Sustainable Software Architecture at DATEV

Michael Kircher

Small Amounts … Big Impact.

Every **digital invoice** saves **50 g** CO_2 (54 g vs. 4 g), but only **27 %** of DATEV customers like to receive digitally.

Digital invoices would save **6.3 million** sheets of paper equal to **23 tons of CO_2**.

Every **digital pay slip** saves **25 g** CO_2 (29 g vs. 4 g), but only **2 million out of 14 million** pay slips are received digitally per month.

Michael Buckow, DATEV Beauftragter für Nachhaltigkeit u. Umweltschutz

INTECHNICA CONSULT GmbH

DATEV

PRODUCT CARBON FOOTPRINT

Elektronische Rechnung vs. Papierrechnung

Intechnica Consult GmbH bestätigt den von uns
für das Unternehmen

DATEV eG
Paumgartnerstraße 6-14, 90429 Nürnberg

erstellten Product Carbon Footprint für jeweils eine vergleichbare Papierrechnung bzw. elektronische Rechnung.
Die Ermittlung erfogIt in Anlehnung an das GHG Protocol (Product Standard).

Papierrechnung:	52,64 g CO_2-Äquvivalente
Elektronische Rechnung:	3,93 g CO_2-Äquvivalente

Der dazugehörige Bericht beschreibt die Rahmenbedinungen und Annahmen, die der Berechnung der Product Carbon Footprints zugrundeliegen.

Nürnberg, 18.07.2022

Max Windsheimer
Berater für Umwelt und Nachhaltigkeit
Intechnica Consult GmbH

Thomas Täuber
Geschäftsführung
Intechnica Consult GmbH

The Sustainability Focus in this Presentation

01
Evolvability
Forces

04
Energy Efficiency
Forces

02
Evolvability
Good Practices

05
Energy Efficiency
Good Practices

03
Evolvability
Hypotheses

06
Energy Efficiency
Hypotheses

DATEV and its Datacenter as Functional and Technical Data Hub

Deployment Locations for DATEV Products

Tax Offices

- Local Database
- Communication w/ DATEV Datacenter

DATEV

‘Classic’ Datacenter

- Batch
- EJB
- PaaS
- IaaS / Hosting

‘Cloud-native' Datacenter

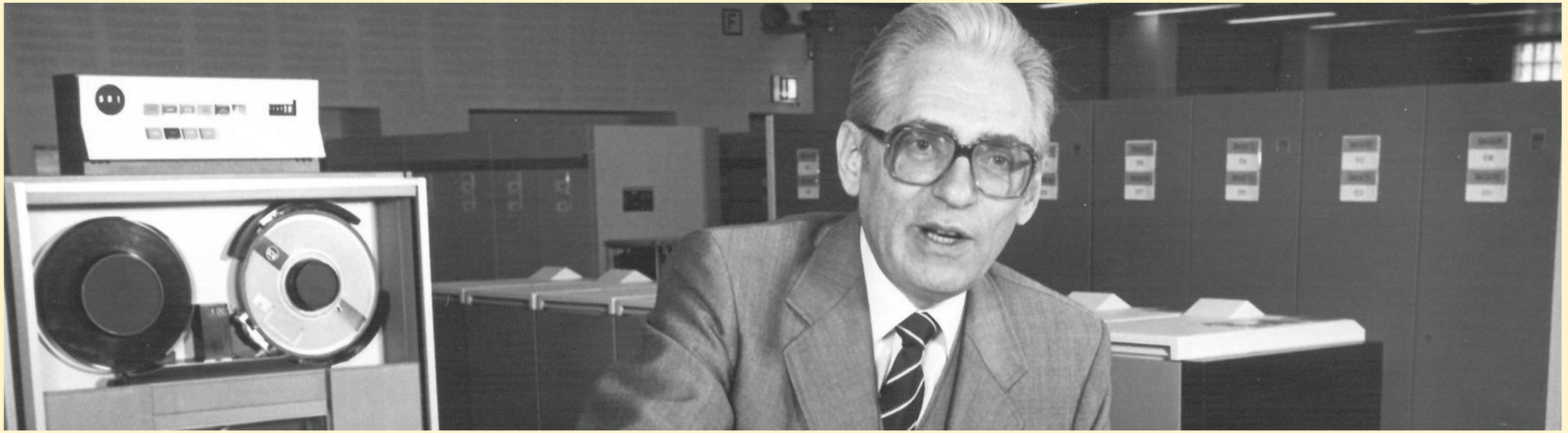
- PaaS
- Containers

Public Cloud

- AWS
- Azure
- European / Sovereign

Evolvability

Evolvability Forces at DATEV

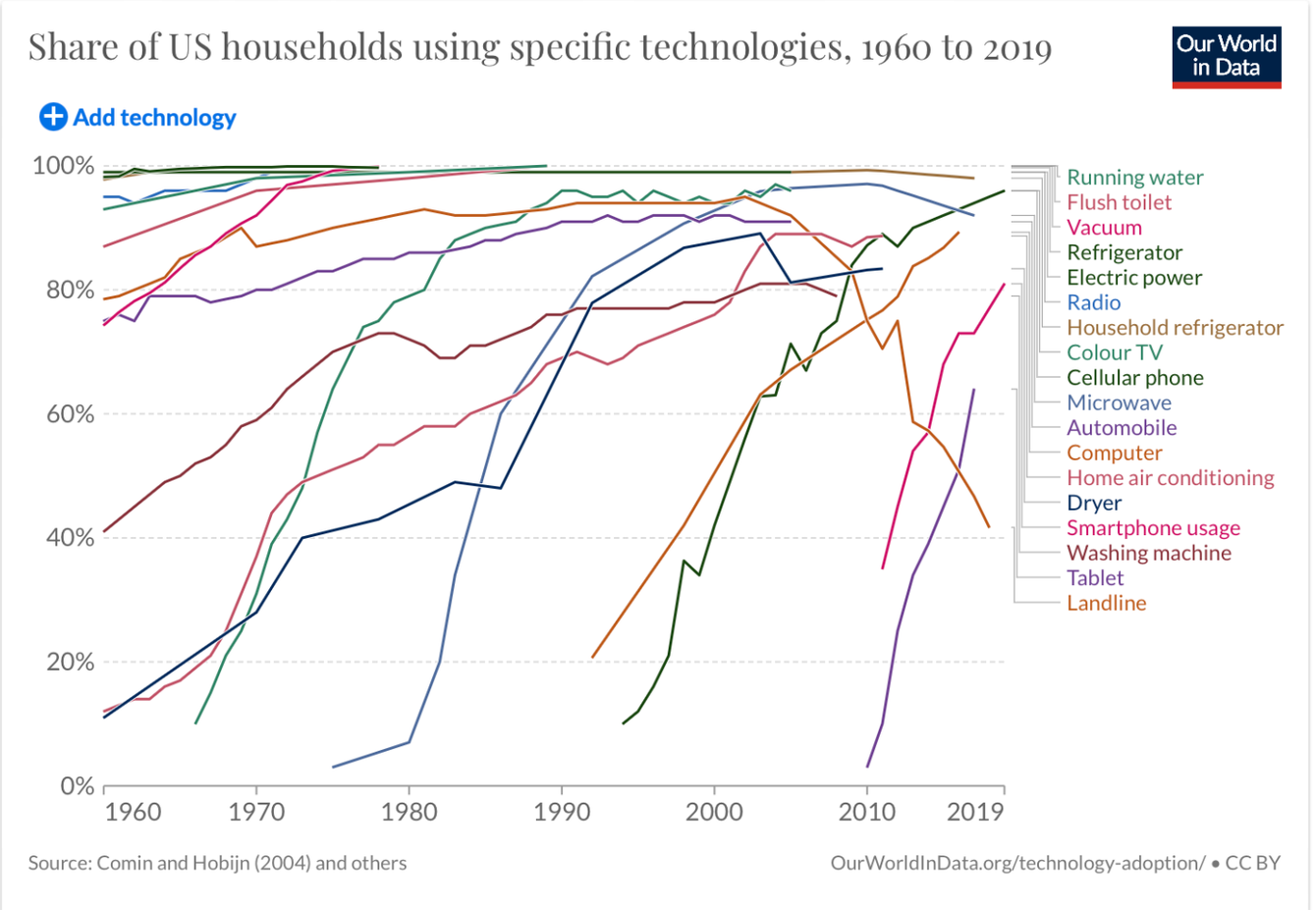


- Archiving requirements: up to 15 years
- High baseline effort for legislation actuality: 60 - 80 % per team
- Very broad portfolio: full-service provider for tax consultants
- Every datacenter technology needs operations staff

Good Practice:
Mainline Technology and Exception Process.

