Webtransport & Webcodecs 实现RTC及其标准参与实践

李德平

2023.06 北京

ByteDance字节跳动

自我介绍

2020--至今

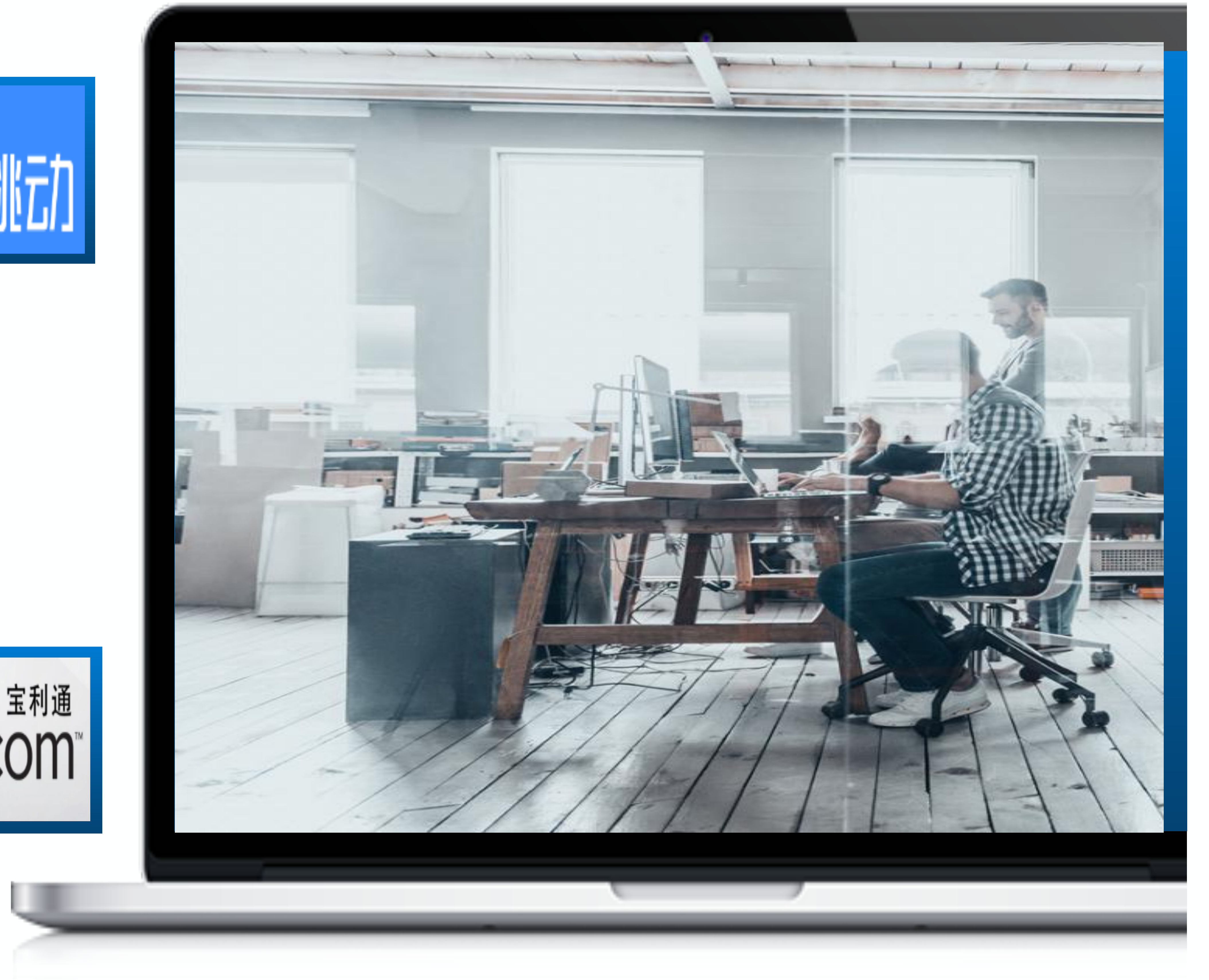
火山引擎RTC

从事WebRTC相关前后端研发工作

2012--2020

宝利通(Polycom)中国研发中心

从事基于标准的视频会议系统研发工作



目录

CONCENTS

ByteDance字节跳动

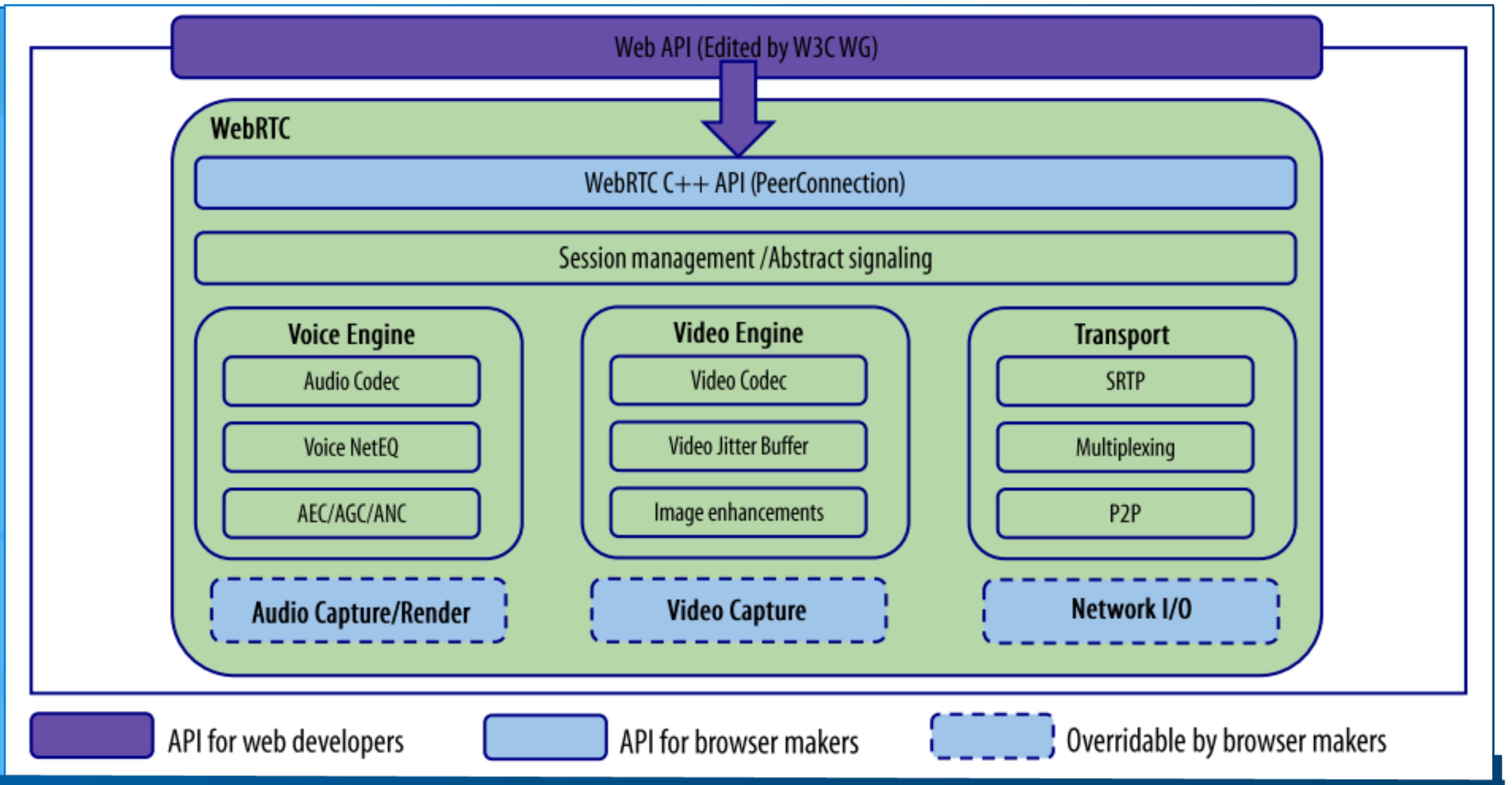
01

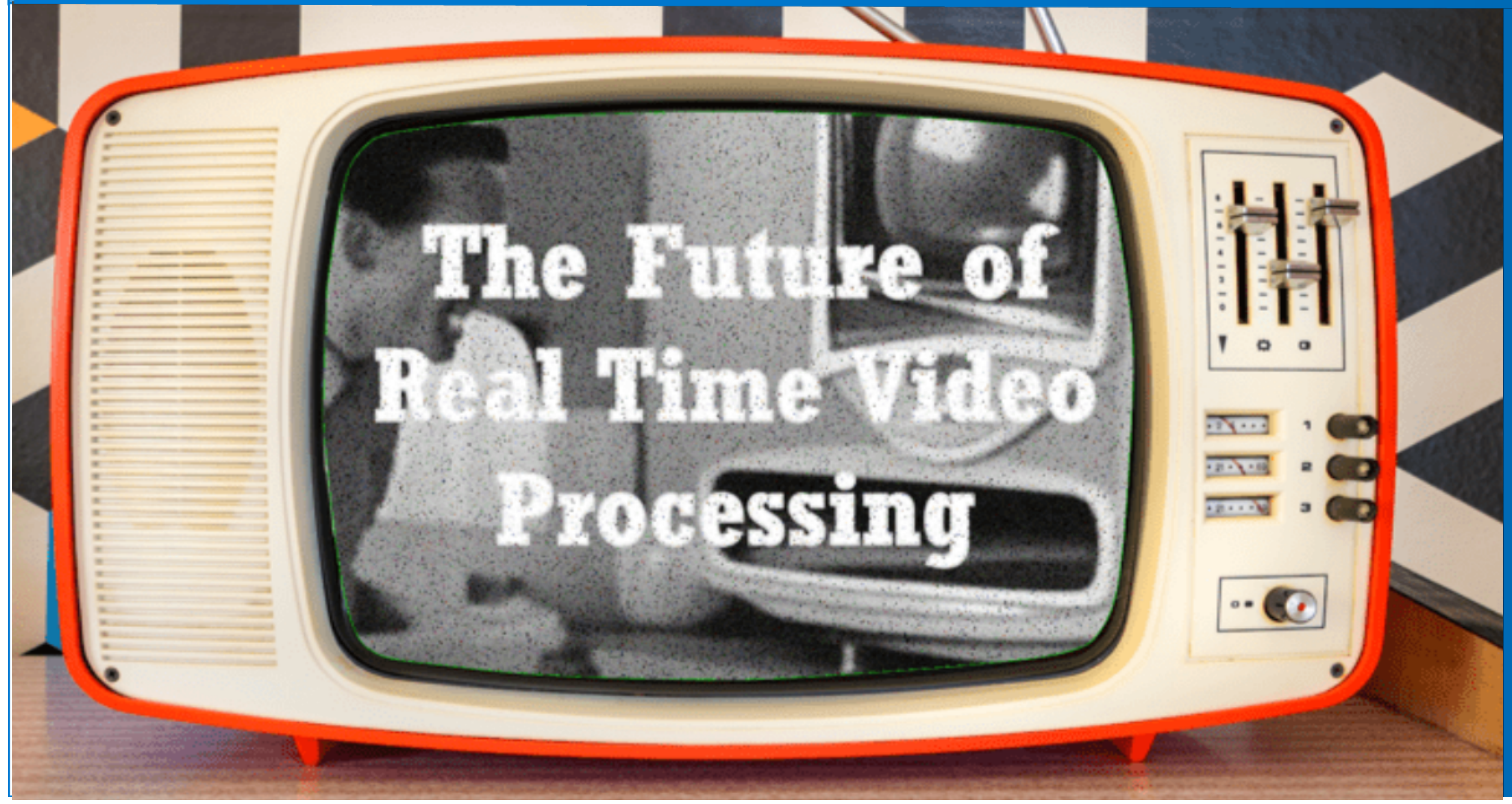
背景

背景

1. WebRTC一直以来是Web端实时音视频的主要选择，但是，WebRTC在浏览器端高度封装，无法满足定制化需求

浏览器端webrtc内置于浏览器引擎内，相比native端可定制化能力差





2. WebRTC由实时通信渗透到低延迟直播场景, Web端RTC提出更高要求

3. H265、B帧、AAC, 媒体二次处理(超分, 虚拟背景,降噪等) ,定制化加密/Qos优化等能力在Native端逐渐普及落地

背景

4. W3C一系列Web媒体处理与实时通信标准正在演进，为实现浏览器端自定义RTC提供了可行性(webtransport,webocdecs,webAssembly,webGPU)

5. 跟进浏览器端RTC最新技术趋势，参与到W3C标准化社区中

02

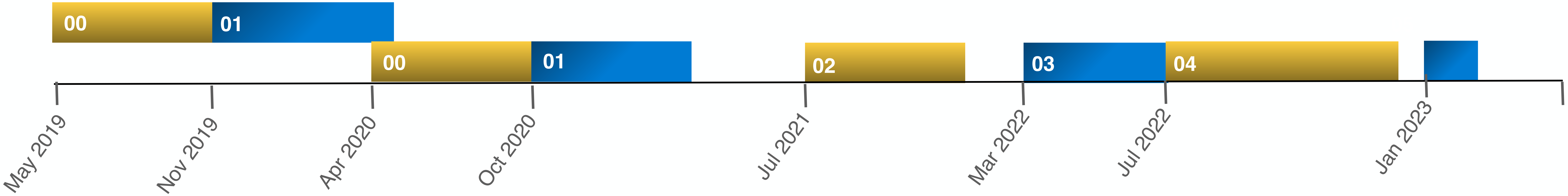
介绍

Webtransport,Webcodecs,Wasm

WebTransport 与IETF

draft-vvv-webtransport-overview

draft-ietf-webtrans-overview



目前有三个草案在推进，还无正式RFC

- 主草案： https://datatracker.ietf.org/doc/draft-ietf-webtrans-overview
- over http3：https://datatracker.ietf.org/doc/draft-ietf-webtrans-http3
- over http2：https://datatracker.ietf.org/doc/draft-ietf-webtrans-http2

WebTransport与W3C

2020年9月W3C创建了WebTransport工作组，由主要浏览器厂商出代表负责日常维护，与IETF密切合作定义了浏览器基于WebTransport通信客户端API，2021年5月第一版发布

只实现了IETF定义的WebTransport over http3协议

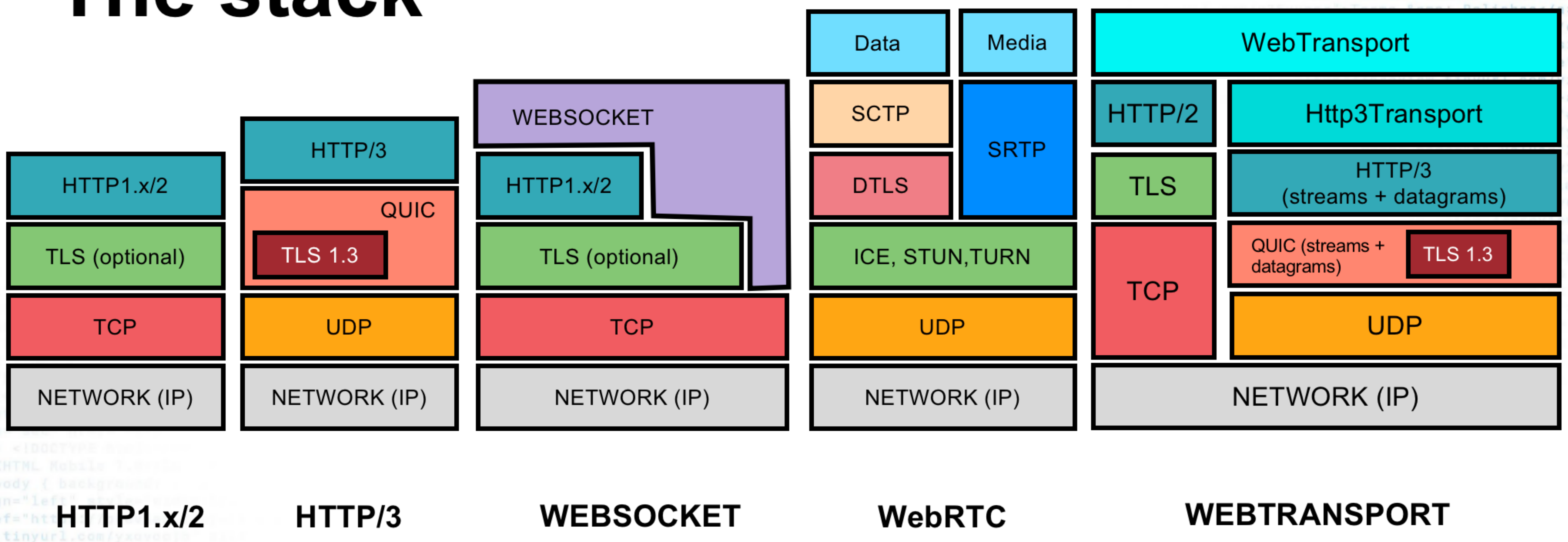
WEBTRANSPORT PUBLICATION HISTORY

2023-04-05	Working Draft
2023-01-18	Working Draft
2022-06-23	Working Draft
2022-03-11	Working Draft
2021-10-14	Working Draft
2021-05-04	First Public Working Draft

WebTransport 与其他协议

https://www.w3.org/TR/webtransport/

The stack



