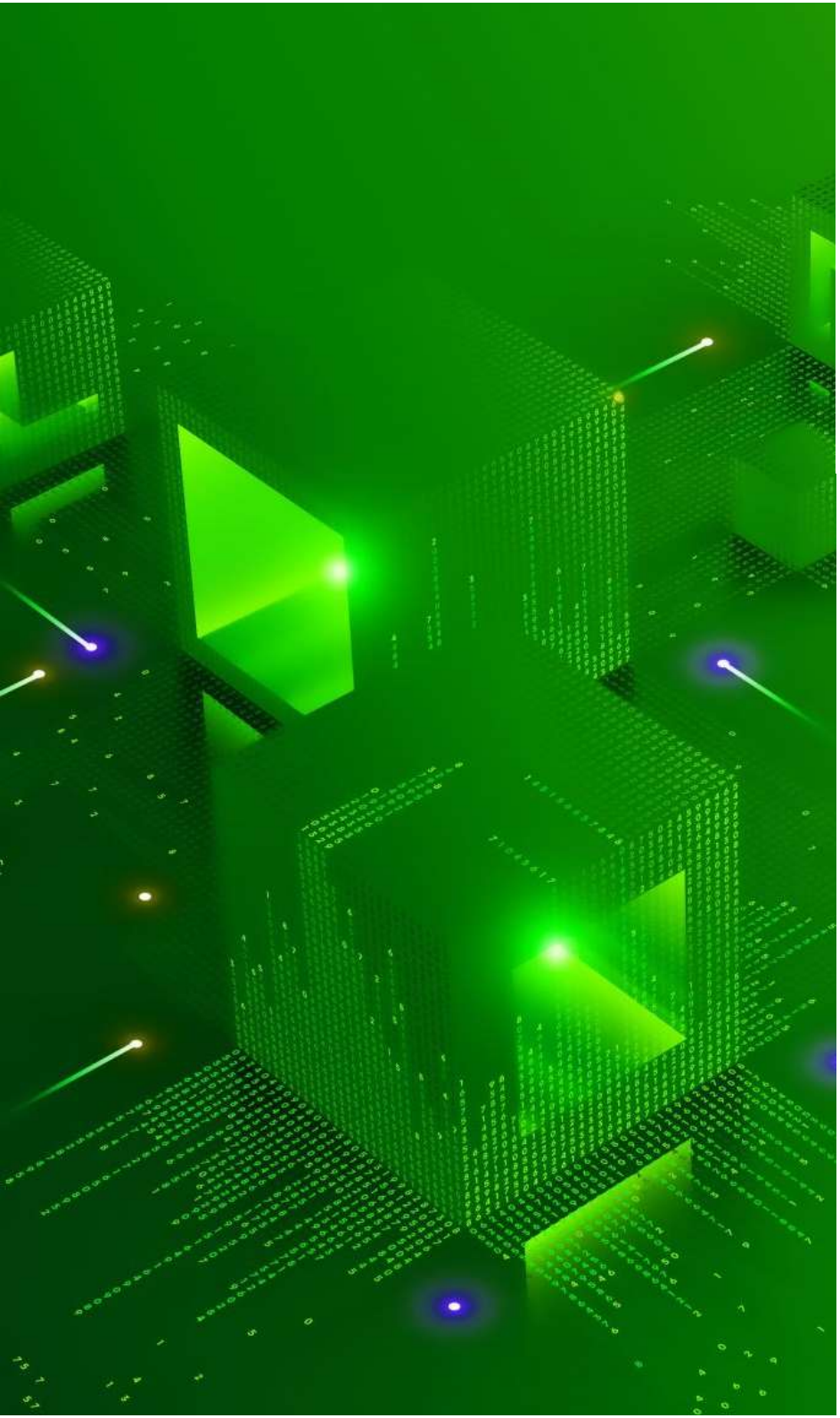
Rekayasa PL Hijau: Peluang dan Tantangan

Prof. Daniel Siahaan
Dept. Informatika – ITS
Institut Teknologi Sepuluh Nopember

16 April 2025 – Seminar Nasional ....

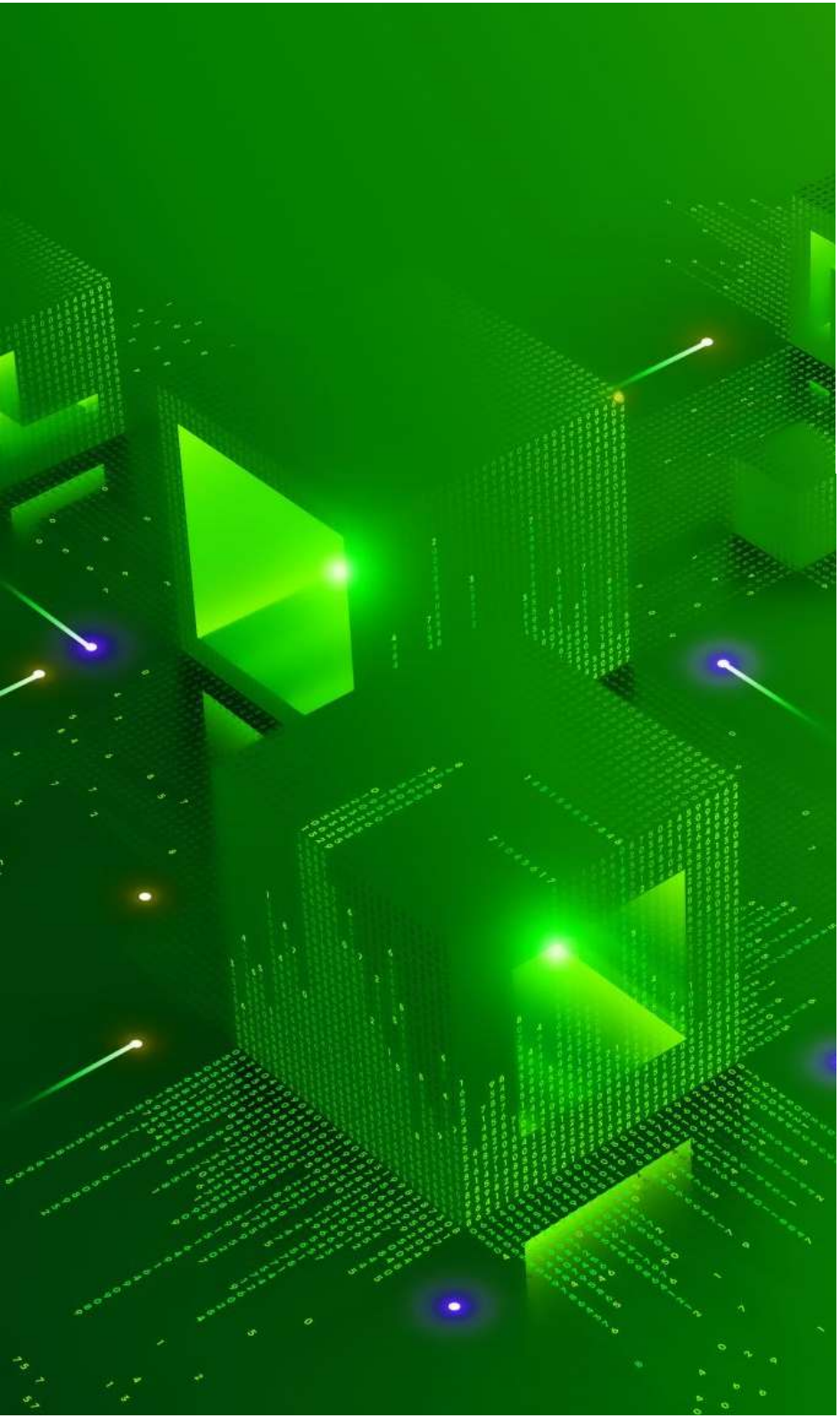


Green Software Engineering

What?

Green Software Engineering focuses on developing software that is energy-efficient and minimizes environmental impact.

It involves optimizing code, reducing energy consumption, and using sustainable computing resources.



Green Software Engineering

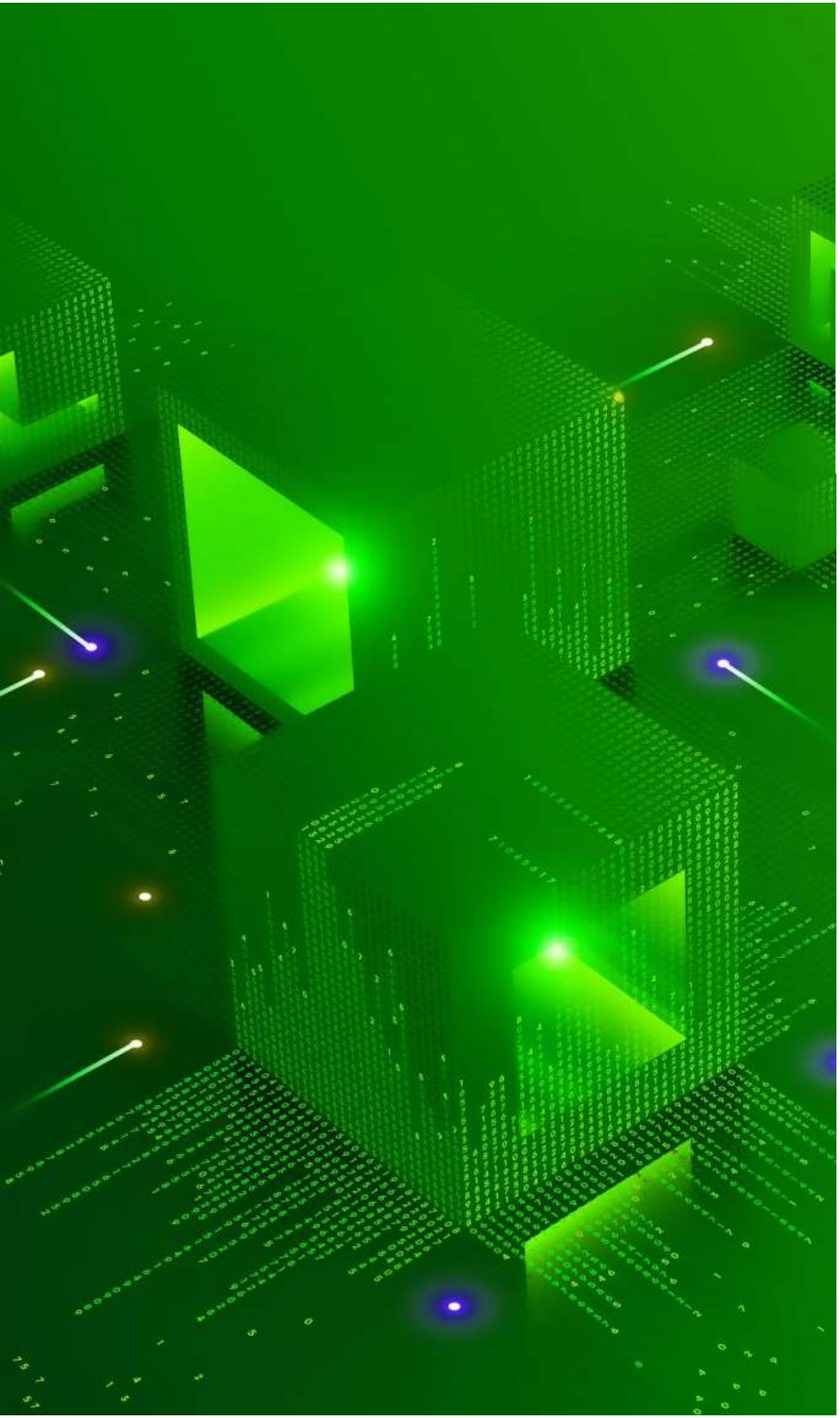
What?

