LF ENERGY SUMMIT 2023

Deep Decarbonization

LF ENERGY SUMMIT 2023
Deep Decarbonization

Green Software Principles & Carbon Aware SDK

Dan Benitah & Szymon Duchniewicz
Avanade

@danbenitah

@Willmisz

Agenda

Intro to GSF and Green Software Engineering

Principles of Green Software

Deeper dive into Carbon Awareness (demand shaping)

Carbon Aware SDK as a means for embedding CA in your applications

SDK Demo

Next steps

WHAT IS THE GREEN SOFTWARE FOUNDATION?

What is the Green Software Foundation?

A non-profit foundation, aiming to build a trusted ecosystem of people, standards, tooling and best practices for Green Software

Manifesto

We have published our manifesto, which is our declaration of the policy and aims of the Foundation. It describes the set of core values of the Foundation and how we operationalise those values.

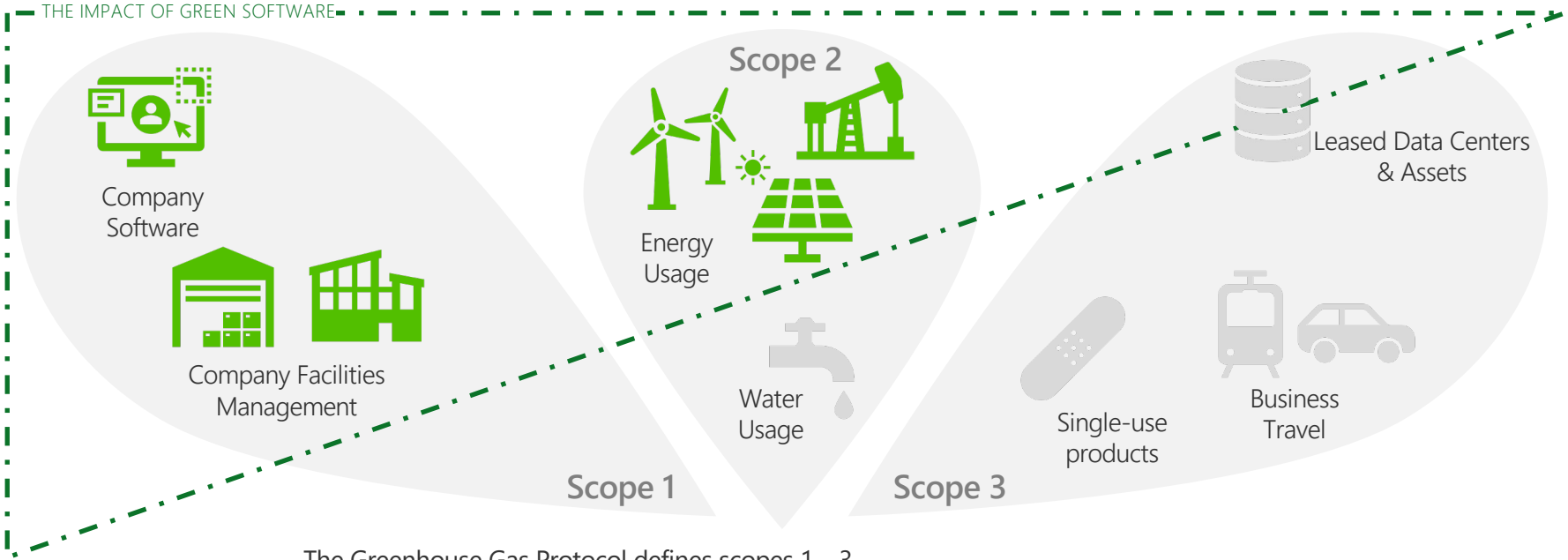
Mission

Build a trusted ecosystem of people, standards, tooling and best practices for creating and building green software.

Vision

Change the culture of building software across the tech industry, so sustainability becomes a core priority to software teams, just as important as performance, security, cost and accessibility.

Scope of Green Software



The Greenhouse Gas Protocol defines scopes 1 – 3

Scope 1 – Direct emissions.
Scope 2 – Indirect emissions from electricity purchases, steam, heating, and cooling.
Scope 3 – Indirect emissions from downstream and upstream activities.

Green
Software
Foundation

Green Software Principles

Core ideas behind building greener, more sustainable software

Green Software Principles

Energy Efficiency

Consume the least amount of electricity possible

Hardware Efficiency

Use the least amount of embodied carbon possible

Carbon Awareness

Do more when the electricity is clean and less when it's dirty

Carbon and Energy Efficiency

Emit the least amount of carbon possible.

Use the least amount of energy possible.

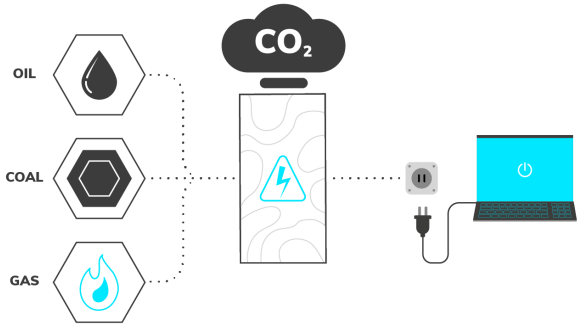
Greenhouse Gases

CO2eq

(carbon dioxide equivalent)

