SDCC 2016
中国软件开发者大会
SOFTWARE DEVELOPER CONFERENCE CHINA

视频自适应播放实践

何李石 @ikbear
helishi@qiniu.com

- 播放器使用场景
- 播放器构成
- 自适应播放
- 开源播放引擎推荐

点播

直播

谁拥有以下设备之一？

- 小米 5、小米 Max、红米 Note、小米平板
- 华为荣耀、华为 Mate、华为 P9 / Plus、荣耀 Note
- iPhone 4 / 4S / 5 / 5S / 6 / 6 Plus / 6s / 6s Plus / 7
- 三星 Galaxy A7 / A9 / S5 / S6 / S7 / Note 4 / Note 7
- Google / LG Nexus 5 / 6
- 魅族、酷派、步步高、Vivo、Oppo

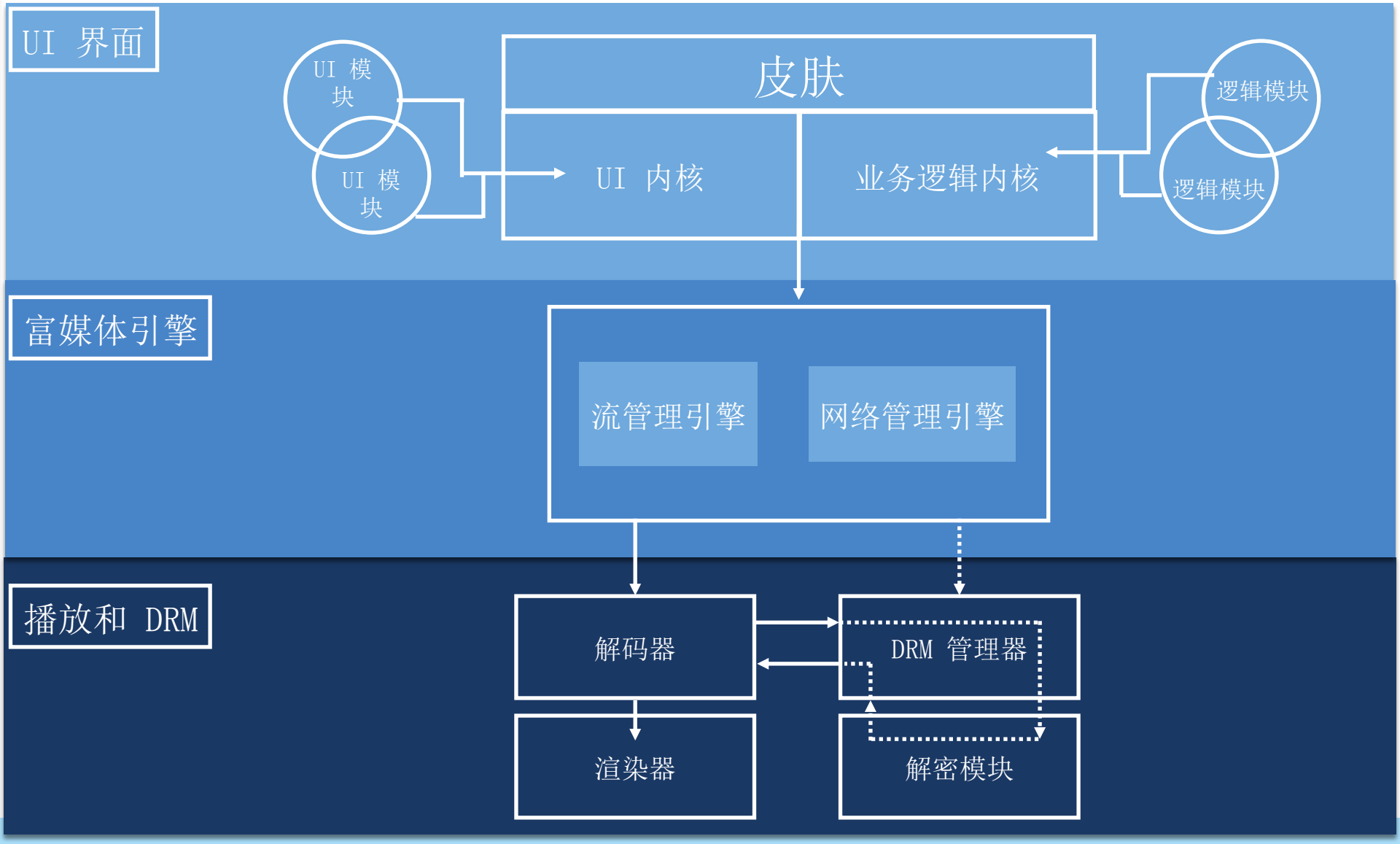
一个复杂的世界

- 多种平台？（软硬件）
- 多种尺寸？
- 网络：2G？3G？4G？Wifi？

解决方案：定制播放器

- 播放器使用场景
- 播放器构成
- 自适应播放
- 开源播放引擎推荐

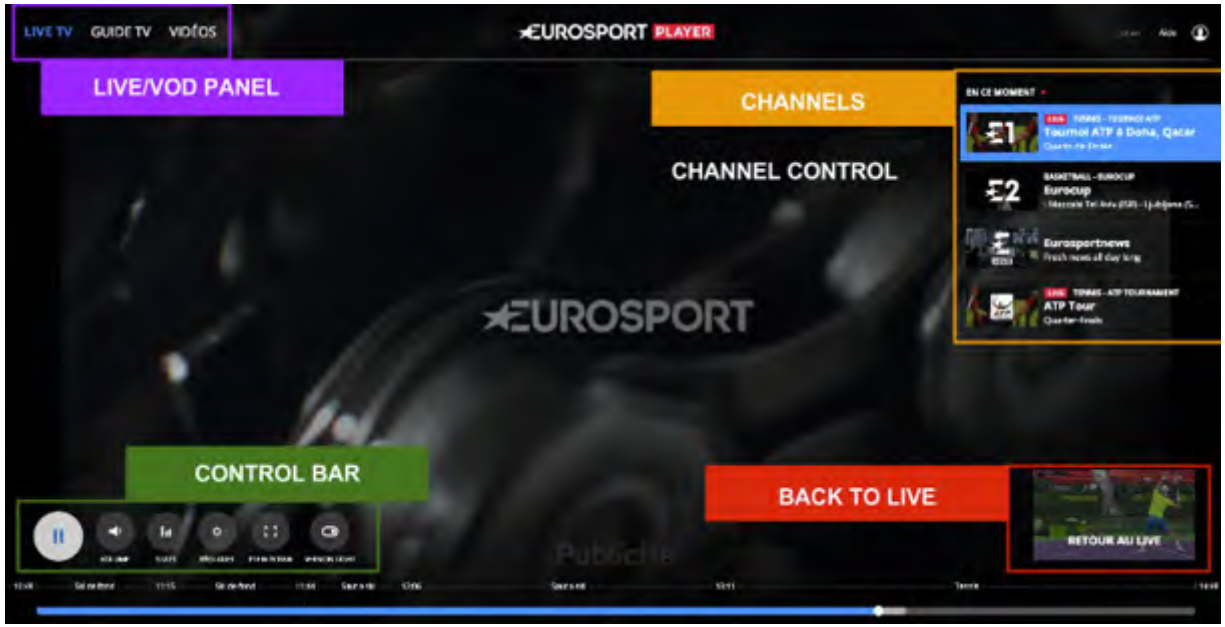
现代播放器架构


UI 界面
UI 模块
UI 模块
皮肤
UI 内核
业务逻辑内核
逻辑模块
逻辑模块
富媒体引擎
流管理引擎
网络管理引擎
播放和 DRM
解码器
渲染器
DRM 管理器
解密模块