WebTransport	WebSocket	WebRTC
可靠&不可靠	可靠	可靠&不可靠
Client/Server架构	Client/Server架构	P2P，Client/Server
演进中	广泛部署	广泛部署
Low level api	Low level api	High level
• Multi stream • 无队首阻塞问题 • 低延迟 • 快速建立连接 • 支持连接迁移	Single stream 队首阻塞问题	依赖ICE，SDP Data channel over SCTP

*图片来源tpac 2022会议

WebTransport

What is exciting about WebTransport?

- A chance to unify the transport and API between
  - Video conferencing & telephony applications
  - Gaming
  - Low latency & live media delivery
- Looking like Http/3 to firewalls, proxies, network switches etc. can greatly facilitate its reach and robustness.
- Browser support gives you billions of addressable clients (in addition to native OS support).
- Datagram access in JavaScript ☺
- When combined with WebCodecs and WebAssembly, closes the gap between native and browser RTC applications.

*图片来源tpac 2022会议

WebCodecs

浏览器端直接/间接使用编解码能力的模块

WebAudio	只支持解码音频文件到PCM数据，不支持实时stream
MediaRecorder	用于录制音视频内容到文件，没有更高级的音视频参数配置，不适用低延迟编码场景
WebRTC	编解码能力内置于底层引擎，RTC能力的一环，无直接使用接口
HTMLMediaElement and Media Source Extensions	支持实时解码，但是依赖于容器格式，缺乏解码控制

2023-05-11	Working Draft
2023-04-27	Working Draft
2023-04-19	Working Draft
2023-04-03	Working Draft
2023-03-17	Working Draft
2023-03-13	Working Draft
2023-03-10	Working Draft
2023-02-09	Working Draft

由W3C Media工作组推进，浏览器端直接访问编解码器API标准，Javascript api实现音视频编解码能力

2021-05-11	Working Draft
2021-05-06	Working Draft
2021-05-05	Working Draft
2021-05-03	Working Draft
2021-04-30	Working Draft
2021-04-27	Working Draft
2021-04-08	First Public Working Draft

WebCodecs

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

音频

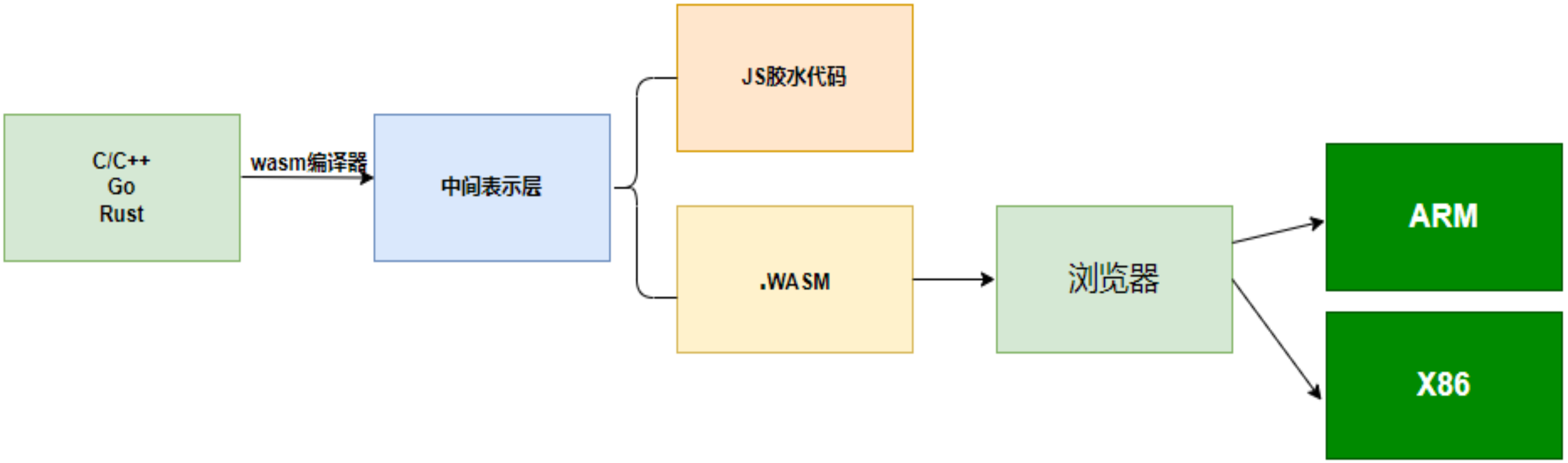
codec string	common name	public specification
flac	Flac	FLAC codec registration [WEBCODECS-FLAC-CODEC-REGISTRATION]
mp3	MP3	MP3 WebCodecs Registration [WEBCODECS-MP3-CODEC-REGISTRATION]
mp4a.*	AAC	AAC WebCodecs Registration [WEBCODECS-AAC-CODEC-REGISTRATION]
opus	Opus	Opus WebCodecs Registration [WEBCODECS-OPUS-CODEC-REGISTRATION]
vorbis	Vorbis	Vorbis WebCodecs Registration [WEBCODECS-VORBIS-CODEC-REGISTRATION]
ulaw	u-law PCM	u-law PCM WebCodecs Registration [WEBCODECS-ULAW-CODEC-REGISTRATION]
alaw	A-law PCM	A-law PCM WebCodecs Registration [WEBCODECS-ALAW-CODEC-REGISTRATION]
pcm-*	Linear PCM	Linear PCM WebCodecs Registration [WEBCODECS-PCM-CODEC-REGISTRATION]

视频

codec string	common name	specification
av01.*	AV1	AV1 codec registration [WEBCODECS-AV1-CODEC-REGISTRATION]
avc1.*, avc3.*	AVC / H.264	AVC (H.264) WebCodecs Registration [WEBCODECS-AVC-CODEC-REGISTRATION]
hev1.*, hvc1.*	HEVC / H.265	HEVC (H.265) WebCodecs Registration [WEBCODECS-HEVC-CODEC-REGISTRATION]
vp8	VP8	VP8 codec registration [WEBCODECS-VP8-CODEC-REGISTRATION]
vp09.*	VP9	VP9 codec registration [WEBCODECS-VP9-CODEC-REGISTRATION]

WebAssembly

1、一种可移植、安全、高效的底层二进制代码格式
2、能把 C、C++、Go、TS 等语言程序在浏览器中运行
3、更适合实现底层的计算密集型操作和图形渲染等任务，与Javascript互补



WebAssembly

已经成为W3C正式标准

2010 Alon Zakai开始开发Emscripten，尝试将WebAssembly编译为C++

2011 Alon Zakai发布Emscripten，并发表《Emscripten;an LLVM-to-JavaScript compiler》论文

2013 asm.js发布，并且成功将游戏引擎Unreal编译为asm.js移植到浏览器

2015 Firefox、Chrome、Safari和Edge浏览器开始合作开发WebAssembly，并且成立了W3C WebAssembly Community Group

2017 Firefox、Chrome、Safari和Edge浏览器相继支持WebAssembly，并合作发表《Bringing the Web up to Speed with WebAssembly》论文

2019 W3C发布WebAssembly正式标准

WebAssembly

得到浏览器广泛支持

WebAssembly - OTHER

Usage % of all users ?

Global 96.39%

WebAssembly or "wasm" is a new portable, size- and load-time-efficient format suitable for compilation to the web.

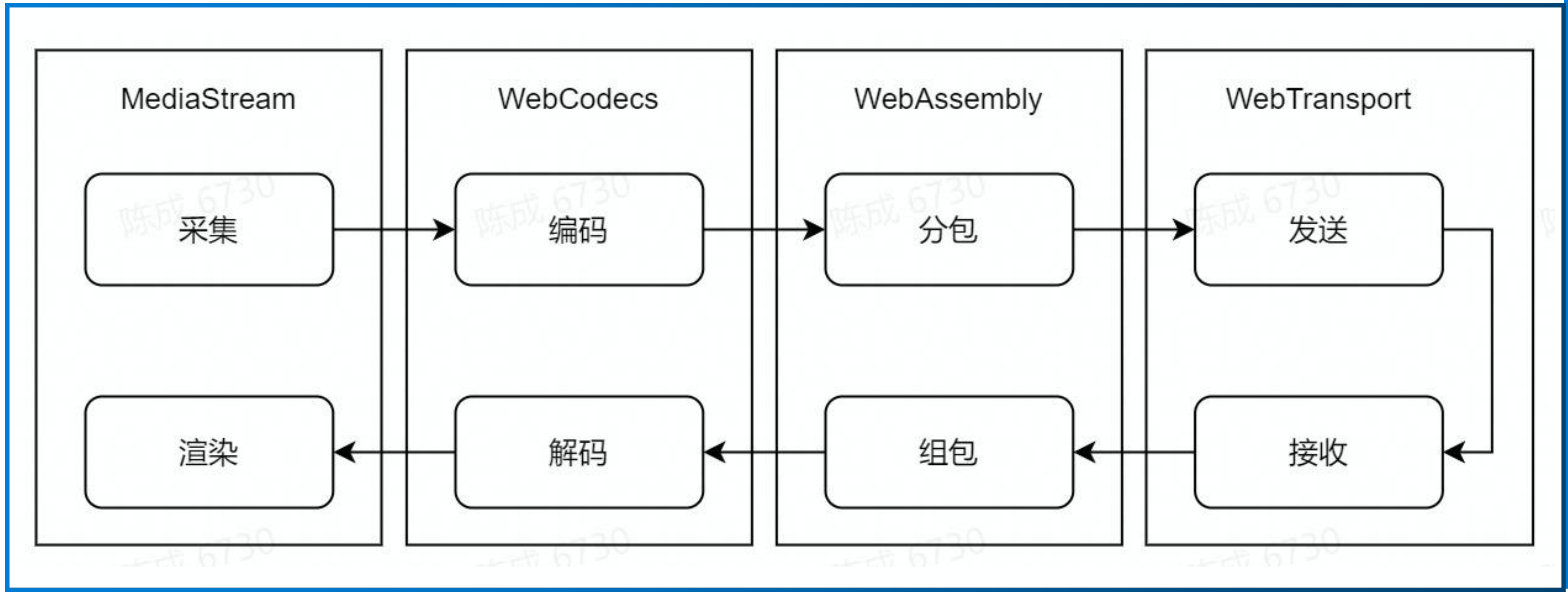
Current aligned | Usage relative | Date relative | Filtered | All

Chrome	Edge *	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile *	UC Browser for Android	Android Browser *	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
			2-46													
4-50	12-14		1 47-51	10-37												
2 51-56	3 15	3.1-10.1	4 52	2 38-43			3.2-10.3	4-6.4								
57-107	16-107	11-16.1	53-106	44-91	6-10		11-16.1	7.2-18.0		12-12.1		2.1-4.4.4				
108	108	16.2	107	92	11	108	16.2	19.0	all	72	13.4	108	107	13.1	13.18	1 2.5
109-111		16.3-TP	108-109				16.3									

