

Presentation on SDP for IP showcase



By Sithideth Viengkhou

April 2019

What is SDP?

- + The Session Description Protocol is a format for describing streaming media parameters. It doesn't deliver any media, but rather helps tie end points together. It is extensible to new media types.
- + It consists of a number of lines of text in the form:
 - + `<type>=<value>`

What is SDP?: Overview

- + There are several sections in a SDP session description:
 - + Session Level:
 - + Starts with a "v=" and ends at the first media-level section
 - + Contains information such as:
 - + v= (protocol version)
 - + o= (originator and session identifier)
 - + s= (session name)
 - + Media Level:
 - + Starts with a "m=" to the next media-level section.
 - + Contains information such as:
 - + m= (media name and transport address)
 - + i=* (media title)
 - + c=* (connection information ---- optional if included at session level)
 - + a=* (zero or more media attribute lines)

What is SDP?: Overview

- + There is also the Time description "section"
 - + $t=$ (time the session is active). In our application it is almost always 0
 - + $r=^*$ (zero or more repeat times)

Important Lines

- + Origin ("o="): Give the originator of the session, who is sending the media
- + Connection Data ("c="): Contains the connection data; base IP Address/ttl
- + Media Description ("m="): Tell if the media is Audio, Video or text and which port port is the media streamed and transport protocol.
- + Attribute ("a="): ST2110 use a few different attributes.
 - + a=rtptime: Is the attribute for the dynamic payload
 - + a=fmtp: Is the attribute for the media format, such as Video format, exact frame rate, colorimetry.
 - + a=ts-refclk: is the reference clock

Simple Yet Effective

- + With a single SDP text, we can describe a single 1080i video stream that a camera generates

```
s=st2110 0-0-0
t=0 0
m=video 20000 RTP/AVP 96
c=IN IP4 239.20.37.1/64
a=source-filter: incl IN IP4 239.20.37.1 192.168.39.37
a=rtpmap:96 raw/90000
a=fmtp:96 sampling=YCbCr-4:2:2; width=1920; height=1080; exactframerate=60000/1001; depth=10; TCS=SDR;
colorimetry=BT709; PM=2110GPM; SSN=ST2110-20:2017; TP=2110TPN; interlace;
a=mediaclock:direct=0
a=ts-refclk:ptp=IEEE1588-2008:08-00-11-ff-fe-22-91-bb:0
```

It Can Describe A Quad Stream That Composes A 12G Video

```
v=0
o=- 1443716955 1443716955 IN IP4 192.168.39.190
s=st2110 0-0-0
t=0 0
a=group:MULTI-2SI 0 1 2 3
m=video 1000 RTP/AVP 96
c=IN IP4 239.20.45.0/64
a=source-filter: incl IN IP4 239.20.45.0 192.168.39.104
a=rtpmap:96 raw/90000
a=fmtp:96 sampling=YCbCr-4:2:2; width=1920; height=1080; exactframerate=60000/1001; depth=10; TCS=SDR; colorimetry=BT709; PM=2110GPM; SSN=ST2110-20:2017; TP=2110TPN;
a=mediaclock:direct=0
a=ts-refclk:ptp=IEEE1588-2008:08-00-11-ff-fe-22-91-bb:0
a=mid:0
m=video 1001 RTP/AVP 96
c=IN IP4 239.20.45.1/64
a=source-filter: incl IN IP4 239.20.45.1 192.168.39.104
a=rtpmap:96 raw/90000
a=fmtp:96 sampling=YCbCr-4:2:2; width=1920; height=1080; exactframerate=60000/1001; depth=10; TCS=SDR; colorimetry=BT709; PM=2110GPM; SSN=ST2110-20:2017; TP=2110TPN;
a=mediaclock:direct=0
a=ts-refclk:ptp=IEEE1588-2008:08-00-11-ff-fe-22-91-bb:0
a=mid:1
m=video 1012 RTP/AVP 96
c=IN IP4 239.20.45.12/64
a=source-filter: incl IN IP4 239.20.45.12 192.168.39.105
a=rtpmap:96 raw/90000
a=fmtp:96 sampling=YCbCr-4:2:2; width=1920; height=1080; exactframerate=60000/1001; depth=10; TCS=SDR; colorimetry=BT709; PM=2110GPM; SSN=ST2110-20:2017; TP=2110TPN;
a=mediaclock:direct=0
a=ts-refclk:ptp=IEEE1588-2008:08-00-11-ff-fe-22-91-bb:0
a=mid:2
m=video 1013 RTP/AVP 96
c=IN IP4 239.20.45.13/64
a=source-filter: incl IN IP4 239.20.45.13 192.168.39.105
a=rtpmap:96 raw/90000
a=fmtp:96 sampling=YCbCr-4:2:2; width=1920; height=1080; exactframerate=60000/1001; depth=10; TCS=SDR; colorimetry=BT709; PM=2110GPM; SSN=ST2110-20:2017; TP=2110TPN;
a=mediaclock:direct=0
a=ts-refclk:ptp=IEEE1588-2008:08-00-11-ff-fe-22-91-bb:0
a=mid:3
```

