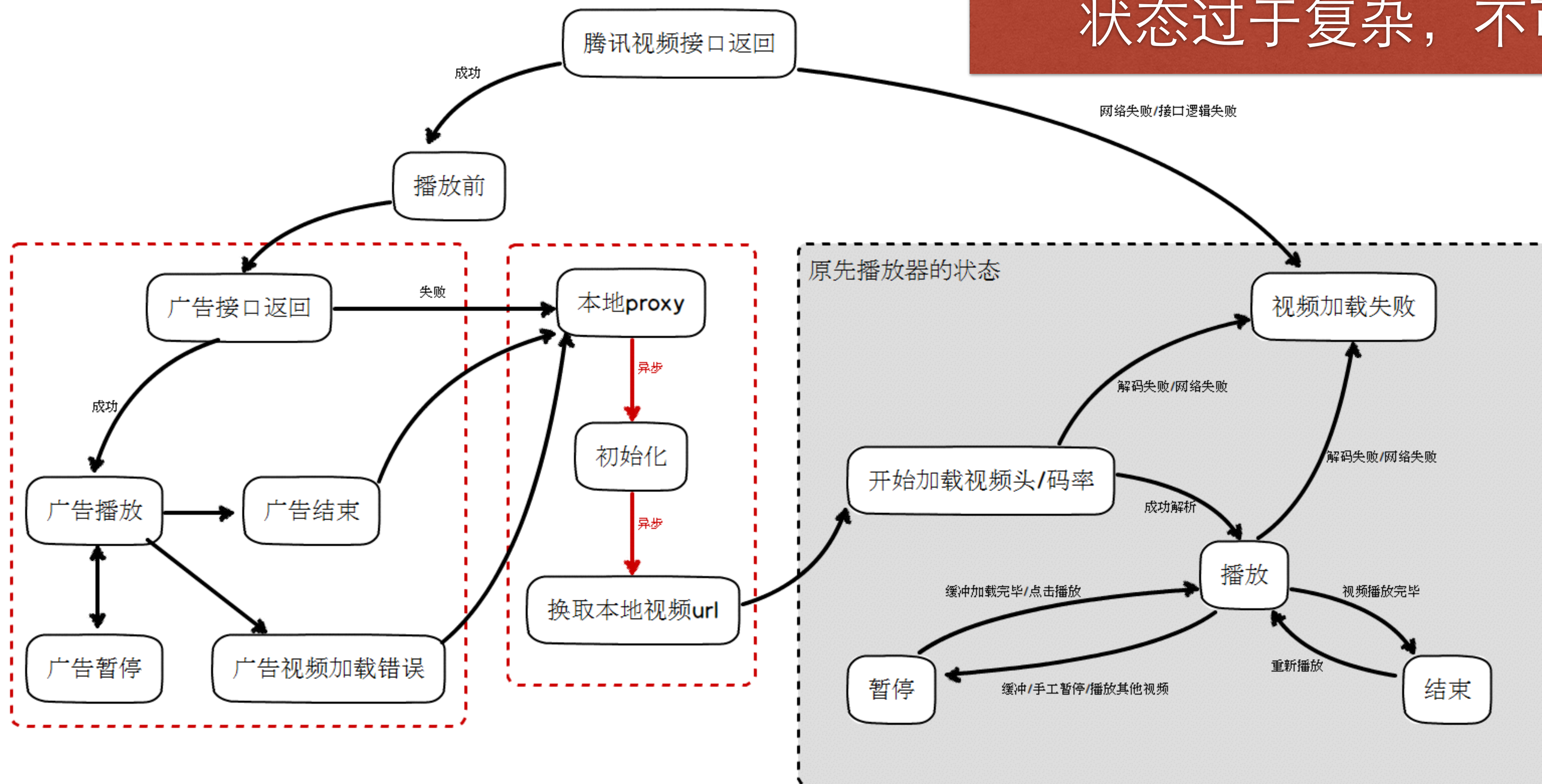


播放器的状态机



播放器的状态机

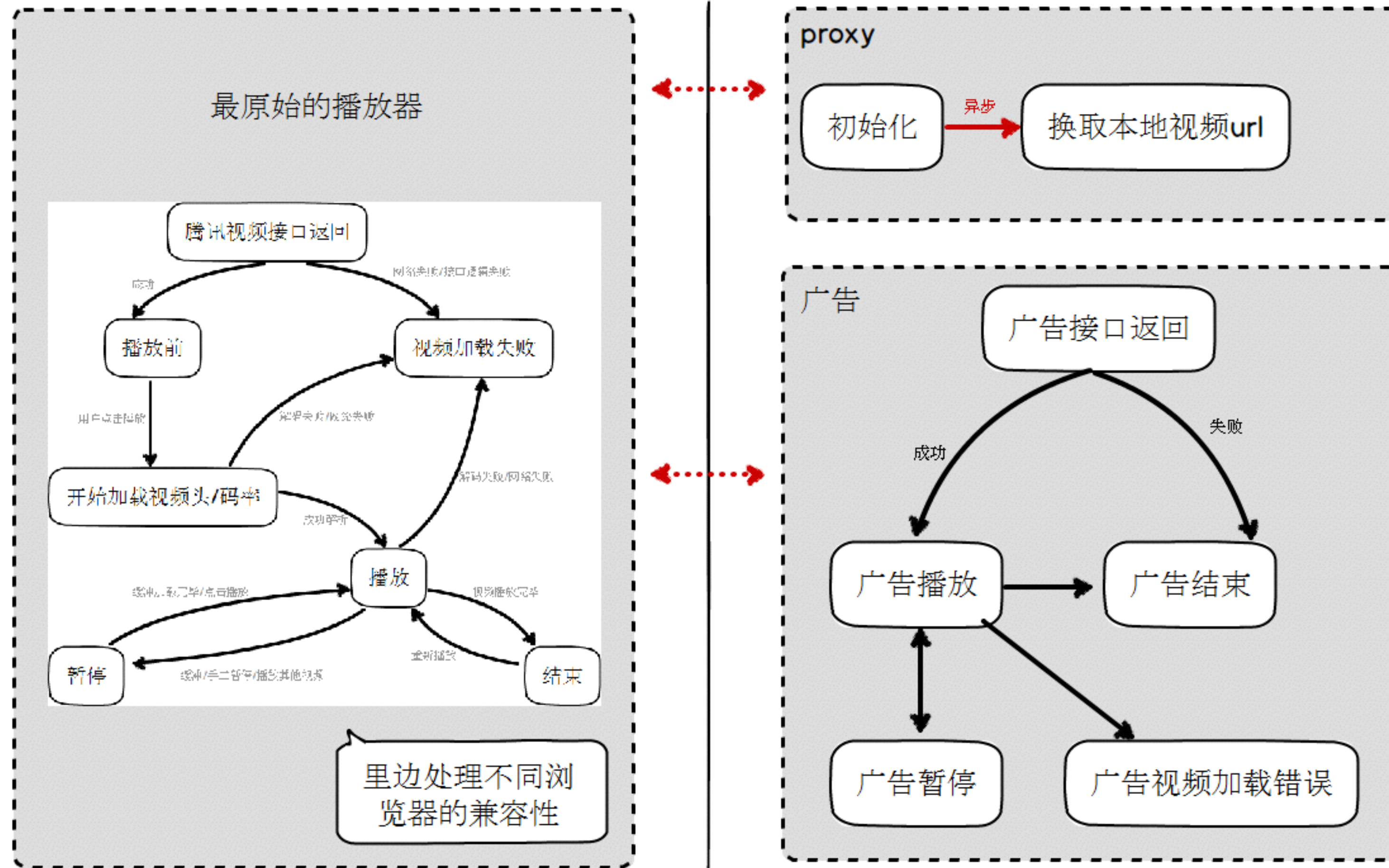
状态过于复杂，不可行



朴素版的缺点：系统不可控

- 状态极多，状态之间的跳转控制复杂
- 扩展能力灰度控制很麻烦
- 编写扩展能力的时候，需要侵入原播放器的流程
- 扩展间耦合过重，改动容易会引发bug

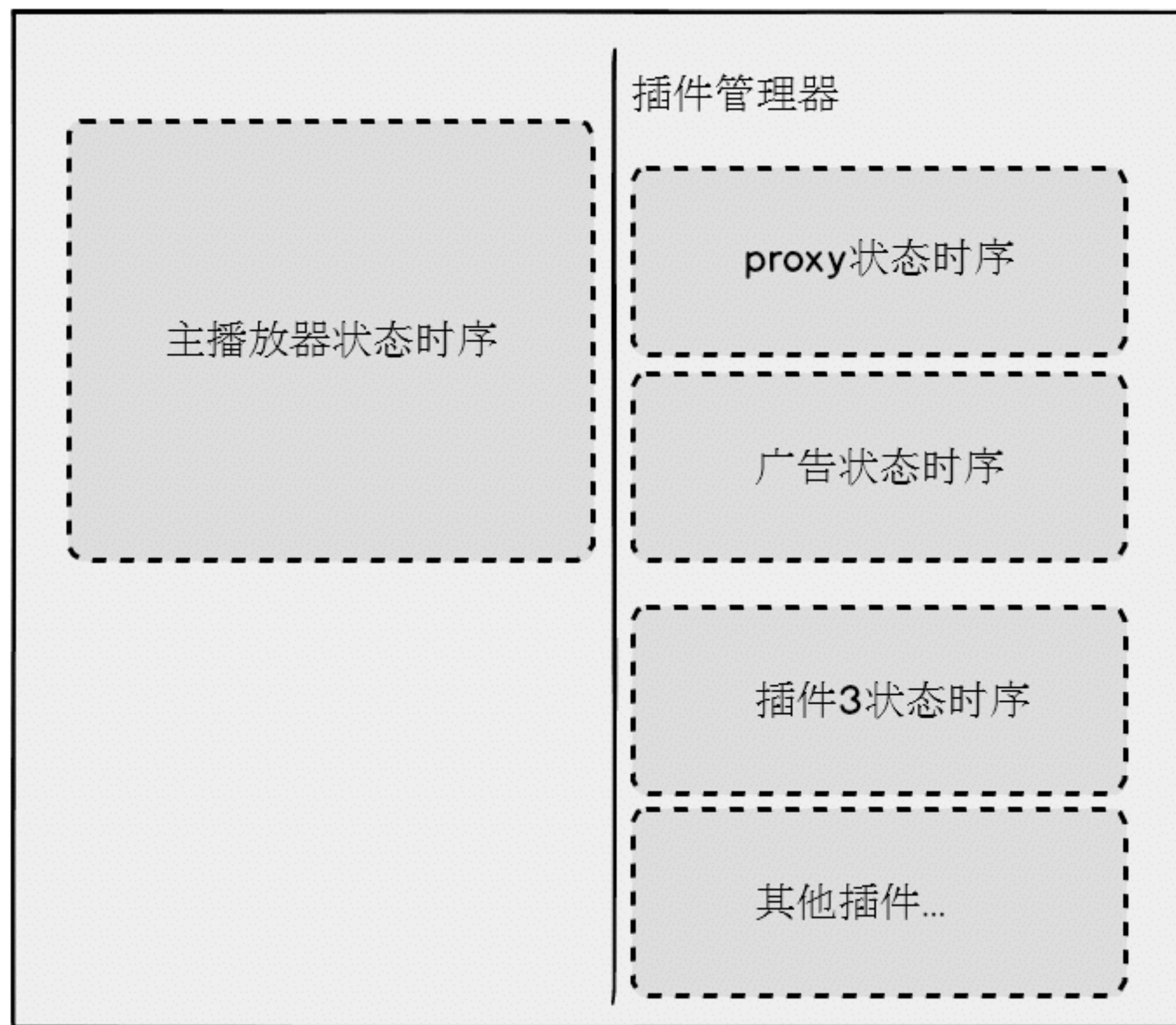
我们的目标



- 主播放器和扩展解耦
- 扩展之间不知道对方存在

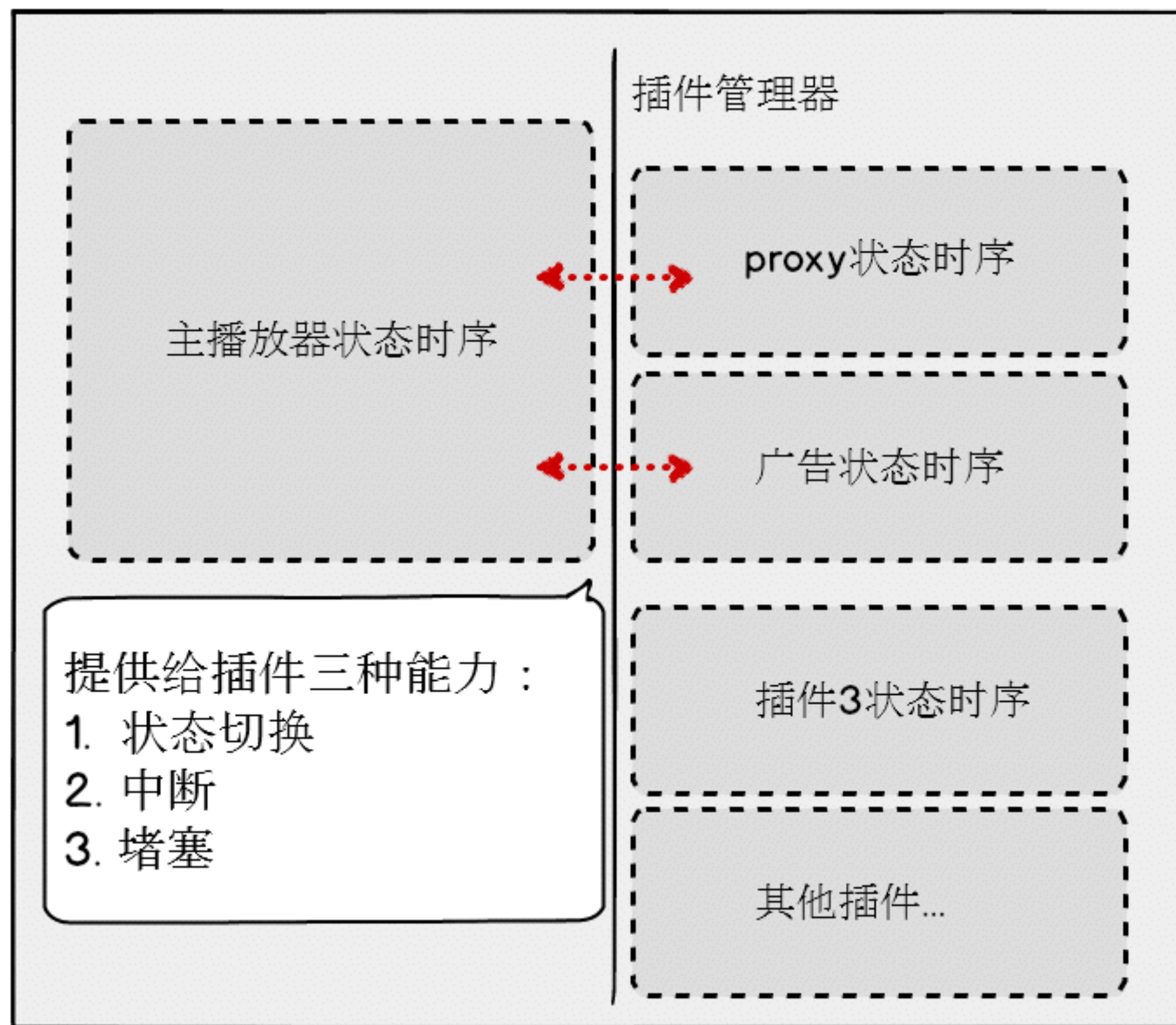
“可热插拔”的思路

对外层看来就是一个统一的播放器



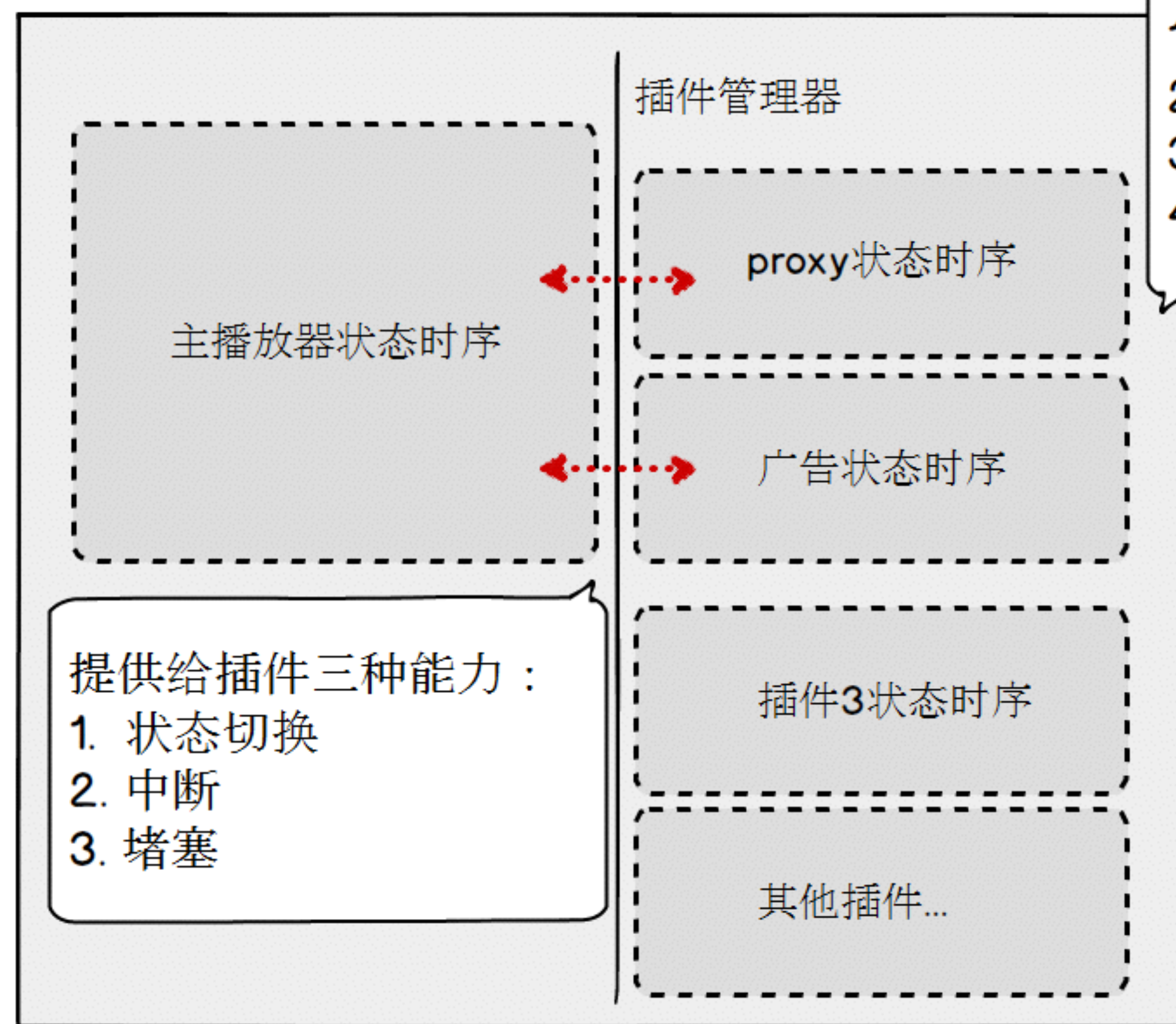
“可热插拔”的思路

对外层看来就是一个统一的播放器



“可热插拔”的思路

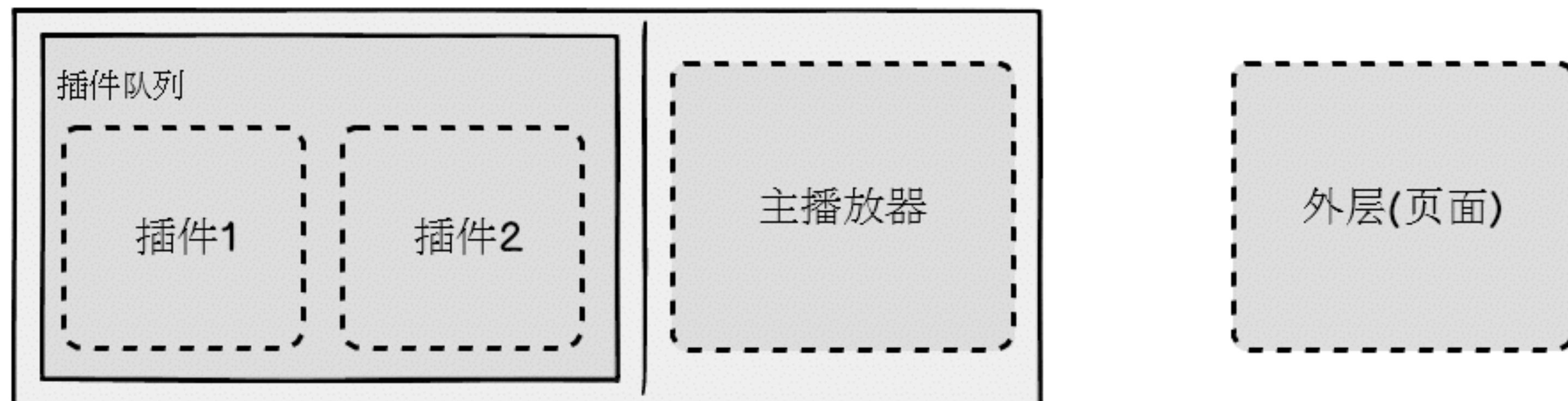
对外层看来就是一个统一的播放器



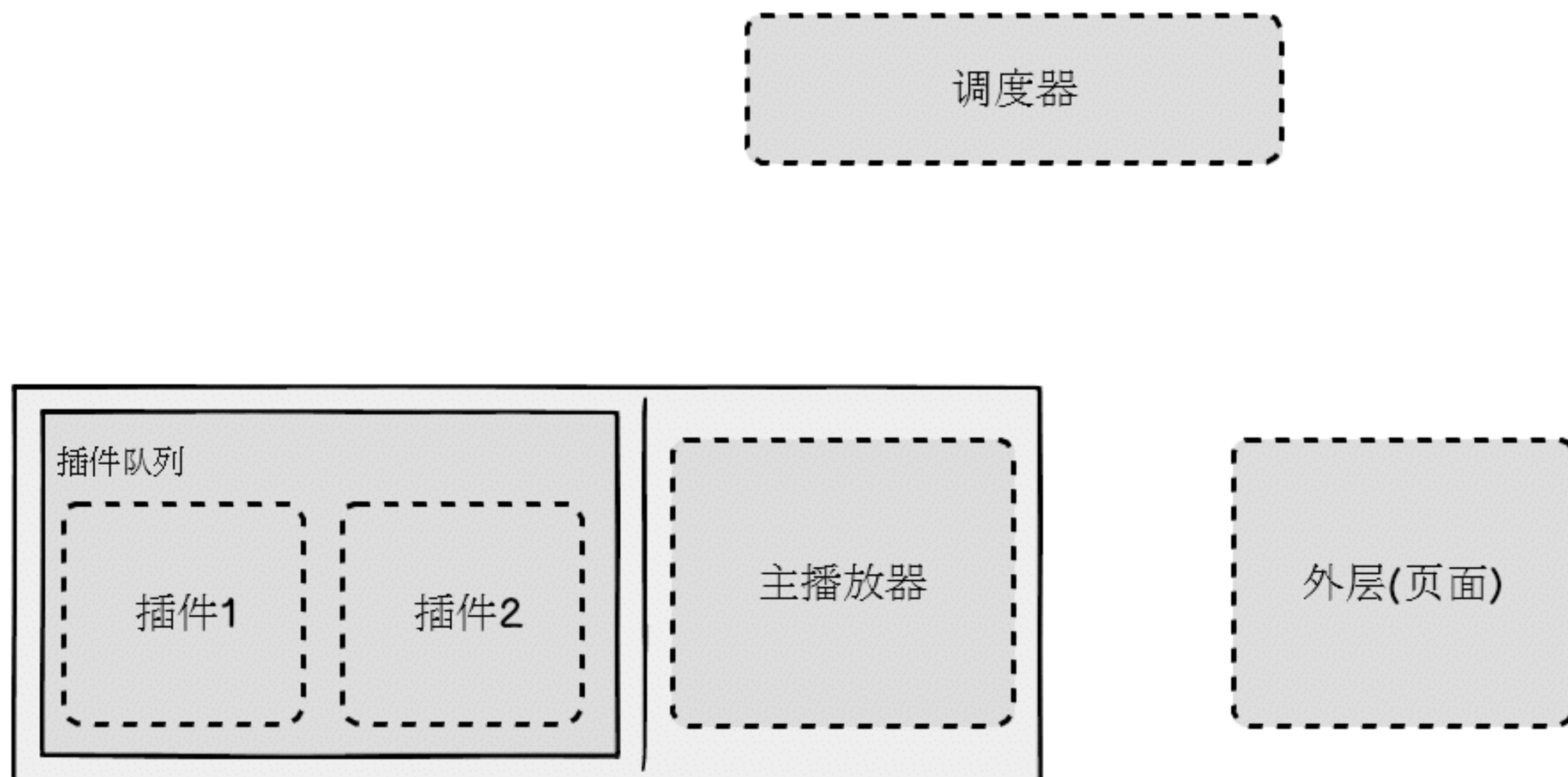
“可热插拔”的设计：

1. 插件只关心内部状态时序
2. 插件可灰度
3. 可并存多个插件
4. 开发极其简单