Unbunding 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

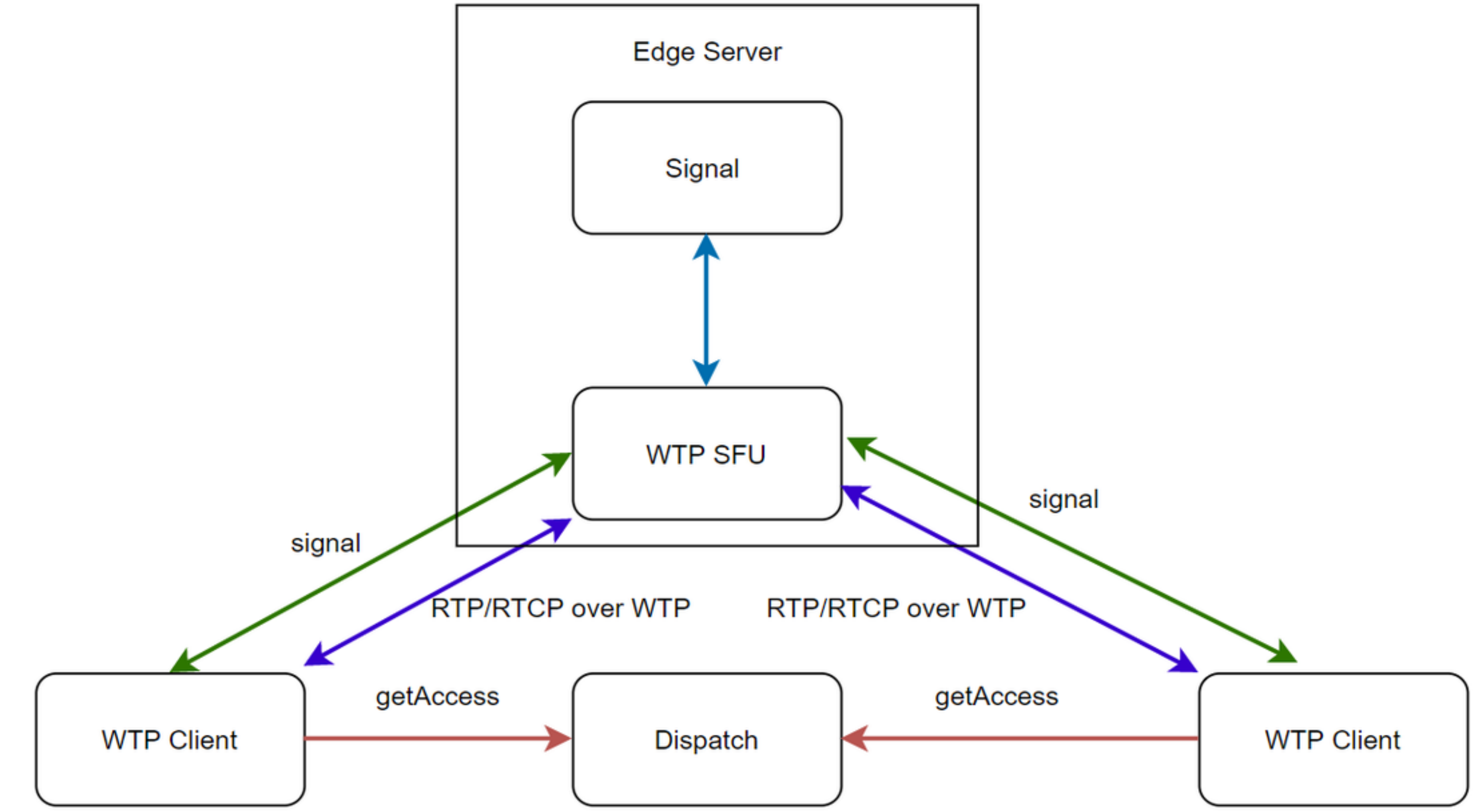


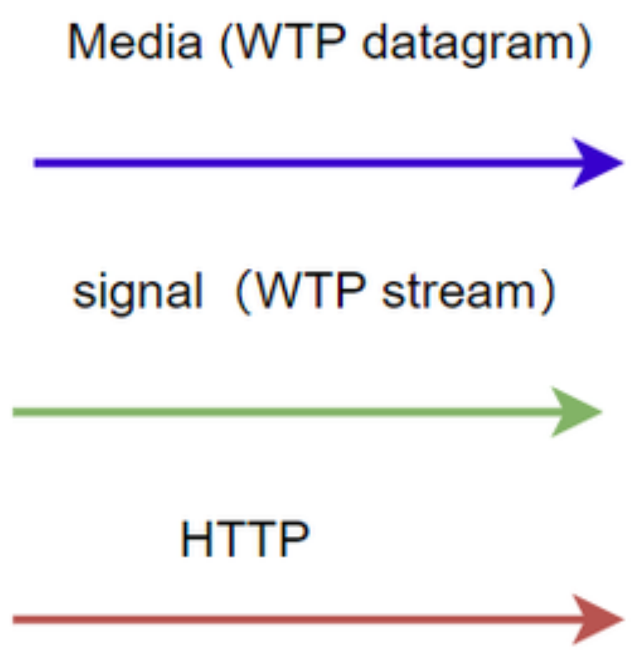
- ☑ 使用WebTransport的不可靠通道
- ☑ 需要自己实现QoS控制及分包组包逻辑
- ☑ 适用于Client-Server
- ☑ 可以更定制化的开发RTC功能

