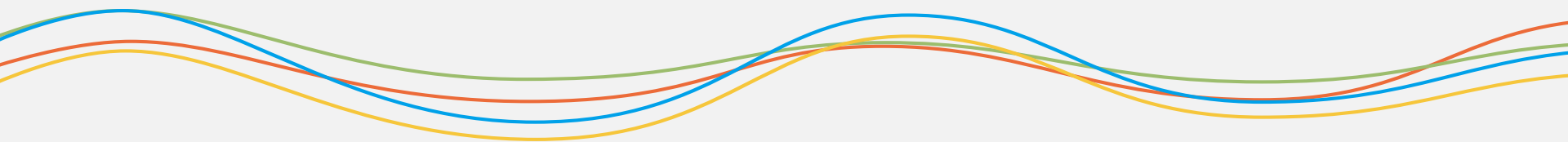
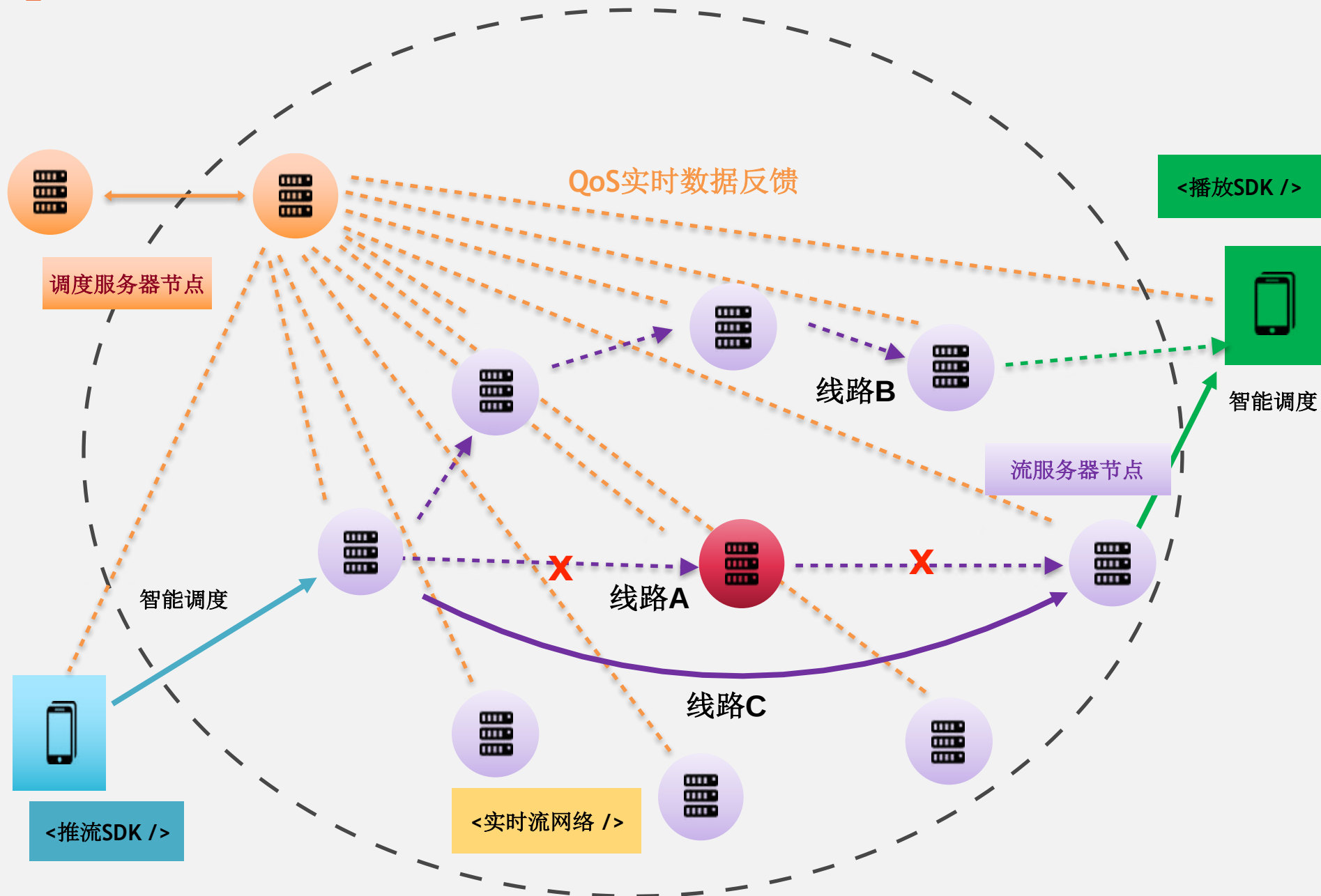


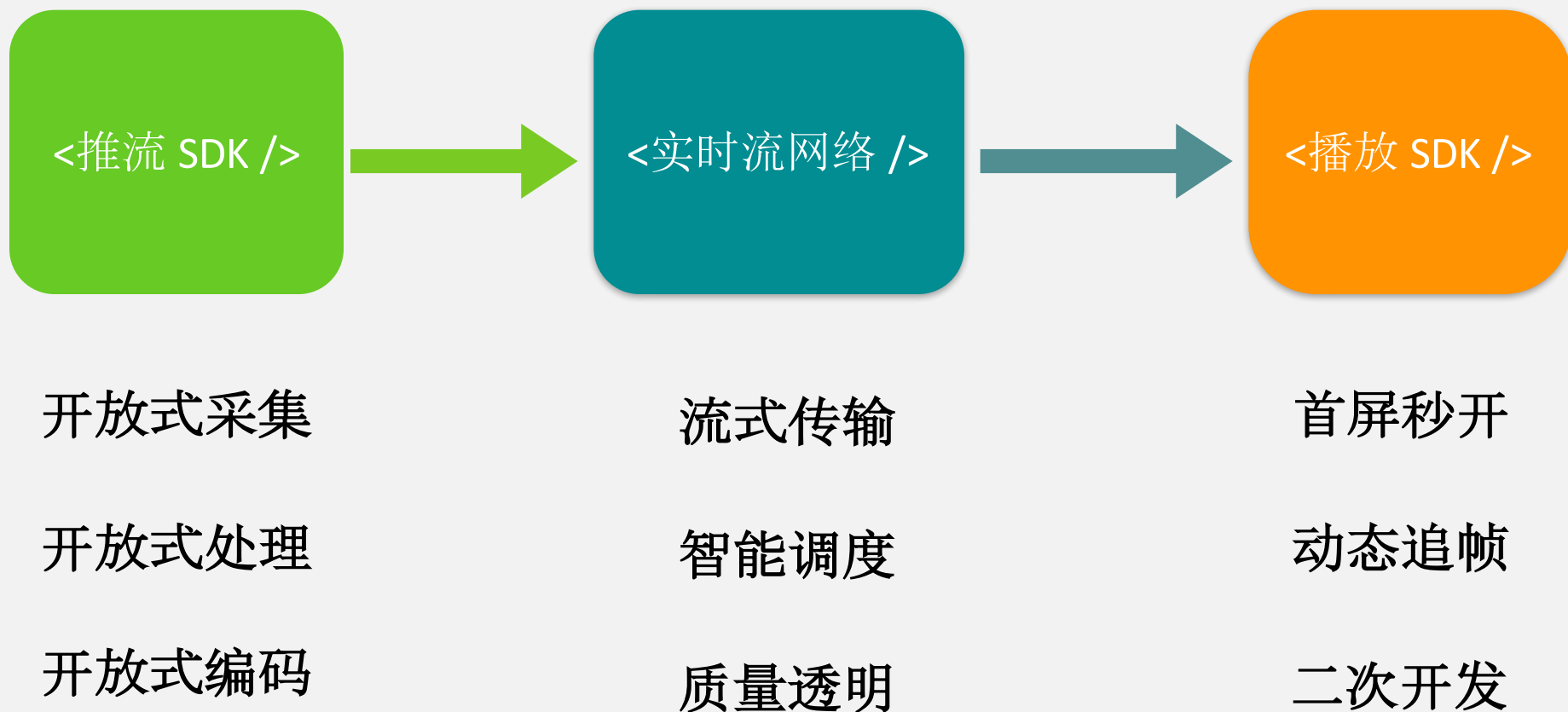
七牛直播云性能优化实践

何李石 @ikbear

helishi@qiniu.com







七牛直播云

推流SDK

播放SDK

直播云API

录制回放点播

