Tools and best practices for Sustainable Software

An overview

GREEN CODING;

Standard slide for starters

Quick info - Arne Tarara / Green Coding Berlin

- Software-Dev 16+ years
- Founder & CEO **Green Coding Berlin GmbH**
  - We do research and development in open source tools for software resource consumption
  - We help companies assess their digital infrastructure resource consumption

SDIA SUSTAINABLE DIGITAL INFRASTRUCTURE ALLIANCE

</> GREEN CODING;

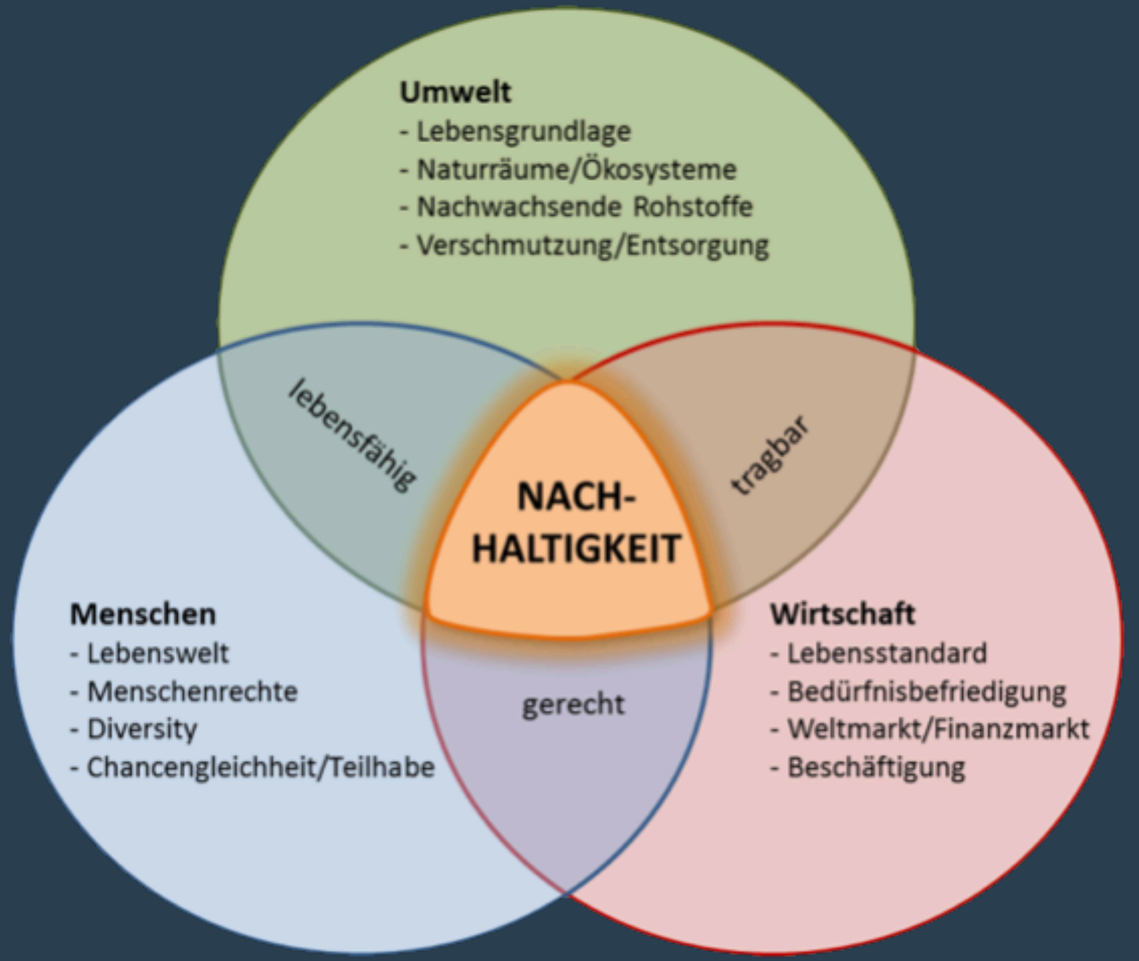
Agenda

25.04.2023

- What is sustainable software engineering?
- Which tools and concepts can I use?
- What are best practices?
- Why is sustainable software hard?

What is the definition of sustainability ...

in the domain of hard- and software



Credit: SDIA

Software uses energy and machines

To make software more sustainable we need to make its consumption visible.

Software uses energy and machines

To make software more sustainable we need to make its consumption visible.

... so what tools do already exist?

