Universitat Oberta de Catalunya

Digital sustainability audits: a *de facto* standard for the Internet carbon footprint.

Master Thesis Report

Máster en Aplicaciones web y desarrollo de sitios

Author: David Monràs Álvarez

Consultor: Carlos Caballero

On the 8th of june of 2020

Copyright

Note of the author

I heartedly dedicate this work to the soil, the air, the green and the blue this planet Earth brings to me, with nothing in return and always giving me new opportunities from the sunrise to the sunset. I will sweat my hands to preserve it and do whatever it takes to bring back justice to it.

With heart and will

until my last breath.

Save the planet

David Monras

Abstract

The Internet is evolving at an unprecedented pace. The data flowing back and forth constantly changes and doubles up every two years. Internet traffic is seeing more and more clients and therefore, websites are mainly focused on scalability but not on sustainability which is becoming wide apart.
Among the earliest adopters, people well aware of climate change are known to be claiming for climate justice and this is increasingly becoming an exploit factor for new businesses seeking to boost their online presence. However, most websites lack the knowledge on how to effectively turn their resources to carbon neutral and most important: how to aim for sustainability.
Therefore, a RESTful API has been developed to perform sustainability audits and stress the essentials that every sustainability-oriented website must have.

Keywords: Sustainability, climate change, Internet, digital audits, RESTful API.

Abstract (spanish version)

Internet está evolucionando a un ritmo sin precedentes. La información que contiene está en un constante cambio y se duplica cada dos años. El tráfico de Internet cada vez cuenta con más clientes y por lo tanto, las páginas web se enfocan en la escalabilidad en lugar de en la sostenibilidad que se está dejando de lado.
Existe un público cada vez más numeroso sensibilizado con la justicia climática siendo el público objetivo de las nuevas empresas que buscan potenciar su imagen online. Sin embargo, estas empresas no poseen el conocimiento necesario para ofrecer unos recursos que alcancen la neutralidad de carbono y lo más importante: que sean sostenibles.
Asimismo, una RESTful API se ha desarrollado para auditar la sostenibilidad y resaltar aquellos aspectos esenciales que cada página web que aspire a ser sostenible debe tener.

Conventions

`This typography` will be used for code snippets.

List of Contents

Figures, tables and codes

Figures

Tables

Codes

1. Preface

The world has already entered 2020 and a major challenge is lying ahead: global warming and its consequences worldwide. Never-seen-before weather fluctuations, extreme and abrupt climate changes, the extinction of thousands of wildlife species, and meanwhile some people keep refusing it and acting foolish. There, at the top of it, lies a gigantic human-created infrastructure which is still growing up and has still a lot to say regarding the climate change: the Internet and the ICT which represents the 2% of the global greenhouse gases emissions and with a scientific prediction of reaching the 8% in the next decade [1].

Although a bad scenario is the best one might expect, the political leaders and major representatives are announcing controversial “enhancements” like the 5G, more investment in AI and in a few words, a tech-oriented society that are hungry for media resources and therefore, are likely to pollute more. People using this technology and becoming the end-users, are still lacking the mass conscience for facing those challenges and to trigger drastic changes for a good reason: one that has to do with protecting what matters more on this planet. Embracing technology does not mean to destroy ecology. Still, many actions can be taken to reverse this situation and it starts with the way the Internet and media is being accessed and consumed.

Has anyone ever thought about the carbon footprint of the streaming video?
An average of 600g CO2/GB [2] downloaded for just a single user, consumed by data center servers remotely allocated and burning fossil fuels to keep up with the pace. Switching to half the video resolution could mean to save up to the seventy-five percent of the energy consumption and if accounting for the fact that the Internet access providers are doing a good practice of: HTTP requests, uploaded media, cache, minification and bundle of the assets, etc., there is definitely a saving in power consumption and a reduction in the digital carbon footprint, and that’s exactly what sustainability audits are, a standard to showcase the essentials to become sustainable online.

2. Objectives

2.1 Main

- Create a new standard of the digital sustainability audits.
- Develop a RESTful API backend infrastructure that implements the standard.

2.2 Secondary

- Master the CDP for auditing websites.
- Scale up the backend infrastructure.

3. Theoretical framework

3.1 The Internet

The Internet is one of those terms that was created before its naming. As usually said in computer science, naming is one of the most difficult tasks. Indeed, it may seem simple to gather a few characters, maybe following an acronym, maybe playing with its pronunciation, but sometimes it's hard to come up with a clear and catchy terminology that persists.

The Internet origin

Back in 1969 in the United States of America, the first troops began to withdraw from the vietnam war, a fact to note if we consider the well-known statement that the biggest advances in technology always come up in a war scenario.
The ARPA comissioned by the DoD decided to invest in a new technology to ensure that their data communication were capable of surviving a nuclear holocaust. Before that, data communication was based on electromagnetic principles such as radio or electric wire and the end-devices (i.e. telegraph and semaphores lines) limited to point to point communications.
This initial effort resulted in the apparition of a packet-switching network, a relic of the modern TCP/IP switching protocol that caught the attention of many: the ARPAnet. Its success was such that the government unclassified it and called it the MILNET. It's initial users were university hosts scattered in the U.S territory that took advantage of its benefits and made its usage popular under the term Internet, a combination of the MILNET and a smaller ARPAnet.
The Internet grew in popularity and in 1987 was running at 56 Kbps in a three-tiered network topology that included a backbone, the regional networks and the local networks.
This rising trend has not stopped since and nowadays the Internet encompasses hundreds of thousands of networks worldwide and delivers data to a wide range of electronic devices in several formats.
It is also maintained and built by national commercial providers (i.e. Internet Service Providers) that grant the Internet access to its paying customers, whereas the government's role is to stop monopolies and regulate the online market by creating new taxes around online services and protecting personal data privacy (although failing from time to time and causing data leakes which are sold to the highest bidders).
The Internet experienced (and still does experience) a paradigm shift.
Here, the focus will be on the 'content' aspect of the Internet that people usually know about and use it on a daily basis (i.e. websites). Of course, a different approach could have been taken, for instance, by naming the computer revolutions that moulded the way communication enabled social networking among other milestones, but for the ongoing project, this is off-topic.

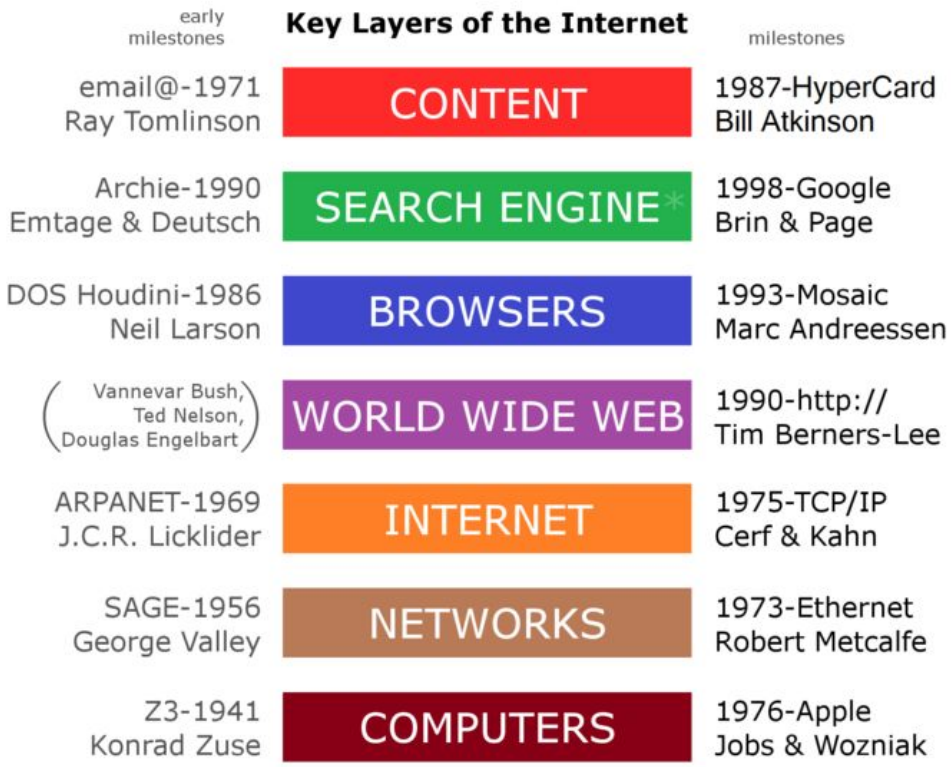

Figure 1 : Seven key layers of the internet are illustrated, showing how users are connected to content (ie, web pages) served by content providers. Significant milestones are highlighted in columns on the left and right, showing key contributions made by some of the instrumental people involved in inventing and developing the technology. Listed on the left side are the highlighted rudimentary advances. Listed on the right side are the more refined advances that helped to enable major breakthroughs.
Source: By Concord hioz CC BY-SA 4.0 [3]

The content layer of the Internet is conceptually sitting at the top, as figure 1 shows.
It was brought by important technology advancements towards an era of virtually Internet dependency, which will be reached after taking a brief detour.
Going back to its starting point, in the decade of the 1990s one began to see (those fortunate to have Internet access) the apparition of primitive web browsers after being released from the CERN facilities.
Digital resources, identified by URLs, formatted in HTML and interlinked by hypertexts, were able to be fetched over the Internet network (from now on the network) using the HTTP.
A user agent running on a web browser was responsible for requesting the resources and handling the networking details with a remote web server.
Altogether, the initial client-server architecture more than satisfied the market demand and eventually the WWW did the standardisation job.

The first Paradigm shift

The first paradigm shift began in 1990 when portal sites such as Yahoo and Lycos were profitable by indexing documents and making them available via their portal sites.
Interestingly, the information that formerly occupied their collections have vanished from the digital world and are no longer visible, not because they have been explicitly removed but because of the vast amount of information that came after them.

Portal sites tackled the indexing problem and grouped documents by their respective categories. So, in order to look for a specific piece of information one had to look for its appropriate category.

Figure 2: A screenshot of Yahoo on 8 Oct, 1999. Licence: Public domain

Why was this an issue? Were not categories suited for making order to a world full of entropy? People have not been particularly good at consensus. Imperial, metric system, international system… they still do not agree on what units should the longitude definition be. So categories for naming something to be accessed by everyone was not a good idea.

The second paradigm shift

The second paradigm shift in 2000, pushed Google search engine to the top of the Internet world. It took a search bar and notably computer power horse to deal with indexing large data. It also took a tech giant with fierce hunger for personal data.

Google's vision was undoubtedly successful, they opened up a new dimension in front of people that were then able to search any thought they came up with. However, they early noticed that the search behaviour of people was oversimplified with the assumption of impersonate searches. Indeed, people used the browser to find an answer to rather personal topics.
For instance, how did the impersonated Google search engine know one's reading and fashion preferences? They needed to know more about people, but of course, without asking directly which would have been a major hassle during the search experience.
Then governments joined the game (if they ever had left it) and liked the idea of having fine detail on personal search interests, so they purposely backed up those practices but always from the shadow and like time proved, they needed almost twenty years to start regulating what and whatnot could be gathered. Still, there's no legal way of getting rid of website's trackers, not besides opting out.

The Persuasive Internet

In 1998, BJ Fogg (Fogg and Fogg 2003) warned in their paper entitled: "Persuasive Computers: Perspectives and Research Directions" about a new branch of study that consisted in the study of computers as persuasive technology, or as he put it: "captology".
Persuasion is an attempt to shape, reinforce, or change behaviours, feelings, or thoughts about an issue, object or actions As the computers themselves do not yet have intentions, those who created, distributed or adopted the technology do so with an intent to affect human attitudes.
He therefore proposed computer's function in three ways: as tools, as media, and as social actors. The tools provided users with a new ability or power, the media conveyed symbolic content or sensory content, and the social actors adopted animated characteristics and played animated roles which in turn, followed social rules or dynamics. All three ways are potentially persuasive as explained down in the table 1.

