More APIs?

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

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

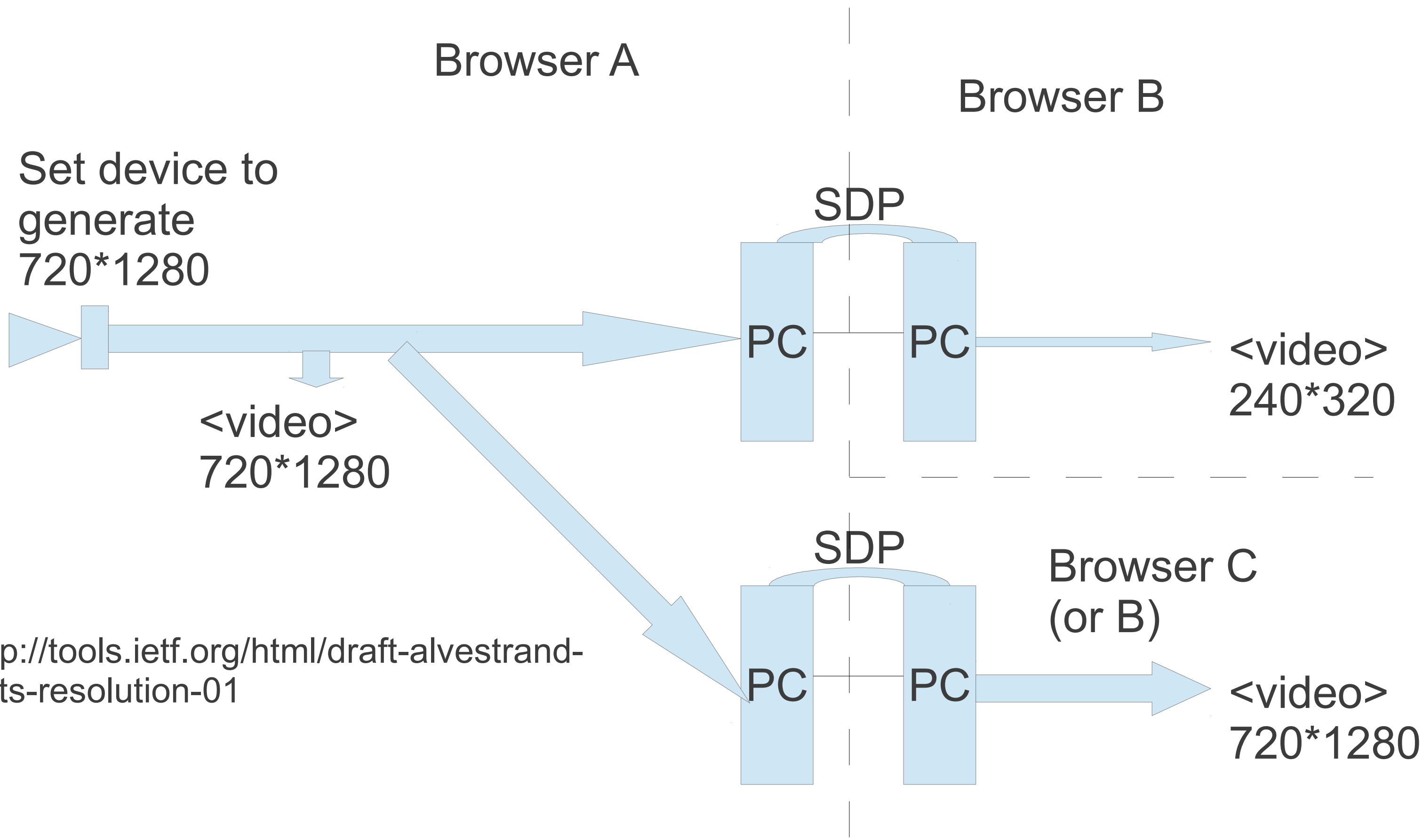
Topics

- 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

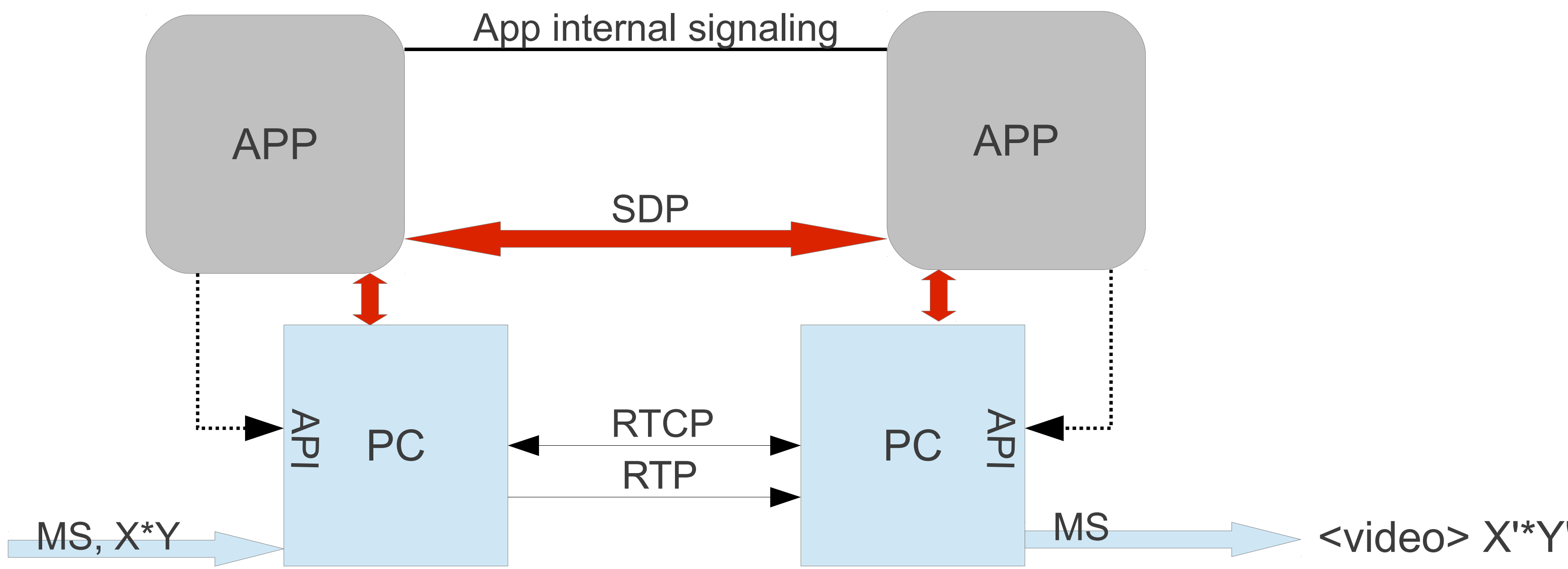
Video width, height (framerate?)



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

Is this a valid use-case?

Width, Height (rate) options

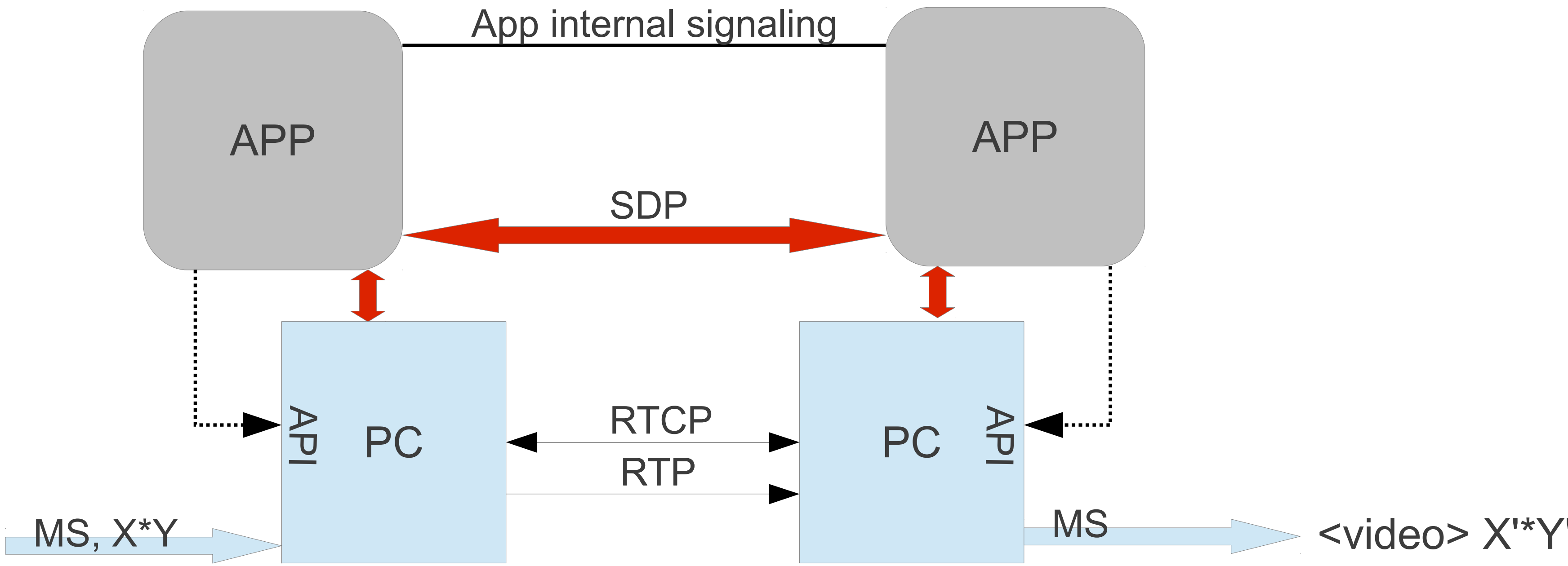


- API: Sending PC, Receiving PC, both, none
  - None = the receiving UA decides based on consumer
- Signaling: app internal, SDP or RTCP