Simply add “a=group”

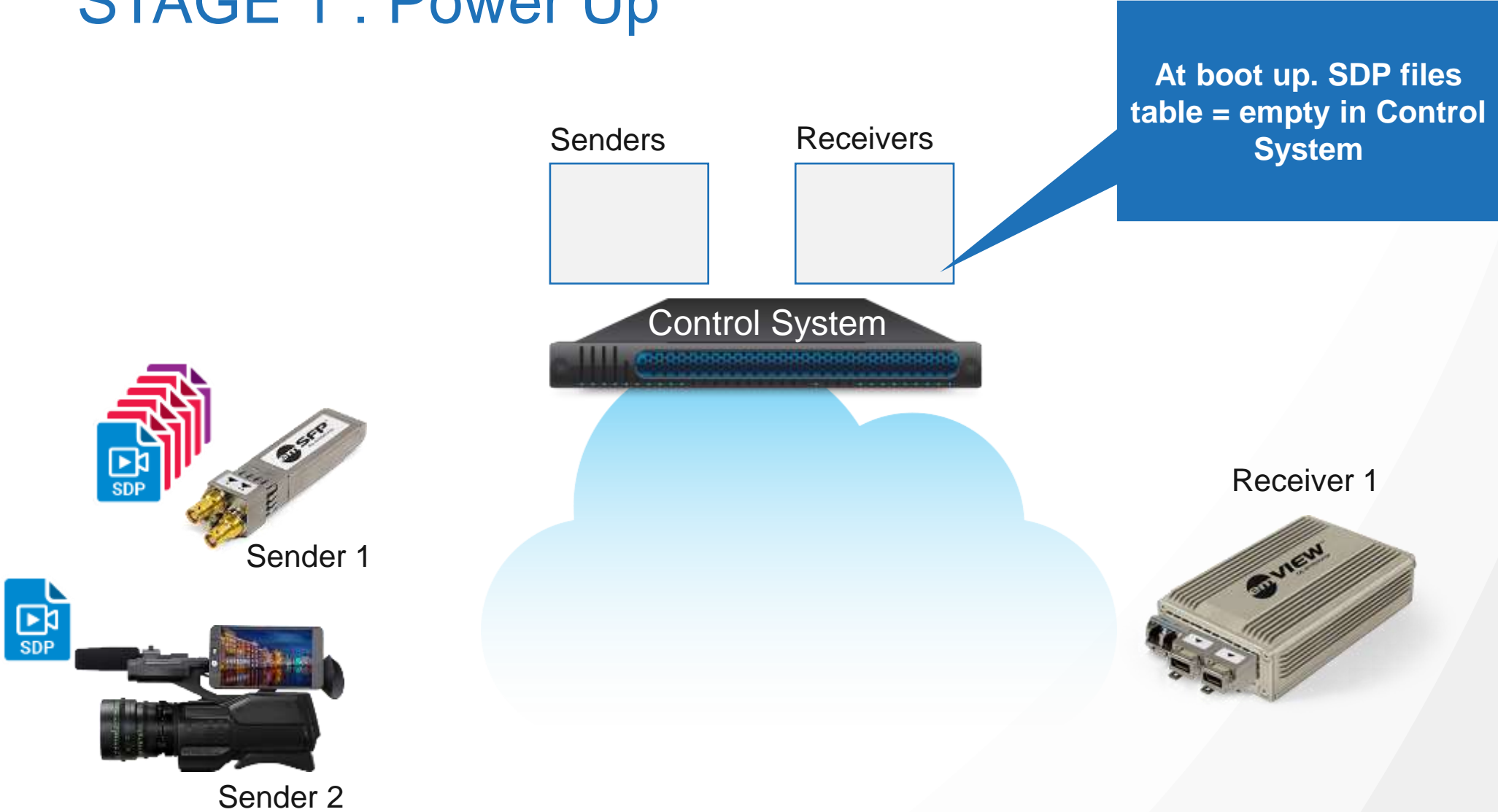
And “a=mid:”

Simple Mechanics: Connecting A Sender To A Receiver

- + This is a real, simplistic example that doesn't take into account all the work in IS04/05 or other protocols that use SDP such as EMBER +

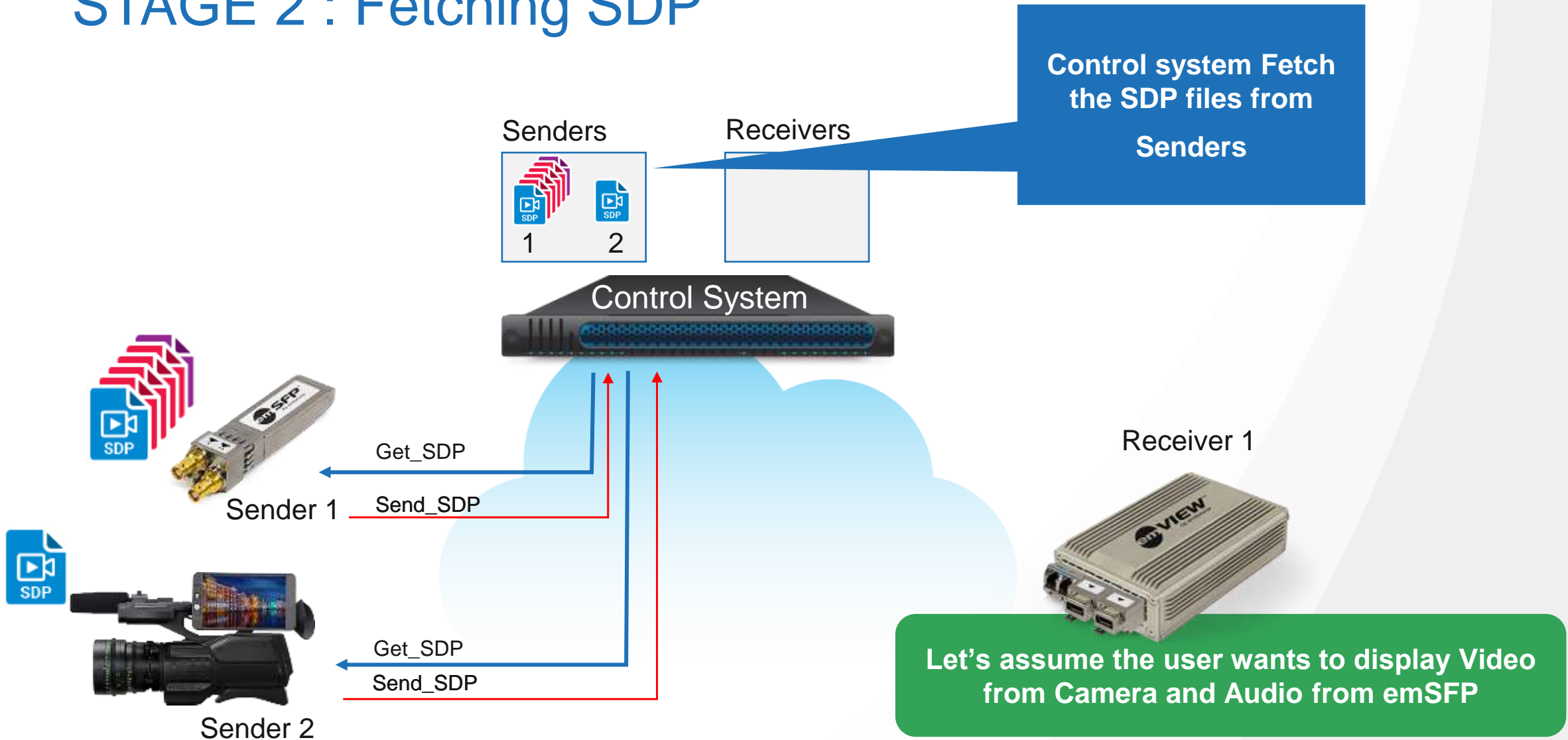
IP System Boot Up & Exchange Of SDP Files