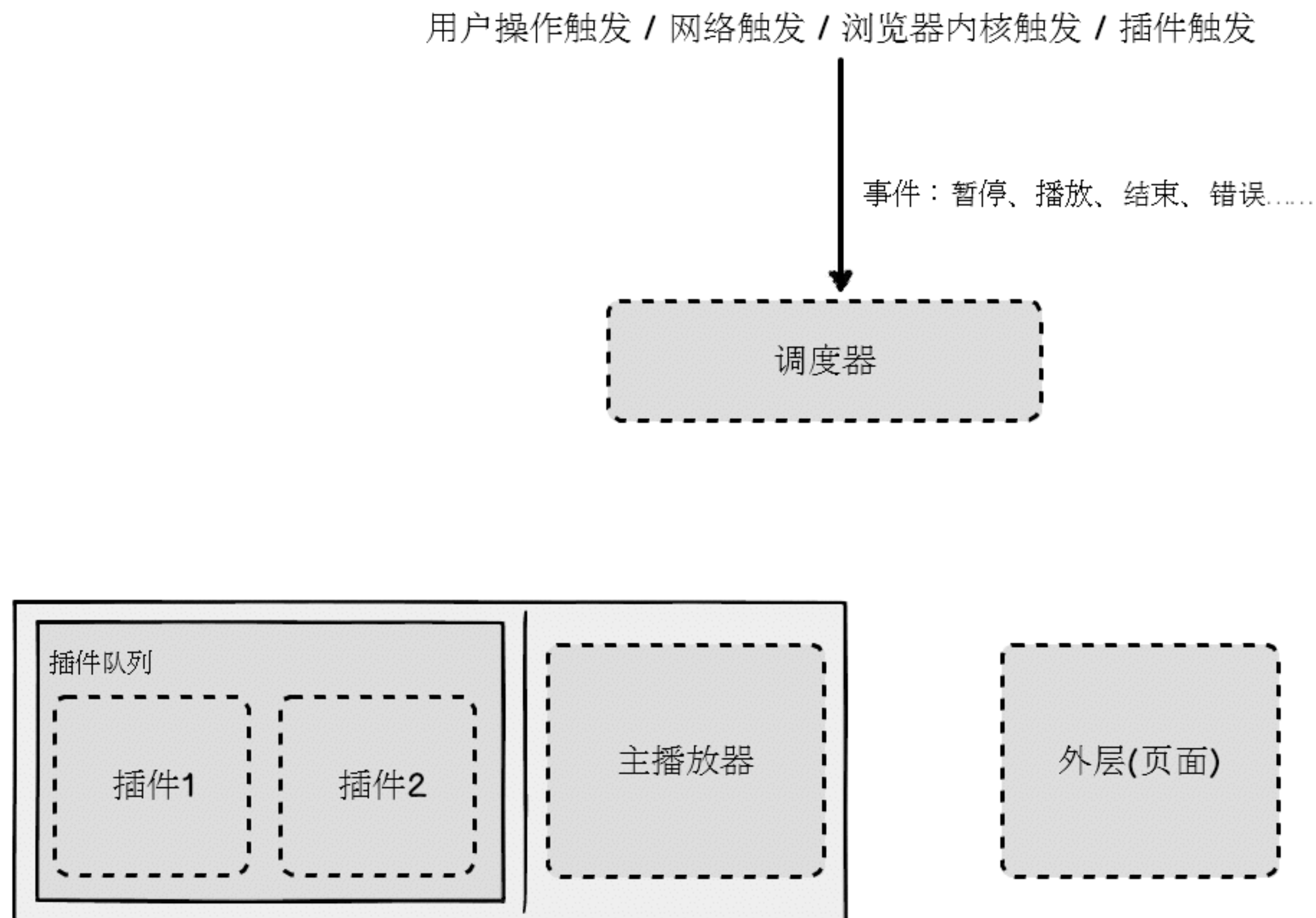
三个关键角色



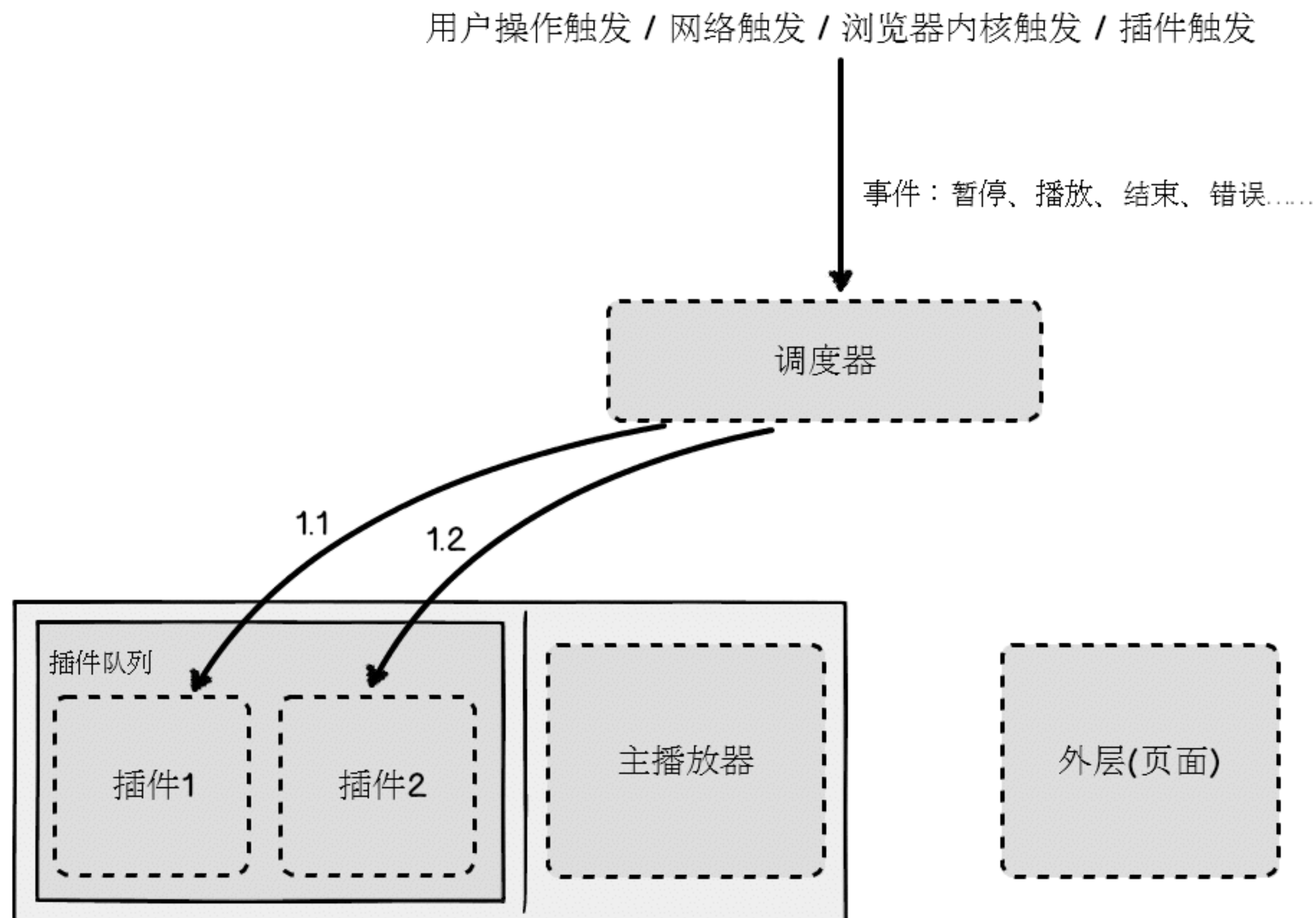
状态调度器



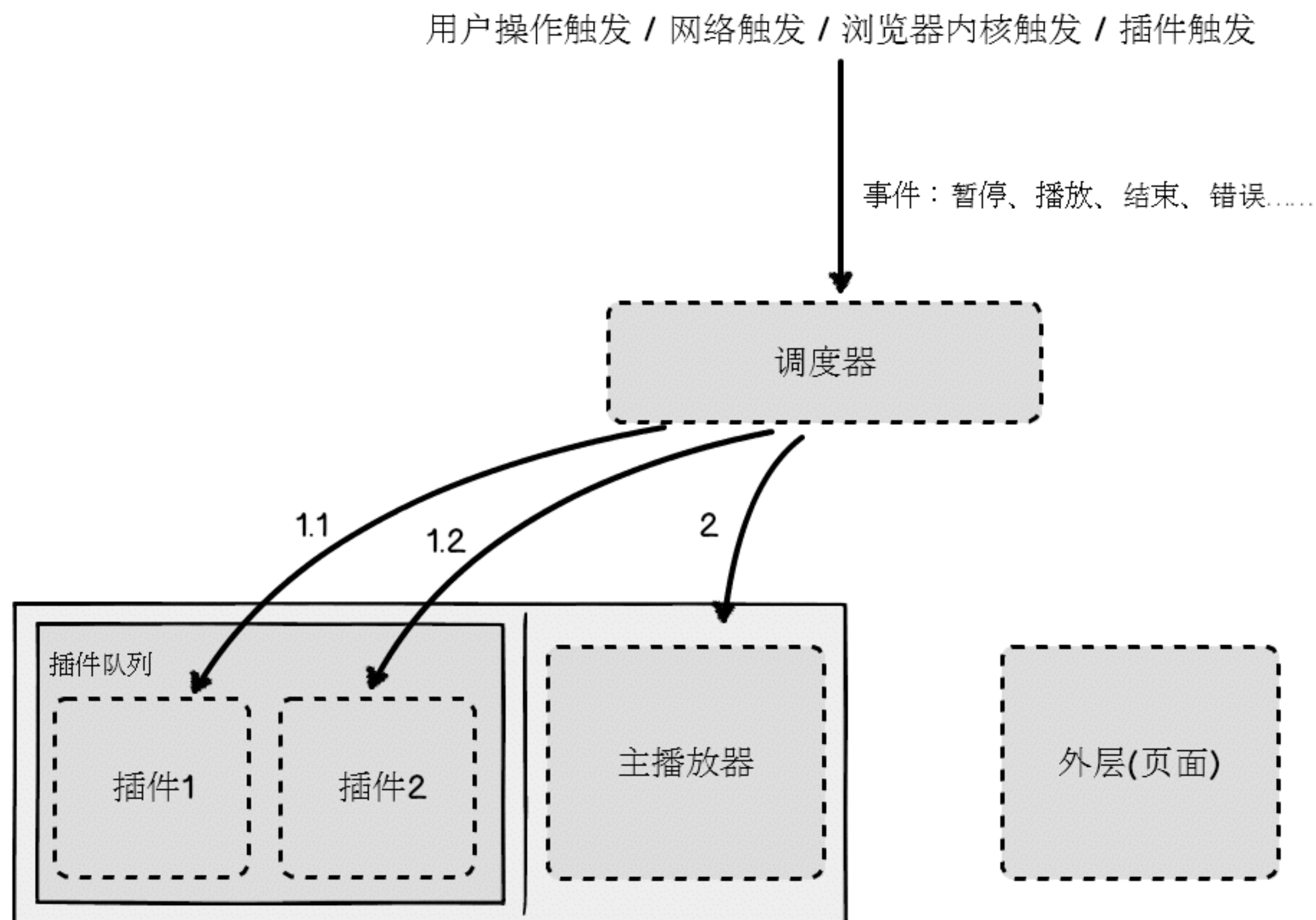
状态调度器的作用



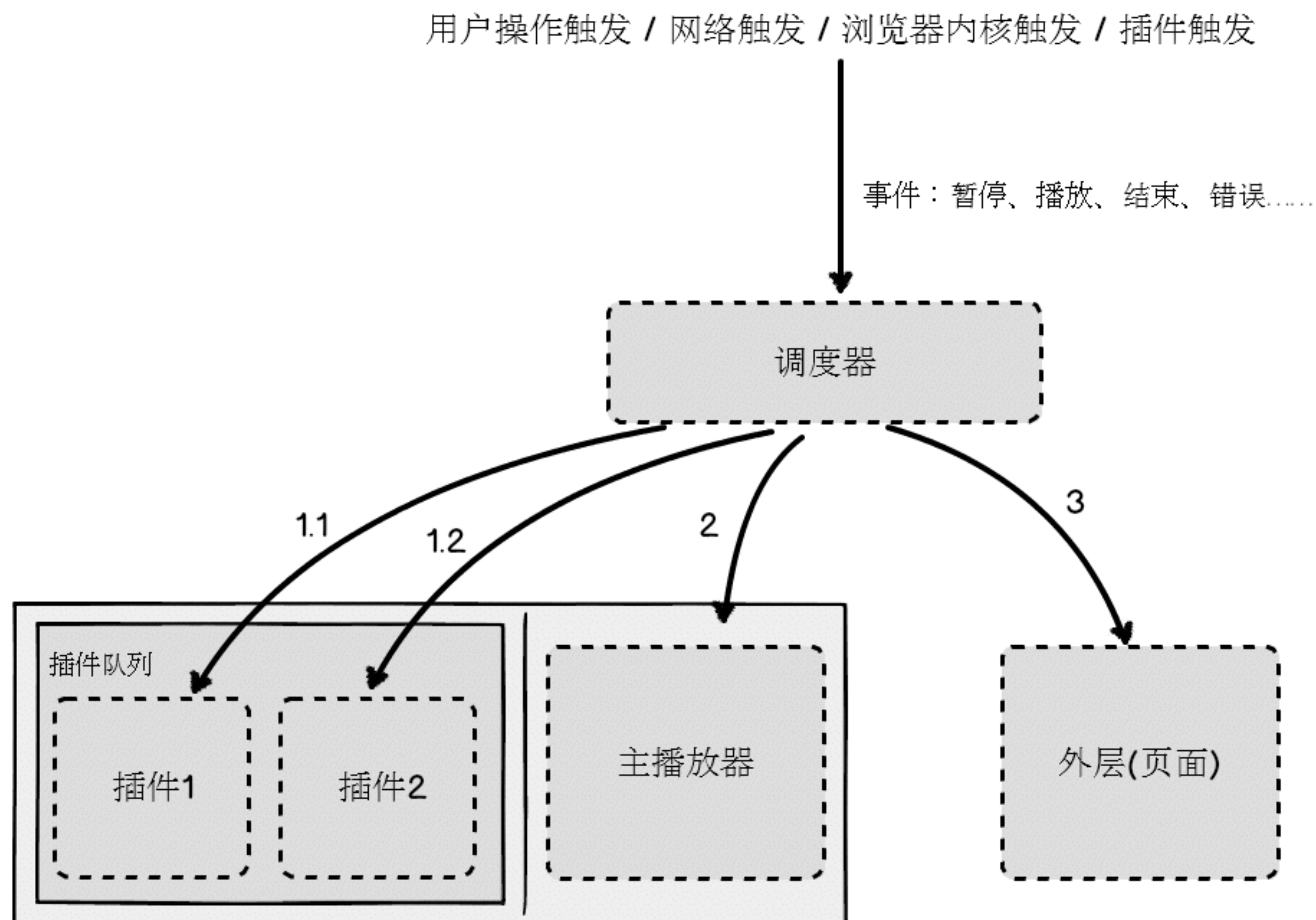
状态调度器的作用



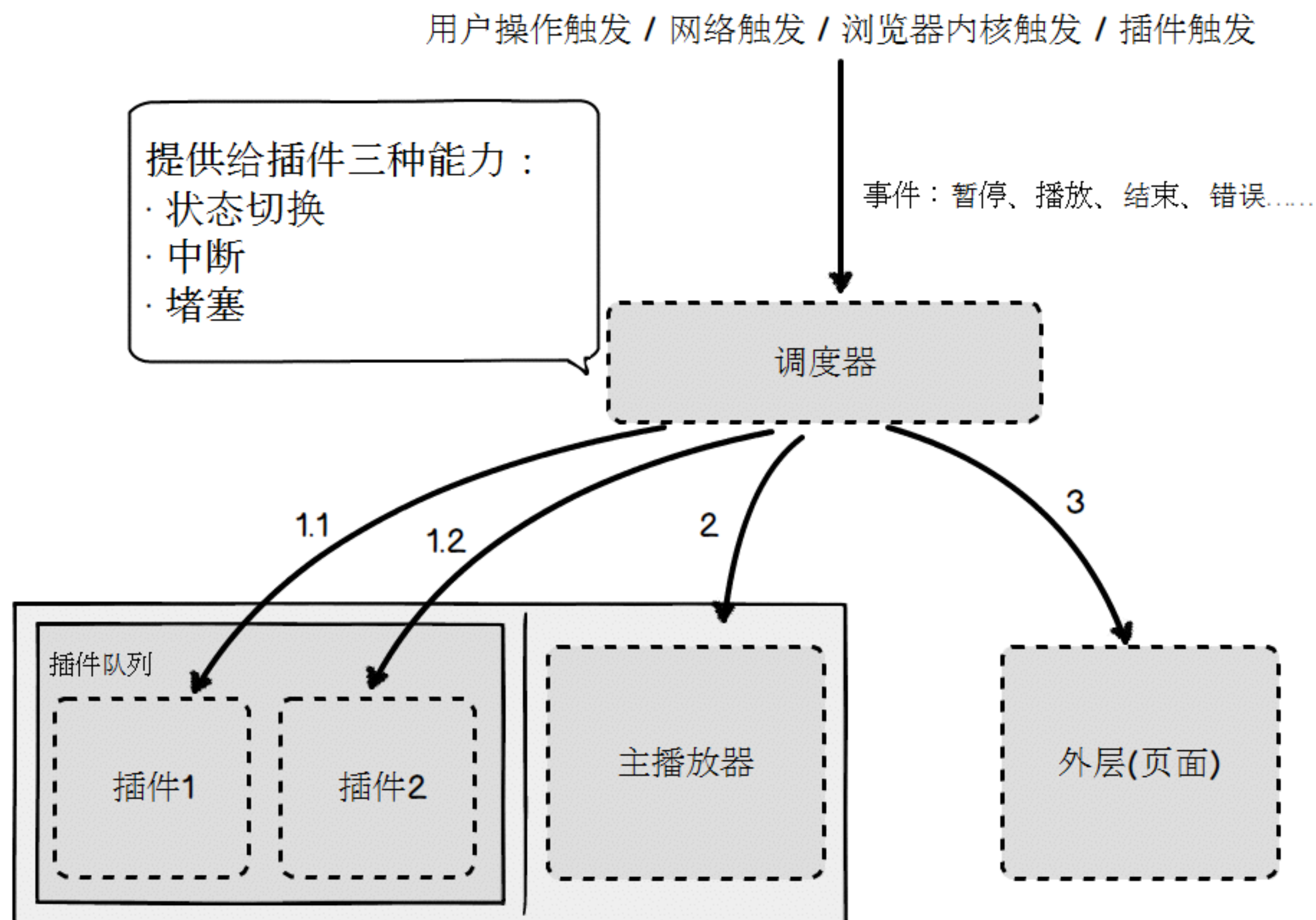
状态调度器的作用



状态调度器的作用



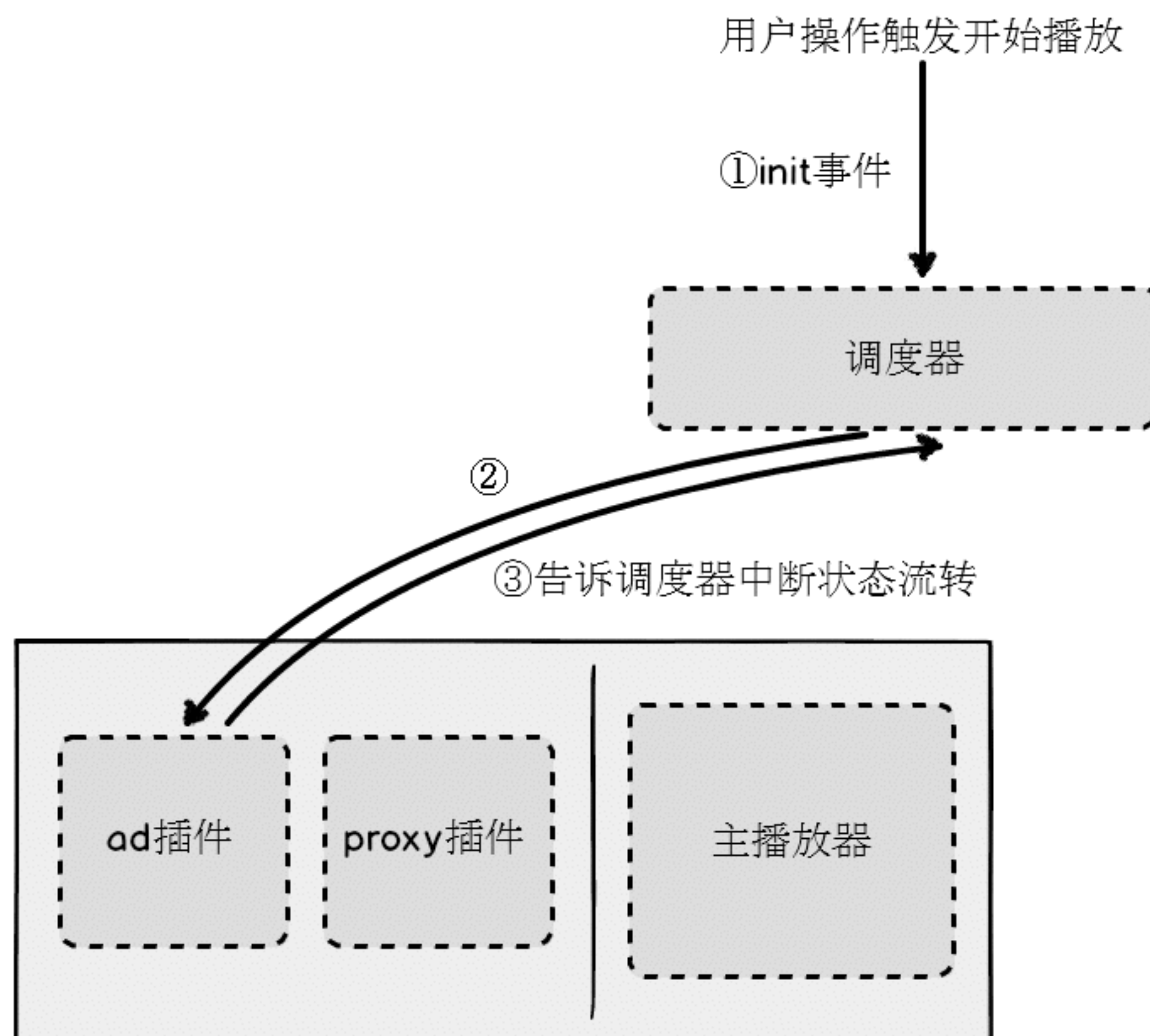
状态调度器的作用



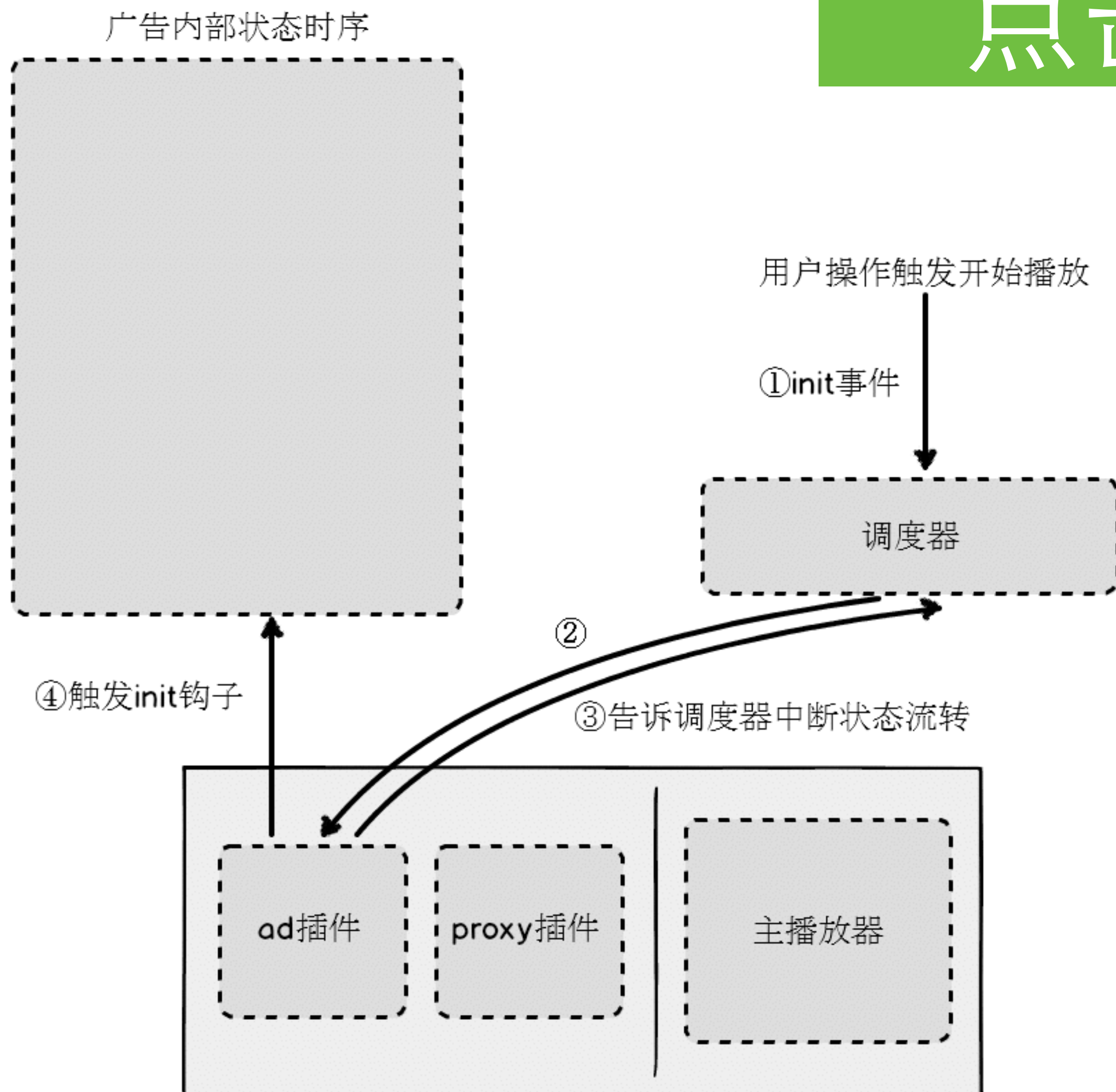
这个状态调度器能否完成前边的需求？

点击播放视频

点击播放

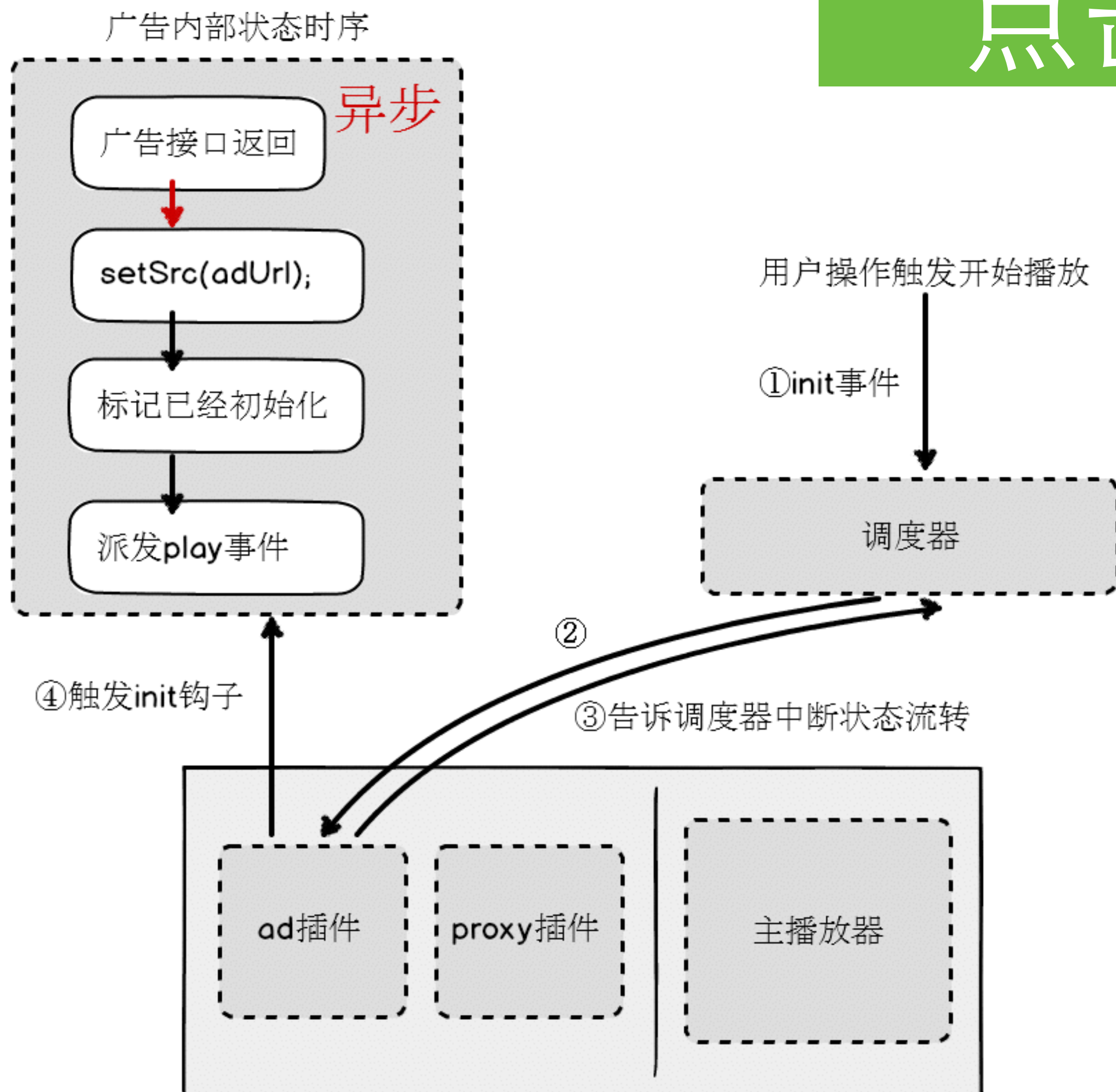


点击播放视频



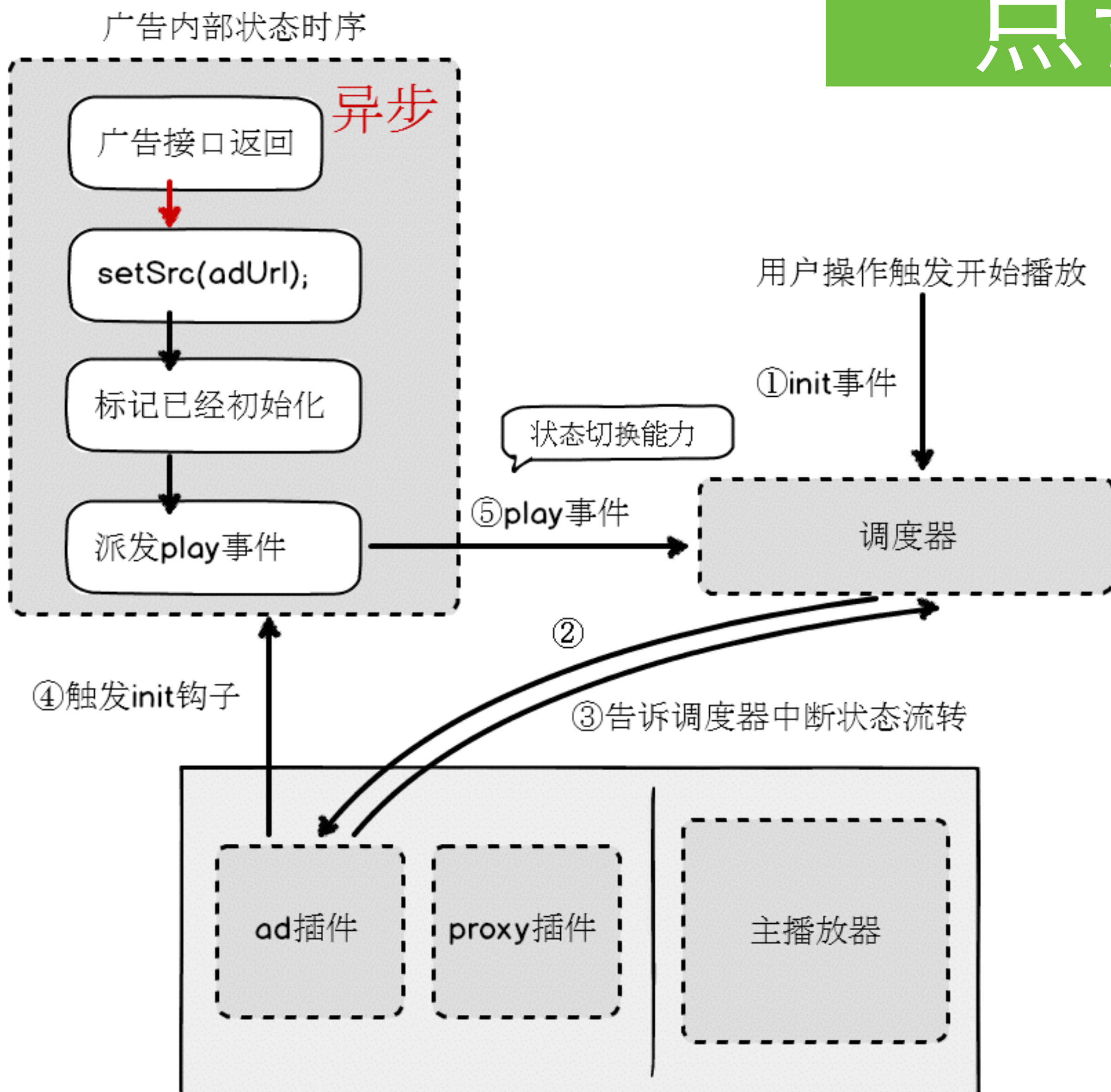
点击播放

点击播放视频



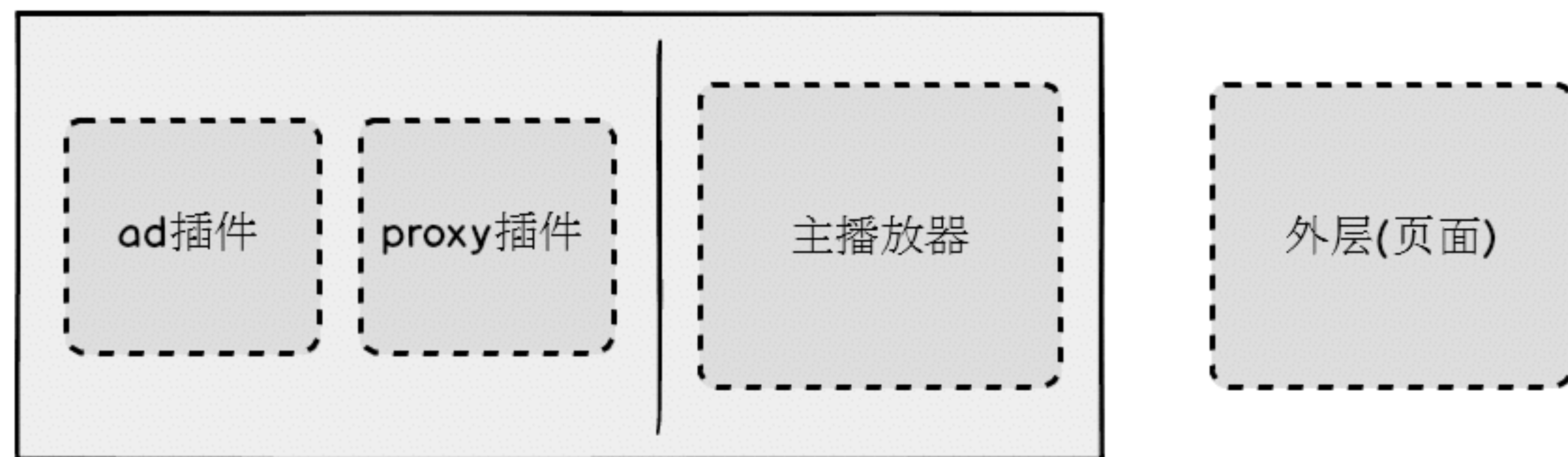
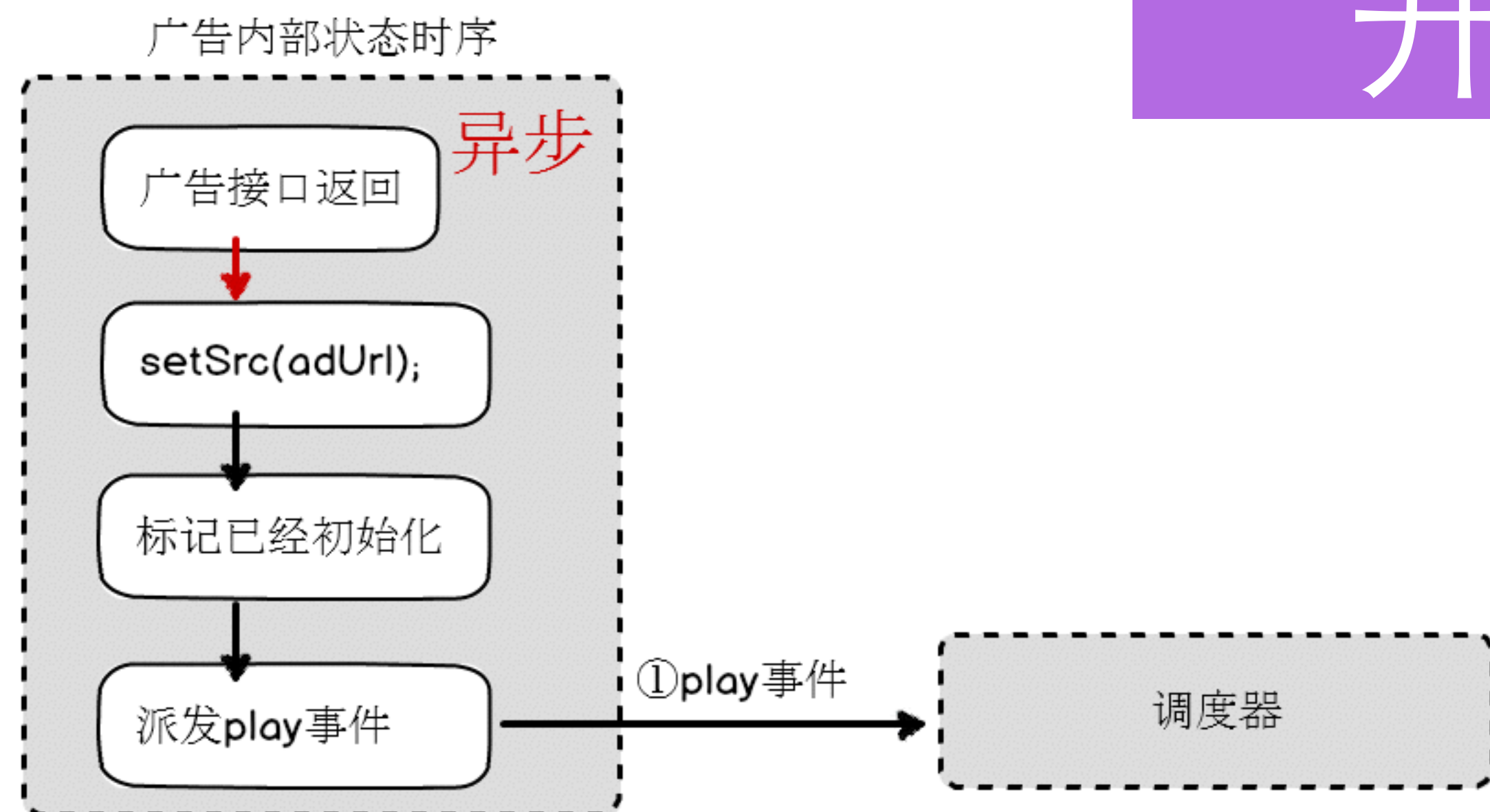
点击播放

点击播放视频

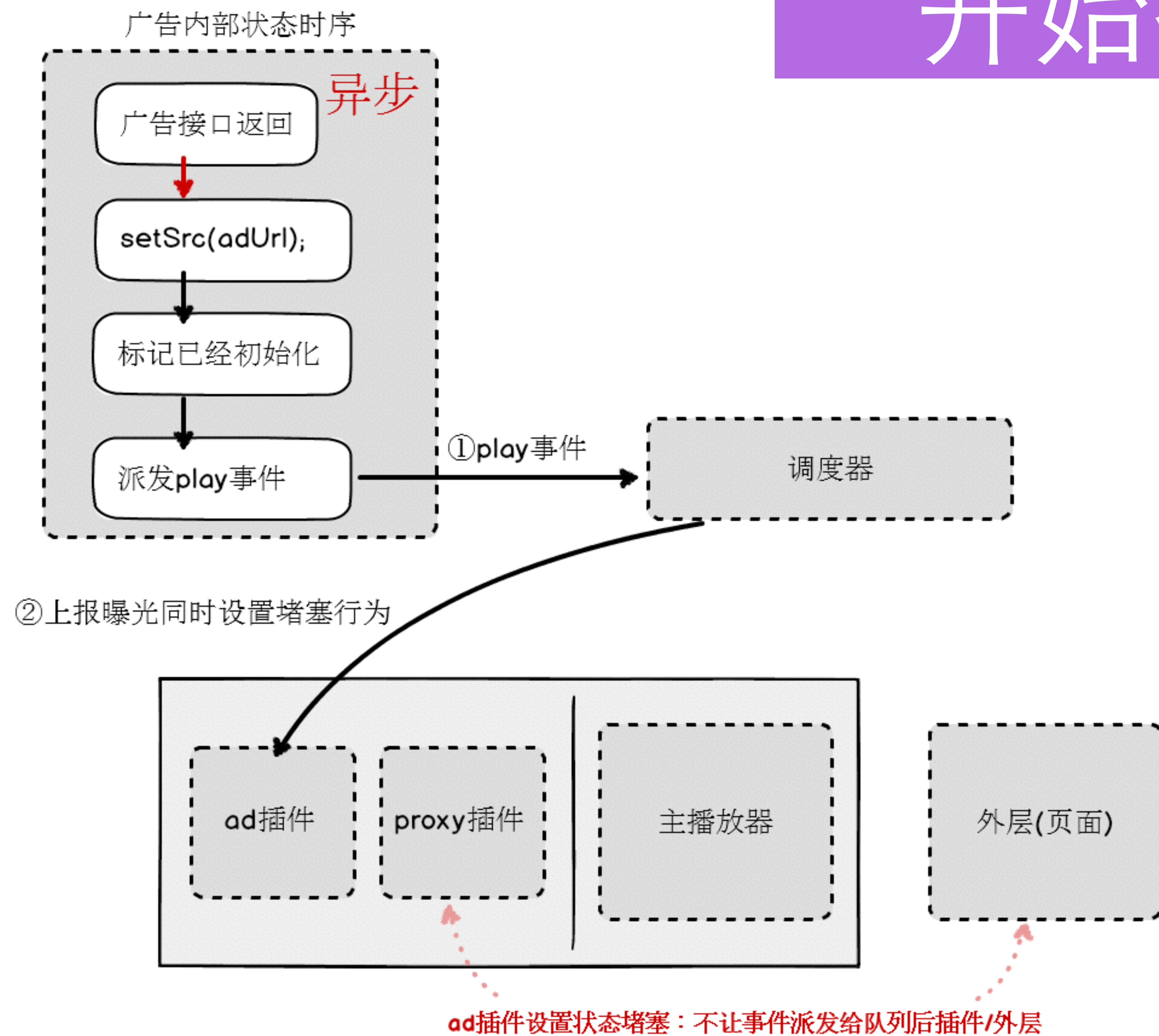


点击播放

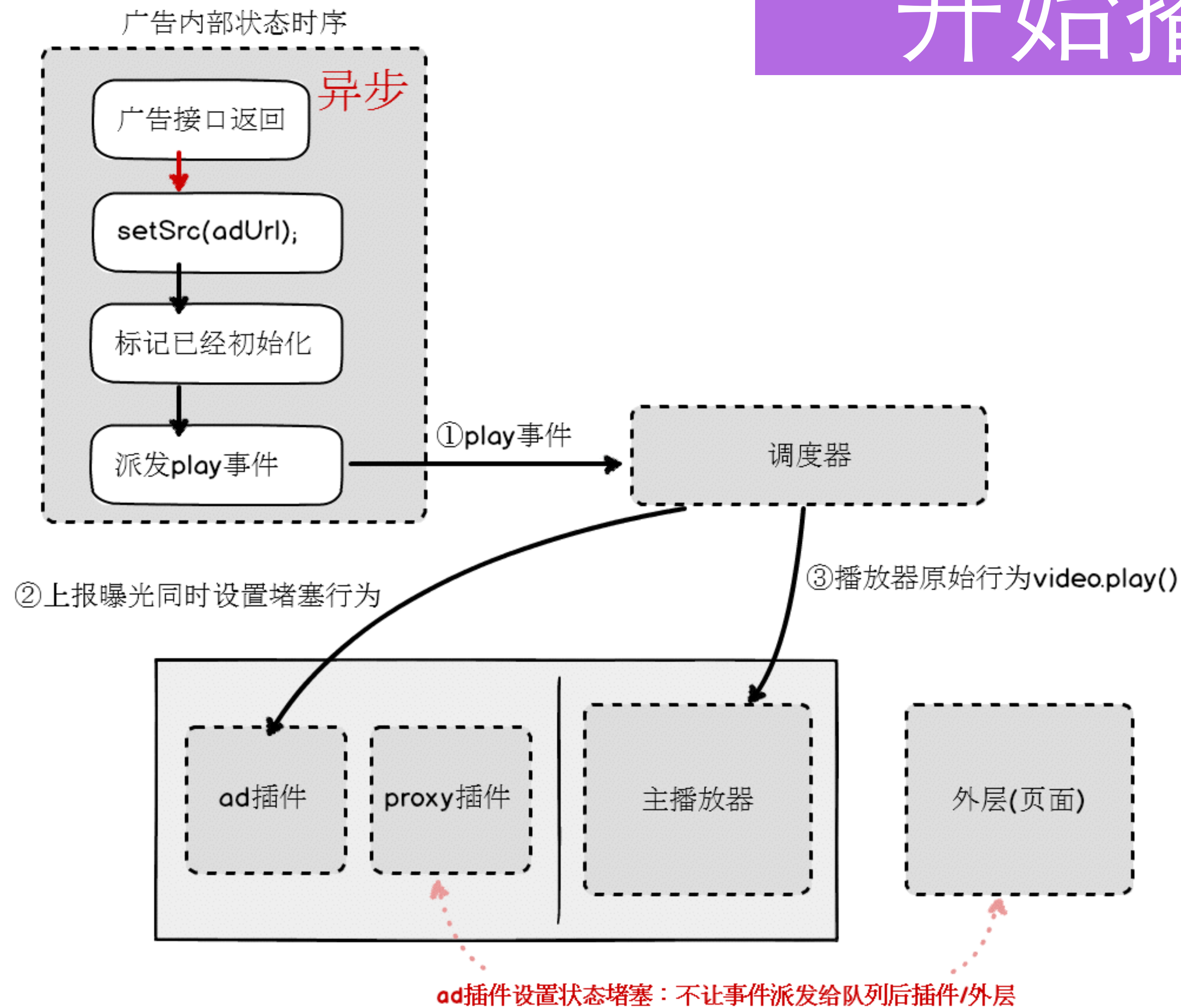
开始播放广告



开始播放广告



开始播放广告



广告结束

视频播放完毕, 浏览器内核触发

①end事件

调度器

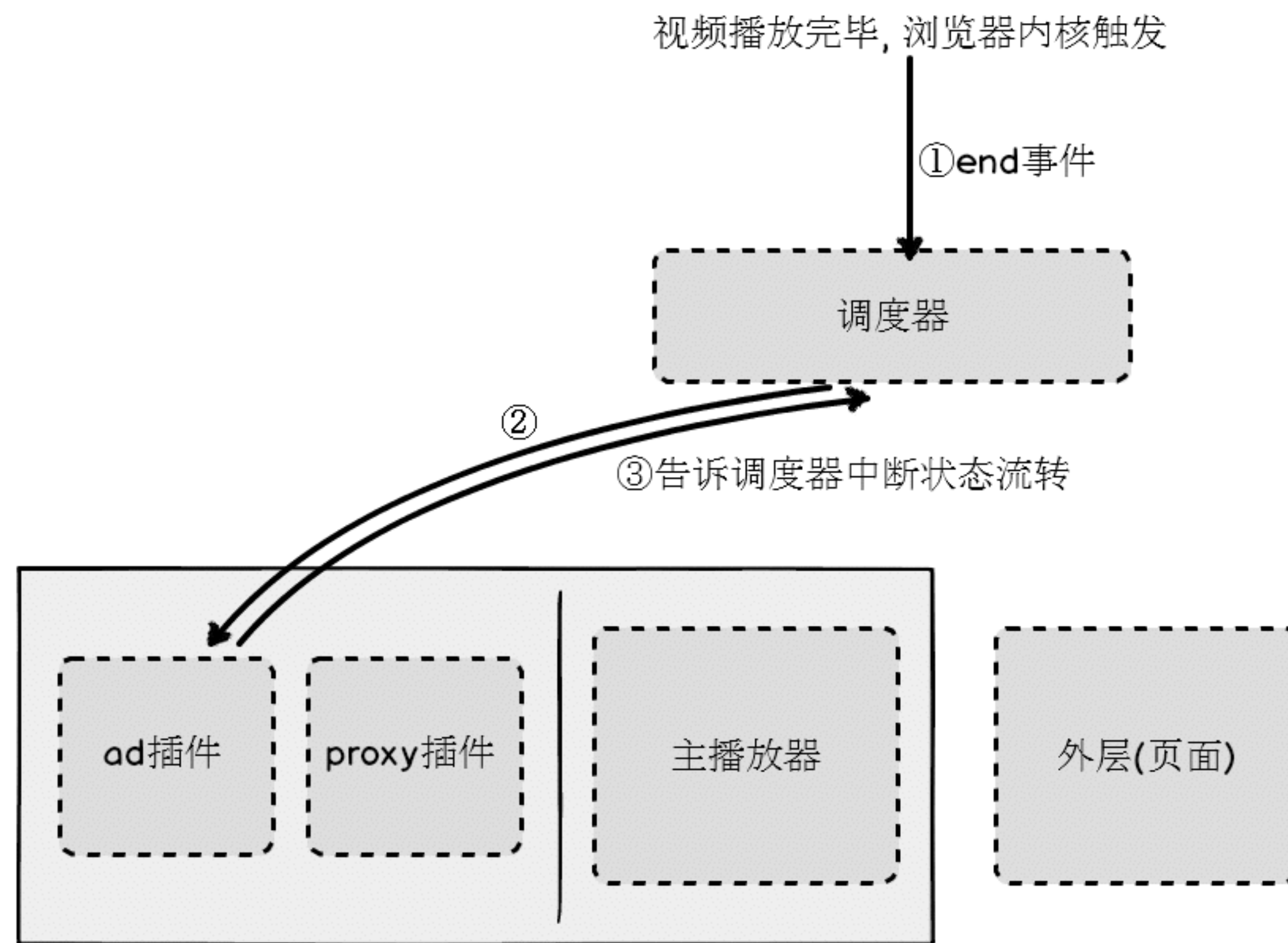
ad插件

proxy插件

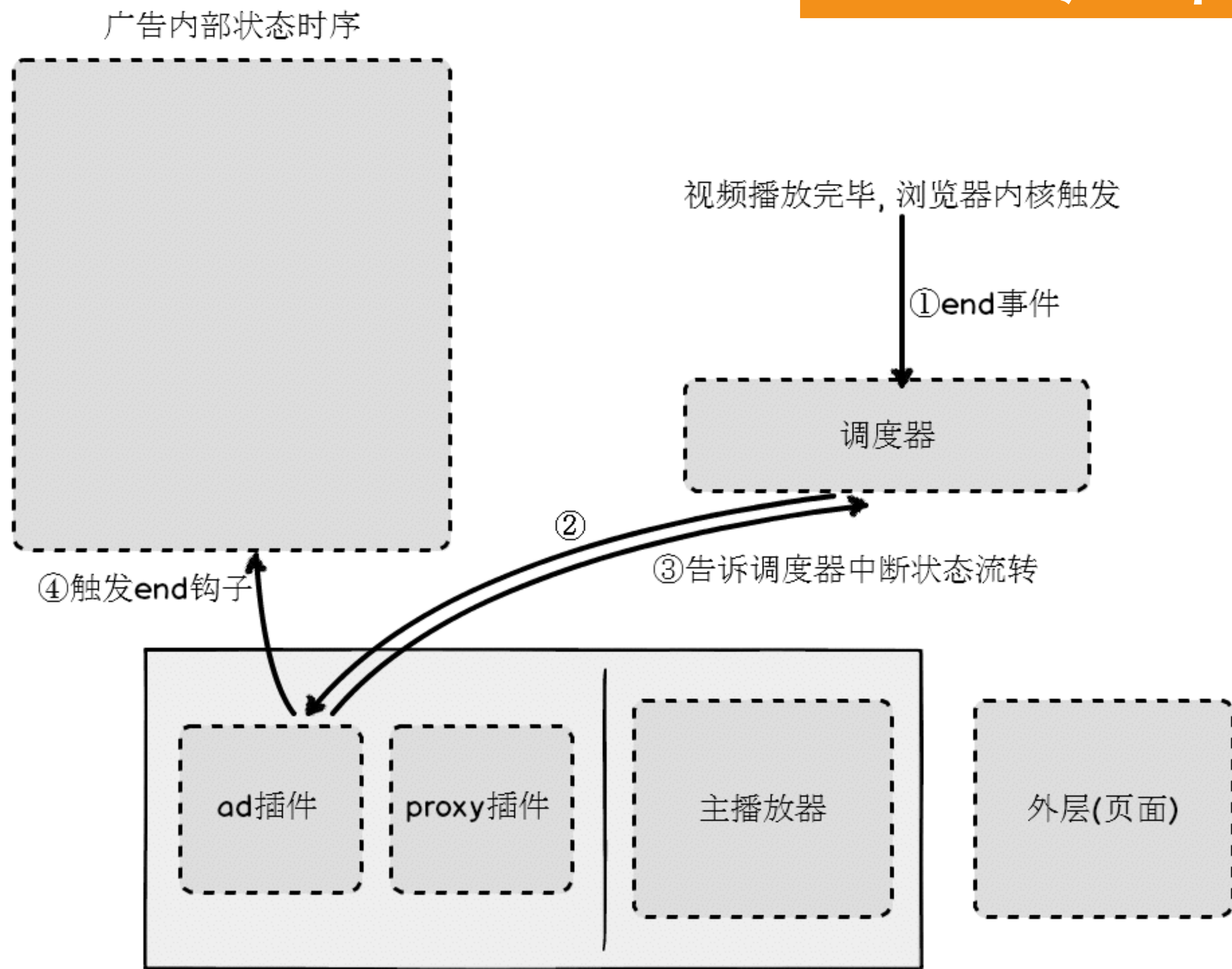
主播放器

外层(页面)

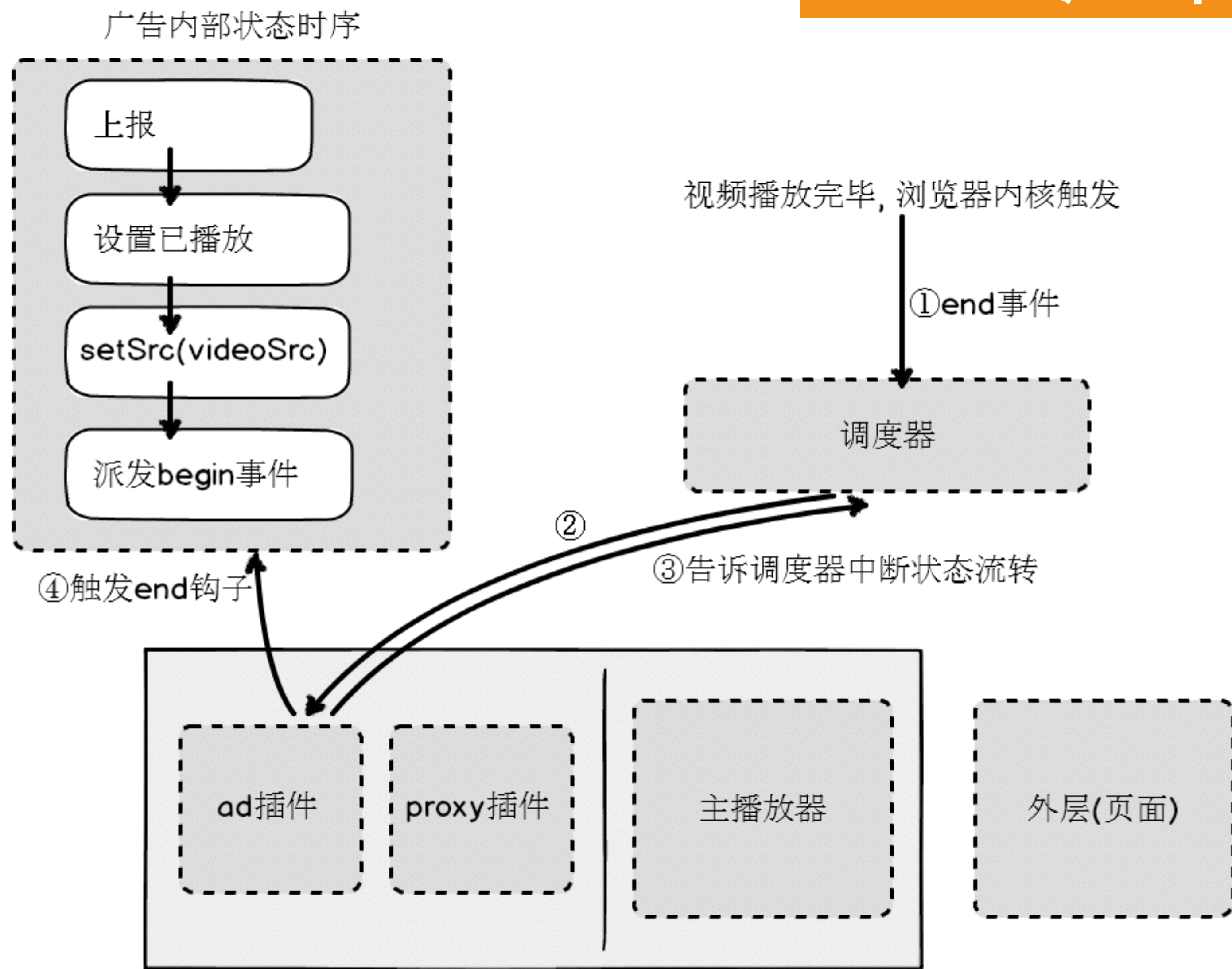
广告结束



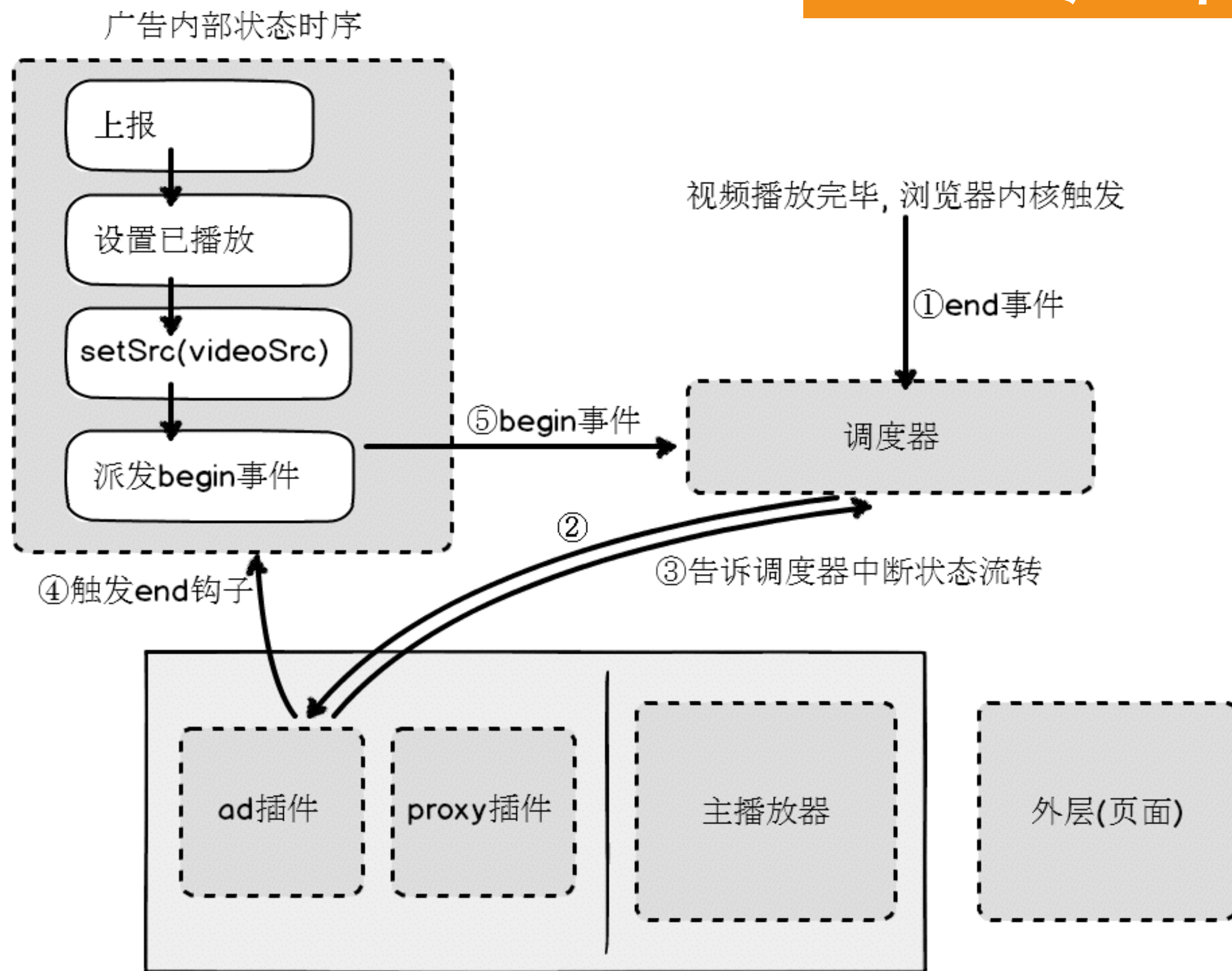
广告结束



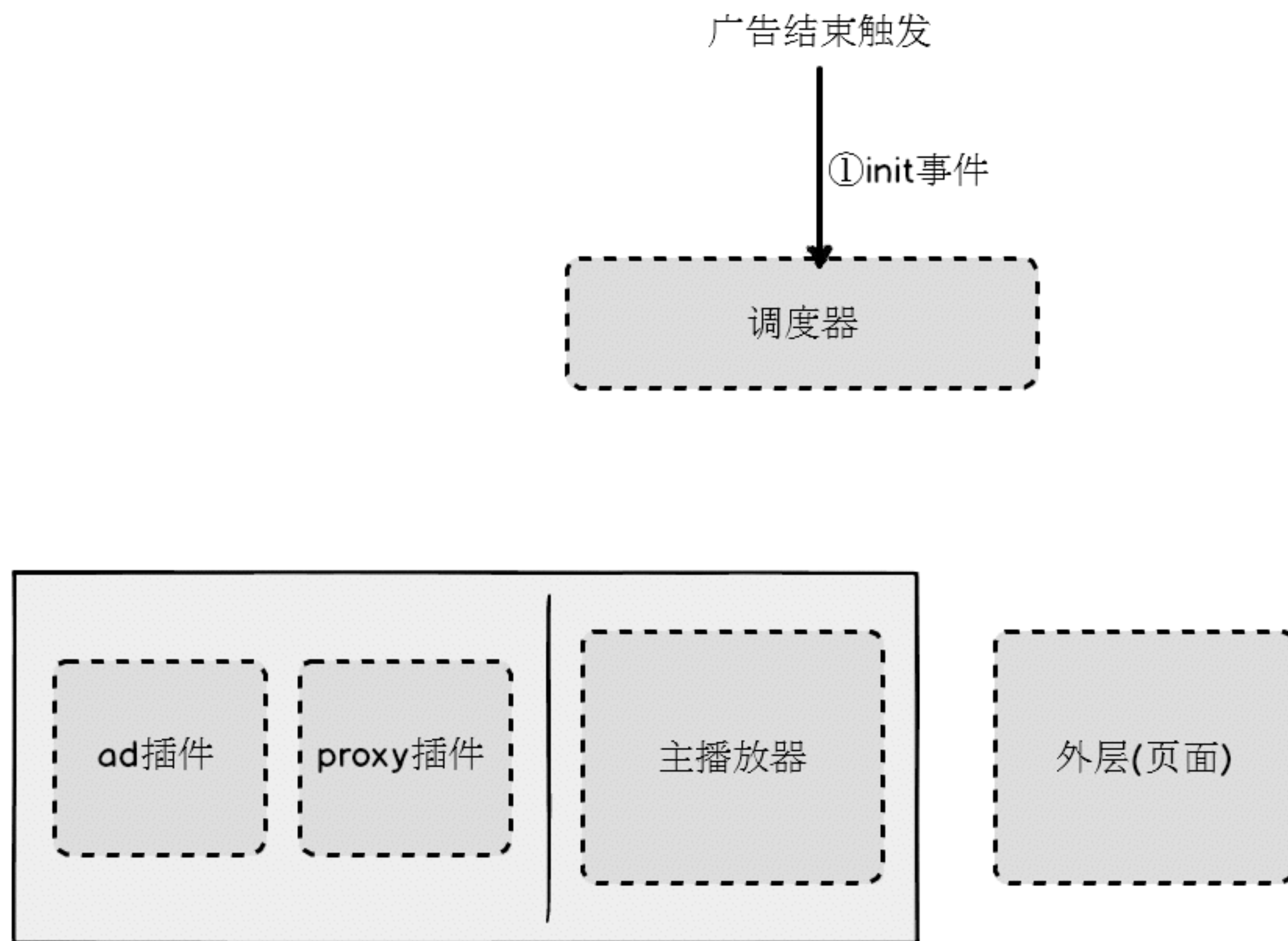
广告结束



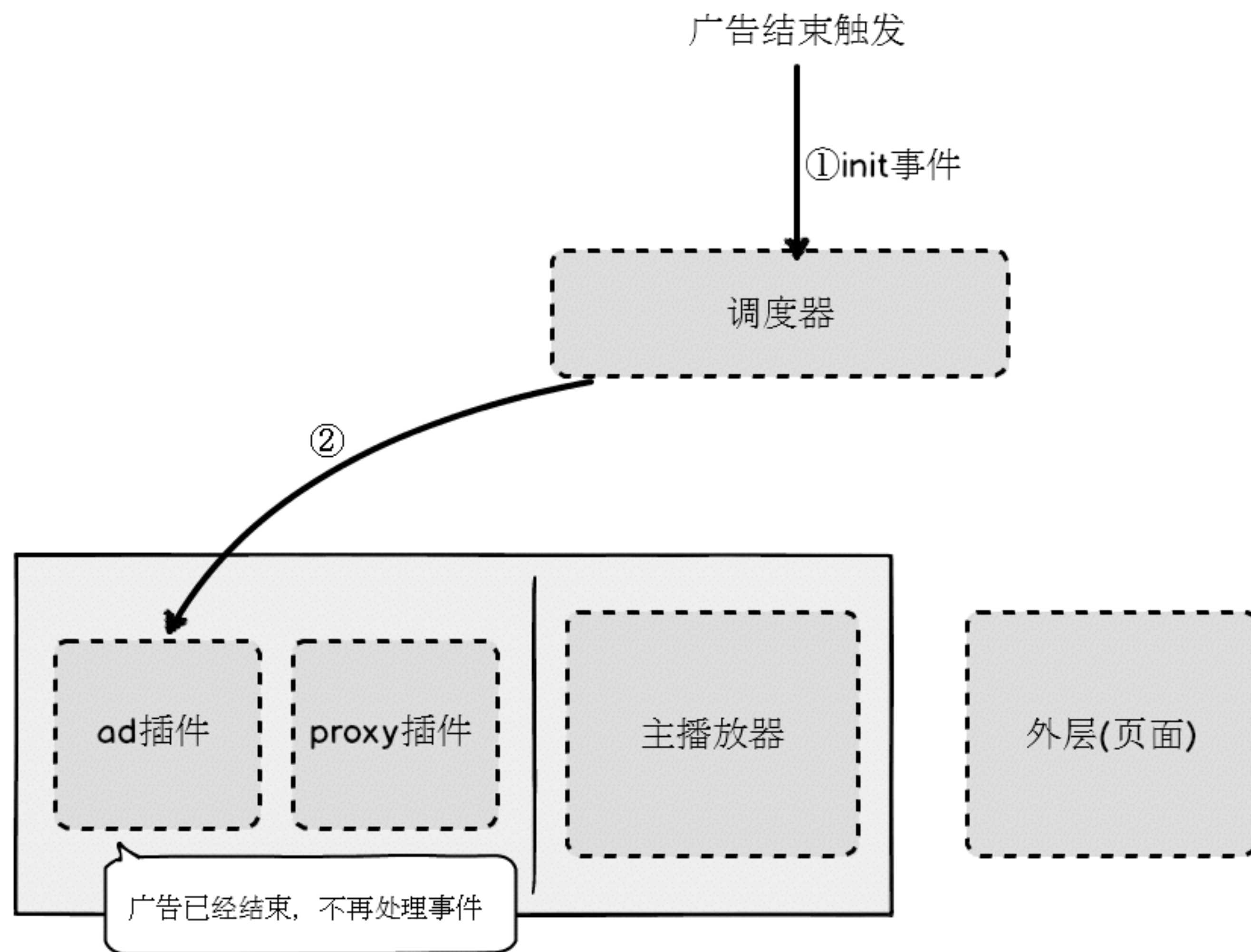
广告结束



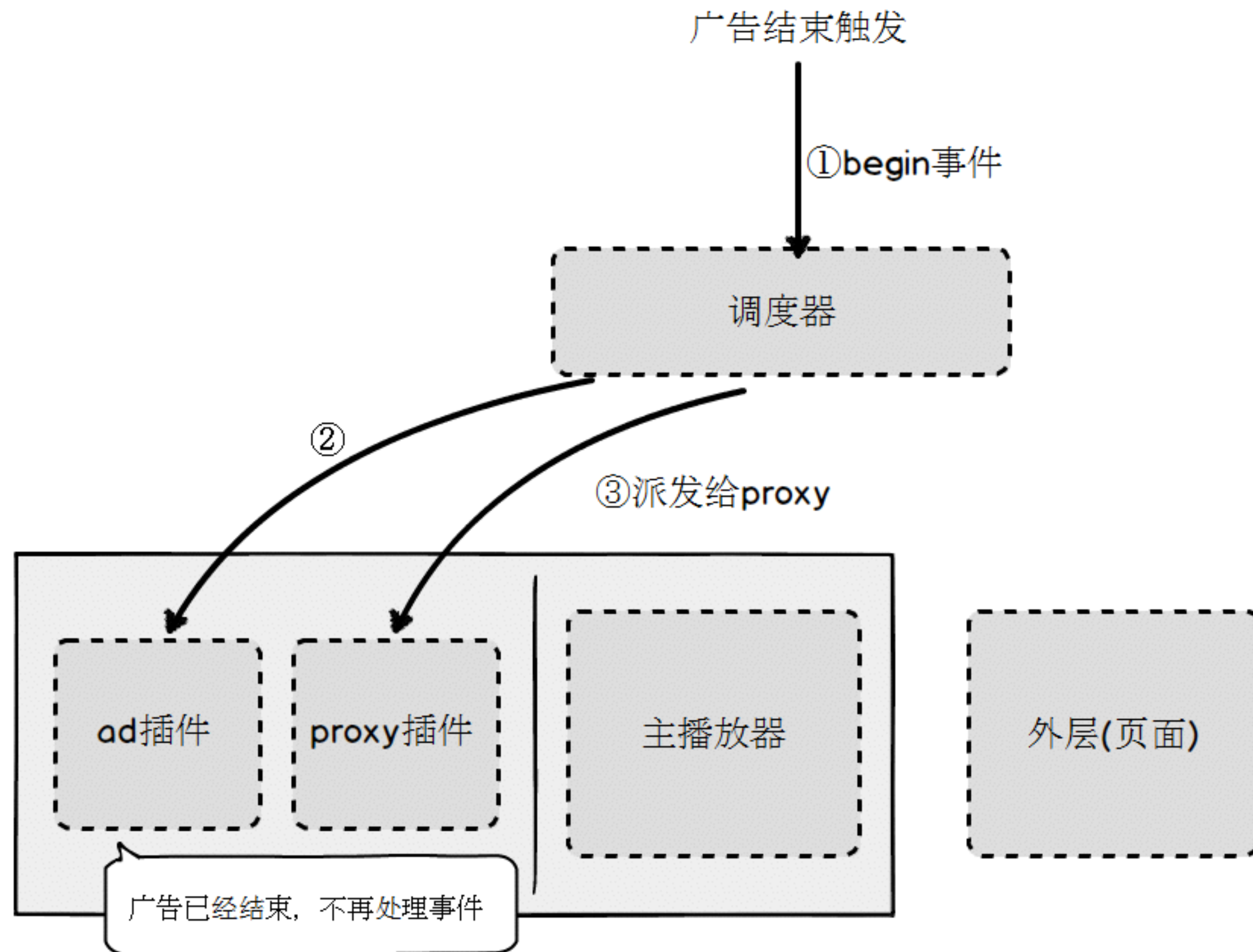
播放主视频



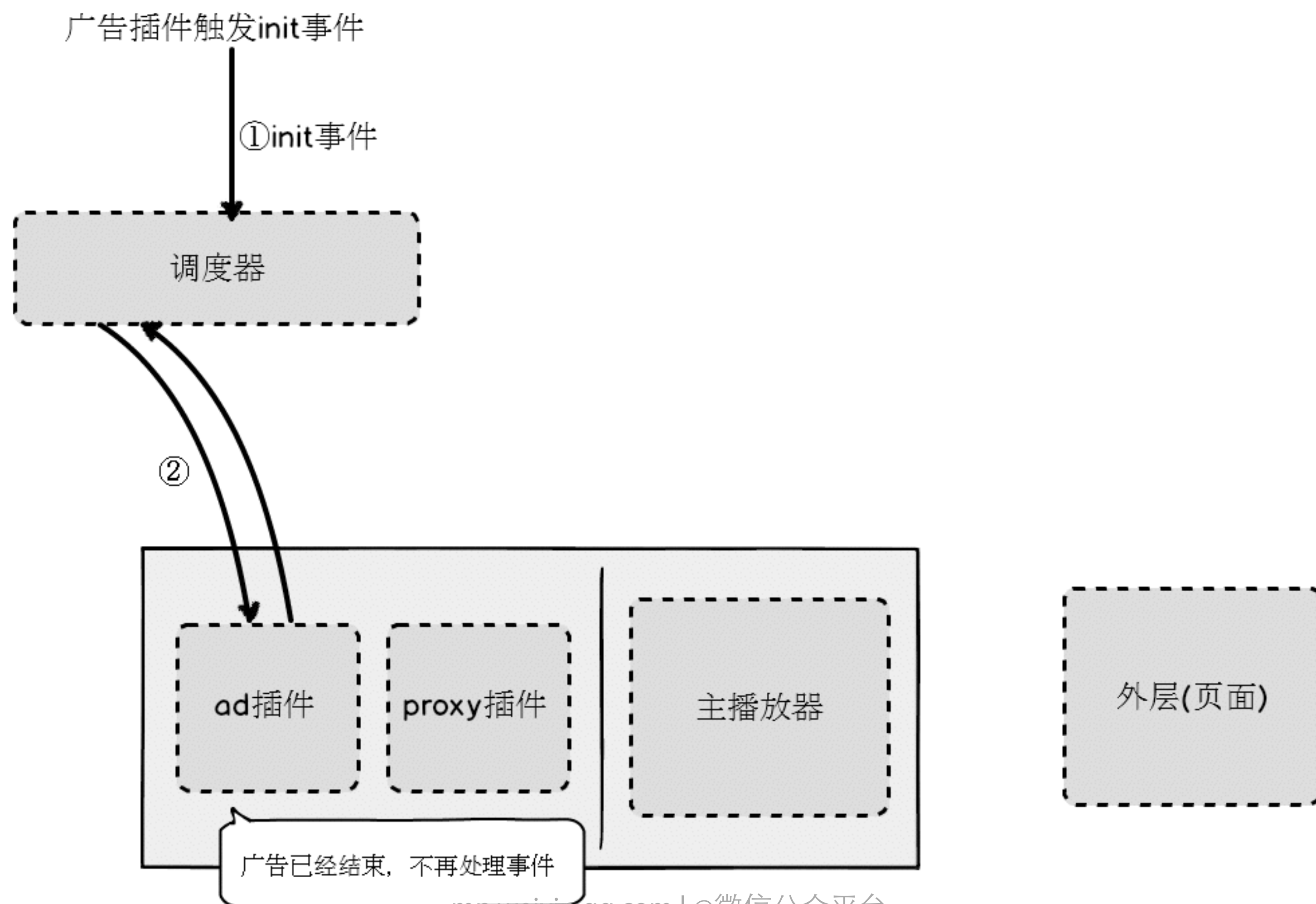
播放主视频



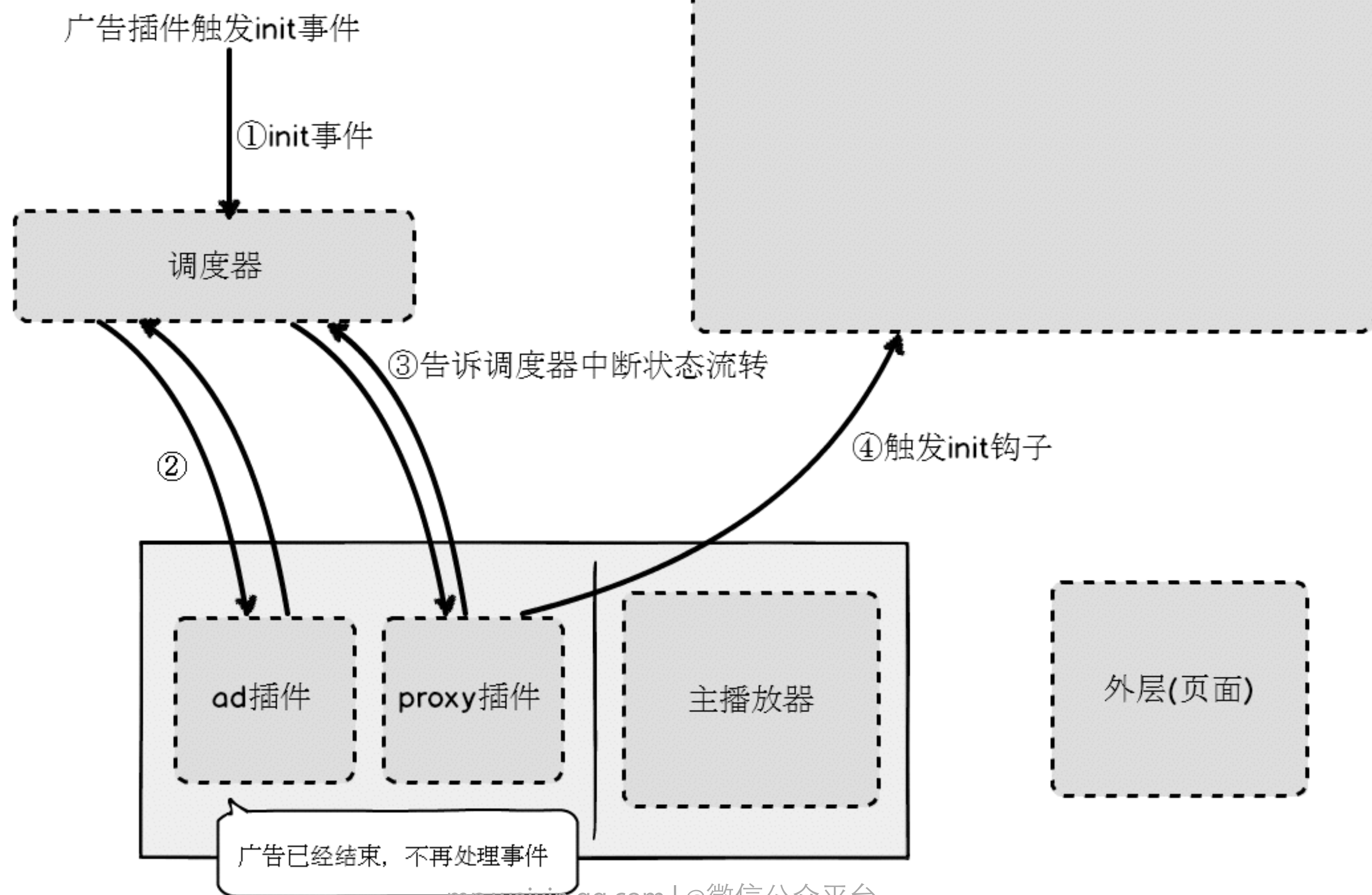
播放主视频



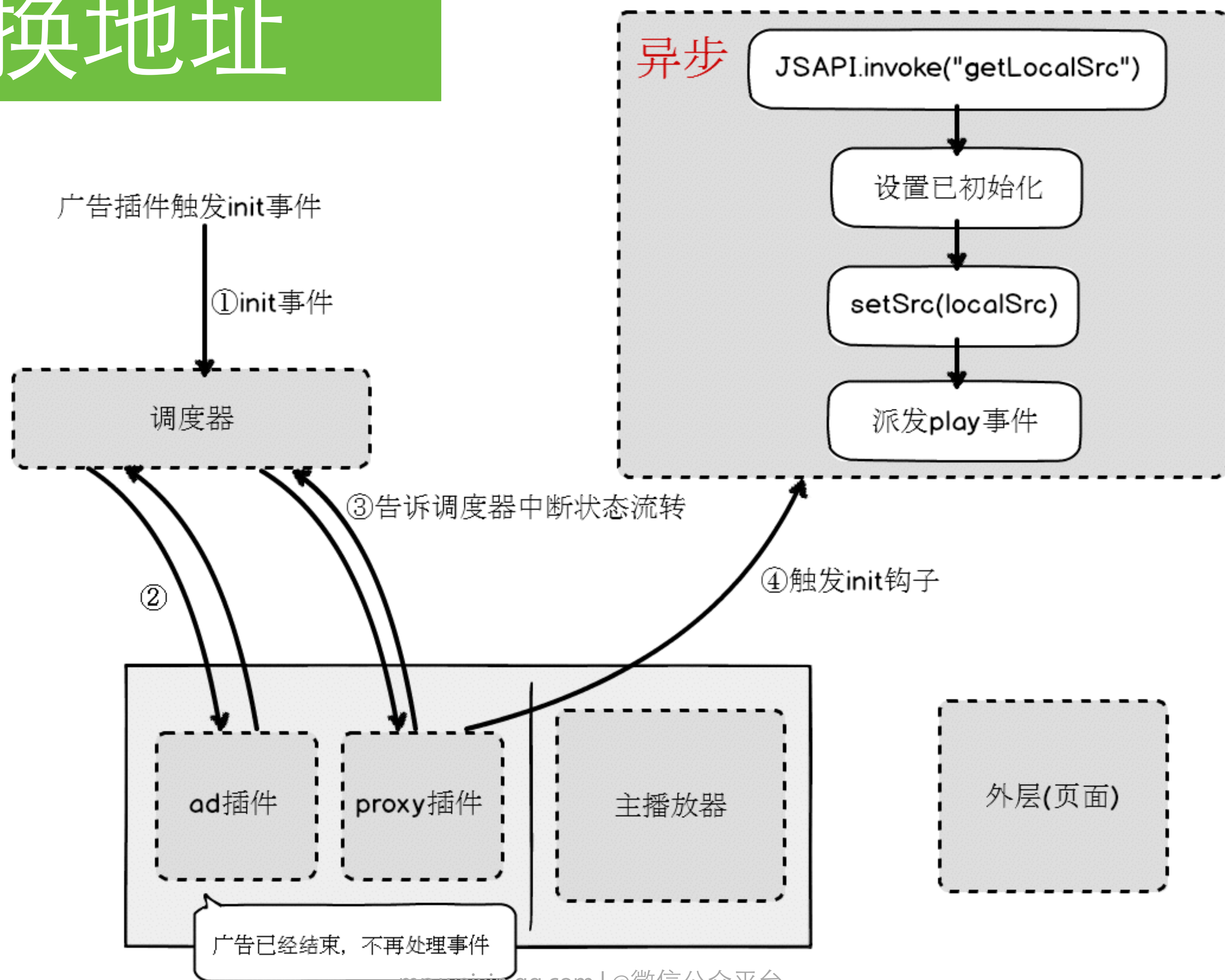
proxy换地址



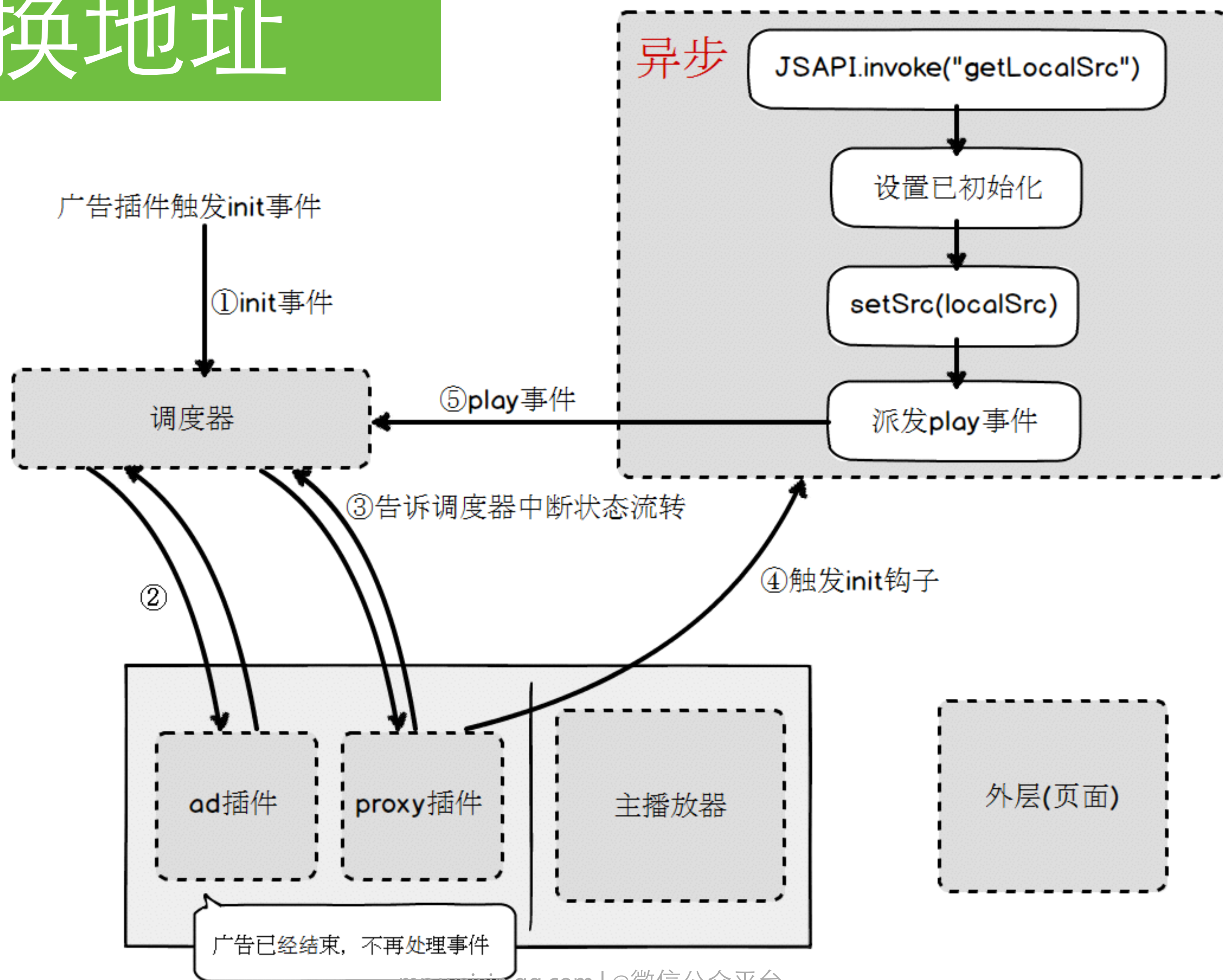
proxy换地址



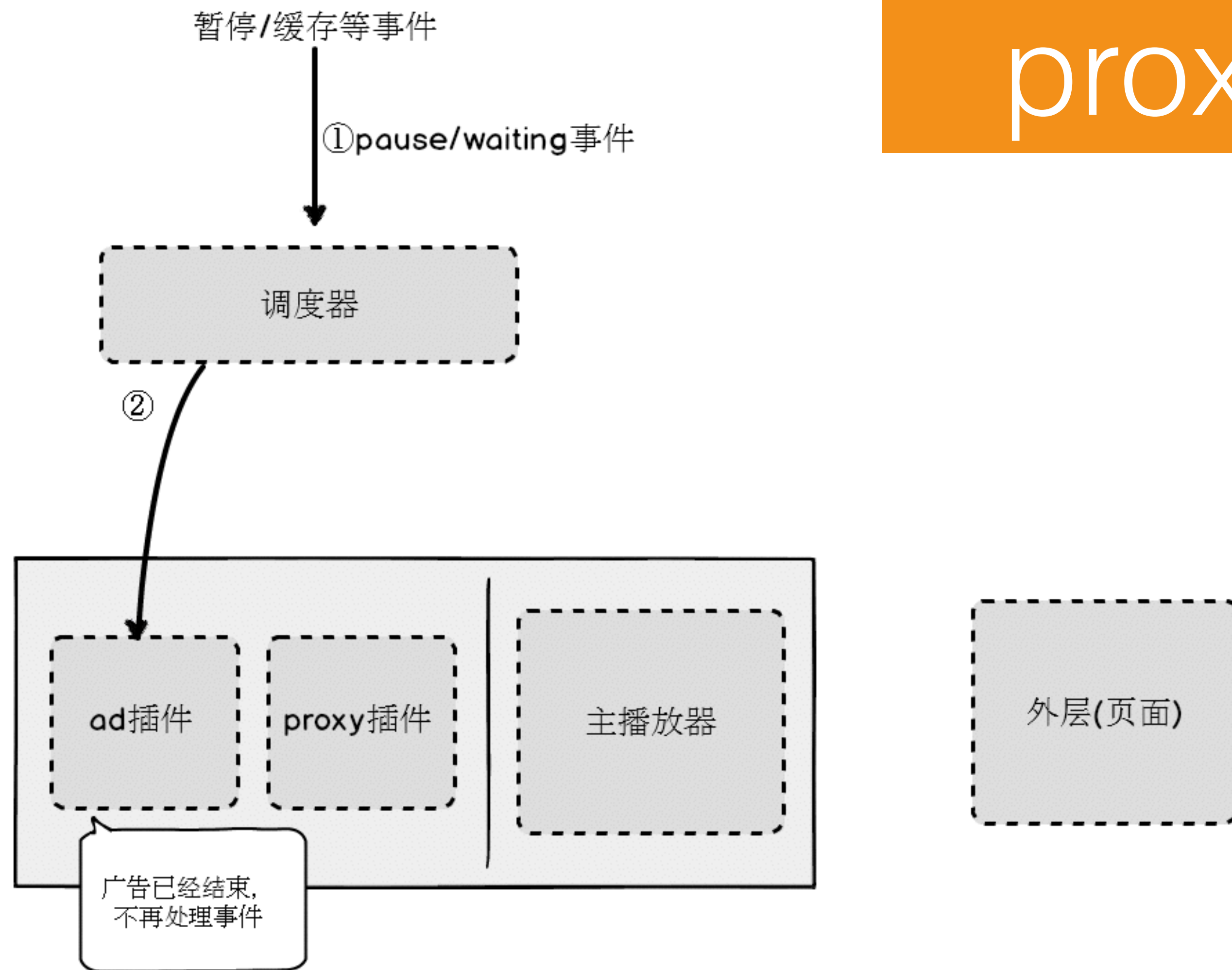
proxy换地址



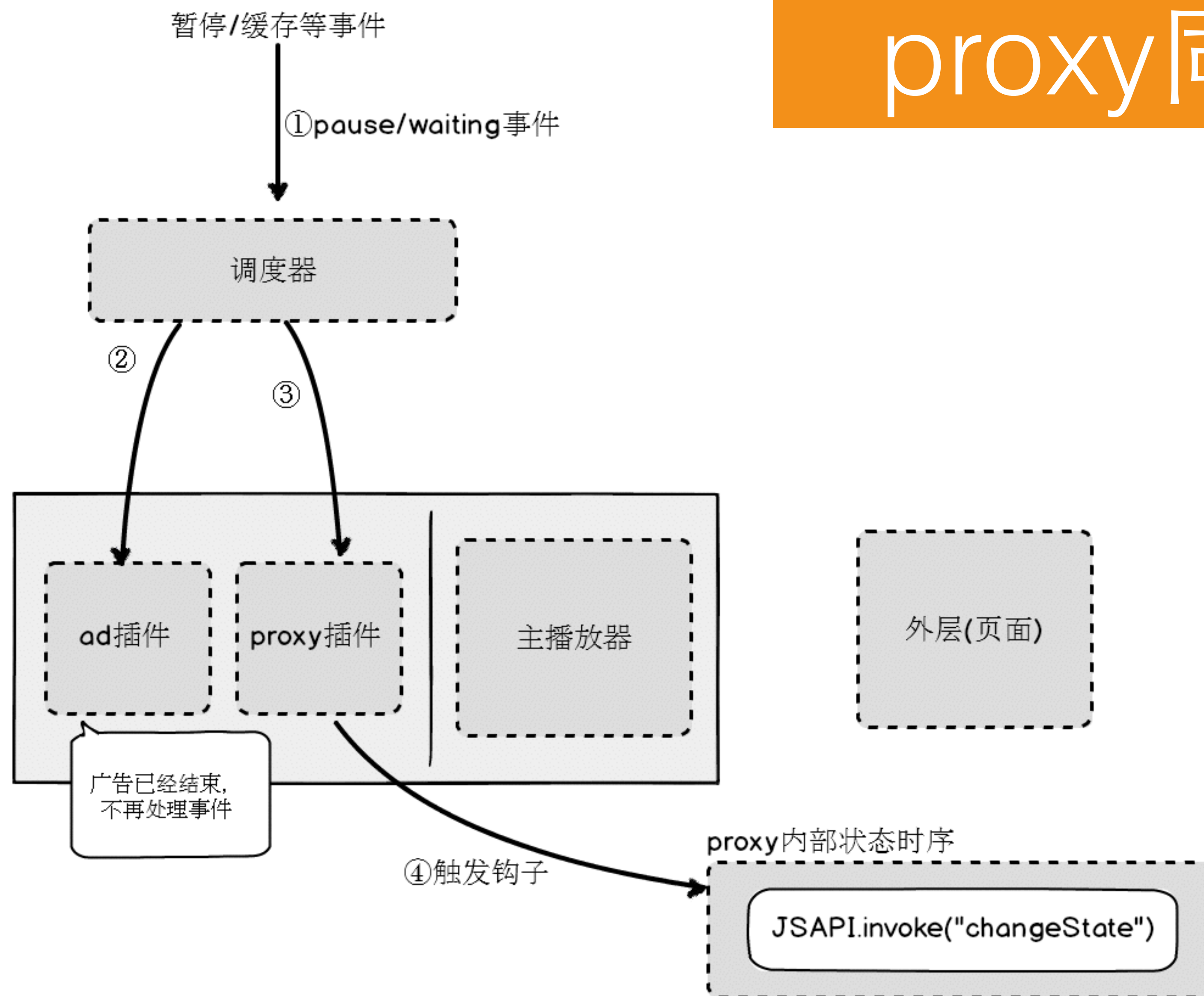
proxy换地址



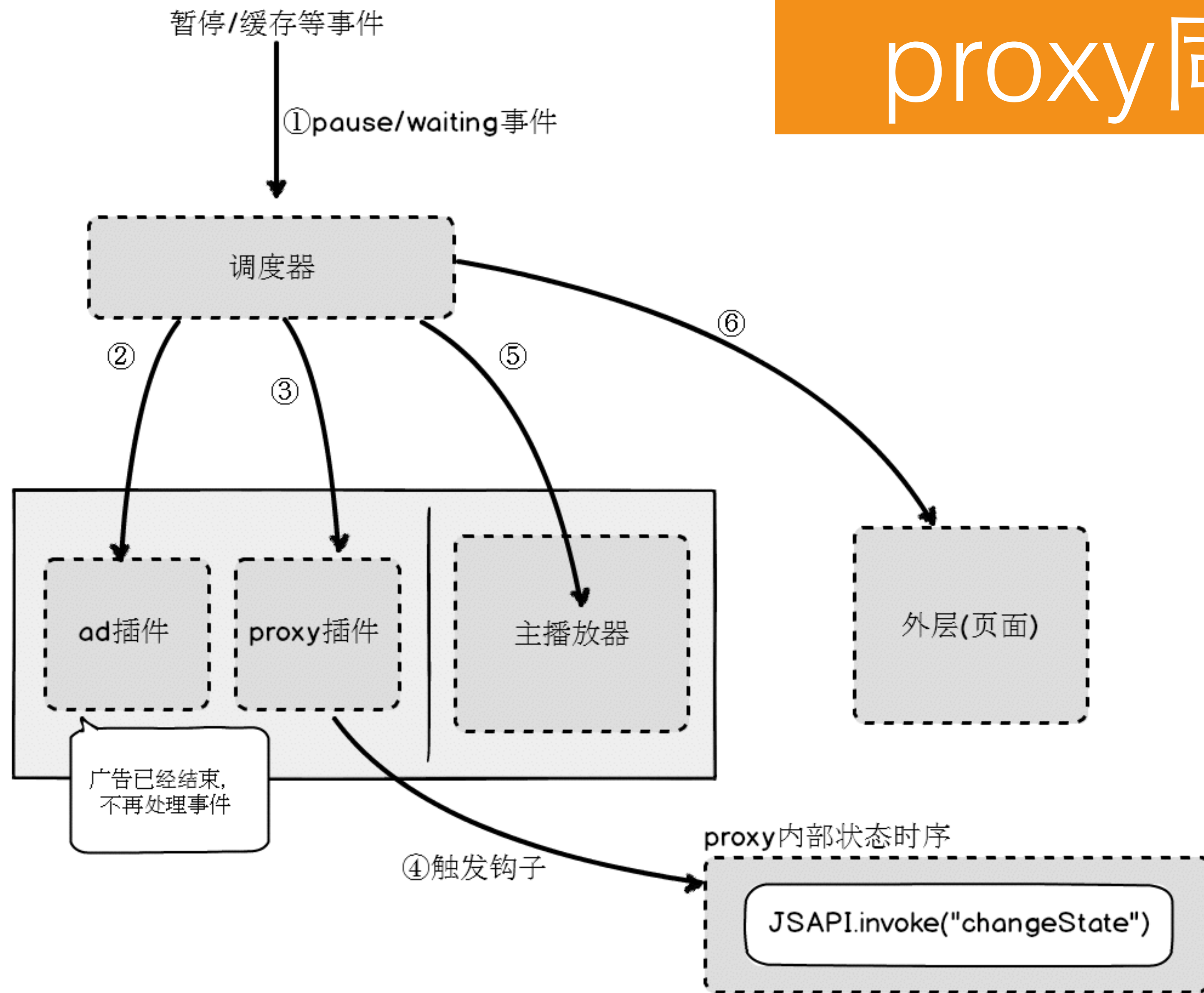
proxy同步状态



