Red Hat | intel

Open Tour

Connecting people and solutions
to accelerate your business

Sustainable Software Architecture

What it really is...

Markus Eisele
markus@redhat.com
@myfear

Markus Eisele

Go-To-Market-Specialist EMEA / “The Developer Guy”

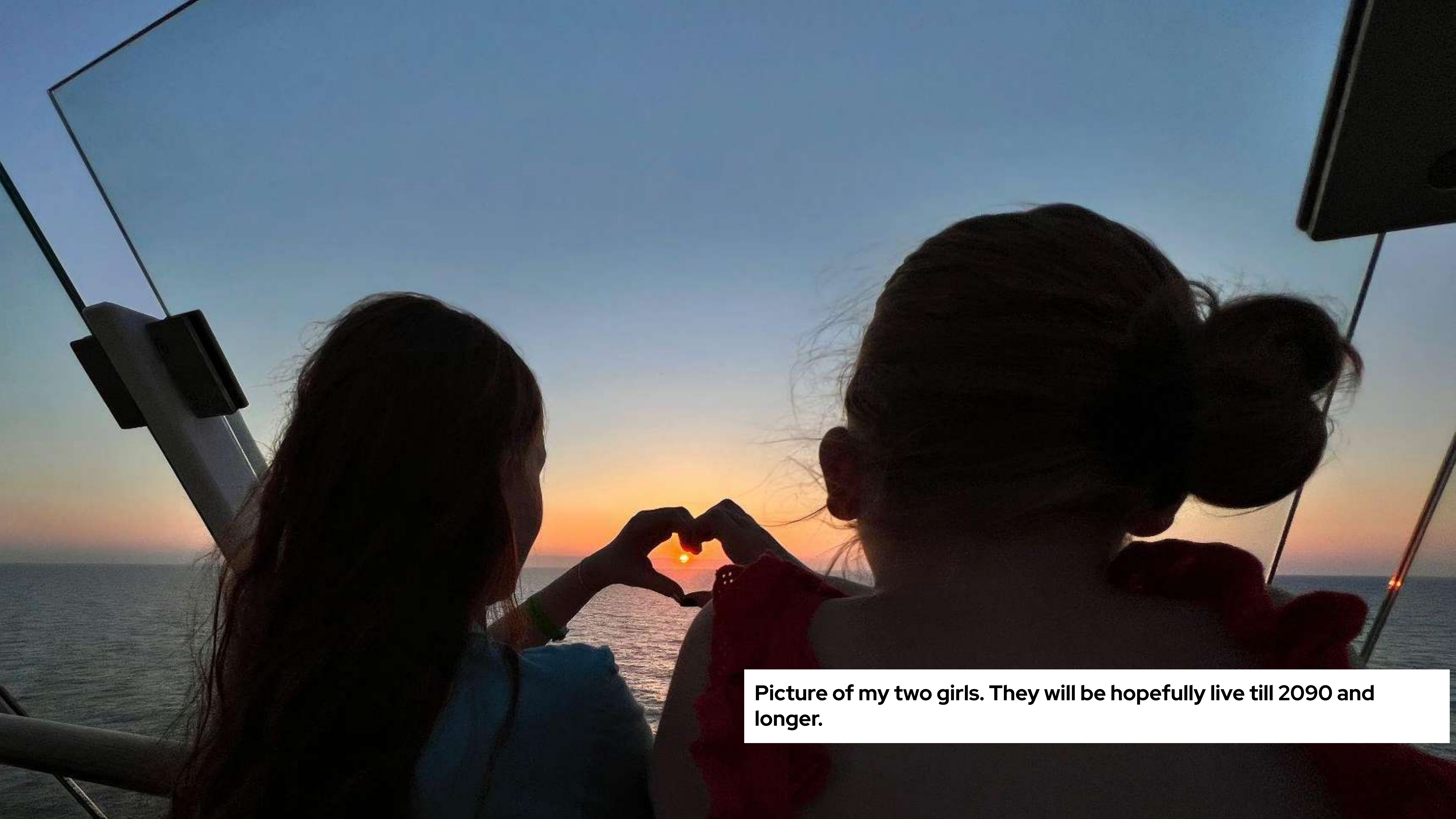
- 15 years developer and architect with Enterprise Java (Automotive, Finance, Insurance)
- 8 years Developer Relations
- 150+ presentations, 200+ articles
- 2 reports and 1 book with O’Reilly

twitter.com/@**myfear**
https://www.linkedin.com/in/markuseisele/
markus@redhat.com

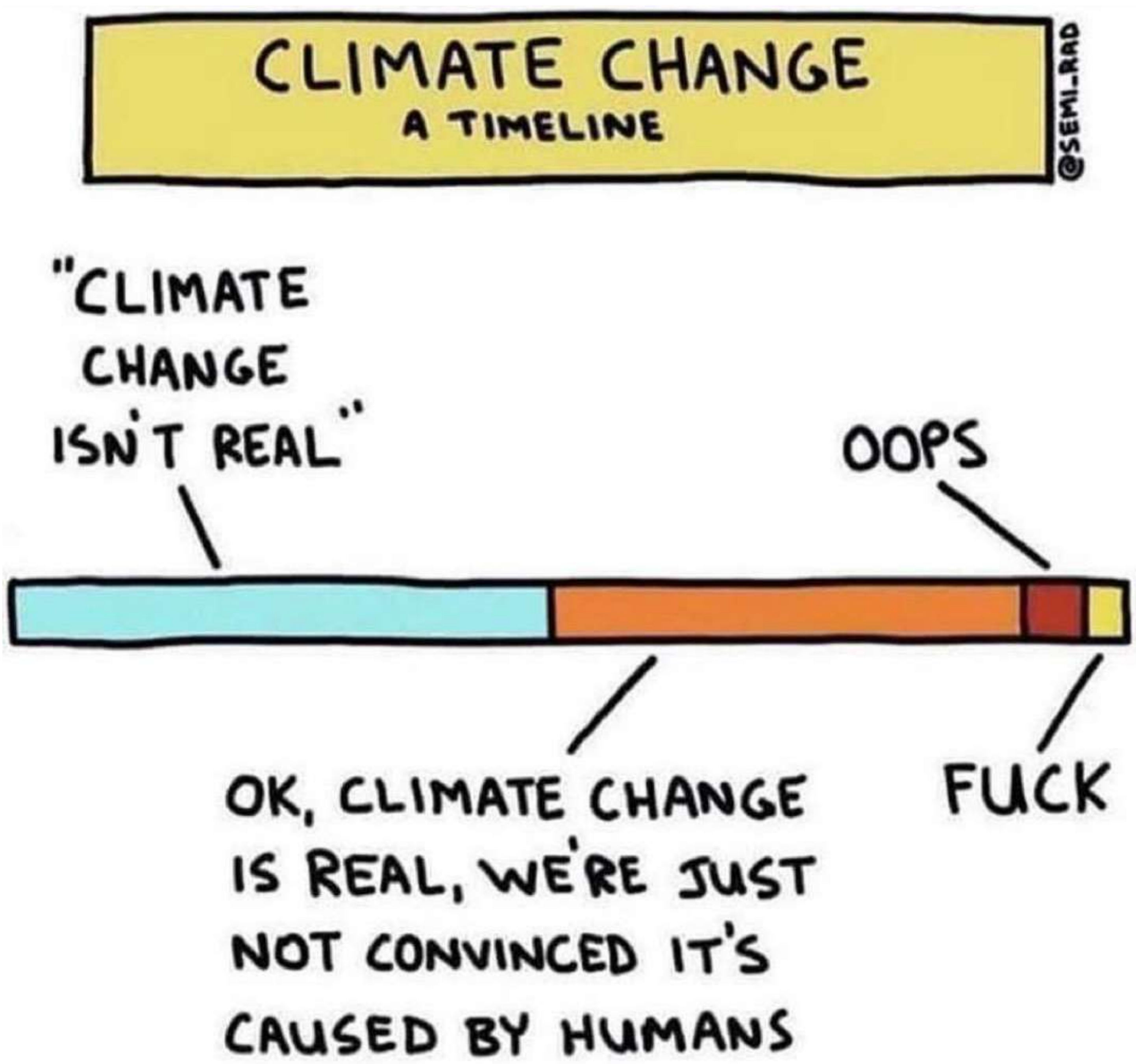
Red Hat

What we gonna look at today: 🤓

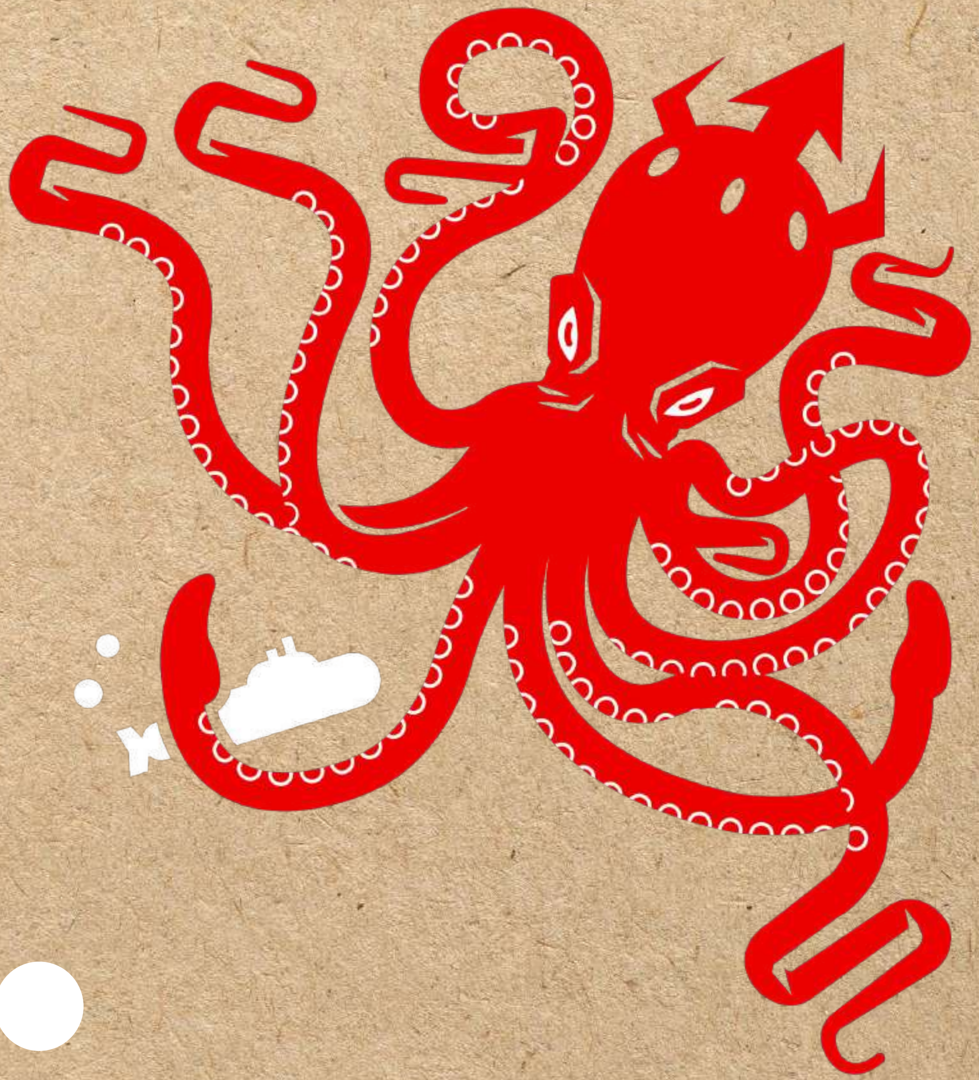
- ❑ Why I am looking at it
- ❑ Definition of ${subject}
- ❑ Why Software and Sustainability?
- ❑ Sustainability and Architecture
- ❑ Isn't there already something?
- ❑ FAANG Goals
- ❑ What can each of us do?



Picture of my two girls. They will be hopefully live till 2090 and longer.


CLIMATE CHANGE
A TIMELINE
@SEMI_RAD
"CLIMATE CHANGE ISN'T REAL"
OOPS
OK, CLIMATE CHANGE IS REAL, WE'RE JUST NOT CONVINCED IT'S CAUSED BY HUMANS
FUCK

Definition of ${subject} 🧐



„wie eine sothane [solche] Conservation und Anbau des Holzes anzustellen / daß es eine continuirliche beständige und nachhaltende Nutzung gebe / weiln es eine unentbehrliche Sache ist / ohne welche das Land in seinem Esse nicht bleiben mag"

Carl von Carlowitz 1713 "Silvicultura oeconomica"

„Humanity has the ability to make development sustainable – to ensure that it meets the needs of the present without compromising the ability of future generations to meet their own needs."

Brundtland Bericht, 1987 und Agenda 21

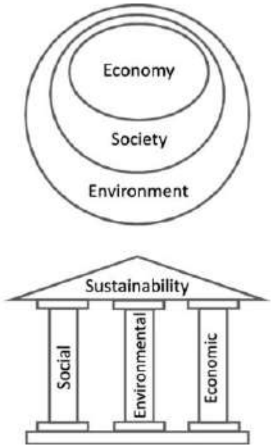
http://www.un-documents.net/our-common-future.pdf

“Sustainability is a societal goal with three dimensions (also called pillars): the environmental, economic and social dimension.”

Wikipedia, Sustainablility

https://en.wikipedia.org/wiki/Sustainability

Language Usage Today



- the **business oriented** definition of sustainability, referring to economic stability.
- der **environmentally oriented** definition, that focuses on environmentally friendly topics.
- der **socio-ecological** definition, that defines sustainability as a pair of socially and environmentally behaviours.

Digital Sustainability?

- “Digital sustainability separates itself from the broader topic of sustainability by explicitly referring to **immateriel goods**, or **knowledge**. Such non-physical resources are non rival and there is basically no consumption of resources happening. ”

https://de.wikipedia.org/wiki/Digitale_Nachhaltigkeit

What about software and sustainability?

Sustainable Software Architecture

- There is no single definition of "sustainability" but many.
- Hence there's no single definition of ${subject}
- But there are similarities:
    - It always is geared towards present and future and has a temporal reference..
    - Resources must be protected. In particular if they are non renewable.
    - Some reference point must be protected on a short- or long term bases.

"Your business or organization can have **negative environmental impacts** like direct or indirect **carbon emissions**, unrecyclable waste, and damage to shared resources like clean water."

Cloud Sustainability, AWS Well-architected FW

Sustainability Pillar AWS Well-Architected Framework, https://docs.aws.amazon.com/wellarchitected/latest/sustainability-pillar/wellarchitected-sustainability-pillar.pdf

Carbon Emissions und Greenhouse-Gases

Carbon Emissions = carbon dioxide emissions

Greenhouse Gases = (u.a.) carbon dioxide, methan, nitrous oxid, fluorocarbons , sulfur hexafluoride und nitrogen trifluoride.

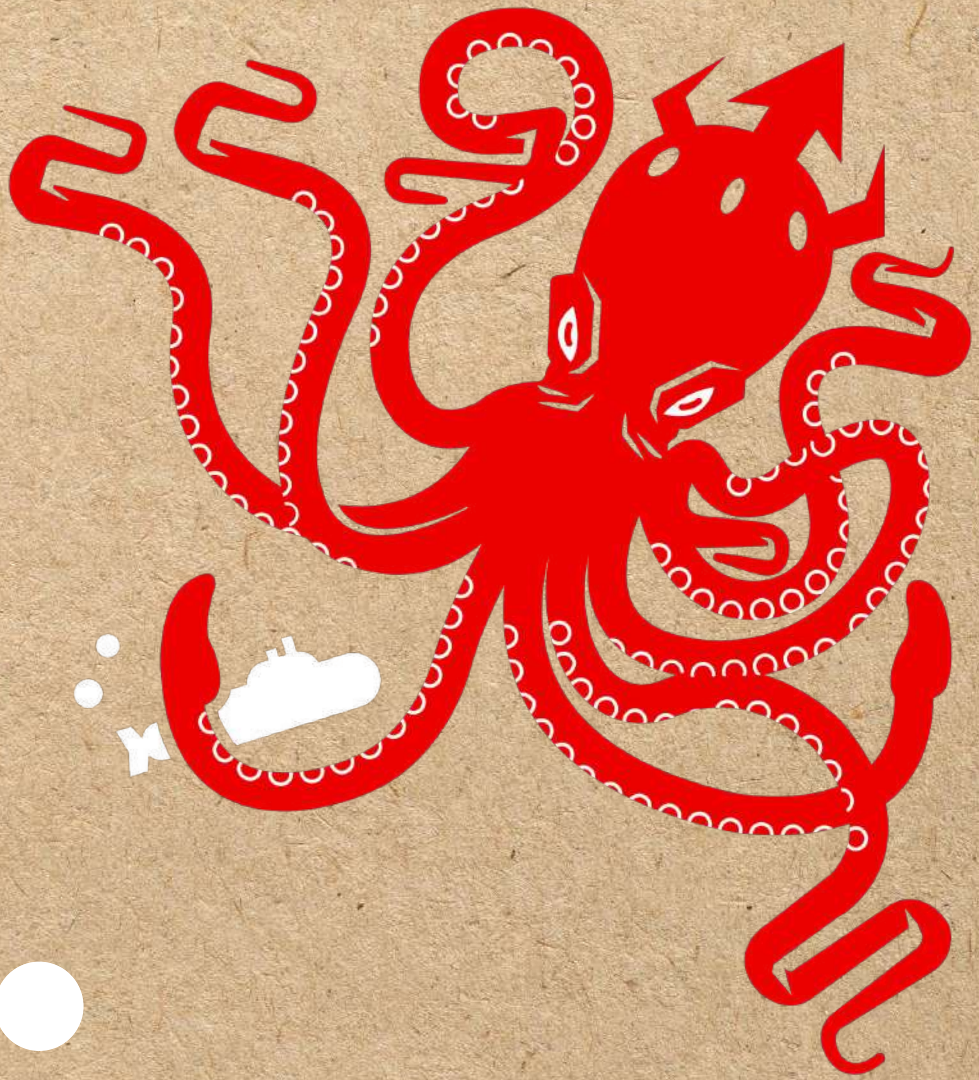
https://de.wikipedia.org/wiki/Treibhausgas

What are we doing wrong in data-centers?

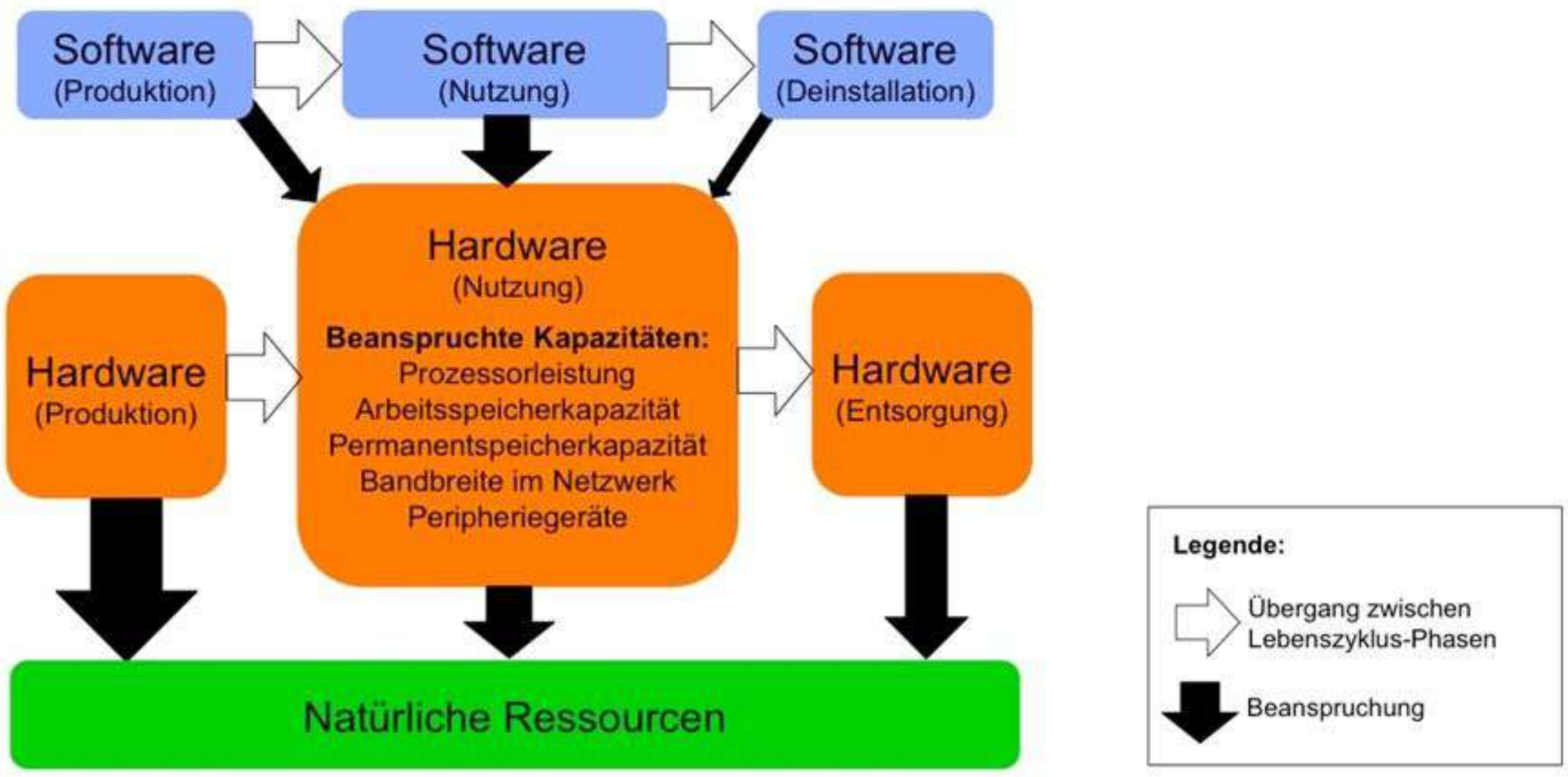
- 90% of greenhouse emissions from datacenters come from usage
- 200 bis 250 Mt CO_2e worldwide in 2020 incl. manufacturing
- 60% of greenhouse emissions result from IT-components like servers, storage and networks.

https://www.bitkom.org/sites/main/files/2020-05/2020-05_bitkom_klimastudie_digitalisierung.pdf

Sustainability and software architecture



UFOPLAN-Projekt „Sustainable Software Design - Entwicklung einer Methodik zur Bewertung der Ressourceneffizienz von Softwareprodukten" (2017)



https://www.umwelt-campus.de/forschung/projekte/green-software-engineering/kriterienkatalog/einleitung

Criteria for sustainable software

Abstrakt formuliert, stehen zwei durch ein Softwareprodukt verursachte Flüsse im Vordergrund der Betrachtung:

There’s basically two main areas in focus for software products:

- Amount of used hardware (new to waste)
- Amount of consumed energy through hardware (electricity to heat)


Universität Zürich
Umwelt-Campus Birkenfeld
Öko-Institut e.V.

Kriterienkatalog für nachhaltige Software

Lorenz Hilty, Stefan Naumann, Yuliyan Maksimov, Eva Kern, Andreas Filler, Achim Guldner, Jens Gröger

Version 01
Stand: 31.05.2017

UFOPLAN-Projekt „Sustainable Software Design - Entwicklung einer Methodik zur Bewertung der Ressourceneffizienz von Softwareprodukten"
FKZ-Nr.: 3715 37 601 0
Im Auftrag des Umweltbundesamtes

Projekt-Ansprechpartner

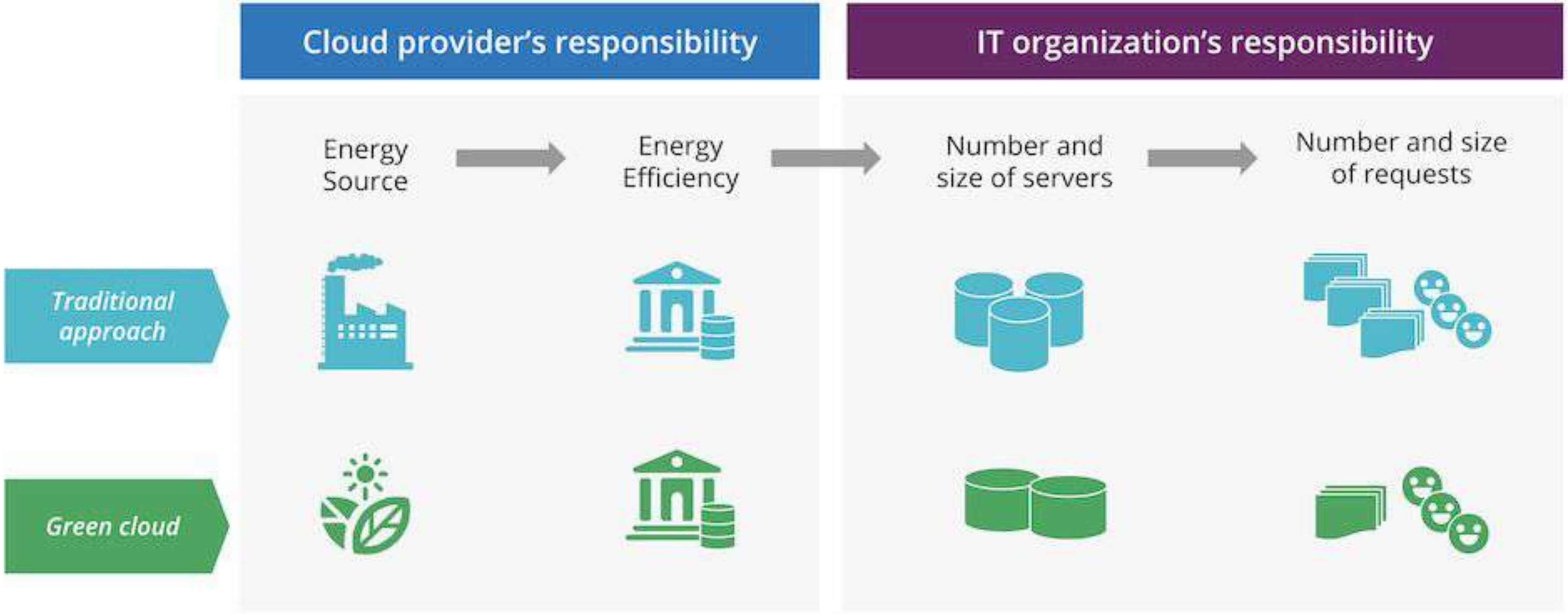
Dipl.-Ing. Jens Gröger
Öko-Institut e.V.
Tel. +49 30 40 50 85 378, j.groeger@oeko.de

Prof. Dr. Stefan Naumann
Hochschule Trier, Standort Umwelt-Campus Birkenfeld
Tel. +49 6782 17 12 17, s.naumann@umwelt-campus.de

Prof. Dr. Lorenz Hilty
University of Zurich
Tel. +41 44 635 67 24, hilty@ifi.uzh.ch

https://www.umwelt-campus.de/fileadmin/Umwelt-Campus/Greensoft/Kriterienkatalog_nachhaltige_Software_v01_2017-05-31.pdf

The four pillars of green cloud optimization

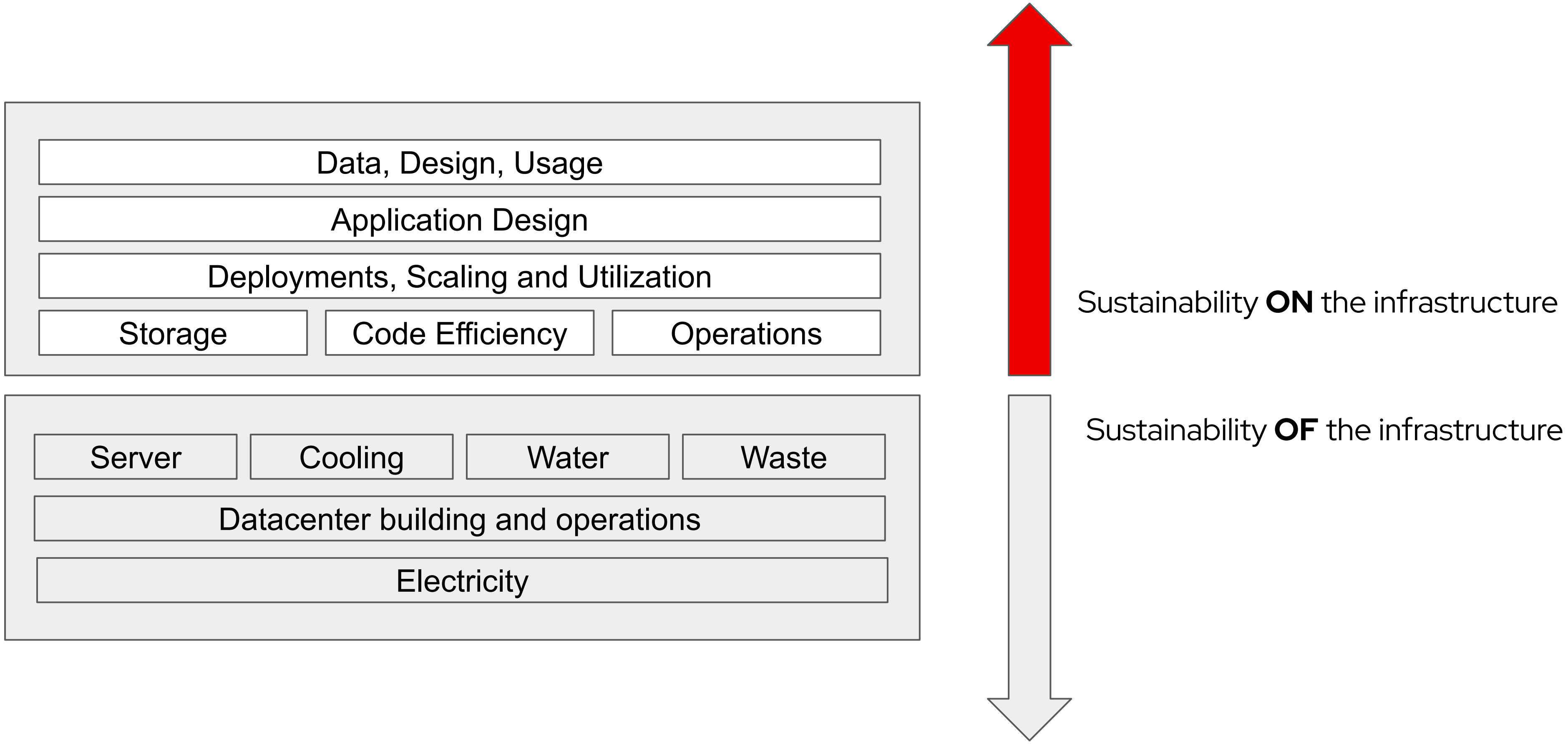


https://www.thoughtworks.com/insights/articles/green-cloud

AWS Design Principles for Sustainability

- Understand your impact
- Establish sustainability goals
- Maximise Utilization
- Anticipate and adapt latest Hardware
- Use managed services
- Reduce downstream impact of your cloud workload

https://docs.aws.amazon.com/wellarchitected/latest/sustainability-pillar/wellarchitected-sustainability-pillar.pdf


Data, Design, Usage
Application Design
Deployments, Scaling and Utilization
Storage
Code Efficiency
Operations
Server
Cooling
Water
Waste
Datacenter building and operations
Electricity
Sustainability **ON** the infrastructure
Sustainability **OF** the infrastructure

Data Design, Usage and Storage

- Implement a data classification policy
- Use technologies that support data access and storage patterns
- Use lifecycle policies to delete unnecessary data
- Minimize over-provisioning in block storage
- Remove unneeded or redundant data
- Use shared file systems or object storage to access common data
- Minimize data movement across networks
- Back up data only when difficult to recreate

Software Application Design

- Add requirements for sustainability to your development process.
- Enable resources to work in parallel to develop, test, and deploy sustainability improvements.
- Test potential improvements using the minimum viable representative components.

Platform Deployments, Utilization and Scaling

- Use minimum viable representative environments to develop and test potential improvements.
- Use automation to maximize utilization of your development and test environments.
- Use instance types with burst capacity, spot instances, and other technologies to align build capacity with use.
- Take advantage of custom-purpose hardware that is specific to your workload.

Code Efficiency

Table 4. Normalized global results for Energy, Time, and Memory

Total					
	Energy		Time		Mb
(c) C	1.00	(c) C	1.00	(c) Pascal	1.00
(c) Rust	1.03	(c) Rust	1.04	(c) Go	1.05
(c) C++	1.34	(c) C++	1.56	(c) C	1.17
(c) Ada	1.70	(c) Ada	1.85	(c) Fortran	1.24
(v) Java	1.98	(v) Java	1.89	(c) C++	1.34
(c) Pascal	2.14	(c) Chapel	2.14	(c) Ada	1.47
(c) Chapel	2.18	(c) Go	2.83	(c) Rust	1.54
(v) Lisp	2.27	(c) Pascal	3.02	(v) Lisp	1.92
(c) Ocaml	2.40	(c) Ocaml	3.09	(c) Haskell	2.45
(c) Fortran	2.52	(v) C#	3.14	(i) PHP	2.57
(c) Swift	2.79	(v) Lisp	3.40	(c) Swift	2.71
(c) Haskell	3.10	(c) Haskell	3.55	(i) Python	2.80
(v) C#	3.14	(c) Swift	4.20	(c) Ocaml	2.82
(c) Go	3.23	(c) Fortran	4.20	(v) C#	2.85
(i) Dart	3.83	(v) F#	6.30	(i) Hack	3.34
(v) F#	4.13	(i) JavaScript	6.52	(v) Racket	3.52
(i) JavaScript	4.45	(i) Dart	6.67	(i) Ruby	3.97
(v) Racket	7.91	(v) Racket	11.27	(c) Chapel	4.00
(i) TypeScript	21.50	(i) Hack	26.99	(v) F#	4.25
(i) Hack	24.02	(i) PHP	27.64	(i) JavaScript	4.59
(i) PHP	29.30	(v) Erlang	36.71	(i) TypeScript	4.69
(v) Erlang	42.23	(i) Jruby	43.44	(v) Java	6.01
(i) Lua	45.98	(i) TypeScript	46.20	(i) Perl	6.62
(i) Jruby	46.54	(i) Ruby	59.34	(i) Lua	6.72
(i) Ruby	69.91	(i) Perl	65.79	(v) Erlang	7.20
(i) Python	75.88	(i) Python	71.90	(i) Dart	8.64
(i) Perl	79.58	(i) Lua	82.91	(i) Jruby	19.84

"Although the most energy efficient language in each benchmark is almost always the fastest one, the **fact is that there is no language which is consistently better than the others**,"

https://sites.google.com/view/energy-efficiency-languages/home

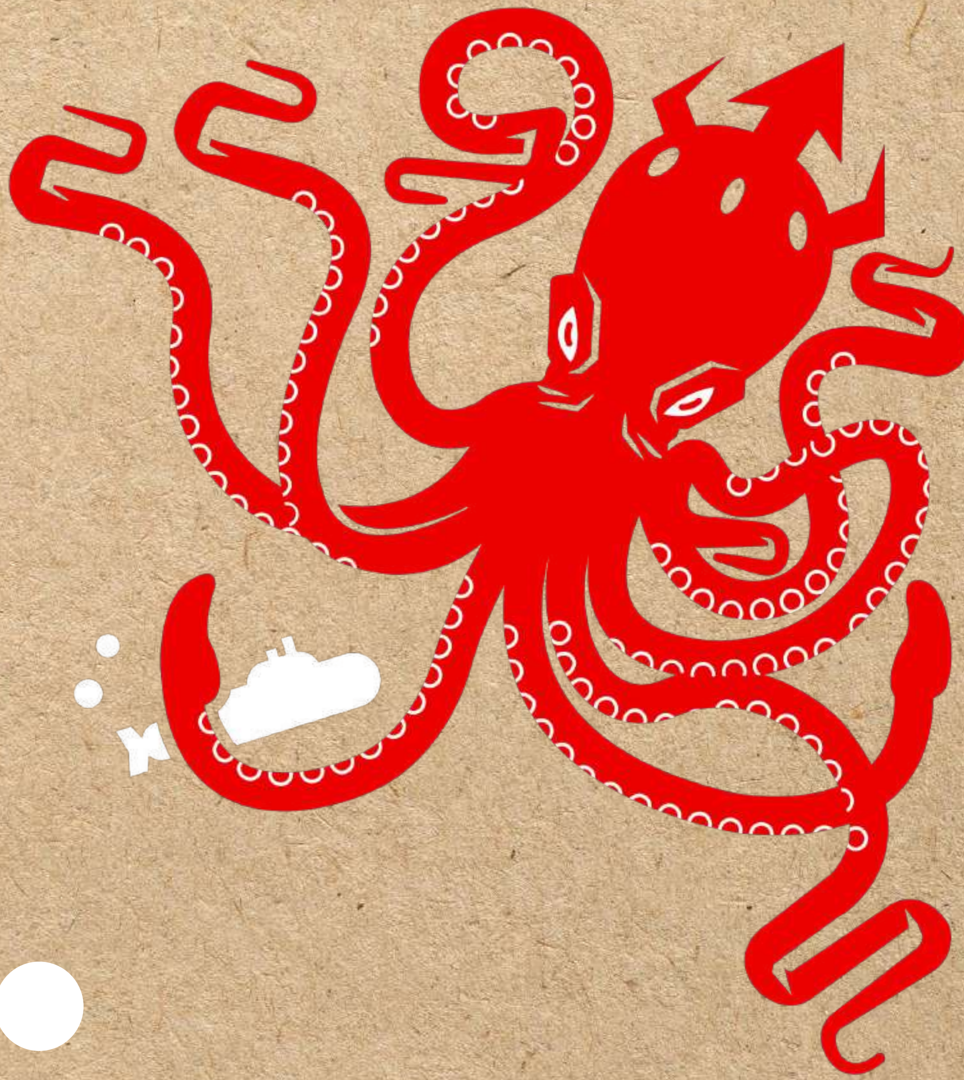
10 recommendations for green software development

- Focus on and control features with higher power consumption and common usage scenarios.
- Reduce data usage. Adopt efficient cache policy, minimise data exchange, and manage the lifecycle of stored data - compress and aggregate data.
- Remove or refactor unused features.
- Detect and remove loops which can't achieve their intended purpose and uselessly consume energy. For example, polling an unreachable server.
- Adapt your app's behavior according to the device power mode or other operating conditions.
- Limit computational accuracy of the application to the desired level which commensurate with the operational needs. For example, you do not need very high resolution geolocation data of your user, if you are just looking for friends nearby.
- Monitor real-time energy consumption of the application, to identify the modules that can be optimized to produce fewer emissions.
- Developing and using a less-power-consuming ML model; creating and sharing reproducible code that will reduce duplicated efforts; and developing and using specialized hardware optimized for AI workload.
- During development monitor real-time power consumption through techniques such as dynamic code analysis.

https://greensoftware.foundation/articles/10-recommendations-for-green-software-development code + data = energy = carbon emissions.

And there are probably more...

Isn’t there something out there already?



“Without data, you're just another person with an opinion.”

W. Edwards Deming

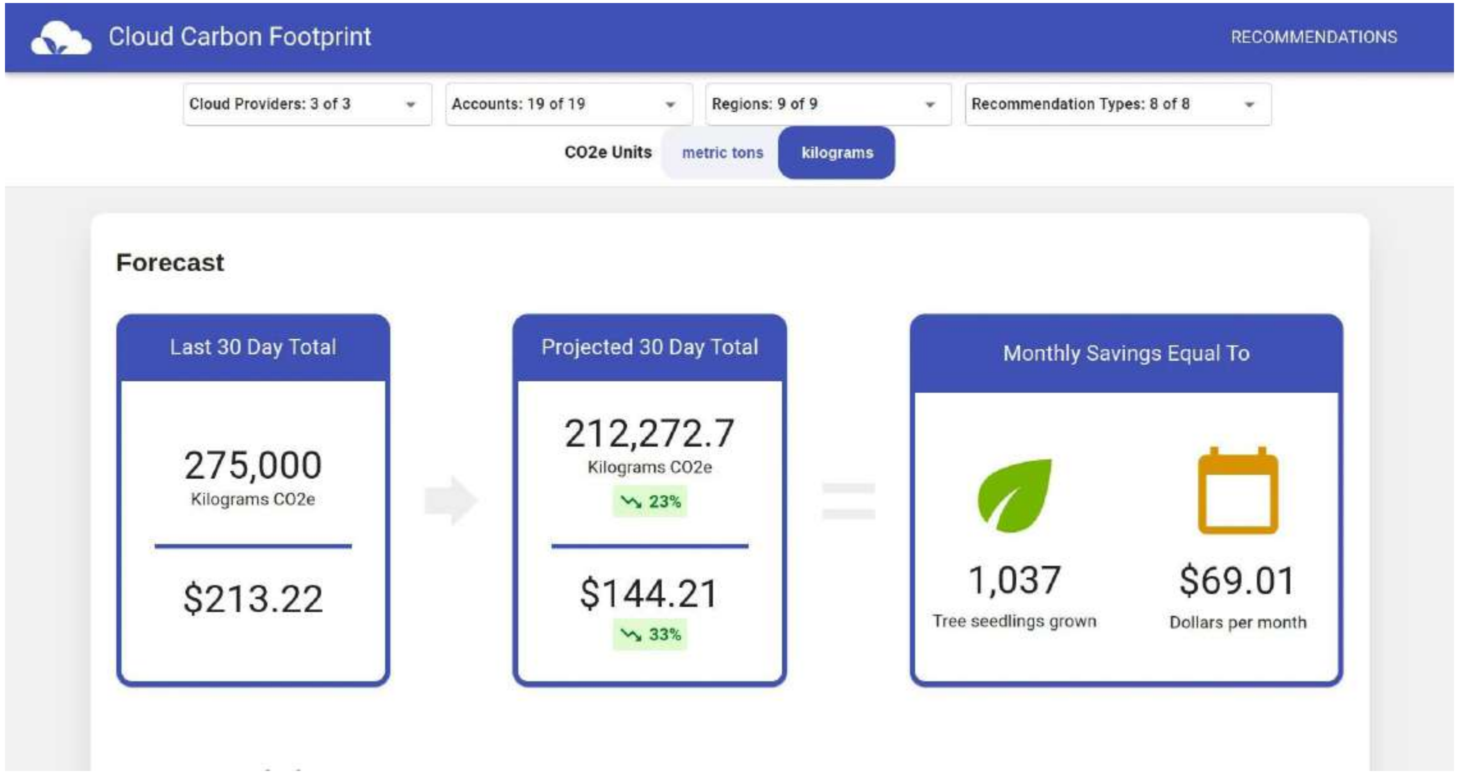
Kube Green

- kube-green is a simple k8s addon that automatically shuts down (some of) your resources when you don't need them.

https://github.com/kube-green/kube-green

Cloud Carbon Footprint

Cloud Carbon Footprint is a tool to estimate energy use (kilowatt-hours) and carbon emissions (metric tons CO2e) from public cloud usage



https://github.com/cloud-carbon-footprint/cloud-carbon-footprint

Scaphandre

Scaphandre [skafɑ̃dʁ] is a metrology agent dedicated to electrical power consumption metrics. The goal of the project is to permit to any company or individual to measure the power consumption of its tech services and get this data in a convenient form, sending it through any monitoring or data analysis toolchain.

https://github.com/hubblo-org/scaphandre

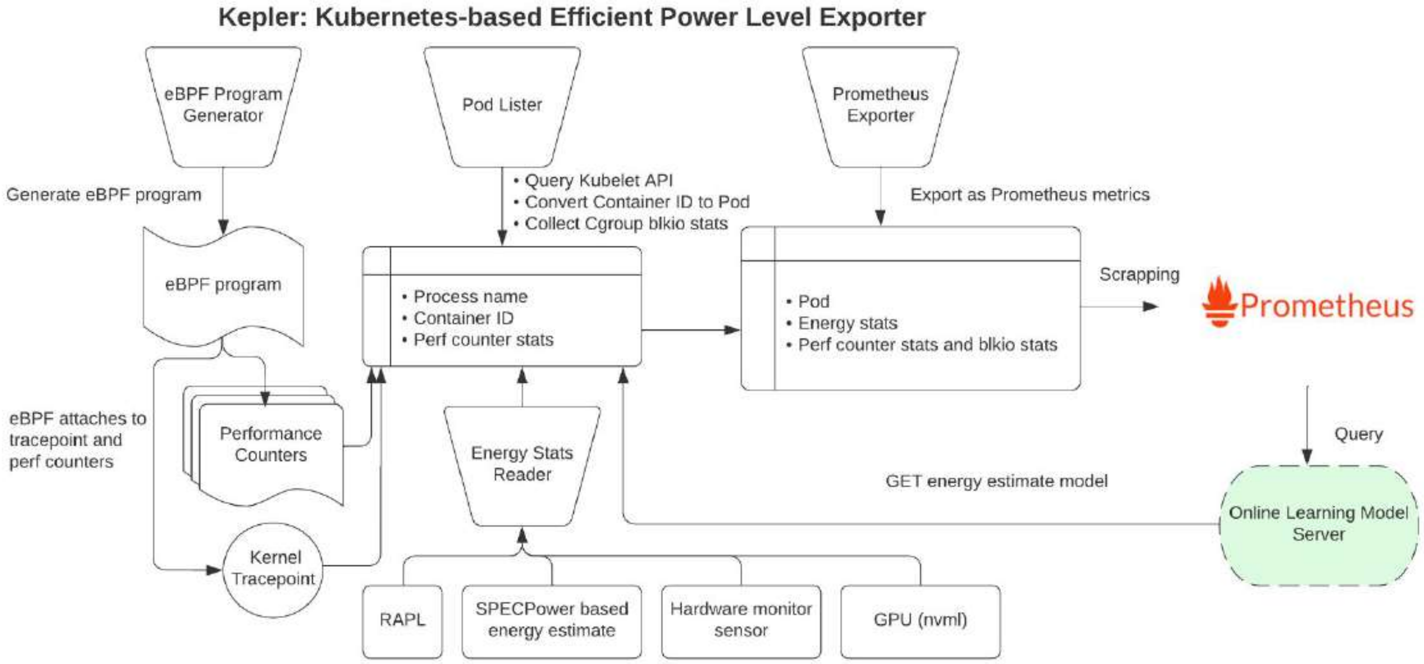
Kube-downscaler

- Scale down / "pause" Kubernetes workload (Deployments, StatefulSets, and/or HorizontalPodAutoscalers and CronJobs too !) during non-work hours.

https://codeberg.org/hjacobs/kube-downscaler

Kepler

Kepler (Kubernetes Efficient Power Level Exporter) uses eBPF to probe energy related system stats and exports as Prometheus metrics



https://github.com/sustainable-computing-io/kepler

FAANG goals

Sustainability Reports ("Nachhaltigkeitsberichte")

- **Facebook/Meta** " Bis 2030 wollen wir in unserer Wertschöpfungskette Netto-Null-Emissionen erreichen." (https://sustainability.fb.com/wp-content/uploads/2021/06/2020_FB_Sustainability-Report.pdf)
- **Amazon** "CO2-neutral (net-zero carbon) bis 2040" (https://nachhaltigkeit.aboutamazon.de/)
- **Apple** "Bis 2030 CO_2 neutral werden" (https://www.apple.com/de/environment/pdf/Apple_Environmental_Progress_Report_2021.pdf)
- **Netflix** "Bis Ende 2022 wird Netflix Netto-Null-Emissionen von Treibhausgasen erreichen." (https://about.netflix.com/de/sustainability)
- **Google** "CO2-frei bis 2030" (https://sustainability.google/intl/de/)

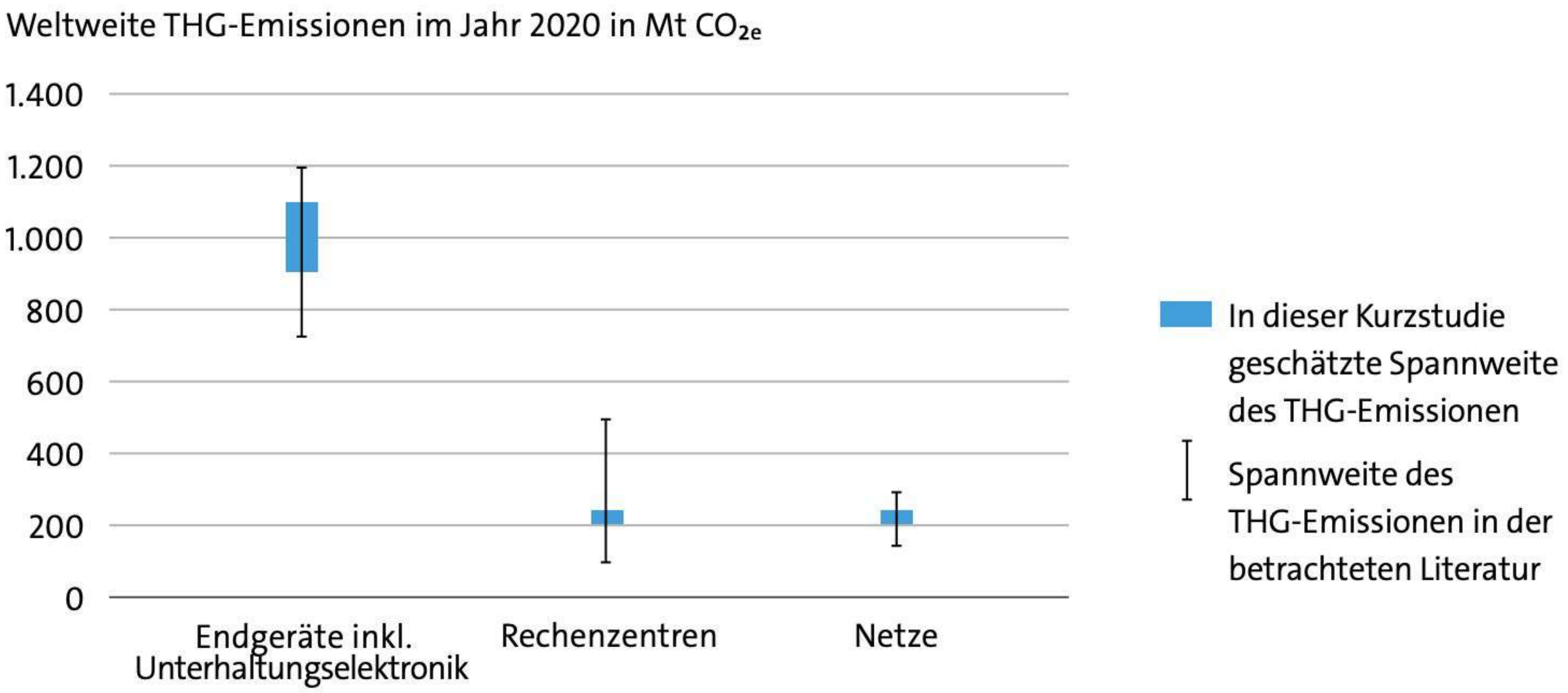
https://ec.europa.eu/info/business-economy-euro/company-reporting-and-auditing/company-reporting/corporate-sustainability-reporting_de

What can each of us do?

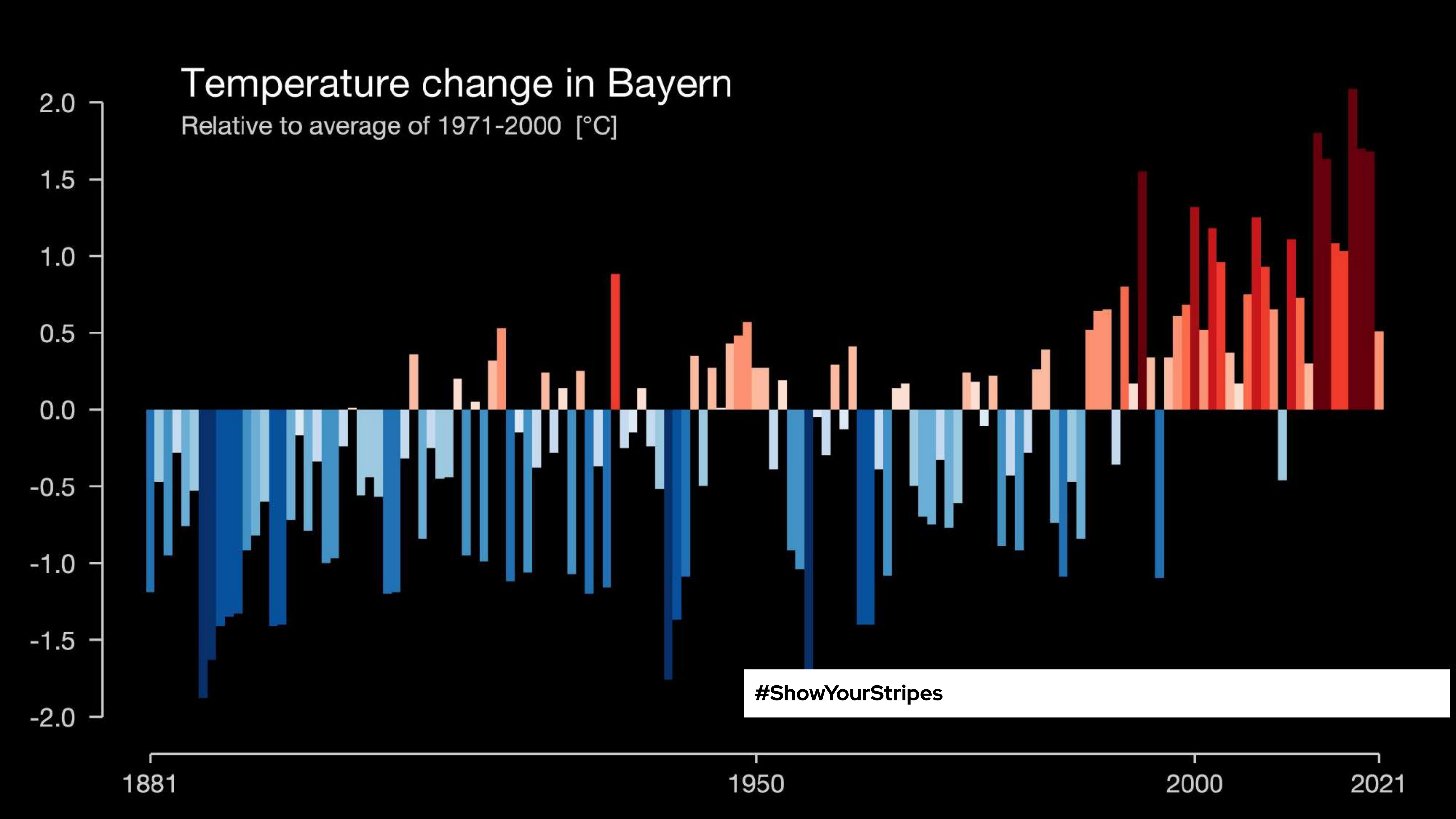
Why NOW? Why ALL OF US?

- The world climate report assumes it’s just three years left to reduce greenhouse gas emissions by 45% to prevent desaster.
- Most companies don’t move fast enough to even get close to this goal.

https://www.ipcc.ch/report/ar6/wg2/



https://www.bitkom.org/sites/main/files/2020-05/2020-05_bitkom_klimastudie_digitalisierung.pdf


Temperature change in Bayern
Relative to average of 1971-2000 [°C]
2.0
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
-2.0
1881
1950
2000
2021
#ShowYourStripes

Red Hat

Red Hat Developer

Join Red Hat Developer.

Build here. Go anywhere.

youtube.com/RedHatDevelopers

linkedin.com/showcase/red-hat-developer

facebook.com/RedHatDeveloperProgram

twitter.com/rhdevelopers

developers.redhat.com/developer-sandbox

Learn containers, Kubernetes, and OpenShift in your browser.

Start exploring in the OpenShift Sandbox.

Try Red Hat's products and technologies
without setup or configuration.

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat