

OpenVivoe

Def-Stan 00-82 Open Source
implementation

Summary



✓ Presentation

✓ Behaviour

✓ Use - Configuration

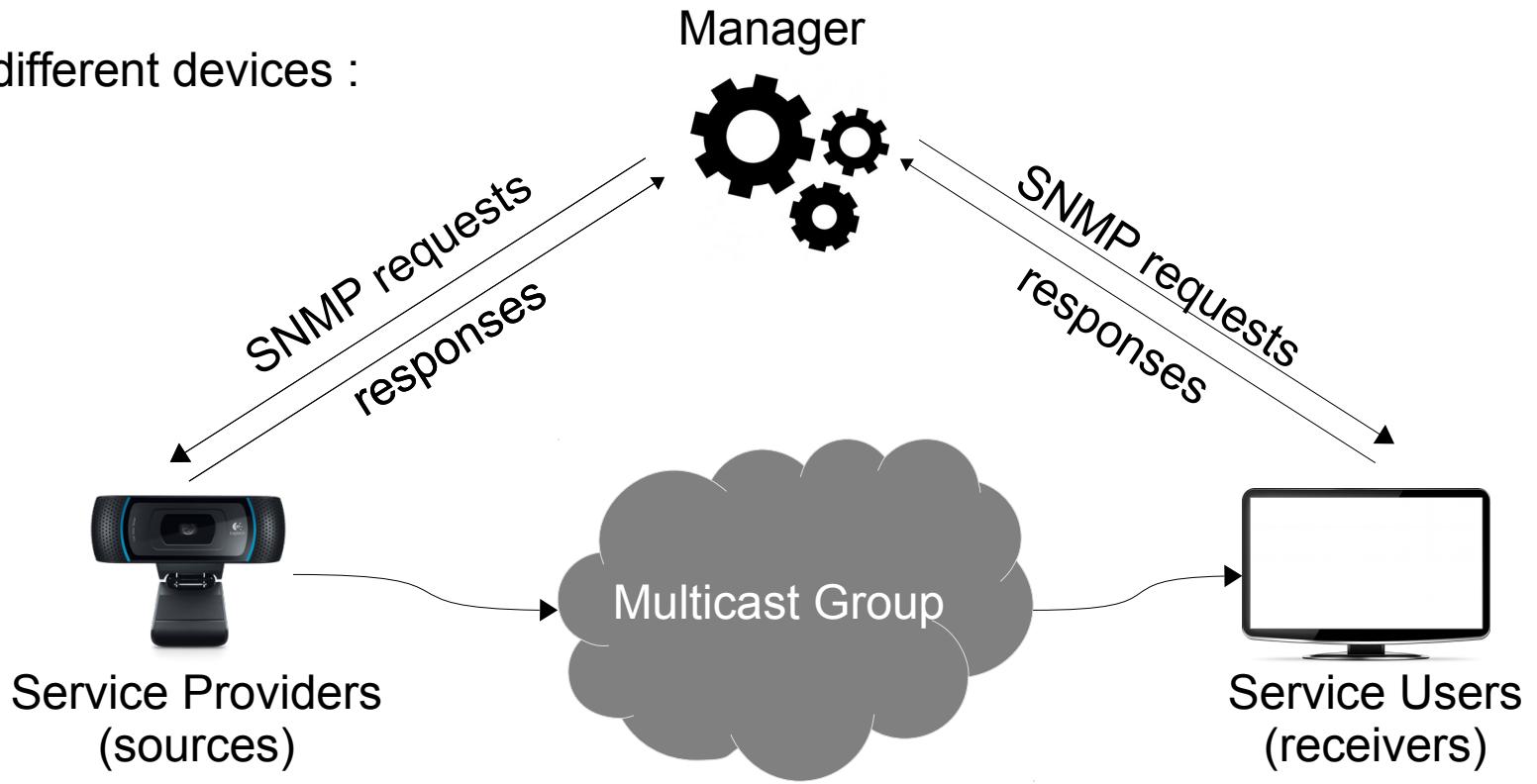
✓ Features

✓ OW's Interest

✓ Questions

How does it work ?

