Software Ecodesign: Estimating and Reducing Software Environmental Footprint

Eco-Conception Logicielle: Estimation et Réduction de l'Empreinte Environnementale des Logiciels

***Thibault* SIMON**

École Doctorale MADIS
Centre Inria de l'Université de Lille
CRIStAL — Spirals research team

Thèse présentée et soutenue le 06/12/2024

pour obtenir le grade de docteur en informatique

Directeur et co-encadrant

Romain ROUVOY	Professeur	Université de Lille
Pierre RUST	Docteur	Orange Innovation

Rapporteurs

Anne-Laure LIGOZAT	Professoresse	ENSIIE
Vincent COURBOULAY	Maître de conférences	La Rochelle Université

Examinateurs

Jean-Christophe ROUTIER	Professeur	Université de Lille (président du jury)
Gaël GUENNEBAUD	Chargé de Recherche	Inria

Résumé

Les sociétés humaines sont devenues de plus en plus dépendantes des services numériques, qui sont désormais omniprésents dans presque tous les aspects de la vie quotidienne et des secteurs économiques. Cette augmentation du nombre de logiciels est étroitement liée à l'essor et au renouvellement rapide des appareils numériques, tant pour les consommateurs que pour l'infrastructure réseau et informatique. Cependant, cette tendance génère une empreinte environnementale qui apparait incompatible avec les limites planétaires. Les logiciels peuvent être un facteur important de cette empreinte environnementale croissante: les déploiements de logiciels influent la consommation d'énergie, et le besoin de logiciels de plus en plus sophistiqués nécessite une plus grande puissance de calcul, accélérant ainsi la fabrication et le renouvellement fréquent des appareils. Il est donc nécessaire d'identifier des leviers d'action holistiques pour réduire l'empreinte environnementale des logiciels. Cependant, les approches actuelles se concentrent souvent uniquement sur la phase d'utilisation, en prenant en compte uniquement la consommation d'énergie et l'empreinte carbone, négligeant ainsi d'autres aspects critiques de l'impact environnemental qui occurrent tout au long du cycle de vie du logiciel.

Dans cette thèse de doctorat, je combine différentes approches issues de divers domaines de recherche pour identifier des leviers significatifs pour réduire l'empreinte environnementale des logiciels. Dans un premier temps, j'évalue les avantages et les inconvénients des analyses *top-down* pour évaluer l'empreinte carbone du secteur des *Technologies de l'Information et de la Communication* (TIC), et je démontre leur intérêt pour évaluer l'impact du secteur sur d'autres catégories d'impact, notamment sur les métaux et les minéraux. En m'appuyant sur la tendance à la hausse observée, j'adopte ensuite une méthodologie *bottom-up* pour développer des outils et des méthodologies permettant d'évaluer et d'identifier des leviers de réduction dans divers aspects de l'empreinte environnementale des services numériques. Plus précisément, j'évalue l'empreinte environnementale des services cloud et des appareils utilisateurs avec une pensée cycle de vie, tout en proposant une nouvelle méthodologie pour suivre systématiquement les incertitudes découlant des sources de référence et des choix de modélisation au sein de ces estimations. Pour aller au-delà de la phase d'usage et de l'empreinte énergétique des logiciels, je propose également une méthodologie et un outil associé pour évaluer de manière holistique les impacts occurants tout au long du cycle de vie du logiciel.

L’estimation d’impacts n’est cependant que la première étape dans l’écoconception des logiciels. J’explore donc les différentes responsabilités des composants logiciels, et introduis un modèle conceptuel pour aider les différentes parties prenantes du logiciel à définir des métriques pour réduire l’empreinte environnementale des logiciels, dans leurs domaines de responsabilité. Dans ce cadre conceptuel, j’introduis une nouvelle métrique de qualité architecturale qui se concentre sur la minimisation du gaspillage de ressources induit par l’architecture du logiciel, en tant que solution simple et implémentable. De plus, je propose une approche pratique pour que les acteurs du logiciel s’efforcent d’atteindre une proportionnalité entre leur empreinte environnementale et l’évolution de l’usage au fil du temps.

Abstract

Human societies have become increasingly dependent on digital services, which now influence nearly every aspect of daily life and economic sectors. This increase in software services is closely tied to the rise and rapid renewal of digital devices, both for consumers and backbone infrastructure. However, this trend results in an environmental footprint that seems incompatible with planetary boundaries. Software can be a significant driver of this environmental footprint: software deployments steer energy consumption, and the need for increasingly sophisticated software requires greater computing power, thereby accelerating the manufacturing and rapid turnover of devices. As such, there is a need to identify holistic action levers to reduce the environmental footprint of software. However, current approaches often focus solely on the use phase, considering only energy consumption and carbon footprint, thereby overlooking other critical aspects of environmental impact that occur throughout the software's entire life cycle.

In this PhD thesis, I combine different approaches from various research fields to identify meaningful levers to reduce software environmental footprint. First, I assess the benefits and drawbacks of top-down analyses to assess the ICT sector's carbon footprint, and demonstrate their benefit to assess the sector's contribution to other impacts, specifically metals and minerals. Building on the observed upward trend, I then adopt a bottom-up methodology to develop tools and methodologies for assessing and identifying hotspots in various aspects of the environmental footprint of digital services. Specifically, I assess the environmental footprint of cloud services and user devices from a life cycle perspective, while proposing a novel methodology to systematically track uncertainties arising from reference sources and modeling choices within these estimations. To move beyond the usage phase and energy footprint of software, I also propose a methodology and associated tool to holistically assess impacts along the entire software life cycle. Impact assessment, however, is only the first step toward software ecodesign. I, therefore, examine the different liabilities of software components and introduce a conceptual model to help software stakeholders define actionable metrics for reducing the environmental footprint of software within their areas of responsibility. Within this conceptual framework, I introduce a new architectural quality metric that focuses on minimizing resource waste induced by the architecture of software, as a straightforward implementable solution. Additionally, I propose a practical approach for software stakeholders to strive towards proportionality between their environmental footprint and the evolution of usage over time.

Acknowledgements

Je souhaite tout d'abord sincèrement remercier mes encadrants, Pierre et Romain, pour m'avoir accompagné tout au long de ces trois années de thèse. La confiance et l'autonomie que vous m'avez accordées m'ont véritablement permis d'explorer différents sujets et de découvrir, ainsi que d'apprécier pleinement, le monde académique.

Je tiens également à remercier les membres du jury pour avoir accepté d'évaluer ce travail. Vos remarques et conseils précieux ont grandement contribué à en améliorer la qualité et la clarté, et je vous en suis profondément reconnaissant.

Un grand merci à chacun des membres de l'équipe SPIRALS que j'ai eu le plaisir de rencontrer. Bien que mes visites à Lille aient été ponctuelles, chacune d'entre elles a été l'occasion de découvrir de nouveaux sujets auprès de personnes passionnées et passionnantes (et, bien sûr, de partager quelques bières).

Remercier mes proches pour leur soutien durant ces trois années de doctorat serait presque hypocrite, tant j'ai apprécié cette expérience et su maintenir des horaires raisonnables pour passer du temps avec vous. Alors,

À mes parents, à mes frères et leurs conjointes, à mes neveux,
À mes amis de Lisieux, de Caen et d'ailleurs,
À ma famille, à ma belle-famille,
À celles et ceux qui ne sont plus là,
Merci à chacun et chacune d'entre vous de faire partie de ma vie et de me rendre heureux.

Valentine, quelques mots ici ne suffiraient pas pour exprimer à quel point tu remplis ma vie de bonheur et d'amour. Arthur, mon fils, merci d'avoir attendu le soir où j'ai terminé la rédaction de ce manuscrit pour venir dans ce monde et transformer le mien.

Table of contents

List of figures

List of tables

Acronyms

CaaS *Container-as-a-Service*. 73, 74, 77, 81, 83

IaaS *Infrastructure-as-a-Service*. 73, 74, 76

ADP *Abiotic resource Depletion Potential*. 7, 34, 36

CAGR *Compound Annual Growth Rate*. 4

CBA *Consumption-Based Accounting*. 26, 27, 31, 33

DQI *Data Quality Indicator*. 46–49

EEE *Electrical and Electronic Equipments*. 6

EEIOA *Environmentally Extended Input-Output Analysis*. 2, 10–12, 18, 24, 26, 28, 30–32, 34–36, 38, 85

FU *Functional Unit*. 8, 50, 79, 80

GHGs *Greenhouse gases*. 5, 11, 13, 14, 18, 29–32, 36, 45, 74, 83, 85

GWP *Global Warming Potential*. 29, 83

ICT *Information and Communication Technologies*. 1, 2, 4–21, 23, 24, 27, 29–37, 42, 45, 47, 49, 59, 68, 72, 85

IOA *Input-Output Analysis*. 10, 11, 25, 26

IPCC *Intergovernmental Panel on Climate Change*. 1, 29

ISIC *International Standard Industrial Classification of All Economic Activities*. 20, 21, 23, 24, 29–32

ISP *Internet Service Provider*. 8, 13

ITU *International Telecommunication Union*. 4, 5, 13, 18, 37, 46, 58

KPI *Key Performance Indicator*. 59, 86

LCA *Life Cycle Assessment*. 7–10, 12, 13, 18–20, 23, 24, 30, 31, 34, 36, 38, 39, 41, 42, 46, 47, 57–59, 67, 68

LCI *Life Cycle Inventory*. 9, 46

MRIO *Multi-Regional Input-Output*. 11, 12, 24

Mt *megaton*. 6

MW *megawatt*. 5

OECD *Organisation for Economic Co-operation and Development*. 18, 19, 21, 28

PBA *Production-Based Accounting*. 26–28

PUE *Power Usage Efficiency*. 20, 43–45

SBTi *Science Based Targets initiative*. 5

SDGs *Sustainable Development Goals*. 15

SDLC *Software Development Life Cycle*. 14, 37, 58, 59, 62, 72, 86

TIC *Technologies de l'Information et de la Communication*. iii

TrFN *Trapezoidal Fuzzy Numbers*. 48–50, 52–54, 58

TWh *terawatt-hours*. 10, 20

UN *United Nations*. 2, 11, 15

UNSD *United Nations Statistics Division*. 19, 20

vCPU *virtual CPU*. 43, 44, 61, 66, 68

vRAM *virtual RAM*. 43, 44

Chapter 1

Introduction

1.1 Motivation

1.1.1 Context

Six out of the nine planetary boundaries introduced by Rockstrom, defining the limits within which humanity can safely operate to avoid destabilizing the Earth's critical environmental systems, have been exceeded [5]. Regarding the climate change boundary, Freitag *et al.* [1] estimate, based on previous studies [6–8], that in 2020 between 2.1% and 3.9% of worldwide greenhouse gas emissions could be attributed to the digital sector.

The latest *Intergovernmental Panel on Climate Change* (IPCC) report [9] states that emissions of greenhouse gas from human activities are responsible for approximately 1.1°C of global warming since 1850–1900 and that—unless there is an immediate, rapid, and large-scale reduction—limiting it to $+1.5$°C will be out of reach. To make its fair contribution to achieving this objective, the ICT sector is challenged to adopt a more sustainable approach and reduce its environmental footprint. Moreover, the environmental impact of ICT extends beyond its contribution to climate change. It also plays a role in the transgression of other planetary boundaries, primarily through the production and operation of both consumer and infrastructure devices.

1.1.2 Problem statement

Faced with this challenge, the software industry is challenged to reduce its environmental footprint. Although often perceived as immaterial, software is heavily reliant on ICT infrastructure, driving its energy consumption and requiring ever-increasing computing power, thus accelerating the manufacturing and rapid turnover of devices.

Therefore, it is essential to rigorously assess the environmental footprint of software in order to identify and implement meaningful action levers to mitigate its impact. However, existing approaches primarily focus on energy consumption, addressing only a fraction of the environmental strain caused by software. These methods frequently overlook the life cycle costs of the hardware involved and only partially consider environmental impacts beyond climate change. By centering their analysis on energy, they also limit their scope to the usage phase of software, neglecting the broader software life cycle, including the development phase.

Impact estimation is the first step in reducing the environmental footprint of software, yet defining actionable metrics from a technical perspective remains a significant challenge. The modular nature of software architecture complicates the assessment of each component's individual contribution to the overall reduction potential. Since not all components possess the same potential to reduce the overall impact, it is essential to focus efforts on those with the highest reduction potential.

In this thesis, we aim to address the following research question: *Which actionable strategies can be identified and implemented to holistically reduce the environmental impact of software?*

1.2 Contributions

This thesis is structured into four main parts:

Background: Chapter 2 explores the relationship between the growth of the ICT sector and its environmental impact, with a particular emphasis on its material footprint. We then examine current impact assessment methodologies employed at both macro and micro levels, evaluating their potential as well as their limitations. Finally, we explore the notion of software environmental footprint, examining existing research on the intersection of software and environmental sustainability goals.

ICT sector's footprint: In Chapter 3, we compare the boundaries used in ICT environmental footprint studies to the *United Nations* (UN)'s economic definition of the sector. Subsequently, we demonstrate how the use of *Environmentally Extended Input-Output Analysis* (EEIOA) can produce estimates that align with this economic definition. Applying this approach, we assess the evolution of the ICT sector's carbon footprint and identify its primary drivers. Additionally, we conduct an analysis of its material footprint, while also highlighting the limitations of this approach.

Software's footprint: Zooming in from the environmental impact of the whole digital sector, Chapter 4 focuses on the footprint of software-based systems. We propose advancements in the methods and tools for assessing the environmental footprint of software: we introduce a new bottom-up approach and accompanying tool for assessing the environmental footprint of cloud instances, along with a method to systematically track uncertainties arising from reference sources and modeling choices. This approach is demonstrated through the modeling of end-user devices. Lastly, we adopt a broader perspective on the environmental footprint of software, beyond hardware resources, and develop a tool and methodology to assess its environmental impact across the entire software lifecycle.

Reducing: In Chapter 5, we introduce a conceptual model that considers the responsibility of various software components and stakeholders in reducing the environmental footprint. Building on this foundation and with a focus on architecture, we propose a new quality metric designed to reduce resource waste. This metric offers a straightforward, implementable solution to reduce the environmental footprint of software components by treating them as distinct entities, rather than as a monolithic system. Finally, we propose a practical approach for software stakeholders to achieve proportionality between their environmental footprint and the evolution of the functional unit over time.

In addition to addressing the research goal, all tools and data generated throughout this dissertation have been made publicly available as open-source or open-data in repository: https://github.com/tibosmn/phd-thesis. This encourages the reproducibility of our results and facilitates future research in the field.

1.3 List of Scientific Publications

Parts of this thesis are adapted from the following publications:

1. T. Simon, D. Ekchajzer, A. Berthelot, E. Fourboul, S. Rince, and R. Rouvoy, "BoaviztAPI: a bottom-up model to assess the environmental impacts of cloud services," in *HotCarbon'24. Workshop on Sustainable Computer Systems*, (Santa Cruz, United States), July 2024

2. É. Guégain, T. Simon, A. Rahier, and R. Rouvoy, "Managing Uncertainties in ICT Services Life Cycle Assessment using Fuzzy Logic," in *International Conference on ICT for Sustainability (ICT4S)*, (Stockhlom, Sweden), IEEE, June 2024

3. T. Simon, P. Rust, R. Rouvoy, and J. Penhoat, "Uncovering the environmental impact of software life cycle," in *International Conference on ICT for Sustainability (ICT4S)*, pp. 176–187, 2023

An additional contribution is under submission:

- S. Cerf, A. Luxey-Bitri, C. Quinton, R. Rouvoy, T. Simon, and C. Truffert, "Untangling the Critical Minerals Knot: when ICT hits the Energy Transitions." working paper or preprint, Dec. 2023

1.4 Other contributions

This thesis has resulted in the creation of a tool and multiple research prototypes. The complete source code for all artifacts has been made available at https://github.com/tibosmn/phd-thesis.

Chapter 2

Background

2.1 Digital Sector Growth and Environmental Impact

2.1.1 The Staggering Growth of the ICT Sector

Societies have become increasingly dependent on digital services, which are now in nearly every aspect of daily life and economic activity. The ongoing digital transformation of modern societies has led to a proliferation of devices, the expansion of network boundaries and capacity, as well as the construction of additional data centers to support the delivery of these services.

According to Cisco's *Annual Internet Report* covering 2018 to 2023 [14], the total number of internet-connected devices increased from 3.9 billions in 2018 to 5.3 in 2023. North America, followed by Western Europe, exhibited the highest adoption rate during this period. Notably, IoT devices showed the most rapid growth, with the number of devices per capita increasing from 2.4 to 3.6 over the same period. Accordingly, per the *International Telecommunication Union* (ITU) [15], 67% of the global population is now connected to the internet, thus leaving 33% offline. This marks a significant increase in internet connectivity, rising from 1 billion people in 2005 to 5.4 billion in 2023. However, this inequality of access results in a *digital divide*, defined as the gap between those who have access to ICT technologies and those who do not [16]. The digital divide presents a substantial threat to the potential benefits of universal ICT access, which has the capacity to foster a global community characterized by improved living standards and enhanced social welfare.

To accommodate with this rapid growth in connectivity and usage, the backbone infrastructure has expanded tremendously. In France, Ahmed and Coupechoux [17] estimate that the total operational power consumption of cellular base stations has grown at a *Compound Annual Growth Rate* (CAGR) of 18.18%, a trend further accelerated by the introduction of 5G technology. According to CISCO, the global cloud data centers traffic has been growing at a 25% CAGR between 2016 and 2021 [18]. As of 2023, Pilz and Heim [19] estimate that around $10,000$ and $30,000$ datacenters exist globally, with approximately 140 facilities having a capacity over 100 *megawatt* (MW). For comparison, French Nuclear power plants generate between 900 MW and 1300 MW [20].

As data centers continue to experience substantial growth [21], their associated environmental impacts are likely to increase, further hindering the path toward the sector's sustainability. According to the Central Statistics Office of Ireland [22], data centers accounted for 5% of all metered energy consumption in 2015, a figure that rose to 21% by 2022. In comparison, urban and rural households contributed respectively to 19% and 10% to metered energy consumption.

2.1.2 Carbon Footprint Trends

The hosting infrastructure is estimated to contribute between 18% and 45% to the ICT sector's global carbon footprint [1, 7, 23, 24].

According to Freitag *et al.* [1], based on previous studies [6–8], the sector was responsible for between 2.1% and 3.9% of greenhouse gas emissions in 2020. While estimating and collecting accurate data within the ICT sector is challenging due to its vast size, complexity, and variability [25–27], the authors also argue that the sector's carbon footprint continues to grow when it should be decreasing. Indeed, authors estimate based on the projections of the ITU [28] that to stay within the 1.5°C global warming limit, the sector's emissions must be reduced by 42% by 2030, 72% by 2040, and 91% by 2050, ultimately achieving net-zero emissions by 2050 to align with the climate change planetary boundary. The *Science Based Targets initiative* (SBTi) has set targets for the ICT sector to reduce greenhouse gas emissions by 45% by 2030 relative to 2015 levels and by 90% by 2050 to align with the climate change planetary boundary.

However, despite their public goals of reaching carbon neutrality, both Microsoft and Amazon reported increases in their carbon footprints in their 2024 *Environmental Sustainability Report*s . Microsoft increased by 29.1% compared to 2020 [29], with scope 3 emissions—primarily driven by supplier purchases and client usage—accounting for 96.3% of their total carbon footprint. Similarly, Google reported a 48% increase in GHGs emissions compared to 2019 [30], which was *"primarily due to increases in data center energy consumption and supply chain emissions"*.

2.1.3 Sector's Materiality

While studies frequently focus on the environmental footprint associated with the energy consumption of ICT, they often overlook the impact of hardware production, which can be significantly greater [31]. For instance, Gupta *et al.* [32] showed that reducing a data center's energy consumption alone fails to reduce its carbon emissions, as hardware manufacturing can play a bigger role in its environmental impact. Moreover, the environmental impacts of ICT extend beyond climate change, as the sector also contributes to the transgression of other planetary boundaries, as evidenced in Europe in [33].

Furthermore, achieving net-zero GHGs emissions by 2050 [34] will require a significant shift away from fossil fuels, with the majority of energy sources transitioning to electricity. Until 2030, this implies a massive ramp-up of *critical mineral immobilization* to shift our energy infrastructure, including renewable production facilities (e.g. solar farms and wind turbines), stationary storage, and non-carbonated transportation. A mineral is considered *critical* when it serves an essential role for energy production, distribution and storage, and when geological, political, or technical reasons may hinder its supply chain. As a matter of fact, the short-term availability of critical minerals may jeopardize the necessary manufacture of electricity plants, storage infrastructures, and electrical vehicles [35].

ICT has an ever-growing demand for minerals [36], both in quantity and diversity, contributing to critical mineral immobilization for its extensive materiality through the spread of data centers, network infrastructures, and devices. Many studies on critical minerals supply chains do not take into account the impact of ICT's constant growth, which has remained a blind spot in the forward-looking transition studies [37]. Yet, the manufacturing of our surrounding digital devices (workstations, screens, tablets, smartphones, desktops, laptops, and even data centers) requires from critical minerals [38]. Emsbo *et al.* even claim that manufacturing a smartphone requires minerals covering up to two-thirds of the periodic table [37].

In a world where natural resources are finite, recycling is often proposed as a solution, particularly in light of the increasing generation of e-waste. The *Global E-waste Monitor* [39] estimates that the cumulative world generation of e-waste rose from 44.4 *megaton* (Mt) in 2014 to 62 Mt in 2022, and is expected to grow to 82 Mt by 2030. This rise is attributed to higher consumption rates of *Electrical and Electronic Equipments* (EEE), shorter product life cycles, and limited repair options. The improper disposal of e-waste leads to significant global disparities and notably causes irreversible pollution to soil [40], and seafood [41]. Unfortunately, recycling activities have been unable to curb, or even match, this growth: only 13.8 Mt were officially documented as properly collected and recycled in 2022. In the European Union, it is estimated that less than 40% of e-waste is recycled [42]. This figure, however, only reflects the proportion of collected electronic equipment and does not account for the actual recovery of materials from these devices.

Bashroush *et al.* [43] demonstrate through two recycling scenarios that the most critical server components cannot be fully recovered, and that a significant portion cannot be recycled at all. Unfortunately, current state-of-the-art recycling technologies are unable to produce the high-purity substances required to manufacture ICT components. The ICT sector uses alloys containing highly diluted minerals, notably for miniaturization and short-lived devices, hence limiting their recycling potential to around 1 % [38, 44]. Moreover, for recycling to take place, the product or infrastructure must have reached the end of its lifecycle and be properly collected, which necessarily takes time, while the ongoing energy transitions require the rapid deployment of an important quantity of *new* infrastructure. For example, copper is currently immobilized for 45 years, on average [45], which seriously limits its availability for recycling. Therefor, for the next two decades our society massively depends on resources of *primary* origin [38]. This implies that if the risks associated with critical mineral depletion are not addressed quickly, global progress towards net-zero will be significantly hampered [46].

For several decades, numerous studies [37, 47, 38, 45] have examined the interdependence of the energy transitions to mineral resources of primary and secondary origins—obtained from mining and recycling [44], respectively. In this context, given the consensual objective of limiting GHG emissions [48] and the acknowledged limited availability of mineral resources, researchers and policy makers need data to support the energy transition and possible upcoming race for minerals. Unfortunately, research communities are still missing adequate analysis tools and metrics to properly evaluate and design effective policies to mitigate the mineral footprint of ICT. Suitable data is not yet available, as existing studies mainly rely on the combination of bottom-up *Life Cycle Assessment* (LCA) data and top-down market data to assess the ICT sector's environmental footprint [1]. When mineral depletion is considered, it is thus typically measured through the *Abiotic resource Depletion Potential* (ADP) impact category, which aggregates various minerals into a single category, hindering detailed studies of individual minerals (e.g. their criticality, grades, recycling, etc.). Furthermore, this indicator has been criticized for assuming that ultimate reserves are a good proxy for extractable reserves and for failing to account for resource quality and accessibility after extraction [49].

In Chapter 3, we introduce a novel approach for evaluating the evolution of the ICT sector's environmental footprint across multiple impact categories, and notably conduct an analysis on specific minerals.

2.2 Environmental Impact Assessment Methodologies

2.2.1 Bottom-Up vs. Top-Down Approaches

Different methodologies and standards are defined and often used in combination to assess the environmental footprint of the ICT sector, as well as for specific products and services. Such approaches for calculating environmental impacts can be categorized as either *top-down* or *bottom-up*. A top-down approach starts by examining the entire system, using aggregate statistics and input-output relationships to model its overall behavior. From this broad perspective, the system is subsequently decomposed into smaller components, allowing analysis and insights at the sub-system or element level. For example, Chan *et al.* [50] employ this approach to model the energy intensity of the Internet. By dividing the total estimated energy consumption by the traffic over a given period, they derived an intensity metric expressed in *kWh per gigabyte of data transferred*. This metric can then be used at a smaller scale to estimate the energy consumption of individual internet exchanges. Unfortunately, top-down approaches generally suffer from an averaging bias, a distortion that occurs when the average of a dataset misrepresents the underlying data or masks important variations, which limits their usefulness in identifying specific action levers. For instance, in the case of modeling the Internet's energy intensity, such an approach only reveals two broad reduction levers: either reducing the amount of data transferred or lowering the overall energy footprint of the ICT sector.

In contrast, a bottom-up approach is based on direct observations from specific case studies [51], which are then generalized to represent larger-scale systems. Schien *et al.* [52] use this approach to model the energy intensity of the Internet, by scaling the energy consumption of devices within a simulated *Internet Service Provider* (ISP) network. This approach directly captures changes made to the system directly, without relying on structural changes, as seen in a top-down approach. As such, it allows for the identification of the specific contribution of each resource to the overall environmental impact. However, this approach may be less representative due to the risk of overgeneralization and the potential to mask significant differences arising from variability between the assessed and the installed systems.

Most studies on the environmental footprint of the ICT sector combine both top-down and bottom-up approaches to obtain more accurate estimations [1]. For example, Malmodin and Lundén [8] used a bottom-up approach to assess the environmental impact of user devices, while employing a top-down approach for data centers and networks.

2.2.2 Life Cycle Assessment

A well-established bottom-up approach is *Life Cycle Assessment* (LCA), as defined by the ISO 14040 [53]. An LCA is the study of the potential environmental impacts contribution of a product or service across its entire life cycle—*i.e.*, from raw material acquisition to waste management via production and use phases. It is both employed in both the industrial and academic context [54, 55]. The analysis is performed for a specific *Functional Unit* (FU), a quantitative measure of the functions provided by the product or service [56], allowing to compare two systems with the same FU from an environmental perspective. By using a systematic overview and perspective, the LCA approach helps to identify the shifting of a potential environmental burden between life cycle stages or individual processes [57]. A *screening LCA* is simpler to conduct and is established based on readily available data [58], and thus provides a high-level view of impacts to identify the main sources and those that require deeper examination.

Ekvall [59] defines two types of LCA: *attributional*, which aims at describing the environmental properties of a life cycle and its subsystems, and *consequential* which aims at describing the effect of changes within the life cycle. Whereas a consequential LCA aims at describing the long-term effect of changes induced by the product or service and thus support decisions in the long term, an attributional one maps the environmental impacts that a product can be made accountable for and is thus better suited to support decisions that aim at improving its life cycle processes [60].

As described in Figure 2.1, the term *Cradle to Gate* refers to a partial LCA that covers the environmental impact of a product from the extraction of raw materials (the cradle) to the point where the product leaves the manufacturing process and is ready for shipping to the consumer (the gate). This approach focuses on the production and manufacturing stages and does not include the potential impacts of the product's use or disposal. On the other hand *Cradle to Grave* is a comprehensive LCA approach that includes all stages of a product's life cycle. This method extends beyond the

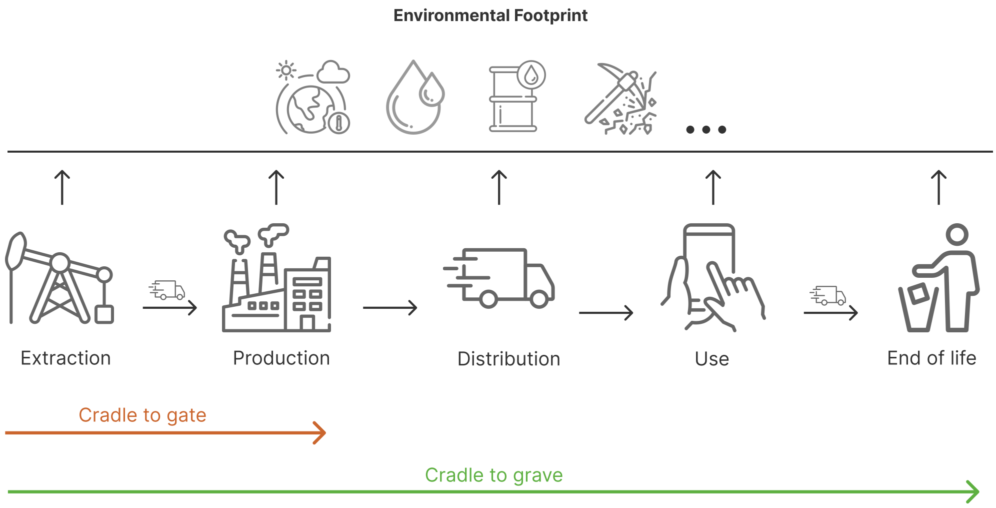

Fig. 2.1 Comparison of Cradle-to-Gate and Cradle-to-Grave Boundaries

manufacturing stage to cover the use, disposal, and potential recycling of the product, effectively capturing the potential environmental impacts from raw material extraction through to the end of the product's life [56].

For ICT products and services, comprehensive analyses from *Cradle to Grave* remain publicly scarce, and the quality of the resulting data varies significantly. When openly available, their scope and system boundaries are not always explicitly stated or consistent with other studies [1], and the uncertainty of the results is rarely quantified [61]. In the case of modular products, such as servers, hardware configurations can have a substantial impact on the asset's environmental impact. In such a case, LCA can suffer from an *empty-shell bias*, where only unrealistically low-end configuration modelings are published, potentially underestimating the true environmental impact.

Lack of reference data and uncertainties

Truncation errors can occur in LCA due to the inability to fully analyze the entire complexity of production chains. As a result, certain upstream chains have to be cut-off, leading to potential inaccuracies and underestimations [62]. These omissions are not the only source of uncertainty within LCAs, as the ecosystem often lacks sufficient data on the environmental impact of the resources it consumes [63], resulting in reference data associated with high uncertainties [64]. Additionally, most *Life Cycle Inventory* (LCI) databases are closed-source, which hinders the goal of conducting transparent and reproducible research [65].

Arushanyan *et al.* [66] notably emphasize that rapid technological development is a significant source of variability in LCA results, impacting the entire life cycle of ICT products and services. They also identify another source of variability arising from the assumptions and hypotheses made during the modeling process. Furthermore, the authors highlight that one of the key challenges in

ICT-related LCAs is the thorough documentation of these assumptions and modeling hypotheses, which is essential for ensuring transparency and consistency in the assessment.

The lack of openly available data, the complexity of ICT systems and their variability make it difficult to assess their impact. For instance, while Andrae *et al.* [67] estimate the worldwide data centers consumption to be around 299 *terawatt-hours* (TWh) in 2020, China's State Grid Energy Research Institute released a report stating that Chinese data centers consumption alone was around 200 TWh the same year [68].

While ICT-related LCA is associated with high levels of uncertainties, the uncertainty of results is rarely quantified [61, 69]. To tackle these limitations, Hischier *et al.* [70] proposes a systematic sensitivity analysis as a solution, as assumptions made at the data inventory level significantly influence the outcomes. Unfortunately, such extensive analyses are time-consuming due to the large data flows to handle.

The handling of uncertainties in LCAs is not limited to the ICT sector [71]. Indeed, the result interpretation phase in LCA is particularly critical and can become subjective and time-consuming. One common approach to assess uncertainties in LCA is the Monte Carlo method [72]. However, it requires a high number of simulations, resulting in a high calculation time [73], and is based on scenarios arbitrarily defined in the sensitivity analysis step. To improve the clarity of interpretation, fuzzy sets have been proposed and adopted as means to quantify and propagate imprecision and uncertainties within various LCA steps [74–76]. However, despite being promising, fuzzy logic is not yet implemented in LCA software [73]. To the best of our knowledge, such methodology has not yet been used within ICT-related LCAs, a field involving particularly high uncertainties.

Limitations

While an LCA reveals direct environmental effects, it does not capture the broader role of ICT as an enabling technology [77]. Notable limitations of LCA include its inability to account for higher order effects such as rebound effects [78]. Additionally, end-of-life impacts are often excluded in ICT-related LCAs due to the scarcity of data on e-waste collection and recycling rates [33, 79].

In Section 4.2, we present a methodology designed to systematically evaluate uncertainties originating from reference data sources, propagate these uncertainties throughout the LCA process, and pinpoint their contributions to the overall uncertainty in the results.

2.2.3 Environmentally Extended Input-Output Analysis

Environmentally Extended Input-Output Analysis (EEIOA) is a top-down environmental impact assessment approach based on *Input-Output Analysis* (IOA), a subfield of economic analysis that models economic systems as networks of exchanges of goods and services between defined economic sectors. This approach was first introduced by Leontief [80], who examined how changes within one economic sector can influence and impact other sectors within the economy. IOA employs an economy-wide top-down perspective, representing the interlinkages between different branches of a national economy or various regional economies. *Multi-Regional Input-Output* (MRIO) models extend this approach to cover the entire global economy, in a format consistent with the recommended accounting systems proposed by the UN, allowing for international comparisons across countries and regions. The data for these models are expressed in monetary flows and are typically reported annually using national statistics.

In IOA, the economy is characterized by a network of input-output relationships structured in a matrix format, where each sector both consumes inputs from and supplies outputs to other sectors and to final demand. This structure allows IOA to assess the broader economic repercussions of changes in one sector, by tracking the interdependencies among various producing and consuming sectors within an economy. Specifically, IOA measures the relationship between a given set of demands for final goods and services and the inputs required to meet those demands. By integrating data on production inputs, such as the resources required, EEIOA enables the calculation of indicators related to output intensities [81].

By analyzing the embedded environmental footprint within monetary flows, for instance using metrics like $kgCO_2eq$ / \$, EEIOA enables the assessment of a sector's environmental footprint across multiple categories, such as GHGs, water consumption, and land use. These factors are derived by combining, normalizing, and adjusting environmental data with economic data to calculate relevant coefficients [82]. This method can help identify emerging trends and critical areas of concern by determining the total upstream resource requirements and associated environmental impacts necessary to meet the final demand of a country or an industry [62]. By tracing inter-business and inter-country monetary flows back to final consumption, this approach enables the assessment of upstream environmental impacts. It does so by utilizing translation factors to account for various environmental consequences, providing a comprehensive analysis of the broader environmental footprint associated with economic consumption. EEIOA has notably been employed to evaluate the upstream footprint for nations [83], households [84], and final goods [85]. The main advantage of input-output analysis is that it allows calculating material footprints for all products or industries, also those with very complex global supply chains, as the whole economic system is included in the calculation system [86, 87]

The most common philosophy in the EEIOA literature is *consumer responsibility* [88], where the ultimate responsibility for environmental impacts is assigned to the end consumer who purchases a good or service [89]. The hypothesis assumed is that the consumer, being at the very end of the value chain, bears the accumulated environmental impacts incurred during production. The key assumption in EEIOA is that all material use and environmental impacts are driven by final demand and that all material use can be attributed to elements of final demand, following a consistent accounting logic. This approach ensures that double counting is avoided, as the production emissions across the full upstream supply chain are allocated only once to final consumption [90].

