More APIs?

Some functionalities that has been discussed, but we still have not added the APIs for it

All in a PeerConnection context
(MediaStreams dealt with in Media Cap TF)

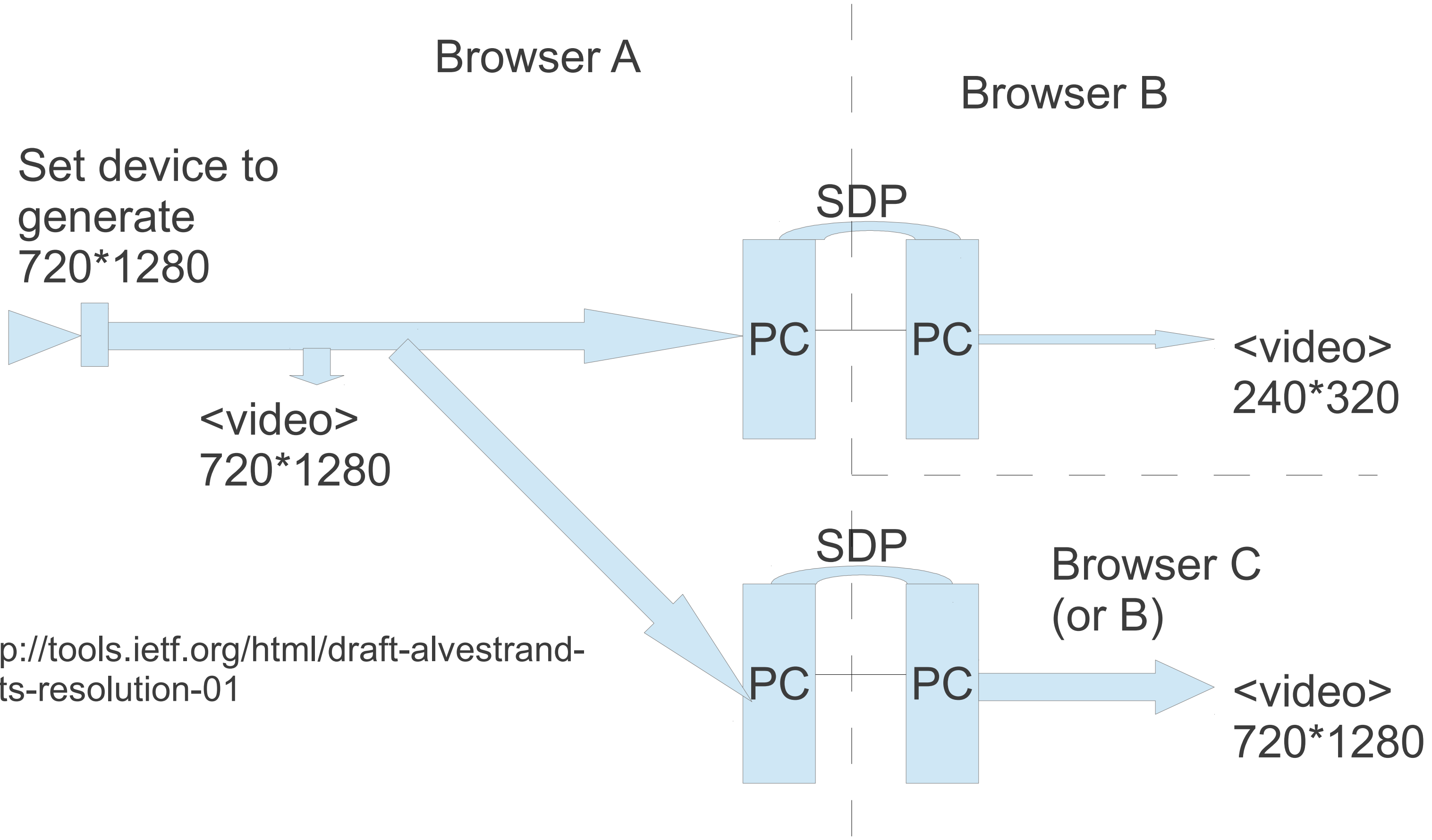
Topics

- What more API surface do we need
  - In v1
  - Can postpone (but have idea of how to solve) to v2
  - What we don't see a need for
- What should be the design principle?
-

