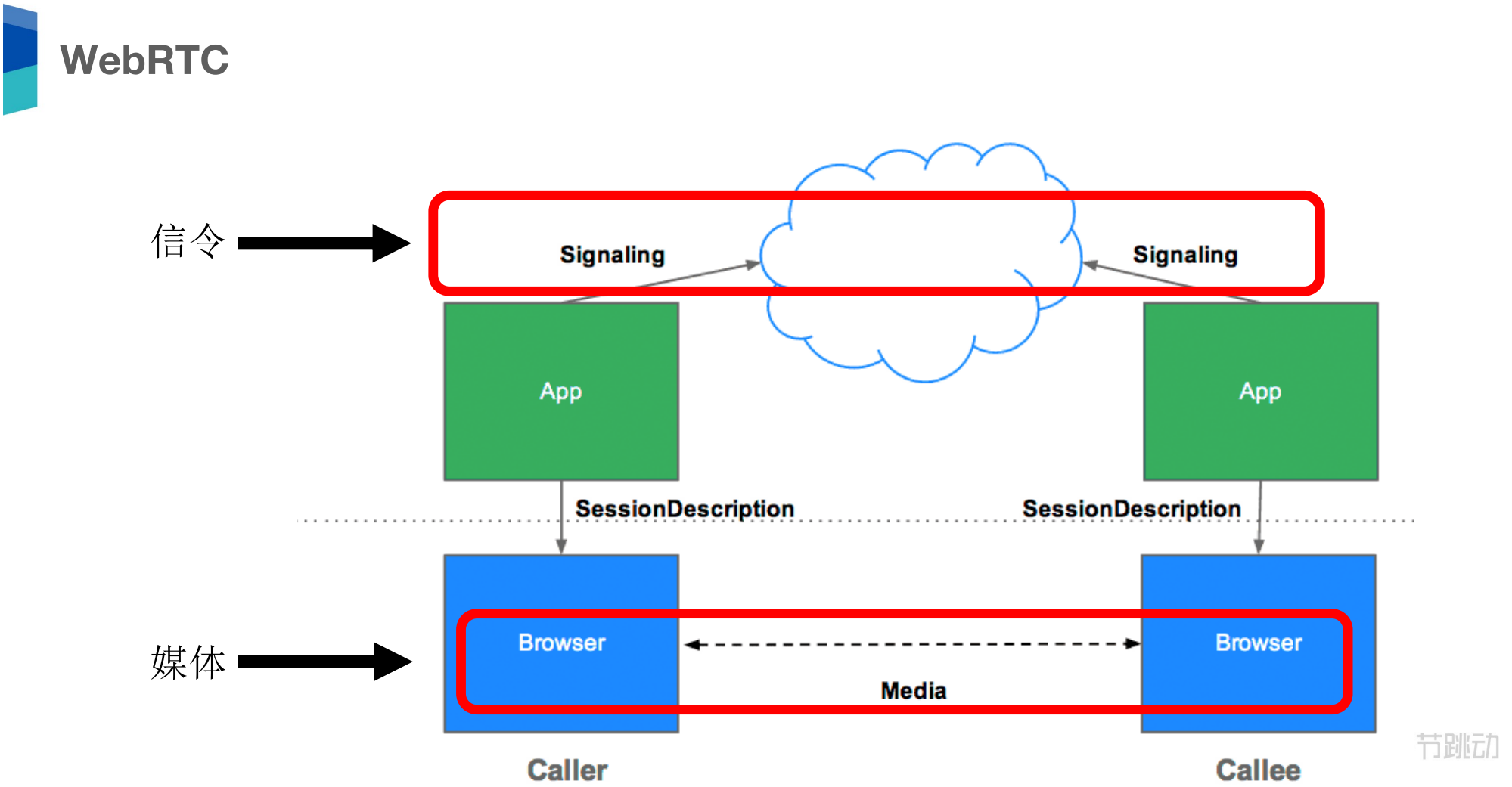
WebRTC 媒体传输探索和信令标准化应用

字节跳动 陈成

ByteDance字节跳动


WebRTC
信令
媒体
Signaling
Signaling
App
App
SessionDescription
SessionDescription
Browser
Browser
Media
Caller
Callee