Options

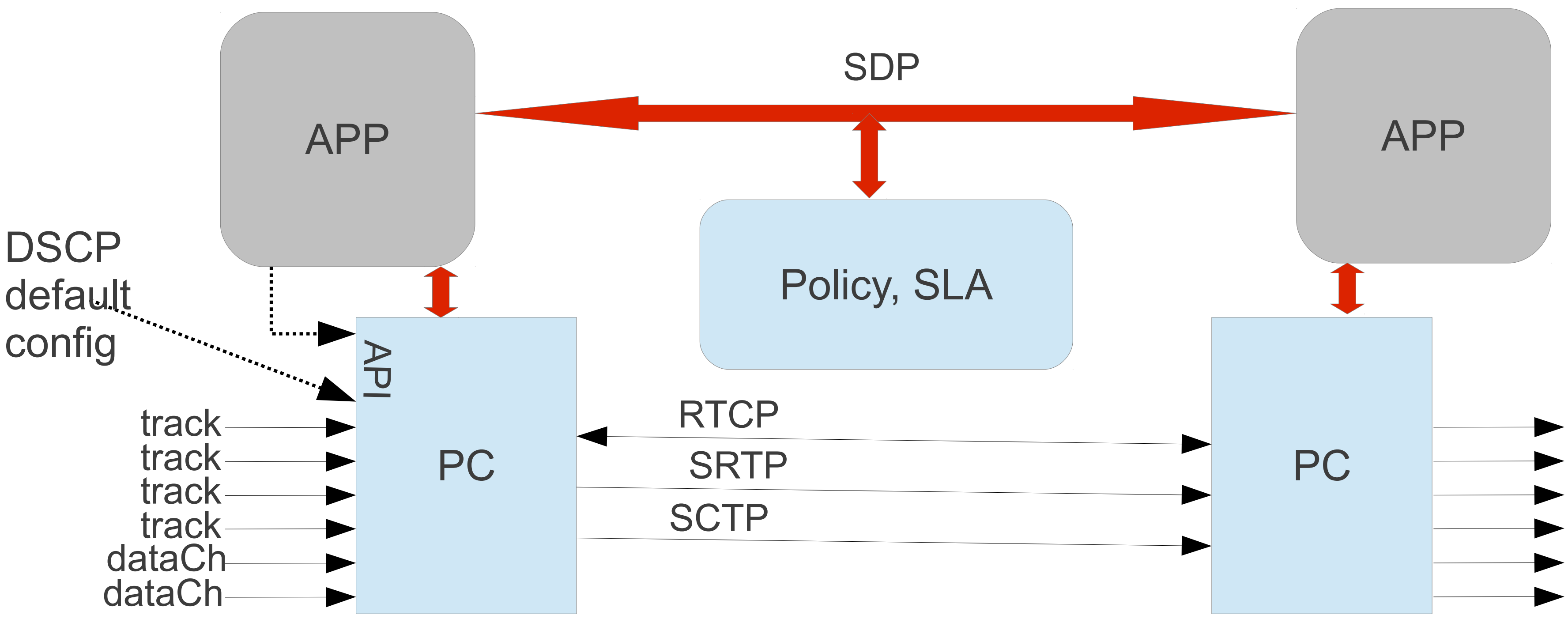
- No API, UA handles: signal via SDP or RTCP
- API at sending PC only
  - App internal signaling to carry from receiver
- API at receiving PC only: signal in SDP or RTCP
  - Receiving app does not know; sending PC adjusts
