E2EE for conference calls In WebRTC

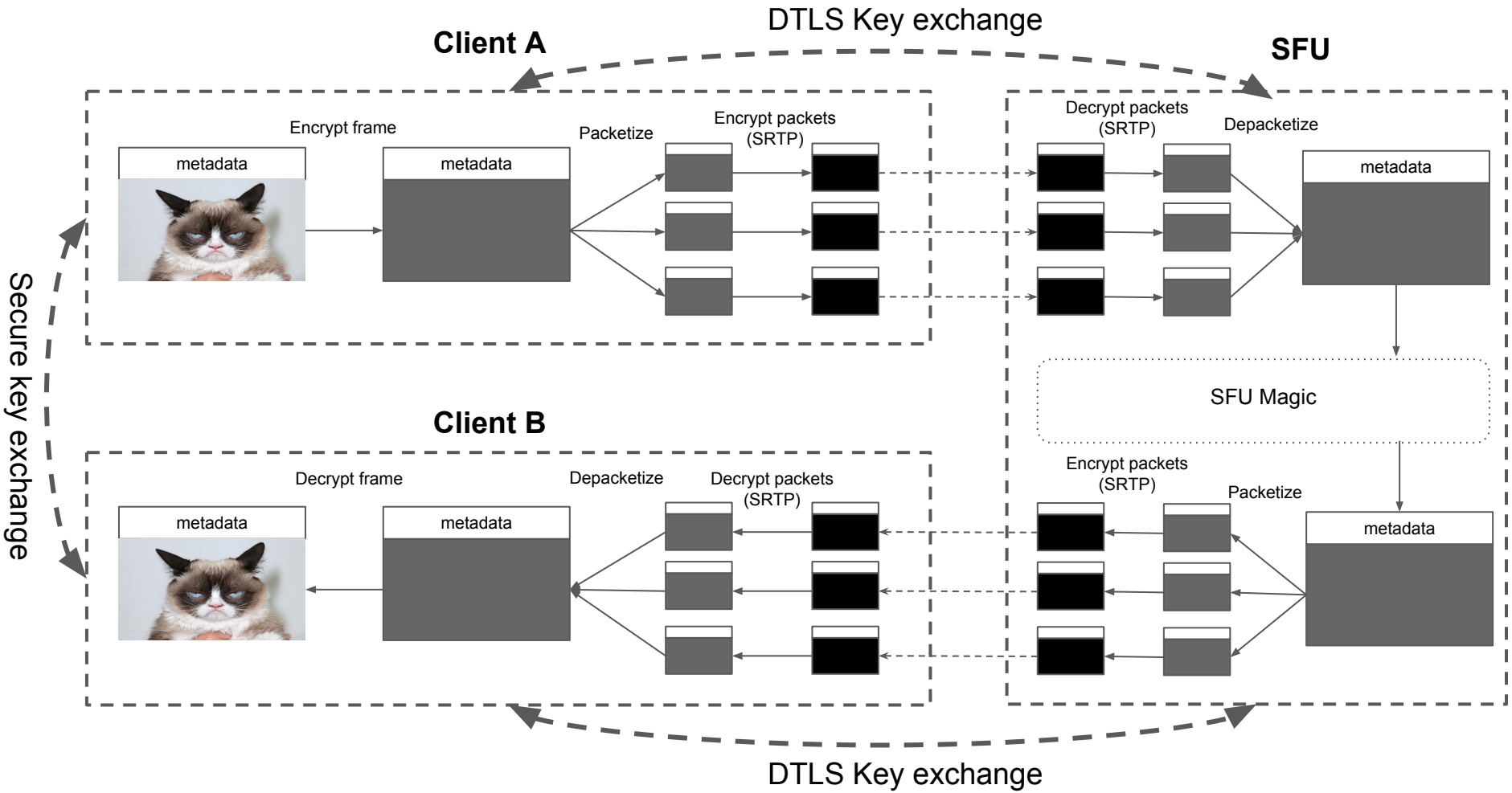
A use case for WebRTC NV

Requirements

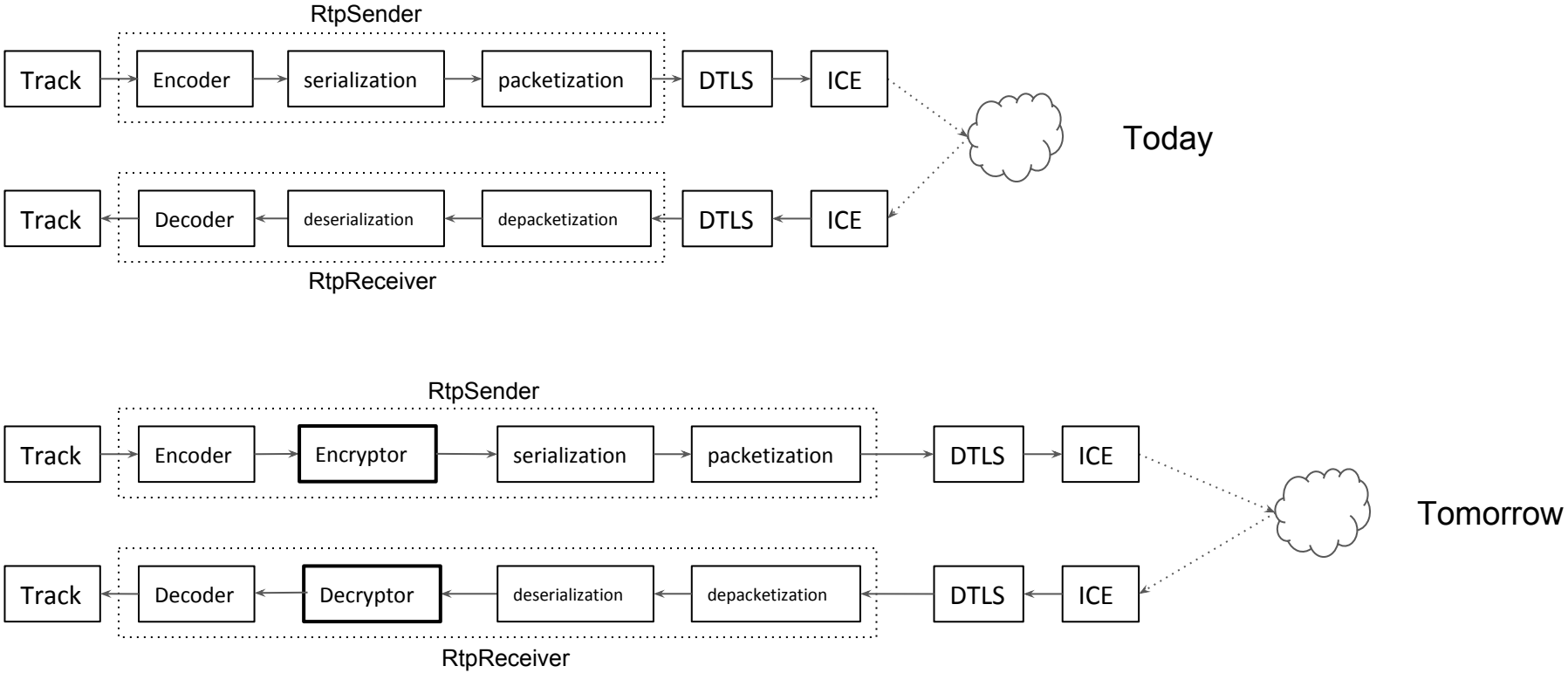
- Encryption/Decryption is done by the application
- Fully trusted applications
  - Partially and non trusted applications not in scope
- Partially trusted SFU
  - Can read only metadata but not contents
- Only Video/Audio
  - Text not in scope

Frame encryption

- RTP packets are encrypted today between clients and SFUs using SRTP (outer)
- A new layer of encryption is required between the clients end to end (inner)
- The new outer encryption layer is per Video frame instead of RTP packets (PERC and variations)
  - Saves bandwidth (Extra IV and MAC per frame)
  - Simpler to implement
  - Not RTP specific, so it can work with other transports
- The encryption protocol details will be discussed at IETF 104


DTLS Key exchange
Client A
SFU
Encrypt frame
Packetize
Encrypt packets
(SRTP)
metadata
metadata
Decrypt packets
(SRTP)
Depacketize
metadata
SFU Magic
Client B
Decrypt frame
Depacketize
Decrypt packets
(SRTP)
metadata
metadata
Encrypt packets
(SRTP)
Packetize
metadata
Secure key exchange
DTLS Key exchange

API changes


RtpSender
Track
Encoder
serialization
packetization
DTLS
ICE
Today
Track
Decoder
deserialization
depacketization
DTLS
ICE
RtpReceiver
RtpSender
Track
Encoder
Encryptor
serialization
packetization
DTLS
ICE
Tomorrow
Track
Decoder
Decryptor
deserialization
depacketization
DTLS
ICE
RtpReceiver

API changes

```
interface FrameEncryptor {
    boolean encrypt(mediaType, ssrc, encodedFrame,
                    additionalData)
}
rtpSender.setFrameEncryptor(...)


interface FrameDecryptor {
    boolean decrypt(mediaType, ssrc, encryptedFrame,
                    additionalData)
}
rtpReceiver.setFrameDecryptor(...)
```

Generic APIs for Video Frame processing cases ?

```
interface OutgoingFrameProcessor {
    boolean process(mediaType, ssrc, encodedFrame,
                    metadata)
}
rtpSender.setFrameProcessor(...)


interface IncomingFrameProcessor {
    boolean process(mediaType, ssrc, encodedFrame,
                    metadata)
}
rtpReceiver.setFrameProcessor(...)
```

Questions for the WG

- Do we have consensus to cover this use case?
- If so,
  - Crypto specific vs generic APIs ?