Using EEIOA with China as a case study, Zhou *et al.* [91] observed a rapid growth in embodied emissions within the ICT sector, attributing this primarily to the expanding final demand for ICT products. San Miguel *et al.* [92] used MRIO to evaluate the environmental and socio-economic sustainability of ICT, particularly in telecommunication networks. They used EEIOA to assess the carbon footprint and employment generation for internet services in six geo-demographic zones, providing a complementary approach to traditional process-based LCA. The study found that most ICT network carbon emissions and employment were associated with the manufacturing of end-user devices in China. The annual carbon footprint of the ICT network in Lima was about three times higher than estimates from conventional LCA, highlighting the difficulties in comparing results from different methodologies due to variations in system boundaries, disaggregation levels, and database precision.

Lutter *et al.* [62] identified two key disadvantages of using EEIOA. The first major drawback is that EEIOA relies on aggregated economic sectors and product groups, operating under the assumption that each sector produces a homogeneous product. This means that when a single sector produces different products with potentially very different material intensities or carbon footprints, these are averaged together, potentially obscuring significant variations. The second limitation lies in the assumption of proportionality between monetary and physical flows. Price disparities between industries can occur, particularly when aggregating various types of materials [93]. This assumption may result in inaccuracies when evaluating environmental impacts, as monetary values may not precisely reflect the actual physical resource consumption or the associated environmental burden.

In Section 3.2, we use EEIOA to assess the carbon footprint of the ICT sector and correlate the results with reference studies

2.3 Software Environmental Footprint

While the environmental footprint of ICT primarily arises from hardware, it can be argued that software nonetheless holds a share of responsibility. Indeed, the energy consumption of ICT is steered by software deployments. The rise of device manufacturing is driven by the development and deployment of increasingly sophisticated software, that is ever more demanding in computing power. Furthermore, e-waste generation is partly driven by software-induced obsolescence: new software generally requires more computing resources, driving the need for renewing existing equipment and in some cases features may even be arbitrarily restricted or updates discontinued on functional hardware.

In 1997, Nathan Myhrvold, the Chief Technology Officer of Microsoft, defined his four laws of software [94] predicting that the size and complexity of software would continue to increase indefinitely, as follows:

- Software is a gas: it expands to fit the container it is in
- Software grows until it becomes limited by Moore's Law
- Software growth makes Moore's Law possible: people buy new hardware because software requires it

- Software is only limited by human ambition and expectation: we'll always find new applications and new users

Nearly thirty years later, these laws remain largely valid, as evidenced by the continually expanding software services industry [95]. However, they can be seen as incompatible with the realities of a finite world, underscoring the responsibility of the software industry to shift this paradigm in order to operate within planetary boundaries.

Of course, one can also rightfully argue that software holds significant potential to enhance understanding of the climate crisis, particularly through the creation and application of climate models [96], as well as supporting mitigation and adaptation strategies [97]. Nonetheless, it is essential that the development and deployment of software also align with sustainability goals. Current efforts to improve software sustainability, however, predominantly concentrate on reducing its energy consumption through performance optimizations [98]. These include strategies such as energy-efficient virtual machine placement in data centers [99], energy monitoring throughout software release cycles [100], and addressing energy-related code smells [101].

However, as previously discussed in Subsection 2.1.2, the energy consumed during the usage phase of a service or product may not always be the primary driver of its environmental impact. Therefore, adopting a more holistic approach that considers the entire life cycle and broader environmental impact categories is essential.

2.3.1 Software Environmental Footprint Estimation Approaches

Several standards have been proposed to estimate the environmental footprint of digital services. While not strictly focusing on software, these approaches target software-based systems and includes software's footprint by taking into account the devices and equipment required to run it.

The ITU L.1410 [102] standard complements the ISO 14040 [53] and 14044 [56] and proposes a *Methodology for environmental life cycle assessments of information and communication technology goods, networks and services*. It defines a set of requirements that LCA practitioners should strive for, as compliance may not always be feasible. In the specific context of ICT services, a three-tier architecture can be considered—encompassing end-user devices, networks, and data centers.

It was refined in France by the *General Principles for the Environmental Labelling of Consumer Products, Methodological Standard for the Environmental Assessment of Digital Service* [103], which aims to provide a set of common rules to assess and inform consumers about the environmental impact of digital services. This includes methodologies to assess the footprint of ISP and hosting services in datacenters. The *Greenhouse Gas Protocol Product Life Cycle Accounting and Reporting Standard (Product Standard)* [104] gives requirements and guidance for organizations to quantify and report an inventory of GHGs emissions and removals associated with a specific product. The GHG protocol *ICT's Sector Guidance* [105] define the assessment of software life cycle GHGs impact with predefined phases (material acquisition and preprocessing, production, distribution and storage, use, and end of life), and consider the software development. However, unlike the LCA approaches, it uses a single category of impact and only accounts for GHGs emissions. Besides, upstream emissions such as buildings, are not taken into account.

Industry working groups dedicated to addressing the environmental footprint of ICT have also suggested alternative methodologies for estimating the footprint of software-based systems. The Green Software Foundation defines the *Software Carbon Intensity* (SCI) [106], methodology to compute the rate of carbon emissions per functional unit for a software system, which was a standardized in 2024 as ISO/IEC 21031 [107]. It takes into account both energy consumption and reserved hardware embodied emissions, but only covers GHGs emissions and the usage phase of a project, not its development. The *Sustainable Digital Infrastructure Alliance* (SDIA) proposes a methodology to compute the carbon emissions of server-side applications [108]. This approach encompasses both energy consumption and embodied carbon emissions, but only accounts for backend impacts and excludes those of the network and end-user devices.

2.3.2 Software Development Life Cycle

The environmental footprint of a software development may be impacted by several factors, spanning human factors to infrastructure choices. When developing software, developers typically follow documentation and guidelines aimed at optimizing maintainability, performance, and usability. However, there is a notable lack of guidance and knowledge on how to reduce the environmental impact of their work [109], as well as the absence of comprehensive sustainability models [110–114]. Despite this, developers are generally willing to adopt sustainability practices if such models and guidelines are made available and can be easily integrated into their workflows [115].

To define a *Software Development Life Cycle* (SDLC), the international standard ISO 12207 [116], entitled *Systems and software engineering - Software life cycle processes*, defines a set of processes organized as requirements, design, implementation, testing, deployment, maintenance and retirement [117]. In projects, these life cycle processes can further be organized in different models, such as Waterfall, V-Model, Scrum… [118]

Trends in green computing studies published since 2012 indicate a clear and growing interest in the field [119]. Several approaches have been proposed [112, 120, 121, 111, 113, 117] that aim to leverage software life cycle models to minimize the environmental impact of software development and operation. Some authors suggest utilizing the software life cycle to reduce the environmental impacts of the different processes. For instance, Kumar *et al.* [120] propose the Green Star model, which assesses and rates each process of the SDLC based on its environmental impact. Lami *et al.* [121] look at the sustainability factors of software processes and propose specific reduction objectives, including carbon footprint, energy consumption, waste generation, and travel. Additionally, they introduce new processes to ensure that these sustainability objectives are met throughout the software life cycle.

SDLC processes can also be used to reduce the environmental impact of the resulting software itself. Shenoy *et al.* [117] propose specific changes to the SDLC and offer guidelines for each process to achieve this goal. Similarly, Dick and Naumann [112] introduce processes specifically aimed at ensuring the product sustainability, such as sustainability reviews and sustainability retrospectives. The GREENSOFT model and its associated metrics [113, 111] provide a comprehensive conceptual framework that addresses both process and product sustainability, emphasizing the importance of considering software products throughout their entire life cycle.

To the best of our knowledge, only two papers tried to assess the carbon footprint of software products' life cycle. The first attempt was conducted by Taina *et al.* [122], where the authors proposed different approaches to assess the carbon footprint of phases on an artificial software project. By giving estimates for each phase, they emphasize that the way software is developed and delivered matters, as well as how usable it is. They mostly focus on energy consumption and paper printing, but conclude that traveling would dominate the result, if included. Kern *et al.* [123] focused on whether employee commuting should be included in software carbon footprint assessment, and conduct two case studies on micro-enterprises in Germany. They compute factors of impact for this context to give an impression of the magnitudes, and show that the distribution of environmental impacts between the build and the run phases can vary greatly depending on the number of copies sold. Both of these approaches do not cover the complete life cycle of the resources used, especially the impact of ICT equipment manufacturing, but only account for their energy consumption.

In Section 4.3, we develop a methodology and associated tool to assess the environmental impact of software from a life cycle perspective, extending the analysis beyond its usage phase.

2.3.3 Sustainable software

Despite being commonly perceived as immaterial, software is continuously expanding its strain on digital infrastructures and devices, whose rapid renewal and growth impede the progress of our society toward achieving *Sustainable Development Goals* (SDGs). *Sustainable development* was defined in the Bruntland report [124] as the *"development that meets the needs of the present without compromising the ability of future generations to meet their own needs"*, and its three dimensions as *social, economic and environmental*. Lago *et al.* [125] introduce a fourth dimension of sustainability for software: the *technical* pillar, as the long-term use of software-intensive systems and their appropriate evolution in a constantly changing execution environment. Hilty *et al.* [126] describe the *sustainable use* of a system as not jeopardizing its ability to fulfill a function for a given period.

Koziolek [127] considers long-living software as sustainable if it can be *cost-efficiently maintained and evolved over its entire life cycle*. Penzenstadler *et al.* [128] present two interpretations of the term *sustainable software*: its purpose being to support sustainability goals, or its code being sustainable, agnostic of its purpose.

The UN proposed 17 SDGs to address poverty, inequality, climate change, environmental degradation, peace, and justice [129]. Calero *et al.* [130] studied their integration with the two interpretations of sustainable software, and concluded that both approaches are complementary and necessary to support them. Examples of software projects whose purpose is to support sustainability goals can be found in Open Sustainable Technology [131] curated list of open-source projects that *"sustain a stable climate, energy supply, biodiversity and natural resources."*

For the second interpretation of software sustainability,—*i.e.*,, focusing only on technical aspects while being agnostic of its purpose—Calero *et al.* [132] defined two categories of indicators: *resource-oriented* ones that cover the sustainable use of the planet, and *well-being* ones that measure the fulfillment of human needs. They stress the importance of combining both to quantify sustainable development. Hilty's framework [126] classifies the positive and negative effects of ICT on the environment among three orders of effect. The first-order, or *direct effect*, encompasses the negative environmental impacts of ICT hardware life cycle. The second-order, or *effect of use*, covers optimization and substitution effects, as well as obsolescence and induction effects. Finally, the third-order or *systemic effects* considers the long-term reaction of the dynamic socio-economic system to the availability of ICT services, notably rebound effects. For Calero *et al.* [133], the second and third orders of effect depend on factors out of developer's control at development time.

However, suitable and actionable metrics to foster sustainability from a software architect's point of view are still lacking. Penzenstadler *et al.* [134] argue that software environmental sustainability must be explicitly considered as a nonfunctional requirement in the software engineering process.

Following the ISO 25010 [135], Jagroep *et al.* [136] frame sustainability as a quality attribute, and focus on resource consumption sub-characteristics. They argue that software architecture can help in considering software as interrelated components rather than a single complex entity, to find the drivers of its energy consumption. While advancing five quality metrics they focus on energy consumption as a sustainability driver, thus overlooking hardware embodied environmental footprint. Lago *et al.* [137] introduced the Sustainability Assessment Framework, which allows to assess quality attributes and identify interplay between these attributes with respect to their impact on the different pillars of sustainability.

Taking the viewpoint of cloud infrastructure consumers instead of providers, Vos *et al.* [138] suggest 18 architectural optimization tactics, yet they also consider exclusively energy efficiency. To evaluate the sustainability of cloud software architectures, Fatima *et al.* [139] notably identified the following quality attributes: resource utilization, scalability, and elasticity. Taina *et al.* [140] has proposed multiple green software quality factors, including *feasibility* (how resource efficient it is to develop, maintain, and discontinue software) and *efficiency* (how resource efficient it is to execute software).

All of these software architecture sustainability metrics have similar objectives: to reduce *waste* and save resources.

However, they overlook waste resulting from resource over-provisioning, allocating more computing resources than necessary for the actual workload, which can occur when systems are designed to handle peak loads but remain underutilized for most of their operation. These metrics also fail to consider the functional usefulness of software, specifically how effectively it satisfies user requirements in relation to its resource consumption. Notably, even if a software application is optimized for computational efficiency, it may still consume a significant amount of resources while providing limited functional value, ultimately contributing to indirect resource waste.

While there is a growing demand for more and more processing capabilities production clusters are observed to function at sub-optimal utilization levels, primarily due to excessive resource reservation by software applications [141]. Delimitrou and Kozyrakis [141] demonstrated on a Twitter production cluster, that CPU utilization is consistently below 20%, even though reservations reach up to 80% of total capacity. Memory use is higher, 40–50%, but still differs from the reserved capacity. Reiss *et al.* [142] showed similar results on a Google cluster, which achieves CPU utilization of 25–35% and memory utilization of around 40%, with 75% and 60% of available capacity reserved, respectively. A Microsoft study [143] unveiled that 75% of jobs submitted to a production cluster were over-provisioned, with as much as 10 times more resources than necessary for 20% of the jobs. Although trial and error can assist software stakeholders in determining the optimal amount of computing resources needed, resource reservation was observed as mainly static: over one month, over 80% of periodic jobs showed no adjustments in their resource provisioning. Sustainability awareness has started to be integrated into software architecture evaluation methods [144], but still lacks suitable metrics that encompass the environmental consequences of over-reserved hardware resources.

In Chapter 5, we propose a conceptual framework intended to guide software stakeholders in the design of software ecodesign metrics. Within the domain of software architecture, we introduce two novel metrics specifically targeting resource waste induced by software design choices.

Background: The growth of the ICT sector has been accompanied by an increase in its environmental footprint, making it imperative for the sector to actively contribute to the path toward sustainable development. For the environmental pillar, this requires not only accounting for all phases of the life cycle of hardware involved, rather than focusing solely on energy consumption, but also tracking and reducing environmental impacts across various impact categories. Resource depletion and mineral footprints, in particular, remain a blind spot, especially in light of the ongoing energy transition.

At the software level, efforts to identify actionable reduction levers remain largely theoretical and lack concrete metrics. When metrics do exist, they primarily address energy consumption, thus overlooking critical issues such as resource waste resulting from over-reservation of resources.

Chapter 3

Assessing the Environmental Impact of the ICT Sector through Environmentally Extended Input-Output Analysis

Given the systems' complexity and rapid evolution, the ICT sector still struggles to understand its environmental impact and lacks openly available data to facilitate such assessments.

The GHGs emissions of the ICT sector has been estimated by three main studies: [67, 7, 8]. Freitag [1] *et al.* conducted the tedious work of bringing the studies to similar scopes in order to compare and assess them, as well as exchanging with their authors to obtain more up-to-date estimates.

These studies employ a hybrid approach by combining both bottom-up and top-down methodologies. However, they primarily rely on GHGs footprint assessed through attributional LCAs, which limits their ability to conduct meaningful analyses of evolutionary trends, and incurring truncation error resulting in omitting upstream emissions coming from the supply chain in their accounting. In this chapter, we first compare the boundaries of these reference studies with the *Organisation for Economic Co-operation and Development* (OECD)'s definition of the ICT sector. Following this, we utilize EEIOA to examine the evolution of the ICT sector's GHGs footprint, thereby assessing the validity of claims regarding the sector's emissions trend. Finally, we employ EEIOA to evaluate the upstream material footprint of the ICT sector, focusing specifically on three key metallic elements: copper, tin, and nickel.

3.1 Understanding the Sector Boundaries

As outlined in Subsection 2.3.1, the ITU L.1410 [102] provides a methodology for determining the environmental footprint of ICT solutions using three tiers: data centers, networks, and consumer devices. These tiers are typically considered as the boundaries when assessing the direct environmental impact of the ICT sector as a whole.

However, the definition of these boundaries, particularly concerning consumer devices, varies across different studies, resulting in outcomes that are difficult to compare. Moreover, this boundary of ICT is narrower than that defined by the *United Nations Statistics Division* (UNSD), which encompasses a broader range of economic activities related to the sector, notably service industries such as software. In this section, we first compare the carbon footprint estimates from key reference studies of the ICT sector and then examine the OECD economic definition of the ICT sector.

3.1.1 Sector Boundaries in Reference Environmental Footprint Studies

Andrae *et al.* [6] focus primarily on the use stage and the cradle-to-gate (cf. Subsection 2.2.2) production electricity footprint of a limited number of ICT devices, as well as the energy consumption of the infrastructure for networks and data centers. Consequently, their assessment only considers the electricity footprint, excluding other manufacturing aspects related to the carbon footprint. Additionally, transport and end-of-life treatment are not included. The authors employ a bottom-up approach for user devices, using LCAs combined with predicted sales numbers. The consumer devices in the study include desktops, monitors, laptops, smartphones, tablets, ordinary mobile phones, phablets, mobile broadband modems, TVs, and home entertainment systems. However, it excludes personal drones, robots, driverless automotive, portable batteries, printers and multi-function devices, digital and video cameras, music players and similar digital media devices, network-connected white goods, smart thermostats, home energy management systems, security systems, and satellites. For data centers and networks, the study uses top-down traffic data trends from Cisco for the use phase and estimates the associated electricity usage based on literature. To assess the production phase energy footprint, the researchers arbitrarily consider a fixed share of the total electricity use: 5%, 10%, and 15% for their best, expected, and worst-case scenarios, respectively. They calculate the electricity consumed to produce all devices in use in a given year, regardless of when these devices were manufactured. Consequently, this approach may furthermore underestimate the production energy for user devices.

Belkhir and Elmeligi [7] include both the production and operational energy of ICT for consumer devices in their analysis. However, they consider the embodied carbon of data centers and networks to be negligibly small, and therefore only account for their operational energy. The analysis includes user devices such as desktops, laptops, displays, tablets, and smartphones, while all other ICT equipment is out of scope, including TVs, set-top boxes, and printers. For each considered consumer device, they calculate a *Lifecycle Annual Footprint* in $kgCO_2eq$ per year by dividing the total production energy by the expected lifetime of each consumer product. To determine the embodied carbon footprint for a given year, this ratio is then multiplied by the number of devices sold during that year. For data centers, the analysis includes servers, communication, storage, cooling, and power. For communication networks, it includes customer premises access equipment (CPAE), office networks, telecom operators, cooling, and power. The estimates for data centers are derived from 2008 data provided by [145], while the network estimates are based on data from the period 2007-2012 as reported by [146]. The range of estimates in this study is significantly smaller than those reported

		Andrae [6] (including TVs)		Belkhir [7]		Malmodin (in [1])	Malmodin(updated in [147])
		Min	Max	Min	Max		
Consumer devices	Usage	350	550	143,9	202	375	228
	Embodied	-	-	176,4	417,7		208
Networks	Usage	110	149	269,1		183,5	168
	Embodied	-	-	0			34
Data centres	Usage	160	165	494,9		127	95
	Embodied	-	-	0			30
3 tiers total embodied		138	209				
Total		620	864	1084,3	1383,7	685,5	763

Table 3.1 Comparison of ICT Footprint in tCO_2eq (Excluding TVs) Across Reference Studies

in [6], primarily due to uncertainties in the carbon footprint of user devices, particularly desktops and displays. The authors indicate in [1] that including TVs could contribute an additional 435 TWh solely from operational energy use. For data centers, their estimate is notably the largest, largely due to the assumption of a *Power Usage Efficiency* (PUE) of 2.

Malmodin and Lundén [8] adopt a broader scope in their analysis, which includes not only the ICT sector but also the Entertainment & Media sector. Their analysis of consumer devices encompasses a wide range of products, including phones, desktops, laptops, computer peripherals, cameras, displays, TVs, audio devices, gaming consoles, and some Internet of Things (IoT) devices. Additionally, they consider "Other digital technologies or trends," which specifically include wearables such as smart watches and fitness trackers, smart energy meters, control units, surveillance cameras, public displays, payment terminals, and internet-connected communication devices in vending machines. Their methodology aligns with the principles used in the two aforementioned studies, as they combine sales numbers with LCAs for user devices. However, for networks and data centers, their estimates are derived from extrapolations based on data collected from company reports, and these estimates fall within the lowest range of the three studies.

The respective carbon footprints for the ICT sector in 2020, as reported in these reference studies, are presented in Table 3.1. The comparison of these studies reveals substantial differences, primarily due to variations in their scopes and methodologies. For instance, while Andrae's study encompasses a broader scope than that of Belkhir, it reports emissions that are up to two times lower, underscoring inconsistencies in how boundaries are defined. A particularly significant source of discrepancy lies in the reporting of embodied emissions, where considerable variation is observed between the studies. Andrae's study lacks detailed data for emissions, whereas Belkhir reports substantially higher values. These wide-ranging results reflect the considerable uncertainties and methodological differences that complicate the accurate estimation of the ICT sector's environmental impact.

3.1.2 Sector Boundaries According to the International Standard Industrial Classification

The ISIC is a system designed to classify worldwide economic data by industry. Developed by the UNSD, it is periodically revised to reflect changes in the global economy and advancements in industries. ISIC covers a wide range of economic activities, from agriculture and manufacturing to services and government functions.

The classification system is structured hierarchically in a four-level structure representing different levels of detail: sections are divided into divisions, which are further divided into groups, and these groups are divided into classes. For example:

- Section D - Manufacturing
  - Division 15 - Manufacture of food products and beverages
    - Group 151 - Production, processing, and preservation of meat, fish, fruit, vegetables, oils, and fats
      - Class 1511 - Production, processing, and preserving of meat and meat products

The ISIC Revision 3.1 [148], released in 2002, proposes an alternate aggregation for the ICT sector, encompassing economic activity generated by the production of ICT goods and services, following the boundaries standardized by the OECD.

Manufacturing industries included are intended to fulfill the function of information processing and communication, including transmission and display, or must use electronic processing to detect, measure, and/or record physical phenomena or to control a physical process. Service industries included should enable the function of information processing and communication by electronic means.

Table 3.2 Economic Classification of the ICT Sector in ISIC v3.1

Division	**Class**	**Description**	**ISIC definition**	**This approach**
30 - Manufacture of office, accounting and computing machinery	3000	Manufacture of office, accounting and computing machinery	✓	✓
31 - Manufacture of electrical machinery and apparatus, N.E.C	3110	Manufacture of electric motors, generators and transformers	✗	✗
	3120	Manufacture of electricity distribution and control apparatus	✗	
	3130	Manufacture of insulated wire and cable	✓	
	3140	Manufacture of accumulators, primary cells and primary batteries	✗	
	3150	Manufacture of electric lamps and lighting equipment	✗	
	3190	Manufacture of other electrical equipment n.e.c.	✗	
32 - Manufacture of radio, television and communication equipment and apparatus	3210	Manufacture of electronic valves and tubes and other electronic components	✓	✓
	3220	Manufacture of television and radio transmitters and apparatus for line telephony and line telegraphy	✓	

Continued on next page

Table 3.2 – *Continued from previous page*

Division	**Class**	**Description**	**ISIC definition**	**This approach**
	3230	Manufacture of television and radio receivers, sound or video recording or reproducing apparatus, and associated goods	✓	
33 - Manufacture of medical, precision and optical instruments, watches and clocks	3311	Manufacture of medical and surgical equipment and orthopaedic appliances	✗	✗
	3312	Manufacture of instruments and appliances for measuring, checking, testing, navigating and other purposes, except industrial process control equipment	✓	
	3313	Manufacture of industrial process control equipment	✓	
	3320	Manufacture of optical instruments and photographic equipment	✗	
	3330	Manufacture of watches and clocks	✗	
51 - Wholesale trade and commission trade, except of motor vehicles and motorcycles	5110	Wholesale on a fee or contract basis	✗	✗
	5121	Wholesale of agricultural raw materials and live animals	✗	
	5122	Wholesale of food, beverages and tobacco	✗	
	5131	Wholesale of textiles, clothing and footwear	✗	
	5139	Wholesale of other household goods	✗	
	5141	Wholesale of solid, liquid and gaseous fuels and related products	✗	
	5142	Wholesale of metals and metal ores	✗	
	5143	Wholesale of construction materials, hardware, plumbing and heating equipment and supplies	✓	
	5149	Wholesale of other intermediate products, waste and scrap	✗	
	5151	Wholesale of computers, computer peripheral equipment and software	✓	
	5152	Wholesale of electronic parts and equipment	✓	
	5159	Wholesale of other machinery, equipment and supplies	✗	
	5190	Other wholesale	✗	
64 - Post and telecommunications	6411	National post activities	✗	✓
	6412	Courier activities other than national post activities	✗	
	6420	Telecommunications	✓	
71 - Renting of machinery and equipment without operator and of personnal and household goods	7111	Renting of land transport equipment	✗	✗
	7112	Renting of water transport equipment	✗	
	7113	Renting of air transport equipment	✗	

Continued on next page

Table 3.2 – *Continued from previous page*

Division	**Class**	**Description**	**ISIC definition**	**This approach**
	7121	Renting of agricultural machinery and equipment	✗	
	7122	Renting of construction and civil engineering machinery and equipment	✗	
	7123	Renting of office machinery and equipment (including computers)	✓	
	7129	Renting of other machinery and equipment n.e.c.	✗	
	7130	Renting of personal and household goods n.e.c.	✗	
72 - Computer and related activities	7210	Hardware consultancy	✓	
	7221	Software publishing	✓	
	7229	Other software consultancy and supply	✓	
	7230	Data processing	✓	✓
	7240	Data base activities	✓	
	7250	Maintenance and repair of office, accounting and computing machinery	✓	
	7290	Other computer related activities	✓	

Table 3.2 presents the divisions and classes that are part of the ICT sector, along with their respective description. One can note that these categories are outdated when compared to the current state of the ICT industry. The sector has undergone significant changes since 2002, with notable omissions such as the lack of a dedicated class for data centers (which are comprised within data processing and database activities). ISIC is periodically revised to account for these ongoing changes in the global economy and advancements within industries.

The latest ISIC revision released in 2008, the fourth one, redefines the industries encompassed within the ICT sector. It stipluates that *The production (goods and services) of a candidate industry must primarily be intended to fulfill or enable the function of information processing and communication by electronic means, including transmission and display"*. As illustrated in Table 3.3, these updated categories more accurately reflect the contemporary structure of the ICT sector. However, the top-down approach that we use is based on the less granular 3.1 revision of ISIC, a limitation that will be further examined in the subsequent section.

3.2 Evolutionary Trends in the Sector's Carbon Footprint

Bottom-up approaches employed by current reference studies suffer from different boundaries, between themselves as well as with economic standards. They rely on LCAs, which inherently imply truncation error resulting in omitting upstream emissions coming from the supply chain (cf Subsection 2.2.2). To adjust for such omissions when assessing the ICT sector environmental footprint, Freitag *et al.* [1] used EEIOA to assess the ratio of supply chain typically omitted by LCA methodologies. They estimate that truncation error causes an omission of around 40% of the total

Class	Description
	ICT manufacturing industries
2610	Manufacture of electronic components and boards
2620	Manufacture of computers and peripheral equipment
2630	Manufacture of communication equipment
2640	Manufacture of consumer electronics
2680	Manufacture of magnetic and optical media
	ICT trade industries
4651	Wholesale of computers, computer peripheral equipment and software
4652	Wholesale of electronic and telecommunications equipment and parts
	ICT services industries
5820	Software publishing
6110	Wired telecommunications activities
6120	Wireless telecommunications activities
6130	Satellite telecommunications activities
6190	Other telecommunications activities
6201	Computer programming activities
6202	Computer consultancy and computer facilities management activities
6209	Other information technology and computer service activities
6311	Data processing, hosting and related activities
6312	Web portals
9511	Repair of computers and peripheral equipment
9512	Repair of communication equipment

Table 3.3 Economic Classification of the ICT Sector in ISIC Rev. 4

embodied carbon and around 18% of the use phase carbon. They draw these findings from their previous studies[149, 150] which uses data from the UK government, which is not representative of a worldwide approach.

However, global EEIOA data is publicly available, notably in Exiobase [151], allowing for more granular handling of truncation errors compared to the generalization of national ratios. As such, we can assess the ICT sector's environmental footprint shifting from current bottom-up approaches [1] to top-down one (cf. Subsection 2.2.1), capable of capturing impacts from the entire supply chain. EEIOA was employed by Charpentier *et al.* [152] to assess the carbon footprint of the ICT sector and to compare the results with reference studies. However, this analysis was limited to a single year and did not encompass other impact categories.

3.2.1 EEIOA theoretical background

While *Multi-Regional Input-Output* (MRIO) tables are generally employed for macroeconomic global analysis, encompassing inter-country and inter-industry flows, this study limits its scope to focus specifically on the environmental impacts within the ICT sector on a global scale. To achieve this, we adopt a single-region economy model that excludes imports and exports, thereby simplifying the equations by treating the model as representative of the global economy.

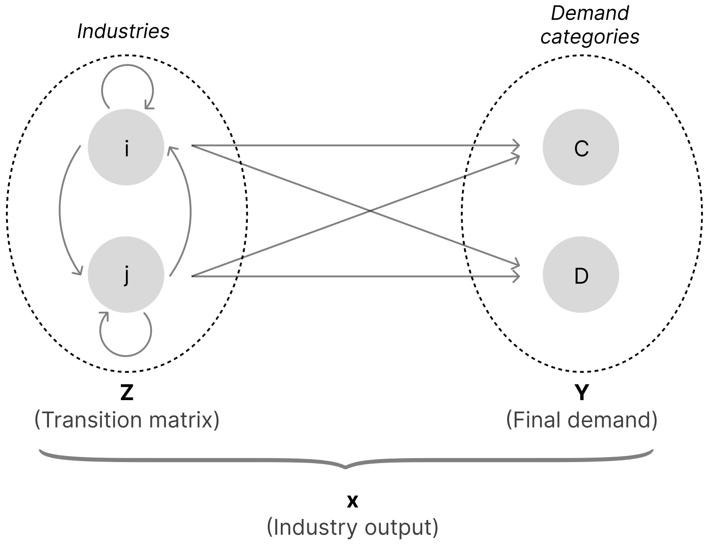

Fig. 3.1 Matrices of *Input-Output Analysis* (IOA)

Input Output Analysis

The foundation of IOA lies in the representation of global inter-industry monetary flows represented via the transaction matrix Z, defined for k industries as follows:

$$Z = \begin{pmatrix} Z_{1,1} & Z_{1,2} & \cdots & Z_{1,k} \\ Z_{2,1} & Z_{2,2} & \cdots & Z_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{k,1} & Z_{k,2} & \cdots & Z_{k,k} \end{pmatrix} \tag{3.1}$$

As such, Z_{ij} represents the trade from industry i to industry j, where i is thus the buyer and j the supplier.

Similarly, the global final demand is represented using matrix Y, where Y_{id} represents the demand for category d for industry i. Examples of demand categories include final consumption expenditure by the government, households, or non-profit organizations, changes in inventories, and gross capital formation, ... Matrices are represented graphically in Figure 3.1

The total final demand for industry i, denoted as y_i is obtained by summing across all demand categories:

$$y_i = \sum_{d=0} Y_{id} \tag{3.2}$$

The total industry output x_i represents the total value of goods and services produced by an industry or sector within an economy, encompassing both intermediate goods used as inputs in the production process and final goods sold to consumers. Consequently, the total industry output x_i for a given industry i can be described as the sum of inter-industry demand, where i is the supplier for j, plus the total final demand y_i:

$$x_i = \sum_{j=0} Z_{ji} + y_i \tag{3.3}$$

The direct requirement matrix A represents the amount of input required from each industry to produce one unit of output in each industry. It is calculated by multiplying global inter-industry flows Z with the diagonalized and inverted industry output x:

$$A = Z\hat{x}^{-1} \tag{3.4}$$

Building on the linear economy assumption inherent to IOA and the established Leontief demand-style modeling approach [153], the total output of industries can be determined for any given vector of final demand y by using the total requirement matrix, also known as the Leontief matrix L. It represents, for each industry, all direct and indirect inputs required along the supply chain to produce one unit of output delivered to final demand.

Here, I denotes the identity matrix, which is sized equivalently to the direct requirement matrix A. The calculation involves multiplying the final demand vector y by the Leontief matrix L, enabling the determination of the total output required across industries to meet the given final demand:

$$x = (\mathrm{I} - A)^{-1} y = Ly \tag{3.5}$$

This multiplication effectively captures the total production needed, considering both direct and indirect inputs across the entire supply chain, to satisfy the specified final demand.

Environmentally Extended Input-Output

As detailed in Subsection 2.2.3, IOA can be extended with various factors of production, notably to represent environmental impacts. In EEIOA, these are represented using the impact factors matrix F, where each row is an environmental impact category, and each column an industry.

Similarly to Equation 3.4, the normalized impact matrix S for an industry output x can thus be obtained via:

$$S = F\hat{x}^{-1} \tag{3.6}$$

To account for environmental impacts, two perspectives are employed: *Production-Based Accounting* (PBA) and *Consumption-Based Accounting* (CBA). PBA measures the environmental impacts generated within a region due to the production of goods and services, including those destined for export [154]. Under this approach, the region is held responsible for all emissions resulting from its production activities, irrespective of where the products are ultimately used or who accounts for the final demand [155, 156]. In contrast, CBA considers all environmental impacts associated

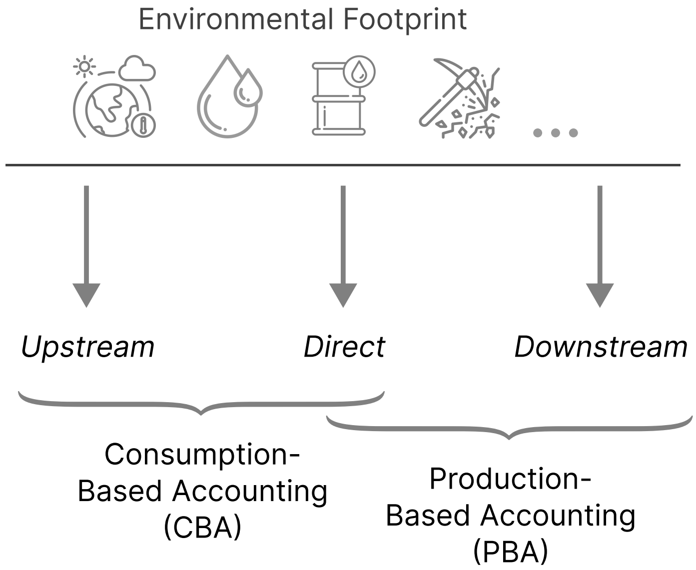

Fig. 3.2 Comparative Scopes of *Production-Based Accounting* (PBA) and *Consumption-Based Accounting* (CBA)

with the goods and services consumed within a region, regardless of where they are produced. This method attributes emissions to the region based on its consumption patterns, meaning that emissions from imported products are distributed to their regions of origin [157, 158]. The difference of *Production-Based Accounting* (PBA) and CBA accounting are summarized in Figure 3.2.

