VERSION 202405

GREENSPECTOR'S METHODOLOGY

HOW DOES GREENSPECTOR ROBUSTLY MEASURE THE ENVIRONMENTAL IMPACTS OF DIGITAL SERVICES ?

Table of content

1. Introduction

Smartphones are integral to the modern landscape, serving a wide range of purposes from productivity to entertainment, and from shopping to controlling connected devices. However, their nomadic nature brings new challenges. On one hand, users must adapt to their limited battery capacity. On the other hand, network connectivity is constrained by the user's data plan and available connection speed. Thus, applications draining battery or data are potential nuisances to the user. The negative consequences of high battery usage extend beyond user dissatisfaction, affecting smartphone batteries' limited lifespan and leading to environmental impact through hardware replacement. In particular, 37% of users declare that they did not attempt to repair their device when a failure occurred, including battery failures[1]. In such a situation, the whole device is replaced, increasing environmental impacts. Similarly, high data usage causes network congestion, prompting service providers to upgrade infrastructure, further contributing to environmental concerns. Therefore, both battery and data usage impact the environment directly through energy consumption and indirectly through hardware life cycles. While this impact remains limited for a single user, it scales proportionally to the number of users and their daily usage of the application. Such impacts are in part responsible for the increasing share of Information and Communication Technologies in humanity's carbon emissions[2]. Recognizing this issue, some countries are adapting their legal frameworks to limit such impacts, *e.g.*, France with the REEN laws[3]. Therefore, digital factories face increasing pressure from users, developers, and legal frameworks to assess and reduce the environmental impact of their software.

Greenspector offers a solution to assess the energy consumption and environmental impact of mobile applications and websites. This solution takes software to review as input and provides results in the form of raw measures of energy and data, a grade, and an estimation of environmental impacts regarding a set of planetary limits.

This document outlines Greenspector's approach to evaluating the environmental impact of software. It begins by introducing the measuring framework employed to automate the assessment of the software being examined. Then, it presents the grading system and the process by which raw performance data is transformed into a singular grade. Lastly, it introduces the impact model and the methodologies employed to address the inherent uncertainty associated with such models.

---

[1]European Commission, Agriculture Consumers, Health, Food Executive Agency, C Duke, et. al.. 2018. Behavioural study on consumers' engagement in the circular economy – Final report. CEU. https://doi.org/10.2818/956512

[2]Xiaoyong Zhou, Dequn Zhou, Qunwei Wang, and Bin Su. 2019. How information and communication technology drives carbon emissions: A sector-level analysis for China. Energy Economics 81 (2019), 380–392.

[3]LegiFrance. 2021. LOI n° 2021-1485 du 15 novembre 2021 visant à réduire l'empreinte environnementale du numérique en France. https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000044327272

2. Measure

This section introduces a framework to measure an application's battery and data usage. This framework contains four components. First, a tool allows for creating user journeys replicating human usage of the application. Then, a measure protocol is defined to monitor the application's performance during such user journeys. The results of such measures are then centralized and processed in a data visualization platform, Greenspector Studio. The final component of this framework is a control tool, Greenspector CLI, used to launch the execution of the test suites. This section details the usage and capabilities of such components.

2.1. User journey

Modern applications contain a collection of features, or functional units, each of them having varying performances. It is thus not possible to reduce the performance of a software to a single indicator, and each functional unit must be independently evaluated to exhaustively assess an application[4]. Therefore, all applications assessed by Greenspector are split into a set of functional units, referred to as *user journeys*. Such journeys aim to reproduce the behavior of an actual user and can include steps such as opening the application, interacting with the content, and navigating through views. Automating such user journeys is essential to guarantee the reproducibility of the measurements. Therefore, they are implemented in an abstract declaration language, GDSL, representing a series of

```
measureStart,CHRGT_home
applicationStart,com.myapp
waitUntilText,Default
pause,${PAUSEAFTERLOAD}
measureStop

measureStart,PAUSE_home
pause,${PAUSEDURATION}
measureStop

measureStart,CHRGT_addEntry
clickByTextExact,Add
waitUntilText,Enter new name
pause,${PAUSEAFTERLOAD}
measureStop

measureStart,ACTION_addEntry
enterText,Example text
clickByText,SAVE
waitUntilText,Example text
pause,${PAUSEAFTERLOAD}
measureStop

applicationCloseAll
```

[4] ISO 14044:2006. Environmental management, Life cycle assessment, Requirements and guidelines https://www.iso.org/standard/38498.html

actions performed sequentially on the device. GDSL supports Android and iOS and includes basic actions such as wait, click, or pause, as well as more complex actions such as launching an application or enabling the GPS. GSDL also offers commands to start and stop performance measures, providing a fine granularity of the metrics, and allowing for assessing each step of the journey, both w.r.t energy and data usage. An excerpt of a basic GDSL script is shown in Figure 1. This script assesses an application called com.myapp, first when launching the application (lines 1 to 5), while idling on the home activity (lines 7 to 9), then during navigation to a different view (lines 11-15), and finally while the user performs an action (line 17 to 22). Delays are added during the journey to mimic the behavior of a human user. Additional commands are performed before the start of the script to clear data of the application, set luminosity to a predefined level, or measure idle power usage. Similarly, additional measures could be performed during this journey to offer a higher performance granularity.

GDSL scripts are stored on the device during the tests and are executed by a package on the device. While this package introduces a risk of affecting the performance of the device, its effect would be consistent for all scripts, and its power usage is included in the idle power usage of the system. This package does not require data transfers during the test and thus does not affect data consumption. Finally, GDSL can interact with applications but also with Web-Views in such applications. Therefore, the framework can be used to assess the performance of both applications and websites accessed through the smartphone browser. In this scenario, the monitored application is the browser, while the user journey represents the use of the website under review.

2.2. Measure protocol

While executing the application on virtual devices would provide metrics regarding the data usage, their energy consumption can not be assumed to properly replicate physical devices. Therefore, all performance assessments are performed on actual devices. Such devices are stabilized before each measure. This stabilization ensures that the battery is at a target level, to avoid non-linear discharge speeds and that the idle power and CPU usage meet expectations. Background services are also monitored to limit their impact. While this stabilization may increase the delay before obtaining results, it ensures high reproducibility and quality for such results.

The smartphone model is fixed for each application, ensuring that physical differences across models do not cause variations in performance. Finally, each user journey is repeated over a predefined number of iterations when the tests are launched. The energy and data usage, and other metrics regarding the state of the device, are monitored through tools such as the Android Debug Bridge[5] on Android, tcpdump for network packet, and external tools on iOS.

[5]Android. 2023. Android Debug Bridge (adb). https://developer.android.com/tools/adb

2.3. Results aggregation

All the metrics gathered during the measure are centralized in Greenspector Studio. This tool allows for monitoring the evolution of each user journey, and each of their respective steps, regarding energy and data usage. The overall grade of each journey is also provided in this tool. This grade is estimated by combining the energy consumption, data usage, and length of each user journey, according to the methodology introduced in Section . This grade is the criterion that determines whether the performance of the journey improved or not and can thus be used as a quality gate of a DevOps pipeline. In addition, the results of all tests are integrated into a dashboard, allowing for monitoring the temporal evolution of an application, as visible in Figure 2 for a given user journey. This dashboard is relevant during the "monitor" and "plan" steps of the DevOps cycle, as it allows for identifying possible improvement and successful optimization.

