


**Your Software & AI**  
emits tonnes of CO<sub>2</sub>.  
**You** can **cut** it.  
*Want to know how?*



**Wilco Burggraaf**  He/Him

Sustainable Digital Architect & Transformation Lead | Low-Waste IT,  
Data & AI, From Code to Operations

**Mitigate.**  
**Innovate.**  
**Sustain.**

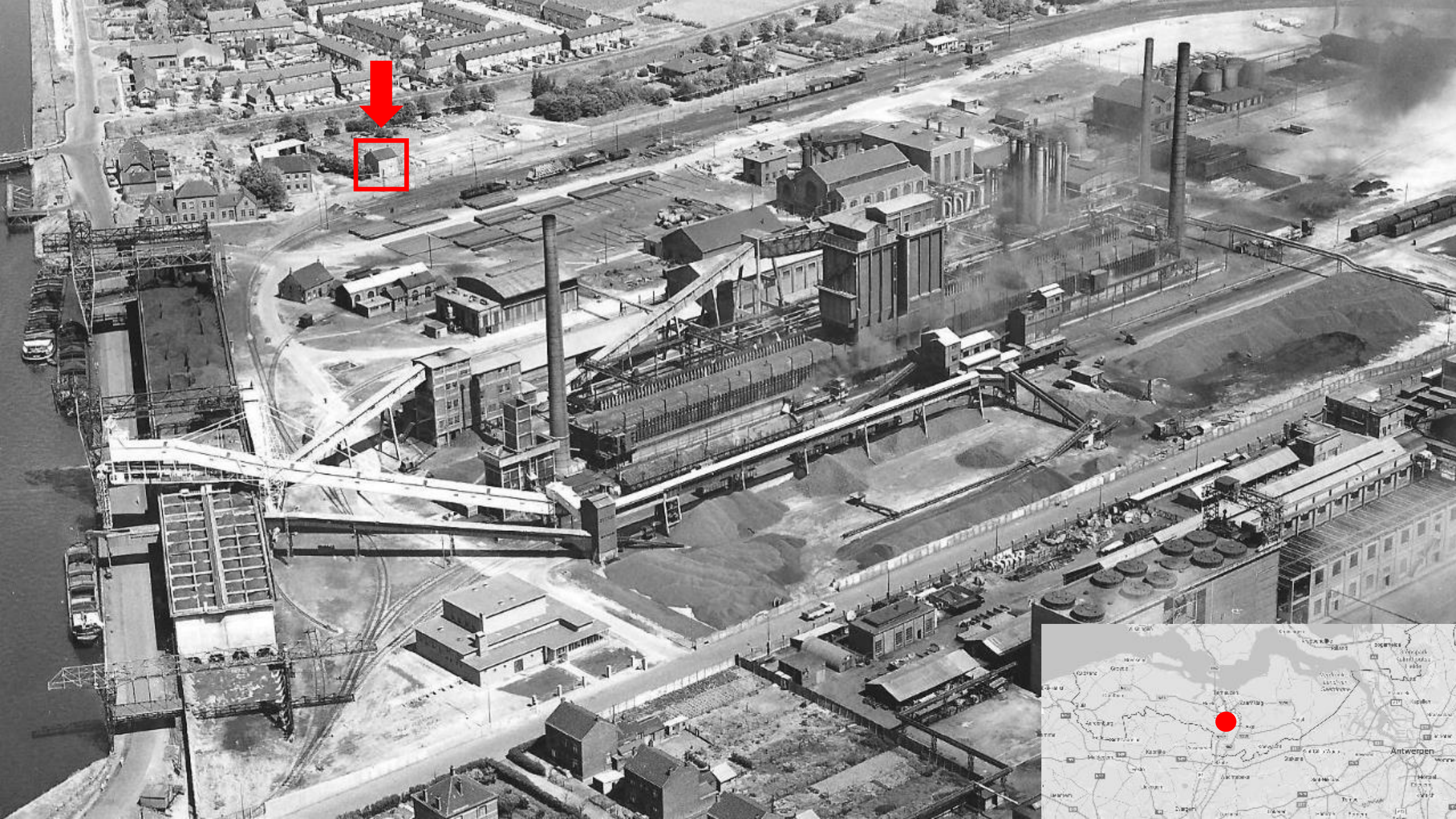


Sustainability Conference for



**SC4RC**

Responsible Research Computing





**HighTech Innovators**  
YOUR SOFTWARE. BETTER

**Digital waste**  
creates measurable  
environmental **harm**

**[Green]**

**Digital waste turns  
unnecessary energy demand  
into measurable emissions**

# [Sustainability]

Sustainable software  
keeps that harm from  
**compounding.**

Technical debt and  
digital waste are  
sustainability problems.

**WHY?**



**Climate  
Change  
Mitigation**



**Climate  
Change  
Adaptation**



**Sustainable Use  
and Protection of  
Water and  
Marine  
Resources**



**Transition to a  
Circular  
Economy**



**Pollution  
Prevention  
and Control**



**Protection and  
Restoration of  
Biodiversity  
and Ecosystems**

Because over 5–10 years,  
technical debt compounds into  
**avoidable** energy, hardware,  
maintenance, and emissions.

