RIMEDO LABS | i14y LAB

O-RAN End-to-End Energy Efficiency Testing Framework

Marcin Hoffmann, Marcin Dryjański, Adrian Kliks

All things wireless ●

Motivation

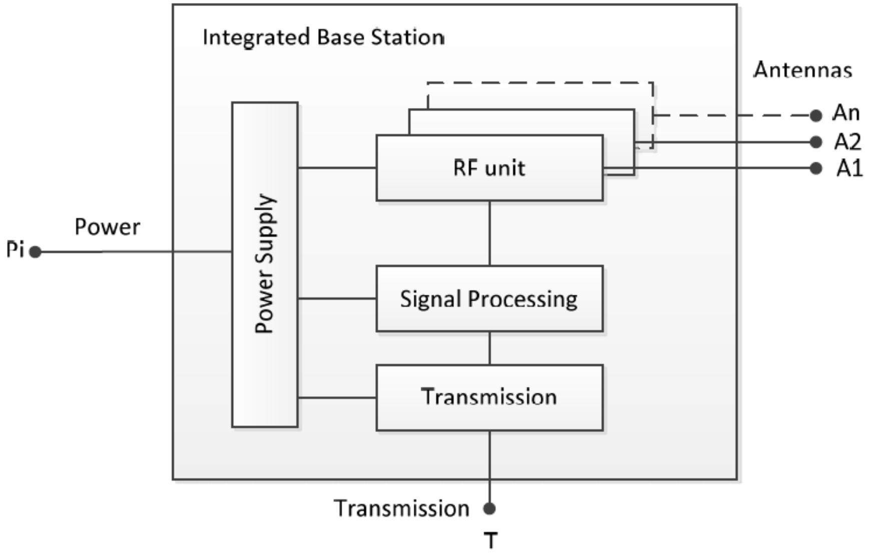
Challenges in O-RAN E2E Energy Efficiency Testing

Multiple hardware and software vendors offer energy-saving solutions that are hard to compare – a lack of unified test definitions and scenarios

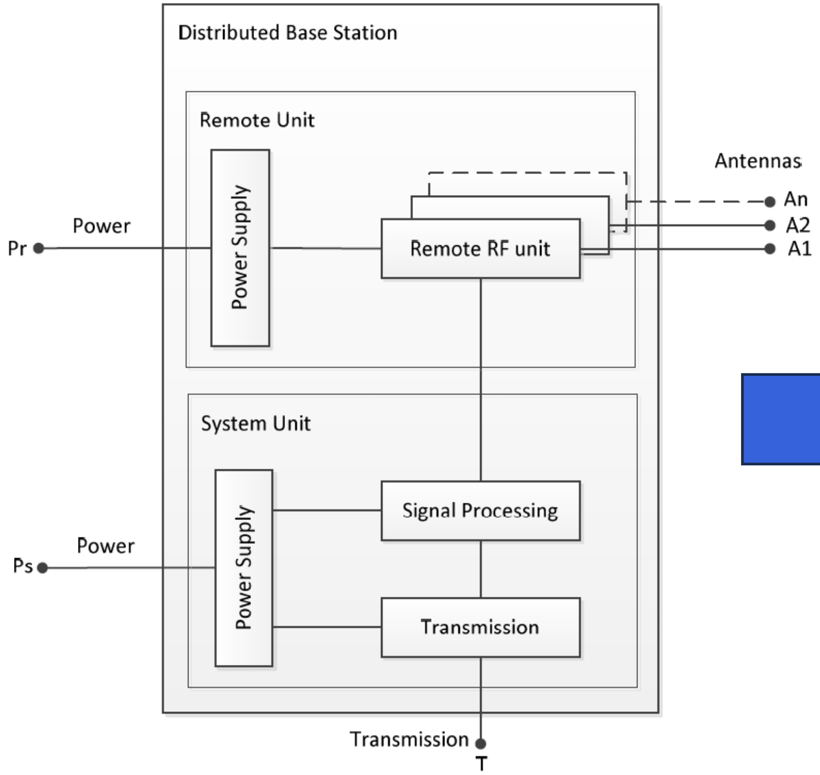
Various organizations put effort into the energy efficiency testing, e.g., ETSI, 3GPP, O-RAN ALLIANCE, NGNM, TIP – yet their work cover different aspects of E2E EE Testing

There is a demand to combine the current standardization efforts into the unified O-RAN E2E EE Testing Framework that would provide common procedures to evaluate energy efficiency of network components, like O-CU/O-DU/O-RU, and xApps/rApps

ETSI – Configuration



Integrated BS model (example)
[ETSI_2027061_TS]



Distributed BS model (example)
[ETSI_2027061_TS]

Build a definition of the O-RAN „component level EE testing" on the top of the ETSI documents for static and dynamic load

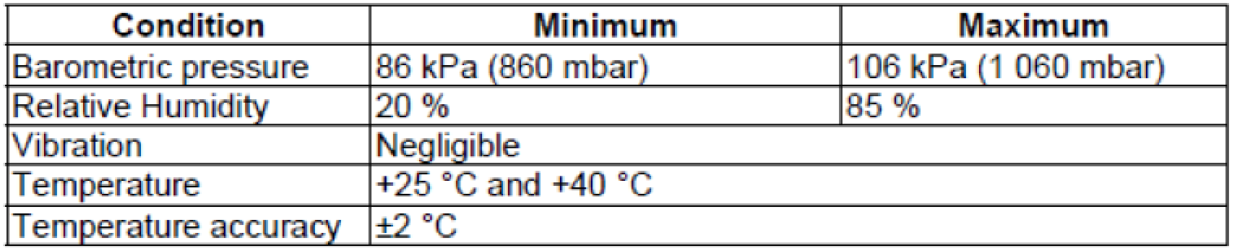
ETSI – Static/Dynamic Load Measurements

Utilize, and extend existing ETSI procedures

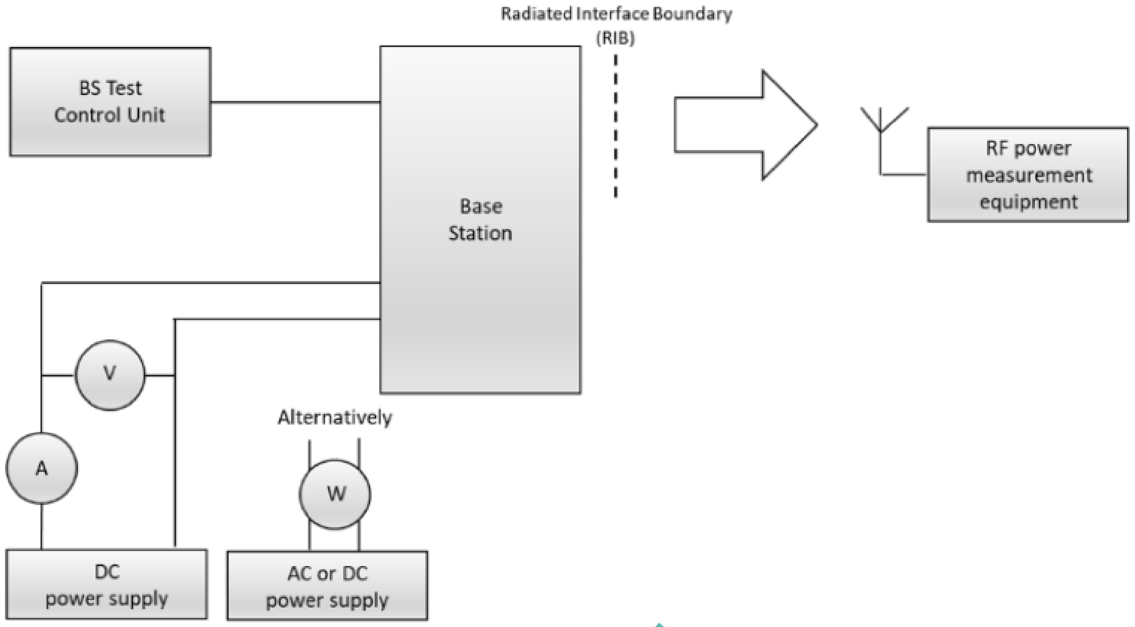
- Power supply requirements
- Environmental conditions
- Static Load measurements
- Dynamic Load measurements
- Measurement reports

Combine the ETSI-defined measurements and procedures with 3GPP, and O-RAN ALLIANCE to create a „component-level EE testing framework”

Table 1 BS environmental conditions [ETSI_103786_TS]



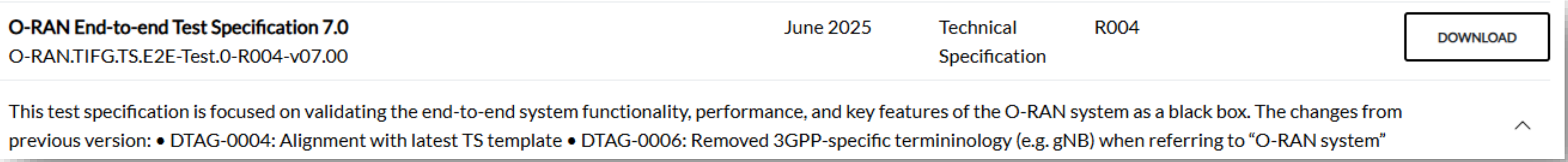
Condition	Minimum	Maximum
Barometric pressure	86 kPa (860 mbar)	106 kPa (1 060 mbar)
Relative Humidity	20 %	85 %
Vibration	Negligible	
Temperature	+25 °C and +40 °C	
Temperature accuracy	±2 °C	



O-RAN Test and Measurement Aspects for NES

TIFG and SuFG

(Energy Measurement Analysis Report)



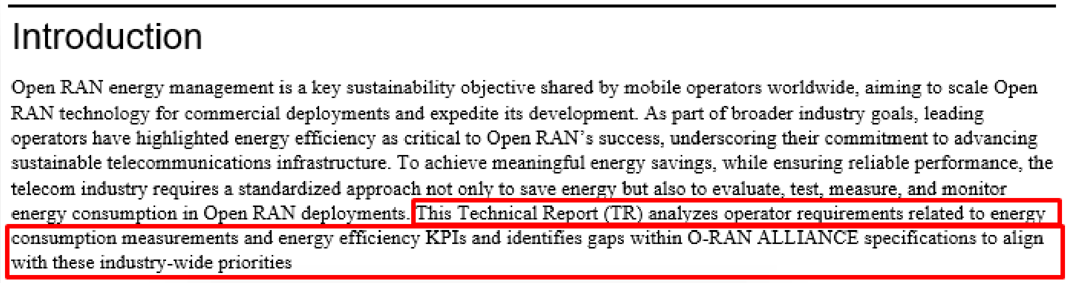
O-RAN End-to-end Test Specification 7.0
O-RAN.TIFG.TS.E2E-Test.0-R004-v07.00

June 2025 | Technical Specification | R004 | DOWNLOAD

This test specification is focused on validating the end-to-end system functionality, performance, and key features of the O-RAN system as a black box. The changes from previous version: • DTAG-0004: Alignment with latest TS template • DTAG-0006: Removed 3GPP-specific termininology (e.g. gNB) when referring to "O-RAN system"

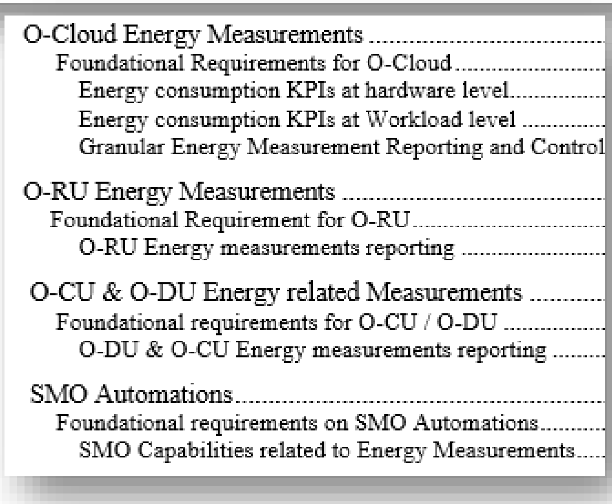
O-RAN.SuFG.TR.NES-Analysis-R004-v01.01

Technical Report

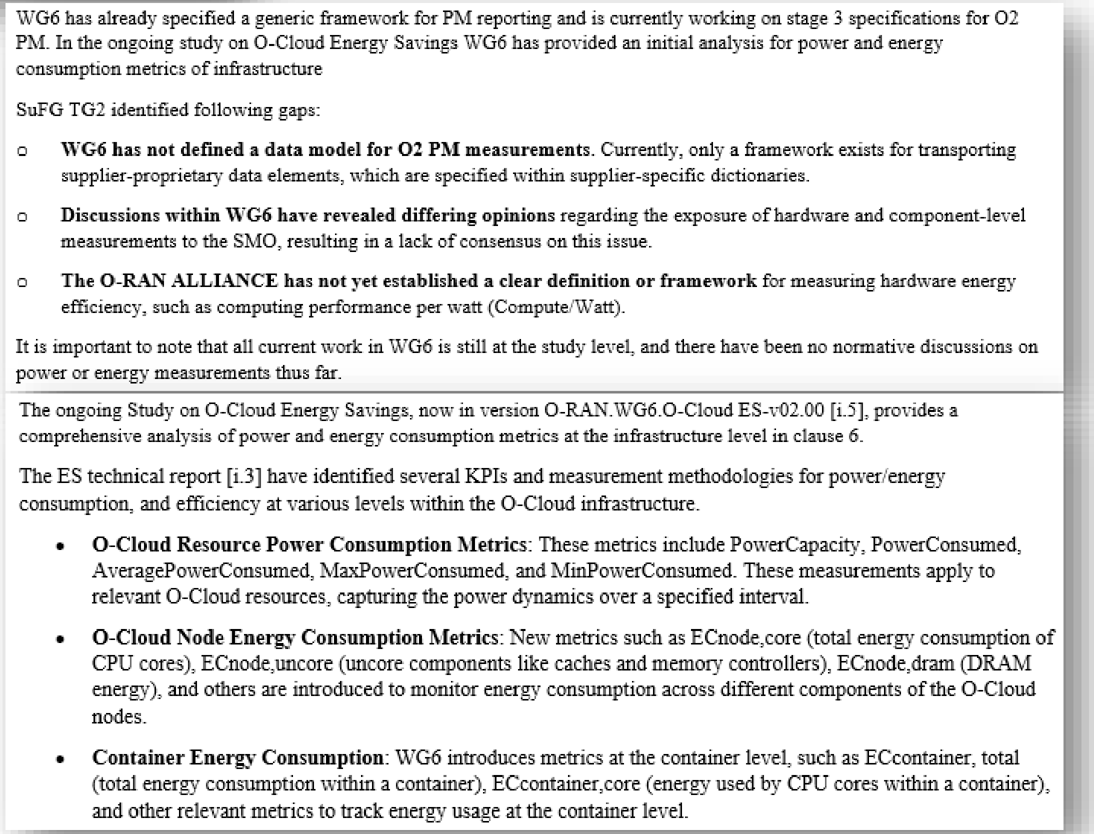


Introduction

Open RAN energy management is a key sustainability objective shared by mobile operators worldwide, aiming to scale Open RAN technology for commercial deployments and expedite its development. As part of broader industry goals, leading operators have highlighted energy efficiency as critical to Open RAN's success, underscoring their commitment to advancing sustainable telecommunications infrastructure. To achieve meaningful energy savings, while ensuring reliable performance, the telecom industry requires a standardized approach not only to save energy but also to evaluate, test, measure, and monitor energy consumption in Open RAN deployments. This Technical Report (TR) analyzes operator requirements related to energy consumption measurements and energy efficiency KPIs and identifies gaps within O-RAN ALLIANCE specifications to align with these industry-wide priorities



Gap analysis for Energy measurements



WG6 has already specified a generic framework for PM reporting and is currently working on stage 3 specifications for O2 PM. In the ongoing study on O-Cloud Energy Savings WG6 has provided an initial analysis for power and energy consumption metrics of infrastructure

SuFG TG2 identified following gaps:

- **WG6 has not defined a data model for O2 PM measurements**. Currently, only a framework exists for transporting supplier-proprietary data elements, which are specified within supplier-specific dictionaries.
- **Discussions within WG6 have revealed differing opinions** regarding the exposure of hardware and component-level measurements to the SMO, resulting in a lack of consensus on this issue.
- **The O-RAN ALLIANCE has not yet established a clear definition or framework** for measuring hardware energy efficiency, such as computing performance per watt (Compute/Watt).

It is important to note that all current work in WG6 is still at the study level, and there have been no normative discussions on power or energy measurements thus far.

The ongoing Study on O-Cloud Energy Savings, now in version O-RAN.WG6.O-Cloud ES-v02.00 [i.5], provides a comprehensive analysis of power and energy consumption metrics at the infrastructure level in clause 6.

