Sustainable Software

How can we quantify and measure "green-ness" of code?

</> GREEN CODING;

Standard slide for starters

Quick info - Arne Tarara / Green Coding Berlin

- Software-Dev 16+ years
- Founder & CEO **Green Coding Berlin GmbH**
  - Digital Infrastructure electricity & emissions research & consulting
  - Active open source tools developers & contributors

GSD COMMUNITY

SDIA SUSTAINABLE DIGITAL INFRASTRUCTURE ALLIANCE

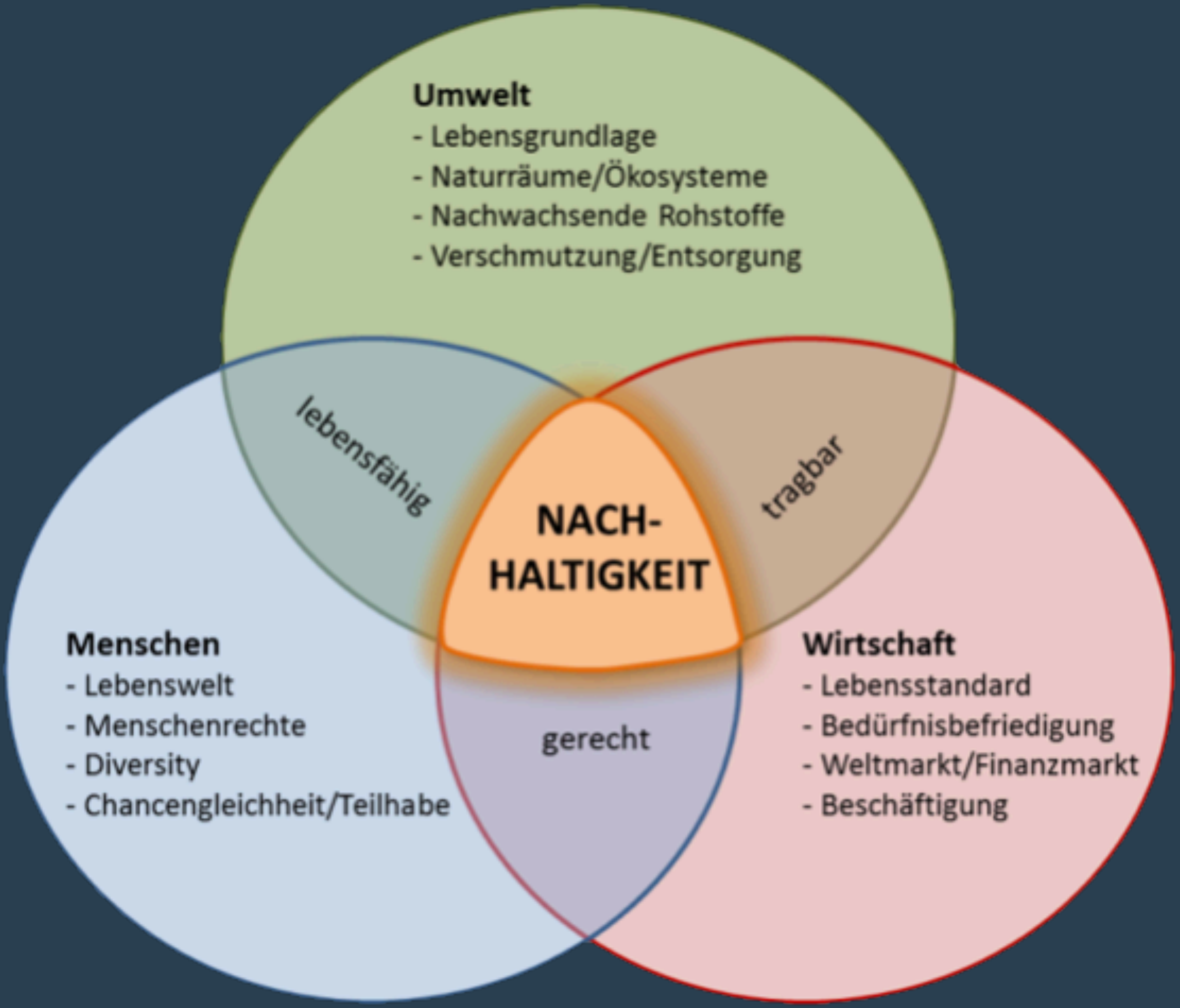
What is the definition of sustainability ...

according to other peoples definitions

- The Brundtland report from the **United Nations (UN)** defines sustainable development as the ability to :
  - *“meet the present needs without compromising the future generation abilities for their own needs“*
- General understanding often says:
  - *... the ability to refill itself at a quicker rate than it is consumed / damaged ...*

What is the definition of sustainability ...

in a more general way



Credit: sdialliance.org (SDIA)

What is the definition of sustainability ...

in terms of software in particular?

