Green Software Metrics

Andreas Brunnert

Professor @ Munich University of Applied Sciences HM

Why?

- No clear definition of what makes Software Green
- The Green Software Measurement Model [1] is by now the most comprehensive overview of research and practice in this area
  - Mostly focus on complete runtimes, e.g., VMs or full containers
  - Not applicable for distributed systems
- Furthermore, the Software Carbon Intensity (SCI) [2] specification by the Green Software Foundation is by now an ISO standard ISO/IEC 21031:2024 [3] but is limited to one functional unit
- **Proposal:** leverage resource demand measurements on transaction level as basis for evaluation

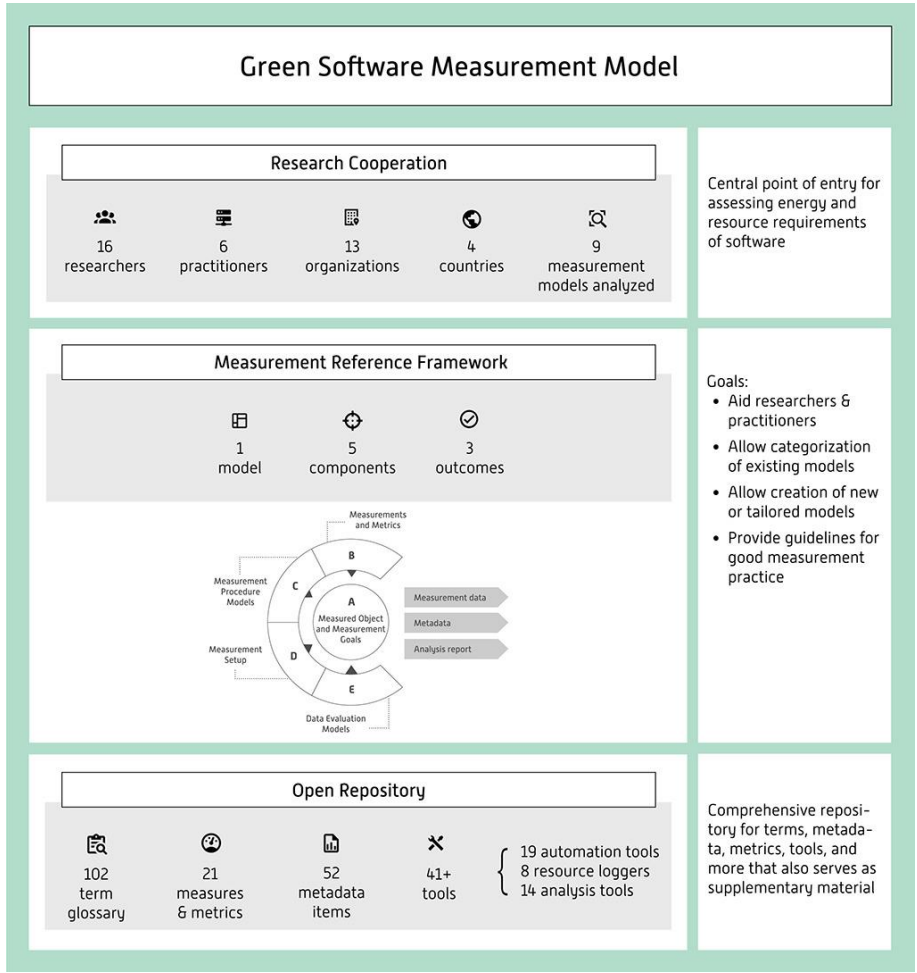
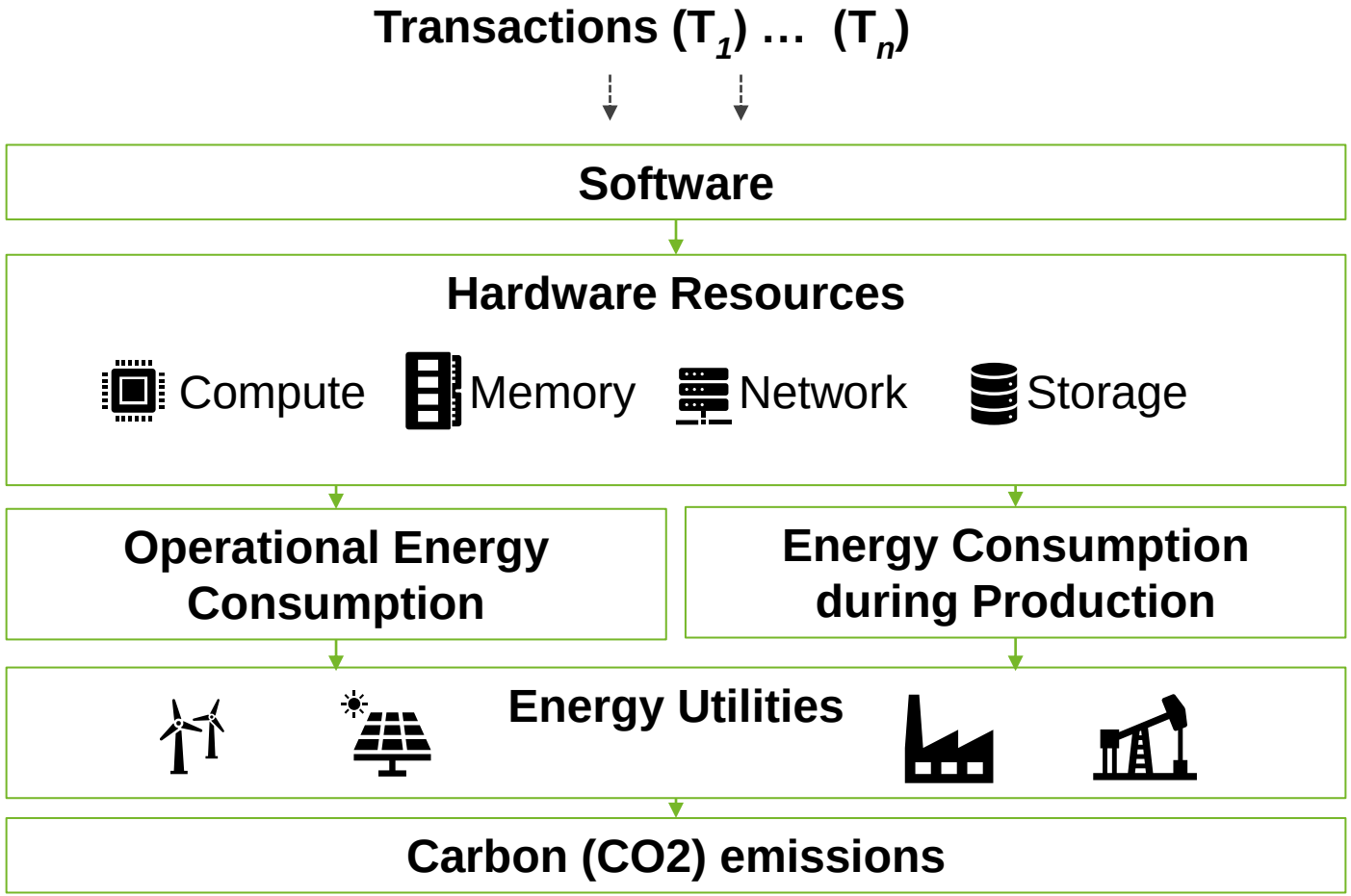
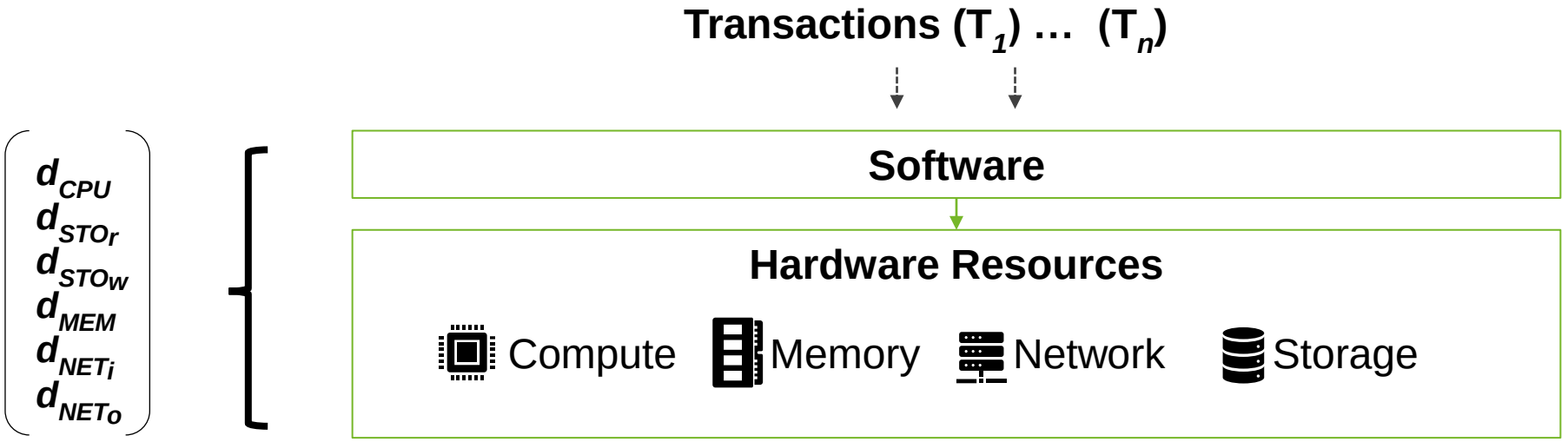


Figure Source: [1]

What?


Transactions (T_1) … (T_n)
Software
Hardware Resources
Compute
Memory
Network
Storage
Operational Energy Consumption
Energy Consumption during Production
Energy Utilities
Carbon (CO2) emissions

What?

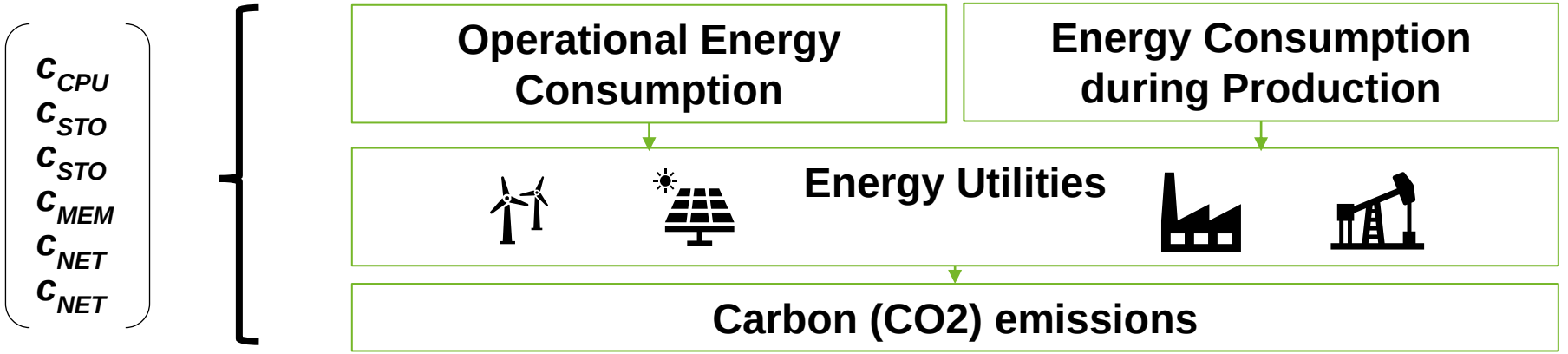


- In [4] we presented an approach to measure the resource demand for each transaction of a distributed system represented in so called resource profiles which consists of resource demand vectors for each component of the system

d = Resource Demand
c = carbon emissions
STO = Storage
r = read
w = write
MEM = Memory
NET = Network
i = Input
o = Output