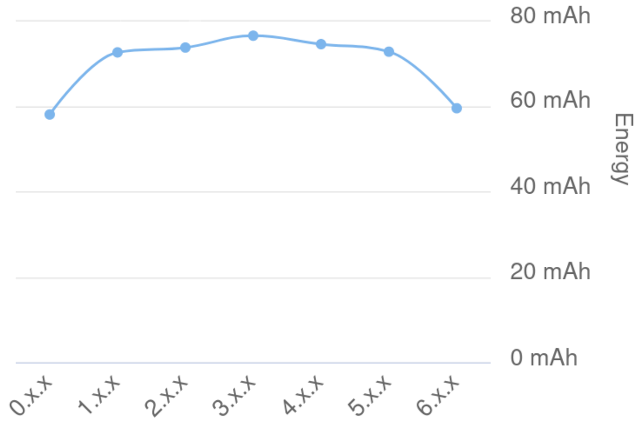

Figure 2: The power usage of a user journey over different versions of an application

2.4. Integration

To properly integrate in an existing test environment, such as Continuous Integration, the process of launching a test has been simplified to a few commands given to a command line tool, Greenspector CLI. This tool must be installed on the device launching the tests. Figure 3 is an example of instructions to launch a test on a new version of an application. The first line declares a new version of the application MyApp in Greenspector Studio. The second line declares which type of device to use, e.g., a Samsung Galaxy S9 running Android 10. Finally, the third line runs the tests. The user journey to execute is test.gdsl, the package to monitor is com.myapp, and Three iterations of the test will be performed. While all results are integrated to the project's dashboard in Greenspector Studio, it is also possible to fetch the Ecoscore directly from the CI/CD pipeline with the help of a dedicated Docker image. This result allows for using the Ecoscore as a quality gate in such a pipeline.

2.5. Discussions

The implementation of this framework is summarized in Figure x. The application and its GDSL scripts to assess are sent from the user's CI/CD to Greenspector's systems through the CLI. They are then sent to our Test bench, which acts as a load balancer and ensures the stabilization of the devices. The tests are executed on the target physical device. The resulting data is gathered by the Test bench and analyzed on the Core server. The resulting Ecoscore can be sent to the CI/CD as a quality gate. The detailed results are made available in Greenspector Studio. These results include the details of the Ecoscore, all the metrics that were monitored during testing as a dashboard, and the environmental impact estimated by our impact model.

Greenspector measure framework helps in monitoring the environmental impact of mobile applications and websites throughout their development. However, it also allows for benchmarking a collection of applications over a similar user journey. Indeed, when several applications offer a similar feature, it is possible to compare the performance of similar user journeys across such applications, to identify which one has the least environmental impact, and thus to recommend this application over the other options. Such a usage of this framework allows for comparing different implementations of a given application. For instance, in Frattaroli et al.[6], the authors developed a functionally identical application with five different development frameworks. After monitoring the battery and data usage and the size of such applications, the authors discuss the impact of each development framework on performance. Such usage of our framework can assist the decision process, both when designing new software services from scratch and when comparing possible implementations of a given feature during development. Such usage fits into the "monitor", "plan", and "code" steps of a DevOps methodology. This framework thus facilitates the integration of environmental considerations throughout various steps of the DevOps cycle, e.g., "monitor", "plan", "code", and "test". By doing so, it contributes toward a DevGreenOps methodology, akin to the DevSecOps approach that incorporates security considerations across all phases of the DevOps methodology.

However, this framework also has specific limitations, in particular due to the usage of physical devices. Similarly to cloud providers abstracting away the constraints of physical server management, this framework abstracts away mobile device management and thus meets similar challenges. Indeed, for a given application, some user journeys can span over several minutes, and such a cost in time will be scaled through the number of iterations of each test, the number of user journeys, and the number of applications to assess, while the pool of available devices remains limited. While the value of this framework relies specifically on the usage of physical devices, such an approach induces a risk of delays in automatic tests. Such delays can be temporarily mitigated by running the tests only on releases instead of commits, skipping such tests for emergency fixes, or deploying enough devices to absorb sudden spikes in test demand. However, such limitations could be tackled by improving load-balancing algorithms in future works.

A second limitation of this framework lies in the focus on the end-user's devices. The impacts of the application on server and network infrastructure are not measured but only estimated through data usage. However, other tools tackle this problem, in particular regarding server usage. The academic tool PowerAPI[7] offers the ability to measure the power usage of physical or virtual servers, while the

---

[6]Frattaroli, Vincent, Olivier Le Goaer, and Olivier Philippot. "Ecological Impact of Native versus Cross-Platform Mobile Apps: a Preliminary Study." *2023 38th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW)*. IEEE, 2023.

[7] https://powerapi.org/

industrial tools Scaphandre[8] and EasyVirt[9] focus on virtualized environments. Such tools are thus complementary to our framework.

Finally, while this approach allows for identifying regression in performance during the development of an application, it may be necessary to rely on additional tools to locate the root cause of such regressions. In that regard, static analysis tools are complementary to our framework.

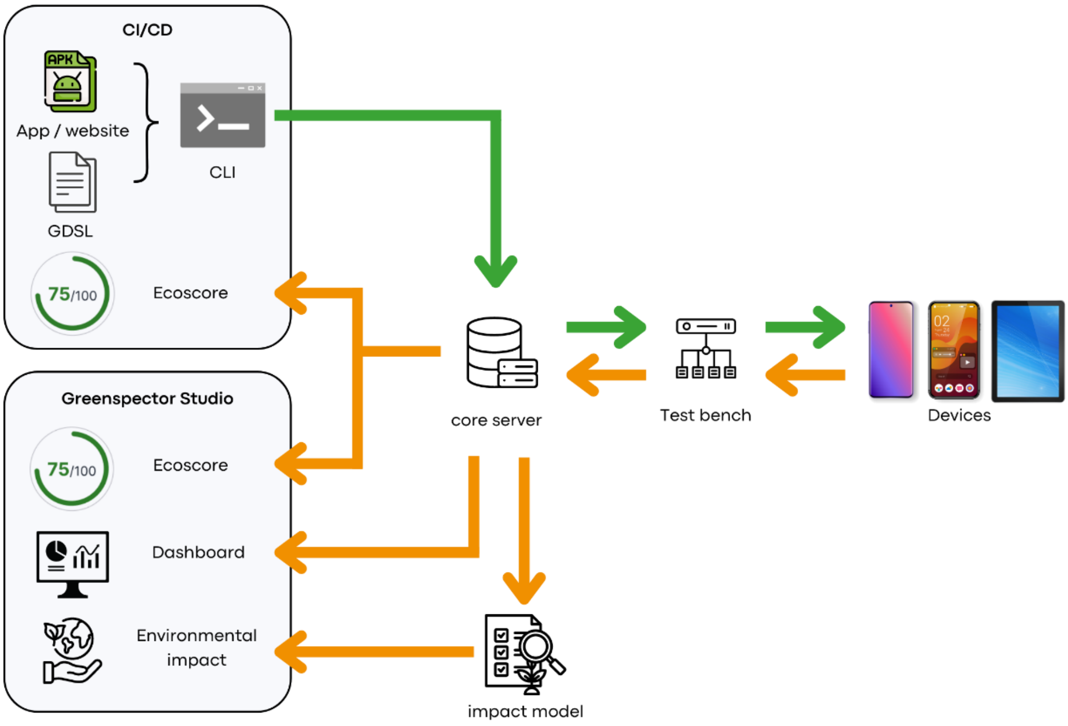

Figure 3 : Greenspector's measure framework

Greenspector's approach assesses the functional units of applications, rather than the applications themselves. The functional units are described in GDSL, an automation language, to replicate the behavior of a human user. The GDSL scripts and the application are sent to Greenspector's testbench with a command-line tool. The applications are systematically tested on stabilized physical devices. This automation allows for monitoring the performance of each functional unit throughout the development of the application. This approach is summarized in Figure 3 .

[8] https://github.com/hubblo-org/scaphandre

[9] https://www.easyvirt.com/en/

3. Grading

While total raw energy and data usage provide a transparent way of communicating the results, they do not allow for a comparison across user journeys of different lengths. Moreover because of their multiplicity, they are not the most convenient to monitor and communicate. This section details how such performance metrics are converted to a single grade, the Ecoscore. The Ecoscore is a grade, on a 0 to 100 scale, quantifying the quality of a given user journey.

This Ecoscore is calculated based on three criteria: the duration of the user journey in seconds, its data usage in Bytes, and its energy usage in Ampere-Hour. Each of such criterion is converted to a score on a scale from 0 to 100. The total Ecoscore is the average of the three corresponding scores. Greenspector rewards higher Ecoscores with a label.

