university of groningen

SE AR CH

Green(er) software for green(er) clouds:
environmental sustainability in cloud-based software systems

HPC & The Roaring 20s of Computing

on the occasion of Henri Bal's retirement symposium

May 2025

Vasilios Andrikopoulos

Associate Professor

v.andrikopoulos@rug.nl

university of groningen

SE AR CH

Take home messages

You can’t manage what you don’t measure *correctly*

Lots of ground to be covered in greening cloud software

university of groningen

SE AR CH

Sustainability

university of groningen

SE AR CH

The Brundtland Report aka Our Common Future

27. 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***. The concept of sustainable development does imply limits - not absolute limits but limitations imposed by the present state of ***technology and social organization on environmental resources*** and by the ability of the biosphere to absorb the effects of human activities. But technology and social organization can be both managed and improved to make way for a new era of economic growth. The Commission believes that widespread poverty is no longer inevitable. Poverty is not only an evil in itself, but sustainable development requires meeting the basic needs of all and extending to all the opportunity to fulfil their aspirations for a better life. A world in which poverty is endemic will always be prone to ecological and other catastrophes.

university of groningen

SE AR CH

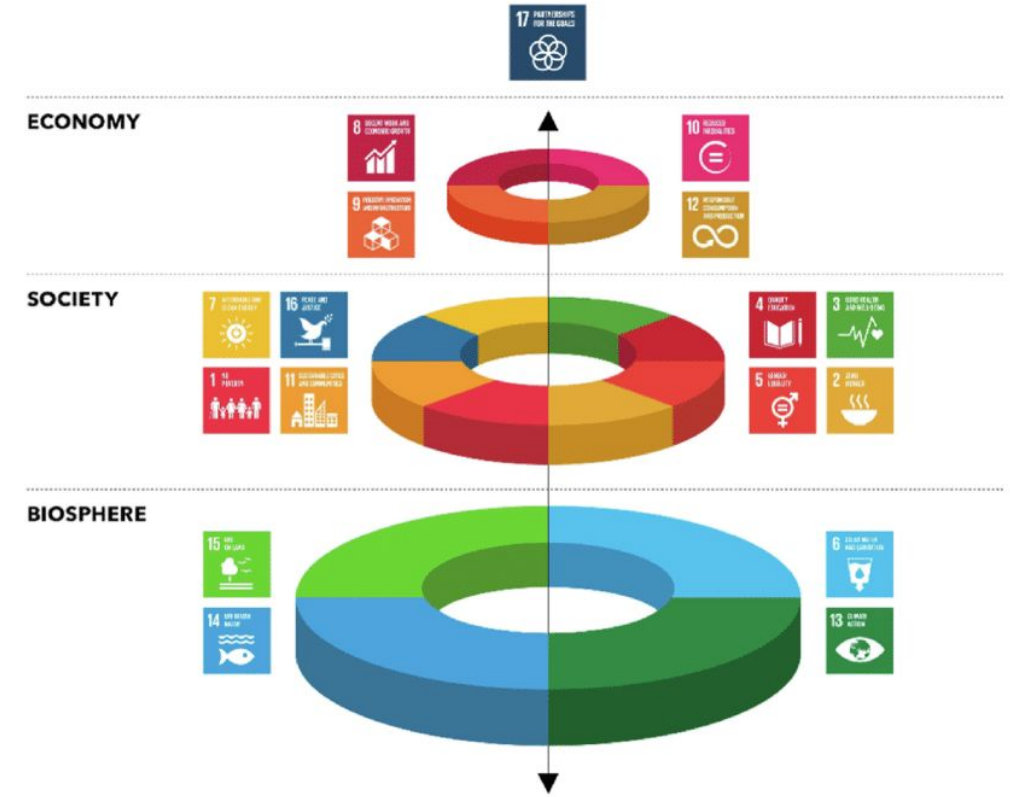
Sustainable Development Goals (SDGs)

university of groningen

SE AR CH

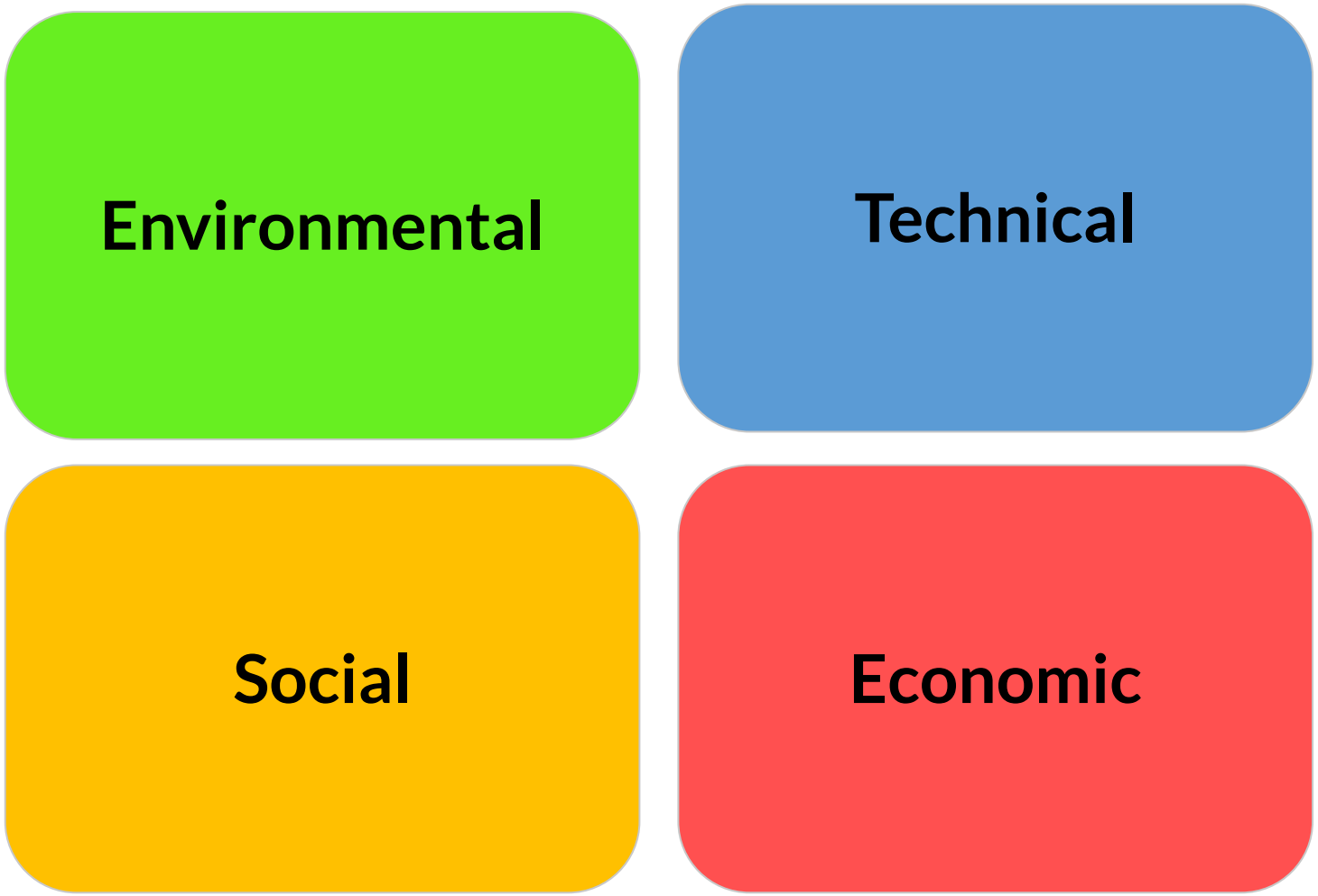
Sustainability Pillars

1. Environmental
2. Societal
3. Economical



Source: [Folke et al. 2016]

