Carbon Emission on Software Application

Good Morning!!!!

Team Members

- Yogesh Gandhi
- Paras Soni
- Abhay Jajodia

Purpose statement

What is carbon Emission?

Carbon emissions refer to the release of carbon compounds, primarily carbon dioxide (CO2) and methane (CH4), into the atmosphere.

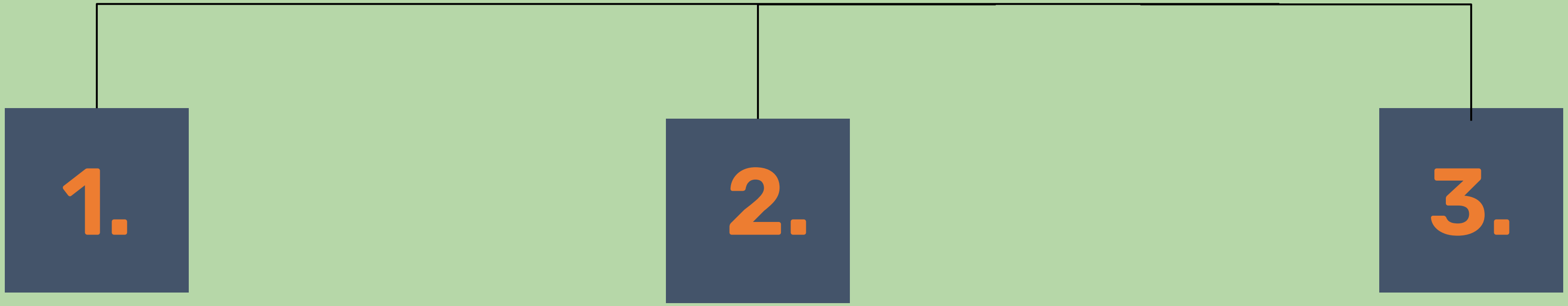
Effects of Carbon Emission?

Carbon emissions, primarily in the form of carbon dioxide (CO2), have profound effects on the environment and human health. Some of the example are like Human health, Climate change, Sea level Rise and so on

What about the human demands?

Software applications contribute to carbon emissions primarily through energy consumption during development, deployment, and usage.

The primary drivers of carbon emissions in software applications.



1.

Data Center Operations

Energy consumption in data centers, often powered by fossil fuels, contributes significantly to emissions.

2.

Inefficient Coding Practices

Poorly optimized code can lead to excessive resource usage, increasing energy demands during operation.

3.

User Behaviours

High-frequency usage and inefficient workflows can elevate energy consumption and carbon output.

Three actions that can be taken to reduce software carbon emissions.

Action	Description	Example
Optimize Code Efficiency	Write clean and optimized code to reduce resource consumption during execution.	Refactoring code to eliminate performance bottlenecks, such as reducing unnecessary loops or using more efficient algorithms like binary search instead of linear search.
Utilize Green Hosting Solutions	Choose cloud providers that prioritize renewable energy and have energy-efficient data centers.	Migrating applications to a cloud service like Google Cloud or AWS that commits to using renewable energy sources for their data centers.
Implement Performance Monitoring Tools	Use tools that analyze application performance to identify inefficiencies and improve energy usage.	Employing tools like New Relic or Datadog to monitor application performance and pinpoint areas where resource usage can be reduced.

Greenhouse Gas Protocol (GHG)

The Greenhouse Gas (GHG) Protocol is a widely-used standard for measuring and managing emissions.

To measure software emissions using the GHG Protocol, organizations should:

- Identify Emission Sources: Determine where emissions are generated within the software lifecycle, including data centers and user interactions.
- Collect Data: Gather data on energy consumption, operational practices, and usage patterns.
- Calculate Emissions: Use emission factors to convert energy consumption into CO2 equivalents (CO2e).

Scope	**Description**
Scope 1	Direct emissions from owned or controlled sources (e.g., fuel combustion).
Scope 2	Indirect emissions from the generation of purchased electricity, steam, heating, and cooling.
Scope 3	All other indirect emissions that occur in a company’s value chain (e.g., emissions from suppliers and product use).