Green Software Engineering focuses on developing software that is **energy-efficient** and **minimizes environmental impact**.

How?

It involves optimizing code, reducing energy consumption, and using sustainable computing resources.

Green Software Engineering



Green Software Engineering

What?

Green Software Engineering focuses on developing software that is energy-efficient and minimizes environmental impact.

How?

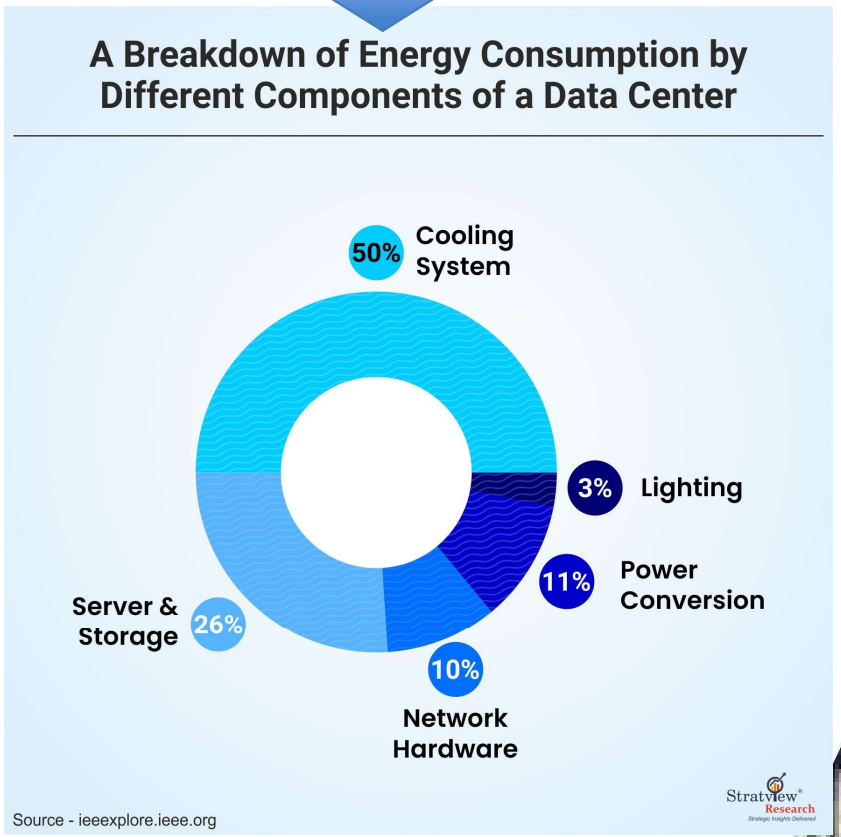
It involves optimizing code, reducing energy consumption, and using sustainable computing resources.

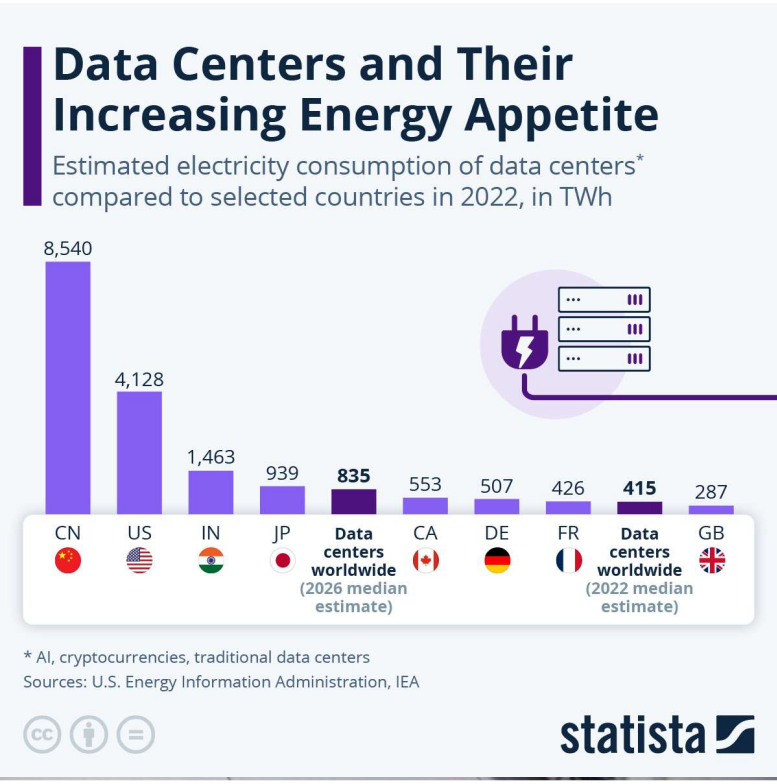
Apakah Industri Sw Berdampak kepada Lingungan

