OPEN SOURCE SOFTWARE

A Tool for Digital Transformation in the Broadcasting Industry
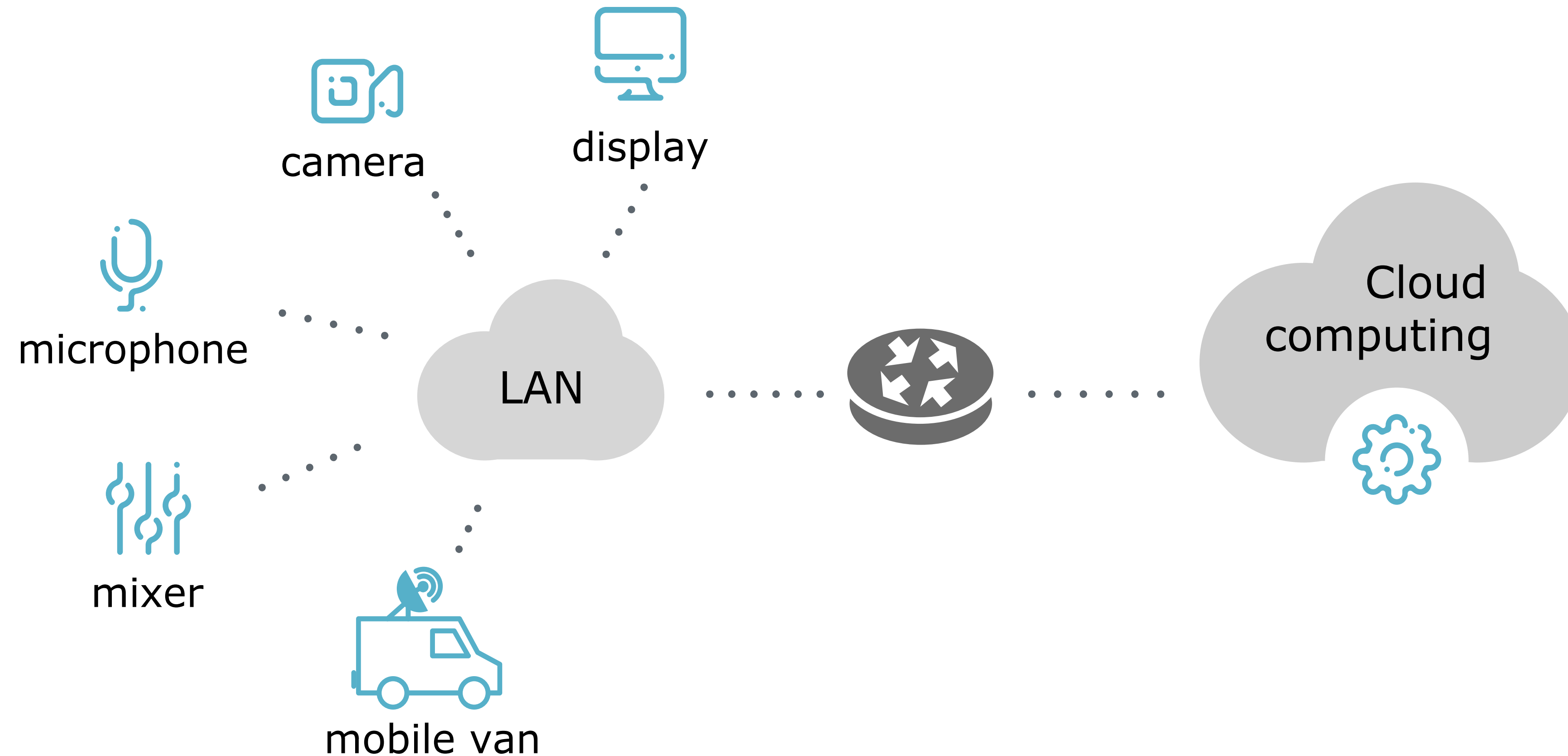
Eloi Bail (MSc)
Senior Software Consultant
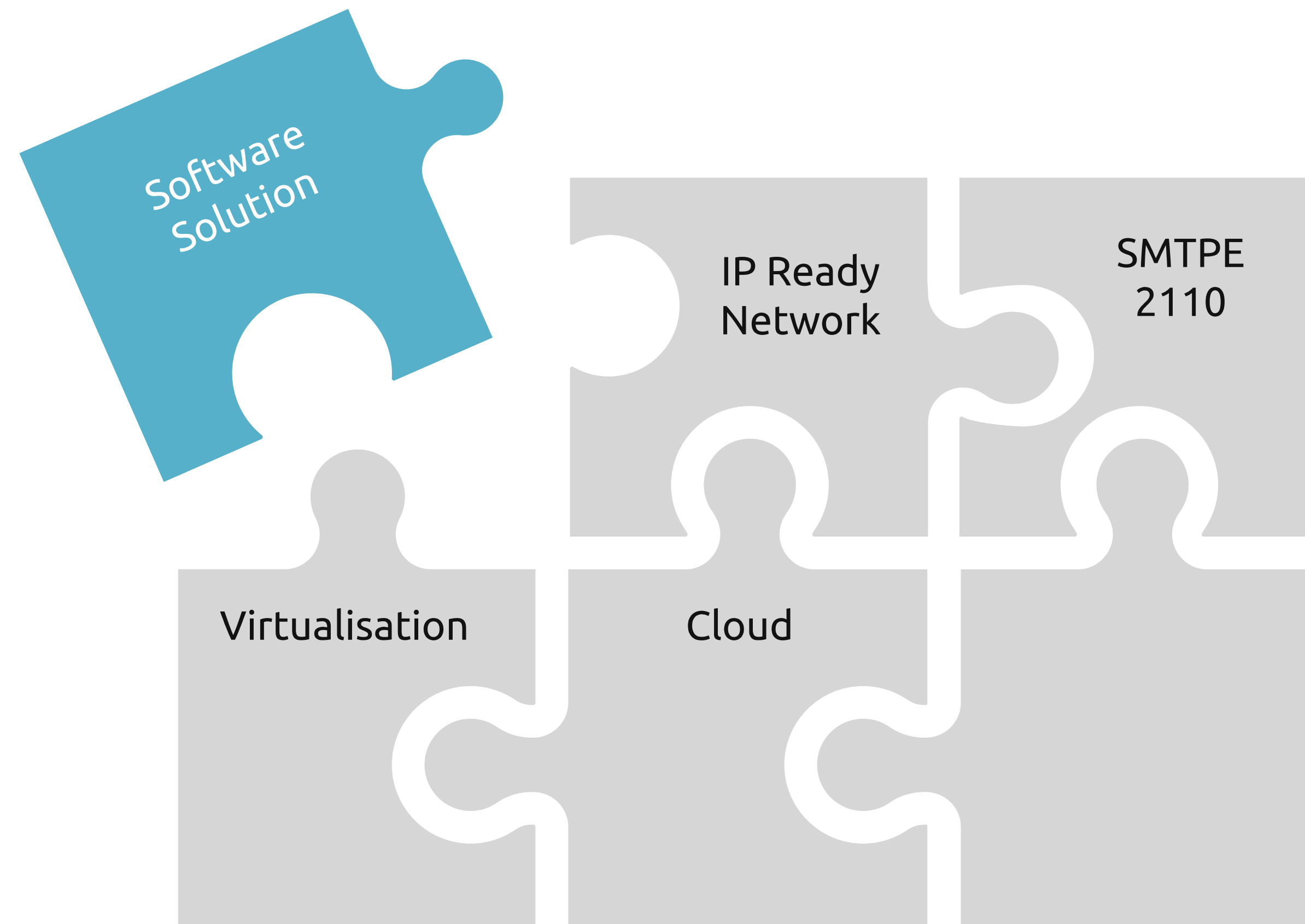
Amir Taherizadeh (Ph.D., M.B.A.)
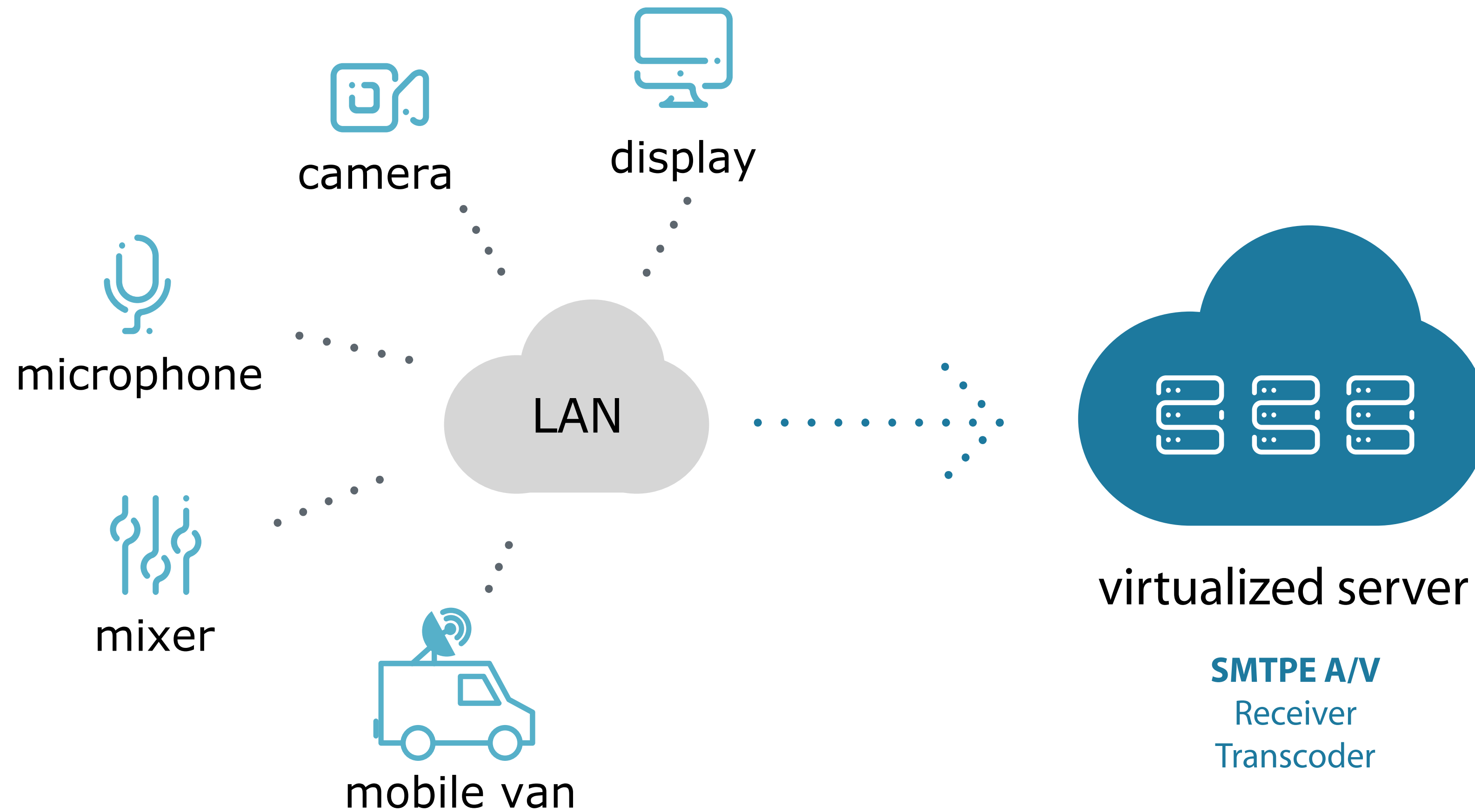*Tech. Marketing and Business Development Consultant*

The target: handle IP content in the cloud computing

camera

display

microphone

LAN

Cloud computing

mixer

mobile van

The solution: Software

Software Solution

IP Ready Network

SMTPE 2110

Virtualisation

Cloud

The target: Handle IP content in the cloud computing

camera

display

microphone

LAN

mixer

mobile van

virtualized server

SMTPE A/V
Receiver
Transcoder

What we have achieved

camera

display

microphone

LAN

mixer

mobile van

dedicated server

SMTPE A/V
Receiver
Transcoder

WAN
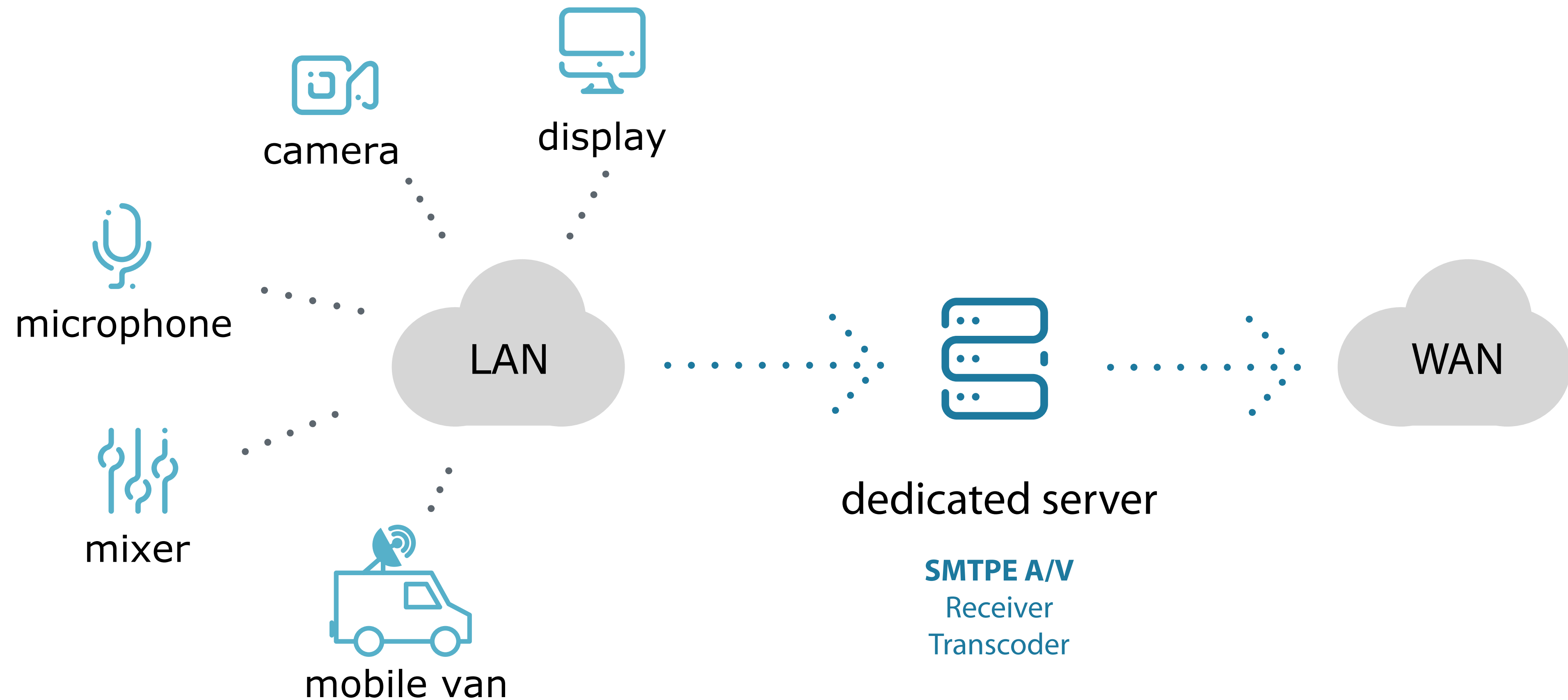
SMTPE 2110: Why software ?

Flexibility

Scalability
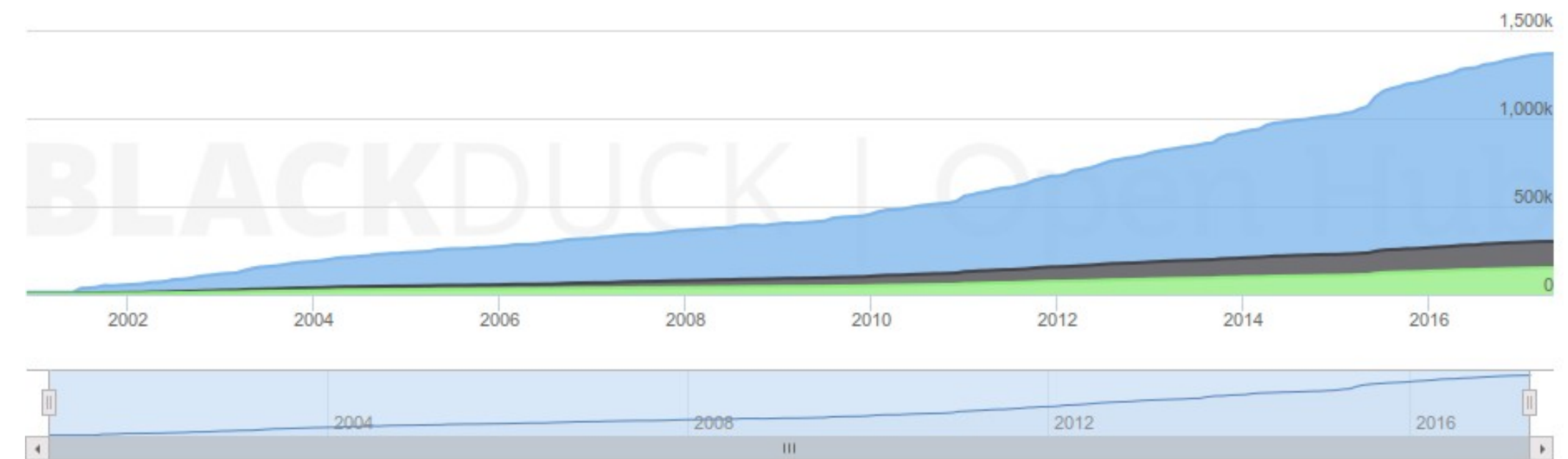
Continuous enhancement

FFmpeg

FFMPEG

What is it?

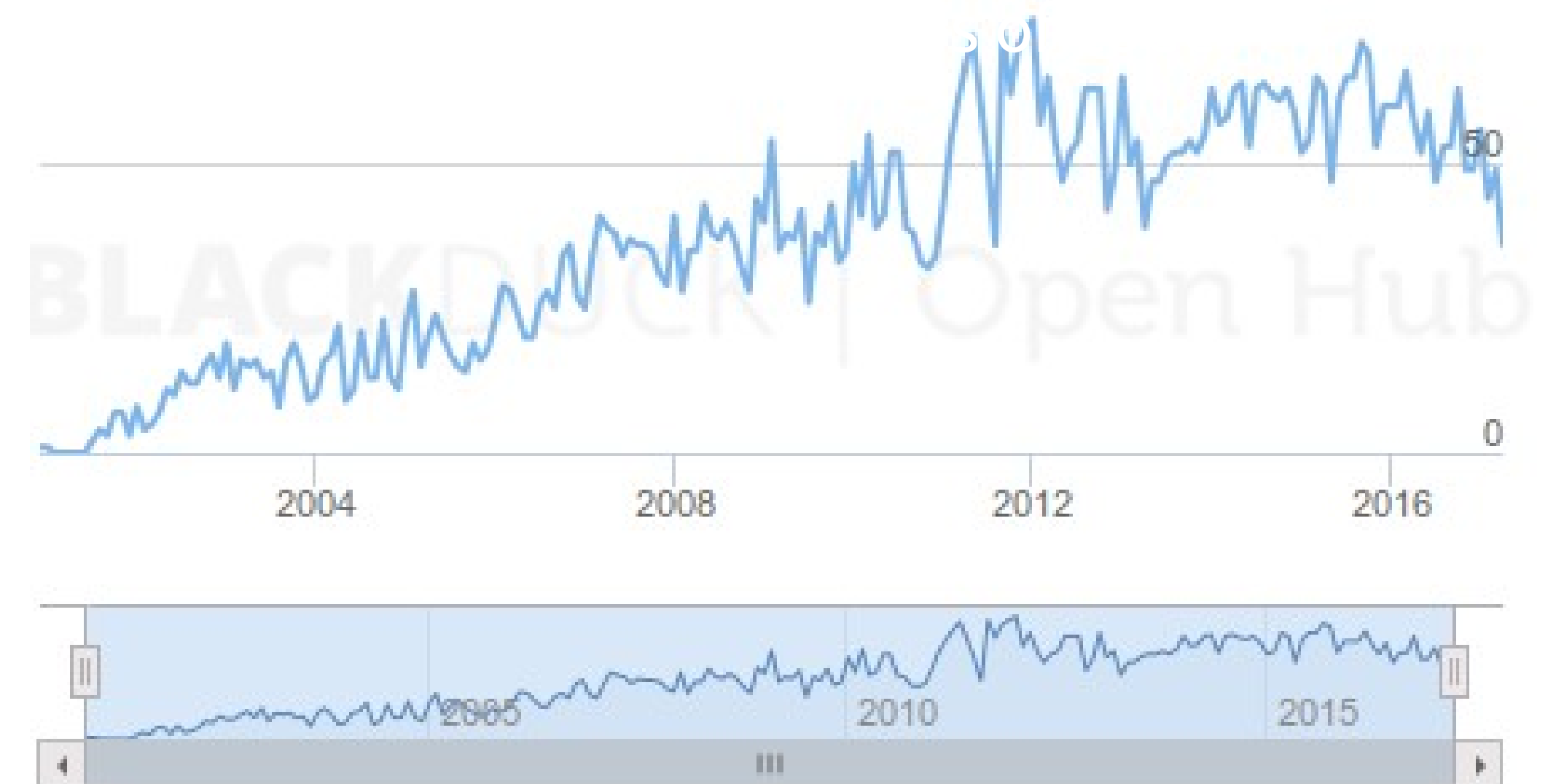A multimedia framework dealing with multimedia formats of all sorts

How long it has been around

Flourishing for 15 years with multiple applications in a variety of industries

It is the bedrock of many mainstream multimedia *(ex: VLC)*
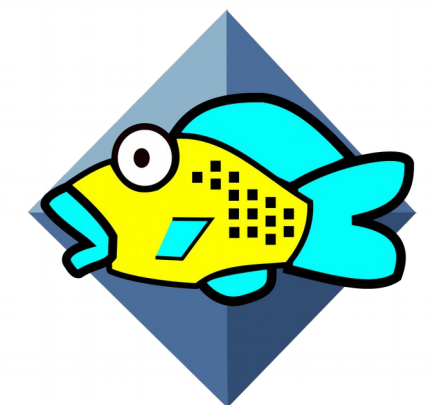

Lines of code

Contributors per month

FFMPEG

A modular solution

- Set of command line tools
- Support of many codecs (**400**)
- Support of many containers
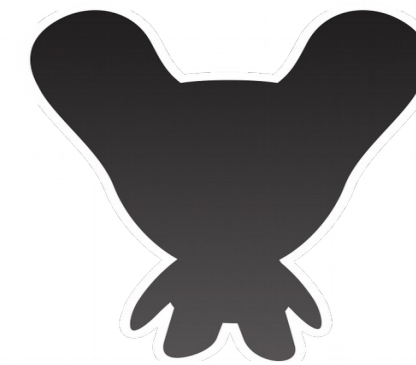- Support of many devices
- Support of many muxers / demuxers

MPEG4

HEVC
High Effecieny Video Coding

H.264
MPEG-4/AVC

VP8 VP9

DOLBY
DIGITAL

opus

AAC

MKV

flac
free lossless audio codec

FFMPEG

Companies contributing
- No full list
- According to contribution

Example of products:

http://pro.jvc.com/pro/attributes/cctv/manual/LST1363-001C_VN-T16U_INSTRUCTIONS_v3.0.pdf

https://docs.sony.com/release//Sup_4584804312.pdf

FFMPEG
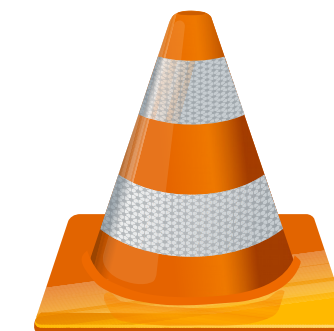
Why Radio Canada chose FFmpeg ?

1. Mature solution
2. Already used internally
3. Match their performance expectations

First contact with the community

- "A SMTPE 2110 receiver with FFmpeg, it is impossible !"

Alternatives

- Gstreamer: not used internally
- Upipe: not mature enough

Our Work with Radio-Canada

Our work with Radio Canada

1. **Hire SFL for our expertise in open source systems**

   - Establish the link between Radio Canada and the FFmpeg community

     - Understand the point of view of the community
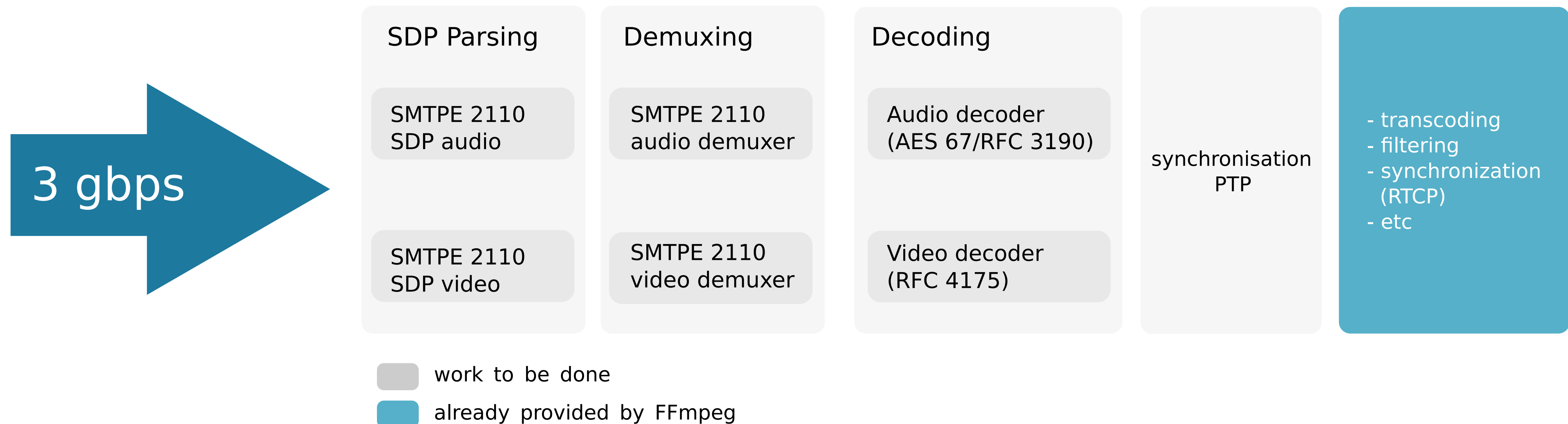
     - Contribute to FFmpeg

2. **Analyze the state of the art**

3. **Figure out what is wrong and what is true**

Our work with Radio Canada

- Doubts on the ability of FFmpeg to handle 3 gbps

- Use of Real Time Protocol (RTP) over udp
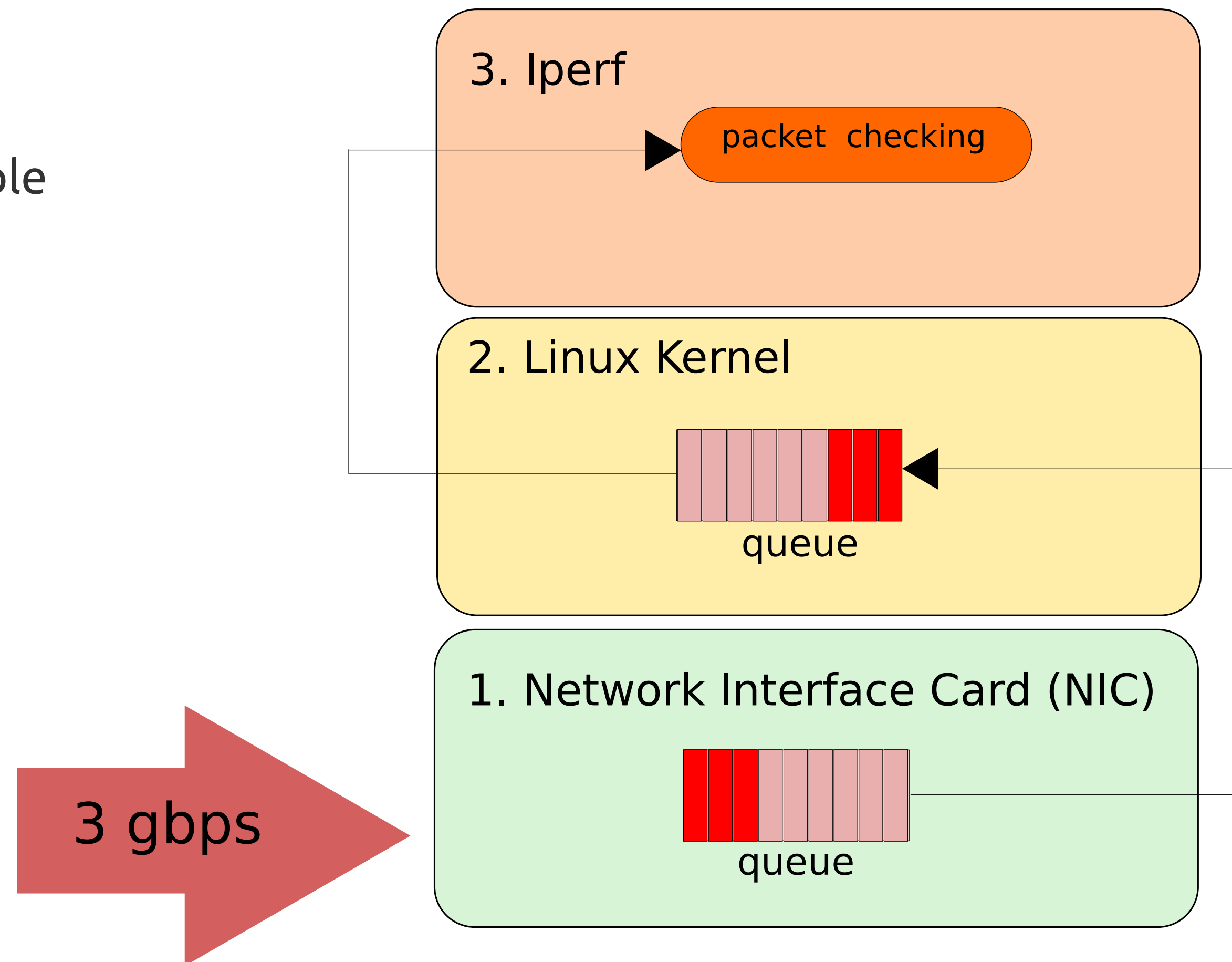
Packets can be dropped !

3 gbps

SDP Parsing	Demuxing	Decoding		
SMTPE 2110 SDP audio	SMTPE 2110 audio demuxer	Audio decoder (AES 67/RFC 3190)	synchronisation PTP	- transcoding - filtering - synchronization (RTCP) - etc
SMTPE 2110 SDP video	SMTPE 2110 video demuxer	Video decoder (RFC 4175)		

work to be done

already provided by FFmpeg

Analysis

- Check first if Linux can handle 3 gbps
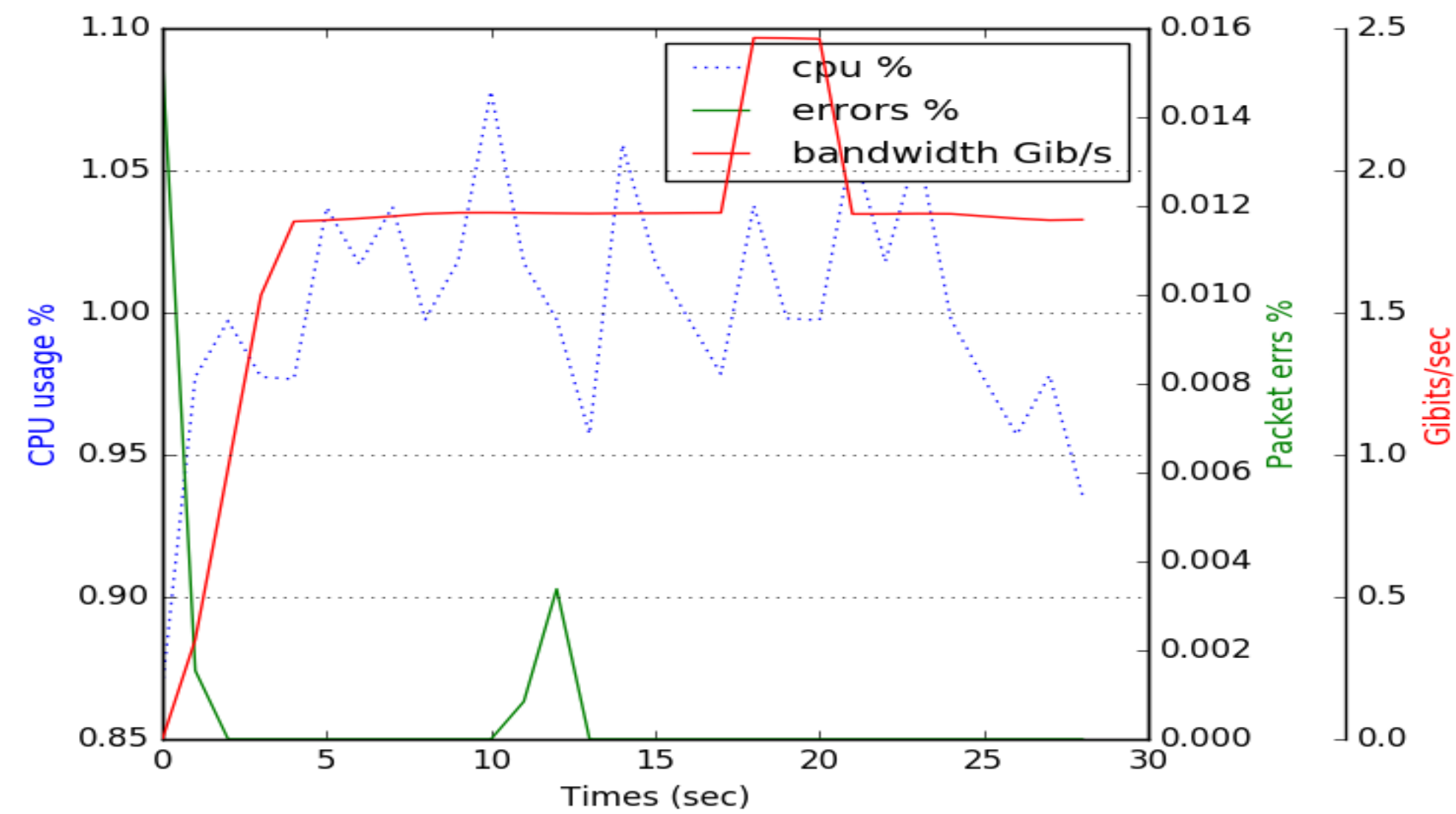
- Use iperf to measure the maximum achievable bandwidth

http://iperf.fr/

Packets can be dropped !

3. Iperf

packet checking

2. Linux Kernel

queue

3 gbps

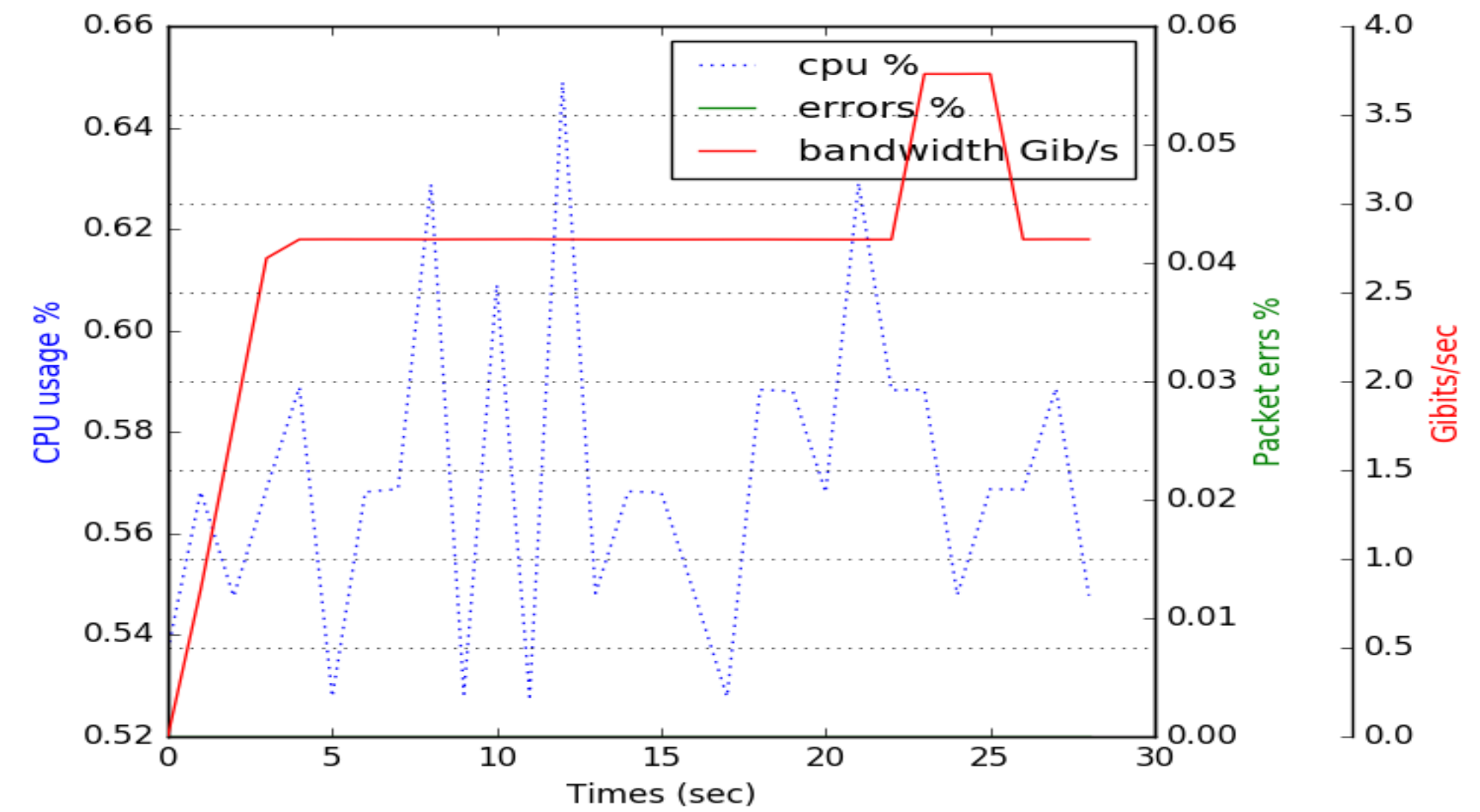**1. Network Interface Card (NIC)**

queue

1. The incidence of UDP Maximal Transmission Unit (MTU)

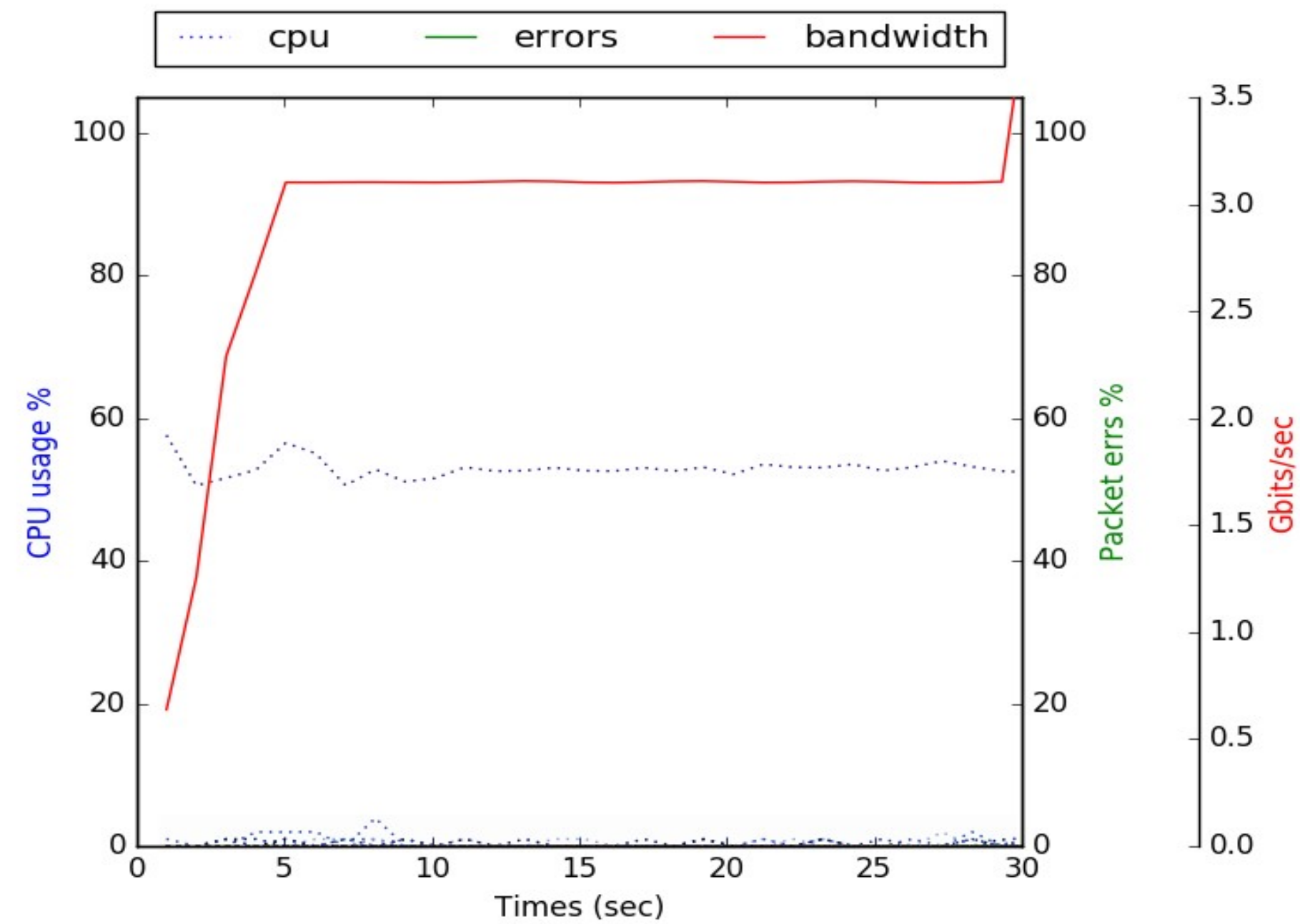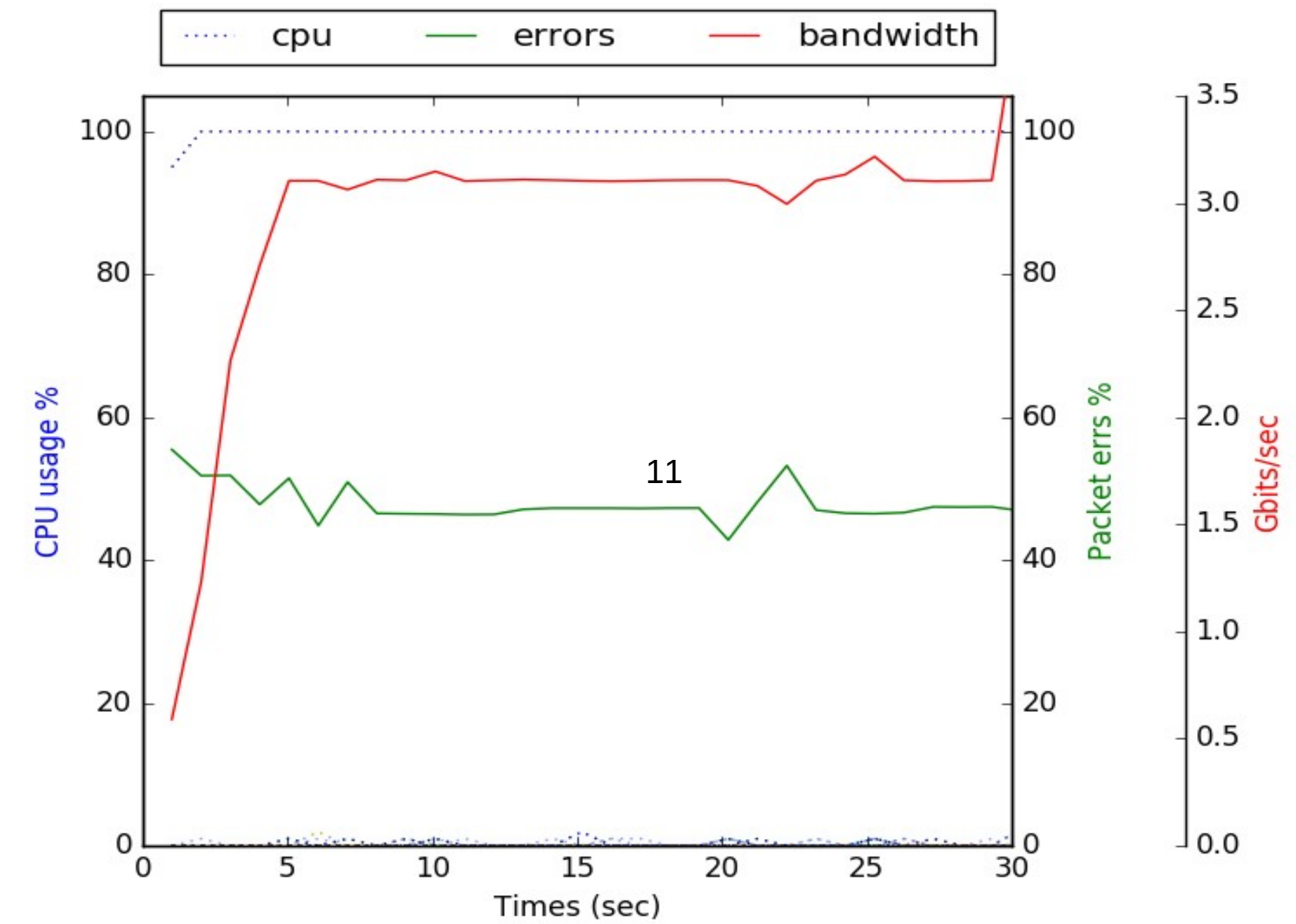**MTU 400**

MTU 1400

2. Influence of processing per packet

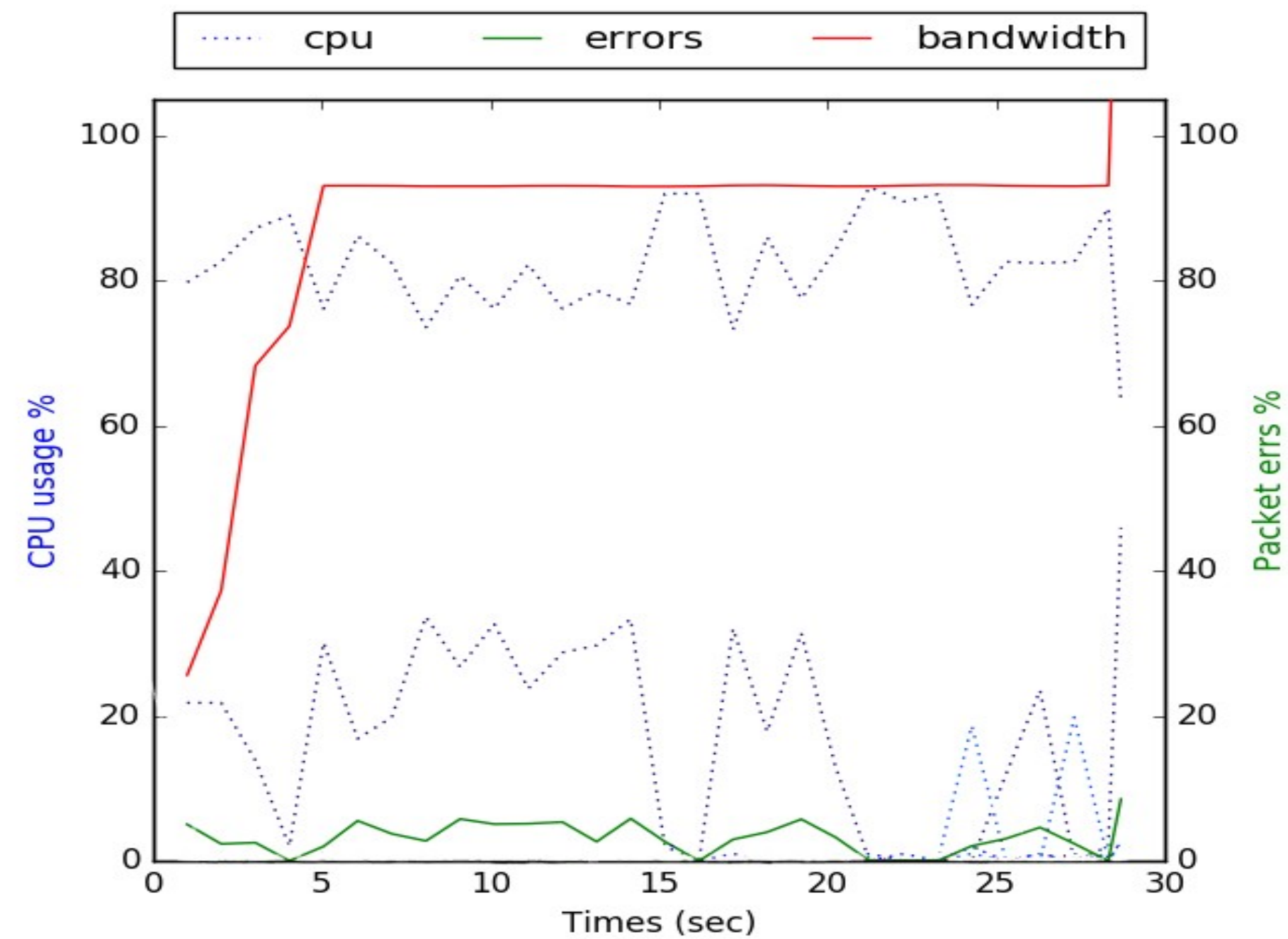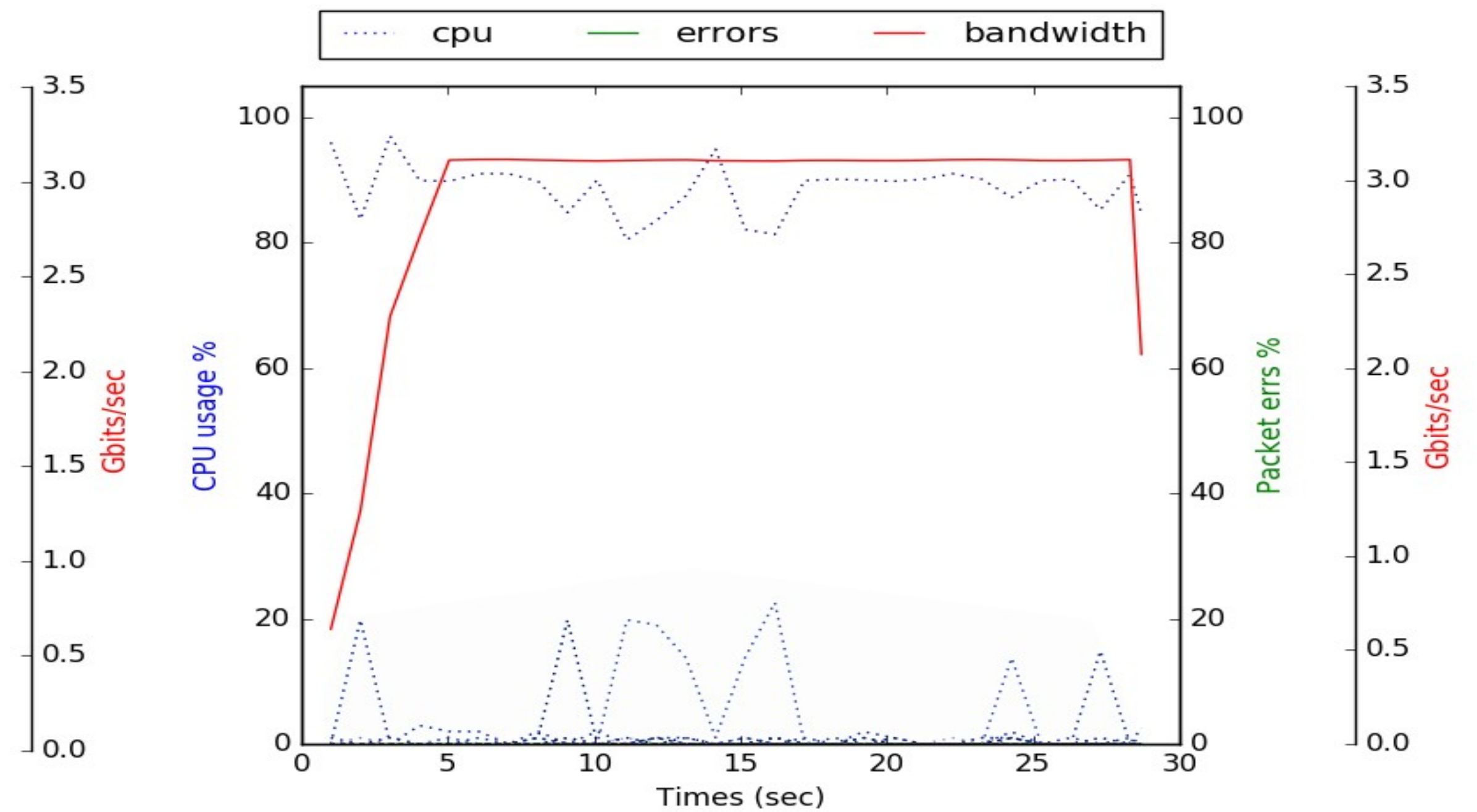## Delay 200

Delay 2000

3. Increase the queue size

Delay=800 – queue=8Mb

Delay=800 - queue=32Mb

Conclusion of analysis

The software should:

1. Deal with maximal MTU (around 1400)

2. Use minimal processing per packet

3. Increase the queues (NIC, kernel)
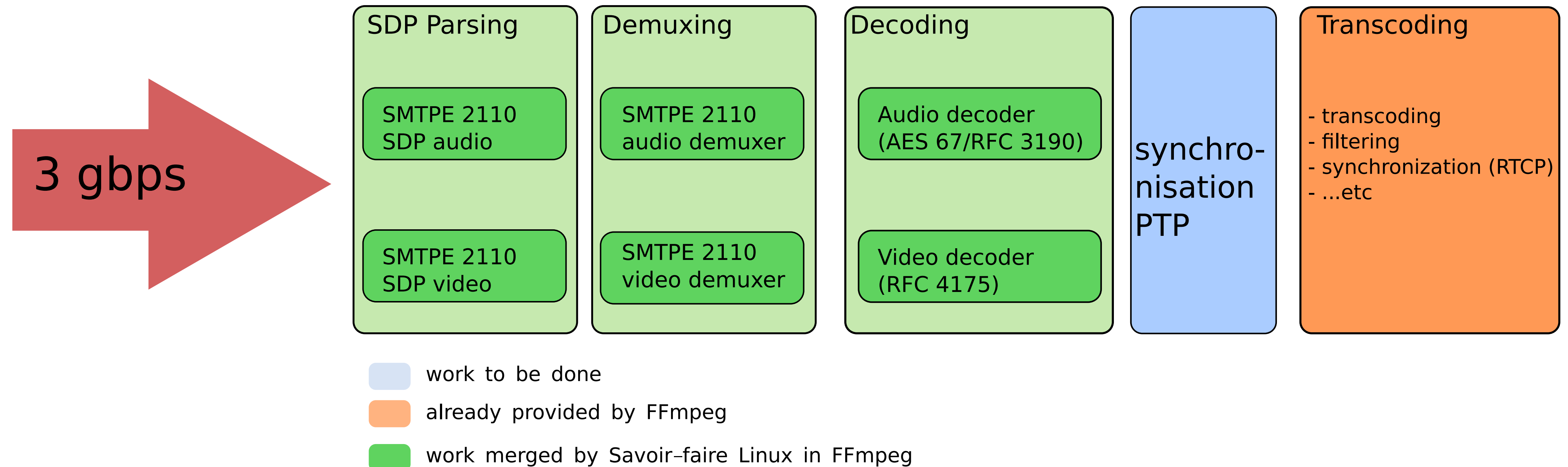
SMTPE 2110 receiver

FFMPEG

- Minimize the processing time per packet removing the jitter buffer

- Ensure to read queues as fast as possible with a dedicated thread

- Size the circular buffer

3. FFmpeg

circular buffer

jitter buffer

2. Linux Kernel

queue

1. Network Interface Card (NIC)

3 gbps

queue
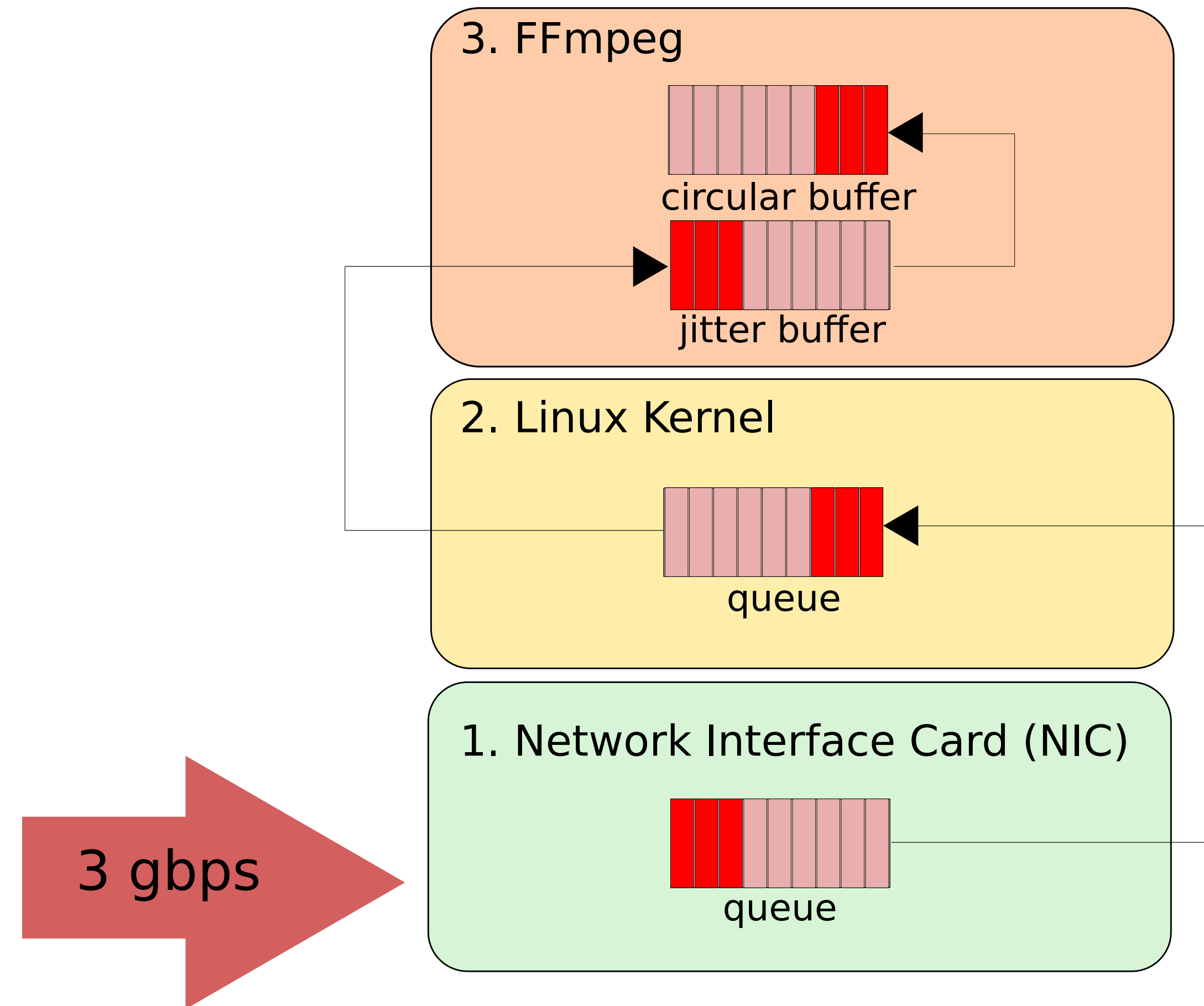
Code merged!

FFMPEG

- Our work merged into FFmpeg on **march 31st 2017**

3 gbps

SDP Parsing	Demuxing	Decoding	synchro-nisation PTP	Transcoding
SMTPE 2110 SDP audio	SMTPE 2110 audio demuxer	Audio decoder (AES 67/RFC 3190)		- transcoding - filtering - synchronization (RTCP) - ...etc
SMTPE 2110 SDP video	SMTPE 2110 video demuxer	Video decoder (RFC 4175)		

work to be done

already provided by FFmpeg

work merged by Savoir-faire Linux in FFmpeg
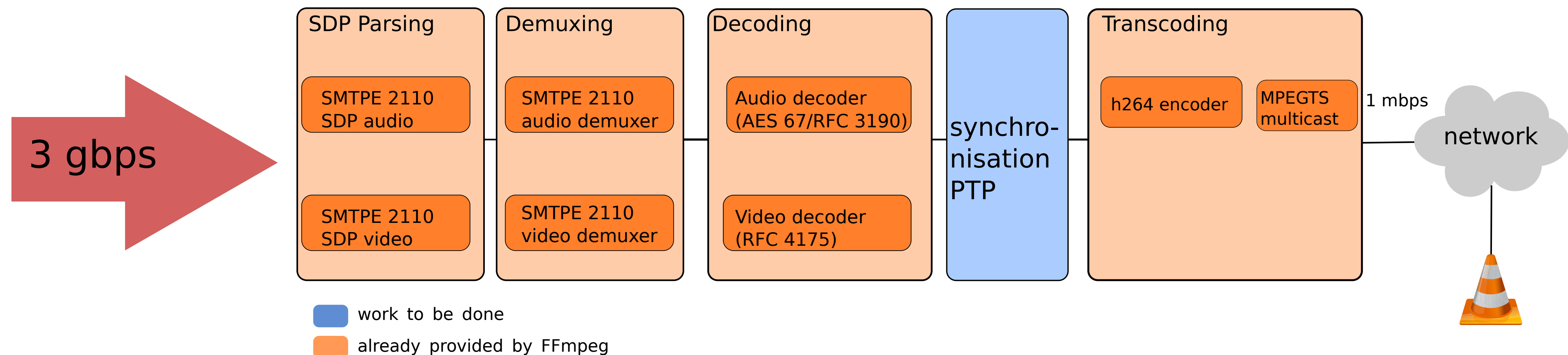
Performance

FFMPEG

- SMTPE 2110 Audio/Video receiver

- **No packet loss**

- **Less than 80% of cpu load**
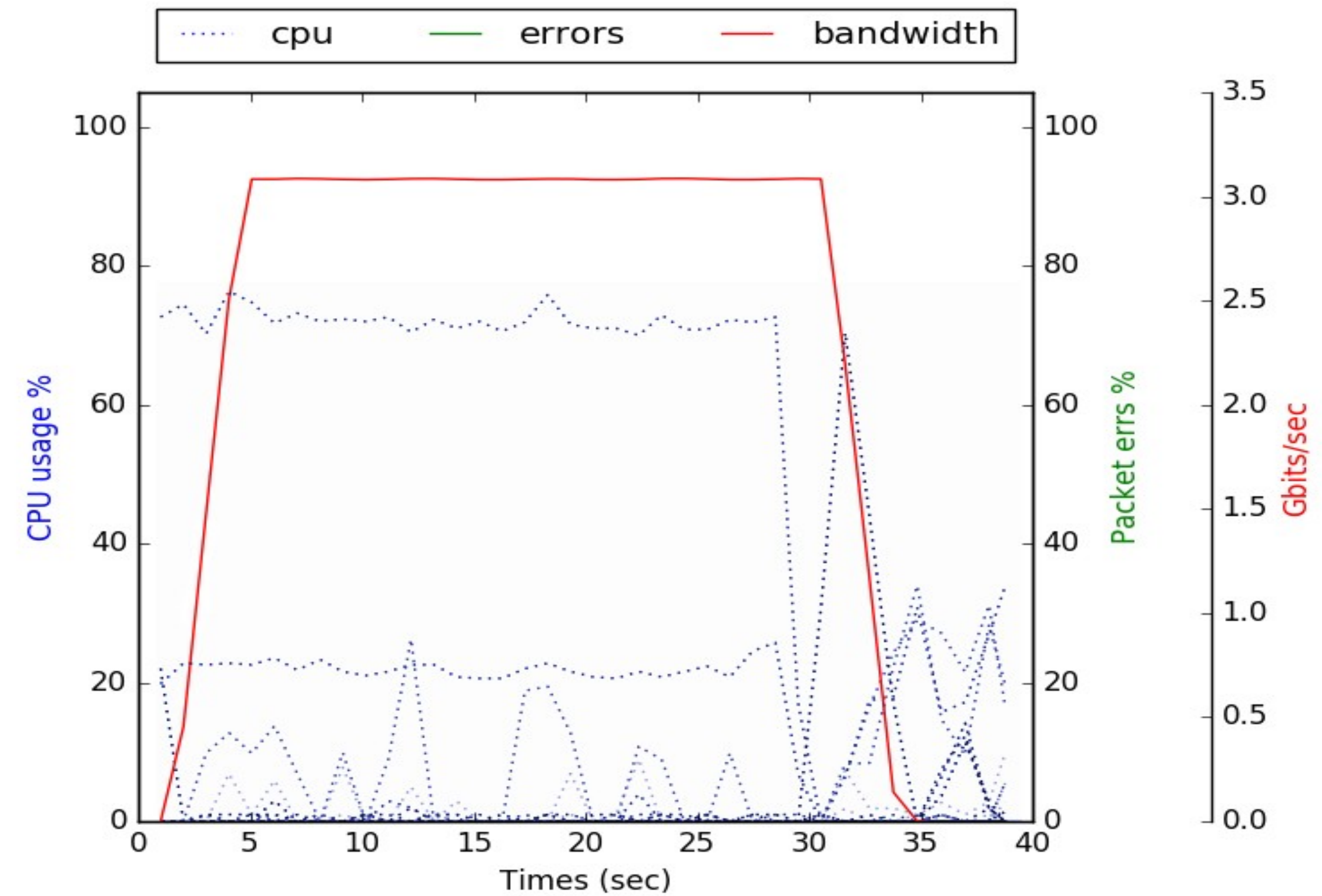
Transcoder example

FFMPEG

- Now use FFmpeg to transcode SMTPE 2110 video

SDP Parsing	Demuxing	Decoding	synchro-nisation PTP	Transcoding
SMTPE 2110 SDP audio	SMTPE 2110 audio demuxer	Audio decoder (AES 67/RFC 3190)		h264 encoder MPEGTS multicast
SMTPE 2110 SDP video	SMTPE 2110 video demuxer	Video decoder (RFC 4175)		

3 gbps

1 mbps

network

■ work to be done
■ already provided by FFmpeg

What is next

- Synchronization over PTP

- Handling of several streams
  - IRQ model
  - Busy polling

- Check latency

The near future

- Full decoding, transcoding, streaming in software
- Cloud based solution scalable for streaming demands

Mutual effort to make this happen !

Open Source Software

The ABCs of Open Source Software

- "can be **freely used, changed, and shared** (in modified or unmodified form) **by anyone**…" (OS Initiative)

- There are **Misconceptions** about it.

- It has **Unique Attributes.**

- It has **Advantages** and **Challenges**

Common Myths About Open Source Software

1. Open Source Software is FREE!

2. Open source is less secure than proprietary software

3. OSS comes without support

4. Open source is harder to maintain

5. Open source is not enterprise-grade
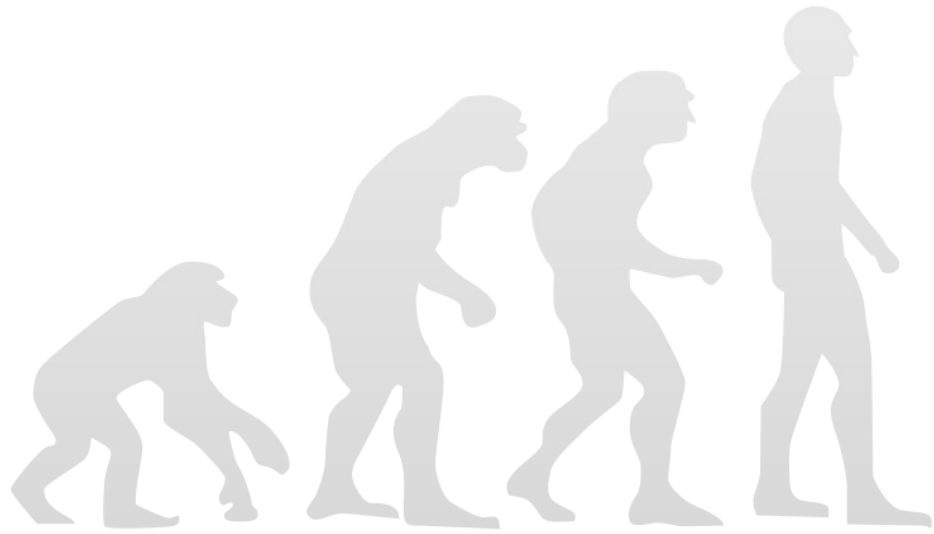
6. OSS is incompatible with proprietary software

https://www.wired.com/insights/2013/03/4-myths-about-open-source-to-put-to-rest/

Open Source Software : A practical case
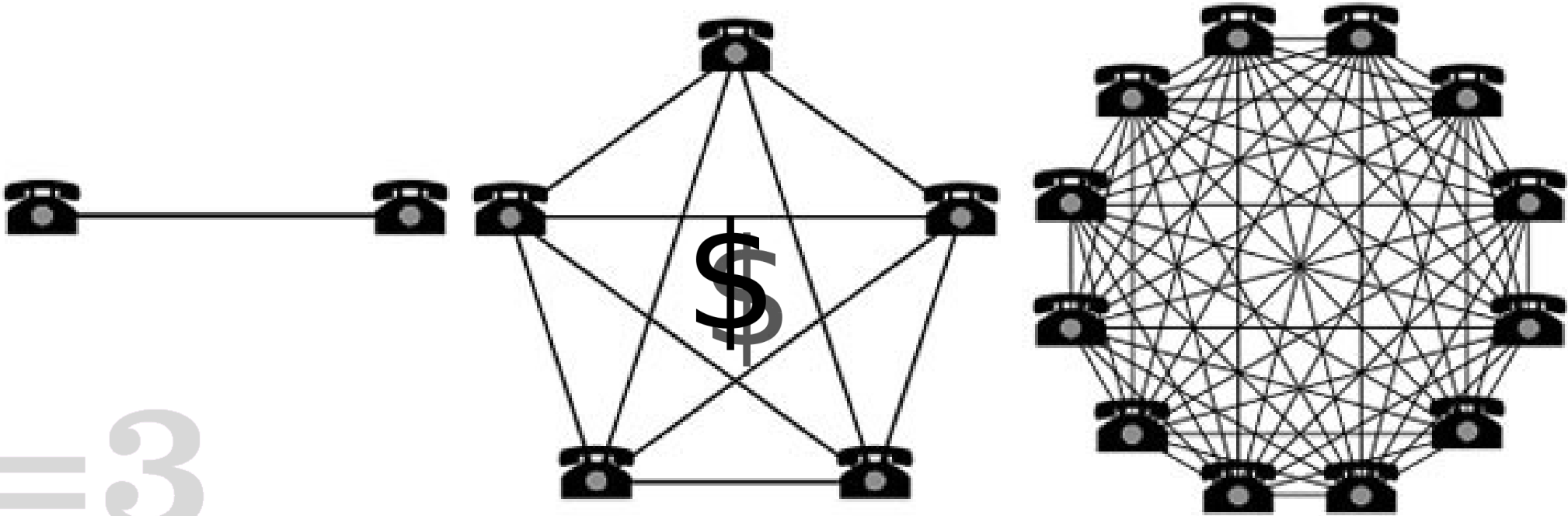
Pragmatic, Optimal, Practical software development

- Expect costs

- Expect technological benefits

- Expect profits

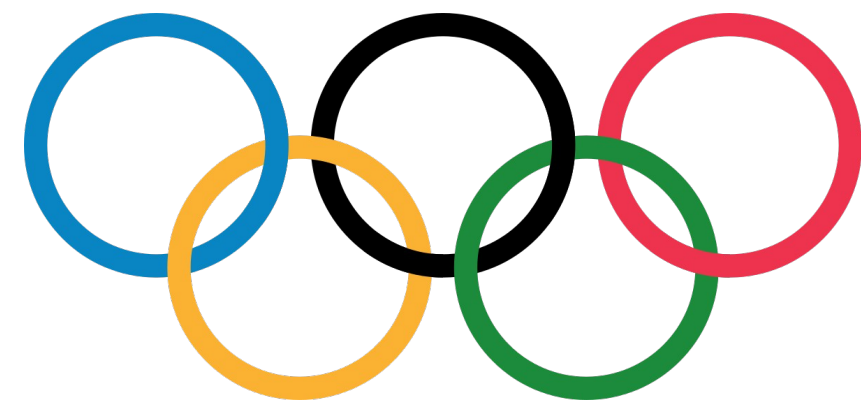# Inherent Attributes Demand Collaborative Development Approach

An Evolutionary Symbiosis: Towards a More Balanced Approach

More Scalability, Flexibility, Openness, and Collaborative Development

—··—

The Key Advantages Of Open Source Software

- Flexibility
- R & D Cost Sharing
- Scalability

"Open source is __not__ dependent on the company or author that originally created it. Even if the company fails, the code continues to __exist__ and be __developed__ by its users."

Main Challenges of Open Source Software

- Not being straight forward to use
- Shortage of apps
- Confusion
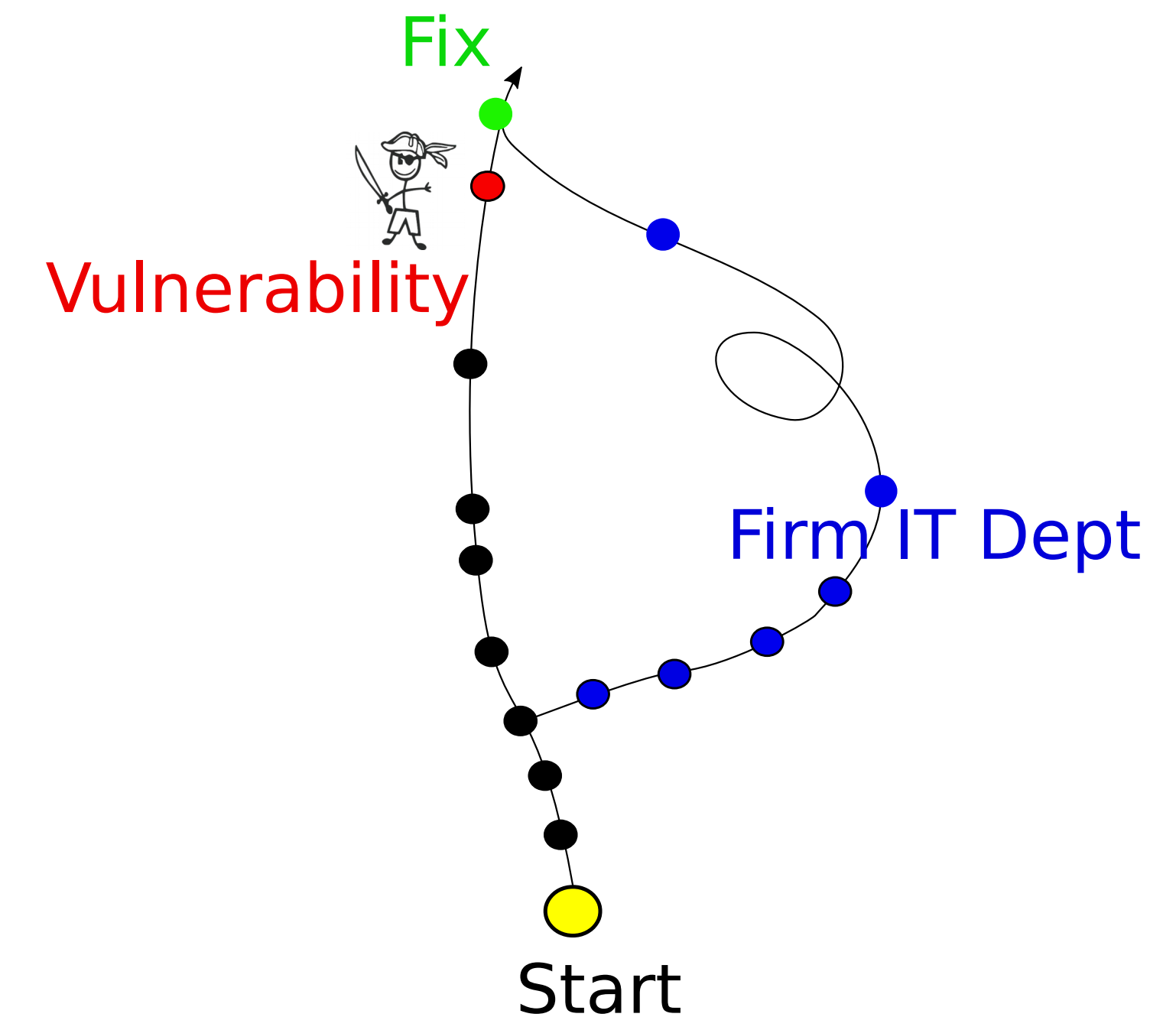- Incompatibility
- User-friendliness
- Indirect costs

逾越節
快樂

Seriously, *Why **INVEST** in Upstream Integration?*

- Maintenance

- Interoperability

- Cost-saving

- Daily Fixes

Upstream Development

Fix

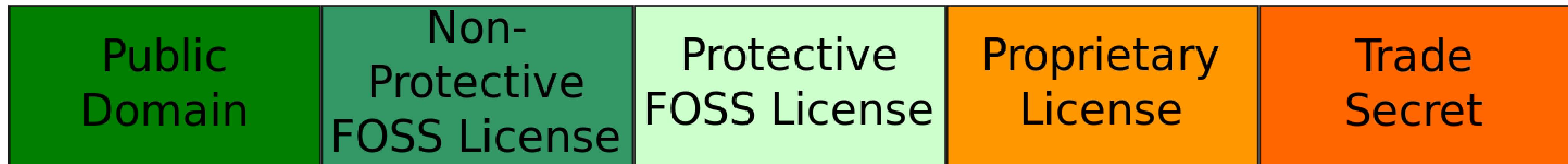Vulnerability

Firm IT Dept

Start

License As an Enabler for Open and Distributed Collaboration

Rights in Copyright

Public Domain	Non-Protective FOSS License	Protective FOSS License	Proprietary License	Trade Secret

All rights relinquished

← more rights granted

more rights retained →

All rights retained

License As an Enabler for Open and Distributed Collaboration

| Copyleft Licenses		Permissive Licenses		Public Domain
Strong	Weak	Less	More	Hakuna matata
GPL	LGPL	Apache 2	MIT, BSDz	
FFmpeg	FFmpeg Mostly	Apache Tomcat	AngularJS, Jenkins	

Copyleft Licensing

- **Freely distribute copies and modified versions** of a work

- **The same rights** must be preserved in derivative works down the line.

- Distribute the software under the same license you received it under.