- None of this translates to software and the ICT industry very well ...
- Software cannot "replenish itself" on it's own
- The ICT sector is a growing business and we want it to be (digitalization = progress and sustainability)

How can we let ICT grow ...

... without inflicting future generations chances and opportunities?

Cloud Carbon Footprint

JoularJX

KEPLER

Do we need more tools?

Scaphandre

CODE CARBON

Green Metrics Tool

Do we need more guidelines?

Voices from the industry and academia

- **Architectural Tips**
  - Micro-Services
  - Load Balancers
  - Scale-to-Zero Architectures
- **GSF Guidelines**
  - Move to the Cloud
  - Carbon Awareness (Time- / Location-Shifting)
  - Right-Size VMs
  - Use less energy (sic!)

- **50 years of performance engineering**
  - compiled languages vs. dynamic
  - caching, bandwith-increase, compression ...
  - loop unrolling, memory layouting, cache locality ...
  - ASCPEM™

Seems like there is already a lot …

So what helps us bring these techniques into effect?

Seems like there is already a lot ...

So what helps us bring these techniques into effect?

Answer: An actual shortage of resources!

Seems like there is already a lot …

So what helps us bring these techniques into effect?

Answer: An actual shortage of resources!

:) No. This is exactly what we as Green Software advocates are trying to avoid

One word on monetary incentives

Typical thinking says that Green Software will only fly if companies also save $$$ when using them.

That is *true*, but in a scaling business this approach is *futile* when absolute numbers are a concern

So what can we do?

Approach #1:

Laws

- **German datacenter law (Energy efficiency act)**
    - Strong validated gains, but backlashes unclear

- **CSRD**
    - Corporate sustainability reporting directive
    - Implication for software unknown
    - Does only snapshots

Approach #2:

Labels

- **Blue Angel for Software**
  - Limited applicability
  - Transparency and distributor commitments
- **Green Software Design Label**
  - Broad applicability
  - Transparency and best practices label
- **Website labels (websitecarbon.com, nachhaltige-website.de, etc.)**
  - Network transfer based assumptions
  - No actual gain in the moment. If server is slow it might even be worse than higher-weight site.

Approach #3:

(Best) Practices

- **Example: CAST / ecoCode / ASCPEM**
    - Best practices from academia
    - Usage validated through static code scanning - CI/CD
    - Unclear if gains by the "best practices" outweigh the scanning costs
- Example: performance engineering
    - Are usually for efficiency. Not for absolute saving in growth

Best Practices?

Even simple questions are hard (impossible) to answer

- **Is email more sustainable than paper?**
  - Paper consumes a fixed amount. Email has pot. infinite storage and processing
- **Is Serverless more sustainable than classic VMs?**
  - No solid data on this (Deno / Isolates / Firecracker)
  - Cloudflare / Amazon did decline when asking for sustainability insights
- **Is using AWS Gravitron more sustainable than Intel**
  - What happens to electronic waste?! Life-Cycle ...?
- **Is Python more sustainable than Rust?**
  - Python uses 80-times the Instructions where as Rust uses 1-3. Still people are not changing because of "cost of development" etc. ....

Quick summary

We need a different approach to sustainability / green-ness for software

Best-Practices currently tied to labels and certifications provide no guarantee and do not work in growth scenarios

Introducing: Software-Lifecycle-Assessment

Quantifying the sustainability of software

- Adding just a simple measurement to a label does not help
- We need a "constant" quantification. Same as in DevOps.
- The software has to be monitored throughout it's evolution.
- In all phases like development, runtime and after use (deletion, EOL, exporting etc.)
-

