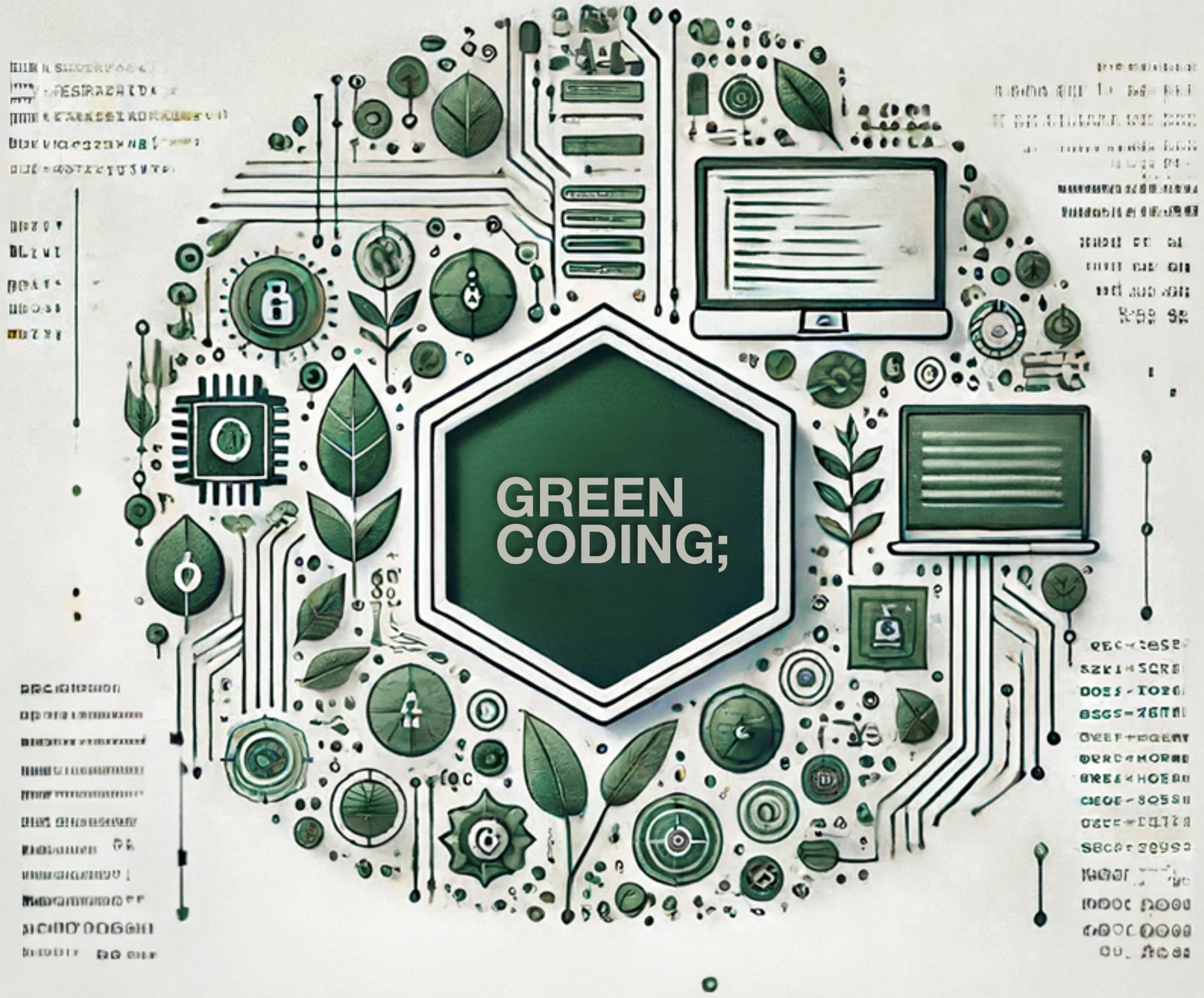


Green Coding Solutions GmbH

Grants and Awards

WSA Winner 2023

Green Screen Coalition

Investitionsbank Berlin

Tackle Software Carbon Emissions

Tools & Management

GREEN CODING;

Who am I

Arne Tarara - Green Coding Solutions GmbH (Germany)

- CEO & Founder - Green Coding Solutions
- We are specialized in the reduction of Software-CO_2-Emissions
- Areas: Consulting & Integration, Research, Measurement Tools & Infrastructure
- We love open source - all our tools are open source and free to use

Lufthansa Industry Solutions

Green Software Foundation

BLUE ANGEL THE GERMAN ECOLABEL

wagtail

Agenda

- **Introduction**:
  - Where exactly does software emit carbon emissions
  - Technology Carbon Standard
- **Mapping emission categories to LCA steps**
- **Open Source Tools Preview**
- **CarbonDB:** Putting everything together
- **Summary**

GREEN CODING;

Technology Carbon Standard - Scott Logic

Upstream Emissions

Category U | 3

Upstream emissions relating to the embodied carbon of hardware, the development of software, and content.

Software

- Off The Shelf including Open-Source Software

Hardware Manufacture, Transport and Installation

- Employee Hardware
- Networking and Infrastructure Hardware
- Servers and Storage Hardware

Content

- Foundation models
- Content and Data

Operational Emissions

Operational emissions are associated to the running of business IT functions and systems. Each operational function may have a hybrid of IT infrastructure across direct and/or indirect emissions.

Direct

Category O | 2

Direct running costs that are attributed to the electricity powering servers, networks and devices.

- Onsite Employee Devices
- Networking and Infrastructure
- Servers and Storage

Category G | 1

- Generators

Indirect

Category C | 3

Indirect running carbon costs that are attributed to external hardware and service solutions.

- Offsite Employee Devices
- Cloud Services
- SaaS
- Managed Services

Downstream Emissions

Category D | 3

Carbon associated with the use of products and services produced by the business.

Customer (B2B, B2C)

- Customer Devices
- Network Data Transfer
- Downstream Infrastructure

https://www.techcarbonstandard.org/

Alright, that's the background

Let's look at how we bring that into something actionable on an organizational level

GSF Maturity Matrix

Assess your companies maturity with Green IT

	ASPIRING	AWARE	ACTING	AWESOME	INSPIRING
commitments	none	carbon neutral	carbon zero with offsets	10% (offset)	1% (offset)
footprint	unknown	know scope 1&2	reducing per unit	reducing absolutely	~zero
metrics	none	report scope 1&2	daily scope 1&2&3	realtime	predicted
carbon ops	none	manual	lightswitch ops	auto-rightsizing	carbon SRE
energy	none	green hosting	dynamic management	demand shaping	24/7 Carbon Free Electricity
devices	none	some targets	10y/90%	10y/100%	rolling repair
utilization	none	some multi-tenant	all multi-tenant	max orchestration	edge integration
products	none	carbon awareness	demand shaping	feature tracking	feature carbon error budgets
training	ad hoc	basic/champions	advanced	you are the trainer	you are the leader

https://greensoftware.foundation/

GSF Maturity Matrix

Assess your companies maturity with Green IT

	ASPIRING	AWARE	ACTING	AWESOME	INSPIRING
commitments	none	carbon neutral	carbon zero with offsets	10% (offset)	1% (offset)
footprint	unknown	know scope 1&2	reducing per unit	reducing absolutely	~zero
metrics	none	report scope 1&2	daily scope 1&2&3	realtime	predicted
carbon ops	none	manual	lightswitch ops	auto-rightsizing	carbon SRE
energy	none	green hosting	dynamic management	demand shaping	24/7 Carbon Free Electricity
devices	none	some targets	10y/90%	10y/100%	rolling repair
utilization	none	some multi-tenant	all multi-tenant	max orchestration	edge integration
products	none	carbon awareness	demand shaping	feature tracking	feature carbon error budgets
training	ad hoc	basic/champions	advanced	you are the trainer	you are the leader

https://greensoftware.foundation/

Challenge

Missing Tools and Missing Interoperability

Challenge: We have many Carbon Measurement Tools, but they are often limited to Bare Metal Servers

Solution: GCS eco system supports all OSs and Virtualized / Non-Virtualized Systems, CI/CD SaaS products, Websites, Developer Machines, AI Models

Challenge: We have not good interoperability between the tools. Often no export functionality

Solution: Standard Export Formats. FOSS Database under your control

Expanding the view even more

Looking at the software "life cycle"

Development	Testing - CI/CD	Load Testing / Optimizations	Monitoring	Reporting / Dashboards
PowerHOG	Eco CI	Green Metrics Tool	Cloud Energy	Carbon DB

Power Hog

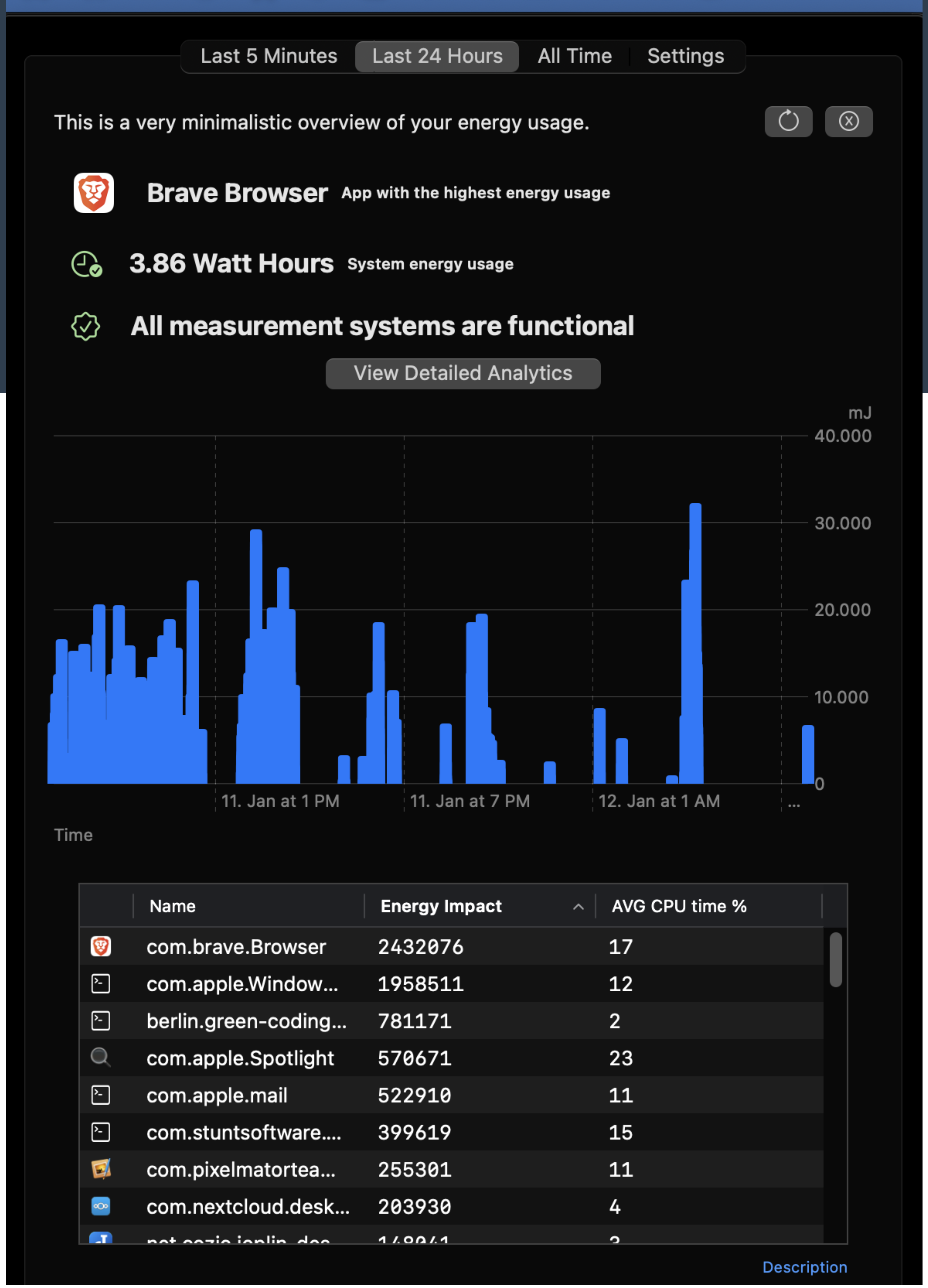
GREEN CODING;

Tech Slide

Power Hog

Development Cost measurement

- Direct measurement of energy on the developer machine
- Granularity per process
- Aggregation per project
- API and drill-down option in dashboard
- Local database for quick analyses



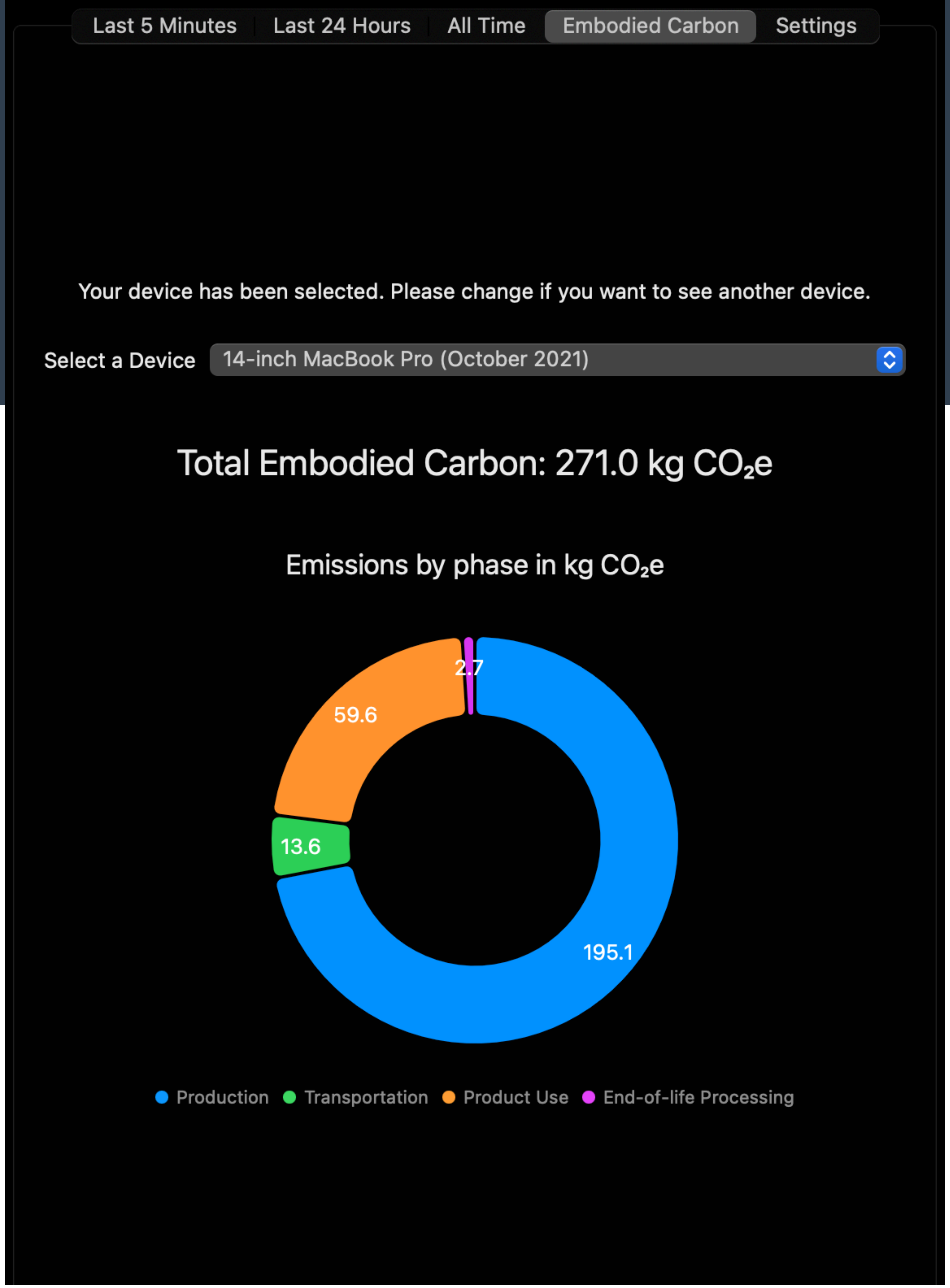
GREEN CODING;

Tech Slide

Power Hog

Embodied Carbon of Devices

- Goal is to find resource consuming software which would trigger a device change
- Keeping the device longer reduces per year embodied carbon
- Aligned with the goals of the Blue Angel for Software :)



GREEN CODING;

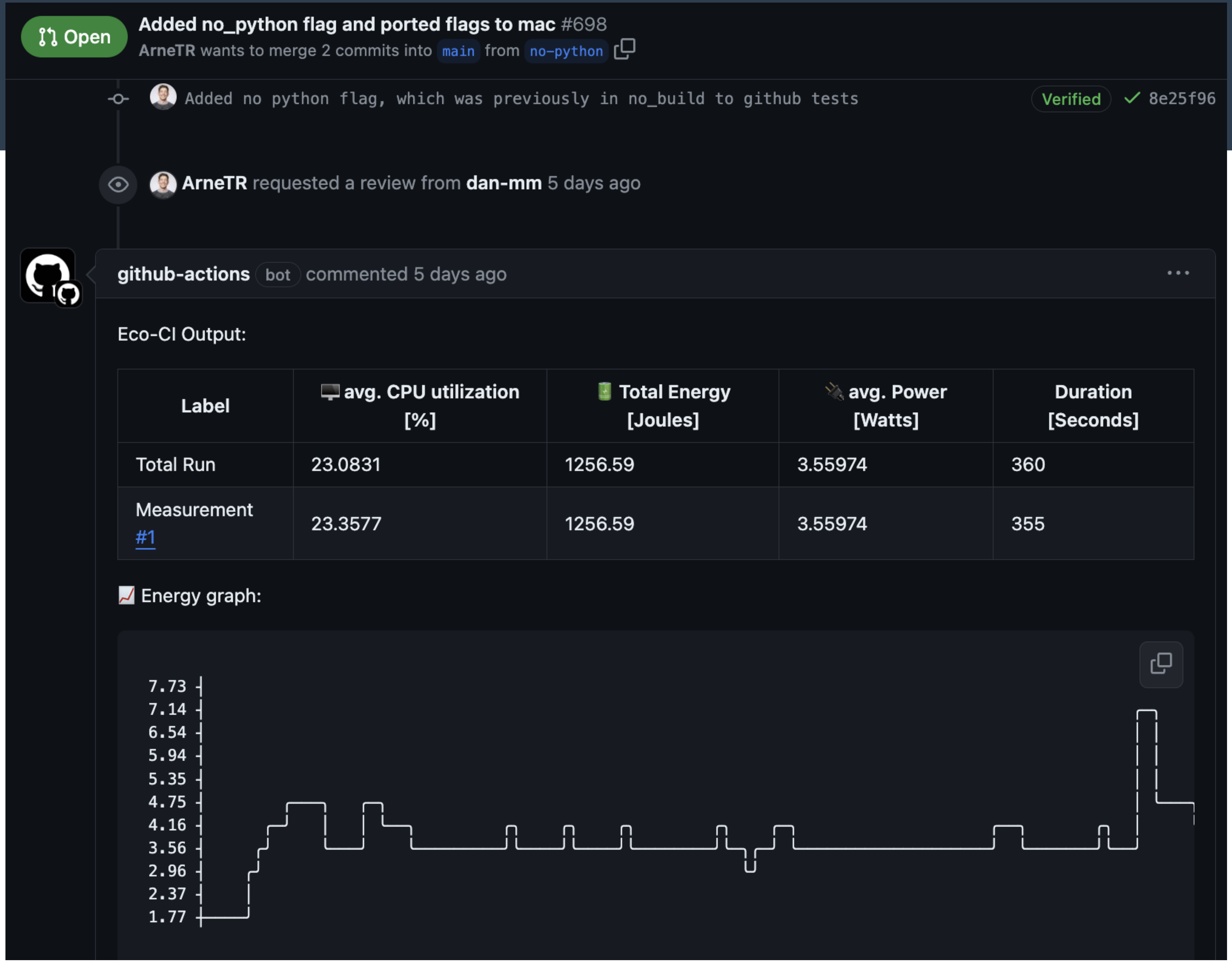
Eco CI

</> GREEN CODING;

Tech Slide

Eco-CI
Estimation in CI / CD Pipelines

- Integration into Github / Gitlab directly
- Export to central dashboard
- Statistical comparisons over time
- PR-triggered measurements
- Support for all major CI / CD systems
- FOSS - MIT License


Open Added no_python flag and ported flags to mac #698
ArneTR wants to merge 2 commits into main from no-python

Added no python flag, which was previously in no_build to github tests Verified 8e25f96

ArneTR requested a review from dan-mm 5 days ago

github-actions bot commented 5 days ago

Eco-CI Output:

Label	avg. CPU utilization [%]	Total Energy [Joules]	avg. Power [Watts]	Duration [Seconds]
Total Run	23.0831	1256.59	3.55974	360
Measurement #1	23.3577	1256.59	3.55974	355

Energy graph:

Example for Github PRs: https://github.com/green-coding-berlin/green-metrics-tool/pull/653

Example for Django Measurements over time: https://metrics.green-coding.io/ci.html?repo=green-coding-berlin/django&branch=main&workflow=60545070

GREEN CODING;

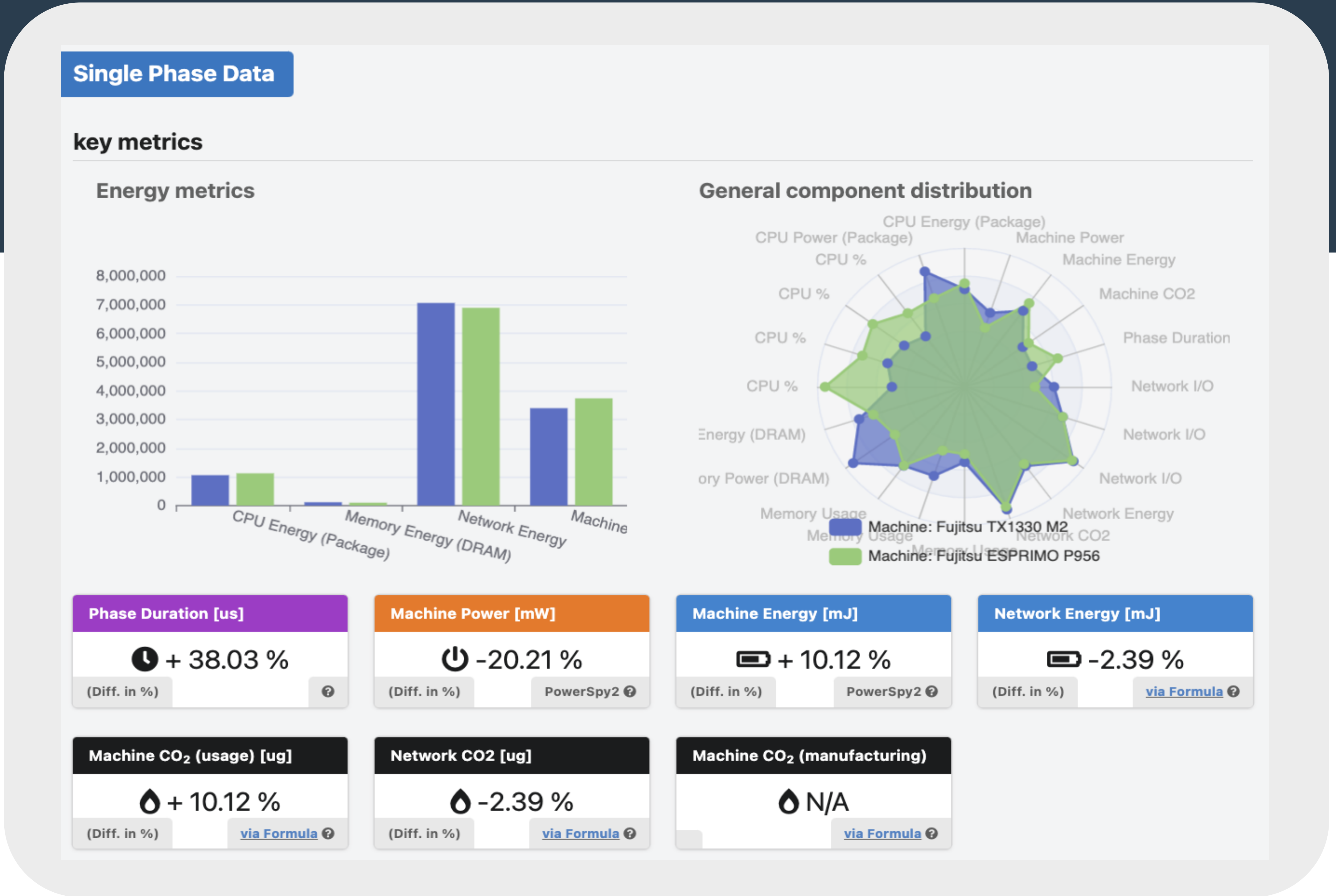
Green Metrics Tool

GREEN CODING;

Tech Slide

Green Metrics Tool

CO2- and Energy-Measurement



- Benchmarking via Standard-Usage-Scenarios
- (academia based via UCB / Öko-Insitut e.V.)
- Reproducibility / DevOps Integration via version control, orchestration and HW-Limits
- Dashboard with statistical comparison
- Modular / Extendable reporters for: Energy (Components, Machine), CO2 (Grid Intensity), Network, Memory, Temperature, Frequency, Embodied Carbon, VMs, distributed architectures ...
- SCI (ISO-Norm) / Blauer Engel compatible (German Eco Label) / ISO 14001/ GHG protocol
- Cluster-Automation with Measurement-Accuracy-Control
- FOSS - AGPL 3.0 License
- Online Dashboard: https://metrics.green-coding.io/index.html

GREEN CODING;

Green Metrics Tool

Create CO2 and Energy profiles of applications

Nextcloud
CMS *click for details*

BADGES

- Energy Cost 4.89 kJ via PSU (AC)
- Energy Cost 1.94 kJ via RAPL
- SCI 162.64 mgCO2e/Talk message

Show measurements

Wagtail
CMS *click for details*

BADGES

- Energy Cost 1.14 kJ via PSU (AC)
- Energy Cost 243.24 J via RAPL
- SCI 18.84 mgCO2e/page request

Show measurements

Wordpress
Blog/CMS *click for details*

BADGES

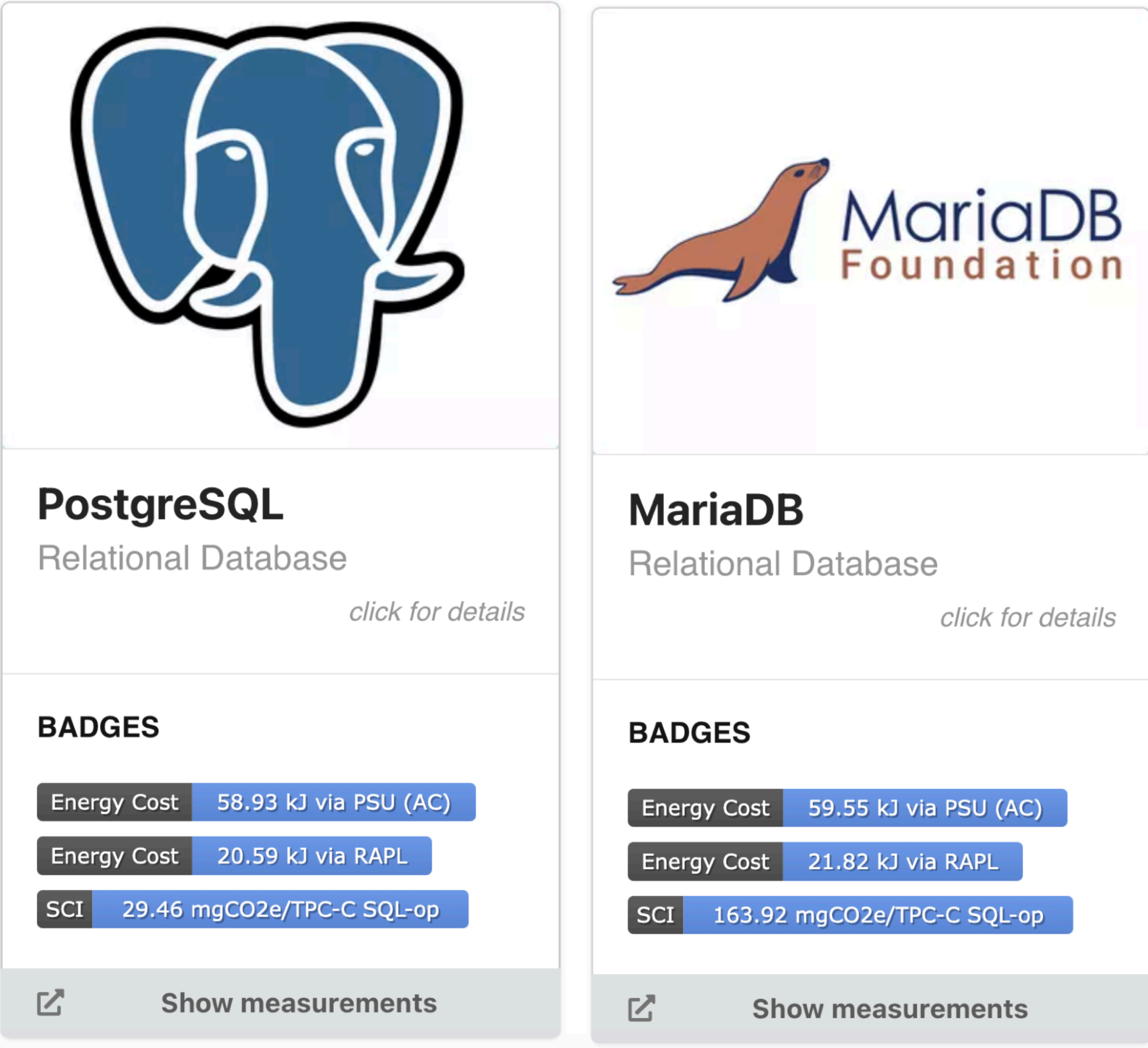
- Energy Cost 635.62 J via PSU (AC)
- Energy Cost 112.62 J via RAPL
- SCI 25.08 mgCO2e/page request

Show measurements

Green Metrics Tool

Quantify, Identify and Optimize Software Applications

- Detailed component break-down of an application
- Software with identical behaviour can have a vastly different carbon profile.
- Potential:
    - Profiling before applications go to production
    - Selection of dependencies
    - Standard Software recommendations for Org.

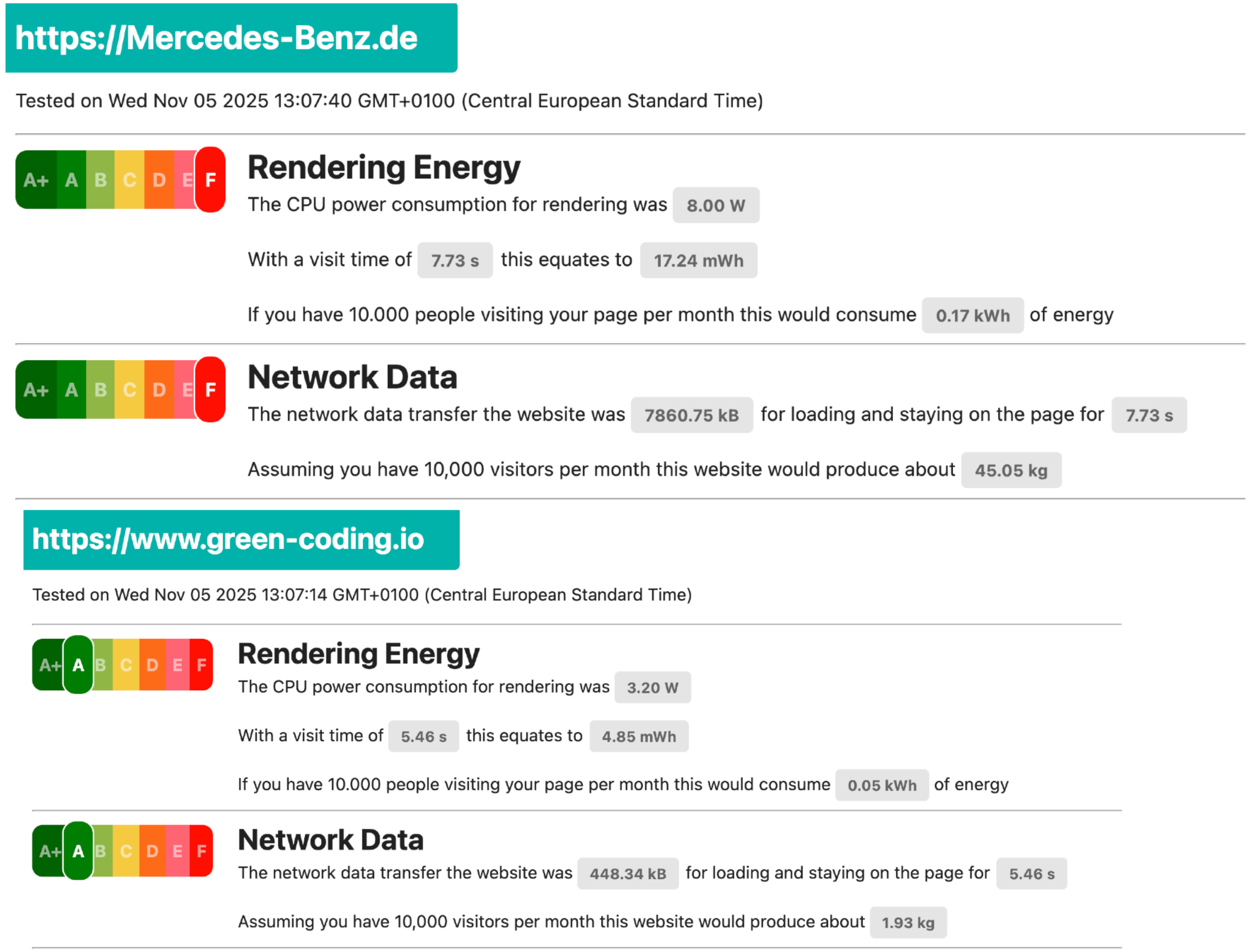


GREEN CODING;

Green Metrics Tool

Understanding outwards visibility: CO2 emissions of Front-Ends, APIs und Websites

- Network data and client-side rendering (e.g., through browsers/apps) also cause CO2 emissions.
- Front ends differ greatly in this regard.
- Front ends contribute significantly to the CO2 profile of an application.
- Emissions are visible and measurable and contribute to the external impact.



Images from https://website-tester.green-coding.io - Try it out! :)

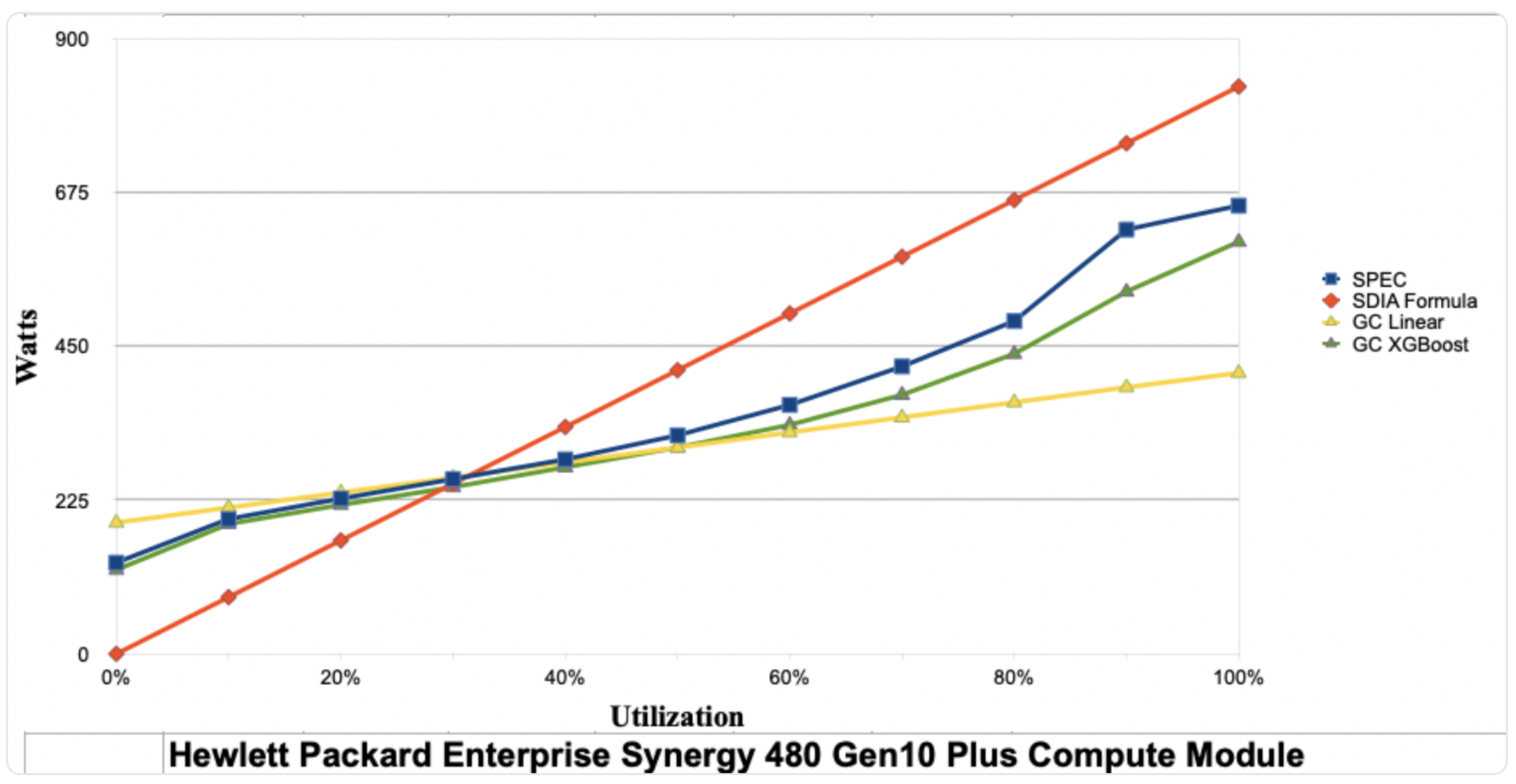
GREEN CODING;