Easy starter: Scaphandre - Hubblo

open-source RAPL based command line tool

- Neat feature: Can split by process

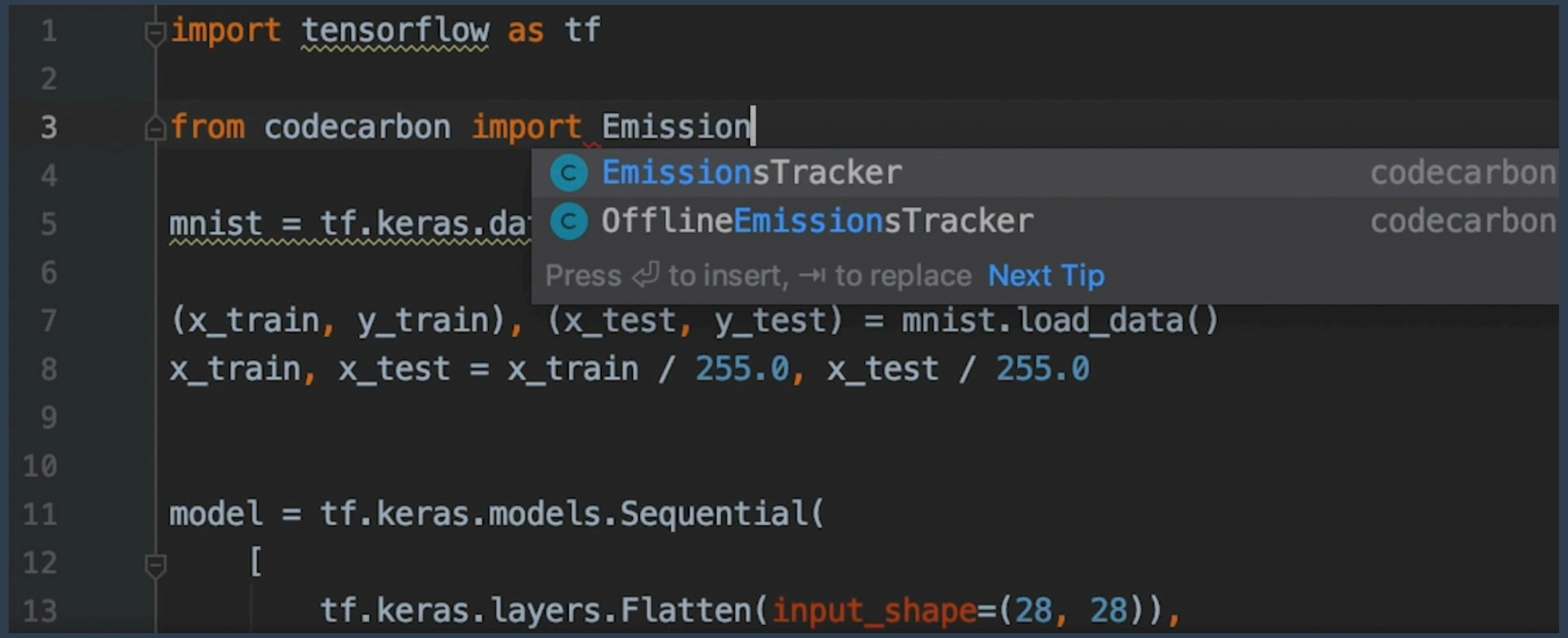
```
Host:   13.1463 W
        package         core            dram            uncore
Socket0 13.1463 W |     10.879847 W     0.748591 W      0.071402 W

Top 5 consumers:
Power           PID     Exe
10.400553 W     16621   "stress"
2.08011 W       16610   "scaphandre"
0.166408 W      2786    "gnome-shell"
0.083204 W      3915    "Xwayland"
0.041602 W      4621    "guake"
----------------------------------------------------------------
```

Ready to use tools

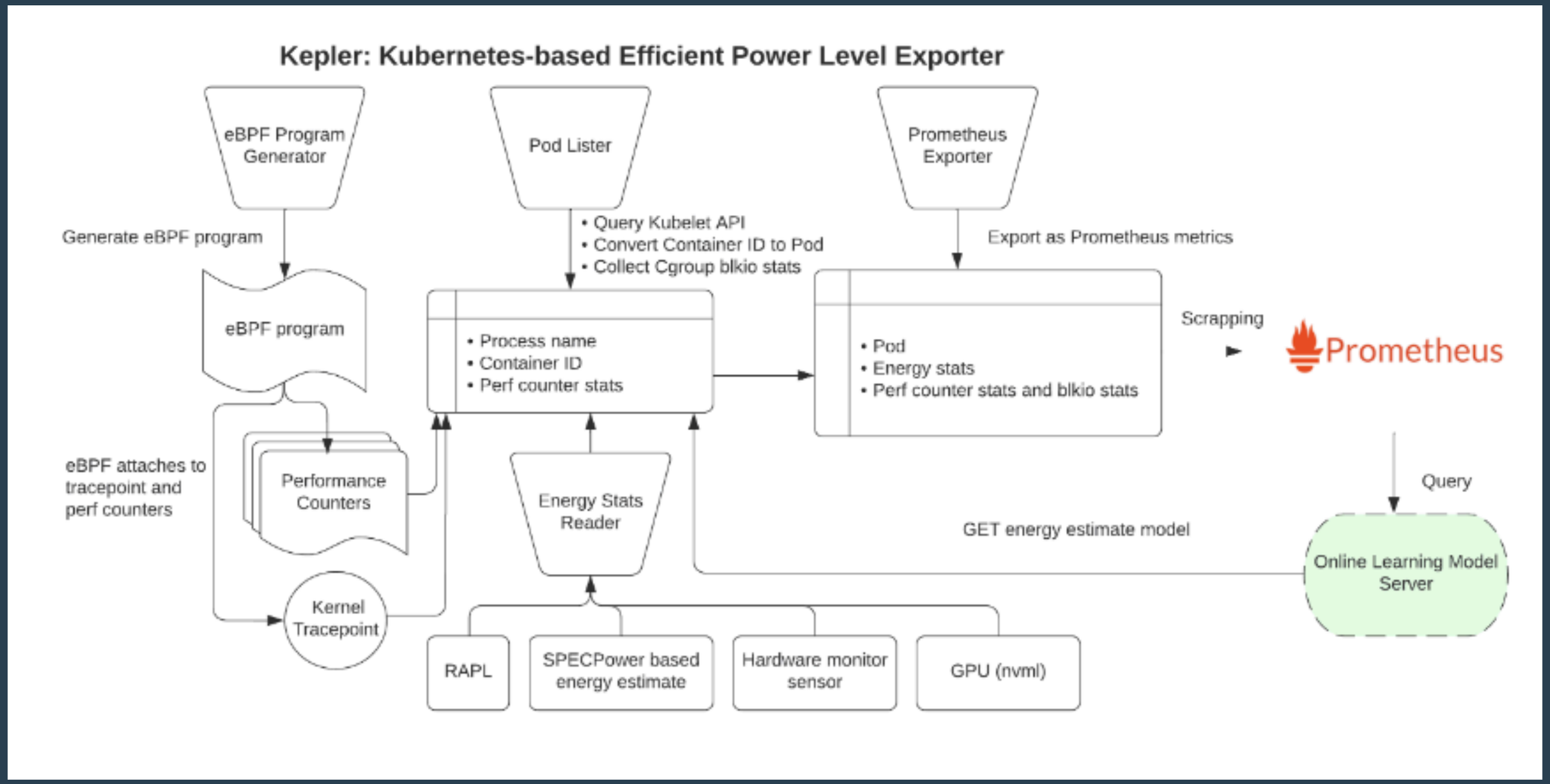
codecarbon.io

- Python
- RAPL-based
- NVIDIA GPU support



Distributed Environments / Clusters

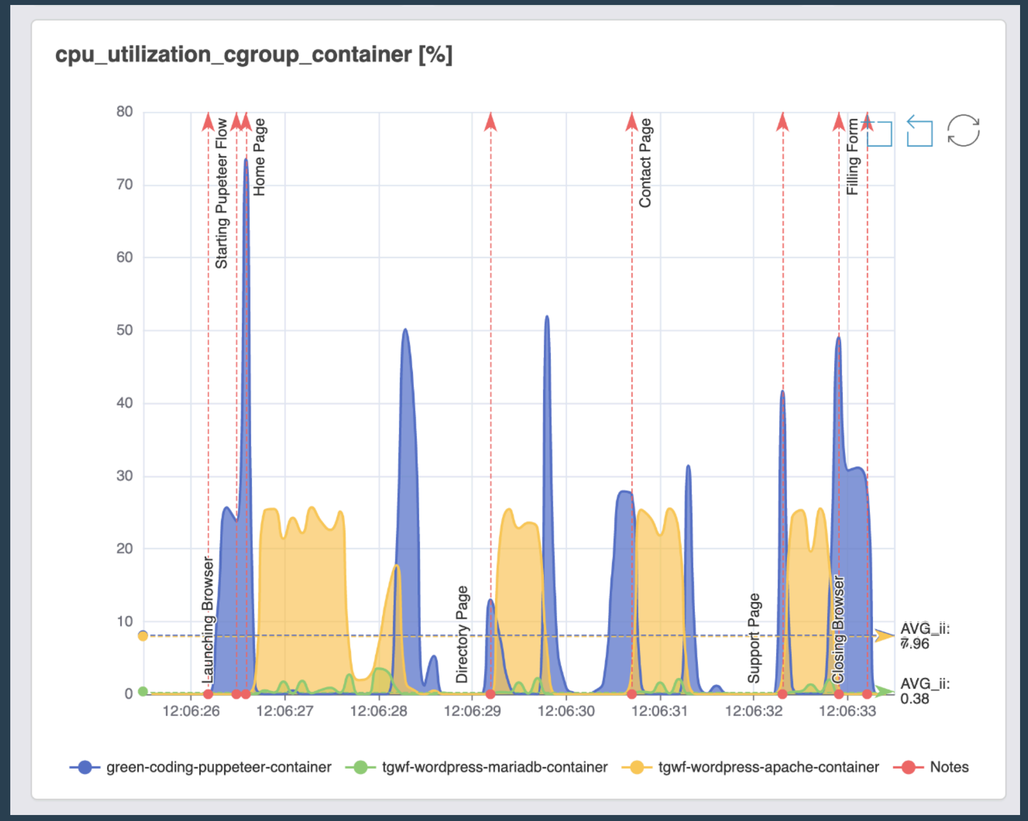