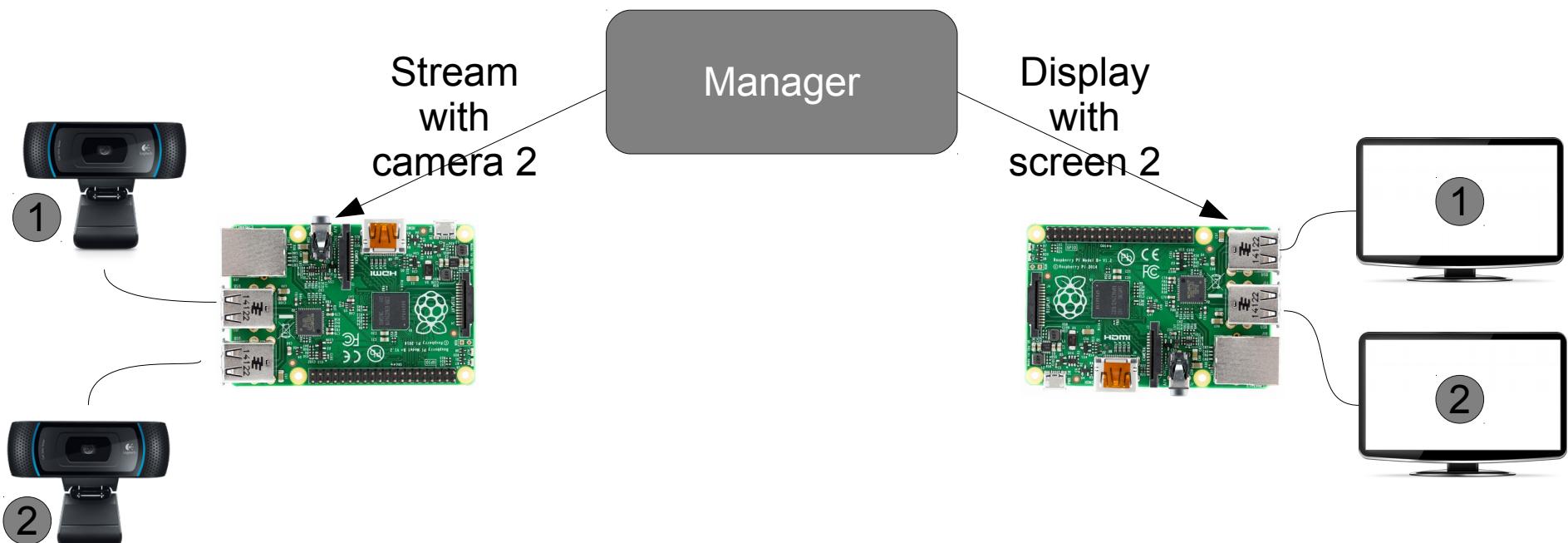
- 3 different devices :



- The manager controls the sources and the receivers. It orders them when to start streaming and when to stop.
- Sources send streams to multicast IP addresses.
- Receivers join multicast groups to receive the streams.

SNMP – streaming management

- SNMP (Simple Network Management Protocol) : manage devices on the network using their “database” called MIB (Management Information Base).
- Send request from manager to Service Provider and Service User to:
 - ✓ Get information on them
 - ✓ Set parameters for streaming
 - ✓ Start/Stop streaming



- The Open Source implementation of a military standard VIVOE. (**Vetronics Infrastructure for Video Over Ethernet**).
- Aims to multicast video streams between devices inside a terrestrial vehicles over an ethernet network.

OpenVivoe is built on several dependencies:



3.5.0



5.7.3



2.48.1



1.8.0

User gives to OpenVivoe (through configuration file):

- Information on each device (description, name of network interface,...)
- **For sources:** a way to retrieve the video source (a gstreamer pipeline command)
- **For receivers:** a way to handle the received video (a gstreamer pipeline command) and a multicast address to listen.

OpenVivoe:

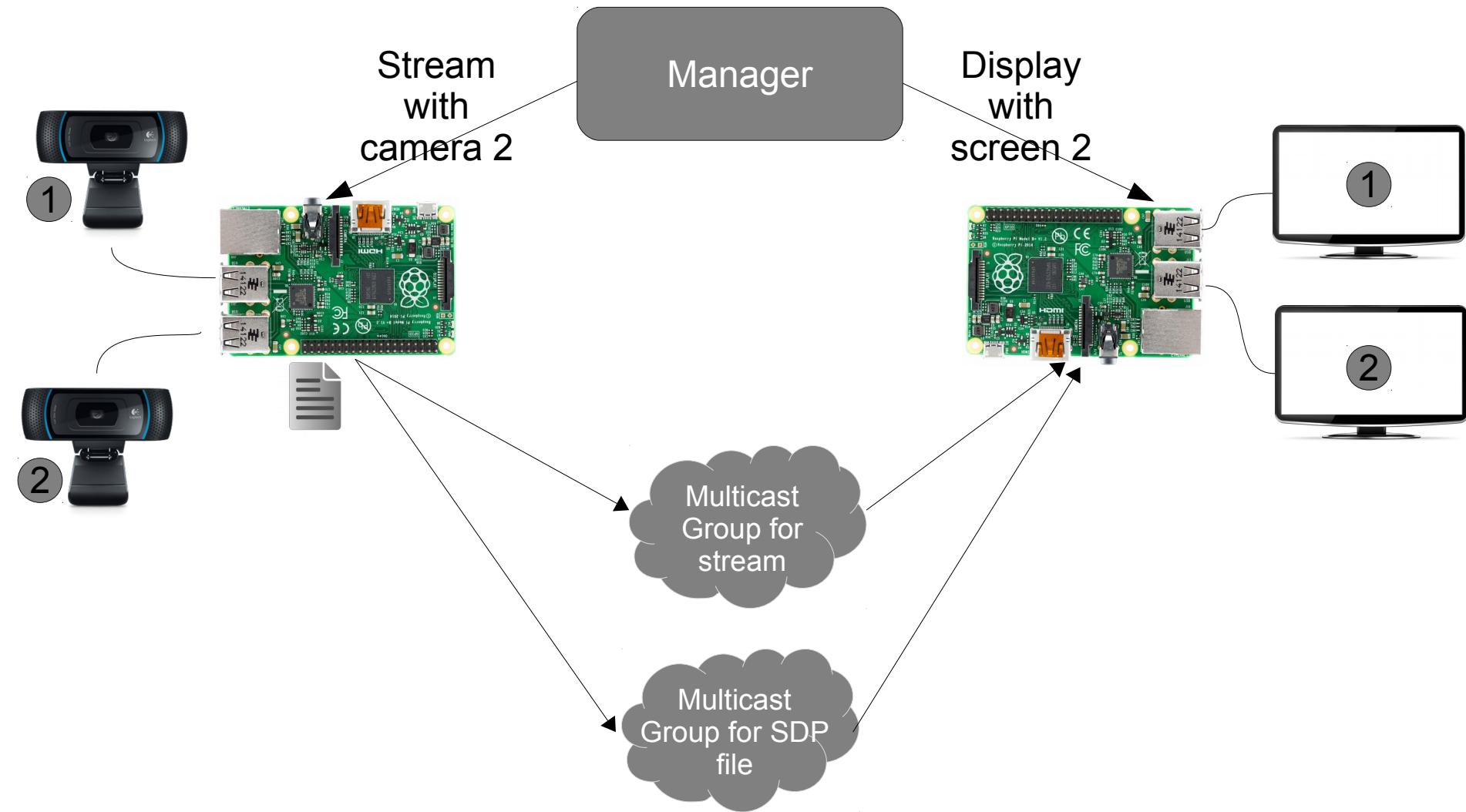
1) On Service Provider's side

- Retrieves the video
- payloads it into RTP packets
- creates a SDP message to announce the stream
- multicasts them on computed multicast IP addresses.

1) On Serve User's side:

- Retrieves the stream from the multicast group
- retrieves the SDP file
- depayloads the video
- handles the stream as specified by the user.

Example



The configuration file



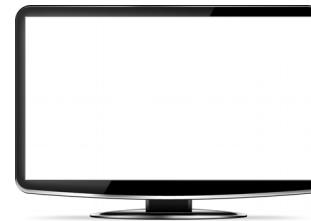
```
[deviceInfo]
DeviceType = 1

[source_1]
ChannelUserDesc = Describe camera1
gst_source = v4l2src \
device=/dev/video0 ! avenc_mpeg4

[source_2]
ChannelUserDesc = Describe camera2
gst_source = v4l2src \
device=/dev/video1 ! openjpegenc
```

Device's IP: 192.168.1.1

Source 1 channel (1) multicast IP: 239.192.1.1
Source 2 channel (2) multicast IP: 239.192.2.1



```
[deviceInfo]
DeviceType = 2

[receiver_1]
DefaultReceiveIP = 239.192.1.1
gst_sink = avdec_mpeg4 ! xvimagesink

[receiver_2]
DefaultReceiveIP = 239.192.2.1
gst_sink = openjpegdec ! xvimagesink
```

Screen 1 decodes stream received from device 1, channel 1 and displays it.
Screen 2 decodes stream received from device 1, channel 2 and displays it.

Normal Mode Behaviour



[REC]