目　录

ByteDance字节跳动

WebRTC 媒体传输探索

ByteDance字节跳动

WebRTC 媒体数据传输流程

采集

摄像头/麦克风：

MediaDevices.getUserMedia

屏幕：

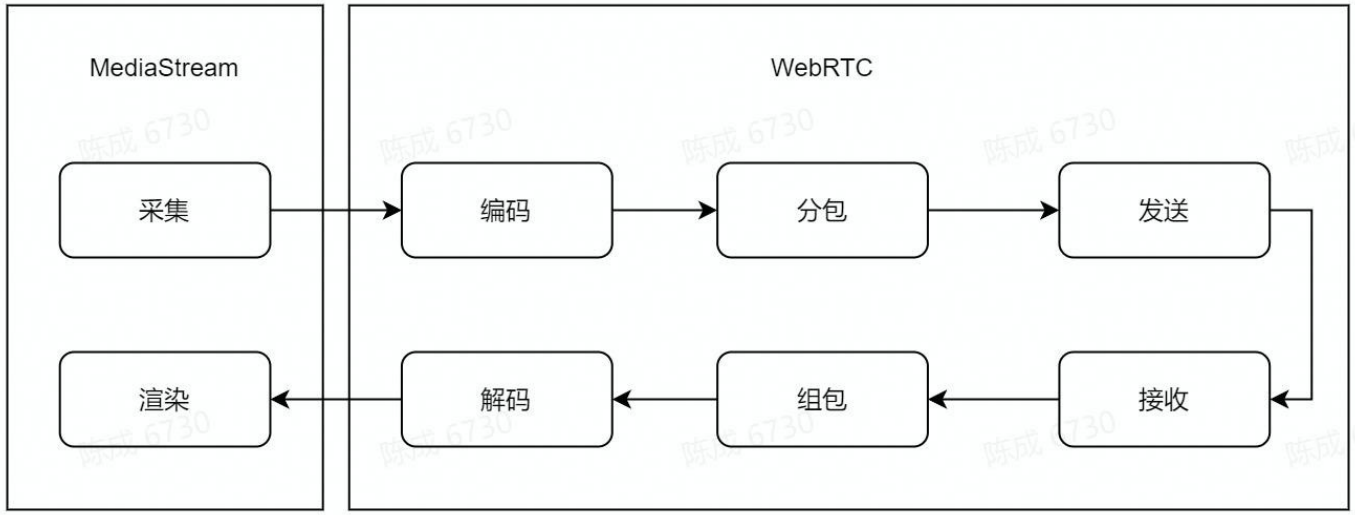
MediaDevices.getDisplayMedia

自定义：

<video/canvas>.captureStream

渲染

<video>.srcObj



ByteDance字节跳动

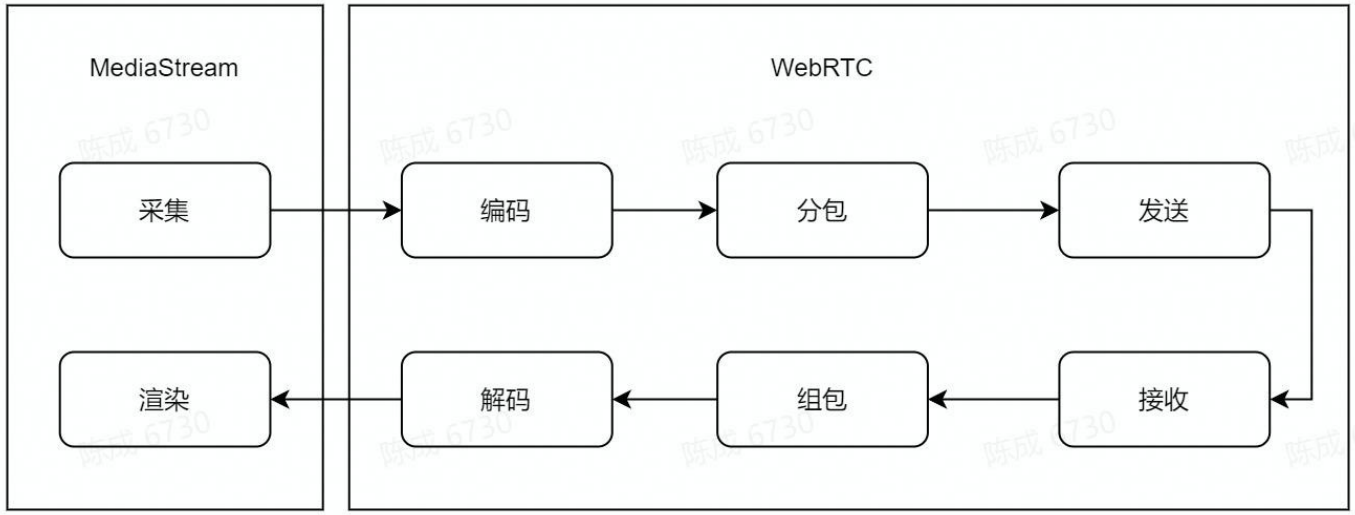
WebRTC 媒体数据传输流程

对于RTC服务提供商的痛点

- 不可定制化
- 各厂商差异性小
- 新功能特性依赖于Google开发

解决方案

- 通过新API来提供定制化的能力
- Unbundling WebRTC



ByteDance字节跳动

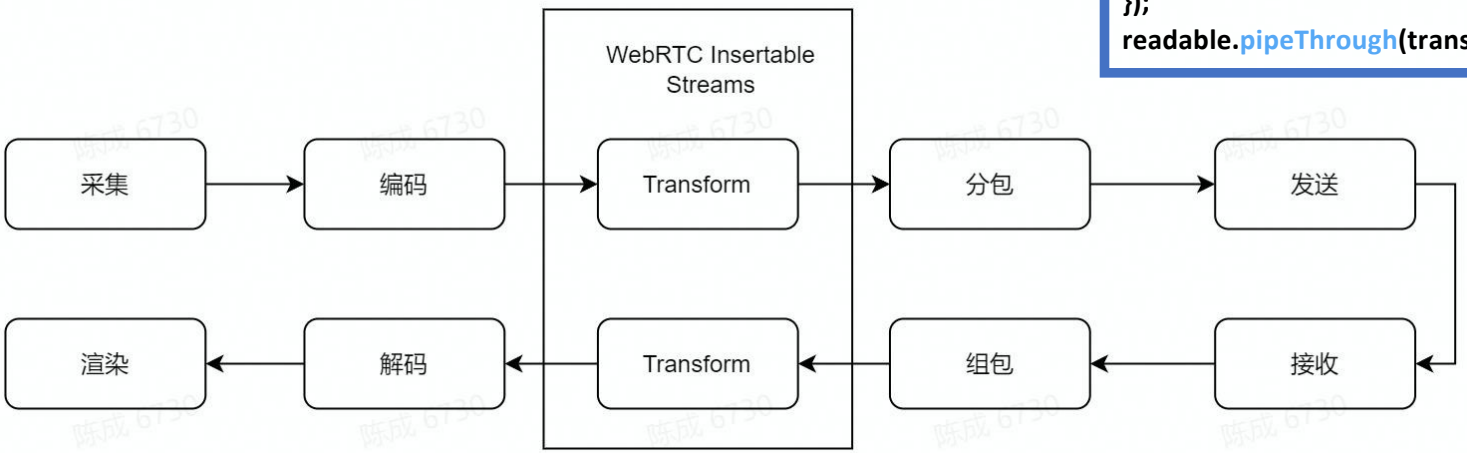
WebRTC Insertable Streams

https://www.w3.org/TR/webrtc-encoded-transform/

对编码后/解码前的数据进行自定义的处理

Chrome 86

```
const { readable, writable } = sender.createEncodedStreams();
const transformer = new TransformStream({
 transform: (chunk, controller) => {
  ...
  chunk.data = newData;
  controller.enqueue(chunk);
 }
});
readable.pipeThrough(transformer).pipeTo(writable);
```



