AN EFFICIENT AND SUSTAINABLE END-TO-END VIDEO STREAMING ARCHITECTURE

P. Angot[1], V. Lepec[1], L. Gregory[2]

[1]Viaccess Orca, France and [2]Ateme, France

ABSTRACT

SMART-CD is a French collaborative project aimed at providing an energy-optimized video streaming solution that allows operators and users to understand and control their environmental impact. By developing clear consumption metrics and optimization technologies for the various steps in the end-to-end distribution chain, the consortium contributes to the greening of video streaming.

This paper will present the context in which the SMART-CD project is situated, the technological challenges to overcome, and the initial results obtained during the first months of joint work. Several concrete optimization strategies currently under development will also be discussed.

INTRODUCTION

A growing awareness of climate change and environmental preservation issues are prompting video service providers to question their energy usage. The digital sector currently accounts for 3% to 4% of global greenhouse gas emissions. This share is expected to grow at a significant annual rate, posing a critical challenge to the sustainability of the video industry. Its major application is video streaming, representing 82% of all consumer IP traffic. The question of how to reduce the carbon impact of video streaming is therefore garnering increased attention from service providers.

Video service providers are rethinking the way they assess and monitor backend infrastructure, combined with accurate client energy consumption, and eventually report the end-to-end platform footprint in an operational usage context. The SMART-CD consortium addresses these challenges in an ecosystem by leveraging multiple heterogeneous stacks, based on state-of-the-art technologies such as 5G ROUTE, mABR, and next-gen video codecs.

This paper will present an approach for developing a sustainable video streaming solution, elaborating on two innovative components making up the solution. First, an agnostic monitoring framework oversees the collection of necessary key indicators (i.e., energy metrics, environmental impact data) from all the components and thereby monitors the environmental footprint of the end-to-end stack. Second, an orchestration agent — which is integrated into the platform’s deployed clusters — dynamically manages platform scale according to a given energy efficiency context.

The paper will conclude by presenting the current progress of the project, the methodology behind SMART-CD's monitoring solution, and the results based on our first experiments, paving the way to the development of a unified orchestration with an environmental focus.

THE SMART-CD PLATFORM ARCHITECTURE

The architecture of the SMART-CD platform has been functionally split into seven major modules. Each of these modules is described in detail below. The following diagram presents a macro view of this architecture.

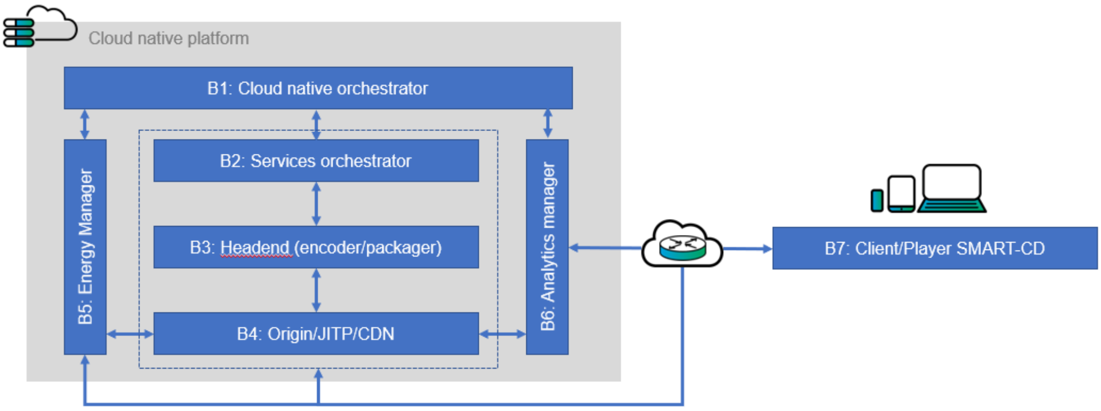

Cloud native platform
B1: Cloud native orchestrator
B2: Services orchestrator
B3: Headend (encoder/packager)
B4: Origin/JITP/CDN
B5: Energy Manager
B6: Analytics manager
B7: Client/Player SMART-CD

Figure 1 : SMART-CD functional components

Components B1 to B6 are key elements of the native cloud platform that will be evaluated during the project:

- B1: The cloud native orchestrator is the brick that dynamically operates the cloud components of the solution deployed as part of the SMART-CD project.
- B2: The service orchestrator is a component supplied by Viaccess-Orca, in charge of managing the video broadcast chain. It is the central element linking components B3, B4, and B7.
- B3: The head-end is made up of Ateme components, which are responsible for content preparation. The encoding and packaging processes must ensure that live TV channels are made available in real time and are therefore relatively resource intensive.
- B4: The CDN provided by Nexedi, coupled to and downstream of the Origin Server, is responsible for distributing live TV channels over the network.
- B5: The Energy Manager, provided by Greenweb, aggregates the energy data available across the entire end-to-end solution for analysis and feeds into component B1.
- B6: The Analytics Manager collects and formats all data (related to energy consumption or otherwise) available on the entire end-to-end solution.

A separate element, Component B7: Client/Player, is the brick available to the end user. The hardware constitution of this component evolves according to the physical location of the end user when consuming live TV channels (mobile vs. in the home). Nevertheless, its software constitution remains identical regardless of the consumption location and includes bricks developed by Motion-Spell, Telecom-Paris and Viaccess-Orca.

The stakeholders in the SMART-CD consortium have specific needs in terms of performance, security, compliance, and cost. Despite this disparate environment, the aim of the project is to bring a higher level of performance optimization to the infrastructures identified and to take advantage of the different cloud computing architectures deployed.

Given the multitude of possibilities for deploying cloud services, AWS appeared to be an appropriate choice for undertaking work on evaluating and optimizing the energy performance of hosted services. The CDN nodes, which are deployed on SlapOS, a French open-source software operating Cloud services, is also part of the end-to-end energy consumption report.

The next sections will present the strategic components in the heart of the SMART-CD platform to manage and optimize its energy consumption.

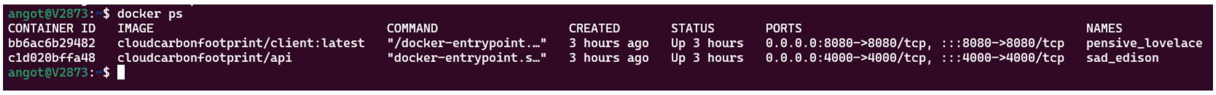
Measurement of Cloud Services Energy Consumption

The back end of the video streaming platform involves numerous pieces of equipment (encoder, packager, service platform, DRM system, origin server, CDN, etc.). This equipment can be distributed and hosted in different ways. Some may be in on-premise servers, others on a private cloud or finally deployed on a public cloud. In the first two cases (on premise or private cloud), we can have control over the resources used by the components to evaluate their energy consumption with various degrees of ease. It's not nearly as simple for components hosted on a public cloud, for various reasons. The first is that public cloud solution providers define an abstraction layer between the logical resource and the physical resource that does not allow direct access to the physical resource. The second reason is that, even if we had access to the consumption data of the physical resource, we are unable to ensure that our component is the only one to request the physical resource throughout the energy consumption measurement.

In this part of the article, we are exclusively interested in this challenge of energy measurement in a public cloud. We target the three main public cloud providers: Amazon, Google and Microsoft. As service providers, the members of the Smart-CD consortium are likely to deploy their solutions on the infrastructure of any cloud provider. Therefore, it is essential to be able to use a single measurement solution regardless of the host (AWS, GCP, Azure). The objective of the Smart-CD project is, in particular, to evaluate the open-source Cloud Carbon Footprint (CCF) solution (https://www.cloudcarbonfootprint.org/).

CCF enables organizations to measure, monitor, and reduce their public cloud carbon emissions. As it supports multiple public cloud providers including AWS, GCP, and Azure, it’s a relevant candidate for a deep analysis all along the Smart-CD project. This evaluation will be carried out throughout the duration of the collaborative project. The first use case as part of the Smart-CD project was the assessment of CCF on Viaccess-Orca components, the Service Delivery Platform (SDP) and the Monitoring as a Service (MaaS), both deployed on AWS.

The required AWS configuration to set up is detailed on the following page (https://www.cloudcarbonfootprint.org/docs/aws). We need at least an AWS account with the correct permissions, the Cost and Usage Billing AWS feature and an Athena DB. In a first step, CCF has been manually deployed on two local environments, macOS and Linux Ubuntu. The CCF application is divided into two components. The first one, API, oversees communication with the cloud provider — AWS in our case. The second one, Client, provides the web interface for displaying the environmental metrics in a predefined dashboard. Both components are run as a Docker container on macOS and Ubuntu and need a correct environmental configuration to integrate each over and with AWS.



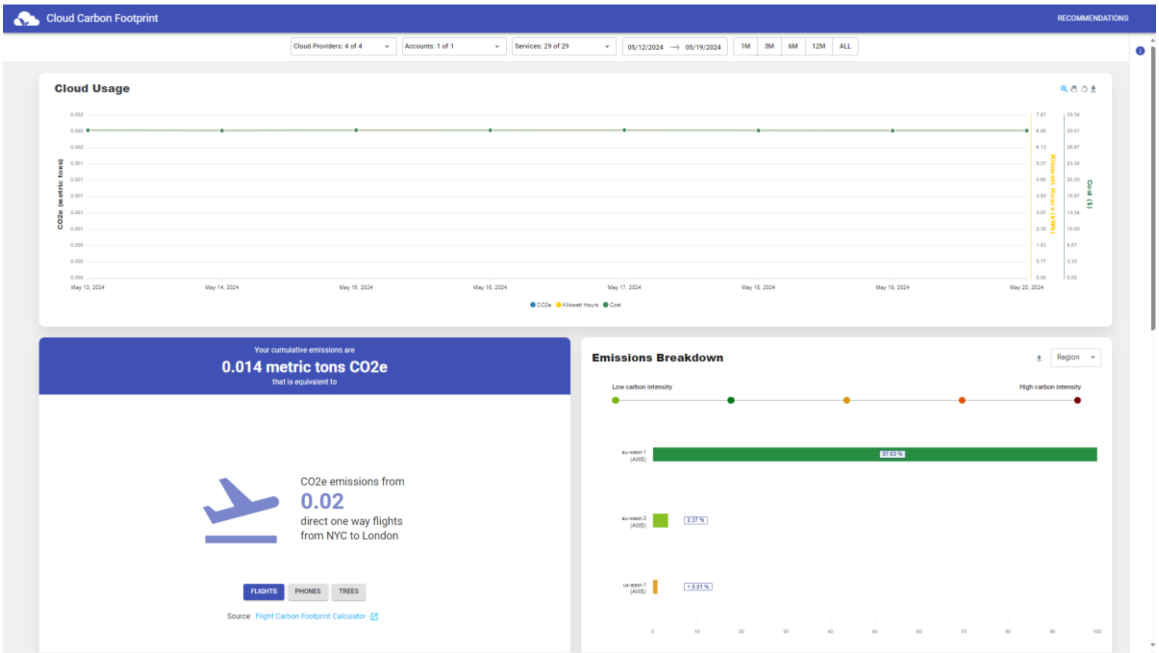
```
angot@V2873:~$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED         STATUS         PORTS                                       NAMES
bb6ac6b29482   cloudcarbonfootprint/client:latest   "/docker-entrypoint.…"   3 hours ago     Up 3 hours     0.0.0.0:8080->8080/tcp, :::8080->8080/tcp   pensive_lovelace
c1d020bffa48   cloudcarbonfootprint/api             "docker-entrypoint.s…"   3 hours ago     Up 3 hours     0.0.0.0:4000->4000/tcp, :::4000->4000/tcp   sad_edison
angot@V2873:~$ 
```

Figure 2 : API and Client docker containers

In the next stages of the project, our ambition is to exploit the data received from AWS in the most precise and personalized way possible. Consequently, we'll be concentrating on using the endpoints provided by the API, i.e.:

- */footprint* provides calculated energy and carbon depending on given parameters.
- */regions/emissions-factors* provides the carbon intensity (CO2e/kWh) of all cloud provider regions.
- */recommendations* provides recommendations from cloud providers and their estimated carbon and energy impact.

This should enable us to establish the energy consumption of each software component deployed on AWS as part of the Smart-CD project, with granularity down to the AWS service level. Currently, the end-to-end platform deployed for the project's testing and integration needs is not being used to the same scale as in production. For this reason, we have observed a linear energy consumption of the SDP and MaaS components on the CCF dashboard below.



Cloud Providers: 4 of 4

Accounts: 1 of 1

Services: 29 of 29

05/12/2024 → 05/19/2024

Your cumulative emissions are
0.014 metric tons CO2e
that is equivalent to

CO2e emissions from
0.02
direct one way flights
from NYC to London

Source: Flight Carbon Footprint Calculator

Figure 3: Cloud Carbon Footprint dashboard

During the week of May 13-20 the MaaS component was used on May 15-16 to test the energy consumption of the devices (S10, S21 and NVIDIA Shield) with different video contents (see elsewhere). Despite the number of tests carried out, they were not large enough to make significant demands on MaaS; thus, this component did not show tangible results in terms of energy consumption over the two days of testing compared with the rest of the week.

This leads us, in the next steps of the project, to consider simulating the provisioning and use of our SDP and MaaS components at scale by using a tool like Apache JMeter, for example. As you can see, we are only at the very beginning of CCF assessment and its possibilities. As well as testing it in GCP and Azure environments, we need to be able to aggregate the data it sends us as finely as possible and, above all, trigger platform orchestration actions based on the data it sends us. These actions could take place both at the level of the hosted business components:

- Increasing/decreasing, in real time, the number of instances of a component
- Moving a component to the most frugal geographical region in terms of energy resource consumption at a given time of the day, week or year

But also, at the level of the cloud provider's services:

- Rightsizing (upsizing/downsizing/terminating) resources
- Optimizing computing resources

These actions will undoubtedly enrich the orchestration component of the end-to-end platform and perhaps call into question the architecture built at the start of the project. In any case, the investigations carried out up to the end of this project may enrich this article or contribute to another tech paper with new and representative elements to present.

FIRST EXPERIMENTATIONS AND RESULTS

Development of a Measurement Protocol

The previously introduced tool, CCF, allows for measuring the impact of services deployed on a public cloud. Its integration is currently underway to monitor the consumption of various components (encoder, packager, service platform, etc.), and it will be applicable during large-scale experiments.

Initial measurements were conducted on a small scale to validate our approach and confirm — or refute — several of our initial hypotheses. The objective of this initial measurement campaign is to model the electrical consumption of an end-to-end video streaming solution to quantify the impact of choosing one video codec over another.

The initial assumption is as follows: the latest generation of video codec, VVC, offers significantly superior performance as compared to its predecessors, particularly AVC, which is currently widely deployed and used. As explained in Jankovic and Keilbach (1), this improved coding efficiency comes with a substantial increase in complexity and, consequently, an increase of the cost of encoding and decoding operations. Therefore, the question is: considering the total consumption of the streaming chain (from encoding to decoding), which codec offers the best energy performance?

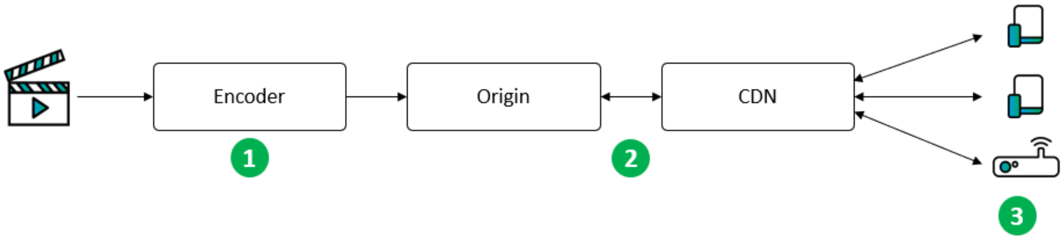


Figure 4: High level view of the end-to-end chain.

This initial experimental protocol comprises the following steps:

1. **Encoding**: Measuring the electrical consumption induced by the encoding process (CPU/RAM) during live video encoding phases.
2. **Distribution**: Estimating the electrical consumption induced by the distribution of the encoded content from step (1).
3. **Decoding**: Measuring the electrical consumption of the CPU during the decoding phase on various devices.

For each step, the employed method varies depending on resource access and available tools.

Measuring the Consumption of the Encoding Step

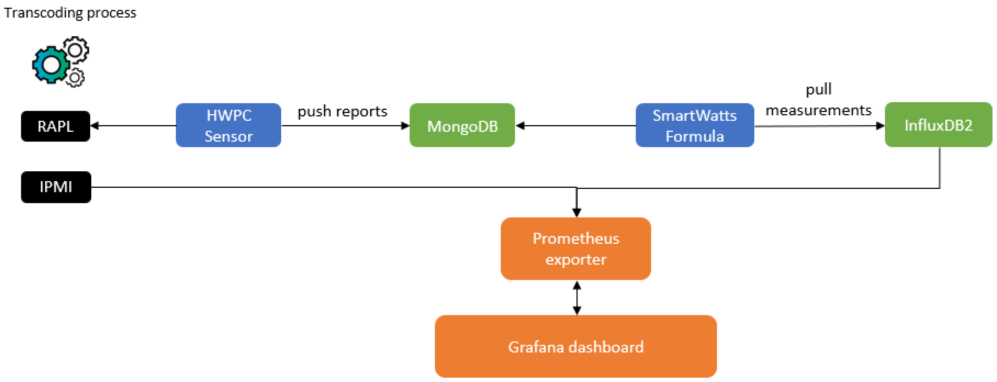


Figure 5 : Services and tools deployed to monitor the resources usage and resulting power consumption during the transcoding process.

This measurement is performed on an Intel Xeon 6150 platform using the *HWPC sensor* and the *SmartWatts formula* from the Power API project, Bourdon and Bordage (2). This tool allows users to monitor the full CPU and RAM usage or a subset of processes using *cgroups*. In our case, we only measured the resource usage by the transcoding process, allowing us to isolate the encoder's consumption from the other operations executed on the server.

Codec	Resolution	BitRate (kbps)
AVC	640x360	365
	960x540	2000
	1280x720	4500
	1920x1080	7800
HEVC	640x360	145
	960x540	1600
	1280x720	3400
	1920x1080	5800
VVC	640x360	110
	960x540	1200
	1280x720	2500
	1920x1080	4350

A basic test vector was prepared to collect data sequentially to compare the transcoder's consumption during the live encoding phases of a sequence in AVC, HEVC and VVC.

The service configuration is based on Apple’s HTTP Live Streaming (HLS) authoring specification for Apple devices for selecting the bitrates used in AVC and HEVC. An extrapolation, considering the current efficiency of the VVC implementation, allowed for defining a coherent range of bitrates for this codec as well.

Modelling the Consumption of the Distribution Step

Measuring the electrical consumption induced by the distribution of the content is challenging. Between the publication of the content on the origin server and its final download by the end device, the data traverses a network comprising multiple types of equipment to which we do not have access.

We therefore relied on the work of Malmodin, J (3) to estimate this electrical consumption, specifically utilizing the two formulas applied for distribution over fixed networks (1) and mobile networks (2):

$$P_{network} = \frac{18\,W}{Users} + \frac{0.05\ {}^{W}\!/_{Mbps}}{DataUsers} \tag{1}$$

$$P_{network} = 1\,W + 1.5\,{}^{W}\!/_{Mbps} + 0.2\,W + 0.03\ {}^{W}\!/_{Mbps} \tag{2}$$

Where:

- *Users*: number of users per households
- *DataUsers*: number of data service users per households, *e.g two users watching Netflix in a household of four*

In addition to the energy cost associated with the distribution of content over the network, we need to consider the cost of storing the content during this phase. Content distribution relies on caching within the CDN to facilitate distribution to many clients, and it involves storing the content for varying durations. This is referred to as a *retention strategy* or *retention period*, which denotes the duration for which a video segment is retained by the CDN.

For this, we rely on the method proposed by CCF and apply the following factors: 0.65 W per terabyte for HDD storage, 1.2 W per terabyte for SSD storage.

Measuring the Consumption of the Decoding Step

The Player developed by Viaccess-Orca integrates an experimental energy consumption measuring probe, which is at this stage implemented for Android platforms. In fact, it makes use of Android file system /proc System API, which extracts and computes energy metrics related to the device processing unit performing the decoding of the video streams. To build a consistent energy consumption model, multiple devices showing heterogeneous hardware capabilities have been tested in this experimentation:

Device	Release date	Chipset	Battery
Samsung Galaxy S10	March 2019	Exynos 9820 8 cores 2.7 GHz	3400 mAh
Samsung Galaxy S21	January 2021	Exynos 2100 8 cores 2.9 GHz	4000 mAh
Nvidia Shield	October 2019	Tegra X1 4 cores	/

Table 1 : Characteristics of the devices used in this experimentation.

All three devices include hardware optimized decoding for AVC and HEVC. The open-source decoder OpenVVC was integrated into Viaccess Orca's player to allow software decoding of the VVC encoded contents.

The contents used to measure the consumption of the player are the contents transcoded and DASH packaged during the first step of this protocol.

Several scenarios have been defined and automated to measure the CPU energy consumption during the decoding phases: using a multi-representation DASH content, using a single-representation (1080p25) DASH content, using or not using the hardware optimizations to decode AVC and HEVC, etc.

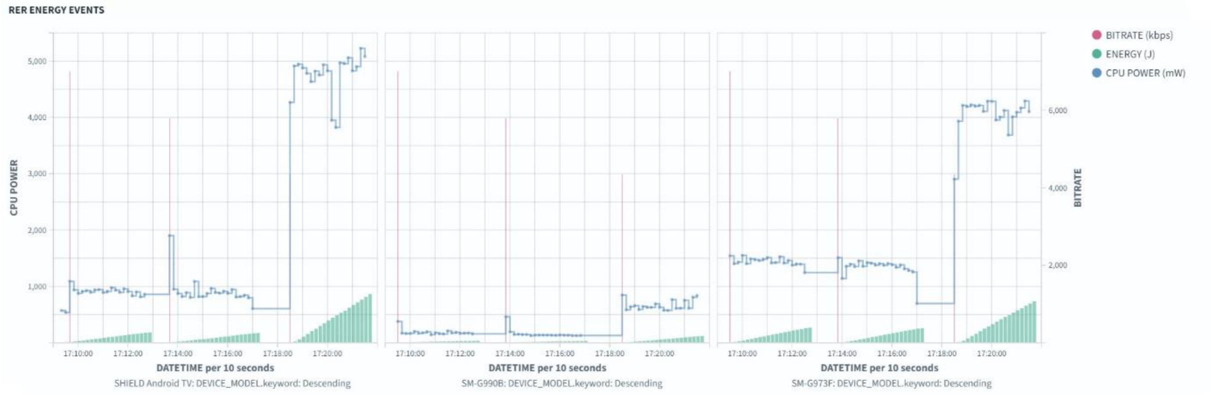
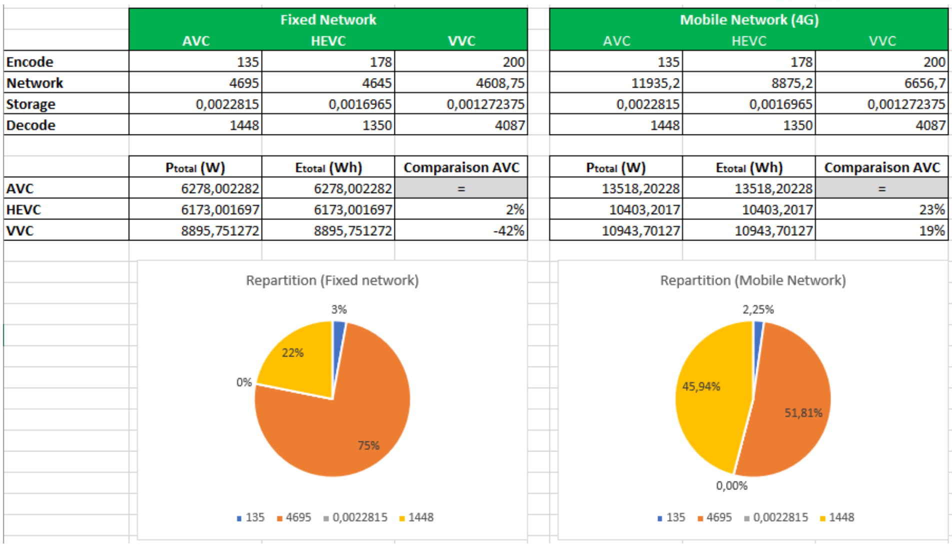


Figure 6 : From left to right: the S10, the S21, and the Shield. This graph highlights the increase in energy consumption induced by VVC content decoding (on the right of each sub-plot) compared to AVC and HEVC (respectively, on the left and in the center) on each end-device.

Aggregation of Measurements and Estimations Across the End-to-End Chain

The results of the measurement campaigns and the previous modeling efforts have been compiled into a single document to derive the estimation of end-to-end electrical consumption for the chain.

$$P_{global} = P_{encoder} + P_{network} + P_{storage} + P_{decoder}$$



	Fixed Network			Mobile Network (4G)		
	AVC	HEVC	VVC	AVC	HEVC	VVC
Encode	135	178	200	135	178	200
Network	4695	4645	4608,75	11935,2	8875,2	6656,7
Storage	0,0022815	0,0016965	0,001272375	0,0022815	0,0016965	0,001272375
Decode	1448	1350	4087	1448	1350	4087

	Ptotal (W)	Etotal (Wh)	Comparaison AVC	Ptotal (W)	Etotal (Wh)	Comparaison AVC
AVC	6278,002282	6278,002282	=	13518,20228	13518,20228	=
HEVC	6173,001697	6173,001697	2%	10403,2017	10403,2017	23%
VVC	8895,751272	8895,751272	-42%	10943,70127	10943,70127	19%

Figure 7 : Depending on the parameters, the overall costs are not the same. In this case, choosing VVC over AVC only makes sense when the content is delivered over a mobile network (+19% energy efficiency).

These initial findings lead to the following conclusions:

- The choice of video codec employed has a significant impact on the overall electrical consumption.
- The components (encoding, network, storage, and decoding) vary in importance within the formula depending on the parameters.
- In all cases, storage is negligible.
- In all cases, distribution is the most significant component, closely followed by decoding.
- Depending on the selected parameters, it may be preferable to favor the distribution of content encoded in AVC, HEVC or VVC. This validates our initial hypothesis that a dynamic choice should be made based on the composition of the "audience" pool.

This protocol and the initial results it has provided, serve as the foundation for the joint monitoring component and allow us to begin the design of the orchestration component of SMART-CD. The following scenarios are being considered:

Audience-aware streaming

Based on the metrics retrieved at each level of the end-to-end chain, and more particularly from the end-devices, the encoder configuration is dynamically updated via instructions sent by the SMART-CD orchestrator: removal of underutilized or deemed too costly representations, or addition of representations better suited to the viewing conditions of a segment of the audience.

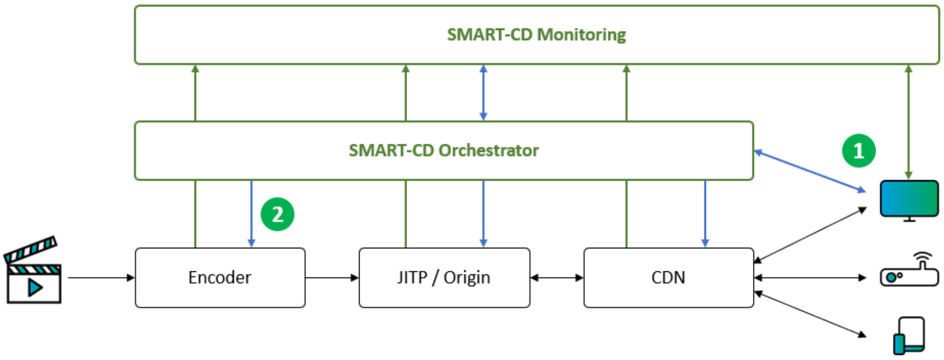


Figure 8 : Interactions between the streaming solution and the SMART-CD components.