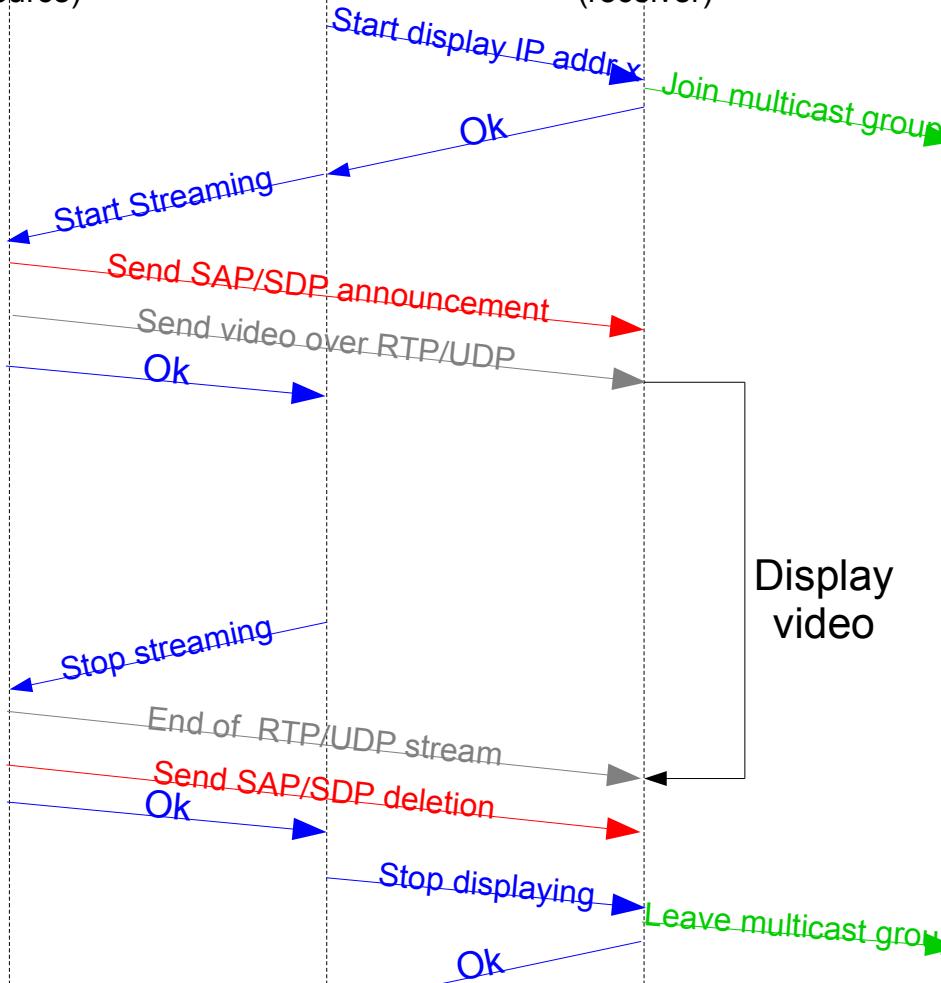


Service
Provider
(source)

Manager

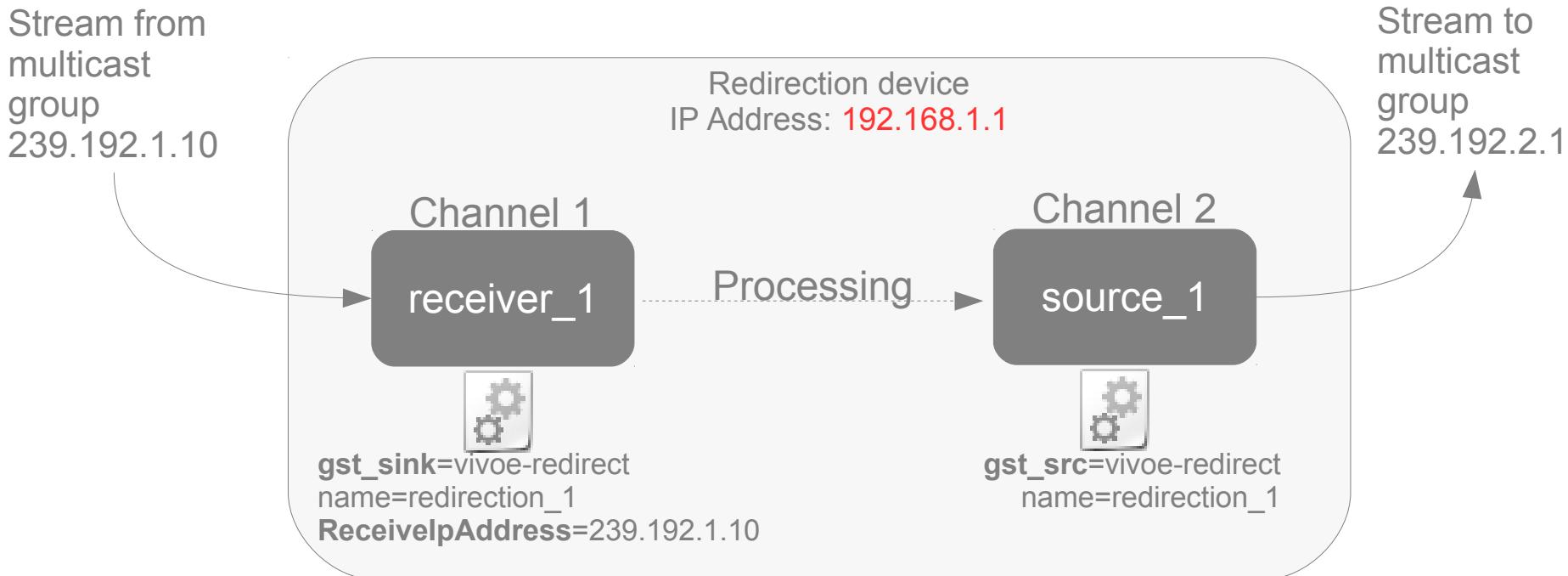
Service
User
(receiver)

