



# NMOS Advanced Streaming Architecture Enhanced Capabilities and more ...

Alain Bouchard, ing



# Public GitHub Repository

- <https://github.com/alabou/NMOS-MatroxOnly>
  - README.md
  - ReceiverCapabilities.md
  - SenderCapabilities.md
  - NMOS With IS-11.md

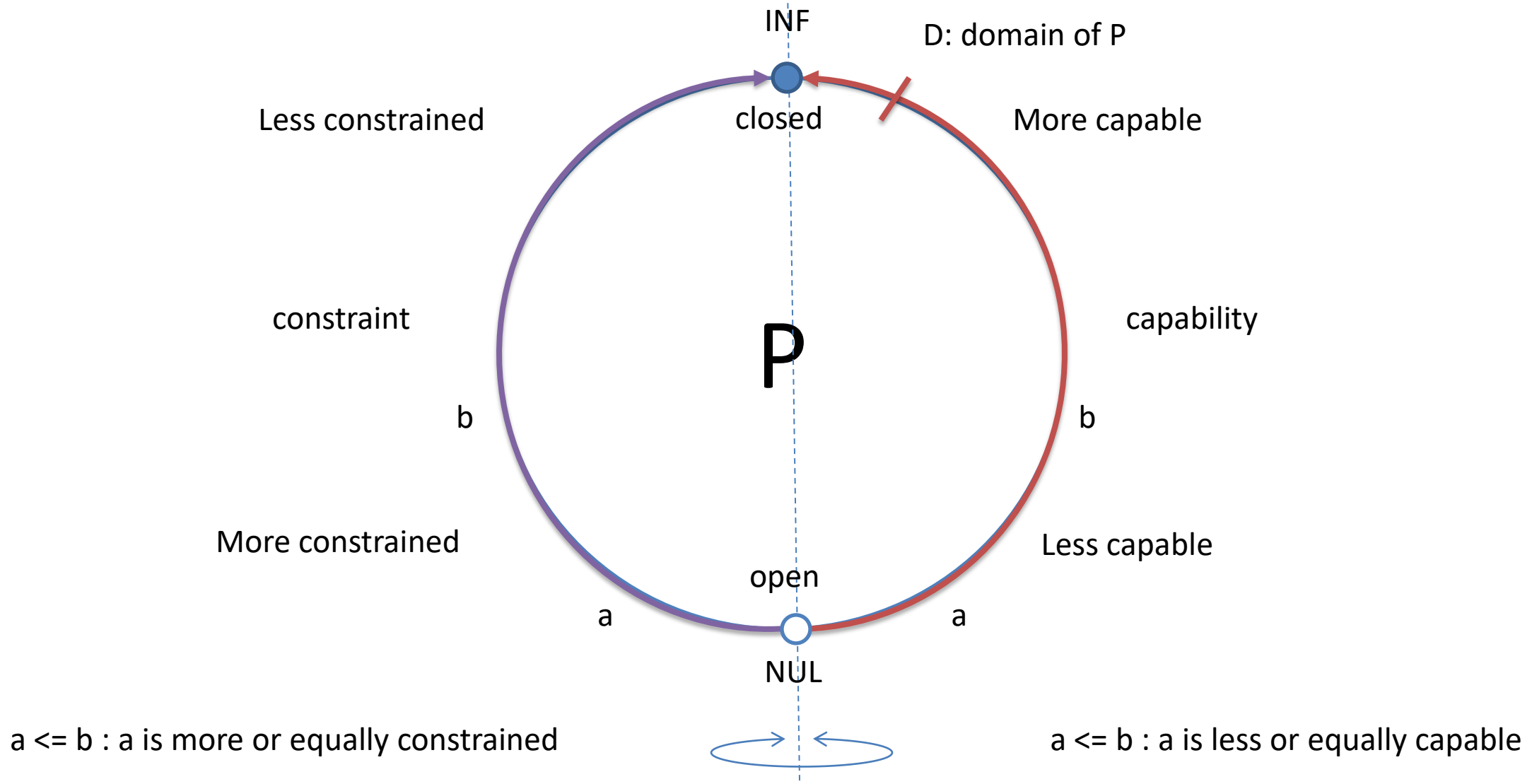
# Capabilities

- Describe what is allowed, available or possible.
- Describe the possible values of a parameter.
  - Capability Space (CapS) of parameter P
- A capability essentially outlines the permissible values or operational space of a parameter, representing what is possible or supported.

# Constraints

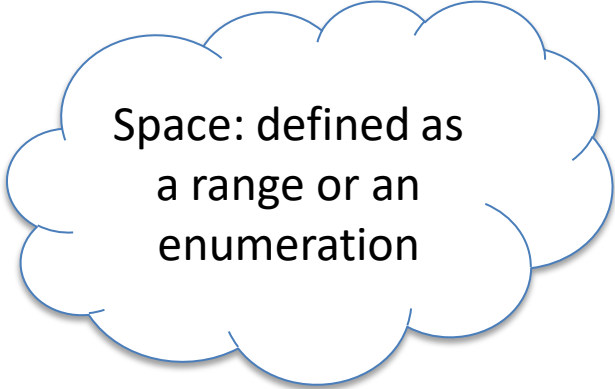
- Constrain what is allowed, available or possible.
- Constrain the possible values of a parameter.
  - Constraint Space (ConS) of parameter P
- A constraint essentially restricts the permissible values or operational space of a parameter, representing what is possible or supported. Its key purpose is to **reduce or limit the flexibility** provided by capabilities.

# Constraints / Capabilities



# Parameter P

- Capability / Constraint Space of P (CapS / ConS)
  - Types: Bool, Integer, Float, Rational, String or Untyped
  - Range: Finite or Infinite
- Domain of P (D)
  - Real and Finite
  - Associated Capability/Constraint Space
    - May extend beyond the domain of P, spanning an infinite range (INF).
- Degrees of freedom of a set of P (DoF)
  - from 1 to infinite