Sustainability as a software quality



Following Lago et al. 2015

university of groningen

SE AR CH

Environmental sustainability

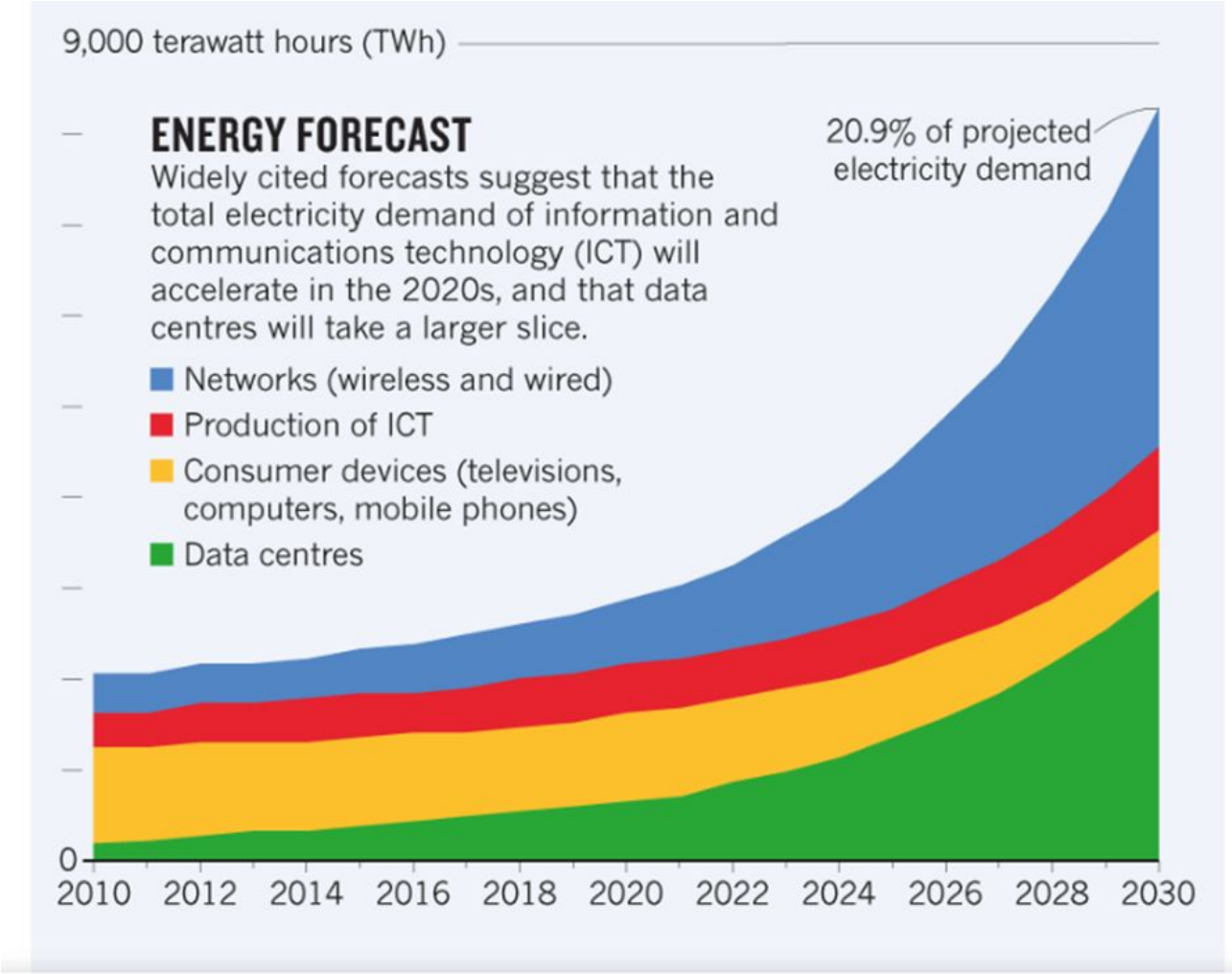
university of groningen

SE AR CH

Environmental impact

Emissions aka
Carbon footprint =

Energy consumption x *Carbon Intensity* + *Inherent emissions*



Source: [Jones 2018]

Measuring and reporting emissions

GreenHouse Gas (GHG) Protocol as the ***de facto* reporting** standard

Scope 1	Directly attributable emissions e.g. running generators
Scope 2	Indirect but controlled emissions e.g. electricity
Scope 3	Indirect but not under control emissions e.g. transportation and disposal of equipment

university of groningen

SE AR CH

Carbon footprint in multi-tenant environments

- How to allocate emissions among tenants of the same service?

What are the Scope 3 emissions attributable to the tenants?

What to include in these emissions?

- How to measure the emissions of a service deployed on the cloud/in a DC?