Discussed/proposed

- Video height, width, framerate
- Receiver inform sender that a stream/track is not played (paused/unattached)
- Setting priority, max bw, min bw per track
- Inform sender side app about media flowing (or not), allowed bw, used bw, congestion, ….
- Pause/resume of tracks
- agc on/off, noise red on/off
- Rejection of offered MediaStream(Track)s
- AEC handling

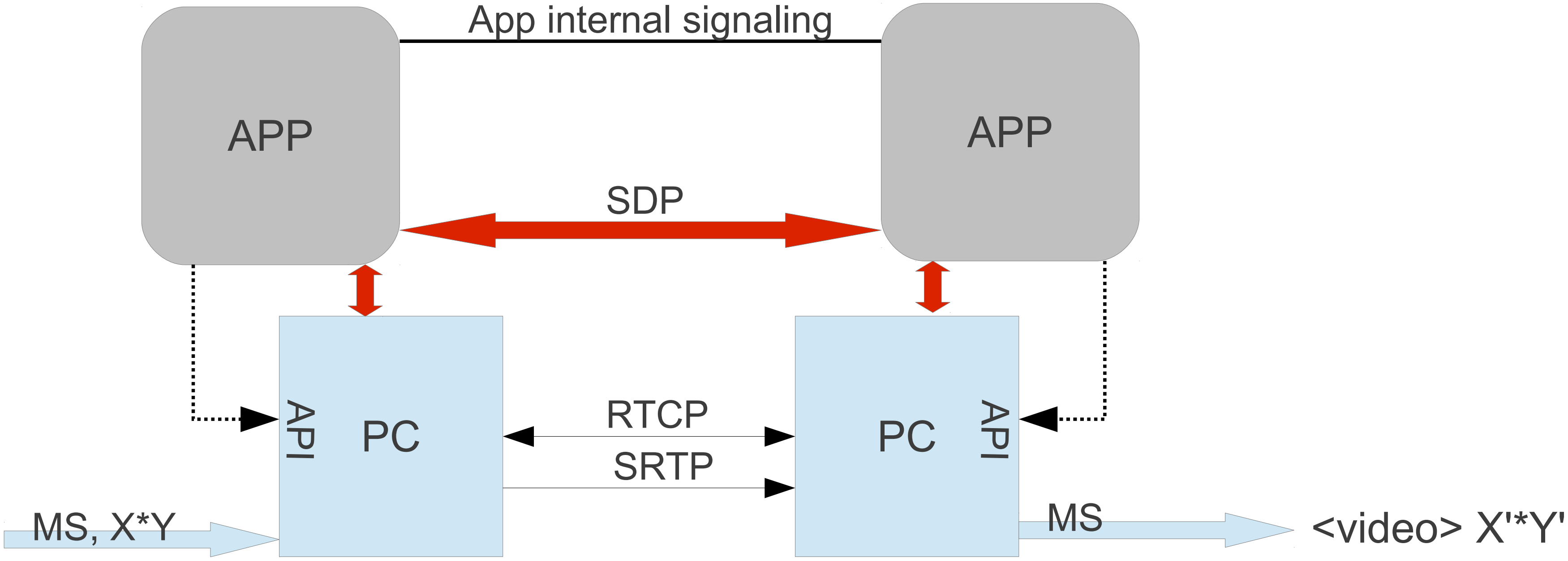
Video width, height (framerate?)



Ref to http://tools.ietf.org/html/draft-alvestrand-constraints-resolution-01

Is this a valid use-case?