Energy efficiency

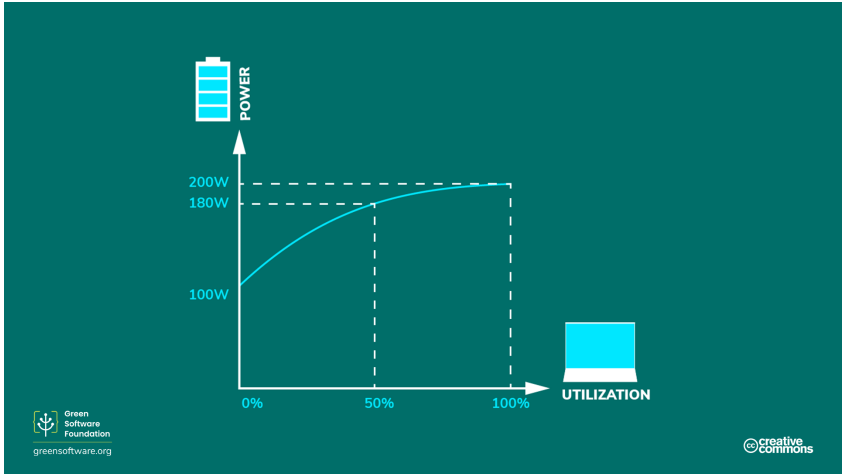
Sources of Energy



Green Software Foundation
greensoftware.org

creative commons

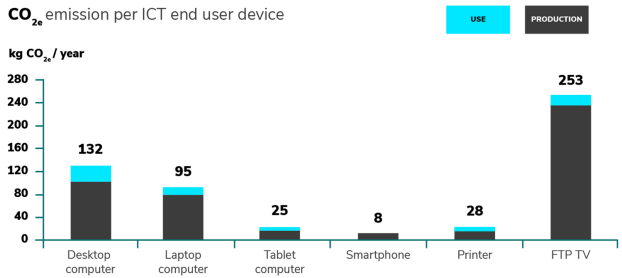
Minimising energy usage



Green Software Principles - continued

Hardware Efficiency

Use the least amount of embodied carbon possible.

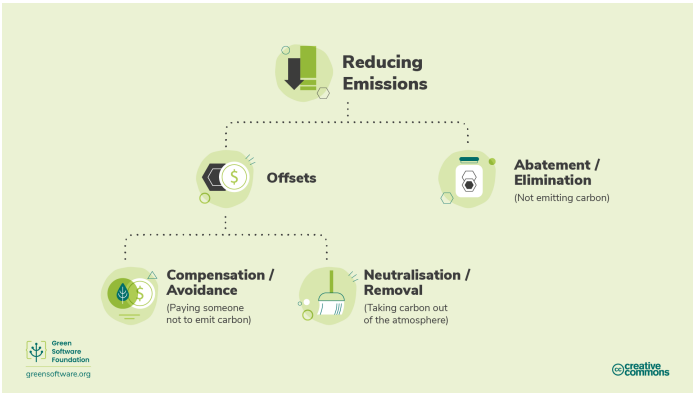


Measurement

What you can't measure, you can't improve.

Climate Commitments

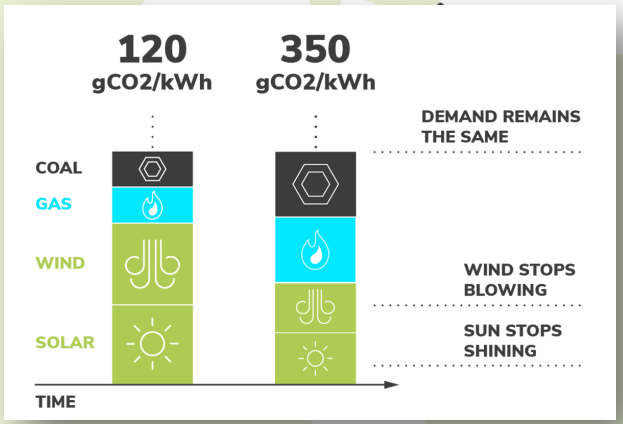
Understand the exact mechanism of carbon reduction.



Carbon Awareness

Do more when the electricity is cleaner and do less when it is dirtier

Carbon Intensity (in CO2e/kWh)

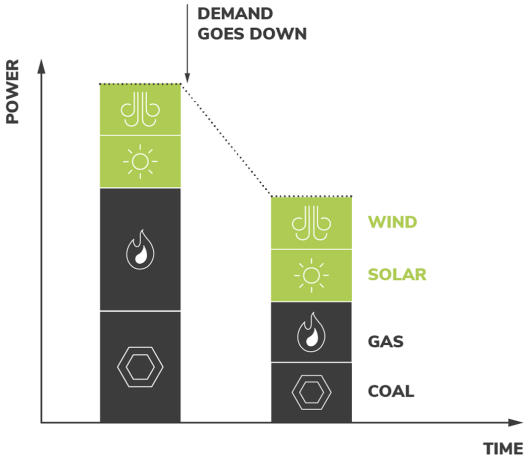


... measures how much carbon equivalent is emitted per kilowatt-hour of electricity consumed

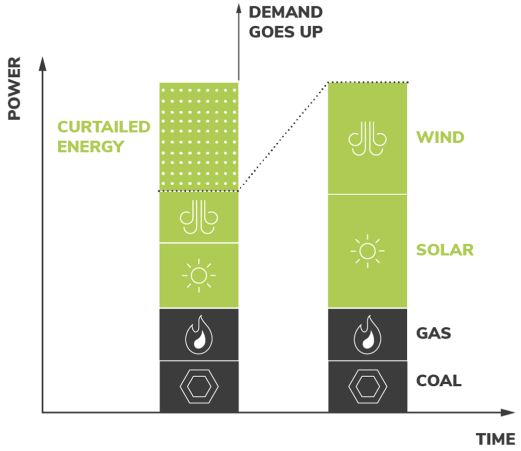
Green Software Foundation

greensoftware.org

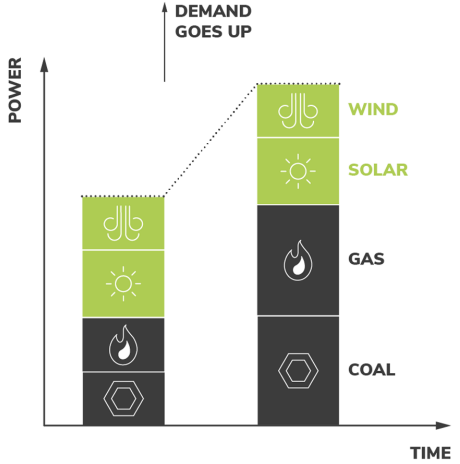
Variability and Energy market



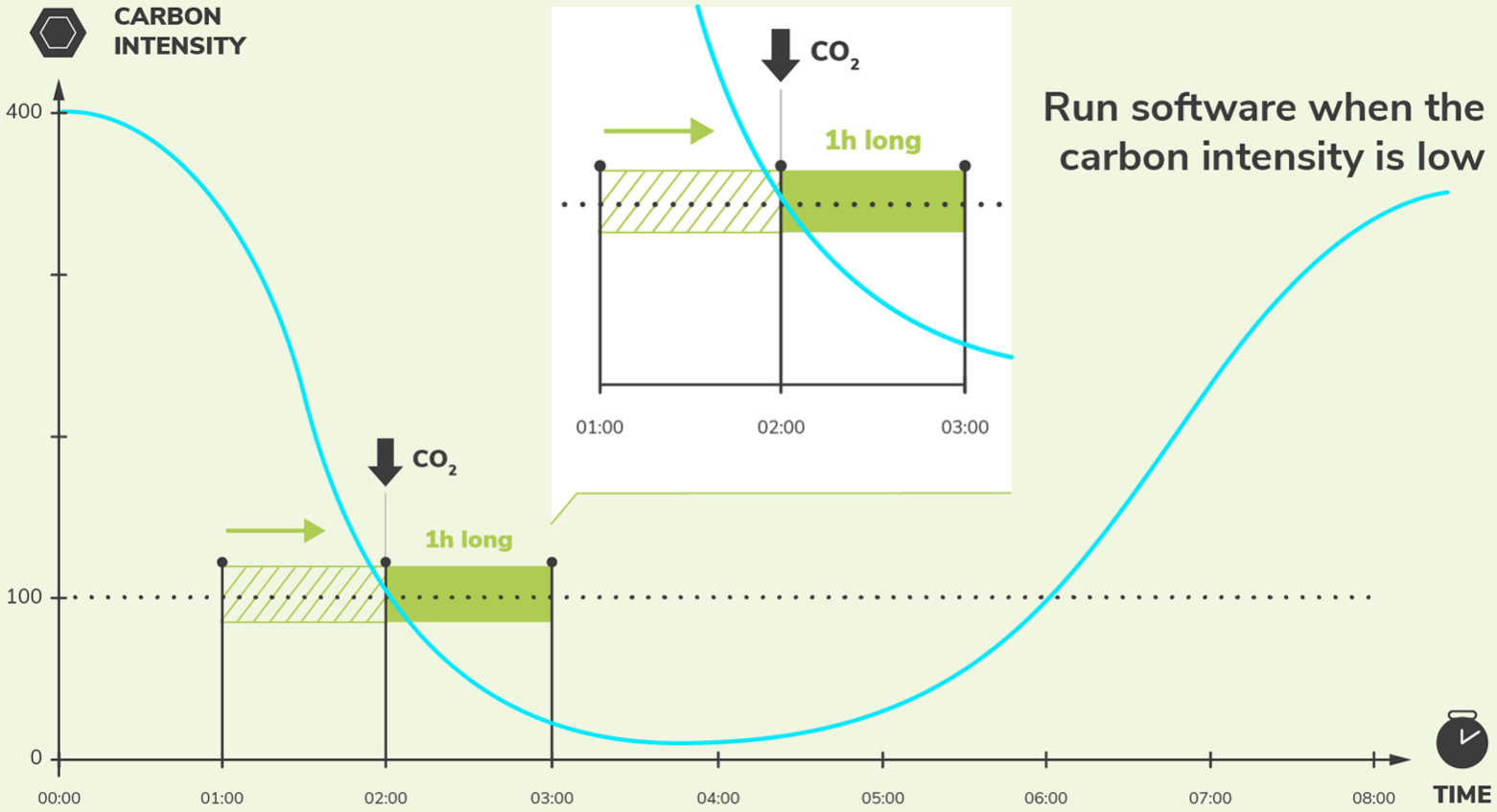
Buy **less** from **fossil** fuel plants



Buy **more renewable energy** that have been **curtailed**

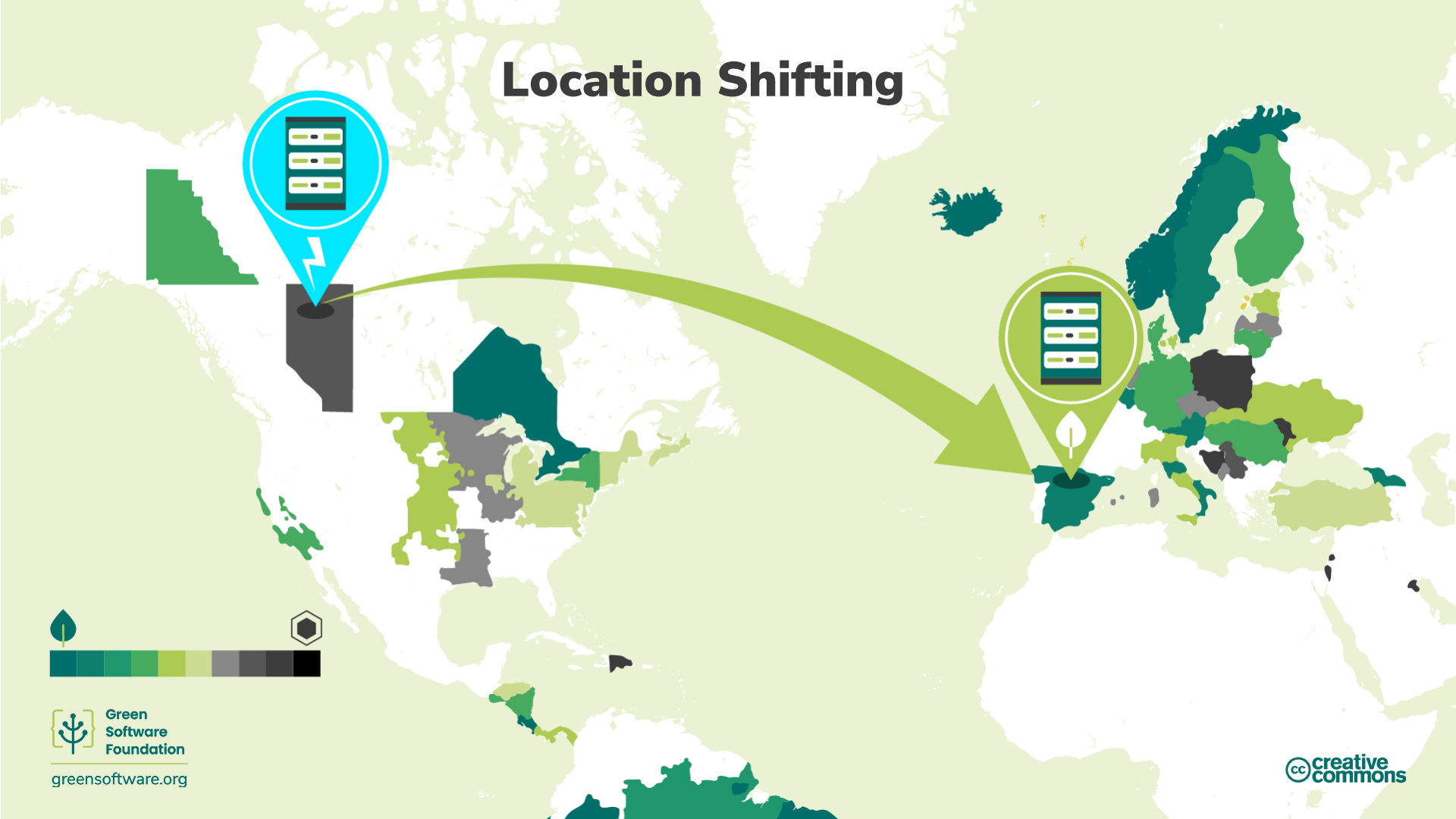


Buy **more** from **fossil** fuel plants



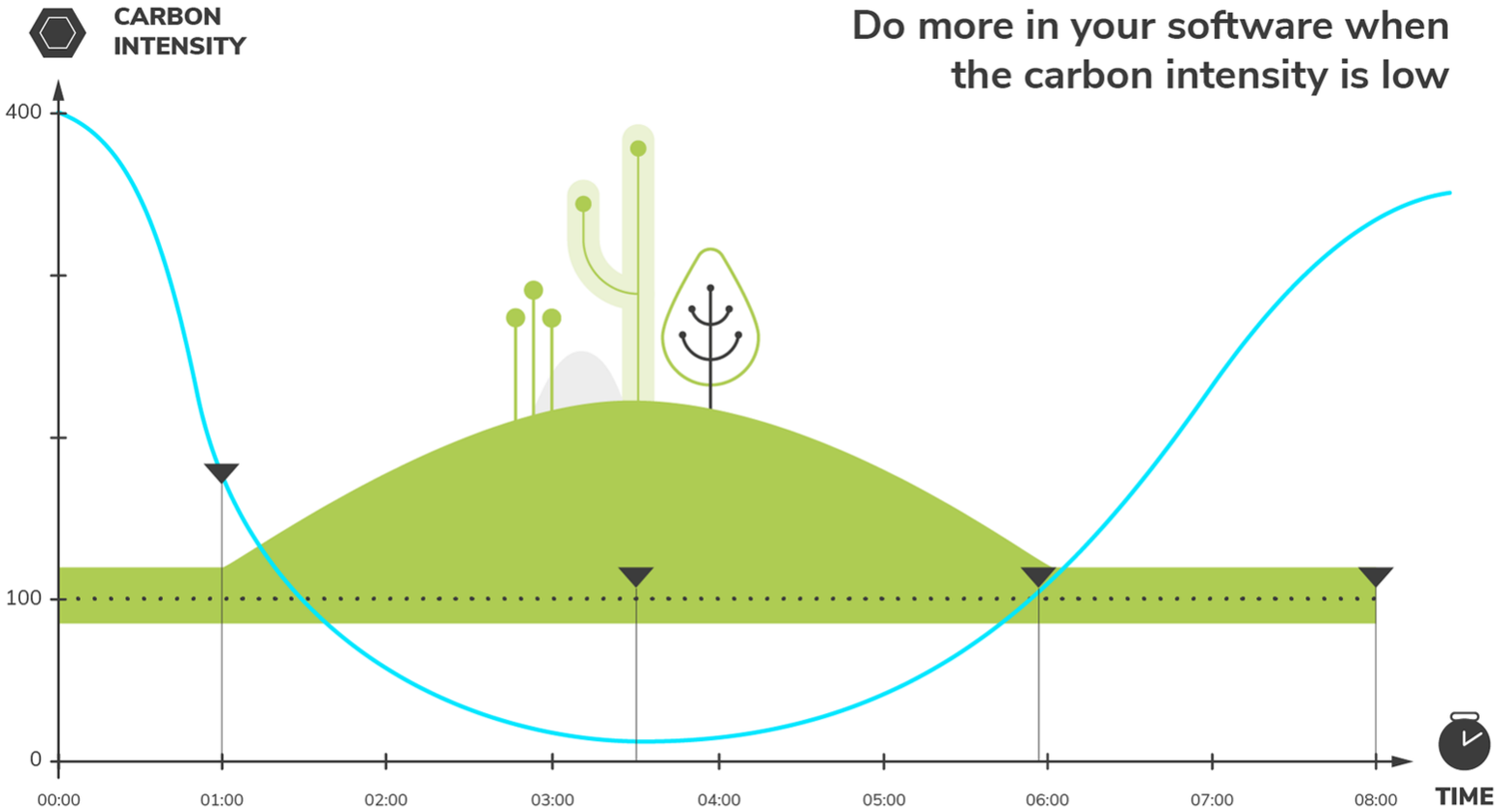
Time Shifting

Location Shifting



Green Software Foundation

greensoftware.org



Demand Shaping

Green Software Foundation

Carbon Aware SDK

Carbon Aware Pain Points

Understanding the return on investment

- **What should we do to reduce carbon intensity significantly? Where do we start?**

Disparate integration approaches

- **Teams reinventing the wheel each time**
- **Different approaches**
- **Different data providers**

Teams blocked without data providers

Auditability

Integrity of the approaches

- How can we be sure we were not, or are not Greenwashing?

Increasing pressure in legislature and regulatory compliance

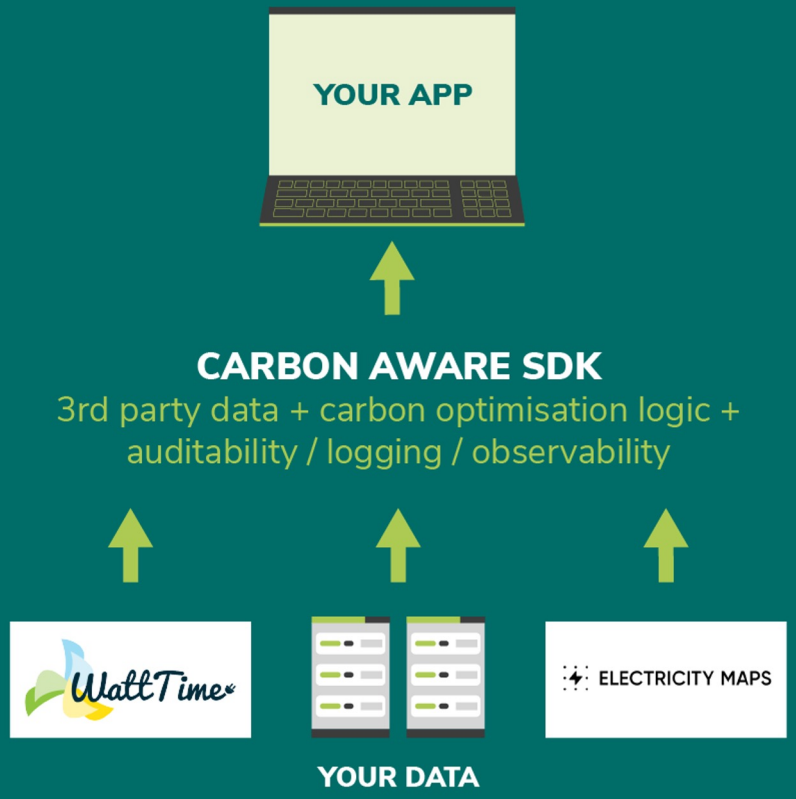
Businesses need to get ahead of the problem

- Simply to understand the scale of the effort required for green software initiatives

Need for a unified approach

Why a Carbon Aware SDK?

- Create carbon aware software and systems in a consistent way
- Measure operational carbon intensity for your platforms
- Run hypothetical models
- Identifying return on investment
- Prioritise carbon optimisation endeavours
- Centralise management of data providers

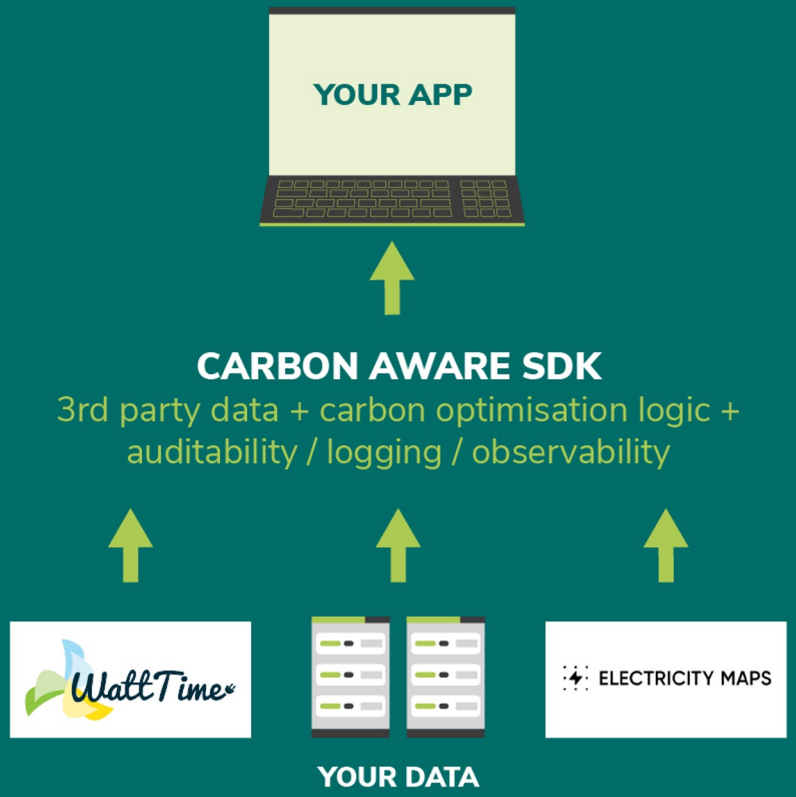


Why a Carbon Aware SDK?

- **Centralise carbon aware intelligence**

  > Provide capability across the business

- **Abstract the data providers**
- **Abstract carbon logic from your software**
- **Seamlessly query across multiple data providers**
- **Protect and manage secrets/keys**
- **Caching, throttling, logging**
- **Standardisation of data sources**



Demo in Jupyter Notebook

- Easy visualisation of the tool "in action"
- Demo of a Python library (OpenAPI generated)
- Used by Data Scientists

Carbon Aware SDK Demo Notebook

This notebook is supposed to give you an introduction to using the Carbon Aware SDK, by interacting with it deployed as a Web API. For a guide on setting it up as a Web API, see the GettingStarted.md guide in the main repo of the project: https://github.com/Green-Software-Foundation/carbon-aware-sdk/blob/dev/GettingStarted.md

**NOTE: This notebook was originally ran on data from 2022-08-15, between July 2022 - October 2022, using WattTime as the Emissions Data Source. Since then, WattTime has introduced further improvements in their prediction models, so the data you get from live queries might differ from Notebooks text

By Making Applications Carbon Aware, we can decrease carbon footprint of software by up to 76%!

Based on example of time shifting and location shifting from westcentralus to francecentral - times used for calculation:

- westcentralus: 2022-08-15 20:55
- francecentral: 2022-08-15 13:45

Time Shifting:

Scenario:

A software engineer is running a computationally intensive data processing job at 8:55 pm every day. The job is run in `westcentralus` Azure Data centre. They suspects that running the task at a different time of day, might produce less carbon emissions, since there is more solar energy in the grid during the day.

1. They wants to check, what is the current carbon intensity in that location.
2. They also would like to know, if it is better for the planet 🌍, to run this task at a different time of day.

They decides to use the Carbon Aware SDK to satisfy their curiosity, and possibly build a business case to move the job to another time, if their findings confirm his suspicions. They have cloned the repository, and hosted it locally as a Web API on https://localhost:5073

1. They starts by running a query on the `/emissions/bylocation` endpoint of the Carbon Aware SDK, for the region `westcentralus` and time 8:55pm:

He wonders whether running it at a different time of day might improve that value, so he makes a query to find the best time of day for running his job, by polling the `/emissions/bylocations/best` endpoint. (It will return a single location and time with the lowest carbon intensity in a given time window). He runs it, telling the SDK to look for best time between 12am on 15.08.2022 and 11:59pm on 15.08.2022.

parameters:

- locations: [`westcentralus`]
- start_time: 2022-08-15 00:00
- end_time: 2022-08-15 23:59
- duration_minutes: 0

In [6]:

```
locations = ["westcentralus"]        # List[str] |  (required)
start_time = parse('2022-08-15T00:00')   # datetime |  (optional)
to_time = parse('2022-08-15T20:55')      # datetime |  (optional)
duration_minutes = 0                     # int |  (optional) (default to 0)
query_params = {
        'location': ["westcentralus"] ,
        'time': parse('2022-08-15T00:00'),
        'toTime': parse('2022-08-15T23:59')
    }
try:
    #api_response = api_instance.get_best_emissions_data_for_locations_by_time(location=locations, time=start_tim
    api_response = api_instance.get_best_emissions_data_for_locations_by_time(query_params=query_params)

    print("Best time for running the job")
    pprint(api_response.body)
except openapi_client.ApiException as e:
    print("Exception when calling CarbonAwareApi->emissions_bylocation_get: %s\n" % e)
```

```
Best time for running the job
({'duration': '00:05:00',
  'location': 'PACE',
  'rating': Decimal('583.77338019000001168024027720093727111816406 25'),
  'time': '2022-08-15T03:30:00+00:00'},)
```

Outcome of time shifting: Carbon footprint decreased by up to 47%!

Location Shifting:

Now that we found the best time for the Software Engineer to run his computationally intensive data processing job, the software engineer is considering if he can further decrease the carbon footprint by moving the job to a different location. Let's suppose the Software Engineer is required to keep the job running in the US, but can move it to a different region inside of US. He can once again check this using the SDK, and the `/emissions/bylocations/best` endpoint!

[9]:
```
us_azure_regions = ['eastus', 'eastus2', 'southcentralus', 'westus2', 'westus3', 'centralus', 'northcentralus', '
start_time = parse('2022-08-15T23:59')   # datetime |  (optional)
to_time = parse('2022-08-15T23:55')      # datetime |  (optional)
duration_minutes = 0                     # int |  (optional) (default to 0)

try:
    api_response = api_instance.get_best_emissions_data_for_locations_by_time(location=us_azure_regions, time=sta
    print("Best Location for running the job at 11:55pm in US: ")
    pprint(api_response)

except openapi_client.ApiException as e:
    print("Exception when calling CarbonAwareApi->emissions_bylocation_get: %s\n" % e)
```

```
Best Location for running the job at 11:55pm in US:
[{'duration': '00:05:00',
 'location': 'AZPS',
 'rating': 344.27660883000004,
 'time': datetime.datetime(2022, 8, 15, 23, 55, tzinfo=tzutc())}]
```

