How accurate are current tools and models for estimating software energy consumption?

Andreas Brunnert[1], Manuel Steinberg[2]

1) Professor @ University of Applied Sciences Munich HM
Founder @ RETIT GmbH

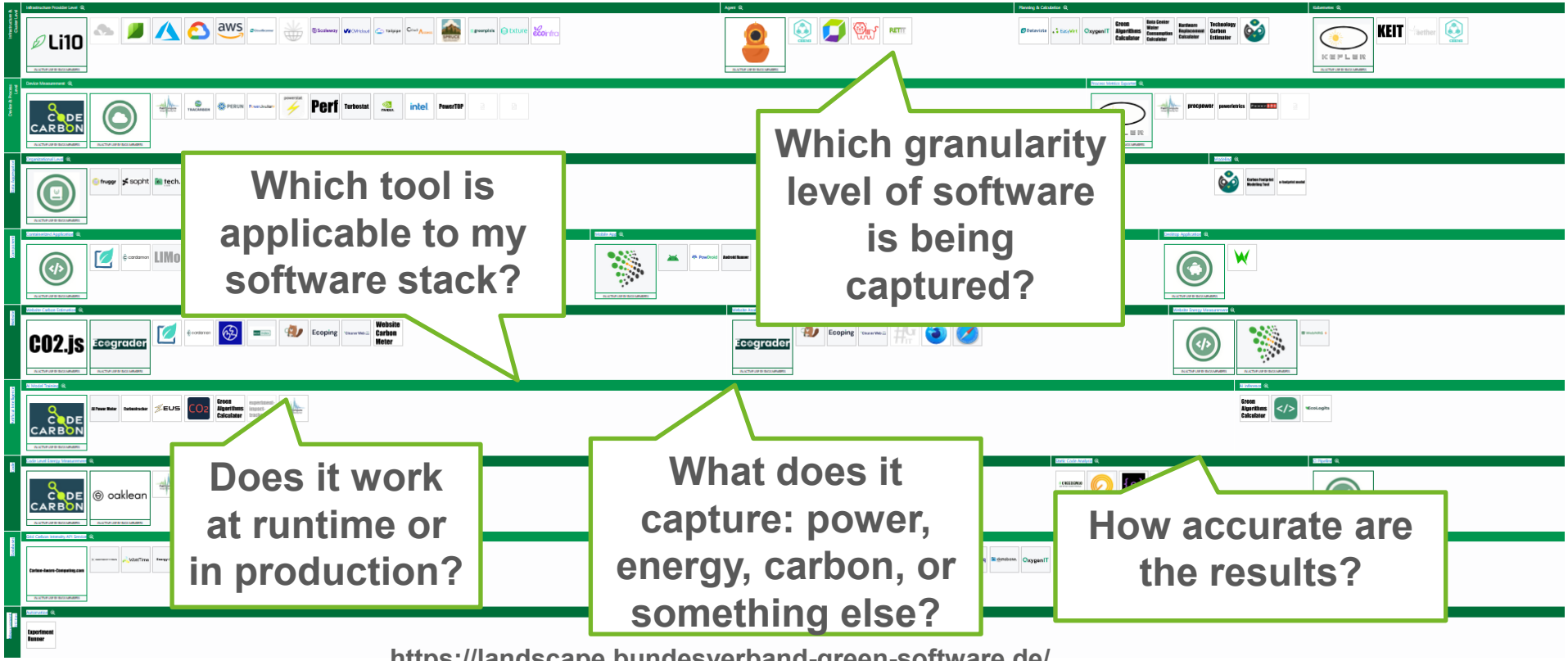
2) PhD Student @ University of Applied Sciences Munich HM

EcoCompute 2025

Agenda

1. Software Energy Consumption
2. Experiment Setup
3. Experimental Results
4. Tool Overhead
5. Summary and Conclusions

Software Energy Consumption



Which tool is applicable to my software stack?

Which granularity level of software is being captured?

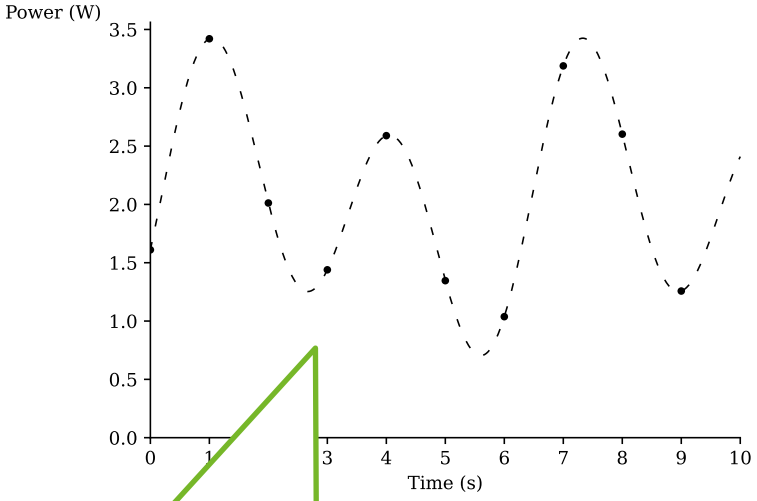
Does it work at runtime or in production?

What does it capture: power, energy, carbon, or something else?