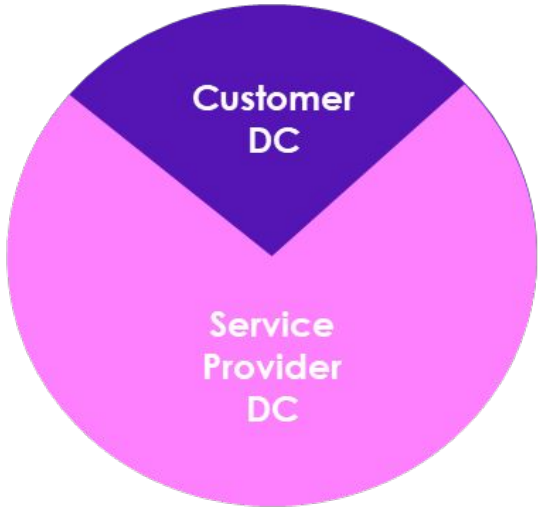
university of groningen

SE AR CH

Collaboration with BT Global Services

Total Energy use = %age of Kilowatt Hours in 6 Data Centers

Measurable Energy Use Distribution



Carbon Emissions Data Fully Measurable

university of groningen

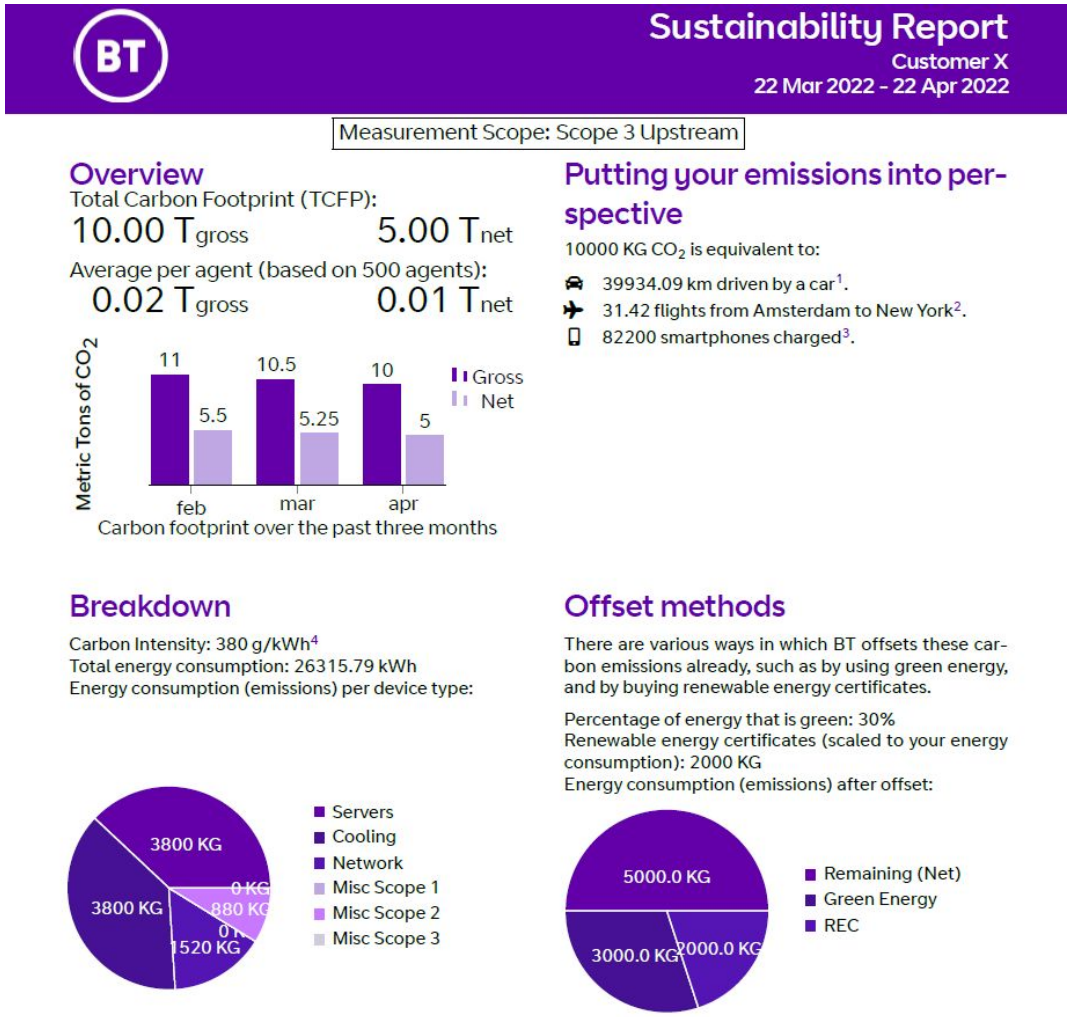
SE AR CH

Sustainability Report generator

Takes *utilization* data as input

Evaluated positively in a round of interviews with account holders

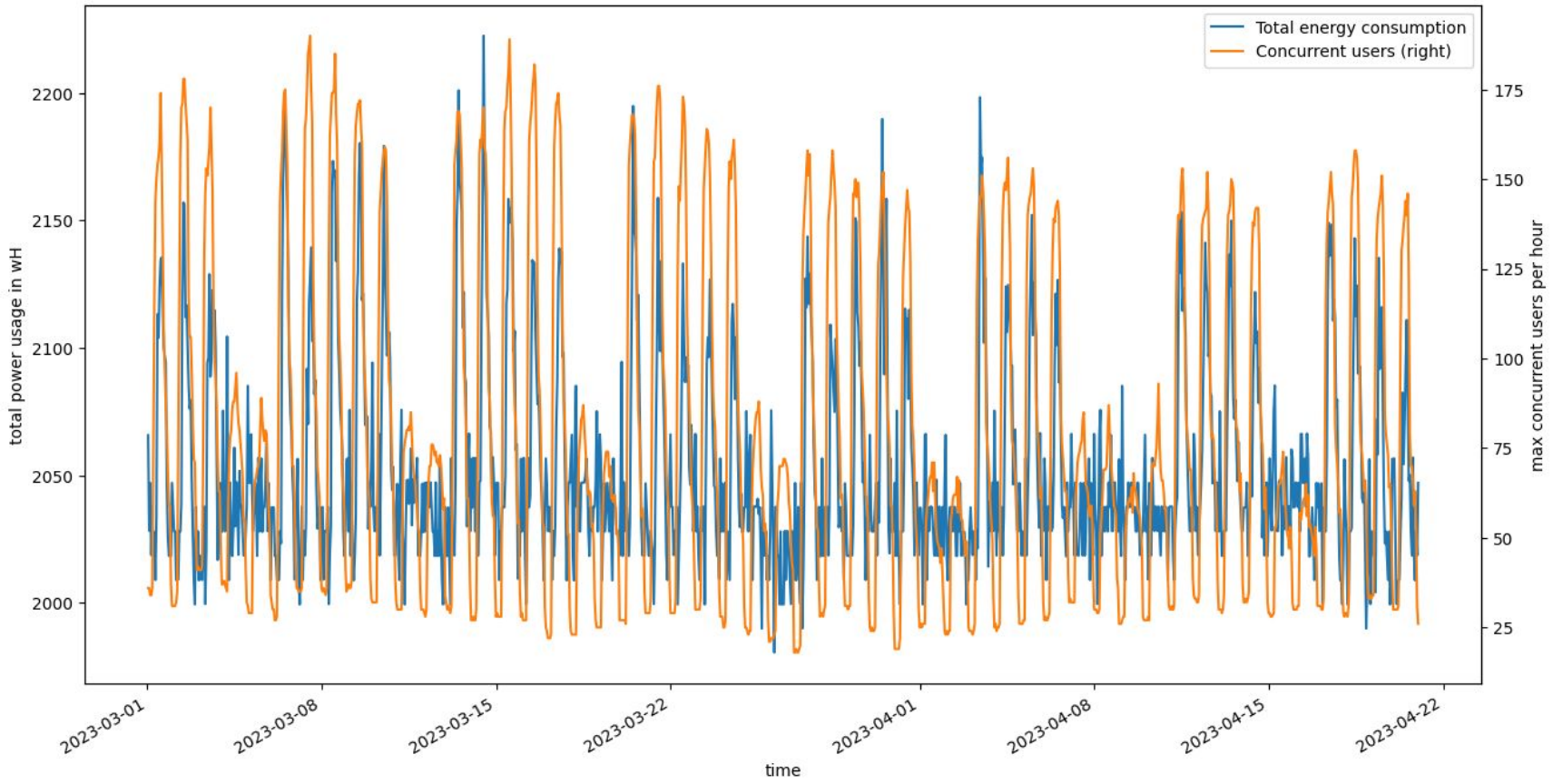
https://arxiv.org/abs/2305.10439



university of groningen

SE AR CH

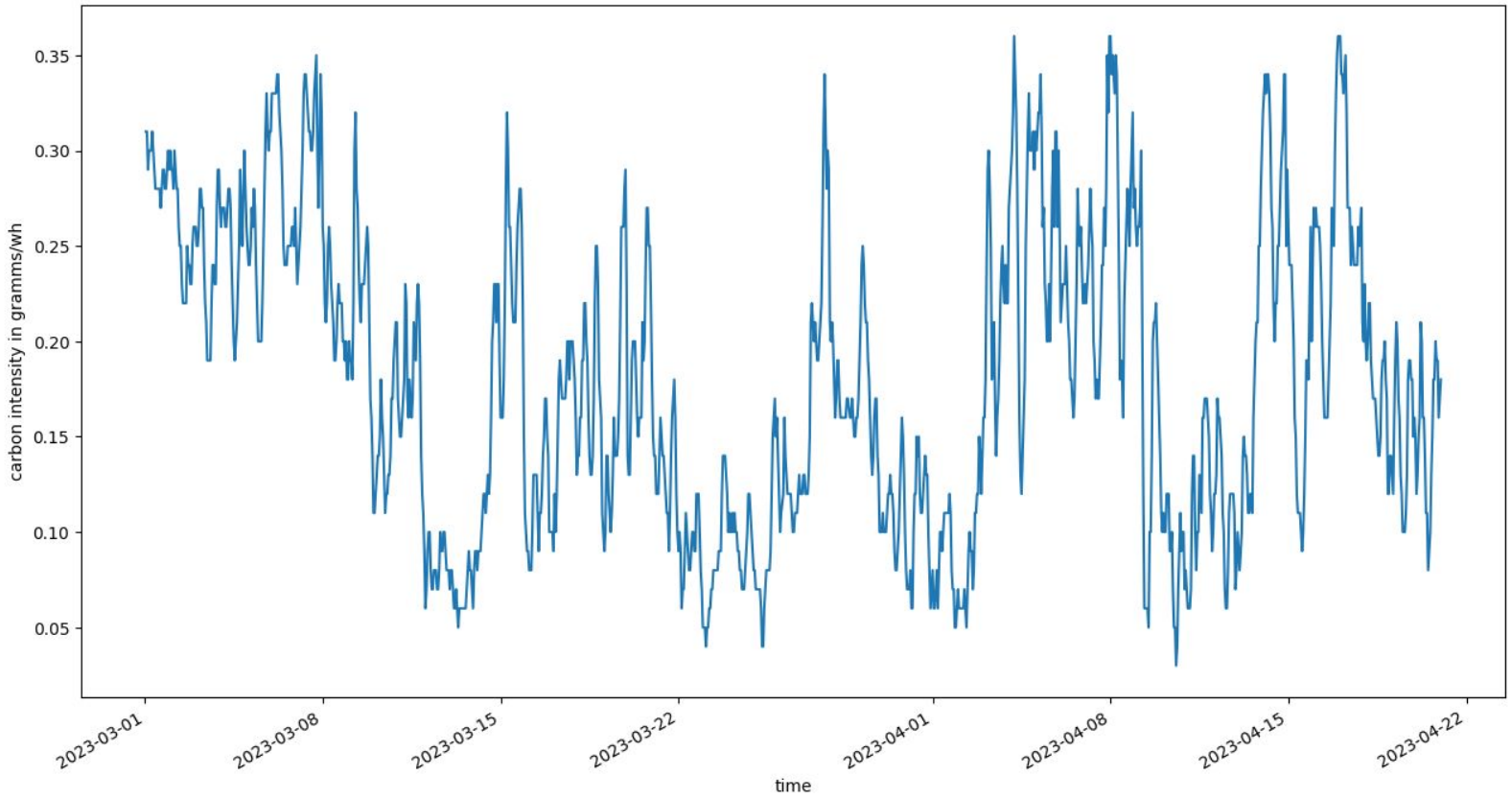
Energy consumption versus usage patterns