- Copyleft software licenses are considered *protective* or *reciprocal.*
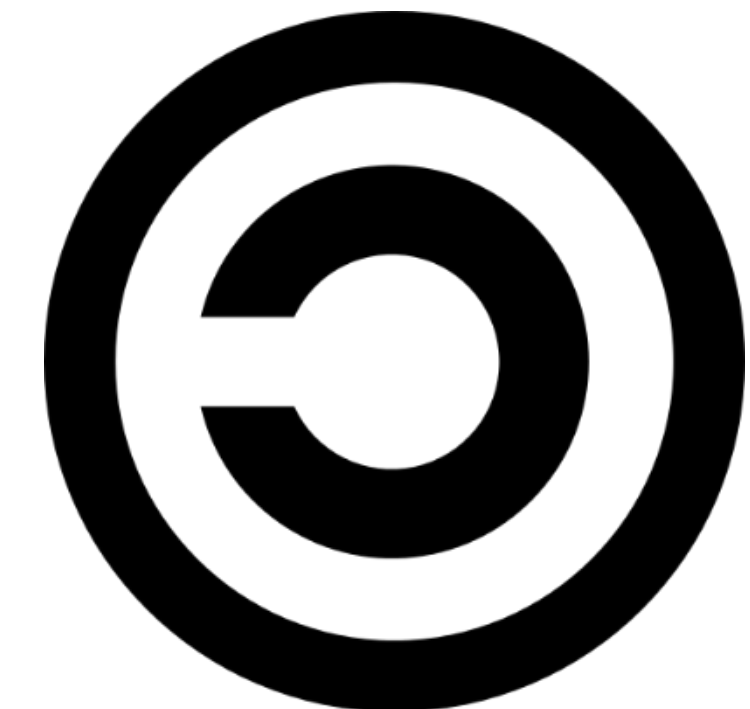
Major Open Source Software Licensing Agreements

- **GNU General Public License (GPL) v3.0**

- **GNU "Lesser" General Public License (LGPL)**

- Apache License 2.0

- MIT license

- Mozilla Public License 2.0

- NASA Open Source Agreement v1.3 (NASA-1.3)
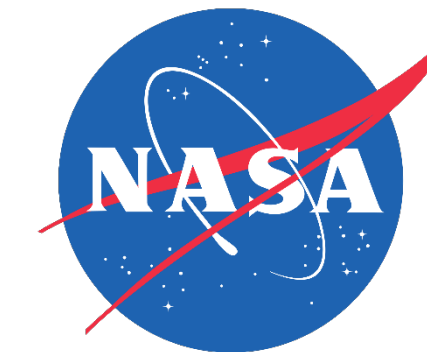
GNU General Public License (GPL)

- Guarantees end users the freedom to run, study, share and modify the software

- derivative work can only be distributed under the same license terms.

- Famous Example: **Linux Kernel**

GNU "Lesser" General Public License (LGPL)

Use & integrate software into your own
(**even proprietary**) software

- No requirement to release the source code of your own components.

- LGPL only requires software be modifiable by end users via source code availability.

- Clear separation between the proprietary and LGPL components.

```c
/*
 * RTP Depacketization of RAW video (TR-03)
 * Copyright (c) 2016 Savoir-faire Linux, Inc
 *
 * This file is part of FFmpeg.
 *
 * FFmpeg is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 2.1 of the License, or (at your option) any later version.
 *
 * FFmpeg is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with FFmpeg; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
 */

/* Development sponsored by CBC/Radio-Canada */

#include "avio_internal.h"
#include "rtpdec_formats.h"
#include "libavutil/avstring.h"
#include "libavutil/pixdesc.h"

struct PayloadContext {
    char *sampling;
    int depth;
    int width;
    int height;

    uint8_t *frame;
    unsigned int frame_size;
    unsigned int pgroup; /* size of the pixel group in bytes */
    unsigned int xinc;

    uint32_t timestamp;
};
```

Licensing Agreements' Implication

What are your goals?

What are your demands?

What will you achieve?

What is your **endgame**?

You have **choices** and **options!**

Licensing Agreements' Implication

Cases	Integration into proprietary libraries and products	Influence technology trajectories	Diffusion and externalities	Trademark concerns	Competition	Patent concerns	Cautious advice
Firm A Tech A	No	Yes	Yes	No	Not relevant	No	GPL
Firm A Tech B	Yes	Yes	No	**Yes**	**Relevant**	Yes	LGPL

License As an Enabler for Open and Distributed Collaboration

Manufactured Product

Community Version

FFmpeg
(LGPL)

Company
Feature
(secret)

FFmpeg
(LGPL)
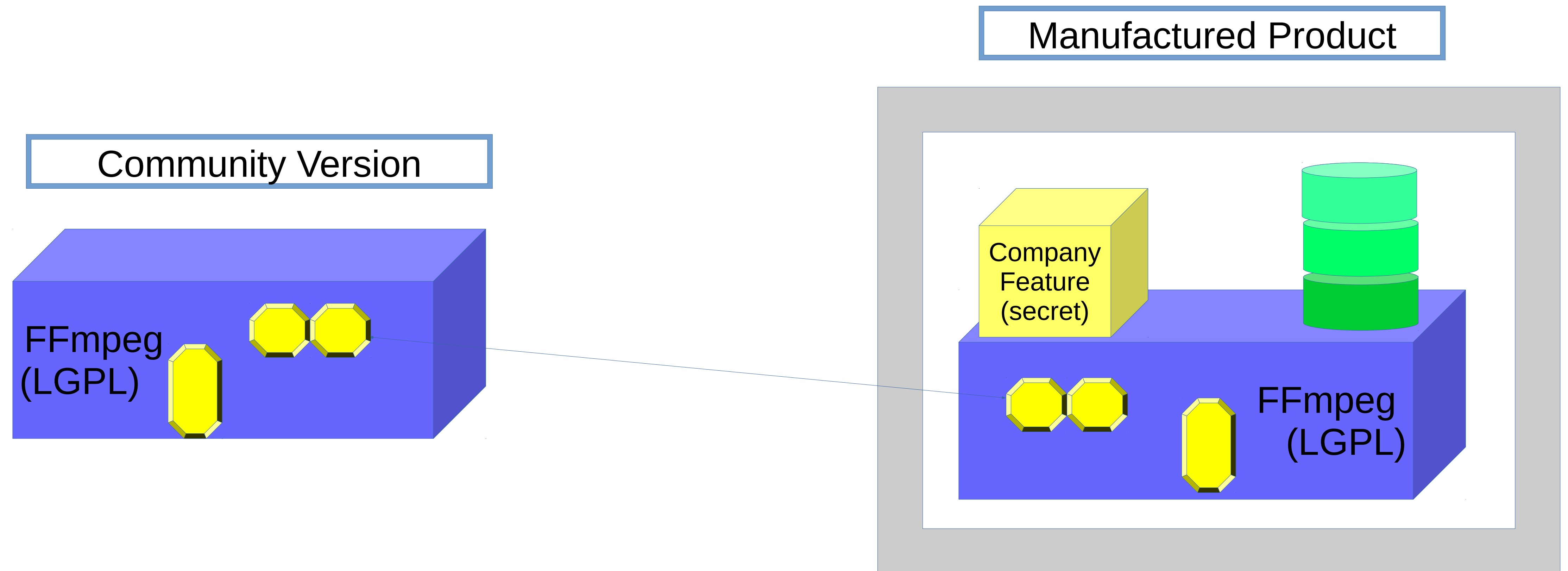
Open Source Software-Based Business Models

Generic Value Creation & Capture Roadmap

Connect

Collaborate

Contribute

Customize

Capture
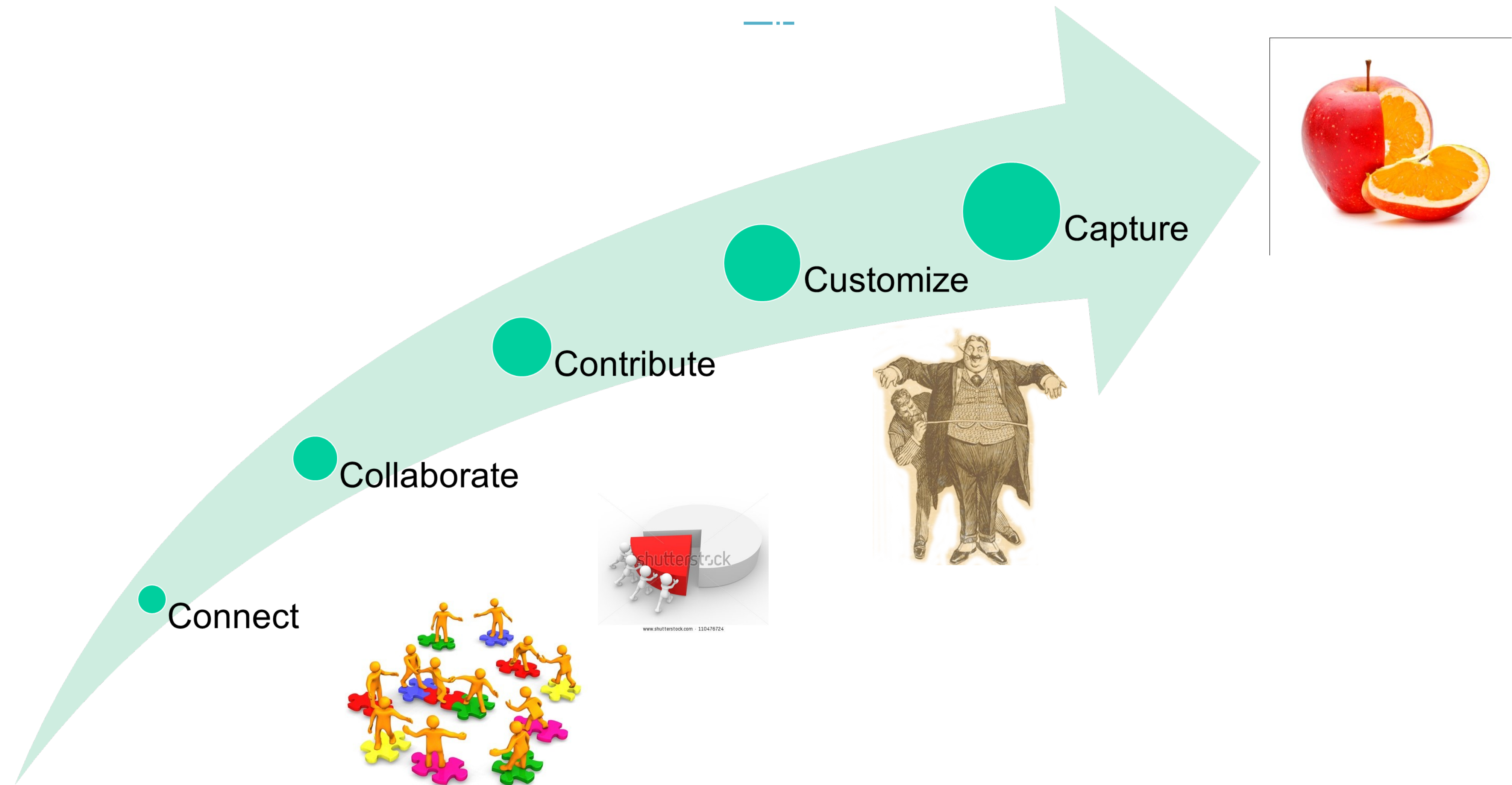
Open Value Creation/Open Business Model in Aerospace Industry

End-user Effect	Higher Value

Multi-Media Equip. in Aircraft

Network Stack of Linux Kernel

Hardware Switches

Hardware Manufacturer Creates Value

Software Consultant Creates Value:
- Embedded in open projects
- In-source required software
- Write missing modules
- Performs integrations
- Performs tests
- Ensures QA
- Creates roadmaps
- Trains

OS Communities Create Value
- Minor assistance: bug fixes
- Major assistance: complex problem solving
- Maintenance and further dev.

Total Value Chain Solution: H-S Complementarity- Higher Demand	Lower Cost	Higher Value

Concluding Remarks

- Every long march starts with the first step! (Lao Tzu, 中國諺語)

- Together, with Radio Canada, we collaborated and made some progress.

- To continue, to remain viable long-term, to accelerate the innovation pace, Radio Canada needs more partners.

- Be Inspired!

- Be Part of this Collaboration.

- Shape your Future.

Digitization Partner

Eloi Bail
Amir Taherizadeh

eloi.bail@savoirfairelinux.com
amir.taherizadeh@savoirfairelinux.com