Cloud Energy

Tech Slide

Cloud Energy
Estimation for Cloud and VMs

- The setup of the model is based on a research paper from Interact DC and the University of East London
- 90%+ Accuracy in/out-of sample
- Near 0% overhead
  - XGBoost implementation
  - POSIX stream compliant
- Usable in minimalistic environments cross-platform (Linux Microcontrollers / Automotive) with only native OS dependencies
- FOSS - MIT License



GREEN CODING;

Ok, nice tool overview

How do I see the bigger picture now?

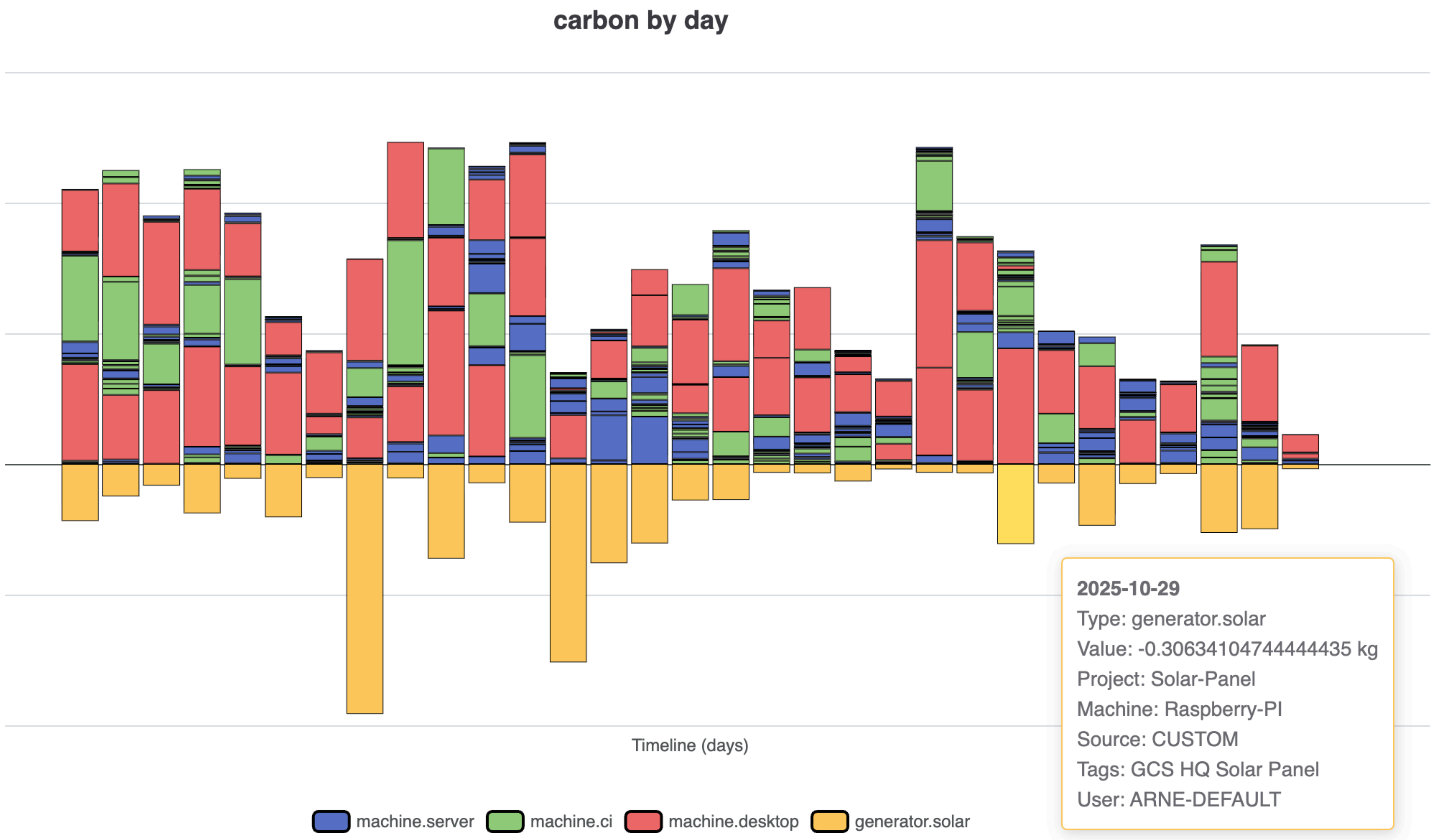
Carbon DB

GREEN CODING;

CarbonDB

Integrating carbon data sources to a centralised dashboard

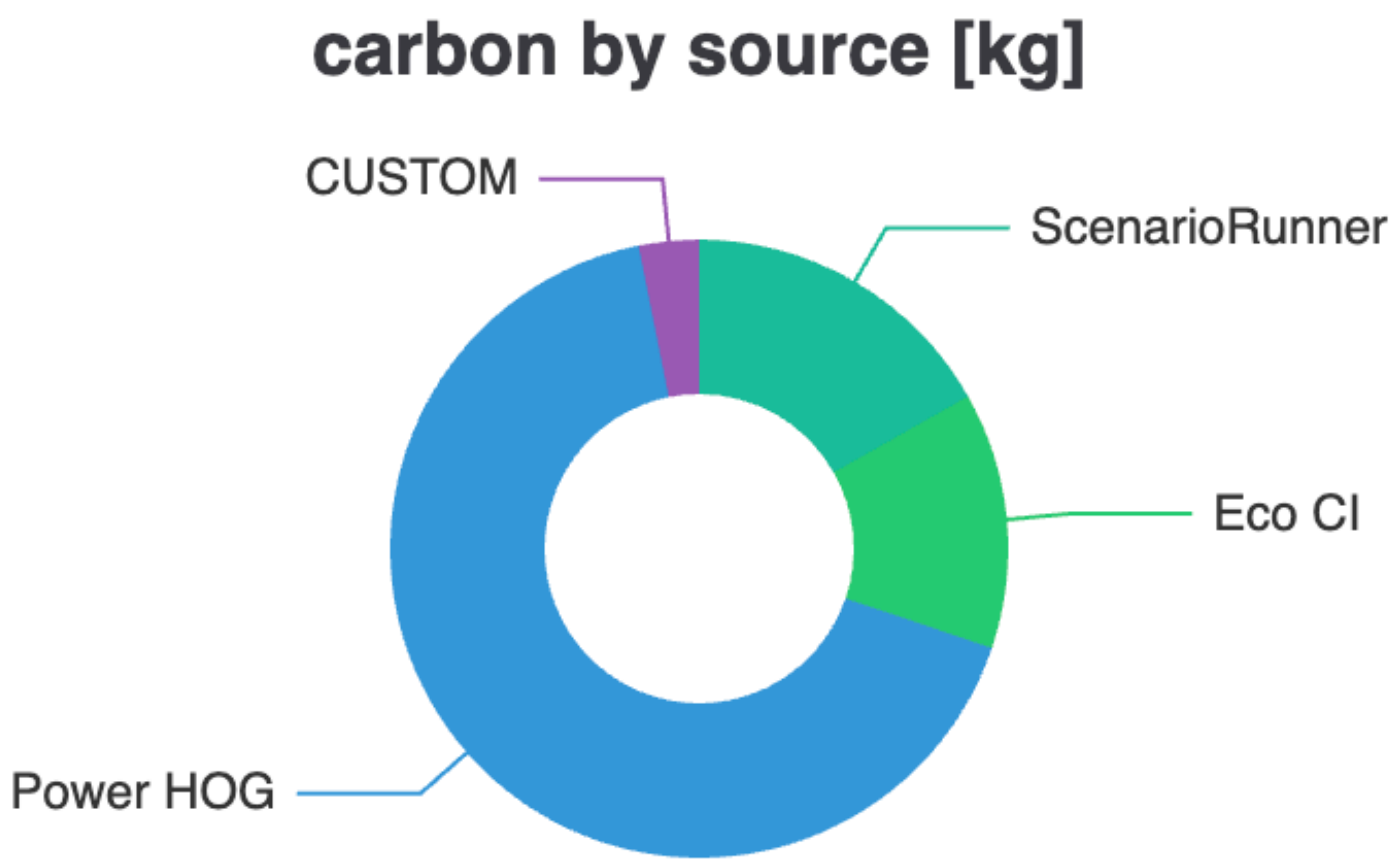
- Aggregation Dashboard for GCS Tool eco system
- Supports arbitrary data inputs also (e.g. own power generation)
- Supports also interfacing with already existing Dashboards like Grafana through API Gateway

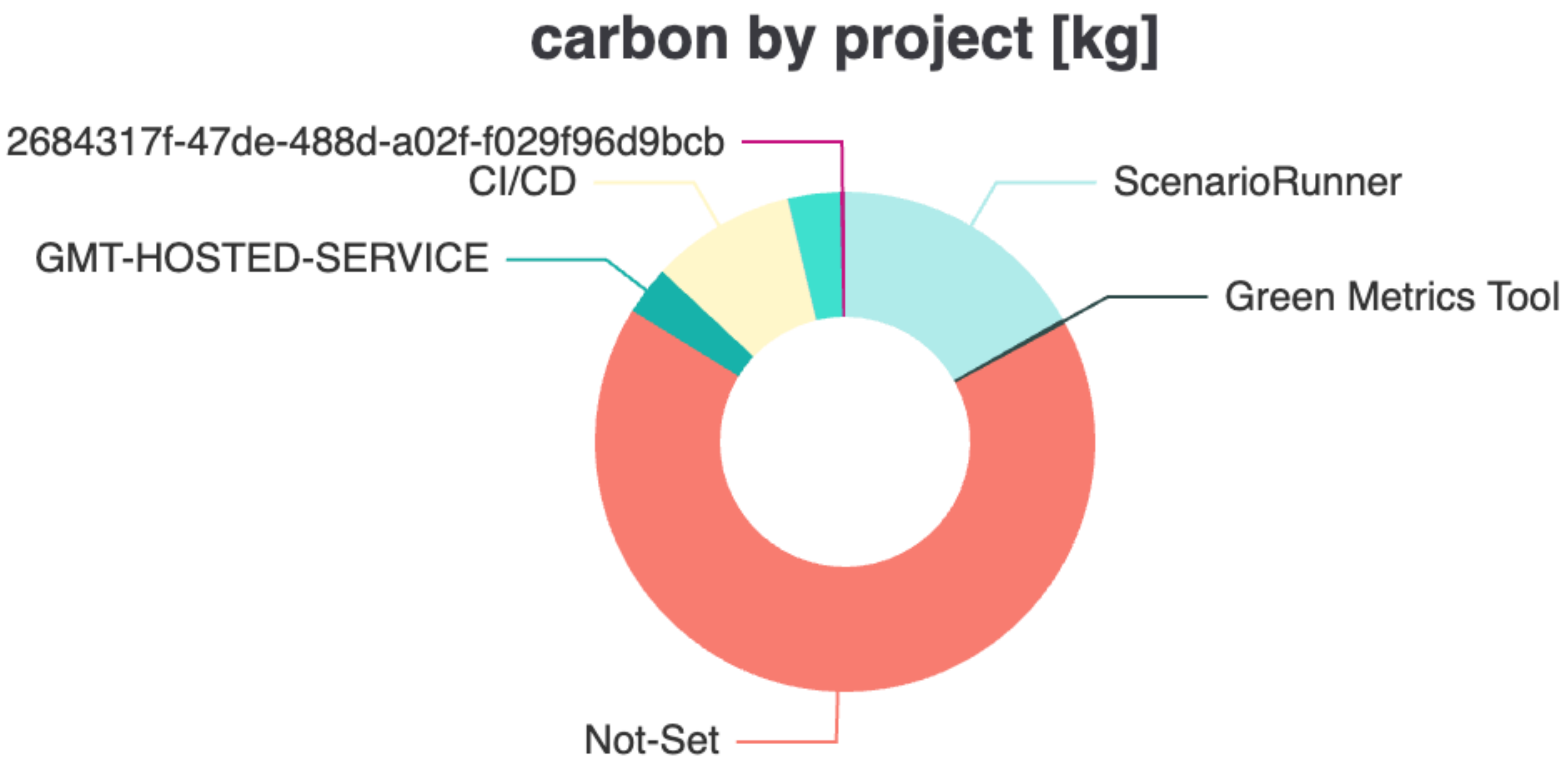


GREEN CODING;

