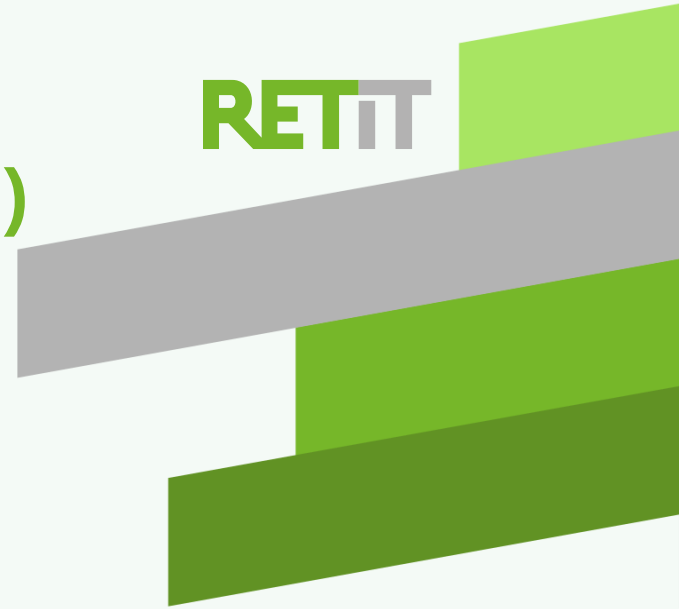
Automatically measure the Software Carbon Intensity (SCI) of your Cloud Software with OpenTelemetry and Cloud Carbon Footprint



Andreas Brunnert

Professor @ University of Applied Sciences Munich HM
Founder @ RETIT GmbH

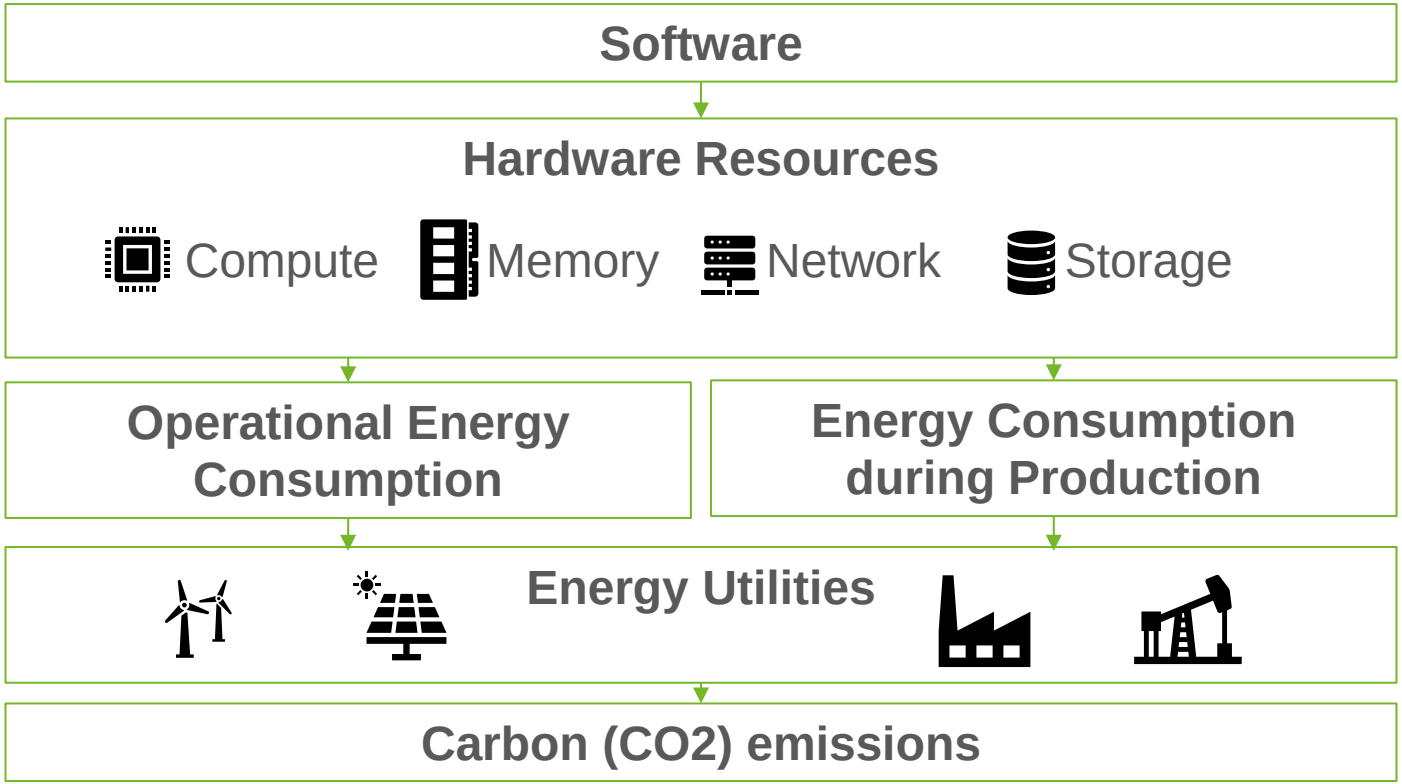
Green Software Foundation (GSF) Global Summit 2024