转码及内容处理

统计报表

内容识别

推流及播放加速

对象存储

内容分发
网络

数据处理

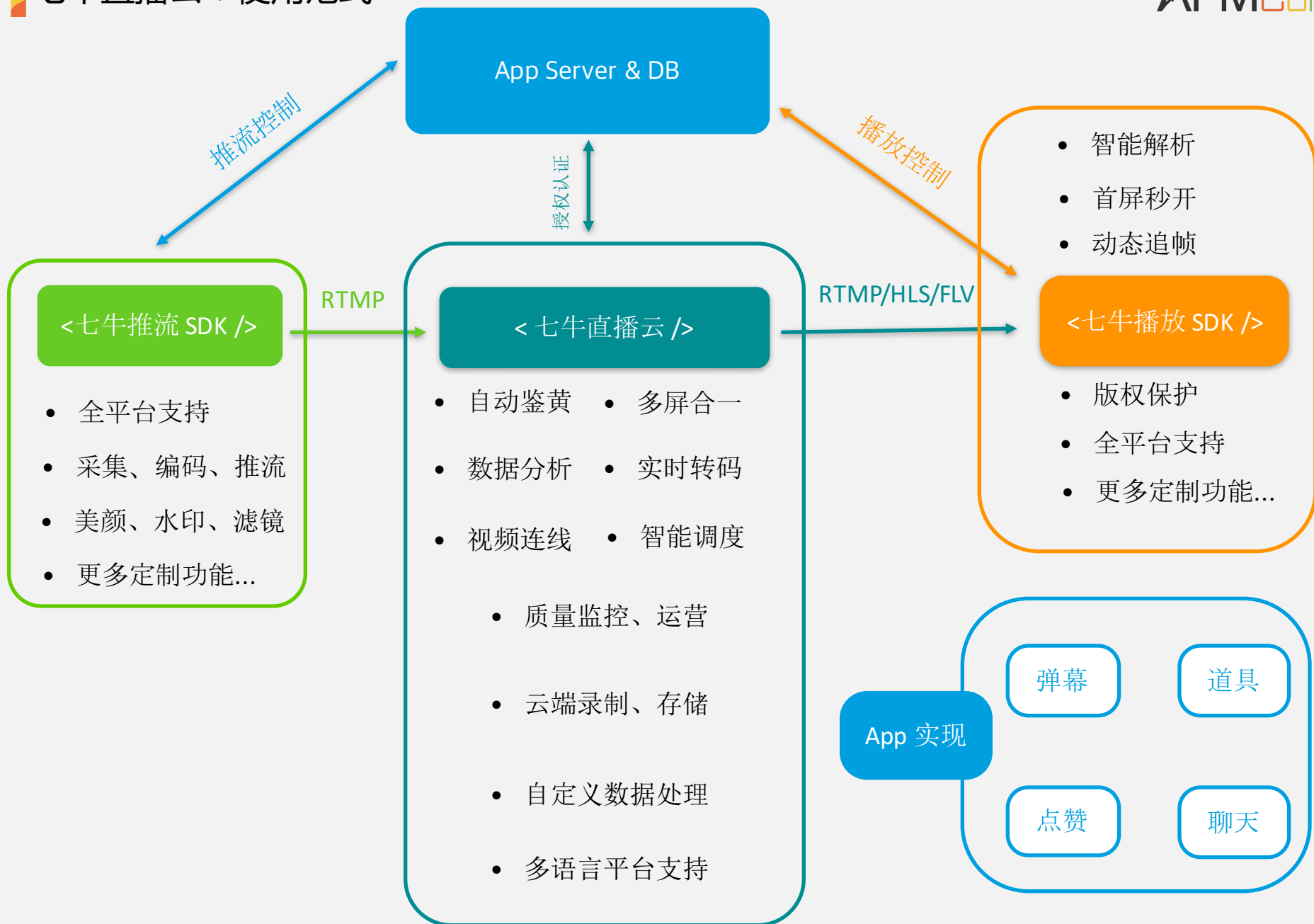
大数据

深度学习

通用计算

实时流网络

统一运营系统