3.1. Intermediary scores

The Ecoscore of a user journey is calculated from the performance of each individual step of such a journey. However, not all steps have the same weight in the overall performance. Steps are thus categorized as *critical* or *non-critical*. Specifically, critical steps are the most relevant in the functional value of the software, whereas non-critical steps are the ones with more limited relevance. The category of each step is determined during the creation of user journeys.

Each step of a user journey is assigned to one of five categories, representing its performance in each indicator. While the categories regarding duration and data usage are estimated using a conversion scale, the grades regarding energy rely on a different approach. Specifically, the grades do not represent the power usage of the device running the user journey, but rather the relative increase in powered usage compared to the idle power usage of the device, *i.e.*, its power usage when the software is not running. For instance, the best grade is obtained for steps increasing the power usage by less than 5%. Furthermore, the category-defining criteria changes depending on the type of step. Specifically, the scale differentiates between loading or action steps, expected to have higher data and energy usage, and pause steps, which may have more substantial durations. The detailed categories for each indicator are provided in Table 1. For instance, an action step executed in 1.5 seconds would be in category 2 regarding the duration indicator. A pause step during 1.5 would be in category 1 of this indicator.

Table 1 : The grading scale for each steps

Category	Duration		Data		Energy		Grade
	Loading action	Pause	Loading action	Pause	Loading action	Pause	
1	<1s	<2s	<5ko	0ko	<x2	< x1,05	1.00
2	<2s	<4s	<25ko	<5ko	<x3	< x1,50	0.75
3	<5s	<10s	<50ko	<10ko	<x4	< x2,00	0.50
4	<10s	<20s	<300ko	<25ko	<x5	< x3,00	0.25
5	≥ 10s	≥ 20s	≥ 300ko	≥ 25ko	≥ x5	≥ x3,00	0.00

The score of a user journey, for a given indicator i and a given category c (*i.e.*, critical or non-critical) is computed as the sum of each possible grade multiplied by the number of steps have such a grade, and then divided by the number of steps, as visible in Equation 1.

$$1. \quad G_i^c = \frac{\sum_{g \in grades} g \times s_g^c}{N_{steps}^c}$$

3.2. Total Ecoscore

The Total Ecoscore of a user journey is an aggregation of the intermediary scores of its steps regarding duration, data, and energy, and regarding critical and non-critical steps.

Specifically, the critical and non-critical scores in each indicator for all steps are first combined to a single score with a weighted average. Specifically, critical steps receive a weight of 0.7, and non-critical steps a weight of 0,3. This step thus provides three global scores regarding the duration, data, and energy consumption of the user journey, as visible in Figure 4. Then, the total Ecoscore of the journey under review is the average between such three global scores. This total Ecoscore is thus a grade between 0 et 100. Such a score allows for assigning a Greenspector eco-design label to the user journey. Specifically, an Ecoscore above 50 entitles to a bronze label, above 70 to silver label, and above 90 to a gold label.

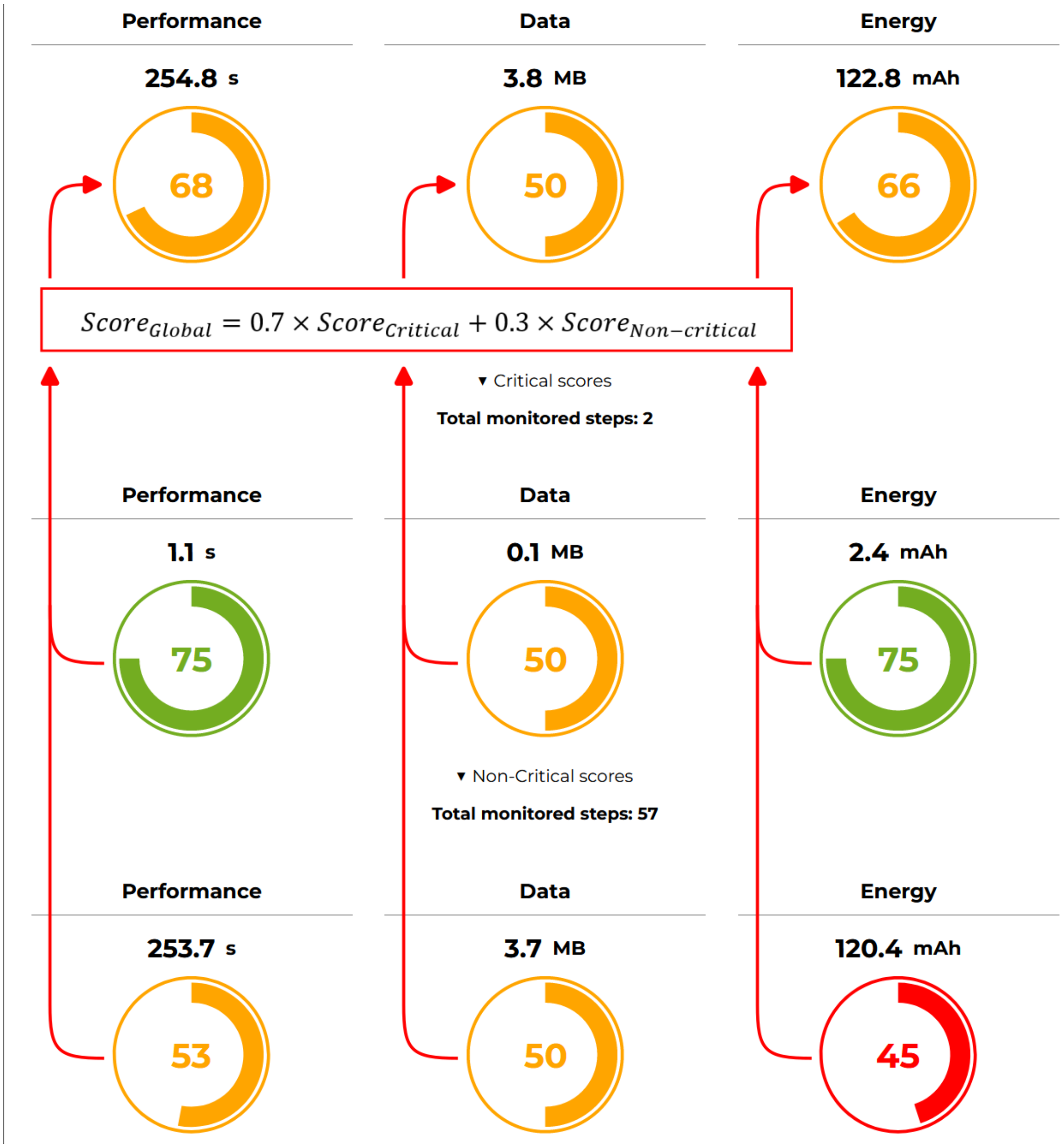

Figure 4: The calculation of the global score for each metric

3.3. Discussions

The Ecoscore presented in this section provides a means to easily assess, compare and communicate the eco-design of user journeys. Such a metric can be used to easily monitor the evolution of a user journey to detect and quantify regressions or improvements. Such a score can be used as a criterion for the quality gate of a DevOps pipeline. However, it is also possible to delves into the root cause of performance evolutions, positive or negative, by analyzing the evolution in each performance indicator of each individual step of the journey.

Nonetheless, the Ecoscore has limitations. In particular, the different steps of user journeys are graded into one of five categories regarding each performance indicator. The global scores are calculated from the number of steps in each category, and the respective weight of such categories. The limitations are two-folded. First, the thresholds between categories and the weight of each category are arbitrary. Second, the use of weighted categories causes thresholds in the global and total scores. For instance, a single-step user journey executed in 2 seconds would have a duration score of 100, whereas its score would drop to 75 with an execution time of 2.1 seconds. Furthermore, such variations can be caused by measurement imprecision rather than an actual evolution of journey under review.

Such limitations can be addressed using a mathematical function rather than arbitrary categories and weights. However, the specific shape of such a function would also be arbitrary, while affecting the clarity of result restitution. This remains an open problem that must be tackled in future work.

