Software Carbon Intensity (SCI) For Web Measuring energy and emissions of web applications

Oliver Winks, Adam Newman, Son Ba Pham
Root & Branch Digital Ltd
oliver@rootandbranch.io, adam@rootandbranch.io, son@rootandbranch.io

August 2025

Abstract

In this paper, we present the **Cardamon Web Model** (CWM), which details how to apply the Software Carbon Intensity (SCI) specification (ISO 21031) to web. This approach addresses some of the limitations we see with single parameter proxies (e.g. cost, data transfer) enabling a more granular measurement of the power consumption and carbon emissions of websites and web applications.

We present the SCI and describe how it relates to each separate segment of web systems; front-end, back-end and networks. We introduce granular bottom-up approaches, presenting front and back end components with their own distinct energy models (e.g. screen, compute, data, idle) and discuss the challenges of expressing networks in the same way.

We then turn theory into practice describing in detail how we take these algebraic formula and implement them physically, introducing the concept of overrideable reference devices. Finally, we reframe the work in terms of user interactions, discussing our design choice to **position user interaction as the fundamental unit block of the CWM** opening new possibilities for web emissions measurement and reduction.

Finally, we frame some of these possibilities as concrete examples describing current and future work implementing and expanding the CWM and how it leads to granular insights and bespoke recommendations for practitioner workflows.

1 Introduction

The environmental impact of digital technologies has become an increasingly urgent concern, particularly as hardware manufacturing and energy demands of digital continue to grow [1] [2] [3]. As essential components of the digital economy, websites and web applications contribute significantly to global energy consumption and carbon emissions [4] [5]. However, accurately quantifying these impacts remains challenging due to the diverse ways they are used and the inherent complexity of web infrastructure.

In this paper, we present the **Cardamon Web Model** (henceforth CWM), named after the Cardamon project [6], an Innovate UK-funded initiative focused on developing tools to assess and reduce software-related emissions. The model is designed to estimate the power consumption and associated carbon emissions of web activity. From measuring single page user interactions, to full user journeys across multiple pages and entire websites.

The CWM builds upon the **Software Carbon Intensity** (SCI) specification [7], taking a bottom-up approach to calculating energy consumption, rather than a top-down methodology like that of other models e.g. **Sustainable Web Design Model** (SWD) [5]. This allows for a more granular analysis of device components that consume energy during web interactions, resulting in greater accuracy and actionable insights.

This paper is organised as follows: Section 2 **Software Carbon Intensity**; introduces the SCI specification. Section 3 **The Cardamon Web Model**; describes the CWM in general terms, outlining its theoretical framework and principles, relating it back to the SCI. Section 4 **Functions, Parameters & Assumptions**; describes the energy consumption functions, constants and assumptions that underpin key calculations. Section 5 **Reference Devices**; discusses reference devices used in the implementation. Section 6 **User Interaction**; introduces user interaction strategies. Finally, Section 7 explores **Future Work** and potential developments to extend the models use cases further expanding its scope and applicability.

2 Software Carbon Intensity

The CWM builds upon the principles of the SCI specification by applying it to the domain of web applications. The SCI is a ISO standard [7] designed to provide a standardised methodology for assessing and comparing the environmental impact of software. Equations (1 - 4) describes the specification in mathematical terms.

$$SCI = (O + M) \quad \text{per } R \tag{1}$$

where: O = operational emissions (i.e. use-phase).
M = embodied emissions (i.e. production and disposal phases).
R = functional unit (e.g. API call, user interaction, etc).

$$O = E \cdot I \tag{2}$$

where: E = energy consumed.
I = region specific carbon intensity.

$$M = TE \cdot TS \cdot RS \tag{3}$$

where: TE = total embodied emissions of the hardware.
TS = share of the total lifespan of the hardware reserved for use by the software.
RS = share of the total available resources of the hardware reserved for use by the software.

Expanding further we have

$$TS = \left(\frac{TiR}{EL}\right), RS = \left(\frac{RR}{ToR}\right) \tag{4}$$

where: TiR = the length of time the hardware is reserved for use by the software.
EL = the expected lifespan of the equipment.
RR = the number of resources reserved for use by the software.
ToR = the total number of resources available.

Throughout this paper we relate the CWM back to this specification by using the same terminology and mathematical notation.

