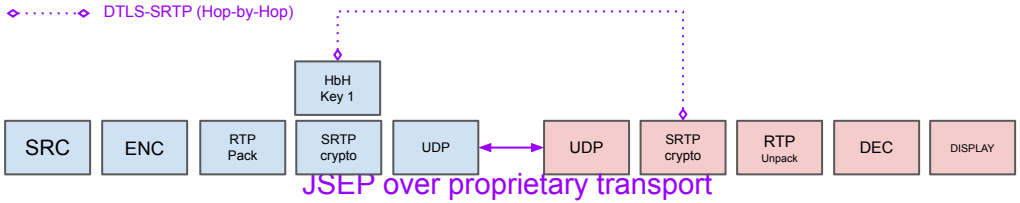
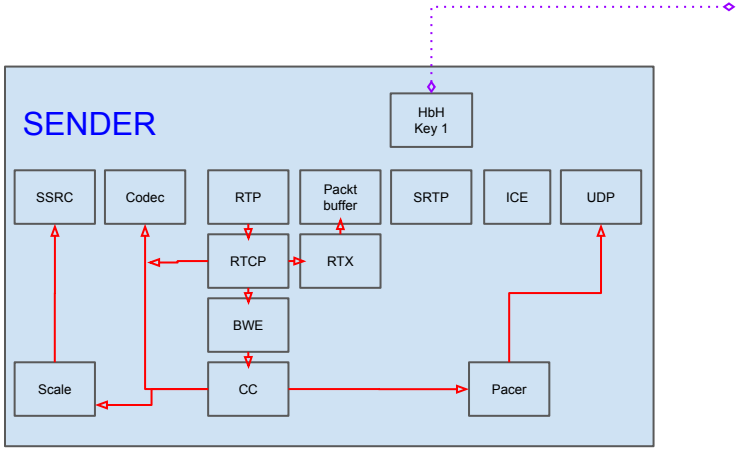
# E2EE over WebRTC
# The Big Picture

**IETF 110 - AVTCore / SFrame / WHIP**

*Dr Alex. Gouaillard & Sergio Murillo, CoSMo, Youenn Fablet, Apple*

# From RTP to WebRTC 1.0 ‘A’ (P2P)




**SRTP**

- DTLS-SRTP (3711)

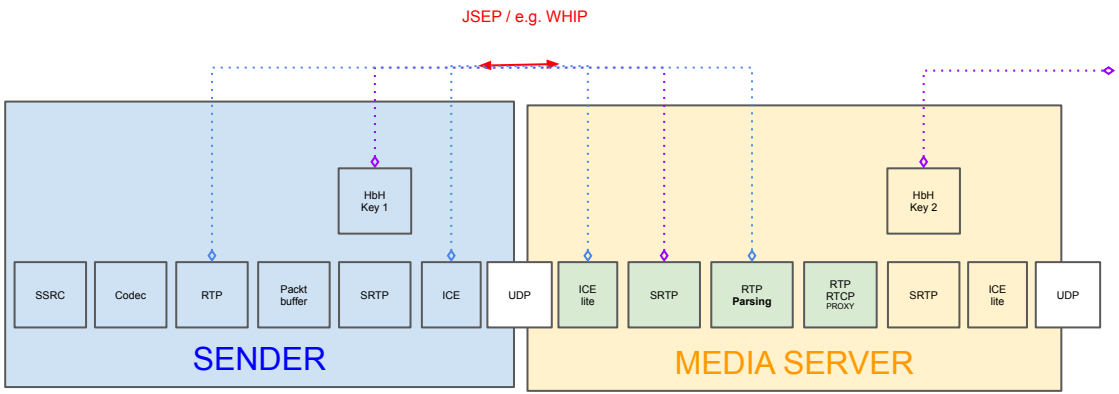
**WEBRTC**

- JSEP (8829)
- ICE (5243 => 8445)

# From RTP to WebRTC 1.0 ‘B’ (SFU)

ENCRYPTION

DTLS (Hop-by-Hop)



