



NMOS Advanced Streaming Architecture

NDI, SRT, RTSP and more ...

Alain Bouchard, ing



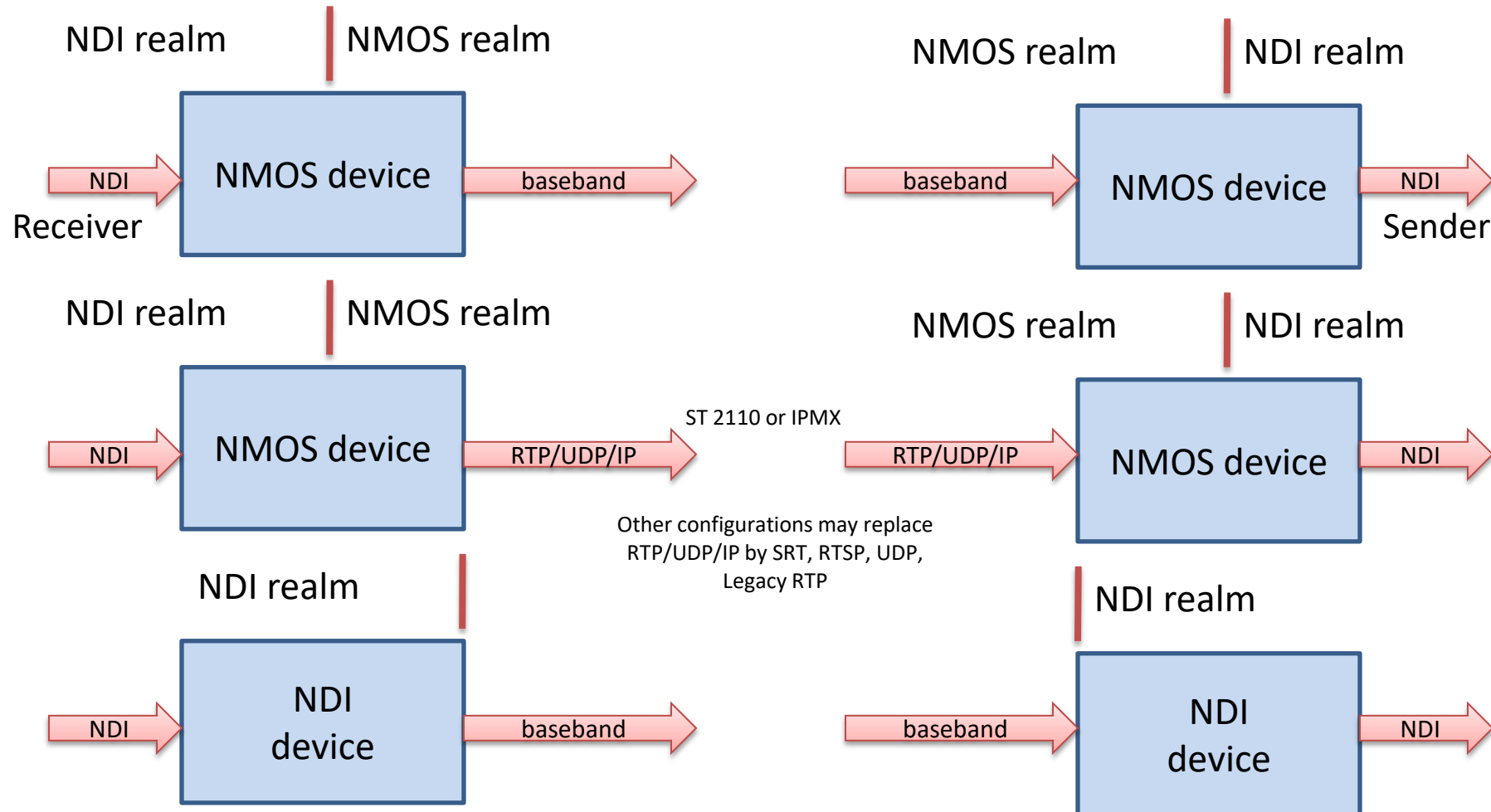
Public GitHub Repository

- <https://github.com/alabou/NMOS-MatroxOnly>
 - README.md
 - NMOS With NDI.md
 - NMOS With SRT.md
 - NMOS With RTSP.md
 - NMOS With UDP.md
 - NMOS With H.222.0

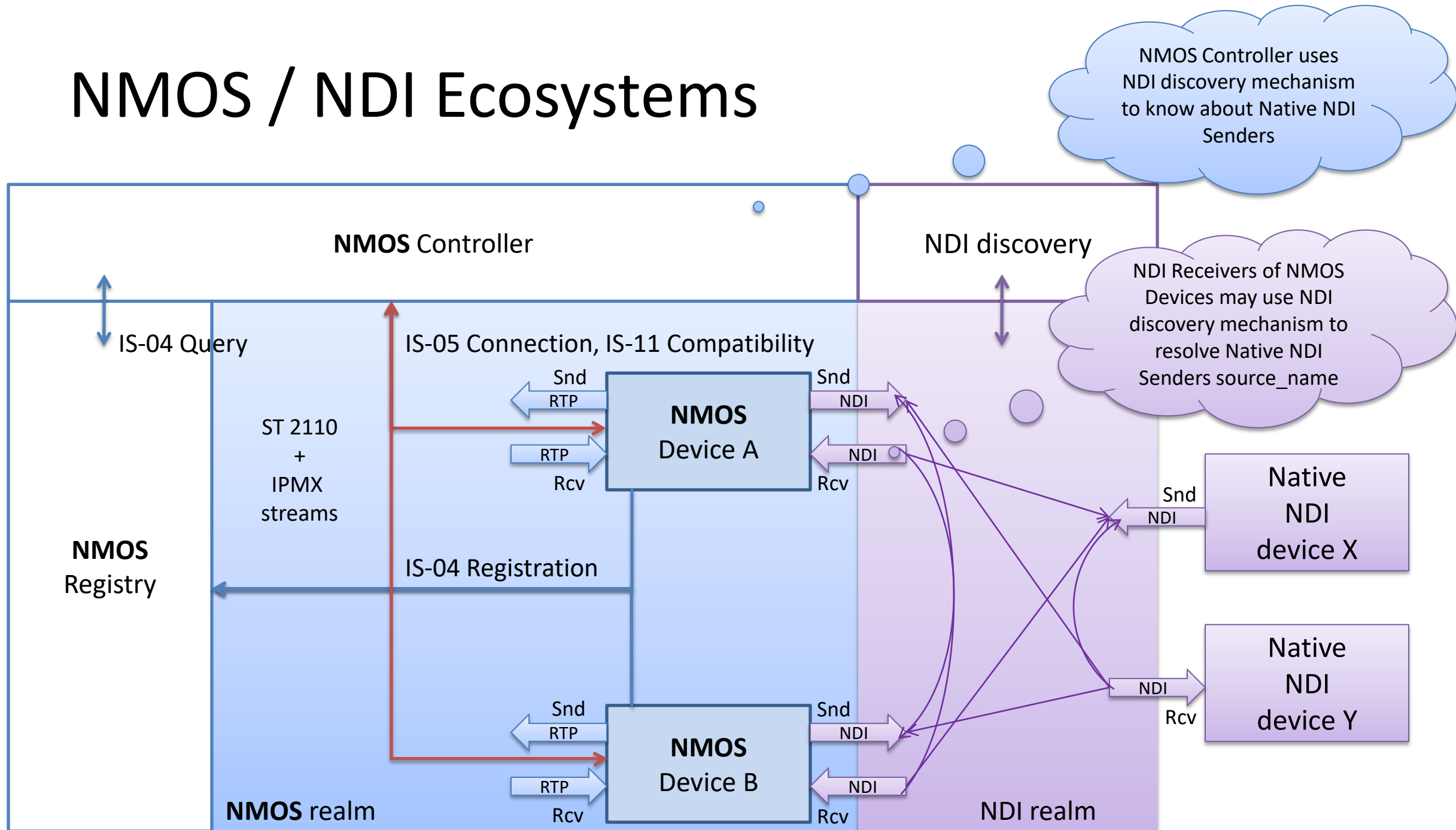
NDI as transport from/to NMOS realm

- The objective is **not**
 - To use NDI as a generic NMOS transport
 - What goes to the NDI realm becomes out-of-scope for NMOS
 - What comes from the NDI realm origins out-of-scope for NMOS
- The objective is
 - To use NDI to capture from the NDI realm
 - Similar to capture from SDI, HDMI
 - To use NDI to playback to the NDI realm
 - Similar to playback to SDI, HDMI

