November 18-20, 2014 - San Jose Convention Center - San Jose, California

WebRTC V
CONFERENCE & EXPO

Explore Changes in Enterprise Communications

#webrtcexpo | @webrtcexpo

www.webrtcexpo.com

Standards Update and Directions

Standards Update & Directions – The W3C Part

Daniel C. Burnett, Ph.D.
Aspect WebRTC Expert
StandardsPlay.com

JavaScript APIs – W3C standards

- Status summary
- Recent highlights
  - Output Device Enumeration
  - Promises
  - Authenticated Origins
  - AddStream -> AddTrack
  - RTCRtpSender/Receiver
  - Screen sharing
  - Other tidbits

Status Summary

- Boring! (This is good)
  - A few big topics, then . . .
  - Many issue and pull requests
- Targeting Last Call WD for Media Capture this year
- Trying for Last Call WD for WebRTC 1Q15

Output Device Enumeration/Selection

- Most requested WebRTC feature for Chrome
- Issue: gUM lets you select input but not output
- Proposal:
  - Include output devices in enumeration of devices with new sinkId (just like sourceId for inputs)
  - Permission grant for device actually grants permission for all devices with same group id
  - See https://www.w3.org/wiki/images/d/d6/Output_Device_Selection%2C_TPAC_2014.pdf
- Decision: use as foundation for new output spec, needs coordination with many other groups in W3C

Promises

- W3C wants all async APIs to return Promises rather than using callbacks
- Issue: Promises becoming popular for APIs, e.g.

```
Navigator.mediaDevices.getUserMedia({audio:true, video:true})
.then(gotStream)
.catch(logError);
```

- Decision:
  - Navigator.getUserMedia() will accept callbacks only
  - Navigator.mediaDevices.getUserMedia() will return a Promise only
  - All async RTCPeerConnection methods will accept callbacks and return a Promise

Authenticated Origins

- W3C wants to require authenticated origins, e.g. HTTPS
- Issue: Unauthenticated origins are insecure
- Proposal:
  - Forbid use of HTTP or other unauthenticated origins
- Decision: Specifications will recommend, but not require, that WebRTC content origins be authenticated

AddStream -> AddTrack

- PCs now operate on tracks rather than streams
- Issue: Need better track-oriented connection info and/or controls
- Proposal:
  - RTCRtpSender addTrack(MST track, MediaStream... streams)
  - void removeTrack(RTCRtpSender sender)
  - onaddstream -> ontrack
- Decision: Agreed, done. Some details still TBD. Existing stream commands will move to polyfill library.

RTCRtpSender/Receiver

- New extension objects (originally) from ORTC
- Issue: Need better track-oriented connection info and/or controls
- Proposal:
  - Several layered proposals from Google including info on
    - ICE transports, remote Certs used, selected candidate pair, encoding parameters (get and set for, e.g. pause/resume, maxBitrate)
    - See https://www.w3.org/2011/04/webrtc/wiki/images/6/6c/WebRTC_RTCSender-Receiver%2C_TPAC_2014.pdf
- Decision: Objects added already, ICE info will be added, but other info and controls are under discussion

Screen Sharing

- Second highest request for Google Chrome
- Discussion:
  - Security is tricky, since web sandboxing model assumes one site can't see another's code
  - Proposal is to identify gUM source as display, window, application, e.g.,
    ```
    Navigator.MediaDevices.getUserMedia({audio:true, video:true, source: "display"})
    ```
- Decision: Needs some work, but everyone wants this ☺

Other Tidbits

- Constraints syntax now finalized – see the Media Capture and Streams specification
- Control over DTLS certificate renewal being considered – maybe using WebCrypto?
- Stats API moving into separate document, many more statistics being defined.

Standards Update & Directions – The IETF Part

Alex Eleftheriadis, Ph.D.
Chief Scientist & Co-founder
Vidyo

WebRTC Spec Space

- W3C
  - JavaScript API
    - W3C.WD-webrtc-20120209, "WebRTC 1.0: Real Time Comm. Between Browsers"
    - W3C.WD-mediacapture-streams-20120628, "Media Capture and Streams"
- IETF
  - Architecture and on-the-wire protocols
  - References from W3C: 8 I-Ds, 6 RFCs, +2 I-Ds informative
  - References from within IETF specs:
    - Normative references: 25 I-Ds, 2 RFCs, +2 individual I-D (not WD)
    - Informative references: 15 I-Ds

Web of Dependencies

- draft-jennings-rtcweb-deps

  Cullen Jennings (Cisco)

  Nov. 10, 2014

```
Network Working Group                                        C. Jennings
Internet-Draft                                                     Cisco
Intended status: Informational                         November 10, 2014
Expires: May 14, 2015


                         WebRTC Dependencies
                     draft-jennings-rtcweb-deps-05

Abstract

   This draft will never be published as an RFC and is meant purely to
   help track the IETF dependencies from the W3C WebRTC documents.

Status of This Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF).  Note that other groups may also distribute
   working documents as Internet-Drafts.  The list of current Internet-
   Drafts is at http://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time.  It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on May 14, 2015.

Copyright Notice

   Copyright (c) 2014 IETF Trust and the persons identified as the
   document authors.  All rights reserved.

   This document is subject to BCP 78 and the IETF Trust's Legal
   Provisions Relating to IETF Documents
   (http://trustee.ietf.org/license-info) in effect on the date of
   publication of this document.  Please review these documents
   carefully, as they describe your rights and restrictions with respect
   to this document.  Code Components extracted from this document must
   include Simplified BSD License text as described in Section 4.e of
   the Trust Legal Provisions and are provided without warranty as
   described in the Simplified BSD License.




Jennings                  Expires May 14, 2015                  [Page 1]
```

ietf-rtcweb-overview (v12, 10/14, 22p)

- Informative Reference from W3C
- Intended as “roadmap” to specs
- High-level overview
  - Architecture and Functionality Groups
  - Data Transport (TURN etc.)
  - Data Framing and Security (RTP, SRTP, etc.)
  - Data Formats (MTI audio and video)
  - Connection Management (JSEP)
  - Presentation and Control (W3C)
  - Local System Functions (echo etc.)

```
                  +------------------------+  On-the-wire
                  |                        |  Protocols
                  |      Servers           |--------->
                  |                        |
                  |                        |
                  +------------------------+
                              ^
                              |
                              |
                              | HTTP/
                              | Websockets
                              |
                              |
                +----------------------------+
                |    Javascript/HTML/CSS     |
                +----------------------------+
             Other  ^                 ^RTC
             APIs   |                 |APIs
                +---|-----------------|------+
                |   |                 |      |
                |                 +---------+|
                |                 | Browser ||  On-the-wire
                | Browser         | RTC     ||  Protocols
                |                 | Function|----------->
                |                 |         ||
                |                 |         ||
                |                 +---------+|
                +---------------------|------+
                                      |
                                      V
                                 Native OS Services
```

ietf-rtcweb-overview

Browser RTC trapezoid

```
       +-----------+             +-----------+
       |   Web     |             |   Web     |
       |           |  Signaling  |           |
       |           |-------------|           |
       |  Server   |   path      |  Server   |
       |           |             |           |
       +-----------+             +-----------+
            /                           \
           /                             \ Application-defined
          /                               \  over
         /                                 \ HTTP/Websockets
        /  Application-defined over         \
       /   HTTP/Websockets                   \
      /                                       \
+-----------+                           +-----------+
|JS/HTML/CSS|                           |JS/HTML/CSS|
+-----------+                           +-----------+
+-----------+                           +-----------+
|           |                           |           |
|           |                           |           |
|  Browser  | ------------------------- |  Browser  |
|           |          Media path       |           |
|           |                           |           |
+-----------+                           +-----------+
```

WebRTC Entities (rtcweb-overview)

1. User Agent: (also called a WebRTC UA or a WebRTC browser) something that conforms to both the protocol specification and the Javascript API.
2. Device: Something that conforms to the protocol specification, but does not claim to implement the Javascript API.
3. Endpoint: either a WebRTC User Agent or a WebRTC device.
4. WebRTC-compatible endpoint: an endpoint that is capable of successfully communicating with a WebRTC Endpoint, but may fail to meet some requirements of a WebRTC endpoint. This may limit where in the network such an endpoint can be attached, or may limit the security guarantees that it offers to others.
5. Gateway: a WebRTC-compatible endpoint that mediates traffic to non-WebRTC entities.

