university of groningen

faculty of science and engineering

A Model for the Total Carbon Footprint of a Cloud-based Software as a Service Provider and its Customers

Master's Thesis

Author
Richard Westerhof
r.s.westerhof.2@student.rug.nl

Date
22$^{\text{nd}}$ of August, 2022

Supervisor
Vasilios Andrikopoulos

Second Supervisor
Daniel Feitosa

External Supervisor
Christos Bantis

Secondary External Supervisor
Richard Atherton

Abstract

Cloud computing has gained a lot of popularity over the past years, to the point where most companies now make use of cloud services in some way. On the surface, cloud computing seems like a great way to offload concerns to the cloud service provider. Such concerns include server management, logistic concerns, and, most importantly for this project, energy consumption and carbon emissions. However, moving carbon emissions from one place to another is not a solution to the global carbon problem.

In cooperation with BT Global Services, this project explores the details of carbon emissions in multi-tenant data centers, which includes how emissions can be estimated using which measured metrics, who is ultimately responsible for these emissions, and how emissions can be divided across tenants. To this end, a model is created which calculates the total carbon emissions for each tenant in a data center. Building on this model, a sustainability report generator presents the information at different levels of detail to the involved stakeholders. Qualities of the resulting sustainability report such as clarity, understandability and auditability are evaluated in collaboration with industrial partners.

Contents

Chapter 1

Introduction

Over the past decade, cloud computing has become very prominent. Cloud computing is a means of providing computing services over the internet [34]. These provided services can have varying levels of abstraction. Infrastructure as a Service (Iaas) has the lowest level of abstraction; it gives users the infrastructure they need to run any software they want. Platform as a Service (PaaS) is the middle ground of abstraction; it allows users to deploy software on an environment tailored to the software. This is often done through using an already set up operating system environment including some tools the user may need. Software as a Service (SaaS) has the highest level of abstraction; it gives the user access to an application already running on the cloud through a thin client. This means the user has very little control, but at the same time this much improves the ease of use.

Most companies make use of cloud services in some way. At the same time, another item which has become much more prominent is the goal to become carbon neutral, and, for some companies, even carbon negative, as is the case with Microsoft[1] for example. Since most services' *gross* emissions (physical emissions produced) cannot be reduced to zero, or cannot yet be reduced to zero, the goal is instead to reduce *net* (emissions not compensated through offset methods) emissions to zero. This is done by offsetting the carbon emissions that a company is responsible for, which can be done through a multitude of ways. For example through the direct use of green energy, but another option is to buy Renewable Energy Certificates (REC) [50], which are a way to ensure that an agreed amount of energy will be generated in a way that is renewable, in other words, zero- (or low) carbon.

The first step in offsetting all produced carbon emissions, is determining how many carbon emissions have been produced. Unfortunately, most cloud service providers do not provide their customers clear insights. However, this has been changing recently as well. Big cloud service providers such as Amazon Web Services (AWS), Google Cloud Platform, and Microsoft Azure have started rolling out dashboards to their customers where they can monitor their energy consumption and estimated carbon emissions. These dashboards are discussed in detail in Section 2.1.

1.1 Cloud Computing and Sustainability

To further introduce the topic of this work, this section contains some general background on the topic of sustainability in cloud computing.

According to Lago et al., there are actually four kinds of sustainability [25]:

- Environmental sustainability had to do with the carbon footprint. To make anything more environmentally sustainable, the carbon footprint should be lowered.

[1] https://blogs.microsoft.com/blog/2020/01/16/microsoft-will-be-carbon-negative-by-2030/

- Technical sustainability has to do with the long term maintenance of a project. When better care is taken to make sure that the project can be continued to be worked on without the need for large refactorings, the project is considered to be better technologically sustainable.

- Social sustainability is about the impact a business has on people. Therefore, a company should stay on good terms with its stakeholders to be socially sustainable.

- Economical sustainability is about the finances. Of course, a business has to make money to keep existing, and in order to be economically sustainable, the business has to not lose money.

In this project, the focus is solely on the environmental aspect of sustainability, therefore, when we refer to sustainability, we actually mean environmental sustainability, unless specifically stated otherwise.

There are many works on the topic of sustainable cloud computing, so many in fact, that there have been several surveys comparing works on this topic over the years. Already in 2011, Beloglazov et al. [8] performed a survey in which they reviewed several works with the topic of sustainability in cloud computing, which mostly focused on application design, energy management, and virtualization. Mastelic et al.[31] performed a similar survey in 2015, which also reviewed works focusing on application design and energy management. A year later, in 2015, Moghaddam et al. [37] performed a similar survey as well, mostly reviewing works which focused on energy management. In 2018, Gill and Buyya [16] performed another such survey, aiming to use a broader focus, also including topics like sustainability metrics, cooling management, and renewable energy, on top of the other topics already focused on by the earlier surveys.

Jagroep et al. [20] noted that while the industry has become aware of the importance of energy efficient hardware, the awareness of the importance of energy efficient software is lacking. Therefore, they conducted a case study in which they investigate how to raise awareness for this topic. Their conclusions are that appropriate stimuli are needed, in their case study this was an energy dashboard, to make people aware of the energy consequences of certain decisions.

Finally, books have also been written which broadly touch the sustainable cloud computing topic, such as the ones by Bitterlin [10] and Smith [43]. These books show that sustainable cloud computing is becoming more than a scientific research topic.

1.2 Three Scopes of Carbon Emissions

Carbon emissions are often categorized in three Scopes, as stated by the Greenhouse Gas Protocol (GHG Protocol) [4], namely Scopes 1, 2, and 3. The Scope which a certain source of carbon emission falls into is determined by how much of a direct consequence the emissions are of a company's operations; Scope 1 are the most direct emissions, whereas Scope 3 are the most indirect emissions.

- Scope 1 includes all emissions which are a direct consequence of a company's operations. This may include things like furnaces, machines, or generators. Therefore, a data center, such as the ones operated by cloud service providers, will usually have very little Scope 1 emissions.

- Scope 2 is dedicated to the emissions caused by the generation of the electricity used by devices. In the case of a data center, these are devices such as servers, network routers and switches, cooling systems, et cetera. This Scope is likely to contain most of the emissions caused by data centers. To help companies report their Scope 2 emissions, Sotos [45] has written a guidance. In this guidance, it is explained that Scope 2 should contain energy consumption in any form, this means that emissions generated from purchased electricity, steam, heat and cooling should all be included.

- Scope 3 is reserved for all other emissions which do not fall into either Scope 1 or 2. This category is labelled as optional in the Greenhouse Gas Protocol, but if the goal is completeness, these emissions can make a large difference and therefore should be acknowledged, though the extent depends on the sector of operations [33]. Similarly to Sotos' work for Scope 2, Bhatia and Ranganathan [9] have written a guidance for Scope 3 emissions. They divided Scope 3 emissions into two sub-categories: Upstream and Downstream emissions. This refers to where in the value chain the emissions occur. Upstream emissions consist of:

  - Purchased goods and services
  - Capital goods
  - Fuel- and energy-related activities (not included in Scope 1 or Scope 2)
  - Upstream transportation and distribution
  - Waste generated in operations
  - Business travel
  - Employee commuting
  - Upstream leased assets

  Downstream emissions consist of:

  - Downstream transportation and distribution
  - Processing of sold products
  - Use of sold products
  - End-of-life treatment of sold products
  - Downstream leased assets
  - Franchises
  - Investments

- As proposed by Matthews et al. [33], a fourth Scope, Scope 4, can be used. The goal of Scope 4 is to take over some of the emissions traditionally placed in Scope 3. This makes both scopes more focused, rather than Scope 3 being a general catch-all for all emissions more indirect than energy consumption. While this proposed Scope would even out the amount of sources in each Scope a bit more, it has not been included in the official Greenhouse Gas Protocol, nor have many companies adopted this Scope into their reports.

Because Scope 3 emissions are noted as optional in the GHG Protocol, Matthews et al. [33] explored how accurately the total carbon footprint can be represented by Scope 1 and 2 emissions, referred to as "tiers" by the authors, for a multitude of sectors. They used an Economic Input-Output Life-Cycle Assessment (EIO-LCA) formula to estimate carbon footprints. This is a formula which can be used to calculate the effect of dependencies between different sectors up to any specified depth. They defined Scope 1 emissions to be all first level emissions in this formula, and Scope 2 to be all second level emissions. Their results showed that only 10% of sectors had 80% or more of their carbon footprint represented by Scope 1 and 2. Finally, they propose to create another Scope, Scope 4, which will include delivery, use, and end-of-life, taken from Scope 3, which will then contain only cradle-to-gate emissions.

Lenzen et al. [27] noted that the LCA calculations only work when the responsibilities are pushed all the way to either end of the production lifecycle; either the very first producer is responsible for all emissions their consumers create by using their product, or the end consumer is responsible for all emissions that have been caused by the production of a product they use. However, it does not make much sense to push the responsibility all the way to one side. Instead, responsibility should be shared, but sharing responsibility leads to double counting. For example, if a company produces a product and reports the emissions from that as its Scope 1 and 2 emissions, and a consumer buys that product and reports it as Scope 3 emissions, then those emissions are double counted. Therefore, Lenzen et al. propose to share the responsibility according to a set percentage; the producer will be responsible for $x\%$ of the emissions, while the consumer will be responsible for $1 - x\%$ of the emissions.

Lenzen and Murray [26] discussed upstream and downstream indirect emissions. Essentially, they argue that any entity that buys a product or service is (partially) responsible for the entire upstream emissions, and any entity that sells a product or service is (partially) responsible for the entire downstream emissions. The further up or down the stream emissions reside, the less responsible an entity is. Therefore, calculating the total emissions is an infinite sum with a finite result.

1.3 Case Summary

BT Global Services is a company that provides cloud services to their customers. One of their products is Cloud Contact Cisco (CCC), which provides call center software through the Software as a Service (SaaS) model. BT's customers use this software for their customer service departments. This service is provided through the use of several data centers worldwide. The end users of this product are the customer services employees of the companies that buy BT's product. These end users are internally referred to as agents.

Now, with three of the biggest cloud service providers, namely Amazon[2], Microsoft[3] and Google[4], starting to offer their customers ways to monitor their carbon footprint in the cloud, see Section 2.1, BT has noticed that its customers are becoming more aware of their impact on the environment, and would like to gain better insight in this. To achieve this, they are curious about the carbon footprint that they are responsible for by using BT's services.

BT aims to satisfy its customer's desires by providing them with a dashboard that will show metrics such as the customer's energy consumption and estimated carbon footprint, quite similar to what Amazon Web Services, Google Cloud Platform, and Microsoft Azure are offering (see Section 2.1). In fact, in cooperation with a third party, they have already created such a dashboard, however, the data that is fed into this dashboard is currently based on power meters that were installed between a small set of servers and the power outlets. The problem with this is that it is not feasible for BT to do this to every single server, network device, and any other device that may be consuming power in each of their data centers, not to mention that this approach also misses other emission sources that do not come from direct energy consumption (in other words, Scope 1 and 3). Therefore, BT is looking for a more thought out approach, that can be implemented using data that they already have access to, for example the usage statistics of their servers.

In order to concretely state the goals of this project, the following research questions are defined. The most important research question of this project is Research question 1.

Research question 1 *How can the Total Carbon Footprint (TCFP) be accurately and fairly attributed to each tenant in a multi-tenant data center?*

This is still quite a general question, but it is the main goal of this project. The reason this is the main research question and the goal of this project is because, while there is some literature

[2] https://dev.to/aws-builders/sustainability-on-aws-reinvent2021-4bk6

[3] https://azure.microsoft.com/en-us/blog/microsoft-sustainability-calculator-helps-enterprises-analyze-the-carbon-emissions-of-their-it-infrastructure/

[4] https://cloud.google.com/carbon-footprint

which describes the factors of the Total Carbon Footprint of an entire data center, there does not seem to be much research done regarding the Total Carbon Footprint of one tenant in a multi-tenant data center. So, by performing this research, the goal is to reinforce this area in the state of the art in the environmental sustainability of cloud computing. In order to answer this main research question, the following, other research questions, Research questions 2 to 5, will need to be answered first.

Research question 2 *Which factors should be included in the calculation of the Total Carbon Footprint?*

Research question 3 *How can the Total Carbon Footprint of a data center be accurately obtained?*

Research question 4 *Who is responsible for which carbon emissions in a data center?*

Research question 5 *How can the Total Carbon Footprint be fairly distributed over all tenants in a multi-tenant data center?*

Combining the answers from Research questions 2 to 5 will provide the answer to Research question 1. Fortunately, Research question 2 can be mostly answered by the GHG Protocol [4], but still needs to be adapted to the realities of cloud computing, and Research question 3 has already been researched in the past, see Chapter 2. Research question 4 on the other hand is still an unsolved problem, and Research question 5 has not been researched in nearly as much depth either.

1.4 Thesis Structure

In this thesis, there has been an introduction to the topic in this chapter. Then, in Chapter 2 we will explore related works on the topic of the sustainability of cloud computing, focusing mainly on the environmental aspect of sustainability. Chapter 3 is about the case study part of this project, which was conducted in cooperation with BT Global Services (BT). This section details why a case study was done, and how it was done, as well as the functional and non-functional requirements for the solution. Then, Chapter 4 focuses on selecting the methods from the background to use in the model for the Total Carbon Footprint and putting them together. Chapter 5 describes the implementation of the model into a program to automatically generate reports for each tenant in a data center, which technologies are used, and the architecture of the program. Next, in Chapter 6, the model and generated reports are validated by interviewing three of BT's customers. In Chapter 7 the project is concluded by giving a short summary and discussing the impact, as well as the limitations. Finally, Chapter 8 discusses some potential future work.

Chapter 2

Related Work

There have been many previous works using many different ways to estimate the environmental impact of cloud computing. In this chapter we will discuss a range of these works sorted into a handful of categories. Section 2.1 describes the dashboards from three large cloud service providers, namely Amazon, Google and Microsoft. Section 2.2 discusses other tools which exist to monitor cloud carbon emissions and energy consumption. Section 2.3 takes a look at other works which describe methods to measure carbon footprint and energy consumption. Section 2.4 investigates works which propose ways to reduce or offset the carbon footprint. Finally, Section 2.5 lists some of the most used cloud simulators that exist.

2.1 Existing Dashboards from Hyperscalers

Three of the biggest cloud service providers (Amazon Web Services, Google Cloud Platform, and Microsoft Azure), collectively known as the "hyperscalers", have all started providing tools to their customers to monitor and gain insight into their carbon footprint from using cloud services.

Amazon Web Services (AWS) [42] offers a sustainability dashboard which has five main categories. The first category shows the total carbon emissions for the customer, and how many emissions they saved by using AWS. The second category is a pie chart of the emissions per region. The third category is a breakdown of emissions per AWS service used, since AWS provides many different services. The fourth category is a line graph of the emissions over time. The fifth and final category is a graph which shows the predicted net carbon emissions for the customer in the coming years. This is based on AWS's renewable energy project roadmap to decrease their net emissions. The time window over which the data is calculated can be changed freely by setting the start and end month. They also have a video on YouTube describing the platform[1]. During this demo they note that AWS only includes their Scope 1 and Scope 2 emissions.

Google Cloud Platform (GCP) has a web page describing some of the features[2] and a YouTube video[3] in which, among demonstrating some other carbon awareness projects, they show the carbon footprint dashboard. The Google dashboard first lists the total carbon emissions of the past year, as well as the current month, followed by the net emissions. Then, in various bar charts, the emissions per month are shown, the emissions by the customer's projects running on Google Cloud, and finally the carbon emissions per service and per region, similar to AWS. Google also has a more in depth web page about their methodology[4].

[1] https://www.youtube.com/watch?v=WqhAnLdg3rg

[2] https://cloud.google.com/carbon-footprint

[3] https://youtu.be/78cGDCGdajE?t=630

[4] https://cloud.google.com/carbon-footprint/docs/methodology

Microsoft Azure [35, 36] has a dashboard with nine different tabs, but unfortunately, only the first three tabs seem to be documented. The first tab shows the carbon emissions over the previous years, as well as the evolution of the net carbon intensity, which shows how much Microsoft is doing to reduce net emissions. The second tab again shows carbon emissions over time, here also broken down by Microsoft Scope 1, 2, and 3, as well as the usage over time, and just like Google and AWS, it also shows the emissions per region and per service. The third tab focuses on the carbon emissions saved by using Azure instead of an on premises data center. It shows the amount of carbon saved by Microsoft's claimed higher efficiency for their data centers, and by Microsoft buying renewable energy. For this comparison, they offer some rudimentary settings to model an on premises data center that is close to the one the customer currently uses. The customer can choose from three different efficiency scales for their own data center, low, medium, and high, and the customer can fill in the green energy percentage.

These dashboards can be used as inspiration for which metrics should appear when considering any carbon emission monitoring solution. However, even if details about the methodology used in the operation have been published, these usually do not go into enough detail about how they actually measure or estimate the total power consumption of IT devices, or the architectures are too different for the methodology to be directly applicable in this project.

It is important to note that, depending on the perspective of the dashboard, that is, whether it is showing the emissions for a tenant or the data center operator, the Scopes change. For example, energy consumed by server devices is a Scope 2 emissions from the perspective of a data center operator, since they purchased the energy. However, from the perspective of a tenant, this is a Scope 3 emission, since it is an emission indirectly caused by the tenant using the services from the data center. Since the mentioned dashboards are tools designed to be used by the tenants, they show the perspective of a tenant.

2.2 Carbon Emission and Energy Consumption Monitoring Tools

Apart from the dashboards from Amazon, Google, and Microsoft mentioned in the previous section, other tools to monitor carbon emissions and energy consumption of cloud services also exist, be it for different platforms or platform agnostic tools. An example of the former is a Python script available on GitHub[5], which can take data from CloudFoundry[6] deployments and do simple estimations based on that data.

An example of a platform agnostic tool, on the other hand, is the CarbonFootprintCalculator[7], which requires the user to manually enter energy consumption. This means that it can even be used for data centers which do not have automated data collection processes, but because of this its calculations are not as in-depth or as specific as those from a tool that can use a wide variety of known and automatically collected data. The website of the Greenhouse Gas Protocol[8] contains a set of spreadsheets which can be filled in and will calculate the carbon footprint. However, their use cases are quite specific and therefore they cannot be used as a general carbon footprint calculator tool in their current state. Furthermore, since these are spreadsheets, interacting with them programmatically is difficult, and thus filling them in would likely be a manual process. Finally, a tool has been developed by ThoughtWorks[9] which combines the data from the dashboards of the three big cloud service providers from Section 2.1, making it a semi-platform agnostic tool. Its layout contains four sections: the first is a graph of the carbon emissions, energy consumption and costs over time, the second shows the amount of flights taken, smartphones charged, and trees grown equivalent to the carbon emissions, the

[5] https://github.com/P-Ehlert/cloudfoundry_footprint

[6] https://www.cloudfoundry.org/

[7] https://github.com/MarosMacko/CarbonFootprintCalculator

[8] https://ghgprotocol.org/Tools_Built_on_GHG_Protocol

[9] https://www.cloudcarbonfootprint.org/docs/overview/

third shows a bar chart of the emissions per region and cloud service provider, and the fourth shows the carbon intensity in each data center that is used on a world map. Unfortunately, since this tool relies on the billing statements of the providers in order to estimate the carbon footprint, it is not possible to use it for this project.

Similar to the dashboards from Amazon, Google, and Microsoft, these tools were designed according to specific requirements which differ from the ones for this project. This makes them unsuitable for the problem at hand.

2.3 Other Approaches Estimating Environmental Impact

At the same time, more theoretical approaches on estimating the environmental impact of cloud computing have been proposed in the literature. These approaches are often not implemented in any existing tools, but they explain which elements should be measured and how calculations should be performed. These can be categorized based on whether they focus on calculating energy consumption or carbon emissions.

2.3.1 Based on Energy Consumption

Early work on the energy consumption of CPUs was performed by Contreras and Martonosi in 2005 [14], which was realized by monitoring specific CPU events. While this was not specifically in the context of cloud services, it is still a novel approach to mapping CPU usage to power consumption, and a similar approach could be utilized in cloud data centers.

Da Silva et al. [15] showed in 2020 that power consumption is actually not totally dependent on CPU usage, like many other models assume. In fact, I/O operations also play a significant role, as well as whether the cores used are spread across multiple CPU sockets or not. They therefore propose a model which considers CPU usage as well as I/O operations, and also is capable of handling multi-socket machines. However, they used specific coefficients that were obtained by benchmarking their machines. Therefore, to use this model, one would have to perform this same benchmark to obtain the correct coefficients.

Bohra and Chaudhary [18] proposed a method which takes into account CPU, memory, hard disk and network usage, which are combined using weights. However, they noted that the weights would differ per workload, and therefore had to be calculated beforehand using a test workload. Smith et al. [44] later developed an automated method to obtain these weights, called CloudMonitor[10]. Kansal et al. [21] created a similar model, which takes CPU, memory, and disk usage into account, again using platform specific weights which can then further be tailored to each specific workload for better accuracy. They also implemented an automated calculation for these parameters in a project called Joulemeter, which has integration for Microsoft's Hyper-V hypervisor to monitor VM data directly. In addition, Lin et al. [28] created DEM, which is another automated power consumption estimation tool which uses methods based on Kansal et al. and Bohra and Chaudhary [21, 18]. It is therefore quite similar in its purpose to CloudMonitor and Joulemeter. The authors did not provide a direct comparison to either of these two applications, but they did compare it against a standard linear power model, as well as an external power meter. In this comparison they found that their method performs closer to the external power meter, which is used as a ground truth, than the linear power model did.

Jagroep et al. [19] created a single score, called the Resource Utilization Score (RUS), which incorporates CPU, memory, hard disk, network, power consumption, and execution time. This score is calculated by showing the individual components in a radar chart and calculating the area the chart covers.

In 2018, Vasques et al. [51] performed an in-depth review of existing methodologies to measure the power consumption of a multitude of components which are present in a data center,

[10] https://github.com/jws7/CloudMonitor

such as CPU, memory, disks, network interfaces, virtualization frameworks, uninterrupted power supplies (UPS) and cooling. They focused specifically on small to medium sized data centers, and conclude that more research is needed in this area.

Mahadevan et al. [29] proposed a formula to compute network equipment power consumption. This formula is used in multiple cloud simulators, such as CloudNetSim++ and GreenCloud (see Section 2.5). According to the authors' tests, the error margin of this formula was below 2%.

Aslan et al. [2] found that there are many papers which attempt to calculate the power consumption of a (Giga)byte transferred over the network, but the estimated amounts varied wildly. Therefore, they decided to investigate, and determined that there are four general methods through which the energy consumption of network usage can be derived. These methods are modeling, annual electricity consumption, direct measurements, and extrapolation. By combining the results from previous studies, they estimated an energy consumption for network usage of 0.06 Kilowatt-hours per Gigabyte. This value was obtained by gathering as many estimates for network power consumption as possible form the existing literature, then scaling all estimates to a common boundary, and finally taking the year of the estimate into account to create a linear model of network energy consumption over the years, which then allowed them to make their estimate. In their calculation, they included the power consumption from the IP core network and the access networks, the IP core network consisting of Internet Service Provider equipment which forms the regional, national and international networks, and the access networks consisting of devices that connect other devices to the network, such as routers and switches. Here, the routers and switches are only those used by the cloud service provider; the network devices inside the end users of the cloud service's homes or businesses are not included.

Baker et al. [5] found that, while the energy efficiencies and carbon footprints of data centers themselves were being optimized, the connection between the clients and the data center was not. Therefore, the proposed a method whose goal is to find the most energy efficient route, which, as shown in the paper, is not necessarily the shortest route, through a network to any of a service's data centers. In order to achieve this, they, of course, had to quantify the energy consumption of a network request, which is the part that will be the most useful for this project.

Patros et al. [40] discuss several optimizations and improvements, such as power capping, using a more efficient network protocol (QUIC), hotspot and coldspot migration, and more, which can be applied to serverless computing to improve the energy efficiency of this style of cloud computing.

Gill and Buyya [16] have performed a broad overview of the current state of the art of sustainable cloud computing. This work includes a detailed background of sustainable cloud computing, an in-depth taxonomy of cloud computing, a conceptual model for cloud computing that focuses on sustainability, and, most importantly for this project, a number of approaches towards measuring and minimizing energy consumption as well as carbon emissions.

A good amount of these works can be used as a source of inspiration for methodologies to use for the model. However, what needs to be kept in mind is the clarity of the model. In other words, is the formula documented clearly, and whether that data is actually available from BT's data centers. Another requirement to use these methods is that power consumption data is not directly available, otherwise there would not be a need to estimate it based on device usage.

2.3.2 Based on Carbon Emissions

Steenhof et al. [46] developed a method to determine the reduction of the carbon footprint by changing to a different data center. A few years later they published a case study in which they determined the carbon emission reductions from the relocation of the GeoChronous project [47]. To quantify these reductions, they used several formulas to calculate the carbon emissions of multiple parts of a data center. They also collected a large list of carbon sinks, sources, and reservoirs which were relevant for their project, which may help get this project started with determining these as well.

Armstrong and Souraya [1] created a methodology which closely follows the Greenhouse Gas Protocol in Chapter 4 of their book. In this chapter, two methods of obtaining emissions are described: top down and bottom up. Bottom up means that individual emissions are calculated for small parts of the data center, for example individual devices. The emissions can then be summed up to get the total emissions of the data center. Top down means that the total emissions of the data center are calculated first, and then divided over individual machines, VMs, services, and/or users. A more detailed methodology in the top down style is described. The emissions are divided according to the percentage of the data center capacity that is assigned to each machine/VM/service/user. Finally, this chapter includes a very broad set of emissions sources which should be taken into account.

Further research into the total carbon emissions throughout a device's lifecycle has been performed by Belkhir and Elmeligi [7]. However, these are mostly focused on end user devices, such as desktops, laptops, displays, smartphones and tablets. For all of these devices, they list the emissions from production energy, essentially the embedded emissions, and the emissions from the use phase energy.

Additionally, the Gill and Buyya survey already mentioned in the works based on energy consumption actually deals with carbon emissions as well, thus it is briefly mentioned again here.

These methodologies that focus on carbon emissions can be used in the model as high level building blocks. The top down versus bottom up consideration is very important, as it essentially shapes the entire model. The methodologies that focus on Scope 3 are important, since Scope 3 is probably the hardest Scope to measure, but that also makes these methodologies the hardest to use.

2.4 Scheduling Approaches to Optimize Carbon Emissions

Naturally, measuring the carbon footprint is only the first step in data center sustainability. Since cloud services are hosted in data centers, the sustainability of these two is closely related. The next step is taking actions which can reduce this carbon footprint. This section is about works which have proposed workload scheduling methods which aim to reduce the carbon footprint of a data center. These works are relevant to provide a broader background, but also because in order to quantify the effectiveness of these measures, metrics are often used, which may be of use later in this project. This is actually another reason why these works are relevant to this work: part of this work is about measuring the carbon footprint of real life data centers from BT. Furthermore, some of the methods listed here may be interesting to BT in the future, because after calculating the Total Carbon Footprint, the next step is to take action to reduce it.

Wajid et al. [52] developed a scheduling algorithm for cloud workloads which can schedule a certain task, first inside the most carbon optimal data center that is available, next, depending on the strategy, it either tries to fill up the capacity of one physical machine, such that as many machines can be turned off when not in use as possible, or it distributes the tasks as evenly as possible over all available physical machines, to reduce the load on all of them as much as possible. In five of their eight tests, the carbon reductions were over 75%, with the highest reduction being 93.90%, whereas the lowest reduction was 32.25%.

Khosravi et al. [23] proposed a scheduling algorithm which considers three metrics for its decision of where to assign a new incoming task: Power Usage Effectiveness (PUE) [3], Data Center Proportional Power (DCPP), and the carbon footprint rate of the energy sources of each available data center. This scheduler was implemented in CloudSim (see Section 2.5). A few years later, they expanded this project by adding more scheduling strategies, taking into account different combinations of metrics [22]. This document also goes into more detail than the original publication did. Xu and Buyya [53] expanded this project with more scheduling strategies once again in 2020.

Mukherjee et al. [38] also created a scheduling algorithm for cloud data centers, but they used heat generation as their heuristic to determine which physical machine would be best suited for an incoming task, which would directly reduce the cooling energy consumption. They implemented four variations of their heuristic, the best of which (SCINT) achieved a minimum energy saving of 4.7% and a maximum energy saving of 60.5% over a simple first come first served (FCFS) algorithm with first fit (FF). However, the downside of this scheduling algorithm is its runtime: whereas FCFS-FF took anywhere from 3.4ms to 173ms to schedule the tasks, their best algorithm took anywhere from 40 minutes to 147 minutes. This is due to the detailed predictions that are made about how the heat generated by one machine will affect the rest of the machines in the same room. They did note that two of their other variations (FCFS-LRH and EDF-LRH) have runtimes much closer to those of FCFS-FF, while still providing energy consumption improvements.

Chien et al. [13] proposed a concept called ZCCloud, which is a scheduler that will schedule workloads when “stranded power” exists on the grid. This is power that is generated by renewable sources, but cannot be used by anything. Their results showed that even when they used twice the amount of hardware resources, but only intermittently powered on their servers for eight hours per day, more than 30% of the jobs experienced at least similar turnaround time, and large jobs usually experienced better turnaround time. Furthermore, they note that their approach should be able to predict quite easily whether a certain task will benefit from this or not, thus suggesting that a hybrid approach may yield better results and be more feasible.

Gupta et al. [17] explored four existing bin-packing techniques to migrate VMs to different physical machines for minimal energy consumption. They found that the energy utilized best fit decreasing (EUBFD) outperformed all of the other approaches in every single test they performed.

2.5 Cloud Simulators

As mentioned before, for smaller cloud service providers, or on-premises data centers, tools which automatically monitor energy consumption and estimate the carbon footprint are usually not available. In such cases, a simulation of the data center can be created to estimate the carbon footprint. Many simulators exist, so many in fact that there have even been surveys dedicated to comparing them, in 2016 by Suryateja [48] and in 2020 by Bambrik [6]. These surveys both compare a vast set of existing cloud simulators. Below are the simulators which are the biggest in terms of features, scale, and popularity. These works are sorted chronologically, but works that heavily build on or expand upon previous works are kept together with the original work.

GreenCloud by Kliazovich et al. [24] is an early cloud simulator from 2010. It has as its two main goals to simulate packet level traffic between nodes and monitor energy consumption.

Cloudsim by Calheiros et al. [11] is a general purpose cloud simulator from 2011. It allows the user to simulate a cloud service as a network of nodes. In the paper, two short case studies were performed to show potential use cases of this tool: the first case study was an evaluation of a hybrid cloud[11] setup, the second case study was a comparison of an energy efficiency optimizing scheduling algorithm versus one that optimizes for performance. More real life use cases from other groups using CloudSim are also presented in this work of Calheiros et al. In 2019, Zakarya and Gillam [54] created an extension of CloudSim, which added dynamic VM creation during runtime, new scheduling algorithms, extra variables to increase performance simulation accuracy, several technical improvements, and more. Furthermore, they compared the results of CloudSim to real life results, and found that CloudSim had 98.63% accuracy.

[11] A setup where a service is provided through the use of both a private and public cloud.

Núñez et al. created iCanCloud [39] in 2012, whose main features are that it is designed with large simulations in mind, has an easily extensible simulated hypervisor to allow custom scheduling policies to be easily implemented, and it has presets for AWS instance types. In 2013, an extension called E-mc^2 was developed by Castañe et al. [12]. This extension consists of a framework which improves the energy calculation abilities.

CloudSched by Tian et al. [49] is a simulator that was published in 2015, which mainly focuses on comparing different scheduling algorithms, however, it does also have metrics for the energy consumption during the simulation.

CloudNetSim++ is a simulator created by Malik et al. [30] in 2017 that allows support for geographically distributed clouds, different scheduling algorithms, heterogeneous nodes, pricing and SLA conformity monitoring, and, most importantly for this thesis, detailed energy consumption.

The most recent simulator, being from 2021, is OpenDC 2.0 by Mastenbroek et al. [32]. OpenDC 2.0 has as its two main features simulation support for serverless architectures and support for Tensorflow machine learning workloads. This simulator also has metrics for energy consumption.

Cloud simulators are interesting because they often have some way to estimate the power consumption related to a simulated workload. The methodologies that they use can be used in this work, as long as the simulation goes into enough depth to actually simulate different usage statistics of different parts of the IT hardware, which most simulators unfortunately do not. Furthermore, a large amount of these simulators are built on methodologies already listed in previous sections.

Chapter 3

Case Study

This project will be conducted as an exploratory case study, as it matches with BT's underlying goal of presenting the results of this project to their customers and use their feedback to review its usefulness. This case study is designed according to the guidelines by Runeson and Höst [41].

In this chapter, we will first go over the case description in Section 3.1. After that, the problems that BT faces, as well as the problems that will occur when undertaking this project will be detailed in Section 3.2. The method that is used to approach this case is described in Section 3.3. The requirements for this project will be listed in Section 3.4. Next the data collection procedures will be described in Section 3.5, followed by a description of the analysis procedures in Section 3.6. Finally, the chapter will be concluded by going over the validity procedures in Section 3.7.

3.1 Case Description

As mentioned in Section 1.3, customers of BT's SaaS product CCC are starting to become interested in the carbon footprint associated with using BT's services. To help its customers gain insights in this, BT's goal is to provide them with a monthly report which provides a detailed but concise overview of the customer's energy consumption and carbon footprint. Additionally, due to the importance of this report, it requires a sound methodology, backed up by academic literature.

The subjects of this case study are a subset of BT's customers who have bought the CCC product, in other words, tenants of the data centers. Each customer in turn has a set of agents actually using the CCC product's services, which could be seen as another layer of multitenancy, but this is not explored much in this project beyond also measuring the average carbon footprint per agent. The goal is to verify with them that the generated report is sufficient in satisfying the requirements (see Section 3.4), as well as get their general opinion on the project.

3.2 Problem Definition

To BT's customers, and therefore to BT, it is important that the carbon footprint calculations are, of course, as accurate as possible. However, as is apparent in the literature, see Chapter 2, defining the complete scope of the carbon footprint is often a matter of opinion. While it is usually unanimously agreed that certain carbon emission sources definitely should be included, most notably the server machines themselves, but also cooling and network equipment, for other elements, such as the power consumption of the network external to the data center, or the carbon emissions from producing data center hardware and buildings, the situation is not as clear.

This actually highlights one of the biggest difficulties in measuring the Total Carbon Footprint: responsibility. It is clear that carbon has been emitted during such processes; who is responsible for it though? Is it the manufacturer or construction company, since they built the product or the building? Is it the data center operator, since they ordered the server hardware and the building to be built? Or is the customers who use the data center's hardware, since they created the demand for these products and data centers to exist? This is an unanswered question and will be one of the focus points of this project.

Another difficulty of this project stems from the way that BT's services are hosted, and the way that the data centers that it uses are organized. BT hosts its services partially on big Infrastructure as a Service (IaaS) providers, for example Amazon Web Services (AWS), and partially on off premises, private cloud data centers; these data center buildings and general infrastructure are not directly owned by BT, but the server hardware is owned by BT. In this work, only the private cloud data centers will be considered; the IaaS providers are out of scope.

Because of this organization, there are two (apparent) levels of multitenancy: BT is one of multiple tenants in each data center it uses, and BT's customers are individual tenants in "BT's part" of the data center. To keep the scope of this project manageable and the goals realistic, it was decided that this project will only focus on their private data centers, as it will be easier to obtain low level data. While plenty of research has been performed on the carbon footprint of data centers, as has been detailed in Chapter 2, these works usually focus on the carbon footprint of an entire data center. However, the multitenancy aspect has not been researched nearly as in-depth. Thus, this will also need to be researched for this project.

The final difficulty is the distinction between net and gross emissions. Since the data centers that BT uses already use a certain amount of green energy, and BT purchases Renewable Energy Certificates, the amount of physical, gross emissions that have been produced during the provision of the service is different from the net difference in the amount of carbon in the atmosphere due to the service being provided.

3.3 Method

In order to approach this case, we will first develop a theoretical model using the literature, as well as the data available from the BT data centers, and the requirements from BT to guide our decisions. This means the model has to be developed in close cooperation with BT to verify the feasibility of each part. Once this is completed, the model can be implemented in a larger program which will take the data that is needed as input for the model, pass it through the model to obtain metrics such as the Total Carbon Footprint, which can then be shown in a report. This report will have to be designed to be easily digestible, since it will be shared with BT's customers. Once all of the development is done, the report will be shown to BT's customers to get their feedback, which can be used to verify the satisfaction of each of the requirements that BT had.

3.4 Requirements

As mentioned in the previous section, one of the final products of this project, aside from this manuscript, and a software deliverable, which will be discussed in-depth in Chapter 5, should be a model which can be used to compute the Total Carbon Footprint of one customer of BT's services. The following requirements were elicited initially from meetings with BT to discuss the general goal of the project, and some requirements were added later after sharing updates on the progress of the project with BT.

3.4.1 Functional Requirements

1. The model must compute the Total Carbon Footprint per tenant.
2. The model must include BT's scope 1, 2, and 3 emissions.

3. The model must compute a detailed breakdown of emissions by source.
4. The reports must be automatically generated on a monthly basis.

3.4.2 Non-Functional Requirements

1. The model must distribute the Total Carbon Footprint to each tenant in a fair manner.
2. The model calculations must be auditable.
3. The model must be as thorough as possible, but at the same time realistic to implement.
4. The report must fit on one page, to force the contents to be concise and the information easy to consume at a glance.

3.5 Data Collection Procedures

The data collection for the requirements of this project was done through regularly scheduled meetings with BT, in person and online, as well as email exchanges to clarify small issues that arose while working out the implementation.

3.6 Analysis Procedures

As briefly mentioned in Section 1.4, the results of the project were presented to three of BT's customers, who were then interviewed to verify the satisfaction of the requirements from Section 3.4. To perform this verification with the selected subjects, video calls were set up in which the entire project was presented and the produced report was shown to the subject, who is acting as the tenant in the data center, and finally a set of questions was asked to determine the satisfaction level with relation to the requirements. The presentations and questions and answers were recorded to allow the attendees of the presentation to focus solely on the ongoing discussion during the meeting. The answers could then be watched back to allow for direct quotes from the discussion to be used. The subjects are kept anonymous for privacy and business related reasons. Since the survey consists of open questions, the analysis has to be done qualitatively. This will be done in Chapter 6, by using arguments made by the participants to argue whether the requirements (see Section 3.4) are satisfied. This is possible due to the recordings of the calls.

3.7 Validity Procedures

As mentioned, to check the external construct validity of this work, interviews will be conducted with three of BT's customers. The questions asked in the interview part of the meetings are the following:

Survey question 1 *How clear is the report and methodology?*

Survey question 2 *How useful is the information from the PDF report and the JSON file?*

Survey question 3 *How appropriate do you consider the method for calculating shared emissions?*

The answers of these questions will be used to determine if the requirements of this project have been met.

Chapter 4

Model Design

In order to obtain the Total Carbon Footprint, a model is needed to calculate it. To create a model, methods from existing works will be used (see Section 2.3). In this chapter, we will describe the model design that was used for this project, and which methodologies were used as the building blocks. Section 4.1 first goes into how responsibility will be calculated. Section 4.2 then breaks down the Total Carbon Footprint into the three scopes of emissions defined in the Greenhouse Gas Protocol. Section 4.3 breaks down Scope 2 even further into the individual device categories and details the methodologies used to calculate the power consumption and carbon emissions for each of them. Finally, Section 4.4 details how certain actions offset carbon emissions, and how a net Total Carbon Footprint, which takes these offsets into account, can be obtained.

The model is designed as follows: we use the three Scopes defined by the GHG Protocol [4] to categorize the different carbon emission sources. Scope 4 by Matthews et al [33] is not used in this model, since the fine grained information of the Scope 3 emissions needed to split it into a separate Scope 4 was not available. Furthermore, the separation will not affect the Total Carbon Footprint. Thus, the Total Carbon Footprint is built up as in Equation 4.1. Since we are interested in the carbon footprint for each tenant in the data center, all elements are already tenant specific unless stated otherwise. This means that the parameters used in each formula should already be tenant specific, or should be made tenant specific. In case of the latter, Equation 4.2 can be used. For all units, their base units are used. This means all energy measurements are done in Watt-hours, and emissions are measured in grams.

$$TCFP = Scope1 + Scope2 + Scope3 \tag{4.1}$$

4.1 Responsibility

When dealing with carbon footprints, it is important to take responsibility into account. Of course, the carbon footprint for any entity should only include the emissions they are actually responsible for, or an appropriate share of the emissions in case the emissions can not easily be attributed to a single entity. This leads to two separate areas in which responsibility can be shared in the scope of this work: between tenants, due to the multitenancy aspect of cloud services, and between the data center and the tenants, as discussed by Lenzen and Murray [26]. In their work, they say that the main reason for sharing responsibility between the data center and the tenants is to prevent double counting of emissions. This becomes important when assessing global carbon emissions, since then the emissions of the cloud service provider and the users would be added up, leading to said double counting. The way this is prevented is through a percentage share, where a certain percentage x is assigned to the cloud service provider, and the remaining percentage $1 - x$ is assigned to the user(s). In case of multiple users, the $1 - x$ percentage can be split just like any other emission that is shared between the users of the cloud service, such as, for example, cooling energy consumption. The equation to calculate the complete share for a tenant in a data center is given in Equation 4.2. This equation includes both areas of responsibility sharing discussed in this paragraph.

$$\begin{aligned} r &= multitenancy\ share * L_{share} \\ multitenancy\ share &= \frac{Tenant_{Scope2}}{DC_{Scope2}} \end{aligned} \tag{4.2}$$

Where L_{share} is the share that is attributed to the tenant according to the Lenzen and Murray responsibility sharing method. In the implementation of the model used in this project, a value of 100% was used for L_{share}.

4.2 Considering the Different Scopes

We can break down each scope into its individual emission sources, as done in Equations 4.3 to 4.5.

$$Scope1 = \sum_{device \in devices} (F_{device} * c_{device} * r) \tag{4.3}$$

Here Scope 1 is a sum of all Scope 1 devices, where the carbon emissions of each device are calculated by multiplying the fuel consumed by the device, F_{device}, by the carbon intensity of the device, c_{device}, which is the amount grams of CO_2 emitted per gram of fuel consumed by the emission source, and r is the tenant's responsibility share.

In BT's case, they do not use any direct emission creating devices (such as generators), and therefore, their Scope 1 emissions are 0. For Scopes 2 and 3, we have:

$$Scope2 = \sum_{DC \in DCs} ((E_{DC_{server}} + E_{DC_{network}} + E_{DC_{cooling}} + E_{DC_{misc}}) * c_{DC} * L_{share}) \tag{4.4}$$

Where $E_{DC_{server}}$ is the energy consumed by the tenant's usage of the servers, $E_{DC_{network}}$ is the energy consumed by the tenant's usage of the network equipment, $E_{DC_{cooling}}$ is the energy consumed by the cooling system as a consequence of the tenant's heat production, $E_{DC_{misc}}$ is the miscellaneous energy consumption for which the tenant is responsible, c_{DC} is the amount of CO_2 emitted per Kilowatt-hour, known as the carbon intensity, and L_{share} is the share that is attributed to the tenant according to the Lenzen and Murray responsibility sharing method.

$$Scope3 = Scope3_{DC} * r \tag{4.5}$$

Where $Scope3_{DC}$ is the total Scope 3 emissions for the entire data center and r is the tenant's responsibility share.

4.3 Breaking Down Scope 2

Scope 2 consists of many different devices. Unfortunately, in BT's data centers, it is not possible to get direct power measurements for each device, as it is not available through software, and therefore would require installing a power meter between each device and the wall socket, which would be costly and time consuming to install. Therefore, models to estimate the power consumption for each main device type should be used.

The server energy component E_{server} in Equation 4.4 can be calculated using multiple different algorithms, namely Algorithm 1 by Da Silva et al., Algorithm 2 by Bohra and Chaudhary., Algorithm 3 by Kansal et al. In the implementation of the model for this project, Algorithm 2 was used, because it was the most feasible in the case of BT's data centers; however, this can be swapped for any other algorithm quite easily.

To calculate the four parameters for the CPU usage, cache data, DRAM data, and disk data, the R script from Listing 4.1 can be used. The output shows a summary of the linear model constructed from the input data, which is constructed through a linear regression using the measured power consumption as the dependent variable, and the CPU utilization, cache data moved, DRAM accessed, and disk data moved as the independent variables. The coefficients are listed in Table 4.1. The idle power is represented by the intercept of the model, since that is a constant that is always added. The value for `cache_data_moved` is N/A because this input data was not available in this project. To get an idea of the accuracy of these model parameters, we can look at the R-squared value. Seeing as the value is rather low, we can conclude that more data would be needed to get a more accurate model for the server energy consumption.

Coefficient	**Value**
`idle_power` (Intercept)	1.650e5
`cpu_util`	1.187e4
`cache_data_moved`	*N/A*
`dram_accessed`	1.828e-4
`disk_data_moved`	1.365e-6

Table 4.1: The coefficients for the Bohra server model obtained using the R script from Listing 4.1.

Listing 4.1: A short R script to obtain the server model parameters for the Bohra model.

```
# read the data from a csv file provided by BT, it contains the power_consumption of
    ↪ each device, the cpu_util, cache_data_moved, dram_accessed, and disk_data_
    ↪ moved
server_data <- read.csv("./data/server_data.csv")
# perform linear regression to get coefficients
server_model <- lm(formula = power_consumption ~ cpu_util + cache_data_moved + dram_
    ↪ accessed + disk_data_moved, data = server_data)
# show summary of model, including coefficients and intercept
summary(server_model)
# Residuals:
#    Min      1Q  Median      3Q     Max
# -75013  -33947  -16523   23843  128785

# Coefficients: (1 not defined because of singularities)
#                   Estimate Std. Error t value Pr(>|t|)
# (Intercept)      1.650e+05  2.563e+04   6.435 2.23e-06 ***
# cpu_util         1.187e+04  4.080e+03   2.910  0.00837 **
# cache_data_moved        NA         NA      NA       NA
# dram_accessed    1.828e-04  6.984e-05   2.617  0.01612 *
# disk_data_moved  1.365e-06  6.892e-07   1.980  0.06094 .
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Residual standard error: 57820 on 21 degrees of freedom
# Multiple R-squared:  0.4423,   Adjusted R-squared:  0.3626
# F-statistic:  5.55 on 3 and 21 DF,  p-value: 0.005764
```

Algorithm 1: Server energy consumption by Da Silva et al. [15].

```
input  : Individual CPU core and IO usages of a set of tasks
output: The energy consumption of a (part of a) server

function cpu_power(task, core_number)
    /* if the task is running on the first core on the socket                */
    if core_number = 0 then
        (maximum_cpu_power - minimum_cpu_power) * $\frac{u}{s \cdot n}$
    else if core configuration is pairwise then
        /* running a task on the core_number$^{th}$ core consumes slightly less
            power than running it on the core_number-1$^{th}$ core           */
        pairwise_cpu_power_scale * cpu_power(task, core_number - 1)
    else if core configuration is unpaired then
        unpaired_cpu_power_scale * cpu_power(task, core_number - 1)
    end
end

function io_power(task, core_number)
    if io is saturated then
        io_staturated_mul = 1
    else
        io_staturated_mul = 0
    end
    if core configuration is pairwise then
        /* running a task on the core_number$^{th}$ core consumes slightly less
            power than running it on the core_number-1$^{th}$ core           */
        pairwise_cpu_power_scale * (1 + io_saturated_scale * io_saturated_mul) *
          cpu_power(task, core_number - 1)
    else if core configuration is unpaired then
        unpaired_cpu_power_scale * (1 + io_saturated_scale * io_saturated_mul) *
          cpu_power(task, core_number - 1)
    end
end

/* server energy                                                             */
$E_{\text{server}} \leftarrow 0$
for task in tasks do
    $E_{\text{server}} \leftarrow$ server_idle_power + cpu_power(task, core_number_of_task) +
      io_power(task, core_number_of_task)
end

return $E_{server}$
```

Algorithm 2: Server energy consumption by Bohra and Chaudhary [18].

```
input  : The amount of certain events that have occurred
output: The energy consumption of a (part of a) server

/* The following parameters are calculated from a benchmark          */
$c_1$, $c_2$, $c_3$, $c_4$ = benchmark()

/* server energy                                                     */
$E_{server}$ ← idle_power + $c_1$ * amount_CPU_CLK_UNHALTED + $c_2$ *
  (amount_INSTRUCTION_CACHE_FETCHES +
  amount_DATA_CACHE_FETCHES) + $c_3$ * amount_DRAM_ACCESSES + $c_4$ *
  (disk_bytes_read + disk_bytes_written)

return $E_{server}$
```

Algorithm 3: Server energy consumption by Kansal et al. [21]

```
input  : Usage of certain server components
output: The energy consumption of a (part of a) server

$\alpha_{cpu}, \gamma_{cpu}, \alpha_{mem}, \gamma_{mem}, \alpha_{br}, \alpha_{bw}, \gamma_{disk}$ ← benchmark()
/* server energy                                                     */
$E_{server}$ ← idle_power + $\alpha_{cpu}$ * $usage_{cpu}$ + $\gamma_{cpu}$ + $\alpha_{mem}$ * $num_{LLCM}$ + $\gamma_{mem}$ + $\alpha_{br}$ *
  $bytes_{read}$ + $\alpha_{bw}$ * $bytes_{written}$ + $\gamma_{disk}$

return $E_{server}$
```

Algorithm 4: Network energy consumption by Mahadevan et al. [29].

```
input  : The amount of ports configured at certain bandwidths
output: The energy consumption of a (part of a) network device.

/* network energy                                                    */
$E_{network}$ ← 0
for net_device in network do
    port_power_sum ← 0
    for port_bandwidth in port_bandwidths do
        port_power_sum ← port_power_sum + num_ports at port_bandwidth *
          port_power at port_bandwidth * port_utilization
    end

    $E_{network}$ ← $E_{network}$ + net_device.idle + net_device.num_linecards * linecard_idle
      + port_power_sum
end

return $E_{network}$
```

For the network energy consumption $E_{network}$, two methods can be used. Either an estimate of $6 * 10^{-8}$ Watt-hours per byte sent through the network device can be used, as proposed by Aslan et al. [2], or to be calculated by Algorithm 4. In this implementation, the estimate by Aslan et al. is used, see Equation 4.6. This is because the detailed information needed for the method from Mahadevan et al. was not available from the data centers used in this project.

$$E_{network} = 6 * 10^{-8} * (bytes\ sent + bytes\ received) \tag{4.6}$$

Cooling energy $E_{cooling}$ is a value that cannot be attributed to each specific tenant directly. Therefore, it has to be split according to the tenant's share of the power consumption that can be directly attributed to each tenant. The implementation of this is in Algorithm 5.

Algorithm 5: Cooling energy division.

```
input  : The power consumption of the cooling devices
output : The energy consumption of the cooling devices a given tenant is responsible for

$E_{cooling} \leftarrow 0$
multitenancy_share ← tenant_attributable_emissions / total_attributable_emissions
for cooling_device in cooling_devices do
    $E_{cooling} \leftarrow E_{cooling}$ + cooling_device.power * multitenancy_share
end

return $E_{cooling}$
```

For the miscellaneous energy consumption E_{misc}, representing energy consumers such as lighting, security systems, et cetera, we use Algorithm 6. The algorithm is essentially a generalized version of Algorithm 5. In this algorithm, a distinction is made between directly attributable emissions and non-directly attributable emissions, as seen by the conditional inside the loop. For each device, the power consumption is multiplied by the data center's carbon intensity.

Algorithm 6: Miscellaneous energy consumption.

```
input  : A list of miscellaneous devices and their power consumption
output : The emissions of all miscellaneous devices in a data center
$E_{misc} \leftarrow 0$
for misc_device in devices do
    if misc_device is a directly attributable emission source then
        $E_{misc} \leftarrow E_{misc}$ + misc_device.power * misc_device.DC.carbon_intensity
    else
        multitenancy_share ← tenant_attributable_emissions / total_attributable_emissions
        $E_{misc} \leftarrow E_{misc}$ + misc_device.power * misc_device.DC.carbon_intensity * multitenancy_share
    end
end
return $E_{misc}$
```

4.4 Calculating Net Emissions

Until now, the calculations have all returned the total amount of carbon emitted. However, certain ways to offset the carbon footprint may be used, which cancel out a part of the emissions. Therefore, the aforementioned Total Carbon Footprint is the gross carbon footprint, and the net Total Carbon Footprint can be calculated by subtracting these offset methods, as shown in Equation 4.7.

$$TCFP_{net} = \sum_{DC \in DCs} (TCFP_{DC} - E_{DC_{green}} * c_{DC} - REC_{DC} * r) \tag{4.7}$$

Where E_{green} is the percentage of energy used that is green, REC_{DC} is the amount of carbon offset for the specified data center by Renewable Energy Certificates (REC), and $TCFP_{DC}$ is the total carbon footprint of the specified data center.

Chapter 5

Program Design and Model Implementation

To automate the creation of the Total Carbon Footprint reports for each of BT's customers, a report generator was developed which takes tenant usage statistics as input and produces a sustainability report in PDF format, as well as a more detailed JSON file. In the following sections, the program architecture and design will be explained. Section 5.1 gives a general overview of the generator and its architecture. Section 5.2 explains which technologies are used and why they were chosen. Section 5.3 provides a detailed look at each of the components in the program architecture and explains their use. Finally, Section 5.4 goes into detail about the two generated reports from the report generator.

5.1 Program Overview

A general overview is shown in Figure 5.1. The application is provided as a bundle with a portable installation of Tex Live. This decision was made to make the application as self-contained as possible, however, since a full TeX Live installation is relatively large (a few Gigabytes compared to the few Megabytes needed for the rest of the application), a bundle without Tex Live is also provided. The bundle also includes the TeX template, as well as a Readme file for instructions. The choice to provide complete bundles like this was made to allow the users of this program to run the program without manually having to install any dependencies, thus providing the smoothest user experience.

5.2 Technologies

The application is written in Kotlin, targeting the JVM to make the compiled application, in this case a JAR file, executable on any machine with a JVM installed, and to allow for the use of the vast amount of packages that are available in repositories such as Maven Central[1]. Kotlin is a modern language with many high level constructs, which makes it more concise and elegant to write. Furthermore, JAR files are highly user friendly, since they only require a JVM to be executed (which most systems will already have installed); all code dependencies can be included in the JAR itself. Another benefit is that JAR files are operating system agnostic[2].

The report template is written in LaTeX, as it is a very powerful tool to create documents, often compiled to PDF. The fact that its source is stored as plain text rather than a proprietary format makes is easy to programatically inject text into it, and compiling it into a PDF file is

[1] https://repo1.maven.org/maven2/

[2] Provided the code itself is also written in an operating system agnostic manner by using abstractions to deal with operating system dependent tasks, for example by using `java.nio.file.Path` to handle file system paths.

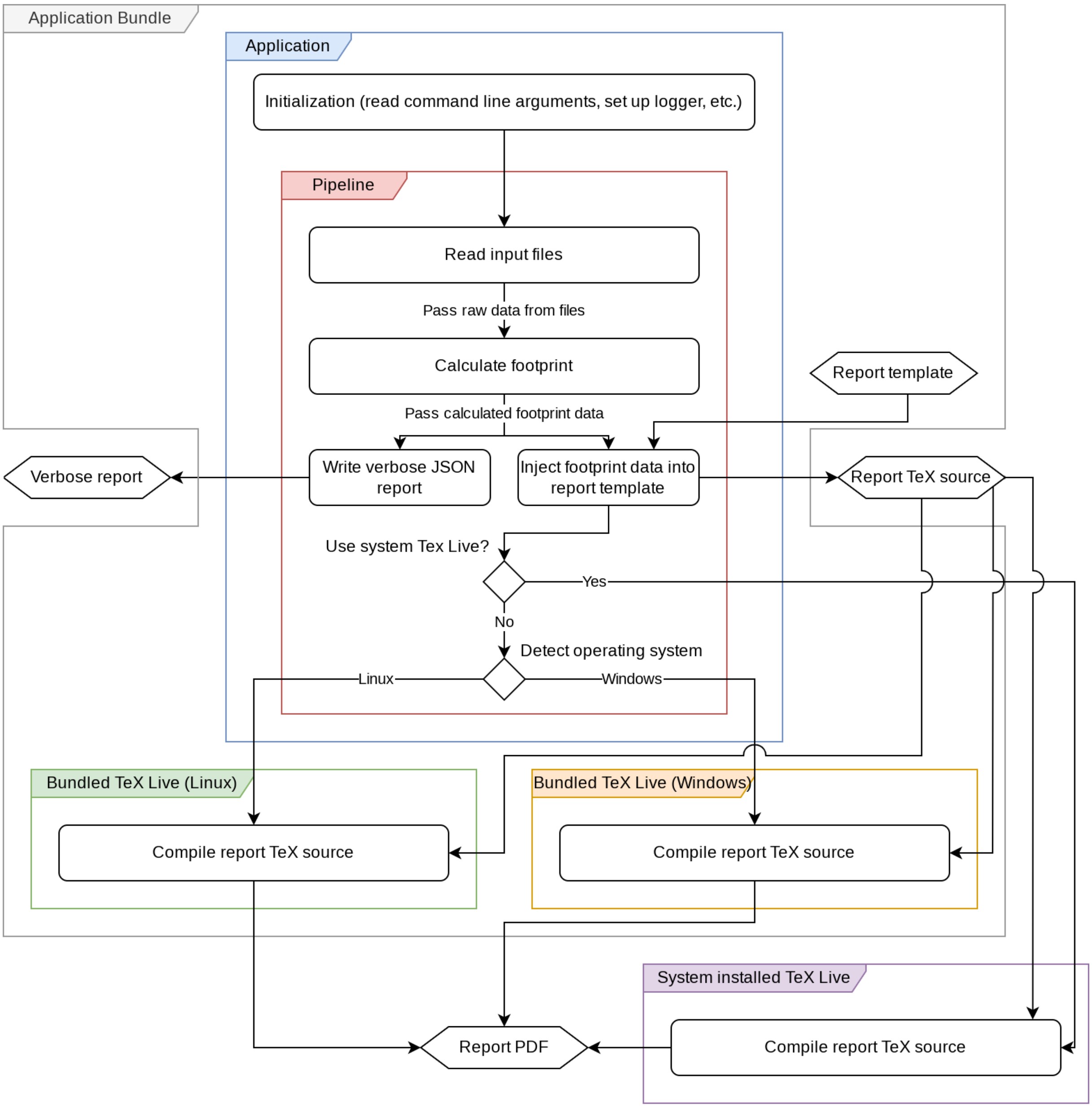

Application Bundle
Application
Initialization (read command line arguments, set up logger, etc.)
Pipeline
Read input files
Pass raw data from files
Calculate footprint
Report template
Pass calculated footprint data
Verbose report
Write verbose JSON report
Inject footprint data into report template
Report TeX source
Use system Tex Live?
Yes
No
Detect operating system
Linux
Windows
Bundled TeX Live (Linux)
Compile report TeX source
Bundled TeX Live (Windows)
Compile report TeX source
System installed TeX Live
Report PDF
Compile report TeX source

Figure 5.1: An overview of the program design. The colored frames show the scope in which actions are happening. The rounded rectangles are actions that happen. The rectangles with tapered ends are files which are either used or created.

as easy as one system call. Thus, the entire of process of generating a report was easily done this way.

5.3 Detailed Look at the Components

In this section, a deeper look into the functionality and the purpose of each part of the application is provided.

5.3.1 Main Application

The main class in the application has a simple purpose: to intialize the program and to catch any uncaught exceptions and send them to the logger. Initialization consists of reading and parsing the command line arguments, initializing the logger, checking the JVM version to ensure compatibility, and starting the pipeline.

5.3.2 Pipeline

The pipeline is the part of the application that deals with the steps related to generating the final PDF report. It consists of three main steps: reading input, calculating the carbon footprint, and creating the report itself. Using a pipeline architecture decouples separate tasks from each other. Each step only needs to ensure that it handles the input in the expected format and outputs a result in the expected format; the actual implementation can easily be adjusted or replaced without modifying the other components in the pipeline.

Reading Input

The input of the program are four CSV files, one which contains the usage information about the network equipment, one which contains usage information about the server equipment, one which contains general information about the data center(s), and one which contains general information about every tenant. The CVS files are exported from the software BT uses to manage their servers and network devices. This step does only the minimal amount of processing on the CSV files, essentially creating a JVM object which represents the data, called a `RawData` object. This object is then passed on to the next step.

Calculating the Carbon Footprint

The next step of the pipeline is to take the `RawData` object and use it to calculate the carbon footprint of each tenant, among other statistics. Indeed, despite the name, the carbon footprint is not the only thing calculated by this component, but also metrics such as total energy consumption, energy consumption per device type, and gross and net emissions, which are the result of BT already making several efforts to offset some of their data centers' emissions. All of these calculated results are composed into one object per tenant, called a `Footprint` object. Thus, the result of this step is a list of `Footprint` objects, one for each tenant. In the current implementation, the reports are generated on a monthly basis, and therefore show the monthly carbon footprint. However, the model itself is timescale-agnostic; by its design the resulting carbon footprint is presented in the same timescale as the input data.

Generating the Report

The final step in the pipeline is to generate a report for each tenant. This is done by using a template, in which the data of each tenant is injected during this stage. Since it is a template, it contains placeholders, which are marked by `@placeholder:"PLACEHOLDER_NAME"`. These placeholders are replaced by their actual values during this stage. Note that this means the template does not compile "as-is". The full source code of the template is listed in Appendix B.

While this stage technically consists of three smaller steps in the current implementation of the program, namely injecting the footprint data into the template, compiling the template to a PDF, and generating a verbose report, this is not relevant to the rest of the pipeline. In other words, a different implementation might consist of a different amount of steps internally, but from the perspective of the pipeline, it is just one step: generating a report.

In this implementation, this is done by taking the template and injecting the footprint data in the places marked for it by placeholders. Then, a LaTeX compiler[3] is invoked to compile the filled in template TeX source to a PDF.

On top of the PDF report, which is designed to be pleasant to look at and easy to understand, there is also a JSON file that gets generated. While this is less easy to understand for most people, it is of course much easier to programmatically deal with, should that be desired. It also contains some more information that is not present in the PDF, to better satisfy the auditability requirement (see Non-functional requirement 2).

Just like the other stages of the pipeline, the implementation of this stage is self contained; it can easily be modified or completely swapped out for a different implementation as long as the required input stays the same.

5.4 Program Output

As described in Figure 5.1, there are two main outputs of the program. The first output is a PDF report, referred to as the Sustainability Report, which shows the most important information in a way that is easy to understand. The second output is a JSON file, which shows much more information. These outputs will be discussed further in the following.

5.4.1 Sustainability Report

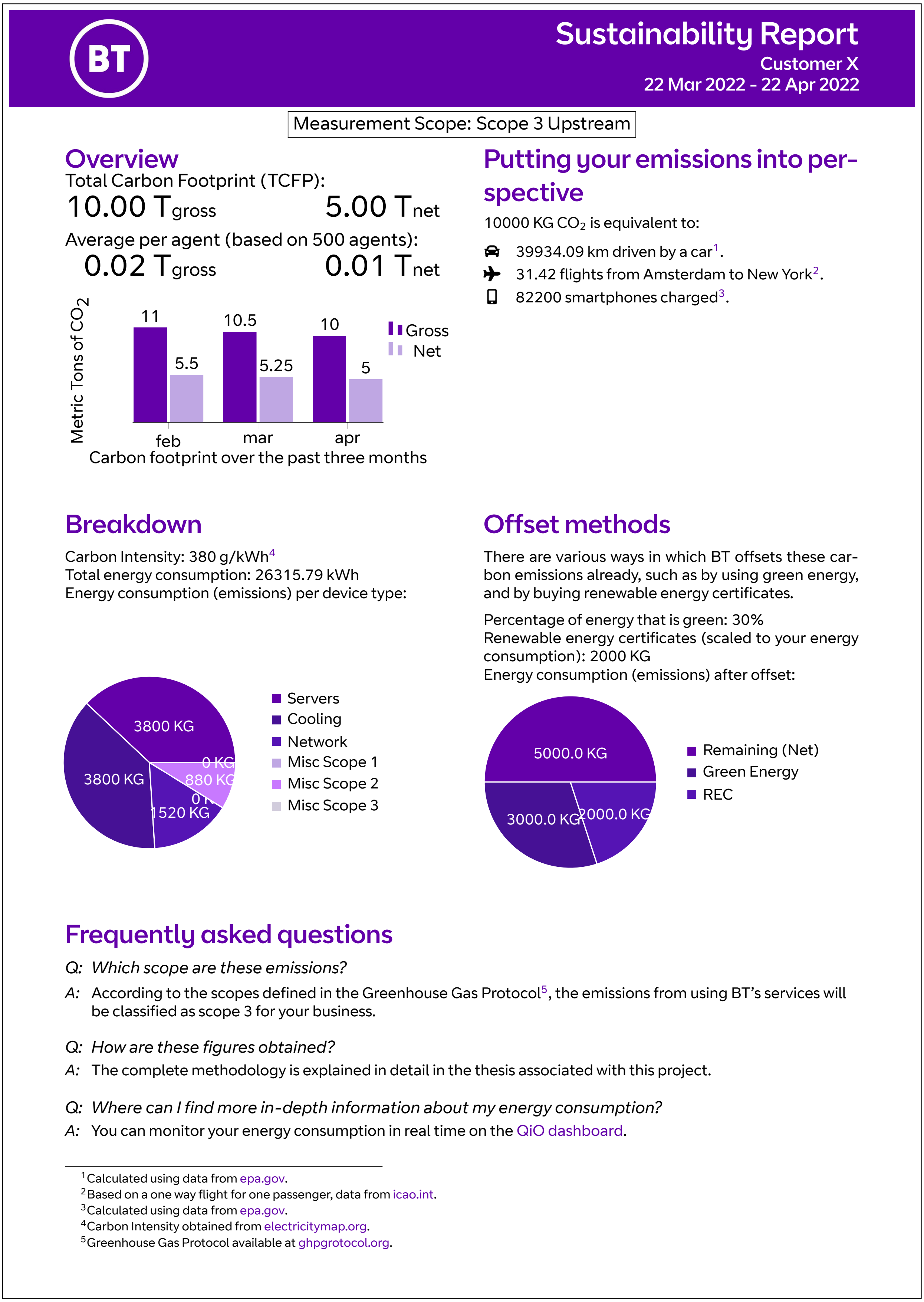
The sustainability report consists of five sections. The first section is a look at the gross and net carbon emissions from the current month, as well as the carbon emissions per agent, and a comparison to the previous two months shown in a bar graph. In the second section, the emissions are put into perspective by comparing them to the number of flights it takes to emit an equal amount of carbon, as well as kilometers driven by car, and the amount of smartphones charged. This is based on the gross emissions. The third section is a breakdown of which sources the emissions consist of, visualized by a pie graph. This pie graph contains the Scope 1, 2, and 3 emissions, but the Scope 2 emissions are broken down into server emissions, cooling emissions, network emissions, and miscellaneous Scope 2 emissions. The fourth section is another pie graph, visualizing the share of emissions offset by each offset method, as well as the net emissions, which is the remaining part of the emissions that have not been offset. Finally, at the end of the page, there is a list of frequently asked questions which the reader can refer to for more information. These detail the Scope of the emissions for the tenant, a reference to this thesis for the full methodology, and a link to the dashboard provided by BT to track energy consumption and carbon emissions in real-time, which is described in Section 1.3. An example report using fictitious data is shown in Figure 5.2. As shown in the figure, the fictitious tenant Customer X is responsbile for a gross carbon footprint of 10 metric tons of $CO_2$for the period of the 22nd of March until the 22nd of April. Of this gross footprint, 3 metric tons have been offset by using green energy, and 2 metric tons have been offset by Renewable Energy Certificates. Furthermore, as shown in the historic graph in the top left, during the month of February this tenant's gross emissions were 11 metric tons, and in the month of March this tenant's gross emissions were 10.5 tons.

5.4.2 JSON Report

The JSON report consists of the data that is used on the sustainability report, and additionally a more detailed breakdown of the emissions. Specifically, the estimated power consumption and emissions of (the tenant's share of) each individual device are listed, grouped by device type, BT

[3] Specifically LuaLaTeX, to make use of the unique features it has, such as using a custom font directly from provided TTF files.

scope, and data center. This improves the auditability of the report, as it allows calculations to be verified and re-done. Because JSON is a widely supported format, with libraries providing functionality for handling it available for essentially every programming language, it is easy to perform new calculations on the data as well, for example using a new model to estimate power consumption. A full, fictitious JSON report is shown in Appendix A. Note that the figures in this report are unrelated to the figures in Figure 5.2, they are merely there to act as placeholders that show the format of the data.



BT

Sustainability Report

Customer X
22 Mar 2022 - 22 Apr 2022

Measurement Scope: Scope 3 Upstream

Overview

Total Carbon Footprint (TCFP):

10.00 Tgross 5.00 Tnet

Average per agent (based on 500 agents):

0.02 Tgross 0.01 Tnet

Metric Tons of CO_2
11
5.5
10.5
5.25
10
5
Gross
Net
feb
mar
apr

Carbon footprint over the past three months

Putting your emissions into perspective

10000 KG CO_2 is equivalent to:

- 39934.09 km driven by a car[1].
- 31.42 flights from Amsterdam to New York[2].
- 82200 smartphones charged[3].

Breakdown

Carbon Intensity: 380 g/kWh[4]
Total energy consumption: 26315.79 kWh
Energy consumption (emissions) per device type:

Offset methods

There are various ways in which BT offsets these carbon emissions already, such as by using green energy, and by buying renewable energy certificates.

Percentage of energy that is green: 30%
Renewable energy certificates (scaled to your energy consumption): 2000 KG
Energy consumption (emissions) after offset:

Frequently asked questions

Q: *Which scope are these emissions?*

A: According to the scopes defined in the Greenhouse Gas Protocol[5], the emissions from using BT's services will be classified as scope 3 for your business.

Q: *How are these figures obtained?*

A: The complete methodology is explained in detail in the thesis associated with this project.

Q: *Where can I find more in-depth information about my energy consumption?*

A: You can monitor your energy consumption in real time on the QiO dashboard.

[1] Calculated using data from epa.gov.
[2] Based on a one way flight for one passenger, data from icao.int.
[3] Calculated using data from epa.gov.
[4] Carbon Intensity obtained from electricitymap.org.
[5] Greenhouse Gas Protocol available at ghpgrotocol.org.

Figure 5.2: A fictitious example of a report produced by the sustainability report generator.

Chapter 6

Evaluation

In this chapter, we will discuss the results of the case study part of this project and validate the satisfaction of the requirements in Section 6.1, by presenting the results of the questions asked to three of the tenants in BT's data centers (as listed in Section 3.7), comparing the tool created in this work against some existing tools in Section 6.2, and finally discussing the limitations of this project in Section 6.3.

6.1 Survey Responses

As mentioned in Section 3.6, part of the evaluation of this project is done through a survey held after a presentation of the project and the generated PDF report and JSON file. In the following sections, the responses of the participants to the survey questions are discussed. As mentioned in Section 3.6 as well, the names of the tenants will be kept anonymous. They have been given numbers 1 through 3, so it is still clear which comments came from which respondent.

6.1.1 Survey Question 1: How clear is the report and methodology?

Overall the respondents thought that the report and methodology were quite clear, however, there were some parts which were a bit unclear. Respondent 1 said: "What I don't quite understand is the renewable bits, is that assuming ... that BT is using some type of solar or wind to generate power or is that just part of how the grid works." Respondent 2 said: "I don't know what's good and what's bad, ... it needs grounding," and as a solution to this: "[For example] a R[ed]A[mber]G[reen] status of what we're seeing, some form of tie-it-together so I can judge it." Respondent 3 shared this view as a general comment before the survey part of the call: "It needs to be quick and easy to say where we're trending." This indicates that some extra clarification on the report might be needed.

6.1.2 Survey Question 2: How useful is the information form the PDF report and the JSON file?

The general consensus among the participants regarding this topic seemed to be that the level of detail is great to have, but the JSON format might be intimidating for people who have not worked with it before. Respondent 1 summarized their opinion as: "The JSON information is good, is there a way you can put it into a pretty sheet like the other stuff [referring to the PDF report]," supporting the usefulness of the detailed JSON information, but saying the format may be intimidating. When asked "Would a spreadsheet cause a less adverse reaction?", they responded: "If you just present it in a spreadsheet, there will be people who will consume it much more easily." Respondent 2 has a similar answer: "It needs to be able to be more easily consumable on a page." Respondent 3 supports the usefulness of the information provided in the reports from this project by saying "The information is of good value and it can be used, not just to improve the customer experience, but to also help the longevity of what we're doing with the customer." This points towards the satisfaction of Non-functional requirement 2, but the respondents did show that the JSON format might make the auditing process take more effort than if the information was presented differently.

6.1.3 Survey Question 3: How appropriate do you consider the method for calculating shared emissions?

The respondents also said that the method used for dividing shared emissions seems fair; Respondent 2 answered: "As long as we can describe our hypotheses and our methodology, and it's auditable and traceable, then it seems fair." respondent 3 said: "It's important when we have a company like *[company name]*, who use data services in Europe and Asia, but not in the Americas, we are tailoring the report to the infrastructure they use." Respondent 1 could not say whether they believed the method for sharing emissions to be fair, but they did show their concern for fairness by asking the following before the sharing method had been discussed: "How, on shared devices, like virtual servers, do you know how much electricity [each] customer's used?" This appears to satisfy Non-functional requirement 1.

6.1.4 General survey comments

As a general remark about the project, respondent 1 said: "Certainly very interesting, and ... there is probably someone, on a contractual level, within *[company name]*, who either has started having conversations about this, or has been having them with someone." Respondent 2 said: "I think this is a great piece of work, and I'd love to go on a journey with you and follow this through. Because ... we all know it's important, and it really resonates. If you look at *[company name]*'s top five objectives, the greenhouse credentials [are] up there." Respondent 3 said: "I can see the level [of detail] that goes into creating the results, which is great."

6.1.5 Remaining Requirements

After having validated Non-functional requirements 1 and 2 through the survey questions, this leaves Functional requirements 1 to 4 and Non-functional requirements 3 and 4 to be validated. Non-functional requirement 3 is satisfied by the extensive detail that has gone into finding potential methodologies, finding which ones work the best for this use case, and implementing them as closely to the original method as possible. Non-functional requirement 4 is satisfied as can be seen in Figure 5.2. Functional requirements 1 and 2 are satisfied through the implementation of the model, see Equation 4.1. Functional requirement 3 is satisfied as can be seen in the PDF report (Figure 5.2), but even more so in the JSON report (Appendix A). Functional requirement 4 is partially satisfied, while the reports are generated on a monthly basis, this is currently not done fully automatically. However, this could be implemented fairly easily on BT's side, by simply invoking the report generator on a new data set once per month and sending the reports to the tenant's email address.

6.2 Comparison Against Existing Tools

Compared to most other tools, such as the dashboards from AWS, GCP, Azure, and Thoughtworks, this one is slightly different. Where most tools aim to give the user a live view of the data on a dashboard, this tool presents the data in a monthly report. However, despite this difference in format, the information that is shown is still mostly matches the information from other tools, such as the historic graph, which is a staple in sustainability report, being present on all dashboards mentioned above, as well as the comparison of the emissions to more tangible things such as the amount of flights taken or smartphones charged, which is present on the

Thoughtworks dashboard. The distinction between gross and net emissions is present on most of these dashboards as well. The detailed breakdown of emissions per Scope and even per device type is more novel.

In terms of methodology, these tools are much harder to compare, since not all of them go into full detail about their methodology, and the underlying infrastructure is completely for each of these cloud service providers.

6.3 Limitations

While we tried to make the model as accurate and complete as possible, there are still several limitations. The first limitation of the model is the assumption of a constant carbon intensity during the month. Since all data is entered as a single figure which represents the average of the month, nuances such as the effect of a varying carbon intensity are lost. This is problematic when, for example, solar panels are used, which produce much less power during the night than during the day, thus the average carbon intensity will be higher. However, if the data center also operates under the heaviest load during the day, the lower carbon intensity should be (primarily) used.

Another limitation was found with the model to predict server energy consumption (see Section 4.3). There was not enough data available to perform a reliable linear regression, thus the values of the parameters that were obtained are likely to be inaccurate. Related to this limitation is the fact that one of the parameters that is measured from the servers, namely the cache data moved, was not possible to measure on the servers that are used in the data centers BT uses.

Furthermore, due to the bottom up approach regarding multitenancy, a device shared by two tenants is considered the same as two separate devices, each occupied by one tenant. To remedy this, a hybrid of top-down and bottom-up could be used, which first estimates the power consumption of each tenant on the device, then estimates the power consumption of the entire device without taking the multitenancy aspect in mind, and uses the ratio between the tenants' estimated power consumption to divide the device's estimate power consumption.

Finally, in terms of using the model and the report generator as a more general purpose tool, there are a few factors holding them back. Most importantly, both were designed with BT's cloud infrastructure in mind. Therefore, the model and the report generator are only applicable if the same data points are available. This requires a certain level of transparency from the data center operator, which may not always be available. Furthermore, the generator is built to produce monthly reports, though the model itself does not contain this limitation; the model produces a carbon footprint on the same timescale as the timescale of the input data. The model is fairly flexible in including various emission sources thanks to the general emissions formula (see Algorithm 6), however, if most emissions are placed in the miscellaneous category, the separation of emissions, which is shown in a pie graph on the report, is largely lost. The model does assume there are a set of server devices and network devices, however, in principle it is not a problem if there is only one or even zero of either of these devices. While the model and report generator were designed with multitenancy in mind, this is not a limitation; if the carbon footprint should be calculated for an entire data center, there will only be one ”tenant”, and all emissions will attributed to that ”tenant”.

Chapter 7

Conclusion

In this chapter, we will conclude this thesis by giving a short summary of the results, answering the research questions in Section 7.1, discussing the relation to existing evidence in Section 7.2, and the impact and implications of the results in Section 7.3

In this project, the author worked together with BT, who is a SaaS provider, to find a method to calculate the Total Carbon Footprint of each tenant in their data centers, focusing on their Cloud Contact Cisco (CCC) product as a starting point. Using existing methods from the literature, we put together a model that calculates this footprint. For some emission sources, emissions can be attributed to a specific tenant easily, by using their usage of the device to predict an amount of carbon emissions. For other emission sources, it is not clear which tenant is responsible for how much of the emissions. In such cases, the same share is attributed as the share of directly attributable emissions. This model was implemented in a program which generates a PDF report showing the monthly figures, and a JSON report which shows a more detailed breakdown of the emissions, down to each individual server device, on top of the same data as the PDF report. The project was evaluated with three of BT's customers. From this evaluation it became clear that they felt that it is an important project and a good first step, but improvements could be made. The thing that all respondents agreed on was that the JSON format can be intimidating for some people, and a more user friendly way to present that data would make the data more accessible, which is important because according to them, the detailed breakdown is very useful.

7.1 Answers to Research Questions

To answer Research question 1, Research questions 2 to 5 have to be answered first. Then by combining the answers, the answer to Research question 1 will be obtained.

To answer Research question 2: According to the GHG Protocol, as seen in Section 1.2, carbon emissions are divided into three scopes. Scope 1 is for direct emissions, Scope 2 is for emissions from energy consumption, and Scope 3 is for any other indirect emissions. For a data center, this means that direct emission sources, such as generators or other kinds of backup power systems should be included in Scope 1, all of the energy consuming devices, such as servers, network routers and switches, cooling systems, but also lighting and security systems, should be included in Scope 2, and finally, the production of the devices used in the data center, the transportation of these devices, the construction of the building, and other indirect emissions should be included in Scope 3. Furthermore, any offset methods that are applied, such as using green energy or buying Renewable Energy Certificates, can be subtracted from the Total Carbon Footprint to obtain the net emissions, as shown in Section 4.4. Net emissions are the emissions that should still be offset, should the company desire to be carbon neutral.

To answer Research question 3: For Scope 1 emissions, a carbon measuring device can be used, or a calculation that uses the amount of carbon emitted either per unit of energy produced or unit of fuel consumed to calculate the carbon emissions. This is shown in Equation 4.3. For Scope 2 emissions, power measurements using dedicated power measuring devices between the wall sockets and the devices would be the most accurate way of determining power consumption at a per-device level. However, installing these power measuring devices may be unfeasible, thus a model to estimate power consumption based on usage statistics can be used. As it turned out, due to the limited amount of data that was available to perform the regressions, this method was not the most accurate though. Another method would be to use the total power consumption of the data center, for example from the energy bill, and divide that according to usage statistics. In such a case it would be harder to design a splitting method, since the usage of a server is hard to compare against the usage of a network switch for example, but it could be a possibility. The possible ways to deal with Scope 2 are listed in Section 4.3, and the full equation is listed in Equation 4.4. For Scope 3 emissions, there is a full list in the GHG Protocol, which is also briefly listed in Section 1.2, however, it turned out in this project that most of those emissions were not (easily) obtainable. Since Scope 3 is currently an optional scope, a genuine best effort is good enough, however, to get the actual Total Carbon Footprint, all of these emissions should be included. The equation used to calculate Scope 3 is also listed in Equation 4.5.

To answer Research question 4: From Section 4.1, responsibility is an unsolved and not much researched problem, however, it does make a lot of sense for entities to share the responsibility of the Total Carbon Footprint. Lenzen et al. did discuss responsibility sharing, but they did not discuss about how to set the percentage shares for each involved party. Thus, more research is probably needed to answer this question, however, two straightforward methods are to share the responsibilities equally between all involved parties, or to push the responsibility to the tenants.

To answer Research question 5: Because of the bottom up approach, servers and network devices can have their energy consumption calculated per tenant according to each tenant's individual usage of these devices. For other devices for which it is not possible to trace back each tenant's individual usage, the energy consumption, and thereby the carbon emissions, can be divided according to the same percentage of total server and network device energy the tenant is responsible for. According to the subjects who participated in the case study, this method seems fair (see Section 6.1.3).

To summarize, and to answer Research question 1: To accurately and fairly attribute the Total Carbon Footprint to each tenant in a multitenant data center, all scopes defined by the Greenhouse Gas Protocol should be included. Furthermore, to most accurately obtain the emissions, the carbon emissions or power consumption should be measured directly, but in case that is infeasible, a model to estimate power consumption or carbon emissions can be used. As for who is responsible for these emissions, this is still a problem that requires more research, but two options are splitting the responsibility equally, or pushing the responsibility to the tenants. Finally, to assign each tenant a most fair share of the emissions of the data center, a bottom up approach works best, since the individual usage of each tenant will then occur in the equations naturally. For emissions that cannot be calculated bottom up, sharing them according to the same percentage as the emissions that can be calculated bottom up seems a fair approach.

7.2 Relation to Existing Evidence

Compared to other approaches in the past, this approach is a more holistic one. Previous works have often focused on measuring or calculating the carbon emissions from a specific part of the data center, mainly on the servers themselves, such as the works from Da Silva et al. [15], Bohra and Chaudhary [18], and Kansal et al. [21], but also often on the network aspect, such as the works from Mahadevan et al. [29] and Aslan et al.[2]. The difference with this work is that, rather than creating an entirely new method, this project combines existing methods into a more complete model. Furthermore, existing works have been mostly in an academic context, whereas this work cooperated with a SaaS provider to tailor the product to what their requirements are, and validated those with three of their customers.

7.3 Impact/Implications

The main impact of this work is the addition of multitenancy to the carbon footprint models. Especially with hyperscalers gaining more ground, multitenancy is a very important aspect of cloud computing. In terms of multitenancy, it became clear that using a bottom up approach is easier when possible. When combined with a model that estimates the power consumption of each emission source in Scope 2, this allows each tenant's usage statistics to be integrated deeply into the model. However, this method is highly dependent on the accuracy of said models, which, as shown in Section 4.3, requires a lot more input data than was available in this project. Breaking a total energy consumption of an entire data center down by usage statistics might seem more accurate in such a case, but the difficulty there is that the usage statistics of different device types are difficult to compare. With more work, both methods could work, and more in-depth research would be needed to accurately conclude which method works best in which situations.

Chapter 8

Future Work

In this chapter we will discuss some potential future work that could be done to further the state of the art on this topic, as well as some improvements that could be made with regards to the methodology of this project.

As mentioned in Section 6.3, the accuracy of the model leaves much to be desired due to limited input data for the regressions. To solve this, a more in-depth investigation would need to be performed into the power consumption of BT's servers under various loads. This can be done in two ways. The first way is to let data be collected over n months after which the first $n-1$ months are used to determine the model parameters, and the last month is used to calculate the accuracy. The second way is to collect data from more servers, as currently only the usage data from three tenants was used, and then use an x percent of the servers to calculate the model parameters, and use the final $100-x$ percent to calculate the accuracy. As was also mentioned, currently the cache data is not available to the model, which is likely to lead to even more inaccuracy. Ideally, adding cache data to the model would be the solution, but if this is not possible, another model could be considered.

Another part of the Total Carbon Footprint that can be explored is the responsibility. Currently, the carbon footprint is pushed entirely onto the tenants, however, as discussed by [27], assigning the tenant a share of the emissions is perhaps a more appropriate method. In this case, more research is needed as to which percentage share works best.

The current Scope 3 emissions should be expanded as well. Currently, Scope 3 is only provided as a single total amount of emissions per month. While this does not affect the calculations for the Total Carbon Footprint itself, it does limit the auditability, since it is not clear where those emissions come from and it cannot be verified whether the numbers are correct.

Similarly, since BT does not have any Scope 1 emissions, those were not investigated in great detail in this project, however, in a more general case of a data center, there might be emission sources which do fall under Scope 1, such as the use of diesel generators to cope with a loss of electricity. Therefore, to generalize this model, Scope 1 should be researched more and the findings should be implemented into the model.

Finally, currently it is the case that data has to be manually inserted in the form of CSV files into the sustainability report generator. Using some automation tools would be beneficial for BT to automate the data insertion. Just like automating the input, automating the handling of the output would be a useful addition too. BT's ultimate use for the generated reports is to send them to their customers. Currently, there is no automated process to do this; sending each report to the correct customer is a manual process. Not only is this time-consuming, it is also more error prone than an automated process.

Chapter 9

Bibliography

[1] Andrew Armstrong and Amaya Souarez. "Guide for assessing GHG emissions of Cloud Computing and Data Center Services". In: *ICT Sector Guidance built on the GHG Protocol Product Life Cycle Accounting and Reporting Standard*. Global e-Sustainability Initiative and Carbon Trust, July 2017. Chap. 4 (cit. on p. 12).

[2] Joshua Aslan et al. "Electricity Intensity of Internet Data Transmission: Untangling the Estimates". In: *Journal of Industrial Ecology* 22.4 (2018), pp. 785–798. DOI: 10.1111/jiec.12630. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/jiec.12630. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/jiec.12630 (cit. on pp. 11, 22, 34).

[3] Victor Avelar, Dan Azevedo, and Alan French. *PUE: A comprehensive examination of the metric*. Whitepaper. 2012 (cit. on p. 12).

[4] Pankaj Bahtia and Janet Ranganathan. *The Greenhouse Gas Protocol: A Corporate Accounting and Reporting Standard (Revised Edition)*. Corporate Standard. World Resources Institute and World Business Council for Sustainable Development, Mar. 2004 (cit. on pp. 4, 7, 18).

[5] Thar Baker et al. "Energy Efficient Cloud Computing Environment via Autonomic Meta-director Framework". In: *2013 Sixth International Conference on Developments in eSystems Engineering*. 2013, pp. 198–203. DOI: 10.1109/DeSE.2013.43 (cit. on p. 11).

[6] Ilyas Bambrik. "A Survey on Cloud Computing Simulation and Modeling". In: *SN Computer Science* 1.249 (2020). DOI: 10.1007/s42979-020-00273-1 (cit. on p. 13).

[7] Lotfi Belkhir and Ahmed Elmeligi. "Assessing ICT global emissions footprint: Trends to 2040 & recommendations". In: *Journal of Cleaner Production* 177 (2018), pp. 448–463. ISSN: 0959-6526. DOI: 10.1016/j.jclepro.2017.12.239. URL: https://www.sciencedirect.com/science/article/pii/S095965261733233X (cit. on p. 12).

[8] Anton Beloglazov et al. "Chapter 3 - A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems". In: ed. by Marvin V. Zelkowitz. Vol. 82. Advances in Computers. Elsevier, 2011, pp. 47–111. DOI: https://doi.org/10.1016/B978-0-12-385512-1.00003-7. URL: https://www.sciencedirect.com/science/article/pii/B9780123855121000037 (cit. on p. 4).

[9] Pankaj Bhatia and Janet Ranganathan. *Corporate Value Chain (Scope 3) Accounting and Reporting Standard*. Corporate Standard. World Resources Institute and World Business Council for Sustainable Development, Sept. 2011 (cit. on p. 5).

[10] Ian F. Bitterlin. *Green IT: Managing your Carbon Footprint*. Ed. by The Chartered Institute for IT BCS. British Informatics Society Limited, May 2012, p. 68. ISBN: 9781780171340 (cit. on p. 4).

[11] Rodrigo N. Calheiros et al. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms". In: *Software: Practice and Experience* 41.1 (2011), pp. 23–50. DOI: 10.1002/spe.995. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/spe.995. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.995 (cit. on p. 13).

[12] Gabriel G. Castañé et al. "E-mc2: A formal framework for energy modelling in cloud computing". In: *Simulation Modelling Practice and Theory* 39 (2013). S.I.Energy efficiency in grids and clouds, pp. 56–75. ISSN: 1569-190X. DOI: 10.1016/j.simpat.2013.05.002. URL: https://www.sciencedirect.com/science/article/pii/S1569190X13000816 (cit. on p. 14).

[13] Andrew A. Chien, Rich Wolski, and Fan Yang. "Zero-Carbon Cloud: A Volatile Resource for High-Performance Computing". In: *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing.* 2015, pp. 1997–2001. DOI: 10.1109/CIT/IUCC/DASC/PICOM.2015.296 (cit. on p. 13).

[14] G. Contreras and M. Martonosi. "Power prediction for Intel XScale processors using performance monitoring unit events". In: *ISLPED '05. Proceedings of the 2005 International Symposium on Low Power Electronics and Design, 2005.* 2005, pp. 221–226. DOI: 10.1109/LPE.2005.195518 (cit. on p. 10).

[15] Rafael Feirrera da Silva et al. "Characterizing, Modeling, and Accurately Simulating Power and Energy Consumption of I/O-intensive Scientific Workflows". In: *Journal of Computational Science* 44 (2020), pp. 101–157. ISSN: 1877-7503. DOI: 10.1016/j.jocs.2020.101157. URL: https://www.sciencedirect.com/science/article/pii/S1877750320304580 (cit. on pp. 10, 21, 34).

[16] Sukhpal Singh Gill and Rajkumar Buyya. "A Taxonomy and Future Directions for Sustainable Cloud Computing: 360 Degree View". In: *ACM Comput. Surv.* 51.5 (Dec. 2018). ISSN: 0360-0300. DOI: 10.1145/3241038. URL: https://doi.org/10.1145/3241038 (cit. on pp. 4, 11).

[17] N Gupta et al. "Cloud computing: a walk through in models, challenges, and energy solutions". In: *IOP Conference Series: Materials Science and Engineering* 1022.1 (Jan. 2021), p. 012070. DOI: 10.1088/1757-899x/1022/1/012070. URL: https://doi.org/10.1088/1757-899x/1022/1/012070 (cit. on p. 13).

[18] Ata E Husain Bohra and Vipin Chaudhary. "VMeter: Power modelling for virtualized clouds". In: *2010 IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW).* 2010, pp. 1–8. DOI: 10.1109/IPDPSW.2010.5470907 (cit. on pp. 10, 22, 34).

[19] Erik Jagroep et al. "A Resource Utilization Score for Software Energy Consumption". In: *Proceedings of ICT for Sustainability 2016.* Atlantis Press, Aug. 2016, pp. 19–28. ISBN: 978-94-6252-224-4. DOI: 10.2991/ict4s-16.2016.3. URL: https://doi.org/10.2991/ict4s-16.2016.3 (cit. on p. 10).

[20] Erik Jagroep et al. "Awakening Awareness on Energy Consumption in Software Engineering". In: *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Society Track (ICSE-SEIS).* 2017, pp. 76–85. DOI: 10.1109/ICSE-SEIS.2017.10 (cit. on p. 4).

[21] Aman Kansal et al. "Virtual Machine Power Metering and Provisioning". In: *Proceedings of the 1st ACM Symposium on Cloud Computing.* SoCC '10. Indianapolis, Indiana, USA: Association for Computing Machinery, 2010, pp. 39–50. ISBN: 9781450300360. DOI: 10.1145/1807128.1807136. URL: https://doi.org/10.1145/1807128.1807136 (cit. on pp. 10, 22, 34).

[22] Atefeh Khosravi, Lachlan L. H. Andrew, and Rajkumar Buyya. "Dynamic VM Placement Method for Minimizing Energy and Carbon Cost in Geographically Distributed Cloud Data Centers". In: *IEEE Transactions on Sustainable Computing* 2.2 (2017), pp. 183–196. DOI: 10.1109/TSUSC.2017.2709980 (cit. on p. 12).

[23] Atefeh Khosravi, Saurabh Kumar Garg, and Rajkumar Buyya. "Energy and Carbon-Efficient Placement of Virtual Machines in Distributed Cloud Data Centers". In: *Euro-Par 2013 Parallel Processing*. Ed. by Felix Wolf, Bernd Mohr, and Dieter an Mey. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 317–328. ISBN: 978-3-642-40047-6 (cit. on p. 12).

[24] Dzimitry Kliazovich, Pascal Bouvry, and Samee Ullah Khan. "GreenCloud: a Packet-Level Simulator of Energy-Aware Cloud Computing Data Centers". In: *The Journal of Supercomputing* 62 (Nov. 2010), pp. 1263–1283. DOI: 10.1007/s11227-010-0504-1 (cit. on p. 13).

[25] Patricia Lago et al. "Framing Sustainability as a Property of Software Quality". In: *Commun. ACM* 58.10 (Sept. 2015), pp. 70–78. ISSN: 0001-0782. DOI: 10.1145/2714560. URL: https://doi.org/10.1145/2714560 (cit. on p. 3).

[26] Manfred Lenzen and Joy Murray. "Conceptualising environmental responsibility". In: *Ecological Economics* 70.2 (2010). Special Section: Ecological Distribution Conflicts, pp. 261–270. ISSN: 0921-8009. DOI: 10.1016/j.ecolecon.2010.04.005. URL: https://www.sciencedirect.com/science/article/pii/S092180091000128X (cit. on pp. 6, 18).

[27] Manfred Lenzen et al. "Shared producer and consumer responsibility — Theory and practice". In: *Ecological Economics* 61.1 (2007), pp. 27–42. ISSN: 0921-8009. DOI: 10.1016/j.ecolecon.2006.05.018. URL: https://www.sciencedirect.com/science/article/pii/S0921800906002953 (cit. on pp. 5, 36).

[28] Weiwei Lin et al. "A cloud server energy consumption measurement system for heterogeneous cloud environments". In: *Information Sciences* 468 (2018), pp. 47–62. ISSN: 0020-0255. DOI: 10.1016/j.ins.2018.08.032. URL: https://www.sciencedirect.com/science/article/pii/S0020025518306364 (cit. on p. 10).

[29] Priya Mahadevan et al. "A Power Benchmarking Framework for Network Devices". In: *NETWORKING 2009*. Ed. by Luigi Fratta et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 795–808 (cit. on pp. 11, 22, 34).

[30] Asad W. Malik et al. "CloudNetSim++: A GUI Based Framework for Modeling and Simulation of Data Centers in OMNeT++". In: *IEEE Transactions on Services Computing* 10.4 (2017), pp. 506–519. DOI: 10.1109/TSC.2015.2496164 (cit. on p. 14).

[31] Toni Mastelic et al. "Cloud Computing: Survey on Energy Efficiency". In: *ACM Comput. Surv.* 47.2 (Dec. 2014). ISSN: 0360-0300. DOI: 10.1145/2656204. URL: https://doi.org/10.1145/2656204 (cit. on p. 4).

[32] Fabian Mastenbroek et al. "OpenDC 2.0: Convenient Modeling and Simulation of Emerging Technologies in Cloud Datacenters". In: *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. 2021, pp. 455–464. DOI: 10.1109/CCGrid51090.2021.00055 (cit. on p. 14).

[33] H. Scott Matthews, Chris T. Hendrickson, and Christopher L. Weber. "The Importance of Carbon Footprint Estimation Boundaries". In: *Environmental Science & Technology* 42.16 (2008). PMID: 18767634, pp. 5839–5842. DOI: 10.1021/es703112w. eprint: https://doi.org/10.1021/es703112w. URL: https://doi.org/10.1021/es703112w (cit. on pp. 5, 18).

[34] Peter Mell and Timothy Grance. *The NIST Definition of Cloud Computing.* Corporate Definition. U.S. National Institute of Standards and Technology, Sept. 2011. URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf (cit. on p. 3).

[35] *The Carbon Benefits of Cloud Computing: A Study on the Microsoft Cloud in Partnership with WSP.* Whitepaper. Microsoft, 2020 (cit. on p. 9).

[36] *Cloud Computing and Sustainability: The Environmental Benefits of Moving to the Cloud.* Whitepaper. Microsoft, Nov. 2010 (cit. on p. 9).

[37] Fahimeh Alizadeh Moghaddam, Patricia Lago, and Paola Grosso. "Energy-Efficient Networking Solutions in Cloud-Based Environments: A Systematic Literature Review". In: *ACM Comput. Surv.* 47.4 (May 2015). ISSN: 0360-0300. DOI: 10.1145/2764464. URL: https://doi.org/10.1145/2764464 (cit. on p. 4).

[38] Tridib Mukherjee et al. "Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers". In: *Computer Networks* 53.17 (2009). Virtualized Data Centers, pp. 2888–2904. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2009.06.008. URL: https://www.sciencedirect.com/science/article/pii/S1389128609002047 (cit. on p. 13).

[39] Alberto Núñez et al. "iCanCloud: A Flexible and Scalable Cloud Infrastructure Simulator". In: *Journal of Grid Computing* 10 (2012), pp. 185–209. DOI: 10.1007/s10723-012-9208-5 (cit. on p. 14).

[40] Panos Patros et al. "Toward Sustainable Serverless Computing". In: *IEEE Internet Computing* 25.6 (2021), pp. 42–50. DOI: 10.1109/MIC.2021.3093105 (cit. on p. 11).

[41] Per Runeson and Martin Höst. "Guidelines for Conducting and Reporting Case Study Research in Software Engineering". In: *Empirical Softw. Engg.* 14.2 (Apr. 2009), pp. 131–164. ISSN: 1382-3256. DOI: 10.1007/s10664-008-9102-8. URL: https://doi.org/10.1007/s10664-008-9102-8 (cit. on p. 15).

[42] Brendan Sissan et al. *Sustainability Pillar: AWS Well-Architected Framework.* Whitepaper. Amazon, Dec. 2021 (cit. on p. 8).

[43] Bud E. Smith. *Green Computing: Tools and Technologies for Saving Energy, Money and Resources.* New York: Auerbach Publications, 2014. ISBN: 9780429088551. DOI: 10.1201/b15098 (cit. on p. 4).

[44] James William Smith et al. "CloudMonitor: Profiling Power Usage". In: *2012 IEEE Fifth International Conference on Cloud Computing.* 2012, pp. 947–948. DOI: 10.1109/CLOUD.2012.112 (cit. on p. 10).

[45] Mary Sotos. *GHG Protocol Scope 2 Guidance.* Corporate Standard. World Resources Institute and World Business Council for Sustainable Development, 2015 (cit. on p. 4).

[46] Paul Steenhof et al. "A protocol for quantifying the carbon reductions achieved through the provision of low or zero carbon ICT services". In: *Sustainable Computing: Informatics and Systems* 2.1 (2012), pp. 23–32. ISSN: 2210-5379. DOI: 10.1016/j.suscom.2012.01.001. URL: https://www.sciencedirect.com/science/article/pii/S2210537912000029 (cit. on p. 11).

[47] Paul Steenhof et al. "Estimating Emission Reduction from Low Carbon Information Technology: The GeoChronos Relocation Project". In: *Advances in Cloud Computing Research.* Ed. by Muthu Ramachandran. Nova Science Publishers, 2014. Chap. 7, pp. 149–172. ISBN: 978-1-63117-192-5 (cit. on p. 11).

[48] Pericherla S Suryateja. "A Comparative Analysis of Cloud Simulators." In: *International Journal of Modern Education & Computer Science* 8.4 (2016) (cit. on p. 13).

[49] Wenhong Tian et al. “A Toolkit for Modeling and Simulation of Real-Time Virtual Machine Allocation in a Cloud Data Center”. In: *IEEE Transactions on Automation Science and Engineering* 12.1 (2015), pp. 153–161. DOI: 10.1109/TASE.2013.2266338 (cit. on p. 14).

[50] *Offsets and RECs: What’s the Difference?* Whitepaper. U.S. Environmental Protection Agency, Feb. 2018. URL: https://www.epa.gov/sites/default/files/2018-03/documents/gpp_guide_recs_offsets.pdf (cit. on p. 3).

[51] Thiago Lara Vasques, Pedro Moura, and Aníbal de Almeida. “A Review on Energy Efficiency and Demand Response with focus on Small and Medium Data Centers”. In: *Energy Efficiency* 12 (2018), pp. 1399–1423. DOI: 10.1007/s12053-018-9753-2 (cit. on p. 10).

[52] Usman Wajid et al. “On Achieving Energy Efficiency and Reducing CO_2 Footprint in Cloud Computing”. In: *IEEE Transactions on Cloud Computing* 4 (July 2015), pp. 5–6. DOI: 10.1109/TCC.2015.2453988 (cit. on p. 12).

[53] Minxian Xu and Rajkumar Buyya. “Managing renewable energy and carbon footprint in multi-cloud computing environments”. In: *Journal of Parallel and Distributed Computing* 135 (2020), pp. 191–202. ISSN: 0743-7315. DOI: j.jpdc.2019.09.015. URL: https://www.sciencedirect.com/science/article/pii/S0743731519303132 (cit. on p. 13).

[54] Muhammad Zakarya and Lee Gillam. “Modelling resource heterogeneities in cloud simulations and quantifying their accuracy”. In: *Simulation Modelling Practice and Theory* 94 (2019), pp. 43–65. ISSN: 1569-190X. DOI: 10.1016/j.simpat.2019.02.003. URL: https://www.sciencedirect.com/science/article/pii/S1569190X19300140 (cit. on p. 13).

Appendix A

JSON report

```
{
  "avgCarbonIntensity": 0.4,
  "greenEnergyPercentage": 50.0,
  "customer": {
    "id": "1234",
    "name": "Fictitious Customer",
    "numAgents": 200
  },
  "period": {
    "name": "jun",
    "startDate": 1651356000,
    "endDate": 1653948000,
    "footprint": {
      "gross": 2.0,
      "net": 1.0,
      "unit": "METRIC_TONS"
    }
  },
  "emissions": {
    "total": {
      "gross": 2.0,
      "net": 1.0,
      "unit": "METRIC_TONS"
    },
    "perAgent": {
      "gross": 0.01,
      "net": 0.005,
      "unit": "METRIC_TONS"
    },
    "offsets": {
      "rec": 0.0,
      "greenEnergy": 1.0,
      "unit": "METRIC_TONS"
    }
  },
  "history": {
    "prevPrevMonth": {
      "name": "apr",
      "startDate": 1646089200,
      "endDate": 1648677600,
      "footprint": {
        "gross": 2.5,
        "net": 1.5,
        "unit": "METRIC_TONS"
      }
    },
    "prevMonth": {
      "name": "may",
      "startDate": 1648764000,
      "endDate": 1651269600,
      "footprint": {
        "gross": 2.2,
        "net": 1.1,
        "unit": "METRIC_TONS"
      }
    },
    "currMonth": {
      "name": "jun",
      "startDate": 1651356000,
      "endDate": 1653948000,
      "footprint": {
        "gross": 2.0,
        "net": 1.0,
        "unit": "METRIC_TONS"
      }
    },
    "massUnit": "METRIC_TONS"
  },
  "perspectives": {
    "car": {
      "equivalent": 7986.901481570225,
      "carbonIntensity": 250.41,
      "link": "https://www.epa.gov/energy/greenhouse-gases-
         ↪ equivalencies-calculator-calculations-and-references\\#
         ↪ miles"
    },
    "flights": {
      "equivalent": 6.283380459,
      "carbonIntensity": 318300.0,
      "link": "https://www.icao.int/environmental-protection/
         ↪ Carbonoffset/Pages/default.aspx"
    },
    "smartphones": {
      "equivalent": 243902.4390,
      "carbonIntensity": 8.22,
      "link": "https://www.epa.gov/energy/greenhouse-gases-
         ↪ equivalencies-calculator-calculations-and-references\\#
         ↪ smartphones"
    }
  },
  "emissionSourceGroups": {
    "servers": {
      "emissions": 0.6,
      "emissionUnit": "METRIC_TONS",
      "energy": 1.5,
      "energyUnit": "MEGAWATT_HOURS"
    },
    "network": {
      "emissions": 0.6,
      "emissionUnit": "METRIC_TONS",
      "energy": 1.5,
      "energyUnit": "MEGAWATT_HOURS"
    },
    "cooling": {
      "emissions": 0.6,
      "emissionUnit": "METRIC_TONS",
      "energy": 1.5,
      "energyUnit": "MEGAWATT_HOURS"
    },
    "miscScope1": {
      "emissions": 0.0,
      "emissionUnit": "METRIC_TONS",
      "energy": 0.0,
      "energyUnit": "MEGAWATT_HOURS"
    },
    "miscScope2": {
      "emissions": 0.0,
      "emissionUnit": "METRIC_TONS",
      "energy": 0.0,
      "energyUnit": "MEGAWATT_HOURS"
    },
    "miscScope3": {
      "emissions": 0.2,
      "emissionUnit": "METRIC_TONS",
      "energy": 0.5,
      "energyUnit": "MEGAWATT_HOURS"
    }
  },
  "energy": {
    "total": {
      "gross": 5.0,
      "net": 2.5,
      "unit": "MEGAWATT_HOURS"
    },
    "green": 2.5,
    "rec": 0.0,
    "energyUnit": "MEGAWATT_HOURS"
  },
  "breakdown": {
    "DC_1234": {
      "isAggregate": false,
      "carbonIntensity": 0.4,
      "name": "Fictitious Data Center",
      "pue": 1.5,
      "directlyAssignableEEO": {
        "isAggregate": true,
        "energy": 120000.0,
        "emissions": 48000.0,
        "offsets": {
          "rec": 0.0,
          "greenEnergy": 24000.0
        }
      },
      "totalDatacenterEEO": {
        "isAggregate": true,
        "energy": 12000000.0,
        "emissions": 4800000.0,
        "offsets": {
          "rec": 0.0,
          "greenEnergy": 240000.0
        }
      },
      "scopes": {
        "scope1": {
          "type": "pipeline.datatypes.Scope1",
          "isAggregate": false,
          "energy": 0.0,
          "emissions": 0.0
        },
        "scope2": {
          "type": "pipeline.datatypes.Scope2",
          "isAggregate": false,
          "energy": 4500000,
          "emissions": 1800000,
          "devices": {
            "servers": {
              "SERVER_1234": {
                "type": "pipeline.datatypes.ServerDevice",
                "isAggregate": false,
                "deviceModel": "ABC_987",
                "energy": 100000.0,
                "emissions": 40000.0,
                "utilization": 0.10,
                "cacheMoved": 2e7,
                "dramAccessed": 5e9,
                "diskMoved": 2e10
              },
            },
            "network": {
              "NETWORK_DEVICE_1234": {
                "type": "pipeline.datatypes.NetworkDevice",
                "isAggregate": false,
                "deviceType": "router",
                "energy": 100000.0,
                "emissions": 40000.0,
                "bytesSent": 1e12,
                "bytesReceived": 1e12
              },
            },
            "cooling": {
              "total_cooling": {
                "type": "pipeline.datatypes.MiscDevice",
                "isAggregate": true,
                "deviceType": "cooling",
                "energy": 50000.0,
                "emissions": 20000.0
              }
            }
          }
        },
        "scope3": {
          "type": "pipeline.datatypes.Scope3",
          "isAggregate": false,
          "energy": 0.5,
          "emissions": 0.2,
          "sources": {
            "misc": {
              "type": "pipeline.datatypes.MiscDevice",
              "isAggregate": false,
              "deviceType": "misc",
              "energy": 0.5,
              "emissions": 0.2
            }
          }
        }
      }
    }
  }
}
```