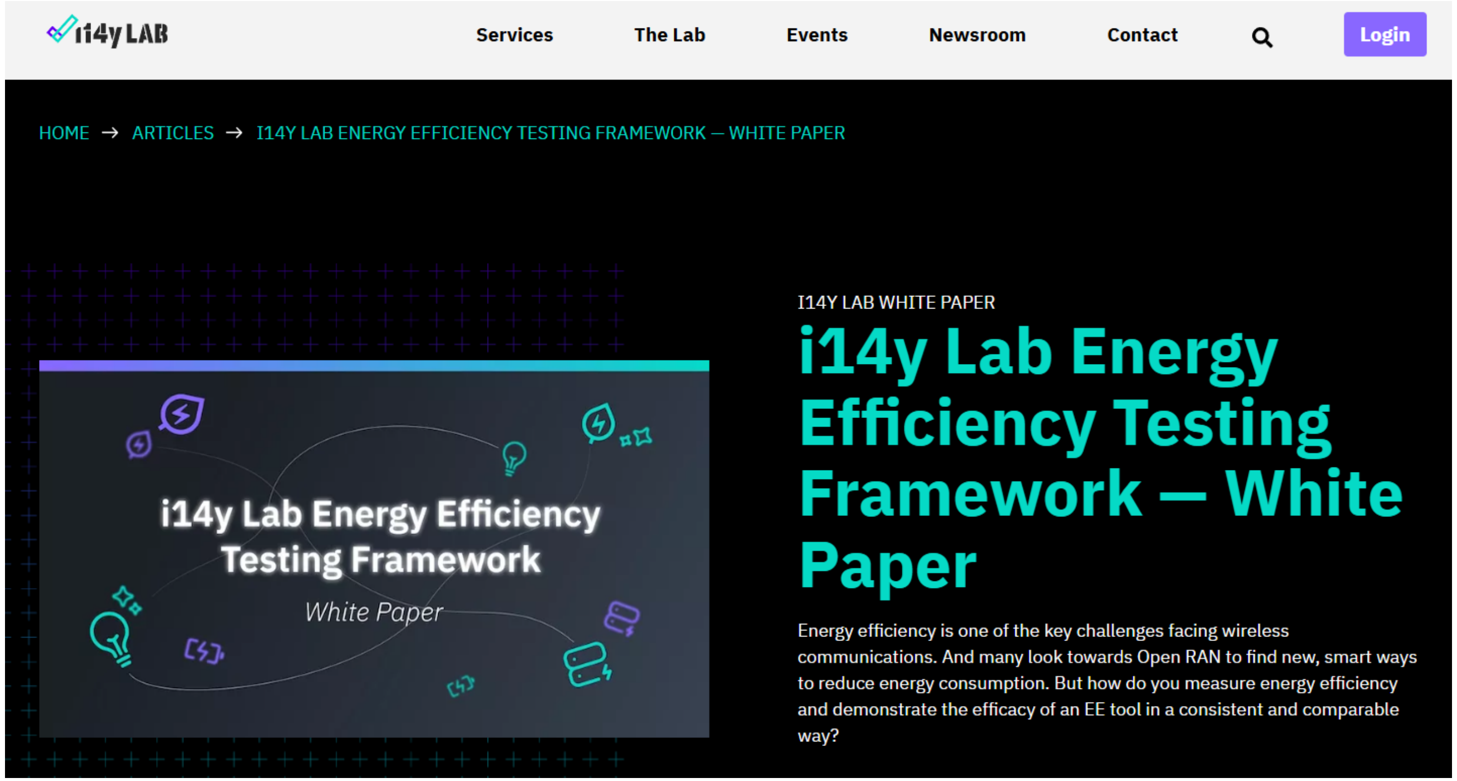
The ES technical report [i.3] have identified several KPIs and measurement methodologies for power/energy consumption, and efficiency at various levels within the O-Cloud infrastructure.

- **O-Cloud Resource Power Consumption Metrics**: These metrics include PowerCapacity, PowerConsumed, AveragePowerConsumed, MaxPowerConsumed, and MinPowerConsumed. These measurements apply to relevant O-Cloud resources, capturing the power dynamics over a specified interval.
- **O-Cloud Node Energy Consumption Metrics**: New metrics such as ECnode,core (total energy consumption of CPU cores), ECnode,uncore (uncore components like caches and memory controllers), ECnode,dram (DRAM energy), and others are introduced to monitor energy consumption across different components of the O-Cloud nodes.
- **Container Energy Consumption**: WG6 introduces metrics at the container level, such as ECcontainer, total (total energy consumption within a container), ECcontainer,core (energy used by CPU cores within a container), and other relevant metrics to track energy usage at the container level.

Summary of Standarization Efforts

Documents	Scope	What is missing?
ETSI	• defines metrics and measurements for EC/EE of a base station • provides models for BSs and requirements for measurement equipment • describes standardized test setups and procedures for power measurements (static and dynamic) for a single base station • standardized reports from the measurements • good starting point for O-RAN extension	• lack O-RAN context and architecture, e.g. 7.2 split • lack of scenarios/configurations for large-scale network power measurements • missing procedures for testing mobile networks with VNF, e.g., virtual CU/DU • QoS in EE definitions
3GPP	• frequently refers to ETSI regarding Energy Efficiency/power consumption • defines reference scenarios, and models for system-level simulations • demand on measurement of VNFs' EE/PC • EE evaluation based on multiple scenarios: MBB, URLLC, mMTC/Rural, urban	• lack of O-RAN context • lack of procedures for E2E EE testing • EE/Power consumption KPIs are not available per HW unit or VNF • most part of the material referring to EE/EC is from TRs, not TS, i.e., not normative
NGMN	• Cloud vs Physical deployment for a mobile network • presents aspects relevant to cloud-type/environment measurements • metering servers, storage, and network components • solutions for VMN/CNF power measurement • optimization of RAN deployment • points gaps/missing elements between the O-RAN and cloud aspects	• missing O-RAN aspects • gathers different aspects, but does not fill the gap between the cloud itself and the network • not a specification
O-RAN	• related directly to O-RAN aspects • defines NES methods, metrics, and requirements associated with the O-RU (COOS and RCR) (quite mature) • concepts, requirements, and use cases optimizing for O-Cloud resources (ORES) along with metrics (early stage) • test procedure, test equipment definition, test setup and configuration, test criteria for E2E testing for O-RAN ES (COOS)	• missing details and tech spec for O-Cloud aspects (early stage on this) • different maturity stages for different use cases regarding ES and E2E testing • different documents for different parts of the system (treating separately O-Cloud from COOS, RCR, ASM) • test specs only for COOS rApp (early stage) • several options for measurements and KPIs (not all are implemented in practice) • missing large-scale testing scenarios for xApp/rApp evaluation
TIP	• a use case, being a framework for a method of reducing EC • requirements, configuration, measurements, goal, interface requirements,	• not related to testing • only for a single use case and very specific • only generic parameters

White paper



HOME → ARTICLES → I14Y LAB ENERGY EFFICIENCY TESTING FRAMEWORK — WHITE PAPER

I14Y LAB WHITE PAPER

i14y Lab Energy Efficiency Testing Framework — White Paper

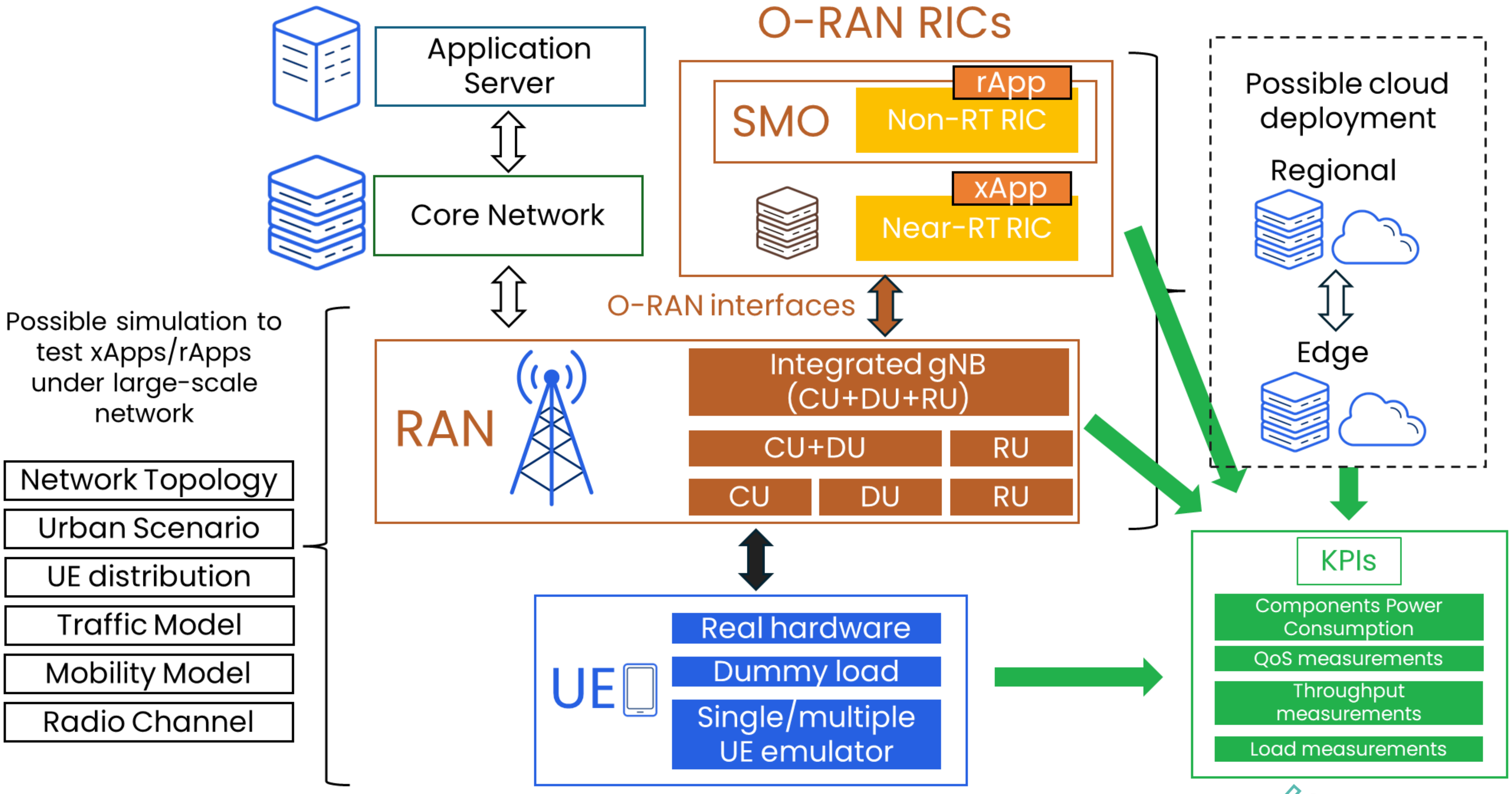
Energy efficiency is one of the key challenges facing wireless communications. And many look towards Open RAN to find new, smart ways to reduce energy consumption. But how do you measure energy efficiency and demonstrate the efficacy of an EE tool in a consistent and comparable way?

White paper QR code

https://www.i14y-lab.com/article/i14y-lab-energy-efficiency-testing-framework-white-paper

Overall Framework Considerations

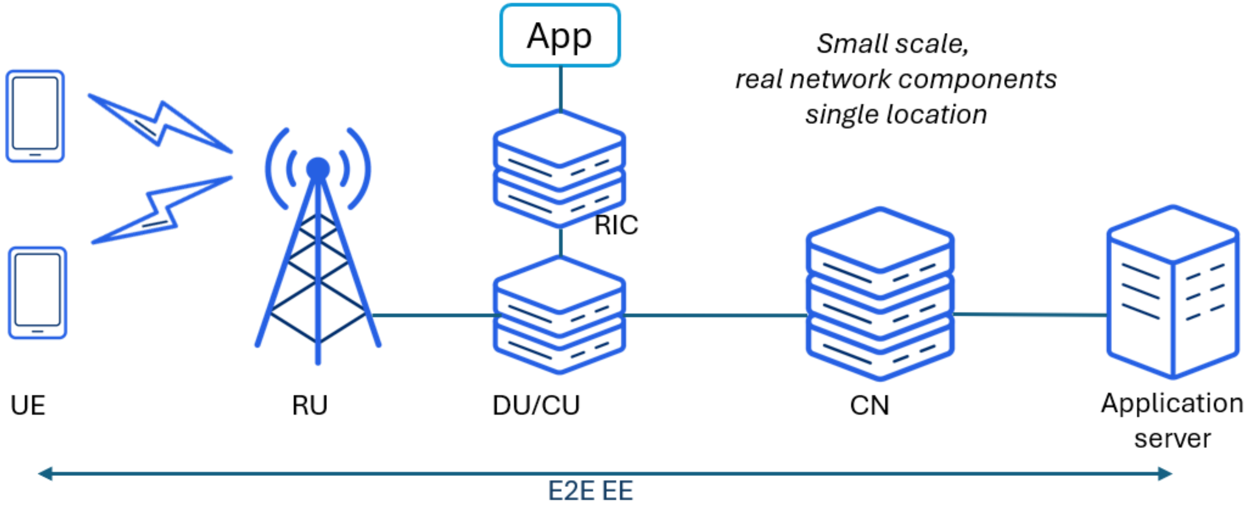
O-RAN E2E Energy Efficiency Testing **Framework**



i14y LAB rimedolabs.com

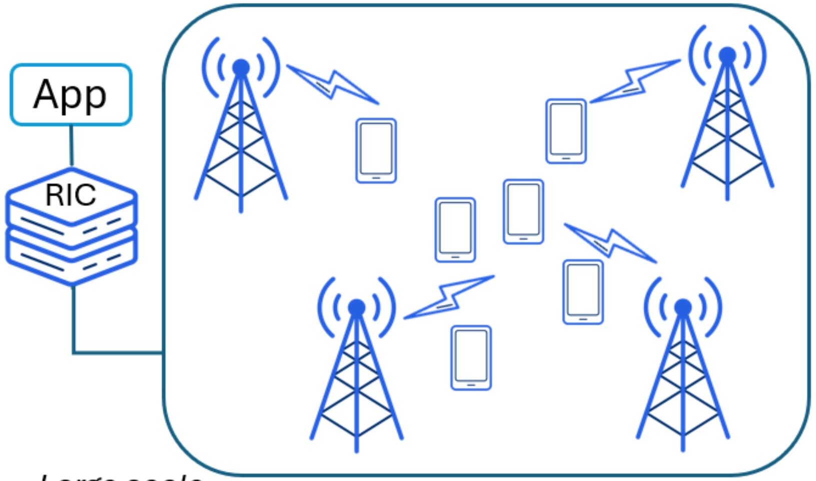
Different Views on E2E Energy Efficiency Testing

Component-level measurements for EE



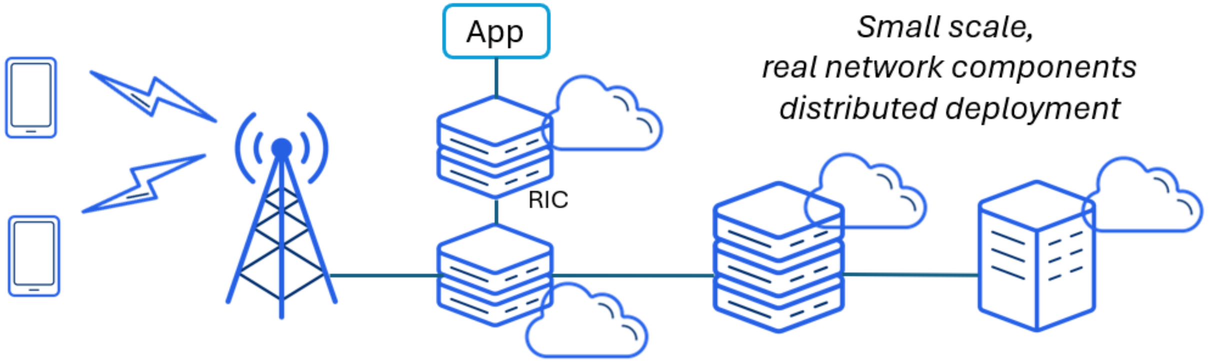
Small scale,
real network components
single location