The distinction between these two approaches underscores a key difference in assigning responsibility: PBA focuses on the producers within the region, while CBA places responsibility on the consumers, taking into account the entire global supply chain. However, in our approach, we consider a single global region, which limits the applicability of distinguishing between exports and imports, as these flows are nonexistent in this context. Nonetheless, inter-industry flows still occur, enabling us to trace their impacts, particularly regarding the upstream footprint of the ICT sector. Therefore, we employ *Consumption-Based Accounting* (CBA) to assess the upstream emissions that originate from the supply chain as well as direct emissions.

To compute the CBA impact of an industry i, the impact induced by each supplying industry j in fulfilling the total final demand y_i should be summed. Specifically, this involves accounting for the contributions of all industries i that supply inputs to industry j to meet its final demand. The direct impact of industry i itself is calculated when $j = i$, representing the emissions or environmental impacts directly associated with the production activities of industry i.

$$D^i_{cba} = \sum_{j=0} L_{ij} \times y_i \times S_j \tag{3.7}$$

Similarly, to compute the PBA impact of an industry i, including both its direct and downstream impacts, all flows originating from other industries j that require inputs from industry i to fulfill their own final demand y_j should be summed as follows:

$$D^i_{pba} = \sum_{j=0} L_{ji} \times y_j \times S_i \tag{3.8}$$

The PBA perspective accounts for the total environmental impact associated with industry i by considering not only its direct emissions but also the downstream effects as it supplies goods and services to other industries j.

EEIOA database

There are several EEIOA databases available, each providing its own unique set of impact factors and transaction matrices. Some notable databases include:

- **EXIOBASE**[1]: A detailed multi-regional EEIOA database that covers 44 countries and 5 rest-of-the-world regions, focusing on a wide range of environmental indicators.
- **EORA**[2]: A global multi-regional input-output (MRIO) database that provides detailed accounts for over 190 countries, covering a broad spectrum of environmental data.
- **GTAP (Global Trade Analysis Project)**[3]: An input-output database with global coverage, widely used for economic and environmental policy analysis, including energy use and greenhouse gas emissions.
- **WIOD (World Input-Output Database)**[4]: A database offering time-series of world input-output tables linked with environmental and socio-economic data for 40 countries plus the rest of the world.
- **OECD Inter-Country Input-Output (ICIO)**[5]: Provides input-output tables and environmental data for OECD and major non-OECD economies, enabling analysis of global supply chains.
- **USEEIO (U.S. Environmentally Extended Input-Output Model)**[6]: Developed by the U.S. EPA, this database is specific to the U.S., focusing on environmental impacts across different sectors.
- **Eurostat**[7]: These are European environmental accounts linked to the national accounts, including input-output tables.

[1]https://www.exiobase.eu/
[2]https://worldmrio.com/
[3]https://www.gtap.agecon.purdue.edu/
[4]http://www.wiod.org/
[5]https://www.oecd.org/sti/ind/inter-country-input-output-tables.htm
[6]https://www.epa.gov/land-research/us-environmentally-extended-input-output-useeio-model
[7]https://ec.europa.eu/eurostat/web/environment/environmental-economy

We chose to use Exiobase 3 because it provides a broader range of environmental impact categories beyond GHGs emissions, which is particularly useful in our subsequent analysis of the mineral and material footprint. Additionally, Exiobase 3 covers an extensive time period up to recent years and includes numerous of economic sectors. It is also part of an ongoing academic research project, based on open-source data and methodologies. However, there are some drawbacks to using Exiobase. One limitation is that it utilizes the ISIC 3.1 classification, which may not fully reflect the current structure of the ICT industry as discussed in Subsection 3.1.2. Additionally, a portion of the data in Exiobase 3 is now-casted, meaning that it is estimated using available data and statistical models before official data is released [159].

Specifically, we use Exiobase 3.8.2 [160], and Pymrio 0.5.3 [161] to handle the data in Python, notably the built-in industry and country aggregation.[8] We use the industry by industry classification, based on the fixed product sales assumption, meaning that an industry's output is sold to other industries and final demand in fixed proportions, and these proportions do not change as production scales up or down. EXIOBASE provides data in constant prices to account for inflation, ensuring that environmental footprint comparisons over time reflect real changes in production and consumption rather than just price level fluctuations [82].

Table 3.2 outlines the ISIC divisions included in the ICT sector within this approach. Since Exiobase only provides data down to the division level and not to the class level, the scope of our analysis differs slightly from the standard ISIC classification. Within a given division, not all classes are included in the ISIC definition of the ICT sector. This limitation requires us to make decisions to avoid either broadening the scope too much or, conversely, omitting critical components of the ICT sector. For instance, in the case of Post and Telecommunications, postal activities are included in our analysis to ensure that telecommunications activities are fully captured. On the other hand, activities such as the renting and wholesale of computers are excluded from our analysis, as including them would require us to consider the entire wholesale or renting sector, rather than focusing specifically on ICT-related goods and services.

3.2.2 ICT carbon footprint using EEIO

From the 126 impact categories available in Exiobase, we use *GHG emissions AR5 (GWP100) | GWP100 (IPCC, 2010)*, meaning that we consider GHGs emissions according to the IPCC Fifth Assessment Report (AR5) with a *Global Warming Potential* (GWP) for 100 years [162].

Figure 3.3 presents a comparison of GHGs emissions from the ICT sector across various reference studies, with boundaries normalized by Freitag *et al.* [1], alongside Exiobase data for the years 2015 and 2020. Notably, Malmodin's study is the only one to suggest a decrease in the ICT sector's carbon footprint, while also reporting the lowest emissions among the four studies. In contrast, Andrae's study exhibits a substantial margin of error, particularly related to consumer devices, resulting in significant variability in its estimates. Exiobase is positioned at the higher end of the emissions'

[8] https://pymrio.readthedocs.io/en/latest/math.html

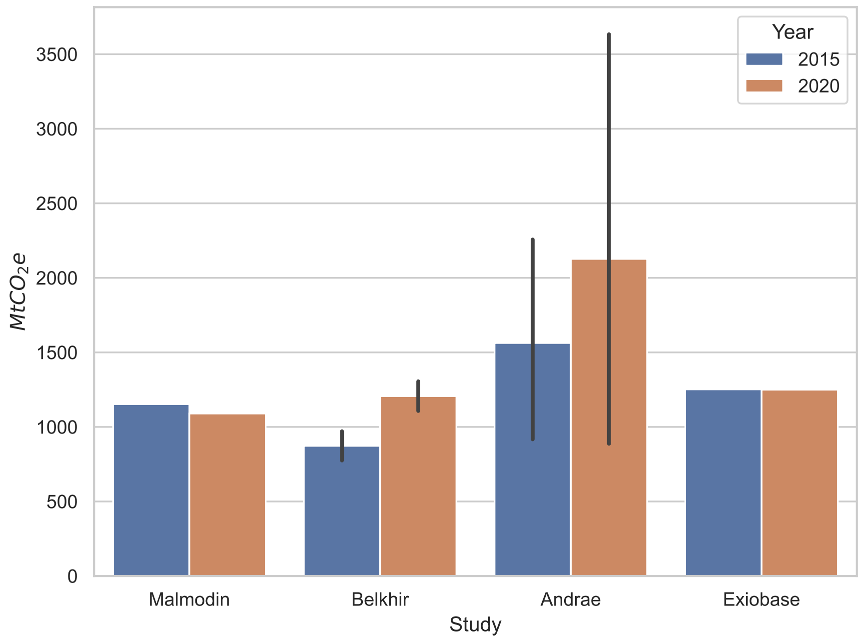

Fig. 3.3 Comparison of GHGs Emissions: Exiobase vs. Reference Studies for 2015 and 2020, with Boundaries Normalized by [1]

spectrum, except when compared to Andrae, whose estimates fall within the upper range of all the studies considered.

As previously discussed, these discrepancies can be attributed to variations in the reference data used in the studies. Nonetheless, the emissions reported by Exiobase are within the same order of magnitude as those in the reference studies despite adopting a different approach, offering a credible basis for comparison. However, similarly to [163], the common finding that EEIOA-based carbon footprints should be higher than LCA-based carbon footprints could not be confirmed. Interestingly, Exiobase suggests a stagnation in the ICT sector's carbon footprint between 2015 and 2020, which hide more complex variations.

Exiobase is updated regularly with more up-to-date data. For each year, both economic and environmental data are handled using the same scope, boundaries, and methodologies. This consistency allows for reliable comparisons of footprints across multiple years, facilitating the identification of trends over time.

Figure 3.4 illustrates the evolution of the GHGs footprint within the disaggregated ICT sector, revealing a general upward trend in carbon emissions across all industries over the 25-year period. For each year, Equation 3.7 is applied using the four industries that comprise the ICT sector according to ISIC v3.1 (cf. Subsection 3.1.2). While Figure 3.3 indicated a similar footprint for the sector between 2015 and 2021, this figure provides a more nuanced perspective. It shows that emissions levels rose steadily from 1995 to 2015, followed by a slower rate of increase between 2015 and 2020, and then a new spike in 2021, aligning more closely with findings from reference studies.

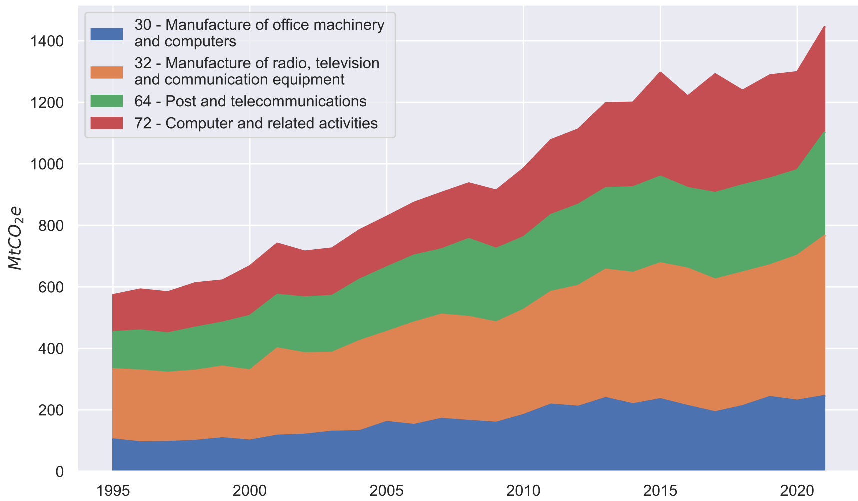

Fig. 3.4 Evolution of the GHGs Emissions Trends of the ICT Sector (CBA, 1995-2021)

The *Post and Telecommunications* industry exhibits more moderate growth in emissions compared to other sectors. The *Manufacture of Office Machinery and Computers* shows an increase after 2000, but its overall emissions remain lower relative to other industries. In contrast, the service industry *Computer and Related Activities* experiences a significant spike starting around 2010, with emissions rising sharply and eventually surpassing those of *Post and Telecommunications* by the end of the period. Despite the steady growth in ICT services, the manufacturing industries continue to dominate GHGs emissions. This indicates that the environmental impact from producing electronics and communications equipment is increasing more rapidly than that of service-oriented sectors, suggesting that the manufacturing processes tied to ICT infrastructure are a significant driver of emissions growth.

If we break down the ICT GHGs emissions for 2020 using Equation 3.7, as shown in Figure 3.5, it is evident that direct emissions are relatively low compared to upstream emissions induced by device manufacturing supply chain. Among upstream emissions, the energy sector has the most substantial impact, primarily due to coal-based production. However, it is not the only upstream sector contributing to emissions, highlighting that focusing exclusively on energy consumption is insufficient to reduce the environmental footprint of ICT.

The reference studies use a more restricted scope of the ICT sector compared to the ISIC's definition of economic activities. This can be partly attributed to the usage of bottom-up approach, which involves considering the number of devices sold and their footprint through LCAs. However, this approach introduces two major sources of uncertainty: the number of devices accounted for each and the impact assessments methods used. In contrast, the EEIOA approach aligns with the economic definition of sectors and employs a top-down methodology. This approach simplifies the analysis by providing a more comprehensive accounting based on financial flows but sacrifices some

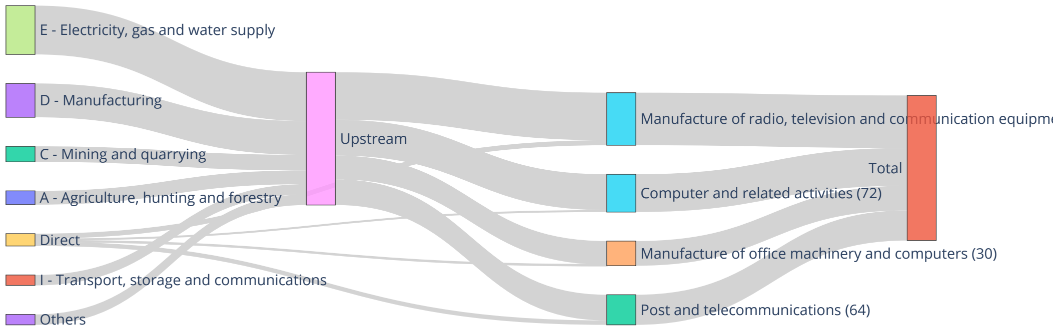

Fig. 3.5 Breakdown of Direct and Upstream GHGs Emissions in the ICT Sector (2020)

granularity in feedback compared to the bottom-up approach. We also observe that a more recent ISIC classification could lead to improved data accuracy.

However, the EEIOA method allows for an easier assessment of the sector's footprint on other impact categories, which we will explore in the next section.

3.3 Evaluating the ICT Sector's Mineral Footprint Using EEIOA

As discussed in Subsection 2.1.3, the sector's footprint is not solely limited to carbon emissions, and its ramification goes further than global warming. However, the interdependent nature of EEIOA notably facilitates the elucidation of critical links between the ICT sector and mineral industries.

3.3.1 Analysis on specific minerals

For metals and minerals, Exiobase primarily utilizes data from the World Mineral Statistics developed by the British Geological Survey [164], along with data from the International Minerals Statistics and Information published by the US Geological Survey [165], and the World Mining Data provided by the Austrian Ministry for Economy and Labour [166].

In this section, we focus on three critical elements: copper, nickel, and tin. Copper is essential for data centers, particularly in servers, power distribution, cooling, and interconnections.[9] Nickel is crucial for energy storage in batteries due to its high energy density, stability, thermal safety, and excellent electrical conductivity. Tin is a key component in solder because of its low melting point,

[9]https://www.visualcapitalist.com/sp/copper-the-critical-mineral-powering-data-centers/

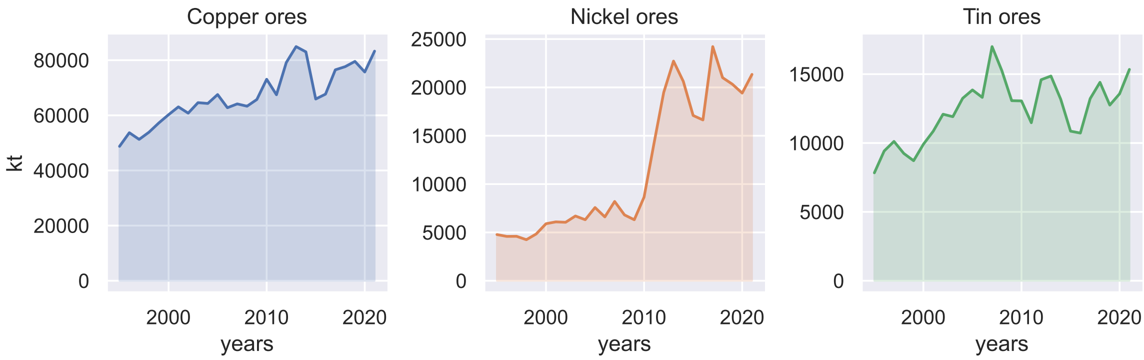

Fig. 3.6 Trends in Copper, Nickel, and Tin Ore consumption by ICT (Exiobase, CBA)

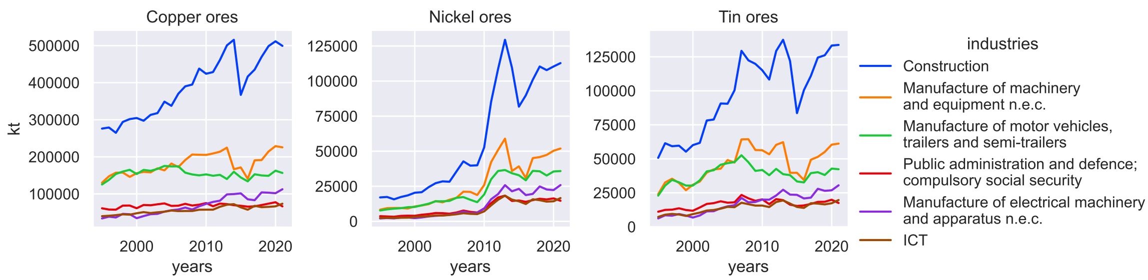

Fig. 3.7 CBA Trends in Copper, Nickel, and Tin Ore Consumption: ICT vs. Five Largest Consumer Industries (1995-2021)

good conductivity, and strong adhesive properties. It is often alloyed with other metals, such as lead or silver, to enhance its properties, which complicates its recycling potential.

Copper, nickel, and tin ore consumption demonstrate an increasing consumption trend by the ICT sector, as illustrated in Figure 3.6, which sum the footprint of individual sectors using Equation 3.7 in a manner similar to Figure 3.5. During this period, copper consumption by the sector nearly doubled, the use of nickel ores has significantly escalated since 2010, and the consumption of tin ores follows a similar upward trajectory, albeit at a less pronounced rate. The drop in 2015 can be explained by a global recession in the mining sector, which was driven by weaker global demand and the slowdown in China's economic growth [167]. However, these trends are anticipated due to the surge in ICT device production, which ultimately drives up the demand for metal and mineral resources. Nevertheless, considering planetary boundaries, this upward trend can be seen as unsustainable given the finite availability of these critical resources, especially in the context of the ongoing energy transition (cf. Subsection 2.1.3).

Assessing the consumption patterns of a single sector is not sufficient to comprehend the complexities of mineral footprints. Figure 3.7 illustrates the trends for copper, nickel, and tin ores among the five largest consumer sectors along with the ICT sector, revealing similar upward trends across all sectors. Notably, the Construction sector emerges as the largest consumer of these minerals. For all the examined minerals, the ICT sector shows the lowest consumption levels, significantly overshadowed

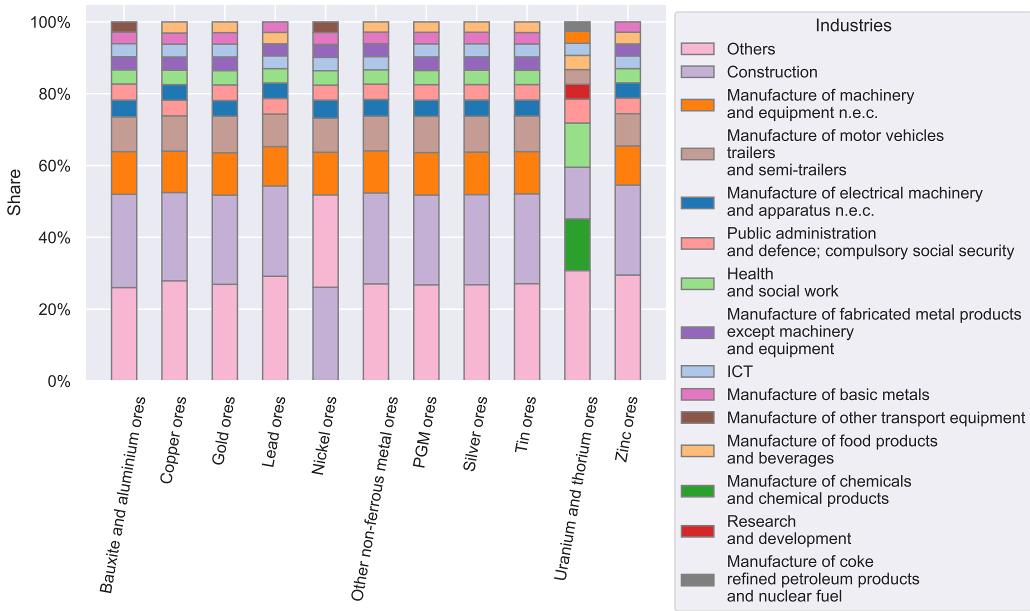

Fig. 3.8 Cumulative Share of the 10 Largest Consumers Over the Period

by the five major industries. This disparity underscores potential future challenges in securing these essential minerals, especially when the needs of the ICT sector compete with those of the largest consumers.

3.3.2 Challenges and Limits of the Methodology

While using EEIOA to assess mineral footprint allows for trend analysis, showing an increasing consumption of minerals each year, it is accompanied by high uncertainties. Figure 3.8 displays the normalized cumulative share of the 10 largest consumers of minerals between 1995 and 2021. For each metal and mineral available within Exiobase, except for uranium and thorium ore, the major consumers show consistent responsibility across the minerals. This suggests that while the extraction quantities taken from mineral databases are accurate, the normalization process to monetary flows hinders granular analysis for individual minerals.

As discussed in Section 3.1, the reference studies on the environmental footprint of the ICT sector primarily rely on hybrid approaches, notably based on LCAs. These approaches, however, inherit limitations associated with the ADP indicator, which aggregates various abiotic resources, thereby hindering detailed analyses of specific metals and minerals. Although Exiobase provides data on metals and minerals, it remains limited and insufficient to fully address these challenges. Future research may benefit from employing alternative databases, such as the Global Material Flows Database,[10] as used in [62]. Nevertheless, Exiobase remains the most comprehensive database for representing material extractive sectors.

[10] https://www.resourcepanel.org/global-material-flows-database

The analysis of the ICT sector using EEIOA, and specifically Exiobase, reveals that although the consumption of metals and minerals by the ICT sector has increased significantly since 1995, it remains a relatively low consumer compared to other sectors. The Construction sector, in particular, has experienced the most rapid growth in mineral consumption. Given the potential future depletion of available minerals and their respective criticality Subsection 2.1.3, the ICT supply chain could face uncertainty in securing essential resources.

Mineral impacts are multi-faceted and need metrics shared among several domains to be understood and collectively acted upon. The assessment of mineral impacts cannot be correctly captured by a single indicator, as the $kgCO_2eq$ indicator captures climate ones. Indeed, available resources widely differ per host mineral, as do financial, environmental & social mining costs. Depending on its criticality, a single mineral can block the whole ICT supply chain, even if used only in limited quantity. [168] case study highlights that climate change has negatively impacted mines over the past decade, yet most infrastructure is designed with outdated climate assumptions, and adaptation planning is limited.

All the more, sourcing by-products aggravates supply difficulties. *By-products*, contrarily to main or co-products, result from smelting & refining and generate less than 1 % of a mine's revenue. Their concentration and, therefore, their economic reality for mining investors, is only known once the mine is in operation and its smelting & refining infrastructure is running [47]. The unavailability of a single one—caused by any supply issue of their main companion minerals—may, therefore, affect the sector as a whole. In future research, the application of EEIOA could be employed to analyze the impacts on the ICT sector by utilizing more comprehensive and current databases containing detailed information on metals and minerals.

ICT sector's footprint: Reference studies on the ICT sector's environmental footprint vary in their boundary definitions, and differ from the economic definition of the sector. All of these studies employ a hybrid bottom-up and top-down approach, which perpetuates the limitations of the ADP impact indicator in LCA, as it aggregates multiples metals and minerals into a single category. This aggregation hinders meaningful analysis of the sector's mineral footprint. *Environmentally Extended Input-Output Analysis* (EEIOA) is a top-down approach that combines economic and environmental data, enabling the study of environmental impacts embedded in monetary flows. This method enables the assessment of a sector's upstream footprint, including the mineral supply chain of the ICT sector. By using normalized data, EEIOA also facilitates trend analysis, demonstrating both the increasing GHGs emissions and mineral consumption of the ICT sector.

In this chapter, we apply this approach to address the absence of trend data concerning the environmental footprint across multiple impact categories within the ICT sector, while maintaining alignment with its economic definition. However, such analyses remain constrained by the quality of the reference environmental data, resulting in significant uncertainty.

Chapter 4

Bottom-Up Approaches to Evaluating Software Environmental Footprint

While macro-level top-down studies are essential for driving transformative change and impulse policymaking, they fall short of providing concrete, actionable insights that software practitioners can readily apply. Given the growing environmental footprint of the ICT sector assessed in Chapter 3, digital services should acknowledge their responsibility and transition towards more sustainable practices.

However, as mentioned in Subsection 2.3.3, software practitioners currently lack the appropriate tools and models to accurately assess and reduce the environmental impact of their work. When such models are available, they often fall short of offering a comprehensive perspective, as they tend to focus solely on energy consumption, while overlooking emissions and impacts generated during the other stages of hardware life cycle.

In this chapter, we propose bottom-up tools designed to provide actionable insights that align with the three-tier framework defined in the ITU L.1410 [102], addressing two of these tiers: backend infrastructure and user devices. We also present a systematic methodology for assessing and identifying sources of uncertainties in environmental analyses. Finally, we conclude with the development of a tool that enables comprehensive life cycle analyses of software products, encompassing all phases of the *Software Development Life Cycle* (SDLC) and addressing a wide range of impact categories.

4.1 Assessing Cloud Services' Environmental Impacts

Confronted with the growing environmental footprint of backend infrastructures, major cloud providers, including AWS [169], Azure [170], and GCP [171], have introduced tools that enable customers to assess the carbon footprint associated with their cloud usage. However, the underlying methodologies adopted by these actors have not been subjected to rigorous transparency procedures, which presents a significant threat to ensuring the consistency of results between tools. This lack of transparency results in inconsistencies that hinder the comparability of results, due to variations in the scope considered and allocation methods employed.

To address these limitations, the open source project *Cloud Carbon Footprint* (CCF) [172] introduced a provider-agnostic approach, but its calculation remains constrained by several factors. For the *resource extraction & manufacturing* phase, the impacts are systematically estimated based data from the Dell R740 LCA [2], irrespective of the actual hardware components present in the system under study, thus potentially leading to an *empty-shell bias* (defined in Subsection 2.2.2). Power consumption is modeled using benchmarks that measure power at the machine level [173], which the CCF associates with *cloud instances* based on CPU architecture. However, CPU architecture appears to be an inadequate proxy for estimating power consumption, not only for the CPU itself but even more so for the entire machine. For instance, in a storage server, the primary source of power consumption is typically the hard drives, underscoring the need for a more granular power model tailored to each specific component. Moreover, since this approach does not consider the actual load factor of the hosting platform, it overlooks the potentially significant unused resources, whose environmental impact is entirely disregarded.

All of these methodologies currently used to assess the environmental impact of cloud infrastructures focus exclusively on the carbon footprint dimension. However, cloud infrastructures also have significant environmental impacts on other dimensions, such as those resulting from metal extraction [174, 33, 175]. This narrow focus on carbon, often referred to as *carbon tunnel vision* [176], obscures other environmental issues that could become critical and potentially risks enabling the transfer of environmental category to another [177, 178].

Developing a comprehensive understanding of the environmental implications of cloud computing equipment and services, including servers and instances, is essential for informed decision-making. In this section, we define a methodology that leverages openly accessible data to ensure that the results are transparent, reproducible, and verifiable. Two types of open data are used:

- **Market & technical data**: characteristics of components, devices, and cloud instances available on the market.

- **Impact factors**: which convert physical quantities into environmental impacts. Those factors are extracted from publicly available life cycle assessments.

Modeling approaches As illustrated in Figure 4.1, the environmental impacts of a *cloud platform* are mostly modeled using a *bottom-up* approach, whereby the impacts of each resource required to fulfill the service it provides are aggregated. Unlike a *top-down* approach, such as EEIOA Section 3.2, this method enables the identification of the proportional contribution of each resource to the overall environmental impacts. The impacts of the technical and building environment are allocated on the *cloud platform* using a top-down approach.

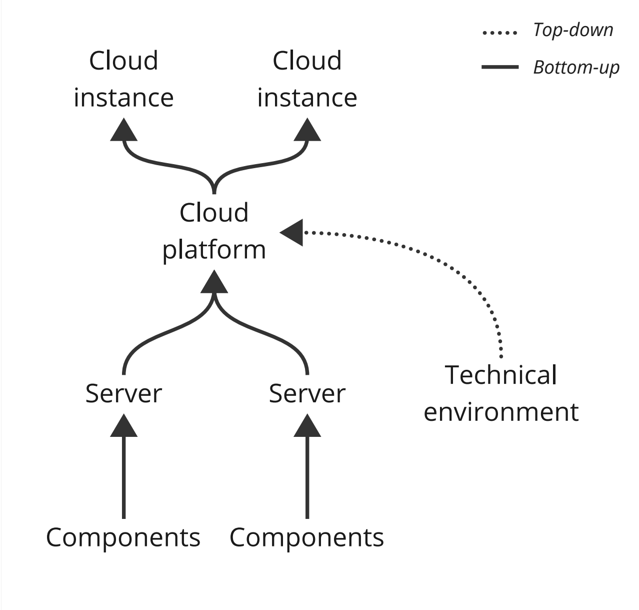

Fig. 4.1 Modeling Approach Used for Estimating Environmental Impacts of a Cloud Platform and Its Instances

4.1.1 Server Environmental Footprint Modeling

Following a bottom-up approach (cf. Figure 4.1), the assessment of the environmental impacts of *cloud instances* starts with a component-level assessment of servers. Their impacts are then aggregated and integrated with the technical and building environment to constitute a *cloud platform*, which is subsequently allocated into individual *cloud instances*.

In this section, we define the foundational layer of the model—the estimation of a server's environmental impact—and further validate it against the LCA of a Dell R740 [2]. This modeling approach is implemented within the open-source project BoaviztAPI [179]. While similar bottom-up methods already exist to do so [180], our approach aims to deliver a more detailed and comprehensive calculation of the embodied footprint, multi-criteria impacts, and a calculation of the usage impact specific to our use case.

In the rest of this thesis, variables denoted as $\mathscr{F}$ represent *impact factors*, which quantify the environmental impacts associated with a specific unit of measurement, while we denote with $\mathscr{I}$ environmental impacts that do not depend on a given quantity and are thus expressed solely in the impact unit. For example $\mathscr{F}_{cpu}^{die}$ is the impact factor quantifying the impact of one mm^2 of CPU die while $\mathscr{I}_{case}^{cpu}$ is a fixed-cost environmental impact for CPU manufacturing, which does not depend on the surface area of the die.

For both $\mathscr{F}$ and $\mathscr{I}$ variables, we propose reference values for the following *impact criteria*:

Constant	Impact	Value	Unit
$\mathscr{F}^{die}_{cpu}$	GWP	1.97	$kgCO_2eq$ /mm2
	ADP	5.87e−07	kgSbeq/mm2
	PE	2.65e+01	MJ/mm2
$\mathscr{I}^{base}_{cpu}$	GWP	9.14	$kgCO_2eq$
	ADP	2.04e−02	kgSbeq
	PE	156.43	MJ
$\mathscr{F}^{die}_{flash}$	GWP	2.20	$kgCO_2eq$ /cm2
	ADP	6.30e−05	kgSbeq/cm2
	PE	2.73e+01	MJ/cm2
$\mathscr{I}^{base}_{ram}$	GWP	5.22	$kgCO_2eq$
	ADP	1.69e−03	kgSbeq
	PE	74.00	MJ
$\mathscr{I}^{base}_{ssd}$	GWP	6.34	$kgCO_2eq$
	ADP	5.63e−04	kgSbeq
	PE	73.98	MJ
$\mathscr{I}^{e}_{hdd}$	GWP	3.11e+01	$kgCO_2eq$
	ADP	2.50e−04	kgSbeq
	PE	2.76e+02	MJ
$\mathscr{F}^{e}_{psu}$	GWP	2.43e+01	$kgCO_2eq$ /kg
	ADP	8.30e−03	kgSbeq/kg
	PE	3.52e+02	MJ/kg
$\mathscr{I}^{e}_{motherboard}$	GWP	6.61e+01	$kgCO_2eq$
	ADP	3.69e−03	kgSbeq
	PE	8.36e+02	MJ
$\mathscr{I}^{e}_{assembly}$	GWP	6.68	$kgCO_2eq$
	ADP	1.41e−06	kgSbeq
	PE	6.86e+01	MJ
$\mathscr{I}^{e}_{rack}$	GWP	1.50e+02	$kgCO_2eq$
	ADP	2.02e−02	kgSbeq
	PE	2.20e+03	MJ

Table 4.1 Impact Constants Extracted from [3] by Boavizta [4]

- *Abiotic Resource Depletion of minerals and metals* (ADP), which assesses the use of minerals and fossil raw materials. This indicator is a model for assessing the contribution of mineral and metal extraction to their progressive scarcity, using antimony grams equivalent as a metric [181].
- *Primary Energy* (PE), which includes all energy, direct and indirect, used in any phase of the life cycle. This represents the total cumulative energy demand of the assessed system [182].
- *Global Warming Potential* (GWP), which evaluates the effects on global warming. This well-known indicator is also a model linking greenhouse gas emissions to global warming. It is expressed in grams of CO_2 equivalent [162].

Embodied impact of a physical server

The impact assessment relies on the life cycle modeling of hardware components, encompassing their most important phases: raw material acquisition, and manufacturing, collectively termed as *embodied impacts*, annoted ^{e}. In all subsequent equations, embodied impact factors $\mathscr{F}^e$ are amortized over the life cycle expectancy $\mathscr{D}$ (in hours) to obtain the embodied impact for one hour of usage.

CPU For most electronic components, the primary source of impacts lies in the process of engraving semi-conductors [183]. Consequently, their impacts directly depend on both their die size and the engraving technology employed.

For CPUs, the die size in mm^2 is multiplied by the corresponding impact factor $\mathscr{F}_{cpu}^{die}$ with a base impact $\mathscr{I}_{cpu}^{base}$ (containing packaging, heatsink socket, and transportation) to obtain an environmental impact factor per hour of usage:

$$\mathscr{F}_{cpu}^{e} = \frac{die_{cpu} \times \mathscr{F}_{cpu}^{die} + \mathscr{I}_{cpu}^{base}}{\mathscr{D}} \tag{4.1}$$

For both of these impact constants, we propose default values developed in Table 4.1 over the three considered impact categories in Table 4.1, based on a 14 nm engraving process.

