

# Green Metrics Tool

Measuring energy use of arbitrary applications & software stacks

 **GREEN CODING;**

# What?

## What is the Green Metrics Tool

- Open Source tool to measure the energy / co2 consumption of software
  - Network I/O, CPU energy, DRAM energy, Hard-Disk energy, AC / DC Total Energy ....
- Measurement according to its "typical use case" (Standard Usage Scenario)
- Visualization on board
- API on board

# Why?

## Why did we build the Green Metrics Tool? Technical parts

- Measuring software energy consumption is still too hard.
  - Goal: Easy as starting a docker container and happen transparently
- Measuring software is complex
  - Best practices and system configuration should be automatically applied

# Why #2

## Why did we build the Green Metrics Tool?

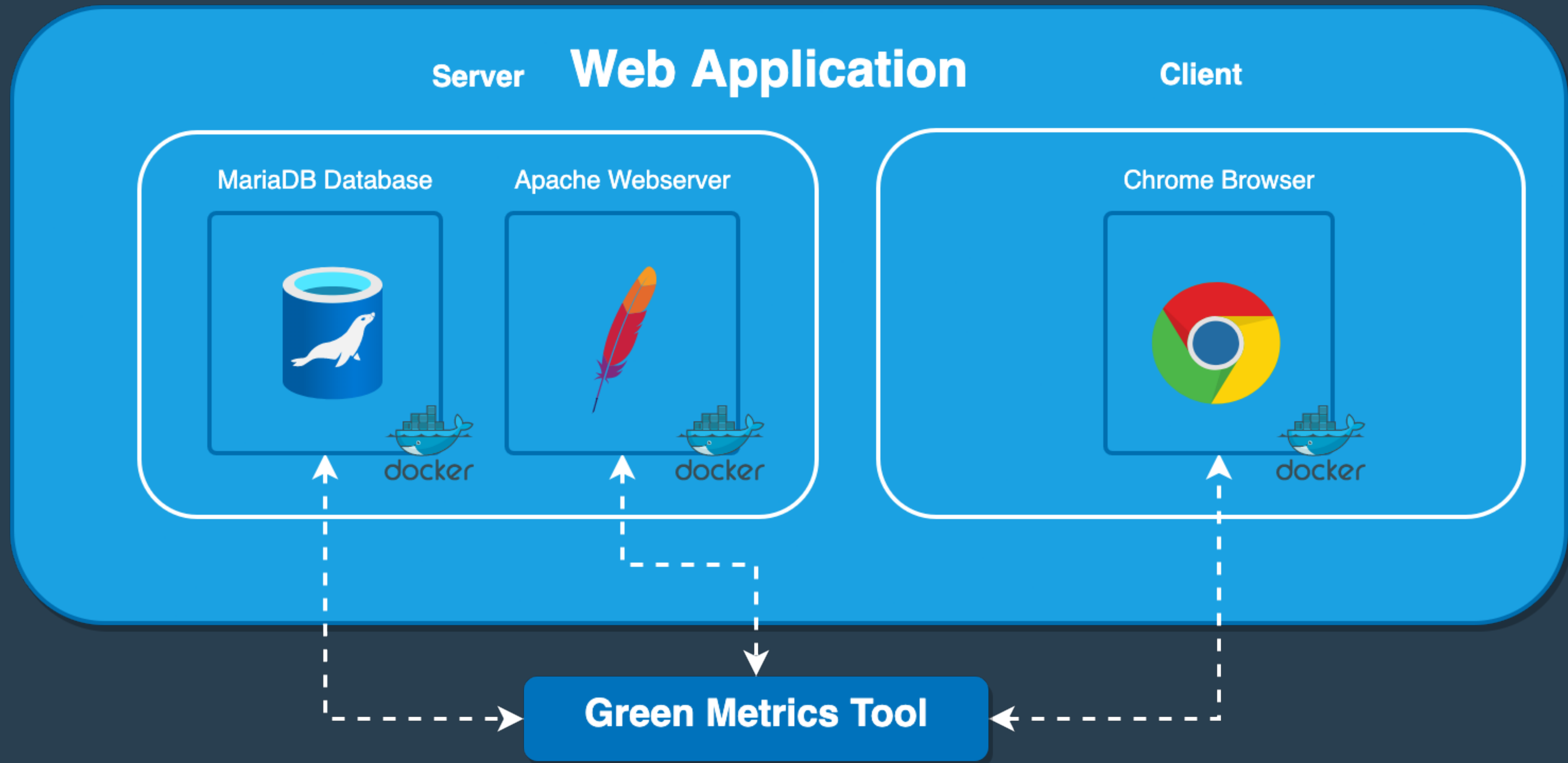
- Comparing software is complex
  - Software must be classified and attributed automatically
- We are missing transparency on the energy consumption of software!
  - Blue Angel has here a good approach (Coming back on this later)