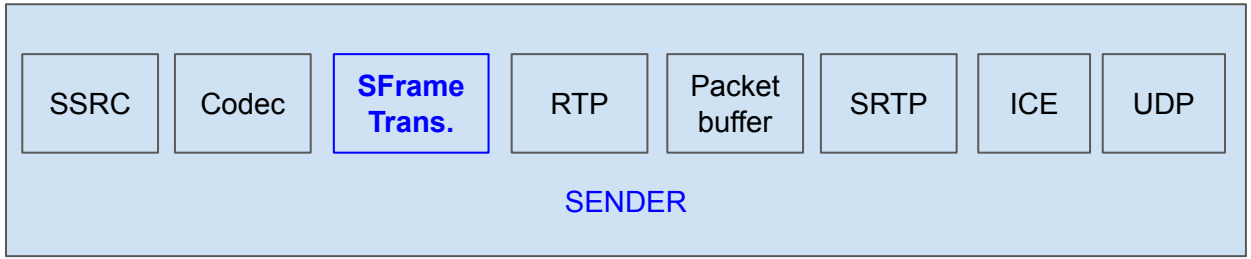

**WEBRTC**

- ICE (rfc5243 => rfc8445)
- trickle (rfc8838)
- ICE PAC (rfc8863)

# E2EE over WebRTC 1.0 ‘B’ (SFU)
# Step 1: “filter” between encoder and packetizer

ENCRYPTION

DTLS (Hop-by-Hop)
MLS (End-to-End)



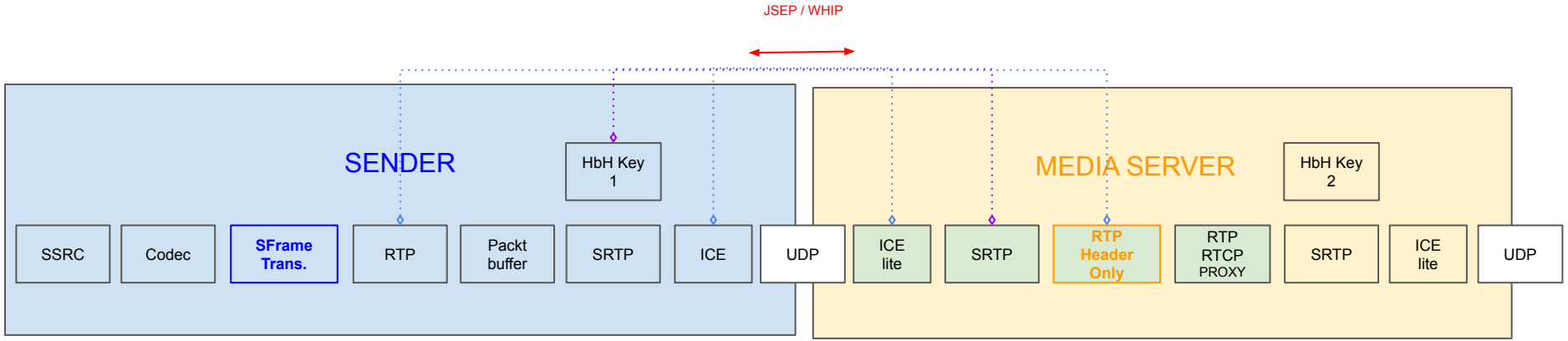

**RTP Payload ?**

- Generic ‘SFrame’ payload
- Codec as APT
-

Full Pres. by Sergio and Youenn

# E2EE over WebRTC 1.0 ‘B’ (SFU)
# Step 2: RTP Header Extension to the SFU rescue

ENCRYPTION
DTLS (Hop-by-Hop)
MLS (End-to-End)




FrameMarking not enough for SVC, DD as a candidate.

- AV1 ready
- Rtp codec agnostic ready
- Implementation feedback available

Likely needs a separate discussion.

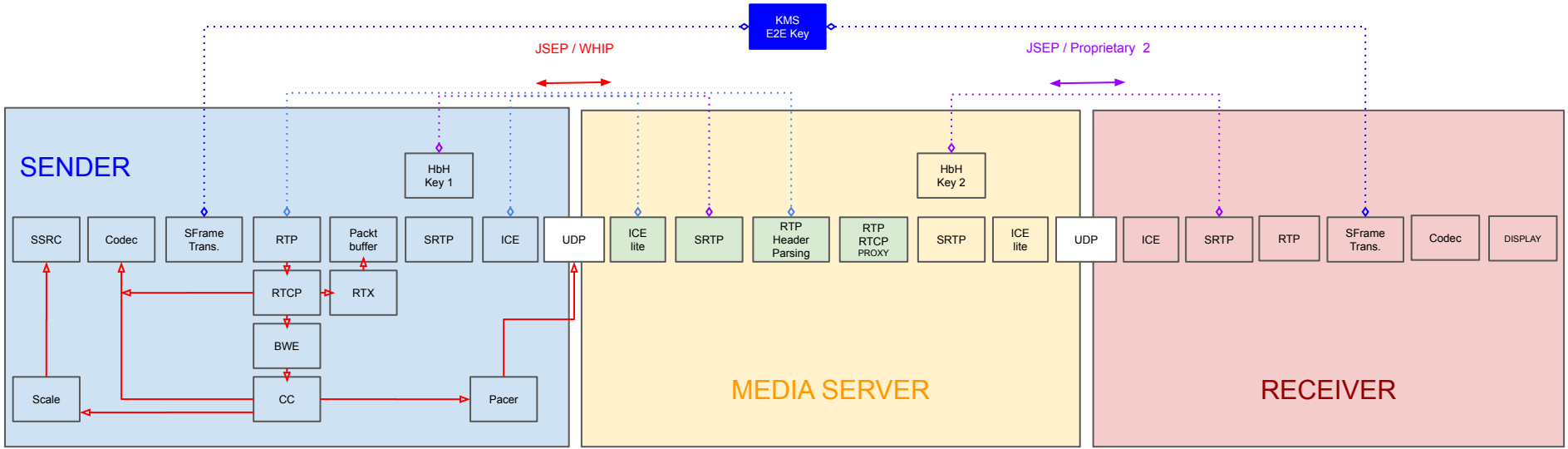
# E2EE over WebRTC 1.0 ‘B’ (SFU)

## Step 3: External Key Exchange w MLS (Richard B.)

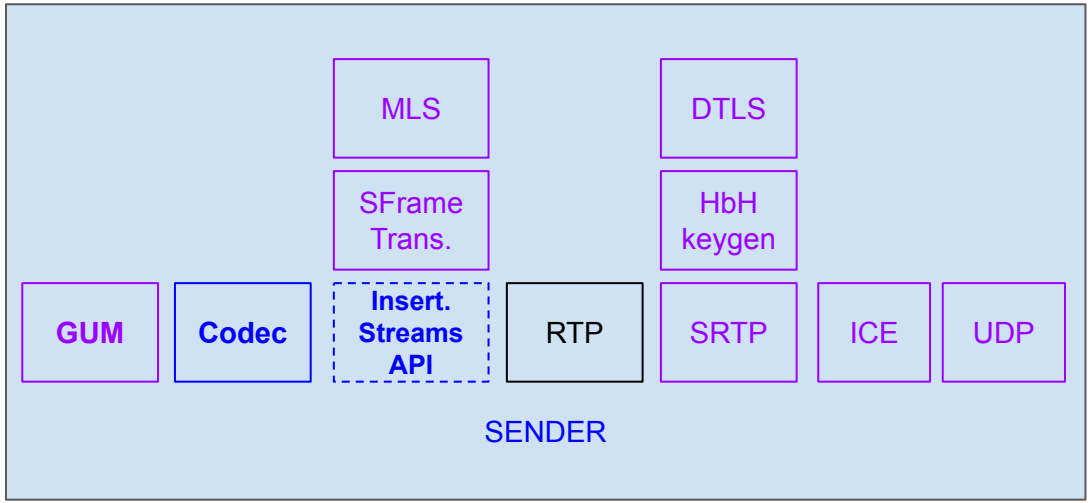
ENCRYPTION

DTLS (Hop-by-Hop)

MLS (End-to-End)



# E2EE over WebRTC 1.0 ‘B’ (SFU)
# Specific Web Trust/Threat model




**WEBRTC 1.0:** all secure

**WEBRTC NV:**

- Opening (Unsecure)
  - Media Creation / Raw Media
  - Media content
- Must keep Secure
  - HbH Key gen / exchange
  - HbH SRTP crypto
  - Ports / hardware (capture)