Introducing Kepler

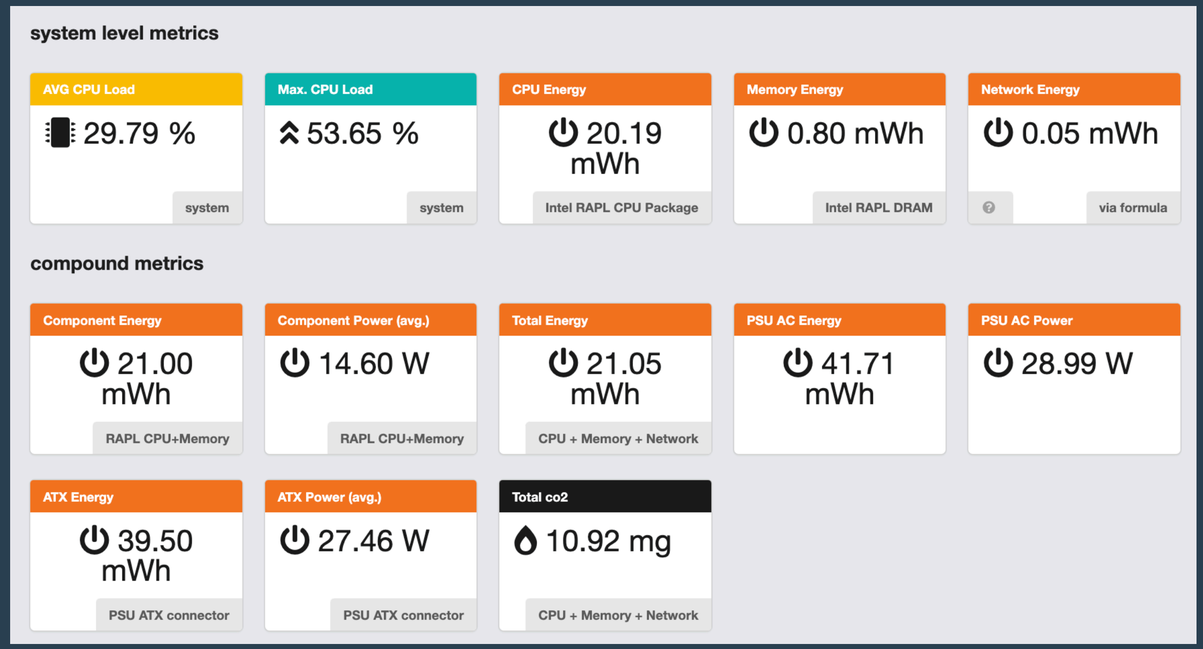


http://sustainable-computing.io

Green Metrics Tool

All-in-One solution for benchmarking, orchestration and transparency

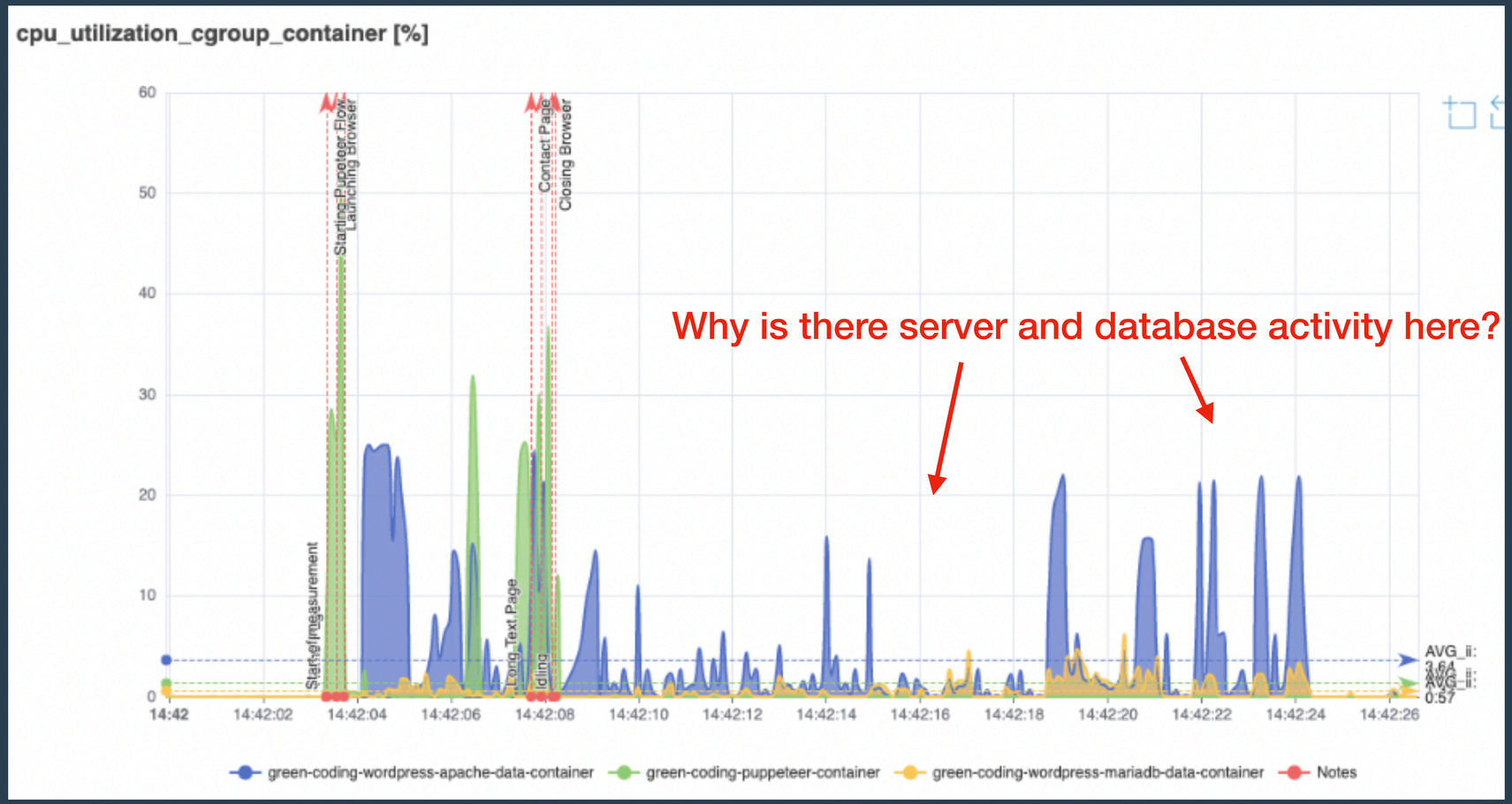




