ecoCompute

November 13, 2025

Methods to determine the environmental footprint of software – on the way to standardization

Part I: Holger Smolinski – Bosch

Part II: Jens Gröger – Öko-Institut

BOSCH

Methods to determine the environmental footprint of software – on the way to standardization

Part 1: Bosch SW Eco Footprint model

BOSCH

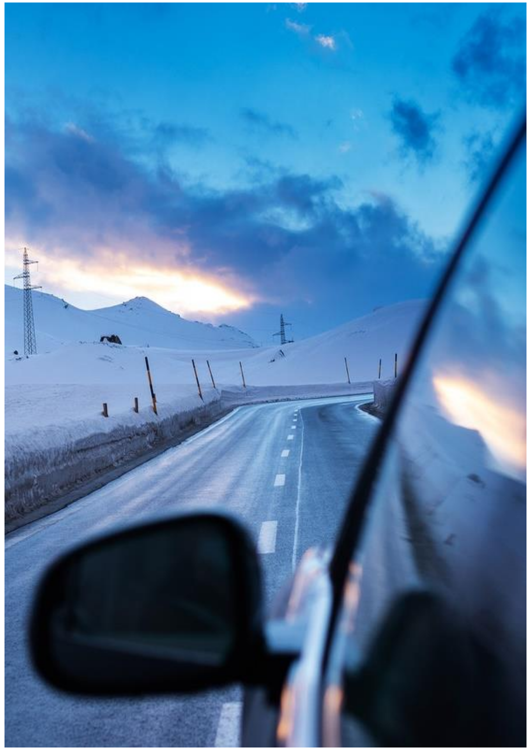



Invented for life

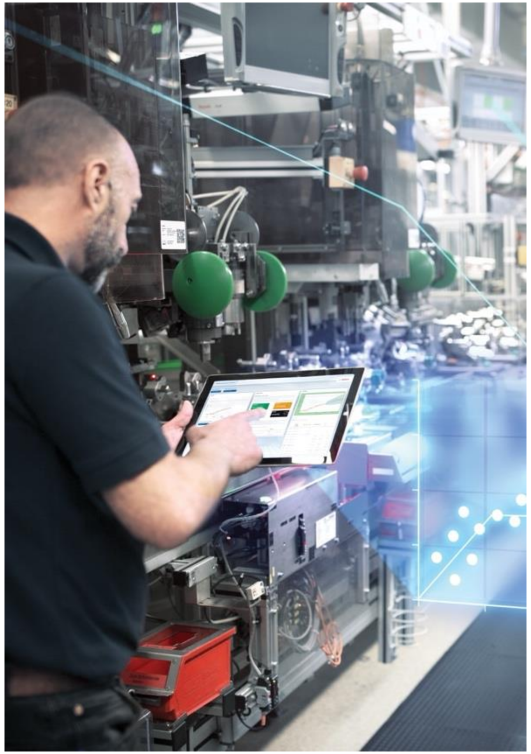



BOSCH

Who we are
Our business sectors



Mobility



Industrial Technology

Consumer Goods

Energy and Building Technology

BOSCH

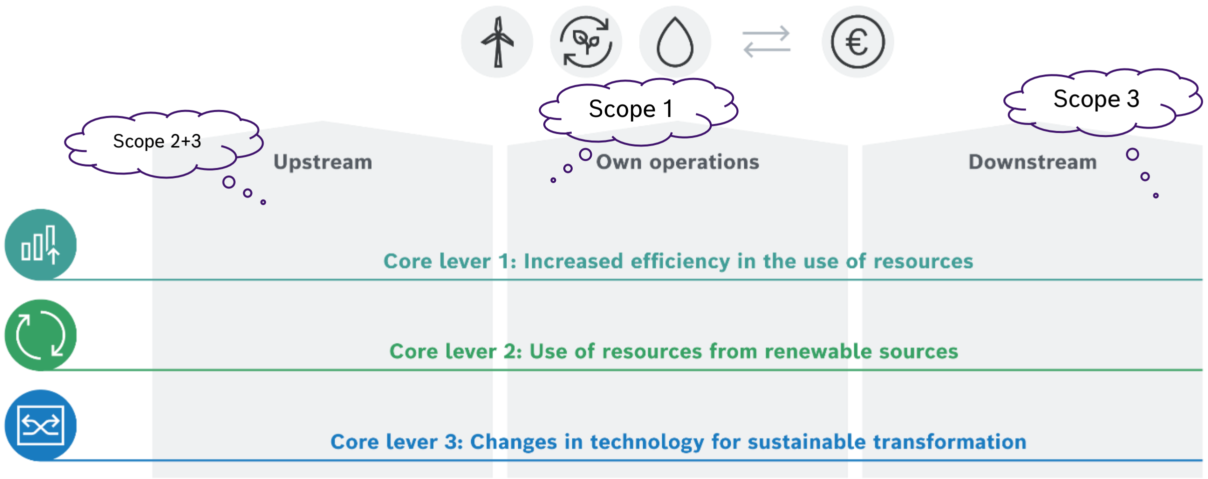
Source: Bosch Sustainability Report 2024

Sustainability at Bosch
Uniform sustainability methodology enables systematic control

The environmental dimensions of our vision for sustainability are underpinned by the same logic. This is characterized by **three core levers**, that can be applied along the value chain.

Sustainability methodology

Core levers for improving sustainability performance along the value chain



Source: Bosch Sustainability Report 2024

Climate action

Carbon neutral in scope 1 & 2 since 2020

3.3 million tons CO_2 in 2018

at our over 400 locations worldwide (scope 1 & 2)

Energy Efficiency → **New Clean Power** → **Green Electricity** → **Carbon Offsets** → **0 Carbon neutral** in scope 1 & 2 since 2020

	Energy Efficiency	New Clean Power	Green Electricity	Carbon Offsets
Goal 2030	Savings: **1.7 TWh**	In-house generation: **400 GWh**	Green Electricity: **100%**	Carbon Offsetting: **max. 0.5m t CO_2**
Achievement 2024	67%	49%	99.5%	531,300 t CO_2

-84% of CO_2 emissions since 2018

Where we want to go
Our sustainability targets

CO_2

-30%

absolute reduction in
scope 3 carbon emissions
by 2030*

*compared with 2018 baseline year

Where we want to go
Our sustainability targets

-25%

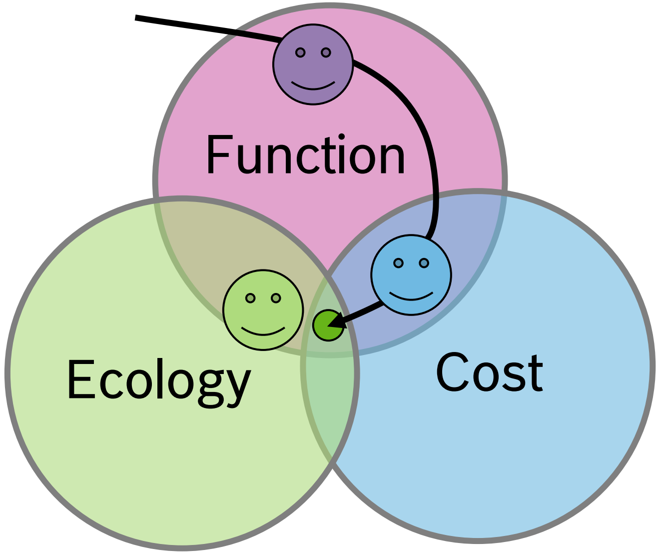
absolute reduction in water withdrawal in regions with water scarcity by 2025*

*compared with 2017 baseline year

BOSCH

Sustainability strategy of Bosch
Design for Sustainability (Eco Design)

Reduce Ecological Footprint, especially Green House Gas emissions,
during the **entire** product lifecycle
by a considerate **product creation process**
with a **Product Lifecycle** view.



Lifecycle Analysis of Software Eco Footprint

SW has a different lifecycle profile than a typical HW

Example:
Carbon Footprint

Software PLC Influencing factors	SW Architecture	Build Pipelines Operational Embodied	SW Deployment & Embodied Emissions	SW Operations	Negligible excpt. Replacement Re-Use and is a topic
PLC Phases*	Concept	Development	Production & Delivery	Product Operations	Phase Out
Hardware PLC Influencing factors	Design decisions Material decisions	Equipment Materials	Equipment Materials	Depends on physical function	Recycling / Second Life / Re-Use / Disposal

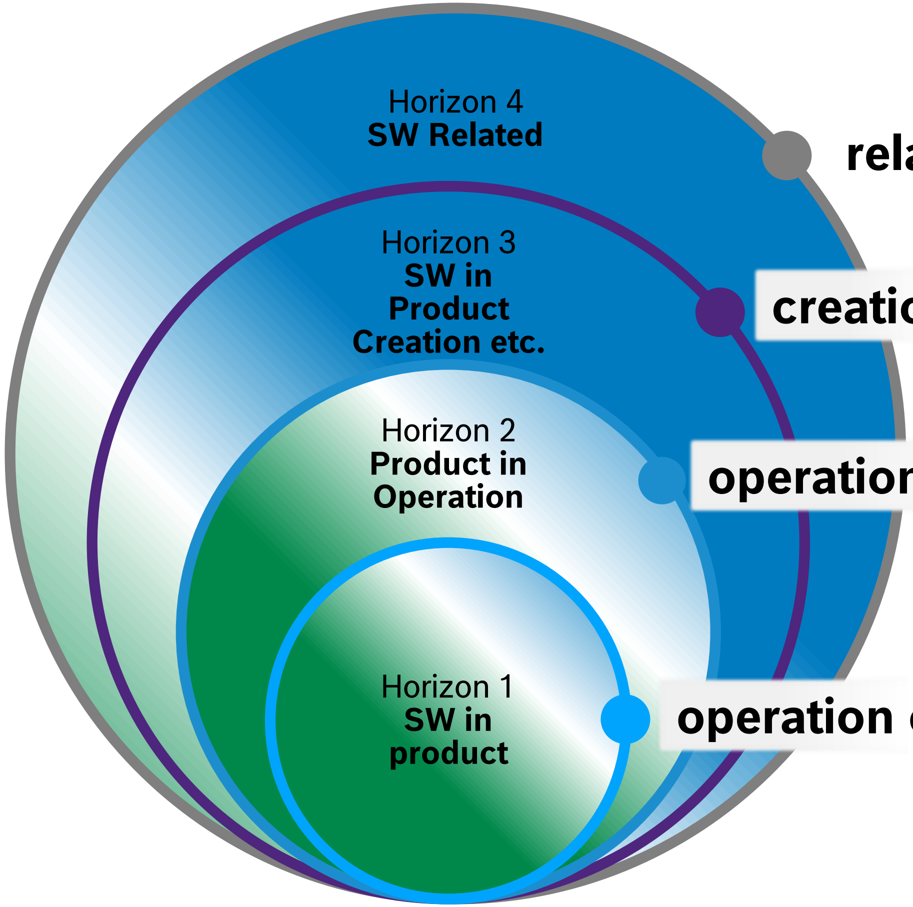
* Product Lifecycle Phases (BES)

PLC = Product Lifecycle, BES = Bosch Engineering System

BOSCH

The Bosch SW Eco Footprint model

Four Horizons and two Perspectives of the eco Footprint



related to the SW (catch all)

creation, maintenance, etc. of SW

operation of supporting SW and IT

operation of SW in/as the product

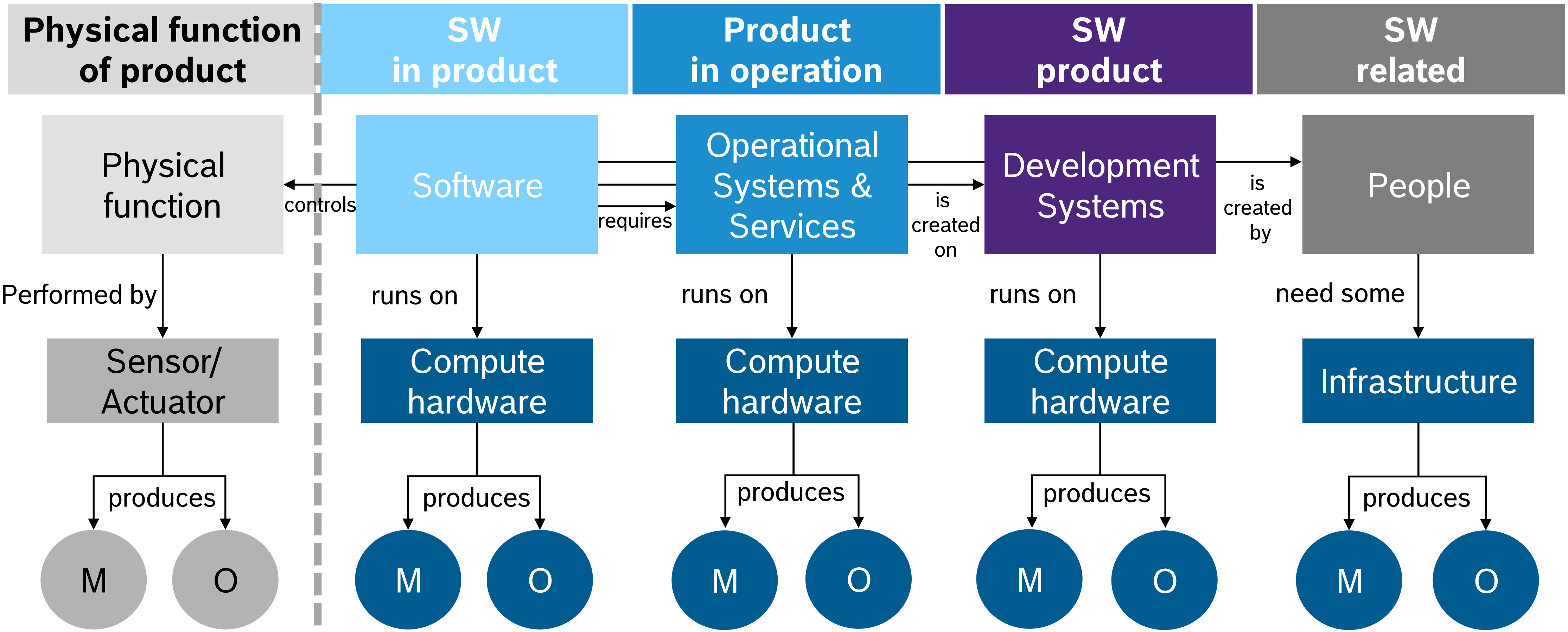
Perspective

Producer

Operator

BOSCH

The Bosch Eco Footprint model
Big picture: The Software Boundary and Four Horizons of SW



M = embodied emissions, O = operational emissions, Derived from SCIS: Software Carbon Intensity Specification

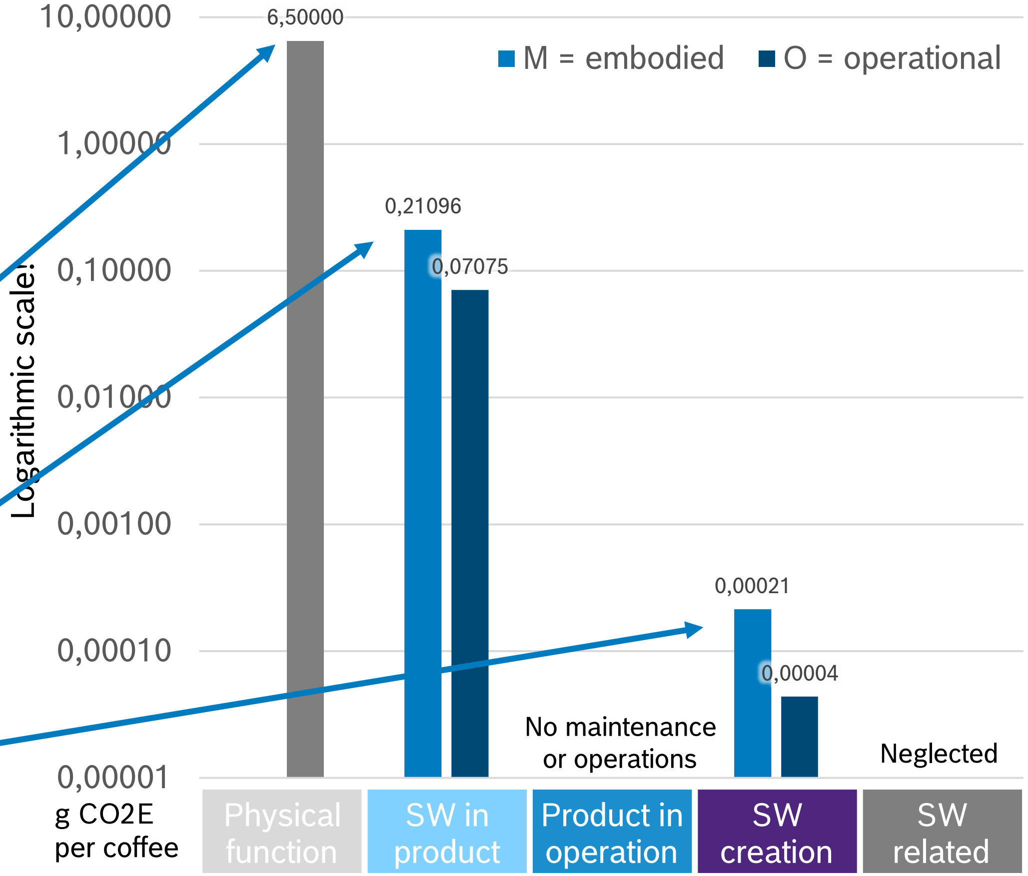
BOSCH

Software PCF (SCF)

Deep dive 1: Coffee maker (offline)

- Product function:
  Produce a cup of coffee
- Function of SW:
  Get user input, control coffee making process, display data.
- Functional unit for calculation:
  One cup of coffee
- Usage Profile:
  5 cups of coffee per day, 10 years usage

- Top levers of SCF:
  - Physical function. No real surprise.
  - $M_{\text{SW in product}} + O_{\text{SW in product}}$ less than 5%, ...
  - ...but similar order of magnitude
  - $M_{\text{SW in product}} > O_{\text{SW in product}}$
  - SW creation negligible



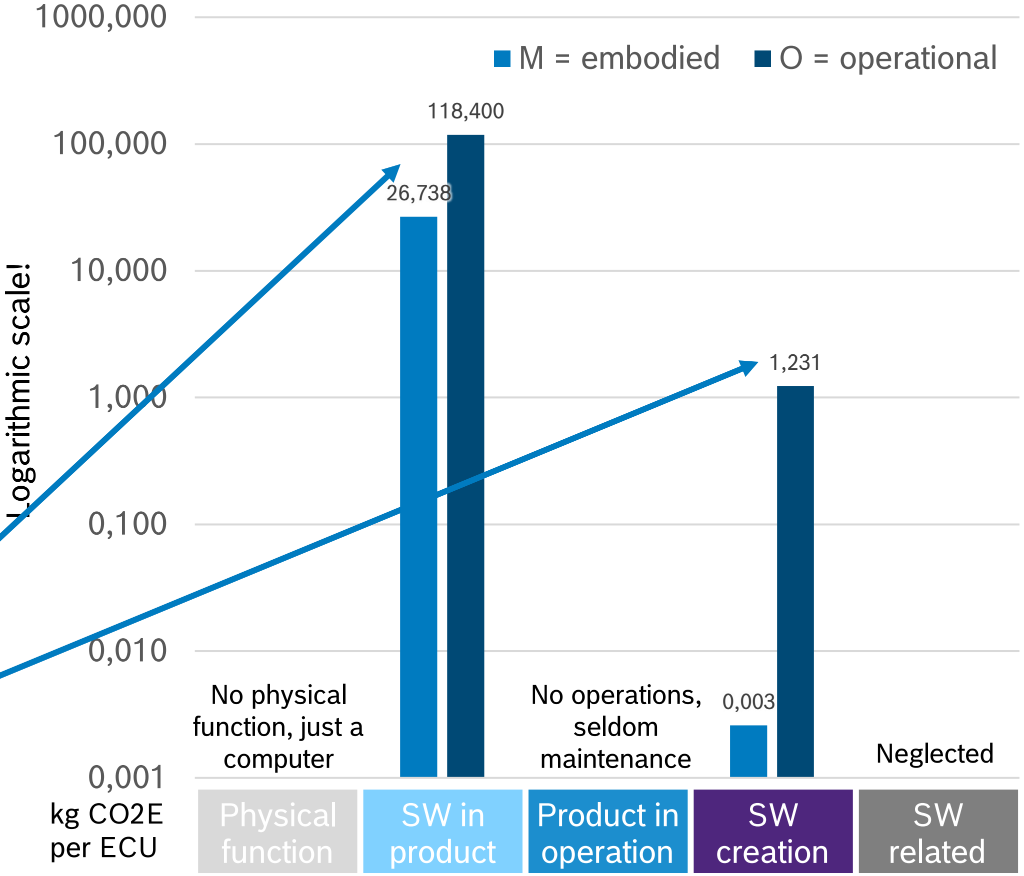
Software PCF (SCF)