CPU die sizes are extracted from the TechPowerUp CPU specs database [184]. Crowd-sourced characteristics for more than 1750 CPU models are available within the BoaviztAPI.[1]

Using Equation 4.1 for two CPU units with a die size of 6.94 cm^2, the non-amortized embodied carbon footprint is estimated at 45.62 $kgCO_2eq$, while it accounts for approximately 46.76 $kgCO_2eq$ in Dell's LCA.

NAND memory The impact factors per hour of both SSD and RAM sticks are obtained with the Equation 4.2, albeit utilizing distinct impact factors for their respective density and capacity.

$$\forall flash \in \{ssd, ram\} : \mathscr{F}_{flash}^{e} = \frac{\frac{capacity}{density} \times \mathscr{F}_{flash}^{die} + \mathscr{I}_{flash}^{base}}{\mathscr{D}} \tag{4.2}$$

For the twelve 32GB sticks using a density of 1.79 GB/mm^2 sourced from [3], we obtain 534.60 $kgCO_2eq$ using Equation 4.2, while the Dell LCA reports 553.33 $kgCO_2eq$.

The server combines multiple SSD disks. Considering a density of 19 GB/cm^2 extracted from [185], we obtain 52.65 $kgCO_2eq$ for the 400 GB disk with Equation 4.2, and $3,607.77$ $kgCO_2eq$ for the 8 3.84 TB ones, versus 64.1 $kgCO_2eq$ and $3,373.5$ $kgCO_2eq$ for the R740, respectively. More up-to-date crawled SSD densities are available within BoaviztAPI.[2]

Others Apart from CPU, memory, and storage, other components are required for a server which we categorized as *others*.

The *Power Supply Unit* (PSU)'s embodied impact is estimated at 72.71 $kgCO_2eq$ using the impact factor $\mathscr{F}_{psu}^{e}$ and the R740 PSU weight of $2,992$ kg [2]. The rest of the component's impact factor is

[1] https://github.com/Boavizta/boaviztapi/blob/main/boaviztapi/data/crowdsourcing/cpu_specs.csv
[2] https://github.com/Boavizta/boaviztapi/blob/main/boaviztapi/data/crowdsourcing/ssd_manufacture.csv

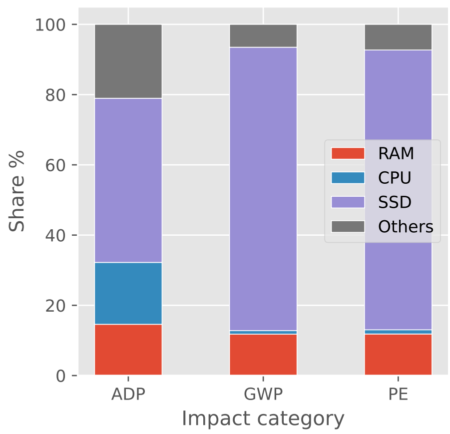

Fig. 4.2 Bottom-Up Modeling of the Manufacturing Environmental Impacts of a Dell R740, with Characteristics Defined in [2], Using Our Approach

static in our approach and computed as follows:

$$\mathscr{F}^{e}_{others} = \frac{\mathscr{I}^{e}_{motherboard} + \mathscr{I}^{e}_{psu} + \mathscr{I}^{e}_{assembly} + \mathscr{I}^{e}_{casing}}{\mathscr{D}} \tag{4.3}$$

By using the open source factors provided in Table 4.1, the remaining server's components account for 295.49 $\mathrm{kgCO_2eq}$, compared to 207.07 $\mathrm{kgCO_2eq}$ reported in the R740 LCA.

Total The total embodied impact factor of a server per hour of usage is finally calculated as the sum of its components' impacts as follows:

$$\mathscr{F}^{e}_{server} = \mathscr{F}^{e}_{cpu} + \mathscr{F}^{e}_{ram} + \mathscr{F}^{e}_{storage} + \mathscr{F}^{e}_{others} \tag{4.4}$$

This results in a total modeled life cycle value—i.e. non-amortized by D—of $4,536.13$ $\mathrm{kgCO_2eq}$ against a $4,244.76$ $\mathrm{kgCO_2eq}$ baseline reported in the Dell R740 LCA [2]. However, without transparent factors, modeling methodology, and allocation choices provided, this 6.42% variation cannot be further detailed, emphasizing the need for open methodologies.

The comprehensive evaluation of environmental impacts across various categories is depicted in Figure 4.2, emphasizing the necessity of adopting holistic methodologies in impact assessments to achieve thorough evaluations of ICT impacts. These impacts vary across different impact categories, highlighting the need to avoid carbon-centric perspectives to reveal potential shifts in impact distribution among various categories.

Usage impact of a physical server

The total energy consumption of a server $\mathscr{E}_{server}$, in Wh, is calculated as the sum of its components (C) respective power consumption $\mathscr{P}_c$, in W, over a given duration $\mathscr{T}$, in hours:

$$\mathscr{E}_{server} = \sum_{c \in C} (\mathscr{P}_c \times \mathscr{T}) \tag{4.5}$$

To obtain the associated environmental impact factor for one hour of usage $\mathscr{F}^u_{server}$, this consumption is multiplied by the impact factor $\mathscr{F}_{em}$ representing the electricity mix—*i.e.*, the environmental impacts associated to the production and transport of energy.

4.1.2 Modeling a Cloud Platform and Its Services

According to Figure 4.1, the subsequent stage of the modeling process is to analyze the impacts of *cloud platforms*, which will be used in the assessment of *cloud instances*.

Embodied & usage impacts of a cloud platform

We define a *cloud platform* as the aggregation of a cluster of servers and their technical and building environment required to provide cloud services. A *cloud platform* offers a pool of resources assigned to cloud services: *virtual CPU* (vCPU), *virtual RAM* (vRAM), storage, and shared resources—power supply, motherboard, casing and technical environment.

To account for the embodied impacts of the technical and building environment, we refer to the study published by ARCEP [186], which reports on embodied impact factors per m^2 of a server room. The technical environment includes the building, generators, chillers, inverters, and batteries, as well as a wide range of equipment such as electrical and network cables, lighting, fuel oil storage tanks, etc. Knowing the electricity consumption per m^2 and the PUE used in the study [186], we infer the embodied impacts of the technical environment $\mathscr{F}^e_{DC}$ as 2.265e−02 kgCO_2e/kWh, 5.740e−01 MJ/kWh, and 1.016e−06 kgSbeq/kWh for respectively GWP, PE and ADP criteria. As such, the embodied impact of the technical and building environment (without network equipments) can be estimated for a given energy consumption $\mathscr{E}$ as follows:

$$\mathscr{I}^e_{env}(\mathscr{E}) = \mathscr{E} \times \mathscr{F}^e_{DC} \tag{4.6}$$

In order to account for the usage impacts of the technical and building environment, the PUE of the cloud infrastructure is applied to the usage impacts of the server. The PUE is defined as the ratio of electricity consumed by the facility to the electricity consumed by the IT equipment [187].

The usage impacts are therefore defined as follows for a given energy consumption $\mathscr{E}$, using the electricity mix $\mathscr{F}_{em}$:

$$\mathscr{I}^u_{env}(\mathscr{E}) = \mathscr{E} \times (PUE - 1) \times \mathscr{F}_{em} \tag{4.7}$$

Embodied & usage impacts of a Cloud instance

A *cloud instance* is modeled as a part of a *cloud platform*. Its impacts encompass both a share of the technical and building environment impacts, $\mathscr{I}_{env}$, and a share of each of the servers' components impact factor $\mathscr{F}_r$, where r denotes the resource (vCPU, vRAM, ...). This share is computed for each component using the quantity assigned to the instance $\mathscr{Q}^u_{res}$ over the total available resources on the platform $\mathscr{Q}^u_{res}$. Such quantity $\mathscr{Q}$ materializes as vCPU for CPUs and GB for RAM and disks. For the *others* resources, which are not explicitly assigned to a *cloud instance*—i.e. motherboard, PSU, assembly, and casing (cf. Equation 4.3)—we have chosen an allocation based on vCPU, although this may depend on the type of *cloud instance*.

$$\mathscr{F}_{inst} = \mathscr{I}_{env}(\mathscr{E}_{inst}) + \sum_{\substack{r \in \{cpu, \\ ram, ssd, others\}}} \frac{\mathscr{Q}^u_r(instance) \times \mathscr{F}_r}{\mathscr{Q}^u_r(platform)} \tag{4.8}$$

Equation 4.8 allows to obtain the instance embodied and usage impacts for an hour, each share is multiplied by the *cloud platform* total impacts for the resource ($\mathscr{F}_r$). This assumes that the server is used continuously over its lifetime. To estimate the share of technical and building environment associated with the instance, Equation 4.6 is used with the instance energy consumption $\mathscr{E}_{inst}$. For one-hour usage, this energy consumption is estimated using Equation 4.5, using the ratio $\frac{\mathscr{Q}^u_r(instance)}{\mathscr{Q}^u_r(platform)}$ to obtain for each component the share to allocate to the instance.

To illustrate the aforementioned approach, we consider a *cloud platform* comprising a single server with technical characteristics and environmental impacts defined in Subsection 4.1.1, with a lifespan of 5 years. We consider that the infrastructure is hosted in France where $\mathscr{F}_{em}$ is equal to $0,098$ kgCO_2eq /kWh [188]. The PUE is set arbitrarily to 1.5. CPU power is estimated at 104.75 W, and each SSD disk consumes 5.7 W.

The impact of a year of usage for a *cloud instance* with 4 vCPUs, 8 GB of RAM, and 80 GB of SSD storage are estimated using Equation 4.8 in Figure 4.3. This highlights the advantages of a bottom-up approach, which allows for the identification and analysis of the environmental footprint of individual components within the overall system. One can observe that contrary to the server in Figure 4.2, the impact of SSDs is smaller due to a small share of disks reserved. The technical and building environment substantially increase the total impacts, both embodied and usage. Using an alternative emission factor $\mathscr{F}_{em}$, such as $0,0580$ kgCO_2eq /kWh proposed for France in 2023 [189], would yield comparable trends. However, locating the cloud platform in a different country with different energy production sources could result in significantly different outcomes.

4.1.3 Limits and Future Work

This approach does not cover the entire perimeter of a cloud infrastructure, including third-party services hosted in the *cloud platform* that serve multiple customers. These can be technical services, such as the control plane, or customer services, such as billing. In addition, the usage impacts of

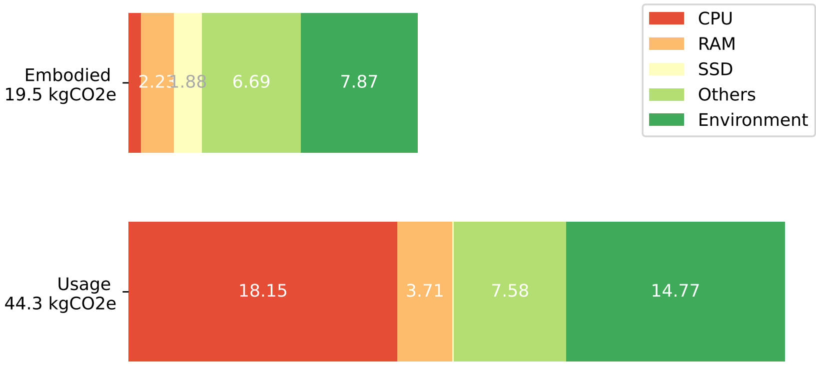

Fig. 4.3 GWP Impact of a Cloud Instance Hosted in France with 4 vCPUs, 8 GB of vRAM, and 80 GB of SSD Storage, Used for One Year (Calculated Using Equation 4.8)

non-server IT equipment, such as network equipment, are not taken into account. This also minimizes the impacts of the technical and building environment, as we do not consider their consumption when applying the PUE. Finally, we do not consider servers in idle states, that are part of the cloud infrastructure and allow for rapid and important scaling. This exclusion is primarily due to the necessity of empirically validating an appropriate allocation ratio for such resources. This assumes that all servers are allocated and occupied, which is not always the case [190]. These exclusions mean that certain parts of the infrastructure impacts remain unallocated to customers.

In addition, our allocation strategy poses some issues. It assumes that no resource is overcommitted (when a resource is used by two or more cloud instances at the same time). If a resource is overcommitted [191], all the instances sharing a physical resource would double account its impacts.

While a top-down approach would allow for a broader scope, the bottom-up approach allows us to pinpoint which part of the *cloud instance* is responsible for the most impacts, empowering stakeholders to identify actionable reduction levers. The implementation in BoaviztAPI [179] has been notably utilized by companies, such as Orange,[3] Sweep,[4] and Sami,[5] as well as for conducting GHGs assessments in France. The research community has also used it as a basis to estimate and reduce various ICT aspects' environmental footprint, such as Kubernetes scheduling, AI, infrastructure management... [192–197].

The currently released version, accessible at https://dataviz.boavizta.org/serversimpact, lacks the *cloud platform* aspect defined in this chapter, and only goes to *server* level, thus undermining the associated environmental footprint.

[3] https://orange.fr
[4] https://sweep.net
[5] https://sami.eco

4.2 Addressing Uncertainties through Fuzzy Logic

When assessing the environmental impact of digital services, the ITU L.1410 [102] standard also includes end-user equipment, or user devices (see Subsection 2.3.1). LCAs depend on LCI databases for secondary data, which provide reference environmental impact factors for various resource types. However, in the fast-evolving ICT sector, such data are often unavailable (cf. Subsection 2.2.2), resulting in the use of substitutes which can introduce significant inaccuracies within the results.

Furthermore, the embodied impact data for these devices can only be estimated and thus cannot be empirically validated. The lack of a definitive *ground truth* for comparison complicates the process of quantifying uncertainties. Additionally, impact estimation inherently involves allocation choices made during the modeling process. The results of such analyses then tend to be broad estimates with significant uncertainty, which are often insufficiently quantified and documented, yet have a substantial influence on the final outcomes.

In this section, we introduce a novel approach utilizing Fuzzy Logic for the aggregation of secondary sources and the systematic assessment of associated uncertainties. We then demonstrate how this approach enables stakeholders to identify sources of uncertainty within the environmental impact assessment modeling process, particularly concerning allocation choices related to end-user devices.

4.2.1 Secondary Data

Assessing Uncertainty with Data Quality Indicators

To address the lack of reference data in the assessment of ICT environmental impacts, it is necessary to combine multiple data sources, each of which may vary in quality and provide potentially divergent estimates. For instance, manufacturers report various embodied impacts of a smartphone such as 33, 57, or 94 $\mathrm{kgCO_2eq}$ [198–200]. Such variations can be caused by divergences in characteristics, manufacturing process, or the LCA methodology and study boundaries chosen. They can significantly influence the final estimated impacts and should be propagated within all computations to be exposed in the final estimation.

Consequently, we propose to adjust the relative weight assigned to each of these sources in the final results according to their relative quality. To quantify this quality, each LCI source is evaluated using a DQI, following the method introduced by Weidema *et al.* [201]. Specifically, the DQI of a source covers 3 key aspects: its *reliability*, *temporality*, and *technological correlation*. The *technological correlation* highlights the similarity between the variable assessed by the source and the variable to model. For instance, when assessing the efficiency of a smartphone charger, studies regarding smartphone chargers have a higher technological correlation than studies focusing on laptop chargers. The *temporality* assesses the obsolescence of the source: older sources are deemed less representative than newer ones. For instance, a source published within the last 3 years is considered very recent, while a source published over 9 years ago is considered highly obsolete. This obsolescence is driven

Score	**Correlation**	**Temporality**	**Reliability**
1	Not representative of the regarded variable	>10 years	Expert opinion
2	Representative of a similar variable	<10 years	Peer-reviewed expert opinion
3	Representative of the regarded variable	<6 years	Manufacturer data
4	Highly representative of the regarded variable	<3 years	Peer-review manufacturer data

Table 4.2 Assessment Criteria for the DQI of a Secondary Source

by both advancements in estimation methods and changes in manufacturing and production processes over time. Finally, the *reliability* reflects the level of confidence placed in the provenance of the source. A peer-reviewed source authored by the device manufacturer is assigned the highest reliability, while a non-peer-reviewed expert opinion has the lowest one.

In contrast to Weidema *et al.* [201], *geographical correlation* is not accounted for, as most of the ICT hardware is produced within a limited geographical area. The *completeness* parameter is also omitted, as its purpose is to account for the limitations of sampling methods, which is not relevant in the LCA of ICT devices. Indeed, LCA focuses on a given subject, and results are not expected to vary between instances of this subject.

Within this approach, DQI scores rely on 4 possible values per indicator, instead of the 5 provided by [201], and the scale is reverted—a higher DQI indicates a higher quality—so that DQI can be used as coefficients when aggregating a collection of sources. Hence, each category is assessed on a scale ranging from 1 to 4, and the overall data source DQI is computed as the sum of these individual scores. Table 4.2 maps the possible values for each category to the corresponding quality indicator. The total DQI of a single source can thus vary between 3 and 12. For instance, a source that is representative of the variable, published by the manufacturer and peer-reviewed, but published more than 10 years ago gives a total DQI of 9.

Fuzzy logic

While multiple sources should be considered to capture a more comprehensive reference impact, averaging these values can lead to errors. Indeed, extreme values and variability are not inherently incorrect and would not be captured by an average value, and thus each source should be weighted by their respective DQI as they do not have consistent quality. To address this constraint, we build on *fuzzy logic*, following the methodology introduced in [202]. In fuzzy logic, variables are not defined by a strict value in $\mathbb{R}$, but rather by a function $\mu_s : \mathbb{R} \to 0..1$ capturing the degree of membership of a value with a given fuzzy set s. A membership degree $\mu_s(v)$ of 1 indicates the certainty that a value v of x is possible, whereas a membership degree of 0 reflects that the fuzzy set does not cover this value. Given this definition, two *crisp sets* are of interest: the *core* capture the range of values

Value ($kgCO_2eq$)	**Year**	**DQI**	**Source**
27	2019	10	https://www.ademe.fr/sites/default/files/assets/documents/poids_carbone-biens-equipement-201809-rapport.pdf
33	2019	10	https://www.ademe.fr/sites/default/files/assets/documents/poids_carbone-biens-equipement-201809-rapport.pdf
39	2019	10	https://www.ademe.fr/sites/default/files/assets/documents/poids_carbone-biens-equipement-201809-rapport.pdf
49.8	2016	10	https://www.ericsson.com/en/reports-and-papers/research-papers/life-cycle-assessment-of-a-smartphone
65.25	2014	7	http://www.apple.com/environment/reports/docs/iPhone5s_product_environmental_report_sept2013.pdf
46.2	2014	7	http://www.apple.com/environment/reports/docs/iPhone5c_product_environmental_report_sept2013.pdf
83.6	2014	7	https://www.apple.com/environment/reports/docs/iPhone6_PER_Sept2014.pdf
93.5	2014	7	https://www.apple.com/environment/reports/docs/iPhone6Plus_PER_Sept2014.pdf
45.3	2016	9	https://www.apple.com/environment/pdf/products/iphone/iPhone_7_PER_sept2016.pdf
54.2	2017	9	https://www.apple.com/environment/pdf/products/iphone/iPhone_7_Plus_PER_sept2017.pdf
47.3	2017	9	https://www.apple.com/environment/pdf/products/iphone/iPhone_8_PER_sept2017.pdf
39.1	2017	9	https://www.apple.com/environment/pdf/products/iphone/iPhone_SE_PER_sept2017.pdf
64.7	2017	9	https://images.apple.com/environment/pdf/products/iphone/iPhone_X_PER_sept2017.pdf
57.9	2016	7	https://consumer.huawei.com/en/support/product-environmental-information/
49.4	2016	7	https://consumer.huawei.com/en/support/product-environmental-information/
49.3	2017	8	https://consumer.huawei.com/en/support/product-environmental-information/
66.5	2017	8	https://consumer.huawei.com/en/support/product-environmental-information/
66.5	2017	8	https://consumer.huawei.com/en/support/product-environmental-information/
65.4	2018	9	https://consumer.huawei.com/en/support/product-environmental-information/
74.9	2018	9	https://consumer.huawei.com/en/support/product-environmental-information/
60.38	2018	10	https://consumer.huawei.com/en/support/product-environmental-information/
67.02	2018	10	https://consumer.huawei.com/en/support/product-environmental-information/
40.4	2013	9	http://kth.diva-portal.org/smash/get/diva2:677729/FULLTEXT01.pdf
32.79	2020	10	https://www.fairphone.com/wp-content/uploads/2020/07/Fairphone_3_LCA.pdf
38.98	2016	9	https://www.fairphone.com/wp-content/uploads/2016/11/Fairphone_2_LCA_Final_20161122.pdf
68.7	2019	10	https://publications.jrc.ec.europa.eu/repository/bitstream/JRC116106/jrc116106_jrc_e4c_task2_smartphones_final_publ_id.pdf
88.5	2022	10	https://www.arcep.fr/uploads/tx_gspublication/etude-numerique-environnement-ademe-arcep-volet02_janv2022.pdf
82.99	2020	11	https://jyx.jyu.fi/handle/123456789/71853

Table 4.3 Secondary Sources for the Carbon Impact of Smartphones and Their Respective DQI Scores

with the highest possibility of being correct, while the *support* represents the values with a non-null membership degree.

A *fuzzy number* is a special case of a fuzzy set that is convex, normalized, and defined in $\mathbb{R}$ as a piece wise continuous membership function. As such, they act as fuzzy intervals. We use *Trapezoidal Fuzzy Numbers* (TrFN), that represents fuzzy numbers with a membership function defined as a trapezoidal shape, where the *support* is wider than the *core* and both are crisp intervals. As such, the *core* is the interval $[m_L, m_R]$, and the *support* ranges in $[L, R]$, hence resulting in the TrFN fuzzy set $< L, m_L, m_R, R >$. Then, Weckenmann *et al.* computes the TrFN for any set of sampled points with Equations 4.9–4.12, with $\bar{x}$ representing the weighted average of the sampled variable, and C_v the coefficient of variation [202]. Therefore, $\bar{x}$ and C_v account for both variations in sources regarding a variable, but also variations in quality.

$$m_L = \frac{\bar{x}}{1 + (0.5 \times C_v)} \tag{4.9}$$

$$m_R = \bar{x} \times (1 + (0.5 \times C_v)) \tag{4.10}$$

$$L = m_L - \bar{x} \times (\frac{1}{1 + (0.5 \times C_v)} - \frac{1}{1 + (2.5 \times C_v)}) \tag{4.11}$$

$$R = m_R + (\bar{x} \times 2 \times C_v) \tag{4.12}$$

To illustrate such intervals, Figure 4.4 depicts the TrFN capturing the embodied impact of a smartphone using the sources and DQI of Table 4.3. To account for quality variations in secondary sources, the main vertical axis is the DQI of each estimated impact in the aggregated secondary

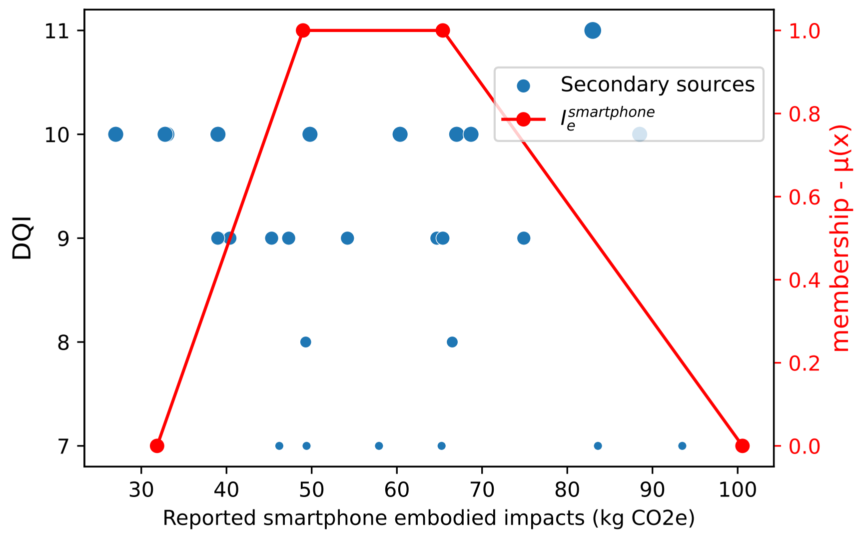

Fig. 4.4 Building the Embodied Impact Factor of a Smartphone, $I_e^{smartphone}$, as a TrFN Inferred from Secondary Sources Weighted by DQI

sources. Then, weighted secondary sources are converted to a TrFN, visible on the secondary vertical axis, with a *support* ranging from 31 to 102 $kgCO_2eq$ (*i.e.*, L to R), and a *core* between 48 and 65 $kgCO_2eq$ (*i.e.*, m_L to m_R). While various data distributions may be reported in practice, due to the lack of samples available, we assume that any variable we consider is expected to follow a normal distribution over a large enough set of secondary sources. This assumption reflects the convergence of estimation and assesses the relevance of TrFN as an appropriate structure for capturing uncertainties of estimations.

4.2.2 Modeling Hypotheses

Digital services extensively rely on devices with limited lifespans, necessitating an impact assessment that considers not only their energy consumption but also a share of their *embodied* footprint, *i.e.*,, the environmental impacts across their entire life cycle. This is particularly relevant for battery-powered devices, such as smartphones, tablets, or laptops, whose lifespans are closely tied to their usage patterns. Specifically, charging a battery reduces its capacity over time, implying that a battery can only undergo a limited number of charge cycles before its becoming unusable, mandating users to replace either the battery or the entire device. Notably, 37% of mobile phone users report that they did not attempt to repair their device following a malfunction, including battery failures [203]. In such cases, the entire device would be replaced instead of the battery, further increasing environmental impacts.

Allocating the embodied impact of ICT devices to specific software applications remains a complex challenge. Unlike energy consumption, which directly results from software usage, the depreciation of a device's embodied footprint necessitates allocation decisions to derive the direct

Label	Name	Unit	L	ml	mr	R
C_{max}	Maximum battery cycles	Cycles	328.720505	500.069021	661.158732	1005.793661
B_{cap}	Battery capacity	Amp-hour	1.727551	2.618037	3.449889	5.228173
$I_e^{bat.}$	Battery embodied impact	$kgCO_2eq$	0.723200	1.166300	1.626300	2.622600
R	Battery to device replacement ratio	%	0.089186	0.170800	0.287163	0.549948
$\overline{R}$	Avg. batteries replacements	/	1.000000	1.000000	1.000000	1.000000
C_{eff}	Charger efficiency	%	59.038467	68.603889	74.518370	86.591849
V	Battery Voltage	Volts	3.789900	3.808500	3.817900	3.836600
F_{em}	Global lectricity mix	$kgCO_2eq$	0.297775	0.557436	0.911557	1.706440

Table 4.4 Fuzzy TrFNs Model of Battery-Powered Devices Characteristics (Data provided by Greenspector)

embodied impact corresponding to the FU. This allocation is frequently time-based, distributing the embodied impact of devices across the days (or hours) of their expected lifespan. As a result, the more intensively a device is used on a daily basis, the lower its environmental impact becomes when measured on a per-hour basis. However, this approach, as seen in Subsection 4.1.1, implicitly assumes that the device is used at full capacity throughout its lifespan, leading to an underestimation of the embodied footprint per hour if the device is not consistently used at maximum capacity.

Allocating Embodied Impact Through Battery Wear

To address the issue of time-based allocation of the embodied impact of user devices, we propose a modeling approach specifically designed for battery-powered devices. Our hypothesis assumes that the more a software drains the device's battery, the higher its associated environmental impact is. Consequently, the embodied impact of the device is allocated proportionally to the total energy capacity the device can hold over its entire lifespan.

Table 4.4 introduces the TrFN variables to model battery-powered devices, along with their value provided by the company Greenspector.[6] The battery capacity B_{cap} and the maximum number of battery cycles C_{max}—the maximum complete charges that the battery can sustain while remaining usable—are used to account for the total battery capacity over its lifetime B_{total} assessed as:

$$B_{total} = C_{max} \times B_{cap} \tag{4.13}$$

The battery's embodied impact ($I_e^{bat.}$) accounts for the various environmental impacts, including its manufacturing and transportation. This impact is also incorporated into the overall device embodied impact (I_e^{device}), as defined in Subsection 4.2.1.

The primary assumption underlying this hypothesis is that the device's battery has a finite number of charge cycles, and therefore a limited lifespan. Once this lifespan is exhausted—when C_{max} is reached—the user will either replace the battery, with a probability of R, or replace the entire device, with a probability of $1 - R$. Additionally, $\overline{R}$ represents the average number of battery replacements

[6]https://greenspector.com/

a user is willing to perform in such situations, *i.e.*, how many times a user who opts to replace the battery will repeat this action.

If the battery is not replaced (with the probability $1-R$) the total device embodied footprint (I_e^{device}) is depreciated over the total quantity of energy that the battery can hold in its lifespan (B_{total}):

$$\frac{(1-R)\times I_e^{device}}{B_{total}} \tag{4.14}$$

However, when the battery is replaced (with the probability R), its own embodied impact ($I_e^{bat.}$) is fully depreciated over its own lifespan, but only a portion of the embodied impact of the remainder of the device ($I_e^{device} - I_e^{bat.}$) is depreciated. For example, if the user replaces the battery once, the device will have two batteries over its lifetime, meaning only half of the device's embodied impact is depreciated over the lifespan of each battery. The embodied impact of the battery ($I_e^{bat.}$) is depreciated based on its capacity, while the embodied impact of the remainder of the device ($I_e^{device} - I_e^{bat.}$) is depreciated over the total number of batteries it will contain in its lifespan, $\overline{R}+1$, where $\overline{R}$ represents the number of additional batteries:

$$\frac{R\times(I_e^{bat.}+\frac{I_e^{device}-I_e^{bat.}}{1+\overline{R}})}{B_{total}} \tag{4.15}$$

Finally, we can allocate the embodied footprint of battery-powered devices per unit of electric charge as the impact factor $F_e^{bat.}$:

$$F_e^{bat.} = \frac{R\times(I_e^{bat.}+\frac{I_e^{device}-I_e^{bat.}}{1+\overline{R}})+(1-R)\times I_e^{device}}{B_{total}} \tag{4.16}$$

The usage impact factor ($F_u^{bat.}$) is estimated by considering the battery's voltage (V) and the electricity mix impact (F_{em}), while also taking into account the charger's efficiency (C_{eff}):

$$F_u^{bat.} = \frac{V\times F_{em}}{C_{eff}} \tag{4.17}$$

Thus, the sum of $F_e^{bat.}$ and $F_u^{bat.}$ provides the total impact factor per unit of energy F^{device}, which represents the environmental impact per unit of energy discharge (in amp-hours) caused by a digital service on a battery-powered device:

$$F^{device} = F_{e(d)}^{bat.} + F_{u(d)}^{bat.} \tag{4.18}$$

4.2.3 Identifying and Mitigating Uncertainty

This allocation approach enables a more representative distribution of the embodied footprint of users' devices compared to traditional time-based methods. However, the inclusion of various variables and modeling choices introduces inherent uncertainties into the analysis.

Fuzzy logic provides a systematic approach for managing these uncertainties by enabling arithmetic operations between fuzzy sets, such as addition, subtraction, multiplication, and division. For instance, the sum of two fuzzy sets, represented as $[a_1,a_2,a_3,a_4]$ and $[b_1,b_2,b_3,b_4]$, is calculated as $[a_1+b_1,a_2+b_2,a_3+b_3,a_4+b_4]$. Similarly, multiplication apply element-wise between sets. In the case of division and substraction, the divisor and subtrahend set is inverted, yielding the operation $[a_1/b_4,a_2/b_3,a_3/b_2,a_4/b_1]$ $[a_1-b_4,a_2-b_3,a_3-b_2,a_4-b_1]$and [204]. In our context, both the multiplier and the divisor are positive values. Any real number $x \in \mathbb{R}$ can be converted into a fuzzy set in the form $[x,x,x,x]$, allowing for the combination of real numbers and fuzzy sets in mathematical operations. Consequently, the result of an equation containing fuzzy sets will also be a fuzzy set [205].

Since fuzzy sets encapsulate both the value and the uncertainty of any variable, arithmetic operations inherently propagate uncertainties throughout each stage of the modeling process. Therefore, by using fuzzy logic in environmental assessments, uncertainties can be systematically calculated and propagated without the need to define specific sensitivity scenarios, such as best- or worst-case scenarios. This ensures that the results reflect the full spectrum of potential estimates and their associated uncertainties.

Relative contribution in arithmetic operations

In this approach, we apply the *core* methodology [206] for the defuzzification of TrFNs, and obtain both an absolute value and the corresponding uncertainty. The core methodology defines the average of m_L and m_R as the central value, expressed as $TrFN_{central} = \frac{m_L+m_R}{2}$. The symmetric margin of uncertainty is then calculated as the difference between the central value and either m_L or m_R. Accordingly, the uncertainty of any fuzzy set is determined by the following equation:

$$TrFN_{uncertainty} = TrFN.m_R - TrFN_{central} \tag{4.19}$$

For example, the F_{em} variable defined in Table 4.4 has a value of 0.73 ± 0.18, where 0.73 is the central value and 0.18 represents the uncertainty margin.

However, while a single variable may have a relatively large uncertainty, it may not significantly impact the overall uncertainty in the final result. On the contrary, a variable with a relatively small uncertainty can largely impact the total uncertainty.