# What is the concept of the tool?

Talking architecture design

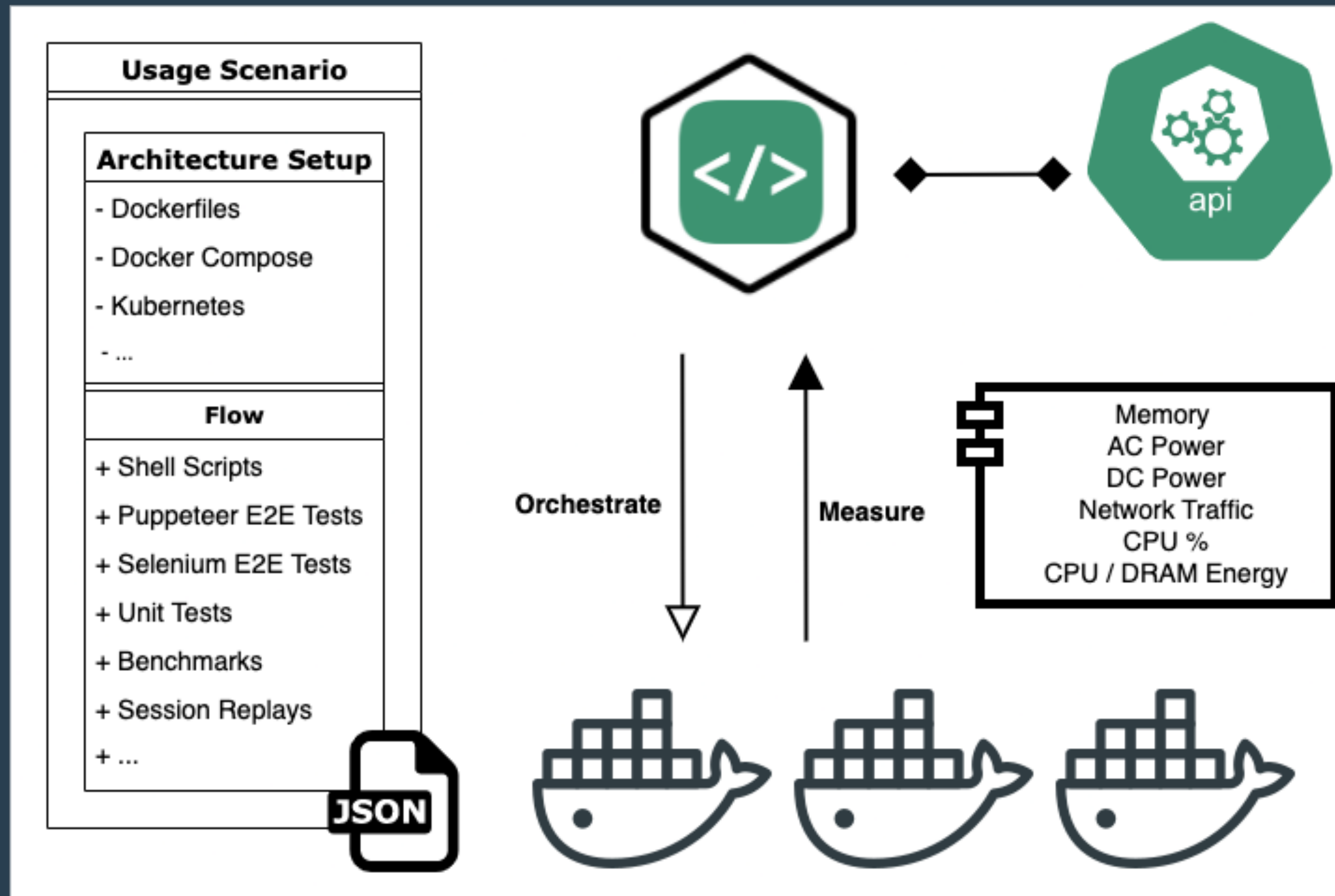
# Concepts of the Green Metrics Tool

Adoption of container approach. Every functionality is a container



# Data Flow in Framework

Ingesting standard infrastructure files. Output as POSIX stream



# How the tool output looks like

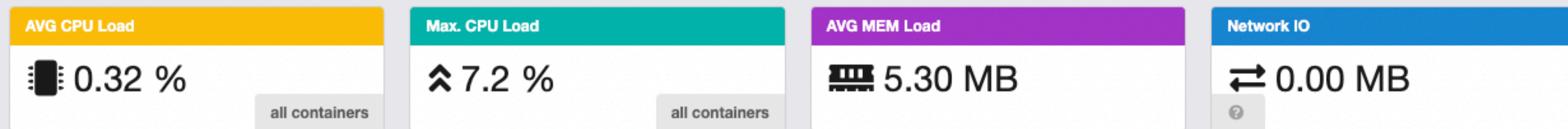
Let's look at some screenshots



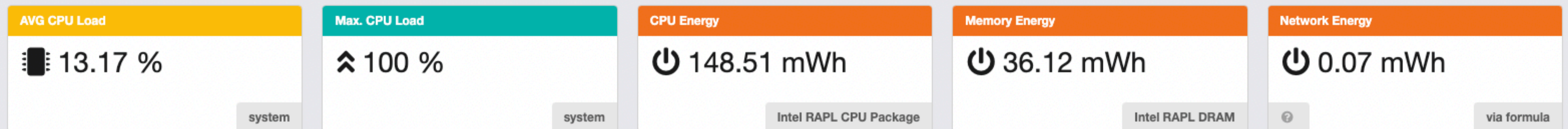
# What the output looks like

## Container, System and compound metrics

### container level metrics



### system level metrics



### compound metrics





# What the output looks like

## Reproducibility through Github repository pinning

### Project Data

General

Measurement

Machine

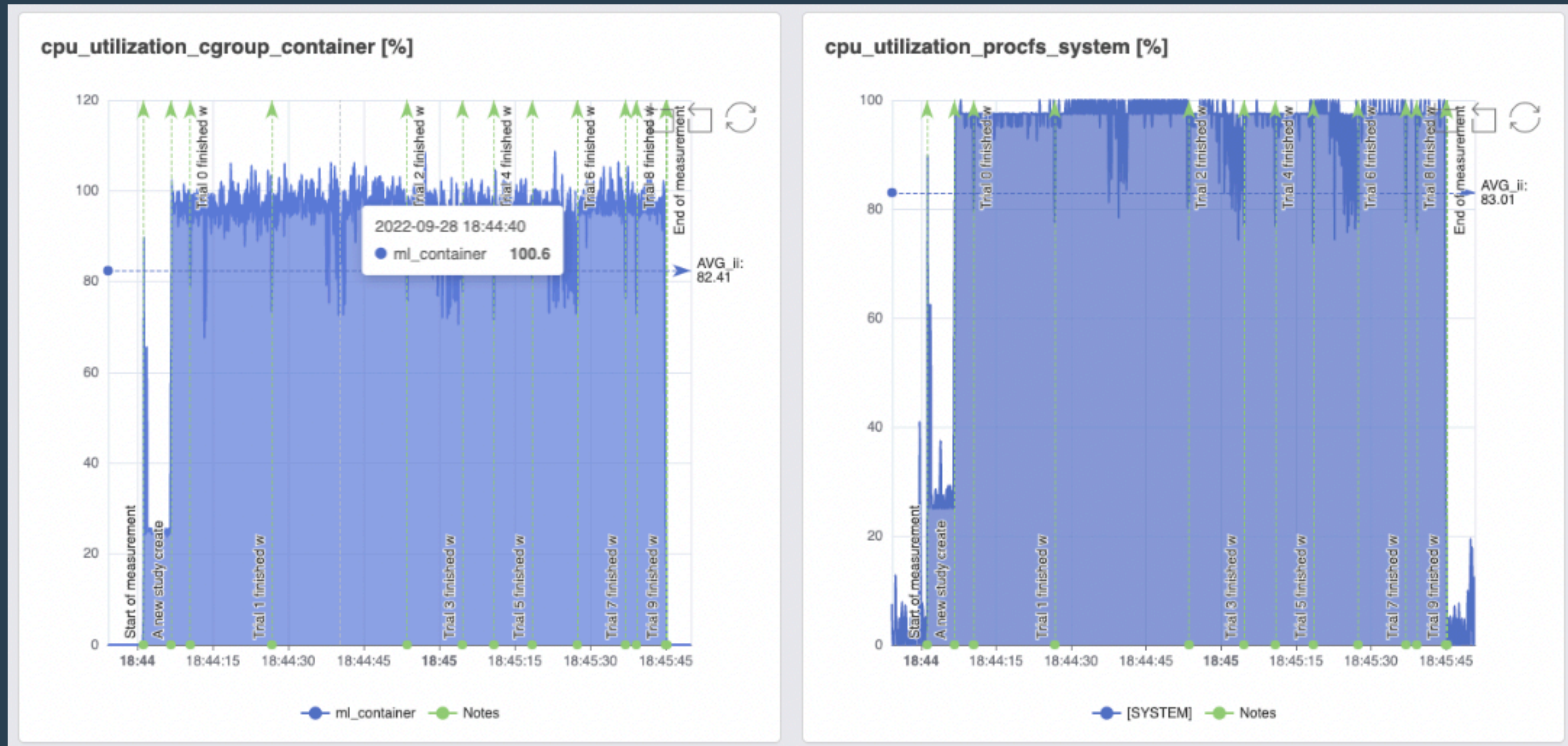
usage\_scenario

<b>id</b>	134dc16a-d6e3-4668-a9e4-b8be7dab6dcb
<b>name</b>	Esprimo-blauer-engel-xdotool-with-AC-DC
<b>uri</b>	<a href="https://github.com/green-coding-berlin/example-applications/tree/main/blauer-engel-okular-xdotool">https://github.com/green-coding-berlin/example-applications/tree/main/blauer-engel-okular-xdotool</a>