Function	Essence	Persuasive affordance
Computer as tool or instrument	Increases capabilities	● Reduce barriers (time, effort, cost) ● Increases self-efficacy ● Provides information for better decision making ● Changes mental models
Computer as medium	Provides experiences	● Provides first-hand learning, insight, visualization, resolve. ● Promotes understanding of cause/effect relationships. ● Motivates through experience, sensation
Computer as social actor	Creates relationship	● Establishes social norms ● Invokes social rules and dynamics ● Provides social support or sanction

Table 1. Computer functions and their persuasive affordances. Source: [4]

One might start to recognise the important issues raised from the merging of the Internet's domain and persuasion. Ethical and value issues come to play a crucial role. And that is precisely, what has been vastly ignored by the Internet's third paradigm shift: social networking.

The Internet's third paradigm shift

So how could still the gap be reduced between information and individuals so they could be closer to what they search for? Asking for it was not an option, because that would have been too mean. But what could be done instead, was to create a website application that behaves as a social actor, so the individual just would need to fill its personal information in order to let others find them. So whenever they meet each other, social networking would be emulated by enabling a new series of tools such as: live chatting, media sharing, relationships mapping, feelings expression by liking others content or posting out one's thoughts, etc.
Now, not only there was a clean and legal way of collecting personal information but also a persuasive medium that may create: addictions in youth, social inhibition, health injuries, sedentarism, lack of empathy, etc.
Hence, is there a perception of the problem introduced by social networking?

Someone once said: "breaking changes are more likely to be adopted if they are tagged as normal". Societies did learn indeed a lot from social networking. They saw it as an opportunity to promote shopping, branding, fashion, and as a means to monetise user visits via advertisements.
And they really can achieve huge benefits, depending on factors such as: website traffic, popularity, etc.

What the earnings are is still unclear, but to put an example everyone is going to comprehend, Mark Zuckerberg, the owner and creator of the tech giant Facebook, who in 2018 testified before the US senator

for the Facebook data breach affecting 50 million user profiles, in response to the senator asking how Facebook created a business model around a free service, Mark simply answered: “Senator, we run ads”.
Nowadays, everytime one searches the Internet or as it has already been put ‘everytime one googles something’, the search query is being monetized. The searching keywords are being collected and kept for further analysis (now being possible by the advent of Artificial Intelligence) and the searching engines show ‘intelligent’ (i.e. an algorithm computing the likelihood of being compatible with a service) advertisements next to the search results. Of course, the service may only get benefits from Google if it catches the attention of an individual and eventually, there is a navigation to the vendor’s website. But most definitely the service had to pay upfront to be listed, so one may conclude that searches are monetized.
In response to the obvious flaws to privacy and also to data protection, the UE in 2016 decided to regulate those practices with the known GDPR 2016/679.
In broad terms it stated the following:

- No data shall be processed if not under explicit consent, contract, public task, vital interest, legitimate interest or legal requirement.
- Declare the lawful basis and purposes for data collecting.
- Disclose any information upon user withdrawal.
- Inform to authorities of data breaches.

It took 16 years for an international organism (2 years plus for enforcing its fulfillment) to start protecting user privacy on the Internet. Still, they haven’t adopted an official position or showed any concerns of the persuasive capabilities of the Internet and as incredible as it might sound, they neither have introduced a solid regulation for websites and search engines that right on this moment are running hundreds of thousands of programs to collect: individuals device information, IP addresses, interaction behaviour, cross-site tracking data, among other nasty data.
Certainly users have now a popup warning them that the current website is making use of cookies to specifically populate user analytics, but If one refuses those cookies, not the functional but the tracking cookies (if there’s still a slim chance to do so) and still attempt to visit the website, in the vast majority of cases, they will be prompted with a white screen which in computer science words means: ‘Intentionally blocking the cookies no content is viewed’. One may wonder whether the cookies the website referred to were a conjunction of functionality plus analytical, so the rejection is justified. But if that were the case shouldn't they have been informed? Also, according to the vigent law, shouldn’t they have splitted down the cookies to let one decide which one to consent?
At the end, if one does not allow those cookies the website’s access will be denied.
The websites running this way carry more troubles than benefits so it is recommended to simply avoid visiting them.

The human profiles behind the Internet

One might now wonder who is being the scenes. To whom those ranking algorithms belong and who might be actively contributing to make them fit the requirements of electoral and political agents?
Computer scientists, full-stack developers among other high tech profiles handle the proprietary trade-secret algorithms, their management and maintenance. With the obfuscated usage of tracking algorithms, user feature-rich metrics are obtained that ultimately, help to sustain ads based business models.
But the underlying question, that everyone seems to avoid, is whether or not it is really needed a version of the Internet which is extremely data oriented as of today. One that is constantly playing at the edge of user privacy because it barely has legal responsibilities.
It all depends on which purpose those data collecting practices serve for.

Towards a purposeful Internet

From the very beginning all the attempts to collect data without explicit user consent and acknowledgement should have been avoided. But still, even if the algorithms have been fed with valid (in this case, with previous user consent) data only the content that was somehow beneficial to users had to be put forward, although one might still have its doubts whether showing an advertisement of a half discounted product would bring any benefit.

If any web developer has ever decided to track their users down under their consent, he should have thought of the existing alternatives such as gathering user opinions based on surveys, etc., but above all, a clear case study should have been specified (i.e. study goals, timelines, variables, etc.). Also, data must have been used appropriately and once the study conclusions were drawn the changes must have been applied to the subject of matter.

Oppositely, the same developer would not be making a good use of the tracking techniques just for curiosity or simply because there is a remote possibility that they would be useful in the future.

One might think of some use cases where user analytics would be convenient.

For instance, why would a website load all the resources in a landing page if the user were just going to click on a navigation link? Or even better, a user rapidly bouncing back to the search results because the content is not what he expected?
Considering the fact that a landing page normally includes media assets like images or embedded videos, which are costly to transfer over the network due to its size, it would be a great benefit to save on bandwidth (and gain on efficiency and sustainability) by just serving the images on demand (i.e. when a visitor scrolls down to a known fixed point), and displaying a mock element until then.
In the described scenario, analytics might be enabled but imperatively after informing the visitors the fact that they are being analysed; maybe via a banner with some settings following a thorough explanation to opt-in or to opt-out.

Worldwide developers of today possess great power to influence the way people use the Internet: their work extends the human-machine interaction and their designs could make user interfaces persuasive and highly addictive. But as often said, with great power comes great responsibility, and it is not an option to forget what are the working gears sustaining the whole infrastructure. Either cannot be forgotten that servers, transport networks and devices all require electricity to be powered on and as of today, the main source of electricity generation involves the combustion of fossil fuels, and as it would be explained in the next section it actively contributes to the warming of the planet Earth and poses a serious threat to life.

3.2 Climate Change

Climate change is indeed a profound challenge but the picture of the society setted to face it down is one of: controversial economics, besieged science, complicated and bitter politics, environmental and non-environmental issues both running in many directions, among other issues. This makes it extremely complicated to reach a global agreement on how to address the challenge adequately [5].

These issues peril to go unattended (as they often do) if we don't put them into their proper context. Therefore, in order to understand the reasons why we focus on climate change, we must first understand the underlying social, political and economical systems, which are the driven force of a society putting at risk ecosystems, wildlife and human beings.
This could be a whole project itself due to its intrinsic complexity, but here I think it is reasonably fair to provide a snapshot of the elements which uphold our society, with the final goal of learning how to better respond to climate change and how this project is committed to this task within the Internet boundaries.

Fossil fuels

Coal, natural gas, crude oil, and its derivatives are known to be fossil fuels because they were formed from the fossilized, buried remains of plants and animals that lived millions of years ago. Due to this fact, they possess a high carbon content which explains its high caloric capacity and why it is used in the power and transportation sector.

Fossil fuels are burned in order to release its energy so it can be transformed to a human profitable form.
The extraction, production, transportation, refination, consumption, etc., phases concerning fossil fuels, are known to emit pollutants that harness our environment.

Greenhouse gases

Among those pollutants and in the volatile form, it is remarkable the environmental impact of the GHG.
They have the property of absorbing and emitting radiant energy within the thermal infrared range, a process described as trapping heat in the atmosphere which in turn makes planets hotter relative to their average surface temperature or generally known as global warming.

GHG may be naturally present in the environment but since the Industrial revolution and mostly since the anthropogenic contribution there has been a substantial increase in their concentration.
The most relevant and their sources are represented in the table 2.

Greenhouse gas	**Anthropogenic source**	**Atmospheric concentration**	**Data sources**
Carbon dioxide (CO2)	Electricity production, Industry	416 ppm (2020) (+0.6% /year)	[6] [7]
Methane (CH4)	Agriculture, Oil mining, Residential	1830 ppb (2019) (+0.4% /year)	[8]
Nitrous oxide (NO2)	Transportation, Industry	No data	
Fluorinated gases	Commercial, Industry	No data	

Table 2: Greenhouse gases. From top to bottom: Top highest concentration, lowest greenhouse effect. Bottom: Lowest concentration, highest greenhouse effect by unit. Source: IPCC

From the above table, one may extract that carbon dioxide is the most predominant GHG, but also the least impactful (unit comparison). Also, that the industry is present in most of the cases. The industry powers the human economy: it creates workplaces, generates goods, promotes consumption, etc., which is one one the reasons why fighting climate change proves to be difficult, because it goes against economic interests. And this seems to be a persistent problem impossible to get rid of as the current COVID-19 pandemic proves: it is important to preserve people's health but the economy is pushing the recovery to the limit, sometimes by shortening deescalation and prioritizing people's favorite spare time hobby (i.e. football) although knowing that pandemic is not controlled and the number of cases may increase. Even so, the efforts made to stop the pandemic are there, and also prove that emissions from human activity can be seriously cut down as a study suggested of -7% [9] .
Only had some of the seen effort put at stopping the COVID-19 pandemic, were put towards fighting climate change, quick and effective results would be advised soon.

On the one hand, the fact that carbon dioxide has the least greenhouse effect of the list of gases listed in table 1 should not be a misunderstanding or a source of relief. In fact, in a complex climate system scenario like this one, carbon dioxide may push parts of the Earth system into abrupt or irreversible changes, generally known as tipping points. Many of their consequences are under study from west African monsoon shifts to permafrost decreasement which may still accelerate climate change by the release of methane and permafrost carbon dioxide. Hence, the key aspect to note is that anthropogenic emissions of carbon dioxide may drive an irreversible climate change situation with irreversible damage to our planet and hugely impact the globalized world, extenuating even more inequalities among people and putting at risk a myriad of species.

Climate change is undoubtedly the biggest challenge the human race has in front of him, and clearly has a lot to say about how the world has been evolving since humans proclaimed their superiority to the rest of species. More than another problem to solve (it is not a simple matter of fixing the emissions) it is a clear symptom that there are several inherent problems with the way society runs.

Limiting global warming to 1.5C

What can be done in order to slow down climate change? Is it still possible to revert the current situation? If yes, what does it take?

Imagine the context where the working gears of the society (with the permission of economics) are redesigned in a way that not only there are less emissions because efficiency excels but also thanks to a responsible consumption. In this scenario, monitoring the carbon dioxide emissions and the average surface temperature globally would bring valuable feedback to assess the progress made.
Limiting global warming in the next decade and therefore climate change, is a rationale suggestion made by the IPCC who also considered the different levels of risk of reaching the feared tipping points. Specifically, they have put forward a global temperature scenario of 1.5 degrees celsius not to be crossed in the next decade, this having its end in 2030.

The Internet domain

While climate change is constantly getting worse and society is placing all the hope in the current decade to slow down its progress, in the background there's a domain that has a receipt for the disaster.
The Internet and the CT accounted for about 3.7% of global greenhouse emissions in 2019 (more than 300 million of metric tonnes. That's 9 zeros of kg).
Most people don't have a clear image of the physical aspect of the Internet. And that's reasonable, because it encompassess sending electromagnetic signals over the air, high speed fiber glass conductors, extensive pipelines crossing oceans to link entire continents, satellites broadcasting data in the millisecond range, data centers performing CPU expensive tasks, devices responding to user interaction, etc.
The list of the elements forming the Internet is vast and extensive and albeit for experienced people it is one of the greatest mysteries and a source of amazement by non-experienced users.

The shift project in 2019 suggested that the current trend for digital overconsumption in the world is not sustainable with respect to the supply of energy and materials it requires.
The digital consumption also referred to as the consumption of goods on the Internet that involves the usage of digital technologies like smartphones or laptops, is increasing exponentially. This has to do with the number of users that every year acquire a smartphone with Internet access and use it to consume media in multiple formats. Images and videos, together accounting for about the 80% of the data traffic, are widely accessible in social networks and web applications. Also, since the advent of cloud computing, which has a

large energy consumption, the content can be distributed and accessed more easily which gets the attention of newcomer users digitally transitioning.

On figure 3, graphs and quantifiable content are illustrated on the energy forecast and the Internet explosion that is currently taking place.

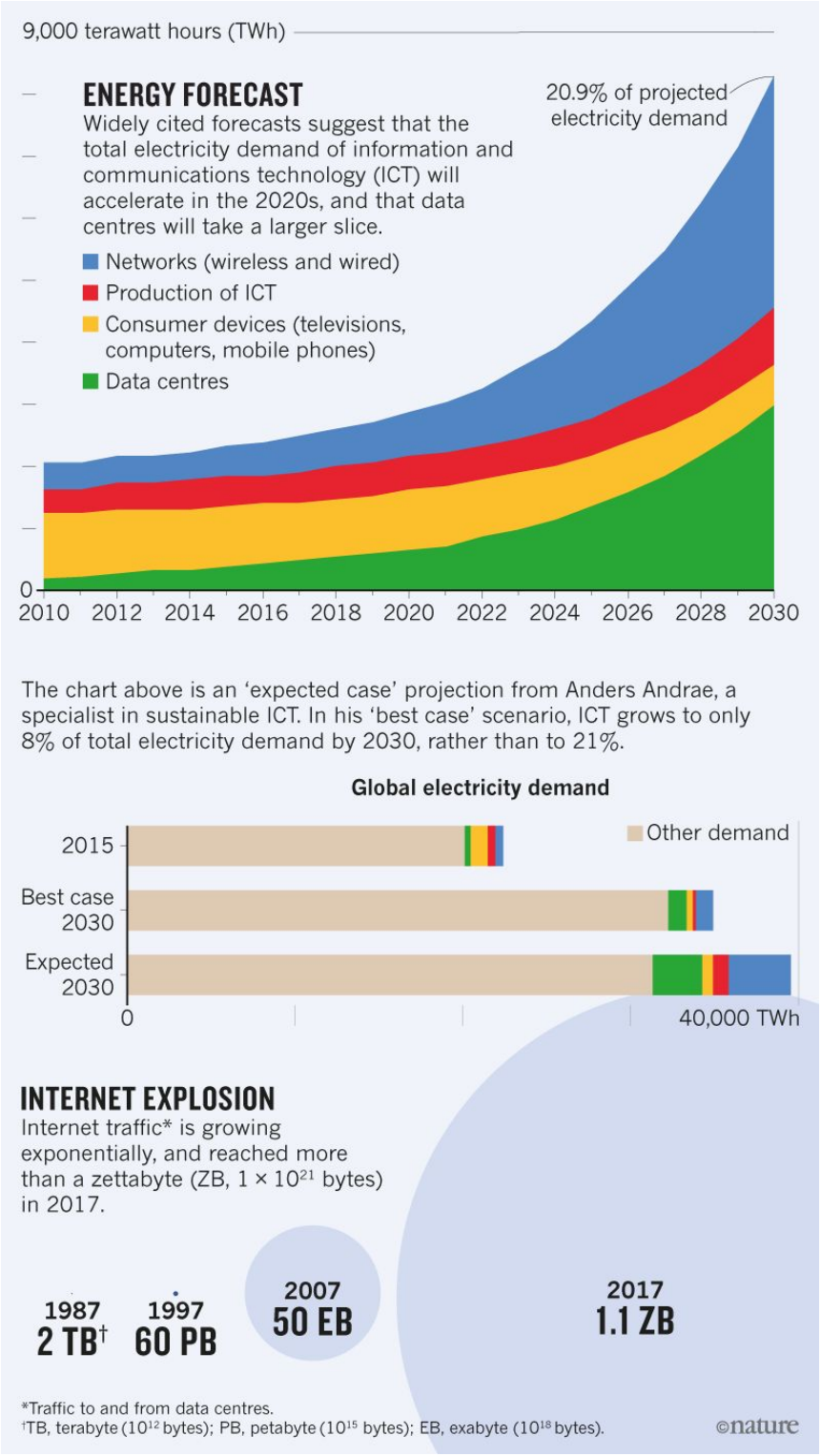

Figure 3: Internet's energy forecast and explosion. Source [10]

Since 2016, mobile users are surpassing desktop users. Technology has adapted to the way people move, live and use the Internet. Now, some of the last generation smartphones have the hardware equivalent of laptops but packed up in 95% less space. Also, this plus of computational power is increasingly becoming cheaper and therefore accessible to the rest of the world. Developing countries have found in smartphones a way to access the Internet, which was in the past impossible due to the obvious inequalities in their system.

As the world bank in 2016 stated: “In developing countries, the households that own a smartphone are more numerous than those who have access to electricity or clean drinking water, and nearly 70% of persons belonging to the lower quintile of the population own a smartphone."
The Internet’s will continue to rise exponentially but the enhancements in networking technology which were limiting the Internet connectivity to 2G networks is expecting to increase global coverage in those areas where building infrastructures were inexistent, is now expected to be possible soon thanks to the launching of low orbital satellites (thousands of them), as the starlink project by SpaceX depicted in figure 4.

Figure 4: The SpaceX starlink satellite captured from the ISS (International Space Station) Source: NASA.

More than another ‘cool’ project, bringing broadband Internet in a period of time when everyone wants to take part in it, it is not the brightest idea in a climate change scenario (another controversial fact regarding economics). The Internet itself may bring positive effects to people well aware of its threats (i.e. own privacy, health issues, etc.) and willing to make a responsible consumption. Nonetheless, as seen in the past section, it took a lot of time to start regulating it and even so abuses and bad practices are rather common.

From now to 2025, the penetration of the mobile internet should increase by 50% and reach 61% of the world’s population. The greater part of the increase (1.75 billion) in new mobile internet users between 2017 and 2025 will come from China (about 350 million new users), India (330 million) and Sub-Saharan Africa (280 millions), who also have the highest carbon intensity ratios [11].

Now more than ever Internet users have to follow a moderate pattern of consumption or like has been put before a ‘digital sobriety’ as a principle of action, altogether with a widespread awareness on the climatic situation, not only in users but also in professionals in charge of developing the technology. Otherwise, there will be a huge risk of leading an out-of-control environmentally digital transitioning.

4. Proposed solution

Digital sustainability audits as the name suggests aims to showcase the aspects involved in a website development, production and maintenance stages which have to be fulfilled in order to preserve sustainability.

Although being broadly used in a myriad of domains, there is not a clear definition of what sustainability is and the term is often interchanged with performance or efficiency.

In this project whenever sustainability is a matter of concern and of discussion it shall be defined as :

" a set of practices designed specifically to handle resources in a consciously and respectful manner so as to preserve at all cost a global balance".

Nowadays, preserving the global balance has to do with protecting the environment and also with making a good usage of its resources. Those resources are present in many stages of a website's lifecycle.
On one hand, digital devices of all forms are necessary to enable full control over the data flow. Those demand many precious ores like copper or silicon which have to be mined (and bear an environmental cost) and as such their availability may become scarcer in the future.
On the other hand, software running on the machines including programs, utilities, fonts, updates, among others, which are created and maintained through energy-intensive processes.

Thus, a *de facto* standard (practical and effective for everyone but not following a legal right) will be proposed to evaluate websites through the usage given to their resources and demonstrated compliance with sustainability practices in the web domain.
To this end, two main categories both encompassing have been created:

- Server
- Design

They encompass specific audits to make a general assessment on the sustainability aspects of a website and shall be thoroughly explained in the next section.

5. Methodology

5.1 Server audits

Most websites do not own the hardware required in order to host their services online. Currently for low to middle range businesses it is less troublesome (and also cheaper) to pay a fee to a third party owning the equipment and command the implementation to suit the needs. Still, they can choose one or another provider according to the price, location, trustiness, energy source, etc.
What websites do really have much more control over are the resources related to software development, like those regarding the backend and frontend infrastructure.
Knowing that, every piece of code developed that serves for a specific functionality will be responsible for the global accounting of a website's environmental impact usually measured as the carbon footprint index in grams of carbon dioxide equivalent.

Therefore, if one were intended to calculate the carbon footprint of a given website it would need to consider the energy consumed in the:

- Data transfer on the network
- Data crunching on data centres
- Rendering in the end device

Also the parameters referring to the :

- Energy intensity of web data
- Energy source used by data centres
- Carbon intensity of electricity
- Website traffic

Data transfer

Data transfer can be measured for each transferred resource in bytes.
Before being transferred, resources are known to be compressed or encoded (depending on the resource type or MIME type) by HTTP compression algorithms like GZIP.

Accounting just for the uncompressed bytes (i.e. the bytes uncompressed on the client-side) would be an oversimplification as seen in different carbon footprint algorithms [12]. Besides, they are not capable of fetching AJAX calls and thus unable to fetch dynamically rendered resources. Fetching less resources and those fetched filled with the extra bytes of uncompression may explain why the resulting carbon footprint is not as bigger as expected.

Data crunching on data centres

Data crunching on data centres is hard to measure mostly because this information is not available and its effects are constantly subject to changes.

As it will be seen in figure 5, there have been three major effects that may have turned data centres more efficient energy-wise: energy efficiency of IT-devices, server virtualization software; which enables multiple applications to run on a single server and migration to large cloud and hyperscale-class data centers; which utilize ultra-efficient cooling systems to minimise energy use [13]. However, several effects also may have helped to raise the service demand: increased global storage capacity, increased traffic, increased instances and installed base of servers.

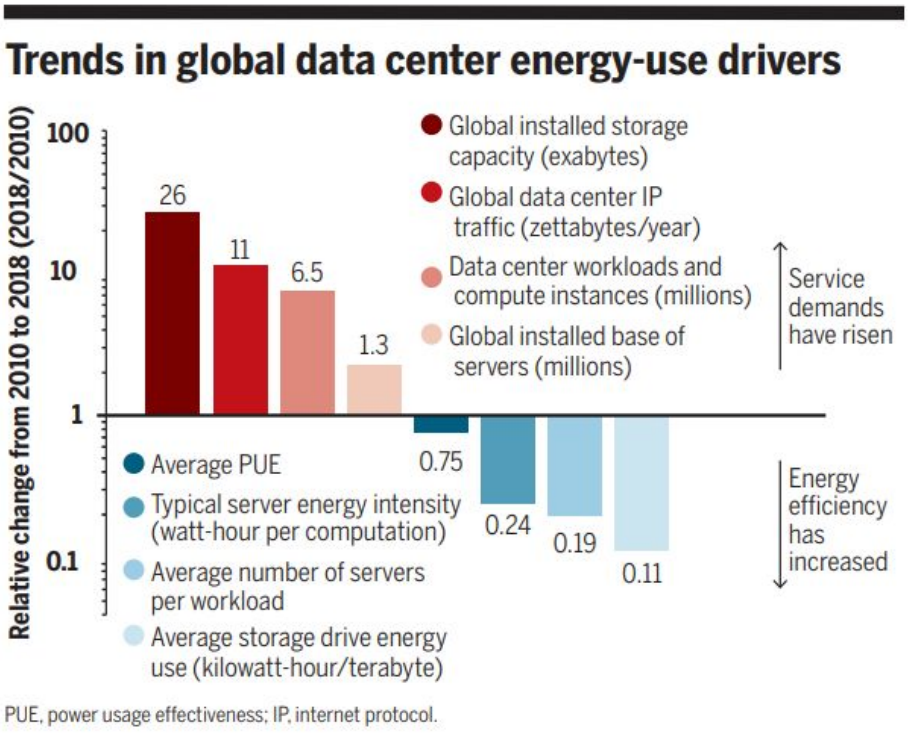

Figure 5: Relative change global data centre energy use (2010 = 1). Source: [13]

Oppositely, the only reasonable parameters that can be obtained from are energy source and location. Both substantially reduce the overall carbon footprint, the former by using a renewable-powered facility and the latter for determining the energy intensity of the region or country where it is settled.

Rendering

The energy consumed during the render stages on end-devices is also a challenge. For an exact measurement, it would take to plug into the battery a device to monitor discharges curves and rates. However, it can be estimated with the max CPU intensity during the process and comparing this value to the known thresholds.

Energy intensity

Regarding the source of the listed parameters, the energy intensity of web data greatly differs from one study to another, ranging from between 136 kWh/GB in 2000 and 0.004 kWh/GB in 2008 [14]. This is due to the system boundary considered, the year which the data apply, characteristics of access networks, methods used or technical assumptions.

Carbon intensity of electricity

Generally speaking the electricity intensity of energy can be divided into two groups: the production and consumption. CBA as has been suggested in [15], is a very important aspect of climate policy, and has to do with assigning responsibility to consumers instead of producers.
This calculator differs from country and regions and arises from the whole life cycle of power plants (i.e. construction, fuel production, operational emissions and decommissioning).
Getting to know the electricity intensity worldwide by regions is such a complex task that even the web services providing the information are still not covering all the areas, see for example TomorrowCo ElectricityMap.

Website traffic

A website's carbon footprint is proportional to the number of times the website was visited or requested through the network. Therefore, the number of visits should be estimated for a given period of time in case the real time traffic for a website is unknown.

Server audits

The table 3 shows the applicable audits in this category.

Audit name	**Explanation**	**Values** ***(Binary / Numeric)***	**References**	**Notes**
Server is green	Evaluates the energy source of the server that hosts and delivers the assets.	Binary	Third-party API [16]	
Digital carbon footprint	See appendix calculation 1.	Numeric	Carbon intensity per region: [17] Electricity intensity of web data: [14]	
Uses compression	Evaluates if the server has enabled HTTP compression on the delivered resources.	Binary		Scripts/xhr containing personal information, woff /woff2 font format and jpg/png images should never be compressed.
Uses HTTP2	Evaluates if the server has HTTP2.0 enabled.	Binary		A requisite for any website prior to using HTTP2.0 is to configure HTTPS.

Table 3: All the applicable server audits by name, explanation, values, references and notes.

Server is green

In order to make an evaluation of the energy source of the website's origin server an IPv4 address is obtained via the remote address header in the HTTP response. Then, this is asserted to the origin hostname and posted to a third party API that is responsible for returning whether the given address is green hosted (using renewable energy source) or not.

In the scenario of having this information locally (i.e. local database) it would be recommended to perform some sort of memory caching and wrap everything in a pure function (i.e. one that always returns the same output to the same input). Unfortunately, a database layer has not been implemented in this project, instead a mechanism of memoization (i.e. store in cache the pure function) provided by a third party NPM library.

This directly benefited the total execution time of the main script while also reducing the number of unnecessary API calls.

Digital carbon footprint

As already explained before, one of the ways to quantitatively measure the environmental impact of a certain activity is by calculating its carbon footprint index.
Refer to the appendix section calculation to find a detailed explanation of the method applied.

Uses compression

Optimising HTML, JS and CSS assets has to do with HTTP compression. It compresses the original bits of information but using less and without losing any. The *de facto* compression technique on the web is GZIP, not due to its high compression ratio but rather to a trade-off between computational cost and compression ratio. Text files may get benefited from HTTP compression and also those that have repeated text like CSS stylesheets. Image formats like PNG, JPG or WebP do not as they are already compressed.
According to the HTTP Archive Almanac, 23% of the total audited websites with the Lighthouse auditing tool could benefit from enabling text based compression. Also, there is still room for improvement for the text formats: text/html, font/ttf, application/json, text/xml, text/plain, image/svg+xml, and image/x-icon.

The developed script will firstly, filter out non applicable resource types by looking at their correspondent MIME type in the ‘content-type’ header, and then get the compression ratio calculated for the given request as the compressed size (also encoded size) over the uncompressed size (also decoded size).

The audit score of carbon footprint is calculated as the (1 - percentile) for a given value in a log-normal distribution specified by the median value at which the score will be 0.5 and a 10th percentile of the given value at which the score will be 0.9. Likewise, the score represents the amount of distribution greater than the given value.
The log-normal distribution is plotted in figure 6.

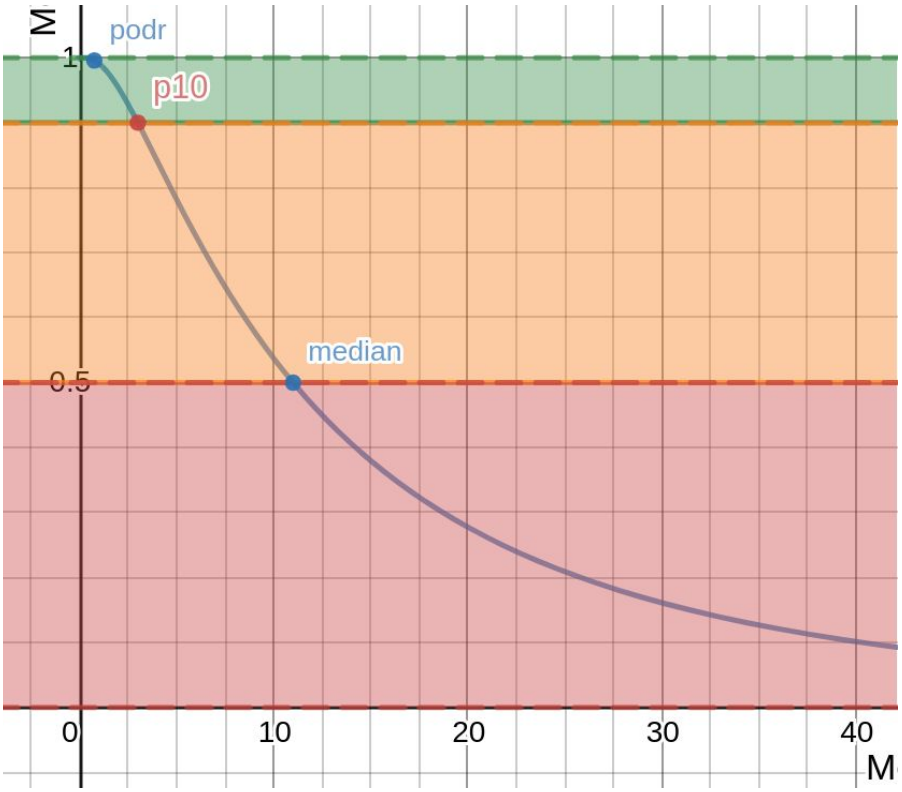

Figure 6: Log-normal distribution of the carbon footprint audit score. The x axis represents the carbon footprint value calculated as (1), and the y axis represents the audit score between [0,1]. More details of this distribution can be found at [18]

Uses HTTP2

HTTP2 is the multiplexed version of HTTP1. Instead of using a single one TCP connection, it performs all the requests in parallel. It has many advantages over its predecessor, namely: increased security; due to the requirement of having HTTPS preconfigured, compression of HTTP headers; which were before unavailable, reduced latency, among others.

If a service worker is requested it should not be considered in the equation as they may be running on HTTP2 but some browsers display wrong information and they are marked as running on HTTP1.

Apart from the last well known issue, it is relatively easy to see if a given request runs on HTTP2. It suffices with fetching the protocol used or knowing the request was served over http and not https it can be deduced that it was neither running on HTTP2.

The code snippet 1 partially shows how this information is obtained using the CDP.

```
export default class CollectTransfer extends Collect {
      static async atPass(passContext: any) {
            const protocol:any = []
            ...
            page._client.on('Network.responseReceived',(data:any) => {
                  if(data && data.response.protocol) {
                        protocol.push({protocol:data.response.protocol,
                        reqId:data.requestId})
                  }
            ...
```

Code snippet 1. Extracted from server/src/collect/transfer.collect.ts in the project's code repository.

The data object returned contains a response property where the protocol information per request is fetched. For the full documentation on the data object returned refer to [19].

5.2 Design audits

Websites of today are developed by a combination of HTML, CSS and JS files. The team ahead of the development force consists of back-end and front-end developers that take on the server and client sides respectively, and also of visual designers that take on the visual aspects such as brand management.
Generally speaking, they aim on transforming software resources to user consumable content.
Hence, their actions not only determine how the website is going to process all the user interactivity but also how the website is going to deal with sustainability.

In order to make a quantitative evaluation of a website's assets one has to look after the assets size in bytes as it reflects the total amount of information contained. An evaluation of the innermost content also proves useful (specially for evaluating how maintainable is the code) but unless the source code is provided it's hard to assess anything as the downloadable content is usually splitted up in chunks of data, minified and uglified to short characters to save on file size.

Applicable audits

The audits proposed in this category, seeks to promote a responsible usage of the assets, remove useless content and take advantage of modern technologies.
The applicable audits are in table 4.

Audit name	**Explanation**	**Values** ***(Binary / Numeric)***
Uses lazy loading	Evaluates whether the non critical images are lazy loaded	Binary
Don't have console logs	Evaluates whether error, warning or info console logs are present.	Binary
Uses font subset	Evaluates whether the downloaded fonts are subsetted to the characters being used.	Binary
Uses WebP image format	Evaluates the images format to be WebP.	Binary

Table 4: Applicable design audits.

Lazy loading

Lazy loading, also known as asynchronous loading, consists of deferring initialization of an initial object until the point at which it is needed. In the design audits, the target subject are images although it can be applied to other subjects.

It was initially discouraged because it was thought it affected the SEO. However, search engine crawlers use alternative techniques to properly index images such as using huge viewport heights to push lazy loaded images or auto scrolling, so it is no longer a reason not to use it.

One might ask why the importance given to images in web pages. Images are the most popular resource type on the web and according to http archive it accounts for the 55% of the total page weight with 1618 kB in desktops and 4% increment in the last 4 years [20] .

Therefore it is reasonably fair to conclude that the number of images in web pages are on the rise and also that they are widely becoming more visual.

Taking a look at the same http archive reference, they have calculated in the state of images report that an average of 830 kB could be saved in mobiles by lazy-loading offscreen images and hidden images, or to put it in other words, the image share on the total page weight could be reduced a 50% with lazy loading. Hence the importance of this audit.

To the date, there are at least three ways to lazy load images in the web:

- Native lazy loading

As of february 2020 the lazy loading in iframes and images is part of the specification [21]. The global coverage on browsers is good (currently 64% [22]) and if rendered on a non-compatible browser the genuity of HTML will make them render as default without breaking the page (i.e. eager loading).

- Intersection Observer API

It provides a solution to asynchronous observe changes in the intersection of a target element with an ancestor element.

- Native JS libraries

The current script that evaluates lazy loading on images, is an implementation of an auto scrolling function altogether with an event listener that filters out new image requests. This is shown in code snippet 2.

```
export class CollectImages extends Collect{
    static async afterPass(passContext:any){
        const lazyImages:string[] = []
            ...
        page.on('requestfinished',(request:any)=>{
            if(request.resourceType()=="image"){
                lazyImages.push(request.url)

            }
      scroll function credits to nagy.zsolt.hun [23]
      await page.evaluate(() => new Promise((resolve) => {
                let scrollTop = -1;
                const interval = setInterval(() => {
                  window.scrollBy(0, 100);
                  if(document.documentElement.scrollTop !== scrollTop) {
                    scrollTop = document.documentElement.scrollTop;
                    return;
                  }
                  clearInterval(interval);
                  resolve();
                }, 10);
              }));
```

Code snippet: 2. Extracted from server/src/collect/images.collect.ts in the project's code repository.

Don't have console logs

Console logs are the text messages that may be obtained as a result of: a deliberated event, an unexpected behaviour or an announcement.
At the development phase of the website it is rather common to output a certain object value to gain knowledge on its behaviour. This is also a common practice for debugging. However, when deploying the website to production those logs should be removed as they are no longer useful.
Similarly, there might be warning logs to announce new breaking changes or methods which will be deprecated in the future. These logs should be addressed by the developer of charge as it may introduce complications to the website, errors or security vulnerabilities.
Finally, error logs may also be produced as result of unexpected behaviours or warning logs that may have transformed to errors when ignored. Those logs must be addressed as soon as possible and it is nonsense to ignore them.
Likewise, it is not feasible to have console logs in the website in the long term as they are a waste of resources.

Console logs are obtained by listening for a console event on the page class in the Puppeteer library. This is shown in the code snippet 3.

```
export class CollectConsole extends Collect {
      static async afterPass(passContext: any): Promise<any> {
            page.on('console', async (message: ConsoleMessage) => {
                  const information = {
                        type: message.type(),
                        text: message.text()
                  };
```

Code snippet: 3. Extracted from server/src/collect/console.collect.ts in the project's code repository.

Uses font subsetting

All resources containing the eot, ttf, woff, woff2, or otf file extensions or a MIME type containing font are considered fonts. According to the HTTP Archive, fonts on average weigh up to 5% (132KB on desktop) of the total page weight (43% increment compared to 4 years ago). Depending on the way fonts are downloaded, they may contain extra characters sets, ligatures, glyphs, weights, etc.,, and sometimes those are not used which may explain the unnecessary increment in the font file size.
Font subsetting refers to the activity of shortening fonts to the strictly glyphs being used.

Currently, Google Fonts is a popular example of a collection of open source fonts and a font delivery system on the web. Fonts delivered this way are: compressed using the .woff font format, in a compatible format thanks to a system detection algorithm and have the possibility of becoming subsetted by url encoding params.

Another way of including font files to a web, is using the @font-face rule in CSS stylesheets and specifying an url or embedding the font as base64 encoded strings. Doing so one can subset the font by specifying the unicode range property, but nonetheless, it still needs to know the glyphs used. To this end, js libraries were made such as Glyphhanger which finds unicode-ranges [24].

The developed script, an extraction of it shown in code snippet 4, loads on the page the Glyphhanger script to detect the unicode range of both local and non-local fonts. After filtering out the local ones, it then looks for @font-face rules in the css stylesheets and most importantly for the unicode-range property which is the indicator whether the given font is being subsetted. CSS stylesheets are stored as string files so a library is needed in order to parse its rules and selectors into an AST. This is done thanks to the NPM library CSSTree.

Uses WebP image format

WebP is a relatively new open-source image format that provides superior (around 26% compared to PNG's [25] lossless or revocable and lossy or irrevocable compression.

WebP also supports alpha-channel compression (transparent layer) which could effectively even more reduce the image size.

The global coverage in browsers is around 77% [26] and it is natively supported on the HTML specification. Using a picture element one can specify a fallback in case the browser in question does not support WebP. This means that firstly, the browser will attempt to load WebP and if it is not supported the next image provided will be loaded instead (e.g. JPG, PNG, etc,.). Hence, it is boldly recommended to use it over PNG, GIF or JPG formats.

The script first gathers all img elements in the document object by using the Page class evaluate method, and then checks them side by side with their MIME type present in the response object.

This is shown in code snippet 5.

```
export class CollectImages extends Collect{
    static async afterPass(passContext:any){
      const fetchImages = async ()=> {
                  return await page.evaluate(()=>{
                                                                                                  return
      Array.from(document.body.querySelectorAll('img')).map(img => {
                          const attrObj:any = {}
                              img.getAttributeNames().forEach(name=>{
                                  attrObj[name] = img.getAttribute(name)
                              })
                              return attrObj
                          })
                      })
              }
```

Code snippet: 4. Extracted from server/src/collect/images.collect.ts from the project's code repository.

6. System architecture

The architecture proposed in this project consists of a back-end with the business logic tier and a front-end with the presentation tier.

The website sustainability audits will be designed as a microservice with a public exposed API so web clients can make post requests in order to run individual audits.

6.1 Server-side

The core of the sustainability audits consists of different layers or stages comprising the enqueueing of jobs to the parsing and evaluation of several collectors.

Pre-enqueuing

The server initially checks the URL validity in two ways:

- URL regular expression validator
- HEAD Request

It is hard to come up with a regular expression that successfully rejects invalid URLs and passes the valid ones. Besides, checking a string against a long and complex expression may impair code performance and it is not an option for recurrent checkings. Instead it is preferable to create a short expression to see if the string matches the well-known http and domain formats of web pages. After that, it may also check the URL not to be included in a URL blacklist comprising, for instance, the self hosting site or localhost.
At this stage. a passing URL might still not be a real-world URL, one that represents a reachable host.
A fast and reliable way of fetching the HTTP headers of a webpage without downloading the whole body, which would be a waste of resources in this case, is to make a HTTP HEAD request. If the response code is OK (i.e. in the series of 200 to 299) then the URL is marked as valid and ready to be enqueued.
This step is useful to check networking state and to prevent a slow network from running a job.

Enqueueing

When the job is enqueued it is automatically assigned a unique ID valid throughout all the process.
In a runner instance, a new worker is listening for new jobs and processing a configurable number of them at a time. This is controlled by the maxConcurrency configuration option in the config file.
Changing this setting is a trade-off between performance and delivery times.

Dequeuing

The strategy followed to dequeue jobs is known as LIFO which may also be configured to FIFO.

Processing

Each job has its own commander instance which controls and communicates the collectors, parsers and evaluators.

Initially, the worker handovers a job to the commander instance. This creates two parallel threads, one for navigation and another for collecting the traces. The navigation consists of launching a headless chromium browser controlled by the third party library, Puppeteer. Collectors are scripts whose aim is to collect valuable metrics from the navigation and transform them to traces that contain the processed information. Collectors run in parallel threads and asynchronously because some of them have to wait for navigational events to dispatch. Transfer, css, js or images are examples of existing collectors.

Then, a parser role is to catch errors and transform collected pieces of information to a format consumable to audits (i.e. traces). Audits are splitted into two categories and also running in parallel and have their own format. They include a meta function containing the audit's id, description, failure and success titles, thresholds, etc., and a method to evaluate traces regarding the established thresholds and eventually, a score is calculated which may be in a binary or numeric format. Optionally and depending on the audit typology, the testimonials can be appended to the output so as to provide extra information and reasoning about the result.

Once all audits are finished, their output is re-parsed in the commander instance and a report object is built comprising a meta block containing a global score from 0 to 100 and an audit block divided into server and design categories and sorted by passes or failed audits. A score is also computed per category.

The full process is depicted at figure 7.

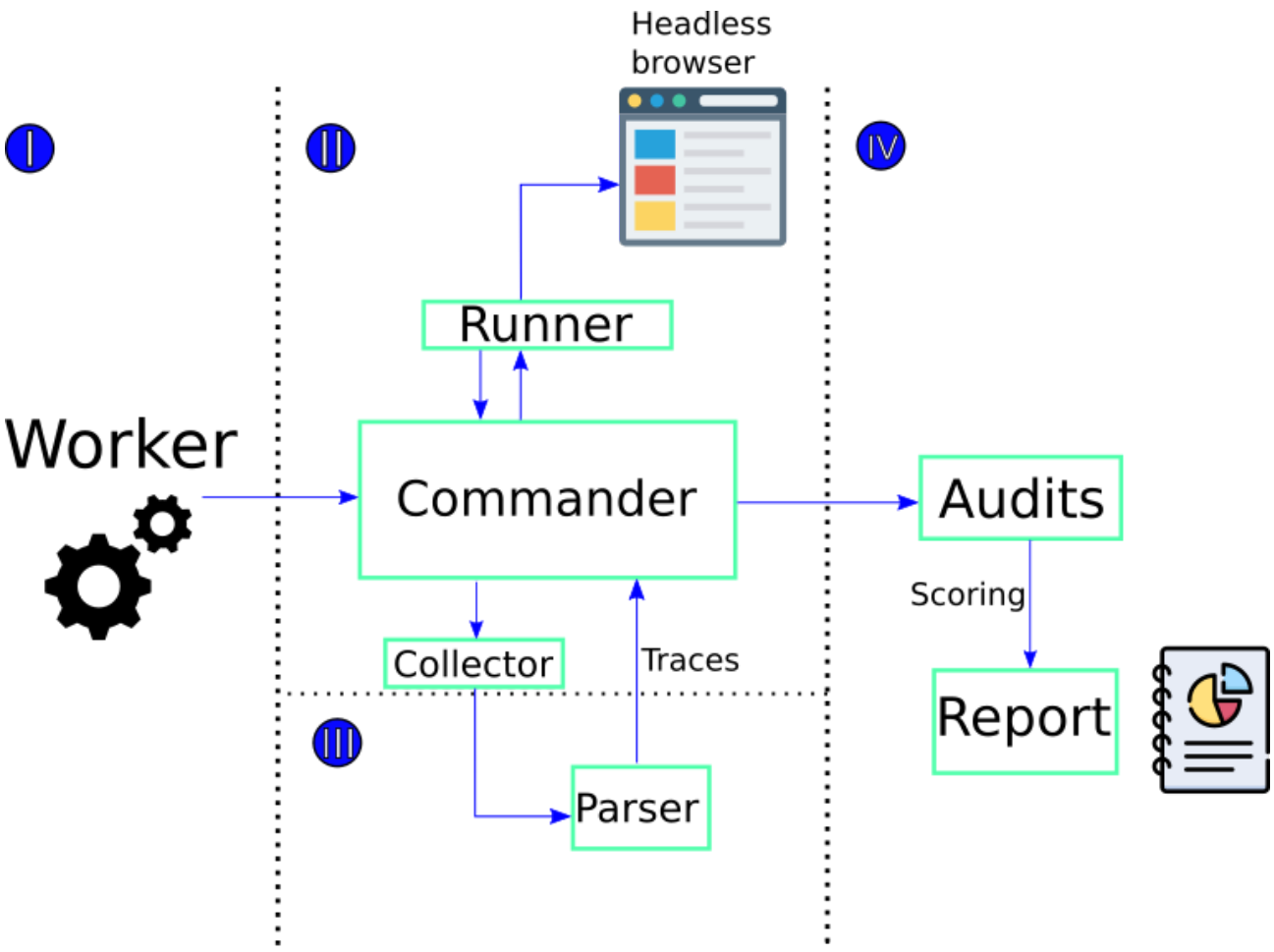

Figure 7: Audits architecture. Credits: Own work. Image credits: Browser svg: From DinosoftLabs, Report svg: From Freepik in flaticon.

Configuration

The config object consists of:

- Puppeteer
  - Max concurrency
  - Puppeteer options
  - Max timeout
- Connection
  - Max throttle
  - Max navigation time
  - Emulated device (user agent, viewport)
  - Locations
- Audits
- Report

- Scoring weights
- Format

6.2 Client-side

On the client side, a user interface will be designed for searching the desired URL in which the whole evaluation will be made upon and also to visually render the results.

Reactive programming principles were applied to the HTML form and HTTP workflow due to its known efficiency over traditional approaches.

The client application itself it is quite simple and encompasses two parts:

- The search form
- The rendering of the report

Search form

It consists of a reactive form in which changes are listened asynchronously following reactive programming principles. A simple URL regular expression ensures the string validity before ever attempting to establish connection with the server endpoint. Also, error messages are shown in a user-friendly manner concerning both client and server problems.
Upon loading state it displays a spinner which is effectively removed when the server has dispatched a response.

Report rendering

It is crucial to deliver json data in a proper way, one that is consumable in the client side, otherwise coding may become inefficient and repetitive.
This step has remarkably introduced order and cleanliness to the report, sorting audits by their passing or failure states and considerably optimising the file size.
As the report itself included numeric values to represent a score, it was thought to be useful to represent those in the most visual way as possible. As a result, a gauge gadget was developed using SVGs.

7. Development environment

7.1 Server-side

The project was developed using NodeJS, a JavaScript runtime environment built on Chrome's V8 JavaScript engine that implements the reactor pattern, a non-blocking, event-driven I/O paradigm.
The programming language of choice was Typescript which is a subset of Javascript but typed.
Typescript learning curve is quite pronounced at the beginning, the syntax may look abstract but with time it becomes a comfortable language to work with.

One of the particularities of Typescript is that it cannot directly run on the browser so it needs to be transpiled back to a version of ECMAScript, the estandarised versions of JavaScript maintained by the ECMA whose latest version is the ES2019.
During Typescript development it is therefore necessary to transpile testing code with ease. Many existing libraries offer this feature but the most popular is the ts-node npm package altogether with a utility that monitors for any changes in the source and automatically restarts the server, Nodemon. Recently, a combination of both have been released: ts-node-dev which will be used instead.

It has also been highlighted the necessity of running the in-memory datastore library Redis. To get it installed locally one may follow the instructions guide in their documentation, but it is not straightforward as it may require third-party installation software. As such, one can instead run a local container on Docker of the latest redis image and simply avoid the hassle.

Code being programmatically being developed needs to be kept in a safe place. One of such places is GitHub which provides a whole new system versioning universe, gently satisfying all developer's needs with a local command line tool: git. All the code can be splitted up following closely the development steps and may consist of several branches. Github platform also serves as an encountering place between developers which may be interested in open-source software and contributes to its development.

Finally, having the code being developed with the type-safe Typescript, transpiled and reloaded with ts-node-dev, pushed and kept at a GitHub repository, it may seem that the technology stack is satisfied, but not far away from that there is still one piece of software that is missing. The API needs to be tested in the localhost environment and therefore it is necessary to have a tool for transferring data via the HTTP protocol. For the ongoing project curl is the chosen one which entirely runs in the terminal. If looking for a more user friendly software capable of also performing complex tasks, Postman may be the choice.

7.2 Client-side

The client-side of the project which consisted of a user interface to run audits and render the results, was developed with Angular 9 a client-side Javascript framework. Angular is known to have a high learning curve, which may be true accounting for the fact that it ships with Typescript and works with the MVC pattern. Ranging from big complex apps to single page applications have been built in Angular. In this project, what triggered its usage was primarily due to having acquired previous experience with it.

8. Planification

Whilst the core idea of the sustainability audits has remained unchanged, the project and the planification encountered unexpected difficulties with regards to its server-side realisation.

The complexity was overwhelming especially when dealing with the chrome network protocol and controlling the headless browser in an asynchronous fashion.

Likewise, the weight given to the server-side was higher in comparison to the client-side, and the respective timings were adjusted.

For a wider perspective of the project's planification, please refer to the following resource [27] and the figure 8.

The project day-to-day workflow was also represented and managed in a dedicated Trello board which was shared with the consultant.

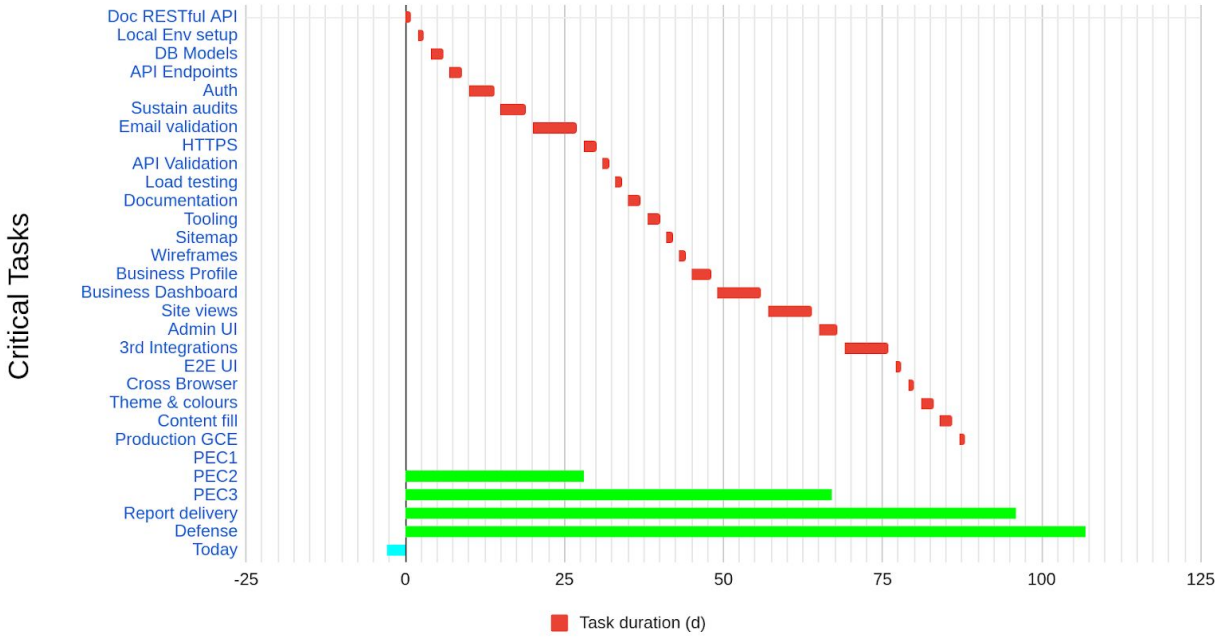

Figure 8: Gantt chart comprising the key activities, key dates and deliveries.

9. Dependencies

The main dependencies used in this project are the following

- Puppeteer
  - A node library which provides a high-level API to control a Chrome or Chromium browser over the DevTools protocol.
- Puppeteer-core
  - A node library built upon Puppeteer but enabling concurrency via workers threads.
- BullMQ
  - A node library for managing distributed jobs and messages working with Redis.
- Express
  - A node web application framework used to expose the REST API.
- NGINX
  - A web server used also for reverse proxying or load balancing.

10. Prototypes

In this chapter, Hi-Fi prototypes covering the user flow will be shown along with an explanation.

10.1 Hi-Fi

The app was designed to be responsive (i.e. adaptable to different viewport widths).

This can be observed in figure 9 where the mobile and desktop app versions are shown.

Also the visual output resulting from a user flow is depicted in figure 10.

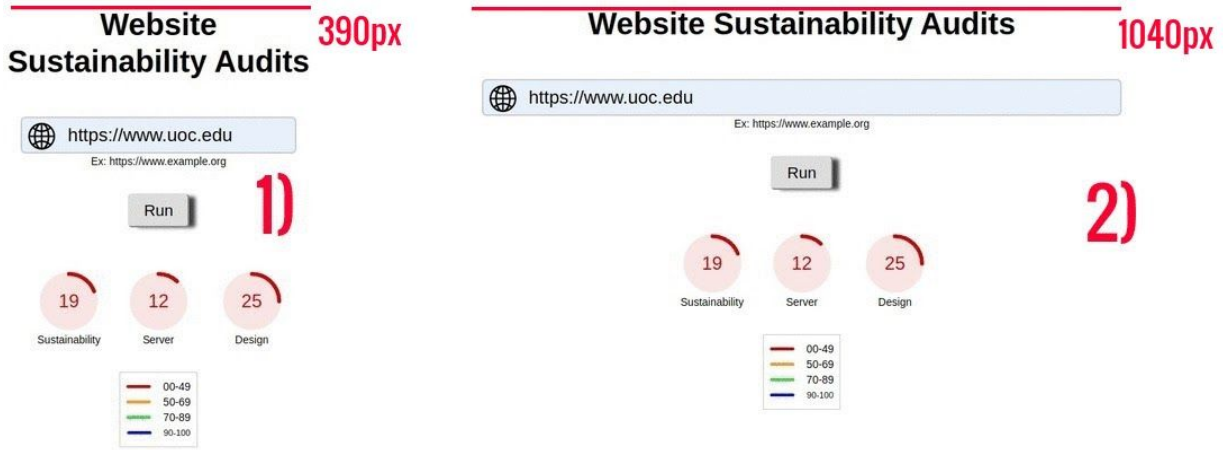

Figure 9: App is responsive. 1) Mobile version of the app at 390px viewport width. 2) Desktop version of the app at 1040px viewport width.

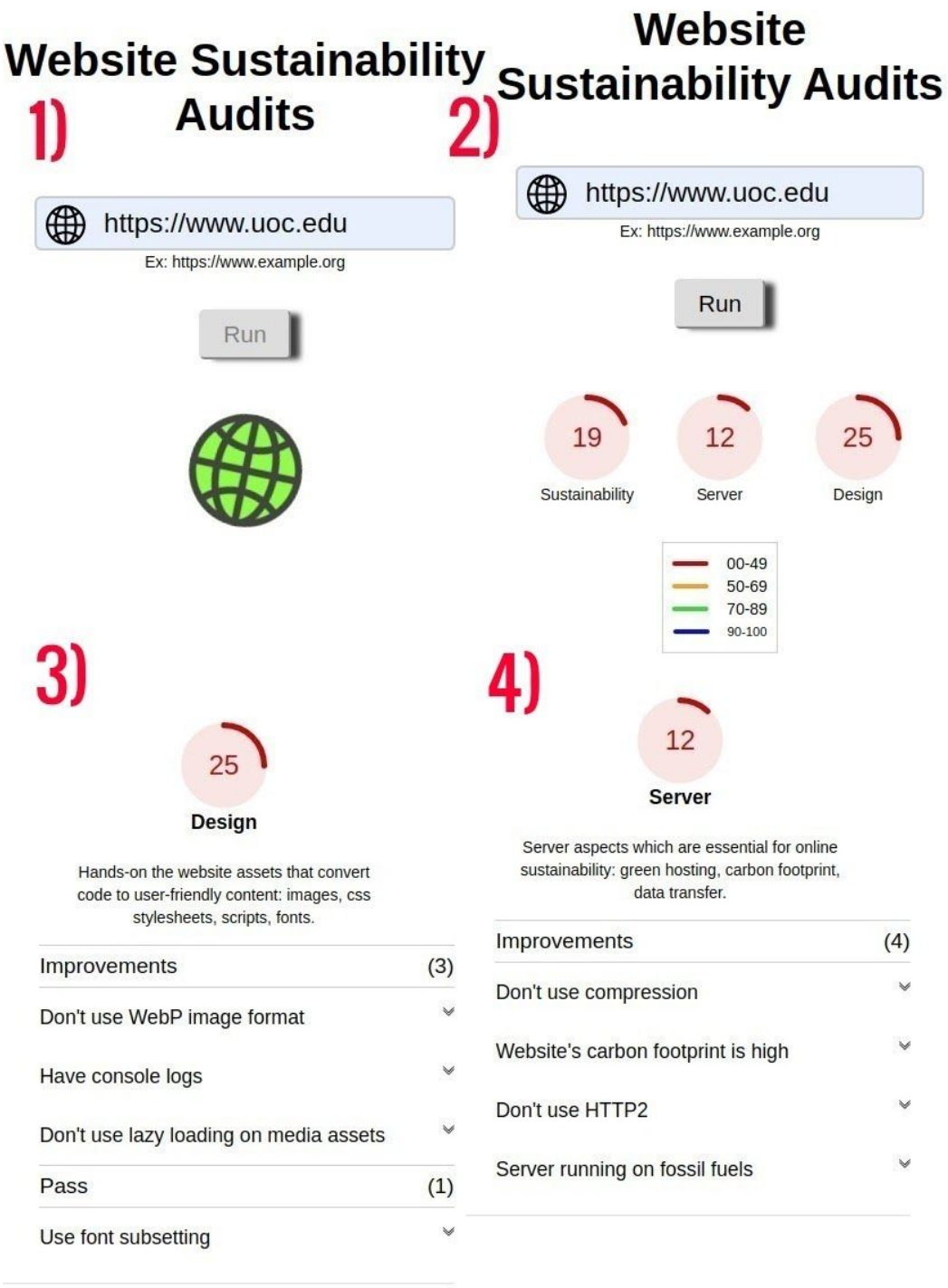

Figure 10: Hi-Fi prototype of the client-side web application. 1) App is running a job and displays a spinner during the job process. 2) The job has completed and three different gauges with a score are displayed for the global sustainability, server and design categories. 3) & 4) Design and server audits respectively, divided into improvements or failed audits and passed audits. All the audits contain extra information in the dropdown section.

11. Usability

The term usability in human-machine interfaces generally refers to the effective human interaction with a website or service.
This is mainly achieved by displaying the information via the device screen and enabling user interaction via the device peripherals (i.e. mouse and keyboard).
The reason usability matters in application design is because most of the time one would be building interfaces and services to target a user need.

11.1 Application goals

Historically, sustainability has been supposed beforehand as it was intention driven. Sustainability was not given to the highest bidder but it did make sense to place it in the mouth of those that for instance, promised to cause little damage to the environment.
Contrarily, the digital sustainability audits aim to provide an objective evaluation on digital sustainability practices. To reduce ambiguity and become widely acknowledged this has to become a standard in the Internet industry.
It should not behave like a static metric, instead it is an indicator that opens up the possibility for website's architects for continuous improvement.
And surely, this might be wrapped up in a service that not only drives the workflow but also offers certificates as a solid proof of sustainable practices.

11.2 Personas

Personas is a useful technique to get a clear image of a target customer. Data used for this purpose should have previously been collected in a research study so it is reliable enough for making solid assumptions around it.
Although this is the best practice, when there is not direct access to the data a workaround is to describe a customer situation and how he would like those problems to be addressed.
This is depicted in figure 11.

Hendrick

age: 32
residence: Paris
education: Software engineering
occupation: Dev Ops

My boss wants to build a clean website so I'm using website sustainable audits to track my progress and get a certificate.

Comfort With Technology

INTERNET

SOFTWARE

MOBILE APPS

SOCIAL NETWORK

Criteria For Success:

A sustainable website and valid certificate

Needs

- Detailed information on carbon emissions
- Ability to keep track of changes
- Recommendations and documentation

Wants

- High granularity of the indicators
- Easy to setup
- Quick results

Values

- Accuracy and reliable audits

Fears

- Failure to obtain a certificate
- I'm using a CMS to build a website but I don't have full access to low level settings

Figure 11: Persona profile describing the most relevant features of the target customer.

12. Security concerns

The purpose of digital sustainability audits is to showcase sustainability in the web but also to let developers iterate over their code following the audits recommendations and eventually, building more sustainable websites.
Given the test and improved personality of the audits the information that is exchanged with the server (especially the report object) should not contain sensitive information more than a set of metrics which anyone with the proper tools can extract from visiting a website.

Nonetheless, HTTPS was enabled both in the client and server. Firebase hosted the web application and automatically dealt with TLS. On the server side it was necessary to generate a certificate authority certificate, in this case Let's Encrypt and properly setup SSL on NGINX server.

Also CORS was configured on the web host thus preventing non-authorized parties from using the RESTful API. This was again handled in the NGINX configuration by adding the full set of allow cross-origin request directives.

13. Testing

In order to check how the server would perform in a real-world scenario, ad hoc testing was done with the nodejs library Artillery.

Setting up the testing suite in table 5, the test consisted in the following phases;

- 1 job / 10 s over 1 minute
- 1 job / s over 5 seconds
- 1 job / 5 s over 2 minutes

Name	**Details**
AWS EC2 t3a.micro instance	2 vCPUs 1 GiB Mem 10% baseline performance per vCPU Network up to 5Gbps
Server config	max concurrency 10 max navigation time 30s max timeout 90s max throttle time 5s

Table 5: Test environment.

In order to recreate a real-world scenario where a client would be looking for real websites, a list of 500 well known URLs have been obtained and processed in javascript. Therefore, a random URL was runned per job.

The results are shown in table 6.

Name	**Value**
Scenarios launched	35
Scenarios completed	35
Request completed	35
Mean response/sec	0.19
Response time (msec)	min: 408.3 max: 21812.6 median: 5478.3 p95: 16455.4 p99: 21812.6

Codes	200: 28 400: 7
Workers spawned	4
Errors	TypeError: Cannot read property 'compressedSize' of undefined MaxListenersExceededWarning: Possible EventEmitter memory leak detected. 11 completed listeners added to [QueueEvents]

Table 6: Test results.

The tests performed really well in the t3a.micro instance with a 4 maximum number of workers and a median response time of 5400 msec. 28 out of 35 requests were issued with 200 status code and 8 of them with 400 status code which meant the websites were unreachable or down. The maximum job rate was 1 job /sec so in the NGINX configuration the request limitation was set to that value. In order to prevent the same user from sending multiple requests, a limitation by IP connection was also set to 1 request for IP at the same time. The errors were fixed by properly handling undefined in compressedSize property and by incrementing the number of event simultaneous event listeners.
Finally, the configuration option that controlled the maxim number of workers was set from 10 to 4 as the test suggested.

14. Installation guide

In order to have a local copy of the software, please follow the next points:

1) Refer to the project repository code in GitHub [28] and clone the stable branch.
2) Run npm install on the local folder. This will download all dependencies listed in package.json, including a chromium browser compatible with the latest Puppeteer version (~123MB).
3) Redis is needed. Download the library from source or docker it. Make sure it is running on port 6379.
4) Run the script npm run dev, which will launch the server listening at localhost:7200. Set an environment variable to PORT to change the port value.
    a) Test the server's health with a GET request at /health,
    b) or post a new job with a POST request at /service/add including a JSON body param 'url' containing the corresponding URL to audit.

    Feel free to change the configuration values to fit the needs.

In order to test the whole implementation (server and client) refer to the following URL: https://susaudits.web.app/ and submit a new job.

.

15. Further studies

Digital sustainability audits is a beta release. It is stable for running a low to medium load of requests and its release is experimental. Further studies and features may be added, some of those listed next.

- Applicability of audits
  - Currently an audit can only pass or fail, but sometimes a website does not meet the audit requirements and therefore should not be applied.
- Developer mode
  - In beta release everyone can audit whatever website. This makes sense at this point because it is still being widely tested expecting to spot errors quickly and become more robust soner. However, in the future with the application gaining traction in the market it will be a requirement to demonstrate the capability to change a website code. Running a job has an environmental cost associated, at least as high as downloading the entire website, but running a job and not making any change to it is not a sustainable practice.
- Run unit and integration tests
  - Testing is essential not only to ensure not breaking any functionality upon new changes but also to make sure the application is doing what is expected from it.
- More audits
  - In beta release 8 audits (4 server, 4 design) have been developed. However, there is still room for improvement and making new audits targeting at: cache practices, digital sobriety (limiting resources), etc.
- Database layer
  - It makes sense to store audit results or at least the most predominant results in a database for debugging purposes and assessing what is the interest generated.
- Publish work
  - The why and how of the project has to be extensively documented and shared among publishing platforms.

16. Market segment

The envisioned web application described in this project, will provide the ability to interact with the auditing server and display the results.
This sort of service is generally defined as SaaS or in other words: the service hosts the application, makes it available to customers and bears the related infrastructure expenses in exchange of a subscription fee proportional to the access level.
The SaaS market has been following an exponential growth over the last 12 years as depicted in the figure 12, and disregarding the pandemic-driven economic crisis happening right at this moment, the trend will most probably keep on rising.
This is mainly explained by the decoupling mentality that the Internet has been experiencing in the last years. The client-side is now more transparent, light-weight, decoupled and it's hard to predict where an application is being served from.
Nowadays, SaaS provides lighter solutions that allow customers to swiftly test powerful built-in functionality without having to deal with the system management complexity and that is a considerable advantage in its own.

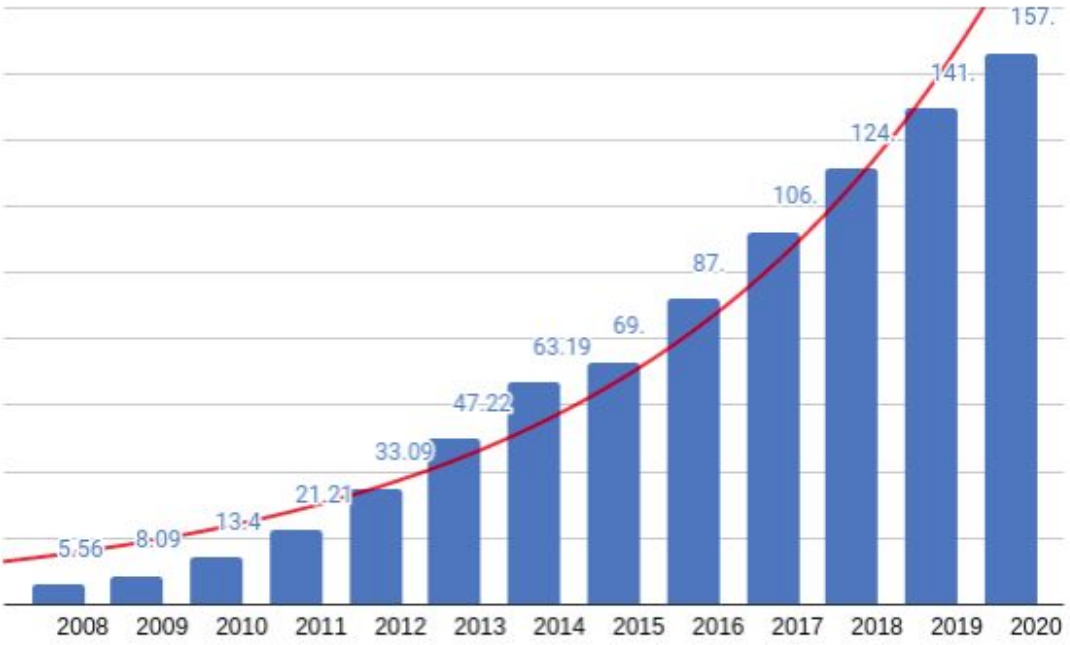

Figure 12: Total SaaS market from 2008 to 2020 in billions of US dollars. Last 4 years are predicted. Source: Statista.

16.1 Competitors

Some companies offer a rather simplistic view of what they call the digital carbon footprint for a given website. An API triggers a job, which takes a website URL and makes a network request to store the static transfer size of downloaded files. Then the total transfer size or the total page weight is compared to its carbon equivalent in grams of CO2 per byte.
In the best scenario, they help to raise awareness on the climate emergency among non-savvy tech people by also comparing this value to well known daily situations (e.g how many km in a standard car one can travel or how long one can keep a bulb on, etc.).

In the worst scenario, they leave out important aspects that matter such as website's cache practices, dynamic rendered content, server location, CDN's, etc., and they do not provide any guidelines regarding how to effectively take action. Therefore, they are far from building a solid standard.
The table 7 contains the most relevant information regarding what already exists in the market. Additionally, three indicators were estimated regarding accuracy, features and popularity.