All WebRTC browsers (UAs) are WebRTC devices, so any requirement on a WebRTC device also applies to a WebRTC browser.

ietf-rtcweb-transports (v7, 10/14, 15p)

- Transport protocols used by WebRTC, including the protocols used for interaction with intermediate boxes such as firewalls, relays and NAT boxes.
- Middlebox Functions
  - MUST: IPv4 & IPv6, ICE (full, not ICE-Lite), TURN (both endpoints behind NATs, with endpoint-dependent mapping), browser config of STUN and TURN servers, TCP for TURN and TLS over TCP (5766 Sec. 2.1), ICE-TCP candidates (RFC 6544), with RTP framing per RFC4751 for content that does not have own framing. HTTP CONNECT and proxy authentication (in WebRTC browsers)
  - SHOULD: ICE happy eyeballs, discard IPv6 permanent addresses in favor of temporary, HTTP CONNECT and proxy authentication (inWebRTC devices)

ietf-rtcweb-transports

- Transport Protocols
  - Secure RTP.
  - Key exchange using DTLS-SRTP.
  - Data transport using SCTP over DTLS over ICE.
  - Multiplexing of DTLS and RTP over the same port pair, as described in the DTLS_SRTP specification [RFC5764], section 5.1.2. All application layer protocol payloads over this DTLS connection are SCTP packets.
- Media Prioritization
  - "normal", "below normal", "high" or "very high" (app tells the browser)
  - May use DSCP markings
  - Local prioritization: should use twice the transmission capacity of the previous level

ietf-rtcweb-rtp-usage (v6, 2/13, 62p)

- How the Real-time Transport Protocol (RTP) is used in the WebRTC context, and gives requirements for which RTP features, profiles, and extensions need to be supported
- RTP and RTCP, with support for multiple SSRCs per RTP session, simultaneously
- SRTP used throughout - "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)- Based Feedback (RTP/SAVPF)" [RFC5124] as extended by [I-D.ietf-avtcore-avp-codecs]

ietf-rtcweb-rtp-usage

- Treatment of multiplexing
  - Session multiplexing required, but for legacy
  - SSRC multiplexing required for new systems
    - Signaling using BUNDLE
  - Alternative solution using a shim layer is being recommended and considered (no consensus)
  - RTP and RTCP multiplexed on same port, reduced-size RTCP
  - Symmetric RTP

ietf-rtcweb-rtp-usage

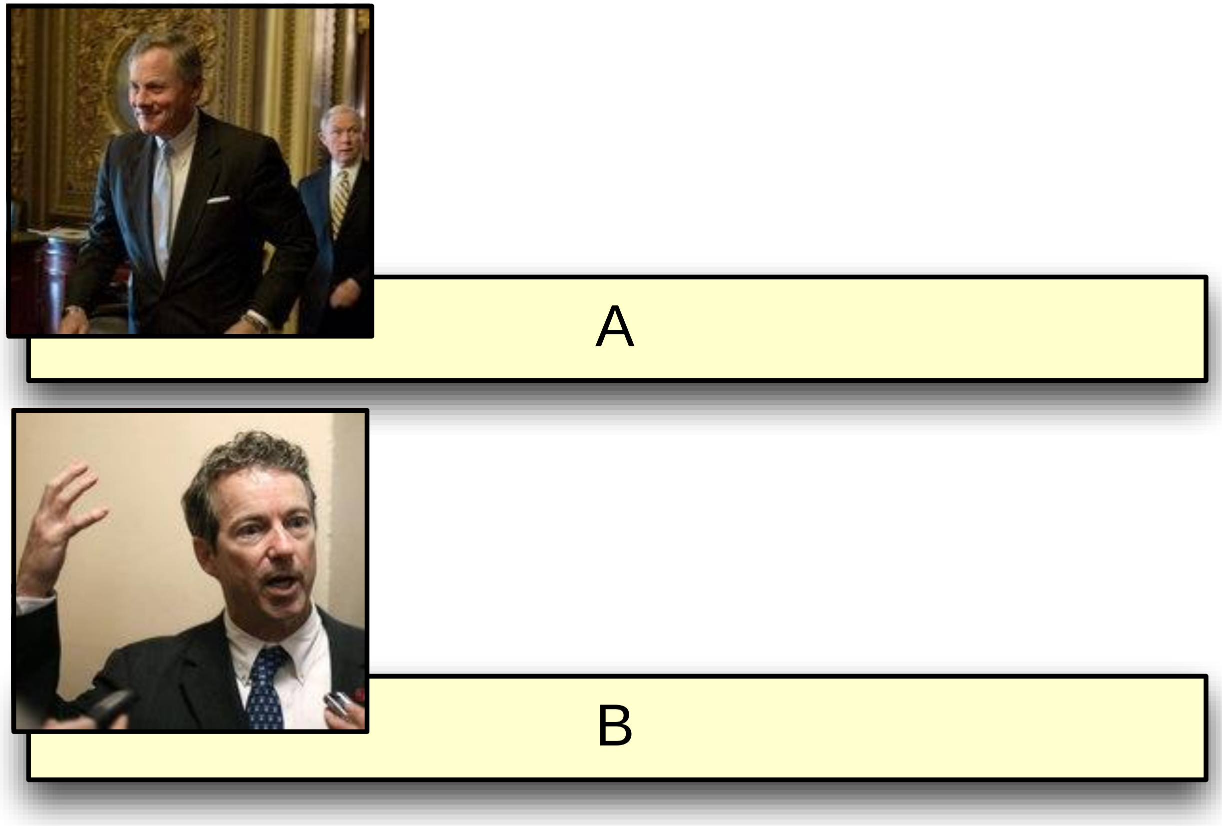
- RTCP conferencing extensions
  - Full Intra Request (FIR): senders required to respond, optional for receivers
  - Picture Loss Indication (PLI): senders required to support, optional for receivers
  - Slice Loss Indication (SLI): optional
  - Reference Picture Selection Indication (RPSI): optional
  - Temporal-Spatial Trade-off Request (TSTR): optional
  - Temporary Maximum Media Stream Bit Rate Request (TMBRR): senders required to respond, optional for receivers
- Header Extensions
  - Rapid sync: recommended
  - Client-to-mixer and mixer-to-client audio level: recommended (with mandatory encryption)

