

Klima freundliche Anwendungen

Aydin Mir Mohammadi

am@bluehands.de

[LinkedIn.com/in/aydin-mir-mohammadi/](https://www.linkedin.com/in/aydin-mir-mohammadi/)

github.com/bluehands

Green Software Development

! 1,5°-Ziel um die Klima-Katastrophe beherrschbar zu machen

! Ca 4% der globalen CO₂-Emissionen aus IT

Tendenz: Sehr Steigend (ca 10% pro Jahr)

! Ökologie ist nicht nur Klima. Es geht auch um Wasser, Ressourcen, Verschmutzung, Fläche

💡 Der Einfachheit wird alles in CO₂ umgerechnet



**Alle 3 Tage ein neues Datacenter
dieses Jahr (nur bei Azure)**



Eigene Disziplin

Clean Code

Agile

Green

With great power comes
great responsibility



Agenda

Einführung in Green Software Development

*

Wo entstehen Emissionen

*

Handlungsfelder

*

Strategien zur Minimierung der Emissionen

*

Tools & Data

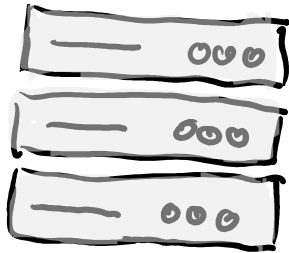


Quellen der CO₂-Emissionen

Scope 3

Gebundene Emissionen

50% - 70%

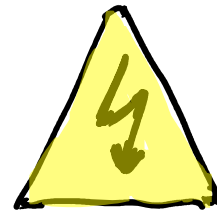


Hardware

Scope 2

Direkte Emissionen

30% - 50%



Strom

Handlungsfelder

Hardware - Effizienz



Energie - Effizienz



CO₂ - Intensität



Daten - Effizienz

[10010010]

Embodied Carbon



In der Hardware stecken Emissionen der Produktion, Lieferung & Entsorgung



Ziel: Die Lebensdauer von Hardware verlängern



5 Jahre statt 4 Jahre \Rightarrow 25%



Ziel: Mehr Software auf Hardware



Doppelte Anwendungen \Rightarrow 50%

Maximale Auslastung



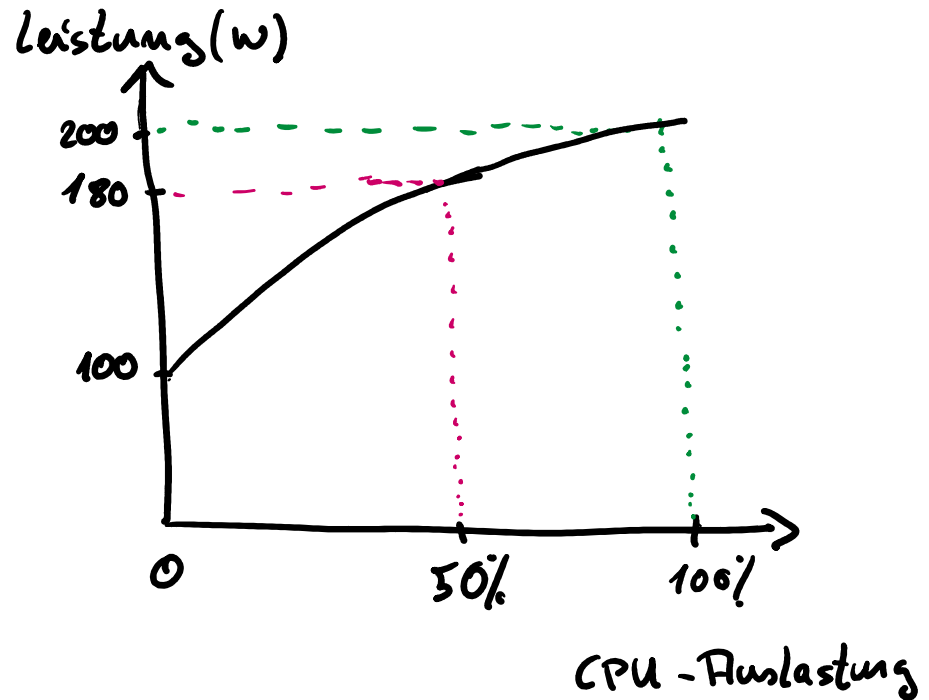
Server ohne Last
verbraucht Strom



Verhältnis Last
und Energie ist
nicht linear



Maximale Auslastung
des Servers erreichen



Messung CO₂-Emissionen



Anwendungsbereiche

- Optimierung : Energie-Messung , CPU
- Reporting : Billing - API , Provider-Dashboard
- Monitoring : CPU plus Offset