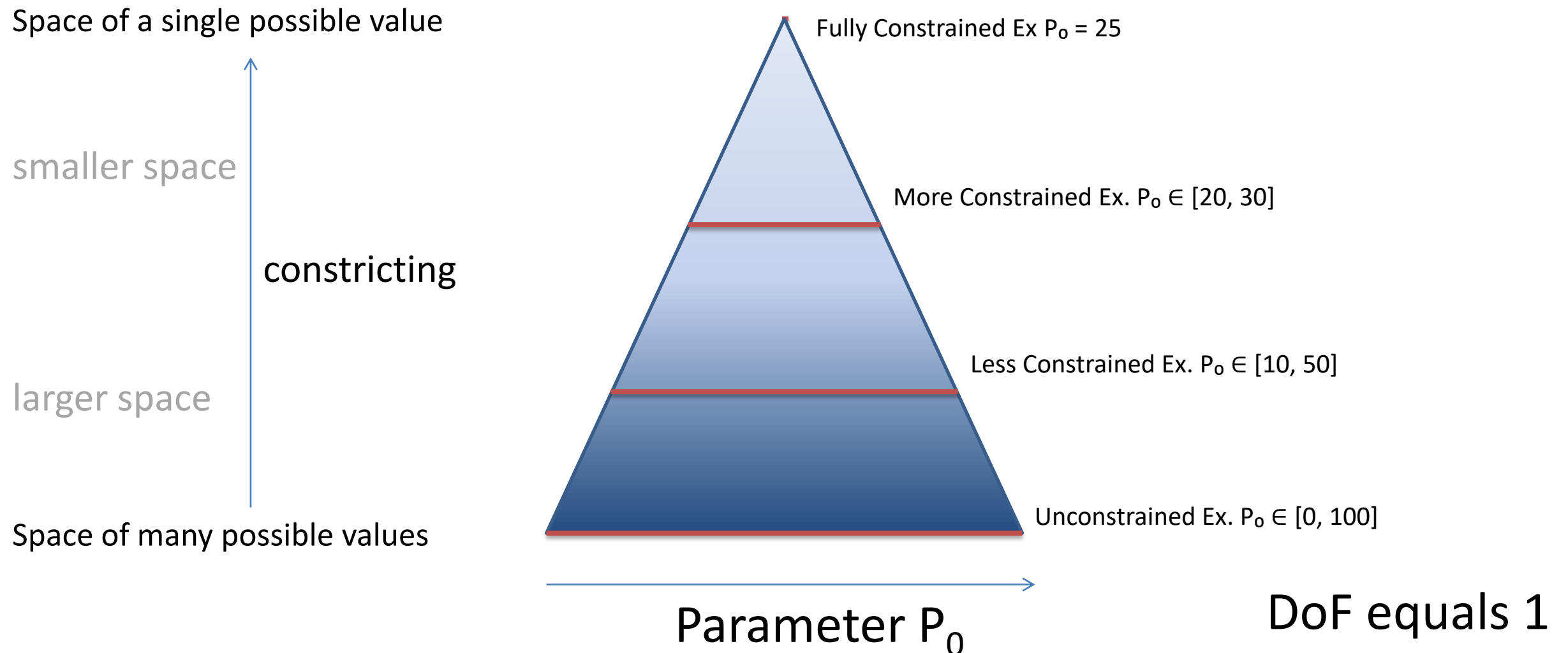


Space: defined as  
a range or an  
enumeration

# Constraint/Capability Duality

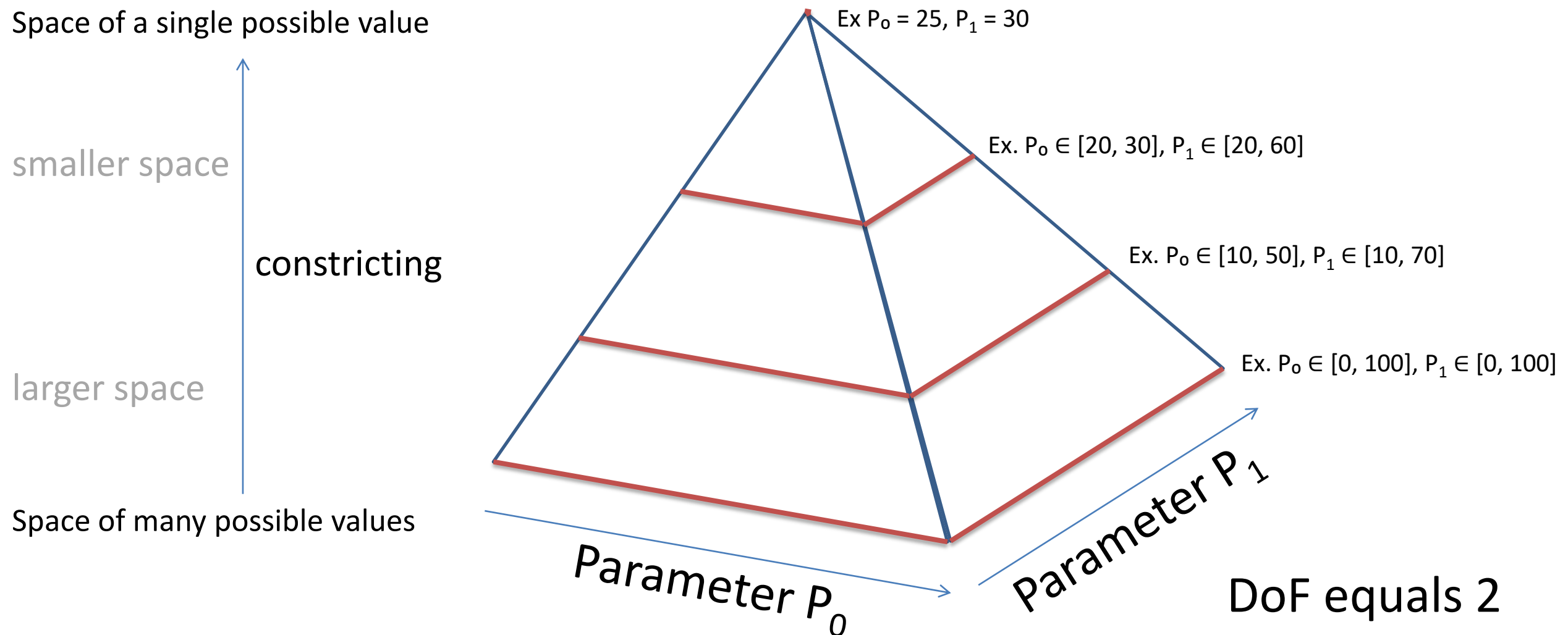
- A range describes either a Capability or a Constraint
  - Represents the Capability / Constraint Space of Parameter P.
- A range identity, Caps or Cons, is fixed at processing time
  - Infinite Range (INF):
    - As a **Capability**: Indicates the parameter supports any value.
    - As a **Constraint**: Implies no restriction is imposed on the parameter.