ietf-rtcweb-rtp-usage

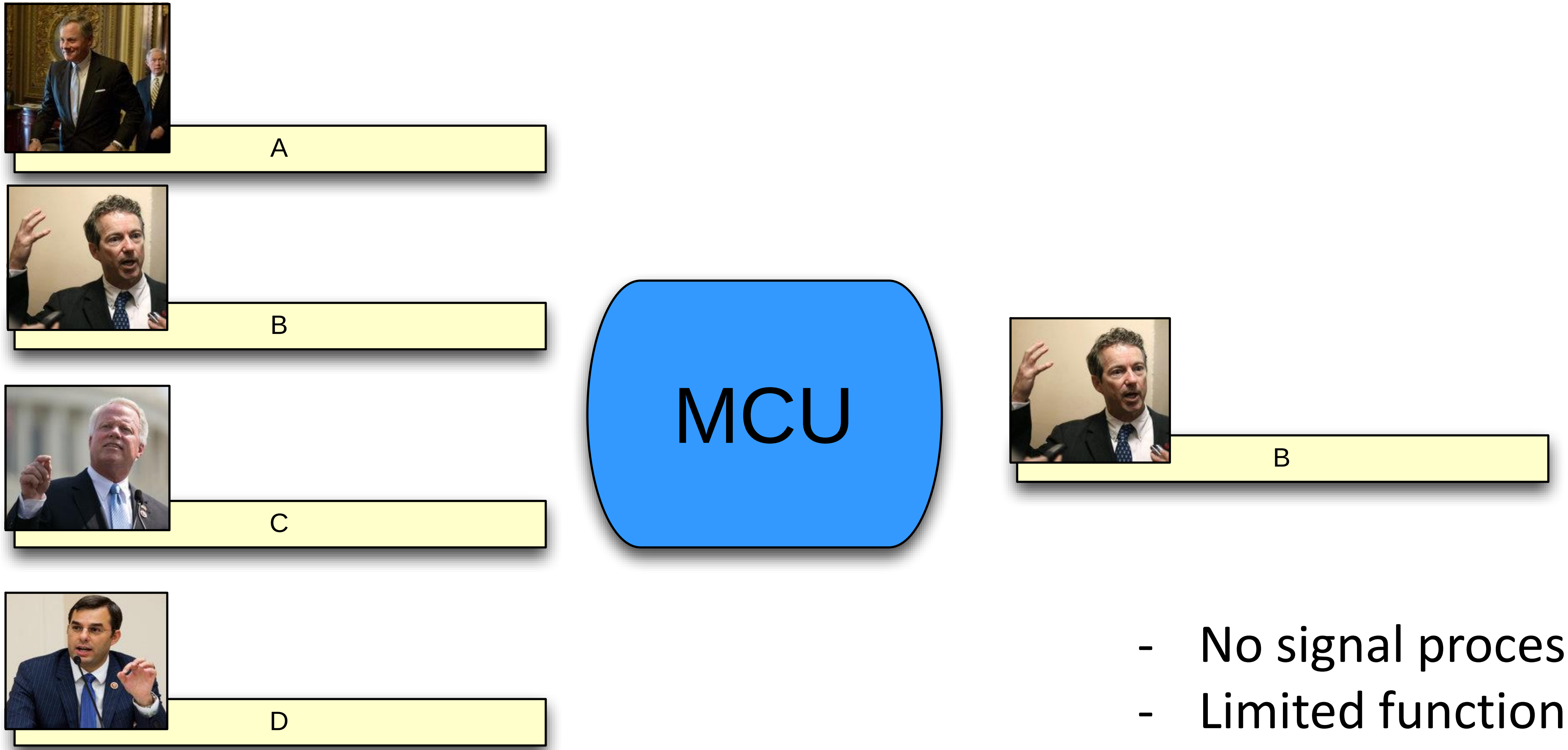
- Error Resilience (in addition to FIR etc.)
  - Generic NACK support (through RTP/SAVPF profile)
  - Senders must understand, but may chose to ignore
  - Receivers must understand retransmission packets
  - No FEC recommendation
- Congestion Control
  - No explicit control algorithm, but circuit breakers mandatory (conditions about when to stop transmitting)
- Simulcast
  - TBD

Some Notes re. Multiplexing

Depicting Video Streams

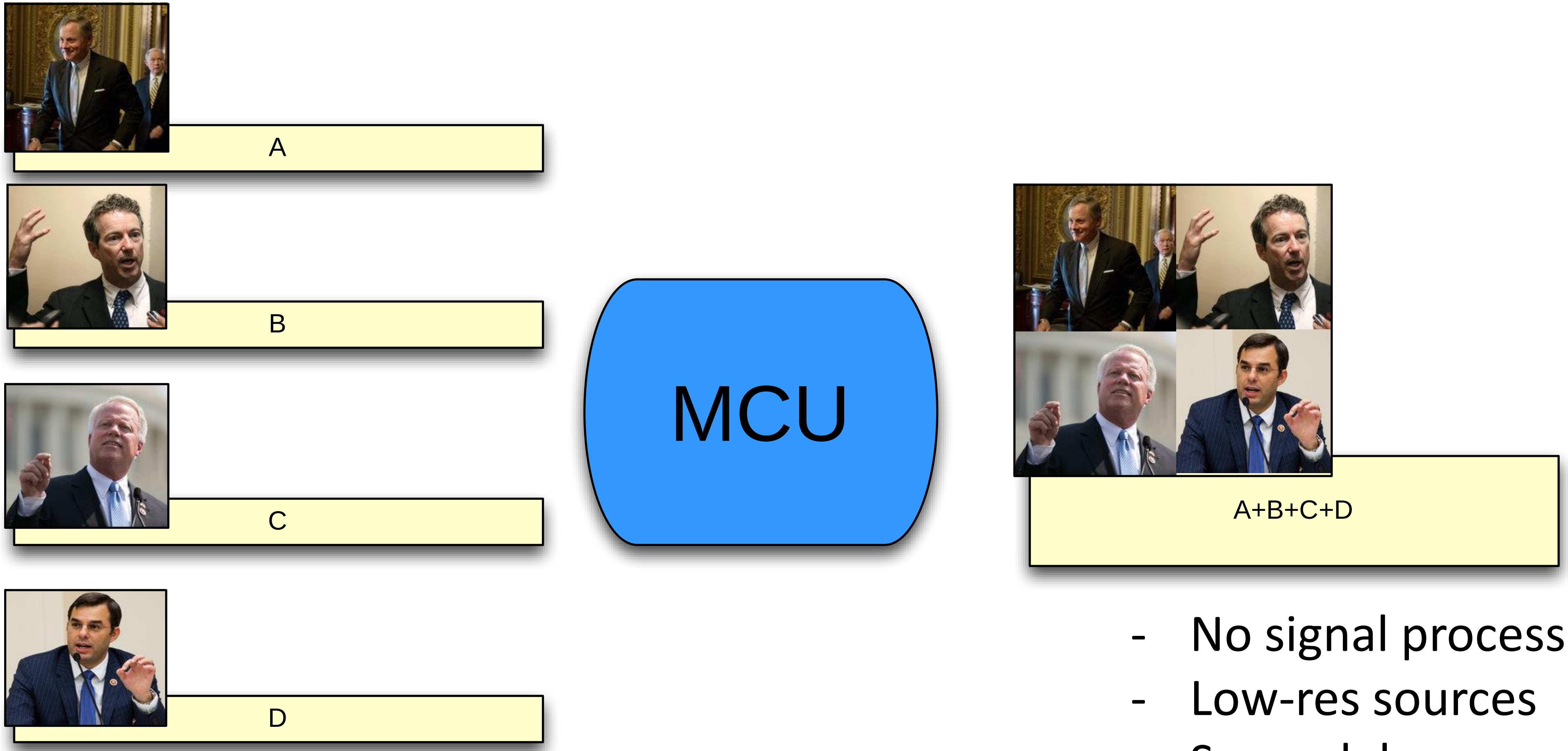


Switching MCU



- No signal processing
- Limited functionality (single view)

Mixing MCU



- No signal processing
- Low-res sources
- Some delay
- Limited layouts

Transcoding MCU


A


B


C


D

MCU


E

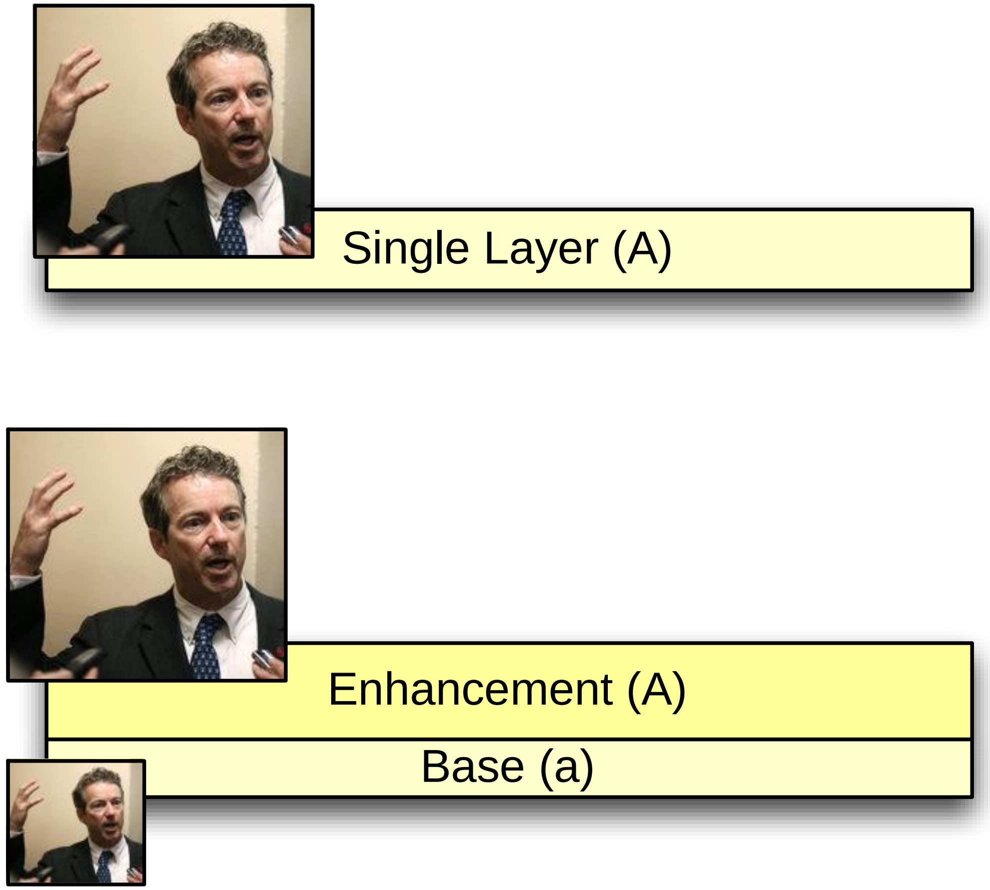
- Very Flexible
- High Delay
- Transcoding Loss
- Complexity & Cost

Standard Topologies

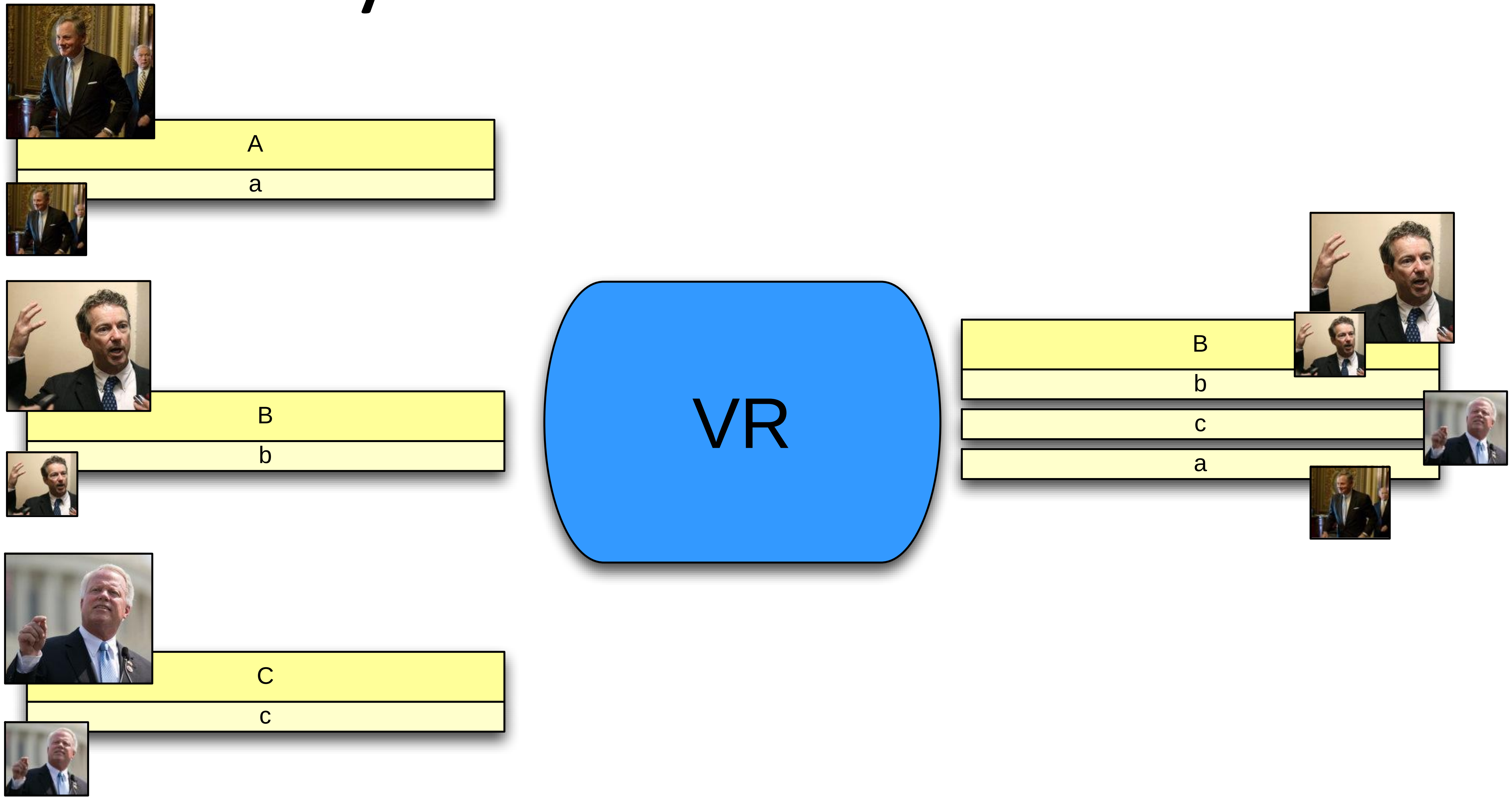
- RFC 5117 (Jan '08)
  - Topo-RTCP-terminating-MCU

```
3.6.  Point to Multipoint Using RTCP-Terminating MCU

   Shortcut name: Topo-RTCP-terminating-MCU

            +---+      +------------+      +---+
            | A |<---->| Multipoint |<---->| B |
            +---+      |  Control   |      +---+
                       |   Unit     |
            +---+      |   (MCU)    |      +---+
            | C |<---->|            |<---->| D |
            +---+      +------------+      +---+

      Figure 8 - Point to Multipoint Using Content Modifying MCUs
```