Global/feature based testing for EE



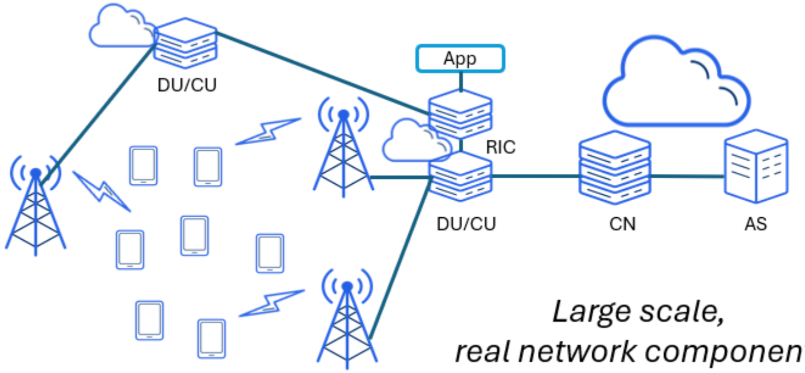
Large scale
Simulated RAN

Deployment view for EE testing



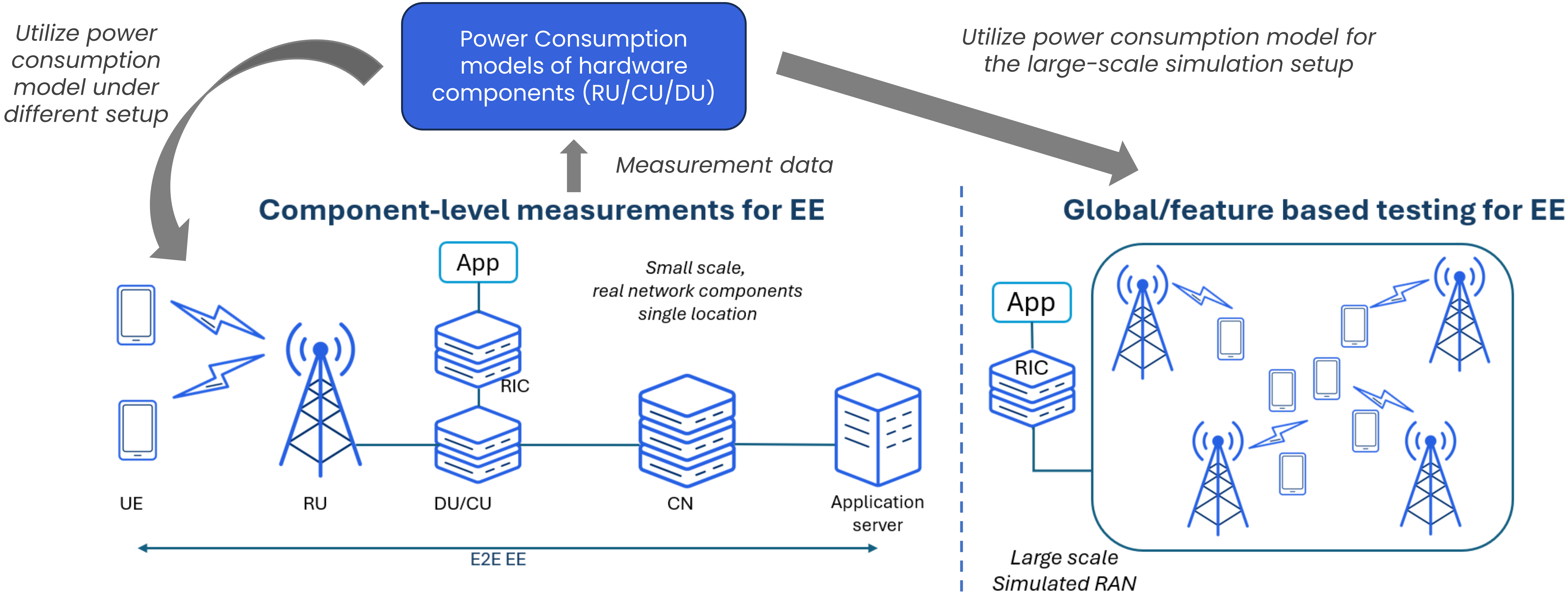
Small scale,
real network components
distributed deployment

Multi-link/wide network EE testing



Large scale,
real network components

Information Exchange and Data Modeling



Power Consumption models of hardware components (RU/CU/DU)

Utilize power consumption model under different setup

Measurement data

Utilize power consumption model for the large-scale simulation setup

Component-level measurements for EE

Small scale, real network components single location

Global/feature based testing for EE

Large scale Simulated RAN

Test of a network E2E full-stack, with a single gNB:

- Measure power consumption of various hardware components, e.g., CU/DU/RU/RIC
- Utilize either real, emulated UE or dummy load
- Produce power consumption models for simulations
- Utilize previously measured data to create power consumption models of different vendors

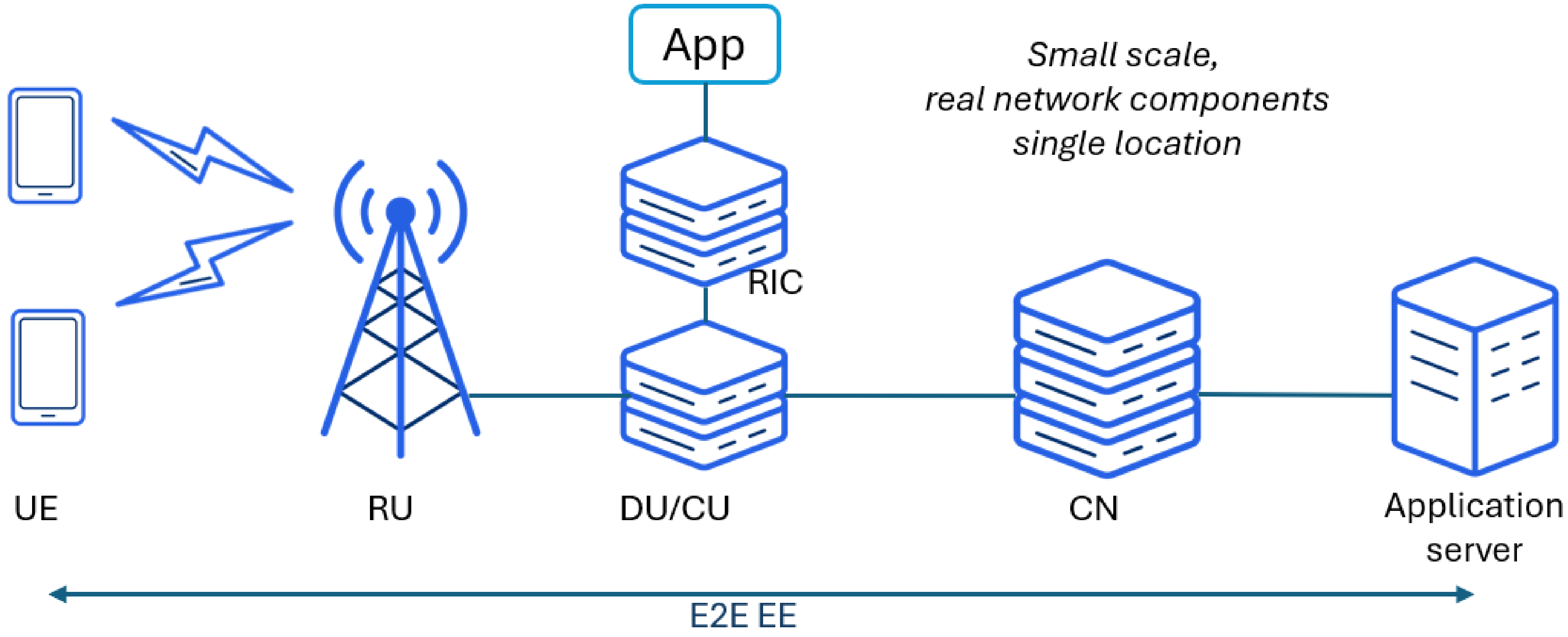
Simulation of a large scale network:

- Test EE/ES provided by xApps/rApps
- Test xApps/rApps signalling overhead
- Test different RICs, e.g., their computational capability, delay between RAN and xApps/rApps

Component Level Measurement of EE

The Goal of the Framework

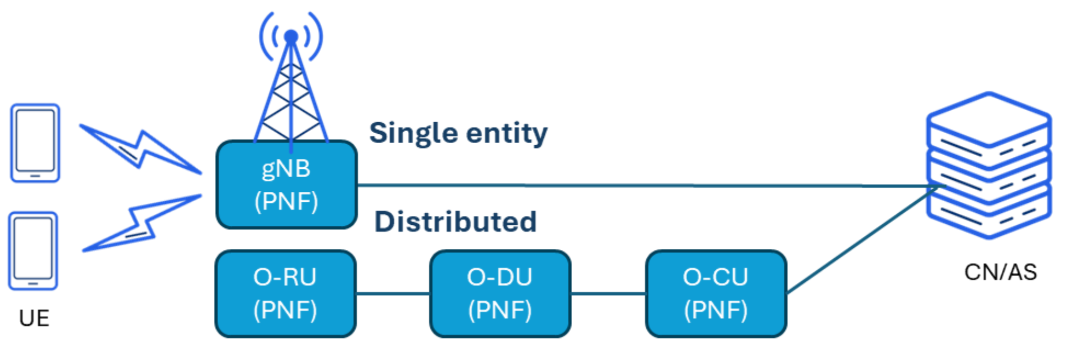
Component-level measurements for EE



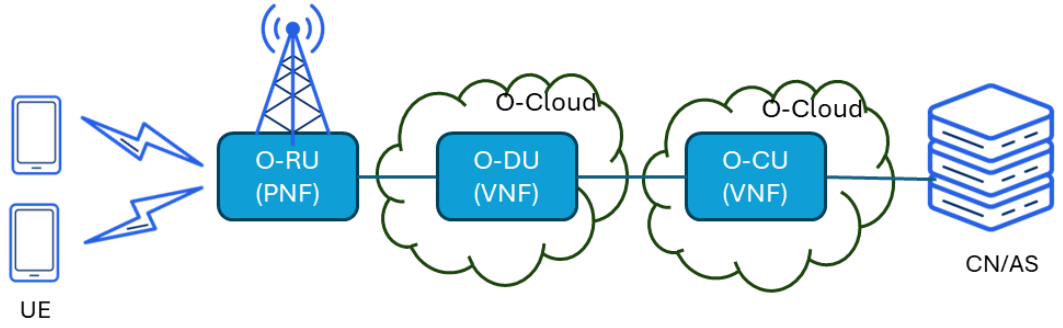
Small scale,
real network components
single location

Deployment Options

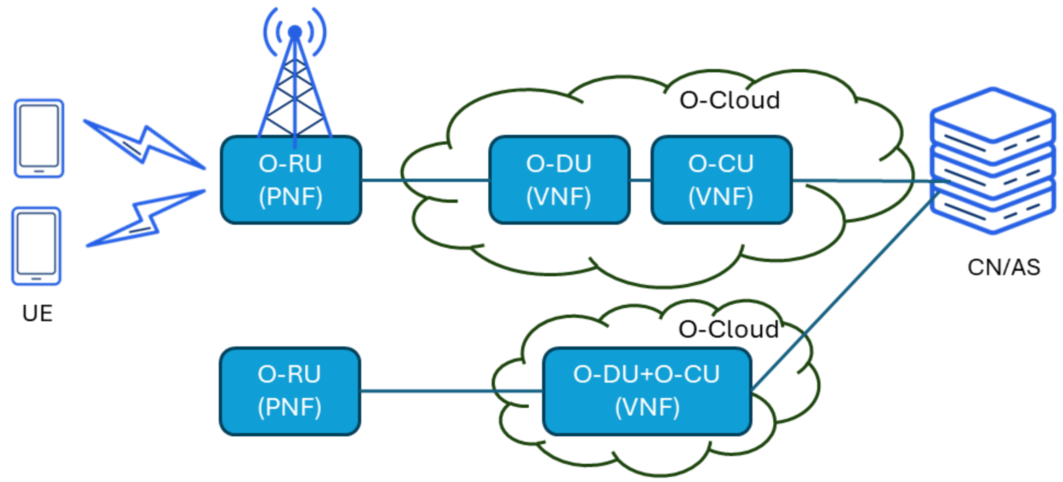
HW-based gNB



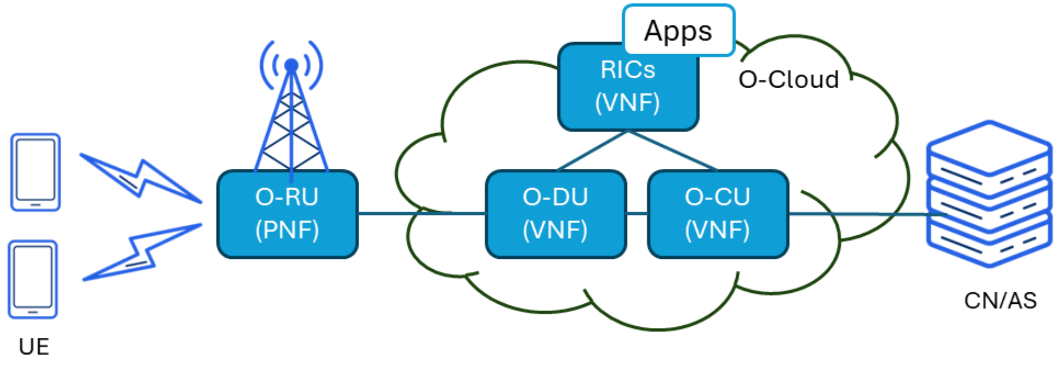
Virtualized and distributed O-CU/O-DU



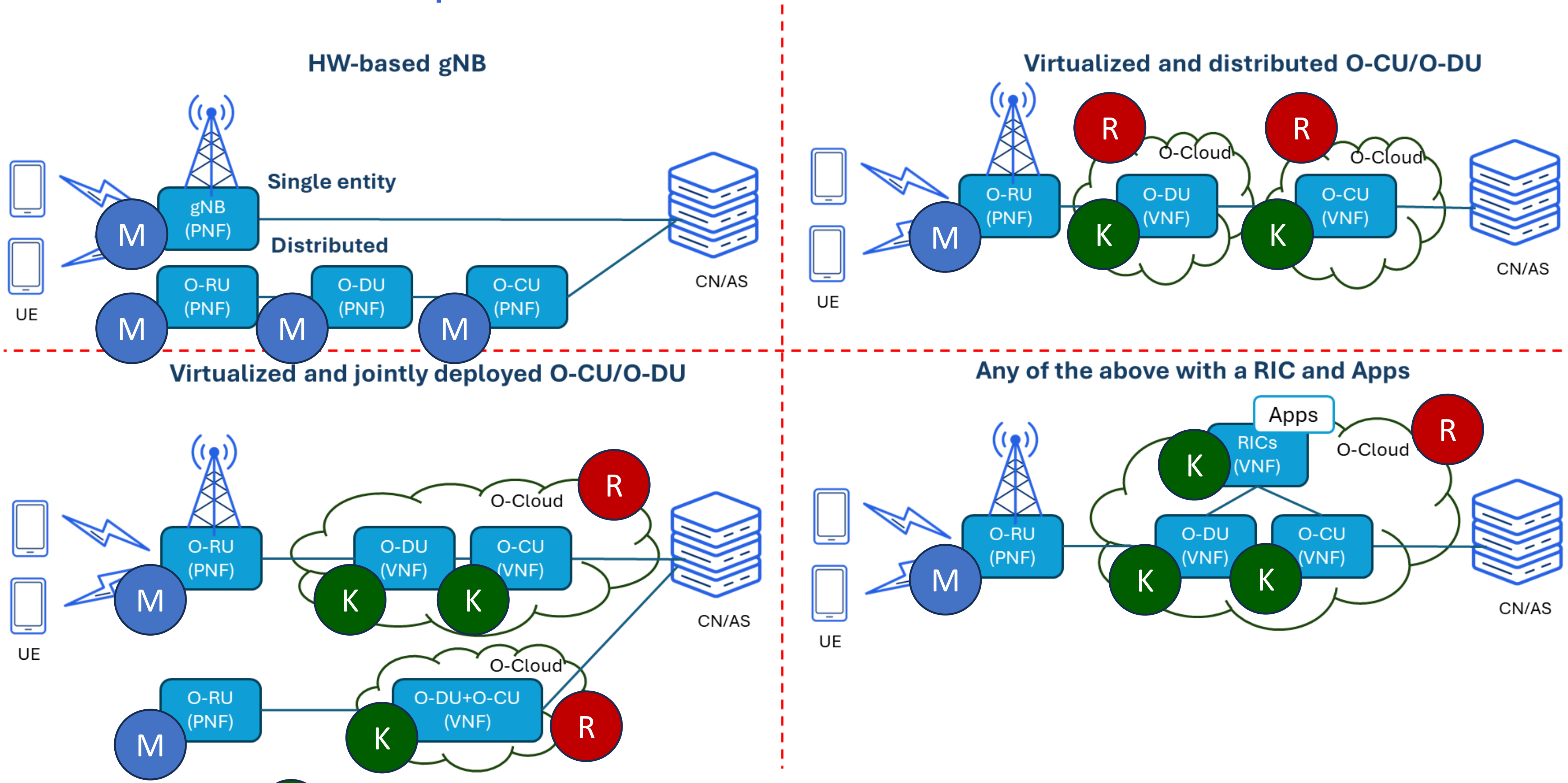
Virtualized and jointly deployed O-CU/O-DU