CarbonDB

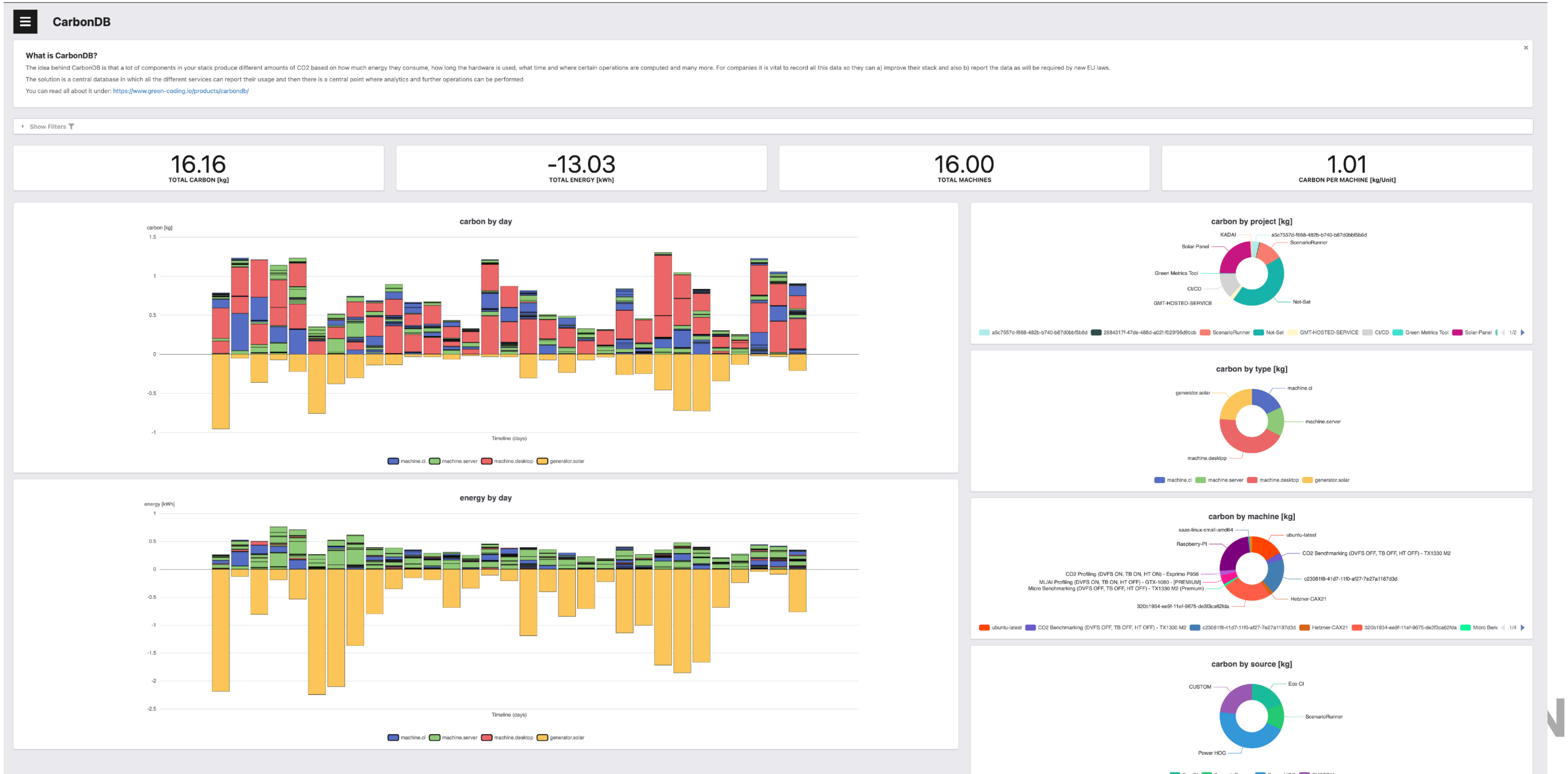
Drill-Downs by user, machine, project, team etc.





Full client view

Anonymized



Summary

What does CarbonDB allow

- Get Insights into **real time relation** of **embodied** vs. **operational** emissions
- Identify **most emitting teams / products**
- Understand how **load shifting / location shifting** impacts your IT infrastructure footprint
- Easily create **monthly reports**
- **Integrate any IT component that can make a web request**

- **Try it out, it is open source!**

GREEN CODING;

Thank you

Want to know more?

- Website / Blog / Newsletter: https://www.green-coding.io
- Demo Open Data Repository: https://metrics.green-coding.io
- Our tools: https://www.green-coding.io/#products
- Our case studies: https://www.green-coding.io/case-studies
- Meetup group (Berlin / Remote): https://www.meetup.com/green-coding
- E-Mail: arne@green-coding.io

GREEN CODING;

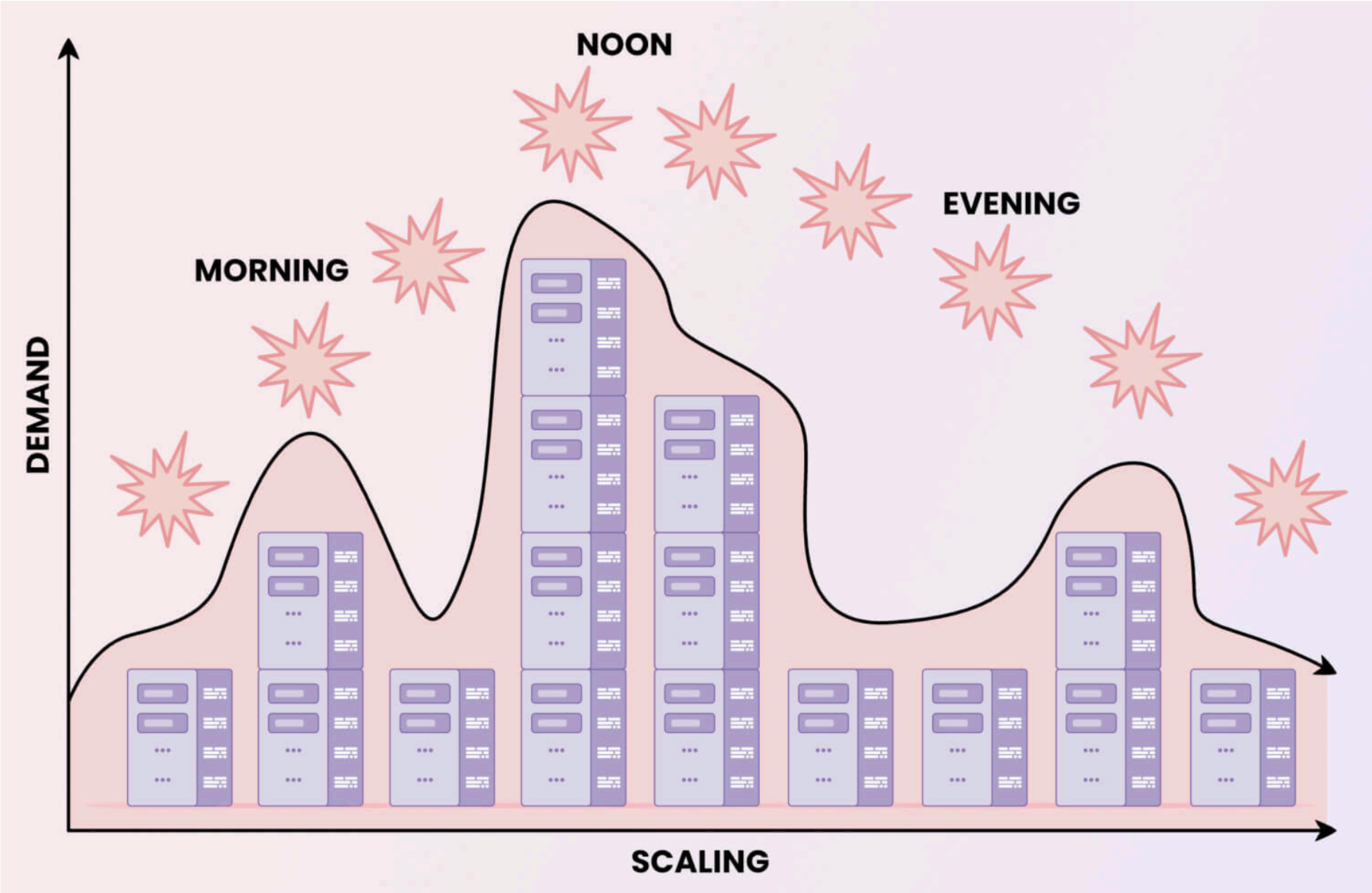
Optimization Potentials

How and where to reduce - Low hanging fruits

Potentials in Architectures

Dynamic scaling VMs - Kubernetes / Docker Swarm etc.

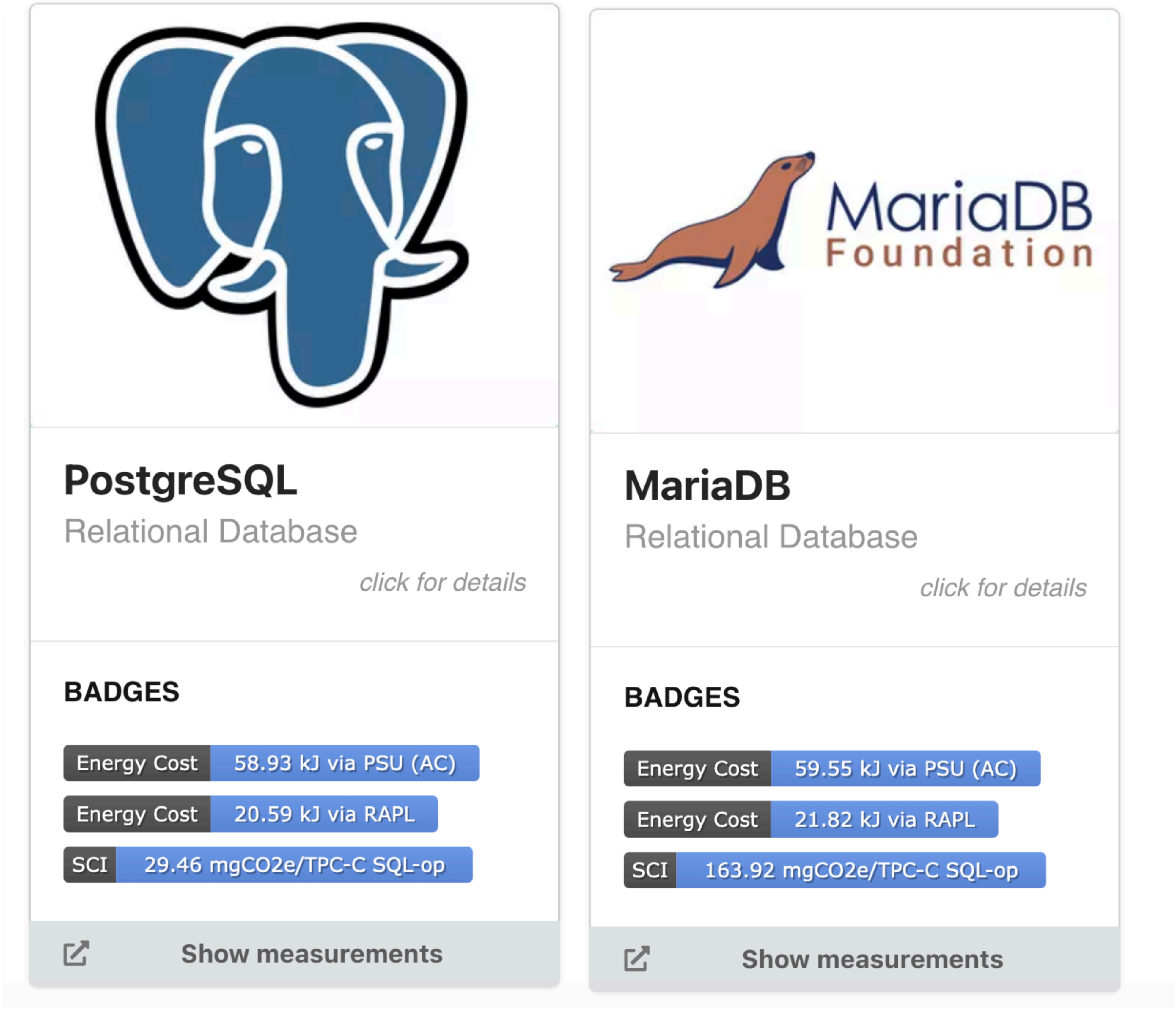
- Machines turn off when no user is using them
- Machines turn on when demand is increasing
- Scale-to-zero architecture can save up to 100% of avoidable emissions (idle time)
- Overhead can often be merged to existing load balancers