Deep dive 2: Vehicle compute unit (VCU)

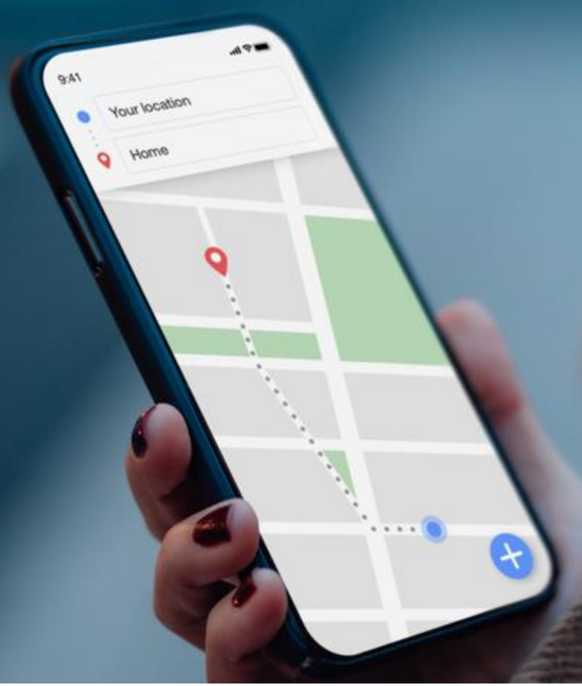
- Product function:
  Compute data, provide safety and comfort to driver.
- Function of SW:
  Collect sensor data, calculate driving trajectory based on environmental model.
- Functional unit for calc.:
  Lifetime of one ECU
- Assumptions, constraints (excerpt):
  1 million units, 100PB training data

Symbolic picture

- Top levers of SCF:
  - $M_{\text{SW in product}} < O_{\text{SW in product}}$, like any computer
  - SW creation negligible due to #units

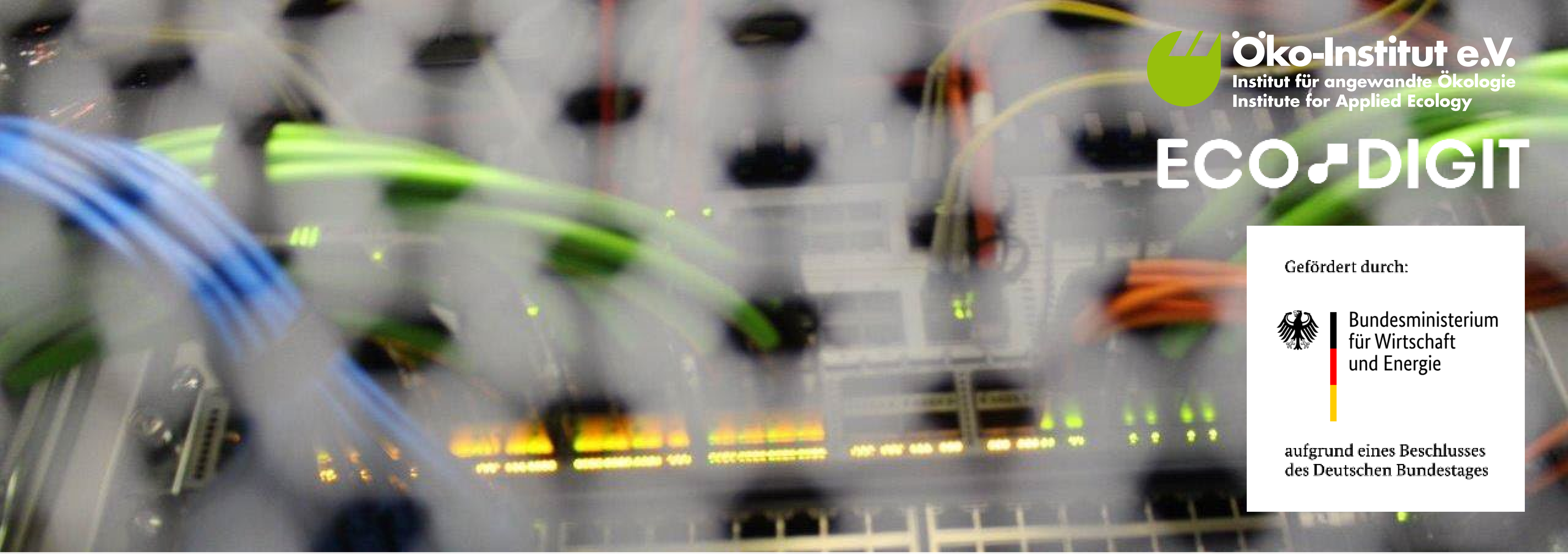


Thanks for your interest in the Eco Footprint Lifecycle Analysis of Software



Contact me:

Holger.Smolinski@de.bosch.com



Öko-Institut e.V.
Institut für angewandte Ökologie
Institute for Applied Ecology

ECO DIGIT

Gefördert durch:
Bundesministerium für Wirtschaft und Energie
aufgrund eines Beschlusses des Deutschen Bundestages

Methods to determine the environmental footprint of software – on the way to standardization – Part 2: Allocation

Jens Gröger, Öko-Institut e.V., ecoCompute, Berlin, 13 November 2025

Challenges in software life cycle assessment

- Software applications usually run on **shared IT infrastructures** (e.g. computers, servers, cloud infrastructure) that are also used by other applications.
- The **manufacturing effort** and the **environmental impacts during the use phase** must be appropriately distributed (allocated) among the various software applications.
- Standardised **allocation rules** must therefore be established.