Any of the above with a RIC and Apps



Power Consumption Measurement Points



HW-based gNB

Virtualized and distributed O-CU/O-DU

Virtualized and jointly deployed O-CU/O-DU

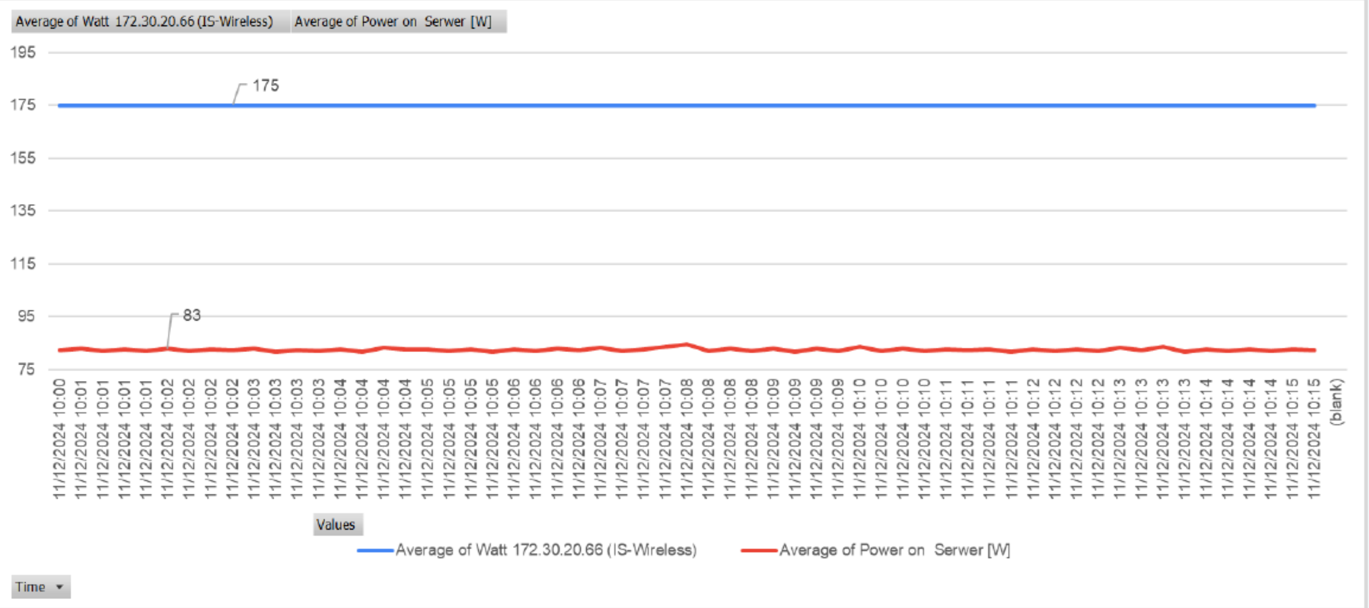
Any of the above with a RIC and Apps

M Hardware meter

K Kepler

R Redfish

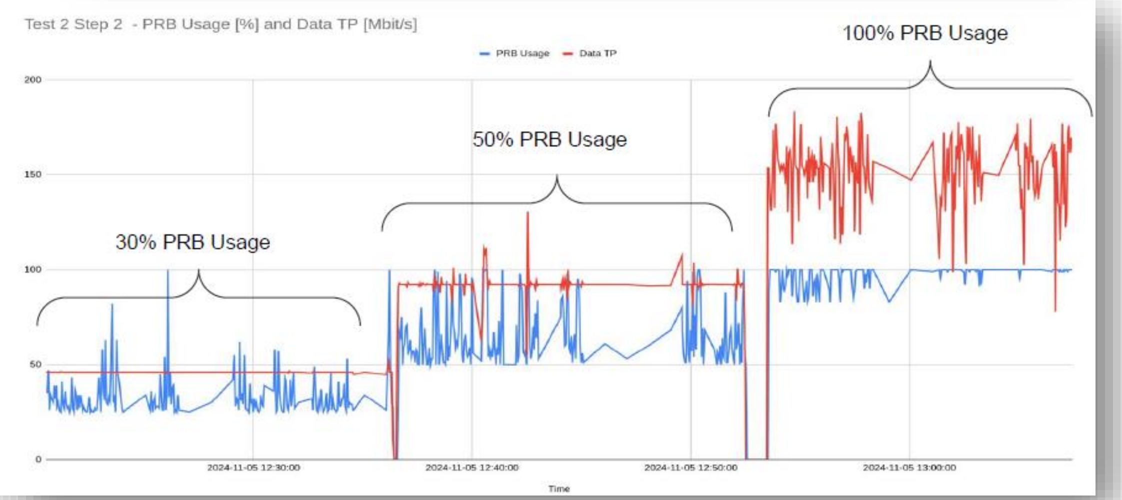
The Gap Between Kepler and Redfish



Redfish measure power Of all hardware layer while Kepler only able to measure power Of selected Pods.
As offset, we could see around 83 watt used for another Pods while CU+DU was not running.
Also, we observed that around 92 watt difference is between Redfish and Kepler.

O-RAN Global PlugFest Fall 2024

Energey Consumption Measurement in
IS-Wireless & Benetel E2E Setup
Date: 2024-10-30

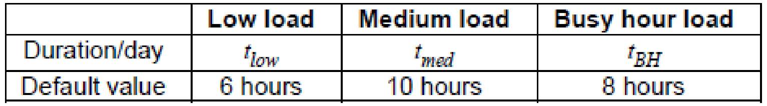


Results of the tests in i14ylab showed a gap between Kepler and Redfish.

i14y LAB rimedolabs.com

The Traffic Profiles

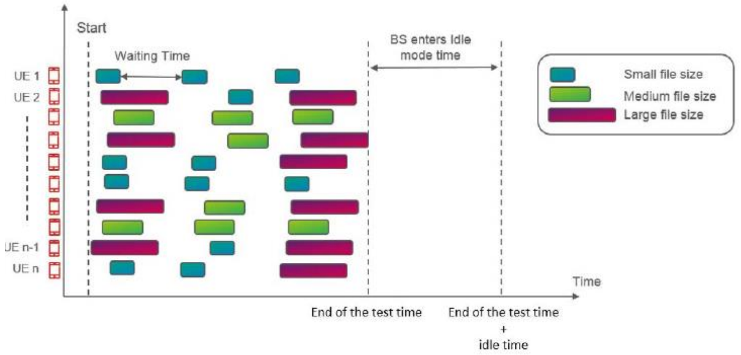
ETSI static load

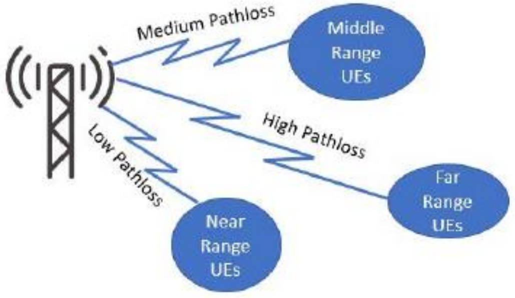


	Low load	Medium load	Busy hour load
Duration/day	t_{low}	t_{med}	t_{BH}
Default value	6 hours	10 hours	8 hours

- Basic measurement for 10%, 30%, 50% and full load
- To obtain better power consumption profile better granularity required
- E.g., from 0 to full load with step of 10%

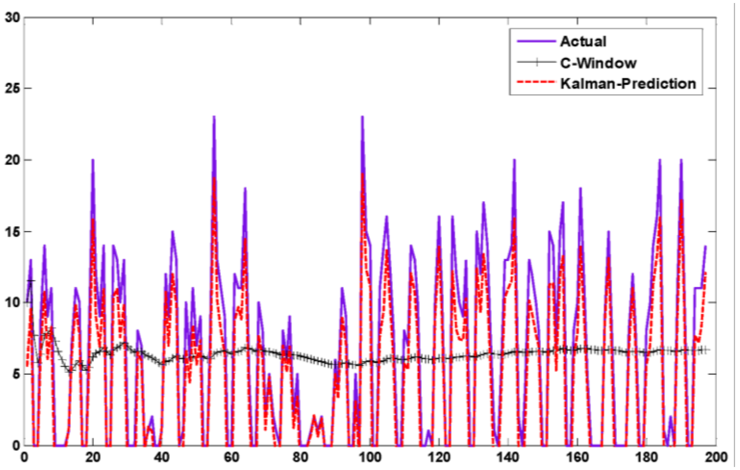
ETSI dynamic load





- More scenarios
- More advanced, requires UE emulators/real UE

Real-world data



- Requires operator data
- Post-processing is needed to establish per UE traffic
- Can provide a real-world evaluation scenarios under traffic variations
- UE emulators + postprocessed data

The Measurement Outcome

Fit the results of the measurement to the parameters of 3GPP TR 38.864 Power Consumption Model

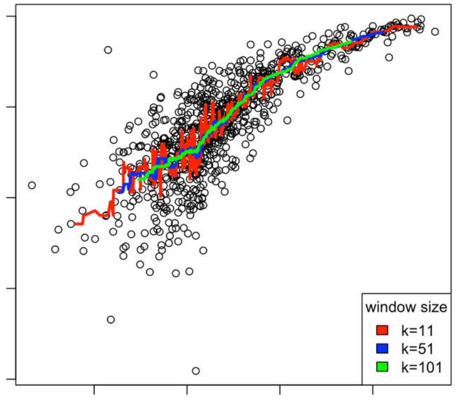
- The 3GPP PC model can scaled concerning the:
  - number of used physical antenna elements, or TX/RX RUs
  - occupied BW/RBs for DL and/or UL in a slot/symbol in one CC
  - number of CCs in CA
  - number of TRPs
  - PSD or transmit power
  - number of DL and/or UL symbols occupied within a slot.

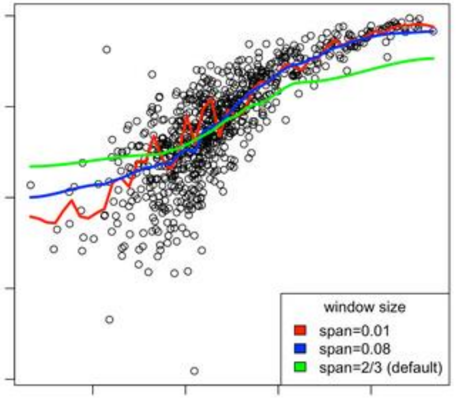
- The BS power consumption for both DL and UL consists of a static and dynamic term, e.g.,

$$P^{DL} = P^{DL}_{static} + P^{DL}_{dynamic}$$

Initial steps:

1. Perform measurement (either static or dynamic) under different:
   1. Number of antenna elements
   2. Occupied BW/RBs
   3. number of CCs in CA
   4. number of TRPs
   5. PSD or transmit power
   6. number of DL and/or UL symbols occupied within a slot
2. Collect power consumption and load statistics
3. Fit the model parameters for various components, O-RU, O-DU, O-CU, etc.



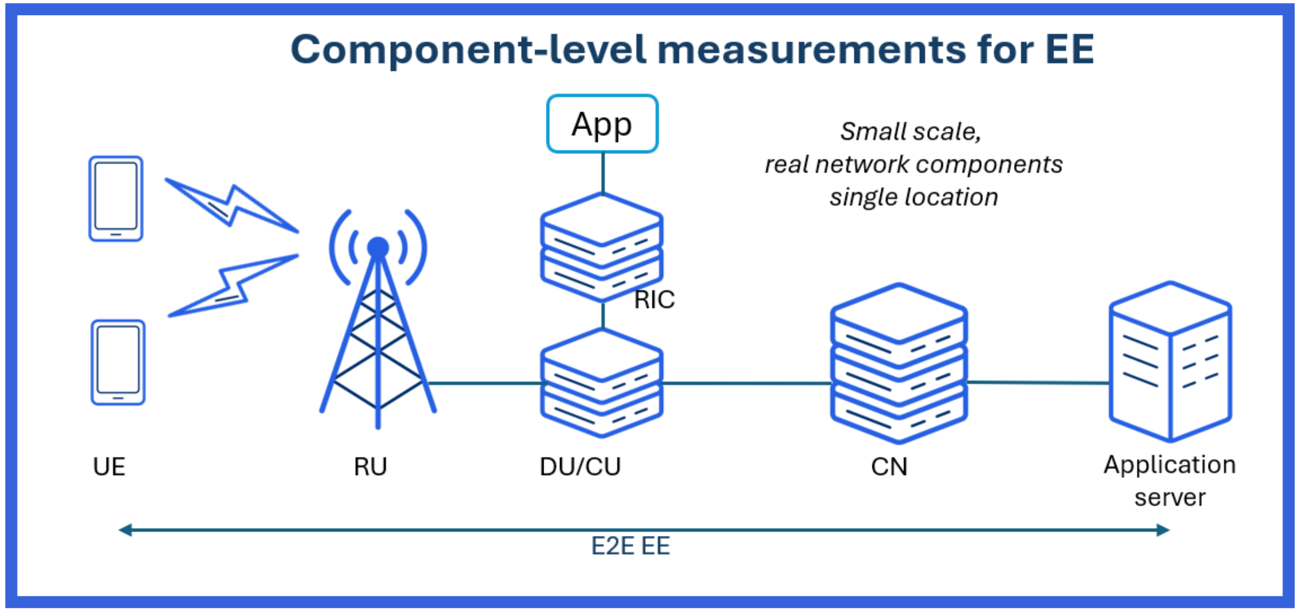


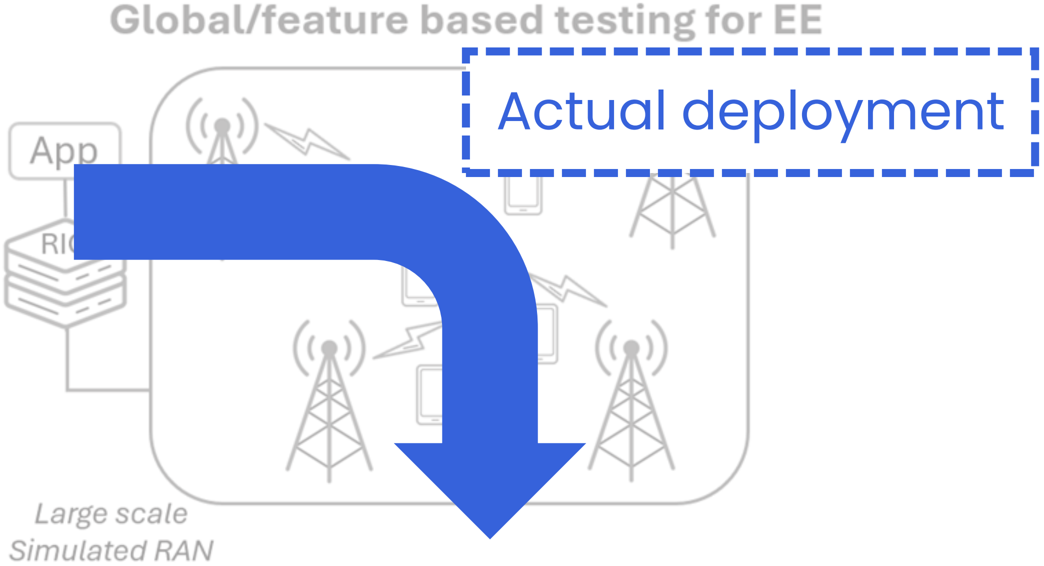
Additional challenges:

- How to define a set of QoS metrics?
- How does power consumption depend on the setup, e.g., is it different for DU while employing various RU/CU?
- How power consumption scales with UE number?

Measuring Power Consumption in a Virtualized Environment

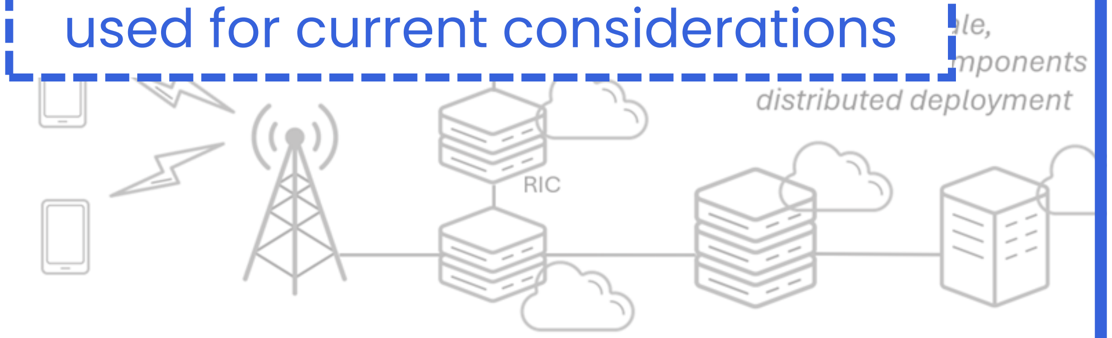
Deployment Option for Energy Measurements