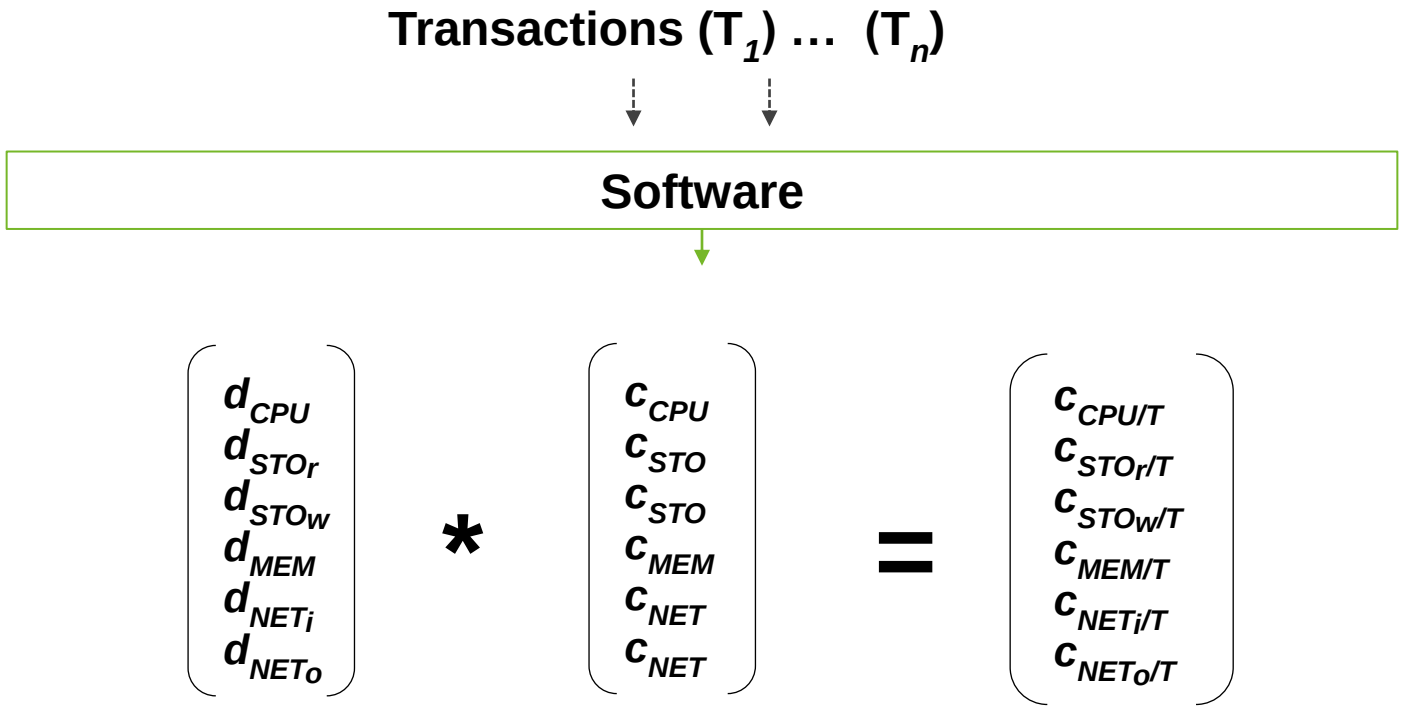
What?

- Climatiq [5] offers a service to collect the carbon emissions for each resource (CPU, memory, storage) of the major cloud providers in a given region
  - Alternative sources are also available but Climatiq is the most easy to use



d = Resource Demand
c = carbon emissions
STO = Storage
r = read
w = write
MEM = Memory
NET = Network
i = Input
o = Output

What?



Carbon emissions for T = c_T = $c_{CPU/T} + c_{STO_r/T} + c_{STO_W/T} + c_{MEM/T} + c_{NET_i/T} + c_{NET_O/T}$

d = Resource Demand
c = carbon emissions
STO = Storage
r = read
w = write
MEM = Memory
NET = Network
i = Input
o = Output

What?

- Example for an example transaction (T)
  - Resource Demand and Carbon Emission Data collected for one minute (60s)
  - The API call was executed 100 times in a minute (60s)
  - Application did not read or write from Storage