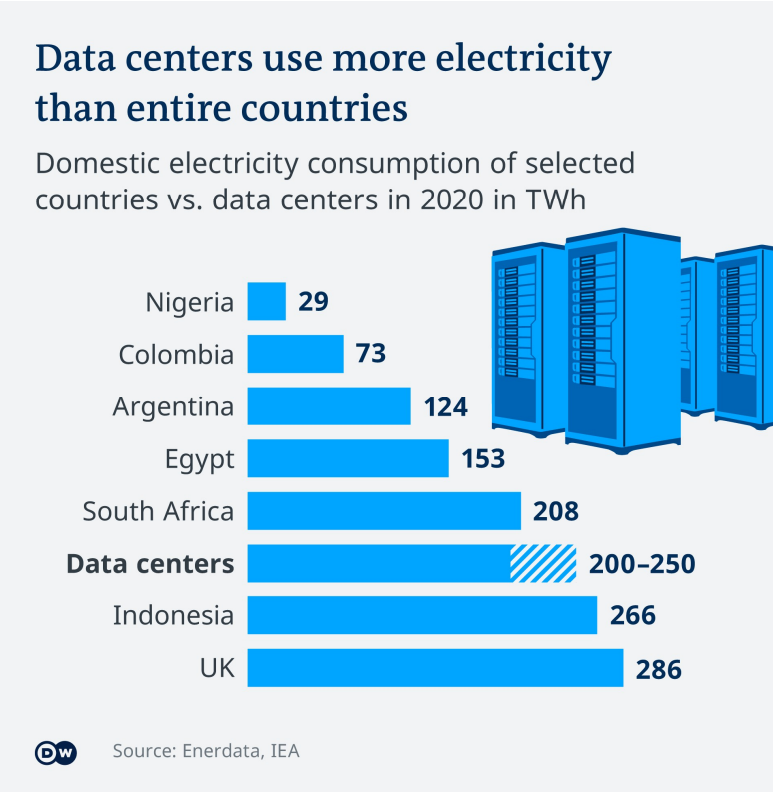
Colossal Data Center

Apakah Industri Sw Berdampak kepada Lingungan

Colossal Data Center







Apakah Industri Sw Berdampak kepada Lingungan

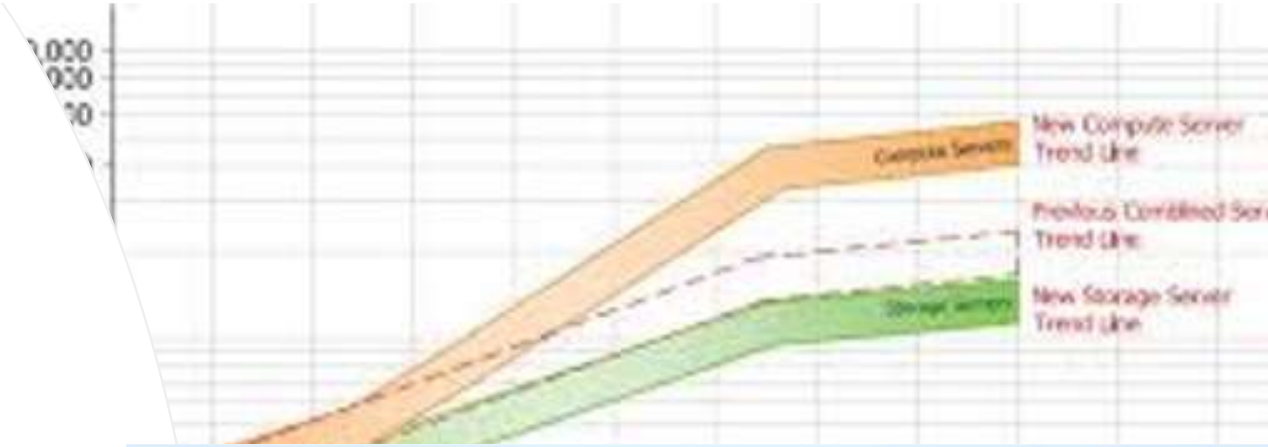
Colossal Data Center

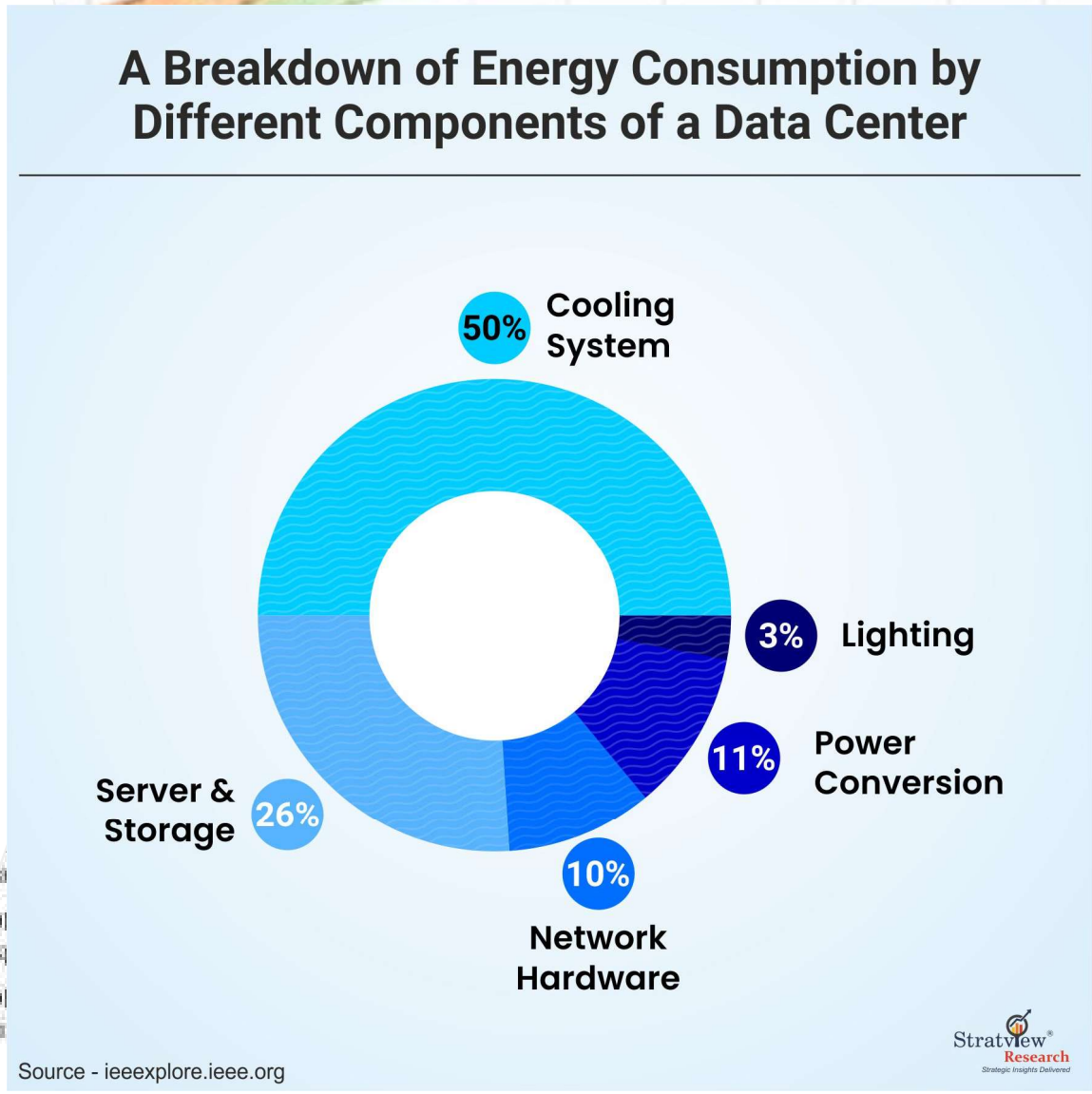
↓

Vast Electricity Consumption

↓

Generate Heat





Apakah Industri Sw Berdampak kepada Lingungan

Colossal Data Center

↓

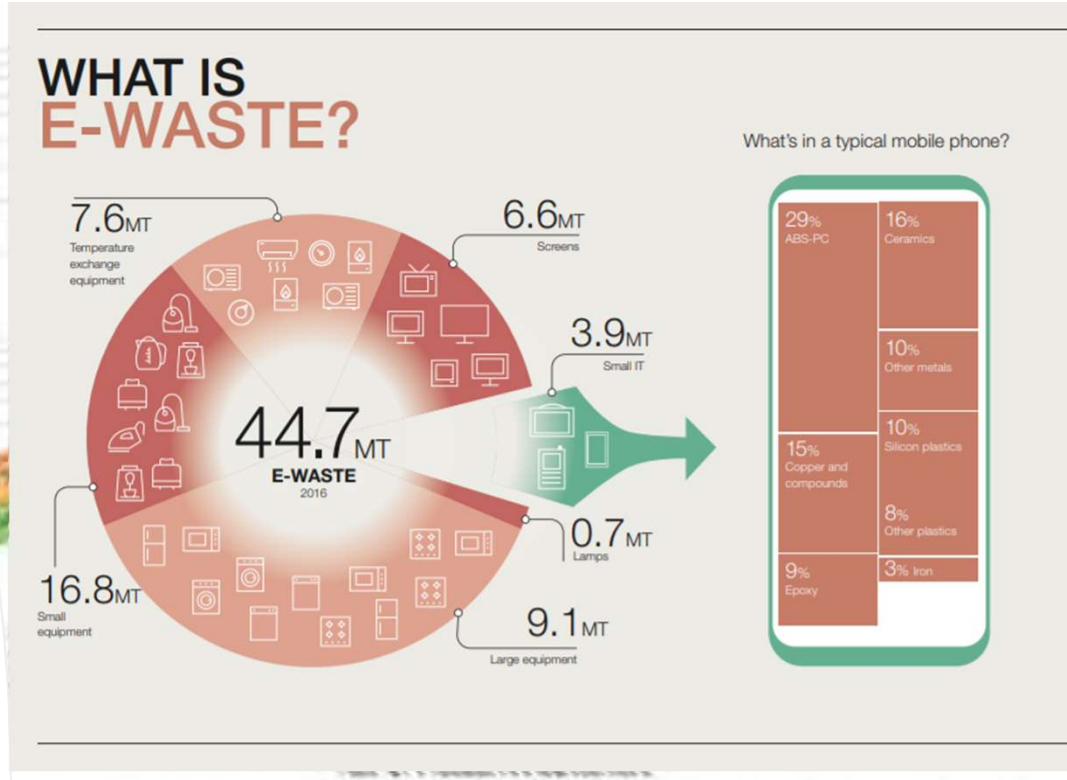
Vast Electricity Consumption

↓

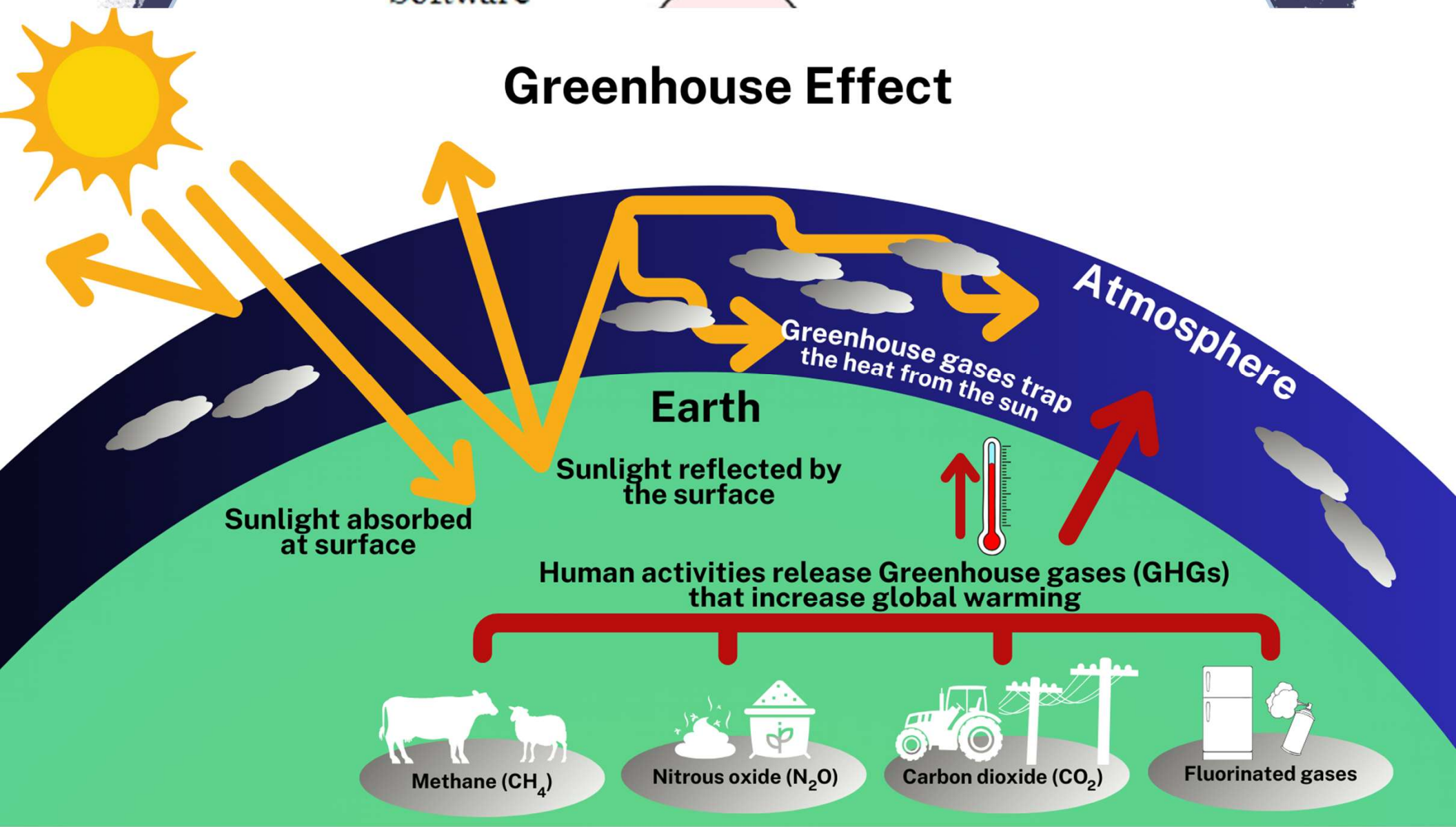
Generate Heat

↓

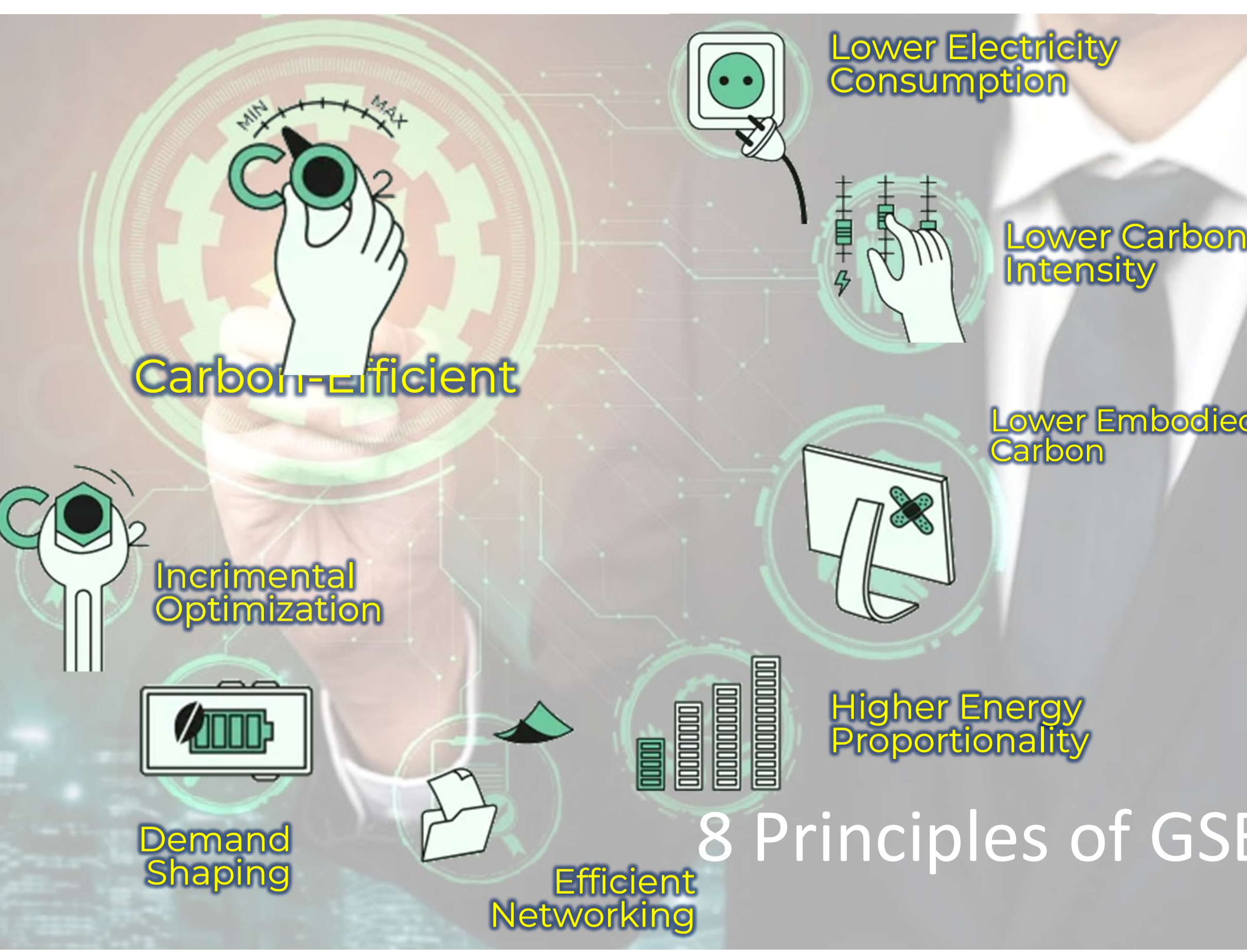
E-Waste



Sustainable
Software



8 Principles of GSE


Lower Electricity Consumption
MIN
MAX
Lower Carbon Intensity
Carbon-Efficient
Lower Embodied Carbon
Incrimental Optimization
Higher Energy Proportionality
Demand Shaping
Efficient Networking
8 Principles of GSE

Filosofi dari

Everyone Has a Role

Sustainability is the Key

GSE?

Mengapa GSE Penting?

Reduces carbon footprint of IT infrastructure

Enhances energy efficiency in data centers

Promotes sustainable computing practices

Helps organizations meet environmental regulations

Manfaat GSE?