# Parameter Space Pyramid (2D)





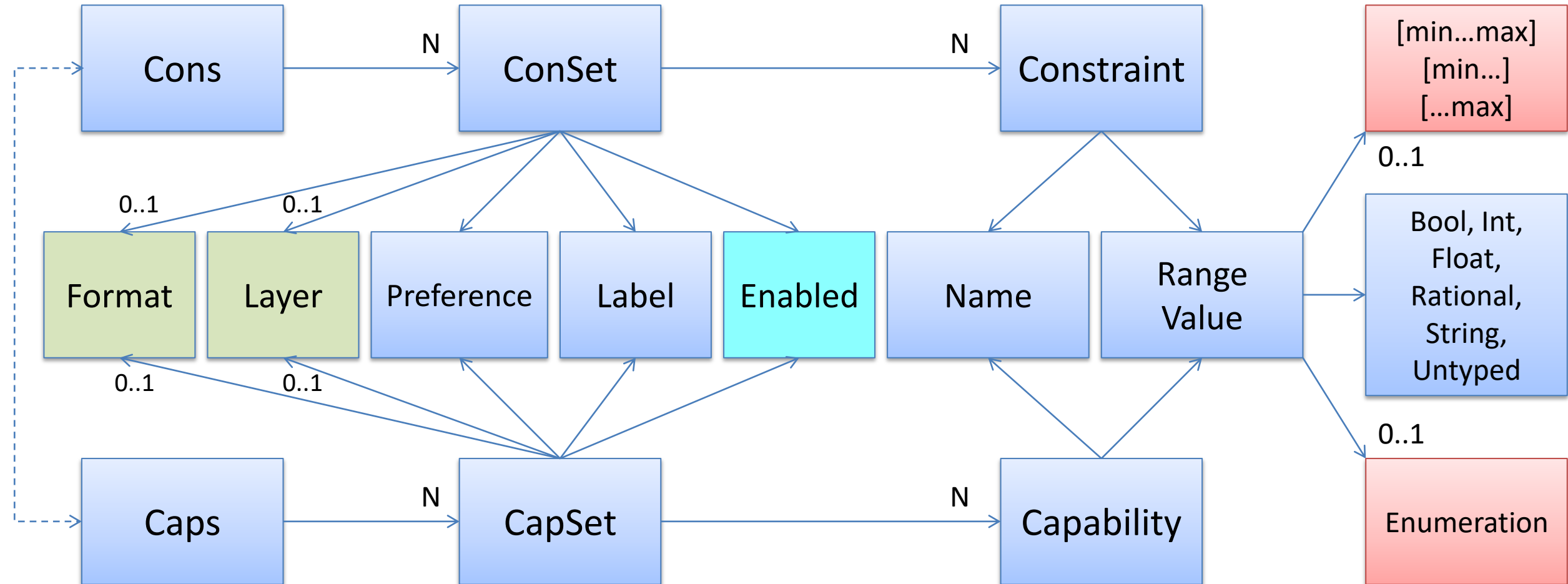
# Parameter Space Pyramid (3D)



# Parameter Space Pyramid (nD)

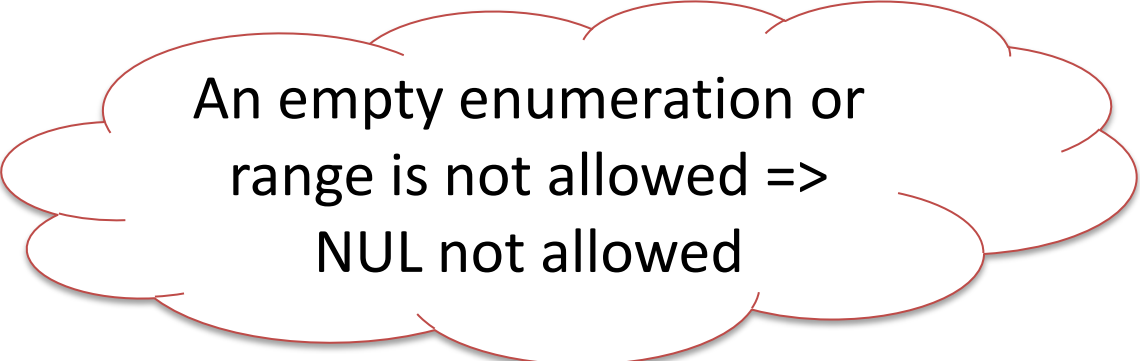
- HyperPyramid of n dimensions
  - Representation for more than 2 parameters

# Enhanced Capabilities

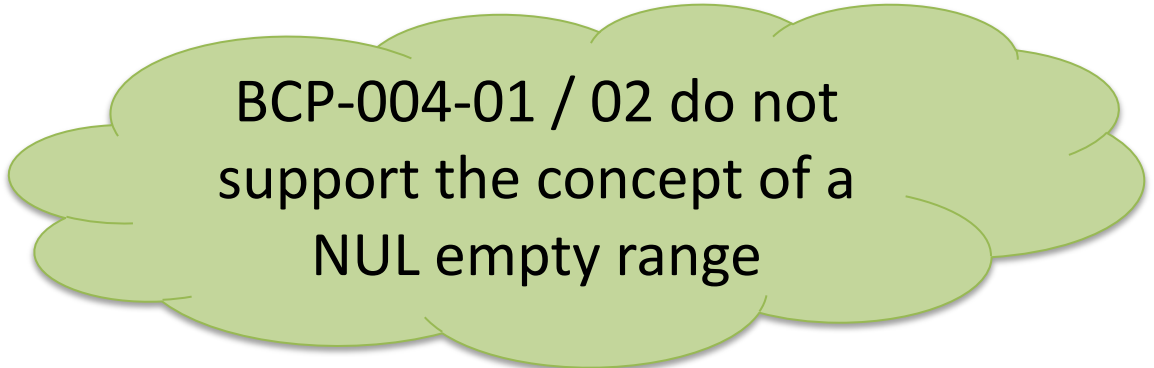


# Infinite Range

- If there is no range and no enumeration  
INF => Infinite range



An empty enumeration or  
range is not allowed =>  
NUL not allowed



BCP-004-01 / 02 do not  
support the concept of a  
NUL empty range

# Flow/Stream Parameters

- Flow parameters => NMOS Flow/Source attributes
- Stream parameters => SDP transport file parameters

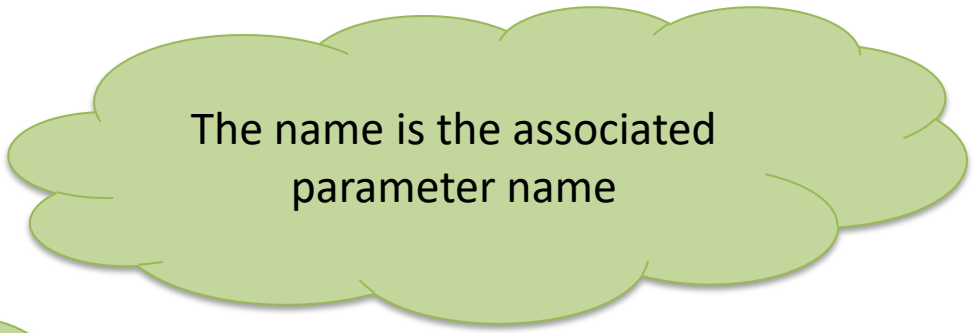
( Constraints  $\leftrightarrow$  Capabilities )  $\rightarrow$  Parameters