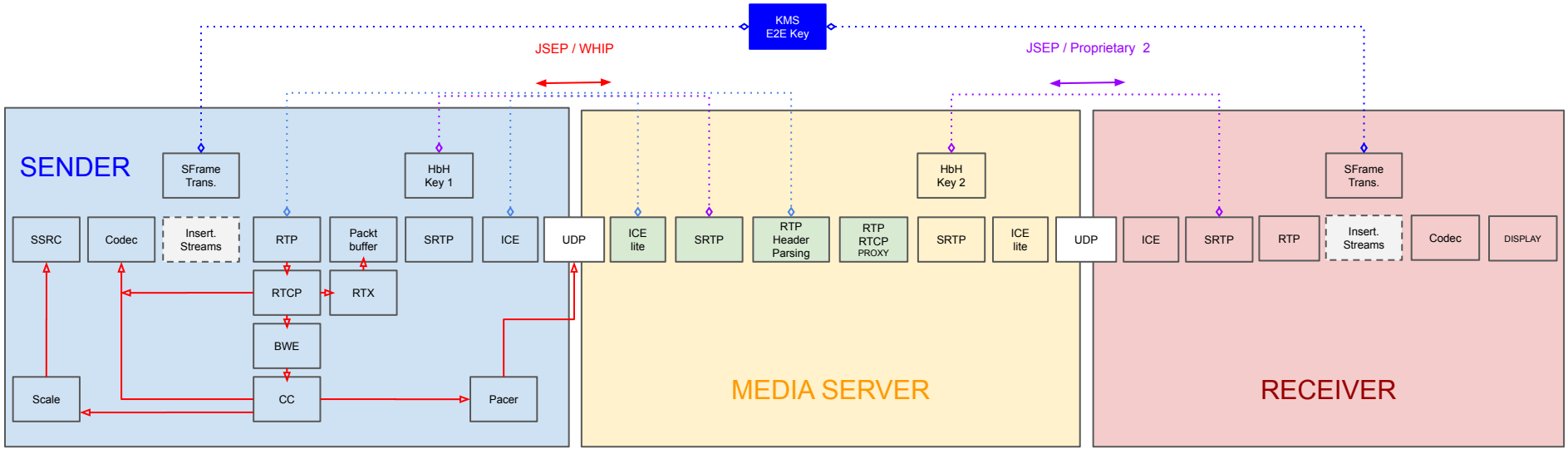
**E2EE:**

- Secure crypto
  - Unsecure Insert. Stream JS API
  - Secure SFrame Transform
- Secure Key gen (KMS)
- Secure Key retrieval / Exchange
  - Secure Key retr. Transform (MLS)

# E2EE over WebRTC 1.0 ‘B’ (SFU)

ENCRYPTION

DTLS (Hop-by-Hop)
MLS (End-to-End)



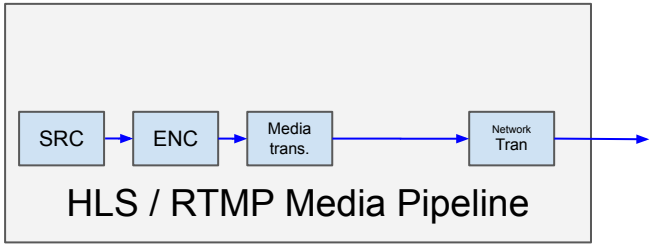
# E2EE for WebRTC 1.0/NV ‘C’ (SFU + multirez)

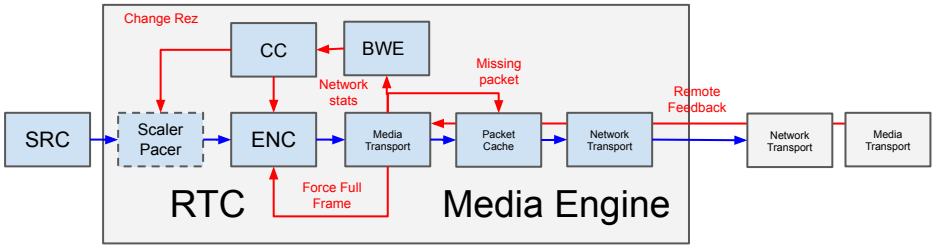
| | |
|---|---|
| AVPF layered | rfc8082 |
| BUNDLE - Multiple streams | rfc8108 |
| FlexFEC | rfc8627 |
| SIMULCAST | rfc8853 |
| Multiple Stream Types | rfc8860 |

```
   In the case of a video codec supporting spatial scalability, each
   spatial layer MUST be split in its own frame by the application
   before passing it to the packetizer.

   The marker bit of each RTP packet in a frame MUST be set according to
   the audio and video profiles specified in [RFC3551].

   The spatial layer frames are sent in ascending order, with the same
   RTP timestamp, and only the last RTP packet of the last spatial layer
   frame will have the marker bit set to 1.
```

# ANNEXES

# From Media Pipeline to (RTP) Media Engine




**RTP / RTCP (`RFC7656`)**

SR/RR:

NACK:

`RFC4588` RTX:

`RFC5109` FEC:

`RFC7656` RED:

PLI:

FIR:

REMB:

TMMBR: