W3C Considerations for WebRTC and DASH for Interactive Streaming

April 5, 2022

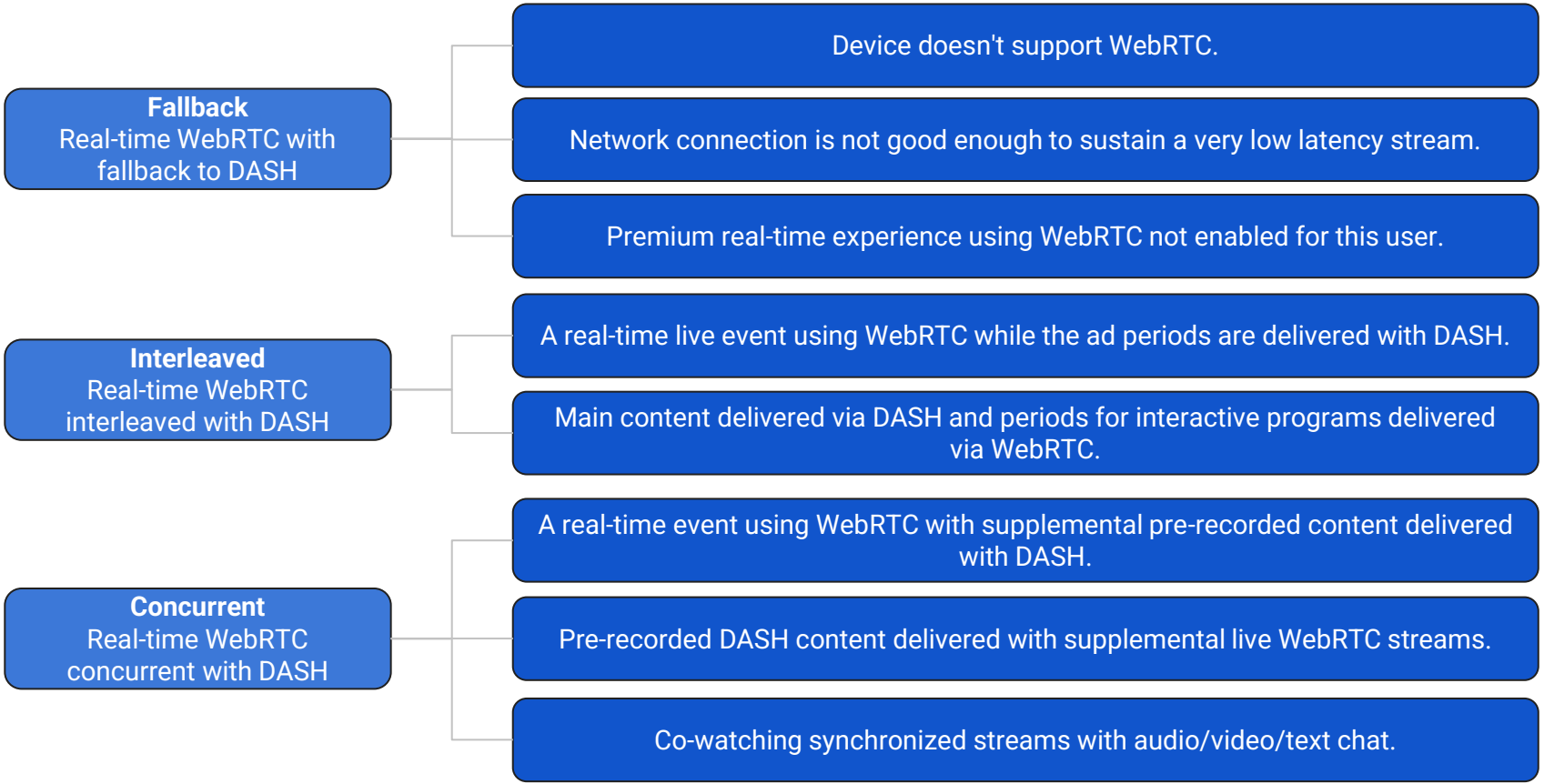
Agenda

Background on DASH-IF exploration

WebRTC requirements and proposed extensions

Moving from requirements to technical proposals

DASH + WebRTC Use Cases



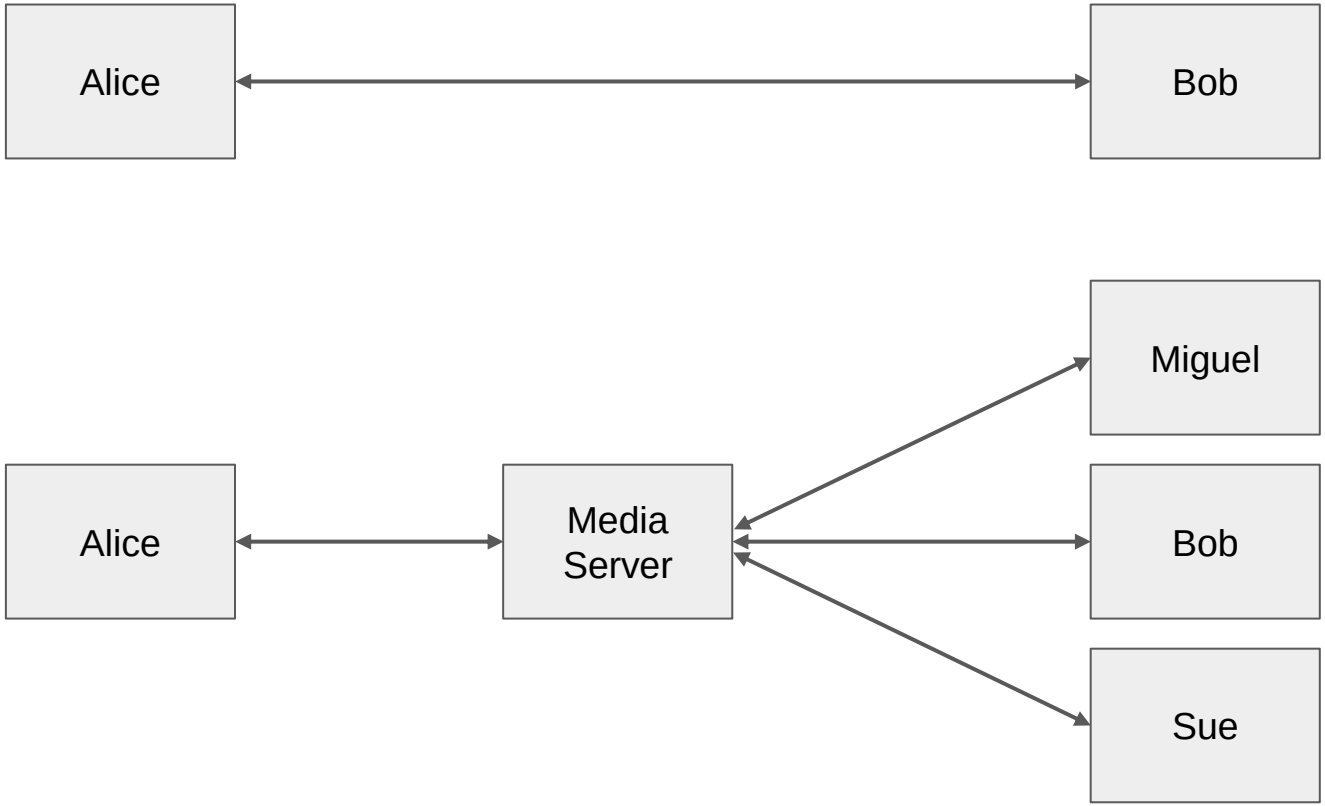
Related WebRTC Next Version (NV) Use Cases

3.2 Low latency P2P broadcast

3.6 Funny Hats

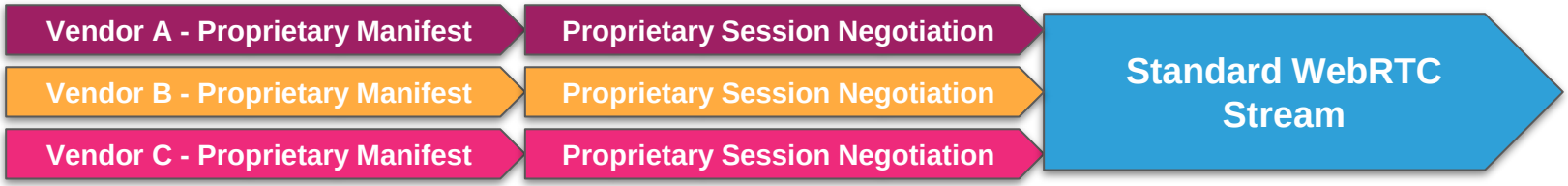
1. **Captioning**
2. **Transcription**
3. **Language translation**
4. Funny hats
5. Background removal or blurring
6. In-browser compositing
7. Voice effects
8. Stress detection

WebRTC "Broadcast"



Standardizing Workflow, From Discovery to Streaming

Current state of Real-Time Streaming



Goal for Real-Time Streaming

Work for WebRTC Extensions

- **Define and select appropriate session management/signaling protocol**
  - Define control protocol for dynamic stream switching that does not require SDP renegotiation
- **Continue development of methods for additional security of streams**
- Define a standardized means to deliver subtitles, closed captions, and other events
- Continue development of a mechanism for time synchronization of timed metadata and DASH periods
- Collection of metrics and client metadata for WebRTC sessions and translation to existing metrics and client metadata, transmission via APIs

Session Negotiation

Current issues:

- Specific transport method for signaling and session negotiation is left up to each application developer.
- Negotiation requires a number of round trips to establish a connection.

Requirements:

- Interoperability: Standard and specific transport methods for signaling and session negotiation
- Minimize the time to the first frame (TTFF)
- Not preclude the use of other protocols such as WebSockets

WHIP: WebRTC-HTTP Ingestion Protocol

WHIP proposes a simple protocol for supporting WebRTC as media ingestion method that:

- Is easy to implement,
- Is as easy to use as current RTMP URIs.
- Is fully compliant with WebRTC and RTCWEB specs.
- Allows for both ingest in traditional media platforms and ingest in WebRTC end-to-end platforms with the lowest possible latency.
- Lowers the requirements on both hardware encoders and broadcasting services to support WebRTC.
- Is usable both in web browsers and in native encoders.

Companion access protocol: WHAP (https://github.com/x186k/whip-whap-js)

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

WHSNP: WebRTC HTTP Session Negotiation Protocol

WHSNP is an HTTP-based protocol for establishing WebRTC communications that:

- Allows WebRTC endpoints to publish to media servers or subscribe to content from media servers, using uniform communication for both subscribing and publishing.
- Uses tokens containing capabilities to streamline negotiation.
- Can be used in both web browsers and native applications.
- Allows manual tuning.
- Is simple to implement.
- Is built on commonly-available standards such as HTTP and JSON.
- Is optimized to minimize TTFF.

Still in development, needs a better acronym.

Securing WebRTC

Current: Secure Real-time Transport Protocol (SRTP) - Transport-level protocol that provides encryption, message authentication and integrity, and replay attack protection to the RTP data in both unicast and multicast applications. This encryption is node-to-node.

Future: end-to-end encryption (E2EE) using Insertable Stream to E2EE, which allows customization or manipulation of the data prior to sending it over the wire or E2EE with secure frame (SFrame).

DRM

Goal is end-to-end encryption with secure key exchange

Current options:

- WebRTC DataChannel to MSE/EME
- WebSockets to MSE/EME (head-of-line blocking).

Future options

- Insertable Stream to MSE/EME (see how FaceTime using insertable streams for end-to-end encryption)
- WebTransport to MSE/EME (requires CMAF, and has some latency issues in different browsers)

Stream Security: Recommended Options

- WebCodecs + EME (without CMAF packaging). Bridge gaps between WebCodecs and EME
- DataChannel or WebTransport or Insertable Streams to WebCodecs and EME (without CMAF packaging)
- Client-side demuxing
- AES-128 CBCS encryption of only the elementary stream
- Following a single encode and encryption path and multi-package approach compatible with DASH

Developing Technical Proposals

Seek advice from experts

Determine best options for standardization

Draft and test

Further Reading

DASH-IF Report Summary: https://dashif.org/webRTC/

Full Report: https://dashif.org/webRTC/report

Interest survey: https://forms.gle/Yy89GGeMsXYQixBZ6