Potentials in dependencies / components

Quantifying dependencies and toolchains in terms of carbon

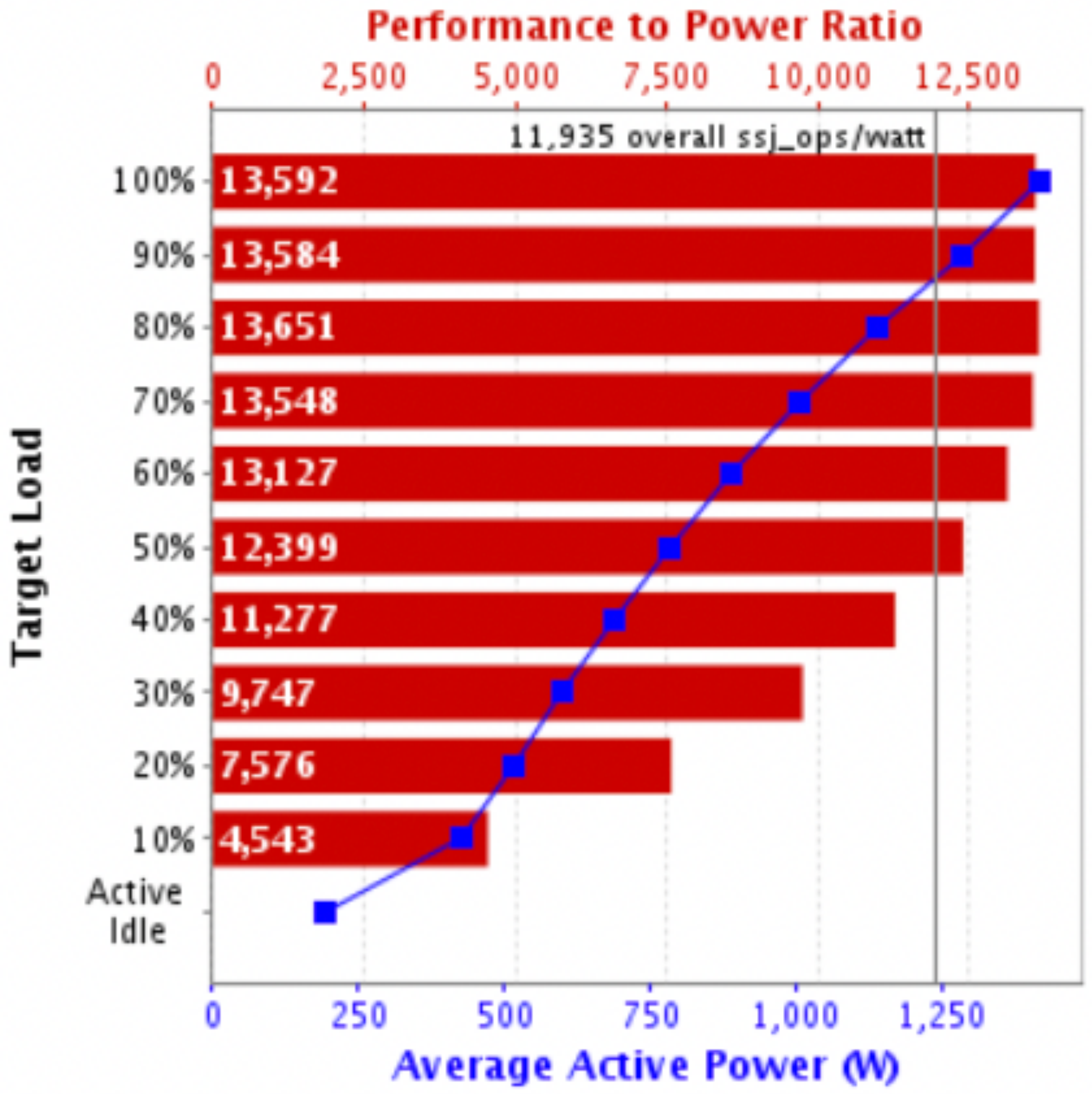
- Software carbon and energy transparency
- Different software products for same use-case can have vastly different energy and carbon profiles
- Levers:
    - Library / Dependency selection
    - Default software recommendations

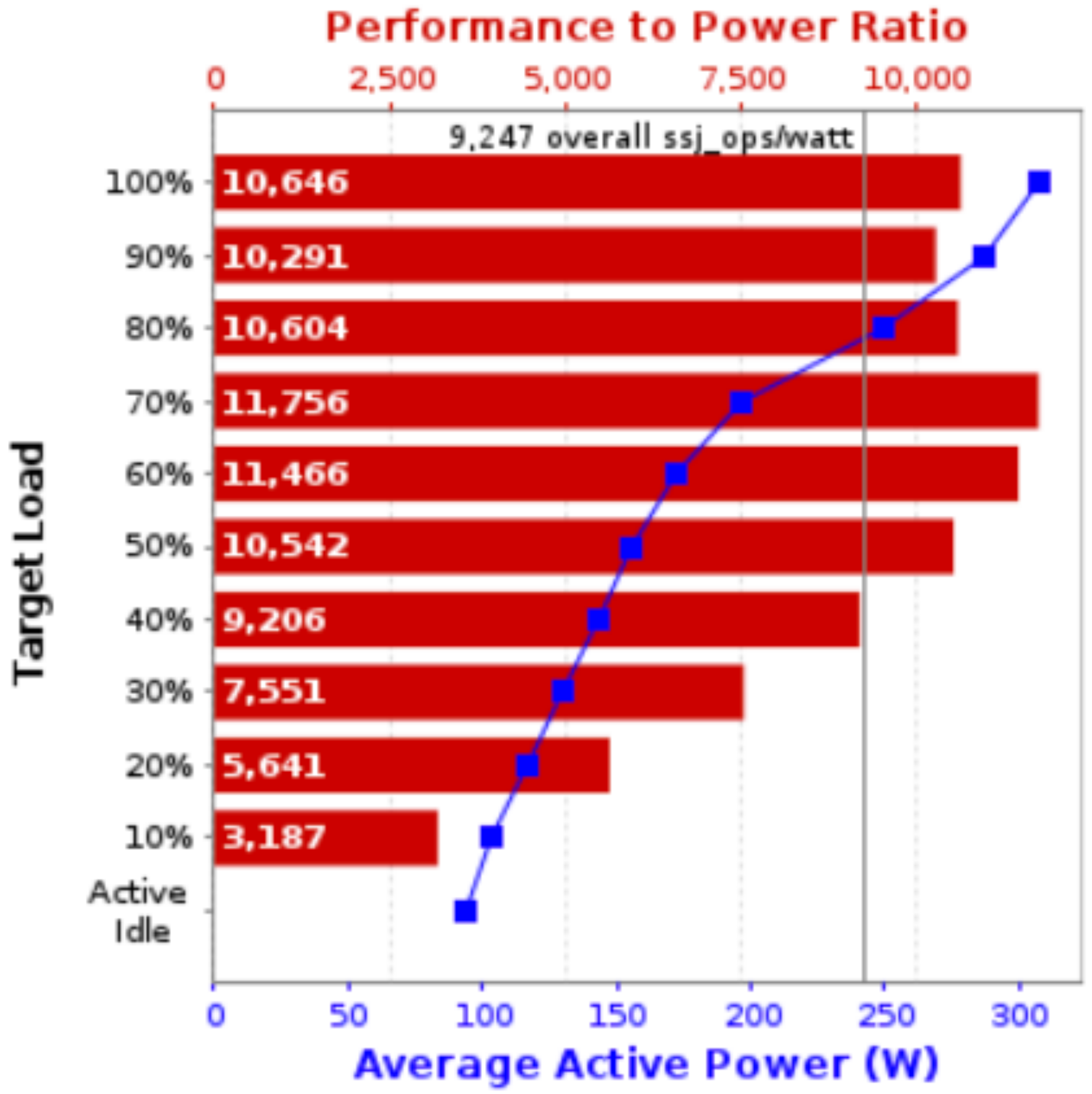


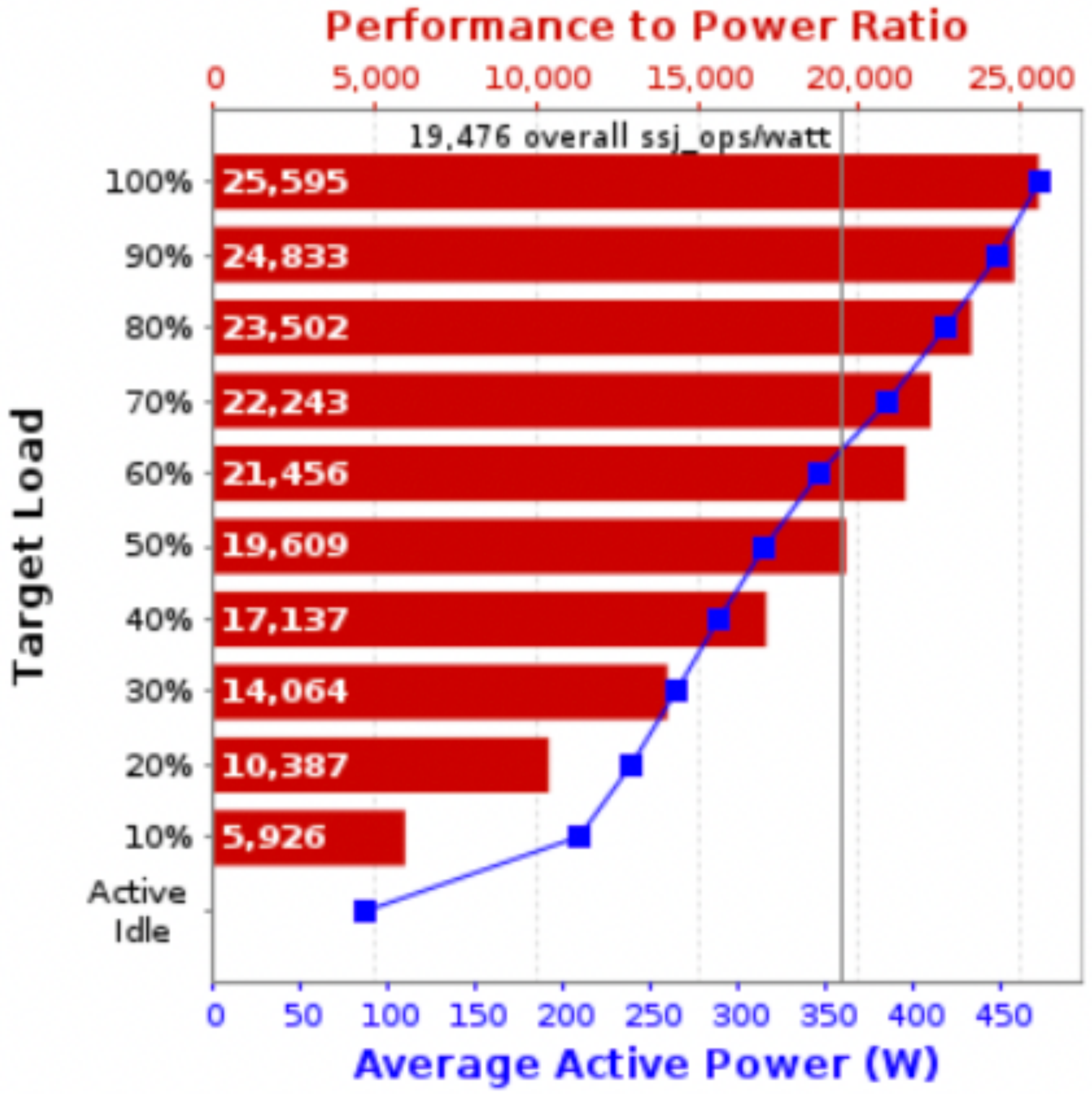
GREEN CODING;

Potentials of different hardware configuration

Power behaviour of hardware and how to leverage non-linearity. Cloud and non-cloud!