皮肤

UI 逻辑

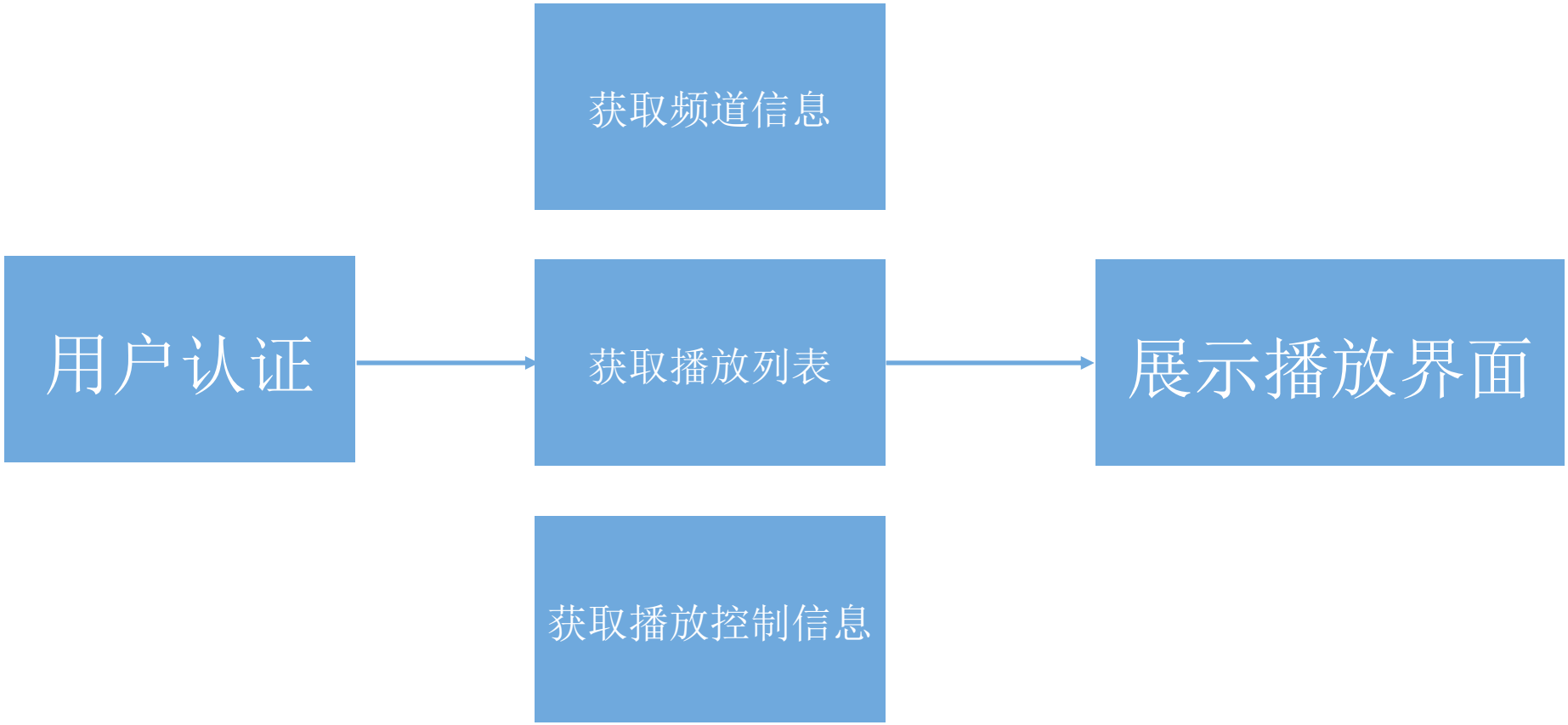
- 播放频道
- 播放列表
- 截图
- 社交分享按钮



业务逻辑（不可见部分）

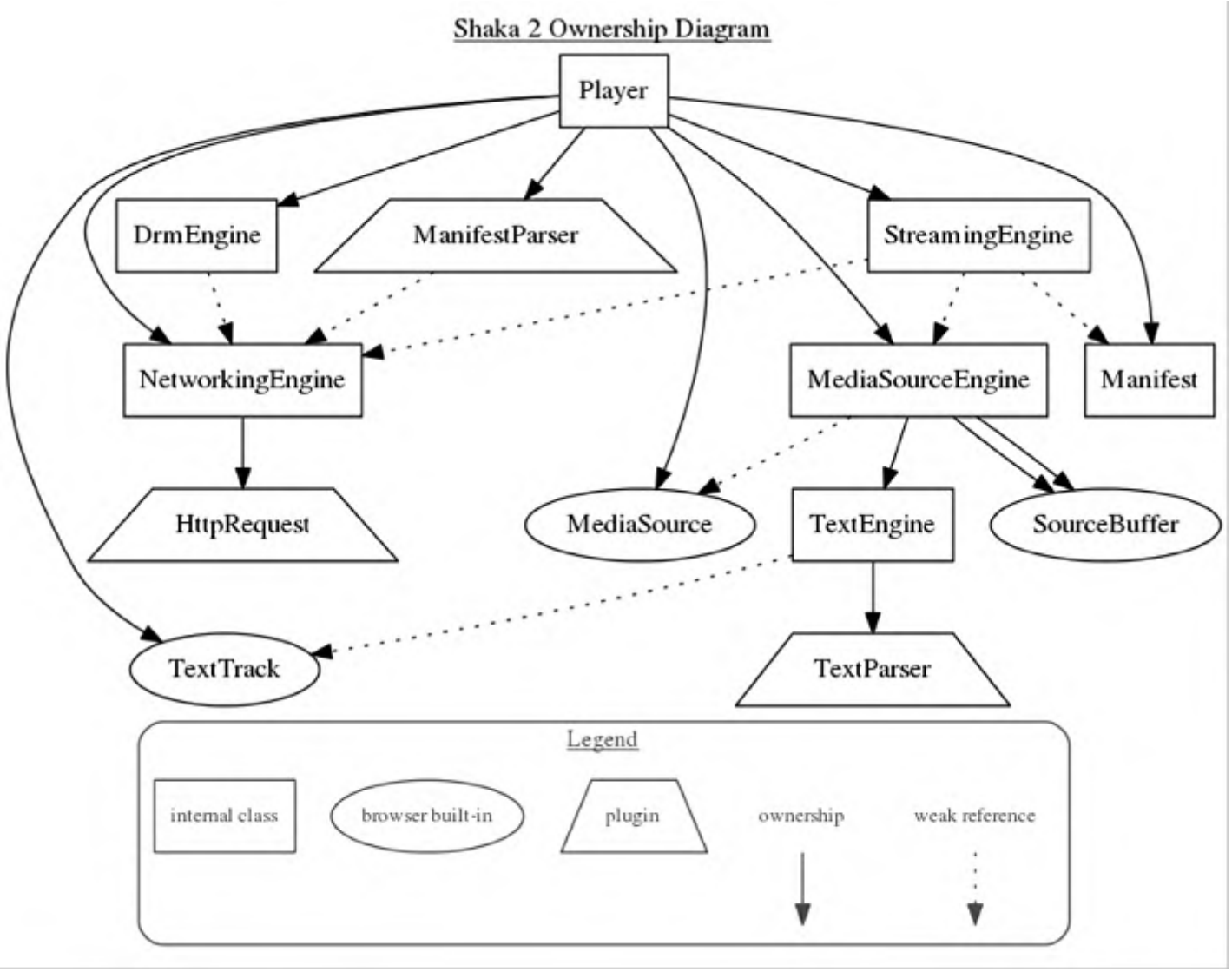
- 设备检测与配置逻辑
  - 浏览检测：hls.js for HTML 5, flashls for Flash
  - 屏幕检测：4K or 2K?
- A/B 测试逻辑：生产环境灰度，测试最新浏览器体验
- 广告逻辑：
  - 认证和支付逻辑
  - 插入视频播放前、中、后期广告