To uncover the uncertainties introduced by modeling choices, it is essential to study how they propagate through each calculation step and determine the contribution of each variable to the total uncertainty. Specifically, in any arithmetic operation involving variables A and B that yields a result C, it implies to quantify how much of the uncertainty originates from A and how much from B. To achieve this, we use the arithmetic operations of Fuzzy Logic. In the case of addition, i.e., $A+B=C$, the absolute uncertainties of each TFN are simply added:

$$C^{add}_{uncertainty} = A_{uncertainty} + B_{uncertainty}$$

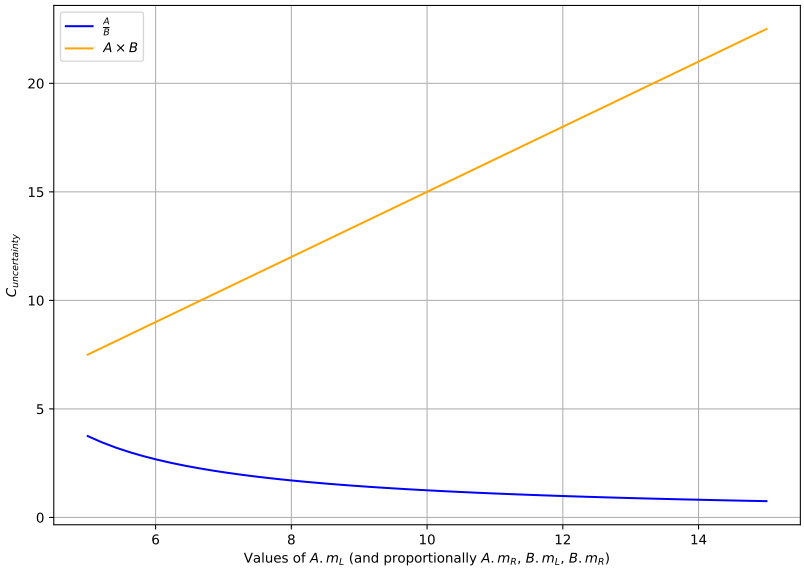

Fig. 4.5 Comparison of Uncertainties Resulting from the Multiplication and Division of TrFNs A and B

Similarly, for subtraction, the absolute uncertainties are subtracted:

$$C^{sub}_{uncertainty} = A_{uncertainty} - B_{uncertainty}$$

In Figure 4.5, we vary m_L and m_R for both A and B proportionally, and assess the resulting uncertainty of C for both operations. It can be observed that multiplication behaves in a *linear* manner, meaning that the uncertainty of C can be derived by summing the uncertainties contributed by A and B:

$$A^{mult}_{share} + B^{mult}_{share} = C^{mult}_{uncertainty}$$

To compute the uncertainty each component of the TrFN is multiplied, such that:

$$C^{mult} = [A_1 \times B_1, A_2 \times B_2, A_3 \times B_3, A_4 \times B_4]$$

As such, the uncertainty of C for multiplication is expressed as:

$$C^{mult}_{uncertainty} = (A.mR \times B.mR) - \frac{(A.mL \times B.mL) + (A.mR \times B.mR)}{2}$$

To determine the contribution of A to C, we fix B at its central value, i.e.,

$$B_{fixed} = [TrFN_{central}, TrFN_{central}, TrFN_{central}, TrFN_{central}]$$

and calculate:

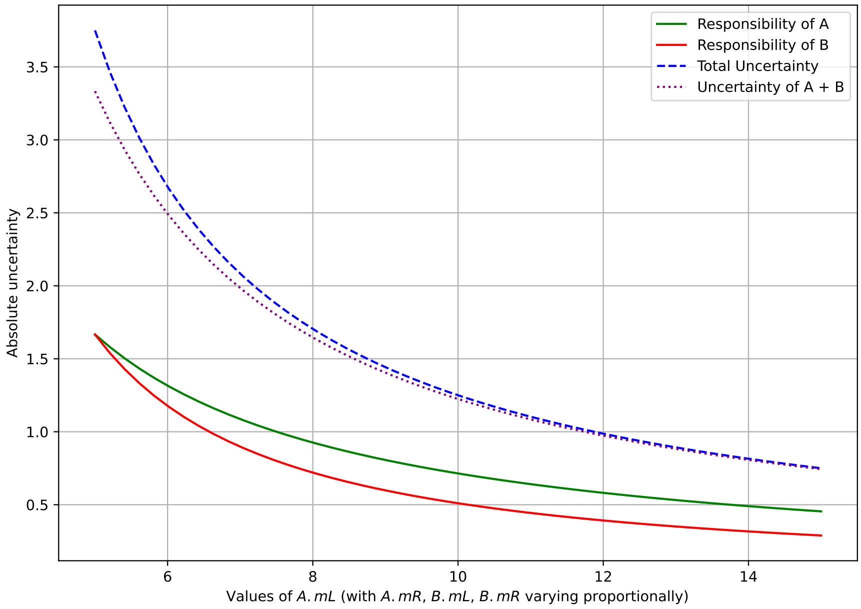

Fig. 4.6 Comparison of Division-Induced Uncertainty in TrFNs A and B Against Total Uncertainty in C

$$A_{share}^{mult} = (A.mR \times B_{fixed}) - \frac{(A.mL \times B_{fixed}) + (A.mR \times B_{fixed})}{2} \tag{4.20}$$

Similarly, to assess the contribution of B to the uncertainty of C, we hold A fixed at its central value. This approach allows us to isolate the uncertainty contributions from both A and B in the multiplication operation.

For the division, Figure 4.5 illustrates that the operation is not linear, meaning the uncertainty of C is not directly proportional to the uncertainties of A and B. This non-linearity means that the uncertainty in C cannot be calculated by simply summing the uncertainties from A and B. For example, when A is divided by B, small variations in B can significantly impact the result, particularly when the lower bound of B approaches zero.

As shown in Figure 4.6, this non-linearity leads to a more complex propagation of uncertainty in the division of TFNs. While the cumulative sum of A and B—assessed by fixing either A or B as detailed in the multiplication case—does not equal the total uncertainty of C, it is consistently less. The gap between the cumulative uncertainty and the total uncertainty narrows as the values decrease. Given this observation, we choose to treat the sum of uncertainties from A and B as a valid approximation for our assessments, acknowledging a slight loss of precision due to the unaccounted difference in the sum.

Uncertainty tree

By evaluating the relative contribution of all variables in Equation 4.18, it becomes possible to systematically trace the sources of uncertainty in the final result. To achieve this, we construct the uncertainty tree depicted in Figure 4.7, where each node represents a calculation step and each leaf corresponds to a variable from Table 4.4. The calculations proceed from the bottom of the tree upwards, starting by computing absolute values.

For example, if $A = [1,2,3,4]$ and $B = [5,6,10,12]$, then:

$$C = A \times B = [1 \times 5, 2 \times 6, 3 \times 10, 4 \times 12] = [5,12,30,48]$$

Following Equation 4.19, the uncertainty of C is calculated as:

$$C_{uncertainty} = C.m_R - C_{central} = 30 - \frac{12+30}{2} = 9$$

To determine the share of uncertainty contributed by A in this multiplication, we use Equation 4.20:

$$A^{mult}_{share} = (3 \times 8) - \frac{(2 \times 8) + (3 \times 8)}{2} = 4$$

Where $B_{fixed} = \frac{6+10}{2} = 8$.

Similarly, for B we obtain: $B^{mult}_{share} = 5$.

To assess the relative contribution of A (and similarly for B), we calculate the percentage:

$$A_{percentage} = \frac{A_{share}}{C_{uncertainty}} \times 100$$

By applying this methodology to each step in the modeling process, we can construct the full uncertainty tree, tracking the relative contributions of each variable and ultimately determining the total uncertainty for F^{device} from Equation 4.18.

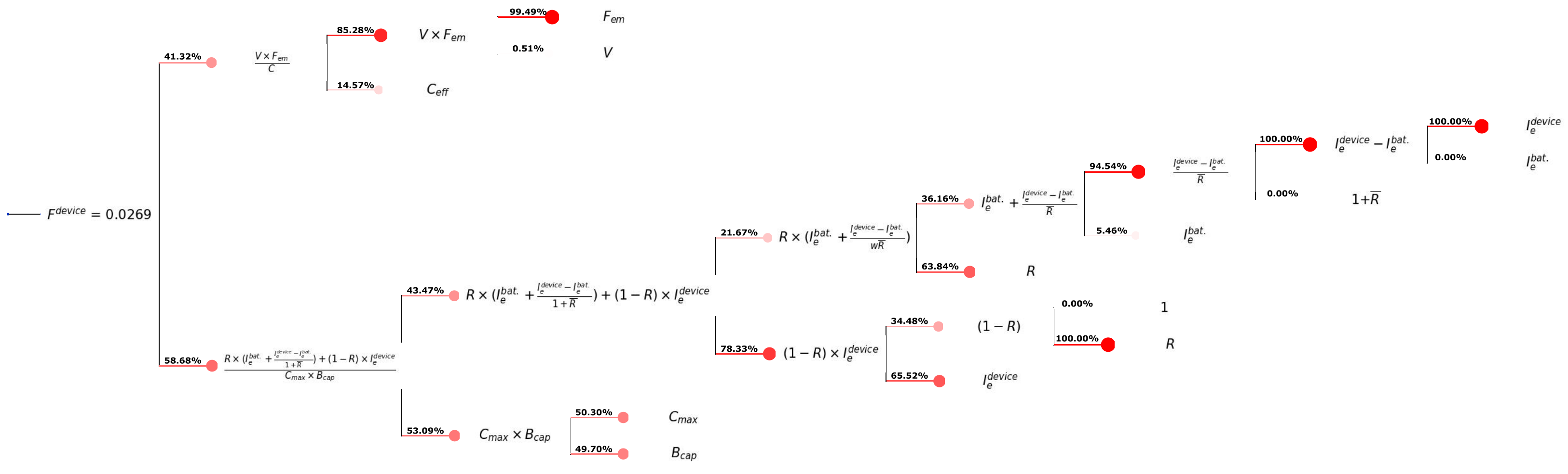

Fig. 4.7 Tree Diagram of Relative Uncertainty for F^{device}

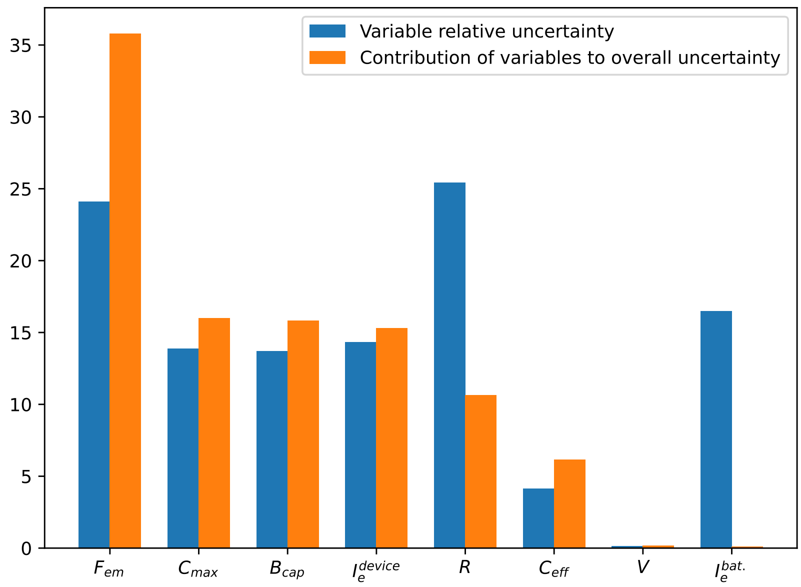

Fig. 4.8 Comparison of Individual Variable Uncertainty and Its Contribution to Total F^{device} Uncertainty

The uncertainty tree alone is not sufficient to precisely identify the sources of uncertainty, which ultimately reside in the individual variables. However, by performing a tree traversal we can generate the plot shown in Figure 4.8, that compares the uncertainty of each variable to the uncertainty it induces in the final calculation of F^{device}. This type of analysis allows to pinpoint the contribution of each variable to the overall uncertainty, even in cases where the variable itself has relatively low uncertainty. For instance, the variable $I_e^{bat.}$ may have an individual uncertainty of 15%, yet it contributes less than 1% to the overall modeling uncertainty. In contrast, F_{em} contributes the most to the overall uncertainty, exceeding the relative uncertainty it holds individually. This highlights the critical role of certain variables in driving overall uncertainty, even if their own uncertainty is initially not the highest.

The analysis of the primary contributors to uncertainty enables the identification of variables that should be more precisely specified, allowing experts to significantly reduce the overall uncertainty in the results. Notably, fuzzy sets can be replaced with fixed values derived from physical measurements or additional information about the specific devices used by the software studied. This systematic analysis and uncertainty propagation offer a significant advantage over traditional Monte Carlo simulations commonly used in LCA, which typically require the definition of discrete scenarios to identify such drivers of uncertainty.

4.2.4 Limits and Future Work

It is generally assumed that the lifespan of battery-powered devices is determined solely by the lifespan of their battery, with the assumption that users replace their devices once the battery becomes unusable.

However, this hypothesis may overlook other factors influencing the decision to replace these devices. In particular, it does not account for hardware and software obsolescence. Battery-powered devices, such as older smartphones, may be replaced due to performance issues, such as slowness in running recent applications, outdated operating systems that are no longer compatible with newer applications, or the availability of more advanced models on the market.

A limitation of the approach we applied lies in the defuzzification process, specifically the *core* method, which considers only the *core* interval and, thus, only two points out of four of the TrFN. In future work, alternative defuzzification approaches, such as the Expected Interval or the Central Interval, should be investigated to more effectively capture the variability of the data within the modeling process and its resulting outputs. Additionally, from a scientific perspective, the Life Cycle Assessment (LCA) method lacks empirical validation regarding its overall results [207]. Therefore, even though using fuzzy logic in LCA for ICT services provides a systematic approach to evaluate and propagate uncertainties, the resulting outcomes still lack empirical validation.

4.3 Assessing the Environmental Footprint of Software Life Cycle

In the previous sections, we followed the established three-tier architecture to assess the environmental footprint of digital services, focusing on two of these tiers: backend infrastructure and user devices. Although the production footprint of software is mentioned in the ITU L.1410 [102] standard, it is almost entirely overlooked in real-world analyses. Most assessments focus solely on the usage phase of software, omitting the rest of its lifecycle. In this section, we propose a holistic approach that delivers actionable insights to identify potential shifting between SDLC phases, such as development and usage, but also hotspots among the various resources consumed to produce and operate software services. The objective is to address the following research question: *How does a holistic perspective provide actionable insights to reduce software environmental footprint?* To achieve this, we have developed a model and an associated tool designed to facilitate the estimation of software's environmental impact, throughout its life cycle across various categories. As recommendations to lower software project impacts cannot be generalized due to their uniqueness, we propose a simple and flexible way to model and adapt to different development models and SDLC.

4.3.1 Approach

When designing this methodology, our goals are threefold. It needs to be: 1. *holistic* by providing a comprehensive perception of the impacts resulting from a software project across its complete lifecycle, 2. *actionable* by delivering actionable insights to project stakeholders, 3. *easy to use* to enforce a widespread adoption.

To meet our first objective, we elected to follow the core principles of an attributional screening LCA, as introduced in Subsection 2.2.2. The perimeter we take into account in our methodology includes all the resources consumed when performing the activities identified in the SDLC, but also other activities to consider a complete *cradle-to-grave* view (cf. Subsection 2.2.2) of the impacts arising from the ICT project under study. This includes for example (and not exhaustively) the distances traveled to sell the product, the man-days required for its development, the production and deployment of ICT equipment and the impact of end-user devices. This offers a high-level view of the impacts and supports identifying key causes within the processes. Additionally, such a holistic perspective allows the identification of potential burden shifts between life cycle phases or resources.

For a software product, such a shift can occur when reducing the hardware resources required, and thus the usage phase impact, by adopting a more efficient programming language. However, this programming language might incur longer development time, which will increase the impacts occurring during the building phase. Impact shifts can also occur between categories, for instance when replacing servers with more energy-efficient ones, thus decreasing the climate change impact, while increasing the land pollution and resource depletion due to the manufacturing of new hardware.

Similarly to Subsection 4.1, the adoption of a *bottom-up* approach allows us to meet our second objective: delivering actionable insights. Once stakeholders have identified major impact sources in their project, they can investigate the effect of decisions taken at a micro-scale on a larger scale. Our approach therefore equips stakeholders with a new *Key Performance Indicator* (KPI) to consider the environmental impacts of their decisions and assess if the benefit outweighs the cost next to others, such as security level, financial costs, redundancy, high availability…

Meeting our third objective, namely *"ease of use"*, is generally difficult when using a bottom-up approach with a large scope. We answer this challenge by proposing a simple and flexible way of modeling a project's resources and activities, which allows using the same methodology for projects using different development models. Our estimates rely on an impact dataset (named *secondary data* in LCA vocabulary), that we have consolidated from existing openly available sources. Finally, we elect to simplify inventory and data collection by using a static view of the project, which we introduce in the following section.

Software life cycle modeling

To adapt to the context of each stakeholder, software life cycle modeling must be adaptable to all types of SDLC models, while keeping a consistent environmental impact computation methodology and reference sources, and thus ensuring a standardized means of comparison across different models.

To do so, it is represented as a static view rather than a time-based one, such as a timeline, to lower the granularity of inputs required while keeping their relevance. For instance, it is not required for stakeholders to detail on each day how many developers worked on the project, the number of test servers used, which would be tedious without an automation, but rather the number of *man-days* and *server-hours* consumed on a given period. Instead of a *day-to-day* basis, time is flattened and the life cycle is modeled as a tree, as depicted in Figure 4.9. In this example, the project contains two key phases, *build* and *run*, which can contain sub-phases, such as *selling*, *development*, and *hosting*. Stakeholders are able to add, move, and remove phases and sub-phases to model their SDLC,

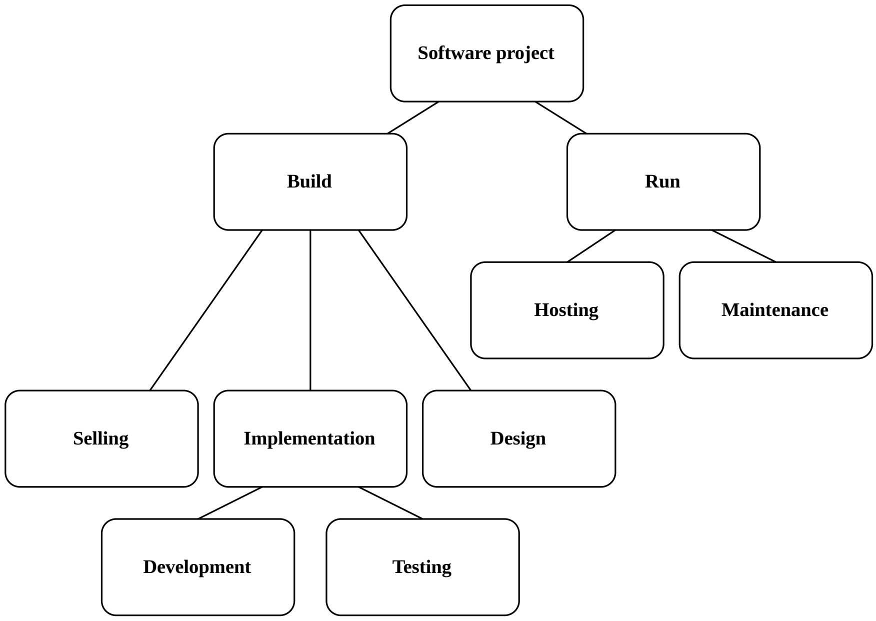

Fig. 4.9 Illustrative Example of a Software Life Cycle Tree

no matter the model and processes or phases they choose. To assist in this process, we propose predefined trees that model the most common life cycles, which they can fully customize to model more accurately their daily activities.

Each phase consumes resources to be completed, which can be of multiple kinds. The phase *coding* consumes *man-days*, but also *server-hours* for developers to test their products. As software projects use common resources, we propose a default dataset with their associated environmental impact, which the model then uses to compute the whole software life cycle impact. This way, we allow stakeholders to set their resources usage accordingly to their needs and policies, in the right life cycle phases, while keeping the same computation methodology between multiple typologies of projects.

Environmental impact computation

As the project life cycle is flattened as a tree representation, its whole environmental impact, represented as an aggregate of impact categories, can be computed using a tree traversal:

$$\mathscr{I}_{project} = \sum_{phase \in phases} \mathscr{I}_{phase} \tag{4.21}$$

Each phase of the project—or node of the tree—can consume resources and require sub-phases. For example, the phase *build* requires the sub-phase *development*, which consumes *man-days* resources. As such, the environmental impact of a phase is the sum of its consumed resources, including the ones consumed by its sub-phases:

$$\mathscr{I}_{phase} = \sum_{res \in R} \mathscr{I}_{res} + \sum_{s \in subphases} \mathscr{I}_s \tag{4.22}$$

The impact of a given phase mostly depends on the resources it consumes. Each one is estimated from the declared quantity weighted to its associated *impact factor* $\mathscr{F}$, which gives an environmental impact per functional unit consumed, such as a *vCPU-hour*, a *plane-kilometer*... A resource impact $\mathscr{I}_{res}$ is therefore computed as:

$$\mathscr{I}_{res} = Q_{res} \times \mathscr{F}_{res} \tag{4.23}$$

The quantity must be expressed in the same unit as the impact factor in order to obtain environmental impact. However, as the model uses a flattened time representation, this quantity cannot directly express recurring events such as a product owner going to meet the customer every month, by plane, for 6 months. The quantity is therefore aggregated from multiple fields: *amount*, the number of functional units used, *period*, the length of the resource consumption, and *frequency*. For instance, for the impact factor *plane*, expressed by kilometer per passenger, the amount will be *100 plane-kilometers.passenger*, the frequency *every month*, and the period *a year*, giving a quantity equal to $12,000$ *plane-kilometers.passenger*, which corresponds to the impact factor functional unit.

If an impact factor is given by a time-based functional unit, such as vCPU-per-hour, the quantity field has to contain a time unit. A *period* of usage is therefore mandatory and the *frequency* field cannot be used in isolation anymore, to avoid canceling out the time unit. The field *duration* is added to represent a recurring scenario: to model a vCPU used two hours per day for a year, the duration will be *a year*, the frequency *every day*, and the duration *two hours*. The quantity value is thus computed as follows:

$$Q = \frac{amount \times duration \times period}{frequency} \tag{4.24}$$

Environmental Model

To foster widespread adoption of our approach to assess software environmental impact, we developed a web application shown in Figure 4.10 as well as an API implementing our approach, available on Github.[7] This application provides software project stakeholders with an easy way to model the life cycle of their project and get a first overview of the associated environmental impacts to quickly identify hotspots in their projects. They can model and save their project characteristics, through phases and resources consumed, and analyze the resulting impact from various standpoints.

Tree view Shown at the center of Figure 4.10, the tree view defines the project life cycle, by modeling its phases as nodes. Each node can be moved or deleted, and new ones can be created at all levels. Stakeholders can adopt a life cycle granularity that suits their study scope, which they

[7] https://github.com/Orange-OpenSource/SoftwareLifecycleEnvImpact

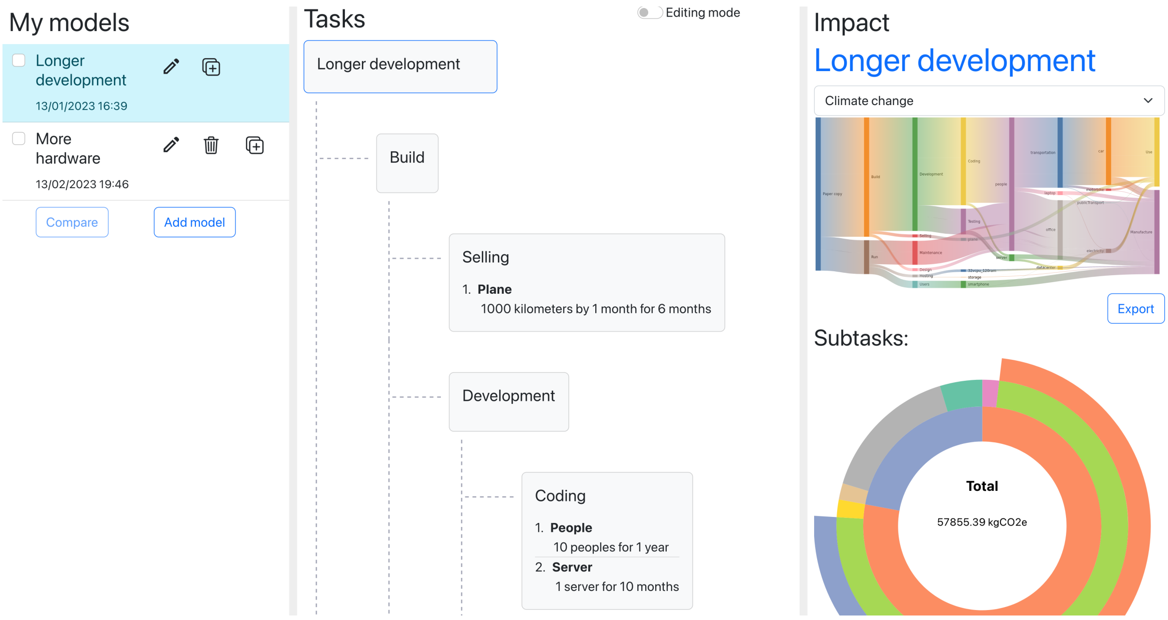

Fig. 4.10 Screenshot of the Web Application Implementing Our Approach

can refine iteratively by updating the tree. Standard predefined life cycle trees are proposed to start modeling from the most common SDLC models.

Resources inputs By switching to editing mode, the resources consumed by each phase can be updated. A form assists in conforming with the associated impact factor unit, while allowing entering time-based events despite the overall static view.

Impact factors The impact factors are exposed by the API, and cannot be updated when modeling projects. Keeping the reference values in a single place, rather than distributing them, as spreadsheets for instance, avoids data tempering that would make comparisons between projects or solutions impossible. It also allows updating at once all the projects modeled using these impact factors, when refining their values.

Results To identify impact hotspots and shifts, the resulting environmental impact for the chosen impact category is displayed on the left side of the application, shown in Figure 4.10. The top figure is the Sankey diagram of the impact combined between the phases and resources, which is then refined solely by phases, as shown in Figure 4.11, and by resource, as shown in Figure 4.12. For the impact by resource, all impact categories are also reported simultaneously, as depicted in Figure 4.14.

Comparison Each project can contain multiple models, which can be used to compare different approaches and scenarios. Shown on the left-hand side of Figure 4.10, these models can be compared

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	368.4	1263.0	3332.9	736.7
Environment/CM	263.1	842.0	1666.5	263.1
Requirements	999.9	1894.5	2666.3	210.5
Design	499.9	3789.0	5332.7	210.5
Implementation	210.5	1368.2	11331.9	999.9
Assessment	210.5	1052.5	7999.0	1263.0
Deployment	78.9	315.7	999.9	1578.7

Table 4.5 Software Effort Distribution of GitLab Using COCOMO II (Person-Months)

between them, using a side-by-side view of the environmental impact described previously, as well as a direct comparison by resources shown in Figure 4.13.

Physical quantities Each physical quantity, such as ones expressed in CO_2e, is represented as a numerical value and a unit of measurement. Our implementation uses the Pint library [208], which allows conversions between units as well as ensures their consistency in computations.

4.3.2 Findings

This section demonstrates the benefits of our approach. We use the Gitlab open-source project to estimate a sampled application life cycle, considering the source code repository for modeling the building phase and the documented hardware requirements for the usage phase.

Using the CLOC tool [209], we extracted $6,241,291$ *Source Lines of Code* (SLOC) from the project repository [210]. We used this value as an input for a *Constructive Cost Model II* (COCOMO II), the most commonly-used algorithmic method for software development cost estimation defined by Bohem *et al.* [211]. Using a *post mortem* approach on an already developed application offers an estimation of the effort required to build a software, as well as its distribution among phases. To obtain a standard software effort distribution, we leave COCOMO cost drivers, such as team cohesion, programmer's capability, and multi-site developments with their default values. We obtained the effort distribution shown in Table 4.5.

As discussing the COCOMO potential limits is out of the scope of this analysis, we consider the effort distribution as representative of a project this size, rather than the actual effort accomplished to design and develop the Gitlab application.

To model the usage phase, we adopted the Gitlab hosting reference architecture for $50,000$ users [212]. We chose the largest one, considering that the application is designed to be as scalable as its biggest instantiation, requiring more effort than for a smaller one, such as for 500 users. Hosting an instance requires hosting services not developed in-house by Gitlab, such as Sidekiq or PostgreSQL, which is realistic for a real-world application. We only consider these services hosting impact and not their development, as done by Kern *et al.* [123]. As the requirements do not specify a storage size, we used the default repository size limit of 10 GB by project, and considered 2 projects per user with a $3\times$ data redundancy; thus obtaining a total of $3,000$ TB of data to store. We do not include end-users in our case study, considering that their usage environmental impact—*i.e.*, mainly through energy consumption—would be marginal in contrast to the manufacturing impact of their device, and that the property of the device does not rest on using Gitlab or not.

In this case study, we assume the development and hosting in France, and we use the corresponding impact factors.

Shifts between phases

As demonstrated in Section 4.2, the hypotheses taken when modeling and reference values used with possibly high margins of error imply uncertainty in an environmental impact assessment, to which there is no ground truth to compare. This implies caution when treating with absolute values, but stakeholders can identify impact hotspots and shifts using a high-level perspective, by manipulating orders of magnitude obtained through a common methodology and reference values.

We first focus on a specific impact category, *Climate Change* indicated in CO_2e, to show how modeling a software life cycle environmental impact can provide actionable insights to lower its environmental impact.

Phases When applying the methodology described in Subsection 4.3.1 with a *run* phase duration of 15 years, we obtained the distribution of tCO_2e emitted shown in Figure 4.11. The impact is largely dominated by the building phase, emitting approximately 14 tCO_2eq , while the running phase is relatively low with roughly 279 tCO_2eq emitted.

This highlights the importance of adopting a life cycle approach, as covering only the software through its usage, which is commonly done, can hide the vast majority of impact which occur during its development: more than 95% in this case.

The *build* phase and its sub-phases only consume *people* resources. As such, their environmental impact distribution is the same one as the COCOMO software effort 4.5, where the *construction* activity dominates the man-days required, resulting in more than 64% of the *build* phase CO_2e emissions, especially for *implementation*, *assessment*, and *design* phases.

Amortization Figure 4.11 highlights that in an ad-hoc development, the *build* phase can largely prevail over software's overall life cycle environmental impact. However, a project the size of Gitlab requires to be hosted for an unrealistically long duration to reach the tipping point where its *run* phase impact will be equal to the *build* phase. With a constant *build* phase emitting $13,945$ tCO_2eq and a year of *run* emitting 18.614 tCO_2eq , this tipping point is thus identified as roughly 749 years of usage. As Gitlab is hosted simultaneously across multiple instances, we consider a *hosting-year* as a single instance hosted for a year. This implies, for instance, that 200 GitLab instances operating over a span of four years would be sufficient to reach this tipping point, a scenario that is certainly attainable for a successful project.

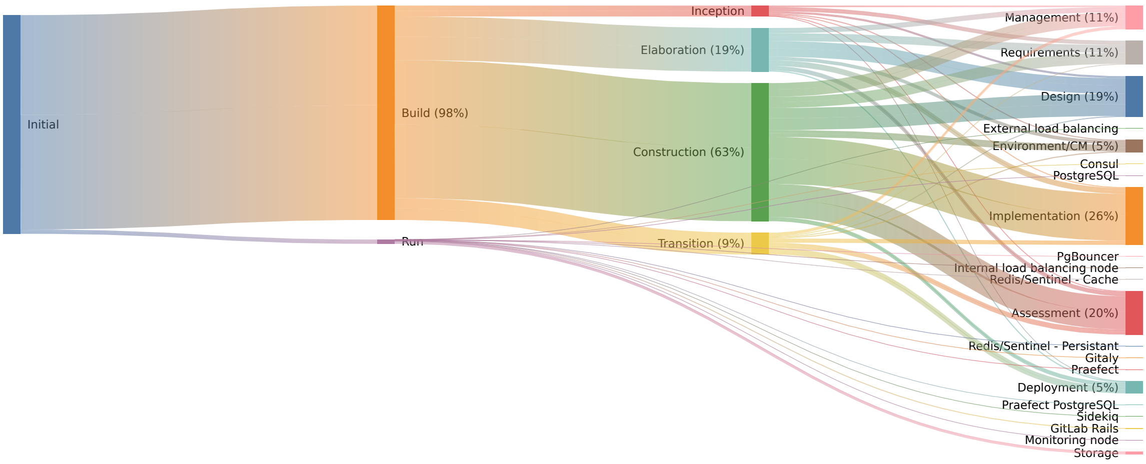

Fig. 4.11 Distribution of $kgCO_2eq$ Among Life Cycle Phases

Shifts between resources consumed

A life cycle approach allows stakeholders to identify the most impactful phases and uncover impacts hidden by focusing on a single one, but does not provide actionable insight to reduce their respective impact. As defined in Subsection 4.3.1 a phase does not have a direct impact, it inherently comes from the resources it consumes. By using a similar holistic approach for resources, stakeholders can identify hotspots and shifts in their environmental impact.

Hotspots

The Sankey diagram of Figure 4.12 depicts the flows of tCO_2e emitted by the consumption of the different resources across the project life cycle. It models a project where impacts are at the tipping point between the *build* and *run* phases, previously defined and established at 749 *hosting-years*. This provides a more realistic distribution of impacts sources and avoid having the *people* resource impact largely dominating the overall one.

For one day, the *people* resource emits roughly 1.21e-2 tCO_2eq and is constituted of transportation, a laptop and an external monitor, as well as an office. While the monitor and laptop impacts are relatively low, the office accounts for almost half of the impact, emitting $6,178$ tCO_2eq . Transportation emits the other half, $7,326$ tCO_2eq , based on a representative usage mix of car, motorbike, public transport and bike usage in France [213, 214] and their associated impact factors [215]. Among these, the car accounts for more than 95% of the overall commuting environmental impact and ultimately becomes one of the main impact hotspots among all resources consumed throughout the software life cycle. We thus concur with Kern *et al.* [123] that commuting can be a key impact on the overall environmental impact of a software life cycle, not only on the building phase.

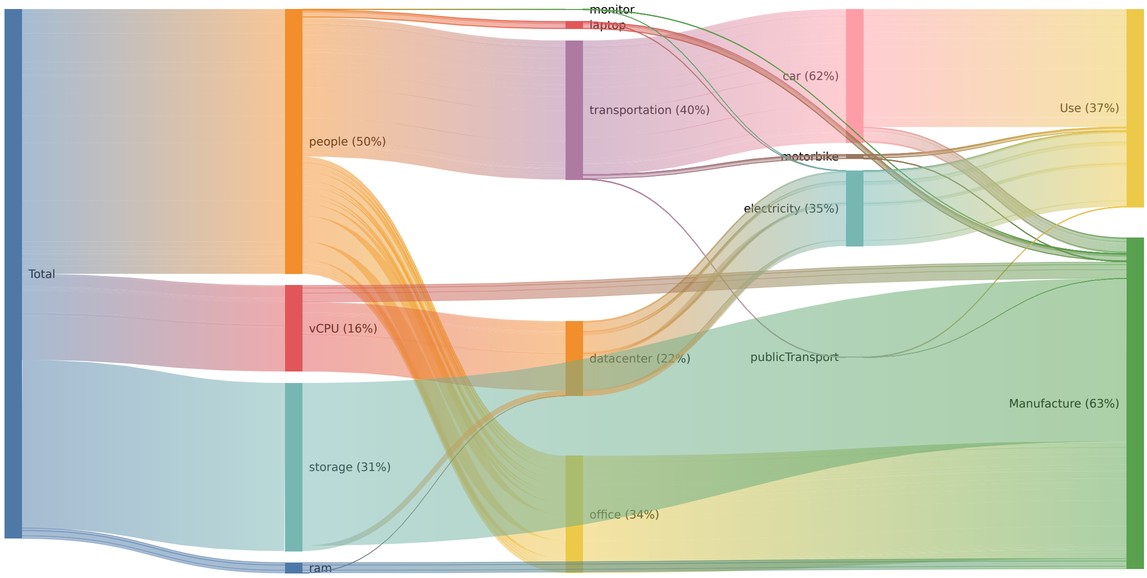

Fig. 4.12 Distribution of $kgCO_2eq$ Among Resources

Manufacture and usage

Figure 4.12 shows that, despite the 758 vCPU and 1636 GB of memory required to host Gitlab instances, the 300 TB of storage largely prevails over the overall CO_2e emissions. Among the 8,853 tCO_2eq emitted, more than 95% comes from its manufacturing. By acknowledging only its usage impact—*i.e.*, energy consumption—it would be considered a relatively low source of impact. However, when accounting for the hardware's complete life cycle, its manufacturing represents one of the primary sources of emissions on the overall project, further underscoring the importance of embodied emissions, as discussed in Subsection 4.1. Additionally, storage emerges as a significant hotspot in the overall project life cycle. Identifying such hotspot allows stakeholders to uncover hidden impacts, as well as prioritize actions, such as prolonging hardware life expectancy or decreasing the number of allocated resources.

Compare scenarios

To compare different approaches, Figure 4.13 depicts a hypothetical scenario contrasting a model at the tipping point between its *build* and *run* phase impacts, and a more hardware-intensive scenario. In the latter case, the overall number of *person-months* required is reduced by 20%, resulting in less efficient software that requires double the hardware resources to host an instance.

Figure 4.13 points out the main drivers of emissions shifts between both approaches. The decrease of *person-month* required results in cars and offices emissions lowering by almost 2,650 tCO_2eq . It is however not sufficient to compensate for the increase of hardware-related ones, resulting in an overall emissions increase of 2299 tCO_2eq . By modeling and comparing different scenarios, stakeholders can quickly get insights into the relative impact shifts between different implementation choices.

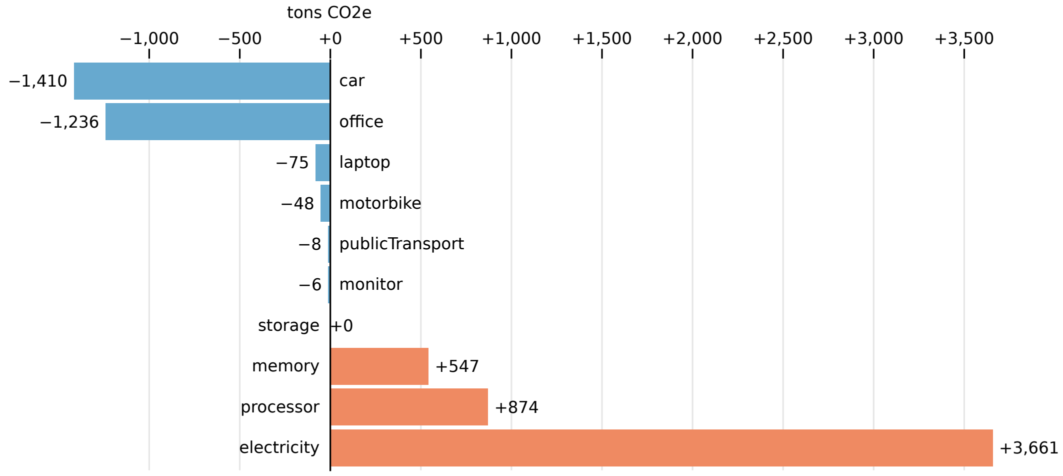

Fig. 4.13 Consequences of a 20% Reduction in Required Effort

Multicriteria

We focused on the climate change impact category to highlight how the methodology can help stakeholders to identify impact shifts between phases and hotspots between resources, and highlight the importance of considering their life cycle impacts. However, an environmental impact is an aggregate of impact categories, and an impact can also shift between these categories.

To identify this shift, Figure 4.14 shows a holistic view of resources' environmental impact along all categories of impact proposed in the tool. Due to the scarcity of open-source data, we were not able to find relevant multi-categories impact factors for transport and buildings. Thus, the figure focuses on the software *use* phase, which uses hardware resources with environmental impact data sourced from [216].

Following the category of impact looked at, the order of magnitude between the main sources can vary greatly. When focusing on *climate change*, a decision such as replacing processors with newer and more energy-efficient ones can reduce their contribution to this impact category. It will however increase their contribution to other categories, such as *water depletion* and *resource depletion* with a possibly worse environmental impact. Furthermore, a holistic view of multiple categories further highlights the major role of hardware manufacturing on the software hosting environmental footprint.

Figure 4.14 displays each category separately and in its respective unit, which can complicate the decision-making process. To better understand their relative magnitude and importance, the ISO 14044 [56] defines optional steps of an LCA: *normalization* and *weighting*. Each indicator result is transformed by dividing it by a reference value, such as the one given by Sala *et al.* [217]. This allows for identifying and prioritizing the relevant categories, as well as obtaining a single score by conducting another optional step, *grouping*. However, as there is no broadly accepted method for weighting and grouping yet [218], we chose not to aggregate impact categories and to report them separately to stakeholders.

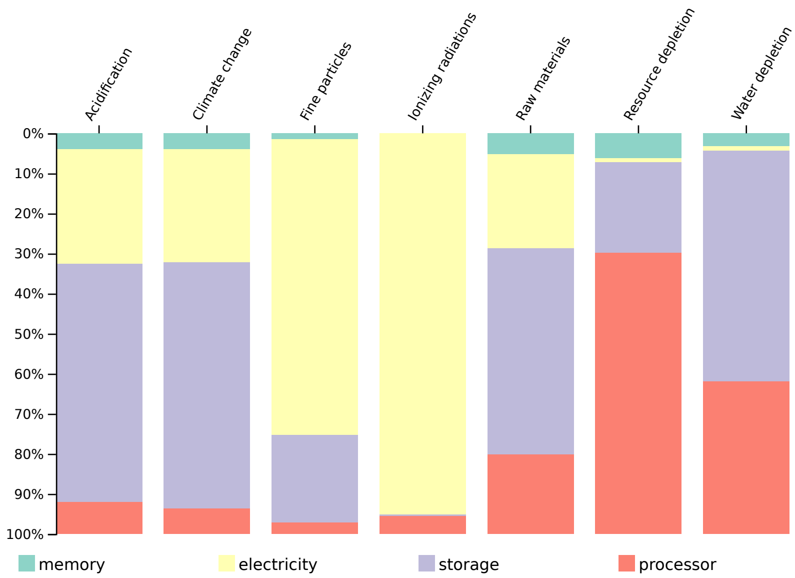

Fig. 4.14 Environmental Impact of Resources in the *Run* Phase

4.3.3 Limits and Future Work

Gitlab is used as a sampled case study to show the methodology and the actionable insights it offers on a realistic dataset of a complex application, but the impact cannot be considered as these of the project itself. Defining a scope, even if it causes truncation error (cf. Subsection 2.2.2), is essential to a life cycle approach, as considering all upstream resources would be too large and thus impossible. As we limit our *build* scope to values obtained through the COCOMO effort estimation methodology, some key elements are missing for an organization, such as supporting employee infrastructure, accountancy, human resources, etc. Furthermore, the Gitlab company is *all-remote*, thus not using offices each day, which plays a key role associated with commuting in our impacts modeling. Their employees work across 65 countries, which we do not consider as we did not add work-related travel. We used a *"black-box"* approach using the source code repository and documented hardware requirements as a best-effort solution, but encourage stakeholders to use the more precise project values available to them, and impact factors better suited to their context to obtain the best insights from the model.

We chose not to consider *Continuous Integration / Continuous Delivery* (CI/CD) runners in our use case, as they would be predominant over the hosting impact. While relevant for the Gitlab product LCA, it is a peculiar use case, highly resource-intensive thus with a high impact. Using the default SaaS runner configuration with 1 vCPU and 3.75 GB of RAM, as well as the 10,000 minutes per month or shared runner time allocated to premium users, we obtain just about 12.77 tCO_2eq per month for the runners, while the overall hosting impact is roughly 1.55 tCO_2eq .

As it uses an attributional approach, our model will be limited to delivering insights into its intrinsic characteristics, but not into the side effects that might occur. An ICT product life cycle analysis will only reveal the first-order effects, not its role as an enabling technology (cf. Subsection 2.2.2).

For instance, reducing the commuting impact by encouraging employees to work from home will reduce the project's overall impact from the model's perspective, but teleworking may also increase the overall distance traveled as stated by Caldarola and Sorrell [219]. Offices-related impacts such as energy consumption for heating, cooling and lighting will be transferred to employees' homes. Hook *et al.* [220] concluded through a systematic review that economy-wide energy savings due to teleworking are typically modest, and can even be negative or nonexistent.

We highlight the importance of considering software's environmental impact over its complete life cycle, and not only its usage through energy consumption. However, the ecosystem lacks data on the environmental impact of resources it uses, as highlighted in Subsection 2.2.2, all the more multi-criteria impact, which we demonstrate the importance of throughought this chapter. Our model outputs are strongly correlated to its reference values, and we modeled an environmental impact using impact factors from France where the electricity is low carbon. In this context, we observed a smaller resource usage than manufacturing footprint, but this cannot be generalized and we encourage stakeholders to use impact factors corresponding to their context.

Finally, we state in Subsection 2.3.3 that in the literature second- and third-order effects, notably rebound effects, are influenced by factors beyond the control of developers at the time of software conception. In contrast, this approach adopts a more holistic view of software, extending beyond the development phase. At the development level, however, developers could engage with issues such as software obsolescence, incorporating more holistic considerations that extend beyond immediate technical concerns. In the following chapter, we examine the potential strategies for developers to mitigate the environmental footprint of software and consider the broader responsibilities of software stakeholders in promoting ecodesign principles, notably in addressing impacts beyond first-order effects.

Evaluating software environmental footprint: Approaches to assessing the environmental footprint of software primarily focus on energy consumption and adopt three tiers architecture: user devices, data centers, and networks. However, reliable and openly available reference data on the embodied footprint of the hardware is still scarce. Open-source tools and methodologies, such as those proposed in this chapter, are essential for enabling comparable, provider-agnostic analyses. We demonstrate that, to obtain actionable insights, analyses should employ bottom-up modeling, as it enables the identification of more granular reduction levers. These tools must also facilitate the assessment, identification, and reduction of uncertainties arising from secondary data, as well as from allocation and modeling choices.

Taking a more holistic approach, which extends beyond the established three-tier architecture to account for the software life cycle footprint, reveals the significant environmental costs associated with the building phase. The impact of this phase is largely driven by employee commuting and the environmental footprint of office spaces.

Chapter 5

Towards Ecodesign: Reducing Software Environmental Impact

In Chapter 4, we proposed various approaches and tools to assess the environmental footprint of software. However, solely estimating software environmental footprint is a first step toward ecodesign, which aims to *"reduce adverse environmental impacts throughout the life cycle of a product."*, according to ISO 14006:2020 [221].

In this chapter, the objective is to identify actionable technical levers that technical stakeholders can employ to reduce the environmental footprint of software, with a particular emphasis on microservices within cloud architectures. To achieve this, we first introduce a conceptual model that outlines the liabilities of architecture in software's environmental impact. Then, we propose two quality metrics for software architecture: we introduce the concept of resource and environmental waste caused by software architecture, and conclude with the idea of software environmental impact proportionality.

5.1 Attributing Environmental Liability Across Software Components

Most quality metrics prioritize energy consumption as the key factor in reducing software's environmental footprint, while neglecting the embodied impact of hardware, which we show the importance of in Chapter 4. Additionally, a narrow focus on usage overlooks the strain imposed on natural resources essential for the manufacturing of ICT devices (cf. Section 3.3). As such, there is a need for quality metrics to adopt a broader perspective than existing approaches that focus on energy consumption by encompassing also hardware to shed light on the consequences of over-reserving resources in software architecture.

The environmental impact attributed to a deployed software (thus excluding its development cost) can be considered as the combination of multiple contributing factors: its *requirements*, *implementation* and *infrastructure*.

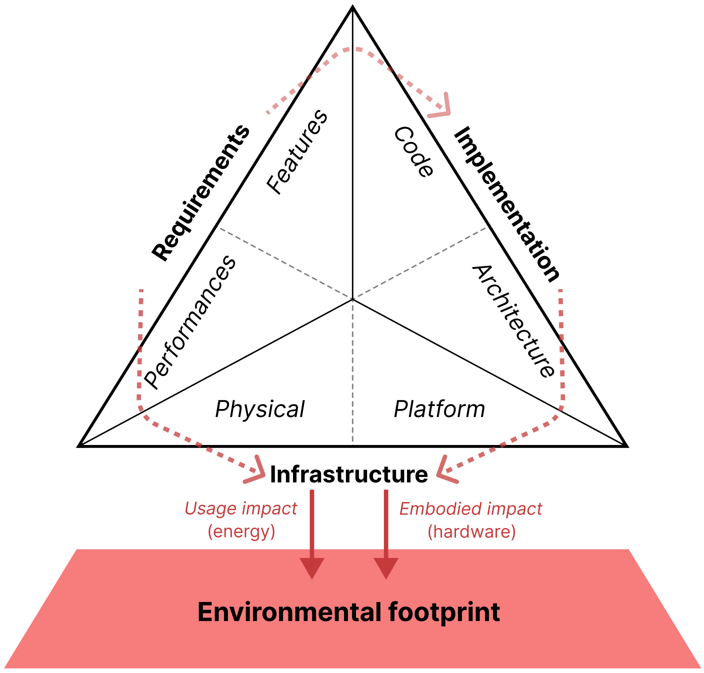

Fig. 5.1 Software Environmental Footprint Liability Framework

Figure 5.1 illustrates these factors with a triangle structure, where these environmental impacts materialize through the *infrastructure*, which lies at the foundational level of the triangle by offering the physical devices and platform solutions that are required to host software services. This level answers the needs expressed by the *implementation*, which itself fulfills the functional and non-functional needs of *requirements*.

The possibilities to decrease the first-order environmental impact (cf. Subsection 2.3.3) of a deployed software are thus threefold: lowering the *requirements* through features or performance, optimizing the *implementation* through code or architecture, or optimizing the *infrastructure*.

Consequently, despite each triangle side's environmental impact indirectly stemming from the infrastructure, each one bears the responsibility to mitigate it. If one side fails to address the pressure it places on the other sides of the triangle, it will impede the ability of the other sides to alleviate their own pressures, creating a cascading effect that ultimately amplifies the overall environmental impact.

We thus define *liability* as the responsibility for each of the three sides' acts and omissions. Each one exhibits its respective levers that can influence others to reduce—or increase—the overall software environmental impact. All these factors are interdependent: if one reduces waste, such as the *infrastructure* using resource over-subscription, but others do not, such as *features* demanding excessively high-performance levels, the final environmental impact may not decrease.

Such conceptual liability model can be used by researchers to define new metrics aimed at reducing the environmental footprint of the resulting software notably in the context of software sustainability (cf Subsection 2.3.3). This approach can also be employed to mitigate the environmental footprint of software beyond first-order effects, by incorporating functional dimensions in addition to technical considerations. In the reminding of this chapter, we specifically focus on *architecture*'s role towards sustainable software. As it is included within the *implementation* side, its liability lies in reducing the pressure it exerts on the *infrastructure*, while adhering to *requirements* of features and performance. In this case, such pressure materializes through computing resources reserved and used.

5.2 Defining a New Quality Metric for Reducing Wasted Resources

Within the conceptual model presented in Figure 5.1, we can derive concrete metrics and KPIs that are actionable at the implementation level, enabling software technical stakeholders to contribute to reducing the environmental footprint within their *liability*. In this section, we propose two quality metrics specifically designed to minimize waste induced by software architecture, particularly in the context of microservices deployments.

5.2.1 Identifying Wasted Resources

Waste, defined by the Cambridge Dictionary as *an unnecessary or wrong use* of resources,[1] has been extensively studied as an improvement factor across various domains of software engineering, particularly through the adoption of lean practices in the SDLC. For production systems, Womack *et al.* [222] introduced the concept of *Lean Thinking*, which focuses on identifying and eliminating waste within a value stream. In this context, waste refers to activities that do not create value for the customer, are unnecessary, and therefore should be removed. This concept was adapted from manufacturing to software engineering through the 7 principles of *Lean Software Development* [223].

However, lean development is primarily a tool for financial cost-efficiency rather than environmental impacts reduction—although it can have secondary benefits in reducing the software life cycle's environmental footprint (cf. Section 4.3). Waste has also been studied from the perspective of infrastructure management (cf. Subsection 2.3.3), and we propose a paradigm shift focusing on the reduction of waste in computing resources *induced* by software architecture, adhering to the conceptual model presented in Section 5.1.

Software-induced waste, whether from over-provision or static reservation of resources, carries environmental consequences beyond energy consumption (Section 2.1) for which software bears responsibility. Software projects are typically sized based on an often overestimated peak usage scenario, leading to resources being reserved over time and rarely scaled down or adjusted dynamically to match the actual needs of architectural components. As a result, most ICT infrastructures continue to operate at low usage levels (cf. Subsection 2.3.3), failing to reduce their associated environmental footprint. To identify such unsustainable practices in software architecture, we define the quality metric *wasted resource* as the «resources that are provisioned by an application though unused», aiming to uncover sustainability antipatterns only through extended observation periods [224].

[1] https://dictionary.cambridge.org/dictionary/english/waste

Given a period $[t_0;t_1]$, the quality metric *wasted resource*, denoted as $\mathscr{W}_r$ for a given resource r, thus depicts the area between *provisioned* ($\mathscr{P}_r$) and *used* ($\mathscr{U}_r$) resources. We note $\mathscr{P}_r(t_0,t_1)$ the amount of resources provisioned during that period, and $\mathscr{U}_r(t_0,t_1)$ the amount of resources used during that same period.

$$\mathscr{P}_r(t_0,t_1) = \int_{t_0}^{t_1} \mathscr{P}_r(t)dt \tag{5.1}$$

$$\mathscr{U}_r(t_0,t_1) = \int_{t_0}^{t_1} \mathscr{U}_r(t)dt \tag{5.2}$$

For a given computing resource (*i.e.*, memory or CPU), the *wasted resources* can then be defined as follows:

$$\forall r \in \{cpu, mem\} : \mathscr{W}_r(t_0,t_1) = \max(\mathscr{P}_r(t_0,t_1) - \mathscr{U}_r(t_0,t_1), 0) \tag{5.3}$$

5.2.2 Architectural case study

We use two benchmarks to demonstrate the consequences of waste induced by software architecture. We use the same software, allowing for a direct comparison between two deployments. As such, the same *requirements* are maintained through identical sets of features and performance thresholds (cf. Figure 5.1), and the code remains unchanged. However, the *infrastructure platform* differs, as we deploy the software using two distinct models on the same physical machine: *Container-as-a-Service* (CAAS) and a *Infrastructure-as-a-Service* (IAAS). The benchmarks were conducted on an HP Z6 G4 workstation, equipped with 96 GB of RAM, two Xeon Gold 5118 CPUs (2.30 GHz, 48 cores), a 5 TB SSD, and running Ubuntu 22.04.

To achieve this, we deploy the open-source application Gitlab,[2] version 17.5. In both CAAS and IAAS, the deployment follows the reference architecture and resource reservation designed to accommodate a workload of up to 2,000 users,[3] while adhering to the application performance thresholds.

Input workload. To simulate the application's usage, we use the *Gitlab Performance Tool*,[4] which conduct performance load testing across the diverse features, routes, and components of the application and assess the completion of performance threshold for each of them. We apply a varying workload to simulate a real application usage, simulating varying users (between 200 and 2,000). We purposely reach the maximum workload of 2,000 users once, representing a software's usage that does not consistently peak. Between tests, we incorporate a 15-minute pause to allow the application to scale down appropriately before scaling up again. We monitor the system for 14 hours to also observe its response to periods of inactivity.

[2]https://gitlab.com/gitlab-org/gitlab
[3]https://docs.gitlab.com/ee/administration/reference_architectures/2k_users.html
[4]https://gitlab.com/gitlab-org/quality/performance

Gitlab on CaaS. To study a CAAS architecture, we deploy the Gitlab Helm chart[5] within k3d,[6], a tool facilitating the creation of containerized k3s clusters.[7] When using this approach, each component of the application is deployed as a set one or several *pods*, whose number automatically adapts to the workload. This provides a very dynamic allocation of resources. The monitoring of hardware resources is done using kube-prometheus-stack,[8] which deploys kube-state-metrics [225] and prometheus-node-exporter [226].

Gitlab on IaaS. To deploy a cluster of virtual machines, the combination of libvirt[9] and QEMU[10] is adopted. Gitlab is deployed using the Linux package.[11] When using this approach a set of virtual machines is created to host the application's components, and resource allocation is static. Hardware resources monitoring is achieved through the built-in Gitlab Prometheus exporter.[12]

5.2.3 Results

Figure 5.2 depicts the induced waste on CPU and memory, respectively $\mathscr{W}_{cpu}$ and $\mathscr{W}_{mem}$, for the same workload on both CAAS and IAAS studied for 14 hours (cf. Subsection 5.2.2). Typically, one can observe that a dynamic reservation strategy considerably reduces *wasted resources*, as computing requirements mostly evolve following the load level. One can also note their significance in statically-provisioned software solutions. This is particularly striking in the case of statically provisioned memory, which is twice larger than what is constantly used by the application, without many variations. In the case of dynamically provisioned memory, however, we observe that *wasted resources* are consequently smaller thanks to an aggressive reservation mechanism.

Using the approach outlined in Subsection 4.1, we assess the GHGs footprint associated with the reservation of monitored resources, particularly CPU and memory. To provide a more accurate representation of the environmental impact of a cloud platform compared to that of the workstation used (cf. Subsection 5.2.2), we utilize the previously mentioned infrastructure footprint. Given that our assessment is based on a sampled application over a relatively short period of time, we do not include the storage footprint in this analysis. Figure 5.3 logically reflects trends similar to those observed in resource usage and reservation (Figure 5.2), due to the proportionality inherent in the impact assessment methodology. The figure, however, highlights a significant footprint associated with usage impact, due to the CPU-intensive nature of the application. Although static allocation results in considerable carbon waste, dynamic allocation shows higher usage footprint peaks.

[5]https://gitlab.com/gitlab-org/charts/gitlab
[6]https://github.com/k3d-io/k3d
[7]https://github.com/k3s-io/k3s
[8]https://github.com/prometheus-community/helm-charts/tree/main/charts/kube-prometheus-stack
[9]https://libvirt.org/
[10]https://www.qemu.org/
[11]https://docs.gitlab.com/omnibus/installation
[12]https://docs.gitlab.com/ee/administration/monitoring/prometheus

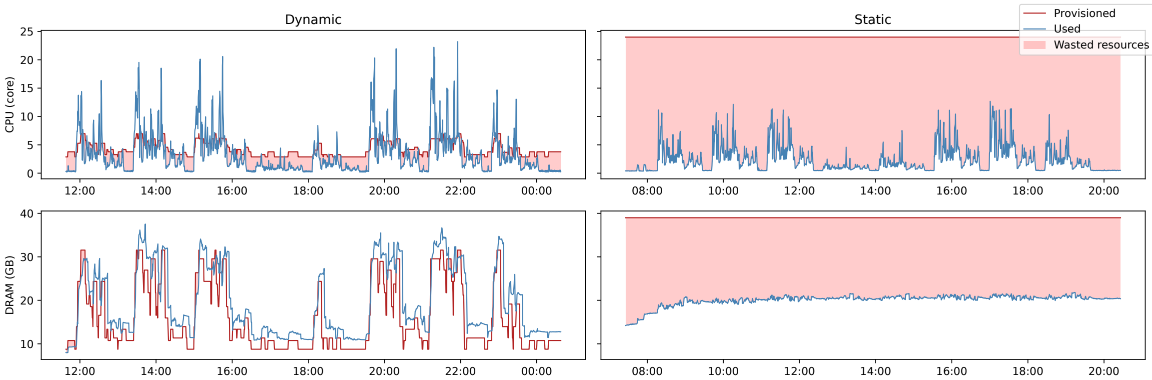

Fig. 5.2 Wasted Resources: $\mathscr{W}cpu$ (Top) and $\mathscr{W}mem$ (Bottom) for Software Benchmarks in CaaS (Dynamic, Left) and IaaS (Static, Right) Environments

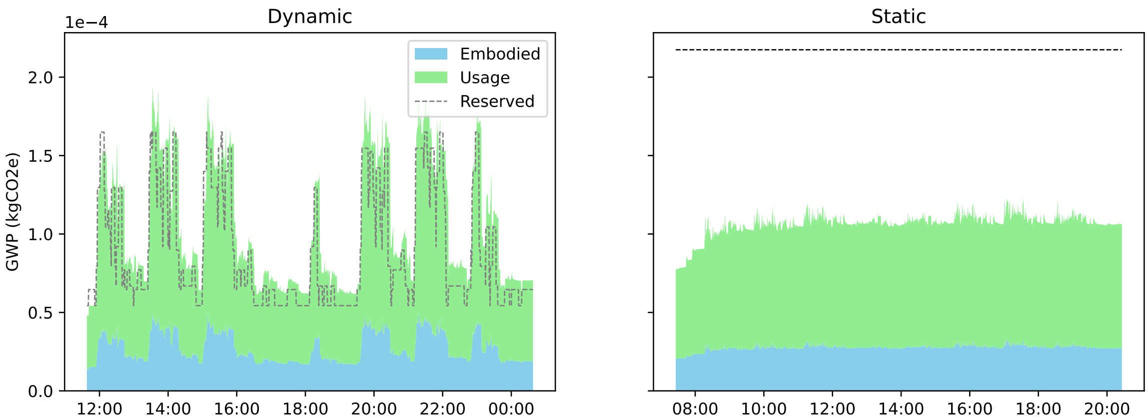

Fig. 5.3 Comparison of Stacked Embodied and Usage GHG Emissions with Over-Reservation Emissions, Evaluated Using the Methodology in Subsection 4.1

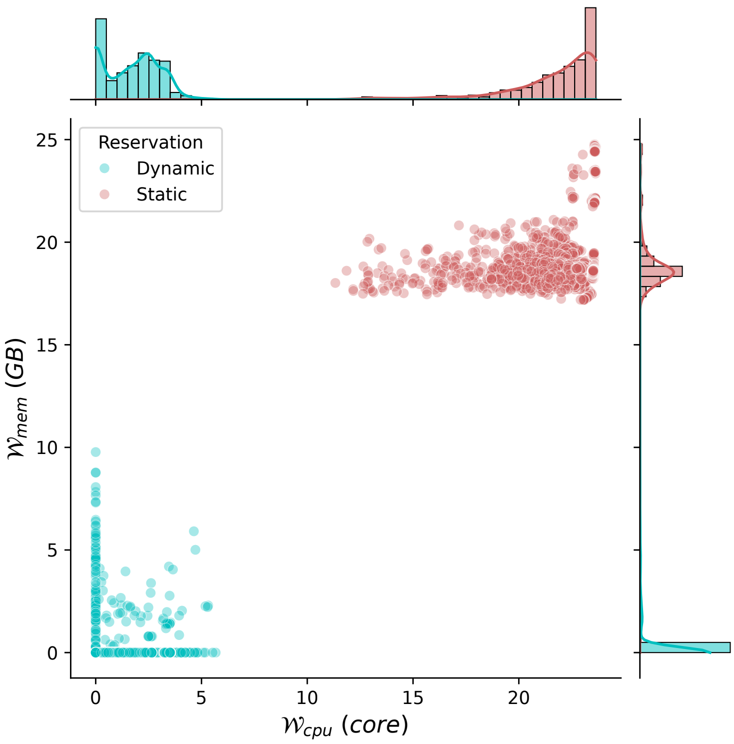

Fig. 5.4 $\mathscr{W}_{cpu}$ & $\mathscr{W}_{mem}$ Relationship

Low usage leads to high waste While being sized for peak usage in IAAS infrastructure, few applications reach it or have a constant load at a high-stress level. Most of their lifetime is spent with an intermittent load, which is not reciprocated in their mainly static resource reservation.

Figure 5.4 illustrates the relationship between $\mathscr{W}_{cpu}$ and $\mathscr{W}_{mem}$ for both architectures deployed. It demonstrates the distinct levels of wasted resources and the correlation between computing resources, and highlights that for both computing resources, an architecture wasting CPU will likely also waste memory.

Furthermore, in the case of statically-provisioned resources, it is striking that most of the period is characterized by low usage, resulting in a notable accumulation of *wasted resources*.

Wasted resources depicts that the distribution of resource utilization levels highlights that most of the time is spent under low load levels, which is contradictory to a provision for peak usage. Its inclusion as a software architecture quality metric would help stakeholders identify such waste, towards more sustainable practices in the long-term.

Breakdown by software service An application-level perspective is useful for highlighting unsustainable patterns but lacks actionable insights for software architects. Software environmental footprint is often treated as a single complex entity instead of acknowledging the interconnected components that it consists of, which fails to deliver detailed insights into problematic components [136].

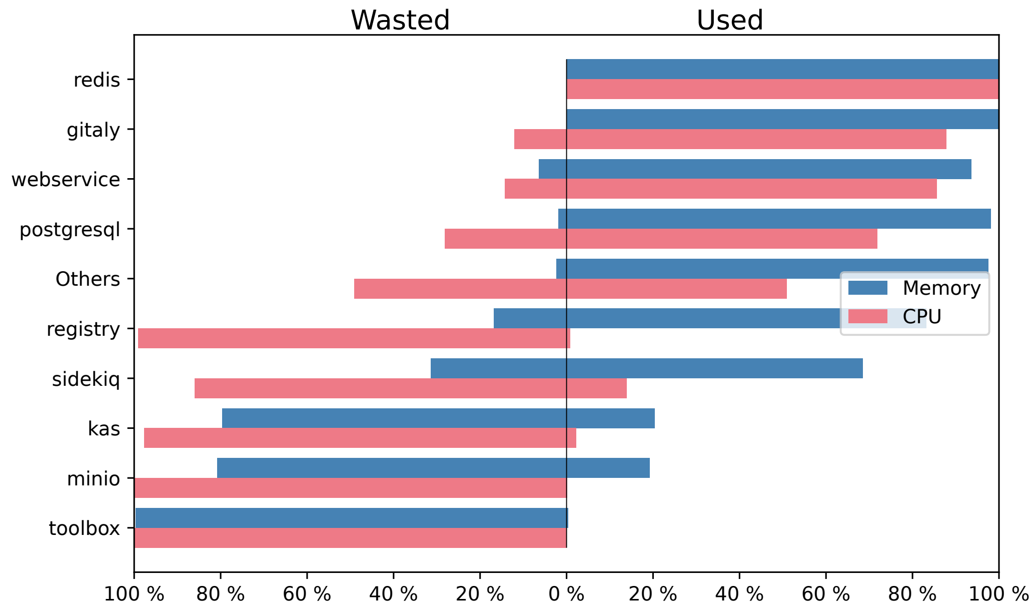

Fig. 5.5 Wasted to Used Resources Breakdown by GitLab Component (Dynamic Reservation)

As such, *wasted resources* metric should be assessed independently for each component to obtain more finely-grained feedback, based on Equation 5.3. Software architecture is particularly suited to finding the root causes of such wastes, as it allows us to consider components rather than software as a monolith.

Figure 5.5 outlines the share of wasted-to-used resources for all Gitlab architecture's components in a CAAS architecture. These components do not share the same responsibility in the overall software-induced *wasted resources*. According to the one looked at, the ratio of wasted-to-used resources can vary significantly among these components. Moreover, following the computing resource considered the behavior can be different, though a correlation is discernible. While some components waste almost all their reserved resources, some others use them fully. It is crucial to address components that emerge as the primary contributors to *wasted resources* as they play a pivotal role in environmental inefficiency.

While a relative comparison provides a good way to identify the main sources of *wasted resources*, it is only the first step to dive deeper into software-induced waste and better understand its root causes. For real-world applications, such assessments should be conducted over longer period of time, to strive towards the overall reduction of the strain placed on the *infrastructure*, and indirectly the environmental impact. In ever-evolving software architectures, constant monitoring can uncover sustainability antipatterns and induced environmental impacts that can be easily avoided.

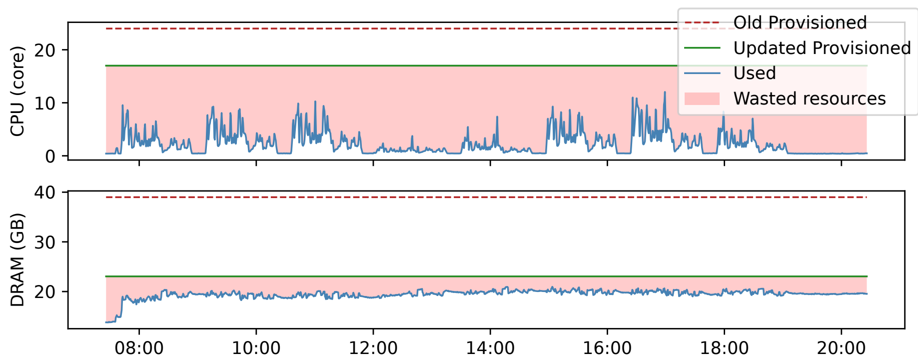

Fig. 5.6 Reducing $\mathscr{W}_{cpu}$ and $\mathscr{W}_{mem}$ via Adjusted Provisioning Strategies

5.2.4 Outcomes of Reducing Wasted Resources

In Section 5.1, we argued that the liability of the *implementation* triangle side to reduce the pressure placed on the *infrastructure* while conforming to *requirements*, notably in features and performance. In this section, we illustrate, how continuous monitoring of *wasted resources* enables software architects to uncover unsustainable resource reservation patterns and promptly implement corrective actions.

In the case of a static resource reservation mechanism, the reduction of *wasted resources* can be achieved by enhancing the correlation between provisioned resources and their observed usage. For example, in our benchmark, we were able to reduce $\mathscr{W}_{cpu}$ by 32% and $\mathscr{W}_{mem}$ by 80%, as shown in Figure 5.6, by sizing closer to the maximum resource used, while still adhering to requirements: the same features and level of performance.

Limitations

In the case of dynamically provisioned resources, we observed in Figure 5.2 that *wasted resources* were already relatively low, and further reduction would yield marginal results for a considerable investment. However, it can also be observed in this figure that *wasted* resources are often none, as *used* resources may exceed *provisioned* one. While beneficial when aiming to lower *wasted resources*, it appears contradictory to the responsibility of the implementation, which is to lower the pressure it places on the infrastructure. Is it reasonable to keep the *infrastructure* triangle side unaware of the actual computing resource requirements of a software?

Moreover, in a multi-tenant environment, such as a cloud infrastructure, resource availability is not ensured unless explicitly provisioned, potentially compromising the performance standards that software architecture should uphold. More importantly, it mandates the infrastructure to over-provision computing resources, fostering unsustainable patterns resulting from software architecture behaviors.