proxy同步状态

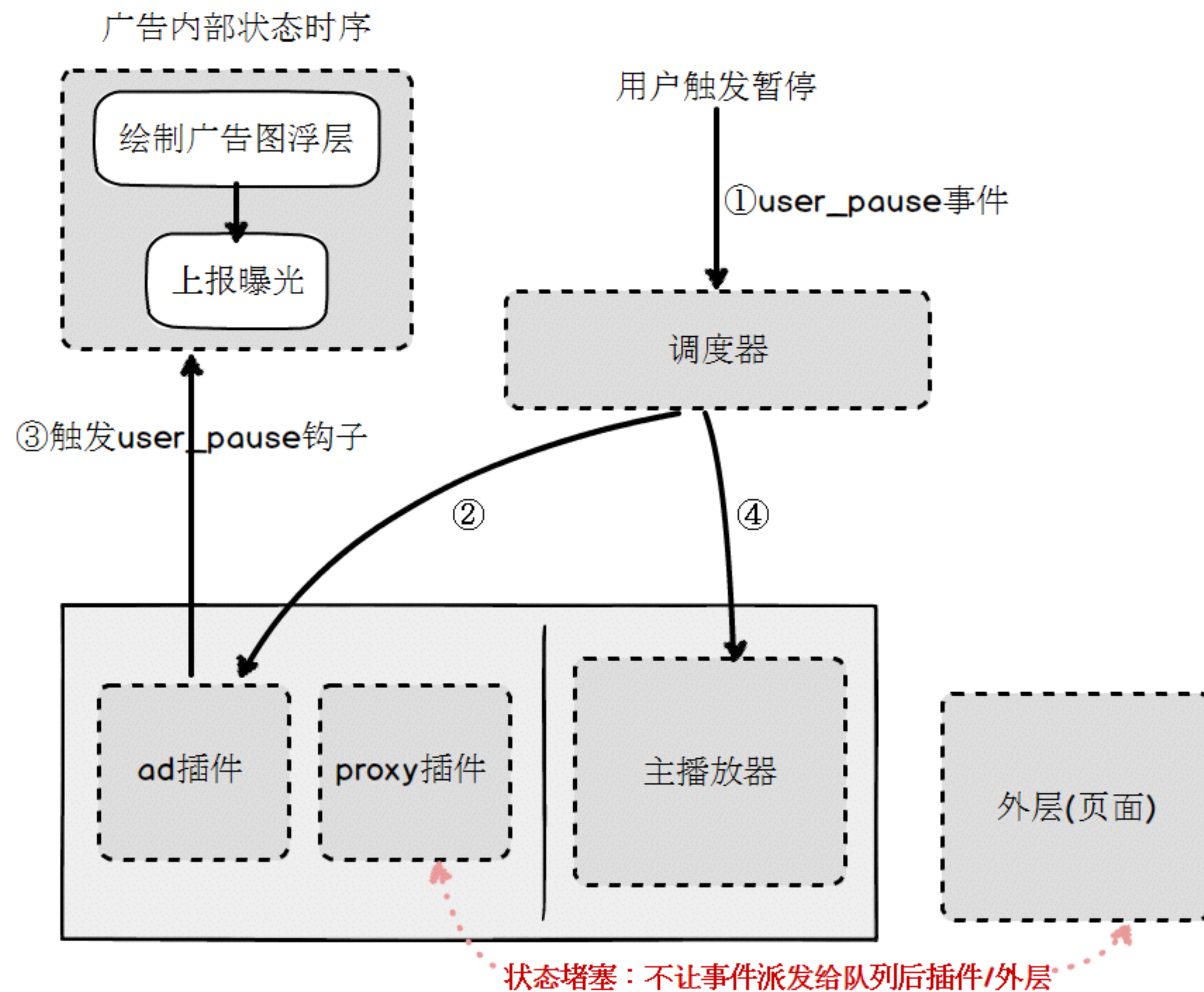


proxy同步状态

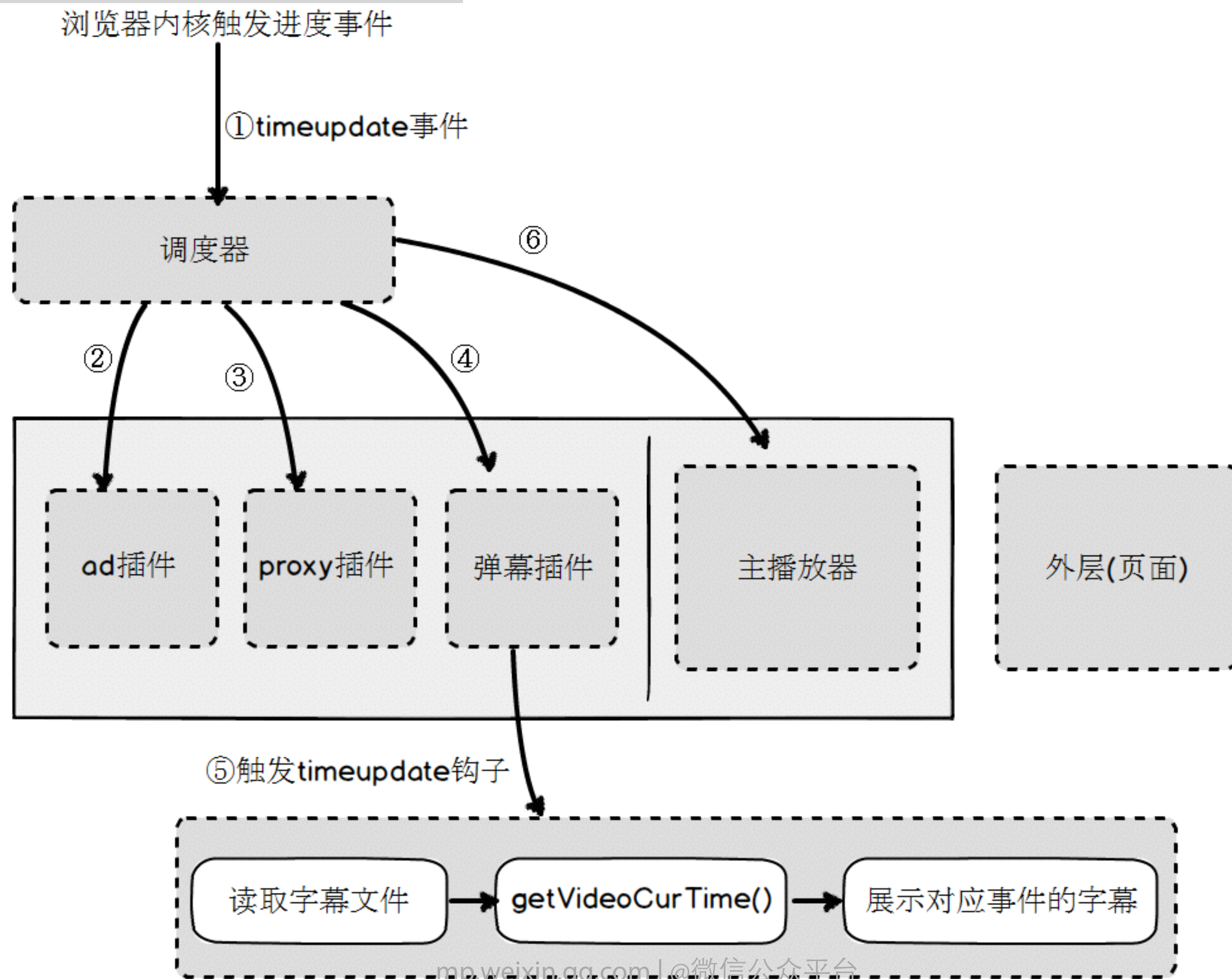


我们还可以完成更多可能的需求

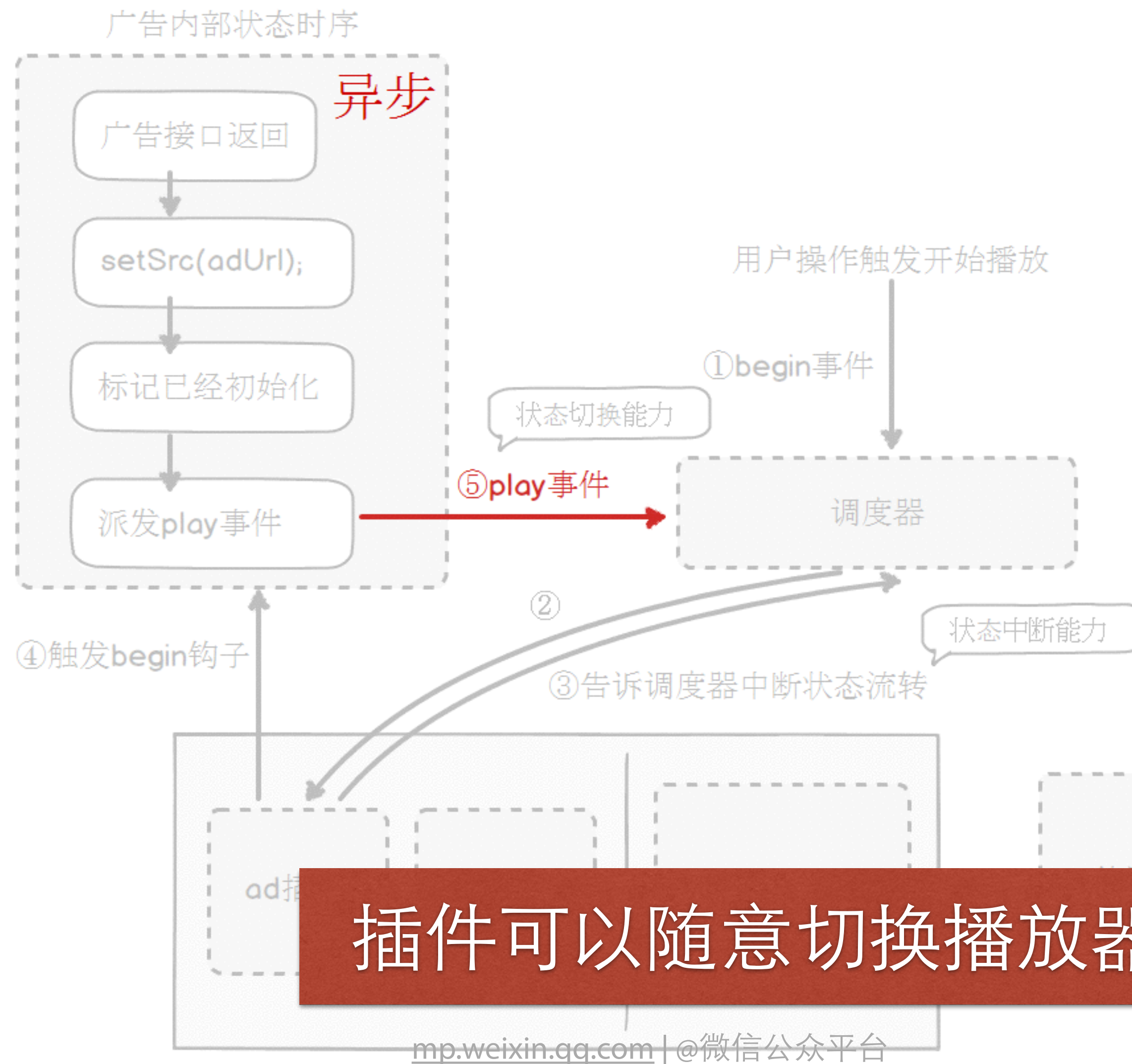
脑洞1：浮层广告



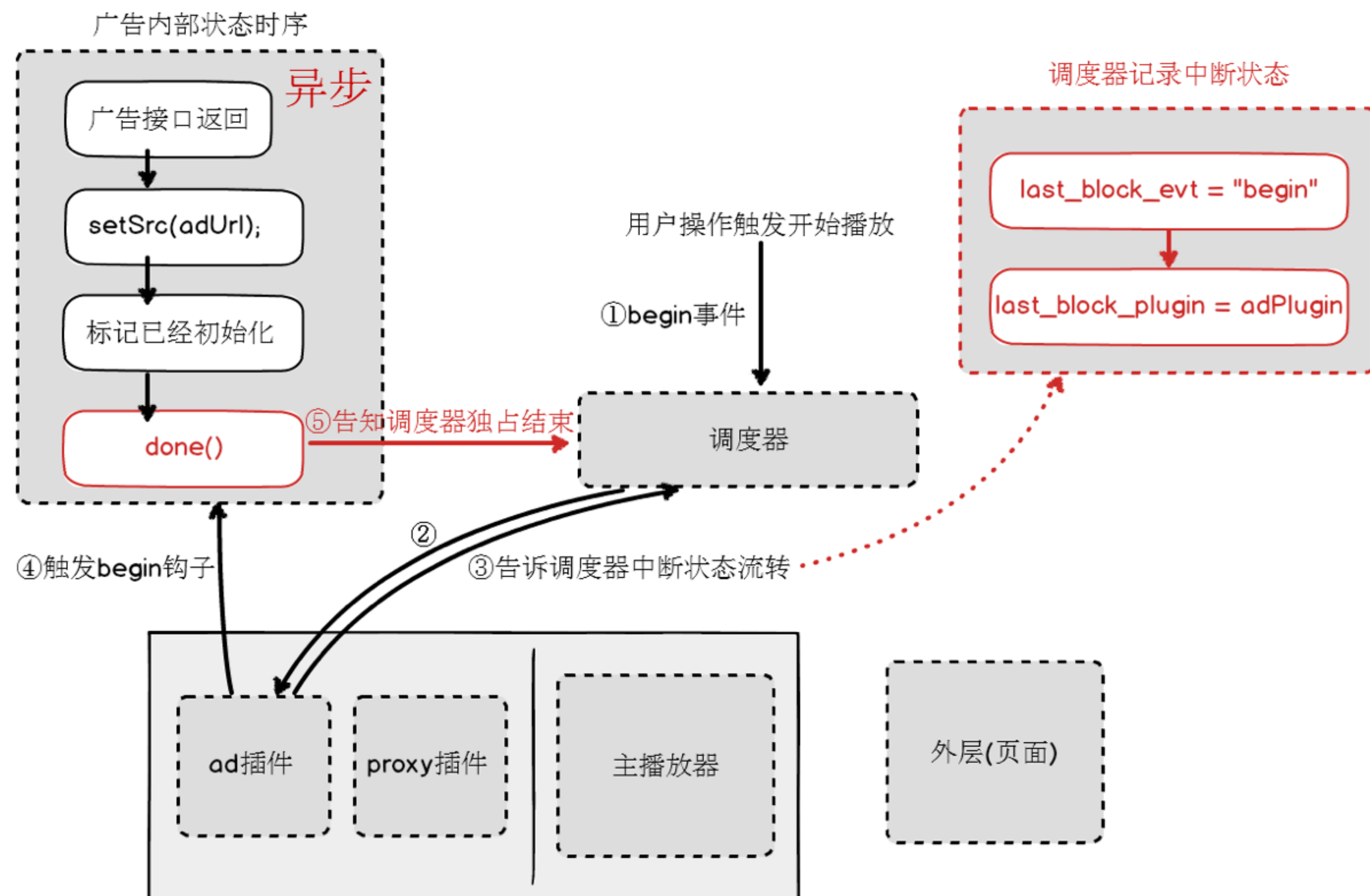
脑洞2：弹幕



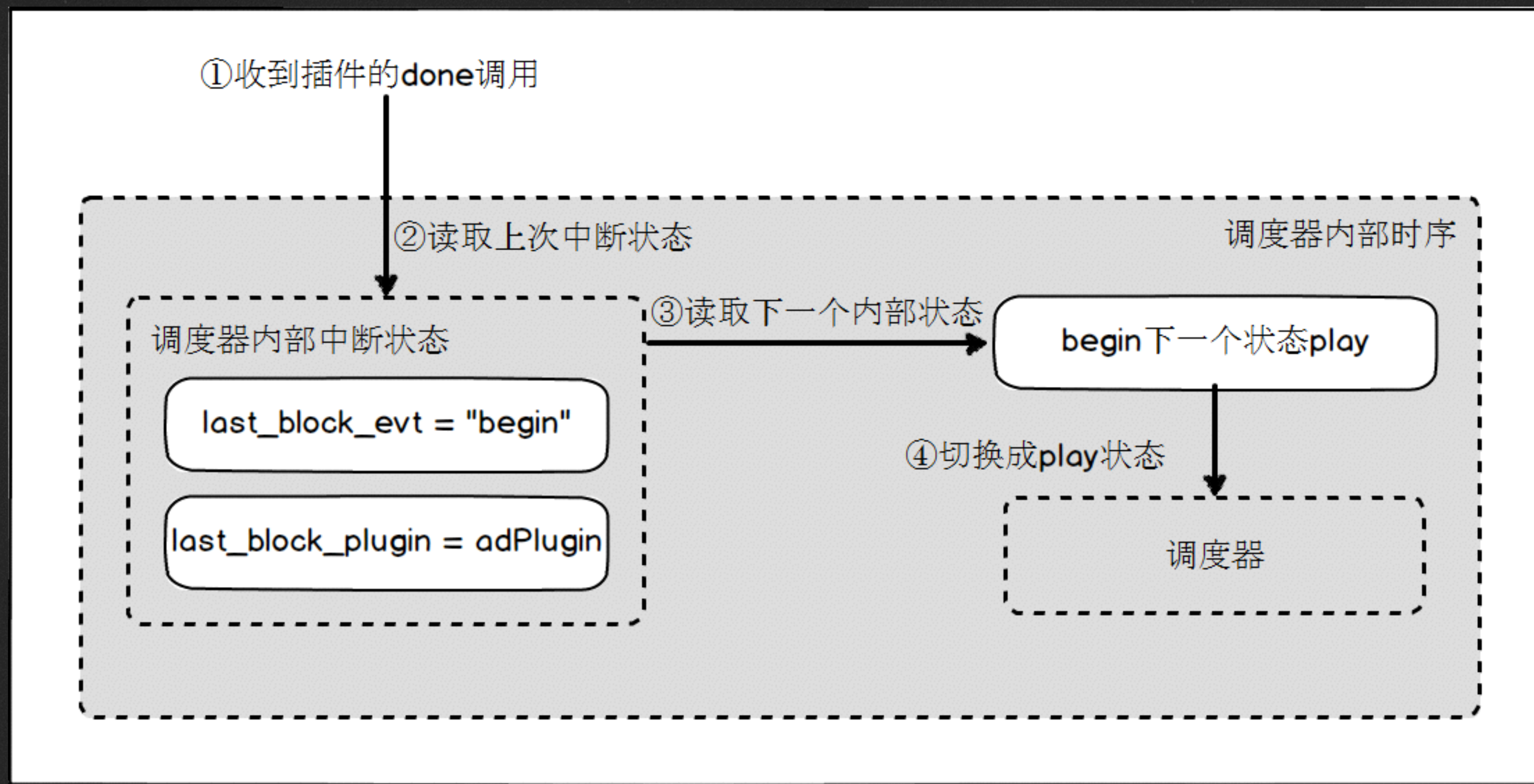
设计上存在小问题



切换能力收归到调度器



调度器内部切换状态



优点

- 可热插拔，可独立灰度插件
- 可装载多个插件
- 每个插件只关注它内部时序，开发极简
- 分开监控，可控发布
- 定位问题容易

合适的技术方案实现只是可控系统的其中一个环节

在发布之后，我们还需要日常运营保证系统的稳定性

Part2 日常运营如何保障系统的稳定性

难以排查的问题



白屏时间过长



排版乱了



图片加载不出来

无从查起，难以解决

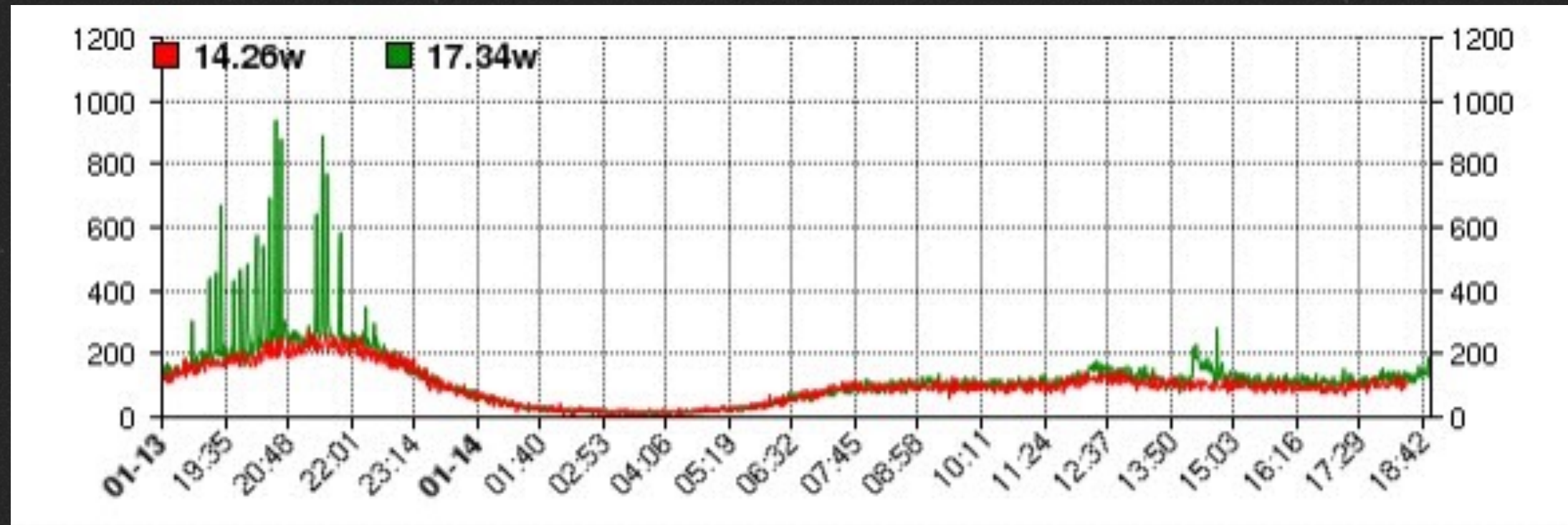
拉后台，请求肯定挂了，查一下

你们CDN挂了，拉不出CSS

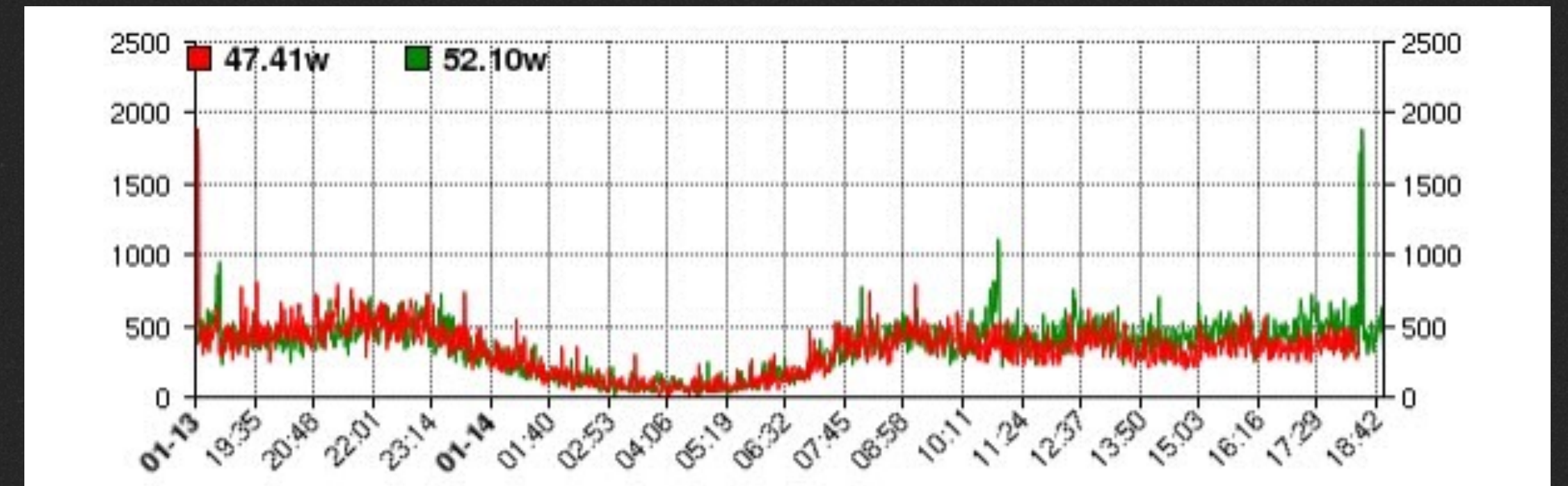
上线系统把文件搞错了，又故障

这问题在我这没重现，你手机可能有问题，重启看看

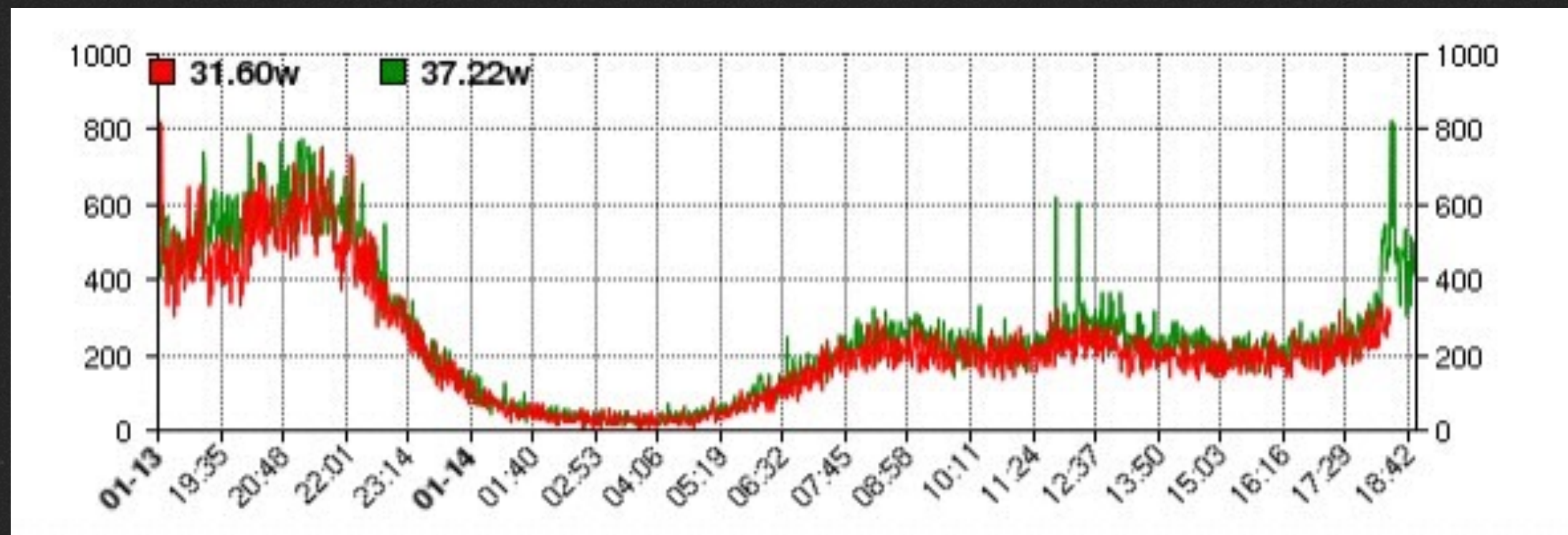
资源拉取的失败监控



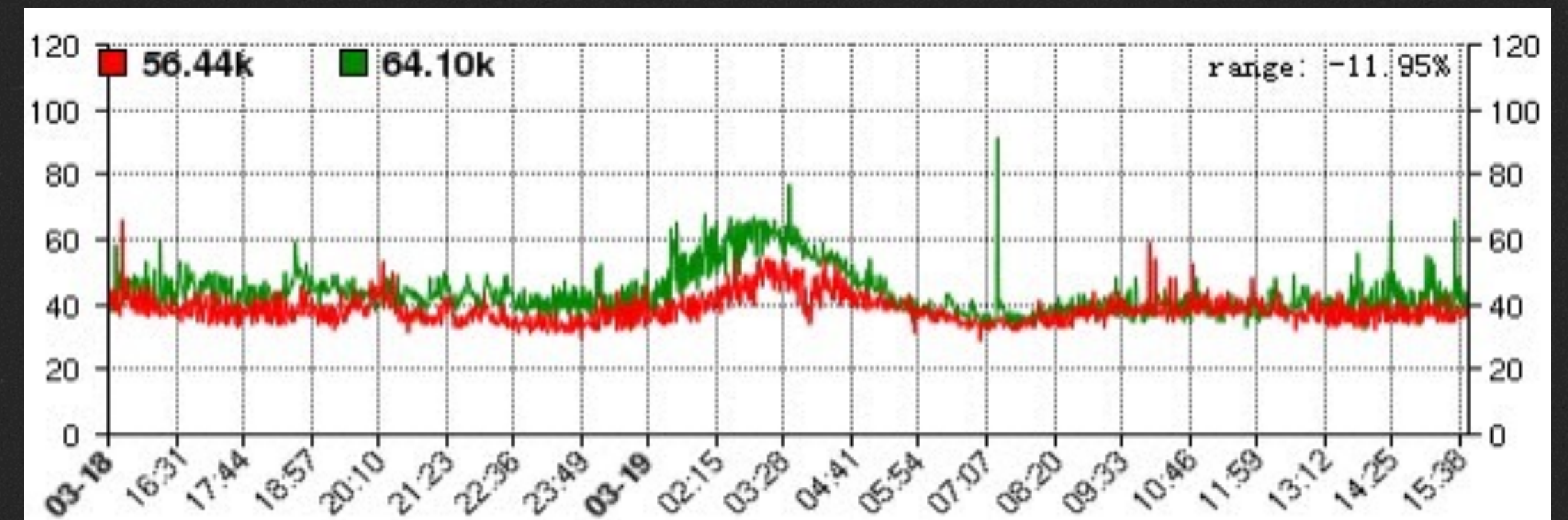
Ajax网络超时



CDN图片加载失败

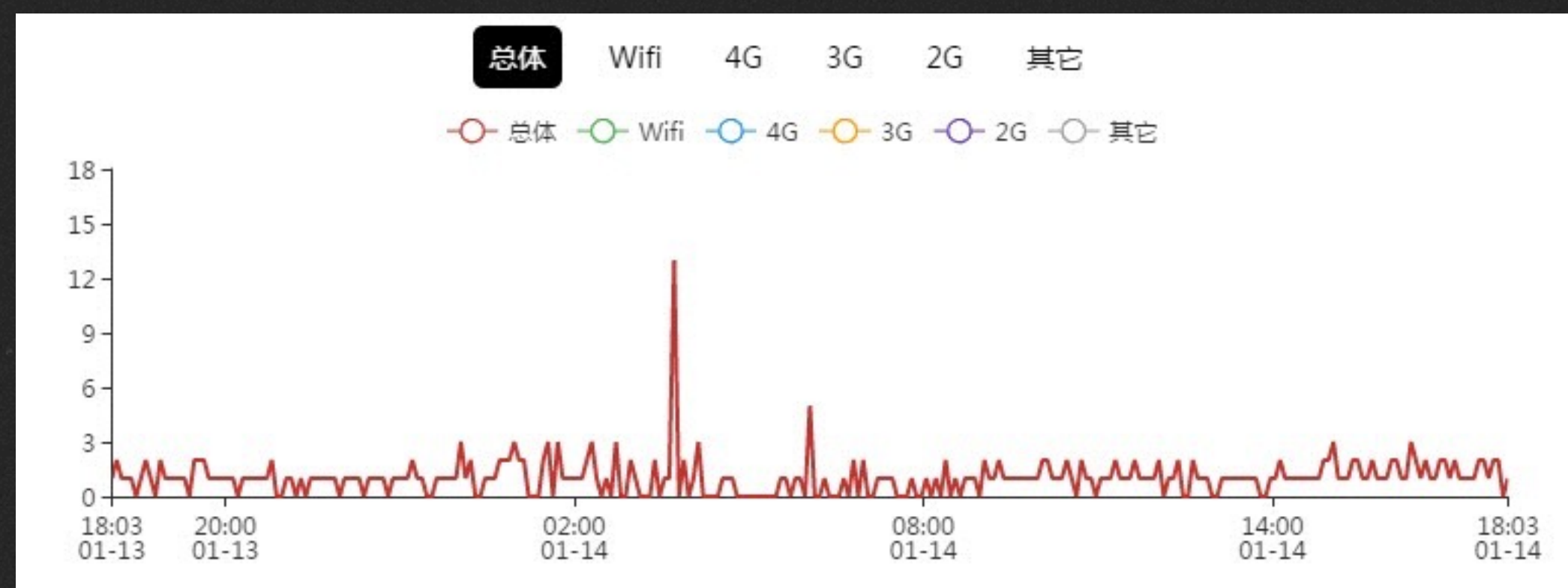


CDN静态资源拉取失败

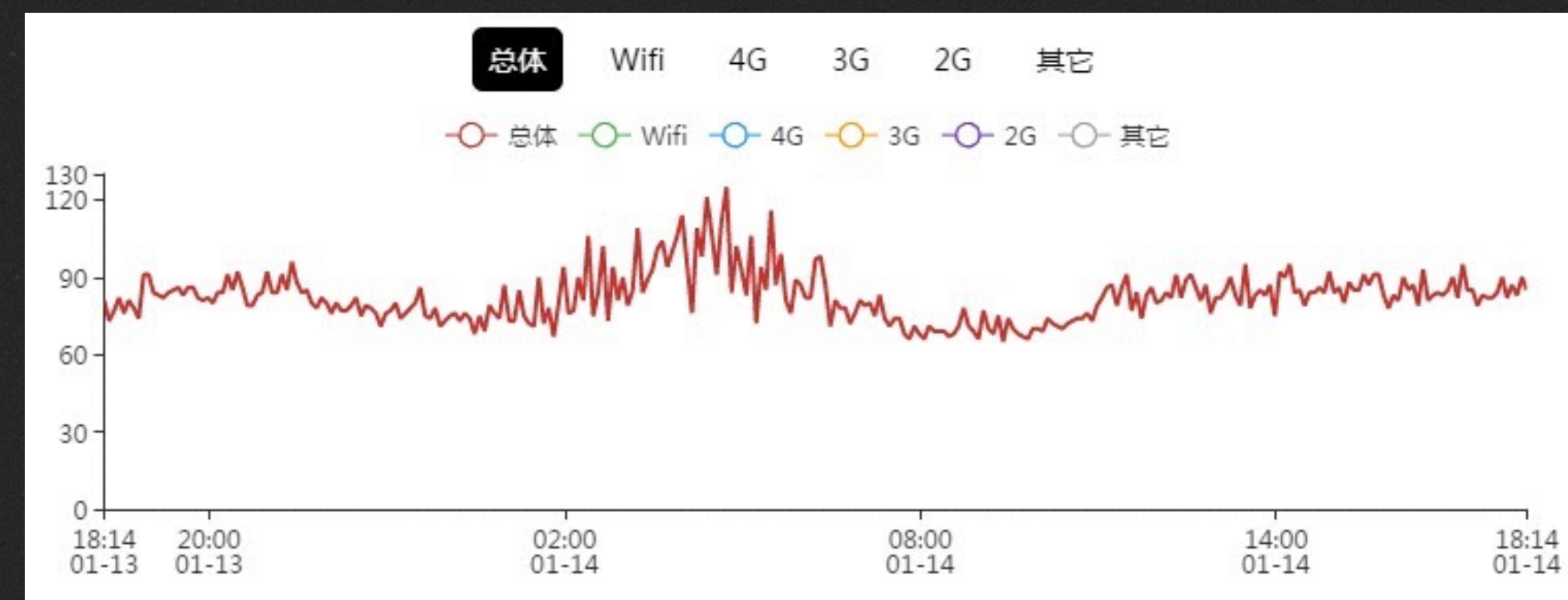


图片加载失败率

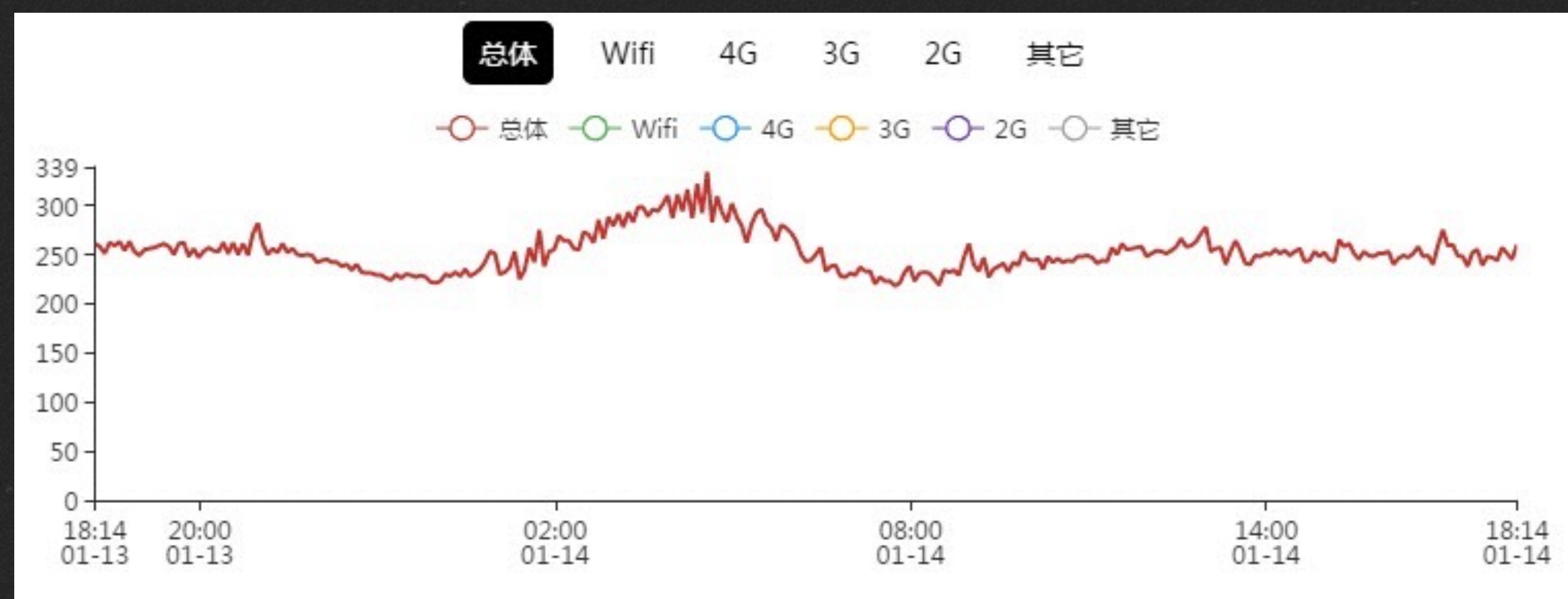
各个阶段耗时监控



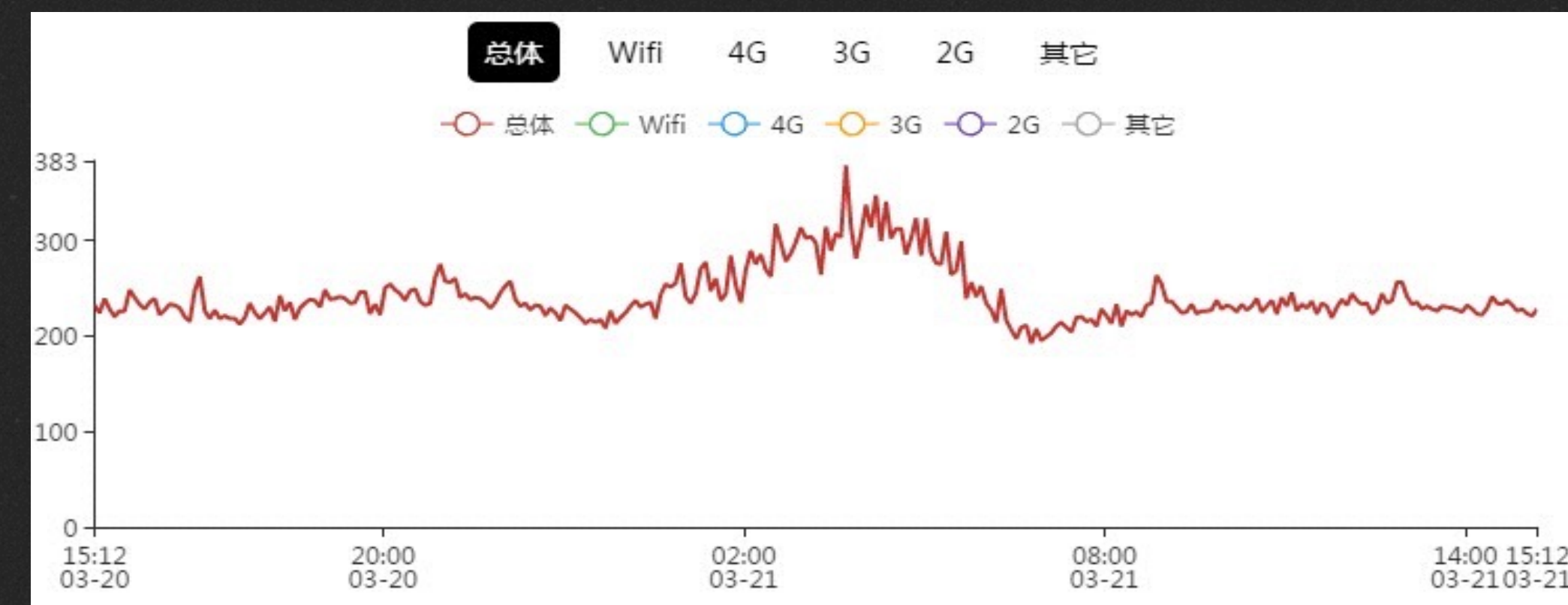
DNS耗时



收包耗时

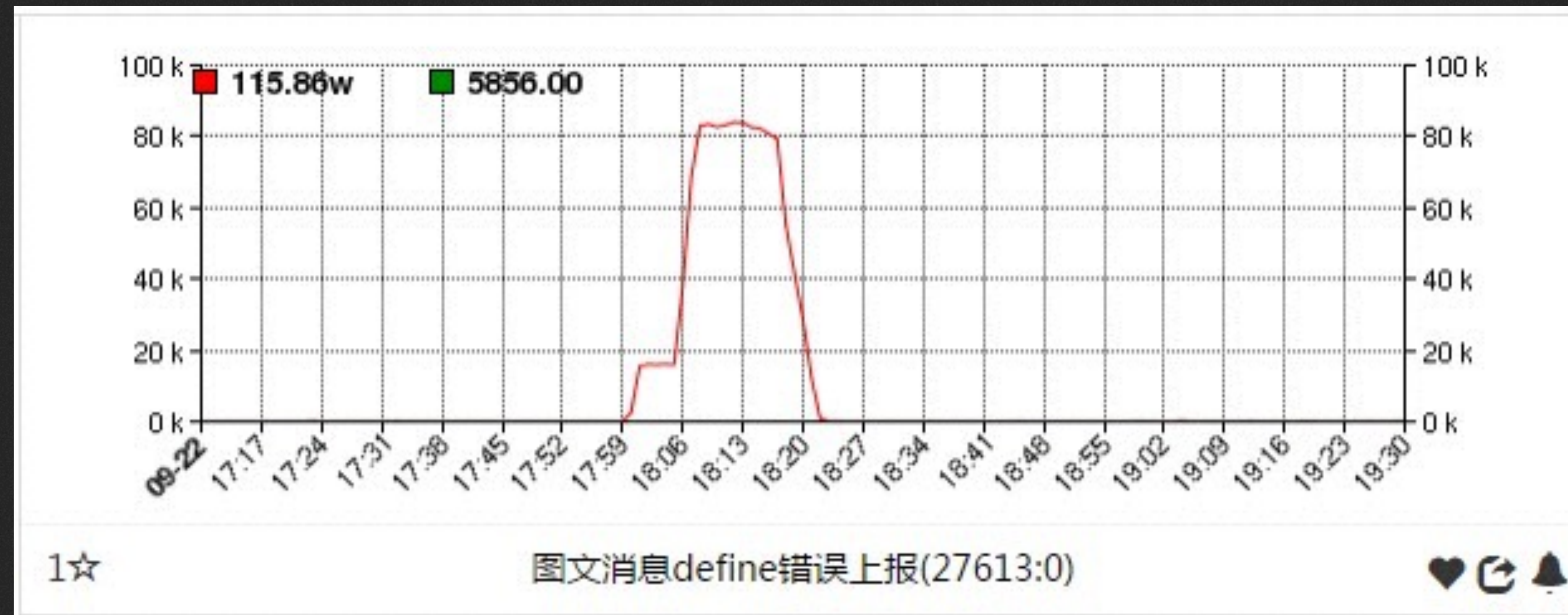


TCP连接耗时



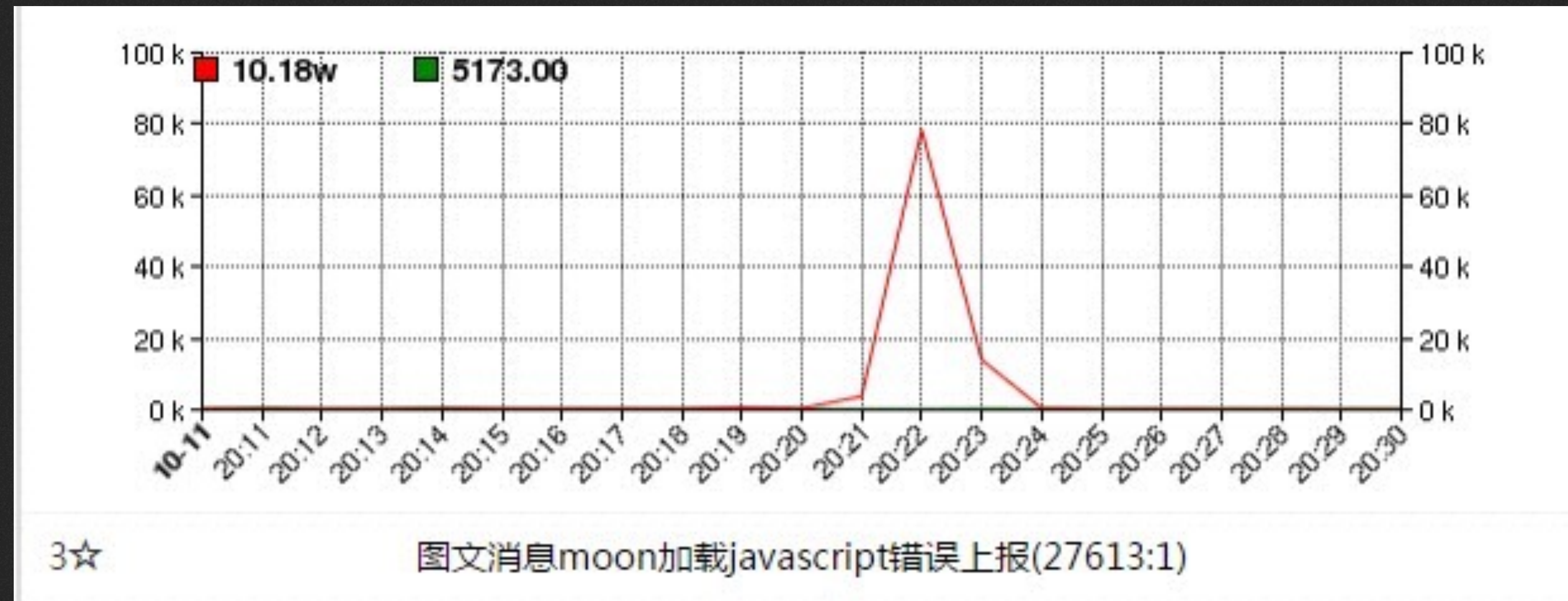
SSL连接耗时

监控脚本运行时异常



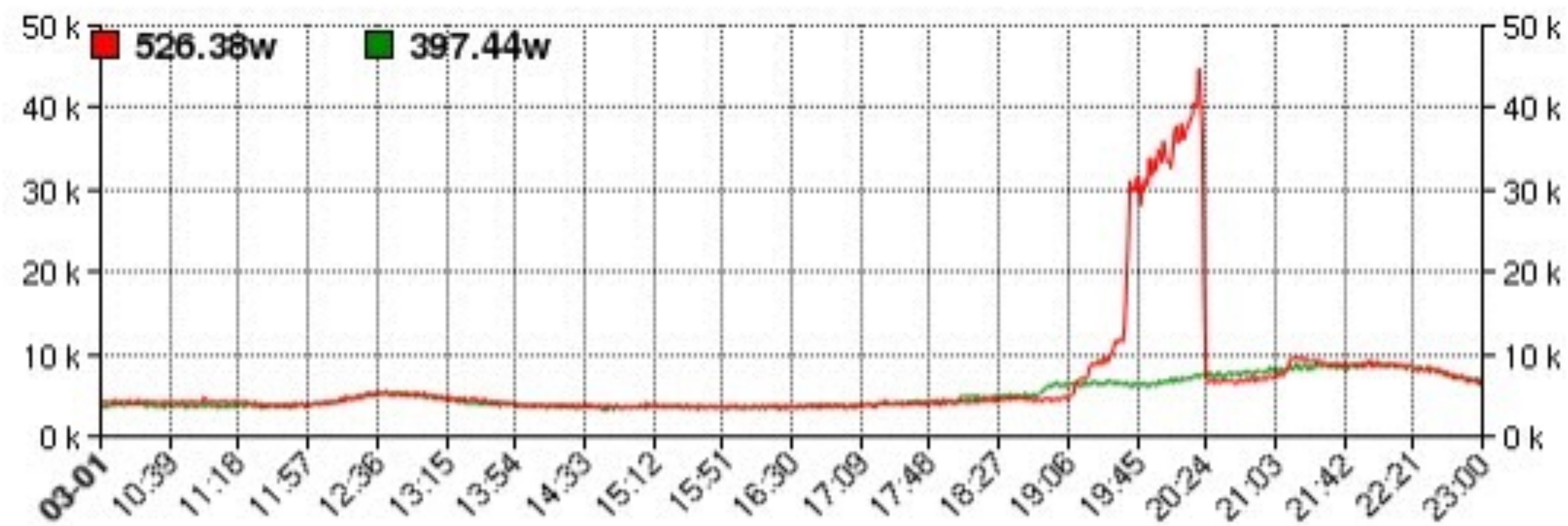
错误地引用空对象的方法

监控脚本运行时异常



资源回源机器没更新资源

监控脚本运行时异常



图文消息ajax network错误上报(27613:4)