Each functional unit receives a grade reflecting their environmental impact. This grade is a combination of the duration, and data and energy usage of each step composing such functional units. This grade allows for identifying performance issues in the different functional units, and to locate the specific steps causing this issue.

4. Estimating environmental impacts

While energy and data usage monitored by Greenspector provide a means to assess the efficiency of a functional unit, it does not provide the full picture of its environmental impacts. Indeed, contextual aspects such as the electricity mix on which the software relies, or the scale of back-end infrastructures, can affect the impacts of the software under review. This section introduces the modelling used by Greenspector to assess such impacts.

4.1. Quality & uncertainty

Greenspector's impact model leverages data quality indicators (DQI) to assess the relevance of its sources of secondary data, and fuzzy logic to capture and propagate uncertainties within ICT-related LCA. This section overviews DQI and fuzzy logic.

4.1.1.1. Defining Data Quality Indicators

ICT-related environmental assessments encounter significant inaccuracies stemming from reference data sources. Such sources, referred to as LCI secondary data in LCA terminology, may report diverging estimates for the same variable and do not have a consistent quality. A large variety of sources representing the state of the art must be accounted for, but their relative weight within the results should be different depending on their quality. To quantify this quality, each LCI source is assessed with a Data Quality Indicator (DQI), following the method introduced by Weidema et al.[10]. Specifically, the DQI of a source covers 3 key aspects: 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. Such obsolescence is caused by both the improvements of estimation methods, as well as changes in the manufacturing and production methods over time. Finally, 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. 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 2 maps the possible values for each category to the corresponding quality indicator. The total DQI of a 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.

Table 2: The criteria to assess the DQI of a source

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-reviewed manufacturer data

4.1.1.2. Propagating impact factors uncertainty

Different secondary sources can yield significantly varying results for the same device. For instance, the embodied impact of a smartphone can be 32.8, 57, or up to 93.5 kgCO2e, depending on the model

[10] Weidema, Bo Pedersen, and Marianne Suhr Wesnæs. "Data quality management for life cycle inventories—an example of using data quality indicators." *Journal of cleaner production* 4.3-4 (1996): 167-174.

and the manufacturer providing the information[11] [12] [13]. Such variations can significantly influence the final estimated impacts and should be propagated within all computations to be exposed in the final estimation. While multiple sources should be considered to capture a more comprehensive reference impact, averaging these values can lead to errors. Extreme values are not inherently incorrect and would not be captured by an average value. Thus, each source should be weighed 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 Weckenmann et al.[14]. In fuzzy logic, variables are not defined by a strict value in $\mathbb{R}$, but rather by a function $\mu_s\colon \mathbb{R} \to 0..1$capturing the degree of membership of a value with a given fuzzy set s. A membership degree of 1 indicates the certainty that a value of x is possible, whereas a membership degree of 0 reflects that the fuzzy sets does not cover this value. Given this definition, two crisp sets are of interest: the core captures the range of values 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 piecewise continuous membership function. As such, they act as fuzzy intervals. This paper only considers Trapezoidal Fuzzy Numbers (TFN) as they allow for a compromise between the complexity and precision of calculations. For fuzzy numbers with a membership function defined as a trapezoidal shape, 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 TFN fuzzy set L, m_L, m_R, R. Then, Weckenmann et al. computes the TFN for any set of sampled points with Equations 1–4, with $\overline{x}$ representing the weighted average of the sampled variable, and C_v the coefficient of variation.

1. $$m_L = \frac{\overline{x}}{1+(0.5\times C_v)}$$
2. $$m_R = \overline{x} \times \left(1 + (0.5 \times C_v)\right)$$
3. $$L = m_L - \overline{x} \times \left(\frac{1}{1+(0.5\times C_v)} - \frac{1}{1+(2.5\times C_v)}\right)$$
4. $$R = m_R - (\overline{x} \times 2 \times C_v)$$

[11] Fairphone, Life cycle assessment of the fairphone 3 (2020). https://www.fairphone.com/wp-content/uploads/2020/07/Fairphone_3_LCA.pdf

[12] Ercan, Mine, et al. "Life cycle assessment of a smartphone." *ICT for Sustainability 2016*. Atlantis Press, 2016.

[13] Apple, iphone 6 plus environmental report (2014). https://www.apple.com/environment/reports/docs/iPhone6Plus PER Sept2014.pdf

[14] Weckenmann, Albert, and Achim Schwan. "Environmental life cycle assessment with support of fuzzy-sets." *The International Journal of Life Cycle Assessment* 6 (2001): 13-18.

To illustrate such intervals,

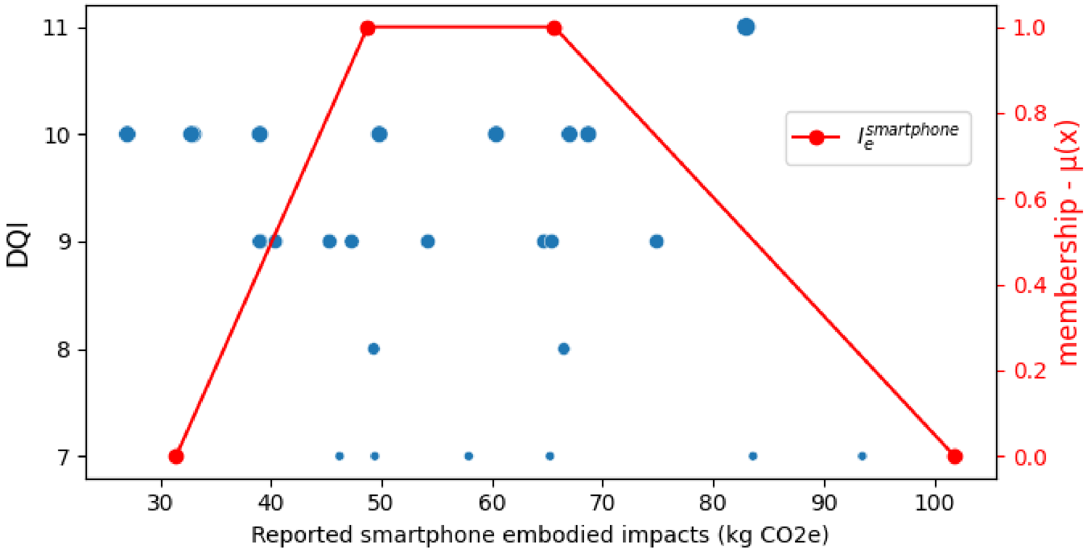

Figure 5 depicts the TFN capturing the embodied impact of a smartphone. To account for quality variations in secondary sources, the main vertical axis is the DQI of each estimated impact in the aggregated secondary sources. Then, weighted secondary sources are converted to a TFN, visible is on the secondary vertical axis, with a support ranging from 31 to 102 kgCO2e, and a core between 48 and 65 kgCO2e. Therefore, x and Cv account for both variations in sources regarding a variable, but also variations in quality. 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 TFN as an appropriate structure for capturing uncertainty of estimations at large.