- Managed lifecycle of datacenter technologies
- Explicit decisions
- Exception process
- Long vs. short technology waves

Conservative selection of technology



Good Practice:
Software Gardening. [1]

In common:

- New expectations on outcome
- Unexpected developments
- Continuous maintenance & clean-up.

Differences:

- Waste degenerates to new life
- True nature is chaos

You need gardening & engineering.

[1] Programming is Gardening, not Engineering, Dave Thomas and Andy Hunt

Photo by Markus Winkler on Unsplash

Hypothesis:
Long Living Applications are Energy Efficient.

- Rely on conservative technology selection
- Manage technical debts [1]
- Introduce energy efficiency focus

Balance energy consumption across lifecycle and impact scopes.

[1] Sustainable Software Architecture, Carola Lilienthal

Source: https://unsplash.com/@jeremybishop

Energy Efficiency

Green Computing – Green IT * – Sustainable IT

Green Hardware

- Hardware production
  - Used resources (materials and energy)
  - Energy for supply chain
- Hardware operation
  - Green energy
  - Reuse of heat loss
  - Ecological cryogen
- Hardware lifecycle
  - Long term support
  - Software compatibility
- From client over router to datacenter

Green Software

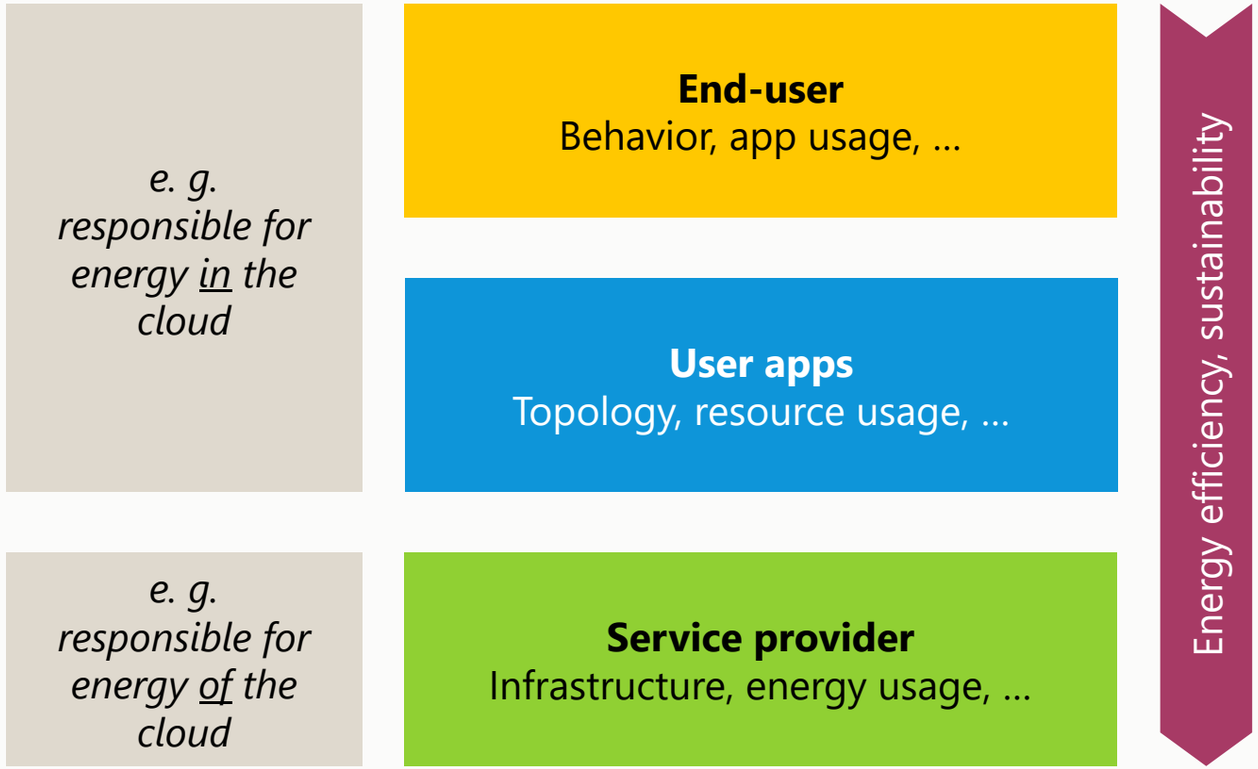
Green Code - Sustainable Software Engineering - Sustainable Software Architecture - etc.

- Development vs. usage
- Resource usage
- Hardware implications
- Hardware dimensioning

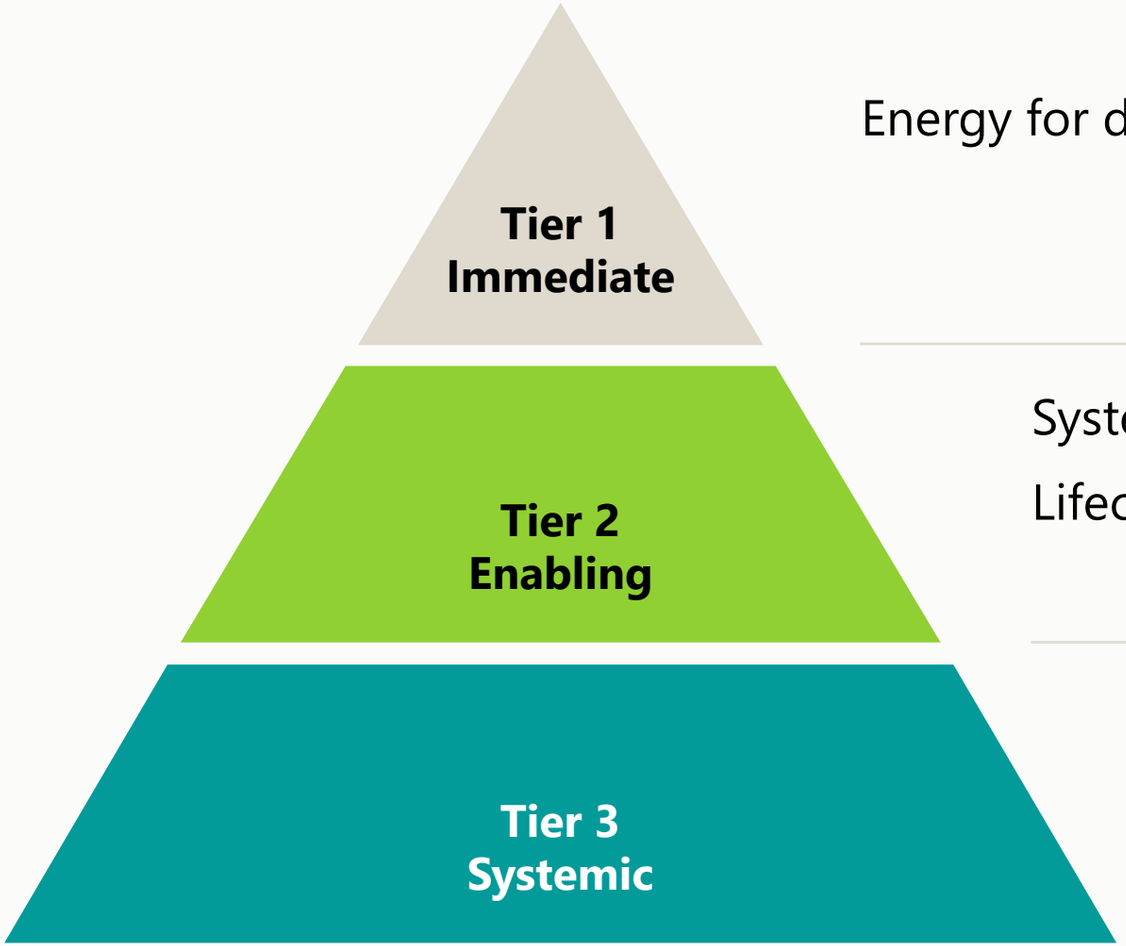
Lacking awareness and numbers

* the “Green Computing” or “Green IT” concept was introduced in 1992 when Energy Star was launched by US Environmental Protection Agency (EPA)

Layers of Responsibility



Sustainability Effects



See https://www.oeko.de/oekodoc/2318/2015-489-de.pdf

Energy for development and usage of the software

Systems that benefit from the software

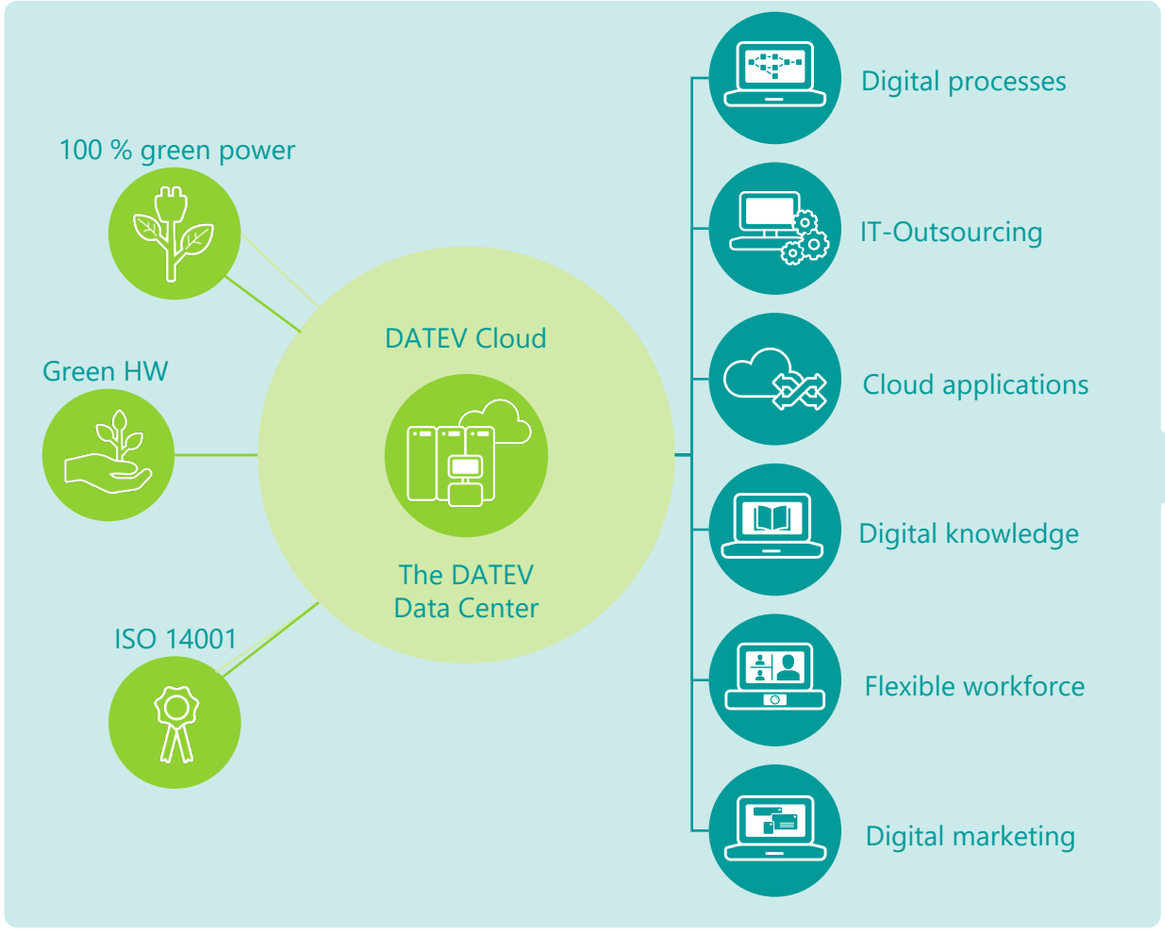
Lifecycle of the hardware and related systems

Collateral effects,
e. g. rebound effects of more efficient or cheaper goods

Energy Efficiency Forces at DATEV

- ‘Beloved’ private datacenter
- Large software portfolio: > 250 apps
- High mainframe invest
- High sensitivity to customer satisfaction

DATEV Digitization and Sustainability: Seizing Opportunities



Consuming fewer resources

Using resources more efficiently

Improving our carbon footprint sustainably

Sustainability gains for consultancies and corporations

Reducing power consumption sustainably by using the cloud

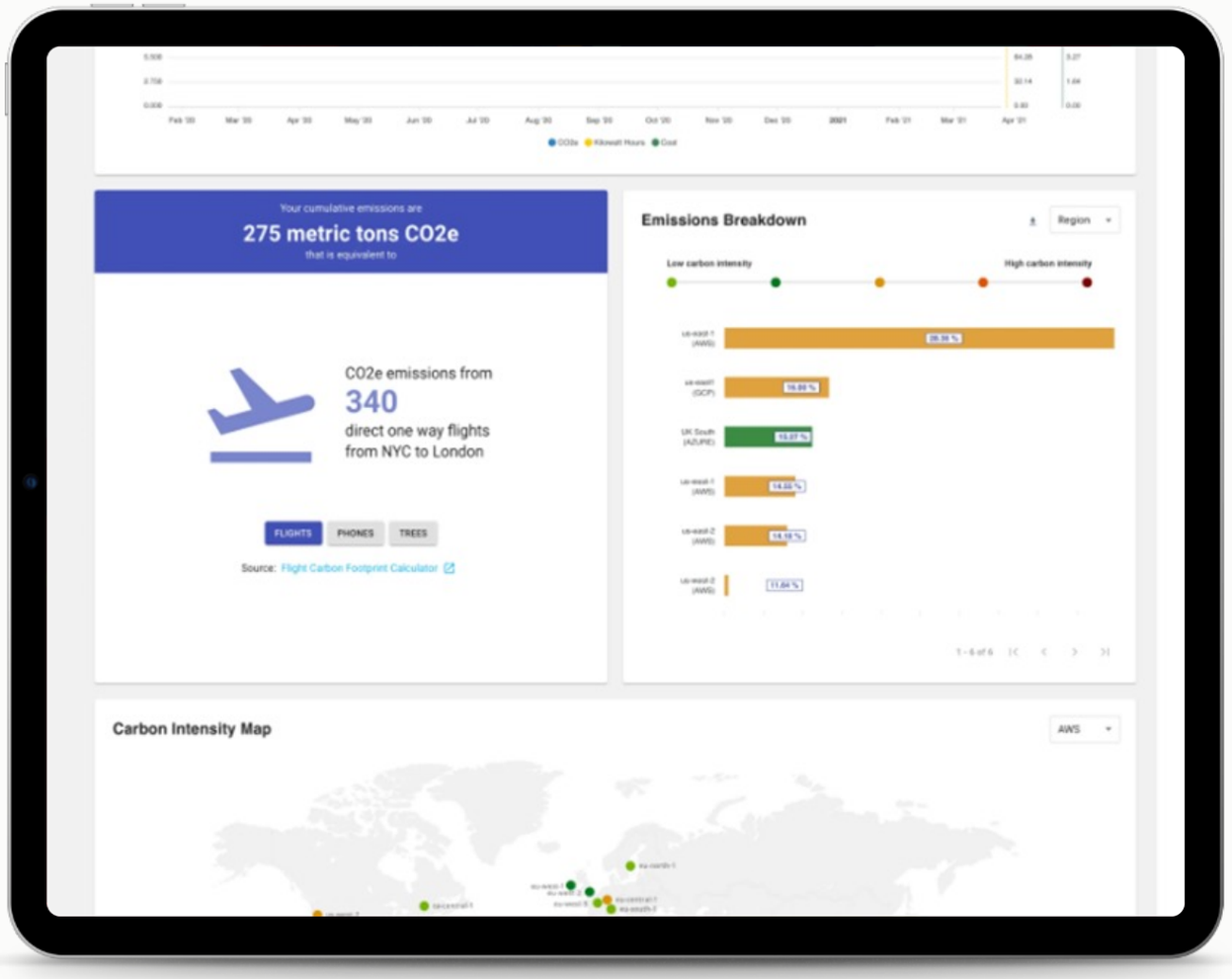
Offering sustainable business models

Developing employees sustainably

Good Practice:
Transparency and Awareness.

- Create awareness
- Create responsibility
- Act in your own part. responsibility
- Measure energy usage

‘Carbon Footprint’ tools of hyperscalers are a good start.

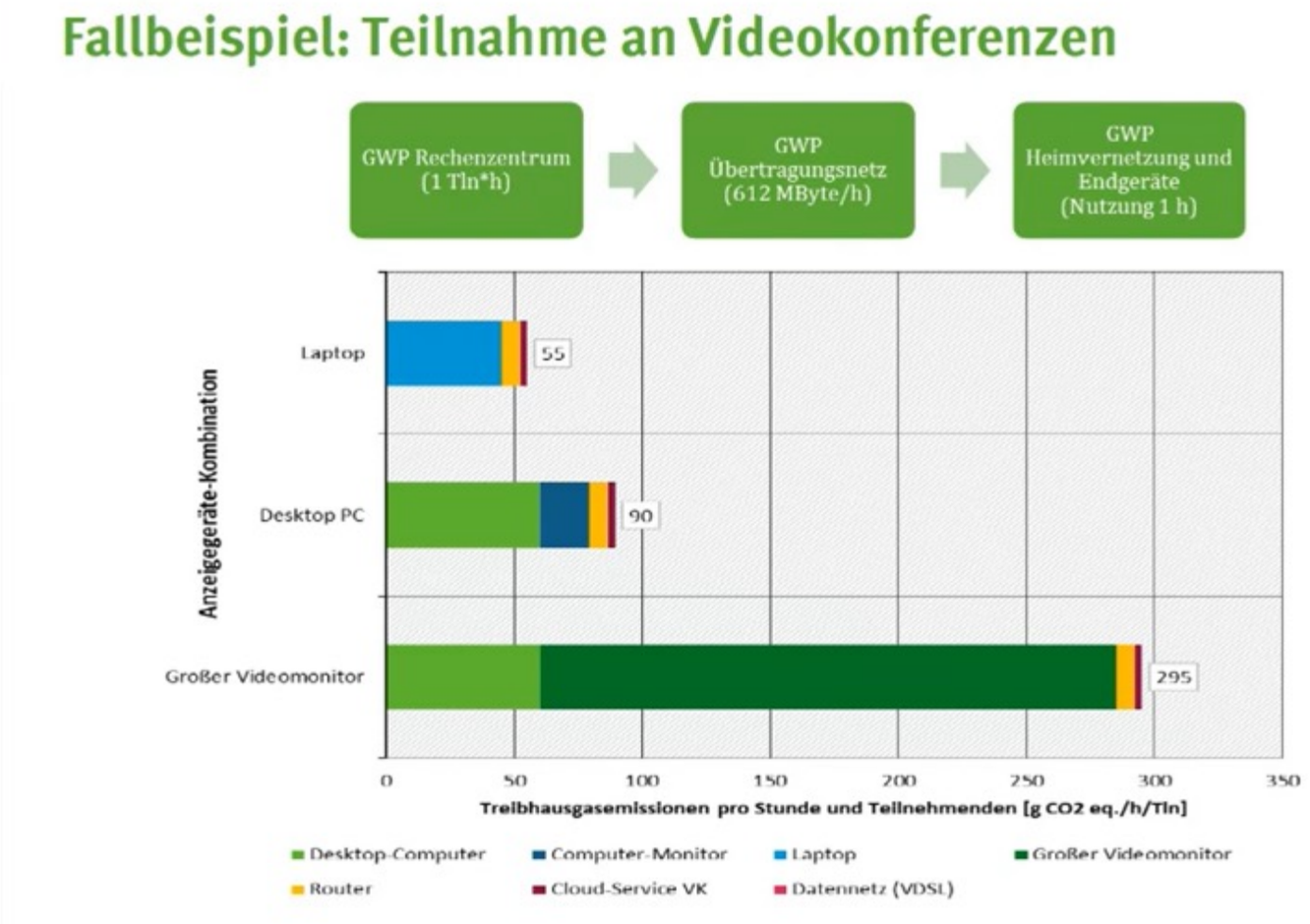


Source: https://www.cloudcarbonfootprint.org/

Good Practice:
Keep an Eye on 'Hidden Infrastructure’.

Display technology & size directly impacts energy consumption. [1]

Include client in end-to-end energy consumption.