Software	GWP
App 1	??
App 2	??
App 3	??
Sum	**e.g. 100 kg CO_2e**

Method for allocating environmental impacts

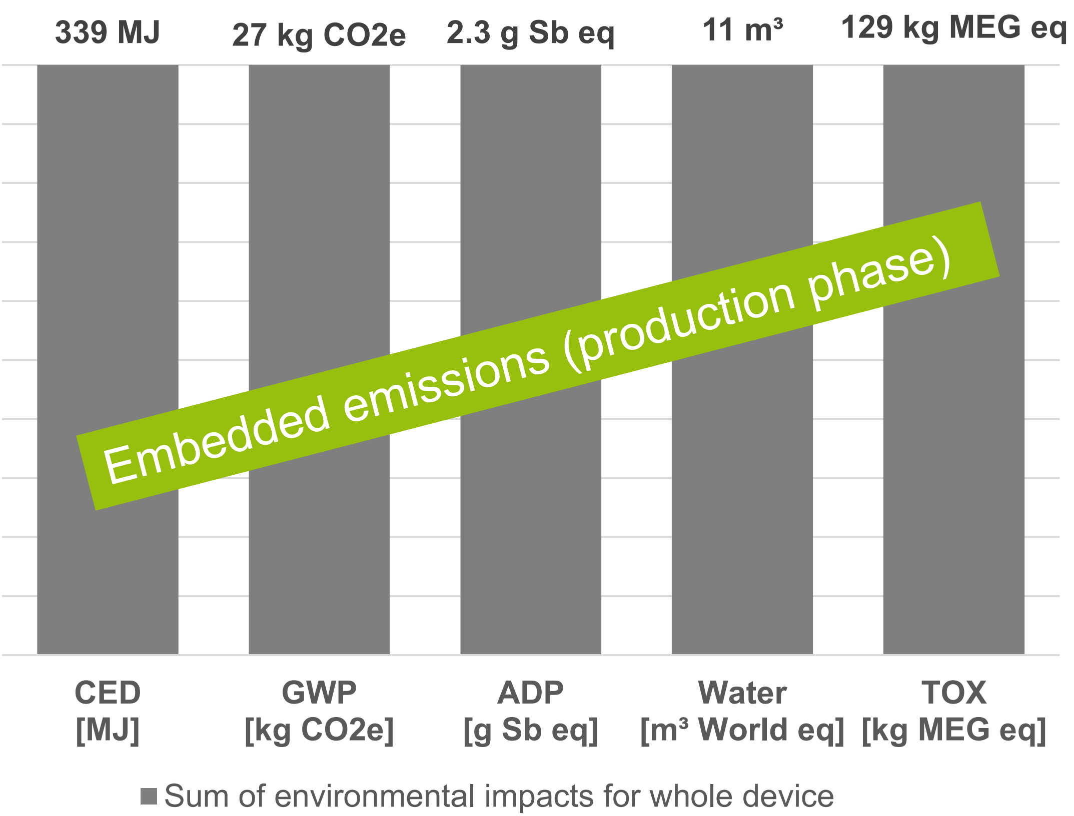
- The method we developed in the ECO DIGIT project allows the accounting of software that runs on shared IT infrastructures (e.g. cloud applications, containers, virtual machines)
- Environmental impacts are allocated using ‘Basic Digital Resources’:
  - Computing power (compute) (e.g. server, user device)
  - Memory performance (memorize) (e.g. RAM)
  - Storage capacity (store) (e.g. storage system,HDD, SSD)
  - Data transfer (transfer) (e.g. switch, router, mobile communications)
- Every **computer provides** a certain amount of basic digital resources over its lifetime
- Every piece of **software uses** a certain amount of basic digital resources to be executed

→ The environmental impact of manufacturing and using the hardware can be allocated to different applications by using the ‘Basic Digital Resources’ as a distribution key

Example: Running 3 Software Apps on a Raspberry Pi 4 Model B

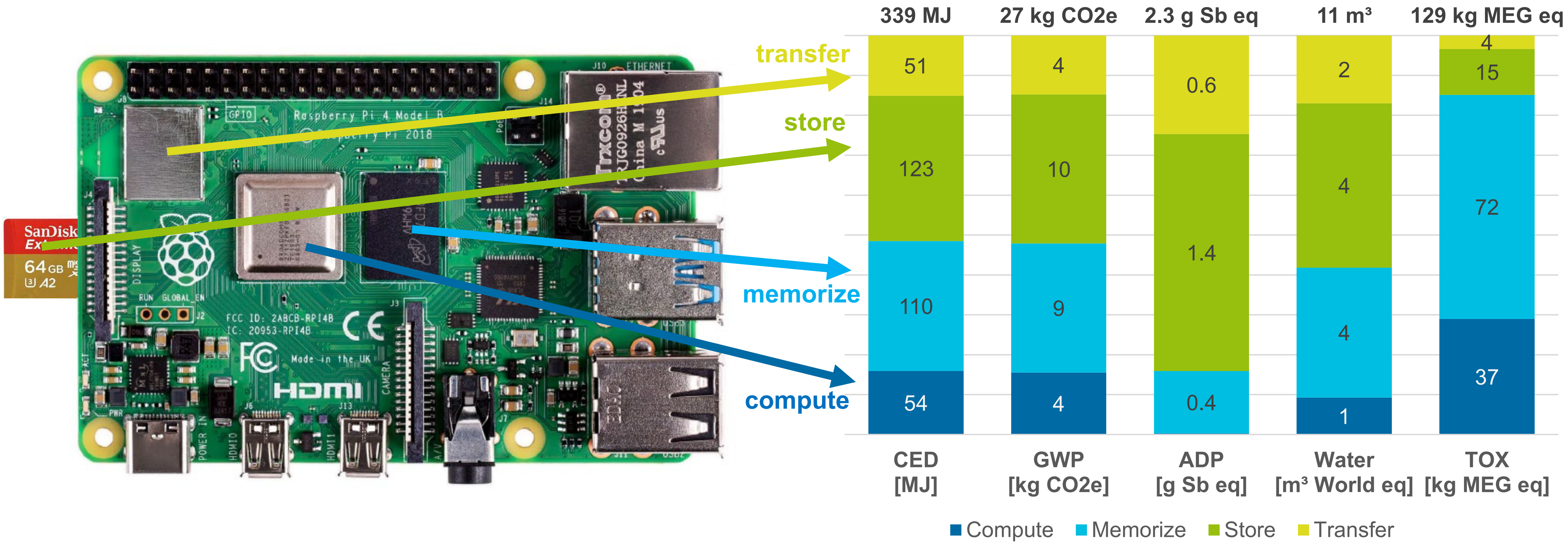
- Generation of a continuous video stream using 3 different software applications:
  - App 1: ecopower.py
  - App 2: ffmpeg
  - App 3: motion
- Delivery of the video stream via the local network
- Receiving the video stream on a client computer running VLC (not part of the measurement and life cycle assessment)