GREEN CODING;

Green Metrics Tool

Container native; For detailed inspection of architecture

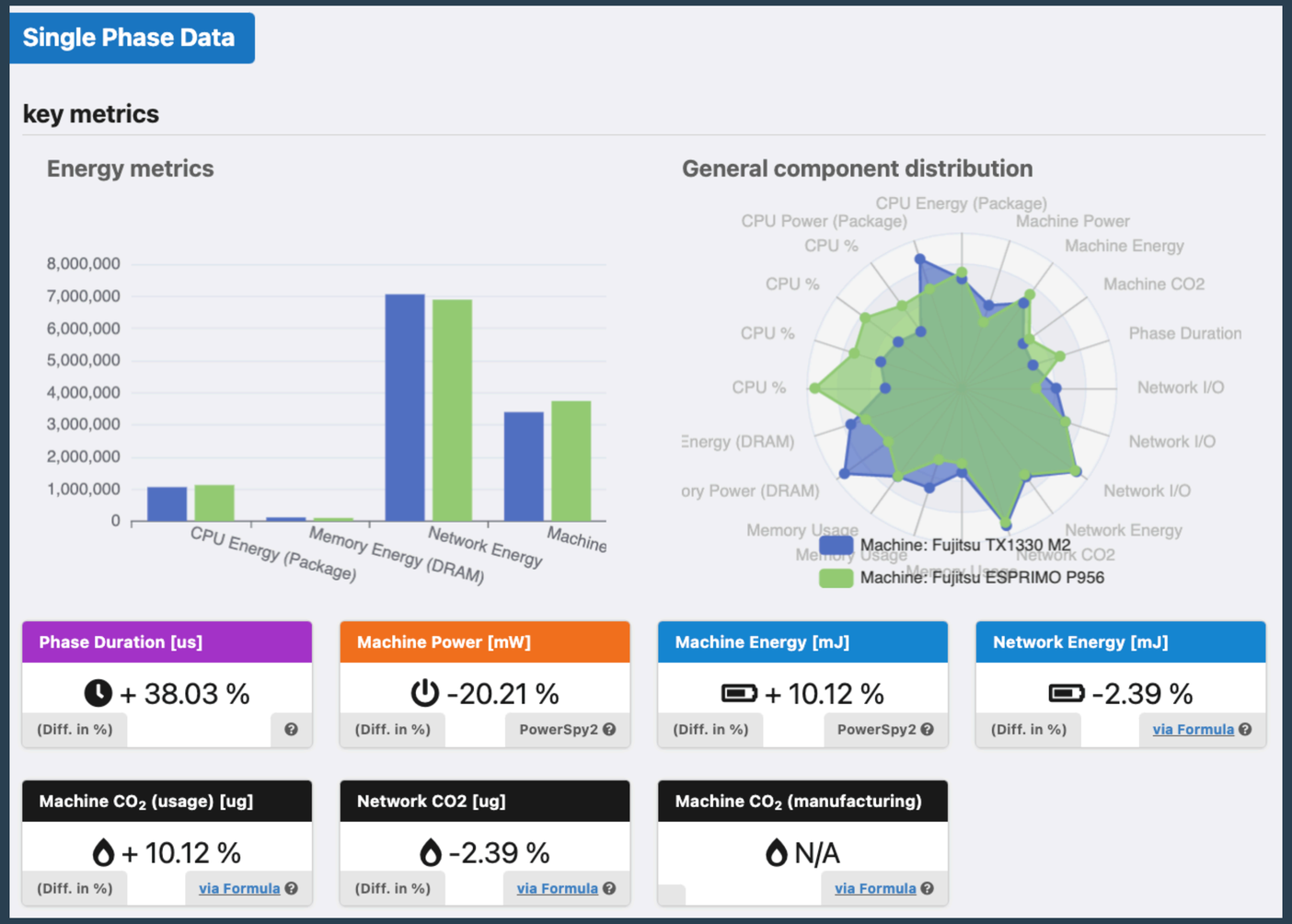


Use-Case mit Wordpress / Django Community

GREEN CODING;