GREEN CODING;

Programming language potentials

Energy efficiency of can differ significantly

- Greenlab study has looked at different programming languages for a pure compute task

- Comparison of energy, runtime and memory footprint

- C# 3x slower than C

- Javascript 6x runtime vs. C, but only 4x energy

- Python 75x more energy than C, 25x of C#
  => Does NOT account for C machine learning libraries

Table 4. Normalized global results for Energy, Time, and Memory

Total					
	Energy		Time		Mb
(c) C	1.00	(c) C	1.00	(c) Pascal	1.00
(c) Rust	1.03	(c) Rust	1.04	(c) Go	1.05
(c) C++	1.34	(c) C++	1.56	(c) C	1.17
(c) Ada	1.70	(c) Ada	1.85	(c) Fortran	1.24
(v) Java	1.98	(v) Java	1.89	(c) C++	1.34
(c) Pascal	2.14	(c) Chapel	2.14	(c) Ada	1.47
(c) Chapel	2.18	(c) Go	2.83	(c) Rust	1.54
(v) Lisp	2.27	(c) Pascal	3.02	(v) Lisp	1.92
(c) Ocaml	2.40	(c) Ocaml	3.09	(c) Haskell	2.45
(c) Fortran	2.52	(v) C#	3.14	(i) PHP	2.57
(c) Swift	2.79	(v) Lisp	3.40	(c) Swift	2.71
(c) Haskell	3.10	(c) Haskell	3.55	(i) Python	2.80
(v) C#	3.14	(c) Swift	4.20	(c) Ocaml	2.82
(c) Go	3.23	(c) Fortran	4.20	(v) C#	2.85
(i) Dart	3.83	(v) F#	6.30	(i) Hack	3.34
(v) F#	4.13	(i) JavaScript	6.52	(v) Racket	3.52
(i) JavaScript	4.45	(i) Dart	6.67	(i) Ruby	3.97
(v) Racket	7.91	(v) Racket	11.27	(c) Chapel	4.00
(i) TypeScript	21.50	(i) Hack	26.99	(v) F#	4.25
(i) Hack	24.02	(i) PHP	27.64	(i) JavaScript	4.59
(i) PHP	29.30	(v) Erlang	36.71	(i) TypeScript	4.69
(v) Erlang	42.23	(i) Jruby	43.44	(v) Java	6.01
(i) Lua	45.98	(i) TypeScript	46.20	(i) Perl	6.62
(i) Jruby	46.54	(i) Ruby	59.34	(i) Lua	6.72
(i) Ruby	69.91	(i) Perl	65.79	(v) Erlang	7.20
(i) Python	75.88	(i) Python	71.90	(i) Dart	8.64
(i) Perl	79.58	(i) Lua	82.91	(i) Jruby	19.84

GREEN CODING;

AI Model Inference

Model difference

Green Coding AI

Show in Joule

Welcome to the Green Coding AI.

AI is spreading like a wildfire but at what cost? This project aims to show how much asking an GPT uses in resources. Please checkout the FAQs.

You

Tell me a short joke plase

deepseek-r1

<think> Alright, so I need to come up with a short joke. Hmm, where do jokes usually come from? I guess they often rely on wordplay or puns because those can be pretty quick once you get the hang of them. Okay, let's think about something everyone knows. Maybe school-related since that's a common theme in jokes. Why did the math book look so sad? Because it had too many problems! Wait, does that make sense? Yeah, "problems" is both plural and a pun on the word for issues. It's simple and gets the point across without being too complicated. I should check if there are any better or funnier options, but this seems to work. It's got a clear setup and a punchline with a twist. I think that's good enough for a short joke. </think> Why did the math book look sad? Because it had too many problems!

8.76	68.05	113.13	0.4786	31.62
CPU WATT	GPU WATT	MACHINE WATT	GCO$_2$EQ EMITTED (OPERATIONAL + EMBODIED)	TIME SECONDS

https://green-coding.ai/?key Share

GREEN CODING;

AI Model Inference

Model difference

Green Coding AI

Show in Joule

Welcome to the Green Coding AI.

AI is spreading like a wildfire but at what cost? This project aims to show how much asking an GPT uses in resources. Please checkout the FAQs.

You

tell me a joke

llama3

```
Here's one: Why couldn't the bicycle stand up by itself? (Wait for it...) Because it was two-tired! Hope that made you smile! Do you want to hear another one?
```

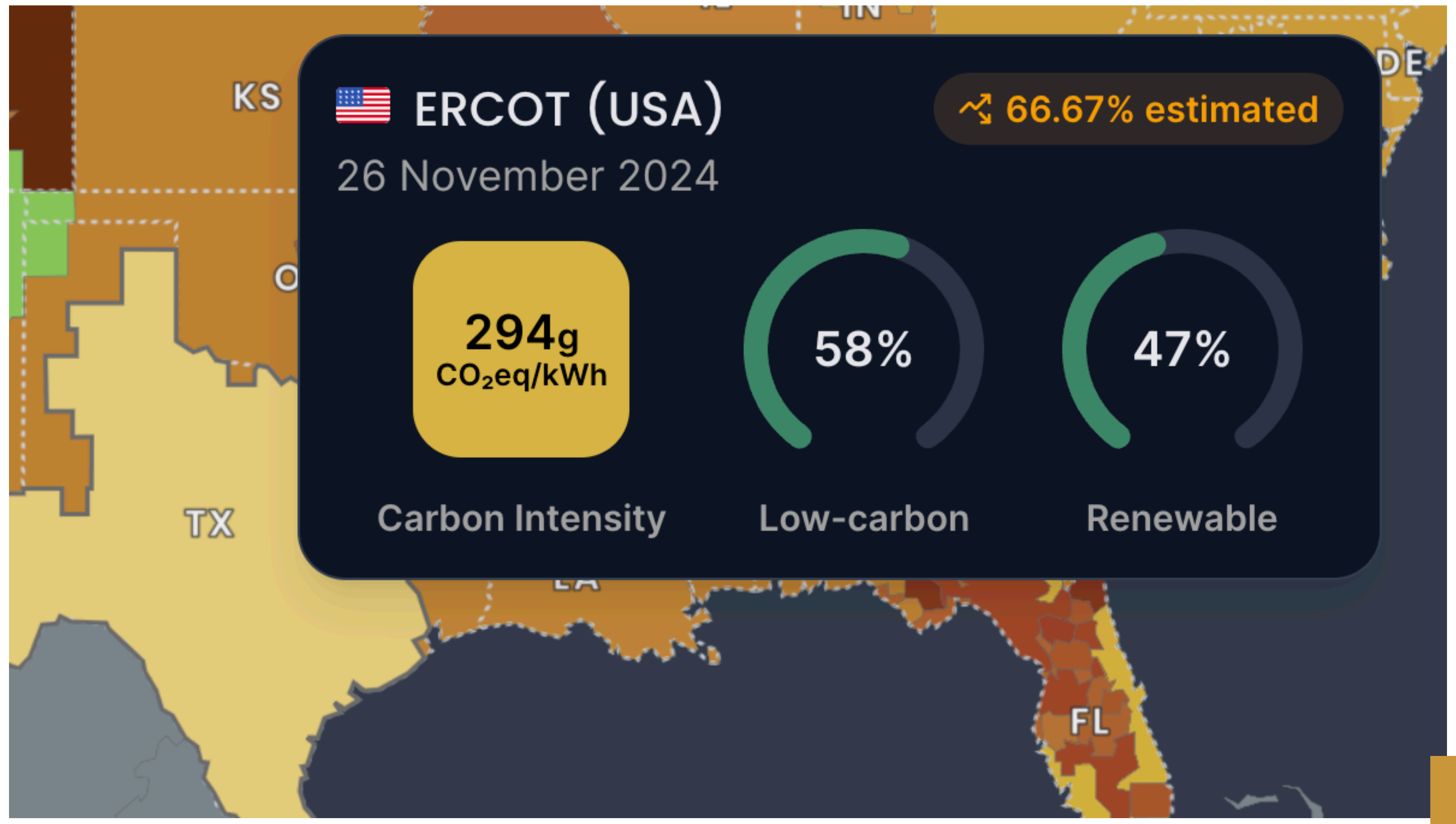
6.58	47.07	84.37	0.3209	27.54
CPU WATT	GPU WATT	MACHINE WATT	GCO_2EQ EMITTED (OPERATIONAL + EMBODIED)	TIME SECONDS

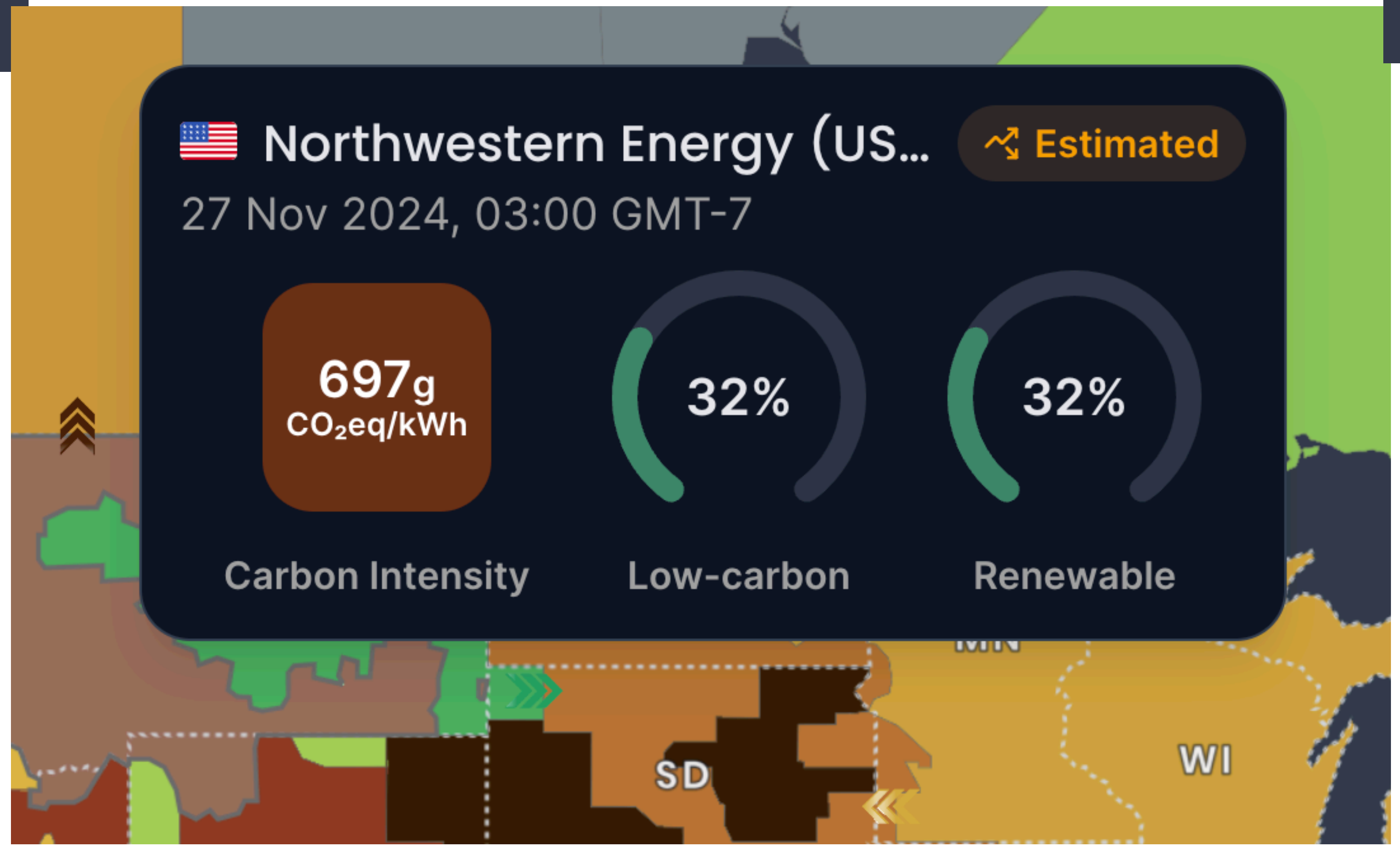
https://green-coding.ai/?key Share

GREEN CODING;

Potential: Save carbon by location shifting

How much can you save by selecting beneficial DC locations?