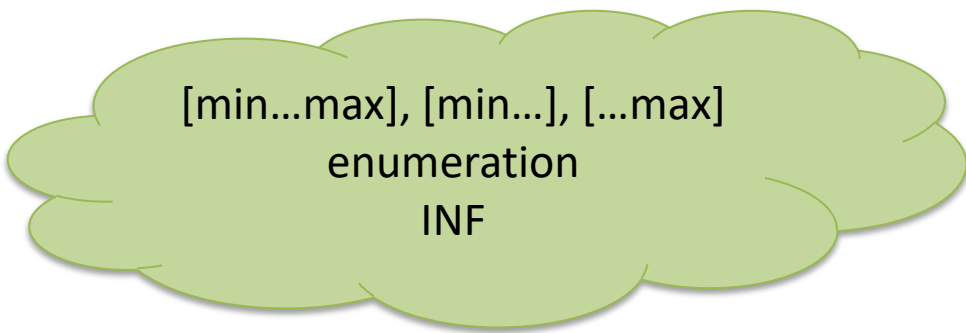
all use the parameter name

# Base Concepts

- Capability / Constraint
  - name: str
  - value: RangeValue



The name is the associated  
parameter name



[min...max], [min...], [...max]  
enumeration  
INF

Examples:

- “media\_type” allows [“video/raw”, “video/jxsv”, “video/h264”, “video/h265”]
- “media\_type” allows INF

# Base Concepts

- CapSet / ConSet

- caps: Dict[str, Capability / Constraint] = dict()
- enabled: bool = False
- preference: int = 0
- label: str = ""
- format: Optional[str] = None
- layer: Optional[int] = None
- layer\_enabled: Optional[bool] = None
- layer\_compatibility\_groups: Optional[List[int]] = None

} Parameter Space

} sub-Flows  
sub-Streams

# Base Concepts

- Caps

- capsets: List[CapSet] = list()

Caps  $\leftrightarrow$  Cons **conversion**

Change the interpretation to be given to unspecified parameters. If “unspecified” is to mean “may take any value” then Caps. If “unspecified” is to mean “don’t care” then Cons.

- Cons

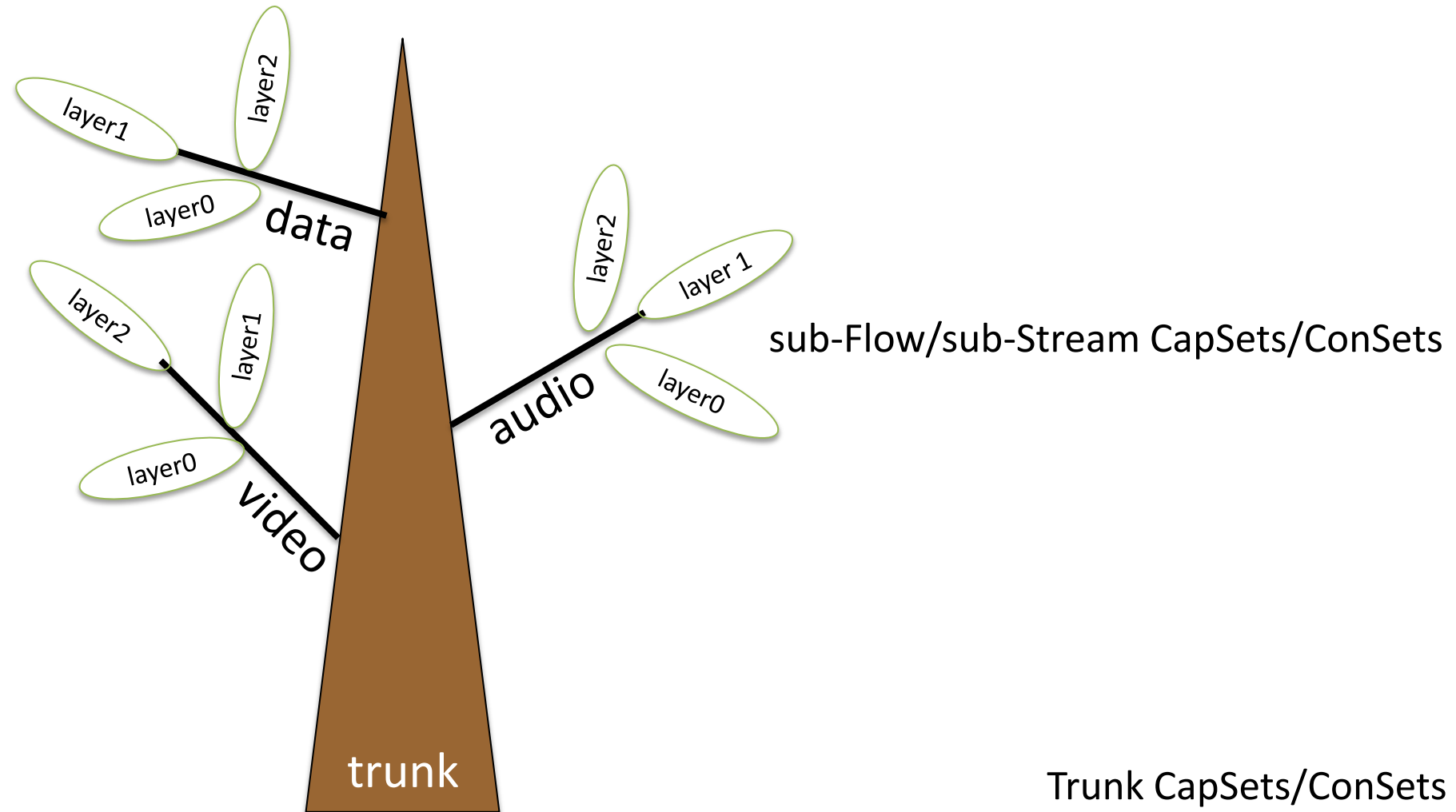
- consets: List[ConSet] = list()

Alternate Parameter Spaces  
Multi-Part Parameter Spaces

- Sender and Receiver Capabilities are Caps (**can do**)
- Sender Active Constraints are Cons (**must do**)



# Enhanced Capabilities => Multiple Parts



# Flows / Streams => trunk

- Flows/Streams CapSets/ConSets
    - enabled is True
    - No `layer_enabled`, `format`, `layer` attributes
    - Optional `layer_compatibility_groups` attribute (mux)
- } Identifies a CapSet/ConSet as being a sub-Flow/sub-Stream.

# sub-Flows / sub-Streams => leaves

- sub-Flows/sub-Streams CapSets/ConSets

- enabled is False => use `layer_enabled` instead
- `layer_enabled`, `format`, `layer` attributes (required)
- `layer_compatibility_groups` attribute (optional)



Identifies a  
CapSet/ConSet  
as being a  
sub-Flow/sub-Stream.

A red arrow originates from the text block and points towards the first list item.

A Controller not supporting enhanced  
Capabilities will simply ignore this  
CapSet/ConSet as it is disabled.