Example: The information collected in (1) indicates that many users are consuming their content via a mobile network. Therefore, it is relevant to add a VVC representation to address these users, thereby reducing the overall energy consumption (2).

Networks hybridization and leveraging 5G mABR

In this case, the information collected in (1) indicates that several clients are consuming the stream within a very restricted geographical area. Therefore, it is relevant to switch from unicast (one session per client) to multicast by initiating a ROUTE session via a 5G mABR link, Angot et al (4).]

Dynamic resources allocation

As our measurement protocol has demonstrated, the content distribution phase significantly impacts the overall energy cost. Reducing the encoding bitrate of a stream (while maintaining the same quality) will therefore have a substantial impact on this cost, even though it will increase the encoding complexity. The popularity of content, known to the service platform, can be quantified and used by the SMART-CD orchestrator to dynamically modify the allocation of resources (CPU, RAM) at the encoder level: highly popular content is allocated more resources, making its encoding more efficient and allowing the encoding bitrate to be reduced. This, in turn, significantly impacts the volume of distributed traffic and thus the induced energy cost. This approach may seem counterintuitive, but convincing results have already been provided by Moussaoui et al (5).

CONCLUSIONS

This paper presented the SMART-CD collaborative project and its current progress, with encouraging results regarding the measurement of the environmental impact of video streaming. The measurement protocol presented appears relevant and will serve as the foundation for the design of the SMART-CD solution and the implementation of the presented use cases.

These conclusions are accompanied by the following reservations:

1. The panel of tested devices is not representative of the actual user base: the two models (Samsung Galaxy S10 and Samsung Galaxy S21) were released in 2019 and 2021. Furthermore, both are high-end smartphones, yet we observe significantly higher electrical consumption (5 to 10 times higher depending on the codec) on the S10.

2. This initial experimental protocol only considers the operational phase of the different components of the chain rather than their entire life cycle. The environmental cost of manufacturing and recycling all the pieces of equipment used in the end-to-end chain significantly influences the calculation of the solution’s carbon footprint.

3. Corollary: the current strategy — for most operators — leans towards frequent renewal of set-top box fleets, most of the time for technological reasons: performance improvements, new features, etc. Nowadays, environmental reasons are also cited: use of recycled materials, energy-efficient consumption modes, etc. However, quantifying the environmental impact of such frequent renewal is not straightforward.

4. Regarding the end-devices, only CPU consumption is measured in the current version of our protocol. It would be necessary to add the consumption of the network chip and the 4G/5G modem. Moreover, an additional section measuring screen consumption is highly relevant given the average power of these components (ranging from a few watts for smartphones to over a hundred watts for 4K screens).

Future versions of the protocol will aim to correct the approximations resulting from the points listed above, to augment the existing database, and to automate the measurement process for subsequent utilization of the results in the project's continuation.

REFERENCES

1. Jankovic, M. and Keilbach, J. 2023. Streaming against the Environment: Digital Infrastructures, Video Compression, and the Environmental Footprint of Video Streaming.
2. Bourdon, A. and Bordage, F. 2013. PowerAPI : mesurer la consommation électrique des logiciels sans wattmètre. GreenIT.fr, 2013. ffhal-00783426
3. Malmodin, J. Ericsson Research, Ericsson AB 2020. The power consumption of mobile and fixed network data services - The case of streaming video and downloading large files. Electronics Goes Green 2020.
4. Angot, P., Lepec, V., Nochimowski, A., Gonon, P., Thienot, C. and Vargas-Rubio, J.C., 2022, Taking steps toward greener streaming.
5. Moussaoui, A., Raulet, M., Guionnet, T. 2022. Dynamic Seamless Resource Allocation for Live Video Compression on a Kubernetes Cluster. SMPTE Motion Imaging Journal, volume 131, May 2022. Pages 45-49.