




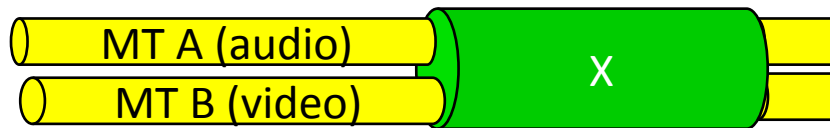
Media Source identifier

Something we need?

Local use

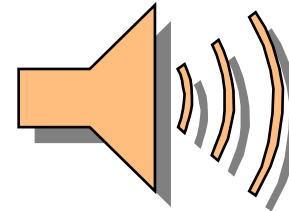
-  MT = MediaTrack (=MediaStreamTrack)
-  MTC = MT Collector (=MediaStream)
-  PC = PeerConnection

App

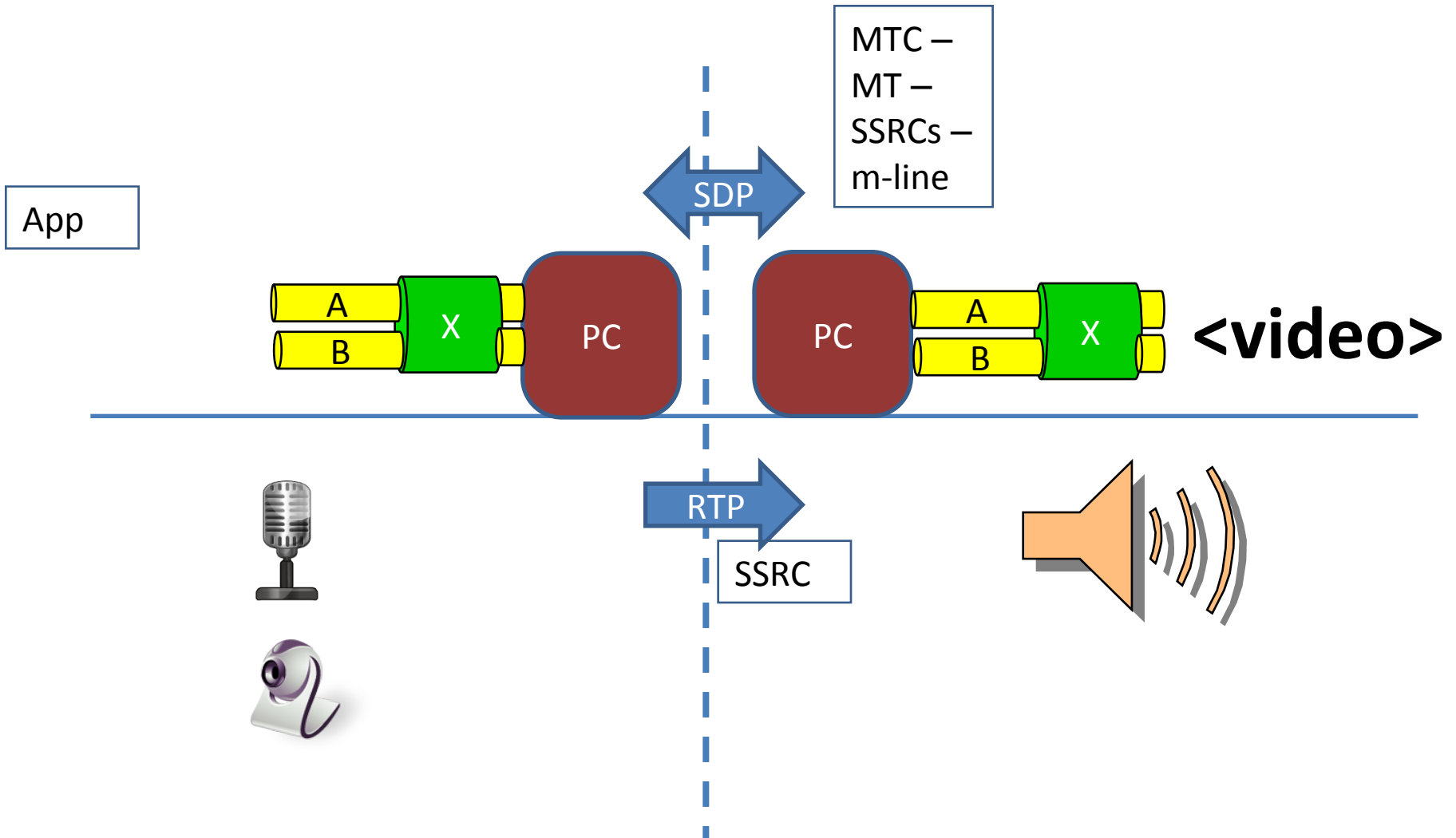


<video>

