

12.06.2023

Zweite Zwischenpräsentation

Bechtle x SGB-SMIT - Ökologischer Fußabdruck

Lukas Heppel, Yannick Pfeiffer, Benjamin Förster, Malte Neef

RECAP: Projektidee - Was werden wir machen?

- Übergeordnetes Ziel: “Den ökologischen Fußabdruck der IT-Landschaft eines global agierenden Unternehmens (SGB-SMIT) quantifizieren und optimieren.”
 - Hintergrund: SGB-SMIT möchte bis 2035 Scope 1 & 2 und bis 2050 Scope 1 & 2 & 3 CO2-Neutral werden → Daher: Möglichst exakte Bestimmung der aktuellen Emissionen nötig
- Bestimmung des gesamten CO2 Fußabdrucks der IT-Landschaft von SGB-SMIT zu aufwendig
- Daher Fokus auf Cloud-Umgebung
 - Cloud-Umgebung hat großen Impact auf die CO2 Emissionen der IT-Landschaft

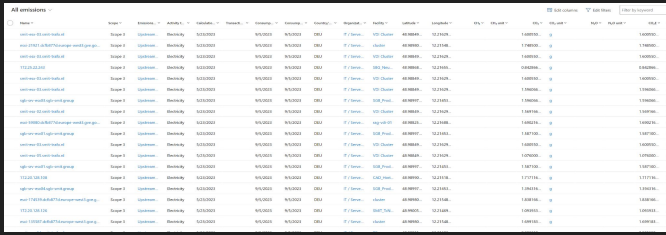
RECAP: unsere Ziele

- Quantifizierung der, durch den täglichen Betrieb der SGB-SMIT Cloud-Umgebung verursachten, CO2 Emissionen
 - Sammeln von Daten zu Stromverbrauch und CPU-Auslastung von Hosts an vier verschiedenen Standorten
 - Importieren dieser Daten in den Microsoft Sustainability Manager
 - Umrechnen dieser Werte in CO2 Emissionen
 - Darstellen dieser Daten in einem Dashboard
 - Auswerten der gesammelten Daten
 - Ableiten von Handlungsempfehlungen für die Optimierung der CO2 Emissionen

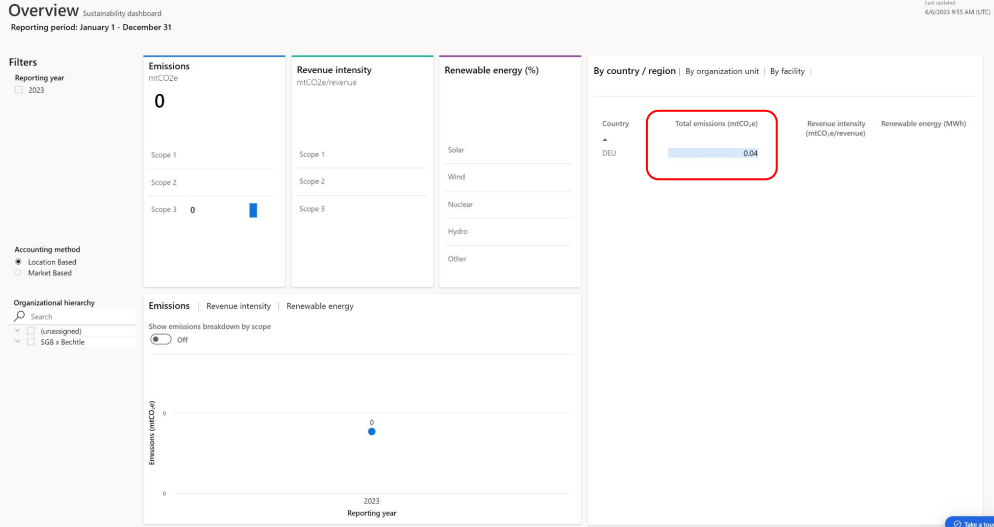
Zwischenstand

- Erreicht:

- Einarbeitung in API von vSphere
- Datenerfassung via PowerShell-Skript
- Vorverarbeitung und Visualisierung der Daten via Python-Skript
- Import der Daten in den MSM



```
process.py ~...
1 import glob
2 from matplotlib import pyplot as plt
3 import pandas as pd
4 from tqdm import tqdm
5
6 # Read all csv files from /src folder and concatenate them into one dataframe
7 df = pd.concat([pd.read_csv(f) for f in glob.glob('src/*.csv')], ignore_index = True)
8
9 # Rename ESXi to Id
10 df.rename(columns={'ESXi': 'Id'}, inplace=True)
11
12 # Convert Timestamp to datetime
13 # Original is day-month-year hour:minute:second
14
15 df['Timestamp'] = pd.to_datetime(df['Timestamp'], format='%d-%m-%Y %H:%M:%S')
16
17 # Create new column for WattHour
18 df['WattHour'] = -1
19 df['DeltaTime'] = -1
20
21 # Order columns, so that Watt, DeltaTime and WattHour are at the end
22 df = df[['Cluster', 'Id', 'Timestamp', 'Watt', 'DeltaTime', 'WattHour']]
23
24 # ----- Power Usage Calculation -----
25
26 # Calculate WattHour for each row by
27 # - finding the next entry with the same Id
28 # - calculating the deltaTime
29 # - multiplying the deltaTime with the Watt value
30 # - adding the result to the WattHour column
31
32 # Sort dataframe by Id and Time
33 df.sort_values(['Id', 'Timestamp'], inplace=True)
34 df.reset_index(drop=True, inplace=True)
35
36 # Iterate over dataframe with tqdm
37 for index, row in tqdm(df.iterrows(), total=df.shape[0]):
38     # dataframe is not iterated in order
39     if index + 1 >= df.shape[0]:
40         break
41     # Get next row
42     next_row = df.iloc[index + 1]
43     # Check if next row is same Id
44     if row['Id'] == next_row['Id']:
45         # Calculate delta time
46         delta_time = next_row['Timestamp'] - row['Timestamp']
47         # Calculate WattHour
48         watt_hour = delta_time.total_seconds() * row['Watt'] * 0.000277777778 # 1 WattSecond =
49         # Add WattHour to WattHour column
50         df.at[index, 'WattHour'] = watt_hour
51         df.at[index, 'DeltaTime'] = delta_time.total_seconds()
52
53 # Drop rows with -1 in WattHour column
54 df = df[df.WattHour != -1]
55
56 # Save dataframe to csv
57 df.to_csv('output.csv', index=False)
58
59 # ----- Power Usage Visualization -----
60
61 exit(0)
62
63 # sort by cluster and timestamp
64 df.sort_values(['Cluster', 'Timestamp'], inplace=True)
65 df.reset_index(drop=True, inplace=True)
66
67 # calculate accumulated watt hour for each cluster
68 df['AccumulatedWattHour'] = df.groupby('Cluster')['WattHour'].cumsum()
69
70 df.to_csv('output_visualization.csv', index=False)
71
72 # Plot WattHour for each cluster, x axis is timestamp, into one plot
73 fig, ax = plt.subplots(figsize=(20,10))
74
75 for name, group in df.groupby('Cluster'):
76     group.plot(x='Timestamp', y='AccumulatedWattHour', ax=ax, label=name)
77
78 plt.xlabel('Time')
79 plt.ylabel('WattHour')
80 plt.title('Power Usage per Cluster')
81 plt.show()
```



Overview Sustainability dashboard
Reporting period: January 1 - December 31

Filters
Reporting year: 2023

Emissions mtcO2e: 0
Scope 1: 0
Scope 2: 0
Scope 3: 0

Revenue intensity mtcO2e/revenue

Renewable energy (%)
Solar
Wind
Nuclear
Hydro
Other

By country / region | By organization unit | By facility |
Country: DEU
Total emissions (mtCO2e): 0.04
Revenue intensity (mtCO2e/revenue)
Renewable energy (MWh)

Accounting method
Location Based
Market Based

Organizational hierarchy
Show emissions breakdown by scope: Off

Emissions | Revenue intensity | Renewable energy
Emissions (mtCO2e) vs Reporting year (2023)

Zwischenstand - Datenanalyse

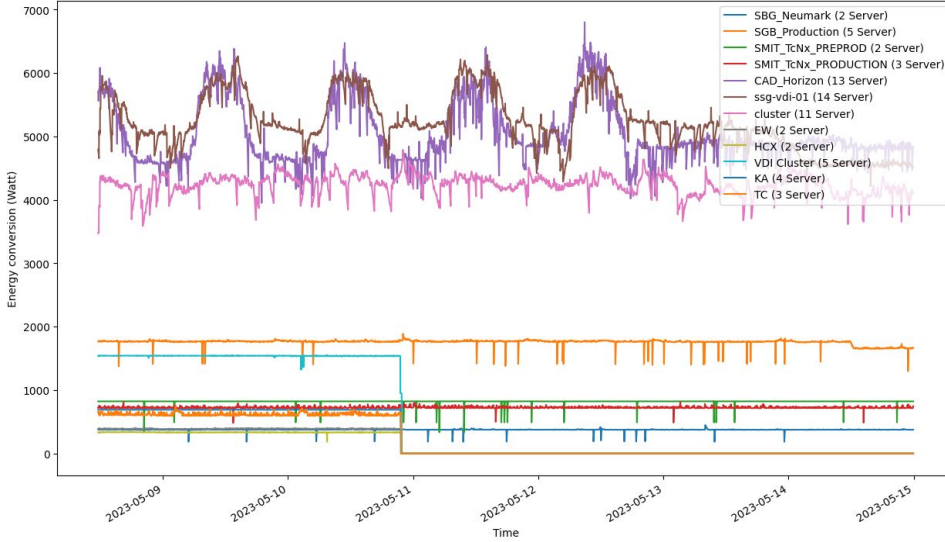
- Inputdaten: CSV Dateien für jeden Tag

Cluster ▼	ESXi ▼	Timestamp ▼	Watt ▼	CPUmhz ▼	VMCount ▼
EW	srv-esx-031.smit-tr.local	8-5-2023 10:34:40	210	279	2
EW	srv-esx-030.smit-tr.local	8-5-2023 10:34:40	181	642	2
HCX	srv-esx-021.smit-tr.local	8-5-2023 10:34:40	175	2519	2
HCX	srv-esx-022.smit-tr.local	8-5-2023 10:34:40	162	124	0
VDI Cluster	smit-esx-04.smit-trafo.nl	8-5-2023 10:35:00	231	87	0
VDI Cluster	smit-esx-01.smit-trafo.nl	8-5-2023 10:35:00	364	1933	9
VDI Cluster	smit-esx-03.smit-trafo.nl	8-5-2023 10:35:00	356	7085	8
VDI Cluster	smit-esx-05.smit-trafo.nl	8-5-2023 10:35:00	237	72	0
VDI Cluster	smit-esx-02.smit-trafo.nl	8-5-2023 10:35:00	350	2627	7
KA	srv-esx-011.smit-tr.local	8-5-2023 10:35:00	180	4272	7
KA	srv-esx-010.smit-tr.local	8-5-2023 10:35:00	181	8585	6
KA	srv-esx-020.smit-tr.local	8-5-2023 10:35:00	167	129	0
KA	srv-esx-012.smit-tr.local	8-5-2023 10:35:00	163	3663	6
TC	smit-esx-14.smit-trafo.nl	8-5-2023 10:35:00	202	1339	8

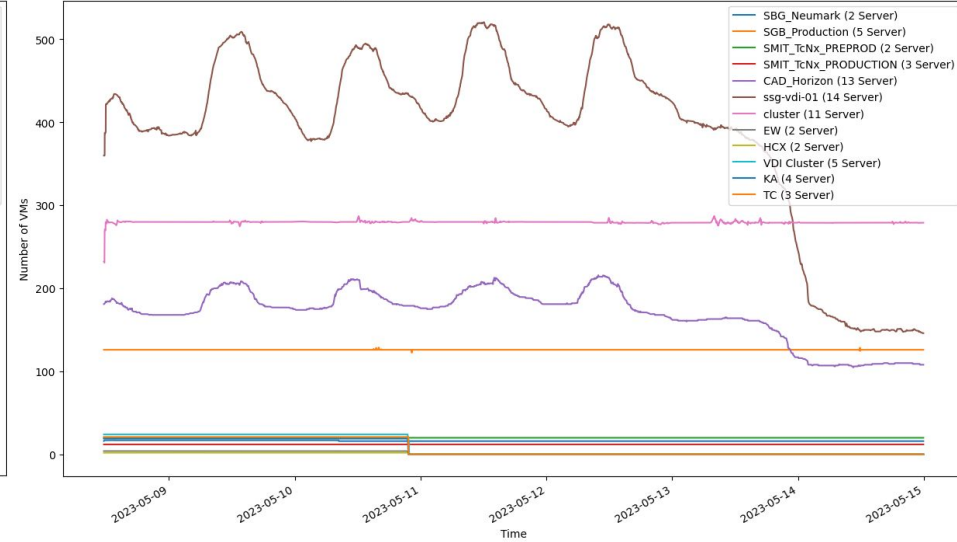
- Insgesamt 66 Server in 12 Clustern
- Bisher nur Daten vom 8.5. - 14.5.

Zwischenstand - Datenanalyse

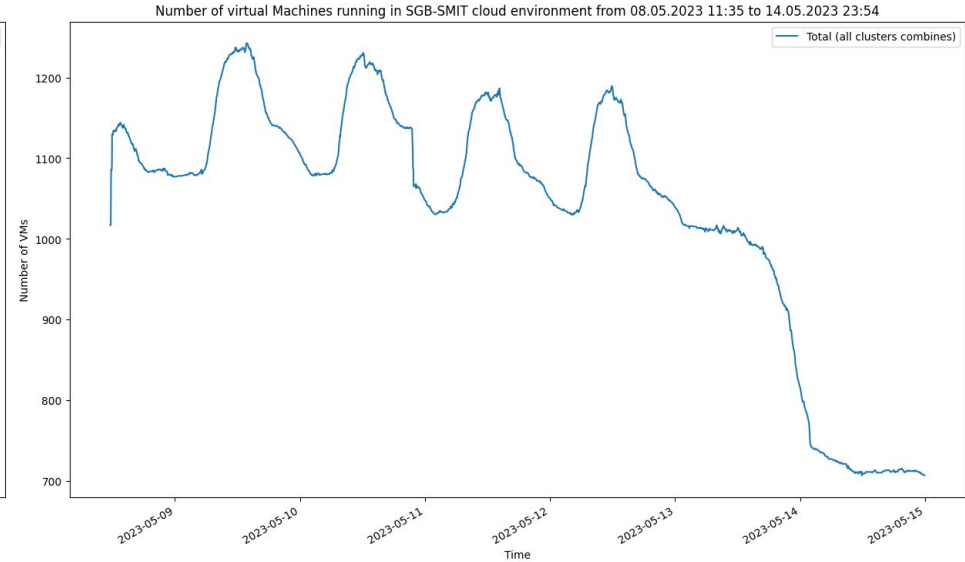
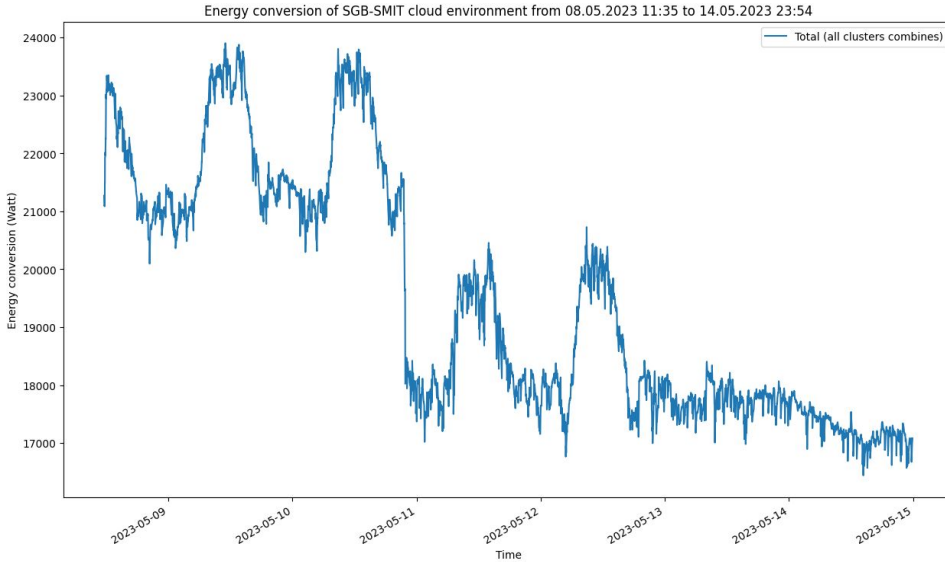
Energy conversion of SGB-SMIT cloud environment from 08.05.2023 11:35 to 14.05.2023 23:54 by cluster



Number of virtual Machines running in SGB-SMIT cloud environment from 08.05.2023 11:35 to 14.05.2023 23:54 by cluster



Zwischenstand - Datenanalyse

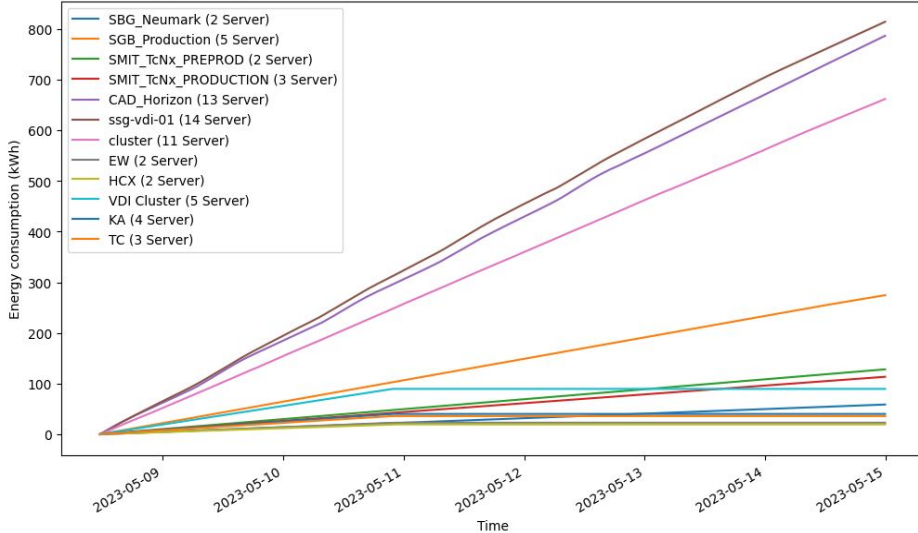


Veranschaulichung:

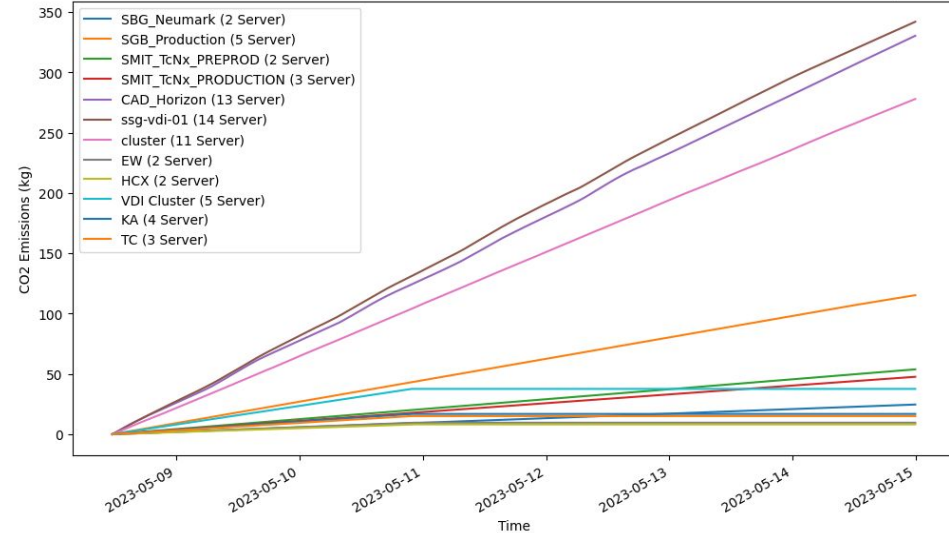
- 24.000W entspricht der Leitung von 400 herkömmlichen 60-Watt Glühbirnen oder von 3428 vergleichbaren LED-Glühbirnen

Zwischenstand - Datenanalyse

Estimated energy consumption of SGB-SMIT cloud environment from 08.05.2023 11:35 to 14.05.2023 23:54 by cluster



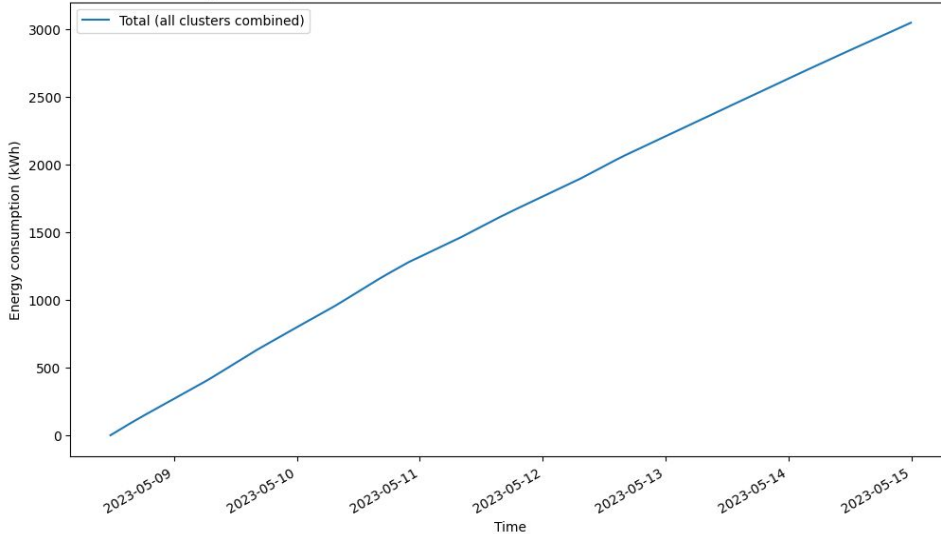
Estimated CO2 Emissions of SGB-SMIT cloud environment from 08.05.2023 11:35 to 14.05.2023 23:54 by cluster



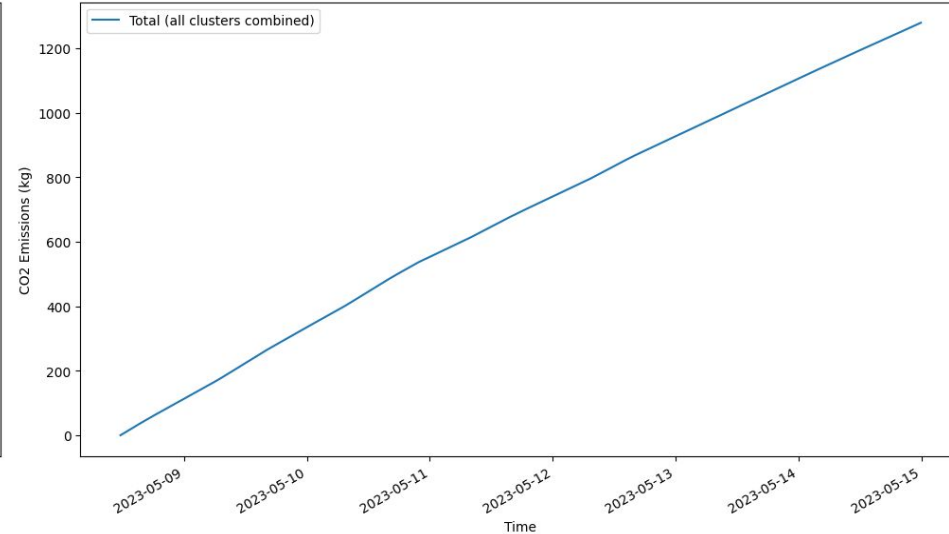
1kWh \cong 420g CO2 (deutscher Strommix 2021, Quelle: Umweltbundesamt)

Zwischenstand - Datenanalyse

Estimated energy consumption of SGB-SMIT cloud environment from 08.05.2023 11:35 to 14.05.2023 23:54



Estimated CO2 Emissions of SGB-SMIT cloud environment from 08.05.2023 11:35 to 14.05.2023 23:54



Veranschaulichung:

- Mit 3000kWH kann ein VW ID3 (19,3kWH / 100km) ca. 150.000km fahren
- Um 1200kg CO2 zu produzieren müsste ein VW Golf (140g / km) ca 8.500km fahren

Zeitplan bis zum Ende

- CO2 Emissionen in Dashboard visuell darstellen
- Optional Live-Schnittstelle implementieren
- Wert für durchschnittliche wöchentliche CO2 Emissionen bestimmen
- Liste mit Empfehlungen für Optimierung der CO2 Emissionen erstellen

12.6.



- Datenerhebung beginnen
- Microsoft Sustainability Manager konfigurieren
- Daten in Sustainability Manager importieren
- Stromverbrauch in Dashboard visuell darstellen
- Festlegen wie wir Stromverbrauch in CO2 Emissionen umrechnen

10.7.



30.7. (6.8.)



- Finalisieren von Projektbericht und Video
- Präsentation der Ergebnisse vor den Projektpartnern SGB-Smit und Bechtle

Zwischenfazit

- Zwischenziele erreicht
- Kommunikation im Team läuft reibungslos
- Alle Studenten des Teams tragen zu den Ergebnissen bei