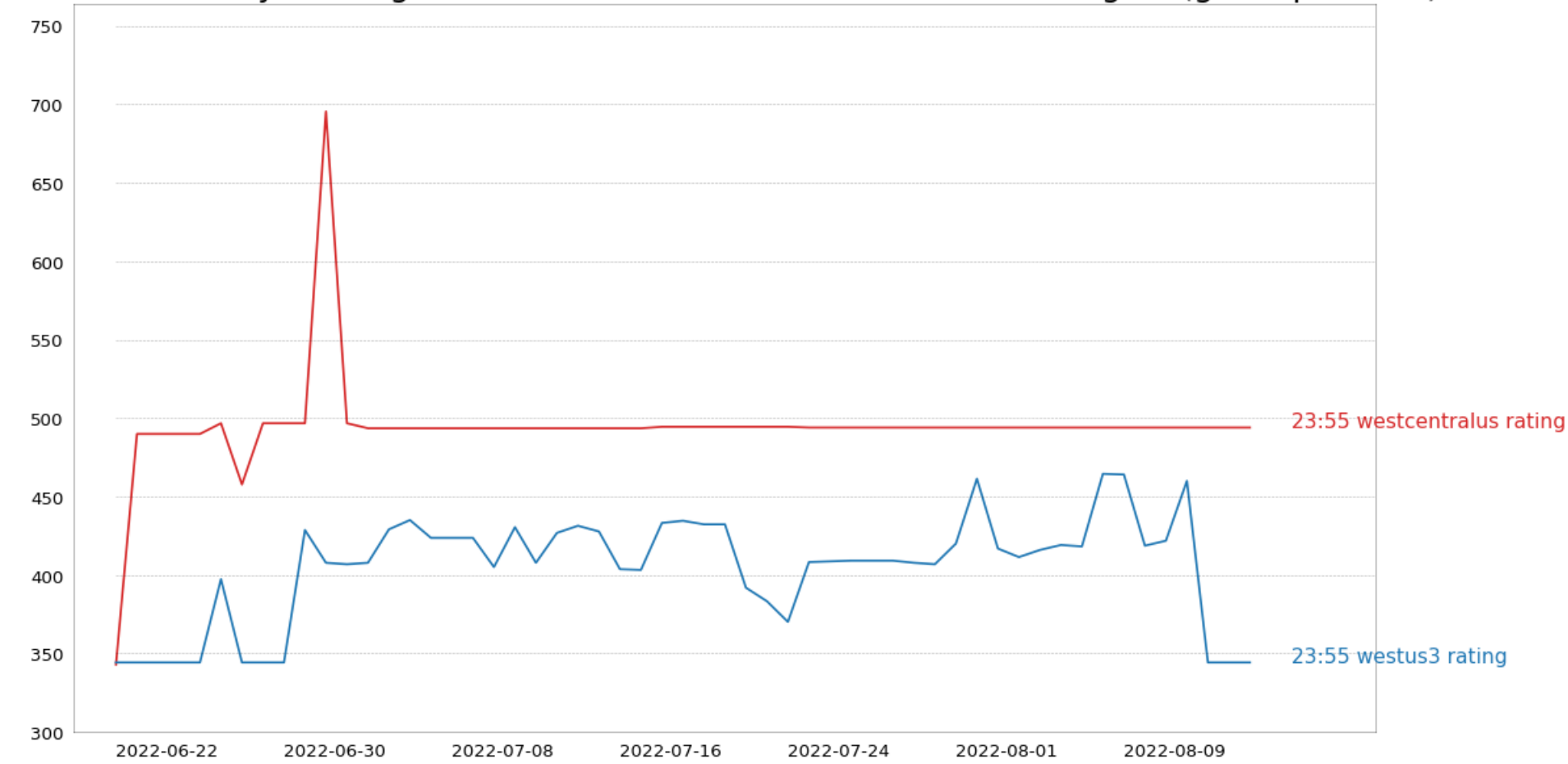
[10]:
```
# Get carbon intensity percentage decrease based on a single day of data:
carbon_intensity_new = api_response[0]['rating']
carbon_intensity_old = 493.96209
perc_decrease = (carbon_intensity_old-carbon_intensity_new)/carbon_intensity_old*100
print ("Carbon intensity percentage decrease as a result of location shifting to westus3 (on that day): {perc_dec
```

```
Carbon intensity percentage decrease as a result of location shifting to westus3 (on that day): 30.30%
```

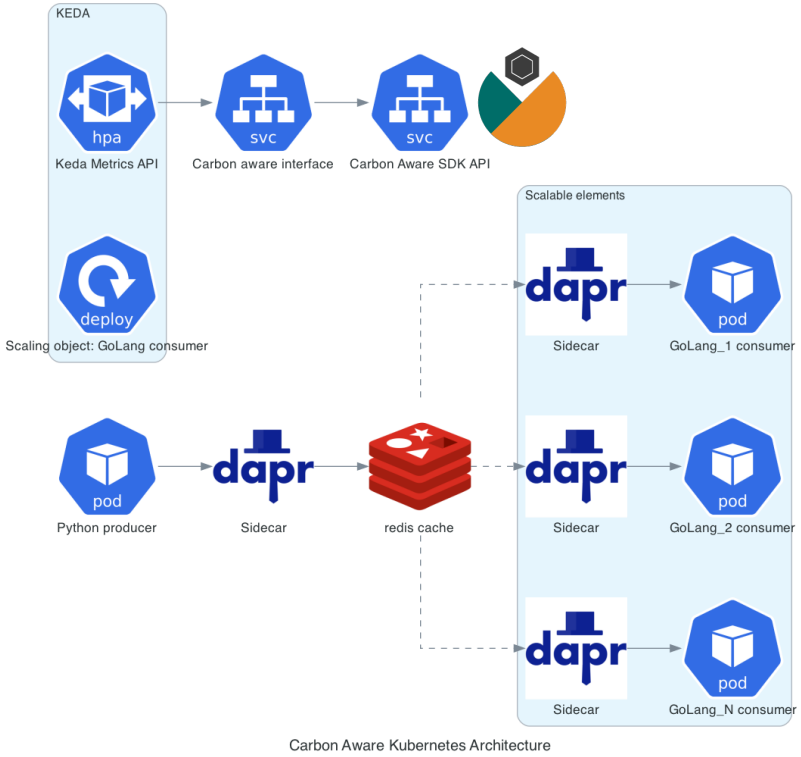
The best alternative data centre in the US at that time is `westus3` which corresponds to the wattTime region `AZPS`. Moving here can give the software engineer a further decrease in emissions by up to 30.30%!

Location shifting: over 55 days

Carbon intensity in the grid for westcentralus and westus3 Azure region (gCO2 per kWh)



Demo of CA SDK In Kubernetes environment



Carbon Aware Kubernetes Architecture

Tech used:

- Carbon awareness:
  - Carbon Aware SDK
- Container orchestration:
  - Kubernetes
- Communication:
  - Dapr
  - Redis
- Scaling:
  - Keda

Carbon Aware Pain Points Addressed

Unified approach

Reporting and forecasting

High integrity and auditability

Wrap up

What to do next?