System



To a peer



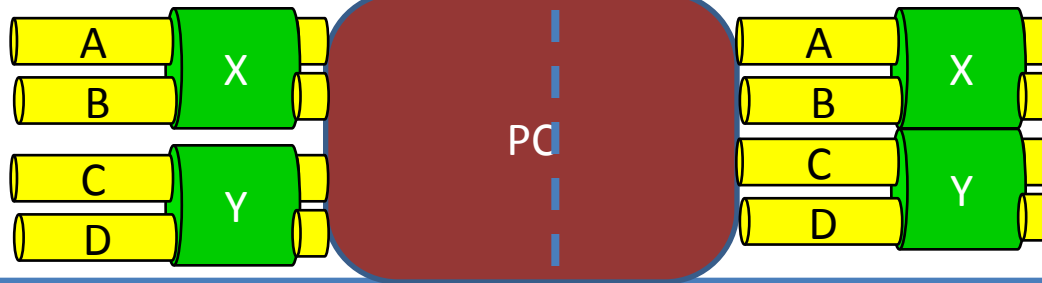
To a peer

A, C: microphone

B: Front cam

D: Rear cam

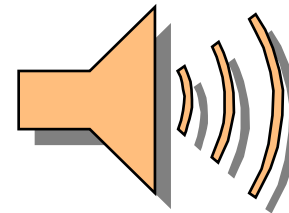
App



Front



Rear



A and C must be played
in full sync to avoid
glitches when switching
source of video element
-> all tracks in full sync

