A Hitchhiker's Guide to Resource Efficient Software

It‘s like sustainability – but more practical…

Speaker

Hendrik Lösch
Management Consultant

hendrik.loesch@zeiss.com
hendrik-loesch.de
slideshare.net/HendrikLsch1

Other Sources

Check out the Bitkom Guide

A three-part guide written in German that provides an overview and various recommendations for action for the respective target groups.

Management Summary

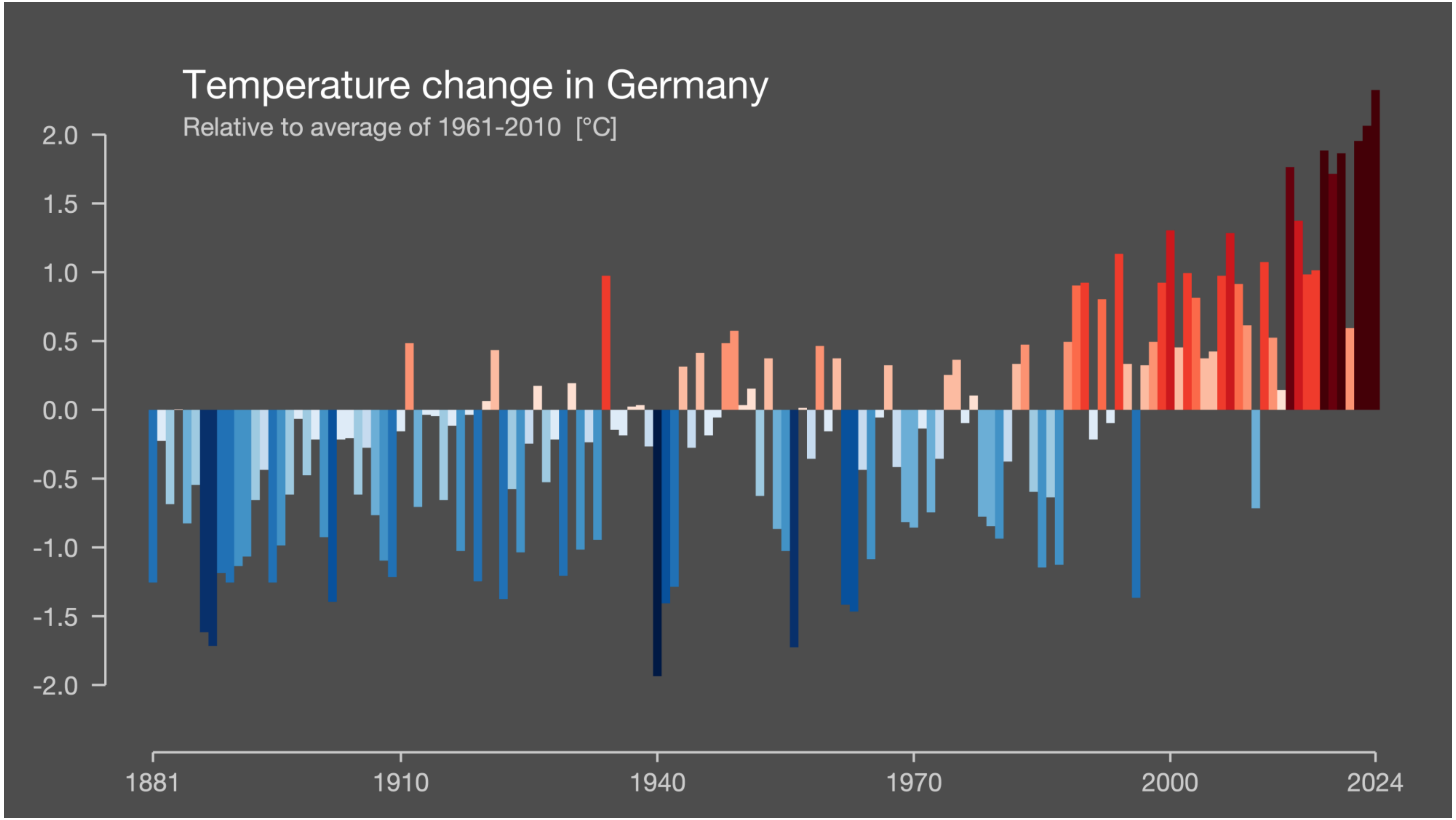
Requirements and Design

Implementation and Operation

https://www.bitkom.org/Themen/Maerkte-Technologien/Softwareentwicklung/Software-Engineering/Ressourceneffizienz-im-Software-Lifecycle

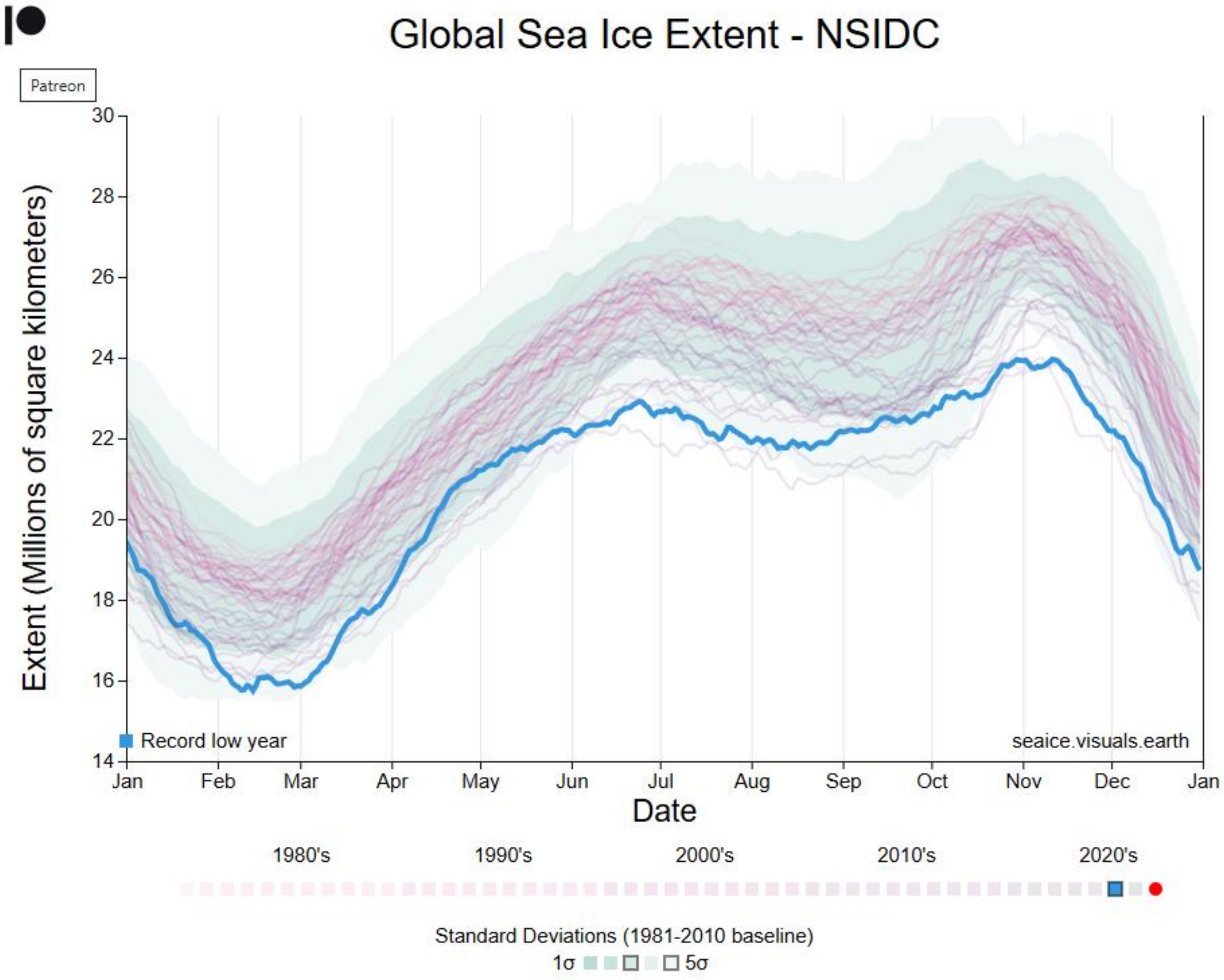
State of (Climate) Crisis

Temperature over time



https://showyourstripes.info/c/europe/germany/all

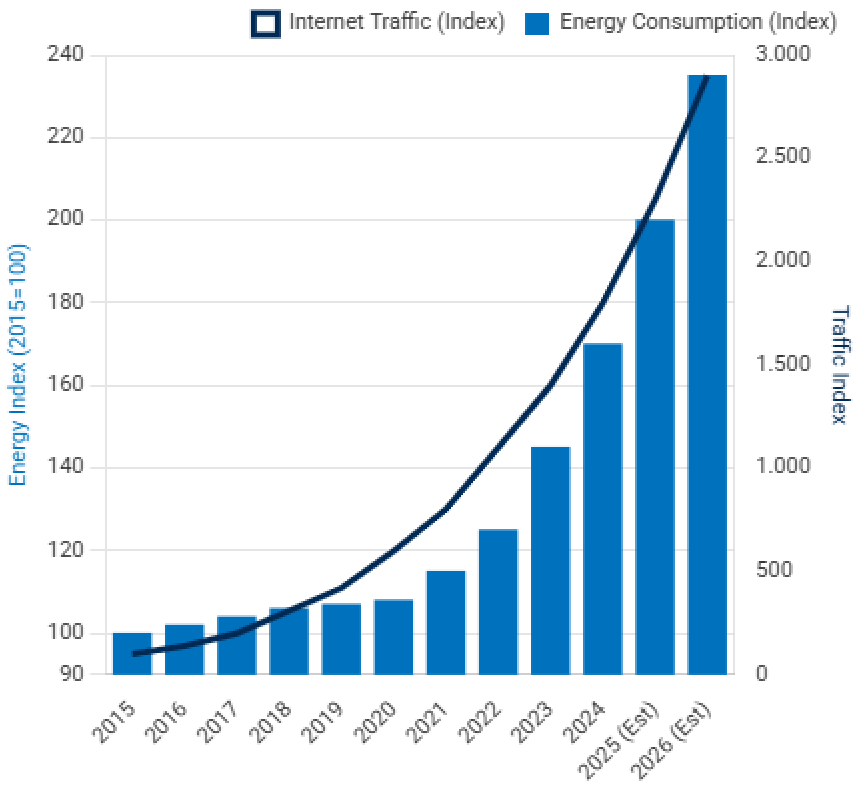
State of (Climate) Crisis
Global Sea Ice



Sustainability
Results from a Gemini deep search…

DECOUPLING & RE-COUPLING

Line: Data Traffic (Exponential Growth)   Bar: Energy (Plateau -> Rise)



GLOBAL ELECTRICITY SHARE

~2.0%
CURRENT (2024)

4.0% +
PROJECTED (2030)

TECHNICAL ANALYSIS

The chart on the left clearly shows: Until 2020, efficiency gains (PUE) kept energy demand (bars) flat, despite exploding traffic (line).

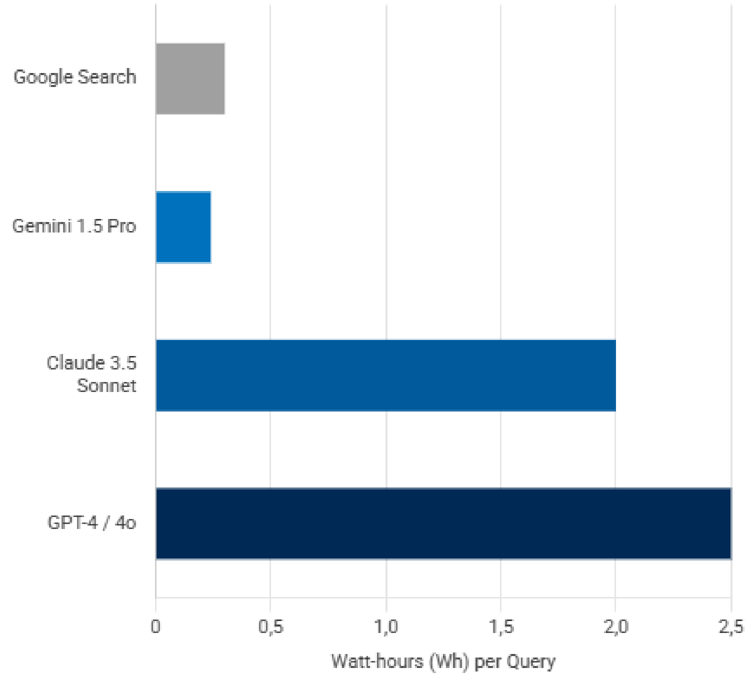
The Turning Point (2022): With the advent of massive generative AI, energy demand is now rising synchronously with traffic again. The era of "free" efficiency is over.

Sustainability
Results from a Gemini deep search…

ZEISS

ENERGY INTENSITY PER QUERY (INFERENCE)

Unit: Watt-hours (Wh) per Prompt | Comparison 2025



COMPARISON: SEARCH ENGINES VS. AI

V3 — **GPT-3 (LEGACY)**
175B Parameters Dense
~502 tCO2e

V4 — **GPT-4 (CURRENT)**
MoE Architecture (Massive)
~13,000 tCO2e
FACTOR 25X

DATA_NOTE: Google (Gemini) uses TPUs v5p, which are more efficient per watt than standard GPUs, lowering inference consumption despite model size.

Generate 1 Image

Model: Stable Diffusion XL

EQUIVALENT TO
100% Smartphone Charge

1000 ChatGPT Queries

Daily Power-User Usage

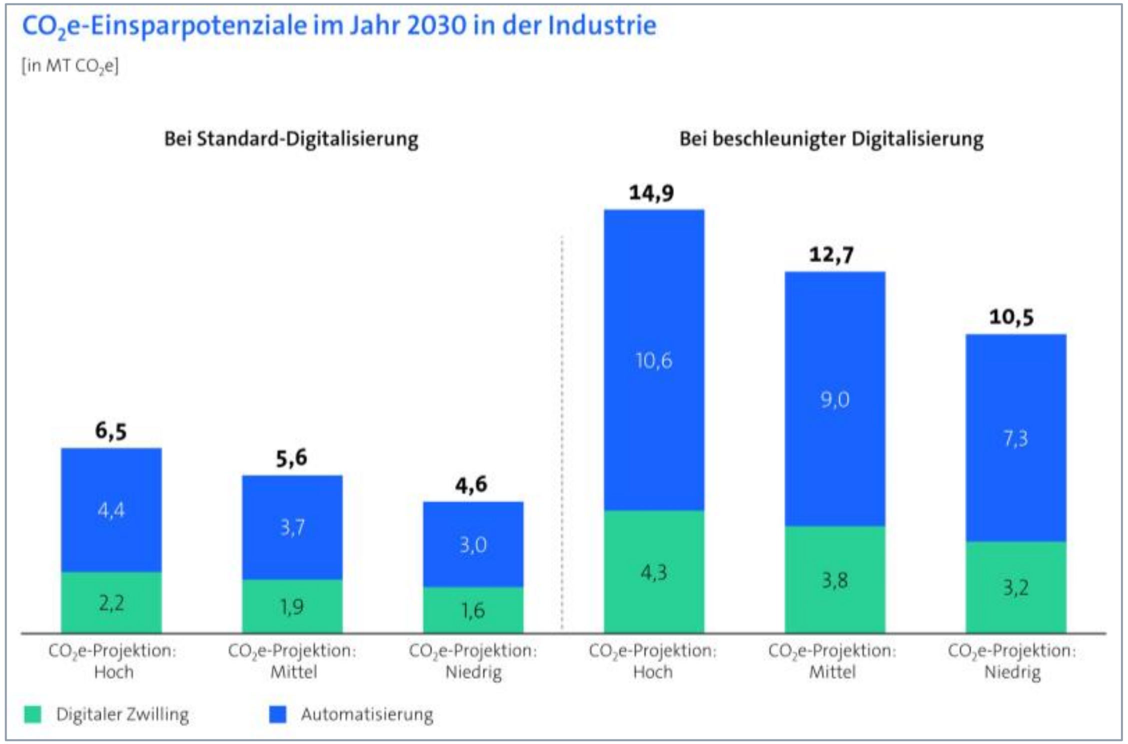
EQUIVALENT TO
Lightbulb burning for 2 Days

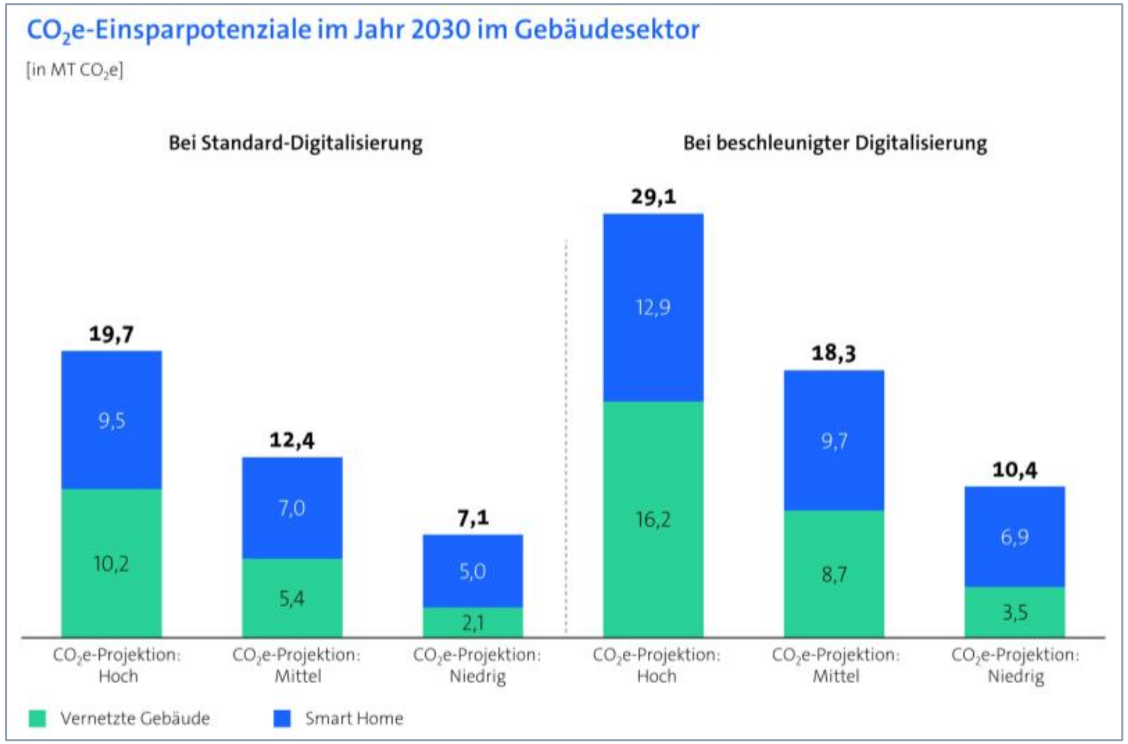
Training GPT-4

Energy & CO2e (Est.)

EQUIVALENT TO
~3000 Transatlantic Flights

Sustainability
Sustainability by Software

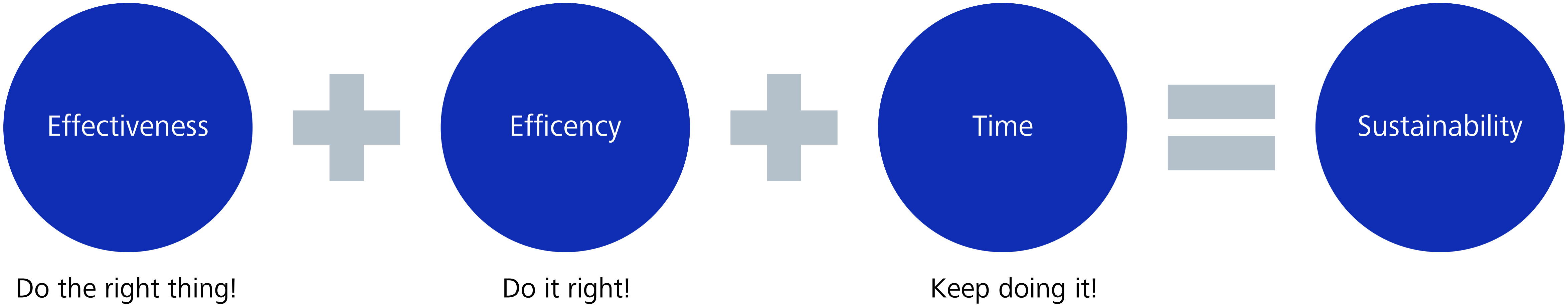




Bitkom is an industry association with the overarching goal is to make Germany a leading digital location, drive forward the digital transformation of the German economy and administration, strengthen digital sovereignty and achieve broad social participation in digital developments.

Sustainability
Sustainability?

"Sustainability is the practice of **ensuring that the consumption of resources does not exceed the** Earth's **ability to regenerate them**, thereby maintaining ecological balance for future generations."



Sustainability
Different perspectives

Sustainability by Software

leveraging software to promote sustainable practices and reduce environmental impact, which can help organizations reduce their environmental footprint.

Sustainability in Software

using sustainable practices in the design and development of software, to reduce the environmental impact of software development and promote responsible business practices.

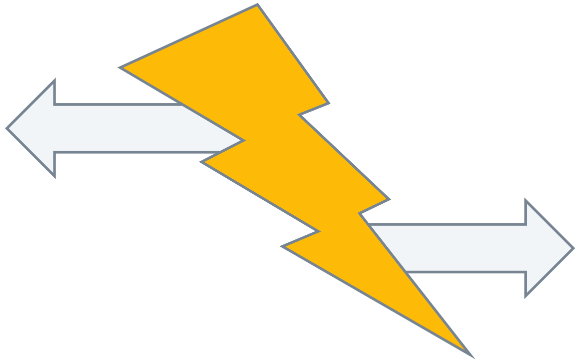
Bitkom Studie „Klimaeffekt der Digitalisierung 2.0"

Sustainability
The two stereotypes…

Hard fact: “Ecological Sustainability” is not a primary business driver!

"I work on sustainability because I genuinely believe it's my responsibility to help protect the planet for future generations—it's what drives me every day."

generated with chat gpt
to proof a point…



“To be honest, I'm working on sustainability because my manager assigned me to the project, and it's part of our company’s current priorities."

generated with chat gpt
to proof a point…

Sustainability

Well, CO2 emissions go up, I guess…

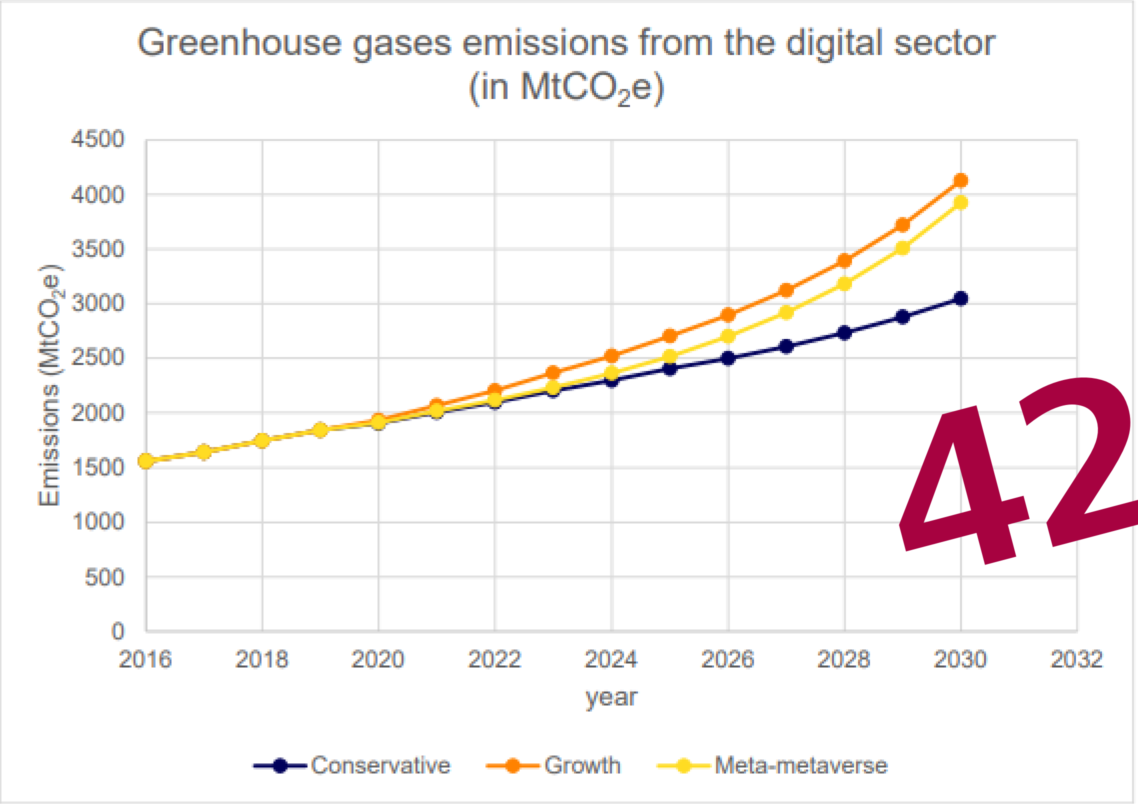


Figure 4 - 2016-2030 trends in greenhouse gas emissions from the digital sector: 2016-2025 Conservative and Growth scenarios (The Shift Project, 2021) extended to 2030 and Meta-metaverse scenario Source: The Shift Project, for the purposes of this report

42?!?

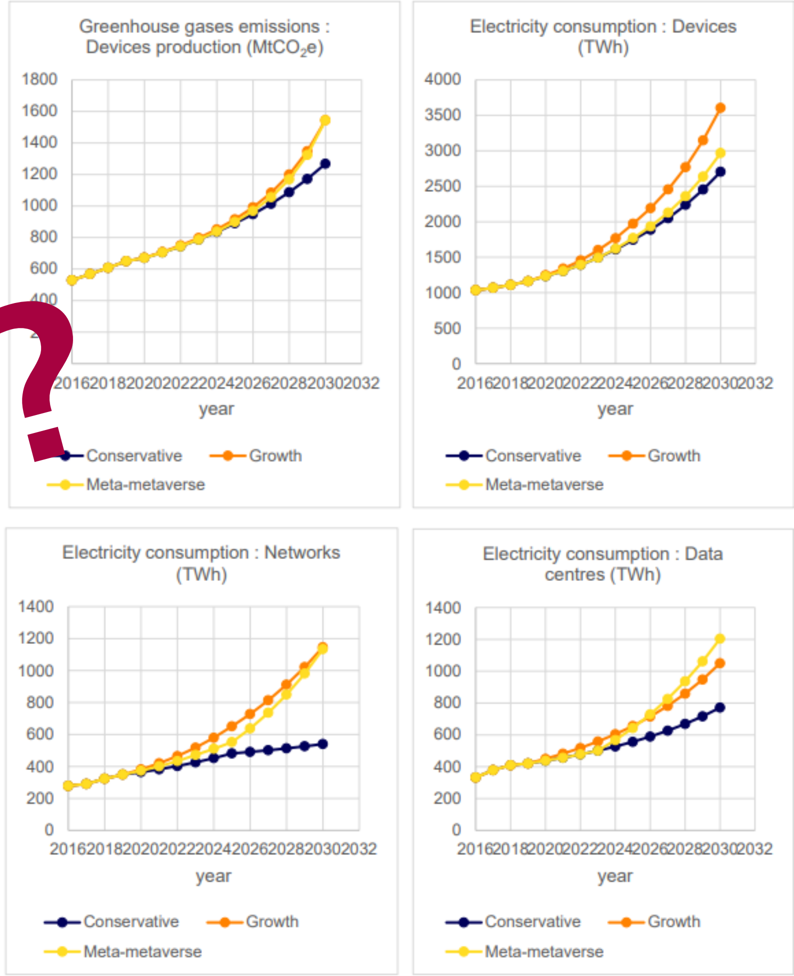


Figure 5 - 2016-2030 trends in greenhouse gas emissions from device production and electricity consumption for each of the 3 components: Extended 2016-2025 Conservative and Growth scenarios (The Shift Project, 2021) and Meta-metaverse scenario. Source: The Shift Project, for the purpose of this report

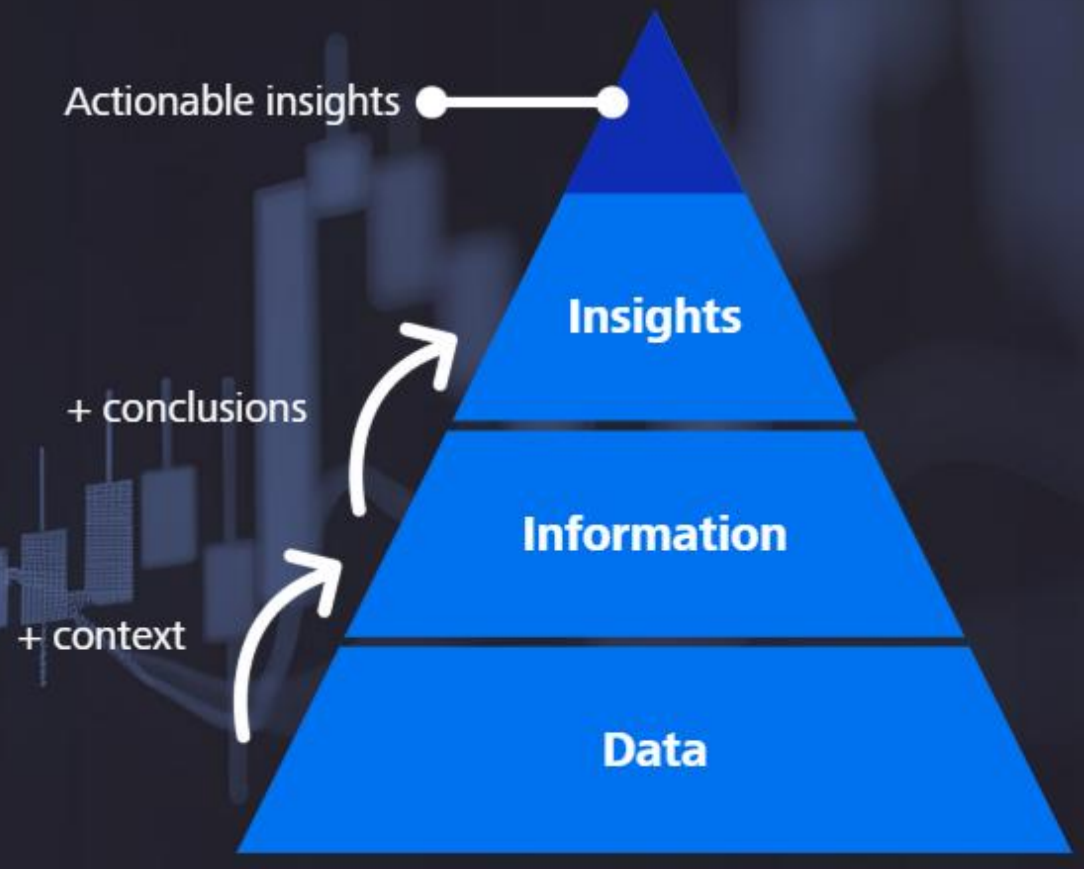
Source: https://theshiftproject.org/app/uploads/2025/02/The-Shift-Project-What-virtual-worlds-for-a-sustainable-real-world-Final-report-March-2024_ENG.pdf

Sustainability
Data vs. Information

Data refers to raw, unprocessed facts and figures, such as numbers, text, and images.

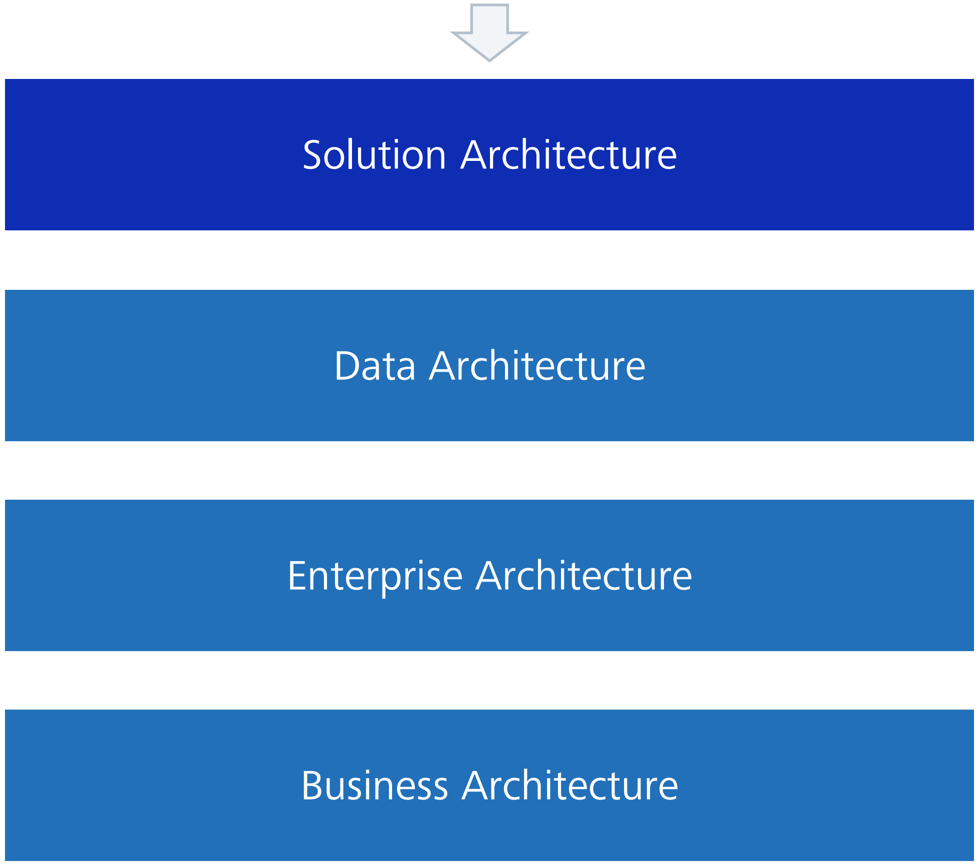
When data is organized and contextualized, it becomes **information**.

Insights arise from analyzing information, revealing patterns and trends that inform decision-making. It is highly critical for any business to connect an insight to an action, hence making it **actionable insights**.



Sustainability
The different views

Many proposals from the sustainability community are aimed at this



Sustainability
The different views

Many proposals from the sustainability community are aimed at this

- Solution Architecture
- Data Architecture
- Enterprise Architecture
- Business Architecture

Sustainability
ESG Approach for Sustainability Reporting

An approach to investing that recommends taking environmental issues, social issues and governance issues into account when deciding which companies to invest in.

Environmental

Examines the impact of the company on nature and the environment.

- **Energy Efficiency**
- Water Management
- **Greenhouse Gas Emissions**
- Biodiversity Loss
- Deforestation / Reforestation
- …

Social

Examines how a company interacts with employees, suppliers, customers and society.

- Employee safety and health
- Working Conditions
- Diversity
- Equity and Inclusion
- Conflicts and Humanitarian Crisis
- …

Governance

Includes a company's internal control systems, practices and procedures.

- Corporate Governance
- Preventing Bribery and Corruption
- Diversity of Board of Directors
- Executive Compensation
- Cybersecurity
- …

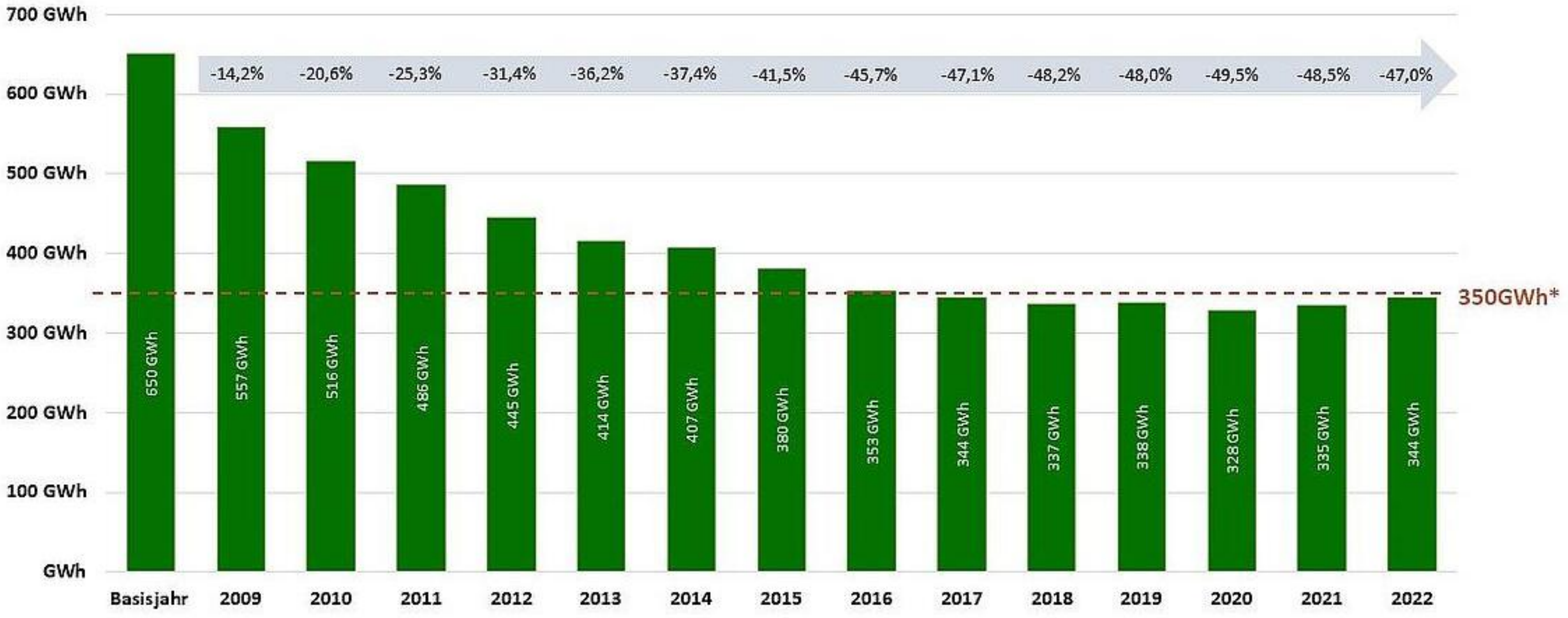
Sustainability
ProGress III – Topic Areas

Germany's resource efficiency program describes how the German government is promoting the more efficient use of resources.

Raw Material Supply	Raw Material Strategy	Renewable Resources	Sustainability and Transparency
Production	Efficiency Consulting	Efficient Production	Product Design
Consumption	Awareness Building	Trade + Consumers	Certification Systems
Construction	Infrastructure	Construction, Development, Renovation	Labeling of Construction Products
ICT	Efficient Devices	Efficient Software	Public Procurement
Cross-cutting	Research and Innovation	Education	Legal Framework

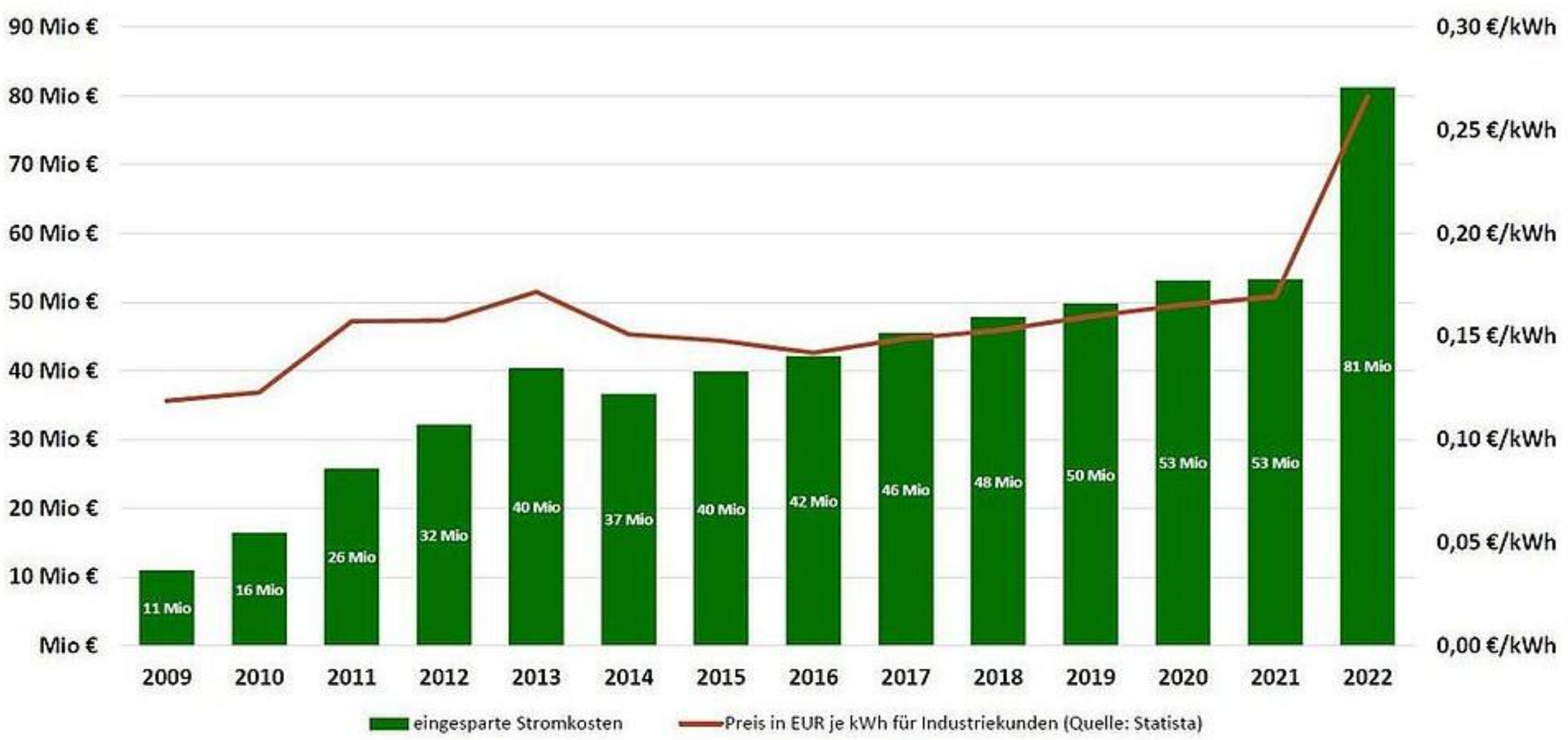
Sustainability

Green IT Initiative of the Federal Government



Sustainability

Green IT Initiative of the Federal Government



*Quelle: statista.com, Industriestrompreise (inklusive Stromsteuer) in Deutschland in den Jahren 1998 bis 2023; https://de.statista.com/statistik/daten/studie/252029/umfrage/industriestrompreise-inkl-stromsteuer-in-deutschland/

https://www.bmuv.de/themen/nachhaltigkeit-digitalisierung/digitalisierung/green-it-initiative

Sustainability
Changes to CSRD

Current Thresholds (as of 2025)

Rule: 2 out of 3 criteria must be met.

A company is required to report if it meets at least **two** of the following three criteria:

Criteria

Balance Sheet Total: > €25 million

Net Turnover: > €50 million

Number of Employees: > 250

Applies to financial years starting on or after January 1, 2024.

Sustainability
Corporate Sustainability Due Diligence Directive (CSDDD)

Key Dates	**Published in EU Official Journal:** July 5, 2024 **Entered into force:** July 25, 2024 **National Law Transposition Deadline:** July 26, 2026
General Scope	The directive applies to large EU and non-EU companies that meet the final thresholds: - Over 1,000 employees and a worldwide net turnover of more than €450 million. - For non-EU companies, it also applies if they generate a net turnover of over €450 million within the EU.
Phased Application (Timeline)	The directive will be phased in, applying to companies based on the following timeline:
From 2027:	Companies with > 5,000 employees and > €1.5 billion turnover.
From 2028:	Companies with > 3,000 employees and > €900 million turnover.
From 2029:	Companies with > 1,000 employees and > €450 million turnover.

Sustainability
News from European Banking Authority

Core Principle	Systematically integrate ESG risks, especially climate-related ones, into risk management and strategic planning.
Key Tool	Use **Scenario Analysis** to assess the impact of ESG factors on the business and risk profile.
Main Applications	**1. Short/Mid-Term Climate Stress Testing:** - To test financial resilience and ensure capital/liquidity adequacy against severe shocks. **2. Long-Term Resilience Analysis:** - To challenge the business model's viability against long-term transition pathways (e.g., EU 2050 climate neutrality).
Methodology	Define plausible scenarios and identify "transmission channels" that translate ESG risks into traditional financial risks (credit, market, etc.).
Proportionality	Requirements are scaled based on the institution's size and complexity, with simplified rules for smaller banks.
Implementation	The guidelines will be phased in, starting from January 2026, complementing the broader ESG risk management framework.

Consultation paper on draft Guidelines on ESG scenario analysis.pdf

The Role of Software Development

Sustainable Programming Languages?

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

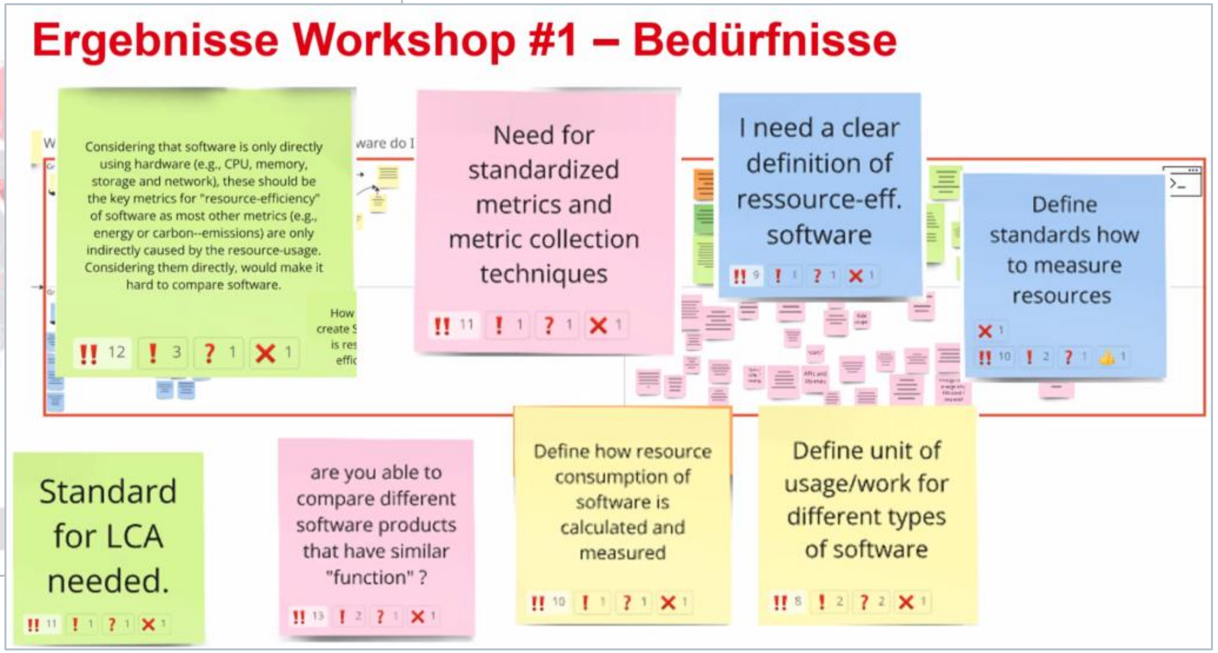
Table 5. Pareto optimal sets for different combination of objectives.

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		

The Role of Software Development

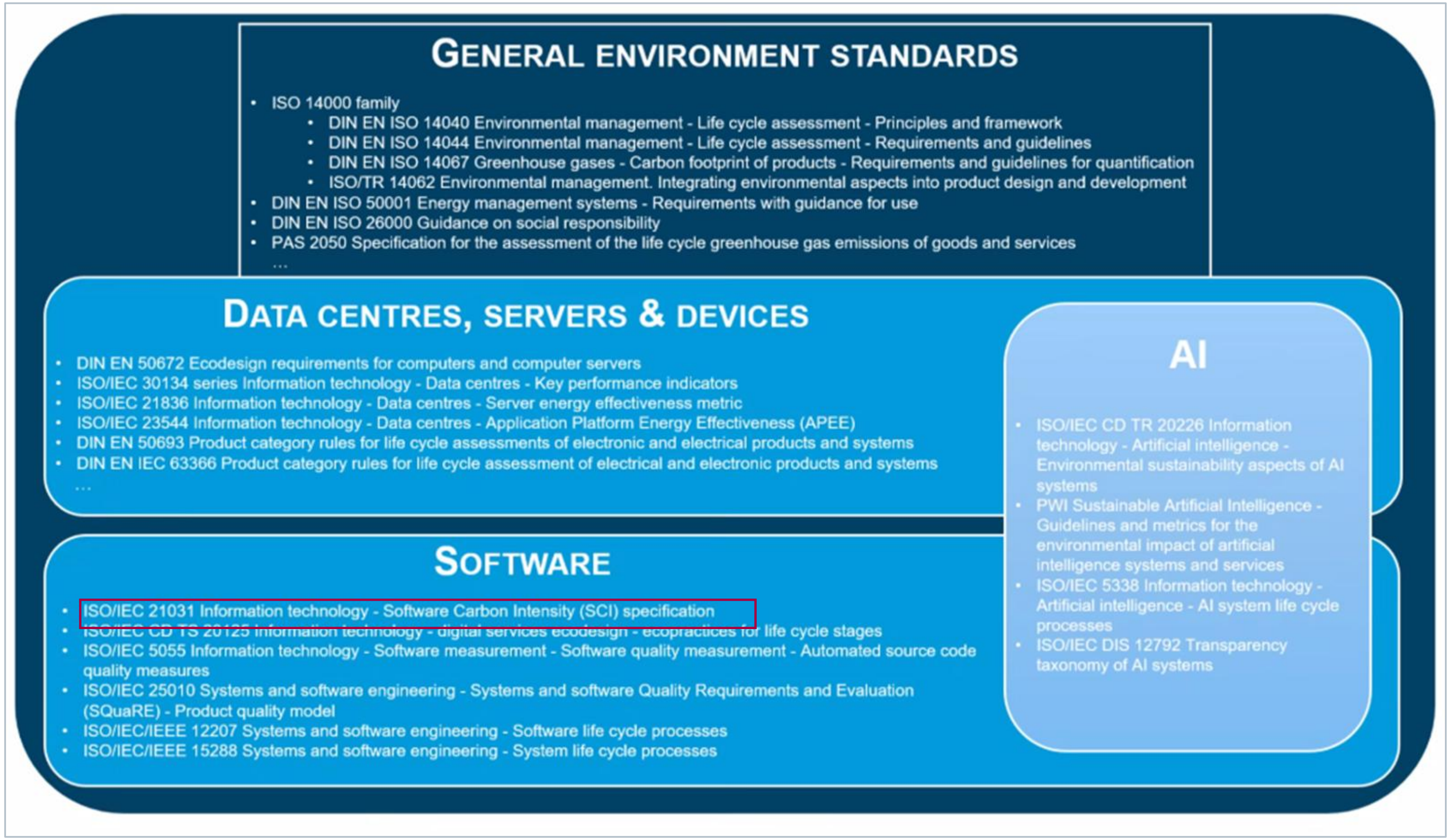
Where is the clearity?

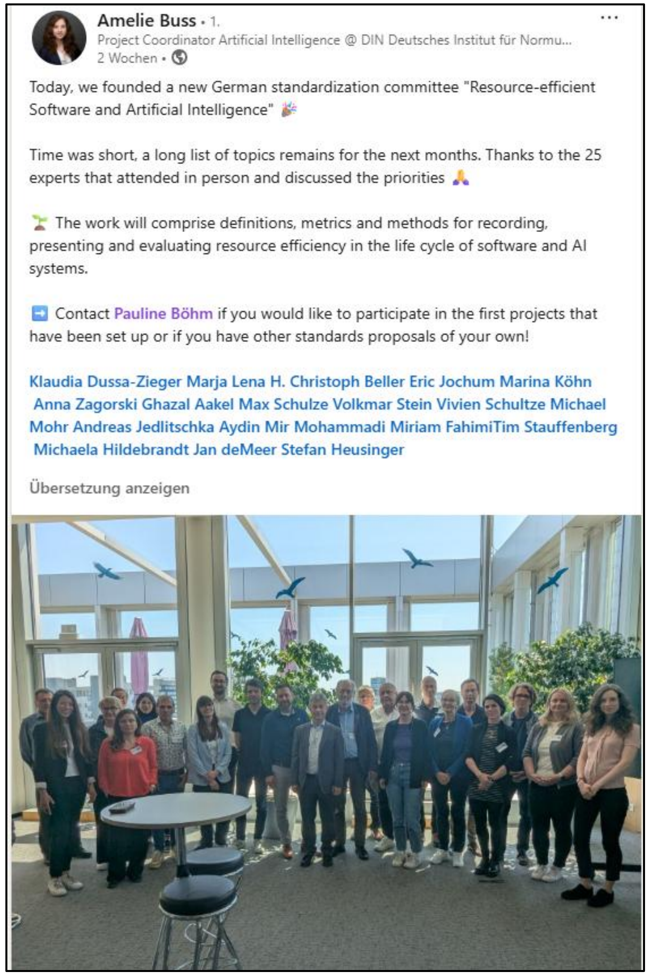
1	As a regulator, I want to assess software companies to ensure that goals of climate and environment protection can be met.
2	As a standards application body, I want to engage with industry stakeholders to ensure the standards for resource-efficient software are practical and widely adoptable.
3	As a software company, I want a precise definition of “resource-efficient software” to optimize accordingly and not be accused of green-washing.
4	As a (sustainability) scientist, I want clearly defined metrics for resource-efficient software to be able to define the impact of methods, techniques and patterns for software.



The Role of Software Development
Standards

and there are even "some" standards…





Source: DIN Open Day topic „Ressourceneffiziente Software & Green AI"

The Role of Software Development

Information technology – Software Carbon Intensity (SCI) specification

A methodology for calculating the rate of carbon emissions for a software system, designed to help developers and architects make informed choices about tools, approaches, architectures, and services.

Energy Efficiency	Hardware Efficiency	Carbon Awareness
Actions taken to make software use less electricity to perform the same function.	Actions taken to make software use fewer physical resources to perform the same function	Actions taken to time- or region shift software computation to take advantage of cleaner, more renewable or lower carbon sources of Electricity.

Source: ISO/IEC 21031:2024

The Role of Software Development

Information technology – Software Carbon Intensity (SCI) specification

Operational Emission

based on the emissions caused by energy consumption

O = (E * I)

E= Energy consumed by a software system

I = Location-based marginal carbon intensity

Embodied Carbon

amount of carbon emitted during the creation and disposal of a hardware device

M = TE * TS * RS

TE = Total Embodied Emissions

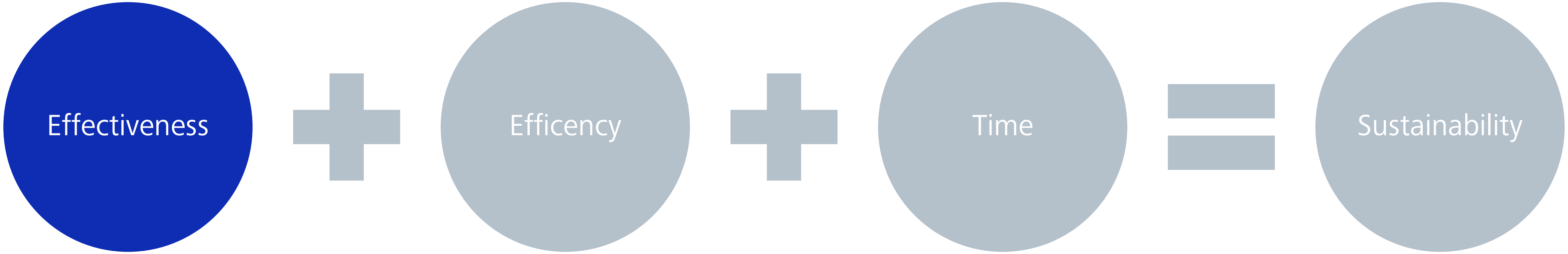
TS = Time-share

RS = Resource-share

Source: ISO/IEC 21031:2024

The Role of Software Development

Effectiveness!



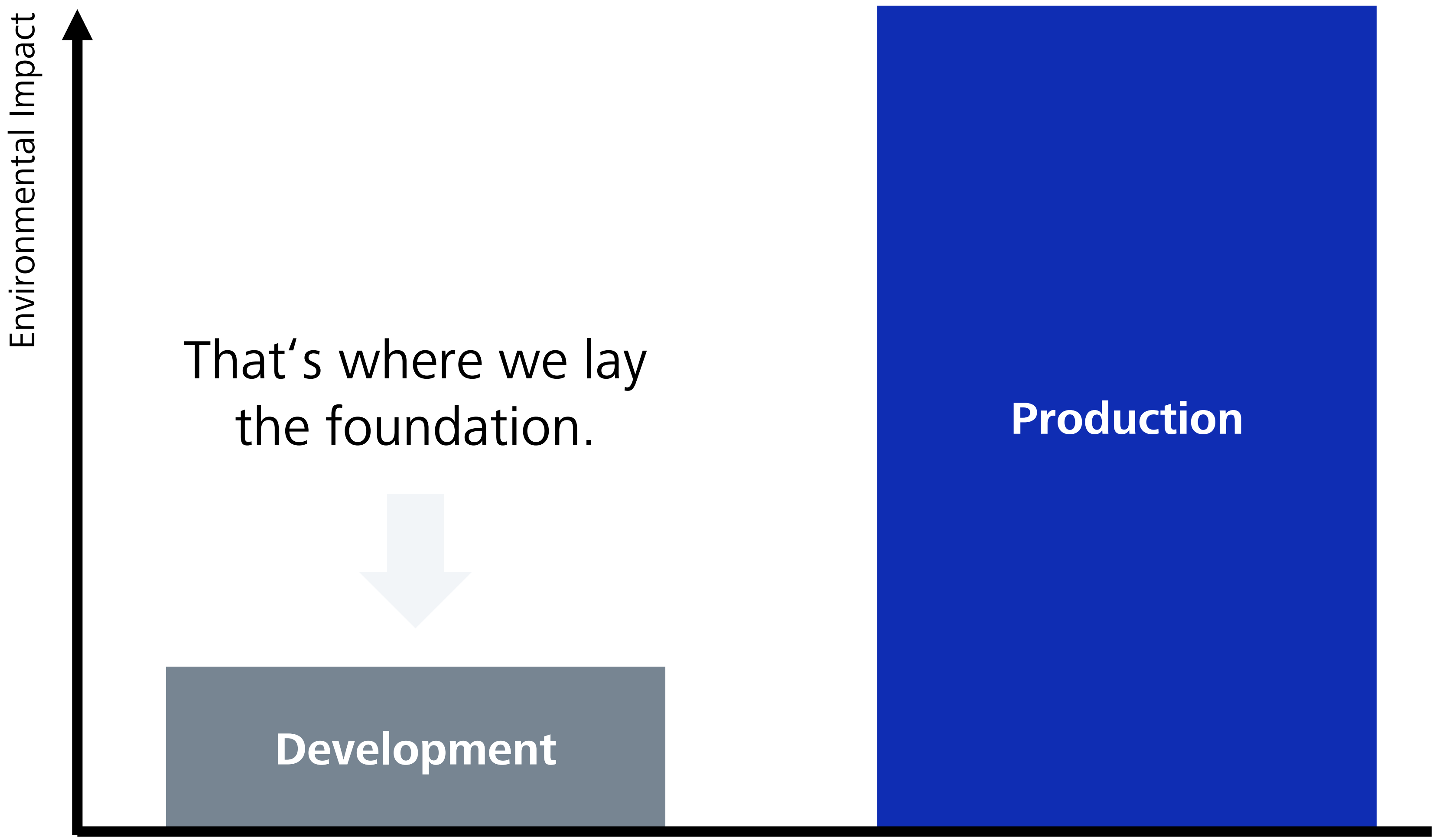
Do the right thing!

Do it right!

Keep doing it!

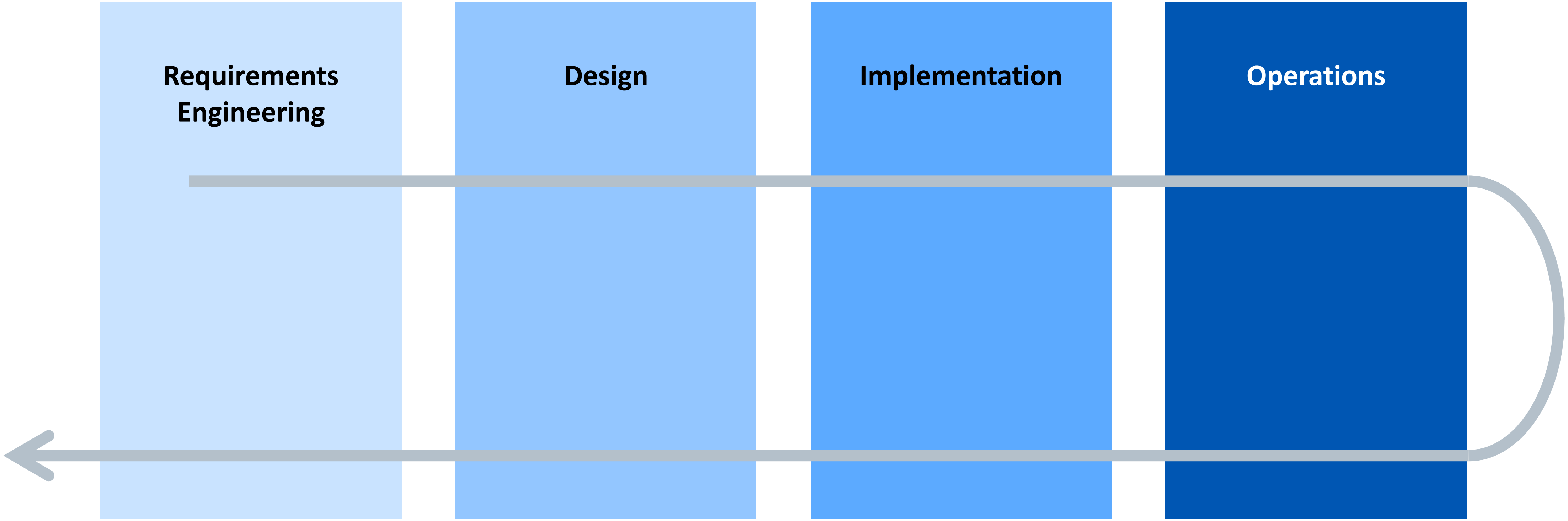
Effectiveness

Usage usually multiplies consumption



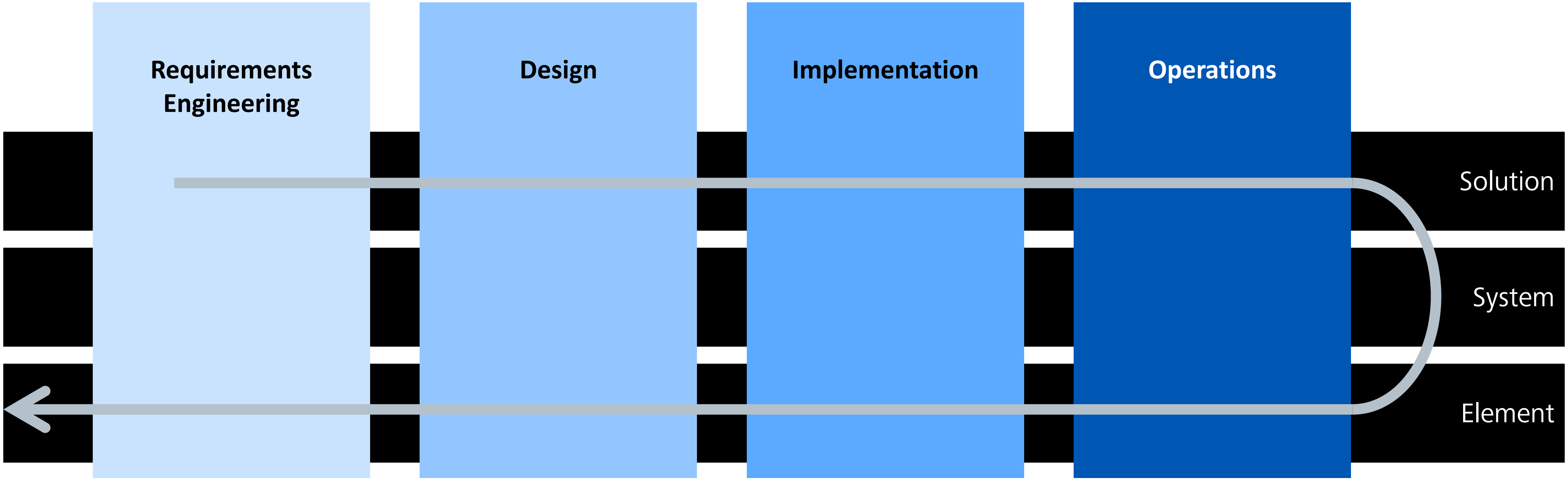
Effectiveness

Ressource Efficiency in the software development process



Work on the big picture and translate requirements and benefits into the working reality of the respective groups of people

Effectiveness
How to be efficient in beeing efficient

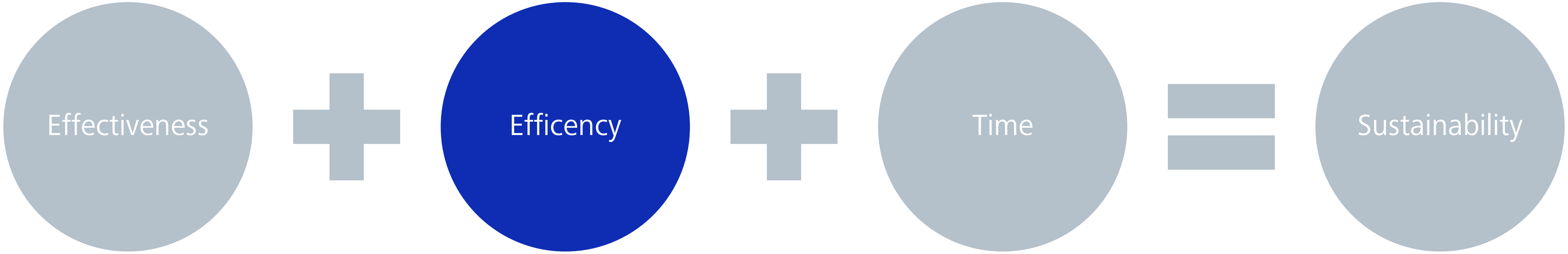


Keep the overall costs of service in mind, not just the development cost!

Effectiveness
Getting the bigger picture by shifting Perspectives

	Clarification of the Assignment	Concept Development	Development	Operation / Further Development
Key Question	What role does resource consumption play in the vision for the planned solution?	How can the planned solution be designed to be as resource-efficient as possible?	How can the solution be implemented as resource-efficiently as possible?	How can the solution be operated as efficiently as possible and how can resource consumption be optimized during operation?
Solution Level	What direct resources would the implementation of the vision consume?	How can business processes and/or the value creation architecture be designed to be resource-efficient?	How can business processes and/or the value creation architecture be implemented in a resource-efficient way?	How can business processes and/or the value creation architecture be operated and optimized to be resource-efficient?
System Level	What resources would the technical system of the solution consume?	How can the technical system of the solution be designed to be resource-efficient?	How can the technical system of the solution be implemented to be resource-efficient?	How can the technical system of the solution be operated and optimized to be resource-efficient?
Element Level	What resources would be consumed by the elements of the solution?	How can individual elements of the solution be designed to be resource-efficient?	How can individual elements of the solution be implemented to be resource-efficient?	How can individual elements of the solution be operated and optimized to be resource-efficient?

Ressource Efficiency
Efficiency!



Do the right thing!

Do it right!

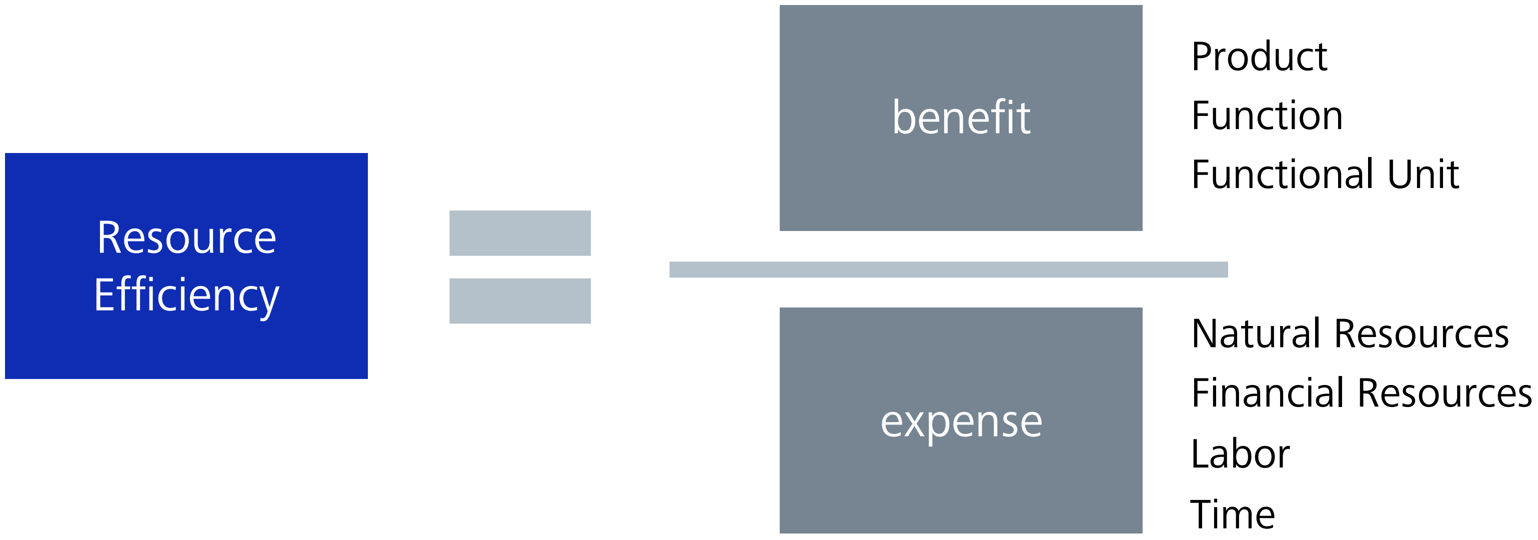
Keep doing it!

Ressource Efficiency
Sustainability or Ressource Efficiency?

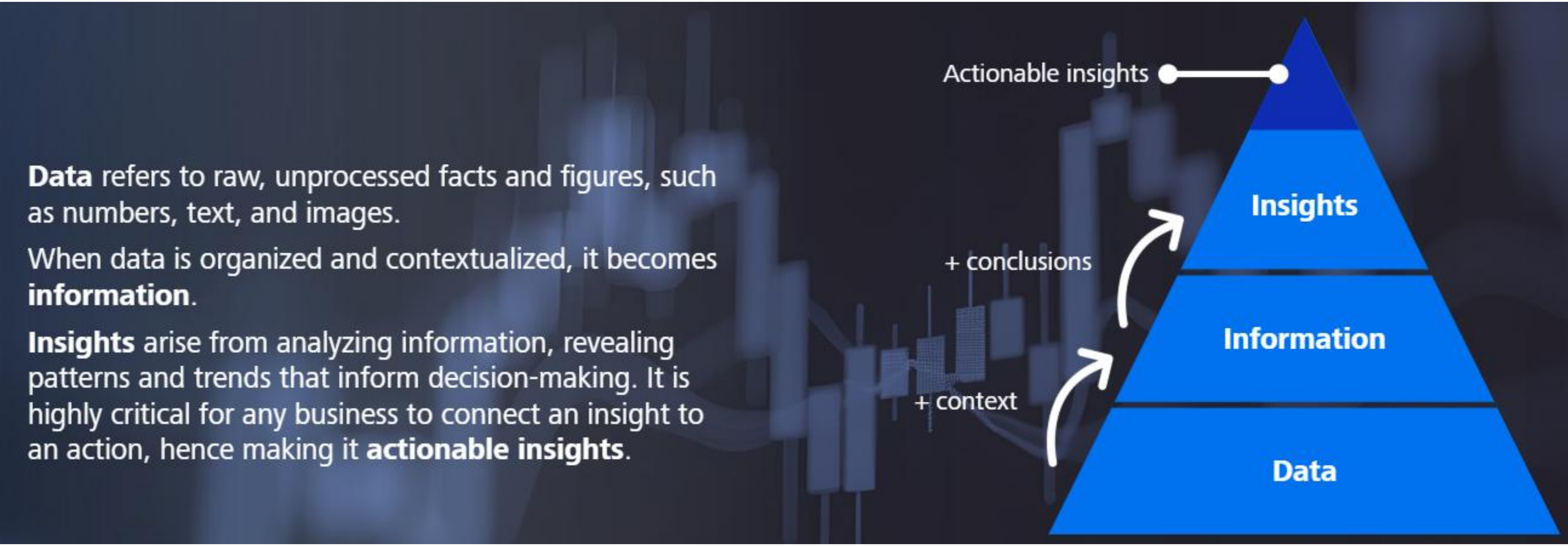
	Sustainability in Software	Ressource Efficiency
Definition	A multi-faceted concept aiming for a balance of environmental responsibility, economic viability, social equity.	Development of software that is maintainable, scalable and adaptable in the long term with optimal utilization of needed resources.
Goals	▪ Supporting social and ethical standards ▪ Reducing the environmental footprint ▪ Reduction of CO2 emissions ▪ Reduction of hardware and energy consumption	▪ Improvement of hardware utilization ▪ Improvement of development costs ▪ Reducing “cost of operation”

Both perspectives are aimed at improving the efficiency and quality of the software. Resource efficiency contributes to sustainability.
Resource efficiency is currently more actionable in daily business.

Ressource Efficiency
Ressource Efficiency



Ressource Efficiency
Data vs. Information



Ressource Efficiency
Compute

Processing Power: CPU cycles for code execution and calculations.
Impact: Inefficient code → higher CPU, energy use, slower performance.

Memory Usage (RAM): Memory needed for active data and instructions.
Impact: High RAM → slower performance (swapping), limits processes/users.

Background Processes: Processes running when not actively used.
Impact: Unnecessary background activity → higher CPU/memory use, impacts performance/battery.
…

Compute

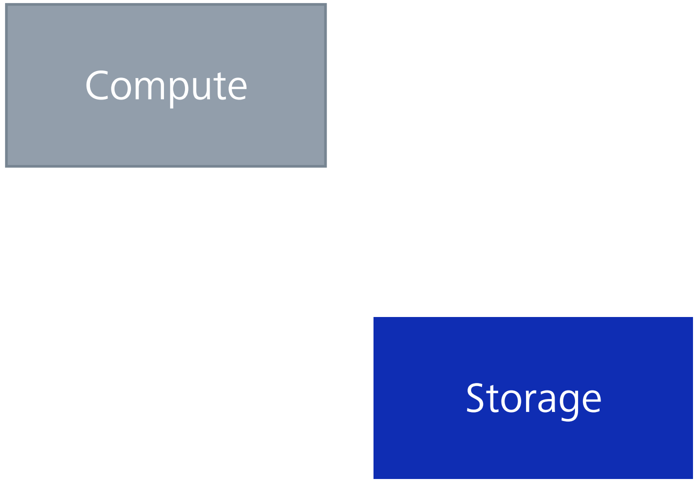
Ressource Efficiency
Storage

Data Persistence: Long-term data storage (databases, files, objects).
Impact: Unnecessary/redundant data → higher storage costs, slower retrieval.

Caching: Storage for performance improvement.
Impact: Poorly managed caches → wasted storage, potential fill-up.

Application Size: Size of the software on disk/in images.
Impact: Larger size → longer download/deployment, higher storage costs.

…



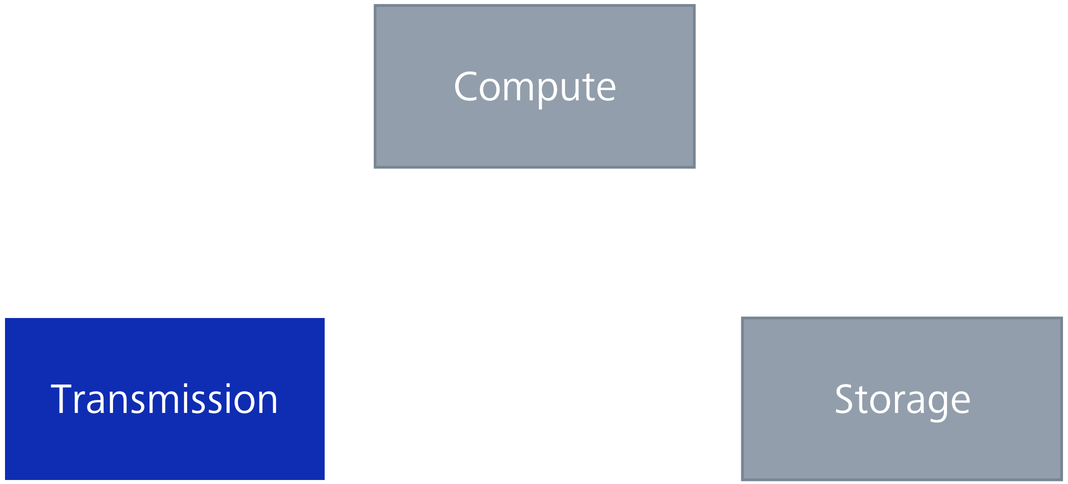
Ressource Efficiency
Compute

Data Transfer: Movement of data across a network.
Impact: High transmission → increased latency, bandwidth costs, energy use.

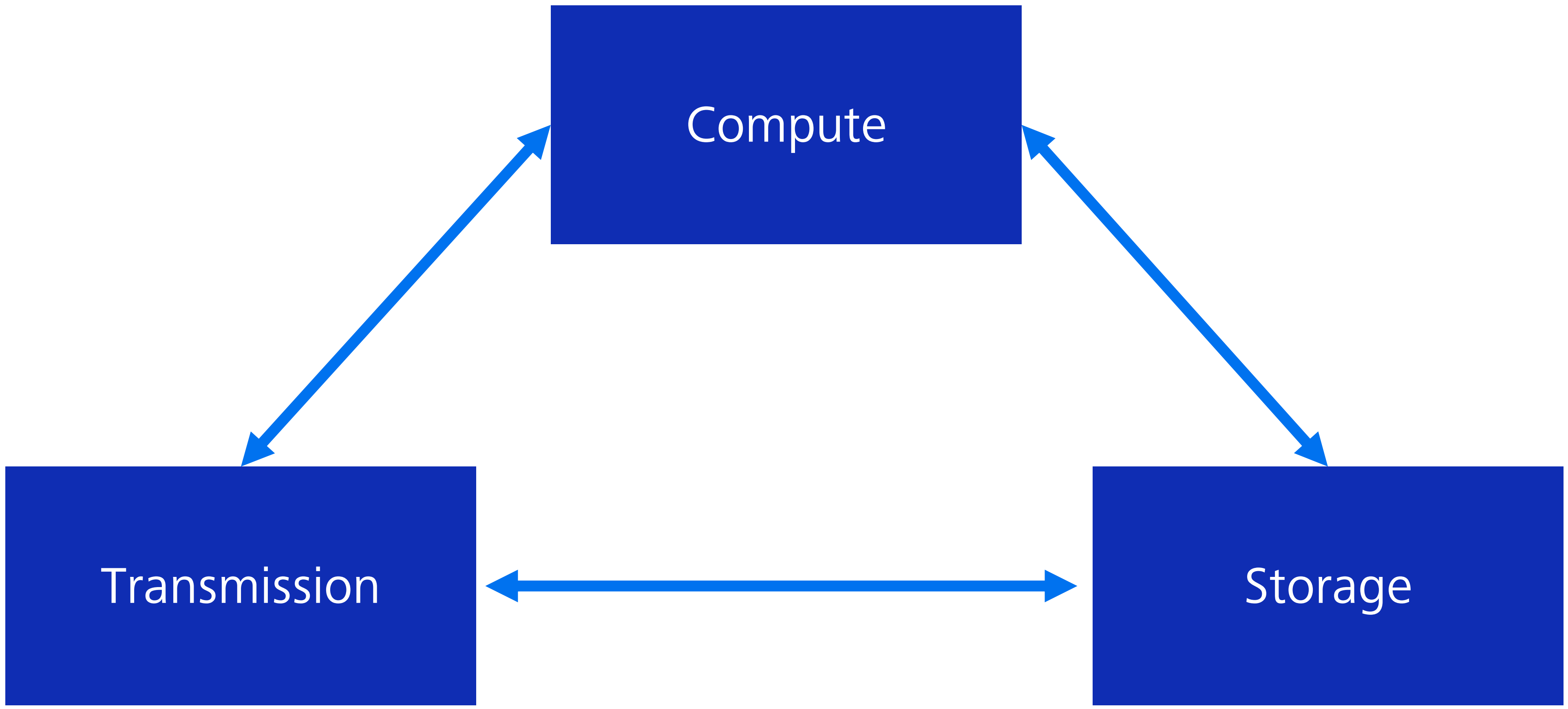
API Calls: Requests and responses between software components.
Impact: Excessive/inefficient calls → network overload, slow performance, higher costs.

Content Delivery: Serving static assets to users.
Impact: Unoptimized content → slower load times, higher delivery costs.

…



Ressource Efficiency
Choose wisely



Ressource Efficiency
Know what you actually need

Consider Interdependencies

- Maintainability *strengthens* Reliability
- Functional Appropriateness *strengthens* Efficiency
- Maintainability and Interaction Capability *strengthen* Security
- Scalability *strengthens* capacity and Efficiency
- Maintainability, Interoperability and adaptability *strengthen* sustainability
- …

ISO 25010

Flexibility
- Installability
- Replaceability
- Scalability
- Adaptability

Maintainability
- Modularity
- Reusability
- Analysability
- Modifiability
- Testability

Reliability
- Availability
- Fault Tolerance
- Faultlessness
- Recoverability

Safety
- Safe Integration
- Operational Constraint
- Fail Safe
- Hazard Warning
- Risk Identification

Functional Suitability
- Functional Appropriateness
- Functional Completeness
- Functional Correctness

Compatibility
- Co-existence
- Interoperability

Security
- Integrity
- Confidentiality
- Authenticity
- Resistance
- Accountability
- Non-repudiation

Performance Efficiency
- Time-Behaviour
- Capacity
- Resource-Utilization

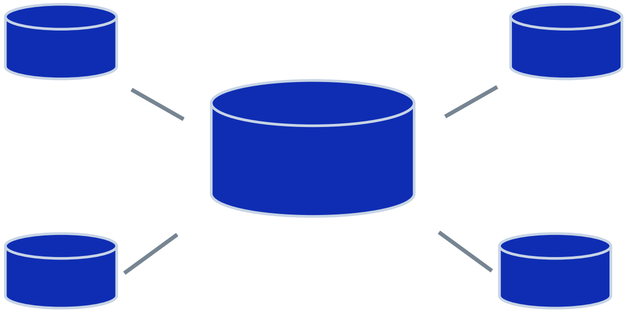
Interaction Capability
- Inclusivity
- User Error Protection
- Learnability
- Operability
- User Engagement
- User Assistance
- Self-Descriptiveness
- Appropriateness Recognizability

Ressource Efficiency
Architecture Patterns

But how to choose?

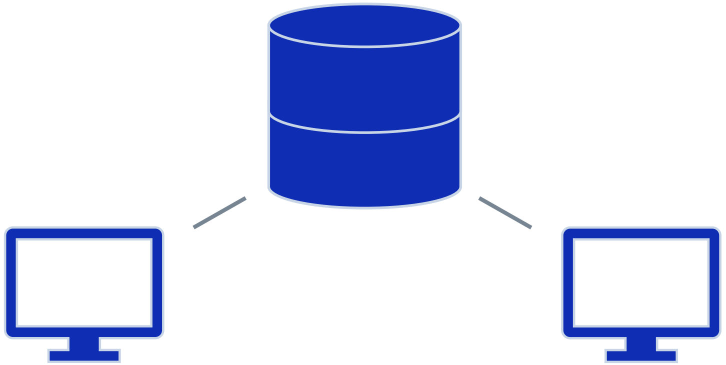
Distributed Architecture

Improves scalability,
but increases transmission costs.

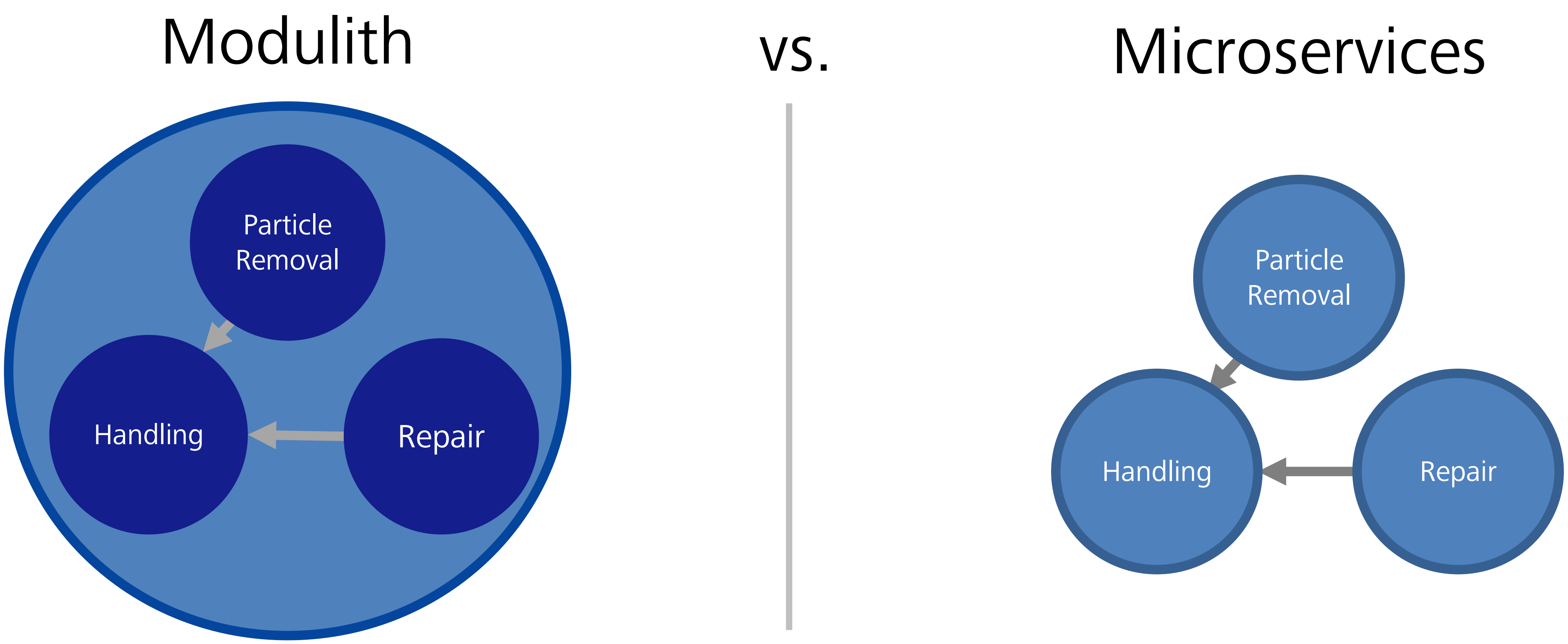


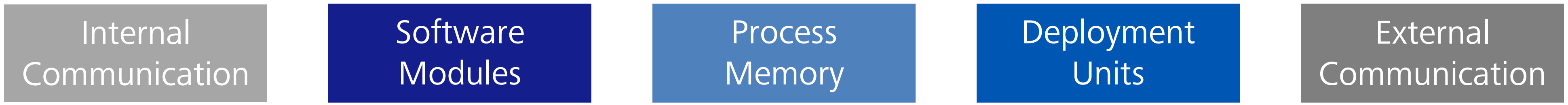
Centralized Architecture

Reduces transmission costs,
but reduces scalability.

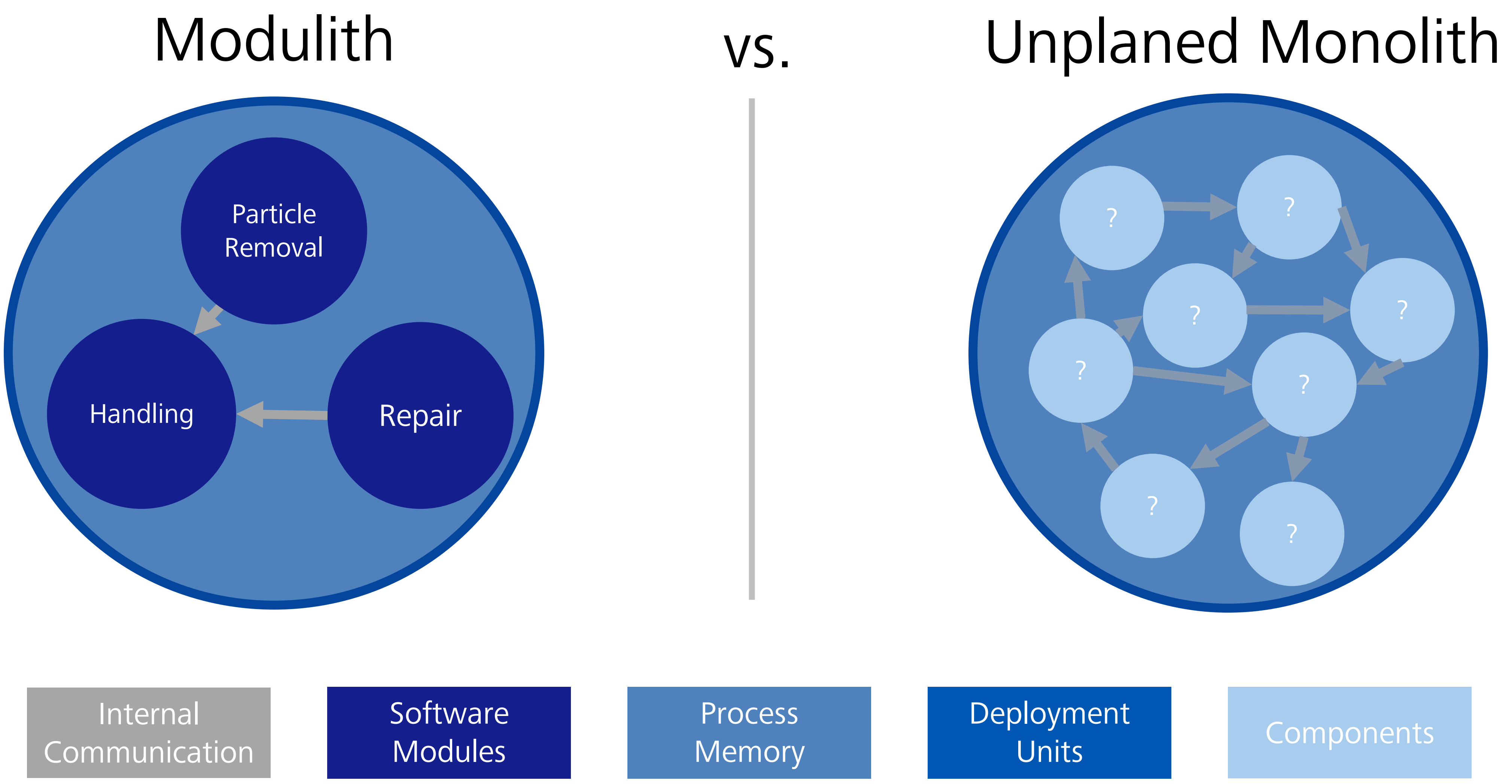


Ressource Efficiency
Moduleth or Microservices





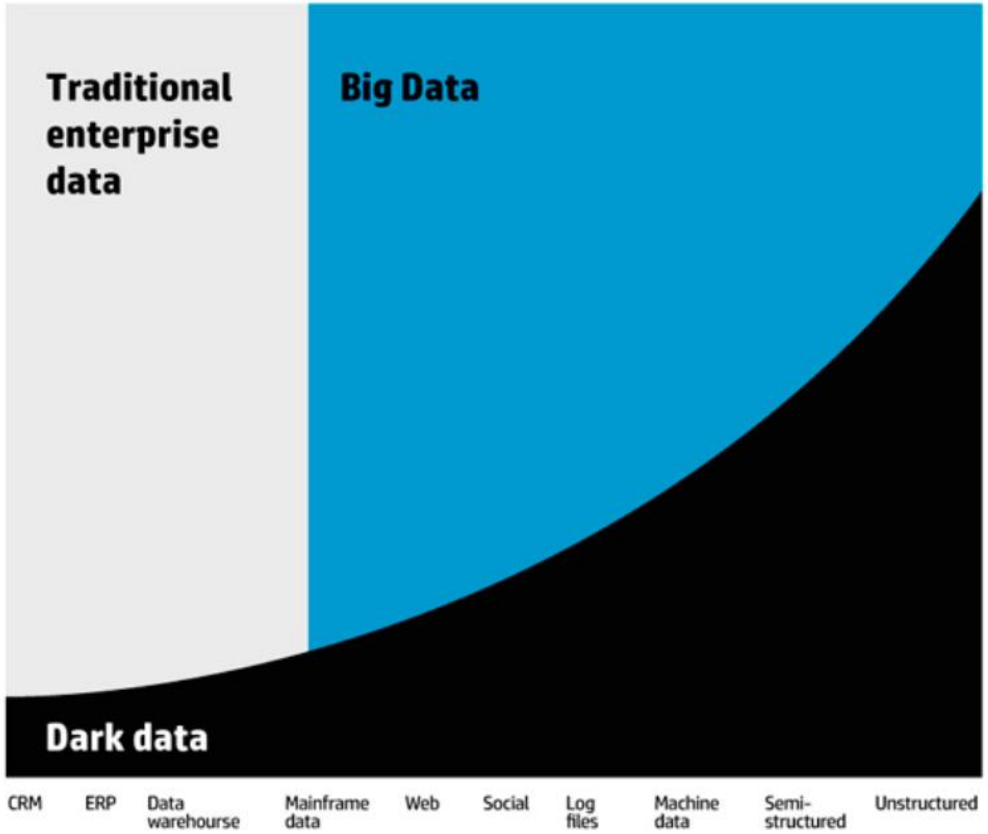
Ressource Efficiency
Finding the right architecture



Ressource Efficiency
Basic Principles as guardrails

„Data is the new oil!“

Clive Humby 2006



Understanding Dark Data | 7wData

Data Avoidance
Avoid any data collection and creation not justified by a clear purpose and compelling necessity.

Data Minimization
Limit collected data to the absolute minimum in scope and duration. Ensure its deletion once the purpose is fulfilled.

Ressource Efficiency
Basic Principles as guardrails

„Data is the new oil!“

Clive Humby 2006

Data Avoidance
Avoid any data collection and creation not justified by a clear purpose and compelling necessity.

Data Minimization
Limit collected data to the absolute minimum in scope and duration. Ensure its deletion once the purpose is fulfilled.

Ressource Efficiency
Basic ideas

Data
- Avoid redundancies
- Use compression and caching
- Optimize locally before transition

Distribution vs. Centralization
- Scale appropriately
- Prefer local optimizations
- Minimize distances

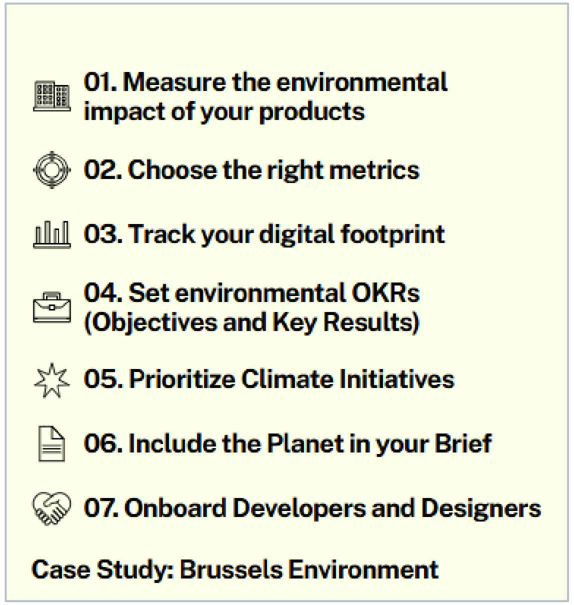
Operation
- Demand-based capacity
- Continuous monitoring
- Time-shifted processing

Procurement
- Purchase resource efficient hardware
- Choose suppliers who follow sustainability best practices
- Take quality seals into account

Initiatives
Climate Productleader Playbook

Best practices and strategies for product managers to embed climate considerations into their work and build more sustainable digital products.

https://climateproductleaders.org/playbook/



Examples

Initiatives
Blue Angel

A German eco-label that certifies software products demonstrating resource and energy efficiency, aiming to reduce their environmental impact.

It works by evaluating software against criteria such as low energy and resource consumption, long-term usability transparency and minimizing hardware replacement.

Initiatives
Blue Angel – Software Criteria

Resource & Energy Efficiency

- Minimum system requiremenst
- Hardware utilisation and electrical power consumption in idle mode
- Hardware utilisation and energy demand when running a standard usage scenario
- Support for the energy management system

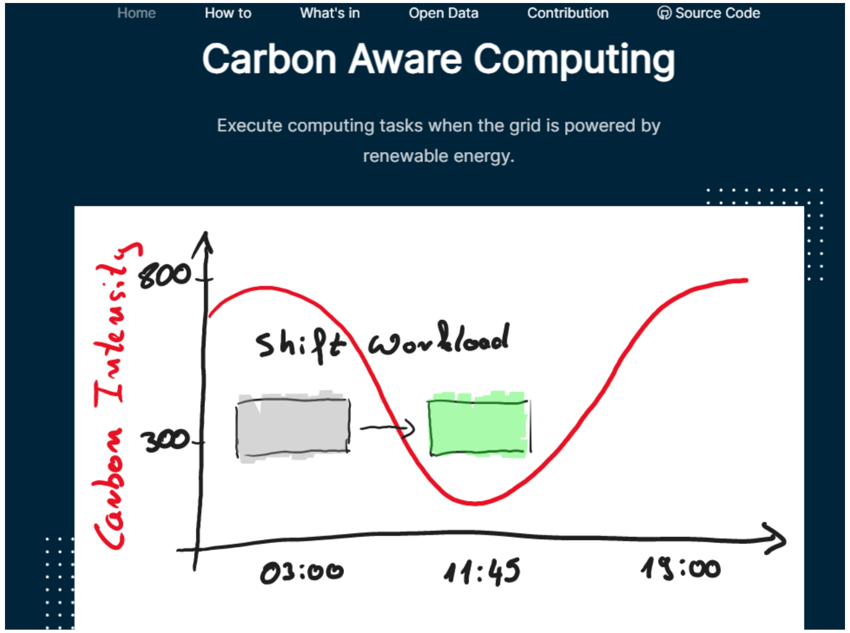
Potential Hardware Operating Life

- Backward compatibility

User Autonomy

- Data formats
- Transparency of the software product
- Continuity of the software product
- Uninstallability
- Offline capability
- Modularity
- Freedom from advertising
- Documentation of the software product, licence conditions and terms of use

Initiatives
Carbon Aware Computing



Libraries & services

A set of packages for .NET, public forecast Web-API and access to the forecast json formatted files.
For more details, download and usage check the GitHub Repository.

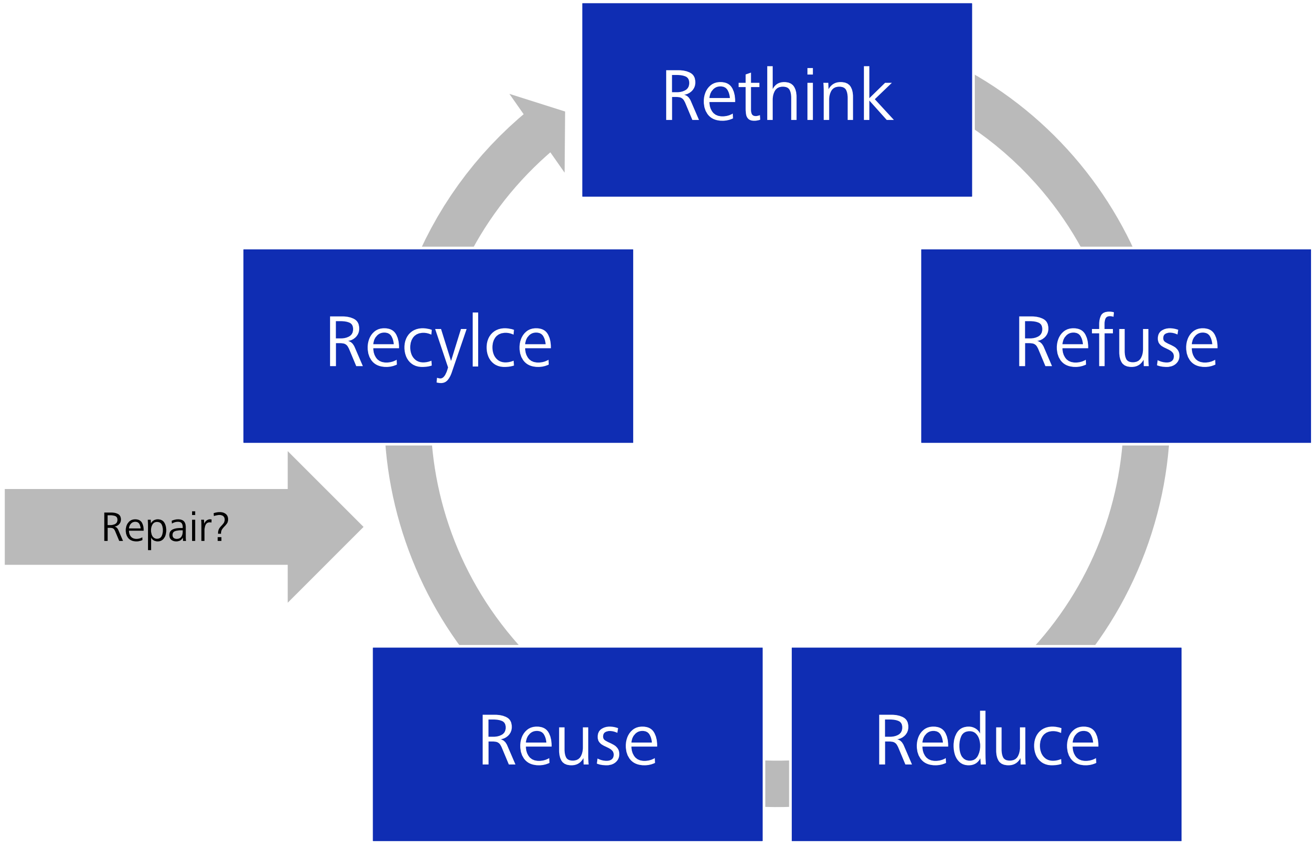
This project delivers a set of libraries, services and data. They are mostly extensions to other projects and all credits belong to them. To forecast the best execution time the Carbon Aware SDK from the Green Software Foundation is used; the forecast data is from Energy Charts provided by Frauenhofer ISE.

- Use the best execution time in your software
- Hangfire Extension
- Forecast Web-API, a subset of the Carbon Aware SDK with open data
- Carbon Aware SDK as NuGet-Package
- PowerShell Cmdlets

Initiatives

The 5Rs of Sustainability

Based on Eurpean Union Legistlation „Waste Framework Directive“
we can also think differently about software.



Soruce on: https://link.springer.com/article/10.1007/s44274-024-00177-4

Summary

Be aware of the Rebound Effect!

The actual definition and discussion is not as clear as this slide makes it appear!

ZEISS

Rebound Effect – The implementation of efficient technology or practices leads to an increase in overall consumption, (partially) offsetting the initial savings.

Economic

The lower cost of using an efficient device encourages more use of that device, or indirectly, as the money saved from efficiency gains is spent on other goods and services that also consume energy.

Psychological

e.g. *moral licensing*. The purchase or use of a more efficient technology is perceived as a good deed that licenses increased preferences for the purchase or use of that or further technology.

Time

technological efficiency improvements bring changes in the time available. Hence, things can be done faster, in parallel or fill in waiting times resulting in an overall higher consumption.

The induction effect: why the rebound effect is only half the story of technology's failure to achieve sustainability (hu-berlin.de)

Summary

Sustainability is more than just saving mother nature

Resilience

Resilience is the capacity of a system or organization to maintain its core functions and adapt gracefully while under environmental, economic, or operational stress.

It fosters sustainability by prioritizing long-term durability over short-term gains, ensuring that the entity survives volatility without requiring a complete, resource-heavy rebuild.

Sovereignty

Sovereignty is the degree of self-determination and independent control an entity has over its critical infrastructure, data, and strategic decision-making.

It enables sustainability by eliminating dependencies on volatile external providers, allowing for the autonomous optimization of resources and the freedom to align long-term operations with ethical and environmental goals.

Security

Security is the practice of safeguarding essential assets and processes to prevent resource depletion and waste caused by disruptions, breaches, or systemic failures.

In a sustainable context, it ensures that organizational investments remain productive over their entire lifecycle, avoiding the high carbon and financial costs of disaster recovery and premature replacement.

Summary
Sustainability is more than just saving mother nature

- **Lower operating costs:** Directly reduces ongoing expenses like electricity.
- **Fewer investments:** Significantly reduces need for new purchases.
- **Lower energy costs:** Immediately saves on electricity bills effectively.
- **Optimized cloud costs:** Minimizes expenses for cloud-based applications.
- **Increased productivity:** Completes tasks quickly and precisely now.
- **Happier customers:** Improves user experience, boosting customer loyalty.
- **Higher competitiveness:** Strengthens market position and secures jobs.
- **Better ROI:** Noticeably increases return on investment.
- **Uncovers hidden costs:** Identifies and eliminates unnecessary expense items.
- **Stronger brand value:** Attracts eco-conscious customers and investors.
- **Longer hardware lifespan:** Reduces frequency of upgrades and replacements.

Summary
Some last remarks

CO2 Emissions are a Cost Factor	Companies incur costs from legal and environmental regulations related to CO2, which can also harm their image and lead to market losses.
Varying CO2 Reduction Approaches	Approaches vary by application, requiring tailored solutions rather than general guidelines.
Balancing CO2 Reduction	Decisions in software development should consider various consumption goals alongside performance, scalability, and maintainability to achieve sustainability.
Integrating Resource Efficiency	Resource efficiency must be part of all software development phases to ensure consistent application and integration into corporate strategies.
Continuous Monitoring	Ongoing analysis is crucial for embedding sustainable strategies and addressing emerging issues.
Data-Driven Adjustments	Effective optimization relies on a solid data foundation to avoid ineffective or counterproductive changes.

ZEISS

Seeing beyond