# IT Baseline – EU Taxonomy Structured Overview

EU Taxonomy Objective	ESRS Reference	IT Topic	Focus Area
1. Climate Change Mitigation	E1-1	IT climate transition plan	Infrastructure, cloud & development
	E1-2	IT mitigation policies	Infrastructure devices & cloud usage
	E1-3	IT mitigation actions & resources	Supporting climate goals
	E1-4	IT climate targets	Datacenters, cloud workloads, devices
	<b>E1-6</b>	<b>IT GHG emissions</b>	<b>Scope 1, 2, 3</b>
	E1-7	IT GHG removal participation	Carbon removal & mitigation initiatives
	2 GOV-3	Sustainability in incentives	Integration into IT leadership KPIs
	2 IRO-1	Climate risk process	Identification of climate risks
	E1-9	Financial impacts	Climate-related physical risks

Mitigate

When does digital activity  
become waste?

# **Underutilization**

CPU cores waste energy  
when idle, misused, or  
inefficiently loaded.

# **Waiting**

Loops keep cores  
fully active while  
doing no useful work.

# **Bottlenecks**

Slow code paths amplify CPU spikes, queue growth, timeouts, and wasted scaling.

If you continuously  
validate **workload quality**,  
most digital waste can be  
prevented before it  
becomes long-term  
**technical debt.**

[Green]

**Digital waste** turns  
unnecessary energy demand  
into **measurable emissions**

Electricity

Emissions

Total electricity mix

Apr 30, 2026, 9:00 PM GMT+2

↔ Synthetic



Carbon intensity

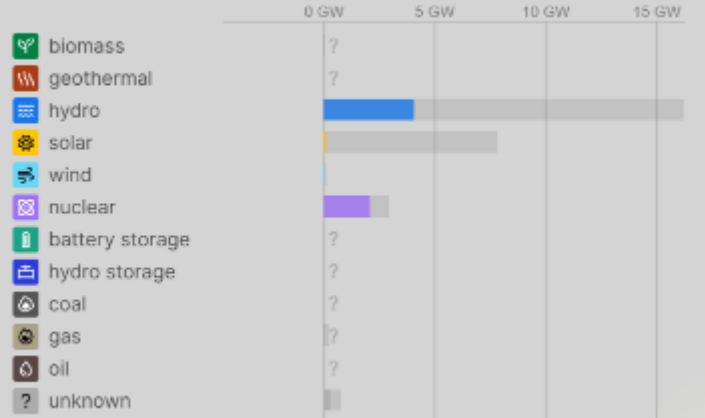


Carbon-free

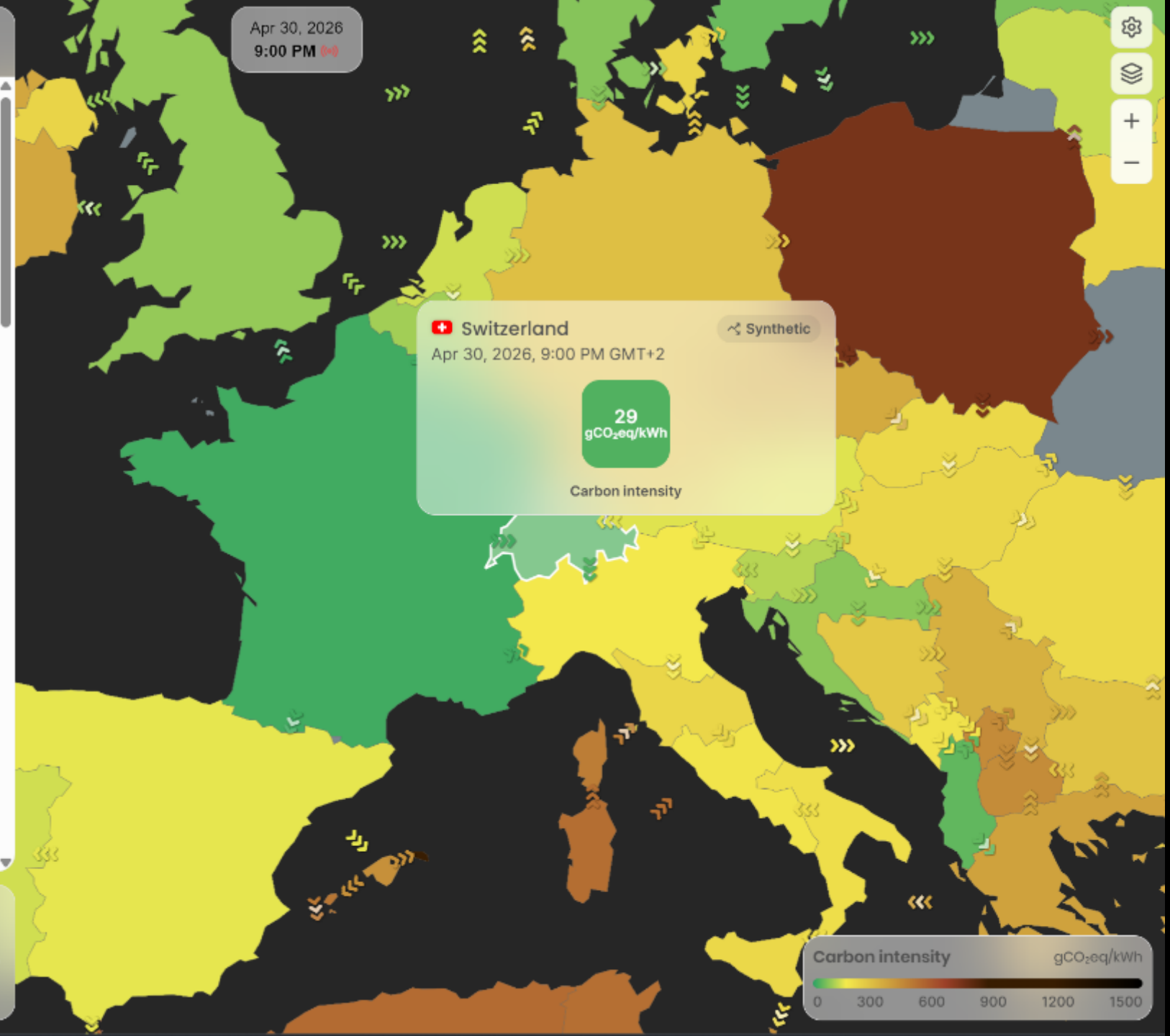


Renewable

● Installed capacity (GW)



● Electricity flow capacity (GW)



Switzerland ⓘ

Apr 30, 2026, 9:00 PM GMT+2

↔ Synthetic

29  
gCO<sub>2</sub>eq/kWh

Carbon intensity



Apr 30, 2026 9:00 PM GMT+2 0:0

15 min ▾

Apr 30 6:00 AM 12:00 PM 6:00 PM 0:0

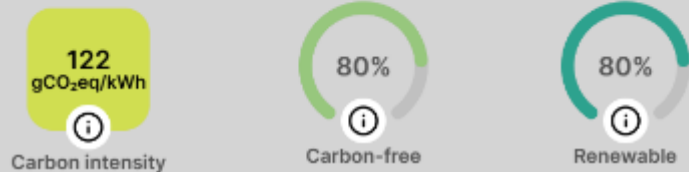
Electricity

Emissions

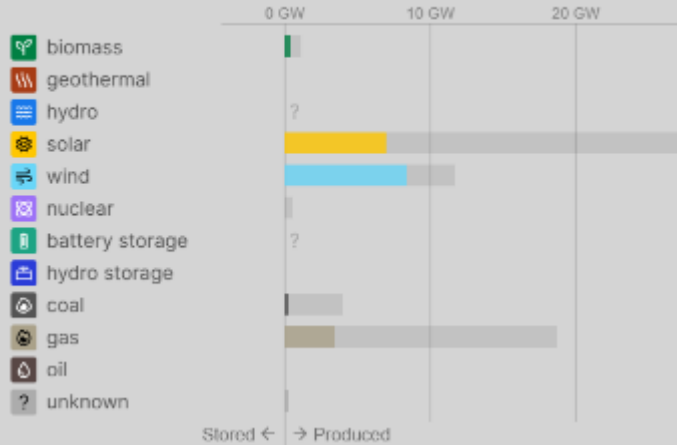
Total electricity mix

Apr 30, 2026, 9:00 PM GMT+2

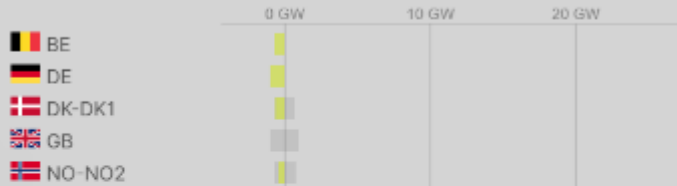
ⓘ Preliminary



● Installed capacity (GW)

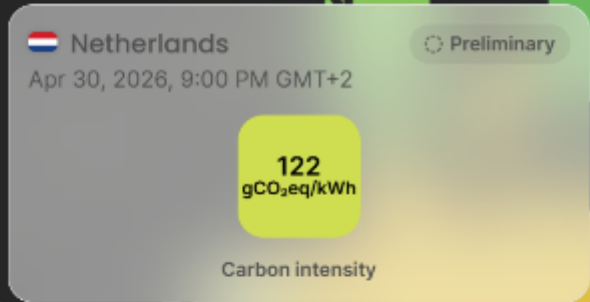


● Electricity flow capacity (GW)



Apr 30, 2026

9:00 PM ⓘ



Apr 30, 2026  
9:00 PM GMT+2 ⓘ

📅 15 min ▾



# [Sustainability]

Sustainable software  
keeps that harm from  
**compounding.**

The issue of always **ON**  
(in the cloud) 24/7

$$\begin{aligned} &1.000 \text{ units} \times 150\text{W} \times \\ &1,09 \text{ PUE} \times 8.760\text{h} = \\ &1.432.260 \text{ kWh/year} \end{aligned}$$

*What do your own  
numbers show?*



# **Three Evidence Pillars**

Use Observability, FinOps, and  
Procurement Data to prove  
sustainable IT impact

# MISCONCEPTION

Assuming code alone explains energy use

## Code

```
for item in records:  
    result = process(item)  
    save(result)
```

Application logic

## Whole story?

Code-only view

Estimated kWh?