# Caps Filtering

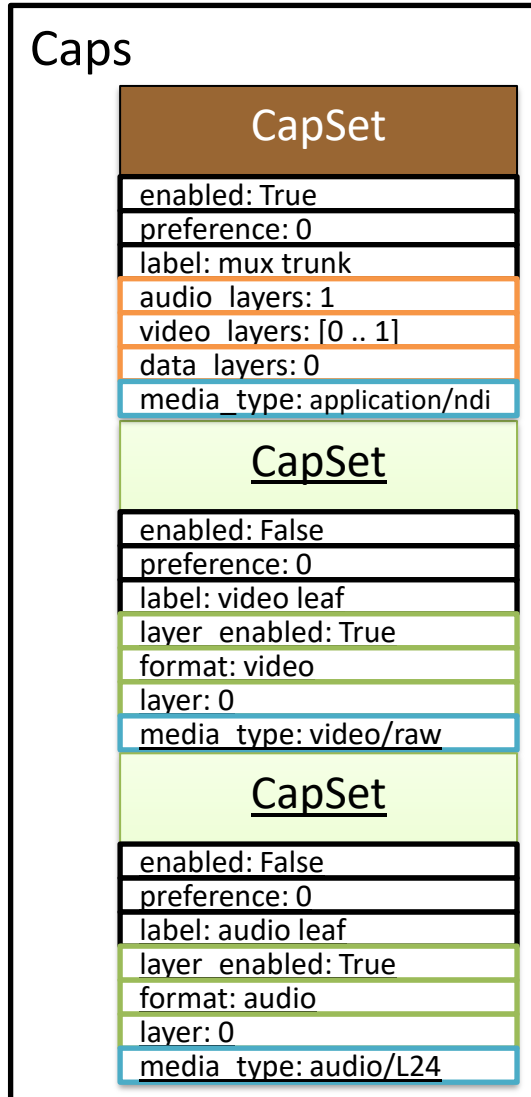
Caps	
CapSet	
enabled:	True
preference:	0
label:	mux trunk
audio layers:	[0 .. 1]
video layers:	1
data layers:	0
media_type:	application/ndi
<u>CapSet</u>	
enabled:	False
preference:	0
label:	video leaf
layer enabled:	True
format:	video
layer:	0
media_type:	video/raw
<u>CapSet</u>	
enabled:	False
preference:	0
label:	audio leaf
layer enabled:	True
format:	audio
layer:	0
media_type:	audio/L24

Extract Trunk

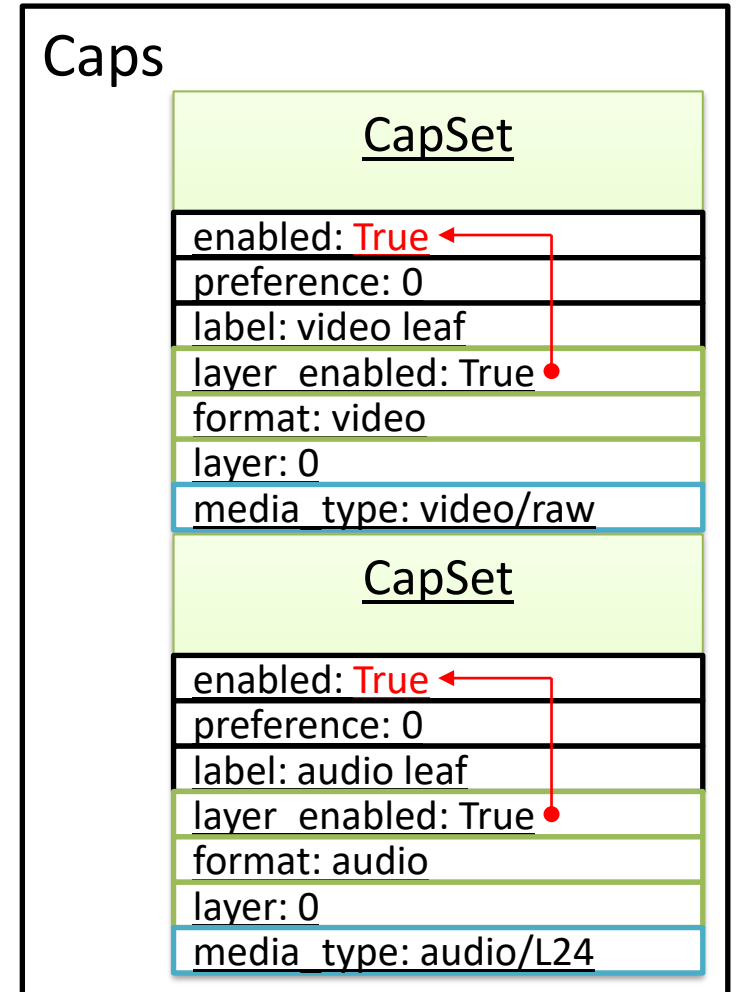
Caps	
CapSet	
enabled:	True
preference:	0
label:	mux trunk
audio layers:	[0 .. 1]
video layers:	1
data layers:	0
media_type:	application/ndi

Leaves CapSet being disabled are not considered by Controller not supporting enhanced capabilities. Leaves media\_type is also not in the media\_types array of a Receiver.