  - Receiving app gets informed (but has no influence); sending PC adjusts
- API 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)



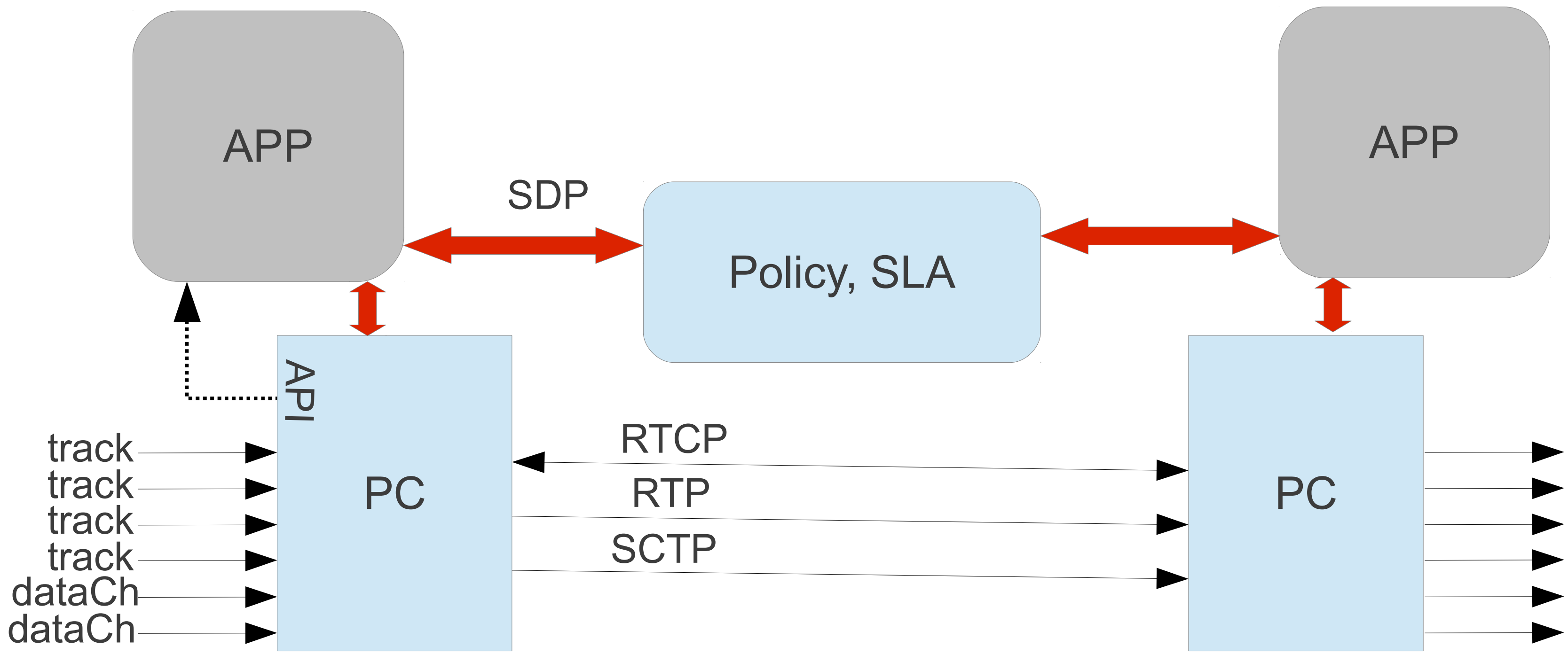
- API: Sending PC, Receiving PC, both, none
  - None = the receiving UA decides based on consumer
- Signaling: app internal, SDP or RTCP