ByteDance字节跳动

WebRTC Insertable Streams

https://www.w3.org/TR/webrtc-encoded-transform/

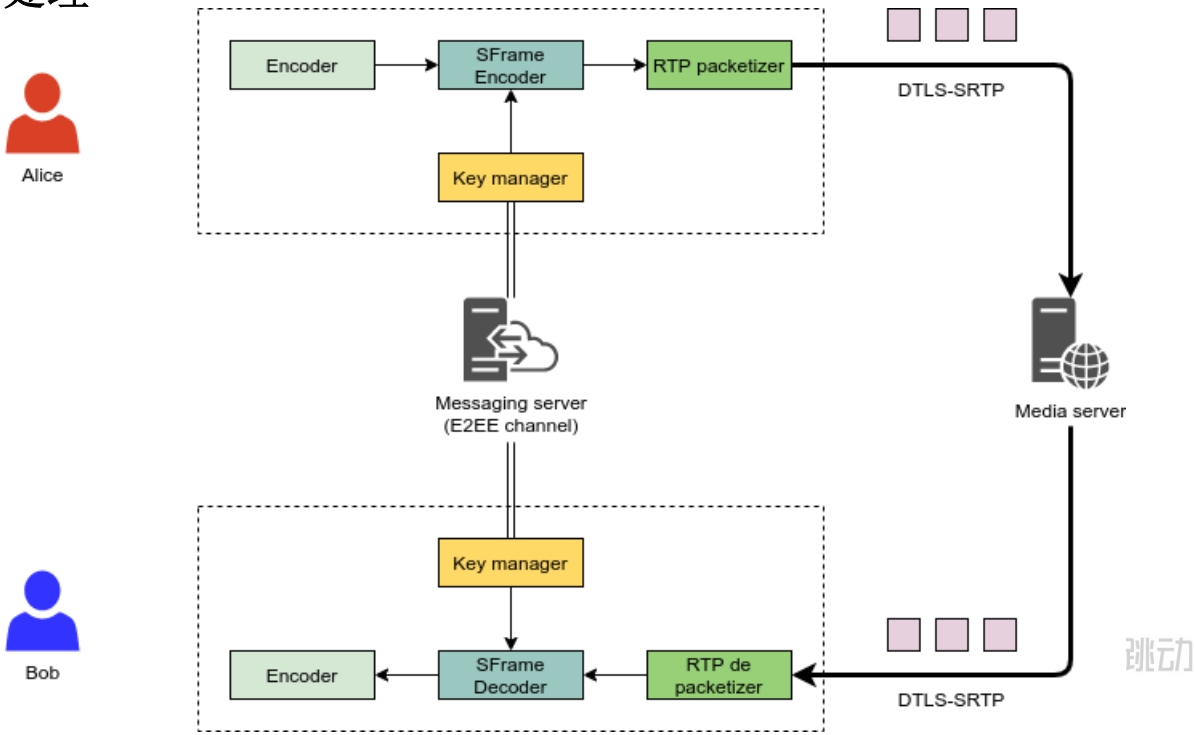
对编码后/解码前的数据进行自定义的处理

Chrome 86

- 端到端加密
- 冗余控制
- SEI

……

Peer connection end to end encryption



WebRTC Insertable Streams

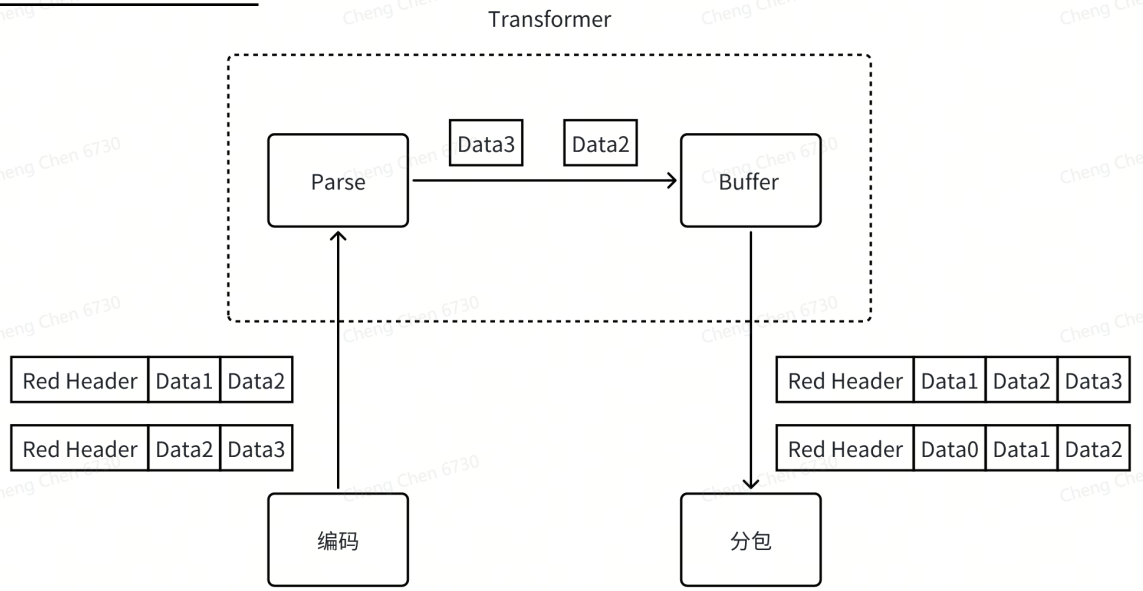
https://www.w3.org/TR/webrtc-encoded-transform/

对编码后/解码前的数据进行自定义的处理

Chrome 86

- 端到端加密
- 冗余控制
- SEI

……



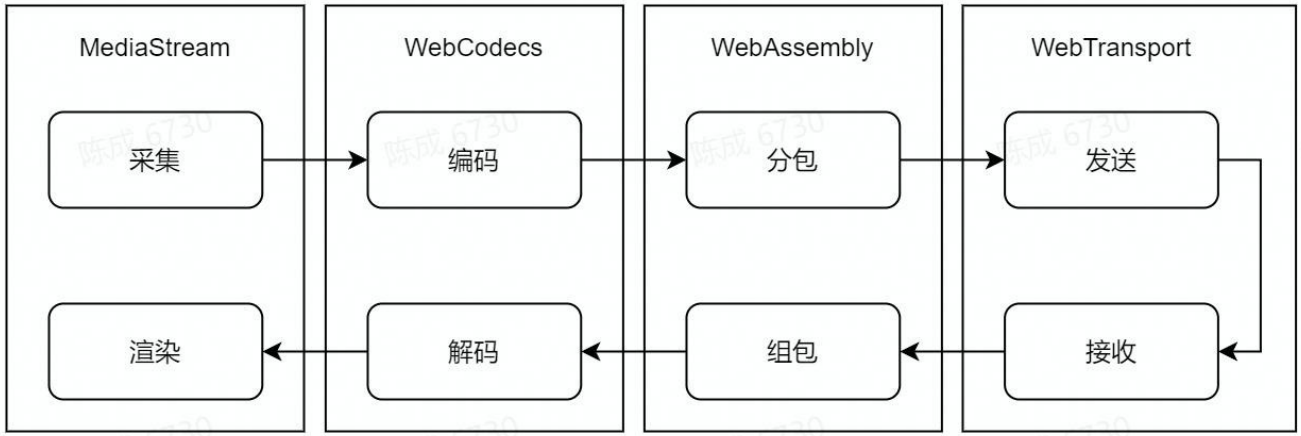
Redundant Audio Data(RED)のカスタムエンコーダを作って WebRTC の音声品質の低下を防ぐ