03

基于WWW的RTC应用实践

整体架构



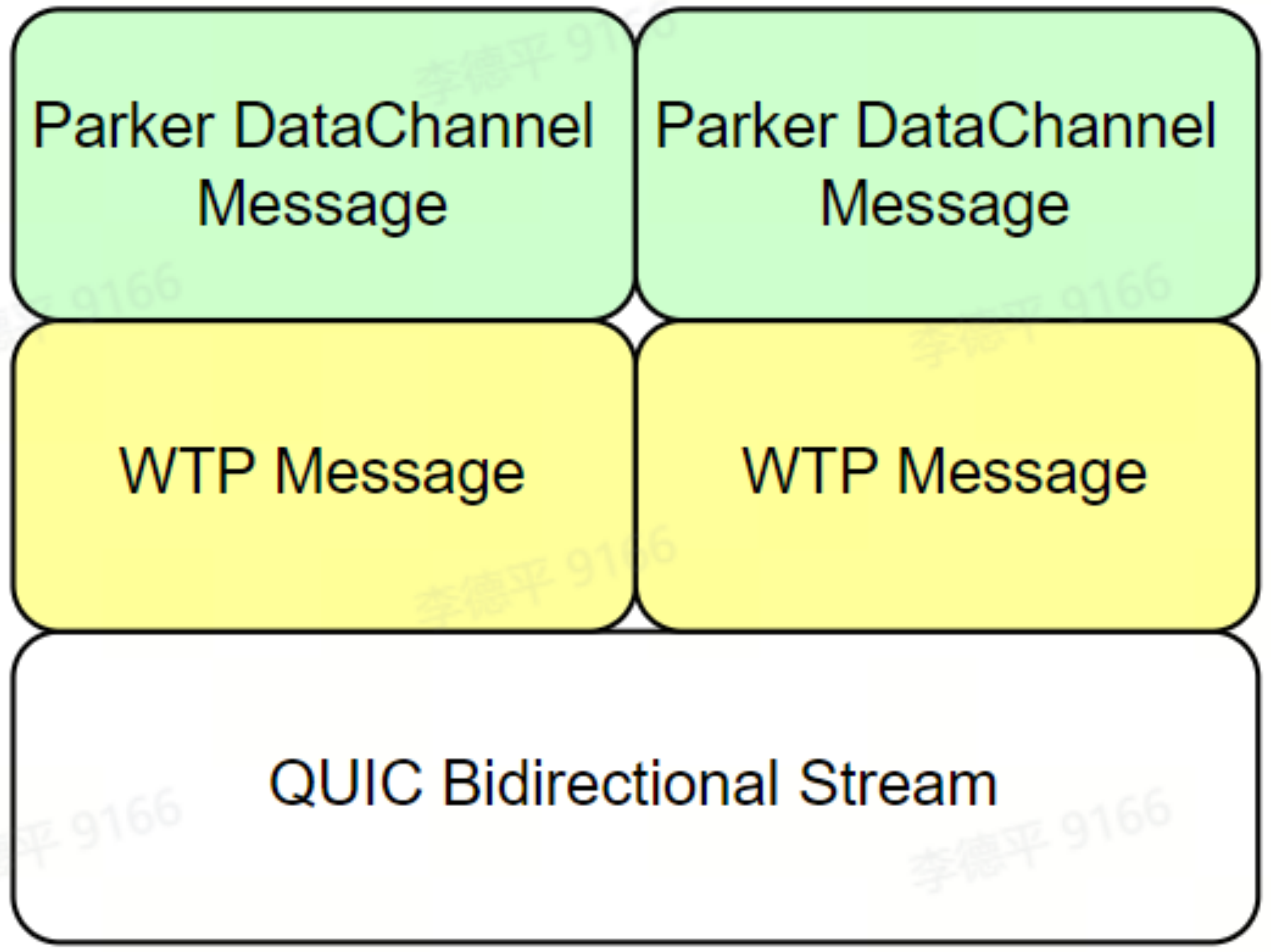


WTP(Webtransport) client 之间推拉流

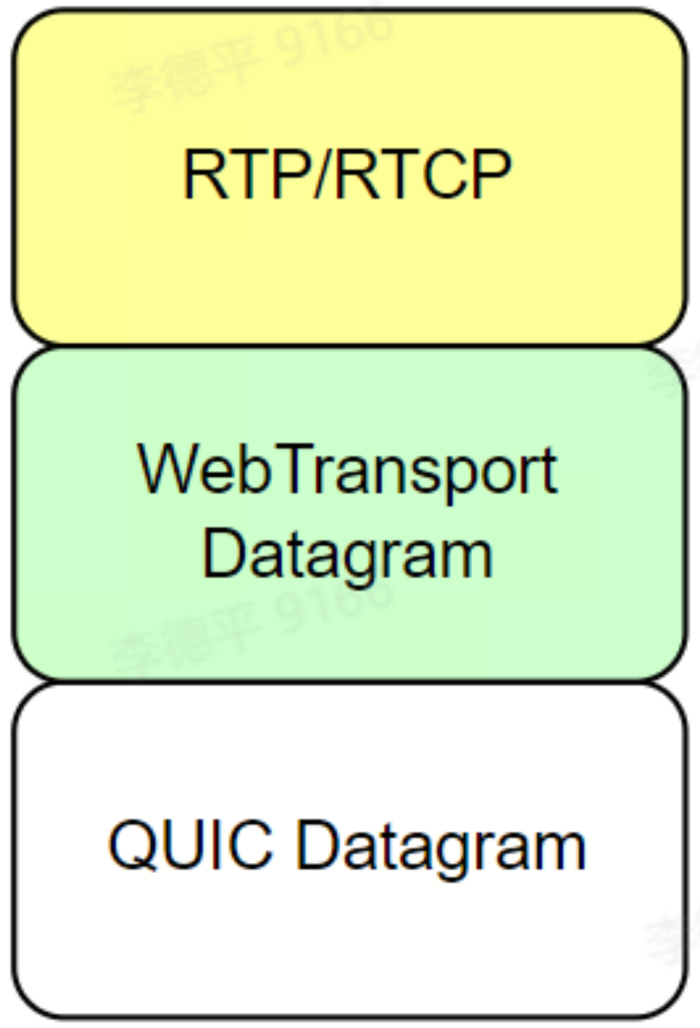
WTP client和Webrtc client 互相推拉流

传输规范

信令



媒体



SDP协商规范

遵循 rfc 8843和 quic sdp 草案

https://datatracker.ietf.org/doc/draft-dawkins-avtcore-sdp-rtp-quic/

弱网对抗

重传：音频，视频重传都走RTX方式，重传包用独立的Payload

带宽估计：由于目前Webtransport底层带宽估计结果没有反馈给上层，需要在wasm模块独立实现带宽估计，采用TCC带宽估计。

FEC/RED:音频可选支持RED/主动重传，视频可选FEC，后续考虑优化支持。

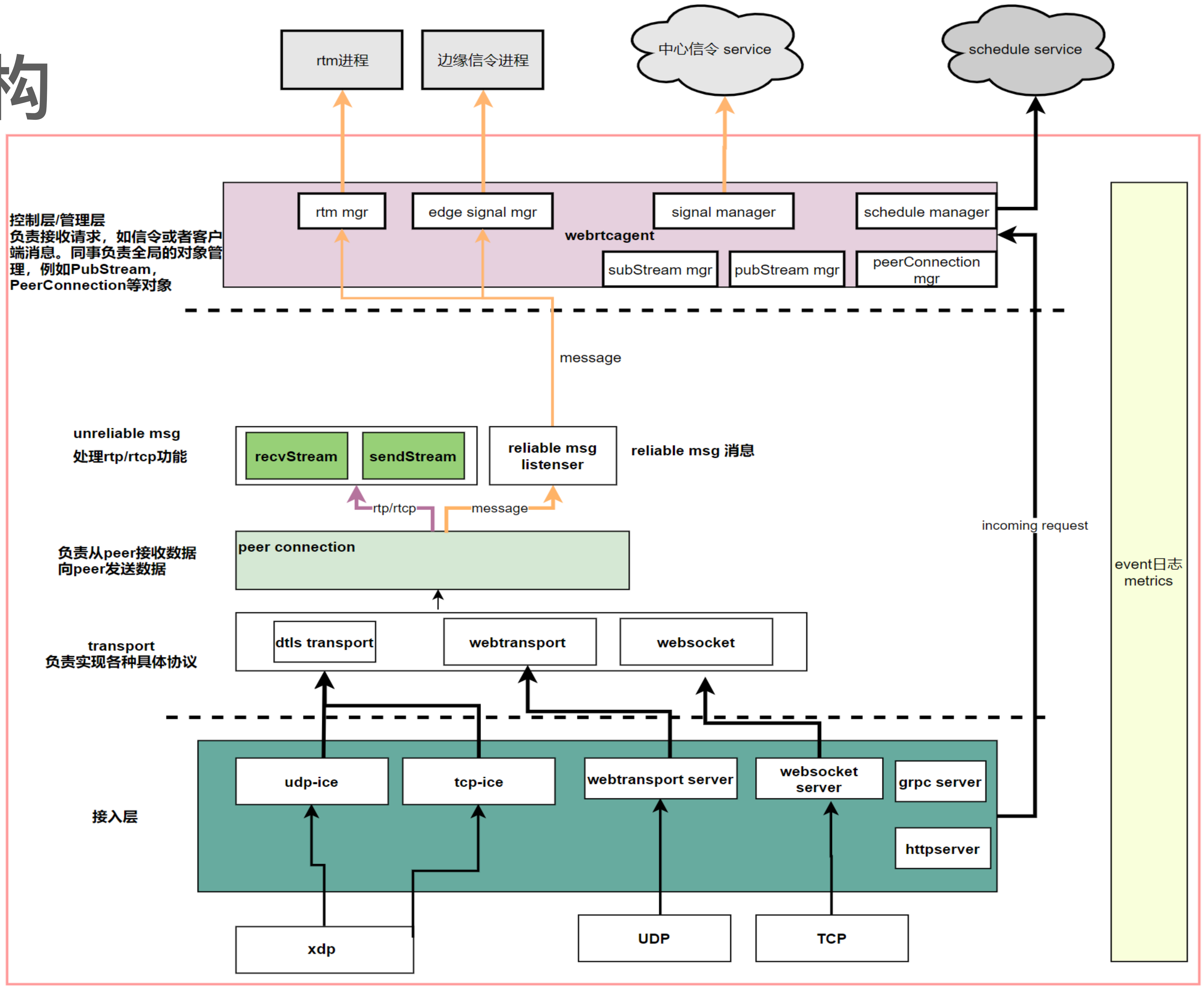
加密

Webtransport 已内置TLS1.3加密支持，属于连接级别，如需实现端到端加密，可在编码后再次加密。

一个推流SDP模板

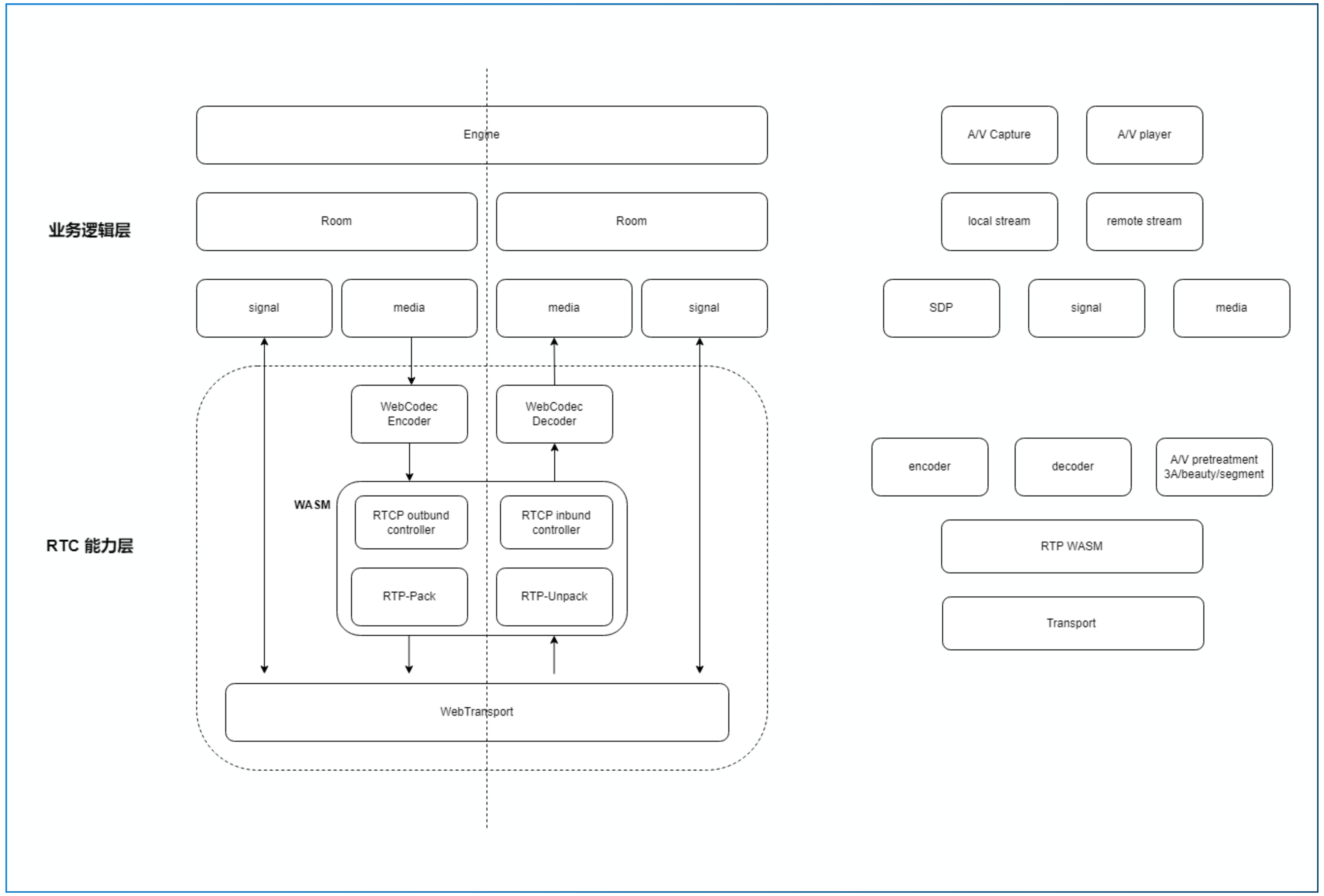
```
v=0
o=- 1086693717173763914 2 IN IP4 127.0.0.1
s=-
c=IN IP4 127.0.0.1
t=0 0
a=msid-semantic: WMS 6eafebab-a200-499c-8375-78d7711d8066
m=audio 9 QUIC/RTP/AVPF 111
a=sendonly
a=extmap:1 urn:ietf:params:rtp-hdrext:ssrc-audio-level
a=extmap:2 http://www.webrtc.org/experiments/rtp-hdrext/abs-send-time
a=extmap:4 urn:ietf:params:rtp-hdrext:sdes:mid
a=mid:0
a=msid:6eafebab-a200-499c-8375-78d7711d8066 3b22855d-03a3-4b40-b5b9-275b0fcb270d
a=ssrc:3941198760 cname:rReV4xeEwBc727Ch
a=rtpmap:111 opus/48000/2
a=rtcp-fb:111 transport-cc
a=rtcp-fb:111 nack
a=fmtp:111 useinbandfec=1
a=rtpmap:114 rtx/48000/2
a=fmtp:114 apt=111
m=video 9 QUIC/RTP/AVPF 108
a=sendonly
a=extmap:14 urn:ietf:params:rtp-hdrext:toffset
a=extmap:2 http://www.webrtc.org/experiments/rtp-hdrext/abs-send-time
a=extmap:13 urn:3gpp:video-orientation
a=extmap:3 http://www.ietf.org/id/draft-holmer-rmcat-transport-wide-cc-extensions-01
a=extmap:4 urn:ietf:params:rtp-hdrext:sdes:mid
a=extmap:10 urn:ietf:params:rtp-hdrext:sdes:rtp-stream-id
a=extmap:11 urn:ietf:params:rtp-hdrext:sdes:repaired-rtp-stream-id
a=mid:1
a=msid:6eafebab-a200-499c-8375-78d7711d8066 3b22855d-03a3-4b40-b5b9-275b0fcb3949
a=rtpmap:108 H264/90000
a=rtcp-fb:108 goog-remb
a=rtcp-fb:108 ccm fir
a=rtcp-fb:108 nack
a=rtcp-fb:108 nack pli
a=rtpmap:109 rtx/90000
a=fmtp:109 apt=108
a=ssrc-group:FID 636835441 737515072
a=ssrc:636835441 cname:rReV4xeEwBc727Ch
a=ssrc:737515072 cname:rReV4xeEwBc727Ch
```

服务端架构

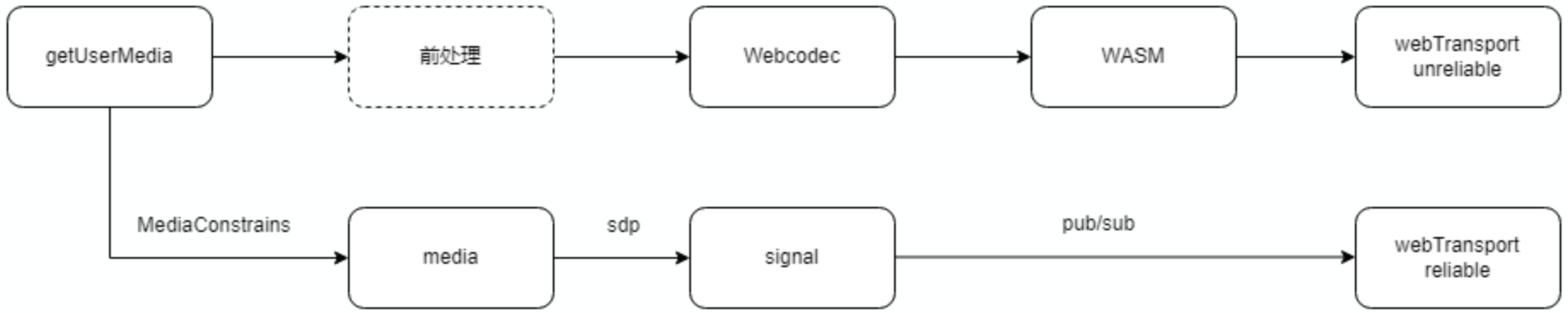


rtm进程
边缘信令进程
中心信令 service
schedule service
控制层/管理层
负责接收请求，如信令或者客户端消息。同事负责全局的对象管理，例如PubStream，PeerConnection等对象
rtm mgr
edge signal mgr
signal manager
schedule manager
webrtcagent
subStream mgr
pubStream mgr
peerConnection mgr
message
unreliable msg
处理rtp/rtcp功能
recvStream
sendStream
reliable msg listenser
reliable msg 消息
rtp/rtcp
message
incoming request
event日志
metrics
负责从peer接收数据
向peer发送数据
peer connection
transport
负责实现各种具体协议
dtls transport
webtransport
websocket
接入层
udp-ice
tcp-ice
webtransport server
websocket server
grpc server
httpserver
xdp
UDP
TCP