3 The Cardamon Web Model

In this section we explore how the CWM extends the SCI specification, applying it to web applications.

We start by breaking a web application into three distinct sub-systems, modelling each individually to estimate their energy consumption. The sub-systems are as follows:

- **Front-end**; the end-user device.
- **Network**; the routers, switches etc used to connect the end-user device to the various servers used to deliver the website.
- **Back-end**; the computers and associated infrastructure used to host and deliver the content to the end-user device. These are usually computers within a data-centre.

Operational and embodied emissions are calculated for each sub-system during an interactivity test which is conducted on physical reference devices for the front end.
Equation 5 shows how we reformulate the SCI from equation 1 to include these high-level sub-systems.

$$SCI_{web} = \sum_{s \in \mathcal{S}} O_s + M_s \tag{5}$$

where: $\mathcal{S}$ = (f, n, b) = set of system components; front-end (f), network (n) and back-end (b).
O_s = operational emissions of sub-system s.
M_s = embodied emissions of sub-system s.

Let's look at these system components in more detail, starting with the front-end:

3.1 Front-End

To estimate the energy consumption associated with end-user devices, our model accounts for the diversity among end user devices by categorising them into distinct categories such as desktop, mobile, and others. Each category represents a broad classification of devices that share similar characteristics in terms of hardware and usage patterns.

For each category, we define a reference device that serves as a representative model (see section 5). These reference devices are further decomposed into their constituent hardware components, such as the CPU, network adapter, screen, and others. Each hardware component has an associated energy consumption function which takes as input a set of metrics specific to the component (CPU utilisation, data transfer etc).

Metrics for each component are gathered directly from the reference devices during an interactivity test in which a set of actions are performed on the web page. This aligns with the functional unit R in the SCI framework.

Interactivity tests represent a user interaction, a behaviour. They can be confined to a single page or span across multiple pages. This design choice effectively makes an interaction with any portion of a single page the fundamental building block of CWM. These blocks of interaction can be tested independently as single page measurements but also pieced together to model specific user journeys as well as entire websites and web applications by identifying representative pages and interactions that capture the app's overall functionality.

Once metrics have been gathered during the interactivity test we can use them as parameters to our component energy models to estimate the breakdown of system energy. We can then optionally use this information along with user analytics to scale the estimated energy consumption by the number of visits of that type originating from each device category, in each region - building a representative estimation of end user emissions.

Lets begin describing this method mathematically from the SCI specification.

$$SCI_f = O_f + M_f \tag{6}$$

where: O_f = total operational emissions across all end-user devices.
M_f = total embodied emissions across all end-user devices.

To apply the front end portion of the CWM to the SCI specification we must unpack both the operational and embodied emissions terms (O_f and M_f respectively).

3.1.1 Operational Emissions

Expanding the operational emissions term O_f to include each device category and scaling by the number of visits results in:

$$O_f = I_f \sum_{d \in \mathcal{D}} h_d (E_{f,d} + E_{i,d}) \tag{7}$$

where: I_f = average carbon intensity across all end-user devices.
h_d = hits (visits) originating from device category d over reporting period.
$E_{f,d}$ = energy consumption of the d'th category of user-device.
$E_{i,d}$ = idle energy consumption during the interactivity test.
$\mathcal{D}$ = set of user device categories (e.g. mobile, desktop).

Even in this form, the connection to the SCI specification is still very clear. The terms I and E from equation 2 are present but have been expanded in the case of energy consumption to include idle consumption E_i and active consumption E_f.

We can further expand these terms like so:

$$E_i = t \cdot e_i \tag{8}$$

where: t = elapsed time of the interactivity test.
e_i = idle energy consumption of the reference device for the category under consideration.

To expand the active energy consumption we must include the hardware components considered for each device category and the metrics gathered for those components during the interactivity test we can:

$$E_f = \sum_{c \in \mathcal{C}} f_c(\mathcal{M}_c) \tag{9}$$

where: $\mathcal{C}$ = set of hardware components $\{CPU, networkadaptor, ...\}$.
f_c = energy consumption function for component c.
$\mathcal{M}_c$ = metrics associated for component c.

