Green Software Design

Strategien für Energie effiziente Lösungen

DIN Chapter Bern
Erupt Bar Lounge
14. November, 2023

```
markus_g@EDONB105 MINGW64 /c/dev/boe/boe-app (develop)
$ whoami
markus_g
```

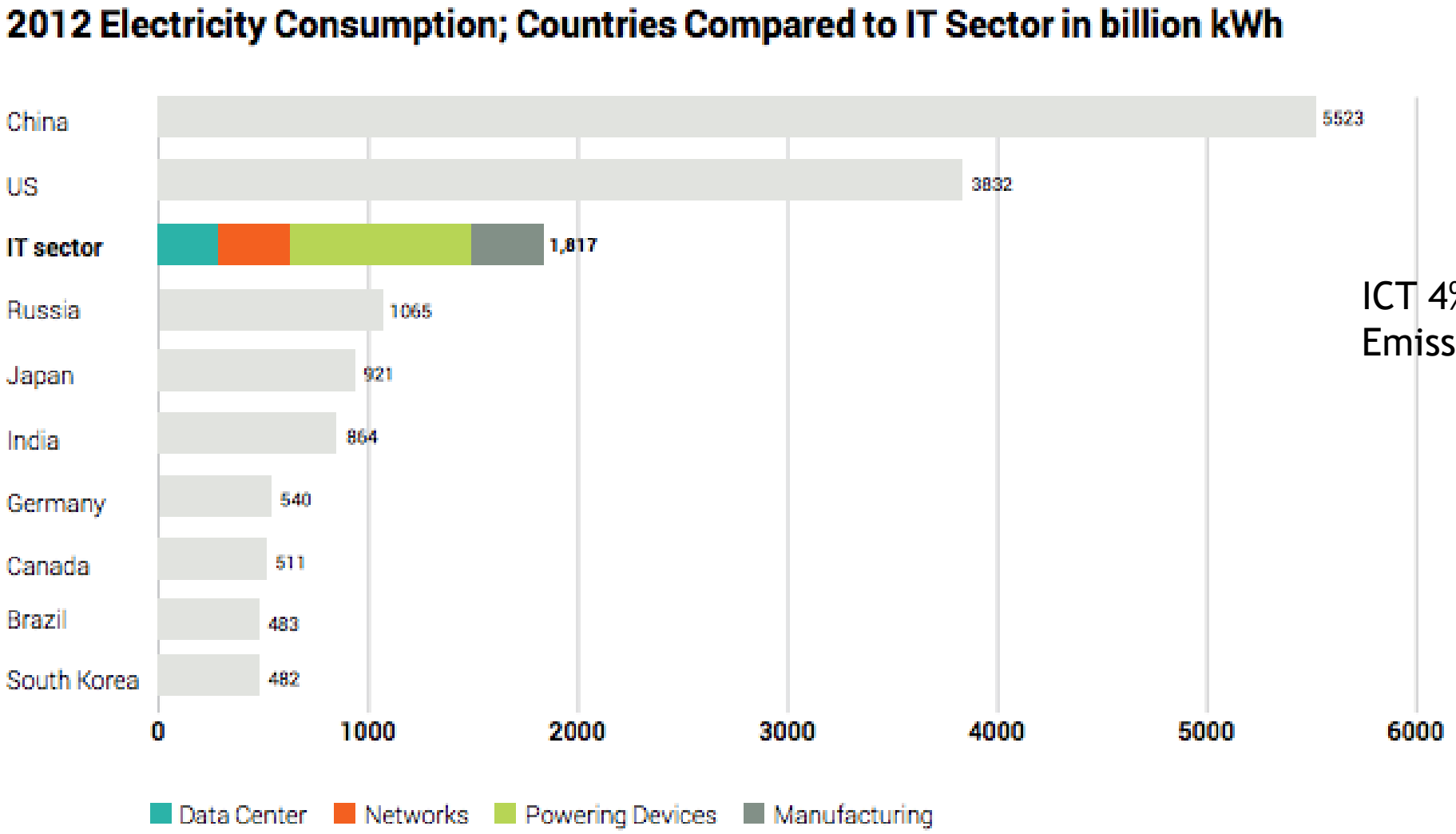
Markus Gallagher

Software Entwickler

DIN Fachgruppe Sustainable IT