Width, Height (rate) options

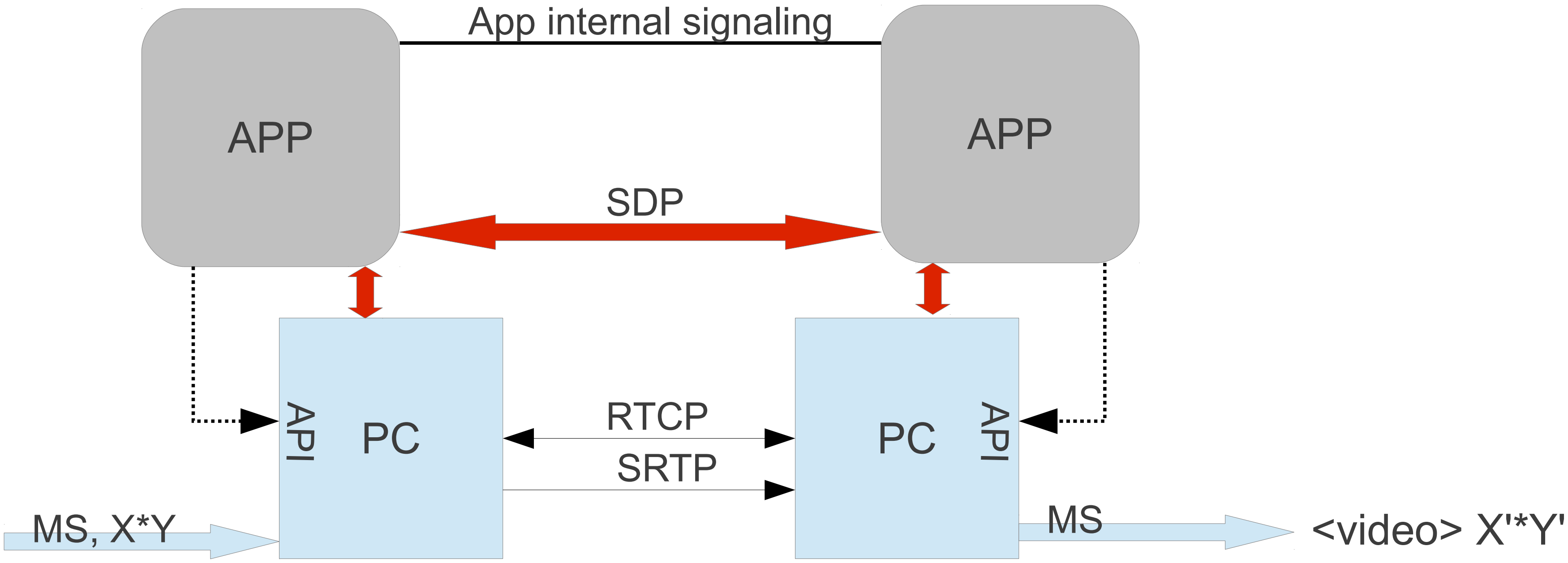


- API: Sending PC, Receiving PC, both, none
  - None = the receiving UA decides based on consumer
- Signaling: app internal, SDP or RTCP

Options

- No API, UA handles: signal via SDP or RTCP
- API at sending PC only
  - App internal signaling to carry from receiver
- API at receiving PC only: signal in SDP or RTCP
  - Receiving app does not know; sending PC adjusts
  - Receiving app gets informed (but has no influence); sending PC adjusts
- API at both ends
  - Dual control – who's in charge?
  - Or, remote API setting results in event at sending side only; sending app in control (using its API)

Receiver inform sender about media not used (unattached/paused)