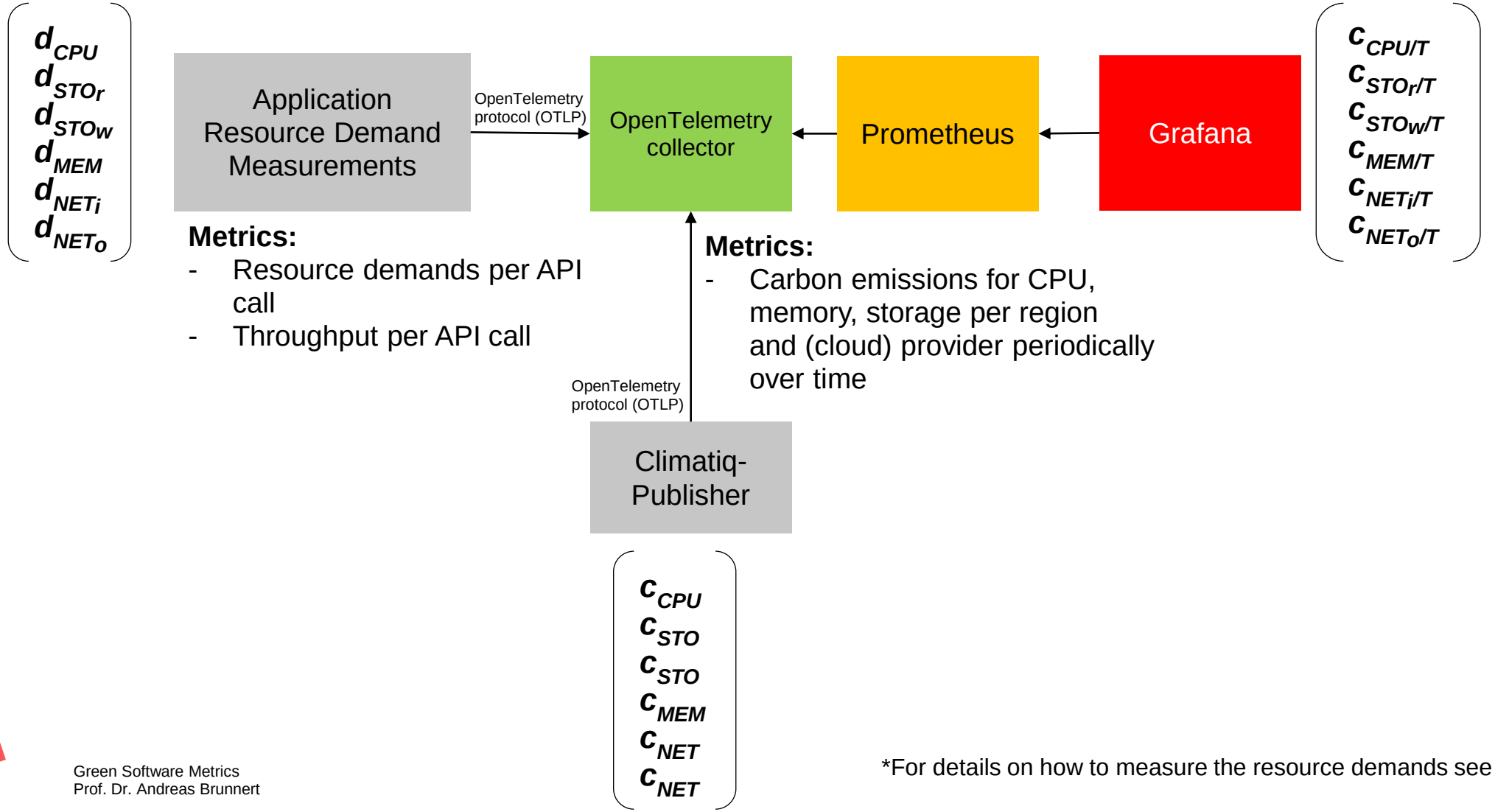
$$\begin{pmatrix} d_{CPU} = 75s \\ d_{STO_r} = 0GB \\ d_{STO_w} = 0GB \\ d_{MEM} = 2GB \\ d_{NET_i} = 0{,}3GB \\ d_{NET_o} = 0{,}1GB \end{pmatrix} * \begin{pmatrix} c_{CPU} = 0{,}2mgCO2e/s \\ c_{STO} = 0{,}0001mgCO2e/GB \\ c_{STO} = 0{,}0001mgCO2e/GB \\ c_{MEM} = 0{,}04mgCO2e/GB \\ c_{NET} = 0{,}1mgCO2e/GB \\ c_{NET} = 0{,}1mgCO2e/GB \end{pmatrix} = \begin{pmatrix} c_{CPU} = 15mgCO2e \\ c_{STO_r} = 0 \\ c_{STO_w} = 0 \\ c_{MEM} = 0{,}08mgCO2e \\ c_{NET_i} = 0{,}03mgCO2e \\ c_{NET_o} = 0{,}01mgCO2e \end{pmatrix} / \begin{pmatrix} 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \end{pmatrix} = \begin{pmatrix} c_{CPU/T} = 0{,}15mgCO2e \\ c_{STO_r/T} = 0 \\ c_{STO_w/T} = 0 \\ c_{MEM/T} = 0{,}0008mgCO2e \\ c_{NET_i/T} = 0{,}0003mgCO2e \\ c_{NET_o/T} = 0{,}0001mgCO2e \end{pmatrix}$$