ByteDance字节跳动

Unbundling WebRTC

https://www.w3.org/TR/webcodecs/ 提供调用浏览器内置编解码器的接口 **Chrome 94**

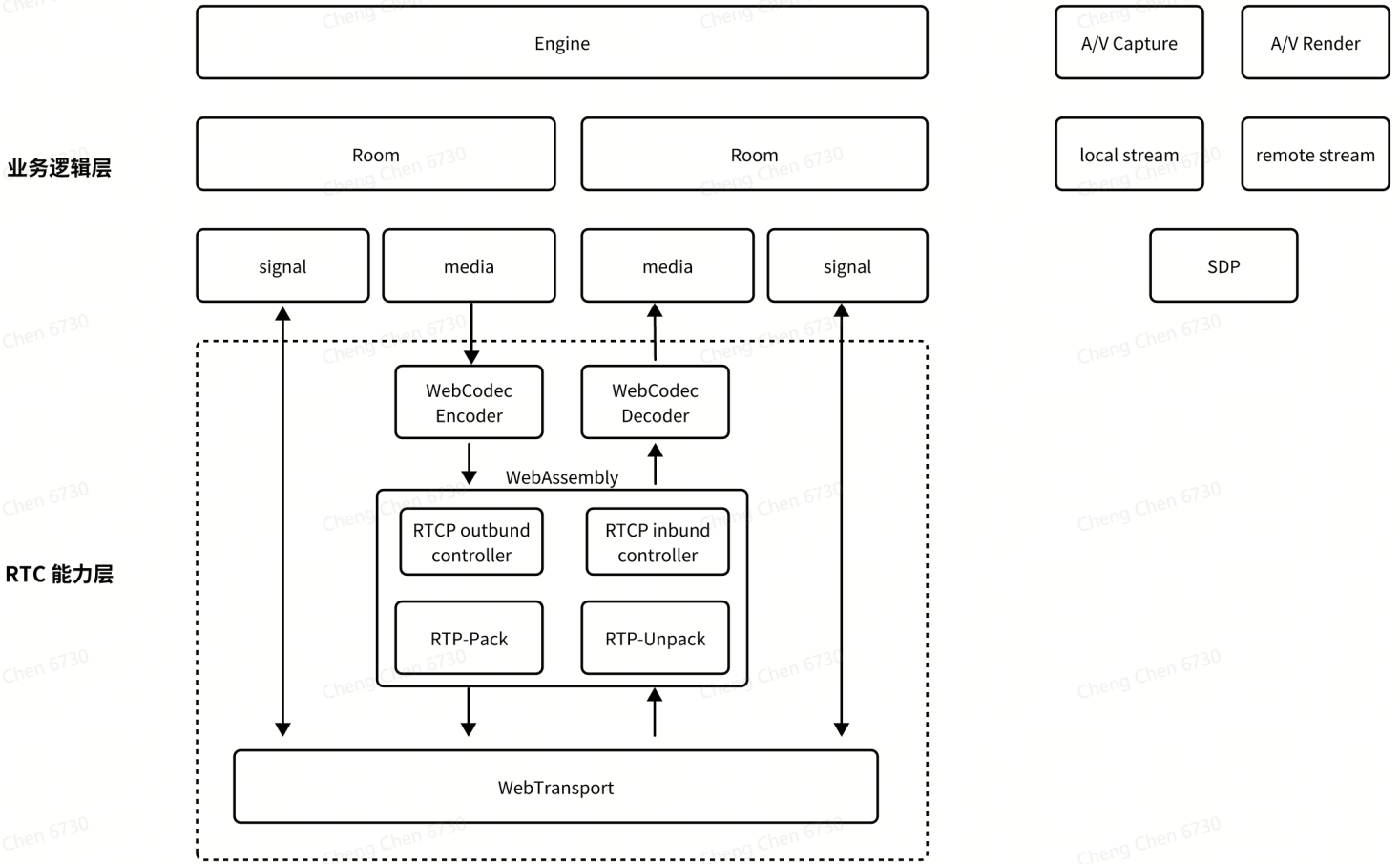
https://www.w3.org/TR/webtransport/ 基于**HTTP3**的双向传输通道 **Chrome 97**