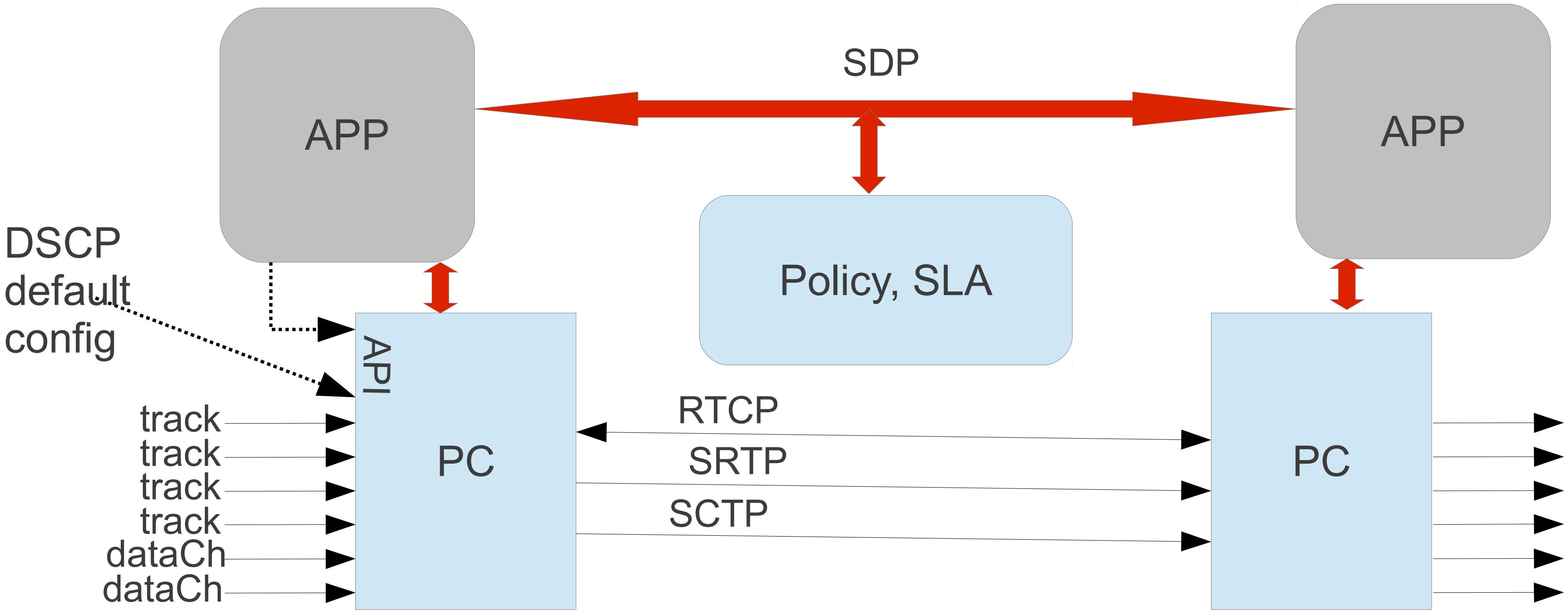


- API: Sending PC, Receiving PC, both, none
  - None = the receiving UA decides based on consumer
- Signaling: app internal, SDP or RTCP

Options (repeated)

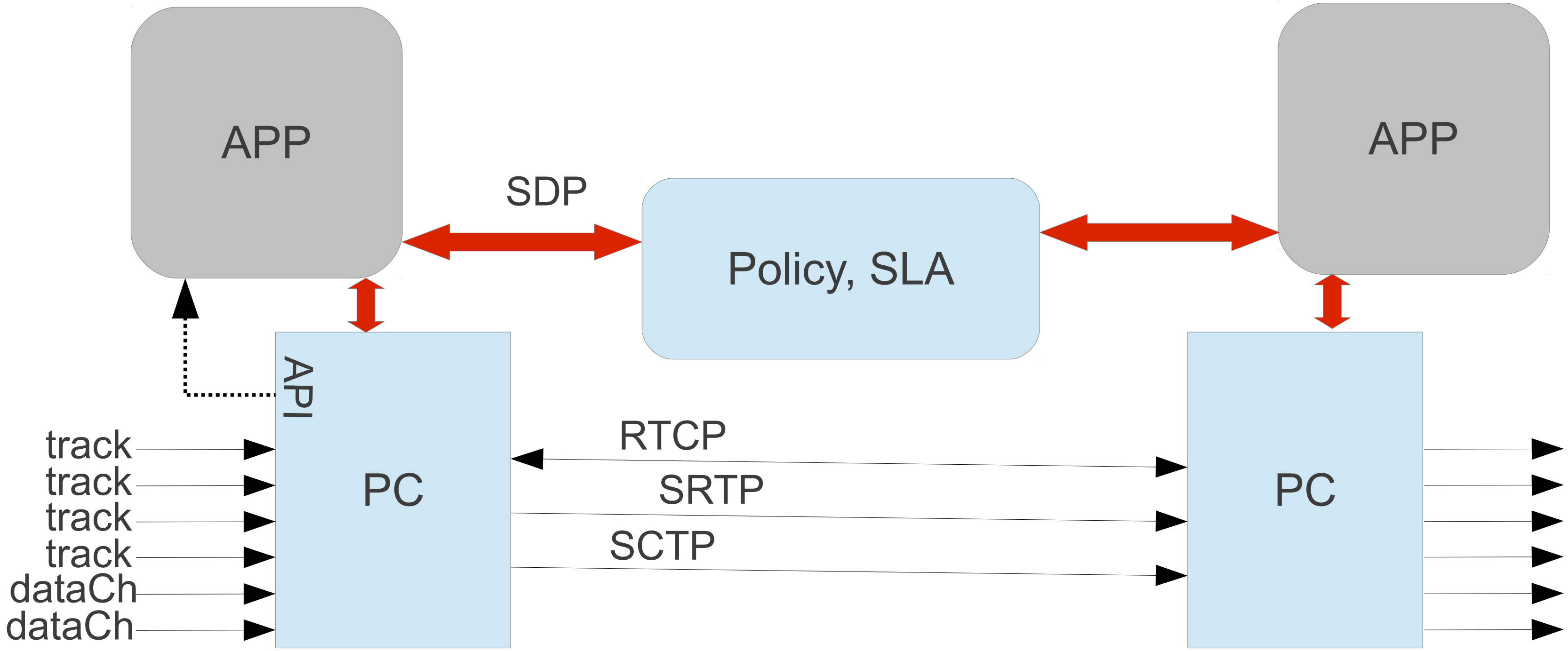
- No API, UA handles: signal via SDP or RTCP
- API at sending PC only
  - App internal signaling to carry from receiver
- API at receiving PC only: signal in SDP or RTCP
  - Receiving app does not know; sending PC adjusts
  - Receiving app gets informed (but has no influence); sending PC adjusts
- API at both ends
  - Dual control – who's in charge?
  - Or, remote API setting results in event at sending side only; sending app in control (using its API)