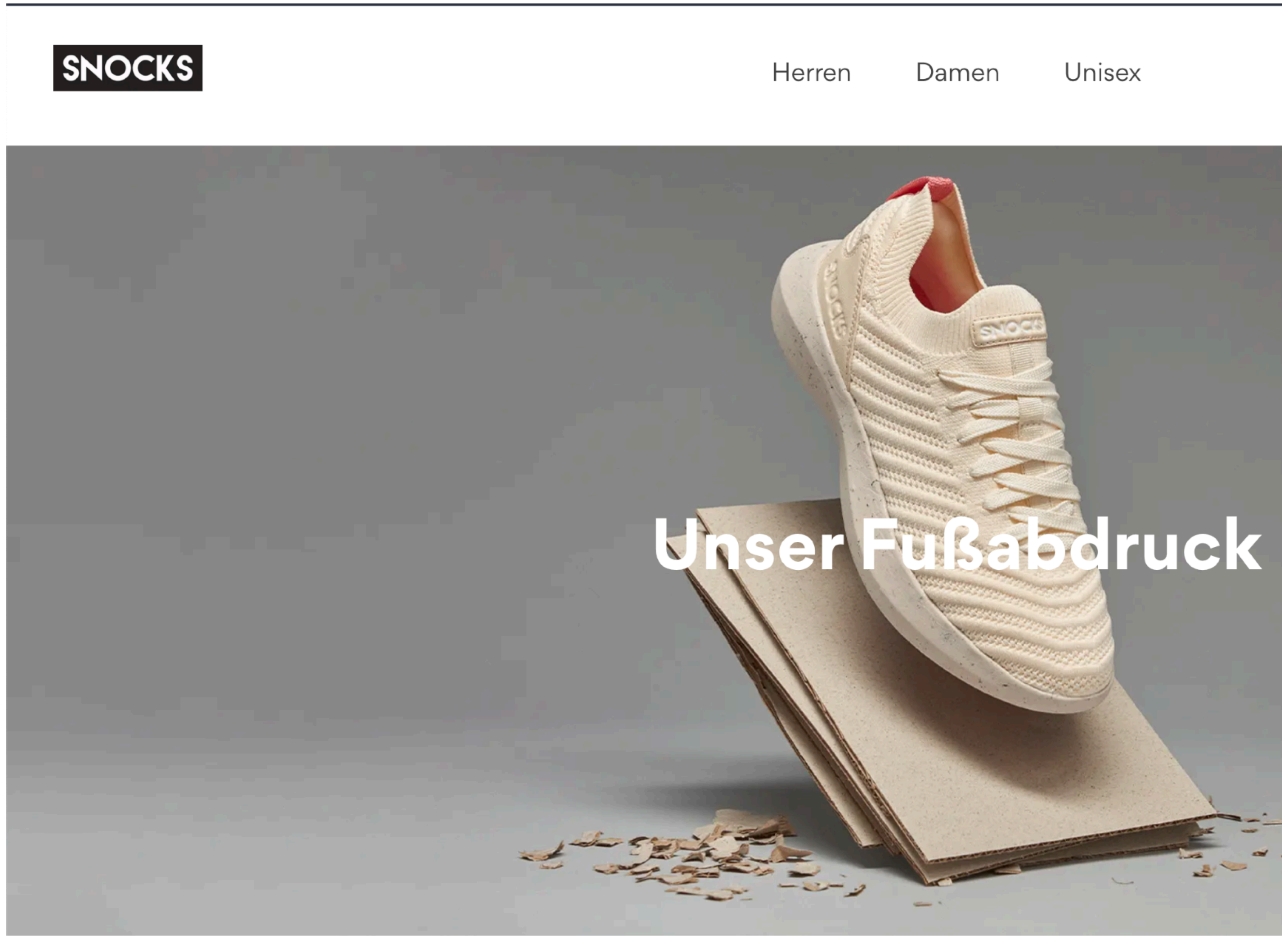


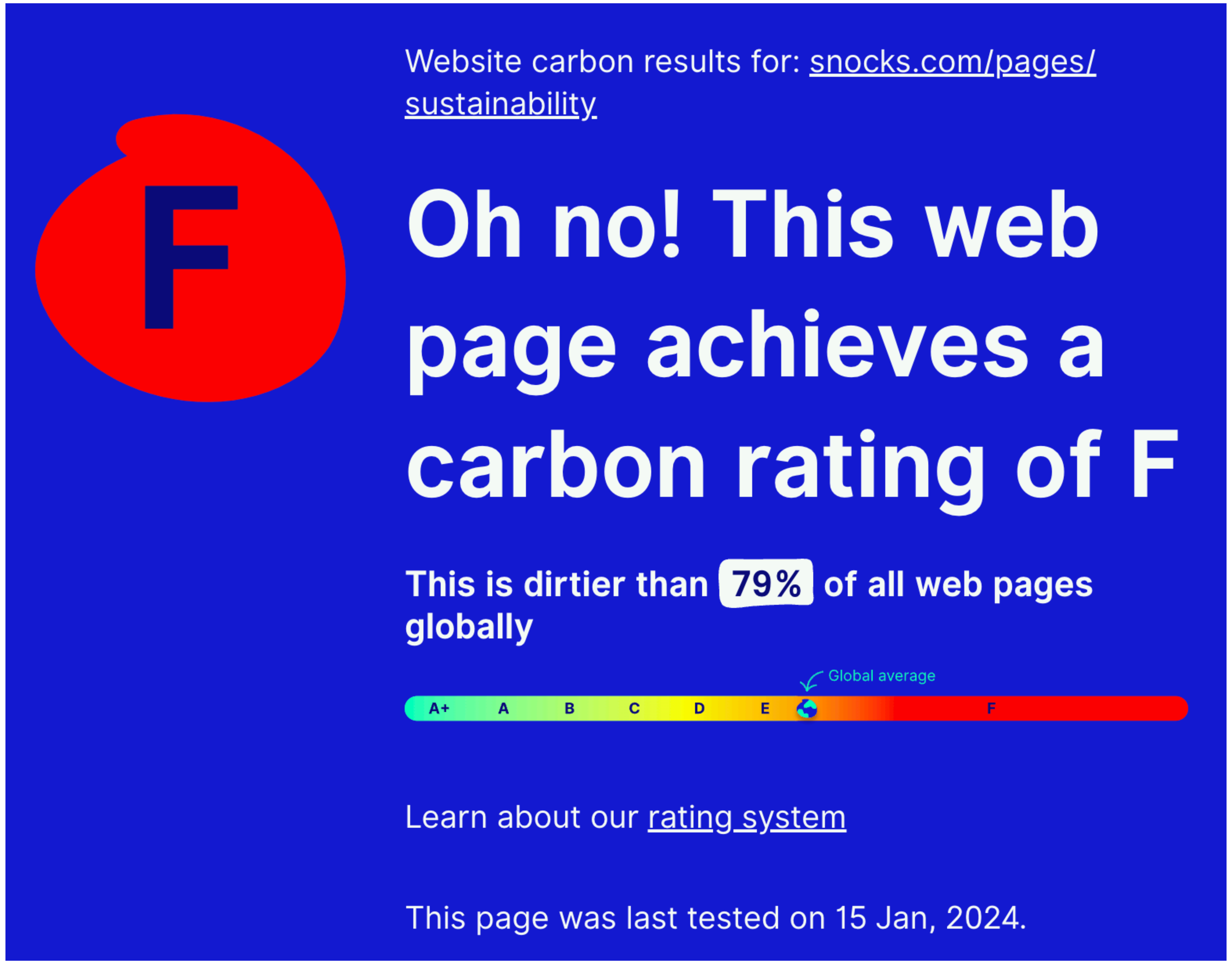


GREEN CODING;

Potential: Avoid a bad PR case

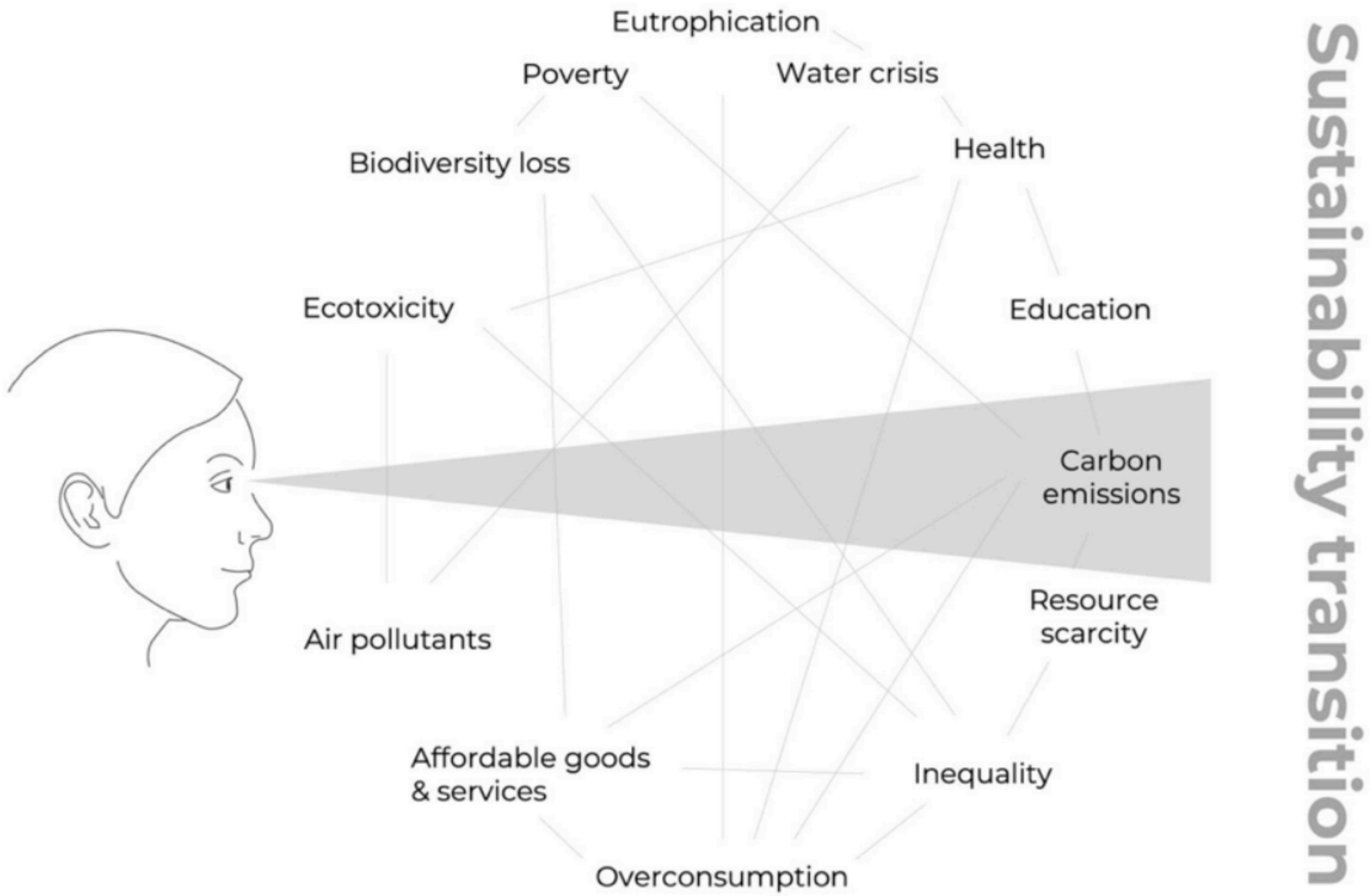
For your customer facing digital services





Carbon tunnel vision

Are we there yet?



GREEN CODING;