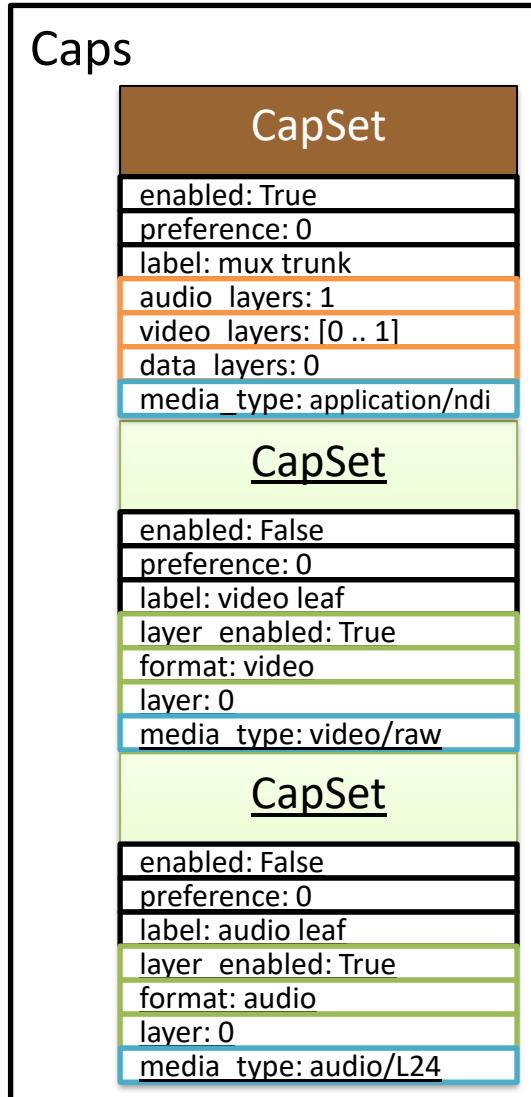
# Caps Filtering



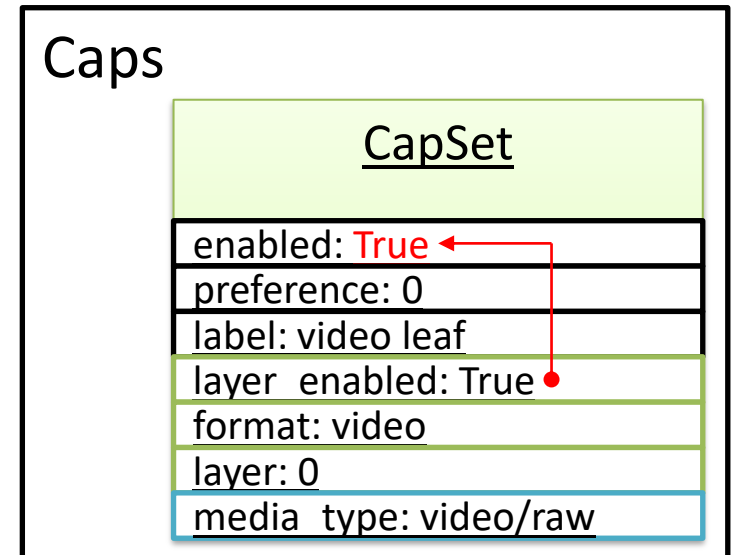
Extract Layers



# Caps Filtering

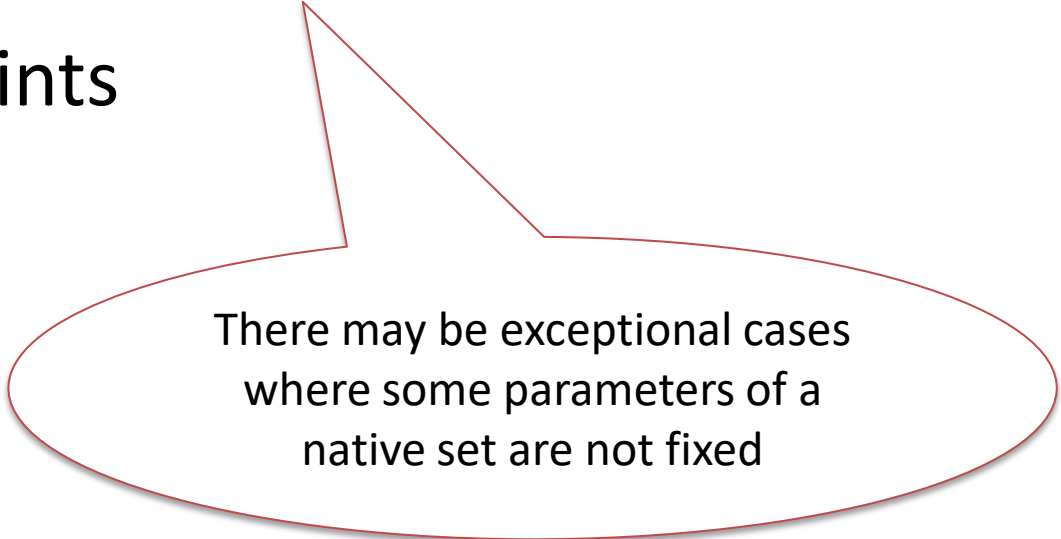


Extract Video Layer 0



# CapSet/ConSet preference

- preference of value 100 **should be special**
  - ^ tip of the parameter space pyramid
    - Represent **native** capabilities
      - One possible value per parameter (for 99% of the parameters)
    - Represent **preferred** constraints
      - First to be considered



There may be exceptional cases  
where some parameters of a  
native set are not fixed

# Stream Compatibility

- Controller
  - Without IS-11
    - Current state versus Receiver(s) Capabilities
      - ⇒ Fail connection if Sender not compliant
  - With IS-11
    - Sender Capabilities versus Receiver(s) Capabilities
      - ⇒ Generate active constraints to make the Sender compliant
      - ⇒ Fully CONFIGURE the Sender



# Stream Compatibility

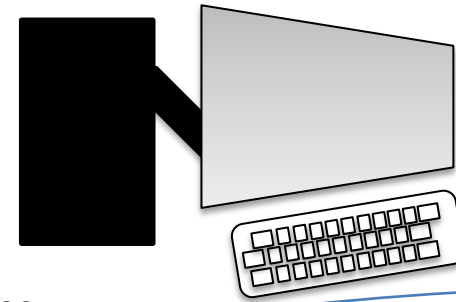
- Controller
  - With IS-11 (Hybrid)
    - Sender Capabilities versus Receiver(s) Capabilities
      - ⇒ Generate active constraints to make the Sender compliant
      - ⇒ Partially CONFIGURE the Sender
    - Current state versus Receiver(s) Capabilities
      - => For parameters that cannot be constrained