$$\text{Carbon emissions for } T = c_T = c_{CPU/T} + c_{STO_r/T} + c_{STO_w/T} + c_{MEM/T} + c_{NET_i/T} + c_{NET_o/T}$$

$$c_T = 0{,}1512mgCO2e$$

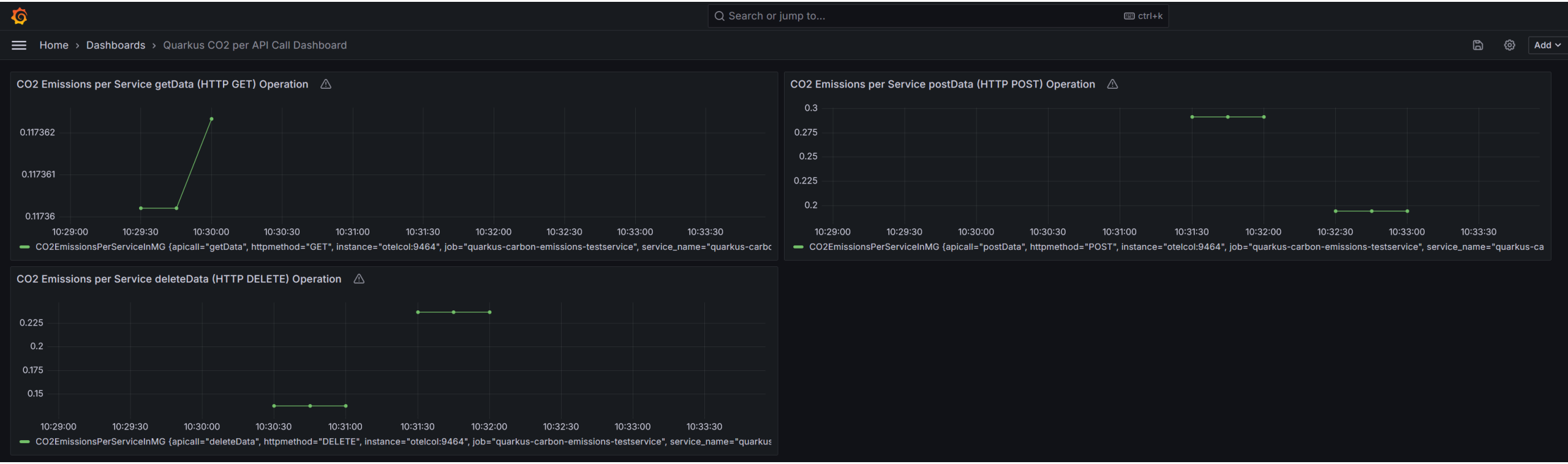
d = Resource Demand
c = carbon emissions
STO = Storage
r = read
w = write
MEM = Memory
NET = Network
i = Input
o = Output

How?



*For details on how to measure the resource demands see [4]

How?



Source: https://github.com/RETIT/quarkus-carbon-emissions

How?

Example Quarkus-based Microservice that emits resource demand and emission data including a Grafana Dashboard for single API Calls

https://github.com/RETIT/quarkus-carbon-emissions

Example Spring-based Microservice that emits resource demand and emission data including a Grafana Dashboard for single API Calls

https://github.com/RETIT/spring-carbon-emissions

Q&A

Questions?

References

[1] A. Guldner, R. Bender, C. Calero, G. S. Fernando, M. Funke, J. Gröger, L. M. Hilty, J. Hörnschemeyer, G.-D. Hoffmann, D. Junger, T. Kennes, S. Kreten, P. Lago, F. Mai, I. Malavolta, J. Murach, K. Obergöker, B. Schmidt, A. Tarara, J. P. De Veaugh-Geiss, S. Weber, M. Westing, V. Wohlgemuth, and S. Naumann. Development and evaluation of a reference measurement model for assessing the resource and energy efficiency of software products and components—green software measurement model (gsmm). Future Generation Computer Systems, 155:402–418, 2024.

[2] Software Carbon Intensity (SCI) Specification, https://sci.greensoftware.foundation/

[3] ISO/IEC 21031:2024, https://www.iso.org/standard/86612.html

[4] A. Brunnert and H. Krcmar. Continuous performance evaluation and capacity planning using resource profiles for enterprise applications. Journal of Systems and Software, 123:239–262, 2017.

[5] Climatiq, https://www.climatiq.io/