简单业务流程



播放引擎

Shaka Player 架构图



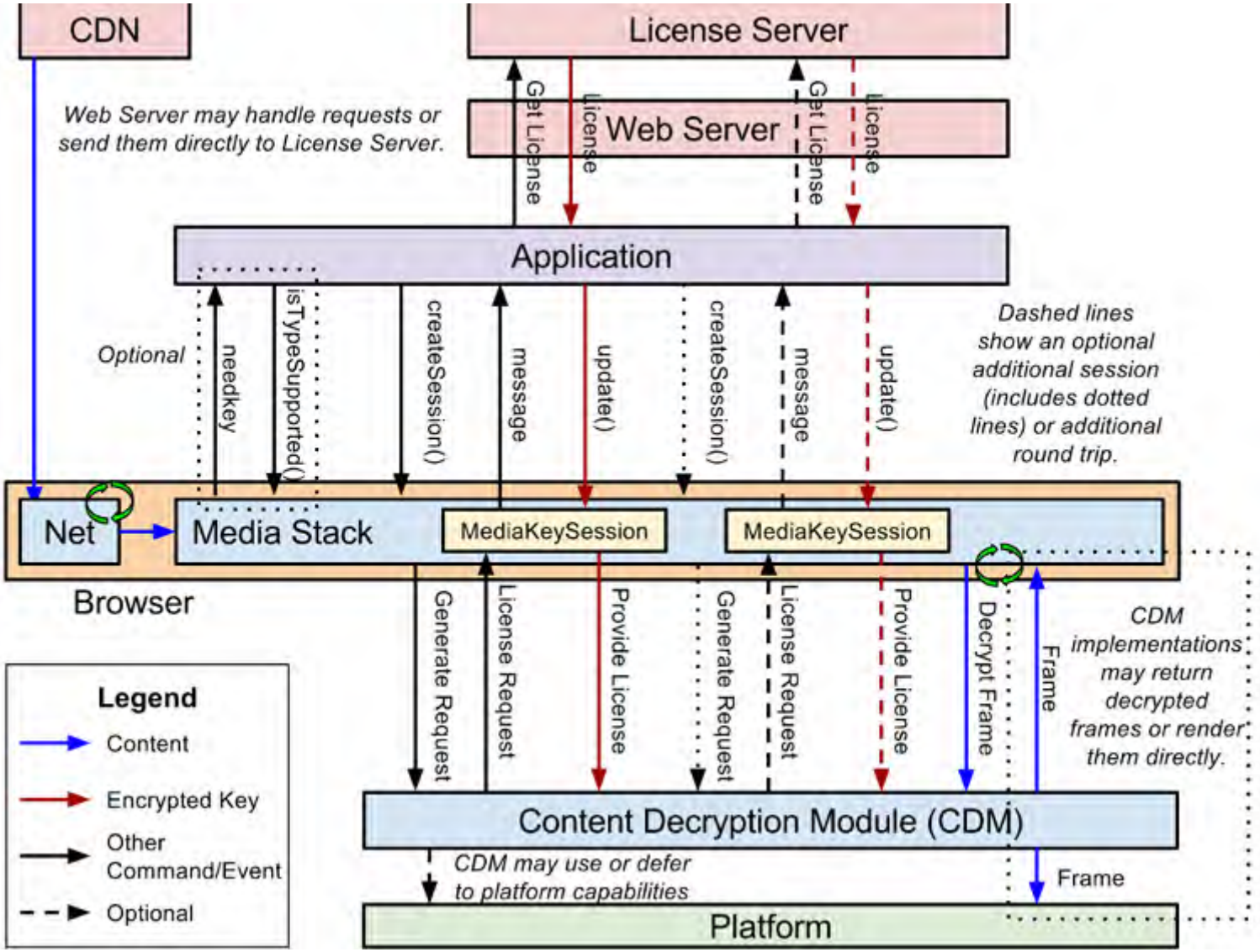
StreamingEngine - 视频流管理

- 声明文件解析和更新
  - HLS 中的 .m3u8 文件
  - MPEG-DASH 中的 .mpd 文件
- 流媒体播放引擎
  - 解码器
  - DRM 解密模块
  - 渲染器

NetworkingEngine - 网络管理

- 下载器（包含重拾策略）
  - 声明文件下载：.m3u8 和 .mpd 文件
  - 多媒体片段下载：.ts 文件
  - DRM License 下载（HTTPS）
- 资源预估模块：带宽、CPU 和帧率资源
- ABR 切换控制器：码率或者画质选择模块
- Buffer 配置：
  - 固定大小 Buffer
  - 可变大小 Buffer

DrmEngine - 版权保护管理



DRM License 权限验证

基于头部的验证：

```
request.headers['CWIP-Auth-Header'] = 'VGhpc0lzQVRlc3QK';
```

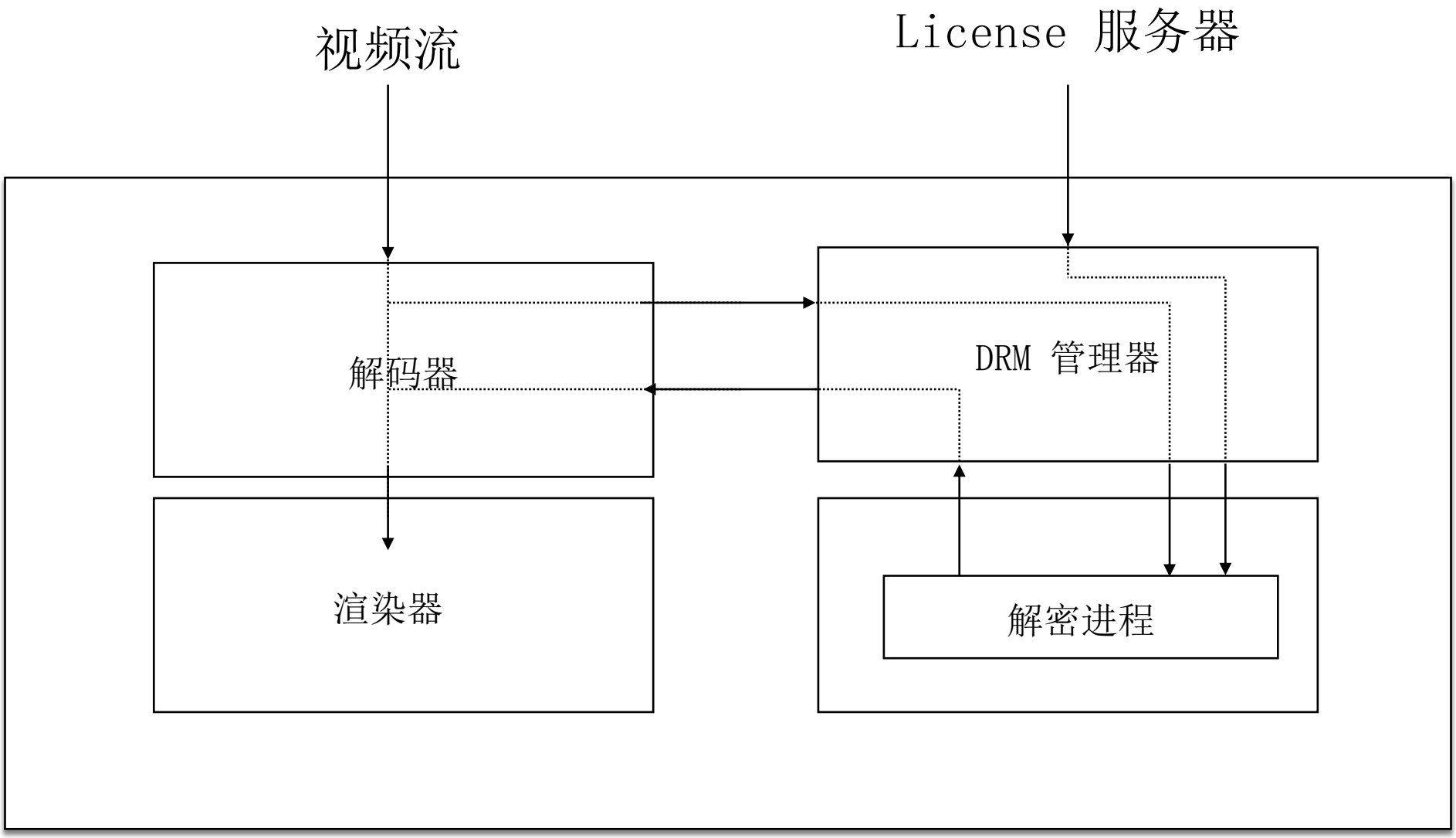
基于参数的验证：

```
request.uris[0] += '?CWIP-Auth-Param=VGhpc0lzQVRlc3QK';
```