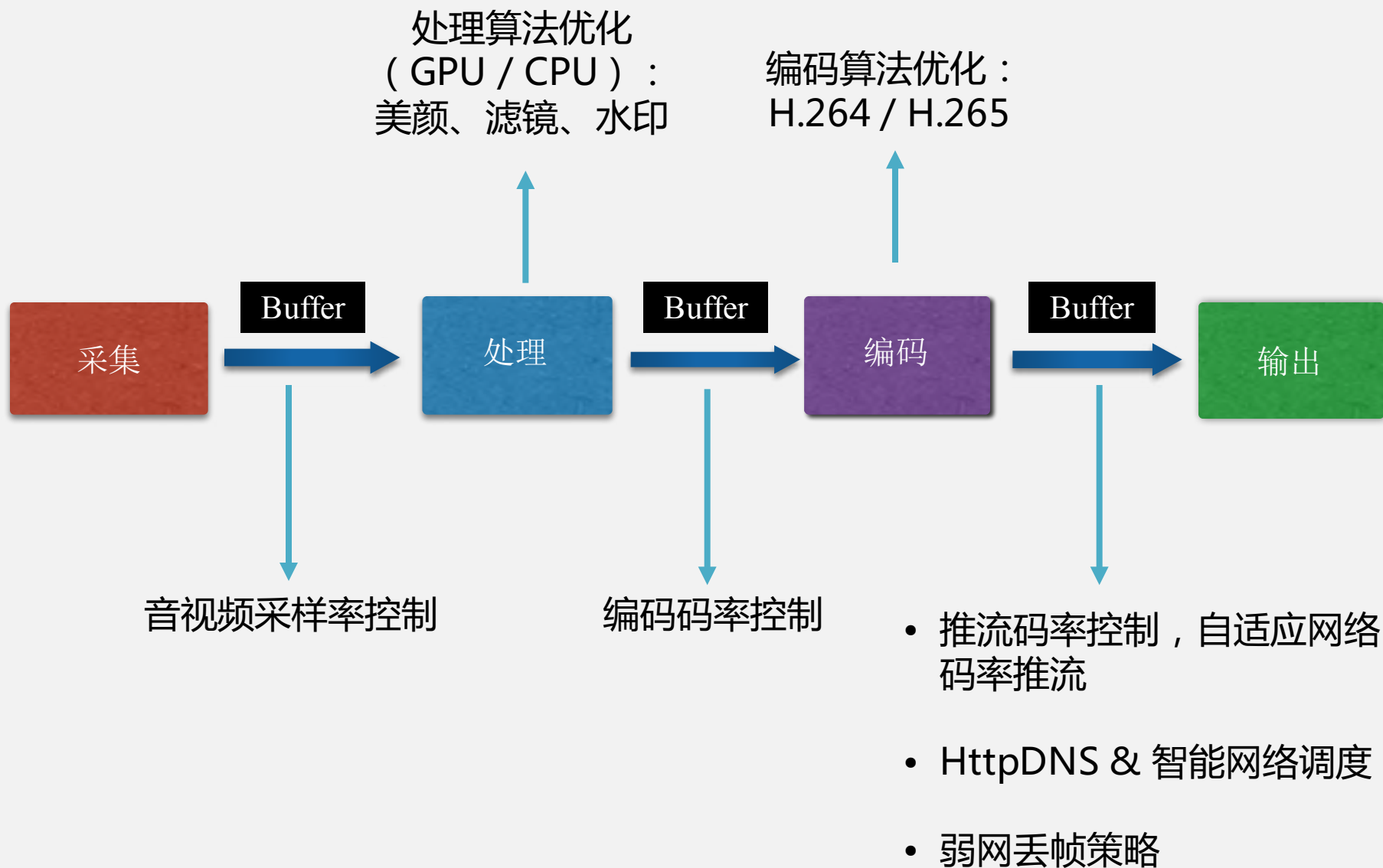
一、直播关键性能指标：

- 首开延迟：1s 以内
- 累计延迟：3s 以内
- 低卡顿率（卡顿时长/推流时长）：
 - FPS 不掉
 - 编解码性能匹配
- 高可用 SLA
- 低成本