1000 2000 3000 4000

?



Traffic



Storage



PUE



Region



Runtime  
behavior

Code matters, but it does not capture workload, facility, and location effects.

Innovate




**OBSERVABILITY**  
CPU utilization,  
traffic, storage,  
wait states



**FINOPS**  
cost, usage,  
idle resources,  
ownership



**PROCUREMENT DATA**  
hardware footprint,  
supplier data,  
region, grid mix



**ACTIONABLE  
INSIGHT**

**ACTIONS**

-  Rightsize workloads
-  Remove idle / zombie resources
-  Reduce traffic and storage waste
-  Choose lower-carbon region / timing

**LESS kWh**



0 1 2 3 4 5

**LESS CO<sub>2</sub>**





# ENERGY STATEMENT

[www.pge.com/MyEnergy](http://www.pge.com/MyEnergy)

Account No: 0123456789-0  
Statement Date: 05/30/2025  
Due Date: 06/20/2025

## Service For:

DEE P. GREENE  
123 MAIN ST.  
RICHMOND, CA 94804

## Your Account Summary

Amount Due on Previous Statement	\$99.76
Payment(s) Received Since Last Statement	-99.76
Previous Unpaid Balance	\$0.00
Current PG&E Electric Delivery Charges	\$104.48
MCE Electric Generation Charges	64.37
Current Gas Charges	71.96

## Questions about your bill?

Mon-Fri 7 a.m.-7 p.m.  
Saturday 8 a.m.-5 p.m.  
Phone: 1-800-743-5000  
[www.pge.com/MyEnergy](http://www.pge.com/MyEnergy)

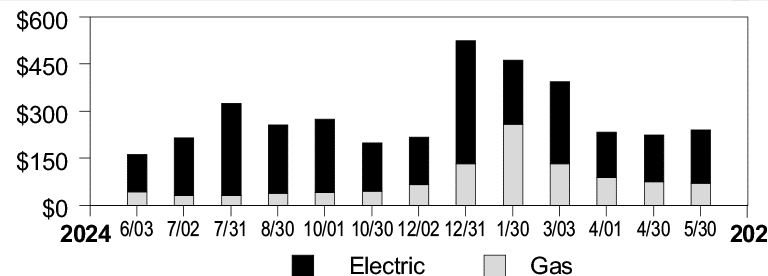
**Total Amount Due by 06/20/2025 \$240.81**

## Ways To Pay

[www.pge.com/waystopay](http://www.pge.com/waystopay)

## Monthly Billing History

### Daily Usage Comparison



1 Year Ago Period	Last Period	Current Period
14.18	13.01	14.46
Electric kWh / Day		
0.78	0.93	0.86
Gas Therms / Day		

Visit [www.pge.com/MyEnergy](http://www.pge.com/MyEnergy) for a detailed bill comparison

## Important Messages

Your charges on this page are separated into delivery charges from PG&E and generation or procurement charges from an energy provider



We have 10 core  
principles to detect  
these actions.

## **Idle Compute Reduction**

Ghost energy, monitoring shuts it down.

## **CPU Load vs Business Value**

Not every spike is user demand.

## **Right-Size Memory**

Runtime feedback updates IaC.

## **IO Visibility**

Stop blaming CPUs, fix DB locks.

## **Validate Early**

Perf testing in PRs.

## **Prune Data**

Shrink microservice / serverless payloads.

## **Compute Location**

Edge, client, cloud – choose wisely.

## **Lightweight execution**

WebAssembly.

## **Performance not by Perception**

New nano/milisecond thresholds.

## **Efficiency in User Terms**

OPUA metric (operations per user action).