基于 cookie 验证（跨域）：

```
document.cookie = "CWIP-Auth-Cookie=VGhpc0lzQVRlc3QK"
```

解码器、渲染器和 DRM 工作流程



- 播放器使用场景
- 播放器构成
- 自适应播放
- 开源播放引擎推荐

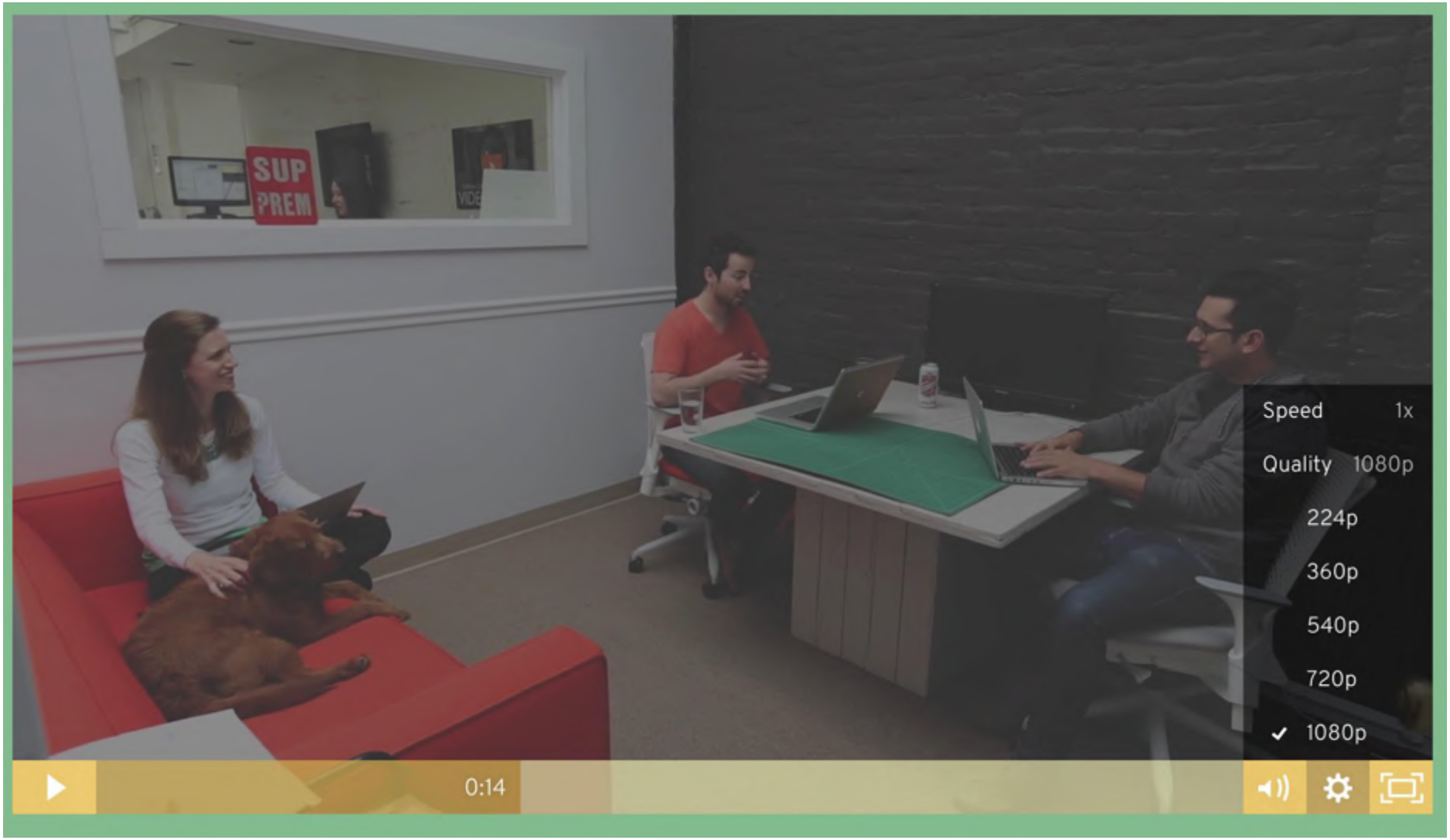
自适应播放

- 设备自适应
  - 设备检测：操作系统、硬件
  - 多尺寸（分辨率）自适应
- 网络自适应
  - 多码率切换

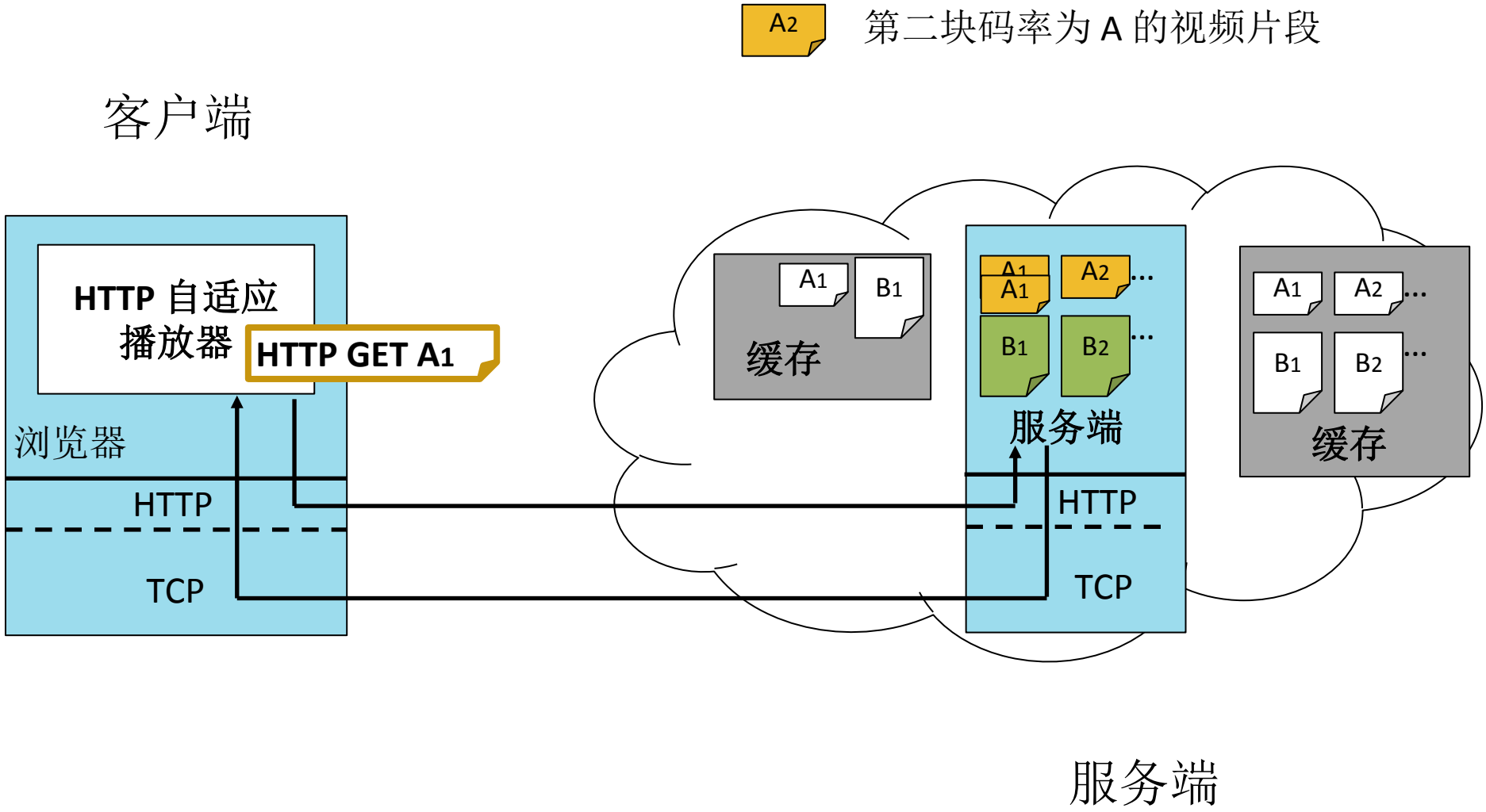
Safari https://www.youtube.com/html5

Chrome https://www.youtube.com/html5

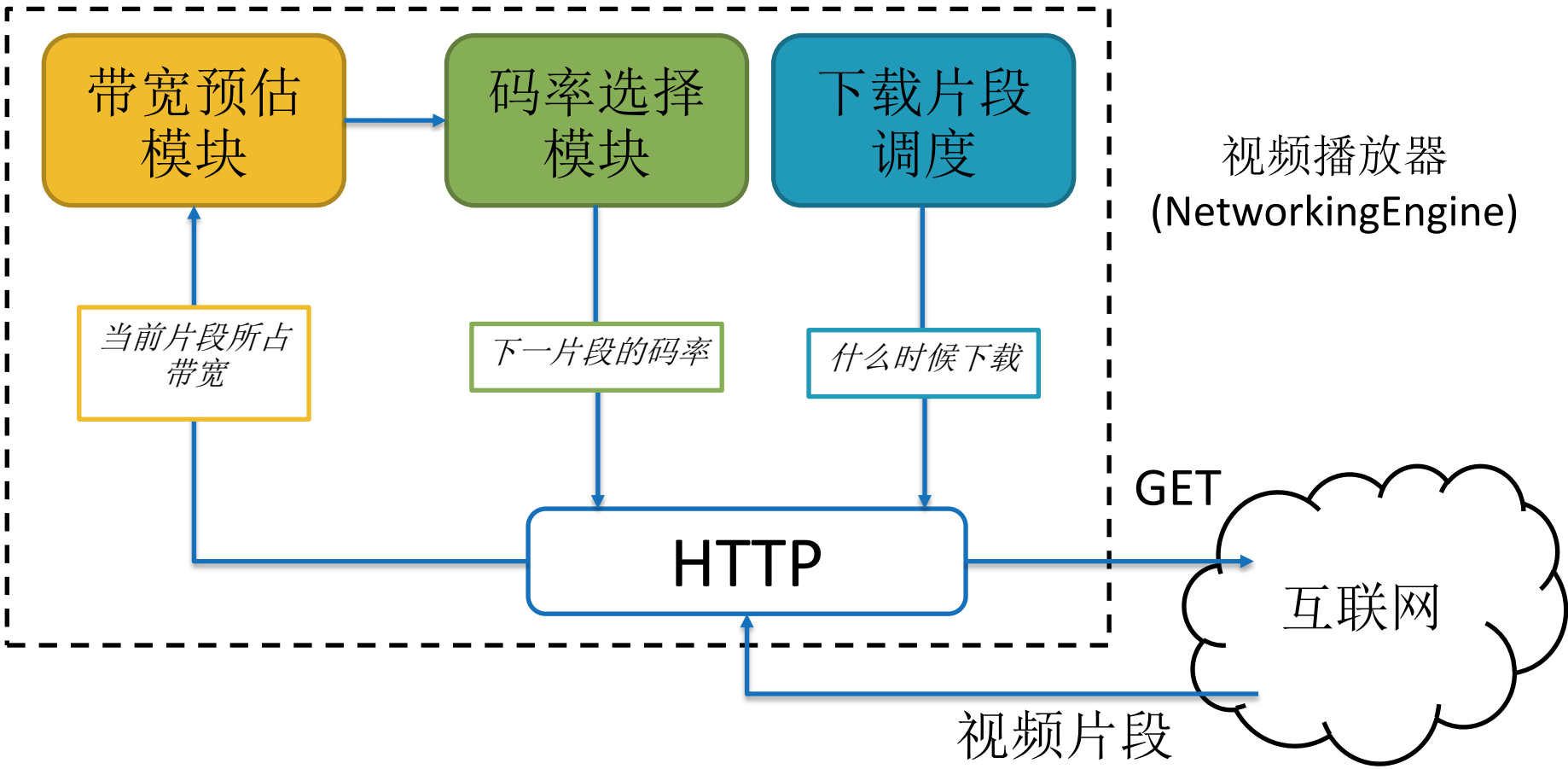
多尺寸自适应


SUP
PREM
Speed 1x
Quality 1080p
224p
360p
540p
720p
1080p
0:14

自适应播放流程


A2
第二块码率为 A 的视频片段
客户端
HTTP 自适应
播放器
HTTP GET A1
浏览器
HTTP
TCP
A1
B1
缓存
A1
A1
A2
B1
B2
服务端
HTTP
TCP
A1
A2
B1
B2
缓存
服务端

自适应播放原理



- 三大模块
- 播放器和外部之间通过反馈实时调整码率、画质

速度 VS. 质量

- 启动时
  - 占用最小可行带宽
  - 启动速度优先，低质量画面开始
- 稳定时
  - 质量优先，通过控制 Buffer 大小控制缓冲率
  - 实时监测网络：带宽预估 => 码率选择 => 带宽预估

编码参数参考

质量	分辨率	视频码率 (kbps)	音频码率 (kbps)	FPS	视频编码器	H.264 Profile	关键帧间隔	音频编码器	音频通道数
Low	480x270	400	64	15/30	H.264	Baseline	2s	AAC	单
Med	640x360	800-1200	96	30	H.264	Main	2s	AAC	单/立体
High	960x540	800-1500	96	30	H.264	Main	2s	AAC	单/立体
HD 720	1280x720	1200-4000	128	30	H.264	Main	2s	AAC	单/立体
HD 1080	1920x1080	4000-8000	192	30/30+	H.264	Main/High	2s	AAC	立体

（网络） 容错度 VS. 缓冲率

- Buffer 越小
  - 容错度越低，网络抖动造成的干扰越大
  - 启动时间越短，缓冲率越小
- Buffer 越大
  - 容错度越高，网络抖动造成的干扰越小
  - 启动时间越长，缓冲率越大

解决方案：多 Buffer 管理策略

- 双 Buffer 缓冲策略
  - 启动时，Buffer 非常小，只要缓冲满即可开始播放
  - 启动后，Buffer 调大，容忍网络抖动甚至故障
- 视频直播中的秒开
  - 服务端缓存第一帧为关键帧
  - 首次启动时无 Buffer（大小为 0），立即播放
  - 耗时来自于：下载第一个片段耗时（1-3s）

播放协议：HLS VS. DASH

- HLS
  - 封装格式老旧，适合视频广播时代：M2TS
  - HLS 不要求片段以关键帧开始
- MPEG-DASH
  - 封装格式更适合互联网上的流媒体传输：碎片化的 MP4 - fMP4
  - MPEG 要求以关键帧开始，适合多码率切换：
    - 不用重复下载已播放片段
    - 不用等待关键帧解码播放，快速切换

- 播放器使用场景
- 播放器构成
- 自适应播放
- 开源播放引擎推荐

播放器对比

平台	格式	播放引擎