STAGE 1 : Power Up



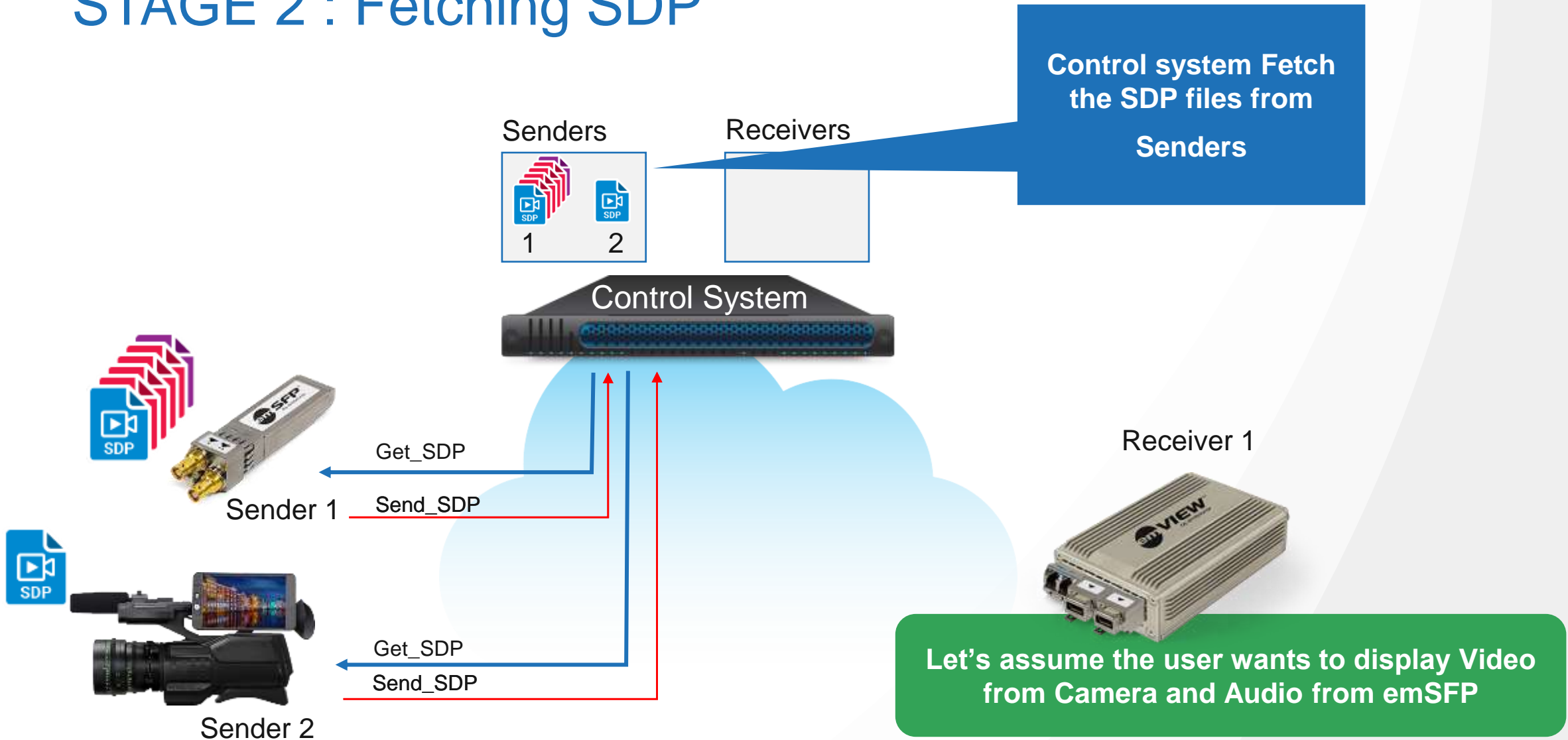
IP System Boot Up & Exchange Of SDP Files