Agenda

- Automatically measure the

1. **Software Carbon Intensity (SCI)**
   - **…** of your cloud software with
2. **OpenTelemetry**
   - … and
3. **Cloud Carbon Footprint**

Software Carbon Emissions

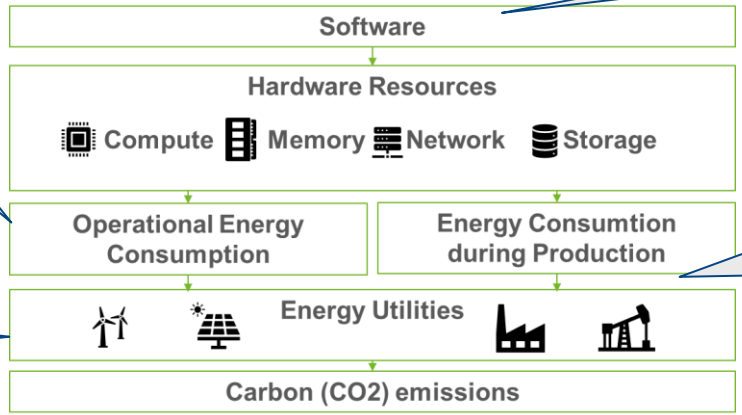


Software Carbon Intensity (SCI) Specification

E = This is the energy consumed by a software system for a functional unit of work.

SCI = ((E * I) + M) per R

R = Functional Unit; this is how software scales, for example per user or per device



I = Carbon emitted per kWh of energy, gCO2eq/kWh

M = Embodied carbon of the hardware that the software is running on

- See https://www.iso.org/standard/86612.html and https://sci.greensoftware.foundation/ for more details

Software Carbon Intensity (SCI) Specification

- ***SCI = ((E * I) + M) per R***

E = This is the energy consumed by a software system for a functional unit of work.

R = Functional Unit; this is how software scales, for example per user or per device

- *E needs to be measured **for a specific** software*
- *R needs to be measured **for a specific** software*
- *I needs to be collected from sources **outside** the software*
- *M needs to be collected from sources **outside** the software*

