



TECHNOLOGY DAY 30.11.2022

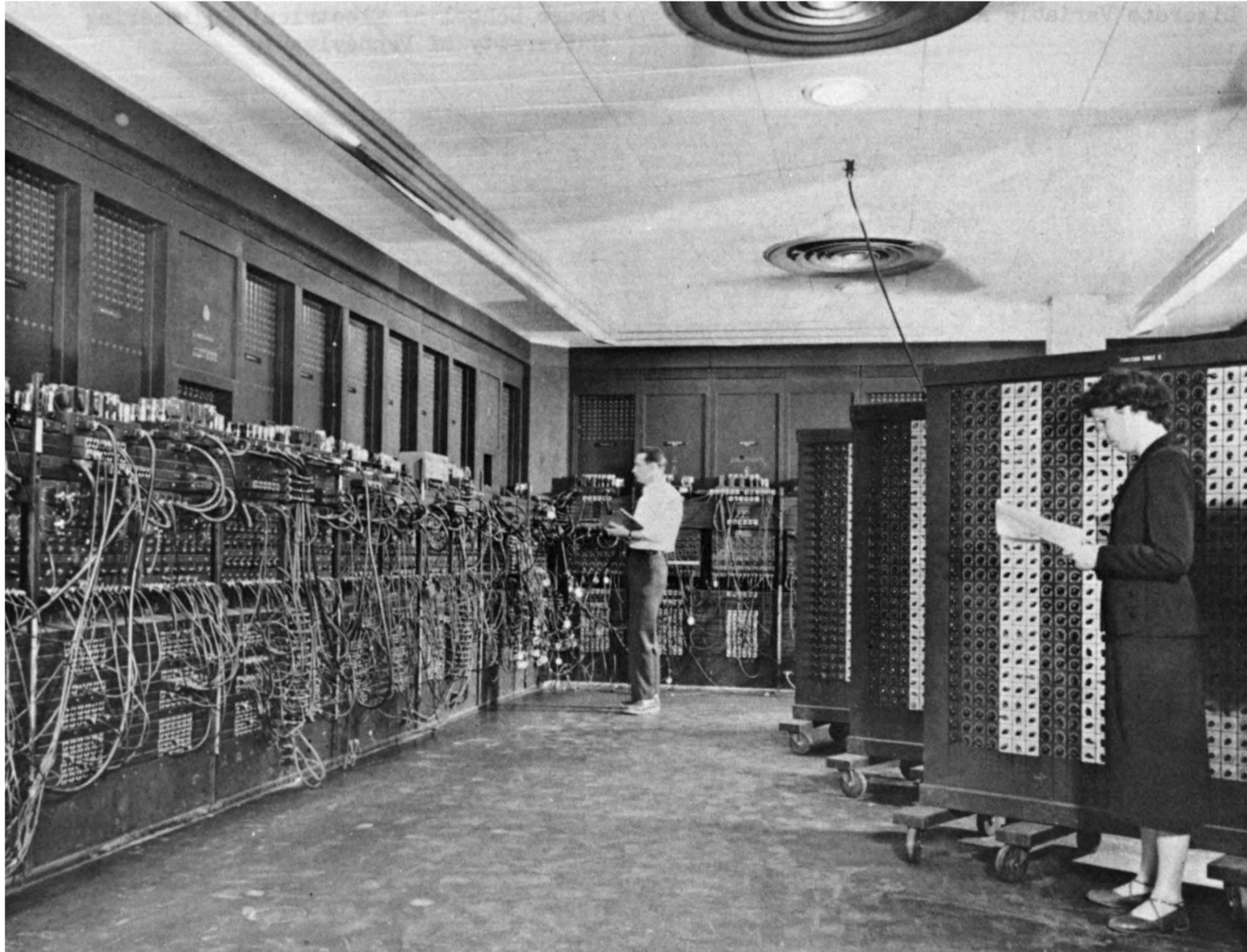
It's not easy being green

INNOQ

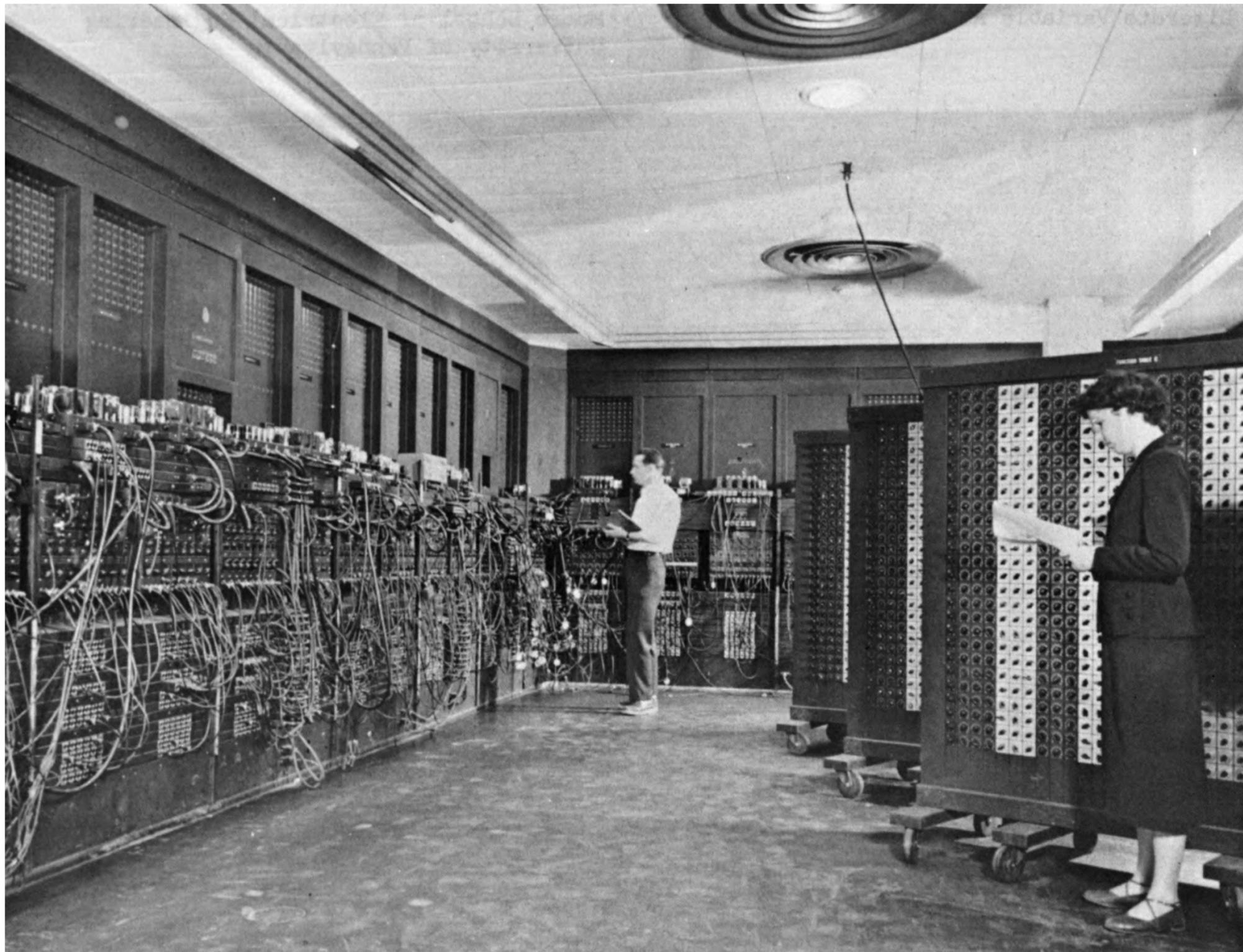


MICHAEL KRÄMER
@MKRAEMERX

Woher kommen wir?



1943



1943



OBAMA'08

YES WE CAN

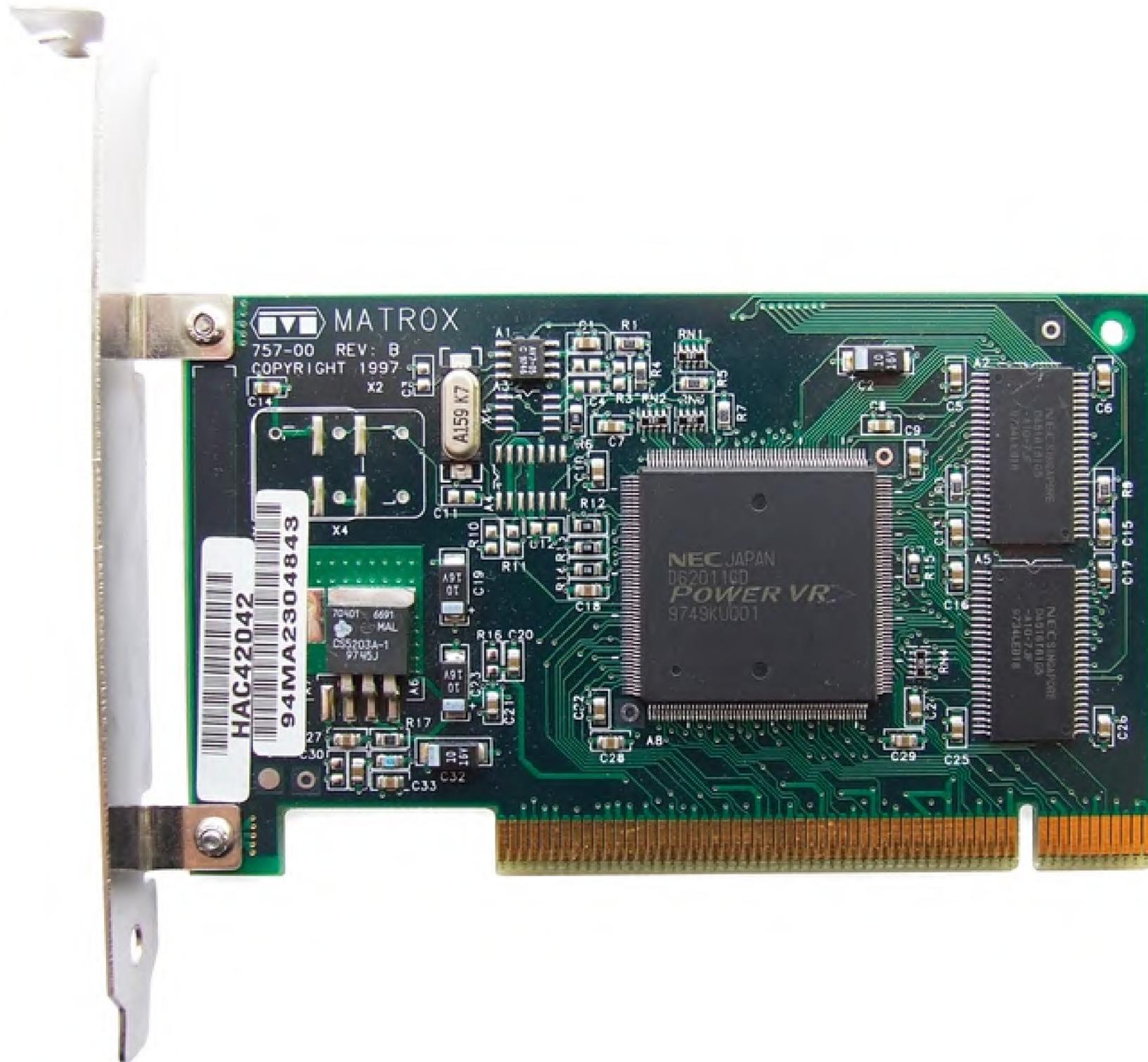
WWW.BARACKOBAMA.COM



~1985

https://www.reddit.com/r/OldSchoolCool/comments/873ggw/nerd_in_the_80s/

1997



Von Trio3D - Eigenes Werk, CC BY-SA 4.0,
<https://commons.wikimedia.org/w/index.php?curid=77555393>

Ausreichend Leistung

64bit

Multi-Core

SSE

> 2GHz

ENTWICKLER TEURER ALS HARDWARE !!!

2007



Ausreichend Leistung !!

8 Cores

2.8 GHz

bis 12GB RAM

2022



By Dinkun Chen - Own work, CC BY-SA 4.0,
<https://commons.wikimedia.org/w/index.php?curid=120356443>

Trends

klein

leicht

Akku

2020



maximale Abstraktion

Cloud-Umgebungen

gar kein physikalischer Kontakt

Container & Orchestrierung

ab 2006

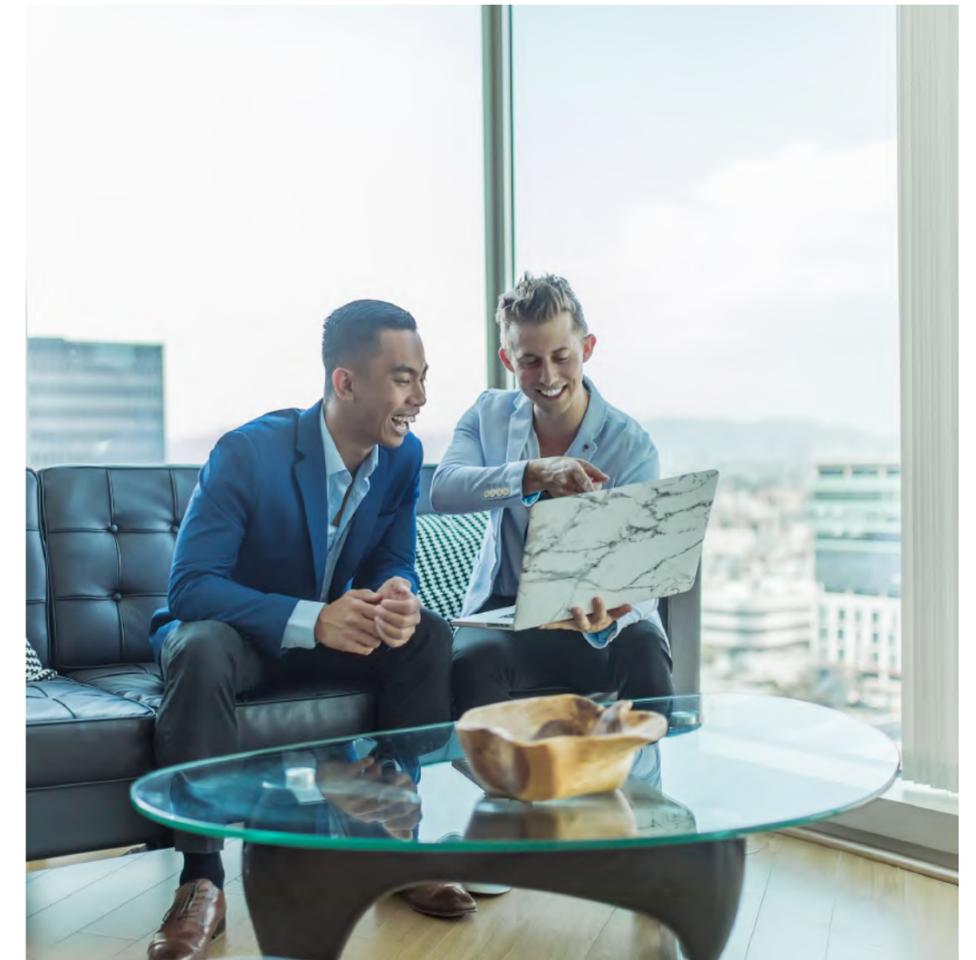
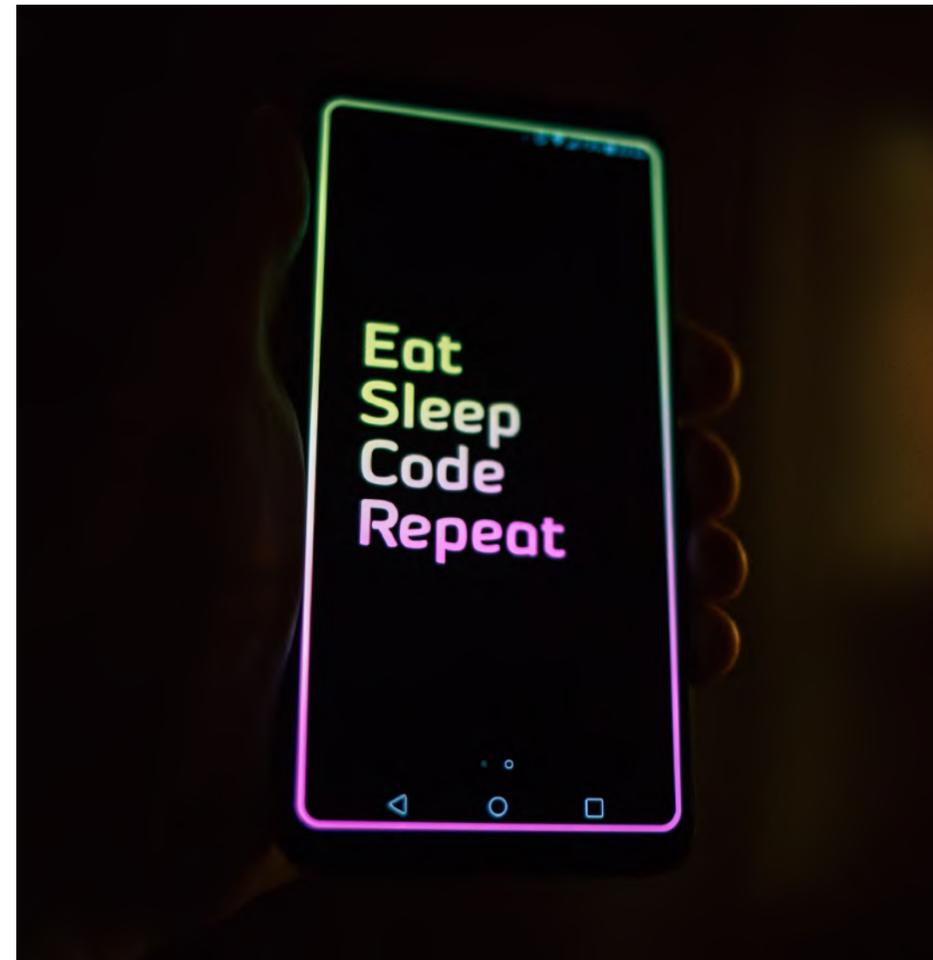
aws



Wie können wir grüner werden?

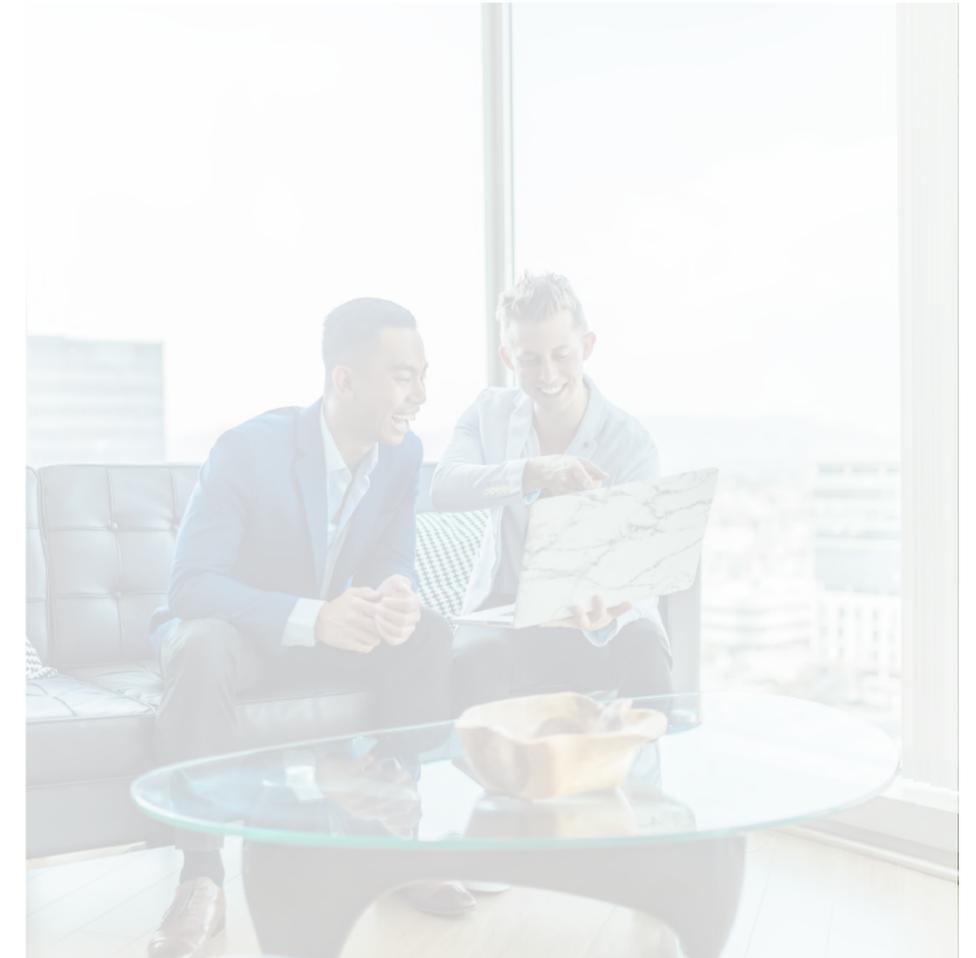
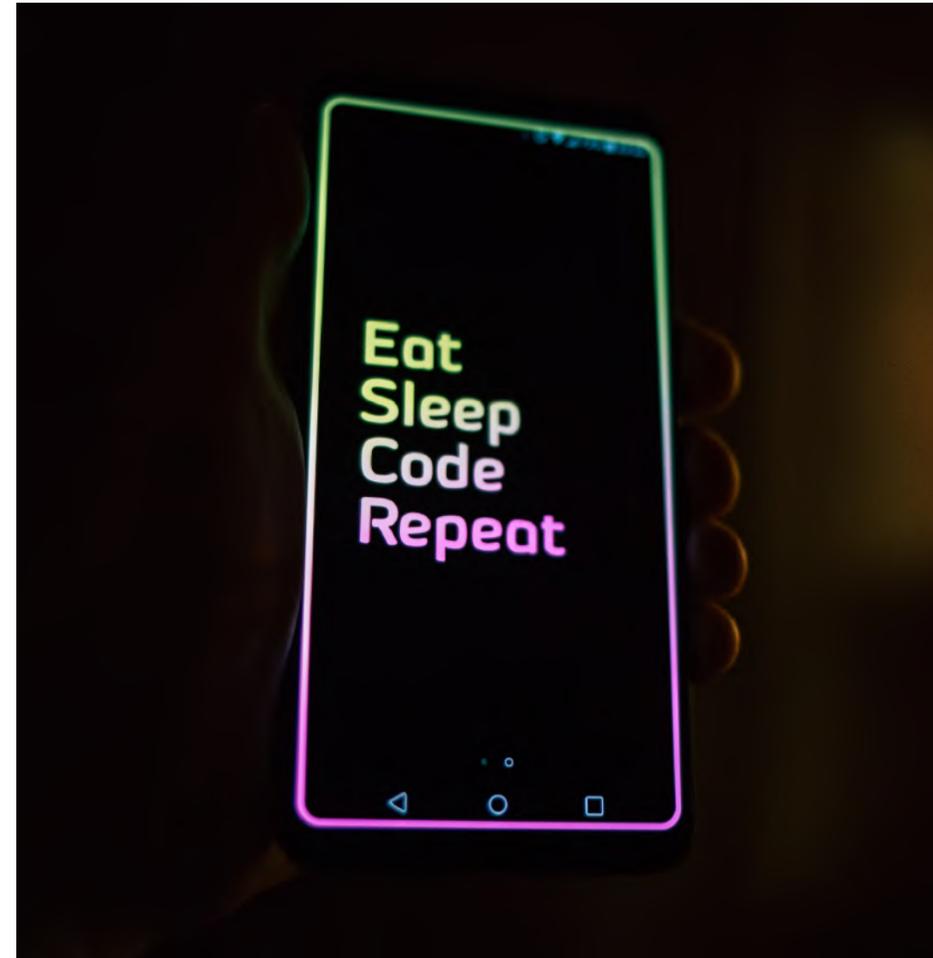
3 Ansatzpunkte

Infrastruktur / Implementierung / Fachlichkeit



3 Ansatzpunkte

Infrastruktur / Implementierung / Fachlichkeit



Energie- & Laufzeiteffizienz.

Die Zutat des Softwareentwicklers.



**Effiziente Software zu bauen ist kein neuer
Job!**

**Effizienz als
Architekturziel**

**wirtschaftlich
vs. ökologisch**

**Vorschnelle
Optimierung
ist die Wurzel
allen Übels** 

**Ressourcen-
effizienz ist ein
Qualitäts-
merkmal**

Schnell \neq effizient.

**Abhängigkeiten zwischen
Softwareentwicklung und Effizienz finden.
Messen!**

Definitionen & Metriken



Green
Coding

Sustainability
Informatics

Footprint

SE4S [Software
Engineering
for
Sustainability]

Green
IT

Permacomputing

Sustainability

Green

Green
Software

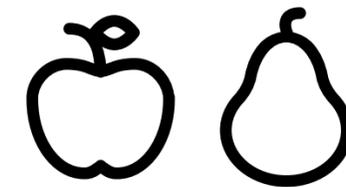
Metriken für Green Software.
Sind leider gar nicht trivial.

Messen - global

- Absolute Werte sehr schwierig zu messen
- Vergleichende Messungen
 - Ressourcenverbrauch einer ganzen Plattform / Komponente
 - Funktioniert in der Cloud
 - Messung / Reduktion über Zeit → ETSY
- Einfachste sinnvolle Einheit finden
 - T CO₂-Equivalent muss nicht sein
 - innerhalb eines Systems (z.B. Cloud-Anbieter) Maschinen vergleichen
 - Kosten zählen
 - Äpfel vs. Birnen !

Tools für globale Messungen

- Cloud
 - Carbon Dashboard in GCP, Azure und AWS
 - leider teils sehr zeitverzögert
 - manche Produkte fehlen (ECS)
 - Berechtigungen schwierig weil mit Billing verbunden
 - Sonst: <https://www.cloudcarbonfootprint.org/>
 - basiert auf ETSY Cloud Jewels / Schätzungen
- On Premise
 - Energiemessgerät
 - RAPL-basierte Tools



Tools für spezifische Messungen

- Powertop auf Prozessebene (mit Einschränkungen weil Batterie-optimiert)
 - auch in der Cloud
- JoularJX
- andere RAPL Tools
- oder bekannte Profiler oder Laufzeitmetriken (viel Laufzeit → viel Ressourcen)

JoularJX

- Sehr spannende Einblicke!
- leider nicht in der Cloud nutzbar (KVM)
- schreibt Files pro Sekunde

```
java.io.RandomAccessFile.readBytes,0.10021630705911803
java.math.BigInteger.multiply,0.20043261411823607
java.lang.StrictMath.<clinit>,0.10021630705911803
java.math.BigInteger.squareToLen,0.40086522823647214
java.math.BigInteger.getLowestSetBit,0.10021630705911803
java.math.BigInteger.pow,0.10021630705911803
java.lang.ref.Reference.waitForReferencePendingList,0.02888
java.math.MutableBigInteger.divideMagnitude,0.10021630705911803
java.math.BigInteger.square,0.10021630705911803
java.lang.Thread.getStackTrace,24.051443828453166
jdk.internal.misc.Unsafe.putReference,1.264264059069434E-4
java.lang.Character.toLowerCase,0.10021630705911803
~
```

```
30/11/2022 01:17:25.395 - [INFO] - JoularJX finished monitoring application with ID 879102
30/11/2022 01:17:25.401 - [INFO] - Program consumed 37.14 joules
30/11/2022 01:17:25.402 - [INFO] - Energy consumption of methods and filtered methods written
```

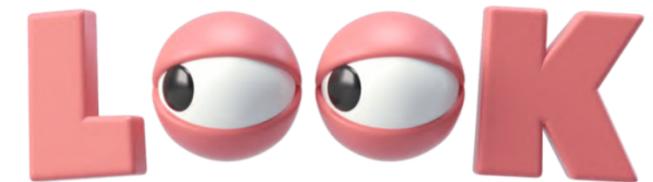
Relevantes Potential finden.

Anstatt Klein Klein zu optimieren.

Rezepte

Autoscale

- Clustermanager und Verteilung sind eine Art Overhead
- aber können ökologisch wertvoll sein wenn nach unten skaliert werden kann
- **ScaleToZero** über Cloud Functions / Lambdas / Serverless
 - bisher meist nicht erste Wahl als Lösungsansatz wegen Limitierungen
 - KNative bietet mehr Kontrolle und mehr Optionen
- sinnvoll in Situationen wo die Last nicht gleichmässig ist



Continuous Integration

- Automatisiertes Testing ist manchmal sehr zäh
- Tests laufen lange
- spätes Feedback, ggf. Abbrüche
- **grosser Ressourcenbedarf** UND schlechte Produktivität
- **Investieren um Tests zu vereinfachen**
 - Gute Teststrategie ist schon Teil der Architektur / des Designs
- Ggf. nicht alle Test bei jedem Commit laufen lassen
- sinnvoll wenn CI-Server / Entwicklungswerkzeuge viele Ressourcen binden

Datenflüsse minimieren

- Generische Datenstrukturen auf Eignung überprüfen
 - Vermeiden nur einen Bruchteil der Daten zu nutzen
 - Sortieren / verarbeiten nur einmalig tun
- Datenbankfunktionen und spezifische Selects nutzen
- evtl. sinnvoll wenn ORM Frameworks genutzt werden und 1:1 zwischen Objekten und Tabellen gemappt wird

Generische Lösungen / Effizienz

- Manche generische Lösungen sind wenig effizient
- Z.B. Kapselung von Verteilung
- Kapselung generell weniger das Problem (Literatur), mehr Generalisierung

- Wartbarkeit / Modularisierung und Laufzeiteffizienz sind nicht immer beste Freunde
- Kompromisse suchen

Jeder Beitrag zählt.

Labels

Methoden / Labels

- Konkret auf Software: Blauer Engel (Deutschland)
- Life Cycle Assessment konkretisiert
 - z.B. Sleep Mode → Analogie zu ScaleToZero
- Energiebedarf muss angegeben werden (Szenarien)
- Life Cycle Ansatz schwierig im agilen Umfeld mit CI

- Software development methodology in a Green IT environment (Frankreich)



Bisher keine Praxiserfahrung, aber bereit für jedes Abenteuer!

Sonstige Labels beziehen sich auf ganze Wertschöpfungsketten.



Wissenschaft & andere sehr spezifische Gedanken

- Effizienzstudien Java Collections
- Programmiersprache
- Quarkus

Links

<https://www.green-software.ch/> (DNS Problem heute, hoffentlich Ende Woche)

Messen

- <https://gitlab.com/joular/joularjx>
- <https://www.cloudcarbonfootprint.org/>

Labels

- https://www.umwelt-campus.de/fileadmin/Umwelt-Campus/Greensoft/Entwurf_BLAUER_ENGEL_Vergabekriterien-Software_20190527k.pdf
- <https://observablehq.com/@mrchrisadams/greening-someone-elses-compute>

Danke! Fragen?



Michael Krämer

michael.kraemer@innoq.com

+41 77 422 40 61

@mkraemerx

innoQ Schweiz GmbH

Gewerbestr. 11
6330 Cham
+41 41 743 01 11

Hardturmstr. 253
8005 Zürich