CO₂-Intensität , Embodied-Carbon , 3dB-Offset, PUE

Software Carbon Intensity

Energie

Embodied Carbon

$$SCI = \left((E \cdot I) + M \right) / R$$

CO₂ Intensity

Functional Einheit

Emissionen

Dienst
(vCPU, TB, GB)

Compute

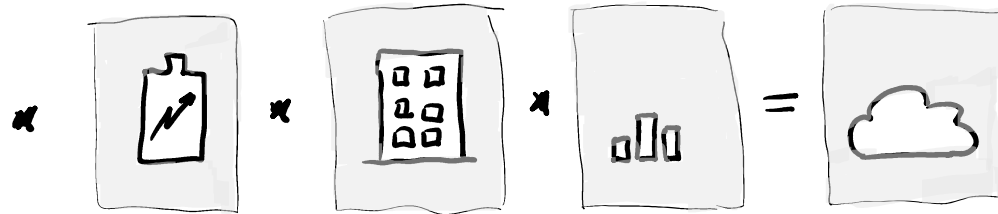
Storage

Network

Memory

Emissionsfaktor
(gWh)

Grid
CO₂-Intensität



Power Usage Effectiveness

Windows Energy Estimation Engine



Im Windows eingebaute Energie-Messung

```
c:\Windows\System32\sru>"C:\Users\MirMohammadi\Source\  
stretch tree of depth 7 check: 255  
64 trees of depth 4 check: 1984  
16 trees of depth 6 check: 2032  
long lived tree of depth 6 check: 127  
  
c:\Windows\System32\sru>powercfg.exe /srutil  
Completed with status 0 (0x00000000)
```

AppId	TimeInMSec	TotalEnergyConsumption
\Device\HarddiskVolume3\Users\MirMohammadi	24518	117
\Device\HarddiskVolume3\Users\MirMohammadi	28700	102

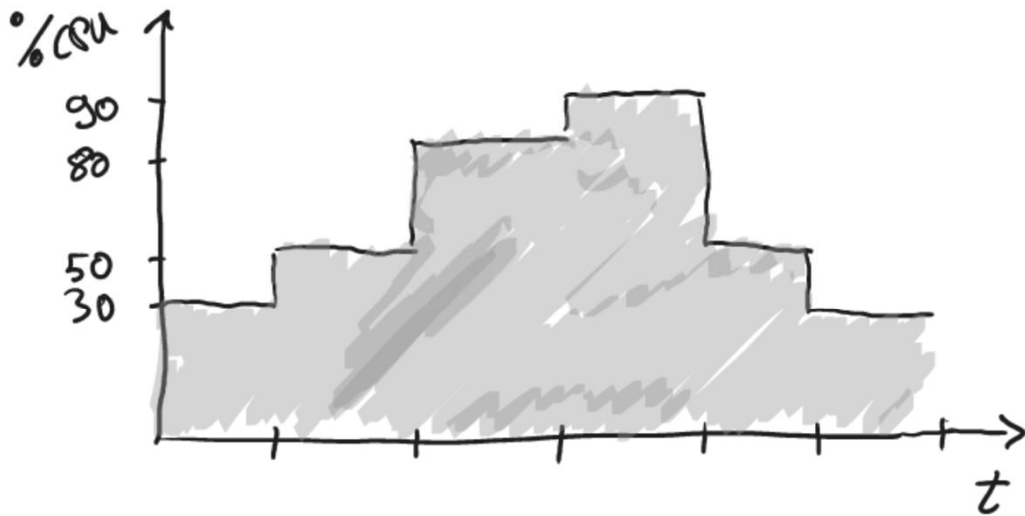


$$1 \text{ mJ} \cong 2,8 \cdot 10^{-10} \text{ kWh}$$

Virtuelle Maschine

$$P_{\text{book}} = \frac{n_{\text{CPU}} \cdot P_{\text{CPU}} + P_{\text{mem}} + n_{\text{GPU}} \cdot P_{\text{GPU}}}{1000}$$