# Stream Compatibility

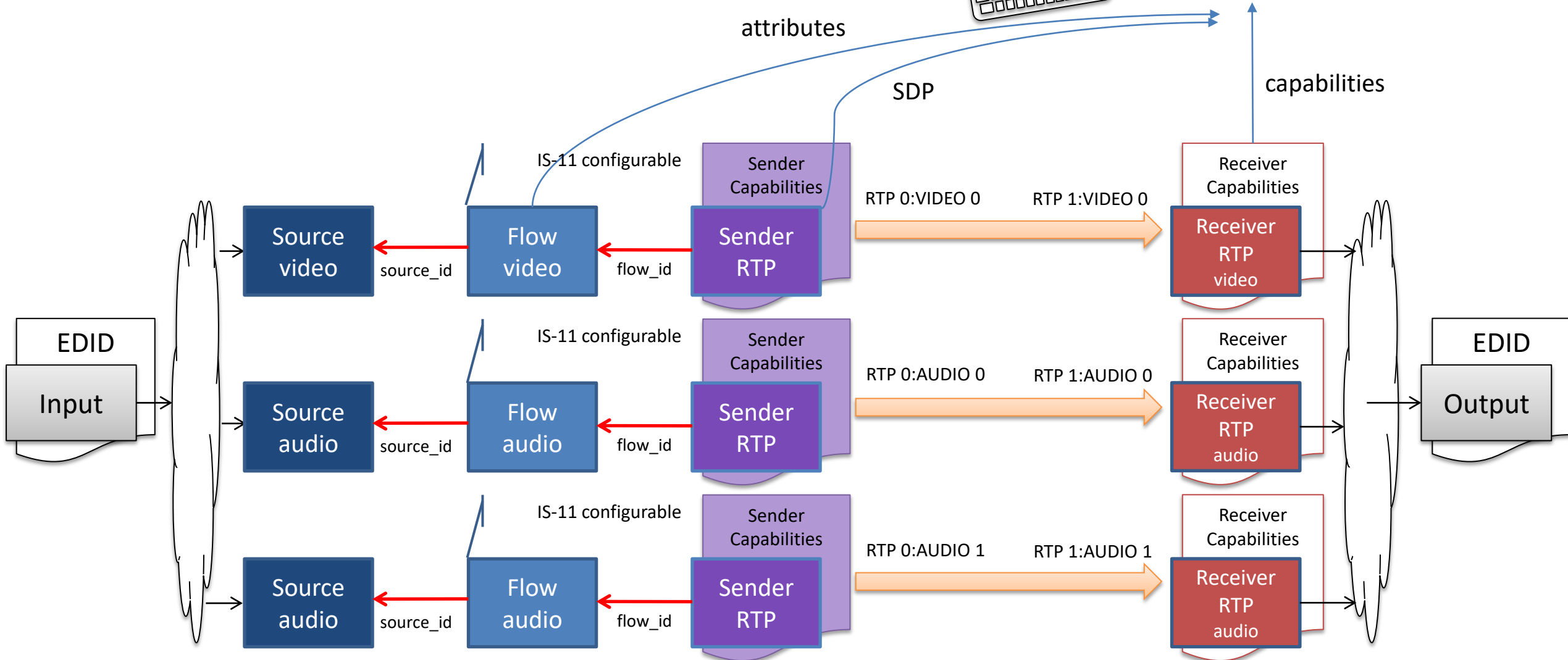
- Controller
  - Ensures that the actual Sender's Flow/Stream parameters are within the parameter space of the subscribed Receivers.
    - For each layer: ensures that the actual Sender's sub-Flow/sub-Stream parameters are within the parameter space of the subscribed Receivers.
  - Ensures that the Sender's constrained Flow/Stream parameters are within the parameter space of the subscribed Receivers.
    - For each layer: ensures that the Sender's constrained sub-Flow/sub-Stream parameters are within the parameter space of the subscribed Receivers.

# Stream Compatibility Assessment

Without IS-11 => current Flow attributes



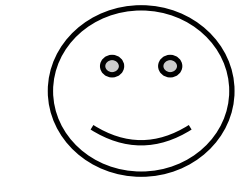
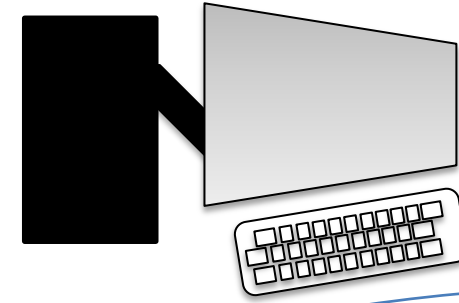
Controller



# Stream Compatibility Assessment

With IS-11 => active constraints

Parameters not supported as active constraints may require an assessment using the actual attributes

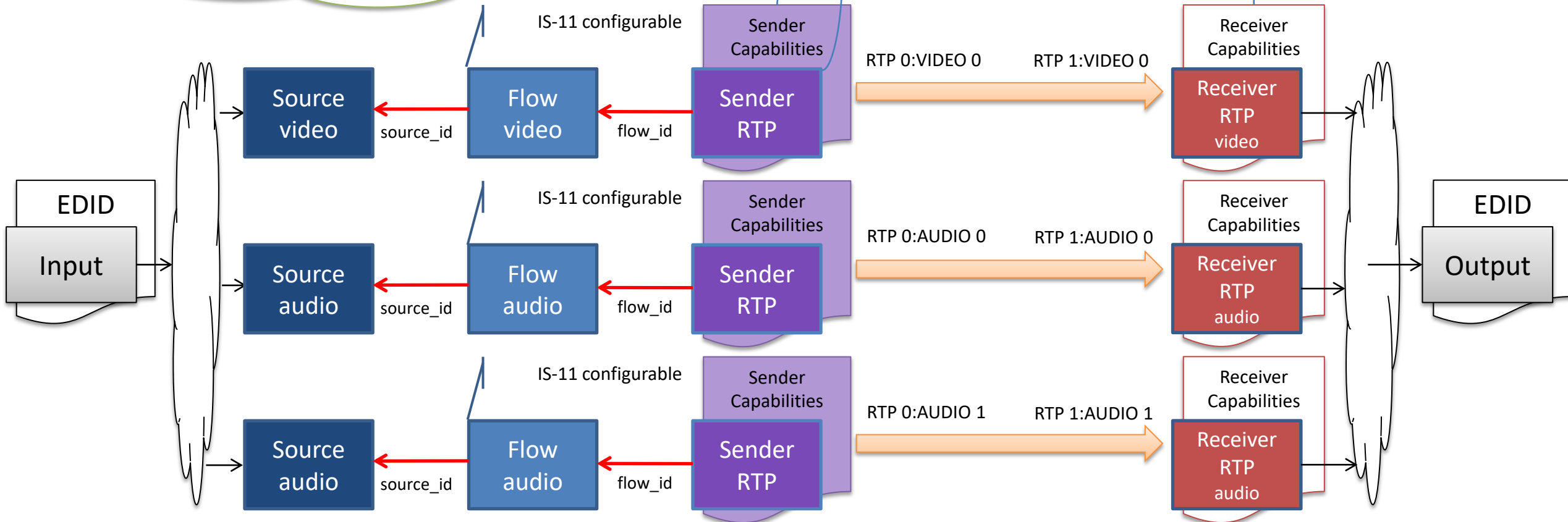


Controller

capabilities

supported constraints

capabilities



# Video Capabilities

video/raw, video/jxsv, video/H264, video/H265



## FORMAT (base)

- media\_type (sdp) (flow)
- grain\_rate (sdp) (flow)
- frame\_width (sdp) (flow)
- frame\_height (sdp) (flow)
- interlace\_mode (sdp) (flow)
- colorspace (sdp) (flow)
- transfer\_characteristic (sdp) (flow)
- color\_sampling (sdp) (flow)
- component\_depth (sdp) (flow)

## TRANSPORT

- st2110\_21\_sender\_type (sdp) (sender)
- clock\_ref\_type (sdp) (source)
- synchronous\_media (sdp) (source)
- info\_block (sdp) (sender)

## FORMAT (coded)

- bit\_rate (flow)
- constant\_bit\_rate (flow)
- profile (sdp) (flow)
- level (sdp) (flow)
- sublevel (sdp) (flow)

## TRANSPORT

- bit\_rate (sdp) (sender)
- packet\_transmission\_mode (sdp) (sender)
- parameter\_sets\_flow\_mode (sender)
- parameter\_sets\_transport\_mode (sdp) (sender)
- hkep (sdp) (sender)
- privacy (sdp) (sender)

