# SFrame in browsers

# SFrame Implementation in browsers

- Can be implemented in native or as JS using insertable streams
- Advantages of a native implementation
  - JS does not need raw access to media content
    - MediaStreamTrack & RTC constructs are sufficient
  - JS does not need raw access to encryption material
    - CryptoKey is sufficient
  - Browsers can implement further protection
    - Browsers keep control of supported algorithms
    - Out-of-render-process media encryption/decryption, isolated streams…

- W3C WebRTC WG Consensus
  - Define a native SFrame transform that integrates with WebRTC audio/video streams

# Native web SFrame transform

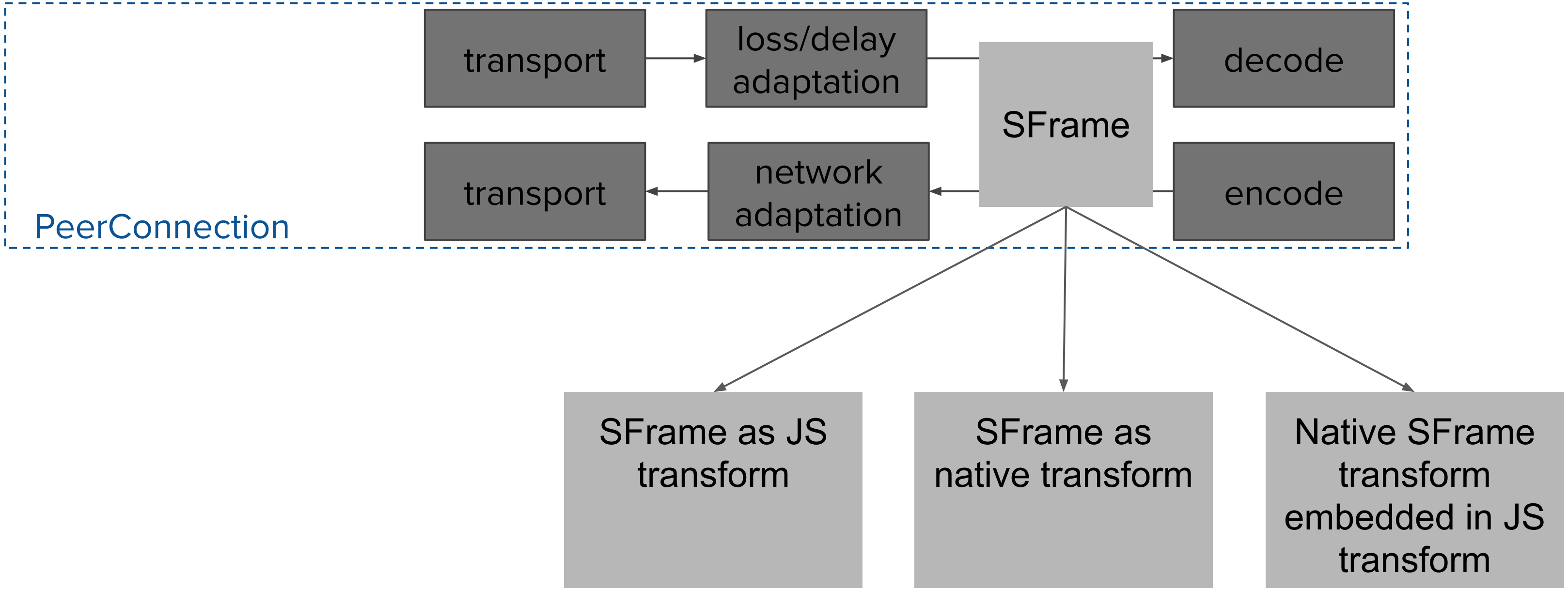
- Can be used standalone in WebRTC
    - JavaScript provides the encryption keys
- Can be used with a native key manager
    - Key manager generates encryption keys
    - JavaScript hands over the keys to SFrame transforms
        - Key material does not need to be exposed to JavaScript
    - Native Key Manager standardization can be done in parallel
- Integration with WebRTC constructs
    - RTCRtpSender/RTCRtpReceiver
- Does not preclude use outside of WebRTC

# SFrame packetization

- SFrame is not working with existing SFUs and existing browsers
  - SFrame is not compatible with all packetizations in use
    - Video packetization in particular
- Need for a generic packetization with non E2E encrypted frame metadata
  - Alternative: SFrame post-transform to adapt to codec-specific packetizations
- Might be useful outside SFrame
  - JS insertable streams might not generate valid media content

- Anyone working on it or interested in helping start that effort?

# SFrame use in browsers



Granularity of generic packetization selection?

Need to have negotiation of SFrame using SDP?

Need to expose SFrame use on the wire?

# SFrame Authentication Signatures

- Goal is to validate that received content is actually coming from a given user
- Sounds like a good idea but
  - *<u>'It is up to the application to decide what to do when signature verification fails'</u>*
  - Potential breakage with intermediaries (SFU frame dropping), network (packet loss)...
  - This might be hard to implement in browsers
    - Plus additional buffering or delay requirements
- Question
  - What is the threat model? In particular, is the SFU part of the threat?
  - How is SFrame Authentication expected to be deployed?
  - Can SFrame authentication use cases be supported differently?
    - One key per incoming stream
    - 'Who is speaking' information sent as side information
    - SFU validating keyId collisions

# SFrame and Data Channel

- Data channel can be used for various data
  - Audio/Video, messages, application-specific structured data (subtitles, RTC game data…)
- Can SFrame be used with Data Channel
  - Spec is transport agnostic
  - Spec is currently focused on audio/video
    - There does not seem to be blockers for use outside of audio/video
- Is there a use case for SFrame with Data Channel?
  - Or other transports like WebSocket, WebTransport...