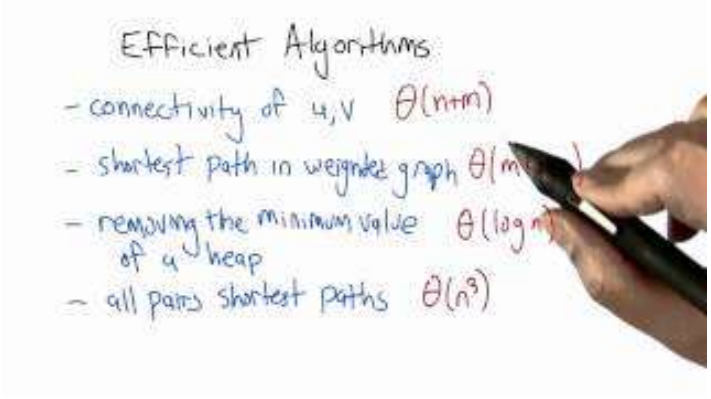
SIMPLER ARCHITECTURE

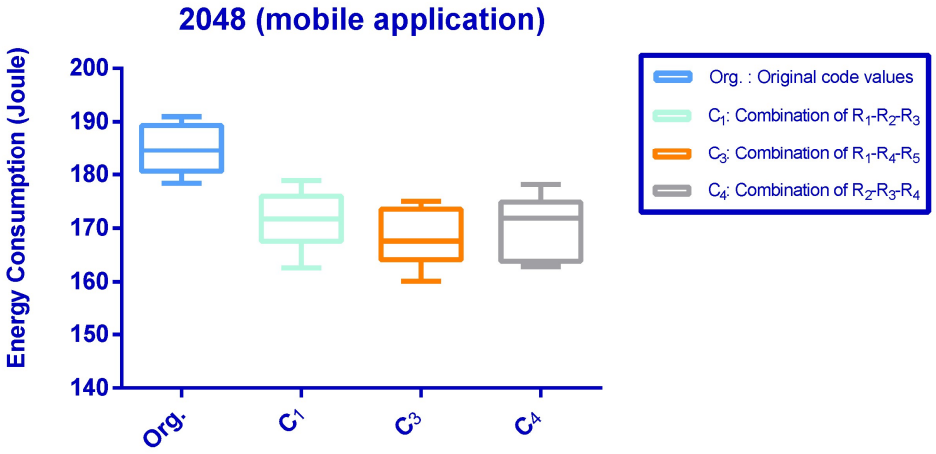
FASTER COMPUTING SPEED

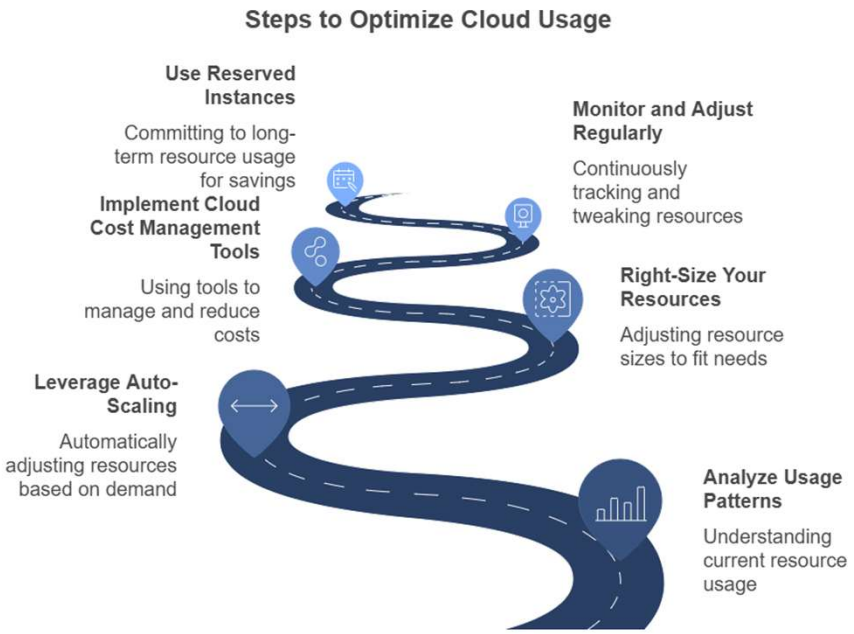
LOW RESOURCE USAGE

BRAND RESILIENCE

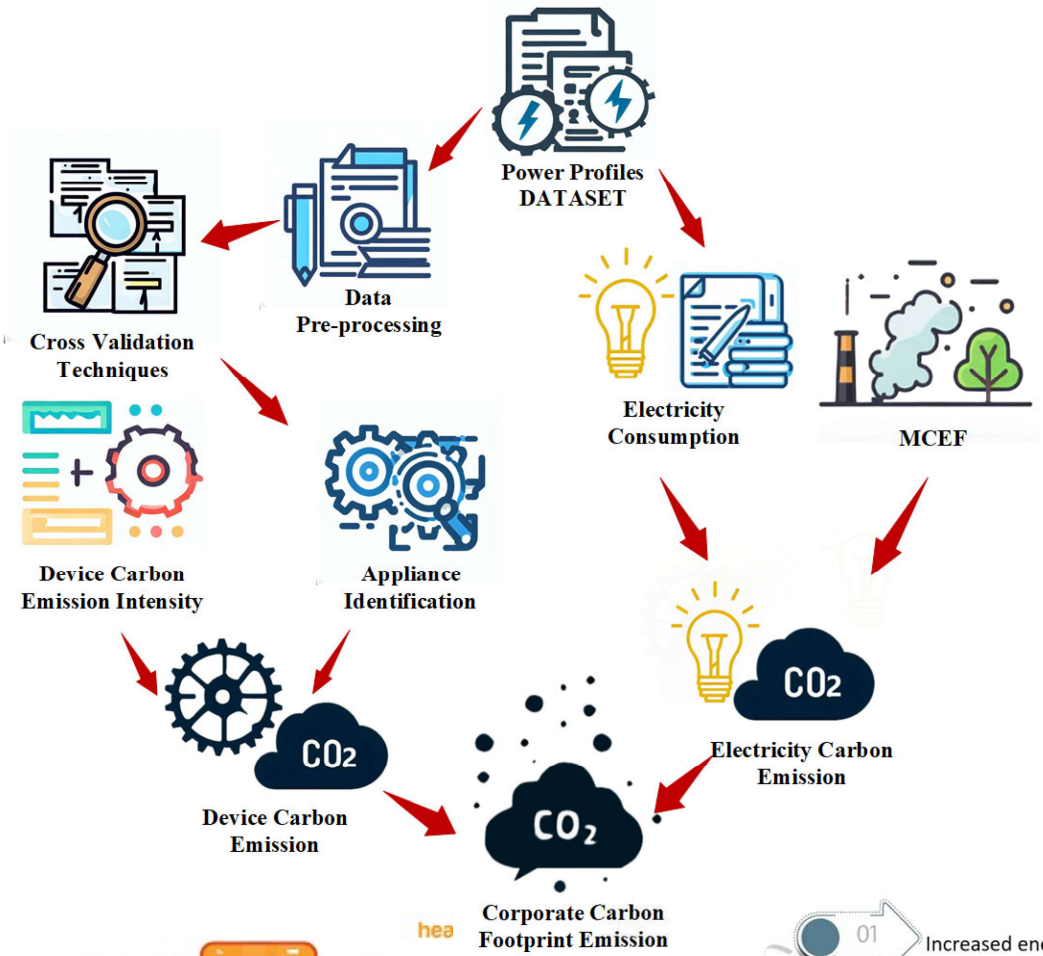
SUSTAINABLE CODING: BEST PRACTICES



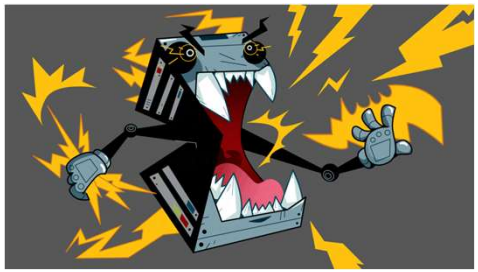




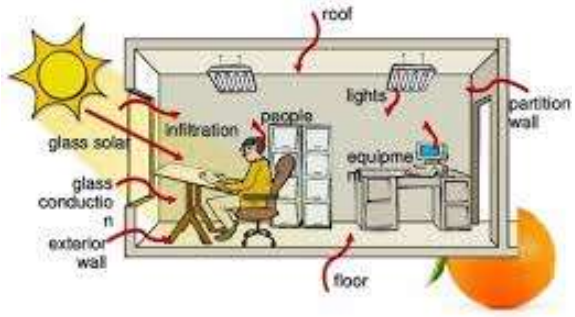
Coding & Carbon Footprints



Power Hungry Data Center

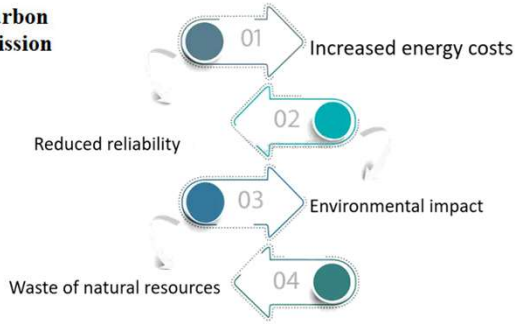


Cooling Requirements



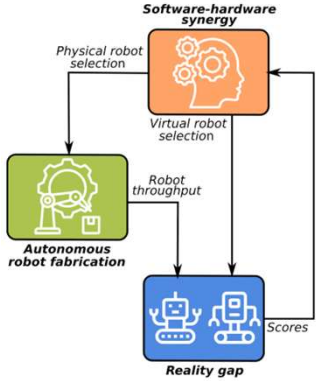
Scalability & Longevity

Optimized Mobile Apps