客户端SDK架构



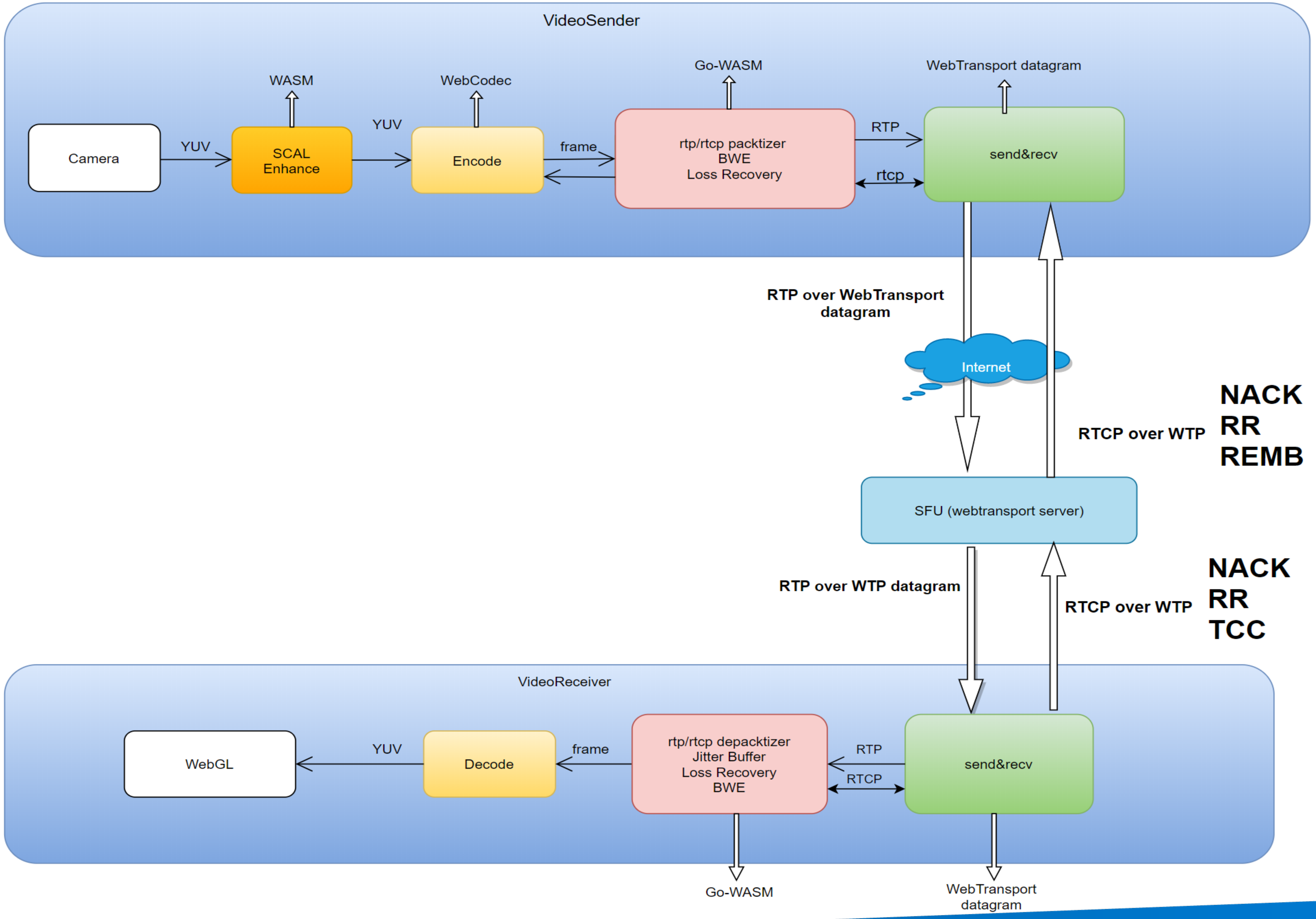
数据流转



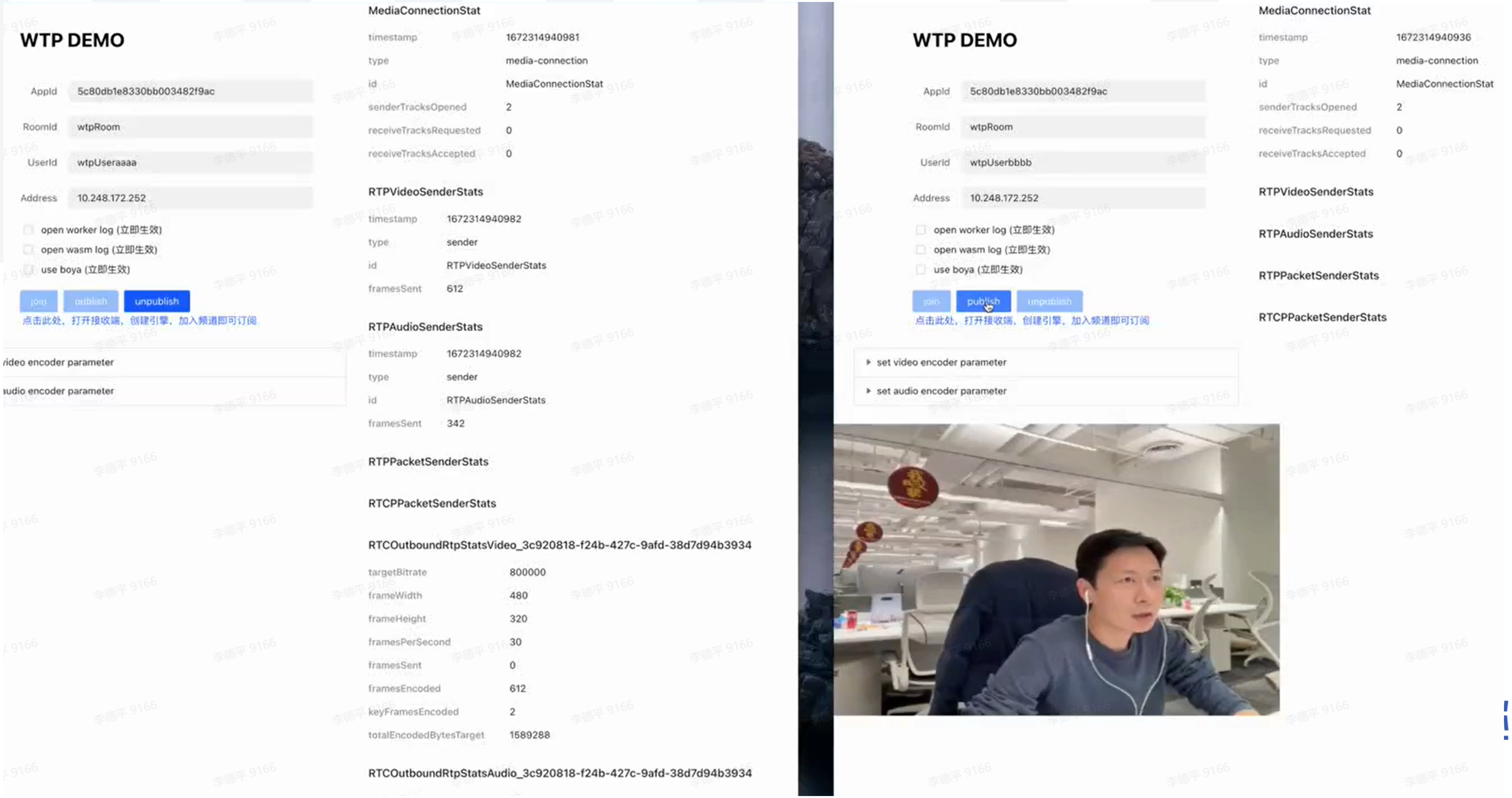
音视频pipeline

- ☑ 音频前处理采用webassembly技术，抽象成JS接口
- ☑ 音视频编解码及控制部分采用webcodecs js接口
- ☑ 媒体数据RTP/RTCP打包拆包，弱网对抗，带宽估计，jitter buffer等通过webassmbly抽象独立模块
- ☑ 信令和媒体传输分别采用WebTransport stream和datagram,共用一个connection

视频发布到接收流程


VideoSender
WASM
WebCodec
Go-WASM
WebTransport datagram
Camera
YUV
SCAL
Enhance
YUV
Encode
frame
rtp/rtcp packtizer
BWE
Loss Recovery
RTP
rtcp
send&recv
RTP over WebTransport
datagram
Internet
RTCP over WTP
NACK
RR
REMB
SFU (webtransport server)
RTP over WTP datagram
RTCP over WTP
NACK
RR
TCC
VideoReceiver
WebGL
YUV
Decode
frame
rtp/rtcp depacktizer
Jitter Buffer
Loss Recovery
BWE
RTP
RTCP
send&recv
Go-WASM
WebTransport
datagram

WTP DEMO


WTP DEMO
AppId 5c80db1e8330bb003482f9ac
RoomId wtpRoom
UserId wtpUseraaaa
Address 10.248.172.252
open worker log (立即生效)
open wasm log (立即生效)
use boya (立即生效)
join
publish
unpublish
点击此处，打开接收端，创建引擎，加入频道即可订阅
video encoder parameter
audio encoder parameter
MediaConnectionStat
timestamp 1672314940981
type media-connection
id MediaConnectionStat
senderTracksOpened 2
receiveTracksRequested 0
receiveTracksAccepted 0
RTPVideoSenderStats
timestamp 1672314940982
type sender
id RTPVideoSenderStats
framesSent 612
RTPAudioSenderStats
timestamp 1672314940982
type sender
id RTPAudioSenderStats
framesSent 342
RTPPacketSenderStats
RTCPPacketSenderStats
RTCOutboundRtpStatsVideo_3c920818-f24b-427c-9afd-38d7d94b3934
targetBitrate 800000
frameWidth 480
frameHeight 320
framesPerSecond 30
framesSent 0
framesEncoded 612
keyFramesEncoded 2
totalEncodedBytesTarget 1589288
RTCOutboundRtpStatsAudio_3c920818-f24b-427c-9afd-38d7d94b3934
WTP DEMO
AppId 5c80db1e8330bb003482f9ac
RoomId wtpRoom
UserId wtpUserbbbb
Address 10.248.172.252
open worker log (立即生效)
open wasm log (立即生效)
use boya (立即生效)
join
publish
unpublish
点击此处，打开接收端，创建引擎，加入频道即可订阅
set video encoder parameter
set audio encoder parameter
MediaConnectionStat
timestamp 1672314940936
type media-connection
id MediaConnectionStat
senderTracksOpened 2
receiveTracksRequested 0
receiveTracksAccepted 0
RTPVideoSenderStats
RTPAudioSenderStats
RTPPacketSenderStats
RTCPPacketSenderStats