Intel Xenon Platinum 8171M
TDP 165W, 2 vCore



→ $T \equiv \underline{4,2 \frac{\text{Wh}}{24\text{h}}}$

Weitere Meßmethoden

- Elektrische Leistungsaufnahme
- Code Carbon. io
- Green Coding Berlin
- Consumption & Billing API
- Kepler

Maßnahmen



Abschalten, Vermeiden, verzichten

Maximale Server Auslastung mit
effizienter Software

Low
Hanging
Fruits



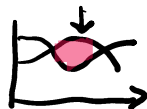
⇒ Container & K8s

⇒ PaaS, App-Service, Functions

⇒ SaaS

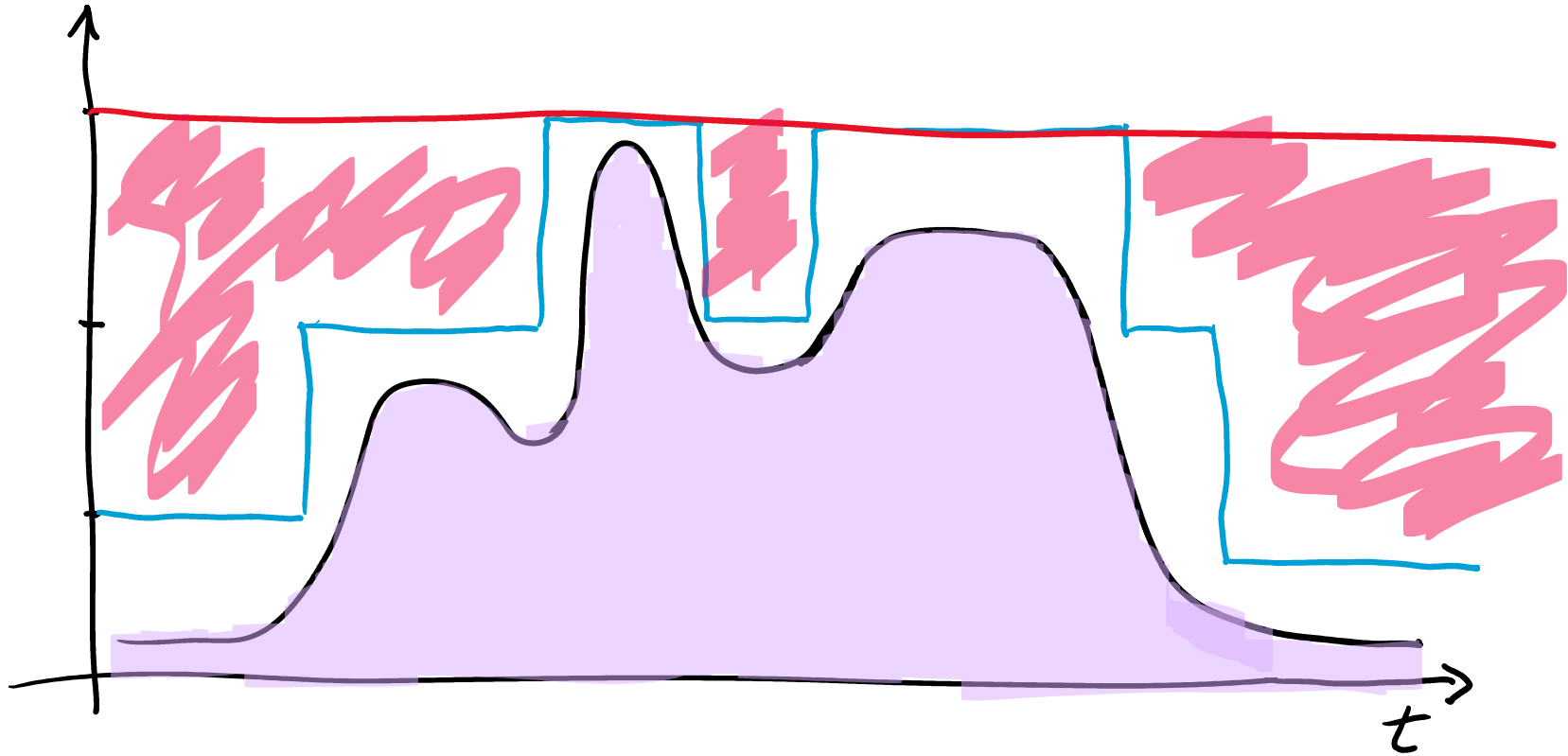


Demand - Shifting

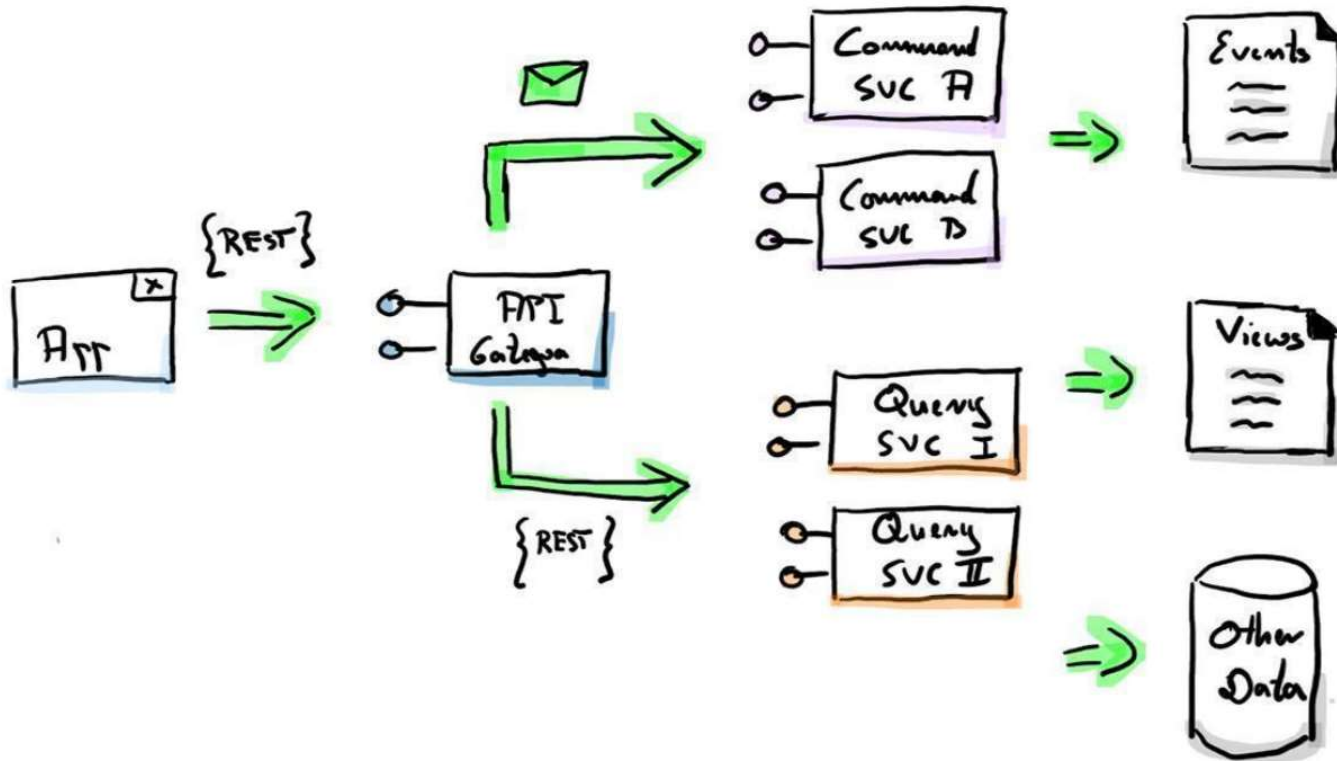


Demand - Shaping

Server Utilization



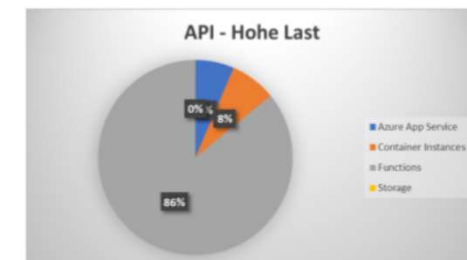
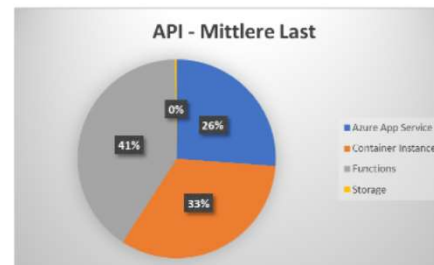
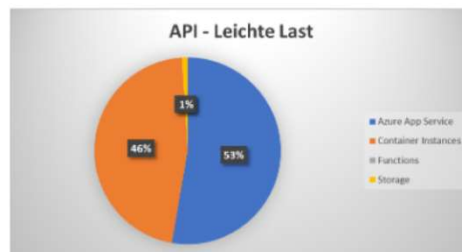
Gold Architecture