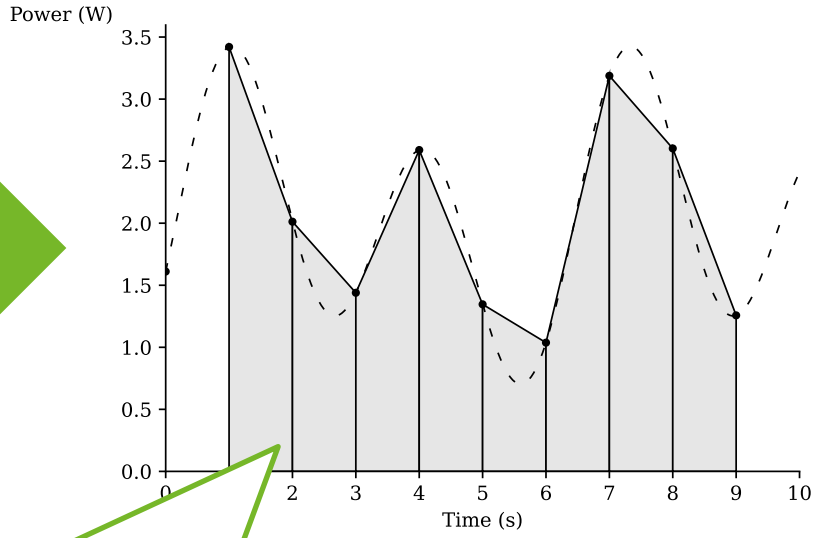
How accurate are the results?

https://landscape.bundesverband-green-software.de/

Software Energy Consumption



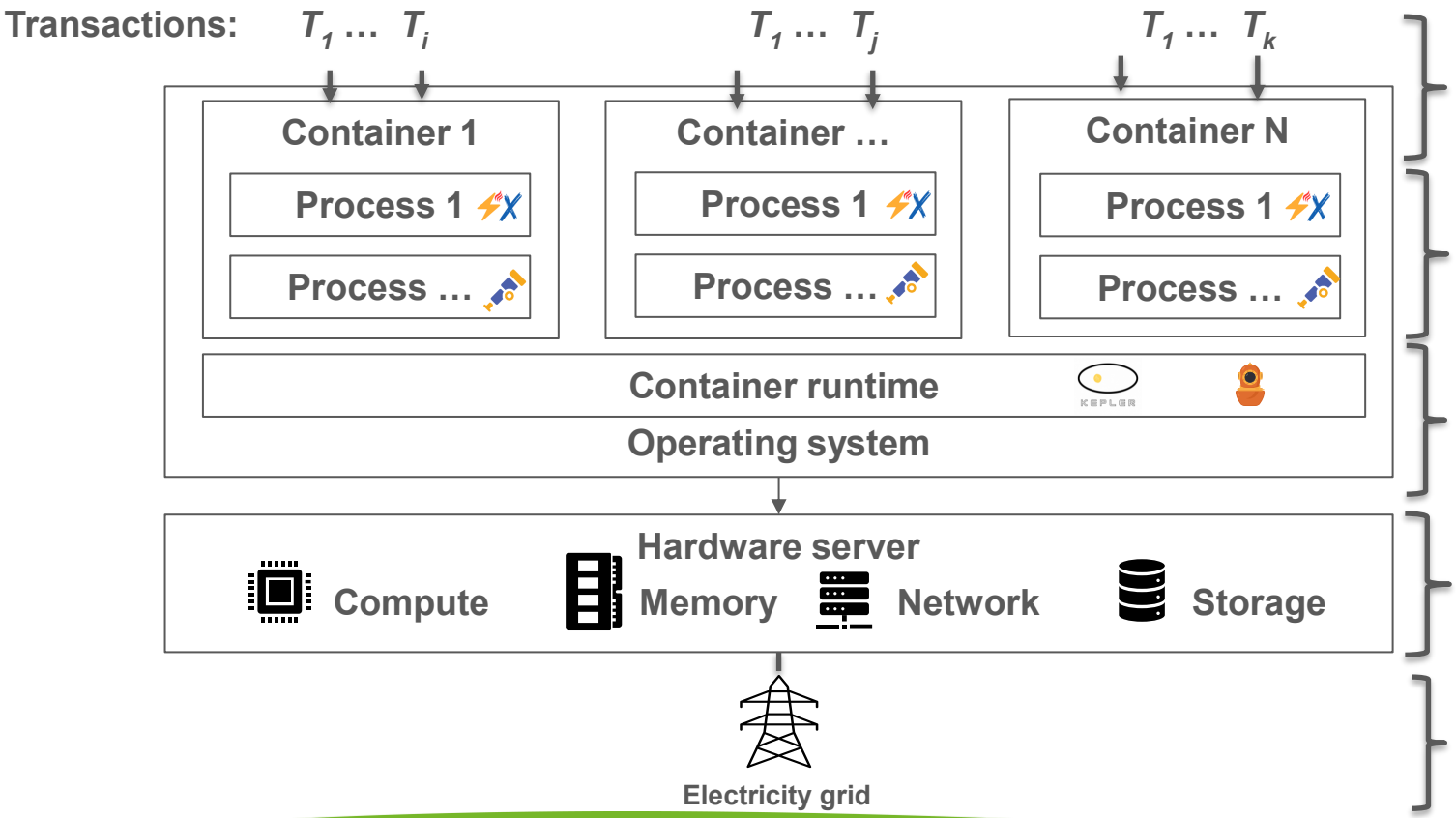
Therefore, you need to get power right in the first place



Energy consumption (Joule) approximates power over time

Source: https://luiscruz.github.io/2023/05/13/energy-units.html

Software Energy Consumption



Transaction-level attributions (e.g., using JoularJX or the OpenTelemetry Java Agent Extension (OTJAE))

Process-level attributions (e.g., using Kepler, OTJAE, JoularJX or Scaphandre)

Container-level attributions (e.g., using Kepler or Scaphandre)

Hardware measurements (e.g., using hardware interfaces such as IPMI, Intel RAPL or Nvidia SMI)

External measurements (e.g., using energy meters between the power plug and the server)

Tool Overview

JoularJX Attribute RAPLpower values to individual transactions using utilization based formulars

OpenTelemetry Java Agent Extension (OTJAE): Derives process transaction consumption solely based on formulars. Also captures memory, disk and network

Transaction-level attributions (e.g., using JoularJX or the OpenTelemetry Java Agent Extension (OTJAE))

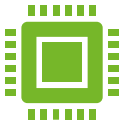
Kepler, Scaphandre, and JoularJX Attribute RAPLpower values to individual processes using different utilization based formulars

OpenTelemetry Java Agent Extension (OTJAE): Derives process power consumption solely based on formulars. Also captures memory, disk and network

Process-level attributions (e.g., using Kepler, OTJAE, JoularJX or Scaphandre)

Kepler and Scaphandre: Attribute RAPL power values to individual containers using different utilization based formulars

Container-level attributions (e.g., using Kepler or Scaphandre)



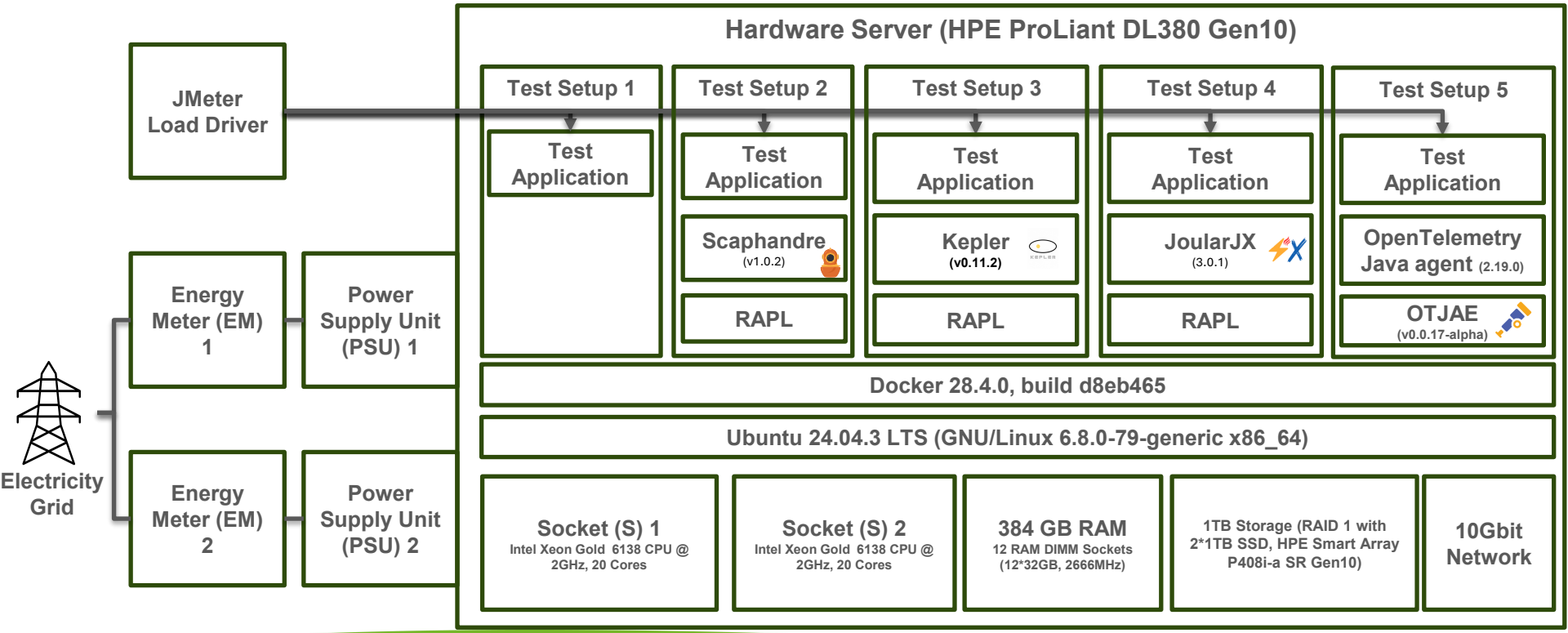
Intel **RAPL** (accessible via **Powercap**) provides continuous energy readings for the processor sockets, converted to average power values per interval.

Hardware measurements (e.g., using hardware interfaces such as IPMI, Intel RAPL or Nvidia SMI)

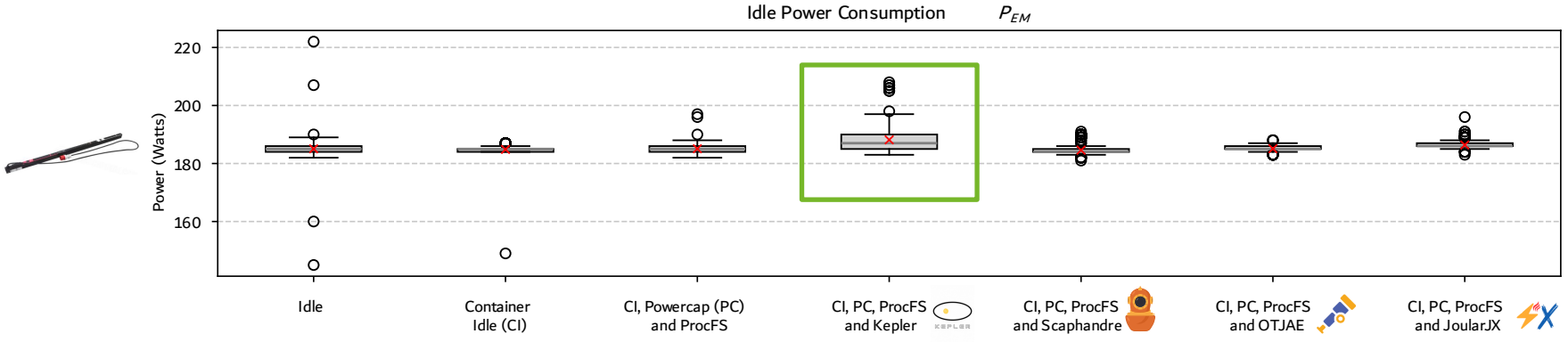
Continuous power readings for each power plug
→ **Ground truth**