Environmental impact of the Raspberry Pi 4 Model B *(embedded emissions)*



- *Embedded emissions*: Environmental impacts associated with the manufacture of hardware
- *Use phase emissions*: Environmental impacts arising from the use of hardware, in particular from its electric energy consumption

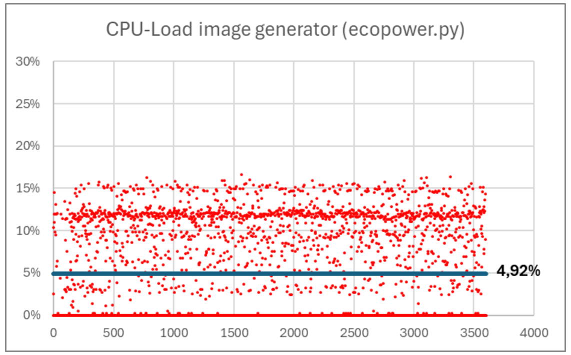
Differentiation of environmental impacts according to Basic Digital Resources

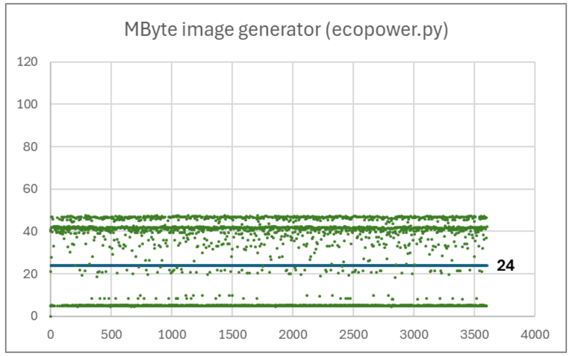


- Breakdown of environmental impact categories into the four „Basic Digital Resources“
- All other components (e.g. circuit board, power supply, heat sink) are distributed proportionally
- This breakdown is also applied to impacts during the use phase (esp. energy consumption).

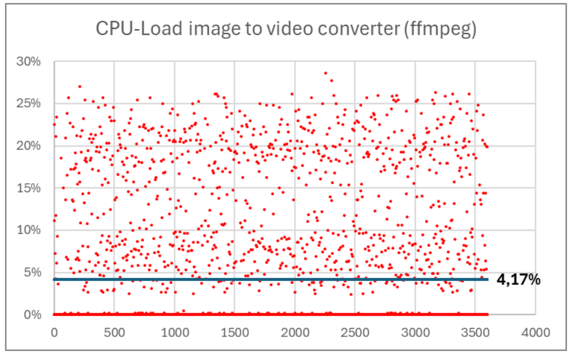
Example: Measuring hardware utilisation with logging tools for 1 hour