Network



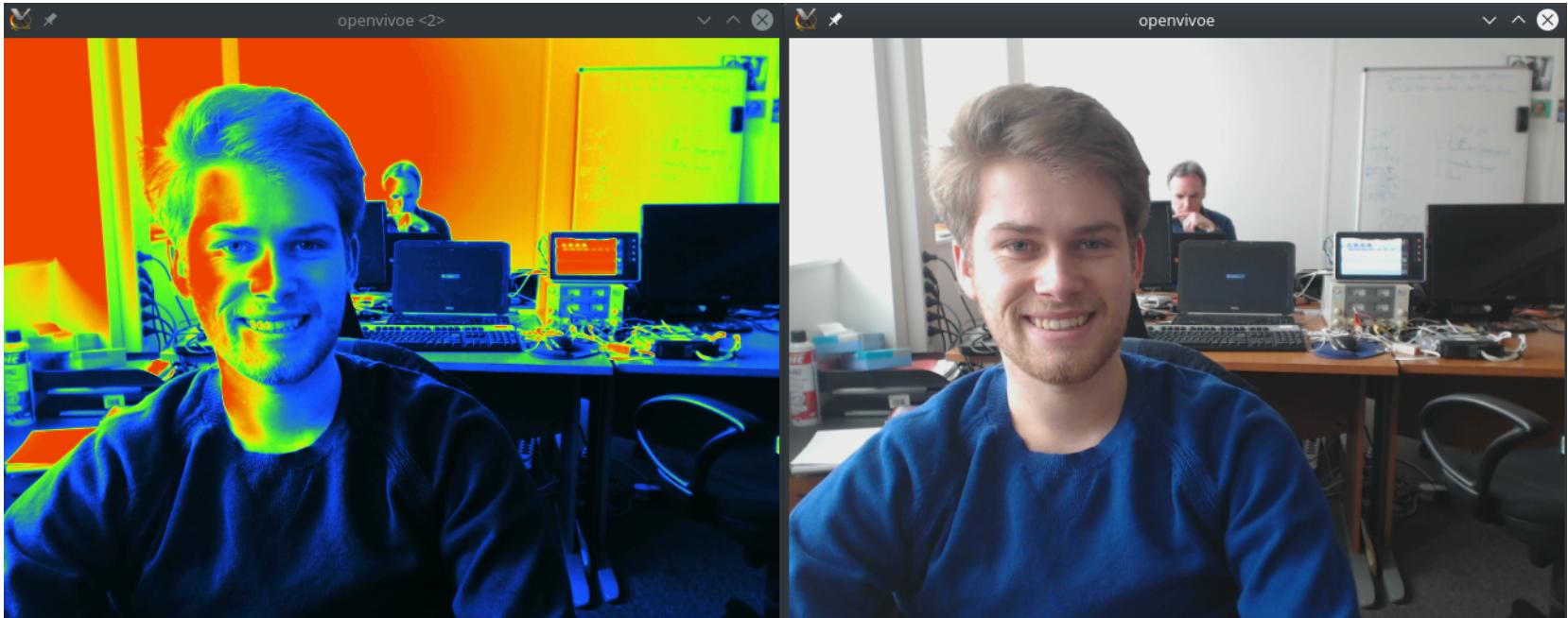
The Redirection

- Additional feature implemented in OpenVivo
- **Principle:** A device that is both Service Provider and Service User will redirect the stream it receives on its “Service User” channel to its “Service Provider” channel and streams it on a different multicast address.



- **Interest:** performs a special processing on the received video, streams it to a new multicast group.
- **Example:** convert a stream from MPEG-4 to JPEG2000 to have the same images in two different encodings.