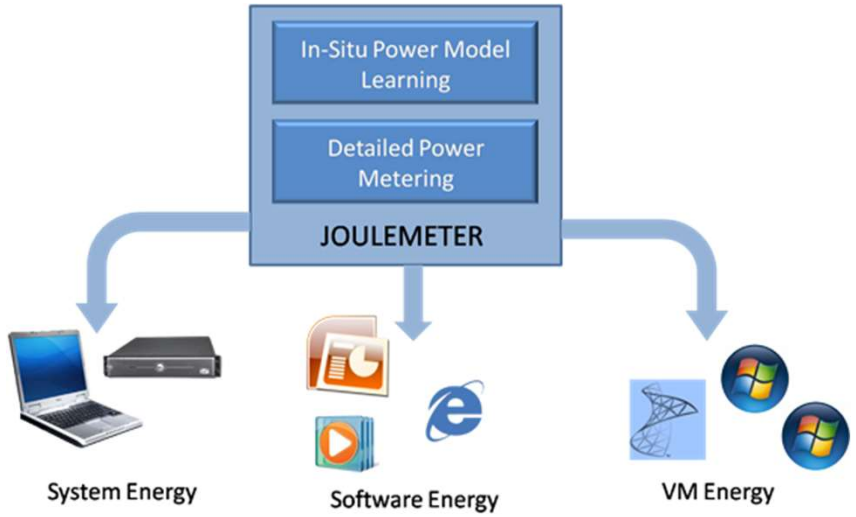


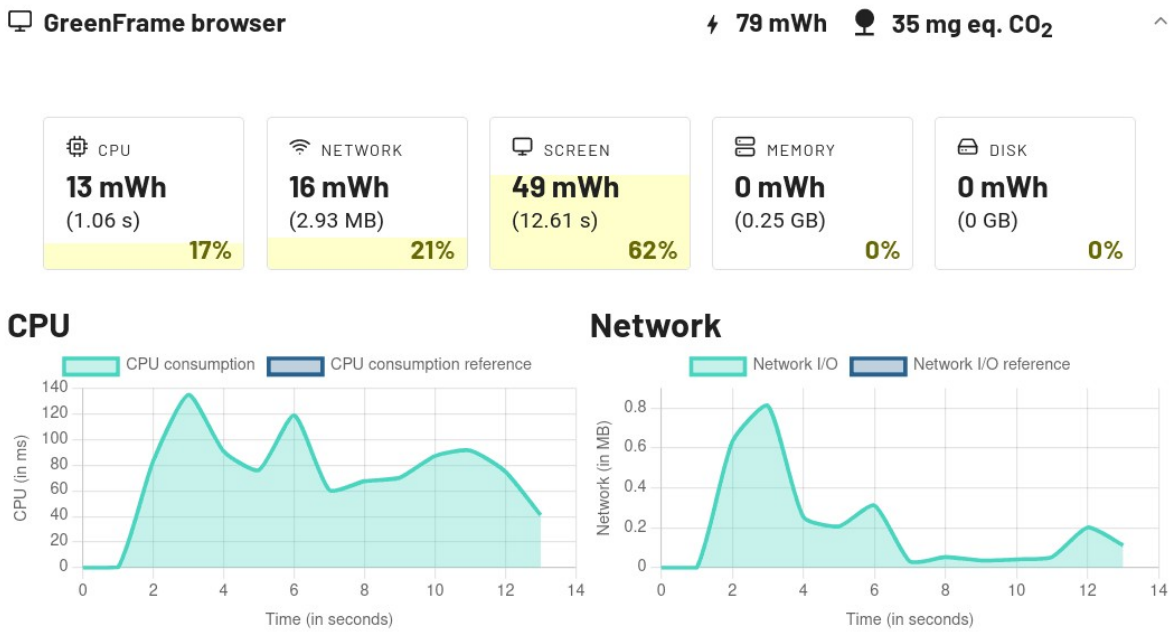
Reduction Transmission Energy

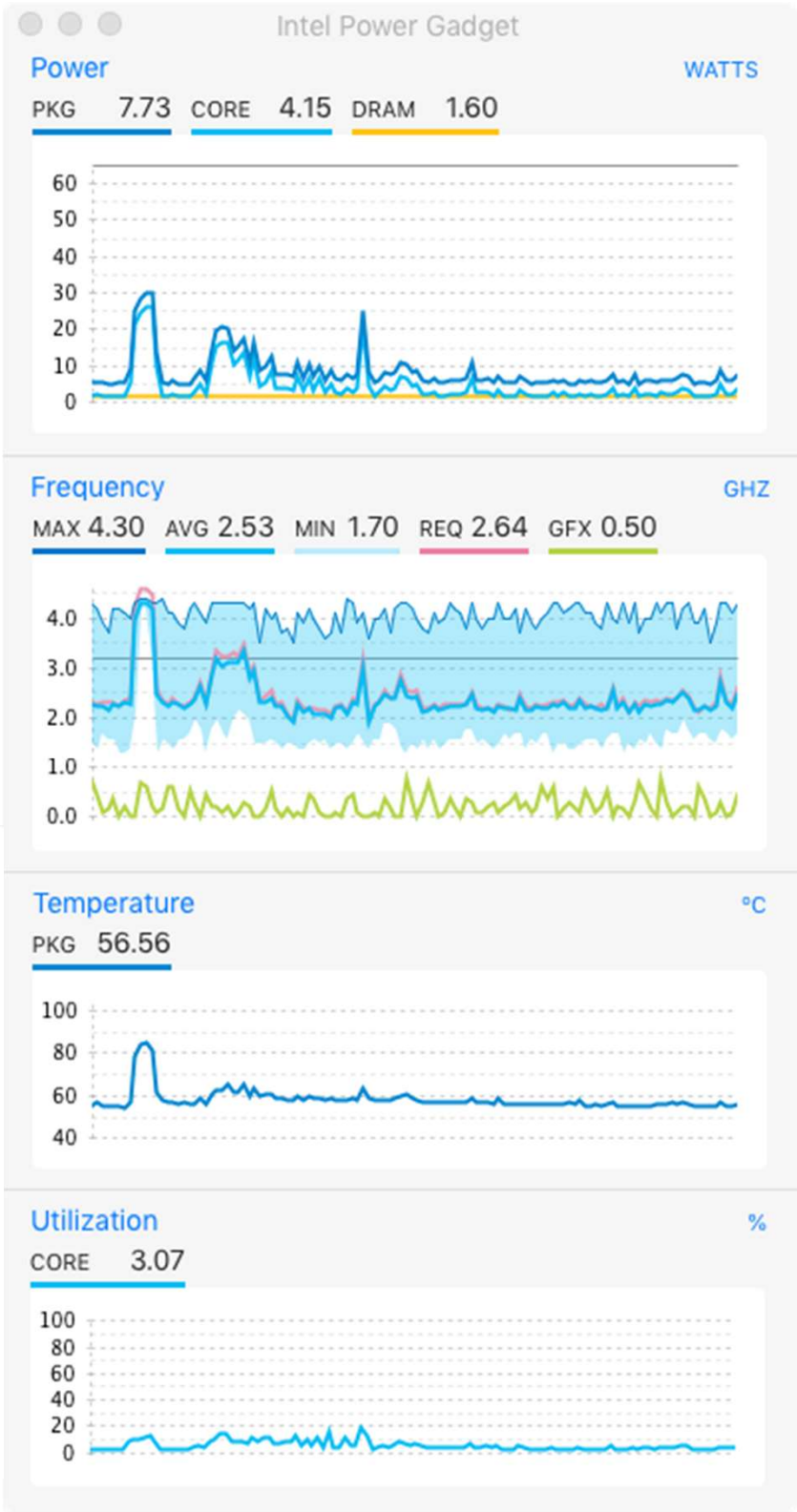
Sw – Hw Synergy



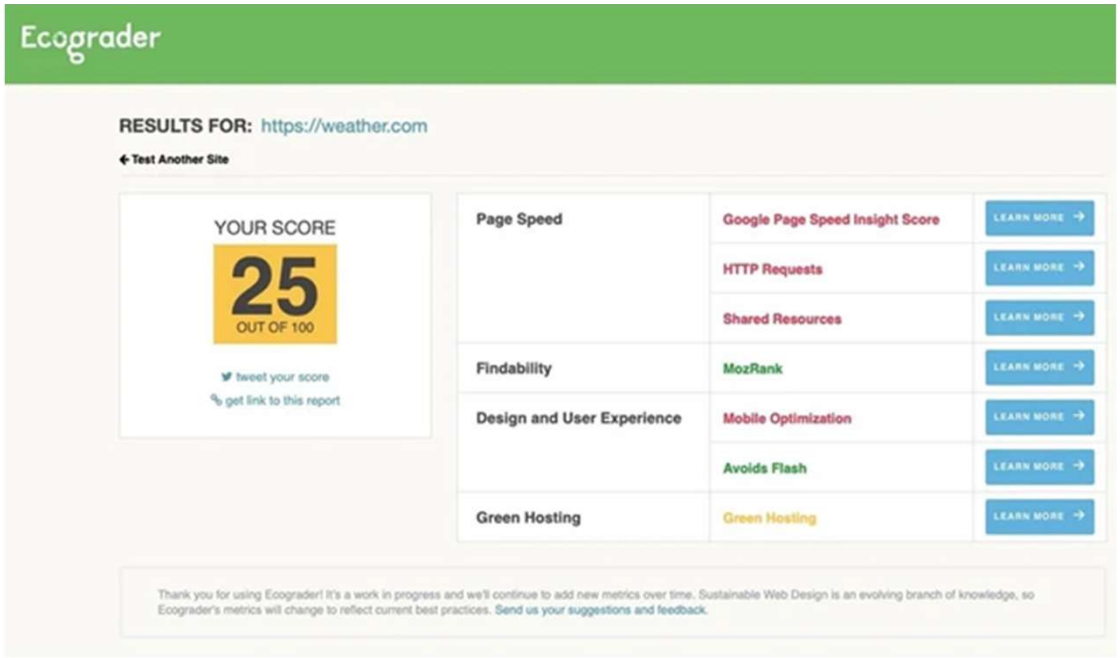
Tools for GSE



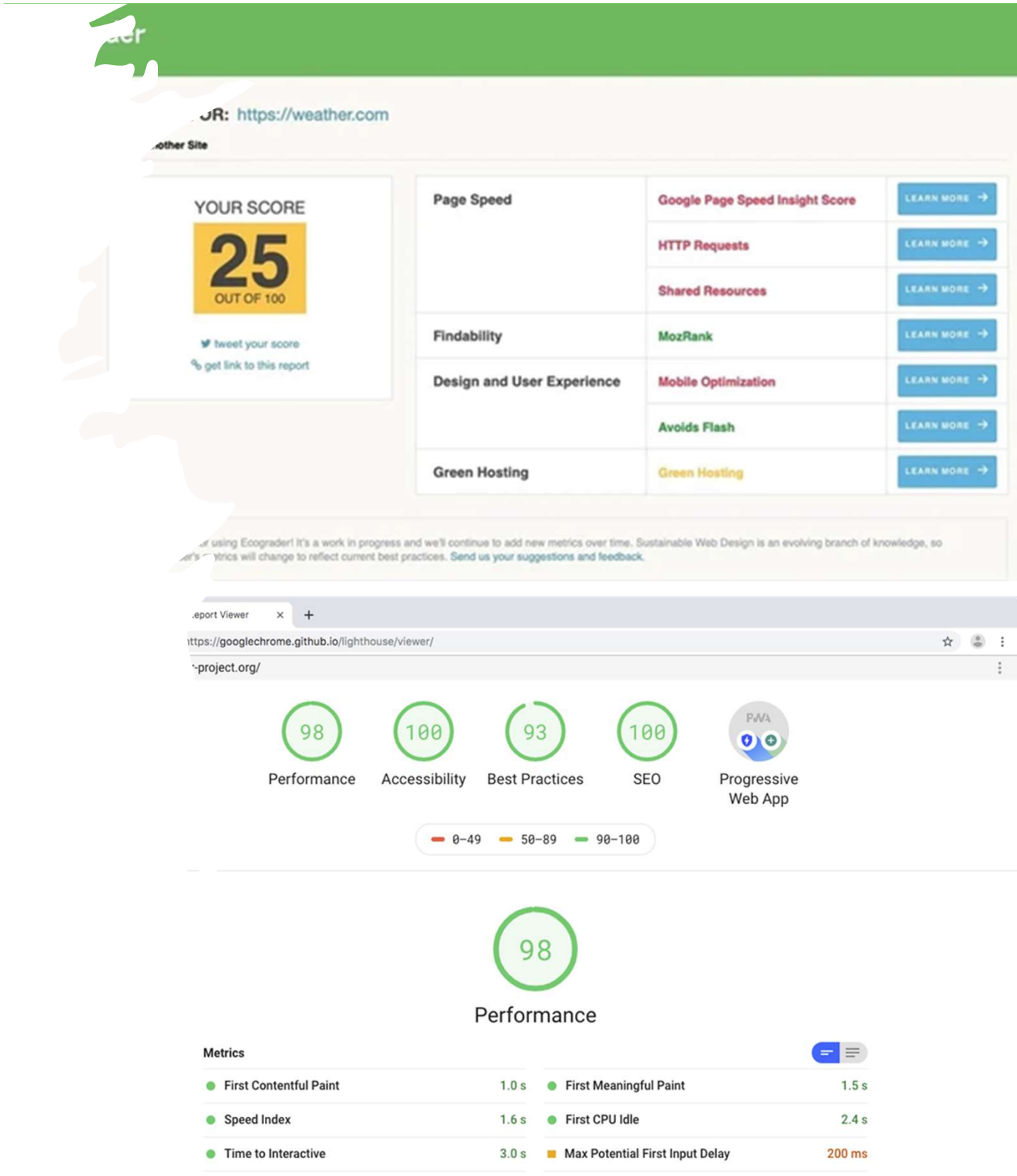




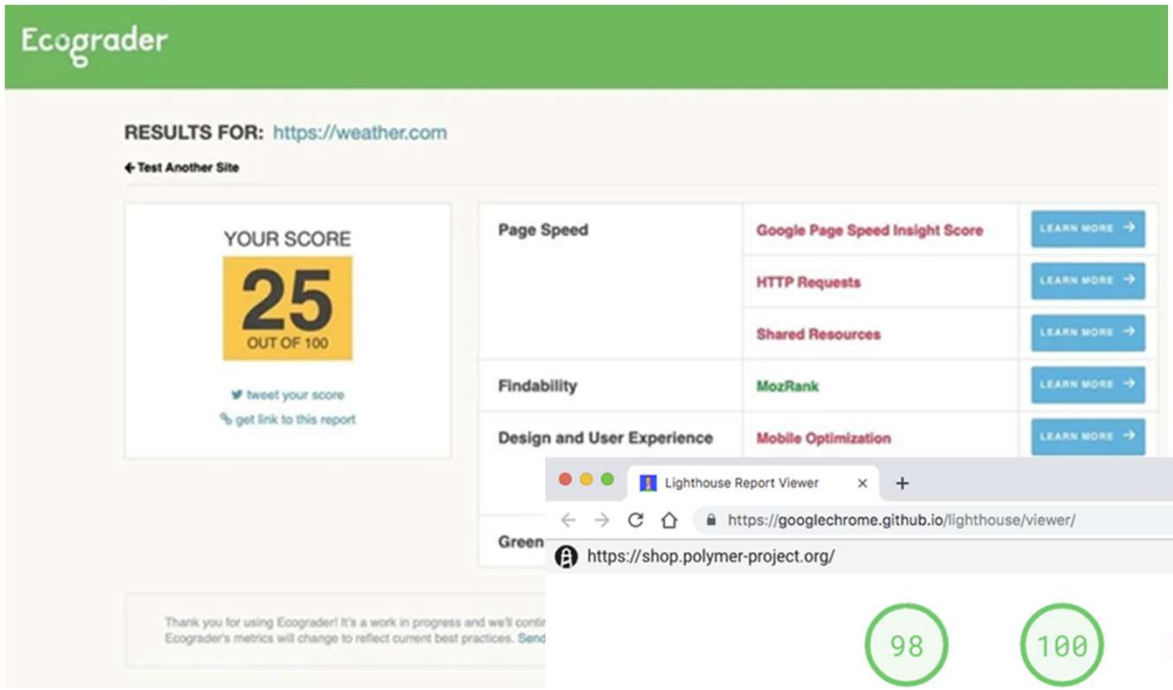
Tools for GSE

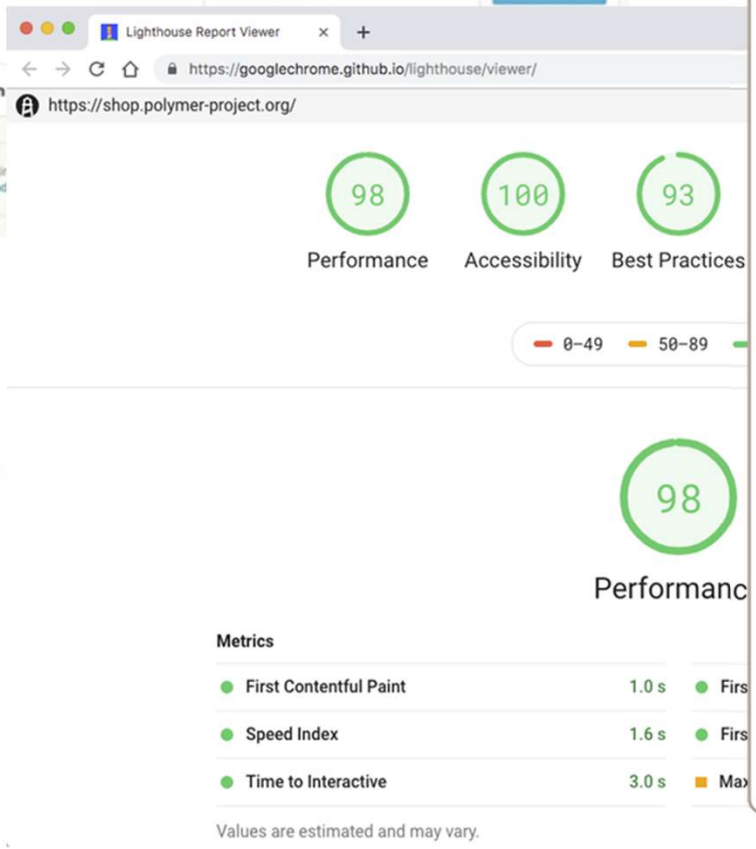


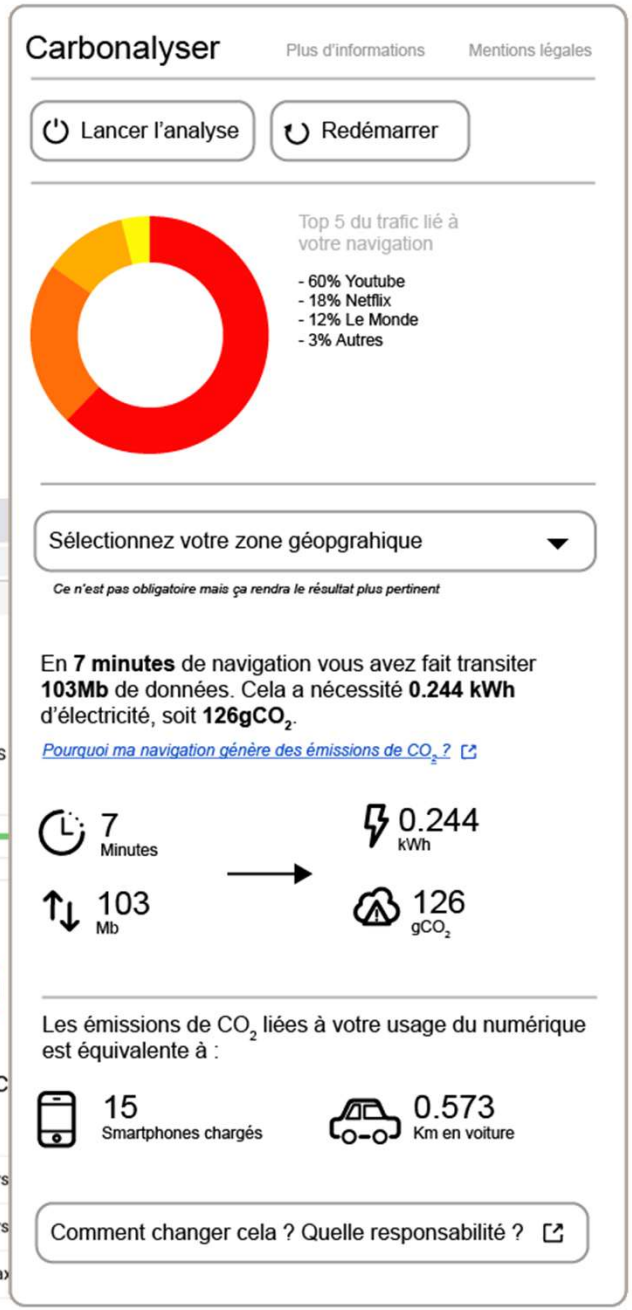
Tools for GSE



Tools for GSE



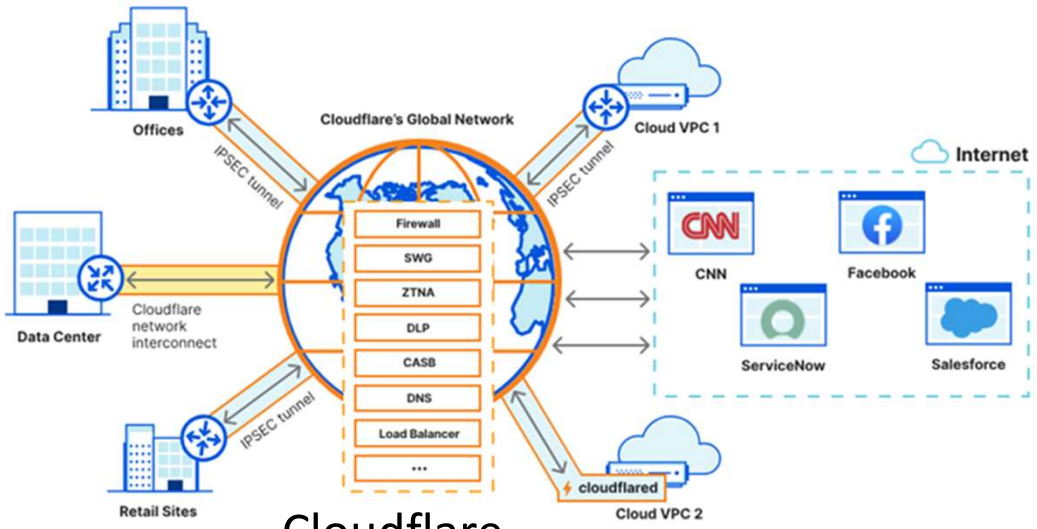




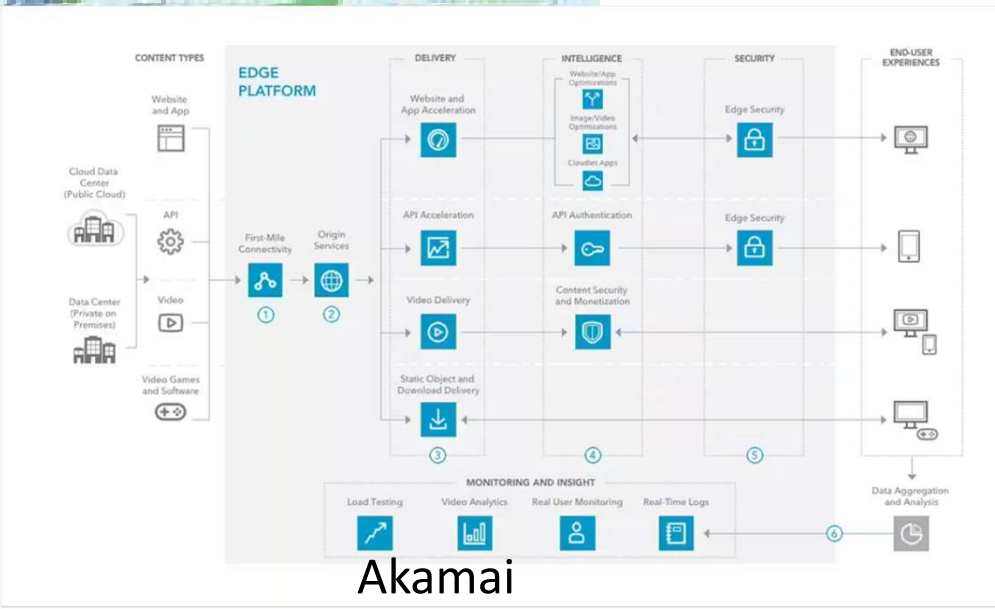
EcoMeter

Tools for GSE

Tools for GSE



Cloudflare



Akamai

Case Studies in Green Software

- Google’s energy-efficient data centers
- Microsoft’s carbon-negative initiatives
- Sustainable software development at IBM
- Green AI models reducing computation power

Industry Impact of Green Software

- IT industry’s role in reducing carbon emissions
- Green coding initiatives in major tech firms
- Policy and regulatory trends towards sustainability
- Future of eco-friendly software development

CLOUD COMPUTING AND GREEN CODING

- **Efficiency at Its Core**
- **Reduced Carbon Footprint**
- **Scale with Care**
- **Less Waste**