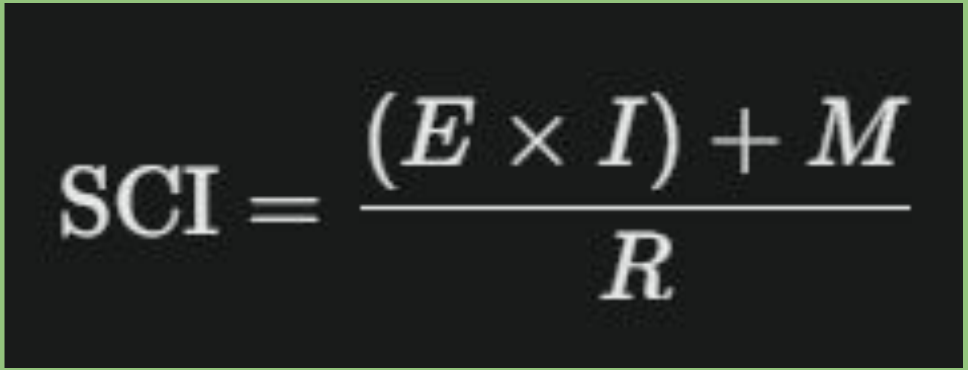
Software Carbon Intensity

SCI is a metric for measuring the environmental impact of software applications in terms of carbon emissions per unit of computational work (e.g., per transaction or per hour of use).

Steps to Measure SCI

1. Define Software Boundary: Identify which components of the software system will be included in the measurement.
2. Select Functional Unit: Choose a functional unit that best describes how the application scales (e.g., per user or transaction).
3. Quantify Emissions: Measure or calculate energy consumption and associated emissions for each component.
4. Report Results: Disclose the SCI score along with the methodology used for calculation.

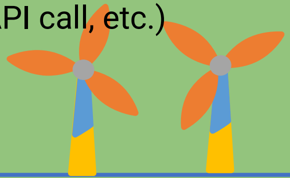
It uses a simple formula to quantify emissions based on energy consumption and other factors:



$$\mathrm{SCI} = \frac{(E \times I) + M}{R}$$

Where:

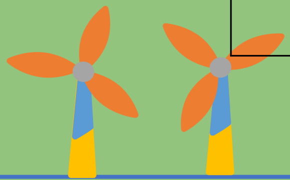
- E = Energy consumed by the software system
- I = Location-based marginal carbon emissions
- M = Embodied emissions of the software system
- R = Functional unit (e.g., carbon per user, API call, etc.)



Different types of corporate climate targets

Corporate climate targets are essential for organizations aiming to reduce their greenhouse gas (GHG) emissions and contribute to global climate goals.

Target Type	Description
Carbon Neutral	Companies aim to balance their carbon emissions by reducing them and purchasing carbon credits to offset any remaining emissions. Typically focuses on CO2 emissions.
Net Zero	A more comprehensive approach that includes all greenhouse gases (GHGs) and requires companies to reduce emissions to as close to zero as possible, with any residual emissions offset through carbon credits.
Science-Based Targets (SBT)	Targets validated by the Science Based Targets Initiative (SBTi) that align with climate science, aiming for significant reductions (e.g., 50-60%) by 2030 to limit global warming to 1.5°C.
Real-Zero Targets	Commitments to completely eliminate all GHG emissions from operations and value chains without relying on carbon credits, currently rare due to technological challenges.
Carbon-Negative Targets	Companies not only aim for net-zero emissions but also seek to remove more GHGs from the atmosphere than they emit, effectively reducing atmospheric concentrations of CO2.



How Green Software Proposals Can Support These Commitments

Green software proposals can play a crucial role in helping organizations meet their climate targets by providing methodologies, tools, and practices that enhance sustainability in software development and usage.

1. Optimizing Resource Use: By developing energy-efficient software applications, companies can reduce the energy consumption associated with their operations, directly impacting Scope 2 emissions.
   - Example: Implementing efficient algorithms or optimizing code can lead to lower server loads and reduced energy usage in data centers.
2. Promoting Renewable Energy Use: Green software initiatives can advocate for the use of renewable energy sources in data centers and cloud services, aligning with net-zero and carbon-neutral targets.
   - Example: Partnering with cloud providers that utilize 100% renewable energy for their data centers helps organizations achieve their sustainability goals.
3. Measuring and Reporting Emissions: Tools and frameworks that help quantify software-related carbon emissions enable organizations to track progress towards their climate targets effectively.
   - Example: Utilizing the Software Carbon Intensity (SCI) specification allows companies to measure the carbon footprint of their applications, providing insights into areas for improvement.