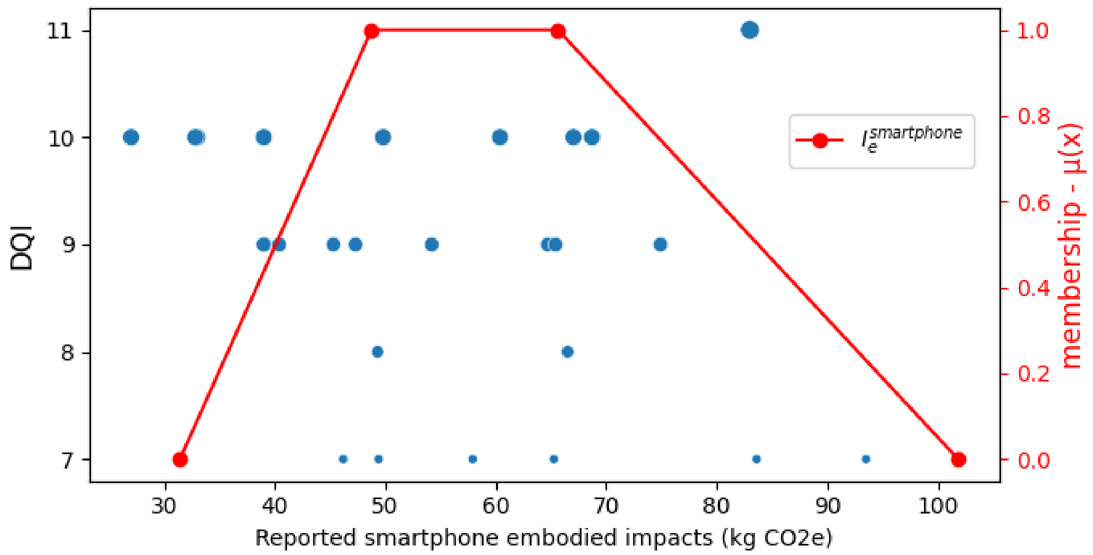

Figure 5: Building the embodied impact factor of a smartphone, as a TFN inferred from 24 secondary sources weighted by their DQI.

Fuzzy logic supports arithmetic operations between fuzzy sets—i.e., additions, subtractions, divisions, and multiplication. For instance, the sum of the sets $[a_1, a_2, a_3, a_4]$and$[b_1, b_2, b_3, b_4]$is $[a_1 + b_1, a_2 + b_2, a_3 + b_3, a_4 + b_4]$. Subtractions apply similarly, but division and multiplication require more advanced tools. However, multiplication can still be approximated with the method presented above, while in divisions, the divisor is reverted to approximate the result—*i.e.*, $[a_1/b_4\,, a_2/b_3\,, a_3/b_2\,, a_4/b_1]$[15]. Furthermore, any number x ∈ ℝ can be converted to the fuzzy set [x, x, x, x] to mix real numbers and fuzzy sets. Thus, the result of an equation containing fuzzy sets is a fuzzy set[16]. Since a fuzzy set contains both the value and the uncertainty of any hypothesis, arithmetic operations propagate uncertainty throughout all steps. The results convey all the possible estimations and uncertainties and, therefore, environmental assessments performed with fuzzy logic do not require to define scenarios, such as best or worst-case scenarios, or time-costly simulations that need additional hypotheses.

To facilitate the communication of the final results, the obtained fuzzy numbers can be converted to a more straightforward format. Notably, environmental assessments are generally communicated as either a singular result with a margin of uncertainty, or as a best, worst, and average scenario, *i.e.,* a set of three results. While different approaches implement such a defuzzification[17], the most straightforward approach with TFN is the central value, using the two points where the membership function has a value of 0.5 as the margin of error, and the average between these two points as the centrale value.

The overall process of obtaining and qualifying data is summarized in Figure 6. The hypotheses of the model and their respective DQI are extracted from the academic and industrial bibliography and converted to a set of fuzzy numbers. Such fuzzy numbers are provided to our impact model, described below, to obtain an estimated impact in the form of the fuzzy number. This fuzzy number is then defuzzified to return an impact in the form of a single value and a margin of uncertainty.

---

[15]moiGrzegorzewski, Przemyslaw, and Karolina Pasternak-Winiarska. "Trapezoidal approximations of fuzzy numbers with restrictions on the support and core." *Proceedings of the 7th conference of the European Society for Fuzzy Logic and Technology*. Atlantis Press, 2011.

[16]Cheng, Xin, and Simon Li. "Interval estimations of building heating energy consumption using the degree-day method and fuzzy numbers." *Buildings* 8.2 (2018): 21.

[17] Van Leekwijck, Werner, and Etienne E. Kerre. «Defuzzification: criteria and classification.» Fuzzy sets and systems 108.2 (1999): 159-178.

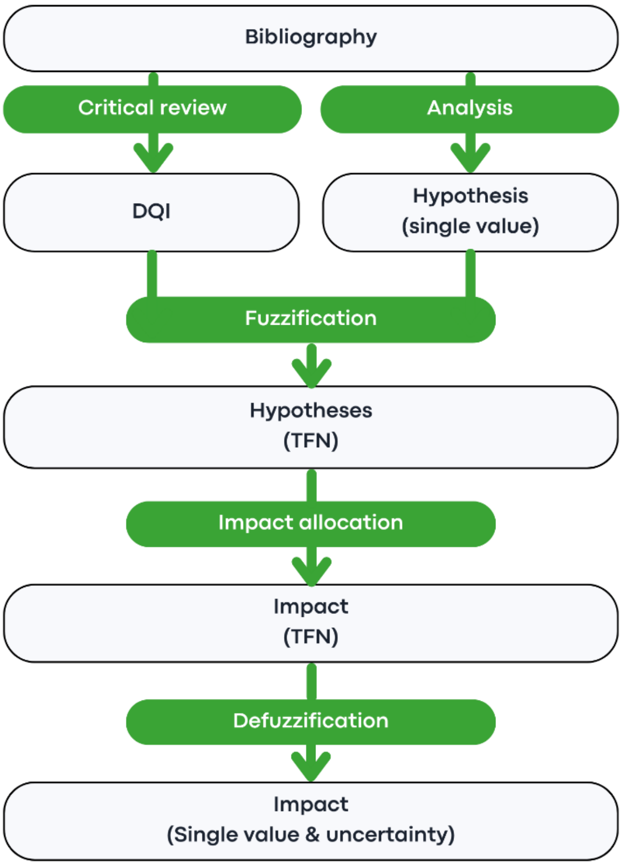

Figure 6: from bibliography to impact with uncertainty

4.2. Impact model

Following LCA methodology to estimate the environmental impact of software services, such as mobile apps and websites, the analysis is carried out for a defined functional unit. The functional unit provides a reference to which inputs (data collected) and outputs (environmental impacts) are related or normalized[18]. Consequently, data collection involves a realistic user journey—i.e., a sequence of actions on the application capturing the usage patterns on the reviewed functional unit. To more accurately capture uncertainties in properties beyond software measures, such as intrinsic network impacts, each of such properties is represented as a fuzzy set constructed, either from estimations drawn from a collection of sources or from measured energy and data usage. Ultimately, the impact assessment phase—where inventory data is translated into an environmental impact—employs the reference data source fuzzy sets and DQI detailed in Section .

---

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

In the following, we adopt a 3-tier architecture—covering end-user devices, networks, and back-ends—to estimate the environmental footprint of ICT services. For each of these tiers, hypotheses and their associated uncertainties are proposed to conduct estimations without requiring extensive knowledge of their technical layout. For a given functional unit capturing a user journey, the analysis covers the application usage on the user's device, the network usage resulting from data transfers during the user journey, and the usage of remote servers to process and store such data. The impacts of these 3 layers are estimated from 3 separate components, and the total impact generated by the user journey under review is thus computed as their sum.

The embodied impacts of ICT devices—arising from manufacturing, raw materials extraction, transport, end of life—can surpass their usage impacts. Consequently, both the embodied and usage impacts are accounted for as illustrated in Figure 7. The embodied impact is depreciated over the lifespan and usage of the related hardware components, for example as the total time spent using a device or the number of requests handled by network equipment. We describe this depreciation process per component.