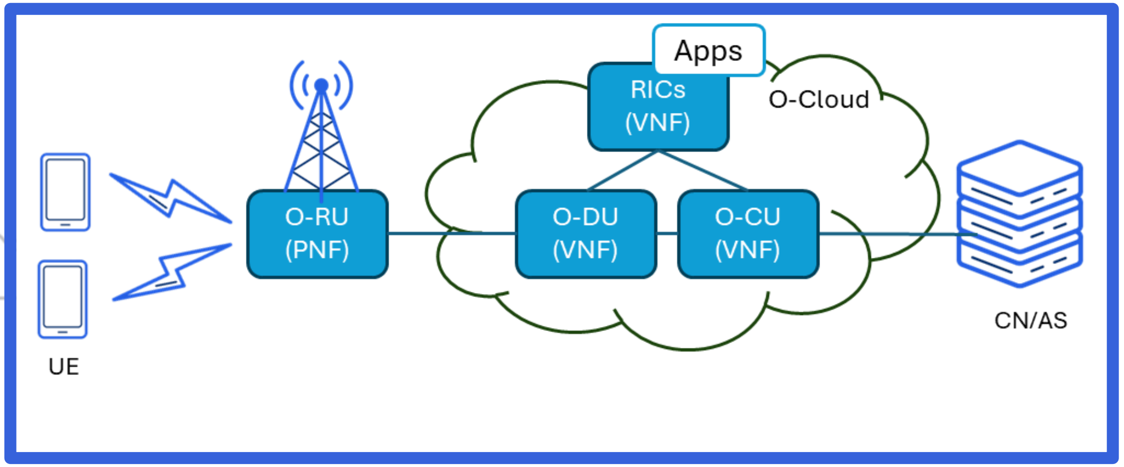




Actual deployment

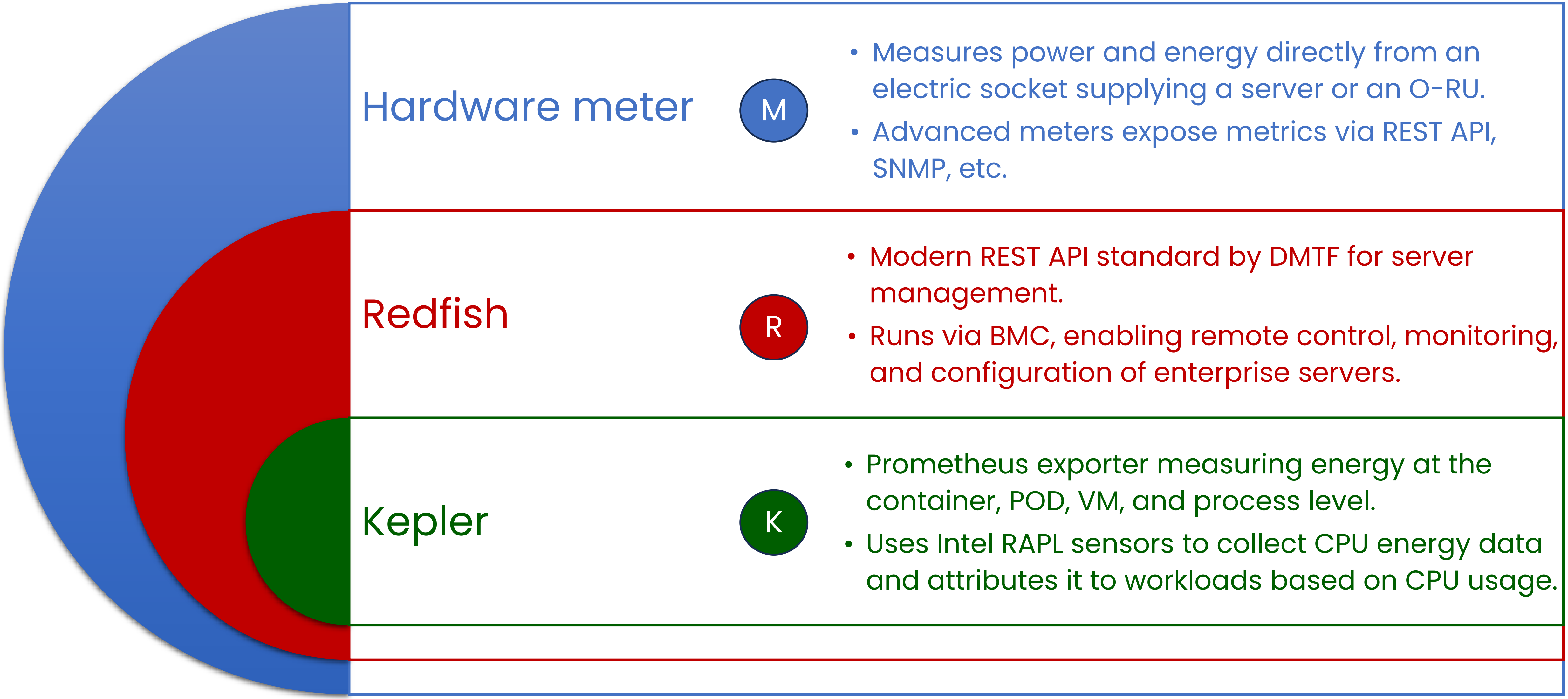
Energy testing framework option used for current considerations





i14y LAB rimedolabs.com

Power Consumption **Measurement Tools**



Hardware meter

M

- Measures power and energy directly from an electric socket supplying a server or an O-RU.
- Advanced meters expose metrics via REST API, SNMP, etc.

Redfish

R

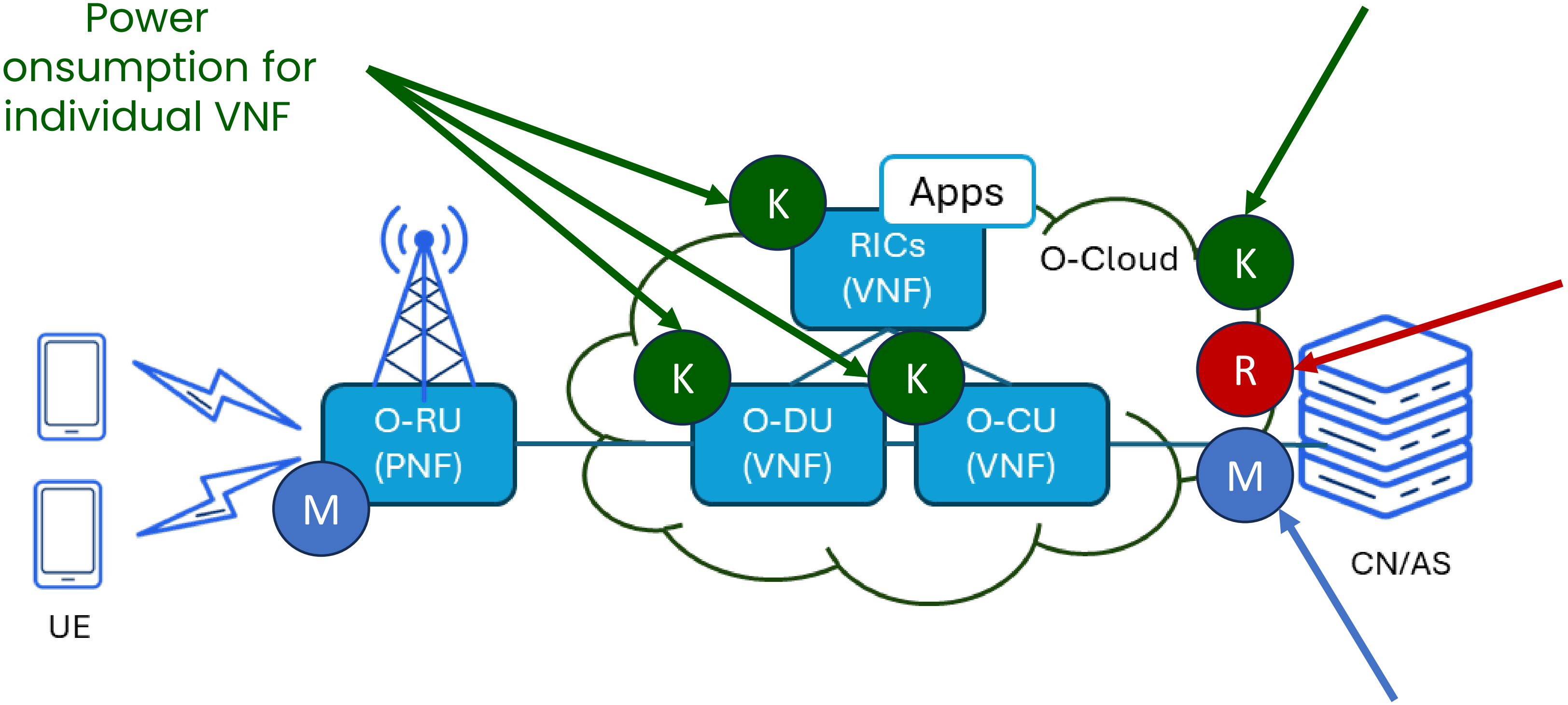
- Modern REST API standard by DMTF for server management.
- Runs via BMC, enabling remote control, monitoring, and configuration of enterprise servers.

Kepler

K

- Prometheus exporter measuring energy at the container, POD, VM, and process level.
- Uses Intel RAPL sensors to collect CPU energy data and attributes it to workloads based on CPU usage.

Measurement Points Definition for Power Consumption



Power consumption for individual VNF

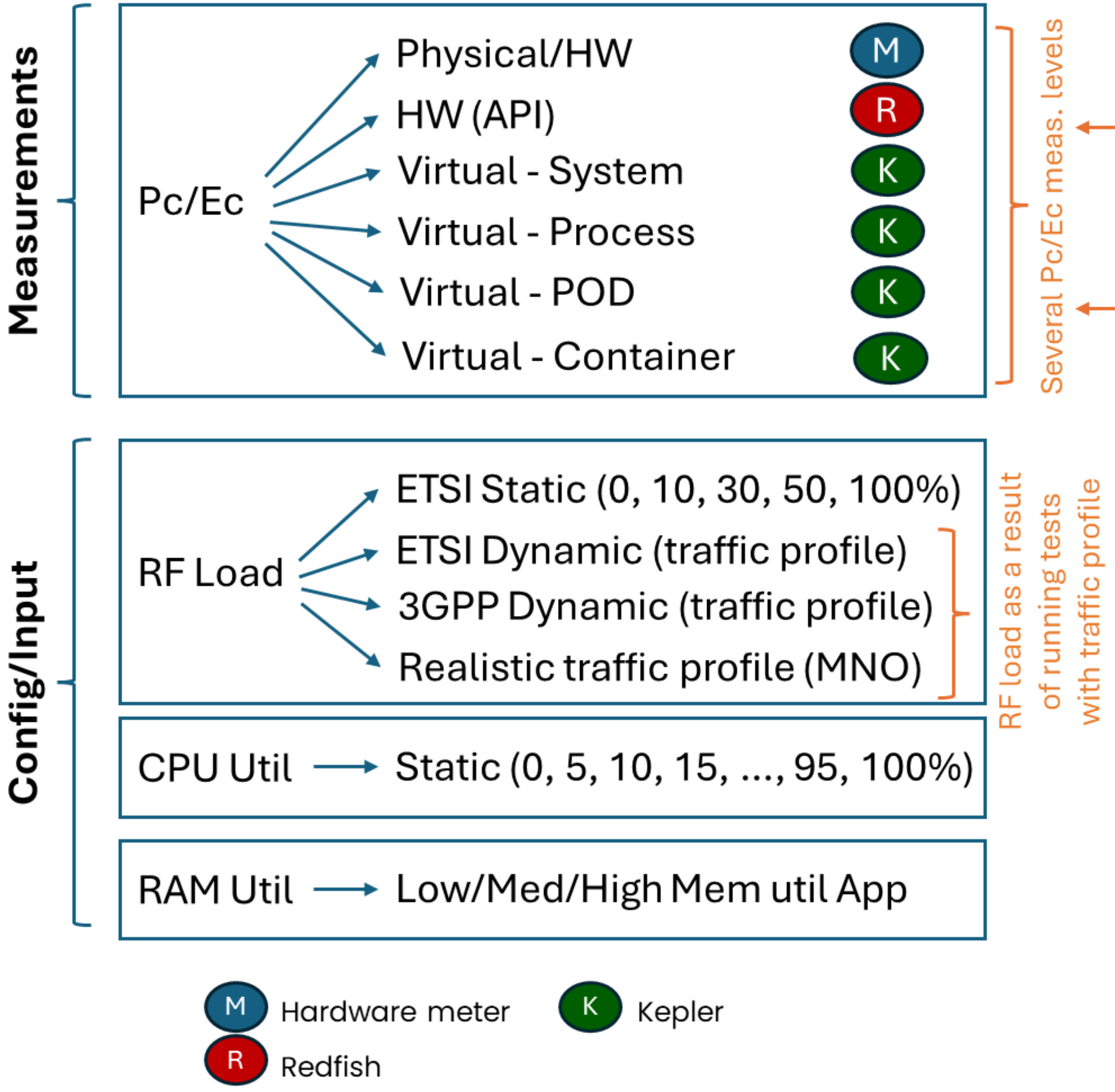
Power consumption for the whole server *(Kepler)*

Power consumption for the whole server *(Redfish)*

Power consumption for the whole server *(Electric socket)*

M	Hardware meter	K	Kepler
R	Redfish		

PC Measuremets - Configuration and Evaluation



Measurements

Pc/Ec

- Physical/HW M
- HW (API) R
- Virtual - System K
- Virtual - Process K
- Virtual - POD K
- Virtual - Container K

Several Pc/Ec meas. levels

Config/Input

RF Load

- ETSI Static (0, 10, 30, 50, 100%)
- ETSI Dynamic (traffic profile)
- 3GPP Dynamic (traffic profile)
- Realistic traffic profile (MNO)

RF load as a result of running tests with traffic profile

CPU Util → Static (0, 5, 10, 15, ..., 95, 100%)

RAM Util → Low/Med/High Mem util App

M Hardware meter

R Redfish

K Kepler

Process

(1) Server Measurements

set CPU → meas Pc, Ec, RAM util

set RAM → meas Pc, Ec, CPU util

generate heatmap/gap analisys (between R and K)

(2) Invoke RAN functions

set RF Load (static) → meas Pc, Ec, CPU, RAM

generate Pc/Ec model

(3) Dynamic traffic profile (data volume), for one traffic class

set traffic profile (ETSI, 3GPP, MNO-realistic)

→ meas Pc, Ec, CPU, Mem, data_vol/thrpt per UE

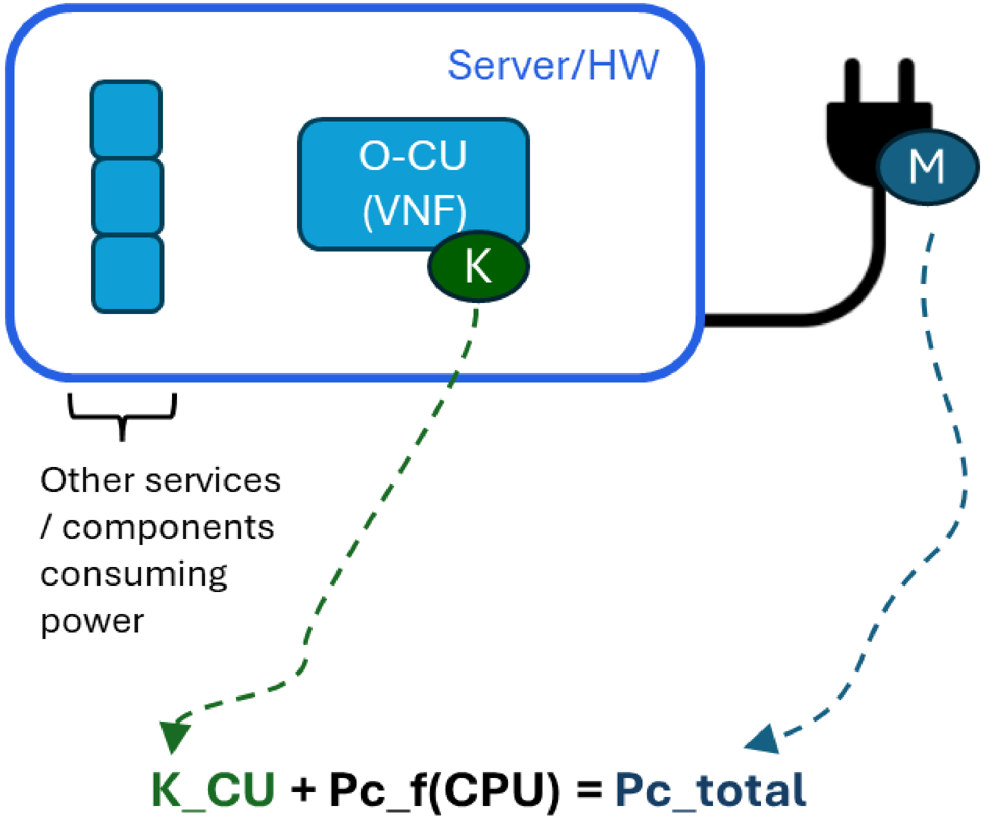
calculate EE_traffic_class = ∑ Data_volume_UE / EC

(4) Repeat (3) for various traffic classes

calculate EE_total = ∑ EE_traffic_class * factor

PC Measurements – Kepler vs Electric Socket

- Are both measurement types needed for reliable testing (i.e., Kepler and Socket-based)?
- Is there a specific function that binds measurement types so that we can measure one and estimate the other?



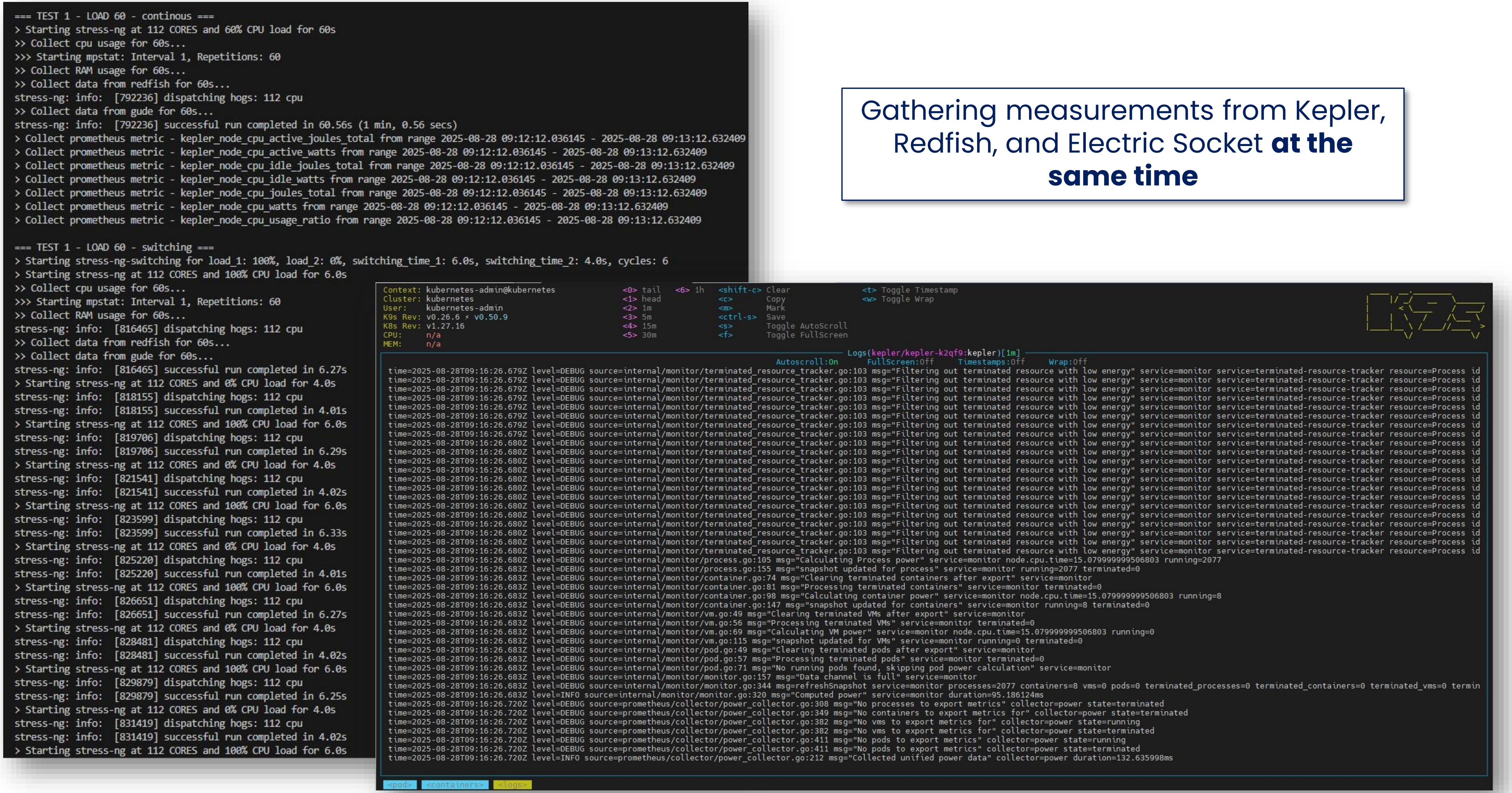
Server/HW
O-CU (VNF)
K
M
Other services / components consuming power
K_CU + Pc_f(CPU) = Pc_total

PC Measurements – Hardware Test Setups

	Dell R750 (enterprise grade server stored in i14y Lab's rack)	**Lenovo Legion 5 15ARH05** (consumer grade laptop)
CPU	2 x Intel Xeon Gold 6330N-2.2GHz (28 cores)	AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz (8 cores)
RAM	512 GB, 3200Mhz	32 GB, 3200Mhz
DISK	2xSSD 1.5TB	SSD 512GB
NIC	1 x 2-port 1GbE BCM5720 1 x 4-port 1 GbE Intel I350 4 x Intel XXV810 2-port 25GbE	1 x 1-port 1 GbE Wi-Fi 6 (802.11 a/b/g/n/ac/ax)
OS	Ubuntu 20.04.6 LTS	Ubuntu 22.04.5 LTS
Actual Hardware		
Power Measurement at Electric Socket	GUDE Electric Socket 	Shelly Smart Electric Socket

Power Consumption **Measurements Collection**

Gathering measurements from Kepler, Redfish, and Electric Socket **at the same time**

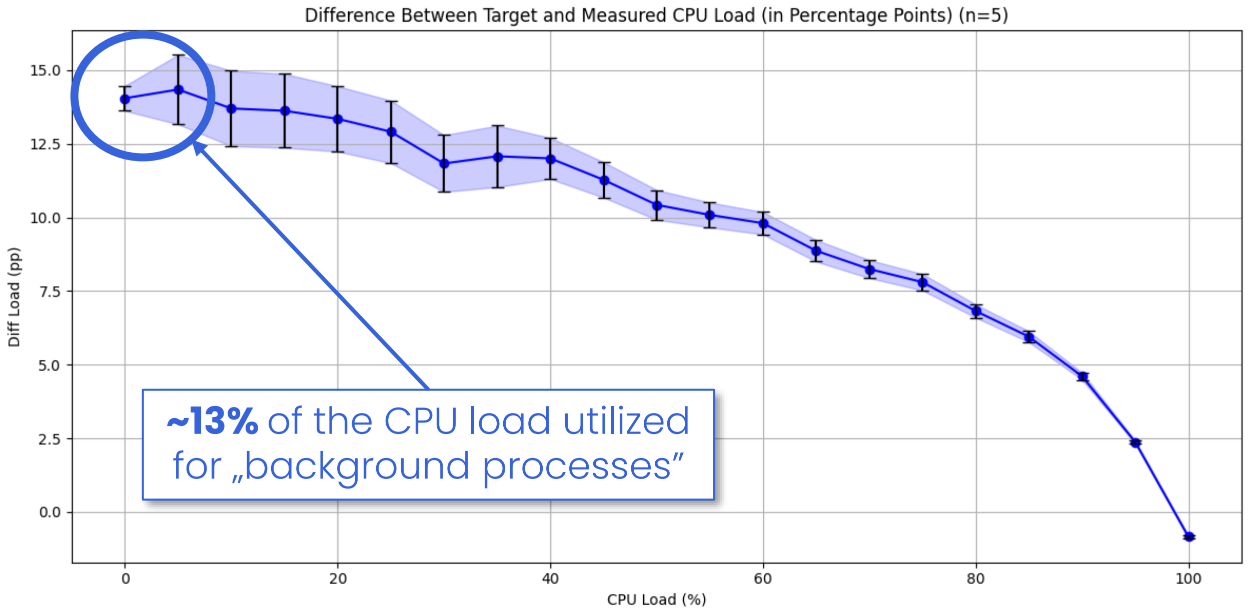


```
=== TEST 1 - LOAD 60 - continous ===
> Starting stress-ng at 112 CORES and 60% CPU load for 60s
>> Collect cpu usage for 60s...
>>> Starting mpstat: Interval 1, Repetitions: 60
>> Collect RAM usage for 60s...
>> Collect data from redfish for 60s...
stress-ng: info:  [792236] dispatching hogs: 112 cpu
>> Collect data from gude for 60s...
stress-ng: info:  [792236] successful run completed in 60.56s (1 min, 0.56 secs)
> Collect prometheus metric - kepler_node_cpu_active_joules_total from range 2025-08-28 09:12:12.036145 - 2025-08-28 09:13:12.632409
> Collect prometheus metric - kepler_node_cpu_active_watts from range 2025-08-28 09:12:12.036145 - 2025-08-28 09:13:12.632409
> Collect prometheus metric - kepler_node_cpu_idle_joules_total from range 2025-08-28 09:12:12.036145 - 2025-08-28 09:13:12.632409
> Collect prometheus metric - kepler_node_cpu_idle_watts from range 2025-08-28 09:12:12.036145 - 2025-08-28 09:13:12.632409
> Collect prometheus metric - kepler_node_cpu_joules_total from range 2025-08-28 09:12:12.036145 - 2025-08-28 09:13:12.632409
> Collect prometheus metric - kepler_node_cpu_watts from range 2025-08-28 09:12:12.036145 - 2025-08-28 09:13:12.632409
> Collect prometheus metric - kepler_node_cpu_usage_ratio from range 2025-08-28 09:12:12.036145 - 2025-08-28 09:13:12.632409

=== TEST 1 - LOAD 60 - switching ===
> Starting stress-ng-switching for load_1: 100%, load_2: 0%, switching_time_1: 6.0s, switching_time_2: 4.0s, cycles: 6
> Starting stress-ng at 112 CORES and 100% CPU load for 6.0s
>> Collect cpu usage for 60s...
>>> Starting mpstat: Interval 1, Repetitions: 60
>> Collect RAM usage for 60s...
stress-ng: info:  [816465] dispatching hogs: 112 cpu
>> Collect data from redfish for 60s...
>> Collect data from gude for 60s...
stress-ng: info:  [816465] successful run completed in 6.27s
> Starting stress-ng at 112 CORES and 0% CPU load for 4.0s
stress-ng: info:  [818155] dispatching hogs: 112 cpu
stress-ng: info:  [818155] successful run completed in 4.01s
> Starting stress-ng at 112 CORES and 100% CPU load for 6.0s
stress-ng: info:  [819706] dispatching hogs: 112 cpu
stress-ng: info:  [819706] successful run completed in 6.29s
> Starting stress-ng at 112 CORES and 0% CPU load for 4.0s
stress-ng: info:  [821541] dispatching hogs: 112 cpu
stress-ng: info:  [821541] successful run completed in 4.02s
> Starting stress-ng at 112 CORES and 100% CPU load for 6.0s
stress-ng: info:  [823599] dispatching hogs: 112 cpu
stress-ng: info:  [823599] successful run completed in 6.33s
> Starting stress-ng at 112 CORES and 0% CPU load for 4.0s
stress-ng: info:  [825220] dispatching hogs: 112 cpu
stress-ng: info:  [825220] successful run completed in 4.01s
> Starting stress-ng at 112 CORES and 100% CPU load for 6.0s
stress-ng: info:  [826651] dispatching hogs: 112 cpu
stress-ng: info:  [826651] successful run completed in 6.27s
> Starting stress-ng at 112 CORES and 0% CPU load for 4.0s
stress-ng: info:  [828481] dispatching hogs: 112 cpu
stress-ng: info:  [828481] successful run completed in 4.02s
> Starting stress-ng at 112 CORES and 100% CPU load for 6.0s
stress-ng: info:  [829879] dispatching hogs: 112 cpu
stress-ng: info:  [829879] successful run completed in 6.25s
> Starting stress-ng at 112 CORES and 0% CPU load for 4.0s
stress-ng: info:  [831419] dispatching hogs: 112 cpu
stress-ng: info:  [831419] successful run completed in 4.02s
> Starting stress-ng at 112 CORES and 100% CPU load for 6.0s
```

```
Context: kubernetes-admin@kubernetes        <0> tail   <6> 1h   <shift-c> Clear              <t> Toggle Timestamp
Cluster: kubernetes                         <1> head            <c>       Copy               <w> Toggle Wrap
User:    kubernetes-admin                   <2> 1m              <m>       Mark
K9s Rev: v0.26.6 ⚡ v0.50.9                  <3> 5m              <ctrl-s>  Save
K8s Rev: v1.27.16                           <4> 15m             <s>       Toggle AutoScroll
CPU:     n/a                                <5> 30m             <f>       Toggle FullScreen
MEM:     n/a
                                                              Logs(kepler/kepler-k2qf9:kepler)[1m]
                                                    Autoscroll:On    FullScreen:Off    Timestamps:Off    Wrap:Off
time=2025-08-28T09:16:26.679Z level=DEBUG source=internal/monitor/terminated_resource_tracker.go:103 msg="Filtering out terminated resource with low energy" service=monitor service=terminated-resource-tracker resource=Process id
time=2025-08-28T09:16:26.679Z level=DEBUG source=internal/monitor/terminated_resource_tracker.go:103 msg="Filtering out terminated resource with low energy" service=monitor service=terminated-resource-tracker resource=Process id
time=2025-08-28T09:16:26.679Z level=DEBUG source=internal/monitor/terminated_resource_tracker.go:103 msg="Filtering out terminated resource with low energy" service=monitor service=terminated-resource-tracker resource=Process id
time=2025-08-28T09:16:26.679Z level=DEBUG source=internal/monitor/terminated_resource_tracker.go:103 msg="Filtering out terminated resource with low energy" service=monitor service=terminated-resource-tracker resource=Process id
time=2025-08-28T09:16:26.679Z level=DEBUG source=internal/monitor/terminated_resource_tracker.go:103 msg="Filtering out terminated resource with low energy" service=monitor service=terminated-resource-tracker resource=Process id
time=2025-08-28T09:16:26.679Z level=DEBUG source=internal/monitor/terminated_resource_tracker.go:103 msg="Filtering out terminated resource with low energy" service=monitor service=terminated-resource-tracker resource=Process id
time=2025-08-28T09:16:26.679Z level=DEBUG source=internal/monitor/terminated_resource_tracker.go:103 msg="Filtering out terminated resource with low energy" service=monitor service=terminated-resource-tracker resource=Process id
time=2025-08-28T09:16:26.679Z level=DEBUG source=internal/monitor/terminated_resource_tracker.go:103 msg="Filtering out terminated resource with low energy" service=monitor service=terminated-resource-tracker resource=Process id
time=2025-08-28T09:16:26.680Z level=DEBUG source=internal/monitor/terminated_resource_tracker.go:103 msg="Filtering out terminated resource with low energy" service=monitor service=terminated-resource-tracker resource=Process id
time=2025-08-28T09:16:26.680Z level=DEBUG source=internal/monitor/terminated_resource_tracker.go:103 msg="Filtering out terminated resource with low energy" service=monitor service=terminated-resource-tracker resource=Process id
time=2025-08-28T09:16:26.680Z level=DEBUG source=internal/monitor/terminated_resource_tracker.go:103 msg="Filtering out terminated resource with low energy" service=monitor service=terminated-resource-tracker resource=Process id
time=2025-08-28T09:16:26.680Z level=DEBUG source=internal/monitor/terminated_resource_tracker.go:103 msg="Filtering out terminated resource with low energy" service=monitor service=terminated-resource-tracker resource=Process id
time=2025-08-28T09:16:26.680Z level=DEBUG source=internal/monitor/terminated_resource_tracker.go:103 msg="Filtering out terminated resource with low energy" service=monitor service=terminated-resource-tracker resource=Process id
time=2025-08-28T09:16:26.680Z level=DEBUG source=internal/monitor/terminated_resource_tracker.go:103 msg="Filtering out terminated resource with low energy" service=monitor service=terminated-resource-tracker resource=Process id
time=2025-08-28T09:16:26.680Z level=DEBUG source=internal/monitor/terminated_resource_tracker.go:103 msg="Filtering out terminated resource with low energy" service=monitor service=terminated-resource-tracker resource=Process id
time=2025-08-28T09:16:26.680Z level=DEBUG source=internal/monitor/terminated_resource_tracker.go:103 msg="Filtering out terminated resource with low energy" service=monitor service=terminated-resource-tracker resource=Process id
time=2025-08-28T09:16:26.680Z level=DEBUG source=internal/monitor/terminated_resource_tracker.go:103 msg="Filtering out terminated resource with low energy" service=monitor service=terminated-resource-tracker resource=Process id
time=2025-08-28T09:16:26.680Z level=DEBUG source=internal/monitor/terminated_resource_tracker.go:103 msg="Filtering out terminated resource with low energy" service=monitor service=terminated-resource-tracker resource=Process id
time=2025-08-28T09:16:26.680Z level=DEBUG source=internal/monitor/terminated_resource_tracker.go:103 msg="Filtering out terminated resource with low energy" service=monitor service=terminated-resource-tracker resource=Process id
time=2025-08-28T09:16:26.680Z level=DEBUG source=internal/monitor/terminated_resource_tracker.go:103 msg="Filtering out terminated resource with low energy" service=monitor service=terminated-resource-tracker resource=Process id
time=2025-08-28T09:16:26.680Z level=DEBUG source=internal/monitor/terminated_resource_tracker.go:103 msg="Filtering out terminated resource with low energy" service=monitor service=terminated-resource-tracker resource=Process id
time=2025-08-28T09:16:26.680Z level=DEBUG source=internal/monitor/process.go:105 msg="Calculating Process power" service=monitor node.cpu.time=15.079999999506803 running=2077
time=2025-08-28T09:16:26.683Z level=DEBUG source=internal/monitor/process.go:155 msg="snapshot updated for process" service=monitor running=2077 terminated=0
time=2025-08-28T09:16:26.683Z level=DEBUG source=internal/monitor/container.go:74 msg="Clearing terminated containers after export" service=monitor
time=2025-08-28T09:16:26.683Z level=DEBUG source=internal/monitor/container.go:81 msg="Processing terminated containers" service=monitor terminated=0
time=2025-08-28T09:16:26.683Z level=DEBUG source=internal/monitor/container.go:98 msg="Calculating container power" service=monitor node.cpu.time=15.079999999506803 running=8
time=2025-08-28T09:16:26.683Z level=DEBUG source=internal/monitor/container.go:147 msg="snapshot updated for containers" service=monitor running=8 terminated=0
time=2025-08-28T09:16:26.683Z level=DEBUG source=internal/monitor/vm.go:49 msg="Clearing terminated VMs after export" service=monitor
time=2025-08-28T09:16:26.683Z level=DEBUG source=internal/monitor/vm.go:56 msg="Processing terminated VMs" service=monitor terminated=0
time=2025-08-28T09:16:26.683Z level=DEBUG source=internal/monitor/vm.go:69 msg="Calculating VM power" service=monitor node.cpu.time=15.079999999506803 running=0
time=2025-08-28T09:16:26.683Z level=DEBUG source=internal/monitor/vm.go:115 msg="snapshot updated for VMs" service=monitor running=0 terminated=0
time=2025-08-28T09:16:26.683Z level=DEBUG source=internal/monitor/pod.go:49 msg="Clearing terminated pods after export" service=monitor
time=2025-08-28T09:16:26.683Z level=DEBUG source=internal/monitor/pod.go:57 msg="Processing terminated pods" service=monitor terminated=0
time=2025-08-28T09:16:26.683Z level=DEBUG source=internal/monitor/pod.go:71 msg="No running pods found, skipping pod power calculation" service=monitor
time=2025-08-28T09:16:26.683Z level=DEBUG source=internal/monitor/monitor.go:157 msg="Data channel is full" service=monitor
time=2025-08-28T09:16:26.683Z level=DEBUG source=internal/monitor/monitor.go:344 msg=refreshSnapshot service=monitor processes=2077 containers=8 vms=0 pods=0 terminated_processes=0 terminated_containers=0 terminated_vms=0 termin
time=2025-08-28T09:16:26.683Z level=INFO source=internal/monitor/monitor.go:320 msg="Computed power" service=monitor duration=95.186124ms
time=2025-08-28T09:16:26.720Z level=DEBUG source=prometheus/collector/power_collector.go:308 msg="No processes to export metrics" collector=power state=terminated
time=2025-08-28T09:16:26.720Z level=DEBUG source=prometheus/collector/power_collector.go:349 msg="No containers to export metrics for" collector=power state=terminated
time=2025-08-28T09:16:26.720Z level=DEBUG source=prometheus/collector/power_collector.go:382 msg="No vms to export metrics for" collector=power state=running
time=2025-08-28T09:16:26.720Z level=DEBUG source=prometheus/collector/power_collector.go:382 msg="No vms to export metrics for" collector=power state=terminated
time=2025-08-28T09:16:26.720Z level=DEBUG source=prometheus/collector/power_collector.go:411 msg="No pods to export metrics" collector=power state=running
time=2025-08-28T09:16:26.720Z level=DEBUG source=prometheus/collector/power_collector.go:411 msg="No pods to export metrics" collector=power state=terminated
time=2025-08-28T09:16:26.720Z level=INFO source=prometheus/collector/power_collector.go:212 msg="Collected unified power data" collector=power duration=132.635998ms

<pod>  <containers>  <logs>
```

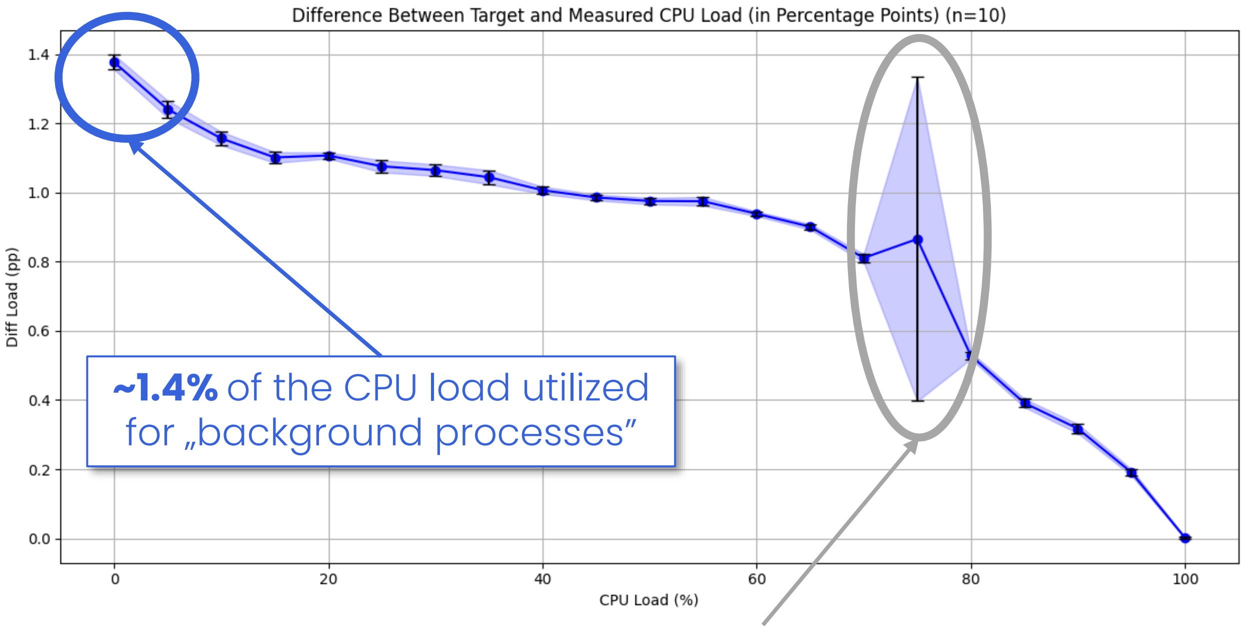
CPU Load - Enforced vs Measured

Dell R750 - Server



~13% of the CPU load utilized for „background processes”

Lenovo Legion 5 15ARH05



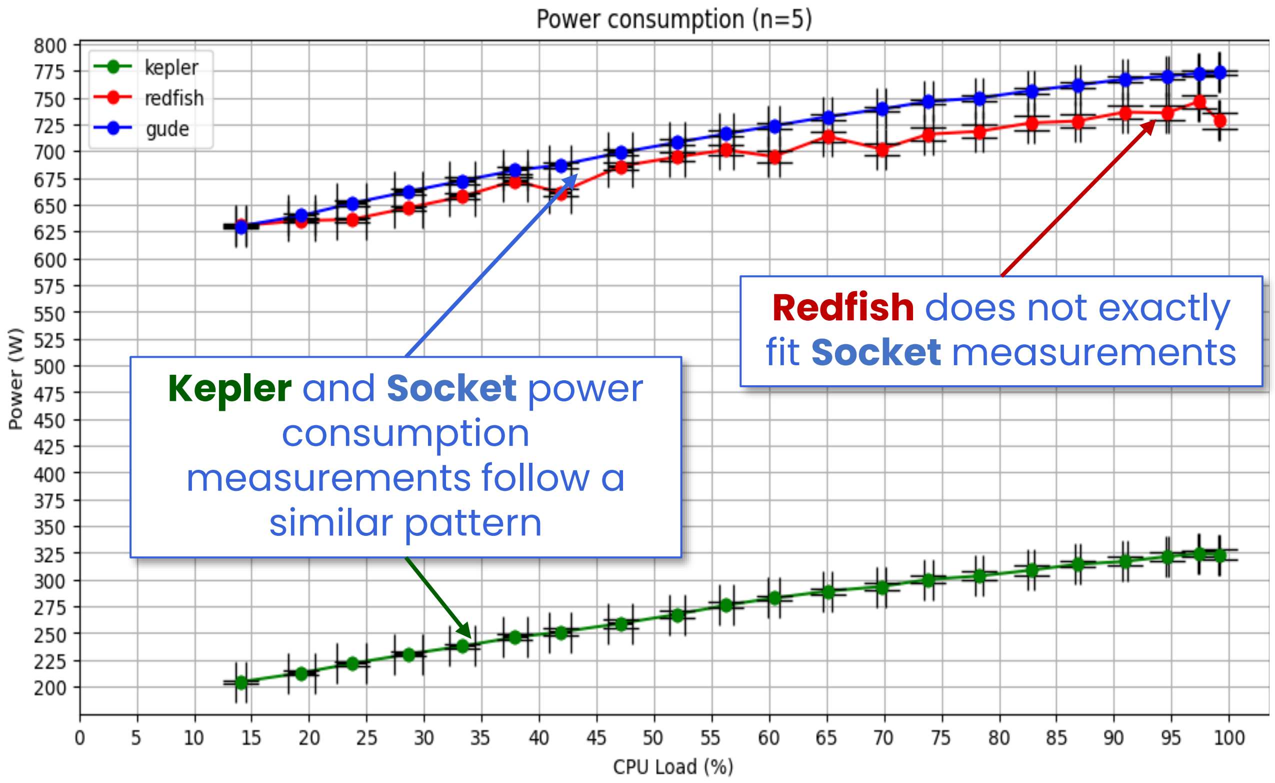
~1.4% of the CPU load utilized for „background processes”

Some anomalies were observed for the laptop measurement, but their effect is minimal

The **difference** between the **CPU load enforced by the stress-test application** and the **actual CPU load** is related to other processes

Static CPU Load - Kepler, Redfish, and Electric Socket

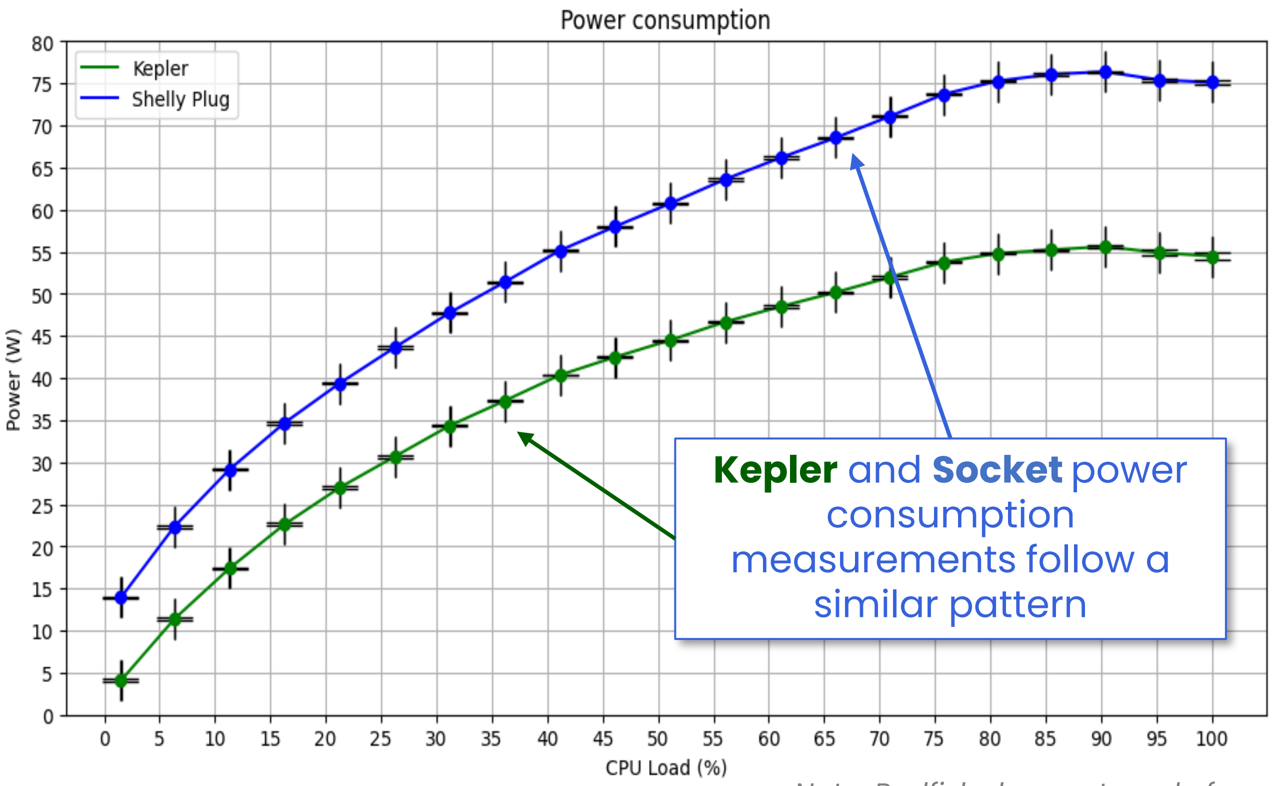
Dell R750 - Server



Redfish does not exactly fit **Socket** measurements

Kepler and **Socket** power consumption measurements follow a similar pattern

Lenovo Legion 5 15ARH05

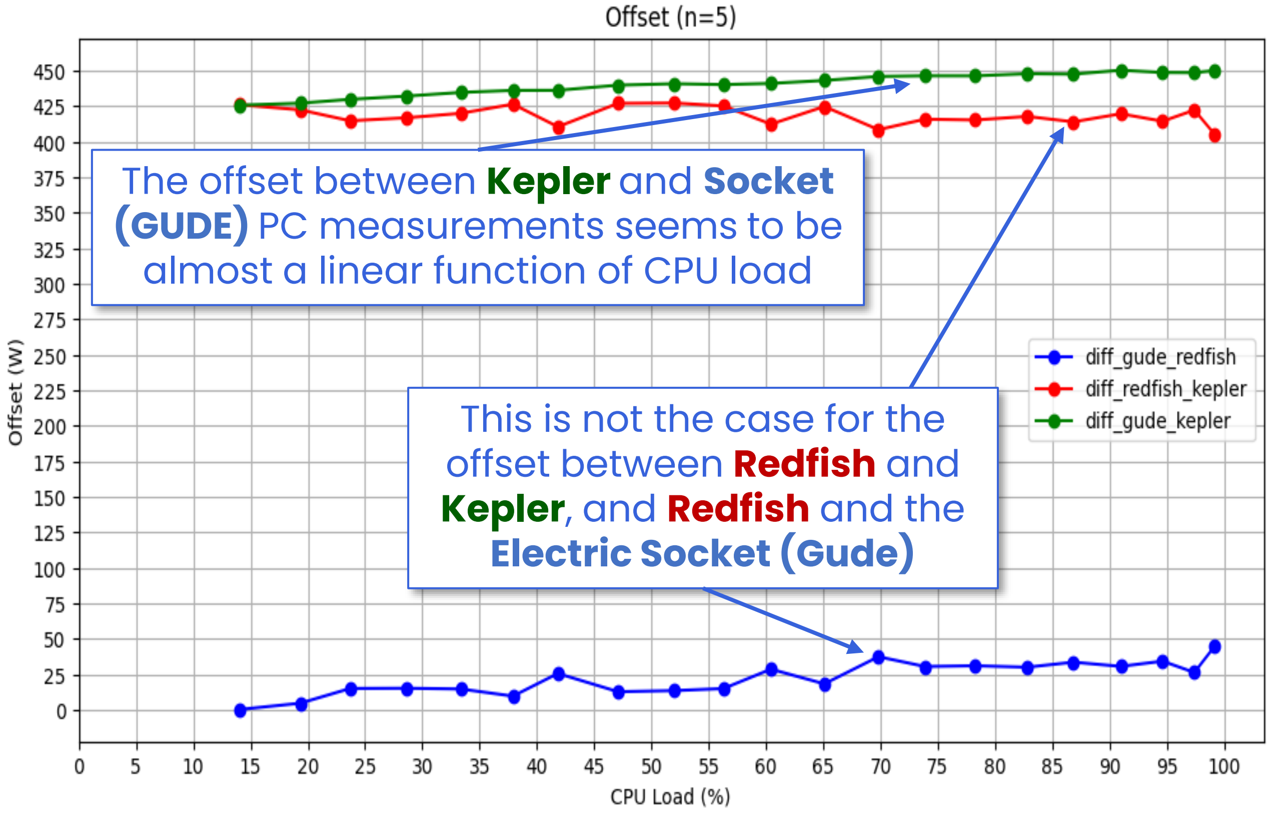


Kepler and **Socket** power consumption measurements follow a similar pattern

Note: Redfish does not apply for consumer grade laptops

Static CPU Load - Offset Between Kepler, Redfish, and Electric Socket

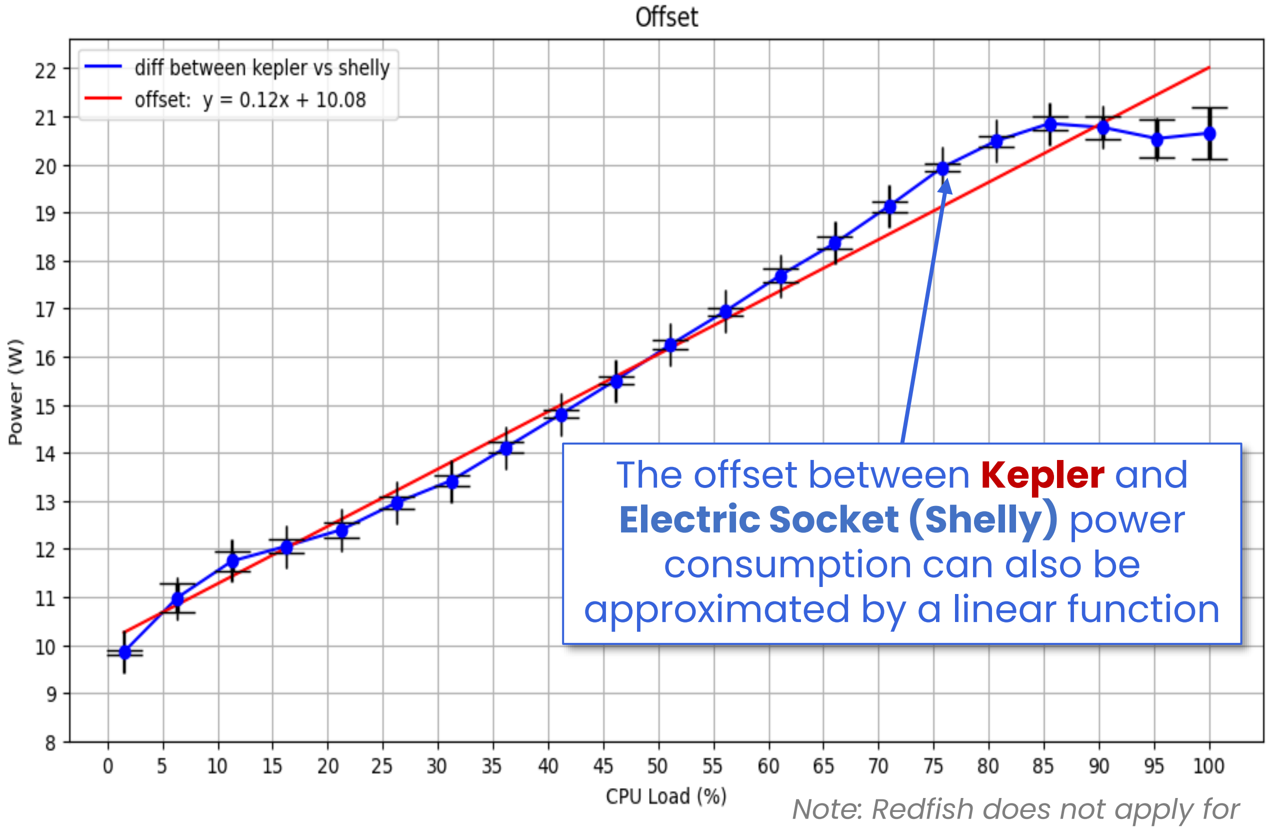
Dell R750



The offset between **Kepler** and **Socket (GUDE)** PC measurements seems to be almost a linear function of CPU load