Szenarien

- Temporal entkoppelt
⇒ Messaging über Service bus
- Temporal gekoppelt
⇒ API über http

Emissionen



Leicht : ~ 0,1 Message / Minute

Hoch : ~ 10 Message / Minute

Mittel : ~ 1 Message / Minute

Dynamisches Hosting



Ein Load-Balancer vor AppService & Function
⇒ Front Door: Kein VNet notwendig

Scale Down



⇒ App Service abschalten
Function einschalten

Scale Up



⇒ Function abschalten
App Service einschalten

QoS

- Dynamische Skalierung mit Keda
 - Skalierung nach Last
- Zusätzlich
 - Max Replica nach CO₂-Intensität . Keda-Operator
- Kube - Green

Angebots - Orientierung

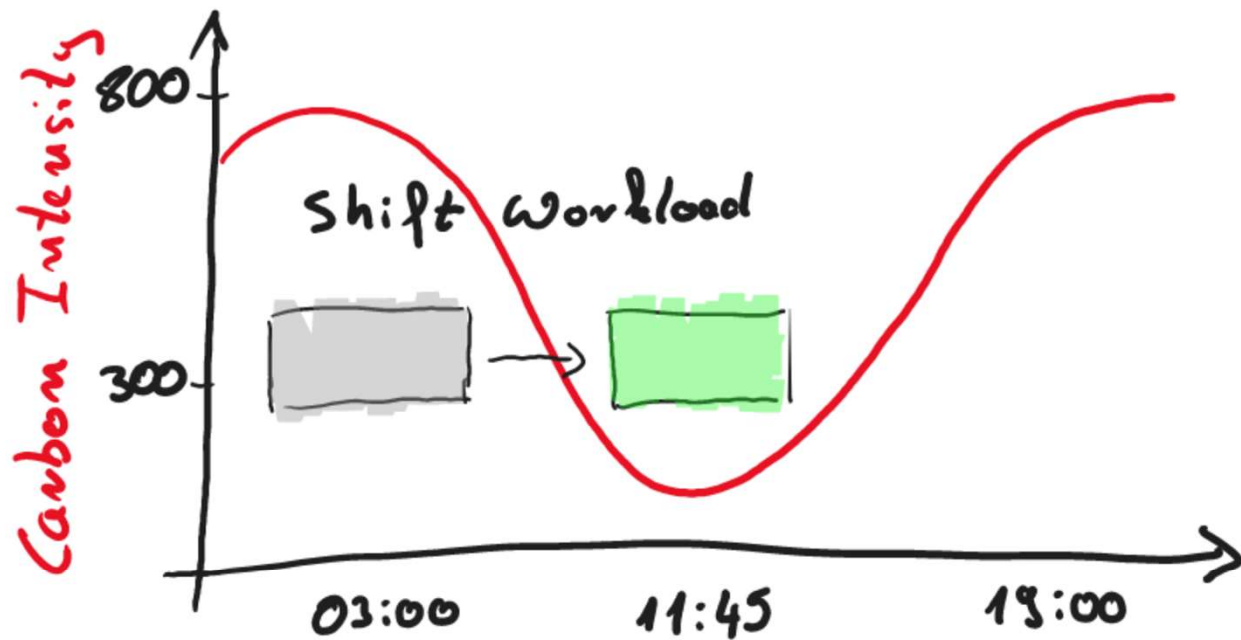
! Aktuell : Wir konsumieren, wenn wir es wollen.
Egal wie die Situation ist.

💡 Zukünftig : Wir konsumieren
wenn regenerativer Strom vorhanden (Zeit)
nur soviel wie regenerativen Strom
vorhanden ist (Menge)

➡ last verschieben

➡ last verringern

Carbon Aware Computing



Tools & Data => carbon-aware-computing.com

Azure Batch



Wir suchen den besten Ausführungs-zeitpunkt

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



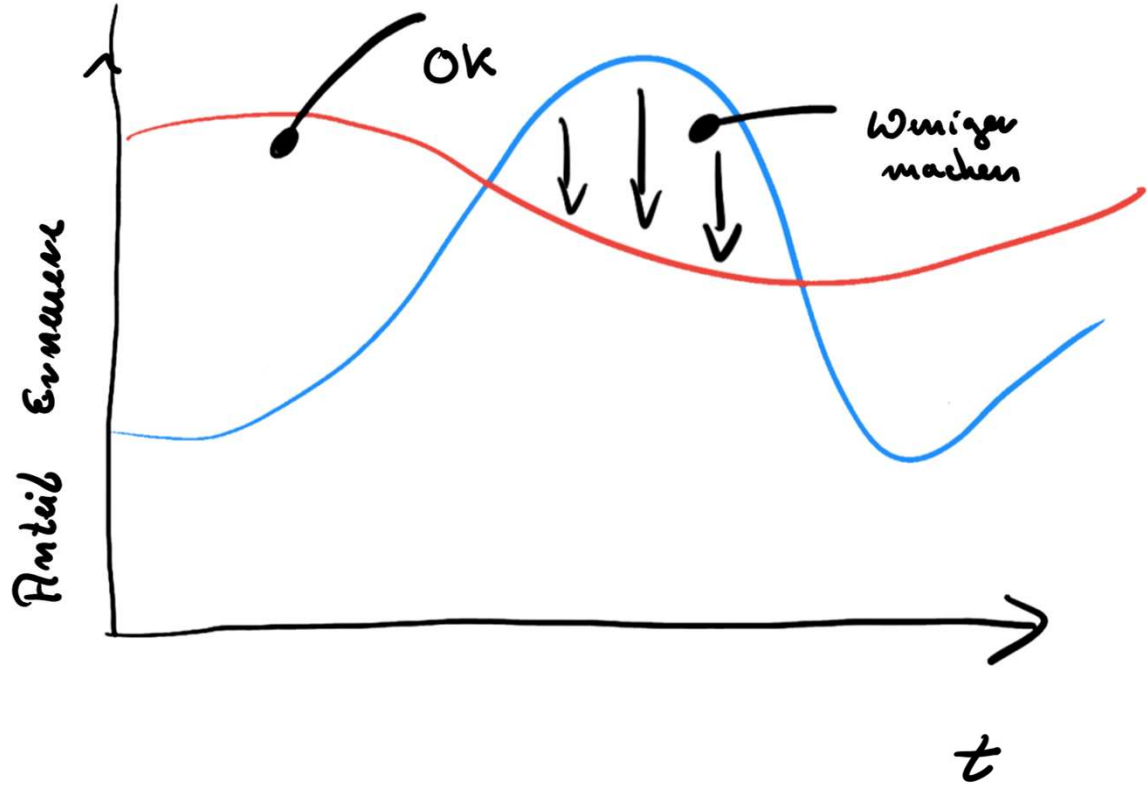
Wir erzeugen einen job schedule

→ DoNotRunUntil



Bonus: Spot VM nutzen

Demand Shaping



Klima Achtsame-Anwendungen



Wir machen den CO_2 -Verbrauch Transparent



Wir lassen die Anwender:innen entscheiden



Öko-Modus

Gemeinsamer Austausch



Green Software
Development
Karlsruhe

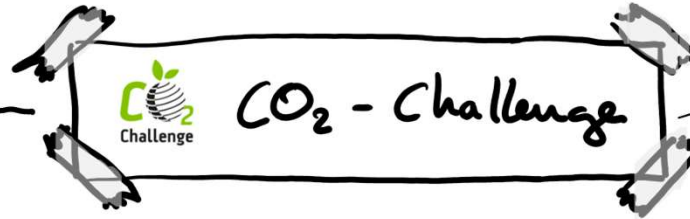
bit.ly/green-dev-ka



Gemeinsamer Austausch

- Green Software Development Manifesto
- Green Coding Berlin
- Green Software Development Stuttgart
 - Stuttgart
 - Frankfurt
 - Düsseldorf
 - München





- Challenge: Wir reduzieren die Emissionen unserer Software um 40%
- Ehrenamtliche Mentor:innen leiten die Unternehmen durch die Challenge
- Große Maßnahmen - Sammlung

Fragen

- Green Software Foundation
- MS Sustainable Software Blog
- Electricitymap.org ; Watttime.org
- Principles.green
- CloudCarbonFootprint.org
- SDI Alliance.org
- Climateaction.tech
- Carbon-aware-computing.com