ROLE OF AI IN GREEN CODING

OPTIMIZATION ALGORITHMS

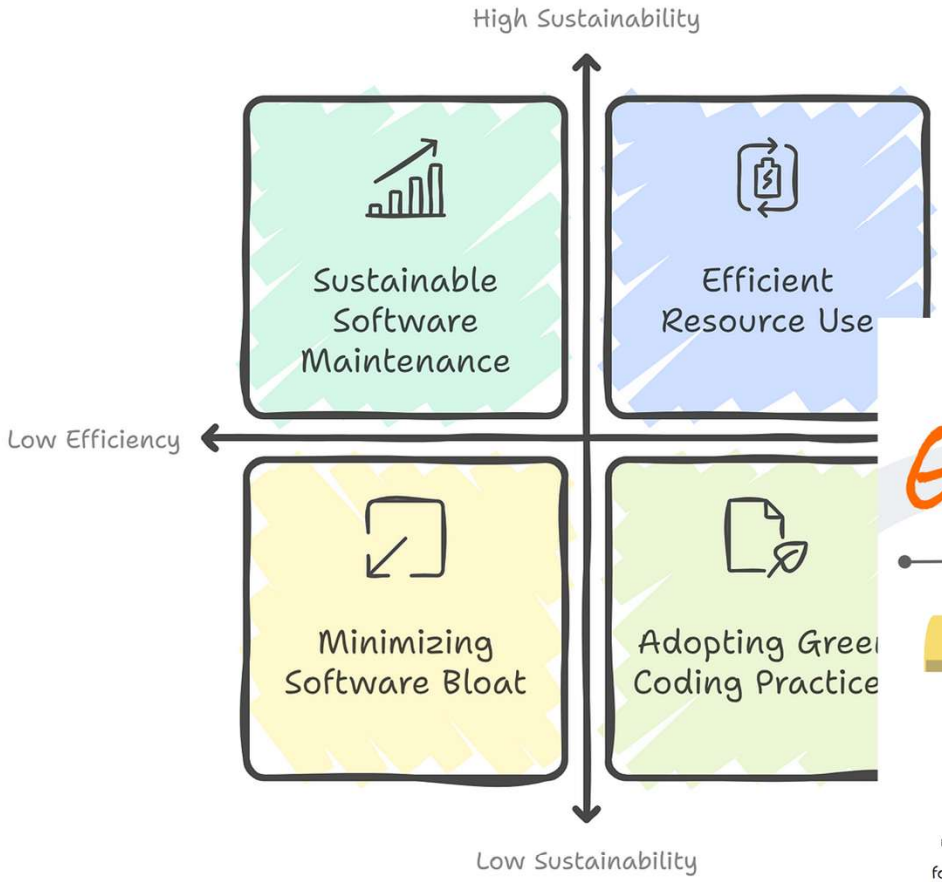
PREDICTIVE ANALYTICS

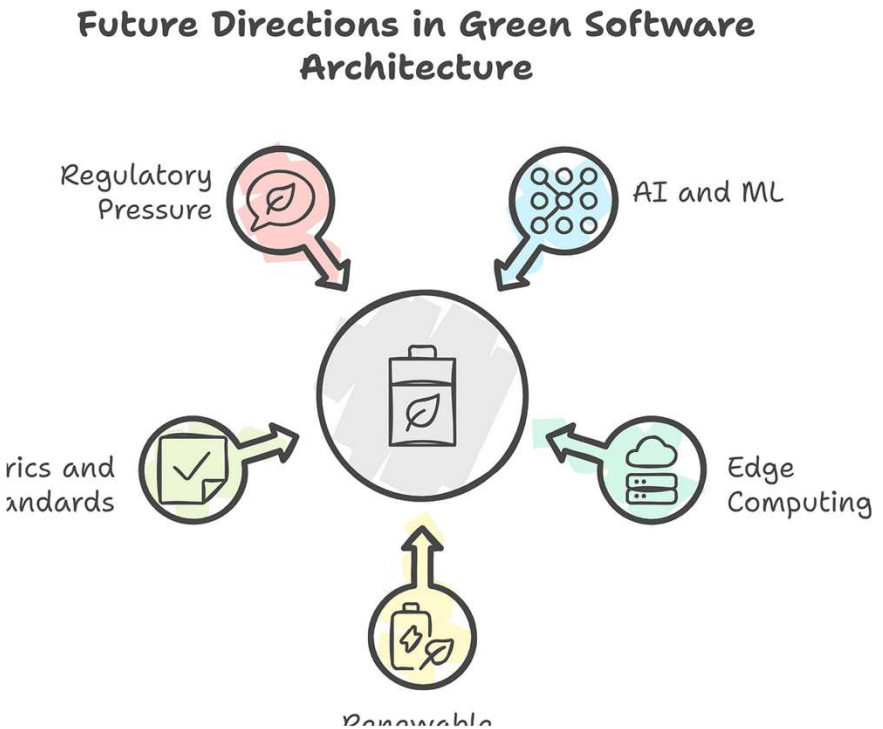
AUTOMATED GREEN TESTING

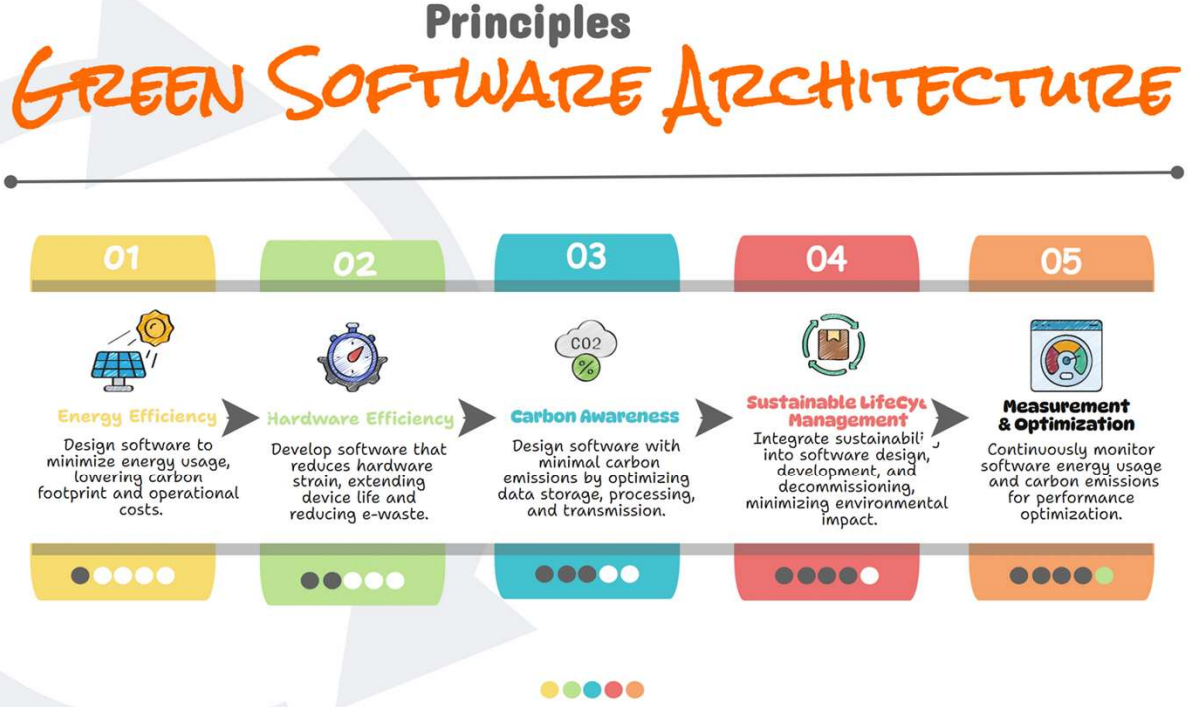
DYNAMIC RESOURCE ALLOCATION

TAILORED LEARNING RESOURCES

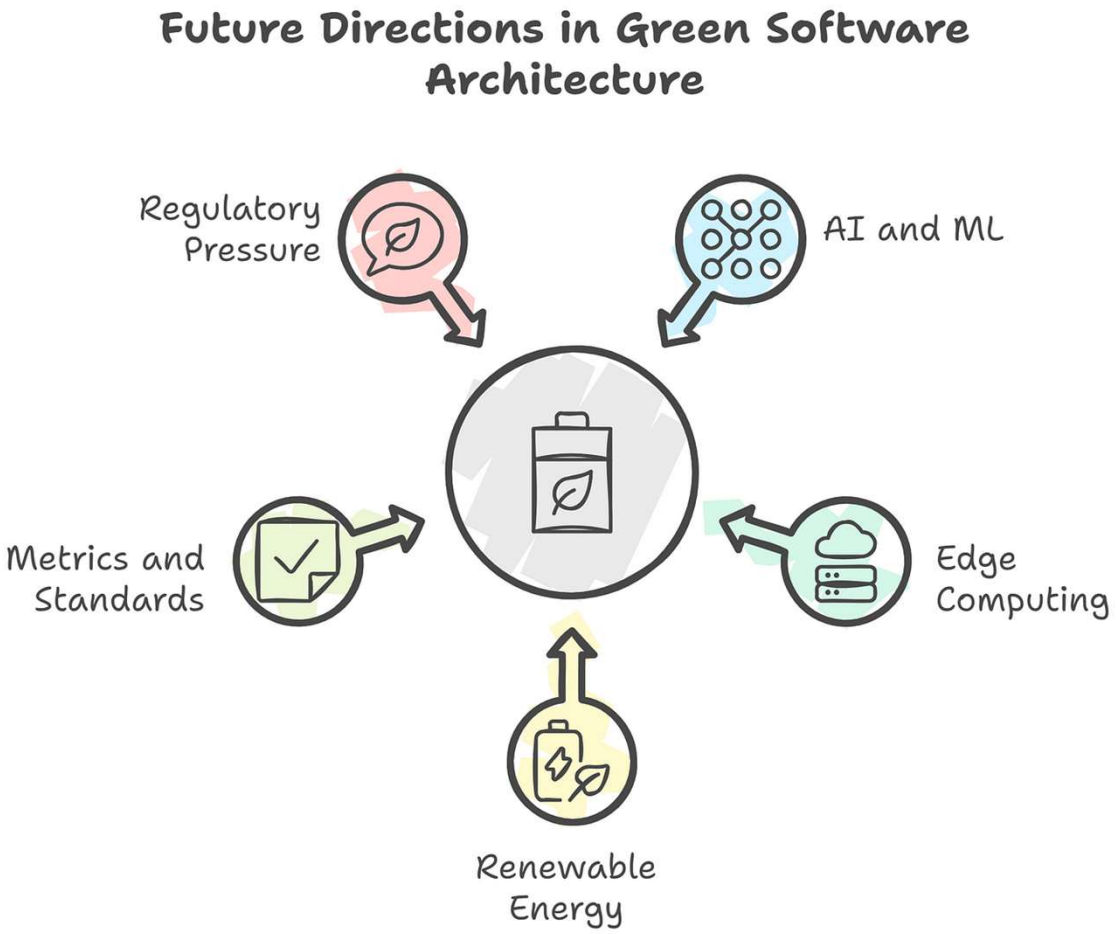
What Future Holds







What Future Holds



Green Coding

Algoritma dan Struktur Data Yang Efisien

Linear Search

```
def count_elements_linear(arr, element):
    count = 0
    for i in range(len(arr)):
        if arr[i] == element:
            count += 1
    return count
```

Hash Table Search

```
def count_elements_hash_table(arr, element):
    count_table = {}
    for i in range(len(arr)):
        if arr[i] in count_table:
            count_table[arr[i]] += 1
        else:
            count_table[arr[i]] = 1
    if element in count_table:
        return count_table[element]
    else:
        return 0
```

Green Coding

Energy-Efficient Hardware

No-Memoization

```
def fibonacci_recursive(n):
    if n < 2:
        return n
    return fibonacci_recursive(n-1) +
fibonacci_recursive(n-2)
```

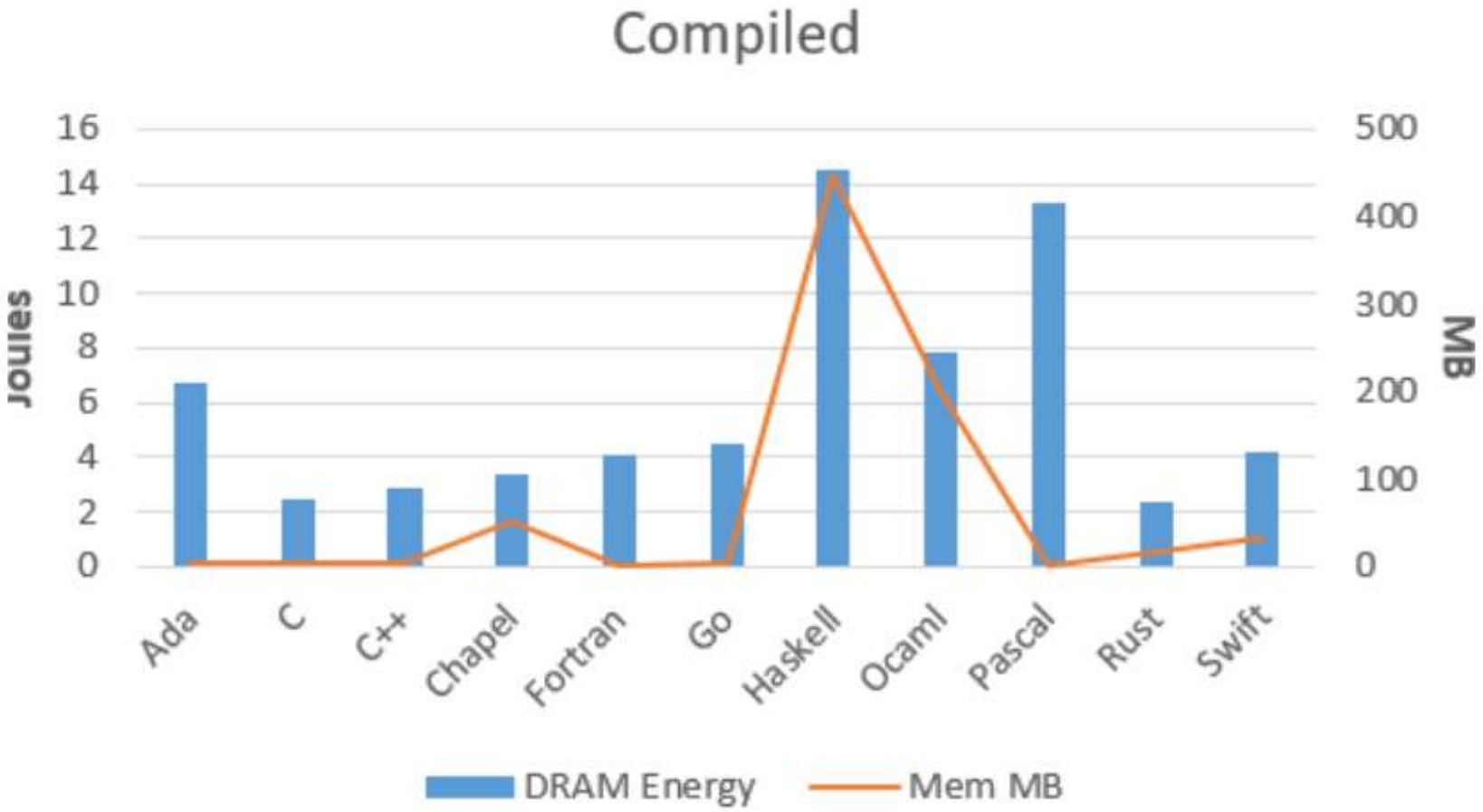
With Memoization

```
ef fibonacci_memoization(n, memo):
    if n in memo:
        return memo[n]
    if n < 2:
        memo[n] = n
    else:
        memo[n] = fibonacci_memoization(n-1, memo)
+ fibonacci_memoization(n-2, memo)
    return memo[n]


def fibonacci(n):
    memo = {}
    return fibonacci_memoization(n, memo)
```

Green Coding

Energy-Efficient Programming Language



Source: David Cassel (2018)

Green Coding

Energy-Efficient Programming Language

	Energy consumed	Run-time
C	57J	2019 ms
Rust	59J	2103 ms
C++	77J	3155 ms
Ada	98J	3740 ms
Java	114J	3821 ms

Language	*Memory space needed*
Pascal	66Mb
Go	69Mb
C	77Mb
Fortran	82Mb
C++	88Mb

Source: David Cassel (2018)

Green Coding

Energy-Efficient Programming Language

	Energy consumed	Run-time
Imperative	125J	5585ms
Object-Oriented	879J	32965ms
Functional	1367J	42740ms
Scripting	2320J	88322 ms

Source: David Cassel (2018)