<b>categories</b>	null
<b>start_measurement</b>	1660921979409053
<b>end_measurement</b>	1660922193286418
<b>last_run</b>	2022-08-19T15:12:49.609115+00:00
<b>created_at</b>	2022-08-19T15:12:49.466296+00:00
<b>measurement_duration_in_s</b>	213.877365



# What the output looks like

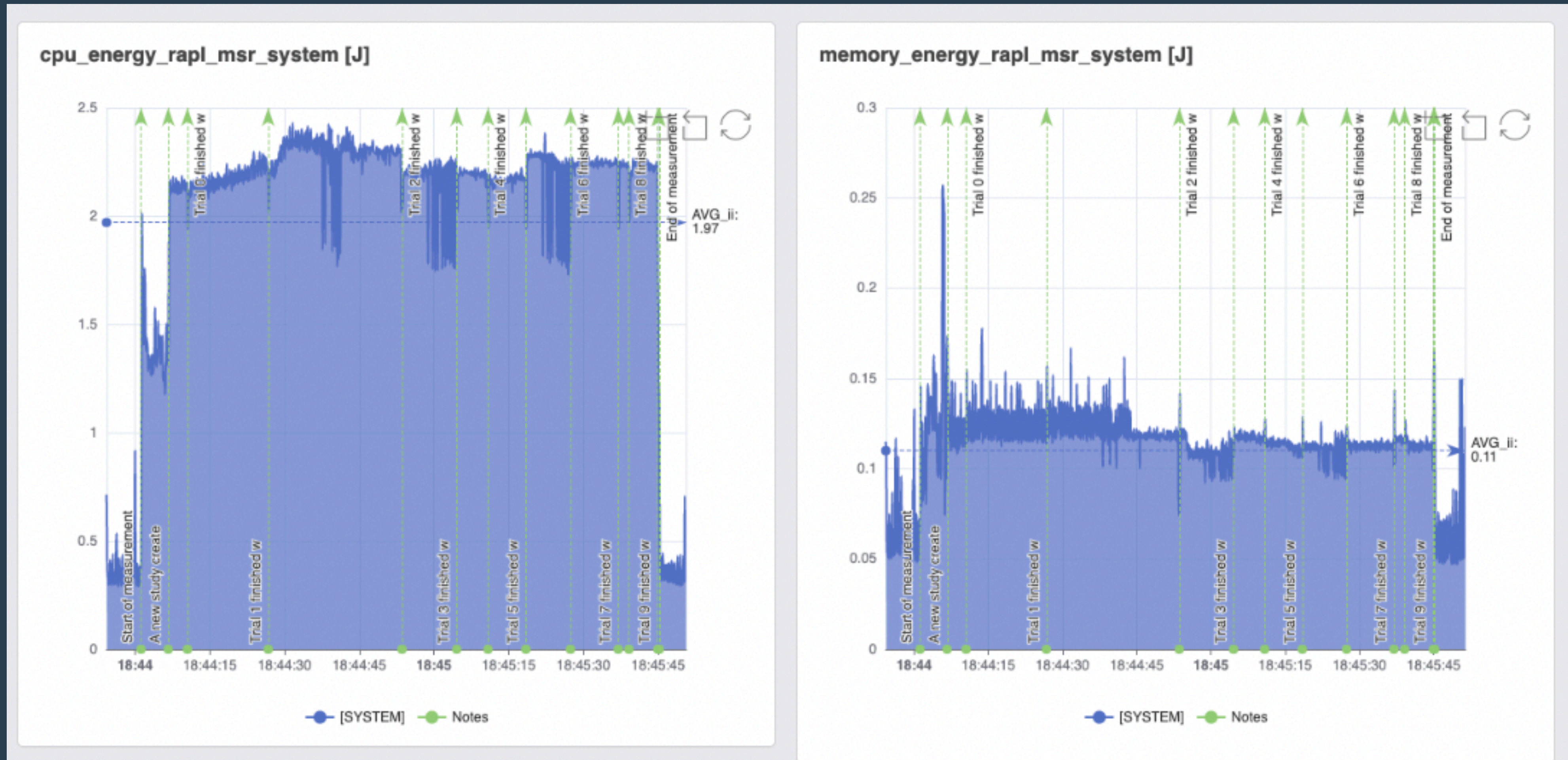
Charts ... obviously ;)





# What the output looks like

Charts ... obviously ;)





# All reporters are modular and stand-alone

Network, DC, AC, Memory, CPU, Energy etc.

## Project Data

General

Measurement

Machine

usage\_scenario

idle-time-end	5
idle-time-start	5
psu.energy.ac.system.provider.PsuEnergyAcSystemProvider	100
cpu.energy.RAPL.MSR.system.provider.CpuEnergyRapIMsrSystemProvider	100
network.io.cgroup.container.provider.NetworkIoCgroupContainerProvider	100
memory.energy.RAPL.MSR.system.provider.MemoryEnergyRapIMsrSystemProvider	100
cpu.utilization.procfs.system.provider.CpuUtilizationProcfsSystemProvider	100
memory.total.cgroup.container.provider.MemoryTotalCgroupContainerProvider	100
cpu.utilization.cgroup.container.provider.CpuUtilizationCgroupContainerProvider	100
flow-process-runtime	600

# What kind of applications are supported?

## in the Green Metrics Tool

# Deeper dive into possible application types

## What applications can the tool consume

- Desktop applications like Firefox that allow containerization
- Command Line applications
  - Console tools
  - Machine-Learning Models
  - ...
- Web Applications / Web Stacks

**What can you do with it?**  
in terms of analysis ...



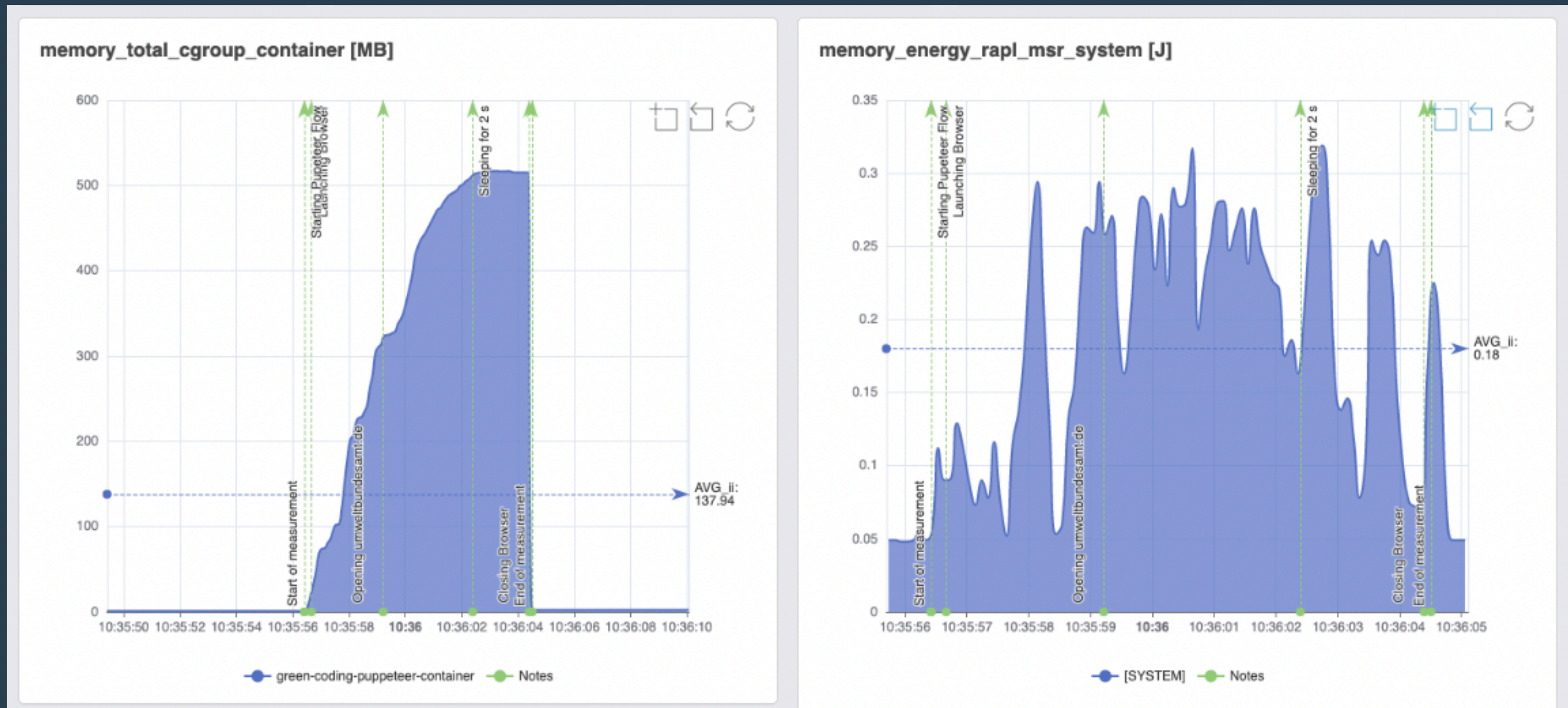
# What analysis can you do with the tool?

## Different code, different scopes

- For **machine learning** apps or long running codes it is more important to understand energy budgets, anomalies and total runtime, cause usually all cores are utilized
- For **web applications** it is important to identify idling systems and non-scaling systems
- For **algorithms** it is important to understand how your code behaves when the system is differently configured (TurboBoost, C-States etc.)

# Identify Memory Anomalies in terms of energy

Memory energy does not equal "memory usage"





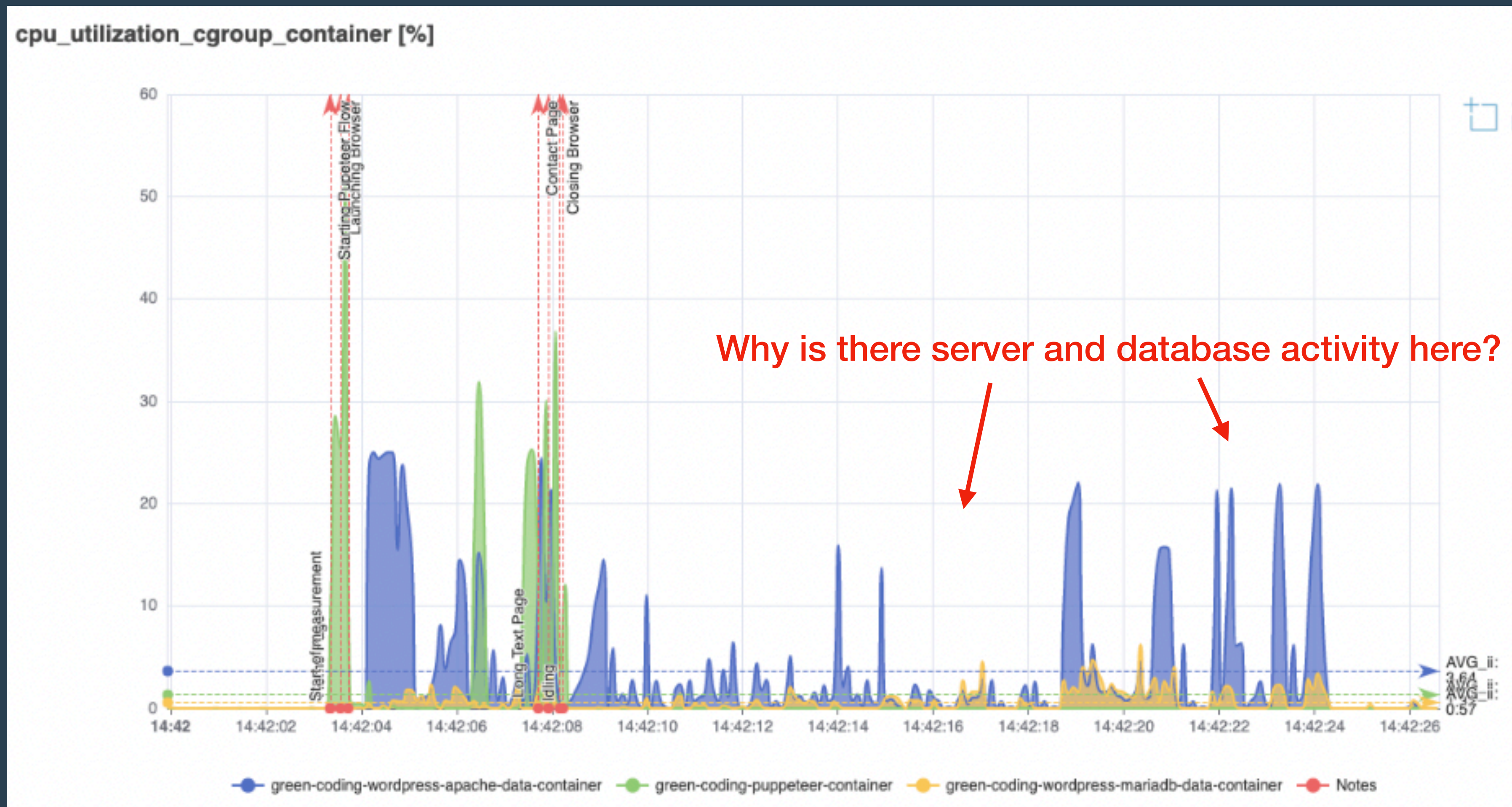
# Energy anomalies in Machine Learning

## Energy anomalies through TurboBoost and Overheating





# Idle time optimizations in web applications



# What is our vision for the tool?

## Roadmap

# What is our vision for the tool?

## Features

- Android applications; Windows support
- Distributed applications (Kubernetes)
- Inline reporting in CI/CD Pipelines (Awareness of testing costs)
- Energy splitting on process / instruction level
- Provide recommendations for energy optimizations