university of groningen

SE AR CH

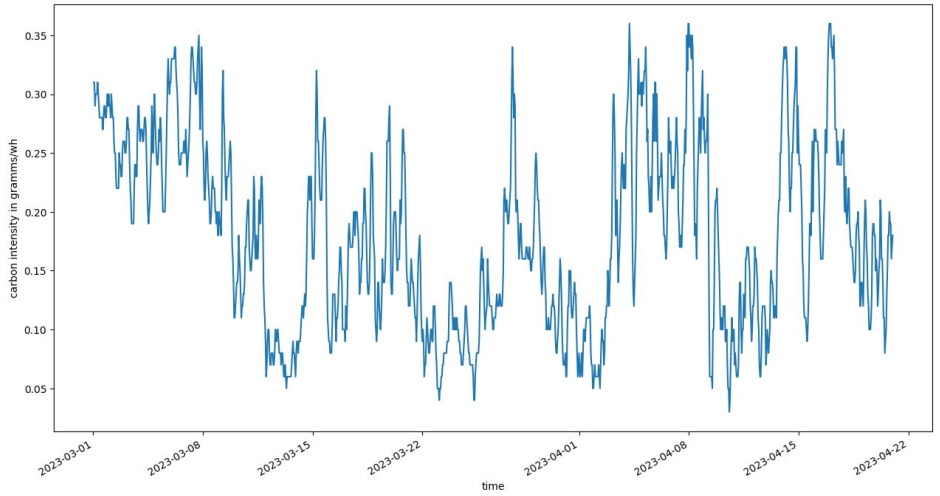
Carbon intensity as function of time

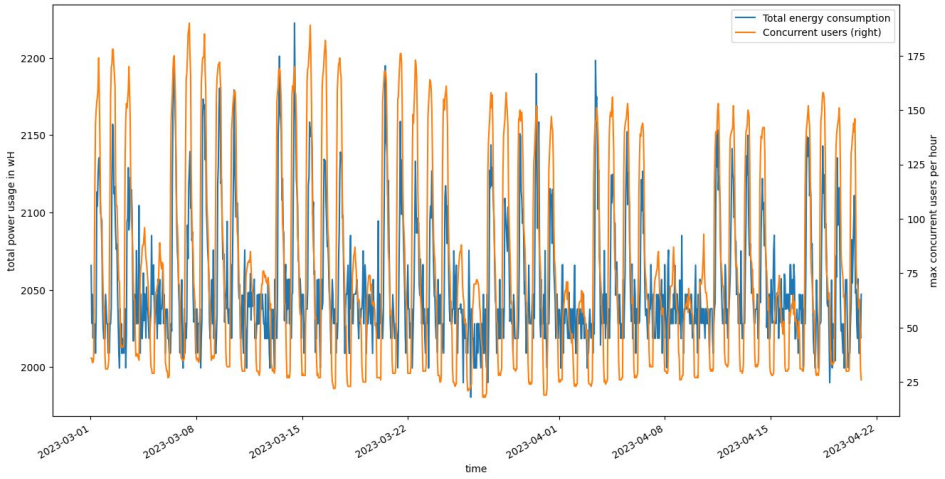


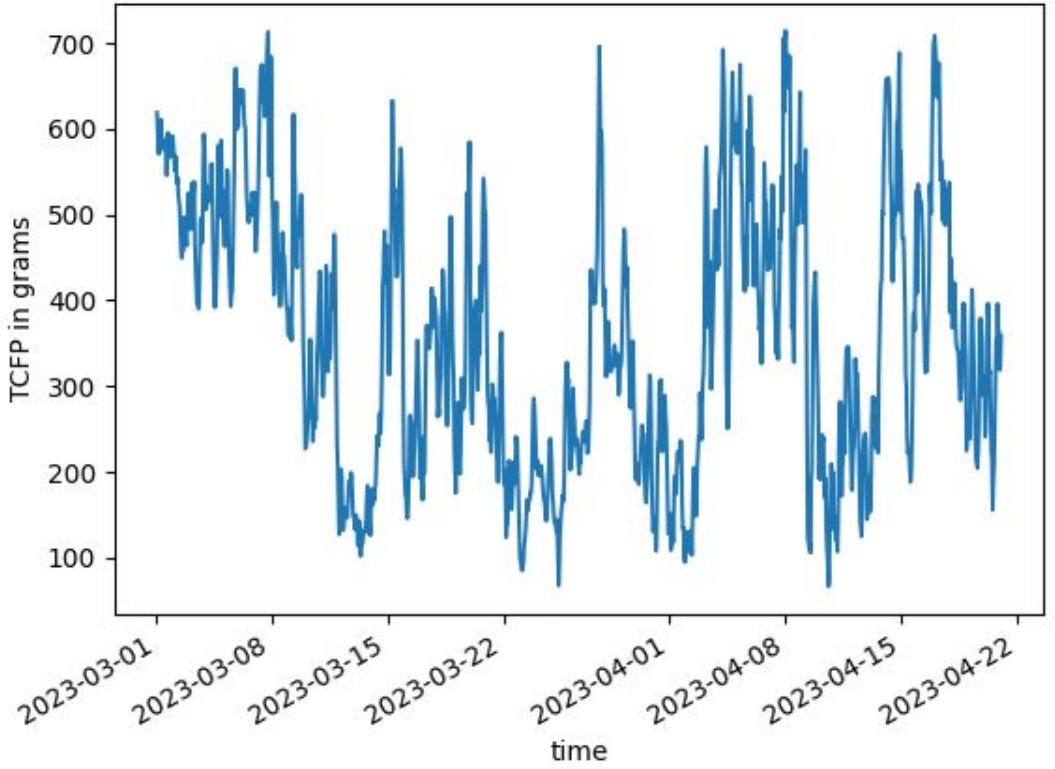
university of groningen

SE AR CH

Carbon footprint as a semi-stochastic process



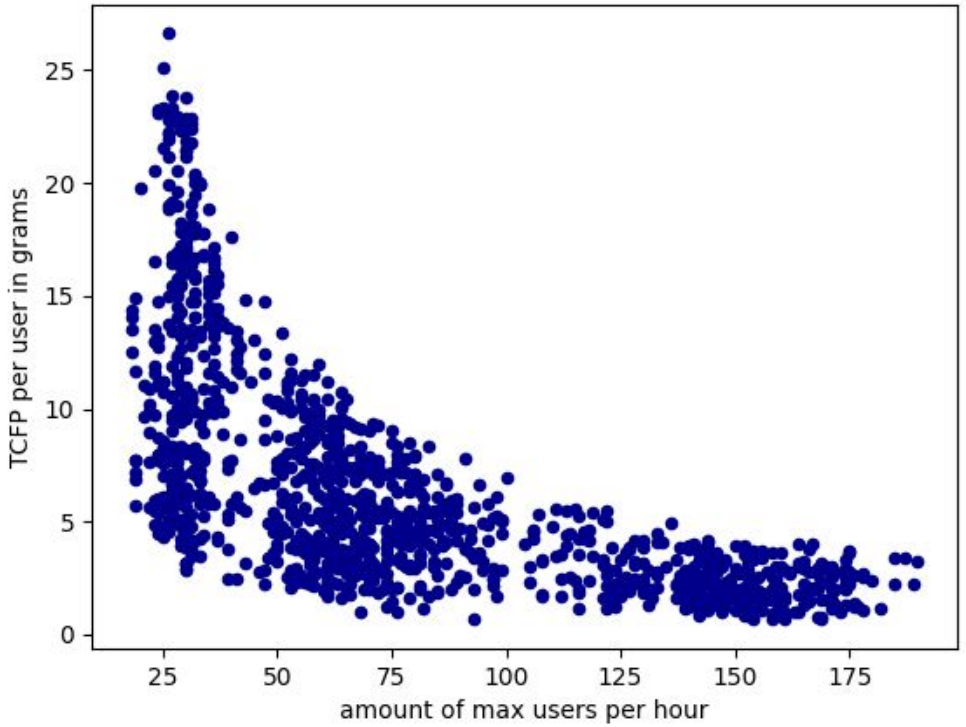


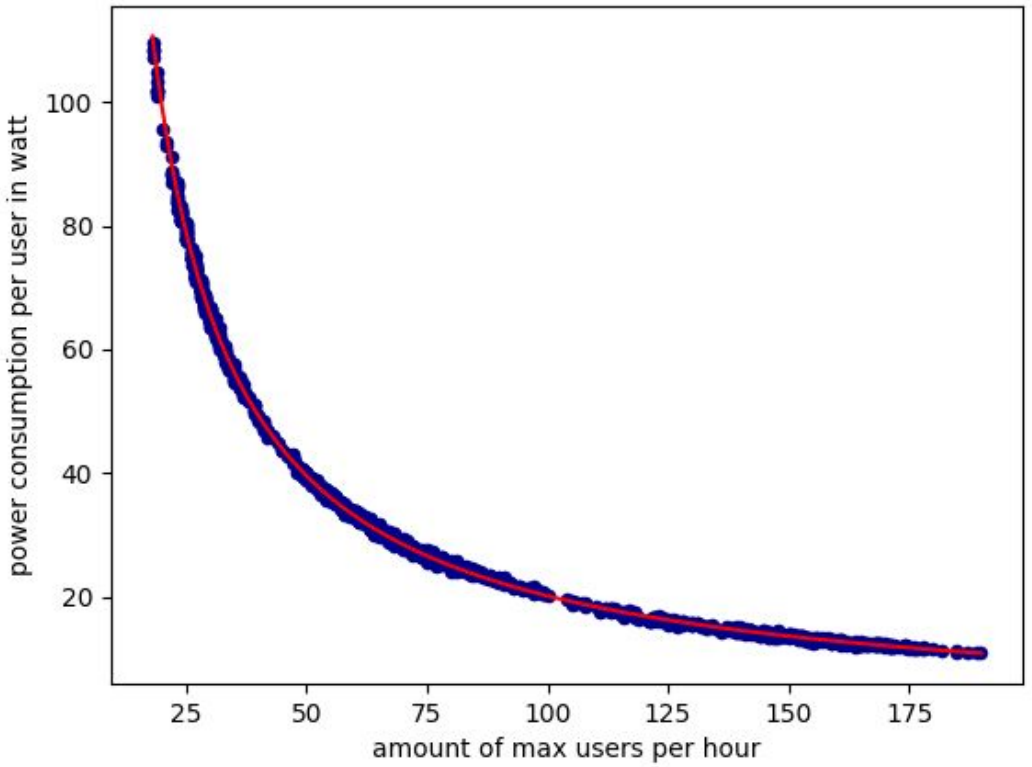


university of groningen

SE AR CH

Footprint/energy consumption per active users





~~Carbon footprint~~ Energy consumption in multi-tenant environments

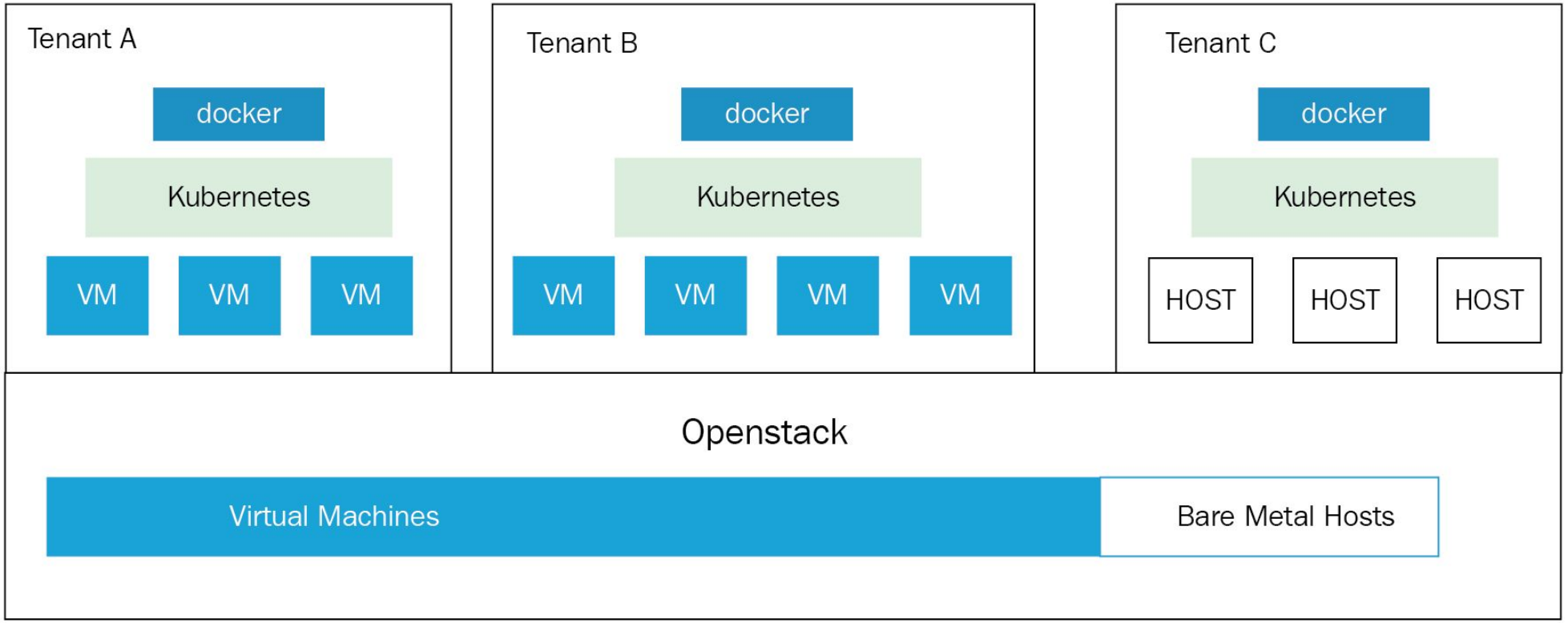
- How to allocate ~~emissions~~ energy consumption among tenants of the same service?

- How to measure the ~~emissions~~ energy consumption of a service deployed on the cloud/in a DC?

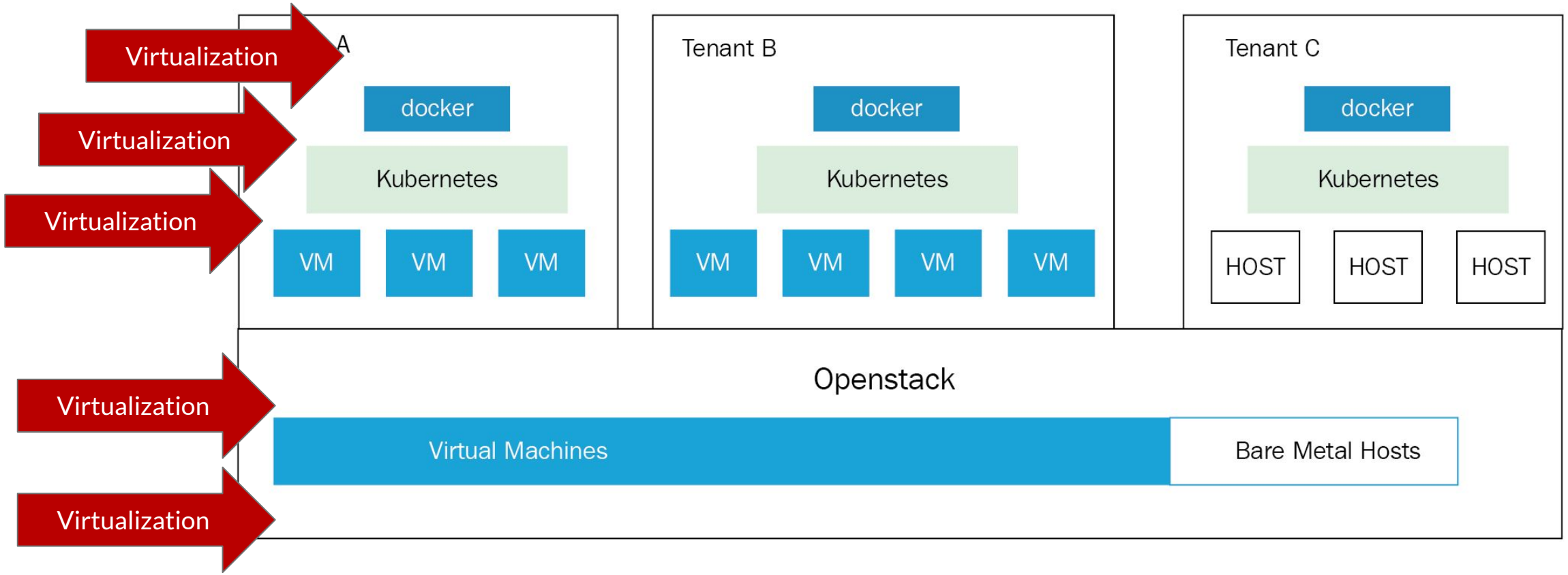
How to cut through all the virtualization layers?

What is the “platform” overhead?