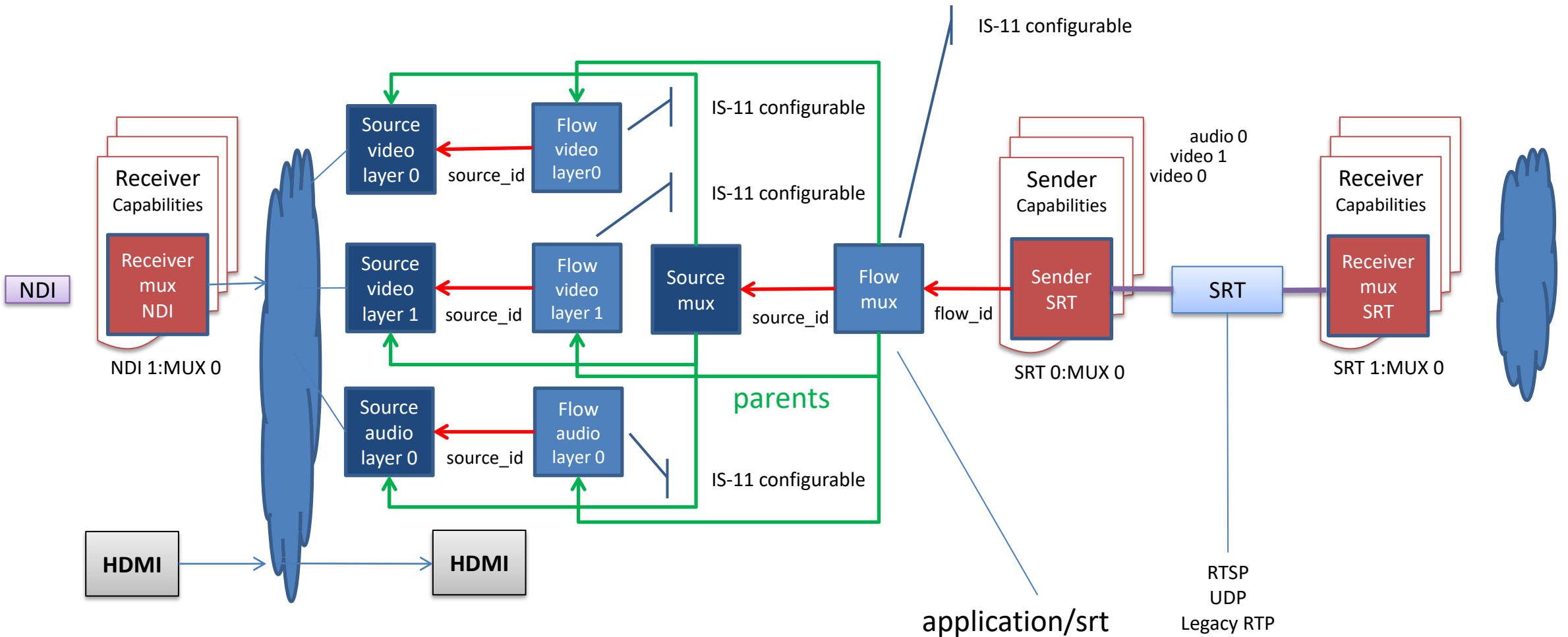
Devices with NDI transport



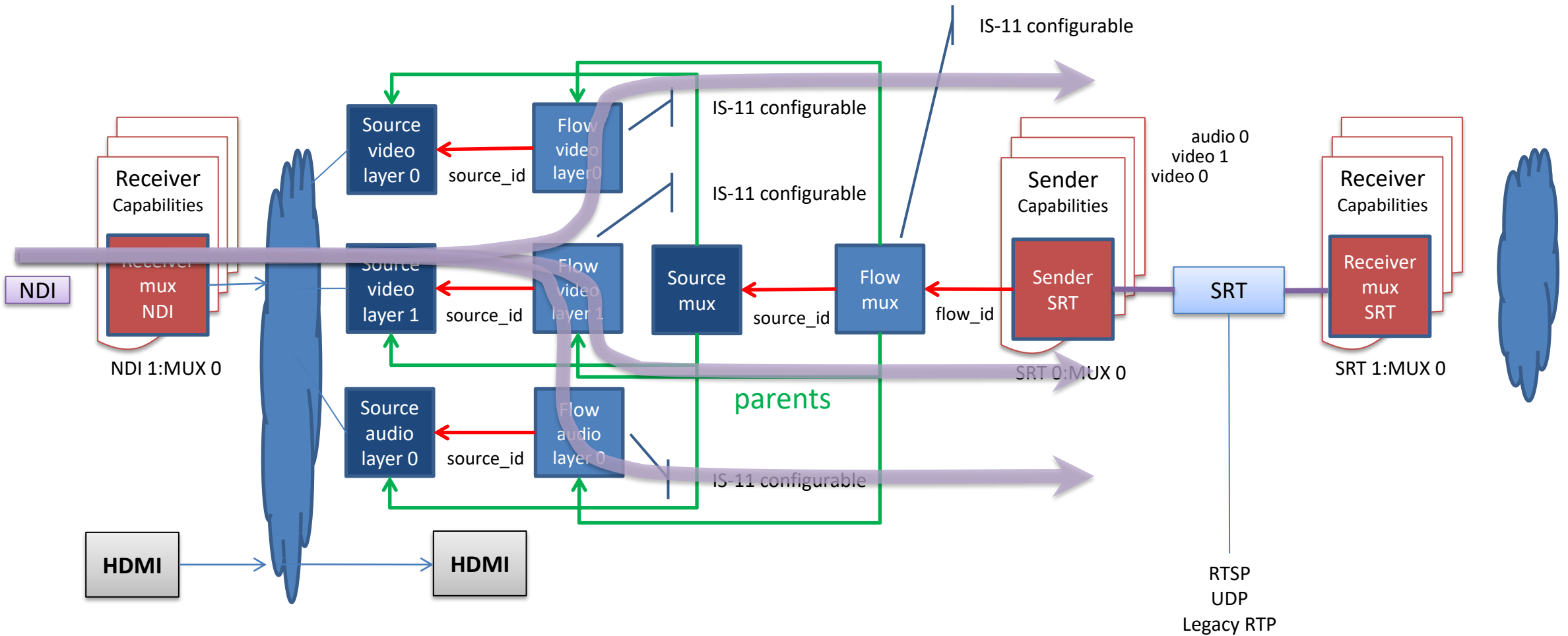
NMOS / NDI Ecosystems



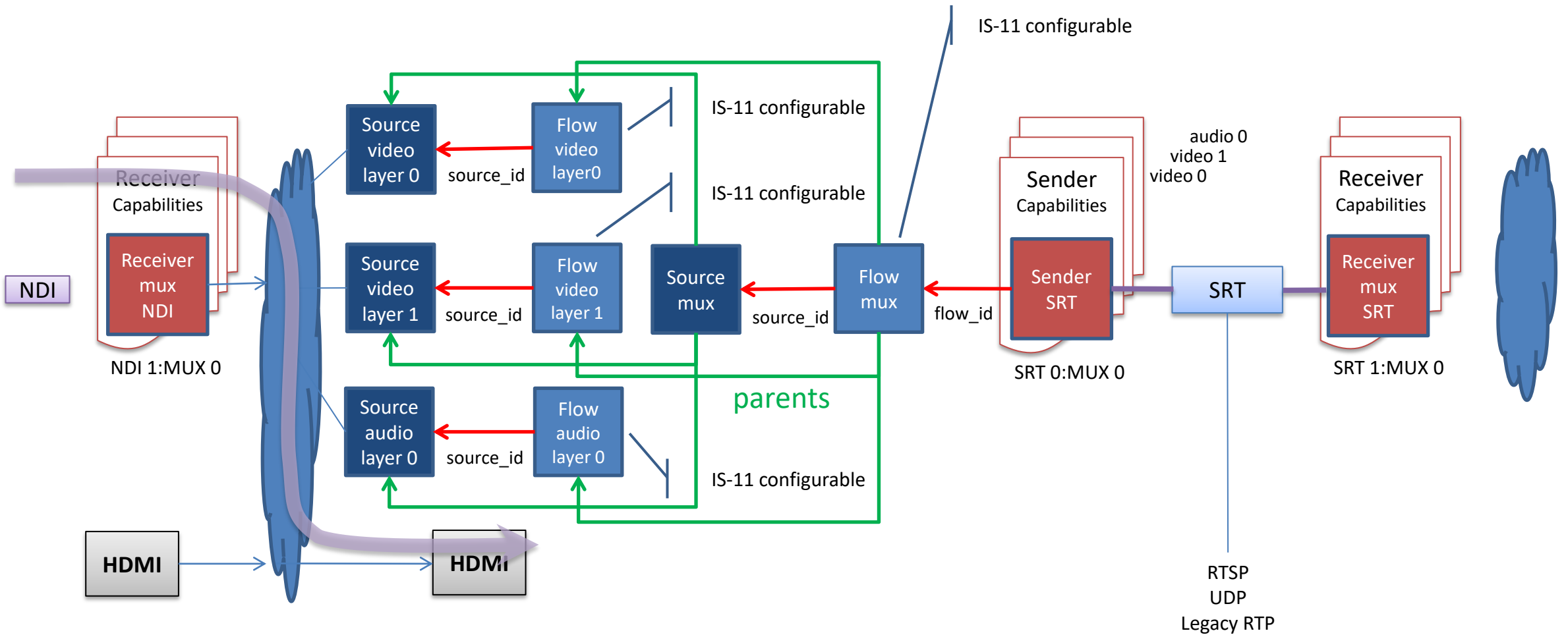
Multiplexed streams over NDI



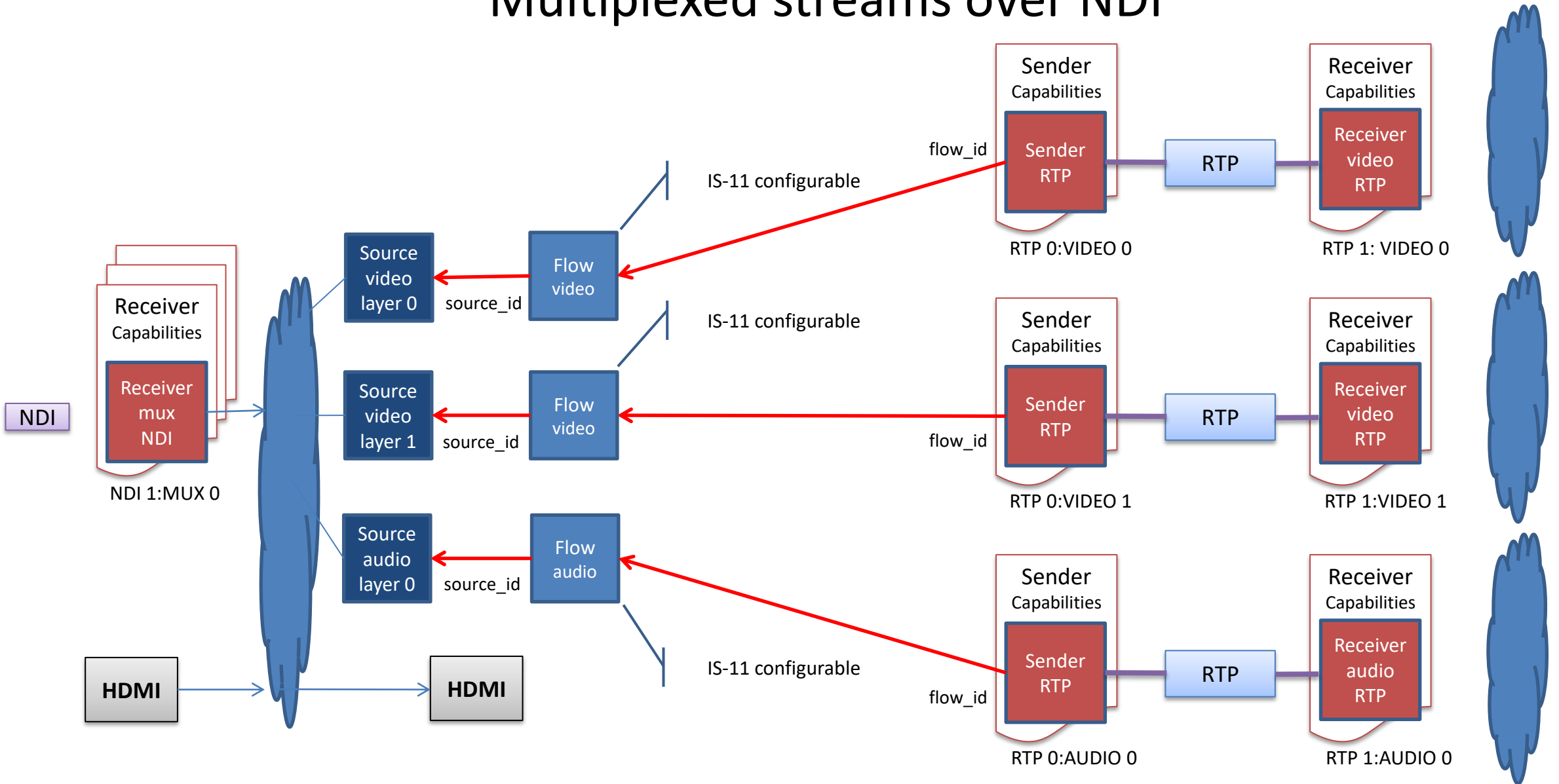
Multiplexed streams over NDI



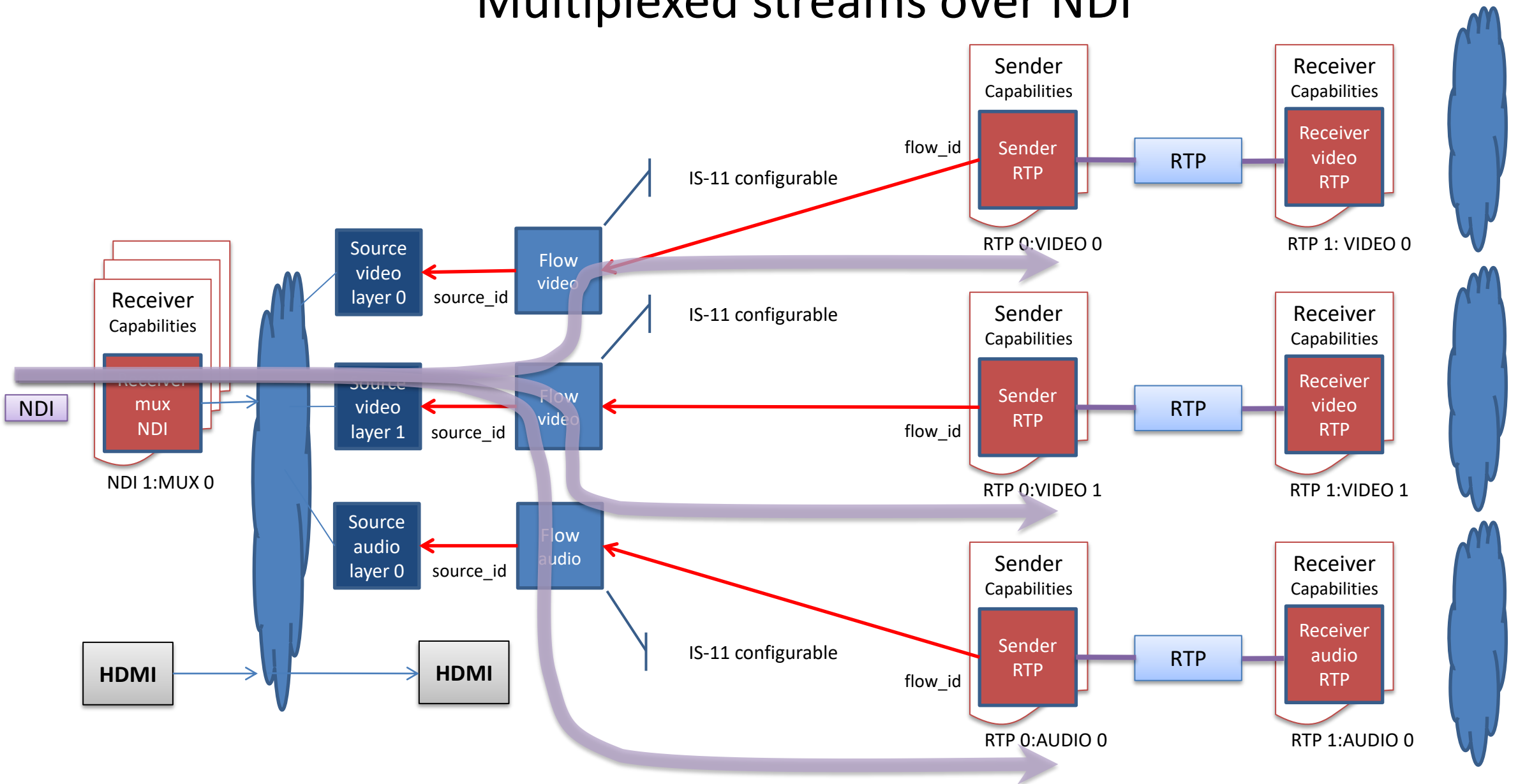
Multiplexed streams over NDI



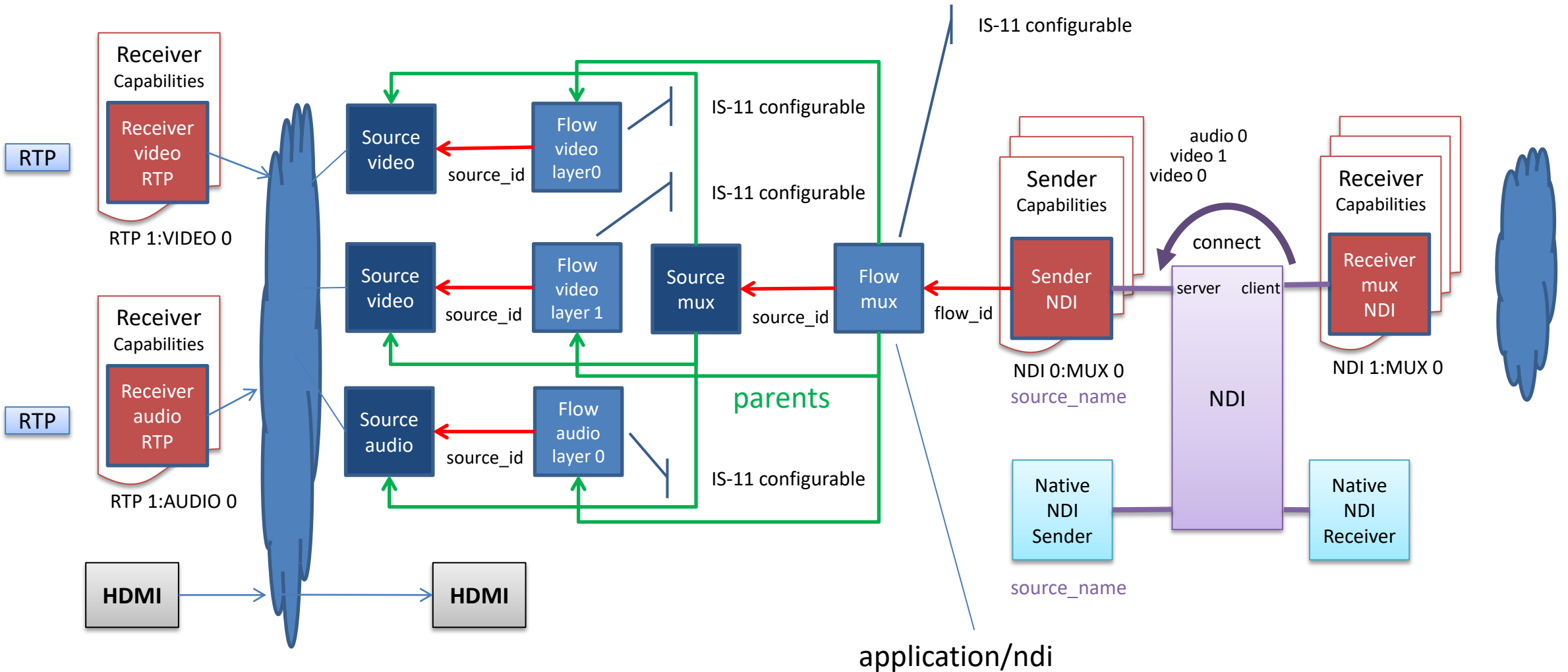
Multiplexed streams over NDI



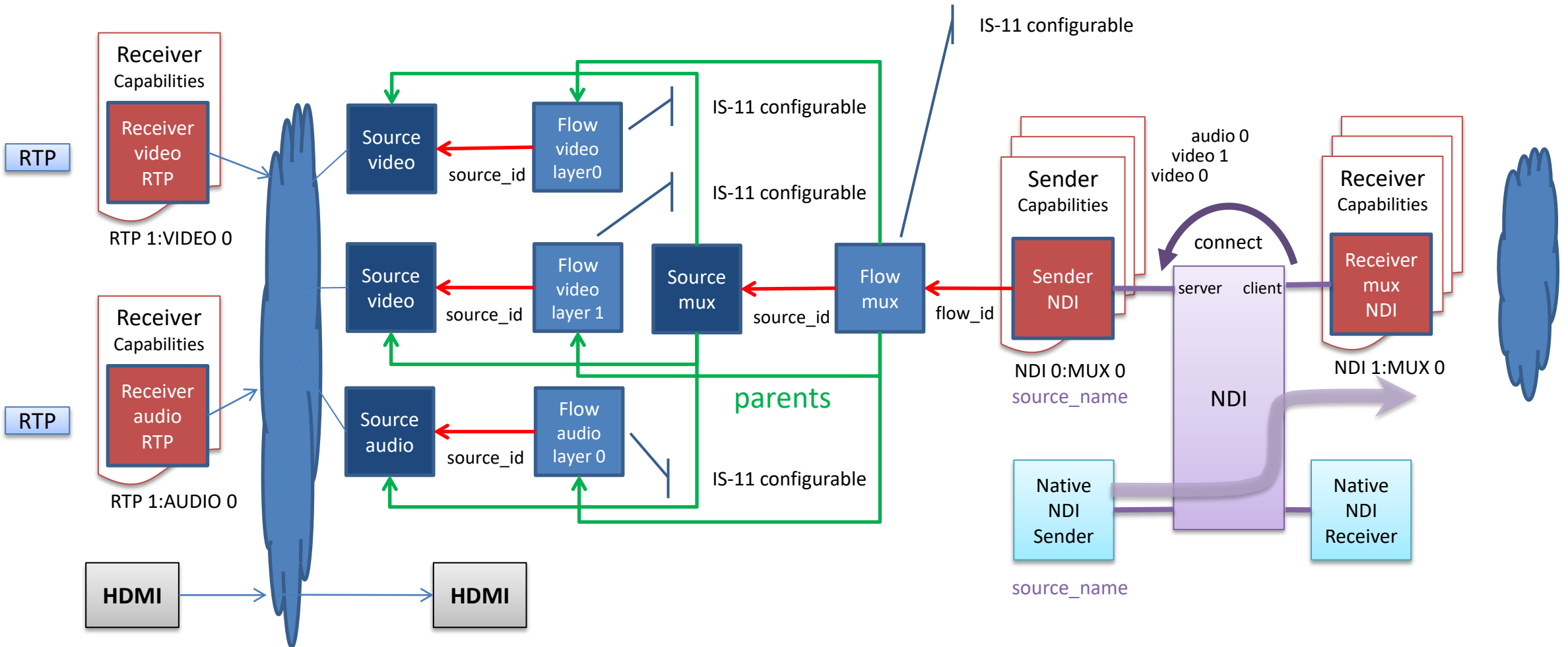
Multiplexed streams over NDI



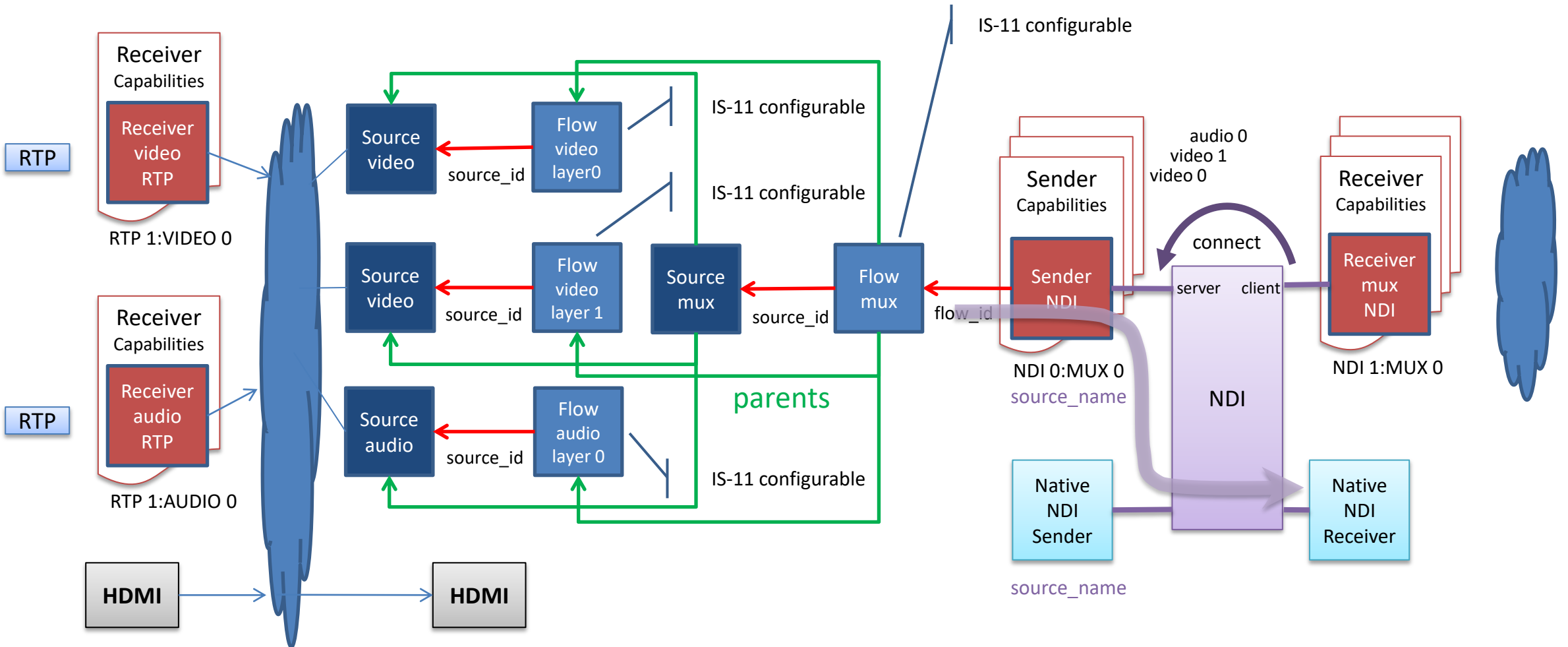
Multiplexed streams over NDI



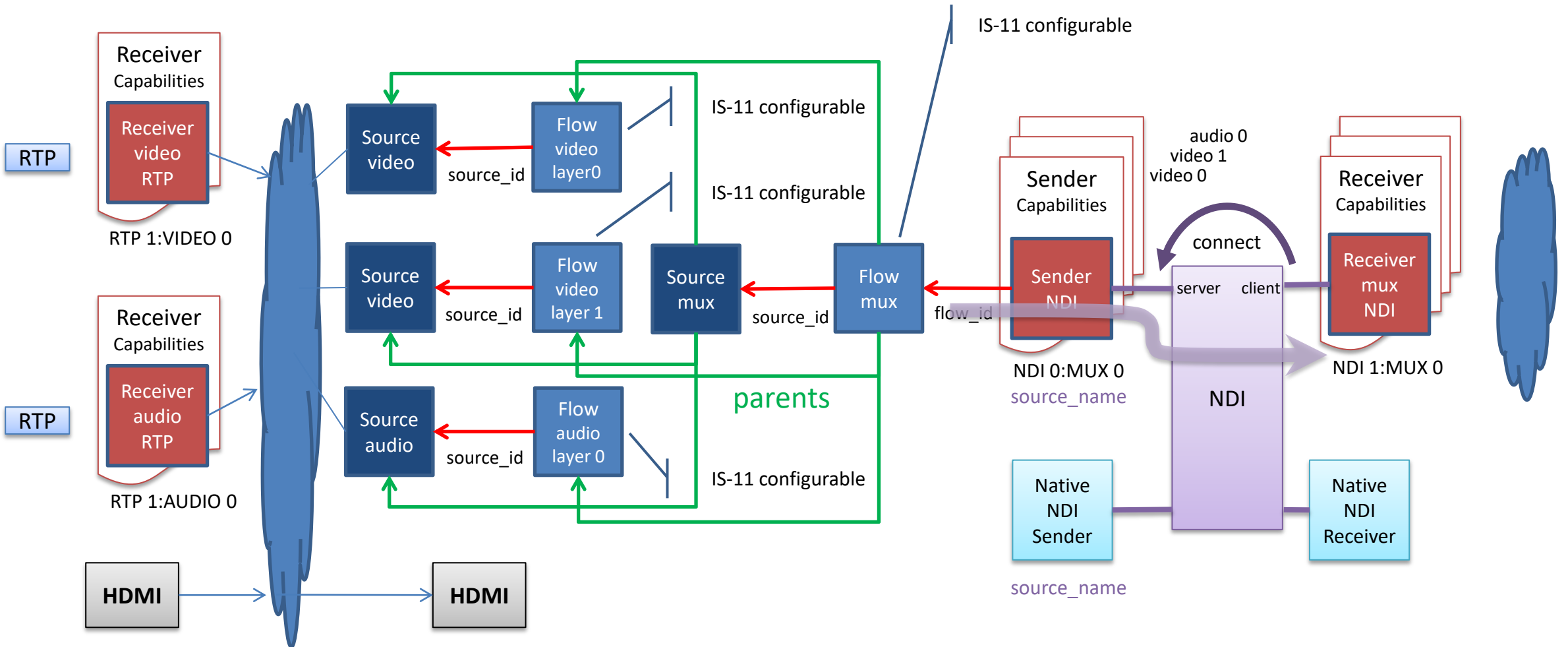
Multiplexed streams over NDI



Multiplexed streams over NDI



Multiplexed streams over NDI