This is not the case for the offset between **Redfish** and **Kepler**, and **Redfish** and the **Electric Socket (Gude)**

Lenovo Legion 5 15ARH05



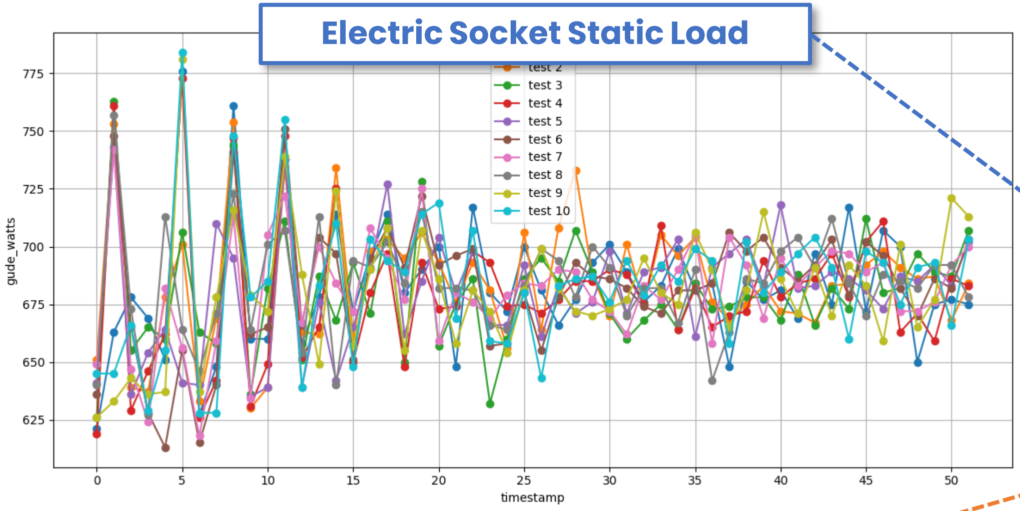
The offset between **Kepler** and **Electric Socket (Shelly)** power consumption can also be approximated by a linear function

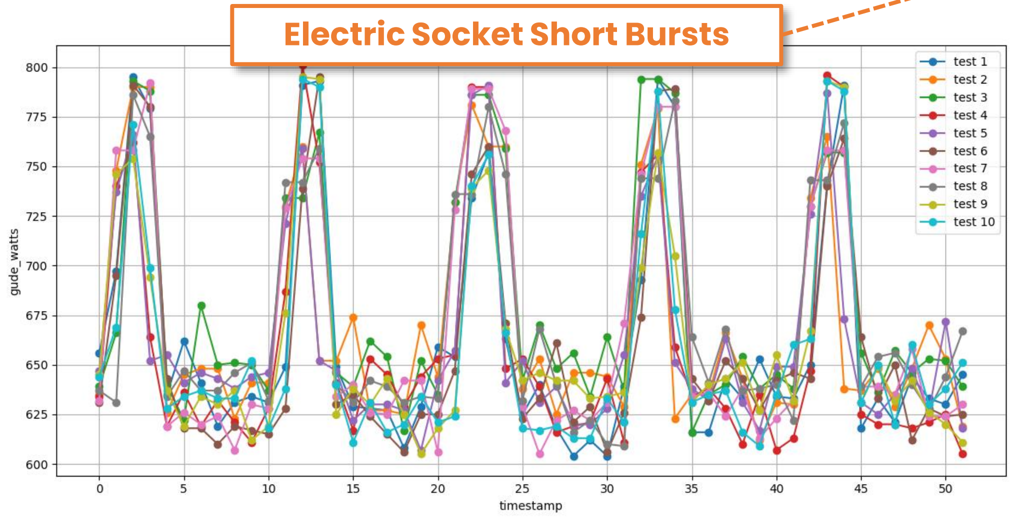
Note: Redfish does not apply for consumer grade laptops

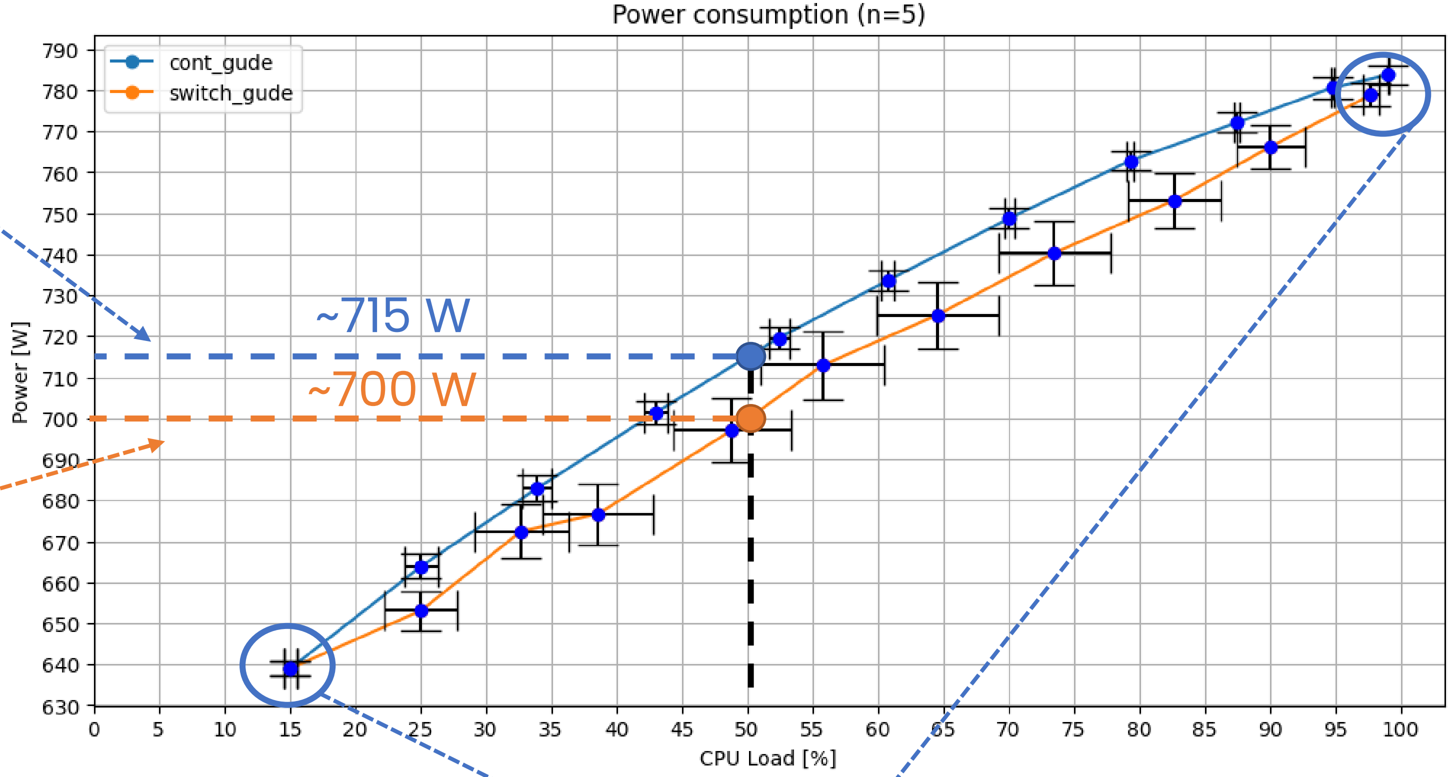
In both cases, the **offset** depends on the **CPU load** and is **close to linear** characteristics

Static vs. Short Bursts CPU Load (Dell R750)

We compared the PC measurements under **static CPU load**, at 50% CPU load, against the **short bursts:** to achieve 50% CPU load, in 10s cycles → 5s of 100% CPU load, then 5s of 0% CPU load







~700 W: 640 x 0.5 + 780 x 0.5=710 W

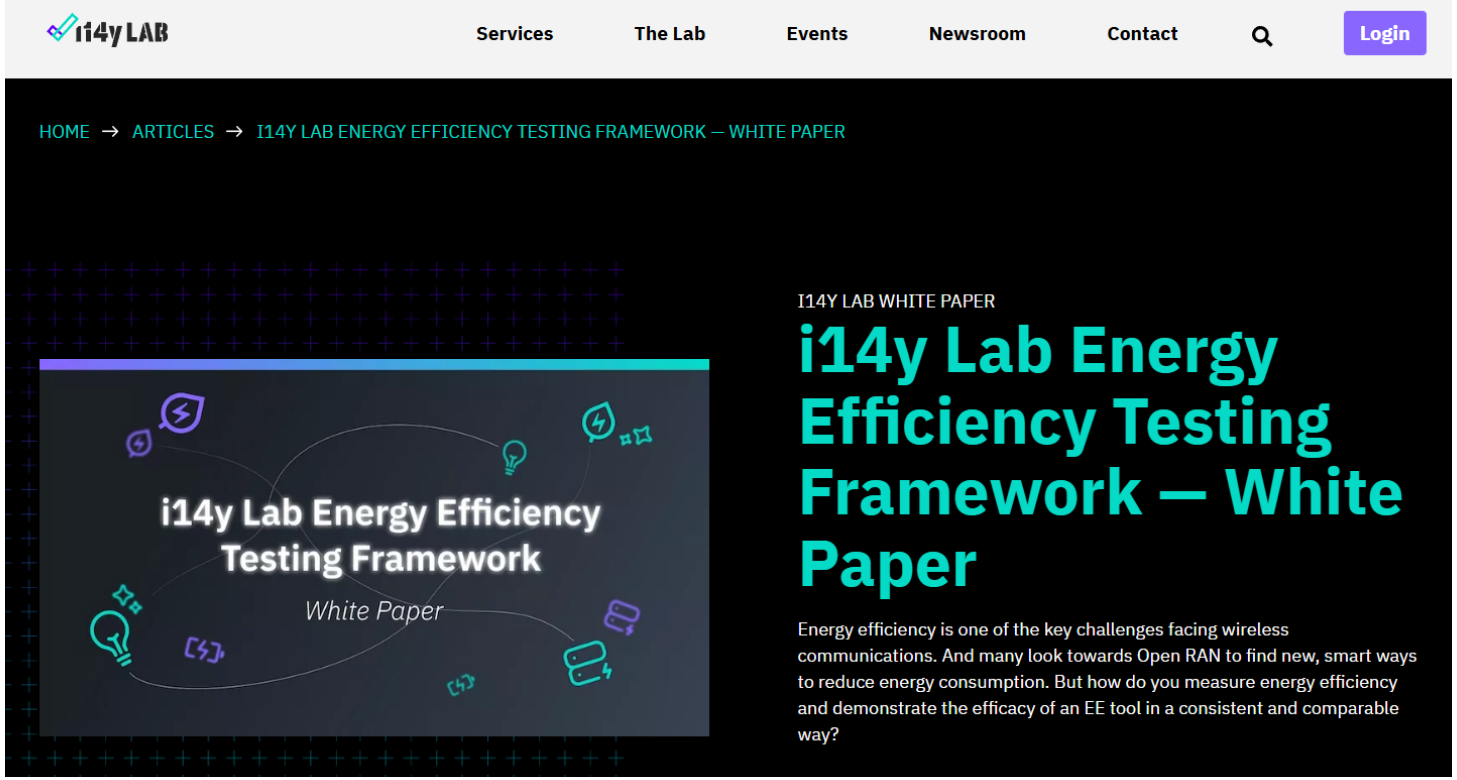
A mismatch of 10W can be caused by a large results spread

Conclusions

Conclusions

- Different standardization bodies address distinct aspects of energy saving, and there is a need for a unified E2E EE testing framework
- The main challenges are related to the standardized methodologies for the VNFs, and various deployment configurations for O-RAN components
- New metrics and benchmarks are essential to the framework. They should take into account both the energy-saving and QoS/performance goals
- The framework's design and execution will involve developing a standardized core testing procedure adaptable to various scenarios.
- The initial focus will be on the Component-level measurements for EE and Global/feature based testing for EE - they are feasible for a lab environment
- Power consumption measurements for a consumer Laptop and a Dell Server were conducted under a static CPU load forced by a third-party APP.
- The gap between Kepler and the Electric Socket follows an almost linear function.
- The next steps cover power consumption measurements of individual PODs, e.g., measurements for RIC and xApps operating under the simulated RAN environment.

White paper



HOME → ARTICLES → I14Y LAB ENERGY EFFICIENCY TESTING FRAMEWORK — WHITE PAPER

I14Y LAB WHITE PAPER

i14y Lab Energy Efficiency Testing Framework — White Paper

Energy efficiency is one of the key challenges facing wireless communications. And many look towards Open RAN to find new, smart ways to reduce energy consumption. But how do you measure energy efficiency and demonstrate the efficacy of an EE tool in a consistent and comparable way?

White paper QR code

https://www.i14y-lab.com/article/i14y-lab-energy-efficiency-testing-framework-white-paper

Meet Rimedo Labs at the Marketplace!

#SUMMIT25 · SEPTEMBER 2025

RIMEDO LABS

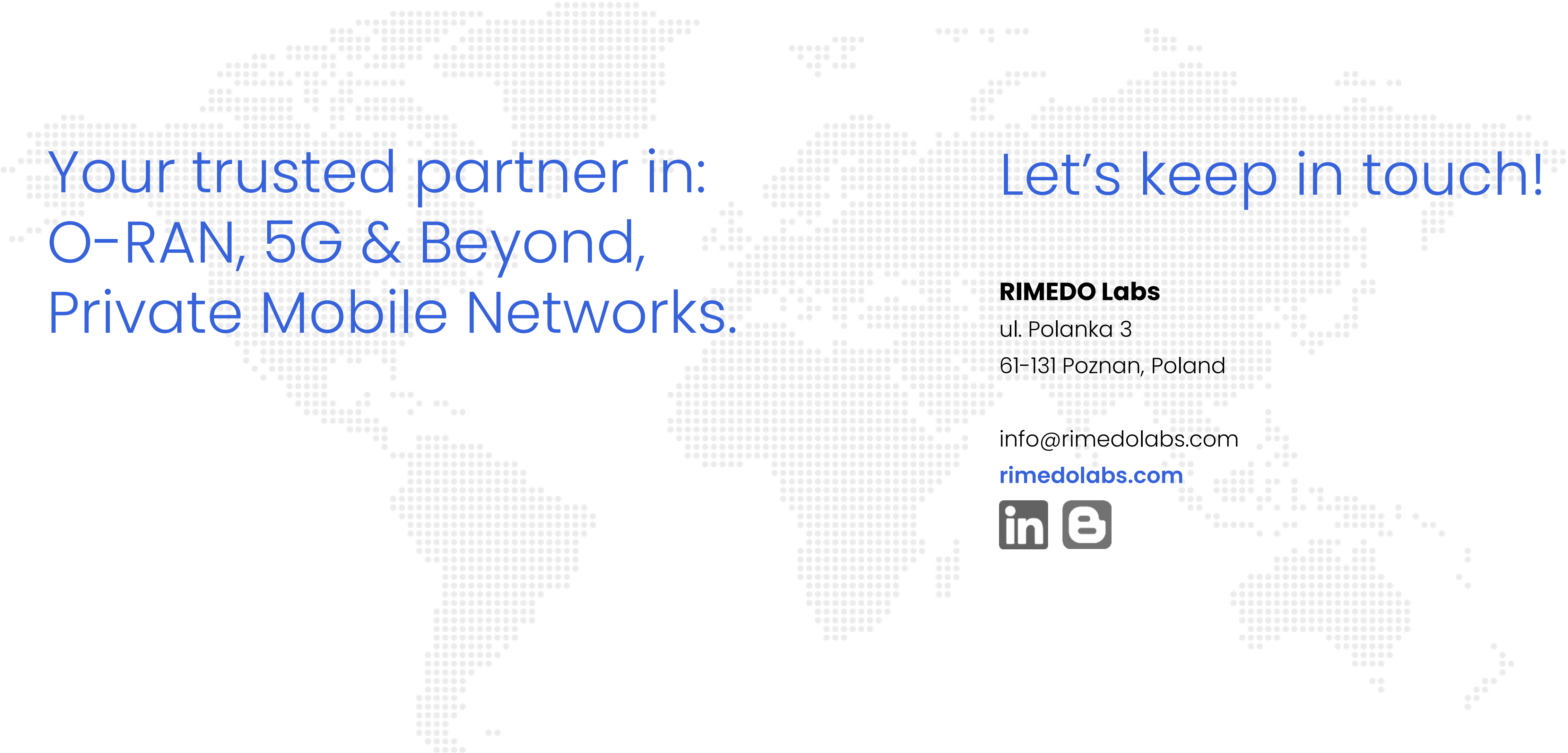
#SUMMIT25
Implement & Consolidate

i14y LAB

Special Thanks

T LABS

i14y LAB



Your trusted partner in: O-RAN, 5G & Beyond, Private Mobile Networks.

Let’s keep in touch!

RIMEDO Labs
ul. Polanka 3
61-131 Poznan, Poland

info@rimedolabs.com
rimedolabs.com

The information contained herein is the property of RIMEDO and is provided only if it is not disclosed, directly or indirectly to a third party, or used for purposes other than those for which it was prepared.

All information discussed in the document is provided "as is" and RIMEDO makes no warranty that this information is fit for purpose. Users use this information at their own risk and responsibility.