# What is our vision for the tool?

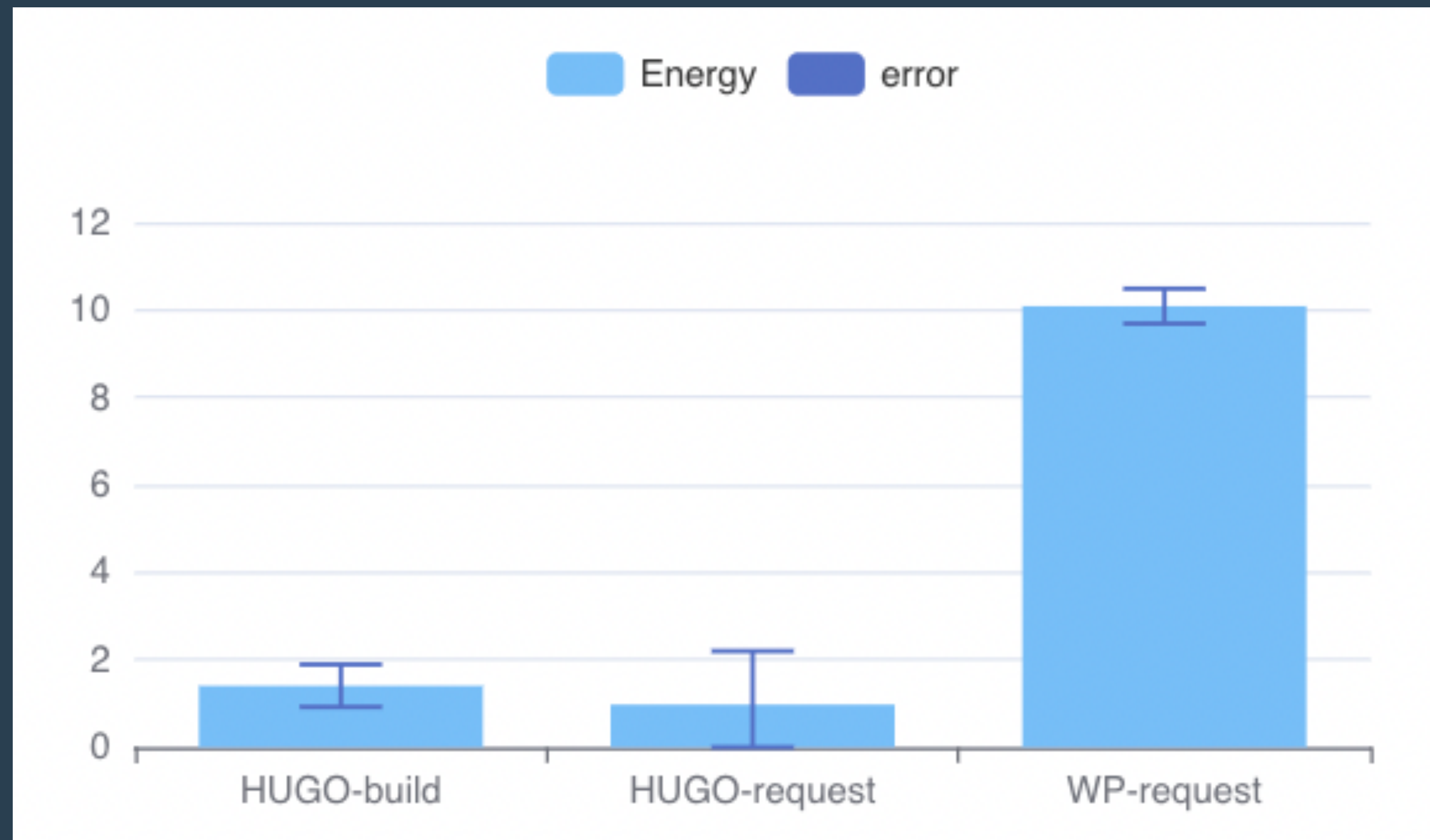
## Providing developers with answers

- Energy efficiency of Wordpress vs. static site?
- Is podman more effective than docker for building?
- Is Flask better than FastAPI than Django for our workflow?
- Hardest goal to achieve: recommendations
- When to make energy optimizations
- What kind of saving do I want to have? 1 g of co2? 1 ton of co2? What is my budget?
- When should I switch architectures?