Source: https://www.youtube.com/watch?v=BEo7oFacxz0

[1] OOP 2022 Keynote, Marina Köhn, Umweltbundesamt

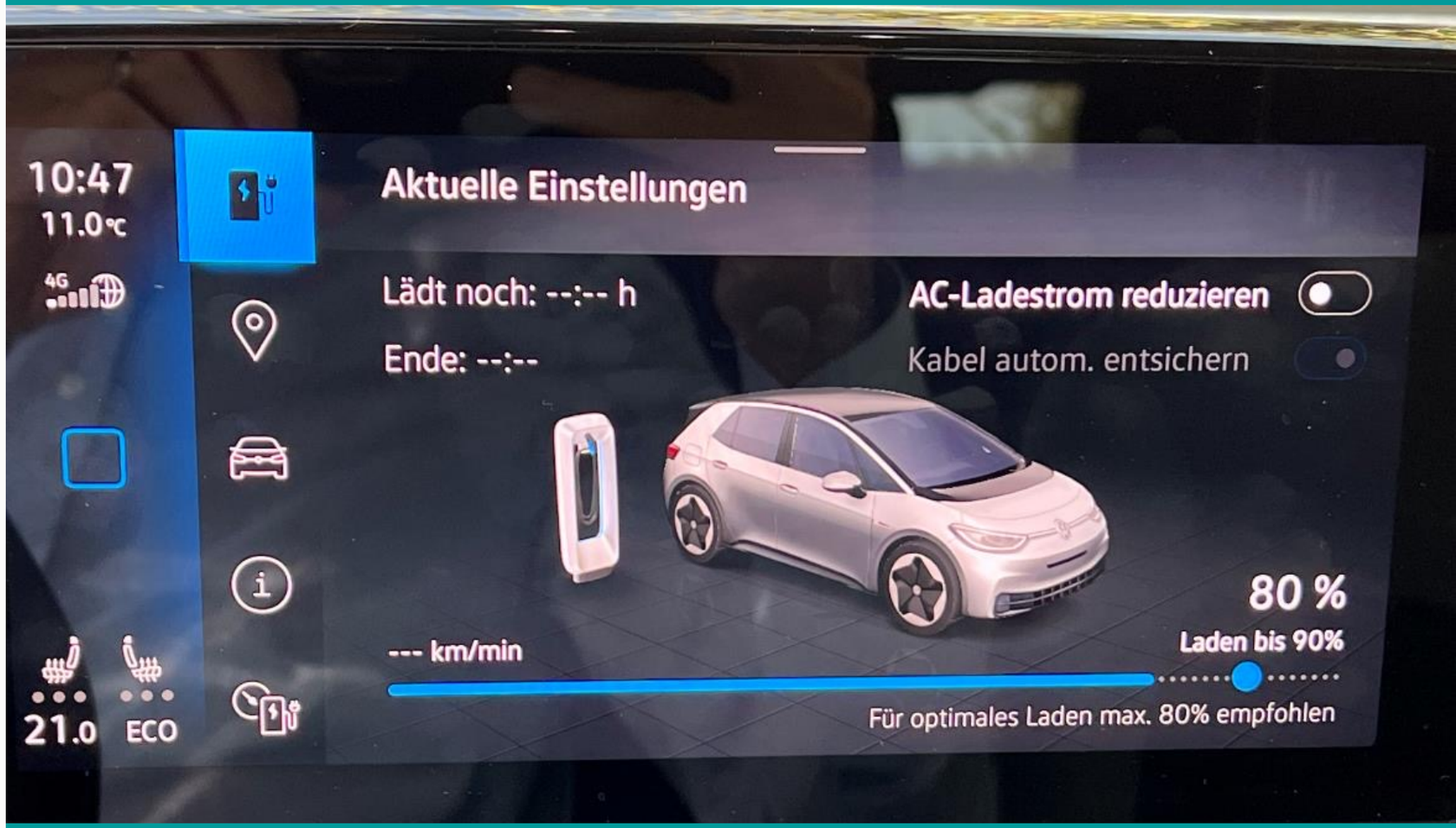
Good Practice: Involve the User.

Inform, warn and ask the user when expensive operations are going to be executed.

Examples:

- VW ID.3 charging UI warning to load vehicle not more than 80 %
- ‘Search dialogs’ and their search scope and criteria

Visualize climate cost.



Good Practice:
Sustainable Architecture Decisions.

- Add ‘Energy Efficiency’ to the quality scenarios
- Include energy efficiency in trade-off analysis

Remember: “You can’t improve what you don’t measure.” [1]

[1] Peter Drucker

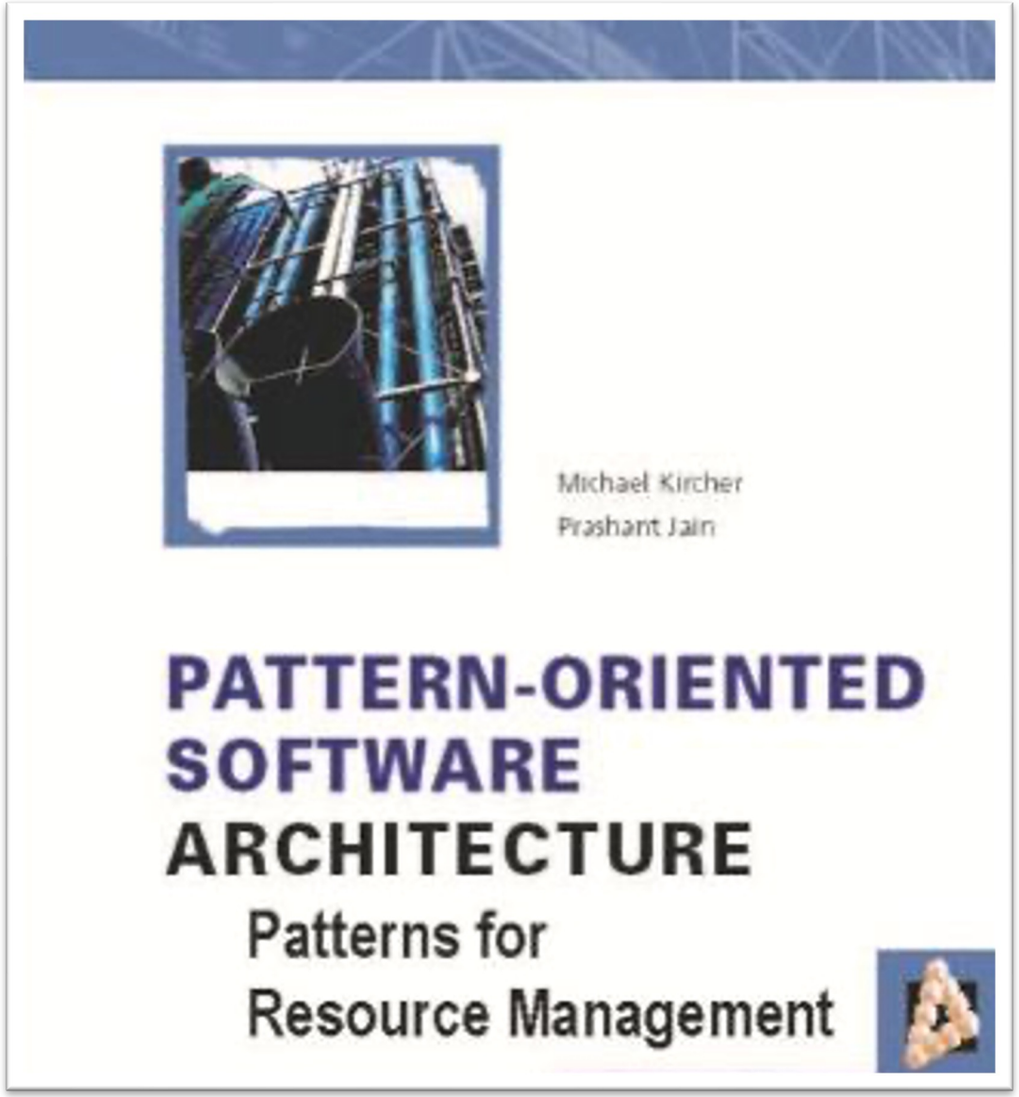

Source: https://unsplash.com/@victor_g

Good Practice:
Efficient Code.