Sustain

# What DevOps+++ Does

**DevOps++ provides**  
**clear, measurable**  
**actions to...**

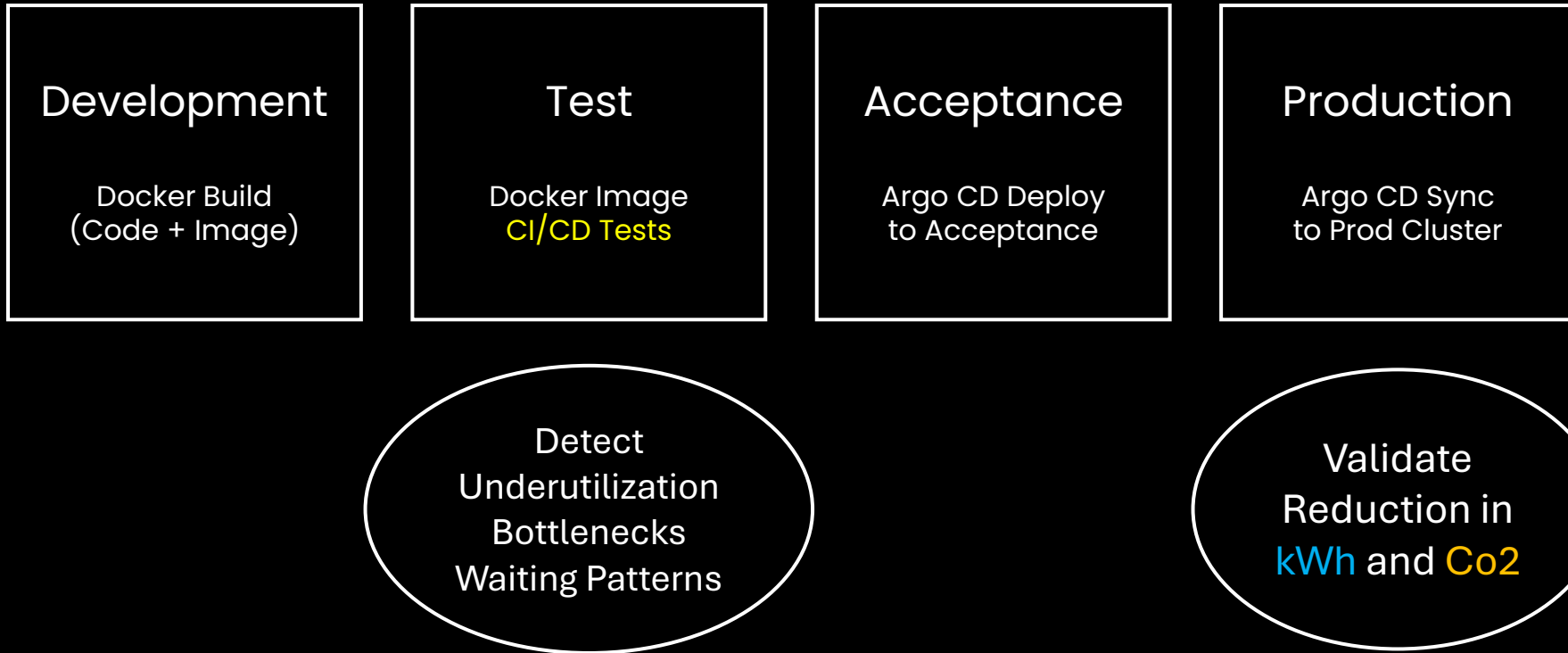
Detect inefficiencies,  
*Idle compute,*  
*bottlenecks,*  
*over-allocation.*

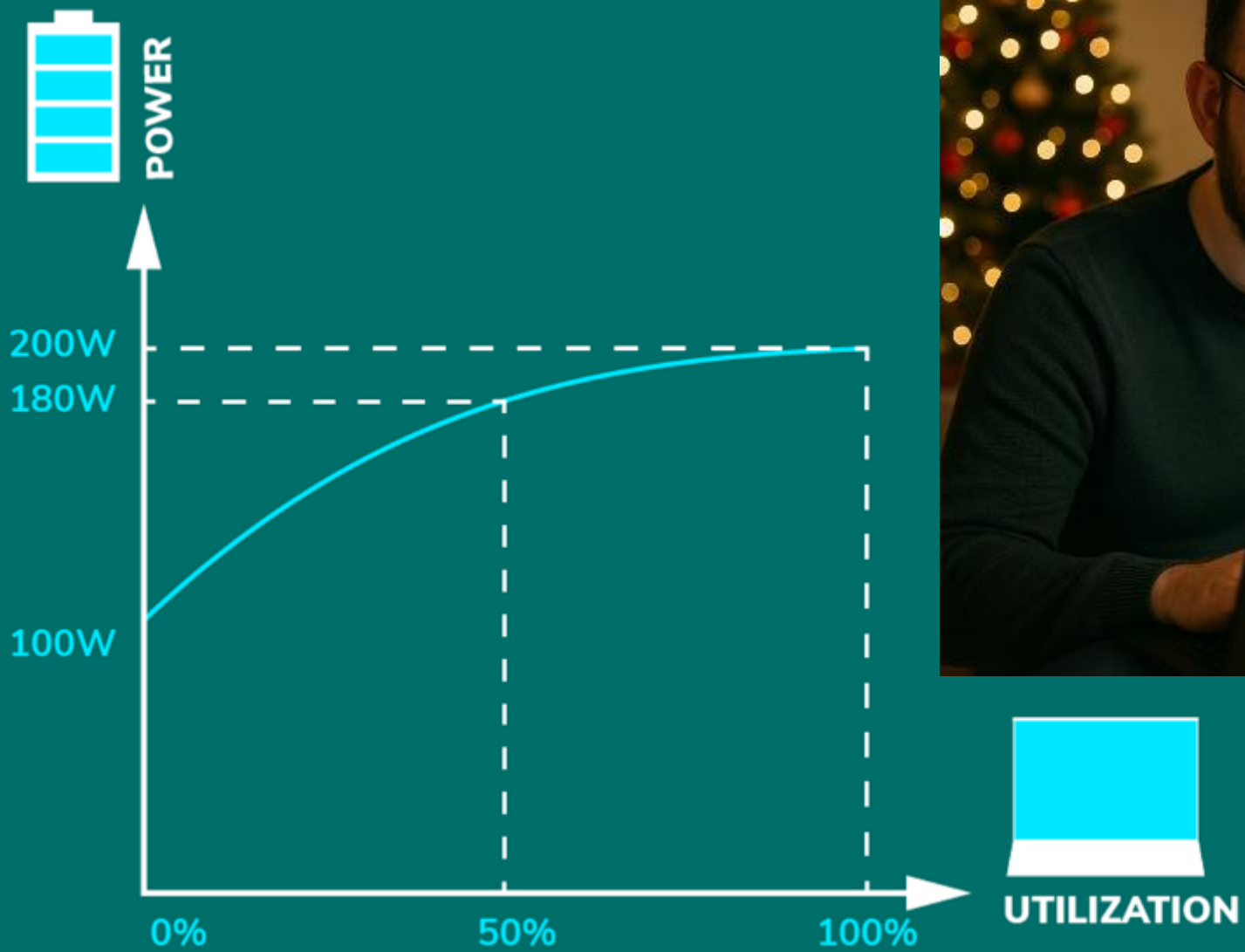
Expose them through  
**integration tests and  
observability.**

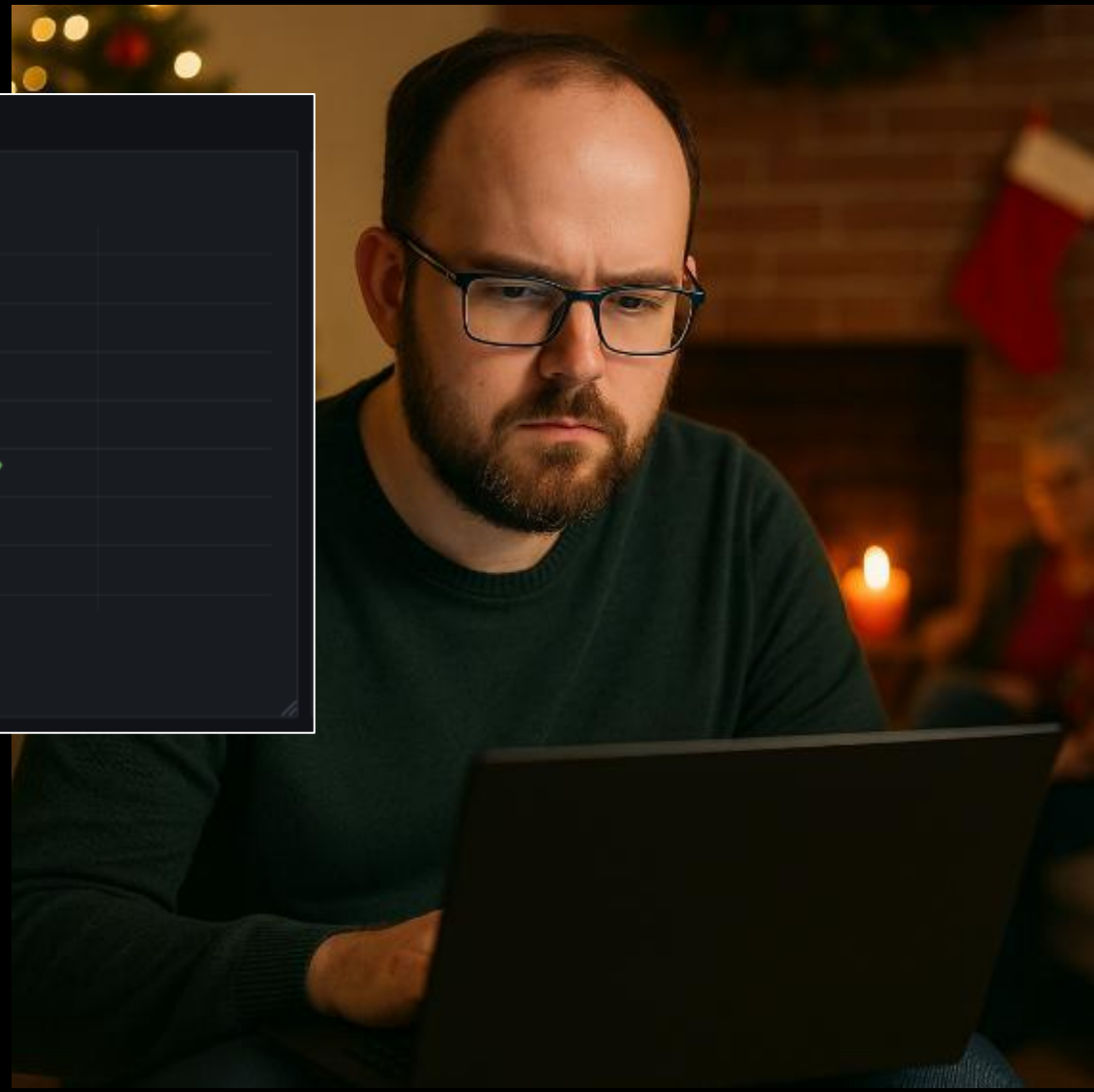
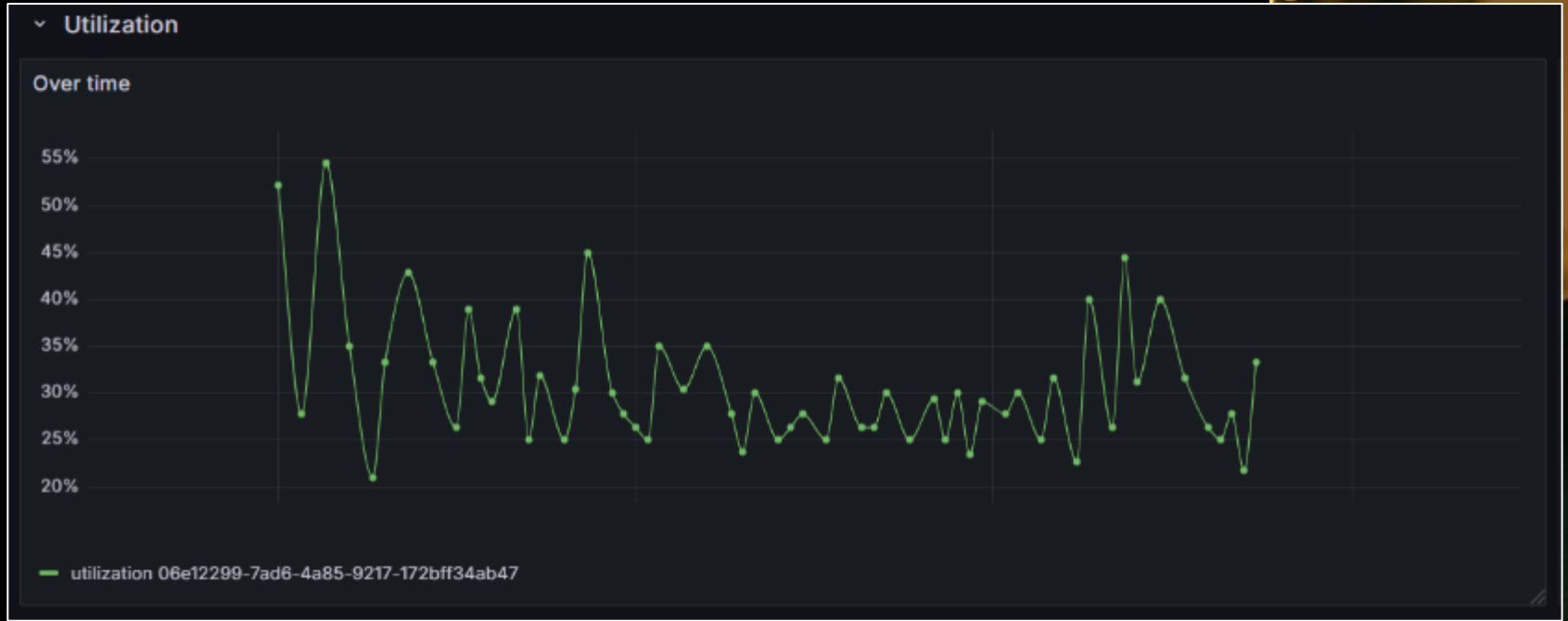
Fix them with **code**,  
**architecture**, and  
**laC** improvements.

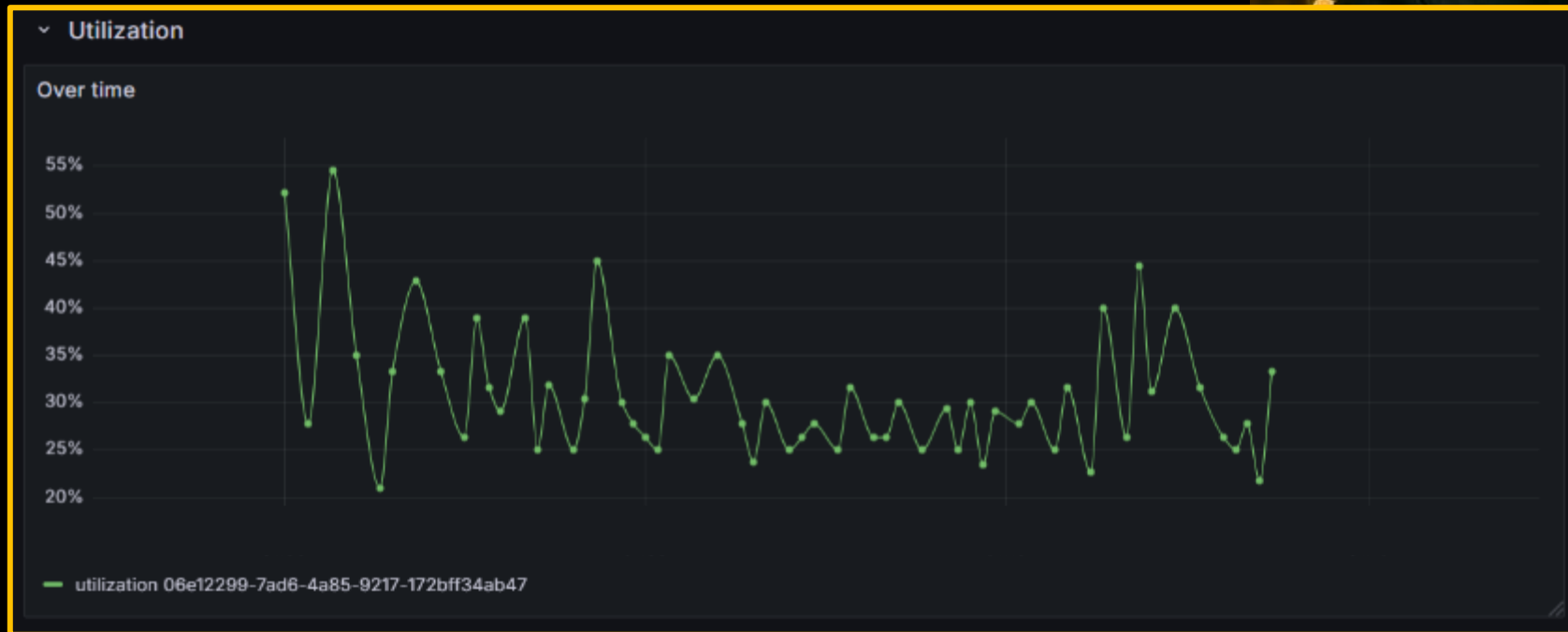
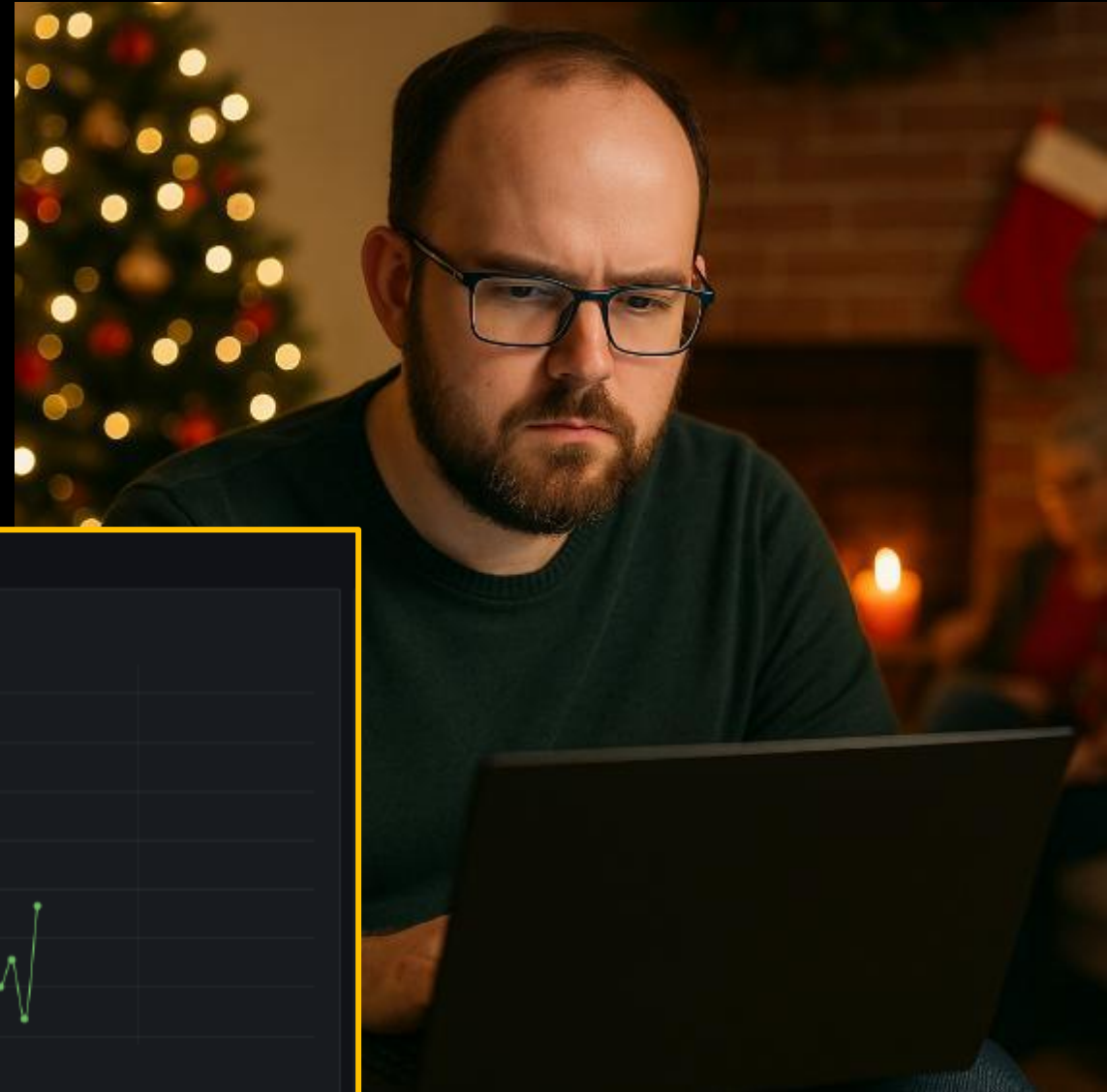
Validate  
improvements, using  
**kWh, CO<sub>2</sub>, cost, and  
performance metrics.**

# From Detection → Measured Impact









# The DevOps++ pillars

Code  
Changes

IaC Changes

Architecture  
Repository  
Updated

Less  
Energy & Co2  
In Production

Detect waste,  
prioritize impact,  
fix what matters,  
verify the reduction,  
and prevent recurrence.