# We want to provide insights

to developers and user. With case studies and comparisons



Comparing a Wordpress implementation with a Static-Site implementation (actual data)



# What is our vision for the tool?

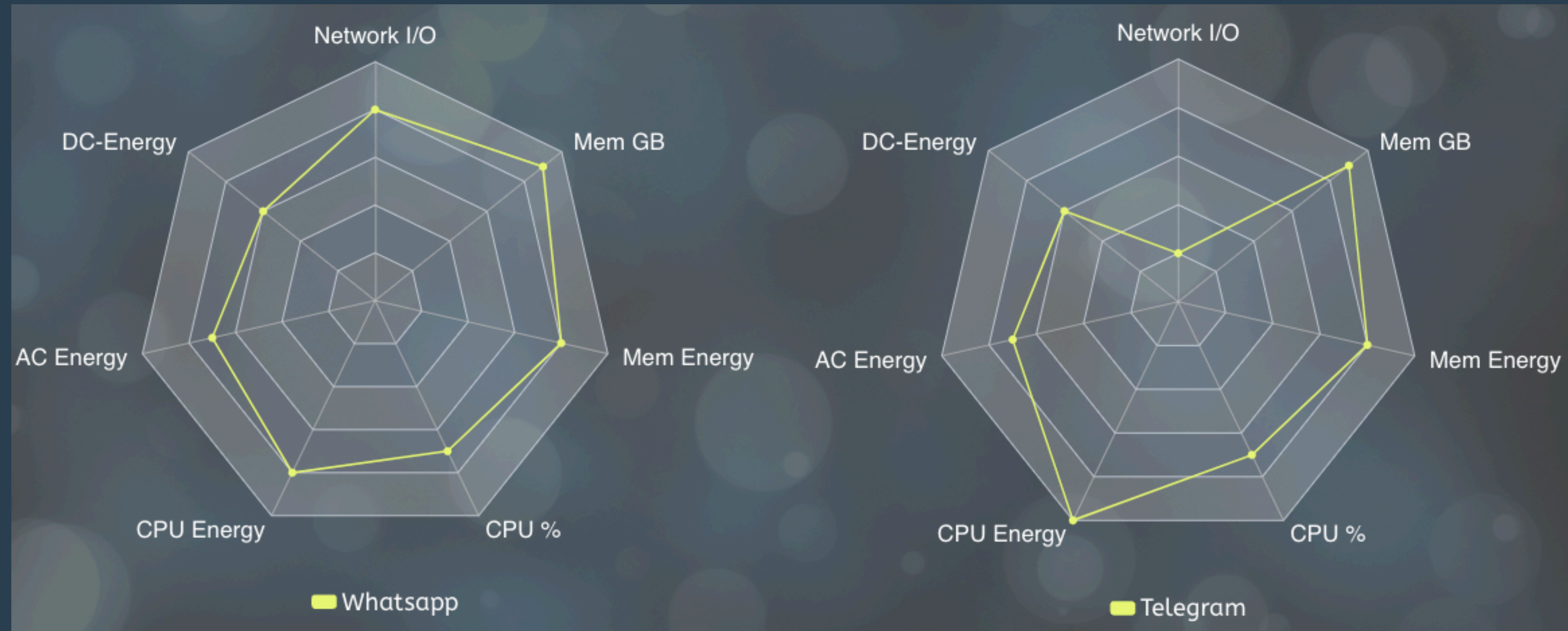
## Providing users with answers

- Comparing software given boundary conditions / usage scenario
- Network level anomalies with Reverse Proxying request
  - Find out if application is calling specific URLs and in what frequency
- Issuing certificates like Blue Angel after min. sustainability criteria are met
-

# What is our vision for the tool?

Providing users with answers to make actionable insights

No actual data! Concept picture!



Comparing Whatsapp vs. Telegram for sending 10 messages and opening app 30 times

# Want deeper dives and more details?

## Follow Green-Coding.org

- Check out our website and blog & newsletter: <https://www.green-coding.org>
- Meetup group: <https://www.meetup.com/green-coding>
- Demo Open Data Repository: <https://metrics.green-coding.org>
- Our tool: <https://github.com/green-coding-berlin/green-metrics-tool>
- <https://www.linkedin.com/in/arne-tarara> / [arne@green-coding.org](mailto:arne@green-coding.org)
- If you wanna present your green software case, please hit us up!



# Another Tool?

Aren't there tools to do this already?

- Scaphandre
- Greenframe.io
- perf\_events
- PAPI
- AMD uProf
- Intel Power Gadget
- Academic research ... lots of! But never with actual raw data ...
- .... more?