方案总结

01 方案优点

1、音视频弱网对抗层面可控制一部分，比如增加RS-FEC，RED等

2、音视频前处理更灵活，借助wasm，可以自定义音视频前后处理

3、编解码控制更灵活，可独立做带宽分配，可使用SVC(AV1)，H265

4、传输层统一，可灵活控制加密方案

02 难点/问题

1、媒体引擎音视频前后处理，媒体Qos部分需要借助wasm自己实现，存在一定技术壁垒

2、传输层Webtransport 针对实时RTC场景支持不够完善，弱网对抗效果差于webrtc

3、如何在web上更高效处理视频数据

04

标准化进展

(webtransport,webcodecs)

How toget involved？

W3C会员单位，有PR权限

- ❑ 订阅W3C webtransport工作组github和邮件列表，参与API讨论/定义
- ❑ 邮件列表：public-webtransport@w3.org
- ❑ github: https://github.com/w3c/webtransport
- ❑ 两周一次在线会议
- ❑ 订阅W3C webcodecs工作组github项目/Media工作组邮件，参与API讨论/定义
- ❑ public-media-wg@w3.org
- ❑ github: https://github.com/w3c/webcodecs

Webtransport 标准化

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

Challenges: Transport

- Many applications require not just encode/decode of media, but also transport.
- WebRTC's RTP transport is not directly accessible.
- RTCDataChannel (NewReno) and WebTransport (BBRv1) congestion control algorithms are not optimized for realtime communications.
  - In server -> client communications (e.g. cloud gaming), you can deploy more appropriate algorithms on the server without interoperability issues.
  - Where the client needs to send media with low-latency (to a server or another client), there is a problem.
  - Paper: https://www.netlab.tkk.fi/~jo/papers/epiq21-rtp-over-quic.pdf (https://dl.acm.org/doi/abs/10.1145/3488660.3493801)
  - Presentation (starts at Slide 14): https://datatracker.ietf.org/meeting/112/materials/slides-112-avtcore-ietf-112-avtcore-03

TPAC 2022

W3C

`congestionControl`, of type WebTransportCongestionControl, defaulting to `"default"`

Optionally specifies an application's preference for a congestion control algorithm tuned for either throughput or low-latency to be used when sending data over this connection. This is a hint to the user agent.

ISSUE 2

This configuration option is considered a feature at risk due to the lack of implementation in browsers of a congestion control algorithm, at the time of writing, that optimizes for low latency.

Webtransport 标准化

单连接上不同包类型优先级能力缺乏统一管理

1. 后台上传日志模块------low priority
2. 实时音视频上传 (可基于datagram/stream) ---------medium priority
3. 信令通道(control channel)----------high priority

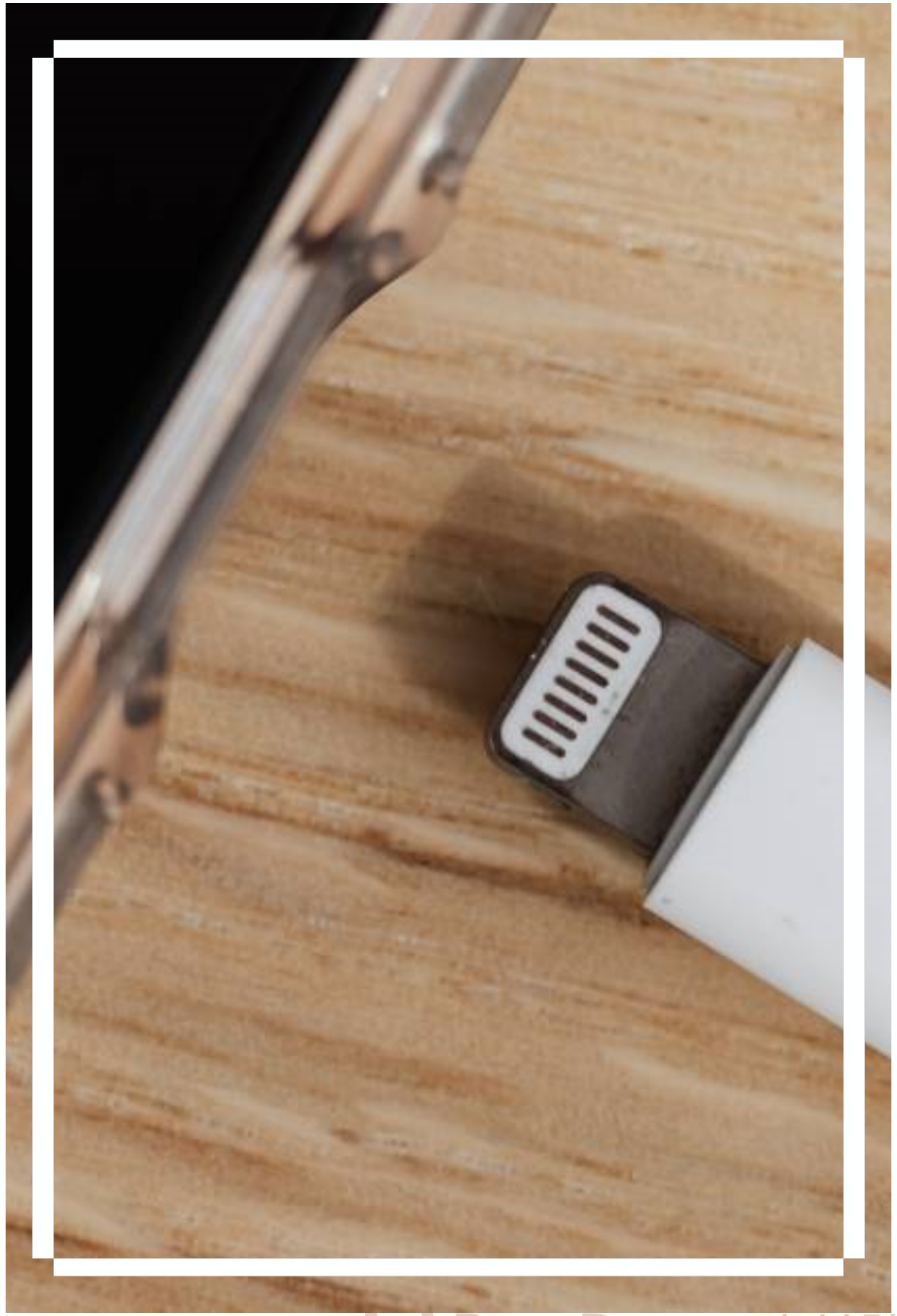
Webtransport 标准化

- webtransport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

https://github.com/w3c/webtransport/pull/421

Stats for congestion control and bandwidth estimation #21

- 目前浏览器端只实现了Webtransport over http3,不支持over http2，http3底层是基于quic, quic只支持udp,导致在不支持udp的网络无法回退到tcp



Webcodecs 标准化

01

opus编码默认60ms帧长不可配，不适用于RTC低延迟要求

02

opus编码还不支持inband-fec, 丢包反馈等特性

03

opus解码端不支持解码inbandfec包, 不支持opus解码器内置PLC能力

火山引擎RTC参与W3C标准

W3C webcodecs
webtransport
社区PR汇总

编号	Title	类别	状态
#540	link in README.md is unreachable	issue	fixed
#544	make fatal errors	discussion	fixed
#551	Add frameDuration attribute to OpusEncoderConfig	PR	merged
#555	fix a typo in 8.2.1	PR	merged
#594	针对opus packetlosspec PR给修改意见	PR	merged
#620	Webcodecs FLAC registry PR	PR	merged
#604	Webcodecs, #Information on encoder performance	discussion	discussion
#606	Is it necessary to add key frame check step for audio decoder	discussion	discussion
#451	Webtransport 包优先级意见	discussion	discussion
#649	webcodecs,增加 音频编码码率模式	PR	merged
#527	add opus inbandfec & dtx configure parameter to webcodecs audio encoder configure	PR	merged

一个PR例子

do we need support specify audio encode bitrateMode interface ? #649

Edit | New issue

Closed · bdrtc opened this issue on Mar 6 · 1 comment · Fixed by #657

bdrtc commented on Mar 6 · Member

some audio codecs(opus, HE AAC) support specify encoder bitrate mode,

according section-2.1.8 of opus rfc6716 and opus api doc ,
opus encoder support VBR and CBR mode, and vbr was used by default.

we can use bitrateMode exist in mediastream-recording,
`constant` correspond to CBR and `variable` correspond to VBR mode.

suggest to add `audioBitrateMode` in audio codec that support vbr/cbr mode, which specifes the BitrateMode that should be used to encode the audio track.

dalecurtis assigned **tguilbert-google** on Mar 11

tguilbert-google commented on Mar 11 · Member

This seems reasonable. I can edit the spec and codec registries in Q2.

Thanks for the suggestion!

👍 1

dalecurtis added registry need-definition and removed registry labels on Mar 17

Assignees
tguilbert-google

Labels
need-definition

Projects
None yet

Milestone
No milestone

Development
Successfully merging a pull request may close this issue.
support specify audio encode bitrateMode interf...
bdrtc/webcodecs

Notifications | Customize
Unsubscribe
You're receiving notifications because you're watching this repository.

3 participants

一个PR例子

support specify audio encode bitrateMode interface #657

Merged Djuffin merged 3 commits into w3c:main from bdrtc:649-support-specify-audio-encode-bitrateMode-interface on Apr 19

Conversation 12 | Commits 3 | Checks 16 | Files changed 1

bdrtc commented on Mar 27 • edited by pr-preview bot — Member

fixes #649

Preview | Diff

support specify audio encode bitrateMode interface ✓ 55094ea

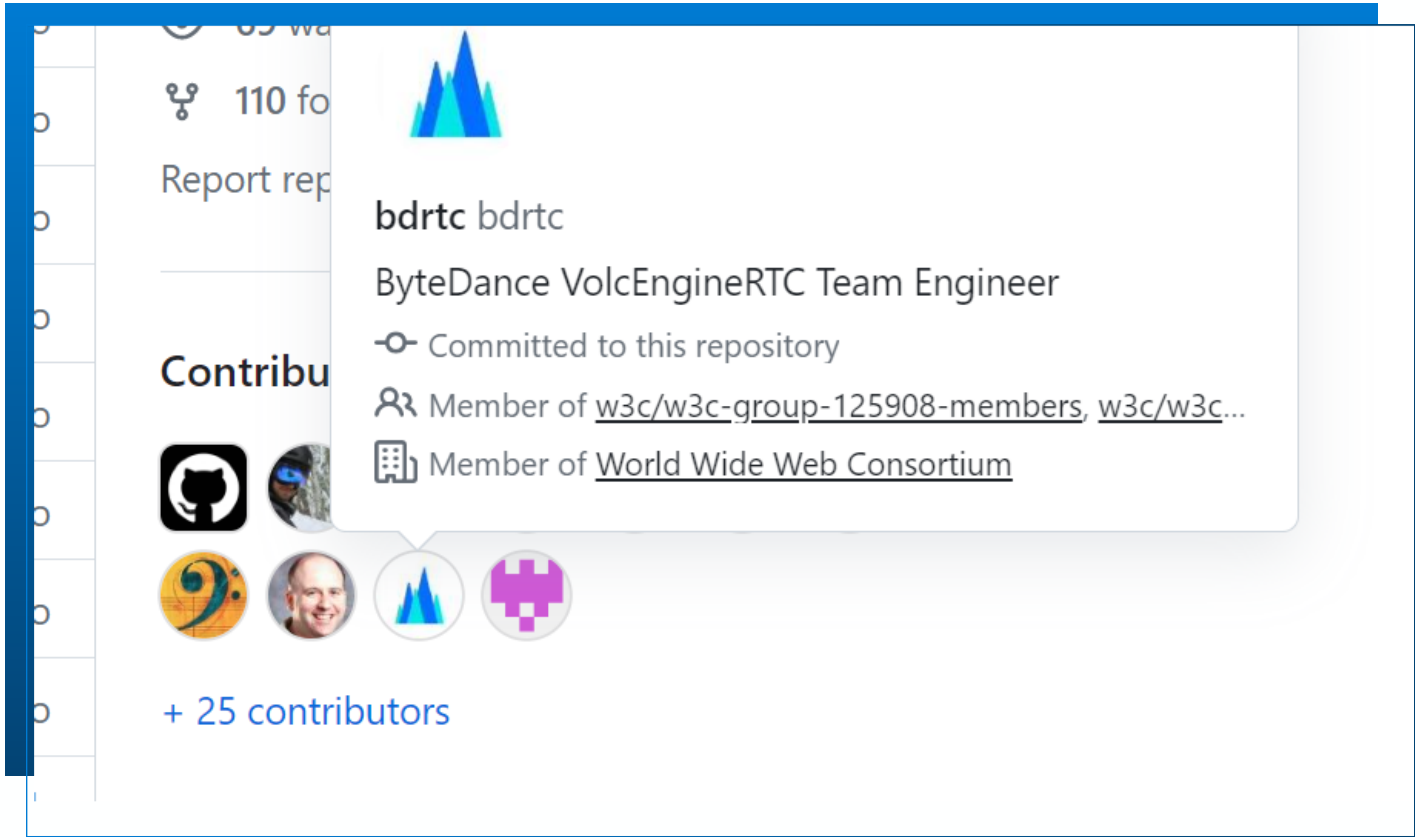
Reviewers
chrisn
tguilbert-google

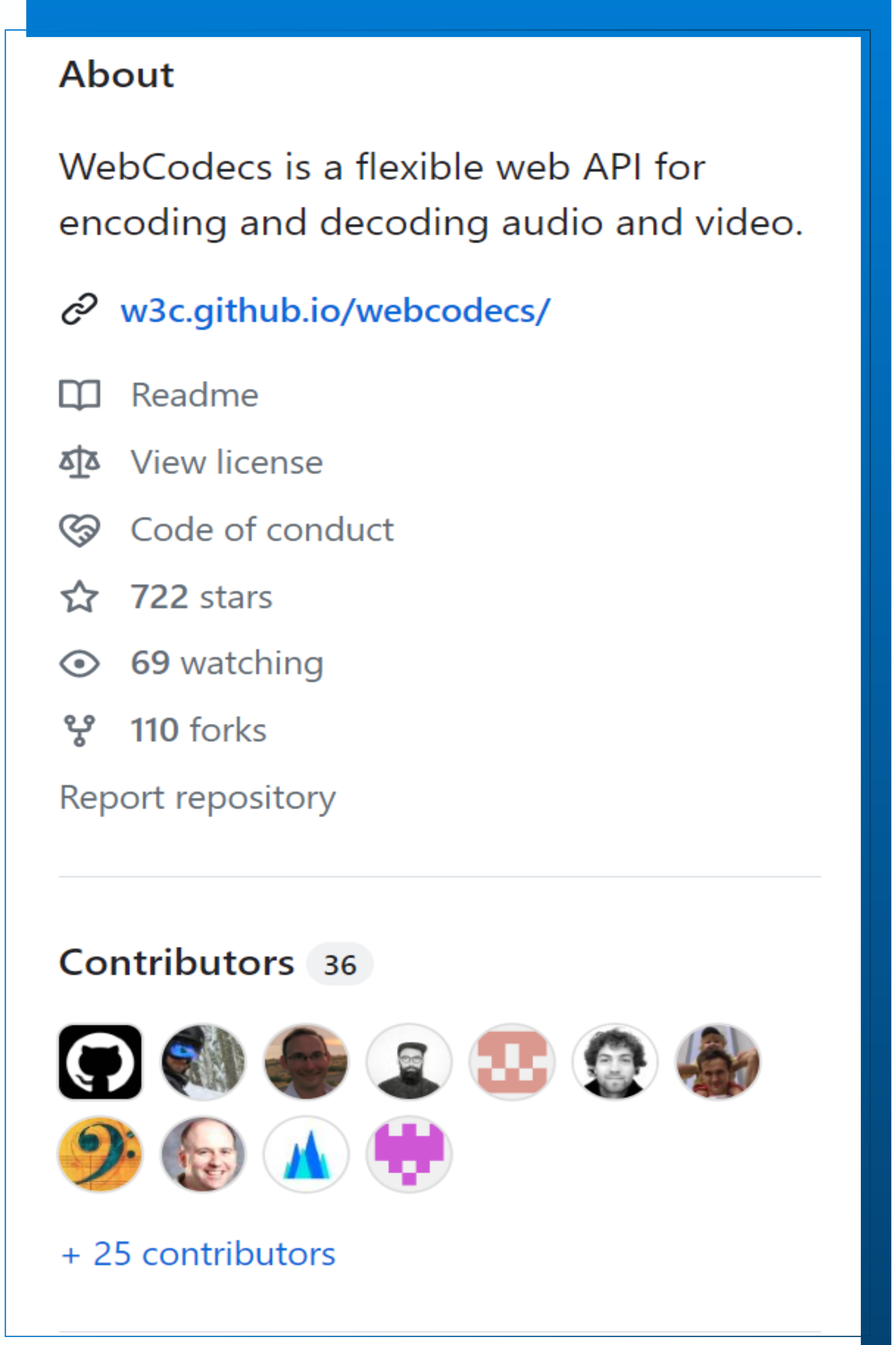
Assignees
No one assigned

Labels
None yet

ByteDance字节跳动

火山引擎RTC参与W3C标准





火山引擎RTC参与W3C标准

W3C WebRTC 工作组有多个草案，包括 webrtc-pc, webrtc-extensions, webrtc-stats, webrtc-svc, webrtc-nv-use-case

提议描述	提议地址	状态
给webrtc-extension spec 提议，增加aes-256加密控制api	https://github.com/w3c/webrtc-extensions/issues/113	无回应
给webrtc-stats spec 提议增加视频卡顿统计接口	https://github.com/w3c/webrtc-stats/issues/695	已经采纳，PR完成
增加js api控制opus编码码率	https://github.com/w3c/webrtc-extensions/issues/117	默认已经可工作
增加视频丢帧统计到发送端	https://github.com/w3c/webrtc-stats/issues/705	
need API to control audio nack & audio RTX	https://github.com/w3c/webrtc-extensions/issues/119	Nov 2022 Interim
支持开启音频DTX	https://github.com/w3c/webrtc-extensions/issues/120	Nov 2022 Interim
接收端丢包率stats	https://github.com/w3c/webrtc-stats/issues/706	
Support ICE Continuous Gathering flag in RTCConfiguration	https://github.com/w3c/webrtc-extensions/issues/121	大量讨论，无结论
Webrtc 浏览器端增加native log callback	https://github.com/w3c/webrtc-extensions/issues/124	有讨论，无结论
use case:disable A/V sync for remote control or online gaming	https://github.com/w3c/webrtc-nv-use-cases/issues/78	有讨论，无结论
关闭硬件编码	https://github.com/w3c/webrtc-extensions/issues/98#issuecomment-1335229695	大量讨论
Webrtc peerconnection 增加ipv6控制能力	https://github.com/w3c/webrtc-extensions/issues/138	大量讨论，无结论
Webrtc extention 修改rtcHeaderExtensions api	https://github.com/w3c/webrtc-extensions/issues/132	
Webrtc-stats 修改playoutDelay为jitterbufferTarget	https://github.com/w3c/webrtc-stats/pull/754	PR
Enabling RTCP XR(RRTR/DLRR) support for non-senders	https://github.com/w3c/webrtc-extensions/issues/165	Discussion

05

总结

总结

- Webtransport, Webcodecs， Webassembly 提供了浏览器端实现自定义RTC产品的可行性, 有深厚积累的公司可以尝试

- 基于Webtransport， Webcodecs, Webassembly的实时音视频通信产品目前还处于发展阶段，距离实现具备Webrtc能力的产品还需要在标准化和实现层面继续打磨

- Webtransport, Webcodecs 相关标准还处于草案演进阶段，对于有技术积累的公司可以积极参与到标准化社区中，为W3C社区贡献力量

THANKS

ByteDance字节跳动