Example of redirection

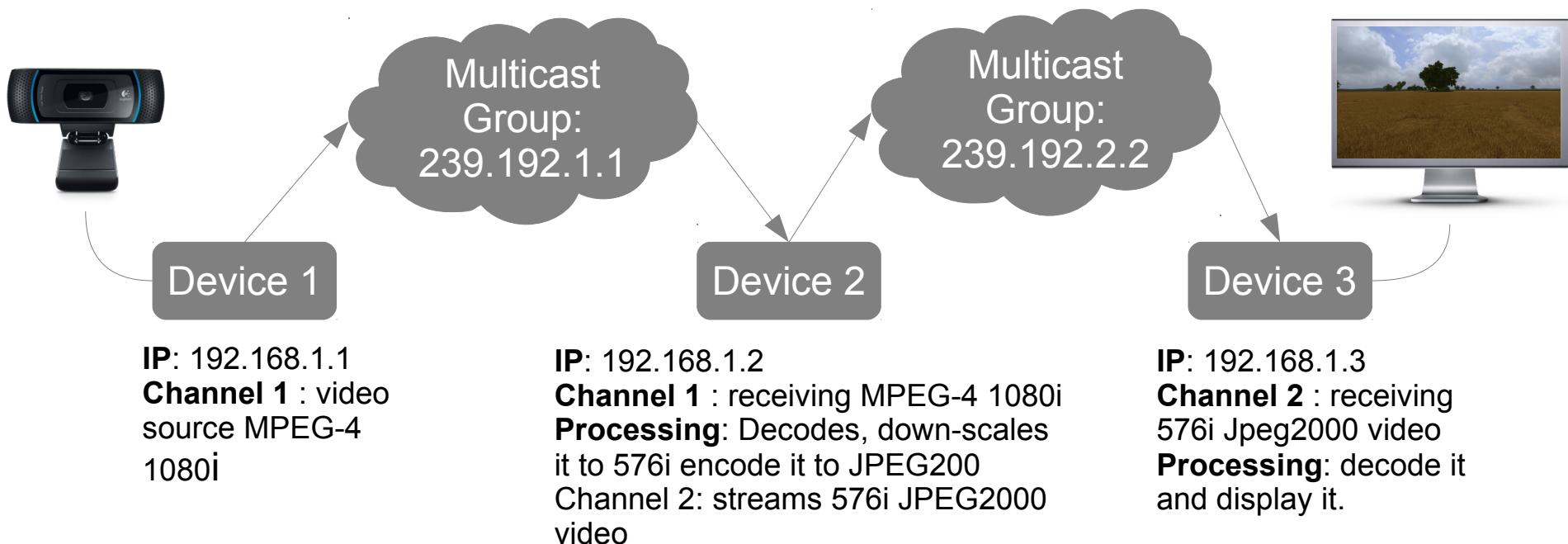


- 3 devices: A source, a redirection device, a display.
- The redirection device applies a thermal filter on the stream received and multicasts the result.

Complex behaviour - donw-scaling

donw-scaling video converter – encoder

- 3 devices:
 - Device 1: Service Provider
 - Device 2: Both
 - Device 3: Service User
- Principle: device 1 streams a A MPEG-4, 1080i video. This video is received by device 2, downscaled to 576i video and the new video is displayed by device3



Device one's configuration file

```
[deviceInfo]
deviceDesc = provider 1

deviceType = 1

ethernetInterface = enp2s0

deviceMode = 2

deviceReset = 1

[source_1]
channelUserDesc = Camera front side left

gst_source = v4l2src device=/dev/video0 ! capsfilter caps="video/x-
raw,width=1920,height=1080,interlace-
mode=(string)interlace,framerate=(fraction)30/1" ! avenc_mpeg4
```

Device two's configuration file

[deviceInfo]

DeviceDesc = redirection device

DeviceType = 3

EthernetInterface = eth0

DeviceMode = 2

DeviceReset = 1

[receiver_1]

defaultReceiveIp = 239.192.1.1

gst_sink = queue ! avdec_mpeg4 ! videoconvert ! videoscale ! caps="video/x-raw,width=720,height=576" ! queue ! openjpegenc ! viveo-redirect name=r1

[source_1]

channelUserDesc=redirection from 239.192.1.1 to 239.192.2.2

gst_source = viveo-redirect name=r1

Device three's configuration file

```
[deviceInfo]
deviceDesc = receiver 1

deviceType = 2

ethernetInterface = enp2s0

deviceMode = 2

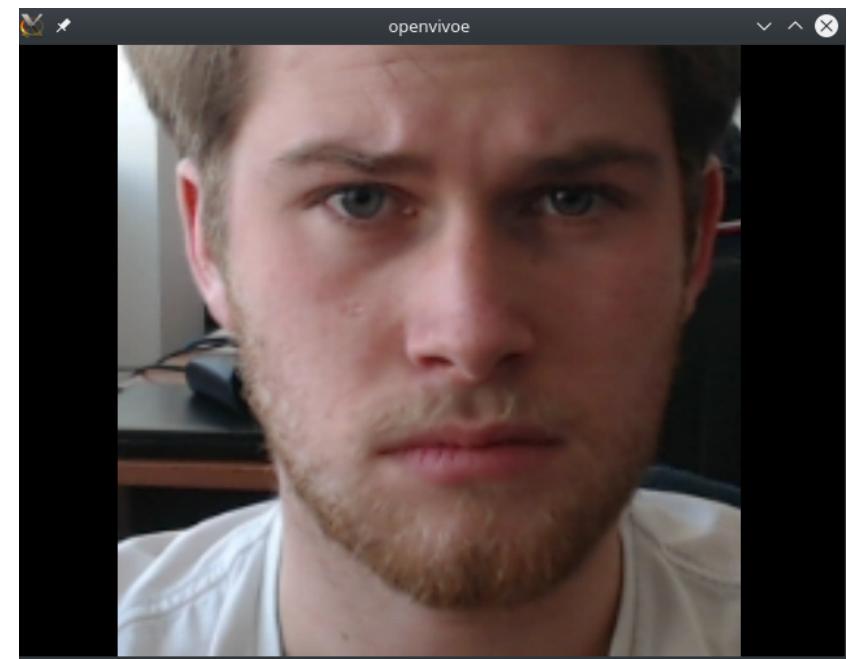
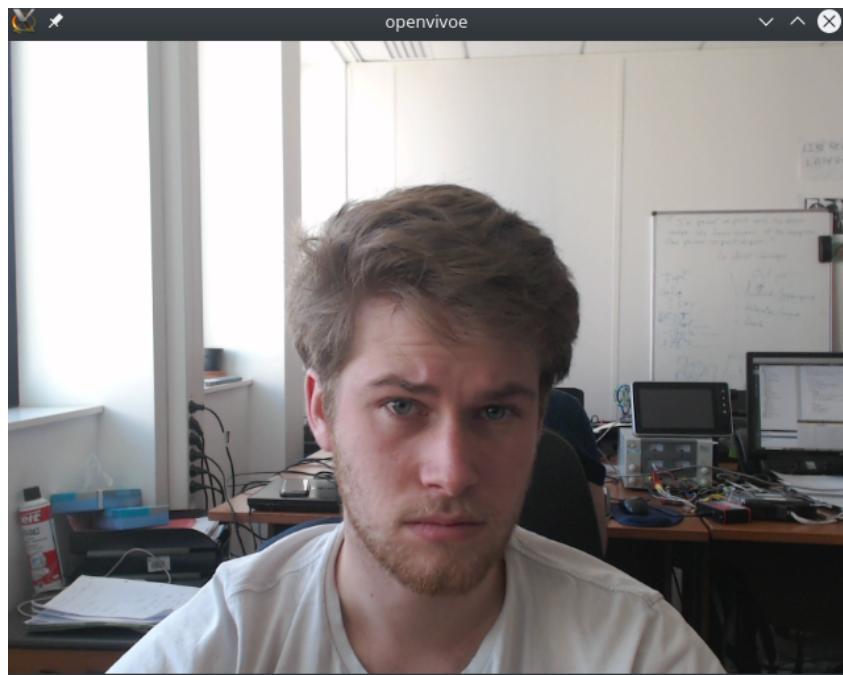
deviceReset = 1

[receiver_1]
defaultReceiveIp = 239.192.2.2

gst_sink = openjpegdec ! Videoconvert ! autovideosink
```