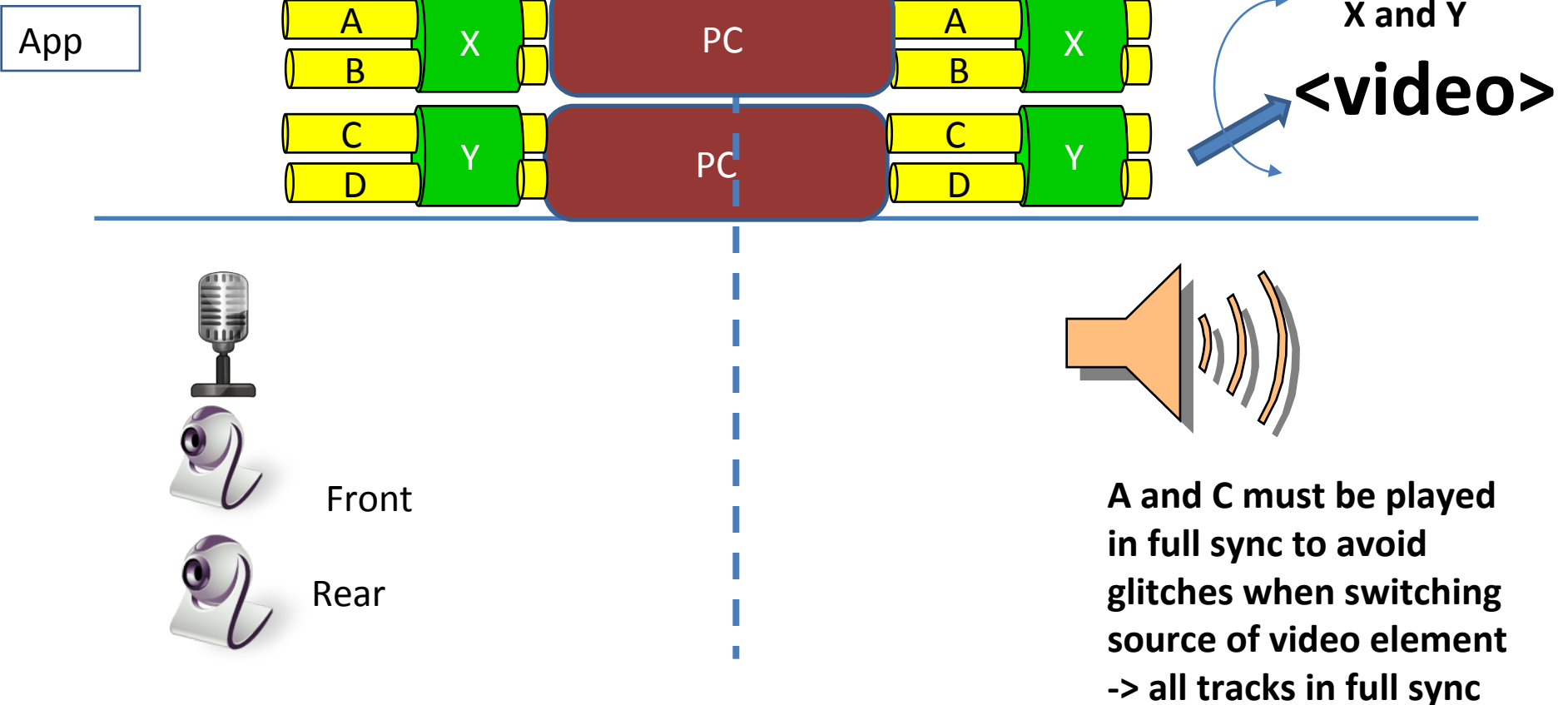
To a peer

The track represent:

A, C: microphone

B: Front cam

D: Rear cam



To a peer

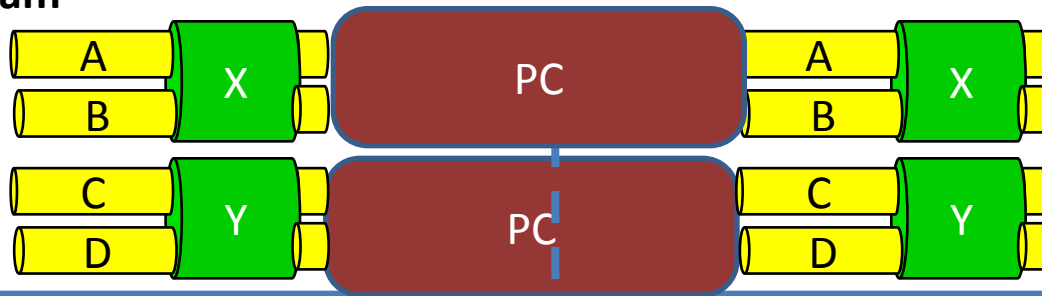
The track represent:

A, C: microphone

B: Front cam

D: Rear cam

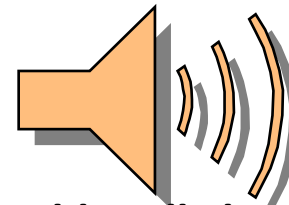
App



Front



Rear



Should really both A
and C play?; makes
more sense to detect
that they are the same
and play one
If both play they must
be in perfect sync