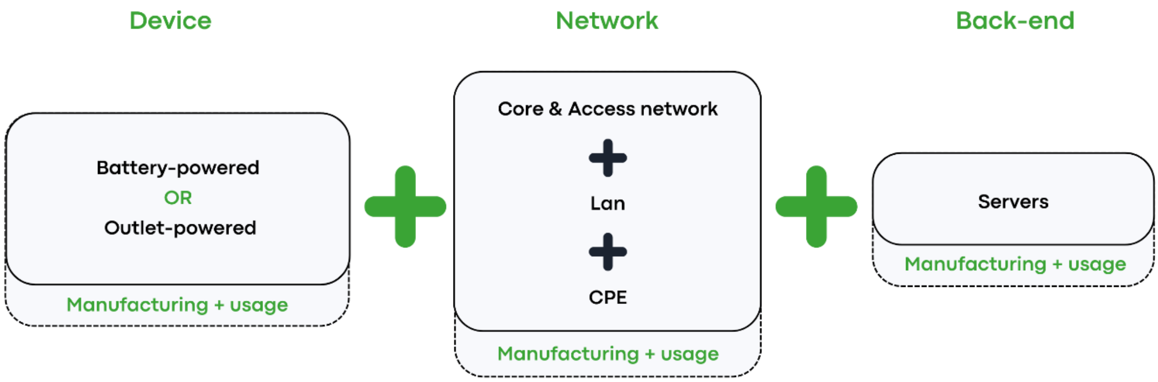

Figure 7: the scopes accounted for in the environmental assessment

The LCA methodology requires to consider multiple impact categories to build a comprehensive analysis of the environmental impacts associated with an ICT service. A description of those proposed by Product Environmental Footprint (PEF) recommendation[19], along with their respective units, is presented in Table 3. It is important to note that reference environmental impact data for the rapidly evolving ICT sector remains scarce and is often restricted to a single impact category, namely climate change expressed in kg CO2e.

Table 3: Impact categories supported by the model

PEF impact factor	Impact unit	Domain
Photochemical ozone formation	kg NMVOCe	Human health

[19]Council of European Union, Commission recommendation (eu) 2021/2279 on the use of the environmental footprint methods to measure and communicate the life cycle environmental performance of products and organisations (2021).

Particulate matter	disease incidence	Human health (respiratory issues)
Ionizing radiation	kBq U235e	Human health (cancer)
Climate change	kg CO2e	Climate change
Acidification	mol H+e	Water and soil acidification
Mineral & metals resource use	kg Sbe	Abiotic resources depletion
Fossils resource use	MJ	Abiotic resources depletion
Freshwater ecotoxicity	CTUe	Ecosystems

In the remainder of this document, environmental impacts are expressed in an abstract unit, referred to as *impact unit*. To compute the impact of a software service in each category, the impact unit should be replaced by the effective unit associated with the impact category under consideration.

The fuzzy set F_{em} represents the impact of a worldwide electricity mix. However, it is possible to replace this default set with the electricity mix of a given country, or subset of countries. Furthermore, the impact of the electricity mix can vary across components to better represent the geographic dispersion of the different tiers. For instance, the end-user devices and network infrastructures may rely on the worldwide electricity mix, while the back-end infrastructures only use the electricity mix of the country where servers are hosted.

4.2.1. Modeling End-user device impacts

ICT services rely extensively on ICT devices, which can be powered either by batteries or electrical outlets. Consequently, the impact of such devices is computed through distinct hypotheses and computation formulas.

4.2.2. Outlet-powered devices

The embodied impact of outlet-powered devices is distributed over their usage time, at the rate of their daily usage over their life expectancy. Usage impact is computed based on the power consumed by the device during the user journey, regarding the location-based electricity mix.

Table 4 presents the variables needed to estimate the impacts of outlet-powered devices. Each variable is adapted to represent the specific type of device under study, such as desktop PCs, Consoles, TVs, or set-up boxes. Embodied impact (I_e^{device}) encompasses the impacts caused by raw material extraction, product manufacturing, transportation, and disposal or reuse. Life expectancy (L) represents the number of years of usage, and daily usage time (U_d) represents the number of hours the device is used daily, while user journey duration (T) corresponds to the time required to perform the functional unit. Power (P) is the power usage of the device. Finally, the electricity-mix impact factor (F_{em}) represents the environmental impacts associated with energy production and transport. An impact factor F_x gives an environmental impact per functional unit, such as impact unit/Joules in this case.

Table 4 : Variables of the outlet device model, per type of device

Variables	Unit
Embodied Impact (I_e^{device})	Impact unit
Lifespan (L)	Seconds
Daily usage time (U_d)	Hours
User journey duration (T)	Seconds
Average power ($\overline{P}$)	Watts
Electricity-mix impact (F_{em})	Impact unit per joule

All such variables are fuzzy sets to capture and propagate their associated uncertainty and can be refined by experts based on their system knowledge. When a hypothesis is refined, its fuzzy set can ultimately be replaced with a single value. For instance, a company using the software under review on company-owned devices can use precise values for life expectancy, daily usage time, and usage time.

The impact induced by a user journey on an outlet-powered device is computed from a share of its embodied impact I_e^{device} imputed to the user journey, and the impact of the device consumption during this journey. Then, Equation 5 models the device's embodied impact imputed to software F_e^{device}, expressed in impact unit per second of usage. This factor is estimated as the depreciation of the embodied impact I_e^{device} over the life expectancy of the device L, at the rate of the device's daily usage time U_d. The software usage impact factor F_u^{device} is computed in Equation 6. The formula first scales down the electricity-mix global impact factor F_{em} P. The total impact of the device I^{device} is finally estimated in Equation 7 as the sum of the embodied and usage impacts attributed to the application for the duration of the user journey T. To better represent an average user journey, the total impact is the sum of the total impact of each type $d$of outlet-powered device, prorata their respective share of the audience $S(d)$. For instance, a share of the audience may watch a streamed video from a desktop, while others watch it from a TV.

5. $$F_e^{device} = \frac{I_e^{device} \times 24}{L \times U_d}$$

6.

7. $$I^{device} = \sum_{d \in devices} \left(F_{e(d)}^{device} + F_{u(d)}^{device}\right) \times T \times S(d)$$

4.2.3. Battery Powered devices

Unlike outlet-powered devices, battery-powered devices, such as smartphones, tablets, or laptops have a lifespan closely tied to their usage. Indeed, charging a battery diminishes its capacity, implying that a battery can only undergo a limited number of charge cycles before its capacity becomes unusable, mandating users to either replace either the battery or the entire device. Therefore, our hypothesis assumes that the greater the software drains the battery, the higher its environmental impact is. The embodied impact of the device is thus allocated across the total energy capacity that the device can hold over its lifespan.

Table 5 introduces the variables to model battery-powered devices. This hypothesis covers different types of devices, such as smartphones, tablets, and laptops, with different properties. Thus, all such variables are only applicable to a given type of device and must be duplicated according to the number of device types to consider. For instance, the average battery capacity of a smartphone $B_{cap}^{smartphone}$ is lower than the average battery capacity of a tablet B_{cap}^{tablet} . The battery embodied impact $I_e^{bat.}$ captures the various impacts of the battery (incl. manufacture, transport), and is also included in the device embodied impact I_e^{device}. The maximum number of battery cycles C_{max} counts the maximum complete charges that the battery can sustain while remaining usable. The battery capacity B_{cap} and battery voltage $V$are used to quantify the drainage of the battery, regarding battery usage E, which is measured in a controlled environment. Meanwhile, the charger efficiency C is used to assess the actual energy usage of the device. Then, the battery-to-device replacement ratio R quantifies how frequently a user opts to replace the battery instead of the whole device when the maximum number of cycles C_{max} is reached. R is the average number of replacements that a user is willing to perform in that situation.

Table 5: Variables of the battery device model, per type of device

Variables	Unit
Measured device discharge (E_d)	Amp-hour
Device embodied impact (battery included) (I_e^{device})	Impact unit
Battery embodied impact ($I_e^{bat.}$)	Impact unit
Maximum battery cycle (C_{max})	Cycles
Battery Voltage (V)	Volts
Battery capacity (B_{cap})	Amp-hour
Charger efficiency (C)	%
Battery-to-device replacement ratio (R)	%
Average batteries replacement ($\overline{R}$)	/
Share of users (S)	%

The primary assumption of this hypothesis is that the battery of the device has a finite number of cycles, and therefore a limited lifespan. When this lifespan is reached, the user will either replace the battery, with the probability R, or the whole device, with the probability of $1 - R$. Thus, the embodied impact of a given type of device (including its battery) $I_e^{bat.}$ is depreciated over the total quantity of energy that the battery can hold in its lifespan, $C_{max} \times B_{cap}$, as presented in Equation 8. However, when the battery is replaced (R), its embodied impact is fully depreciated over its lifespan, but only a share of the embodied impact of the remainder of the device—*i.e.*, $I_e^{device} - I_e^{bat.}$, is depreciated.