Regions of Interest : Roi

- Features: control Region of Interest within the MIB and SNMP.
- 2 kinds of Roi:
 - Scalable (interpolated or decimated)
 - Non-scalable



Regions of Interest : Roi



ROI non
scalable



ROI
interpolated



ROI
decimated



channel
VertRes

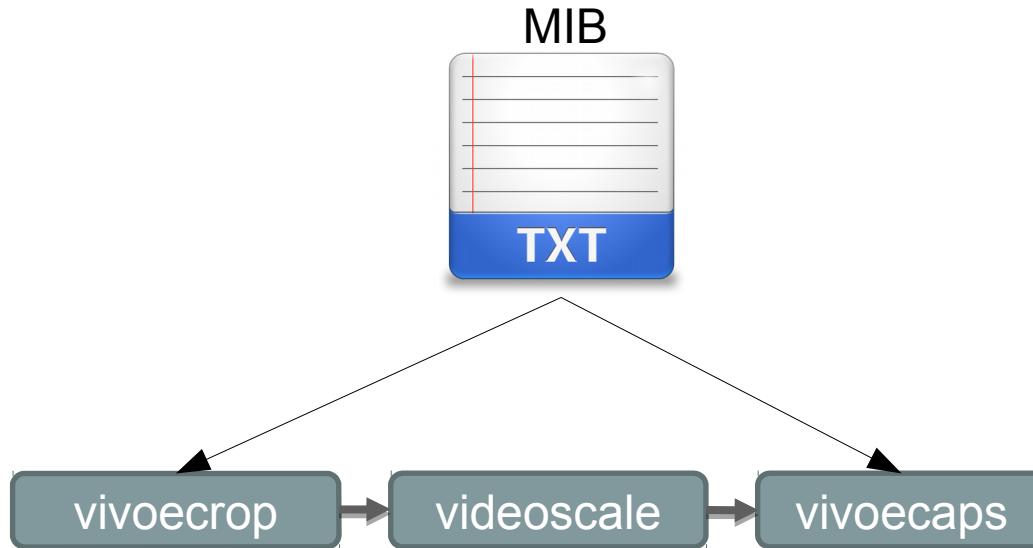


channel
VertRes



channel
VertRes





[source_1]

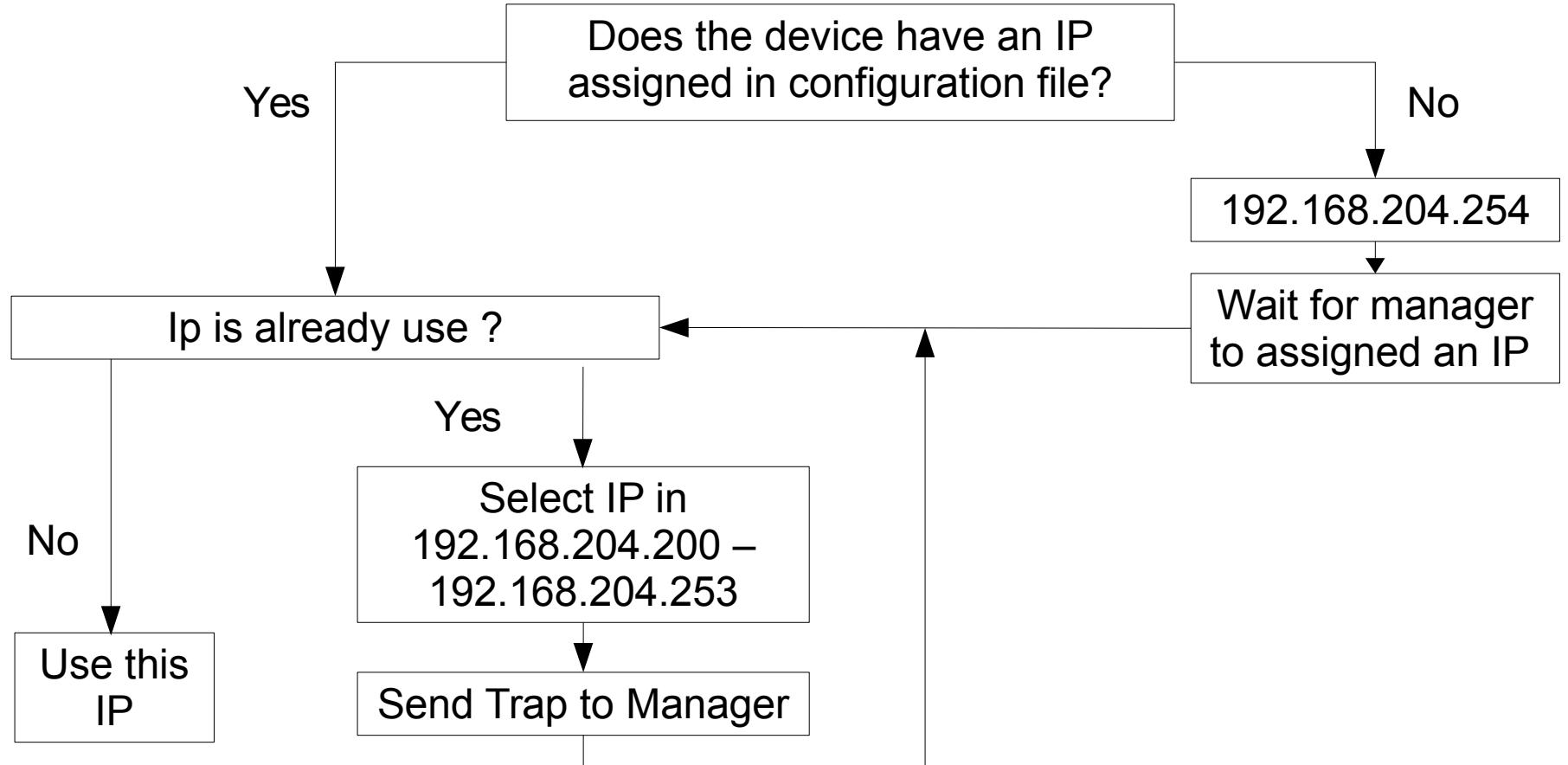
```
channelUserDesc = Camera with ROI
```

```
gst_source = gst_source=v4l2src device=/dev/video0 ! capsfilter caps="video/x-raw,format=I420,width=640,height=480,interlace-mode=(string)progressive,framerate=(fraction)20/1" ! vivoecrop ! videoscale ! vivoecaps ! avenc_mpeg4
```

- Vivoecrop and vivoecaps behave like videocrop and capsfilter, except that they retrieve their properties directly from the MIB

VVOOE IP assignment scheme

- The VVOOE standard commands to use a specific static IP assignment scheme.
- Here is how it is implemented in OpenVivoee:



- Not totally implemented yet.

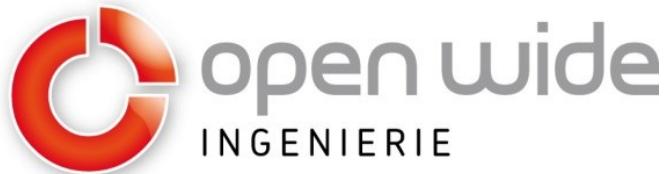
- Open Vivoe uses the **GNU General Public License version 2**.
- It is a completely free and open source software.
- Source code is available on GitHub at :
<https://github.com/Openwide-Ingenierie/openvivoe>
- A Website on the project has also been created at:
<http://openwide-ingenierie.github.io/openvivoe/>





- ☛ Follow Open Source news on LinuxEmbedded.fr
- ☛ Soon an article about OpenVivoe will be published

THANKS !



23, rue David
75013 Paris
Tél. : 01 42 68 28 00

36, rue Jacques Babinet
31100 Toulouse
Tel : 06 08 73 05 92

Your contact: Rosen Jérémie – Phone– jeremy.rosen@smile.fr

Your contact: Vasseur Hoël – 06 85 51 56 48– hoel.vasseur@smile.fr