To a peer via a peer

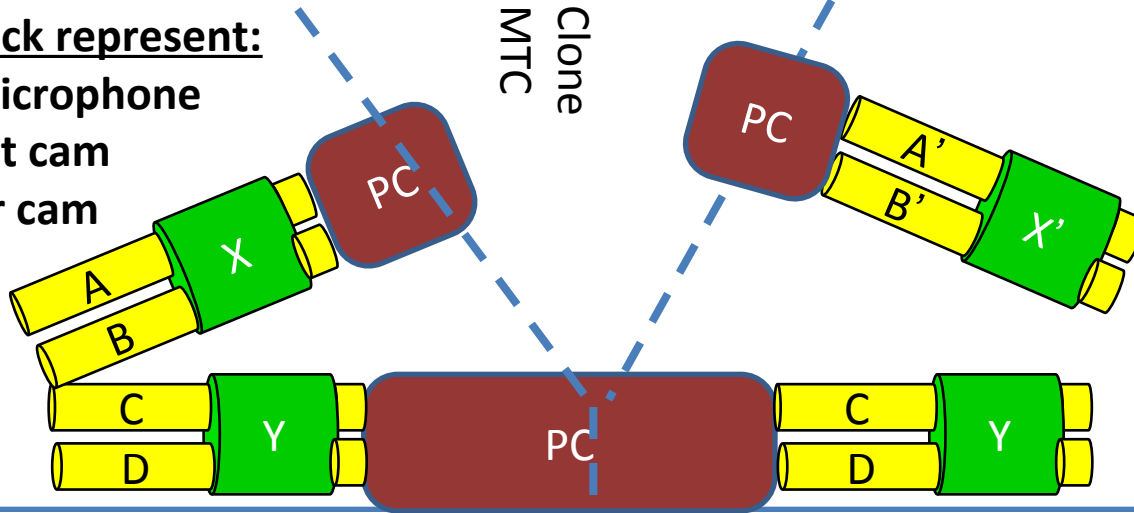
The track represent:

A, C: microphone

B: Front cam

D: Rear cam

App



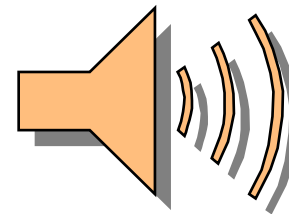
<video>

<video>



Front

Rear



Should it not also in this case be possible to detect that C and A' represent the same source?

Loopback

The track represent:

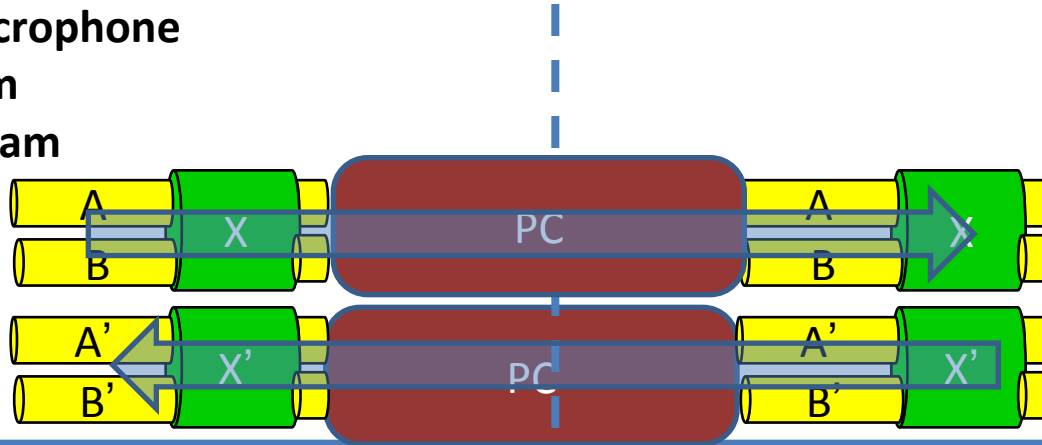
A, A': microphone

B, B': cam

D: Rear cam

App

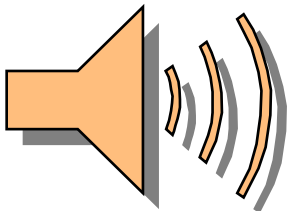
<video>



Clone MTC X -
> X'



Should it be possible to
detect loopback?
Can't be done from MT
or MTC Id's



Summary

- (There is a need for the **application** to be able to identify
 - MediaStream's
 - MediaStreamTrack's)
- There is a need to synchronize
 - Across MediaTrackCollector's
 - Across PeerConnection's
- However, there is also seem to be a need for the **system** to be able to identify sources
 - SSRC can't be used
 - Must be different (for the same source) to enable individual handling if in the same RTP session
 - Imageattr
 - Pause/resume = enable/disable
 - May have to be different even if in different RTP sessions (due to collisions)
 - (Layered coding, simulcast
 - FEC)
 - No need for the app to know it