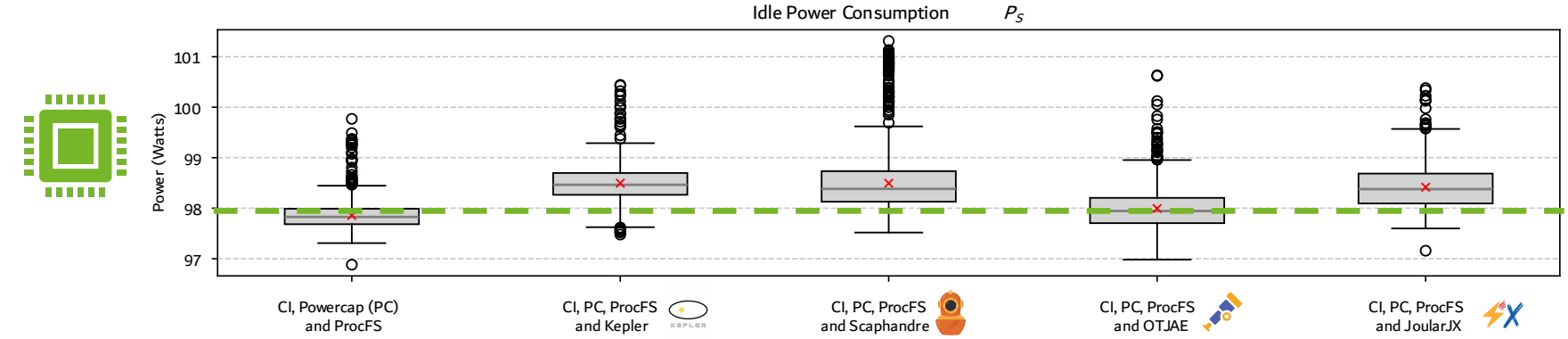
External measurements (e.g., using energy meters between the power plug and the server)