- 使用WebTransport的不可靠通道
- 需要自己实现QoS控制及分包组包逻辑
- 适用于Client-Server
- 可以更定制化地开发RTC功能

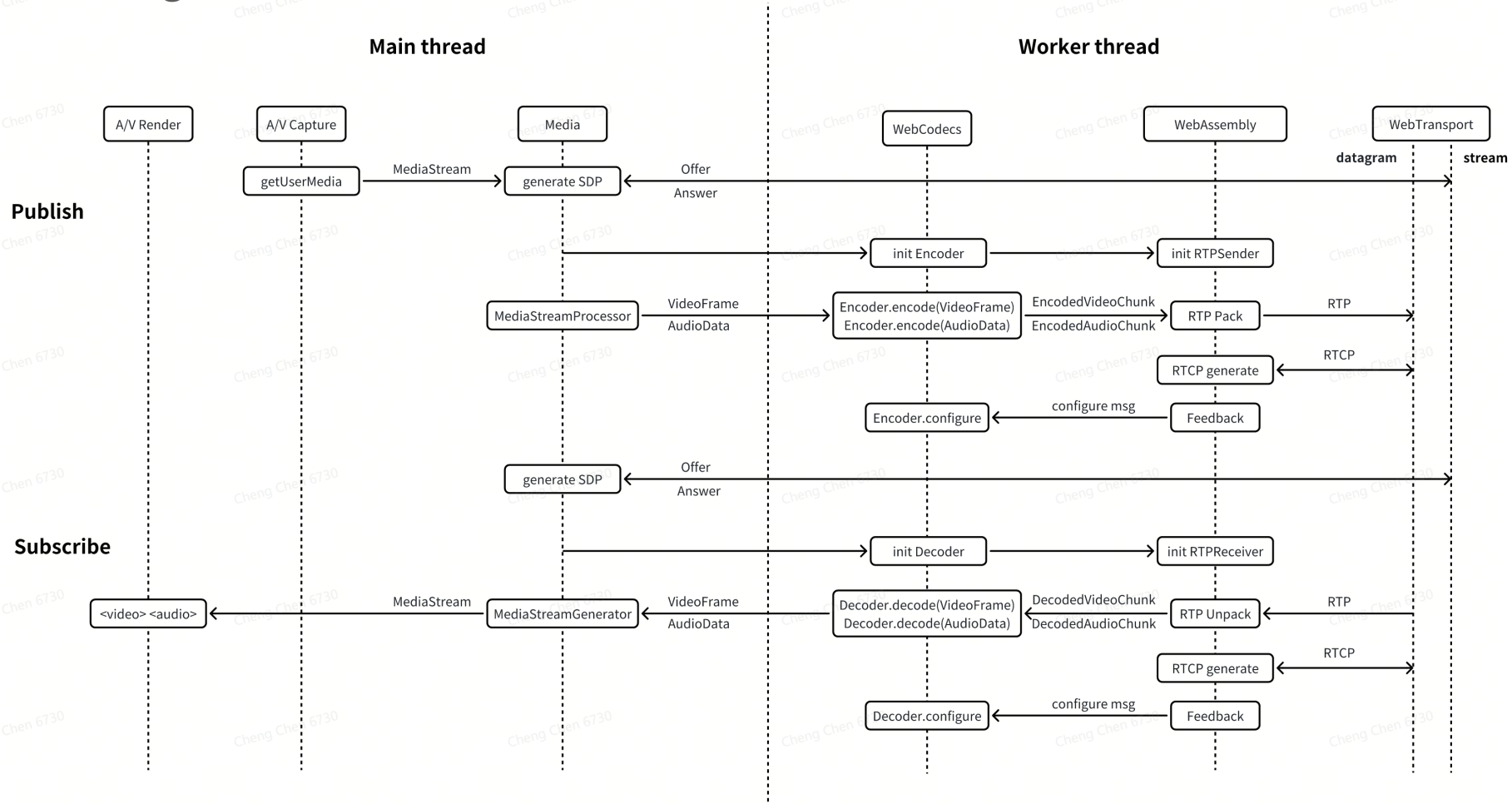
ByteDance字节跳动

Unbundling WebRTC — SDK架构



字节跳动

Unbundling WebRTC — SDK时序图



Unbundling WebRTC — 问题和总结

问题：

WebTransport

- 底层quic协议内置拥塞控制对实时音视频通信场景不够友好
- 单连接上不同包类型优先级能力欠缺
- 底层带宽估计信息没有反馈给应用开发者，上层需要自己独立带宽估计，带宽分配