STAGE 2 : Fetching SDP



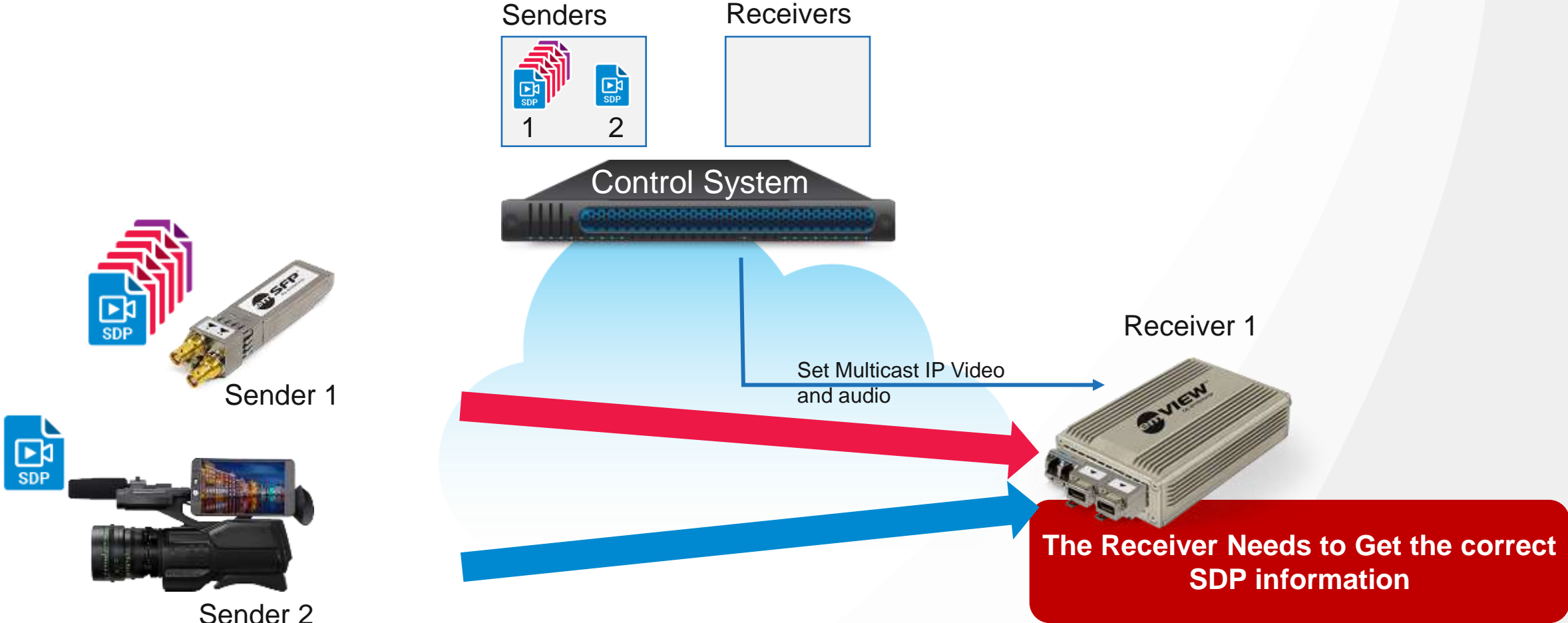
IP System Boot Up & Exchange Of SDP Files

STAGE 2 : Fetching SDP



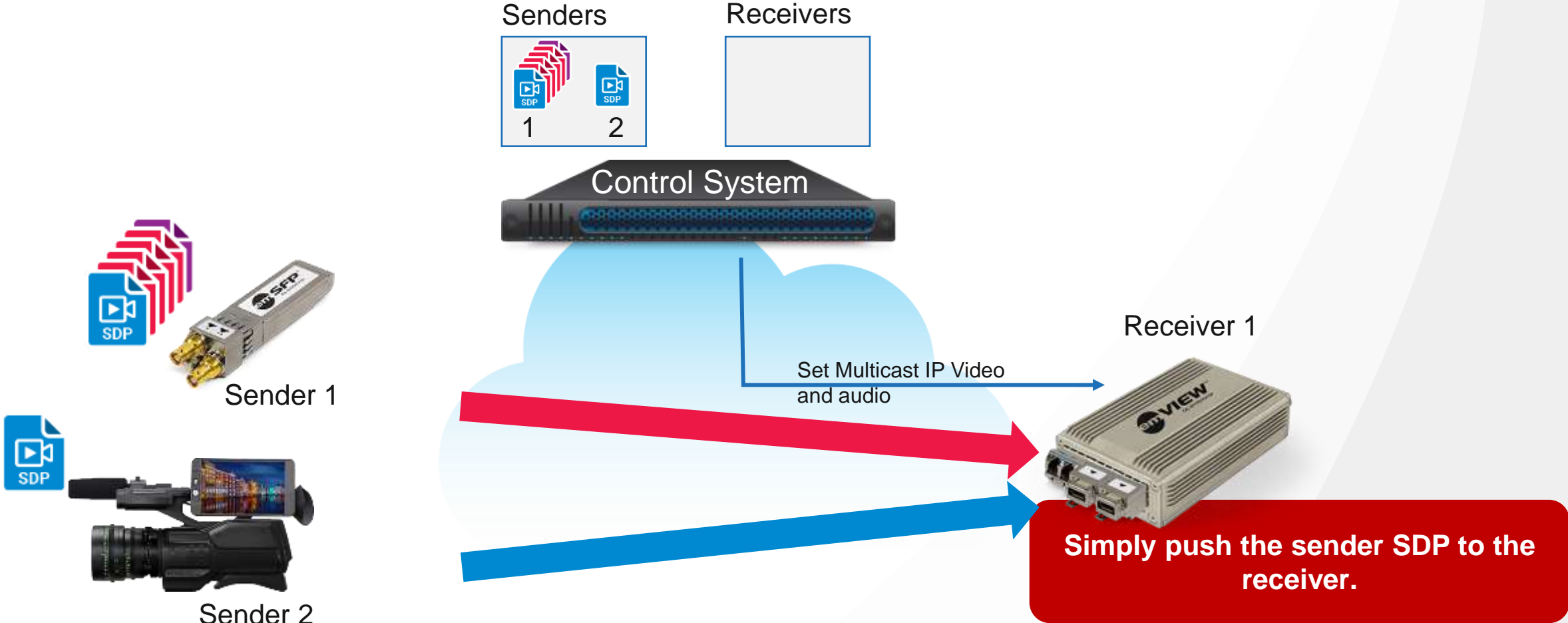
IP System Boot Up & Exchange Of SDP Files