NDI Use-Cases

Sender	Receiver	Connection
NMOS	NMOS	The NMOS Controller discovers Senders and Receivers through the Registry using the IS-04 APIs. It configures the Sender using IS-11 active constraints and activates it using IS-05. The Controller then connect a Receiver to the source of a Sender by providing transport parameters with IS-05 (no SDP).
NMOS	Native	The NMOS Controller discovers Senders through the Registry using the IS-04 APIs. It configures the Sender using IS-11 active constraints and activates it using IS-05. Native NDI Receivers discover NMOS controlled NDI Senders via the NDI discovery mechanism and connect to a Sender's source using the native NDI control protocol.
Native	NMOS	The NMOS Controller discovers Native NDI Senders using the NDI discovery mechanism, and NMOS controlled NDI Receivers through the Registry using IS-04 APIs. The Controller connects the Receiver to the Native NDI Sender's source by providing transport parameters using IS-05 (no SDP).
Native	Native	The NMOS Controller is not involved. Native NDI Senders and Receivers operate entirely within the NDI ecosystem, using their own discovery and control mechanisms.

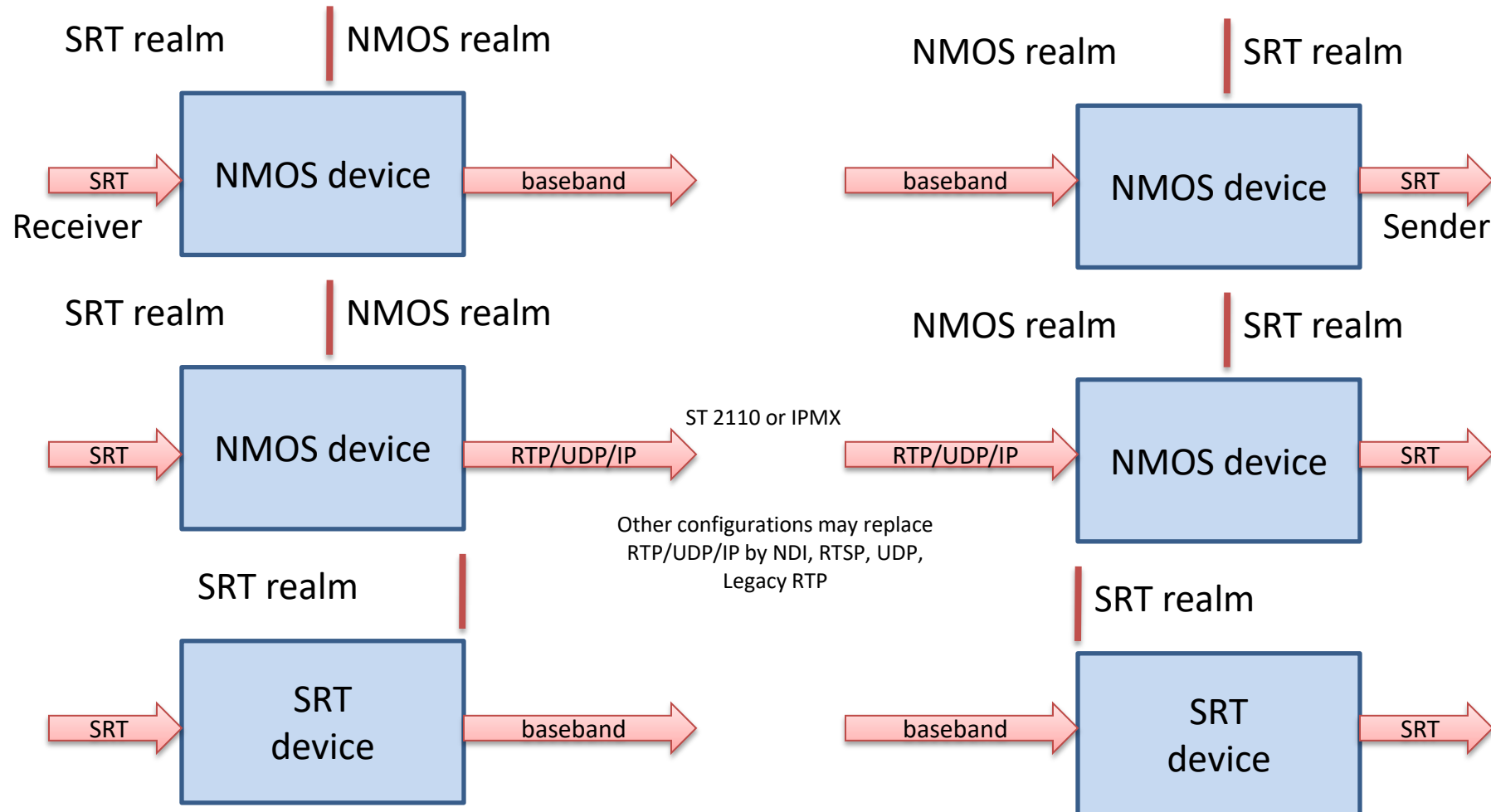
Privacy Encryption

- NDI does not support encryption
 - A device should not allow forwarding a stream received with privacy encryption to NDI unless expressly allowed by the administrator.