Therefore, to complement the *wasted resources* metrics and address such limitations, a complementary metric can be used *stolen resources*, illustrated in Figure 5.7. Stolen resources represents

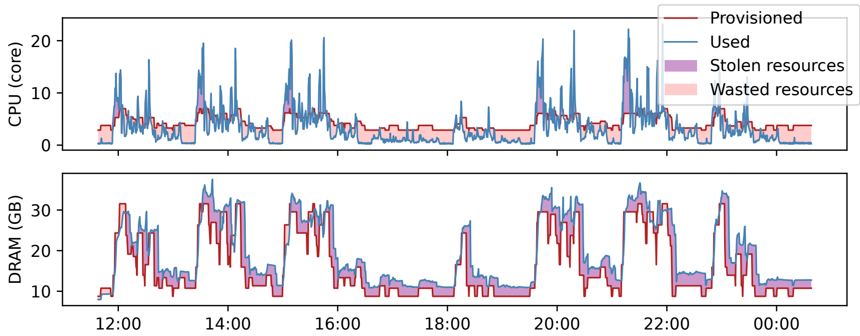

Fig. 5.7 Balancing Wasted Resources $\mathscr{W}r$ Against Stolen Resources $\mathscr{S}r$

the share of resources an architecture uses without explicit reservation to the *implementation* side for their provision:

$$\forall r \in \{cpu, mem\} : \mathscr{S}_r(t_0,t_1) = \max(\mathscr{U}_r(t_0,t_1) - \mathscr{P}_r(t_0,t_1), 0) \tag{5.4}$$

5.3 Seeking Proportionality in Software Environmental Footprint

The limitation of the *wasted resources* metric is that, although it addresses the issue of unused resources, it does not offer a thorough assessment of how efficiently resources are utilized. However, improving resource consumption remains essential, as software can still exhibit high resource usage even under periods of low activity and workload. For example, software may experience problems such as memory leaks or shadow processes. To address this, we propose a novel approach that evaluates software resource usage in relation to the evolution of its *Functional Unit* (FU).

5.3.1 Conceptual Modeling

According to the definition of a FU in Subsection 2.2.2, it serves as a reference point for quantifying the performance of a product or service in relation to its primary function. The FU therefore reflects *how much* of the service or product is being provided for the assessment.

In the context of software, the FU can take various forms. For example, it could be performance-related and be measured by the average response time to a request, the number of users served, or using more traditional performance quality metrics. This FU is not necessarily limited to technical aspects; it can also encompass non-technical outcomes, such as the financial benefits generated by the software or its critical importance within an industrial system. Importantly, the FU represents performance over a specified time period, rather than a singular, temporary observation.

In this approach, all resources utilized by the software to fulfill the FU are attributed to that specific function. In the example of the GitLab application, we define it as the number of packets transmitted by the application over a 14-hour period. Consequently, all associated resources and resulting impacts are allocated to this defined quantity. Therefore, the ecodesign of a software application involves minimizing the environmental footprint per unit of workload and evaluating the proportionality between this FU evolution and the observed resource consumption and associated impact.

Ideally, the environmental impact of a software and subsequently its components should be directly proportional to their usage: if a component remains unused, it should not generate any impact. Each component should strive to handle its workload as resource-efficiently as possible, thus minimizing its environmental footprint. Not monitoring this relationship may lead to the emergence of sustainability antipatterns, where components with low usage still contribute disproportionately high environmental impacts.

Resource usage is considered optimal when software components handle a workload w while consuming the minimum possible resources. A straightforward approach to defining this optimal resource consumption is to assume that it is strictly proportional to the *minimum* observed usage. Therefore, for a given workload w, the ideal resource consumption res_{ideal} for any resource (such as CPU, memory, or energy consumption) is expressed as:

$$res_{ideal}(w) = res_{min} \times w \tag{5.5}$$

In this context, considering w as the number of packets transmitted, the constant res_{min} can be determined as:

$$res_{min} = min_{0 \leq t \leq T} \frac{res_{software}(t)}{packets_{transmitted}(t)} \tag{5.6}$$

Where $t \in T$ reflects the time in a period T, during which we calculate the minimal value of the $res_{software}$ over $packets_{transmitted}$ ratio, without considering the workload w.

Such a simple approach however lacks actionable insights for developers, by having a single parameter to optimize. As such, we elect for a more complex modeling with the following format:

$$res_{optimal}(w) = a \times w^b + c \tag{5.7}$$

This formula models resource usage as a function of workload, where a serves as the scaling factor that defines the baseline amount of resources required per unit of workload. This coefficient establishes the initial relationship between workload and resource consumption. The exponent b governs how resource usage scales with the workload in a non-linear fashion. Specifically, when $b < 1$, it indicates diminishing marginal costs, meaning that as workload increases, the additional resources required per unit of workload decrease. This pattern is often observed with resources such as memory, which tends to exhibit high idle consumption and decreasing incremental costs as workload rises. The term c represents this idle resource consumption, reflecting the constant usage of resources even in the absence of workload. This idle cost is independent of the workload and accounts for the system's baseline consumption.

These different factors provide software engineers with various levers for optimizing resource efficiency. Reducing the scaling factor a would lower the baseline resource consumption per unit of workload, leading to more efficient resource use for individual tasks. Decreasing the exponent b would result in greater efficiency gains at higher workloads, making the system less sensitive to increases in workload. Finally, reducing the idle cost c would minimize resource usage when the system is idle, thereby reducing the baseline overhead during periods of inactivity. Together, these adjustments would result in a more efficient system, characterized by lower resource consumption both when idle and as workload scales.

5.3.2 Results

To show the benefits of resource proportionality, we consider the CAAS architecture deployment of GitLab, as it demonstrated the lowest level of *wasted resources* in Subsection 5.2.2. The deployment characteristics and workload remain unchanged, but in addition to monitoring memory and CPU resource usage, we also track CPU energy consumption per microservice using the PowerAPI software power meter [227]. While energy consumption was modeled in Figure 5.3, we now directly measure it as we focus specifically on usage footprint optimizations. Energy consumption is more closely tied to workload fluctuations, necessitating finer-grained monitoring. This allows for more detailed environmental impact modeling, considering both embodied and usage footprints, in line with the methodology outlined in Subsection 4.1.

Figure 5.8 illustrates the modeling of these three resources using Equation 5.7. On the left, a scatter plot displays the observed resource consumption against the number of transmitted packets, highlighting their relationship. As seen in the previous section, most data points are concentrated in the lower range of transmitted packets, reflecting the fact that the workload rarely reaches peak levels. For each component, res_{ideal} is calculated using Equation 5.5, demonstrating that real-world applications do not exhibit strict proportionality by default.

To determine the initial values of a, b, and c, the curve fitting function from the Scikit-learn library [228] is used. This function uses non-linear least squares optimization to align the model with the data. By providing the observed workload over the period as input and the corresponding resource usage as output, the curve fitting algorithm iteratively adjusts a, b, and c to minimize the difference between predicted and actual resource usage. This approach yields the most accurate estimates of these parameters, offering a precise representation of the relationship between workload and resource consumption, including both non-linear scaling and idle usage.

However, these factors should be arbitrarily reduced in order to work towards reducing resource consumption. In the *Optimized* scenario, each of the derived factors a, b and c is reduced by 10%. The effect of this optimization over the observed period is displayed on the right. For CPU usage, the reduction primarily affects peak periods, as CPU consumption is already largely proportional to the workload, with minimal idle consumption. In contrast, for memory, the reduction mainly impacts idle consumption, which is substantial, resulting in a usage pattern that is less dependent on

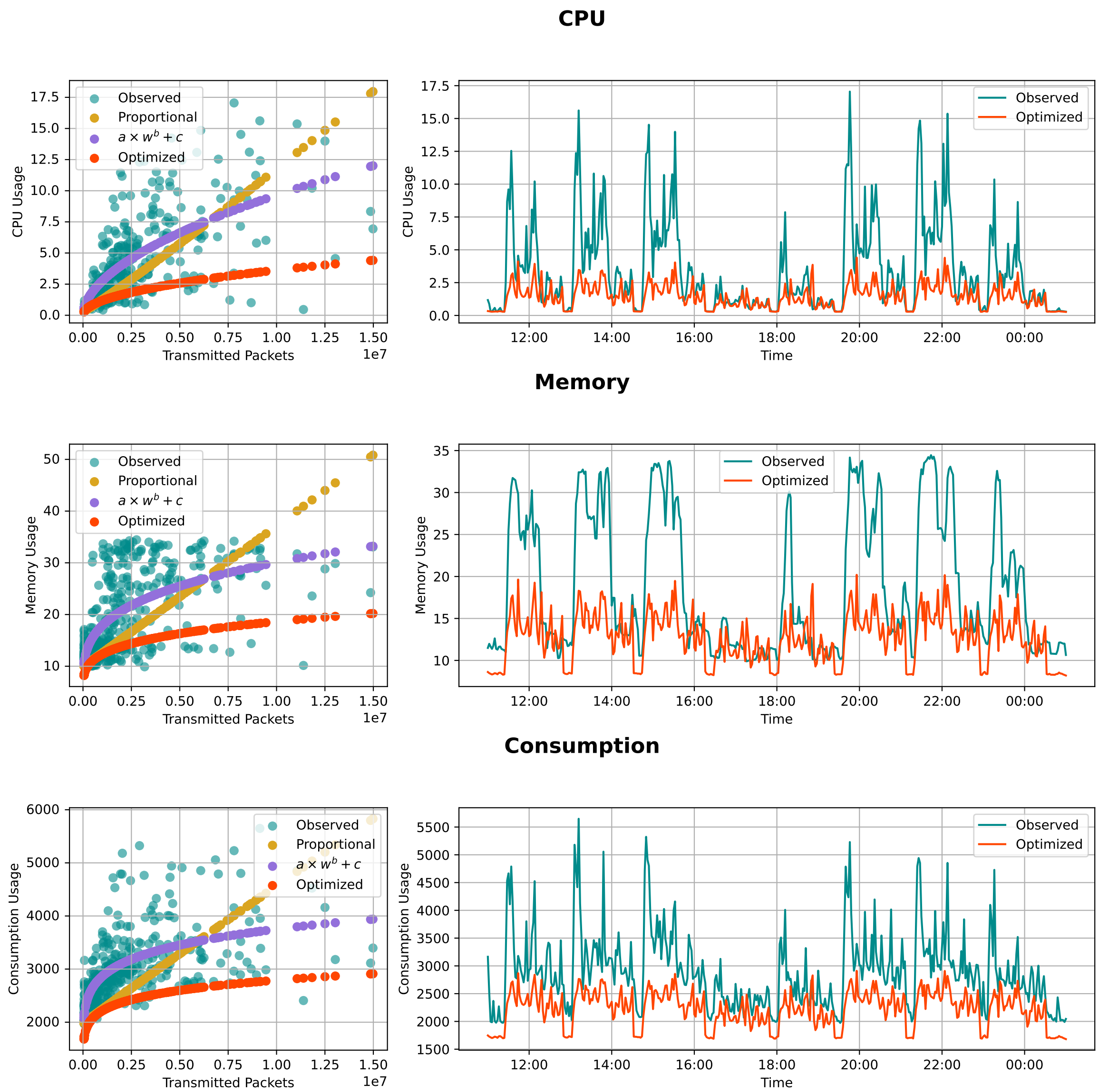

Fig. 5.8 Optimizing Resource Use Through Proportionality with Software's Functional Unit

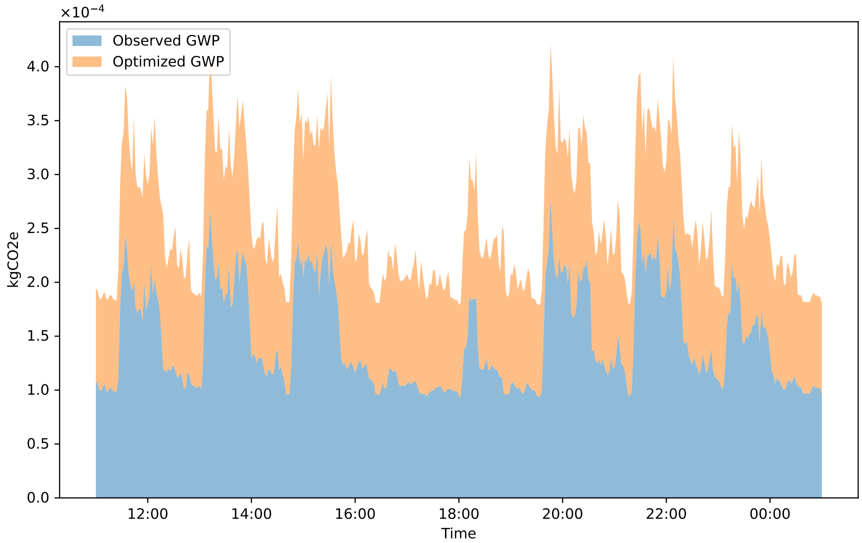

Fig. 5.9 GWP Reduction with a 10% Optimization of a, b, and c

workload fluctuations. Finally, for energy consumption, similarly to CPU usage, the optimization flattens consumption peaks, leading to a more consistent and efficient usage profile.

Estimating the environmental footprint of the infrastructure using the approach and data developed in Subsection 4.1 demonstrates that a 10% improvement in the parameters a, b, and c could result in a 24.88% reduction in *Global Warming Potential* (GWP) over the observed period and given workload. Figure 5.9 illustrates this reduction, showing that the observed values consistently exceed the optimized GWP, with a significant and persistent gap between the two. Both the observed and ideal GWP curves exhibit fluctuations throughout the day, peaking during periods of higher activity and dropping during lower usage, induced by the proportional impact modeling of the workload on emissions.

As in Subsection 5.2.3, such analysis can be conducted at the software component level, as demonstrated in Figure 5.5, to identify the components that deviate the most from the ideal. By pinpointing these components, stakeholders can prioritize optimizations where the greatest gains in resource efficiency and reduction in environmental impact can be achieved.

In real-world applications, prioritizing action levers based on their relative effectiveness in reducing emissions and optimizing resource usage is essential. Figure 5.10 presents the potential reduction in GHGs achieved by optimizing each parameter individually, as well as by optimizing all parameters simultaneously. Varying a, b, and c individually shows distinct trends in workload reduction. In the Gitlab CaaS benchmark, optimizing a leads to proportional benefits, while adjusting c, the idle cost, produces negligible results. The results reveal that optimizing the b parameter, which represents the marginal cost, offers the greatest reduction potential, particularly up to 80%, at which point the optimization of a becomes more significant. However, such high-level optimizations are largely theoretical. The curve illustrating the cumulative effect when all parameters

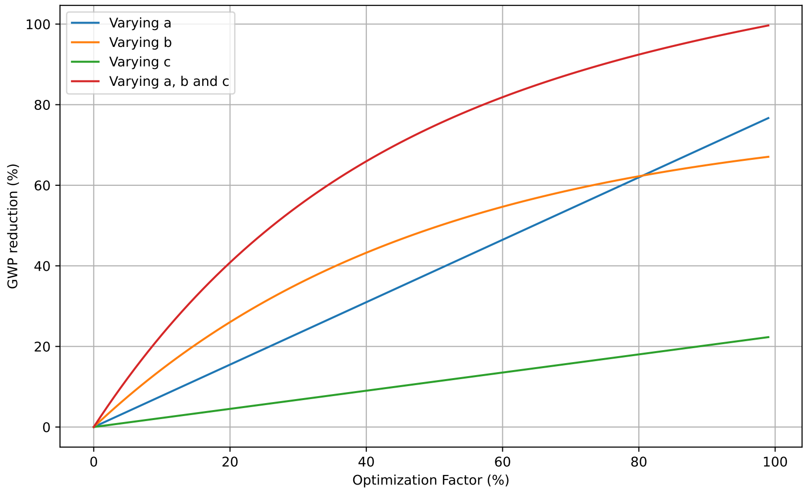

Fig. 5.10 GWP Reduction Impact of Parameter Reductions in the $a \times w^b + c$ Model for GitLab

are adjusted together, indicates a higher potential for optimization when multiple parameters are modified simultaneously, compared to changing them individually. This analysis should be repeated for different software systems, as the unique characteristics of each application may affect the generalizability of these findings. Nevertheless, this type of analysis enables stakeholders to identify the most impactful action levers for reducing emissions and optimizing resource efficiency.

Sotware ecodesign: Developing software-specific technical mitigation levers must align with their respective areas of responsibility and liability of software stakeholders. For software architecture, a practical and rapidly implementable solution is implementing technical quality metrics addressing resource waste, often the result of over-sizing systems to accommodate for peak usage. Additionally, software architecture enables more detailed analyses by viewing software as a collection of components, rather than a monolithic entity.
A key principle in resource management is ensuring proportionality between the software's environmental impact and its functional usefulness. However, software ecodesign should not be confined solely to optimizing implementation and infrastructure; it must also address the functional aspects of the software itself, ensuring that it aligns with planetary boundaries at every level of design and operation.

Chapter 6

Conclusion

In this thesis, we draw upon methodologies from various research fields to identify meaningful action levers, aimed at holistically estimating and reducing the environmental impact of software. First, we employ economic tools to provide a broad overview of the ICT sector environmental footprint as a whole. Building on prior research, we demonstrate the continuous growth of the sector's GHGs emissions since 1995. Then, we propose an approach of such methodology to assess the sector's material footprint evolution, while underscoring the inherent challenges and uncertainties involved in conducting such analyses. In the second chapter, we transition from macro-level assessments to micro-level analyses by proposing new tools and methodologies to assess the various environmental impacts of software. We particularly emphasize on the uncertainties inherent to such analyses, while also highlighting the need for a holistic perspective to take sourced decisions to reduce software footprint. Finally, we build upon these analyses, methodologies and tools to reflect on the liability of software in mitigating the sector's overall environmental impact. Based on a conceptual model, we introduce two new metrics suitable for software architecture: wasted resources and proportionality.

6.1 Perspectives

6.1.1 Short-term

The ICT sector still struggles to fully understand its environmental footprint. In future research, EEIOA could be employed as a way to better understand the intricacy of its supply chain, particularly by accounting for the embodied footprint of hardware. This approach could help generalize the inclusion of embodied impacts in impact reduction strategies by providing open-data, rather than focusing exclusively on energy consumption. A more thorough analysis of the supply chain would also enhance the consideration of other environmental impact categories associated with the ICT sector, such as water usage and land use. Additionally, this methodology could be utilized for prospective scenario analysis, particularly to explore the sector's reliance on the mineral supply chain. However, our research identified limitations in the quality of environmental data within the database used, mandating for research using alternative databases. The results should be challenged using alternative quantification methods, such as material flow analysis (MFA), to more thoroughly evaluate the validity of the findings obtained.

The scarcity of data, combined with the prevalence of closed-source methodologies, hinders meaningful comparisons of environmental footprints. To address this, both the research and software communities should work toward the development of open-source commons, including tools and methodologies such as BoaviztAPI, partially introduced in Subsection 4.1. In the short term, I plan to incorporate uncertainty management into this tool, which is already widely recognized by industry stakeholders in France. This enhancement will allow users to interpret the results as orders of magnitude, emphasizing that these tools generate estimations rather than precise measurements. Furthermore, it highlights that these estimations are strongly influenced by allocation choice, which can have a substantial impact on the final results. Additionally, I aim to integrate more advanced modeling techniques, particularly those developed in this thesis, including approaches that account for wasted resources in cloud infrastructures.

The limited quality of data on the embodied environmental footprint of ICT hardware is often cited as a justification for focusing solely on energy consumption. In future research, I plan to expand upon the conceptual liability model introduced in this thesis to create actionable metrics for software stakeholders. These metrics will go beyond energy consumption, highlighting the material impact of digital infrastructures, which is frequently neglected when energy is the sole focus.

6.1.2 Long-term

Focusing solely on software optimization is only a first step toward rethinking the future of software within planetary boundaries. While software stakeholders are accustomed to using KPI to monitor and improve various aspects of software, the ecodesign of software cannot be reduced to the addition of new metrics alone.

Ecodesign must be integrated into every phase of the SDLC, to weigh the environmental costs against the benefits of the implementation of a solution. Further research should not only aim to optimize software to fulfill its functional requirements in the most efficient way, but also to critically evaluate the intended goals of software in light of planetary boundaries. Importantly, the responsibility for ecodesign cannot only rest with technical stakeholders. It requires a broader, interdisciplinary approach to ensure that the future of software development truly with sustainability goals.

References

[1] C. Freitag, M. Berners-Lee, K. Widdicks, B. Knowles, G. S. Blair, and A. Friday, "The real climate and transformative impact of ict: A critique of estimates, trends, and regulations," *Patterns*, vol. 2, no. 9, p. 100340, 2021.

[2] Busa, Andreas and Hegeman, Malcolm, "Life cycle assessment of dell r740," 2019.

[3] J. Gröger, R. Liu, L. Stobbe, J. Druschke, and N. Richter, *Green Cloud Computing: lebenszyklusbasierte Datenerhebung zu Umweltwirkungen des Cloud Computing: Abschlussbericht*. Umweltbundesamt, 2021.

[4] "Carbon Footprint reporting methodology," 2024. [Online; accessed 05. Sep. 2024].

[5] K. Richardson, W. Steffen, W. Lucht, J. Bendtsen, S. E. Cornell, J. F. Donges, M. Drüke, I. Fetzer, G. Bala, W. von Bloh, G. Feulner, S. Fiedler, D. Gerten, T. Gleeson, M. Hofmann, W. Huiskamp, M. Kummu, C. Mohan, D. Nogués-Bravo, S. Petri, M. Porkka, S. Rahmstorf, S. Schaphoff, K. Thonicke, A. Tobian, V. Virkki, L. Wang-Erlandsson, L. Weber, and J. Rockström, "Earth beyond six of nine planetary boundaries," *Science Advances*, vol. 9, no. 37, p. eadh2458, 2023.

[6] A. S. G. Andrae and T. Edler, "On global electricity usage of communication technology: Trends to 2030," *Challenges*, vol. 6, no. 1, pp. 117–157, 2015.

[7] L. Belkhir and A. Elmeligi, "Assessing ict global emissions footprint: Trends to 2040 & recommendations," *Journal of Cleaner Production*, vol. 177, pp. 448–463, 2018.

[8] J. Malmodin and D. Lundén, "The energy and carbon footprint of the global ict and e&m sectors 2010-2015," *Sustainability*, vol. 10, no. 9, 2018.

[9] IPCC, *Climate Change 2022: Mitigation of Climate Change. Contribution of Working Group III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge, UK and New York, NY, USA: Cambridge University Press, 2022.

[10] T. Simon, D. Ekchajzer, A. Berthelot, E. Fourboul, S. Rince, and R. Rouvoy, "BoaviztAPI: a bottom-up model to assess the environmental impacts of cloud services," in *HotCarbon'24. Workshop on Sustainable Computer Systems*, (Santa Cruz, United States), July 2024.

[11] É. Guégain, T. Simon, A. Rahier, and R. Rouvoy, "Managing Uncertainties in ICT Services Life Cycle Assessment using Fuzzy Logic," in *International Conference on ICT for Sustainability (ICT4S)*, (Stockhlom, Sweden), IEEE, June 2024.

[12] T. Simon, P. Rust, R. Rouvoy, and J. Penhoat, "Uncovering the environmental impact of software life cycle," in *International Conference on ICT for Sustainability (ICT4S)*, pp. 176–187, 2023.

[13] S. Cerf, A. Luxey-Bitri, C. Quinton, R. Rouvoy, T. Simon, and C. Truffert, "Untangling the Critical Minerals Knot: when ICT hits the Energy Transitions." working paper or preprint, Dec. 2023.

[14] "Cisco Annual Internet Report (2018–2023) White Paper," Jan. 2022. [Online; accessed 12. Jul. 2024].

[15] "Facts and Figures 2023 - Internet use," July 2024. [Online; accessed 12. Jul. 2024].

[16] S. Lythreatis, S. K. Singh, and A.-N. El-Kassar, "The digital divide: A review and future research agenda," *Technological Forecasting and Social Change*, vol. 175, p. 121359, 2022.

[17] A. Ahmed and M. Coupechoux, "The long road to sobriety: Estimating the operational power consumption of cellular base stations in france," in *International Conference on ICT for Sustainability (ICT4S)*, pp. 188–196, 2023.

[18] C. V. Networking, "Cisco global cloud index: Forecast and methodology, 2016–2021," *White paper. Cisco Public, San Jose*, vol. 1, 2016.

[19] K. Pilz and L. Heim, "Compute at scale: A broad investigation into the data center industry," 2023.

[20] "EDF," Jan. 2024. [Online; accessed 12. Aug. 2024].

[21] International Energy Agency, "Electricity 2024 - analysis and key findings," 2024. A report by the International Energy Agency.

[22] "Key Findings Data Centres Metered Electricity Consumption 2023 - Central Statistics Office," July 2024. [Online; accessed 24. Jul. 2024].

[23] D. Lundén, J. Malmodin, P. Bergmark, and N. Lövehagen, "Electricity consumption and operational carbon emissions of european telecom network operators," *Sustainability*, vol. 14, no. 5, 2022.

[24] A. Pasek, H. Vaughan, and N. Starosielski, "The world wide web of carbon: Toward a relational footprinting of information and communications technology's climate impacts," *Big Data & Society*, vol. 10, no. 1, p. 20539517231158994, 2023.

[25] J. Malmodin, D. Lundén, Å. Moberg, G. Andersson, and M. Nilsson, "Life cycle assessment of ict," *Journal of Industrial Ecology*, vol. 18, no. 6, pp. 829–845, 2014.

[26] R. Itten, R. Hischier, A. S. G. Andrae, J. C. T. Bieser, L. Cabernard, A. Falke, H. Ferreboeuf, L. M. Hilty, R. L. Keller, E. Lees-Perasso, C. Preist, and M. Stucki, "Digital transformation—life cycle assessment of digital services, multifunctional devices and cloud computing," *The International Journal of Life Cycle Assessment*, vol. 25, pp. 2093–2098, Oct 2020.

[27] T. Pirson and D. Bol, "Assessing the embodied carbon footprint of iot edge devices with a bottom-up life-cycle approach," *Journal of Cleaner Production*, vol. 322, p. 128966, 2021.

[28] neilmacdonald, "Cutting industry emissions and fighting the climate crisis - ITU — itu.int." https://www.itu.int/hub/2023/11/cutting-industry-emissions-and-fighting-the-climate-crisis/. [Accessed 18-10-2024].

[29] "Microsoft 2024 Environmental Sustainability Report," July 2024. [Online; accessed 13. Jul. 2024].

[30] “2024 Environmental Report - Google Sustainability,” July 2024. [Online; accessed 13. Jul. 2024].

[31] J. Suckling and J. Lee, “Redefining scope: The true environmental impact of smartphones?,” *The International Journal of Life Cycle Assessment*, vol. 20, pp. 1181–1196, Aug. 2015.

[32] U. Gupta, Y. G. Kim, S. Lee, J. Tse, H. S. Lee, G. Wei, D. Brooks, and C. Wu, “Chasing carbon: The elusive environmental footprint of computing,” *CoRR*, vol. abs/2011.02839, 2020.

[33] Bordage, F., de Montenay, L., Benqassem, S., Delmas-Orgelet, J., Domon, F., Prunel, D., Vateau, C. et Lees Perasso, E., “Digital technologies in Europe: an environmental life cycle approach,” 2021. [Online; accessed 18. Mar. 2024].

[34] M. Pathak, R. Slade, P. Shukla, J. Skea, R. Pichs-Madruga, and D. Ürge-Vorsatz, “Technical Summary,” in *Climate Change 2022: Mitigation of Climate Change. Contribution of Working Group III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change* (P. Shukla, J. Skea, R. Slade, A. A. Khourdajie, R. van Diemen, D. McCollum, M. Pathak, S. Some, P. Vyas, R. Fradera, M. Belkacemi, A. Hasija, G. Lisboa, S. Luz, and J. Malley, eds.), Cambridge University Press, 2022.

[35] IEA, “The Role of Critical Minerals in Clean Energy Transitions,” tech. rep., IEA, May 2021.

[36] C. Fitzpatrick, E. Olivetti, T. R. Miller, R. Roth, and R. Kirchain, “Conflict minerals in the compute sector: Estimating extent of tin, tantalum, tungsten, and gold use in ict products,” *Environmental Science & Technology*, vol. 49, pp. 974–981, Jan 2015.

[37] P. Emsbo, C. Lawley, and K. Czarnota, “Geological surveys unite to improve critical mineral security,” *Eos Science News*, 2021.

[38] O. Vidal, H. Le Boulzec, B. Andrieu, and F. Verzier, “Modelling the Demand and Access of Mineral Resources in a Changing World,” *MDPI Sustainability*, vol. 14, p. 11, Dec. 2021.

[39] P. B. Cornelis, K. Ruediger, Y. Tales, M. Rosie, D. Elena, A. Shahana, B. Garam, D. Otmar, F.-C. Elena, F. Vanessa, G. Vanessa, H. Sunil, H. Shunichi, I. Giulia, S. K. Deepali, d. C. Vittoria Luda, L. Yuliya, N. Innocent, P. Noémie, and W. Michelle, *The Global E-waste Monitor 2024: Quantities, flows and the circular economy potential*. International Telecommunication Union (ITU) and United Nations Institute for Training and Research (UNITAR), 2024.

[40] F.-C. Mihai, M.-G. Gnoni, C. Meidiana, C. Ezeah, and V. Elia, “Chapter 1 - waste electrical and electronic equipment (weee): Flows, quantities, and management—a global scenario,” in *Electronic Waste Management and Treatment Technology* (M. N. V. Prasad and M. Vithanage, eds.), pp. 1–34, Butterworth-Heinemann, 2019.

[41] E. Garcia-Vazquez, V. Geslin, P. Turrero, N. Rodriguez, G. Machado-Schiaffino, and A. Ardura, “Oceanic karma? eco-ethical gaps in african eee metal cycle may hit back through seafood contamination,” *Science of The Total Environment*, vol. 762, p. 143098, 2021.

[42] “Waste electrical and electronic equipment (WEEE) by waste management operations - open scope, 6 product categories (from 2018 onwards) - Products Datasets - Eurostat,” Sept. 2022. [Online; accessed 27. Sep. 2022].

[43] R. Bashroush, N. Rteil, R. Kenny, and A. Wynne, “Optimizing server refresh cycles: The case for circular economy with an aging moore’s law,” *IEEE Transactions on Sustainable Computing*, vol. 7, pp. 189–200, jan 2022.

[44] C. W. Babbitt, S. Althaf, F. Cruz Rios, M. M. Bilec, and T. Graedel, “The role of design in circular economy solutions for critical materials,” *One Earth*, vol. 4, pp. 353–362, Mar. 2021.

[45] A. G. Jones, “Mining for net zero: The impossible task,” *The Leading Edge*, vol. 42, pp. 266–276, Apr. 2023.

[46] IEA, “Net Zero by 2050 - A Roadmap for the Global Energy Sector,” tech. rep., IEA, 2021.

[47] B. A. McNulty and S. M. Jowitt, “Barriers to and uncertainties in understanding and quantifying global critical mineral and element supply,” *iScience*, vol. 24, no. 7, p. 102809, 2021.

[48] “United Nations Treaty Collection,” Sept. 2023. [Online; accessed 26. Sep. 2023].

[49] A. Beylot, J. Dewulf, T. Greffe, S. Muller, and G.-A. Blengini, “Mineral resources depletion, dissipation and accessibility in lca: a critical analysis,” *The International Journal of Life Cycle Assessment*, vol. 29, pp. 890–908, May 2024.

[50] C. A. Chan, A. F. Gygax, E. Wong, C. A. Leckie, A. Nirmalathas, and D. C. Kilper, “Methodologies for assessing the use-phase power consumption and greenhouse gas emissions of telecommunications network services,” *Environmental Science & Technology*, vol. 47, no. 1, pp. 485–492, 2013. PMID: 23211093.

[51] V. C. Coroama and L. M. Hilty, “Assessing internet energy intensity: A review of methods and results,” *Environmental Impact Assessment Review*, vol. 45, pp. 63–68, 2014.

[52] D. Schien, V. C. Coroama, L. M. Hilty, and C. Preist, “The energy intensity of the internet: Edge and core networks,” in *ICT Innovations for Sustainability* (L. M. Hilty and B. Aebischer, eds.), (Cham), pp. 157–170, Springer International Publishing, 2015.

[53] “ISO 14040:2006,” July 2023. [Online; accessed 18. Jul. 2023].

[54] I. Boustead, “LCA — how it came about: — the beginning in the u.k.,” *The International Journal of Life Cycle Assessment*, vol. 1, pp. 147–150, September 1996.

[55] R. G. Hunt, W. E. Franklin, and R. G. Hunt, “LCA — how it came about: — personal reflections on the origin and the development of LCA in the USA,” *The International Journal of Life Cycle Assessment*, vol. 1, no. 1, pp. 4–7, 1996.

[56] “ISO 14044:2006,” Feb. 2023. [Online; accessed 7. Feb. 2023].

[57] M. Finkbeiner, A. Inaba, R. Tan, K. Christiansen, and H.-J. Klüppel, “The new international standards for life cycle assessment: Iso 14040 and iso 14044,” *The international journal of life cycle assessment*, vol. 11, no. 2, pp. 80–85, 2006.

[58] H. Wenzel, “Application dependency of lca methodology: Key variables and their mode of influencing the method,” *The International Journal of Life Cycle Assessment*, vol. 3, pp. 281–288, Sep 1998.

[59] T. Ekvall, “Cleaner production tools: Lca and beyond,” *Journal of Cleaner Production*, vol. 10, no. 5, pp. 403–406, 2002. Integrating greener product development perspectives.

[60] B. A. Sandén and M. Karlström, “Positive and negative feedback in consequential life-cycle assessment,” *Journal of Cleaner Production*, vol. 15, no. 15, pp. 1469–1481, 2007.

[61] T. Billstein, A. Björklund, and T. Rydberg, “Life cycle assessment of network traffic: A review of challenges and possible solutions,” *Sustainability*, vol. 13, no. 20, 2021.

[62] S. Lutter, S. Giljum, and M. Bruckner, “A review and comparative assessment of existing approaches to calculate material footprints,” *Ecological Economics*, vol. 127, pp. 1–10, 2016.

[63] D. Mytton, “Assessing the suitability of the Greenhouse Gas Protocol for calculation of emissions from public cloud computing workloads,” *Journal of Cloud Computing*, vol. 9, p. 45, Dec. 2020.

[64] L.-P. P.-V. Clément, Q. E. Jacquemotte, and L. M. Hilty, “Sources of variation in life cycle assessments of smartphones and tablet computers,” *Environmental Impact Assessment Review*, vol. 84, p. 106416, 2020.

[65] S. Pauliuk, G. Majeau-Bettez, C. L. Mutel, B. Steubing, and K. Stadler, “Lifting industrial ecology modeling to a new level of quality and transparency: A call for more transparent publications and a collaborative open source software framework,” *Journal of Industrial Ecology*, vol. 19, no. 6, pp. 937–949, 2015.

[66] Y. Arushanyan, E. Ekener-Petersen, and G. Finnveden, “Lessons learned – review of lcas for ict products and services,” *Computers in Industry*, vol. 65, no. 2, pp. 211–234, 2014.

[67] A. S. Andrae, “New perspectives on internet electricity use in 2030,” *Engineering and Applied Science Letters*, vol. 3, no. 2, pp. 19–31, 2020.

[68] “Green data centers in focus,” Dec. 2021. [Online; accessed 5. Jul. 2022].

[69] N. Vandromme, T. Dandres, M. Elsa, R. Samson, S. Khazri, R. F. Moghaddam, K. K. Nguyen, Y. Lemieux, and M. Cheriet, “Life cycle assessment of videoconferencing with call management servers relying on virtualization,” in *International Conference on ICT for Sustainability (ICT4S)*, pp. 281–289, 2014/08.

[70] R. Hischier, M. A. Achachlouei, and L. M. Hilty, “Evaluating the sustainability of electronic media: Strategies for life cycle inventory data collection and their implications for lca results,” *Environmental Modelling & Software*, vol. 56, pp. 27–36, 2014. Thematic issue on Modelling and evaluating the sustainability of smart solutions.

[71] R. K. Rosenbaum, S. Georgiadis, and P. Fantke, *Uncertainty Management and Sensitivity Analysis*, pp. 271–321. Cham: Springer International Publishing, 2018.

[72] S.-C. Lo, H. wen Ma, and S.-L. Lo, “Quantifying and reducing uncertainty in life cycle assessment using the bayesian monte carlo method,” *Science of The Total Environment*, vol. 340, no. 1, pp. 23–33, 2005.

[73] E. Igos, E. Benetto, R. Meyer, P. Baustert, and B. Othoniel, “How to treat uncertainties in life cycle assessment studies?,” *The International Journal of Life Cycle Assessment*, vol. 24, pp. 794–807, Apr 2019.

[74] A. Weckenmann and A. Schwan, “Environmental life cycle assessment with support of fuzzy-sets,” *The International Journal of Life Cycle Assessment*, vol. 6, pp. 13–18, Jan 2001.

[75] G. Egilmez, S. Gumus, M. Kucukvar, and O. Tatari, “A fuzzy data envelopment analysis framework for dealing with uncertainty impacts of input–output life cycle assessment models on eco-efficiency assessment,” *Journal of Cleaner Production*, vol. 129, pp. 622–636, 2016.

[76] B. Gonzalez, B. Adenso-Dıaz, and P. Gonzalez-Torre, “A fuzzy logic approach for the impact assessment in lca,” *Resources, Conservation and Recycling*, vol. 37, no. 1, pp. 61–79, 2002.

[77] L. M. Hilty, "Co2 reduction with ict: Prospects and barriers.," in *EnviroInfo (1)*, pp. 35–42, 2007.

[78] A. Rasoldier, J. Combaz, A. Girault, K. Marquet, and S. Quinton, "How realistic are claims about the benefits of using digital technologies for GHG emissions mitigation?," in *Eighth Workshop on Computing within Limits 2022*, LIMITS, jun 21 2022. https://limits.pubpub.org/pub/real.

[79] V. Forti, C. Baldé, R. Kuehr, and G. Bel, *The Global E-waste Monitor 2020. Quantities, flows, and the circular economy potential*. United Nations University (UNU)/United Nations Institute for Training and Research (UNITAR) – co-hosted SCYCLE Programme, International Telecommunication Union (ITU) & International Solid Waste Association (ISWA), 2020.

[80] W. Leontief, *Input-output economics*. Oxford University Press, 1986.

[81] R. E. Miller and P. D. Blair, *Input-output analysis: foundations and extensions*. Cambridge university press, 2009.

[82] K. Stadler, R. Wood, T. Bulavskaya, C.-J. Södersten, M. Simas, S. Schmidt, A. Usubiaga, J. Acosta-Fernández, J. Kuenen, M. Bruckner, S. Giljum, S. Lutter, S. Merciai, J. H. Schmidt, M. C. Theurl, C. Plutzar, T. Kastner, N. Eisenmenger, K.-H. Erb, A. de Koning, and A. Tukker, "Exiobase 3: Developing a time series of detailed environmentally extended multi-regional input-output tables," *Journal of Industrial Ecology*, vol. 22, no. 3, pp. 502–515, 2018.

[83] D. Caro, S. Bastianoni, S. Borghesi, and F. M. Pulselli, "On the feasibility of a consumer-based allocation method in national ghg inventories," *Ecological Indicators*, vol. 36, pp. 640–643, 2014.

[84] B. A. Thomas and I. L. Azevedo, "Estimating direct and indirect rebound effects for u.s. households with input-output analysis part 1: Theoretical framework," *Ecological Economics*, vol. 86, pp. 199–210, 2013. Sustainable Urbanisation: A resilient future.

[85] Y. Virtanen, S. Kurppa, M. Saarinen, J.-M. Katajajuuri, K. Usva, I. Mäenpää, J. Mäkelä, J. Grönroos, and A. Nissinen, "Carbon footprint of food – approaches from national input–output statistics and a lca of a food portion," *Journal of Cleaner Production*, vol. 19, no. 16, pp. 1849–1856, 2011. Promoting Transformation towards Sustainable Consumption and Production in a Resource and Energy Intensive Economy - the Case of Finland.

[86] M. Bruckner, S. Giljum, C. Lutz, and K. S. Wiebe, "Materials embodied in international trade – global material extraction and consumption between 1995 and 2005," *Global Environmental Change*, vol. 22, no. 3, pp. 568–576, 2012. Global transformations, social metabolism and the dynamics of socio-environmental conflicts.

[87] Z.-M. Chen and G. Chen, "Virtual water accounting for the globalized world economy: National water footprint and international virtual water trade," *Ecological Indicators*, vol. 28, pp. 142–149, 2013. 10 years Ecological Indicators.

[88] J. Kitzes, "An introduction to environmentally-extended input-output analysis," *Resources*, vol. 2, no. 4, pp. 489–503, 2013.

[89] M. Lenzen, J. Murray, F. Sack, and T. Wiedmann, "Shared producer and consumer responsibility — theory and practice," *Ecological Economics*, vol. 61, no. 1, pp. 27–42, 2007.

[90] M. L. Peter L. Daniels and S. J. Kenway, "The ins and outs of water use - a review of multi-region input-output analysis and water footprints for regional sustainability analysis and policy," *Economic Systems Research*, vol. 23, no. 4, pp. 353–370, 2011.

[91] X. Zhou, D. Zhou, Q. Wang, and B. Su, "How information and communication technology drives carbon emissions: A sector-level analysis for china," *Energy Economics*, vol. 81, pp. 380–392, 2019.

[92] G. San Miguel, B. M. Bañales, D. Ruiz, S. Álvarez, J. Pérez, and M. T. Arredondo, "Carbon footprint and employment generation produced by ict networks for internet deployment: A multi-regional input-output analysis," *Science of The Total Environment*, vol. 914, p. 169776, 2024.

[93] K. Schoer, J. Weinzettel, J. Kovanda, J. Giegrich, and C. Lauwigi, "Raw material consumption of the european union–concept, calculation method, and results," *Environ Sci Technol*, vol. 46, pp. 8903–8909, Aug. 2012.

[94] N. Myhrvold, "The future of software, the software industry, and windows '47," in *ACM97: The Next 50 Years of Computing*, ACM '97, (New York, NY, USA), Association for Computing Machinery, 1997.

[95] A. Arora, V. Arunachalam, J. Asundi, and R. Fernandes, "The indian software services industry," *Research Policy*, vol. 30, no. 8, pp. 1267–1287, 2001.

[96] D. Randall, R. Wood, S. Bony, R. Colman, T. Fichefet, J. Fyfe, V. Kattsov, A. Pitman, J. Shukla, J. Srinivasan, S. Ronald, A. Sumi, and K. Taylor, *Climate Models and Their Evaluation*. 01 2007.

[97] D. Rolnick, P. L. Donti, L. H. Kaack, K. Kochanski, A. Lacoste, K. Sankaran, A. S. Ross, N. Milojevic-Dupont, N. Jaques, A. Waldman-Brown, A. S. Luccioni, T. Maharaj, E. D. Sherwin, S. K. Mukkavilli, K. P. Kording, C. P. Gomes, A. Y. Ng, D. Hassabis, J. C. Platt, F. Creutzig, J. Chayes, and Y. Bengio, "Tackling climate change with machine learning," *ACM Comput. Surv.*, vol. 55, feb 2022.

[98] P. Lago, Q. Gu, and P. Bozzelli, *A systematic literature review of green software metrics*. VU Technical Report, 2014.

[99] S. K. Mishra, D. Puthal, B. Sahoo, P. P. Jayaraman, S. Jun, A. Y. Zomaya, and R. Ranjan, "Energy-efficient vm-placement in cloud data center," *Sustainable Computing: Informatics and Systems*, vol. 20, pp. 48–55, 2018.

[100] E. A. Jagroep, J. M. van der Werf, S. Brinkkemper, G. Procaccianti, P. Lago, L. Blom, and R. van Vliet, "Software energy profiling: Comparing releases of a software product," in *Proceedings of the 38th International Conference on Software Engineering Companion*, ICSE '16, (New York, NY, USA), p. 523–532, Association for Computing Machinery, 2016.

[101] A. Carette, M. Younes, G. Hecht, N. Moha, and R. Rouvoy, "Investigating the energy impact of android smells," in *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, (Los Alamitos, CA, USA), pp. 115–126, IEEE Computer Society, feb 2017.

[102] ITU-T, "Methodology for environmental life cycle assessments of information and communication technology goods, networks and services," Recommendation L.410, International Telecommunication Union, Geneva, July 2014.

[103] ADEME, "General principles for the environmental labelling of consumer products, methodological standard for the environmental assessment of digital service," tech. rep., ADEME, july 2021. [Online; accessed 13. Aug. 2024].

[104] P. Bhatia, C. Cummis, L. Draucker, D. Rich, H. Lahd, and A. Brown, “Greenhouse gas protocol product life cycle accounting and reporting standard,” 2011.

[105] G. Protocol, “Ict sector guidance built on the ghg protocol product life cycle accounting and reporting standard,” *Global: GHG Protocol*, 2017.

[106] Green-Software-Foundation, “software_carbon_intensity,” Jan. 2023. [Online; accessed 13. Aug. 2024].

[107] I. IEC, “Iso/iec 21031: Information technology – software carbon intensity (sci) specification,” *Switzerland: ISO*, 2024.

[108] S. D. C. F. S. G. (SG), “Methodology for calculating the environmental footprint of server-side applications,” 2022. [Online; accessed 24. Jan. 2023].

[109] Z. Ournani, R. Rouvoy, P. Rust, and J. Penhoat, “On reducing the energy consumption of software: From hurdles to requirements,” in *Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, ESEM ’20, (New York, NY, USA), Association for Computing Machinery, 2020.

[110] A. D. Komeil Raisian, Jamaiah Yahaya, “Green measurements for software product based on sustainability dimensions,” *Computer Systems Science and Engineering*, vol. 41, no. 1, pp. 271–288, 2022.

[111] S. S. Mahmoud and I. Ahmad, “A green model for sustainable software engineering,” *International Journal of Software Engineering and Its Applications*, vol. 7, no. 4, pp. 55–74, 2013.

[112] M. Dick and S. Naumann, “Enhancing software engineering processes towards sustainable software product design,” in *Integration of Environmental Information in Europe: Proceedings of the 24th International Conference on Informatics for Environmental Protection, EnviroInfo 2010, Cologne/Bonn, Germany, October 6-8, 2010* (K. Greve and A. B. Cremers, eds.), pp. 706–715, Shaker Verlag, Aachen, 2010.

[113] S. Naumann, M. Dick, E. Kern, and T. Johann, “The GREENSOFT Model: A reference model for green and sustainable software and its engineering,” *Sustainable Computing: Informatics and Systems*, vol. 1, pp. 294–304, Dec. 2011.

[114] S. S. Shenoy and R. Eeratta, “Green software development model: An approach towards sustainable software development,” in *2011 Annual IEEE India Conference*, pp. 1–6, 2011.

[115] R. Verdecchia, P. Lago, and C. de Vries, “The future of sustainable digital infrastructures: A landscape of solutions, adoption factors, impediments, open problems, and scenarios,” *Sustainable Computing: Informatics and Systems*, vol. 35, p. 100767, 2022.

[116] I. ISO *et al.*, “Iso/iec/ieee 12207: 2017, systems and software engineering–software life cycle processes,” *International Organization for Standardization*, 2017.

[117] S. S. Shenoy and R. Eeratta, “Green software development model: An approach towards sustainable software development,” in *2011 Annual IEEE India Conference*, pp. 1–6, 2011.

[118] N. B. Ruparelia, “Software development lifecycle models,” *SIGSOFT Softw. Eng. Notes*, vol. 35, pp. 8–13, may 2010.

[119] A. Chandani, S. Wagholikar, and O. Prakash, “A bibliometric analysis of green computing,” in *Information and Communication Technology for Competitive Strategies (ICTCS 2021)* (A. Joshi, M. Mahmud, and R. G. Ragel, eds.), (Singapore), pp. 547–557, Springer Nature Singapore, 2023.

[120] M. Kumar, “A green it star model approach for software development life cycle,” *international journal of Advanced technology in engineering and Science*, vol. 3, pp. 548–559, 03 2015.

[121] G. Lami, F. Fabbrini, and M. Fusani, “Software sustainability from a process-centric perspective,” in *Systems, Software and Services Process Improvement* (D. Winkler, R. V. O’Connor, and R. Messnarz, eds.), (Berlin, Heidelberg), pp. 97–108, Springer Berlin Heidelberg, 2012.

[122] J. Taina, “How green is your software?,” in *International Conference of Software Business*, pp. 151–162, Springer, 2010.

[123] E. Kern, M. Dick, S. Naumann, and T. Hiller, “Impacts of software and its engineering on the carbon footprint of ict,” *Environmental Impact Assessment Review*, vol. 52, pp. 53–61, 2015.

[124] World Commission on Environment and Development, *Our Common Future*. London, England: Oxford University Press, Apr. 1987.

[125] P. Lago, S. A. Koçak, I. Crnkovic, and B. Penzenstadler, “Framing sustainability as a property of software quality,” *Commun. ACM*, vol. 58, pp. 70–78, sep 2015.

[126] L. M. Hilty and B. Aebischer, *ICT for Sustainability: An Emerging Research Field*. Cham: Springer International Publishing, 2015.

[127] H. Koziolek, “Sustainability evaluation of software architectures: A systematic review,” in *Proceedings of the Joint ACM SIGSOFT Conference – QoSA and ACM SIGSOFT Symposium – ISARCS on Quality of Software Architectures – QoSA and Architecting Critical Systems – ISARCS*, QoSA-ISARCS ’11, (New York, NY, USA), p. 3–12, Association for Computing Machinery, 2011.

[128] B. Penzenstadler, A. Raturi, D. Richardson, C. Calero, H. Femmer, and X. Franch, “Systematic mapping study on software engineering for sustainability (se4s),” in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, EASE ’14, (New York, NY, USA), Association for Computing Machinery, 2014.

[129] “THE 17 GOALS | Sustainable Development,” Sept. 2023. [Online; accessed 26. Sep. 2023].

[130] C. Calero, M. Ángeles Moraga, and F. García, “Software, sustainability, and un sustainable development goals,” *IT Professional*, vol. 24, no. 1, pp. 41–48, 2022.

[131] protontypes, “open-sustainable-technology,” May 2023. [Online; accessed 22. May 2023].

[132] C. Calero and M. Piattini, “Puzzling out software sustainability,” *Sustainable Computing: Informatics and Systems*, vol. 16, pp. 117–124, 2017.

[133] C. Calero, M. Á. Moraga, M. F. Bertoa, and L. Duboc, “Quality in use and software greenability,” in *RE4SuSy@RE*, 2014.

[134] B. Penzenstadler, A. Raturi, D. Richardson, and B. Tomlinson, “Safety, security, now sustainability: The nonfunctional requirement for the 21st century,” *IEEE Software*, vol. 31, no. 3, pp. 40–47, 2014.

[135] I. IEC, "Iso/iec 25010: System and software engineering-system and software quality requirements and evaluation (square)-system and software quality models," *Switzerland: ISO*, 2011.

[136] E. Jagroep, J. M. van der Werf, S. Brinkkemper, L. Blom, and R. van Vliet, "Extending software architecture views with an energy consumption perspective," *Computing*, vol. 99, pp. 553–573, Jun 2017.

[137] P. Lago, S. A. Koçak, I. Crnkovic, and B. Penzenstadler, "Framing sustainability as a property of software quality," *Commun. ACM*, vol. 58, p. 70–78, sep 2015.

[138] S. Vos, P. Lago, R. Verdecchia, and I. Heitlager, "Architectural tactics to optimize software for energy efficiency in the public cloud," in *International Conference on ICT for Sustainability (ICT4S)*, pp. 77–87, 2022.

[139] I. Fatima and P. Lago, "Towards a sustainability-aware software architecture evaluation for cloud-based software services," 2023.

[140] J. Taina, "Good, bad, and beautiful software - in search of green software quality factors," *CEPIS UPGRADE*, vol. 2011, pp. 22–27, Oct. 2011.

[141] C. Delimitrou and C. Kozyrakis, "Quasar: Resource-efficient and qos-aware cluster management," in *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '14, (New York, NY, USA), pp. 127–144, Association for Computing Machinery, 2014.

[142] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Heterogeneity and dynamicity of clouds at scale: Google trace analysis," in *Proceedings of the Third ACM Symposium on Cloud Computing*, SoCC '12, (New York, NY, USA), Association for Computing Machinery, 2012.

[143] S. A. Jyothi, C. Curino, I. Menache, S. M. Narayanamurthy, A. Tumanov, J. Yaniv, R. Mavlyutov, I. n. Goiri, S. Krishnan, J. Kulkarni, and S. Rao, "Morpheus: towards automated slos for enterprise clusters," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, (USA), p. 117–134, USENIX Association, 2016.

[144] I. Fatima and P. Lago, "A review of software architecture evaluation methods for sustainability assessment," in *2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C)*, pp. 191–194, 2023.

[145] W. Vereecken, L. Deboosere, P. Simoens, B. Vermeulen, D. Colle, C. Develder, M. Pickavet, B. Dhoedt, and P. Demeester, "Energy efficiency in thin client solutions," in *Networks for Grid Applications* (A. Doulamis, J. Mambretti, I. Tomkos, and T. Varvarigou, eds.), (Berlin, Heidelberg), pp. 109–116, Springer Berlin Heidelberg, 2010.

[146] W. Van Heddeghem, S. Lambert, B. Lannoo, D. Colle, M. Pickavet, and P. Demeester, "Trends in worldwide ict electricity consumption from 2007 to 2012," *Computer Communications*, vol. 50, pp. 64–76, 2014. Green Networking.

[147] J. Malmodin, N. Lövehagen, P. Bergmark, and D. Lundén, "Ict sector electricity consumption and greenhouse gas emissions – 2020 outcome," *Telecommunications Policy*, vol. 48, no. 3, p. 102701, 2024.

[148] UN, *International Standard Industrial Classification of All Economic Activities (ISIC), Rev.3.1*. Statistical Papers (Ser. M), United Nations, Feb. 2002.

[149] M. Berners-Lee, D. Howard, J. Moss, K. Kaivanto, and W. Scott, "Greenhouse gas footprinting for small businesses — the use of input–output data," *Science of The Total Environment*, vol. 409, no. 5, pp. 883–891, 2011.

[150] C. Kennelly, M. Berners-Lee, and C. Hewitt, "Hybrid life-cycle assessment for robust, best-practice carbon accounting," *Journal of Cleaner Production*, vol. 208, pp. 35–43, 2019.

[151] K. Stadler, R. Wood, T. Bulavskaya, C.-J. Södersten, M. Simas, S. Schmidt, A. Usubiaga, J. Acosta-Fernández, J. Kuenen, M. Bruckner, S. Giljum, S. Lutter, S. Merciai, J. H. Schmidt, M. C. Theurl, C. Plutzar, T. Kastner, N. Eisenmenger, K.-H. Erb, A. Koning, and A. Tukker, "Exiobase 3," 2021.

[152] F. Charpentier, B. Martins, and P. Bourcier, "Estimating the carbon footprint of ict using input-output analysis: Dealing with overcounting and other challenges," in *2023 International Conference on ICT for Sustainability (ICT4S)*, pp. 154–163, 2023.

[153] W. Leontief, "Environmental repercussions and the economic structure: An input-output approach," *The Review of Economics and Statistics*, vol. 52, no. 3, pp. 262–271, 1970.

[154] G. P. Peters, "From production-based to consumption-based national emission inventories," *Ecological Economics*, vol. 65, no. 1, pp. 13–23, 2008.

[155] G. Atkinson, K. Hamilton, G. Ruta, and D. Van Der Mensbrugghe, "Trade in 'virtual carbon': Empirical results and implications for policy," *Global Environmental Change*, vol. 21, no. 2, pp. 563–574, 2011. Special Issue on The Politics and Policy of Carbon Capture and Storage.

[156] K. W. Steininger, C. Lininger, L. H. Meyer, P. Muñoz, and T. Schinko, "Multiple carbon accounting to support just and effective climate policies," *Nature Climate Change*, vol. 6, pp. 35–41, Jan 2016.

[157] K. Feng, K. Hubacek, L. Sun, and Z. Liu, "Consumption-based co2 accounting of china's megacities: The case of beijing, tianjin, shanghai and chongqing," *Ecological Indicators*, vol. 47, pp. 26–31, 2014. Integrated Ecological Indicators for Sustainable Urban Ecosystem Evaluation and Management.

[158] T. Wiedmann, H. C. Wilting, M. Lenzen, S. Lutter, and V. Palm, "Quo vadis mrio? methodological, data and institutional requirements for multi-region input–output analysis," *Ecological Economics*, vol. 70, no. 11, pp. 1937–1945, 2011. Special Section - Earth System Governance: Accountability and Legitimacy.

[159] D. Giannone, L. Reichlin, and D. Small, "Nowcasting: The real-time informational content of macroeconomic data," *Journal of Monetary Economics*, vol. 55, no. 4, pp. 665–676, 2008.

[160] K. Stadler, R. Wood, T. Bulavskaya, C.-J. Södersten, M. Simas, S. Schmidt, A. Usubiaga, J. Acosta-Fernández, J. Kuenen, M. Bruckner, S. Giljum, S. Lutter, S. Merciai, J. H. Schmidt, M. C. Theurl, C. Plutzar, T. Kastner, N. Eisenmenger, K.-H. Erb, A. Koning, and A. Tukker, "Exiobase 3," 2021.

[161] "pymrio," July 2024. [Online; accessed 24. Jul. 2024].

[162] T. F. Stocker, D. Qin, G.-K. Plattner, M. Tignor, S. K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex, and P. M. Midgley, "Climate change 2013: The physical science basis," tech. rep., Intergovernmental Panel on Climate Change, Working Group I Contribution to the IPCC Fifth Assessment Report (AR5)(Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA.), 2013. 1535 pp.

[163] B. Steubing, A. de Koning, S. Merciai, and A. Tukker, "How do carbon footprints from lca and eeioa databases compare? a comparison of ecoinvent and exiobase," *Journal of Industrial Ecology*, vol. 26, no. 4, pp. 1406–1422, 2022.

[164] "UK and world mineral statistics datasets - British Geological Survey," Jan. 2024. [Online; accessed 3. Sep. 2024].

[165] "International Minerals Statistics and Information | U.S. Geological Survey," Sept. 2024. [Online; accessed 3. Sep. 2024].

[166] C. Reichl, M. Schatz, and G. Zsak, "World mining data," *Minerals Production International Organizing Committee for the World Mining Congresses*, vol. 32, no. 1, pp. 1–261, 2014.

[167] I. M. F. R. Dept., *World Economic Outlook, October 2015: Adjusting to Lower Commodity Prices*. International Monetary Fund, 2015.

[168] T. D. Pearce, J. D. Ford, J. Prno, F. Duerden, J. Pittman, M. Beaumier, L. Berrang-Ford, and B. Smit, "Climate change and mining in canada," *Mitigation and Adaptation Strategies for Global Change*, vol. 16, pp. 347–368, Mar 2011.

[169] "Understanding your carbon emission estimations - AWS Billing," 2024. [Online; accessed 18. Mar. 2024].

[170] "A new approach for Scope 3 emissions transparency," 2021. [Online; accessed 18. Mar. 2024].

[171] "Carbon Footprint reporting methodology," 2024. [Online; accessed 18. Mar. 2024].

[172] "Overview | Cloud Carbon Footprint," Feb. 2024. [Online; accessed 18. Mar. 2024].

[173] "All Published SPEC SPECpower_ssj2008 Results," May 2024. [Online; accessed 8. May 2024].

[174] B. K. Sovacool, "The precarious political economy of cobalt: Balancing prosperity, poverty, and brutality in artisanal and industrial mining in the democratic republic of the congo," *The Extractive Industries and Society*, vol. 6, no. 3, pp. 915–939, 2019.

[175] G. Pitron, *The Dark Cloud: how the digital world is costing the earth*. Scribe Publications, 2023.

[176] Jan Konietzko, "Moving beyond carbon tunnel vision with a sustainability data strategy," 2022.

[177] R. L. Lewison, A. F. Johnson, J. Gan, R. Pelc, K. Westfall, and M. Helvey, "Accounting for unintended consequences of resource policy: Connecting research that addresses displacement of environmental impacts," *Conservation Letters*, vol. 12, no. 3, p. e12628, 2019.

[178] M. Amidpour and M. H. Khoshgoftar Manesh, "Chapter 11 - environmental impacts consideration," in *Cogeneration and Polygeneration Systems* (M. Amidpour and M. H. Khoshgoftar Manesh, eds.), pp. 195–213, Academic Press, 2021.

[179] Boavizta, "boaviztapi," Sept. 2022. [Online; accessed 19. Aug. 2024].

[180] U. Gupta, M. Elgamal, G. Hills, G.-Y. Wei, H.-H. S. Lee, D. Brooks, and C.-J. Wu, "ACT: designing sustainable computer systems with an architectural carbon modeling tool," in *Proceedings of the 49th Annual International Symposium on Computer Architecture*, ISCA '22, pp. 784–799, Association for Computing Machinery, 2022.

[181] L. van Oers, J. B. Guinée, and R. Heijungs, "Abiotic resource depletion potentials (ADPs) for elements revisited—updating ultimate reserve estimates and introducing time series for production data," *The International Journal of Life Cycle Assessment*, vol. 25, no. 2, pp. 294–308, 2020.

[182] R. Frischknecht, F. Wyss, S. Büsser Knöpfel, T. Lützkendorf, and M. Balouktsi, "Cumulative energy demand in LCA: the energy harvested approach," *The International Journal of Life Cycle Assessment*, vol. 20, no. 7, pp. 957–969, 2015.

[183] T. Pirson, T. P. Delhaye, A. G. Pip, G. Le Brun, and D. Bol, "The Environmental Footprint of IC Production: Review, Analysis, and Lessons From Historical Trends," *IEEE Trans. Semicond. Manuf.*, vol. PP, p. 1, Feb. 2023.

[184] TechPowerUp, "Cpu specs database," 2024. [Online; accessed 18. Mar. 2024].

[185] J. Choe, "Memory technology 2021: Trends & challenges," in *International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*, pp. 111–115, 2021.

[186] Y. Aiouch, A. Chanoine, L. Corbet, P. Drapeau, L. Ollion, and V. Vigneron, "Evaluation de l'impact environnemental du numérique en France et analyse prospective - La librairie ADEME," 2022.

[187] C. L. Belady and C. G. Malone, "Metrics and an infrastructure model to evaluate data center efficiency," in *Proceedings of the ASME 2007 InterPACK Conference collocated with the ASME/JSME 2007 Thermal Engineering Heat Transfer Summer Conference*, vol. ASME 2007 InterPACK Conference, Volume 1, pp. 751–755, 07 2007.

[188] N. Scarlat, M. Prussi, and M. Padella, "Quantification of the carbon intensity of electricity produced and used in Europe," *Appl. Energy*, vol. 305, p. 117901, Jan. 2022.

[189] "Base Empreinte," Sept. 2024. [Online; accessed 20. Sep. 2024].

[190] J. W. Pierre Delforge, "America's data centers consuming and wasting growing amounts of energy," tech. rep., NRDC, 2015.

[191] P. Jacquet, T. Ledoux, and R. Rouvoy, "ScroogeVM: Boosting cloud resource utilization with dynamic oversubscription," *IEEE Transactions on Sustainable Computing*, pp. 1–13, 2024. Publisher: IEEE.

[192] J. V. Andreasen, "Carbon Efficient Karpenter: Optimizing Kubernetes Cluster Autoscaling for Carbon Efficiency," Master's thesis, Aarhus University, 2024.

[193] M. Proske, *Ecodesign Concepts and Environmental Assessments Addressing Product Obsolescence*. PhD thesis, TU Berlin, 2023. https://depositonce.tu-berlin.de/bitstreams/677ac517-6c03-41c2-b285-5a1449771df3/download.

[194] M. Cortes-Cornax, P. Lago, and C. Roncancio, "A case study of environmental considerations and opportunities in cyber physical systems," *IET Software*, vol. 17, no. 4, pp. 424–434, 2023.

[195] B. Everman, T. Villwock, D. Chen, N. Soto, O. Zhang, and Z. Zong, "Evaluating the carbon impact of large language models at the inference stage," in *2023 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, pp. 150–157, 2023.

[196] A. Berthelot, E. Caron, M. Jay, and L. Lefèvre, "Estimating the environmental impact of generative-ai services using an lca-based methodology," *Procedia CIRP*, vol. 122, pp. 707–712, 2024. 31st CIRP Conference on Life Cycle Engineering.

[197] N. Patel, A. Mamandipoor, D. Quinn, and M. Alian, "Xfm: Accelerated software-defined far memory," in *Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO '23, (New York, NY, USA), p. 769–783, Association for Computing Machinery, 2023.

[198] Fairphone, "Life cycle assessment of the fairphone 3," 2020.

[199] M. Ercan, J. Malmodin, P. Bergmark, E. Kimfalk, and E. Nilsson, "Life cycle assessment of a smartphone," in *International Conference on ICT for Sustainability (ICT4S)*, pp. 124–133, 2016/08.

[200] Apple, "iphone 6 plus environmental report," 2014.

[201] B. P. Weidema and M. S. Wesnæs, "Data quality management for life cycle inventories—an example of using data quality indicators," *Journal of Cleaner Production*, vol. 4, no. 3, pp. 167–174, 1996.

[202] A. Weckenmann and A. Schwan, "Environmental life cycle assessment with support of fuzzy-sets," *The International Journal of Life Cycle Assessment*, vol. 6, pp. 13–18, 2001.

[203] A. Cerulli-Harms, J. Suter, W. Landzaat, C. Duke, A. Rodriguez Diaz, L. Porch, K. Svatikova, J. Vermeulen, T. Smit, F. Dekeulenaer, *et al.*, *Behavioural study on consumers' engagement in the circular economy*. 2018.

[204] P. Grzegorzewski and K. Pasternak-Winiarska, "Trapezoidal approximations of fuzzy numbers with restrictions on the support and core," in *Proceedings of the 7th conference of the European Society for Fuzzy Logic and Technology*, pp. 749–756, Atlantis Press, 2011.

[205] X. Cheng and S. Li, "Interval estimations of building heating energy consumption using the degree-day method and fuzzy numbers," *Buildings*, vol. 8, no. 2, p. 21, 2018.

[206] W. Van Leekwijck and E. E. Kerre, "Defuzzification: criteria and classification," *Fuzzy sets and systems*, vol. 108, no. 2, pp. 159–178, 1999.

[207] A. Ciroth, "Validation – the missing link in life cycle assessment. towards pragmatic lcas," *The International Journal of Life Cycle Assessment*, vol. 11, pp. 295–297, Sep 2006.

[208] hgrecco, "pint," Feb. 2023. [Online; accessed 14. Feb. 2023].

[209] AlDanial, "cloc," Jan. 2023. [Online; accessed 17. Jan. 2023].

[210] "GitLab.org / GitLab · GitLab," Jan. 2023. [Online; accessed 17. Jan. 2023].

[211] B. Boehm, B. Clark, E. Horowitz, C. Westland, R. Madachy, and R. Selby, "Cost models for future software life cycle processes: Cocomo 2.0," *Annals of Software Engineering*, vol. 1, pp. 57–94, Dec 1995.

[212] "Reference architecture: up to 50,000 users | GitLab," Jan. 2023. [Online; accessed 23. Jan. 2023].

[213] "Résultats détaillés de l'enquête mobilité des personnes 2019," Feb. 2023. [Online; accessed 14. Feb. 2023].

[214] "Les cadres parcourent chaque année près de 3000 km de plus que les employés pour se rendre au travail | L'Observatoire des Territoires," Feb. 2023. [Online; accessed 14. Feb. 2023].

[215] "ADEME - Site Bilans GES," Feb. 2023. [Online; accessed 14. Feb. 2023].

[216] “Négaoctet (Base Impacts - Documentation - Numérique),” Sept. 2023. [Online; accessed 20. Sep. 2023].

[217] S. S, C. E, S. M, and P. R, “Global normalisation factors for the environmental footprint and life cycle assessment,” no. KJ-NA-28984-EN-C (print),KJ-NA-28984-EN-N (online), 2017.

[218] A. Roesch, S. Sala, and N. Jungbluth, “Normalization and weighting: the open challenge in lca,” *The International Journal of Life Cycle Assessment*, vol. 25, pp. 1859–1865, Sep 2020.

[219] B. Caldarola and S. Sorrell, “Do teleworkers travel less? evidence from the english national travel survey,” *Transportation Research Part A: Policy and Practice*, vol. 159, pp. 282–303, 2022.

[220] A. Hook, V. Court, B. K. Sovacool, and S. Sorrell, “A systematic review of the energy and climate impacts of teleworking,” *Environmental Research Letters*, vol. 15, p. 093003, aug 2020.

[221] “ISO 14006:2020,” Feb. 2023. [Online; accessed 7. Feb. 2023].

[222] J. P. Womack and D. T. Jones, *Lean thinking*. New York, NY: Simon & Schuster, July 2003.

[223] M. Poppendieck and T. Poppendieck, *Lean software development*. Boston, MA: Addison-Wesley Educational, May 2003.

[224] P. Lago, “Architecture design decision maps for software sustainability,” in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, pp. 61–64, 2019.

[225] kubernetes, “kube-state-metrics,” Jan. 2024. [Online; accessed 8. Jan. 2024].

[226] prometheus community, “helm-charts,” Jan. 2024. [Online; accessed 8. Jan. 2024].

[227] G. Fieni, D. R. Acero, P. Rust, and R. Rouvoy, “Powerapi: A python framework for building software-defined power meters,” *Journal of Open Source Software*, vol. 9, no. 98, p. 6670, 2024.

[228] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.