- Resource management patterns [1] like Caching and Lazy Acquisition
- Performance profiling
- Identify most frequent paths
- Front-end optimizations: content size and refresh rate
- Asynchronous programming

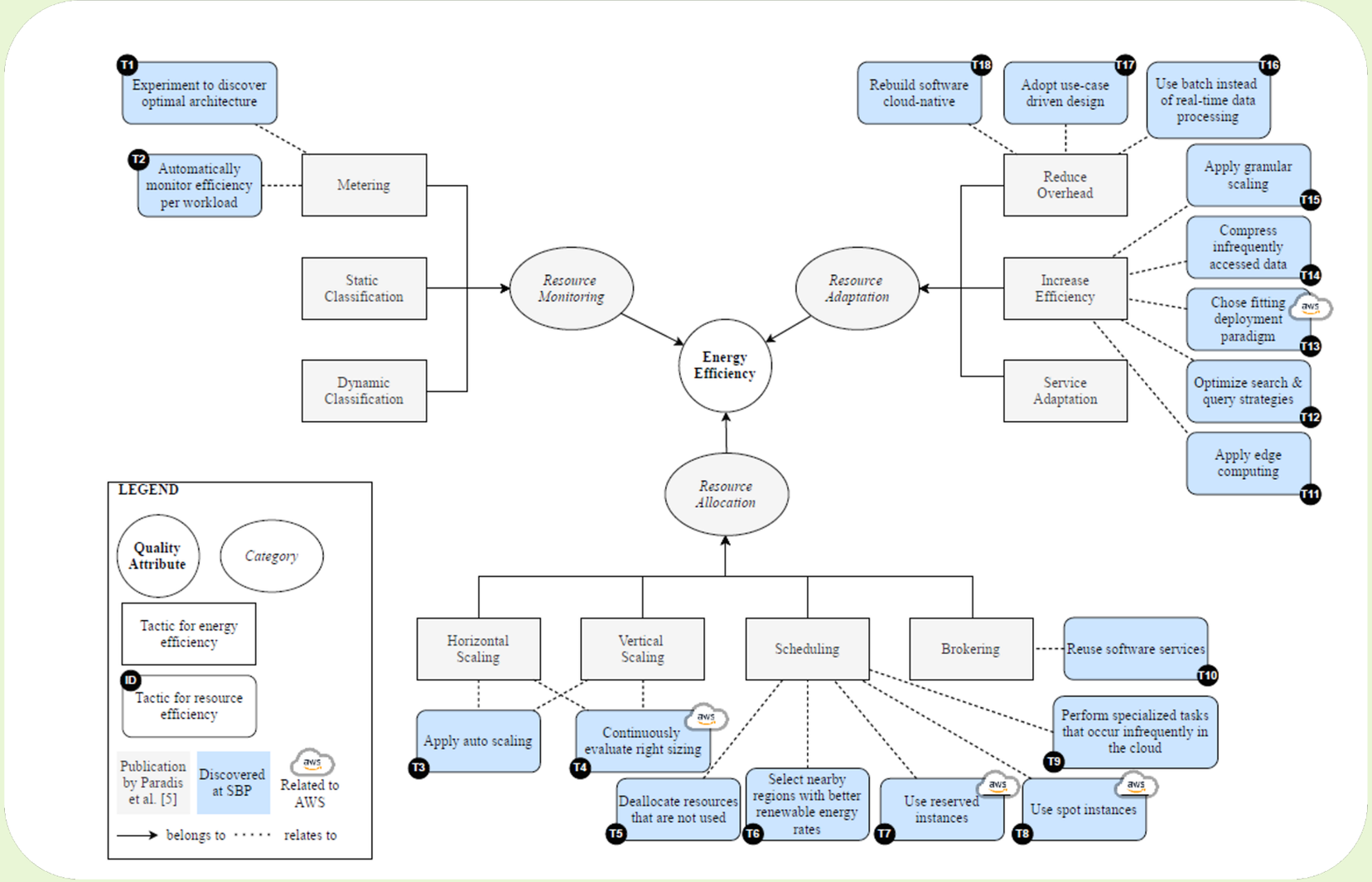
Hypotheses:

- Serverless programming is more sustainable?
- Performant software is more sustainable?



[1] POSA 3: Patterns for Resource Management, Wiley, 2004

Good practice:
Architectural Tactics […] in the Public Cloud.



See Architectural Tactics to Optimize Software for Energy Efficiency in the Public Cloud by Sophie Vos, Patricia Lago, et al

Good Practice: Organizational Measures.

- Establish a ‘Sustainability Organization’ with the role of ‘Sustainability Officers’
- Allow for slack to allow for grass-roots initiatives
- Distinguish between incremental and ‘net new’ initiatives
- Talk numbers: holistic metrics
- Green tech research

~~Hypothesis~~/Good Practice:
Avoid Blockchain Except for Self-Sovereign Identities.

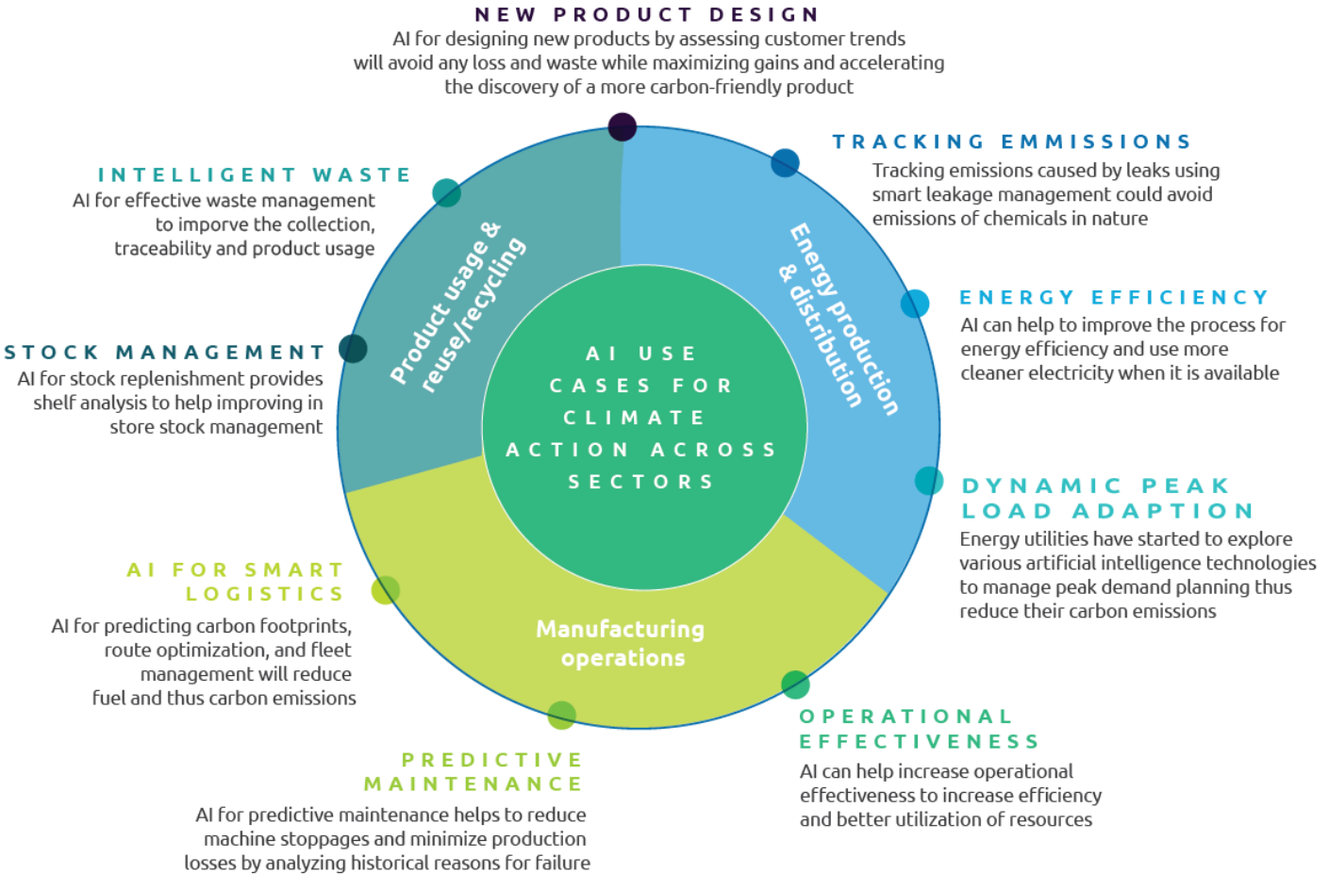
- Starting point for Blockchain: missing trust
- Key pattern: proof of work
- Trade-off between non-repudiation vs. almost all other qualities
- Energy consumption unreasonably wasteful
- Rectification only for self-sovereign identities
- Other blockchain use cases reg. purpose or ecological footprint questionable.