二、直播挑战：

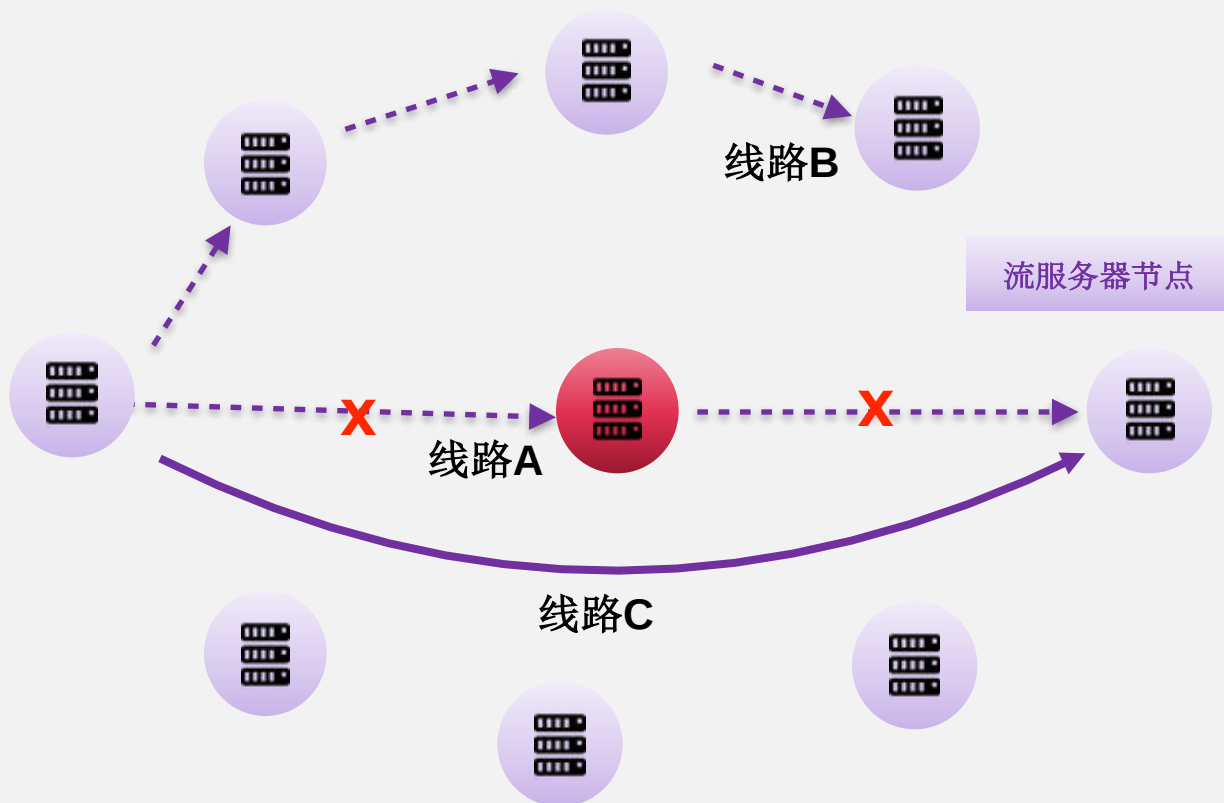
- 实时传输：传输协议和网络
- 海量终端用户：惊群效应
- 实时转存：海量存储
- 多屏多终端适配：
 - 实时转码（服务端）
 - 软硬编解码兼容

推流端



- 推流 QoS 模块
 - 实时监测（听云：监控调优 + 报警）：
 - 推流失败次数
 - 卡顿次数
 - 卡顿时长
 - 离线汇总数据：
 - 卡顿次数
 - 卡顿时长
 - 卡顿率（平均卡顿时长/平均推流时长）
 - 平均推流时长
 - 可用性
- 推流报障模块
 - App → SDK → Server → 分析报告