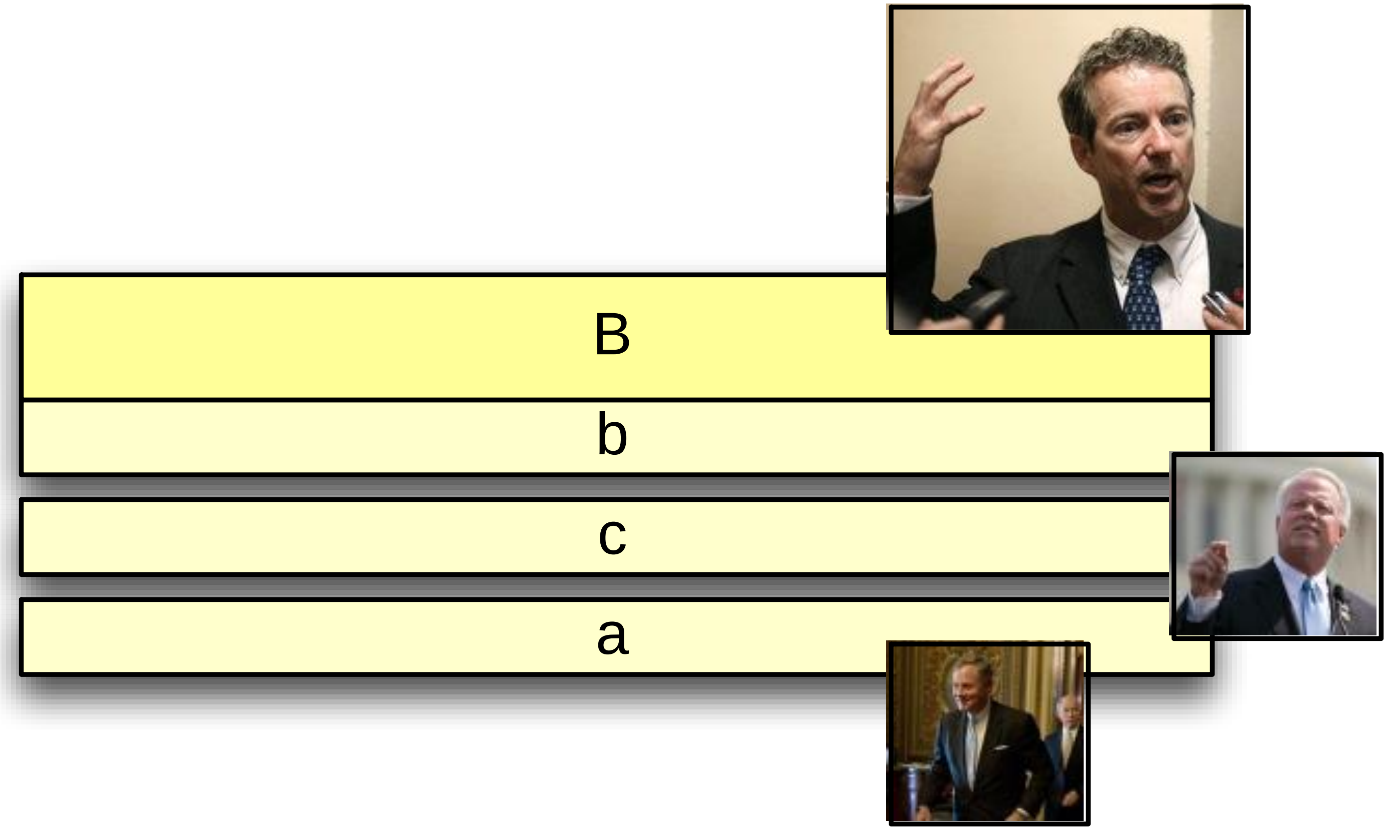
Scalable Video Coding (SVC)



“VidyoRouter”



Endpoint Design

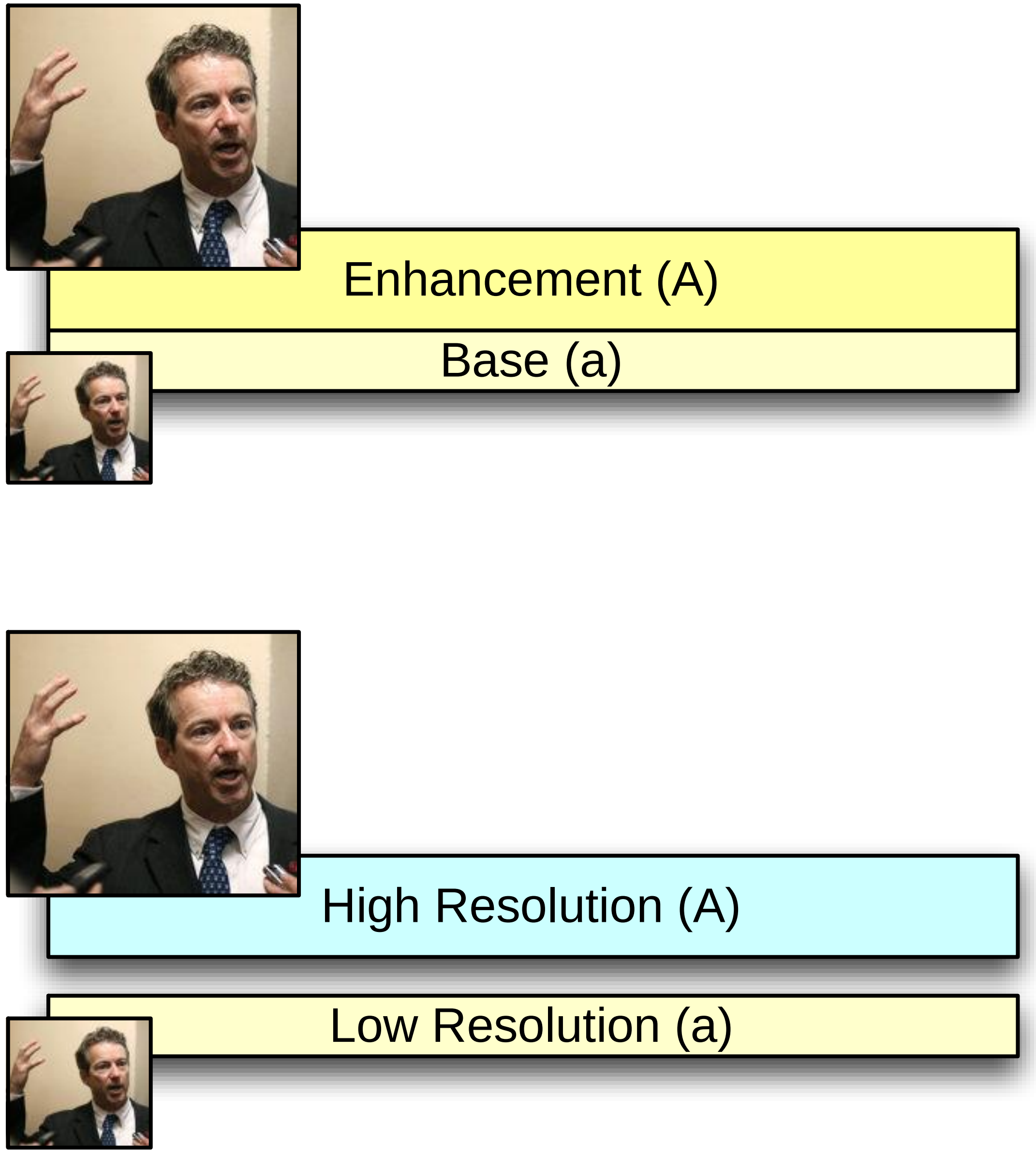


multi-stream
+
composition
in endpoint

(~ = browser)