Transition Schwarzenburgerland

Motivation

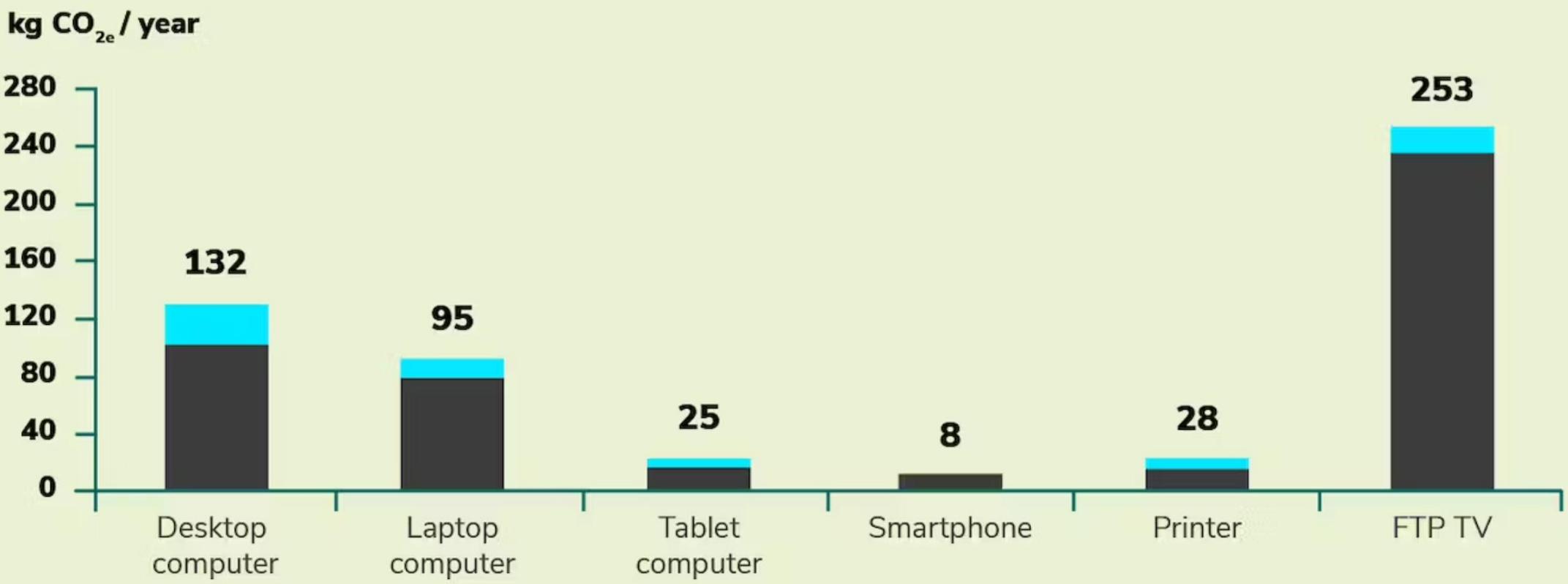


ICT 4% aller CO^2 Emissionen

Source: Emerging Trends in Electricity Consumption for Consumer ICT, Peter Corcoran and Andres Andrae (2013) and CIA World Factbook. China/Russia/Canada figures are from 2014.

DIRECT EFFECTS

CO_{2e} emission per ICT end user device



REDUCTION OF ENVIRONMENTAL IMPACT FROM PRODUCTION BY

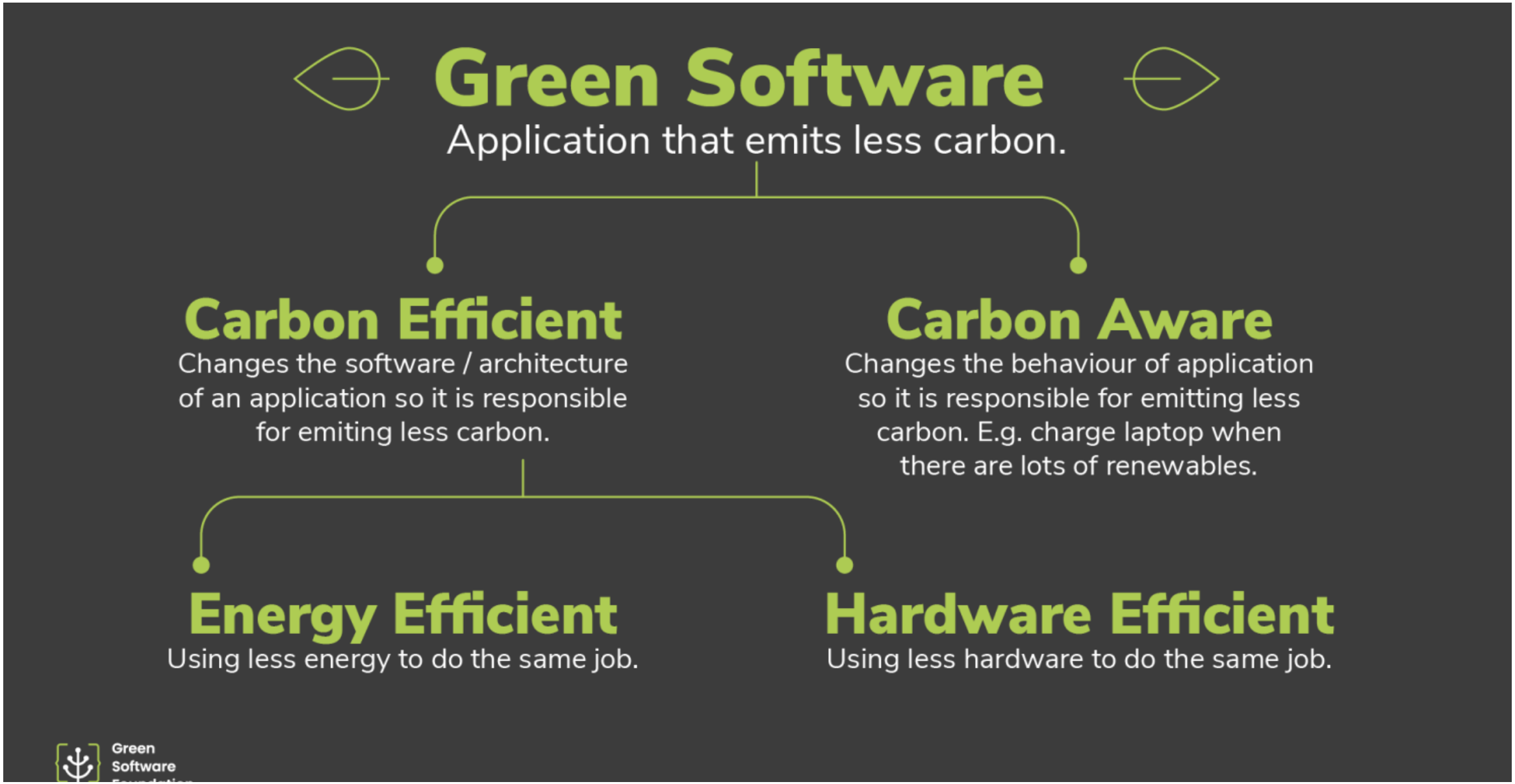
- Reduction of the number of devices (e.g., through lifetime extension, fewer devices per person)
- Increase in energy and material efficiency in production

Green Software

Bereiche mit dem grösstem Impact

HARDWARE

DATA CENTERS


Green Software
Application that emits less carbon.
Carbon Efficient
Changes the software / architecture of an application so it is responsible for emiting less carbon.
Carbon Aware
Changes the behaviour of application so it is responsible for emitting less carbon. E.g. charge laptop when there are lots of renewables.
Energy Efficient
Using less energy to do the same job.
Hardware Efficient
Using less hardware to do the same job.
Green Software

Zuerst Messen…

Was

- Gebundenere Kohlenstoff (Hardware)
- Energie und deren Quelle
- Unsere Software
  - Hot Paths
  - Rechenintensive Operationen
  - Build Pipelines
  - Long Running Transactions

Wie

- **Regelmässig**
- Wattstundenzähler
- Cloud / ISP Daten

Hosting

- Präferiere Cloud über On-Prem
  - AWS Lambda & Azure Function
  - Cloud Providers haben Net Null versprochen
- Führe ein Dialog mit Hosting Partner bzgl. Sustainability
- Hosting Kosten ≈ CO^2 Ausstoss
- Höhere Auslastung ≈ Effizienter
- Power Usage Effectiveness (PUE)
  - 1.5 = 1 für Compute, 0.5 für Kühlung, etc.

Architecture

Jetzt

- Mandantenfähigkeit
- Autoscale (up and down)
- Layers reduzieren
- Micro Services (!)
- Priorisiere Workloads

Zukunft

- Time Shifting
  - Carbon Aware async Jobs
- Demand Shaping
  - Full Features mit grüner Energie

«Always on is unsustainable» ~Paul Johnston

“There is nothing so useless as doing efficiently that which should not be done at all.”

~ Peter Drucker

Design

- Rückwärtskompatibel
- LTS > 10 Jahre
- Auslastung erhöhen
  - Cloud
  - Asynchron
  - Multi-threading

Energy efficiency of programming languages

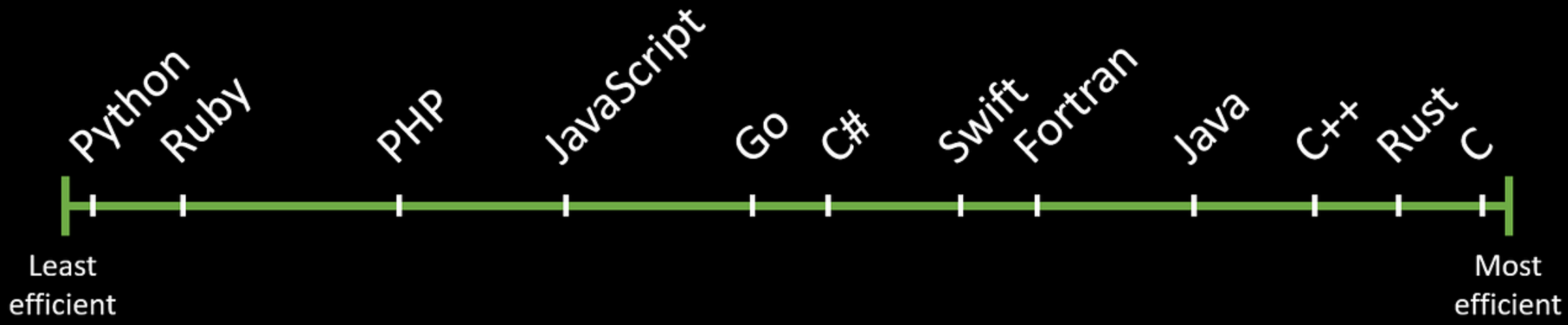


Illustration by Kasper Groes Albin Ludvigsen. Source: *Source: Pereira, R. et al. (2017)*

Optimieren

Ziel:

Möglichst wenig Energie aus fossilen Brennstoffen

- Skalen Effekt
- Custom Code < Libraries < Better Libraries
- *f(cpu cycles) ≈ f(energy)*
- Minimiere die benötigte Hardware
- Achtung
  - Jevons Paradox
  - Dev Productivity Hit
  - Kosten!

Build (if you must)

Ändere die Anforderungen

Custom Code < Bibliotheken < Bessere Bibliotheken

Optimiere Speicherung und Daten Abfragen

Reduziere Netzwerk Verkehr

Push-to-Client

Benutze performantere Programmiersprachen

- Dev / Tooling
  - AI
    - Carbon
    - Energy
  - Cloud based
    - AWS
    - Azure
    - Google
    - Multicloud
  - Code based
  - General purpose
    - Emissions
    - Energy
  - OS based
    - Android
    - Linux
    - Web
    - Windows

Tools

https://github.com/Green-Software-Foundation/awesome-green-software/tree/dev

Fazit

- Verlängere Lebensdauer von Hard- und Software
- Erhöhe Auslastung der Server
- Reduziere Anforderungen an Performance und Datenhaltung

Referenzen

- https://greenlab.di.uminho.pt/wp-content/uploads/2017/09/paperSLE.pdf
- Building Green Software, by Anne Currie, Sarah Hsu, Sara Bergman, Release Date July 2024, O'Reilly Media, Inc., ISBN 9781098150563
- Green Software Foundation
- https://ciandt.com/ca/en-ca/article/climate-crisis-and-technology-sector
- https://climate.nasa.gov/

Herzlichen Dank für Deine Aufmerksamkeit!