# Audio Capabilities

audio/L16, audio/L20, audio/L24, audio/AM824, audio/mpeg4-generic, audio/MP4A-LATM



## FORMAT (base)

- media\_type (sdp) (flow)
- channel\_count (sdp) (source)
- sample\_rate (sdp) (flow)
- sample\_depth (sdp) (flow)

## TRANSPORT

- packet\_time (sdp)
- max\_packet\_time (sdp)
- st2110\_21\_sender\_type (sdp) (sender)
- channel\_order (sdp)
- clock\_ref\_type (sdp) (source)
- synchronous\_media (sdp) (source)
- info\_block (sdp) (sender)

## FORMAT (coded)

- bit\_rate (flow)
- constant\_bit\_rate (flow)
- profile (sdp) (flow)
- level (sdp) (flow)

## TRANSPORT

- bit\_rate (sdp) (sender)
- packet\_transmission\_mode (sdp) (sender)
- parameter\_sets\_flow\_mode (sender)
- parameter\_sets\_transport\_mode (sdp) (sender)
- hkep (sdp) (sender)
- privacy (sdp) (sender)

# Data Capabilities

video/smpte291, application/usb



## FORMAT (base)

- media\_type (sdp) (flow)
- event\_type (flow)

## TRANSPORT

- st2110\_21\_sender\_type (sdp) (sender)
- clock\_ref\_type (sdp) (source)
- synchronous\_media (sdp) (source)
- info\_block (sdp) (sender)

## TRANSPORT

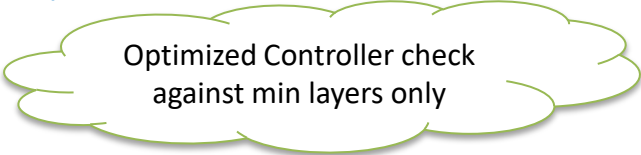
- privacy (sdp) (sender)

# Mux Capabilities

application/MP2T, application/mp2t, application/ndi, application/rtsp, application/AM824

## FORMAT (mux)

- media\_type (sdp) (flow)
- video\_layers (flow)
- audio\_layers (flow)
- data\_layers (flow)



Optimized Controller check  
against min layers only

## FORMAT (video sub-stream base)

- media\_type (parent flow)
- grain\_rate (parent flow)
- frame\_width (parent flow)
- frame\_height (parent flow)
- interlace\_mode (parent flow)
- colorspace (parent flow)
- transfer\_characteristic (parent flow)
- color\_sampling (parent flow)
- component\_depth (parent flow)

## FORMAT (data sub-stream base)

- media\_type (parent flow)
- event\_type (parent flow)

## FORMAT (audio sub-stream base)

- media\_type (parent flow)
- channel\_count (parent source)
- sample\_rate (parent flow)
- sample\_depth (parent flow)

## FORMAT (audio/video sub-stream coded)

- bit\_rate (parent flow)
- constant\_bit\_rate (parent flow)
- profile (parent flow)
- level (parent flow)
- sublevel (parent flow)

## TRANSPORT (mux)

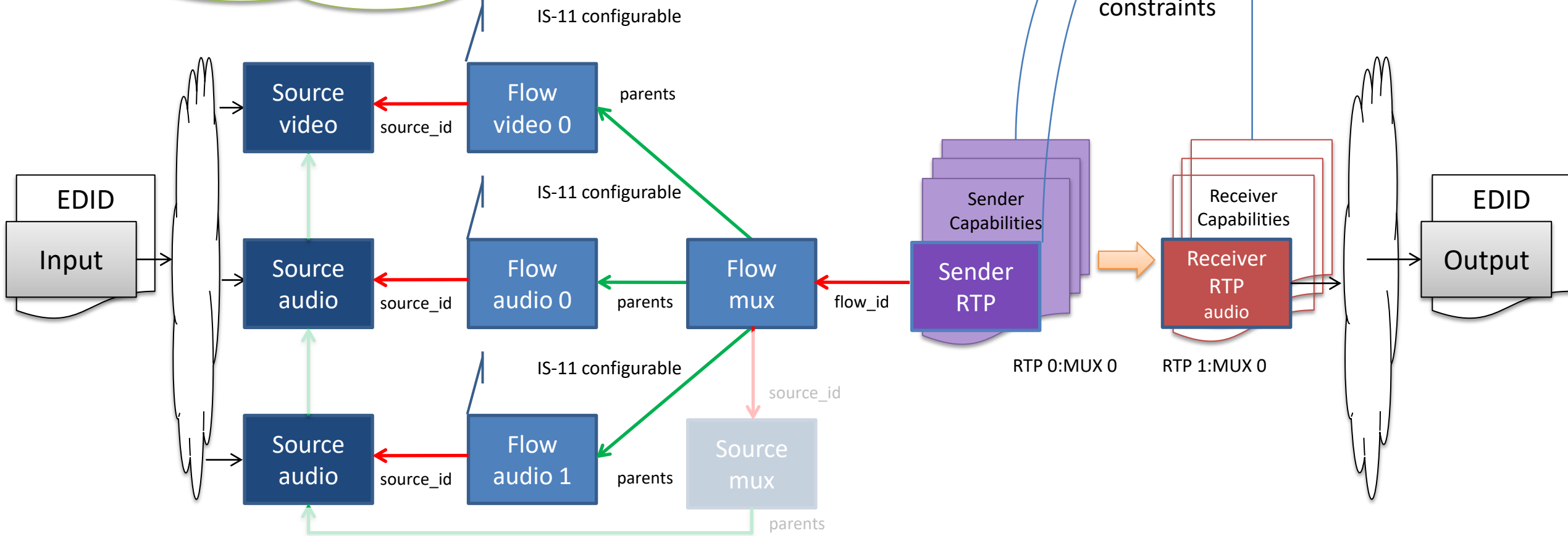
- hkep (sdp) (sender)
- privacy (sdp) (sender)
- clock\_ref\_type (sdp) (source)
- synchronous\_media (sdp) (source)



# Stream Compatibility Assessment

With IS-11 => active constraints

Parameters not supported as active constraints may require an assessment using the actual attributes



# Compatibility Groups

- For multi-part Caps/Cons
  - Group CapSets/ ConSets that are compatible
    - Ex. A multiplexed stream support
      - video: raw or HEVC
      - audio: PCM or AAC
    - If HEVC video is used, then AAC audio must also be used
      - » Put video raw and audio PCM in group 1
      - » Put video HEVC and audio AAC in group 2
  - Leaves and Trunk CapSets/ConSets compatibility groups

- This concludes our overview of NMOS Enhanced Capabilities, a key feature of Matrox NMOS Advanced Streaming Architecture.
- If you have any questions, feel free to reach out at [abouchar@matrox.com](mailto: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/>**