Perfectly matches WebRTC client architecture

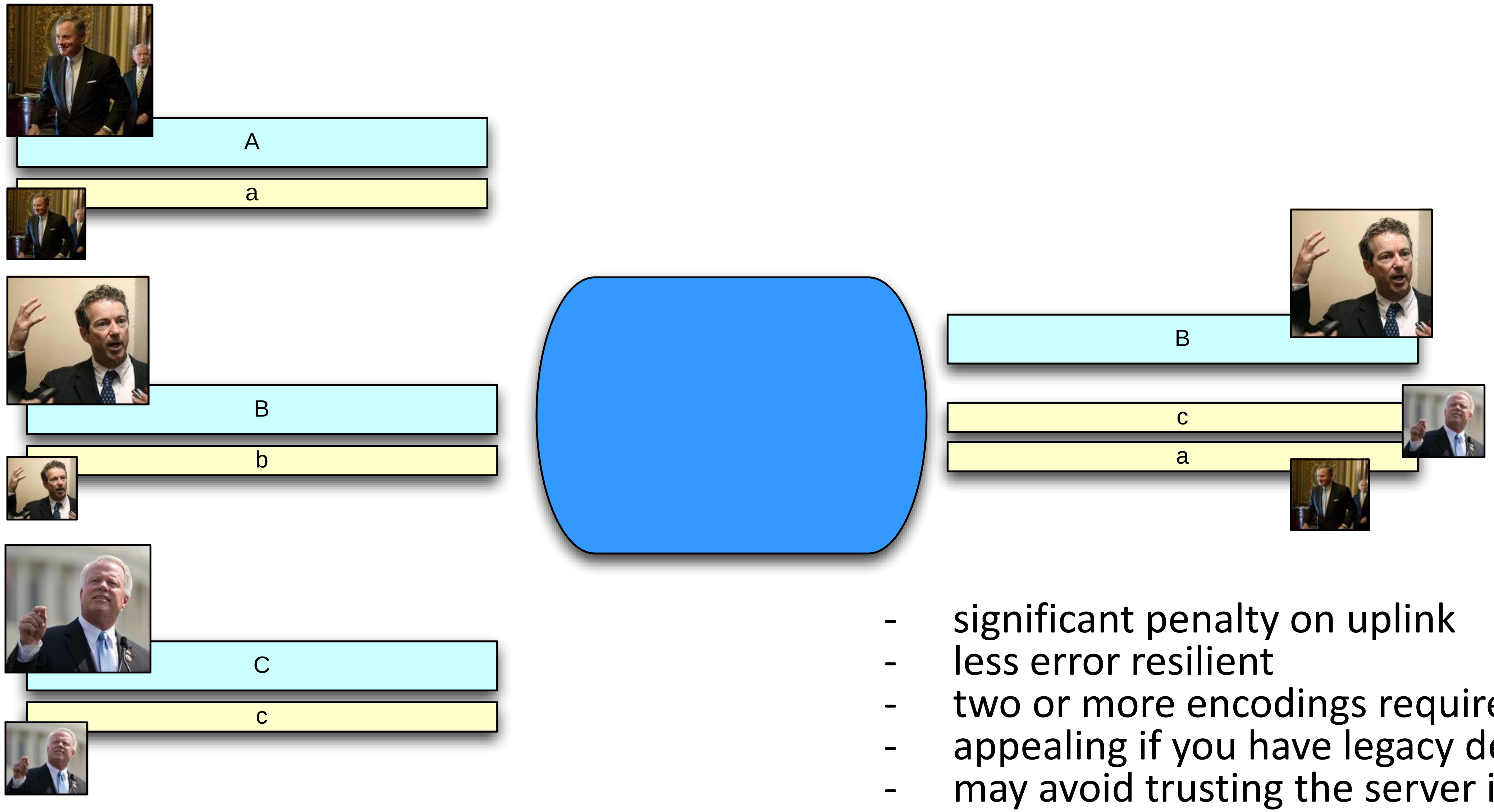
Simulcasting



For 2:1 resolution ratios:

- ~50% overhead vs. single layer
- ~20% overhead vs. scalable

Simulcast Edition



- significant penalty on uplink
- less error resilient
- two or more encodings required
- appealing if you have legacy decoders
- may avoid trusting the server in SRTP

Who is using all this?

In production or development:

- Cisco
- Google
- Polycom
- Microsoft
- Vidyo
- …

Many others are using SVC alone

New (standard) topologies

- ‘relay’ is used (as forwarding everything)
- draft-westerlund-avtcore-rtp-topologies-update-02 (Westerlund & Wenger)
  - “media projecting middlebox”
- My proposal in Oct. 2013 :

  Selective Forwarding Unit - SFU
- Now in:

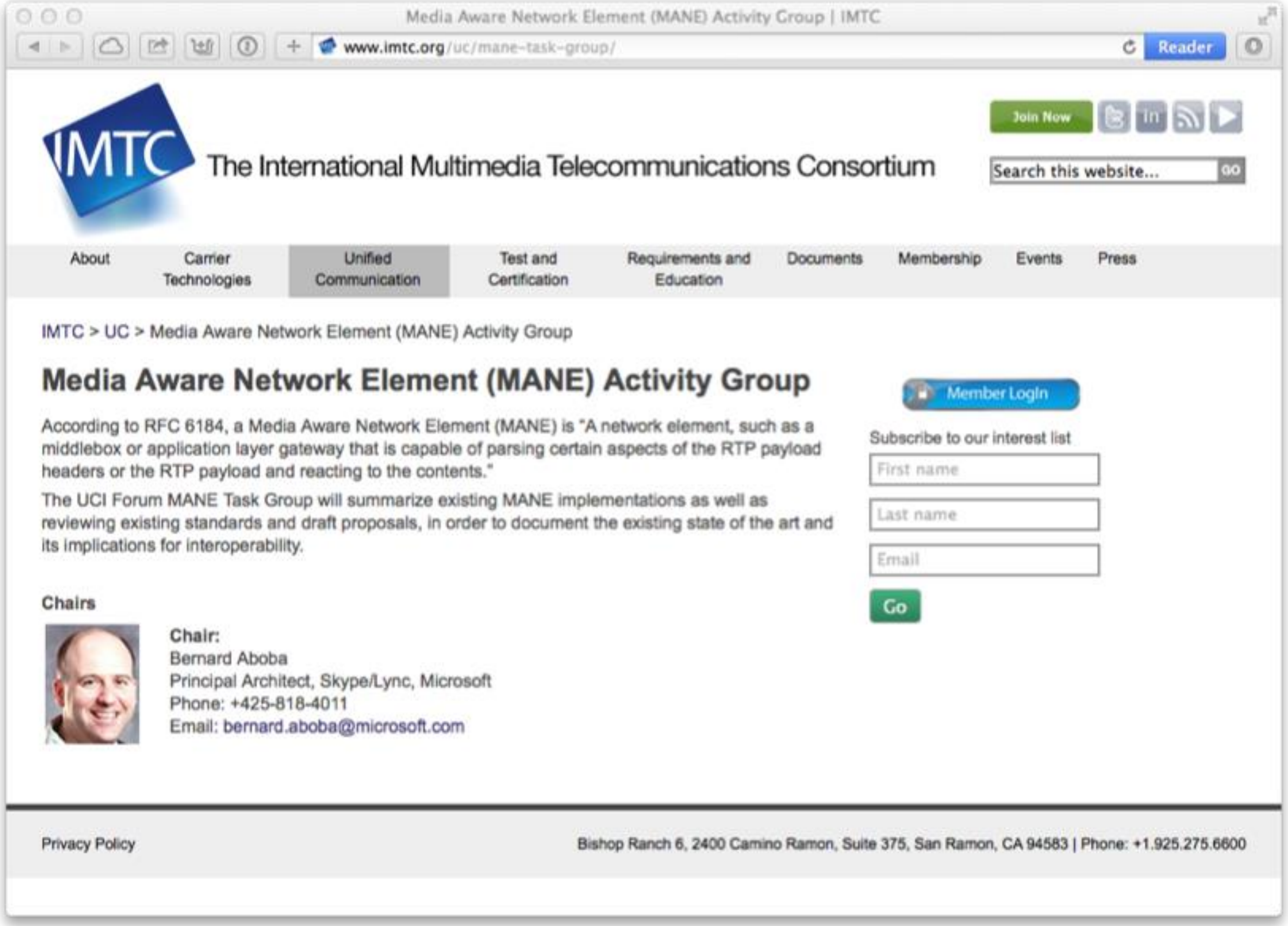
  draft-ietf-avtcore-rtp-topologies-update

  Section 3.7, “Selective Forwarding Middlebox”

IMTC MANE Activity Group – www.imtc.org

Summarize existing MANE implementations as well as reviewing existing standards and draft proposals, in order to document the existing state of the art and its implications for interoperability.