I = Carbon emitted per kWh of energy, gCO2eq/kWh

M = Embodied carbon of the hardware that the software is running on

Software Carbon Intensity (SCI) Specification

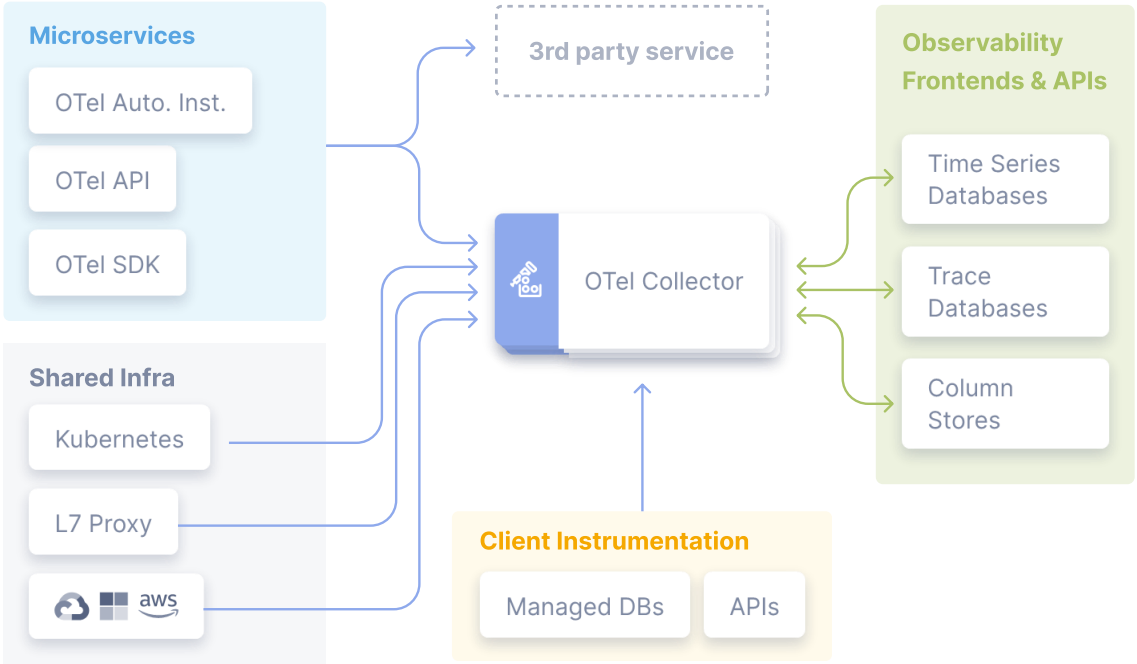
- ***SCI = ((E * I) + M) per R***
- *E needs to be measured **for a specific** software*
- *R needs to be measured **for a specific** software*

https://opentelemetry.io

- *I needs to be collected from sources **outside** the software*
- *M needs to be collected from sources **outside** the software*

https://www.cloudcarbonfootprint.org

OpenTelemetry



https://opentelemetry.io/docs/

OpenTelemetry

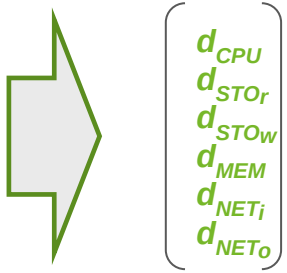
- OpenTelemetry automatically captures the response time and call count (***R*** of the SCI formular) of a particular transaction
- Our OpenTelemetry extension that also collects the resource demands for Java-based software systems (https://github.com/RETIT/opentelemetry-javaagent-extension)

```
YourService {
    yourAPI {
        d_before = measureResourceDemandBefore()

        doBusinessWork(…)

        d_after = measureResourceDemandAfter()

        d = d_after - d_before
    }
}
```



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

OpenTelemetry + Cloud Carbon Footprint Data

- To calculate the ***E*** value of the SCI formula we need to know how much power (P) is required to provide the resource demanded (d) by the software and for how much time (t) has passed
- In order to do this, the cloud carbon footprint website published a methodology to calculate the energy for cloud-based environments: https://www.cloudcarbonfootprint.org/docs/methodology
- Following this methodology, we can transform the resource demand to energy consumption values as shown in *Brunnert / Gutzy (2024): Extending the OpenTelemetry Java Auto-Instrumentation Agent to Publish Green Software Metrics (preprint available here)*

E for CPU

$$P_{CPU} = P_{CPUmin} + (CPU_{util} * (P_{CPUmax} - P_{CPUmin}))$$

$$P_{CPUinclPUE} = P_{CPU} * PUE$$

$$CPU_{utilT} = (\sum_{t_0}^{t_n} d_{CPU}) \div (CPU_{util} * (t_n - t_0))$$

$$P_{TCPU} = P_{CPUinclPUE} * CPU_{utilT}$$

$$E_{CPU} = (P_{TCPU} * (t_n - t_0))/1000$$

$P_{CPUmin/max}$ values are different values derived by SPECpower benchmarks depending on CPU type (can be found on the CCF website)

PUE (Power Usage Effectiveness) values are different constants depending on cloud provider (can be found on the CCF website)