STAGE 3 : Configuring Receiver



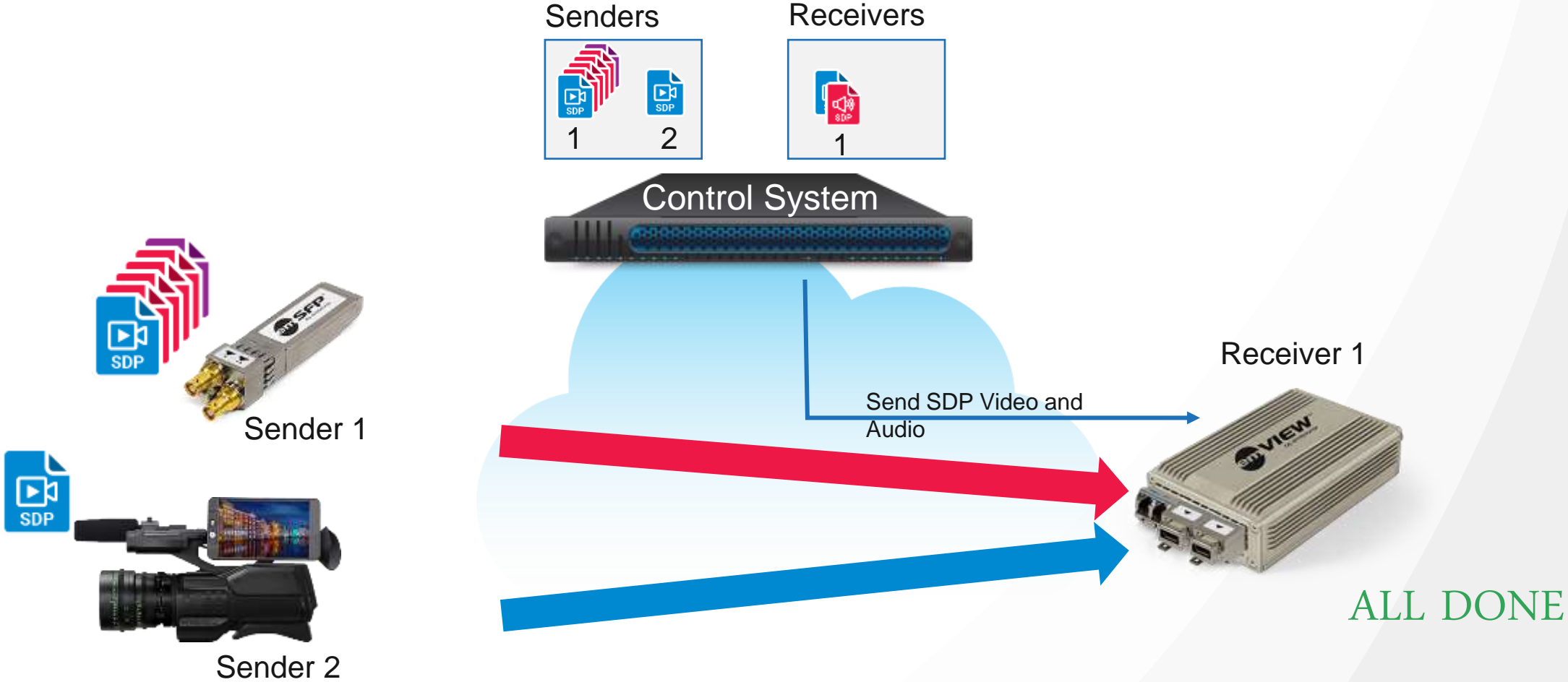
IP System Boot Up & Exchange Of SDP Files

STAGE 3 : Configuring Receiver



IP System Boot Up & Exchange Of SDP Files

STAGE 4 : Push SDP in Receiver



Potential Improvements

- + What if for a small installation, the media stream would be self descriptive with the SDP being a companion stream such as SAP (Session Announcement Protocol). Where the SDP of the streamed media is periodically streamed by the Sender. So any change in media stream, such as format change will also send to listening receiver(s).

Thanks!