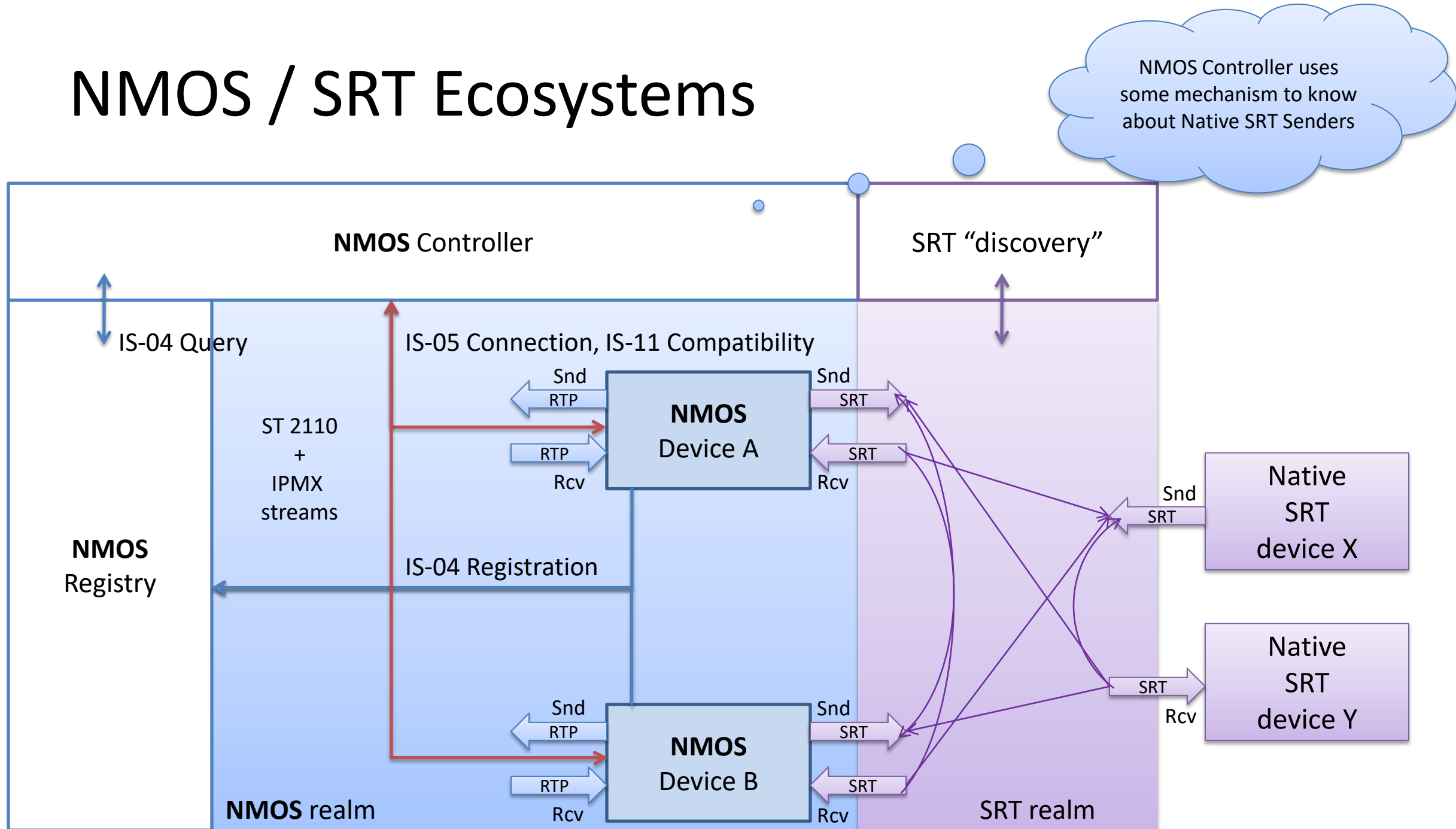
SRT as transport from/to NMOS realm

- The objective is **not**
 - To use SRT as a generic NMOS transport
 - What goes to the SRT realm becomes out-of-scope for NMOS
 - What comes from the SRT realm origins out-of-scope for NMOS
- The objective is
 - To use SRT to capture from the SRT realm
 - Similar to capture from SDI, HDMI
 - To use SRT to playback to the SRT realm
 - Similar to playback to SDI, HDMI
 - To use SRT to bridge NMOS domains
 - Senders and Receivers are NMOS controlled
 - Contribution links between production sites