WebCodecs

- opus编码默认60ms帧长不可配，不适用于RTC低延迟要求
- opus编码还不支持inband-fec, 丢包反馈等特性

浏览器兼容性和实现

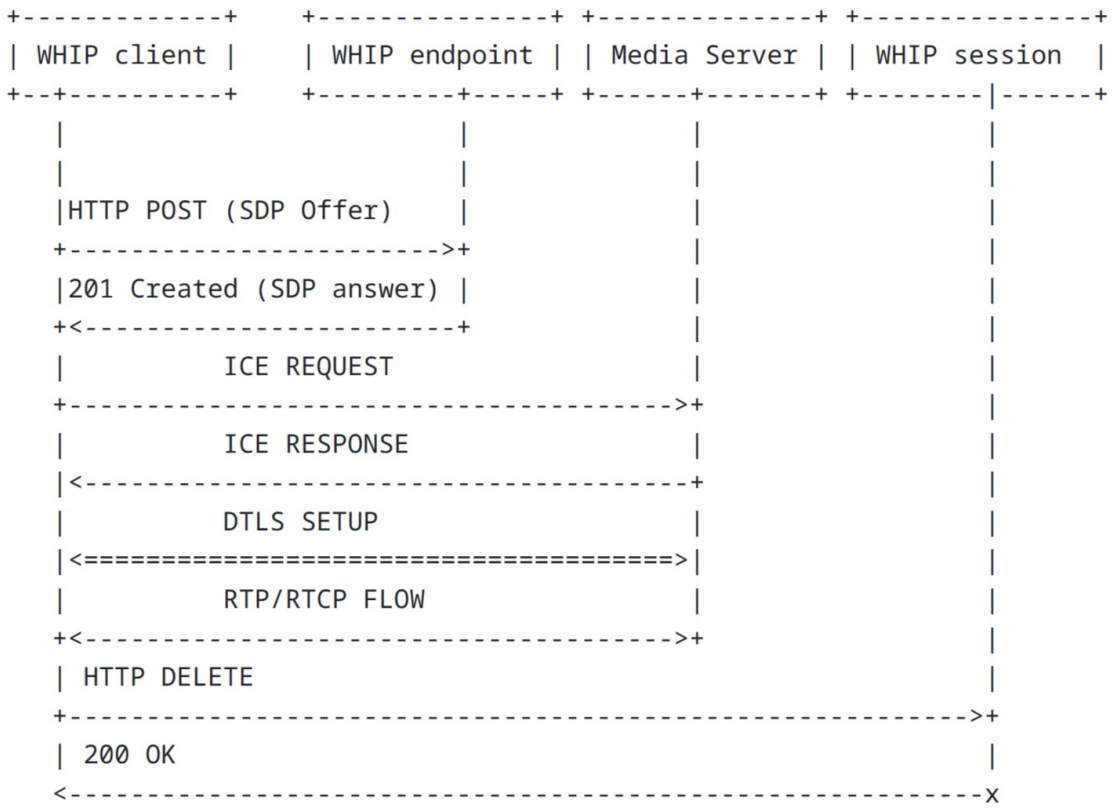
总结：距离实现具备WebRTC能力的产品化还需要时间打磨

ByteDance字节跳动

WebRTC 信令标准化应用

ByteDance字节跳动

WHIP/WHEP — WebRTC HTTP Ingestion/Egress Protocol



草案现状

WHIP: 由Millicast发起，目前正在被IESG审查，即将成为正式标准

https://datatracker.ietf.org/doc/draft-ietf-wish-whip/

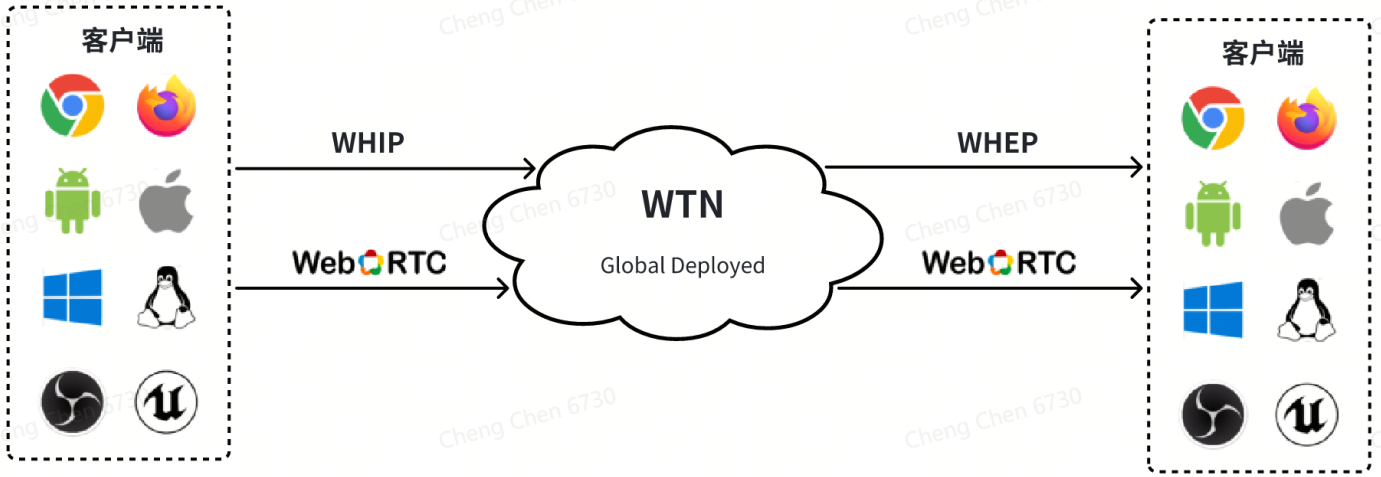
WHEP: 由ByteDance和Millicast发起，目前已被IETF工作组采纳，成为工作组草案，目标2024年底提交IESG审查