User analytics often contains information about where, geographically, each visit to the web page originated from. This data can be used to calculate the average carbon intensity across all visits. However, in the absence of this user analytics data the global average carbon intensity can be used instead.

$$I_f = \sum_{r \in \mathcal{R}} q_r I_r \tag{10}$$

where: q = proportion of views originating from region r.
I = carbon intensity of region r during the test.
$\mathcal{R}$ = Set of all geographical regions.

3.1.2 Embodied Emissions

$$M_f = t \sum_{d \in \mathcal{D}} h_d m_d \tag{11}$$

where: t = elapsed time of the interactivity test.
h_d = hits (visits) originating from device category d over reporting period.
m_d = embodied emissions rate of end-user device category d.
$\mathcal{D}$ = set of user device categories (e.g. mobile, desktop).

Equation (11) shows how the total embodied emissions of end-user devices is calculated in the CWM. Our formulation differs from that of the SCI specification but is completely compatible with it when we expand the definition of m_d.

Using the terminology defined in equations 3 and 4 and making the assumption that during the interactivity test the device is used exclusively for that task we can expand m_d as follows:

$$m_d = TE_d \left(\frac{TiR_d}{EL_d} \right) RS \tag{12}$$

Under the assumption that the device is used exclusively for the purpose of rendering the web page during the interactivity test equation 12 simplifies to:

$$m_d = TE_d \left(\frac{TiR_d}{EL_d} \right) \tag{13}$$

It is important to note that the CWM in it's current form is likely to overestimate the embodied emissions of the front-end due to the above assumption. Modern devices are likely to be performing many other tasks whilst rendering a web page.

3.2 Network

Network infrastructure presents unique challenges. With front-end and back-end subsystems we have access to usage metrics for the constituent hardware components so they can be modelled bottom-up. With networks the route that data packets take is usually non-deterministic and we very rarely have access to the hardware data because most of the infrastructure is not accessible. For instance with undersea cables, switches, routers, internet exchange points etc. Consequently, we use a top-down modelling approach to estimate energy consumption of networks.

To model the energy consumption of the network infrastructure, we rely on estimates derived from academic research and established models. One such model is the Sustainable Web Design model [5], which provides an estimate of the energy consumed during data transfer over the global networking infrastructure. This model calculates energy consumption by dividing the total estimated energy consumption of the global networking infrastructure (310 TWh per year [11]) by the total estimated data transfer of the internet (5.29 ZB per year [12]). This results in an average energy consumption of:

$$0.059 = \left(\frac{310 \times 10^9}{5.29 \times 10^{12}} \right) \tag{14}$$

We acknowledge the complexity of network emissions estimations and the limitations of an energy intensity approach i.e. 'emissions per byte'. In the absence of an accepted standard for network emissions we chose to adopt this approach rather than ignore network emissions completely and discouraging data transfer optimisation. We will continue to align our method with the latest research in this space as it evolves.

3.3 Back-End

The back-end subsystem refers to the servers and associated infrastructure responsible for hosting and delivering web content to end-user devices. Estimating the energy consumption and carbon emissions of the back-end can be approached in two ways:

3.3.1 Measuring user interaction tests

The first approach mirrors the methodology used for the front-end. This involves conducting an interactivity test on the front-end while simultaneously recording metrics from the back-end infrastructure. The same energy consumption functions described in Section 3.1 are applied, with the exception of screen energy consumption, as servers typically do not have screens attached. However, this approach requires the application to be tested in a private development or test environment to avoid noise from external requests. For accurate results, the test environment must closely resemble the production environment, which may not always be feasible due to technical constraints. Additionally, this approach requires specialised tools capable of measuring the energy consumption of various back-end processes during the interactivity test. One such tool is Cardamon Core [16].

3.3.2 Measuring overall activity

The second approach, which is described in detail in a separate paper, involves using metrics gathered from the production environment over the period of interest. Most cloud platforms provide metadata about the servers provisioned (e.g. instance type, number of virtual CPUs) and record basic metrics such as CPU utilisation and network traffic. These metrics can be used to estimate energy consumption using various component models. Importantly, this approach is not limited to cloud platforms. As long as metrics are being gathered, the same methodology can also be applied to on-premises data centres couple with a technique to handle the inherent uncertainty with shared cloud resources. Many modern on-premises setups include monitoring tools that track server performance and resource usage, enabling similar estimations of energy consumption and carbon emissions. Unlike the front-end approach, this method relies on historical data spanning the entire reporting period, which is typically several weeks or months, and accounts for all activity on the site during that time.