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 new media element design)
- 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
- Open Question: is the sending app informed? How?
- Question: what is the need if the media is not transmitted anyway?

AEC

- A PeerConnection must make sure that any media received and played do not leak into outgoing audio streams (if any)
- 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	?

Basic API options

- Setting per track:
  - PeerConn method, using track as selector and constraints
    - pc.applyConstraints(track, constraints);
  - Using stand alone objects
    - speakCamTransport.dimension.request(width, height);
- Checking:
  - PeerConnection
    - pc.getStatus(track, function () {do something}); //getStats?
  - Stand alone object
    - Var status = speakCamTransport.flowing;
- Notification of change:
  - Event fired?

Current support (Sender side per track)

- Setting height, width, agc, noise red, …
  - Constraints at addStream() time
  - Can't change, doesn't handle addTrack()
- Pause/resume
  - Enable/disable track?
- Setting priority, max bw, min bw
  - Not supported (could use constraints at addStream)
- Being informed about flowing, allowed bw, congestion
  - Not supported, could in principle use stats

API options (non exhaustive) width/height

GetUserMedia =>camStream
var speakerCam = camStream.videoTracks[0]; //if length <>0

- Constaints at addStream
  - pc.addStream(camStream, constraints);
  - How
- Setting using a selector a la stats (would be analogous if applied on the receiver side):

  pc.addStream(camStream);
  pc.setDimension(speakerCam, 320*240);
  pc.getDimension(speakerCam);