Virtualization layers in a Cloud-native application



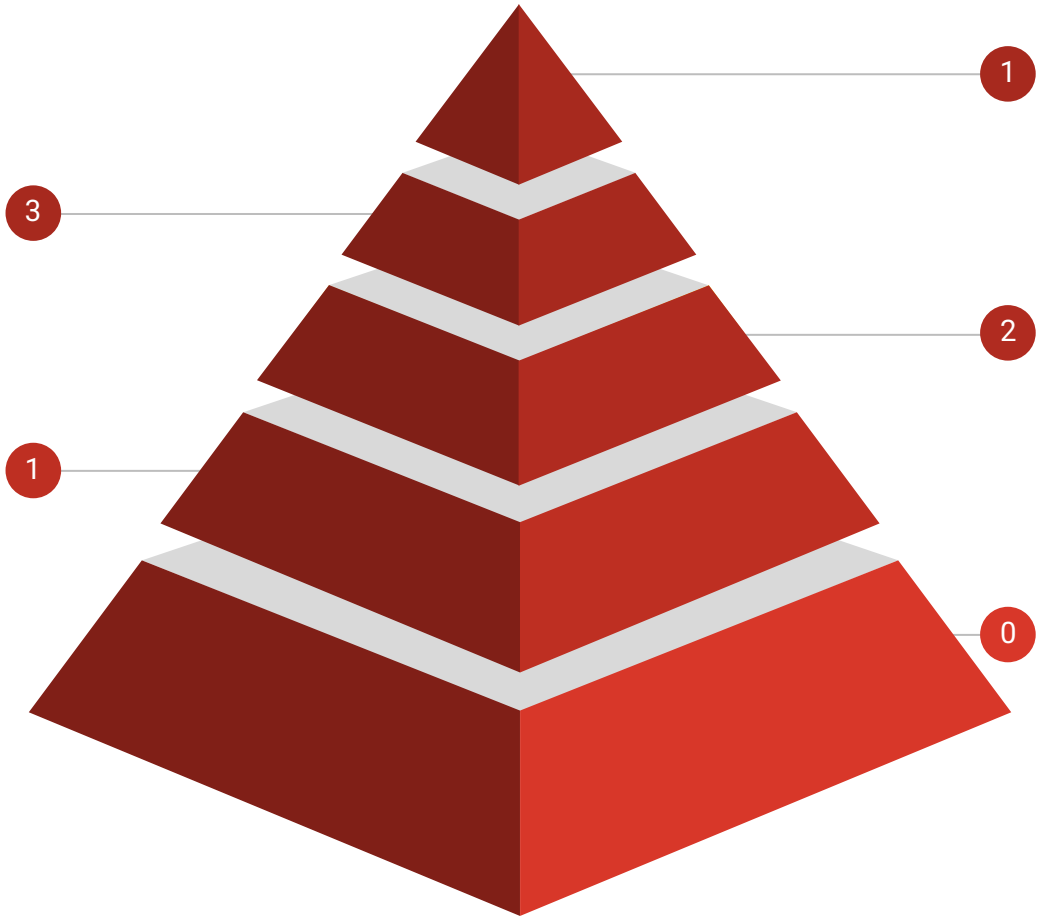
Virtualization layers in a Cloud-native application



university of groningen

SE AR CH

Observability Stack for Cloud-native applications



Application 1
OpenTelemetry demo application

OpenStack + VM 2
ineffective tooling, no direct measurements for OpenStack
Scaphandre for VMs