Appendix B

Report Template

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TO BE COMPILED WITH LUALATEX                                  %
% Also requires either the BT Curve font to be installed on the %
% system, or the ttf files to be in the working directory.      %
% To tell this document to use the system font, set             %
% USE_SYSTEM_BT_FONT=true in your system's environment vars,    %
% if this variable is not found or not set to true (exact,      %
% all lowercase), it will automatically try to use the ttf.     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\documentclass[a4paper, 11pt]{article}

\usepackage[margin=1.3cm]{geometry}
\usepackage{xcolor}
\usepackage{anyfontsize}
\usepackage[colorlinks]{hyperref}
\usepackage{graphicx}
\usepackage{fancyhdr}
\usepackage{todonotes}
\usepackage{float}
\usepackage{tikz}
\usepackage{pgfplots}
\usepackage{pgf-pie}
\usepackage{fontspec}
\usepackage{ifthen}
\usepackage{catchfile}
\usepackage{tabularx}
\usepackage{pbox}
\usepackage{xurl}
\usepackage{titlesec}
\usepackage{multicol}
\usepackage{vwcol}
\usepackage{fontawesome5}
% enable footnotes in tables and tabulars
\usepackage{footnote}
\makesavenoteenv{tabular}
\makesavenoteenv{table}
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% font settings                                                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% determine whether to use the system font by checking USE_SYSTEM_BT_
   ↪ FONT
\CatchFileEdef{\usesysfont}{|"kpsewhich -var-value=USE_SYSTEM_BT_FONT
   ↪ "}{\endlinechar=-1 }
% if it is true, use the system font
\ifthenelse{\equal{\usesysfont}{true}}{
\setmainfont{BT Curve}
\newfontfamily{\btheader}{BT Curve}[UprightFont = * Headline]
} % if not, use the .ttf files we received directly
{
\setmainfont{BTCurve_}[
    Path = BT_Curve/,
    Extension = .ttf,
    UprightFont = *Rg,
    BoldFont = *Bd,
    ItalicFont = *It,
    BoldItalicFont = *BdIt,
]
\newfontfamily{\btheader}{BTCurve_Headline}[Path=BT_Curve/, Extension
   ↪ = .ttf,]
}

\renewcommand{\footnotesize}{\scriptsize}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% length settings                                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\setlength{\voffset}{58pt}
\addtolength{\voffset}{-1in}

\setlength{\headheight}{58pt}

\setlength{\headsep}{20pt}

\setlength{\topmargin}{-\headheight}

\newlength{\lsmargin}
\setlength{\lsmargin}{\oddsidemargin}
\addtolength{\lsmargin}{\hoffset}
\addtolength{\lsmargin}{1in}

\newlength{\rsmargin}
\setlength{\rsmargin}{\evensidemargin}
\addtolength{\rsmargin}{\marginparwidth}
\addtolength{\rsmargin}{-\marginparsep}
\addtolength{\rsmargin}{1cm}
```

```
\setlength{\parindent}{0pt}

\setlength{\tabcolsep}{0pt} % Default value: 6pt

\setlength{\footskip}{0pt}

% fixes footnotes being pushed off the bottom of the page
\addtolength{\textheight}{-\headheight}
\addtolength{\textheight}{24pt}

\setlength{\columnsep}{1cm}


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% color settings                                                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\definecolor{btpurple}{HTML}{6300a9}
\definecolor{btpurple2}{HTML}{461195}
\definecolor{btpurple3}{HTML}{5514b4}
\definecolor{btpurple4}{HTML}{bfa7e3}

\hypersetup{allcolors=btpurple}


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% fancy page settings                                           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\pagestyle{fancy}
\fancyhf{}%

\fancyhead[L]{\colorbox{btpurple}{\parbox[b][\headheight]{\headwidth
   ↪ }{\vfill\hspace{\lsmargin}\includegraphics[height=0.9\
   ↪ headheight]{bt-logo.jpg}\vfill}}}%

\fancyhead[R]{\parbox{0.5\linewidth}{\raggedleft\huge\textcolor{white
   ↪ }{\textbf{
    \btheader Sustainability Report\\
    \vspace{1mm}
    \large \detokenize{@placeholder:"customer.name"} \\
    \vspace{-4mm}
    % the negative hspace has to be at the end to make sure the last
       ↪ line is right aligned with the other two
    \large \detokenize{@placeholder:"period.start"} - \detokenize{
       ↪ @placeholder:"period.end"} \hspace{-0.1cm}
}}}\hspace{\rsmargin}\vspace{2mm}}%


\fancyhfoffset[L]{\lsmargin}
\fancyhfoffset[R]{\rsmargin}

\renewcommand{\headrulewidth}{0pt}%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% additional helper commands                                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\newcommand{\cotwo}[0]{CO$_\mathrm{2}$}
\newcommand{\answer}[1]{\textit{A:}\hspace{2mm} \parbox[t]{\linewidth
   ↪ -0.9cm}{#1}}


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% section header formatting                                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% sections:
\titlespacing*{\section}{0pt}{5mm}{1mm}
\titleformat{\section}{\LARGE\btheader\color{btpurple}}{}{0em}{}
% subsections:
\titlespacing*{\subsection}{0pt}{4mm}{1mm}


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% begin document                                                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\begin{document}
\fontsize{10}{12}\selectfont


%%%%%%%%%%%%%%%%%%%%% measurement scope tag %%%%%%%%%%%%%%%%%%%%%
\vspace*{-1.1cm}
\begin{center}\fbox{\large Measurement Scope: Scope 3 Upstream}\end{
   ↪ center}


%%%%%%%%%%%%%%%%%%%%%%% overview section %%%%%%%%%%%%%%%%%%%%%%%%
\begin{multicols}{2}

\vspace{-0.8cm}
\section{Overview}
\vspace*{-0.4cm}
{
\fontsize{20}{24}\selectfont
\renewcommand{\arraystretch}{0.8} % Default value: 1
\begin{tabular}{r l@{\hspace{1cm}} r l}
    \multicolumn{4}{l}{\large Total Carbon Footprint (TCFP):} \\
    \detokenize{@placeholder:"footprint.total.gross"} & ~\detokenize{
       ↪ @placeholder:"footprint.massUnit.abbreviation"}{\large
       ↪ gross} & \detokenize{@placeholder:"footprint.total.net"} &
       ↪ ~\detokenize{@placeholder:"footprint.massUnit.abbreviation
       ↪ "}{\large net} \\
    \multicolumn{4}{l}{\large Average per agent (based on \detokenize
       ↪ {@placeholder:"customer.numAgents"} agents):} \\
    \detokenize{@placeholder:"footprint.perAgent.gross"} & ~\
       ↪ detokenize{@placeholder:"footprint.massUnit.abbreviation
       ↪ "}{\large gross} & \detokenize{@placeholder:"footprint.
       ↪ perAgent.net"} & ~\detokenize{@placeholder:"footprint.
       ↪ massUnit.abbreviation"}{\large net}
\end{tabular}
}


%%%%%%%%%%%%%%%%%%%%%%% historical graph %%%%%%%%%%%%%%%%%%%%%%%
\vspace{-0.3cm}
\begin{figure}[H]
    \begin{tikzpicture}
        \begin{axis}[ybar,
            symbolic x coords={@placeholder:"history.prevPrevMonth.
               ↪ name", @placeholder:"history.prevMonth.name",
               ↪ @placeholder:"history.currMonth.name"},
            xtick=data,
            ytick=\empty,
            ylabel={\detokenize{@placeholder:"history.massUnit.
               ↪ unitName"} of CO$_\textrm{2}$},
            xlabel={Carbon footprint over the past three months},
            axis x line*=bottom,
            y axis line style={draw=none},
            ymin = 0,
            nodes near coords={\pgfmathprintnumber[precision=10,
               ↪ assume math mode=true]{\pgfplotspointmeta}},
            bar width=22pt,
            enlarge x limits=0.2,
            width=0.85\linewidth,
            height=4cm,
            legend style={draw=none, at={(1,1)}, anchor=north west}
        ]
        \addplot[fill=btpurple, draw opacity=0] coordinates {
            (@placeholder:"history.prevPrevMonth.name", @placeholder
               ↪ :"history.prevPrevMonth.footprint.gross")
            (@placeholder:"history.prevMonth.name", @placeholder:"
               ↪ history.prevMonth.footprint.gross")
            (@placeholder:"history.currMonth.name", @placeholder:"
               ↪ history.currMonth.footprint.gross")
        };
        \addplot[fill=btpurple4, draw opacity=0] coordinates {
            (@placeholder:"history.prevPrevMonth.name", @placeholder
               ↪ :"history.prevPrevMonth.footprint.net")
            (@placeholder:"history.prevMonth.name", @placeholder:"
               ↪ history.prevMonth.footprint.net")
            (@placeholder:"history.currMonth.name", @placeholder:"
               ↪ history.currMonth.footprint.net")
        };
        \legend{Gross, Net};
        \end{axis}
        \end{tikzpicture}
    \label{fig:history}
\end{figure}


%%%%%%%%%%%%%%%%%%%%%% perspective section %%%%%%%%%%%%%%%%%%%%%%
\section{Putting your emissions into perspective}
\detokenize{@placeholder:"footprint.total.gross"} \detokenize{
   ↪ @placeholder:"footprint.massUnit.abbreviation"} \cotwo{} is
   ↪ equivalent to:

\vspace*{2mm}
{
\renewcommand{\arraystretch}{1.25}
\begin{tabular}{c@{\hspace{3.5mm}}l}
    \faIcon{car} & \detokenize{@placeholder:"perspective.carKm"} km
       ↪ driven by a car\footnote{Calculated using data from \href{
       ↪ https://www.epa.gov/energy/greenhouse-gases-equivalencies-
       ↪ calculator-calculations-and-references\#miles}{epa.gov}.}.
       ↪ \\
    \faIcon{plane} & \detokenize{@placeholder:"perspective.flights"}
       ↪ flights from Amsterdam to New York\footnote{Based on a one
       ↪ way flight for one passenger, data from \href{https://www.
       ↪ icao.int/environmental-protection/Carbonoffset/Pages/
       ↪ default.aspx}{icao.int}.}. \\
    \faIcon{mobile-alt} & \detokenize{@placeholder:"perspective.
       ↪ smartphones"} smartphones charged\footnote{Calculated using
       ↪  data from \href{https://www.epa.gov/energy/greenhouse-
       ↪ gases-equivalencies-calculator-calculations-and-references
       ↪ \#smartphones}{epa.gov}.}.
\end{tabular}
}
\end{multicols}


%%%%%%%%%%%%%%%%%%%%%%% breakdown section %%%%%%%%%%%%%%%%%%%%%%%
\begin{multicols}{2}
\section{Breakdown}
Carbon Intensity: \detokenize{@placeholder:"carbonIntensity"} g/kWh\
   ↪ footnote{Carbon Intensity obtained from \href{https://app.
   ↪ electricitymap.org/map}{electricitymap.org}.}\\
Total energy consumption: \detokenize{@placeholder:"energy.total.
   ↪ gross"} \detokenize{@placeholder:"energy.powerUnit.abbreviation
   ↪ "}\\
Energy consumption (emissions) per device type: \vspace{-6mm}\\
\vspace*{15.85mm}
\begin{figure}[H]
% this figure needs negative hspace, but the one from the next
   ↪ section does not
\hspace*{-3.5mm}
    \begin{tikzpicture}
    \small \color{white}
    \pie[sum=auto, text=legend, after number={~\detokenize{
       ↪ @placeholder:"footprint.massUnit.abbreviation"}}, radius=2,
       ↪  color={btpurple!100, btpurple2!100, btpurple3!100,
       ↪ btpurple4!100}]{
        @placeholder:"devices.servers.emissions"/{\color{black}\large
           ↪  Servers},
        @placeholder:"devices.cooling.emissions"/{\color{black}\large
           ↪  Cooling},
        @placeholder:"devices.network.emissions"/{\color{black}\large
           ↪  Network devices},
        @placeholder:"devices.misc.emissions"/{\color{black}\large
           ↪ Miscellaneous}
    }
    \end{tikzpicture}
    \label{fig:energy_per_category}
\end{figure}


%%%%%%%%%%%%%%%%%%%%%%%% offset section %%%%%%%%%%%%%%%%%%%%%%%%%%
\section{Offset methods}
There are various ways in which BT offsets these carbon emissions
   ↪ already, such as by using green energy, and by buying renewable
   ↪  energy certificates. \vspace{2mm}\\
Percentage of energy that is green: \detokenize{@placeholder:"
   ↪ greenEnergyPercentage"}\% \\
Renewable energy certificates (scaled to your energy consumption): \
   ↪ detokenize{@placeholder:"footprint.offsets.rec"} \detokenize{
   ↪ @placeholder:"footprint.massUnit.abbreviation"} \\
Energy consumption (emissions) after offset: \vspace{-6mm}\\
\begin{figure}[H]
    \begin{tikzpicture}
    \small \color{white}
    \pie[sum=auto, text=legend, after number={~\detokenize{
       ↪ @placeholder:"footprint.massUnit.abbreviation"}}, radius=2,
       ↪  color={btpurple!100, btpurple2!100, btpurple3!100,
       ↪ btpurple4!100}]{
        @placeholder:"footprint.total.net"/{\color{black}\large
           ↪ Remaining (Net)},
        @placeholder:"footprint.offsets.greenEnergy"/{\color{black}\
           ↪ large Green Energy},
        @placeholder:"footprint.offsets.rec"/{\color{black}\large REC
           ↪ }
    }
    \end{tikzpicture}
    \label{fig:offset_energy}
\end{figure}

\end{multicols}
```

```
%%%%%%%%%%%%%%%%%%%%%%%%% faq section %%%%%%%%%%%%%%%%%%%%%%%%%%
\vfill
\section{Frequently asked questions}
\titleformat{\subsection}{\normalfont\normalsize\itshape}{Q:}{2mm}{}

\subsection{Which scope are these emissions?}
\answer{According to the scopes defined in the Greenhouse Gas
   ↪ Protocol\footnotemark, the emissions from using BT's services
   ↪ will be classified as scope 3 for your business.} \footnotetext
   ↪ {Greenhouse Gas Protocol available at \href{https://ghgprotocol
   ↪ .org/corporate-standard}{ghpgrotocol.org}.}

\subsection{How are these figures obtained?}
\answer{The complete methodology is explained in detail in the thesis
   ↪  associated with this project.}

\subsection{Where can I find more in-depth information about my
   ↪ energy consumption?}
\answer{You can monitor your energy consumption in real time on the \
   ↪ href{censored}{QiO dashboard}.}


\vspace{5mm}
\end{document}
```