后台某个CGI出现404

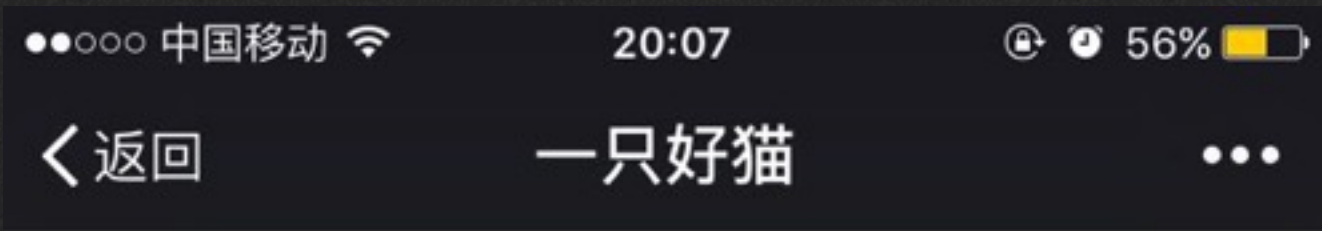
监控能帮助我们主动发现问题
但是面对海量用户，我们还是有很多难以定位的问题

棘手的问题



后台查看监控图表，服务正常

棘手的问题



我有一个美女推荐给单身的你，
欢迎转发

(原创) 2016-12-07 catfan 一只好猫

图片加载不出来

"又年底了，不想再一个人吃饭逛街打游戏旅游以及深夜失眠了，即使再优秀，也心累。如果你也跟我有一样的想法，就走出这一步吧。我是什么样的女孩？个性来说，爱笑无厘头，容易被感动也有自己的想法，会多方角度去替别人着想，会去哄人，是周围众多朋友的活宝，同时喜欢冒险，有赌性有个性。

人会不会闷？肯定不会啦，日常休闲活动，游泳玩狼人杀打王者荣耀麻将德州看电影看书听音乐

来源：一只好猫

一只好猫

一只好猫

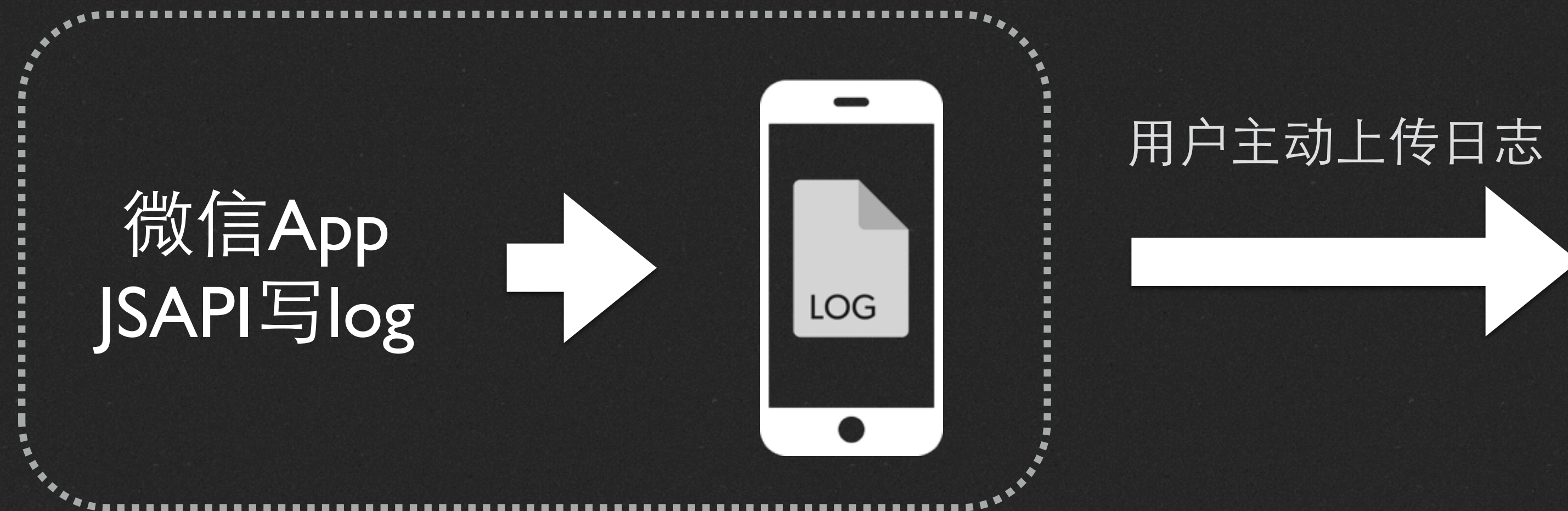
CDN图片服务正常

我们需要有能力恢复用户的现场

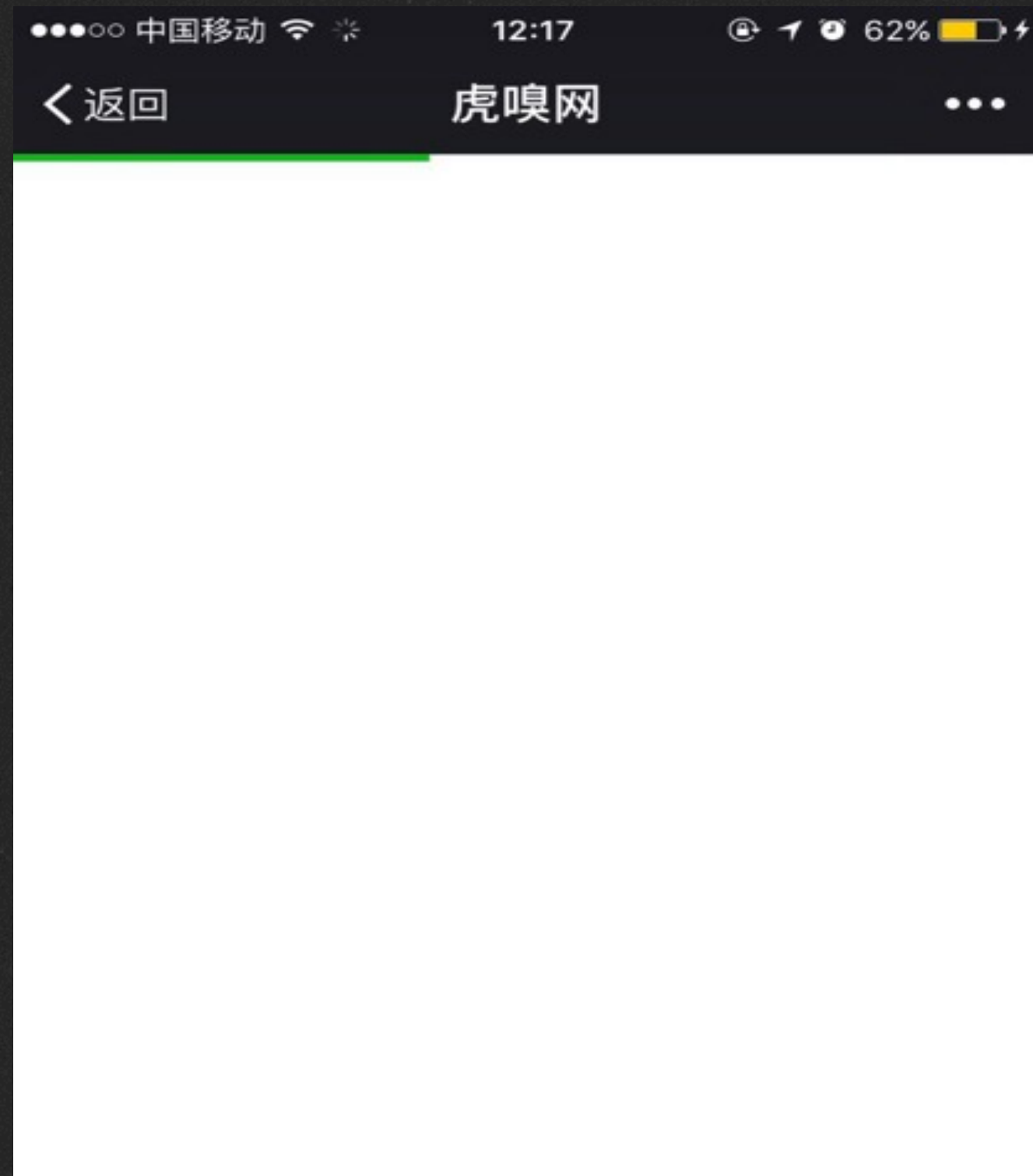
在客户端侧记录现场

1. 日志系统

把前端运行日志记录在微信客户端log文件，通过用户主动上传日志分析现场



白屏过久



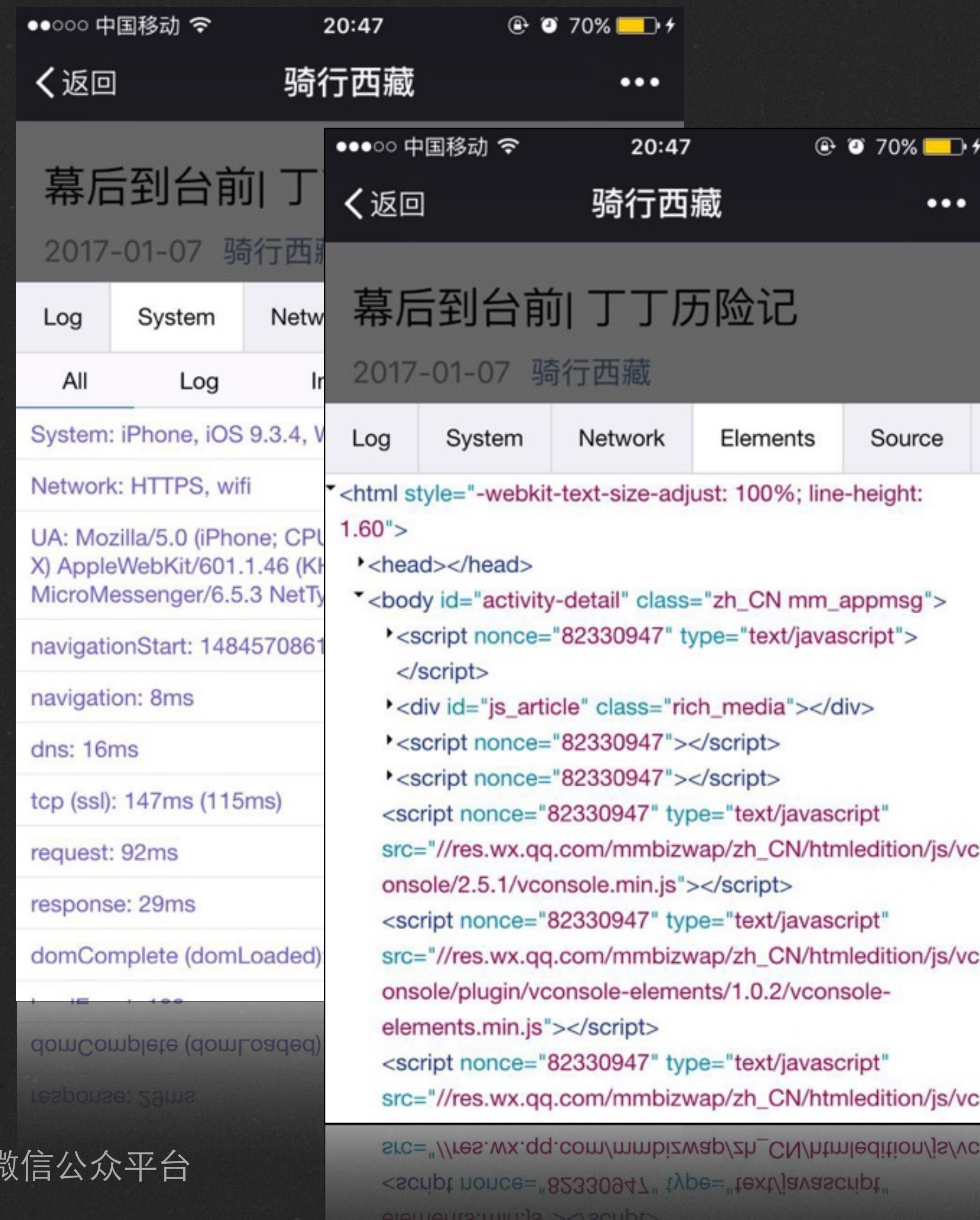
```
124067