Requesting BW, Priority, DSCP, QoS



- https://www.w3.org/Bugs/Public/show_bug.cgi?id=15861
- Transport provider consent
- SDP good place to signal
  - Trust
  - Stats API to verify

Feedback on flowing, bw allocated, bw used, congestion situation



- https://www.w3.org/Bugs/Public/show_bug.cgi?id=
- Stats API?

A couple of small ones

- Sender side pause/resume of tracks
  - Currently we have enable/disable on MediaStreamTrack object (but does not fit that well with media element design)
  - Alternative to have this on PeerConnection instead
    - Less ambiguity
    - Allows control per peer
- AGC on/off, Noise Reduction on/off
  - Sender side only, no signaling, simple

Reject MediaStream(Track)s

- Currently (at least without SDP munging) not possible
- We could add an API
  - The SDP answer would in one way or another tell the sending UA that those MS(T)s should not be part of the session
- Question: is the sending app informed? How?
- Question: what is the need if the media is not transmitted anyway?
  - Free up resources?

AEC

- A PeerConnection must make sure that any media played do not leak into any outgoing MediaStreams
- Should this be possible to disable (e.g. when using headphones)
-

SDES

- I'll skip this until after the IETF discussion has concluded on whether this will be a rtcweb feature or not

What	When	How	Signaling
Video height, width, framerate	?	API(where)? Automatic?	Depends
Receiver inform sender track not used	?	API? Automatic?	Yes
Request priority, bw, … per track	?	Sender side API	Yes
BW, congestion feedback	?	Sender side API?	Yes
Pause/resume tracks	?	Sender side API	Yes
AGC, NR on off	?	Sender side API	No
Reject MediaStream(Track)s offered	?	Receiver side API	Yes
AEC	?	Receiver side API	?
SDES	?	Sender side	?

What	When	How	Signaling
Video height, width, framerate	?	Receiver side driven, automatic, but possible to override by API	Yes
Receiver inform sender track not used	?	Receiver side driven, automatic	Yes
Request priority, bw, … per track	?	Sender side API	Yes
BW, congestion feedback	?	Sender side API	Yes
Pause/resume tracks	?	Sender side API	Yes
AGC, NR on off	?	Microphone API; investigate if the same mike can deliver two versions	No
Reject MediaStream(Track)s offered	?	Receiver side API	Yes
AEC	?	MediaStreamTrack API	?
SDES	?	Sender side	?