Devices with SRT transport

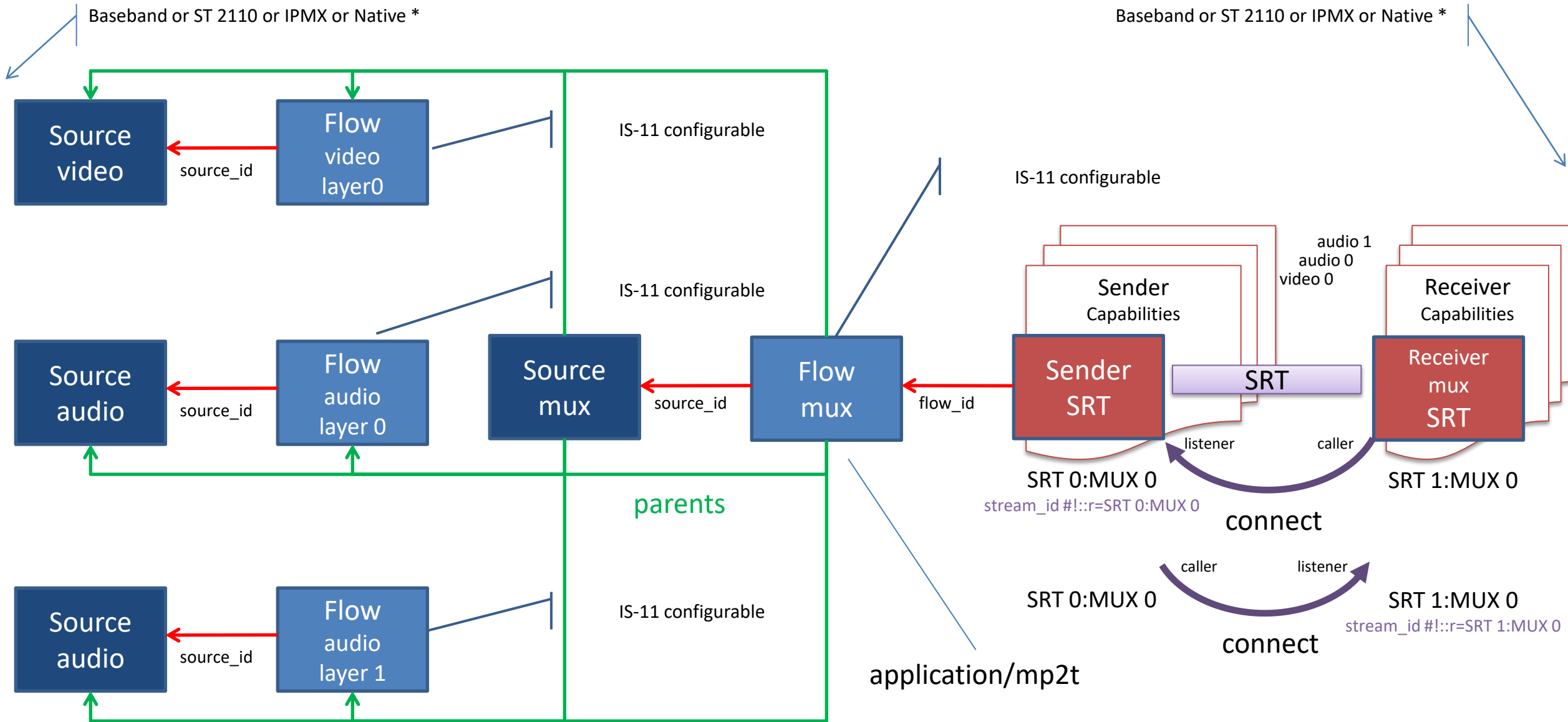


NMOS / SRT Ecosystems



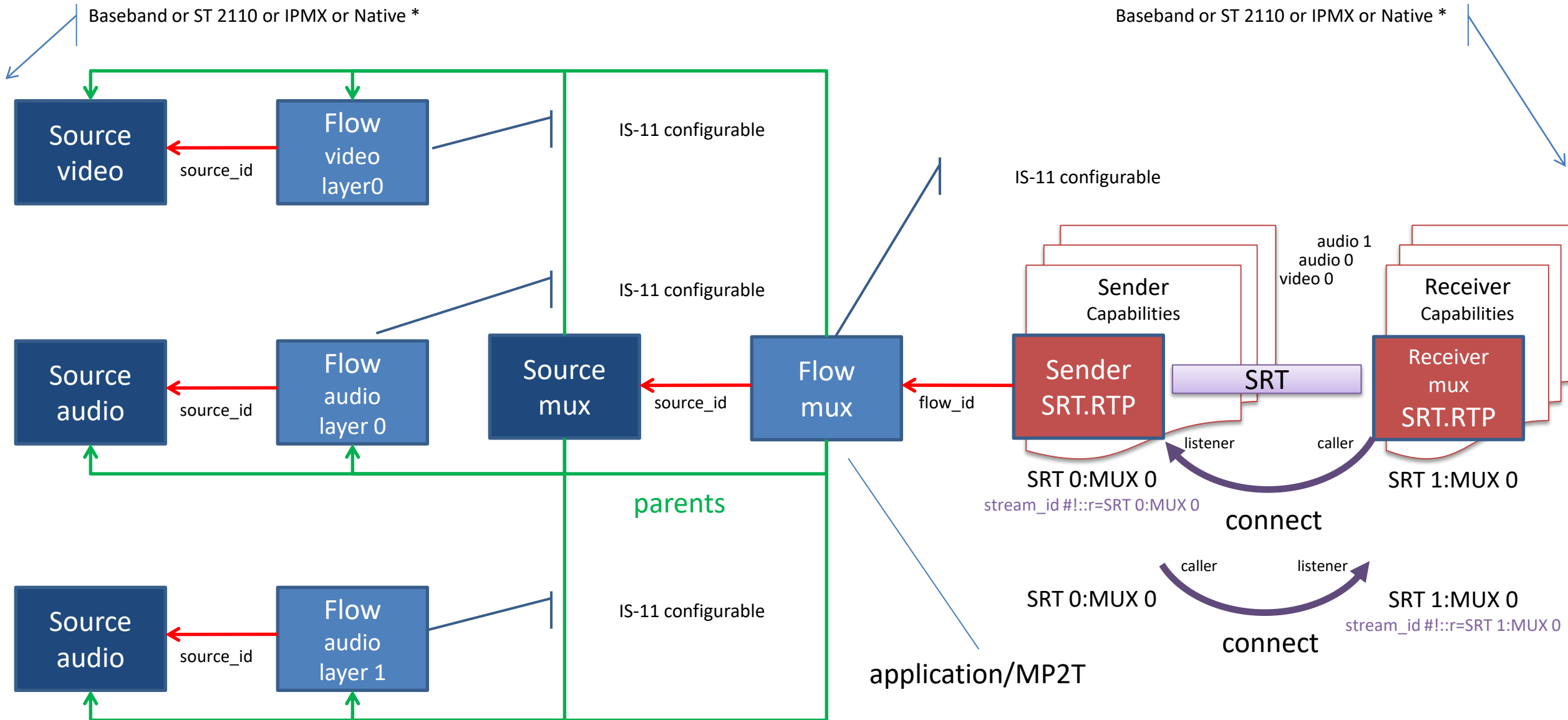
Multiplexed streams over SRT

SDP transport file
from the Sender



Multiplexed RTP streams over SRT

SDP transport file
from the Sender



Independent RTP streams over SRT

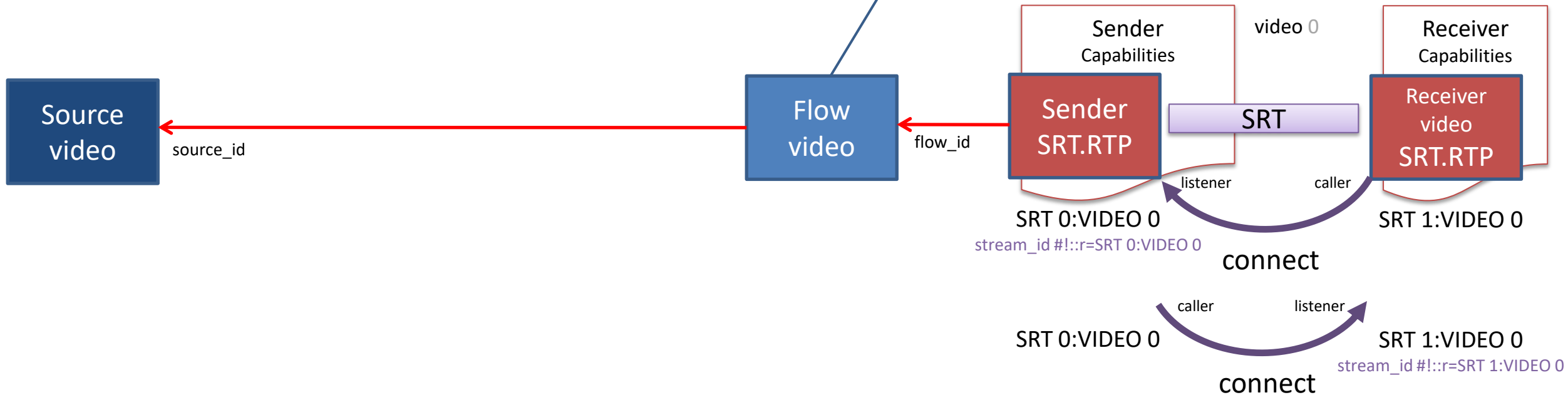
SDP transport file
from the Sender

Baseband or ST 2110 or IPMX or Native *

Baseband or ST 2110 or IPMX or Native *

Could be audio, video and data RTP streams

IS-11 configurable



SRT Use-Cases

Sender	Receiver	Connection
NMOS	NMOS	<p>Sender is a listener:</p> <p>The NMOS Controller discovers Senders and Receivers through the Registry using the IS-04 APIs. It configures the Sender using IS-11 active constraints and activates it as a listener using IS-05. The Controller then connects the Receiver to the Sender using the SDP transport file using IS-05. If the source_id feature is used by the Sender, the Receiver's source_id transport parameter is set at activation. The source_id value corresponds to the groupid of the SRT Sender.</p>
NMOS	Native	<p>Sender is a caller:</p> <p>The NMOS Controller discovers Senders through the Registry using the IS-04 APIs, and discovers Native SRT Receivers using an unspecified mechanism. It configures the Sender using IS-11 active constraints and activates it, as a caller to the Native Receiver, using IS-05. Native SRT Receivers listen for incoming connections from the Senders. If the source_id feature is used by the Native SRT Receiver, the Sender's source_id transport parameter is set at activation. The source_id value is defined by the Native SRT Receiver.</p>
Native	NMOS	<p>Sender is a listener:</p> <p>The NMOS Controller discovers Native SRT Senders using an unspecified mechanism, and discovers Receivers through the Registry using the IS-04 APIs. It connects the Receiver to the Native SRT Sender using the IS-05 transport parameters or an SDP transport file. If the source_id feature is used by the Native SRT Sender, the Receiver's source_id transport parameter is set at activation.</p>
Native	Native	<p>The NMOS Controller is not involved. Native SRT Senders and Receivers operate entirely within the SRT ecosystem, using their own discovery and control mechanisms.</p>

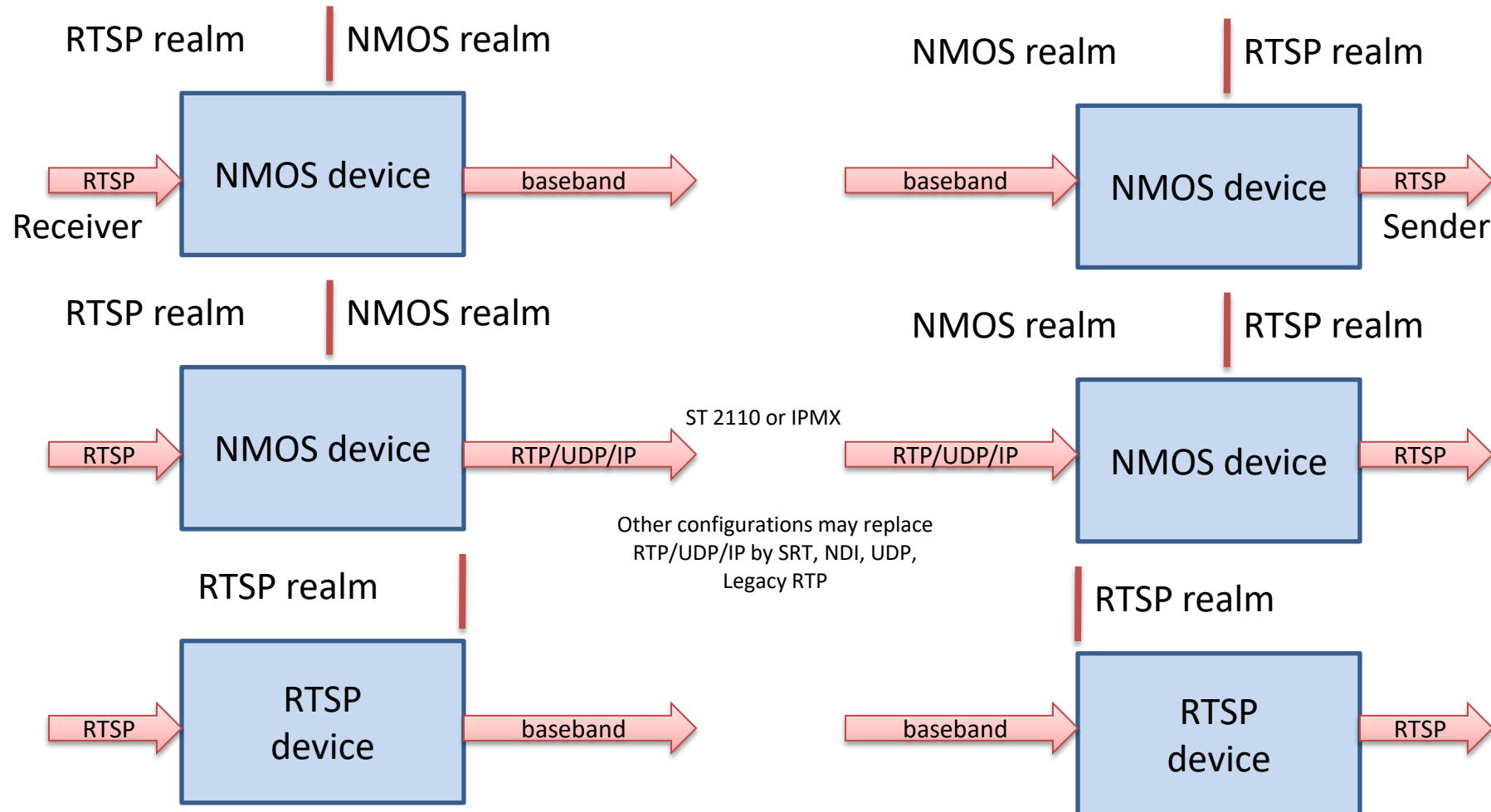
Privacy Encryption

- SRT supports encryption
 - When both the SRT Sender and Receiver are NMOS compliant devices, the SRT transport adaptation of the IPMX Privacy Encryption Protocol (PEP) is used to derive the SRT passphrase that controls encryption.
 - When an NMOS compliant SRT Sender transmits to a Native SRT Receiver, encryption is controlled by the NMOS Sender. In this case, a vendor specific mechanism must be provided to securely obtain the encryption key (passphrase) from the Node. The administrator must then provide the passphrase to the Native SRT Receiver. This Pre-Share Key (PSK) must be used only in scenarios involving Native SRT peers.
 - When a native SRT Sender transmits to an NMOS compliant Receiver, encryption is controlled by the NMOS Controller. A vendor specific mechanism must be provided to securely obtain the encryption key (passphrase) from the Controller. The administrator must then provide the passphrase to the Native SRT Sender. In this case, the Controller has knowledge of the PSK, which again must be used only in scenarios involving Native SRT peers.
 - A device should not allow forwarding a stream received with privacy encryption over an unencrypted SRT transport unless expressly authorized by the administrator.