Green Metrics Tool

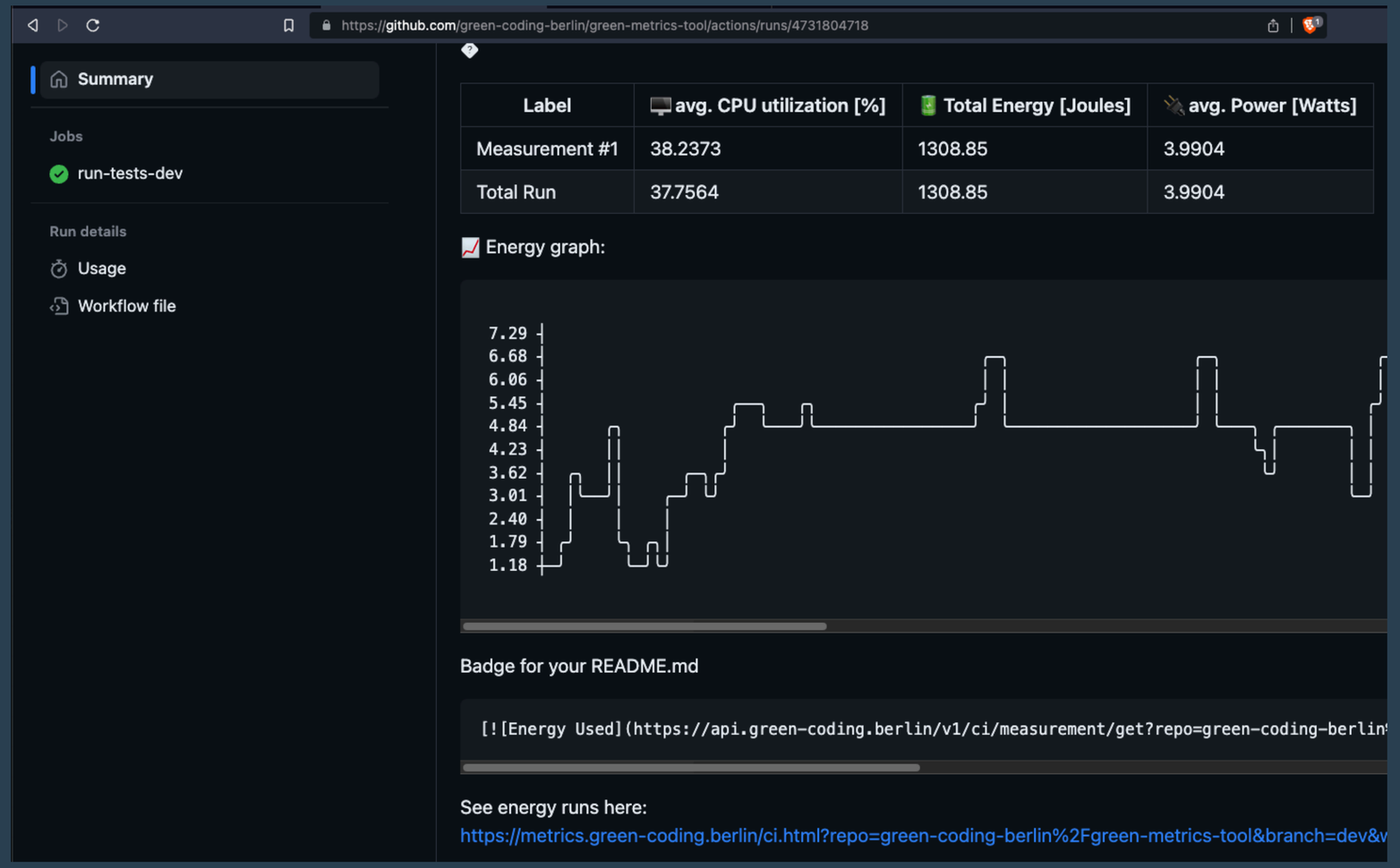
Statistical comparisons and API included



GREEN CODING;

Eco CI Tools - Green Coding Berlin

Energy inside of a CI / CD Pipeline



Eco-CI Tools on Github - Integrated into Actions directly

GREEN CODING;

Weitere Tools ...

for questions regarding a specific tool, please ask in the Q&A!

- powertop
- powermetrics
- Cloud Carbon Footprint
- turbostat
- powerJoularX
- PAPI
- greenframe.io
- ...

Ok, so know I know some tools

Which should I use? And how do I interpret the data?

Are best practices for sustainable software?

Best Practices

According to the Blue Angel for Software

- **Measure your system**
- **Go for reusability**
- **Go for exportability / interoperability**
- **Code must also run on older hardware**
  - To fight device obsolescence (within reason)

Best Practices

According to the Green Software Foundation - A suprising mix ....

- **Tips that are well known and not really "green software"**
  - Cache Static data
  - Use energy efficient AI models 🤡
  - Minify JS / CSS
  - Right-Size VMs
  - ...
- **Tips that are debatable**
  - Run AI models at the edge
  - Use sustainable regions -> Depends on the network transfer. Hard to wage
  - Stateless design -> On its own no gain
- Source: https://patterns.greensoftware.foundation/catalog/cloud/match-utilization-requirements-of-vm/

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. ....

Problems

with current "sustainability" best practices

- **Often they just employ common sense**
  - Like**:** "run less tests"
- **Often generic advice**
  - **"**Use sustainable libraries"
- **Often can be harmful also**
  - Switching machines brings electronic waste
- **Hard to generalize at the moment**
  - Because we have no data
- **Can contain pitfalls**
  Like: "Less runtime is always better" **... no**: TurboBoost, HyperThreading etc. may inverse result

- We plead for a simpler solution ....

Best Practices

But how do we get there

- You just have to measure!
- You have to ask (service providers) for the metrics you don't know.
- And to treat energy and CO2 as a first order metric .... constantly (git-ops)!
- We need a lot of data for general statements
- And we need to include the development and delivery phase of software
  - Isolated benchmarks and claims are helpful for a isolated view, but not for global sustainability

Best Practices

An approach for a framework atm.

1. **Quantify** your system!
   How much am I using for what service (Green Metrics Tool, Scaphandre, Kepler etc.)

2. **Green Energy** (-> Green Web Foundation)
   But not on the moon please!

3. **Idle machines** (Async / Polling / Microservices)
   Only makes sense if you can really turn machines off

4. **Architecture Overhead** (Backups, Logs, Redundancy, Over-Provisioning of Services / Machines)

5. **Language Overhead** (C-Extensions, WebAssembly, Language-Swaps)

6. **Code Changes**
   Even 99% Code runtime reduction is of limited help if the machine then idles

Best Practices

Some general guidance

- **No fancyness until proven helpful**
- **Use less resources -> Yes, common sense!**
  - Services, Machines etc.
- **No off-loading / Backlashes**
  - We are more sustainable, we now use a data provider for that machine!
  - We now use 5 GB less memory (and calculate everything on request :( )
- **General Question: If everybody would do it this way. Would we have enough resources?**
  - Simple chat-bot => ChatGPT
  - Portfolio Website: VPS with Wordpress and database

Thank you! Time for Q&A

Follow green-coding.berlin

- Website / Blog / Newsletter: https://www.green-coding.berlin
- Demo Open Data Repository: https://metrics.green-coding.berlin

- Unsere Projekte: https://www.green-coding.berlin/#projects
- Unsere Case-Studies: https://www.green-coding.berlin/case-studies

- Meetup Gruppe (Berlin): https://www.meetup.com/green-coding

- https://www.linkedin.com/in/arne-tarara / arne@green-coding.berlin

GREEN CODING;