From the 2023 State of Green Software report by Green Software Foundation
Find out more at stateof.greensoftware.org

Carbon-aware software is central to decarbonization

Green Software Foundation

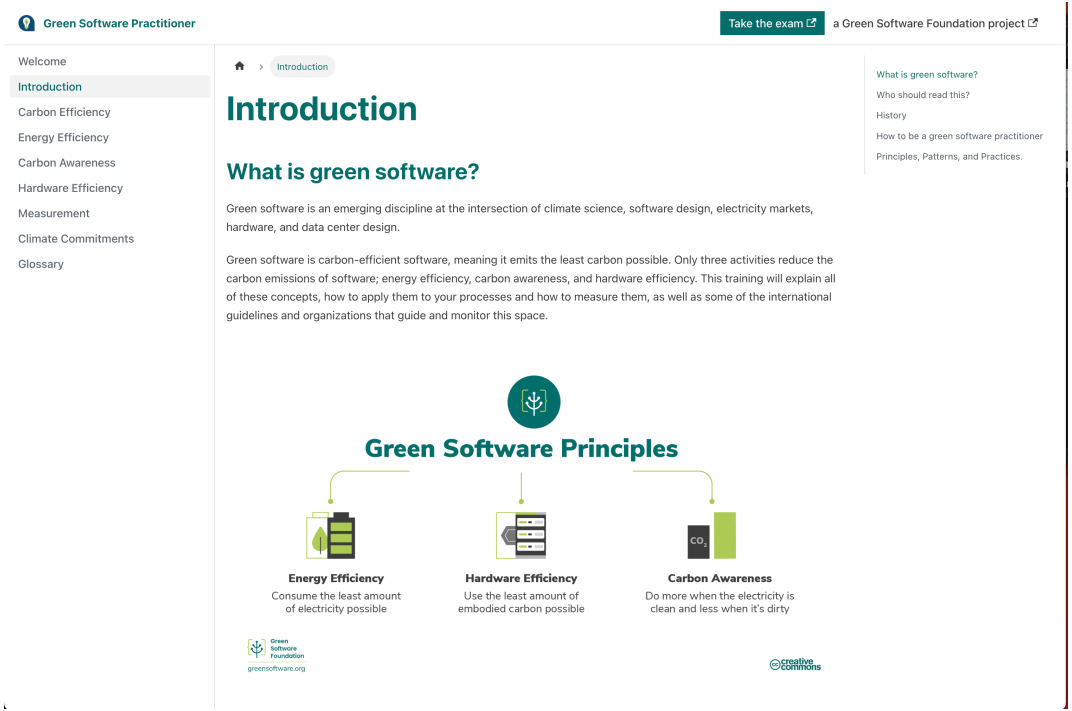
Become a Certified Green Software Practitioner

Learn & validate your knowledge

Take the exam

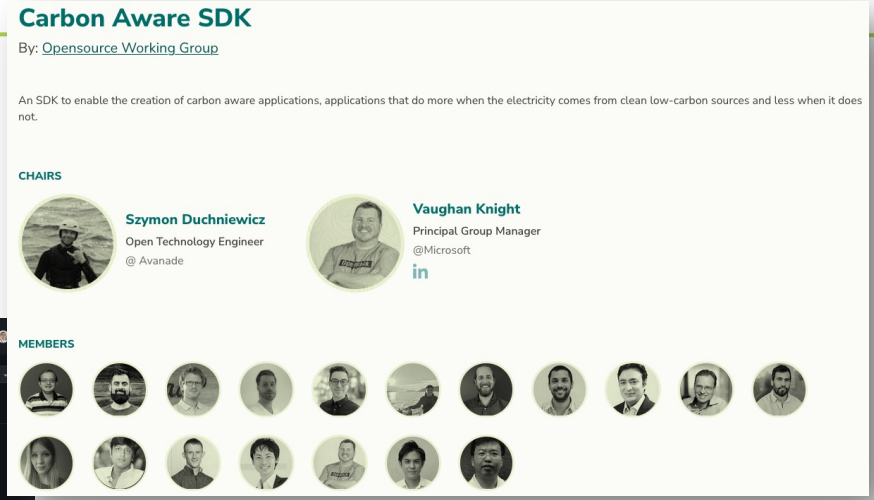
SCAN ME

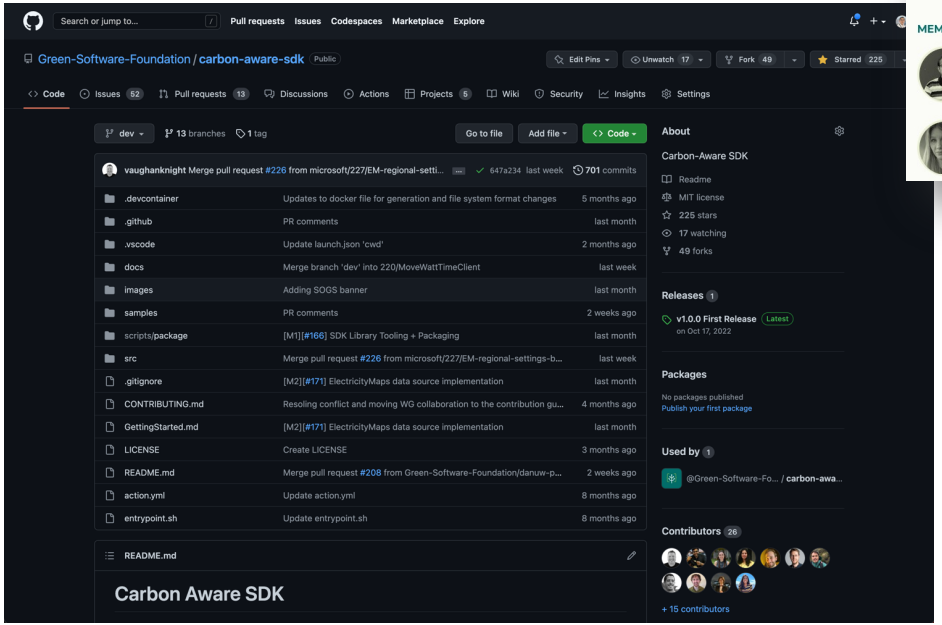
LINUX FOUNDATION



What do we need?

You!





Thank you to all our contributors

Learn more at

greensoftware.foundation

LF ENERGY SUMMIT 2023
Deep Decarbonization

Thank you for listening

Any questions?