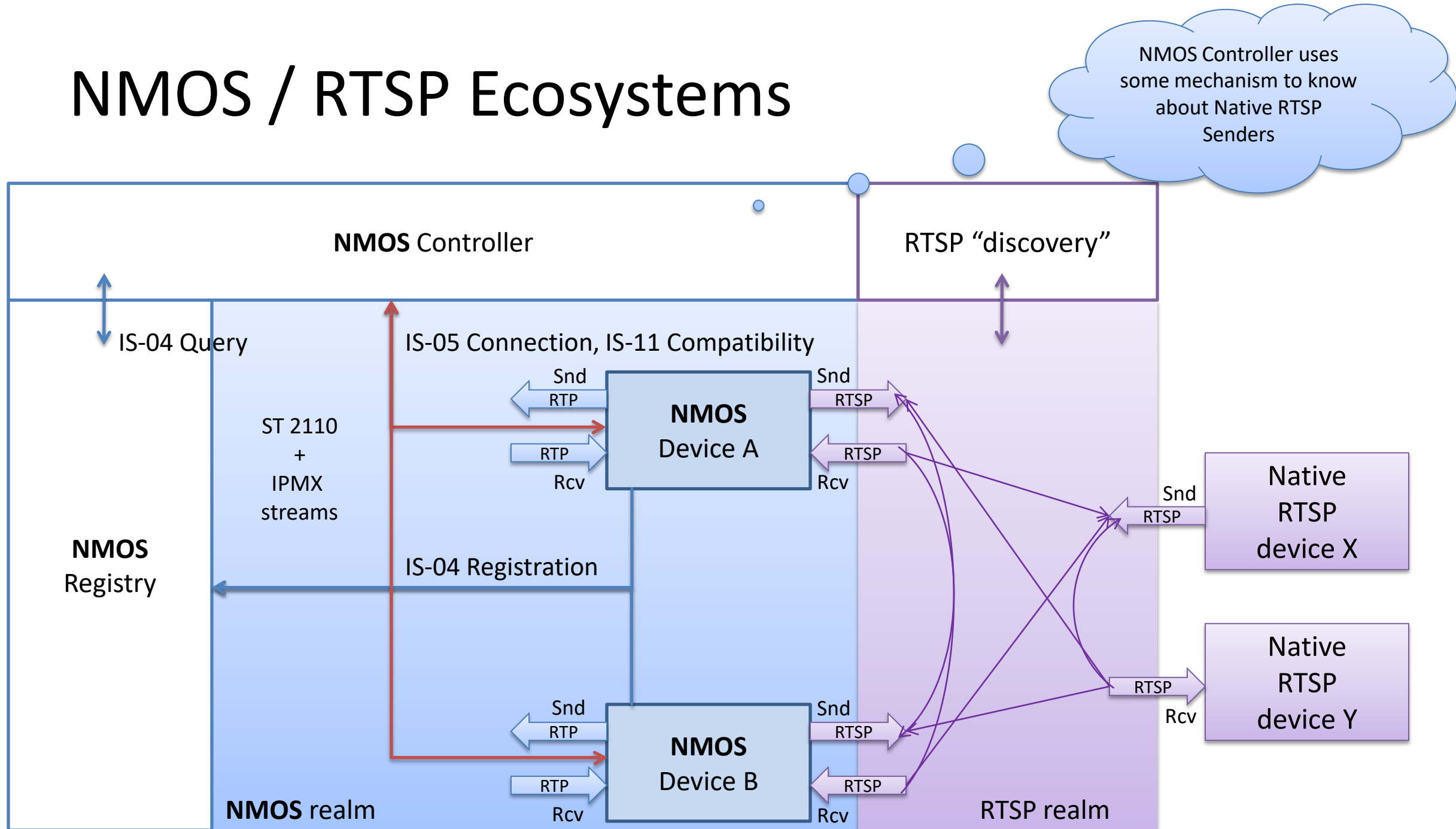
RTSP as transport from/to NMOS realm

- The objective is **not**
 - To use RTSP as a generic NMOS transport
 - What goes to the RTSP realm becomes out-of-scope for NMOS
 - What comes from the RTSP realm origins out-of-scope for NMOS
- The objective is
 - To use RTSP to capture from the NDI realm
 - Similar to capture from SDI, HDMI
 - To use RTSP to playback to the NDI realm
 - Similar to playback to SDI, HDMI
 - To use RTSP to enhance stream management
 - mult-unicast, aggregated sub-streams, dynamic transports

Devices with RTSP transport

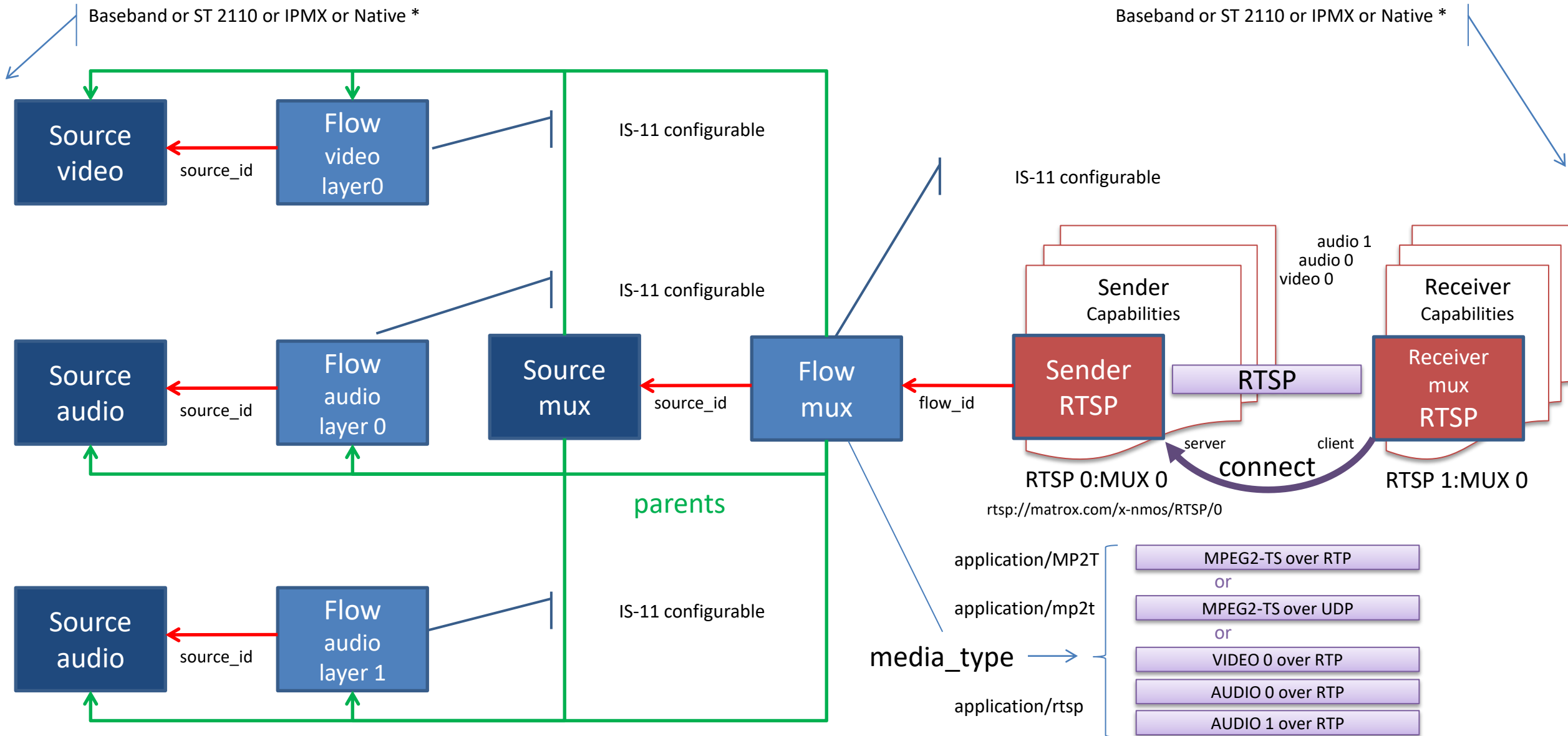


NMOS / RTSP Ecosystems



Multiplexed streams over RTSP

SDP transport file
from the Sender



RTSP Use-Cases

Sender	Receiver	Connection
NMOS	NMOS	The NMOS Controller discovers Senders and Receivers through the Registry using the IS-04 APIs. It configures the Sender using IS-11 active constraints and activates it using IS-05. The Controller then connects the Receiver to the Sender using the SDP transport file.
NMOS	Native	The NMOS Controller discovers Senders through the Registry using the IS-04 APIs, and discovers Native RTSP Receivers using an unspecified mechanism. It configures the Sender using IS-11 active constraints and activates it using IS-05. The Native Receiver connects to the Sender using the RTSP control URL from the Sender's SDP transport file.
Native	NMOS	The NMOS Controller discovers Native RTSP Senders using an unspecified mechanism, and discovers Receivers through the Registry using the IS-04 APIs. It connects a Receiver to the Native RTSP Sender by providing an SDP transport file that describes the Sender's control URL, or by providing the Sender's control URL through the optional ext_rtsp_session_control transport parameter of the Receiver.
Native	Native	The NMOS Controller is not involved. Native RTSP Senders and Receivers operate entirely within the SRT ecosystem, using their own discovery and control mechanisms.

Privacy Encryption

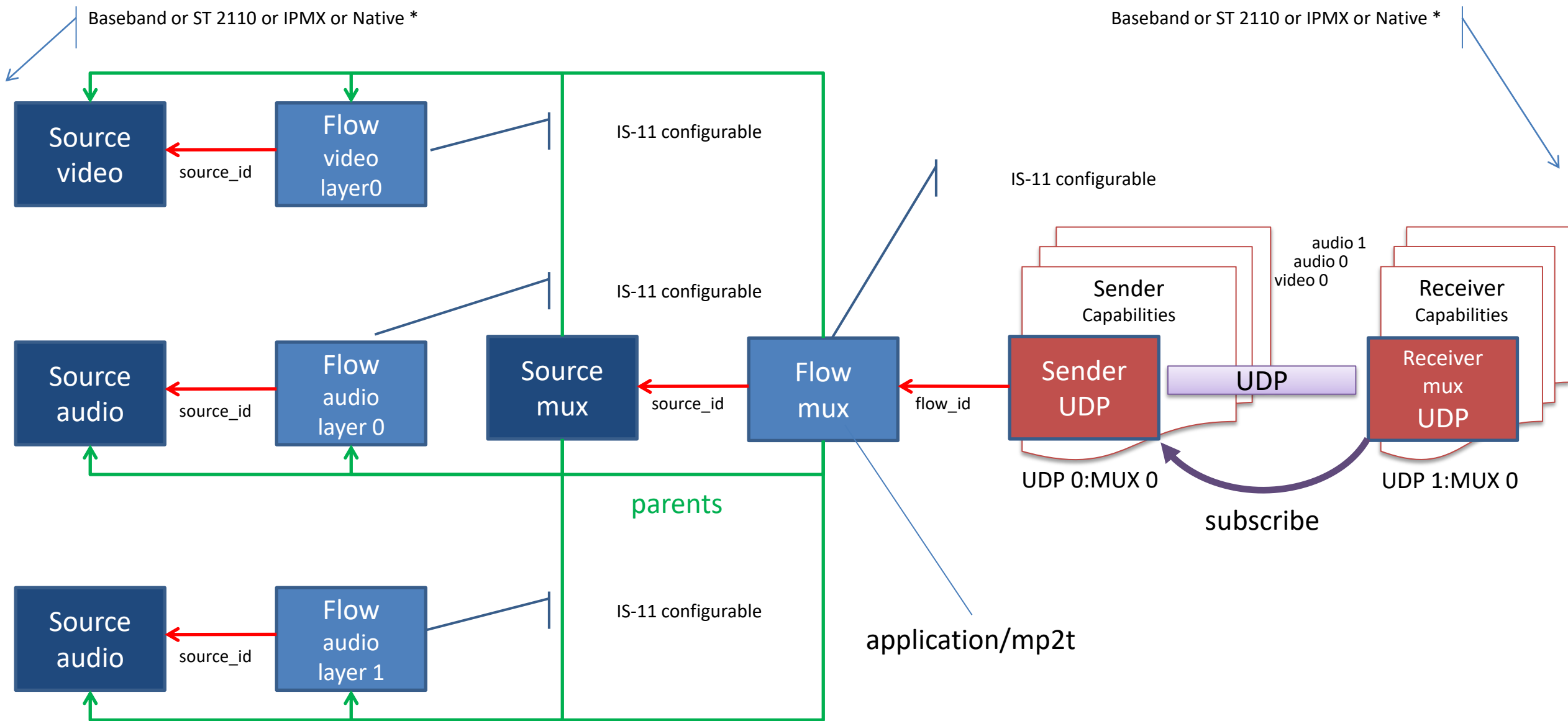
- RTSP supports encryption
 - When both the RTSP Sender and Receiver are NMOS compliant devices, the RTSP transport adaptation of the IPMX Privacy Encryption Protocol (PEP) is used to encrypt the independent RTP streams, the MPEG2-TS over RTP stream, or the MPEG2-TS over UDP stream.
 - When either peer is not NMOS compliant, it is assumed that there is no encryption, even if RTSP supports SRTP encryption.
 - A device should not allow forwarding a stream received with privacy encryption over an unencrypted RTSP transport unless expressly authorized by the administrator.

Additional transports

- MPEG2-TS over UDP
 - From RTSP when mux media_type is “application/mp2t”
 - From RTP when mux media_type is “application/mp2t”
- MPEG2-TS over Legacy RTP
 - From RTSP when mux media_type is “application/MP2T”
 - From RTP when mux media_type is “MP2T”
- Audio, Video and Data streams over Legacy RTP
 - When a stream is not compliant with ST 2110, nor with IPMX

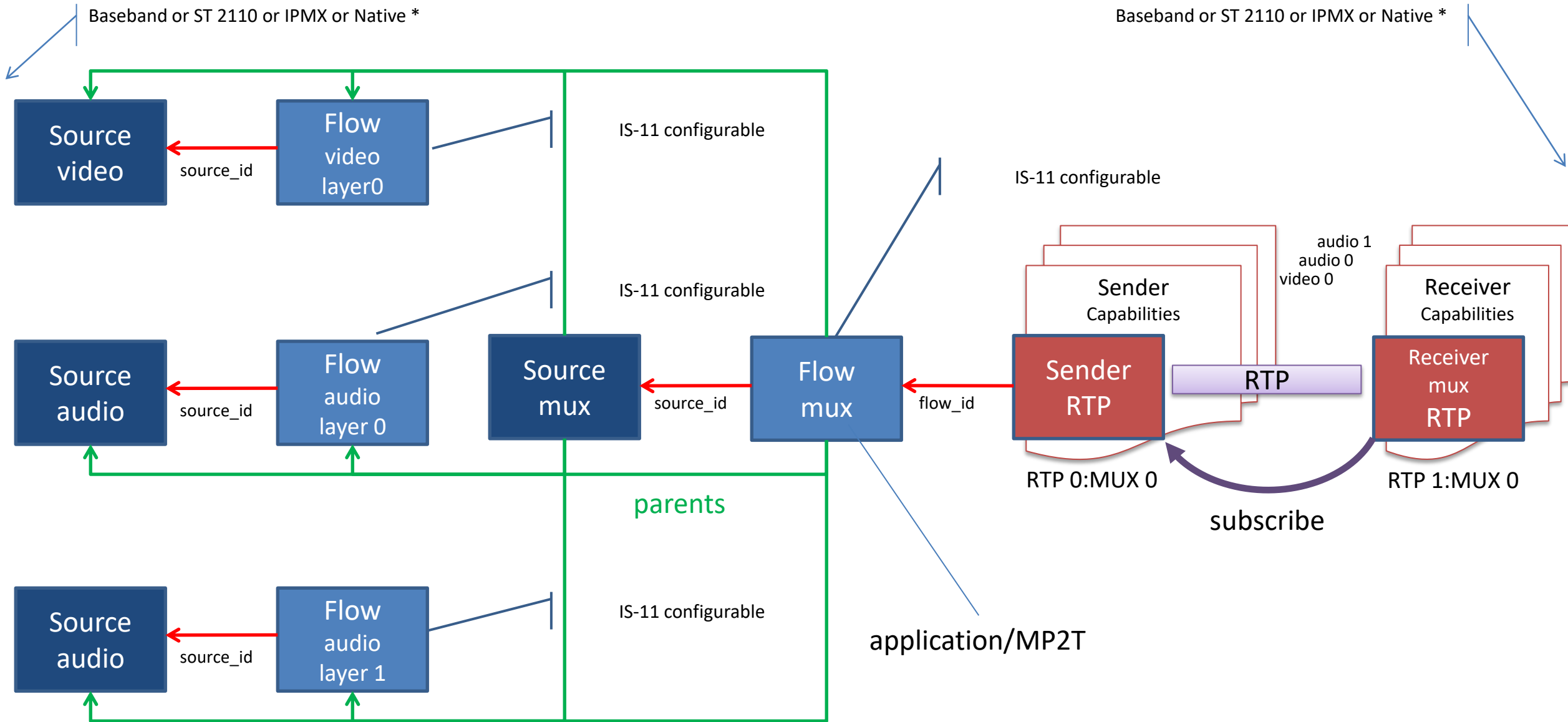
Multiplexed streams over UDP

SDP transport file
from the Sender



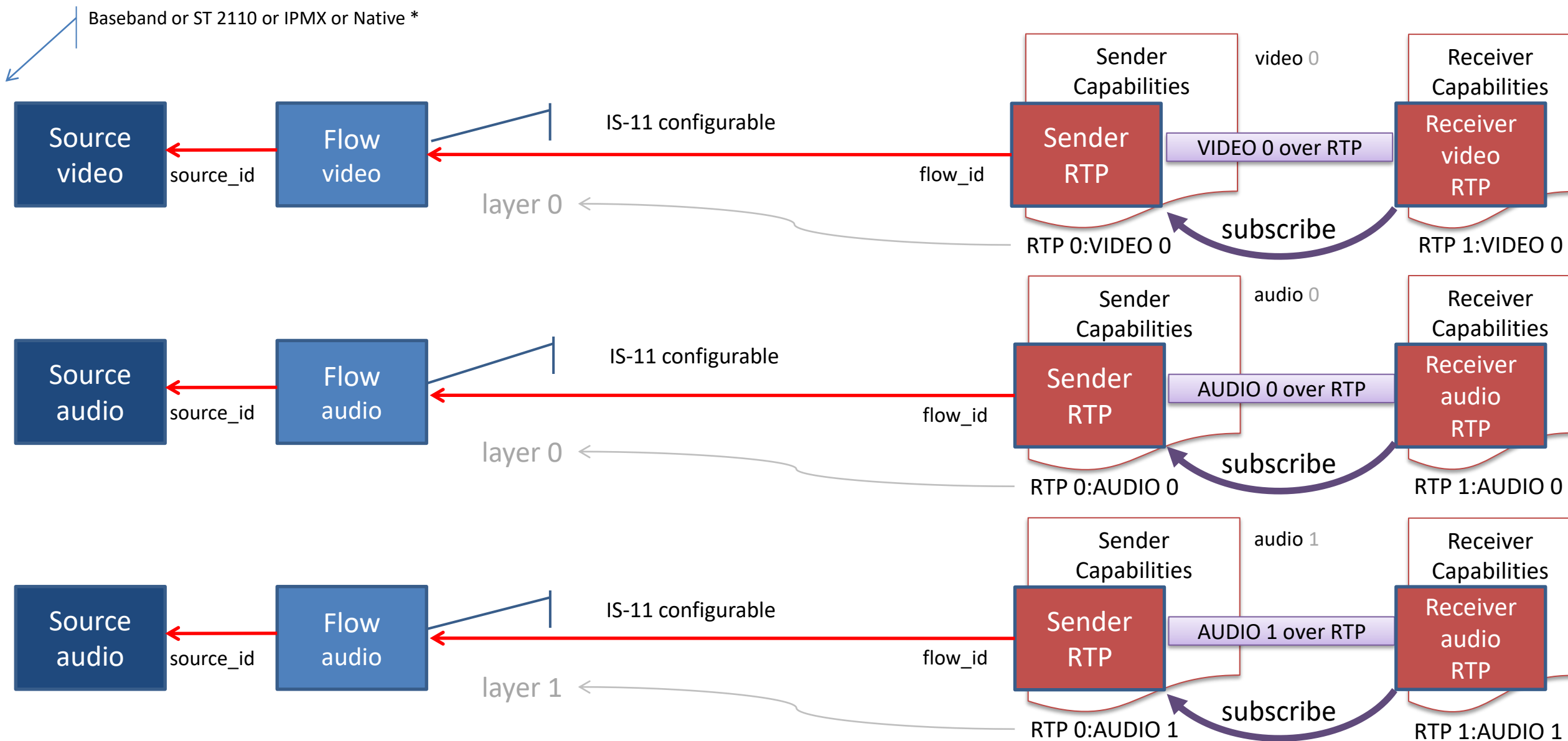
Multiplexed streams over Legacy RTP

SDP transport file
from the Sender



Independent Streams over Legacy RTP

SDP transport file
from the Senders



- This concludes our overview of NMOS with NDI, SRT, and RTSP transports, key features of Matrox NMOS Advanced Streaming Architecture.
- If you have any questions, feel free to reach out at abouchar@matrox.com.
- Thank you for attending.

Copyright (c) 2025, Matrox Graphics Inc.

This work, including the associated documentation, is licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0).

You are free to share and adapt this material for any purpose, provided that you give appropriate credit to Matrox Graphics Inc.

To view a copy of this license, visit:

<https://creativecommons.org/licenses/by/4.0/>