See Blockchain sicher gestalten - Konzepte, Anforderungen, Bewertungen, BSI and Confronting the Carbon-footprint Challenge of Blockchain

Hypothesis:
Artificial Intelligence is Critical Regarding Energy Efficiency.

- Algorithm search and model training are very energy intensive
- Danger of rebound effects by using AI to beat the climate crises [1]

Balance sustainability effects on tiers against each other.



Source: https://www.capgemini.com/it-it/service/perform-ai/sustainable-ai/

[1] 'GreenTech Made in Germany', MHP, July 2022

Hypothesis:
The Mainframe is More Efficient than x86 Cluster.

- Mainframes are designed to run at > 90 % CPU utilization, while most x86-based server farms operate at less than 60 %
- IBM: “IBM LinuxONE Emperor 4 servers can reduce energy consumption by 75% and space by 50% for the same workloads on x86 servers.” [1]

Can I make use of this?

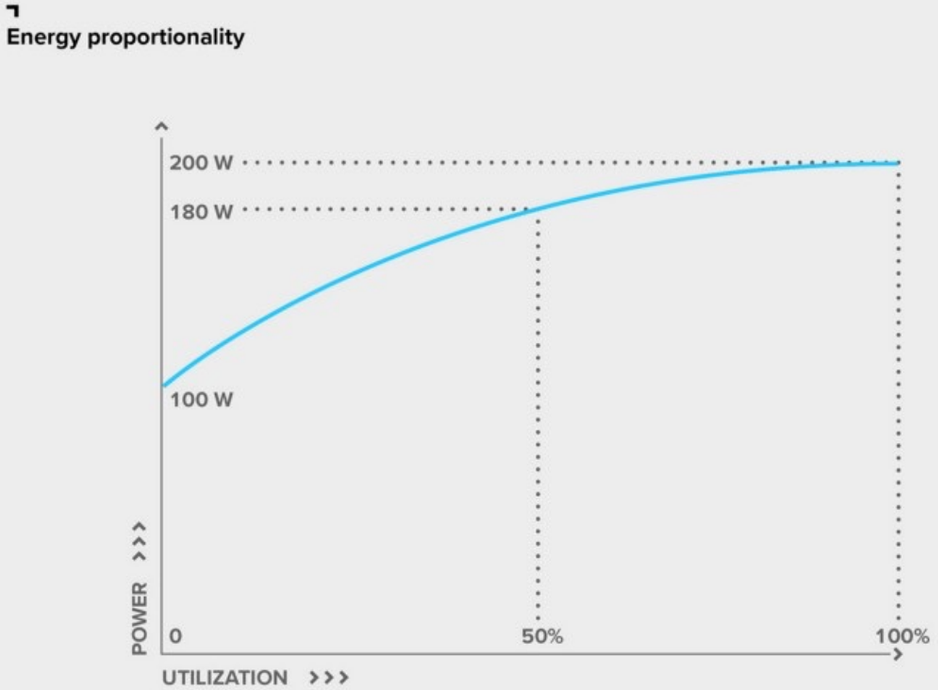
How does the event driven load influence the mainframe I/O optimization?

[1] IBM website

Source: https://www.ibm.com/topics/mainframe/

Hypothesis: Hyperscalers are More Energy Efficient than Private Clouds.

- Higher CPU utilization in public clouds
- “Running applications in AWS Cloud can help to lower carbon footprint by 88%” [1] … when compared to enterprise data center.
- “For example … [public clouds] … run at approximately 65% utilization, while on-premise data centers reach only 12–18%.” [2]
- Compare ”Dark Tactics”: “Overconsumption”, ”Superfluous usage” … of data traffic [3]



Source: https://medium.com/gft-engineering/greencoding-the-keys-to-lower-emissions-67ab55028d6

[1] ‘The Carbon Reduction Opportunity of Moving to Amazon Web Services’ by 451 Research, October 2019

[2] Green Coding: The keys to lower emissions, GFT

[3] Awesome and Dark Tactics, Digital Sustainability Center

Hypothesis: Hyperscalers Adopt CPU Architectures.

First indications that hyperscalers consider change:

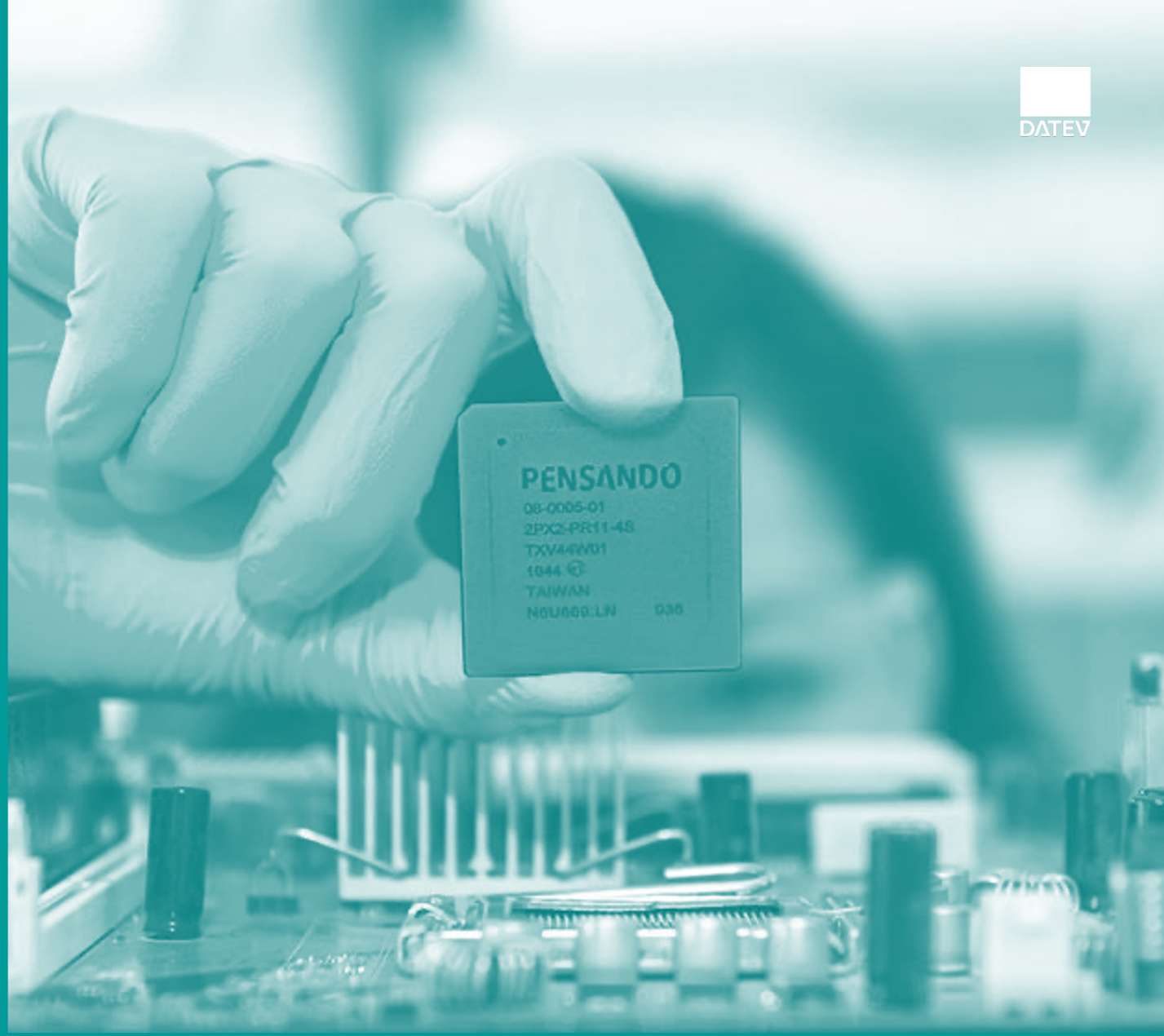
- ARM CPUs are expected to be more resource efficient in general [1]

  E. g. AWS Graviton CPUs

- Data Processing Units (DPU) offload network virtualization overhead from CPUs

  e. g. AMD Pensando

[1] ARM vs x86, RedHat and ARM Chips Gaining in Data Centers, ...



Source: https://www.amd.com/de/solutions/infrastructure-acceleration

Hypothesis:
Programming Language Makes ‘Some’ Difference.

- Multi dimensional optimization
  - Energy, time, memory
  - Skills, expressiveness, support
- A faster language is not always the most energy efficient [1]

Major differences only regarding interpretation vs. compilation

Table 5. Pareto optimal sets for different combination of objectives.

Time & Memory	Energy & Time	Energy & Memory	Energy & Time & Memory
C • Pascal • Go	C	C • Pascal	C • Pascal • Go
Rust • C++ • Fortran	Rust	Rust • C++ • Fortran • Go	Rust • C++ • Fortran
Ada	C++	Ada	Ada
Java • Chapel • Lisp • Ocaml	Ada	Java • Chapel • Lisp	Java • Chapel • Lisp • Ocaml
Haskell • C#	Java	OCaml • Swift • Haskell	Swift • Haskell • C#
Swift • PHP	Pascal • Chapel	C# • PHP	Dart • F# • Racket • Hack • PHP
F# • Racket • Hack • Python	Lisp • Ocaml • Go	Dart • F# • Racket • Hack • Python	JavaScript • Ruby • Python
JavaScript • Ruby	Fortran • Haskell • C#	JavaScript • Ruby	TypeScript • Erlang
Dart • TypeScript • Erlang	Swift	TypeScript	Lua • JRuby • Perl
JRuby • Perl	Dart • F#	Erlang • Lua • Perl	
Lua	JavaScript	JRuby	
	Racket		
	TypeScript • Hack		
	PHP		
	Erlang		
	Lua • JRuby		
	Ruby		

Source: https://greenlab.di.uminho.pt/wp-content/uploads/2017/09/paperSLE.pdf

[1] Energy Efficiency across Programming Languages, Pereira et al

‘Blauer Engel’ for Software Products

- Label for “resource- and energy-efficient software products” [1]
- First certification in 2022
- Focus is desktop SW
- 2024 expected to cover also cloud SW
- RAL and Umweltbundesamt need to enforce and advertise more

Until now, little to no impact.

[1] Ressourcen- und energieeffiziente Softwareprodukte

Source: https://www.blauer-engel.de/

My Personal Summary

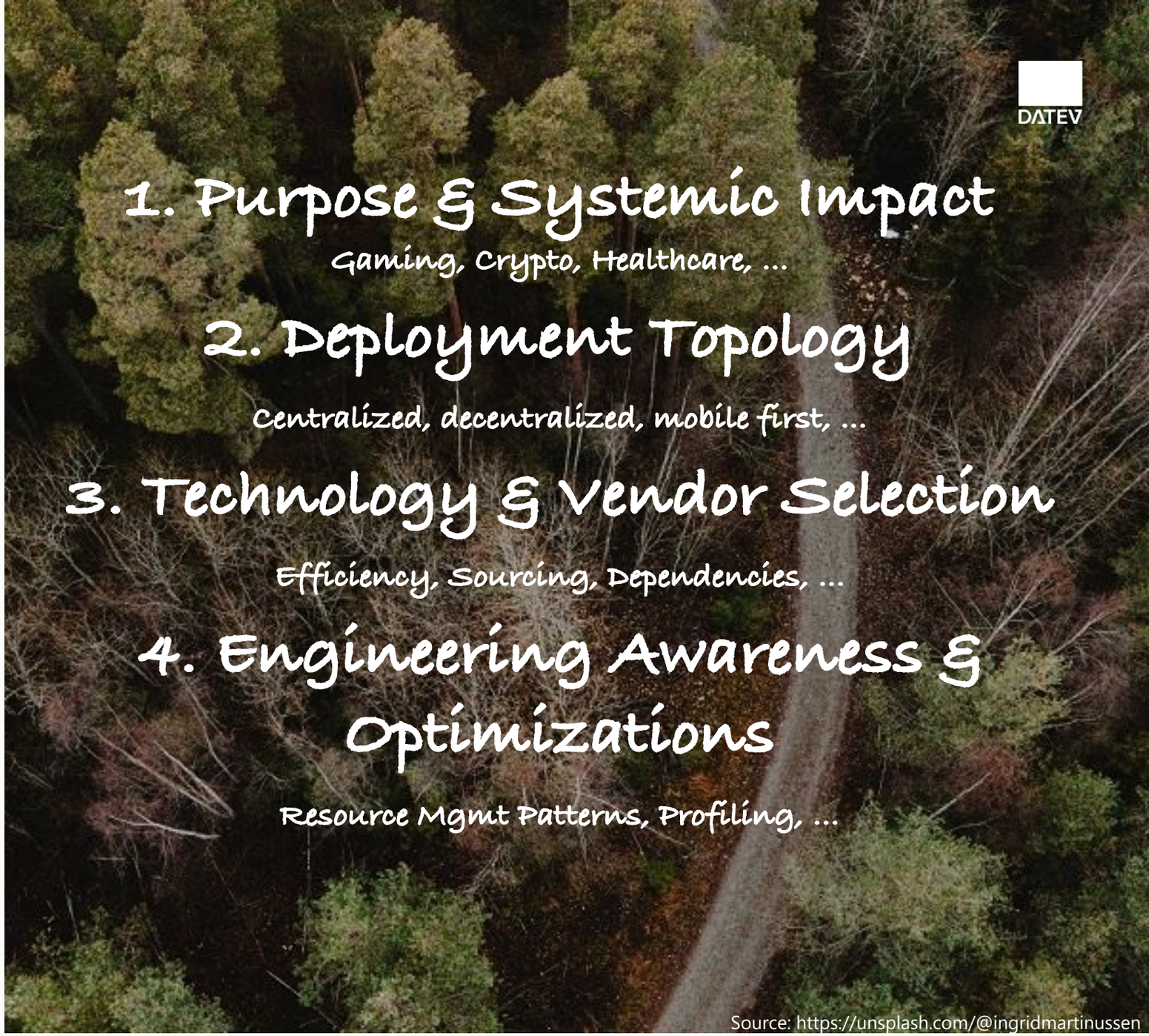
Start top down, from global to local:

- Systemic > enabling > direct impact [1]
- App purpose & user interaction
- Coarse grained deployment of compute and storage
- Optimization of computes, storages, transfers

Strong correlation w/ performance and cost optimizations. [2]

[1] Nachhaltige Software (oeko.de)

[2] Sustainable Software Engineering Microsoft.com



Source: https://unsplash.com/@ingridmartinussen

Interesting Organizations and Persons

- Umweltbundesamt Software u. Umwelt, e. g. Marina Köhn
- Vrije Universiteit Amsterdam, e. g. Prof. Patricia Lago, esp. Awesome and Dark Tactics
- Green Software Foundation, esp. Awesome Green Software
- Sustainable Digital Infrastructure Alliance (SDIA), e. g. Project SoftAWERE
- Umwelt Campus Birkenfeld, e. g. Prof. Stefan Naumann
- Blauer Engel für ressourcen- und energieeffiziente Software Produkte

Shaping the future. Together.