服务端

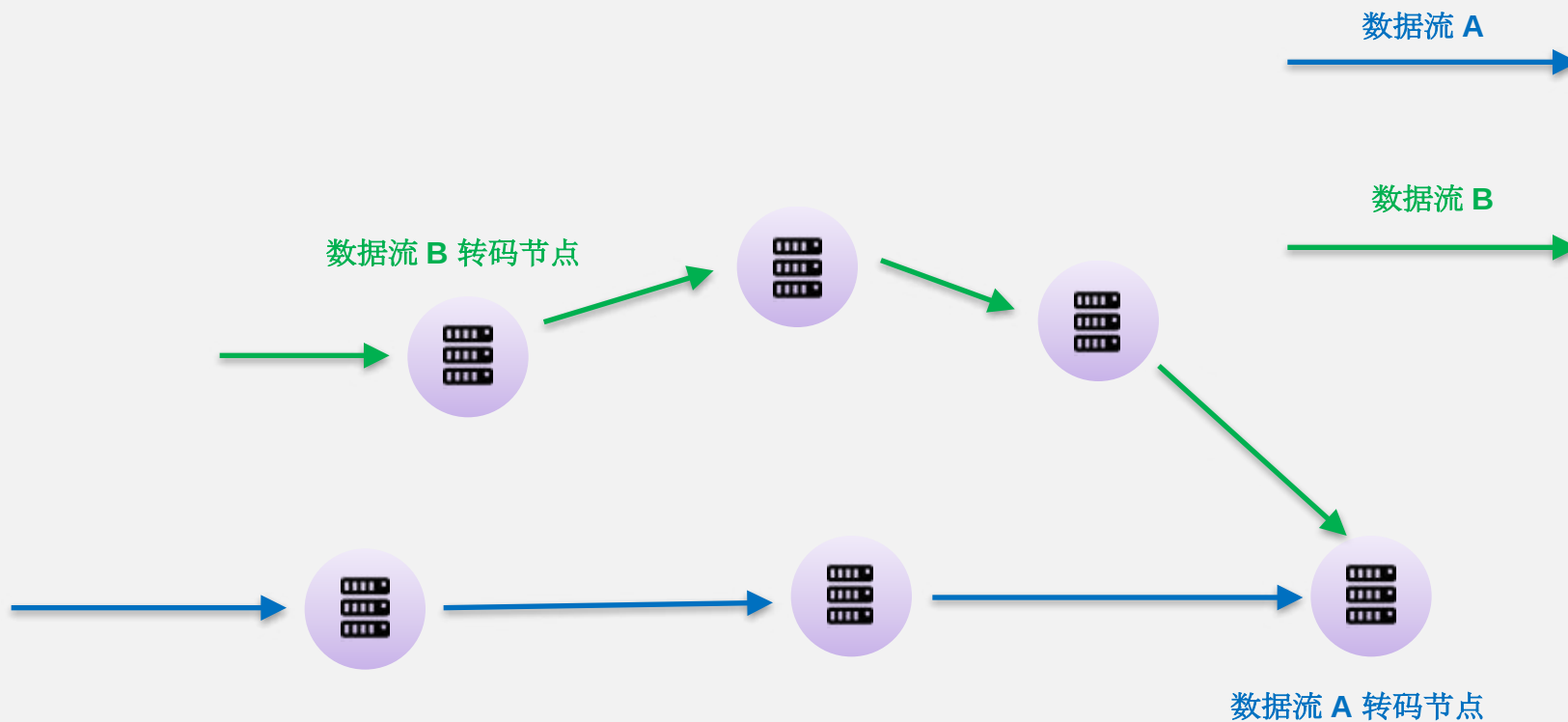


- 去中心化网络：
 - 全球覆盖，快速构建 PoP (Point of Presence) 节点
 - 边缘节点缓存分流：抵抗惊群效应
- 节点无状态化：高容错度
- 智能调度
 - 智能传输，全局最优路径
 - 智能运维，智能监控节点状态，快速上线新节点，实时下线故障节点，快速扩容缩容
 - eDNS 智能解析：地区 + 运营商 + 节点
 - 配合客户端 Query 测速调度
- 监测 (调度中心)
 - 节点保活探测
 - 节点连通性探测
 - 秒级帧率统计
 - 汇总 fps 数据：卡顿率计算 → 优化
 - 实时 fps 数据：播流链路质量监控

	全称	协议	原理	延时
RTMP	Real Time Messaging Protocol	长连接 TCP	每个时刻的数据，收到后立刻转发	1~3 秒
HLS	HTTP Live Streaming	短连接 HTTP	集合一段时间数据，生成 ts 切片文件，更新m3u8	> 10 秒
HTTP-FLV	RTMP over HTTP	长连接 HTTP	同RTMP，使用 HTTP协议	1~3 秒

	优点	缺点	适用场景
RTMP HTTP-FLV	低延时	跨平台差 Flash Player 以外的 平台都需要做移植	即时，有互动需求
HLS	跨平台 可点播回放	高延时 多次请求，网络质量 影响大	单向广播