Chaired by Bernard Aboba (Microsoft)



draft-ietf-rtcweb-video (v2, 10/14, 9p)

- Requirements and considerations for WebRTC applications to send and receive video across a network. It specifies the video processing that is required, as well as video codecs and their parameters.
- No consensus on MTI: VP8 vs. H.264

- News Flash (11/13/2014, IETF Honolulu): MTI = H.264 + VP8
  - Browsers must implement both
  - Non-browsers must implement both, unless one is declared royalty-free, in which case it is the only one they have to use

draft-ietf-rtcweb-audio(v7, 10/14, 6p)

- Outlines the audio codec and processing requirements for WebRTC client application and endpoint devices.
- Required:
  - Opus + Opus payload format (still in I-D)
  - G.711 PCMA and PCMU
  - RFC 3389 Comfort Noise (CN)
  - DTMF per RFC4733
- Recommendations for signal level normalization, including filter selection
- Acoustic Echo Cancellation recommended

draft-ietf-rtcweb-data-channels (v12, 9/14, 16p)

- Specifies the non-media data transport aspects.
- Provides an architectural overview of how the Stream Control Transmission Protocol (SCTP) is used in the WebRTC context as a generic transport service allowing WEB-browsers to exchange generic data from peer to peer.
- SCTP over DTLS over UDP
  - confidentiality, source authenticated, and integrity protected transfers
- ICE
  - Middlebox traversal in IPv4 and IPv6 networks.

draft-ietf-rtcweb-data-channels

- SCTP features:
  - Multiple unidirectional streams
  - TCP-friendly congestion control, modifiable for integration with the SRTP media stream congestion control
  - Support for both ordered and out-of-order message delivery
  - Arbitrary large messages through segmentation and reassembly
  - PMTU discovery
  - Support for reliable or partially reliable message transport

WebRTC Protocol Layers

DCEP: Data Channel Establishment Protocol

Multiplexing using SCTP's PPID

```
                    +------+------+------+
                    | DCEP | UTF-8|Binary|
                    |      | data | data |
                    +------+------+------+
                    |        SCTP        |
+----------------------------------------+
| STUN | SRTP |            DTLS          |
+----------------------------------------+
|                  ICE                   |
+----------------------------------------+
| UDP1 | UDP2 | UDP3 | ...               |
+----------------------------------------+
```

draft-ietf-rtcweb-data-channels

- SCTP over DTLS encapsulation (per I-D.ietf-tsvwg-sctp-dtls-encaps)
- SCTP Protocol Extensions:
  - Stream reconfiguration extension (RFC 6525) (stream "reset", used for closing channels)
  - Dynamic address reconfiguration extension (from RFC 5061), but only to support stream reset.
  - Partial reliability extension (RFC 3758). In addition to the timed reliability PR-SCTP policy defined in [RFC3758], the limited retransmission policy defined in I-D.ietf-tsvwg-sctp-prpolicies MUST be supported. Limiting the number of retransmissions to zero combined with unordered delivery provides a UDP-like service where each user message is sent exactly once and delivered in the order received.
- Transfer of user data:
  - PPIDs for: String/String Empty (JS string in UTF-8), Binary/Binary Empty (JS ArrayBuffer, ArrayBufferView, or Blob)

draft-ietf-rtcweb-data-protocol (v8, 9/14, 12p)

- Data Channel Establishment Protocol
- Simple protocol for establishing symmetric Data Channels between the peers.
- Uses a two way handshake and allows sending of user data without waiting for the handshake to complete.
- Channel Properties:
  - Reliable or unreliable message transmission
  - In-order or out-of-order delivery
  - Priority
  - Optional label
  - Optional protocol
  - Streams

draft-ietf-rtcweb-data-protocol

- DATA_CHANNEL_OPEN sent, DATA_CHANNEL_ACK response
- Messages sent multiplexed with user data, using SCTP payload protocol identifier (PPID) to demux
- Stream ID value selection based on DTLS role (client uses even numbers), to avoid collisions between sender and receiver
- Message types and channel types (to be registered with IANA)

Name	Type	Reference
Reserved	0x00	[RFCXXXX]
Reserved	0x01	[RFCXXXX]
DATA_CHANNEL_ACK	0x02	[RFCXXXX]
DATA_CHANNEL_OPEN	0x03	[RFCXXXX]
Unassigned	0x04-0xfe	
Reserved	0xff	[RFCXXXX]

Name	Type	Reference
DATA_CHANNEL_RELIABLE	0x00	[RFCXXXX]
DATA_CHANNEL_RELIABLE_UNORDERED	0x80	[RFCXXXX]
DATA_CHANNEL_PARTIAL_RELIABLE_REXMIT	0x01	[RFCXXXX]
DATA_CHANNEL_PARTIAL_RELIABLE_REXMIT_UNORDERED	0x81	[RFCXXXX]
DATA_CHANNEL_PARTIAL_RELIABLE_TIMED	0x02	[RFCXXXX]
DATA_CHANNEL_PARTIAL_RELIABLE_TIMED_UNORDERED	0x82	[RFCXXXX]
Reserved	0x7f	[RFCXXXX]
Reserved	0xff	[RFCXXXX]
Unassigned	rest	

draft-ietf-rtcweb-jsep (v8, 10/14, 65p)

- Javascript Session Establishment Protocol
- Elements:
  - Passing local and remote session descriptions
  - Interacting with the ICE state machine
- JSEP Signaling Model:

```
+-----------+                                +-----------+
|  Web App  |<--- App-Specific Signaling -->|  Web App  |
+-----------+                                +-----------+
      ^                                            ^
      |  SDP                                       |  SDP
      V                                            V
+-----------+                                +-----------+
|  Browser  |<----------- Media ------------>|  Browser  |
+-----------+                                +-----------+
```

draft-ietf-rtcweb-jsep

- Use of SDP for internal representation of session descriptions
- SDP syntax encapsulated into a SessionDescription object.
- JavaScript applications can treat it as a “blob”
- SDP interactions:
  - createOffer
  - createAnswer
  - setLocalDescription
  - setRemoteDescription
- SDP types: “offer”, “answer”, “pranswer”, “rollback” (undo setLocal if offer is not accepted)

JSEP State Machine

```
                    setRemote(OFFER)               setLocal(PRANSWER)
                        /-----\                               /-----\
                        |     |                               |     |
                        v     |                               v     |
         +---------------+    |                +---------------+    |
         |               |----/                |               |----/
         |               | setLocal(PRANSWER)  |               |
         |  Remote-Offer |------------------- >| Local-Pranswer|
         |               |                     |               |
         |               |                     |               |
         +---------------+                     +---------------+
              ^   |                                   |
              |   | setLocal(ANSWER)                  |
setRemote(OFFER)  |                                   |
              |   V                  setLocal(ANSWER) |
         +---------------+                            |
         |               |                            |
         |               |<---------------------------+
         |    Stable     |
         |               |<---------------------------+
         |               |                            |
         +---------------+          setRemote(ANSWER) |
              ^   |                                   |
              |   | setLocal(OFFER)                   |
setRemote(ANSWER) |                                   |
              |   V                                   |
         +---------------+                     +---------------+
         |               |                     |               |
         |               | setRemote(PRANSWER) |               |
         |  Local-Offer  |------------------- >|Remote-Pranswer|
         |               |                     |               |
         |               |----\                |               |----\
         +---------------+    |                +---------------+    |
                        ^     |                               ^     |
                        |     |                               |     |
                        \-----/                               \-----/
                    setLocal(OFFER)               setRemote(PRANSWER)
```

draft-ietf-rtcweb-jsep

- ICE - gathering phase
  - Triggered by addition of new m= lines in local desription, or new ICE credentials in the session description indicating an ICE restart.
  - Use of new ICE credentials can be triggered explicitly by the application, or implicitly by the browser in response to changes in the ICE configuration.
  - When a new gathering phase starts, the ICE Agent will notify the application through a callback. When each new ICE candidate becomes available, the ICE Agent will supply it to the application via an additional callback; these candidates will also automatically be added to the local session.
  - When all candidates have been gathered, a callback will be dispatched to signal that the gathering process is complete.
- Optionally ICE trickling for faster media setup

draft-ietf-rtcweb-security-arch (v9, 2/14, 43p)