For instance, if the user replaces its battery once, the device will feature 2 batteries over its lifespan, so only half of the embodied impact of the device is depreciated over the lifespan of each battery. For users replacing the whole device, the embodied impact of I_e^{device} is depreciated on this total quantity of energy. However, for users only replacing their battery, the embodied impact of the battery $I_e^{bat.}$ is depreciated over its capacity, but the embodied impact of the remainder of the device, $I_e^{device} - I_e^{bat.}$, is only depreciated over the number of batteries it will contain—*i.e.*, n + 1 with n being the number of replacements.

The usage impact factor $F_u^{bat.}$ of a given device type is estimated using V the voltage of the battery and the electricity-mix impact F_{em}, while accounting for $C$the efficiency of the charger, as reported in Equation 9. Both Equation 8 and Equation 9 compute an impact factor per unit of electric charge. Thus, the sum of $F_e^{bat.}$ and $F_u^{bat.}$ is the total impact factor per unit of energy, which can then be multiplied by the measured electric discharge of the user journey E_d, as modeled in Equation 10. The total impact of the functional unit on end-user devices, I^{device} d of battery-powered device pro rata $S(d)$ their respective share of the audience

$$8. \quad F_e^{bat.} = \frac{R \times \left(I_e^{bat.} + \frac{I_e^{device} - I_e^{bat.}}{1+\overline{R}} \right) + (1-R) \times I_e^{device}}{C_{max} \times B_{cap}}$$

$$9. \quad F_u^{bat.} = \frac{V \times F_{em}}{C}$$

$$10. \; I^{device} = \sum_{d \in devices_d^E} \left(F_{e(d)}^{bat.} + F_{u(d)}^{bat.} \right) \times S(d)$$

4.2.4. Modeling Network Layers impacts

The network tier is composed of heterogeneous layers. The core network represents the internal network of the network service provider, while the access network is the infrastructure allowing end-users to reach this network. In addition, the Local Area Network (LAN) of the user can be accounted for. Notably, in fiber or xDSL networks, the user is equipped with Customer-Premises Equipment (CPE), but not in GSM networks. To accurately model these different technical layouts, the impact of the core and access networks, the CPE and the LAN itself, their respective impact are computed separately.

To better capture an average user journey, a combination of various types of network connections (ADSL, fiber, mobile...) is considered. In contrast to end-user devices, the network impact is not estimated with regard to power usage. The geographic distribution of network components makes it challenging to precisely assess the overall consumption of a given request. Such impacts are thus estimated as an impact per unit of transmitted data.

4.2.4.1. Core & Access networks

Table 6: Variables of the network mode, per network type

Variables	Unit
Device data transfer (D)	GB
Network type share (S)	%
Access network – Usage impact (F_u^{access})	Wh/GB
Access network – Embodied impact (F_e^{access})	Impact unit/GB
Core network – Usage impact (F_u^{core})	Wh/GB
Core network – Embodied impact (F_e^{core})	Impact unit/GB
Network bandwidth (B_{net})	GB/s
CPE – Average power usage ($\overline{P}_{CPE}$)	Watts
CPE – Embodied impact (I_e^{CPE})	Impact unit
CPE – Daily usage (U_{CPE})	Seconds

Embodied impacts are separately accounted for access F_e^{access} and core networks F_e^{core}, and usage impacts with F_u^{access} and F_u^{core}, respectively. The combined embodied impact for both core and access networks F_u^{can}, as impact unit per unit of data transmitted, is computed in Equation 11. Similarly, the total usage impact for both networks F_e^{can} is computed as the sum of energy consumption per data transmitted, converted into the relevant impact factor using the electricity mix emission F_{em}, in Equation 12. Finally in Equation 13 the resulting embodied and usage impact per data transmitted for a given network n is multiplied by the amount of data transmitted by the software D, to obtain I^{can} the total impact of the core and access network.

$$11.\ F_e^{can} = F_e^{core} + F_e^{access}$$
$$12.\ F_u^{can} = (F_u^{core} + F_u^{access}) \times F_{em}$$
$$13.\ I^{can} = (F_e^{can} + F_u^{can}) \times D$$

4.2.4.2. CPE

Wired connections—i.e., fiber or xDSL—rely on Customer-Premise Equipment (CPE), such as a modem or optical network terminal. In contrast to the core and access networks, the power usage of the CPE can be empirically assessed. As such devices are outlet-powered, the impact of a CPE, denoted as I^{CPE}, can be estimated using Equation 5, Equation 6 and Equation 7, where I_e^{device}, U_d, and $\overline{P}$ are replaced by I^{CPE}, U_{CPE}, and $\overline{P}_{CPE}$, respectively

4.2.4.3. LAN

Table 7: Variables of the LAN model, per LAN equipment

Variables	Unit
Access point bandwidth (B_{net})	GB/s
Average power usage ($\overline{P}$)	W
LAN bandwidth (B)	GB/s
Embodied impact (I_e^{lan})	Impact unit
Lifespan (L)	Seconds

Finally, the LAN of the user is modeled as a set of equipment including firewalls, switches, and WiFi access points. As for the CPE, the LAN impacts are quantified wrt. a usage time. The associated variables are listed in Table 7. The embodied impact of the devices is depreciated over their average lifespan, prorata their usage ratio in Equation 14, by providing a depreciation in impact factor per unit of time. Similarly, the sum of energy consumption of all the LAN components is converted into impact factor per unit of time by reusing Equation 6. The resulting embodied and usage impacts per unit of time are then summed and multiplied by the usage duration—i.e., the transmitted data D divided by the network speed—in Equation 15.

$$14.\ F_e^{lan} = \frac{I_e^{lan} \times B_{net}}{L \times B}$$
$$15.\ I^{lan} = \frac{D}{B_{net}} \times \sum_{q \in eq}\left(F_{e(q)}^{lan} + F_{u(q)}^{lan}\right)$$

4.2.4.4. TOTAL network impact

The total impact of the network $I^{network}$ is then computed in Equation 16 as the sum of impacts of the core and access networks, the CPE, and the LAN for all network types, prorata the share of users behind such network. For networks without CPE, I_n^{CPE} is 0, while software only used within a company may have a network mix of 100% fiber, with CPE and LAN. Contrarily, software used by users on their own device use a network mix, such as 50% 5G, no CPE and no LAN, and 50% fiber, with CPE and no LAN.

$$16.\ I^{network} = \sum_{n \in network}\left(I_n^{can} + I_n^{cpe} + I_n^{lan}\right) \times S(n)$$

4.2.5. Modeling back-end Infrastructures impacts

Table 8: Variables of the back-end model

Variables	Unit
Request count (N_r)	/
Server max requests per second (N_{rps})	/
Server embodied impactI_e^{server})	Impact unit
Server lifespan (L)	Seconds
Server average usage (load) (U)	/ (%)
Average power usage ($\overline{P}$)	Watts
Power usage efficiency (PUE)	/

The back-end tier estimates the environmental impact of servers with regard to the requests executed by the software during a given user journey.

I_e^{server} the embodied impact of a server, which is depreciated over the maximum number of requests that this server will handle throughout its lifetime $L \times N_{rps} \times U$, to obtain an embodied impact unit per request handled, $F_e^{backend}$.

To assess the usage impact of servers, the usage impact factor per request $F_u^{backend}$is computed in Equation 18. It is the total impact per second $\overline{P} \times PUE \times F_{em}$ of the server, divided by the average number of requests handled every second $N_{rps} \times U$.