https://datatracker.ietf.org/doc/draft-ietf-wish-whep/

ByteDance 字节跳动

WebRTC Transmission Network

RTC厂商通过WHIP和WHEP协议，可以将服务端的能力对外开放，实现直接的推拉流，方便开发者，促进更多的WebRTC应用

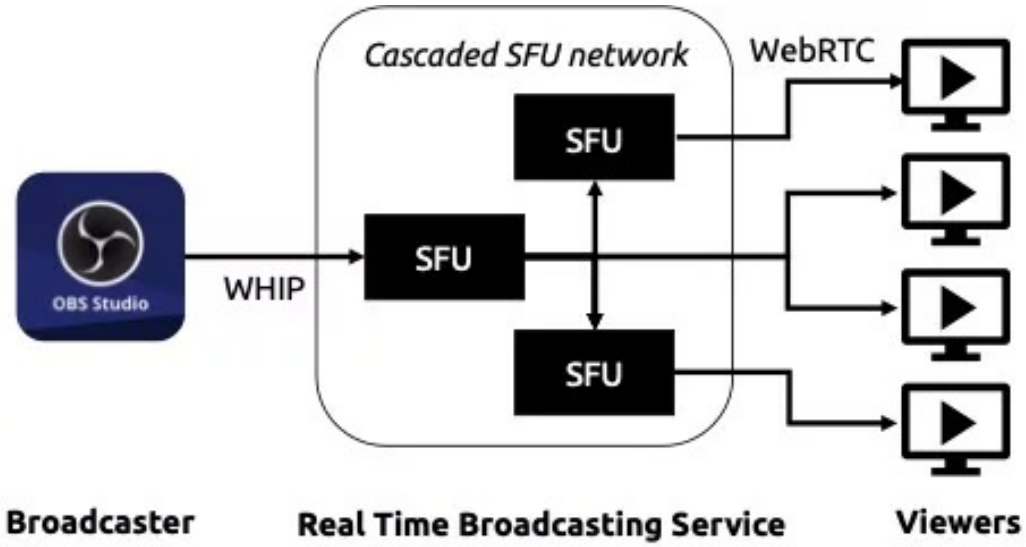


WTN vendor

- Millicast (Dolby.io)
- ByteDance
- Cloudflare

ByteDance字节跳动

应用 – 低延时分发



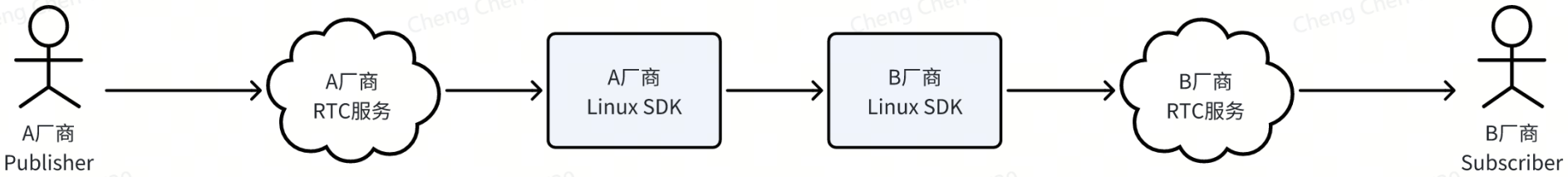
OBS 30 正式支持WHIP推流

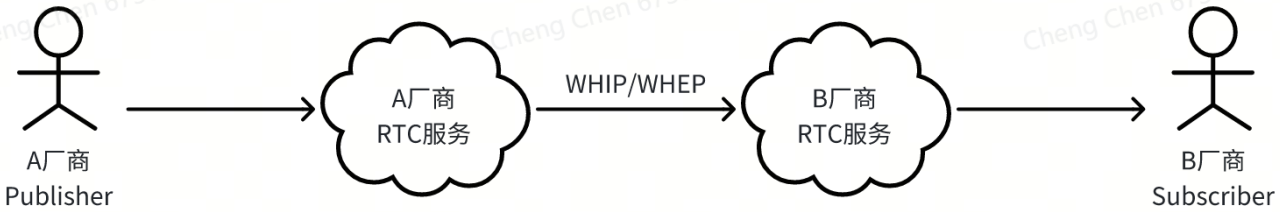
Client	Server
•OBS	•SRS
•GStreamer	•Janus
•FFmpeg*	•Livekit
•DJI	