- WebRTC Security Architecture

A call with Identity Provider (IdP):

```
                +----------------+
                |                |
                |   Signaling    |
                |   Server       |
                |                |
                +----------------+
                    ^        ^
                   /          \
           HTTPS  /            \   HTTPS
                 /              \
                /                \
               v                  v
            JS API              JS API
       +-----------+          +-----------+
       |           |  Media   |           |
 Alice |  Browser  |<-------->|  Browser  | Bob
       |           |(DTLS+SRTP)|          |
       +-----------+          +-----------+
             ^      ^--+    +--^     ^
             |         |    |        |
             v         |    |        v
       +-----------+   |    |   +-----------+
       |           |<-------+   |           |
       |   IdP1    |   |        |   IdP2    |
       |           |   +------->|           |
       +-----------+            +-----------+
```

Sequencing (Alice)

- Alice clicks to Call Bob
- JS button callback creates PeerConnection
- JS creates MediaStream with MediaStreamTracks connected to audio and video inputs
- ➢ Browser prompts Alice for permission
- Signaling messages (via JSEP) containing:
  - Media information
  - ICE candidates
  - Fingerprint attribute binding the communication to a key pair
- ➢ Prior to sending out the signaling message, the PeerConnection code contacts the identity service and obtains an assertion binding Alice's identity to her fingerprint. The exact details depend on the identity service (but PeerConnection can be agnostic).
- The message is sent to the signaling server (over TLS), and then to Bob's browser.

Sequencing (Bob)

- The JS on Bob's browser processes it, and alerts Bob to the incoming call and to Alice's identity.
- Alice has provided an identity assertion and so Bob's browser contacts Alice's identity provider (in a generic way so the browser has no specific knowledge of the IdP) to verify the assertion. This allows the browser to display a trusted element in the browser chrome indicating that a call is coming in from Alice.
- If Bob agrees (by, e.g., clicking a button) a PeerConnection is instantiated with the message from Alice's side.
- Then, a similar process occurs as on Alice's browser: Bob's browser prompts him for device permission, the media streams are created, and a return signaling message containing media information, ICE candidates, and a fingerprint is sent back to Alice via the signaling service. If Bob has a relationship with an IdP, the message will also come with an identity assertion.

Peer Authentication

- Details in W3C API spec
- PeerConnection downloads JS from a specific location on the IdP domain ("IdP proxy")
- IdP proxy and browser communicate using a secure MessageChannel.
- Browser is agnostic of IdP specifics.

```
+------------------------------------+
|  https://calling-site.example.com  |
|                                    |
|                                    |
|                                    |
|          Calling JS Code           |
|                 ^                  |
|                 | API Calls        |
|                 v                  |
|          PeerConnection            |
|                 ^                  |
|                 | postMessage()    |
|                 v                  |
|    +-------------------------+     |     +---------------+
|    | https://idp.example.org |     |     |               |
|    |                         |<--------->|   Identity    |
|    |        IdP JS           |     |     |   Provider    |
|    |                         |     |     |               |
|    +-------------------------+     |     +---------------+
|                                    |
+------------------------------------+
```

draft-ietf-rtcweb-security-arch

- Consent Freshness – avoid continuously sending data when the other party is gone. Uses timeouts on timed STUN Binding requests/responses.

Times Estimates (from Jennings)

ETA	Draft Name
2014 Nov	[I-D.ietf-tram-alpn]
2014 Dec	[I-D.ietf-payload-vp8]
2014 Dec	[I-D.ietf-rtcweb-data-channel]
2014 Dec	[I-D.ietf-rtcweb-data-protocol]
2014 Dec	[I-D.ietf-rtcweb-security-arch]
2014 Dec	[I-D.ietf-rtcweb-security]
2015 Jan	[I-D.ietf-payload-rtp-h265]
2015 Jan	[I-D.ietf-rtcweb-constraints-registry]
2015 Jan	[I-D.ietf-rtcweb-rtp-usage]
2015 Jan	[I-D.ietf-rtcweb-transports]
2015 Feb	[I-D.ietf-httpbis-header-compression]
2015 Feb	[I-D.ietf-httpbis-http2]

Times Estimates (from Jennings)

2015 Feb	[I-D.ietf-mmusic-sdp-bundle-negotiation]
2015 Feb	[I-D.ietf-mmusic-sdp-mux-attributes]
2015 Feb	[I-D.ietf-rtcweb-alpn]
2015 Feb	[I-D.ietf-rtcweb-stun-consent-freshness]
2015 Feb	[I-D.ietf-tsvwg-sctp-dtls-encaps]
2015 Feb	[I-D.ietf-tsvwg-sctp-ndata]
2015 Feb	[I-D.ietf-tsvwg-sctp-prpolicies]
2015 Mar	[I-D.ietf-mmusic-msid]
2015 Mar	[I-D.ietf-mmusic-sctp-sdp]
2015 Mar	[I-D.ietf-payload-rtp-opus]
2015 April	[I-D.ietf-httpbis-tunnel-protocol]
2015 May	[I-D.ietf-rtcweb-audio]
2015 May	[I-D.ietf-rtcweb-jsep]
2015 May	[I-D.ietf-rtcweb-overview]
2015 May	[I-D.ietf-rtcweb-video]

Times Estimates (from Jennings)

```
|             | [I-D.ietf-avtcore-multi-media-rtp-session]        |
|             |                                                   |
|             | [I-D.ietf-avtcore-rtp-circuit-breakers]           |
|             |                                                   |
|             | [I-D.ietf-avtcore-rtp-multi-stream-optimisation]  |
|             |                                                   |
|             | [I-D.ietf-avtcore-rtp-multi-stream]               |
|             |                                                   |
|             | [I-D.ietf-mmusic-trickle-ice]                     |
|             |                                                   |
|             | [I-D.ietf-tsvwg-rtcweb-qos]                       |
|             |                                                   |
|             | [I-D.reddy-mmusic-ice-happy-eyeballs]             |
|             |                                                   |
|  [RFC6904]  | [I-D.ietf-avtcore-srtp-encrypted-header-ext]      |
|             |                                                   |
|  [RFC7007]  | [I-D.ietf-avtcore-avp-codecs]                     |
```

Times Estimates (from Jennings)

[RFC7022]	[I-D.ietf-avtcore-6222bis]
[RFC7064]	[I-D.nandakumar-rtcweb-stun-uri]
[RFC7065]	[I-D.petithuguenin-behave-turn-uris]
[RFC7160]	[I-D.ietf-avtext-multiple-clock-rates]
[RFC7301]	[I-D.ietf-tls-applayerprotoneg]
[RFC7350]	[I-D.ietf-tram-stun-dtls]

Next-Generation Work

- ORTC (“WebRTC 1.1”)
- IMTC MANE AG
- Scalable (VP9, HEVC v2) and Simulcast support