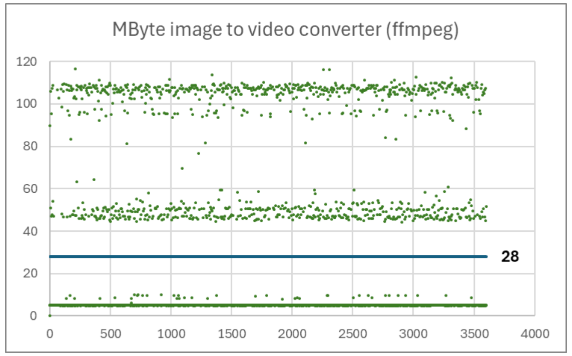
App 1: ecopower.py



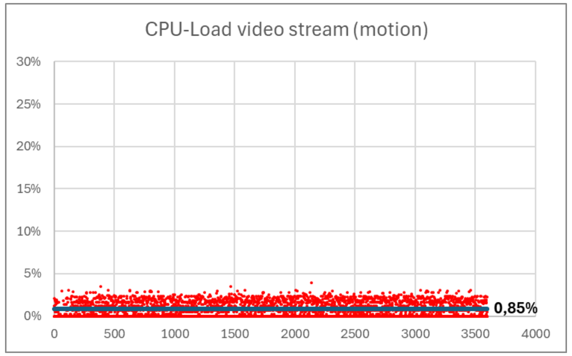


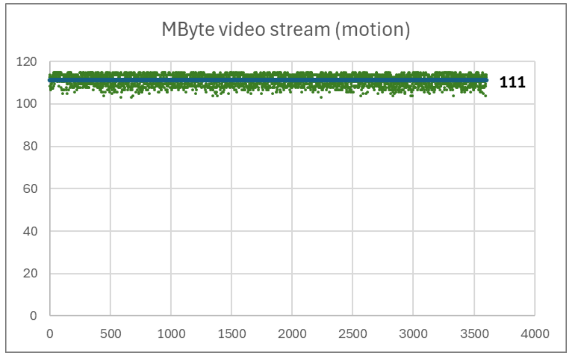
App 2: ffmpeg





App 3: motion



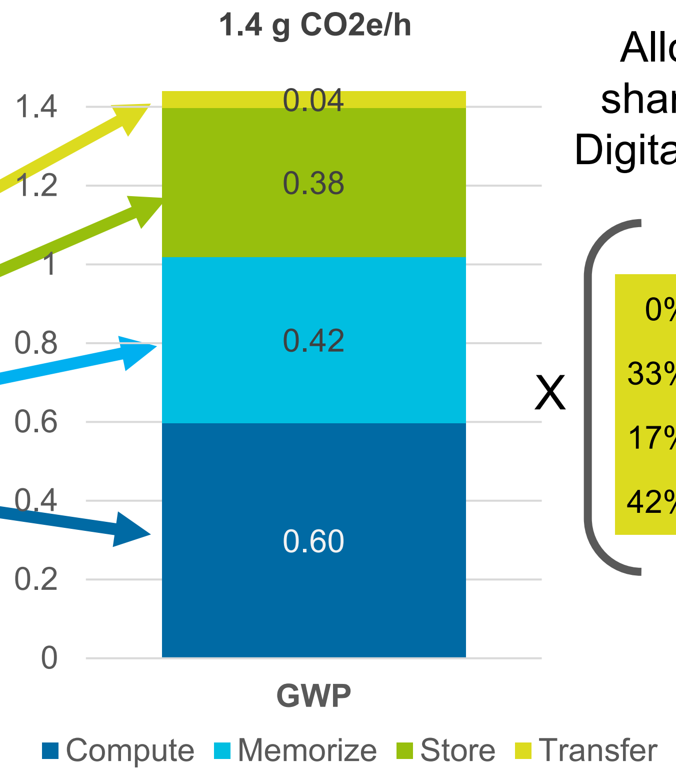


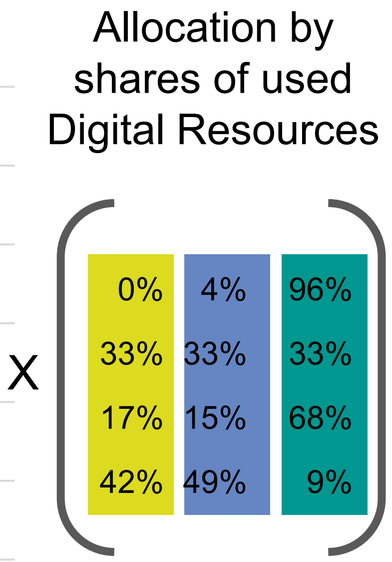
	ecopower.py	**ffmpeg**	**motion**	**Sum**
CPU-Load_average	4,92%	4,17%	0,85%	9,94%
RAM-used_average	24 MByte	28 MByte	111 MByte	163 MByte
SSD-used (assumption: 1/3 of total used)	2,1 GByte	2,1 GByte	2,1 GByte	6,4 GByte
Network-speed_average	2 kByte/s	0 kByte/s	48 kByte/s	50 kByte/s

Environmental impact of energy consumption (*usephase emissions*)