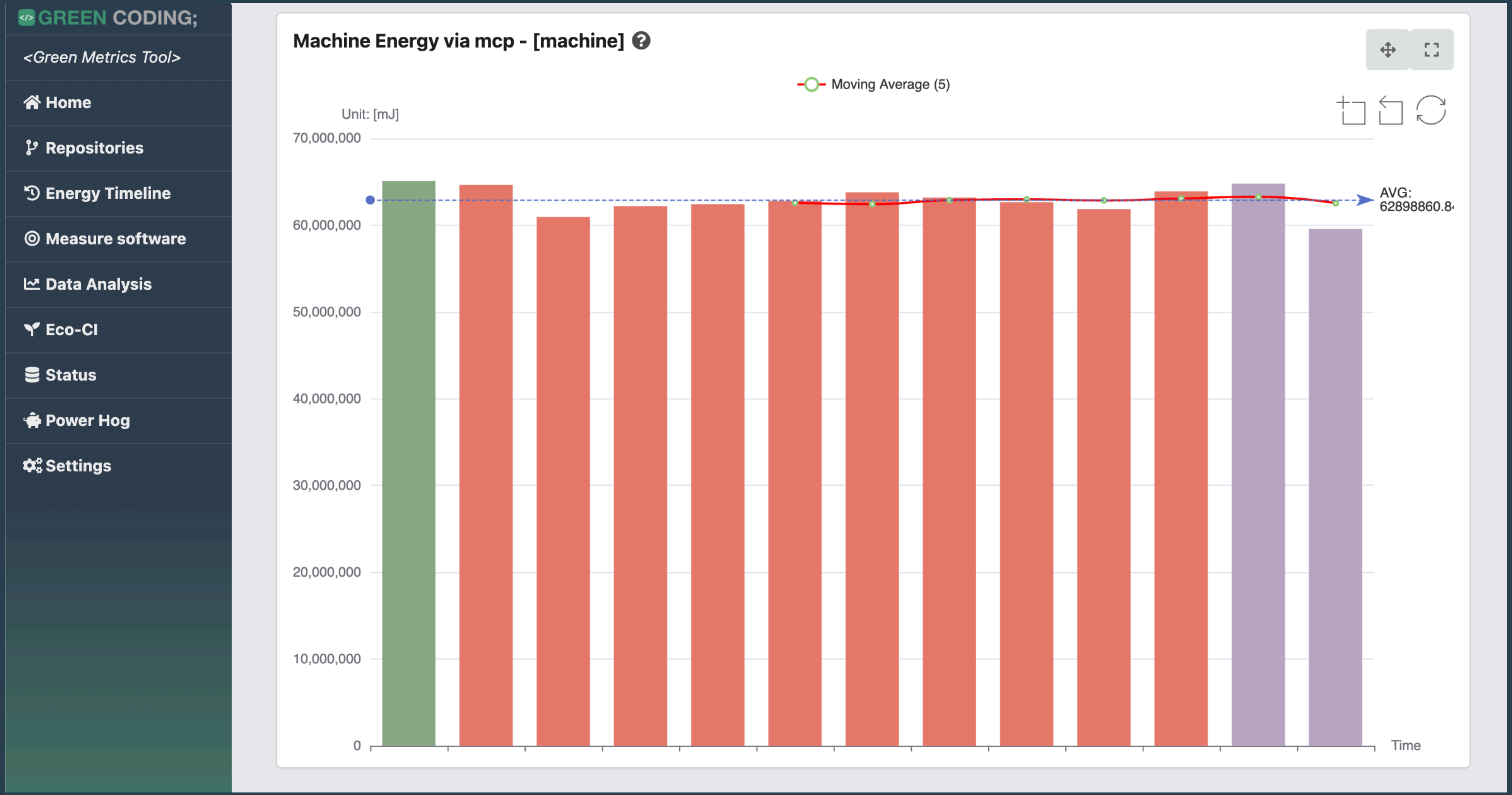
Introducing: Software-Lifecycle-Assessment

But what makes it now "green"?

- For a software to qualify as green it has to meet certain criteria:
  - All resource consumption has to be reported
  - The relative resource consumption slope must be negative to combat product growth (Absolute consumption stays level)
  - If comparative values emerge it has to achieve best-in-class