4 Functions, Parameters & Assumptions

4.1 Functions

In section 3.1.1 we introduced the concept of energy consumption functions for various pieces of hardware. This section details the functions for all hardware components currently considered in the CWM.

4.1.1 CPU

CPU energy consumption is estimated using the Boavizta power consumption profile model [8]. This model estimates power consumption using the following logarithmic function:

$$P(u) = a \cdot \ln(b \cdot (u + c)) + d \tag{15}$$

where: $P(u)$ = Power consumption (W) at utilization u (percentage).
a, b, c, d = Coefficients specific to CPU architecture and TDP.

This power consumption profile captures the non-linear relationship between CPU utilisation and its power consumption. The coefficients a, b, c and d define the shape of the power curve.

We obtain these coefficents by using non-linear regression against example solutions provided by the TEADS Thermal Design Power (TDP) scaling factor table [8]:

TDP scaling factors				
u	0%	10%	50%	100%
$P(u)$	0.12	0.32	0.75	1.02

4.1.2 Network Adapter

To calculate network adapter energy we take active and idle draw numbers from the reference devices, the transfer rate that the network adapter is capable of sending/receiving and the amount of data sent to a device.

$$P_{nwa} = (N_{active} \cdot (B/N_t)) + (N_{idle} \cdot (U_t - (B/N_t))) \tag{16}$$

where: N_idle = Network adaptor idle energy
N_active = Network adaptor active energy
N_t = Network adaptor transfer rate
U_t = The amount of time of the user interaction
B = megabytes transferred to device

Here we have that B/N_t is the amount of time the network adapter is active, factoring that by the active power draw N_{active} that gives the amount of energy consumed by the adaptor whilst active. Then the inactive time is the user interaction length minus the active time which is factored by the network idle power N_{idle}. These together give us the total energy consumed by nw adapter when idle during the interactivity test. We apply the same approach to both front and back end.

There is an assumption we are making that an actor is responsible for the network adapter being turned on (idle) for the whole period of the interactivity test. This could be questioned when in reality it is likely that nw is being used by other applications during that time. We have assumed this because we lack data. This could be a topic of future work but we also note that our observations are that data adapter energy is neglible when compared with other components in the system.

4.1.3 Screen

Screen energy depends on two factors: The colour profile of the rendered image and the technology used for rendering.

For the purposes of the Cardamon model we consider the two most prominent technologies on the market today: LED and LCD.

Because of the ways these two technologies work LED displays consume different amounts of energy depending on the colours they render. Lighter images consume more energy because the screen must mix red, green and blue channel to create the desired colour. However, LCD displays typically exhibit little difference in energy consumption when rendering different colours.

The Cardamon model uses the colour profile of the webpage, and the probability that it is being rendered on an LED display, to estimate the energy consumption of the screen.

$$P_{scr} = P_{led} + P_{lcd} \tag{17}$$
$$P_{led} = U_{led} \cdot \left(P_{led}^{min} + v_{col} \cdot \left(P_{led}^{max} - P_{led}^{min}\right)\right) \tag{18}$$
$$P_{lcd} = (1 - U_{led}) \cdot P_{lcd}^{pow} \tag{19}$$

where: P_{led} = Power consumed by LED display.

P_{lcd} = Power consumed by LCD display.

P_{led}^{min} = The min power consumption of the LED reference device.

P_{led}^{max} = The max power consumption of the LED reference device.

P_{lcd}^{pow} = The power consumption of the LCD reference device.

U_{led} = The proportion of LED displays for the current device category.

v_{col} = The colour profile score $[0 - 1]$ of the webpage.

To calculate the colour profile score for a webpage we first obtain the average colour by averaging the red, green and blue channels of all pixels. This is then scaled by the power factor for each channel. The pixel power of LED technologies is dependent on the specific manufactured screen but the research into different devices is consistent [17] [18] [19] that red light is comparable but less power than green which in turn is less than blue. We use blended weights from three different physical screens to obtain the power factors detailed below.