Experiment Setup

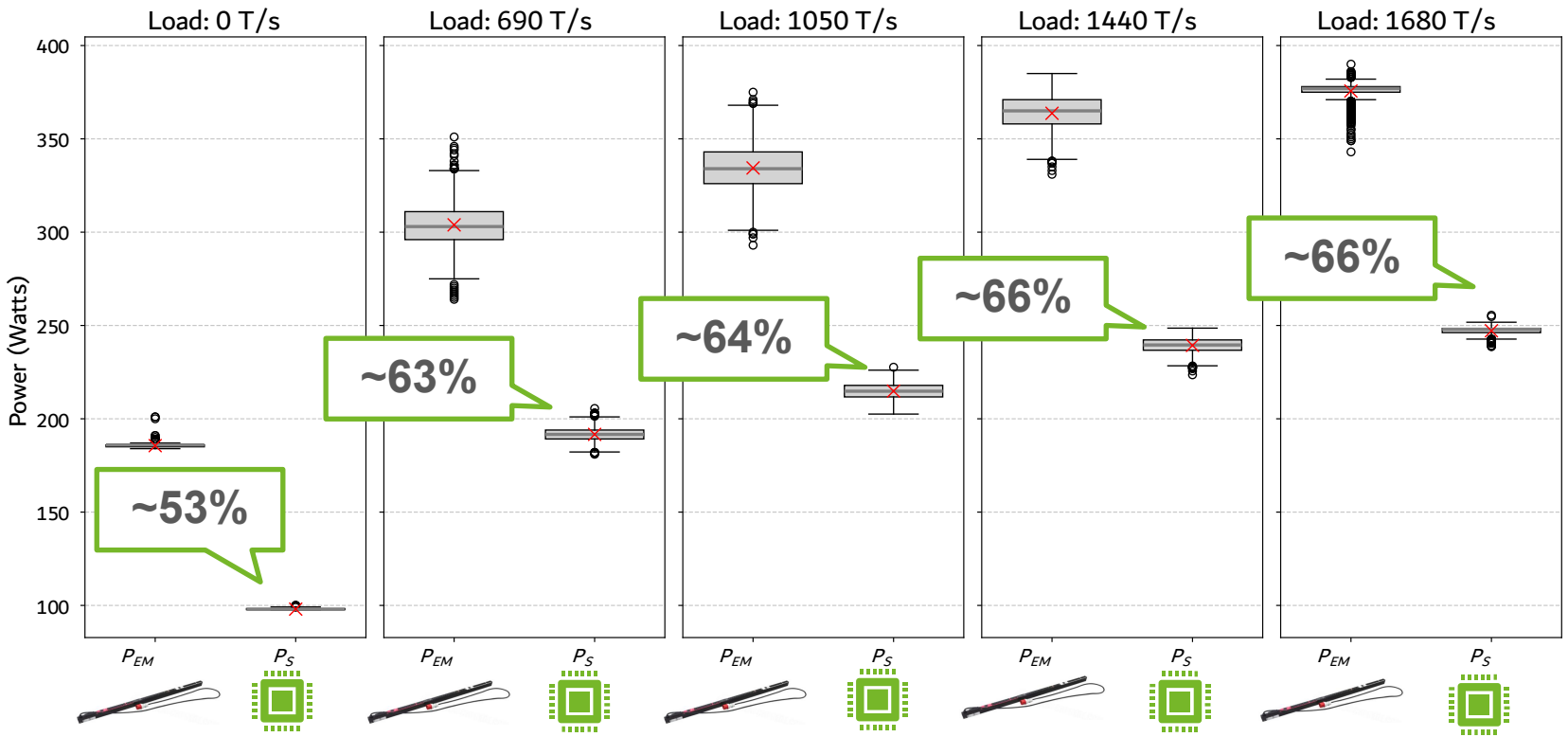


Experimental Results – Idle Measurements

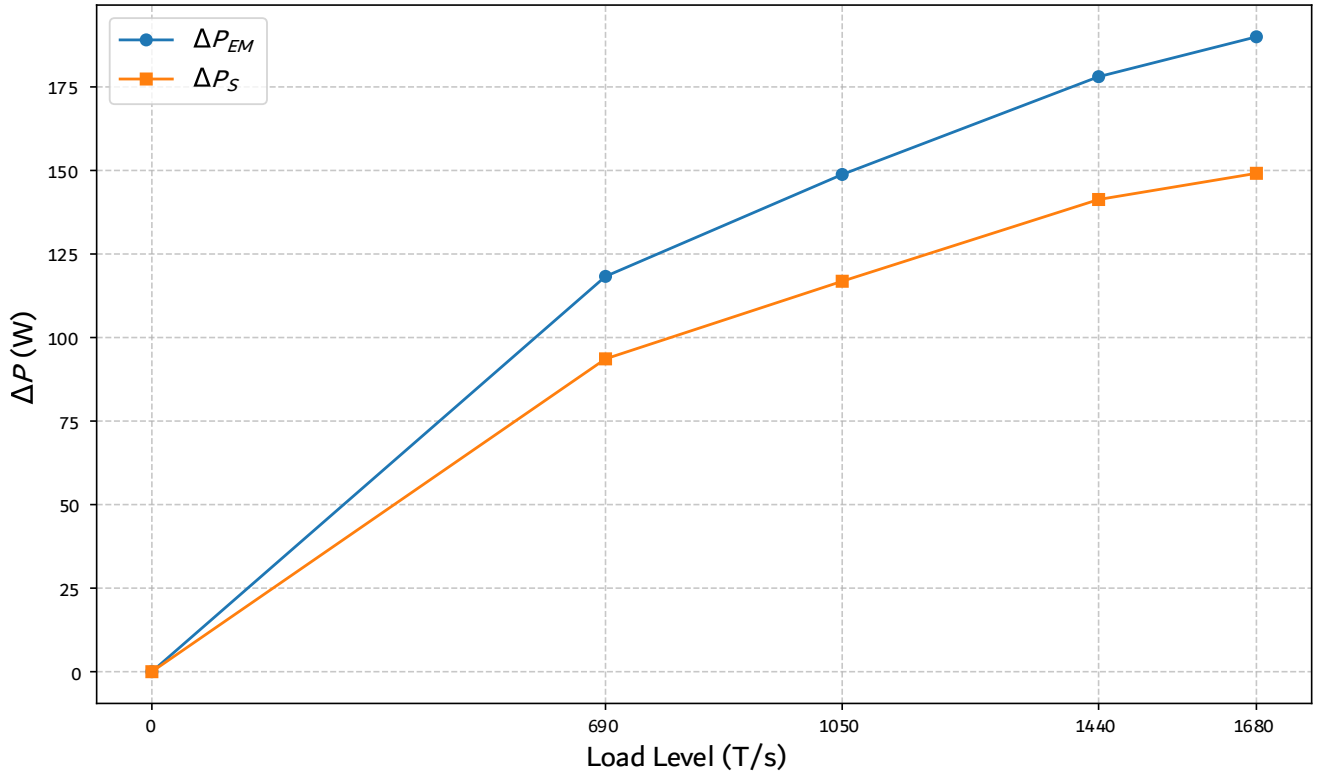


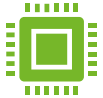


Experimental Results - External and Hardware Measurements

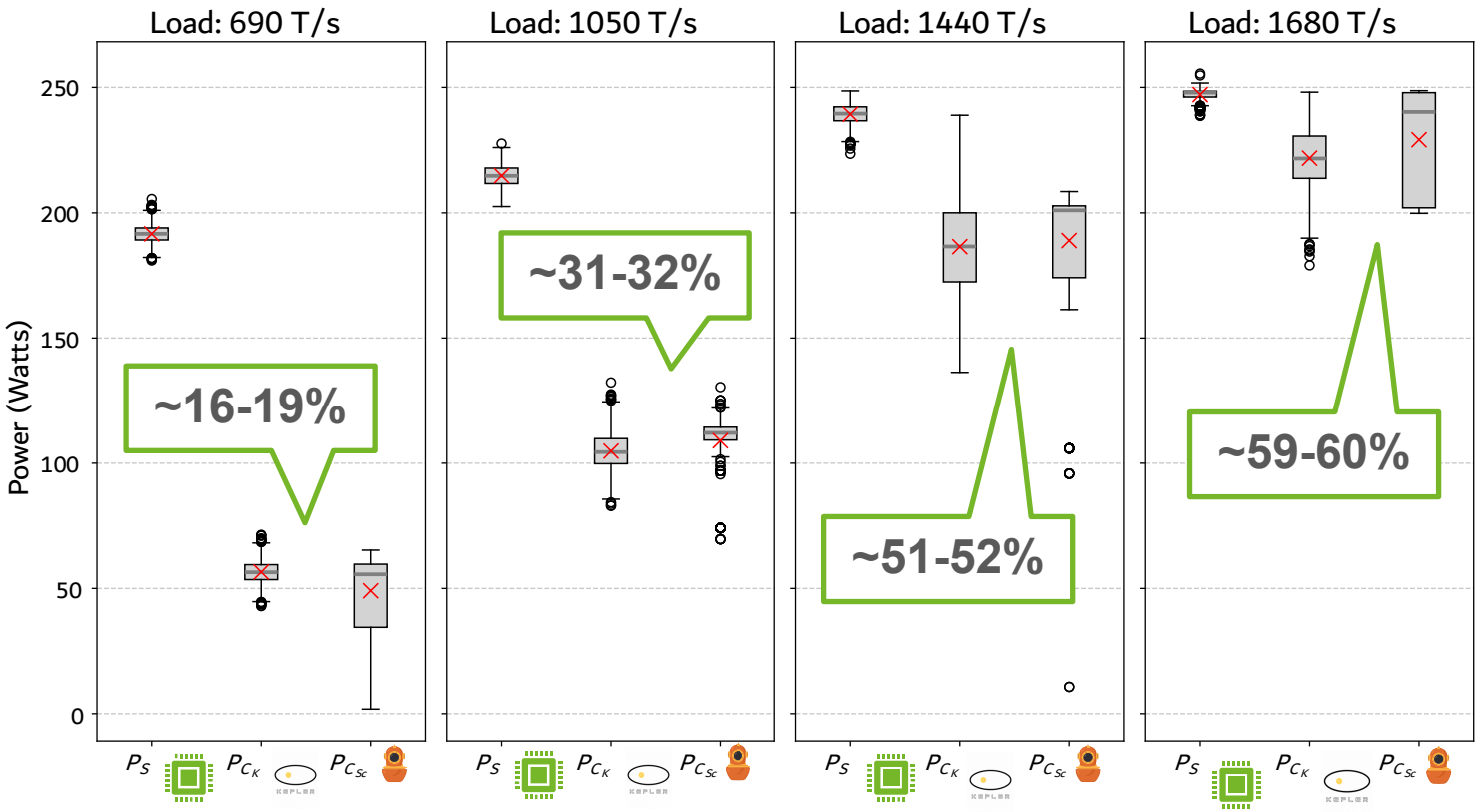


Experimental Results - External and Hardware Measurements