Finally, to estimate the server's total impact, the embodied and usage impacts per request are multiplied by the number of requests performed during the user journey N_r in Equation 19.

$$17.\ F_e^{backend} = \frac{I_e^{server}}{L \times N_r}$$

$$18.\ F_u^{backend} = \frac{\overline{P} \times PUE \times F_{em}}{N_{rps} \times U}$$

$$19.\ I^{backend} = \left(F_e^{backend} + F_u^{backend}\right) \times N_r$$

4.2.6. Combining impact models

Finally, the total impact of a single execution of the user journey under review, I, is computed in Equation 20 as the sum of the devices, network, and back-end impacts induced by the user journey—*i.e.*, the functional unit.

$$20.\ I = I^{device} + I^{network} + I^{backend}$$

While this value represents the average environmental impact of a single execution of a user journey, it can be multiplied by the total number of executions by all users over a period of time to be compliant with ICT services functional unit, as illustrated in Figure 8.

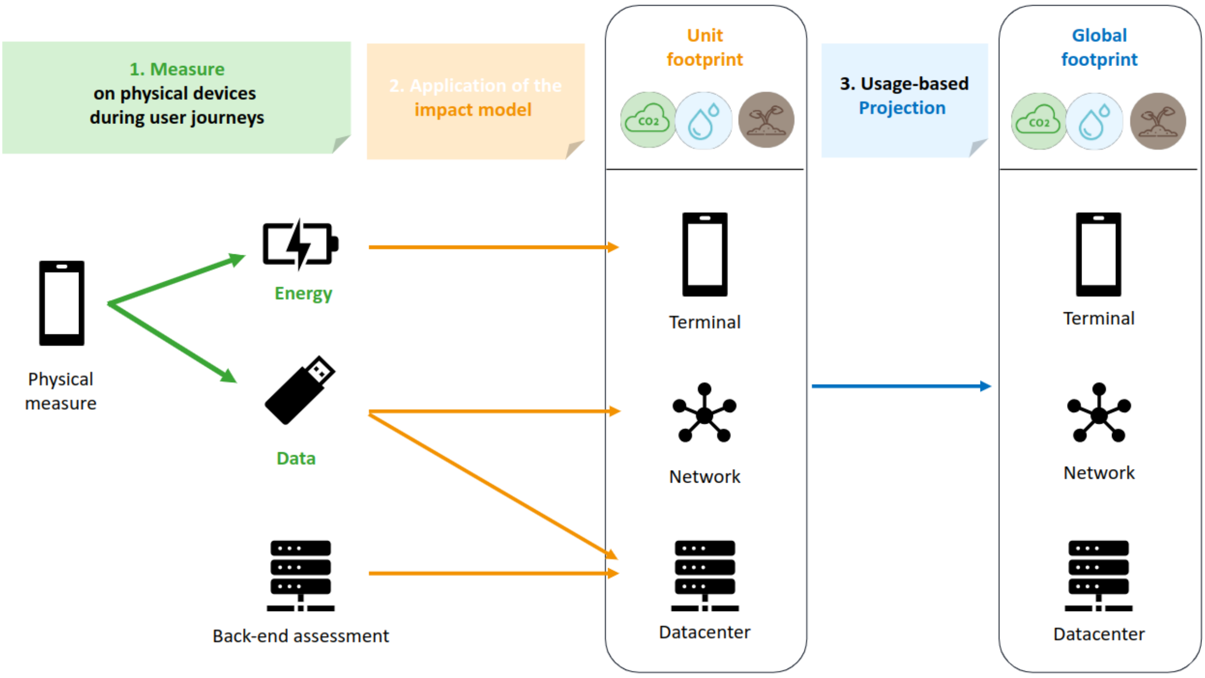

Figure 8: from measure to global impact

4.3. Discussion

From a scientific perspective, the LCA method lacks empirical validation regarding the overall result[20]. Therefore, even if using fuzzy logic in ICT services LCA offers a systematic approach to evaluate and propagate uncertainties, the outcome keeps lacking empirical validation. Furthermore, it is also significantly influenced by the secondary-origin data used to build a set of hypotheses regarding the environmental impact of considered functional units. Ultimately, the quality of estimations is largely constrained by the quality of such sources and the hypothesis derived from them. In particular,

[20]Ciroth, Andreas. "Validation–The missing link in life cycle assessment. Towards pragmatic LCAs." *The International Journal of Life Cycle Assessment* 11.5 (2006): 295-297.

environmental impact data sources are still scarce, and certain impact categories are almost not quantified at all. For instance, to the best of our knowledge, very few sources address the environmental impacts of network infrastructures in other categories than climate change.

4.3.1. End-user's device

It is assumed that the lifespan of battery-powered devices is solely determined by the lifespan of their battery and that users replace their devices when the battery becomes unusable. However, such a hypothesis may overlook other factors influencing the decision to replace these devices. Specifically, hardware and software obsolescence are not accounted for. Battery-powered devices, such as older smartphones, can be replaced due to slowness when executing recent applications, outdated operating systems unsupported by newer applications, or the availability of newer models in the market.

Similarly, it is assumed that the lifespan of outlet-powered devices is fixed and independent of their usage. As the embodied impact of these devices is depreciated over their daily usage, increasing this value reduces the impact factors per unit of time. However, higher usage may also lead users to replace their devices earlier, thus increasing this impact factor. Such considerations are not modeled as they are particularly difficult to detect and quantify, while not directly related to the assessed software.

Due to these different hypotheses, the modeling of battery-powered and outlet-powered devices diverges. In low-impact electricity mixes, increasing the daily usage of battery-powered devices increases their total impact, whereas increasing the daily usage of outlet-powered devices diminishes their total impact. Such divergences require specific explanations when discussing the analysis outcomes.

4.3.2. Network

The network component relies on hypotheses regarding the imputed embodied impact and usage impact of such infrastructures. Such hypotheses are expressed as energy or impact units per amount of data transmitted and are drawn from the literature with no additional imputation formula. Indeed, the network is considered a black box. The exact network topology of an average user is not reasonably ascertainable, and thus such hypothesis can not be specified. Thus, the uncertainty of network impacts can not be reduced. The total uncertainty of the results remains high after specifying hypotheses regarding end-user devices and back-end infrastructures. Therefore, additional research on the specific impact of the network is necessary in future work, to improve this component and reduce uncertainty when specific information is available regarding the network.

4.3.3. Back-end

Finally, the modeling of back-end infrastructures faces some limitations. The embodied impact of a server is allocated to the maximum number of requests it can handle throughout its lifespan. This hypothesis assumes that the hardware operates at maximal load during its entire lifespan, which is largely an overestimation. Consequently, the result of the back-end layer may be underestimated. If the server only receives half of the maximum requests per second, then the impact factor of each request would be twice as high.

To assess the environmental impact of a given functional unit, we account for both the energy used by the functional unit, and the manufacturing of hardware involved in using the functional unit.

The impacts caused by end-user's devices are estimated from the energy consumption of the functional unit, while network and back-end impacts are imputed from its data usage. As this approach relies on a large number of hypotheses, we rely on fuzzy logic and a systematic quality assessment of our source to provide the uncertainty associated with this impact.

Furthermore, the inventory data is collected according to LCA methodology, with regard to the functional unit. This means that a black-box approach is used for back-ends, assuming a set of requests rather than the technical processes involved in handling such requests. This can lead to significant underestimations, as the infrastructure considered only includes the server, excluding components such as management layers, virtualization, or storage. Finally, third-party services that can be integrated into the software are not accounted for, such as analytics or external content. All requests toward such services are inputted to the back-end of the software under review.

5. Conclusion

This document gathers the methodology used by Greenspector to assess the environmental impact of mobile software. This methodology is composed of three main components:

- a measures framework, automating the measure of data and energy usage on physical devices,
- a grading scale, allowing for assessing the impact of functional unit as a single criterion,
- an impact model, to estimate the impact of the functional unit with regard to the usage and manufacturing of hardware, and accounting for end-user’s devices, network, and back-end infrastructure.

While this methodology is built on scientific and industrial the state-of-the-art, it still has limitations. Understanding such limitations is fundamental to properly interpret results provided by Greenspector. Improving our methodology and addressing such limitations is our priority, and this document will be updated as new improvements are released.