Note, depending on the implementation of any tooling, we may need to account for the alpha value for opacity also. If calculating directly from the rendered browser DOM then background pixels would also need to be considered. A simple solution to this is by calculating the RGB from a screenshot where the opacity is baked into the pixel RGB values.

$$v_{avg} = (k_r \cdot C_r) + (k_g \cdot C_g) + (k_b \cdot C_b) \quad (20)$$

$$v_{max} = (k_r + k_g + k_b) \cdot 255 \quad (21)$$

$$v = \frac{v_{avg}}{v_{max}} \quad (22)$$

where: v_{avg} = Score of average pixel

v_{max} = Maximum score possible (score for white pixel)

k_r = Power factor of red LEDs = 0.897

k_g = Power factor of green LEDs = 1.0

k_b = Power factor of blue LEDs = 1.757

C_r = Value for red channel of average pixel

C_g = Value for green channel of average pixel

C_b = Value for blue channel of average pixel

5 Reference Devices

Detailed information about the devices employed in the use of a website are either difficult or impossible to know. For instance, which end user devices are used to access a website or application? They are numerous and the device landscape is ever changing. In the case of servers running within a data-centre the information is possible for the vendor to get but most often not made readily available.

That said, we can always do the best job possible with the data available and for the data and physical devices we don't have access to we can use sensible reference devices. In both the end user and server (data centre) cases, CWM uses reference devices to provide data for device hardware where it is not available.

These reference devices are crucial to progress. The industry needs to support behavioural change and place relevant energy and carbon data into practitioner workflows, to influence more sustainable decision making. Without reference devices this is simply not possible, perfect data is not available though we expect this will improve (e.g. through consumer pressure and regulation). Reference devices unlock immediate opportunity to make progress now and begin to position information with those making decisions (creative, strategic and technical). They will always be required to some extent (e.g. in the case of end user device proliferation).

In cases where the hardware is known, reference devices are not required and data relating to the known hardware can be substituted in the CWM.

This section describes how these reference devices are chosen and how resources are mapped to the reference devices.

For embodied data we use specific data sheets where available or the reference device as a proxy where not. For operational data we take the utilisation metrics (as described in the previous section), which can be run on any device, mapping them as inputs to the relevant operational energy component model, as described in previous sections.

Note, the reference devices detailed below are examples and non-exhaustive, this repository is completely extendable. Devices and accompanying data sheets can be added over time and blended together to be more representative of real world conditions. For this paper, we present just one example for each but in implementation this is flexible.

5.1 End-User Desktop

Some modern desktops have the screen and desktop built in to the same device. We have chosen to present a more traditional desktop setup of a computer with a monitor (screen) as separate devices. Example reference device(s)

Model	Description	Min (W)	Max (W)
Philips, 272B7QPJEB (LCD) [9]	27" 2560 x 1440	0.30	19.68
Alienware, AW2725D (OLED) [10]	27" 2560 x 1440	0.30	25.30

5.2 End-User Mobile

There are billions of mobile phones worldwide increasing each year [13] with many different model types. Below is an example reference device for a mobile phone that can be easily augmented with others or replaced.

Model	Description	Idle (W)
TBD	Mobile Phone	TBD

5.3 Servers

There are less servers worldwide than end user devices [citation?] but still many different models. The exact same approach applies to servers.

Model	Description	Idle (W)
PowerEdge R660xs [14] [15]	Low Performance	127.1
PowerEdge R660xs [14] [15]	Typical Performance	138.7
PowerEdge R660xs [14] [15]	High Performance	201.8

We note this represents power from everything inside the rack server (including compute, storage, memory). The data centre's Power Usage Effectiveness (PUE) can be used to factor the infrastructural overhead on top of the IT hardware.

6 User Interaction

As described earlier, user interactions are the fundamental building blocks of the CWM. When a human (or bot) visits any web page or uses any application they are interacting with it. These user interactions can be anything from a simple page load, to adding items to a basket, processing a purchase, reading an article, watching a video, playing a game etc. We can think of the whole internet as a series of user interactions. Given that a user interaction is the fundamental unit of CWM, being able to run an interactivity test, we can do all sorts of interesting things. We will highlight two main use cases.

6.1 Estimating Total Website Emissions

Given representative user analytics data we can estimate total emissions of end user activity across a site. To do this we identify representative user journeys and implement them as scriptable tests, running them across a representative sample of pages from a website (or sub section of a site). This is useful for the purpose of reporting end user emissions but the total numbers obtained are not optimisable because the numbers are dependent on user activity which is not directly in our control. However, we can prioritise the user journeys (based on overall emissions) and use the approach in 6.1 to optimise as per SCI. By monotiring system metrics during an interaction we build on the ideas of the Sustainable Web Design Model, which rates pages using data based on page loads (only), by unlocking insights for more involved user interactions. Out of the box our own implementation of the CWM runs a simpe scroll test mimicing a basic user interation on a single page for the purpose of generic footprinting but the full power of the CWM is realised when bespoke user interactions for a given product are modelled to optimise specific user journeys.

6.2 Estimating User Journeys

We can script (or record in tools like Chrome developer tools) re-playable user interactions, we call this an interactivity test. We can replay these interactions and use CWM to estimate energy and emissions of specific user journeys. Thus, giving us the ability to consistently replay scenarios and optimise user journeys. Recognising key user journies within sites and applications, then optimising them gives product development teams a great opportunity to reduce emissions. This aligns with SCI 'per R' functional unit.

7 Future Work

This paper lays the foundation for an approach to applying the SCI to web applications. There is much more work to be done in this area and we are continuing our research. Here are some of the areas we are focusing to build on this initial work.

- Ratings A power based rating system for all web activity that aligns with current industry rating systems [In progress]
- Reference devices Extending device distributions. The CWM can easily support multiple reference devices in each category and even more categories (e.g. tablet, TV) [Planned]
- CDNs Many sites deploy static assets into Content Delivery Networks so that users can access data closer to their location. These are network deployed storage optimised servers that we model within CWM back end model and could improve with more granular data on CDNs.
- Network We plan to conduct research into the power draw of different network technologies (top down) and map device types to weighted distributions to obtain a more nuanced number for network [Planned]
- GPU We plan to bring GPU into this model allowing us to better handle GPU intensive applications. [Planned]
- RAM Initial research suggest that RAM energy is much less variable than CPU with utilisation level, so in the current model RAM is "covered" into idle energy. There is a small improvement to verify our assumption or to model RAM energy specifically. Regardless of which avenue is taken for energy we will improve on the embodied emissions modelling by including RAM in the component modelling for embodied resource share [In progress]

- Component models The operational energy component models could also be improved by extending research into the individual component models. [Unplanned, academic collaboration called for]

- Page sampling Work has already begun on demonstrating how to take this approach, coupled with statistical methods, and extrapolate measurement samples to more accurately estimate entire large scale websites. [In progress]

References

[1] https://www.iea.org/energy-system/buildings/data-centres-and-data-transmission-networks/

[2] https://www.iea.org/news/ai-is-set-to-drive-surging-electricity-demand-from-data-centres-while-offering-the-potential-to-transform-how-the-energy-sector-works

[3] https://www.statista.com/outlook/cmo/consumer-electronics/worldwide

[4] https://www.ericsson.com/en/reports-and-papers/white-papers/ict-energy-evolution-telecom-data-centers-and-ai

[5] https://sustainablewebdesign.org/estimating-digital-emissions/

[6] https://cardamon.io

[7] https://www.iso.org/standard/86612.html

[8] https://doc.api.boavizta.org/Explanations/components/cpu/

[9] https://www.documents.philips.com/assets/20240209/9d500fda9c9640ca8535b11100e732bb.pdf

[10] https://gzhls.at/blob/ldb/1/5/f/9/bfacc16e0d69f5f3bea181ec3faa41ef2a88.pdf

[11] https://www.iea.org/energy-system/buildings/data-centres-and-data-transmission-networks

[12] https://www.itu.int/itu-d/reports/statistics/2023/10/10/ff23-internet-traffic/

[13] https://www.statista.com/statistics/245501/multiple-mobile-device-ownership-worldwide/

[14] https://www.delltechnologies.com/asset/da-dk/products/servers/technical-support/poweredge-r660xs-spec-sheet.pdf

[15] https://data.energystar.gov/Active-Specifications/ENERGY-STAR-Certified-Enterprise-Servers/qifb-fcj2/data_preview

[16] https://github.com/Root-Branch/cardamon-core

[17] https://ieeexplore.ieee.org/document/7273525

[18] https://www.usenix.org/conference/hotpower14/workshop-program/presentation/chen

[19] https://www.yecl.org/publications/dong09dac.pdf