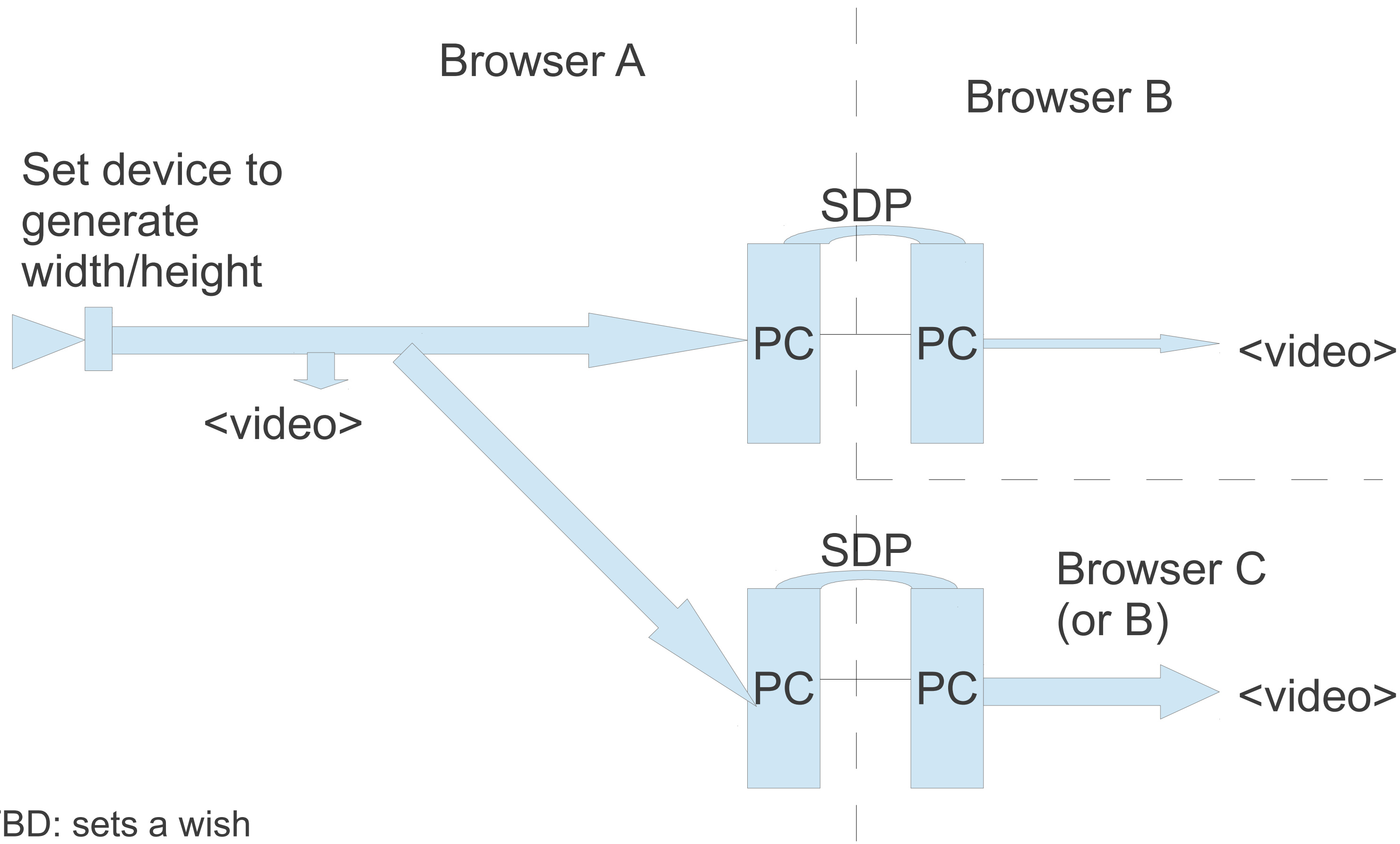
  - Or using constraints

  pc.addStream(camStream);
  pc.applyConstraints(speakerCam, {constraints});
  pc.getStats(speakerCam, successCb);

- Special control object (analogous if applied on the receiver side):

  pc.addStream(camStream);
  outBndStream = pc.localStreams[pc.localStreams.length – 1];
  outBndStream.videoTracks[0].dimensions.request(320*240);

Sender side: bw, priority



- API: TBD: sets a wish
- BW: SDP bandwidth attributes (establishes agreement between endpoints and connection provider(s))
  - Can lead to a lower allowed bw than wanted allocated
- Priority:
  - Per track
  - Influence congestion control, DSCP, ….
  -

(Stream/track) receiver side

- “No consumer”
- Display size (width, height)
- Automatic, or via API?

Receiver side

- Allow app to reject an offered MediaStream
  - On MediaStream or MediaStreamTrack level?
- inform the sender of used / useful width/height
- tell the sender that a stream/track is not played (paused/unattached)
  - Allows saving transmission

Unclear which side

- Echo cancellation

Unclear which side

- Echo cancellation