Backup Slides

Demo

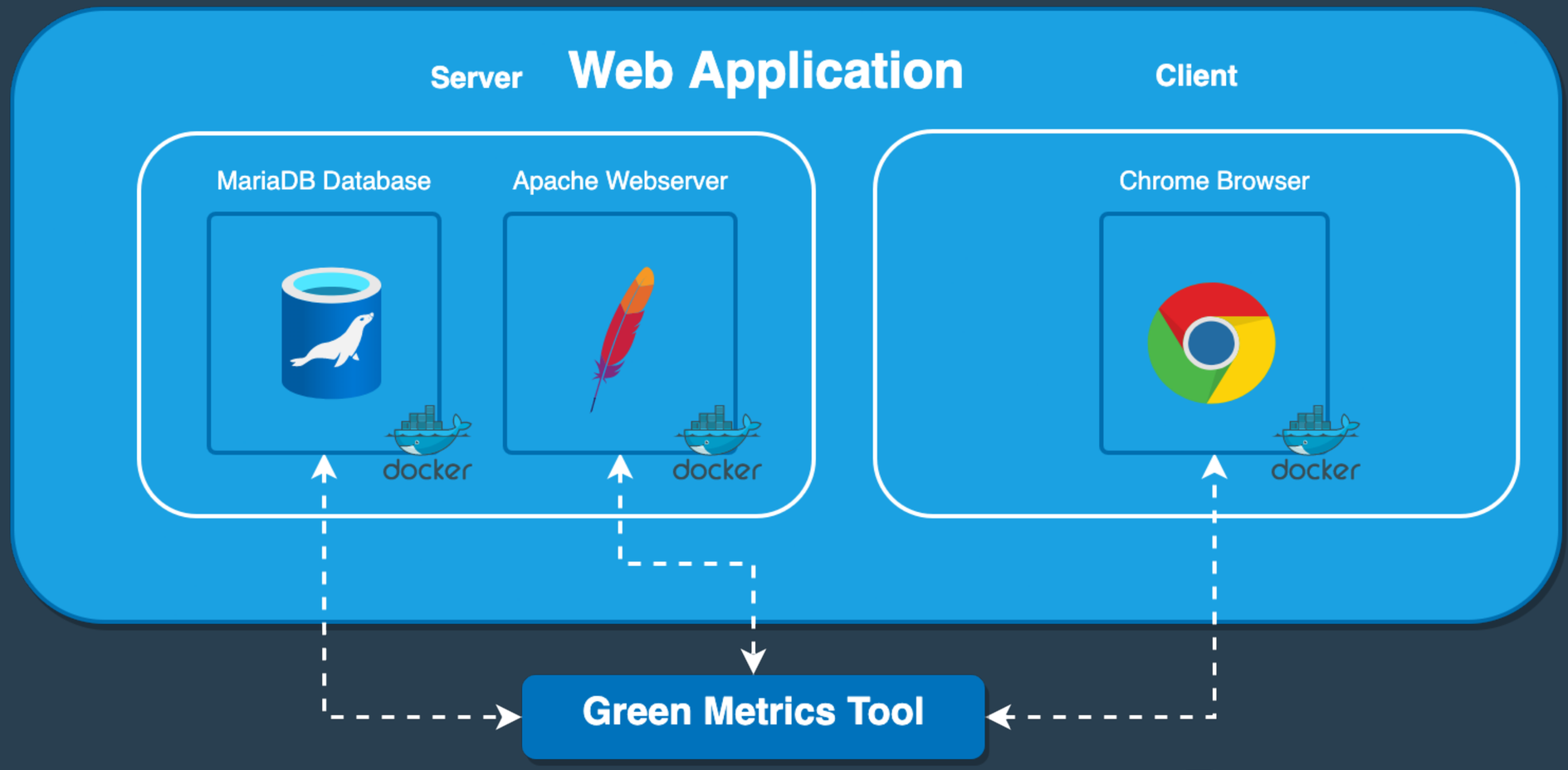
https://preview.green-coding.berlin

-

https://github.com/green-coding-berlin/bakerydemo-gold-benchmark

Green Metrics Tool - Schaubild

Container Aufbau für Client-Server Apps

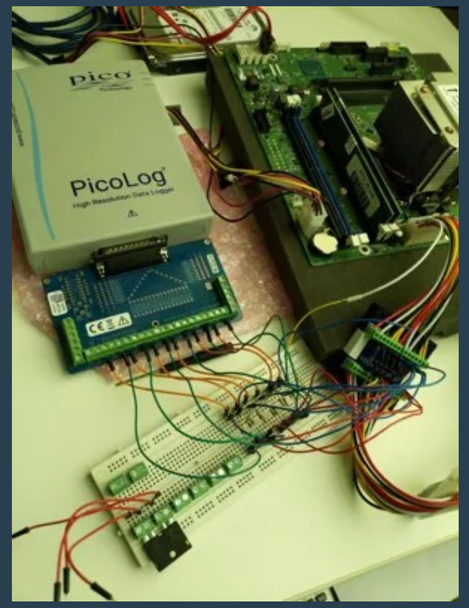


GREEN CODING;

Green Metrics Tool - Hardware Setup

Mess-Sensoren im Green Metrics Tool

- AC / DC Energie
  - IPMI
  - GUDE (Blauer Engel)
  - PowerSpy2 - Consumer Grade Oscilloscope
  - Custom Mainboard Connector for Fujitsu ESPRIMO
- CPU % / CPU-Frequenz
- Temperatur
- CPU / DRAM Energie
  - RAPL
- Netzwerk Datenverkehr / Energie
- Container-Metriken
- ... weitere Reporter als Open-Source Plugin-In möglich



GREEN CODING;

Green Metrics Tool - Cluster Setup

Aktuelle Rechner im Green Metrics Tool Cluster

- Fujitsu ESPRIMO P956 - Blue Angel compatible (Ubuntu)
- Fujitsu TX1330 M2 - Single-Tenant Server (Ubuntu)
- Quanta Leopard - Multi-Tenant Server - SoftAWERE compatible (Ubuntu)
- Intel Mac 13" Q3-2015
- M1 Mac 13" Q1-2022

Green Metrics Tool - Supported Software

Software Kategorien und OS

- **Server / Client Anwendungen**
  - Beispiele Nextcloud und Django CMS
- **CLI Anwendungen**
  - Beispiele curl oder ML Anwendungen (sklearn, pytorch)
- **Desktop Apps**
  - Beispiele: Google Chrome, Firefox etc.
- **OS**
  - macOS
  - Linux (Ubuntu & Fedora)
  - Windows (nur Gesamtleistung an der Steckdose)