- P_average = 3.42 W
- Software usage for 1 hour

E_tr	=	0,1	Wh
E_st	=	0,9	Wh
E_me	=	1	Wh
E_co	=	1,42	Wh
Summe	=	3,42	Wh

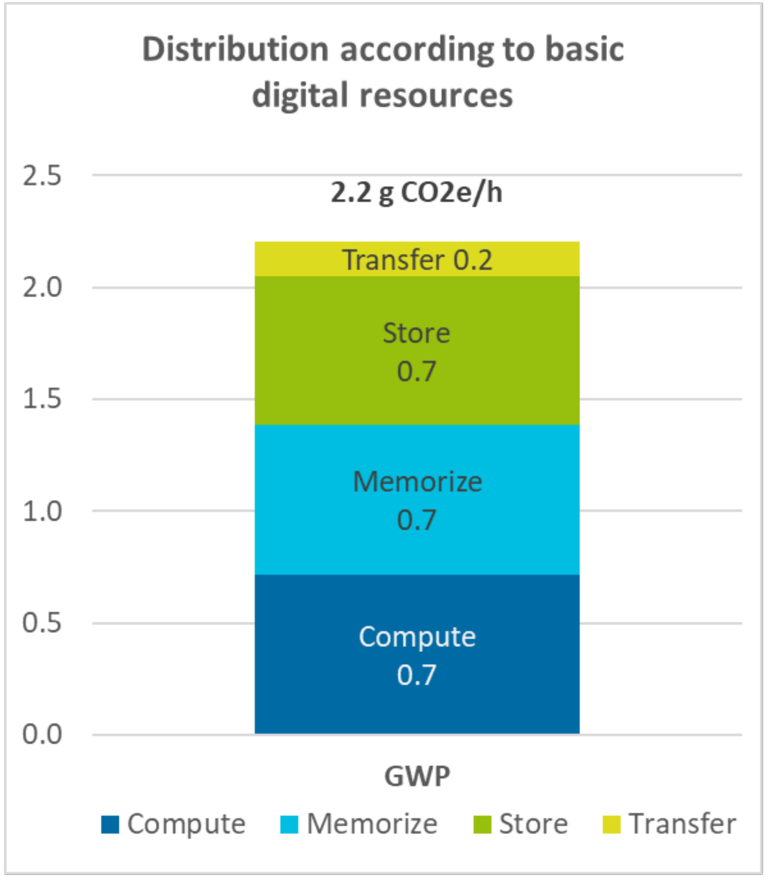


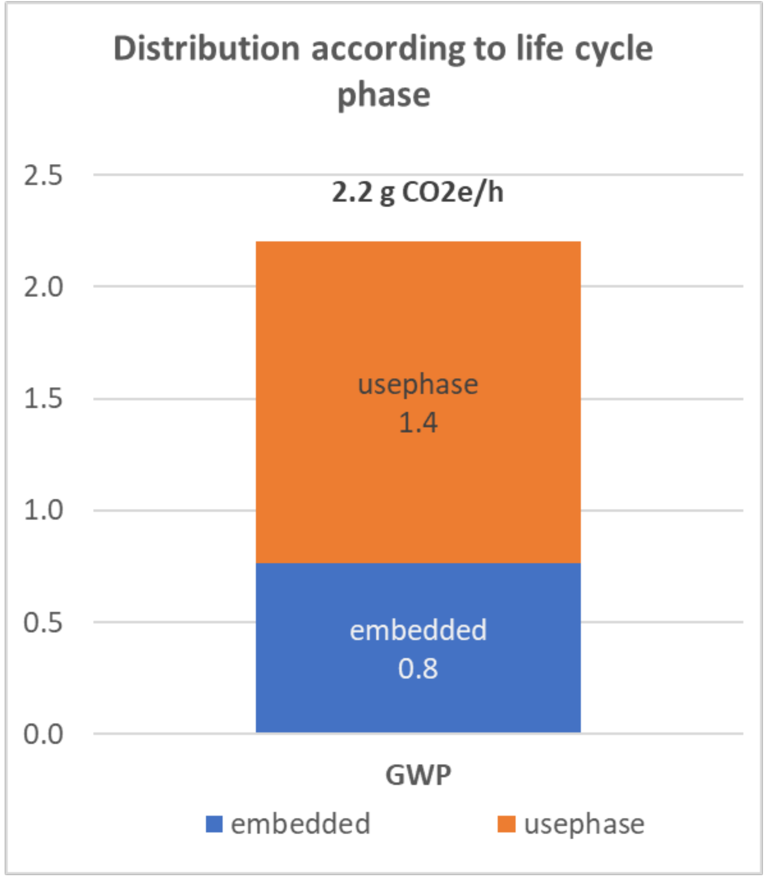


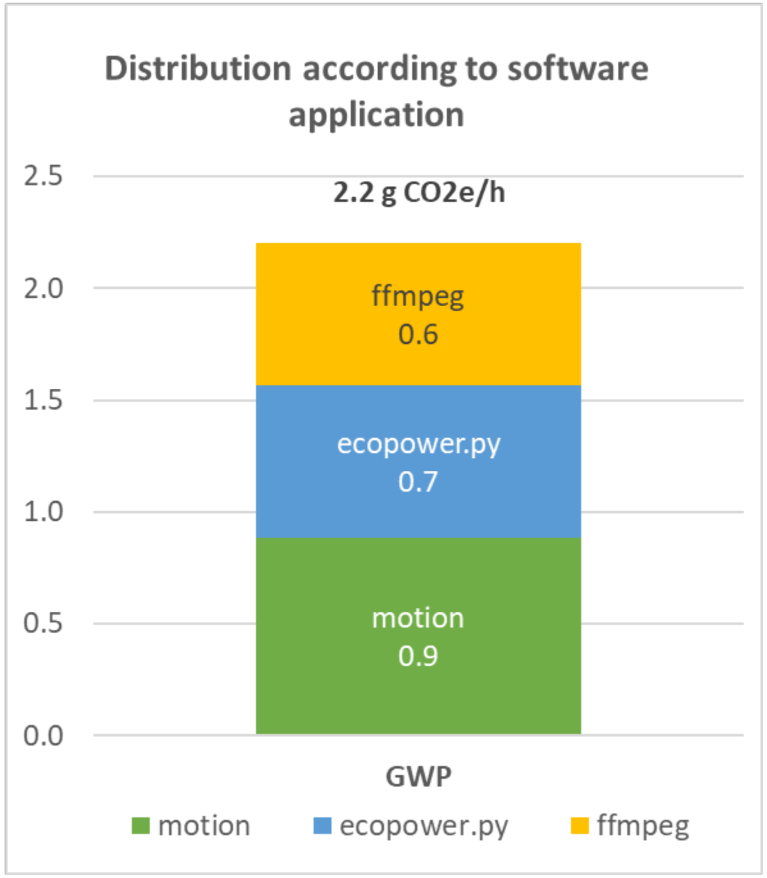
- Allocation shown here using the example of Global Warming Potential (GWP) in the use phase

→ This allocation is applied in the same way to manufacturing effort.

Results: Running 3 Software Apps *(embedded + use phase emissions)*







Conclusions from using the LCA methodology

- The described (eco:digit) methodology can be used to determine the **environmental impact** of software, software bundles, and software running in **shared infrastructures**.
- It works with **any environmental impact category** (e.g. GWP, ADP, Water, WEEE, TOX)
- It helps identify **where** the **major environmental impacts** lie (e.g. manufacturing of hardware, energy demand in use phase, choice of physical components).
- It also highlights the **key issues** to be optimised through green coding (e.g. CPU or RAM usage, data transfer, idle consumption).
- It provides a direct response to any change of code or hardware to **environ. improvements**.
- It includes the **utilisation** and **lifetime of hardware** in the assessment and allows also their optimization.

→ The method should not only be used in a scientific context

→ Standardisation could be used to make it available to a wider target group

On the way to standardization (1): Overview of existing initiatives for determining the environmental impact of software

Metrics initiatives

- https://github.com ...
  - /Breakend/experiment-impact-tracker
  - /cloud-carbon-footprint/cloud-carbon-footprint
  - /fvaleye/tracarbon
  - /mlco2/codecarbon
  - /saintslab/carbontracker
  - /sb-ai-lab/Eco2AI
- https://aws.amazon.com/de/sustainability/tools/aws-customer-carbon-footprint-tool/
- https://www.microsoft.com/en-us/sustainability/emissions-impact-dashboard
- https://cloud.google.com/carbon-footprint

Standardisation initiatives

- ISO/IEC 21031 Information technology - Software Carbon Intensity (SCI) specification
- ISO/IEC CD TS 20125 Information technology - digital services ecodesign - ecopractices for life cycle stages
- ISO/IEC 5055 Information technology - Software measurement - Software quality measurement - Automated source code quality measures
- ISO/IEC 25010 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Product quality model
- ISO/IEC/IEEE 12207 Systems and software engineering - Software life cycle processes
- ISO/IEC CD TR 20226 Information technology - Artificial intelligence - Environmental sustainability aspects of AI systems
- ISO/IEC 5338 Information technology - Artificial intelligence - AI system life cycle processes
- ISO/IEC DIS 12792 Transparency taxonomy of AI systems
- CEN-CENELEC Preliminary Work Item (PWI) Sustainable Artificial Intelligence - Guidelines and metrics for the environmental impact of artificial intelligence systems and services

→ There are a variety of measuring and standardisation initiatives for software

- However, their focus is often limited (e.g. only energy consumption in the use phase) and they cover only a few environmental impacts (e.g. greenhouse gas emissions)

On the way to standardization (2): Necessity for standardisation of a life cycle assessment method for software

- Expectations for a standardised method:*
  - Standards for resource-efficient software may **define testing and measurement methods** to assess various aspects of software and software development processes, including their energy efficiency.
  - Standards can help companies to **clarify and differentiate terms** in the realm of resource-efficient software (e.g. sustainable tech, green coding, green software, green ICT).
  - Certification bodies can validate the adoption of resource-efficient software practices by providing **recognized certifications** that demonstrate compliance with standards.
  - Standards may **prevent greenwashing** and build trust regarding sustainability claims made by companies. This may concern consumers as well as companies in the choice of a product or service providers.
  - **Transparency**: Allowing users and consumers to have a **clear indication** of the energy consumption of the respective software or digital services.

* Source (shortened list): ISO Open Consultation: Resource-efficient software (Oct 24 – Jul 25)

https://www.din.de/de/forschung-und-innovation/themen/klimawandel/ressourceneffiziente-software-im-fokus-der-iso-open-consultation

On the way to standardization (3): Our current standardisation activities

- DIN Deutsches Institut für Normung e. V.
  - Normenausschuss Informationstechnik und Anwendungen (NA 043)
    - Fachbereich Grundnormen der Informationstechnik (NA 043-01 FB)
      - Arbeitsausschuss Ressourceneffiziente Software und Künstliche Intelligenz (NA 043-01-08 AA)
        - **DIN-Arbeitsgruppe: Bilanzierung des Ressourcenbedarfs von Software, KI-Anwendungen und digitalen Dienstleistungen**
        - **(DIN working group: ≈ „Assessing the resource requirements of software, AI applications and digital services”)**
- The working group is preparing the basics for an ISO standard that describes how the environmental impact of software can be determined.
- It is based on existing **life cycle assessment standards** and also specifies
  - how the **scope** of the assessment must be described,
  - which **life cycle phases** must be taken into account,
  - which **allocation rules** can be applied,
  - and how the **quality** of the assessment results can be ensured.

Thank you very much for your attention!

Dipl.-Ing. Jens Gröger
Research Coordinator for Sustainable Digital Infrastructures
Öko-Institut e.V. – Berlin Office
Products and Material Flows Division
Borkumstraße 2, 13189 Berlin
Tel: 030 - 40 50 85 - 378
j.groeger@oeko.de

Methodology description: https://ecodigit.de/en/home/publications

Photo: Google Cloud