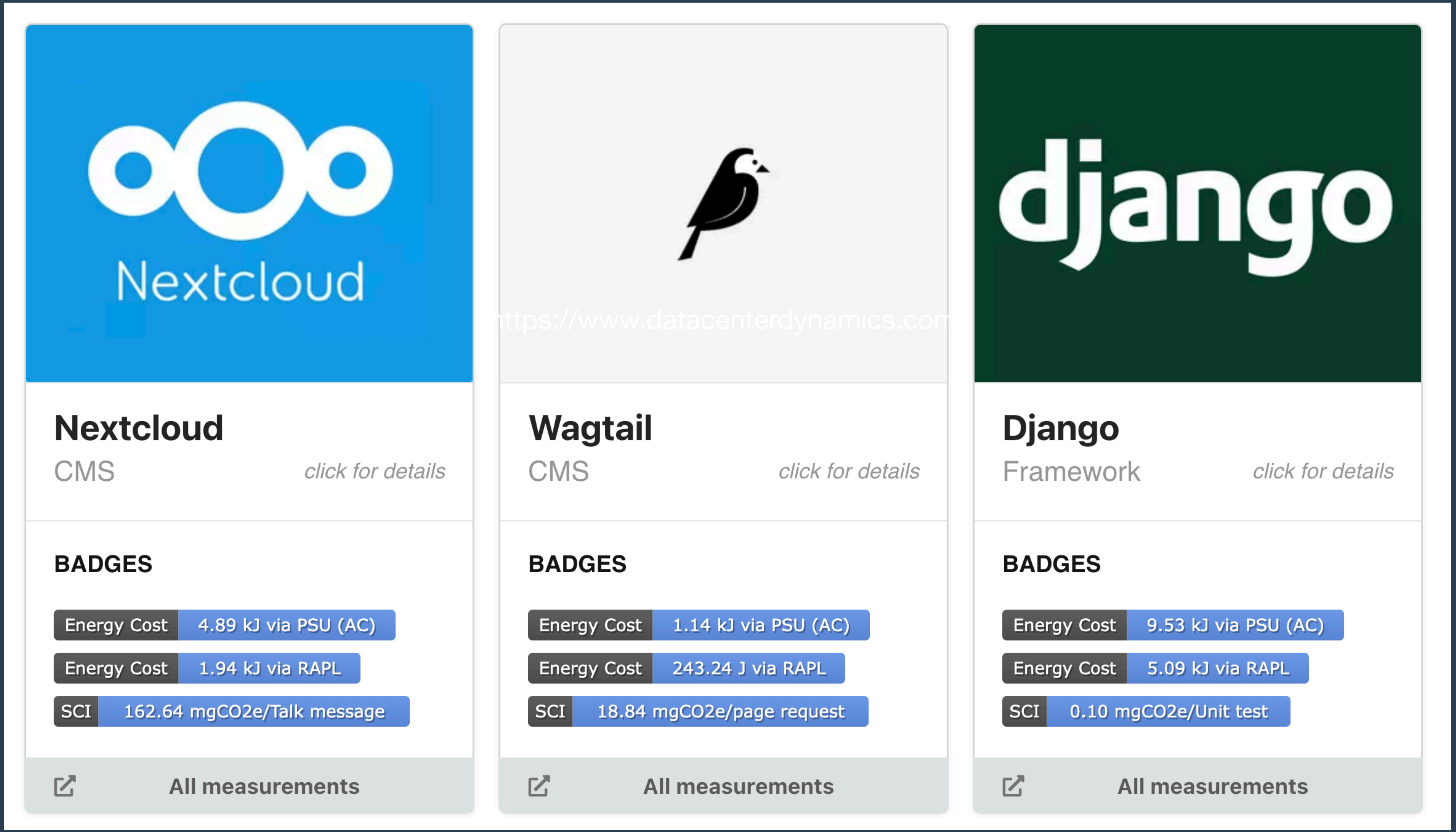
How to constantly quantify

An approach with our "Energy-Timeline" solution



How to get to absolute "best in class"

A solution with our "Energy-ID" project - Using core features as benchmark



ttps://www.datacenterdynamics.com

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

All 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

All measurements

Django
Framework
click for details

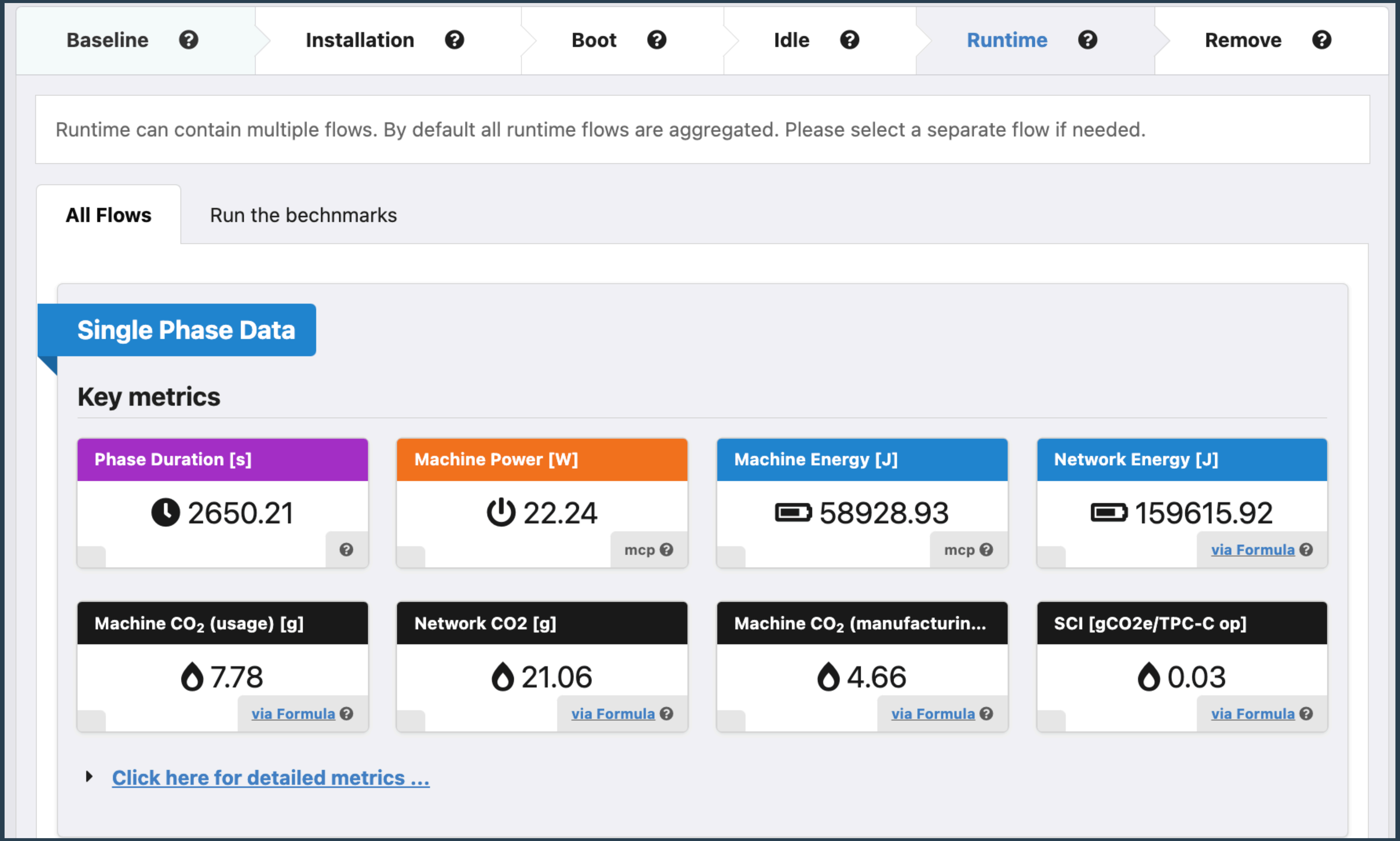
BADGES

Energy Cost 9.53 kJ via PSU (AC)
Energy Cost 5.09 kJ via RAPL
SCI 0.10 mgCO2e/Unit test

All measurements

How to get to absolute "best in class"

Using industry standard cases: TPC-C / Speedometer



Runtime can contain multiple flows. By default all runtime flows are aggregated. Please select a separate flow if needed.

Single Phase Data

Key metrics

Phase Duration [s]	Machine Power [W]	Machine Energy [J]	Network Energy [J]
2650.21	22.24 mcp	58928.93 mcp	159615.92 via Formula

Machine CO₂ (usage) [g]	Network CO2 [g]	Machine CO₂ (manufacturin...	SCI [gCO2e/TPC-C op]
7.78 via Formula	21.06 via Formula	4.66 via Formula	0.03 via Formula

Click here for detailed metrics ...

Sources

- GSF SCER: https://github.com/Green-Software-Foundation/sci/issues/359
- GSF SCI: https://sci-guide.greensoftware.foundation/
- GSF Best Practices: https://patterns.greensoftware.foundation/guide/
- IMDA & Microsoft Green Software: https://www.imda.gov.sg/-/media/imda/files/infocomm-media-landscape/sg-digital/microsoft-imda-digital-sustainability-guideline.pdf
- Green Coding Berlin - Energy ID: https://www.green-coding.berlin/projects/energy-id/
- Green Coding Berlin - Energy Timeline: https://metrics.green-coding.berlin/energy-timeline.html
- Wagtail Gold Standard: https://github.com/wagtail/wagtail/discussions/8843