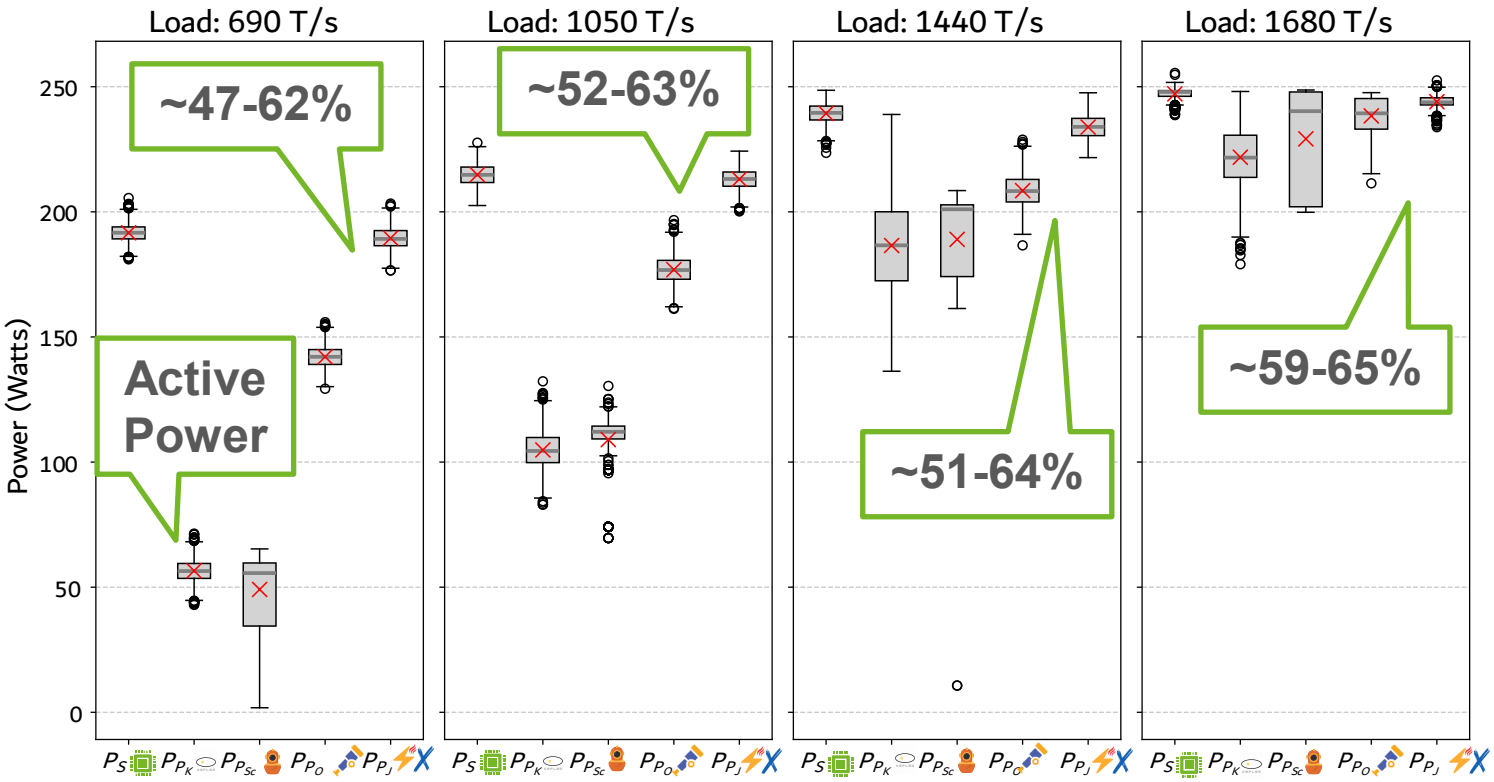




Experimental Results – Container Level



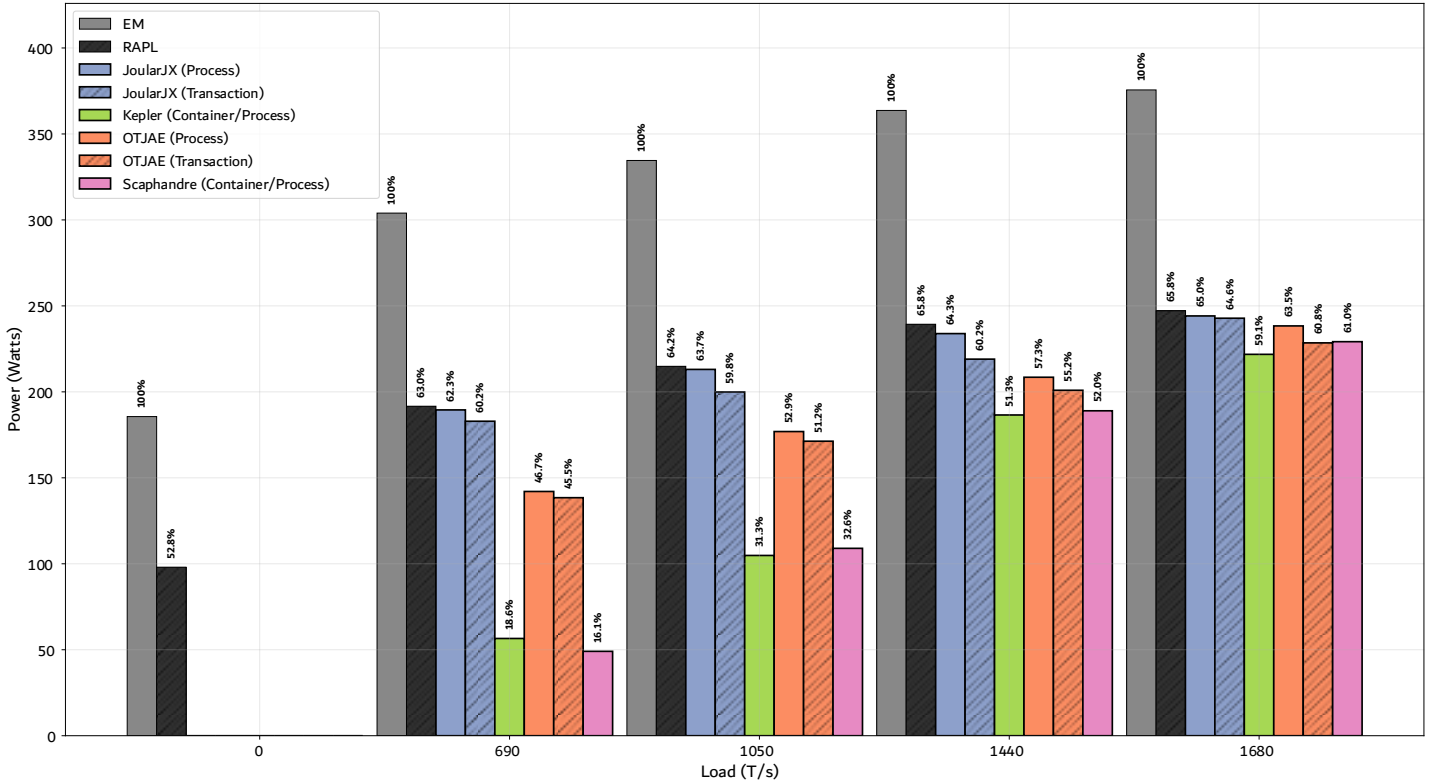
Experimental Results – Process Level



Experimental Results – Transaction Level

Load	Transaction	OTJAE		JoularJX	
		P_T	P_Δ	P_T	P_Δ
230T/s	*GET*	19.31W	-3.62W	23.64W	-6.59W
230T/s	*POST*	35.67W		45.73W	
230T/s	*DELETE*	83.46W		113.54W	
350T/s	*GET*	23.47W	-5.61W	24.45W	-13.13W
350T/s	*POST*	44.48W		50.05W	
350T/s	*DELETE*	103.35W		125.42W	
480T/s	*GET*	28.11W	-7.58W	29.99W	-14.85W
480T/s	*POST*	52.29W		56.04W	
480T/s	*DELETE*	120.52W		133.00W	
560T/s	*GET*	32.69W	-9.87W	49.07W	-1.33W
560T/s	*POST*	59.29W		66.48W	
560T/s	DELETE	136.50W		127.28W	

Experimental Results

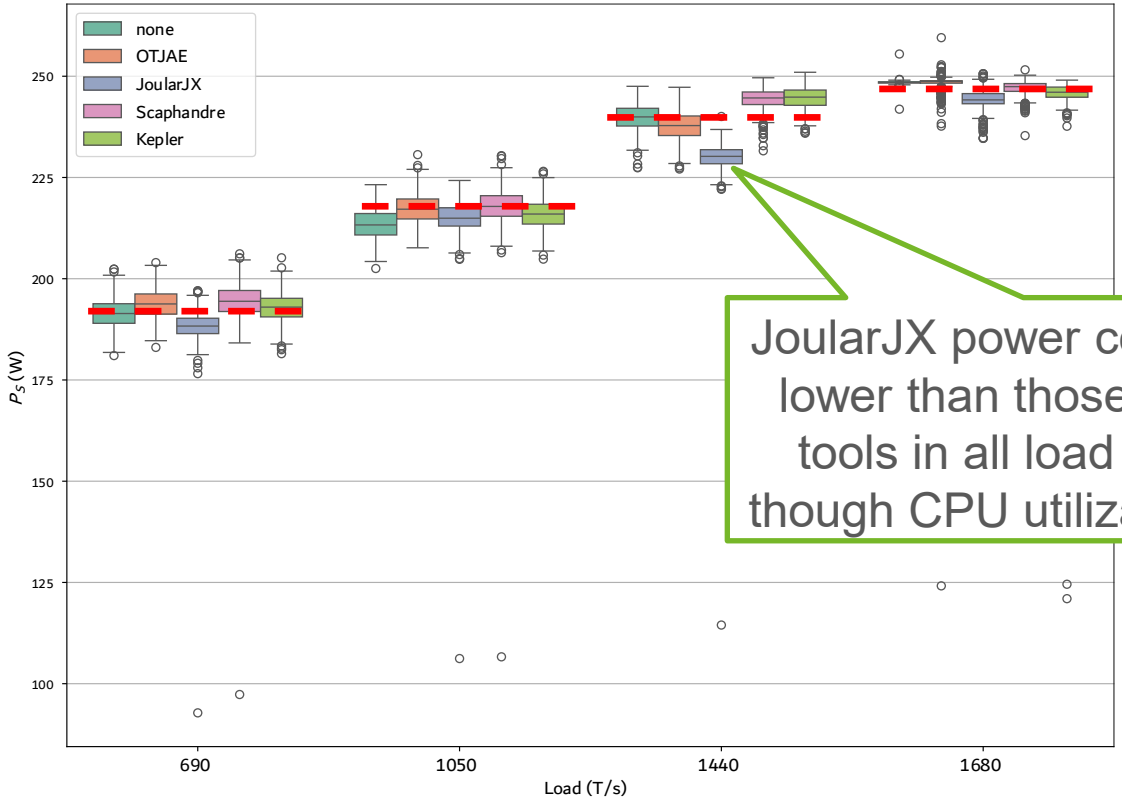


Tool Overhead

Load	CPU_{none}	CPU_{Kepler}	$CPU_{Scaphandre}$	CPU_{OTJAE}	$CPU_{JoularJX}$
0T/s	0.01%	0.08%	0.09%	0.02%	0.10%
690T/s	28.52%	29.35%	29.66%	29.23%	30.57%
1050T/s	48.58%	48.74%	51.61%	52.64%	54.31%
1440T/s	74.67%	77.10%	78.84%	73.76%	85.05%
1680T/s	94.30%	89.72%	95.89%	93.78%	93.44%

Likely related to network overhead of metric capture

Tool Overhead



JoularJX power consumption is lower than those of the other tools in all load levels even though CPU utilization is higher

Summary and Conclusions

- **Accuracy of tools:**
  - Container/process level: 16–65%
  - Transaction level: 45–64%
  - Kepler and Scaphandre show a significant underestimation at low CPU utilization
    - → as this is very common scenario in data centers you must account for this

- **Limitations of RAPL-based tools:**
  - Capture only CPU & DRAM power
  - Miss network, storage → measure only 53–66% of total system power.

- **Model-based approaches:**
  - OTJAE shows accuracy comparable to RAPL-based tools
  - Promising for cloud environments without hardware access.
  - → this enables the possibility to more easily represent additional resources

References

- **Tools:**
  - Powercap: https://www.kernel.org/doc/html/next/power/powercap/powercap.html
  - Kepler: https://github.com/sustainable-computing-io/kepler
  - Scaphandre: https://github.com/hubblo-org/scaphandre
  - JoularJX: https://github.com/joular/joularjx
  - OTJAE: https://github.com/RETIT/opentelemetry-javaagent-extension
- **Replication Package and Data:**
  - https://github.com/hm-green-it-lab/jss2025
- **Further reading:**
  - Software Energy Consumption: https://luiscruz.github.io/2023/05/13/energy-units.html
  - Kepler paper (note: architecture change since v0.10.0): https://ieeexplore.ieee.org/document/10254956
  - Kepler since v0.10.0: https://github.com/sustainable-computing-io/kepler/blob/v0.11.2/docs/developer/power-attribution-guide.md
  - JoularJX paper: https://ieeexplore.ieee.org/document/9826760
  - OTJAE paper: https://dl.gi.de/items/3cbc03f9-64b5-41a8-be00-d45cea2412cb
  - OTJAE / JoularJX evaluation paper: https://dl.acm.org/doi/10.1145/3696630.3728709

Thanks a lot for your attention.

RETIT

Questions?

Andreas Brunnert
brunnert@retit.de

Manuel Steinberg
manuel.steinberg@hm.edu