HTML5	HLS	hls.js, videojs-contrib-hls
	DASH	Dash.js, Shakaplayer, rx-player
	Smooth Streaming	rx-player
Flash	HLS	Flashls / flv.js
	DASH	Dash.as
Android	HLS	ExoPlayer, 原生
	DASH	ExoPlayer, 原生
iOS	HLS	原生
	DASH	暂无

hls.js 和 flv.js 原理

- hls.js (https://github.com/dailymotion/hls.js)
  - 支持 HTML5 下使用 <video> 播放 HLS
  - 将 MPEG-2 格式视频转换成 fMP4，通过 MediaSource Extensions API 填充 SourceBuffer
- flv.js (https://github.com/Bilibili/flv.js)
  - 支持 HTML5 下使用 <video> 标签播放 FLV
  - 将 FLV 格式视频转换成 fMP4，通过 MediaSource Extensions API 填充 SourceBuffer

总结

- 现代浏览器开放了更多的 API 支持播放器的构建。Android 和 iOS 上也提供了更多开源的库方便定制播放器：
  - 七牛推流、播放 SDK: https://github.com/pili-engineering
  - HLS 播放引擎：https://github.com/dailymotion/hls.js
  - FLV 播放引擎：https://github.com/Bilibili/flv.js
- 基于 MPEG-DASH 可以做到更好的自适应播放

SDCC 2016

中国软件开发者大会

SOFTWARE DEVELOPER CONFERENCE CHINA

谢 谢！