E for MEM / STO / NET

$$E_{GB} = (P_{GB} * PUE * (t_n - t_0))/1000$$

$$\sum_{t_0}^{t_n} d_{MEM} * 10^{-9} * E_{GB}$$

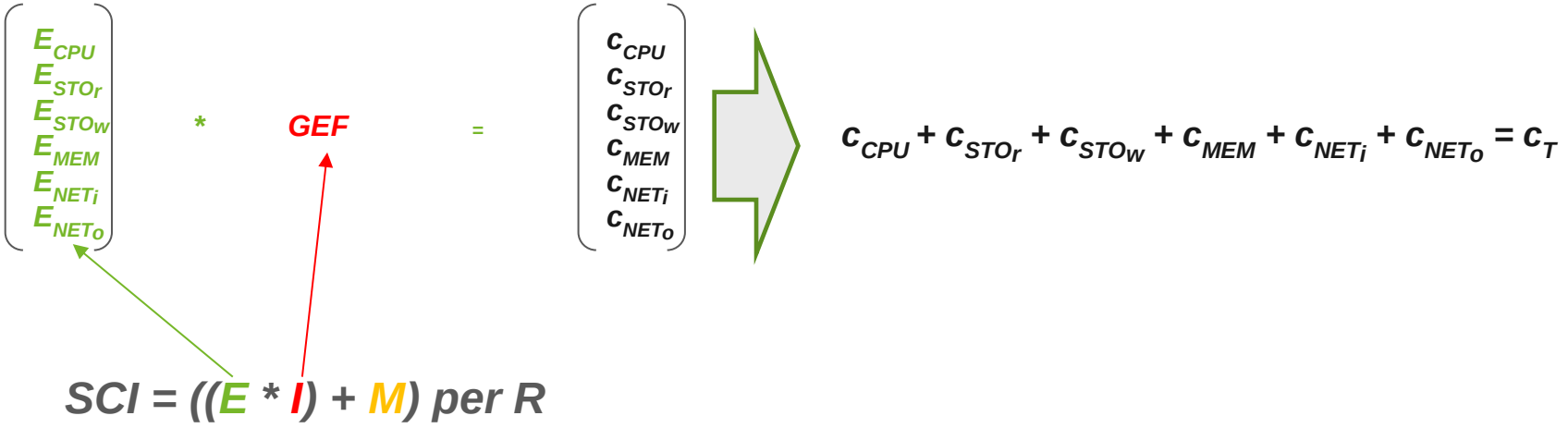
$$\sum_{t_0}^{t_n} d_{NET} * 10^{-9} * E_{GB}$$

$$\sum_{t_0}^{t_n} d_{STO} * 10^{-9} * E_{GB}$$

P_{GB} values are different constants depending on MEM, STO and NET type (can be found on the CCF website)

OpenTelemetry + Cloud Carbon Footprint Data

- Once we have the ***E*** values, the next value to calculate the SCI is the ***I*** value
- In order to do this, CCF publishes Grid Emission Factors (GEF) for the different regions in which the cloud providers operate (https://www.cloudcarbonfootprint.org/docs/methodology/#carbon-estimates-co2e)
- Using the GEF value, we can calculate carbon emissions of the energy consumption like this:



OpenTelemetry + Cloud Carbon Footprint Data

- What is still missing is the embodied carbon value (***M***) which depends on the server type used to operate the software and how much of the server is used by the software (e.g., when the software runs in a VM or container)
- The information about the total embodied emissions (TEE) is also available from Cloud Carbon Footprint (CCF) for most of the instance types of the main cloud providers (https://www.cloudcarbonfootprint.org/docs/methodology/#embodied-emissions)
- The TEE value first need to be scaled down to the time the server used and the fraction of the resources used by the current instance type:

$$M = TEE * 0.0289 * (t_n - t_0) * (CPU_{count_i} / CPU_{count_s})$$

- Furthermore, we cannot allocate the full embodied carbon on each transaction type, therefore we reduce it to the fraction used by the current transaction

$$M_T = M * CPU_{utilT}$$

$$SCI = c_T + M_T$$

$$SCI = ((E * I) + M) \text{ per } R$$

We can omit the R as our data is already scoped to individual transactions (T)

Demo of the OpenTelemetry Extension!

- Extension to the OpenTelemetry Java Agent
  - Publishes the resource demands per Span and as Metrics
  - Metrics only for top level transactions

https://github.com/RETIT/opentelemetry-javaagent-extension

Thanks a lot for your attention.

Questions?

Andreas Brunnert
brunnert@retit.de

RETIT
Resource Efficient Technologies & IT Systems