Physical 0
through PowerPlug measurements

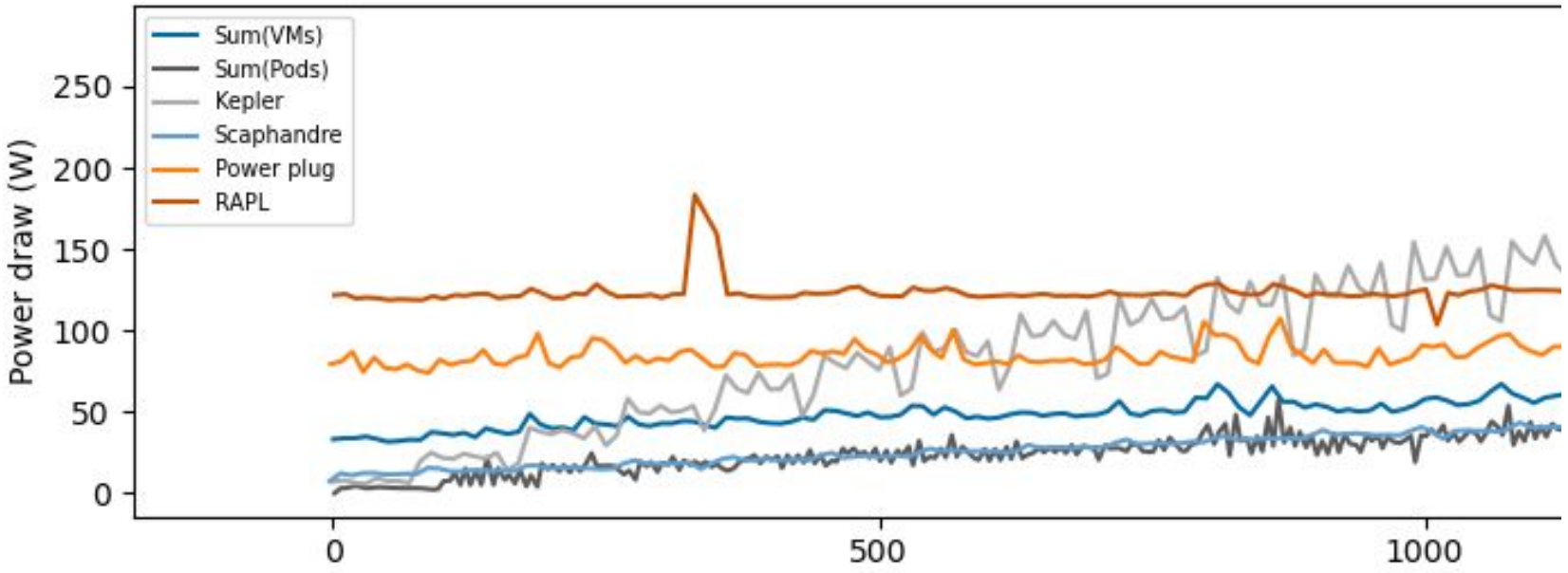
Kubernetes 3
cAdvisor, Kepler

Bare metal 1
node_exporter + RAPL

university of groningen

SE AR CH

Energy consumption distribution in the cluster

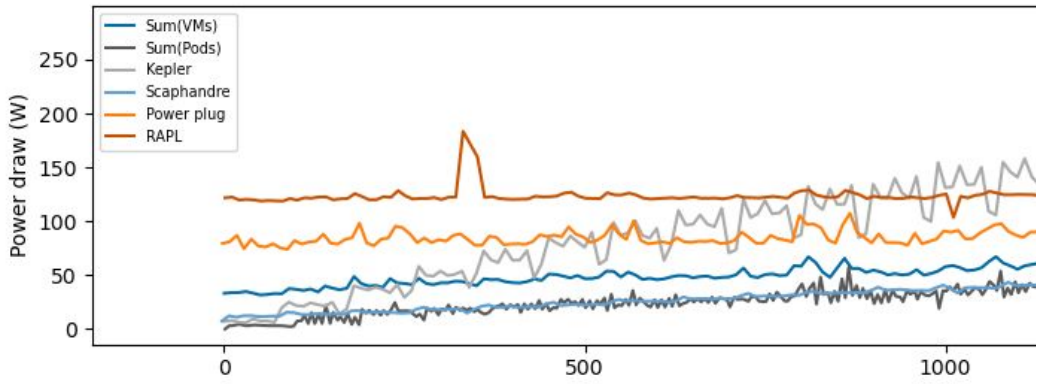


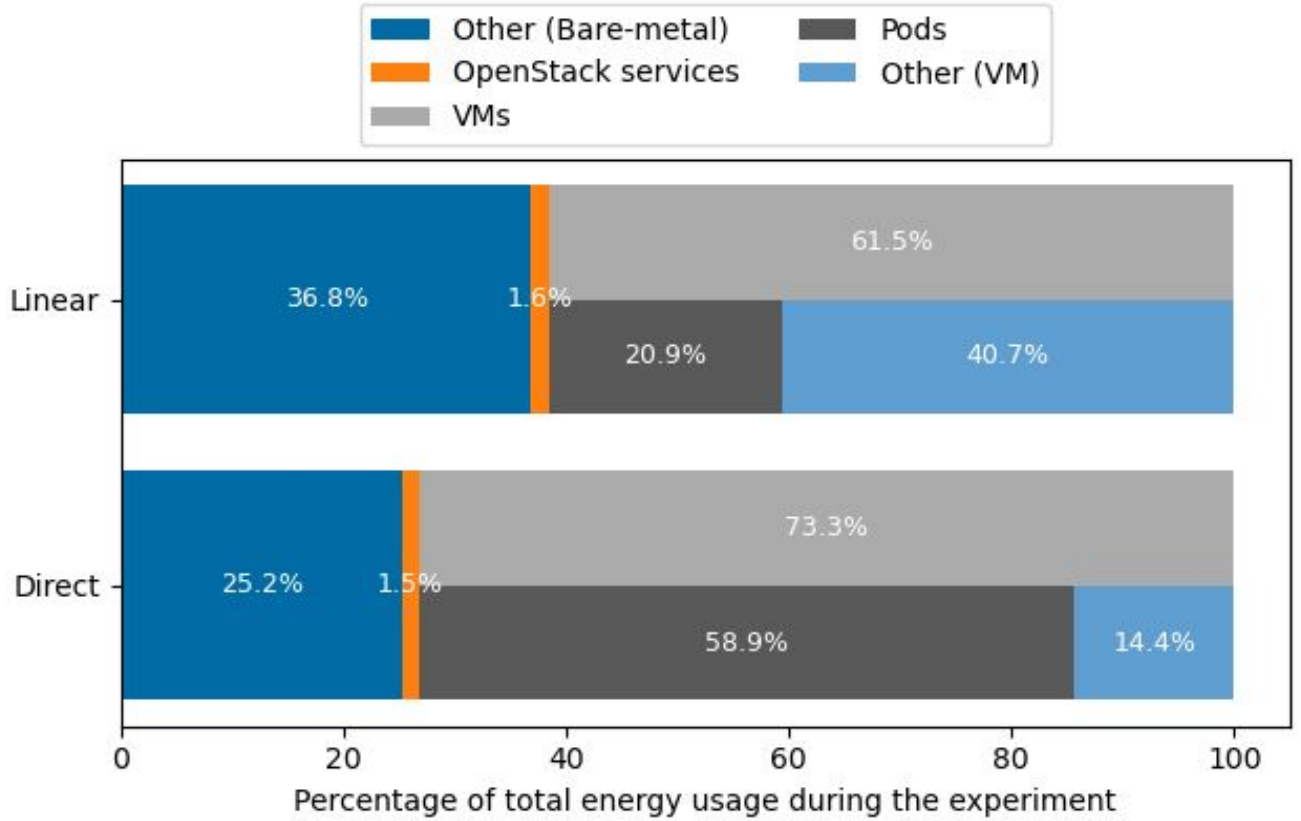
https://doi.org/10.1145/3676151.3719371

university of groningen

SE AR CH

Energy consumption distribution in the cluster





https://doi.org/10.1145/3676151.3719371

university of groningen

SE AR CH

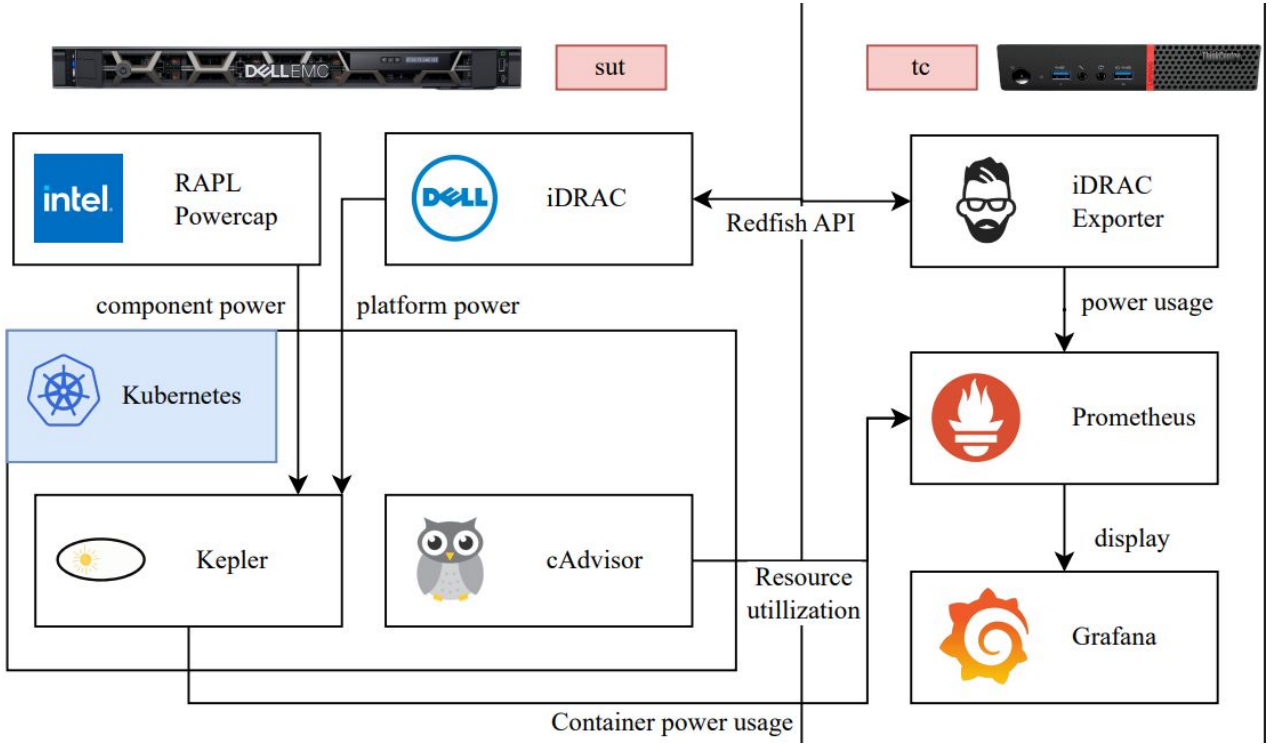
Digging deeper in Kubernetes

Kepler

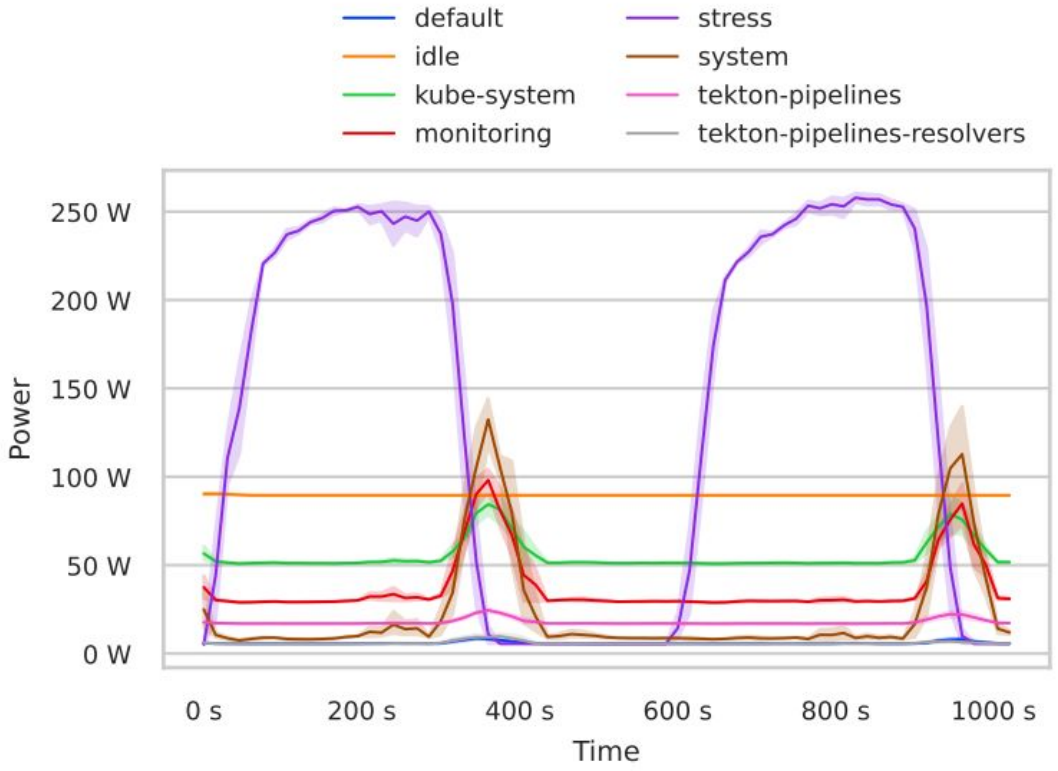
Kepler (Kubernetes-based Efficient Power Level Exporter) uses eBPF to probe energy related system stats and exports as Prometheus metrics.