Company	**Bus. Model**	**Start Date**	**Calc. method**	**Value**	**Code**	**Accuracy (0-3)**	**Features (0-3)**	**Popularity (0-3)**
WholeGrain	Ltd, UK	~2012	Document transfer size	Website carbon audit (gCO2/view)	https://gitlab.com/wholegrain/carbon-api-2-0,	1	1	2 (Top SEO ranked)
The Shift Project	NPO	~2001	Network transfer while navigating	Carbonalyser Add-On	https://github.com/carbonalyser/Carbonalyser	1	2	1
The Green Web Foundation	NPO	~2013	Own dataset for green hosts	Audits on demand CO2 API	https://github.com/thegreenwebfoundation/co2.js	1	1	2 ~400 weekly downloads

Table 7: Current competitors.

17. Viability

In order to assess the project viability three major factors will be considered.

17.1 SDGs

SDG's are the acronym for Sustainable Development Goals. They represent the collection of 17 global goals to achieve a better and above all a more sustainable world. They are part of the UN Resolution 70/1 or also known as the 2030 agenda and they are intended to be achieved by the year 2030. They are depicted in figure 12.

Figure: 12 The 17 SDGs. This project targets the 12 and 13 goals; responsible consumption and production and climate action, respectively. Source: UN.

The involvement of relevant stakeholders into funding SDGs initiatives, proves the vested interest of political agencies to support the 2030 agenda. Since the apparition of the SDGs, a wide range of public and private funds (i.e. Cisco Global Problem Solver 2020, Bezos Earth funds, etc.) have appeared and in this is yet expected to increase in the following years.

As seen in the figure 12, this project strongly aims to meet the 12th and 13th goals. There would be a substantial inflow of companies engaged with the benefits of developing their operations on-line. And with the politico-economic system having its sights on the 2030 Agenda, this would be an exploit factor for emerging businesses which would lack the knowledge and mostly the time for a deep code review of their solutions. Therefore, the digital sustainability audits will ensure the required sustainable scope and ease the transition.

18. Conclusions

With hard times to come in a complex and dramatic climate change scenario, the Internet is liable to follow sustainability practices for the common good. Digital sustainability audits have become a must for existing and new coming websites, to showcase the essentials for sustainability and to motivate a wave of greenness and environmental awareness in the Internet.

The project's goals were fulfilled: a maintainable and logical structured RESTful API was developed, mastered the chrome developer tool network protocol for launching and controlling the lifecycle of a headless browser and a scalable system capable of processing multiple jobs in parallel. Also, a user friendly website application for submitting new jobs and displaying the results.

Yet, the project has still to evolve and see new audits implemented and engage the open-source community to contribute and create new existing and robust iterations on the code.

Now is the time for taking real action on climate change and web developers' role is crucial. Contemporary developers of the green Internet's paradigm shift have to rise up and above all, have to make sure sustainability is preserved in all of their projects.

Appendix 1. Project deliveries

Project's main repository source code: [28].

Appendix 2. Calculations

Calculation 1. Digital carbon footprint

$$C_F(x,y,z) = \sum_{i=0}^{N} \frac{E_{Si}(x) * E_I(y) * C_I(z) * D_T}{1024^2} \quad [g\ CO2eq/\ 100\ visits] \tag{1}$$

$$E_S(x) = \{E_S \ \ if\ x = 1\ \}\ ;\ \{U_S\ if\ x = 0\ \} \tag{2}$$

$$E_I(y) = \{E_{IC} \ \ if\ x = 1\ \}\ ;\ \{E_{IC} + E_{ID}\ if\ x = 0\ \} \tag{3}$$

where:

i is the current request

N is the total number of requests

$C_F(x,y,z)$ is the Carbon footprint in $g\ CO2eq\ /\ 100visits$

$E_S(x)$ is the Encoded transfersize in $bytes$ for $x = 1$ if the value is available or $x = 0$ otherwise.

U_S is the Uncompressed transfersize in $bytes$.

$E_I(y)$ is the Energy intensity of the Internet in kWh/GB for $y = 1$ if the server is green hosted or $y = 0$ otherwise.

E_{IC} is the Energy intensity of the Internet's core network.

E_{ID} is the Energy intensity of an average data centre.

$C_I(z)$ is the Carbon intensity in $g\ CO2eq\ /\ kWh$ for z location.

D_T is the Daily traffic of an average website in $visits/day$.

The table 1A shows the variables values and their sources.

Variable	Value	Source
E_{IC}	0.052	[14]
E_{ID}	0.06	[14]
$C_I(z)$	JSON	[17]
D_T	100	None

Table 1A: Variable values and sources in the carbon footprint calculation.

Appendix 3. Glossary

(In order of appearance)

RESTful API: REpresentational State Transfer Application Programming Interface

ICT: Information and Communication Technology

HTTP: Hypertext Transfer Protocol

CDP: Chrome Devtools Protocol

ARPA: Advanced Research Projects Agency

DoD: Department of Defense

TCP/IP: Transmission Control Protocol / Internet Protocol

MILNET: MILitary NETwork

Kbps: Kilobits per second

CERN: European Organization for Nuclear Research

HTML: HyperText Markup Language

WWW: World Wide Web

URL: Uniform Resource Locator

UE: European Union

GDPR : 2016/679 General Data Protection Regulation

IP: Internet Protocol

GHG: GreenHouse Gases

IPCC: Intergovernmental Panel on Climate Change

COVID-19: COronaVIrus Disease

CT: Communication Technology

CPU: Computer Hardware Unit

MIME: Multipurpose Internet Mail Extensions

GZIP: GNU Zip

AJAX: Asynchronous JavaScript and XML

IT-devices: Information Technology devices

(kWh)/GB: kilowatt per hour per gigabyte

CBA: Consumption Based Accounting

HTTP2: Hypertext Transfer Protocol 2

HTTPS: Hypertext Transfer Protocol Secure

NPM: NodeJS Package Manager

JS: JavaScript

CSS: Cascading Style Sheets

PNG: Portable Network Graphics

JPG Joint Photographic Group

SEO: Search Engine Optimisation

AST: Abstract Syntax Tree

LIFO: Last In First Out

FIFO: First In First Out

SVG: Scalable Vector Graphics

ECMA: European Computer MAnufacturers

MVC: Model View Component

Hi-Fi: High Fidelity

TLS: Transport Layer Security

SSL: Secure Sockets Layer

JSON: JavaScript Object Notation

SaaS: Software as a Service

CDN: Content Delivery Network

Appendix 4. References

[1] N. Jones, 'How to stop data centres from gobbling up the world's electricity', *Nature*, vol. 561, no. 7722, pp. 163–166, Sep. 2018, doi: 10.1038/d41586-018-06610-y.

[2] C. L. Weber, J. G. Koomey, and H. Scott Matthews, 'The Energy and Climate Change Implications of Different Music Delivery Methods', *Journal of Industrial Ecology*, vol. 14, no. 5. pp. 754–769, 2010, doi: 10.1111/j.1530-9290.2010.00269.x.

[3] 'File:Internet Key Layers.png - Wikimedia Commons'. https://commons.wikimedia.org/w/index.php?curid=36916050 (accessed May 12, 2020).

[4] B. J. Fogg and G. E. Fogg, *Persuasive Technology: Using Computers to Change What We Think and Do*. Morgan Kaufmann, 2003.

[5] J. S. Dryzek, R. B. Norgaard, and D. Schlosberg, 'Climate Change and Society: Approaches and Responses', *Oxford Handbooks Online*. 2011, doi: 10.1093/oxfordhb/9780199566600.003.0001.

[6] 'Scripps Institution of Oceanography, UC San Diego |'. https://scripps.ucsd.edu/ (accessed Jun. 01, 2020).

[7] 'Greenhouse gas concentrations | Copernicus'. https://climate.copernicus.eu/ESOTC/2019/greenhouse-gas-concentrations (accessed Jun. 01, 2020).

[8] 'Greenhouse gas concentrations | Copernicus'. https://climate.copernicus.eu/ESOTC/2019/greenhouse-gas-concentrations (accessed Jun. 01, 2020).

[9] C. Le Quéré *et al.*, 'Temporary reduction in daily global CO 2 emissions during the COVID-19 forced confinement', *Nat. Clim. Chang.*, pp. 1–7, May 2020, doi: 10.1038/s41558-020-0797-x.

[10] A. Andrae and T. Edler, 'On Global Electricity Usage of Communication Technology: Trends to 2030', *Challenges*, vol. 6, no. 1. pp. 117–157, 2015, doi: 10.3390/challe6010117.

[11] 'The shift project ICT Report'. https://theshiftproject.org/wp-content/uploads/2019/03/Lean-ICT-Report_The-Shift-Project_2019.pdf

[12] 'How does it work? - Website Carbon Calculator', *Website Carbon Calculator*. https://www.websitecarbon.com/how-does-it-work/ (accessed May 20, 2020).

[13] E. Masanet, A. Shehabi, N. Lei, S. Smith, and J. Koomey, 'Recalibrating global data center energy-use estimates', *Science*, vol. 367, no. 6481, pp. 984–986, Feb. 2020, doi: 10.1126/science.aba3758.

[14] J. Aslan, K. Mayers, J. G. Koomey, and C. France, 'Electricity Intensity of Internet Data Transmission: Untangling the Estimates: Electricity Intensity of Data Transmission', *J. Ind. Ecol.*, vol. 22, no. 4, pp. 785–798, Aug. 2018, doi: 10.1111/jiec.12630.

[15] B. Tranberg, O. Corradi, B. Lajoie, T. Gibon, I. Staffell, and G. B. Andresen, 'Real-time carbon accounting method for the European electricity markets', *Energy Strategy Reviews*, vol. 26, p. 100367, Nov. 2019, doi: 10.1016/j.esr.2019.100367.

[16] 'The Green Web Foundation | Together towards a green web'. https://www.thegreenwebfoundation.org/ (accessed Jun. 06, 2020).

[17] tmrowco, 'tmrowco/electricitymap-contrib', *GitHub*. https://github.com/tmrowco/electricitymap-contrib (accessed May 21, 2020).

[18] 'Desmos | Graphing Calculator', *Desmos*. https://www.desmos.com/calculator/p3z5exil8h. (accessed Jun. 03, 2020).

[19] 'Chrome DevTools Protocol'. https://chromedevtools.github.io/devtools-protocol/tot/Network/#event-loadingFinished (accessed Jun. 03, 2020).

[20] ‘Magento: Page Weight’. https://httparchive.org/reports/page-weight?lens=magento#bytesImg (accessed Jun. 02, 2020).
[21] ‘HTML Standard’. https://html.spec.whatwg.org/multipage/urls-and-fetching.html#lazy-loading-attributes (accessed Jun. 02, 2020).
[22] ‘Can I use... Support tables for HTML5, CSS3, etc’. https://caniuse.com/#search=lazy%20loading (accessed Jun. 02, 2020).
[23] ‘Puppeteer - scroll down until you can’t anymore’, *Stack Overflow*. https://stackoverflow.com/questions/51529332/puppeteer-scroll-down-until-you-cant-anymore (accessed Jun. 03, 2020).
[24] filamentgroup, ‘filamentgroup/glyphhanger’, *GitHub*. https://github.com/filamentgroup/glyphhanger (accessed Jun. 03, 2020).
[25] ‘Lossless and Transparency Encoding in WebP | Google Developers’, *Google Developers*. https://developers.google.com/speed/webp/docs/webp_lossless_alpha_study (accessed Jun. 03, 2020).
[26] ‘Can I use... Support tables for HTML5, CSS3, etc’. https://caniuse.com/#feat=webp (accessed Jun. 03, 2020).
[27] ‘project-organization’, *Google Docs*. https://docs.google.com/spreadsheets/d/1MOGZNLvKoicT_7G2ANVhvFzxvJb1faqxxq8cj5GTScI/edit?usp=sharing&usp=embed_facebook (accessed Jun. 04, 2020).
[28] ‘Digital Sustainability Audits: Source Code Repository’. https://github.com/digital-audits/sustainability/ (accessed Jun. 06, 2020).