124068
124069
124070
124071     level = info;
124072     msg = "[WechatFe][Appmsg]·dns:2863^^^ ssl:100^^^ tcp:139^^^ request:90^^^ ge
firstViewTime:3325^^^ interactiveTime:3302";
124073 } callback 1005
124074
```

日志里边发现DNS解析过久，发现某个小运营商路由问题

在客户端侧展示现场

2.手机端调试面板vConsole

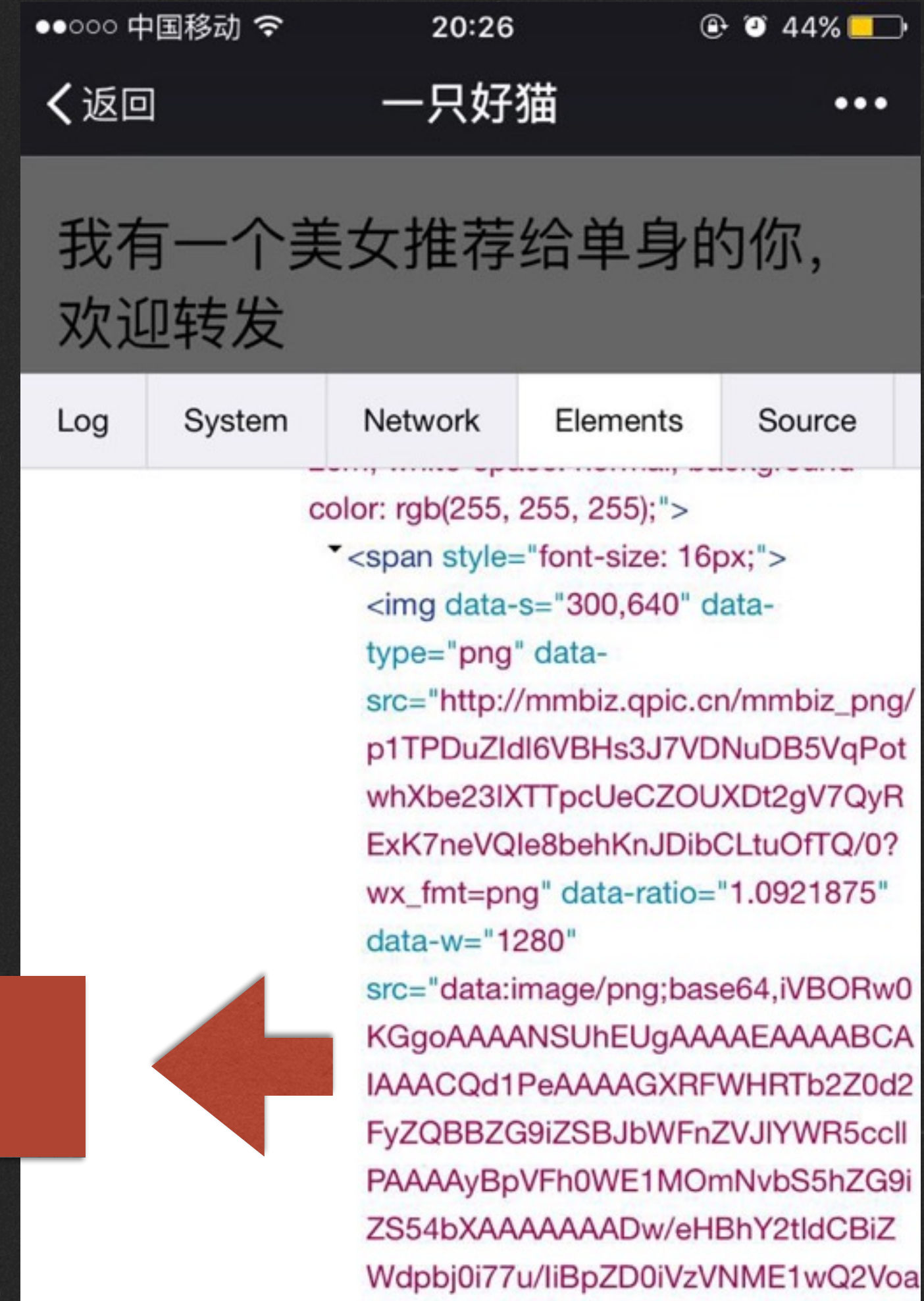
由于手机端要远程调试网页十分困难，我们开发了调试面板，用于远程定位现场问题



图片无法显示

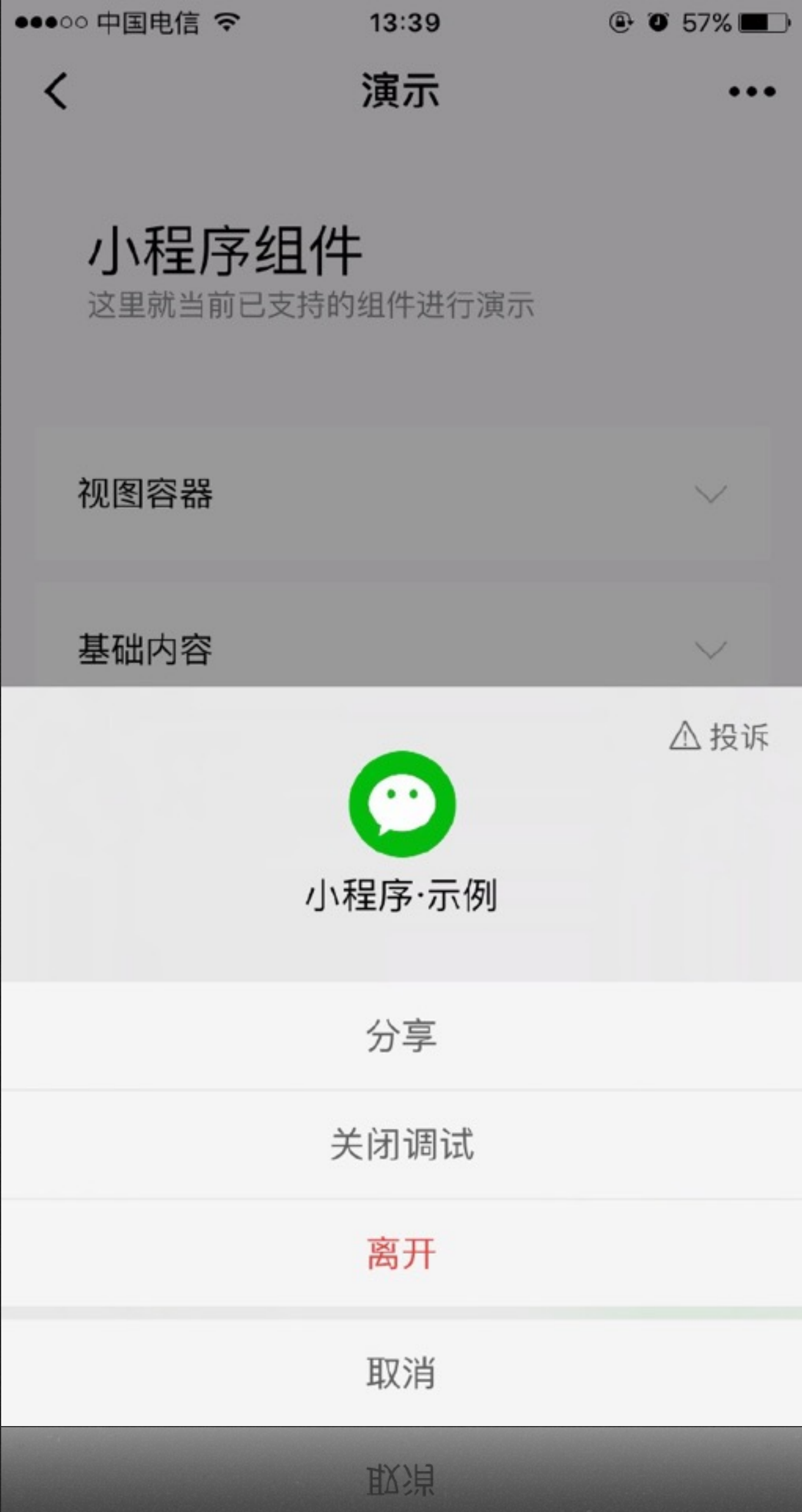


图片显示成一像素



vConsole开源

<https://github.com/wechatfe/vconsole>



小程序调试模式使用vConsole

总结

1.技术方案可控

通过灵活的技术方案使
得系统可控

2.监控和发现问题

能够通过各种手段发现
排查现网问题

END

Thanks for watching

微信开放平台基础部

郭志伟(raphealguo)

2017-04-16