定制传输协议？



- 实时转存储：靠近存储，集中式转码
- 转码调度：Docker 容器虚拟化 PaaS
 - 稳定性：转码流程相互隔离，互不影响
 - 智能调度：实时监控节点可用资源，实时调度转码任务，最大化资源利用率

RTMP： 原始码率 > 目标码率



靠近推流端转码

HLS： RTMP 效率 > HTTP 效率

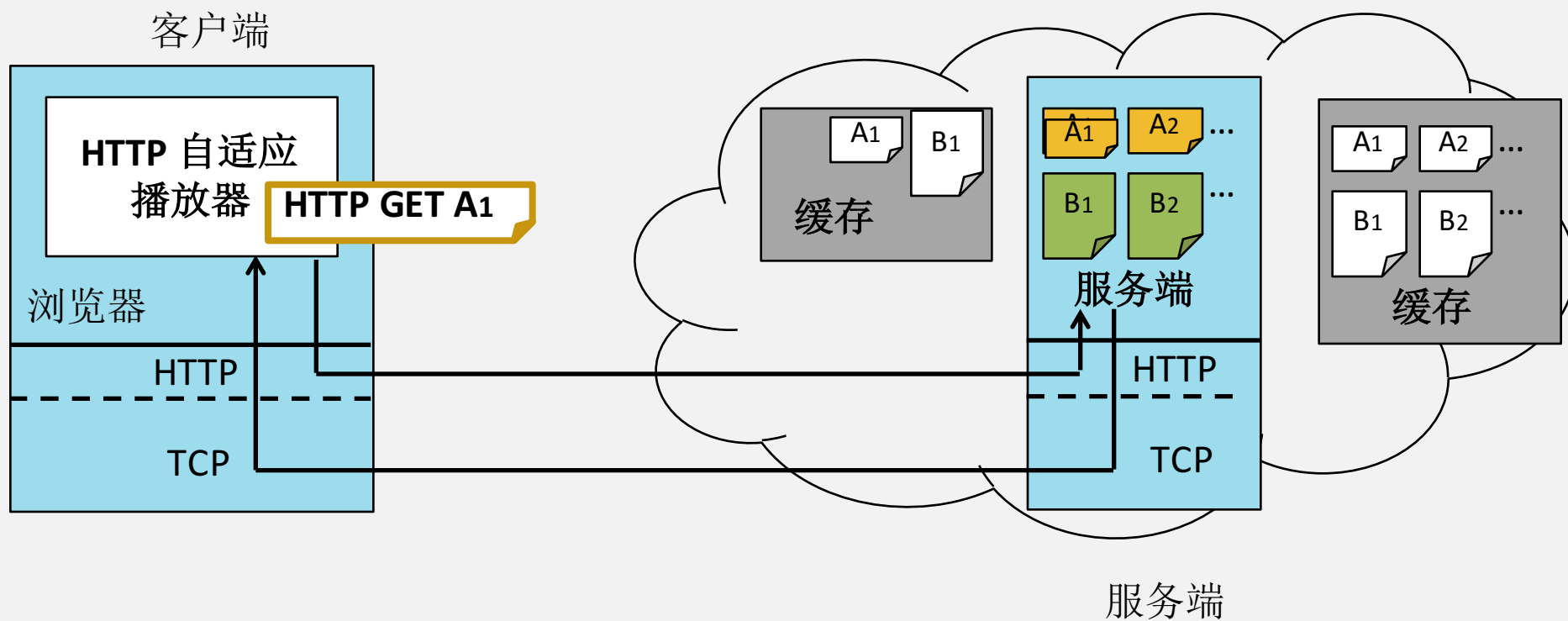


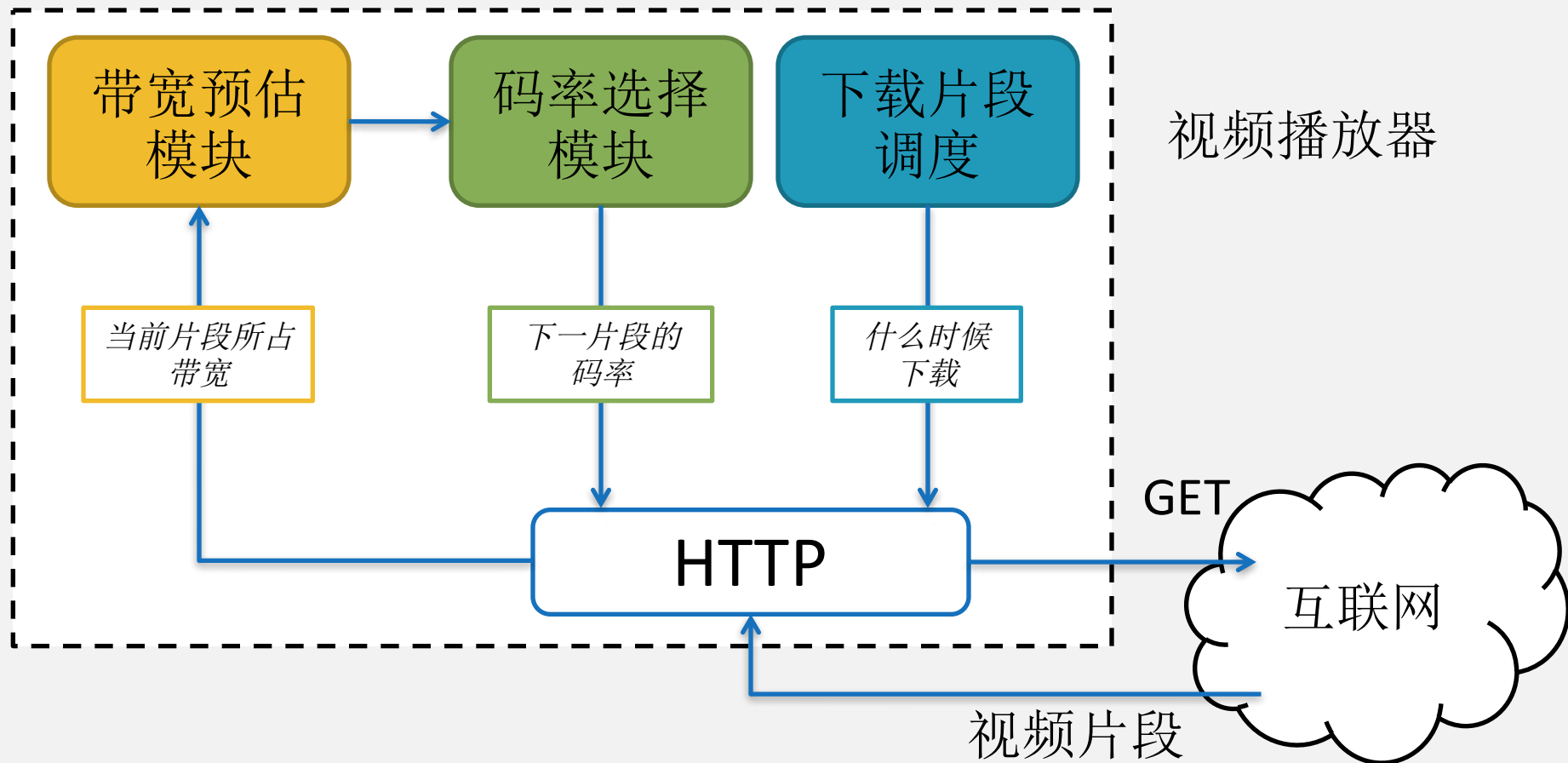
边缘切片，延迟
控制在 6-7s

播放端

- 首开延迟优化：
 - 服务端 GOP 缓存：0.5s 首开时间（Flash 播放器）
 - 播放器缓存大小优化：越小越好，尽快播放解码
- 累计延迟消除优化：网络抖动
 - 播放器缓存大小优化：与首开延迟优化相反，越大越好
 - 实时丢帧（非关键帧）
 - 去掉 B 帧，加快解码
- 卡顿率优化：自适应码率
 - 带宽不足情况下自动降低码率，减少网络延迟带来的卡顿感

A2 第二块码率为 A 的视频片段





1. 三大模块
2. 播放器和外部网络之间通过反馈实时调整码率

- 播流 QoS 模块
 - 实时监测（听云：监控调优 + 报警）：
 - 首开时间
 - 播流失败次数
 - 卡顿次数
 - 卡顿时长
 - 离线汇总数据：
 - 总缓存时长
 - 可用性
 - 平均下载速度
 - 平均播流时长
 - 缓存率（缓冲时长/播流时长）
- 播流报障模块
 - App → SDK → Server → 分析报告

THANK YOU & Q & A