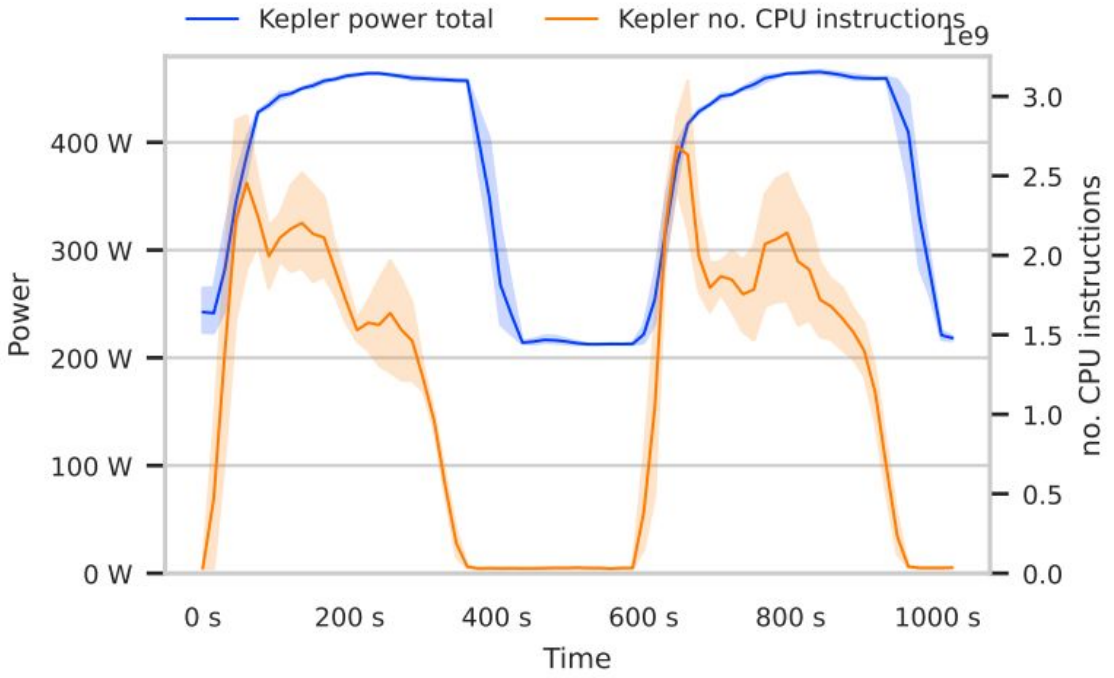
Kepler was accepted to CNCF on May 17, 2023 at the **Sandbox** maturity level.

VISIT PROJECT WEBSITE



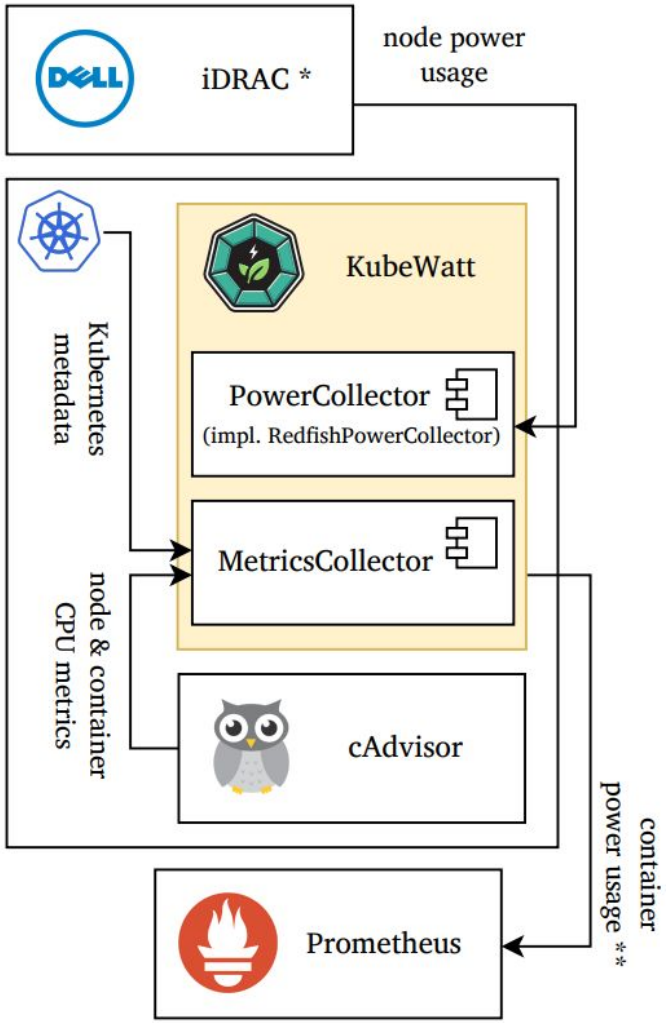
The problem(s) with Kepler

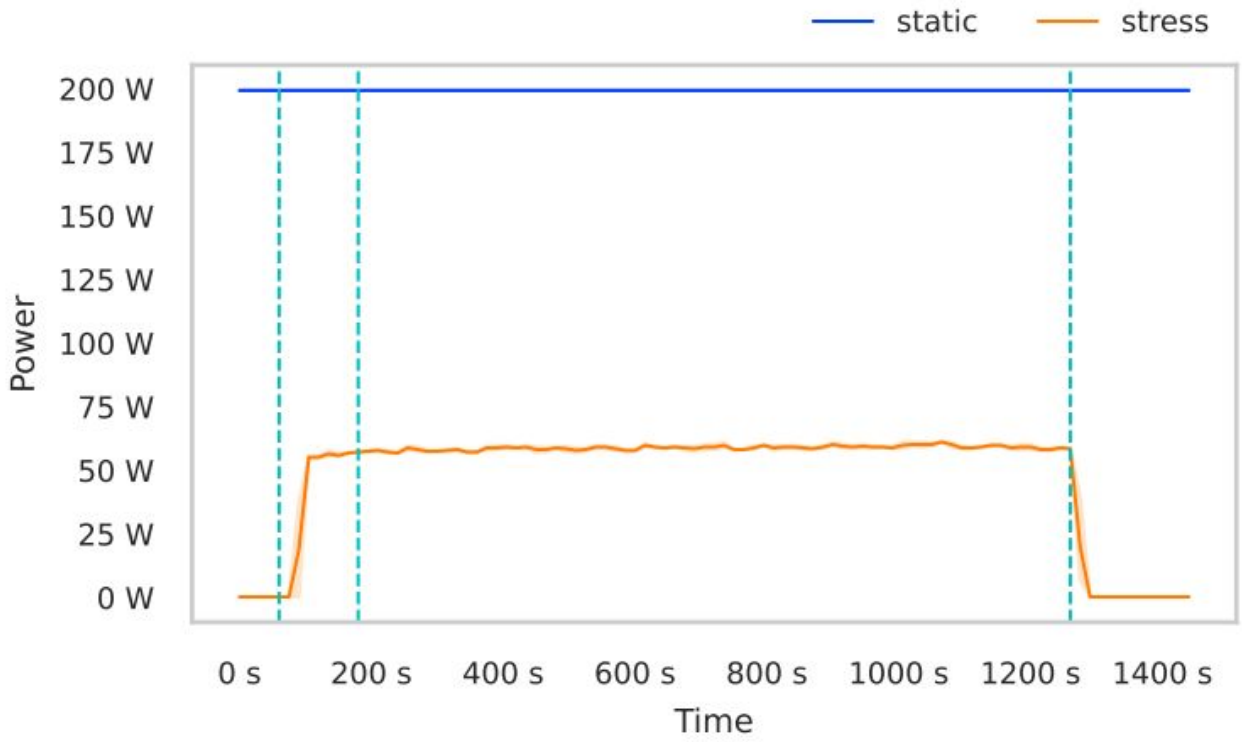




https://arxiv.org/abs/2504.10702

KubeWatt to the rescue





https://github.com/bjornpijnacker/kubewatt

In summary

university of groningen

SE AR CH

Take home messages

You can't manage what you don't measure *correctly*

- Measuring emissions/energy in the cloud is a wicked problem

Lots of ground to be covered in greening cloud software

- Improvements required in observability tools *and* platform software