•Flowcaster	

*非原生支持

ByteDance字节跳动

应用 – 厂商互通

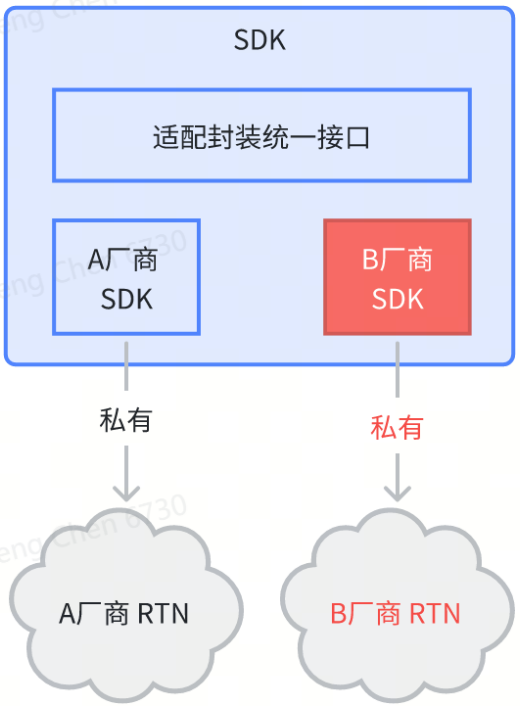




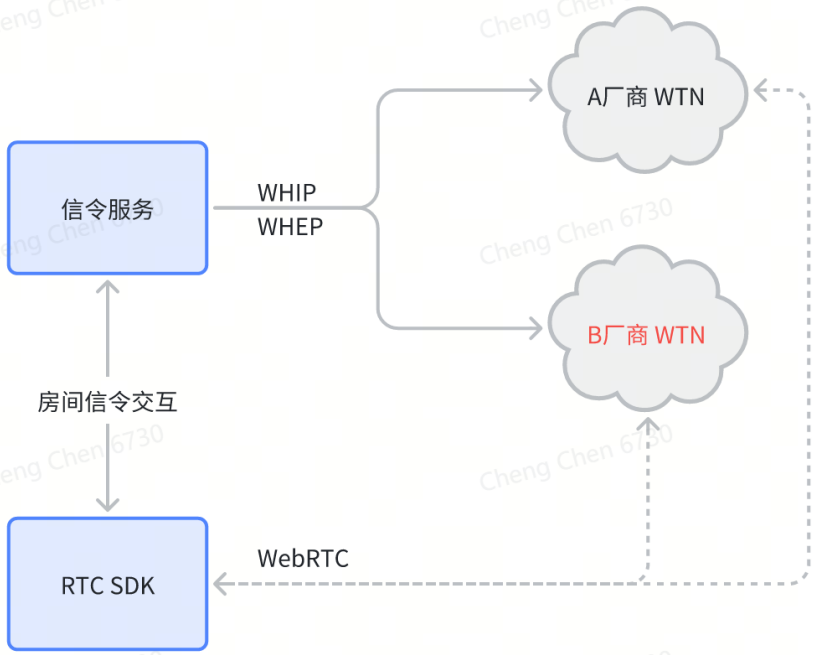
Cheng Chen 6730
ByteDance字节跳动

应用 — 线路备份

旧架构



新架构



Cheng Chen 6730
ByteDance 字节跳动

总结

总结

- RTC厂商需要更多WebRTC媒体传输的定制化能力，来发挥各自的技术，实现彼此的差异化
- 利用WebCodecs和WebTransport实现WebRTC Unbundling，可以释放更多开发的能力，但还需要不断地探索和时间的打磨
- 随着WHIP和WHEP的提出，客户端有了标准化的信令来快速接入RTC厂商服务端，方便了WebRTC应用的开发，也使得各厂商可以实现互联互通

ByteDance字节跳动

THANKS.

ByteDance字节跳动