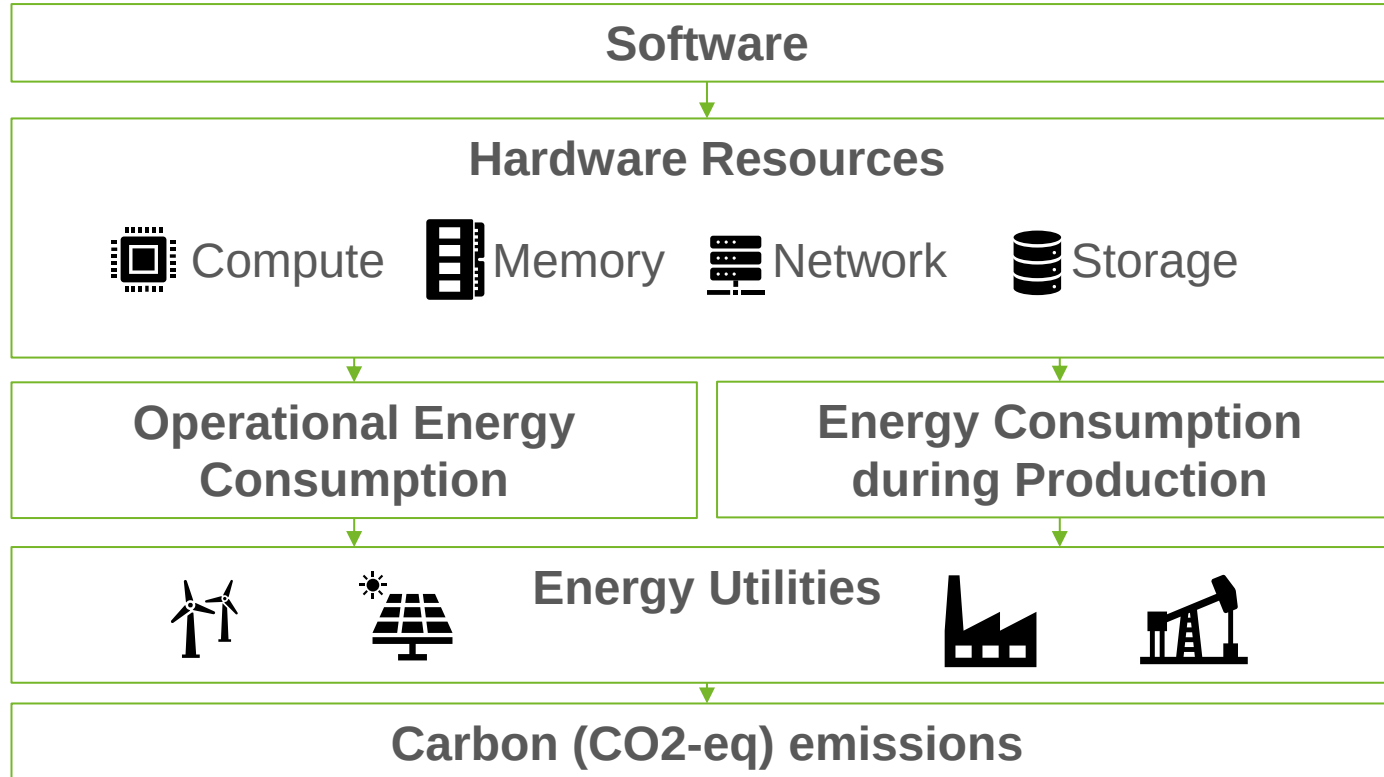


Wie misst man die CO2-Emissionen von Microservices?

Andreas Brunnert

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

CO2-Emissionen von Software



Software Carbon Intensity (SCI) Spezifikation

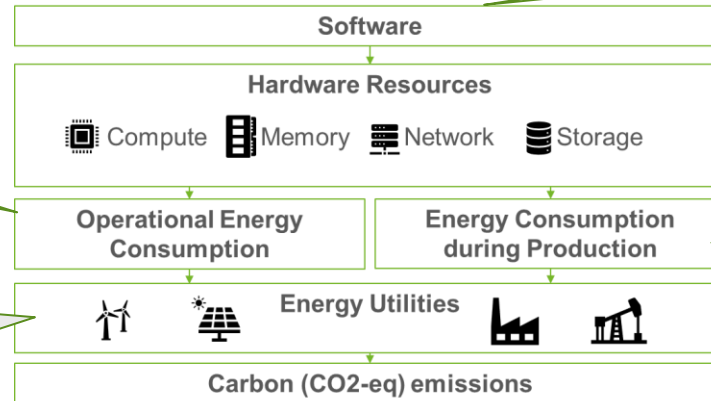
$$SCI = ((E * I) + M) \text{ per } R$$

R = Funktionale Einheit; so skaliert die Software, zum Beispiel pro Benutzer oder pro API-Call

E = Dies ist die Energie, die ein Softwaresystem zur Laufzeit verbraucht.

I = Kohlenstoffemissionen pro kWh Energie, gCO₂eq/kWh

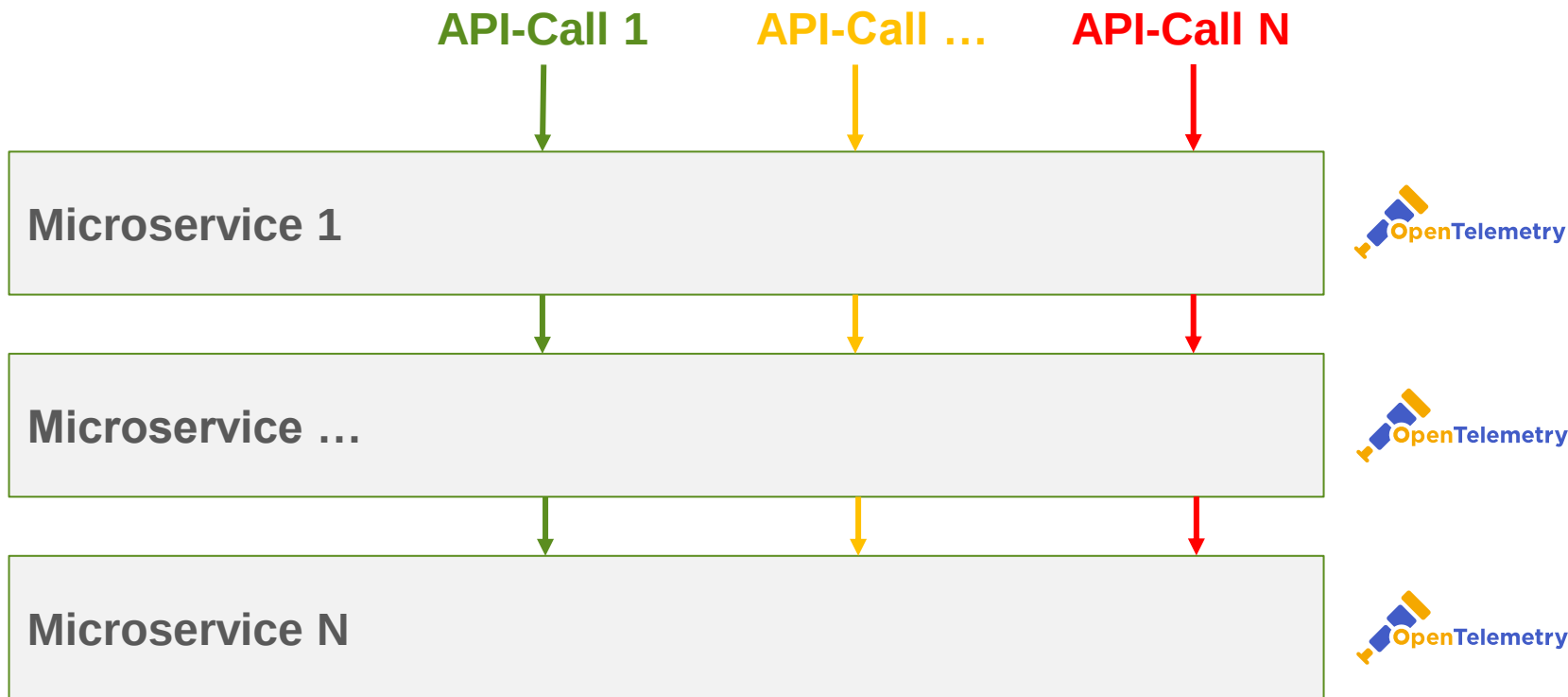
M = CO₂-eq Emissionen zur Herstellung der Hardware / Software



- Mehr Details: <https://www.iso.org/standard/86612.html> und <https://sci.greensoftware.foundation/>

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

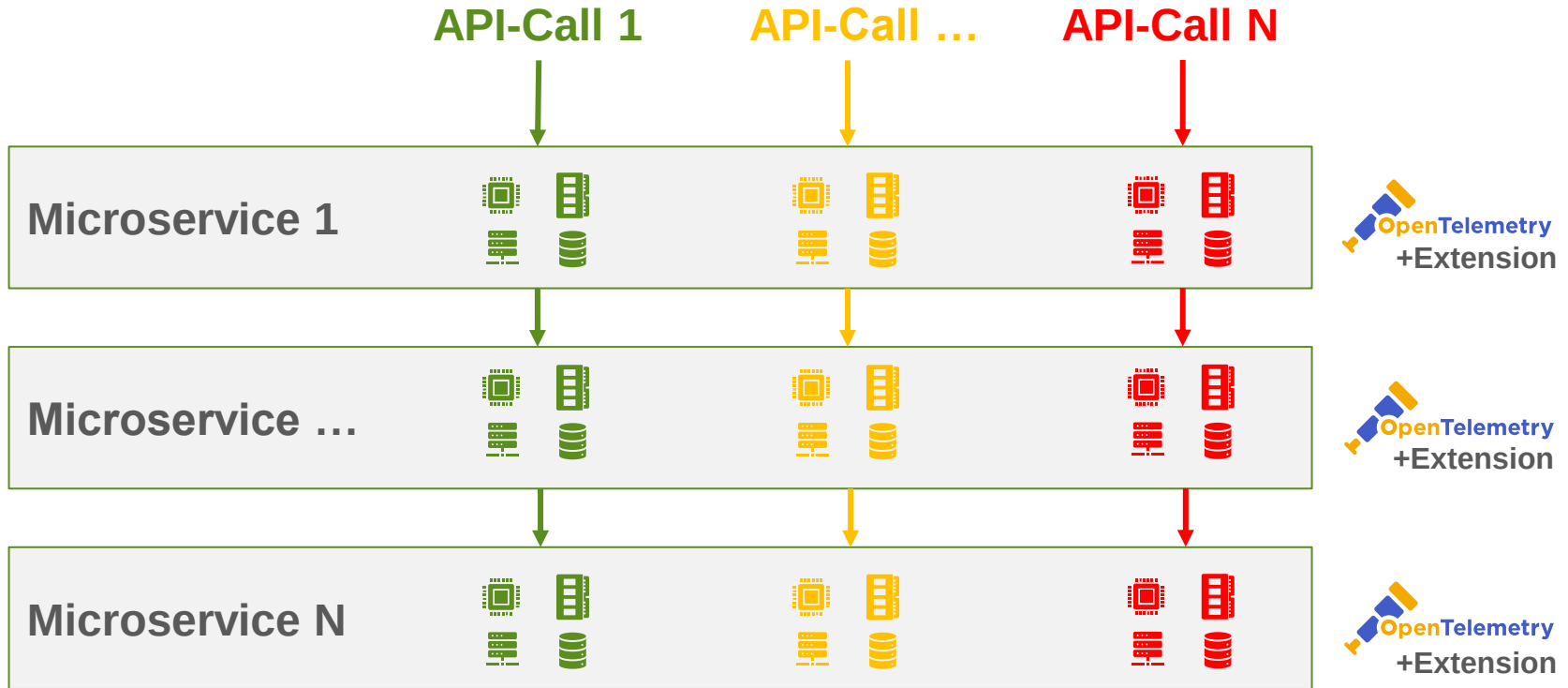
Microservices – *R* ist typischerweise ein “API-Call”



<https://opentelemetry.io/>

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

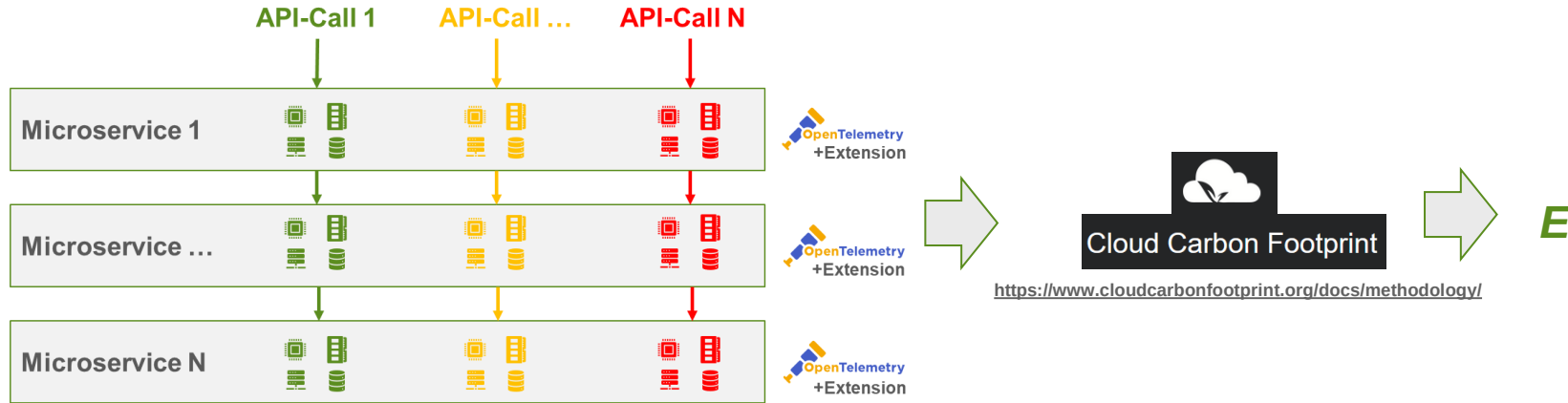
Microservices – E kann (oft) nicht direkt gemessen werden



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

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

Microservices – E wird über Modelle aus den erfassten Daten ermittelt



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

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

Microservices – *I* kann aus unterschiedlichen Quellen kommen

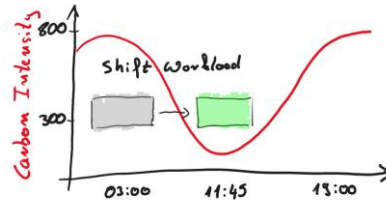


Cloud Carbon Footprint

<https://github.com/cloud-carbon-footprint/ccf-coefficients>



<https://app.electricitymaps.com/>



<https://www.carbon-aware-computing.com/>



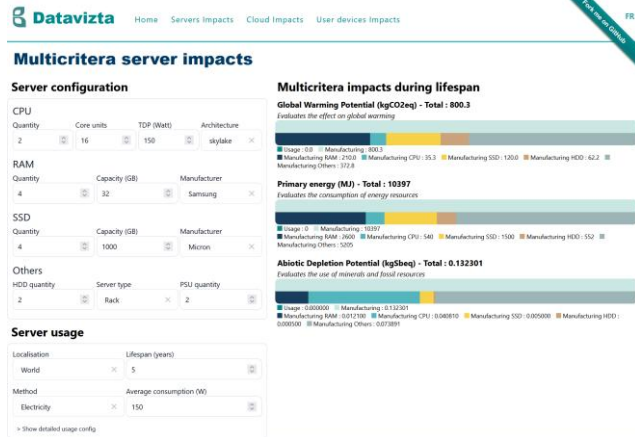
$$SCI = ((E * I) + M) \text{ per } R$$

Microservices – M wird von unterschiedlichen Quellen zur Verfügung gestellt



Cloud Carbon Footprint

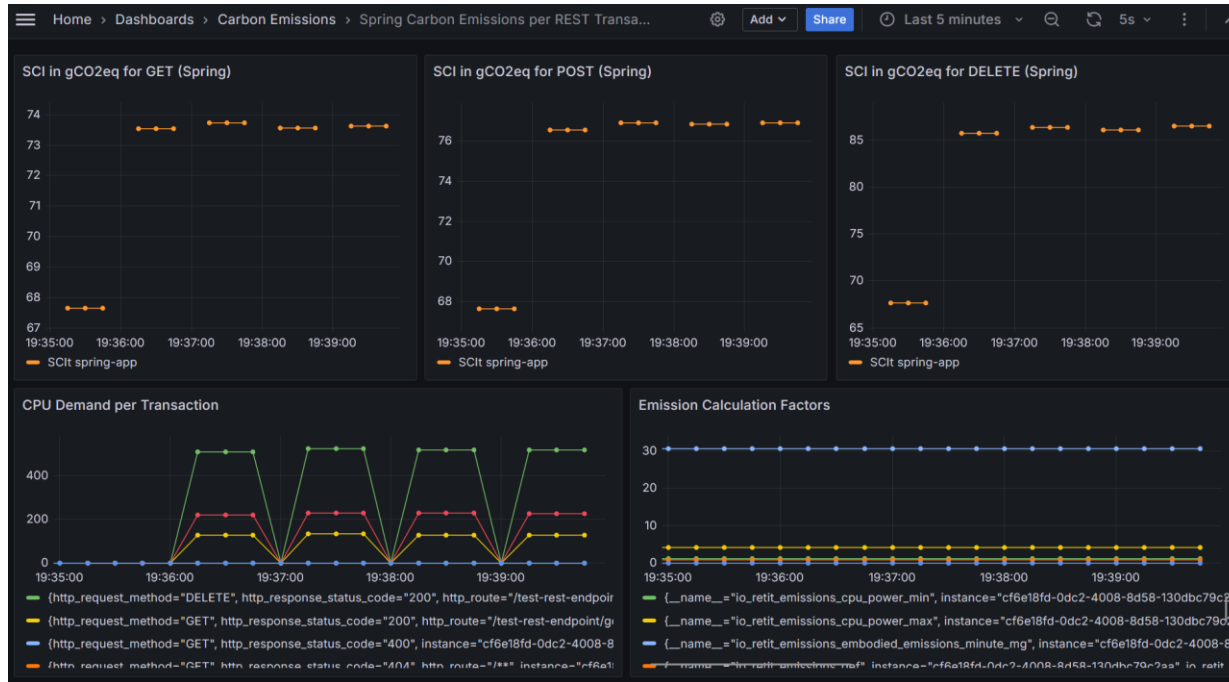
<https://github.com/cloud-carbon-footprint/ccf-coefficients>



M

<https://dataviz.boavizta.org/serversimpact>

Demo für einen Microservice in Java (Spring)



Demo: <https://github.com/RETIT/opentelemetry-javaagent-extension/tree/main/examples/spring-rest-service>

Referenzen

- OpenTelemetry Java Agent:
 - <https://github.com/open-telemetry/opentelemetry-java-instrumentation>
- OpenTelemetry Java-Agent Extension:
 - <https://github.com/RETIT/opentelemetry-javaagent-extension>
- Wissenschaftliche Artikel über die OpenTelemetry Java-Agent Extension:
 - <https://dl.acm.org/doi/10.1145/3629527.3652883>
 - https://fb-swt.gi.de/fileadmin/FB/SWT/Softwaretechnik-Trends/Verzeichnis/Band_44_Heft_4/SSP24_16_camera-ready_5255.pdf
 - <https://www.ret.it.de/wp-content/uploads/2025/04/devopssustain25-final80.pdf>
- Cloud Carbon Footprint – Methodology:
 - <https://www.cloudcarbonfootprint.org/docs/methodology>
- Cloud Carbon Footprint – Coefficients:
 - <https://github.com/cloud-carbon-footprint/ccf-coefficients>

Vielen Dank für Ihre Aufmerksamkeit!

Fragen?

Andreas Brunnert

brunnert@retit.de



Resource Efficient Technologies & IT Systems